THE BEHAVIORAL SPACE OF ZEBRAFISH LOCOMOTION AND ITS NEURAL NETWORK ANALOG

BY

KIRAN GIRDHAR

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Biophysics and Computational Biology
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2015

Urbana, Illinois

Doctoral Committee:

Associate Professor Yann R. Chemla, Co-Chair and Co-Director of Research
Professor Martin Gruebele, Co-Chair and Co-Director of Research
Professor Mark Nelson
Professor Fred Delcoymn

Abstract

How simple is the underlying control mechanism for the complex locomotion of vertebrates? I explore this question for the free-swimming behavior of zebrafish larvae. A parameter-independent method, similar to that used in studies of worms and flies, is applied to analyze swimming movies of fish. The motion itself yields a natural set of fish "eigenshapes" as axes, rather than the experimenter imposing a choice of coordinates. Three eigenshape coordinates are sufficient to construct a quantitative "postural space" that captures >96% of the observed zebrafish locomotion. Viewed in postural space, swim bouts are manifested as trajectories consisting of cycles of shapes repeated in succession. To classify behavioral patterns quantitatively and to understand behavioral variations among an ensemble of fish, we construct a "behavioral space" using multi-dimensional scaling (MDS). This method turns each cycle of a trajectory into a single point in behavioral space, and clusters points based on behavioral similarity. Clustering analysis reveals three known behavioral patterns—scoots, turns, rests—but shows that these do not represent discrete states, but rather extremes of a continuum. The behavioral space not only classifies fish by their behavior but also distinguishes fish by age. In addition to this, I have quantified escape response behavior of fish to acoustic stimuli. A parameter-free analysis was done on escape response fish movies and free-swimming movie together. The analysis showed a set of three eigenshapes is sufficient to construct the quantitative postural space to observe two different behaviors: 'escape response' and free-swimming' on same axes. With the insight into fish behavior from postural space and behavioral space, I construct a two-channel neural network model for fish locomotion, which produces strikingly similar postural space and behavioral space dynamics compared to real zebrafish.

To my parents Puran Girdhar and Asha Girdhar

and my best friend Lt. Swati Gupta

Acknowledgements

I would like to thank my co-advisors Yann Chemla and Martin Gruebele. This thesis would not have been possible without their guidance. Since I am a slow learner, I want to thank them for always giving me enough time to learn. My Friday meetings with them have taught me so much about data science which I could never have learned from a class.

I was always impressed by Yann's perfection and meticulousness for scientific work which has always made me to push my limits and learn more. Martin's optimism taught me how to stay calm and keep trying for a solution to a research problem. I cannot thank Martin enough for his patience in explaining even the most basic concepts to me.

I want to thank Mark Nelson for useful discussions on neural modeling of fish behavior.

I want to thank Ruopei Feng for contributing to improve the algorithms and instrumentation of 2D imaging of fish. His wonderful understanding of instruments has taught me a lot about hardware.

I want to thank my best friends from graduate school Piyush Labhsetwar and Arnab Mukherjee for supporting me all throughout the journey of PhD and for wonderful discussions on biophysics and molecular biology. I wouldn't have learned biology on my own as I learned from them.

I want to thank my wonderful labmates Sukrit Suksombat, Zhi Qi, Barbara Stekas, Kevin Whitley, Alice Troitskaia, Isaac Li, Rustem Khavizhov, Roshni Bano and Vishal Kottadiel. Everyday I wanted to come to lab to do great science and meet my lab mates ☺. I want to thank them and especially to Sukrit Suksombat for listening to my complaints on every damn thing from science to politics.

I want to thank Gruebele lab members Hannah Gelman, Apratim Dhar, Simon Ebbinghaus, Anna Jean Wirth, Eduardo Berrios, Irisbel Guzman, Duc Nguyen, Shu-Han Chao, Kapil Dave, Drishti Guin, Sumit Ashtekar, Maxim Prigozhim, Minghao Guo, Praveen Choudhary and Krishnarjun Sarkar for listening to my semester group meetings patiently and giving useful advice to improve my research work reasoning. I want to thank Hannah Gelman for being my support and a great friend during my Ph.D.

I want to thank CPLC for funding my research and providing me the opportunity to mentor in summer school for 4 years. It was a great experience to meet graduate students and post-docs from all over the world.

I want to thank my undergraduates Maria, Ha, Sankalp and Chamila for helping me in data analysis and building setups to image fish.

I want to thank Barbara Stekas for organizing 'Fancy Fridays.' Without Fancy Fridays in lab, my wardrobe would have been full of boring Illinois t-shirts and jeans.

I want to thank Alice, Kevin and Vishal for making coffee everyday. Without caffeine intake I would have never finished my thesis work ☺.

I want to thank Vasudha, Digvijay and Anustup for helping me to get into Loomis Building whenever I was locked out. I want to thank Utsav Agrawal, Debapriya Banerjee and Piyush Labhsetwar for being my wonderful roommates and making my graduate school days full of fun and beautiful memories. I want to thank Mansi and Tuseeta for feeding me Indian snacks and chai every weekend which was always an energy boost for my next week's work.

I want to thank my very good friend Charles from biophysics class. Without him, I wouldn't have never learned so much about USA culture and politics.

I would like to thank Charles, Barbara and Kevin who helped me in proof-reading thesis chapters. I want to thank Prabhat Tripathi for entertaining me and supporting me while I wrote my thesis.

I want to thank my very good friend from undergraduate school Nitin Arora for always encouraging me to pursue my career in research and being a great support during my PhD. I want to thank my best friend from undergraduate school Lt. Swati Gupta who taught me how to have fun in life. I am grateful to her for always being my moral support.

Finally, I want to thank my parents who taught me to never give up and to stay focused and confident in life. Their teachings have always helped me in reaching my goals.

Table of Contents

List of Figures

Chapter 1. Introduction

This chapter introduces the underlying background information and the goals of my thesis. It begins with an introduction to a vertebrate "zebrafish" which is the model organism for my thesis work and its swimming behavior. Then, I have outlined the research problems of my thesis work. I have given a brief introduction to tools that were used to study these research problems. In the end of this chapter I have provided a summary of each chapter of this thesis.

1.1 Background

1.1.1 Why study behavior?

Behavior is a direct reflection of neural activity and its modulation by external stimuli. Many tools are available to study behavior—for example, electrophysiological techniques to probe neural circuitry [1], non-invasive behavioral assays that simply record the motion of an animal [2-5], and genetic manipulation that perturb the system [6, 7].

In the past, there have been an enormous amount of findings in understanding role of genes [8] and proteins [9] in neural processes that generates behavior, but quantitative characterization of behavior went very slowly. Progress in characterizing behavior was inhibited by the fact that the methods available were not robust enough. For example, forcing an observer to link individual movements into a chain of behavior or just counting discrete actions. With the advent of faster speed in computers, we can now automatically track animals or humans and measure the behavioral phenotypes of flies [2], worms [10], ants [11], humans [12] etc. This opens the window to analyze the behavior quantitatively with new available tools.

1.1.2 Why study zebrafish (vertebrate) swimming?

The zebrafish (*Danio rerio*) (see **Figure 1.1**) is a tropical freshwater fish and native to the Himalayan region. It is a popular model organism in genetics [13], behavioral biology [14], neurobiology [15] and developmental biology [16] labs. Zebrafish eggs are large relative to other fish, (0.7 mm in diameter at the time of fertilization) and optically transparent. For studied easily under a dissecting microscope. Zebrafish development is fast and larvae start to display stimulus-

response, free-swimming behaviors within 5 days post-fertilization which makes it a useful organism to understand behavior.

Figure 1.1: Zebrafish (*Danio rerio*). This figure shows the female zebrafish (top) with fat belly and male zebrafish (bottom) with thin and slender body.

I chose zebrafish larvae 6-10 days post-fertilization (dpf) as a behavioral model organism because of the following reasons: It is complex enough as a vertebrate and yet has small number of (200-220) locomotor neurons projecting from brainstem [17] to exhibit interesting behaviors like prey tracking, escape response to different kinds of stimuli (acoustic, touch & light) and optomotor response [18]. The 6-10 dpf old fish are 3-4 mm long and their high production rate of embryos (200 per weeks) makes it easy to design cost-effective behavioral assays. The larvae's transparency makes it easy to study neuronal pathways [19-21] in order to correlate behavior with neuronal activity. In other words, the complexity of animal behavior combined with simplicity of the larvae brain, limited sets of behavioral patterns and low experimental cost makes it an interesting behavior model organism. A typical zebrafish larva swimming occurs in episodes of ~250 ms duration with intermittent resting periods (see **Movie D1.0.1**). **Figure 1.2** shows still image of zebrafish larva swimming from a video recorded in our lab at camera speed of 500 fps.

1.1.3 Why do quantitative modeling of zebrafish behavior?

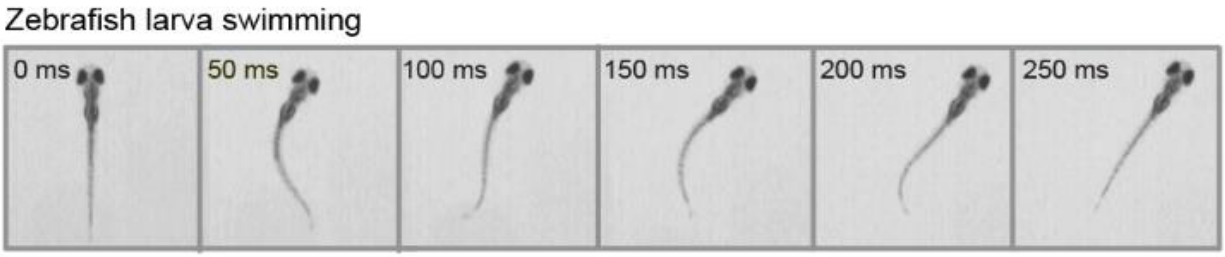

Figure 1.2: Zebrafish larva swimming. Still images of a swim bout from a representative movie of free-swimming zebrafish larva, recorded at 500 fps.

Extensive studies have categorized the behavioral patterns of zebrafish [22, 23]. A common approach [17] to study their swimming is to classify the behavior according to predetermined sets of parameters related to fish shape and swimming direction: change in head angle, angular velocity, and bend duration, to name a few. By applying boundary criteria to these predetermined parameters, the zebrafish free-swimming escape response (response to external stimuli like a pressure pulse, acoustic or tactile), and prey tracking behaviors [22, 24-28] are classified into discrete patterns: the routine turn (or R-turn), scoot, C-turn, J-turn, etc. For example, R-turns and scoots have been defined as changes in the head angle by >30-40° and <30-40°, respectively [17, 22]. Based on head angle criteria, **Figure 1.3** shows an example of a turn and scoot behavior.

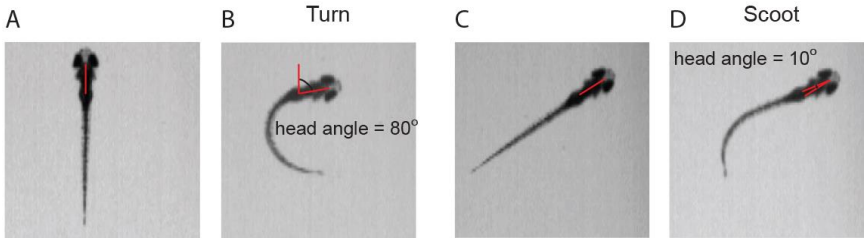

Figure 1.3: Previous classification of behavior of free-swimming zebrafish. The head angle was determined by drawing a line from the midline of the anterior end of the swim bladder to the midline of the tip of the snout (A, C) The line (red) was drawn manually at the beginning of the swim bout. (B, D) It shows the frame where the bend is completed and change in heading has reached maximum, before stopping and reversing direction. In this example, the change in heading which is the head angle is 80°, 10° and have been classified as a turn and a scoot respectively.

A limitation of this approach is that it relies on *a priori* parameters and imposed criteria to define the organism's behavioral repertoire. The question arises whether the animal itself can provide a "parameter-free" basis for the description of its motions.

Recent work [29, 30] has drawn on methods in artificial intelligence to analyze quantitatively the different set of behaviors an entity can perform. A study of the invertebrate *C. elegans* by Stephens *et al.* [31] provided a powerful method to quantify worm locomotion in terms of the worm's shapes. Its movements can be represented as a simple cyclic trajectory in a "postural space", which consists of a few carefully chosen coordinates to represent the multiple shapes the organism cycles through. Other studies [32-35] have shown that the complexity and dimensionality of behaviors can be studied by embedding such trajectories in a low-dimensional "behavioral space". My thesis is on applying similar approaches to zebrafish to understand the behavioral patterns in their free-swimming.

The key steps are summarized with actual data in **Figure 1.4**. (A) I film fish in a non-invasive behavioral assay. (B) A parameter-free analysis yields three orthogonal "eigenshapes" that reconstruct accurately zebrafish postures as a function of time. (C) In the resulting three-dimensional "postural space" the temporal sequence of shapes appears as a trajectory. (D) To quantify the range of behaviors spanned by these trajectories, I compare multiple swim bouts and animals of different ages and measure the similarity of their trajectories to one another.

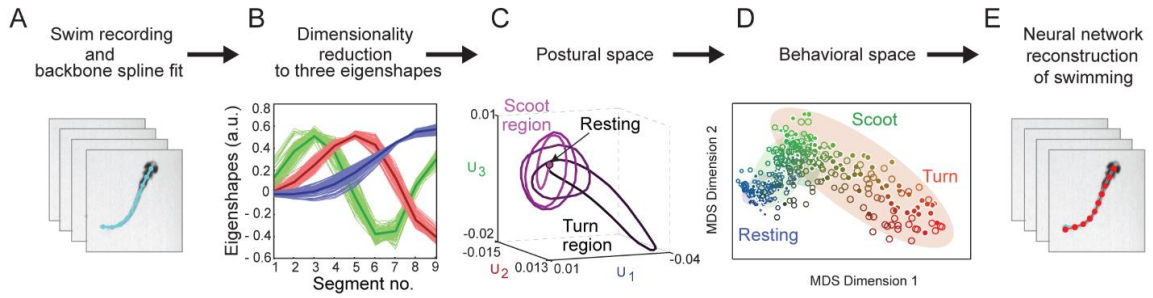

Figure 1.4: Workflow from swim observation to neuro-kinematic model. (A) Larval zebrafish swimming behavior is recorded by video and the fish backbone is fitted to a 10-point spline. (B) A linear combination of three "eigenshapes" accurately reconstructs the backbone shapes of the zebrafish. (C) A swimming bout is represented as a trajectory in the "postural space" spanned by the three eigenshapes. (D) A "behavioral space" generated by multi-dimensional scaling reduces each cycle of a trajectory to a point, and clusters them by their similarity. (E) A 2-channel neuro-kinematic model is constructed based on the observed behavioral patterns and evaluated using the same work flow.

1.2 Research goals

The primary goal of this thesis is to develop a "parameter-free" quantitative model to understand the complexity of free-swimming behavior of zebrafish, since existing methods quantify zebrafish swimming using *a priori* parameters and imposed criteria to define the organism's behavioral repertoire. My thesis work answers following questions: Can the motion itself tell us which coordinates are needed to describe behavior quantitatively, how to describe the relationship among behaviors without *a priori* classification, and ultimately how to construct a neural model for the behavior and check its success? I have listed the goals of my research, which answer the abovementioned questions.

1.2.1 How can we quantify the complex swimming behavior of zebrafish?

Until now, zebrafish swimming behavior has been quantified using parameters like bend angle, bend amplitude, angular velocity [17, 22] etc. Current methods require many parameters to quantify swimming. I propose to use a parameter-free analysis to quantify zebrafish swimming and see if it can gives us the coordinates that are needed to describe the behavior.

Basically, the idea is to apply a dimensionality reduction method on high dimensional zebrafish swimming postural data that will yield an orthogonal set of zebrafish postures, or "eigenshapes." A low-dimensional "postural-space" will be constructed whose coordinates would be eigenshapes. In this postural space the temporal sequence of shapes appears as a trajectory.

1.2.2 How to interpret free-swimming behavior in a new coordinate system?

The existing methods [17, 22] use criteria like head angle to classify the free-swimming behavior of zebrafish into turns (>30-40°) and scoots (<30-40°) as shown in **Figure 1.3**. I propose the concept of postural space to visualize free-swimming trajectories. One can easily infer the kind of behavior of zebrafish (turn-like, scoot-like or resting period) from visual inspection of trajectories in postural space.

1.2.3 How many behavioral patterns are in free-swimming behavior?

To compare the range of behavior spanned by a population of free-swimming trajectories in postural space, we measure the similarity of their trajectories to one another. We propose the concept of behavioral space, which is constructed from postural space. The behavioral space converts each trajectory from postural space into a point where distinct behavioral patterns appear as well-separated clusters of points.

1.2.4 Do we need new or more coordinates to see acoustically stimulated swimming trajectories in the postural space?

I expanded a neural model to study the acoustic stimuli response of zebrafish. The goal was to see the range of behaviors in postural space in response to acoustic stimuli.

1.2.5 What neural parameters control different behaviors in zebrafish swimming?

One existing theoretical neural model of zebrafish has shown the effect of synaptic weights on frequency of swimming fish. There has been no work done which lists the neural parameters that create different behaviors. I have reformed the existing theoretical neural model to create turn-like and scoot-like behavior. I construct a two-channel neural network/kinematics model based on xenopus swimming [36] that mimics zebrafish swimming robustly. Thus I close the loop from initial observation to neuro-kinematic simulation of zebrafish behavior.

1.3 Tools

1.3.1 Image analysis

Image analysis involves processing an image into fundamental components in order to extract data. For our application, the fundamental component is zebrafish backbone shape. Image analysis includes steps such as removing noise, finding shapes, detecting objects and measuring objects and image properties of an object. Image analysis is a broad term that covers a range of techniques that generally fit into these subcategories: background subtraction to remove noise, image segmentation to isolate regions and objects of interest and region analysis to extract statistical data (**Chapter 2**).

1.3.2 Big dataset analysis

One of the challenges for this thesis work is to develop effective ways to analyze high-dimensional data of zebrafish backbone shapes. Decomposing the data into orthogonal set of eigenshapes, singular value decomposition (SVD) [37] can be valuable tool. SVD is a common technique for analysis of multivariate data, and zebrafish backbone shapes data are well suited to analysis using SVD (see **Chapter 3**).

Another challenge with big datasets is comparing data points in a dataset to see if any patterns exists. In statistics, methods such as multidimensional scaling are required to visualize the level of similarity of individual cases of each data point (**Chapter 4**).

1.3.3 Modeling and simulation

Theoretical neural models [38] are mathematical models to show the underlying processing in a neural circuit. Due to the complexity of the brain, the experimental errors are ill-defined, but the different theoretical models of a particular system can be compared according to how closely they reproduce experimental data or respond to specific input parameters. A clearly formulated theoretical model can guide the researcher in designing experiments to probe neural circuitry.

In this thesis, I have reformulated the existing theoretical neural model to reproduce the behavioral patterns seen in experimental data. To do this, I performed some simulations to adjust the parameters in the neural model to get behavioral patterns (**Chapter 6**)

1.4 Summary

The primary goal of this thesis was to develop a quantitative model to understand the complex behavior of zebrafish. I started with building an imaging platform to capture zebrafish swimming at high-camera speed (500 fps). I have described the set-up in the beginning of **Chapter 2**. The next step was to take a large dataset (~50GB) of free-swimming videos and preprocess them to extract the shape of the zebrafish backbone. To do this I have used various image analysis techniques, which are described in **Chapter 2**. The extracted sequences of zebrafish backbone shapes were then decomposed to orthogonal set of eigenshapes using

singular value decomposition (SVD). Using the results from SVD, I constructed a low-dimensional coordinate system to visualize fish swimming trajectories (**Chapter 3**). After this, I compared the fish swimming trajectories in low-D space to determine the number of behavioral patterns in free-swimming behavior (**Chapter 4**). I used this parameter-free method to determine behavioral patterns in acoustic stimuli response and compared with free-swimming behavioral patterns (**Chapter 5**.) After obtaining the objective method to quantify zebrafish swimming, I used this method to simulate turn-like and a scoot-like behavior exhibiting fish-swimming trajectories using a theoretical neural model. The idea was to use the concept of a new coordinate system to simulate different behaviors of free-swimming zebrafish (**Chapter 6**).

Chapter 2. Acquisition and preprocessing of zebrafish swimming videos

The primary goal of this thesis is to develop a parameter-free method to understand the complexity of zebrafish swimming behavior. In this chapter, I will describe the set-up for capturing video of free-swimming zebrafish and the preprocessing of acquired videos in order to quantify fish swimming behavior using the parameter-free method in **Chapter 3.**

2.1 Background

Zebrafish are great behavioral model organisms because their larvae display a variety of behaviors such as response to stimuli, learning, tracking prey, etc. The quantification of such behaviors may facilitate our understanding of the underlying neural activity that produces these behaviors. This thesis focuses on developing a quantitative model to describe zebrafish free-swimming and escape response behaviors. As described in **Chapter 1**, the traditional approach to quantifying zebrafish swimming behavior is to a) record a sequence of images of swimming zebrafish, and then b) analyze it with pre-defined parameters [17, 22]. These pre-defined parameters, such as bend amplitude, tail beat frequency, head-tail angle, etc., have allowed researchers to quantify free-swimming behavior and distinguish it from other behaviors, such as escape response and prey-tracking [24, 27, 28, 39-42].

Our approach allows us to quantify free-swimming behavior without any a priori definition of parameters. Our model takes fish videos as an input and quantifies fish swimming behaviors in a subspace of fish shapes rather than the coordinates imposed by an experimenter.

This chapter illustrates the experimental methods which are: a) zebrafish handling and breeding, b) the instrument setup for taking swimming videos, c) performing a behavioral experiment to take free-swimming zebrafish data and d) step-wise algorithm for preprocessing the data to be input into our proposed parameter-free method, described in **Chapter 3**.

2.2 Zebrafish handling

All experiments were performed on zebrafish (*Danio rerio*) larvae age 7-10 days post fertilization (dpf). The larvae were obtained from breeding wild-type zebrafish adults. Adult zebrafish were kept in a Z-hab mini system (Aquatic habitats, Beverly, MA) fish facility at 28.5 °C on a 14h:10h light:dark cycle. Embryos were obtained from adult fish breeding and were raised at 28.5 °C in 10% Hanks solution [43]. Animals were treated in accordance with the guidelines approved by the University of Illinois Institutional Animal Care and Use Committee protocol # 13327. For breeding procedures, see **Appendix A**.

2.3 Video acquisition set-up

I imaged swimming larvae using a high speed camera (IDT vision N-3) mounted to a stereo microscope (Edmund Optics 6V head + 10X eyepiece) as shown in **Figure 2.1**. This allowed us to image of the entire 21-mm diameter Petri dish in which the larvae swam. A halogen light source (Edmund Optics MI-150 high-intensity illuminator) was used to illuminate the sample, and its light passed through a series of long-pass filters (780 nm and 830 nm) in order to obtain IR wavelengths (>810 nm).

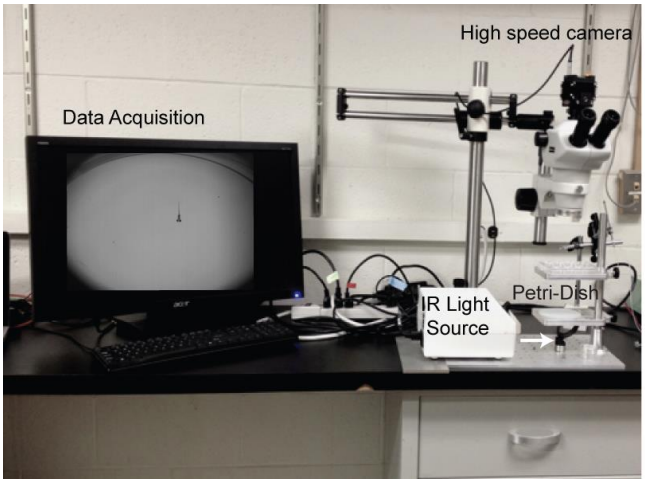

Figure 2.1: Video acquisition setup. The figure shows a high speed camera mounted to a stereo microscope to image larvae swimming in 21-mm diameter Petri dish. For IR illumination, a halogen light source coupled with long pass filters was placed perpendicular to the Petri-dish. Video acquisition was done using the IDT camera software installed on lab computer.

IR light is preferable to visible light since the larva cannot detect it [17]. A circular diffuser (100 mm dia. 120 grit ground glass diffuser, Edmund Optics) was used for uniform, homogeneous illumination. The diffuser and Petri dish were mounted on a custom-built stainless steel holder on a bread board. To ensure that the movement of the larva was restricted to the x-y plane as much as possible, measurements were carried out in water 2 mm in depth.

2.3.1 Free-swimming behavioral experiment

For each experiment, a single 7-10 dpf fish larva was placed in a 21-mm Petri dish in 10% Hanks solution at room temperature. The larvae were illuminated from the bottom with IR light. I recorded movies of a free-swimming larva at 500 fps using a high speed camera (Diagnostic Instruments) and the Motion Studio Software Suite. Each video typically had 4-5 swimming episodes ~250 ms with intermittent pauses [see **Movie D1.0.1**].

2.4 Preprocessing of acquired videos of swimming zebrafish

2.4.1 Raw video segmentation

All zebrafish swimming videos were analyzed using the MATLAB image and video processing toolbox. 50%-70% of each video consists of frames where fish are not in motion. In order to minimize the computational cost of data analysis, we have limited our analysis to frames where fish are in motion. Therefore, the first step is to segment the acquired raw videos into movies of swimming episodes by cropping out the pause frames (where fish are still). The pause frames or resting periods were identified using a motion detection-based background subtraction [44] method. This method calculates the difference between the current image and previous image. The difference is compared with a threshold value to determine whether the frame is a 'moving frame' or 'pause frame'.

I have divided the video segmentation algorithm into two parts 1) detect frames where the fish is moving and 2) determine the static background and subtract it from each image. Any frame can be viewed as matrix I with m rows and n columns. To detect frames where the fish is moving, I began by calculating the temporal sum of squares of frame difference (**equation** (2.1)) The video was downsampled at the sampling rate of 20 Hz so as to observe the significant change

in fish movement. A threshold 'T_1' of frame difference in range of 4000-5000 was empirically determined. When the difference between two consecutive frames is above this T_1, the fish is moving. **Figure 2.2A** shows an example of a raw video segmented into five swim bouts movies using the threshold value $T_1 = 4000$.

$$FD_t = \sum_{j=1}^{n}\sum_{i=1}^{m}(I_t(i,j) - I_{t+20}(i,j))^2 \quad (2.1)$$

$$FD_t = \begin{Bmatrix} moving\ \ frame\ if\ FD_t > T_1 \\ pause\ \ frame\ otherwise \end{Bmatrix} \quad (2.2)$$

Since our camera is static and illumination is constant over the whole video, the background image can be easily obtained by sorting the pixels of the movie. The foreground object 'fish' has pixel values in the intensity range of 0-45 and the background has pixel values of intensity 45-160. Sorting by pixel values moves all background pixels (45-160) to the last few frames of a movie. Choosing any frame from the upper quartile of sequences of frames in a video gives a stable background image as it is far from the foreground pixels and background noise. **Figure 2.2C** shows an example of a movie frame after background subtraction.

2.4.2 Movie frame segmentation

A stepwise image segmentation algorithm was followed on every frame of a swim bout movie obtained after background subtraction to identify the outline of a fish backbone. This subsection includes the steps to convert grayscale images to thin fish backbone shapes. **Figure 2.2B-F** shows the output of these steps applied to a background subtracted grayscale image.

2.4.2.1 Recognition of head of a fish

The very first step after subtracting the background is recognizing the head of the fish in every frame of the swim bout movie. I define the head location as the midpoint between the centroids of the fish's eyes (as labeled in red in **Figure 2.2D**.) I locate the position in each frame because subsequent processing steps can produce variation in the length of the fish skeleton

from frame to frame. Recognizing head location fixes a point on the fish body which is known in all frames and minimizes the standard deviation in length of the fish skeleton for a movie.

The head location recognition has two steps: A) adjust the intensity of an image such that the eyes are the brightest objects, and then B) detect the eyes using a built-in MATLAB function 'imfindcircles' which finds the most circular objects in an image. The algorithm uses the Hough transform [45] to detect circular object in an image. I used a custom GUI to adjust the intensity of a first frame of a movie and use that value for the rest of the video. See the GUI in **Figure B.1**, **Appendix B**.

2.4.2.2 Binary conversion of a grayscale image

Next, the frames of swim bout movies were converted to 1 bit (black/white) using a built-in MATLAB function called 'im2bw'. This function converts the input grayscale image to a binary image by replacing pixels with luminance greater than a threshold level with the value 1 (white) and all other pixels with the value 0 (black). The threshold is selected using a custom GUI which facilitates finding the best threshold value for a particular movie. See the GUI in **Figure B.1**, **Appendix B**. **Figure 2.2E** shows an example of binary conversion of a grayscale image in **Figure 2.2B**.

2.4.2.3 Thinning of fish shape from binary image

A thinning algorithm was applied to the swim bout movie frames to find the center of the fish shape, corresponding to the backbone. Thinning is a morphological operation [46] in digital image analysis that is used to remove selected foreground pixels from binary images, somewhat like erosion. For our application, it reduces a fish shape in binary format to single pixel thickness. I applied the thinning operation of 'bwmorph,' a built-in MATLAB function, to all movies. **Figure 2.2F** shows an example of thinning operation on a binary image in **Figure 2.2E**.

Sometimes the thinning operation results in unwanted branching of a fish backbone (see **Figure B.2**, **Appendix B**.) I wrote a function which finds all endpoints of a fish shape after thinning and then calculates the arc length from head location to other end points. The fish backbone is selected as the one which has longest arc length.

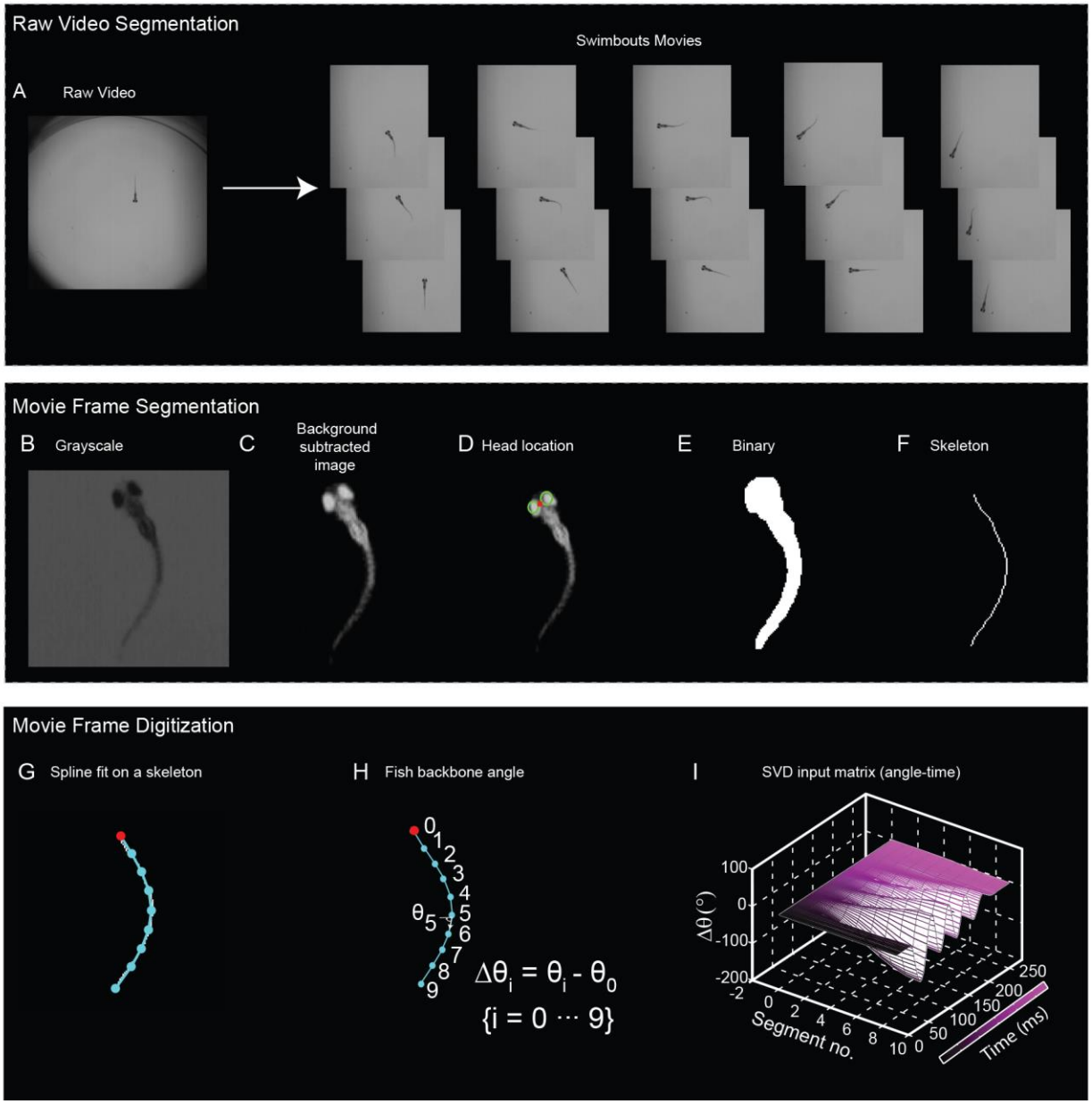

Figure 2.2: Video preprocessing steps. Raw video segmentation**.** (A) A video acquired at 500 fps segmented into 5 swim bouts. Still images of a swim bout from a representative movie are shown. Movie frame segmentation: (B) An image from a frame in the movie. (C) The same image after applying the background subtraction algorithm. (D) Recognition of head location (red) as a midpoint of centroid of eyes (green.) (E) Binary image obtained from thresholding the background-subtracted image in C. (F) "thinning" of the binary image. Movie frame digitization (G) Cubic spline fit to fish backbone (cyan line) with ten equally spaced points (cyan circles). (H) Tangent vectors (white arrow) at 10 evenly spaced segments along the backbone from head (s_0 = 0) to tail (s_9 = 9) point along a direction $\theta(s_j)$. (I) Swim bout parameterized by $SVD\ as\ \Delta\theta(s_j,t_i) = \theta(s_j,t_i) - \theta(0,t_i)$

2.4.3 Movie frame digitization

After frame segmentation, the fish shape is reduced to a single-valued vector of bright pixels in every frame of the swim bout movie. The coordinates of the fish backbone are extracted from every frame for further data analysis. This step avoids the analysis on background pixel and reduces the computation cost.

Since the backbone data is digitized, I used cubic spline fitting to smooth the fish backbone data. This subsection describes how to fit a cubic spline to two-dimensional fish backbone pixels and how to reduce that to one-dimensional fish spine angles.

2.4.3.1 A spline fit to a fish backbone skeleton

Before fitting the 2-dimensional fish backbone pixels to a cubic spline [47], I ordered the extracted data from head to tail. I calculated the arc length (s) of a fish backbone and fitted to a cubic spline individually on x-axis data and y-axis data as a function of arc length. The fitting was done using the 'csaps' built-in MATLAB function. The fitting function for each axis was then calculated at *n* evenly spaced points along the backbone arc length. **Figure 2.2G** shows an example of spline fitting at 10 evenly spaced points along the fish backbone (**Figure 2.2F**.)

2.4.3.2 Fish spine angles

The backbone of the fish in each frame was fit to a cubic spline. As shown in **Figure 2.2H-I**, the spline curve was converted into a one-dimensional array of spine angles $\theta\left(s_j,t_i\right)$, measured along the normalized arc length of the fish, *s*. Spine angles were measured at *n* = 10 regular intervals from head ($s_0 = 0$) to tail ($s_9 = 1$). All spine angles are given relative to the head angle, $\Delta\theta\left(s_j,t_i\right)=\theta\left(s_j,t_i\right)-\theta\left(s_0,t_i\right)$. **Figure 2.2** shows an example of a swim bout parameterized by $\Delta\theta\left(s_j,t_i\right)$. One can see, $\Delta\theta=0$ at the head and increases to maximum values towards the tip of the tail.

Chapter 3. Free-swimming trajectory in low-D postural space

In the previous chapter, I described both the setup used to capture free-swimming zebrafish videos and each step for preprocessing of acquired videos. Zebrafish larvae swim in episodes of ~250 ms with intermittent rest periods. The data preprocessing was done to extract the swimming episodes or swim bout movies from the acquired free-swimming zebrafish videos using our setup. Each frame of a swim bout movie was then preprocessed using image analysis algorithms to extract the zebrafish backbone shape. The 2-dimensional backbone shape was reduced to a one-dimensional array of fish spine angles $\Delta\theta\left(s_j, t_i\right)$.

There are two objectives of this chapter: 1) to show how one can quantify the swimming of zebrafish using a parameter-free method. 2) To demonstrate how one can interpret the behavior in a new coordinate system that comes from parameter-free analysis. For the first objective I take a large dataset of fish spine angles $\Delta\theta\left(s_j, t_i\right)$ of all swim bout movies and apply singular value decomposition (SVD) [48]. For the second one, I introduce the concept of "postural space" to visualize behavior and the method to construct postural space**.** To my surprise, one can visualize all stereotyped behaviors: turns, scoots, and resting periods in a low-dimensional space where the coordinates are zebrafish backbone shapes.

3.1 Background

As described in **Chapter 1**, previous studies have classified the free-swimming behavior of zebrafish as turns and scoots. Turns and scoots were identified from parameters like head angle, bend curvature, head-tail angle, angular velocity, etc. A turn is defined as a bend in response to no apparent stimulus, and has a head angle (change in head orientation) of at least 30°, whereas a scoot has no apparent change in head orientation (head angle < 30°) [17, 22].

The limitations of this approach are: the experimenter has pre-defined parameters (e.g. head angle and bend curvature) and includes arbitrary criteria (e.g. turns defined as head-angle > 30°) to classify fish swimming behavior. Our approach is adapted from previous work done on worms, taking the large dataset of zebrafish spine angles $\Delta\theta\left(s_j, t_i\right)$ and using SVD to see if the

motion itself can tell us which coordinates are needed to describe behavior quantitatively. In the subsections of this chapter, I explain the steps of how to apply SVD to zebrafish spine angles $\Delta\theta\left(s_j,t_i\right)$ and how to visualize the swimming behavior of both a single larva and a population of larvae in a new coordinate system.

3.2 Low-D space representation of zebrafish free-swimming

SVD is a parameter-free linear algebra technique that, in our application, reduces all shapes to a small number of linearly independent "eigenshapes." Before going into the detailed description of using SVD on zebrafish shapes, I start this section with a mathematical definition of SVD which will help in understanding the implementation of SVD on high-dimensional data of zebrafish spine angles.

3.2.1 Mathematical definition of SVD

Let *X* be an *m* x *n* matrix data with *rank r*. SVD is the factorization of the matrix X given by:

$$X = USV^T \tag{3.1}$$

where *U* is an *m* x *n* matrix, *S* is an *n* x *n* diagonal matrix, and V^T is also an *n* x *n* matrix. The columns of *U* and V are called the *left singular vectors* $\{u_k\}$, and *right singular vectors* $\{v_k\}$, respectively. The left singular vectors form an orthonormal basis for the swim bout movie, and right singular vectors, which are rows of V^T, form an orthonormal basis for the zebrafish spine angles. The elements of *S* are only nonzero on the diagonal, and are called the *singular values*. Thus, $S = \text{diag}(S_1,\ldots,S_n)$. Furthermore, $S_k > 0$ for $1 \le k \le r$, and $S_k = 0$ for $(r+1) \le k \le n$. The singular vectors are sorted from most important (largest singular value; $k = 1$) to least important (smallest singular value; $k = n$).

If the matrix is not too large, the SVD of a matrix can be calculated by evaluating the eigenvectors of XX^T and X^TX. The columns of *U* and *V* are eigenvectors of XX^T and X^TX respectively. Finally, the elements of the singular value S diagonal matrix are the eigenvalues of XX^T. The other approach is to calculate the columns of *V* and *U* using:

$$X^TX = VS^2V^T \tag{3.2}$$

$$U = XVS^{-1} \tag{3.3}$$

In practice there are several algorithms for calculating SVD with varying accuracy and speed.

3.2.2 Singular value decomposition of zebrafish spine angles

For our application, matrix X in **equation (3.2)** contains zebrafish spine angles $\Delta\theta(s_j, t_i)$. This was obtained after preprocessing of free-swimming zebrafish videos (see **Chapter 2**). The SVD input matrix $\Delta\theta(s,t)$ is an *m* × *n* matrix in which each row contains the spine angles from head to tail, and each column represents the time. The matrix $\Delta\theta(s,t)$ was then decomposed by SVD into a set of *n* = 10 linearly independent, orthonormal basis functions V_k, as given by the relation

$$\Delta\theta(s_j, t_i) = \sum_{k=1}^{n} U_k(t_i) S_{kk} V_k(s_j) \tag{3.4}$$

Figure 3.1 shows a graphical representation of the SVD input matrix and its decomposition. The basis functions $V_k(s_j)$ represent eigenshapes of the zebrafish backbone. Each $U_k(t_i)$ represents the amplitude of the k^{th} basis function $V_k(s_j)$ at each time point t_i. S_{kk} is an n × n diagonal matrix of singular values. The linear combination of these three matrices reconstructs the spline-fitted backbone shapes exactly, according to **equation (3.4)** which is discussed in detail in the next subsection.

As discussed in **section 3.2.1** the singular vectors are determined by first sorting them from high-to-low singular values, with the highest singular value in the upper left index of the *S* matrix. S_{kk} is conventionally normalized to 1, such that each singular value represents the fractional contribution a basis function makes to the overall swimming behavior. The key element of this analysis is that many of the singular values in the matrix S_{kk} may be small, and thus many basis functions can be left out of the sum in **equation (3.4)** with negligible effect.

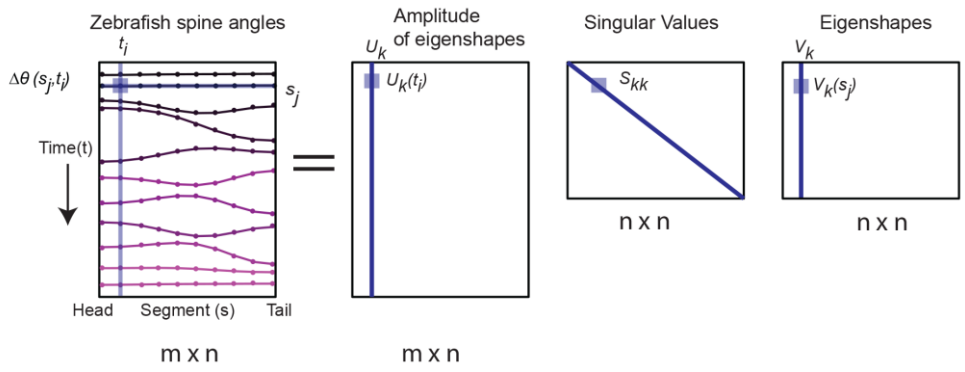

Figure 3.1: Graphical representation of SVD. Singular value decomposition of the matrix of zebrafish spine angles to an orthogonal set of basis functions (eigenshapes) (V_k) with singular values S_{kk} and amplitude of each eigenshape U_k.

Performing SVD on all movies (N = 115) from a population of 20 fish reveals that 96% (see **Figure 3.2B**) of the variation in $\Delta\theta$ is accounted by the first three eigenshapes only, (*i.e.* taking the summation in **equation (3.4)** only up to n = 3). Plotting the three eigenshapes (light blue, red, and green in **Figure 3.2A**) from individually analyzed movies shows that the basis functions were similar across the population of fish. The collective eigenshapes shown in dark colors (dark blue, red, and green) were obtained by analyzing all movies together.

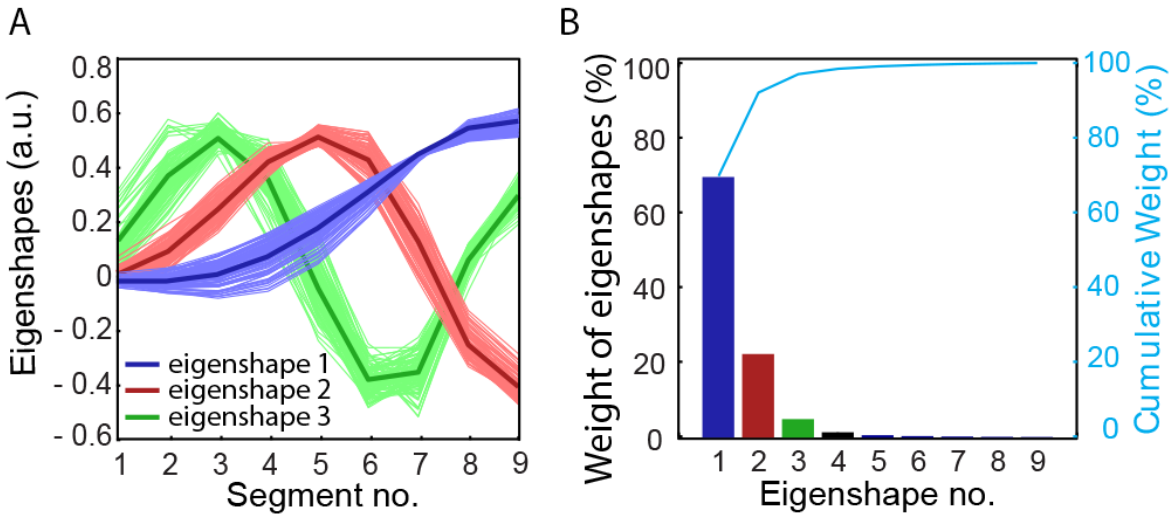

Figure 3.2: SVD of zebrafish spine angles. (A) Singular value decomposition of $\Delta\theta$ into eigenshapes $V_k(s_j)$ (j = 1, 2... 9), the first three of which are plotted (red, blue, and green, respectively). The light colors correspond to eigenshapes determined from analyzing individual fish from a population, and dark colors are the collective eigenshapes from analyzing the entire population at once. (B) Bar plot of % weights ($S_{kk}/\sum_{k'=1}^{10} S_{k'k'}$, where S_{kk} are the singular values) of each eigenfunction $V_k(s_j)$. The right axis in cyan shows the cumulative contribution of each eigenshape. The first three eigenshapes contribute 96% of the total variance in $\Delta\theta$.

3.2.3 Why are only three eigenshapes required?

I calculated the reconstructed matrix of spine angles $\Delta\theta^r(s,t)$ using **equation (3.4)** using only three eigenshapes by setting the basis function $V_k = 0\{k = 4-10\}$. The residual error was ~7% between the spine angles $\Delta\theta(s_j,t_i)$ and reconstructed spine angles $\Delta\theta^r(s_j,t_i)$ using the first three eigenshapes as shown in **Figure 3.3.** This result shows that our new coordinate system, where there are only three dimensions (k = 3) is sufficient to reproduce the swimming shapes of fish.

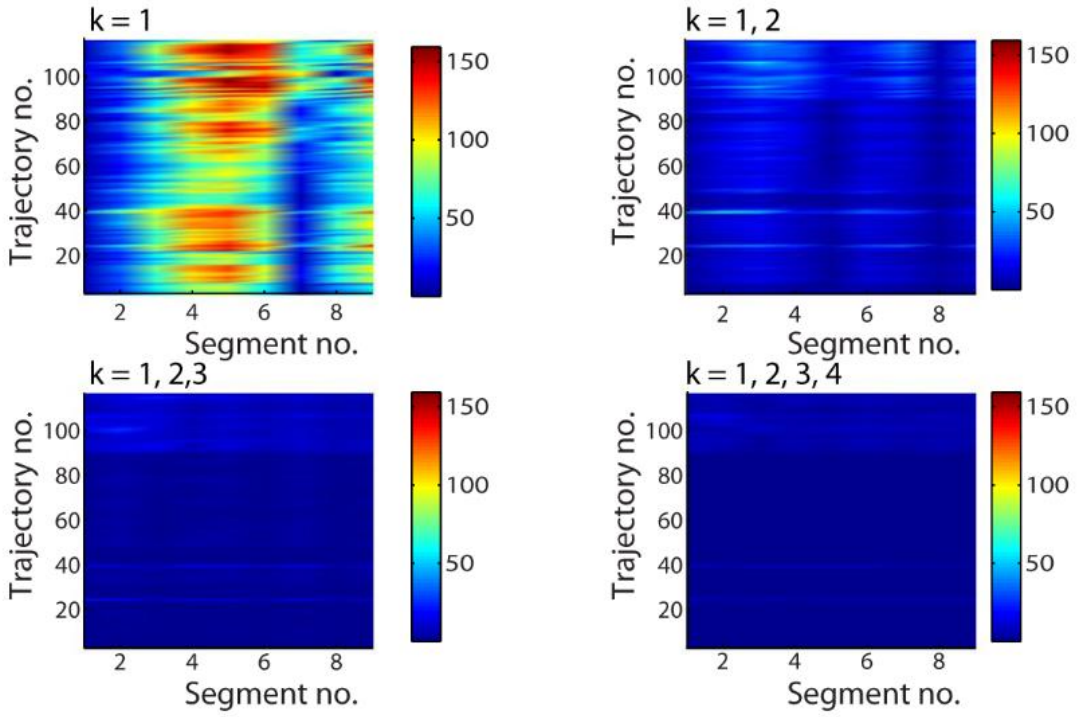

Figure 3.3: Residual error in reconstructed spine angles using eigenshapes. The residual error between all trajectories and their reconstruction $\Delta\theta(s_j,t_i) - \Delta\theta^r(s_j,t_i)$ using the first eigenshape (k = 1; top-left panel), the first two eigenshapes (k = 1, 2; top right), the first three eigenshapes (k = 1, 2, 3; bottom left) and the first four eigenshapes (k = 1, 2, 3, 4; bottom right). The difference in residual error is insignificant for more than 3 eigenshapes.

3.2.4 Why was the spline on the fish backbone sampled for n=10?

I sampled the spine angles over a range n = 2–100 evenly spaced intervals, and found that the singular values from SVD analysis converged for $n \geq 10$ (**Figure 3.4**) A value of n = 10 spline points thus keeps the number of angles to the necessary minimum while capturing all of the fish swimming shapes.

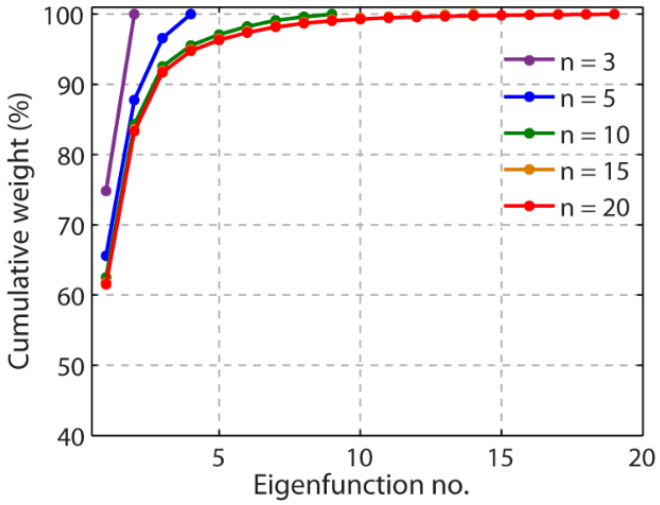

Figure 3.4: Effect of number of sampled points on zebrafish backbone decomposition into eigenshapes. Each line on the plot is the cumulative contribution of each eigenshape (as in **Figure 3.2B**) obtained for movies with different numbers of sampled points on the backbone spline: n = 3, 5, 10, 15, 20. The first three eigenshapes contribute 96% of the total variance in $\Delta\theta$ when n > 8. There is no significant difference in the eigenshape contribution for n ≥ 10.

3.2.5 Real space representation of fish backbone eigenshapes

The next step was to translate the eigenshapes into real space. The eigenshape $V_k\left(s_j\right)$ can be visualized in real space by translating the fish spine angles into Cartesian coordinates X_j, Y_j. I have defined $\Delta\theta_k(s_j,t_i)$ as the fish spine angle reconstructed from the k^{th} eigenshape:

$$\Delta\theta_k(s_j,t_i) \equiv U_k(t_i)S_{kk}V_k(s_j) \tag{3.5}$$

The corresponding Cartesian coordinates X_j, Y_j are given by:

$$X_j = \sum_{j=1}^{n} \cos\left(\Delta\theta_k(s_j, t_i)\right)$$
$$Y_j = \sum_{j=1}^{n} \sin\left(\Delta\theta_k(s_j, t_i)\right) \qquad (3.6)$$

Figure 3.5 displays the real-space representations of the 3 eigenshapes for each frame (t_i) in a representative movie. The highlighted shapes correspond to three frames in which the contributions from one eigenshape is large (V_1, blue; V_2, red; V_3, green). Eigenshapes V_1, V_2, and V_3 correspond to shapes with one bend, two bends, and three bends, respectively.

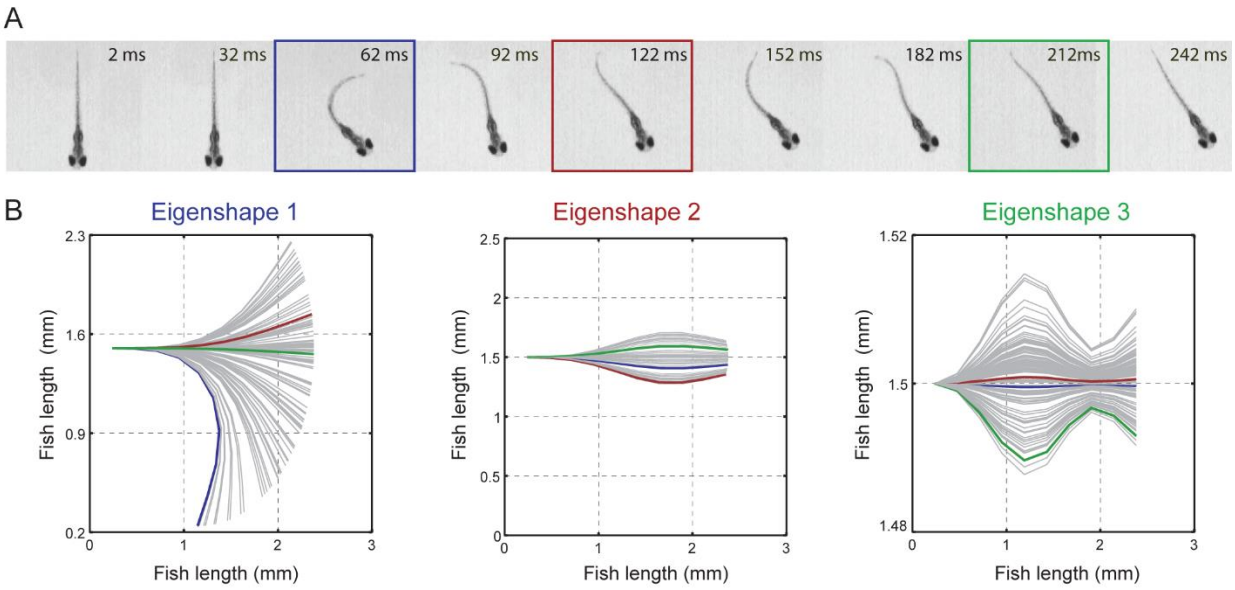

Figure 3.5: Real-space representation of zebrafish swimming eigenshapes. (A) Snapshots of a movie of a free-swimming zebrafish recorded at 500 fps. (B) Shown are the real-space shapes (gray lines) corresponding to the basis function $V_k(s_j)$ where k = 1, 2, 3 (left, middle, and right panels, respectively) for the swim bout shown in A. These shapes were reconstructed using only eigenshape 1, 2, 3, respectively, as described by **equation (3.6)**. Each of the shapes in color corresponds to the frames marked with blue, red and green boxes in A. The zebrafish shape in the blue frame consists mostly of eigenshape k = 1 (left panel, solid blue line), i.e. a shape with a single bend. The zebrafish shape in the red frame consists mostly of eigenshape k = 2 (middle panel, solid red line), i.e. a shape with a two bends. The zebrafish shape in the green frame consists mostly of eigenshape k = 3 (right panel, solid green line), i.e. a shape with a three bend.

3.3 Postural space of a single fish

This chapter introduces the concept of "postural space" to visualize zebrafish swimming behavior. The postural space is a three dimensional space where coordinates are eigenshapes and are sufficient to capture >96% of the observed zebrafish locomotion (**Figure 3.2B**.) In this section, I demonstrate the method of constructing the postural space and interpret behavior from the postural space for a single larva.

3.3.1 Visualization of turns and scoots in the postural space

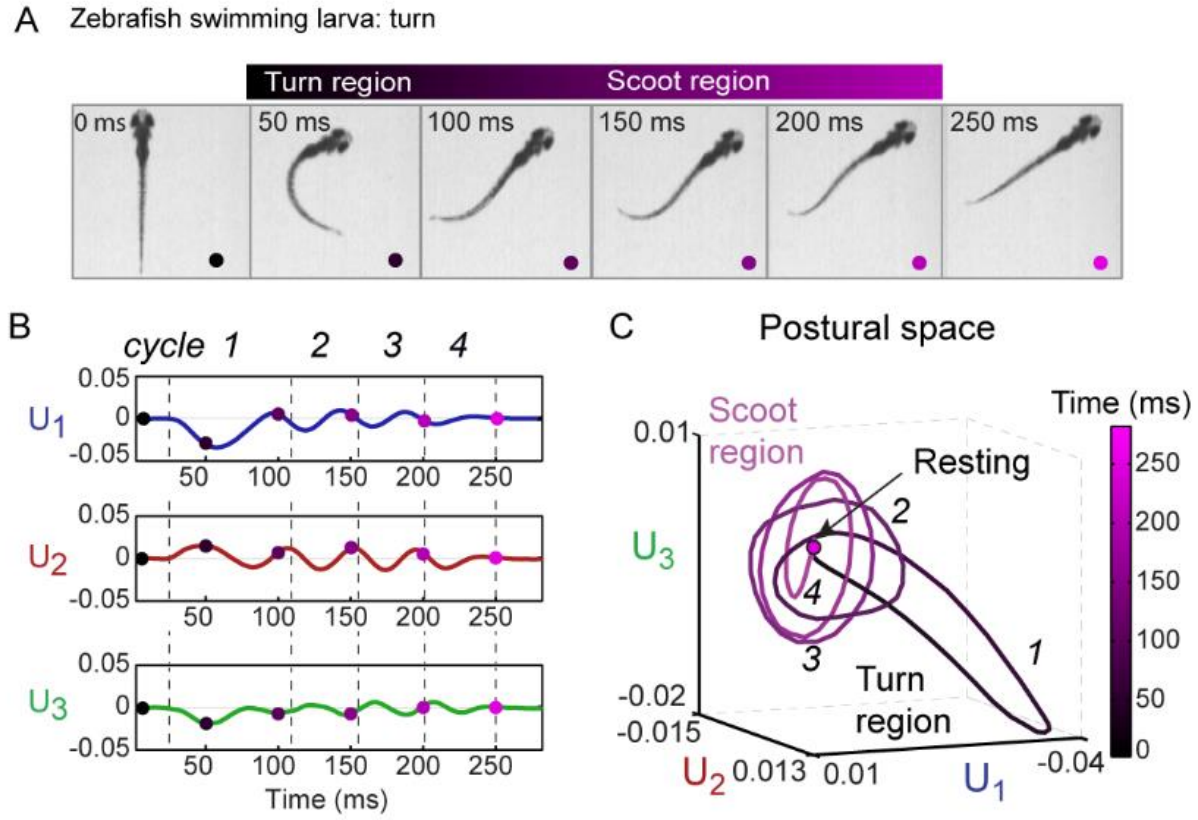

Figure 3.6: Representation of turn bout of a single larva in low-dimensional postural space. (A) Still images of a representative turning bout during free-swimming. As discussed in the text, it is convenient to divide the bout into a "turn" region (t = 50-100 ms) followed by a "scoot" region (100-250 ms). (B) Plot of the amplitudes $U_1(t)$, $U_2(t)$, and $U_3(t)$ of the three collective eigenshapes corresponding to the movie in A. Each amplitude undergoes multiple oscillation cycles before returning to zero. The regions marked by dashed lines and labeled as cycles (1-4) in $U_1(t)$, $U_2(t)$, and $U_3(t)$ are obtained from the oscillation cycles in $U_1(t)$. The colored dots mark time points corresponding to the still images in A. (C) Representation of a turn bout in postural space. The three-dimensional coordinates of the trajectory are the amplitudes $U_1(t)$, $U_2(t)$, and $U_3(t)$ in B. In this space, the bout involves a turn region (t = 50-100 ms), represented as a bent ellipse (cycle 1), followed by a scoot region (t = 100-250 ms) represented as multiple cycles (2-4) along the flat ellipses, and a final return to the rest behavior. Throughout, time (0–250 ms) is represented by the black–magenta colormap.

A postural space can be constructed by plotting the amplitudes U_k (k = 1, 2 and 3) of basis function V_k. **Figure 3.6** shows an example of a zebrafish turning bout and how it can be visualized as a trajectory in the postural space. At each time point t_i of the movie, the zebrafish backbone shape is represented by a set of three amplitudes $\{U_k(t_i)\}$ with k = 1, 2, and 3. **Figure 3.6B** plots these three amplitudes U_k vs. time, and in **Figure 3.6C** the three amplitudes define the coordinates of this fish's trajectory in the three-dimensional postural space. Represented in this manner, a swimming bout appears as a sequence of cycles of shapes (see **Movie D1.0.2**). For this particular example the swim bout has four cycles, as shown by dashed lines in **Figure 3.6B**, that explore different regions of the postural space in **Figure 3.6C**.

Figure 3.7C shows an example of a scoot bout in postural space (see **Movie D1.0.3**). It has oscillation cycles predominantly in U_2 and U_3, returning to a fixed point corresponding to a "resting" fish. In contrast, the turn bout in **Figure 3.6C** takes an excursion along the U_1–U_3 plane lasting between one half and one whole oscillation of cycle 1, before shifting to cycles 2-4 in U_2 and U_3 , which are similar to those in the scoot trajectory (see cycles 1-5 in **Figure 3.7C**), and eventually returning to the resting behavior.

The sign of U_1 corresponds to the direction of the turn or scoot, with $U_1 < 0$ corresponding to a right turn or right undulation in scoot and $U_1 > 0$ to a left turn or left undulation in scoot. For this analysis, I have considered these cases to be mirror images of the same behavior. To distinguish the different cycles of a trajectory in postural space, I found it useful to name the excursion in the U_1–U_3 plane "turn region", the oscillation cycles in the U_2–U_3 plane "scoot region", and decaying to zero amplitude as "resting" behavior. Thus a swimming bout can progress through these regions in postural space in succession.

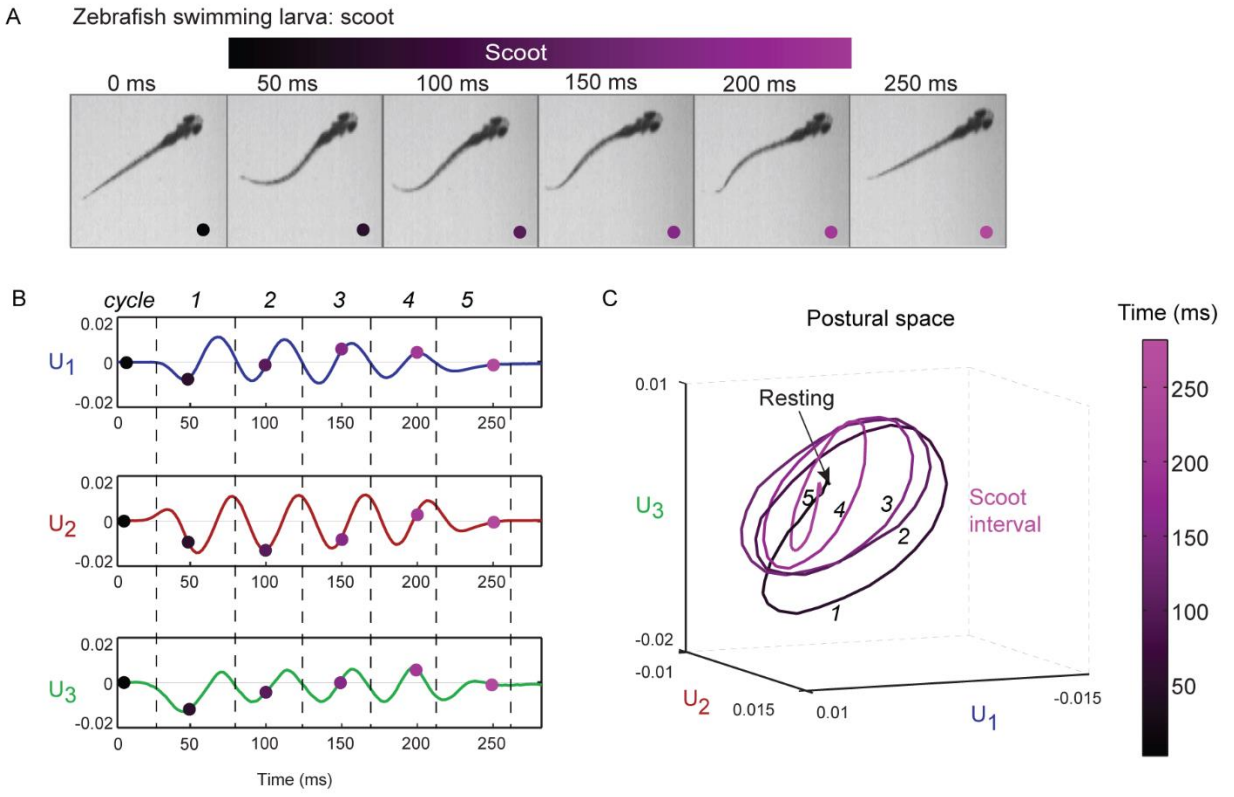

Figure 3.7: Representation of scoot bout of a single larva in low-dimensional postural space. (A) Still images of a representative scooting bout (from 50-250 ms) during free-swimming. (B) Plot of the amplitudes $U_1(t)$, $U_2(t)$, and $U_3(t)$ of the three collective eigenshapes corresponding to the movie in A The regions marked by dashed lines and labeled as cycles (1-5) in $U_1(t)$, $U_2(t)$, and $U_3(t)$ are obtained from the oscillation cycles in $U_1(t)$. (C) Representation of a scoot in a postural space. The three-dimensional coordinates of the trajectory are the amplitudes $U_1(t)$, $U_2(t)$, and $U_3(t)$ in B. The bout entails multiple cycles (1-5) along a flat ellipse in this space. Throughout, time (0 – 250 ms) is represented by the black–magenta colormap.

3.4 Postural space of a population of fish

3.4.1 Visualization of turns and scoots in postural space

I took all the swimming trajectories (n = 115) from 20 organisms and constructed the postural space to visualize the behavior of a population of fish on one plot. **Figure 3.8** shows the postural space of a population of fish. The most important feature to observe in **Figure 3.8** is that the oscillation cycles in magenta, predominantly in U_2 and U_3 (scoot region), and the oscillation cycles in black shade (turn region) along the U_1–U_3 plane are in continuum with each other. This

observation raises the question: are turns and scoots distinct behavioral states or extremes of the same behavior? I answer this question in detail in the next chapter.

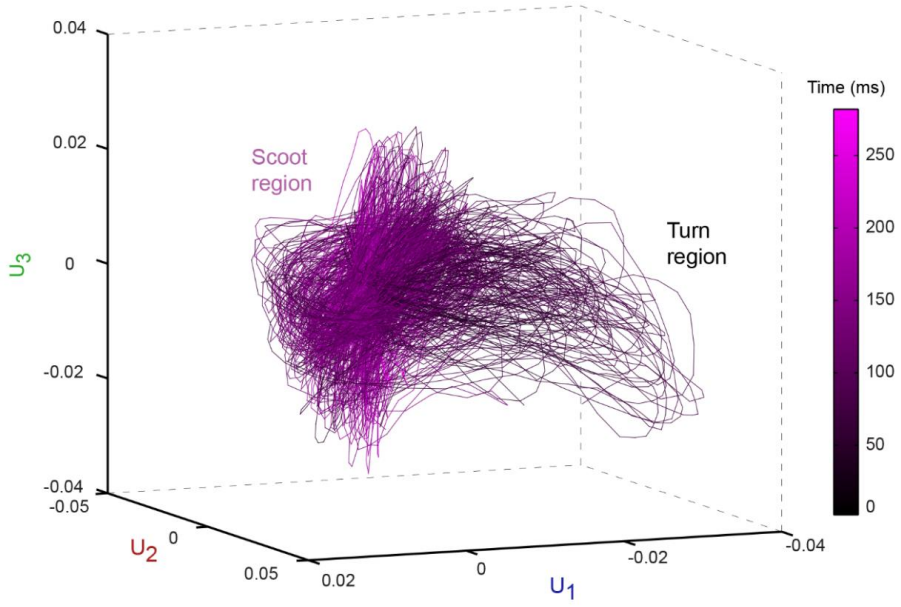

Figure 3.8: Representation of swimming trajectories of a population of fish in low-dimensional postural space. The three-dimensional coordinates of the trajectory are the amplitudes $U_1(t)$, $U_2(t)$, and $U_3(t)$. The multiple cycles along the flat ellipse (bent ellipse) in magenta (black) represent scoot-like (turn-like) regions of swim trajectories. Throughout, time (0-250 ms) is represented by the black-magenta colormap.

3.5 Discussion

The goal of this chapter was to show how one can quantify the zebrafish swimming using a parameter-free approach, followed by how to interpret the swimming behavior in a new coordinate system obtained from this parameter-free analysis. In **Chapter 2**, I showed that any free-swimming zebrafish movie can be condensed to a one-dimensional array of spine angles $\Delta\theta\left(s_j,t_i\right)$. SVD of $\Delta\theta\left(s_j,t_i\right)$ showed that three eigenshapes (**Figure 3.2**) describe zebrafish swimming with an accuracy of 96%. The reconstructed zebrafish spine angles using only these three eigenshapes showed only 4% of residual error (**Figure 3.3**). The projection of zebrafish swimming onto the low-dimensional postural space of eigenshapes provides a parameter-free analysis to quantify behavior. The postural space of a single larva (**Figure 3.6, Figure 3.7**) reveals that all stereotyped behaviors of free-swimming fish can be viewed as repeating cycles of

postures that appear as a combination of a bent ellipse and a flat ellipse for a turning bout or just a flat ellipse for a scooting bout in the low-dimensional space. The postural space of a population of larvae (**Figure 3.8**) shows that there are no distinct classes of behaviors; instead they form a continuum with each other. I discuss results that test the presence of a continuum of behaviors in the next chapter.

Chapter 4. Behavioral analysis of free-swimming fish

Previous chapters showed that free-swimming zebrafish movies can be visualized in the low-dimensional 'postural space' with coordinates of eigenshapes. In this postural space, free-swimming behavior can be viewed as oscillatory cycles. For example, during a scooting bout, a fish repeats a cycle of postures, projected as a flat ellipse in postural space, until eventually returning to a resting state. In contrasts, during a turning bout the fish adopts a different set of shapes in its first cycle, but then returns to the same postural cycles of the scooting bout. A visual inspection of postural space for a population of fish (**Figure 3.8**) showed a continuum between the turn and scoot regions of trajectories. This observation motivated me to quantify the behavioral variability among trajectories that lie in this continuum between turns and scoots.

In this chapter, I show how differences between trajectories in postural space are quantified and later classified into different swimming behaviors. First, differences between trajectories are calculated using multidimensional scaling (MDS) that embeds every cycle of a trajectory in a postural space as a point in an even lower dimensional 'behavioral space.' Next, I show that the oscillation cycles cluster into distinct classes representing different behavioral patterns. I use a kernel density estimation and k-means clustering method the revealed clustering was weak, with no strong boundaries in the data.

4.1 Alignment of trajectories in postural space

As described in **Chapter 3,** a postural space is a three-dimensional space where axes are eigenshapes (V_k). Each point in the three-dimensional postural space is given by the amplitude (U_1, U_2, U_3) of three eigenshapes (V_1, V_2, V_3) for an instantaneous fish swimming posture. The temporal sequence of shapes appears as a trajectory, which consist of cycles of shapes repeated in succession.

In order to quantify the behavioral differences between trajectories in the postural space for an ensemble of fish, amplitudes U_1, U_2 and U_3 must be aligned temporally**. Subsection 4.1.1** explains these steps in detail.

4.1.1 Temporal alignment of amplitudes

Temporal alignment of the amplitudes was achieved in two steps. First, the time axes are normalized by matching the duration of the first oscillation period in $U_1(t)$, termed "cycle 1" for all trajectories; see **Figure 4.1.** The nth cycle is defined as the time from the (2n – 1)th and (2n + 1)th zero crossings of amplitude U_1. Second, I maximize the overlap between the trajectories using a [49]. Each trajectory α can then be assigned a "normalized time" $\tilde{t}^{\alpha} \equiv a_{\alpha} t + b_{\alpha}$ with the set of parameters $\{a_{\alpha}\}$ (scaling time) and $\{b_{\alpha}\}$ (offset in time) optimized as described in **Appendix B**. **Figure 4.1** shows the plot of normalized time vs. amplitudes (U_1, U_2, U_3). The amplitude data is divided into six cycles, which correspond to the oscillation cycles seen in postural space.

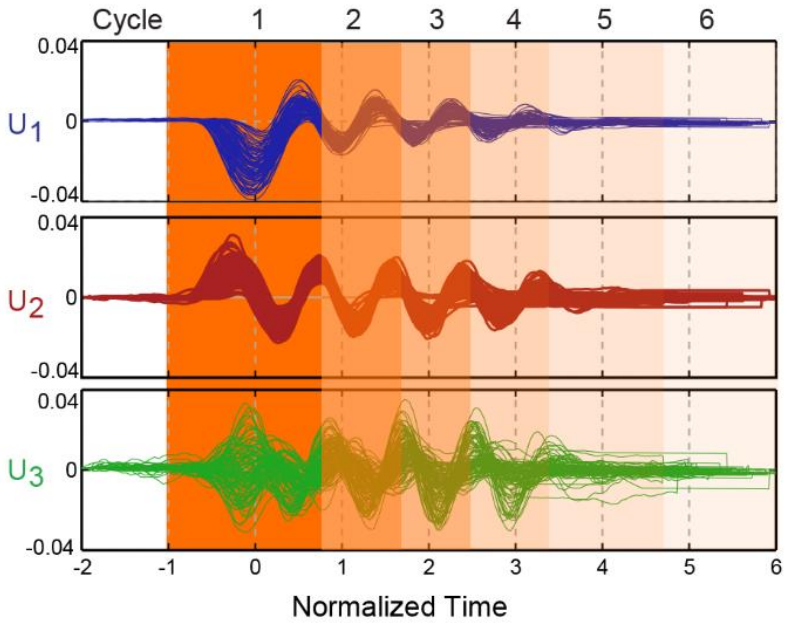

Figure 4.1: Temporal alignment of amplitudes U_1, U_2, U_3. The amplitudes U_1, U_2, U_3 of each eigenshape V_k (k = 1, 2, 3 in blue, red and green, respectively) for all fish swimming trajectories were aligned in time using a Lagrange multiplier optimization method [49]. The time axes for each trajectory were shifted and scaled by the duration of cycle 1 to obtain maximum overlap between all sets of trajectories. Cycle nth is the time period demarcated by the (2n-1)th and (2n + 1)th zero crossings of U_1, respectively in decreasing orange hue.

4.2 Multi-Dimensional Scaling (MDS)

4.2.1 Mathematical definition of MDS

Multi-dimensional scaling (MDS) analysis [50, 51] is used to find a spatial configuration of objects when some metric of (dis)similarity is known. In other words, MDS gives the number of dimensions that are required to visualize similarities or dissimilarities (distances) between the objects. For example, suppose I take a matrix of distances between major US cities. As a result of the MDS analysis on this matrix, would result in a two-dimensional representation of the locations of the cities, that is, a two-dimensional map.

The input into MDS analysis is a dissimilarity or distance matrix d_{ij}, given by **equation(4.1)** , which is the pairwise distances between *n* objects. d_{ij} is the Euclidean distance between ith and jth trajectories. The MDS algorithm will find a configuration of objects in *p*-dimensional space such that the coordinates (*x*) of the *n* objects along the *p* dimensions yield a Euclidean distance matrix whose elements are as close as possible to the elements of the given distance matrix d_{ij} (see **equation (4.2)**)

$$d_{ij} = \begin{pmatrix} d_{11} & d_{22} \rightleftharpoons & d_{1n} \\ \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} \cdots & d_{nn} \end{pmatrix} \tag{4.1}$$

$$|| x_i - x_j || \approx d_{ij} \quad for\ all\ i, j \in 1.....n \tag{4.2}$$

Once I know how many dimensions (p) are required to visualize the configuration of ***n*** objects, the relative error is calculated between our input dissimilarity matrix *d* and the MDS-reconstructed dissimilarity matrix *d'* (see **equation (4.3) .**) $d_{ij}^{\prime 2}$ is the reconstructed dissimilarity matrix of pairwise distances between the obtained configuration of points x_1, x_2, x_3 ….. x_n from analysis. The maximum relative error is calculated as a function of number of dimensions.

$$Max.\ relative\ error = \frac{d_{\alpha\beta}^2 - d_{\alpha\beta}^{\prime 2}}{d_{\alpha\beta}^{\prime 2}} \tag{4.3}$$

4.2.2 MDS of trajectories from postural space

The goal of this chapter is to quantify the behavioral variability among oscillation cycles of trajectories embedded in the postural space for an ensemble of fish. MDS analysis was applied to trajectories in postural space to see if they could be mapped onto a lower dimensional 'behavioral space' that could more readily demonstrate variation in fish swimming behavior across trajectories. As explained above, the first step in MDS analysis is to find the dissimilarity or distance matrix. The dissimilarity matrix $d^2_{\alpha\beta}$ was calculated between trajectories for each oscillation cycle (see **Figure 4.1**) from the Euclidean distance between pairs α, β of time-aligned spine bend angles $\Delta\theta_\alpha$ and $\Delta\theta_\beta$. The dissimilarity matrix can be expressed in terms of the amplitudes U_1, U_2, U_3 as in **equation (4.4).**

$$d^2_{\alpha\beta} = \sum_i \sum_{k=1}^{3} S^2_{kk} \left[U^\alpha_k \left(\tilde{t}^\alpha_i \right) - U^\beta_k \left(\tilde{t}^\beta_i \right) \right]^2 \qquad (4.4)$$

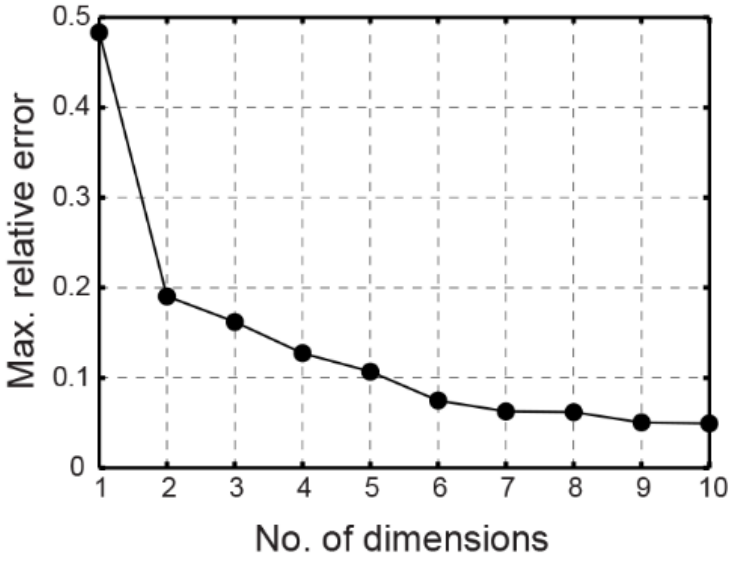

Figure 4.2: Maximum relative error. Maximum relative error between the dissimilarity matrix $d^2_{\alpha\beta}$ (see **equation(4.4)**) and the MDS-reconstructed dissimilarity matrix $d'^2_{\alpha\beta}$, plotted as a function of number of MDS dimensions. The maximum relative error was defined as $\left(d^2_{\alpha\beta} - d'^2_{\alpha\beta}\right) \Big/ d'^2_{\alpha\beta}$.

For the derivation of **equation (4.4)** see **Appendix B**. **Figure 4.2** shows that three-dimensions are required to visualize ~85% of behavioral variability in an ensemble of trajectories (not to be confused with the three axes of postural space).

4.3 Behavioral space

Behavioral space is the low-dimensional space where each oscillation cycle of a trajectory in postural space is embedded as a single point, and the distance between two points represents how similar the pair is. In this section, a new classification of behavior using the concept of behavioral space is generated and discussed. Behaviors of two sets of younger (7-8 dpf) and older larvae (9-10 dpf) are compared to determine if age alters swimming behavior significantly.

4.3.1 Visualizing fish swimming in behavioral space

Figure 4.3 and **Figure 4.6** show fish swimming trajectories plotted in MDS dimensions 1-2 and 2-3, respectively. Each point in behavioral space is from an oscillation cycle of a trajectory from younger (7-8 dpf) or older larvae (9-10 dpf). Points are coded as follows: filled circles are younger (7-8 dpf) and open circles older larvae (9-10 dpf). Color corresponds to position in the behavioral space and time (i.e. oscillation cycle labeled in **Figure 4.4A-B**): the green hue represents position along MDS dimension 1 (< 0 for less green, > 0 for more green), red hue to dimension 2 (< 0 for less red, > 0 for more red), and blue hue for cycle number (oscillation cycle 1, 2, 3…, for less blue to more blue).

4.3.2 Classification of behavior

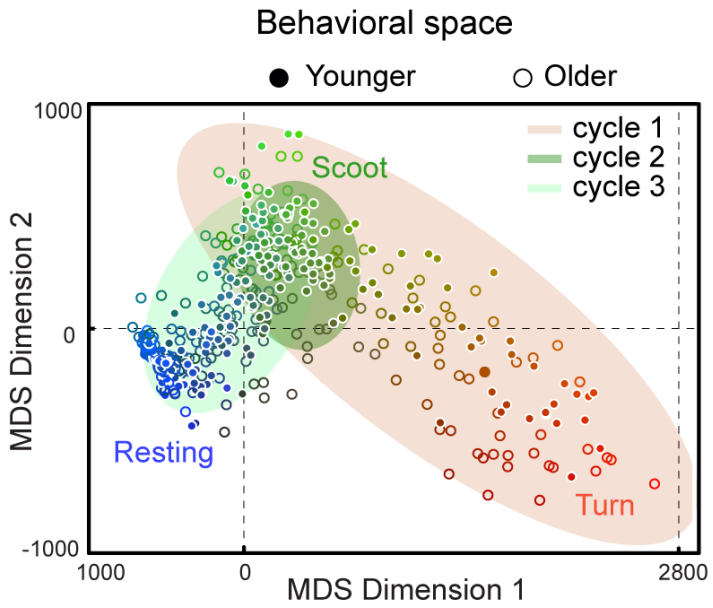

Figure 4.3: Classification of behavior in behavioral space. Ensemble of swim bout trajectories (*N* = 115) embedded by multi-dimensional scaling (MDS) into low-dimensional "behavioral" space. Each cycle of every trajectory is represented by a single point (filled circles for younger larvae (*N* = 8; 7-8 dpf), open circles for older larvae (*N* = 12; 9-10 dpf). The distances between points reflect their behavioral similarities/differences. MDS dimensions 1 and 2 reveal behavioral regions corresponding to turns (red points), scoots (green) and rests (blue). The shaded ellipses demarcate the points from each cycle of the trajectory.

The behavioral space in **Figure 4.3A** represents graphically both the temporal sequence of patterns during a swim bout and the variability in behavior between bouts. The first oscillation cycle (points under red shaded area) captures most of this variability, spanning almost the full range of MDS dimensions 1 and 2. MDS dimensions 1 and 2 correlate well with turning and scooting, with turn-like trajectories localized at large positive values of MDS dimension 1 and scoot-like trajectories at small values. In contrast, the second oscillation cycle (points under the green shaded area; see also **Figure B.3** in **Appendix B**) displays much less variability, with all points localized in one region. This cluster occupies the same region of behavioral space as the scoot-like points in cycle 1, indicating that fish undergoing a scoot simply repeat the same pattern of postures over multiple cycles. Turning fish, on the other hand, generate a different oscillation cycle 1 before returning to the same scoot-like cycle. Successive oscillation cycles (green, purple, blue, and gray shaded areas; see also **Figure B.3** in **Appendix B**) either overlap with cycle 2 (i.e.

scoots) or shift to a third locus of points close to the origin in behavioral space. Inspection of these movies and trajectories indicates that the latter correspond to “terminated” cycles, i.e. oscillations decaying to the resting behavior.

In **Figure 4.4,** the amplitudes of U_1, U_2, U_3 of the three eigenshapes V_1, V_2, V_3 are plotted for every recorded swim bout and colored according to their locations in behavioral space (red for turns, green for scoots, blue for resting). It is interesting to note that most of the fish-to-fish variability appears in U_3 during the first cycle. Although the contribution of U_3 to the zebrafish shape is less than the other two modes (<5%, see **Figure 3.2**), it is essential to quantify fish-to-fish variation. In **Figure 4.5A**, the same set of trajectories is plotted in postural space.

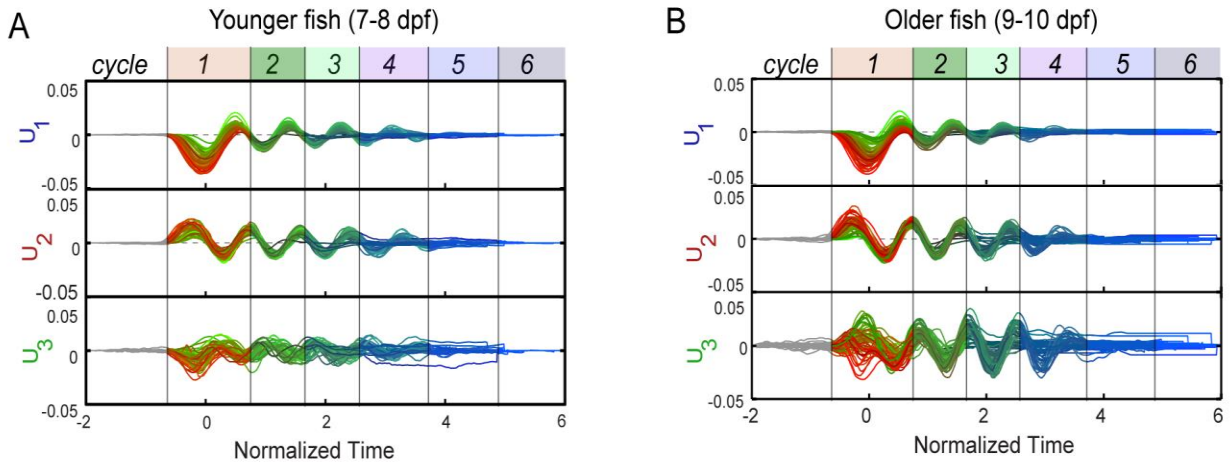

Figure 4.4: Amplitudes U_1, U_2, and U_3 vs. normalized time of the trajectories. (A) Amplitudes U_1, U_2, and U_3 vs. normalized time of the trajectories in **Figure 4.3** from younger larvae. (B) Same plot for older larvae. Shaded areas demarcate each cycle of the trajectory.

As for the single trajectories in **Figure 3.6C** and **Figure 3.7C** the postural spaces in **Figure 4.5A-B** show some trajectories (red) with excursions along U_1–U_3 (a so-called "turn region") and others (green) lying mostly along the U_2–U_3 plane (a "scoot region").

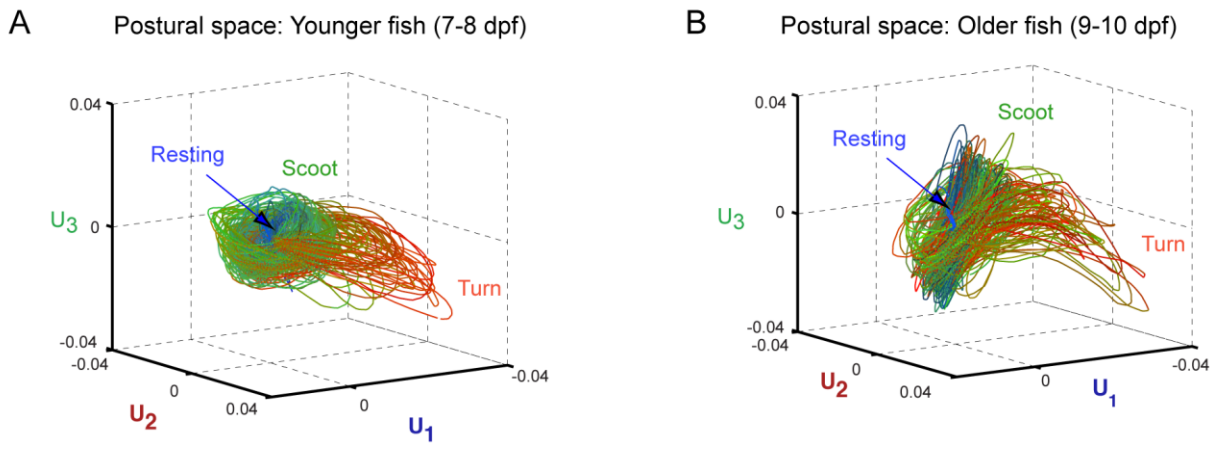

Figure 4.5: Postural space. (A-B) Postural space representation of the trajectories in **Figure 4.4A** and **Figure 4.4B** respectively. Throughout, color represents the location in the behavioral space and cycle number: in the RGB colormap, red and green channels correspond to position along MDS dimension 1 and 2, respectively, and the blue channel to the cycle number. In A and B, cycle number is also represented by symbol size (large to small corresponding to cycle 1 to 6).

These observations confirm the qualitative picture formed from individual trajectories and offer a new way of classifying behavior. Every swim bout can be decomposed into a set of cycles where every cycle represents a behavioral pattern, which lie on the continuum between the turn-like and scoot regions or the scoot and resting regions (see **Movie D1.0.4** and **Movie D1.0.5**).

One more interesting observation came out from behavioral space that trajectories from individual fish revealed the same behavioral variability within the bouts of the single fish as for the population, and there was no strong clustering of the single-fish trajectories in behavioral space (see **Figure B.4** in **Appendix B**).

4.3.3 Classification of fish by age

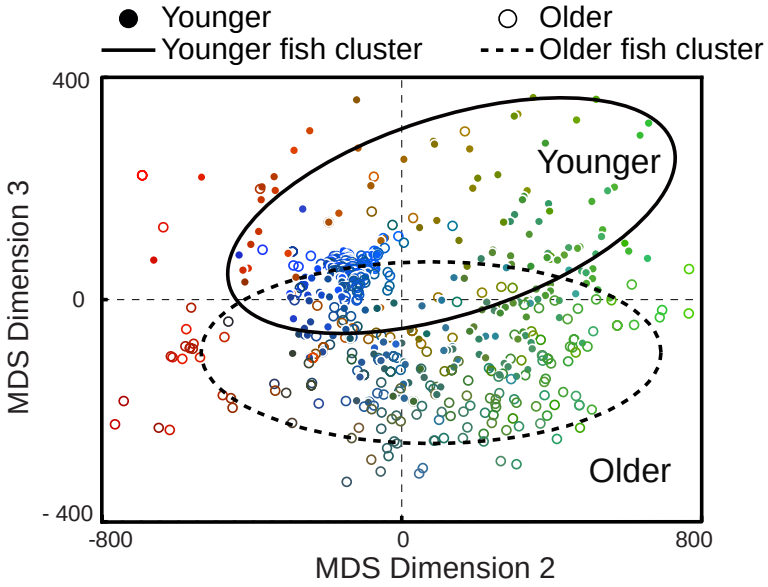

Figure 4.6: Classification of age in behavioral space. MDS dimensions 2 and 3 depicting the differences in zebrafish age (solid ellipse for younger larvae; dotted ellipse for older larvae).

The behavioral space also revealed differences between age groups. **Figure 4.6** shows swimming trajectories projected onto MDS dimensions 2 and 3. Trajectories from younger (filled circles) and older (open circles) larvae occupy separate regions. The changing with age is most prominent in cycles 1 and 2 (see **Figure B.3A in Appendix B**). Inspection of **Figure 4.5A-B** and **Figure 4.6A-B** show that older fish display more variation in U_3 in cycle 1 compared to younger fish; their turn interval (red trajectories) is considerably more bent than that of the younger larvae, indicating a systematic difference in shape. Nevertheless, although fish of different age seem to adopt different postures, they undergo the same temporal sequence of patterns in the behavioral space (**Figure 4.3**.)

4.4 Clustering of data in behavioral space

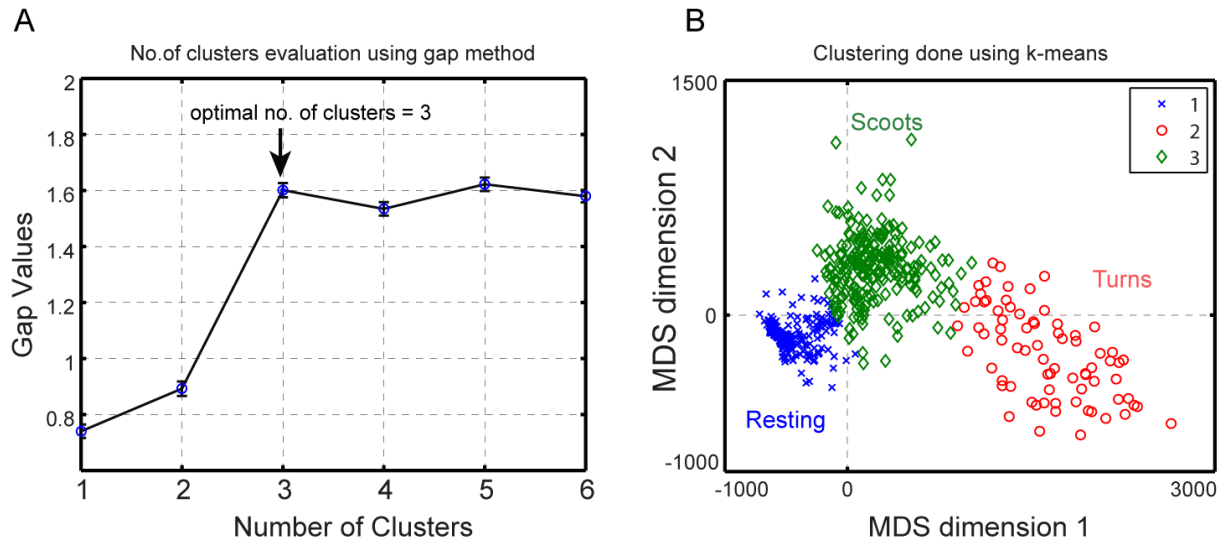

Figure 4.7: Clustering analysis of zebrafish swimming behavior. (A) *K*-means clustering of the trajectories in **Figure 4.3**. The optimal number of clusters were evaluated using the "gap" method, which searches for gaps in the data. The gap value criterion reaches a maximum at three, showing that the data are best described by three clusters. (B) Plot of behavioral space data from **Figure 4.3** clustered into three groups, corresponding roughly to turns (red circles), scoots (green diamonds), and rests (blue x's).

The analysis from **section 4.4** (see **Figure 4.3**) suggests each cycle of a swim bout can be classified according to regions in the behavioral space. I applied a *k*-means clustering [52] algorithm to the data in **Figure 4.3** and a gap method [53] to evaluate the optimal number of clusters in behavioral space. The gap method shows the behavioral space has 3 clusters as shown in **Figure 4.7A**. **Figure 4.7B** shows that postural cycles are optimally grouped into three clusters corresponding to turn-like behavior, scoot-like behavior, and to the resting behavior. The first two groupings partially overlaps with classifications of scoots and R-turns based on bend angle (<40° and >40° respectively) used in previous studies (see **Figure 4.9**) However, it is important to note that the clustering is weak, with no strong boundaries in the data, and points occupying regions of behavioral space between these groupings. **Figure 4.8** shows the probability distribution of trajectories in behavioral space calculated using kernel density estimation method. The distribution reveals the presence of two strong peaks: 'scoot' and 'resting' whereas turn appears as a tail of scoot peak.

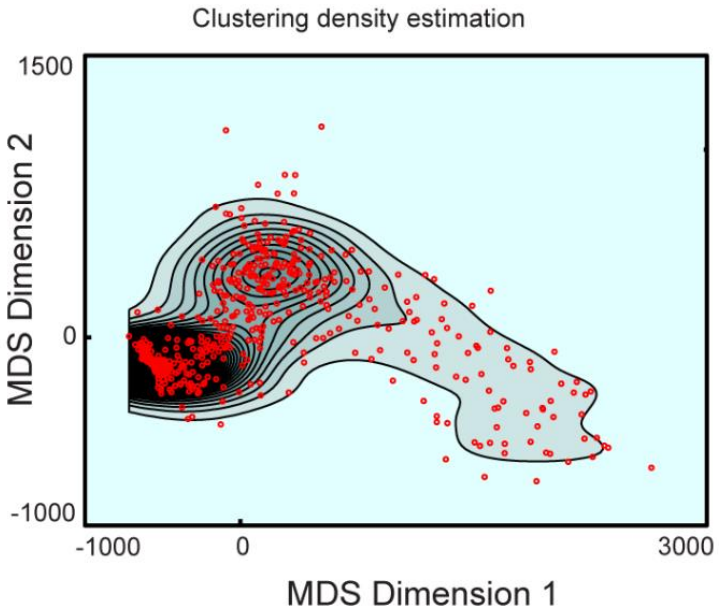

Figure 4.8: Clustering of trajectories on behavioral space. *K*ernel density estimation [54] of trajectories in **Figure 4.3**. It shows that turns are tail from scoot cluster

4.5 Clustering of data using previous methods

Previous studies have used measurements of the fish head angle and fish bend angle during the first oscillation cycle to classify swimming behavior. To see how our new classification method compares, I used the coloring from MDS (described in **Section 4.3.1**) and plotted the fish bend angle vs. fish head angle in first cycle to look for any correlations (**Figure 4.9A**).

Figure 4.9A shows that large values of fish bend angle in red shade (that corresponds to large values of MDS dimension 1) correlate well with turning. In contrast, scooting corresponding to small values of MDS dimension 1 in green shade correlate well with small values of fish bend angle location along MDS dimension 1 and 2, respectively. Green trajectories have lower bend and head angles, corresponding to scoots, whereas red trajectories have larger bend and head angles, corresponding to turns. **Figure 4.9B** shows the behavioral space first cycle where each trajectory is classified as scoots (<40°, magenta) and turns (>40°, black) based on the fish bend angle parameters.

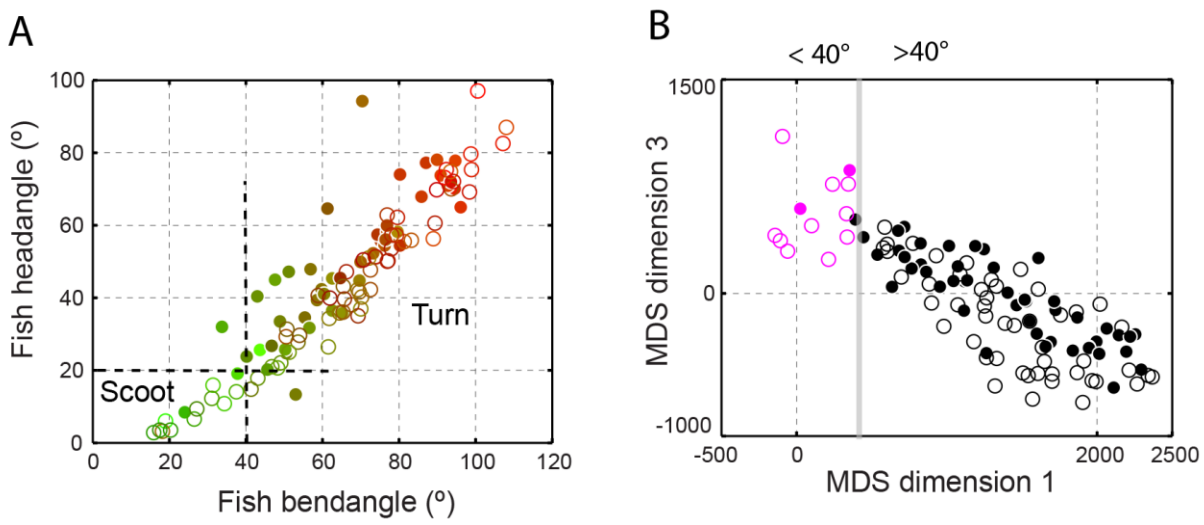

Figure 4.9: Previous classification of behavior and multi-dimensional scaling of zebrafish trajectories. (A) Plot of fish bend angle vs. fish head angle calculated using the method described in [22] and correlation with MDS of zebrafish trajectories. Each point represents the first cycle of a trajectory. The color of each point represents the location of the trajectory in behavioral space (as in **Figure 4.3**), with red and green corresponding to location along MDS dimension 1 and 2, respectively. Green trajectories have lower bend and head angles, corresponding to scoots, whereas red trajectories have larger bend and head angles, corresponding to turns. (B) Plot of the first cycle of trajectories in MDS dimensions 1 and 2 and correlation with fish bend angle. Trajectories classified as scoots (<40°, magenta) and turns (>40°, black) based on the fish bend angle parameters. Throughout, open circles represent older larvae (9-10 dpf), filled circles younger larvae (7-8 dpf).

4.6 Discussion

As shown in **Figure 4.3** the behavioral space described by all collected zebrafish trajectories spans at least three-dimensions. These coordinates not only capture the differences between turning and scooting bouts but also reflect unexpected differences in postures between younger and older organisms. The behavioral space in **Figure 4.3** helps us view swimming bouts as a sequence of postural cycles in time. During a scooting bout, the fish repeats the same cycle of postures until eventually coming to rest. This contrasts with a turning bout, during which the fish adopts a different set of shapes in its first cycle, but then returns to the same postural cycles of the scooting bout. The behavioral space also reveals the continuum and variability of swimming behavior. There is a significant amount of variation in trajectories within the repertoire of the single organism, similar to that between different organisms. Although the behavioral

space suggests clustering into three behavioral patterns corresponding to turns, scoots, and rests, these are not necessarily discrete, in the sense that motion between stereotyped behaviors can and does occur. In agreement with [22], where two broad overlapping peaks were found in the head angle distribution for turns and scoots, I observe a continuum of motion bounded by extremal swimming motions.

Chapter 5. Escape response of zebrafish larvae

In previous chapters, I have described the parameter-free analysis of free-swimming behavior of zebrafish. The primary purpose was to find the new coordinates to project free-swimming behavior of zebrafish. The parameter-free analysis of free-swimming zebrafish movies produces the three-dimensional space whose axes are eigenshapes and coordinates are amplitudes of eigenshapes. Any sequence of a free-swimming fish can be visualized temporally in this 3D space called 'postural space'. After establishing the robust parameter-free method to visualize the free-swimming behavior in a low-dimensional space, free-swimming trajectories were quantified to study behavioral variation among an ensemble of fish. The concept of behavioral space was introduced to classify the behavioral variation using a *priori* classification. In this chapter, I have used a similar approach to quantify the escape response of acoustically stimulated larvae. The escape response in a larval zebrafish can be elicited by three ways: 1) by a brief pulse of water puffed out of a pipette positioned near the head or the tail of fish, 2) a short pulse of sound wave causing vibrations in the water and 3) by tactile stimuli, using a horse hair to touch the tail or head of the fish. This chapter discuss the escape response caused by a short pulse of sound.

5.1 Introduction

Zebrafish has specialized bones known as Weberian ossicles [55] that connect the swim bladder to the sacculi (auditory system of zebrafish). Recent evidence shows that zebrafish have a hearing range of 100-4000 Hz [28, 39, 56, 57], and are most sensitive at 800 Hz.

Previous studies have classified escape responses into behavioral patterns: 'C-turn' and 'burst swims' [17, 28, 58]. C-turn is characterized by high angular velocity and are usually divided into two stages: an initial C-bend in the backbone away from the source of stimuli followed by one or more left and right movement of tail. Burst swimming pattern exhibits larger bend angles and higher speeds than scoots, as well as a bend location is closer to the middle of the body.

Previous classification use parameter such as angular velocity to differentiate between C-turn (>13° ms^{-1}) and free-swimming turns (<13° ms^{-1}) [25]. Angular velocity is the change in orientation of the larva during a turn divided by the duration from the start of the turn to maximal bend of the turn. The parameter 'initial bend duration' is defined as the time between the beginning of the turn and the point at which the bending of the larval zebrafish is maximal. The initial bend duration of escape turns is 6–14 ms, whereas routine turns have a duration of 24–34 ms. Bursts swims are differentiated from slow swims by the bend location in the frame with maximum bend. Bend locations of 0.7 or more body lengths caudal were classified as slow swims, while episodes with a more rostral bend location were classified as burst swims.

The limitation with this approach is it has new set of pre-defined parameters to differentiate the escape response and free-swimming behavior of a zebrafish larva. As described in **Chapter 1** head angle and bend angle parameters classify free-swimming behavior into turns and scoots. In contrast to angular velocity, initial bend duration classifies escape response behavior into 'C-turn' and 'burst swims or fast scoots'. The goal of this chapter is to use similar parameter-free analysis as described in **Chapter 2**, **3** and **4** to analyze escape response behavior of zebrafish.

This chapter aims to answer the following questions: Do we need new or more coordinates to see acoustically stimulated swimming trajectories in the postural space? Do acoustically stimulated behaviors clusters separately from swimming behavior on the behavioral space?

5.2 Experimental procedures

5.2.1 Zebrafish handling

All experiments were performed on zebrafish (*Danio rerio*) larvae age 9-10 days post fertilization (dpf). The larvae were obtained from breeding wild-type zebrafish adults. Adult zebrafish were kept in a Z-hab mini system (Aquatic habitats, Beverly, MA) fish facility at 28.5 °C on a 14h:10h light:dark cycle. Embryos were obtained from adult fish breeding and were raised at 28.5 °C in 10% Hanks solution [8]. Animals were treated in accordance with the guidelines

approved by the University of Illinois Institutional Animal Care and Use Committee protocol # 13327. For breeding procedures, see **Appendix A**.

5.2.2 Behavioral experiment

The video acquisition set up to capture the escape response of larvae was similar to that used for studying free-swimming (for details see **Chapter 2**.) For each experiment, a single 9-10 dpf fish larva was placed in a well of 6 well plate in 10% Hanks solution at room temperature. The larvae were illuminated from the bottom with IR LED (SKU: T17712) array of wavelength 830 nm. The loudspeaker was positioned right below the well where larva was placed (see **Figure 5.1**.) A custom-designed program was created in MATLAB (Mathworks) to control the USB interface loudspeaker with an output range 90 Hz-20 kHz (Logitech S-160). Acoustic stimuli consisted of sinusoids of 500 Hz frequency and 20 ms duration. The movies of escape response of larva were recorded at 1000 fps using a high-speed camera (Diagnostic Instruments) and the Motion Studio Software Suite. The larva was startled 2 seconds after beginning recording. Each video had 1 escape response event of duration ~200 ms. The number of escape events (N = 31) was from 5 zebrafish larvae.

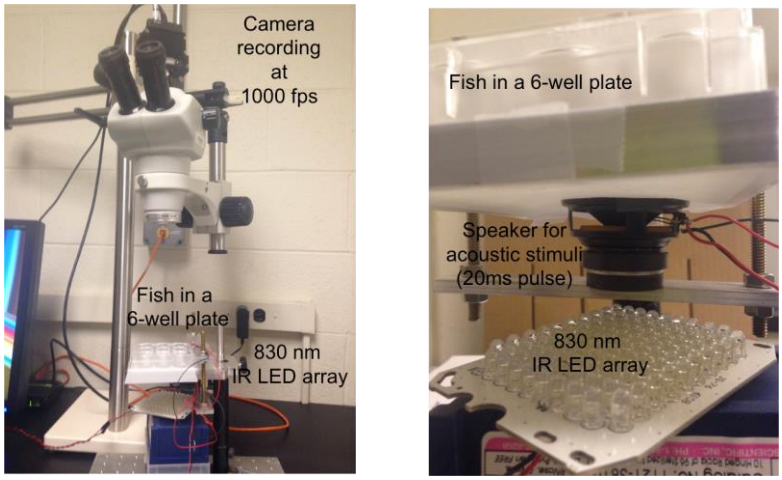

Figure 5.1: Video acquisition setup to startle larvae acoustically. The figure in left panel shows a high speed camera mounted to a stereo microscope to image escape response of larvae swimming in a well of a 6 well plate. For acoustic stimuli, a USB interface loud speaker with an output range 90 Hz-20 KHz was placed below the 6 well plate (right).For IR illumination, a LED array of wavelength of 830 nm was placed below the 6-well plate. Video acquisition was done using the IDT camera software installed on lab computer.

5.2.3 Image analysis

After video acquisition, the videos were preprocessed to convert each movie frame into a set of fish spine angles from head to tail using the similar recipe described in **Chapter 2**.

The first video preprocessing step 'video segmentation' step was skipped because each video had one escape response event. This step is used to segment free-swimming videos into many swim bouts. After this movie frame segmentation, fish backbone digitization as described in **Chapter 2** was followed to convert every frame into a set of fish spine angles $\Delta\theta\left(s_j, t_i\right)$.

5.3 Parameter-free analysis

The preprocessing of escape response movies converted each movie frame into a set of fish spine angles $\Delta\theta\left(s_j, t_i\right)$ from head to tail. The SVD input matrix has N = 115 free-swimming trajectories and N = 31 escape response trajectories. The free-swimming and escape response $\Delta\theta_{FreeSwim+Escape}\left(s_j, t_i\right)$ data sets were analyzed together to have one common set of coordinates to describe these behaviors.

$$\Delta\theta_{FreeSwim+Escape}\left(s_j, t_i\right) = \sum_{k=1}^{n} U_k(t_i) S_{kk} V_k(s_j) \qquad (5.1)$$

Each $U_k(t_i)$ represents the amplitude of the k^{th} basis function $V_k(s_j)$ at each time point t_i. S_{kk} is an n × n diagonal matrix of singular values. As discussed in **Chapter 3** the singular values were determined by first sorting them from high-to-low singular values, with the highest singular value in the upper left index of the *S* matrix.

Performing SVD on all free-swimming movies (*N* = 115) from a population of 20 fish and escape response movies (N = 31) from 5 fish reveals that 94% (see **Figure 5.2B**) of the variation in $\Delta\theta$ is accounted by the first three eigenshapes only. The eigenshapes shown in **Figure 5.2A** were obtained by analyzing all movies together.

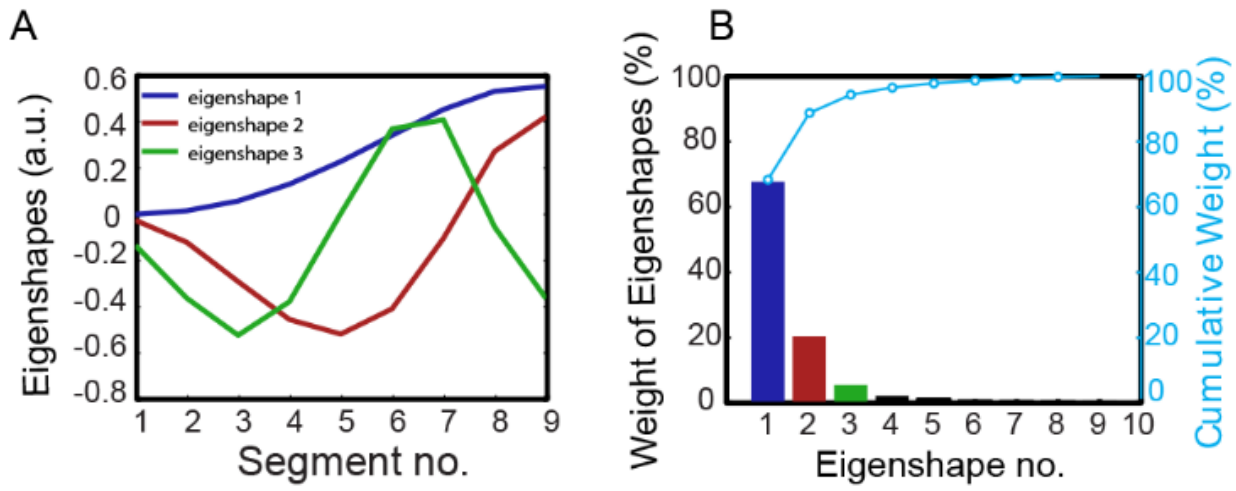

Figure 5.2: SVD of zebrafish spine angles of free-swimming and escape response data sets. (A) Singular value decomposition of $\Delta\theta$ into eigenshapes $V_k(s_j)$ (j = 1, 2... 9), the first three of which are plotted (red, blue, and green, respectively). These are the collective eigenshapes from analyzing free-swimming and escape response data at once. (B) Bar plot of % weights ($S_{kk}/\sum_{k'=1}^{10} S_{k'k'}$, where S_{kk} are the singular values) of each eigenfunction $V_k(s_j)$. The right axis in cyan shows the cumulative contribution of each eigenshape. The first three eigenshapes contribute 94% of the total variance in $\Delta\theta$.

5.4 Postural space

The next step was to visualize free-swimming behavior and escape response in the same coordinate space. I have used the robust 'postural space' which is a three-dimensional space where coordinates are eigenshapes (V_1, V_2, V_3) and are sufficient to capture >94% of the observed free-swimming and escape responses of zebrafish.

5.4.1 Visualization of free-swimming and escape responses of a population of fish

I took all the free-swimming (N=115) and escape response (N=31) trajectories from 25 organisms and constructed the postural space to visualize the behaviors using just one set of postural space coordinates. **Figure 5.3A** shows the visualization of postural space of a population of fish. The most important feature to observe in **Figure 5.3A** is that some trajectories have high amplitude in U_3 and U_1 in the first oscillation cycles. These are escape response trajectories of zebrafish larvae. The free-swimming trajectories in magenta and purple color have relatively smaller magnitude of amplitude U_3 and U_1 than escape response trajectories in black. The scoot

or burst swims region of escape response trajectories along U_2 and U_3 have higher amplitude than free-swimming trajectories.

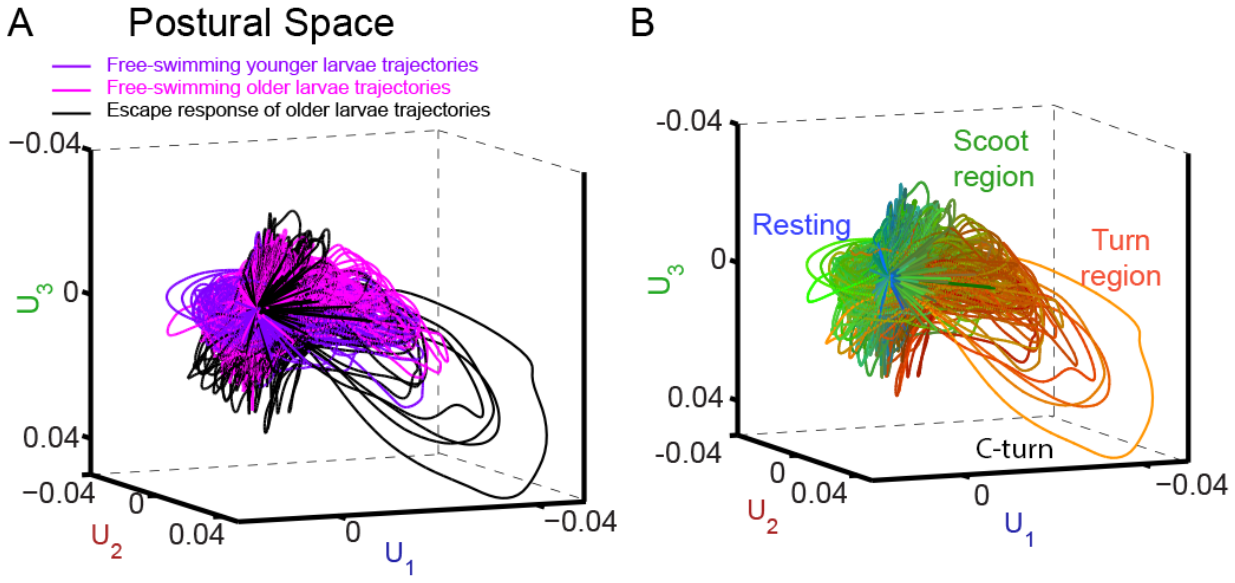

Figure 5.3: Visualization of escape response and free-swimming trajectories of a population of fish in low-dimensional postural space. (A-B) The three-dimensional coordinates of the trajectory are the amplitudes $U_1(t)$, $U_2(t)$, and $U_3(t)$. (A) Visualization of postural space using three colors for each data set: free-swimming younger larvae in purple, free-swimming older larvae in magenta, and escape response of older larvae in black. The oscillation cycles of the escape response have high magnitude along U_3 - U_2 and U_3 – U_1 in comparison to free-swim trajectories. (B) Visualization of postural space using color representing the location in the behavioral space and cycle number. In the RGB colormap, red and green channels correspond to position along MDS dimension 1 and 2, respectively, and the blue channel to the cycle number. The first oscillation cycle of trajectories along the bent ellipse in orange shade are escape response of larvae.

Figure 5.3B shows the postural space using the color map from behavioral space (discussed in next section). The escape response and free-swimming trajectories are shown in orange and red, respectively. The trajectory with highest amplitude of U_3 and U_1 was identified as a C-turn behavioral pattern. The identification was done using the previous method (see **section 5.1**) and using visual inspection of movie frames corresponding to the trajectory. **Figure 5.4** shows the representative C-turning bout with C-turn region (45 ms movie frame) and burst swim region (65-150 ms frames)

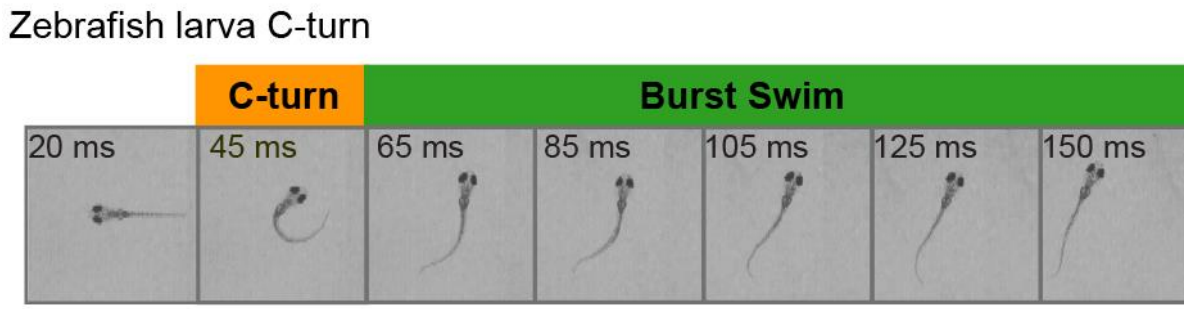

Figure 5.4: Escape responses of zebrafish larva. Still images of a representative C-turn region and burst swim during escape response recorded at 1000fps.

The trajectories with oscillation cycles in orange shade in postural space appears as a continuum with oscillation cycle in red shade from free-swimming behavior. This observation raises the question: Are C-turns from escape response and turns from free-swim separate behaviors?

5.5 Classification of behavior

To quantity the behavioral variation among free-swimming trajectories and escape response trajectories, I did MDS analysis as describe in **Chapter 4**.

5.5.1 Behavioral classification using parameter-free analysis

MDS analysis was performed to quantify the behavioral variability among oscillation cycles of free-swimming and escape response trajectories embedded in the postural space for a population of fish (**Figure 5.3**.) As described in **Chapter 4**, MDS analysis converts each oscillation cycle of a trajectory into a single point and embeds it in a low-dimensional space. The input to MDS analysis is the dissimilarity matrix $d^2_{\alpha\beta}$ between trajectories for each oscillation cycle (see **Figure 4.1** for definition of oscillation cycle, and **equation (4.4)** for the equation of the dissimilarity matrix) of postural space from the Euclidean distance between pairs α, β of time-aligned spine bend angles $\Delta\theta_\alpha$ and $\Delta\theta_\beta$. **Figure 5.5** shows that only 3 dimensions (similar to the

results of MDS on only free-swimming data) capture ~85% of behavioral variability among the trajectories in postural space.

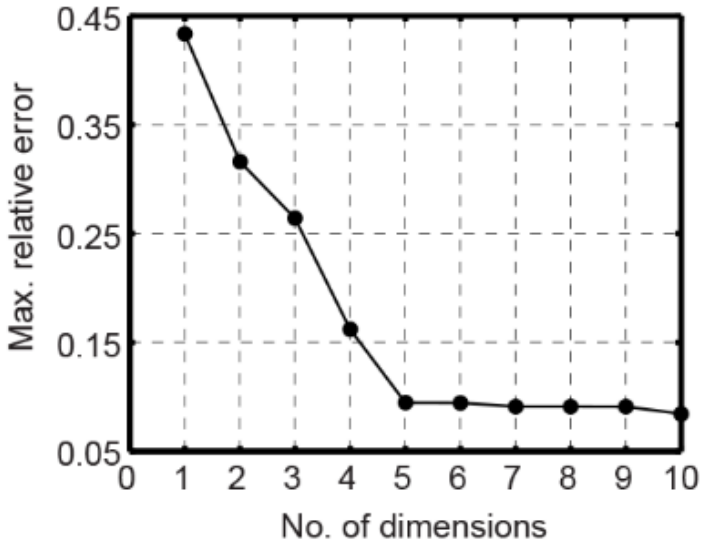

Figure 5.5: Maximum relative error of MDS analysis on free-swimming and escape response data sets. Maximum relative error between the dissimilarity matrix $d^2_{\alpha\beta}$ (see **Eq.(4.4)**) and the MDS-reconstructed dissimilarity matrix $d'^2_{\alpha\beta}$, plotted as a function of number of MDS dimensions. The maximum relative error was defined as $\left(d^2_{\alpha\beta} - d'^2_{\alpha\beta}\right) \Big/ d'^2_{\alpha\beta}$.

5.5.2 Behavioral space of free-swimming and escape response trajectories

Figure 5.6 and **Figure 5.7** show MDS dimensions 1-2 and 2-3, respectively. Each point on the combined behavioral space is from an oscillation cycle of a trajectory from free-swimming younger (7-8 dpf)**,** open circles free-swimming older larvae (9-10 dpf) and open square escape response larvae (9-10dpf). **Figure 5.6A-F** shows the behavioral space resolved by cycle where the color of each trajectory corresponds to the position in the behavioral space and time: the green hue represents position along MDS dimension 1 (< 0 for less green, > 0 for more green), red hue along dimension 2 (< 0 for less red, > 0 for more red), and blue hue for cycle number (oscillation cycle 1, 2, 3…, for less blue to more blue). MDS dimensions 1 and 2 correlate well with C-turn, turning and scooting, with turn-like trajectories localized at large positive values of MDS dimension 1 and scoot-like trajectories at small values. The behavioral space in **Figure 5.6A** shows

the first oscillation cycle of escape response trajectories in open squares. Most of these open squares are positioned around the 'turn-region' of free-swimming trajectories and are not separated as a cluster. However, the number of points is too small to make any strong statistical argument.

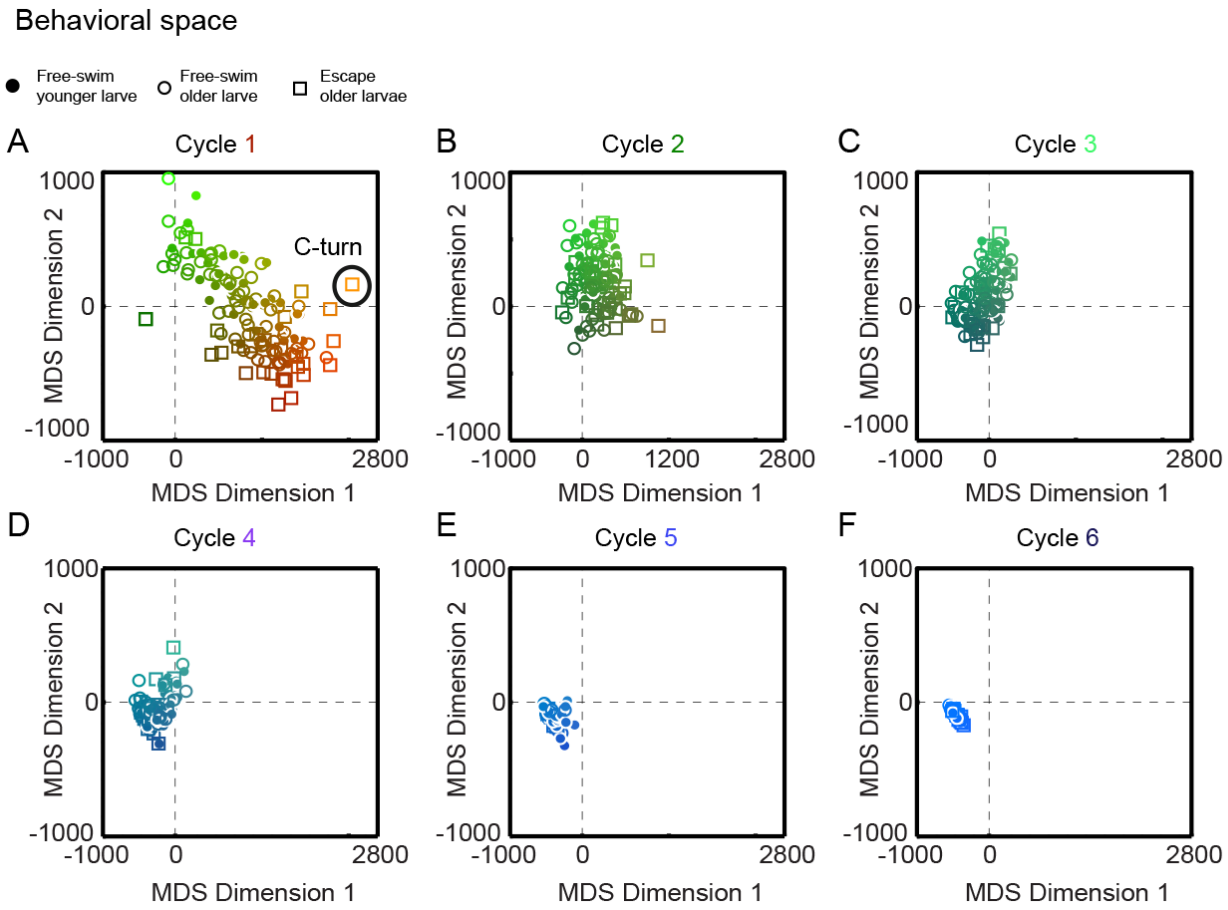

Figure 5.6: Behavioral space of free-swimming zebrafish and escape response of zebrafish resolved by cycle. (A-F) Behavioral space in MDS dimensions 1 and 2 plotting each oscillation cycle separately (top panel for cycle 1, 2, 3) and (bottom left panel for cycle 4, 5, 6.). Filled circles are for free-swimming younger larvae (N = 8; 7-8 dpf), open circles for older free-swimming larvae (N = 12; 9-10 dpf) and open squares for escape response of older larva (N = 5; 9-10 dpf).

5.5.3 Classification by age of free-swimming and escape response trajectories

The behavioral space of combined datasets of free-swimming and escape response of zebrafish larvae also revealed differences between age groups for larvae free- swimming behavior. **Figure 5.7** shows swimming trajectories projected onto MDS dimensions 2 and 3 for

only cycle 1. For older larvae, free-swimming trajectories (open circles) and escape response trajectories (open squares) occupy the same regions and are well separated from the free-swimming trajectories for younger larvae (filled circles.)

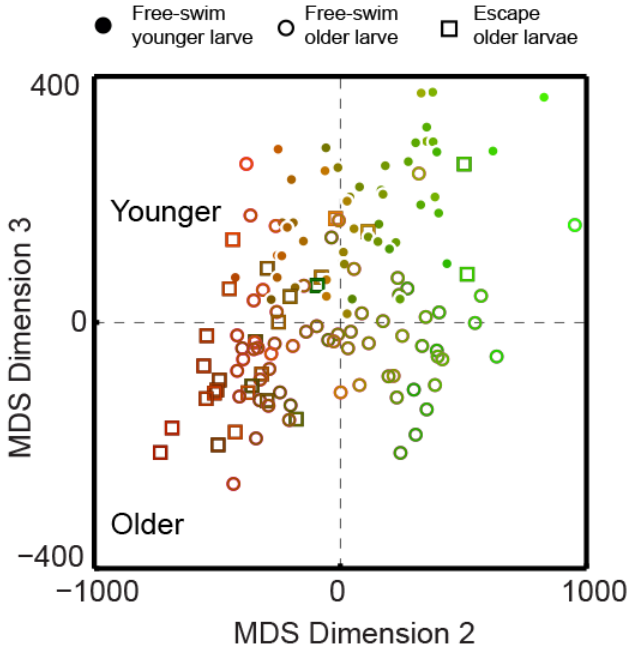

Figure 5.7: Classification of age in behavioral space of free-swimming and escape response of zebrafish. MDS dimensions 2 and 3 reveal age differences in cycle 1. Filled circles are for free-swimming younger larvae (*N* = 8; 7-8 dpf), open circles for older free-swimming larvae (*N* = 12; 9-10 dpf) and open squares are for escape response of older larva (N = 5; 9-10 dpf).

5.6 Discussion

This chapter shows that only three eigenshapes are required to project the free-swimming and escape response of zebrafish larva. The postural space show escape response trajectories with strong C-turn corresponds in higher amplitudes of U_3. However, the number of points (N = 31) is too small to make any strong statistical argument. The escape response trajectories don't appear as a separate cluster but instead form a continuum with free-swim trajectories. This statement needs to be tested with more data collection.

Chapter 6. Neural modeling and simulation of turns and scoots

Previous chapters described a model based on parameter-free analysis. This analysis provides tools to visualize zebrafish swimming behavior temporally in a low-dimensional space and to quantify the variation among behaviors for a population of larvae. With this insight on larval postures and classification of behaviors, I worked on improving existing neuro-kinematic models.

The existing neuro-kinematic model translates artificial neural signals (a series of delta pulses representing action potentials) into observable larval behaviors. To do so, the model transforms the series of delta pulses into perturbations on the radius of curvature of a line segment representing the trunk of the larva. The limitations of this model are following: the model does not represent the full backbone of fish from head to tail and the model does not produce the larva swimming with decreasing tail beat frequency in a swim bout [36].

I constructed a network with separate channels controlling the left and the right sides of the organism, anticipating that signals of identical amplitude to both channels would produce a symmetric fish spine oscillation (i.e. a scoot) while unequal amplitudes would produce a turn in one direction. To test this idea, the new model was optimized to produce scoot and turn bout behaviors. I have also worked on improving the stiffness function in the model which was initially assumed to be linear from head to tail. I used the fish spine angles data from experimental traces to optimize the new stiffness function. In addition to improving the new model, I have used parameter-free analysis to visualize the simulated swimming behavior and the concept of behavioral space to classify the simulated behavior. I have tested the robustness of the model by adding varying levels of white Gaussian noise to the model parameters. It was interesting to see that the swimming behavior of the new neuro-kinematic model was stable against noise as large as 70% and 90% in the amplitude and delay parameters of the neural signal.

6.1 Introduction to the neural model

Figure 6.1 shows the customized version of the neural model of zebrafish. The model divides the fish backbone into a right and left half, each with 10 equal neural segments (s_j) of neurons interconnected by appropriate synapses.

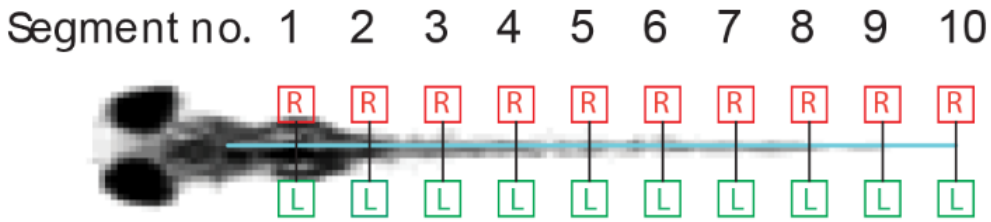

Figure 6.1: A schematic of neural model. This figure depicts the fish backbone divided into ten segments $(s_j,\ j=1,2,3...10)$ on either side of the backbone

The right and left halves each receive an input trigger signal $F_{osc}\left(s_j,t_i\right)$ in the form of a train of sharp pulses. The output of each neural segment along the backbone is convolved with a bi-exponential function representing the temporal response of the neuromuscular junction. The result is the force exerted by the muscles $F_m\left(s_j,t_i\right)$ at each segment on the left and right given by **equation (6.1).** The rise τ_1 and fall time τ_2 of a bi-exponential synapse are 6 ms and 8 ms respectively [36].

$$F_m(s_j,t_i)=\int_{-\infty}^{t} F_{osc}(s_j,t_i^{'})[\exp^{\frac{(t'-t)}{\tau 2}}-\exp^{\frac{(t'-t)}{\tau 1}}]dt' \tag{6.1}$$

$$R(s_j,t_i)=\frac{F_m(s_j,t_i)}{W(s_j)} \tag{6.2}$$

$$\frac{1}{R(s_j,t_i)}=\left|\frac{d\theta}{ds}\right| \tag{6.3}$$

From this force function, I used **equation (6.2)** to determine the radius of curvature of the zebrafish backbone at each segment position, where $W(s_j)$ is the fish stiffness along its spine. Finally I calculated the tangential angles $\Delta\theta(s_j, t_i)$ by integrating **equation (6.3).** The neuro-kinematic model thus produced simulated time traces of $\Delta\theta(s_j, t_i)$ on which I could perform the same parameter-free analysis as explained in **Chapter 2** and **Chapter 3**.

6.2 Simulating turns and scoots

This section discusses the procedure to determine the start value of set of parameters that reproduces the observed behavior.

6.2.1 Defining parameters

The input to the model is the spike train $F_{osc}(s_j, t_i)$: a sequence of alternating left-right delta pulses sent to the brain and stiffness function $W(s_j)$ from head (j = 0) to tail (j = 9). The spike train for larval behavior has four input parameters for optimization: τ_f^l firing time of the spike, amplitudes a_l of each spike, and segment-to-segment delays d_l of each spike and n_c number of alternating left-right movements (or half-cycles). The subscript l = 1, 2 ... n_c is the half-cycle number (see **Figure 6.2A-B**).

The start value of half cycle number input parameter n_c was obtained by determining the number of zero crossings of the tail angle $\Delta\theta(s_9, t_i)$ from the test experimental data. For example, **Figure 6.2A-B** show a 3D plot of $\Delta\theta(s_j, t_i)$ from a turn and a scoot, where n_c = 7 and 8 respectively. Since young fish larvae swim with a decreasing tail beat frequency, the model had the firing times τ_f^l, amplitudes a_l, and segment to segment delays d_l to be functions of the half-cycle number l = 1, 2 ... n_c .

In order to calculate the start value of the model parameters τ_f , I measured the times t_l at which the zero crossings occurred for the head and tail segments s_1 and s_9, respectively. The starting value for the segment-to-segment delay d_l of the l-th half-cycle was:

$$d_l = \frac{t_l(s_9) - t_l(s_1)}{9} \tag{6.5}$$

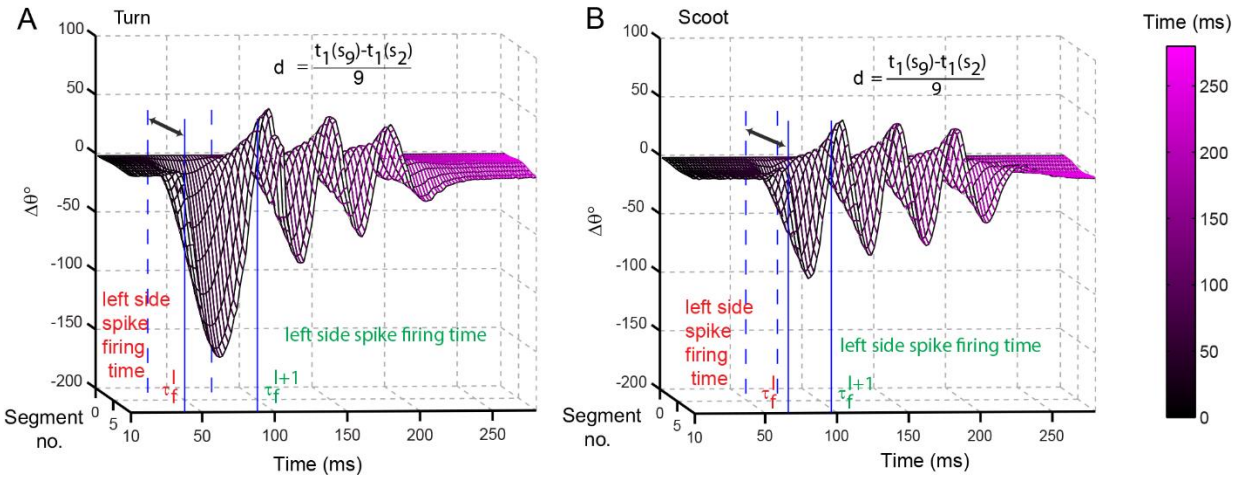

Figure 6.2: Test data traces of a scoot and turn bout. (A-B) Examples of $\Delta\theta(s_j, t_i)$ for a turn and a scoot trajectory, respectively. The zero crossings of $\Delta\theta(s_j, t_i)$ are labeled at j = 1 (close to the head), 9 (tail) in dotted and solid lines, respectively. These are used to calculate the segment-to-segment delay of the neural signal.

Finally, the starting value for the amplitude of the l-th half-cycle was:

$$a_l = \begin{cases} 1 & l \text{ even} \\ -1 & l \text{ odd} \end{cases} \tag{6.6}$$

and the corresponding spike train is:

$$F_{osc}(s_j, t_i) = \begin{cases} a_l & \tau_f^l \\ 0 & \text{otherwise} \end{cases} \tag{6.7}$$

6.2.2 Simulated spike train

The next step was to optimize the model parameters against test data traces of a scoot and turn bout as shown in **Figure 6.2A-B.** The input to the model is the spike train $F_{osc}(s_j, t_i)$ obtained from calculating the start parameters using the experimental data and the stiffness function $W\left(s_j\right)$ The model calculates the fish spine angles $\Delta\theta^{neuro}(s_j, t_i)$ integrating **equation (6.3)** and optimizes the parameters such that the sum of differences squared between the fish backbone angles $\Delta\theta^{neuro}(s_j, t_i)$ generated by the neural network and the fish backbone angles $\Delta\theta(s_j, t_i)$ obtained from the experimental data is minimum. I used a genetic algorithm[59] in MATLAB to run this optimization problem. **Figure 6.3** show the simulated spike train from the model for a turn and scoot bout. **Figure 6.3** show the amplitude of strength of signal decrease with time to generate varying larval tail beat frequency. The simple neuro-kinematic model generates turn-like trajectories by increasing the amplitude *a* to one side of the organism relative to the other and increasing the segment-to-segment delay *d* relative to a scoot during the first oscillation cycle. Thus, despite the high dimensionality of the parameter space of the model, only two parameters are sufficient to generate the observed behavioral patterns.

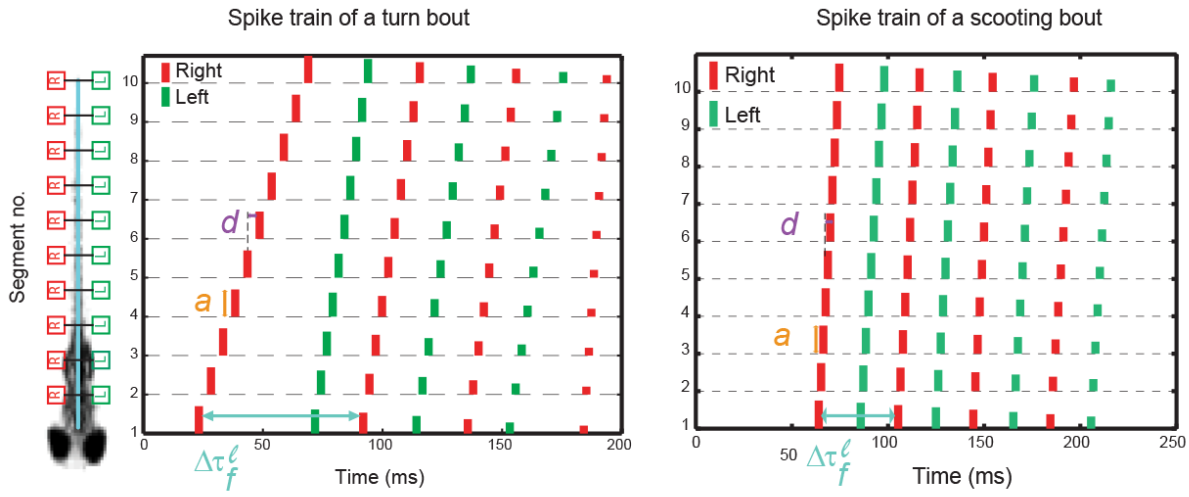

Figure 6.3: Spike train of a turn and scoot swim bout of zebrafish free-swimming. The leftmost panel shows a schematic of neuro-kinematic model depicting the fish backbone divided into ten segments on the right and left sides. Middle and right panels show spike train of a turn and scoot bout respectively generated by an optimized neural model. The right and left spike trains are shown in red and green, respectively. The height of each spike represents the amplitude *a* of the stimulus. τ_f is the firing time for each spike in the head segment. The segment-to-segment delay *d* is the time difference between spikes in adjacent segments of the fish backbone.

6.2.3 Stiffness function

The initial function $W(s_j)$ was chosen to be linear from head to tail. The new stiffness function was optimized at the time of the maximum amplitude of the tail angle $\Delta\theta(s_9,t_i)$ from a turn trajectory. **Figure 6.4** shows the optimized stiffness function, $W(s_j)$ which decreases monotonically from the head toward the tail, indicating greater flexibility towards the tail, then increases again over the last tail segment.

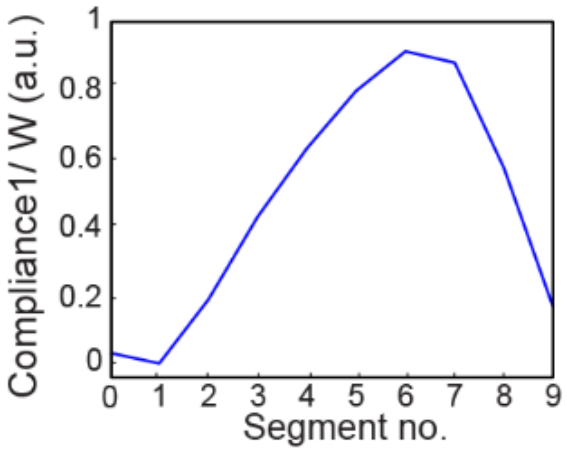

Figure 6.4: Stiffness function. Plot of head-to-tail zebrafish compliance ($1/W(s_j)$) obtained after optimization of neuro-kinematic model against experimental data.

6.2.4 Parameter-free analysis of simulated fish spine angles

The next question is: How robust is the new parameter-free analysis method at describing simulated zebrafish behavior? To answer this, I performed the parameter-free analysis on a simulated set of fish spine angles for a turn and scoot bout $\Delta\theta^{neuro}(s_j,t_i)$. I converted the fish spine angles $\Delta\theta^{neuro}(s_j,t_i)$ into the fish backbone. **Figure 6.5** compares the fish backbone from neural network simulation (red dotted line) is compared with the one from experimental data (cyan).For full demonstration of simulated turn-like and scoot-like behavior see **Movie D1.0.6**. **Figure 6.6** shows that three eigenshapes are again sufficient to describe the neuro-kinematic model's behavior.

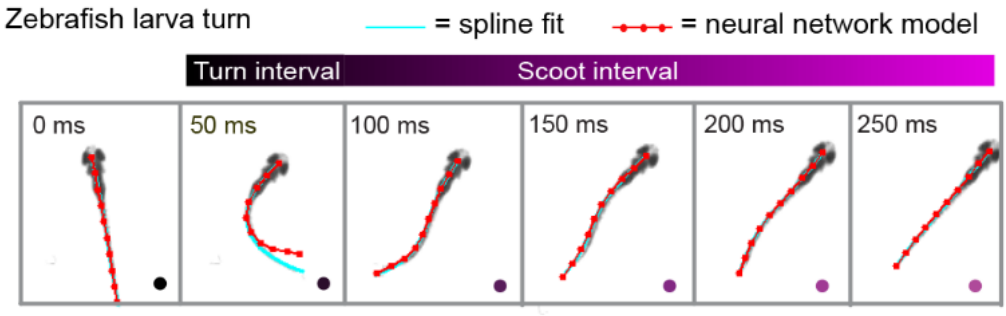

Figure 6.5: Minimal neuro-kinematic simulation of zebrafish free-swimming. (A) Still images from a free-swimming zebrafish movie. Each snapshot shows the fish backbone reduced to a thin skeleton fitted to a cubic spline fit (cyan) and obtained from the neural simulation (red), respectively. The neural model was optimized against the experimental data in **Figure 6.2A**.

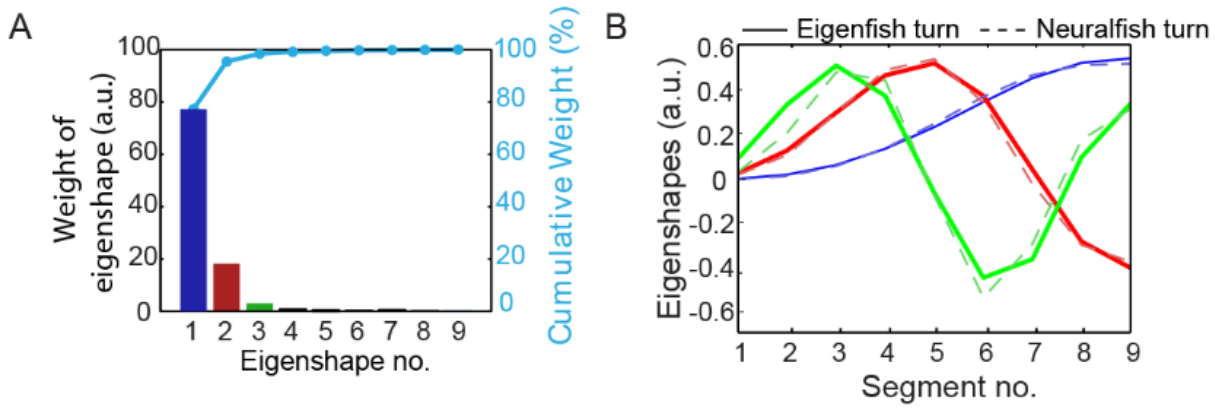

Figure 6.6: SVD of zebrafish spine angles from neural model. (A) Bar plot of % weights of each eigenshapes $V_k(s_j)$ (j = 1, 2... 9) (left axis) and cumulative contribution of each eigenshape (right axis, cyan). The first three eigenshapes contribute 98% in total to the variance in $\Delta\theta$. (B) Decomposition of simulated swimming traces from neuro-kinematic model into eigenshapes $V_k(s_j)$. The eigenshapes from the neural model of a turn (dashed light red, blue, and green lines) match those from experimental data of a turn (solid red, blue, and green)

6.2.5 Postural space and behavioral space of simulated neural model

I constructed the postural space for the simulated fish swimming from the neural network model in **Figure 6.7**. The postural space dynamics for a turning bout has a turn region, a scoot region and a resting point which are similar to the corresponding features in the postural space from free-swimming/turns experimental data traces.

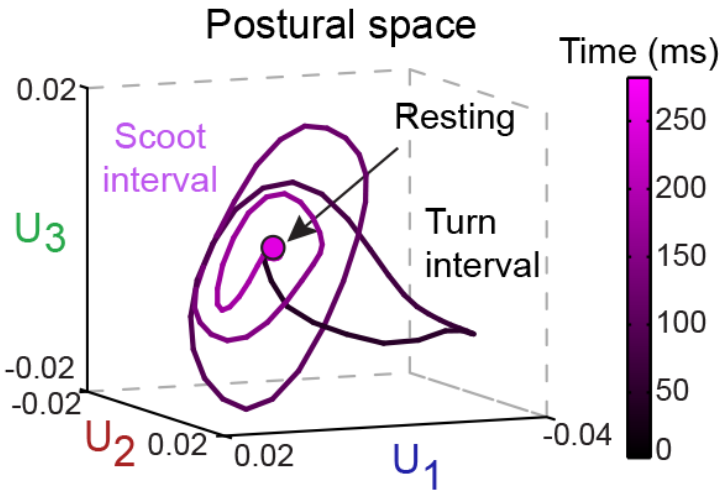

Figure 6.7: Postural space of simulated trajectory from neural model. Simulated neuro-kinematic model trajectory of a turning bout in postural space.

Figure 6.8A shows the behavioral space dynamics of a simulated neural turn and scoot bout that falls closer to experimental turn and scoot bout. This proves that, the behavioral and postural space are robust tools to analyze the swimming behaviors.

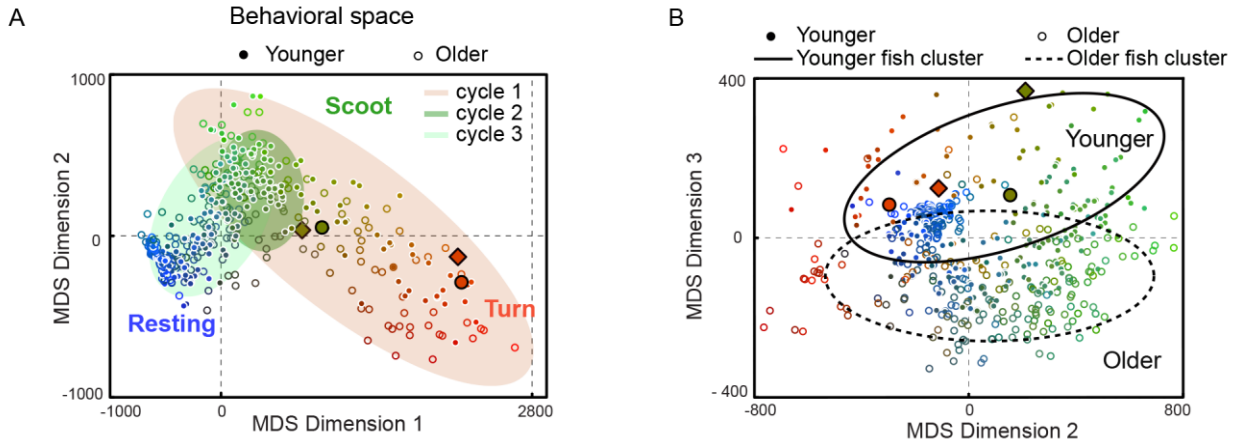

Figure 6.8: Behavioral space of experimental data and simulated neural data. (A–B) Ensemble of swim bout trajectories (*N* = 117) embedded by multi-dimensional scaling (MDS) into low-dimensional "behavioral" space. Each cycle of every trajectory is represented by a single point (filled circles for younger larvae (*N* = 8; 7-8 dpf), open circles for older larvae (*N* = 12; 9-10 dpf)). The distance between points reflect their behavioral similarities/differences. MDS dimensions 1 and 2 (A) reveal behavioral regions corresponding to turns (red points), scoots (green) and rests (blue). The shaded ellipses demarcate the points from each cycle of the trajectory. MDS dimensions 2 and 3 (B) reveal differences with zebrafish age (solid ellipse for younger larvae; dotted ellipse for older larvae). Simulated trajectories from the neuro-kinematic model in **Figure 6.3**, and **Figure 6.2A, B** and its training data are shown in diamonds and circles with black outline, respectively.

6.3 Stability of neural model

I have also tested the robustness of the model by adding varying levels of white Gaussian noise to the model parameters a, τ_f, and d (**Figure 6.9A-C**). For simulation details see **Appendix C.** I found that the swimming behavior of the neuro-kinematic model was stable against noise as large as 70% and 90% in the amplitude a and delay d. Fluctuations in d simply affected the duration of the cycle. The model was most sensitive to noise in the firing time τ_f. Not surprisingly, noise that significantly altered the timing pattern between the left and right sides of the organism affected the cyclic motion required for swimming. A behavioral space was constructed to compare the simulated trajectories with varying signal-to-noise ratio. **Figure 6.9D-F** shows the

scattering of trajectories in behavioral space with increasing signal-to-noise ratio (SNR) for parameter *'a'* and 'τ_f' respectively whereas the trajectories with highest signal-to-noise ratio overlapped with lowest SNR for parameter *'d'.*

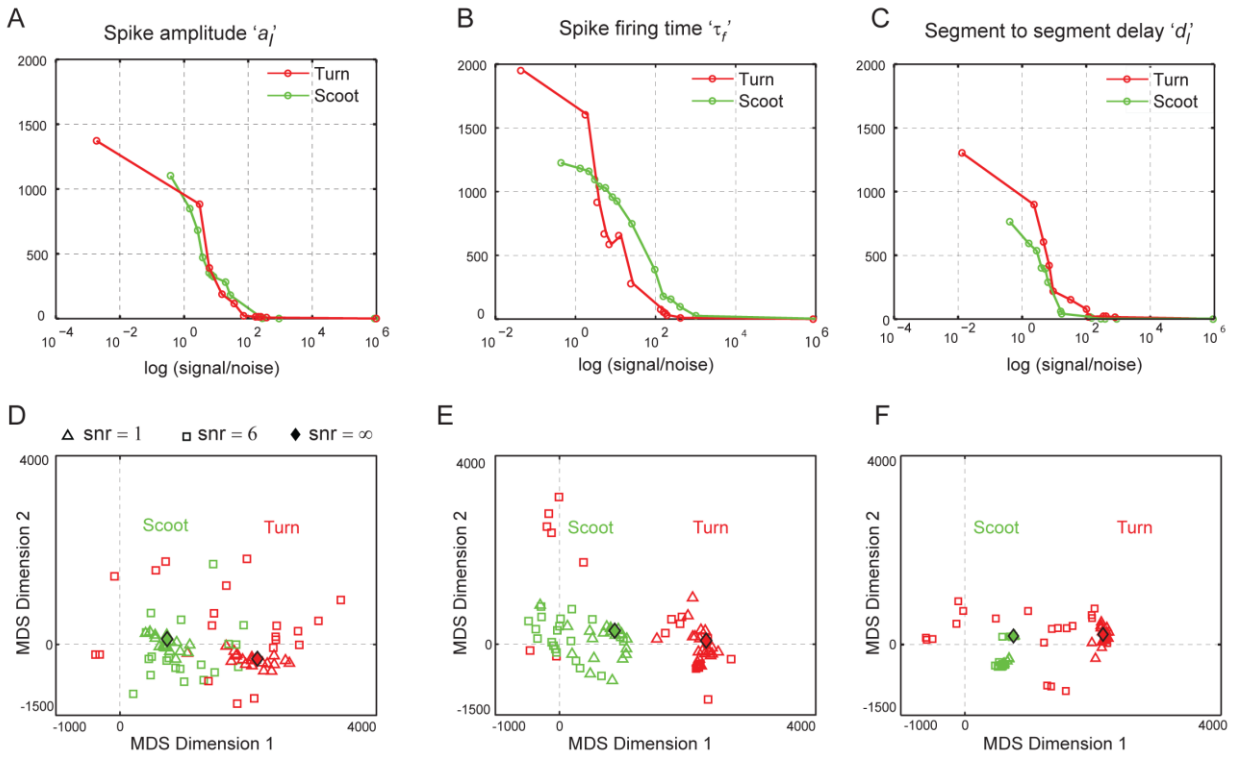

Figure 6.9: Effect of noise on neural model. (A-C) Response of the neural model to white Gaussian noise added to the model parameters $\{a_l, \tau_f^l, d_l\}$, respectively, for models that produce scoots (green) and turns (red). As described in SI Methods, the response is calculated as $d^2_{\alpha\beta(noise)} - d^2_{\alpha\beta(neuro)}$. (D-F) The first cycle of the simulated trajectories with noise added to the parameters $\{a_l, \tau_f^l, d_l\}$, respectively, embedded in the same behavioral space as **Figure 6.8A**. Simulated trajectories with signal-to-noise (snr) ratio = 1, 6, and ∞ are shown in triangles, squares, and diamonds, respectively. Red & green colored symbols of each kind represent simulated turning & scooting trajectories.

6.4 Discussion

A key observation from the neuro-kinematic model is that increasing the amplitude *a* of the first neural burst fed into the left/right input neurons of the model and increasing the propagation delay *d* very rapidly switches the dynamics from scoot to turn. Thus the region of behavioral space between scoots and turns is narrow in neural parameter space. Moreover, the

model robustly produces swimming patterns within the behavioral space of the real organism in the presence of noise in amplitude and segment-to-segment delay. (In contrast, the model is more sensitive to timing noise between left/right input neurons.) Of course, the model should not be taken to comment on the neural architecture of the real organism. The model shows that a small set of parameter is sufficient to elicit all the observed swimming behaviors quantitatively.

References

1. Smith, B., et al., *Pseudorabies virus expressing enhanced green fluorescent protein: A tool for in vitro electrophysiological analysis of transsynaptically labeled neurons in identified central nervous system circuits.* Proceedings of the National Academy of Sciences of the United States of America, 2000. **97**(16): p. 9264-9269.
2. Dankert, H., et al., *Automated monitoring and analysis of social behavior in Drosophila.* Nature Methods, 2009. **6**(4): p. 297-303.
3. Colomb, J., et al., *Open Source Tracking and Analysis of Adult Drosophila Locomotion in Buridan's Paradigm with and without Visual Targets.* Plos One, 2012. **7**(8).
4. Roussel, N., et al., *A computational model for C-elegans locomotory behavior: Application to multiworm tracking.* Ieee Transactions on Biomedical Engineering, 2007. **54**(10): p. 1786-1797.
5. Fero, K., T. Yokogawa, and H.A. Burgess, *The behavioral repertoire of larval zebrafish*, in *Zebrafish models in neurobehavioral research*. 2011, Springer. p. 249-291.
6. Granato, M., et al., *Genes controlling and mediating locomotion behavior of the zebrafish embryo and larva.* Development, 1996. **123**: p. 399-413.
7. Fetcho, J.R. and S. Higashijima, *Optical and genetic approaches toward understanding neuronal circuits in zebrafish.* Integr Comp Biol, 2004. **44**(1): p. 57-70.
8. Granato, M., et al., *Genes controlling and mediating locomotion behavior of the zebrafish embryo and larva.* Development, 1996. **123**: p. 399-413.
9. Zhao, X., et al., *Amigo adhesion protein regulates development of neural circuits in zebrafish brain.* J Biol Chem, 2014. **289**(29): p. 19958-75.
10. Wang, S.J. and Z.W. Wang, *Track-a-worm, an open-source system for quantitative assessment of C. elegans locomotory and bending behavior.* PLoS One, 2013. **8**(7): p. e69653.
11. Fasciano, T., et al., *Tracking Multiple Ants in a Colony.* 2013 Ieee Workshop on Applications of Computer Vision (Wacv), 2013: p. 534-540.
12. Zhao, T. and R. Nevatia, *Tracking multiple humans in complex situations.* Ieee Transactions on Pattern Analysis and Machine Intelligence, 2004. **26**(9): p. 1208-1221.
13. Norton, W. and L. Bally-Cuif, *Adult zebrafish as a model organism for behavioural genetics.* Bmc Neuroscience, 2010. **11**.
14. Miklosi, A. and R.J. Andrew, *The Zebrafish as a Model for Behavioral Studies.* Zebrafish, 2006. **3**(2): p. 227-234.
15. Hughes, V., *Mapping brain networks: Fish-bowl neuroscience.* Nature, 2013. **493**(7433): p. 466-8.
16. Dawid, I.B., *Developmental biology of zebrafish.* Understanding and Optimizing Human Development:: From Cells to Patients to Populations, 2004. **1038**: p. 88-93.
17. Budick, S.A. and D.M. O'Malley, *Locomotor repertoire of the larval zebrafish: swimming, turning and prey capture.* J Exp Biol, 2000. **203**(Pt 17): p. 2565-79.
18. Roeser, T. and H. Baier, *Visuomotor behaviors in larval zebrafish after GFP-guided laser ablation of the optic tectum.* J Neurosci, 2003. **23**(9): p. 3726-34.
19. Higashijima, S., et al., *Imaging neuronal activity during zebrafish behavior with a genetically encoded calcium indicator.* J Neurophysiol, 2003. **90**(6): p. 3986-97.

20. McLean, D.L. and J.R. Fetcho, *Using Imaging and genetics in zebrafish to study developing spinal circuits in vivo.* Developmental Neurobiology, 2008. **68**(6): p. 817-834.
21. Ahrens, M.B., et al., *Whole-brain functional imaging at cellular resolution using light-sheet microscopy.* Nature Methods, 2013. **10**(5): p. 413-+.
22. Burgess, H.A. and M. Granato, *Modulation of locomotor activity in larval zebrafish during light adaptation.* J Exp Biol, 2007. **210**(Pt 14): p. 2526-39.
23. Mirat, O., et al., *ZebraZoom: an automated program for high-throughput behavioral analysis and categorization.* Front Neural Circuits, 2013. **7**: p. 107.
24. Bianco, I.H., A.R. Kampff, and F. Engert, *Prey capture behavior evoked by simple visual stimuli in larval zebrafish.* Front Syst Neurosci, 2011. **5**: p. 101.
25. de Esch, C., et al., *Locomotor activity assay in zebrafish larvae: influence of age, strain and ethanol.* Neurotoxicol Teratol, 2012. **34**(4): p. 425-33.
26. McClenahan, P., M. Troup, and E.K. Scott, *Fin-tail coordination during escape and predatory behavior in larval zebrafish.* PLoS One, 2012. **7**(2): p. e32295.
27. Colwill, R.M. and R. Creton, *Imaging escape and avoidance behavior in zebrafish larvae.* Rev Neurosci, 2011. **22**(1): p. 63-73.
28. Zeddies, D.G. and R.R. Fay, *Development of the acoustically evoked behavioral response in zebrafish to pure tones.* J Exp Biol, 2005. **208**(Pt 7): p. 1363-72.
29. Goldberg, D., M.J. Matari, and #263, *Coordinating mobile robot group behavior using a model of interaction dynamics*, in *Proceedings of the third annual conference on Autonomous Agents*. 1999, ACM: Seattle, Washington, USA. p. 100-107.
30. Tomlinson, B., *Negative Behavior Space in the Design of Interactive Agents.* AIIDE, 2005.
31. Stephens, G.J., et al., *Dimensionality and dynamics in the behavior of C. elegans.* PLoS Comput Biol, 2008. **4**(4): p. e1000028.
32. Berman, G.J., et al., *Mapping the stereotyped behaviour of freely moving fruit flies.* Journal of the Royal Society Interface, 2014. **11**(99): p. 20140672.
33. Jordan, D., et al., *Behavioral diversity in microbes and low-dimensional phenotypic spaces.* Proc Natl Acad Sci U S A, 2013. **110**(34): p. 14018-23.
34. Gallagher, T., et al., *The Geometry of Locomotive Behavioral States in C. elegans.* Plos One, 2013. **8**(3): p. e59865.
35. Shoval, O., et al., *Evolutionary Trade-Offs, Pareto Optimality, and the Geometry of Phenotype Space.* Science, 2012. **336**(6085): p. 1157-1160.
36. Hill, S.A., et al., *Neurokinematic modeling of complex swimming patterns of the larval zebrafish.* Neurocomputing, 2005. **65**: p. 61-68.
37. Higham, D.J., G. Kalna, and J.K. Vass, *Analysis of the Singular Value Decomposition as a Tool for Processing Microarray Expression Data.* Algoritmy 2005: 17th Conference on Scientific Computing, Proceedings, 2005: p. 250-259.
38. Ekeberg, O., *A Combined Neuronal and Mechanical Model of Fish Swimming.* Biological Cybernetics, 1993. **69**(5-6): p. 363-374.
39. Bhandiwad, A.A., et al., *Auditory sensitivity of larval zebrafish (Danio rerio) measured using a behavioral prepulse inhibition assay.* J Exp Biol, 2013. **216**(Pt 18): p. 3504-13.
40. Borla, M.A., et al., *Prey capture by larval zebrafish: evidence for fine axial motor control.* Brain Behav Evol, 2002. **60**(4): p. 207-29.

41. McElligott, M.B. and D.M. O'malley, *Prey tracking by larval zebrafish: axial kinematics and visual control.* Brain Behav Evol, 2005. **66**(3): p. 177-96.
42. Müller, U.K. and J.L. van Leeuwen, *Swimming of larval zebrafish: ontogeny of body waves and implications for locomotory development.* J Exp Biol, 2004. **207**(Pt 5): p. 853-68.
43. Westerfield, M., *The zebrafish book : a guide for the laboratory use of zebrafish (Brachydanio rerio)*. 1993, Eugene, OR: M. Westerfield.
44. Catherine, *Implementation of Background Subtraction Algorithm for Motion Detection.* 2013 International Conference on New Media Studies (Conmedia), 2013.
45. Schmid, V., G. Bader, and E. Lueder, *Shift, rotation and scale invariant shape recognition system using an optical Hough Transform.* Machine Vision Applications in Industrial Inspection Vi, 1998. **3306**: p. 102-112.
46. Liu, L.R., *Morphological Hit-or-Miss Transform for Binary and Gray-Tone Image-Processing and Its Optical Implementation.* Optical Engineering, 1994. **33**(10): p. 3447-3455.
47. Chung, K.L. and W.M. Yan, *A Fast Algorithm for Cubic B-Spline Curve-Fitting.* Computers & Graphics, 1994. **18**(3): p. 327-334.
48. Golub, G.H. and C. Reinsch, *Singular Value Decomposition and Least Squares Solutions.* Numerische Mathematik, 1970. **14**(5): p. 403-&.
49. Bellman, R., *Dynamic Programming and Lagrange Multipliers.* Proceedings of the National Academy of Sciences of the United States of America, 1956. **42**(10): p. 767-769.
50. Shepard, R.N., et al., *Multidimensional scaling; theory and applications in the behavioral sciences*. 1972, New York,: Seminar Press.
51. Williams, C.K.I., *On a connection between kernel PCA and metric multidimensional scaling.* Machine Learning, 2002. **46**(1-3): p. 11-19.
52. Jain, A.K., *Data clustering: 50 years beyond K-means.* Pattern Recognition Letters, 2010. **31**(8): p. 651-666.
53. Tibshirani, R., G. Walther, and T. Hastie, *Estimating the number of clusters in a data set via the gap statistic.* Journal of the Royal Statistical Society Series B-Statistical Methodology, 2001. **63**: p. 411-423.
54. Anderson, T.K., *Kernel density estimation and K-means clustering to profile road accident hotspots.* Accident Analysis and Prevention, 2009. **41**(3): p. 359-364.
55. Platt, C., *Zebrafish Inner-Ear Sensory Surfaces Are Similar to Those in Goldfish.* Hearing Research, 1993. **65**(1-2): p. 133-140.
56. Higgs, D.M., et al., *Development of form and function in peripheral auditory structures of the zebrafish (Danio rerio).* The Journal of the Acoustical Society of America, 2003. **113**(2): p. 1145.
57. Higgs, D.M., et al., *Age- and size-related changes in the inner ear and hearing ability of the adult zebrafish (Danio rerio).* Jaro, 2002. **3**(2): p. 174-184.
58. McClenahan, P., M. Troup, and E.K. Scott, *Fin-Tail Coordination during Escape and Predatory Behavior in Larval Zebrafish.* Plos One, 2012. **7**(2).
59. Gen, M. and R. Cheng, *Genetic algorithms and engineering optimization*. Wiley series in engineering design and automation. 2000, New York: Wiley. xvi, 495 p.
60. Laratta, A. and F. Zironi, *Computation of Lagrange multipliers for linear least squares problems with equality constraints.* Computing, 2001. **67**(4): p. 335-350.

Appendix A. Animals

This appendix includes detailed procedure on zebrafish breeding and embryo media to store fish embryos after breeding.

A.1 Zebrafish breeding

Zebrafish initiate breeding at the onset of light. Fertilized eggs can be obtained through pairwise breeding. Feed the fish 15 min before transferring into breeding tank. Transfer one female and one male to opposite sides of the breeding tank. Females can be distinguished from males because of their bigger underbelly. Males can also be distinguished from females because they are more slender and darker in color than females. Pairwise breeding is usually set up late in the afternoon after feeding. Remove the divider the next morning shortly after the onset of light. Allow mating to occur undisturbed for 20 min [43] or until sufficient numbers of embryos are laid at the bottom of the tank. In the meantime prepare embryo media:

Embryo Media:

1ml Hank's Stock #1 + 0.1 ml Hank's Stock #2 + 1.0 ml Hank's Stock #4 + 1ml Hank's Stock #5 + 1.0 ml fresh Hank's Stock #6 in 95.9 ml dd H2O

Use about 10 drops 1 M NaOH to pH 7.2.

Stock #1:8.0 g NaCl, 0.4 g KCl in 100 ml dd H2O

Stock #2: 0.358 g Na2HPO4 Anhydrous, 0.60 g KH2PO4 in 100 ml ddH2O

Stock #4: 0.72 g CaCl2 in 50 ml ddH2O

Stock #5: 1.23 g MgSO4x7H2O in 50 ml dd H2O

Stock #6: 0.35 g NaHCO3 in 10.0 ml dd H2O

** Place the Petri dish in the incubator at 28.5˚C when setting up the microscope for imaging.*

After breeding, return the fish to their tanks. Collect the eggs using a strainer. Transfer the embryos to a Petri dish by rinsing the strainer with embryos.

Appendix B. Data analysis

This appendix includes supporting information on image analysis from **Chapter 2**, parameter-free analysis from **Chapter 3** and derivation of equations from **Chapter 3**, **4**.

B.1 Image analysis

Figure B.1: MATLAB GUI for image analysis. This figure shows the customized toolbar with four panels: original grayscale image of first frame of a swim bout movie, background subtracted image with the slider at the to adjust the intensity of image for head recognition step, binary image: with the slider at the bottom to adjust the threshold to convert the original grayscale image to binary image and last panel of resultant skeletonized image.

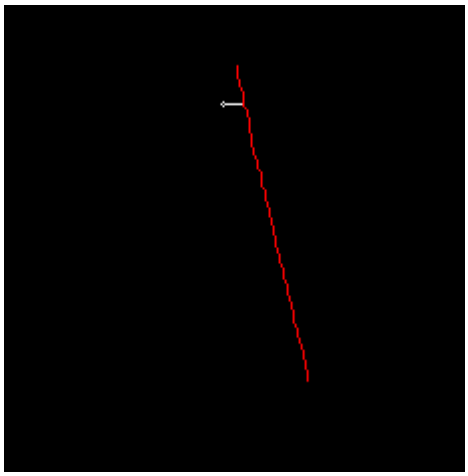

Figure B.2: Skeleton of fish with branches. After thinning of binary image of fish shape, fish skeleton has branches. This figure shows the results after branch cleaning algorithm that searches for the fish backbone (red) ignoring the branches.

B.2 Time series alignment

In **Chapter 4**, I quantified the behavioral variability in zebrafish free-swimming by measuring the "distance" between trajectories. To this end, I first aligned the amplitudes U_1, U_2, U_3 of each fish trajectory temporally, introducing a time shift and scaling the time axis, to maximize the overlap between them. Each trajectory α was assigned a "normalized time" $\tilde{t}^{\alpha} \equiv a_{\alpha} t + b_{\alpha}$ and the set of parameters $\{a_{\alpha}\}$ and $\{b_{\alpha}\}$ were optimized as described below.

The data were preprocessed before optimization by normalizing the time axis of U_1, U_2, U_3 to the time period of cycle 1 of U_1. Cycle 1 was defined as the time from the first to the third zero crossings of amplitude U_1 as labeled in **Figure 4.1** and **Figure 4.4.** To find the optimal set of normalized times maximizing the overlap between trajectories, we minimized the following function:

$$F = \min\left(\frac{2}{m(n(n-1))} \sum_{\alpha=1}^{N} \sum_{\beta=1}^{\alpha} \sum_{i=1}^{m} \| S_{\alpha}(\tilde{t}_i^{\alpha}) - S_{\beta}(\tilde{t}_i^{\beta}) \|^2 - \lambda_a \sum_{\alpha=1}^{N} (a_{\alpha} - 1) - \lambda_b \sum_{\alpha=1}^{N} (b_{\alpha} - 0) \right) \quad \text{(B1.1)}$$

over all frames i = 1,2,3… m, where

$$\| S_{\alpha}(\tilde{t}_i^{\alpha}) - S_{\beta}(\tilde{t}_i^{\beta}) \|^2 \equiv [U_{1\alpha}(\tilde{t}_i^{\alpha}) - U_{1\beta}(\tilde{t}_i^{\beta})]^2 + [U_{2\alpha}(\tilde{t}_i^{\alpha}) - U_{2\beta}(\tilde{t}_i^{\beta})]^2 \quad \text{(B1.2)}$$

and N is the number of trajectories, λ_a is a Lagrange multiplier [60] that ensures the sum of the parameter a_{α} over all trajectories is 1 and λ_b is a Lagrange multiplier that ensures the sum of the parameter b_{α} over all trajectories is 0. I minimized the least square difference over only two dimensions U_1, U_2 as we found that the variability in U_3 to affect the minimization. The results of this optimization are shown in **Figure 4.1** and **Figure 4.4.**

B.3 Dissimilarity matrix for MDS embedding

In **Chapter 4**, I have determined the "distance" or dissimilarity matrix $d^2_{\alpha\beta}$ between trajectories from the Euclidean distance between pairs α, β of time-aligned spine bend angles $\Delta\theta_\alpha$ and $\Delta\theta_\beta$. I show here how the dissimilarity matrix can be expressed in terms of the amplitudes U_1, U_2, U_3. The distance metric $d^2_{\alpha\beta}$ is given by:

$$d^2_{\alpha\beta} = \sum_{i,j} \left[\Delta\theta_\alpha\left(s_j, \tilde{t}_i^{\,\alpha}\right) - \Delta\theta_\beta\left(s_j, \tilde{t}_i^{\,\beta}\right) \right]^2 \qquad \text{(B.2.1)}$$

where $\tilde{t}_\alpha$ and $\tilde{t}_\beta$ are the normalized times for each trajectory. Substituting **equation (B.2.1)** for $\Delta\theta_\alpha$ and $\Delta\theta_\beta$ yields:

$$d^2_{\alpha\beta} = \sum_{i,j} \left\{ \sum_{k=1}^{n} S_{kk} V_k\left(s_j\right) \left[U_k^\alpha\left(\tilde{t}_i^{\,\alpha}\right) - U_k^\beta\left(\tilde{t}_i^{\,\beta}\right) \right] \right\}^2 \qquad \text{(B.2.2)}$$

which expands to:

$$d^2_{\alpha\beta} = \sum_{i,j} \left\{ \sum_k \sum_{k'} S_{kk} V_k\left(s_j\right) S_{k'k'} V_{k'}\left(s_j\right) \left[U_k^\alpha\left(\tilde{t}_i^{\,\alpha}\right) - U_k^\beta\left(\tilde{t}_i^{\,\beta}\right) \right] \left[U_{k'}^\alpha\left(\tilde{t}_i^{\,\alpha}\right) - U_{k'}^\beta\left(\tilde{t}_i^{\,\beta}\right) \right] \right\} \qquad \text{(B.2.3)}$$

From the orthonormality of the eigenshapes, i.e. $\sum_j V_k\left(s_j\right) V_{k'}\left(s_j\right) = \delta_{kk'}$, it follows that

$$d^2_{\alpha\beta} = \sum_i \sum_{k=1}^{n} S_{kk}^2 \left[U_k^\alpha\left(\tilde{t}_i^{\,\alpha}\right) - U_k^\beta\left(\tilde{t}_i^{\,\beta}\right) \right]^2 \qquad \text{(B.2.4)}$$

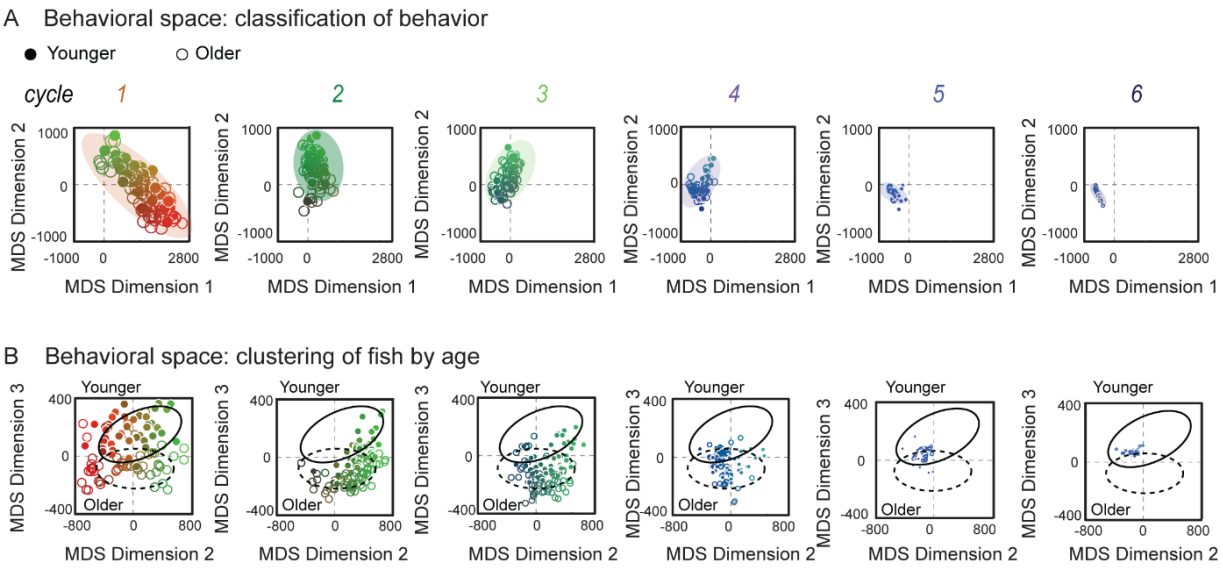

Figure B.3: Behavioral space of free-swimming zebrafish resolved by cycle. (A) Behavioral space in MDS dimensions 1 and 2 from **Figure 4.3**, plotting each oscillation cycle separately (leftmost panel for cycle 1, rightmost panel for cycle 6). (B) Behavioral space in MDS dimensions 2 and 3 from **Figure 4.6**, plotting each oscillation cycle separately. Throughout, the same colormap and symbols from **Figure 4.3** are used.

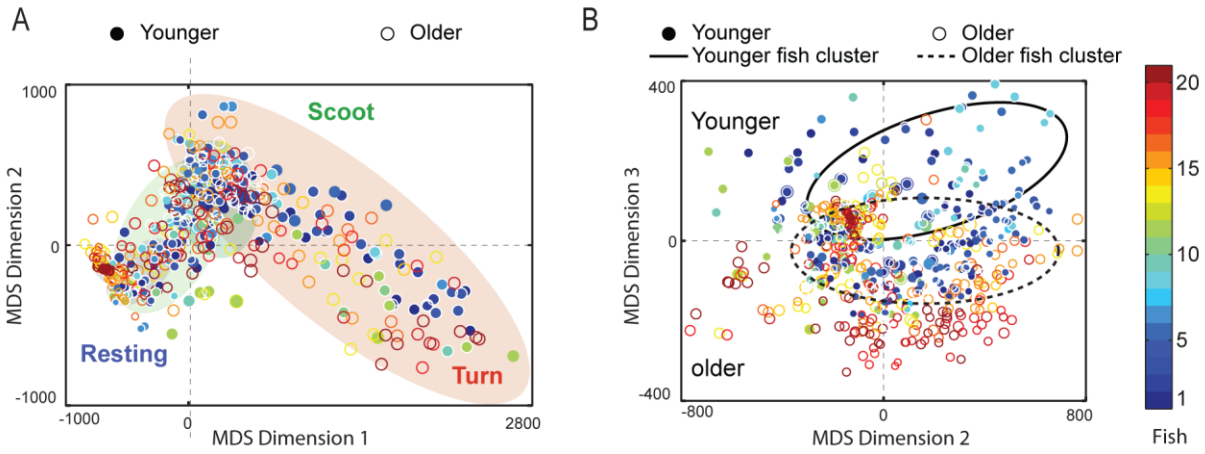

Figure B.4: Behavioral variability in a single zebrafish vs. population. (A) Behavioral space in MDS dimensions 1 and 2 from **Figure 4.3** with each fish (*N* = 20) represented by a separate color (see colormap). Older larvae are represented by open circles, younger by filled circles. The same level of behavioral variability is observed at the single-fish level as at the population level. Individual fish do not exhibit any preference for one type of behavior pattern. (B) Behavioral space in MDS dimensions 2 and 3 from **Figure 4.6** with each fish represented different colors as in A. Fish trajectories separate by age as demarcated by the elliptical outlines (solid for younger, dashed for older larvae).

Appendix C. Neural model

C.1 Stability of trajectories produced by neural model

We investigated the effect of noise on model parameters $\{\tau_f^l, a_l, d_l\}$ to evaluate the stability of trajectories $\Delta\theta^{neuro}(s_j, t_i)$ generated by the neural network. The impact of noise on these parameters was evaluated from the difference between $d^2_{\alpha\beta(neuro)}$ (see **Eq.(B.2.4)**) generated by the neural network in the absence of noise and $d^2_{\alpha\beta(noise)}$ obtained after adding noise. **Figure 6.9** displays $d^2_{\alpha\beta(noise)} - d^2_{\alpha\beta(neuro)}$ as a function of white Gaussian noise added at varying levels to model parameters (plotted as signal-to-noise ratio). We then embedded the trajectories generated by the neural network with noise in a behavioral space with the experimental data. **Figure 6.9D-F** show simulated scoot (green colored symbols) and turn (red colored symbols) trajectories with signal-to-noise ratio = 1 (triangles), 6 (squares), and ∞ (diamonds) in *a*, τ_f, and *d* (bottom left panel to bottom right panel). Trajectories with noise in spike firing time τ_f, are scattered the most in behavioral space, indicating a high sensitivity to noise in this parameter. We believe this is because noise disrupts the firing sequence of the right and left halves of the backbone, which leads to non-cyclic trajectories. In contrast, noise in the segment-to-segment delay has the least effect on behavior. Interestingly noise in spike amplitude *a* can convert a scoot trajectory into a turn and vice versa (see points with signal-to-noise ratio = 6).

Appendix D. List of movies

This appendix includes the list of movies of free-swimming zebrafish, postural space and simulated free-swimming from neural model as describe in **Chapter 2**, **3**, **4**, and **6**.

Movie D1.0.1: Free swimming zebrafish larva. The video file is of a freely swimming larvae in a petri dish recorded at a rate of 500 frames per second (fps) using a high-speed camera. The playback rate of this video is 100 fps. Its shows total of 3 swimming episodes with 4 intermittent pauses (resting).

Movie D1.0.2: Postural space representing a turn and a scoot region of a zebrafish larva. The left side of the video file shows a free swimming larva and a spline curve in cyan fitted to its backbone. The right side of the video file shows the corresponding 3D postural space. The dot on 3D plot shows the contribution of amplitudes $U_1(t)$, $U_2(t)$, and $U_3(t)$ of eigenfunction V_1, V_2 and V_3 respectively for an instantaneous frame of a swimming fish on right side. In this space, the bout involves a turn region (t = 50-100 ms), represented as a bent ellipse, followed by a scoot region (t = 100-250 ms) represented as multiple cycles along the flat ellipses, and a final return to the rest behavior. Throughout, time (0–250 ms) is represented by the black–magenta colormap.

Movie D1.0.3: Postural space representing a scoot region of a zebrafish larva. The left side of the video file shows a free swimming larva and a spline curve in cyan fitted to its backbone. The right side of the video file shows the corresponding 3D postural space. The dot on 3D plot shows the contribution of amplitudes $U_1(t)$, $U_2(t)$, and $U_3(t)$ of eigenfunction V_1, V_2 and V_3 respectively for an instantaneous frame of a swimming fish on right side. In this space, the bout involves a scoot region (t = 50-250 ms) represented as multiple cycles along the flat ellipses, and a final return to the rest behavior. Throughout, time (0–250 ms) is represented by the black–magenta colormap.

Movie D1.0.4: Postural space of a population of younger zebrafish (7-8 dpf). The video file shows 3D postural space of the trajectories of a population of younger fish. Each data point is the contribution of amplitudes $U_1(t)$, $U_2(t)$, and $U_3(t)$ of eigenshape V_1, V_2 and V_3 respectively (**Figure 4.5A**). The video sweeps with the increase in elevation angle of 3D plot to show the continuum between two of the extremes—turn-like trajectories and scoot-like trajectories, rather than clustering into very distinct behavioral states. For details on colormap see **section 4.3.1.**

Movie D1.0.5: Postural space of a population of older zebrafish (9-10 dpf). The video file shows 3D postural space of the trajectories of a population of older fish. Each data point is the contribution of amplitudes $U_1(t)$, $U_2(t)$, and $U_3(t)$ of eigenfunction V_1, V_2 and V_3 respectively (**Figure 4.5B**). The video sweeps with the increase in elevation angle of 3D plot to show the continuum between two extremes—turn-like trajectories and scoot-like trajectories, rather than clustering into very distinct behavioral states. For details on colormap see **section 4.3.1.**

Movie D1.0.6: Simulated turn and scoot of zebrafish using neural network model. The left side of the video file shows a free swimming larva in a petri dish recorded at a rate of 500 frames per second (fps) using a high-speed

camera and a spline curve in cyan fitted to its backbone. The right side shows the same free swimming larva and a spline curve in red fitted to its backbone obtained using neural network model. The video has 2 swimming episodes (a turn preceded by a scoot & a scoot) with one intermittent pause. The colored marker represents the tail position at each time point. The playback rate of this video is 10 fps.