High-throughput kinematic tracking of bird wings using inertial sensor arrays

by

Shreeram Senthivasan

B.Sc. (Hons.), University of Toronto, 2016

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

MASTER OF SCIENCE

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

(Zoology)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

June 2020

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

High-throughput kinematic tracking of bird wings using inertial sensor arrays

submitted by Shreeram Senthivasan in partial fulfillment of the requirements for

the degree of Master of Science

in Zoology

Examining Committee:

Douglas Altshuler, Professor, Department of Zoology, UBC
Supervisor

Philip Matthews, Assistant Professor, Department of Zoology, UBC
Supervisory Committee Member

Robert Shadwick, Professor, Department of Zoology, UBC
Additional Examiner

Additional Supervisory Committee Members:

Patricia Schulte, Professor, Department of Zoology, UBC
Supervisory Committee Member

Abstract

Birds accomplish an impressive diversity of flight maneuvers primarily through variation in the motion of their wings. It is for this reason that an understanding of wing kinematics is of broad interest to the study of the physics and control of bird flight. However, current optical approaches to animal motion capture struggle to automate the tracking of fixed points on the wing due to the periodic folding and occlusion of flight surfaces during flapping flight. This greatly increases the time and effort needed to record wing kinematics, as the raw data must be digitized by hand. In this thesis, I made progress towards a new high-throughput approach to recording wing kinematics using body- and wing-mounted inertial sensors. The data loggers designed for this purpose have a mass of 4g including a battery, and are capable of collecting inertial data from up to four sensors at 450Hz for 20 minutes. An accompanying set of R functions were developed to estimate the orientations of the body and wing segments from the raw inertial data, which in turn were used to estimate joint angles over time. These tools were validated against an optical motion capture system and were able estimate the orientation of individual bodies within 3° and angles between bodies within 6° in controlled tests. However, the tools could not be used to record the wing kinematics of pigeons because the angular velocity of the wings exceeded the sensing range of the gyroscopes used in the current design. Specialized high-range gyroscopes are commercially available and could be incorporated into future designs to overcome this limitation. Inertial motion capture has

the potential to be a high-throughput, cost-effective, and portable alternative to high-speed video for recording wing kinematics in freely flying birds. This approach could also be used to record detailed kinematics in other animal systems where optical motion capture is infeasible, such as in situations with poor visibility or to study behaviours that occur over large spatial scales.

Lay Summary

Understanding the subtle changes birds make to their wing strokes during flight is important to furthering our understanding of how birds move and maneuver in the air. However, current methods for recording three-dimensional animal motion using high-speed camera arrays are not well suited for tracking the wings of flying birds. Bird wings fold, flex, and can be obscured during flapping flight, which makes it difficult for algorithms to recognize fixed points on the wing without direct human supervision. This adds considerable time and effort to recording wing motion. In this thesis, I made progress towards developing a new high-throughput way of recording animal motion using purpose-built body-mounted data loggers and data processing software. Although further changes to the hardware are needed to record the motion of bird wings in flight, I demonstrate that this approach could provide a cost-effective, portable, and automated alternative to current methods of animal motion capture.

Preface

This dissertation is the original, unpublished work of the author, S. Senthivasan, with supervision and feedback from D.L. Altshuler. B. Yechuri contributed the final circuit board layouts presented in Chapter 2. All of the research was conducted in the Altshuler Laboratory at the University of British Columbia, Vancouver campus. The work with live birds described in Chapter 2 was approved under the UBC Animal Care Certificate number A19-0113.

Table of Contents

List of Tables

List of Figures

Acknowledgements

I would like to thank Doug, the members of the Altshuler Lab, and my friends throughout the department and beyond. You were all friendly, supportive, and an absolute pleasure to spend time with both in and out of the lab. I would also like to thank my friends and family back home who always supported me even when neither of us really knew what exactly it was that I was doing here.

Chapter 1

Introduction

Three independent evolutions of powered flight are represented in extant animals [1] and all three of these taxa achieve this unique form of locomotion through the flapping of analogous, specialized appendages we call wings [2–4]. Unlike in human-engineered systems that typically disentangle the structures responsible for thrust generation (e.g., propellers, jet engines), lift generation (e.g., wing surfaces, spoilers, flaps), and attitude control (e.g., ailerons, elevators, rudders), flighted animals accomplish all of these tasks primarily by variation in the motion of the wings [1]. It is for this reason that wing kinematics are of broad interest in the study of the physics and control of animal flight [5–7].

However, current methods for recording animal motion using arrays of high-speed cameras [8, 9] struggle to automate the collection of bird wing kinematics, in part due to the considerable morphing of wings during flapping flight [10]. This problem is generally overcome by manually assessing every frame from every camera in the array and correcting the digitization as necessary. This manual digitization requires considerable time and effort, which is reflected in the limited sample sizes of most studies that make use of detailed bird wing kinematics [11–14].

In contrast, high-throughput methods exist for collecting wing kinematic data

from tethered flies which allows in part for much larger samples sizes and more detailed analyses [15, 16]. The development of automated methods for recording body position [17] and aerodynamic forces [18] in bird flight similarly enabled researchers to address large comparative [19] and more detailed mechanistic [20] questions.

In this thesis, I made progress towards the development of a new set of tools for the high-throughput collection of wing kinematic data from freely flying birds. I use body-mounted inertial sensors to track the orientations of the body and wing segments during flight, which in turn could be used to reconstruct the overall motion of the wings. As part of this approach, I rely heavily upon previous work in two branches of mathematics that may be unfamiliar to many readers—namely hidden state estimation using recursive Bayesian inference and calculations on orientations with unit quaternions. I briefly introduce these two topics as they relate to the current problem in the following two sections.

1.1 Recursive Bayesian estimation

Recursive Bayesian estimation is a general framework for estimating a time varying process, or state, that cannot be measured directly [21]. The approach here is to use information about how the system changes over time as well as direct observations of related variables to recursively refine our estimate of the hidden state.

This is useful to us as we cannot use inertial sensors to directly measure how birds position their wings at each point in a wing beat. Instead, we directly measure the angular velocity, linear acceleration, and magnetic heading of individual body segments, which provide us with indirect information about the motion of the wings as a whole. Recursive Bayesian estimation provides this link between what we can measure and what we want to know.

1.1.1 Modelling systems with recursive Bayesian estimation

To apply the recursive Bayesian framework, we need to be able to make two assumptions about the structure of the causal relationships in the system. First, we require that the probability of the state at any given time depends only the probability of the state in the previous time step. This can be written as

$$p(x_t \mid x_{1:t-1}) = p(x_t \mid x_{t-1})$$

where $p(a \mid b)$ is the probability of a given b, x_t is the state at time t, and $x_{1:t}$ is the collection of states from time 1 to time t. This is also known as the Markov property and greatly simplifies the process of modelling how the state evolves over time [22]. Substituting $(t+1)$ for t above gives us the expression for the transformation that maps the probability density of the current state to the same for the next time step.

$$p(x_{t+1} \mid x_t) \tag{1.1}$$

We call this transformation the transition function and it must be constructed to model the system we would like to explore. This in turn requires that we have some prior knowledge about how the system behaves.

The other key assumption is that the direct measurements, or the observation of the system, made at any time step depend only the current state. This can be written as

$$p(y_t \mid x_{1:t}, y_{1:t-1}) = p(y_t \mid x_t)$$

where $p(a \mid b, c)$ is the probability of a given the joint probability of b and c, and y_t is the observation at time t. This is important as it allows us to use Bayes theorem to use express the probability of the current state in terms of the current

observation [23].

$$p(x_t \mid y_t) = \frac{p(y_t \mid x_t)p(x_t)}{p(y_t)} \tag{1.2}$$

Note that $p(y_t \mid x_t)$, or the probability of the current observation given the current state, is a relationship in our system and must be modelled just like the transition function.

Substituting the transition function (1.1) at time $(t-1)$ for the probability of the state prior to observation incorporation in (1.2), we get an expression for the posterior probability of the state given both lines of evidence.

$$p(x_t \mid y_t, x_{t-1}) = \frac{p(y_t \mid x_t)p(x_t \mid x_{t-1})}{p(y_t)} \tag{1.3}$$

This is the expression that we wish to solve for at every time step using recursive Bayesian estimation. However, $p(x_{t-1})$ or the probability of the state at the previous time step in (1.3) could also be rewritten using (1.3) itself.

$$p\Big(x_t \mid y_t, p(x_{t-1} \mid y_{t-1}, x_{t-2})\Big) = \frac{p(y_t \mid x_t)p\Big(x_t \mid p(x_{t-1} \mid y_{t-1}, x_{t-2})\Big)}{p(y_t)}$$

$$p(x_t \mid y_t, y_{t-1}, x_{t-2}) = \frac{p(y_t \mid x_t)p(x_t \mid y_{t-1}, x_{t-2})}{p(y_t)}$$

Repeating this substitution recursively, we get

$$p(x_t \mid y_{1:t}, x_0) = \frac{p(y_t \mid x_t)p(x_t \mid y_{1:t-1}, x_0)}{p(y_t)}$$

where x_0 is the initial state. Note that the probability distribution of x_0 can be taken to be the uniform distribution if we do not have any prior knowledge about the initial state. This, admittedly abstractly, demonstrates how recursive Bayesian estimation can be used to estimate a hidden state using only sequential indirect

observations. A visual demonstration of this idea using a toy example is provided below in Section 1.1.2.

Finally, if we choose our state x_t such that it includes both the hidden variables of interest and the observable variables y_t, we can see that our two key assumptions are equivalent to assuming that the system can be modelled as a partially observed Markov chain [22]. Such a Markov chain is shown visually in Figure 1.1 using a state transition diagram.

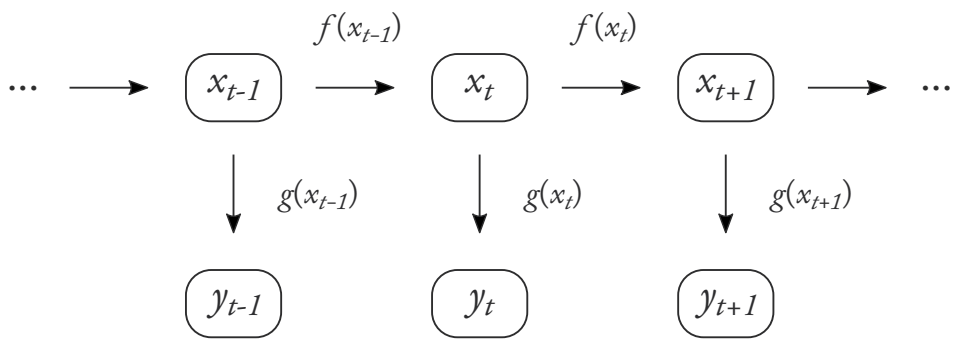

Figure 1.1: State transition diagram for a partially observed Markov chain. The true state x_t evolves over time following some transition function f that depends only the previous state. The observation y_t is the subset of the true state that can be measured directly and is related to the current state by the function g. These two functions are sufficient to characterize the Markov chain [22].

1.1.2 A visual example with Kalman filters

The Kalman filter is one of the simplest and most widely-used implementations of recursive Bayesian estimation [24–26]. By requiring the models to be linear and the probabilities to be normally distributed, Kalman filters achieve impressive computational efficiency, with computation time increasing linearly with the complexity of the model [27]. It is likely for this reason that the navigational computer on board the Apollo 11 spacecraft used a Kalman filter to estimate its trajectory [28], despite having less computing power than a TI-84 Plus graphing calculator [29].

To help develop an intuition for how recursive Bayesian inference can be used

to refine our estimates of hidden variables, I provide a visual example to applying a Kalman filter to a simple estimation problem. In this example we are interested in estimating the velocity of an animal using a video recording of it running along a straight track. Notice that while the video clearly captures information about the velocity (hidden state) of the animal, no one frame (observation) in isolation provides any reliable measure of the same. Each frame does however provide us with the position of the animal along the track, which can be related to the hidden state.

We start by considering everything we know given only the first frame of the recording. We have a pretty clear idea of the position of the animal, but it is important to recognize that the measurement will not be perfect. The measurement error in this example could be due to motion blur or lens distortion along the margins of the frame. In contrast, we have no information about the velocity of the animal or any covariance between the animals velocity and position. Figure 1.2A shows a two-dimensional normal distribution that visualizes what we know about these two variables.

Next, we apply the transition function to generate a prior estimate for the state in the second frame. As we have no reason to suspect that the velocity of the animal will change one way or the other over a time step, our transition function should reflect this by leaving the expected velocity unchanged over the transition. Our expectation for the position however will depend linearly on the velocity of the animal as well as the previous position. To visualize this transition, we can pick points along the current probability distribution and track how they are transformed by the transition function. We can then use the transformed points to visually reconstruct the prior probability of the state in the second frame. Figure 1.2B shows this procedure as well as the resulting transitioned probability density distribution.

We can now overlay the actual measurement information from the second frame

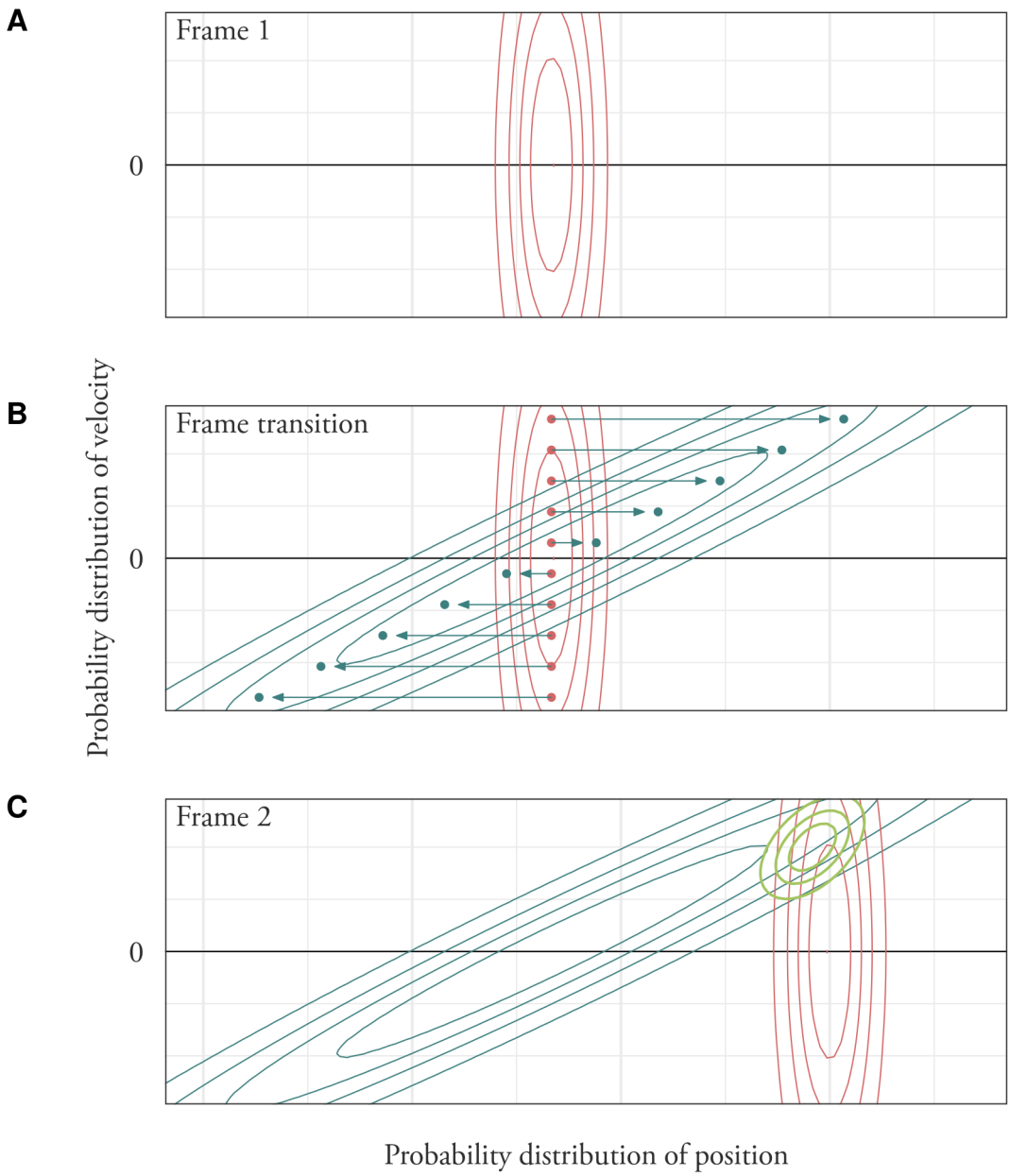

Figure 1.2: A visual example of a Kalman filter applied to a simple estimation problem. Bivariate normal distributions describing estimates of the state are shown using contour plots. Points are discrete samples from the normal distributions and are used to visualize transformations on the probabilities, which are represented by the corresponding arrows. Colour is used to indicate the information used to construct a given plot element. Red indicates an observation, blue indicates a state transition, and green indicates that both lines of evidence were used. The sequence shows (A) an initial observation, (B) the modelled transition between observations, and (C) the incorporation of a new observation. The filter is continued by repeating the transition and observation incorporation steps for all remaining observations.

over our prior estimate from the previous transition. As before, this observation provides reliable information about the position of the animal but no information about the velocity whatsoever. Combining these two probability distributions however provides us with a reasonably precise estimate of the velocity of the animal in the second frame. This measurement incorporation is illustrated in Figure 1.2C.

This two-step procedure of transition and measurement incorporation can then be repeated for each subsequent frame to estimate velocity over the whole recording. Note that the hidden state is never directly measured, but information is extracted from the observations as well as the modelled relationship between the observation and the hidden state.

1.1.3 Extension beyond Kalman filters

Despite the computational benefits of using Kalman filters, the two additional assumptions made regarding model linearity and normally distributed errors are not always appropriate for the system being modelled [30]. Other recursive Bayesian estimation algorithms may better approximate the system in these situation, though this flexibility generally comes with some cost to computational efficiency. The extended and unscented Kalman filters for example relax the need for model linearity in different ways [30, 31], while others like the Monte Carlo filter and dynamic generalized linear models use different probability distributions to model systems where variables are not normally distributed [32, 33].

We have also only considered recursive Bayesian filters up to this point. Filters in this context are algorithms that solve the filtering problem [34]

$$p(x_t \mid y_{1:t}) \qquad (t = 1, \ldots, T)$$

where T is the final time step in a given sequence of observations. In words, filters estimate each state using all of the observations up to and including the

current time step. Predictors are another set of algorithms within the recursive Bayesian framework that solve the prediction problem [34]

$$p(x_t \mid y_{1:t-1}) \qquad (t = 1, \ldots, T)$$

This is similar to the filtering problem, except that observation from the current time step is not available. In contrast, smoothers use observations from the entire sequence to estimate each state to solve the smoothing problem [34]

$$p(x_t \mid y_{1:T}) \qquad (t = 1, \ldots, T)$$

Filters will generally outperform predictors and smoothers will generally outperform filters, but the additional observations needed to use these algorithms are not always available when the estimate is needed.

The algorithms discussed thus far also implicitly assume that the model parameters are known more or less exactly. In linear systems with unknown parameters it is generally tractable to calculate a maximum likelihood estimate of the parameters given the observations [35]. In more complex systems however, it may be necessary to simultaneously estimate both the parameters and the state using a recursive Bayesian framework [36, 37]. This is not a perfect solution as it can result in poor performance in early time steps where neither the state nor the parameters are well known. One solution to this problem is to first use all of the available data to recursively generate a probability distribution of the parameters before using this posterior parameter distribution to recursively estimate the state as usual [38]. Although this solution could be used to filter or predict the state, as the posterior parameter distribution is built using the entire sequence of observations, it suffers from the same dependence on the availability of future observations as smoothing algorithms.

1.1.4 Summary

Recursive Bayesian estimation provides a probabilistic framework for estimating variables of interest using indirect measurements taken over time. However, the actual calculations used in a given implementation will depend on the system and the way we choose to model it. To reconstruct bird wing kinematics from inertial data using this framework, we need a mathematical representation for kinematics that can be related back to our direct observations. In the following section, I introduce a number system that can be used for such a representation.

1.2 Kinematics with quaternions

One approach to quantifying kinematics is to start by modelling the body as a series of rigid body segments connected by simple joints [39]. Although body segments are never truly rigid [40] and biological joints do allow some lateral motion [41], this model is a useful approximation for most vertebrate systems.

Under this model, all we need to reconstruct the kinematics of an animal are the angles of all the relevant joints measured along each axis and recorded over time. A useful observation is that we can easily calculate joint angles if we know the orientations of the surrounding two body segments. For example, if you know that your forearm and upper arm are aligned, you know that your elbow is straight regardless of what you are doing with your shoulder. Accordingly, the problem of quantifying kinematics quickly becomes a problem of tracking the orientation of individual body segments over time. This is especially convenient as recursive Bayesian estimation can be used to estimate orientation from inertial data [42, 43]. What we need now is a set of tools for working with orientation.

1.2.1 Representing orientations

Representing and manipulating orientations in three dimensions can often feel unintuitive, in part because it is easy to confuse the mathematical idea of an orientation with orientations in common parlance. A useful analogy for thinking about the former is to view orientation as the angular counterpart to vector position. Just as position in three dimensions can be represented with a series of translations along three perpendicular axes, orientation can be thought of as a comparable series of rotations. This is made explicit in the Euler angle system [44]. Although this is a valid and complete system for working with orientations, the analogy is not perfect. The order in which the three rotations are applied cannot be changed without changing the resulting orientation [44]; it is ambiguous whether the three axes should rotate with the body or remain fixed as you apply the three rotations [44]; and the space is discontinuous when there is a rotation of $\pm 90°$ in pitch [45], but not along the other two axes.

An alternative approach is to view an orientation as a transformation between two frames [46]. Although a frame is properly any set of basis vectors for a given space [47], for the purposes of this project I will restrict this definition to mean three mutually perpendicular unit vectors in three-dimensional space. We can represent physical objects with these frames by choosing the three vectors to align with some natural axes of the object, such as along the walls and floor a square room, or the anteroposterior, mediolateral, and dorsoventral axes of an animal. I will also restrict our discussion to frames that satisfy the right-hand rule to avoid issues with mirrored frames [48].

This analogy makes it clear that an orientation requires at least two frames to make sense. A single object floating in a vacuum can be assigned a corresponding object frame, but without some frame of reference it is not clear what its orientation should be. This reference frame can be external, such as the frame of another object

floating by, or it can be the frame of the first object at another point in time. Both of these choices are valid, but the resulting orientations represent completely different transformations. This is why it is helpful to be explicit about which frames are being discussed, especially as we start manipulating orientations.

The other necessary piece of information needed to represent an orientation is the transformation between the two frames [46]. The ambiguity we saw in the Euler angle system stems from the fact there are many ways decompose this transformation into three sequential rotations around perpendicular axes. This highlights the need for being clear about which conventions are being used when working with Euler angles [44].

Another option is to represent the transformation as a single smooth rotation between the two frames characterized by an axis and magnitude of rotation [49]. This is useful as every rotation will have a unique axis-angle representation that minimizes the magnitude of the rotation. The only exception is when the magnitude of the rotation is 180°, in which case rotating either way around the axis of rotation is equivalent. In contrast, the Euler angle system has infinitely many equivalent representations when the pitch is ±90° due to gimbal lock [45]. See Figure 2.7 for a visual demonstration of this problem.

Although the axis-angle representation is an intuitive and complete description of the transformation between two frames, it does not provide an algorithm for actually applying this rotation to vectors in three dimensions or other orientations represented in this way [49]. Rotation matrices provide such an algorithm by way of matrix multiplication [47]. This system also represents the rotation between frames as a single transformation, which overcomes the issue of gimbal lock. However, converting between axis-angle representations and rotation matrices is relatively slow [50], as is matrix orthogonalization without bias [51]. As both these calculations will be used very often in the following methods, I chose to use a different system

for representing orientation.

1.2.2 Unit quaternions

Quaternions are a four-dimensional extension of complex numbers, written as

$$\begin{aligned}\vec{q} &= w + x\hat{i} + y\hat{j} + z\hat{k} \\ &= \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix}\end{aligned} \tag{1.4}$$

where $\hat{i}$, $\hat{j}$, and $\hat{k}$ are unit numbers perpendicular the real numbers, much like i in complex numbers [52].

As with rotation matrices, a unit quaternion encodes all the information needed to rotate between two frames [46]. One way to intuit the approximate transformation encoded by a given unit quaternion is to consider what each component represents in isolation. The unit quaternion $[1\ 0\ 0\ 0]^T$ represents no rotation whatsoever, while $[0\ 1\ 0\ 0]^T$, $[0\ 0\ 1\ 0]^T$, and $[0\ 0\ 0\ 1]^T$ represent 180° rotations around x, y, and z respectively. Note that the transpose (q^T) of these quaternions are used here for compactness. Other unit quaternions can then be thought of as mixes of these four orientations weighted by their corresponding components.

Throughout this paper, I use the notation ${}^{B}_{A}\hat{q}$ to express the unit quaternion representation of the transformation from frame A to frame B. This is equivalent to the orientation of frame B in the reference frame A [46].

1.2.3 Axis-angle conversion

We can construct a quaternion from an axis-angle pair using the following definition:

$$ {}^{B}_{A}\hat{q} = \begin{bmatrix} cos(\theta/2) \\ \hat{n}\ sin(\theta/2) \end{bmatrix} \tag{1.5} $$

where θ is the magnitude of the rotation between frame A and frame B and $\hat{n}$ is the three-dimensional unit vector parallel to the axis of rotation [46].

This can also be used to construct the inverse of a quaternion, or in other words the transformation from frame B back to frame A [49]. We do this by negating the magnitude of the rotation in the axis-angle representation.

$$ \begin{aligned} {}^{B}_{A}\hat{q}^{-1} &= {}^{A}_{B}\hat{q} \\ &= \begin{bmatrix} cos(-\theta/2) \\ \hat{n}\ sin(-\theta/2) \end{bmatrix} \\ &= \begin{bmatrix} cos(\theta/2) \\ -\hat{n}\ sin(\theta/2) \end{bmatrix} \end{aligned} \tag{1.6} $$

This allows us to efficiently compute the inverse of a unit quaternion simply by negating the final three components. Calculating the inverses of rotation matrices, for comparison, is considerably slower [47].

1.2.4 Applying quaternion rotations

Using our analogy of orientations as transformations, we can also use unit quaternions to rotate other objects. With rotation matrices, this rotation was accomplished using matrix multiplication [47]. With unit quaternions, we use quaternion multiplication, also known as the Hamilton product [49].

Quaternion multiplication is similar to matrix multiplication in that it is not commutative, ie

$$q_a \times q_b \neq q_b \times q_a$$

where $\times$ is the Hamilton product [47, 49]. Also as with matrices, we apply quaternion transformations by multiplying specifically from the left [46, 47]. For example, if we start with a unit quaternion representing the transformation from frame A to frame B (i.e., ${}^{B}_{A}\hat{q}$), we can apply another transformation on the left to move from frame B to a new frame C (i.e., ${}^{C}_{B}\hat{q}$).

$${}^{C}_{B}\hat{q} \times {}^{B}_{A}\hat{q} = {}^{C}_{A}\hat{q} \tag{1.7}$$

Reusing the analogy of orientations as transformations, this composition can be viewed either as either a series of rotations from frame A to B to C, or as a change of reference frame for the orientation of C from frame B to frame A.

We can also apply quaternion rotations to vectors in three dimensions, but the process is a little more complicated as the Hamilton product is only defined for quaternions. To get around this, we first cast the three-dimensional vector as a quaternion with the real component set to zero [46]. For an arbitrary vector in three dimensions $\vec{v}$, this looks like so:

$$\begin{bmatrix} 0 \\ \vec{v} \end{bmatrix}$$

The procedure for rotating this vector is also different from the procedure for

rotating a unit quaternion and is known as conjugation [46].

$$\begin{bmatrix} 0 \\ \vec{w} \end{bmatrix} = {}^{B}_{A}\hat{q} \times \begin{bmatrix} 0 \\ \vec{v} \end{bmatrix} \times {}^{B}_{A}\hat{q}^{-1} \tag{1.8}$$

where $\vec{v}$ is an arbitrary three-dimensional vector and $\vec{w}$ is the new rotated vector [46]. As before, there are a couple different ways of interpreting this transformation. We could choose to view $\vec{w}$ as $\vec{v}$ rotated by the transformation from frame A to frame B, or we could could view $\vec{v}$ as a vector expressed in frame A and $\vec{w}$ as that same vector now expressed in a new coordinate system, B. Both of these interpretations will be useful to us later.

1.2.5 Calculating angles across a hinge with quaternions

To demonstrate how unit quaternions can be used to solve problems, I show the procedure for calculating the angle across a simple hinge that can only rotate around a single axis.

We begin by naming three frames in our system: the left side of the hinge (L), the right side of the hinge (R), and some external frame of reference (E). Suppose we have been been tracking the two sides of the hinge independently and know the orientation of each side of the hinge relative to the external frame. In other words, we know ${}^{L}_{E}\hat{q}$ and ${}^{R}_{E}\hat{q}$.

Our first step is to get rid of the external frame of reference E, because it has no bearing on the angle of the hinge. Notice that the two orientations can be interpreted as the transformations from frame E to L and from E to R. As they both have frame E in common, we can manipulate the two unit quaternions to find the transformation from L to E to R. To do this we need to invert the first quaternion using (1.6) to find the transformation from L to E and then apply the second quaternion using (1.7) to go from E to R.

$$ {}^{R}_{L}\hat{q} = {}^{R}_{E}\hat{q} \times {}^{L}_{E}\hat{q}^{-1} $$

We now have a unit quaternion that represents the transformation from the left side of the hinge directly to the right side. As our simple hinge can only rotate around one axis, the angle of the hinge is simply the magnitude of the rotation described by the aforementioned transformation. Using the axis-angle representation of a unit quaternion (1.5), we can solve for this angle θ using just the first component of the unit quaternion.

$$ {}^{R}_{L}q_w = cos\left(\frac{\theta}{2}\right) $$

$$ \theta = 2\ cos^{-1}\left({}^{R}_{L}q_w\right) $$

where ${}^{R}_{L}q_w$ is the first component of the unit quaternion ${}^{R}_{L}\hat{q}$, as introduced in (1.4).

1.2.6 Summary

Unit quaternions are a robust and computationally efficient way of representing and manipulating orientation. By modelling an animal as a series of rigid bodies connected by joints, quaternions can also be used to compactly describe kinematics. This, along with recursive Bayesian estimation, provides us with all the mathematical tools we need to implement inertial motion capture using body-mounted data loggers.

Chapter 2

Tool development

2.1 Introduction

Current studies of animal kinematics commonly rely on data from arrays of high-speed cameras to reconstruct the position of marked points on the body using Direct Linear Transformation (DLT) [53–55]. Although these methods have been shown to be quite reliable [9], they require direct line of sight of markers by multiple cameras simultaneously to reliably reconstruct position. This is problematic for markers placed on the wings of birds as they are prone to occlusion by the wings, body, and feathers over the course of a wing stroke. Although this problem can be mitigated to some degree by using greater numbers of cameras and perspectives, markers can still be completely hidden from view during wing flexion or if the wings come together at the top or bottom of a wing stroke. Using many inward facing cameras to improve marker tracking also requires considerable space and equipment to record wing kinematics in a relatively small flight volume, which can limit the diversity of species and behaviours that can reliably be recorded.

The other key constraint that makes DLT inviable for high-throughput recording of bird wing kinematics is the considerable time required to digitize the resulting

footage to ensure that all markers are appropriately labelled in every frame of every perspective [5, 11, 13, 14]. Newer implementations such as DeepLabCut [8] and the latest iteration of DLTdv [9] as well as proprietary systems such as OptiTrack make significant strides in automating this process for most use cases. However, due to the inconsistency of marker capture across frames as a result of occlusion, considerable effort is still required to check and clean the results by hand. Methods that rely on shape-based feature recognition, such as those used in some studies of insect flight [56], improve upon these shortcomings but cannot reliably deal with the degree of wing morphing seen in bird flight [10] or even some insect taxa [57]. Similar photogrammetric [58] and structured light [59] approaches that do not attempt feature recognition can be used to efficiently record the body profile of flying birds, but isolating the wings from the body and extracting kinematic data again requires manual effort.

An alternative to these optical approaches would be to use body-mounted inertial sensors to track the orientations of the major skeletal components of the wing over time. Although inertial motion capture has recently gained considerable popularity in human exercise science [60–62], there has been considerably less interest in other fields. Further, the few studies that apply these methods to other systems have been restricted to large mammals [63, 64], perhaps due to size limitations. However, the growing popularity of microelectronics and small-batch circuit assembly services makes it both feasible and cost-effective to design and manufacture complete inertial data logging systems small enough to be mounted on birds.

Developing customized hardware for inertial motion capture comes with its own challenges when it comes to estimating orientation from the raw sensor data. Most human applications of inertial motion capture depend on proprietary sensor fusion algorithms for orientation estimation that come as part of commercial inertial motion capture systems [60, 61]. In contrast, most open-source code for estimating

orientation using inertial data optimize for computational speed over accuracy [42, 43]. This is likely driven by use cases where orientation must be estimated in real-time but where precision is less critical, such as in drone stabilization and for user interaction with mobile applications. Instead, other recursive Bayesian approaches exist that better optimize for accuracy and account for uncertainty in model parameters [38, 65]. However, this comes at the cost of being far slower to compute, which is likely why there is currently no openly available code to implement orientation estimation using these methods.

To develop a high-throughput, portable, and minimally invasive means of recording wing kinematics, I designed and programmed body-mounted sensors to collect inertial data from the body and wings of birds in flight and implemented a Monte Carlo recursive Bayesian smoother with parameter learning in R to estimate kinematics from the raw inertial data.

2.2 Hardware

Inertial motion capture models the subject as a series of rigid bodies connected by joints that can freely flex and extend over time [39]. Inertial data collected from each of the rigid bodies of interest are used to estimate the orientation of each body segment in space at every time step. Joint dynamics can then be inferred from the relative orientation of adjacent body segments and are sufficient to describe body kinematics in this model framework.

To collect the data needed for this approach, the inertial data loggers must be capable of taking inertial measurements at each rigid body; storing the collected data for *post hoc* orientation estimation; timing and coordinating sampling across components; and providing and regulating power for untethered operation. The inertial data logger design is conceptually split into four distinct subsystems that accomplish

each of these tasks: the inertial sensor array, flash memory unit, microcontroller, and power system.

Physically, the logger is split into a main board and multiple external sensor boards that can be wired to the main board as needed. The main board provides all the components to log data from a single rigid body, provide basic I/O, and interface with a computer. As a result, the main board is relatively large, with a footprint of 21 × 23mm and a mass of 1.3g without a battery. It is intended to be mounted on and take measurements from the torso of the bird. The schematic for the main board is shown in Figure 2.1 and the two-sided board layout is shown in Figure 2.2. Schematics and board layouts were made in Eagle (Autodesk Inc).

The external sensor boards consist only of a sensor and two capacitors for power filtering (see Section 2.2.4 for details). The main circuit supplies power and control signals to the external sensor boards, which are used to take inertial measurements from the relevant segments of the wing. In addition to providing flexibility in the number of sensors to log from and the wire lengths between sensors, this modular design also allows for minimizing the mass that needs to be mounted on the wing. Each external sensor board is 0.1g on a 8 × 9mm footprint. The schematic for the external sensor board is shown in Figure 2.3 and the two-sided board layout is shown in Figure 2.4. The finalized designs of both boards were printed and assembled in a small-batch production run by Seeed Fusion. Including parts, assembly, and shipping, the main boards work out to $60 CAD per unit for 5 boards, and the external sensor boards work out to just under $13 CAD per unit for 15 boards.

The data loggers are capable of reliably logging data at 1.8kHz, but this data rate must be split across the number of sensors that are collecting data. This works out to a maximum sampling rate of 900Hz, 600Hz, or 450Hz, when logging from 2, 3, or 4 sensors, respectively, which is comparable to many high-speed video systems.

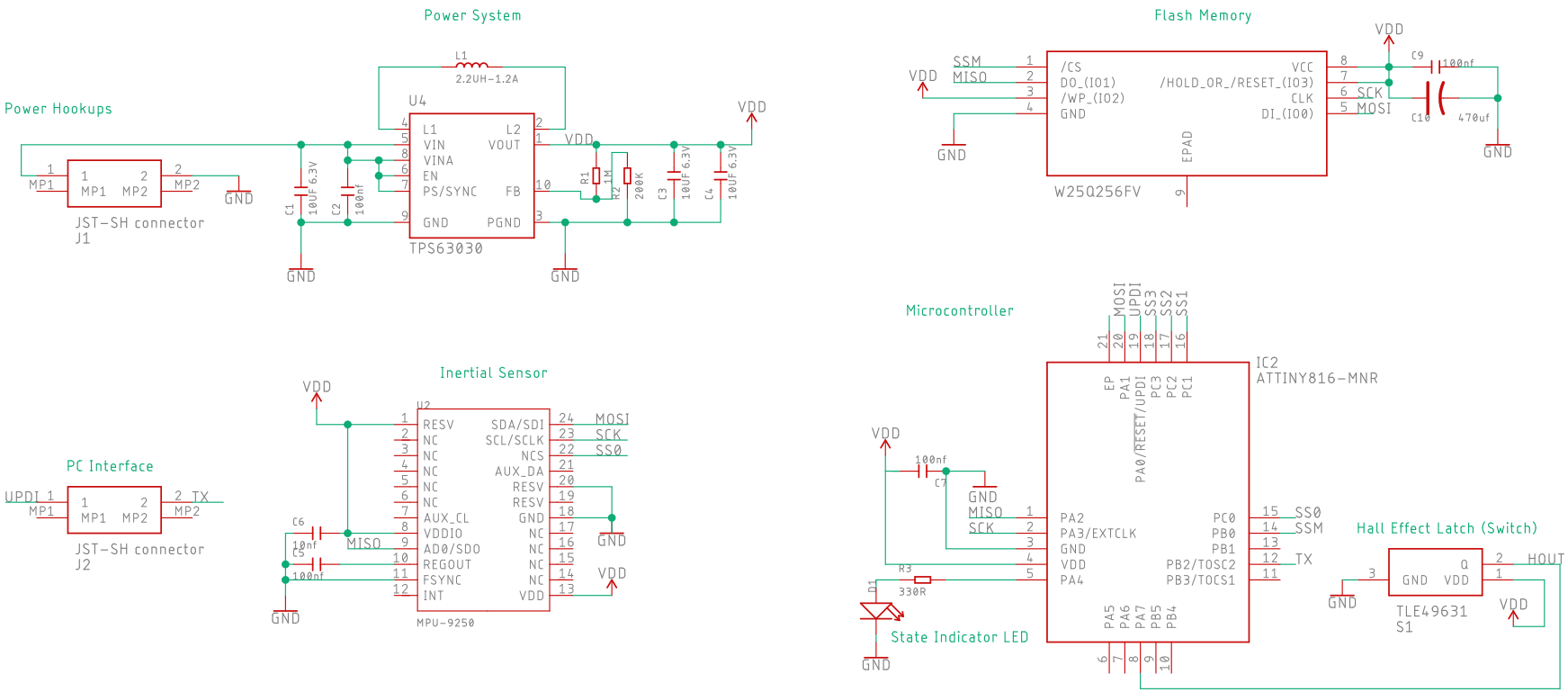

Figure 2.1: Schematic of the main board of the inertial data logger.

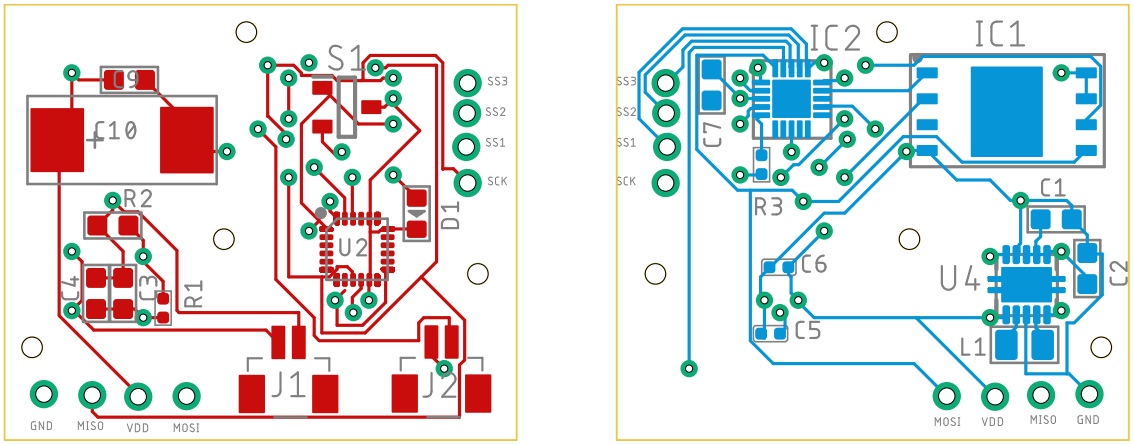

Figure 2.2: Two-sided board layout of the main board of the inertial data logger. The top layer is shown on the left and the bottom layer is shown on the right. Copper traces on the top surface are shown in red, those on the bottom are shown in blue, and copper that extends through the middle layer is shown in teal. Yellow lines show board boundaries. A 1mm scale is shown in black.

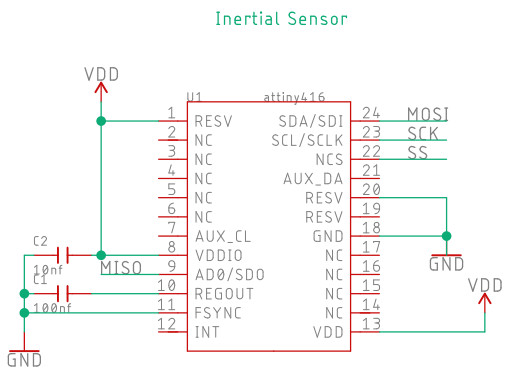

Figure 2.3: Schematic of the external sensor board of the inertial data logger.

2.2.1 Inertial sensor array

The inertial sensor array is a collection of magnetic, angular rate, and gravity (MARG) sensors that are all connected to a common set of communication lines. While local magnetism is not an inertial measurement, MARG sensors are used over sensors that only measure angular velocity and acceleration as acceleration alone is not enough to correct for drift in gyroscope measurements along all axes. See Section 2.4 for more details.

I chose to use the MPU-9250 MARG sensors from Invensense in the sensor array

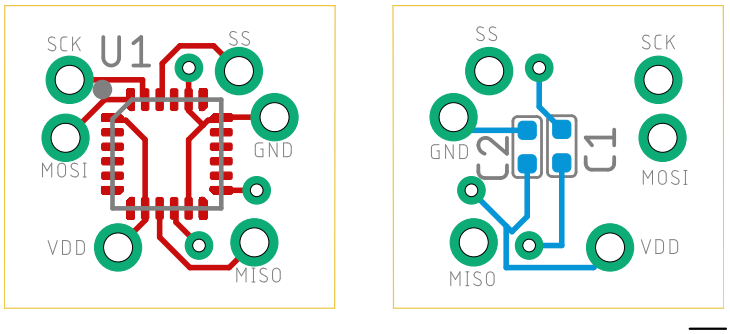

Figure 2.4: Two-sided board layout of the external sensor board of the inertial data logger. The top layer is shown on the left and the bottom layer is shown on the right. Copper traces on the top surface are shown in red, those on the bottom are shown in blue, and copper that extends through the middle layer is shown in teal. Yellow lines show board boundaries. A 1mm scale is shown in black.

as the other prominent MARG sensors on the market did not have a physical gyroscope (mCube MC6470), had lower maximum communication speeds (STMicroelectronics LSM9DS1TR), had irregular packages that made it difficult to test during the design phase (Bosch BMX055), or were simply too large for this application (Xsens MTi-1). Another option would be to spread the gyroscope, accelerometer and magnetometer functionality of each sensor across two or more chips. Although this greatly increases the options available to include specialized sensors with significantly better performance, I chose to use an all-in-one chip to minimize the hardware size and limit the number of communication transactions needed per sample.

It should be noted that the MPU-9250 chips are reaching the end of their production cycle and Invensense has released the ICM-20948 chip as its replacement. However, as these sensors are relatively new and not widely adopted, they were not stocked by the circuit assembly service I used (Seeed Fusion). Modifying the design to use the newer sensors would accordingly have added a considerable lead time during production. Future developments should consider switching to the newer chips, especially for its tolerance of lower running voltages, which would improve power efficiency.

I used the Serial Peripheral Interface (SPI) protocol to communicate with the

sensors in the array. Although this does require an extra wire for every additional sensor in the array, unlike the Inter-Integrated Circuit (I^2C) protocol, it is significantly faster than I^2C and does not have issues addressing multiple identical chips on the same set of communication lines.

2.2.2 Flash memory unit

As power loss is not always avoidable for systems running on battery power, a non-volatile memory system was needed to store the data collected from the sensors even in the event of power loss. The system also needs to be electronically addressed, as physically addressed systems are too large for this application. Of the systems meeting these requirements, Erasable Programmable Read Only Memory (EPROM) and Ferroelectric Random Access Memory (F-RAM) are not available with sufficient capacity, which only leaves flash memory.

I chose to use the W25Q256FV, a 256Mb NOR flash memory chip from Winbond, for this purpose. These chips are significantly smaller than a microSD card and slot, and data can be written directly to memory without a file system layer to improve write speeds and space efficiency. However, as the memory system is not removable, the data logger must be unmounted from the bird and connected to a computer to offload data between trials.

The 256Mb chips can store 1.8 million data samples from the MPU-9250 sensors after accounting for the space lost to block segmentation. For an array of 4 sensors sampling at 400Hz, this corresponds to just under 20 minutes of data collection. Recently released NAND flash memory chips from Winbond provide a drop-in replacement that would increase the storage capacity of the current data loggers four fold.

2.2.3 Microcontroller

I chose to use the ATtiny816 AVR microcontroller from Microchip Technology to control the data logger. While there are many viable options for small, low-power microcontrollers, the AVR architecture has the most community support and tools for prototyping. The ATtiny 1-series are the AVR chips available in the smallest footprints and with the lowest power consumption. Despite this, they can run at reasonably high clock speeds (20MHz) and have sufficient I/O for this design.

I developed the firmware for the data loggers in C, which splits operation into three modes: a stand-by mode that is entered when the loggers are connected to power; a data collection mode that polls the sensor array for data at regular intervals and writes the data to memory; and a data output mode that reads the data from memory and outputs it over a Transistor-Transistor Logic (TTL) interface. A single latching hall effect switch is used to choose between data collection and data output modes, and disconnecting power is used to end data collection or output. A single LED is similarly used to indicate the current status and prompt actions during body axes alignment as described in Section 2.3.

Interfacing with computers is provided by a two-pin JST SH port, which was chosen for its small size. One pin connects to the reset pin of the microcontroller and allows for modification of the microcontroller firmware and fuses using the Unified Program and Debug Interface (UPDI). The other pin is the transmission line of a TTL interface, which is used to output data. This pin can be directly tied to the receiving pin of a USB interface and interpreted by a computer, given a common ground. During data output, the microcontroller sends the Unicode representation of each byte in hexadecimal, separated by the Unicode character for a new line at the end of every data sample, or 18 bytes. The computer then just needs to read in the stream and write it directly to a file to save the data from a given trial. An R script is then used to convert the raw data into natural units and format the data

into an appropriately labelled CSV file. The script currently requires the user to manually indicate the number of sensors in the array and the sampling frequency.

2.2.4 Power system

The components used in the rest of the design can all run at 2.5 to 3.3V. However as small batteries tend to have considerable fall off in output voltage over a discharge cycle, a more reliable approach is to use a battery with a slightly higher nominal voltage with a voltage regulator. Although linear regulators require fewer accessory components, I chose to use a switching regulator for its considerable increase in efficiency. Buck-boost switching regulators also have the added benefit of being able to boost voltage back up if the battery dips below the target voltage. I chose to use the TPS63030 from Texas Instruments primarily because it was stocked on-site by Seeed Fusion, but any low-power switching regulator should produce comparable results. Future designs that opt to use the newer sensors from Invensense would also have the option of running the entire system at 1.8V, which should further improve power efficiency.

For the power source, I chose to use the 70mAh, 3.7V lithium polymer batteries sold by TinyCircuits for testing. Although these batteries are quite large at 1.9g, they are cheap, easy to recharge, and can power the device for 3.5 hours between charges, which is useful for testing. For actual use, smaller batteries are available that could greatly decrease the size of the complete data logger. The 10mAh, 3.1V MS920SE lithium batteries from Seiko Instruments for example, has a mass of only 0.5g, but require additional paperwork to be imported into Canada due to fire hazards during transport.

Finally, a series of passive components are placed around the circuit for power conditioning. Two 10µF capacitors are placed in parallel across the output and ground of the switching regulator as recommended by Texas Instruments and are

used to filter the ripple currents produced by switching and system-wide low-frequency noise. Each active component also has a 100nF bypass capacitor placed across their power and ground pins to filter out high-frequency noise. Somewhat surprisingly, reading and writing to flash memory accounts for roughly 80% of the expected peak current draw during operation. I chose to place a large 470µF capacitor near the flash memory to unambiguously isolate these loads from the rest of the power supply, as is commonly done with motors and other components with large current draws. However, future designs could likely scale this back safely as these current draw peaks are on the order of 20mA, which in perspective are not all that large.

2.2.5 Mounting

The inertial sensors must be rigidly mounted to the body sections they are tracking to accurately estimate orientation and infer kinematics further down the line. To mount the the main data logger board to the torso of the birds, I used a custom designed elastic harness. I first tied the main board down to a piece of card stock using the mounting holes drilled during manufacturing (see Figure 2.2). The card stock is meant to sit against the back of the birds with relief cuts along the sides to avoid affecting range of motion of the wings. I wove two 1/4" flat elastic loops through holes on the fore and aft ends of the card stock to act as head and body loops for the harness. The loops are opened prior to mounting and are closed and fitted to the birds using small double D-ring fasteners. Finally, another elastic is used to connect the two harness loops along the ventral surface of the bird with another double D-ring fastener used for fitting. To remove the harness, the three fasteners are opened in reverse order and the harness can then be lifted off the back of the bird.

I mounted the external sensor boards to the ventral surface of the wings, where

the feathers could be moved out of the way to expose the skin covering the manus and long bones of the wing. A single drop of cyanoacrylate was used to mount the boards and they were removed with gentle scraping after each trial. In the few cases where the cyanoacrylate made contact with nearby down feathers, the feathers were either plucked or trimmed to minimize stress for the birds. The wires connecting the main board to external sensor boards were braided together and sized to allow complete natural range of motion of the wings without risking entanglement.

2.3 Body axes alignment

The raw data from the inertial data loggers represent the motion of the sensor relative to an inertial frame of reference (often called the world frame), measured along the internal sensing axes of the chip, or the sensor frame. This is problematic as even slight variation in the way the sensor array is mounted will change the relationship between the individual sensor frames and the rigid bodies we are trying to track, or the body frame. Further, there is no way of relating the orientations estimated from one sensor to another without information about the initial arrangement of the sensors and rigid bodies.

To overcome these two issues while minimizing time spent handling the birds, I chose to adapt the two-pose body axes alignment used in some human inertial motion capture studies [60, 66]. After mounting the sensors, I held the wings closed against the body and collected 100 samples from the sensor array as the bird was held in each of two poses. I held the back of the bird upright in the first pose to find the anteroposterior axis in each of the individual sensor frames using the direction of gravity, as measured by the accelerometers. I then held the bird level to find the dorsoventral axis in a similar manner, again for each sensor. The mediolateral axis expressed in the sensor frame was then be found by taking the cross product

of these two vectors. I then redefined the dorsoventral axis using the cross product of the mediolateral and anteroposterior axes to ensure that all three are mutually perpendicular. After normalization, these three vectors form a basis for the body frame in the sensor frame for each sensor in the array. I used these bases to rotate all of the raw data from the individual sensor frames to their corresponding body frames prior to estimating orientation. This should correct for any variation across trials in how the sensors are mounted. By holding the wing in a known posture during the alignment, we also know the initial arrangement of the rigid bodies. This is important for measuring joint angles after the orientation of individual sensors have been estimated.

2.4 Orientation estimation

Orientation estimation using inertial measurements primarily relies on angular rate data from gyroscopes. As orientation is analogous to angular position, a first-order approximation would be simply to integrate the angular velocity of the rigid body over time. Given the initial orientation, perfect gyroscope measurements, and an infinite sampling rate, the orientation of a rigid body can be calculated exactly by this method. In practice, measurement error and information lost to discretization in time leads to the accumulation of error in orientation estimates over time.

To correct for this accumulating drift, we need a way of assessing our orientation estimate at a given time without reference to previous estimates. An intuitive option is to use the relative orientation of locally-fixed reference vectors, like gravity and the Earth's magnetic field, to assess a given estimate of the orientation of the rigid body. Put another way, if gravity is pulling you up, there is a good chance you are upside down.

The challenge then is to make the most efficient use of these two distinct methods

of inferring orientation to produce a single best estimate at every time point. A useful observation is that this system is well approximated by a Markov chain with continuous states. The state (i.e., orientation) at any point in time depends only on the previous state and the angular velocity during the transition between states. Further, we can make observations (i.e., orientation of reference vectors) that depend only on the current state. This congruence allows us to apply a recursive Bayesian approach to estimating the orientation using the inertial sensor data. See Section 1.1 for an introduction to Bayesian inference with Markov models.

To facilitate calculations on orientation, I use the unit quaternion representation of orientations throughout this project. This representation does not suffer from gimbal lock, unlike Euler angles [45], and is slightly more computationally efficient and numerically stable in recursion than rotation matrices [67]. Euler angles are however used in visualizations as they are more intuitive to most readers. Please see Section 1.2 for a brief introduction to working with unit quaternions as orientations.

2.4.1 Algorithm choice

I chose to use a Monte Carlo smoother with parameter estimation [37, 38, 65] to estimate orientation from the inertial data. A Monte Carlo approach is used over more compact state descriptions that depend on model linearity (e.g., Kalman filters [27]) near-linearity (e.g., extended Kalman filters [30]), or Gaussian-distributed probabilities (e.g., unscented Kalman filters [68]) as the observation incorporation is strongly non-linear. It should be noted however that the Bingham and matrix von Mises-Fisher distributions provide analogs to multivariate normal distributions in n-dimensional spheres. These can be used to adapt non-Monte Carlo recursive Bayesian algorithms to linear models of orientation using unit quaternions, as this space is a 4-dimensional unit sphere. The rationale for using a non-linear model here is explored in more detail below, in Section 2.4.2.

I chose to use a smoother in favour of a filter as estimation accuracy is more important than computational efficiency for this application. Please note that in the context of recursive Bayesian inference, a filter is an algorithm that estimates the probability density distribution of a time-varying state, x_t, given all observations up to that time point, $y_{1:t}$. In contrast, a smoother is an algorithm that does the same using all available observations, including those take after time t. See Section 1.1.3 for more information.

This is especially important here as the measurement uncertainty of the observations will not be constant over time. During periods with relatively unreliable observations, filtering algorithms can only make use of reliable observations from the past to minimize error in the state estimate. Smoothers on the other hand can make use of information from both the past and future to inform estimates of the state during these periods. This of course requires the estimation to be done after-the-fact, which limits the utility of smoothers in many other applications. Recursively incorporating both past and future information into the state estimate at every time point also adds considerable computational complexity. In the case of Monte Carlo smoothers and filters, Yang *et al.* [38] found that the computation time grew quadratically with the length of the input for smoothing as opposed to linearly for filtering.

I also chose to use an algorithm that incorporates parameter estimation [37], as a number of the model parameters could not feasibly be characterized for every sensor in every inertial logger.

2.4.2 Model definition

As described in Section 1.1, recursive Bayesian inference requires modelling two processes: state transitions and the relationship between observations and the state. Smoothing also requires an inverse state transition function, but this is generally

straight forward to construct from the forward transition function. These processes are necessarily stochastic and are characterized by measurement and transition errors that in turn must be modelled.

Modelling error

I chose to model the measurement error for all three types of sensors as a linear transformation plus Gaussian noise. Using the accelerometer data as an example, we get:

$$a = (\tilde{a} + \tau_a)\, s_a + \nu_a \qquad (\nu_a \sim \mathcal{N}_3(0, \sigma_a I_3)) \tag{2.1}$$

where a is the true acceleration experienced by the rigid body, $\tilde{a}$ is the raw measurement, τ_a is the systemic bias or trim along each axis, s_a is the sensitivity of the sensor along each axis, $\mathcal{N}_3(\alpha, \beta)$ is the normal distribution function in $\mathbb{R}^3$ with mean α and variance-covariance β, σ_a is the variance of the measurement noise, and I_3 is the identity matrix in $\mathbb{R}^3$. I chose to model the Gaussian noise as independent and with equal variation along each axis primarily to limit the complexity of the parameters that would be estimated by the algorithm.

The trim vector for the gyroscope can easily be estimated and corrected for by averaging many samples taken when the sensor is stationary. This allows us to simplify this model as

$$\tilde{g} = g s_g + \nu_g \qquad (\nu_g \sim \mathcal{N}_3(0, \sigma_g I_3))$$

The transition error on the other hand depends only on the random variation in the angular rate measurements as well as error due to discretization in time. Because I am modelling both of these processes as normally distributed with a mean of zero and because the gyroscope noise will only ever be used when transitioning states,

I chose to consolidate their respective parameters into a single parameter σ_T that characterizes the overall variation of the transition error, modelled as a multivariate normal distribution.

This gives us 7 model parameters with a total dimensionality of 18.

$$\phi = \begin{bmatrix} \tau_a & \tau_m & s_a & s_m & s_g & \sigma_a & \sigma_m & \sigma_T \end{bmatrix}^T$$

State transition

The state transition function in a Monte Carlo Markov chain should map each particle in the state distribution at a given time step to a new particle at the next time step. As a particle in this model is a point estimate of orientation, any transition between particles can be expressed as a single rotation. Using the unit quaternion representation of orientations and (1.7), we can write this transition as

$$\tfrac{B}{W}\hat{q}_{t+1}^{(i)} = \tfrac{\Delta B}{W}\hat{q}_t^{(i)} \times \tfrac{B}{W}\hat{q}_t^{(i)} \tag{2.2}$$

where $\tfrac{B}{W}\hat{q}$ is the orientation of the body (B) in with world frame (W) represented as a unit quaternion, $\tfrac{\Delta B}{W}\hat{q}$ is the change in this orientation over a time step, and i is the particle index. See Section 1.2.4 for a primer on rotating orientations.

We can then use the axis-angle representation (1.5) to decompose the change in orientation over the time step.

$$\begin{aligned} \tfrac{B}{W}\hat{q}_{t+1}^{(i)} &= \begin{bmatrix} cos\left(\frac{\theta_t}{2}\right) \\ {}_W\hat{n}_t \; sin\left(\frac{\theta_t}{2}\right) \end{bmatrix} \times \tfrac{B}{W}\hat{q}_t^{(i)} \\ &= \begin{bmatrix} cos\left(\frac{\dot{\theta}_t}{2}\Delta t\right) \\ {}_W\hat{n}_t \; sin\left(\frac{\dot{\theta}_t}{2}\Delta t\right) \end{bmatrix} \times \tfrac{B}{W}\hat{q}_t^{(i)} \end{aligned}$$

where θ is the magnitude of the rotation, $\dot{\theta}$ is the scalar angular rate, and ${}_W\hat{n}$ is the axis of the rotation in the world frame.

The scalar angular rate is the magnitude of the corrected gyroscope reading.

$$\begin{aligned}\dot{\theta}_t &= \|g_t\| \\ &= \|\tilde{g}_t s_g\|\end{aligned}$$

The axis of the rotation can similarly be calculated by normalization.

$$\begin{aligned}{}_B\hat{n} &= \hat{g}_t \\ &= \frac{g_t}{\|g_t\|} \\ &= \frac{\tilde{g}_t s_g}{\|\tilde{g}_t s_g\|}\end{aligned}$$

In practise, this also requires a check to ensure that the magnitude of the corrected gyroscope reading is not zero. As the gyroscope data is collected in the body frame, the axis of rotation calculated from it must be rotated into the world frame using the current orientation of the body and (1.8).

$$\begin{bmatrix} 0 \\ {}_W\hat{n} \end{bmatrix} = {}^B_W\hat{q}_t^{(i)} \times \begin{bmatrix} 0 \\ {}_B\hat{n} \end{bmatrix} \times \left({}^B_W\hat{q}_t^{(i)}\right)^{-1}$$

where $\hat{q}^{-1}$ is the inverse of the unit quaternion $\hat{q}$. Note that the procedure for applying quaternion rotations to vectors in $\mathbb{R}^3$ differs from the procedure for applying quaternion rotations to other quaternions. See Section 1.2.4 for more details.

Substituting these back into (2.2) gives us the deterministic transition function.

$$\overline{{}^{B}_{W}\hat{q}^{(i)}_{t+1}} = \left(\begin{bmatrix} cos\left(\frac{\|\tilde{g}_t s_g\|}{2}\Delta t\right) \\ 0_3 \end{bmatrix} + {}^{B}_{W}\hat{q}^{(i)}_{t} \times \begin{bmatrix} 0 \\ \frac{\tilde{g}_t s_g}{\|\tilde{g}_t s_g\|} \end{bmatrix} \times \left({}^{B}_{W}\hat{q}^{(i)}_{t}\right)^{-1} \right) \times {}^{B}_{W}\hat{q}^{(i)}_{t}$$

where 0_3 is the zero vector in $\mathbb{R}^3$. I then sample from a normal distribution centered at the deterministic solution to incorporate transition error.

$${}^{B}_{W}q^{(i)}_{t+1} \sim \mathcal{N}_4\left(\overline{{}^{B}_{W}\hat{q}^{(i)}_{t+1}},\ \sigma_T I_4\right)$$

The resulting quaternion must finally be normalized to ensure that it is a unit quaternion.

Observation incorporation

Observation incorporation is accomplished by finding the joint probability of the state given all past information (i.e., the results of the previous state transition) and the state given the current observation.

$$\begin{aligned} p(x_t \mid y_{1:t}) &= p(x_t \mid y_{1:t-1}, y_t) \\ &= p(x_t \mid x_{t-1}, y_t) \qquad (t = 1, \ldots, T) \end{aligned} \tag{2.3}$$

using the assumptions of a Markov model applied recursively.

Joint probabilities of distributions approximated by sets of particles are calculated using weighted bootstrapping. The challenge here lies in appropriately assigning weights to the particles given the observations from the corresponding time point. The general procedure for this weighting assignment using the measurement of reference vectors is given below, followed by considerations specific to the physical constraints of accelerometers and magnetometers individually.

Orienting with reference vectors — The accelerometer and magnetometer data both provide indirect measurements of orientation through the direct measurement of vectors that are locally uniform and static in the world frame, i.e., gravity and the Earth's magnetic field, respectively. The key idea is that a rotation that describes the transformation of a sensor from one orientation to another in the world frame also describes the inverse of this transformation for the reference vector as measured by the sensor. Put another way, turning yourself (sensor) 90°clockwise is equivalent to turning the room (reference vector) 90°counterclockwise, from your (sensors) perspective.

Orientation with reference vectors in this way requires a measurement of the reference in both the current and initial orientation. Data samples collected in the second posture of body axes alignment are averaged and normalized after correction for scale and trim to estimate ${}_B\hat{a}_0$ and ${}_B\hat{m}_0$, the gravity and magnetic field vectors measured in the initial orientation. See Section 2.3 for details on the body axes alignment procedure.

Observations of the reference vectors are not used to directly estimate orientation as any valid observation could be produced by an infinite set of orientations. This set is spanned by rotating any one valid orientation around the reference vector. Equivalently, you can tell that you are upright by the direction of gravity, but not which cardinal direction you are facing. The inverse of this transformation, however, is useful as the expected measurement of the reference vector is unique for a given orientation and initial measurement. If you believe that you are upright and facing west, you expect gravity to pull you down.

The generalized particle weight for a sensor modality y with measurement error modelled as in (2.1) can then be modelled as

$$
\begin{aligned}
w_{t,y}^{(i)} &= \mathcal{N}_3\left({}_B\bar{y}_t^{(i)} \mid {}_B\hat{y}_t,\ \sigma_y \rho_{y,t} I_3\right) \\
&= \mathcal{N}_3\left({}_B\bar{y}_t^{(i)} \middle| \frac{\left({}_B\tilde{y}_t - \tau_y\right) s_y}{\left\|\left({}_B\tilde{y}_t - \tau_y\right) s_y\right\|},\ \sigma_y \rho_{y,t} I_3\right)
\end{aligned}
$$

where $w^{(i)}$ is the weight of the particle i, $\mathcal{N}(\gamma \mid \alpha, \beta)$ is the density of the multivariate normal distribution $\mathcal{N}(\alpha, \beta)$ evaluated at γ, $\bar{y}^{(i)}$ is the expected normalized sensor reading given particle i, $\hat{y}$ is the actual corrected normalized sensor reading, $\tilde{y}$ is the sensor reading prior to correction, and $\rho_{y,t}$ is a time-dependent correction for the measurement noise, σ_y, that depends on the physical properties of the sensor. The expected sensor reading here is calculated by rotation of the initial reading, as described above.

$$
\begin{bmatrix} 0 \\ {}_B\bar{y}_t^{(i)} \end{bmatrix} = \left({}_W^B\hat{q}_t^{(i)}\right)^{-1} \times \begin{bmatrix} 0 \\ {}_B\hat{y}_0 \end{bmatrix} \times {}_W^B\hat{q}_t^{(i)}
$$

Accelerometer correction — Accelerometer data is challenging to use in isolation as gravity and the motion of the body in the world frame can not reliably be distinguished. I approximate this motion-dependent increase in measurement uncertainty as being proportional to the relative change in the magnitude of the accelerometer reading from 1G. That is, for every n-fold increase or decrease in the magnitude of the measured acceleration from the stationary case, I scale the accelerometer measurement error by n.

$$
\rho_{a,t} = \left|\ln\left(\left\|{}_Ba_t\right\|\right)\right| + 1
$$

This correction evaluates to 1 when the magnitude of the accelerometer reading is 1G and grows without bound as the magnitude of the accelerometer approaches 0 or ∞.

The correction notably does not account for non-zero linear accelerations that sum with gravity to produce readings with magnitude 1G, such as an acceleration of 2G downwards. As these situations are expected to be transient and observation incorporation works by recursion over time, the effect of these cases is expected to be minimal.

Magnetometer correction — The primary concern with magnetometer readings are that they update slowly relative to gyroscopes and accelerometers. While this is likely not a concern for angular rates on the scale of human limb kinematics, bird wing kinematics are considerably faster. I chose to approximate the rate-dependent increase in measurement error as

$$\rho_{m,t} = \mathrm{e}^{\|{}_B g_t\|}$$

This correction evaluates to 1 when there is no angular motion and increases without bound as the magnitude of the angular velocity tends to ∞.

The weights calculated for each particle given each sensor modality is then combined by summation to produce the final weight for each particle. These weights are then normalized and used to resample the prior distribution of the state produced by the previous state transition to produce the posterior filtering distribution for the state.

Inverse transition function

Many smoothing algorithms, including the one used here [65], use a forwards filtering, backwards resampling approach. In these algorithms, the filtered distribution is first calculated recursively forwards in time before iterating backwards over time to calculate the joint probability of the state given all observations. Using the equation for the filtered distribution (2.3), this can be expressed as

$$
\begin{aligned}
p(x_t \mid y_{1:T}) &= p(x_t \mid y_{1:t}, y_{t+1:T}) \\
&= p(x_t \mid \tilde{x}_t, x_{t+1}) \qquad (t = 1, \ldots, T-1)
\end{aligned}
$$

where $\tilde{x}_t$ is the filtered distribution $p(x_t \mid y_{1:t})$. Note that this backwards recursion is possible because the filtered and the smoothed distributions are the same when $t = T$.

To evaluate this joint probability with particle distributions we need to assign bootstrapping weights to the filtered particles given the smoothed state in the following time step. Using (2.2), I chose to model the smoothing weights for this inverse transition as

$$
\omega_t^{(i)} = \mathcal{N}_4 \left({}^{\Delta B}_{W}\hat{q}_t \times {}^{B}_{W}\tilde{q}_t^{(i)} \mid {}^{B}_{W}\hat{q}_{t+1},\ \sigma_T I_4 \right)
$$

where $\tilde{q}_t^{(i)}$ is a particle (unit quaternion) in the filtered distribution.

The final smoothed distribution at each time would normally be found by bootstrap resampling the filtered particles using these weights, following Godsill *et al* [65]. However, this procedure is modified as described in the refiltering smoother algorithm [38] to marginalize the smoothed distribution over the distribution of the model parameters estimated following Liu and West [37]. For individuals draws from the posterior distribution of the parameter set, a full set of particles is filtered forwards in time and a single smoothed trajectory is sampled backwards during the refiltering step. This set of trajectories is the final smoothed orientation estimate.

2.5 Validation

I ran a series of tests to validate the tools and procedures developed in this project. The tests were designed to incrementally assess the assumptions that must be made

to infer bird wing kinematics from inertial measurements collected by body-mounted inertial data loggers.

2.5.1 Orientation estimation of a single sensor

Motion capture using inertial sensors depends on reliably being able to estimate the orientation of sensors in the world frame. To assess the accuracy with which we can estimate orientation using these sensors, I recorded the orientation of a single sensor over time as I rotated the board by hand. I then compared the resulting orientation estimates to those from an optical motion capture system, OptiTrack (NaturalPoint Inc).

I placed retro-reflective markers onto the board of the inertial data logger to facilitate tracking by both motion capture systems and used an external trigger device (eSync2, NaturalPoint Inc) to synchronize the two systems at a 200Hz sampling rate. I had to modify the data logger code to use the external trigger rather than the internal clock for timing samples. No other changes were made to the sensors. To distinguish estimation error from error due to imprecise body axes alignment, the body frame for the inertial motion capture was taken to be co-incident with the sensor frame. The body frame axes in OptiTrack were similarly aligned to the edges of the circuit board. Although this does not account for slight errors in how the sensor was mounted to the board or how the board was cut, this was far more reproducible than attempting to align to the sensor itself due to its small size. Three cameras were used to record the markers in the optical system. 10,000 particles were simulated for the Monte Carlo smoother and 16 trajectories were sampled in the refiltering step.

During the recording, the sensor was manually rotated so as to maximize excursion without losing sight of the markers in the optical tracking. The accuracy of the inertial motion capture was assessed using the angular distance between the two

orientation estimates at each time point.

Euler angle traces of the orientation estimated by the two systems are shown in Figure 2.5A and the absolute angular distance between measures are shown in Figure 2.5B. The Euler angles shown here and elsewhere in the paper are Tait-Bryan angles, applied in the order yaw, pitch, roll, and are active, intrinsic rotations. The absolute error of the orientation estimation remains within 3° for the entire trace, with error increasing as the sensors departed from the initial orientation, but decreasing as the sensor returned. Slight misalignment between the two motion capture system body axes and gyroscope scaling errors could both account in part for this pattern.

2.5.2 Body axes alignment

It is unreasonable to assume that the inertial sensors can be mounted onto the birds in exactly the same orientation every time. Accordingly, we use body axes alignment as described in Section 2.3 to estimate orientation in biologically relevant axes irrespective of the orientation of the sensor relative to the body.

To validate this procedure, I compared the orientation estimates from two sensors mounted on a single rigid body with their sensing axes misaligned both from the body frame and from one another. If orientation is estimated directly without rotating the raw data to align to the body frame, the two estimates should diverge as the rigid body moves away from its initial orientation. This is because the sensors will view each incremental rotation as being made around different axes. However, if the body axes alignment procedure is successful, both sensors should be able to track the orientation of the rigid body as a whole, which should not depend on the mounting orientation of each individual sensor. We can assess the repeatability of the body axes alignment using the angular distance between these two orientation estimates at each time point. Further, to qualitatively ensure that the body axes inferred by the alignment procedure coincide with those of the actual rigid body,

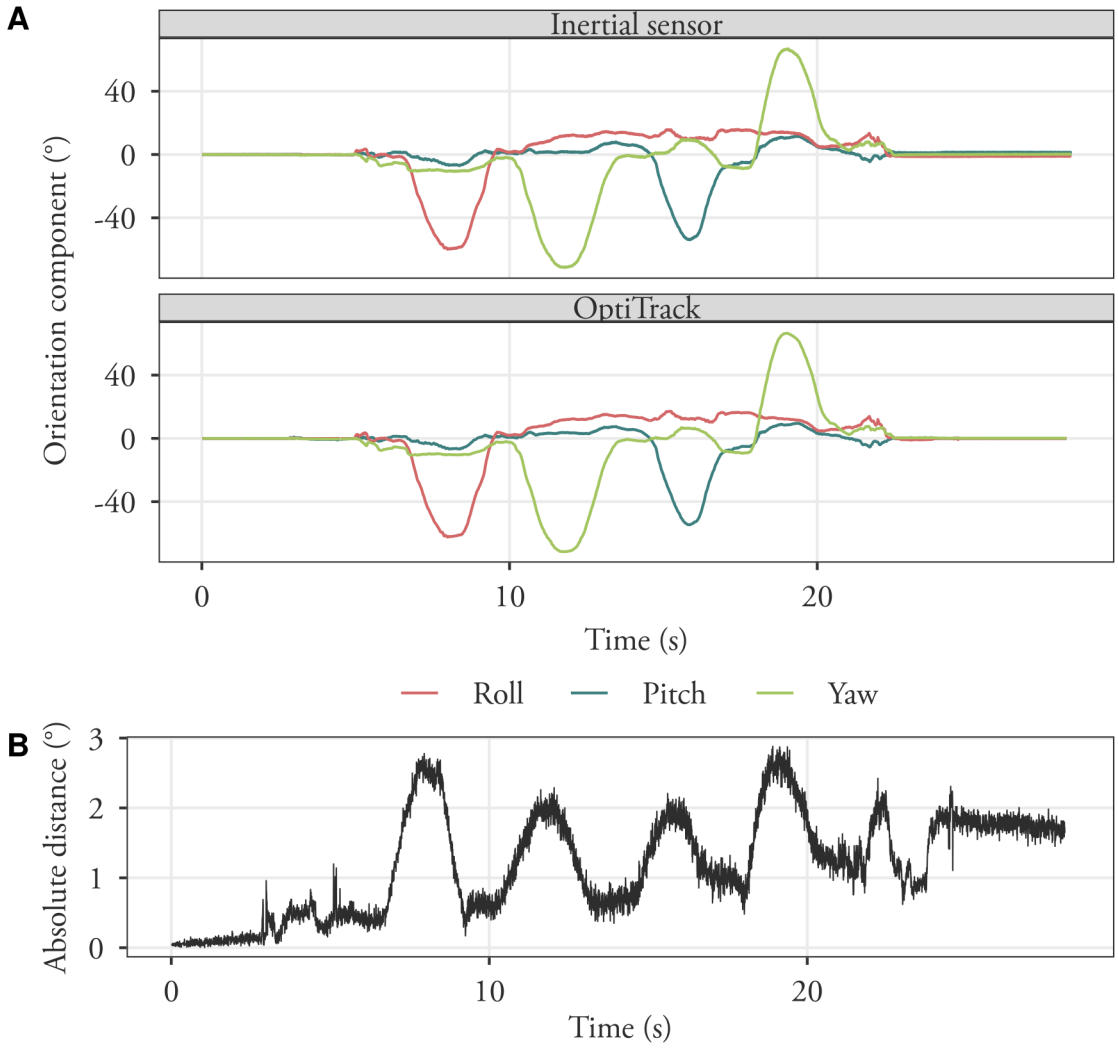

Figure 2.5: Comparison of orientation estimates of a single rigid body made using inertial sensors and an optical motion tracking system, OptiTrack. (A) Traces of Euler angles estimated using the two systems. (B) Absolute angular distance between the two orientations estimates.

I rotated the rigid body 90° around each of the true body axes in turn during the recording. Specifically, I rotated the rigid body around the z, y, and x axes, in that order, which correspond to rotations in yaw, pitch, and roll, respectively. Data were collected at 800Hz, but were down sampled to 400Hz prior to estimation to save processing time. As before, 10,000 particles were used in the Monte Carlo smoother and 16 trajectories were sampled in the refiltering.

Euler angle traces of the orientations estimated by the two sensors without body axes alignment are shown in Figure 2.6A. As expected, the two estimates diverge except when in the initial orientation. Further, the three rotations did not align with the body frame as expected given the recorded motion for either sensor. Coincidentally, the first sensor did read the second rotation as a pure rotation around z, but this rotation should be a 90° rotation around around y in the body frame.

In contrast, the Euler angle traces estimated by the two sensors with body alignment (Figure 2.6B) show considerably more agreement. The key exception is during the second rotation where the two sensors appear to disagree by over 90° along two axes. However, this rotation should be a 90° rotation in pitch, which results in gimbal lock in the Euler angle system. Accordingly, slight changes in orientation around this point can result in very large and even discontinuous changes in Euler angles. Here, both sensors estimate a pitch of approximately -90°, but the first sensor also estimates a roll and yaw of approximately ±180° while the second sensor estimates a roll and yaw of approximately -90° and 90°, respectively. Figure 2.7 demonstrates how these two Euler angle triplets are equivalent to a pure pitch of -90°. Accordingly, the two sensors show strong agreement after alignment and both show rotations around the expected body axes in the expected order.

Finally, the absolute distance between the two estimates using body axes alignment are shown in Figure 2.6C. Error between the estimates remains within 6° for the entire trace, with error increasing as the body departs from its initial orienta-

tion. This is consistent with the previous results with a single sensor as error is compounded when comparing two estimated values. In practise, this means that the orientation at which a sensor is mounted onto a given body segment can be corrected for within 6° with the body axes alignment.

2.5.3 Joint angle measurement

The tests up to this point establish whether or not we can reliably track the orientation of individual rigid sections of the body over time. However, to extract useful kinematic data, we also need to be able to reliably measure relative orientation between body sections. This should allow us to extract joint angles along biologically relevant axes.

To assess the accuracy with which we can measure joint angles, I built a simple hinge and compared joint angle estimates made over time by the inertial sensors and OptiTrack. The hinge consisted of a piece of foam core board which was scored to bend along a single axis. I placed an inertial sensor on either side of the hinge and aligned the axis of rotation to the x axis of the body frame using body axes alignment. Retro-reflective markers were also placed on either side of the hinge and an eSync2 external trigger was again used to synchronize samples in both motion capture systems at 200Hz. As before, the sensors were modified to use the external trigger. However, the data samples used by the inertial system to define body axes were driven by the internal clock and were not recorded in the optical tracking system as the markers were not visible in the calibration postures. As before, 10,000 particles were simulated in the Monte Carlo smoother and 16 trajectories were sampled during refiltering. In addition to manually flexing the hinge, I also rotated the entire assembly continuously through the entire recording to test whether or not the joint angle estimates could reliably reject the motion of the body.

To measure joint angles, I start by finding a quaternion that represents the

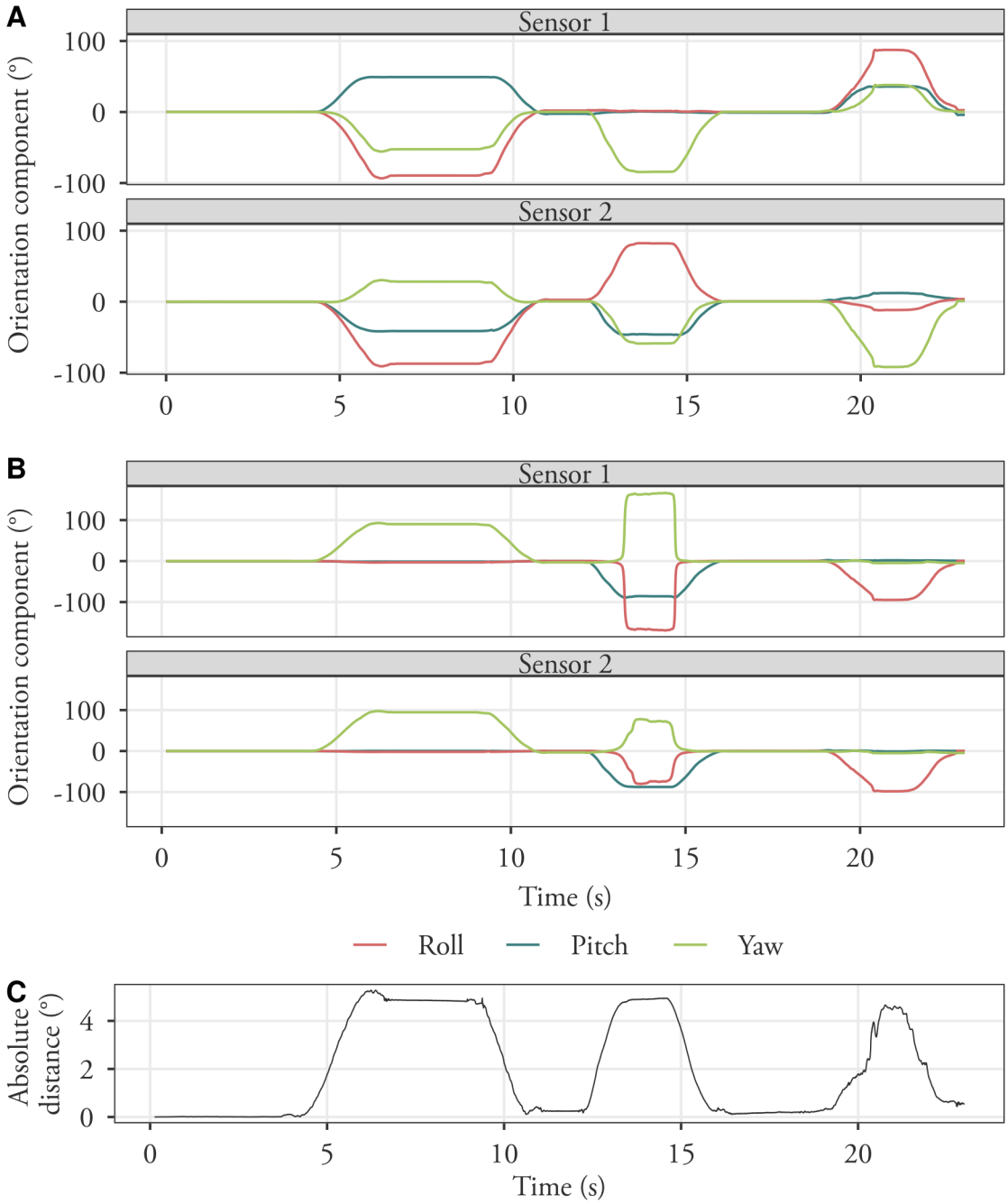

Figure 2.6: Comparison of orientation estimates of a single rigid body made by two misaligned sensors with and without body axes alignment. (A) Euler angles estimated from the two inertial sensors without axis alignment. (B) Euler angles estimated from the two inertial sensors with axis alignment as described in Section 2.3. (C) Absolute angular distance between the two orientations estimates with axis alignment.

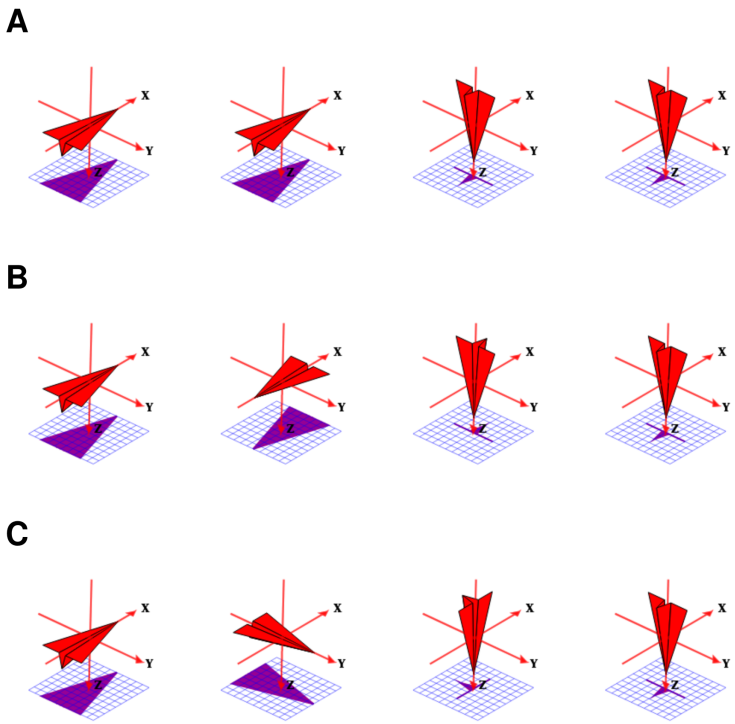

Figure 2.7: Demonstration of the Euler angle singularity point with three different angle triplets that encode the same orientation. Rows correspond to the three different angle triplets. The first column of each row shows a paper airplane in an unrotated orientation and each subsequent column applies another component of the Euler angle corresponding to its row. The three Euler angle triplets are: (A) Yaw 0°, Pitch -90°, Roll 0°; (B) Yaw 180°, Pitch -90°, Roll -180°; (C) Yaw 90°, Pitch -90°, Roll -90°. The Euler angles here and elsewhere in the paper are applied in the order yaw, pitch, roll, and are intrinsic, active rotations. The images in this figure were generated using the Euler angle visualization tool found at http://danceswithcode.net/engineeringnotes/rotations_in_3d/demo3D/rotations_in_3d_tool.html [69].

rotation from the first body segment to the second body segment for each sample in time. This rotation is equivalent to the orientation of the second body segment in the first body segment frame. As described in Section 1.2.4, this change of frame can be achieved by conjugation:

$$ {}^{B_n}_{W}\hat{q}^{-1} \times {}^{B_{n+1}}_{W}\hat{q} = {}^{B_{n+1}}_{B_n}\hat{q} $$

where ${}^{B_n}_{W}\hat{q}$ is the unit quaternion representing the orientation of the nth rigid body (B) in the world frame (W).

From this relative orientation, we can use the axis-angle representation of a quaternion (1.5) to solve for the magnitude of the rotation for every sample in time:

$$ {}^{B_{n+1}}_{B_n}\theta = 2\ cos^{-1}\left({}^{B_{n+1}}_{B_n}q_w\right) $$

Note that q_w here is the first, or real, component of a unit quaternion $\hat{q}$.

I then use the difference between the joint angles estimated by the two systems to assess the accuracy of the joint angle measures. Further, as the hinge angle was aligned to the x axis of the body frame, we can assess our ability to measure along biologically relevant axes using the angles measured across the hinge along the other two axes.

The joint angles measured over time by the two systems are shown in Figure 2.8A. Both systems show strong agreement, although the data collected by OptiTrack show more jitter as well as a few frames with missing data. This is likely due to accidental partial occlusion of markers by my hand as I rotated and flexed the hinge within the recording volume. Error between the two estimates are shown in Figure 2.8B and remain within 5° over the recording. The joint angles estimated by the inertial sensors also are not consistently over- or underestimated relative to the reference.

Figure 2.9A demonstrates how the motion of the body is rejected from the orien-

tation estimates by using another body segment as a reference frame. As expected, the estimated orientations show no clear patterns when plotted relative to the world frame as the flexion of the hinge cannot be distinguished from the motion of the entire assembly. However, by using the relative orientation of the two segments, it is clear that the rotation seen across the hinge is primarily around the x axis of the body, or in other words, roll. Figure 2.9B shows the angles measured across the hinge along the other two body axes, or pitch and yaw. As before, the error remains within 5° over the recording, which is consistent with the previous results.

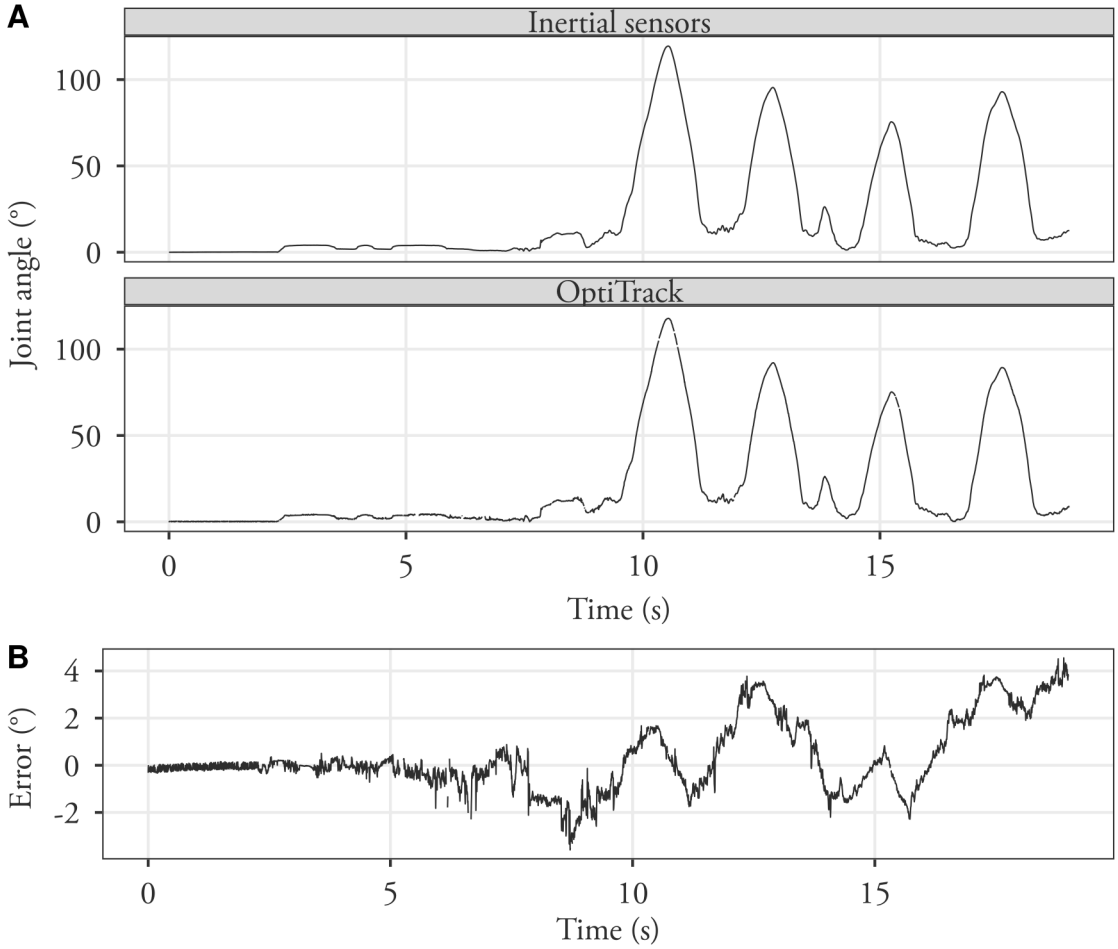

Figure 2.8: Comparison of joint angle estimates made from a simple hinge using inertial sensors and OptiTrack. (A) Angles estimated across the hinge by the two systems over time. (B) Difference between the two estimates over time.

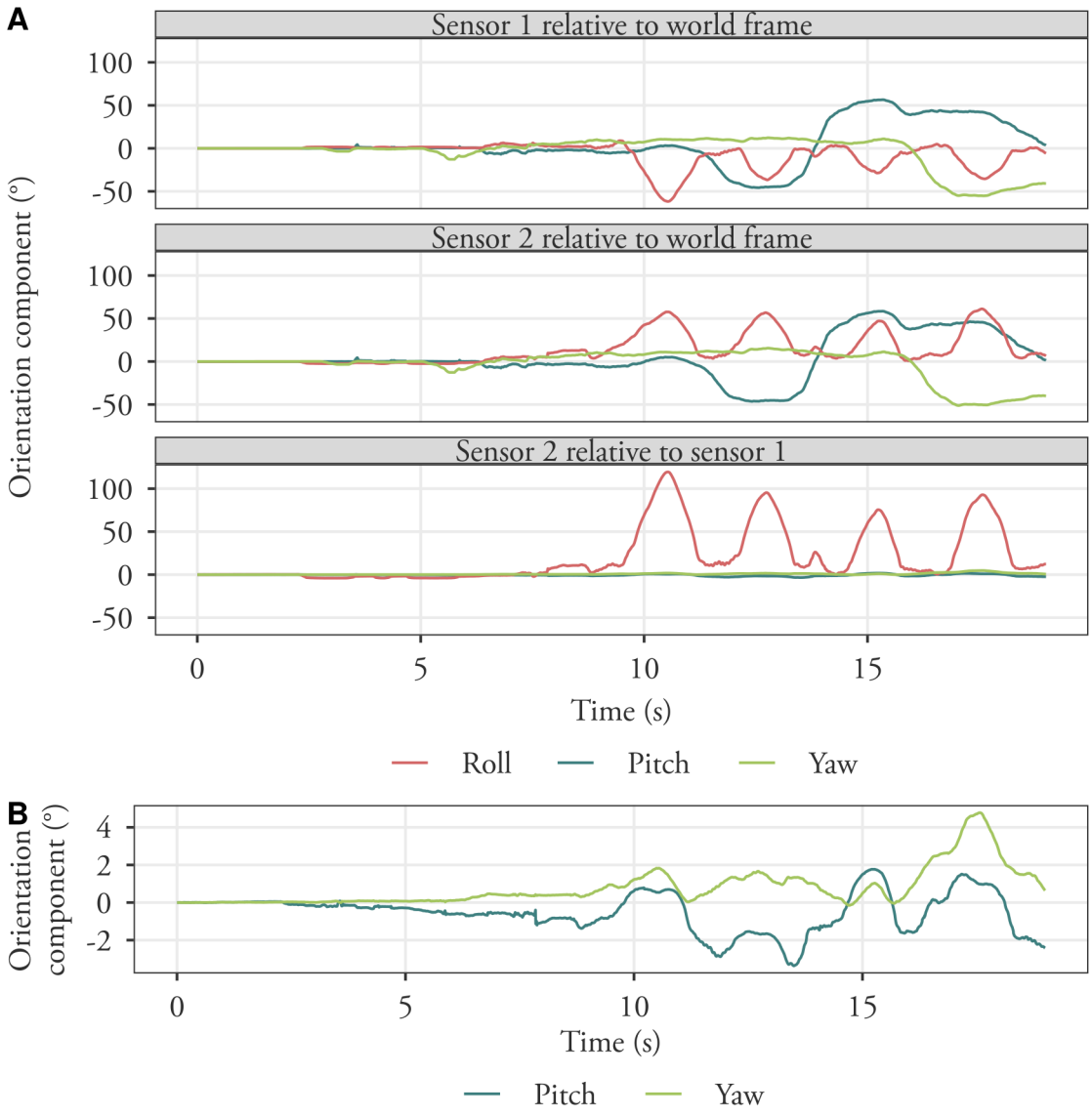

Figure 2.9: Demonstration of how the motion of the body can be rejected to record joint dynamics by changing frames of reference. (A) Orientations of two sensors mounted across a simple hinge plotted over time using different frames of reference. (B) Angles measured across the hinge along body axes perpendicular to the expected axis of rotation.

2.5.4 Estimating wing stroke parameters in a deceased bird

One common goal with tracking wing kinematics is estimating wing stroke parameters that might change over some experimental manipulation [11, 13]. In particular, stroke amplitude and the inclination of the stroke plane are of common interest[12, 70, 71]. To assess the accuracy with which these parameters can be estimated, I compared the estimates made over a range of simulated wing strokes using both the inertial tracking system and OptiTrack. The primary benefit of validating these measurements initially with simulated wing strokes in deceased birds is the relative ease with which the two motion capture systems can be aligned and synchronized.

I mounted sensors to the body and forearm of a deceased pigeon (*Columba livia*) provided by the Beaty Biodiversity Museum as described in Section 2.2.5. As before, I used retro-reflective markers to track the body and wings of the pigeon in OptiTrack and an external trigger to synchronize the two motion capture systems at 200Hz. The sensors were similarly modified to use the external trigger, except for during body axes alignment. 10,000 particles were simulated in the Monte Carlo smoother and 16 trajectories were sampled during refiltering.

To calculate stroke amplitude, I first calculate the relative orientation of the forearm relative to the body using the data from the two sensors as in (2.5.3). I then separate the trace into individual strokes using the wing elevation, which correspond to the local extrema in rotation around the longitudinal axis of the bird, or roll in the body frame. I then find the quaternion representing the shortest rotation between these two stroke end points for each stroke, again by conjugation.

$$ {}^{wing}_{body}\hat{q}_{stroke\ end} \times {}^{wing}_{body}\hat{q}^{-1}_{stroke\ start} = {}^{stroke\ end}_{stroke\ start}\hat{q} $$

Using the axis-angle representation (1.5) we can solve for the magnitude of this rotation as in (2.5.3). However, this magnitude includes average wing twist over

the stroke, which is generally not the goal with estimates of wing stroke amplitude. To account for this, I use (1.5) and (2.5.3) to solve for the axis of the rotation that describes the wing stroke.

$$_{stroke\ start}\hat{n} = \begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} / \ sin\left(\frac{\theta}{2}\right)$$

However, this is still in the frame of the wing at the start of the stroke. To find the axis of the wing stroke rotation in the body frame of the bird I rotate the vector using the body orientation quaternion at the start of the stroke using (1.8).

$$\begin{bmatrix} 0 \\ \\ _{body}\hat{n} \end{bmatrix} = {}^{wing}_{body}\hat{q}^{-1}_{stroke\ start} \times \begin{bmatrix} 0 \\ \\ _{stroke\ start}\hat{n} \end{bmatrix} \times {}^{wing}_{body}\hat{q}_{stroke\ start}$$

The projection of this axis of rotation onto the sagittal plane of the bird then represents the component of the wing stroke rotation that does not include twist. Therefore, stroke amplitude can be calculated as the magnitude of the total rotation multiplied by the proportion of the rotation that does not include twist.

$$stroke\ amplitude = \frac{\left\|_{body}\hat{n}_{\|sagittal}\right\|}{\left\|_{body}\hat{n}\right\|}\ \theta_{stroke}$$

where $\hat{\alpha}_{\|\beta}$ is the projection of the unit vector $\hat{\alpha}$ onto the plane β.

I can then calculate the inclination of the wing stroke plane relative to the body axes using the inverse tangent of the axis of the stroke rotation. Once again, this vector must be expressed in the body frame and projected onto the sagittal plane to reject twist.

$$stroke\ inclination = tan^{-1}\left(\frac{_{body}n_z}{_{body}n_x}\right)$$

As the pitch of the body relative to the world frame could change over the course of the stroke, I average the body pitch over the two stroke end points to approximate a body pitch for the whole stroke. I then calculate the inclination of the stroke plane in the world frame by subtracting the averaged body pitch from the stroke plane relative to the body.

Finally, I compare these estimated wing stroke parameters to those calculated from the optical motion capture data for corresponding strokes to assess the accuracy of this approach.

The two estimates of the orientation of the wing relative to the body are shown in Figure 2.10 with dashed lines indicating the ranges over which individual strokes were identified. Given the considerable noise in the OptiTrack traces near the wing stroke end points, I chose to filter the OptiTrack data by the first derivative of the x component of the relative orientation of the wing. The filtered OptiTrack data are also shown in Figure 2.10.

The zero points for the orientations estimated by the two systems differ as the markers on the wing are not visible during the closed wing posture used to define body axes in the inertial system. This results in the vertical offset seen between the Euler angle traces estimated by the two systems in Figure 2.10. Despite this, the wing stroke signal is clear in both systems. Further, as the wing stroke parameters are all calculated using the difference between the wing stroke end point orientations, this vertical offset should not affect our estimates. The wing stroke amplitude, inclination relative to the body, and inclination relative to the world frame for the three complete simulated wing strokes are shown in Table 2.1 along with the residual error. All stroke parameters were estimated within 5° for the three simulated wing strokes, which is consistent with previous results. Wing stroke amplitude was consistently overestimated by about 3°. However, due to the low sample size, it is difficult to tell if this was a coincidence or indicative of any systemic error.

Stroke amplitude (°)

Stroke index	Inertial sensor	OptiTrack	Difference
1	106.80	104.16	2.65
2	90.30	87.48	2.83
3	90.77	87.96	2.81

Stroke plane inclination, relative to body (°)

Stroke index	Inertial sensor	OptiTrack	Difference
1	-17.02	-12.44	-4.58
2	-16.10	-13.17	-2.93
3	-15.53	-15.63	0.10

Stroke plane inclination, relative to world (°)

Stroke index	Inertial sensor	OptiTrack	Difference
1	-5.67	-3.21	-2.47
2	-3.74	-3.08	-0.66
3	-3.26	-4.93	1.67

Table 2.1: Wing stroke parameters estimated using inertial sensors and OptiTrack for three simulated wing strokes.

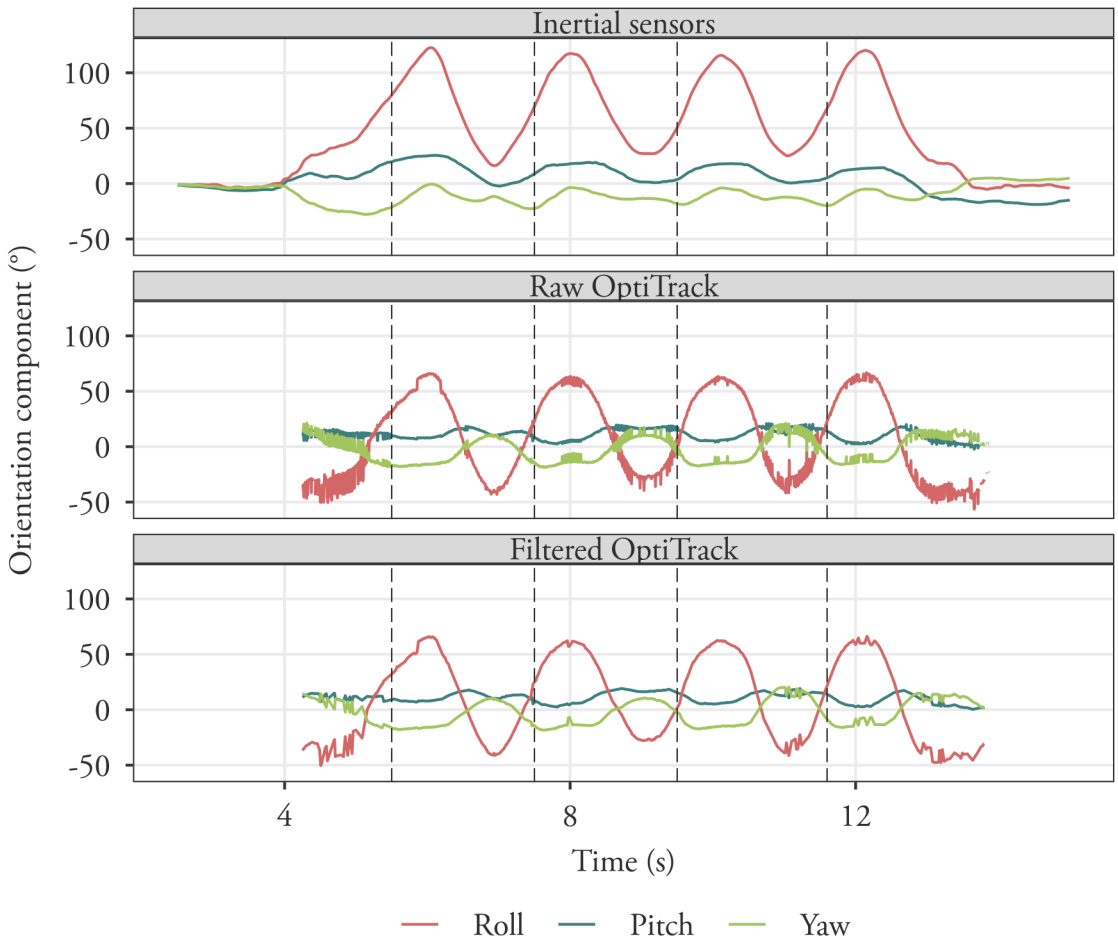

Figure 2.10: Orientation of the wing relative to the body estimated over three simulated wing strokes of a deceased pigeon using both inertial sensors and OptiTrack. Both the raw orientations estimated by OptiTrack as well as the filtered data used for estimating stroke parameters are shown. Vertical dashed lines indicate ranges over which individual strokes were defined.

2.5.5 Estimating wing stroke parameters *in vivo*

To assess the accuracy of these methods *in vivo*, I repeated the previous test using live pigeons. I once again mounted the sensors on the body and forearm of the pigeon along with retro-reflective markers for optical tracking. After defining the body axes for the inertial motion capture system, I released the pigeons from chest height and recorded the wing kinematics as they flew to the ground. I caught the pigeons after each flight and removed the sensors and markers before returning the

pigeons to their housing cage.

Unlike in previous comparisons between the two systems, an external trigger could not be used as it would require the birds to be tethered. Instead, I had planned to identify wing strokes independently in both systems and match them sequentially. I then planned to compare wing stroke parameters estimated for each pair of wing strokes in the two systems.

I set both systems to maximize sampling rate with the inertial data loggers sampling at 800Hz and OptiTrack recording at 240Hz with four cameras. As before, the body axes alignment postures were not captured in the OptiTrack recording as the markers were not visible with wings held closed. 10,000 particles were simulated in the Monte Carlo smoother and 16 trajectories were sampled during refiltering.

Euler angle traces of the orientation of the wing relative to the body as estimated by the inertial sensors are shown in Figure 2.11A. Over successive strokes, the wing appears to drift in roll, ultimately leading to wing motions that are completely unreasonable. The problem here is most clearly seen in the raw gyroscope data from the sensor mounted on the forearm, as shown in Figure 2.11B. The wing strokes consistently exceed the gyroscope measurement range of $\pm 2000°/s$, which after integration leads to consistent underestimation of motion. Accordingly, the drift seen in the estimated orientation of the wing is likely due to unequal portions of the upstroke and downstroke exceeding the sensing threshold.

2.6 Discussion

The inertial data loggers and orientation estimation algorithm developed in this project were able estimate the orientation of a single rigid body within 3° in controlled, simulated situations. This translated to an error of 6° when assessing body axes alignment and estimating joint dynamics, likely due to the accumulation of

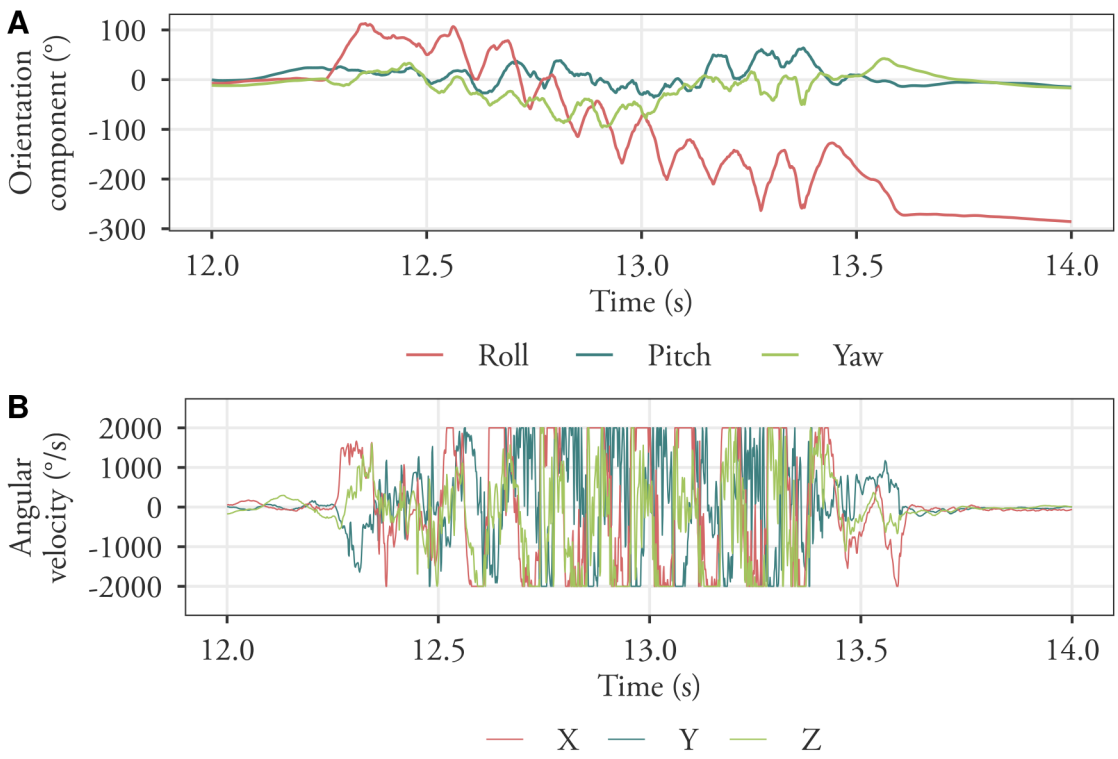

Figure 2.11: Data collected from the inertial tracking system mounted on a live pigeon during a short free flight. (A) Orientation of the wing relative to the body shows unreasonable drift in estimated roll. (B) Raw gyroscope data from the inertial sensor mounted on the wing shows motions that exceed the hardware sensing limits of ±2000°/s.

error from using two estimated orientations [72]. However, the data loggers could not be used for tracking wing kinematics *in vivo* as the sensors used in the current design did not have the necessary gyroscope sensing range.

The most direct solution to this problem would be to modify the hardware to use different inertial sensors. Assuming a flapping frequency of around 10Hz, which is typical for pigeons [73], a wing stroke amplitude of 180°, and a sinusoidal model of wing motion, we would expect a maximum angular velocity of approximately ±5600°/s around the stroke axis. Most gyroscopes built into MARG sensor chips have a maximum sensing range of ±2000°/s (e.g., LSM9DS1TR, BMX055, MX6470). This range likely reflect the typical angular rates experienced by sensors in mobile and automotive applications. These markets were valued at $1.2 and $2.7 billion

respectively in 2016 [74] and are likely to be responsible for the diversity and competitive pricing of inertial sensors available today. As a result, it is unlikely that newer iterations of these general purpose inertial sensors will greatly increase sensing range unless the market changes to include popular applications that require it. Such a change has happened in part with some newer sensors designed explicitly for smart wearable athletic devices, such as the ICM-20601 from Invensense, that have a gyroscope sensing range of $\pm 4000°/s$.

To achieve the sensing ranges needed for recording bird wing kinematics using currently available off-the-shelf components, we must make use of specialized sensors that separate MARG sensor functionality over multiple chips. The ADXRS649 by Analog Devices, for example, provides a gyroscope with a sensing range of $\pm 20{,}000°/s$ in a 6×7mm footprint. Note that a specialized magnetometer is not needed as the magnitude of the local magnetic field does not vary with angular rate. While the wing-mounted sensors will experience linear accelerations beyond the sensing range of the current data loggers, a specialized accelerometer is not necessary as these clipped observations are already disregarded by the orientation estimation algorithm as they do not provide information about the direction of gravity. See Section 2.4.2 for details. As a result, the sensing range issue can be overcome by pairing a specialized gyroscope to every sensor currently in the design. The MPU-9250 sensors in the current design could also be swapped out for the smaller MC6470 sensors as integrated gyroscope function would no longer be important.

Another option would be to limit the application of these sensors to bird species that have flapping frequencies within the sensing range of the current sensors, or near drop-in replacements like the aforementioned ICM-20601. Using the previous assumptions, the current data loggers should be capable of tracking the wing kinematics for flapping frequencies up to about 3.5Hz, or 7Hz with the sensors designed for athletic wearables. Table 2.2, adapted from Pennycuick [75], lists flapping

frequencies of various species measured in the field. With a safety factor of 50%, large corvids (*Corvus corone*), raptors (*Buteo buteo, Milvus milvus*), herons (*Ardea cinera*), waterfowl (*Cygnus olor*), and gulls (*Larus canus, Larus argentinus, Larus marinus*) could all feasibly be studied with minimal modification to the current hardware. The large raptors are of particular interest as many of these birds are trained for falconry, which could allow the current data loggers to be assessed in a fairly controlled setting. For comparison, modifying the data loggers to use the ADXRS649 gyroscope would allow for tracking wing kinematics for flapping frequencies up to 35Hz, or a little over 25Hz with a 50% safety factor. This would encompass the flapping frequencies seen in pigeons as well as the sixteen other species listed in Table 2.2.

Using the data loggers as is with a restricted set of species would severely limit their utility for comparative studies, as it would introduce a strong size bias. However, some bias is already unavoidable as a number of species, such as the 5g black-tailed gnatcatcher (*Polioptila melanura*), are too small to feasibly have untethered inertial data loggers mounted on them, at least given currently available electronic components. Further, these tools would still provide a novel high-throughput means of collecting wing kinematic data for addressing questions within species.

Beyond studies of bird flight, the tools developed in this project could also be used with minimal modification to implement inertial motion capture in many other vertebrate systems. Cheetahs and greyhounds, for example, only achieve a maximum stride frequency of about 4Hz during sprints [76]. Similarly, while some thunniform swimmers can achieve tail beat frequencies of around 10-12Hz during bursts [77], many fish employ much slower forms of body-caudal propulsion [78].

The primary benefit of these tools over the current commercially available inertial motion capture systems are their small size, high sample rates, and untethered operation [60–62]. Outside of human kinesiology, inertial motion capture has pre-

Species	Equivalent air speed ($m\,s^{-1}$)		Wingbeat frequency (Hz)	
	Observed	V_{mp}	Observed	f_{ref}
Corvus corone	10.5±2.03	12.1±0.976	3.84±0.0522	4.74±0.181
Sturnus vulgaris	17.3±1.75	10.2±0.864	10.6±1.25	9.97±0.385
Fringilla coelebs	15.3±3.46	7.82±0.715	18.2±1.74	10.8±0.691
Buteo buteo	9.31±1.92	12.3±1.21	3.63±0.168	3.54±0.218
Accipiter nisus	8.72±2.40	10.5±1.03	5.10±0.321	6.30±0.388
Milvus milvus	7.90±1.81	10.9±1.08	2.88±0.0926	2.75±0.170
Ardea cinerea	11.0±1.66	11.9±1.17	2.90±0.0354	2.79±0.172
Cygnus olor	16.0±0.700	19.4±2.08	3.38±0.0863	3.37±0.264
Anas penelope	17.1±1.99	14.3±1.41	6.83±0.330	7.27±0.500
Somateria mollissima	20.2±3.51	15.9±2.19	6.47±0.294	6.60±0.586
Phalacrocorax carbo	15.0±1.80	16.6±1.39	4.83±0.156	5.09±0.265
Columba palumbus	15.4±2.06	12.9±1.27	5.61±0.256	6.81±0.420
Larus ridibundus	10.1±1.89	9.42±0.964	3.27±0.0110	4.04±0.274
Larus canus	11.6±1.74	9.62±0.946	2.98±0.151	3.50±0.216
Larus argentatus	11.8±2.07	11.9±1.06	3.13±0.0761	3.61±0.218
Larus marinus	12.8±1.31	12.6±1.35	2.91±0.223	3.14±0.215

Numbers after the ± symbols are standard deviations of observed quantities and uncertainties of calculated quantities (N=6).
V_{mp}, minimum power speed; f_{ref}, calculated equivalent wingbeat frequency.

Table 2.2: Flight speed and flapping frequency of sixteen species collected in the field using an anemometer and video data. Adapted from Pennycuick [75].

viously only been applied to studies of gait in horses and dogs [63, 64], both of which are large and easily trained. These traits were likely necessary to mount large, tethered sensors to different parts of the body while trying to elicit natural behaviours.

The benefits of these tools over optical motion capture for systems other than bird flight are not so immediately clear. The primary drive for using inertial motion capture here was to overcome the challenges with optical motion capture that stem from the occlusion of marked points on the wing during flapping flight. In situations where markers can reliably be kept in view, optical motion capture will likely be preferred if only for the availability of community support with many of these more established tools, like DLTdv [9]. Newer iterations of these tools, like OptiTrack, can also do much of the work of digitizing the raw data, so long as markers remain sufficiently separated from one another and remain in view.

Put another way, the value of inertial motion capture for recording animal kinematics is primarily in recording behaviours that cannot easily be recorded using cameras. This includes behaviours that occur in situations with limited visibility, such as in burrows, murky water, or at night. Behaviours that occur over spatial scales that cannot feasibly be recorded by camera arrays are also of interest, such as migration, movement through forest canopy, and even prey capture in some species. Given the small size of the inertial data loggers, they could also be worked into existing long-term tracking devices. Although the loggers are not designed for long-term tracking, recent advances in automated behaviour classification using body-mounted accelerometer data [79–81] could be leveraged to trigger detailed kinematic data collection during behaviours of interest in a natural context. This targetted approach to kinematic data collection in freely roaming animals would not be possible with optical motion capture.

Chapter 3

Conclusion

Detailed wing kinematics are of broad interest to the study of bird flight [82–84], but current optical approaches to animal motion capture [8, 9] are ill-suited to collect these data. The considerable morphing of the wings [10] and periodic occlusion of flight surfaces during flapping flight makes it difficult to automate the tracking of fixed points on the wing using video data [85]. This necessitates manual digitization of the raw videos, which can be a major bottleneck to data collection.

In this thesis, I developed a series of tools with the goal of creating a high-throughput approach to the collection of wing kinematic data using body-mounted inertial data loggers. The physical hardware consisted of a modular design which can collect data from up to four sensors at 450Hz for 20min in a single session, all at around 4 grams including a battery. The accompanying orientation estimation algorithm used a recursive Bayesian framework and could estimate angles between sensors and simulated wing stroke parameters within 6° in situations where the angular velocities remained within the sensing range of the data loggers. An R package compiling these functions for general use is currently in preparation for submission to the Comprehensive R Archive Network (CRAN). The inertial data logger designs and firmware have similarly been made available online [86].

The most significant short-coming of the current design is the gyroscope, which does not have the sensing range to collect data from most bird species. However, as discussed in Section 2.6, the components needed to overcome this limitation are commercially available today. The non-removable memory system posed another limitation to the utility of the current design as the entire data logger needed to be unmounted from the subject between trials to transfer the data to a computer. Removable flash memory, like microSD cards, may be preferable in future designs despite their relatively slower speed and larger size for this reason. See Section 2.2.2 for the details of this choice for the current design.

Although I did not achieve my initial goal of producing a fully-realized high-throughput method of recording wing kinematics in birds, I believe that I have demonstrated the potential of inertial motion capture to this end. To my knowledge, this is the first application of biological inertial motion capture outside of humans, horses, and dogs [60, 63, 64], enabled in part by the small size of the new data loggers. I am also unaware of any other publicly available software for orientation estimation that uses the entire set of inertial measurements from a recording to estimate orientation at every time point *post hoc* [42, 43]. See Section 1.1.3 for the mathematical basis of this approach and Section 2.4.1 for the rationale for using this approach here.

3.1 Significance of high-throughput methods

The value of high-throughput methods in other applications is most clearly seen by the new sorts of studies these methods enable and the new questions these studies allow us to address. High-throughput methods for genome sequencing, for example, have allowed for studies of population-level genomic variation [87] and the development of statistical approaches to identifying functional genetic polymorphisms [88].

Similar advances in drug screening approaches for pharmacological studies have allowed researchers to validate newly proposed theoretical models for predicting drug solubility and permeability [89].

Although I was not able to apply the tools I developed in this thesis to address a novel question due to the limitations of the current hardware, we can still consider the value of high-throughput methods to wing kinematic data collection using this framework. In the following sections, I outline two studies that would be infeasible with current optical methods of recording wing kinematics in birds and the value these studies would provide to the field.

3.1.1 Formalizing flight style

Flight style, or an analogue to gait for bird flight, is a concept that has been approached at various times throughout the literature [90–93]. Categorizing flight kinematics in this way is useful as it helps summarize a complex set of traits into a much smaller number of ecologically or physiologically relevant groups, much like swimming styles for fish locomotion [78] and pollination syndromes for plant reproductive morphology [94]. Concise and informative flight style schemes are particularly useful for comparative work as it makes flight kinematics more tractable for inclusion as an explanatory variable or covariate [90, 95, 96]. However, an uninformative categorization scheme could be at best confusing or at worst misleading.

The various takes on flight style generally split the variation in flight kinematics into discrete categories based on one or a few criteria, such as the frequency and length of glides between bouts of flapping [91] or flapping frequency and expected metabolic costs [96]. However, it is not always clear what the appropriate number of discrete groups should be, as different schemes can have as few as three [97] or as many as eight [91] different groups. The variation in flight kinematics across and within some species also might fill the space between flight styles that are commonly

thought of as distinct [91, 98]. Accordingly, it might be more appropriate to think of flight style as a continuous space varying along one or more principle axes, or as a few distinct styles with considerable variation within each group. One approach to addressing this ambiguity would be to use a k-means clustering algorithm with model selection to find the most informative number of groups within the total space of wing kinematic variation across species [99]. To quantify wing kinematics for this approach, we could take the discrete Fourier transformation of the shoulder and elbow angles along each axis over time for a given flight [100]. In addition to capturing common parameters of interest like stroke amplitude and stroke plane inclination, this approach should also capture information about the length and frequency of intermittent glides or bounding. This is because flight styles with intermittent behaviours can be thought of as piecewise periodic functions, which Fourier transformations can be used to reconstruct [101].

Using this framework, we could also assess how informative the groupings proposed by existing flight style schemes are using an information criterion [102]. This could then be used to compare existing flight schemes to one another as well as to the most informative model from the clustering algorithm. The ratio of flight kinematic variation within groups to the variation across groups could also be interesting as it would give an indication of whether flight styles are very distinct and internally uniform (much lower variation within groups than across) or only loosely separated with meaningful internal variation (comparable variation within and across groups).

While there is nothing particularly novel about this statistical approach, it is currently completely infeasible to meaningfully sample the total variation in wing kinematics across species, even within a singular behaviour like level flight. Previous studies have made use of summarized flight parameters collected from a diversity of species, such as flight speed [75, 103], flapping frequency [75, 92], and the presence or absence of specific behaviours [90]. However, a high-throughput means of collecting

detailed wing kinematics would be needed to collect the scale of data needed for a complete reassessment flight style schemes.

3.1.2 Deconstructing flight control

Although it is well established that animals that use flapping flight primarily maneuver in the air using variation in their wing kinematics [1], it is less clear which aspects of the wing stroke birds actually modify to elicit specific changes to their flight path.

Part of the difficulty with addressing this problem is that there often many different ways a bird could theoretically accomplish a given change to their flight path. Dakin *et al* [19], for example, found that hummingbirds could turn either by rotating in place while hovering, banking during forward flight to perform a smooth arc turn, or by pitching up to perform a tight pitch-roll turn to quickly change direction. Lower-level maneuvers like changing pitch or yaw might similarly be achieved in different ways or could even depend on context, such as the flight speed of the bird. Accordingly, an analysis of the kinematics used to effect one of these behaviours (e.g., changes in elevation or speed) would require sufficient power to distinguish between distinct kinematic modes and to catch potentially subtle differences in the wing stroke. Characterizing the kinematic changes that underlie multiple types of maneuvers only compound this need for large sample sizes, and quickly becomes infeasible using current methods.

The other key challenge to this sort of study would be motivating the birds to produce the behaviours that we are interested in. One option would be to allow the birds to fly freely in a space and use a correlational approach to explore the connection between the wing kinematics and flight parameters of interest [19, 56]. A more direct approach however could use artificial visual flow stimuli to motivate specific changes to level flight. Visual flow is the global motion seen by an animal

as it moves through its environment [104], and these cues have been found to be important for course correction in a diversity of species [105–107]. Various studies have also previously used these cues to experimentally manipulate behaviour by presenting animals with artificial visual stimuli corresponding to different visual flow fields [108–110].

Putting these ideas together, pigeons could be flown in flight tunnels with visual stimuli projected onto the walls as in Dakin *et al* [108]. A stationary, uniform stimulus would be used as a control treatment, while moving horizontal and vertical gratings could be used to induce vertical and horizontal visual flow, respectively. We expect from past work with other birds that pigeons will correct their altitude to track the vertical visual flow in the horizontal grating treatments [108], and similarly may slow down or speed up to track the horizontal visual flow in the vertical grating treatments [111]. OptiTrack or similar optical tracking methods could be used to track the position of the birds in the tunnel to test this expectation [19]. Wing kinematic data collected by body-mounted inertial sensors could then be used to test specific hypotheses about the wing kinematics the underlie the changes in trajectory. For example, we might hypothesize that changes in elevation are achieved by modifying the stroke amplitude to control lift production [112], while flight speed is controlled by inclining the stroke plane to redirect aerodynamic forces [12].

3.2 Summary

Understanding the variation in bird wing kinematics across different contexts and levels of organization is an important step to furthering our understanding of the physics and control of bird flight. However, current methods for animal motion capture are ill-suited for this problem and require considerable time and effort to

employ. A high-throughput alternative would allow for both broader comparative and more detailed mechanistic studies of bird flight. To this end, I began development on a high-throughput, body-mounted inertial motion capture system for recording wing kinematics in freely flying birds. Although the data loggers require further modification to allow them to sense the range of angular rates experienced during flapping flight, I believe I have demonstrated the potential of this approach to implementing a cheap, portable, and automated means of animal motion capture.

References

1. Chin, D. D. & Lentink, D. Flapping wing aerodynamics: from insects to vertebrates. *The Journal of Experimental Biology* **219,** 920–932 (2016).

2. Hedenstrom, A. & Johansson, L. C. Bat flight: aerodynamics, kinematics and flight morphology. *Journal of Experimental Biology* **218,** 653–663 (2015).

3. Tobalske, B. W. Biomechanics of bird flight. *Journal of Experimental Biology* **210,** 3135–3146 (2007).

4. Dickinson, M. H., Lehmann, F. & Pane, S. P. Wing rotation and the aerodynamic basis of insect flight. *Science* **284,** 1954–1960 (1999).

5. Ravi, S. *et al.* Modulation of flight muscle recruitment and wing rotation enables hummingbirds to mitigate aerial roll perturbations. *Current Biology* **30,** 187–195.e4 (2020).

6. Iriarte-Diaz, J. & Swartz, S. M. Kinematics of slow turn maneuvering in the fruit bat *Cynopterus brachyotis*. *Journal of Experimental Biology* **211,** 3478–3489 (2008).

7. Lehmann, F. The aerodynamic effects of wing-wing interaction in flapping insect wings. *Journal of Experimental Biology* **208,** 3075–3092 (2005).

8. Mathis, A. *et al.* DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience* **21,** 1281–1289 (2018).

9. Hedrick, T. L. Software techniques for two- and three-dimensional kinematic measurements of biological and biomimetic systems. *Bioinspiration & Biomimetics* **3,** 034001 (2008).

10. Tobalske, B. & Dial, K. Flight kinematics of black-billed magpies and pigeons over a wide range of speeds. *Journal of Experimental Biology* **199,** 263–280 (1996).

11. Lapsansky, A. B., Igo, J. A. & Tobalske, B. W. Zebra finch (*Taeniopygia guttata*) shift toward aerodynamically efficient flight kinematics in response to an artificial load. *Biology Open* **8,** bio042572 (2019).

12. Chin, D. D. & Lentink, D. Birds repurpose the role of drag and lift to take off and land. *Nature Communications* **10** (2019).

13. Tomotani, B. M. & Muijres, F. T. A songbird compensates for wing molt during escape flights by reducing the molt gap and increasing angle of attack. *The Journal of Experimental Biology* **222,** jeb195396 (2019).

14. McFarlane, L., Altringham, J. D. & Askew, G. N. Intra-specific variation in wing morphology and its impact on take-off performance in blue tits (*Cyanistes caeruleus*) during escape flights. *The Journal of Experimental Biology* **219,** 1369–1377 (2016).

15. Dickerson, B. H., de Souza, A. M., Huda, A. & Dickinson, M. H. Flies regulate wing motion via active control of a dual-function gyroscope. *Current Biology* **29,** 3517–3524.e3 (2019).

16. Chakraborty, S., Bartussek, J., Fry, S. N. & Zapotocky, M. Independently controlled wing stroke patterns in the fruit fly *Drosophila melanogaster*. *PLoS ONE* **10** (ed Gronenberg, W.) e0116813 (2015).

17. Straw, A. D., Branson, K., Neumann, T. R. & Dickinson, M. H. Multi-camera real-time three-dimensional tracking of multiple flying animals. *Journal of The Royal Society Interface* **8,** 395–409 (2010).

18. Lentink, D. Accurate fluid force measurement based on control surface integration. *Experiments in Fluids* **59** (2017).

19. Dakin, R., Segre, P. S., Straw, A. D. & Altshuler, D. L. Morphology, muscle capacity, skill, and maneuvering ability in hummingbirds. *Science* **359,** 653–657 (2018).

20. Ingersoll, R. & Lentink, D. How the hummingbird wingbeat is tuned for efficient hovering. *The Journal of Experimental Biology* **221,** jeb178228 (2018).

21. Sarkka, S. *Bayesian filtering and smoothing* (Cambridge University Press, 2009).

22. Gagniuc, P. A. *Markov chains* (John Wiley & Sons, Inc., 2017).

23. Broemeling, L. D. *Bayesian analysis of linear models* (Routledge, 2017).

24. Bajo, M. Improving storm surge forecast in Venice with a unidimensional Kalman filter. *Estuarine, Coastal and Shelf Science* **239,** 106773 (2020).

25. Durantin, G., Scannella, S., Gateau, T., Delorme, A. & Dehais, F. Processing functional near infrared spectroscopy signal with a Kalman filter to assess working memory during simulated flight. *Frontiers in Human Neuroscience* **9** (2016).

26. Dorner, B., Peterman, R. M. & Haeseker, S. L. Historical trends in productivity of 120 Pacific pink, chum, and sockeye salmon stocks reconstructed by using a Kalman filter. *Canadian Journal of Fisheries and Aquatic Sciences* **65,** 1842–1866 (2008).

27. Kalman, R. E. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering* **82,** 35–45 (1960).

28. McGee, L. A. & Schmidt, S. F. *Discovery of the Kalman filter as a practical tool for aerospace and industry* tech. rep. NASA-TM-86847, N86-13311 (National Aeronautics and Space Administration, 1985).

29. Kendall, G. *Would your mobile phone be powerful enough to get you to the moon?* https://theconversation.com/would-your-mobile-phone-be-powerful-enough-to-get-you-to-the-moon-115933 (2020).

30. Kalman, R. E. & Bucy, R. S. New results in linear filtering and prediction theory. *Journal of Basic Engineering* **83,** 95–108 (1961).

31. Wan, E. A. & van der Merwe, R. in *Kalman filtering and neural networks* 221–280 (John Wiley & Sons, Inc., 2001).

32. Kitagawa, G. Non-Gaussian state-space modeling of nonstationary time series. *Journal of the American Statistical Association* **82,** 1032 (1987).

33. Smith, R. & Miller, J. A non-Gaussian state-space model and application to prediction of records. *Journal of the Royal Statistical Society Series B-Methodological* **48,** 79–88 (1986).

34. Aravkin, A., Burke, J. V., Ljung, L., Lozano, A. & Pillonetto, G. Generalized Kalman smoothing: modeling and algorithms. *Automatica* **86,** 63–86 (2017).

35. Sharia, T. Recursive parameter estimation: convergence. *Statistical Inference for Stochastic Processes* **11,** 157–175 (2007).

36. Kitagawa, G. A self-organizing state-space model. *Journal of the American Statistical Association* **93,** 1203 (1998).

37. Liu, J. & West, M. in *Sequential Monte Carlo methods in practice* 197–223 (Springer New York, 2001).

38. Yang, B., Stroud, J. R. & Huerta, G. Sequential Monte Carlo smoothing with parameter estimation. *Bayesian Analysis* **13,** 1137–1161 (2018).

39. Seel, T., Raisch, J. & Schauer, T. IMU-based joint angle measurement for gait analysis. *Sensors* **14,** 6891–6909 (2014).

40. Burr, D. B. Why bones bend but don't break. *Journal of Musculoskeletal & Neuronal Interactions* **11,** 270–285 (2011).

41. Paletta, G. A., Warner, J. J., Warren, R. F., Deutsch, A. & Altchek, D. W. Shoulder kinematics with two-plane x-ray evaluation in patients with anterior instability or rotator cuff tearing. *Journal of Shoulder and Elbow Surgery* **6,** 516–527 (1997).

42. Madgwick, S. O. H., Harrison, A. J. L. & Vaidyanathan, R. *Estimation of IMU and MARG orientation using a gradient descent algorithm* in *2011 IEEE International Conference on Rehabilitation Robotics* (IEEE, 2011).

43. Mahony, R., Hamel, T. & Pflimlin, J.-M. Nonlinear complementary filters on the special orthogonal group. *IEEE Transactions on Automatic Control* **53,** 1203–1218 (2008).

44. Weisstein, E. W. *Euler Angles. From MathWorld–A Wolfram Web Resource* https://mathworld.wolfram.com/EulerAngles.html (2020).

45. Hemingway, E. G. & O'Reilly, O. M. Perspectives on Euler angle singularities, gimbal lock, and the orthogonality of applied forces and applied moments. *Multibody System Dynamics* **44,** 31–56 (2018).

46. Duncan, M. *Applied geometry for computer graphics and CAD* (Springer London, 2005).

47. Strang, G. *Linear algebra and its applications* (Thomson, Brooks/Cole, Belmont, CA, 2006).

48. Weisstein, E. W. *Vector space orientation. From MathWorld–A Wolfram Web Resource* https://mathworld.wolfram.com/VectorSpaceOrientation.html (2020).

49. Lengyel, E. *Mathematics for 3d game programming and computer graphics* (Charles River Media, Inc., USA, 2001).

50. Ma, K. F., Cui, X. M., Huang, G. P., Yuan, D. B. & Cai, Q. K. A solution for space resection based on angle-axis representation. *Applied Mechanics and Materials* **638-640,** 2123–2128 (2014).

51. Leon, S. J., Björck, Å. & Gander, W. Gram-Schmidt orthogonalization: 100 years and more. *Numerical Linear Algebra with Applications* **20,** 492–532 (2012).

52. Hamilton, W. R. II. On quaternions; or on a new system of imaginaries in algebra. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **25,** 10–13 (1844).

53. Clemente, C. J. *et al.* Moving in complex environments: a biomechanical analysis of locomotion on inclined and narrow substrates. *The Journal of Experimental Biology* **222,** jeb189654 (2019).

54. Gellman, E. D., Tandler, T. R. & Ellerby, D. J. Swimming from coast to coast: a novel fixed-gear swimming gait in fish. *Biology Letters* **15,** 20190270 (2019).

55. Nasir, N., Lehmann, F. O., Schützner, P., Mat, S. & Mohd, N. N. Dynamics of body kinematics of freely flying houseflies responding to air turbulence. *Journal of Asia-Pacific Entomology* **22,** 1082–1089 (2019).

56. Walker, S. M., Thomas, A. L. & Taylor, G. K. Photogrammetric reconstruction of high-resolution surface topographies and deformable wing kinematics of tethered locusts and free-flying hoverflies. *Journal of The Royal Society Interface* **6,** 351–366 (2008).

57. Brackenbury, J. H. Kinematics of take-off and climbing flight in butterflies. *Journal of Zoology* **224,** 251–270 (1991).

58. Durston, N. E., Wan, X., Liu, J. G. & Windsor, S. P. Avian surface reconstruction in free flight with application to flight stability analysis of a barn owl and peregrine falcon. *The Journal of Experimental Biology* **222,** jeb185488 (2019).

59. Deetjen, M. E., Biewener, A. A. & Lentink, D. High-speed surface reconstruction of a flying bird using structured light. *The Journal of Experimental Biology* **220,** 1956–1961 (2017).

60. Robert-Lachaine, X., Parent, G., Fuentes, A., Hagemeister, N. & Aissaoui, R. Inertial motion capture validation of 3D knee kinematics at various gait speed on the treadmill with a double-pose calibration. *Gait & Posture* **77,** 132–137 (2020).

61. Robert-Lachaine, X., Mecheri, H., Muller, A., Larue, C. & Plamondon, A. Validation of a low-cost inertial motion capture system for whole-body motion analysis. *Journal of Biomechanics* **99,** 109520 (2020).

62. Abdelhady, M., van den Bogert, A. J. & Simon, D. A high-fidelity wearable system for measuring lower-limb kinetics and kinematics. *IEEE Sensors Journal* **19,** 12482–12493 (2019).

63. Egan, S., Brama, P. & McGrath, D. Research trends in equine movement analysis, future opportunities and potential barriers in the digital age: A scoping review from 1978 to 2018. *Equine Veterinary Journal* **51,** 813–824 (2019).

64. Duerr, F. *et al.* Evaluation of inertial measurement units as a novel method for kinematic gait evaluation in dogs. *Veterinary and Comparative Orthopaedics and Traumatology* **29,** 475–483 (2016).

65. Godsill, S. J., Doucet, A. & West, M. Monte Carlo Smoothing for nonlinear time series. *Journal of the American Statistical Association* **99,** 156–168 (2004).

66. Palermo, E., Rossi, S., Marini, F., Patanè, F. & Cappa, P. Experimental evaluation of accuracy and repeatability of a novel body-to-sensor calibration procedure for inertial sensor-based gait analysis. *Measurement* **52,** 145–155 (2014).

67. Cariow, A., Cariowa, G. & Majorkowska-Mech, D. An algorithm for quaternion–based 3D rotation. *International Journal of Applied Mathematics and Computer Science* **30,** 149–160 (2020).

68. Julier, S. J. & Uhlmann, J. K. *New extension of the Kalman filter to nonlinear systems* in *Signal Processing, Sensor Fusion, and Target Recognition VI* (ed Kadar, I.) (SPIE, 1997).

69. Rose, D. *Euler Angle Visualization Tool* http://danceswithcode.net/engineeringnotes/rotations_in_3d/demo3D/%20rotations_in_3d_tool.html (2020).

70. Altshuler, D. L., Quicazan-Rubio, E. M., Segre, P. S. & Middleton, K. M. Wingbeat kinematics and motor control of yaw turns in Anna's hummingbirds (*Calypte anna*). *Journal of Experimental Biology* **215,** 4070–4084 (2012).

71. Berg, A. M. & Biewener, A. A. Wing and body kinematics of takeoff and landing flight in the pigeon (*Columba livia*). *Journal of Experimental Biology* **213,** 1651–1658 (2010).

72. For Guides in Metrology, J. C. *JCGM 100: Evaluation of Measurement Data - Guide to the Expression of Uncertainty in Measurement* tech. rep. (Joint Committee for Guides in Metrology, 2008).

73. Dial, K. P. Activity patterns of the wing muscles of the pigeon (Columba livia) during different modes of flight. *Journal of Experimental Zoology* **262,** 357–373 (1992).

74. Robin, L. *New dynamics of the MEMS inertial sensor market* in *China Semiconductor Technology International Conference* (Electrochemical Society, 2012).

75. Pennycuick, C. Speeds and wingbeat frequencies of migrating birds compared with calculated benchmarks. *Journal of Experimental Biology* **204,** 3283–3294 (2001).

76. Hudson, P. E., Corr, S. A. & Wilson, A. M. High speed galloping in the cheetah (*Acinonyx jubatus*) and the racing greyhound (*Canis familiaris*): spatio-temporal and kinetic characteristics. *Journal of Experimental Biology* **215,** 2425–2434 (2012).

77. Svendsen, M. B. S. *et al.* Maximum swimming speeds of sailfish and three other large marine predatory fish species based on muscle contraction time and stride length: a myth revisited. *Biology Open* **5,** 1415–1419 (2016).

78. Sfakiotakis, M., Lane, D. & Davies, J. Review of fish swimming modes for aquatic locomotion. *IEEE Journal of Oceanic Engineering* **24,** 237–252 (1999).

79. Fuchs, N. T. & Caudill, C. C. Classifying and inferring behaviors using real-time acceleration biotelemetry in reproductive steelhead trout (*Oncorhynchus mykiss*). *Ecology and Evolution* **9,** 11329–11343 (2019).

80. Chakravarty, P., Cozzi, G., Ozgul, A. & Aminian, K. A novel biomechanical approach for animal behaviour recognition using accelerometers. *Methods in Ecology and Evolution* **10** (ed O'Hara, R. B.) 802–814 (2019).

81. Studd, E. K. *et al.* Behavioral classification of low-frequency acceleration and temperature data from a free-ranging small mammal. *Ecology and Evolution* **9,** 619–630 (2018).

82. Clark, C. J. & Mistick, E. A. Kinematic control of male Allen's hummingbird wing trill over a range of flight speeds. *The Journal of Experimental Biology* **221,** jeb173625 (2018).

83. Gurka, R. *et al.* Flow pattern similarities in the near wake of three bird species suggest a common role for unsteady aerodynamic effects in lift generation. *Interface Focus* **7,** 20160090 (2017).

84. Mahalingam, S. & Welch, K. C. Neuromuscular control of hovering wingbeat kinematics in response to distinct flight challenges in the ruby-throated hummingbird, Archilochus colubris. *Journal of Experimental Biology* **216,** 4161–4171 (2013).

85. Smeulders, A. W. M. *et al.* Visual Tracking: An Experimental Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **36,** 1442–1468 (2014).

86. Senthivasan, S. *kinavr* https://github.com/shreeramsenthi/kinavr. 2020.

87. Altshuler, D. *et al.* A map of human genome variation from population-scale sequencing. *Nature* **467,** 1061–1073 (2010).

88. Wang, K., Li, M. & Hakonarson, H. ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data. *Nucleic Acids Research* **38,** e164–e164 (2010).

89. Lipinski, C. A., Lombardo, F., Dominy, B. W. & Feeney, P. J. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced Drug Delivery Reviews* **23,** 3–25 (1997).

90. Baliga, V. B., Szabo, I. & Altshuler, D. L. Range of motion in the avian wing is strongly associated with flight behavior and body mass. *Science Advances* **5,** eaaw6670 (2019).

91. Bruderer, B., Peter, D., Boldt, A. & Liechti, F. Wing-beat characteristics of birds recorded with tracking radar and cine camera. *Ibis* **152,** 272–291 (2010).

92. Pennycuick, C. Wingbeat frequency of birds in steady cruising flight: New data and improved predictions. *Journal of Experimental Biology* **199,** 1613–1618 (1996).

93. Savile, O. B. O. Adaptive evolution in the avian wing. *Evolution* **11,** 212–224 (1957).

94. Fenster, C. B., Armbruster, W. S., Wilson, P., Dudash, M. R. & Thomson, J. D. Pollination syndromes and floral specialization. *Annual Review of Ecology, Evolution, and Systematics* **35,** 375–403 (2004).

95. Shatkovska, O. V. & Ghazali, M. Relationship between developmental modes, flight styles, and wing morphology in birds. *The European Zoological Journal* **84,** 390–401 (2017).

96. Viscor, G. & Fuster, J. Relationships between morphological parameters in birds with different flying habits. *Comparative Biochemistry and Physiology Part A: Physiology* **87,** 231–249 (1987).

97. Lees, J. J., Dimitriadis, G. & Nudds, R. L. The influence of flight style on the aerodynamic properties of avian wings as fixed lifting surfaces. *PeerJ* **4,** e2495 (2016).

98. Usherwood, J. R. Physiological, aerodynamic and geometric constraints of flapping account for bird gaits, and bounding and flap-gliding flight strategies. *Journal of Theoretical Biology* **408,** 42–52 (2016).

99. Xu, R. & WunschII, D. Survey of clustering algorithms. *IEEE Transactions on Neural Networks* **16,** 645–678 (2005).

100. Shumway, R. H. & Stoffer, D. S. *Time series analysis and its applications* (Springer International Publishing, 2017).

101. Batenkov, D. & Yomdin, Y. Algebraic Fourier reconstruction of piecewise smooth functions. *Mathematics of Computation* **81,** 277–318 (2012).

102. Akogul, S. & Erisoglu, M. A comparison of information criteria in clustering based on mixture of multivariate normal distributions. *Mathematical and Computational Applications* **21,** 34 (2016).

103. Alerstam, T., Rosén, M., Bäckman, J., Ericson, P. G. P. & Hellgren, O. Flight speeds among bird species: Allometric and phylogenetic effects. *PLoS Biology* **5** (ed Sheldon, B.) e197 (2007).

104. Lappe, M., Bremmer, F. & van den Berg, A. Perception of self-motion from visual flow. *Trends in Cognitive Sciences* **3,** 329–336 (1999).

105. Gläser, N. *et al.* Harbor seals (*Phoca vitulina*) can perceive optic flow under water. *PLoS ONE* **9** (ed Ropert-Coudert, Y.) e103555 (2014).

106. Reiser, M. B. & Dickinson, M. H. Visual motion speed determines a behavioral switch from forward flight to expansion avoidance in *Drosophila*. *Journal of Experimental Biology* **216,** 719–732 (2012).

107. Mohler, B. J., Thompson, W. B., Creem-Regehr, S. H., Pick, H. L. & Warren, W. H. Visual flow influences gait transition speed and preferred walking speed. *Experimental Brain Research* **181,** 221–228 (2007).

108. Dakin, R., Fellows, T. K. & Altshuler, D. L. Visual guidance of forward flight in hummingbirds reveals control based on image features instead of pattern velocity. *Proceedings of the National Academy of Sciences* **113,** 8849–8854 (2016).

109. Saleem, A. B., Ayaz, A., Jeffery, K. J., Harris, K. D. & Carandini, M. Integration of visual motion and locomotion in mouse visual cortex. *Nature Neuroscience* **16,** 1864–1869 (2013).

110. Mulavara, A. P. *et al.* Exposure to a rotating virtual environment during treadmill locomotion causes adaptation in heading direction. *Experimental Brain Research* **166,** 210–219 (2005).

111. Schiffner, I. & Srinivasan, M. V. Direct evidence for vision-based control of flight speed in budgerigars. *Scientific Reports* **5** (2015).

112. Groom, D. J. E., Toledo, M. C. B. & Welch, K. C. Wingbeat kinematics and energetics during weightlifting in hovering hummingbirds across an elevational gradient. *Journal of Comparative Physiology B* **187,** 165–182 (2016).