Metric Estimation and Data Analysis Tools for the Sharkduino Animal Tag System

A thesis submitted in partial fulfillment of the requirement
for the degree of Bachelor of Science in Physics
from the College of William and Mary in Virginia,

by

Dara Kharabi

Accepted

Prof. Wouter Deconinck

Prof. Irina Novikova

Prof. Kevin C. Weng

Williamsburg, Virginia
May 2018

Contents

Acknowledgments **iii**

List of Figures **iv**

Abstract **v**

1 Introduction **1**

1.1 Purpose and Relevance 1

1.2 Problem Statement 4

2 Solutions & Techniques **6**

2.1 Data Ingestion, Processing, and Cleaning 6

2.2 Visualizing Raw Tag Data 6

2.3 Overall Dynamic Body Acceleration 9

2.3.1 Determining ODBA 9

2.3.2 Visualizing ODBA 11

2.4 Tailbeat Frequency 11

2.4.1 Initial Challenges 11

2.4.2 Naïve Methods: Counting Zero-Crossings in A_z and G_x 11

2.4.3 Integrating G_x and Correcting for Drift 13

2.4.4 Spectral Methods 18

2.4.5 Post-Processing Techniques: Advantages and Drawbacks . . . 21

3 Results and Conclusions **23**

3.1 Conclusions 23

3.2 Further Work 24

Acknowledgments

First and foremost, I would like to thank the heads of Sharkduino, Dr. Wouter Deconinck and Dr. Kevin Weng: without their incredible knowledge, excellent intuition, and saint-like patience, this thesis (and the work behind it) would not exist. Moreover, I am grateful to my major advisor, Dr. Irina Novikova, for helping guide me through the undergraduate physics process, and for not backing away from the difficult conversations that process sometimes entails.

I would like to also thank my labmates, Cal Bilenkin, Ben Powell, and Seanna Nam, as well as Sharkduino alumnus William Laney, for providing not just advice, help, and camaraderie, but also hardware, firmware, and infrastructure that made Sharkduino data analysis feasible and rewarding.

Collaboration with the Virginia Institute of Marine Science is at the heart of Sharkduino. In particular, the VIMS Eastern Shore Lab facilities were integral to this work. I would like to thank Dan Crear, Rich Brill, Peter Bushnell, Gail Schwieterman, Sean Fate, P.G. Ross, Justin Paul, Reade Boniwell, Stephanie Bonniwell, Jim Brister, and Richard Snyder for assisting with experiments and helping collect the datasets that this work is based on.

Finally, I would like to thank my mother, Roya, and my sister, Darya, for their unending confidence, friendship, and support.

List of Figures

1.1 Sharkduino Animal Tag Photograph 2

1.2 Sharkduino Tag Attached to Shark 3

1.3 Sandbar Shark Tag Placement and Coordinate Diagram 3

1.4 Visual Demonstration of Tailbeat Frequency 5

2.1 Demonstration of Tag Date Interpolation 7

2.2 Summary Plots for All Datasets 8

2.3 SharkPlotViewer: A Web Interface for Sharkduino Plot Data 9

2.4 ODBA Over One Minute of Shark Data 10

2.5 Tailbeat Frequency: Counting Zero-Crossings in A_z 12

2.6 Tailbeat Frequency: Counting Zero-Crossings in G_x 13

2.7 Tag_Visualizer: 3D Tag Orientation Visualization 16

2.8 Yaw Data from the Madgwick AHRS Algorithm 17

2.9 Tailbeat Frequency: Counting Zero-Crossings in AHRS Yaw 18

2.10 Tailbeat Frequency: PSD vs. Simple Periodogram 19

2.11 Tailbeat Frequency: Spectrogram 20

2.12 Comparison of Filtering Regimes 22

Abstract

This work investigates several avenues into data analysis for the Sharkduino Animal Tag System. First, we detail the animal tag system and its advantages over contemporary marine biology sensor tag solutions. We then describe the infrastructure built to facilitate basic data import, cleaning, and date interpolation processes. We explain the methodology used for visualizing summary sensor data, and present data analysis tools for exploring line plots, visualizing shark orientation, and performing batch data cleaning/visualization tasks on large (tens of gigabytes) datasets.

Two metrics of interest to marine biologists are explored in detail: Overall Dynamic Body Acceleration (ODBA) and tailbeat frequency. We begin by briefly exploring the implementation of the reference calculation of ODBA used for Sharkduino data analysis. We then explore and compare three methods for calculating tailbeat frequency: Naïve accelerometer/gyro sensor, AHRS yaw, and spectral analysis. A brief discussion of the potential advantages and downsides of various filtering and tapering techniques follows. We finish by enumerating some avenues for further research.

Chapter 1

Introduction

1.1 Purpose and Relevance

One of the greatest obstacles marine biologists face in comparison to their terrestrial counterparts is the difficulty of direct observation. The lives of many such animals are poorly understood by science, largely due to the difficulty and expense associated with following and observing an animal that lives underwater. The advent of electronic sensor tags in the last two decades has opened a promising new avenue toward understanding marine life. Moreover, advances in accessible programmable microcontrollers, wearable sensors, and data analysis in the last 5 years are pushing the envelope with regard to what an animal-mounted sensor tag can do within reasonable size and cost parameters.

The Virginia Institute for Marine Science studies sandbar sharks, a migratory shark species that visit Virginia's Chesapeake Bay in the summer. The eating, resting, and mating patterns of these sharks are, as yet, poorly understood. The current body of research in this area employs primarily ASIC and FPGA-based sensor tags [1]. These tags are custom-built for marine science and often equipped with only a clock and an accelerometer [2].

The Sharkduino project aims to improve upon the current research by developing

a Arduino microcontroller-based sensor tag built entirely from off-the-shelf parts (See Figures 1.1 and 1.2), along with an accompanying suite of data analysis software packages. This approach would offer marine scientists a more flexible animal tag with a more comprehensive sensor package at a fraction of the price commanded by existing solutions made from custom parts. The associated suite of open-source data-analysis software aims to make data analysis approaches like machine learning and sensor fusion more accessible to marine scientists, allowing for a more nuanced understanding of animal tag data and, hence, of animal behavior. Very promising research and development has already been completed around the hardware portion of the project [3]. This work describes key initiatives and improvements regarding the data analysis portion of the project.

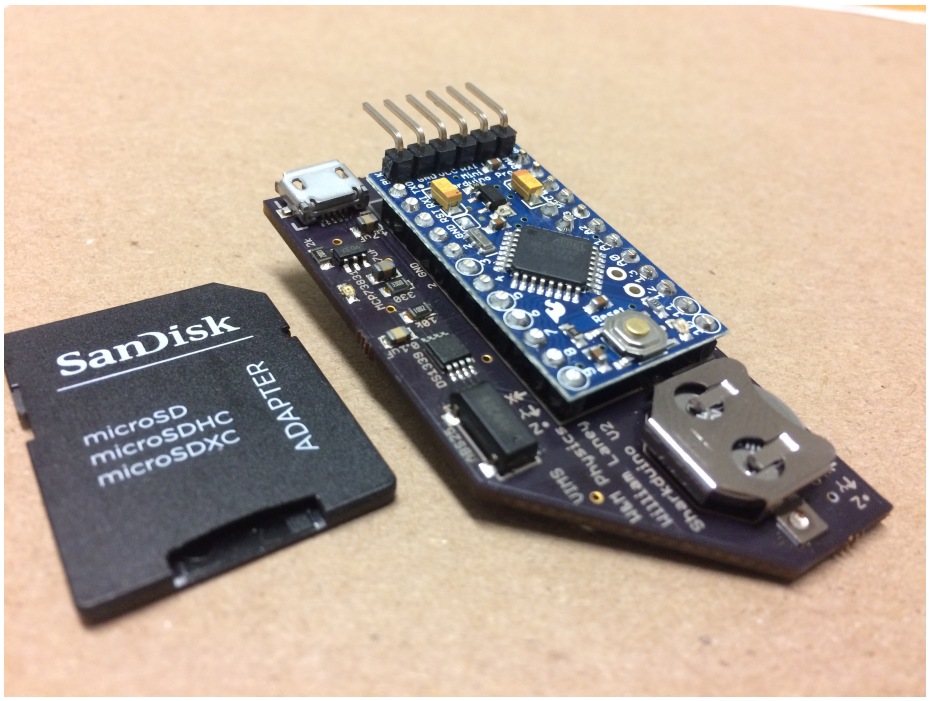

Figure 1.1: Image of a v2 Sharkduino animal tag. Full-size SD card adapter included for scale.

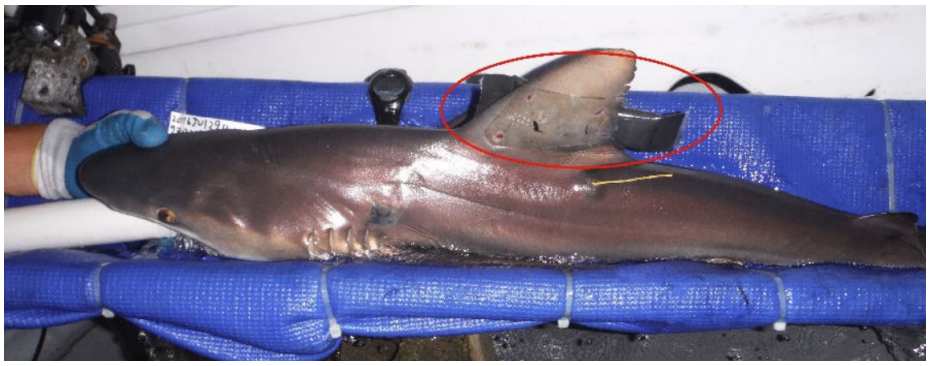

Figure 1.2: Sharkduino animal tag in waterproof enclosure, attached to a sandbar shark. All work with sharks was conducted in accordance with an approved animal care protocol, `IACUC-2017-05-26-12133-kcweng`.

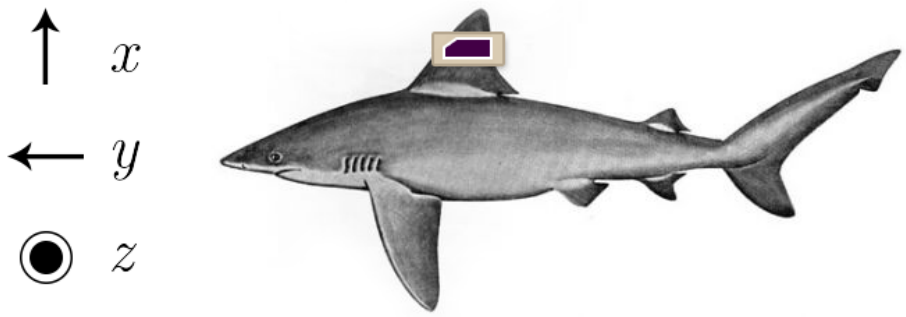

Figure 1.3: This image demonstrates the positioning of the Sharkduino Tag on a sandbar shark's dorsal fin. The coordinate system on the left side is used for all data analysis in this document. Positive angular velocities move clockwise around their respective axes. Sandbar shark image adapted from [4].

1.2 Problem Statement

While the Sharkduino project encompasses all research concerning the development and use of the Sharkduino animal tag, the work described in this report is concerned entirely with metric calculation portions of the data analysis aspect of the project. Data analysis for Sharkduino aims to eventually provide marine scientists with a set of tools with which to process and analyze data from Sharkduino animal tags. Metric calculation involves identifying or constructing metrics that would help marine biologists ascertain the state and behavior of the tagged animals, then determining reliable methods of deriving those metrics from the raw animal tag data and visualizing them in clear and accurate ways. This work is accomplished through a set of R scripts and libraries.

The raw data from the Sharkduino tag comprises:

- Three axes (A_x, A_y, and A_z) of linear proper acceleration from the accelerometer (range: $\pm$8g, error: $\pm$0.1g)
- Three axes (G_x, G_y, and G_z) of angular velocity from the gyro sensor (range: $\pm$2,000dps (degrees per second), error: $\pm$2dps)
- Date and time, from a real-time-clock. Time is precise to the second, and accurate to 2ppm[5], or just over a minute per year.

Please see Figure 1.3 for an explanation of the coordinate system used.

The data analysis work aims to deliver three products:

- Tailbeat frequency, i.e., the average rate of the fish's tail's side-to-side oscillation, is a key metric for scientists studying fish (See Figure 1.4). In many fish, tailbeat frequency can be a proxy for swimming speed, energy level, digestion, and other biologically significant quantities [6]. A reasonably accurate

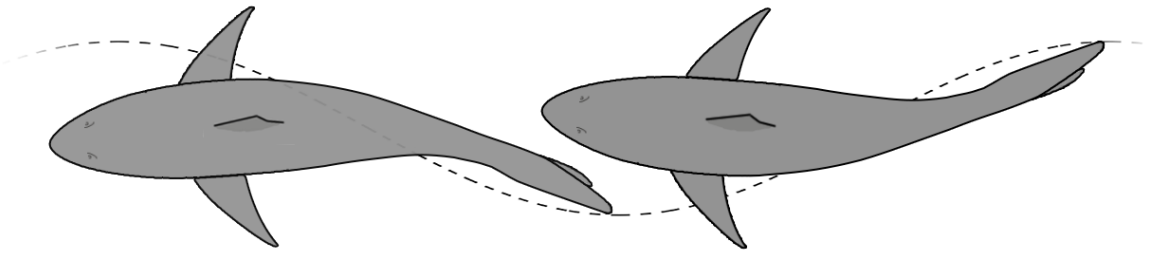

Figure 1.4: As a sandbar shark swims, its tail makes a periodic side-to-side twisting motion. The rate of this motion is tailbeat frequency, an important biological metric we aim to calculate using motion data from a tag mounted on the shark's dorsal fin.

method of estimating average tailbeat frequency from a slice of animal tag data is desired.

- Another useful metric is Overall Dynamic Body Acceleration (ODBA), which summarizes the total coordinate acceleration of a body in all axes into one dimension of data. ODBA is used frequently in studies of animal behavior as a proxy for energy expenditure [7]. We believe this metric will have similar uses for studying sharks, hence the desire to correctly calculate ODBA from animal tag data.

- Finally, effective methods for visualizing these metrics, as well as raw tag data like acceleration and angular velocity, are a priority. Data visualization is an important prerequisite for understanding the tag data, evaluating the effectiveness of data analysis strategies, identifying bugs, and extracting useful insights about the tagged animals.

Chapter 2

Solutions & Techniques

General methods for data processing and visualization will be presented first, followed by sections detailing the specifics of attaining both ODBA and tailbeat frequency.

2.1 Data Ingestion, Processing, and Cleaning

Before plotting and analyzing the animal tag data, some basic processing operations must be performed. The raw CSV data is read into an R `data.table` (a faster and more flexible analog to the built-in `data.frame` collection) and fields in the data are assigned appropriate types. The Sharkduino tag samples accelerometer and gyro sensor data at 25 Hz, but writes date and time from the real-time clock approximately once every 6 seconds. As a result, date and time values exist only intermittently in the raw data, and must be interpolated. Data rows are assigned times based a series of linear interpolations between consecutive date/time rows (see Figure 2.1).

Once these steps are complete, the processed data can be written back to the disk in .CSV or .RData formats, or kept in memory for subsequent analysis.

2.2 Visualizing Raw Tag Data

Initially, a number of basic plots were constructed using datasets collected from sharks captured and studied at VIMS in summer 2016 (see Figure 2.2). R scripts were created

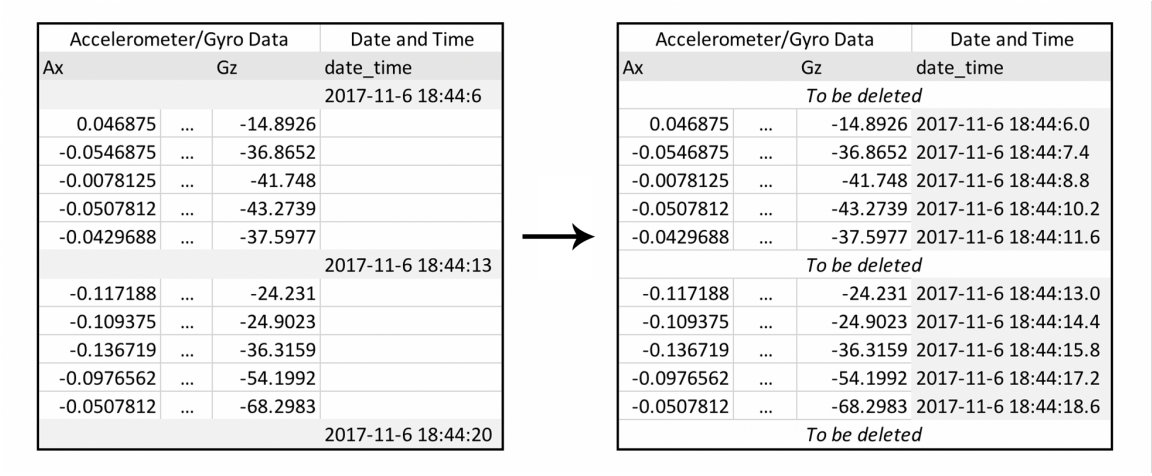

Figure 2.1: The sparse date rows present in the raw animal tag data are interpolated linearly, thereby determining dates and times for the data rows in between. Once this task is complete, the date-only rows are discarded.

to generate line plots of sensor data over time. Six time-series (3 axes of Accelerometer data and 3 axes of gyro sensor data) were plotted in this way. Because the chosen plotting library (`ggplot2` with `cowplot`) can take several minutes to generate plots with more that 100,000 points, routines for subsampling the data were developed. Initially, a simple decimation (selecting every nth sample) method was used. Later, a more sophisticated method was added that uses the average of every n points as the sub-sample (this is a 1D analog to box sampling in 2D image processing). These routines allow for rapid generation of summary plots, at the expense of clarity in the plots' high frequency components.

Once these plots were standardized, it was important to generate and maintain a standard set of summary statistics and visualizations for each dataset in the project's historical archives. This task was initially accomplished with a purpose-built batch script, but the desire for new summary chart types and the growing size of the group's dataset library limited the efficacy of this approach. These obstacles eventually led to the development of a more flexible solution called Data_Pipe, comprising a lightweight

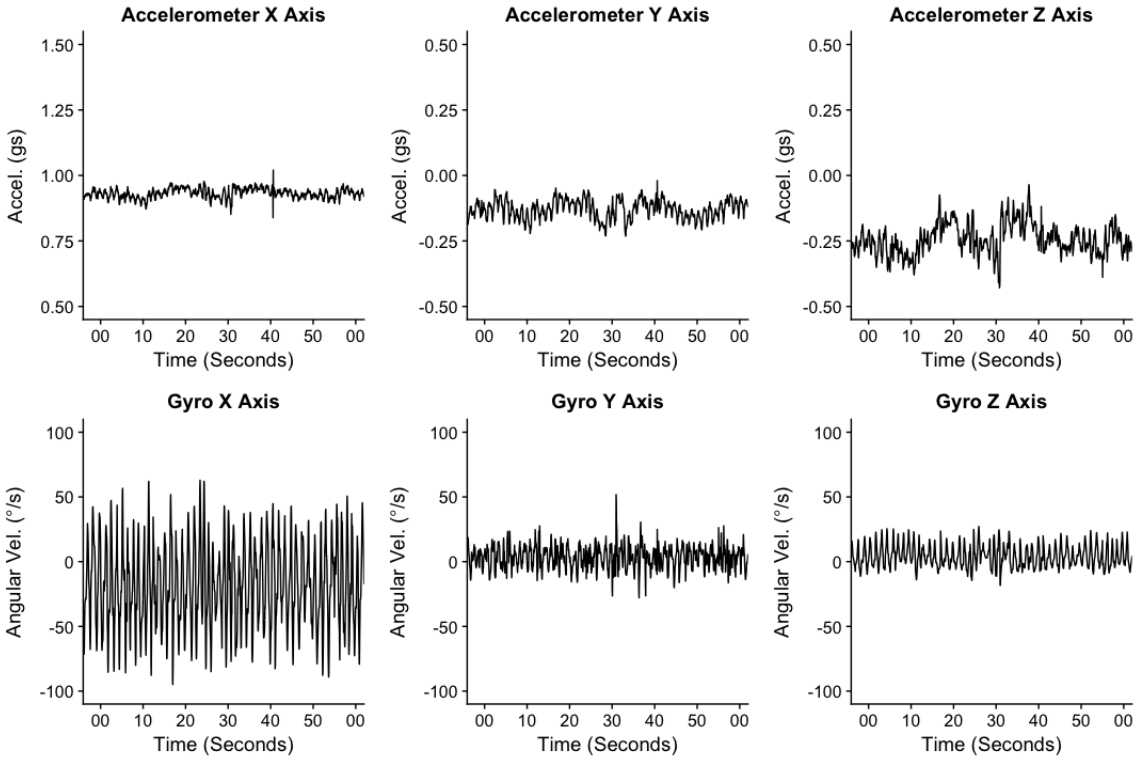

Figure 2.2: Analysis of every dataset begins with line plots of each of the six axes of raw sensor data. The plots shown cover one minute of shark swimming data, but plots used for analysis may include a week or more's worth of sensor data. These plots can be generated and updated *en masse* using Data_Pipe.

framework for executing batch jobs in parallel across all datasets in the Sharkduino archive.

Finally, the need for rapidly configurable, interactive line plots of sensor data led to the development of SharkPlotViewer, a web application developed with R and the Shiny package (see Figure 2.3). SharkPlotViewer allows a user to view any axis of a Sharkduino dataset, zoom in and out, and scroll back and forth through time. Multiple axes of the same dataset may also be viewed side-by-side. SharkPlotViewer proved helpful for exploring scatterplots during meetings and as a proof-of-concept for interactive web apps, but the line plots it presented were limited in their usefulness and explanatory power.

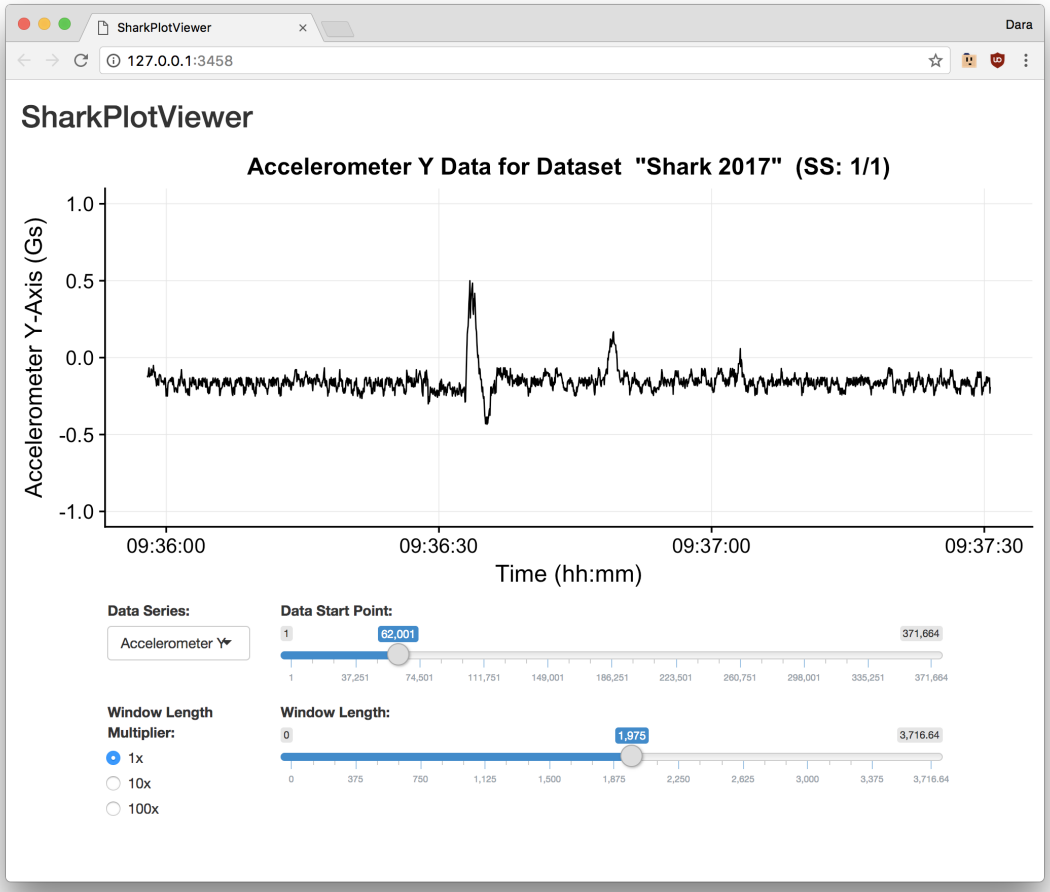

Figure 2.3: The SharkPlotViewer web interface.

2.3 Overall Dynamic Body Acceleration

2.3.1 Determining ODBA

Overall Dynamic Body Acceleration, or ODBA, is an aggregate metric constructed from 3 axes of acceleration data. Since its popularization in 2006[8], ODBA has gained favor in the zoology, marine biology, and medical devices communities as a proxy for animal energy expenditure. Gleiss, et al.[7] provide a survey of the metric's use in the literature, along with background theory as to its usefulness in estimating

animal metabolic rates. Compared to related metrics, such as acceleration vector magnitude, ODBA appears to provide more durable and accurate results [9]. For all of these reasons, a method for determining and visualizing ODBA from tag data was desired.

Calculation of ODBA was done according to the method described by Wilson, et al. Static acceleration was removed by way of subtracting from each axis of acceleration data a copy smoothed via 2s rolling-mean filter (Wilson, et al's paper uses a 1s rolling mean filter, but the filter window size may vary depending on the data being analyzed. the 2s window size was arrived at experimentally by comparing ODBA output from multiple window size settings). The absolute values of the resulting three components were then summed, yielding one dimension of ODBA data.

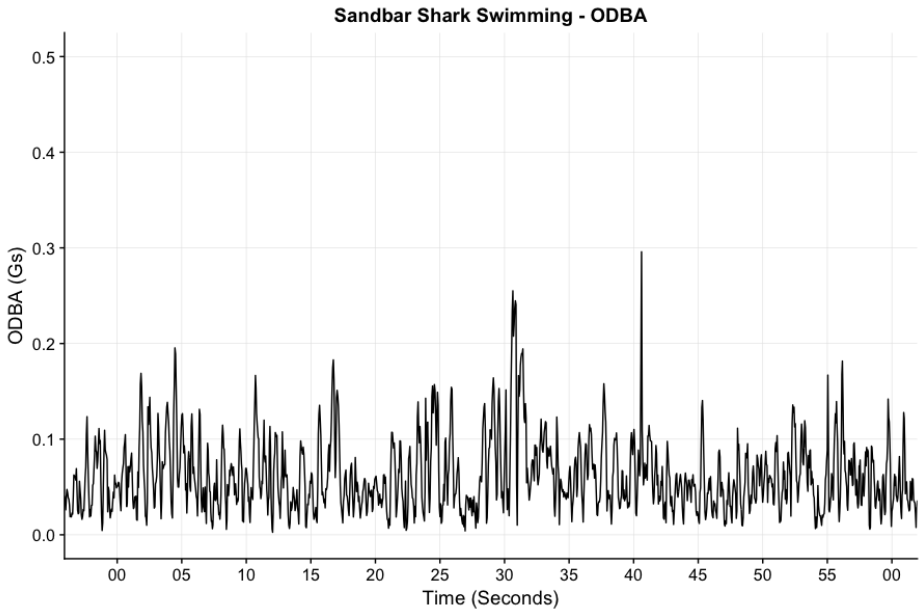

Figure 2.4: This chart shows ODBA for one minute of shark swimming data. Unless noted otherwise, all plots in this document with the same time-scale (e.g. this figure and Figure 2.2) display values for the same window of time data.

2.3.2 Visualizing ODBA

Once a proper method for calculating ODBA was implemented, A graph of ODBA vs. time was added to the six one-axis charts created automatically for each dataset (see Figure 2.4).

2.4 Tailbeat Frequency

2.4.1 Initial Challenges

Unlike ODBA, which is defined as the result of a calculation, tailbeat frequency maps to a specific biological phenomenon. As such, in constructing metrics to determine tailbeat frequency without a source of authoritative ground truth, we run into a great deal of ambiguity. Given one measurement, there is no way to identify whether it is accurate. Given two measurements that disagree, there is ambiguity as to which one is more accurate. The approach selected to curtail this ambiguity was to construct four different tailbeat frequency metrics, each based on a different set of base assumptions, and then compare them. Consensus between these different metrics would suggest agreement with the underlying biological phenomenon.

2.4.2 Naïve Methods: Counting Zero-Crossings in $\mathbf{A_z}$ and $\mathbf{G_x}$

The first method explored was based on the assumption that the periodic twisting motion of the shark would result in alternating regimes of positive and negative acceleration in the axis (Az) transverse to the shark (see Figure 2.5).

This method gave a plausible, if inflated, count of tailbeat frequency. One clear shortcoming of the technique was that high-frequency noise in the accelerometer data caused a large number of insignificant zero-crossings. Two approaches to mitigating this behavior are low-pass filtering of the accelerometer (discussed further in Section 2.4.5), and using data from the less noise-prone gyro sensor instead. To this end, the

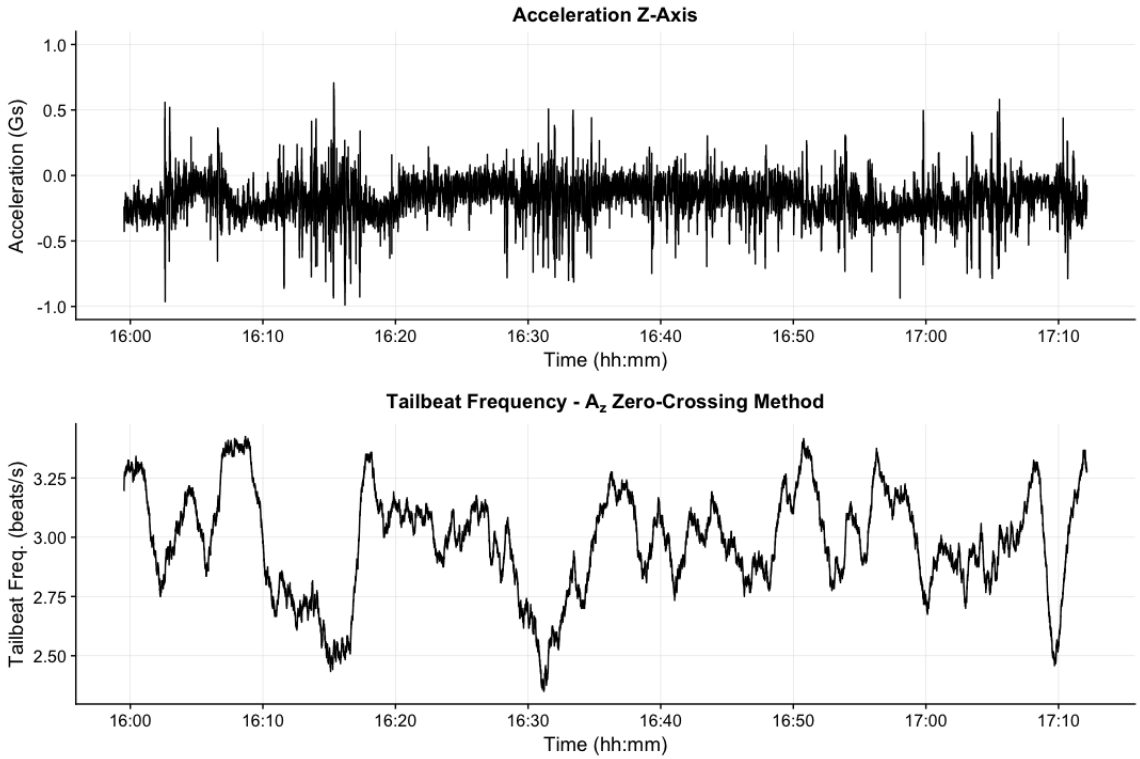

Figure 2.5: Naïve tailbeat frequency calculation via counting zero-crossings in the accelerometer's Z-axis. This and following charts cover an hour of shark swimming data, beginning with the 1-minute window used for Figure 2.4 and previous.

same method was used to investigate whether the angular velocity in the shark's yaw axis tracked tailbeat frequency. This method gave a lower value for tailbeat frequency than the accelerometer-based approach, perhaps due to the absence of high-frequency noise in the gyro sensor (see Figure 2.6).

The tailbeat frequency plots for the hour of data visualized in Figures 2.5 and 2.6 show no clear correlation. However, the plots do express interesting behavior around the high-amplitude A_z/G_x regions centered on 16:15 and 16:32. In these regions, the A_z method's calculation for tailbeat frequency is minimized, while the G_x method's calculation rises in value. More research must be done to determine whether this behavior is anomalous or characteristic of these two metric calculation methods.

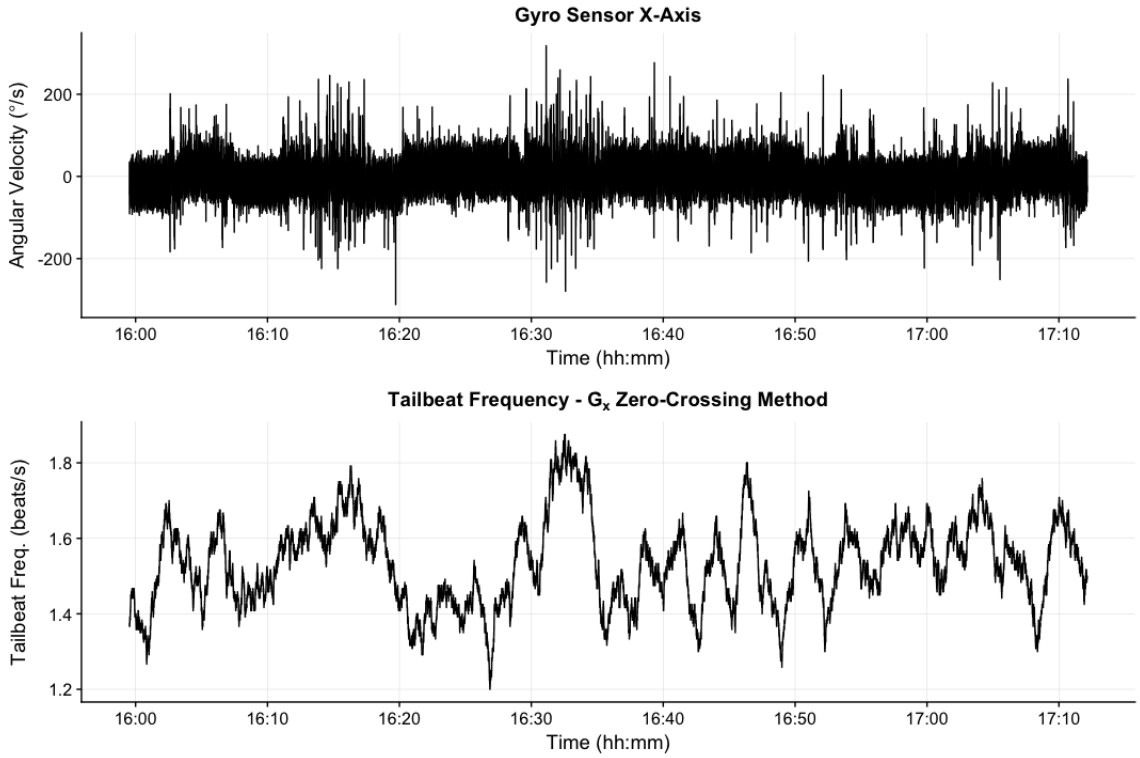

Figure 2.6: Somewhat improved tailbeat frequency calculation via counting zero-crossings in the gyro sensor's Z-axis. The value given by this method should be unaffected by noise in the accelerometer, but is prone to gyro sensor drift. For the window of data displayed, the G_x method exhibits a slightly smaller range of values than the A_z method.

2.4.3 Integrating G_x and Correcting for Drift

A logical extension of the zero-crossing method detailed above is to integrate the angular velocity data from the gyro sensor to yield angular position, or orientation. Combining this data with a known reference point would allow us to model the various ways in which the tag turned throughout the length of the dataset. While promising, this approach comes with a significant caveat: because obtaining orientation necessitates integrating gyro sensor data, error in this value (called drift) becomes cumulative, increasing so rapidly that the orientation metric becomes useless after just a few minutes.

Luckily, this is a well-known problem in aeronautics and wearable technology, and has a fairly straightforward solution: although drift in the orientation is cumulative, error in the accelerometer is not. Given an adequate sensor fusion algorithm, the data from the accelerometer may be used to correct for the drift in the gyro sensor, allowing for a stable orientation metric. Sensor fusion algorithms of this type are called Attitude-Heading Reference System (AHRS) algorithms, and are the basis of units found inside many consumer electronic devices.

Several AHRS algorithms exist in the literature. Many of these were developed for aerospace applications and perform best at sampling rates higher than the Sharkduino tag's default 25Hz, but some newer algorithms have shown adequate performance in the literature at sampling rates as low as 10Hz [10]. The industry standard solution for AHRS in aerospace, robotics, MEMS, and motion-capture industries has long been the Kalman Filter and its assorted variants (most notable the Extended Kalman Filter). While highly reliable and very well-characterized, Kalman Filter-based AHRS algorithms are relatively difficult to implement in software and don't perform well at sampling rates below 100Hz [11]. Another algorithm, developed in 2009 by Madgwick [10], has rapidly become a mainstay of RC drone programming. Madgwick's algorithm is structurally simpler than most alternatives and performs well at Sharkduino's sampling rate. For these reasons, Madgwick's algorithm was selected for use in Sharkduino data analysis.

Implementation of the algorithm experienced a number of setbacks, including poorly written third-party packages and problems inherent with Euler angles. Implementation initially began with the attempted integration of the R AHRS package. After noting anomalous results and several software crashes, a cursory examination of the package's source code revealed that it was riddled with both mathematical and programming errors, rendering it unusable. Without a pre-packaged AHRS algorithm,

it became necessary to port the algorithm over to R. An Arduino implementation of the algorithm, intended for use in RC quadcopters, was used as the source for this port. Once the port was completed, the ported code was checked against Madgwick's original MATLAB code and corrections from the project's online community to ensure its validity.

A second problem presented itself when Euler angles were used to represent the algorithm's output. Euler angles are prone to a problem called gimbal lock, wherein orienting an object in certain positions can cause the loss of a degree of freedom, hence preventing smooth transitions to some new orientations [12]. Furthermore, Euler angles map to orientations in a many-to-one fashion, allowing some ambiguity in terms of how an orientation is expressed. These obstacles were sidestepped by replacing Euler angle-based Cartesian rotations in the code with quaternion-derived rotation matrices (for more details on quaternions in AHRS, see [13]). In this way, the AHRS code avoids using Euler angles entirely.

To properly understand and test the AHRS algorithm, intuitive methods for visualizing tag orientation had to exist. In response to this need, a Processing script called tag_visualizer was created. Given a set of Euler angles or quaternions (determined from animal tag data using the AHRS algorithm), tag_visualizer draws a 3D representation of the animal tag (see Figure 2.7) and animates the change in orientation over time. This script vastly expedited the process of debugging the AHRS algorithm.

Once the AHRS algorithm was implemented and tested to work accurately, we could move on to implementing an orientation-based calculation for tailbeat frequency. This was done by first rotating the dataset to remove any offsets caused by the angle at which the tag affixes to the shark's dorsal fin, then looking for zero crossings in the first derivative of the shark's yaw (see Figure 2.8 for a description

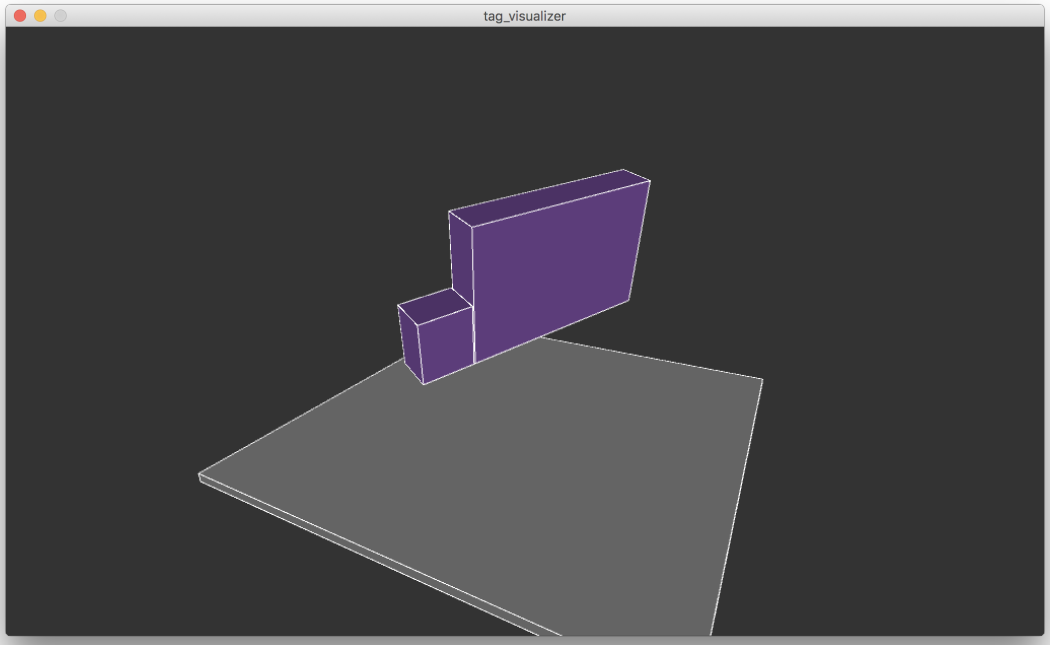

Figure 2.7: A still screenshot from the Tag_Visualizer Processing Sketch. The shape on-screen (a rough approximation of the Sharkduino tag) rotates in 3D space according to the orientation data derived from the tag's sensors.

of the derived yaw value and its derivative, and Figure 2.9 for an example of the yaw zero-crossings tailbeat frequency calculation method). This approach is methodologically identical to the Gx zero-crossing method described above, but is improved via the AHRS algorithm, granting increased resistance to noise, gyroscope drift, and mounting offset errors. A disadvantage is that the AHRS algorithm relies on more consequential assumptions than the previously explored methods, meaning its output is more vulnerable to potential sources of skew. Another weakness is that the AHRS algorithm effectively has a slew rate, meaning it can't faithfully reproduce tailbeat frequency during periods of more chaotic or high-speed shark motion without introducing an unacceptable quantity of noise.

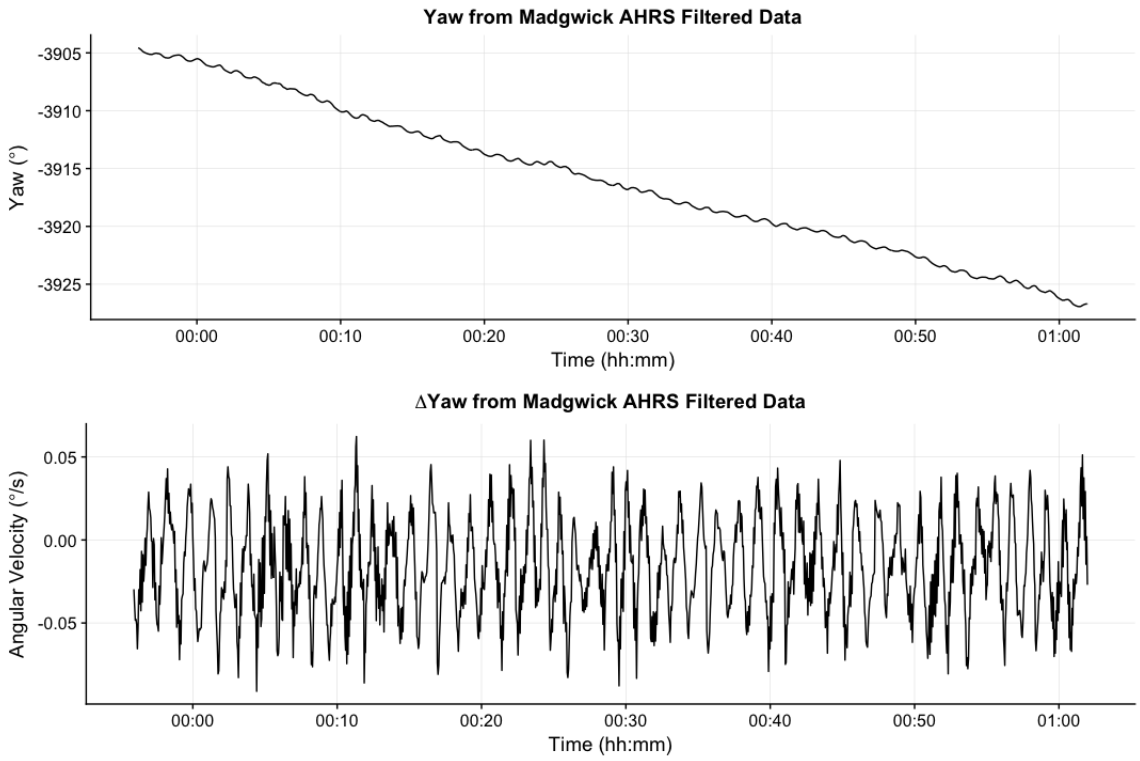

Figure 2.8: This figure shows two views of yaw derived from accelerometer and gyro data using the Madgwick AHRS algorithm. The time range covers one minute of shark data. The top plot shows us the shark's apparent yaw: the regular, low-amplitude oscillations likely correspond to shark tailbeats, while the linear downward trend is likely because sandbar sharks, when confined in a tank, will swim in a constant, circular manner.
Taking the time derivative of the top plot gives us a plot of angular velocity, shown bottom. Here, the linear component due to the shark's circular motion is no longer present, and a fairly consistent sinusoidal regime may be observed. This quantity should be a preferable alternative to G_x for calculating tailbeat frequency.

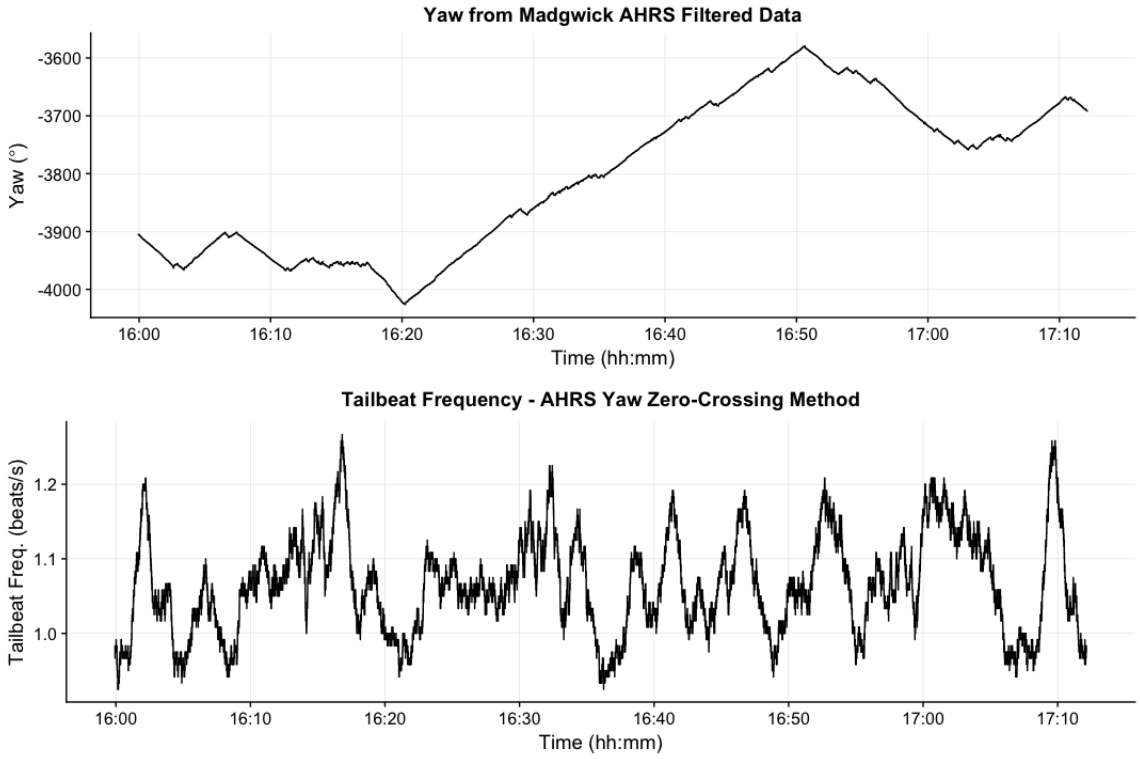

Figure 2.9: Tailbeat frequency is calculated over an hour of shark data by counting zero crossings in Δyaw. As described in Figure 2.8, this should be a preferable method for determining the metric. It's also worth noting that changes in the slope of the upper yaw plot likely correspond to events wherein the Sandbar shark performed a flip and began to circle the tank in the opposite direction.

2.4.4 Spectral Methods

Because shark tailbeats are regular and express periodic qualities, a spectral approach was also considered. Initially, the R package `psd` was used to generate sine-multi-tapered power spectrum distribution plots for short (2-10 second) windows of shark accelerometer data (see Figure 2.10). These plots were characterized by low power above 3 Hz, a few peaks in the 0.5-3Hz range, and moderately high power near 0Hz.

It soon became apparent that feeding filtered data into a multi-tapered power spectrum distribution imposes many external assumptions on the dataset, potentially skewing it to the point of invalidating any insights. With this in mind, a simple,

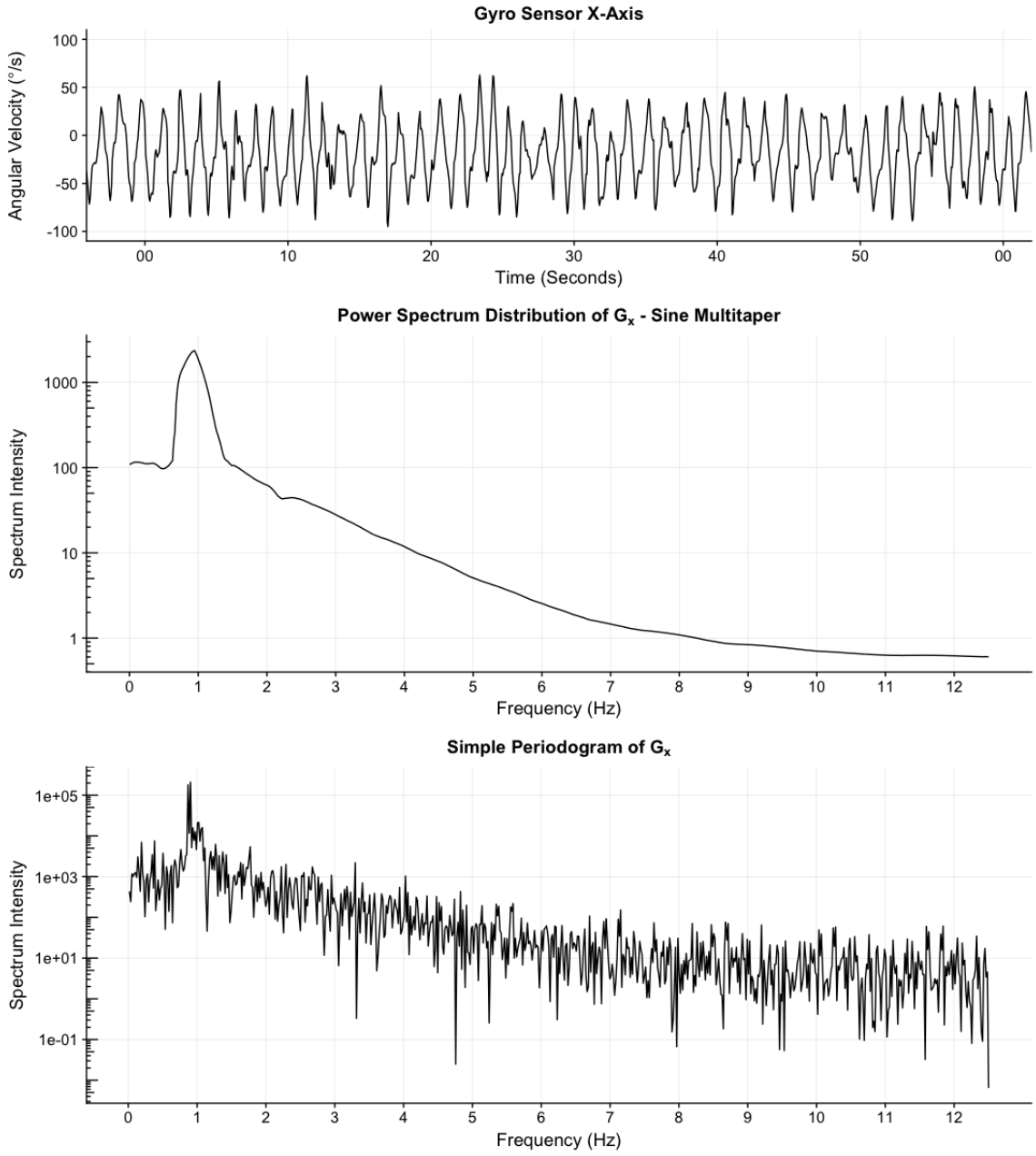

Figure 2.10: Two spectral plots are shown here, along with the source gyro sensor data. The plots cover a minute of shark swimming data. The middle plot is a sine-multi-tapered power spectrum distribution generated using R's `psd` package, while the lower plot is a simple periodogram, generated by running the source data through a FFT and only minimally processing the result. In both plots, a clear peak is visible just below 1Hz.

unfiltered periodogram was implemented, revealing similar regions as the prior PSD. Spectrograms (generated with R's `signal` package) were also used to analyze the data. A spectrogram uses vertical slices of color to represent the frequency content of a signal over time. The spectrograms we created exhibited the same general composition as the prior PSDs and periodograms, but clearly displayed a persisting spectral line at around 1Hz (see Figure 2.11). This spectral line also appeared to sink as the dataset moved into the evening hours. It is possible that this line is caused by shark's tail beats, and that tailbeat frequency slows as the shark becomes quiescent, but more analysis must be performed to better evaluate this theory.

A major disadvantage of the spectral approach is that, despite the ease with which qualitative insights may be derived from spectral visualizations, quantitative

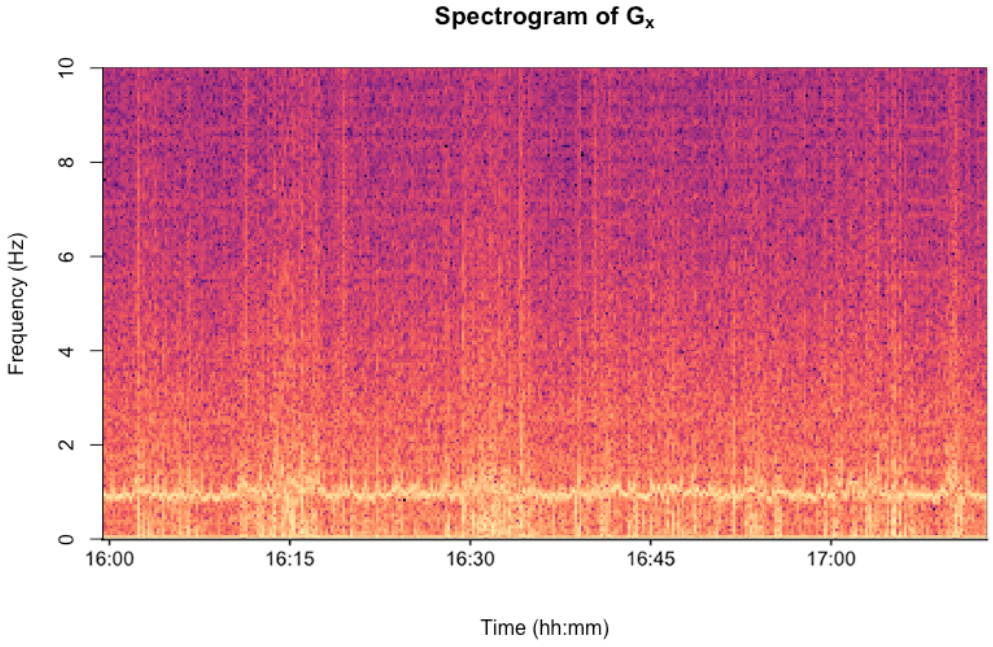

Figure 2.11: A spectrogram of shark G_x data over an hour-long time window. The source gyro sensor data for this plot is visible in Figure 2.6, and it may be worth comparing these two figures. Moreover, a persistent spectral harmonic is present just below 1Hz.

measures are much more difficult to accurately extract via this approach. Although a spectral line is clearly visible in the above spectrogram, extracting this line effectively and without many external assumptions proves difficult. Two methods of doing this were examined: a top Nth peak approach, and an approach using frequency distance from the last identified peak. Both methods provided unsatisfactory results.

2.4.5 Post-Processing Techniques: Advantages and Drawbacks

A number of post-processing techniques were investigated in order to improve the accuracy of tailbeat frequency calculation. Most notably, 3rd order Butterworth high-pass and low-pass filters were used to attenuate frequency content in frequency ranges where tailbeats seemed unlikely (less than 0.1Hz and above 6Hz). While this method did improve the consistency of the metric, it also limits its predictive power. Additionally, filtering appears to impart a slight rightward phase shift (See Figure 2.12). We conclude that filtering should be used as a post-processing step only for metrics wherein the agreement with a ground truth has already been confirmed. This way, the advantages of filtering may be derived without potentially making a poorly conceived metric look like a well-calculated one.

Various tapering and smoothing techniques were also employed to reduce artifacting and noise in the power spectrum distributions generated as part of the spectral investigation. These included AR fitting, Daniell smoothers, a split cosine bell taper, and a sine multi-taper (See Figure 2.10 for an example). While these techniques improved the readability of the resulting figures, they also obscured a great deal of potentially useful information. As a result, all tapering techniques, save for the bell taper, were abandoned in favor of raw periodograms.

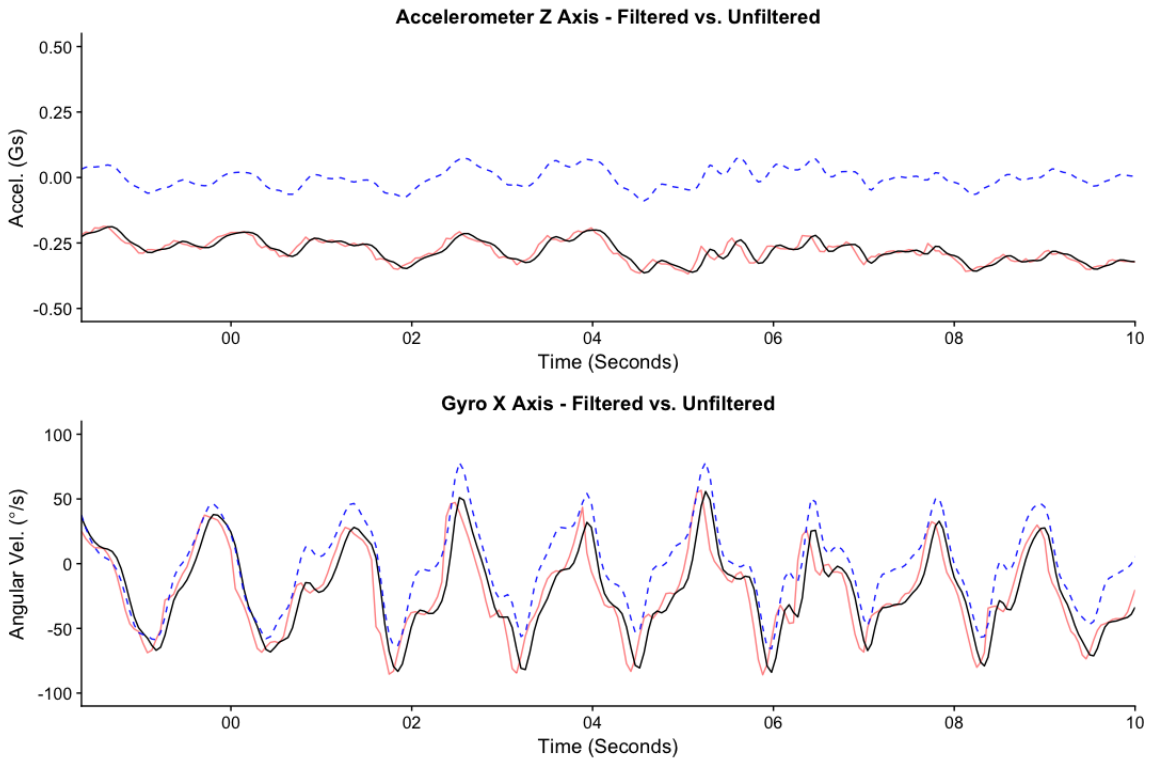

Figure 2.12: Here, some basic filters for accelerometer and gyroscope data are compared. In each plot, the red line depicts the original, unfiltered sensor data, and the black line represents the data after being sent through a 3-pole low-pass Butterworth filter. Additionally, the dashed blue line represents the sensor data after both a low-pass and a high-pass Butterworth filtering step. In preliminary tests, high and low-pass filtering with conservative cutoff values (e.g., 0.1Hz and 6Hz, respectively) appeared to increase stability without significantly altering the results range for all tailbeat frequency calculations – except for the A_z method, where filtering significantly lowered the calculated value.

Chapter 3

Results and Conclusions

3.1 Conclusions

In this report, several aspects of data analysis for the Sharkduino Animal Tag System were explored. Processes for importing, cleaning, and visualizing tag data were explained, and the development of tools to further these aims was discussed.

Two key metrics, ODBA and Tailbeat Frequency, were investigated. Calculating ODBA was relatively straightforward, but tailbeat frequency proved to be more difficult to ascertain, primarily due to the lack of "ground truth" data for comparison and verification.

Three methods of tailbeat frequency were discussed: Naïve accelerometer and gyro, AHRS yaw, and spectral. Given a one-hour range of sandbar shark swimming data, these methods gave values ranging from over 3Hz (naïve accelerometer) to just under 1Hz (spectral). While not conclusive, this set of methods establish a range for the correct value for tailbeat frequency. Furthermore, the most sophisticated calculation methods (AHRS yaw and spectral) give average values that are only 0.2Hz apart (0.9Hz $\pm$ 0.1Hz for spectral vs. 1.1Hz $\pm$ 0.2Hz for AHRS yaw), suggesting the actual value may lie within this range.

Tapering and filtering methods for tailbeat frequency metrics were also discussed.

These methods can be advantageous to metric stability, but also increase the number of external assumptions levied on the metric calculation. For this reason, filtering and tapering are avoided in Sharkduino metric calculation until a metric may be validated with ground-truth data, such as video recordings, and confirmed to correctly proxy the relevant biological phenomenon.

3.2 Further Work

This work is only a first step toward a complete understanding of animal tag data from sandbar sharks. Some eventual goals were known at the start of this project, but many promising leads also became apparent in the process of reaching the results presented in this paper.

One definite improvement would be to compare the metrics presented in this research with "ground truth" data, such as a tailbeat count manually collected from video recordings of a tagged shark's swimming behavior. Because sand bar sharks are migratory and only remain in the Chesapeake Bay area during the summer months, getting this kind of corroborating video footage has proved difficult thus far. This type of analysis would, however, help us refine the metrics already developed, as well as providing ideas for new biologically significant metrics.

Machine learning (ML) based data analysis has long been an eventual goal for Sharkduino data analysis. In this vein, the research group has already developed a supervised machine learning solution that uses a support vector machine (SVM) algorithm and training data derived from videotaped shark activity to classify various behaviors in the animals. However, most of this analysis relies exclusively on accelerometer and time data. A logical next step would be to use the metrics developed in this work as the basis for machine learning models, hence making use of the tag's gyro sensor to improve classification.

Another form of machine-learning based data analysis is so-called *"unsupervised"* ML, wherein an algorithm classifies data without the use of a training dataset. In our case, the ideal algorithm would scan raw data, derived metrics, and any other features available to it, and then call out points of interest, anomalies, and other novelties in the data, which could then be analyzed further by hand. The ODBA, tailbeat frequency, and orientation data derived in this research would serve as derived metrics or features for such an algorithm. While this initiative is still in its early stages, unsupervised machine learning appears to be a promising direction for further research.

Looking to the marine biology side of Sharkduino, a key long-term goal is to better understand sandbar shark behavior. An important next step from developing metrics to do this is to test these metrics to see if they are, in fact, biologically significant. If so, we may be able to use animal tag metrics like ODBA to extract powerful insights about sandbar sharks' daily behavior, digestion, feeding and mating activity, and quiescence. This will bring the project closer to its goal of better understanding the lives and behavior of sandbar sharks.

Bibliography

[1] Nicholas M. Whitney et al. "Identifying shark mating behaviour using three-dimensional acceleration loggers". English. In: *Endangered Species Research* 10 (Feb. 2010), pp. 71–82. DOI: 10.3354/esr00247.

[2] *X16-1D USB Accelerometer Data Logger User Manual*. Rev. B. Gulf Coast Data Concepts. Mar. 2016. URL: http://www.gcdataconcepts.com/GCDC_X16-1D_User_Manual.pdf.

[3] William Laney. "Continuing Improvements of the Sharkduino Animal Tag System". Bachelor's Thesis. The College of William and Mary, Dec. 2017.

[4] Susumu Kato, Stewart Springer, and Mary H. Wagner. *Field Guide to Eastern Pacific and Hawaiian Sharks*. Washington, DC: U.S. Fish and Wildlife Service, Dec. 1967.

[5] *DS3234: Extremely Accurate SPI Bus RTC with Integrated Crystal and SRAM*. 3rd ed. Maxim Integrated. July 2010.

[6] Ryo Kawabe et al. "Direct measurement of the swimming speed, tailbeat, and body angle of Japanese flounder (Paralichthys olivaceus)". English. In: *ICES Journal of Marine Science* 61.7 (2004), p. 10801087. DOI: 10.1016/j.icesjms.2004.07.014.

[7] Adrian C. Gleiss, Rory P. Wilson, and Emily L. Shepard. "Making overall dynamic body acceleration work: on the theory of acceleration as a proxy for energy expenditure". In: *Methods in Ecology and Evolution* 2.1 (2011), pp. 23–33. DOI: 10.1111/j.2041-210X.2010.00057.x. URL: https://doi.org/10.1111/j.2041-210X.2010.00057.x.

[8] Rory P. Wilson et al. "Moving Towards Acceleration for Estimates of Activity-Specific Metabolic Rate in Free-Living Animals: The Case of the Cormorant". English. In: *Journal of Animal Ecology* 75.5 (Sept. 2006), pp. 1081–1090. DOI: 10.1111/j.1365-2656.2006.01127.x. URL: https://www.jstor.org/stable/3838400.

[9] Lewis G. Halsey et al. "Accelerometry to Estimate Energy Expenditure during Activity: Best Practice with Data Loggers". In: *Physiological and Biochemical Zoology* 82.4 (2009), pp. 396–404. DOI: 10.1086/589815. URL: https://doi.org/10.1086/589815.

[10] Sebastian O. H. Madgwick. “An efficient orientation filter for inertial and inertial/magnetic sensor arrays”. PhD thesis. University of Bristol, Dec. 2010.

[11] Pasquale Daponte et al. “Experimental comparison of orientation estimation algorithms in motion tracking for rehabilitation”. In: *2014 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*. June 2014. DOI: 10.1109/MeMeA.2014.6860048.

[12] James Diebel. “Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors”. 2006.

[13] Sebastian O. H. Madgwick. *Quaternions*. Tech. rep. Sept. 2011. URL: http://x-io.co.uk/res/doc/quaternions.pdf.