State Estimation of a Cheetah Spine and Tail Using an Inertial Sensor Network

Presented by:
Callen Fisher

Prepared for:
Dr. A. Patel
Prof. E. Boje
Dept. of Electrical and Electronics Engineering
University of Cape Town

Submitted to the Department of Electrical Engineering at the University of Cape Town in fulfilment of the academic requirements for a Masters of Science degree in Electrical Engineering.

November 9, 2015
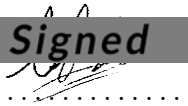
Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.

2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this report from the work(s) of other people has been attributed, and has been cited and referenced.

3. This report is my own work.

4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof.

Signature:.................................

Callen Fisher

Date: November 9, 2015

Acknowledgements

Abstract

The cheetah (*Acinonyx jubatus*) is by far the most manoeuvrable and agile terrestrial animal [1]. Little is known, in terms of biomechanics, about how it achieves these incredible feats of manoeuvrability [2]. The transient motions of the cheetah all involve rapid flicking of its tail and flexing of its spine [3].

The aim of the research was to develop tools (hardware and software) that can be used to gain a better understanding of the cheetah tail and spine by capturing its motion. A mechanical rig was used to simulate the tail and spine motion. This insight may inspire and aid in the design of bio-inspired robotic platforms.

A previous assumption was that the tail is heavy and acts as a counter balance or rudder [4] [5] [6] [7] [8], yet this was never tested. Contrary to this assumption, necropsy results determined that the tail was in fact light with a relatively low inertia value. Fur from the tail was used in wind tunnel experiments to determine the drag coefficient of a cheetah tail.

No researchers have actively sought to track the motion of a cheetah's spine and tail during rapid manoeuvres via placing multiple sensors on a cheetah. This requires the development of a 3D dynamic model of the spine and tail to accurately study the motion of the cheetah.

A wireless sensor network was built and three different filters and state estimation algorithms were designed and validated with a mechanical rig and camera system. The sensor network consists of three sensors on the tail (base, middle and tip) as well as a hypothetical collar sensor (GPS and WiFi were not implemented).

The raw sensor data was filtered using the kinematic and kinetic equations generated using Lagrange Dynamics by modelling the spine and tail as a system of interconnected rigid beams. The filters are in the form of the Extended Kalman Filter (EKF) and simulated in Matlab using a combination of analytical and numeric methods. The filters are advanced to include the torque generated by the tail on the body. This required modelling the drag effect of the cheetah's tail.

To conclude the research, initial investigation into a single beam tail as well as a position estimator were conducted. A single beam tail model results in fewer sensors being placed onto the cheetah and the position estimator enables the filter and state estimation algorithms to go from a stationary rig to a moving cheetah.

Contents

List of Figures

List of Tables

Glossary and Acronyms

IMU: Inertial measurement unit
KF: Kalman Filter
EKF: Extended Kalman Filter
RMS: Root mean square
DCM: Direction cosine matrix
PWM: Pulse width modulation
MOCAP: Motion capture
M: Mass matrix
C: Coriolis and centrifugal force matrix
G: Potential energy matrix
Q: Generalized force and torque matrix
q: Generalized coordinates vector
2D: Two dimensions
3D: Three dimensions
COM: Centre of mass
GPS: Global Positioning System
dps: Degrees per second
RF: Radio frequency
RTOS: Real time operating system
fps: Frames per second
COP: Centre of pressure

List of Symbols

I: Identity matrix
J: Moment of inertia
θ: Pitch angle
ψ: Yaw angle
ϕ: Roll angle
τ: Applied torque
ω: Angular rate
∂: Partial derivative
ρ: Density of air
C_D: Drag coefficient
Δ: Change in
α: Angular acceleration
Δt: Sample interval

Chapter 1

Introduction

For many years engineers and biologists have been observing the cheetah (*Acinonyx jubatus*) due to its incredible high speed in rough terrain [9] and this has led to many bio-inspired robotic platforms [1] [10] [11]. The aim of this research was to analyse and track the motion of the cheetah's spine and tail, see Figure 1.1, as well as to estimate the applied torques due to flicking the tail. For a complete understanding of this animal, high fidelity data is required for the whole body motion, which has not been previously possible to obtain.

Figure 1.1: Image showing a cheetah chasing a gazelle. Note the rigid and straight tail as well as the flexed spine. Image from [12].

1.1 Motivation and Background to the Study

Nature documentaries [13] [14] [15] involving cheetahs running in rough terrain, along with the MIT Robo-Cheetah [11] [16] and Boston Dynamics cheetah robot "Wildcat" [10] were the inspiration of this project and subsequent research. The cheetah is the fastest terrestrial mammal [17] [18] yet there is little available insight as to how the animal achieves its manoeuvrability.

Little is known about the transient dynamics (rapid accelerating, braking and turning) compared to the steady state dynamics (walking, trotting, running etc.) of quadruped animals [2], the cheetah in particular. As yet there are no available kinematics (the motion of a system of bodies) or kinetics (the forces that cause motion) model of the whole body of the cheetah that can be used to gain insight to its motion.

In order to study the motion of the cheetah accurately, wild cheetahs need to be examined. Wilson *et al.* [20] and [19] are the only researchers that have attached sensors to a cheetah. A single sensor was attached to a tracking collar and therefore only captured the animal as a point mass and ignored the rest of the body. To date, multiple sensors have not been attached to a cheetah in order to track the whole-body motion.

There are currently no available or feasible motion capture systems that can be used to analyse the motion of the spine and tail of the cheetah in the wild. A Vicon [21] camera system can be used but due to the fact that wild cheetahs need to be studied, this method is not feasible due to the capture region limitation of the camera system. A better understanding of how the cheetah moves will aid in many bio inspired robotic platforms.

1.2 Aims of the Study

This research was aimed at developing a system to estimate the states of a cheetah's spine and tail during rapid manoeuvres using low cost MEMs sensors.

1.2.1 Objectives of the Study

The objectives of the research were to generate a kinematics and kinetics model, using Lagrange dynamics, of the cheetah (the spine and the tail) to help analyse its motion through animations and to calculate the torques induced onto the body due to the rapid flicking of the tail. Insight for the development of the models, which were implemented numerically and analytically in Matlab, came via observing the anatomy of the cheetah with special focus on the tail. The models were used along with a unique custom built non-invasive wireless MOCAP system along with state estimation algorithms to track

and study the motion of the cheetah. The model shed light on the torques generated by the tail [3] and included the recently measured aerodynamic effects of the tail.

A mechanical test rig was designed to simulate a cheetah bending its spine and flicking its tail. The MOCAP system was placed onto the rig to simulate the data that was required for the state estimation algorithm. The tests were performed under a camera system in order to validate the results of the state estimation algorithm. Unfortunately the system was not tested on wild cheetahs.

1.2.2   Purpose of the Study

The purpose of the research was to develop a dynamic model of the cheetah, which is an endangered animal. This aids in studying the motion of the spine and tail to further the bio-inspired and bio-logging fields of research. To date, no researchers have studied the cheetah's high speed manoeuvres by placing multiple sensors onto the animal. This research will help engineers and biologists understand how the animal manoeuvres in unpredictable and unstructured terrain.

The results from the developed motion capture system and state estimation algorithms, which were in the form of an EKF (extended Kalman filter), will aid in studying the cheetah's transient manoeuvres. The algorithms also estimated the reaction torque generated by the tail; via drag experiments, a better understanding of the aerodynamic effect on the cheetah's body due to the tail was obtained.

The outcome of the research will help aid many bio-inspired robots and researchers. It is aimed at helping biologists and engineers to get a better understanding of the cheetah by analysing the motion of the spine and tail with insight into the torque generated by the tail. It also gives insight into the aerodynamic effects of the cheetah tail.

1.3   Scope and Limitations

The outcome of the project was an algorithm that is capable of tracking the states of a cheetah's tail and spine. This includes a full 3D kinematics and kinetics model of the cheetah (spine and tail, the scope does not include the legs) along with an animation to visualize the model.

A wireless sensor network was developed and placed onto a mechanical rig which simulated a cheetah bending its spine and flicking its tail. The data from the sensor network was used on the developed models, filters and state estimation algorithms. These algorithms were validated by a camera system.

One of the limitations of this research was that sensors cannot simply be attached to a

wild animal. Ethics approval was required and therefore was not part of the scope of the project. The sensor network was tested on a mechanical rig which resulted in the legs not being modelled. Another limitation was that a camera-based motion capture system [22] was used to validate the results of the state estimation algorithms. Due to the fact that a camera system was used to validate the results, the forward velocity of the cheetah was not part of the scope.

Due to video observations [13] [14] [15], cheetah tail fur was placed in a wind tunnel to determine the aerodynamic effects of the fur. This was then compared to a smooth cylinder and the results from the experiment were used to determine the torque that was generated by the tail on the body of the cheetah. This effect was modelled.

1.4 Research Questions

The following research questions were asked and answered during the study:

1. Can one estimate the states of a cheetah's tail and spine using a low cost IMU sensor network?

2. Is it possible to calculate an estimate of the torques induced onto the cheetah's body by its tail using low cost IMU sensors?

1.5 Plan of Development

The dissertation begins by explaining all the relevant and available literature with respect to the cheetah, motion capture systems (focusing on high speed and animal motion capture techniques) as well as the required filtering and state estimation algorithms in Chapter 2.

The methodology, detailed in Chapter 3, discusses how the system was developed and tested. The hardware and software design of the motion capture system was described in detail, including the sensors used and the layout of the sensor nodes, in Chapter 4.

To determine the drag coefficient of the fur covered cheetah tail, aerodynamic tests were performed and these were discussed in detail in Chapter 5. The filtering, state estimation algorithms and kinematics and kinetics models of the cheetah were developed and are described in Chapter 6 along with the results of their performance.

The discussion and recommendations section follows in Chapter 7. The dissertation then culminates with the conclusion and future work section in Chapter 8. The layout of the dissertation is shown in Figure 1.2.
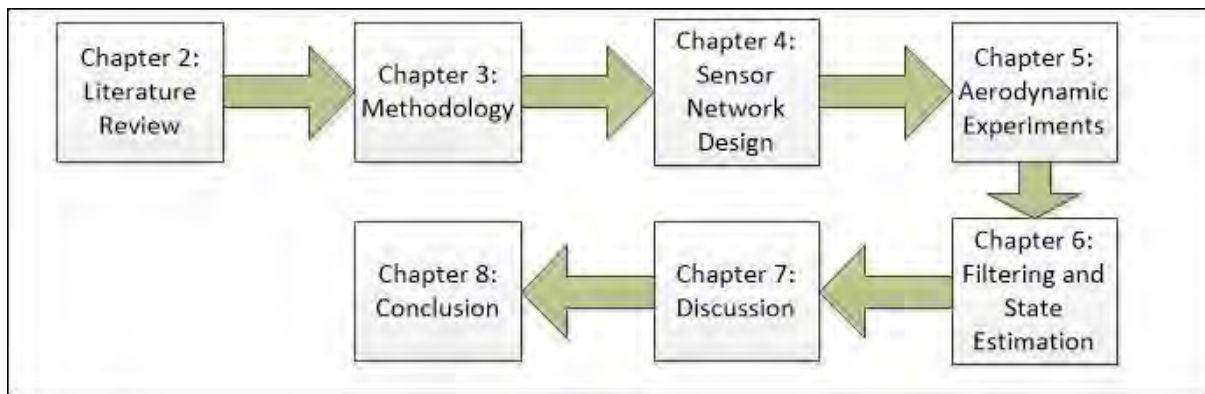
Figure 1.2: Layout of the dissertation.

Chapter 2

Literature Review and Theory Development

Evolution has played a vital role in solving many of nature's problems and difficulties such as flight [23], locomotion [24] and improving the strength of man made materials [25]. From studying nature we, as humans, can get a better understanding of how to solve problems in an engineering perspective. This has led to two dominant engineering fields namely bio-inspiration and bio-mimicry. Bio-mimicry involves directly copying things that are observed in nature [26], whereas bio-inspired engineering involves observing nature and applying the underlying principle, idea or theory [27].

There are many problems with copying nature. The main problem is that animals in nature have general features. When viewed from an engineering design perspective, compared to man made things which have exact specifications and are duplicates of each other, due to mass production. Animals adapt to their unique features, no two cheetah tails are the same weight or size, whereas man made devices cannot adapt or grow.

The latest animal of interest is the cheetah [17] [18] [28]. Studying and understanding an animal that is considered to be the fastest and most manoeuvrable terrestrial animal [1] has led to the design and development of many quadruped robotic platforms including the MIT Robo-Cheetah [11] [16] and the Boston Dynamics cheetah robot, Spot [10], Figure 2.1. The Boston Dynamics robot is currently the fastest running robot and is capable of reaching an impressive speed of 12 m/s, while the MIT Cheetah robot's main focus is energy efficient locomotion. Both of these platforms have focused on steady state locomotion and not aggressive transients that are performed by the cheetah.

There have been a few nature inspired robots that have focused on the high speed transients such as Dima [30], TailBot [31], TAYLRoACH [32] and FlipBot [33]. These platforms are mainly wheeled and focus on implementing a tail in order to improve their manoeuvrability. Their inspiration has been from a combination of the cheetah and the lizard as both these animals have been observed to use their tails to improve their
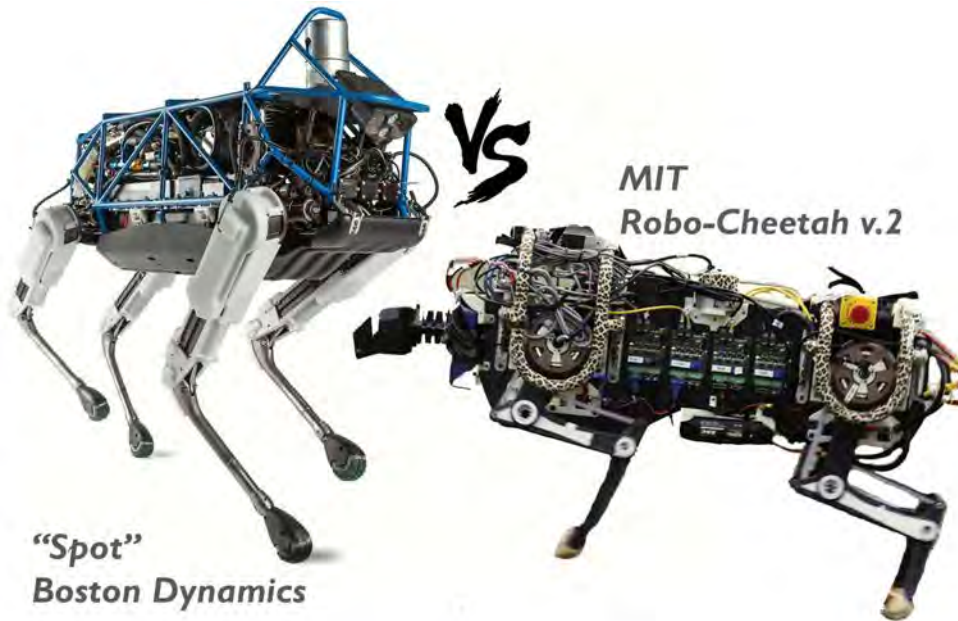
Figure 2.1: Two quadruped robotic platforms that inspired this research. On the left is the Boston Dynamics quadruped that is capable of running at 12 m/s and on the right is the MIT-Robo-Cheetah quadruped that focuses on energy efficient manoeuvrability. Image from [29].

manoeuvrability.

2.1 Manoeuvrability

Manoeuvrability is defined as the ability to change the body's velocity vector in a controlled manner [1]. This change can be due to angular velocity (turning), scaling of the vector (increase or decrease in size resulting in acceleration or braking respectively) or a combination of the two. Manoeuvrability is crucial for an animal's survival, in terms of capturing prey for food and escaping predators [20]. Wilson *et al.* states that an animal's success in hunting is attributed to the animals ability to rapidly accelerate and to mimic the evasive manoeuvres of its prey during a pursuit [20].

The cheetah is considered one of the most manoeuvrable and fastest land mammals [20]. During a hunt, a slower animal can escape by performing evasive manoeuvres and therefore the cheetah cannot only rely on its speed. The cheetah's incredible speed and ability to mimic manoeuvres of its prey has resulted in it having a high chase to kill ratio in terms of hunting [20].

Williams *et al.* has developed an over pitch model for quadruped animals [34]. The over pitch problem occurs when an animal (or other objects such as cars) tries to accelerate or brake too quickly. The front of the body will rise up when rapidly accelerating and vice versa when braking. This can cause the animal to topple over, become unstable or lose traction with the ground. All these possible effects can either injure the animal or hamper its maximum acceleration or braking abilities.

2.2 The Cheetah

It has been reported that the top speed of the cheetah is approximately 29 m/s [35] (Wilson et al. measured a velocity of 25 m/s with accelerations of up to 18 m/s^2 [20], see Figure 2.2). The reason why little research has been done on locomotion and modelling of transient dynamics (acceleration, braking and turning) is due to the complex dynamics involved [2], however a dynamic model of the cheetah will help in the development of future bio-inspired robotic platforms.

Figure 2.2: Image showing the cheetah running. The cheetah's top speed is approximately 29 m/s [35]. Image from [36].

Currently the only research that has focused on the cheetah (excluding cheetah inspired robots) are necropsy studies [37] along with video and behaviour analysis studies [20] [19]. The behaviour analysis studies involved placing sensors onto a cheetah and only two studies have been known to have done so [20] [19]. These studies used one sensor on the collar of the cheetah and the data was used to determine if the cheetah was hunting, sleeping, feeding or mating. The faster dynamics of the cheetah (the motion of the spine and tail) were not part of the scope of the research.

It is hypothesized that the tail of the cheetah is used to generate a reaction torque on the body [1] [3] [38], see Figure 2.3. This can aid in turn initiation and to counteract the over pitch problem [34] in terms of acceleration and deceleration. Turn initiation is vital as the cheetah has a large stride length and therefore being able to initiate a turn while still in the air is vital to hunting [37].

According to the available literature it has been assumed that the cheetah's tail is a heavy tail with a high moment of inertia [7] [8], however from necropsy data, it has been shown

Figure 2.3: Image showing the cheetah rapidly flicking its tail. It has been hypothesized that the tail of the cheetah is used to generate a reaction torque on the body [1] [3] [38]. Image from [39].

that the tail is in fact a light and relatively low inertia tail [37], see Chapter 5. The author of this thesis along with the supervisors (Dr. Amir Patel and Prof. Edward Boje) now hypothesize that aerodynamic effects play a large role in generating the reaction torques on the body of the cheetah. A paper confirming this hypothesis is currently under review [40].

To the author's knowledge there is no available kinematics or kinetics model of the cheetah that can be used to track the states of the tail and spine in order to analyse the motion of the cheetah. This research developed such a model that included the aerodynamic effects of the cheetah tail. The model is discussed in future chapters.

2.3 Modelling

Before the motion of the cheetah can be accurately tracked and analysed, the dynamics need to be modelled.

2.3.1 Kinematics and Kinetics

One way to capture the dynamics of the system is to generate a kinematics and kinetics model [41]. Kinematics is a branch of classical mechanics which describes the motion of a point, an object or a system of objects. The motion is described in terms of

position, velocity and acceleration. Forces, pressures and torques are not taken into account in kinematics. These form part of kinetics, which is deduced from kinematics by the introduction of mass [41].

There are two methods to generate a kinematics model of a system, the forward and inverse methods [42]. With forward kinematics the position of any joint can be calculated from the knowledge of the link connectivity, link length and joint rotation. The model is represented by a tree of joints, each with a local coordinate frame. The position of any point on the body can be found by traversing the tree structure.

With inverse kinematics you obtain the body posture by estimating the joint angles with the positions of a set of joints (including at least the end effectors) known [42]. The problem with this method is that if each link does not have a sensor, then it can lead to a situation where multiple solutions are possible as seen in Figure 2.4.
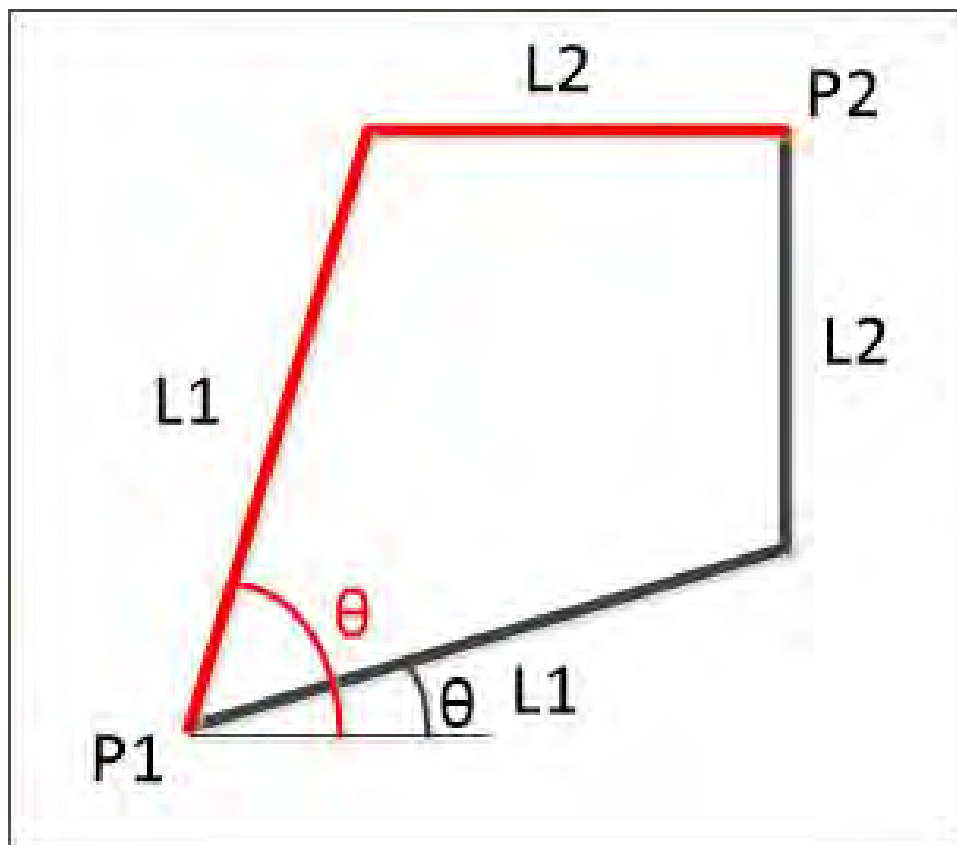
Figure 2.4: Two possible solutions, solution 1 is in black, solution 2 is in red, when using inverse kinematics. In inverse kinematics, the two end effectors, P_1 and P_2, are known and the joint angle, θ, is unknown, which results in two solutions.

The simplest way to model the system in order to generate the kinematics and kinetics equations is to assume that the system consists of a number of rigid beams connected together. This drastically simplifies the system and the equations of motion. Rigid beam simplification is a good first approximation of the system that can later be modelled as a flexible beam or in another, more complicated manner. Another simplification that is used is to assume the rigid beam is a point mass with its total mass located at the centre of mass (COM).

Rigid beams cannot undergo any deformation which includes bending and stretching (or shrinking) due to applied forces or torques [42]. There are a number of errors that are introduced when using the rigid beam assumption. These errors are only present if the system being modelled is very flexible and undergoes large bending or deformation. These errors can be alleviated by modelling the flexible structure with a number of rigid beams. For example instead of modelling a flexible beam with one rigid beam, it can be modelled with a number of rigid beams. The more rigid beams you use, the closer it will resemble the flexible beam as seen in Figure 2.5.
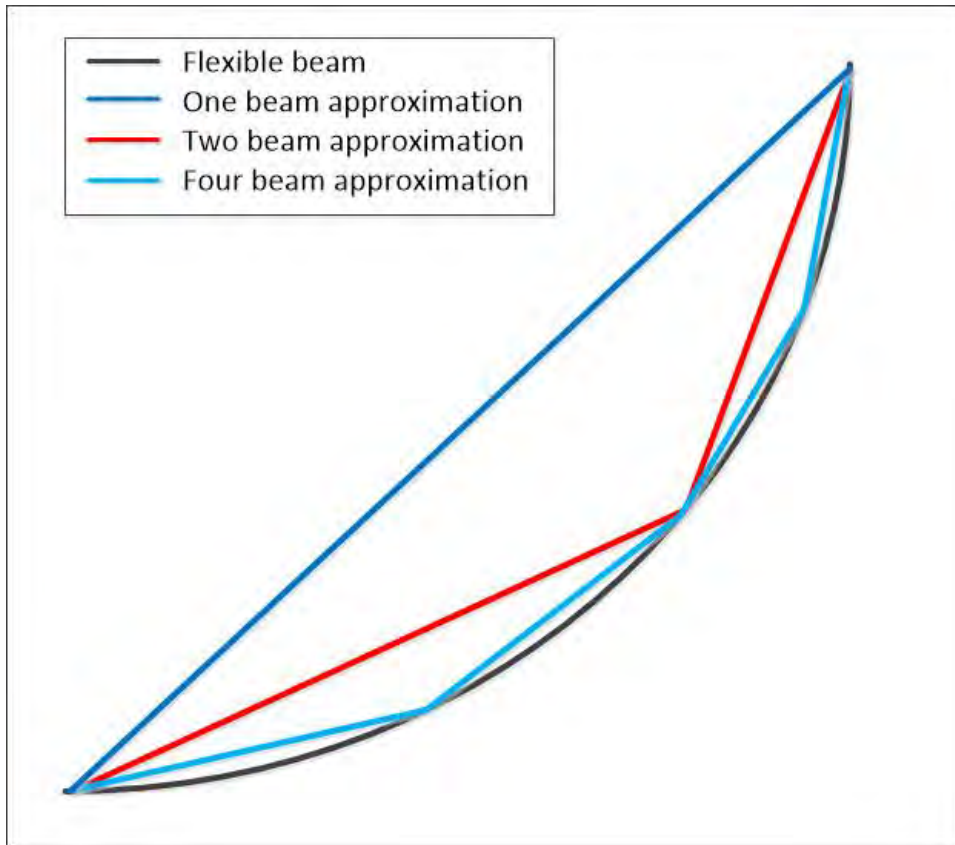
Figure 2.5: Modelling a flexible beam using multiple rigid beams. The more beams used, the closer it will resemble a flexible beam.

A multibody system consists of multiple interconnected rigid bodies. Multibody dynamics also extends to the flexible body case. Each of the bodies are capable of rotation and translation depending on the degrees of freedom of the joints connecting the bodies.

There are two common forms of multibody systems. These include serial chains and tree systems [43]. A serial chain system consists of multiple bodies connected in series to each other. A tree system consists of multiple branches with beams in parallel.

These systems can form open and closed chains. A serial chain system is considered an open chain system. A closed system occurs when the ends of the bodies either attach to form a loop or two ends are attached to the "ground". These structures can be seen in Figure 2.6.
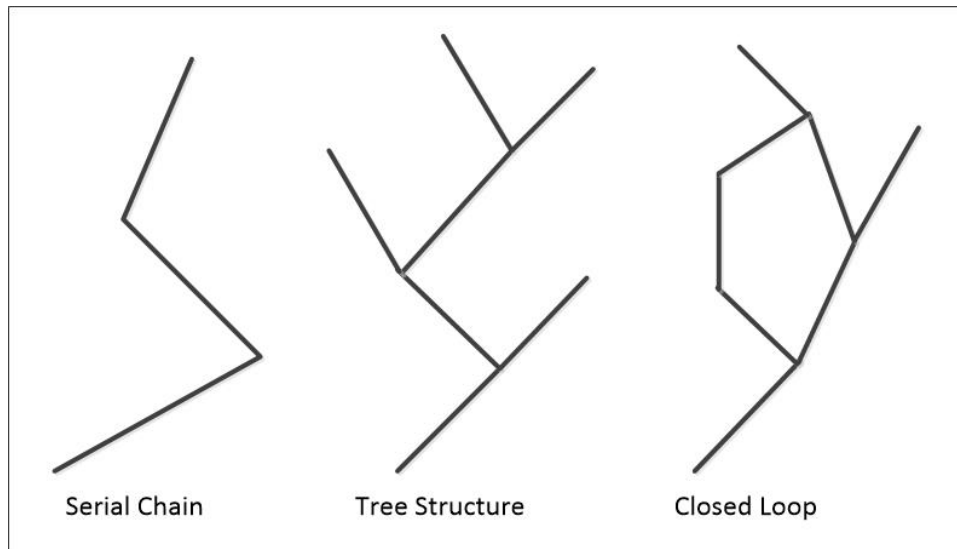
Figure 2.6: Image showing the difference between a serial chain, tree structure and a closed loop structure.

2.3.2 Lagrange Dynamics

Lagrange dynamics is used to model the dynamics of a system of objects (a multibody system). Lagrangian dynamics is a re-formulation of classical mechanics. It uses Newton's three laws of motion [44] and the principle of stationary action which is also known as the principle of least action [45]. Lagrange dynamics can be used on a system whether or not the system conserves energy or momentum. Often the system will be modelled as a system of rigid beams and point masses.

The principle of stationary action is a variational principle which is used to determine the equations of motion of a mechanical system [45]. This is done by assuming that the mechanical system follows a path where the average difference between the kinetic and potential energy is minimized over a specified time period. It is called "least" as the solution to the equations of motion involve finding the path that has least change between two points.

There are two methods to solve the Lagrange system to find the trajectory (motion) of the objects. The first method is to solve the constraints of the system as extra equations, often making use of Lagrange multipliers, and the second method is to incorporate the constraints directly through the choice of generalized coordinates [46] (known as the second kind of Lagrangian equations).

The use of generalized coordinates can drastically simplify the analysis of the system [46]. To model the system, the number of generalized coordinates used must accurately describe the system, and incorporate all the system constraints. The kinetic and potential energies must be written in terms of these generalized coordinates. The number of generalized coordinates represents the number of degrees of freedom of the system.

The Lagrange Equations

The system is described by the kinetic and potential energy of the bodies. The kinetic energy due to linear velocity for a system of particles (where the rigid beams are modelled as a point mass with the mass located at the COM of the beam) is

$$T_l = \frac{1}{2} \sum_{i=1}^{n} m_i \dot{\mathbf{r}}_i^T \dot{\mathbf{r}}_i \tag{2.1}$$

where n is the number of rigid bodies, m_i is the mass of the ith rigid body, and $\dot{\mathbf{r}}_i$ is the velocity of the COM of the ith rigid beam. The ^{T} represents the transpose of the matrix. The kinetic energy due to angular rotation for a system of rigid bodies is

$$T_r = \frac{1}{2} \sum_{i=1}^{n} \boldsymbol{\omega}_i^T \mathbf{I}_i \boldsymbol{\omega}_i \tag{2.2}$$

where ω_i is the angular rate of the ith rigid body and I_i is the moment of inertia of that beam. The total kinetic energy is in the form:

$$T = T_l + T_r \tag{2.3}$$

and the potential energy of the system is

$$V = \sum_{i=1}^{n} m_i \mathbf{g}^T \mathbf{p} \tag{2.4}$$

where $\mathbf{g}$ is the inertial gravity vector ($[0,0,9.81]^T$) and $\mathbf{p}$ is the position of the COM. Lagrange dynamics uses the Lagrangian function, which summarises the dynamics of the entire system and includes all the energies, generalized forces and the constraints of the system. The Lagrangian is defined as

$$L = T - V. \tag{2.5}$$

The solution for the system equations (solving the second kind of Lagrangian equations) is in the form: [46]

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{q}_i}) = \frac{\partial L}{\partial q_i} - \mathbf{Q}_i. \tag{2.6}$$

There is a common representation of Lagrange dynamics that will be used throughout this report which is the matrix representation of the above equations known as the manipulator equation [46]. The manipulator equation is in the following form:

$$\mathbf{M(q)\ddot{q}} + \mathbf{C(q,\dot{q})\dot{q}} + \mathbf{G(q)} = \mathbf{B\tau} + \mathbf{Q} + \mathbf{A}^T\boldsymbol{\lambda} \tag{2.7}$$

where $\mathbf{M(q)}$ is the mass matrix, $\mathbf{C(q,\dot{q})}$ contains the Coriolis and centrifugal forces, $\mathbf{G(q)}$ contains the potential energy of the system and $\mathbf{Q}$ represents the generalized forces and torques. The vector $\mathbf{q}$ contains the generalized coordinates that describe the system. The matrix $\mathbf{A}$ is the constraint Jacobian and $\boldsymbol{\lambda}$ is the constraint force. The vector $\boldsymbol{\tau}$ is the input torque and $\mathbf{B}$ is the input Jacobian.

In order to solve the above equation for the equations of motion of the system, Equation 2.7 is re-arranged and solved for $\mathbf{\ddot{q}}$ as seen in Equation 2.8. This is then integrated to solve for $\mathbf{\dot{q}}$ and integrated again for $\mathbf{q}$.

$$\mathbf{\ddot{q}} = \mathbf{M(q)}^{-1}(-\mathbf{C(q,\dot{q})\dot{q}} - \mathbf{G(q)} + \mathbf{Q} + \mathbf{B\tau} + \mathbf{A}^T\boldsymbol{\lambda}) \tag{2.8}$$

If there are any constraints then

$$\boldsymbol{\lambda} = -(\mathbf{A}\mathbf{M(q)}^{-1}\mathbf{A}^T)^{-1}(\mathbf{A}\mathbf{M(q)}^{-1}(\mathbf{B\tau} - \mathbf{C(q,\dot{q})\dot{q}} - \mathbf{G(q)} + \mathbf{Q}) + \mathbf{A}\mathbf{\ddot{q}}) \tag{2.9}$$

is used to solve for the constraint force [47].

D'Alemberts Principle and Generalized Forces

D'Alemberts principle deals with the generalized forces by introducing the concept of virtual work which is a result of the applied forces and inertial forces [48]. The virtual work of a particle is:

$$\delta W = \sum_{i=1}^{n}(\mathbf{F}_i - m_i\mathbf{\ddot{r}}_i).\delta\mathbf{r}_i = 0 \tag{2.10}$$

where n is the number of particles and $\delta\mathbf{r}_i$ is the virtual displacement of the ith particle. The term $\delta\mathbf{r}_i$ can be expanded to [48]:

$$\delta\mathbf{r}_i = \sum_{j=1}^{l} \frac{\partial \mathbf{r}_i}{\partial q_j}\delta q_j \qquad (2.11)$$

Therefore the virtual work thus can be re-written in terms of generalized forces as [48]:

$$\delta W = \sum_{j=1}^{l}\sum_{i=1}^{n}(\mathbf{F}_i - m_i\ddot{\mathbf{r}}_i).\frac{\partial \mathbf{r}_i}{\partial q_j}\delta q_j = 0 \qquad (2.12)$$

where there are l generalized coordinates. This is then converted into the generalized forces by dividing the virtual work by δq as follows:

$$Q_j = \frac{\delta W}{\delta q_i} = \sum_{i}^{n}\mathbf{F}_i.\frac{\partial \mathbf{r}_i}{\partial q_j} \qquad (2.13)$$

where n is the number of forces, $\mathbf{F}_i$ is the ith applied force, $\mathbf{r}_i$ is where the force acts and q_j is the jth generalized coordinate, where $j = 1 \ldots l$ for an l degree of freedom system.

2.3.3 Aerodynamics

The motion of the cheetah tail and spine needs to be captured and analysed. From necropsy data it was determined that the tail is in fact light and has a relatively small moment of inertia value [37]. The tail was noted to be very furry with especially long hair at the tip. Aerodynamic tests were performed on the cheetah fur to determine its effects on the torque generated by the tail. The experiments and results are discussed in detail in Chapter 5. The aerodynamic effects need to be included in the Lagrange model using D'Alemberts principle and generalized forces.

Aerodynamic Force

When a fluid flows past a cylinder, or a cylinder moves through a fluid, it experiences a normal overall force. The perpendicular force is due to the pressure difference between two sides of a body and the force is in the direction from high pressure to low pressure [49].

Figure 2.7 shows the force coefficient versus Reynolds number for a smooth cylinder. The coefficient varies with the velocity of the object through the medium (or the velocity of the medium past the object). In this case the medium is air. There is a general trend which decreases at high velocities and eventually breaks down [50]. The mean force coefficient of an infinite cylinder is less than a finite cylinder due to the effect of the air flowing over the ends of the cylinder which tends to increase the force [50]. This effect must be taken into account when modelling the aerodynamic effect of the tail as the tip of the tail is furrier then the rest and therefore a different coefficient will emerge for the tip component to the rest of the tail.
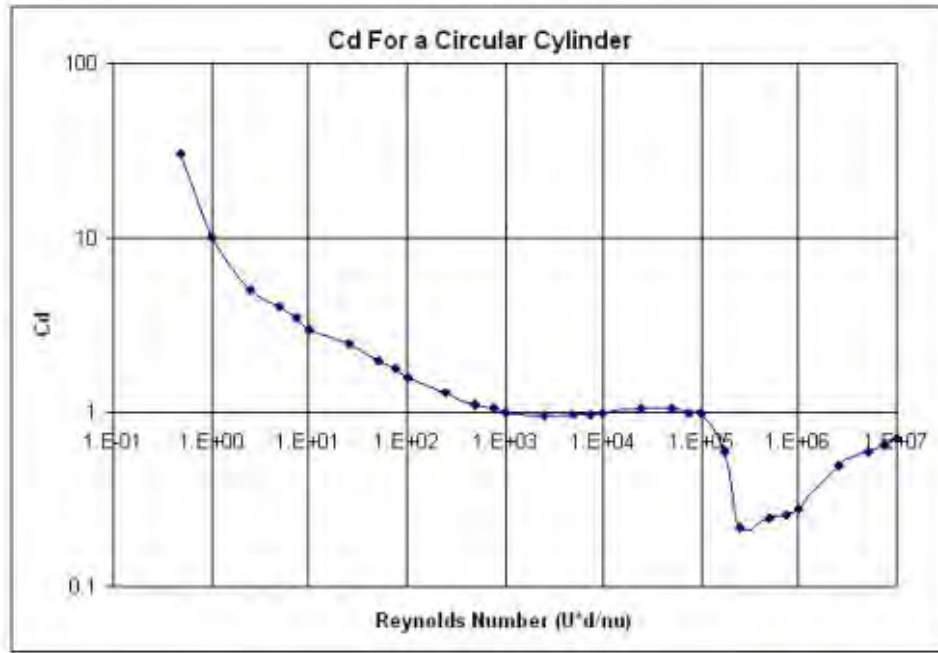
Figure 2.7: Image showing textbook data of a smooth circular cylinders force coefficient versus Reynold's number. Notice the breakdown at high Reynolds number. Image from [51].

The equation to calculate the aerodynamic force of an object 2.14 is

$$\mathbf{F}_D = \frac{1}{2}\rho\mathbf{v}^2 C_D A \tag{2.14}$$

where ρ is the density of the medium (air). The velocity of the body of medium is represented by $\mathbf{v}$, C_D is the force coefficient and A is the reference area. The reference area is not important as long as the same reference area is used throughout all experiments and comparisons [52]. The resulting coefficient will be in terms of this reference area. Thus when quoting an aerodynamic coefficient it is important to specify what area was used.

16

Reynolds Number

Reynolds numbers are used to compare two objects of different dimensions, at different velocities or in different fluids [49]. The equation to calculate the Reynolds number 2.15 is

$$R_e = \frac{\rho v l}{\mu} \tag{2.15}$$

where ρ is the density of the fluid, v is the characteristic velocity, l is the characteristic diameter and μ is the dynamic viscosity of the fluid.

Laminar flow occurs at low Reynolds numbers (low velocities) whereas turbulent flow occurs at higher Reynolds numbers [50]. Laminar flow is considered smooth and ordered flow while turbulent flow is chaotic in nature. The one problem with fluid flow is that a small change in surface texture and shape can drastically change the flow and resulting Reynolds numbers. This must be kept in mind when comparing two objects with different surface textures and dimensions.

2.4 Rotation Matrices

In order to use Lagrange dynamics to model the system, the attitude of each of the rigid bodies must be known in order to calculate the position and velocity of the COM of each body, which determines the kinetic and potential energy. A rotation matrix must be generated to rotate vectors and rates between the body and inertial frames.

There are a number of different ways to represent the attitude of a body. Attitude is defined as the roll, pitch and yaw of an object (its orientation in space), and specifies how the body frame is orientated with respect to the inertial reference frame. There are a number of methods of rotating the body frame vectors to the inertial frame and vice versa. The three most common ways are described below:

2.4.1 Euler Angles

Euler's rotational theorem states that any two independent orthonormal coordinate frames can be related by a sequence of rotations (no more than 3 rotations) about a coordinate axis, where no two successive rotations may be about the same axis [53]. Representing an order of rotations, this leads to twelve possible rotation combinations as shown in Table 2.1.

Table 2.1: Different Euler rotations

Symbol	Order
$\mathbf{R}_X(\phi)\ \mathbf{R}_Y(\theta)\ \mathbf{R}_Z(\psi)$	Roll, Pitch, Yaw
$\mathbf{R}_Y(\theta)\ \mathbf{R}_Z(\psi)\ \mathbf{R}_X(\phi)$	Pitch, Yaw, Roll
$\mathbf{R}_Z(\psi)\ \mathbf{R}_X(\phi)\ \mathbf{R}_Y(\theta)$	Yaw, Roll, Pitch
$\mathbf{R}_X(\phi)\ \mathbf{R}_Z(\psi)\ \mathbf{R}_Y(\theta)$	Roll, Yaw, Pitch
$\mathbf{R}_Y(\theta)\ \mathbf{R}_X(\phi)\ \mathbf{R}_Z(\psi)$	Pitch, Roll, Yaw
$\mathbf{R}_Z(\psi)\ \mathbf{R}_Y(\theta)\ \mathbf{R}_X(\phi)$	Yaw, Pitch, Roll
$\mathbf{R}_X(\phi)\ \mathbf{R}_Y(\theta)\ \mathbf{R}_X(\phi)$	Roll, Pitch, Roll
$\mathbf{R}_Y(\theta)\ \mathbf{R}_Z(\psi)\ \mathbf{R}_Y(\theta)$	Pitch, Yaw, Pitch
$\mathbf{R}_Z(\psi)\ \mathbf{R}_X(\phi)\ \mathbf{R}_Z(\psi)$	Yaw, Roll, Yaw
$\mathbf{R}_Y(\theta)\ \mathbf{R}_X(\phi)\ \mathbf{R}_Y(\theta)$	Pitch, Roll, Pitch
$\mathbf{R}_Z(\psi)\ \mathbf{R}_Y(\theta)\ \mathbf{R}_Z(\psi)$	Yaw, Pitch, Yaw
$\mathbf{R}_X(\phi)\ \mathbf{R}_Z(\psi)\ \mathbf{R}_X(\phi)$	Roll, Yaw, Roll

The roll rotation about the X axis is represented by $\mathbf{R}_x(\phi)$, $\mathbf{R}_y(\theta)$ is a pitch rotation about the Y axis and $\mathbf{R}_z(\psi)$ is a yaw rotation about the Z axis, see Figure 2.8. These three rotations represent how the body reference frame is orientated with respect to the inertial frame. Each rotation is based on the new reference frame that was obtained from the previous rotation [53]. The advantage of using Euler angles is that it is simple and intuitive. It also uses three parameters for three degrees of freedom and can be used directly.

The disadvantage of the Euler representation is that it suffers from a singularity known as gimbal lock. Gimbal lock is defined as the loss of one degree of freedom in a three degree of freedom system. This occurs when two axes are driven into a parallel configuration. This results in a loss of a degree of freedom. If a sensor with its three axes perpendicular (roll, pitch and yaw angles each zero) and the sensor pitches to 90 degrees, then the sensors yaw axis becomes parallel to the inertial roll axis and therefore changes in inertial yaw cannot be determined. The roll, pitch and yaw rotations are as follows:

$$\mathbf{R}_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \tag{2.16}$$

$$\mathbf{R}_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{2.17}$$

$$\mathbf{R}_z(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.18}$$
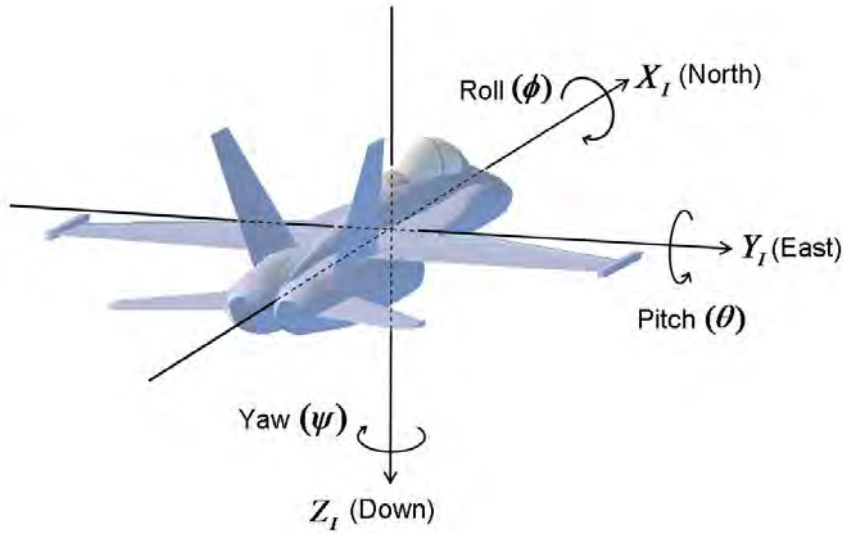
Figure 2.8: Image showing the roll, pitch and yaw axes. Image from [54].

2.4.2 DCM: Direction Cosine Matrix

The second common method for representing rotations is the DCM (direction cosine matrix) which requires 9 parameters (a 3 by 3 rotation matrix) to represent three degrees of freedom [55] and is formed using three Euler rotations. The DCM rotates a body frame vector to the inertial frame and the transpose rotates it back due to the matrix being orthonormal (orthogonal and normal, therefore $\mathbf{R}\mathbf{R}^T = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix). The roll-pitch-yaw Euler sequence DCM matrix can be expressed as:

$$\mathbf{R} = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \quad (2.19)$$

2.4.3 Quaternion

The quaternion is an extension of complex numbers. Quaternions are represented by a vector of 4 elements which represents three degrees of freedom [56]. This makes them less efficient than Euler angles but they do not suffer from a singularity. The quaternion consists of a real scalar and a three dimensional "imaginary" part [56].

In terms of mathematical operations the quaternion is more efficient as it only uses product and addition operators compared to Euler angles and the DCM which rely on trigonometric functions (which take longer to compute on a microprocessor).

The main disadvantage of the quaternion is that it is not intuitive and they cannot be analysed directly. The three dimensional imaginary part is a unit vector and the constant represents how much it rotates around the unit vector [56] as shown in Figure 2.9.
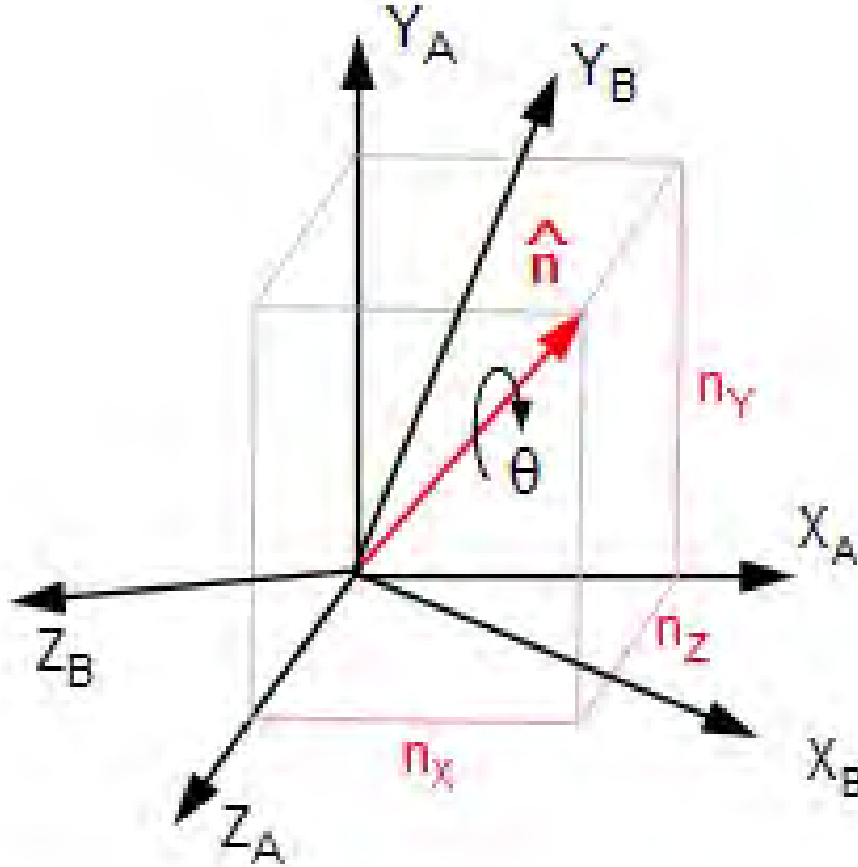
Figure 2.9: Image showing the quaternion unit vector, $\hat{\mathbf{n}}$, and θ which is a quaternion rotation about this vector. Image from [57].

Euler's rotational theorem is used with quaternions. The theorem states that the coordinate frames of two systems can be described by a single rotation about a single fixed axis [56]. The quaternion represents this rotation and axis.

2.4.4 Vector Rotation Comparison

Euler angles, the DCM and quaternion rotation matrices are compared in Table 2.2.

Euler angles were chosen as the singularity was not reached in the model, they are very intuitive and can be used directly. Due to the fact that all processing is done offline, the computational complexity of Euler angles was not an issue and neither was the memory problem.

Table 2.2: Comparison of the different ways to represent attitude

Category	Euler	DCM	Quaternion
Memory usage (elements)	3	9	4
Intuitive	very	average	not
Computational intensity	average	very	simple
Ease of use	simple	average	hard
Singularity	yes	no	no

2.4.5 Rotation of Angular Rates and Gyroscope Measurements

The above methods can only be used to rotate vectors between two reference frames. This works for the position, velocity and acceleration vectors however it does not work for angular rates.

The angular rates from the gyroscope represent a measurement in the body reference frame. In order to integrate the angular rates to calculate the attitude (roll, pitch and yaw angle) the rates need to be rotated into the inertial reference frame. Due to the fact that the angular rates are rates of change about an axis (roll, pitch and yaw axis) they require a different rotation matrix [58].

When the Euler sequence [roll, pitch, yaw] is applied, the Euler roll angle, which is applied first in this sequence, is not the final roll axis as the yaw and pitch rotations have moved the axis [58]. The first Euler angle (roll) undergoes two additional rotations, while the second Euler angle (pitch) undergoes only one additional rotation. The third Euler angle (yaw) undergoes no additional rotations [58]. The rotation matrix is in the form:

$$
\begin{aligned}
\omega_{body} &= \mathbf{R}(\phi)\mathbf{R}(\theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi}_{inertial} \end{bmatrix} + \mathbf{R}(\phi) \begin{bmatrix} 0 \\ \dot{\theta}_{inertial} \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\phi}_{inertial} \\ 0 \\ 0 \end{bmatrix} \\
\omega_{inertial} &= \begin{bmatrix} 0 \\ 0 \\ \dot{\psi}_{body} \end{bmatrix} + \mathbf{R}(\psi) \begin{bmatrix} 0 \\ \dot{\theta}_{body} \\ 0 \end{bmatrix} + \mathbf{R}(\psi)\mathbf{R}(\theta) \begin{bmatrix} \dot{\phi}_{body} \\ 0 \\ 0 \end{bmatrix}
\end{aligned}
\tag{2.20}
$$

2.5 MOCAP: Motion Capture

Once the system has been modelled, the motion needs to be tracked and analysed. This can be done using a MOCAP system. There are three general approaches to capturing the motion of an animal or system. These include a vision system, wearable sensors and floor sensors (for gait analysis). Each approach has its own advantages and disadvantages.

Motion tracking and motion capturing has become popular in many different fields, such

as stroke rehabilitation, gait analysis and motion monitoring [59]. Floor sensors have been used by biologists to determine an estimate of the power and reaction forces transferred between the body (or the leg) and the environment (commonly the ground) during gait analysis [60].

Motion capture has been used in many fields of robotics, virtual reality, health care (gait analysis), sign language interpolation and tele-presence (giving the effect that someone is present when they aren't through telerobotics [61]). It also has many advantages in the medical field such as determining the severity of a stroke, Parkinson's disease, multiple sclerosis and examining the degradation of the nervous system[62] (due to age, trauma or disease).

Motion capturing combines a number of sensors (gyroscopes, accelerometers, cameras, light sources, rotary encoders, etc.) to capture the motion of a system. Most available motion capturing systems are targeted at high accuracy motion capturing of relatively slow moving objects. These systems are expensive and difficult to calibrate, like the Vicon system [21], and generally have a small capture volume and are not suited to outdoor activities. The capture volume is described as the volume of space where the system can accurately locate the position of the point.

The MOCAP system that was developed for this research needed to be as small as possible as animals behave differently in the presence of humans and objects that are not found in their natural environment [63]. The cheetah will also be in the wild and therefore the MOCAP system cannot rely on emissions from sources and there must be no capture volume limitations.

2.5.1 Types of MOCAP Systems

There are a multitude of motion capture systems that operate on various principles. A few of the more common types are discussed in brief detail below:

1. Mechanical System [65]

   Mechanical motion capture systems involve attaching an exoskeleton to the system being tracked. The negative aspect of a mechanical system is that they are expensive and they restrict the motion of the object being tracked. These systems can only track one body at a time and require some form of global localization. Mechanical systems use rotary encoders or potentiometers on each of the joints.

2. Magnetic System [65]

   These systems require field from an artificial magnetic source. They use sensors that measure the low frequency magnetic field. Using triangulation one can determine the attitude of the sensor relative to the magnetic field and therefore track the body that the sensor is attached to.

3. Optical System [65]

   Optical motion capture systems rely on line of sight and are often influenced by the environment's ambient light. These systems are hard to use in real time due to the processing of each image from each camera. A popular (and expensive) system is the Vicon system that uses a number of cameras. These systems need to be calibrated and take a long time to set up, have small capture volumes and require emissions from a source. The larger the capture volume the more cameras that are required and therefore more processing is required. Due to the line of sight element, often the object being tracked will block the tracking marker; this reduces the accuracy of the system. Due to the cameras this system is not feasible for outdoor motion tracking.

4. Acoustic System [65]

   This system also requires emissions from a source. Audio transmitters are attached to the system and an external receiver measures the time it takes to receive a known signal. Using triangulation the position of the system can be determined.

5. Inertial and Magnetic System [65]

   Magnetometers, accelerometers and gyroscope sensors are becoming increasingly popular due to MEMs technology. These sensors are becoming smaller, more efficient, more accurate and cheaper. This type of motion capture system does not suffer from capture volume and line of sight limitations and does not require emissions from a source. They are, however, prone to drift and other errors.

6. Camera traps [66]

   A stationary camera that is activated using motion sensors captures footage of bypassing animals. The cameras are often camouflaged and therefore do not effect the animal or make it behave differently due to the presence of a foreign object. Due to the fact that the cameras are not on an animal they do not have to be small and can have large batteries.

2.5.2 List of MOCAP Requirements

Below is a list of the requirements for a MOCAP system to track the motion of the spine and tail of a cheetah:

1. The system must be non-invasive and must not irritate the cheetah.

2. The system must be marker-less and not rely on emissions from a source.

3. The sample rate of the system must be higher than or equal to 100 Hz, see Section 4.2.1.

4. The sensors must be fully wireless.

5. The system must be low cost.

6. The system must not suffer from a capture volume limitation.

2.5.3   Feasible MOCAP System

With the requirements in mind, an inertial and magnetic system was used for the motion capture system. These systems are the most feasible because they do not suffer from capture volume limitations and no emissions from a source are required. MEMs sensors are small and therefore the system was made to not interfere with the motion of the cheetah. The sensors were non-invasive and strapped onto the cheetah (the mechanical rig).

2.5.4   Available Inertial and Magnetic Motion Capture Systems

There are a number of different commercially available inertial and magnetic motion capture systems. In this section the advantages and disadvantages of each system will be presented.

1. The VN-100 system [67]

    This is a single IMU unit that has an AHRS (attitude heading reference system) filter. The sensor contains a 9 axis IMU and a pressure sensor. The size of the sensor is $36 \times 33 \times 9$ mm and has a weight of 13 grams. The output rate is 300 Hz. This sensor has a good output rate but is a single unit. If there was a network of sensors this rate would decrease. All the filtering algorithms, as well as the dynamic model of the system still need to be generated to link the sensors together.

2. The Xsens MVN system [68]

    This system consists of seventeen IMU sensors which are either wired together or communicate wirelessly. The wired system has an output rate of 240 Hz and the wireless system has an output rate of 60 Hz which is too slow to track a cheetah's movements, see Section 4.2.1. The sensors are also fairly bulky. The system consists of a lycra suit or large sensor straps. Each sensor implements an attitude EKF filter to determine the orientation of the appendage and there is no full body dynamic model. Xsens quote a 3 degree error in the measurement but do not mention if this is during static motions or during high speed dynamic motions. The system only has a range of 150 m for the wired system and 50 m for the wireless system.

3. The Shadow system [69]

    The Shadow system also consists of 17 IMU sensors and each sensor is $35 \times 25 \times 10$ mm. The sensors are connected via wires to a central controller that is $80 \times 40 \times 20$ mm and requires a battery of $93 \times 46 \times 25$ mm. This results in a total on body mass of 1.11 kg. The negative aspects of this system are that it is not wireless and has a large mass. The controller wirelessly transmits the data to the base station.

4. The Tech IMU CV4 system [70]

    This system has up to 16 IMU sensors and has an output rate of up to 200 Hz. The sensors are $36 \times 26 \times 11$ mm and weigh 9 grams. The network is a wired network

with wireless communication to the base station. This network would be perfect if the individual sensors were slightly smaller and wireless.

5. "Alpine system" [71]

   This system combines 13 IMU sensors, 2 GPS devices and 2 cameras to track the position of an Alpine Ski Racer. This is not feasible as cameras cannot be attached to a cheetah and the cost of this system with differential GPS, along with two cameras, is not considered low cost.

6. Opal sensors [72]

   The Opal sensor is a research grade wearable and wireless sensor network. It can have up to 24 sensor nodes and has a 16 hour battery life. The data rate of the sensors is variable from 20 Hz to 128 Hz. At US $ 2000 per sensor, the network is not considered a low cost solution.

7. Moca Sensor Network [73]

   This is "motion capture with accelerometers" (MOCA). It has four nodes and a acquisition board. The data output rate is 100 Hz and is a wired configuration which is not suitable to be placed onto a cheetah.

8. LynxNet [74]

   The LynxNet is a motion tracking system that has a single sensor (a tracking collar) on an animal and can track multiple animals at the same time. For this dissertation, multiple sensors are needed on a single animal and therefore this network is not feasible.

The above commercially available systems are not feasible to be placed onto a cheetah. The closest competition to the developed sensor network and the most feasible available system is the Xsens [68] system which is too large to be placed on the cheetah tail, see Figure 2.10. There is also a chance that the sensors on the tail will get damaged or bitten off and therefore the sensors must be low cost. A quote was obtained for the Xsens system and is shown in Appendix A.14. The sensor network itself costs € 11 500 for the wireless system and € 6 990 for the wired system. This is approximately[1] R 163 896 and R 99 620 respectively.

The above systems often come with their own software (such as MotionLab, MT Software Suite or MVN Studio). The problem with this is that one does not get access to the raw data to use with state estimation algorithms. Xsens systems use the MVN Studio software. There is a licence fee in order to use their software. A lifetime licence (with support for 3 years) costs €13 900 which is approximately[1] R198 101.

A custom built low cost MOCAP system along with the filtering and state estimation algorithms was developed to track the motion of the cheetah. The MOCAP system consisted of the wireless sensor network, filtering process and state estimation algorithms.

---

[1]Using an exchange rate of R14.25 to €1.

Figure 2.10: Image showing the Xsens sensor. Note that it is too large to be placed onto the cheetah tail without interfering with the cheetah. Image from [68].

2.6 Sensors: MEMs and GPS

The MOCAP system uses low cost MEMs sensors and a GPS module for global position localization. Sensors are used to sense a characteristic in its environment or to detect an event.

MEMs stand for Microelectromechanical systems and is the technology of very small devices [75]. The size of a MEMs device generally ranges from 20 micrometres to a millimetre. In its most general form MEMs technology can be defined as miniaturized mechanical and electro-mechanical elements such as micro sensors, micro actuators, micro-electronics and micro structures. This research focuses on the application of micro sensors in terms of MEMs.

Common attitude determination techniques rely on accelerometers, magnetometers and gyroscopes [76] [77] [78]. These sensors, along with their advantages and disadvantages are described in detail below:

1. Gyroscope

   A gyroscope measures the angular rate of the sensor. Low cost MEMs gyroscopes suffer from drift, bias and noise errors. These errors can be reduced by temperature calibrating the sensors. The advantages of the sensor is that by integrating the rate one can get the attitude of the sensor given the initial conditions of the sensor with no offset in the measurement (the sensor must be calibrated).

2. Magnetometer

A magnetometer measures the local magnetic field. It is generally used to determine the heading of the sensor. The Earth's magnetic field is considered locally constant. The disadvantages of the magnetometer is that it also picks up other magnetic fields from ferrous material, magnets and electric current. A magnetometer needs to be calibrated before it is used as it suffers from bias errors.

3. Accelerometer

   The accelerometer measures all forms of acceleration including the gravity vector and linear acceleration. When the sensor is not accelerating (stationary or constant velocity) it measures the gravity vector that can be used to determine the pitch and roll of the sensor. The accelerometer needs to be calibrated to reduce the bias and offset in the sensor. When the sensor is accelerating the gravity vector is no longer recoverable and therefore the pitch and roll of the sensor is lost.

It should be noted that all MEMs sensors suffer from noise, bias, offset, quantisation error and sampling frequency errors. These will be addressed in the forthcoming sections.

A GPS module will also be used in the sensor network. GPS is a space based system that uses satellites to accurately determine the position and velocity of a GPS sensor [79]. GPS measurements suffer from atmospheric (ionospheric and tropospheric) errors along with other errors due to multipath signals, satellite geometry and system clock errors [79]. These errors are calibrated out using embedded software, multiple frequencies and the correct antennas. A problem with GPS devices is that their update rate is slow compared to the update rate of MEMs sensors.

To improve the accuracy of the common GPS module from meters to centimetre accuracy, an RTK GPS module can be used. RTK stands for Real Time Kinematic and the GPS module relies on the phase of the carrier wave instead of the information contained on the carrier wave. This device also only relies on a single reference station (which can be an interpolated virtual station) in order to provide real time corrections with centimetre accuracy.

2.7 Prediction, Estimation and Sensor Fusion

Filtering a random process is called estimation which is a well-defined statistical technique [80]. There are two kinds of state estimation techniques, the linear case and the non-linear case.

Whenever the states of a system need to be predicted using noisy data from sensors, an estimation technique is used. All estimators that will be employed in this research are discrete estimators as they will be implemented in software.

Prediction and estimation requires mathematical equations that capture the dynamics of the system and a set of equations that relate the measured data to the dynamics (states)

of the system. The states of a system provide a complete representation of the system behaviour at any given instant of time. These equations will be generated using Lagrange dynamics described previously.

2.7.1   The Kalman Filter and the Extended Kalman Filter

The Kalman filter (KF) and the extended Kalman filter (EKF) are efficient mathematical approaches to estimating the state of a process which lead to the least square error of the state estimate being minimized [81]. The statistical description of the noise that corrupts the signals [82] (model and measurement noise) is taken into account in the algorithm.

The EKF is the non-linear version of the Kalman filter and comprises two parts. The first part is to predict the new (*a priori*) state estimate using the previous state estimate and the covariance matrix. This is known as the prediction stage. The second part (*a posteriori*) is to update the predicted state estimate using data from the sensors and the filter measurement equations. This is called the update stage. The filter acts as a form of sensor fusion where information from multiple sensors are fused to generate more meaningful data that the individual sensors cannot measure on their own [83].

The EKF works on the basis that the linearised dynamics approximately equals the true non-linear dynamics in the local region of operation [81]. The KF and the EKF work on the assumption that the process and measurement noises are Gaussian (a normal distribution in the time domain with an average time domain value of zero), white (zero mean and uniform power across the frequency band) noise [83].

The residual of the filter is the difference between the predicted measurements and the actual measurements. The Kalman gain matrix (K) is the blending factor that gives the optimal weight between the predicted states and the measured outputs.

The state covariance matrix in the filter is a measure of the degree to which two variables change or vary together. If the value is positive then they vary in the same direction.

The drawback of the EKF is that the filter can become unstable if the assumption of local linearity is violated (for example if the actual and estimated state vector are too far apart). Obtaining the Jacobian matrix (see Equation 2.30 and 2.31) is also a non-trivial task and can lead to implementation difficulties and often numeric methods are used to solve the filter. There are other possible filters such as the UKF [84] (unscented Kalman filter), particle filter [85] or H-Infinity filter [86]. The EKF was chosen due to the fact that it is an optimal filter under certain conditions such as locally linear, must have observable states and the process and measurement noise must be zero mean Gaussian white noise [83].

KF and EKF Equations [87]

Consider a non-linear system with the following transition and observation models:

$$\begin{aligned} \mathbf{x}_{k+1} &= f(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}(k) \\ \mathbf{z}_k &= \mathbf{H}\mathbf{x}_k + \mathbf{v}(k) \end{aligned} \tag{2.21}$$

where $\mathbf{x}$ is the state vector and $\mathbf{u}$ is the control vector. The zero mean white process noise is $\mathbf{w}(k)$ and $\mathbf{v}(k)$ is the zero mean white measurement noise. The process and measurement noise has the following covariance:

$$\begin{aligned} \mathbf{Q}(k) &= E[\mathbf{w}(k)\mathbf{w}(k)^T] \\ \mathbf{R}(k) &= E[\mathbf{v}(k)\mathbf{v}(k)^T] \end{aligned} \tag{2.22}$$

where $E[]$ is the expected value. The prediction equations are therefore:

$$\text{The predicted state estimate: } \hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}) \tag{2.23}$$

$$\text{The predicted covariance estimate: } \mathbf{P}_{k|k-1} = \mathbf{F}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1} \tag{2.24}$$

The update equations are:

$$\text{The innovation or measurement residual: } \tilde{\mathbf{y}}_k = \mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1}) \tag{2.25}$$

$$\text{The innovation or measurement covariance: } \mathbf{S}_k = \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k \tag{2.26}$$

$$\text{The Kalman gain: } \mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{S}_k^{-1} \tag{2.27}$$

$$\text{Updated state estimate: } \hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\tilde{\mathbf{y}}_k \tag{2.28}$$

$$\text{Updated covariance estimate: } \mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1} \tag{2.29}$$

For the non-linear case, the state transition and observation matrices are defined as follows:

$$\mathbf{F}_{k-1} = \frac{\partial f}{\partial \mathbf{x}}|_{\hat{\mathbf{x}}_{k|k-1}, \mathbf{u}_{k-1}} \tag{2.30}$$

$$\mathbf{H}_k = \frac{\partial h}{\partial \mathbf{x}}|_{\hat{\mathbf{x}}_{k|k-1}} \tag{2.31}$$

The **F** matrix is used to update the predicted covariance after the prediction stage and the **H** matrix is used to calculate the Kalman gain matrix (**K**) along with updating the measurement covariance. A flow diagram depicting the recursive nature of the Kalman filter is shown in Figure 2.11.
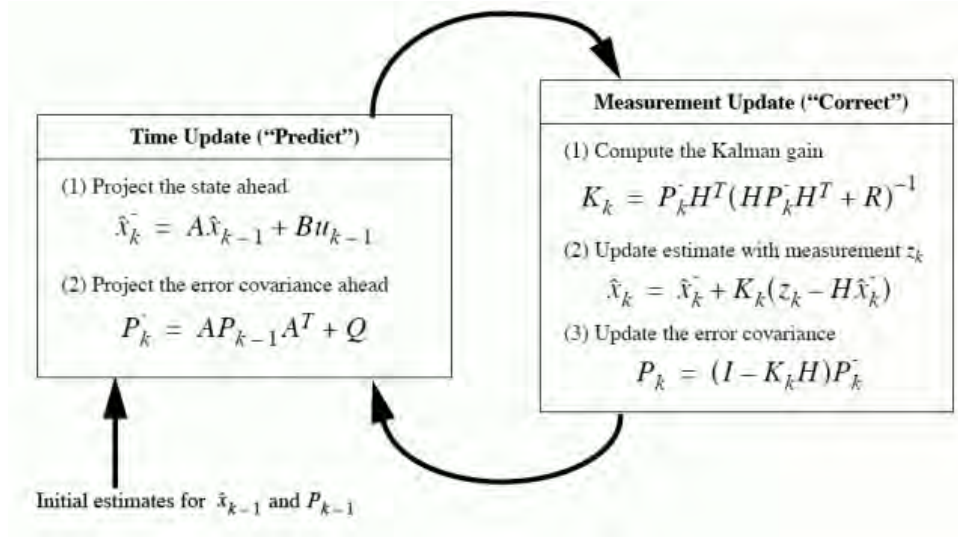
Figure 2.11: Image showing the recursive nature of the Kalman Filter. Note the two stages of the filter, time update and measurement update. Image from [88].

Numeric Integration

Due to the complexity of many multibody systems, the equations of motion are often calculated (numerically) during each time step instead of using a symbolic expression of the equations. If a numerical method is used then the equations need to be numerically integrated for the states of the system and numerically partially differentiated for the Jacobian matrix in the Kalman filter. For fixed step discrete time integration, the following formula is used to integrate a variable with respect to time:

$$\mathbf{q}_i = \mathbf{q}_{i-1} + \Delta t \dot{\mathbf{q}}_i \tag{2.32}$$

where i is the time step and Δt is the sample interval.

Numeric Differentiation

Partial numeric differentiation is used to calculate the **F** matrix in the Kalman Filter. The format for the **F** matrix is:

$$\mathbf{F} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \tag{2.33}$$

There are three methods for numeric differentiation [89] [90]. These are shown below:

1. Forward difference:

$$y' \approx \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \tag{2.34}$$

2. Backwards difference:

$$y' \approx \frac{y_i - y_{i-1}}{x_i - x_{i-1}} \tag{2.35}$$

3. Centralized difference:

$$y' \approx \frac{y_{i+1} - y_{i-1}}{2h} \tag{2.36}$$
$$where \ \ h = x_i - x_{i-1} = x_{i+1} - x_i$$

For partial numeric differentiation to generate the $\mathbf{F}$ matrix the above formula is modified (using the centralized method):

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \approx \mathbf{F}'_t(:, i) \approx \frac{f_t(\mathbf{x}_t + \epsilon \mathbf{e}_i) - f_t(\mathbf{x}_t - \epsilon \mathbf{e}_i)}{2\epsilon} \tag{2.37}$$

Where e is the basis vector with all entries equal to zero except for the ith element which is equal to one and is the variable that is being partially differentiated. x is the current states of the filter. ϵ is chosen to be a small number, but not too small to avoid numerical issues. The magnitude of ϵ is dependent on the rate of change that is expected.

2.7.2 TRIAD (Triaxial Attitude Determination) Algorithm

The TRIAD algorithm can either be used as an attitude filter on its own or it can be used as the update measurement for the Kalman filter. This drastically simplifies the KF update stage as the TRIAD algorithms output represents the states of the Kalman filter. The disadvantage of this algorithm is that linear accelerations corrupt the estimates as it relies on only reading the gravity vector, and not linear accelerations.

The TRIAD algorithm requires two vectors that are known in the inertial frame, for example the Earth's gravity and magnetic vector, and can be measured in the body frame [91] [92]. The inertial frame vectors must not be parallel or antiparallel. Assume

31

there are two vectors in the inertial frame, $\mathbf{R}_1$ and $\mathbf{R}_2$, and the corresponding two vectors measured in the body frame are $\mathbf{r}_1$ and $\mathbf{r}_2$. The body and inertial vectors are related by a rotation matrix as seen below:

$$\mathbf{R}_i = \mathbf{A}\mathbf{r}_i \tag{2.38}$$

where $\mathbf{A}$ is the rotation matrix that transforms the body vectors to the inertial frame and $i = 1, 2$. The rotation matrix has the following properties:

$$\mathbf{A}^T\mathbf{A} = \mathbf{I}$$
$$det(\mathbf{A}) = 1 \tag{2.39}$$

It must be noted that the matrix also transforms the cross product as follows [91] [92]:

$$\mathbf{R}_1 \times \mathbf{R}_2 = \mathbf{A}(\mathbf{r}_1 \times \mathbf{r}_2) \tag{2.40}$$

The TRIAD algorithm estimates the DCM by the following set of equations [93]:

$$[\mathbf{R}_1 \vdots \mathbf{R}_2 \vdots (\mathbf{R}_1 \times \mathbf{R}_2)] = \mathbf{A}[\mathbf{r}_1 \vdots \mathbf{r}_2 \vdots (\mathbf{r}_1 \times \mathbf{r}_2)] \tag{2.41}$$

where the vertical dots separate the column vectors, following the same notation as [91]. Therefore to solve for the TRIAD rotation matrix the following equation is used [91] [92]:

$$\mathbf{A} = [\mathbf{R}_1 \vdots \mathbf{R}_2 \vdots (\mathbf{R}_1 \times \mathbf{R}_2)][\mathbf{r}_1 \vdots \mathbf{r}_2 \vdots (\mathbf{r}_1 \times \mathbf{r}_2)]' \tag{2.42}$$

For a system that contains noise, the following equation is used to solve for the rotation matrix [91] [92]:

$$\mathbf{A} = \left[\frac{\mathbf{R}_1}{|\mathbf{R}_1|} \vdots \frac{\mathbf{R}_2}{|\mathbf{R}_2|} \vdots \frac{(\mathbf{R}_1 \times \mathbf{R}_2)}{|(\mathbf{R}_1 \times \mathbf{R}_2)|} \right] \left[\frac{\mathbf{r}_1}{|\mathbf{r}_1|} \vdots \frac{\mathbf{r}_2}{|\mathbf{r}_2|} \vdots \frac{(\mathbf{r}_1 \times \mathbf{r}_2)}{|(\mathbf{r}_1 \times \mathbf{r}_2)|} \right]' \tag{2.43}$$

where $\mathbf{A}$ represents [93]

$$\mathbf{A}(\phi, \theta, \psi) = \begin{bmatrix} \cos\theta\cos\psi & -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi \\ \cos\theta\sin\psi & \cos\phi\cos\psi + \cos\phi\sin\theta\sin\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix}$$

$$(2.44)$$

The disadvantage of the TRIAD algorithm is the fact that it relies on two locally constant vectors that can be measured in the body frame and known in the inertial frame. Typically the Earth's magnetic vector and gravity vector are used as they are known in the inertial reference frame. Accelerometers and magnetometers are used to measure the vectors in the body frame. A problem arises due to the fact that the sensors measure other parameters as well. Accelerometers measure the linear acceleration as well as the gravity vector. The magnetometer measures the Earth's magnetic vector as well as fields due to ferromagnetic material. This affects the TRIAD algorithm and makes it less accurate as it does not take these disturbances into account. As such, the TRIAD algorithm works better during slow movements with low linear accelerations.

2.8 Wireless Network

The wireless network is the heart of the required MOCAP system. Each sensor is a stand-alone device that wirelessly transmits its data to the base station. This enables the sensors to be small and independent of each other.

2.8.1 Wireless Communication Devices

There are a number of different wireless communication devices and protocols that can be used in the sensor network to communicate the data. Some common and feasible devices are compared in Table 2.3 and listed below.

1. Nordic module (NRF24L01) [94]

   The Nordic chip operates in the ISM band at 2.4 GHz and has a data transmission rate of 2 Mbps. It has a low current draw of 11.3 mA for transmission and 13.5 mA for receiving data. In standby mode the chip draws 22 μA. The device communicates with the microcontroller through SPI with a baud rate as high as 10 Mbps. It is capable of transmitting a 32 byte bi-directional packet.

2. Xbee/Zigbee [95]

   The Xbee and Zigbee protocols run on the 802.15.4 standard and are capable of 250 kbps (non continuous data transmission). Xbee packetizes the data and has a fairly inefficient protocol. The range for an Xbee is 30 m indoors and 90 m outdoors.

Zigbee has more freedom with respect to the protocol and has a range of 100 m outdoors.

3. WiFi [96]

There are a number of different wireless standards that are available and the benefits depend on the standard that has been employed. The theoretical data rate for WiFi is 1 Mbps to 6.75 Gbps. There are a few key factors to consider when selecting a WiFi protocol and module. These are data rate, range and power consumption.

The 802.11a standard runs at 5.3 GHz or 5.8 GHz with a range of 30 m. The data rate is 54 Mbps with a typical throughput of 18 to 22 Mbps.

The 802.11b operates at 2.4 GHz with a range of 30 m and a throughput of 6 Mbps.

The 802.11g operates at 2.4 GHz with a throughput of 22 Mbps.

WiFi is used for large data rates and is not idea for small sensor devices due to the power required (over 370 mA for transmitting data and over 110 mA to receive data). The modules are also generally larger than standard communication devices.

4. Bluetooth LE [97]

Bluetooth is generally the replacement for Xbee for larger data rates and quantities. Bluetooth LE or version 4 is a low energy protocol that boasts an efficient protocol of 66% efficiency (66 % of the transmitted bytes contain the required data, 44 % contains the address and error checking bytes). Bluetooth has a data rate of 24 Mbps. The different classes of the module effect the range of transmission (class 1 has a range of 100 m, class 2 has a range of 50 m and class 3 has a range of 10 m). Each class has a different power rating and therefore a different current consumption.

2.8.2   Comparison of Communication Protocols

A comparison of the communication protocols is given below:

Table 2.3: Comparison of communication protocols

Category	Xbee	Nordic	Bluetooth LE	WiFi
Range	90 m	100 m	100 m	30 m
Data Rate	250 kbps	2 Mbps	24 Mbps	54 Mbps
Tx current	340 mA	11.3 mA	15 mA	370 mA
Rx current	55 mA	12.3 mA	15 mA	110 mA
Price	R 385	R 68	R 239	R 207
Package size	24.38 × 27.61 mm	34 × 14 mm	15 × 15 mm	30.5 × 19.4 mm
Interface	UART	SPI	I2C, UART, SPI	I2C, UART, SPI

2.9 Summary of Literature

The required available literature has been reviewed in the above chapter. As can be seen, little research has focused on studying the cheetah's spine and tail dynamics which justifies this study. Additionally, the available MOCAP systems are not feasible to track the motion of wild cheetahs. The literature review covered all the relevant background required to design and build a low cost MOCAP system and to develop the state estimation and filtering algorithms.

Chapter 3

Methodology

There are two main problems that were addressed in this research. The first is the design (hardware and software) of the sensor network that was used to obtain data from the cheetah (the mechanical rig), and the second was the filtering and state estimation algorithms which used the data from the sensor network to track the motion of the cheetah's spine and tail (the mechanical rig) and required a dynamic model of the system.

This chapter explains the research, design and implementation process that was followed in this thesis to solve the above problems and to answer the questions stated in Section 1.4.

3.1   Modelling and Simulating the System

The dynamic system (the cheetah spine and tail) was modelled using Lagrange dynamics [44] [98] and the rigid beam assumption [42]. The tail and spine were each modelled with two rigid beams as seen in Figure 3.1.

From video analysis of a cheetah flicking its tail during rapid manoeuvres [13], it can be seen that the tail is kept fairly rigid, see Figure 3.2, and therefore the rigid beam assumption for the tail is a reasonable starting point. The spine of the cheetah has also been observed in video footage [14], and it has been noted that the spine flexes near the rear while the front section remains fairly rigid. Therefore by modelling the spine with two rigid beams, of different lengths, this effect was captured.

It is assumed that the two rigid beams that represent the tail can only pitch and yaw, whereas the spine rigid beams can roll, pitch and yaw. From necropsy data, of a cheetah that was euthanized due to renal failure [40], it was observed that the tail is powered by four large muscles (two muscles on the top, and two at the bottom, as seen in Figure 3.3. The two muscles responsible for pitch torques are the M. sarcocaudalis dorsalis

Figure 3.1: Image showing the rigid beams used to model the spine and tail. Two beams were used to model the spine and two for the tail. The white squares represent the sensors that are placed onto the cheetah. Image modified from [99].

Figure 3.2: Image showing a cheetah chasing a gazelle. Note the rigid tail. Image from [100].

medialis and lateralis muscles [38]. The two muscles responsible for yaw torques are the M. coccygeus medialis and lateralis muscles [38]) and therefore the tail is not capable of generating a roll motion. The spine is powered by a more complex array of muscles and can therefore roll, pitch and yaw.

The tail was noted to be extremely furry and therefore aerodynamic experiments were performed. The fur was attached to a rig and placed in a wind tunnel as shown in Figure 3.4. The force generated was recorded using a scale (Camry EK3252 electronic scale) and the drag coefficient of the fur was calculated. The scale measured the force required to keep the middle cylinder of the rig in its initial position. The drag effect is discussed in more detail in Chapter 5. The aerodynamic effects were taken into account when generating the Lagrange model of the tail.

Figure 3.3: Necropsy images of the tail. Note the four main tendons (the white streaks down the tail that connect the four main muscles to the tail vertebrae) on the left and the diameter of the tail and fur on the right. The tip becomes increasingly thin and the fur becomes increasingly thick. Images from [38].

The system was simulated in Matlab using a combination of analytical and numerical methods. The filtering and state estimation algorithms were designed using the kinematics and kinetics models that were developed using Lagrange dynamics. The algorithms are developed in Chapter 6.

3.2   Data Capture

The developed filters and state estimation algorithms require sensor data that can be related to the motion of the system. A sensor network was developed to capture the dynamics of the cheetah. The data from the sensor network was then used in the algorithms to estimate and predict the motion of the cheetah (or the mechanical rig).
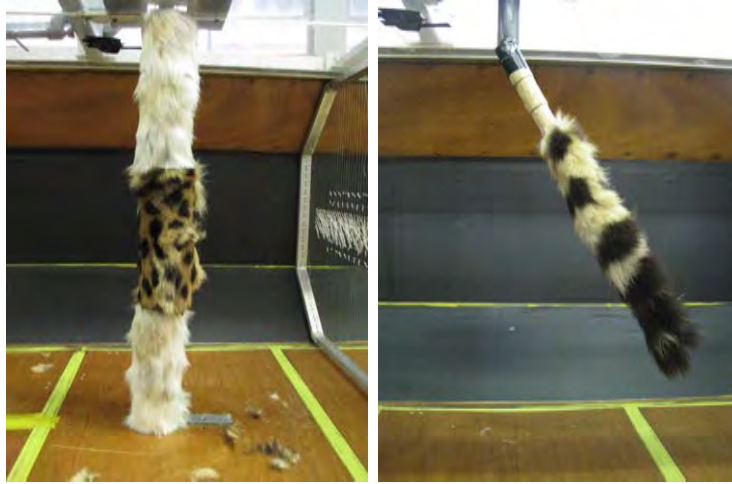
Figure 3.4: Cheetah fur placed on the two wind tunnel rigs. Fake fur (the white fur on the left) was used to remove the boundary layer effects. The reason for the fake fur was due to the scarcity of cheetah fur. The rig on the left is used for the cylinder test, the rig on the right is used for the tip test.

The sensor network is developed and discussed in detail in Chapter 4.

From video analysis [13] [14] [15], it was decided to have a sensor on the collar of the cheetah, one at the base of the tail, one in the middle of the tail and another at the tip of the tail as can be seen in Figure 3.1. From this configuration the 3D motion of the cheetah can be captured.

From the literature, motion sensing often makes use of multiple 9 axis IMUs [76] [77] [78] that contains a three axis accelerometer, a three axis magnetometer and a three axis gyroscope. Therefore each sensor will contain a 9 axis IMU and the collar will have an additional GPS module for absolute position determination. All sensors were wirelessly connected.

3.3   Testing the Algorithms and Models

In order to test the models, filters and state estimation algorithms that were generated, a mechanical rig to simulate a tail flick and spine motion was built and the sensor network was placed onto the rig. Multiple rigs were designed and built to test the different algorithms. The first rig simulated a tail flick for the pitch case, while the second rig simulated a tail flick as well as spine motion for the pitch case. The tail and spine of the rig were made of a hollow foam cylinder and a plastic pipe respectively. This allows for the spine and tail to slightly flex during a tail flick as observed when a cheetah flicks its tail.

The tail rig flicks the tail from a vertically down position, to a vertically up position and back down again. The cheetah generally flicks its tail just over 90 degrees and therefore if the sensor network can track a 180 degree flick it will be able to capture the motion

of the cheetah tail. The tail rig can be seen in Figure 3.5. A flow diagram depicting the operation of the mechanical rig is shown in Figure 3.6.

(a) Side view   (b) Front view
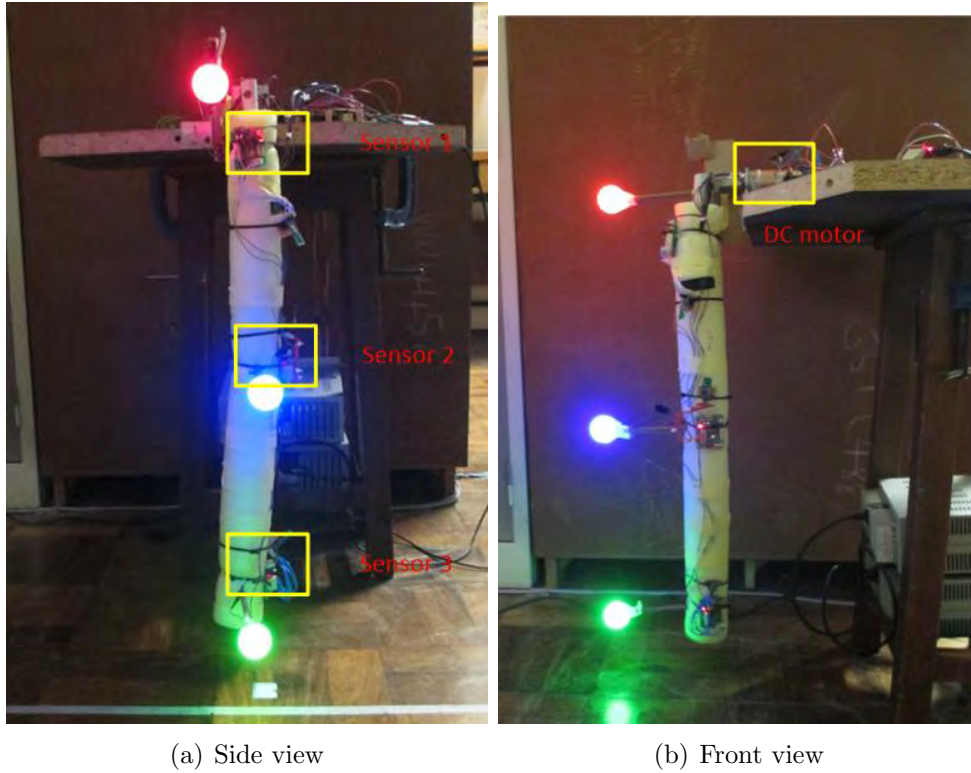
(c) The rig performing a 180 degree flick up and then back down again.

Figure 3.5: Images of the tail pitch rig. The LEDs are tracked by the camera system and represent the position of each sensor. The red LED comes on when the data is being logged by the sensor network. This is used to synchronize the camera data to the sensor data. Image c shows the tail rig during a flick.

The spine and tail rig flicks the tail from a horizontal position, to a vertical position and back again. The rig also allows passive bending of the spine. The rig can be seen in
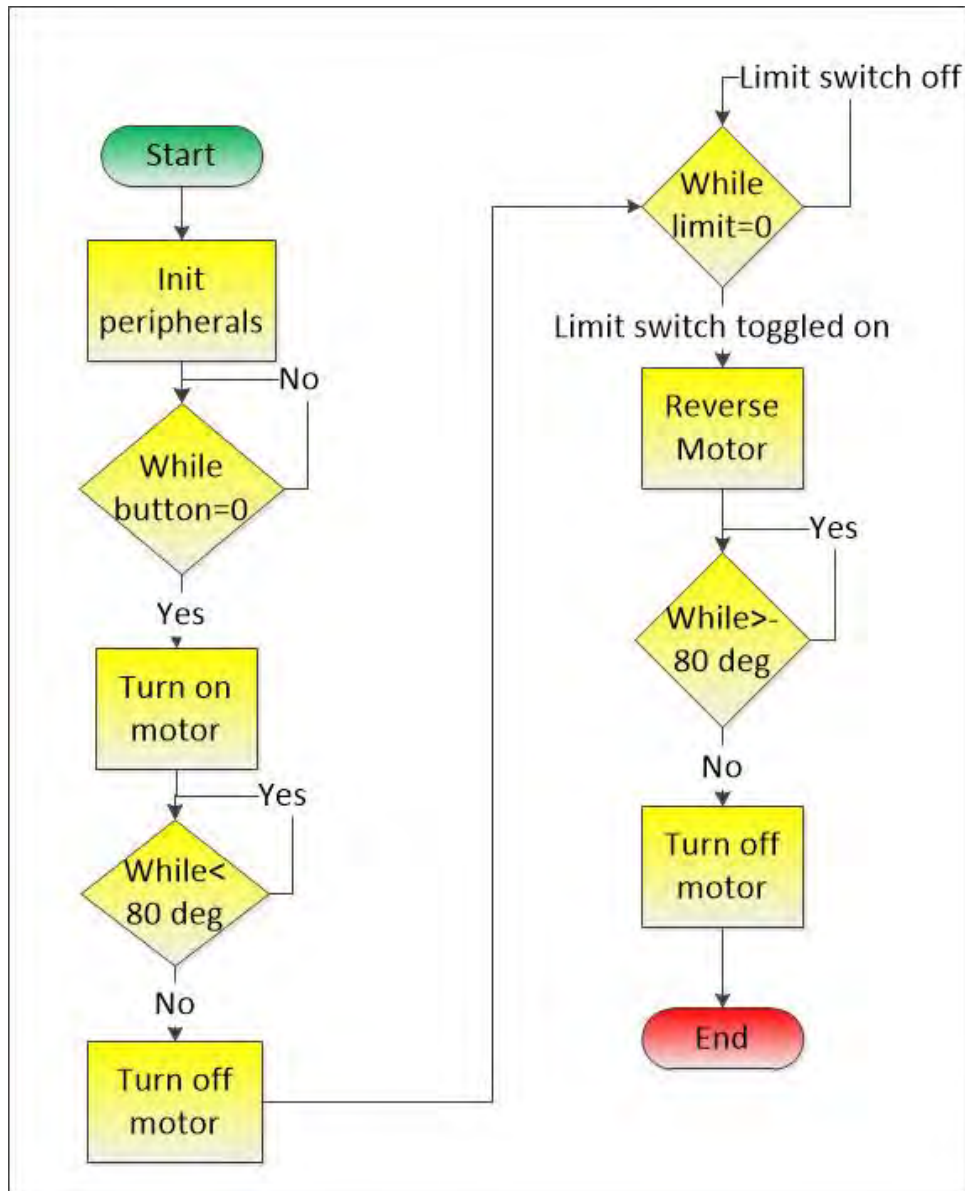
Figure 3.6: A flow diagram depicting how the mechanical rig operates. The motor is actuated after the user pushes a button. The motor pitches the tail until it reaches 80 degrees (horizontal is 0 degrees) and then the motor turns off and the momentum keeps the tail going until it hits the limit switches at the 90 degree mark. The direction of the motor is then reversed. The same occurs for the downward motion.

Figure 3.7.

A DC motor (Maxon DCX-35L, gear ratio of 26, motor torque constant of 13.7 mNm/A, 12 V motor with a no load speed of 8140 rpm) was used to simulate the spine bending and the tail flicking. A rotary encoder and mechanical end stops with limit switches were used to control the flick of the tail. An LED was used to synchronize the camera data to the logged sensor data. One of the LEDs on the rig turned on when the sensor network started data logging and turned off when it stopped logging. This was picked up by the camera system. A current sensor (ACS714) was attached to the motor and the current was logged. This reading was used to calculate the motor torque to compare it to the estimated torque from the state estimation algorithms. It was only used on the tail rig
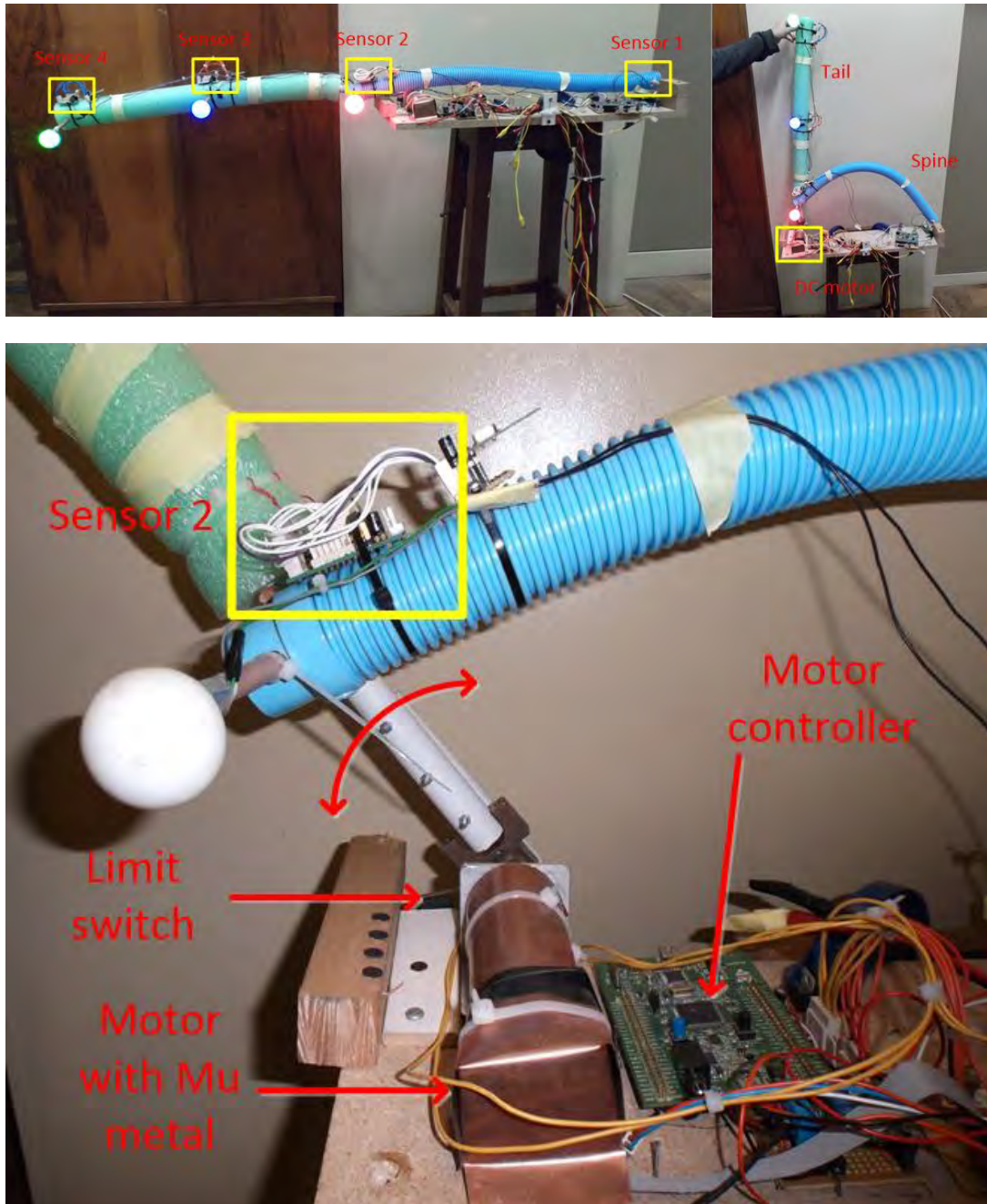
Figure 3.7: Spine and tail rig. The rig rotates the tail from the horizontal position seen on the left to the vertical position seen on the right. Note the LEDs/blobs that are tracked by the camera system and the bending of the spine. Image at the bottom shows how the motor flicks the tail and bends the spine. Note the DC motor covered in mu metal.

as the estimated torque state is a tail torque. The tail and spine rig used a single DC motor to actuate the spine and the tail and therefore the motor current cannot be used to compare to the estimated tail torque.

The peak angular velocity of the tail was calculated to be approximately 17 rad/s from video footage [101] of wild cheetahs. This value was used to select the speed of the motor for the tail flick. The tail on the mechanical rig flicked at 8 rad/s for the tail rig and 11 rad/s for the spine and tail rig. The models were developed for tame cheetahs that

flick their tails at a lower angular rate which was determined to be 11 rad/s from footage filmed at Cheetah Outreach in Somerset West, South Africa. Figure 3.8 shows a captive cheetah flicking its tail at 11 rad/s.

Figure 3.8: Images filmed at Cheetah Outreach. Note how bushy and straight the tail is. The frame rate was 120 fps and the images span over 17 frames. The cheetah can be seen performing a 90 degree flick in 0.1417 seconds.

A testing procedure was developed and followed for each test performed on the mechanical rig. The procedure is as follows:

1. Set up the mechanical rig in its initial position.

2. Turn on the camera system.

3. Turn on the mechanical rig.

4. Begin data logging on the camera system.

5. Begin data logging on the sensor network attached to the mechanical rig (synchronization light turns on).

6. Perform the tail flick.

7. Stop data logging on the sensor network (synchronization light turns off).

8. Stop data logging on the camera system.

9. Shut down all systems.

3.4 Verification of the Developed Algorithms

To verify the accuracy of the filters, the mechanical rig was placed under a MOCAP camera system. The camera system is capable of tracking the position of red, green and blue LEDs. A bright LED (red, green or blue) was placed inside a table tennis ball and attached to each sensor that was on the rig. Therefore the position generated by the developed algorithms can be compared to the position generated from the MOCAP camera system. The LEDs can be seen in Figure 3.5 and Figure 3.7. Four Playstation

Eye cameras were used in the camera MOCAP system. The system and all the camera code was developed by Mr. Yashren Reddi [22].

Two cost functions were developed to compare the performance of the developed algorithms. The first cost function was the mean square error between the estimated position and the position from the camera system (the Euclidean distance). The formula for the cost function is expressed as:

$$cost = \sqrt{\frac{\sum_{i=1}^{n} \left(\mathbf{p}_{Filter} - \mathbf{p}_{camera}\right)^2}{n}} \tag{3.1}$$

where n is the number of samples. The cost function is in metres. The second cost function involves taking the camera system position data and calculating what the pitch and yaw angles should be. The cost function is then the mean square error between the camera data angles and the estimated angles. The formula for the cost function is shown in Equation 3.2. This cost function does not work for the 3D tail and spine algorithms as the roll angle cannot be determined with the camera system.

$$cost = \sqrt{\frac{\sum_{i=1}^{n} \left(\theta_{Filter} - \theta_{camera}\right)^2}{n}} \times \frac{180}{\pi} \tag{3.2}$$

where n is the number of samples. The cost function is in degrees.

3.5 Limitations

The mechanical rig that was used to simulate the motion of the spine and tail used a large DC motor that saturated the magnetometers that were part of the sensor network. The magnetometer data was simulated from camera data in Simulink. This however is not a crucial problem as there will be no magnetic interference when the sensor network is deployed on wild animals.

Mu metal [102] was used in attempt to shield the magnetic field from the motor. However, Mu metal only blocks low frequency magnetic fields (such as permanent stationary magnets) and does not block the magnetic field generated by the current that is flowing through the motor (the current/voltage was pulsed at 8 kHz to control the motor). The effects of the motor magnetic field on the magnetometer can be seen in Figure 3.9.

Additionally, the camera system [22] used to validate the results operated at 60 Hz compared to the 100 Hz of the developed sensor network. Therefore the data from the camera system was interpolated to 100 Hz, using a Simulink rate transition block, to enable comparison. The rate transition block implements a linear interpolation algorithm.
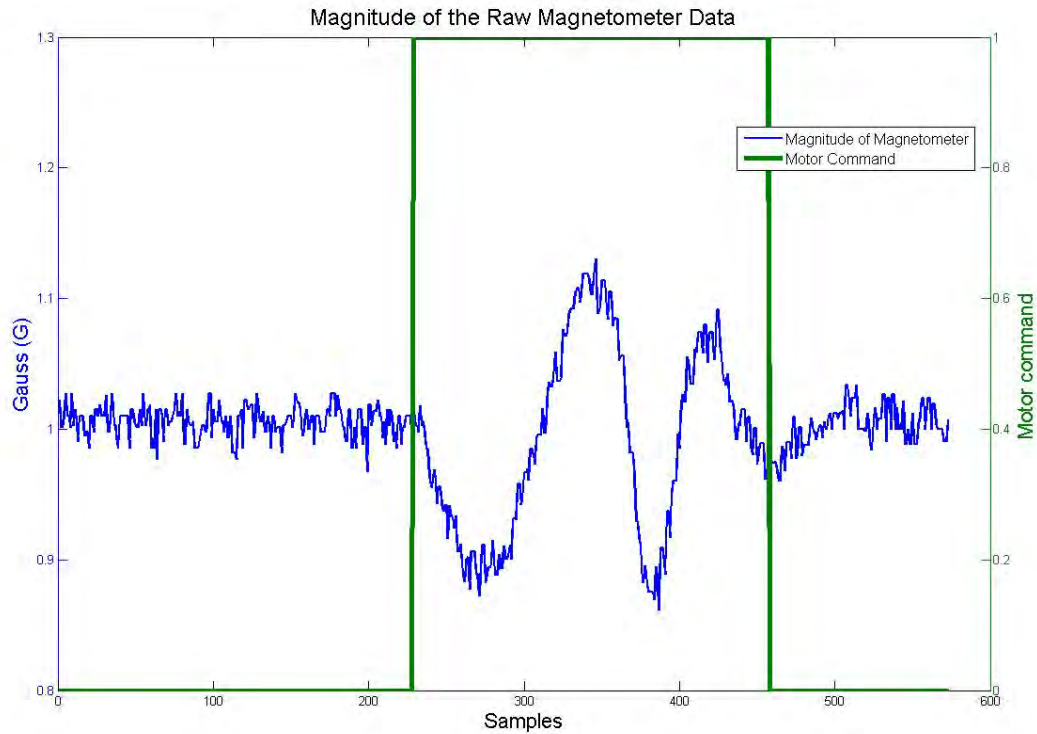
Figure 3.9: Image showing the magnetometer error. Data is of the magnitude of the magnetometer data during a tail flick. The magnitude should remain constant. Note how the magnetometer magnitude changes when the motor turns on. The green line represents the state of the motor. When it is 1, the motor is on, when it is zero, the motor is off.

There is also noise present in the camera system data that was not quantified and taken into account. The camera system data was assumed accurate and was used as the absolute position of the sensors. The sensor network system can thus, at best, only be as accurate as the camera system.

The camera system was designed to track a triangle of known dimensions, with a bright light (one red, one blue and one green) at each of the corners of the triangle. This configuration allows for extremely accurate position and attitude tracking (millimetre precision) [22]. The precision of the system is lost when a single light source is used.

Chapter 4

MOCAP Sensor Network

In recent years MEMs sensors have become increasingly popular [75] due to their ever decreasing cost and use in many every day devices such as cell phones, remote controls (for games and many flying toys), tablets and smart watches to name but a few. This has driven the prices of these devices down. MEMs sensors are becoming smaller, cheaper, more accurate, efficient and easily available. This makes them the ideal types of sensors for a low cost, high speed and small motion capture system.

The purpose of the sensor network was to be placed onto a cheetah or the mechanical test rig and to capture the dynamics of the system. The data was then used with the generated algorithms to track the motion and states of the cheetah tail and spine (or the mechanical rig) which was then further analysed. The sensor network is flexible in that by changing the models, the sensor network can be used to track a different configuration of bodies.

There are a number of available MOCAP systems such as the Xsens [68] system, but as discussed in the literature review, Section 2.5, they are not feasible for this research for several reasons. Therefore a custom MOCAP system was designed and built.

In this chapter the low cost, high speed and flexible motion capture sensor network (hardware and software) that was developed is explained and discussed in detail.

4.1 Hardware Design

4.1.1 The Sensors

The sensor network consists of a distributed array of MEMs sensors. The network consists of one collar sensor, three tail sensors and a base station. A tail sensor is placed at the

base, the middle and at the tip of the tail. This configuration was shown to adequately capture the spine and tail states.

The tail sensors consist of an INEMO-M1 and a Nordic NRF24L01 RF transceiver for communication. The collar consists of an STM32F4, an INEMO-M1, a Nordic NRF24L01 RF transceiver, a WiFi module (not implemented, but used to communicate to the PC, a wired connection was used during testing) and a GPS module (not implemented). These will be implemented at a later stage.

INEMO-M1

Four INEMO-M1, Figure 4.1, SOB (system-on-board) chips were available and therefore were used in the sensor network. The advantages of the chip were that it was small (13 mm × 13 mm × 2 mm) and came with all the required sensors on one board. The only additional component needed was an RF device to transmit the data. The cost of the sensor, see Section 4.3, was outweighed by the ease of development.
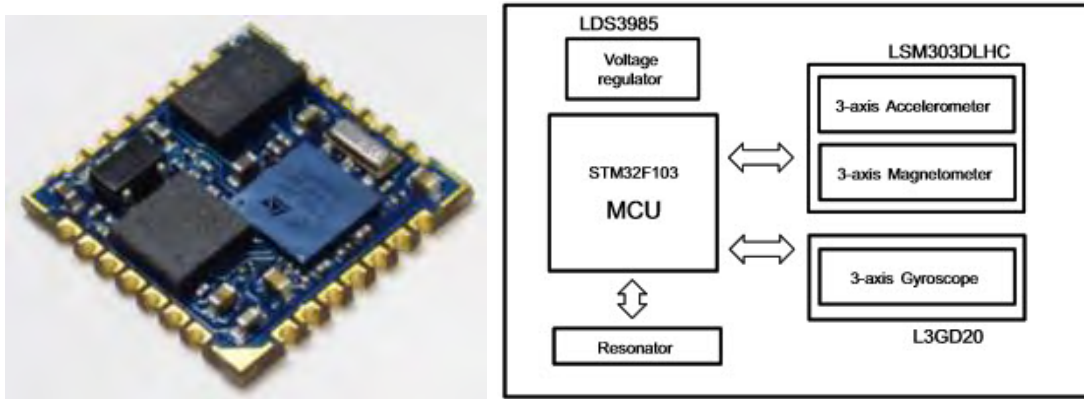
Figure 4.1: The INEMO chip is shown on the left. Image from [103]. The functional block diagram of the chip is shown on the right. Image from [104].

The INEMO-M1 has a 9 axis IMU, a temperature sensor, a 32 bit ARM Cortex M3 micro controller and a voltage regulator (LDS3985M33R, which is an ultra-low drop and low noise, 300 mA regulator) on board. The 9 axis IMU consists of a 6-axis digital e-compass module (LSM303DLHC that contains a 3 axis magnetometer and a 3 axis accelerometer) and a 3 axis digital gyroscope (L3GD20). The INEMO-M1 draws 46 mA when active (wile reading IMU data with all 9 axes active) and 0.22 mA when in sleep mode.

The e-compass is capable of ± 2g, ± 4g and ± 8g settings for the accelerometer and ± 1.3 gauss to ± 8.1 gauss settings for the magnetometer. The gyroscope can be set to ± 250 degrees per second (dps), ± 500 dps or ± 2000 dps. The micro controller allows for on board processing and data handling before the data is transmitted.

The mechanical test rig that the sensors were attached to, rotates the tail at 11 radians per second as discussed in Section 3.3. Therefore the gyroscope was set to ± 2000 degrees

per second as 11 radians per second equals 630 degrees per second. At this angular rate, a sensor at the tip of the tail will experience about 70 m/s^2 normal acceleration ($a_n = \omega^2 r$) and therefore the accelerometers were set to the $\pm$ 8g setting.

Nordic NRF24L01 RF module

From a comparison of the available wireless modules in the literature review, Section 2.8, the Nordic NRF24L01 RF module was considered the most feasible device to be used due to its power consumption, baud rate, low cost and package size.

This module operates in the 2.4 GHz ISM (industrial, scientific and medical) band and was used to communicate and transmit data between the tail sensors and the collar. The module is capable of an over the air baud rate of 2 Mbps and the INEMO-M1 interfaces to the module via SPI with a maximum baud rate of 10 Mbps. The RF module comes with a built in chip antenna which made the circuitry design for the tail sensor much simpler.

The size of the module is 34 mm by 14 mm. The module performs on-board CRC (cyclic redundancy check, a form of error checking) computation along with data packetization, handles acknowledgements and can be programmed to re-transmit the data if it gets corrupted while transmitting. The module can be synchronized to receive data from up to six other devices, each with its unique data pipe and address.

Each packet can be up to 32 bytes long, and can be attached to an acknowledgement byte for bi-directional communication. The data packet is appended to a larger packet that is transmitted by the module. The packet consists of a preamble byte, a five byte address, the payload (up to 32 bytes) and a two byte CRC value. The packet is shown in Figure 4.2.

Preamble 1 byte	Address 3-5 bytes	Payload 0-32 bytes	CRC 1-2 bytes
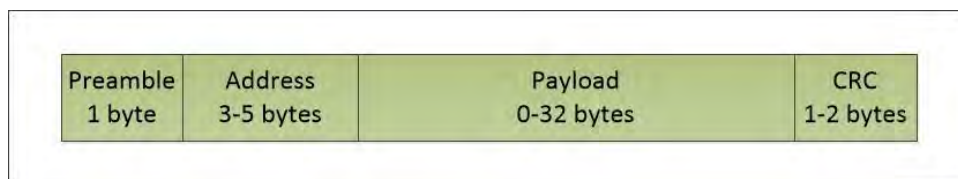
Figure 4.2: Image showing the packet that the Nordic NRF24L01 RF chip transmits.

During transmission the chip draws 11.3 mA of current and while it is receiving data it draws 13.5 mA of current. The devices standby current is 22 μA. It has a range of 100 m on its maximum power setting. The module is shown in Figure 4.3.

The Battery

Due to the low power consumption of the sensor module, a single cell Lipo battery (1S, 20C) will be used to power the device. The battery has the correct voltage range for the
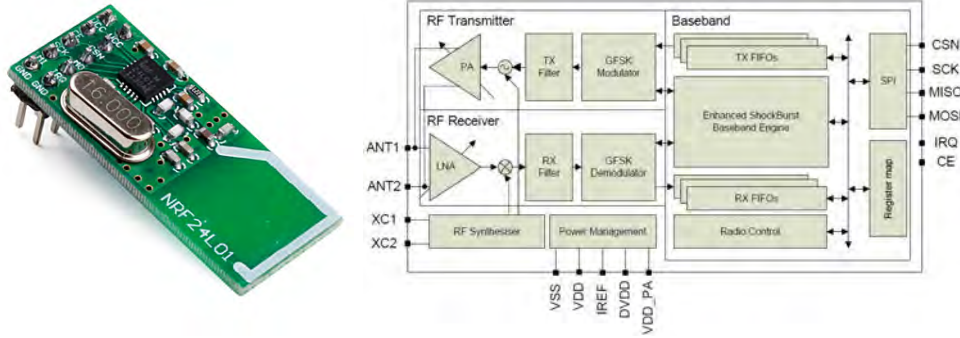
Figure 4.3: Image on the left is of the Nordic chip that was used. The chip has a chip antenna and is 34 mm by 14 mm. Image from [105]. Image on the right shows the functional block diagram of the chip. Image from [106].

INEMO-M1 and therefore an additional regulator was not required. The INEMO accepts 2.4 V to 3.7 V and the Nordic NRF24L01 chip accepts 3.3 V as the input voltage.

The battery is rated at 3.7 V and is a 138 mAh battery. The size of the battery is 32 × 20 × 7 mm and it weighs 8 grams. The battery will allow the sensor to last for over two hours of constant data transmission (sensor only transmits for a maximum of 1 ms for every 10 ms and takes 2 ms to read the data from the MEMs sensors. The rest of the time it is in low power mode, and therefore will last longer).

The battery would be adequate for testing on tame cheetahs at Cheetah Outreach, but not for wild cheetahs. For the wild tests some form of energy harvesting technique will be required as the sensors will need to operate for months at a time. This could be implemented using solar panels on the collar as done in [20]. This is big and bulky and will not work for the tail sensors. The tail sensors will require a high energy density battery in combination with clever software and electronics to maximise its efficiency and operating time.

4.1.2   PCB Design

A development board for the INEMO-M1 was designed in Altium Designer and developed in order to test the sensor code, MEMs sensors and RF module. The board is shown in Figure 4.4a. A second revision of the board was developed once all the components were finalized and the code was working. This board is smaller (31.6 × 17.6 mm) and has all the peripherals on one board. This board is shown in Figure 4.4b and the block diagram of the board is shown in Figure 4.5b.

A prototype collar sensor was also developed. This was developed on Vero board as the WiFi and GPS modules were not implemented. The collar consisted of an INEMO-M1, NRF24L01 chip, STM32-F4 microprocessor and a wired connection to the PC base station. The block diagram of the collar sensor is shown in Figure 4.5a.
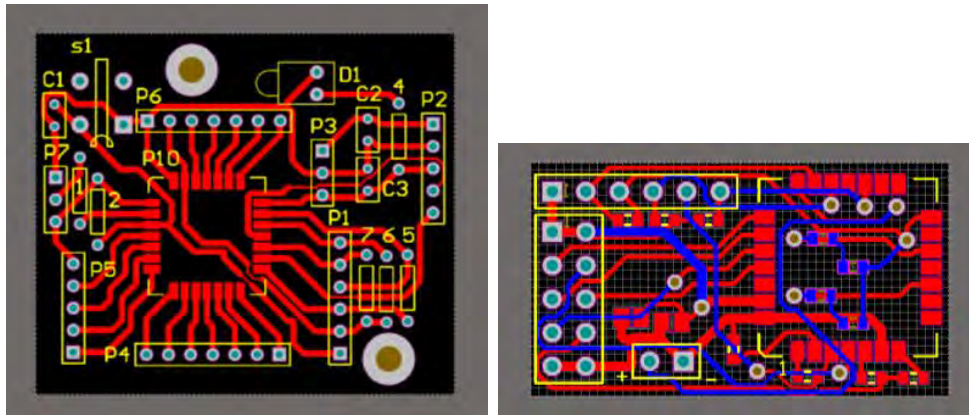
(a) Development board 1 track layout.

(b) Development board 2 track layout.
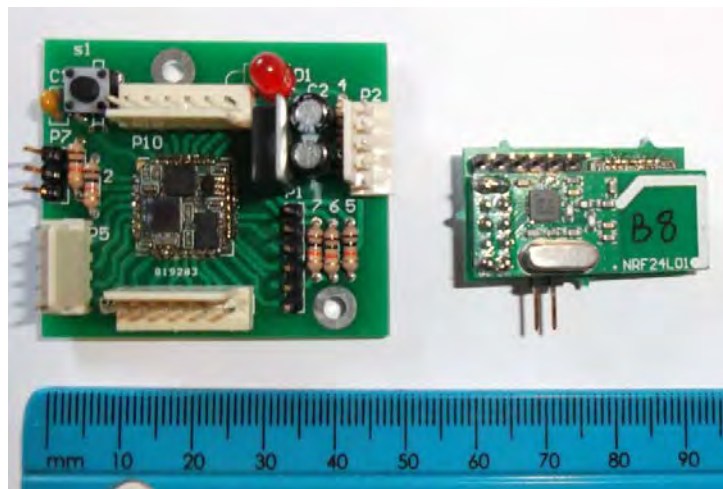
(c) The two development boards. The board on the right has the NRF24L01 chip as a daughter board.

Figure 4.4: INEMO development boards.

4.1.3 Sensor Network Layout

The sensor network consists of three different components. There is the base station, the collar sensor and the three tail sensors. The base station can only communicate with the collar sensor and receives all the sensor data from the collar. The base station consists of a PC that runs an application that was developed to check that the communication between the collar and the tail sensors is functioning and all the received data is displayed onto the computer screen in real time.

The collar sensor is the master and communicates with the three tail sensors via the NRF24L01 RF chips. The tail sensors are the slave sensors. The layout of the system is in the form of a common star topology as seen in Figure 4.6. Each tail sensor communicates wirelessly with the master which in turn communicates wirelessly with the base station via WiFi. The wireless base station was not implemented in this project, instead a wired connection was used.
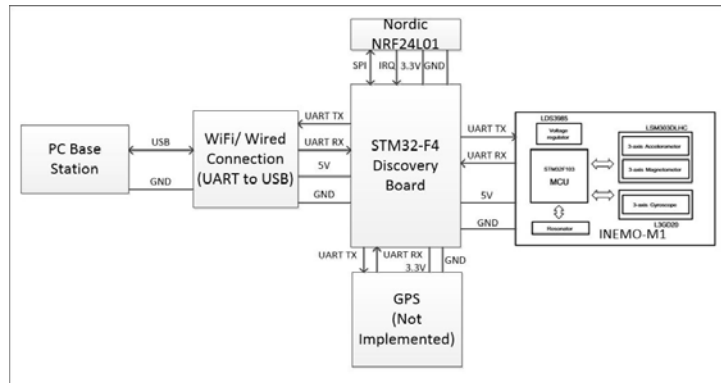
(a) Collar sensor block diagram.
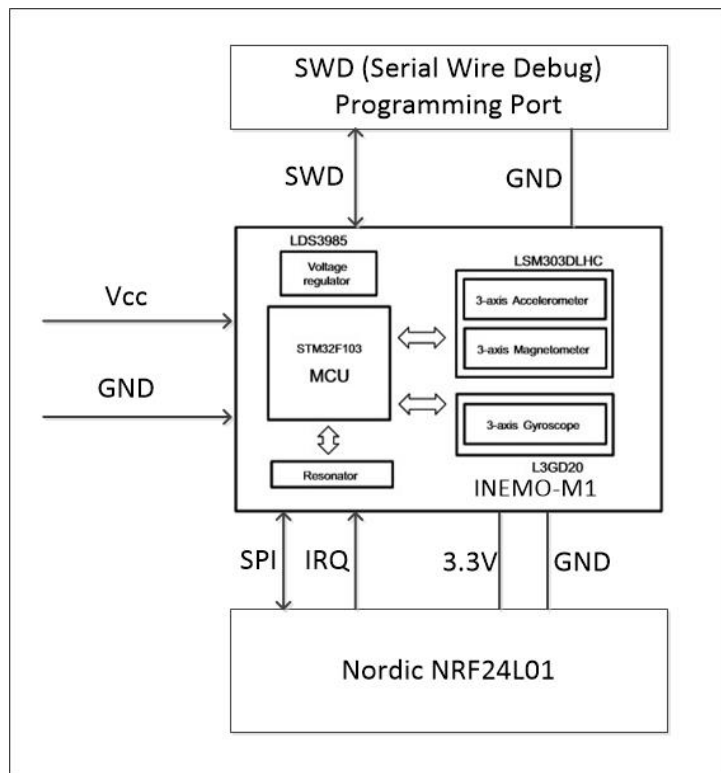
(b) Tail sensor block diagram.

Figure 4.5: The top image shows the block diagram of the collar sensor. The WiFi and GPS modules were not implemented. The bottom diagram shows the block diagram for the tail sensor.
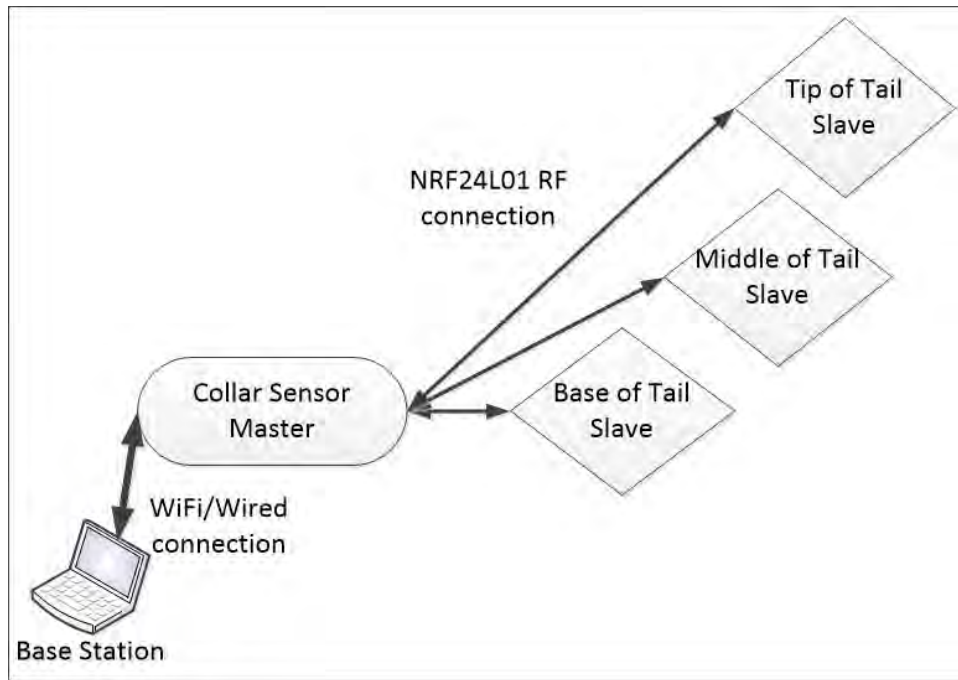
Figure 4.6: Image showing the topology used for the sensor network.

4.2 Software Design

The master sensor (the collar sensor) runs an RTOS (Real time operating system) called freeRTOS [107]. The RTOS has two main tasks. The task with the highest priority is the task responsible for communicating with and getting the data from the tail sensors. The reason why this task has the highest priority is due to the fact that the data from the tail sensors need to be synchronized, which is handled by this task. The second task is responsible for sending the data back to the base station. This task has a lower priority as the data received on the PC was not time critical. An RTOS ensures that the networks sensor data is captured every 10 ms (a frequency of 100 Hz and a sample interval of 10 ms).

The collar sends out a request to all the tail sensors, at the same time, for them to read the MEMs data (the 9 axis IMU). This ensures that the data from all the sensors are synchronized. The collar then individually sends a request to each sensor requesting its data. This data is then collated and sent to the PC. An activity diagram of how the collar sensor works is shown in Figure 4.7.

The tail sensor begins by setting up its required peripherals and then waits for the request to read the MEMs sensors. Once it has read all the data it waits for the request to send its data to the collar sensor. Each tail sensor has a unique request code and a common read data code. When the tail receives data on the NRF24L01 chip, an interrupt is generated and wakes up the processor. An activity diagram of how the tail sensor works is shown in Figure 4.8.

A PC application was developed in Visual C# that allows for data logging functionality
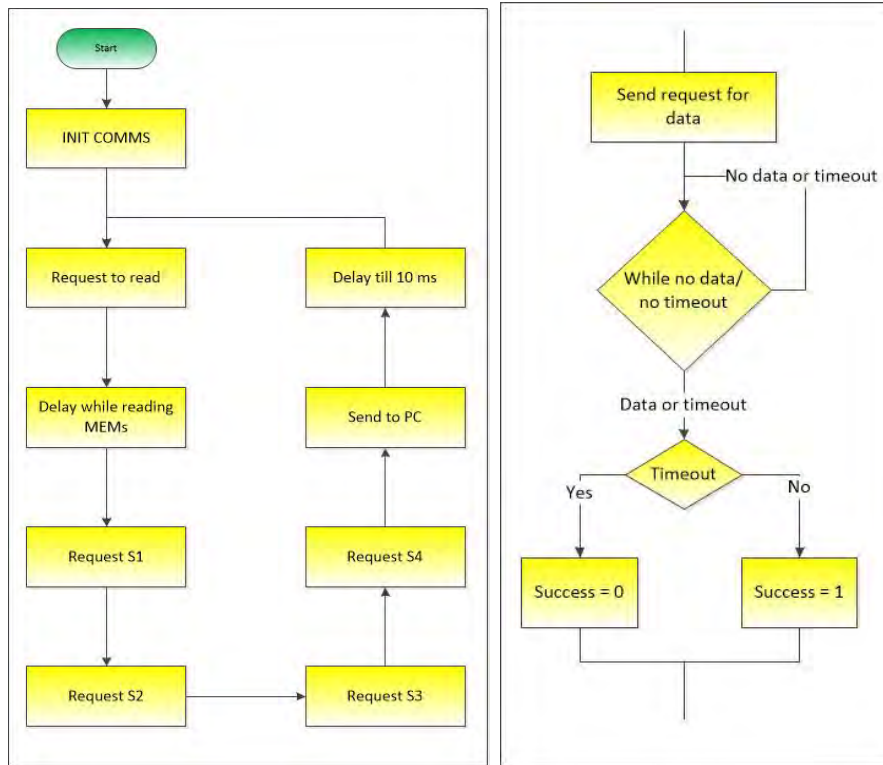
(a) Activity diagram of the collar sensor. (b) Activity diagram of the request sensor block.

Figure 4.7: Image on the left showing the activity diagram of the collar sensor. Due to complexity and repetition, the request sensor (S1, S2, S3 and S4) activities have been removed and are shown on the right. A success byte was transmitted to the PC to determine which sensor packet was received successfully. Note S4 is the collar sensor, while S1 to S3 are the tail sensors.

and to check if the communication between the collar and the tail sensors is functioning. The data is displayed on the application for the user to see and the GUI (graphical user interface) is shown in Figure 4.9. Due to the fact that the application can log data, the collar did not have data logging functionality. The collar receives data from the slaves, added its own IMU sensor data, packetizes it and transmits it to the base station. The sensors were calibrated before the tests were performed. The calibration algorithms are discussed in detail in Appendix A.1.

4.2.1 MOCAP Frequency Validation

Consider the case of a sensor at the tip of the tail, while the tail is rotating. The length of the tail is 0.7 m and has a peak angular rate of 11 rad/s (see Section 3.3). Two rates are compared, 60 Hz (the frequency of the camera system) and 100 Hz (the frequency of the sensor network). The angular displacement in one time step of the sensor at the tip of the tail (0.6 m from the pivot point) is as follows:
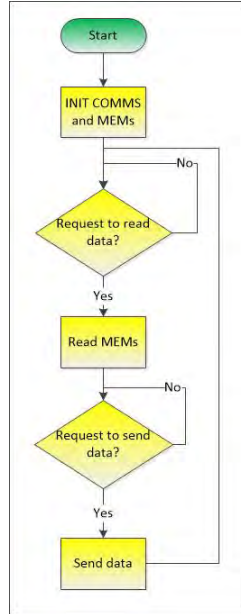
Figure 4.8: Image showing the activity diagram of the tail sensor.
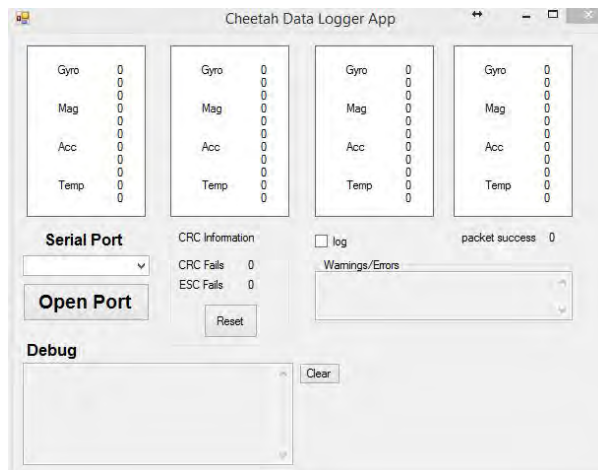
Figure 4.9: Image showing the C # application that was used to data log the data from the sensor network. The application runs on the base station.

$$\Delta\theta = \omega\Delta t = \frac{\omega}{f}$$

$$\Delta\theta_{100\ Hz} = \frac{11}{100} = 0.11\ radians = 6.3\ degrees \quad (4.1)$$

$$\Delta\theta_{60\ Hz} = \frac{11}{60} = 0.1833\ radians = 10.5\ degrees$$

where f is the frequency, ω is the angular rate and Δt is the sample interval. This corresponds to an arc length per time step of:

$$s = r\theta$$
$$s_{100 \ Hz} = 0.6 \times 0.11 = 0.066m \quad\quad (4.2)$$
$$s_{60 \ Hz} = 0.6 \times 0.1833 = 0.11m$$

where s is the arc length, r is the radius from the pivot point to the sensor and θ is the angular displacement. Therefore by having the system operate at a higher frequency the maximum displacement that needs to be tracked is 0.066 m compared to 0.11 m. This will help improve the accuracy of the system and enable it to track higher speed manoeuvres. With the use of a real time operating system the network is capable of data logging at 100 Hz.

4.3 Cost of the Sensor Network

Cost along with ease of development were two of the chief design goals for the motion capture network. As such, special care was taken to select the components. The individual prices for all the components of the tail sensor are listed in Table 4.1 and the individual prices for all the components of the collar sensor are listed in Table 4.2:

Table 4.1: Cost per tail sensor and bill of materials

Item	Quantity	Price
INEMO	1	R 700
PCB	1	R 47
NRF24L01	1	R 68
Pin header	1	R 2.57
Battery	1	R 25
10K 0603 resistor	5	R 0.30
4u7 0805 capacitor	2	R 1.74
	TOTAL	R 844.61

The base station can use the built in WiFi module which is found in most laptops. The total cost of the network is $3 \times 844.61 + 1821.95 = R \ 4 \ 354.83$. The cost of the Xsens sensor network (with software and a lifetime licence) was quoted at approximately R 361 997 (Appendix A.14).

A cheaper sensor network can be developed using a surface mount IMU and a simple low power 8-bit microcontroller instead of the available INEMO-M1 boards. This will increase the complexity of the PCB design and will make assembling and developing the boards much harder. Therefore the cost of the INEMO-M1 is outweighed by its ease of development. With mass production the cost of the components (especially the PCB boards) will drastically decrease resulting in a cheaper sensor network.

Table 4.2: Cost per collar sensor and bill of materials

Item	Quantity	Price
STM F4	1	R 190
WiFi	1	R 207
GPS (NEO-6)	1	R 445
INEMO	1	R 700
PCB	1	R 67
NRF24L01	1	R 68
Pin header	1	R 2.57
Battery	1	R 130
10K 0603 resistor	10	R 0.60
4u7 0805 capacitor	4	R 3.48
lm317	1	R 7
resistors	10	R 1.30
	TOTAL	R 1 821.95

Chapter 5

Cheetah Fur Experiments

During the course of this research a cheetah was euthanized due to renal failure at the National Zoological Gardens in Pretoria, South Africa and the tail became available for this research. It was hypothesized that the fur had aerodynamic effects and this effect was tested and quantified using a tail mock-up rig in a wind tunnel. The aerodynamic experiments performed, along with COM and inertia calculations, are described in this chapter.

The research conducted in this chapter was performed by the author along with Dr. Amir Patel and Prof. Edward Boje. This is a brief revision as to why the aerodynamic effect of the tail was modelled. Further and more detailed research and discussion was done in Dr. Amir Patel's PhD [38] and in a journal paper that is under revision [40]. The author's contribution to this work was to construct the wind tunnel rigs and to perform the experiments. The author also wrote the Matlab code to calculate the drag coefficient, centre of mass and moment of inertia of the tail.

5.1 The Cheetah

Dr. Amir Patel (supervisor of this research) got the opportunity to observe the necropsy and to help record the data that was used for this research. The cheetah tail fur was then brought back to Cape Town, South Africa, to be examined.

The cheetah was 12 years old and weighed 27 kg. Other than renal disease the cheetah was examined by an expert vet and was determined to be anatomically normal. The tail length of the cheetah was 74 cm and the length of the body (from the neck to the base of the tail) was measured to be 77.5 cm. The mass of the skinless tail was 519.5 g and the mass of the fur and skin was 162 g. The total tail mass was approximately 2.5 % of the body mass.

From images of before the necropsy, the cheetah tail was assumed to be a cylinder of roughly uniform diameter as seen in Figure 5.1, but from the necropsy results it was observed that the tail gets increasingly thin towards the tip. The fur at the tip of the tail was noted to be the longest and thickest. Due to this observation, it was decided to investigate the effects of the fur on the torque produced by the tail on the body of the cheetah. After the experiments the fur was preserved using Borax salt.

Figure 5.1: Image of the fur from the necropsy. Note the long and thick tip fur. Image from [38].

5.2 COM and Inertia Calculations

The moment of inertia of the tail (about the base of the tail) and the COM of the tail were calculated from the necropsy data. The necropsy data included all the relevant measurements (mass of the bones, mass of the muscles, size of the bones and location of the bones).

To calculate the moment of inertia of the tail the following formula was used for the kth vertebra (modelled as a cylinder):

$$I_{COM_k} = \frac{1}{12}m_k(3r_k^2 + l_k^2) \tag{5.1}$$

where m_k is the mass of the kth bone and muscle, r_k is the radius of the cylinder that the bone was modelled as and l_k is the thickness of the bone. I_{COM_k} was then shifted to find

the moment of inertia about the base of the tail using the parallel axis theorem which is as follows:

$$I_{base_k} = I_{COM_k} + m_k H_k^2 \tag{5.2}$$

where H_k is the distance from the base of the tail to the COM of the kth bone. This is summed over all the bones to find the total moment of inertia of the tail. The total moment of inertia of the tail about the base was calculated to be 0.0588 kgm^2.

The following formula was used to calculate the distance, L_{COM}, from the pivot point to the COM of the tail:

$$L_{COM} = \frac{\sum_{k=1}^{n} m_k H_k}{m_{net}} \tag{5.3}$$

where there are n vertebrae, m_k is the mass of the kth bone and muscle, H_k is the distance from the base to the COM of the kth vertebrae, m_{net} is the total mass of the tail and L_{COM} is the distance from the pivot point to the COM of the tail. It was calculated that L_{COM} was 18.61 cm from the base of the tail and the total length of the tail was 74 cm long. This resulted in the COM of the tail being 26 % of the tail's length away from the base of the tail.

These values are plausible as they fall in the range of other known data for the COM and inertia of the tail from [37]. The mean moment of inertia of the five tails in [37] was 0.045 kgm^2 with a standard deviation of 0.012 kgm^2 and the mean COM was 20 cm with a standard deviation of 3 cm.

From the necropsy data, the tail was shown to not be as heavy as may be expected from the literature [7] [8] and the COM was close to the base of the tail resulting in a relatively low inertia value. Therefore the assumption that the tail is heavy and has high inertia is incorrect.

It was then hypothesised that the aerodynamic forces could produce significant torque on the body in combination with the inertial torques. Aerodynamic experiments were thus performed to validate this hypothesis.

5.3 Cheetah Fur Experiments

The aerodynamic effects of the cheetah tail during dynamic rapid manoeuvres is still unknown. It was observed that the tail becomes thin towards the tip whereas the fur gets longer. The tip of the tail travels at the highest velocity when the tail is flicked. It

was therefore decided to perform an experiment to quantify the aerodynamic effects of the cheetah fur.

Two different experiments were performed. The first experiment used the fur from the middle of the tail and the second experiment used the fur from the tip of the tail. The second experiment investigated the effects of the angle of attack of the tail. Due to the fact that the cheetah is running forward and flicking its tail, the angle of attack of the tip varies drastically over a 90 degree flick. This is due to the forward velocity vector being added to the angular velocity vector of the tail as seen in Figure 5.2. More details on this experiment can be found in [38] and [40].
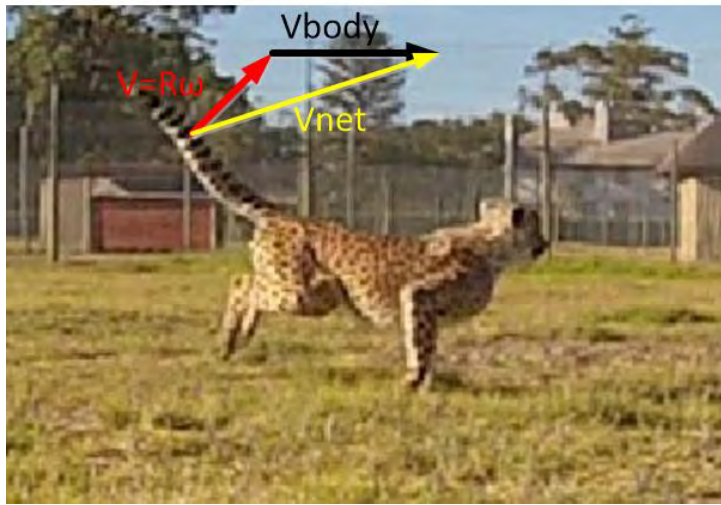
Figure 5.2: Image showing the velocity vector components. Note how the forward velocity component due to the cheetah running changes the angle of attack of the net velocity experienced by the flicking tail. Image filmed at Cheetah Outreach, Somerset West, South Africa.

5.3.1   Test Rig

Only the results from the first experiment that used the fur from the middle of the tail was needed to accurately model the aerodynamic effects for the dynamic models of the spine and tail. The reason why the angle of attack experiment results were not needed was due to the fact that the mechanical rig was stationary and therefore V_{body} in Figure 5.2 was zero and the resulting drag force was always perpendicular to the tail. A rig to calculate the force generated by the tail was designed and built. The rig consisted of three cylinders of 50 mm PVC pipe placed vertically in a row in the wind tunnel. The top and bottom cylinders were fixed to the wind tunnel and were used to remove/reduce the boundary layer and end effects. By removing end effects we are assuming the cylinder is infinitely long. The rig can be seen in Figure 5.3.

The middle cylinder was free to move and was attached to a centre beam that went through the top cylinder and out the top of the wind tunnel. This was connected to a pivot and transferred a force on a scale as seen in Figure 5.3.

The wind speed of the wind tunnel was adjusted from 10 m/s to 30 m/s in 5 m/s intervals,
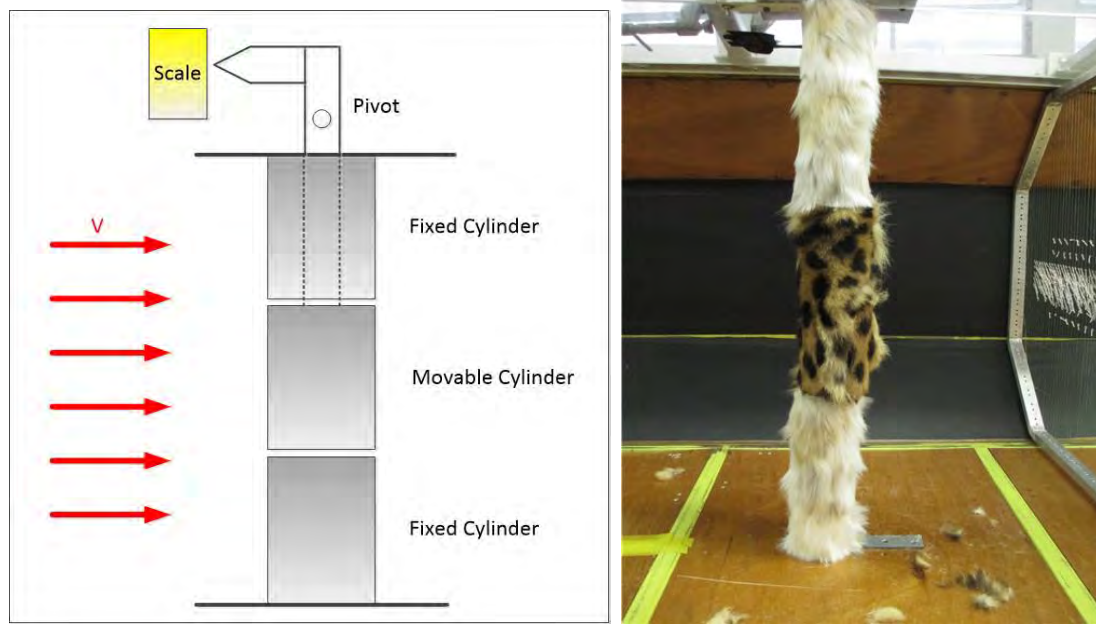
(a) Image showing how the rig works.

(b) The cylinder rig with cheetah fur.

Figure 5.3: Image showing how the cylinder rig works. Note the three cylinders and the scale to measure the force generated. The middle cylinder is free to rotate. It is attached to a beam that pivots above the wind tunnel. The force that the scale exerts to keep the centre cylinder in position is recorded. The arrows represent the direction of the wind. The other two cylinders are attached to the wind tunnel and are used to remove the boundary layer effects.

and nine measurements of the force exerted by the lever arm onto the scale, which keeps the middle cylinder in line with the other two cylinders, was recorded. The temperature and humidity were also recorded at the time of the experiment.

To ensure the rig accurately measured the drag force, and the drag coefficient was calculated correctly, the experiment was first run using the smooth cylinders with no fur. The results were compared to known textbook data of a smooth cylinder to validate the rig set-up and calibration. Reynolds numbers were required and it was determined that the rig accurately measures the drag force. The comparison of textbook and experimental drag coefficients versus Reynolds number can be seen in Figure 5.4.

The cheetah fur was then attached to the middle cylinder and fake fur (furry material from a fabric shop to simulate cheetah fur) on the other two cylinders using quick drying glue. Fake fur was used due to the scarcity of cheetah fur and helped emulate an infinite cylinder by removing the end effects.

5.4   Wind Tunnel Results

It was hypothesized that the hair on the tail helps generate a reaction torque on the body of the cheetah due to its aerodynamic properties. The fur gives the tail a larger effective
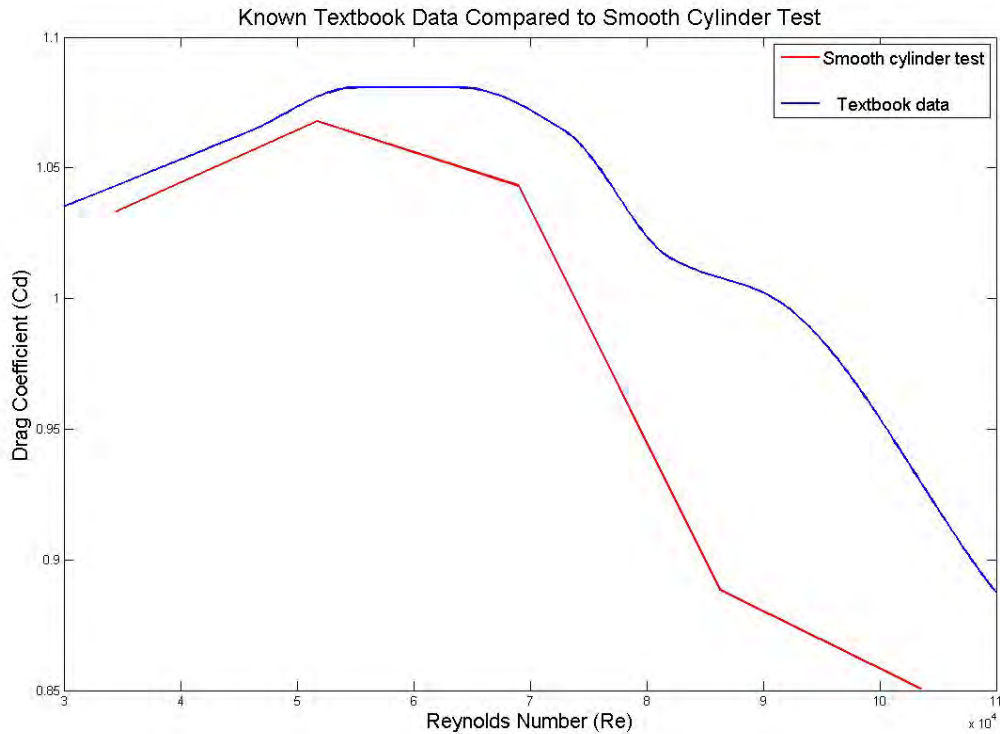
Figure 5.4: Image showing the comparison of the smooth cylinder and text book data. As can be seen the test results follow the same general curve of known textbook data [51]. This validates the wind tunnel rig set up and calculations.

area (see Section 5.4.1 for more detail) which is beneficial as fur is lighter than muscle. If the tail was the size of the effective diameter and hairless, it would increase the mass of the tail and require larger muscles and more energy to actuate the tail. This will effect the maximum acceleration of the tail. A lighter tail allows for faster motion and larger drag forces.

It must be noted that the fur was observed to collapse at higher wind velocities. From wildlife footage this effect was not observed. This has lead to the belief that some form of piloerection [108] function may be used to keep the fur from collapsing.

The results of the experiment are shown in Figure 5.5. As can be seen the drag coefficient of the cheetah fur was higher than the smooth PVC cylinder. The drag coefficient also remained relatively constant across the different wind speeds. The smooth cylinder results start to break down at higher velocities. The diameter of the furry cylinder that was used was 0.057 m. This was measured using a vernier and clamping the sides of the cylinder with fur on it. This however is not valid as there is no distinct fur-air boundary and the diameter was taken as the skin boundary. This introduced the effective diameter that is discussed in Section 5.4.1.
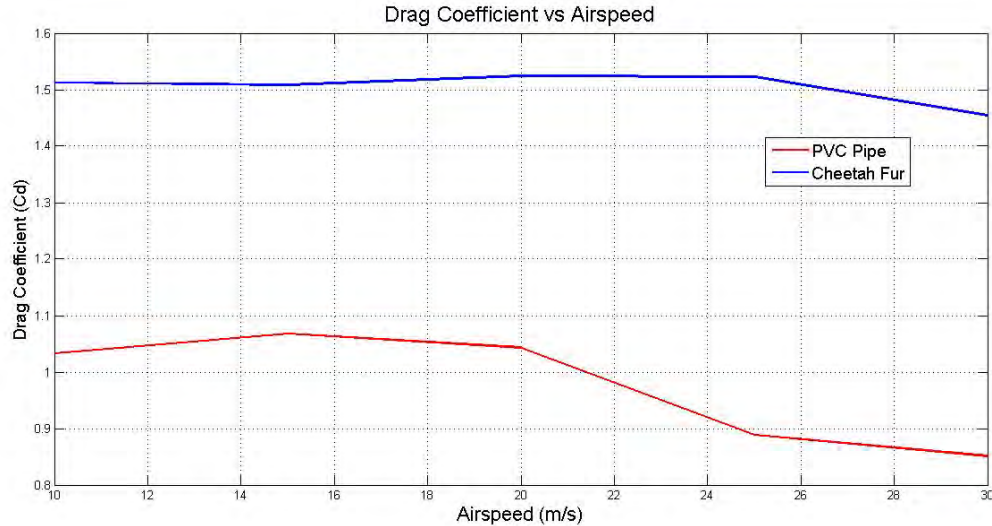
Figure 5.5: Graph showing the drag coefficients calculated from the cylinder tests. As a reference the smooth cylinder data is also included. Note that the furry cylinder drag coefficient is much higher than the smooth cylinder. The diameter of the smooth cylinder was 50 mm and the diameter of the furry cylinder was taken as the skin boundary of 57 mm.

5.4.1 The Effective Diameter

Due to the fact that the cheetah tail with its fur does not have a distinct fur-air boundary an effective area was defined. This area made the drag force at 10 m/s match that of the smooth cylinder. This effective area can then be compared to the area of the skin which was measured with a vernier. This indicates what size smooth cylinder must be used to generate the same force as the smaller, furry cylinder.

The actual diameter of the cylinder with fur was 0.057 m but to get the measured data to line up with smooth cylinder data, an effective diameter of 0.0835 m was used which is 46 % larger. This resulted in the graph shown in Figure 5.6. As can be seen the cheetah fur drag does not break down at the same Reynolds number as the smooth cylinder.

5.5 Drag versus Inertia Torque

A simulation of a cheetah flicking its tail while running forwards at 20 m/s was developed in [38]. The simulation compared the inertial and aerodynamic torques generated by the tail. The results are shown in Figure 5.7. As can be seen the aerodynamic torque plays a vital role in the net torque generated by the tail.

As can be seen the aerodynamic effect of the tail plays a vital role in generating a reaction torque on the body. Therefore the aerodynamic and inertia effect, relatively low inertia and the position of the COM, were taken into account when modelling the cheetah tail. The advantages of the fur is that the tail has a smaller mass cost and the aerodynamic

63

Figure 5.6: Graph comparing the calculated drag coefficients of the furry cylinder using the effective area compared to the drag coefficient of a smooth cylinder. Note how the smooth cylinder collapses before the furry cylinder does.
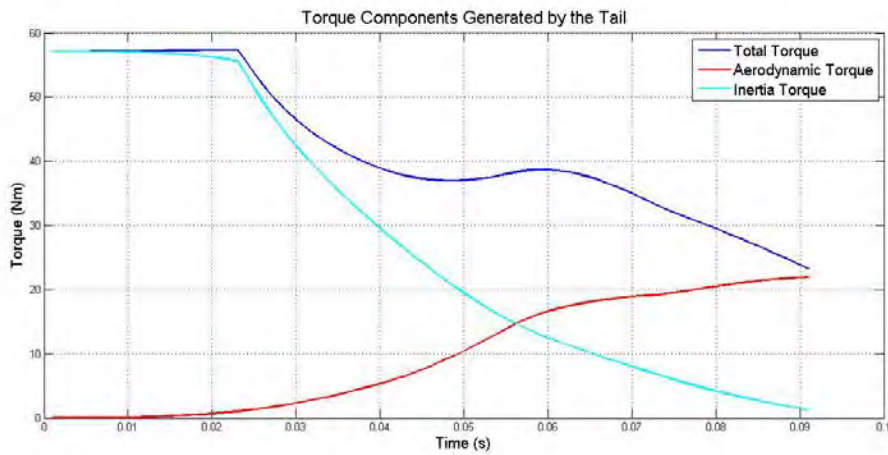
Figure 5.7: Drag torque versus inertia torque. Note how the inertia torque tends to zero as the drag torque increases. Image from [38].

effects from the fur break down at higher Reynolds numbers compared to a smooth cylinder.

Chapter 6

Filtering and State Estimation

In order to track the states of the spine and tail of a cheetah (or the mechanical rig), a sensor network was attached to the system and data from the sensors (accelerometers, magnetometers and gyroscopes) were logged. The data, along with the dynamic model of the system, was used in the filtering and state estimation algorithms that are developed in this chapter.

6.1 Aim of the State Estimation Algorithms

The aim of the state estimation algorithms were to track the positions of the sensors and therefore the motion of the spine and tail. The sensors were placed onto a mechanical rig which was situated under a camera system. The camera system provided a ground truth that was used to validate the performance of the algorithms using two cost functions.

The first cost function was the RMS error between the estimated position of the sensor and the ground truth of the sensor. The second cost function was the RMS error between the estimated angular states and the angles calculated from the ground truth measurements.

The mechanical rig flicked the tail and bent the spine in order to generate sensor data that was used in the state estimation algorithms. The data was logged at 100 Hz and the camera system data was logged at 60 Hz, and interpolated to 100 Hz.

6.2 Tuning the Filter Covariance

The filter covariance was tuned using Matlab's Fmincon and GlobalSearch optimization functions. These two Matlab functions optimize a cost function by finding the global minimum. For consistency, only the angular rate noise covariances were tuned in this

procedure. The angle noise covariances were set to zero as they are kinematically dependent on the angular rates. The torque covariances are derived and discussed in Section 6.2.1. Two cost functions were developed, see Section 3.4, to compare the performance of the algorithms. The algorithms were optimized using the RMS error of the position of the sensors cost function. These tuned values formed a diagonal process covariance matrix that was required by the Kalman filter. The measurement covariance matrix of the Kalman filter was obtained from the sensor datasheets.

6.2.1 Tuning the Torque Covariance

To get the covariance coefficients for the torques, the expected accelerations were modelled. It was assumed that a 90 degree flick, in 0.5 seconds, must be tracked. The angle of the tail during a flick was modelled as a ramp and contains two acceleration steps that are shown in Figure 6.1. The commanded torque to achieve this acceleration is modelled as a step (according to Newton's law of motion, $J\alpha = \tau$, therefore a step acceleration results in a step torque). The resulting angular velocity of the tail was therefore modelled as two ramp functions.
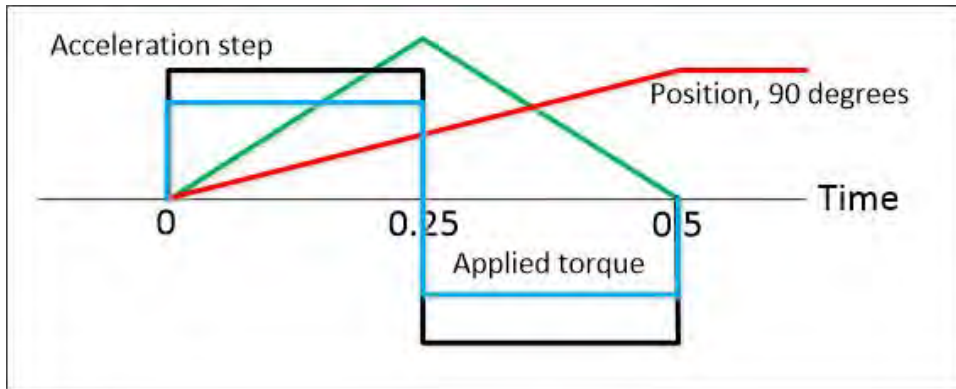
Figure 6.1: Image showing the acceleration steps, applied torque, velocity and position of the tail. A 90 degree flick was modelled as shown in the red line. The green line is the modelled angular velocity of the tail. The blue line is the modelled torque profile and the black arrows are the modelled acceleration spikes.

Using the standard angular position equation, $\theta_2 = \theta_1 + w_1 t + 1/2\alpha t^2$, the acceleration at the halfway point, 45 degrees, can be determined. This occurs at 0.25 seconds and the angular acceleration is $\alpha = 25\ rad/s^2$ (the initial conditions are zero).

Therefore the applied torque for an acceleration of $25\ rad/s^2$ is $25J$ where J is the moment of inertia of the tail. This is the expected torque value. The covariance of the expected values is the square of the value. Due to the fact that the system is a discrete system, the expected value was discretized and resulted in the following covariance value:

$$covariance = \tau^2(\Delta t)^2 = (25J)^2(\Delta t)^2 \tag{6.1}$$

where Δt is the sample interval of the system.

6.3 The Tail System

The tail was modelled using two rigid beams. Three algorithms were developed: *TRIAD KF*; *Multibody*; and *Multibody with torque*. These algorithms track the motion of the tail in the 2D case, shown in Figure 6.2a, and were then extended to the 3D case, shown in the Figure 6.2b. In the 2D case the beams can only pitch while in the 3D case they can pitch and yaw. The algorithms are developed below. The subscript 1 represents the first beam of the tail (closest to the body) and subscript 2 represents the second beam. θ is the pitch angle and ψ is the yaw angle.

The difference between the three types of algorithms that were developed was that the *TRIAD KF* algorithm relies on the TRIAD algorithm as the update for the KF. The TRIAD algorithm [109] was explained in detail in the literature review (see Section 2.7.2). The difference between the *Multibody* and *Multibody with torque* algorithm was that the *Multibody* algorithm assumes that the applied torque on the tail, due to the tail muscles, is equal to the potential energy and aerodynamic torques and therefore cancel each other out. In other words the applied torque, potential energy and aerodynamic effect are set to zero. The *Multibody and torque* algorithm models the potential energy, aerodynamic force and applied torque.

6.3.1 Algorithm 1: The *TRIAD KF* Algorithm

The first algorithm that was developed was the *TRIAD KF* algorithm. This algorithm used the TRIAD algorithm to generate the update measurements for the Kalman Filter. The gyroscope measurements were used for the prediction equations. The TRIAD algorithm is one of the most common attitude determination algorithms and is frequently used in the literature [91] [92] [109].

The outputs of the TRIAD algorithm were θ_1 and θ_2 for the 2D system and θ_1, θ_2, ψ_1 and ψ_2 for the 3D case, as seen in Figure 6.2, and were used as the measurement data for the Kalman filter. The TRIAD output noise covariance (the Kalman filter measurement noise covariance) as well as the Kalman filter process noise was determined using Fmincon and GlobalSearch, see Section 6.2. The layout of the *TRIAD KF* algorithm is shown in Figure 6.3. The *TRIAD KF* algorithm results are used as the benchmark and the other filters are compared to its results.
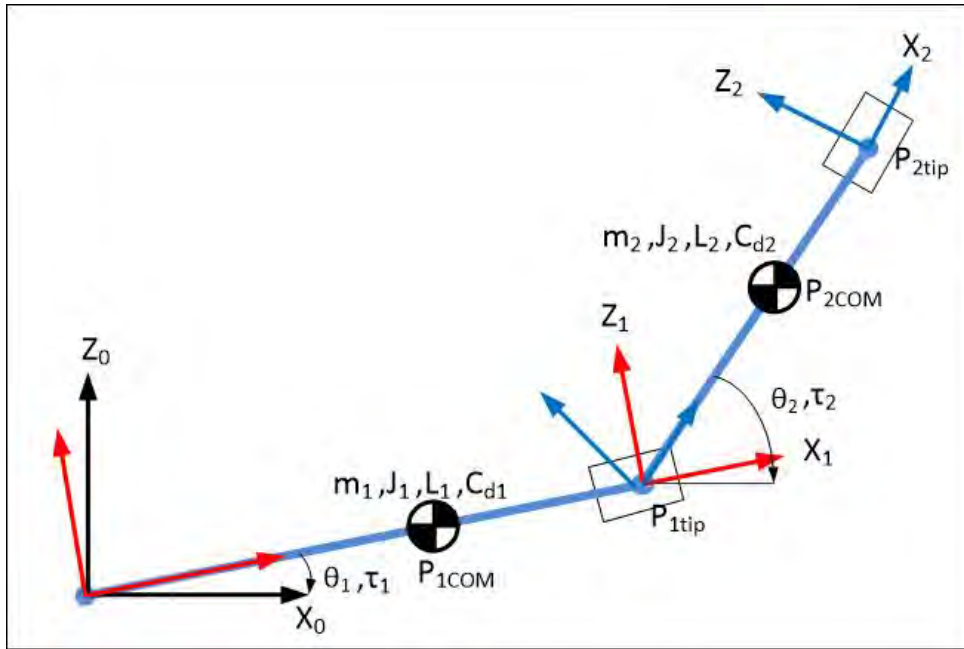
(a) 2D tail system. The beams can only pitch.
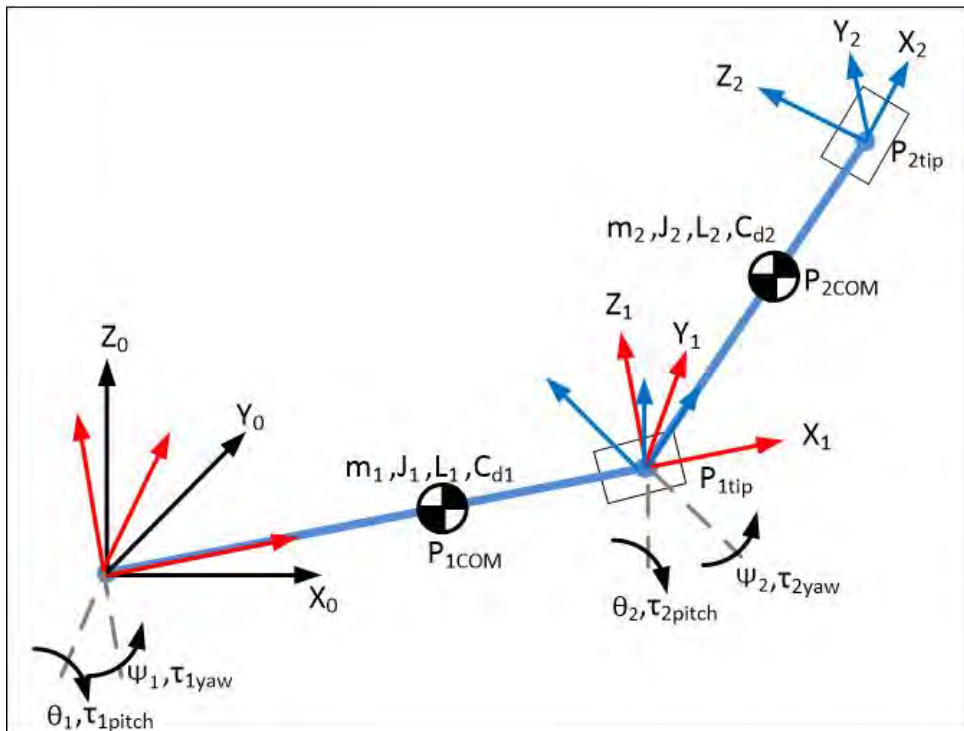
(b) 3D tail system. The beams can pitch and yaw

Figure 6.2: 2D and 3D System diagrams for the tail models

2D System

A pseudo code algorithm for the TRIAD algorithm is shown in Appendix A.8. The state variables of the Kalman filter are $\mathbf{x} = \begin{bmatrix} \theta_1 & \theta_2 \end{bmatrix}'$. The prediction equation is as follows:
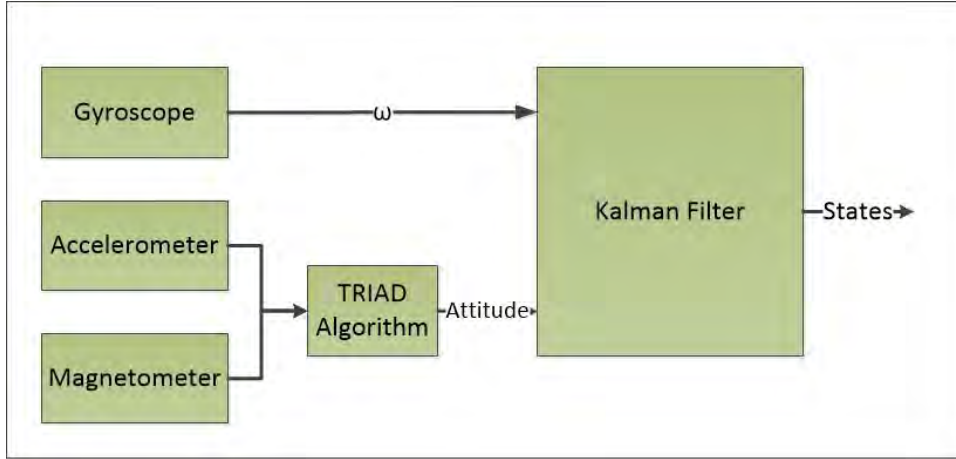
Figure 6.3: Block diagram showing the gyroscope measurements and outputs of the TRIAD algorithm feeding into the KF algorithm. The magnetometer and accelerometer measurements feed into the TRIAD algorithm.

$$\hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1})$$
$$\hat{\mathbf{x}}_{k|k-1} = \hat{\mathbf{x}}_{k-1|k-1} + \Delta t \begin{bmatrix} \omega_{inertial_1} & \omega_{inertial_2} \end{bmatrix}' \tag{6.2}$$

where ω_i is the respective angular rate from the gyroscope reading rotated into the inertial frame and Δt is the sample interval of the filter. The $\mathbf{H}$ matrix is the identity due to the fact that the measurement data was in the same form as the states which is shown in Equation 6.3. This is one of the advantages of having the TRIAD algorithm feed into the Kalman filter. It drastically simplifies the measurement update stage.

$$\mathbf{z}_k = h(\mathbf{x}_k)$$
$$\mathbf{z}_k = [TRIAD_{\theta_1} \ TRIAD_{\theta_2}]$$
$$\mathbf{H}_k = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}|_{\hat{\mathbf{x}}_{k|k-1}} \tag{6.3}$$
$$\mathbf{H}_k = \mathbf{I}$$

The algorithm was optimized using the camera data and data from the sensor network that was placed on the mechanical rig during a simulated tail flick. The results of the algorithm are shown in Figures 6.4 and 6.5. A cumulative distribution function of the error is shown in Figure 6.6. The two cost functions equated to an RMS error of 20.36 degrees and an RMS error of 0.16 metres. The results are discussed and compared further in Section 6.5.
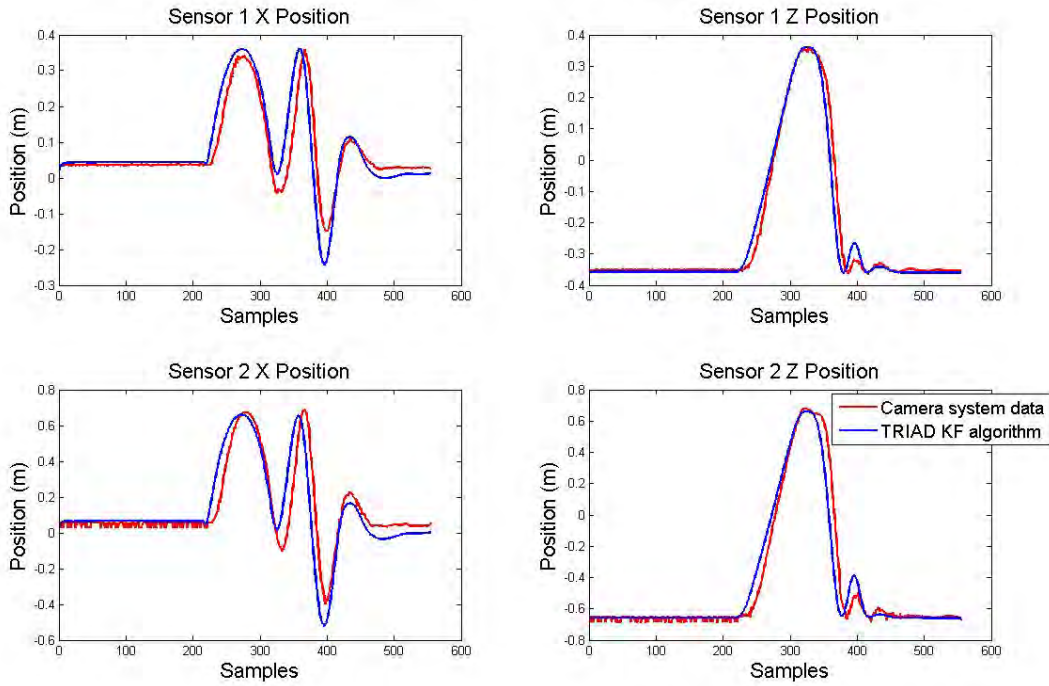
Figure 6.4: 2D tail *TRIAD KF* estimated positions results.
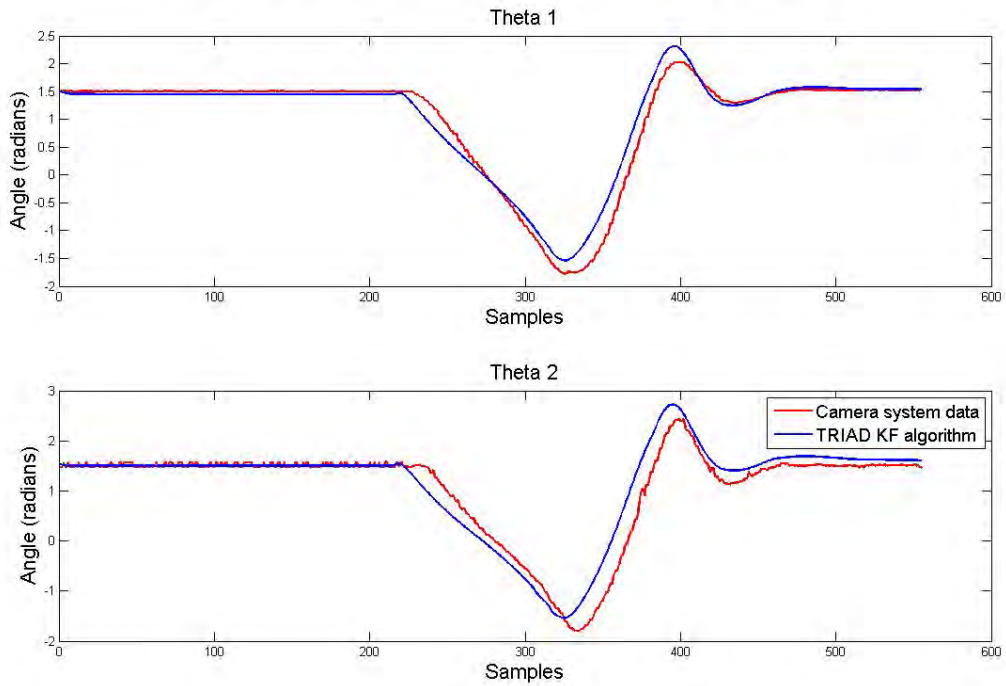
Figure 6.5: 2D tail *TRIAD KF* estimated angles results.
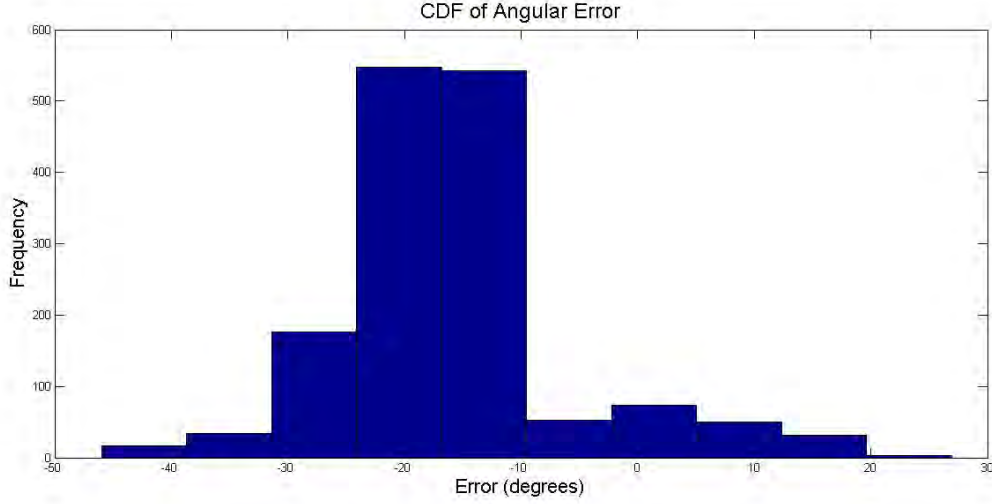
Figure 6.6: A cumulative distribution function of the angular error.

3D System

The 3D version of the TRIAD algorithm is very similar to the 2D version. The state variables of the Kalman filter are $\mathbf{x} = \begin{bmatrix} \theta_1 & \psi_1 & \theta_2 & \psi_2 \end{bmatrix}'$. The prediction equation is shown below:

$$
\begin{aligned}
\hat{\mathbf{x}}_{k|k-1} &= f\big(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}\big) \\
\hat{\mathbf{x}}_{k|k-1} &= \hat{\mathbf{x}}_{k-1|k-1} + \Delta t \begin{bmatrix} \omega_{\theta_{1I}} & \omega_{\psi_{1I}} & \omega_{\theta_{2I}} & \omega_{\psi_{2I}} \end{bmatrix}'
\end{aligned}
\tag{6.4}
$$

The inertial angular rates are derived in Appendix A.2.1. The $\mathbf{F}$ matrix is:

$$
\begin{aligned}
\mathbf{F} &= \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\big|_{\hat{\mathbf{x}}_{k|k-1}, \mathbf{u}_{k-1}} \\
\mathbf{F} &= \mathbf{I} + \Delta t \frac{\partial diag\big[\omega_{\theta_{1I}}\ \omega_{\psi_{1I}}\ \omega_{\theta_{2I}}\ \omega_{\psi_{2I}}\big]}{\partial \mathbf{x}}\big|_{\hat{\mathbf{x}}_{k|k-1}, \mathbf{u}_{k-1}}
\end{aligned}
\tag{6.5}
$$

The $\mathbf{H}$ matrix is calculated as follows:

71

$$\mathbf{z}_k = h(\mathbf{x}_k)$$
$$\mathbf{z}_k = [TRIAD_{\theta_1} \ TRIAD_{\psi_1} \ TRIAD_{\theta_2} \ TRIAD_{\psi_2}]$$
$$\mathbf{H}_k = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}|_{\hat{\mathbf{x}}_{k|k-1}} \tag{6.6}$$
$$\mathbf{H}_k = \mathbf{I}$$

The results are shown in Figures 6.7 and 6.8. A cumulative distribution function of the angular error is shown in Figure 6.9. The algorithm resulted in an RMS error of 49.2 degrees which is equivalent to an RMS error of 0.23 metres. The results are discussed and compared further in Section 6.5.
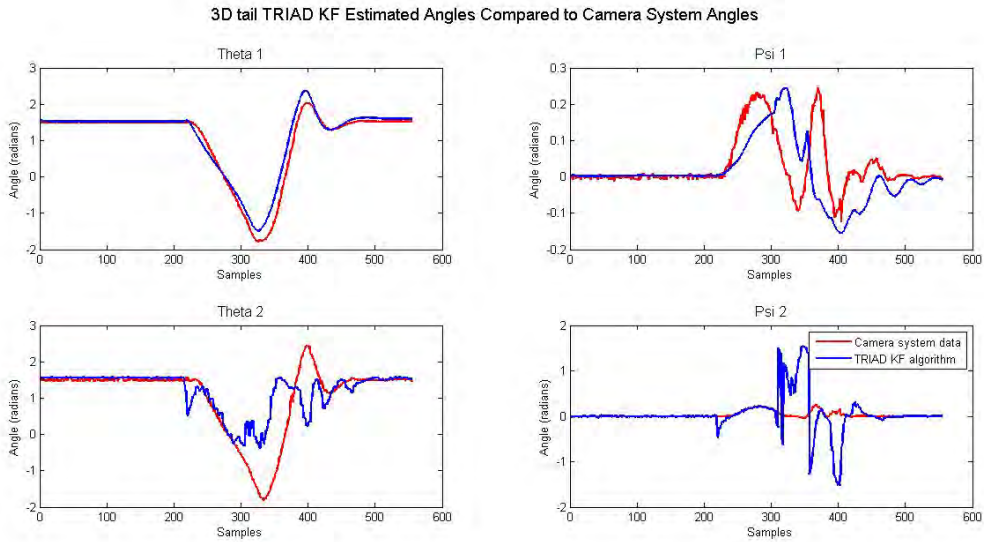
Figure 6.7: 3D tail *TRIAD KF* estimated positions results.
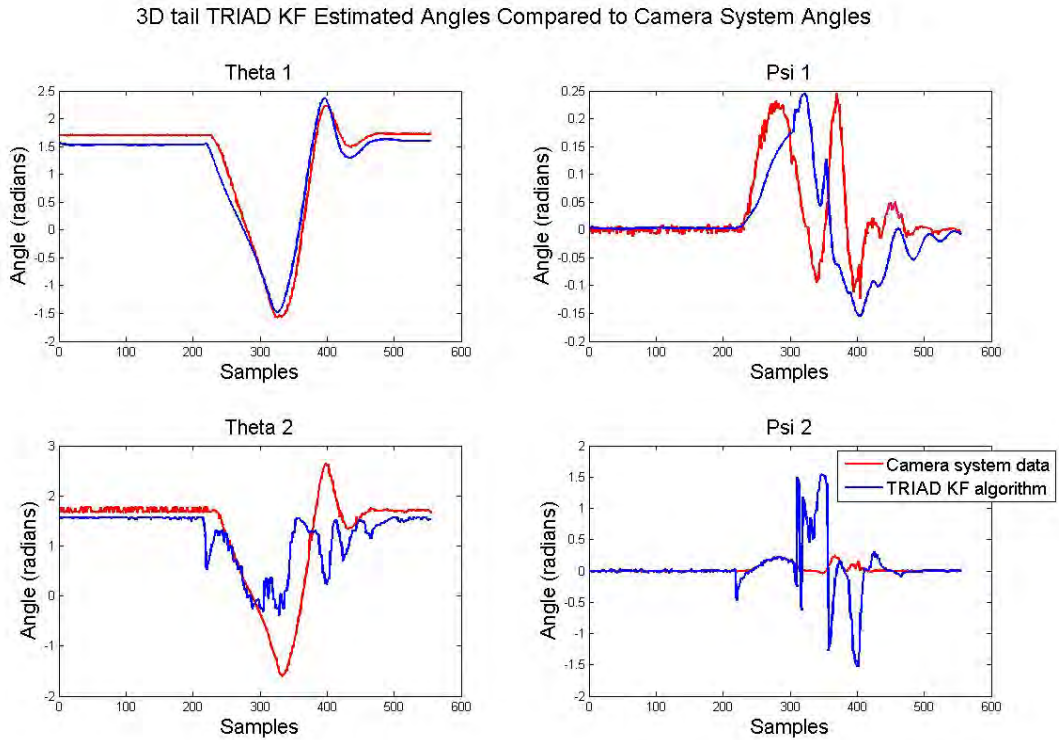
Figure 6.8: 3D tail *TRIAD KF* estimated angles results.
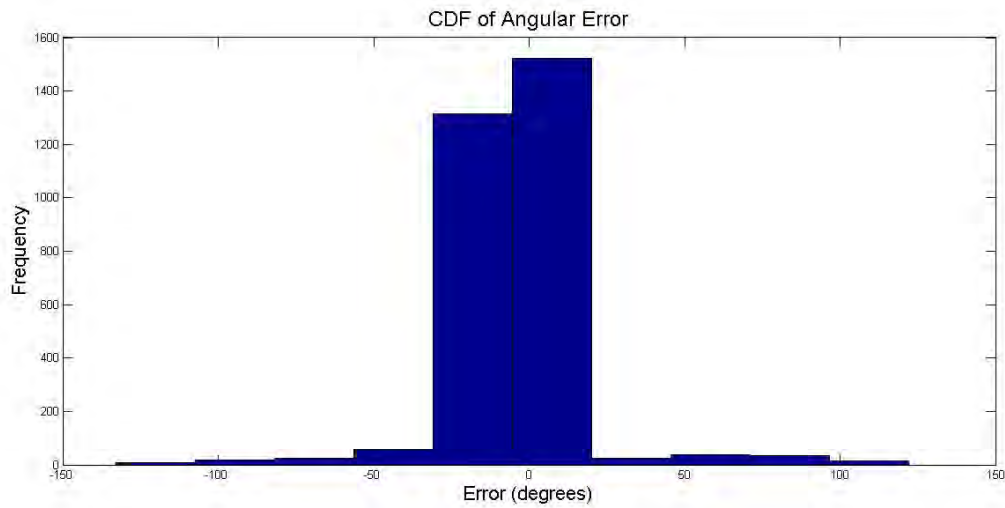
Figure 6.9: A cumulative distribution function of the angular error.

6.3.2 Algorithm 2: The *Multibody* Algorithm

The *Multibody* algorithm does not rely on the TRIAD algorithm, instead it utilizes the dynamic model of the system to predict the future states. These states then get updated using the measurement equations. The dynamics of the system are first modelled using Lagrange dynamics and then applied to the filter algorithms.

2D System

The generalized coordinates, see Section 2.3.2, for the Lagrange dynamics are $\mathbf{q} = \begin{bmatrix} \theta_1 & \theta_2 \end{bmatrix}'$ as seen in Figure 6.2. The kinetic energy for the system is shown in Equation 6.7. The velocity of the COM of the rigid beams is required. The equations to calculate the position, $\mathbf{p}_i$, and velocity, $\dot{\mathbf{p}}_i$, of the COM are shown in Appendix A.3.

$$
\begin{aligned}
T_{translation} &= \frac{1}{2}(m_1 \dot{\mathbf{p}}_{1_{COM}}^T \dot{\mathbf{p}}_{1_{COM}} + m_2 \dot{\mathbf{p}}_{2_{COM}}^T \dot{\mathbf{p}}_{2_{COM}}) \\
T_{rotation} &= \frac{1}{2}(J_1 \dot{\theta}_1^2 + J_2 \dot{\theta}_2^2)
\end{aligned}
\tag{6.7}
$$

where m_i is the mass of the relevant beam and J_i is the moment of inertia of the ith beam, about the point of rotation. The potential energy, generalized forces and applied torques due to the tail muscles were set to zero from the assumption that the applied torque equals the effect of the potential energy and aerodynamic force and therefore cancel each other out. The $\mathbf{M(q)}$ and $\mathbf{C(q,\dot{q})}$ matrices are calculated in Appendix A.4 and are defined in Section 2.3.2. Equation 6.8 is used to solve for the equations of motion as derived in Section 2.3.2.

$$
\ddot{\mathbf{q}} = \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = \mathbf{M(q)}^{-1}(-\mathbf{C(q,\dot{q})}\dot{\mathbf{q}})
\tag{6.8}
$$

where $\mathbf{M(q)}^{-1}$ is the inverse of the mass matrix. The matrices are described in Section 2.3.2. The Kalman filter now needs to be derived. The state variables of the Kalman filter are $\mathbf{x} = \begin{bmatrix} \dot{\theta}_1 & \dot{\theta}_2 & \theta_1 & \theta_2 \end{bmatrix}'$. The prediction and update equations are provided by Equation 6.9 and Equation 6.10 respectively.

$$
\begin{aligned}
\hat{\mathbf{x}}_{k|k-1} &= f(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}) \\
\hat{\mathbf{x}}_{k|k-1} &= \hat{\mathbf{x}}_{k-1|k-1} + \Delta t \begin{bmatrix} \ddot{\theta}_1 & \ddot{\theta}_2 & \dot{\theta}_1 & \dot{\theta}_2 \end{bmatrix}'
\end{aligned}
\tag{6.9}
$$

$$
\begin{aligned}
\mathbf{z}_k &= h(\mathbf{x}_k) \\
\mathbf{z}_k &= \begin{bmatrix} acc_{1_X} & acc_{1_Z} & \omega_{1_Y} & acc_{2_X} & acc_{2_Z} & \omega_{2_Y} \end{bmatrix}'
\end{aligned}
\tag{6.10}
$$

where $\ddot{\theta}_1$ and $\ddot{\theta}_2$ come from the equations of motion and the update equations (acc_{1_X} etc. are derived in Appendix A.6). The results of the algorithm are shown in Figures 6.10 and 6.11 with the cumulative distribution function of the error shown in Figure 6.12.

The angle cost function equated to an RMS error of 14.64 degrees while the position cost function equated to an RMS error of 0.13 metres. The results are discussed and compared further in Section 6.5.
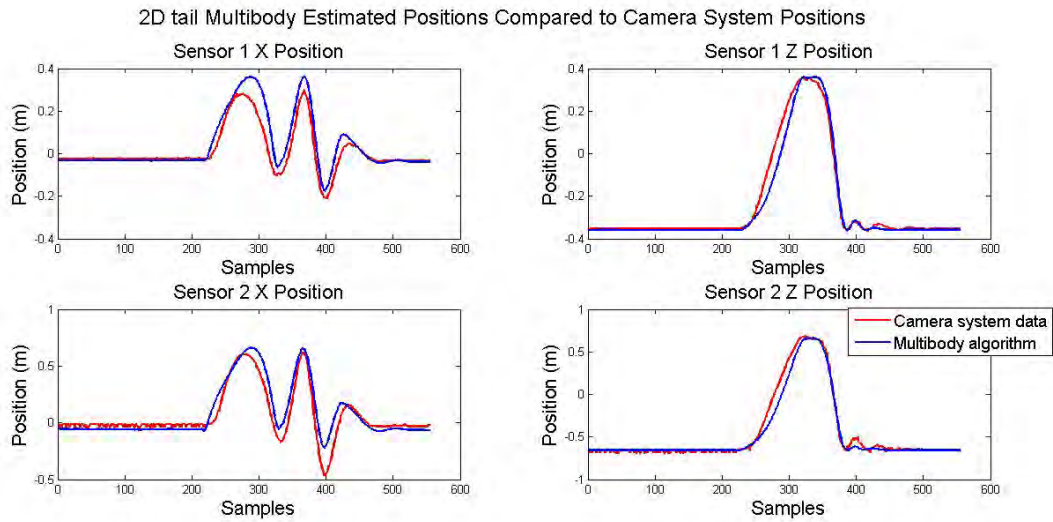
Figure 6.10: 2D tail *Multibody* estimated positions compared to the camera system.
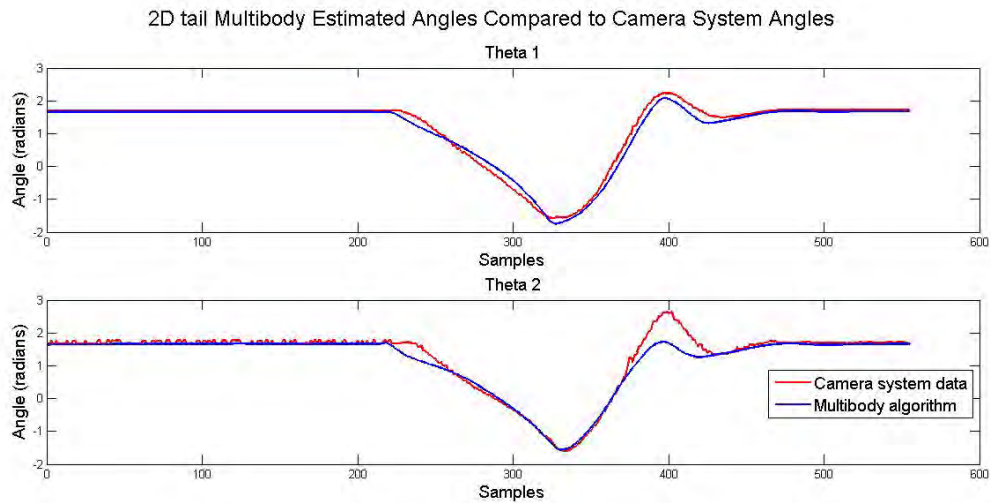
Figure 6.11: 2D tail *Multibody* estimated angles compared to angles calculated using the camera system data.

75
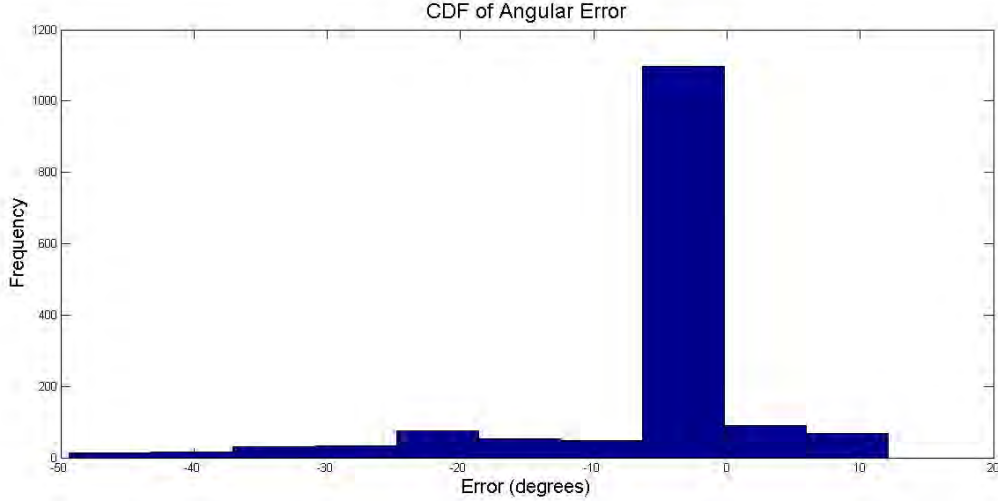
Figure 6.12: A cumulative distribution function of the angular error.

3D System

The 3D system is very similar to the 2D system except that, instead of having each rigid beam only being able to pitch, they can now pitch and yaw (no roll as discussed in the methodology, Section 3.1). The Lagrange generalized coordinates are $\mathbf{q} = \begin{bmatrix} \theta_1 & \psi_1 & \theta_2 & \psi_2 \end{bmatrix}'$. The kinetic energy is

$$
T_{translation} = \frac{1}{2}(m_1 \dot{\mathbf{p}}_{1COM}^T \dot{\mathbf{p}}_{1COM} + m_2 \dot{\mathbf{p}}_{2COM}^T \dot{\mathbf{p}}_{2COM})
$$

$$
T_{rotational} = \frac{1}{2}(\mathbf{J}_{mid}(\mathbf{R}_{gyroItoB}(0, \theta_1, \psi_1) \begin{bmatrix} 0 \\ \dot{\theta}_1 \\ \dot{\psi}_1 \end{bmatrix})^2 + \mathbf{J}_{tip}(\mathbf{R}_{gyroItoB}(0, \theta_2, \psi_2) \begin{bmatrix} 0 \\ \dot{\theta}_2 \\ \dot{\psi}_2 \end{bmatrix})^2).
$$

$$(6.11)$$

The rotation matrices are derived in Appendix A.2 along with the velocities of the COM, $\dot{\mathbf{p}}_i$, in Appendix A.3. The diagonal matrices containing the moment of inertia values about the point of rotation (roll, pitch and yaw axis) are $\mathbf{J}_{mid}$ and $\mathbf{J}_{tip}$. The relevant mass of the beam is m_i. The potential energy and the generalized forces, including the applied torque from the tail muscles, are zero. The equations of motion of the system are shown in Equation 6.12, and were derived in Section 2.3.2.

$$
\ddot{\mathbf{q}} = \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\psi}_1 \\ \ddot{\theta}_2 \\ \ddot{\psi}_2 \end{bmatrix} = \mathbf{M}(\mathbf{q})^{-1}(-\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}})
$$

$$(6.12)$$

76

The $\mathbf{M(q)}$ and $\mathbf{C(q,\dot{q})}$ matrices are derived in Appendix A.4 and discussed in Section 2.3.2. The state variables of the Kalman filter along with the prediction and update equations are shown in Equations 6.13 to 6.15.

$$\mathbf{x}_k = \begin{bmatrix} \dot{\theta}_1 & \dot{\psi}_1 & \dot{\theta}_2 & \dot{\psi}_2 & \theta_1 & \psi_1 & \theta_2 & \psi_2 \end{bmatrix}' \tag{6.13}$$

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= f\big(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}\big) \\ \hat{\mathbf{x}}_{k|k-1} &= \hat{\mathbf{x}}_{k-1|k-1} + \Delta t \hat{\dot{\mathbf{x}}}_{k-1|k-1} \end{aligned} \tag{6.14}$$

$$\begin{aligned} \mathbf{z}_k &= h(\mathbf{x}_k) \\ \mathbf{z}_k &= [acc_{1_X} \ \ acc_{1_Z} \ \ mag_{1_X} \ \ mag_{1_Y} \ \ mag_{1_Z} \ \ \omega_{1_Y} \ \ \omega_{1_Z} \ \ \ldots \\ & \quad acc_{2_X} \ \ acc_{2_Z} \ \ mag_{2_X} \ \ mag_{2_Y} \ \ mag_{2_Z} \ \ \omega_{2_Y} \ \ \omega_{2_Z}]' \end{aligned} \tag{6.15}$$

where the measurement equations (acc_{1_X} etc. are derived in Appendix A.6). The results of the algorithm are shown in Figures 6.13 and 6.14 with the cumulative distribution function of the angular error shown in Figure 6.15. The cost functions for the algorithm equated to an RMS error of 15.98 degrees and 0.12 metres. The results are discussed and compared further in Section 6.5.
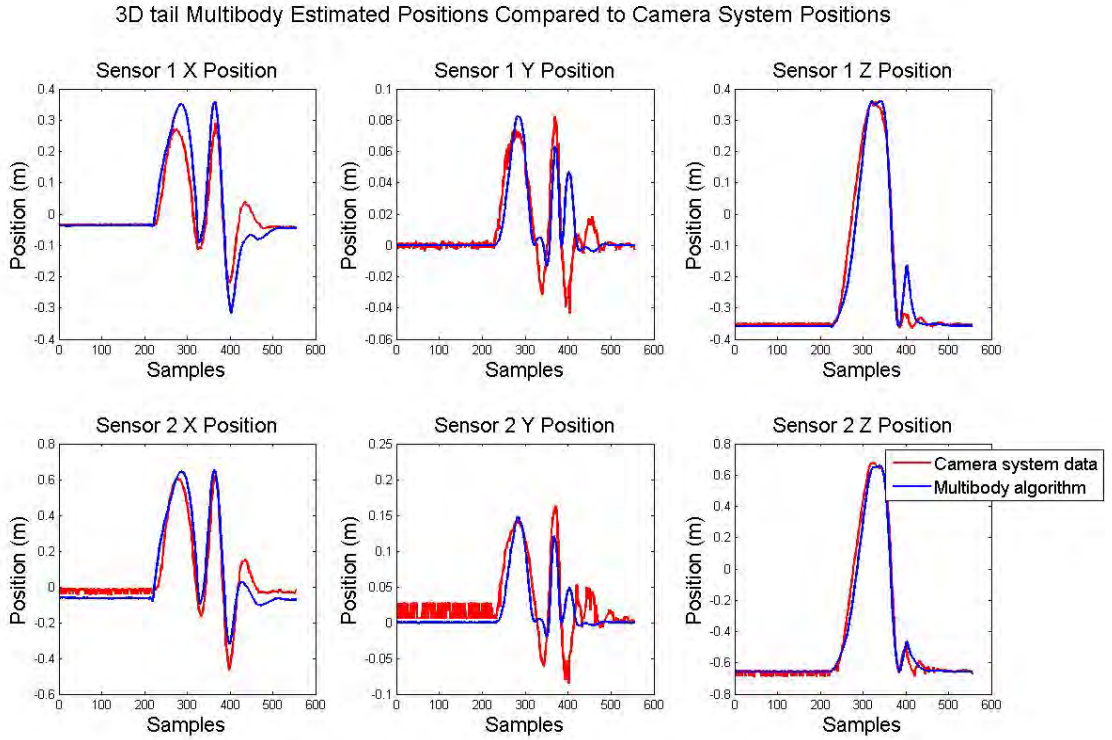
Figure 6.13: 3D tail *Multibody* estimated positions compared to the camera system data.
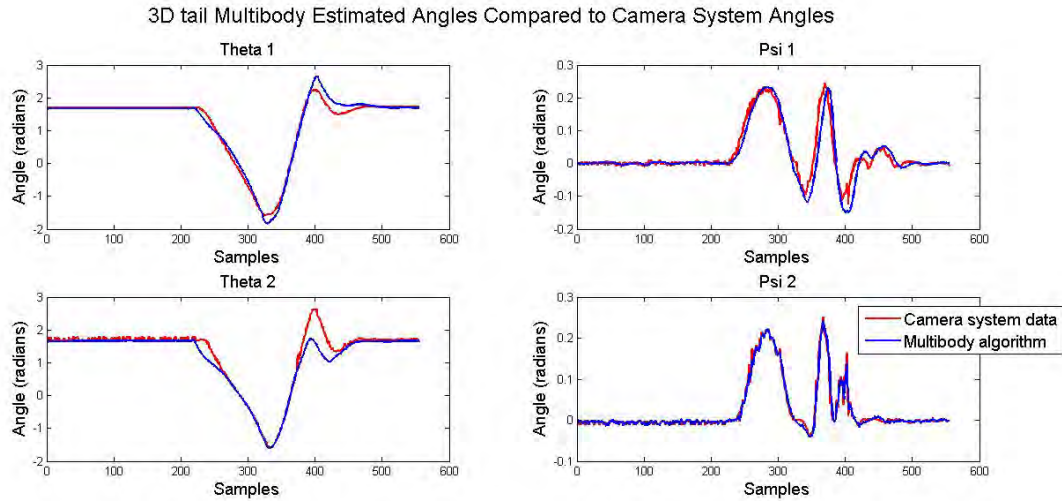
Figure 6.14: 3D tail *Multibody* estimated angles compared to the angles calculated using the camera data.
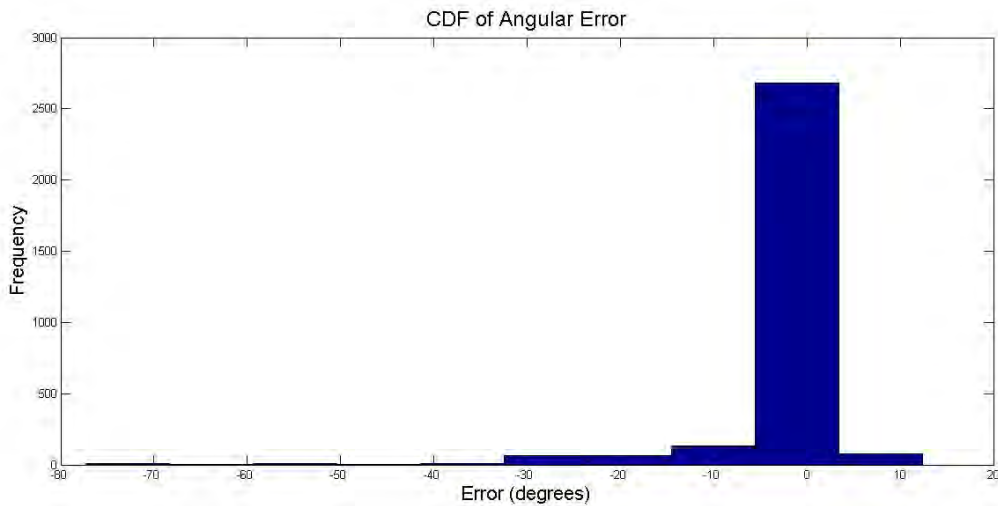
Figure 6.15: A cumulative distribution function of the angular error.

6.3.3 Algorithm 3: The *Multibody and Torque* Algorithm

The aerodynamic effect of the fur was determined to be significant, and therefore this effect was modelled using generalized forces and D'Alemberts principle as discussed in Section 2.3.2. An applied torque on the tail due to the tail muscles was also modelled along with the potential energy. See Figure 6.2 for location of the torques.

2D System

The Lagrange generalized coordinates are $\mathbf{q} = \begin{bmatrix} \theta_1 & \theta_2 \end{bmatrix}'$. The potential energy is no longer zero. It is shown in Equation 6.16 along with the kinetic energy.

$$
\begin{aligned}
T_{translation} &= \frac{1}{2}(m_1 \dot{\mathbf{p}}_{1COM}^T \dot{\mathbf{p}}_{1COM} + m_2 \dot{\mathbf{p}}_{2COM}^T \dot{\mathbf{p}}_{2COM}) \\
T_{rotation} &= \frac{1}{2}(J_1 \dot{\theta}_1^2 + J_2 \dot{\theta}_2^2) \\
U &= m_1 \mathbf{g} \mathbf{p}_{1COM} + m_2 \mathbf{g} \mathbf{p}_{2COM}
\end{aligned}
\tag{6.16}
$$

where $\mathbf{g}$ is equal to $[0, 0, 9.81]\ m/s^2$, m_i is the mass of the ith beam, J_i is the moment of inertia of the beam and $\dot{\mathbf{p}}_i$ is the velocity of the COM which is derived in Appendix A.3. Generalized forces take into account the aerodynamic effects. This is shown below.

$$
Q_{generalized\ force_j} = \sum_i^n \mathbf{f}_i \cdot \frac{\partial \mathbf{r}_i}{\partial q_j}
\tag{6.17}
$$

The aerodynamic force was modelled as a point force by integrating over the length of the beam. The point force acts at the centre of pressure (COP). The force, $\mathbf{f}$, and length, l_{COP}, (the distance from the pivot point to the COP) is derived in Appendix A.5. The position where the force acts, $\mathbf{r}$, are derived in Appendix A.10.

The vector $\mathbf{Q}$ contains the generalized forces and the applied torques due to the tail muscles and therefore Equation 6.17 gets modified to:

$$
\mathbf{Q} = \mathbf{Q}_{generalized\ forces} + \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}
\tag{6.18}
$$

where τ_1 and τ_2 are the applied torques due to the tail muscles. The pseudo code algorithm to calculate the generalized forces is in Appendix A.5.1. The equations of motion derived in Section 2.3.2 are as follows:

$$
\ddot{\mathbf{q}} = \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = \mathbf{M}(\mathbf{q})^{-1}(-\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{G}(\mathbf{q}) + \mathbf{Q})
\tag{6.19}
$$

The matrices are derived in Appendix A.4. The filter state variables, the prediction equation and update equation are shown in Equation 6.20 to Equation 6.22.

$$\mathbf{x}_k = \begin{bmatrix} \tau_1 & \tau_2 & \dot{\theta}_1 & \dot{\theta}_2 & \theta_1 & \theta_2 \end{bmatrix}' \tag{6.20}$$

$$\hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1})$$
$$\hat{\mathbf{x}}_{k|k-1} = \hat{\mathbf{x}}_{k-1|k-1} + \Delta t \dot{\hat{\mathbf{x}}}_{k-1|k-1} \tag{6.21}$$

$$\mathbf{z}_k = h(\mathbf{x}_k)$$
$$\mathbf{z}_k = \begin{bmatrix} acc_{1_X} & acc_{1_Z} & \omega_{1_Y} & acc_{2_X} & acc_{2_Z} & \omega_{2_Y} \end{bmatrix}' \tag{6.22}$$

where $\dot{\tau}_1$ and $\dot{\tau}_2$ from $\dot{\mathbf{x}}$, Equation 6.21, are equal to zero. The applied torque on the tail is unknown and therefore was modelled as a random walk [110]. The definition of a random walk process is an integration process driven by a zero mean white noise function. The measurement equations (acc_{1_X} etc.) are derived in Appendix A.6.

The results of the algorithm are shown in Figures 6.16 and 6.17. The cumulative distribution function of the angular error is shown in Figure 6.18. An RMS error of 14.16 degrees and 0.13 metres was achieved by this algorithm. The results are discussed and compared further in Section 6.5.
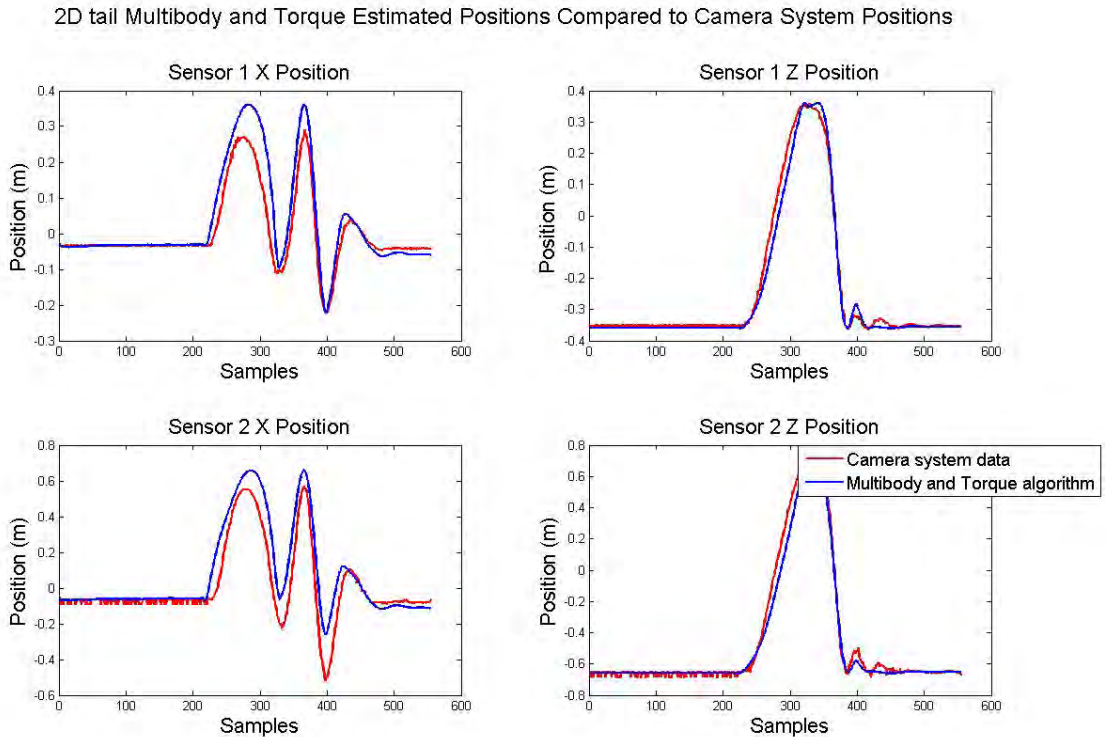
Figure 6.16: 2D tail *Multibody and torque* estimated positions compared to the camera system positions.
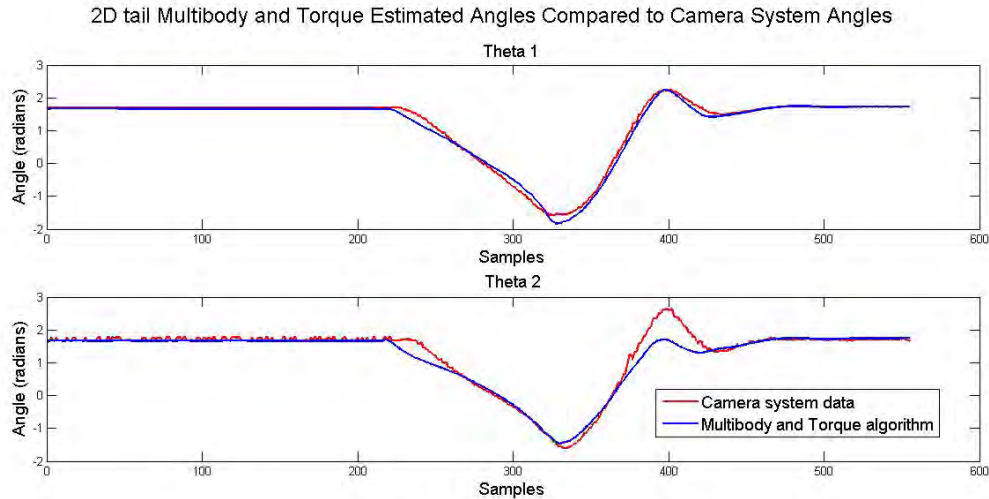
Figure 6.17: 2D tail *Multibody and torque* estimated angles compared to the angles calculated using the camera system data.
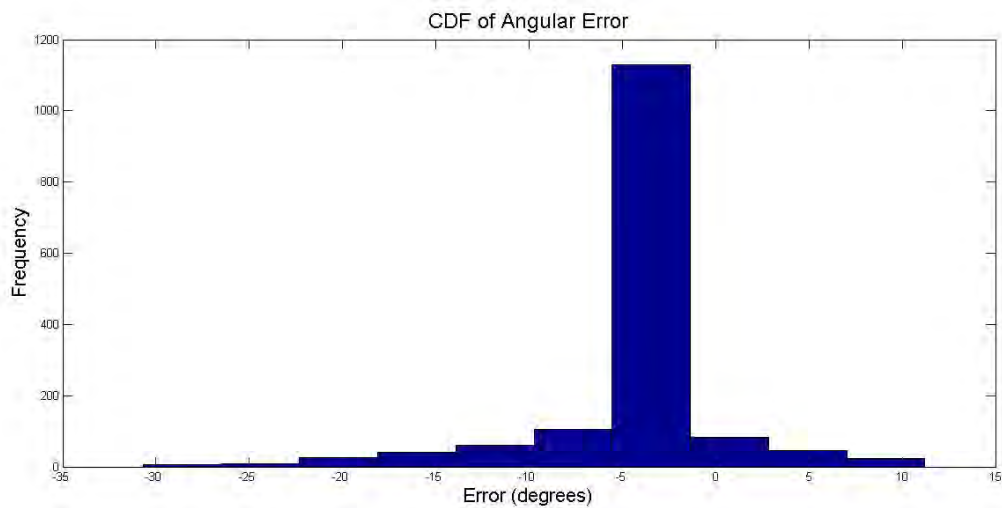
Figure 6.18: A cumulative distribution function of the angular error.

The estimated torque, τ_1, was compared to the logged motor torque and is shown in Figure 6.19. As can be seen, the logged torque and the estimated torque follow the same general shape. Readers should note that the motor torque is obtained from a current reading, and using the motor constant, it is converted to a torque. The efficiency, motor gearbox and mechanical end stops are not modelled. The torque estimate in the state estimation algorithm was modelled as a random walk.

81
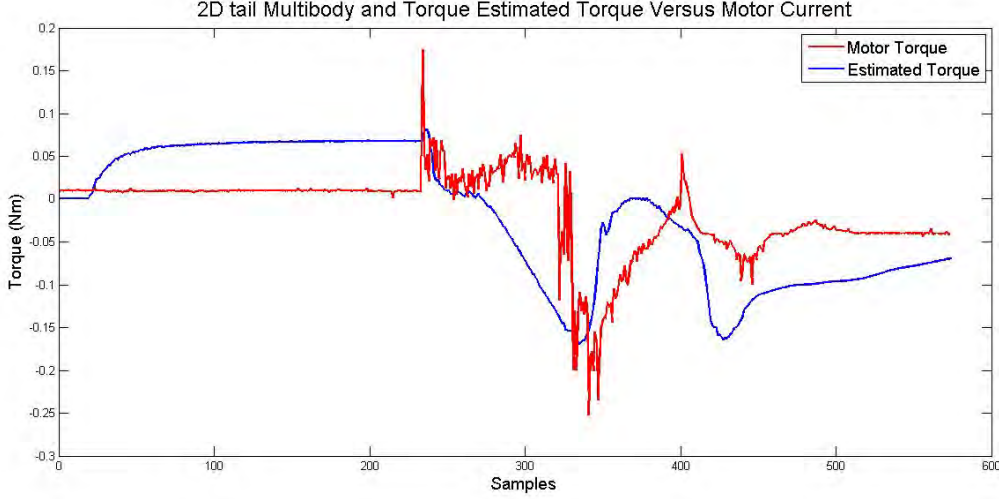
Figure 6.19: A comparison of the motor torque and the estimated torque. The graphs follow the same general trend. Readers must remember that the motor efficiency and gearbox was not modelled. The mechanical end stops were not modelled.

3D System

The Lagrange generalized forces are $\mathbf{q} = \begin{bmatrix} \theta_1 & \psi_1 & \theta_2 & \psi_2 \end{bmatrix}'$. The kinetic and potential energy are as follows:

$$T_{translation} = \frac{1}{2}(m_1 \dot{\mathbf{p}}_{1COM}^T \dot{\mathbf{p}}_{1COM} + m_2 \dot{\mathbf{p}}_{2COM}^T \dot{\mathbf{p}}_{2COM})$$

$$T_{rotational} = \frac{1}{2}(\mathbf{J}_{mid}(\mathbf{R}_{gyroItoB}(0, \theta_1, \psi_1) \begin{bmatrix} 0 \\ \dot{\theta}_1 \\ \dot{\psi}_1 \end{bmatrix})^2 + \mathbf{J}_{tip}(\mathbf{R}_{gyroItoB}(0, \theta_2, \psi_2) \begin{bmatrix} 0 \\ \dot{\theta}_2 \\ \dot{\psi}_2 \end{bmatrix})^2)$$

$$U = m_1 \mathbf{g} \mathbf{P}_{1COM} + m_2 \mathbf{g} \mathbf{P}_{2COM}$$

$$(6.23)$$

where the rotation matrices are defined in Appendix A.2, m_i is the mass of the ith beam, $\mathbf{J}_{mid}$ and $\mathbf{J}_{tip}$ contain the moment of inertia of the beam in the relevant axis (roll, pitch and yaw axis) and $\mathbf{g}$ is $[0, 0, 9.81]\ m/s^2$. The position of the COM, $\mathbf{p}_i$, and its velocity, $\dot{\mathbf{p}}_i$, are derived in Appendix A.3. Using generalized forces, Equation 6.24, the aerodynamic force was modelled as a point force by integrating over the length of the beam. The point force acts at the centre of pressure (COP).

$$Q_{generalized\ force_j} = \sum_{i}^{n} \mathbf{f}_i \cdot \frac{\partial \mathbf{r}_i}{\partial q_j} \qquad (6.24)$$

82

The **f** vector contain all the applied point forces and is derived in Appendix A.5 along with l_{COP}. The length, l_{COP}, is the distance from the pivot to the COP, where the point force is applied. The distance **r** is defined in Appendix A.10. The applied torques due to the tail muscles are appended to the $\mathbf{Q}generalized\ force_j$ matrix as follows:

$$\mathbf{Q} = \mathbf{Q}_{generalized\ force_j} + \begin{bmatrix} \tau_{1_{pitch}} \\ \tau_{1_{yaw}} \\ \tau_{2_{pitch}} \\ \tau_{2_{yaw}} \end{bmatrix} \tag{6.25}$$

The equations of motion that were derived in Section 2.3.2, are as follows:

$$\ddot{\mathbf{q}} = \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = \mathbf{M}(\mathbf{q})^{-1}(-\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{G}(\mathbf{q}) + \mathbf{Q}) \tag{6.26}$$

The pseudo code algorithm to calculate the generalized forces is in Appendix A.5.1. The calculation of $\mathbf{G}(\mathbf{q})$, $\mathbf{M}(\mathbf{q})$ and $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$ is shown in Appendix A.4 and discussed in Section 2.3.2. The Kalman filter state variables are shown in Equation 6.27, followed by the prediction and update equation.

$$\begin{aligned} \mathbf{x}_k = [&\tau_{1_{pitch}}\ \tau_{1_{yaw}}\ \tau_{2_{pitch}}\ \tau_{2_{yaw}}\ \cdots \\ &\dot{\theta}_1\ \dot{\psi}_1\ \dot{\theta}_2\ \dot{\psi}_2\ \theta_1\ \psi_1\ \theta_2\ \psi_2]' \end{aligned} \tag{6.27}$$

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= f(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}) \\ \hat{\mathbf{x}}_{k|k-1} &= \hat{\mathbf{x}}_{k-1|k-1} + \Delta t \dot{\hat{\mathbf{x}}}_{k-1|k-1} \end{aligned} \tag{6.28}$$

$$\begin{aligned} \mathbf{z}_k &= h(\mathbf{x}_k) \\ \mathbf{z}_k = [&acc_{1_X}\ acc_{1_Z}\ mag_{1_X}\ mag_{1_Y}\ mag_{1_Z}\ \omega_{1_Y}\ \omega_{1_Z}\ \cdots \\ &acc_{2_X}\ acc_{2_Z}\ mag_{2_X}\ mag_{2_Y}\ mag_{2_Z}\ \omega_{2_Y}\ \omega_{2_Z}]' \end{aligned} \tag{6.29}$$

where the measurement equations (acc_{1_X} etc.) are derived in Appendix A.6. The results of the algorithm are shown in Figures 6.20 and 6.21 with the cumulative distribution function of the angular error shown in Figure 6.22. An RMS error of 13.20 degrees and 0.12 metres was achieved by this algorithm. The results are discussed and compared further in Section 6.5.
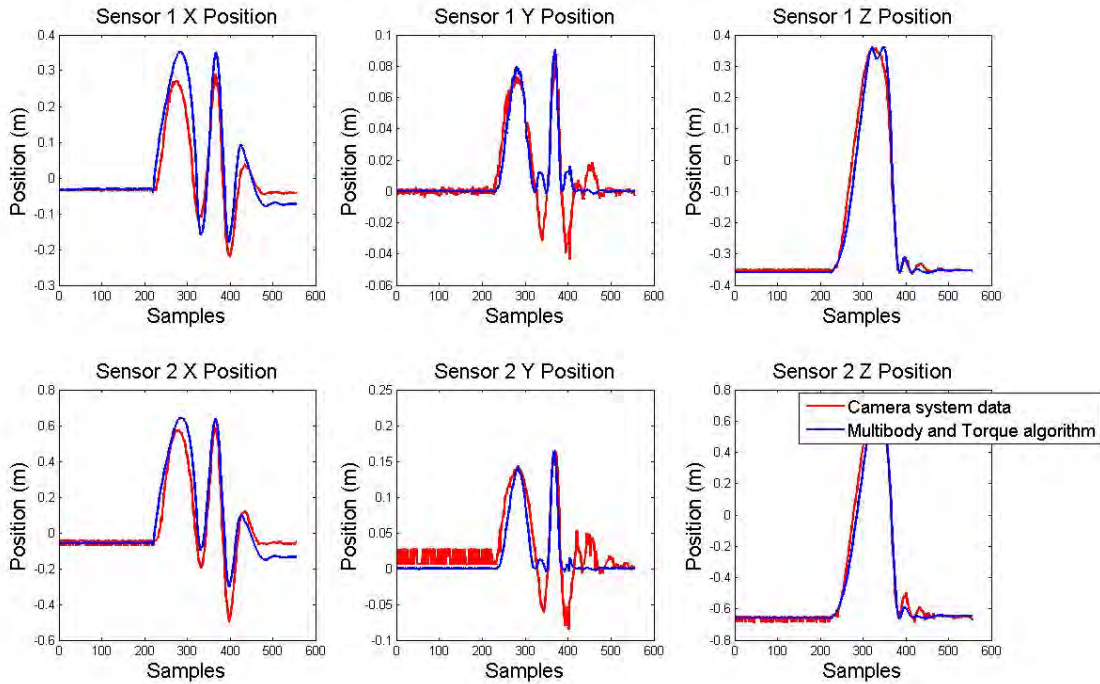
Figure 6.20: 3D tail *Multibody and torque* estimated positions compared to the camera system positions.
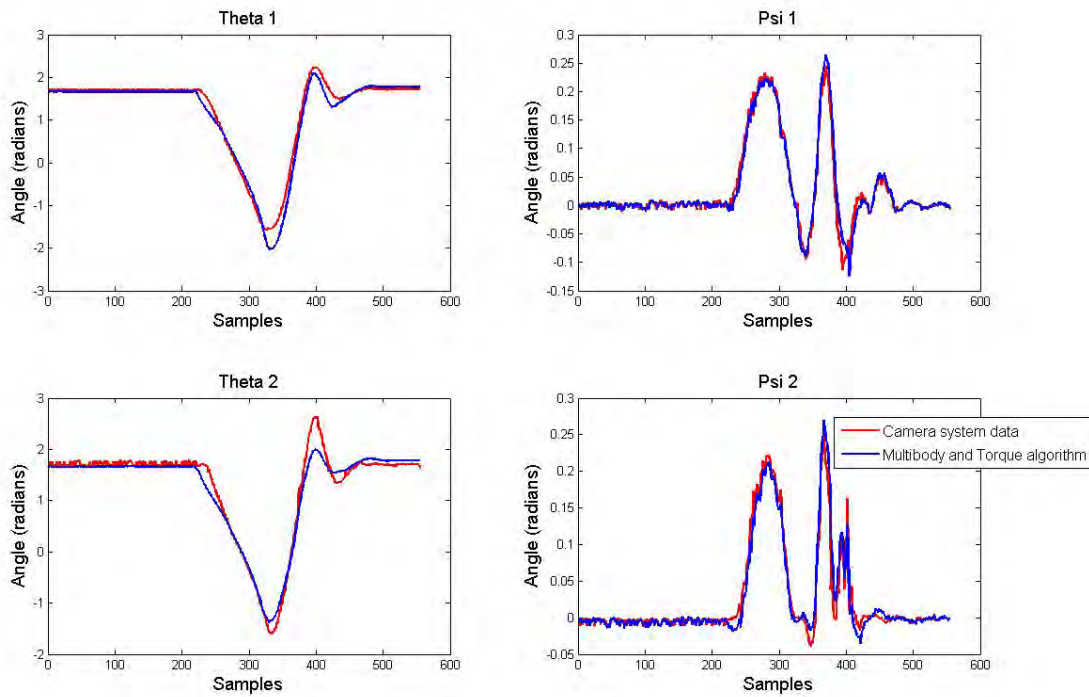
Figure 6.21: 3D tail *Multibody and torque* estimated angles compared to the angles calculated using the camera system data.

84
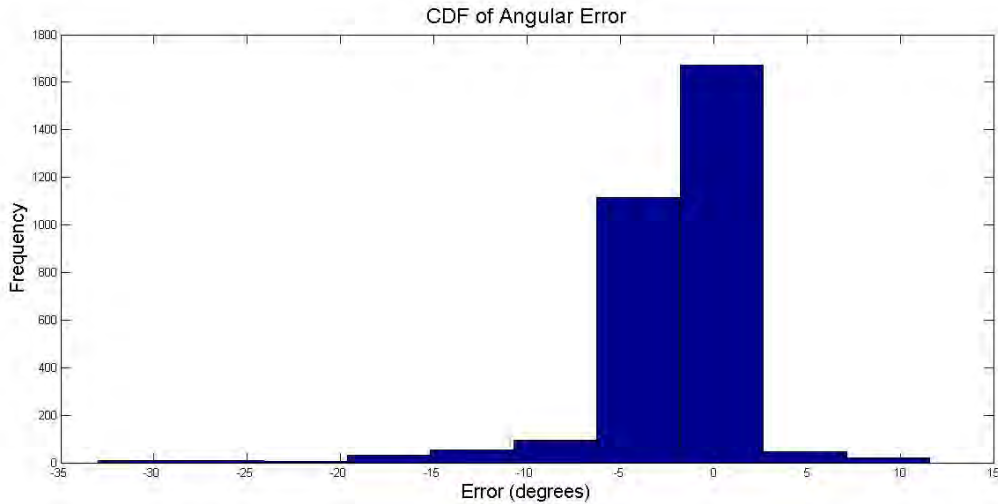
Figure 6.22: A cumulative distribution function of the angular error.

The estimated torque for $\tau_{1_{pitch}}$ was compared to the logged motor torque and is shown in Figure 6.23. As can be seen the logged torque and estimated torque follow the same general curve.
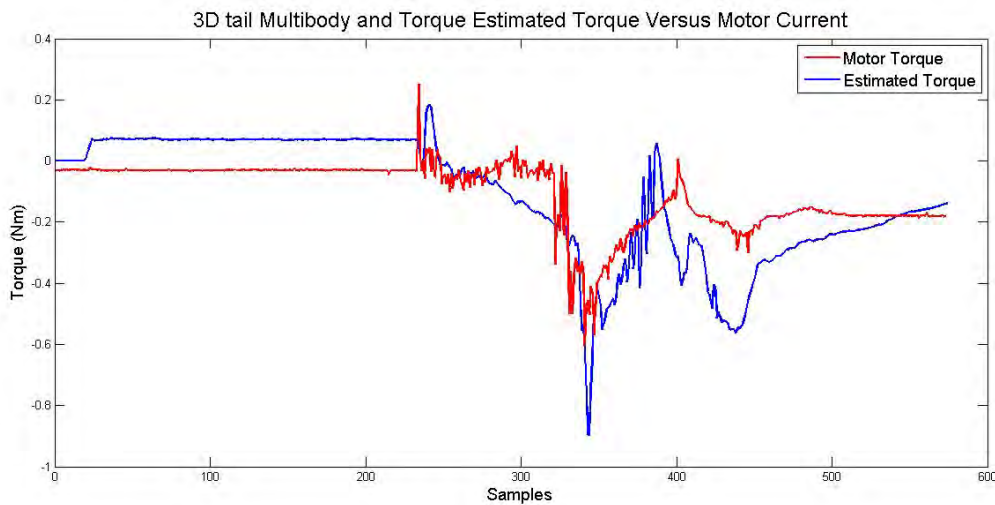
Figure 6.23: A comparison of the motor torque and the estimated torque. The graphs follow the same general trend. Readers must note that the motor efficiency and gearbox were not modelled. The mechanical end stops were not modelled.

6.4 The Tail and Spine System

A 2D spine was added to the 2D tail system described above and the algorithms were modified to take the additional dynamics into account. The 2D spine and tail system is shown in Figure 6.24a. This was then extended to the 3D case (shown in Figure 6.24b). In the 2D case, the beams can only pitch while in the 3D case, the spine beams can roll,

pitch and yaw while the tail beams can only pitch and yaw. A subscript 1 represents the first beam of the spine, subscript 2 represents the second beam of the spine, subscript 3 represents the first beam of the tail and subscript 4 represents the second beam of the tail. The roll angle of the beam is represented by ϕ, θ is the pitch angle of the beam and ψ is the yaw angle of the beam.
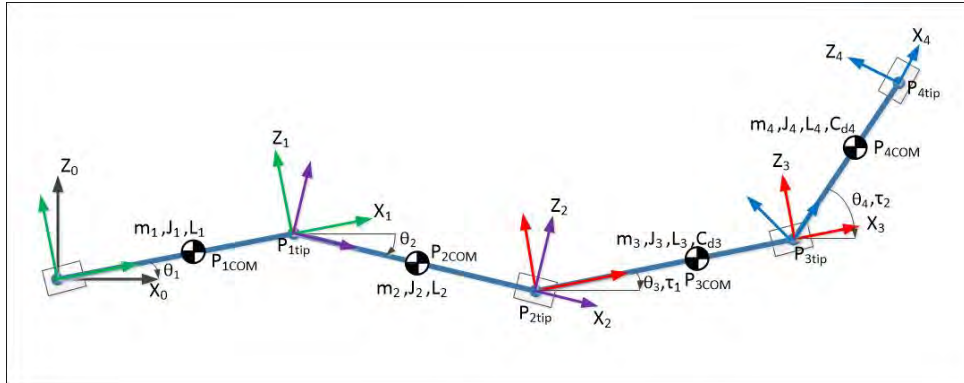
(a) 2D spine and tail system. The beams can only pitch.
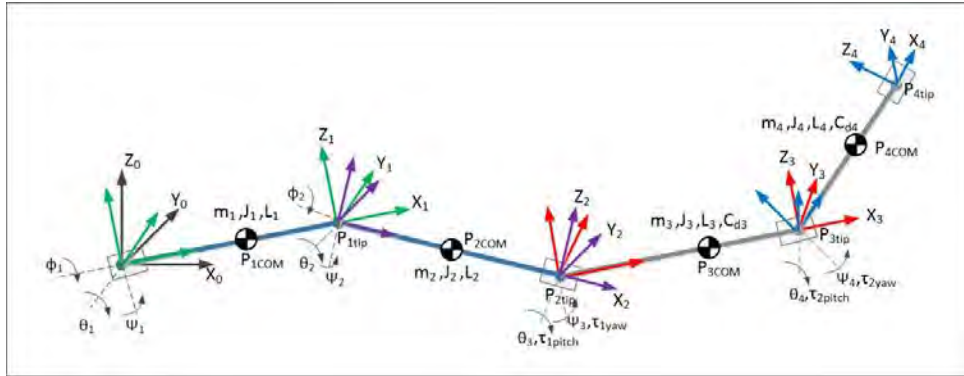
(b) 3D spine and tail system. The two spine beams can roll, pitch and yaw. Where as the two tail beams can only pitch and yaw.

Figure 6.24: System diagram for spine and tail models.

The 3D case resulted in a number of challenges. The equations of motion were too large to differentiate, the **F** matrix for the Kalman filter could not be generated and the mass matrix could not be symbolically inverted. To solve these problems, the algorithm was implemented numerically [89] [90] instead of analytically. This involves calculating the mass matrix numerically every time step, then inverting it and calculating the equations of motion. These were then numerically partially differentiated to solve for the **F** matrix for the Kalman filter. The numeric differentiation and integration is discussed in Section 2.7.1 and derived in Appendix A.7. Due to the poorer performance of the *TRIAD KF* algorithm, it was not extended to the spine.

6.4.1 Algorithm 1: The *Multibody* Algorithm

The *Multibody* algorithm utilizes the model of the dynamics of the system to predict the future states. These states then get updated using the measurement equations. The dynamics of the system are first modelled using Lagrange dynamics and then applied to the filter algorithms. The aerodynamic effect, potential energy and applied tail torques are set to zero as it is assumed that the applied torque is equal to, and cancels out, the aerodynamic and potential energy effect.

2D System

The generalized coordinates for the Lagrange dynamics system are $\mathbf{q} = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 & \theta_4 \end{bmatrix}'$ and the kinetic energy is:

$$
\begin{aligned}
T_{translation} =& \frac{1}{2}(m_1 \dot{\mathbf{p}}_{1_{COM}}^T \dot{\mathbf{p}}_{1_{COM}} + m_2 \dot{\mathbf{p}}_{2_{COM}}^T \dot{\mathbf{p}}_{2_{COM}} + m_3 \dot{\mathbf{p}}_{3_{COM}}^T \dot{\mathbf{p}}_{3_{COM}} + m_4 \dot{\mathbf{p}}_{4_{COM}}^T \dot{\mathbf{p}}_{4_{COM}}) \\
T_{rotational} =& \frac{1}{2}(J_1 \dot{\theta}_1^2 + J_2 \dot{\theta}_2^2 + J_3 \dot{\theta}_3^2 + J_4 \dot{\theta}_4^2)
\end{aligned}
$$

(6.30)

where m_i is the mass of the ith beam and J_i is the moment of inertia of the beam with respect to the pivot point. The velocity, $\dot{\mathbf{p}}_i$, terms are derived in Appendix A.3. The potential energy of the system, the applied torque from the tail muscles and the aerodynamic effects are set to zero because it is assumed the applied torque cancels out the potential energy and the aerodynamic torques. The equations of motion are as follows:

$$
\ddot{\mathbf{q}} = \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \\ \ddot{\theta}_4 \end{bmatrix} = \mathbf{M}(\mathbf{q})^{-1}(-\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}})
$$

(6.31)

The calculation of $\mathbf{M}(\mathbf{q})$ and $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$ is shown in Appendix A.4 and discussed in Section 2.3.2. The state variables of the Kalman filter are shown in Equation 6.32 followed by the prediction and update equation.

$$
\mathbf{x}_k = \begin{bmatrix} \dot{\theta}_1 & \dot{\theta}_2 & \dot{\theta}_3 & \dot{\theta}_4 & \theta_1 & \theta_2 & \theta_3 & \theta_4 \end{bmatrix}'
$$

(6.32)

$$\hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1})$$
$$\hat{\mathbf{x}}_{k|k-1} = \hat{\mathbf{x}}_{k-1|k-1} + \Delta t \hat{\dot{\mathbf{x}}}_{k-1|k-1}$$

(6.33)

$$\mathbf{z}_k = h(\mathbf{x}_k)$$
$$\mathbf{z}_k = [acc_{1_X} \ acc_{1_Z} \ \omega_{1_Y} \ acc_{2_X} \ acc_{2_Z} \ \omega_{2_Y} \dots$$
$$acc_{3_X} \ acc_{3_Z} \ \omega_{3_Y} \ acc_{4_X} \ acc_{4_Z} \ \omega_{4_Y}]'$$

(6.34)

where the measurement equations (acc_{1_X} etc.) are derived in Appendix A.6. The results of the algorithm are shown in Figures 6.25 and 6.26. The cumulative distribution function of the angular error is shown in Figure 6.27. The cost functions of the algorithm resulted in an RMS error of 14.84 degrees and 0.15 metres. The results are discussed and compared further in Section 6.5.
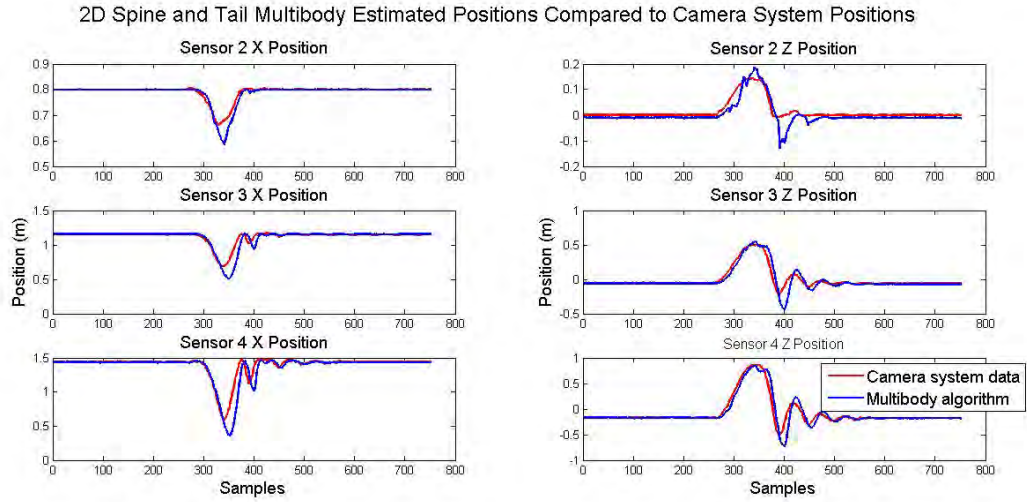
Figure 6.25: 2D spine and tail *Multibody* estimated positions compared to camera system positions.
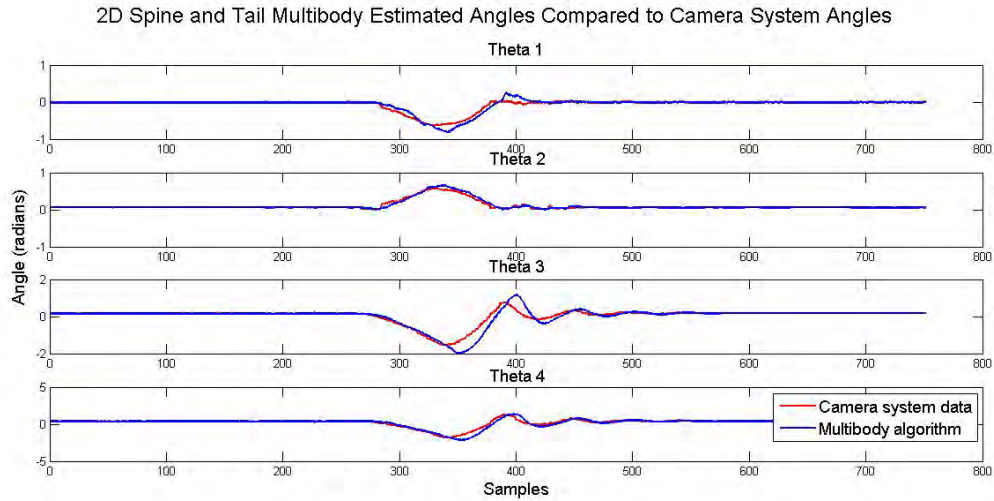
Figure 6.26: 2D spine and tail *Multibody* estimated angles compared to angles calculated using the camera system data.
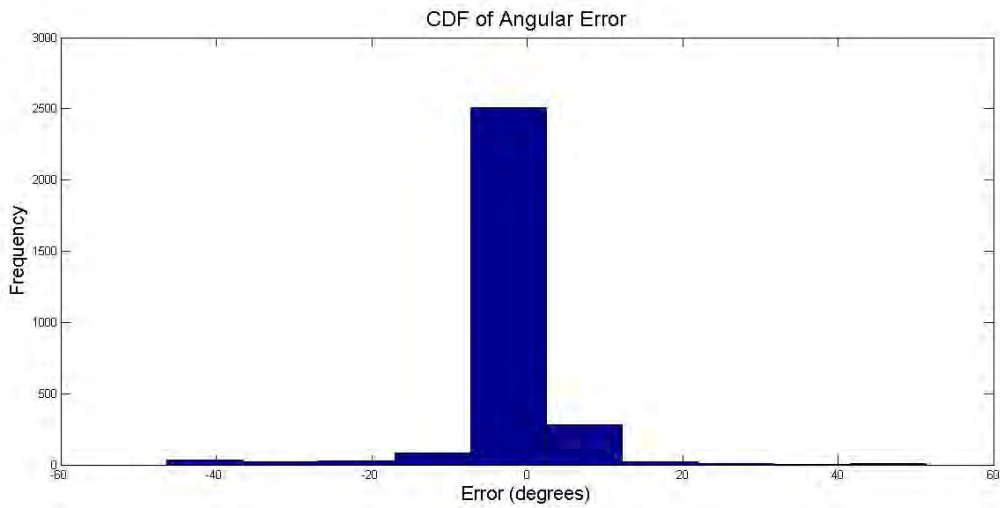
Figure 6.27: A cumulative distribution function of the angular error.

3D System

The generalized coordinates for the Lagrange dynamics are
$\mathbf{q} = \begin{bmatrix} \phi_1 & \theta_1 & \psi_1 & \phi_2 & \theta_2 & \psi_2 & \theta_3 & \psi_3 & \theta_4 & \psi_4 \end{bmatrix}'$. The kinetic energy is

$$T_{translation} = \frac{1}{2}(m_1 \dot{\mathbf{p}}_{1COM}^T \dot{\mathbf{p}}_{1COM} + m_2 \dot{\mathbf{p}}_{2COM}^T \dot{\mathbf{p}}_{2COM} + m_3 \dot{\mathbf{p}}_{3COM}^T \dot{\mathbf{p}}_{3COM} + m_4 \dot{\mathbf{p}}_{4COM}^T \dot{\mathbf{p}}_{4COM})$$

$$T_{rotational} = \frac{1}{2}(\mathbf{J}_1(\mathbf{R}_{gyroItoB}(\phi_1, \theta_1, \psi_1) \begin{bmatrix} \dot{\phi}_1 \\ \dot{\theta}_1 \\ \dot{\phi}_1 \end{bmatrix})^2 + \mathbf{J}_2(\mathbf{R}_{gyroItoB}(\phi_2, \theta_2, \psi_2) \begin{bmatrix} \dot{\phi}_2 \\ \dot{\theta}_2 \\ \dot{\phi}_2 \end{bmatrix})^2 \dots$$

$$\dots + \mathbf{J}_3(\mathbf{R}_{gyroItoB}(0, \theta_3, \psi_3) \begin{bmatrix} 0 \\ \dot{\theta}_3 \\ \dot{\phi}_3 \end{bmatrix})^2 + \mathbf{J}_4(\mathbf{R}_{gyroItoB}(0, \theta_4, \psi_4) \begin{bmatrix} 0 \\ \dot{\theta}_4 \\ \dot{\phi}_4 \end{bmatrix})^2)$$

$$(6.35)$$

where m_i is the mass of the ith beam and J_i is a matrix containing the moment of inertia about the pivot point of the beam. $\dot{\mathbf{p}}_i$ is derived in Appendix A.3. The rotational matrices are derived in Appendix A.2. The potential energy and the generalized forces are zero. The equations of motion are as follows:

$$\ddot{\mathbf{q}} = \begin{bmatrix} \ddot{\phi}_1 & \ddot{\theta}_1 & \ddot{\psi}_1 & \ddot{\phi}_2 & \ddot{\theta}_2 & \ddot{\psi}_2 & \ddot{\theta}_3 & \ddot{\psi}_3 & \ddot{\theta}_4 & \ddot{\psi}_4 \end{bmatrix}' = \mathbf{M}(\mathbf{q})^{-1}(-\mathbf{C}\dot{\mathbf{q}}) \qquad (6.36)$$

The calculation of $\mathbf{M}(\mathbf{q})$ and $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$ is shown in Appendix A.4. The state variables of the Kalman filter are shown in Equation 6.37 followed by the prediction and update equations.

$$\mathbf{x}_k = [\dot{\phi}_1\ \dot{\theta}_1\ \dot{\psi}_1\ \dot{\phi}_2\ \dot{\theta}_2\ \dot{\psi}_2\ \dot{\theta}_3\dot{\psi}_3\ \dot{\theta}_4\ \dot{\psi}_4 ...$$
$$\phi_1\ \theta_1\ \psi_1\ \phi_2\ \theta_2\ \psi_2\ \theta_3\ \psi_3\ \theta_4\ \psi_4]' \qquad (6.37)$$

$$\hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1})$$
$$\hat{\mathbf{x}}_{k|k-1} = \hat{\mathbf{x}}_{k-1|k-1} + \Delta t \dot{\hat{\mathbf{x}}}_{k-1|k-1} \qquad (6.38)$$

$$\mathbf{z}_k = h(\mathbf{x}_k)$$
$$\mathbf{z}_k = [acc_{1_X}\ acc_{1_Z}\ mag_{1_X}\ mag_{1_Y}\ mag_{1_Z}\ \omega_{1_Y}\ \omega_{1_Z} \dots$$
$$acc_{2_X}\ acc_{2_Z}\ mag_{2_X}\ mag_{2_Y}\ mag_{2_Z}\ \omega_{2_Y}\ \omega_{2_Z} \dots \qquad (6.39)$$
$$acc_{3_X}\ acc_{3_Z}\ mag_{3_X}\ mag_{3_Y}\ mag_{3_Z}\ \omega_{3_Y}\ \omega_{3_Z} \dots$$
$$acc_{4_X}\ acc_{4_Z}\ mag_{4_X}\ mag_{4_Y}\ mag_{4_Z}\ \omega_{4_Y}\ \omega_{4_Z}]'$$

where the measurement equations (acc_{1_X} etc.) are derived in Appendix A.6. The results of the algorithm are shown in Figure 6.28 with the cumulative distribution function of

the position error shown in Figure 6.29. Due to the limitations of the camera system (the roll angle cannot be calculated due to the camera system tracking a single point) the RMS error, of the angles, could not be calculated. The angles are shown in Appendix A.9. The position cost function equated to an RMS error of 0.12 metres. The results are discussed and compared further in Section 6.5.
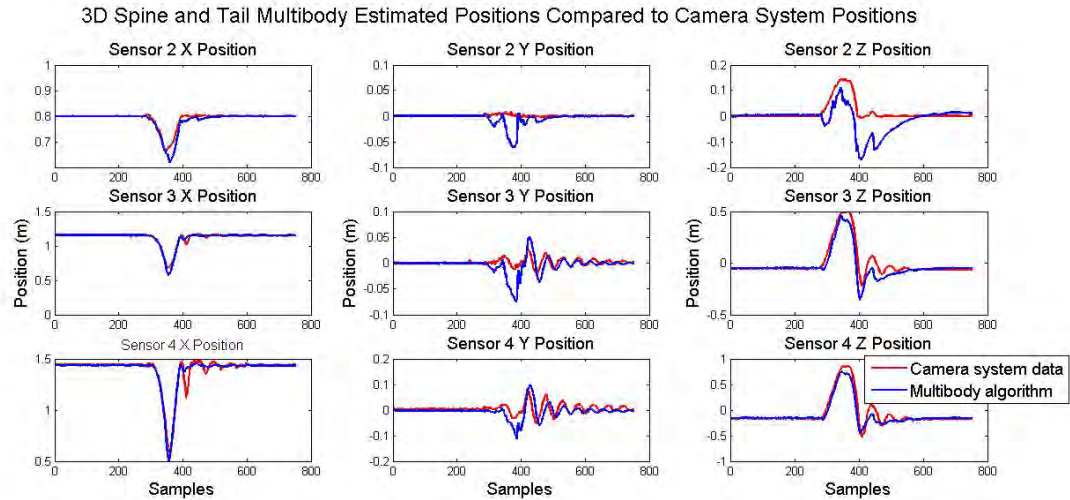
Figure 6.28: 3D spine and tail *Multibody* estimated positions compared to the camera system positions.
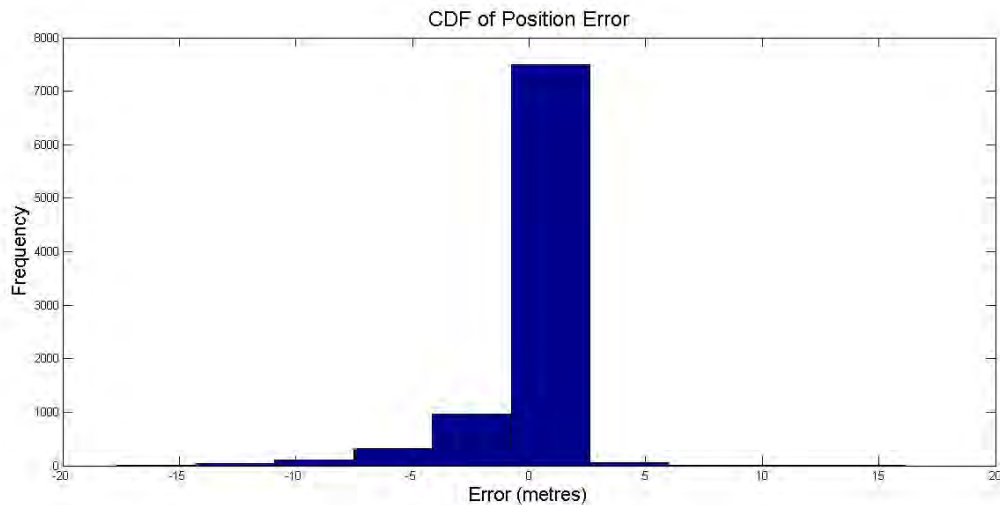
Figure 6.29: A cumulative distribution function of the angular error.

6.4.2 Algorithm 2: The *Multibody and Torque* Algorithm

The aerodynamic effect of the fur was determined to be significant and was modelled using generalized forces and D'Alemberts principle. An applied torque on the tail due to the tail muscles was also modelled and so was the potential energy. Due to the fact that only the tail torque was of interest, the potential energy, aerodynamic force and applied torque for the spine were set to zero. Figure 6.24 shows where the torques act.

2D System

The generalized coordinates for the Lagrange model are $\mathbf{q} = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 & \theta_4 \end{bmatrix}'$. The kinetic and potential are as follows:

$$
\begin{aligned}
T_{translation} &= \frac{1}{2}(m_1\dot{\mathbf{p}}_{1COM}^T\dot{\mathbf{p}}_{1COM} + m_2\dot{\mathbf{p}}_{2COM}^T\dot{\mathbf{p}}_{2COM} + m_3\dot{\mathbf{p}}_{3COM}^T\dot{\mathbf{p}}_{3COM} + m_4\dot{\mathbf{p}}_{4COM}^T\dot{\mathbf{p}}_{4COM}) \\
T_{rotational} &= \frac{1}{2}(J_1\dot{\theta}_1^2 + J_2\dot{\theta}_2^2 + J_3\dot{\theta}_3^2 + J_4\dot{\theta}_4^2) \\
U &= m_3\mathbf{g}\mathbf{p}_{3COM} + m_4\mathbf{g}\mathbf{p}_{4COM}
\end{aligned}
$$

$$(6.40)$$

where m_i is the mass of the ith beam, J_i is the moment of inertia of the beam and $\mathbf{g}$ is $[0, 0, 9.81]$ m/s^2. The position, $\mathbf{p}_i$, and velocity, $\dot{\mathbf{p}}_i$, are derived in Appendix A.3. The reason why there is no potential energy for the spine is because the spine torque is unknown and not modelled. Therefore we assume that the applied torque is equal to the potential energy of the spine and the aerodynamic effect on the spine. The focus of this research is to calculate the torque that is transferred from the tail onto the body. Using the principle of generalized forces, Equation 6.41, the aerodynamic force was modelled as a point force by integrating over the length of the beam.

$$
Q_{generalized\ force_j} = \sum_i^n \mathbf{f}_i \cdot \frac{\partial \mathbf{r}_i}{\partial q_j}
$$

$$(6.41)$$

The aerodynamic point force acts at the COP. The force, $\mathbf{f}$, is derived in Appendix A.5 along with l_{COP}. The position where the force acts, $\mathbf{r}$, is derived in Appendix A.10. The $\mathbf{Q}$generalized force matrix is appended with the applied torques due to the tail muscles as follows:

$$
\mathbf{Q} = \mathbf{Q}generalized\ force + \begin{bmatrix} 0 \\ 0 \\ \tau_1 \\ \tau_2 \end{bmatrix}
$$

$$(6.42)$$

The equations of motion derived in Section 2.3.2 are as follows:

$$
\ddot{\mathbf{q}} = \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \\ \ddot{\theta}_4 \end{bmatrix} = \mathbf{M}(\mathbf{q})^{-1}(-\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{G}(\mathbf{q}) + \mathbf{Q})
$$

$$(6.43)$$

The pseudo code algorithm to calculate the generalized forces is in Appendix A.5.1. The calculation of $\mathbf{G}(\mathbf{q})$, $\mathbf{M}(\mathbf{q})$ and $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$ is shown in Appendix A.4. The state variables of the Kalman filter are shown in Equation 6.44, followed by the prediction and update equations.

$$\mathbf{x}_k = \begin{bmatrix} \tau_1 & \tau_2 & \dot{\theta}_1 & \dot{\theta}_2 & \dot{\theta}_3 & \dot{\theta}_4 & \theta_1 & \theta_2 & \theta_3 & \theta_4 \end{bmatrix}' \tag{6.44}$$

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= f\big(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}\big) \\ \hat{\mathbf{x}}_{k|k-1} &= \hat{\mathbf{x}}_{k-1|k-1} + \Delta t \dot{\hat{\mathbf{x}}}_{k-1|k-1} \end{aligned} \tag{6.45}$$

$$\begin{aligned} \mathbf{z}_k &= h(\mathbf{x}_k) \\ \mathbf{z}_k &= \big[acc_{1_X} \ acc_{1_Z} \ \omega_{1_Y} \ acc_{2_X} \ acc_{2_Z} \ \omega_{2_Y} \dots \\ & \quad acc_{3_X} \ acc_{3_Z} \ \omega_{3_Y} \ acc_{4_X} \ acc_{4_Z} \ \omega_{4_Y} \big]' \end{aligned} \tag{6.46}$$

where the measurement equations (acc_{1_X} etc.) are derived in Appendix A.6. The results of the algorithm are shown in Figures 6.30 to 6.32. A cumulative distribution function of the angular error is shown in Figure 6.33. The angle cost function equated to an RMS error of 14.94 while the angle cost function equated to an RMS error of 0.16 metres. The results are discussed and compared further in Section 6.5.
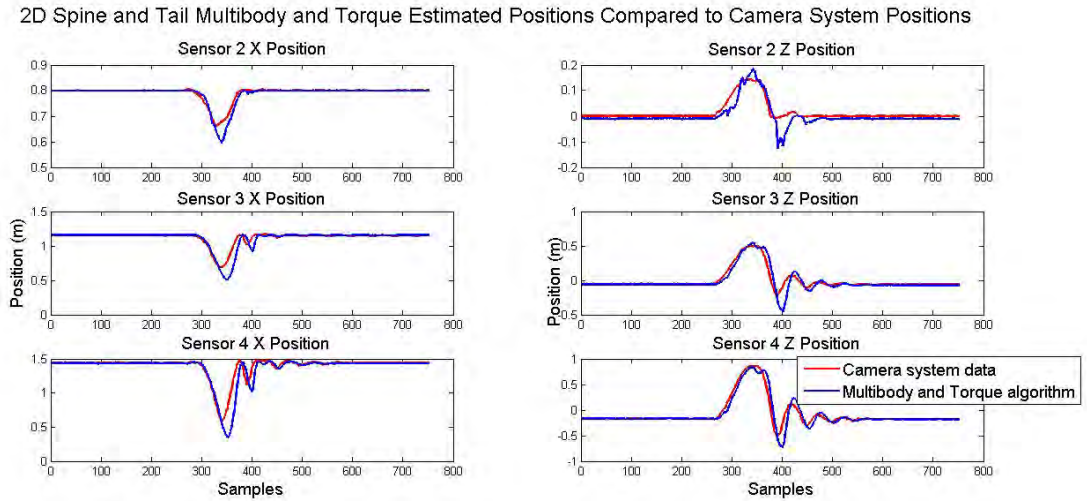
Figure 6.30: 2D spine and tail *Multibody and torque* estimated positions compared to the camera system positions.
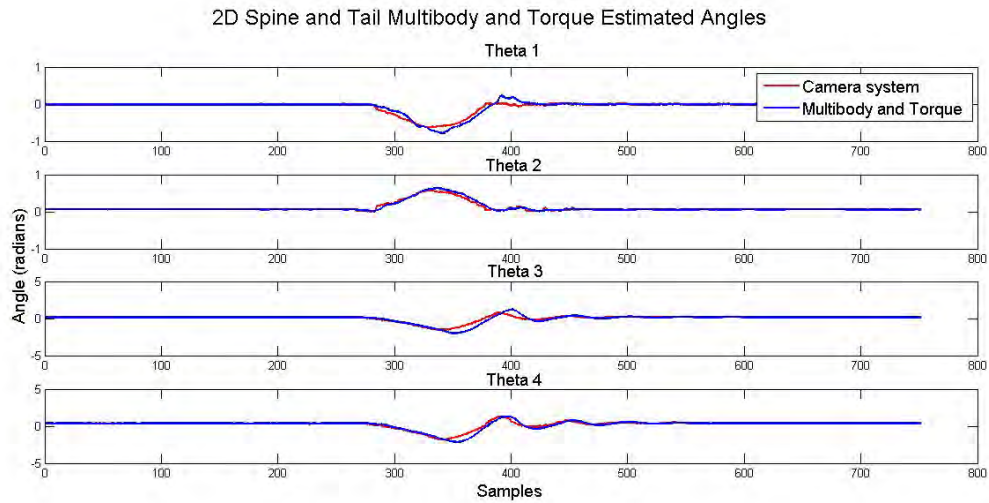
Figure 6.31: 2D spine and tail *Multibody and torque* estimated angles compared to the angles calculated using the camera system data.
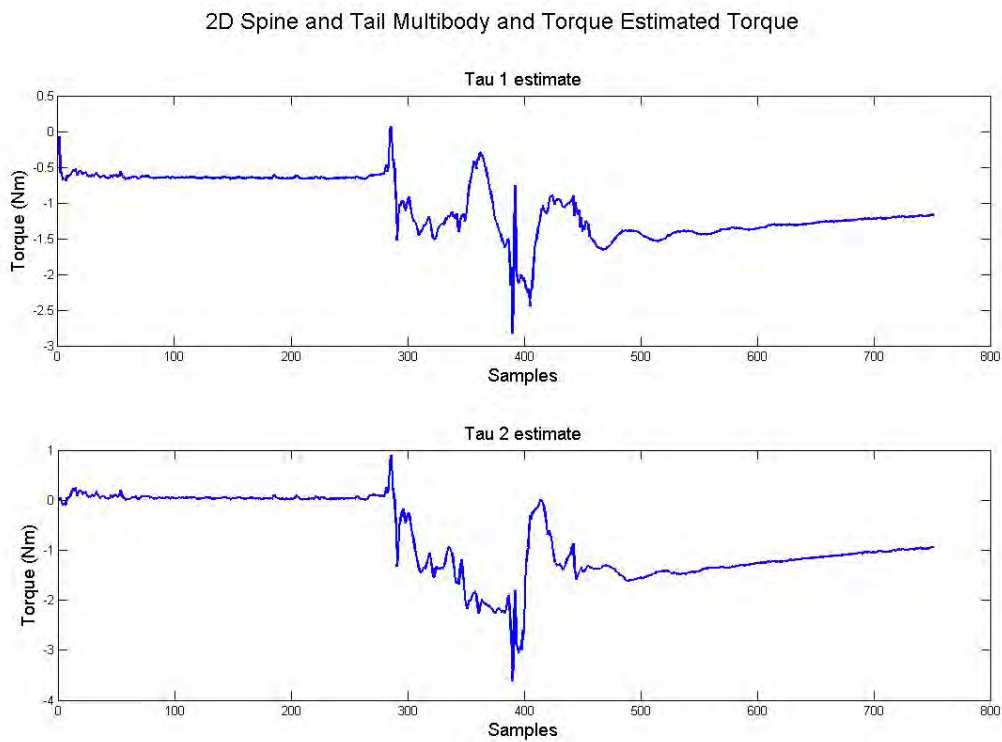
Figure 6.32: 2D spine and tail *Multibody and torque* estimated tail torques.

94
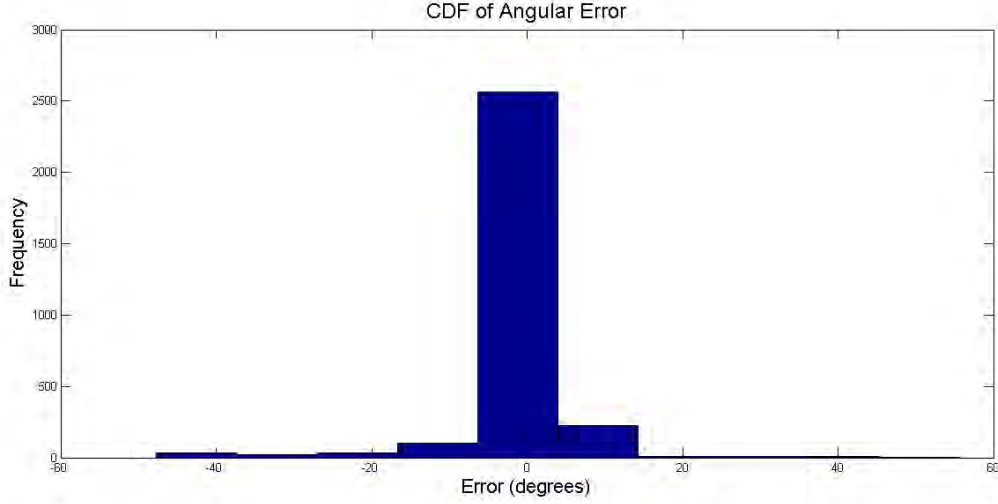
Figure 6.33: A cumulative distribution function of the angular error.

3D System

The generalized coordinates for the Lagrange model are $\mathbf{q} = [\phi_1\ \theta_1\ \psi_1\ \phi_2\ \theta_2\ \psi_2\ \theta_3\ \psi_3\ \theta_4\ \psi_4]'$. The Kinetic and potential energy of the system are as follows:

$$
\begin{aligned}
T_{translation} =&\frac{1}{2}(m_1\dot{\mathbf{p}}_{1_{COM}}^T \dot{\mathbf{p}}_{1_{COM}} + m_2\dot{\mathbf{p}}_{2_{COM}}^T \dot{\mathbf{p}}_{2_{COM}} + m_3\dot{\mathbf{p}}_{3_{COM}}^T \dot{\mathbf{p}}_{3_{COM}} + m_4\dot{\mathbf{p}}_{4_{COM}}^T \dot{\mathbf{p}}_{4_{COM}}) \\
T_{rotational} =& \frac{1}{2}(\mathbf{J}_1(\mathbf{R}_{gyroItoB}(\phi_1,\theta_1,\psi_1) \begin{bmatrix} \dot{\phi}_1 \\ \dot{\theta}_1 \\ \dot{\phi}_1 \end{bmatrix})^2 + \mathbf{J}_2(\mathbf{R}_{gyroItoB}(\phi_2,\theta_2,\psi_2) \begin{bmatrix} \dot{\phi}_2 \\ \dot{\theta}_2 \\ \dot{\phi}_2 \end{bmatrix})^2 \ldots \\
& \cdots + \mathbf{J}_3(\mathbf{R}_{gyroItoB}(0,\theta_3,\psi_3) \begin{bmatrix} 0 \\ \dot{\theta}_3 \\ \dot{\phi}_3 \end{bmatrix})^2 + \mathbf{J}_4(\mathbf{R}_{gyroItoB}(0,\theta_4,\psi_4) \begin{bmatrix} 0 \\ \dot{\theta}_4 \\ \dot{\phi}_4 \end{bmatrix})^2) \\
U =&m_3\mathbf{g}\mathbf{p}_{3_{COM}} + m_4\mathbf{g}\mathbf{p}_{4_{COM}}
\end{aligned}
$$

$$(6.47)$$

where m_i is the mass of the ith beam, $\mathbf{g}$ is $[0,\ 0,\ 9.81]\ m/s^2$ and the rotation matrices are derived in Appendix A.2. Using the principle of generalized forces, Equation 6.48, the aerodynamic force was modelled as a point force by integrating over the length of the beam.

$$Q_{generalized\ force_j} = \sum_i^n \mathbf{f}_i \cdot \frac{\partial \mathbf{r}_i}{\partial q_j} \tag{6.48}$$

The point force acts at the COP. The force, $\mathbf{f}$, is defined in Appendix A.5 along with l_{COP}. The $\mathbf{f}$ vector contains all the aerodynamic point forces. The rotation matrices are defined in Appendix A.2. Where $\mathbf{r}$ is defined in Appendix A.10.

The $\mathbf{Q}_{generalized\ force}$ matrix is appended with the applied torques due to the tail muscles as follows:

$$\mathbf{Q} = \mathbf{Q}_{generalized\ force} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \tau_{1_{pitch}} & \tau_{1_{yaw}} & \tau_{2_{pitch}} & \tau_{2_{yaw}} \end{bmatrix}' \qquad (6.49)$$

The equations of motion derived in Section 2.3.2 are as follows:

$$\ddot{\mathbf{q}} = \begin{bmatrix} \ddot{\phi}_1 & \ddot{\theta}_1 & \ddot{\psi}_1 & \ddot{\phi}_2 & \ddot{\theta}_2 & \ddot{\psi}_2 & \ddot{\theta}_3 & \ddot{\psi}_3 & \ddot{\theta}_4 & \ddot{\psi}_4 \end{bmatrix}' = \mathbf{M}(\mathbf{q})^{-1}(-\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{G}(\mathbf{q}) + \mathbf{Q}) \qquad (6.50)$$

The pseudo code algorithm to calculate the generalized forces is in Appendix A.5.1. The calculation of $\mathbf{G}(\mathbf{q})$, $\mathbf{M}(\mathbf{q})$ and $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$ is shown in Appendix A.4. The state variables of the Kalman filter are shown in Equation 6.51 followed by the prediction and update equations.

$$\begin{aligned} \mathbf{x}_k = [& \tau_{1_{pitch}} \ \tau_{1_{yaw}} \ \tau_{2_{pitch}} \ \tau_{2_{yaw}} \ \dot{\phi}_1 \ \dot{\theta}_1 ... \\ & \dot{\psi}_1 \ \dot{\phi}_2 \ \dot{\theta}_2 \ \dot{\psi}_2 \ \dot{\theta}_3 \ \dot{\psi}_3 \ \dot{\theta}_4 \ \dot{\psi}_4 \ \phi_1 ... \\ & \theta_1 \ \psi_1 \ \phi_2 \ \theta_2 \ \psi_2 \ \theta_3 \ \psi_3 \ \theta_4 \ \psi_4]' \end{aligned} \qquad (6.51)$$

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= f(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}) \\ \hat{\mathbf{x}}_{k|k-1} &= \hat{\mathbf{x}}_{k-1|k-1} + \Delta t \dot{\hat{\mathbf{x}}}_{k-1|k-1} \end{aligned} \qquad (6.52)$$

$$\begin{aligned} \mathbf{z}_k &= h(\mathbf{x}_k) \\ \mathbf{z}_k = [& acc_{1_X} \ acc_{1_Z} \ mag_{1_X} \ mag_{1_Y} \ mag_{1_Z} \ \omega_{1_Y} \ \omega_{1_Z} \ ... \\ & acc_{2_X} \ acc_{2_Z} \ mag_{2_X} \ mag_{2_Y} \ mag_{2_Z} \ \omega_{2_Y} \ \omega_{2_Z} \ ... \\ & acc_{3_X} \ acc_{3_Z} \ mag_{3_X} \ mag_{3_Y} \ mag_{3_Z} \ \omega_{3_Y} \ \omega_{3_Z} \ ... \\ & acc_{4_X} \ acc_{4_Z} \ mag_{4_X} \ mag_{4_Y} \ mag_{4_Z} \ \omega_{4_Y} \ \omega_{4_Z}]' \end{aligned} \qquad (6.53)$$

where the measurement equations (acc_{1_X} etc.) are derived in Appendix A.6. The results of the algorithm are shown in Figure 6.34 and 6.35. A cumulative distribution function of the position error is shown in Figure 6.36. Due to the limitations of the camera system

(the roll angle cannot be calculated due to the camera system tracking a single point) the RMS error, of the angles, could not be calculated. The RMS error equated to 0.11 metres for the above algorithm. The results are discussed and compared further in Section 6.5.
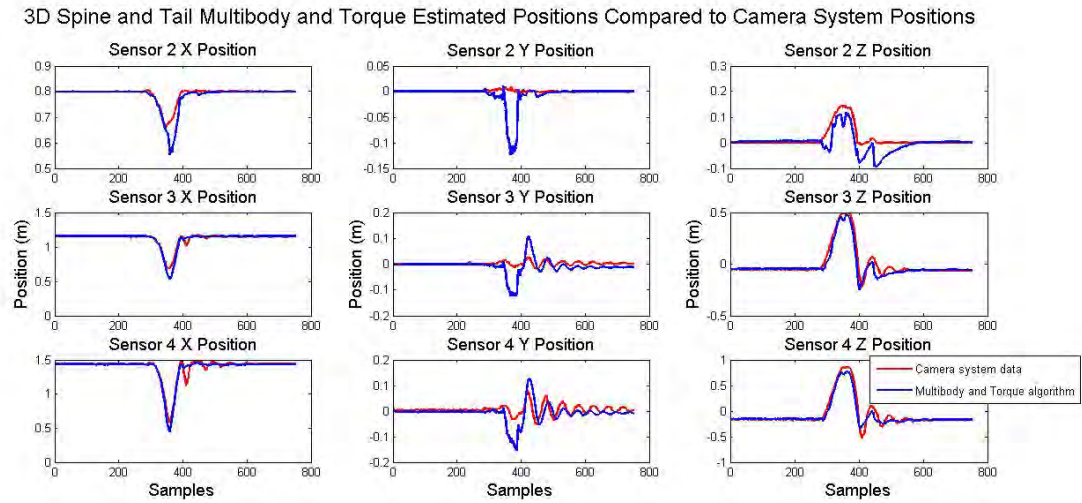
Figure 6.34: 3D spine and tail *Multibody and torque* estimated positions compared to the camera system positions.
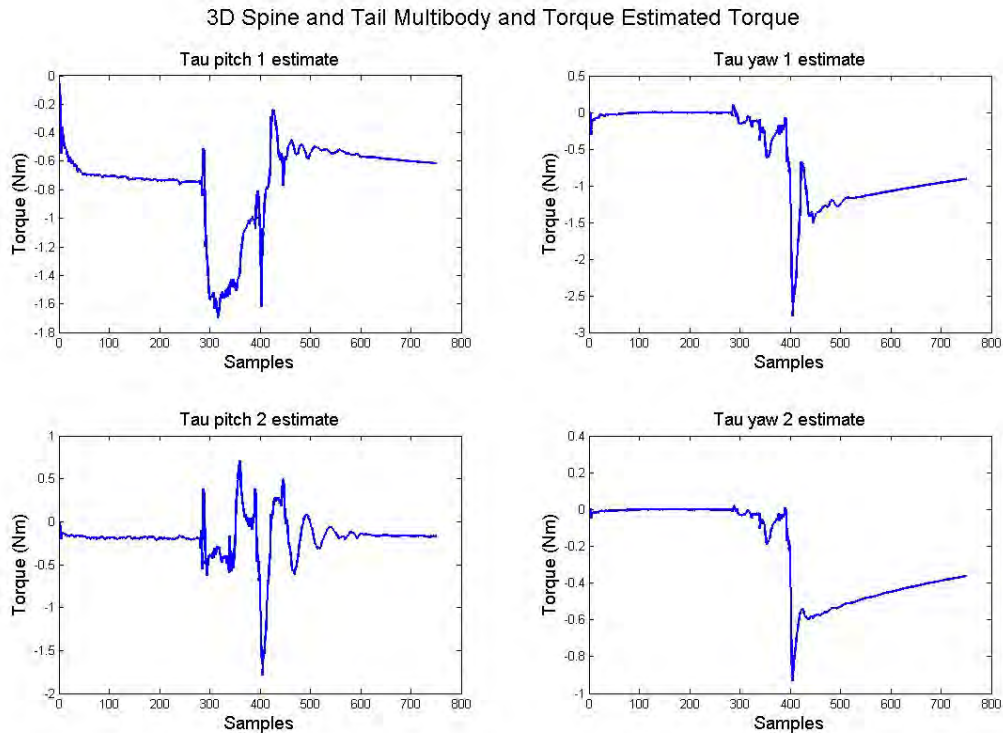
Figure 6.35: 3D spine tail *Multibody and torque* estimated tail torques.
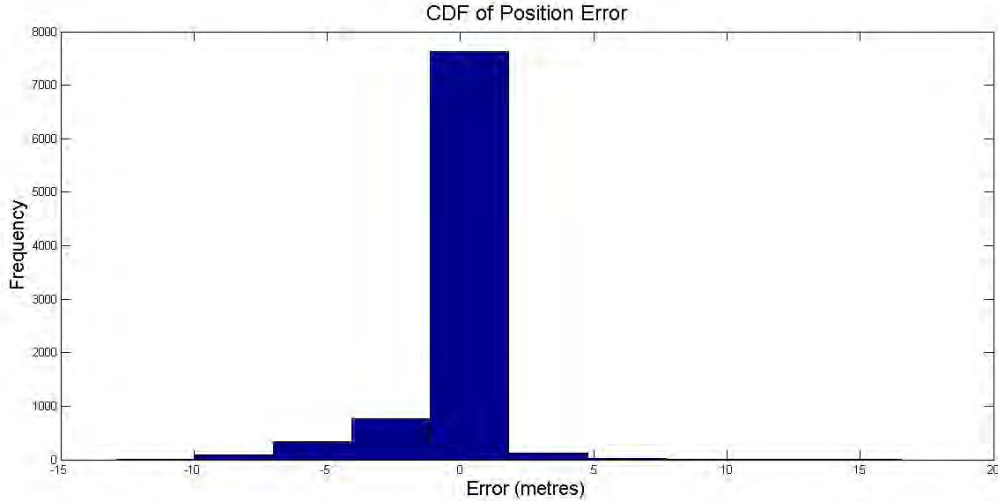
Figure 6.36: A cumulative distribution function of the angular error.

6.5 Summary of Cheetah Model Results

The above results are summarized in Table 6.1. These results are based on the data set that was used to tune the filter.

Table 6.1: Summary of cheetah model results on tuned data set

Algorithm	RMS error (degrees)	RMS error (metres)
2D tail *Multibody*	14.64	0.13
2D tail *Multibody and torque*	14.16	0.13
2D tail *TRIAD KF*	20.36	0.16
3D tail *Multibody*	15.98	0.12
3D tail *Multibody and torque*	13.20	0.12
3D tail *TRIAD KF*	49.20	0.23
2D tail and spine *Multibody*	14.84	0.15
2D tail and spine *Multibody and torque*	14.94	0.16
3D tail and spine *Multibody*	NA	0.12
3D tail and spine *Multibody and torque*	NA	0.11

A number of other tests were performed. The covariances that were determined, by using a data set to tune the filter, were then applied to other data sets to determine how well the filter performed on unseen data. Five additional tail flicks were performed and four additional spine and tail flicks were performed. The results are shown in Appendix A.11. The average of the results are shown in Table 6.2 and represented graphically in Figure 6.37 and 6.38.

98

Table 6.2: Summary of cheetah model results (Average)

Algorithm	RMS error (degrees)	RMS error (metres)
2D tail *Multibody*	14.38	0.12
2D tail *Multibody and torque*	7.99	0.10
2D tail *TRIAD KF*	19.07	0.13
3D tail *Multibody*	15.85	0.12
3D tail *Multibody and torque*	7.92	0.11
3D tail *TRIAD KF*	42.20	0.21
2D tail and spine *Multibody*	7.68	0.07
2D tail and spine *Multibody and torque*	8.13	0.07
3D tail and spine *Multibody*	NA	0.12
3D tail and spine *Multibody and torque*	NA	0.13
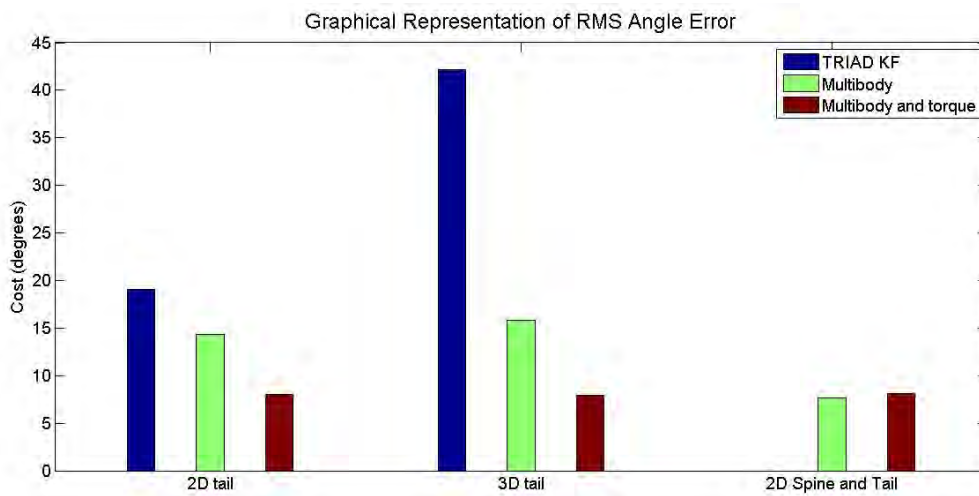
Figure 6.37: A graphical representation of the angular RMS error for the above algorithms.
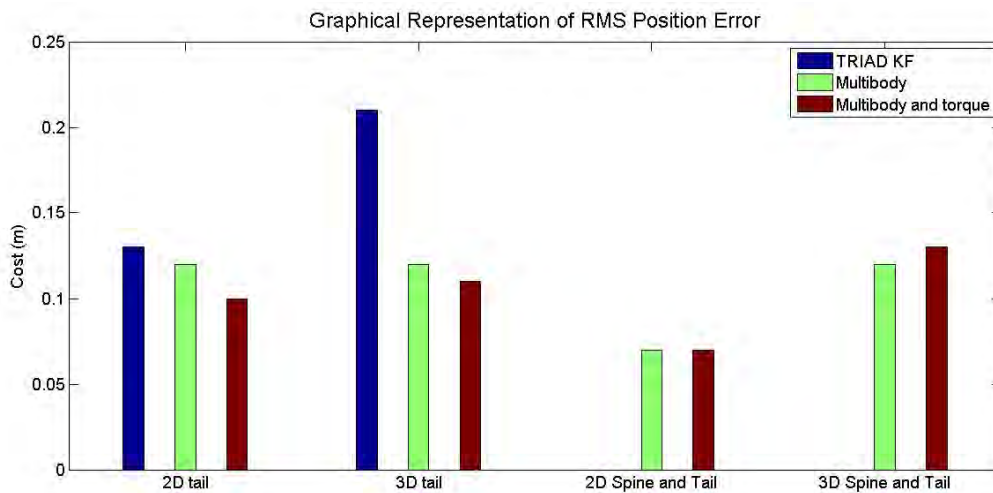
Figure 6.38: A graphical representation of the position RMS error for the above algorithms.

As can be seen by the results the *TRIAD KF* algorithm did not perform as well as the other two tail algorithms. This is due to the linear accelerations affecting the TRIAD algorithm. Due to the poor performance of the *TRIAD KF* algorithm, it was not extended to the spine model.

The filters were tuned, see Section 6.2, for position accuracy and not angular accuracy due to the flexing in the tail and spine. Therefore the angles generated by the filters were used to calculate the positions assuming rigid beams for the spine and tail which were compared to the positions from the camera system. The cost functions were derived in Section 3.4.

As can be seen by these results, two algorithms were developed that can track the motion of the cheetah tail and spine with an RMS error in metres of less than 0.15 m in all cases. These algorithms also estimated other states of the system such as the angular rates and torques generated by the tail.

The tail states can be accurately tracked by the algorithms, but when the spine was introduced in Section 6.4, the accuracy of the system fell due to the fact that the spine was a lot more flexible than the tail and therefore the rigid beam model could not accurately track the motion. As discussed in Section 2.3.1, it is recommended that more rigid beams should be used to track the state variables of the cheetah's spine.

The algorithms were also capable of estimating the torque of the tail. The estimated torque was compared to the motor torque, which was calculated using the current drawn by the motor multiplied by the motor constant. It was shown that the estimated torque has the same general shape as the motor torque. The estimated torque does not track the motor torque exactly due to the fact that the motor efficiency along with the gearbox and mechanical end stops were not modelled. The applied tail torque in the algorithms were modelled as a random walk.

Chapter 7

Discussion and Recommendations

The main aim of the research was to develop a system to estimate the state variables of a cheetah's spine and tail using low cost MEMs sensors. This required the development of a wireless MOCAP system along with dynamic models of the tail and spine. Three algorithms were generated to track the motion of the tail and due to their performances only two were extended to track the motion of the spine. The network was placed on a mechanical rig that simulated the motion of the cheetah and the algorithms were validated using a camera system. Deploying the sensors in the wild was out of the scope of this research.

As seen by the results in Section 6.5, two algorithms were generated that can successfully track the motion of the cheetah's spine and tail. The algorithms were able to track a spine and tail flick, in 3D, with an RMS error of less than 0.15 metres. Insight into the tail torque was also gained, and compared to the logged motor torque, from the *Multibody and torque* algorithm.

The Xsens sensor network that was discussed in Section 2.5 is the closest competitor to the developed sensor network. Xsens boasts an RMS error of 3 degrees per sensor yet they do not say whether it is a static or dynamic error. The algorithms that were developed were for an entire system of rigid bodies and achieved an RMS error of less than 16 degrees in all cases for dynamic tests using a much cheaper sensor network system (the developed network costs under R 5 000 while the Xsens network without the software costs R 163 896). The filters were tuned to optimise their position estimates and not their angles; the reason for this was due to the flexibility of the cheetah spine and tail.

From necropsy data, COM and inertia calculations were performed and the results were found to coincide with published values in [37]. These results go against the commonly assumed idea that the cheetah tail is heavy with a large moment of inertia value [4] [5] [6] [7] [8]. From aerodynamic experiments the coefficient of the cheetah's tail was determined and was included in the state estimation algorithms.

The algorithms became increasingly complex (for the 3D spine and tail case) and were

therefore implemented using numerical techniques instead of generating the equations of motion analytically. To determine if the numeric implementation was functioning correctly, a 2D tail filter was implemented analytically and numerically and the results were compared. The analytical cost function (RMS error of 0.1251 m) for the position error is better than the numerical (RMS error of 0.1354 m) version. For a slight loss in accuracy, the algorithms can be easily implemented. As can be seen the numerical implementation of the filters is an accurate and viable solution to handle the increasingly complex dynamic equations.

There are three main limitations to the research performed. The first limitation is the camera system that was used to validate the results. The camera system operated at 60 Hz compared to the 100 Hz of the developed sensor network. Therefore the data from the camera system was interpolated to 100 Hz. There is also noise present in the camera system data that was not quantified or taken into account. The camera system data was assumed accurate and was used as the absolute position of the sensors. It is recommended to use a more accurate camera system such as the Vicon system to validate future results or to perform a thorough calibration of the camera system and to quantify the accuracy of the system.

The second limitation was due to the mechanical rig that was built to test and simulate motion for the developed sensor network. The mechanical rig used a large DC motor that saturated the magnetometers that were part of the sensor network. The magnetometer data was simulated from camera data. This however is not a crucial problem as there will be no magnetic interference when the sensor network is deployed on wild animals.

The third limitation is with regard to the torque estimate. From the results, the algorithms containing a torque estimate generally performed better. The limitation is that the torque covariance is tuned for a certain flick profile. As soon as the tail flicks faster or slower, the algorithm will no longer be able to track the motion reliably as it will result in a different torque profile.

These limitations do have an impact on the presented results. Although good results were achieved, due to the camera system having noise in the measurements, the algorithms cannot be tuned correctly as the camera data is assumed to be the absolute position of the sensor with no noise. Therefore the developed algorithms can only be as good as the camera system. The algorithms that have a torque estimate can only track a single flick profile. The data can either be analysed beforehand to determine the torque profile and then select a torque covariance to match the flick or the pure kinematics models can be used instead as it can track multiple flick profiles.

It is recommended to keep the sensor as small and as light as possible. The current version of the sensor weighs 10 grams and is 32 × 19 mm. If the sensors are too big and heavy they will alter the dynamics of the cheetahs tail along with irritating the cheetah. This could change how the cheetah behaves. The sensors must also be firmly attached to the cheetah in order to stop them from either slipping off or moving/rotating during a rapid manoeuvre. One problem that will need to be solved is that the sensors will not be mounted perfectly straight and will need to be rotated so that the sensor axes line

up with the body axes. A method to attach the sensors to the cheetah must still be determined.

In order to place the sensors onto a cheetah, the animal will need to be captured and sedated. This is a complex and expensive task that is not enjoyed by the animal, therefore the sensors must be extremely robust and have sufficient battery life to last for a few months at a time. It is recommended that the sensors be switched off during the day and night and left in sleep mode during sunrise and sunset, when the cheetah is most active. During these times the sensors can be woken up by an interrupt that is generated by an accelerometer spike, indicating that the cheetah has started running. This concept was used by [20] on their cheetah collar.

A limitation of the current developed sensor network is that the collar needs to be within 100 metres of the base station (when WiFi is implemented). Future versions of the collar will contain a GSM module in order to upload the data to the internet.

Chapter 8

Conclusions and Future Work

A state estimation and prediction algorithm was developed to track the motion of the cheetah spine and tail. This required the development of a MOCAP system and dynamic models. To date, no researchers have developed a system that can be used to analyse and study the dynamic motion of the cheetah. The research of this MSc engineering study has answered the questions that were stated in Section 1.4 as follows:

1. One can estimate the states of the cheetahs tail and spine using a low cost IMU sensor network.

2. It is possible to calculate an estimate of the torques induced onto the cheetah's body, by its tail, using low cost IMU sensors.

As can be seen by the results obtained in Section 6.5 which were discussed in Chapter 7, a state estimation algorithm was generated that can accurately track the motion of the spine and tail of the mechanical rig using data from a custom built, low cost motion capture system. The algorithms were validated using a camera system and achieved an RMS error of less than 16 degrees in all cases (not including the *TRIAD KF* algorithm that was not used due to its poor performance).

These algorithms can be used to analyse the motion of the spine and tail and will provide insight into the motion. It was determined that the tail cannot perform a roll motion due to the layout of the muscles that power the tail. The spine however is powered by a more complex array of muscles and can therefore roll, pitch and yaw. It was also observed that the spine bends near the rear, with the front half remaining relatively rigid. This effect was captured in the model of the cheetah using two rigid beams of different lengths.

The tail was observed to flick rapidly during the transient motions performed by the cheetah. The aerodynamic and inertial effects of the tail were modelled and it was determined that the aerodynamic effect plays a vital role in the generated torque. Due to the fact that the aerodynamics and inertia of the tail were modelled, a torque state was

included in the algorithms to estimate the applied torque (which is equal to the reaction torque on the body). The literature assumed that the tail was a heavy and high inertia tail [4] [5] [6] [7] [8] but it was discovered that the tail was in fact fairly light (about 2.5 % of the body mass) and that it has a relatively low inertia value.

To date, the motion of the cheetah has not been rigorously analysed or studied. There is also little knowledge on the effects of the spine and tail on the manoeuvrability of the animal. This research has generated an initial model that can be used to analyse the motion and to gain further insight into the animal.

The next step is to develop these algorithms further and to perform field tests on wild or captive cheetahs. The motion of the spine and tail can then be fully analysed. As can be seen, it is possible to track the motion of a cheetah tail and spine using a low cost (less than R 5 000) motion capture system costing a fraction of the cost of available systems.

8.1 Future Work

Future work should involve developing the system with the aim of deploying the sensor network onto cheetahs in the wild. This will involve modifying the developed algorithms to take into account the linear motion of the animal. Initial investigation and planning has been done and is described below.

8.1.1 GPS Integration

The algorithms were designed and developed on a stationary mechanical test rig as discussed in the introduction, Section 1.2.1. When the sensor network is placed onto the cheetah, it will not remain stationary. The filter algorithms need to be modified to take this into account. A position estimate filter was designed on simulated GPS (at 10 Hz) and IMU (at 100 Hz) data. The simulation included the cheetah accelerating and braking along with angular rates simulating the bending of the spine. The results are shown below. The equations for the filter are derived in Appendix A.13.

This is just an initial investigation into the position filter that will allow the developed algorithms to move from a stationary rig to a moving animal. The results are summarized in Table 8.1.

Table 8.1: Summary of position filter results

GPS	RMS error in metres
GPS	1.7897
no GPS	87.0966

8.1.2 Investigation Into Fewer Sensors

Due to the fact that placing sensors onto a wild animal is a challenging task, the fewer sensors placed onto the animal the better. It was therefore decided to conduct the initial investigation into what results would be achieved if the tail was modelled as a single beam. The *Multibody* and *Multibody with torque* tail algorithms were modified (for the 3D case only).

The equations of the modified algorithms are shown in Appendix A.12.1. The results can be seen in Figure 8.2 and Figure 8.3. As can be seen the tip position of the tail is tracked fairly accurately and will give a good enough approximation of the tail movement compared to the systems derived in Chapter 6. Further investigation must still be done.

The average RMS error in metres of the one beam *Multibody* algorithm was 0.1242 and for the *Multibody and torque* algorithm the average was 0.1214. When these results are compared to the 3D tail algorithms they achieved an average RMS error of 0.1194 and 0.1077 respectively. As can be seen accurate results using fewer sensors were obtained.

The MOCAP system will be further developed to a state that it is ready to be deployed into the wild. The developed algorithms will be modified, as modelling the applied torque as a random walk is not a valid solution that works for all tail flick profiles. Therefore a pure kinematics model, that estimates the forces and torques using accelerometer data as an input to the prediction stage of the Kalman filter, will be developed and tested.
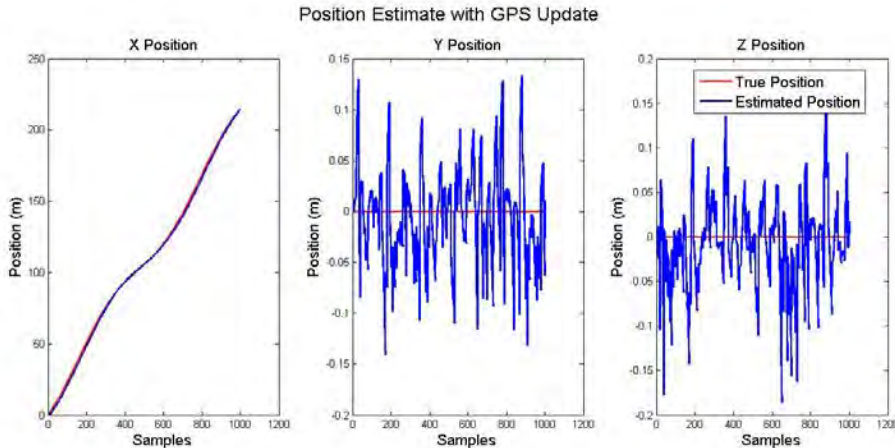
(a) position filter with 10 Hz GPS update.
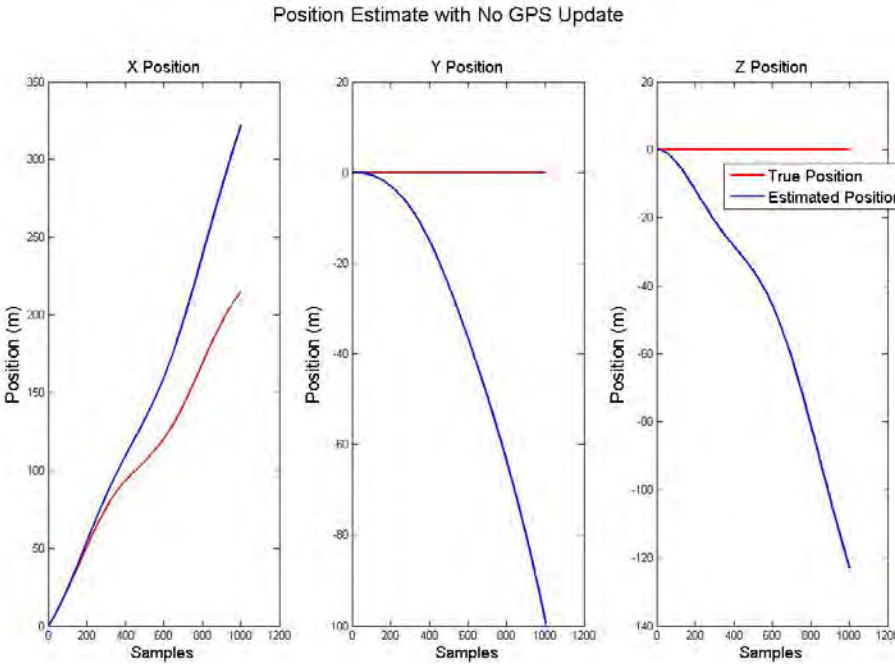
(b) position filter with no GPS update.

Figure 8.1: Position filter results.
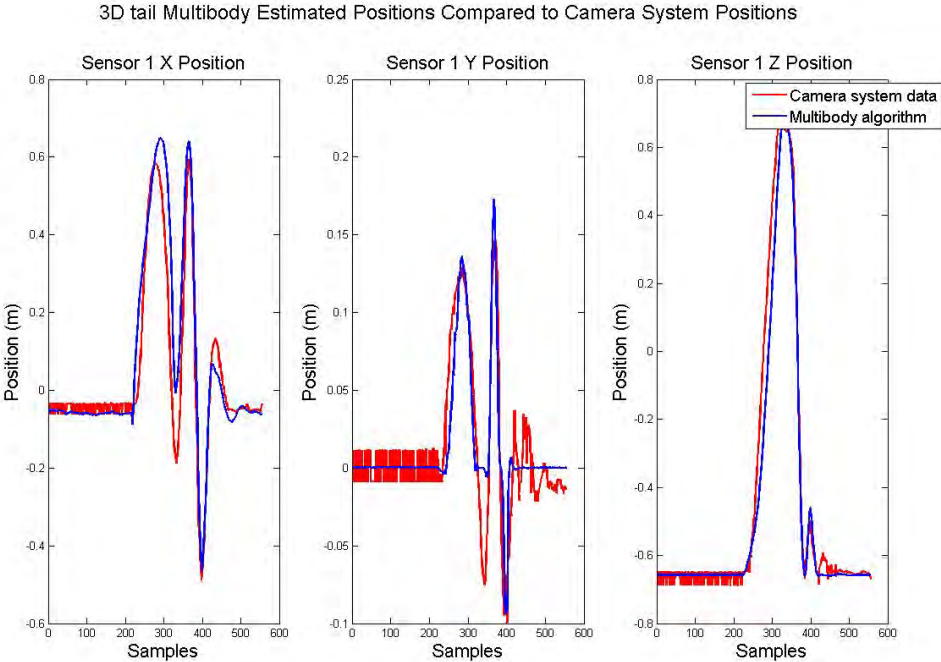
Figure 8.2: Position results for the one beam *Multibody* algorithm.
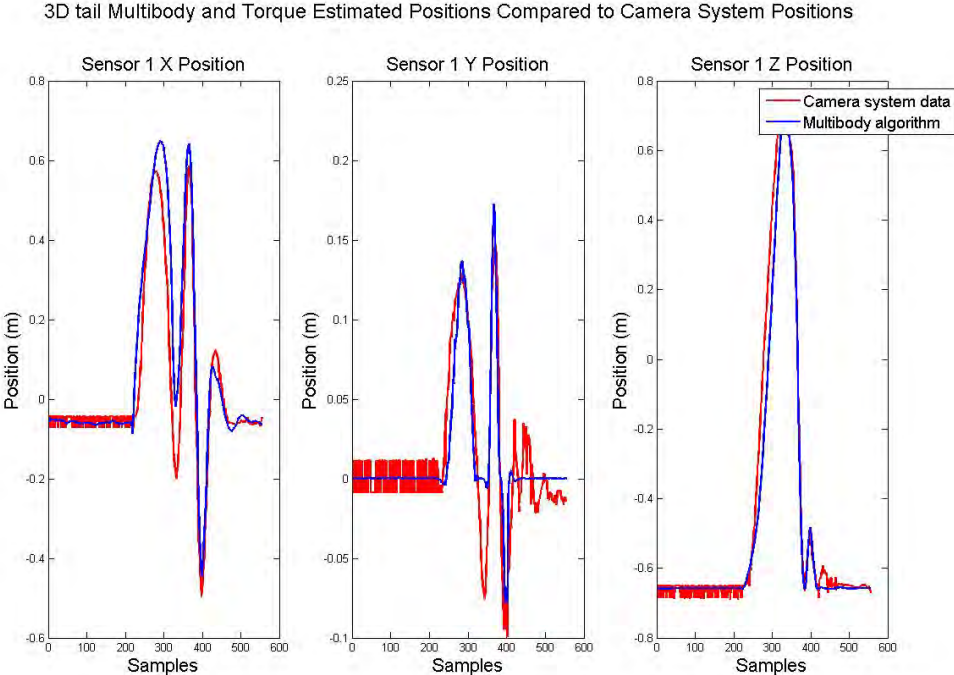
Figure 8.3: Position results for the one beam *Multibody and Torque* algorithm.

Bibliography

[1] A. Patel and M. Braae. "An Actuated Tail Increases Rapid Acceleration Manoeuvres in Quadruped Robots." *Innovations and Advances in Computing, Informatics, Systems Sciences, Networking and Engineering*, pp. 69-76, 2015.

[2] R. McNeill Alexander, *Principles of Animal Locomotion,* Princeton University Press, 2006.

[3] R. Briggs, J. Lee, M. Haberland and S. Kim, "Tails in Biomimetic Design: Analysis, Simulation, and Experiment," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2012.

[4] Julia Layton. (2014) What Makes a Cheetah Run So Fast? [Online].`http://animals.howstuffworks.com/mammals/cheetah-speed1.htm`

[5] Animal Planet. (2014) Animal Planet: Cheetah Spine and Tail. [Online]. `http://www.animalplanet.com/tv-shows/animal-planet-presents/videos/survival-guideserengeti-cheetah-spine-tail.html`

[6] Sabi Sabi Game Reserve. (2015) Sabi Sabi Wild Facts: The Cheetah. [Online]. `http://www.sabisabi.com/wildfacts/cheetah`

[7] S.E. Thompson, "Built for Speed: The Extraordinary, Enigmatic Cheetah," Lerner Pub Group, 1998.

[8] M. Hilderbrand, "Further Studies on Locomotion of the Cheetah," *Journal of Mammalogy*, vol. 42, no. 1, pp. 84-91, 1961.

[9] N. C. C. Sharp, "Timed running speed of a cheetah," *Journal of Zoology*, vol. 241, no. 3, pp. 493-494, 1997.

[10] Boston Dynamics. (2015) CHEETAH- Fastest Legged Robot. [Online]. `http://www.bostondynamics.com/robot_cheetah.html`

[11] S. Seok, "Highly parallelized control programming methodologies using multicore CPU and FPGA for highly dynamic multi-DoF mobile robots, applied to the MIT Cheetah." Massachusetts Institute of Technology, 2014.

[12] Austin Thomas Photography. (2015) Cheetah image wins International Gold Medal. [Online]. `http://www.austin-thomas.co.uk/blog/?p=425`

[13] Youtube. (2015) Cheetah turning fast. `https://www.youtube.com/watch?v=emwccTx5vLg`

[14] Youtube. (2015) Cheetah vs Greyhound. `https://www.youtube.com/watch?v=jc8Hno4M0Qs`

[15] Youtube. (2015) Cheetah vs Gazelle. `https://www.youtube.com/watch?v=KTScfs-o2vQ`

[16] J. Chu, "New algorithm enables MIT cheetah robot to run and jump, untethered, across grass," PHYS ORG, 2014.

[17] K. Knight, "How Cheetahs Outpace Greyhounds." *The Journal of Experimental Biology*, vol. 215, 2012.

[18] P. Hudson, S. Corr and A. Wilson, "High speed galloping in the cheetah (Acinonyx jubatus) and the racing greyhound (Canis familiaris): spatio-temporal and kinetic characteristics," *The Journal of experimental biology*, vol. 215, pp. 2425-2434, 2012.

[19] J. W. Wilson, M. G. L. Mills, R. P. Wilson, G. Peters, M. E. J. Mills, J. R. Speakman, S. M. Durant, N. C. Bennett, N. J. Marks and M. Scantlebury, "Cheetahs, *Acinonyx jubatus*, balance turn capacity with pace when chasing prey," *Biology Letters*, 2013.

[20] A. Wilson, J.C. Lowe, K. Roskilly, P. Hudson, K.A. Golabek and J.W. McNutt, "Locomotion dynamics of hunting in wild cheetahs," *Nature*, vol. 498, pp. 185-189, 2013, Nature Publishing Group.

[21] VICON. (2015) VICON. [online] `http://www.vicon.com/`

[22] Y. Reddi and E. Boje, "System Identification for Low-cost Small-scale Helicopters," *World Congress of International Federation of Automated Control (IFAC)*, vol. 19, 2014.

[23] LentinkLab (2015) Biological Questions [online] `http://lentinklab.stanford.edu/welcome/biological_questions`

[24] A. T. Baisch, P. S. Sreetharan and R. J. Wood, "Biologically-Inspired Locomotion of a 2g Hexapod Robot", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.

[25] U. G. K. Wegst, H. Bai, E. Saiz, A. P. Tomsia and R. O. Ritchie, "Bioinspired Structural Materials", *nature materials*, 2014.

[26] Biomimicry (2015) What is biomimicry? [online] `http://biomimicry.org/what-is-biomimicry/`

[27] BioInspired (2015) So what? Now What? [online] `http://bioinspired.sinet.ca/`

[28] J. Bertram and A. Gutmann, "Motions of the running horse and cheetah revisited: fundamental mechanics of the transverse and rotary gallop," *Journal of The Royal Society Interface*, vol. 6, pp. 549-559, 2009.

[29] DailyTech. (2015) MIT Robo-Cheetah v2 vs Spot. `http://images.dailytech.com/nimage/MIT_Robo_Cheetah_v2_vs_Spot_FP_Wide.jpg`

[30] A. Patel and M. Braae. "Rapid turning at high-speed: Inspirations from the cheetah's tail," *Intelligent Robots and Systems (IROS)*, pp. 5506-5511, 2013, IEEE.

[31] E. Chang-Siu, T. Libby, M. Tomizuka and R. Full, "A lizard-inspired active tail enables rapid maneuvers and dynamic stabilization in a terrestrial robot," *Intelligent Robots and Systems (IROS)*, pp. 1887-1894, 2011, IEEE.

[32] N.J. Kohut, D.W. Haldane, D. Zarrouk and R.S. Fearing, "Effect of inertial tail on yaw rate of 45 gram legged robot," *Int. Conf. Climbing Walk. Robot. Support Technol. Mob. Mach*, pp. 157-164, 2012.

[33] C. Fisher and A. Patel, "FlipBot: A Lizard Inspired Stunt Robot," *IFAC World Congress of International Federation of Automated Control (IFAC)*, vol. 19, pp. 4837-4842, 2014.

[34] S.B Williams, T. Huiling, J.R. Usherwood, and A. Wilson, "Pitch then power: Limitations to acceleration in quadrupeds," *Biology Letters*, vol. 5, pp. 610-613, 2009.

[35] N. C. C. Sharp, "Timed running speed of the cheetah (Acinonyx jubatus)," *J. Zool*, vol. 241, pp. 493494, 1997.

[36] National Geographic. (2015) Cheetah Sprinting After Prey. [Online]. `http://animals.nationalgeographic.com/wallpaper/animals/photos/cheetahs/cheetah-run/`

[37] P.E. Hudson, "The structural and functional specialisation of locomotion in the cheetah (Acinonyx jubatus)," Ph.D. dissertation, Royal Veterinary College, University of London, London, 2011.

[38] A. Patel. "Understanding the Motions of the Cheetah Tail Using Robotics," Ph.D. dissertation, Dept. Elect. Eng, UCT, Cape Town, 2014.

[39] WorkSmart (2015) Stay clear of cheetah programmers [online] `http://www.worksmartdb.com/run-away-from-that-cheetah/`

[40] A. Patel, E. Boje, C. Fisher, L. Louis, and E. Lane. "Quasi Steady State Aerodynamics of the Cheetah Tail," unpublished.

[41] Thitima Jintanawan, "Advanced Dynamics," Chulalongkorn University.

[42] A. Gracia, *Kinematics of Robots*, Idaho State University.

[43] S.K. Saha, "Recursive Dynamics Algorithms for Serial, Parallel, and Closed-chain Multibody Systems" *Indo-US Workshop on Protein Kinematics & Protein Conformations*, IISc, Bangalore, 2007.

[44] B. Guenter and S.H. Lee, "Symbolic Lagrangian Multibody Dynamics," Microsoft Research, 1994.

[45] C. Lanczos, *Variational Principles and Lagrangian Mechanics*, Courier Corporation, 1970, ch. 6 and 7, Vol. 4.

[46] M. Zefran and F. Bullo. *Lagrangian dynamics*, Robotics and Automation Handbook, 2005.

[47] R.M. Murray, Z. Li, and S.S. Sastry, *A Mathematical Introduction to Robotic Manipulation,* CRC Press, 1994.

[48] C. Whelan, *The Principles of Dynamics*, 2000, Pembroke College, Cambridge.

[49] M. Mallick, A. Kumar, N. Tamboli, A.B. Kulkarni, P. Sati, V. Devi, S. Kumar, N. Gogoi, S. Bordoloi, B. Sharma and others, "Study on Drag Coefficient for the Flow Past a Cylinder," *International Journal of Civil Engineering Research*, vol. 5, pp. 301-304, 2014.

[50] J. L. Heseltine, "Flow Around a Circular Cylinder with a Free End," M.S. thesis, Dept. Mech. Eng, University of Saskatchewan, Canada, 2003.

[51] BikeTechReview. (2015) Skinsuits and Boundary Layers [online] `http://www.biketechreview.com/aerodynamics/misc/466-skinsuits-and-boundary-layers`

[52] D. Yeo and N. Jones, "Computational Study on 3-D Aerodynamic Characteristics of Flow around a Yawed, Inclined, Circular Cylinder," Newmark Structural Engineering Laboratory, University of Illinois at Urbana-Champaign, 2011.

[53] H. Ardakani and T. Bridges, "Review of the 3-2-1 Euler Angles: a yaw-pitch-roll sequence," Department of Mathematics, University of Surrey, Guildford UK, 2010.

[54] Chrobotics. (2015) Understanding Euler Angles [online] `http://www.chrobotics.com/library/understanding-euler-angles`

[55] W. Premerlani and P. Bizard, "Direction cosine matrix IMU: Theory," DIY DRONE: USA, 2009.

[56] K. Großekatthöfer and Z. Yoon, "Introduction into quaternions for spacecraft attitude representation," TU Berlin, 2012.

[57] TobyNorris. (2015) Quaternion Background. [online] `http://www.tobynorris.com/work/prog/csharp/quatview/help/orientations_and_quaternions.htm`

[58] "Kinematics of Moving Frames" in *Design of electromechanical robotic systems*, MIT OpenCourseWare, 2009, ch 9.

[59] E. Wade and M. J. Mataric, "Design and Testing of Lightweight Inexpensive Motion-Capture Devices with Application to Clinical Gait Analysis," *3rd International Conference on Pervasive Computing Technologies for Healthcare*, 2009.

[60] R. Rainone, B. Gardner and J. Frost, "Gait Efficiency Analysis Using Three Axis Force Plate," B.S thesis, Worcester Polytechnic Institute, 2008.

[61] N. Miller, O. Jenkins, M. Kallmann and M. Mataric, "Motion capture from inertial sensing for untethered humanoid teleoperation," *4th IEEE/RAS International Conference on Humanoid Robots*, 2004.

[62] S. Das, L. Trutoiu, A. Murai, D. Alcindor, M. Oh, F. D. L. Torre and J. Hodgins, "Quantitative Measurement of Motion Symptoms in Parkinson's Disease: A study with Full-body Motion Capture Data," *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2011.

[63] D. Brown, R. Kays, M. Wikelski, R. Wilson and P. Klimley, "Observing the unwatchable through acceleration logging of animal behavior," 2014, Bibliothek der Universität Konstanz.

[64] S. Hooker, M. Biuw, B. McConnell, P. Miller and C. Sparling, "Bio-logging science: logging and relaying physical and biological data using animal-attached tags," *Deep Sea Research Part II: Topical Studies in Oceanography*, vol. 54, pp. 177-182, 2007, Pergamon.

[65] P. Nogueira, *Motion capture fundamentals*, 2011.

[66] R. Kays, B. Kranstauber, P. Jansen, C. Carbone, M. Rowcliffe, T. Fountain and S. Tilak, "Camera traps as sensor networks for monitoring animal communities," *IEEE 34th Conference on Local Computer Networks*, pp. 811-818, 2009.

[67] VECTORNAV. (2015) VECTORNAV: VN-100. [online] `http://www.vectornav.com/products/vn100-rugged`

[68] XSENSE. (2015) XSENSE: products [online] `https://www.xsens.com/?gclid=CJXt5NvIkccCFZTItAodcFsF8Q`

[69] SHADOW. (2015) SHADOW. [online] `http://www.motionshadow.com/`

[70] TECHNAID. (2015) TECHNAID: Inertial Measurement Unit. [online] `http://www.technaid.com/en/products/inertial-measurement-unit`

[71] M. Brodie, A. Walmsley and W. Page, "Fusion motion capture: a prototype system using inertial measurement units and GPS for the biomechanical analysis of ski racing," *Sports Technology*, pp 17-28, 2008, Wiley Online Library.

[72] APDM. (2015) APDM: wearable-sensors [online] `http://www.apdm.com/wearable-sensors/`

[73] E. Farella, L. Benini, B. Ricco and A. Acquaviva, "MOCA: A low-power, low-cost motion capture system based on integrated accelerometers," *Advances in Multimedia*, 2007, Hindawi Publishing Corp.

[74] R. Zviedris, A. Elsts, G. Strazdins, A. Mednis and L. Selavo, "LynxNet: wild animal monitoring using sensor networks," *Real-World Wireless Sensor Networks*, 2010, Springer.

[75] MEMs Journal (2015) Emerging Technology and application trends [online] `http://www.memsjournal.com/`

[76] D. Li, R. Landry and P. Lavoie, "Low-cost MEMS sensor-based attitude determination system by integration of magnetometers and GPS: A real-data test and performance evaluation," *Position, Location and Navigation Symposium*, 2008 IEEE/ION, pp. 1190-1198, 2008.

[77] L. Kang, L. Ye, K. Song and Y. Zhou, "Attitude heading reference system using MEMS inertial sensors with dual-axis rotation," *Sensors*, vol. 14, pp. 18075-18095, 2014, Multidisciplinary Digital Publishing Institute.

[78] S. Madgwick, "An efficient orientation filter for inertial and inertial/magnetic sensor arrays," Report x-io and University of Bristol (UK), 2010.

[79] P. H. Dana, "Global Positioning System (GPS) Time Dissemination for Real-Time Applications," *Real-Time systems*, vol. 12, pp 9-40, 1997.

[80] H Durrant-Whyte, "Introduction to Estimation and the Kalman Filter," Australian Centre for Field Robotics, The University of Sydney, 2006.

[81] J. Andrade-Cetto, "The Kalman Filter," Institut de Rob'otica i Inform'atica Industrial, UPC-CSIC, 2002.

[82] S. Hanykin, *Kalman Filtering and Neural Networks*, 2001.

[83] R. Faragher, "Understanding the Basis of the Kalman Filter Via a Simple and Intuitive Derivation," IEEE Trans. Signal Process., 2012

[84] J. L. Crassidis and F. L. Markley, "Unscented Filtering for Spacecraft Attitude Estimation," 2003.

[85] Y. Cheng and J. L. Crassidis, "Particle Filtering for Sequential Spacecraft Attitude Estimation," *AIAA Guidance, Navigation, and Control Conference and Exhibit.* Reston, VA: AIAA, 2004.

[86] D. Choukroun, L. Cooper and N. Berman, "A Stochastic H∞ Filter for Spacecraft Attitude Estimation."

[87] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," Department of Computer Science, University of North Carolina, Chapel Hill.

[88] IR-US beacons (2015) Infrared-Ultrasonic Beacons [online] `http://www.convict.lu/htm/rob/ir_us.htm`

[89] T. Meis and U. Marcowitz, *Numerical solution of partial differential equations*, vol. 32, 2012, Springer Science & Business Media.

[90] W. Ames, *Numerical methods for partial differential equations*, 2014, Academic press.

[91] L. Markley, "Attitude determination using two vector measurements," NASA CONFERENCE PUBLICATION, pages 39-52, 1999, NASA.

[92] S. Tanygin and M. Shuster, "The many triad algorithms," presented as paper AAS-07-104 at the AAS/AIAA 17th Space Flight Mechanics Meeting, Sedona, Arizona, pp 81-99, 2007.

[93] W. Adiprawita, A.S. Ahmad and J. Semibiring, "Unmanned Aerial Vehicle Instrumentation for Rapid Aerial Photo System," arXiv preprint arXiv:0804.3894, 2008.

[94] SparkFun (2015) nRF24L01+ Single Chip 2.4GHz Transceiver [online] `https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Pluss_Preliminary_Product_Specification_v1_0.pdf`

[95] SparkFun, (2015) XBee/XBee-PRO RF Modules [online] `https://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-Datasheet.pdf`

[96] K. L. Calvert, "802.11 Wireless LAN Protocol"

[97] P. Smith, "Comparisons between Low Power Wireless Technologies," *Marketing Manager HBU, CSR plc*

[98] Z. Hurak, "From double pendulum with massless rods to simple robotic arms: mathematical modeling using Lagrange approach," 2012.

[99] Wonderpolis. (2015) Can a human outrun a cheetah. [online] `http://wonderopolis.org/wonder/can-a-human-outrun-a-cheetah/`

[100] 500 px (2015) Cheetah Chasing Gazelle [online] `https://500px.com/photo/741583/cheetah-chasing-gazelle-by-austin-thomas`

[101] Youtube (2015) Predator Cheetah vs Antelope (National Geographic) [online] `https://www.youtube.com/watch?v=zNqTmiaYD4w`

[102] MuMetal. (2015) MuMetal. [online] `http://www.mumetal.com/about_mumetal.html`

[103] Digi-Key (2015) STMicroelectronics INEMO-M1 [online] `http://www.digikey.com/product-detail/en/INEMO-M1/497-14191-ND/4542053`

[104] st (2015) INEMO-M1 [online] `http://www.st.com/web/en/resource/technical/document/datasheet/DM00056715.pdf`

[105] Miniinthebox. (2015) Electronics DIY NRF24L01. [online] `http://www.miniinthebox.com/electronics-diy-nrf24l01-2-4ghz-wireless-communication-module_p340882.html`

[106] Nordic Semi (2015) nRF24L01 Product Specification `https://www.nordicsemi.com/eng/content/download/2730/34105/file/nRF24L01_Product_Specification_v2_0.pdf`

[107] FreeRTOS. (2014) FreeRTOS. [Online]. http://www.freertos.org/

[108] J.W.S. Bradshaw, *The Behaviour of the Domestic Cat*, 1st ed.: CABI, 1992.

[109] Chris Hall, "Attitude Determination," 2013, ch.4.

[110] A random walk process (2015) [online] `http://cmapskm.ihmc.us/rid=1052458884462_996058812_7176/randomwalk.pdf`

[111] Freescale, "AN4399 Application Note, High Precision Calibration of a Three-Axis Accelerometer," Rev.1, 2013.

[112] MathWorks. (2015) MathWorks: Magnetometer calibration [online] `www.mathworks.com/matlabcentral/fileexchange/23398-magnetometer-calibration`

[113] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," Matrix, volume 58, 2006.

Appendix A

A.1 Calibration of MEMs

Before the MEMs sensors can be used, they need to be calibrated. The calibration removes static biases that are present and also ensures that the sensor axis lines up with the PCB/body axis. The calibration also removes co-axis coupling that occurs when the sensor axes are not perpendicular to each other.

A.1.1 Gyroscope Calibration

A static calibration algorithm was applied to the gyroscope raw data. In order to calibrate the sensor, data was logged at 100 Hz for ten minutes with the sensor stationary. The angular rate in all three axes as well as the temperature from the temperature sensor was logged.

The sensor was placed in the sun while the data was logged. This ensured that the temperature changed while the data was logged. A graph of the angular rate (which ideally should be zero) versus the temperature was plotted and is shown in Figure A.1.
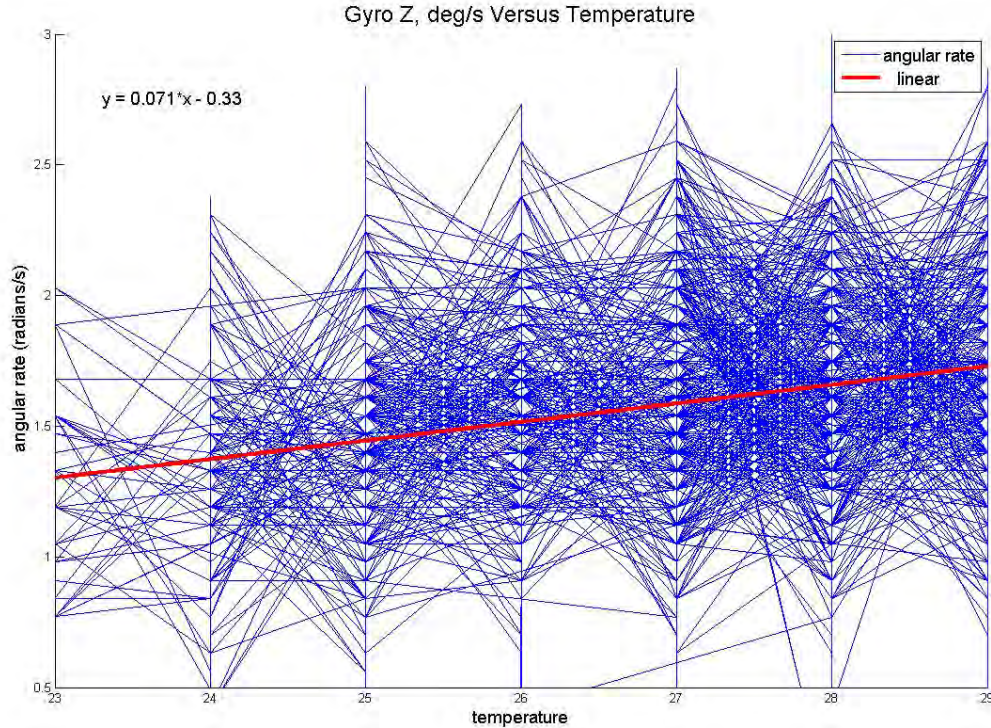
Figure A.1: Graph showing the angular rate versus the temperature. Note how the angular rate bias changes with temperature.

A straight line was fitted to the data. This straight line is used to find the bias at a specific temperature, as the bias of the sensor changes with the change in temperature.

The line has the form $y = mx + c$ where c is the zero degree Celsius bias and m is the gradient of the temperature dependence. The current temperature is x and y is the temperature dependent bias of the gyroscope. This value needs to be subtracted from the gyroscope reading to get the actual angular rate.

Further dynamic calibration of the gyroscopes can be performed using a turn table that rotates at a known angular rate. This rate can be compared to the angular rate determined by the gyroscope and any deviances can be calibrated out.

A.1.2   Accelerometer Calibration

There are many different calibration algorithms that can be used to calibrate the accelerometer. A static calibration algorithm was used. This means that the calibration algorithm will try to approximate the gravity vector as closely as possible in the relevant axis.

Some common algorithms include the 6 parameter, 12 parameter and 15 parameter calibration algorithms [111]. The 12 parameter algorithm was used and takes into account

117

the cross axis interactions and any rotation due to mounting the sensor on the PCB incorrectly. It is the most general linear calibration of accelerometers and uses a total of 12 parameters to calibrate the sensor.

The sensor is placed in six different positions (one at a time) and data is logged at 100 Hz for at least a minute in each position. The six positions are the positive gravity vector in each of the three axes and the negative gravity vector in each of the three axes. The format of the calibrated data is expressed as:

$$\begin{bmatrix} Y_x \\ Y_y \\ Y_z \end{bmatrix} = \begin{bmatrix} w_{xx} & w_{xy} & w_{xz} \\ w_{yx} & w_{yy} & w_{yz} \\ w_{zx} & w_{zy} & w_{zz} \end{bmatrix} \begin{bmatrix} G_x \\ G_y \\ G_z \end{bmatrix} + \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} \tag{A.1}$$

where $\mathbf{Y}$ is the calibrated data, $\mathbf{G}$ is the raw data, w are 9 of the parameters that form the gain and rotation matrix and $\mathbf{V}$ are the biases (the remaining 3 calibration parameters). This equation can also be written in the following form:

$$\begin{bmatrix} Y_x & Y_y & Y_z \end{bmatrix} = \begin{bmatrix} G_x & G_y & G_z & 1 \end{bmatrix} \begin{bmatrix} w_{xx} & w_{xy} & w_{xz} \\ w_{yx} & w_{yy} & w_{yz} \\ w_{zx} & w_{zy} & w_{zz} \\ V_x & V_y & V_z \end{bmatrix} \tag{A.2}$$

This is shortened to

$$\mathbf{Y} = \mathbf{GX} \tag{A.3}$$

where $\mathbf{Y}$ is the calibrated data, $\mathbf{G}$ is the raw data with a 1 appended to it and $\mathbf{X}$ are the 12 calibration parameters. To calculate the twelve calibration parameters the following formula is used:

$$\mathbf{X} = \left[\mathbf{G}^T \mathbf{G} \right]^{-1} \mathbf{G}^T \mathbf{Y} \tag{A.4}$$

Where $\mathbf{G}$ is a 3 by n matrix of all the logged raw data from the 6 different positions and $\mathbf{Y}$ is a 3 by n matrix which corresponds to what the accelerometer should be measuring (There are six possible values it can measure: $[0,0,\pm1],[0,\pm1,0]$ and $[\pm1,0,0]$). The sum of the number of measurements, $n = n_1 + n_2 + n_3 + n_4 + n_5 + n_6$, corresponds to the total number of samples logged in the six different positions.

A.1.3 Magnetometer Calibration

The magnetometer is subjected to hard and soft-iron disturbances due to ferrous magnetic material. A calibrated magnetometer that is rotated through all the angles will give readings that form a sphere centred at the origin with a radius equal to the length of the magnitude of the local magnetic vector.

The disturbances and errors due to cross axis interference and mounting misalignment cause the measurements to generate an ellipsoid, as seen in Figure A.2, that is not centred about the origin. The calibration takes this ellipsoid and maps it to a sphere about the origin.
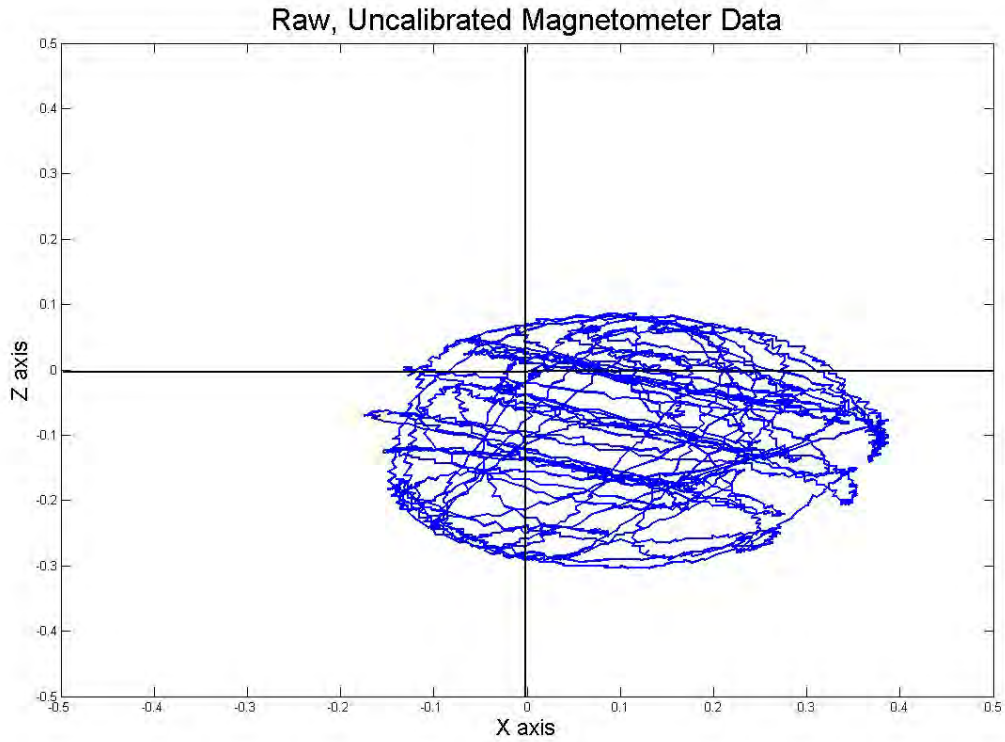
(a) uncalibrated magnetometer.
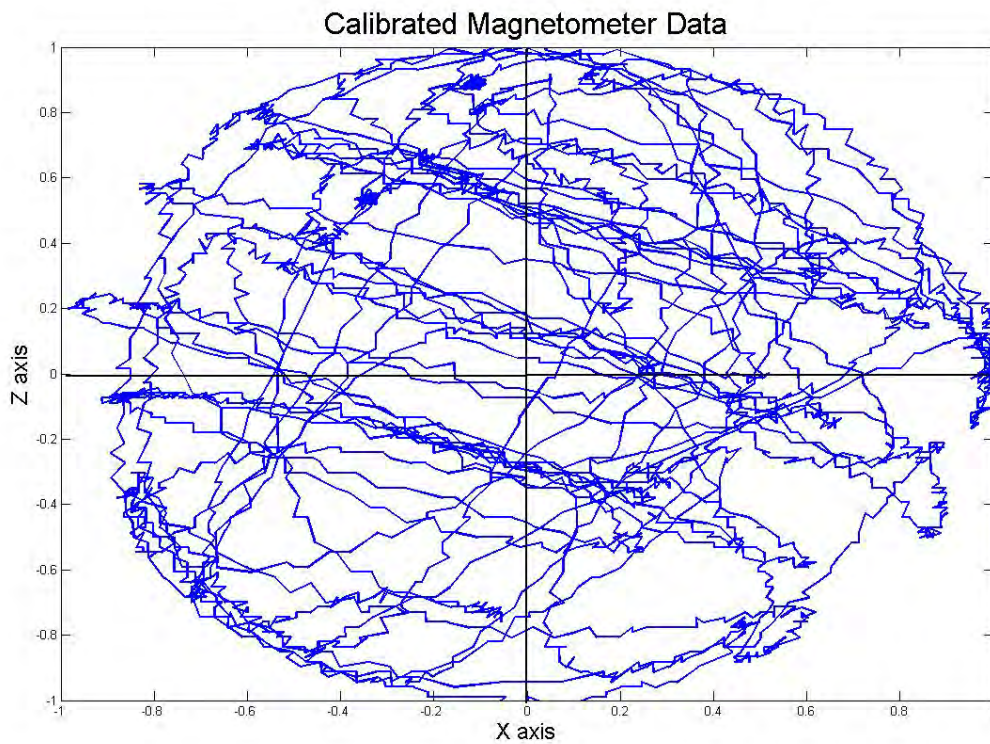
(b) calibrated magnetometer.

Figure A.2: Image showing the uncalibrated magnetometer data at the top and the calibrated data at the bottom. Note how the calibrated data is more spherical and centred around the origin. Also note the magnitude of the sphere. The black lines show the location of the origin.

The calibration consists of 10 parameters. Three are to model the hard iron effect, 6 model the soft iron effect and one takes into account the local magnetic strength [112] (the length of the magnetic vector).

To calibrate the magnetometer, the three axes of the magnetometer were logged for a few minutes while the magnetometer was rotated slowly in 3D space. The reason the magnetometer was rotated was to try and generate a sphere of measurements. The magnetometer calibration code was edited from [112].

A.2   Rotation Matrices

There are two different types of rotation matrices that are required for the Lagrange dynamic equations. The first is to rotate vectors from the body frame to the inertial frame and back again. The second type is to rotate angular rates from the body frame to the inertial frame and back again.

A.2.1   Relevant Versus Absolute Angle

There are two ways to define the angles in a system. There are relevant and absolute angles. A relative angle is when the attitude of a body is defined with respect to the previous body. An absolute angle is where the attitude of a body is defined with respect to the inertial frame.

To compare the difference in the complexity between relative and absolute angles, the position of the COM of the second beam, shown in Figure A.3, is calculated and compared. The positions are shown in Equation A.5 and, as can be seen, the relative equation is more complex. This is only a simple example and becomes more complex when there are more beams. The position is used to calculate the velocity, acceleration and the energy of the system. Therefore, by simplifying the position equation, one simplifies the equations of motion of the system.
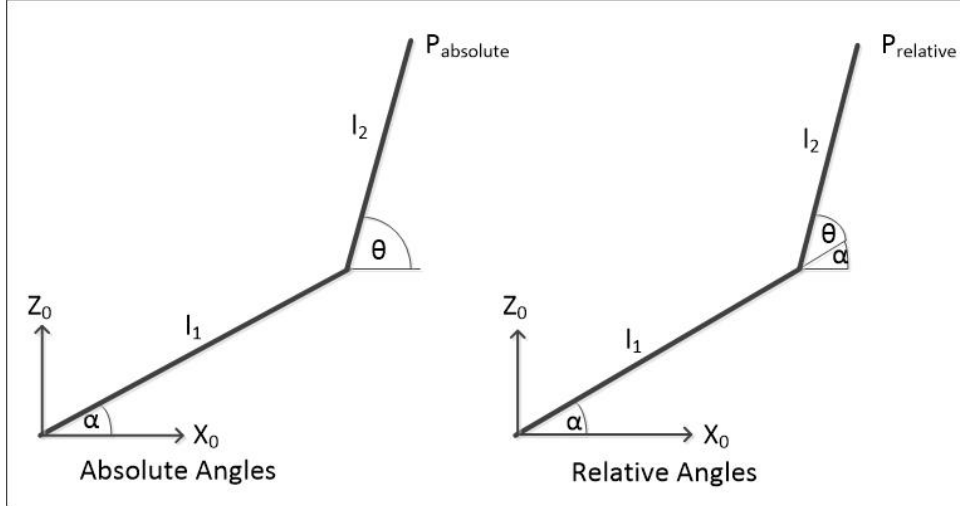
Figure A.3: Absolute angles versus relative angles.

$$\mathbf{p}_{relative} = \mathbf{R}_Z(\psi_1)\mathbf{R}_Y(\theta_1)\mathbf{R}_X(\phi_1)\mathbf{R}_Z(\psi_2)\mathbf{R}_Y(\theta_2)\mathbf{R}_X(\phi_2) \begin{bmatrix} l_2 \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_Z(\psi_1)\mathbf{R}_Y(\theta_1)\mathbf{R}_X(\phi_1) \begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{p}_{absolute} = \mathbf{R}_Z(\psi_2)\mathbf{R}_Y(\theta_2)\mathbf{R}_X(\phi_2) \begin{bmatrix} l_2 \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_Z(\psi_1)\mathbf{R}_Y(\theta_1)\mathbf{R}_X(\phi_1) \begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix}$$

$$(A.5)$$

In this dissertation, all the systems were modelled using absolute angles instead of relative angles.

Rotating Vectors

The Euler sequence that is used is the roll-pitch-yaw sequence [113]. The rotation matrix that is used is:

$$\mathbf{R}_{BtoI}(\phi, \theta, \psi) = \mathbf{R}_Z(\psi)\mathbf{R}_Y(\theta)\mathbf{R}_X(\phi)$$
$$\mathbf{R}_{ItoB}(\phi, \theta, \psi) = \mathbf{R}_{BtoI}(\phi, \theta, \psi)'$$

$$(A.6)$$

where the roll, pitch and yaw rotations are defined in Equation A.7 to A.9.

$$\mathbf{R}_X(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \tag{A.7}$$

$$\mathbf{R}_Y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{A.8}$$

$$\mathbf{R}_Z(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.9}$$

The *BtoI* stands for body frame to inertial frame and *ItoB* stands for inertial frame to body frame. If a particular angle is not used (for example if the rigid body cannot roll) then its individual rotation matrix is replaced with the identity matrix.

Rotating Angular Rates

To rotate angular rates the following formula was used [58]:

$$
\begin{aligned}
\boldsymbol{\omega}_{body} &= \mathbf{R}(\phi)\mathbf{R}(\theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi}_{inertial} \end{bmatrix} + \mathbf{R}(\phi) \begin{bmatrix} 0 \\ \dot{\theta}_{inertial} \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\phi}_{inertial} \\ 0 \\ 0 \end{bmatrix} \\
\boldsymbol{\omega}_{inertial} &= \begin{bmatrix} 0 \\ 0 \\ \dot{\psi}_{body} \end{bmatrix} + \mathbf{R}(\psi) \begin{bmatrix} 0 \\ \dot{\theta}_{body} \\ 0 \end{bmatrix} + \mathbf{R}(\psi)\mathbf{R}(\theta) \begin{bmatrix} \dot{\phi}_{body} \\ 0 \\ 0 \end{bmatrix}
\end{aligned}
\tag{A.10}
$$

This is explained in detail in Section 2.4.5.

A.3 Calculating Position, Velocity and Acceleration of a Rigid Beam

The position of the COM of each rigid body is defined in the body reference frame and then rotated into the inertial frame. The position of the COM in the body reference frame is shown in Equation A.11 and the position of the COM in the inertial frame is expressed as:

$$\mathbf{p}_{body_i} = \begin{bmatrix} l_i \\ 0 \\ 0 \end{bmatrix} \tag{A.11}$$

$$\mathbf{p}_{inertial_i} = \mathbf{R}_{BtoI}(\phi_i, \theta_i, \psi_i) * \mathbf{p}_{body_i} + \sum_{j=1}^{i-1} \mathbf{p}_{inertial_j} \tag{A.12}$$

where $\sum_{j=1}^{i-1} \mathbf{p}_{inertial_j}$ is the offset due to the previous bodies in the kinematic chain. The rotation matrix that is used to rotate the position is defined in Appendix A.2. The velocity of the COM is calculated by differentiating the position with respect to time (in the inertial reference frame) as follows:

$$\dot{\mathbf{p}}_{inertial} = \frac{\partial \mathbf{p}_{inertial}}{\partial \mathbf{q}} \frac{d\mathbf{q}}{dt} = jacobian(\mathbf{p}_{inertial}, \mathbf{q})\dot{\mathbf{q}} \tag{A.13}$$

The acceleration of the COM is calculated by differentiating the velocity of the COM as follows:

$$\ddot{\mathbf{p}}_{inertial} = \frac{\partial \dot{\mathbf{p}}_{inertial}}{\partial \dot{\mathbf{q}}} \frac{d\dot{\mathbf{q}}}{dt} = jacobian(\dot{\mathbf{P}}_{inertial}, \dot{\mathbf{q}})\ddot{\mathbf{q}} \tag{A.14}$$

A.4   Lagrange Matrices Calculation

The pseudo code to generate the $\mathbf{M(q)}$, $\mathbf{C(q,\dot{q})}$ and $\mathbf{G(q)}$ matrices are shown below. The $\mathbf{M(q)}$ matrix is the mass matrix, the $\mathbf{C(q,\dot{q})}$ matrix contains the Coriolis and centrifugal accelerations and the $\mathbf{G(q)}$ matrix contains the potential energy terms.

M: Mass Matrix

The Mass matrix is calculated as follows [46]:

$$\mathbf{M(q)} = jacobian(jacobian(T_{total}, \dot{\mathbf{q}})', \dot{\mathbf{q}}) = \frac{\partial (\frac{\partial T_{total}}{\partial \mathbf{q}})'}{\partial \mathbf{q}} \tag{A.15}$$

Where the Jacobian function is defined as follows:

$$jacobian(\mathbf{f}, \dot{\mathbf{x}}) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \cdots & \frac{\partial f}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \qquad (A.16)$$

C: Coriolis and Centrifugal Matrix

The following psudo code algorithm was used to calculate the $\mathbf{C(q,\dot{q})}$ matrix [46]:

$$
\begin{aligned}
& for \ \ i = 1 : length(\mathbf{q}) \\
& \quad for \ \ j = 1 : length(\mathbf{q}) \\
& \qquad for \ \ k = 1 : length(\mathbf{q}) \\
& \qquad\quad if(k == 1) \\
& \qquad\qquad \mathbf{C}(i,j) = 0.5 * (diff(\mathbf{M}(i,j), \mathbf{q}(k)) + diff(\mathbf{M}(i,k), \mathbf{q}(j))... \\
& \qquad\qquad\quad ... - diff(\mathbf{M}(j,k), \mathbf{q}(i))) * \dot{\mathbf{q}}(k); \\
& \qquad\quad else \\
& \qquad\qquad \mathbf{C}(i,j) = \mathbf{C}(i,j) + 0.5 * (diff(\mathbf{M}(i,j), \mathbf{q}(k)) + diff(\mathbf{M}(i,k), \mathbf{q}(j))... \\
& \qquad\qquad\quad ... - diff(\mathbf{M}(j,k), \mathbf{q}(i))) * \dot{\mathbf{q}}(k); \\
& \qquad\quad end \\
& \qquad end \\
& \quad end \\
& end
\end{aligned}
$$

$$(A.17)$$

Where the diff function is as follows:

$$\text{diff}(\mathbf{f}, \mathbf{x}) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \qquad (A.18)$$

G: Potential Energy Matrix

The following psudo code algorithm was used to calculate the $\mathbf{G(q)}$ matrix [46]:

$$
\begin{aligned}
& for \ \ i = 1 : length(\mathbf{q}) \\
& \quad \mathbf{G}(i) = diff(V_{tot}, \mathbf{q}(i)); \\
& end
\end{aligned}
$$

$$(A.19)$$

Where V_{tot} is the total potential energy of the system.

A.5 Generalized Forces

Using D'Alembert's principle and generalized forces, the aerodynamic force was modelled as a point force by integrating over the length of the beam. D'Alembert's principle is as follows:

$$Q_{generalized\ force_j} = \sum_i^n \mathbf{f}_i \cdot \frac{\partial \mathbf{r}_i}{\partial q_j} \tag{A.20}$$

The point force acts at the centre of pressure (COP). The variable, l_{COP}, is the distance from the pivot to the COP, where the point force is applied. The $\mathbf{r}$ vector contains all the positions of where the above forces act. The position is with respect to the origin in the inertial frame. The vector, $\mathbf{r}$, takes into account the translation of the second beam due to the rotation of the first beam. The force, $\mathbf{f}$, is defined as follows:

$$\mathbf{f}_{1_{inertial}} = \mathbf{R}_{BtoI}(0, \theta_1, \psi_1) \begin{bmatrix} 0 \\ f_{1_y} \\ f_{1_z} \end{bmatrix}$$

$$\mathbf{f}_{2_{inertial}} = \mathbf{R}_{BtoI}(0, \theta_2, \psi_2) \begin{bmatrix} 0 \\ f_{2_y} \\ f_{2_z} \end{bmatrix}$$

$$f_{1_y} = \frac{1}{2}\rho C_d A_1 \dot{\psi}_{1B}^2 l_{COP_1} \tag{A.21}$$

$$f_{1_z} = \frac{1}{2}\rho C_d A_1 \dot{\theta}_{1B}^2 l_{COP_1}$$

$$f_{2_y} = \frac{1}{2}\rho C_d A_2 \dot{\psi}_{2B}^2 l_{COP_2}$$

$$f_{2_z} = \frac{1}{2}\rho C_d A_2 \dot{\theta}_{2B}^2 l_{COP_2}$$

$$\mathbf{f} = \begin{bmatrix} \mathbf{f}_{1_{inertial}} & \mathbf{f}_{2_{inertial}} \end{bmatrix}$$

where f_{i_y} and f_{i_z} (with i=1,2) are the relevant aerodynamic force components in the body frame. The $\mathbf{f}$ vector contain all the applied point forces. The required rotation matrices are derived in Appendix A.2. To calculate the COP point and l_{COP}, the following formula was used:

$$l_{COP_1} = \frac{\tau}{F} = \frac{\int_0^{l_1} x f_1(x) dx}{\int_0^{l_1} p_1(x) dx}$$

$$f_1(x) = \frac{1}{2} \rho C_d \dot{\theta_1}^2 x$$

$$l_{COP_2} = \frac{T}{F} = \frac{\int_0^{l_2} x f_2(x) dx}{\int_0^{l_2} p_2(x) dx} \qquad \text{(A.22)}$$

$$f_2(x) = \frac{1}{2} \rho C_d \dot{\theta_2}^2 x$$

A.5.1   Generalized Force Matlab Code

To calculate the generalized forces the following pseudo code was used:

$$
\begin{aligned}
&\mathbf{F}_1 = \mathbf{R}_{BtoI}(\phi, \theta, \psi) * [F_x; F_y; F_z] \\
&\mathbf{F}_2 = \mathbf{R}_{BtoI}(\phi, \theta, \psi) * [F_x; F_y; F_z] \\
&\mathbf{r}_1 = distance \ to \ F_1; \\
&\mathbf{r}_2 = distance \ to \ F_2; \\
&\mathbf{f} = [F_1, F_2]; \\
&\mathbf{r} = [r_1, r_2]; \\
&counter = 1; \\
&for \ j = 1 : 1 : length(\mathbf{q}) \\
&\quad for \ i = 1 : 1 : length(f(1,:)) \\
&\quad\quad dr(:, counter) = diff(r(:,i), q(j)); \\
&\quad\quad \mathbf{Q}(j) = \mathbf{Q}(j) + sum(f(:,i).dr(:, counter)); \\
&\quad\quad counter = counter + 1; \\
&\quad end \\
&end
\end{aligned}
\qquad \text{(A.23)}
$$

Where $\mathbf{f}$ is the vector of point forces in the inertial frame and $\mathbf{r}$ is the vector of where the forces act with respect to the inertial frame.

A.6   Kalman Filter Measurement Equations

The Kalman filter update stage compares the sensor data to a prediction of the data using the current states of the filter. The prediction equations for the MEMs sensors that

are used are:

$$\textbf{accelerometer} = \mathbf{R}_{ItoB}(\phi,\theta,\psi) \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \mathbf{R}_{ItoB}(\phi,\theta,\psi)\ddot{\mathbf{p}}_{sensor}$$

$$\textbf{gyroscope} = \mathbf{R}_{gyroItoB}(\phi,\theta,\psi) \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{A.24}$$

$$\textbf{magnetometer} = \mathbf{R}_{ItoB}(\phi,\theta,\psi) \begin{bmatrix} magVecX \\ magVecY \\ magVecZ \end{bmatrix}$$

where the analytic equations for $\ddot{\mathbf{p}}$ are derived from Appendix A.3.

A.7   Numeric Implementation

To numerically integrate a variable, it was multiplied by the sampling time. The algorithm below was used to partially numerically differentiate with respect to one variable.

$$systemEquations1 = \mathbf{M(q)}'(-\mathbf{C(q,\dot{q})}_{+\epsilon} * \dot{\mathbf{q}}_{+\epsilon} - \mathbf{G(q)}_{+\epsilon} + \mathbf{Q}_{+\epsilon})$$
$$systemEquations2 = \mathbf{M(q)}'(-\mathbf{C(q,\dot{q})}_{-\epsilon} * \dot{\mathbf{q}}_{-\epsilon} - \mathbf{G(q)}_{-\epsilon} + \mathbf{Q}_{-\epsilon}) \tag{A.25}$$
$$f'_x = (systemEquations1 - systemEquations2)/(2*\epsilon)$$

Where the matrices are calculated with either ϵ being added to the variable or subtracted from the variable that you are partially differentiating with respect to.

A.8   TRIAD Algorithm Psudo Code

The following code is a psudo code algorithm for the TRIAD algorithm [109]:

$$
\begin{aligned}
\mathbf{g}_i &= [0; 0; 9.81]\\
\mathbf{m}_i &= [magVecX\ magVecY\ magVecZ]'\\
\mathbf{g}_b &= [accX\ accY\ accZ]'\\
\mathbf{m}_b &= [magX\ magY\ magZ]'\\
\mathbf{i}_r &= \mathbf{g}_i/norm(\mathbf{g}_i)\\
\mathbf{j}_r &= (cross(\mathbf{i}_r, \mathbf{m}_i))/norm(cross(\mathbf{i}_r, \mathbf{m}_i))\\
\mathbf{k}_r &= cross(\mathbf{i}_r, \mathbf{j}_r)\\
\mathbf{i}_b &= \mathbf{g}_b/norm(\mathbf{g}_b)\\
\mathbf{j}_b &= (cross(\mathbf{i}_b, \mathbf{m}_b))/norm(cross(\mathbf{i}_b, \mathbf{m}_b))\\
\mathbf{k}_b &= cross(\mathbf{i}_b, \mathbf{j}_b)\\
\mathbf{C}_{bn} &= [\mathbf{i}_b\ \mathbf{j}_b\ \mathbf{k}_b] * ([\mathbf{i}_r\ \mathbf{j}_r\ \mathbf{k}_r]')\\
\phi &= atan(\mathbf{C}_{bn}(2,3)/\mathbf{C}_{bn}(3,3))\\
\theta &= -1 * asin(\mathbf{C}_{bn}(1,3))\\
\psi &= atan2(\mathbf{C}_{bn}(1,2), \mathbf{C}_{bn}(1,1))
\end{aligned}
\tag{A.26}
$$

Where $\mathbf{g}_i$ is the gravity vector in the inertial frame, $\mathbf{g}_b$ is the measured gravity vector in the body frame, $\mathbf{m}_i$ is the magnetic vector in the inertial frame and $\mathbf{m}_b$ is the measured magnetic vector in the body frame.

A.9   Multibody Estimated Angles

The estimated angles for the *Multibody* Algorithm is shown in Figure A.4.

Figure A.4: 3D spine and tail *Multibody* estimated angles. The reader should note that the tail can only pitch and yaw.

A.10 Calculating r for the Applied Force

Below are the equations used to calculate where the applied force, due to the aerodynamic effect, acts. This position is described from the origin to the COP in the inertial frame. The rotation matrices are defined in Appendix A.2.

A.10.1 2D Tail System r Equation

$$
\begin{aligned}
\mathbf{r}_1 &= \mathbf{R}_{BtoI}(0, \theta_1, 0) \begin{bmatrix} l_{COP_1} \\ 0 \\ 0 \end{bmatrix} \\
\mathbf{r}_2 &= \mathbf{R}_{BtoI}(0, \theta_2, 0) \begin{bmatrix} l_{COP_2} \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_{BtoI}(0, \theta_1, 0) \begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix} \\
\mathbf{r} &= \begin{bmatrix} r_1 & r_2 \end{bmatrix}
\end{aligned}
\tag{A.27}
$$

A.10.2   3D Tail System r Equation

$$\mathbf{r}_1 = \mathbf{R}_{BtoI}(0,\theta_1,\psi_1)\begin{bmatrix} l_{COP_1} \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{r}_2 = \mathbf{R}_{BtoI}(0,\theta_2,\psi_2)\begin{bmatrix} l_{COP_2} \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_{BtoI}(0,\theta_1,\psi_1)\begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix} \tag{A.28}$$

$$\mathbf{r} = \begin{bmatrix} r_1 & r_2 \end{bmatrix}$$

A.10.3   2D Spine and Tail System r Equation

$$\mathbf{r}_1 = \mathbf{R}_{BtoI}(0,\theta_3,0)\begin{bmatrix} l_{COP_1} \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_{BtoI}(0,\theta_1,0)\begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_{BtoI}(0,\theta_2,0)\begin{bmatrix} l_2 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{r}_2 = \mathbf{R}_{BtoI}(0,\theta_4,0)\begin{bmatrix} l_{COP_2} \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_{BtoI}(0,\theta_1,0)\begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_{BtoI}(0,\theta_2,0)\begin{bmatrix} l_2 \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_{BtoI}(0,\theta_3,0)\begin{bmatrix} l_3 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{r} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 \end{bmatrix}$$
$$\tag{A.29}$$

A.10.4   3D Spine and Tail System r Equation

$$\mathbf{r}_1 = \mathbf{R}_{BtoI}(0,\theta_3,\psi_3)\begin{bmatrix} l_{COP_1} \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_{BtoI}(\phi_1,\theta_1,\psi_1)\begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_{BtoI}(\phi_2,\theta_2,\psi_2)\begin{bmatrix} l_2 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{r}_2 = \mathbf{R}_{BtoI}(0,\theta_4,\psi_4)\begin{bmatrix} l_{COP_2} \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_{BtoI}(\phi_1,\theta_1,\psi_1)\begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_{BtoI}(\phi_2,\theta_2,\psi_2)\begin{bmatrix} l_2 \\ 0 \\ 0 \end{bmatrix} + \dots$$

$$+ \mathbf{R}_{BtoI}(0,\theta_3,\psi_3)\begin{bmatrix} l_3 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{r} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 \end{bmatrix}$$
$$\tag{A.30}$$

A.11   Additional Results

The results for the tail *Multibody* algorithm are shown in Table A.1, the tail *Multibody and torque* algorithm in Table A.2 and Table A.3 for the tail *TRIAD KF* algorithm. The

spine and tail results for the *Multibody* algorithm are shown in Table A.4 and Table A.5 for the *Multibody and torque* algorithm. Note that the results that are quoted in this section are for the entire system of rigid bodies.

Table A.1: Results for the Tail *Multibody* algorithm

Data Set	RMS error (degrees)	RMS error (metres)
2D tail *Multibody*		
data 1	15.82	0.12
data 2	13.33	0.11
data 3	13.31	0.11
data 4	14.87	0.12
data 5	14.61	0.11
3D tail *Multibody*		
data 1	17.07	0.12
data 2	15.58	0.12
data 3	15.00	0.11
data 4	15.66	0.12
data 5	15.94	0.12

Table A.2: Results for the Tail *Multibody and torque* algorithm

Data Set	RMS error (degrees)	RMS error (metres)
2D tail *Multibody and torque*		
data 1	9.05	0.10
data 2	7.61	0.11
data 3	7.27	0.10
data 4	8.22	0.10
data 5	7.84	0.11
3D tail *Multibody and torque*		
data 1	8.85	0.10
data 2	7.55	0.12
data 3	7.35	0.10
data 4	8.13	0.11
data 5	7.77	0.11

A.12 One Beam Tail

Placing a number of sensors onto a wild cheetah, especially the tail which is rapidly flicking, is not feasible and could irritate the cheetah. Investigation into modelling the tail as a single beam was therefore conducted. The algorithms in Chapter 6 were modified for this case and described below. Only the 3D case was analysed.

Table A.3: Results for the Tail *TRIAD KF* algorithm

Data set	RMS error (degrees)	RMS error (metres)
2D tail *TRIAD KF*		
data 1	26.10	0.15
data 2	18.39	0.15
data 3	19.33	0.09
data 4	15.05	0.12
data 5	16.46	0.13
3D tail *TRIAD KF*		
data 1	42.08	0.21
data 2	40.72	0.22
data 3	40.67	0.18
data 4	42.40	0.22
data 5	42.12	0.23

Table A.4: Results for the Tail and Spine *Multibody* algorithm

Data set	RMS error (degrees)	RMS error (metres)
2D tail and spine *Multibody*		
data 1	7.76	0.08
data 2	9.40	0.08
data 3	5.31	0.05
data 4	8.25	0.06
3D tail and spine *Multibody*		
data 1	NA	0.14
data 2	NA	0.13
data 3	NA	0.12
data 4	NA	0.09

A.12.1   Algorithm 1: The *Multibody* Algorithm

The tail was modelled as a single beam that can pitch and yaw (no roll as discussed in the Methodology, Section 3.1). The Lagrange generalized coordinates are $\mathbf{q} = \begin{bmatrix} \theta_1 & \psi_1 \end{bmatrix}'$. The kinetic energy is shown below followed by the Lagrangian equation:

$$
\begin{aligned}
T_{translation} &= \frac{1}{2}(m_1 \dot{\mathbf{p}}_{1_{COM}}^T \dot{\mathbf{p}}_{1_{COM}}) \\
T_{rotational} &= \frac{1}{2}(J(\mathbf{R}_{gyroItoB}(\phi, \theta, \psi) \begin{bmatrix} 0 \\ \dot{\theta}_1 \\ \dot{\psi}_1 \end{bmatrix})^2)
\end{aligned}
\tag{A.31}
$$

$$
L = T_{translation} + T_{rotational}
\tag{A.32}
$$

Table A.5: Results for the Tail and Spine *Multibody and torque* algorithm

Data set	RMS error (degrees)	RMS error (metres)
2D tail and spine *Multibody and torque*		
data 1	8.21	0.08
data 2	9.74	0.08
data 3	5.68	0.05
data 4	8.92	0.06
3D tail and spine *Multibody and torque*		
data 1	NA	0.13
data 2	NA	0.11
data 3	NA	0.13
data 4	NA	0.15

The potential energy, applied torque due to the tail muscles and the generalized forces are set to zero. The equations of motion of the system that were derived in Chapter 2 are shown below:

$$\ddot{\mathbf{q}} = \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\psi}_1 \end{bmatrix} = \mathbf{M}(\mathbf{q})^{-1}(-\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}) \tag{A.33}$$

The $\mathbf{M}(\mathbf{q})$ and $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$ matrices are derived in Appendix A.4. The state variables of the Kalman filter along with the prediction and update states are shown below:

$$\mathbf{x}_k = \begin{bmatrix} \dot{\theta}_1 & \dot{\psi}_1 & \theta_1 & \psi_1 \end{bmatrix}' \tag{A.34}$$

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= f(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}) \\ \hat{\mathbf{x}}_{k|k-1} &= \hat{\mathbf{x}}_{k-1|k-1} + \Delta t \dot{\hat{\mathbf{x}}}_{k-1|k-1} \\ \mathbf{F} &= \frac{\partial f}{\partial \mathbf{x}}|_{\hat{\mathbf{x}}_{k|k-1}, \mathbf{u}_{k-1}} \end{aligned} \tag{A.35}$$

$$\begin{aligned} \mathbf{z}_k &= h(\mathbf{x}_k) \\ \mathbf{z}_k &= \begin{bmatrix} acc_{1_X} & acc_{1_Z} & mag_{1_X} & mag_{1_Y} & mag_{1_Z} & \omega_{1_Y} & \omega_{1_Z} \end{bmatrix}' \\ \mathbf{H}_k &= \frac{\partial h}{\partial \mathbf{x}}|_{\hat{\mathbf{x}}_{k|k-1}} \end{aligned} \tag{A.36}$$

where the measurement equations (acc_{1_X} etc.) are derived in Appendix A.6. The results of the algorithm are shown in Figure A.5.
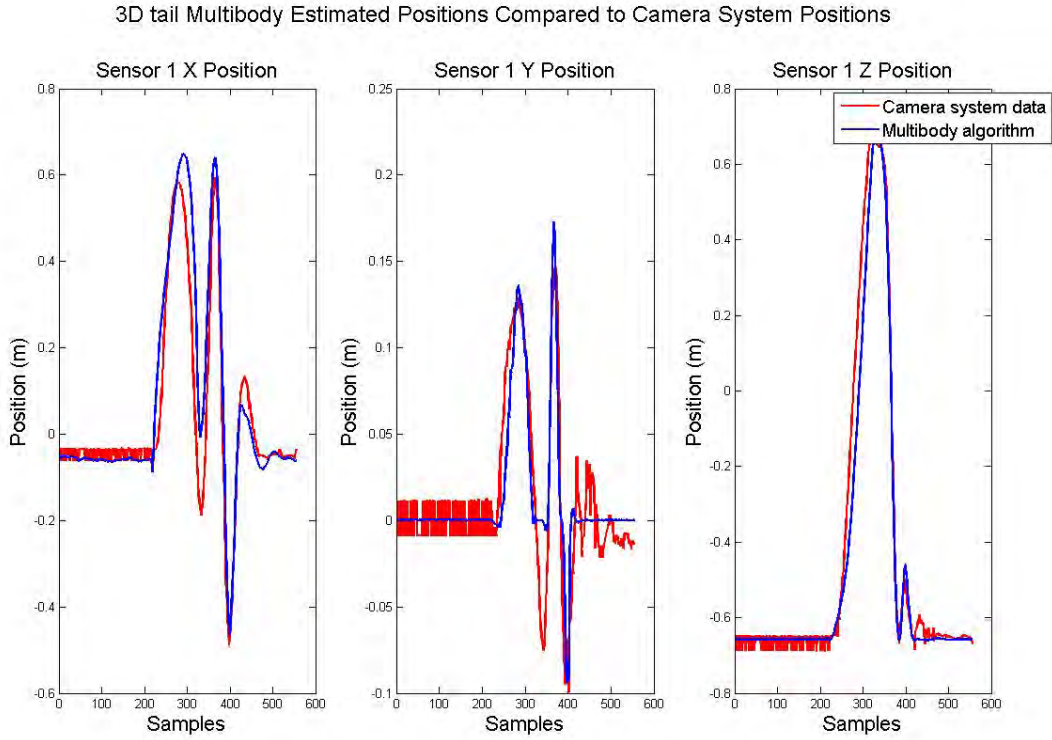
Figure A.5: 3D tail *Multibody* estimated position results.

More results for this algorithm are shown in Appendix A.12.3.

A.12.2 Algorithm 2: The *Multibody and Torque* Algorithm

The Lagrange generalized coordinates are $\mathbf{q} = \begin{bmatrix} \theta_1 & \psi_1 \end{bmatrix}'$. The kinetic and potential energy followed by the Lagrangian equation is as follows:

$$
\begin{aligned}
T_{translation} &= \frac{1}{2}(m_1 \dot{\mathbf{p}}_{1_{COM}}^T \dot{\mathbf{p}}_{1_{COM}}) \\
T_{rotational} &= \frac{1}{2}(J(\mathbf{R}_{gyroItoB}(\phi, \theta, \psi) \begin{bmatrix} 0 \\ \dot{\theta}_1 \\ \dot{\psi}_1 \end{bmatrix})^2) \\
U &= m_1 \mathbf{g} \mathbf{p}_{1_{COM}}
\end{aligned}
\tag{A.37}
$$

$$
L = T_{translation} + T_{rotational} - U \tag{A.38}
$$

The generalized forces are defined below:

$$\mathbf{Q}_{generalized\ force_j} = \sum_i^n \mathbf{f}_i \cdot \frac{\partial \mathbf{r}_i}{\partial q_j} \tag{A.39}$$

The aerodynamic force was modelled as a point force by integrating over the length of the beam. The point force acts at the centre of pressure (COP). The force, $\mathbf{f}$, is defined in Appendix A.5. The $\mathbf{f}$ vector contain all the applied point forces. The length, l_{COP}, is the distance from the pivot to the COP, where the point force is applied and is derived in Appendix A.5. The variable $\mathbf{r}$ is defined as:

$$\mathbf{r} = \mathbf{R}_{BtoI}(\phi, \theta, \psi) \begin{bmatrix} l_{COP_1} \\ 0 \\ 0 \end{bmatrix} \tag{A.40}$$

The $\mathbf{r}$ vector contains all the positions of where the above forces act. The position is with respect to the origin in the inertial frame. The $\mathbf{Q}$ matrix is combined with the applied torques.

$$\mathbf{Q} = \mathbf{Q}_{generalized\ force} + \begin{bmatrix} \tau_{1_{pitch}} \\ \tau_{1_{yaw}} \end{bmatrix} \tag{A.41}$$

The equations of motion are:

$$\ddot{\mathbf{q}} = \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = \mathbf{M}(\mathbf{q})^{-1}(-\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{G}(\mathbf{q}) + \mathbf{Q}) \tag{A.42}$$

The matrices are derived in Appendix A.4. The state variables of the Kalman filter are:

$$\mathbf{x}_k = \begin{bmatrix} \tau_{1_{pitch}} & \tau_{1_{yaw}} & \dot{\theta}_1 & \dot{\psi}_1 & \theta_1 & \psi_1 \end{bmatrix}' \tag{A.43}$$

The prediction and update equations are as follows:

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= f(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}) \\ \hat{\mathbf{x}}_{k|k-1} &= \hat{\mathbf{x}}_{k-1|k-1} + \Delta t \dot{\hat{\mathbf{x}}}_{k-1|k-1} \\ \mathbf{F} &= \frac{\partial f}{\partial \mathbf{x}}\Big|_{\hat{\mathbf{x}}_{k|k-1}, \mathbf{u}_{k-1}} \end{aligned} \tag{A.44}$$

$$\mathbf{z}_k = h(\mathbf{x}_k)$$
$$\mathbf{z}_k = \begin{bmatrix} acc_{1_X} & acc_{1_Z} & mag_{1_X} & mag_{1_Y} & mag_{1_Z} & \omega_{1_Y} & \omega_{1_Z} \end{bmatrix}' \qquad (A.45)$$
$$\mathbf{H}_k = \frac{\partial h}{\partial \mathbf{x}}|_{\hat{\mathbf{x}}_{k|k-1}}$$

Where the $\dot{\tau}_i$ states in $\dot{\mathbf{x}}$ are set to zero. The results of the algorithm are shown in Figure A.6. More results for this algorithm are shown in Appendix A.12.3.
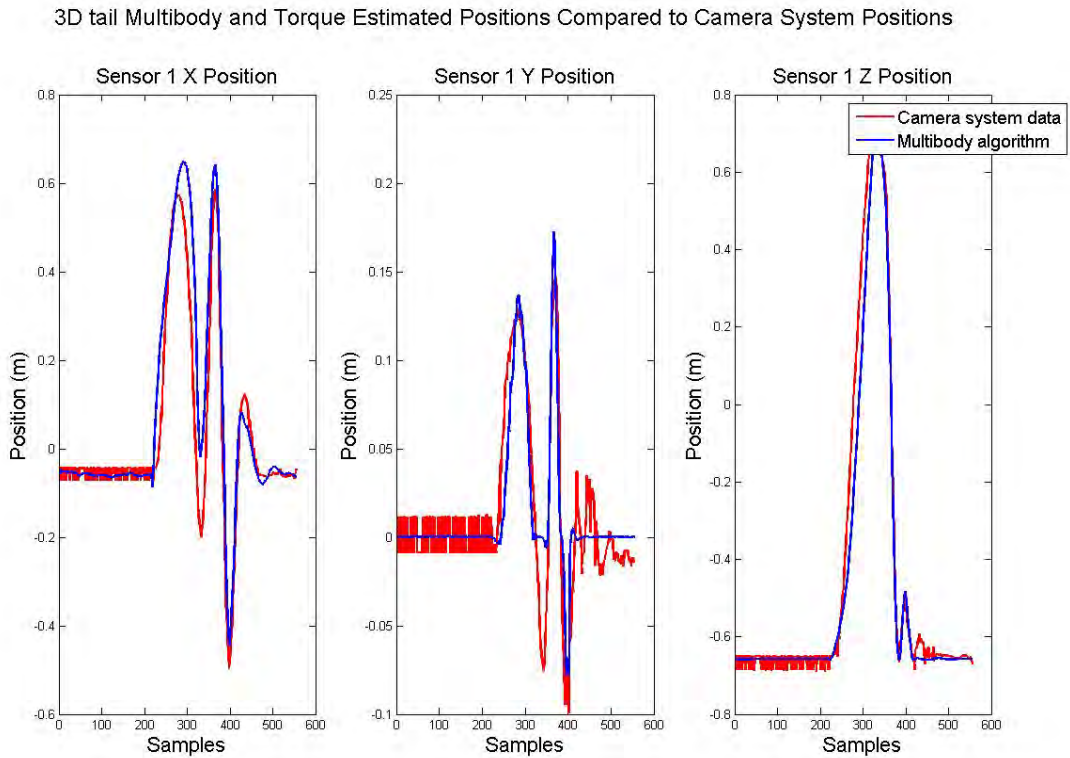
Figure A.6: 3D tail *Multibody and torque* position results.

A.12.3   Summary of Results

The Table A.6 shows the position errors of the algorithms on five different data sets.

The average RMS error for the one beam *Multibody* algorithm is 0.1242 compared to 0.1194 for the two beam algorithm. The average RMS error for the one beam *Multibody and torque* algorithms is 0.1214 compared to 0.1077 for the two beam algorithm. As can be seen that the results of the single beam algorithm are fairly accurate. This allows for fewer sensors to be placed on the cheetah and still maintain accurate results.

<div align="center">Table A.6: Single beam algorithm errors</div>

Multibody	RMS error in metres
data 1	0.13404
data 2	0.11836
data 3	0.11489
data 4	0.13092
data 5	0.12297
Multibody and torque	RMS error in metres
data 1	0.13265
data 2	0.11593
data 3	0.11256
data 4	0.12586
data 5	0.12022

A.13 GPS Filter

To determine the absolute position of the cheetah, a simple position filter was generated. The filter made use of the collar IMU and a GPS module. Due to the fact that this was tested indoors the data was simulated in Matlab. The GPS data was simulated at 10 Hz and the IMU data was simulated at 100 Hz. Noise was added to the data and the IMU was made to oscillate to simulate a cheetah running.

The state variables of the Kalman filter were $\mathbf{x} = \begin{bmatrix} \phi & \theta & \psi & v_x & v_y & v_z & p_x & p_y & p_z \end{bmatrix}'$ where the angles are the angles of the IMU sensor, the v terms are the linear velocity states and the p terms are the position states. The prediction and update equations are as follows:

$$\begin{aligned}
\hat{\mathbf{x}}_{k|k-1} &= f(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}) \\
\hat{\mathbf{x}}_{k|k-1} &= \hat{\mathbf{x}}_{k-1|k-1} + \Delta t \dot{\hat{\mathbf{x}}}_{k-1|k-1} \\
\mathbf{F} &= \frac{\partial f}{\partial \mathbf{x}}\big|_{\hat{\mathbf{x}}_{k|k-1}, \mathbf{u}_{k-1}}
\end{aligned} \tag{A.46}$$

$$\begin{aligned}
\mathbf{z}_k &= h(\mathbf{x}_k) \\
\mathbf{z}_1 &= \begin{bmatrix} a_x & a_y & a_z & \omega_x & \omega_y & \omega_z & mag_x & mag_y & mag_z \end{bmatrix}' \\
\mathbf{z}_2 &= \begin{bmatrix} a_x & a_y & a_z & \omega_x & \omega_y & \omega_z & mag_x & mag_y & mag_z & GPS_x & GPS_y & GPS_z \end{bmatrix}' \\
\mathbf{H}_k &= \frac{\partial h}{\partial \mathbf{x}}\big|_{\hat{\mathbf{x}}_{k|k-1}}
\end{aligned} \tag{A.47}$$

Due to the fact that the IMU update rate and GPS update rate were different, there were two update equations. The second update equation only runs when the GPS data is available.

A.14   Xsens Quote

Attached is the Xsens quote obtained on 2015-03-30.

Dear Callen Fisher,

Thank you for your request and your interest in our technology.

As you probably know, the Xsens MVN portfolio consists of full-body, wearable motion capture (Mocap) solutions, to be sure you have the full overview of our portfolio and all the option, see below all details:

The Hardware
We actually have 2 different products:

The Awinda = **6 990€ hardware price**
This System is fully wireless, all the Motion Trackers have their own battery inside, and a docking station to charge them.
This system is delivered with straps (include by default in the package), there is no Lycra suit for this model.
More convenient for many different size of people. All specs details below.

The Links = **11 500€ hardware price**
This system has all motion trackers wired inside the lycra suit (include by default in the package) and a battery pack on the back of the suit.
This system fit for high demanding environment.
All data are sent wireless to the computer.
 All specs details below.

Hardware specs:

Feature	MVN Link	MVN Awinda
Accuracy mocap	++	++/+
Low latency	✓	✗
Robust data link	✓	✗
Easy battery management	✓	✗
Battery life	9.5 h	6 h
Long range	50/150m	20/50m
GPS (future feature)	✓	✗
On-body buffering	120s	~10s
Internal update rate	1800Hz	1800Hz
Update rate	240Hz	60Hz
Accessible / actor friendly	✗	✓
Ideal for groups (education)	✗	✓
Hygiene / washing	✗	✓
Image (wireless is better?!)	✗	✓

Note that we can use both Links and Awinda together simultaneously.

The Software

Figure A.7: Xsens quote, Mon 2015-03-30, page 1

We have 2 versions of our software.
MVN Studio Standard
MVN Studio Pro
Our software version is delivered with a subscription, different model are available, see below all details
 Software specs:

Features	MVNS	MVNS Pro
Multi-person	✓	✓
Network Streamer (plug-ins)	✗	✓
Limited Configurations	✓	all
Limited Scenarios	✓	all
Limited Fusion Engine modes	✓	all
Customize scene	✗	✓
Hands-off calibration	✓	✓
Reference Camera support	✗	✓
Export as movie	✗	✓

Software Subscription for 1 year

Yearly License	MVN Awinda: € 6,990	MVN Link: € 11,500
MVN Studio : € 5,400	€ 12,390	€ 16,900
MVN Studio Pro : € 9,500	€ 16,490	€ 21,000

Subscription per year give you the access to the software and support for 1 year.
At the end of this period, you are not able to continue using the software you have to renew your subscription. Note that you can after a year upgrade to the lifetime license

Software Subscription for 3 years, lifetime license

Lifetime License	MVN Awinda: € 6,990	MVN Link: € 11,500
MVN Studio : € 13,900	€ 20,890	€ 25,400
MVN Studio Pro : € 22,500	€ 29,490	€ 34,000

This subscription give you the support for 3 years and include all the new features and new version. After the 3 years, if you do not want to renew your subscription, you can still use the license, as it is a lifetime license. Of course, you will be able to use the latest download eligible version you have.

It will be a pleasure to give you more in depth info on the systems and pricing.
Please let me know the best time to set up a call or a Skype demo.
Looking forward to hearing from you.
Regards

Sebastien TERRA
Entertainment Sales Manager
+31 88 97367 36 Work
+31 61 17273 63 Mobile

Figure A.8: Xsens quote, Mon 2015-03-30, page 2