# Leveraging Temporal Dynamics with Neural-Inspired Sensing and Control

Thomas Leonard Mohren

A DISSERTATION
SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy
University of Washington
2020

READING COMMITTEE
Steven Brunton, Co-Chair
Thomas Daniel, Co-Chair
Sawyer Fuller

PROGRAM AUTHORIZED TO OFFER DEGREE:
MECHANICAL ENGINEERING

University of Washington

## Abstract

# Leveraging Temporal Dynamics with Neural-Inspired Sensing and Control

Thomas Leonard Mohren

Co-Chairs of the supervisory committee:
Steven Brunton
Department of Mechanical Engineering
&
Thomas Daniel
Department of Biology

Flying insects are known for their fast and robust control while being challenged with sensory delays, an unsteady environment and by having limited computation power. One important component of this robust control is the sensory feedback from arrays of mechanoreceptors found on wings and wing-derived halteres. By combining structural simulation with experimentally derived neural processing models we gain insight into mechanisms involved in detecting body rotation by mechanosensory oscillating appendages. I found that it is the combination of the temporal encoding of strain by mechanoreceptors with the spatial layout of the sensors on the wing that allows for the detection of minute rotation-induced differences in wing deformation. Although several studies have presented analytical models of haltere deformation, a high fidelity Finite Element Analysis (FEA) revealed novel deformation modes resulting from haltere asymmetry. Using a neuronal spiking model on the strain from the FEA simulations, we found spike timing along the circumference of the haltere base changed with body rotation. The timing change was larger than the experimentally-observed timing variability of the individual mechanosensors at all but the top and bottom of the haltere base. This gives credence to the hypothesis of timing-based detection and encoding of rotation, in addition to the recruitment based detection commonly described in the literature. The importance of timing in mechanosensation in insect flight led to the investigation of a timing-based feedforward controller that I tested on a the partially denied inverted pendulum. Using this timing-based feedforward controller, a close-to-optimal controller could be learned in much fewer trials than a brute force search. This neural-inspired controller holds promise for engineered systems where the number of trials is limited and state measurements are denied in parts of it's state space.

# Contents

# Acknowledgements

Ever since losing my passport on that first flight to Seattle luck has been on my side. I found incredibly welcoming colleagues in the Daniel lab, I was adopted into the Olson family, and it turned out my passport slipped from my pocket to that space between the seats. Working on my PhD has been an incredibly fulfilling experience and I have met many wonderful people along the way. I cannot put to words how thankful I am of your friendship and kindness.

Nothing would have come of this thesis without my advisors Steve Brunton and Tom Daniel, thank you for your incredibly positive attitude, support and encouragement. I have learned so much about the scientific process and how to effectively manage time (and advisors). Thank you Sawyer Fuller and John Tuthill for serving on my committee and for your helpful interactions. Submitting publications was almost enjoyable thanks to my other co-authors Bing Brunton, Jessica Fox, Annika Eberle and Per Reinhall. Thanks Tanvi Deora, Eatai Roth, Brandon Pratt and Bradley Dickerson for helpful discussions and for reviewing my publications before submission. Thank you, lab members of the Daniel, Riffel, Brunton and Brunton labs for being such an open and welcoming crowd. I also want to thank Alison Mehravari and Wanwisa Kisalang for mental support and helpful advice.

Thank you friends, roommates and climbing partners for keeping me sane and safe. Thanks Eleanor for cheering me up and making every day more wonderful. Thanks mom, dad and Jim, I promise I'll visit again soon! Lastly, a special thanks to Kelly and Mary-Ann, for the countless dinners and for dragging ping-pong tables around town.

# Introduction

Millions of years of evolution has led to an incredible diversity of natural systems. Animals and plants have adapted to the environment in striking fashion, performing far outside the limits of current-day engineered systems. Yet as humans we design and build systems that have to deal with similar challenges, such as control in turbulence and time-varying under-actuation. With decreased cost of electronics and new computational techniques, we are at the forefront of deploying robotics in unstructured environments. Unstructured environments is where biological systems excel and where engineered systems traditionally have failed. In the quest for technological progress, we stand to gain by learning from the example solutions that nature provides us.

Biological systems have served as inspiration for human technology throughout the centuries. Famously, Leonardo da Vinci designed flying machines based on bird wings in the late fifteen hundreds. On a smaller scale, new technologies for dry adhesion emerged from research on gecko toe pads. Neural networks, the most powerful classification techniques at time of writing, took inspiration from neural connections in the brain. Certainly, great developments have originated from the field of biomimicry. Yet one must keep in mind that the properties (such as morphology) of biological systems have developed under constraints and limitations that are not relevant in the engineered system. Therefore, to fully utilize the features we observe in animals and plants, we require a mechanistic understanding. It is the new mechanism that we can adapt into synthetic design. Using these principles from nature, we can design synthetic systems in novel ways.

A particularly striking example of biological virtuosity is the flight of insects. Flies move in seemingly erratic and random ways, but try to swat one, and you'll often miss. When replaying slow motion video of insect flight, it becomes clear insects rely on an intricate control network. One observes that the gaze of flies is stabilized by rapid feedback to the neck muscles [1], and flies recover from gusts and disturbances remarkably quickly [2] . But perhaps more remarkable is the quick adaptation to a change in system properties. When a bee's wing is severely damaged its wing amplitude changes to compensate, and its flight performance is hardly altered [3]. This high performance with wide robustness to disturbances and uncertainty in its system is unrivaled in engineering.

This is all the more remarkable considering the challenges in insect flight: The environment is highly unsteady and unpredictable. Flight is inherently unstable around the pitch axis [4]. Neural transduction and computation lead to time delays on the same order as the dynamics of flight. Furthermore, computational power by the small number of neurons in the insect brain is limited.

The remarkable performance combined with the stringent limitations would be impossible to recreate with current-day engineering technology. By understanding the principles that underly this insect behavior we can advance technology, eventually enabling us to design engineered systems in similar situations.

In insect evolution, flight has proven to be an incredibly effective adaptation. One of the defining traits of the different insect orders is the differentiation in their wing structure. The ancestral insect had four wings, much like dragonflies do today. In Beetles, the front wings turned into protective shells, and in moths and wasps the four wings interlock. In the order of the true flies *Diptera* however, the hind wings evolved into sensory organs called halteres. These hammer-like structures still beat at same frequency as the wings do, but their deformation is solely a function of gyroscopic forces. Wings also experience these gyroscopic forces, and it is these forces that carry a signature of body rotation. The resultant local deformations and strains are encoded by mechanoreceptors which are found in abundance on both the wings and halteres.

Flapping flight is inherently unstable, requiring fast feedback to be stabilized. Although insects rely on vision during flight, it takes over 40 ms for a visual stimulus to lead to a behavioral response [5], [6]. This latency is too slow to be used for stabilization [4]. Instead, insects rely on the much faster sensing by mechanoreceptors called campaniform sensilla. They are found in concentrated fields at the base of the wing, and radiating outwards along the veins of the wing. Halteres have arrays of campaniform sensilla at their base with similar encoding properties to those on the wings of moths. These campaniform sensilla on wings and halteres are spiking neurons; once triggered, the campaniform sends an action potential down its axon. Such a trigger is followed by a 3 millisecond refractory period before another spike can occur, resulting in a spike only once or twice a wingbeat. Both wing and haltere campaniform sensilla have direct innervations to the flight muscles [7], minimizing latency from sensing to actuation.

How the campaniform turns a temporal strain signal into an action potential – how it encodes strain – can be measured by recording from the neuron while stimulating the appendage, generally with band-limited white noise [8]. Through exposing the campaniform to a large set of stimuli, and recording the stimuli that trigger spikes, we obtain an *ensemble* of spike triggers (the STE, Spike Triggered Ensemble). Using different types of analysis of this ensemble, we can create probabilistic firing models of the campaniform sensilla. The commonly used average of the ensemble is referred to as the Spike Triggered Average (STA). Next, we can project the STA onto the original stimulus to obtain a similarity measure. This similarity between the original stimuli and the STA indicates the specificity of the neuron, i.e. how close does the stimulus need to match the desired feature to elicit a spike response. By plotting a histogram of the similarity and fitting a function, we obtain the so called non-linear decision function (NLD). The NLD provides the probability of a spike occuring, given a certain similarity. A similar but potentially more accurate model can be obtained by looking at different features of the ensemble simultaneously through principal component analysis (PCA) [9]. Interestingly, the encoding properties of campaniform sensilla on the wings of hawkmoths and on the halteres of craneflies are remarkably similar [8], [10]. Notably, when provided the exact same stimulus, the timing of the elicited spikes shows very little variation

($< 0.2$ ms).

The discrete event-triggered and large arrays of sensors are remarkably different from single-sensor continuous measurements common in engineered systems. How does this contribute to the sensing capabilities of flying insects, and if so, can we harness similar benefits by using neural encoding on sensory data in engineered systems? A major unsolved challenge in understanding biological control is how the environment and internal state are represented in the nervous system. However, we can come up with computational models designed after the input-output relationship observed in behavioral experiments. Models come in different gradations of complexity, trading off ability to predict specific systems versus a more general applicability. We can use these models to gain understanding and to discover broad mechanistic principles of control in living systems.

Modern day engineered systems often rely on only a few sensors, such as accelerometers for stabilization and GPS measurements for navigation. Such systems require high accuracy, filtering and reliability of those individual sensors. To address this, mission critical sensors like angle of attack sensors on aircraft often have a secondary sensor for redundancy. Alternatively, state estimators can combine measurements of single or multiple sensors with an internal model of the system. State estimators like Kalman filters are commonly used to reject noise and smoothen feedback control. By having knowledge of the model, estimates of model uncertainty and sensor noise, Kalman estimators simulate how the system would propagate during one time step and provides an updated estimate of the full state. Proven to be extremely useful, state estimators require an accurate model of the system however. The paradigm of having only a few sensors and having an accurate system model is very different from the large arrays of sensors and limited computational power in biology.

Modern control systems generally consist of several layers. Higher-level decision making and planning outlines trajectories from one state to the next. Without noise and perfect model information it would be possible to control such a trajectory entirely in feed-forward manner; the control action belonging to a certain change in state is found by inverting the system dynamics. In practice, this fails for two reasons: the system will drift due to noise in the system model or due to external disturbances. The solution is feedback control; sense and estimate the state to determine an appropriate control action to compensate for noise and disturbances. Feedback is ubiquitous in engineering and biology, and classical control theory provides many useful tools for designing and analyzing behavior of linear systems. Interestingly, the development of the proportional-integral-derivative (PID) controller is an early example of biology-inspired control, Minorsky based the PID controller on how captains steered ships [11]. However, most systems are not linear, and in many systems we do not have access to an accurate model of the system dynamics.

Recent efforts in control theory have focused on learning and decision making through data-driven methods. A tree of mathematical functions can be used to find controllers for high dimensional fluid control problems [12]–[14]. Neural networks have been incorporated with model predictive control, circumventing the need for a mathematical system model [15]–[17]. Iterative learning control and reinforcement learning are model free, focusing on learning controller through experimental or simulated trials [18]–[20]. These powerful approaches have enabled control of systems outside of the scope of classical control. Yet they can be computationally intractable, requiring many trials or accurate simulations. The generalizable nature of these methods can prevent them from finding lower-dimensional representations, potentially leading to longer training times or failed convergence.

In this thesis I investigate the neuromechanics of sensory feedback in insect flight control with the aim of developing new methods for modern control challenges. My research is motivated by the following overarching questions:

- How does neural encoding by individual mechanoreceptors contribute to feedback in insect flight control?
- How does the nervous system of insects use timing and time history to resolve small differences in deformation from gyroscopic forces?

To answer these questions I have collaborated with experimentalists to derive models of neural encoding of the campaniform sensilla on the wings of moths and halteres of flies. Through structural simulation of the oscillating wings and halteres combined with inertial rotation I obtained local strain on the entire surface throughout a wingbeat cycle. Combining strain with the experimentally derived encoding models provides a neural representation of local deformation. This neural representation in turn carries the signature of the inertial rotations experienced by the insect. It is this model chain through which we can gain understanding of the distal computing in the oscillating sensory organs during flight.

With this insight we can now probe deeper, not only learning how the biological system works, but also how we can employ these mechanisms for engineering design. The approach of combining structure with neural-inspired sensing allows us to ask the reverse engineering questions:

- Why are mechanosensors placed at these locations?
- What is it about the feature-sensitivity of these mechanosensors that aides in the detection of rotation?

In my attempt to answer these questions, the accurate timing and temporal feature detection showed to be crucially important. This led to the question that inspired the final chapter:

- Can we utilize timing parameters as an efficient method for learning in a new type of controllers?

To test this, I designed a timing-based controller on a variation to the benchmark inverted pendulum. This use case serves as an exploration of the potential of such timing-based controllers in more complicated control systems. My efforts to answer these questions have led to the three chapters in this thesis:

In chapter one I investigate the role of neural encoding by mechanosensors on the wings of hawkmoths. To stabilize the inherently open-loop unstable flapping flight, hawkmoths rely on abdominal reflexes elicited by rotationally induced wing-deformations [21] However, detecting the minute additional wing deformation that results from body rotation is exceedingly difficult, demonstrated by the fact that there are over 200 mechanoreceptors on the wing. These mechanoreceptors called campaniform sensilla encode local strain by sending action potentials through their axon in response to particular temporal strain signals. By combining the strain obtained from a flapping wing simulation with experimentally derived neural encoding we predicted the probability of firing for hypothetical sensors for all locations on the wing surface. We then used this to determine the ability to discern body rotation using different combinations of sensors and different neural encoder models. Interestingly, the temporal feature detection of the neural encoding was crucial in enabling the classification of body rotation. Furthermore, using sparse optimization, roughly 10 sensors were sufficient to classify with similar accuracy to the entire array of over 1000 sensors. Additional simulations revealed that many different encoder shapes led to similar classification accuracy, showing that the shape is not critical.

Chapter two describes how the spike timing by campaniform sensilla at the base of halteres encodes body rotation. Insects belonging to the highly successful order of True Flies (or Diptera) have only two wings. The hind wings of their evolutionary ancestor evolved into small hammer-like sensory structures that still beat at the same frequency as the fore wings of the fly. This oscillatory motion elicits inertial deformations and makes them susceptible to small Coriolis forces during inertial body rotations as experienced during voluntary turns and involuntary motions from wing gusts. Sensory information from the campaniform sensilla at the base of halteres is critically important for flight control. Behavioral experiments show that wing motion is directly affected by body rotations [22], body rotation affects neck muscle activation to stabilize head motion [23], and campaniform sensilla at the haltere base directly innervate wing steering muscles [7]. Yet how the array of campaniform sensilla is used to detect the minor change strains resulting from body rotations is still unknown. Finite element simulations of a simplified haltere structures under different rotational conditions revealed centrifugally induced bending and twisting deformations previously not considered. With the probabilistic firing model similar to the moth wing, I found that spike timing changed at nearly every location along the circumference of the haltere base, enabling timing-based classification of body rotation.

Findings from the previous chapters inspired the timing-based feedforward control strategy employed on the inverted pendulum in chapter 3. Many modern day engineering challenges are similar to those faced by flying insects. Environments are highly unsteady, system dynamics are of the same order as delays from sensory transduction and computation, and both control effectiveness and state estimate vary with time. As a model for such a system I used the benchmark inverted pendulum problem with the additional restrictions that state can only be sensed around it's inverted equilibrium and with limited control torque. By optimizing start and stop time for the feed-forward controller during the sensor-denied phase, the controller performed near-optimally. Furthermore, the iteratively learned timing-based feedforward controller found a tiered, near-optimal solution in far fewer trails.

Chapters 2 and 3 have appeared in journal publications [24], [25]. Because the co-authors heavily contributed to the writing of the final paper, chapter 1 is an earlier draft of the journal publication [26].

# Chapter 1

## Neural-inspired Sensors Enable Sparse, Efficient Classification of Spatiotemporal Data

Mohren T.L. , Daniel, T.L., Brunton, S.L. and Brunton, B.W.

Proceedings of the National Academy of Sciences, 2018.

## Abstract

Recent developments in sparse sensing techniques have enabled many advances in controlling complex fluids, facial recognition and object classification. However, current sparse sensing implementations generally focus on spatial data only. In this work we show the necessity of neural-inspired encoding for classification by utilizing the the spatio-temporal nature of sensory data in combination with sparse optimization. We test our method on a structural model of a flapping wing with an array of strain sensors, with the aim of classifying its rotational state. The experimentally derived neural encoder is applied to the strain field prior to finding the optimal sensor locations for classification. We found that applying a neurally inspired encoder was necessary to classify the inertial rotation state. Furthermore, we found that we need roughly half the amount of optimally placed sensors compared to randomly placed sensors for the same classification accuracy. As we added white noise disturbances to both the flapping motion and the inertial rotation, the classification accuracy was reduced only when disturbance levels exceeded 10% or more of the steady angular rate. To show the significance of the shape of the neural encoder, we approximated experimentally obtained encoders with parameterized functions. We found that the parameters derived from experimental encoders performed close to the best performing parameter set. With this model we demonstrate that combining neural encoding and sparse sensing is beneficial to high dimensional classification tasks.

## Introduction

Insects rely on structures with mechanosensory arrays for flight control [22], [27]–[29]. These structures deform in response to aerodynamic or inertial loads. Mechanosensation in insects has a latency of around 3 milliseconds, which is extremely fast compared to other sensory systems [5]. Vision, on the other, is too slow to account for the rapid feedback observed in natural flight [6], [30]. Located on the wings of hawkmoths are over 300 campaniform sensilla in a distinct pattern [21], [31], a similar number of sensors as are found on the wings of flies [32]. Furthermore, the

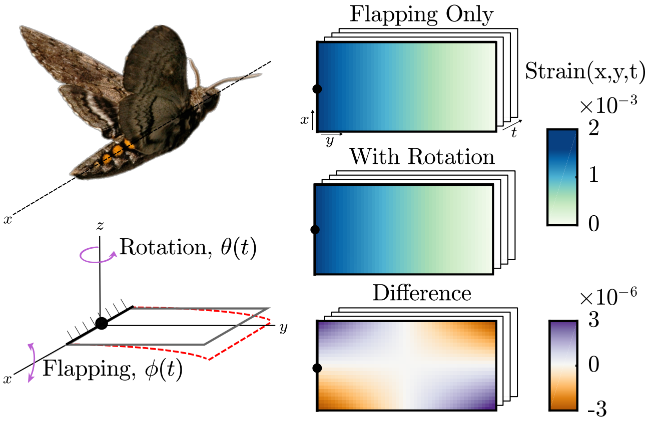

Figure 1.1: A simulated flapping wing model with and without externally induced body rotation differ by a twisting mode *three orders-of-magnitude* smaller in magnitude than the dominant flapping mode. From the flapping wing simulation, we obtain span-wise normal strain over a dense grid on the wing as a function of space $(x, y)$ and time $t$. Photo of hawkmoth by A. Hinterwirth.

encoding properties of these campaniforms on hawkmoth wings is strikingly similar to those of campaniform sensilla found on the halteres of flies. These halteres are known to serve an important role in gaze stabilization [1] and detecting inertial rotations [33], both functions that are important in flying in turbulant environments. The halteres are evolutionary derivatives of hind wings, suggesting that the hind wings of insect species without halteres have a similar function. There are obviously aerodynamic forces acting on flapping wings, but at the scale of hawkmoths these are minor compared to inertial forces generated by the wing itself [34]. Indeed, recent work shows that inertial rotations lead to a very small additional wing twist [35], and that wing twist results in a stabilizing reflex from the abdomen [21]. However, it remains an open question as to how the array of mechanosensors on the moth wing can detect this wing twist mode.

In the nervous system, information can be encoded in a multitude of ways, common examples being by the frequency of occurence of action potentials (firing rate), or by the time difference of action potentials between neurons [36]. Electrophysiological recordings of afferent wing neurons show that the campaniform sensilla on hawkmoth wings are sensitive to very particular features in the strain time-history [10]. Furthermore, spike timing precision is within 0.2 milliseconds [21].

The field of sparse sensing and much of machine learning uses the principle that natural high dimensional data often has coherence, and thus can be transformed into a lower dimensional representation, allowing for the truncation of smaller modes [37]–[40]. The low dimensionality of this representation suggests that not all measurements are necessary, and that if sub sampled, the full signal can be reconstructed with high fidelity [41], [42]. If the goal is not to reconstruct the full image, but instead to classify the image into a category, even fewer measurements are necessary. This has been shown in image recognition, where randomly sub sampled images can be classified with roughly equal accuracy with respect to the full image [43]. Furthermore, if sensor locations are selected for their ability to classify, the number of sensors required becomes much smaller than if they are placed randomly [44]. For high-dimensional systems such as images, solving for the optimal sensor locations is computationally infeasible, but with convex optimization the optimal set can be approximated with the $L_1$ norm [45]. Rather than the least squares $L_2$ norm, the $L_1$ norm promotes sparsity in the solution, but this norm requires more computational efford. Furthermore, in case of many near-optimal solutions, finding the absolute optimal can be intractable. A way to deal with this uniqueness problem is to introduce a combination between the $L_1$ and $L_2$ norms, referred to as an elastic net [46]. We will use this combination of the $L_1$ and $L_2$ norm to determine optimal sensor locations.

First we will show that neural encoding is necessary for classification, and that optimal sensor placement outperforms random sensor placement if under 30 sensors are placed. Second, we show that added disturbances to wing stroke and inertial rotations reduce the classification accuracy only when disturbances are of the same order as the steady perturbation. Lastly, we explore effect of the neural encoder shape to classification accuracy.

# Methods

We use an Euler-Lagrange model of a flapping wing based on structural and flight parameters of the hawkmoth *Manduca sexta*. Because we hypothesise that neural encoding is especially useful in noise rejection, disturbance is added to both the flapping angular velocity and the inertial rotation velocity. We will then derive a neural encoder that approximates experimental recordings [10]. This neural encoder is then used to filter the strain from the Euler-Lagrange model determine the probability of firing over the entire wing over time. With this spatio-temporal data we will select sensor sets of different sizes either randomly or through an optimal placement algorithm. We then train different sensor sets on 90 % of the neurally encoded strain and test classification accuracy on remaining 10%. The fraction of correct classifications in the test data determines the classification accuracy for that particular sensor set.

## Structural Simulation

We use a simplified Euler-Lagrange model for a flapping flat plate to obtain wing strain for different prescribed inertial rotations (simulation developed by Eberle et al. [35]). The flat plate has a span of 50 mm, chord length of 25 mm, a thickness of 0.0127 mm, and an E-modulus of 3 GPa to roughly match the structural characteristics of a *Manduca Sexta* wing. This is consistent with previous work on hawkmoth structural wing studies [35], [48]. The wing flaps with an amplitude of 30 degrees at a frequency of 25 Hz and, to resemble the hawkmoth wing stroke more closely, we add an additional harmonic at 50 Hz of 6 degrees amplitude. The wing stroke angle is thus:

$$\phi = 30\left(\sin 2\pi f t + 0.2 \sin 2\pi 2 f t\right).$$

A sigmoidal startup ensures transient effects are dampened out before the training and testing phase. The simulation runs for 4 seconds, where the first second is disregarded in classification to ensure transient effects are excluded. The total flapping velocity is defined as $\dot{\phi}$ and consists of a mean rotation $\bar{\dot{\phi}}$ and of disturbance $\dot{\phi}^*$. The total inertial rotation is defined as $\dot{\theta}$ and consists of a mean rotation $\bar{\dot{\theta}}$ and of disturbance $\dot{\theta}^*$. The wing is perturbed under two different inertial rotations, signifying the two different classes.

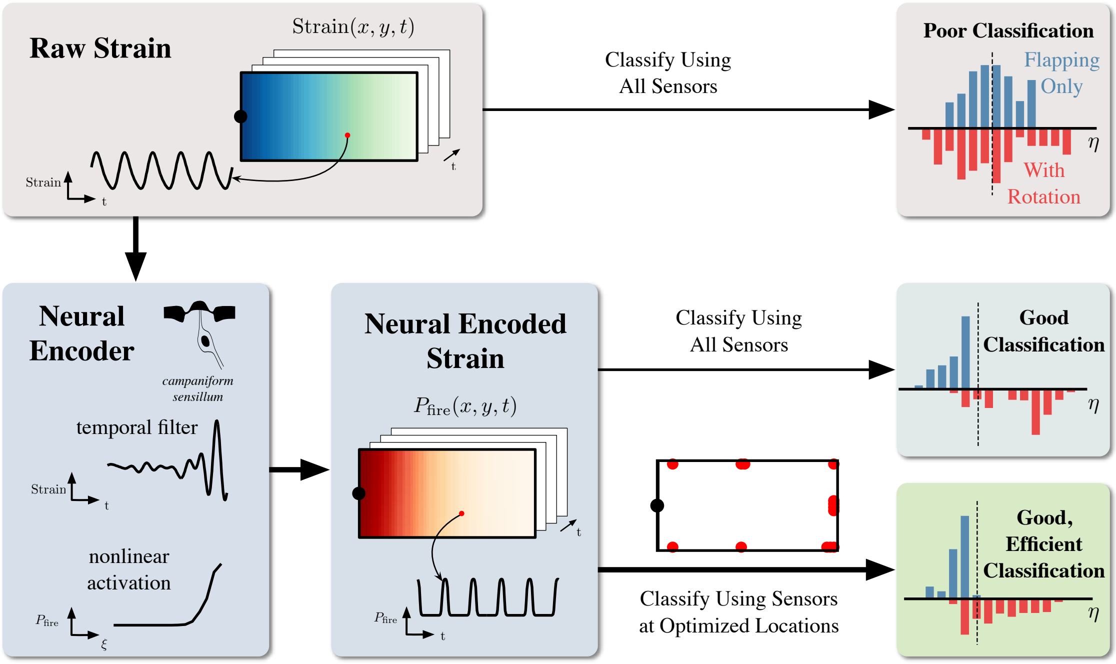

Figure 1.2: A schematic of classifying body rotation using sparse neural inspired strain sensors placed on a flapping wing. Raw span-wise normal strain is obtained from the structural simulation in two conditions, flapping only and flapping with rotation [35]. The raw strain in these two conditions are not linearly separable, leading to poor classification even using all the sensor locations. Alternatively, raw strain is encoded by a neural-inspired filter and transformed into the probability of a mechanosensory neuron firing an action potential [10], [47]. The neural encoder is approximated by experimental recordings of campaniform sensilla and summarized as a temporal spike triggered average (STA) filter followed by a nonlinear activation function, transforming raw strain into probability of the neuron firing an action potential $P_{\text{fire}}$. We define $P_{\text{fire}}$ to be neural encoded strain. Neural encoded strain separates well with a linear classifier; further, this performance can be achieved remarkably efficiently using approximately 10 sensors at key locations [44].

- In the case of flapping, there is no constant rotation rate ($\bar{\dot{\theta}} = 0$ rad/s).
- In the case of flapping and rotating, there is a constant rotation rate ($\bar{\dot{\theta}} = 10$ rad/s).

For figures 1.1,1.2,1.3 and 1.5 we add band limited white noise disturbance $\dot{\theta}^*$ with standard deviation of 0.1 rad/s and $\dot{\phi}^*$ with standard deviation of 0.31 rad/s (1% of $\bar{\dot{\theta}}$ ( (1% of $\bar{\dot{\theta}}$ and $\bar{\dot{\phi}}$ respectively). For figure 1.4 we explore standard deviations of [0.01,0.1,1, 10] rad/s for $\bar{\dot{\theta}}$ , and [0.031,0.31,3.1,31] for $\dot{\phi}^*$. Without any disturbance, the wing shows a dominant strain mode due to bending, independent of inertial rotation speed. When rotation is present ($\bar{\dot{\theta}} = 10$), a twisting mode is observed [35]. This twisting mode results in normal strain of three orders smaller than the dominant strain mode due to bending (figure 1.1) .

## Neural encoding

We use the experimentally derived responses of campaniform sensilla to create a probabilistic firing model, as outlined in [49]. This probabilistic model is used to obtain the probability of firing as a function of time and position on the wing (figure 1.2). The probabilistic firing model is a two step process. First, the stimulus is convolved with the temporal feature (spike triggered average, or

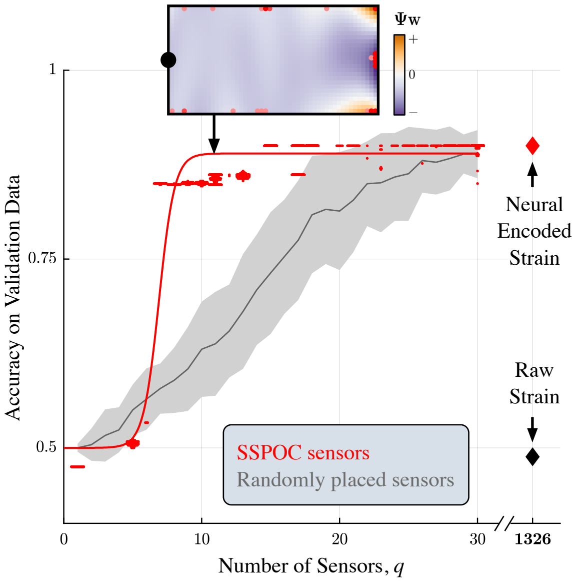

Figure 1.3: Classification using about 10 neural encoded sensors placed at key locations on the wing achieves accuracy comparable with classification using all sensors. Flapping wing structural simulations were computed with moderate disturbance amplitudes ($[\dot{\phi}^*, \dot{\theta}^*] = [0.31, 0.1]$ rad/s). The classification accuracies shown are validated on an epoch of the simulation separate from what had been used for training. Sparse sensors are learned from training data from trials with random disturbances using SSPOC (red, each dot is an individual trial) and compared to randomly placed sensors (gray, mean and stdev). The red line is a sigmoidal fit to the SSPOC sensors accuracy. The inset shows a probability distribution of SSPOC sensor locations on the wing for $q = 11$ sensors, averaged over 100 training sets with random instances of noisy disturbances. The opacity of the red dots are proportional to the likeliness of sensor solutions at that location; most sensors are found at the periphery of the wing. The background of the inset shows the full-state discriminant vector $\mathbf{\Psi w}$.

STA) to determine similarity $\xi$. Second, the non-linear decision function (NLD) takes $\xi$ and maps this to the probability of firing according to it's curve. The result is the Neural Encoded Strain in figure 1.2. We will use a equation A1.19 to represent the STA, and a sigmoidal function A1.20 to represent the NLD. Both are fitted to experimental recordings of campaniform sensilla on the wing of a hawkmoth [10]. This allows us to vary the shape of the neural filters, and explore the effect of filter shape on classifcation accuracy.

$$STA(t) = \cos\left(f(t+\tau)\right) exp(\frac{-(t+\tau)^2}{\sigma^2}) \tag{1.1}$$

$$NLD(s) = \frac{1}{1 + exp(-\zeta(s-\eta))} \tag{1.2}$$

Where $\sigma$ is the width of the STA, $t$ is time in milliseconds, $\tau$ is the delay of the STA, and f is the frequency in kHz. $\zeta$ is the gradient of the slope at NLD(s) equals 0.5, $s$ is the projection of the STA on the strain, and $\eta$ is the delay.

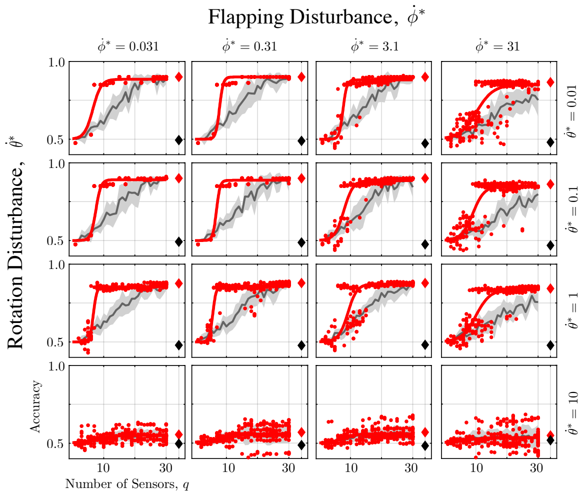

Figure 1.4: Classification accuracy is robust for moderate to large magnitude disturbances in flapping $\phi$ and in rotation $\theta$. Each panel of the 4 by 4 grid shows the classification accuracy for varying number of sensors (grey: random sensors, red: SSPOC sensors, black diamond: all sensors without encoding, red diamond: all sensors with encoding). The levels of rotation disturbance represent 0.1,1,10, and 100% of the standard deviation of steady flapping $\bar{\dot{\phi}}$ and of the magnitude of constant rotation $\bar{\dot{\theta}}$.

## Optimal sensor placement

To show the effect of optimal placement, we will compare different amounts of sensors, and different placement conditions. The sensor sets will have 1 to 30 sensors, and will be either randomly placed or optimally placed using sparse optimization. We use the SSPOC algorithm with an elastic net (equation A1.38) and solve this with convex optimization software CVX [44], [46], [50], [51].

$$\boldsymbol{s} = \underset{\boldsymbol{s'}}{\text{argmin}}\ 0.9 \cdot ||\boldsymbol{s'}||_1 + 0.1 \cdot ||\boldsymbol{s'}||_2, \text{ subject to } \boldsymbol{\Psi_r^T s'} = \boldsymbol{w} \tag{1.3}$$

We use Linear Discriminant Analysis (LDA) to classify rotational state. For each of the two classes (1. Flapping, and 2. Flapping & Rotating), a the last 3 seconds of a 4 second structural simulation was used (grey box in figure Appendix 1 .1), with a sampling rate of 1 kHz. With a flapping frequency of 25 Hz, this equals about 75 wing stroke cycles for each class, 150 strokes as the total data set. The first 90% of the samples for each class was used for training the weights of the individual sensors, the last 10% was used in the testing set to determine the classification accuracy. For every sensor set size, we ran 10 iterations for both random placement and the optimal sensor placement algorithm.

# Results

In the first section, we will show the benefit of applying neural encoding to strain data for the classification task and how a subset of sensors performs if placed randomly versus optimally. Secondly, we study the effect of disturbance on classification accuracy. Lastly, we demonstrate how variations in the neural encoder parameters alter the ability to classify inertial rotations.

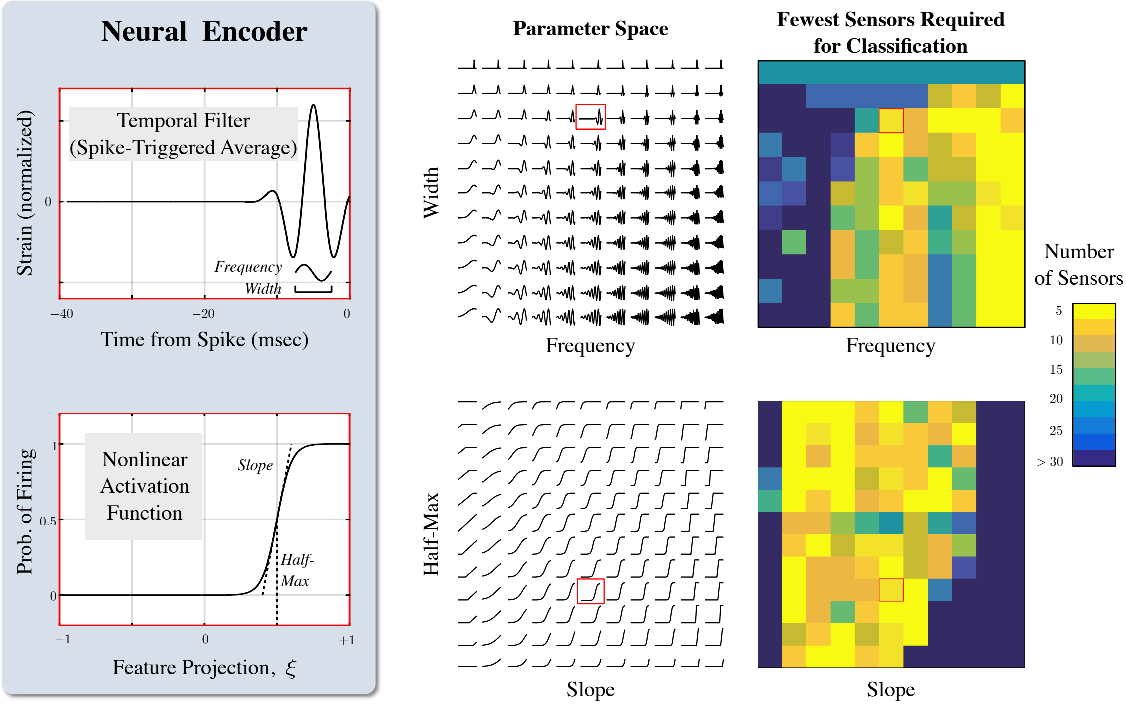

Figure 1.5: The experimentally derived neural encoders are found on a large plateau in parameter space with similar neural-inspired encoders. The temporal filter STA is parameterized by its frequency and width (Eq. 19), and the nonlinear activation function is a sigmoid parameterized by its slope and half-max (Eq. 20). The top and bottom rows show systematic variations of the STA and the nonlinear activation function by manipulating their parameters, respectively. The middle column visualizes the family of these neural-inspired encoders. We assess each encoder by the fewest sensors required to achieve 75% classification accuracy, and these numbers of sensors are shown as heat maps in the right column. The experimentally fit encoder functions (red boxes) are well suited to achieve classification along with a family of similar neural-inspired encoders.

## Neural inspired filter is required for detection of insect inertial rotation

The twisting mode present only when inertial rotations are present is exceedingly small, as can be seen in figure 1.1. This makes that when we try to classify inertial rotations using the strain directly, classification accuracy hovers around 0.5 (figure 1.3. This is the case with sparsely placed sensors, and if all 1326 sensors on the wing are used (in the case of 2 classes, this is equivalent to random guessing). With neural encoding, classification accuracy reaches roughly 90 % if all sensors are used. However, even with a randomly placed subset of sensors only thirty sensorsgive a similar accuracy. At thirty sensors the mean accuracy curve begins to asymptote towards the accuracy obtained from all sensors. The results for a given number of randomly placed sensors has a Gaussian distribution, the region within standard deviaton of the mean is shown in grey in figure 1.3. Optimal sensors perform superior to randomly placed sensors in the region between 5 and 30 sensors, after which they asymptote at the same level as the randomly placed sensors. The sigmoidal fit to the mean lines of both random and optimally placed sensor accuracy intersect with the 75% accuracy (as shown by the dotted lines in figure 1.3). We will use this as an indication of success for classification accuracy for further results.

### Classification robust to disturbance

For the two different rotational velocities we applied different combinations of disturbance (figure 1.4). We found that for disturbance levels up to 10% of the rotational velocities (1 rad/s for rotational velocity and 3.12 for flapping velocity), classification accuracy was not noticeably affected. For a flapping velocity disturbance of 31.2, equalling the standard deviation of the steady flapping velocity, there was a decline in both the accuracy of randomly placed sensors and optimally placed sensors. The asymptotic value dropped to about 80% and more sensors were required sensors to obtain for both random and optimal sensors. The rotational velocity disturbance drastically affected classification accuracy, albeit only when disturbance magnitude equaled steady rotation rate magnitude. Even if all sensors were used, only barely was classification accuracy better than 50 %.

### Classification accuracy insensitive to encoder shape

The neural encoding properties of campaniform sensilla look to be similar across species, although there seem to be different types of encoding properties within species [10], [52]. In figure 1.5 we explore variations of the experimentally obtained temporal feature. Parameterizing the feature with equation A1.19, we alter the frequency and width. It becomes aparent that higher frequency features perform better. Furthermore, small feature widths (about equal to those observed in electro-physiological experiments) perform better than wide, larger time-scale features. Note that the top row has such a narrow feature that it does not affect the strain, i.e. here the encoding is only achieved by the non-linear decision function. This hints at the fact that the projection of the temporal feature improves classification accuracy, but it is not *required* for classification. We can interpret the feature as acting as a temporal filter for disturbance.

Similarly to the temporal filter, we now explore variations in the non-linear function. The non-linear decision functions observed in experiments generally have a sigmoidal shape, although variations are observed between experiments [8], [10]. From figure 1.5 we see that within bounds the location of the slope mid-point does not have a significant impact on classification accuracy. The sharpness of the curve however, shoes a clear trend for randomly placed sensors, favoring a gentle decision function over a sharper threshold function. For optimally placed sensors, the trend is not as obvious. There are clear ounds where decision functions perform poorly (such as the linear ramp functions on the left and the sharp thresholds on the right), but there seem to be several parameter values that perform well. Note that the middle of the left column has a straingt line function, meaning that the strain would not be altered if it were to pass through it. From this we can conclude that the non-linear function is *required* for classification. In the case without any disturbance, the temporal feature might be left out, whereas the non-linear decision function is still required to classify inertial rotations.

## Conclusion & Discussion

Our model of the flapping flat plate shows that neural encoding not only improves, but is actually required for classifying rotational state in our model. From our sparse sensor placement optimization we find that sensors are distributed consistently and that in the order of 10 optimally placed sensors perform nearly as well as the full array of sensors in classifying inertial rotation. Random sensors paired with neural encoding also approached classification accuracy equal to the full array of sensors, but it took more randomly placed sensors to achieve a similar classification accuracy. In both random and optimally placed sensors, classification accuracy was insensitive to low levels of rotational disturbance. The fact that a small subset of samples performs nearly as well as the full set therefore still holds in case of high levels of disturbance.

When looking at the importance of shape of the neural encoder we found that indeed the shape of both the temporal feature and the non-linear decision functions significantly affected the classification accuracy. Contrary to expectation, narrow, high-frequency features generally performed best. If the temporal feature was turned off, more sensors were required to classify but classification was still possible. We therefore interpret the temporal feature to act as a noise filter that is not necessarily required, but that reduces the number of sensors required to classify rotation. The experimentally approximated temporal feature performed nearly as well as the best performing feature we found. Contrary to expectation, we found that a gradual slope for the non-linear decision feature led to the highest accuracy. The location of the slope mid-point did not have as much of an impact. If the non-linear function was left out, the classifyer was no longer able to detect rotation, indicating that contrary to the temporal feature, the non-linear function is crucial in detecting rotation.

For classification, we used Linear Discriminant Analysis (LDA) in combination with Principal Component Analysis (PCA), a basic but powerful classification method with significant off-line but very rudimentary on-line computation. Although this particular method was not able to classify rotations with strain alone, other methods might be able to. However, we hypothesise that applying neural encoding to classification data brings benefits to more involved classification schemes as well.

With the model in this paper we show that incorporating spatio-temporal data using neural encoding can enable classification tasks where spatial data alone cannot. We invision neurally-inspired filters can be used in high dimensional sensory systems that are energetically or computationally constrained. Biological sensors have circumvented these issues with sensory systems by having event based action potentials. To apply a similar feat in an engineered system, one would need a hardware implementation of a sensor with neural filter. Such a sensor would send a binary pulse only in case of a notable temporal signal, decreasing downstream computation and power requirements. In this study we used a one-feature based neural encoder for our model, but recent evidence suggests that campaniform sensilla on halteres respond to two different features in the stimulus [8]. Such an implementation would result in a more selective encoder, further enhancing the efficiency of the sensory system. Much work remains to be done in understanding the biological mechanism of sensing complex features by sensory arrays, and perhaps even more so in the application of biologically inspired encoding mechanisms for engineered sensory systems.

# Chapter 2

## Coriolis and Centrifugal Forces Drive Haltere Deformations and Influence Spike Timing

Mohren T.L. , Daniel, T.L., Eberle, A.L., Reinhall, P.G. and Fox, J.L.

Journal of the Royal Society Interface, 2019.

## Abstract

The halteres of flies are mechanosensory organs that serve a crucial role in the control of agile flight, providing sensory input for rapid course corrections to perturbations. Derived from hind wings, halteres are actively flapped and are thus subject to a variety of inertial forces as animals undergo complex flight trajectories. Previous analyses of halteres modeled them as a point-mass, showing that Coriolis forces lead to subtle deflections orthogonal to the plane of flapping. By design these models could not consider the effects of force gradients associated with a mass distribution. Nor could they reveal three dimensional spatio-temporal patterns of strain that result from those forces. In addition, diversity in the geometry of halteres, such as shape and asymmetries, could not be simply modeled with a point mass on a massless rod. To study the effects of mass distributions and asymmetries, we examine the haltere subject to both flapping and body rotations using 3D finite element simulations. We focus on a set of simplified geometries, in which we vary the stalk and bulb shape. We find that haltere mass distribution gives rise to two unreported deformation modes: (1) halteres twist with a magnitude that strongly depends on stalk and bulb geometry, and (2) halteres with an asymmetric mass distribution experience out-of-plane bending due to centrifugal forces, independent of body rotation. Since local strains at the base of the haltere drive deformations of mechanosensory neurons, we combined measured neural encoding mechanisms with our structural analyses to predict the spatial and temporal patterns of neural activity. This activity depends on both the flapping and rotation dynamics, and we show how the timing of neural activity is a viable mechanism for rotation-rate encoding. Our results provide new insights in haltere dynamics and show the viability for timing based encoding of fly body rotations by halteres.

## Introduction

Animals control movement via the integration of inputs from multiple sensory modalities. In insect flight control sensory inputs are largely dominated by both visual and mechanosensory systems.

In the extremely rapid dynamics associated with insect flight, feedback control via visual input is often too slow to provide adjustments to the flight path in response to perturbations [2], [5], [6], leading to pitch instabilities. To compensate for relatively slower visual input, rapid feedback from mechanosensory structures often serves a crucial role in flight control [29].

In dipteran insects (the true flies), hindwings have evolved into specialized mechanosensory structures called halteres. These organs provide exceedingly rapid feedback information about the animal's body dynamics [22]. As halteres flap and the body rotates, Coriolis forces drive subtle deformations of the halteres that are perpendicular to the plane of flapping. Coriolis forces are directly proportional to both the flapping frequency and the body rotation rate [53]–[55]. The dynamics associated with Coriolis-induced motions have been widely studied over the greater part of the last century, with a consensus view that gyroscopic sensing capabilities follow from the detection of small out-of-plane bending [53]–[55] (for the mathematical basis see supplement). This out-of-plane deflection is transformed into neural signals via mechanical sensory structures, called campaniform sensilla, located in dense fields at the haltere's base. These sensilla provide rapid flight feedback via fast electrotonic synapses onto the flight motor neurons [7]. Sensory information from the haltere nerve is also important in gaze stabilization [23], [56]. While there is significant behavioral and electrophysiological evidence for sensory roles of halteres [8], [49], [53], the tiny, rotation-induced bending deformations have not been observed experimentally. In addition, it has been historically thought that detection of such small lateral deflections would require strong directional sensitivity of the campaniform sensilla. While some animals, like spiders, are known to employ mechanosensors, called slit sensilla, which possess high directional sensitivity [57]–[59], whether the directionality of strain is encoded by campaniform sensilla on halteres remains unclear.

Prior analytic studies of halteres have modeled them as a single point mass at the end of a massless rod, allowing only deformations that result from both Coriolis forces and in-plane inertial bending [53]–[55]. In contrast, more recent models of insect wings suggest that Coriolis forces on distributed masses lead to torsional deformations which had not been previously considered [35]. Indeed, stabilizing reflexes have been observed in moths in response to magnetically induced wing twist [21]. It is thus possible that the distribution of mass in halteres could similarly yield torsional dynamics, even if it is symmetrically distributed with respect to the stalk. Moreover, asymmetries abound, with impressive geometric variation of halteres across a wide range of taxa [60] (see example in figure 2.1A). In addition, asymmetric mass distributions of the bulb could potentially lead to relatively large out-of-plane deformations due to centrifugal forces, an additional component of haltere dynamics that had not been examined previously. Thus two modes of deformation—torsion and centrifugally generated bending—remain unexplored. Here we ask four key questions: what are the three-dimensional deformations, stresses, and strains in halteres when undergoing concurrent flapping and rotation? What are the dominant forces leading to those deformations? How do the structural features of halteres influence the relative magnitudes of emergent deformations? And how do such deformations ultimately lead to patterns of neural activation? Answers to these questions will guide a deeper understanding of haltere design and function, as well as the roles of sensory information and spike timing [61] in insect flight control. Moreover, the mechanisms by which complex body dynamics are transformed into neural signals via the deformations of halteres

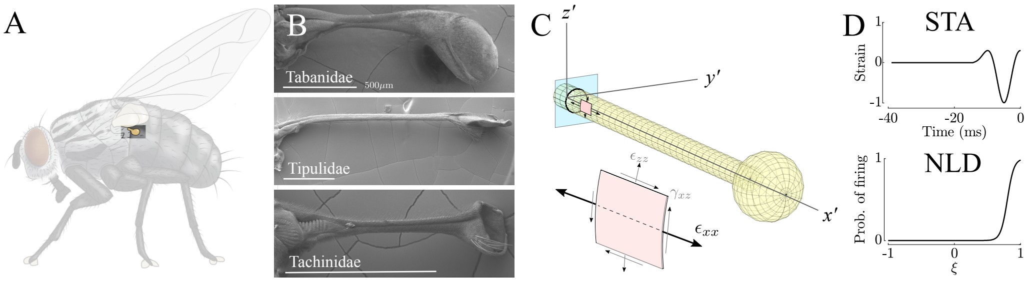

Figure 2.1: Insects belonging to the order of the the true flies (Diptera), possess halteres—dumbbell-shaped mechanosensory organs associated with inertial sensing (A). Three SEM images show different haltere geometries associated with specimens from three different genera (B, courtesy of Sweta Agrawal). The geometry of our basic finite element mode (C) with normal and shear strain components near the base of the haltere. Panel D shows the normalized spike triggered average (STA) and nonlinear decision function (NLD) derived from the encoding properties of cranefly haltere neurons. The parameter $\xi$ is the normalized projection of the STA onto the stimulus from a white noise analysis. Both the STA and NLD are derived from Fox et al. [8].

can inspire novel design for synthetic gyroscopic sensors.

In contrast to prior studies of halteres that used point mass simulations, we address these questions using 3D finite element models to examine spatial and temporal patterns of the deformations and strains associated with haltere dynamics. We combine strain data from the finite element models with neural encoding properties obtained from electrophysiological recordings to develop predictive models of spatial and temporal patterns of neural signals from campaniform sensilla at the haltere base. We find that both centrifugal forces and Coriolis-induced torsion contribute to the strain patterns at the base of the haltere, and that these forces can influence the timing of neural activation. Moreover, the geometry of halteres influence the relative contributions of these forces in the mechanism of motion encoding. Taken together, our results suggest a new view of the mechanics and neurobiology of rotation sensing in flying insects.

# Materials and Methods

Our model of the haltere geometry and kinematics in the structural simulations was inspired by the cranefly haltere from the genus *Tipulidae*. We assumed homogeneous material properties and a cylindrical stalk geometry to establish a mechanistic understanding of the deformation modes.

### Geometry and Kinematics

Our base model consisted of a hollow stalk of circular cross-section with an outer radius of 150 $\mu$m and an inner radius of 50 $\mu$m. The end of the stalk supported a bulb with the center of mass at a distance of $5 \cdot 10^3$ $\mu$m from the base of the haltere. The bulb had a radius of 500 $\mu$m.

We assume uniform material properties, and use a Young's modulus of $E = 1.5$ GPa, as obtained from nanoindentation on halteres [62]. We set both the stalk and bulb density to $\rho = 1200$ kg/m$^3$ to be consistent with previous modeling [62] and with prior estimates of cuticle stiffness [34]. Furthermore, we assume a Poisson ratio of $\nu = 0.33$. We note that the Young's modulus of scerotised cuticle can range from 1-20 GPa, and it's density falls between 1 -1.3 kgm$^{-3}$ [63]. We modeled the bulb as a rigid element, as the moment due to gyroscopic accelerations is smallest at the tip and results in a small contribution to the total haltere deformation.

To investigate the dependence of haltere deformation on haltere shape, we modeled two different stalk cross-sections and three bulb geometries. We compared the spherical bulb to elliptical bulbs with a major axis length of 1000 $\mu$m and two minor axes lengths of 353 $\mu$m , maintaining the same internal volume and center-of-mass distance from the base of the haltere (fig. 2.3). To investigate the effect of asymmetry, we also offset the bulb center of mass in either the $y'$ or $z'$ direction by 150 $\mu$m. To examine the role of torsional and bending stiffness of the stalk, we modeled both the circular-shaped cross-section (CS) as described previously, as well as a plus-sign shaped cross-section (PS) consisting of two orthogonal rectangles with a height of 481 $\mu$m and a thickness of 40.8 $\mu$m, corresponding to an aspect ratio of 12. These two cross-sectional shapes were scaled to have an identical second moment of area $I$, but a vastly different torsional moment of inertia $J$ (by an approximate factor of 38, fig. 2.2, derivation in the supplement).

We fixed the base of the haltere model to a rigid base plate that we rotated in the Y and Z-axis. We prescribed the flapping motion as the haltere stroke angle about the Y-axis:

$$\phi(t) = A_f \sin(2\pi f_\phi t) \text{ [rad]}, \tag{2.1}$$

where $f_\phi = 40$ Hz, and amplitude $A_f = \pi/2$ rad, values that approximate observations of cranefly halteres [49], [64]. Interestingly, the haltere flapping frequency is mechanically coupled to wingbeat frequency [65].

We then specified one of two different rotational velocities to the haltere frame around the Z-axis:

$$\dot{\theta}_{\text{Flapping only}} = 0 \text{ [rad/s]}, \tag{2.2}$$

$$\dot{\theta}_{\text{With Rotation}} = 10 \text{ [rad/s]}. \tag{2.3}$$

We chose a conservative rotation rate of 10 rad/s since yaw rates between 14 and 28 rad/s are commonly observed in flying insects [66]. Both the flapping and rotation were ramped up using a sigmoidal function during the startup phase.

We used a radially symmetric mesh for the stalk using linear hexahedral elements (see also figure 2.1C). We specified both inner and outer circumference of the stalk to have 16 equally spaced nodes. We prescribed 5 nodes at equal radial distances, resulting in 4 by 4 elements per stalk quadrant. We used 31 elements along the length of the stalk. The bulb is comprised of 4 elements per quadrant in both curved directions. We added Rayleigh damping of $\alpha_k = 1 \cdot 10^{-5}$, and $\alpha_m = 0$ to ensure stability of our solution. The selection of damping values do not change results within two orders of magnitude of the value we selected. However, large damping values can counter intuitively lead to instabilities. Such damping induced dynamics may relate to a phenomenon known as damping induced rotational instabilities [67]. Our simulations were implemented in COMSOL 5.0.

### Deformation and Strain

We took the location of the two nodes at the extreme sides of the haltere bulb to calculate the deformation angle from the deformations $\delta$ in the haltere's local x'y'z'-reference frame with in-plane bending, $\Delta\phi = \tan(\delta_z/x_d)$, and out-of-plane bending, $\Delta\theta = \tan(\delta_y/x_d)$. We used the two side points to calculate the twist angle $\Delta\gamma = \tan(\delta_z/y_d)$, with $\delta_z$ the distance between the two points along the z' axis, and $y_d$ the distance between the two points along the y'-axis. In this study we used the normal strain in the x'-direction of the local frame ($\epsilon_{xx}$), as it is the dominant strain signal. We exported this strain from the finite element software, but we note that strain relates to the local curvature according to the Euler-Bernoulli beam theory in thin beams experiencing small deflections [68] :

$$\epsilon_{xx} = -z\frac{d^2w}{dx^2}, \tag{2.4}$$

with $z$ the distance with respect to the neutral axis of the beam, and $w$ the deflection in the z'-axis.

### Neural encoding

To predict spiking from our simulations, we used normal strain in the spanwise direction of the haltere ($\epsilon_{xx}$), at the locations along the outer circumference of the haltere and two mesh nodes outward from the base (300 $\mu$m distally). We focus here on a single example geometry to illustrate the transformation of spatio-temporal patterns of strain to patterns of neural activation. This strain was passed through a two-step spike prediction method, using the neural encoding properties obtained from electrophysiological recordings from cranefly haltere neurons following methods outlined by [8], [10], [26], [49]. These recordings revealed the temporal patterns of mechanical strain stimuli that would most likely lead to an action potential from haltere neurons. The average temporal pattern that elicits a neuron to fire, the spike triggered average (STA), was derived from averaging across the ensemble of stimulus histories preceeding the spikes of that neuron. Given the STA and the distribution of stimuli that generate action potentials, we used Bayes' theorem to identify the probability of action potential generation for a stimulus. This probability of firing followed a sigmoidal curve, often referred to as the Non-Linear Decision Function (NLD). With the STA and NLD, we then determined the probability of action potentials occurring over time for any given temporal strain signal. We used STA and NLD estimates from a white noise motion stimulus provided to the tip of the haltere [8]. We used methods outlined in [8] to scale the displacement stimulus to strain at the base of the haltere. We assume that a spike occurs at the peak of the probability of firing, but only if this peak probability of firing exceeded a threshold value of 0.9. The transformation from strain to firing rate and spiking is shown in figure 2.1D.

The simulation data from the computational models is available at Zenodo, `https://doi.org/10.5281/zenodo.2542944`. We used MATLAB [69], to analyze the data. Our code to analyze the finite element results and to implement the neural encoding can be found at `https://github.com/tlmohren/Haltere-code`.

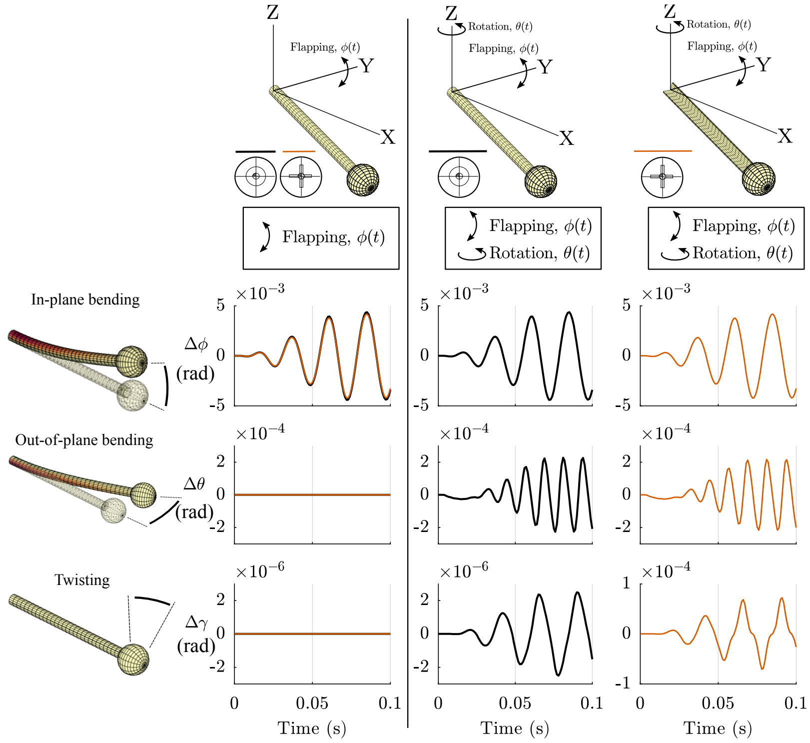

Figure 2.2: Three deformation modes are shown for two different stalk geometries. The left column of graphs represents the deformations that results from pure flapping motions with two stalk geometries: circular cross-section (CS, black) and a plus-sign shaped cross-section (PS, red). The deformations that occur for the CS cross-section (middle column) and the PS cross-section (right column) experiencing both flapping and rotation show the importance of torsional relative to bending stiffness. Note the different axis magnitudes for the CS-shape stalk demonstrating the importance of polar moment of area relative to second moment of area.

# Results

Our results are consistent with those of earlier studies that indicated significant in-plane bending and a subtle out-of-plane bending induced by the Coriolis force acting on a flapping and rotating haltere [53]–[55] (fig. 2.2). For our flapping haltere, not subject to Coriolis forces, the in-plane bending attains a peak-to-peak amplitude of approximately 0.01 rad. That amplitude is about 25 times greater than the out-of-plane amplitude when the haltere is subject to Coriolis forces resulting from the orthogonal body rotations we imposed in the model. Moreover, the frequency of the out-of-plane bending is twice that of the flapping frequency, a result consistent with all prior studies of halteres [53]–[55]. Interestingly, a haltere with a spherical bulb and a cylindrical shaft also experiences torsional deformation (twisting) which is much smaller in amplitude than either the in-plane or out-of-plane bending (middle panel, blue lines). That torsional deformation has a dominant frequency that is at the same frequency as the flapping motions.

**Stalk geometry**. Circular cross-sections (CS) represent structures that have the highest resistance to torsional deformations: their polar moment of area $J$ is maximum for a given amount of material. Haltere stalks are not perfectly circular. Instead, they have a groove running along the stalk that reduces their polar moment of area [60]. To explore the consequences of different beam

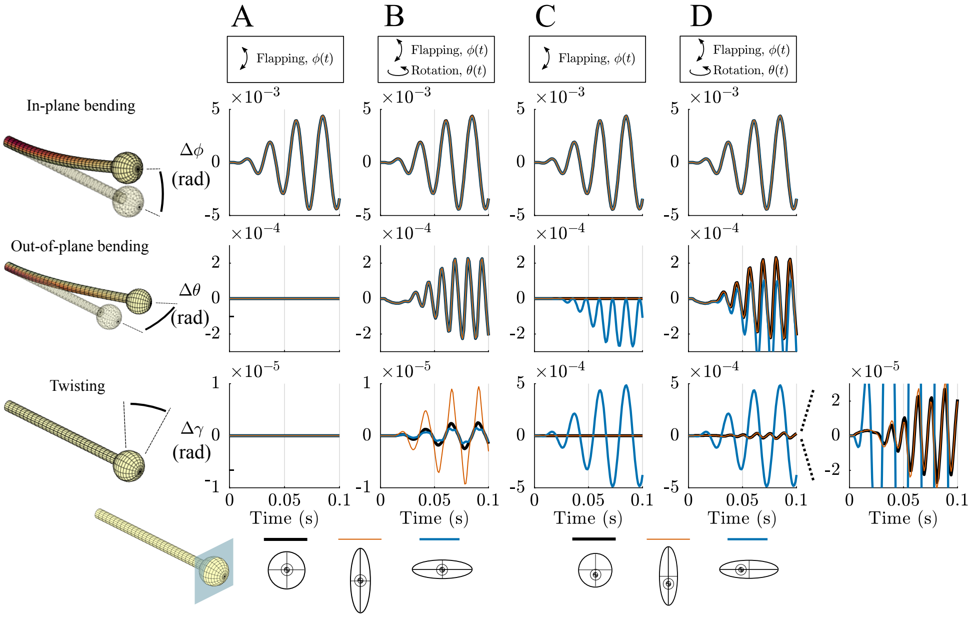

Figure 2.3: Three deformation modes are shown for different bulb geometries, (A and B are symmetrically mounted, C and D are asymmetric). In column A, both symmetrically mounted ellipsoidal and spherical bulbs have identical deformations when experiencing only flapping motions (all three therefore have the same line color). Column B shows the predicted deformations for the three symmetrically mounted bulbs with identical deformations for in-plane and out-of-plane bending experiencing flapping and rotation. Torsional motions are greatest for the vertically oriented bulb. The predicted deformations for asymmetrically mounted bulbs experiencing only flapping (column C) and for both flapping and rotations (column D) show the influence of both Coriolis and centrifugal forces. The far right graph is an expansion of the twisting deformations in column D. Note the different axis magnitudes.

cross-sections, we created the plus-shaped cross-section (PS) that has the same second moment $I$ of area as cylinder, but a significantly lower $J$. This PS cross-section has an identical out-of-plane displacement as we found for a circular cross-section (fig. 2.2, middle of right column). However, the torsional deformation of the PS cross-section is much greater than that for the CS cross-section and of same order of magnitude as the out-of-plane deformation (fig. 2.2, right column, bottom row).

**Haltere asymmetries**. Because haltere deformations depend quite strongly on the distribution of mass, we explored the consequences of two geometric scenarios. In one case, we asked how deformations depend on the distribution of mass for bulbs symmetrically mounted to a shaft with the same CS cross-section. In this scenario, we compared the deformations of halteres with ellipsoidal bulbs to that of a spherical bulb, all with the same total mass. All three bulb shapes lead to identical out-of-plane bending deformations when subject to both flapping and rotation. None demonstrated out-of-plane or torsional deformations for flapping only (fig. 2.3, column A). Interestingly, mass distributions that are symmetric with respect to the shaft and oriented in the plane of the body rotation, and parallel to the plane of flapping (column B, vertical bulb, red lines) experience much greater twisting than that of either a spherical bulb (black line) or an ellipsoidal bulb oriented orthogonally to the plane of flapping (horizontal bulb, blue line)

Even for a haltere with an asymmetric bulb, solely undergoing flapping excitation, an additional deformation mode arises (fig. 2.3, column C). First, for a horizontal ellipsoid bulb, a centrifugal force (see Supplemental information for the equations) leads to out-of-plane bending motions that are of the same magnitude as those seen for halteres experiencing Coriolis forces (blue line, column C, middle panel). In this instance, centrifugal forces yield deformations at twice the frequency of the flapping motion, which is the same frequency that occurs with Coriolis forces. Morever, there is a net offset to the bending angle of the haltere resulting from a lateral torque due to centrifugal forces. The vertical ellipsoidal bulb which is asymmetrically mounted on the shaft (red line) also experiences centrifugal forces, but those are manifest as subtle changes in the in-plane bending (the difference for in-plane bending is only $1 \cdot 10^{-4}$ rad).

When rotation is added, both asymmetrically mounted ellipsoidal bulbs experience large out-of-plane bending at twice the flapping frequency due to Coriolis forces (middle panel, column D). The vertically oriented ellipsoidal bulb (red line) and spherical bulb (black line) have similar out-of-plane bending motions. However, the horizontal ellipsoidal bulb experiences significant twisting motions, driven largely by the high amplitude flapping motions (lower panels, colums C and D), with a modest Coriolis-induced torsion that is also apparent for vertical ellipsoidal bulb.

**Spike timing along circumference** The temporal pattern of normal strain at five locations around the circumference of the shaft was processed with a neural filter to predict the probability of neuronal firing (spiking) as a function of angular position around the base of the haltere (fig. 2.4) subject to both flapping and rotation. The normal strain at the top is dominated by in-plane bending, whereas the normal strain at the side is dominated by the Coriolis force. There is also a modest normal strain present on the side (black line, left panel) and everywhere along the haltere that arises from a centrifugal forces tensioning the haltere shaft because of the flapping motion. The difference in spike times with Coriolis forces present increases toward the lateral positions of the haltere base. At an angle $\alpha = \pi/2$ rad, neurons only spike in the presence of rotations and do so at twice the flapping frequency.

Spike timing is a function of angular position around the haltere base (fig. 2.5). For the flapping haltere, the spikes for the dorsal side occur simultaneously, and they are in anti-phase with the spikes on the ventral side (vertical black lines). With rotation, the left side of the haltere experiences a delay in spike timing that increases with closer proximity to the lateral margin. Interestingly, the right side experiences a spike timing difference of similar magnitude, yet here they precede the spikes from flapping alone. Only when rotation is present does spiking occur at the lateral margins, and it does so at double the flapping frequency.

Our predictions of spike timing differences in the presence of Coriolis forces exceeds the jitter observed in spike timing of haltere campaniform sensilla (recordings had a mean jitter of approximately 0.2 milliseconds, [49]). This jitter represents the temporal error with which neurons respond to repeated sinusoidal stimuli. Thus, we predict that haltere strains processed through a neural filter are observable by the insect neural system.

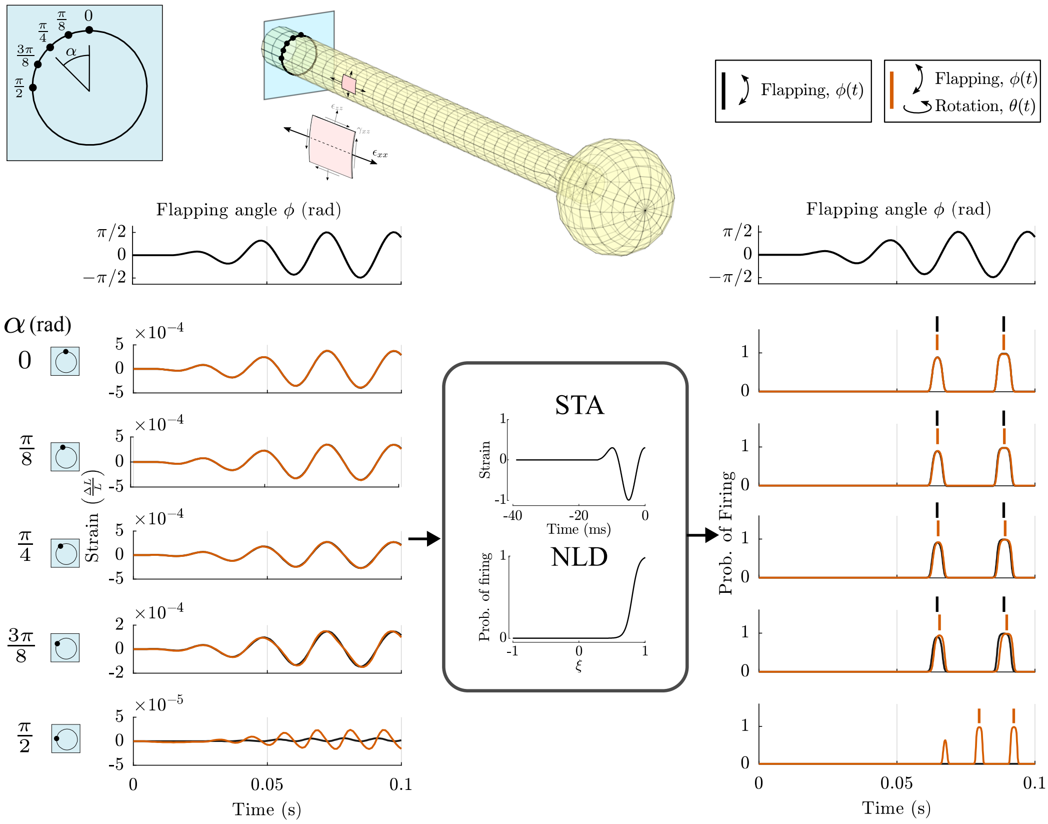

Figure 2.4: The strain along the first quarter of the haltere circumference for flapping and flapping with rotation is converted into probability of firing through the neural encoding properties from haltere campaniforms, as described in [8]. The spike timing on the top of the haltere does not change due to rotation, but for larger $\alpha$ the spikes have an increasing timing difference. Finally, at the sides of the haltere, there no more spikes without rotation. We use use strain in the longitudinal direction (indicated with $\epsilon_{xx}$ on the shell element).

# Discussion and Conclusions

Using finite element models, we have shown how fly halteres respond to complex forces and, in turn, how their form influences their function. In contrast to prior studies which modeled the haltere as a point mass on a massless rod [53], [55], here we show that complex spatial-temporal patterns of strain emerge from rectilinear and rotational accelerations of the haltere mass. Though previous models were crucial to understanding and interpreting haltere neural encoding and haltere-mediated behaviors, they ignored strains that might occur due to the distributed mass of the haltere and the three dimensional structure of its base. We show that halteres with distributed mass exhibit twisting deformations in their stalks, and these deformations may be essential to haltere function. Such twisting deformations were shown to contribute to the sensory activity in the campaniform sensilla of moth wings [35], and we show here that similar forces act on the haltere. Halteres evolved from hindwings, and despite the dramatic alteration of the ancestral wing shape, we demonstrate here that these similar forces also influence sensing capabilities in halteres.

It is important to note that the geometries we used in this study are deliberately simplified in order to understand general behaviors of distributed masses responding to various accelerations.

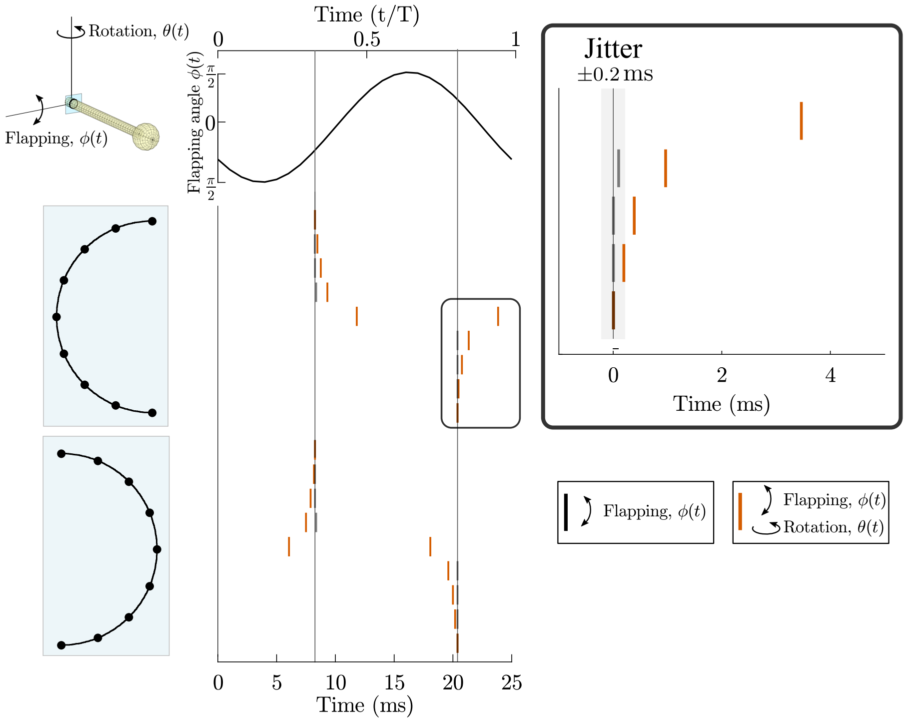

Figure 2.5: We use the spike prediction method to predict spiking along the circumference of the haltere for flapping (black stripes), and flapping with rotation (red stripes). At the side extremes, no spikes occur unless the haltere rotates, and at the top and bottom there is no timing difference. The intermediate locations on the left side of the haltere cross-section experience a spiking lag as a consequence of rotation. Conversely, the right side locations experience a leading time difference. This timing difference is for a single location is between 0.2 and 1 milliseconds, equal or larger than the spike timing variability of 0.2 milliseconds between stroke cycles observed in electrophysiological recordings of crane fly halteres [49].

Moreover, we used linear material properties whose values were inspired by prior measurements [62]. Additionally, the magnitudes we report here depend on the frequencies of flapping and the rates of body rotation which could vary immensely among diverse taxa and conditions. Despite these limitations, we can draw general conclusions about the modes of deformations and their relationship to geometric aspects of halteres.

### Coriolis and centrifugal forces deform halteres

Coriolis forces generate torsional moments about the stalk. The resultant twisting depends strongly on stalk geometry, and we demonstrate that a torsionally-compliant stalk has over 20 times more twist than the tubular stalk used in other simulations. Haltere bulbs show considerable morphological asymmetry [60], and our model shows that this asymmetry contributes to torsion in the haltere. We also find that if the center of mass of the haltere is mounted asymmetrically outside of the flapping plane, centrifugal forces causes an out-of-plane bending as well as haltere twisting, regardless of the body's rotation. This bending may also influence sensing ability of the haltere campaniform sensilla, and has not been previously considered. It would be interesting to explore in more detail the vast range of haltere bulb and stalk morpholpogies in the context of their responses to both Coriolis and centrifugal forces. Additionally, for asymmetric halteres, we found that the spike timing is altered only slightly by the additional centrifugal force associated with a horizontal asymmetry of the bulb (fig. 2.3). Unlike the results shown here, how asymmetry plays into the timing of spikes depends quite strongly on the value of parameters used here and represents an interesting future direction for understanding the sensory consequences of variation in haltere morphology.

### Predictions of spike times from strain patterns provide a mechanism for sensing rotation

As the fly's body rotates in the yaw plane, the timing of the spikes in neurons from campaniform sensilla at various locations on the haltere will change (fig. 2.5). These spike timing changes were predicted seventy years ago by Pringle [53] as a primary mechanism of encoding rotations; however, technological limitations prevented him from observing the spiking activity of single neurons. Recently single haltere neurons were shown to shift their spike timing when the haltere was oscillated dorso-ventrally (i.e. simulating flapping flight) in various lateral sweep locations (roughly stimulating the position of the haltere when it is pushed out of its natural stroke plane by a yaw body rotation) [61]. These experiments provided support for Pringle's hypothesis, and the results here are in agreement with them.

In the proposed mechanism, a laterally-positioned campaniform sensillum would excite a downstream neuron at different times in response to body rotations of different velocities. The timing of the post-synaptic excitation from this sensillum could then be compared to the timing of excitation from a dorsal or ventral sensillum, which do not shift their spike times with body rotation (fig. 2.5). This simple time shift of a single spike would provide a mechanism for a single central neuron to determine if the body is rotating, and its rotation speed could be determined by the elapsed time between post-synaptic excitations. When looking at the timing of action potentials of campaniform sensilla along the circumference of the haltere base, we find that the spike timing changes beyond the jitter threshold [49] as a result of body rotation.

We also note that the maximum phase shifts occurred in campaniform sensilla located on the anterior and posterior aspects (i.e. the lateral part of the stalk cross-section) on our haltere model. Campaniform sensilla located on the dorsal and ventral aspects (i.e. the top and bottom of the model's stalk) did not shift their phase significantly during rotations, maintaining a consistent firing phase with respect to the haltere's flapping. This result is consistent with the prediction by Pringle that sensilla of the dorsal scapal plate would maintain the firing time, and the sensilla of the dorsal basal plate would shift their timing during rotations [53].

We note that Pringle based this hypothesis on the orientation of the sensilla themselves: the domes of the dorsal scapal plate are oriented with maximal sensitivity in the plane parallel to the haltere stalk, whereas the domes of the dorsal basal plate are about 30 degrees rotated from this plane. In our simulations, the nodes of the model are not endowed with specific orientation sensitivity; rather their spike timing based on the spanwise strain ($\epsilon_{xx}$) close to the base of the haltere, as it is the dominant strain.

### Multiple nonlinearities set limits for detectable rotation rates

While the Coriolis force is linearly proportional to rotation rate, the nonlinearities of the haltere primary afferents can include a threshold below which they may not fire, resulting in a minimum Coriolis force (and thus a minimum rotation rate) that is necessary for specific primary afferents to fire spikes. The primary afferent neurons synapse onto motoneurons and interneurons that perform their own computations with the input they receive from the haltere. In some models of these neurons [7], [61]), they act as coincidence detectors, requiring simultaneous input from multiple sensilla before firing. The coincidental firing of multiple afferents is facilitated by the change in spike time that occurs during rotations (fig. 2.5, [61]). At lower rotation rates, the spike timing differences will be smaller (fig. 2.5) and, at some locations along the circumference, the difference might be on the same order as the jitter. This may set a lower limit for detectability, as the ratio of in-plane over out-of-plane bending depends on flapping frequency $f_\phi$ (see supplement). One possible consequence of this limit is that slower body rotations will not be sensed at higher wingbeat frequencies. In some species, then, it may be advantageous to limit body rotations to short, high-speed bursts like the body saccades seen in *Drosophila* [70]. These saccades are known to be useful for increasing the duration of stable vision, thus minimizing visual motion blur, and allows extraction of spatial information [1], [71], [72]. We suggest they may also ensure that the halteres are best able to detect and control the body's movement. Indeed there are head roll reflexes in response to haltere feedback—most of those occur at high rotations rates, whereas for slower rotation rates, head roll responses mostly follow the slower vision based feedback [1], [73].

In this paper we focus only on yaw perturbation on a single haltere, though saccades can occur with a combination of rotation about all three axes [74]. It is thus possible that the strain at the base of both halteres contains enough information to disentangle rotations about all three axes [55].

**Finite element modeling reveals new sensing modes and mechanisms**
Using 3D finite element models of simplified geometries, combined with neural encoding mechanisms, we show that the shape halteres can strongly influences timing of neuronal activity of mechanosensory cells distributed around the base of halteres. That timing will depend upon the rotation speed and flapping frequency and can be encoded and used to guide steering behaviors in a mechanism that is much faster than in the visual system. Moreover, we have shown that the haltere's shape plays a profound role in the patterns of haltere deformation. Though halteres evolved in only two insect orders (Diptera and Strepsiptera) [75], the general stalk-and-knob shape has been remarkably conserved, with variations on the theme found in different flies with different stalk lengths, bulb sizes, and stalk/bulb ratios [60]. By melding concepts from computational neuroscience, such as stimulus feature selectivity, with three dimensional structural dynamics of a mechanosensory organ we gain new insight into the functional organization of halteres. The approach can provide inroads to future work on the vast diversity of haltere geometries found among the Diptera.

# Chapter 3

## Learning Precisely Timed Feedforward Control of the Sensor-Denied Inverted Pendulum

Mohren T.L. , Daniel, T.L. and Brunton, S.L.

arXiv, 2019.

## Abstract

Time delays due to signal latency, computational complexity, and sensor-denied environments, pose a critical challenge in both engineered and biological control systems. In this work, we investigate biologically inspired strategies to develop predictive control laws for engineered systems with time delays of similar order to their system dynamics. We demonstrate this approach on the nonlinear pendulum with partially denied observations, so that it is only possible to measure the state of the system near the upright position. Given a large disturbance that overwhelms the local feedback controller, it is necessary to add or remove energy from the pendulum so that it returns to the upright position after one full revolution. The partial observation near the upright position introduces a significant delay between observations and the region where actuation is most effective. Thus, we develop a learning algorithm that integrates sensor information into a triggered feedforward control signal to overcome this delay with minimal computation, training data, and set of control decisions. This simple controller can serve as a model for many biological systems, and can be implemented in engineered systems with time delays.

## Introduction

Time delays between sensor measurements and control actions pose a significant challenge in engineered control systems, degrading robust performance and eventually leading to instability in the closed-loop system [76]. There are numerous sources of delay, including in the sensor and actuator hardware, in signal transduction, and in the computation of a control action. However, many biological systems maintain robust control performance despite delays of the same order as the system dynamics in their sensorimotor control circuits, providing proof-by-existence that it is possible to effectively manage these delays. For example, a baseball batter will initiate the swing before the ball leaves the pitcher's hand [77]. Similarly, information processing in the eyes and brain of a fly and other insects takes far longer than one of its wing strokes [5], [6]. Delay, in combination with limited computation power, a complex and uncertain environment, and a large set of control actions provides a compelling and relevant set of challenges for modern engineered control systems. Even when the control system has low latency, partially denied sensor environments will lead to delays between sensing and actuation. Many biological control systems employ a strategy of fast feedforward control in sensor-denied environments [78] that may be used in concert with a slow supervisory feedback control to achieve the incredible observed performance and robustness [79]. In this work, we will investigate biologically-inspired control strategies to learn triggered feedforward control actions that overcome time delays arising from a partially denied sensor. Event-based feedforward control with delays requires learning in both engineered and natural systems. In biology, motor babbling in infants underlies much of the learned feedforward controls that eventually develop [80]. In engineered systems, such motor babbling is data-intensive, involving either large simulations from which actions are learned or a physical instantiation in robotic devices [81].

There are considerable current research efforts to understand and distill how biological systems handle perturbations and large time delays in their control architectures [82]. In the nervous system of animals, information is conveyed through neurons by means of discrete action potentials. The timing of these discrete events conveys information and greatly affects muscle activation [24], [83]. Furthermore, event-triggered sensing and control are known to have advantages in energetic and computational cost compared to common control architectures in engineered systems [84]. In biology, event-based sensing is exceedingly common, to the extent that computation partially takes place at the sensor level [26]. Indeed, action potentials represent timing events that are computationally efficient and reduce noise [85].

The timing-based feedforward control strategy observed in biological systems, such as prey capture by dragonflies [86] or human motor control [87], motivates an investigation to determine if there are advantages to this approach in engineered systems, or if this is simply a biological idiosyncrasy. Time delays have been shown to have some advantages in control design, such as deadbeat control of continuous systems [88], stabilizing oscillatory systems [89], and in simplifying control design [90]. We formulate the timing-based feedforward control design as an optimization problem, and leverage the wide range of powerful optimization techniques. If this optimization is performed online from experiential data, then we may call this a *learning strategy.* Indeed, techniques in machine learning are being rapidly integrated into control design [91], including for model predictive control with deep neural networks [15], [16], reinforcement learning [18]–[20], genetic programming control in fluids [12]–[14], and iterative learning control. A number of compelling examples have explored optimization to learn biologically inspired maneuvers related to flight [92]–[94] and swimming [95]. Many of these learning approaches may be used to learn the optimal timing-based *triggered* feedforward control explored here.

To demonstrate timing-based control, we will investigate the simple pendulum, a common and well-studied benchmark for non-linear control [96]. There are several approaches to designing controllers for such nonlinear systems, such as energy methods [97], relying on passivity properties of the system [98], [99] and LQR-trees [100]. The controller typically has a hybrid form, consisting of a nonlinear controller for the swing up and a linearized feedback controller to stabilize the inverted equilibrium [96]. In the case of limited or saturated actuation, bang-bang control generally provides

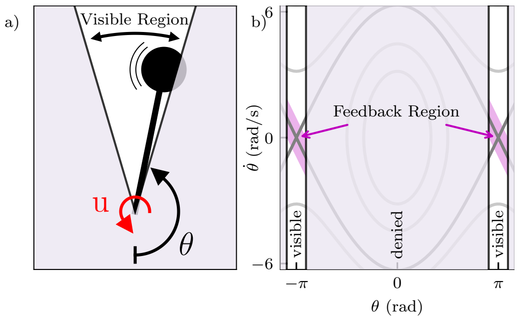

Figure 3.1: The pendulum model with denied sensory information, except in a small region near the inverted equilibrium. State measurements are available only in the region $\pi \pm 0.3$ radians. In the visible region, the pendulum can be controlled with state feedback. In the rest of the state space, measurements are denied. Open loop control torque is still possible in the denied region.

an optimal minimum-time controller [96], [101]. Discrete timing control has also been used on the pendulum [102].

Unlike these previous studies, we will consider the pendulum with partially denied observations (partially denied sensing), so that it is only possible to measure the state of the system near the upright position where feedback is effective, as shown in Fig. 3.1. When the system experiences a large disturbance and leaves the feedback region of convergence, it is necessary to add or remove energy so that it returns to the upright position after one full revolution. The optimal time to pump in energy is at the point of maximum velocity (shown in Fig. 3.2), as in the case of a child on a swing [103]. However, the system is not observable in this region of space, and so the proper control action must be pre-planned and timed based on the sensor information near the upright position. Thus, partial observation near the upright position and the maximum velocity occurring in the downward position introduces a delay, making this a suitable problem to explore our timing-based feedforward learning strategy. In particular, we leverage optimization based on limited measurement data from the pendulum to *learn* the appropriate control action and timing, demonstrating the ability to overcome the gap between information availability and control effectiveness.

## Benchmark problem

Our benchmark system is the simple frictionless pendulum with a single degree of freedom $\theta$ and a torque input $u$. With a massless rod, the pendulum equation of motion is

$$ml^2\ddot{\theta} = mgl\sin\theta + u, \tag{3.1}$$

which may be written in terms of $x_1 = \theta$ and $x_2 = \dot{\theta}$

$$\dot{x}_1 = x_2 \tag{3.2}$$

$$\dot{x}_2 = \frac{g}{l}\sin(x_1) + \frac{1}{ml^2}u. \tag{3.3}$$

We choose the parameters to be $l = 1\,\mathrm{m}$, $m = 1\,\mathrm{kg}$ and $g = -10\mathrm{ms}^2$. The controller saturates at a torque of $3\,\mathrm{Nm}$, which is the torque due to gravity at a pendulum angle of approximately $0.3\,\mathrm{rad}$

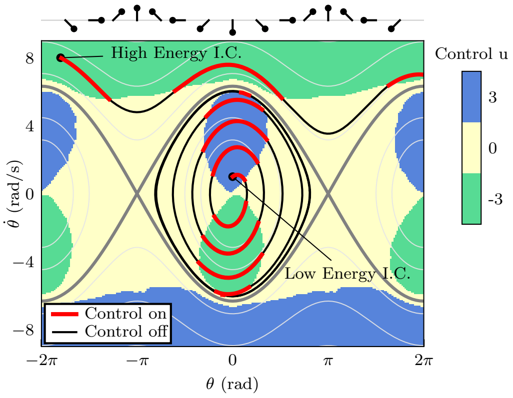

Figure 3.2: The state space diagram of the optimal controller for the simple pendulum with bang-bang control and cost ratio $R/Q = 100$, where R is control cost, and Q is the cost of error in the pendulum energy (see equation 3.4). Two initial conditions (I.C.s) are shown, one with a low energy starting point, requiring pumping up to obtain the inverted equilibrium, and one high energy starting point, requiring energy bleeding at certain phases of the state space. Blue and green regions of the state space are regions where control is on, as denoted by the red colored sections of the trajectories. These regions of effective actuation will be learned from an online optimization based entirely on information in the visible region in Fig. 3.1.

(17 degrees).

In our example, full observations of the state (i.e., pendulum angle and angular velocity) will be available in a limited region near the upright position, within the range $\pi - 0.3 \leq \theta \leq \pi + 0.3$. This leaves a narrow region that can be controlled by full state feedback. However, if a disturbance drives the pendulum outside of this visible region, the controller is sensor-blind and must use limited information from the visible region to pre-plan a triggered feedforward control action. This obscured state space is shown in Fig. 3.1.

Next, we will describe the full-state feedback controller near the pendulum-up configuration and the global cost function that is used to optimize the feedforward control in the sensor-denied region. The goal will be to learn the regions of effective actuation in Fig. 3.2, and the precise time to trigger actuation, from an online optimization based entirely on information in the visible region in Fig. 3.1. It is important to note that we have not added disturbances or noise, although this is an important avenue of ongoing work.

## Feedback control near the inverted fixed point

The pendulum has two fixed points, one at the neutrally stable downward position, and another at the unstable inverted position. After linearization about the inverted equilibrium $x_1 = \pi$, so that $\mathbf{x} = \left[\theta - \pi, \dot{\theta}\right]^T$, the pendulum can be made stable by applying full-state feedback control $u = -\mathbf{K}\mathbf{x}$:

$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x}.$$

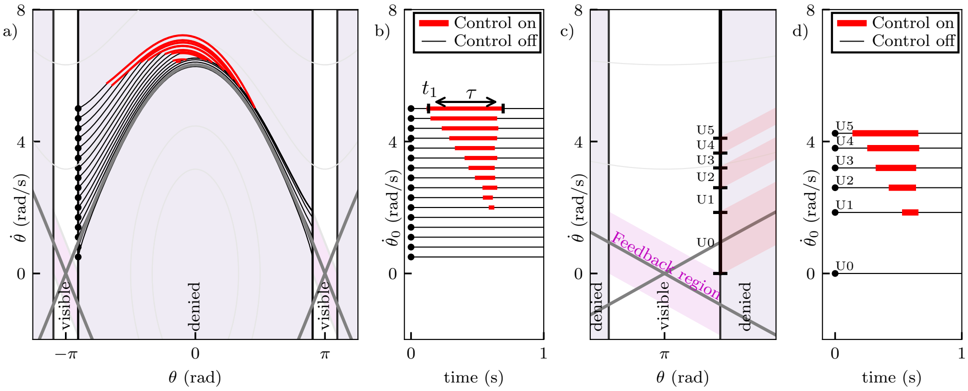

Figure 3.3: (a) For an array of initial velocities $\dot{\theta}_0$, we optimize the start-time $t_1$ and duration $\tau$ of the feedforward, bang-bang control protocols to drive the pendulum to the feedback attracting region after one revolution. (b) For low initial velocities $\dot{\theta}_0 \leq 1.8$, no control is required to swing back to the attractor. For increasing velocities, the start-time gets smaller, and the duration gets longer. (c) We find a subset of feedforward trajectories that converge into the feedback attractor. (d) By deciding on one out of these 6 protocols, any initial velocity $0 \geq \dot{\theta}_0 \leq 5$ will be returned to the feedback region.

We choose gains such that an eigenvector of the closed-loop system aligns with the homoclinic orbit connecting the unstable saddle at the inverted equilibrium to itself through one revolution of the pendulum:

$$\frac{x_2}{x_1} = \left.\frac{\partial\dot{\theta}}{\partial\theta}\right|_{\pi^-} = -\sqrt{\frac{-g}{l}}.$$

We then solve for the gains in $\mathbf{K}$, noting that

$$(\mathbf{A} - \mathbf{BK})\mathbf{x} = \lambda\mathbf{x} \quad \Longrightarrow \quad k_2 = k_1\sqrt{\frac{-g}{l}} + 2ml^2\sqrt{\frac{-g}{l}}.$$

Given a choice of $k_1 = 1000$, this results in a gain $K = [1000, 309.9]$ with saturation at $u = 3$Nm. This gain creates the region of attraction near the inverted equilibrium, referred to as the Feedback region in Fig. 3.3c.

## Reference control

Outside the feedback controller region of attraction, we aim to find a reference control law that effectively any initial condition to the inverted equilibrium within a reasonable amount of time and without expending too much actuation energy. This control law will be solved for using brute force and will provide a benchmark comparison for our learned controller in the next section. We define our cost function to be a cumulative penalty on a combination of state error and control effort. Instead of a standard state error term given by the Euclidean distance to the goal state, we instead use the difference between the total energy of the system and the goal energy at the inverted equilibrium. Energy is a particularly relevant quantity of interest for this problem. Moreover, it has been recently shown that control may be more effective in terms of eigenfunctions of the Koopman operator, such as the total energy function [104]. This approach is also closely related

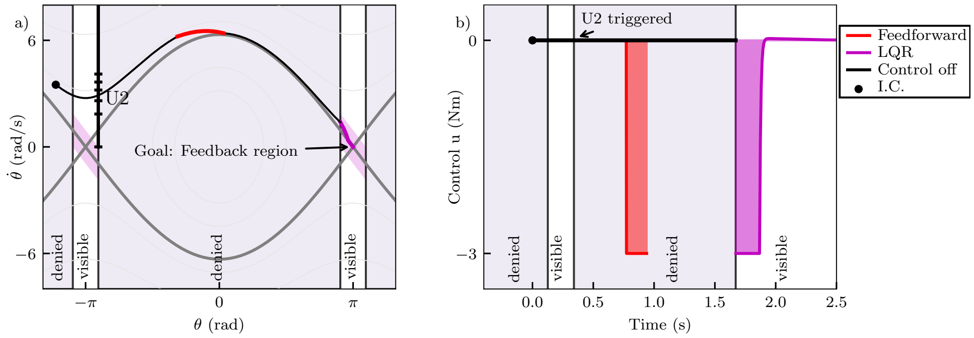

Figure 3.4: The combination of feedforward and feedback controller implemented on the partially observed pendulum. The pendulum first moves through the visible region, and as it enters the sensor-denied environment, the $U_2$ control protocol is selected. After time $t_1$ the control is triggered, remaining on (red) between $t_1 \leq t \leq t_1 + \tau$. The protocol is such that, once the pendulum reaches the visible region again, it now falls in the feedback region of attraction (purple). Figure (a) shows the trajectory in state space, (b) shows the timing and magnitude of the controller.

to Lyapunov functions, which are useful for stabilizing nonlinear systems. The cost is then given by:

$$J = \int_0^\infty \left(Qe^2 + Ru^2\right)\, d\tau, \tag{3.4}$$

with $e = E_{\text{total}} - E_{\text{final}}$, $Q = 1$, and $R = 100$. Outside of the feedback region we use a bang-bang controller with $u \in \{-3, 0, 3\}$Nm.

Before describing the results of our procedure to learn the control given only information in the sensor-visible region, here we derive an effective controller through brute-force optimization with full information via dynamic programming [105], shown in Fig. 3.2. This bang-bang controller will provide a benchmark for our learning results in the next section.

With the prescribed cost function, control is active only during phases of the swing where velocity is highest, and thus the most change in energy is achieved:

$$\Delta E = \int_{t_1}^{t_1+\Delta t} u\dot{\theta}(\tau)\, d\tau. \tag{3.5}$$

From a low-energy initial condition, it will take multiple swings to pump up to the homoclinic orbit (Fig. 3.2).

## Learning Time-Triggered Control

Now we consider the problem of learning a near-optimal predictive control law when it is impossible to measure the state when the control is most effective. In particular, we will consider trajectories originating in the sensor-visible window that have been pushed outside of the feedback effective region by a large disturbance. This controller will necessarily need to overcome the time-delay between sensing and effective actuation, which will manifest as a pre-planned control strategy consisting of bang-bang control at a precise future time. The decision on what feedforward protocol to use is made on the decision boundary, which we define to be the boundary in state space where the pendulum leaves the visible region. The feedforward controller is parameterized by the time $t_1$(s) from the decision boundary that the bang-bang control is turned on, and the duration of control $\tau$(s). Note that the control parameters will be a function of the energy of the pendulum as it leaves the visible region, as shown in Fig. 3.3. The control parameters are optimized to minimize the cost function from above, with an additional penalty for failing to reach the feedback region of attraction at the end of one revolution. First, we will optimize the control parameters in an *offline* learning procedure to demonstrate the process. Afterward, we will describe an *online* exploration and optimization procedure that provides a more realistic and useful learning strategy.

Starting from an array of initial conditions on the decision boundary, we find the associated set of $t_1$ and $\tau$ parameters (Fig. 3.3b) through optimization. The cost function is not differentiable over $t_1$ and $\tau$, and there is a large "valley" of near-optimal parameter combinations (see appendix for details). Therefore, we use a Nelder-Mead optimization, in combination with basin-hopping to escape potential local minima. Obtaining start-times and durations for various trajectories leaving the visible region, we observe a near-linear increase in duration. As the initial velocity increases, the total energy of the pendulum increases quadratically, whereas total change in energy achieved over the same time increases linearly with velocity (Eq. 3.5), thus requiring a longer control duration. We can attribute the earlier start-times $t_1$ with increasing initial velocity $\dot{\theta}_0$ to two causes: 1) the entire trajectory is faster, so the apex of the trajectory occurs earlier, and 2), the longer duration control-on phase has to be centered around the apex, thus leading to an earlier start-time. Furthermore, we note that trajectories with low initial velocity ($\dot{\theta}_0 \leq 1.8$ rad/s) reach the feedback region without requiring the controller to turn on during the obscured region (Fig. 3.3b). Interestingly, there are parts of the visible state space outside of the feedback region, where the only way to reach the feedback region is to swing through without any control. For higher initial velocities $\dot{\theta}_0 > 6.1$ rad/s (not shown in Fig. 3.3), the feedforward controller will be on at all times, while still not reaching the inverted equilibrium.

Next, we seek to reduce this set of feedforward controllers into a minimal set, which may be interpreted as a decision on the visible boundary about which feedforward control protocol to trigger (see Fig. 3.3c-d). In the biological setting, it is feasible that there may be advantages to having a smaller set of controllers to choose from. We determine this minimal set by starting with the lowest initial condition protocol, and increasing the initial velocity until the control protocol no longer drives the trajectory to the feedback region within a single revolution of the pendulum. At that limiting initial velocity, we then find the control protocol with the highest initial velocity that still drives the pendulum to the feedback region within one revolution. This procedure is continued until we obtain a minimal set of 6 control protocols that will drive initial velocities $0 \leq \dot{\theta}_0 \leq 5$ to the feedback region (Fig. 3.3d). We observe that for lower velocities, no action is required to return to the inverted equilibrium. For higher velocities, the duration increases and the start time decreases, which is to be expected since the protocols are drawn from the larger set of protocols in Fig. 3.3. We note that for higher initial velocities the protocols are spaced closer together. This is again due to the kinetic energy increasing quadratically.

```
if In feedback region then
    $u = -Kx$, with $|u| \leq 3$;
else
    if Crossing decision boundary then
        Select control protocol $U_j$ ;
        look up parameters $t_1$ and $\tau$ belonging to $U_j$ ;
        Start timer $t$ ;
        u = 0 ;
    else
        if $t_1 \leq t \leq (t_1 + \tau)$ then
            $u = -3$ ;
        else
            $u = 0$ ;
        end
    end
end
```

**Algorithm 1:** Decision scheme for the combined feedback and feedforward controller.

We now implement the small subset of triggered feedforward protocols in combination with feedback control in the region of attraction according to algorithm 1. When the trajectory crosses the decision boundary a feedforward control protocol is triggered with the parameters $t_1$ and $\tau$. After time $t_1$ the bang-bang control is triggered for duration $\tau$, after which the pendulum passively evolves to the region where feedback control is activated. If the system is in the visible region, but not in the feedback region, no control is exerted. The implementation of this combined controller is shown in Fig. 3.4. The pendulum starts from the initial condition in the obscured region, moves through the visible region, where it is observed but then moves too fast to be stabilized to the equilibrium. As it moves through the decision boundary to the obscured region, the $U_2$ protocol is triggered. Now out of view, the controller is active between 0.43 and 0.60 seconds and is brought back close to the homoclinic orbit. When back in view, the pendulum enters the feedback region and it is stabilized around the inverted equilibrium with feedback.

Lastly, we continuously explore the state space in an *online* learning procedure, without resetting the simulation between initial conditions (Fig.3.5a). Because the controller removes energy from the system, we explore new initial conditions by adding energy in the visible region after each revolution. When the pendulum leaves the visible region, we try a set of $t_1$ and $\tau$, and evaluate the cost of the trial once it reaches the visible region again. We discretize the cost matrix as follows: initial velocities $\dot{\theta}_0$ in bins with a 0.4 rad/s width in the range $[0, 5]$, $\tau$ in steps of 0.05 seconds in the range $[0, 0.6]$ seconds, and $t_1$ in steps of 0.05 in the range $[0.2, 0.6]$ seconds. To prevent the controller bleeding too much energy and thereby not entering the visible region again, we build up $t_1$ and $\tau$ from zero with discrete increments, starting from the lowest initial velocities. Once $\tau$ is sufficiently large to drive the pendulum back to the feedback region, we stop exploring larger $\tau$'s from that initial condition and explore higher initial velocities. Using the online learning procedure, the control law was found using 1072 trials, requiring approximately 211 seconds of computation on a laptop with 1.8GHz processor and 8 Gb RAM. This continuous exploration shows that with far fewer trials the set of protocols still approximates the brute-force solution solution (Fig. 3.4b). However, the smaller number of trials comes at the cost of precision in both

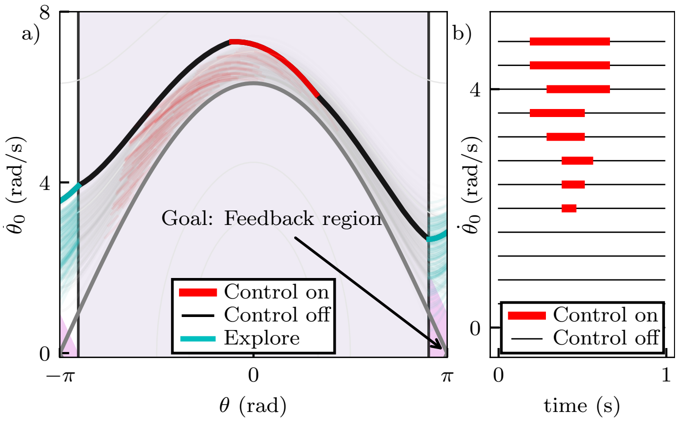

Figure 3.5: We implement an online learning strategy that optimizes the parameters of the feedforward controller. We use an exploration protocol where the controller adds energy to the system to set up higher velocity initial conditions (a). Each time the pendulum leaves the visible region, a new combination of feedforward parameters $t_1$ and $\tau$ is tested. Using this continuous exploration and optimization, we find a set of controllers that approximates the brute-force control solution (b).

start-time and duration, although this may be refined with more data. Furthermore, by binning the initial conditions, suboptimal combinations might lead to the lowest observed cost as a result of a favorable initial condition. The trade-off between number of trials and performance will be an interesting direction for future research. It is worth noting that this approach is closely related to reinforcement learning, and advances in this field may improve the performance of the trial-based online learning in this example.

## Conclusions

In this work we explored how to learn a timing-based control strategy inspired by biological systems. In particular, we developed a hybrid feedforward and feedback controller for a nonlinear pendulum where it is only possible to measure the system near the upright configuration. As is the case for motor babbling, we simulated a vast number of control actions and used a data-driven approach to learn the set of control actions given limited sensing and delay through a data-driven optimization procedure. We found that it is possible to optimize a triggered feedforward controller using only information available in the visible region. For simplicity, we restricted ourselves to a bang-bang controller with two parameters – the start time and duration. The two-parameter solution was learned with relatively few trials, as is often the case in natural systems. By grouping the feedforward protocols into a minimal set, we reduced the number of choices while still driving the system to the upright position. Finally, we have also developed a more realistic online learning procedure, consisting of both exploration and parameter learning.

There are a number of important future avenues suggested by this work. First, we did not consider the effect of measurement noise or exogenous disturbances outside of the visible region. This work is meant to be a proof of concept and to develop a benchmark problem to explore biological learning and control strategies. In the future, it will be important to explore the robustness of these strategies to noise and disturbances. As in most machine learning algorithms, it is difficult to prove any performance or convergence guarantees, although it is believed that this approach is likely to be robust to small random noise and disturbances. However, it is possible to think of several types of *worst-case scenario* disturbances that might break this controller, and carefully characterizing the robustness is an important avenue of future work. More generally, the interface of learning and control will need to address the issue of certifiability and other guarantees, likely with uncertainty quantification. Though the decision on control protocol was based solely on the system state when crossing the decision boundary, measurement noise could be addressed by using a state estimator for the the entire presence in the visible region. Indeed, nonlinear neural-inspired temporal filters may be incorporated into the decision process serving that function. Such neural-inspired approaches are known to improve decisions and classifications in biological systems [26]. We can imagine a scenario where measurements are in fact not obscured, but where measurement noise is time dependent. Feedforward protocols can then be triggered when the certainty in the state estimate drops below some desired level. While many engineered systems currently rely on continuous sensory information and controllability, many biological systems have found ways to deal with time-varying observability and control authority. Thus, although this paper tested timing-based feedforward on a simple non-linear problem with favorable passivity properties, we believe that applying these ideas to more complex systems will be a fruitful area of future work.

# Acknowledgements

This work was supported by the Komen Endowed Chair and the Washington Research Foundation to TLD, the Air Force Office of Scientific Research (AFOSR) under awards FA9550-14-1-0398 to TLD and FA9550-19-1-0386 to SLB and TLD, and the Army Research Office (ARO) under awards W911NF-19-1-0045 and W911NF-17-1-0306 to SLB. We would also like to thank Bing Brunton for valuable discussions on sparse sensor systems.

# Code and Data

All code can be found on `https://github.com/tlmohren/timed_feedforward_control`.

# Conclusions & Discussion

In solving some of the complex engineering challenges, we stand to learn a lot from living systems. The encoding of sensory information to event-based action potentials is not only an energy efficient and noise-reducing method of communication [85], but is also a form of distal computation that allows simple downstream decisions. Distributed arrays are able to detect small differences on top of large salient signals. Timing-based feed-forward controls do not only allow for exploiting time-varying sensor and control effectiveness, but also require much fewer iterations to learn a near-optimal controller. Contrary to most engineered systems, timing is ubiquitous for sensing and control in living system.

During my thesis there were several research avenues that I was not able to delve into. The campaniform sensilla at the base of halteres have been shown to simultaneously select for two features [8]. Using a probabilistic firing method that selects for more features will likely improve the classification ability, potentially requiring less sensors for the same classification task. The current algorithm classifies rotation state every snapshot taken. However, the insect likely can only detect thee state once per wing stroke because of the 3 millisecond refractory period of neurons. Would the pooling of information in time by using different sensory delays help the classification of flight state? The optimal sensor locations would likely differ from those found using the single feature neural encoding model. Furthermore, there is still much to be learned about sensor placement when combining neural encoding and sparse placement algorithms. How critical is the placement of the sensor? Would a sensor in close vicinity do just as well?

The structural models used in these studies were intentionally simplified to gain mechanistic insights into underlying principles. This generality comes at the cost of making inferences on specific species, and of specific mechanisms. Since accurate descriptions of biological material properties are notoriously difficult, it will be very challenging to create high-fidelity models that would lead to very specific mechanistic understandings. Yet I believe more specificity in particular cases will certainly lead to deeper insights. As the haltere paper touched upon, it is hard to imagine the abundant haltere asymmetry to be without function. The campaniform field locations are so well preserved across species, with such a defined grid pattern in many species, that one cannot help to conclude that interesting mechanisms must be at work here. Even at the campaniform level, we lack a thorough understanding of how local strain is transformed into vertical displacement, before leading to an action potential. Mechanoreception in halteres is still a rich area for future discoveries.

More so than the other two projects the pendulum study initiated a broad range of potential experiments and avenues. Much like in living systems, the drivers for least control effort and state error minimization might oppose the desire for more accurate state measurements and certainty of control efficacy. The sensor-denied pendulum can serve as a model of a system where the sensors are not totally denied, but where state estimate might have variable accuracy due to increased sampling time or variable sensor noise. One can think of a Kalman-filter type sensor fusion where sensor accuracy is not constant during the cycle. Furthermore, we tested the timing-based feedforward controller on the 1-dimensional pendulum, but it would be very interesting to see if we could find a similar near-optimal set of timing parameters for a very high-dimensional system with an unknown low-dimensional representation. In the current system we exploit different regions of state space for both sensor and actuator effectiveness, but in the presence of disturbance of system noise we might choose to exploit different properties. Perhaps we value not maximum actuator efficiency, but the fact that the derivative of actuator efficiency is small. This way we have a greater certainty of the impulse applied to the system, though for the case of the single pendulum the greatest actuator efficiency coincides with the lowest efficiency derivative. To test these ideas we could use a systems like the cart-pendulum where the pendulum is flexible, and where we have strain measurements along it's length. This introduces noise, model uncertainty and time-varying sensor accuracy. With more such studies, we can start to transform principles of timing-based feedforward to tested engineering solutions.

# Bibliography

[1] S. J. Huston and H. G. Krapp, "Nonlinear integration of visual and haltere inputs in fly neck motor neurons," *Journal of Neuroscience*, vol. 29, no. 42, pp. 13 097–13 105, 2009.

[2] S. B. Fuller, A. D. Straw, M. Y. Peek, R. M. Murray, and M. H. Dickinson, "Flying drosophila stabilize their vision-based velocity controller by sensing wind with their antennae," *Proceedings of the National Academy of Sciences*, vol. 111, no. 13, E1182–E1191, 2014.

[3] A. M. Mountcastle and S. A. Combes, "Biomechanical strategies for mitigating collision damage in insect wings: Structural design versus embedded elastic materials," *Journal of Experimental Biology*, vol. 217, no. 7, pp. 1108–1115, 2014.

[4] L. Ristroph, G. Ristroph, S. Morozova, A. J. Bergou, S. Chang, J. Guckenheimer, Z. J. Wang, and I. Cohen, "Active and passive stabilization of body pitch in insect flight," *Journal of The Royal Society Interface*, vol. 10, no. 85, p. 20 130 237, 2013.

[5] M. F. Land and T. S. Collett, "Chasing behaviour of houseflies (fannia canicularis)," *Journal of comparative physiology*, vol. 89, no. 4, pp. 331–357, 1974.

[6] J. C. Theobald, E. J. Warrant, and D. C. O'Carroll, "Wide-field motion tuning in nocturnal hawkmoths," *Proceedings of the Royal Society of London B: Biological Sciences*, rspb20091677, 2009.

[7] A. Fayyazuddin and M. H. Dickinson, "Haltere afferents provide direct, electrotonic input to a steering motor neuron in the blowfly, calliphora," *Journal of Neuroscience*, vol. 16, no. 16, pp. 5225–5232, 1996.

[8] J. L. Fox, A. L. Fairhall, and T. L. Daniel, "Encoding properties of haltere neurons enable motion feature detection in a biological gyroscope," *Proceedings of the National Academy of Sciences*, vol. 107, no. 8, pp. 3840–3845, 2010.

[9] F. Rieke, D. Warland, R. D. R. Van Steveninck, W. S. Bialek, *et al.*, *Spikes: exploring the neural code*, 1. MIT press Cambridge, 1999, vol. 7.

[10] B. Pratt, T. Deora, T. Mohren, and T. Daniel, "Neural evidence supports a dual sensory-motor role for insect wings," *Proceedings of the Royal Society B: Biological Sciences*, vol. 284, no. 1862, p. 20 170 969, 2017.

[11] N. Minorsky, "Directional stability of automatically steered bodies," *Journal of the American Society for Naval Engineers*, vol. 34, no. 2, pp. 280–309, 1922.

[12] T. Duriez, S. L. Brunton, and B. R. Noack, *Machine Learning Control: Taming Nonlinear Dynamics and Turbulence*. Springer, 2016.

[13] S. L. Brunton and B. R. Noack, “Closed-loop turbulence control: Progress and challenges,” *Applied Mechanics Reviews*, vol. 67, pp. 050 801–1–050801–48, 2015.

[14] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, “Machine learning for fluid mechanics,” *Annual Review of Fluid Mechanics*, vol. 52, 2020.

[15] I. Lenz, R. A. Knepper, and A. Saxena, “Deepmpc: Learning deep latent features for model predictive control.,” in *Robotics: Science and Systems*, Rome, Italy, 2015.

[16] E. Kaiser, J. N. Kutz, and S. L. Brunton, “Sparse identification of nonlinear dynamics for model predictive control in the low-data limit,” *Proc. R. Soc. A*, vol. 474, no. 2219, 2018.

[17] T. Baumeister, S. L. Brunton, and J. N. Kutz, “Deep learning and model predictive control for self-tuning mode-locked lasers,” *JOSA B*, vol. 35, no. 3, pp. 617–626, 2018.

[18] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, 1. MIT press Cambridge, 1998, vol. 1.

[19] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, p. 529, 2015.

[20] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning,” *arXiv preprint arXiv:1509.02971*, 2015.

[21] B. H. Dickerson, Z. N. Aldworth, and T. L. Daniel, “Control of moth flight posture is mediated by wing mechanosensory feedback,” *Journal of Experimental Biology*, vol. 217, no. 13, pp. 2301–2308, 2014.

[22] A. Sherman and M. H. Dickinson, “A comparison of visual and haltere-mediated equilibrium reflexes in the fruit fly drosophila melanogaster,” *Journal of Experimental Biology*, vol. 206, no. 2, pp. 295–302, 2003.

[23] J. Haag, A. Wertz, and A. Borst, “Central gating of fly optomotor response,” *Proceedings of the National Academy of Sciences*, vol. 107, no. 46, pp. 20 104–20 109, 2010.

[24] T. Mohren, T. Daniel, A. Eberle, P. Reinhall, and J. Fox, “Coriolis and centrifugal forces drive haltere deformations and influence spike timing,” *J. Roy. Soc. Interface*, vol. 16, no. 153, p. 20 190 035, 2019.

[25] T. L. Mohren, T. L. Daniel, and S. L. Brunton, “Learning precisely timed feedforward control of the sensor-denied inverted pendulum,” *arXiv preprint arXiv:1912.04922*, 2019.

[26] T. L. Mohren, T. L. Daniel, S. L. Brunton, and B. W. Brunton, “Neural-inspired sensors enable sparse, efficient classification of spatiotemporal data,” vol. 115, no. 42, pp. 10 564–10 569, 2018.

[27] G. Nalbach and R Hengstenberg, “The halteres of the blowfly calliphora,” *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, vol. 175, no. 6, pp. 695–708, 1994.

[28] S. P. Sane, A. Dieudonné, M. A. Willis, and T. L. Daniel, “Antennal mechanosensors mediate flight control in moths,” *science*, vol. 315, no. 5813, pp. 863–866, 2007.

[29] G. K. Taylor and H. G. Krapp, "Sensory systems and flight stability: What do insects measure and why?" *Advances in insect physiology*, vol. 34, pp. 231–316, 2007.

[30] T. Collett and M. Land, "Visual control of flight behaviour in the hoverflysyritta pipiens l.," *Journal of Comparative Physiology*, vol. 99, no. 1, pp. 1–66, 1975.

[31] U. J. Dombrowski, "Untersuchungen zur funktionellen organisation des flugsystems von manduca sexta (l)," *Dissertation,*, 1991.

[32] E. S. Cole and J. Palka, "The pattern of campaniform sensilla on the wing and haltere of drosophila melanogaster and several of its homeotic mutants," *Development*, vol. 71, no. 1, pp. 41–61, 1982.

[33] M. H. Dickinson, "Haltere–mediated equilibrium reflexes of the fruit fly, drosophila melanogaster," *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, vol. 354, no. 1385, pp. 903–916, 1999.

[34] S. A. Combes and T. L. Daniel, "Into thin air: Contributions of aerodynamic and inertial-elastic forces to wing bending in the hawkmoth manduca sexta," *Journal of Experimental Biology*, vol. 206, no. 17, pp. 2999–3006, 2003.

[35] A. L. Eberle, B. H. Dickerson, P. G. Reinhall, and T. L. Daniel, "A new twist on gyroscopic sensing: Body rotations lead to torsion in flapping, flexing insect wings," *Journal of the Royal Society Interface*, vol. 12, no. 104, p. 20 141 088, 2015.

[36] B. Hassenstein and W Reichardt, "Systemtheoretische analyse der zeit-reihenfolgen- und vorzeichenauswertung bei der bewegungsperzeption des rüsselkäfers chlorophanus," *Zeitschrift für Naturforschung11b*, pp. 513–524, 1956.

[37] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[38] M. Loève, *Probability theory: foundations, random sequences.* van Nostrand Princeton, NJ, 1955.

[39] H. Hotelling, "Analysis of a complex of statistical variables into principal components.," *Journal of educational psychology*, vol. 24, no. 6, p. 417, 1933.

[40] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936.

[41] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on information theory*, vol. 52, no. 2, pp. 489–509, 2006.

[42] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE transactions on information theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[43] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[44] B. Brunton, S. Brunton, J. Proctor, and J. Kutz, "Sparse sensor placement optimization for classification," *SIAM Journal on Applied Mathematics*, vol. 76, no. 5, pp. 2099–2122, 2016.

[45] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 451–462, 2009.

[46] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[47] J. Aljadeff, B. J. Lansdell, A. L. Fairhall, and D. Kleinfeld, "Analysis of neuronal spike trains, deconstructed," *Neuron*, vol. 91, no. 2, pp. 221–259, 2016.

[48] S. Combes and T. Daniel, "Flexural stiffness in insect wings ii. spatial distribution and dynamic wing bending," *Journal of Experimental Biology*, vol. 206, no. 17, pp. 2989–2997, 2003.

[49] J. L. Fox and T. L. Daniel, "A neural basis for gyroscopic force measurement in the halteres of holorusia," *Journal of Comparative Physiology A*, vol. 194, no. 10, pp. 887–897, 2008.

[50] M. Grant and S. Boyd, *CVX: Matlab software for disciplined convex programming, version 2.1*, `http://cvxr.com/cvx`, Mar. 2014.

[51] M. Grant, V. B. Stephen Boyd, S. Boyd, and H. Kimura, "Graph implementations for nonsmooth convex programs, in recent advances in learning and control," *Springer-Verlag Limited*, Lecture Notes in Control and Information Sciences, pp. 95–110, 2008.

[52] M. H. Dickinson, "Comparison of encoding properties of campaniform sensilla on the fly wing," *Journal of experimental Biology*, vol. 261, pp. 245–261, 1990.

[53] J. W. S. Pringle, "The gyroscopic mechanism of the halteres of diptera," *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, vol. 233, no. 602, pp. 347–384, 1948.

[54] G Nalbach, "The halteres of the blowfly calliphora," *Journal of Comparative Physiology A*, vol. 173, no. 3, pp. 293–300, 1993.

[55] R. A. Thompson, M. F. Wehling, J. H. Evers, and W. E. Dixon, "Body rate decoupling using haltere mid-stroke measurements for inertial flight stabilization in diptera," *Journal of Comparative Physiology A*, vol. 195, no. 1, pp. 99–112, 2009.

[56] D. C. SANDEMAN and H Markl, "Head movements in flies (calliphora) produced by deflexion of the halteres," *Journal of Experimental Biology*, vol. 85, no. 1, pp. 43–60, 1980.

[57] J. W. S. Pringle, "The function of the lyriform organs of arachnids," *Journal of Experimental Biology*, vol. 32, no. 2, pp. 270–278, 1955.

[58] A. S. French and P. H. Torkkeli, "Mechanotransduction in spider slit sensilla," *Canadian journal of physiology and pharmacology*, vol. 82, no. 8-9, pp. 541–548, 2004.

[59] F. G. Barth, "Spider mechanoreceptors," *Current opinion in neurobiology*, vol. 14, no. 4, pp. 415–422, 2004.

[60] S. Agrawal, D. Grimaldi, and J. L. Fox, "Haltere morphology and campaniform sensilla arrangement across diptera," *Arthropod structure & development*, vol. 46, no. 2, pp. 215–229, 2017.

[61] A. M. Yarger and J. L. Fox, "Single mechanosensory neurons encode lateral displacements using precise spike timing and thresholds," *Proc. R. Soc. B*, vol. 285, no. 1887, p. 20 181 759, 2018.

[62] R. Parween, R. Pratap, T. Deora, and S. P. Sane, "Modeling strain sensing by the gyroscopic halteres, in the dipteran soldier fly, hermetia illucens," *Mechanics Based Design of Structures and Machines*, vol. 42, no. 3, pp. 371–385, 2014.

[63] J. F. Vincent and U. G. Wegst, "Design and mechanical properties of insect cuticle," *Arthropod structure & development*, vol. 33, no. 3, pp. 187–199, 2004.

[64] C. P. Ellington, "The aerodynamics of hovering insect flight. vi. lift and power requirements," *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, vol. 305, no. 1122, pp. 145–181, 1984.

[65] T. Deora, A. K. Singh, and S. P. Sane, "Biomechanical basis of wing and haltere coordination in flies," *Proceedings of the National Academy of Sciences*, vol. 112, no. 5, pp. 1481–1486, 2015.

[66] T. L. Hedrick, B. Cheng, and X. Deng, "Wingbeat time and the scaling of passive rotational damping in flapping flight," *Science*, vol. 324, no. 5924, pp. 252–255, 2009.

[67] A. M. Bloch, P. S. Krishnaprasad, J. E. Marsden, and T. S. Ratiu, "Dissipation induced instabilities," in *Annales de l'Institut Henri Poincare (C) Non Linear Analysis*, Elsevier, vol. 11, 1994, pp. 37–90.

[68] J. M. Gere and S. P. Timoshenko, "Mechanics of materials, 1997," *PWS-KENT Publishing Company, ISBN 0*, vol. 534, no. 92174, p. 4, 1997.

[69] MATLAB, *version 9.3.0 (R2017b)*. Natick, Massachusetts: The MathWorks Inc., 2017.

[70] J.-M. Mongeau and M. A. Frye, "Drosophila spatiotemporally integrates visual signals to control saccades," *Current Biology*, vol. 27, no. 19, pp. 2901–2914, 2017.

[71] C Schilstra and J. H. Van Hateren, "Stabilizing gaze in flying blowflies," *Nature*, vol. 395, no. 6703, p. 654, 1998.

[72] R. Kern, J. H. Van Hateren, C. Michaelis, J. P. Lindemann, and M. Egelhaaf, "Function of a fly motion-sensitive neuron matches eye movements during free flight," *PLoS biology*, vol. 3, no. 6, e171, 2005.

[73] R. Hengstenberg, "Multisensory control in insect oculomotor systems," *Rev. Oculomot. Res*, vol. 5, pp. 285–298, 1993.

[74] B. R. H. Geurten, R. Kern, E. Braun, and M. Egelhaaf, "A syntax of hoverfly flight prototypes," *Journal of Experimental Biology*, vol. 213, no. 14, pp. 2461–2475, 2010.

[75] D. R. Angelini and T. C. Kaufman, "Comparative developmental genetics and the evolution of arthropod body plans," *Annu. Rev. Genet.*, vol. 39, pp. 95–119, 2005.

[76] J. C. Doyle, B. A. Francis, and A. R. Tannenbaum, *Feedback control theory.* Courier Corporation, 2013.

[77] S. Müller and B. Abernethy, "Expert anticipatory skill in striking sports: A review and a model," *Research Quarterly for Exercise and Sport*, vol. 83, no. 2, pp. 175–187, 2012.

[78] S. Cheng, “Reaching a target within a gps-denied or costly area: A two-stage optimal control approach,” PhD thesis, 2018.

[79] N. J. Cowan, M. M. Ankarali, J. P. Dyhr, M. S. Madhav, E. Roth, S. Sefati, S. Sponberg, S. A. Stamper, E. S. Fortune, and T. L. Daniel, “Feedback control as a framework for understanding tradeoffs in biology,” *American Zoologist*, vol. 54, no. 2, pp. 223–237, 2014.

[80] A. N. Meltzoff and M. K. Moore, “Explaining facial imitation: A theoretical model,” *Infant and child development*, vol. 6, no. 3-4, pp. 179–192, 1997.

[81] K. Takahashi, T. Ogata, H. Yamada, H. Tjandra, and S. Sugano, “Effective motion learning for a flexible-joint robot using motor babbling,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2015, pp. 2723–2728.

[82] R. J. Full and D. E. Koditschek, “Templates and anchors: Neuromechanical hypotheses of legged locomotion on land,” *Journal of experimental biology*, vol. 202, no. 23, pp. 3325–3332, 1999.

[83] S Sponberg and T. Daniel, “Abdicating power for control: A precision timing strategy to modulate function of flight power muscles,” *Proc. R. Soc. B*, vol. 279, no. 1744, pp. 3958–3966, 2012.

[84] W. Heemels, K. H. Johansson, and P. Tabuada, “An introduction to event-triggered and self-triggered control,” in *51st IEEE CDC*, 2012, pp. 3270–3285.

[85] B. Sengupta, M. Stemmler, S. B. Laughlin, and J. E. Niven, “Action potential energy efficiency varies among neuron types in vertebrates and invertebrates,” *PLoS computational biology*, vol. 6, no. 7, e1000840, 2010.

[86] M. Mischiati, H.-T. Lin, P. Herold, E. Imler, R. Olberg, and A. Leonardo, “Internal models direct dragonfly interception steering,” *Nature*, vol. 517, no. 7534, p. 333, 2015.

[87] A. J. Bastian, “Learning to predict the future: The cerebellum adapts feedforward movement control,” *Current opinion in neurobiology*, vol. 16, no. 6, pp. 645–649, 2006.

[88] K Watanabe, E Nobuyama, and A Kojima, “Recent advances in control of time delay systems-a tutorial review,” in *Proc. 35th IEEE CDC*, vol. 2, 1996, pp. 2083–2089.

[89] C Abdallah, P. Dorato, J Benites-Read, and R Byrne, “Delayed positive feedback can stabilize oscillatory systems,” in *1993 American Control Conference*, IEEE, 1993, pp. 3106–3107.

[90] J. Lavaei, S. Sojoudi, and R. M. Murray, “Simple delay-based implementation of continuous-time controllers,” in *Proceedings of the 2010 American Control Conference*, IEEE, 2010, pp. 5781–5788.

[91] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control.* Cambridge University Press, 2019.

[92] H. J. Kim, M. I. Jordan, S. Sastry, and A. Y. Ng, “Autonomous helicopter flight via reinforcement learning,” in *Advances in neural information processing systems*, 2004, pp. 799–806.

[93] R. Tedrake, Z. Jackowski, R. Cory, J. W. Roberts, and W. Hoburg, “Learning to fly like a bird,” in *14th International Symposium on Robotics Research. Lucerne, Switzerland*, 2009.

[94] G. Novati, L. Mahadevan, and P. Koumoutsakos, "Controlled gliding and perching through deep-reinforcement-learning," *Physical Review Fluids*, vol. 4, no. 9, p. 093 902, 2019.

[95] S. Verma, G. Novati, and P. Koumoutsakos, "Efficient collective swimming by harnessing vortices through deep reinforcement learning," *Proc. Nat. Acad. Sci.*, vol. 115, no. 23, pp. 5849–5854, 2018.

[96] O. Boubaker, "The inverted pendulum: A fundamental benchmark in control theory and robotics," in *Education and e-Learning Innovations (ICEELI), 2012 international conference on*, IEEE, 2012, pp. 1–6.

[97] K. J. Åström and K. Furuta, "Swinging up a pendulum by energy control," *Automatica*, vol. 36, no. 2, pp. 287–295, 2000.

[98] R. Lozano and I. Fantoni, "Passivity based control of the inverted pendulum," in *Perspectives in control*, Springer, 1998, pp. 83–95.

[99] R. Ortega, A. J. Van Der Schaft, I. Mareels, and B. Maschke, "Putting energy back in control," *IEEE Control Systems Magazine*, vol. 21, no. 2, pp. 18–33, 2001.

[100] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts, "LQR-trees: Feedback motion planning via sums-of-squares verification," *Int. J. Robotics Research*, vol. 29, no. 8, pp. 1038–1052, 2010.

[101] K. Furuta, M. Yamakita, and S. Kobayashi, "Swing up control of inverted pendulum," in *Proc. IECON'91*, 1991, pp. 2193–2198.

[102] P. A. Bhounsule, A. Ruina, and G. Stiesberg, "Discrete-decision continuous-actuation control: Balance of an inverted pendulum and pumping a pendulum swing," *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 5, p. 051 012, 2015.

[103] B. Piccoli and J. Kulkarni, "Pumping a swing by standing and squatting: Do children pump time optimally?" *IEEE Control Systems Magazine*, vol. 25, no. 4, pp. 48–56, 2005.

[104] E. Kaiser, J. N. Kutz, and S. L. Brunton, "Data-driven discovery of Koopman eigenfunctions for control," *arXiv preprint arXiv:1707.01146*, 2017.

[105] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-dynamic programming.* Athena Scientific Belmont, MA, 1996, vol. 5.

# Appendix 1

## Neural-inspired Sensors Enable Sparse, Efficient Classification of Spatiotemporal Data

This section supplements the main text, providing details of our methods as well as additional figures to support our results. We also include here a nomenclature to summarize the variables and notational conventions we are using in this paper. All of the code we developed as part of this paper are openly available as a GitHub repository as described in Sec. Appendix 1 .

First, Sec. Appendix 1 describes the flapping wing simulation with and without rotation, its implementation, and extraction of span-wise normal strain from this simulation. Raw strain computed on a dense grid on the wing is filtered by a neural encoder model. Our neural encoder model is taken directly from experimental recordings of campaniform sensilla on moth wings, and Sec. Appendix 1 briefly describes these experiments as well as how we derived a functional approximation of the neural encoding. Sec. Appendix 1 details how raw strain and neural encoded strain data is used to formulate and solve a classification task. Next, we select a small subset of sensors among the dense grid using sparsity-promoting optimization using an algorithm described in Sec. Appendix 1 , and these few sensors are able to solve the same classification task. The relationship between number of sensors and validated classifier accuracy is fit by a sigmoidal function (Sec. Appendix 1 ). Finally, Sec. Appendix 1 include 5 additional figures to supplement the results described in the main text.

### Code and Data Access

All code we developed to run the simulations and perform the analyses is available in a repository accessible at `github.com/tlmohren/Mohren_WingSparseSensors`. The code is implemented in MATLAB 2015a. To reproduce the figures, a basic MATLAB installation is sufficient. To run the simulations and solve the sparse optimization problems, our code is dependent on two toolboxes: MATLAB's symbolic toolbox and CVX (`www.cvx.com`).

### Structural Model

To simulate a flapping wing, we use an Euler-Lagrange model for a flapping flat plate and obtain strain for different prescribed inertial rotations. The model is based on Eberle et al. [35] and modified to allow additional velocity disturbances. The flat plate has a span of 50 mm, chord length of 25 mm, a thickness of 0.0127 mm, and an E-modulus of 3 GPa; these parameters are

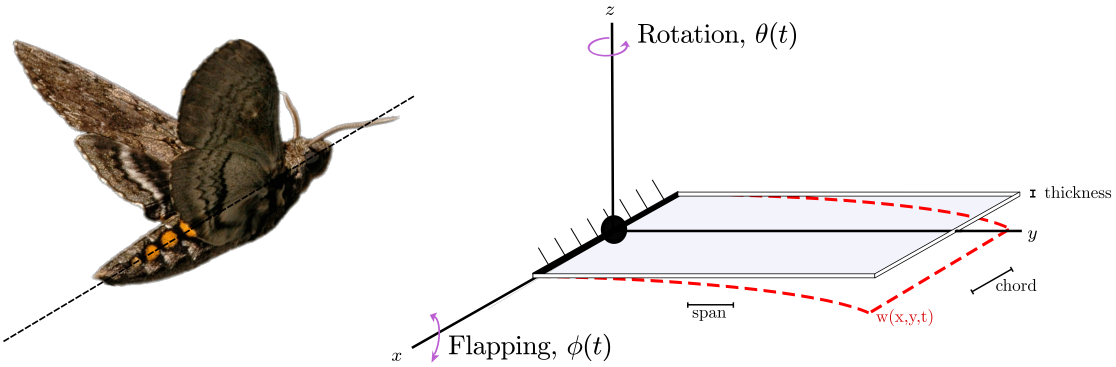

Figure Appendix 1 .1: The flapping wing is modeled by a flapping flexing plate, with parameters scaled to match a hawkmoth wing. The plate is excited by the flapping angle $\phi(t)$ around the $x$-axis, and the rotation angle $\theta(t)$ around the $z$-axis.

chosen to be consistent with previous work on hawkmoth structural wing studies [35], [48].

The simulated wing flaps with an amplitude of $\pi/6$ radians at a frequency of $f_\phi = 25$ cycles per second (Hz). A harmonic at 50 Hz is at 1/5 the magnitude of the dominant frequency. Specifically, the steady flapping is

$$\phi(t) = \frac{\pi}{6}\left(\sin(2 \cdot 10^{-3}\pi f_\phi t) + \frac{1}{5}\sin(4 \cdot 10^{-3}\pi f_\phi t)\right), \tag{A1.1}$$

where time $t$ has units of milliseconds.

In addition to flapping $\phi(t)$, the wing is perturbed by one of two different inertial rotation velocities $\dot{\theta}$:

$$\dot{\theta} = 0 \text{ rad/s (without rotation)}, \tag{A1.2}$$

$$\dot{\theta} = 10 \text{ rad/s (with rotation)}. \tag{A1.3}$$

### Physics of the flapping wing

A useful perspective on the forces present on the plate can be gained by defining a rotating reference frame $R$. We define the position of a point on the centerline of the wing at a distance $L$ from the origin as $\mathbf{r}$ and the rotation of the local frame as $\omega$:

$$\mathbf{r} = [0, L\cos(\phi), L\sin(\phi)], \tag{A1.4}$$

$$\omega = [0, 0, \dot{\theta}]. \tag{A1.5}$$

The kinematics in the local frame $R$ is related to global acceleration $I$ as

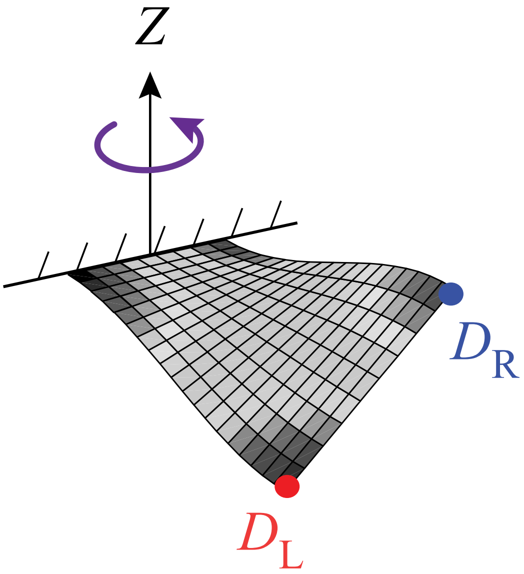

Figure Appendix 1 .2: A flat plate undergoing both flapping and rotation will undergo a twisting deformation. For the flapping flat plate, this results in an additional strain three orders smaller than strain caused by the main bending deformation mode [35].

$$\left|\frac{\partial^2 \mathbf{r}}{\partial t^2}\right|_I = \left|\frac{\partial^2 \mathbf{r}}{\partial t^2}\right|_R + \frac{\partial \omega}{\partial t} \times \mathbf{r} + 2\omega \times \frac{\partial \mathbf{r}}{\partial t} + \omega \times (\omega \times \mathbf{r}), \tag{A1.6}$$

$$\left|\frac{\partial^2 \mathbf{r}}{\partial t^2}\right|_I = \begin{bmatrix} 2L\sin(\phi)\dot{\phi}\dot{\theta} - L\cos(\phi)\ddot{\theta} \\ -L\sin(\phi)\ddot{\phi} - L\cos(\phi)(\dot{\phi}^2 + \dot{\theta}^2) \\ L\cos(\phi)\ddot{\phi} - L\sin(\phi)\dot{\phi}^2 \end{bmatrix}^T \begin{bmatrix} i \\ j \\ k \end{bmatrix}. \tag{A1.7}$$

Here the accelerations in the $i$ direction are present only for nonzero $\dot{\theta}$. Acceleration in this direction results in a twisting mode on the flat plate (Fig. Appendix 1 .2, [35]). This acceleration is generally referred to as the Coriolis acceleration. With use of the small angle approximation, it becomes clear that this acceleration occurs at twice the flapping frequency of $\phi \propto \sin(f_\phi t)$:

$$2L\sin(\phi)\dot{\phi}\dot{\theta} \approx 2L\phi\dot{\phi}\dot{\theta} \tag{A1.8}$$

$$\propto \sin(f_\phi t)\cos(f_\phi t) \tag{A1.9}$$

$$\propto \sin(2 f_\phi t). \tag{A1.10}$$

### Adding random disturbances on velocity

To simulate structurally relevant noise experienced by flapping wings, we added white noise disturbances to the steady flapping and rotation velocities. Specifically, the total flapping velocity $\dot{\phi}_T(t)$ is the sum of steady rotation velocity $\dot{\phi}(t)$ and the added disturbance $\dot{\phi}^*(t)$. Similarity, the total inertial rotational velocity $\dot{\theta}_T(t)$ is the sum of steady rotational velocity $\dot{\theta}$ and the added disturbance $\dot{\theta}^*(t)$.

We modeled disturbances as band limited white noise, summing 15 sinusoids at random frequencies between 1 and 10 Hz and at random phases:

$$\dot{\phi}^*(t) = A_{\dot{\phi}^*} \sum_{i=1}^{15} \sin(2\rho t + \kappa), \tag{A1.11}$$

$$\dot{\theta}^*(t) = A_{\dot{\theta}^*} \sum_{i=1}^{15} \sin(2\rho t + \kappa). \tag{A1.12}$$

Here the amplitude of the disturbances are $A_{\dot{\phi}^*}$ and $A_{\dot{\theta}^*}$. We chose the range of disturbance amplitudes to correspond to 0.1,1,10, and 100% of the standard deviation of steady flapping $\bar{\dot{\phi}}$ and 0.1,1,10, and 100% of the magnitude of constant rotation $\bar{\dot{\theta}}$. There are two random variables, $\rho$ and $\kappa$; $\rho$ is a random frequency drawn in the range of $[1, 10]$ Hz, and $\kappa$ is a random phase drawn in the range of $[0, 2\pi]$.

### Ramp up of flapping and rotational velocities in simulation

If the Euler-Lagrange simulations were started at the flapping and rotational velocities described above, undesirable high frequency deformation modes would be excited. To prevent these modes from contaminating the results, both flapping and rotational velocities are multiplied by a sigmoidal startup ramp $\nu$:

$$\nu = \frac{(2 \cdot 10^{-3}\pi f t)^3}{10 + (2 \cdot 10^{-3}\pi f t)^3}. \tag{A1.13}$$

Fig. Appendix 1 .3 shows the the total flapping and rotational velocities experienced by our model, including this startup ramp, with and without added disturbances.

### Implementation of model

We use rectangular shape functions that are fixed in $x$ and $y$ at the corners, but that are free to deform in $w$ at the free edge.

$$\mathbf{r}(x, y, t) = [x,\ y,\ w(x, y, t)]^T \tag{A1.14}$$

Using the shape functions, we obtain 3 degrees of freedom for the two free wing corners $\mathbf{q}$,

$$\mathbf{q}(t) = [\delta_3,\ \phi_3,\ \theta_3,\ \delta_4,\ \phi_4,\ \theta_4]^T. \tag{A1.15}$$

The displacement is then a function of the shape functions $\mathbf{N}(x, y)$ and $\mathbf{q}$:

$$w(x, y, t) = [\mathbf{N}(x, y)]^T \mathbf{q}(t). \tag{A1.16}$$

Using Lagrange's equation, we can then obtain the system of equations:

$$\frac{d^2\mathbf{q}(t)}{dt^2} = -\mathbf{M}^{-1}\mathbf{M}_a \frac{dv_0}{dt} + \left(\frac{d\mathbf{\Phi}}{dt}\right)^2 \mathbf{q}(t) - \mathbf{M}^{-1}\mathbf{K}\mathbf{q}(t) + \mathbf{M}^{-1}\mathbf{I}_c\mathbf{\Omega} - \mathbf{M}^{-1}\eta\frac{d\mathbf{q}(t)}{dt}. \tag{A1.17}$$

The system of ODE's from [35] with modified rotation angles are solved with MATLAB's ODE45 (5th order Runga-Kutta solver) for $t \in [1, 4000]$ milliseconds.

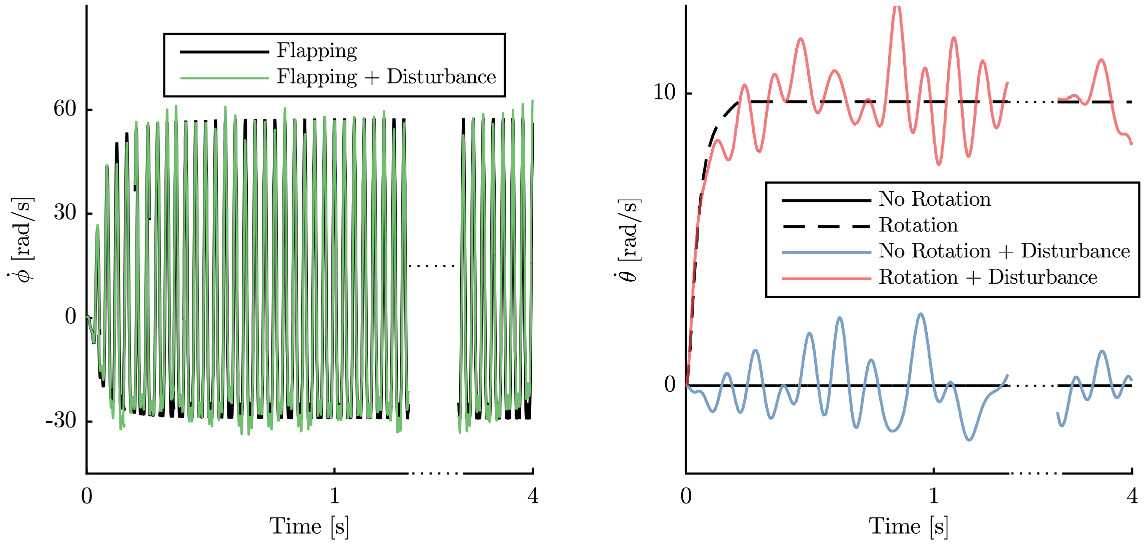

Figure Appendix 1 .3: The left graph shows the flapping velocity without disturbance (black) and with disturbance (green, standard deviation $\sigma = 3.1$ (rad/s)). The right graph shows rotational velocity. The solid black line shows a rotational velocity of zero, where the blue line shows no steady rotational velocity but a disturbance with $\sigma = 1$ (rad/s). The striped black line shows a steady rotational velocity of 10 (rad/s). The yellow has a steady rotational velocity of 10 (rad/s) with a disturbance of $\sigma = 1$ (rad/s). The simulations had a 960 millisecond transient phase that was discarded for the classification task.

### Computing strain

The spanwise strain over the wing $\epsilon_y(x, y, t)$ relates directly to the local curvature through the double partial derivative of the shape function:

$$\epsilon_y(x, y, t) = -\frac{h}{2}\frac{\partial^2 w(x, y, t)}{\partial y^2}. \tag{A1.18}$$

The chordwise strain $\epsilon_x$ is much smaller than the spanwise strain $\epsilon_y$, and there is no indication that campaniform sensilla can detect shear strain $\epsilon_{xy}$. Therefore, we will use the spanwise strain $\epsilon_y$ for our experiments and refer to it as $\epsilon$ in this paper.

Since $w(x, y, t)$ is a continuous function, we can specify our sensor locations anywhere on the wing surface. We chose to compute strain over a grid with 0.1 centimeter spacing starting at the edges. This space results in 51 spanwise and 26 chordwise points, for a total of 1326 sensor locations.

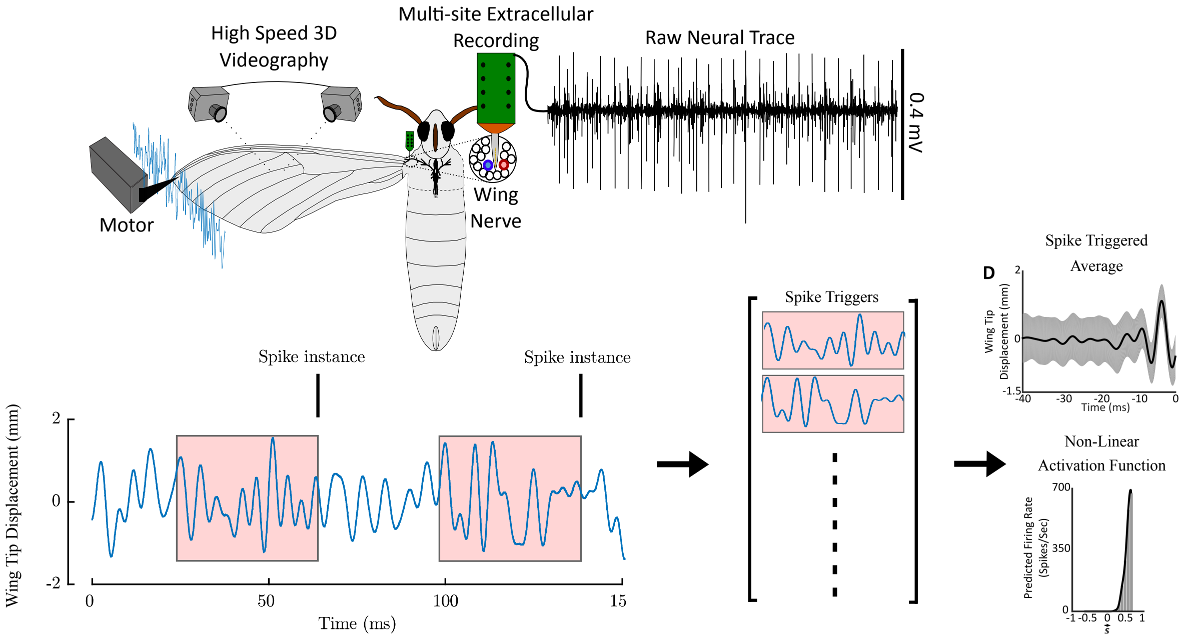

Figure Appendix 1 .4: Pratt et al. [10] characterized the neural response to strain of campaniform sensilla on hawkmoth wings by recording from the wing nerve and exciting the wing tip with band-limited white noise displacement. The strong feature selectivity of the spike triggered average and non-linear activation functions they observed were similar to observations in campaniform sensilla on the base of halteres of flies.

## Neural encoding

The action potential responses of campaniform sensilla to strain on the hawkmoth wing had been characterized by Pratt et al. [10]. Here we briefly describe these experiments and the neural encoding functions fit to the experimental data.

### Experimental electrophysiological recordings

Campaniform sensilla on a wing fire action potentials in response to the time-history of mechanical forces they experience. By recording from the wing nerve while stimulating the wing tip with a motor, one can characterize the stimulus that lead to neuronal firing. A schematic diagram of the experimental setup is shown in Fig. Appendix 1 .4, and details of the experiment are found in [10].

### Fitting STA & NLA functions

To summarize the responses of campaniform sensilla to mechanical stimulus, we compute the spike triggered average (STA) and a nonlinear activation (NLA) function to fit the experimental recordings.

The STA is approximated as a function of time $t$ before a spike at $t = 0$,

$$\mathrm{STA}(t, f_{\mathrm{STA}}, a, b) = \cos\left(f_{\mathrm{STA}}(t + a)\right) \exp\left(\frac{-(t+a)^2}{b^2}\right), \tag{A1.19}$$

where $f_{\mathrm{STA}}$ is the STA frequency, $a$ is the delay, and $b$ is the width.

The strain $\epsilon(x, y, t)$ is convolved with the STA to obtain the strain projection on this STA

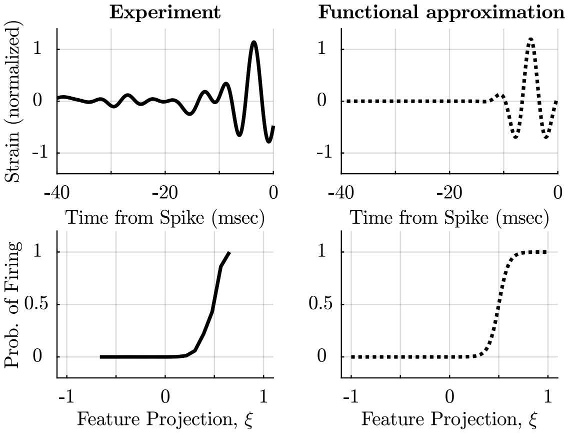

Figure Appendix 1 .5: We approximated a typical experimentally observed feature and non linear activation functions from [10] with parameterized functions according to equations A1.19 and A1.20.

feature $\xi(x, y, t)$. Next, $\xi(x, y, t)$ is mapped through a nonlinear activation function

$$\mathrm{NLA}(\xi, c, d) = \frac{1}{1 + \exp\left(-c(\xi - d)\right)}, \tag{A1.20}$$

where $c$ determines the slope and $d$ is the position of the function at half maximum. Fig. Appendix 1 .5 shows the experimental STA and NLA as well as their best functional approximations.

### A probabilistic firing model

We use the STA and NLA functions to transform the raw strain data from our structural simulation into probability of firing through a two-step process (Fig. Appendix 1 .6). First, we apply a discrete convolution to the strain with the STA to obtain $\xi$,

$$\xi(x, y, t) = \frac{1}{C_\xi} \sum_{\tau=-39}^{0} \epsilon(x, y, t - \tau) \cdot \mathrm{STA}(\tau). \tag{A1.21}$$

Second, we input $\xi$ into the NLA ((A1.22)).

$$P_{\text{fire}}(x, y, t) = \mathrm{NLA}(\xi(x, y, t)). \tag{A1.22}$$

The output is the probability of firing an action potential, which we define as the neural encoded strain. Here $C_\xi$ is a constant to normalize the probability of firing and is determined by taking the maximum non-normalized $\xi$ over all sensors. $t$ is time in milliseconds. The probability of firing over the wing over time will form our data matrix $\mathbf{X}$ in the next section.

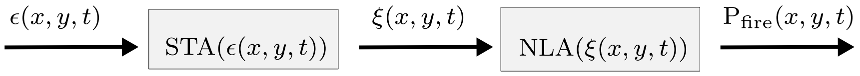

Figure Appendix 1 .6: The strain $\epsilon(x, y, t)$ is converted to probability of firing by the two step process. First we apply the discrete convolution of the strain with the STA and second, the NLA takes the resulting feature projection $\xi$, and outputs the probability of firing.

## Formulating the classification task

The strain data computed by the structural model as data to formulate a classification task. This section details how we define the training data, fit the classifiers, and assess the performance of classifiers on validation data.

### Constructing the data matrix

To construct build a classifier, we first define the training data. The data matrix $\mathbf{X}$ comprises vectorized strain data from the two classes, flapping alone and flapping with rotation. Each row in $\mathbf{X}$ is data from a single sensor. For the full sensor set, $\mathbf{X}$ has 1326 rows.

$$\mathbf{X} = \begin{bmatrix} \vdots & & \vdots & \vdots & & \vdots \\ \mathbf{x_1^F} & \dots & \mathbf{x_k^F} & \mathbf{x_1^R} & \dots & \mathbf{x_k^R} \\ \underbrace{\vdots \qquad\qquad \vdots}_{\text{Flapping}} & & & \underbrace{\vdots \qquad\qquad \vdots}_{\text{With rotation}} & & \end{bmatrix}, \tag{A1.23}$$

where $\mathbf{x^F}$ is strain data from flapping alone, $\mathbf{x^R}$ is strain data from flapping with rotation, and $k$ is the time index.

### Training and validation

The training data is made from the first 90% of snapshots for each class, which are assembled into $\mathbf{X}_{\text{train}}$. The last 10% of snapshots of data, taken from an epoch of the simulation after the training data, make $\mathbf{X}_{\text{test}}$. We use $\mathbf{X}_{\text{train}}$ to fit the classifier, which will be used to predict the class for each of the snapshots in $\mathbf{X}_{\text{test}}$. This construction is shown schematically in Fig. Appendix 1 .7.

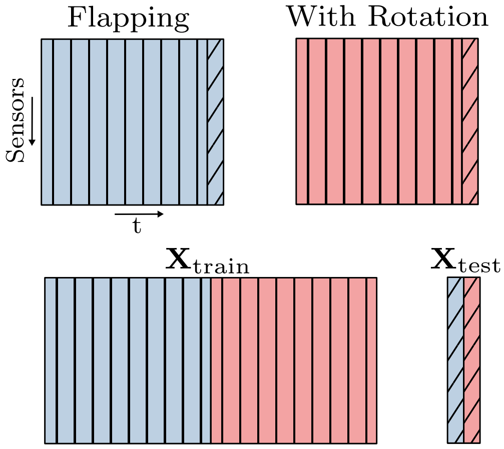

Figure Appendix 1 .7: Construction of the training and validation data. The first 90% of data snapshots of each class is assembled into $\mathbf{X}_{\text{train}}$, the last 10% are assembled into $\mathbf{X}_{\text{test}}$.

### Building linear classifiers

To classify the data, we use linear discriminant analysis (LDA) on the training data. LDA computes a vector $\mathbf{w}$, where the data projected onto $\mathbf{w}$ is linearly separable. In particular, we solve for

$$\mathbf{w} = \arg\max_{\mathbf{w}'} \frac{\mathbf{w}'^T \mathbf{S_B} \mathbf{w}'}{\mathbf{w}'^T \mathbf{S_W} \mathbf{w}'}, \tag{A1.24}$$

where $\mathbf{S_B}$ and $\mathbf{S_W}$ are the between-class and within-class scatter matrices, respectively. They are computed from the training data as follows:

$$\mathbf{S_W} = \sum_{j=1}^{c} \sum_{i \in c_j} (\mathbf{X}_{\text{train},i} - \mu_j)(\mathbf{X}_{\text{train},i} - \mu_j)^T, \tag{A1.25}$$

$$\mathbf{S_B} = \sum_{j=1}^{c} N_j (\mu_j - \mu)(\mu_j - \mu)^T. \tag{A1.26}$$

$$\tag{A1.27}$$

Here $c$ is the total number of classes, $c_j$ are all the observations in the $j^{th}$ class, $\mu_j$ is the centroid of class $j$, and $\mu$ is the centroid of all the training data.

To solve (A1.24), we solve for the eigendecomposition of $\mathbf{S_W}^{-1}\mathbf{S_B}$, and $\mathbf{w}$ is the eigenvector corresponding to the largest eigenvalue:

$$\mathbf{S_W}^{-1} \mathbf{S_B} \mathbf{w} = \lambda \mathbf{w}. \tag{A1.28}$$

Note that the above linear classifier can be trained if the number of examples in the training set is at least $q + c$ to guarantee that $\mathbf{S_W}$ is not singular, where $q$ is the number of sensors and $c$ is the number of classes to be classified.

Next, the data is projected onto $\mathbf{w}$

$$\eta = \mathbf{w}^T \mathbf{X}_{\text{train}}, \tag{A1.29}$$

and we apply a threshold in $\eta$ that separates the two classes.

For each class, we determine the mean $\mu_j$ and standard deviation $\sigma_j$. Assuming a gaussian distribution, we solve for where the two Gaussian probability density functions (A1.30) intersect in between the two means (Fig. Appendix 1 .8). If there is no intersection in between $\mu_1$ and $\mu_2$, we take the threshold to be the geometric middle $\frac{\mu_1+\mu_2}{2}$. The Gaussian probability density function is given by:

$$G_j = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(x - \mu_j)^2}{2\sigma_j^2}\right). \tag{A1.30}$$

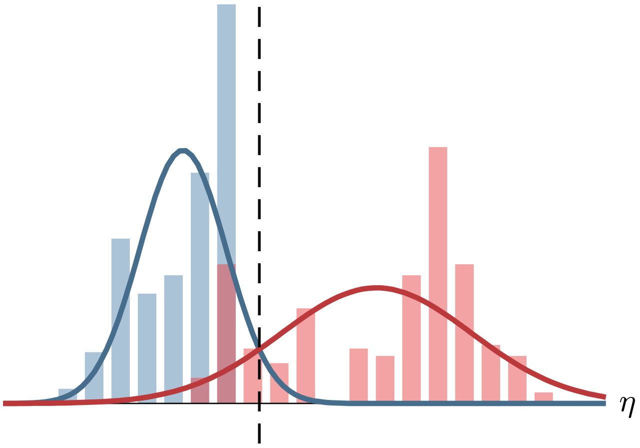

Figure Appendix 1 .8: The threshold between the two classes is determined by the intersection of the Gaussian probability density functions belonging to each class.

### Classifier validation

After training classifier by fitting $\mathbf{w}$, this classifier is applied to and assessed on the validation data $\mathbf{X}_{\text{test}}$,

$$\eta_{\text{test}} = \mathbf{w}^T \mathbf{X}_{\text{test}}, \tag{A1.31}$$

and the same threshold fit to the training data is applied to assign each test sample to a category. The validated accuracy is computed by comparing these categories to the known classes from the simulation.

### Comparison with common nonlinear classifiers

To show that nonlinear filtering in time, as inspired by neural encoding, is crucial for classification, we compare the validated classification accuracy of our PCA-LDA approach to a variety of common nonlinear classifiers. In particular, we train the following algorithms on the classification task: support vector machine (SVM) with different nonlinear kernels, decision trees, and neural networks. The neural network is implemented as a multi-layer perceptron (MLP) with a rectified linear unit (ReLU) nonlinearity. All of these classifiers are fit using the python *sklearn* library. The results are shown in table Appendix 1 .1. The implementation of the comparison can be found in a Jupyter notebook file in our code repository. While some of these nonlinear classifiers perform better than chance, none of them approach the accuracy obtained with neural encoded strain (0.9, see Fig. 3)

## Sparse sensor placement optimization

This section describes our approach to learn a small handful of sparsely placed sensors that perform the body rotation classification. We summarize the sparse sensor placement for optimal classification (SSPOC, [44]). Next, we describe two ways the sparse optimization we used in this paper have been extended from [44]. First, we truncate selective singular vector features based on the discriminant vector $\mathbf{w}$. And second, we use an elastic net penalty instead of a $\ell_1$ penalty.

### SSPOC

We take advantage of the observation that the high-dimensional data $\mathbf{x} \in \mathbb{R}^n$ may have a *low-rank* representation:

$$\mathbf{x} = \mathbf{\Psi}_r \mathbf{a}, \qquad \mathbf{a} \in \mathbb{R}^r. \tag{A1.32}$$

The goal of sparse sensor selection is to design a measurement matrix $\mathbf{C} \in \mathbb{R}^{q \times n}$ with a very small number of optimized measurements ($q \ll n$):

$$\mathbf{y} = \mathbf{C}\mathbf{x} = \mathbf{C}\mathbf{\Psi}_r \mathbf{a}. \tag{A1.33}$$

Further, we solve for $\mathbf{C}$ consisting of rows of the identity matrix, so that each sensor is a point measurement.

SSPOC [44] is a sensor selection approach to find a solution $\mathbf{C}$ so that the linear discrimination between classes is achievable with the sparse measurements $\mathbf{y}$. In briefly, we first reduce the dimensionality of $\mathbf{X}$ using the singular value decomposition (SVD):

$$\mathbf{X} = \mathbf{\Psi}\mathbf{\Sigma}\mathbf{V}^T \approx \mathbf{\Psi_r}\mathbf{\Sigma_r}\mathbf{V_r}^T, \tag{A1.34}$$

where we take advantage of order of the singular values to truncate $\mathbf{\Psi}$, $\mathbf{\Sigma}$, and $\mathbf{V}$ to their first $r$ features. The data in $\mathbf{X}$ may be projected to $\mathbb{R}^r$ using $\mathbf{\Psi_r}$,

$$\mathbf{a} = \mathbf{\Psi_r}^T \mathbf{X}. \tag{A1.35}$$

Next, we use LDA to solve for the discriminant vector $\mathbf{w}$ using $\mathbf{a}$ as the training data, so that the discriminant threshold is applied in $\eta$:

$$\eta = \mathbf{w}^T \mathbf{a} = \mathbf{w}^T \mathbf{\Psi_r}^T \mathbf{X}. \tag{A1.36}$$

Finally, we solve for the sparse vector $\mathbf{s} \in \mathbb{R}^n$:

$$\mathbf{s} = \underset{\mathbf{s}'}{\operatorname{argmin}} \left\| \mathbf{s}' \right\|_1, \qquad \text{subject to} \ \ \mathbf{\Psi_r}^T \mathbf{s}' = \mathbf{w}, \tag{A1.37}$$

where $\mathbf{s}$ comprises mostly zeros, and the non-zero entries of $\mathbf{s}$ correspond to sensor locations and rows of the identity matrix selected for the measurement matrix $\mathbf{C}$.

Generally speaking, the number of sensors $q$ selected by this approach is approximately $r$, so the choice of $r$ determines the number of sensors desired.

### Singular value feature selection for SSPOC

In the previous section, we described truncating the SVD basis $\mathbf{\Psi}$ to its first $r$ columns, corresponding to the $r$ largest singular values. However, these first $r$ features may not necessarily be the ones supporting the largest separation between classes. Here, we use an alternative criterion to select which columns of $\mathbf{\Psi}$ are used in the sparse optimization by re-weighing each according to the LDA discriminant vector.

Specifically, the singular values are re-weighted according to the magnitude of $\mathbf{w}$, and the $\rho$ largest entries of $\mathbf{\Sigma_r}|\mathbf{w}|$ determine the column of $\mathbf{\Psi}$ that form a new truncated basis $\mathbf{\Psi}_\rho$. It follows that we solve (A1.37) using $\mathbf{\Psi}_\rho$ and $\mathbf{w}_\rho$, which produces approximately $\rho$ sensors.

### Sparse optimization with an elastic net penalty

We observed that solutions to (A1.37) using convex optimization tools sometimes do not converge to optimal solution. Therefore, in this paper we use a related optimization using an elastic net penalty, which balances the ratio of penalty for the $\ell_1$ and $\ell_2$ norms of $\mathbf{s}$:

$$\mathbf{s} = \underset{\mathbf{s}'}{\operatorname{argmin}}\ \alpha||\mathbf{s}'||_1 + (1-\alpha)||\mathbf{s}'||_2,\ \text{subject to } \mathbf{\Psi}_\rho^{\mathbf{T}}\mathbf{s}' = \mathbf{w}_\rho. \tag{A1.38}$$

For this paper, we use $\alpha = 0.9$.

## Sigmoidal fit to classification accuracy

We observed that the validated accuracy $A$ depends on the number of sensors $q$ in a sigmoidal relationship, so we fit the results shown in Figs. 3 and 4 of the main paper with a sigmoidal function with 3 parameters:

$$A(q) = \frac{\frac{1}{2} + c_1}{1 + \exp(-\frac{q-c_2}{c_3})}. \tag{A1.39}$$

To summarize these curves over different neural encoders, the results in Fig. 5 of the main paper present the fewest number of sensors required for classification at 75% accuracy. This number is determined by solving for $q$ at which $A(q)$ crosses 0.75.

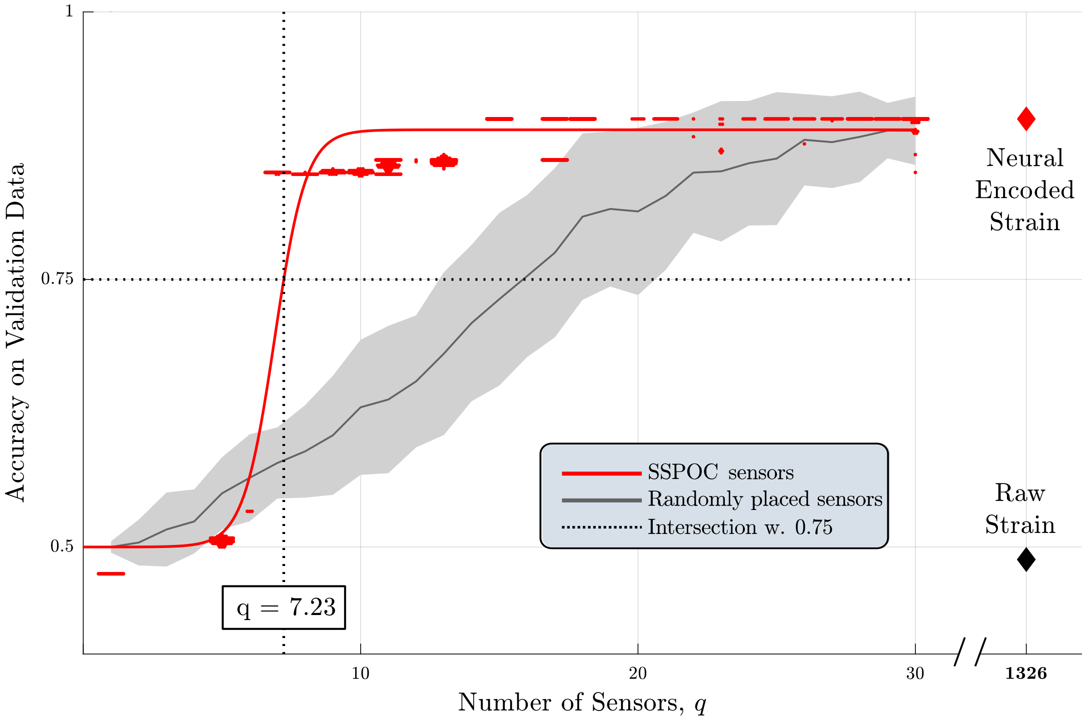

Figure Appendix 1 .9: We determined the number of sensors required for good classification by fitting a sigmoid (eq. A1.39) to the classification accuracy versus the number of sensors. The dashed line shows the sigmoidal fit with constants $[c_1, c_2, c_3] = [0.378, 6.904, 0.583]$, intersecting with the 0.75 accuracy line at q = 7.29.

## Supplementary results figures

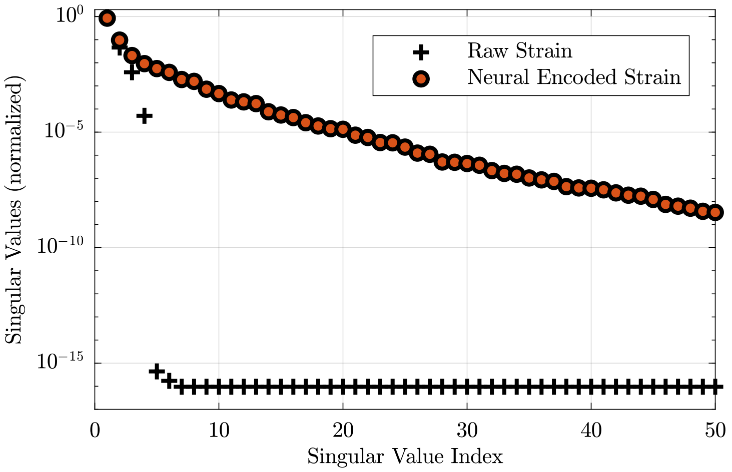

Figure Appendix 1 .10: The normalized singular values for raw strain (black plus) and neural encoded strain (red circle).

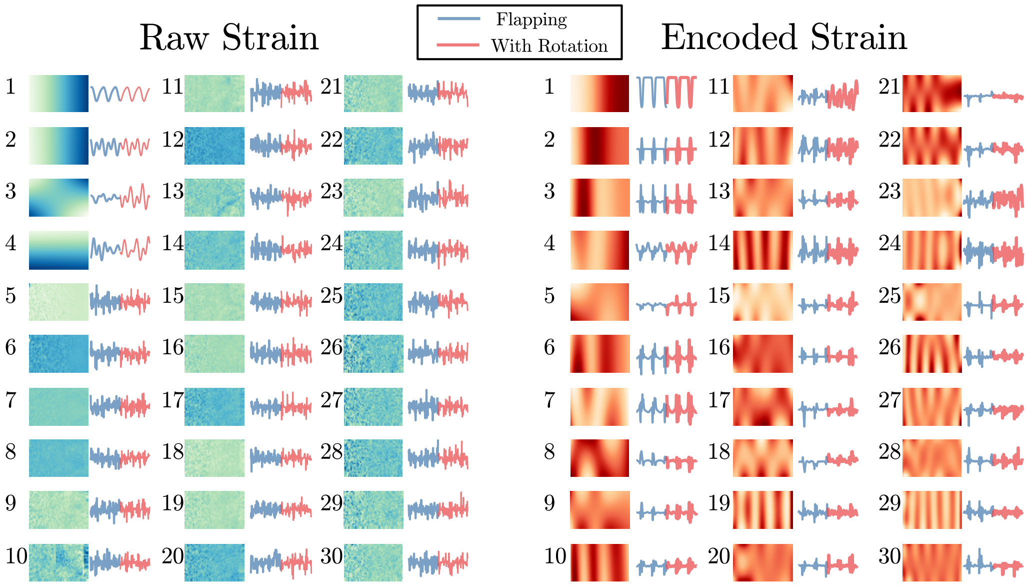

Figure Appendix 1 .11: Singular Value Decomposition modes for raw strain(left) and encoded strain (right). The plate shows the mode shape and it's associated number indicates the Singular Value Index. The blue signal shows the presence of that mode when the wing is flapping, the red signal is the presence of that mode for flapping with rotation.

Table Appendix 1 .1: Comparison of validated classification accuracy of various classifiers on full grid of raw strain measurements, averaged over 10 instances of moderate disturbance amplitudes of $\dot{\phi}^*, \dot{\theta}^* = [0.31, 0.1]$ rad/s. The nonlinear classifiers are described in more detail in Sec. Appendix 1 . While some of these nonlinear classifiers perform better than chance, none of them consistently approach the accuracy obtained with neural encoded strain (0.90, see Fig. 3).

| Classifier | Accuracy, mean ± std |
|---|---|
| PCA-LDA | $0.50 \pm 0.0045$ |
| SVM, linear kernel | $0.610 \pm 0.0$ |
| SVM, sigmoid kernel | $0.50 \pm 0.0039$ |
| SVM, quadratic kernel | $0.56 \pm 0.0$ |
| SVM, RBF kernel | $0.59 \pm 0.024$ |
| Decision Tree | $0.63 \pm 0.16$ |
| Neural Net, MLP | $0.77 \pm 0.15$ |

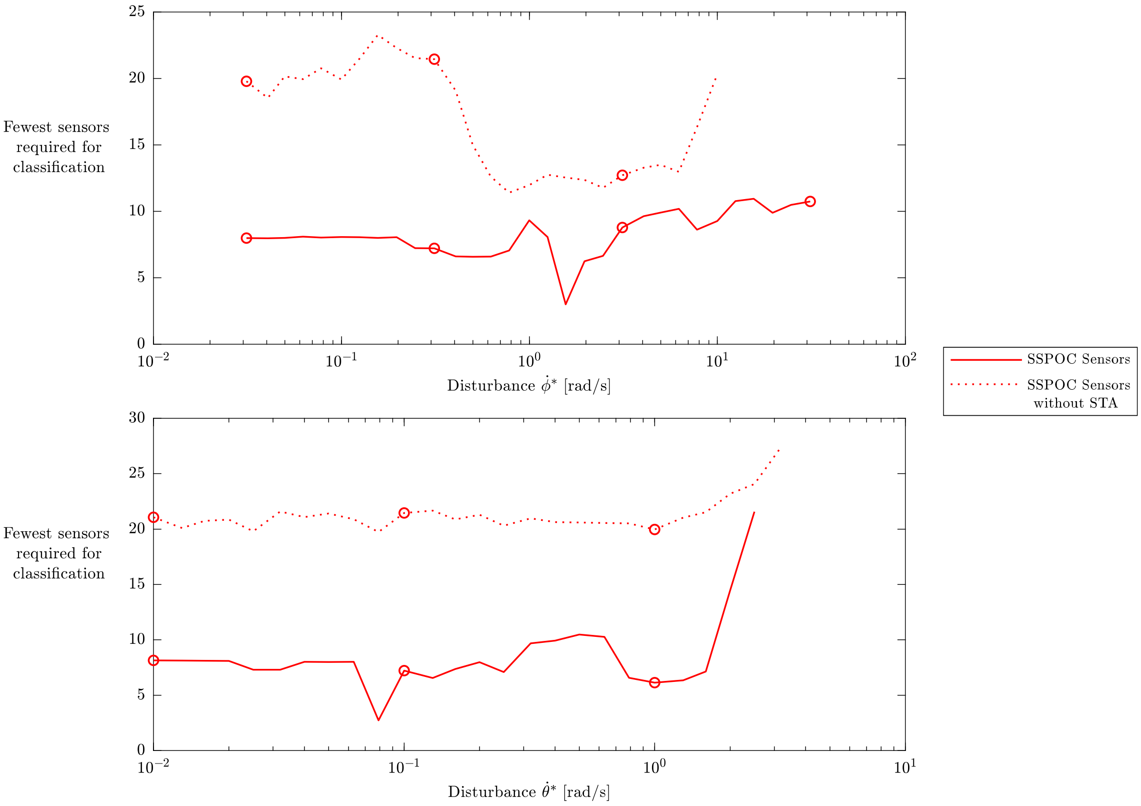

Figure Appendix 1 .12: The top plot shows a more detailed plot of the number of sensors, q, required for 75% accuracy versus an increase in flapping disturbance $\dot{\phi}^*$, with $\dot{\theta}^*$ constant at 0.1. The dotted line shows classification accuracy without STA. The circle represent the number of sensors for the plots that were shown in figure 4 of the main paper, corresponding to the red bar in the matrix figure on the top right. The bottom plot shows a detailed plot for constant $\dot{\phi}^* = 0.31$ and varying $\dot{\theta}^*$.

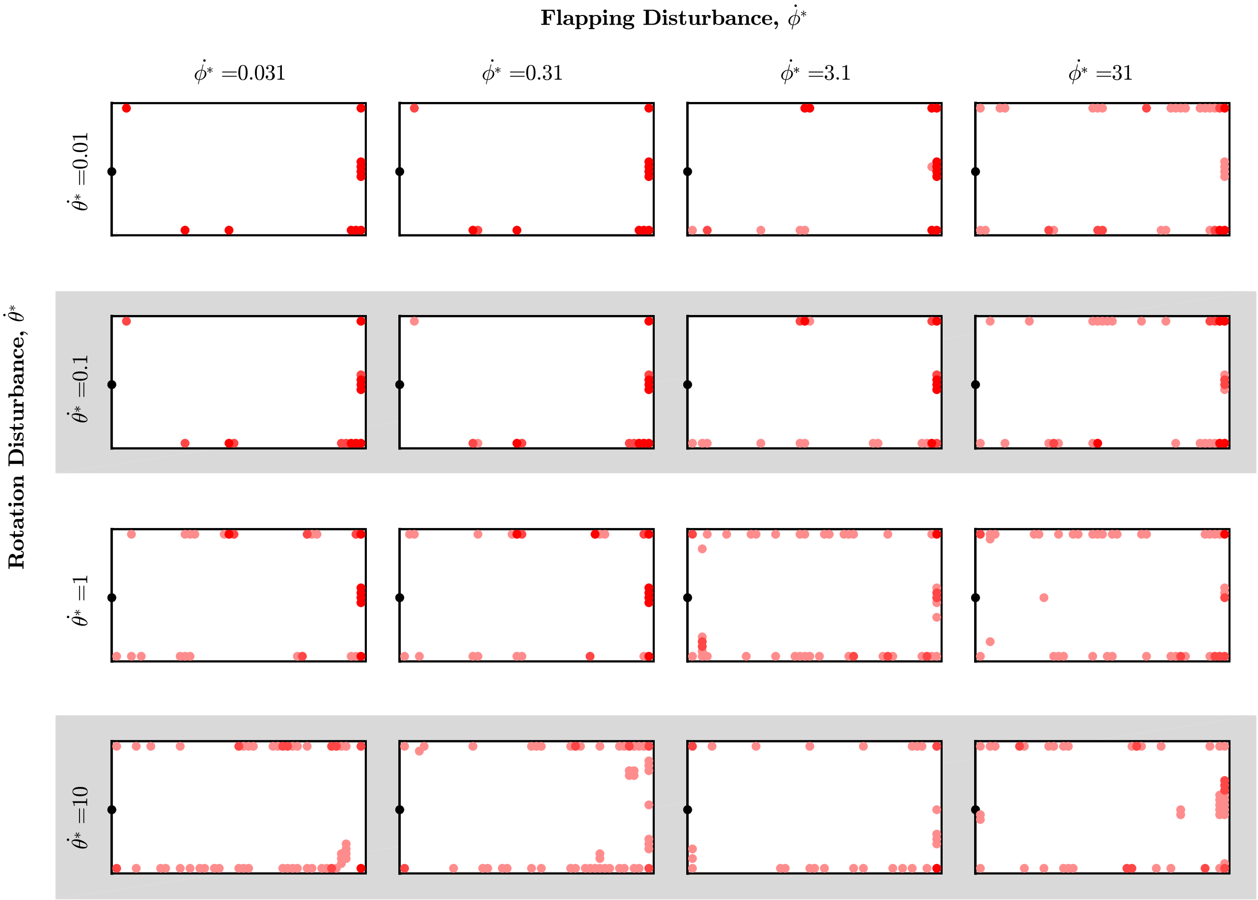

Figure Appendix 1 .13: The sensor locations for $q = 11$ under different disturbance level combinations, darker red indicating a higher probability of sensor placement.

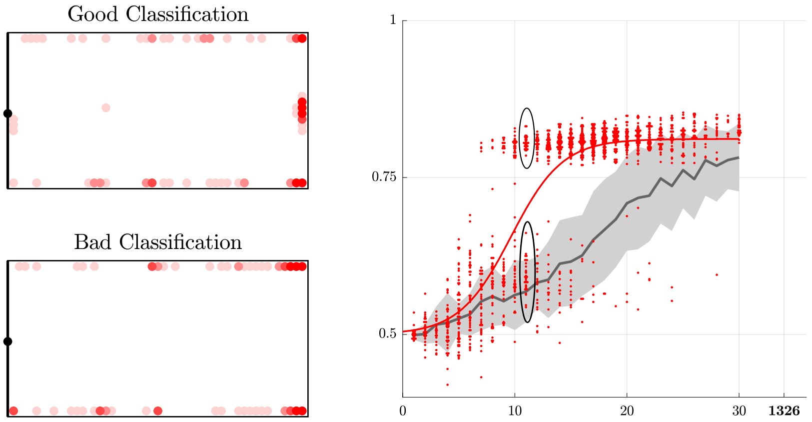

Figure Appendix 1 .14: The sensor placement for $q = 11$ under disturbance conditions $\dot{\phi}^* = 31$ $\dot{\theta}^* = 1$. The right figure shows the bimodal distribution of the classification accuracy. The sensor locations are shown for the succesful classification group (top) and the poor classification group (bottom), with darker red indicating a higher probability of sensor placement.

# Appendix 2

## Coriolis and Centrifugal Forces Drive Haltere Deformations and Influence Spike Timing

### Polar moment of area for cylindrical and plus-shaped stalk cross-sections

For the two beams in figure 2.2, we calculate the expected reduction in polar moment of area $J$ while keeping the second moment of area $I$ identical. For a circular shaped (CS) cross-section shaft we know the second moment of area $I_x$ and the second polar moment of inertia $J$ to be:

$$I_{\mathrm{CS}} = \frac{\pi}{4}(r_2^4 - r_1^4), \tag{A2.1}$$

$$J_{\mathrm{CS}} = \frac{\pi}{2}(r_2^4 - r_1^4), \tag{A2.2}$$

where $r_2$ is the outer radius and $r_1$ is the inner radius of the hollow cylinder.

To find $I$ and $J$ for the plus-sign shaped cross-section (PS), we can approximate the plus by two orthogonal rectangles. For rectangles we have the following cross-sectional properties:

$$I_{\mathrm{rectangle}} = \frac{1}{12}bh^3, \tag{A2.3}$$

$$J_{\mathrm{rectangle}} \approx \beta hb^3, \quad \text{with } \beta = 0.312 \text{ for } A = \frac{h}{b} = 10. \tag{A2.4}$$

Here $b$ is the base width of the rectangle, $h$ is the height, and $\beta$ is a parameter that depends on aspect ratio $A$ and goes to 1/3 as $A$ approaches infinity.

We can nowapproximate the second moment of area and the polar moment of inertia of the plus-sign shaped cross-section (PS):

$$I_{\mathrm{PS}} = \frac{1}{12}\left(bh^3 + (h-2b)b^3\right) \approx \frac{1}{12}bh^3 \tag{A2.5}$$

$$J_{\mathrm{PS}} \approx 2\beta hb^3 \tag{A2.6}$$

We now set the second moment of area to be equal for the two different cross-sections:

$$I_{\mathrm{CS}} = I_{\mathrm{PS}}, \tag{A2.7}$$

$$\frac{\pi}{4}(r_2^4 - r_1^4) \approx \frac{1}{12} b h^3, \tag{A2.8}$$

$$\frac{\pi}{4}(r_2^4 - r_1^4) \approx \frac{1}{12} b^4 A^3. \tag{A2.9}$$

We can then define the ratio of the polar moment of area of the CS over the polar moment of inertia of the PS:

$$\frac{J_{\mathrm{CS}}}{J_{\mathrm{PS}}} \approx \frac{\frac{\pi}{2}(r_2^4 - r_1^4)}{2(\beta A b^4)}, \tag{A2.10}$$

$$\approx \frac{\frac{\pi}{4}(r_2^4 - r_1^4)}{\beta A b^4}, \tag{A2.11}$$

$$\approx \frac{\frac{1}{12} b^4 A^3}{\beta A b^4}, \tag{A2.12}$$

$$\boxed{\frac{J_{\mathrm{CS}}}{J_{\mathrm{PS}}} \approx \frac{A^2}{12\beta}.} \tag{A2.13}$$

For the simulations in figure 2.2 we set the aspect ratio to 12. Given this aspect ratio, we would expect the torsional rigidity to be reduced by an approximate factor of $\frac{1}{38}$.

## Proportionality and frequency analysis of haltere accelerations

The forces on the stalk enacted by the bulb in it's specified trajectory can be described by the following equation:

$$F_{\text{stalk,x'y'z'}} = -m_{\text{bulb}} a_{\text{bulb, x'y'z'}}. \tag{A2.14}$$

We can then derive the accelerations on the bulb using the equation for a rotating reference frame:

$$a_{\mathrm{xyz}} = \frac{\partial^2 \mathbf{r}}{\partial t^2} + \underbrace{\frac{\partial \boldsymbol{\omega}}{\partial t} \times \mathbf{r}}_{\text{Euler}} + \underbrace{2\boldsymbol{\omega} \times \frac{\partial \mathbf{r}}{\partial t}}_{\text{Coriolis}} + \underbrace{\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r})}_{\text{centrifugal}}, \tag{A2.15}$$

where $a_{\mathrm{xyz}}$ are the accelerations in the rotating (insect) reference frame.

In the case of the flapping haltere of radius $R$ in the reference frame of the rotating insect, we can define position $\mathbf{r}$ and rotation $\boldsymbol{\omega}$ as:

$$\mathbf{r} = R \begin{bmatrix} \cos\phi \\ 0 \\ \sin\phi \end{bmatrix}, \boldsymbol{\omega} = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta} \end{bmatrix}. \tag{A2.16}$$

When we insert this into equation A2.15, we find the accelerations in the moving reference frame:

$$a_{\mathrm{xyz}} = R \begin{bmatrix} -\ddot{\phi}\sin\phi - \dot{\phi}^2\cos\phi \\ 0 \\ \ddot{\phi}\cos\phi - \dot{\phi}^2\sin\phi \end{bmatrix} + R \begin{bmatrix} 0 \\ \ddot{\theta}\cos\phi \\ 0 \end{bmatrix} + R \begin{bmatrix} 0 \\ -2\dot{\theta}\dot{\phi}\sin\phi \\ 0 \end{bmatrix} + R \begin{bmatrix} -\dot{\theta}^2\cos\phi \\ 0 \\ 0 \end{bmatrix}, \tag{A2.17}$$

$$a_{\mathrm{xyz}} = R \begin{bmatrix} -\ddot{\phi}\sin\phi - \cos\phi(\dot{\phi}^2 + \dot{\theta}^2) \\ \ddot{\theta}\cos\phi - 2\dot{\theta}\dot{\phi}\sin\phi \\ \ddot{\phi}\cos\phi - \dot{\phi}^2\sin\phi \end{bmatrix}. \tag{A2.18}$$

For our purposes it's most useful to transform the accelerations to the local frame that moves with the haltere $(x'y'z')$ :

$$\begin{bmatrix} e_{x'} \\ e_{y'} \\ e_{z'} \end{bmatrix} = \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix} \begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix}. \tag{A2.19}$$

We now obtain accelerations in the local haltere frame:

$$a_{\mathrm{x'y'z'}} = \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix} R \begin{bmatrix} -\ddot{\phi}\sin\phi - \cos\phi(\dot{\phi}^2 + \dot{\theta}^2) \\ \ddot{\theta}\cos\phi - 2\dot{\theta}\dot{\phi}\sin\phi \\ \ddot{\phi}\cos\phi - \dot{\phi}^2\sin\phi \end{bmatrix}, \tag{A2.20}$$

$$a_{\mathrm{x'y'z'}} = R \begin{bmatrix} -\cos^2\phi(\dot{\theta}^2) - \dot{\phi}^2 \\ \ddot{\theta}\cos\phi - 2\dot{\theta}\dot{\phi}\sin\phi \\ \ddot{\phi} + \cos\phi\sin\phi\dot{\theta}^2 \end{bmatrix}. \tag{A2.21}$$

We can now insert them into equation A2.14 and separate out the Euler, Coriolis and centrifugal forces:

$$F_{\mathrm{stalk,\ x'y'z'}} = m_{\mathrm{bulb}} R \left( \underbrace{\begin{bmatrix} 0 \\ -\ddot{\theta}\cos\phi \\ -\ddot{\phi} \end{bmatrix}}_{\text{Euler force}} + \underbrace{\begin{bmatrix} 0 \\ 2\dot{\theta}\dot{\phi}\sin\phi \\ 0 \end{bmatrix}}_{\text{Coriolis force}} + \underbrace{\begin{bmatrix} (\dot{\phi}^2 + \dot{\theta}^2\cos^2\phi) \\ 0 \\ -\dot{\theta}^2\cos\phi\sin\phi \end{bmatrix}}_{\text{centrifugal force}} \right). \tag{A2.22}$$

We prescribed $\phi$ to be sinusoidal, and for convenience we define a scaled time parameter $\tilde{t}$:

$$\phi = \frac{\pi}{2}\sin(\tilde{t}), \quad \text{where } \tilde{t} = 2\pi f_\phi t. \tag{A2.23}$$

If we now insert $\phi$ into equation A2.22, we obtain terms like $\cos\phi = \cos(\sin(\tilde{t}))$. These do not provide direct intuition to the frequency of the signals, but we can deconstruct them into their dominant fourier series coefficients, with $g(k\tilde{t})$ corresponding to frequency terms of $k$ and higher.

$$F_{\text{Euler}} = m_{\text{bulb}} R \begin{bmatrix} 0 \\ -\ddot{\theta}\cos\phi \\ -\ddot{\phi} \end{bmatrix}, \tag{A2.24}$$

$$\approx m_{\text{bulb}} R \begin{bmatrix} 0 \\ -\ddot{\theta}\left[0.5 + 0.5\cos(2\tilde{t}) + g(4\tilde{t})\right] \\ -2\pi^3 f_\phi^2 \cos(\tilde{t}) \end{bmatrix}, \tag{A2.25}$$

$$F_{\text{Coriolis}} = m_{\text{bulb}} R \begin{bmatrix} 0 \\ 2\dot{\theta}\dot{\phi}\sin\phi \\ 0 \end{bmatrix}, \tag{A2.26}$$

$$\approx m_{\text{bulb}} R \begin{bmatrix} 0 \\ 2\dot{\theta}\pi^2 f_\phi \left[0.64\sin(2\tilde{t}) + g(4\tilde{t})\right] \\ 0 \end{bmatrix}, \tag{A2.27}$$

$$F_{\text{centrifugal}} = m_{\text{bulb}} R \begin{bmatrix} (\dot{\phi}^2 + \dot{\theta}^2\cos^2\phi) \\ 0 \\ -\dot{\theta}^2\cos\phi\sin\phi \end{bmatrix}, \tag{A2.28}$$

$$\approx m_{\text{bulb}} R \begin{bmatrix} \pi^4 f_\phi^2 \left[0.5 + 0.5\cos(2\tilde{t}))\right] - \dot{\theta}^2\left[0.35 + 0.5\cos(2\tilde{t}) + g(4\tilde{t})\right] \\ 0 \\ -\dot{\theta}^2\left[0.3\sin(\tilde{t}) + .3\sin(3\tilde{t}) + g(5\tilde{t})\right] \end{bmatrix}. \tag{A2.29}$$

In the $x'$ direction (along the span of the haltere) we observe that the centrifugal force scales with $f_\phi^2$ and occurs at frequency of $2f_\phi$ due to the flapping motion. In our simulations we see the $2f_\phi$ component in centrifugally excited out-of-plane bending of the horizontally asymetric bulb (fig. 2.3).

In the $y'$ direction (orthogonal to the plane of flapping), we see Coriolis force depends linearly on both $f_\phi$ and $\dot{\theta}$. It is composed of a dominant frequency component of $2f_\phi$ and lower amplitude harmonics. The Euler force in $y'$ is only observed during the startup of our simulations.

In the $z'$ direction we see a dominant Euler force due to the flapping motion, scaling with $f_\phi^2$.

## Neural encoder

The STA and NLD are approximated by the following function forms;

$$\mathrm{STA} = -\cos\left(f_{\mathrm{STA}}(t+a)\right)\exp\left(\frac{-(t+a)^2}{b^2}\right), \tag{A2.30}$$

where $f_{\mathrm{STA}}$ is the frequency of the waveform, $a$ is the time of the wave peak to the time of spike, $b$ is the width of the waveform, and

$$\mathrm{NLD} = \frac{1}{1+\exp\left(-c(\xi-d)\right)}, \tag{A2.31}$$

where $\xi$ is the normalized projection of the STA on the stimulus, $c$ is the gradient and $d$ is the slope at half maximum.

## Model geometry

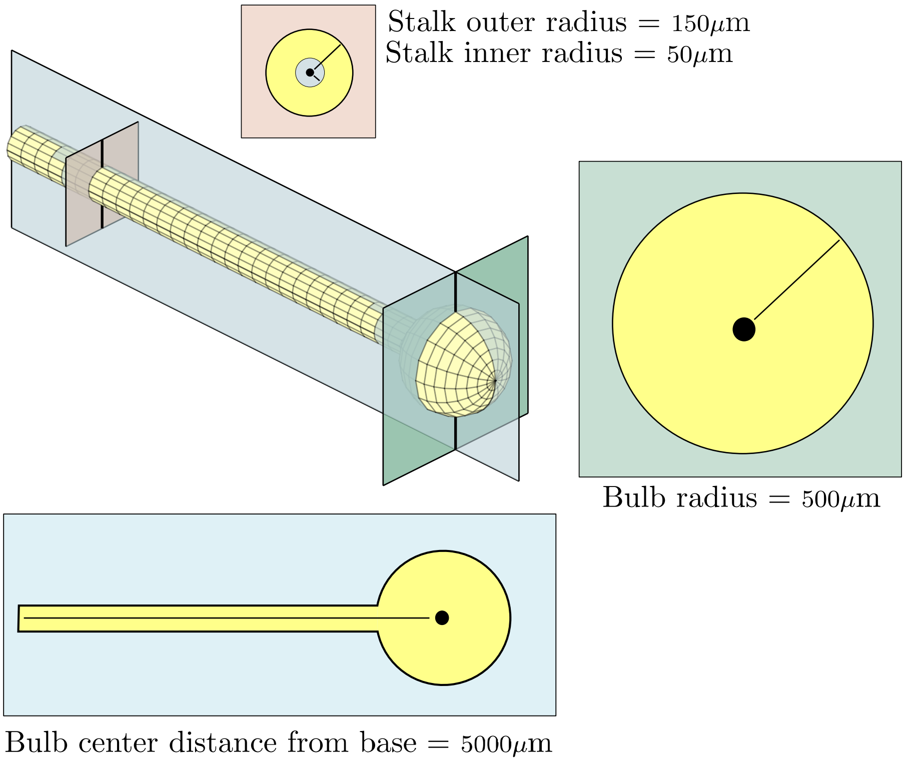

Figure S1: The basic haltere geometry, showing the cross-sections of the stalk and bulb.

# Mesh convergence

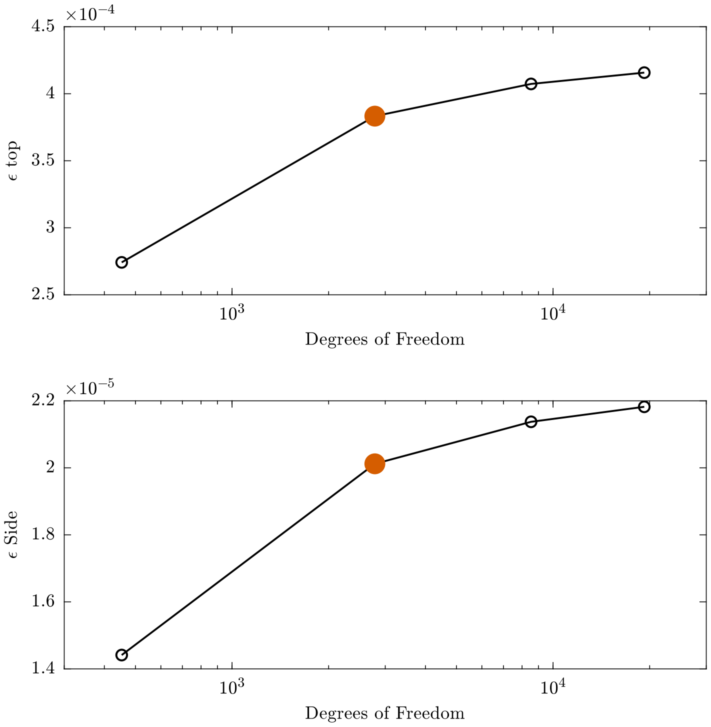

Figure S2: The strain on the top and left side note of the haltere at 300 $\mu$m from the base of the haltere, plotted against degrees of freedom in the model. Our model had 2785 degrees of freedom, indicated in red.

# Damping convergence

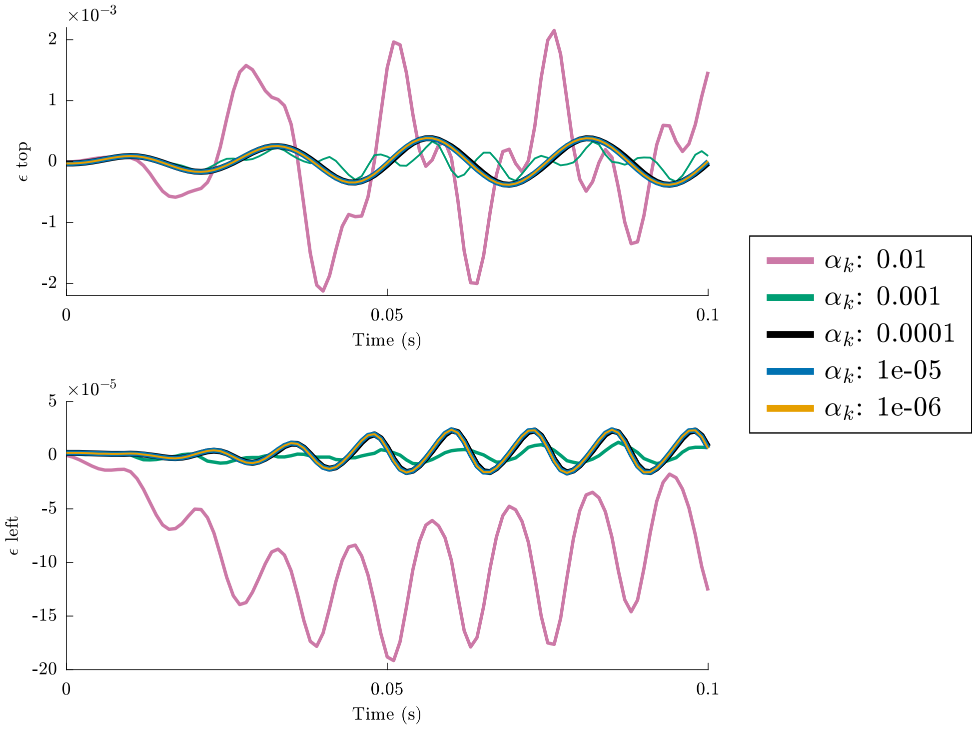

Figure S3: The strain on the top and left side note of the haltere at 300 $\mu$m from the base of the haltere for different values of damping ratio.

## Code and data accessibility

The simulation data from the computational models is available at Zenodo, `https://doi.org/10.5281/zenodo.2542944`. We used MATLAB [69], to analyze the data. Our code to analyze the finite element results and to implement the neural encoding can be found at `https://github.com/tlmohren/Haltere-code`.

## Competing interests

The authors have no competing interests

## Authors contributions

TLM and ALE created the finite element models. JLF was responsible for the measurements and analysis of neural encoding data. TLD, TLM, JLF drafted the bulk of the manuscript. TLM provided all the final derivations in the appendix. PGR and TLD coordinated the study. All authors gave final approval of the manuscript.

## Acknowledgements

The authors thank Dr. Sweta Agrawal for providing original images of halteres (fig. 1) and for critical reading of the manuscript. We are also grateful for comments and suggestions from Dr. Tanvi Deora and Brandon Pratt.

**Funding**

This work was supported by AFOSR grant FA9550-11-1-0155 and FA9550-14-1-0398, the Komen Endowed Chair and funds from the Washington Research Foundation to TLD. ALE also received funding from NSF GRFP (DGE-0718124).

# Appendix 3

## Learning Precisely Timed Feedforward Control of the Sensor-Denied Inverted Pendulum

## Additional derivations

Because there is no damping and no disturbance in the system, the pendulum's total energy is constant without forcing. The resulting energy levels form orbits in state space that the pendulum will follow. The most notable is the homoclinic orbit, which has the same total energy as the inverted equilibrium. When the pendulum is on this orbit, it will move into the inverted equilibrium over time. The equation for the homoclinic orbit is:

$$\dot{\theta} = \pm\sqrt{\frac{-2g}{l}}\sqrt{1+\cos\theta}\ \text{, with derivative:} \tag{A3.1}$$

$$\frac{\partial\dot{\theta}}{\partial\theta} = \pm\sqrt{\frac{-g}{2l}}\frac{\sin\theta}{\sqrt{1+\cos\theta}} \tag{A3.2}$$

We derive the equation for the homoclinic orbit by setting the sum of kinetic and potential energy of the pendulum equal to the potential energy at the inverted equilibrium:

$$0.5ml^2\dot{\theta}^2 - mgl(1-\cos\theta) = -2glm \tag{A3.3}$$

$$0.5l\dot{\theta}^2 = -2g + g(1-\cos\theta) \tag{A3.4}$$

$$\dot{\theta}^2 = \frac{-4g + 2g(1-\cos\theta)}{l} \tag{A3.5}$$

$$\dot{\theta} = \pm\sqrt{\frac{-4g + 2g(1-\cos\theta)}{l}} \tag{A3.6}$$

$$\dot{\theta} = \pm\sqrt{\frac{-2g(1+\cos\theta)}{l}} \tag{A3.7}$$

$$\dot{\theta} = \pm\sqrt{\frac{-2g}{l}}\sqrt{1+\cos\theta} \tag{A3.8}$$

The derivative of this equation is of interest, especially near the inverted equilibrium:

$$\frac{\partial \dot{\theta}}{\partial \theta} = \pm\sqrt{\frac{-2g}{l}}\frac{1}{2}\frac{-\sin\theta}{\sqrt{1+\cos\theta}} \tag{A3.9}$$

$$\frac{\partial \dot{\theta}}{\partial \theta} = \pm\sqrt{\frac{-g}{2l}}\frac{\sin\theta}{\sqrt{1+\cos\theta}} \tag{A3.10}$$

$$\lim_{\theta\to\pi^-}\frac{\sin\theta}{\sqrt{1+\cos\theta}} = \sqrt{2} \tag{A3.11}$$

$$\left.\frac{\partial \dot{\theta}}{\partial \theta}\right|_{\pi^-} = \pm\sqrt{\frac{-g}{l}} \tag{A3.12}$$

We determine the gains of the feedback controller by tuning the eigenvector of the feedback-stabilized system. We start with the linearized system with feedback around the fixed point $\theta = \pi$:

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ \frac{g}{l} & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix} u. \tag{A3.13}$$

We introduce state feedback $u = -Kx$:

$$\dot{x} = (A - BK)x, \tag{A3.14}$$

$$(A - BK - I\lambda)x = 0, \tag{A3.15}$$

$$\begin{bmatrix} -\lambda & 1 \\ \frac{g}{l} - \frac{k_1}{ml^2}) & (-\frac{k_2}{ml^2} - \lambda) \end{bmatrix} x = 0. \tag{A3.16}$$

We use the top row of Eq. (A3.16) to solve for the eigenvalue as a function of the eigenvector components:

$$-\lambda x_1 + x_2 = 0 \tag{A3.17}$$

$$\lambda = \frac{x_2}{x_1} \tag{A3.18}$$

We now solve the second row of Eq. (A3.16) for gain $k_2$:

$$(\frac{g}{l} - \frac{k_1}{ml^2})x_1 + (-\frac{k_2}{ml^2} - \lambda)x_2 = 0 \tag{A3.19}$$

$$(gml - k_1)\frac{x_1}{x_2} = k_2 + \lambda ml^2 \tag{A3.20}$$

$$k_2 = (gml - k_1)\frac{x_1}{x_2} - \lambda ml^2 \tag{A3.21}$$

$$k_2 = -k_1\frac{x_1}{x_2} + gml\frac{x_1}{x_2} - \frac{x_2}{x_1}ml^2 \tag{A3.22}$$

We set the eigenvector direction to the homoclinic orbit at the inverted equilibrium:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \theta - \pi \\ \dot{\theta} \end{bmatrix} \tag{A3.23}$$

$$\left.\frac{\partial \dot{\theta}}{\partial \theta}\right|_{\pi^-} = -\sqrt{\frac{-g}{l}} \tag{A3.24}$$

$$\frac{x_2}{x_1} = -\sqrt{\frac{-g}{l}} \tag{A3.25}$$

We can insert this relation into Eq. (A3.22) to obtain $k_2$ as a function of the system parameters and gain $k_1$:

$$k_2 = -k_1 \frac{x_1}{x_2} + gml \frac{x_1}{x_2} - \frac{x_2}{x_1} ml^2 \tag{A3.26}$$

$$k_2 = k_1 \sqrt{\frac{-g}{l}} - gml \sqrt{\frac{l}{-g}} + \sqrt{\frac{-g}{l}} ml^2 \tag{A3.27}$$

$$k_2 = k_1 \sqrt{\frac{-g}{l}} + 2ml^2 \sqrt{\frac{-g}{l}} \tag{A3.28}$$

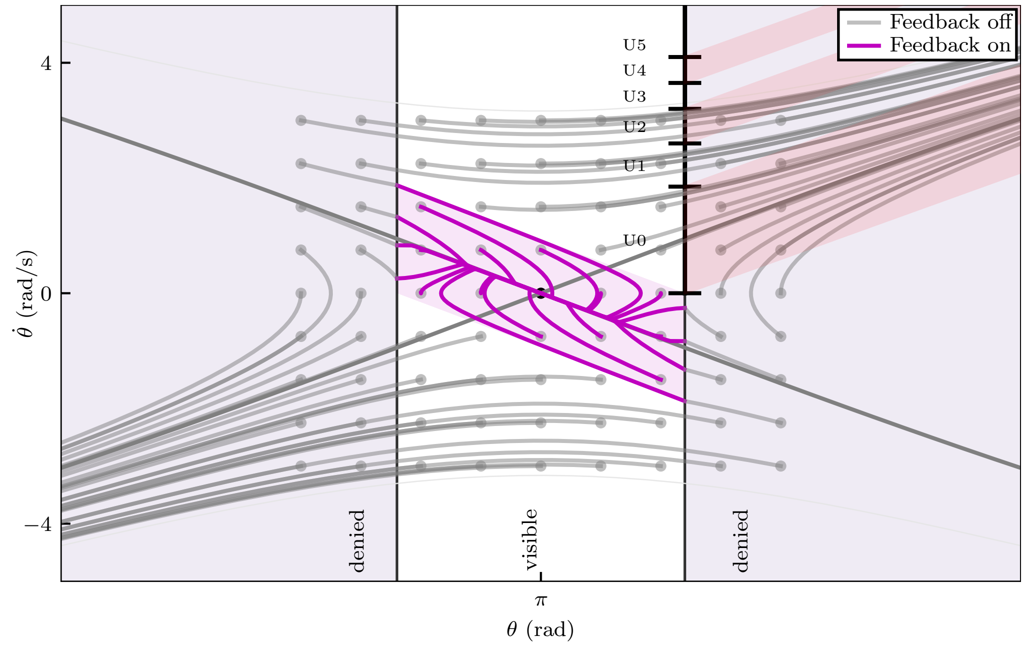

Figure S1: The region of attraction for the feedback controller for an array of initial conditions around the inverted fixed point. The purple trajectories start in the visible region and converge to the fixed point. There is a set of points in the denied region that move into feedback region (purple). Many initial conditions in the visible region are outside of the feedback region of attraction and move to the denied region.

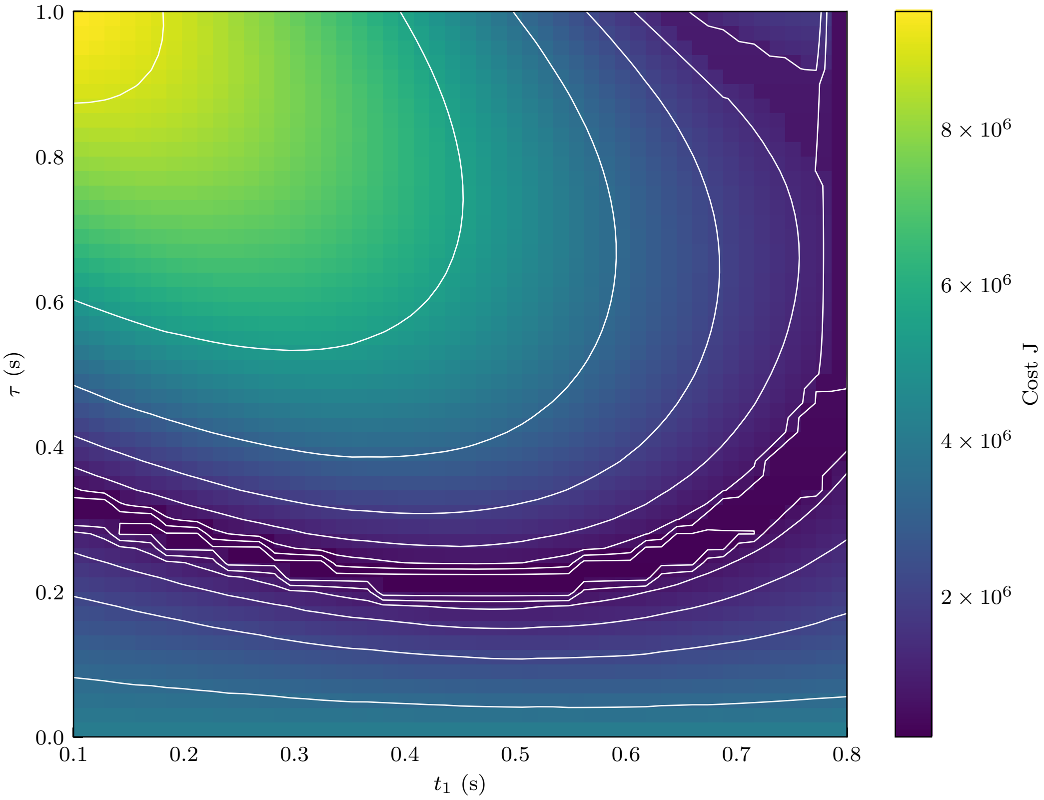

Figure S2: The cost plane for the initial condition of $[\theta, \dot{\theta}] = [-\pi + 0.3, 3]$. The optimal region (deep blue) is a long trench of similar cost.