**A Tale of a Tail**

by

Evan H. Chang-Siu

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Engineering - Mechanical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Masayoshi Tomizuka, Chair
Professor Robert J. Full
Professor J. Karl Hedrick

Spring 2013

**A Tale of a Tail**

## Abstract

A Tale of a Tail

by

Evan H. Chang-Siu

Doctor of Philosophy in Engineering - Mechanical Engineering

University of California, Berkeley

Professor Masayoshi Tomizuka, Chair

Rapid terrestrial locomotion is a fascinating but difficult problem that roboticists face. Considerations must be made towards dynamics, controls, estimation, the environment, electronics, and mechanical design. Wheeled and legged robots have been heavily researched and have gained much success, however they still have their limitations, especially when ground contact is lost. Inspired by the high performance of lizards actively using their tails to stabilize perturbations and rapidly reorient their body, we propose to add an active inertial appendage to terrestrial robots that will greatly improve their stability and maneuverability.

In this dissertation we present three main focus areas. We first develop the capabilities of a novel tailed robot with an active single degree-of-freedom (DOF) tail. We then discuss the single degree of freedom orientation sensing developed for this robot. Finally, we construct a nonlinear controller to enable reorientation of the body of a 2-link robot in three-dimensions while being constrained to a tail that only has two DOF of actuation.

The first 177 $(g)$ active tailed robot, *Tailbot*, has a single DOF of actuation and contains attitude inertial sensors. By utilizing both contact forces and zero net angular momentum maneuvering, this tailed robot can rapidly right itself in a fall, avoid flipping over after a large perturbation, and smoothly transition between surfaces of different slopes. We used a modeling approach to show that a tail-like design offers significant advantages to other alternatives, including reaction wheels, when the speed of response is important.

In order to provide the reliable orientation sensing for the single DOF robot we design a novel time varying complementary filter. Complementary filtering (CF) is a signal processing method that is commonly used for the fusion of gyroscope and accelerometer measurements in order to robustly estimate the attitude of a rigid body. The traditional CF uses linear time invariant filters to combine the two different measurements of the same physical quantity. Here we present an extension to the CF method with time-varying filters. A fuzzy logic method is developed to adjust the parameters. Stability analysis as well as experimental results are presented to verify the proposed method.

Finally, we propose a nonlinear control scheme for attitude control of a falling, two link active tailed robot with only two DOF of actuation. We derive a simplified expression for the robot's

angular momentum and invert this expression to solve for the shape velocities that drive the body's angular momentum to a desired value. By choosing a body angular velocity vector parallel to the axis of error rotation, the controller steers the robot towards its desired orientation. The proposed scheme is accomplished through feedback laws as opposed to feedforward trajectory generation, is fairly robust to model uncertainties, and is simple enough to implement on a low power microcontroller. We verify our approach by implementing the controller on a small 175 $(g)$ robot platform, enabling rapid maneuvers approaching the spectacular capability of animals.

To my dear friend Guido Saavedra, may you rest in peace.

# Contents

# List of Figures

# List of Tables

# Acknowledgments

I would like gratefully acknowledge my lovely and patient wife Kaori Noguchi and energetic and inspiring son Kenta Siu. You have both made my life complete.

I am deeply indebted to my family for their love and support.

Without the initial trust, continuing support, and thoughtful guidance of my advisors Prof. Masayoshi Tomizuka and Prof. Robert J. Full, this PhD would not have been possible. I thank you both tremendously.

I sincerely appreciate the help and discussion with all my close friends, collaborators, and labmates. Most notably: Mr. Sumio Sugita, Mr. Yasayuki Matsuda, Dr. Wenjie Chen, Prof. KC Kong, Dr. Hoday Stearns, Mr. Matthew Brown, Mr. Dennis Wai, Mr. Pedro Reynoso, Ms. Debbie Li, Mr. Jonathan Beard, Ms. Talia Moore, Mr. Yizhou Wang, Ms. Raechel Tan, Mr. Dominick Lim, Mr. Brendan Till, Mr. Janno Lunenburg, Mr. Mike Chan, Mr. Chen Xu, Prof. Joonbum Bae, Dr. Sanggyum Kim, Dr. Nancy Dong, Dr, Emma Yu, Prof. Sandipan Mishra, Dr. Benjamin Fine, Dr. Takashi Nagata, and Mr. Aaron Johnson.

My engineering education would not have been complete without the amazing supporting staff: Mr. Gordon Long, Mr. Mick Franssen, Mr. Scott McCormick, Mr. Thomas Clark, Mr. Rene Viray, and Mr. El Bennett.

I am deeply appreciative to my labs; the Mechanical System Control Lab (MSC) and the Center for Interdisciplinary Bio-inspiration in Education and Research (CiBER).

I would like to thank Prof. Liwei Lin, Prof. Andrew Packard, Prof. Roberto Horowitz, Prof. J. Karl Hedrick, Prof. Dan Koditscheck, Prof. Pieter Abbeel, Prof. Ron Fearing, and Prof. Hari Dharan for their advice and encouragement.

I would like to thank the IB135L class of 2010 and ME134 class of 2012 for their inspiration.

I would like to acknowledge the National Instruments for their generous donation of hardware and software, the funding from the United States Army Research Laboratory under the Micro Autonomous Science and Technology Collaborative Technology Alliance, and especially the National Science Foundation CiBER IGERT Award DGE-0903711.

Lastly, I would like to make a special dedication to my close friend and great collaborator Mr. Thomas Libby. This has been quite a journey and I look forward to working together in the future.

# Chapter 1

# INTRODUCTION

Robots can drive with treads [1], run with legs [2, 3], and climb with adhesive pads [4, 5, 6] over a variety of challenging surfaces. Despite these great advances in stable, steady running performance with a diverse array of body platforms, mobile robots still lag far behind animals in maneuverability and robustness to large perturbations. Some of these robots have fast, stable steady-state dynamics, even in the face of rough terrain, however, this is often due to carefully tuned passive dynamics, rather than feedback control [2, 3]. Large perturbations can push a robot outside the basin of stability of these passive designs, leading to catastrophic failure. To approach the robust behaviors of fast terrestrial animals, robots must not only have well tuned passive dynamics, but also be capable of rapid, sensory-driven feedback control.

A major difficulty for executing such control is that ground contact forces in rough terrain may be unpredictable or unavailable due to missed footholds, slippery substrates or unanticipated aerial phases. Recent studies in lizards have revealed a potentially elegant solution: they use their tails for stability, rapid reorientation, and perturbation recovery [7, 8], either through direct contact with the surface, or through zero net angular momentum maneuvers, similar to the well-known righting of a falling cat [9, 10], or of tumbling gymnasts [11, 12]. In both cases, attitude control is relatively simplified - torque applied through the tail yields an instantaneous, predictable counter-torque on the body. The result is remarkable performance for a terrestrial animal - geckos can right themselves in $0.106\ (s)$, effect large changes in pitch, roll, and yaw while mid air, and climb up a vertical wall at over 15 body lengths per second [7, 13].

Zero net angular momentum control has been extensively proposed in space robots and satellites, for both attitude control and manipulation [14, 15, 16], and reaction wheels have been employed for fine direction control in satellites [17]. Other robots have employed angular momentum for brachiation [18], hopping [19], and for landing control [20, 21]. Momentum-based control has appeared in terrestrial vehicles for zero-speed bicycle stabilization [22, 23], and has been implemented in a small robotic bicycle [24]. Recent robots have incorporated simple tails for stability in climbing [4, 5], and in a robot that runs on water [25]. These tails rely on passive substrate contact, but could be easily adapted to add the capability we show here.

We hypothesize that we can greatly expand the basin of stability for dynamic locomotion

by supplementing terrestrial robots with an actuated tail-like limb capable of both contact- and inertial-based torque inputs. Our objective in this dissertation is to test this hypothesis through the design, modeling, and control of a bio-inspired active tailed robot. Further, our motivation for this work is to show the added benefit a tail can have to existing mobile robot platforms. A major concern in design is that the tail and associated actuation be of relatively low mass cost and not necessitate a major redesign of the robot. An effective controller then should be computationally lightweight so as to not place excessive demands on the existing infrastructure.

As a first step toward capturing the incredible mobility of natural tails, we examined the capability of a single DOF tail to stabilize planar dynamic tasks by creating the first robot with a controllable active tail (Fig. 2.2). We show that a simple active tail with a simple controller can enable rapid reorientation and recovery from large perturbations. Further, we show that a tail-like design has significant advantages over other similar mechanisms, including reaction wheels and back-bending.

One fundamental challenge of using active inertial appendages for body control is sensing the orientation or attitude of the robot, which is primarily due to the limitations of sensors. For the single DOF robot we developed a novel time-varying complementary filtering (TVCF) method to estimate its pitch angle with respect to the horizon. While many sensors may work for this purpose, accelerometers and rate gyroscopes are well suited. Moreover, gyroscopes and accelerometers are appropriate for mobile applications because they are light weight, provide fast update rates, require low computation, and are low in cost. Since accelerometers can detect gravity, they can be utilized to estimate the direction of gravity when stationary. On the other hand, gyroscopes measure body angular rates and make it possible to estimate the attitude by integrating the signal. From these two sensing methods the same physical quantity, pitch or roll, can be estimated. Since each sensor provides different frequency domain characteristics, an improved result is expected if the two methods are fused intelligently.

The Kinematic Kalman filter (KKF) is a well known signal processing algorithm that can effectively fuse the signals. However, the design of the KKF requires knowledge of the type of noise characteristics, and it is optimal only when the noise is zero mean, white, and Gaussian distributed. In practice, the noise often exhibits more complicated characteristics, which deteriorates the optimality of the KKF. Complementary filtering (CF) is an alternative signal processing method that can be used for the same purpose [26, 27, 28]. The CF method has been widely applied to flying systems [29] and terrestrial robotics [30] due to its uncomplicated structure and effectiveness [28]. Also, when compared to the steady state KKF, CF has shown to be equivalent while requiring less computation [31]. In particular, the CF method is effective when the same physical quantity can be estimated from multiple sensing methods with different frequency characteristics.

The proposed TVCF method utilizes linear time-varying (LTV) filters for estimating the angle since the reliability of the sensors in the frequency domain is time-varying. For example, the accelerometer is most reliable during stationary conditions. The advantage of this method is that it improves the performance of the CF method with more degrees of freedom in the design of the filters. The main issue is determining the best conditions with which to adapt the LTV filters. A fuzzy logic scheme is proposed to formalize the qualitative conditions for adaptation. With

the inclusion of a triple axis magnetometer, this TVCF method has even been extended to three dimensions [32], but is not presented in this dissertation.

The extension, however, to control a robot in three dimensions is presented here. The single DOF tailed robot is limited to planar maneuvers, and perturbations to the body out of the plane of the tail's stroke are not stabilizable. The ability to swing their tails in any direction gives lizards the remarkable ability to rotate themselves during a fall and land on their feet from any orientation (Fig. 4.1). In contrast to cats and other mammals, lizards rely primarily on their tail for aerial reorientation, using quick sweeps to induce maneuvers in pitch or yaw, and cyclic spiraling motions to produce rolls or other elaborate motions. In this dissertation we also expand upon the single DOF and we show that a rigid tail with two DOF endows a simple robot (Fig. 4.2) with similar capability.

Much of the literature is concerned with space robots with articulated appendages [14, 15, 16]; only more recently has the idea has been applied to stabilizing terrestrial robots. Solutions to the zero angular momentum multi-link orientation control problem have generally relied on feedforward motion planning. Examples include an optimal control approach [33], Lie Bracket control for roll maneuvers [34], and gradient descent methods [35]. Model predictive control [36] and other computation-intensive online methods may work for space robots, but not for small, rapidly locomoting terrestrial robots, where falls last only fractions of a second.

We propose a purely feedback controller based on a reduced-complexity model of the system and an intuitive steering idea, and show effectiveness of the approach by simulation using the full model. We implement the controller on a small robotic platform (Fig. 4.2) weighing 175 $(g)$, with a cheap, commercially available microcontroller (Arduino Pro Mini). The controller runs at around 60 $Hz$, including sensing and state estimation. The robot is capable of reorientation from a multi-axis rotation with over 60° eigenangle error, in 100 $(ms)$, isolated pitch/yaw motions of 90°, or right itself from nearly upside-down (150°). Final error after maneuvers is relatively small; in practice the controller need only orient the bottom of the robot downwards so that legs or suspension can cushion the fall.

To highlight the initiation of research in active tailed robotics at this current point in time the 8.8 $(kg)$ legged robot XRhex has successfully been outfitted with a single DOF tail [37]. This robot is nearly two orders of magnitude larger and has similar performance. As with the single DOF, we believe this work will continue to initiate a trend towards research in the development and control of active inertial appendages in multiple degrees of freedom.

## 1.1 Dissertation Outline

The development of active tailed robotics is organized in this dissertation as follows. Note that Chapters 2-4 present the major contributions of the dissertation.

- Chapter 2: This chapter will cover the modeling, control, simulation, development, and experimental results of the single DOF *Tailbot*. The effectiveness at mitigating perturbations and reorienting in midair is shown in the sagittal plane.

- Chapter 3: This chapter will detail the proposed novel time-varying-complementary-filter sensing algorithm that is used for the single DOF. This method works well for sensing orientation in a single DOF and is general enough to be applied to more than the application of *Tailbot* from Chapter 2.
- Chapter 4: This chapter will develop the modeling, control, simulation, development, and experimental results of the two actuated DOF *Tailbot*. In this chapter a nonlinear controller is analytically justified and verified through simulation. The controller is then implemented on a two-actuated-DOF robot. The robot is able to correct in roll, pitch and yaw simultaneously and account for singularity conditions.
- Chapter 5: Here concluding remarks will be made as well as a look toward open issues and future work.

# Chapter 2

# 1DOF *Tailbot*

## 2.1 INTRODUCTION

In this work featured in [38], we were inspired by the robust transition behavior of agama lizards to examine whether an active tail could expand the performance of a terrestrial robot. Figure 2.1 shows an agama performing a rapid transition over a low-friction substrate. When the feet slip during the jump, the perturbation in ground reaction forces generates angular momentum, which causes the animal to tumble forward through the air. Unchecked, this momentum would rotate the lizard's body outside of the range where it could safely land on the wall. To prevent the body from rotating, the animal rapidly swings its tail towards its head, transferring momentum from one link to the other. This behavior enables a significantly higher success rate than would be predicted for animals without a tail [8].

As a first step toward capturing the incredible mobility of natural tails we examine the capability of a single DOF tail to stabilize planar dynamic tasks by creating the first robot with a controllable active tail (Fig. 2.2). Here we show that a simple active tail with a simple controller can enable rapid reorientation and recovery from large perturbations. Further, we show that a tail-like design has significant advantages over other similar mechanisms, including reaction wheels and back-bending.

This chapter is organized as follows. In section 2.2 we cover the setup of the dynamical modeling of the tailed system and compare in simulation an active tail to an active flywheel. In section 2.3 we detail the design of the robot. In section 2.4 the experimental results are covered and lastly in section 2.5, the conclusions are drawn.

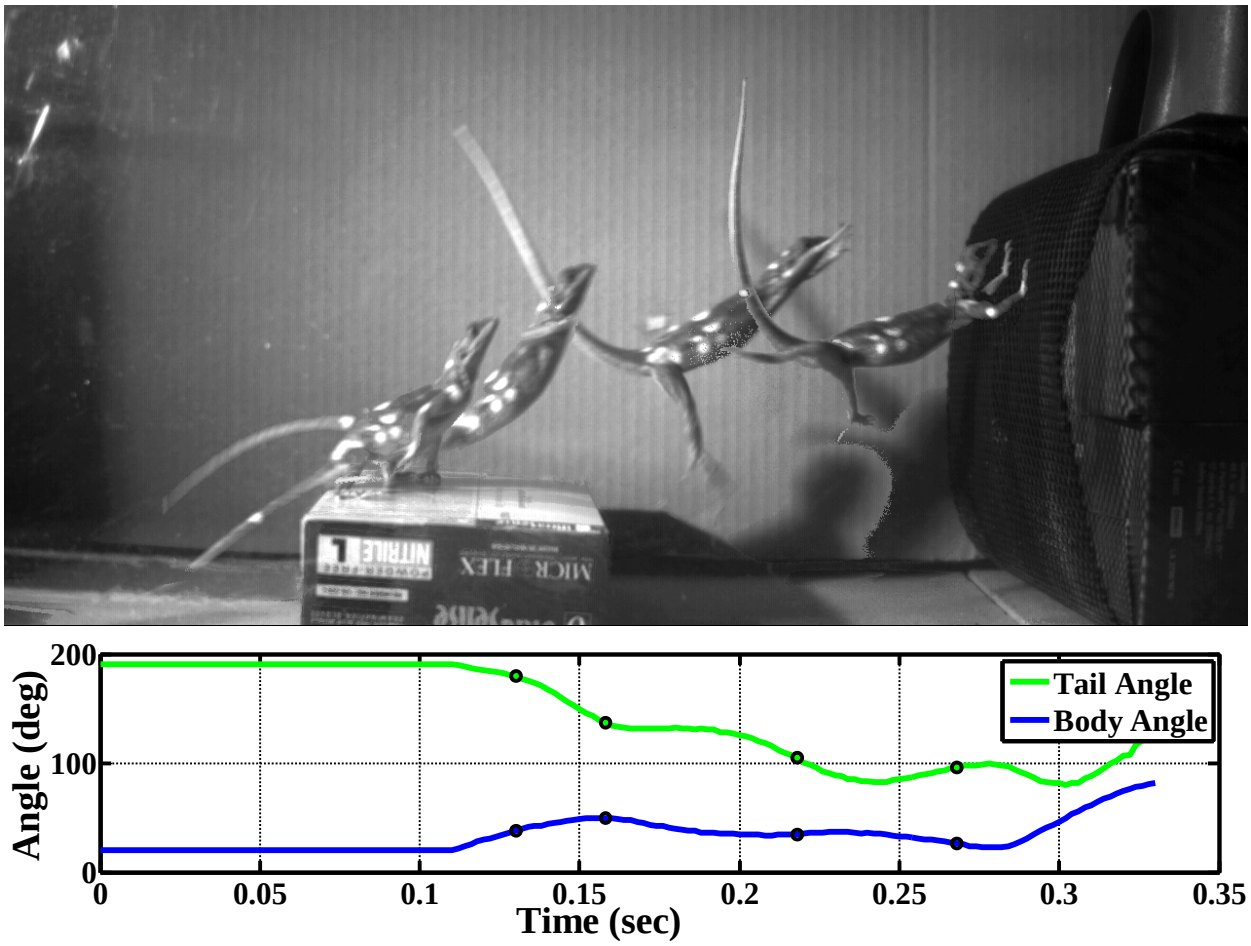



Figure 2.1: Overlay plot of a lizard jumping off a low-friction surface. Note the use of tail indicated by the green line during flight, the near constant body angle indicated by the blue line, and the successful landing on the wall.

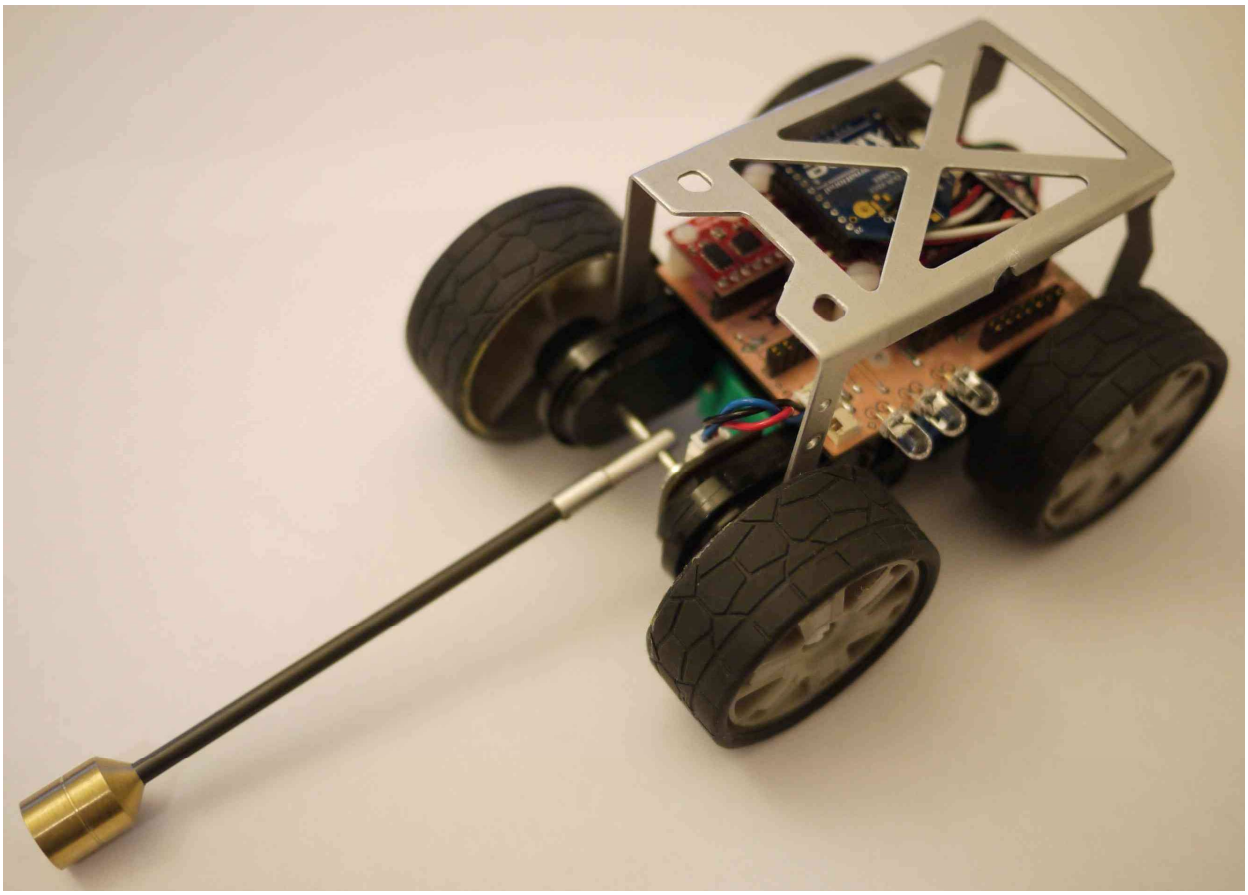


Figure 2.2: Single actuated DOF *Tailbot*, prototype of tailed car.

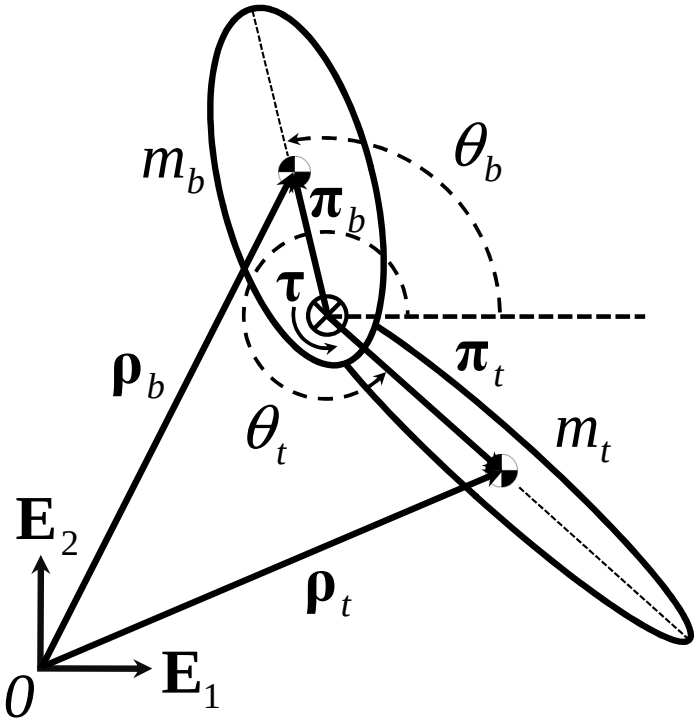



Figure 2.3: Diagram of the two link planar model for aerial reorientation.

## 2.2 Modeling and Analysis

### 2.2.1 Equations of Motion

To inform the design of our robot, we derived a planar two-link rigid body model for aerial reorientation with the absolute body and tail angles referenced to horizontal as shown in Fig. 4.3. The two bodies are joined by a pin, where relative torque, $\tau$, is provided by a DC motor at the pivot.

For ease of simulation, we developed a state space model of the system in the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\tau \tag{2.1}$$

where the states, $\mathbf{x} = [\theta_b, \dot{\theta}_b, \theta_t, \dot{\theta}_t]^T$, consist of the angular position and velocity of the links. The details of which are shown in Appendix A.

### 2.2.2 Active Tail vs. Active Flywheel

Compared to the tail-like design we present here, disk-like inertial loads, [17], [24], are not limited in stroke. This is a clear advantage for momentum wheels in situations that demand large rotations of the body, or when angular momentum must be stored for long periods. However, disk-like

mechanisms face reasonable limits on size; the diameter of the disk must be significantly less than the size of the vehicle. This makes it difficult to increase the MOI since MOI is quadratically dependent on length while only linearly dependent on mass. A tail could reasonably be several times the length of a robot; indeed lizards with tails over twice body length are common [39]. Hence, to achieve the same rotational impulse with comparable weight, a disk-like design will utilize small MOI and large angular velocity, while a tail-like design will use large MOI and low speed.

The negative effects of increased tail length are not immediately clear, but could include increased weight to maintain sufficient stiffness and decreased mobility. Spinning a disk to high speed requires relatively high mechanical power, generates vibration, and results in gyroscopic effects that may reduce maneuverability. Rotations in multiple axes would require multiple disks, or a compact mechanism to rotate a single disk to an arbitrary axis. Studies in gliding geckos have revealed that a single tail can effect rotations in all three axes with only two DOF [7]. The packaging flexibility afforded by tail-like inertial appendages can result in much larger inertia for the same weight.

We used our dynamic model to examine the relative performance of a flywheel design for our test platform over a range of reasonable masses and lengths. The COM of the flywheel was positioned at the COM of the vehicle body. We limited the maximum allowable diameter to fit within the dimensions of the robot. With comparable mass and tail length/wheel radius, performance offered by the two geometries is nearly identical; differences are primarily due to the fact that the tail is not attached to the COM of the robot. However, the overall dimension (diameter) of the wheel must be twice the length of the tail for comparable performance. Practical applications would likely tolerate diameters of less than one half body length. Therefore, when the diameter of the disk is exactly half the length of the body and the mass of the disk is reasonably 10% of the body, from Fig. 2.4, it would take 0.066 $(s)$ to rotate the body 45°. For a tailed design the length can be made arbitrarily long and for the case when the relative tail length is equivalent to body length then half of the previous rise time is achieved, 0.033 $(s)$, as shown in Fig. 2.5. For the disk to acquire the same time performance its mass would have to be increased to twice the body mass.

Our model predicts that momentum wheels are ideal for applications where slow, precise adjustments are necessary, as in satellites or for static balancing. For rapid, dynamic adjustments of less than one half rotation, a tail can be more compact and significantly lighter. Tails have another significant advantage over momentum wheels for dynamic robots - the ability to contact the substrate and change the total angular momentum of the system. Geckos were observed using small motions of their tails to contact the substrate and prevent pitch-back during rapid climbing [13]. We found tail-substrate interaction in our robot to be beneficial, particularly in reducing the drawback of limited tail stroke.

Another proposed mechanism of zero net angular momentum maneuvering is exemplified in the well-known falling cat [9] and the robots and models which have followed its design, where back-bending is the primary source of internal momentum. Compared to a tail-like inertial mechanism, in back-bending, the two links are similar in mass and length, and appendages or wheels are mounted on both links. This mechanism enabled air-righting in a robot [34], though the response

was significantly slower than that observed in lizards. As observed by [34], the primary disadvantage is that in this case the orientation of the second link is also important; this constrains the performance of reorientation and makes pitch adjustments like those we show here impossible.

## 2.3 Robot Design

### 2.3.1 Mobile Robot Platform

To explore the effects of adding an active tail to an existing robot, we modified a small terrestrial vehicle in the form of a commercial radio-controlled car (*FlipZ, Radio Shack*). The car was designed with two independent DC motors to drive the left and right sets of wheels. We reconfigured the gear trains such that one motor drove the front wheels, while the other drove a single revolute joint at the rear, onto which we secured a $5/32$" $(0.397\ (cm))$ diameter carbon fiber tube and brass tip. The total tail stroke available was $255^\circ$ beginning at $70^\circ$ between the body and tail when fully pitched up as shown in Fig. 2.6. We measured the moment of inertia for the body and tail following the pendulum method in [40]. The final parameters of the robot are listed in Table 2.1.

The preexisting control circuitry was replaced with custom electronics that included a microcontroller (*Arduino*, pro-mini, 3.3 $(V)$, 8 $(MHz)$), a motor driver (*Pololu* FNG-6612), a three-axis accelerometer (*Analog Devices*, ADXL345, $\pm 2$ $(gees)$) and gyroscope (*Invensense*, IDG-3200, $\pm 2000$ $(^\circ/s)$), wireless communication module (*Xbee*), and LED indicator lights for timing and visualizing control effort. The robot was powered with two lithium polymer 138 $(mAh)$ batteries. Finally, a roll cage was fabricated to protect the electronics.

### 2.3.2 Tail Design

Our intent is to describe potential benefits to adding a tail to an existing platform; hence we constrained the design space to variables pertaining to the tail only, specifically motor power, gear ratio (input:output speed), and tail dimensions (length, mass, and moment of inertia). By recognizing that motor choice would be constrained by mass and packaging, we fixed motor torque and speed. We idealized the tail as a thin rod with a point mass at its end, eliminating tail MOI, and used the gear ratio that minimized maneuvering time for each combination of mass and length. Hence the final design space for our simple tail was two dimensional.

We used our dynamic model to evaluate maneuvering performance over a range of reasonable values for mass and length (2% to 20% body mass, one half to twice body length). To eliminate the effects of controller performance, we simulated aerial maneuvering with maximum open-loop torque. We measured the time required for the body to rotate $45^\circ$, along with the corresponding relative tail stroke required for the maneuver as shown in Fig. 2.7 when the links are initially $180^\circ$ apart. For each combination of mass and length, we selected the gear ratio that yielded fastest rise time using a Nelder-Mead simplex direct search (*fmincon, MATLAB*). Optimal gear ratio varied significantly with tail geometry, from $16:1$ in the lower left corner of the design space of Fig. 2.5,

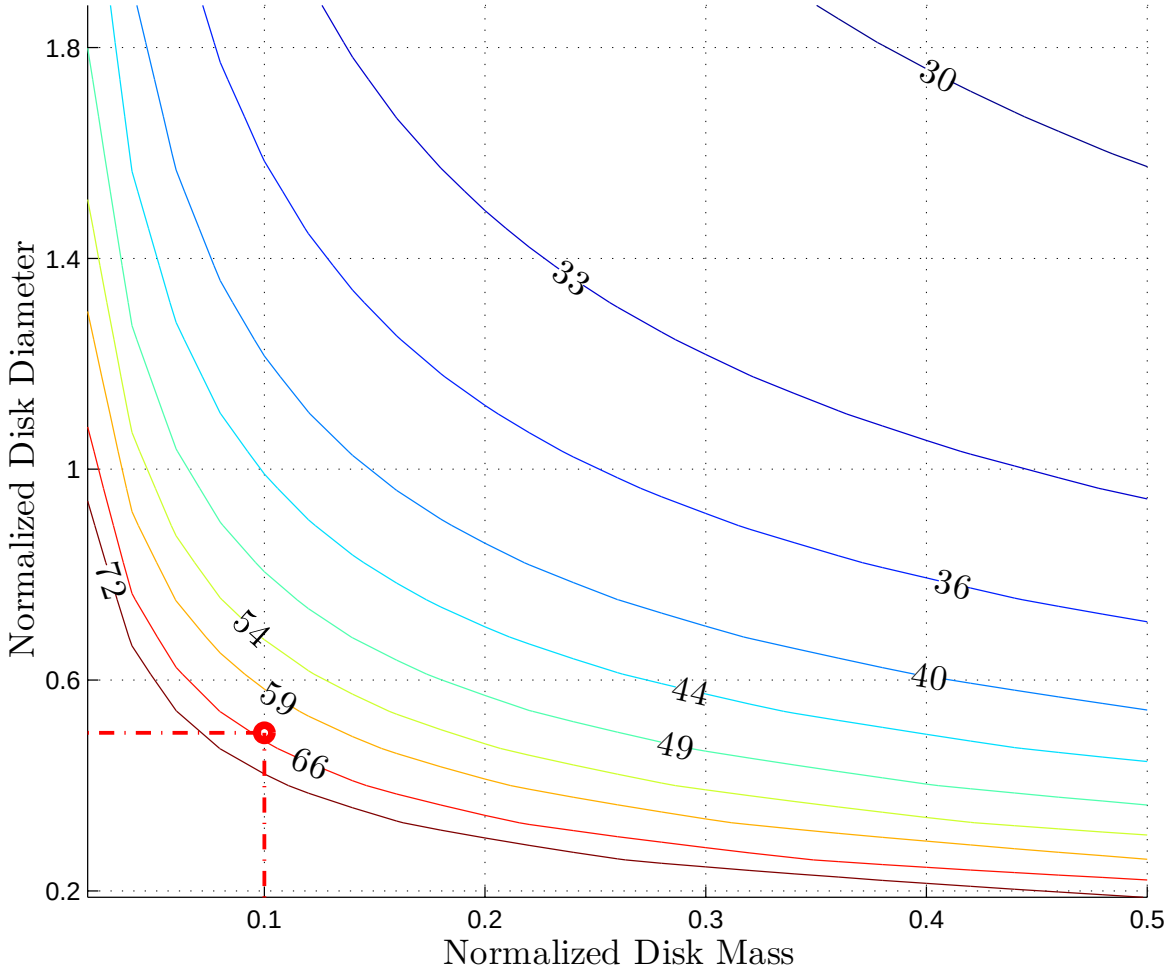



Figure 2.4: Contour plot of time in milliseconds for the body to rotate 45° using a *disk* design with disk mass and diameter normalized against body mass and length respectively. The red dot indicates the rise time (66 *ms*) when the relative mass of the tail is 10% and the diameter of wheel is half the length of the body. As relative disk mass and length increase, time decreases.

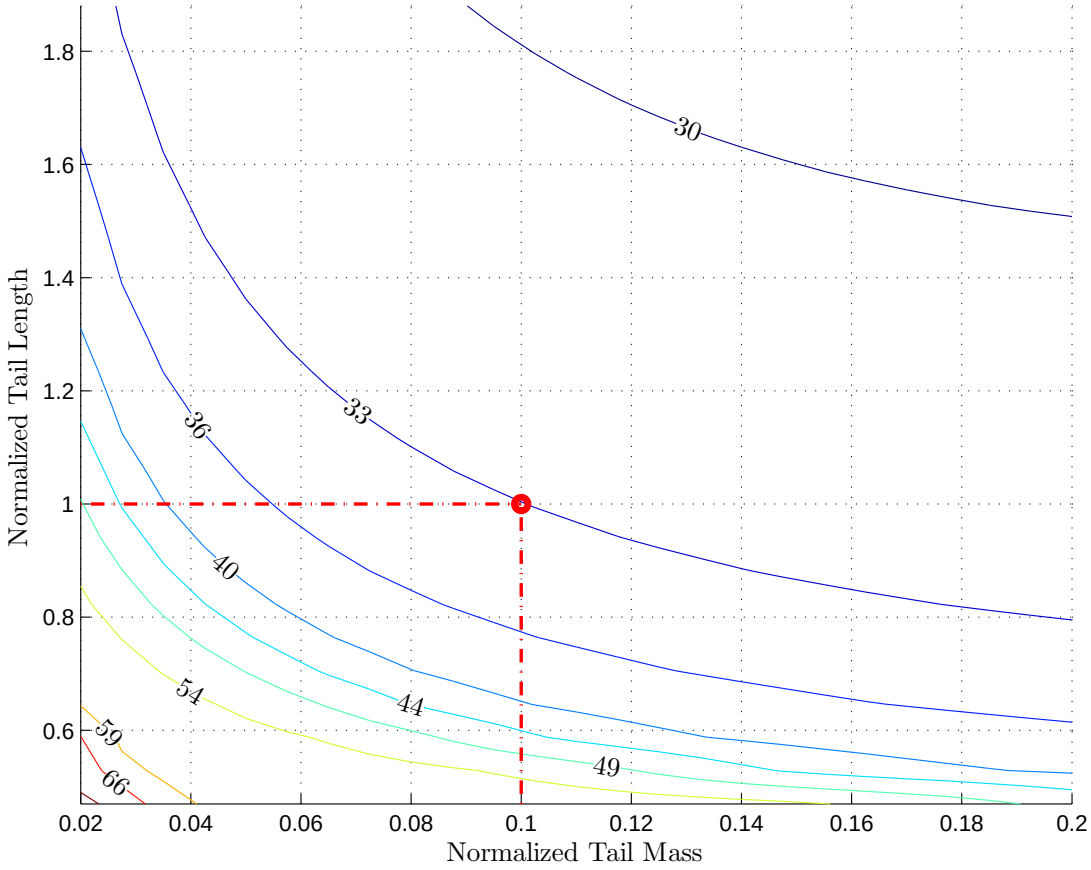



Figure 2.5: Contour plot of time in milliseconds for the body to rotate 45° using a *tail* design with tail mass and length normalized against body mass and length respectively. The red dot indicates the rise time (33 *ms*) when the relative mass of the tail is 10% and the length of tail is equal to the length of the body. As relative tail mass and length increase, time decreases.

to 90 : 1 in the upper right. Both rise time and tail stroke were sensitive to changes in mass and length; in general, increasing either resulted in decreases in both rise time and required tail stroke.

The actual constraints on mass and length for a mobile robot in the field would be greatly task- and platform-dependent. While a climbing robot may prioritize low weight, a vehicle meant for tight spaces or rough terrain may require a shorter tail. In practice, tail stroke will be limited by the physical dimensions of the body. In this case, stroke may be used as a third constraint - the maximum allowable tail stroke for desired body stroke will yield a locus of points in the relative mass-length plane as shown in Fig. 2.7. Given the lack of real performance constraints in our robot, we chose to fix tail mass to 10% of body mass, and to inspire relative stroke requirements from the agamas, which on average use a relative tail stroke of two times body stroke (ie, absolute body and tail stroke will be roughly equal for small rotations). These two constraints lead to a robot tail length (with point mass at tip) of 10.3 $(cm)$, about $0.9X$ body length. Our chosen tail geometry results in a predicted open loop rise time of 0.033 $(s)$. Optimal gear ratio was about 60 : 1. In the parameter range chosen, performance was relatively insensitive to design changes. Increasing tail length by 10% would decrease rise time 2% and tail stroke by 14%; increasing tail mass 10% would decrease rise time 1% and tail stroke 6%. In general the quadratic dependence of inertial loads on length results in more sensitivity to length change than mass change.

Due to space constraints on our existing platform, we were unable to accommodate the ideal gear ratio for the tail motor; we instead used a ratio of 17 : 1. While our simulation assumed a massless rod, the actual tail's mass is 17 $(g)$ (10% body mass) with rod and coupling, has an MOI of $1.4 \times 10^{-5}$ $(kgm^2)$, and an overall length of 12.2 $(cm)$ ($1.05X$ body length) as listed in Table 2.1. These concessions result in slightly reduced performance; the model predicts a relative tail stroke of 47° and estimates the robot will need 0.05 $(s)$ to complete the maneuver. The agamas in [41] had tails that weighted about 20% body mass, and were $1.3X$ body length on average. While lizard tails are roughly conical, we recognize that animal morphology is subject to a variety of other constraints, including development, musculature, and the multi-functional nature of tails. With the inclusion of the actuator (geartrain and motor), which was placed on the body, the total mass to add a tail was 28 $(g)$ or 18.7% of the body without actuator. Hence, while our engineered

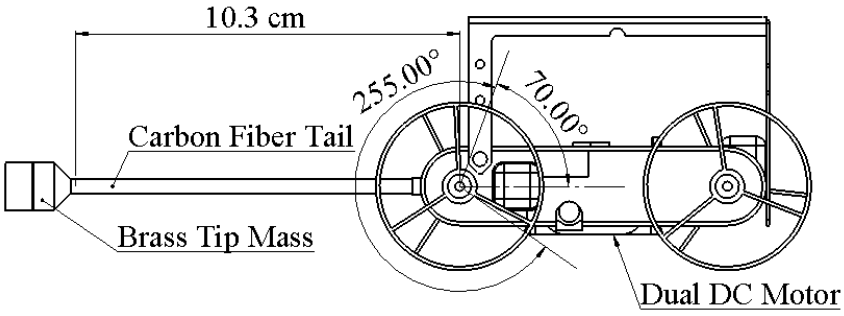



Figure 2.6: Side view of *Tailbot* displaying the motor location, tail design, and tail stroke. The specified distance between the COM of the tip mass to center of the pivot is 10.3 $(cm)$.

Table 2.1: *Tailbot* Measurements.

| Parameter | Value |
|---|---|
| overall size | $2.8 \times 10^{-1} \times 9.3 \times 10^{-2} \times 7.0 \times 10^{-2}$ $(m)$ |
| stall torque | $5.0 \times 10^{-3}$ $(Nm)$ |
| no load ang. vel. | $5.4 \times 10^{3}$ $(rad/s)$ |
| gear ratio | $17.28 : 1$ |
| max voltage | $8$ $(V)$ |
| body length | $1.17 \times 10^{-1}$ $(m)$ |
| tail length | $1.27 \times 10^{-1}$ $(m)$ |
| body COM to pivot | $4.5 \times 10^{-2}$ $(m)$ |
| body mass | $1.6 \times 10^{-1}$ $(kg)$ |
| body COM inertia | $1.54 \times 10^{-4}$ $(kgm^2)$ |
| tail geartrain & motor | $1.11 \times 10^{-2}$ $(kg)$ |
| tail COM to pivot | $1.03 \times 10^{-1}$ $(m)$ |
| tail mass | $1.7 \times 10^{-2}$ $(kg)$ |
| tail COM inertia | $1.4 \times 10^{-5}$ $(kgm^2)$ |

tail yields similar kinematic performance to that of the natural system, our design uses slightly less mass but 19% less overall length relative to body.

### 2.3.3 Sensing and Feedback Control

Since there is only one actuator for two DOF, the system is underactuated and the angle of both body and tail cannot be controlled simultaneously. At this stage, the orientation of the body will be deemed most important; hence we ignore tail angle and simply close the loop on body angle. In Chapter 4 we extend actuation to more degrees of freedom and use the tail angle information in the control scheme.

Over short time scales, the body angle can be accurately estimated from a gyroscope, but integration drift will accumulate error over time. Alternatively, an accelerometer can estimate orientation from the angle of the gravity vector with respect to body-fixed axes, but this estimate is sensitive to other accelerations and is not available during free fall. We devised a time varying complementary filtering (TVCF) method, which is discussed in Chapter 3 and [42] and the following chapter to combine these measurements and robustly estimate the instantaneous body angle. We

collected kinematic data at 500 frames-per-second ($fps$) with a high speed camera (*HS1, Fastec Imaging*) and compared actual body angle to the estimate from the TVCF method. Results show that the method achieved a root-mean-square (RMS) error of less than 10°.

As a first approach to an active tail controller, we implemented a proportional-derivative (PD) controller based on the sensor fusion estimate of the body angle. The input to the controller was the desired body angle, $\theta_{ref}$. The gains were manually tuned to produce acceptable rise time and settling time performance given a step signal at $\theta_{ref}$. The controller was unaware of tail angle, except for limit switches that prevented the motor from applying torque against the tail's bump stops. We found this simple controller sufficient for the rapid, gross maneuvers of interest in

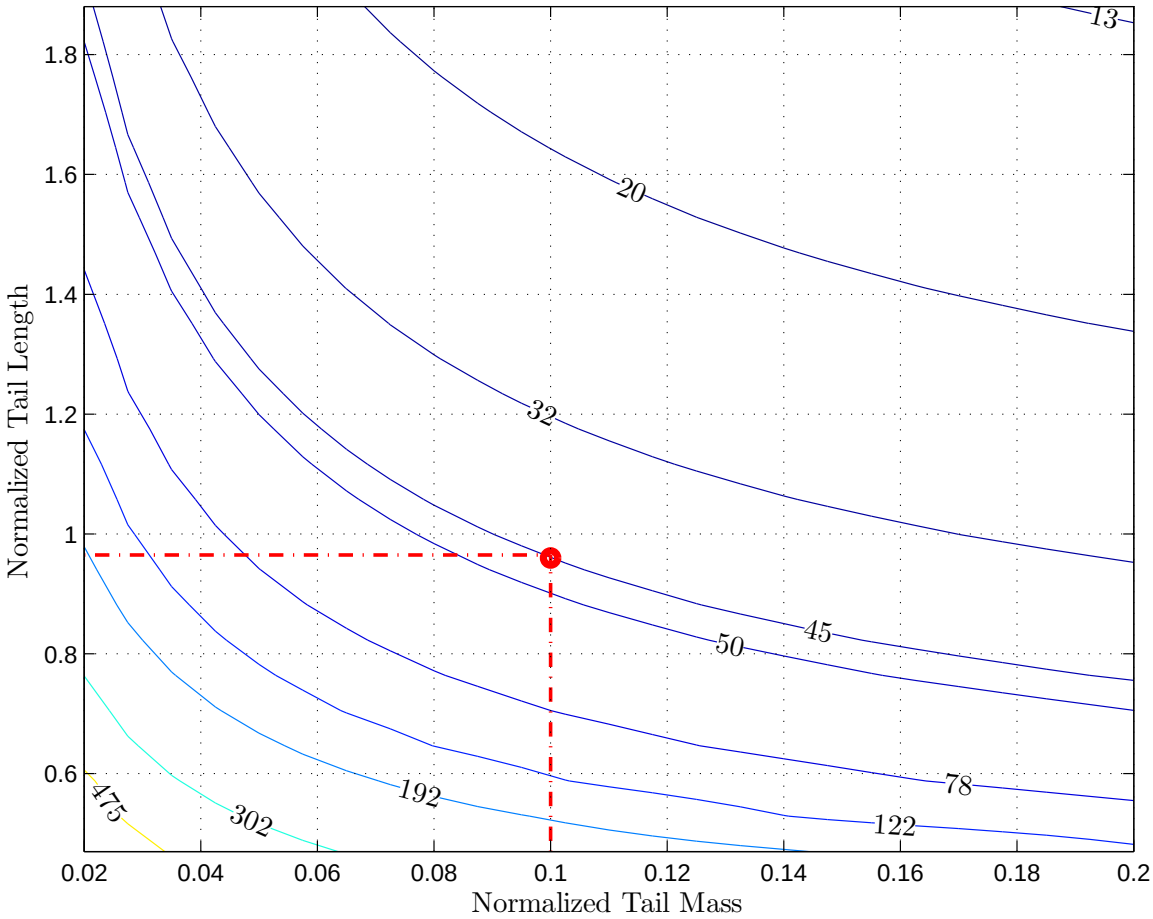



Figure 2.7: Contour plot of absolute tail angle in degrees for the body to rotate 45° using a *tail* design with tail mass and length normalized against body mass and length respectively. The red dot indicates the design point for *Tailbot* where the desired relative mass of the tail is 10% and the desired absolute stroke of the tail is equivalent to the body stroke of 45°. As relative tail mass and length increase, the tail stroke decreases.

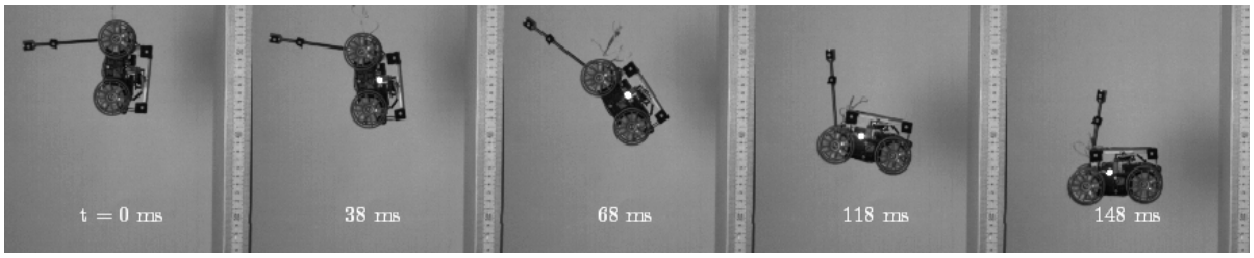



Figure 2.8: Representative plot of a zero net angular momentum maneuver where the body was able to orient itself 90° from its initial condition in 0.138 $(s)$.

dynamic terrestrial locomotion.

## 2.4 Experimental Results

### 2.4.1 Experimental Methods

We evaluated the effectiveness of our active tail by observing the robot's behavior in three simple representations of locomotory challenges: a fall, a large perturbation, and a transition between surfaces of greatly differing slopes. For each of the following trials, the robot program was initiated wirelessly from a host computer. Kinematics were captured by both the onboard sensors and by a high speed camera at 500 $(fps)$. Retro-reflective dots surrounded with black felt provided high-contrast markers for use with auto-digitizing software (*ProAnalyst, Xcitex Inc.*); angles and segment angular momentum were calculated with custom software (*MATLAB, Mathworks Inc.*).

For all experiments, the trials were repeated five times; measurements are reported as (mean $\pm$ one standard deviation). For the following simulation verification and falling experiments we ensured that no initial angular momentum was imparted to the system by releasing the robot from rest by a string. The specified controller was triggered when free fall was detected by monitoring the magnitude of the total acceleration.

### 2.4.2 Simulation Verification

To verify the model's predictive ability, we performed a free-fall test at maximum control effort and measured the time and tail stroke required to reorient the body 45°. The observed absolute tail stroke was ($49 \pm 2°$), which was only 2° larger than the model-predicted value. The difference in stroke may be due to error in the method of determining MOI. The time to move the body to 45° was ($0.064 \pm 0.0026$ $(s)$). The increase in time from the predicted 0.05 $(s)$ can be attributed to the unmodeled friction in the system and error in the measurements of the maximum supply voltage, stall torque and no-load angular velocity of the DC motor.

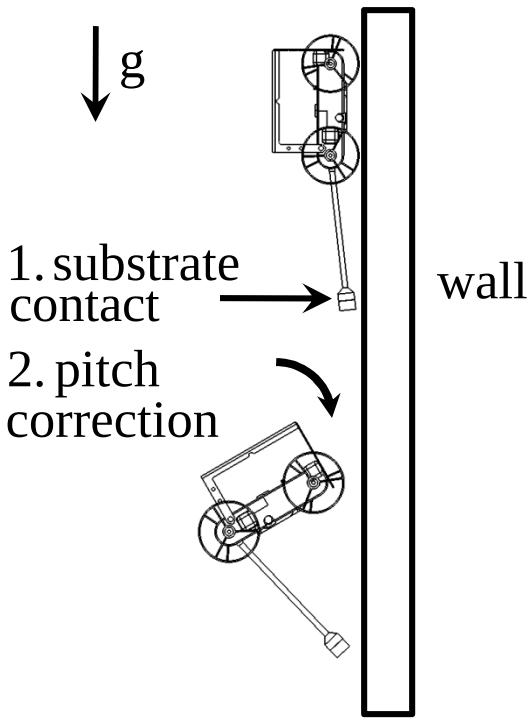



Figure 2.9: Diagram of controlled body angle correction after falling from a wall.

### 2.4.3 Falling

Surviving falls remains a great challenge for climbing robots. When substrate contact is lost and unrecoverable, the ability to reorient and land on energy-absorbing limbs could be critical to avoid damage to the robot [43]. Roll cages have been shown to protect sensitive parts during falls [44], but increase overall size and may decrease mobility.

#### Pure Fall

To test the ability of *Tailbot* to rapidly reorient itself in a fall with a zero net angular momentum turn, the feedback controller was implemented with $\theta_{ref} = 0^\circ$ and triggered at the detection of free-fall. We suspended the robot nose-down, allowing the tail to hang passively, and released the robot. *Tailbot* righted itself in about one body length of drop distance, and held the desired orientation until impact. The rise time for this maneuver was $(0.138 \pm 1.09\ (s))$. A representative trial is shown in Fig. 2.8.

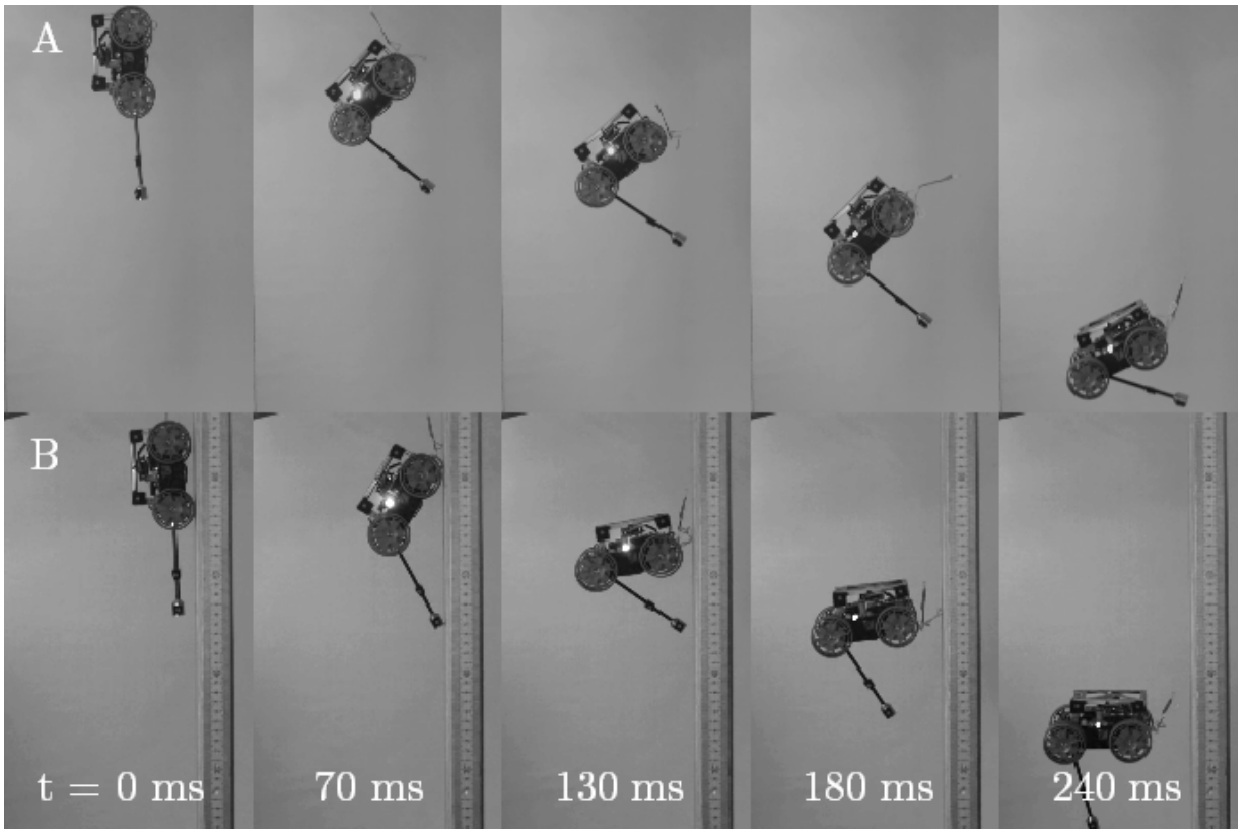



Figure 2.10: A. The robot attempts to regulate the body angle, but is unable to do so due to tail saturation. B. When the robot is able to contact the wall and inject angular momentum into the system, saturation is avoided. With enough falling distance the tail will rotate enough to avoid being landed upon.

### Falling from a Wall

While *Tailbot* can self-right from the initial orientation in Fig. 2.8, in a practical setting the tail would likely be held approximately parallel to the body, leaving less than 135° of available tail stroke for inertial maneuvering. In the previous section, the tail needed over 180° of relative stroke to rotate the body 90°, indicating that tail stroke saturation would prevent complete reorientation. Further, the wall itself may interfere with righting. To examine these issues, we dropped the robot in a parallel configuration both against a wall and in free space and compared righting performance. Again, the feedback controller initiated on free fall with $\theta_{ref} = 0°$. A representative sequence for each trial is shown in Fig. 2.10.

As predicted, the robot in open space was unable to fully regulate the body angle due to the physical constraint on tail stroke; the tail rotated a relative ($117.9 \pm 6.3°$) and hit the bump stop, which resulted in a maximum body angle stroke of only ($54.3 \pm 4.1°$). When the fall occurred

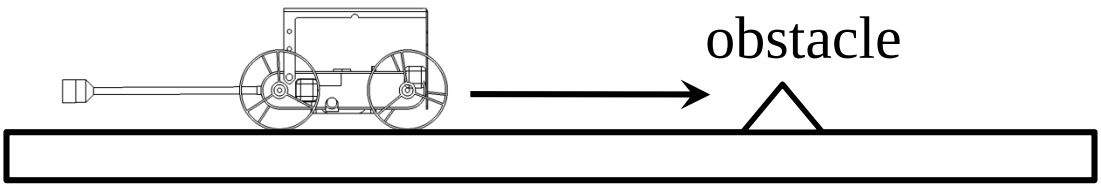



Figure 2.11: Diagram of single obstacle.

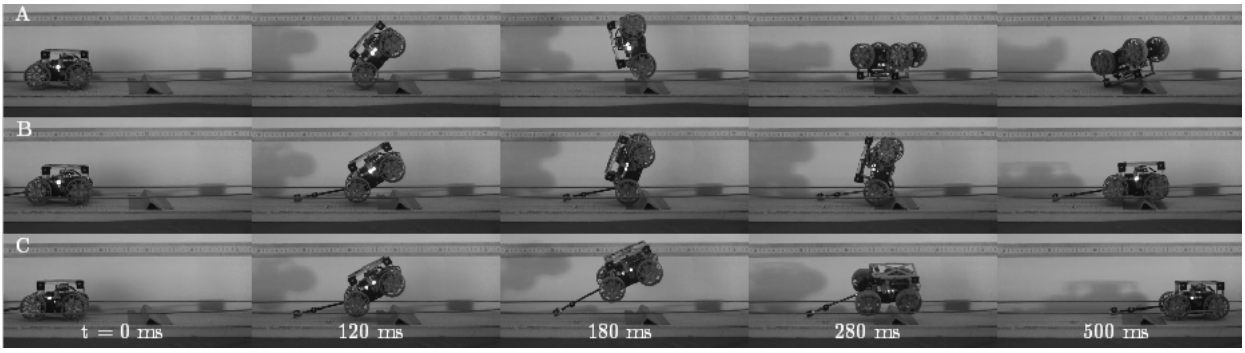



Figure 2.12: A. With no tail the robot catastrophically fails when traversing the obstacle. B. When the tail is passive the robot succeeds in traversing the obstacle, but the peak body angle of the robot is large ($> 60°$). C. When the body angle is controlled by the tail then the robot is able to successfully traverse the obstacle with much less peak body angle ($< 30°$).

from the wall, both the tail and the front wheel contacted the wall during the maneuver (Fig. 2.9). The contact force from the tail tip imparted net angular momentum to the robot, enabling complete reorientation without saturation. However, the angular momentum remained during the entire course of the fall, requiring the controller to continuously rotate the tail to regulate body angle. Given enough falling distance, this effect would further allow the tail to rotate away from the underbelly of the robot when landing, however a future tail design should incorporate a selective flexibility at the tail-base if landing on the tail became unavoidable. In this case, the ability to contact the substrate to generate angular momentum, and the ability to store angular momentum in an inertial structure were both required to complete the maneuver. When the mass was removed from the tail tip, the robot overshot horizontally and continued pitching forward. In less than 2.5 body lengths of drop distance, the robot was pointed straight down.

### 2.4.4 Perturbation During Terrestrial Locomotion

To test the ability of the active tail to help mitigate perturbations in a rapid locomotory task, we drove *Tailbot* along a track with a single obstacle placed in the path as shown in Fig. 2.11. The geometry of the obstacle and speed of the robot was chosen to elicit a perturbation that exceeded the stability boundary of *Tailbot* without a tail.

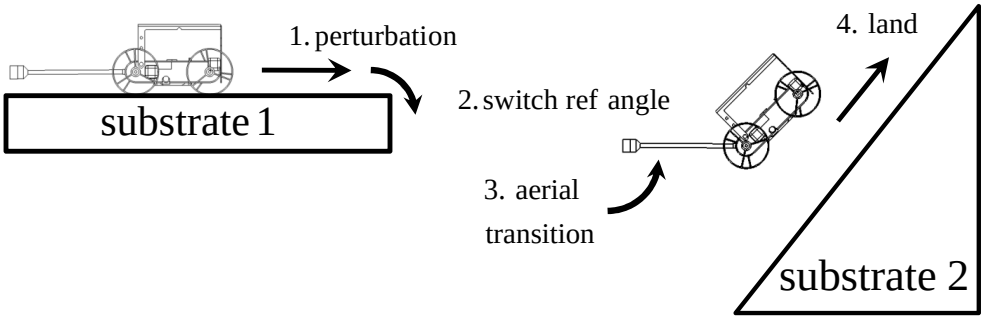



Figure 2.13: Diagram of transition task from flat substrate to inclined wall.

We drove the robot into the obstacle with the same forward speed for each of three different cases: *Tailbot* without tail, *Tailbot* with a passive tail (controller off), and *Tailbot* with an active tail using the PD feedback controller. The feedback controller was implemented with $\theta_{ref} = 0^\circ$ during the entire trial.

The maximum body angle due to the perturbation was recorded for each case. A representative sequence of these trials is shown in Fig. 2.12 on the next page. From row A, the tailless robot failed in every trial. From row B, the robot with a passive tail was able to overcome the obstacle, but the maximum measured body angle was large ($67 \pm 6.9^\circ$). With the feedback controller on as shown in row C, the maximum body angle was reduced by over half ($28 \pm 2.8^\circ$). During this trial the tail was able to compensate for a large perturbation even though the tail made contact with the substrate during the perturbation, since the contact and inertial torque occur in the same direction.

### 2.4.5 Completing Transitions

Figure 2.13 shows a task to evaluate the ability of *Tailbot* to transition between surfaces of significantly different slopes using an inertial maneuver. We drove the robot off a horizontal surface towards an inclined surface. On horizontal ground we set the reference angle $\theta_{ref} = 0^\circ$; when the robot left the surface we switched the reference angle to $\theta_{ref} = 45^\circ$, to achieve the proper landing angle. For simplicity, the switching was accomplished with feedforward timing, but in real applications, switching could be facilitated by visual servoing or other high-level control systems. Due to a moment resulting from gravity as the robot rolls off the horizontal platform, a disturbance was introduced into the system in the form of pitch-axis angular momentum.


Figure 2.14: Representative trial demonstrating a rapid transition task which occurs in 0.25 $(s)$.

In all trials, the robot initially pitched down due to the perturbation before swinging the tail and directing the body to match the wall angle and landing safely. By combining visual servoing with an inertial tail, a robot could use short aerial phases to smooth the transitions between uneven surfaces.

## 2.5 SUMMARY

We have shown that a robot with an appendage designed for both inertial and contact torques could enable rapid air-reorientation, improve fall survivability, and increase mobility and stability over difficult terrain. Our robot can reorient $90^\circ$ in one body length of vertical fall, survive perturbations that would flip an tailless robot, and effect rapid angle changes to smooth transitions between surfaces. Our bio-inspired tail-like design has several advantages over alternative technologies such as reaction wheels: low weight, high rotational inertia, and the capability to apply external torques via substrate contact. We found that contact torques and inertial torques can be modulated with the same simple controller to produce predictable, directed body rotation.

Addition of a tail to the next generation of dynamic legged robots could increase their ability to perform in environments too inaccessible or dangerous for humans, such as in search-and-rescue missions.

To highlight the initiation of active tailed robotic research, collaboration at this time has resulted in the 8.8 $(kg)$ legged robot XRhex to be successfully outfitted with a single DOF tail [37] as shown in Fig. 2.15. This robot is nearly two orders of magnitude larger and has similar performance.

In Chapter 4 we will increase the actuation DOF by one and show that orientation control in 3 dimensions can be achieved.

Figure 2.15: Side by side comparison of *Rhex* and *Tailbot* [37].

# Chapter 3

# Sensing 1DOF Orientation

## 3.1 Introduction

In this work featured in [42], our objective was to obtain a better attitude estimate of our 1 DOF robot presented in the previous chapter. Various signal processing algorithms have been studied to estimate this physical quantity from fusing multiple sensing methods. A kinematic Kalman filter (KKF) can fuse sensors that are kinematically related. For example, the acceleration measured by an accelerometer can be related to the position measured by an encoder [45, 46], so that a mathematical model between the two signals (i.e., a double integrator) can be established. In particular, the KKF has shown good performance when estimating the position and velocity of a robotic arm's end-effector [47]. The design of the KKF requires knowledge of the type of noise characteristics, and it is optimal only when the noise is zero mean, white, and Gaussian distributed. However, the noise in practice often exhibits more complicated characteristics, which deteriorates the optimality of KKF.

The conventional CF method utilizes linear-time-invariant (LTI) filters with different frequency characteristics as diagramed in Fig. 3.1, such that only the reliable components are selectively extracted in the frequency domain. Since integration is required to estimate the pitch or roll attitude from a rate gyroscope, where the integration amplifies uncertainty at low frequencies, the reliability

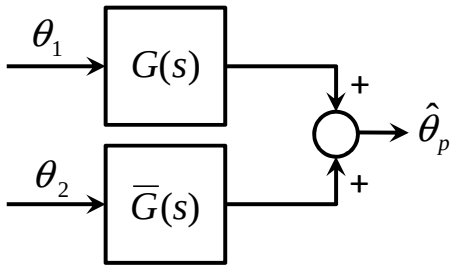



Figure 3.1: Block diagram of general two input complementary filtering.

of the estimate by gyroscope is poor at low frequencies (e.g., drift). The limitation in estimating attitude from accelerometers is evident due to the difficulty in decoupling the gravity vector from the higher frequency motion accelerations, and thus its reliability is poor at high frequencies. Therefore, the CF method can be applied to this problem, where a lowpass filter and a highpass filter are used as the filters, $G(s)$ and $\bar{G}(s)$, in Fig. 3.1. However, the challenge of the CF method is to design the LTI filters according to the frequency dependent reliabilities. When the frequency characteristics are time-invariant and clearly distinguished, the design of the filters require minor tuning. However, in many cases, the frequency characteristics are not fixed and the simple LTI filters may not produce the best result.

A time-varying complementary filtering (TVCF) method is proposed as shown in Fig. 3.2. The proposed method utilizes linear time-varying (LTV) filters for estimating a physical quantity from multiple sensors whose reliability in the frequency domain is time-varying. It is able to improve the performance of the CF method by adding more degrees of freedom in the design of the filters. The main issue is determining the best conditions with which to adapt the LTV filters. A fuzzy logic scheme is proposed to formalize the qualitative conditions for adaptation. The performance is verified against encoder measurements from a 3DOF apparatus that allows for instantaneous translations and rotations in the $\mathbf{E}_1 - \mathbf{E}_2$ plane as opposed to a fixed pivot.

This chapter is organized as follows. In section 3.2, the standard CF method will be reviewed. In section 3.3 the details of the TVCF will be explained along with a proof of stability. Then, in section 3.4, the experimental results will be discussed displaying the benefit of the method, and finally the conclusions will be drawn in section 3.5.

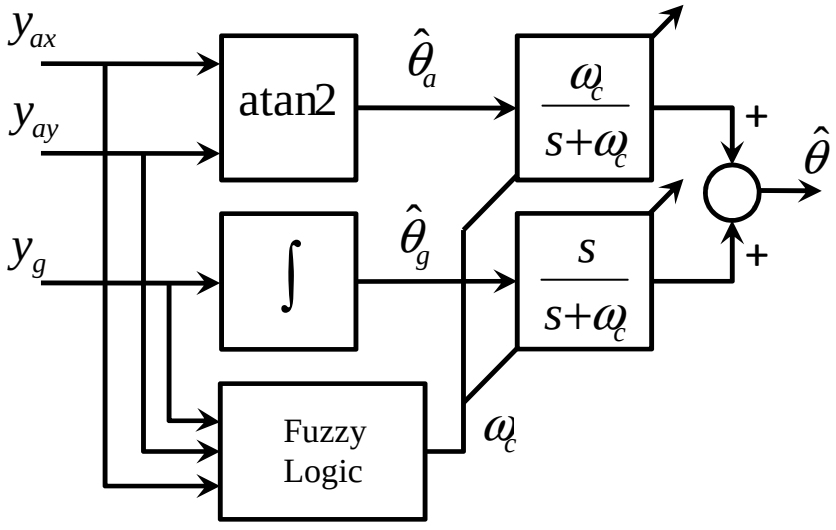



Figure 3.2: Block diagram of proposed TVCF method with gyroscope and accelerometer signals.

## 3.2 Previous Works: CF Method

The main idea of the CF method, represented in Fig. 3.1, is to fuse two measurements, $\theta_1(t)$ and $\theta_2(t)$, of the same physical quantity, $\theta_p(t)$. $\bar{G}(s)$ is often designed as $1-G(s)$, such that $G(s)+\bar{G}(s)=1$. In this case, the estimate $\hat{\theta}_p(t)$ is exactly the physical quantity if the measurements are without noise and error.

Suppose that $\theta_1(t)$ and $\theta_2(t)$ are reliable at low and high frequencies respectively. In a first order implementation, $G(s)$ can be set as a low pass filter with a fixed cut off frequency, $\omega_c$, i.e.

$$G(s)=\frac{\omega_c}{s+\omega_c}. \tag{3.1}$$

The complement of $G(s)$, i.e., $\bar{G}(s)=1-G(s)$, is then

$$\bar{G}(s)=\frac{s}{s+\omega_c}, \tag{3.2}$$

which becomes a high pass filter. Note that $G(s)+\bar{G}(s)=1$ for all $s$ and consequently there is no phase delay in the estimate, $\hat{\theta}_p(t)$.

If $\theta_1(t)=\theta_p(t)+\eta_1(t)$ and $\theta_2(t)=\theta_p(t)+\eta_2(t)$, where $\eta_1$ is high frequency noise and $\eta_2$ is low frequency noise both relative to $\omega_c$, then by (3.1, 3.2) the estimate by CF is

$$\begin{aligned}\hat{\theta}_p &= G(s)[\theta_p+\eta_1]+\bar{G}(s)[\theta_p+\eta_2] &(3.3)\\ &= \theta_p+G(s)[\eta_1]+\bar{G}(s)[\eta_2]. &(3.4)\end{aligned}$$

where the signals are in the Laplace domain. If the cut-off frequency, $\omega_c$, is properly selected, the magnitude of $G(s)[\eta_1]+\bar{G}(s)[\eta_2]$ in the time domain is small compared to $\theta_p$ due to the characteristics of the noise and LTI filters.

With the goal of estimating the pitch or roll attitude, $\theta(t)$, which will now be referred to as just attitude, of a rigid body as shown in Fig. 3.3, three measurements from a single axis gyroscope and a dual axis accelerometer are used and represented as

$$\begin{aligned}y_g(t) &= \boldsymbol{\omega}(t)\cdot\mathbf{b}_3+\beta_g(t)+\eta_g(t) &(3.5)\\ y_{ax}(t) &= \mathbf{a}_m(t)\cdot\mathbf{b}_1+\mathbf{g}\cdot\mathbf{b}_1+\beta_{ax}(t)+\eta_{ax}(t) &(3.6)\\ y_{ay}(t) &= \mathbf{a}_m(t)\cdot\mathbf{b}_2+\mathbf{g}\cdot\mathbf{b}_2+\beta_{ay}(t)+\eta_{ay}(t) &(3.7)\end{aligned}$$

where $\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{b}_3$, are body frame coordinates perpendicular to each other. The gyroscope measurement, $y_g(t)$, aligned along $\mathbf{b}_3$, measures the angular rate, $\boldsymbol{\omega}(t)=\dot{\theta}(t)\mathbf{b}_3$, but also contains bias, $\beta_g(t)$, and noise, $\eta_g(t)$. The two orthogonal accelerometer measurements, $y_{ax}(t)$ and $y_{ay}(t)$, are aligned along the axes, $\mathbf{b}_1$ and $\mathbf{b}_2$, respectively. They measure the body fixed basis components of the gravity vector, $\mathbf{g}\cdot\mathbf{b}_1$ and $\mathbf{g}\cdot\mathbf{b}_2$, where $\mathbf{g}=-g\mathbf{E}_2$, and the motion accelerations, $\mathbf{a}_m(t)\cdot\mathbf{b}_1$ and $\mathbf{a}_m(t)\cdot\mathbf{b}_2$. The motion accelerations are assumed to be much more significant than the bias or noise in the accelerometer sensors.

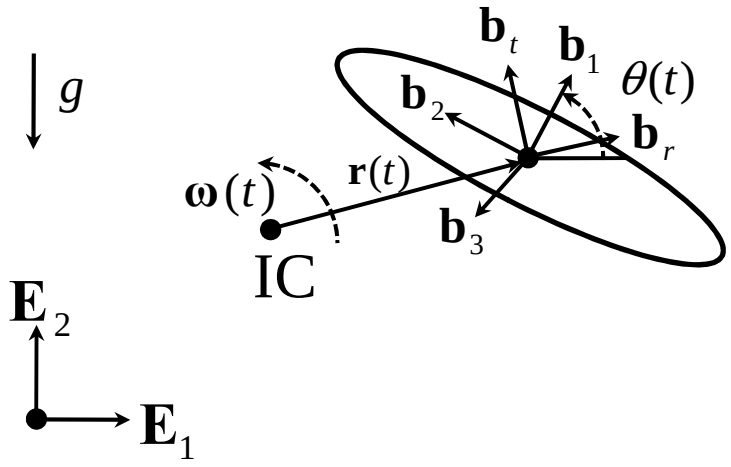



Figure 3.3: General diagram of a planar rigid body.

With a known initial condition, $\theta_o$, the single axis gyroscope can be integrated to achieve an attitude estimate, $\hat{\theta}_g(t) = \int_0^t y_g(\tau)d\tau + \theta_o$. On the other hand, in the near static or constant velocity condition, the motion accelerations, $\mathbf{a}_m(t)$ in (3.6, 3.7), are near zero and the orthogonal components of the dual axis accelerometer can estimate the attitude from the gravity vector by $\hat{\theta}_a(t) = \arctan 2(y_{ax}(t), y_{ay}(t))$. The $\arctan 2()$ function is chosen as opposed to the $\arctan()$ function to estimate $\theta(t)$, since the range is $[-\pi, \pi)$ $(rad)$ and not just $(-\frac{\pi}{2}, \frac{\pi}{2})$ $(rad)$.

Individually, the gyroscope gives a decent estimate of angle at higher frequencies, but this estimate tends to drift due to the bias and noise. The accelerometer gives an accurate static estimate, but this estimate is predominately corrupted by motion accelerations. Therefore, the attitude estimates from the gyroscope and accelerometer can be modeled as

$$\hat{\theta}_g(t) = \theta(t) + \eta_d(t) \tag{3.8}$$
$$\hat{\theta}_a(t) = \theta(t) + \eta_m(t) \tag{3.9}$$

where $\theta(t)$ is the true attitude angle to be estimated by CF, and $\eta_d(t)$ and $\eta_m(t)$ are the measurement errors corresponding to the drift and motion acceleration terms respectively. In order to apply the CF method, $\eta_d(t)$ and $\eta_m(t)$ are treated as noise; $\eta_d(t)$ is low frequency and $\eta_m(t)$ is high frequency. Using this method a more accurate estimate of $\theta(t)$ compared to the individual sensors can be produced. Choosing a larger $\omega_c$ places more trust on the accelerometer estimate, $\hat{\theta}_a(t)$, while choosing a smaller $\omega_c$ places more trust on the gyroscope estimate, $\hat{\theta}_g(t)$.

## 3.3 TIME VARYING COMPLEMENTARY FILTERING

During general planar motion, the rigid body is undergoing translation and rotation about an instantaneous center, which is designated by (IC) as shown in Fig. 3.3. The motion acceleration in

Table 3.1: Rule Basis of Fuzzy Logic.

| Condition | $\lvert\lVert\mathbf{y}_a(t)\rVert_2 - g\rvert$ | $\lVert\dot{\mathbf{y}}_a(t)\rVert_2$ | $\lvert y_g(t)\rvert$ | $\lvert\dot{y}_g(t)\rvert$ | $\mu(t)$ |
|---|---|---|---|---|---|
| stationary or no rotation & const. lin. vel. | *low* | *low* | *low* | *low* | 1 |
| linear acceleration | *high* | $N/A$ | $N/A$ | $N/A$ | 0 |
| nonstationary, dashed circle ($\lVert y_a\rVert_2 = g$) | $N/A$ | *high* | $N/A$ | $N/A$ | 0 |
| centripetal acceleration | $N/A$ | $N/A$ | *high* | $N/A$ | 0 |
| tangential acceleration | $N/A$ | $N/A$ | $N/A$ | *high* | 0 |

(3.6, 3.7) constrained to the $\mathbf{E}_1 - \mathbf{E}_2$ plane can be written as

$$\mathbf{a}_m = \mathbf{a}_t + \boldsymbol{\omega} \times [\boldsymbol{\omega} \times \mathbf{r}] + \boldsymbol{\alpha} \times \mathbf{r} \tag{3.10}$$

where $\mathbf{a}_t$ is the translational acceleration and $\mathbf{r} = r\mathbf{b}_r$ is the radial vector. Although, $\boldsymbol{\omega}$ and $\boldsymbol{\alpha}$ in (3.10) are arbitrary vectors in three dimensional space, it is assumed that $\boldsymbol{\omega} = \dot{\theta}\mathbf{b}_3$ and $\boldsymbol{\alpha} = \ddot{\theta}\mathbf{b}_3$, for the sake of simplicity. The last two terms constitute the centripetal and tangential acceleration. Ideally, if the motion accelerations, $\mathbf{a}_m$, could be measured or estimated, then they could be decoupled from the gravity vector, $\mathbf{g}$, in (3.6) and (3.7). This would allow for the accelerometer estimate to be trusted for a higher frequency range. However, the translational acceleration, $\mathbf{a}_t$, and the instantaneous center, which determines the the magnitude and direction of the radial vector, $\mathbf{r}$, cannot be directly measured. Therefore, if it can be detected *when* the motion accelerations occur, then the cut off frequency, $\omega_c$ in (3.1) and (3.2), can be adjusted to a low value, $\omega_{low}$, and effectively filter out the high frequency noise, $\eta_m(t)$, in (3.9). Conversely, when the rigid body is not accelerating, the cutoff frequency can be increased to a high value, $\omega_{high}$, to update the estimate from the accelerometer. Based on these considerations, it is proposed to use the Time Varying Complementary Filter (TVCF) approach.

In order to formalize our engineering intuition we chose a fuzzy logic method to discern the non-accelerating state. A simple rule-base to determine the cut-off frequency is described in Table 3.1. The last column contains the output of the fuzzy logic, $\mu$, which is valued in the range of [0,1] and signifies how to transition between $\omega_{high}$ or $\omega_{low}$ as

$$\omega_c(t) = \mu(t)\omega_{high} + (1 - \mu(t))\omega_{low}. \tag{3.11}$$

Notice that if $\mu = 1$ then $\omega_c = \omega_{high}$ thus signifying when the estimate from the accelerometer is more trustworthy. Inputs to the fuzzy logic, which determine the value of $\mu$, can be classified qualitatively as either *low* or *high*. A membership function will later be explained to convert this qualitative descriptions to quantitative logic values. With abuse of notations, the TVCFs are represented by (3.1) and (3.2) with $\omega_c$ replaced by $\omega_c(t)$ as in (3.11).

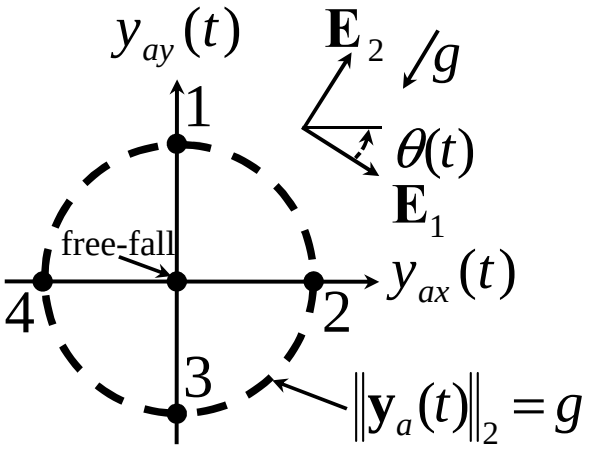



Figure 3.4: Circle of ambiguity from the accelerometer sensors due to coupling of motion accelerations and gravitational field.

From (3.10) the centripetal and tangential accelerations occur when the angular rate, $\dot{\theta}(t)$, and the angular acceleration, $\ddot{\theta}(t)$, are *high*. This information can be determined from the magnitude of the gyroscope signal, $|y_g(t)|$, and the signals time derivative, $|\dot{y}_g(t)|$, which is represented in the last two rows of Table 3.1.

The rigid body is assumed to exist in a gravitational field of $-1g$ in the $\mathbf{E}_2$ world frame direction. The accelerometer measurement vector is first defined as $\mathbf{y}_a(t) = [y_{ax}(t)\ y_{ay}(t)]^T$. Any translational acceleration will occur when the magnitude of the accelerometer measurement vector, $\|\mathbf{y}_a(t)\|_2 = \sqrt{y_{ax}(t)^2 + y_{ay}(t)^2}$, is not close to the gravitational acceleration, $g$. One example is free-fall; $\|\mathbf{y}_a(t)\|_2 = 0$. Therefore, the estimate by accelerometer is not trustable if $|\|\mathbf{y}_a(t)\|_2 - g|$ is *high*. This logic is represented by row two in Table 3.1.

### 3.3.1 Fuzzy Logic

The equation, $\|\mathbf{y}_a(t)\|_2 = g$, represented by the dashed circle of ambiguity in Fig. 3.4, does hold the exact point of non-acceleration, but also contains points of acceleration and shows the limitation of only checking $|\|\mathbf{y}_a(t)\|_2 - g|$. This fact is apparent if one briefly assumes that the sensor signals, $y_{ax}(t)$ and $y_{ay}(t)$, are aligned with the $\mathbf{E}_1$ and $\mathbf{E}_2$ axes respectively, then the non-acceleration point of interest is point 1, which cannot be distinguished from, falling down at $2g$ (point 3), or accelerating perfectly horizontal and down at $1g$ (point 2 and 4). Since the dashed circle is not stable in any sense, other than possibly the non-acceleration state, it would be improbable to remain along the circle for a prolonged period of time. Therefore, with this assumption, acceleration can be discerned from non-acceleration even when $|\|\mathbf{y}_a(t)\|_2 - g|$ is *low* by checking if the magnitude of the *jerk*, $\|\dot{\mathbf{y}}_a(t)\|_2 = \sqrt{\dot{y}_{ax}(t)^2 + \dot{y}_{ay}(t)^2}$, is *high*. This logic is represented in the third row of Table 3.1.

With the logic built, the inputs to the fuzzy membership function, $x_i$, diagrammed in Fig. 3.5

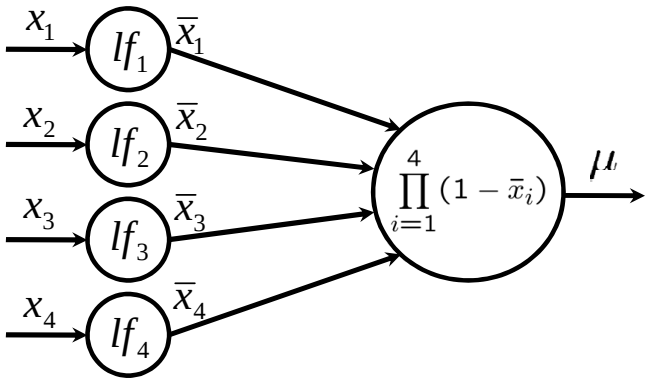



Figure 3.5: Network diagram of fuzzy logic where $lf_i$ represents the fuzzy membership function in Fig. 3.6.

can be selected from the column headers in Table 3.1 as

$$x_1(t) = |\|\mathbf{y}_a(t)\|_2 - g| \tag{3.12}$$
$$x_2(t) = \|\dot{\mathbf{y}}_a(t)\|_2 \tag{3.13}$$
$$x_3(t) = |y_g(t)| \tag{3.14}$$
$$x_4(t) = |\dot{y}_g(t)|. \tag{3.15}$$

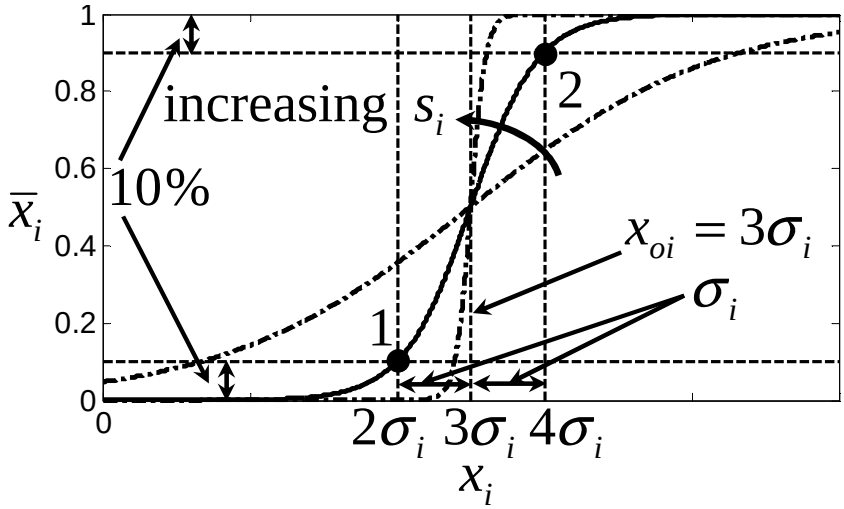



Figure 3.6: The membership function, $lf_i$, of (3.16) displaying the result of increasing the slope $s_i$. The intersections of the 10 % lines and $\sigma_i$ lines at points 1 and 2 designate the design points for calculation of $s_i$.

These inputs are then scaled by a membership function, $lf_i$, to a quantitative unitless output, $\bar{x}_i$, with range of (0,1) representing the *low* and *high* descriptions in the logic table. Although any membership function can be used, a hyperbolic tangent function

$$\bar{x}_i(t) = 0.5(1 + \tanh(s_i(x_i(t) - x_{oi}))) \tag{3.16}$$

was chosen for its smooth transition properties shown by Fig. 3.6. Here, $x_{oi}$ represents the 50 % transition point or when the value of the input, $x_i$, corresponds to an output, $\bar{x}_i = 0.50$. The value $s_i$ represents the slope of transition, for example larger values of $s_i$ corresponds to a sharper transition.

From inspection of the logic table it is clear that only when *all* the inputs are low it is valid to assume a non-accelerating state and to choose $\mu = 1$. Therefore, the rule basis of the fuzzy logic in Table 3.1 can be formulated by a multiplication function, i.e.,

$$\mu(t) = \prod_{i=1}^{4}(1 - \bar{x}_i(t)). \tag{3.17}$$

Notice that (3.17) returns one only when all $\bar{x}_i = 0$, which follows the rules in Table 3.1.

### 3.3.2 Stability Analysis

In the previous section, a continuous time representation of the TVCF was presented. Since the TVCF must be implemented in the discrete time domain, the stability of TVCF is analyzed in the discrete time domain with the noise signals ignored in the analysis. Although many continuous to discrete time transformations are available [48], *Tustin's* bilinear transformation, was chosen so that the LTV filter relationship, $G(z) + \bar{G}(z) = 1$, could be maintained where $z^{-1}$ is a step delay operator. If the same input signal, $u$, is assumed to enter both LTV filters and $y_{lp}$ and $y_{hp}$, are the outputs of $G(z)[u]$ and $\bar{G}(z)[u]$ respectively then the resulting difference equations are

$$\begin{aligned} y_{lp}(k) &= -\frac{-2+\omega_c(k)dt}{2+\omega_c(k)dt}y_{lp}(k-1) \\ &\quad +\frac{\omega_c(k)dt}{2+\omega_c(k)dt}u(k) + \frac{\omega_c(k)dt}{2+\omega_c(k)dt}u(k-1) \end{aligned} \tag{3.18}$$

$$\begin{aligned} y_{hp}(k) &= -\frac{-2+\omega_c(k)dt}{2+\omega_c(k)dt}y_{hp}(k-1) \\ &\quad +\frac{2}{2+\omega_c(k)dt}u(k) - \frac{2}{2+\omega_c(k)dt}u(k-1) \end{aligned} \tag{3.19}$$

where $dt$ is the sampling time. The output, $y(k)$, of the TVCF is calculated by adding (3.18) and (3.19), which results in

$$\begin{aligned} y(k) &= y_{lp}(k) + y_{hp}(k) \\ &= -\rho(k)y(k-1) + \rho(k)u(k-1) + u(k) \end{aligned} \tag{3.20}$$

where

$$\rho(k) = \frac{-2 + \omega_c(k)dt}{2 + \omega_c(k)dt}. \tag{3.21}$$

In the case where $\rho(k)$ is constant, it can be easily proven that the discrete time transfer function relating $u(k)$ to $y(k)$ obtained from (3.20) becomes one and that $y(k) = u(k)$ for $k \neq 0$. To prove the error convergence in the time-varying case the error is first defined as $e(k) = y(k) - u(k)$. By rearranging (3.20), the error dynamics is obtained as

$$e(k) = -\rho(k)e(k-1). \tag{3.22}$$

For a given initial condition, $e(0)$, the solution of (3.22) is

$$e(k) = [\prod_{i=1}^{k} -\rho(i)]e(0). \tag{3.23}$$

If $|\rho(k)| < 1$ for all $k$ then the output of the TVCF, $y(k)$, converges to $u(k)$ monotonically even when $e(0) \neq 0$. Note in (3.21) that $|\rho(k)| < 1$ as long as $\omega_c(k)dt > 0$. Therefore, the stability of the TVCF is guaranteed if $\omega_c(k) > 0$ for all $k$ since the sampling time is always $dt > 0$.

## 3.4 EXPERIMENTAL RESULTS

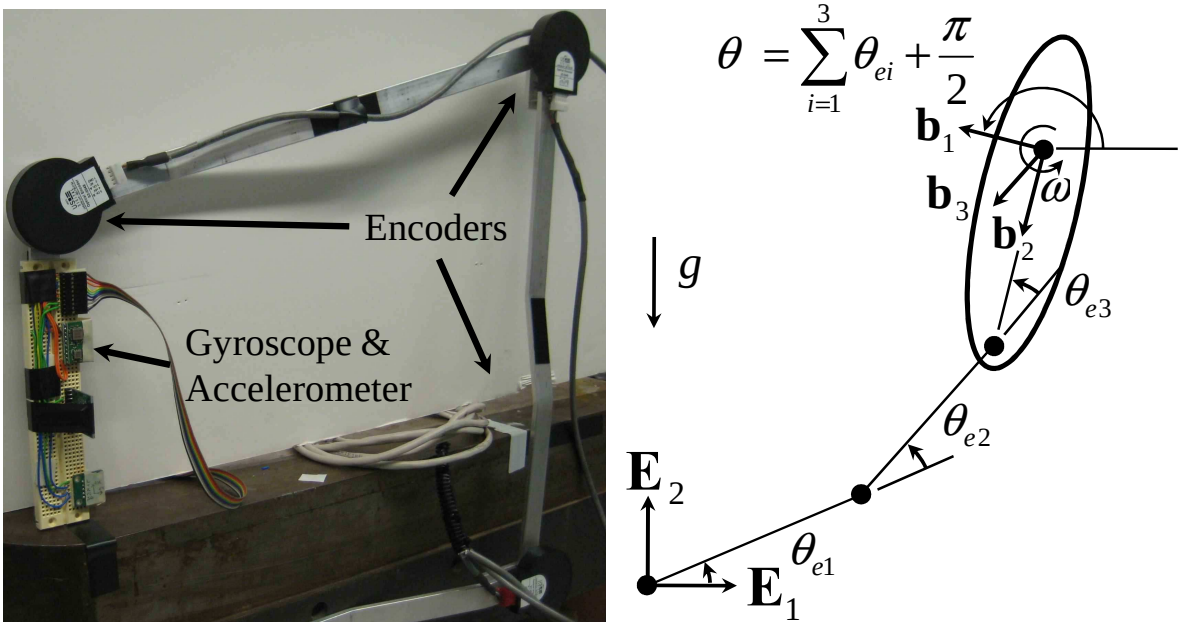
### 3.4.1 Experimental Apparatus

For verification of the proposed TVCF method, the true attitude angle should be measured by an accurate measurement system. In this paper, a linkage system equipped with high resolution encoders was set up for this purpose as shown in Fig. 3.7a. The linkage system was designed with 3DOF to allow for rotation and translation in the $\mathbf{E}_1 - \mathbf{E}_2$ plane as shown in Fig. 3.7b. The true attitude, $\theta(t)$, was calculated from the sum of three high resolution joint encoder measurements, $\theta_{ei}(t)$, as, $\theta(t) = \sum_{i=1}^{3} \theta_{ei}(t) + \frac{\pi}{2}$.

The algorithms and data collection were implemented on National Instrument's 7831R FPGA and LabVIEW with a fixed sampling time of 0.005 $(s)$. An Invensense, IDG-650, $\pm 2000$ $(\frac{deg}{s})$, gyroscope and an Analog Devices, ADXL203, $\pm 1.7g$, dual axis accelerometer were mounted on the third link at the origin of the body fixed basis in Fig. 3.7b. For verification a US Digital, E6, 2048 count per revolution, optical encoder was mounted at each joint. In each experimental trial, the link was randomly hand shaken and rotated while also providing short periods of rest.

### 3.4.2 Selection of Parameters

The TVCF method requires the selection of ten parameters; $\omega_{low}$, $\omega_{high}$, and all of the $x_{oi}$ and $s_i$ values. The following can serve as a guideline for choosing them. The parameters $x_{oi}$ and $s_i$ determine the non-accelerating conditions. If the input signal into each membership function, $\bar{x}_i$,




(a) Photograph of experimental apparatus.

(b) Schematic of experimental apparatus.

Figure 3.7: Experimental apparatus for 1DOF attitude verification /FIXME symbol problem

are assumed to be zero mean and normally distributed when the system is static then the parameters can be conveniently calculated from one specification; the standard deviation, $\sigma_i$, of the noise of, $x_i$. The 50 % transition point, $x_{oi}$, can be set equal to $3\sigma_i$ to encapsulate 99.7 % of the $x_i$ in a statistical sense when those signals are static. The slope, $s_i$, can be calculated by fixing points 1 and 2 in Fig. 3.6 where $x_i$ is one $\sigma_i$ away from $x_{oi}$ and $\bar{x}_i(t)$ is equal to 10 % of the the total distance from either bound of $\bar{x}_i(t)$. With these design points, the slope can be calculated from (3.16) as

$$s_i = \frac{\tanh^{-1}(-0.8)}{-\sigma_i}. \tag{3.24}$$

Sensor bias significantly affects the performance of detecting non-acceleration in the signals, $\bar{x}_1$ and $\bar{x}_3$, but does not affect $\bar{x}_2$ and $\bar{x}_4$ since they are differentiated signals. Therefore, multiplying $\sigma_1$ and $\sigma_3$ by a factor, $\gamma_i > 1$, $i = 1,3$, produces larger fictitious standard deviations that can account for the upper limit of the sensor bias. By inspection of the signals during static conditions, a value of $\gamma_i = 5$ allowed the capture of the upper limit of the sensor bias. Both the sensor bias and standard deviation of the signal noise can be determined experimentally or calculated from the sensor specifications.

The parameter, $\omega_{low}$, should be chosen low enough to ignore the accelerometer during motion. A preselected value of $\omega_{low} = 0.01$ $(\frac{rad}{s})$ was found to be appropriate. When the algorithm decides to trust the accelerometer estimate, the cutoff frequency adjusts towards $\omega_{high}$. Also at this time,

the attitude estimate may be affected by drift, but converges faster to the accelerometer estimate with a larger value of $\omega_{high}$. The upper limit of $\omega_{high}$ should conservatively be no more than $1/25$ of the sampling time; $\omega_{high} < 50\ (\frac{rad}{s})$ for a sampling rate of $0.005\ (s)$ [49]. A preselected value of $\omega_{high} = 10\ (\frac{rad}{s})$ was found appropriate. In the Section 3.4.4, the adequacy of these two parameters will be justified. The preselected set of parameters are shown in the second column of Table 3.2.

### 3.4.3 Experimental Results

Experiments were performed to obtain the attitude estimates from the accelerometer, gyroscope, CF method and TVCF method. For each estimation method, ten, one minute long, trials were performed. The experimental error was also recorded and defined by the difference between the true angle and the estimate of the various methods as $e_{exp}(k) = \theta(k) - \hat{\theta}(k)$. The performance of each trial was evaluated by the root-mean-square (rms) of the experimental error, $\xi$, defined by

$$\xi = \sqrt{\frac{\sum_{k=1}^{n} e_{exp}^2(k)}{n}} \tag{3.25}$$

where $n$ is the total number of samples. The ten $\xi$ values for each method are plotted in an errorbar plot as shown in Fig. 3.8. The distribution of $\xi$'s for the attitude estimate from the accelerometers has a mean of approximately $45^\circ$ and a low variance, while that for the gyroscope estimate has a high variance and a mean of approximately $60^\circ$. The attitude estimate from the CF method has a mean of approximately $10^\circ$ while the mean for the proposed TVCF method is approximately $5^\circ$, which is a significant improvement (*t-test*, $p = 0.0124 < 0.05$) [50]. A probability value, $p < 0.05$, indicates that there is a significant difference between the mean of the two sets of $\xi$.

Fig. 3.9 and Fig. 3.10 are representative experimental attitude and error plots of one of the ten experiments for the CF and TVCF methods. They are taken over the same trials and time periods respectively. Both, the conventional CF method and TVCF match the true angle measured by the encoder fairly well. Since the gyroscope estimate drifts, both estimates will suffer from drift during periods of movement. However, during periods of rest denoted by the grey areas in Fig. 3.11 and Fig. 3.12, the TVCF is able to adapt the complementary filters cutoff frequency to high and the estimate can quickly converge to the true attitude faster than the CF method.

### 3.4.4 Adequacy of the Selected Parameters

The adequacy of selecting the values for $\omega_{high}$ and $\omega_{low}$ are analyzed here. Ten, one minute trials recording the data from the encoder, gyroscope, and accelerometer were collected and the TVCF algorithm was implemented off-line. The values of $\omega_{high}$ and $\omega_{low}$ were varied while the precalculated membership parameters, $x_{oi}$ and $s_i$, were fixed. The median rms errors between the encoder data and the off-line TVCF estimate as defined by (3.25) were plotted in Fig. 3.13. Notice the median rms error for the preselected values, $\omega_{high} = 10\ (\frac{rad}{s})$ and $\omega_{low} = 0.1\ (\frac{rad}{s})$ mentioned in

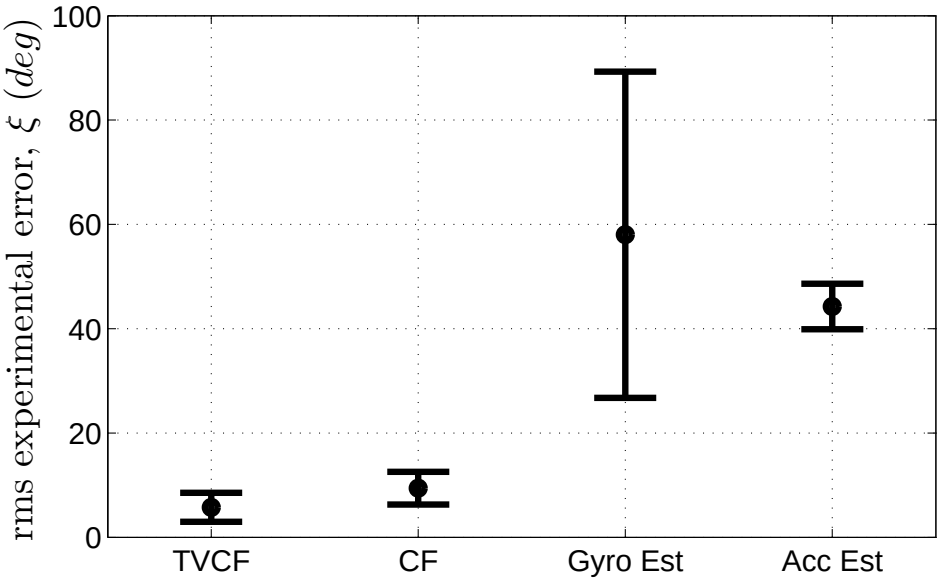



Figure 3.8: Errorbar plot comparing the rms experimental error, $\xi$, for each of the different methods. The center circle in each category represents the mean value while the error bars denote one standard deviation.

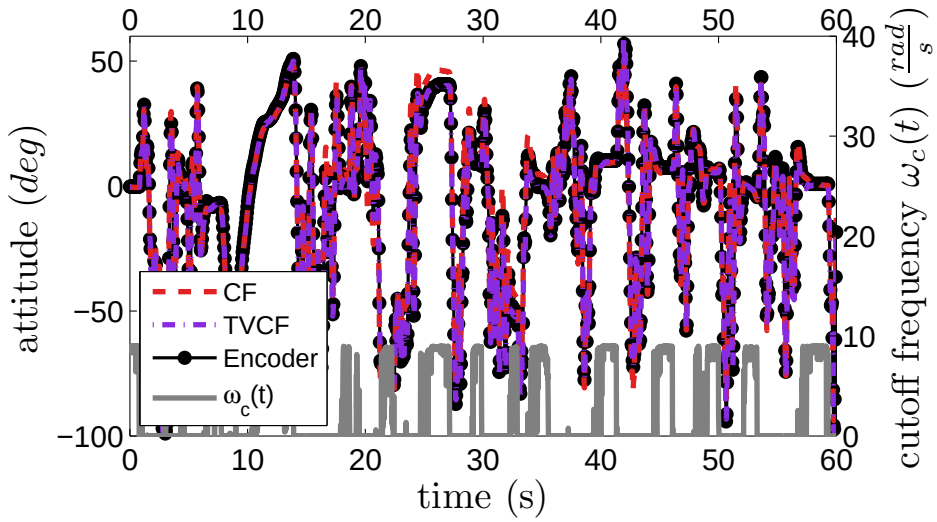



Figure 3.9: Representative plot of the estimated attitude for each of the different methods as well as the corresponding $\omega_c(t)$ for the TVCF.

section 3.4.2, fall near the minimum of the plot. Also, the median rms errors surrounding the preselected values deviate little and give confidence in their selection if the TVCF method were to be implemented on other systems.

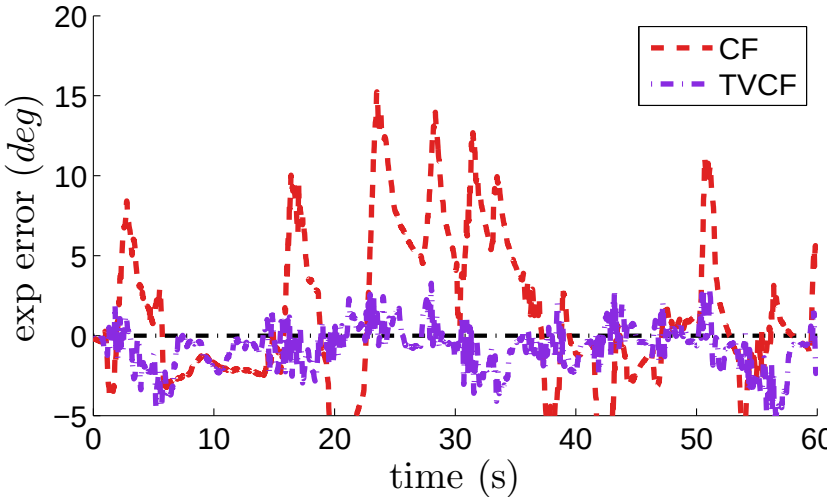



Figure 3.10: Representative plot of the experimental error, $e_{exp}$, for each of the different methods.

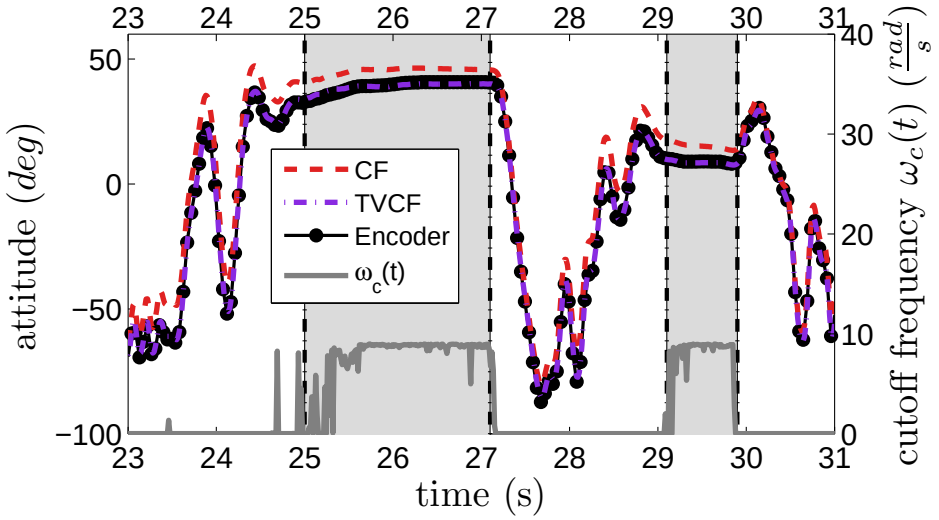



Figure 3.11: Zoomed in plot of Fig. 3.9. Grey areas represent periods of rest.

Additionally, to test the adequacy of all the parameters, $\omega_{high}$, $\omega_{low}$, $x_{oi}$, and $s_i$, an off-line optimization procedure obtained the optimized values using the preselected values as the initial parameters. Again, the TVCF algorithm was implemented off-line with the same data. The *Matlab* function *fmincon* was employed for the off-line optimization procedure and minimized the same rms of the error. As expected, the optimization produced ten varying sets of values, which were then averaged into one final set of optimized parameters as shown in the right column of Table 3.2. The real time performance using the precalculated parameters were then tested against the performance using the optimized values. As expected, the TVCF with optimized parameters showed

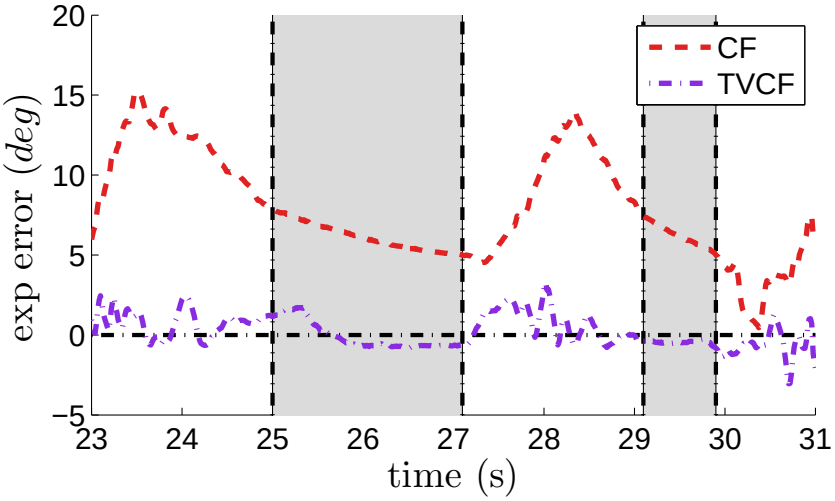



Figure 3.12: Zoomed in plot of Fig. 3.10. Grey areas represent periods of rest.

Table 3.2: Parameters.

| Parameter | Initial Value | Optimized Value |
|---|---|---|
| $\omega_{high}$ | 10 $(\frac{rad}{s})$ | 9.1 $(\frac{rad}{s})$ |
| $\omega_{low}$ | 0.01 $(\frac{rad}{s})$ | 0.024 $(\frac{rad}{s})$ |
| $\sigma_1$ | 0.0017 $(g)$ | 2.56 $(g)$ |
| $\sigma_2$ | 0.004 $(\frac{dg}{dt})$ | 1.4 $(\frac{dg}{dt})$ |
| $\sigma_3$ | 1.2 $(\frac{deg}{s})$ | 5.8 $(\frac{deg}{s})$ |
| $\sigma_4$ | 320 $(\frac{deg}{s^2})$ | 320 $(\frac{deg}{s^2})$ |

a slight decrease in the median experimental error as shown in Fig. 3.14, but the improvement was negligible (*t-test*, $p = 0.4252 > 0.05$) also indicating a low sensitivity. This result supports the adequacy of the aforementioned guidelines since any verification by encoder data would most likely not be available in practical settings.

## 3.5 SUMMARY

A new approach for a single DOF attitude estimation was proposed using a TVCF method. Significant improvements were achieved over the conventional CF method while still maintaining ease of implementation. The effectiveness has been verified with experimental results. This work may

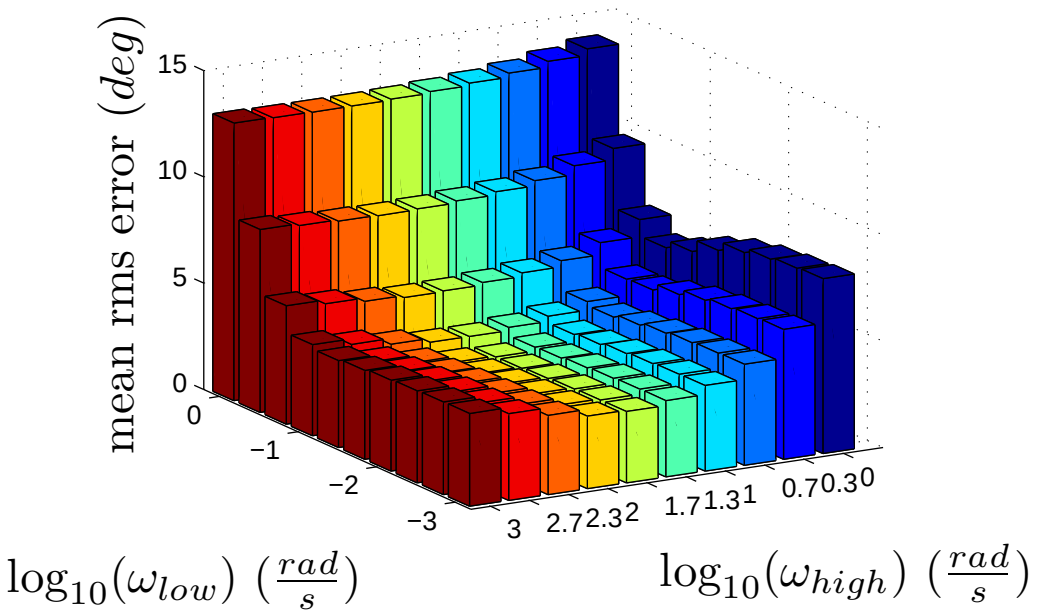



Figure 3.13: Three dimensional plot of the TVCF method performance for varying values of $\omega_{high}$ and $\omega_{low}$. Each column represents the mean rms error over ten, one minute trials. For the preselected parameters in the previous section, the performance sensitivity is quite low due to the flat geometry at the bottom of the basin.

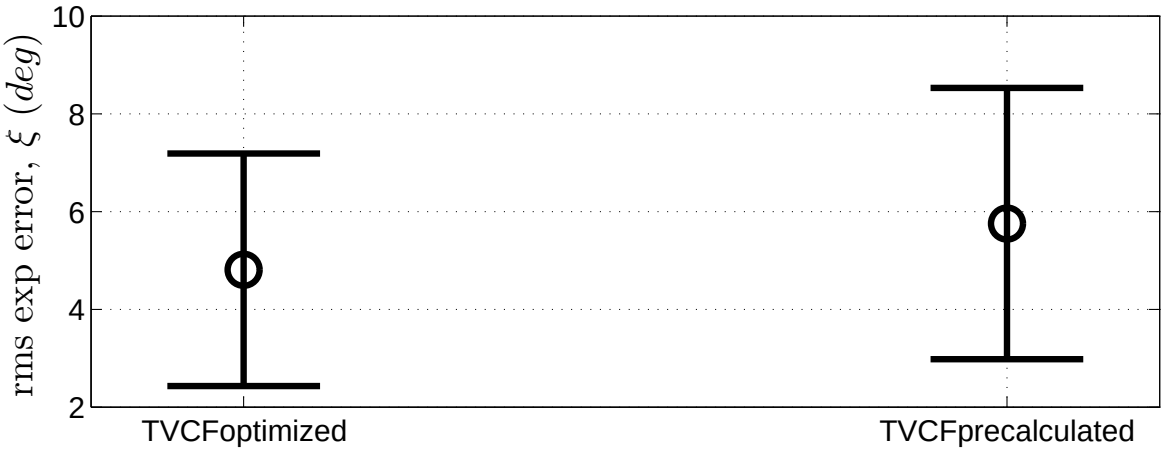



Figure 3.14: Errorbar plot comparing the rms experimental error, $\xi$, of the TVCF with optimized and precalculated parameters. The center circle in each category represents the mean value while the error bars denote one standard deviation.

benefit a general set of applications that require effective yet tractable attitude estimates in pitch and roll, for example Segway robots or sagittal limb orientation estimates in walking robots.

# Chapter 4

# Control of 3DOF orientation with a 2DOF Tail

## 4.1 INTRODUCTION

The ability to swing their tails in any direction gives lizards the remarkable ability to rotate themselves during a fall and land on their feet from any orientation (Fig. 4.1).

These single DOF tailed robots are limited to planar maneuvers, and perturbations out of the plane of the tail are not stabilizable. In contrast to cats and other mammals, lizards rely primarily on their tail for aerial reorientation, using quick sweeps to induce maneuvers in pitch or yaw, and cyclic spiraling motions to produce rolls or other elaborate motions. Here and in [51] we show that a rigid tail with two DOF endows a simple robot (Fig. 4.2) with similar capability.

Solutions to the zero angular momentum multi-link orientation control problem have generally relied on feedforward motion planning. Examples include an optimal control approach [33], Lie Bracket control for roll maneuvers [34], and gradient descent methods [35]. Model predictive control [36] and other computation-intensive online methods may work for space robots, but not for small, rapidly locomoting terrestrial robots, where falls last only fractions of a second. Further, our motivation for this work is that adding a tail be beneficial to existing mobile robot platforms. A major concern in design is that the tail and associated actuation be of relatively low mass cost and not necessitate a major redesign of the robot. An effective controller then should be computationally lightweight so as to not place excessive demands on the existing infrastructure.

We propose a purely feedback controller based on a reduced-complexity model of the system and an intuitive steering idea, and show convergence of the full model in simulation. We implement the controller on a small 175 $(g)$ robotic platform (Fig. 4.2), with a cheap, commercially available microcontroller (Arduino Pro Mini) the controller runs at around 60 $Hz$, including sensing and state estimation. The robot is capable of reorientation from a multi-axis rotation with over 60° eigenangle error, in 100 $(ms)$, isolated pitch/yaw motions of 90°, or right itself from nearly upside-down (150°). Final error after maneuvers is relatively small; in practice the controller need only

orient the bottom of the robot downwards so that legs or suspension can cushion the fall.

This chapter is organized as follows. In section 4.2 we define the generalized coordinates and setup the three dimensional dynamical modeling of the tailed system. In section 4.3 we develop the nonlinear 3 dimensional controller for the 2DOF tail based on the angular momentum constraint. In section 4.4 we present simulation results. In section 4.5 and 4.6 we briefly discuss the design of the two link robot and how the sensing is achieved. In section 4.7 the experimental results are presented and lastly in section 4.8 the conclusions are drawn.

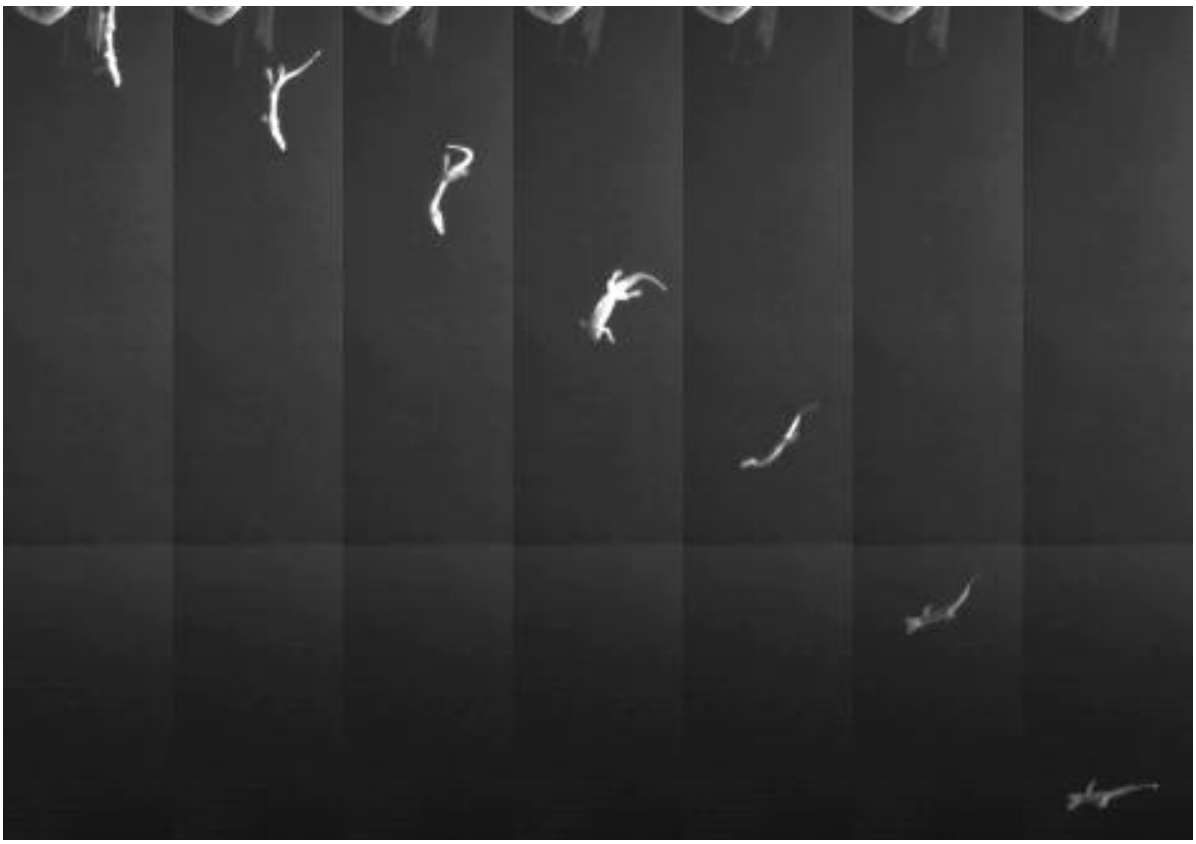


Figure 4.1: Lizard reorienting during a free fall.

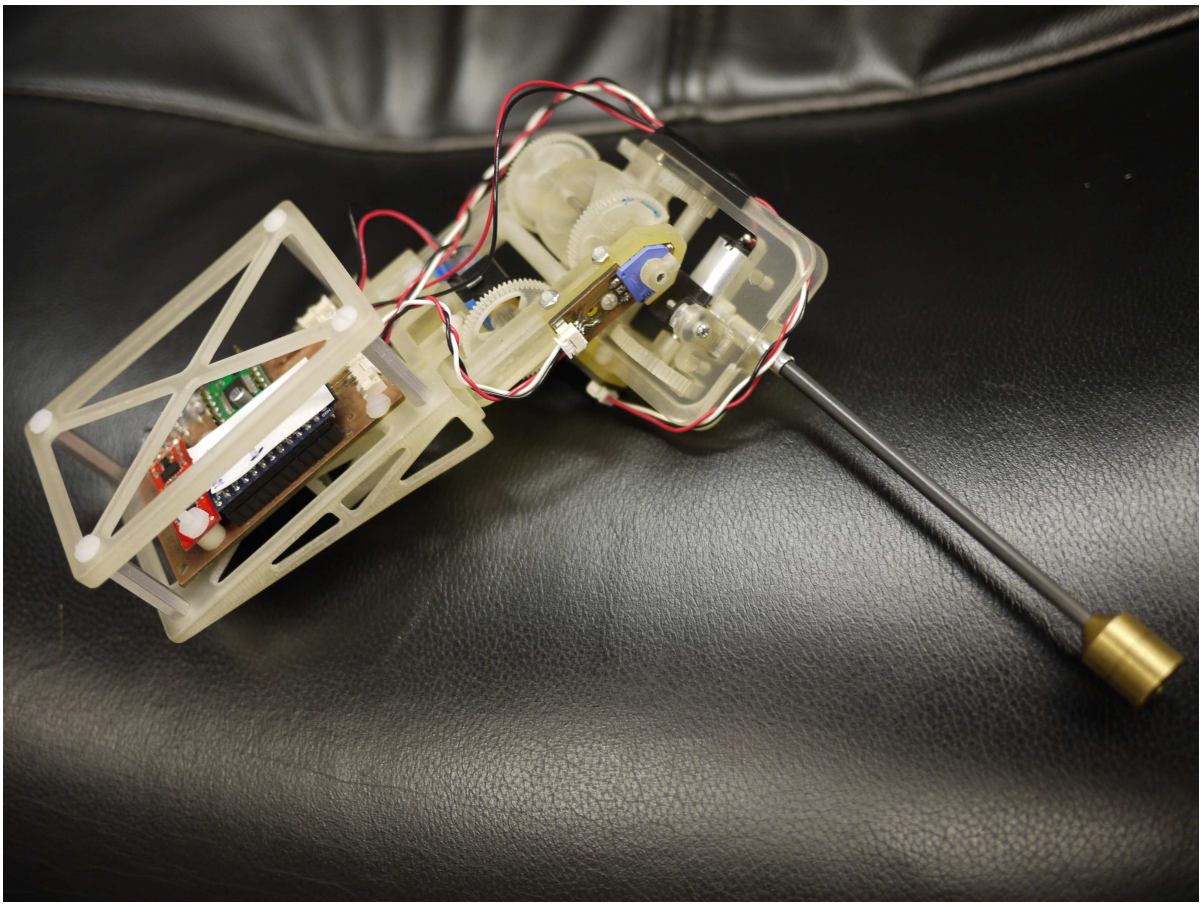


Figure 4.2: Actuated two-DOF *Tailbot*.

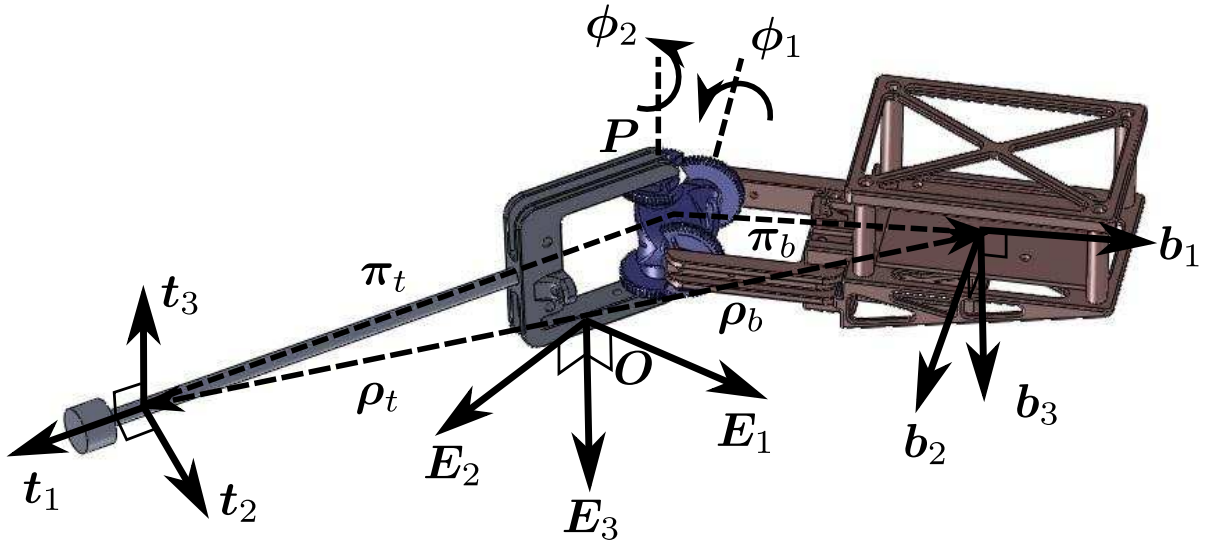



Figure 4.3: Diagram of active-tailed robot. An actuated two-DOF joint connects the robot's body (right, in brown) to its tail (left, in green).

## 4.2 Generalized Coordinates and Modeling

For ease of analysis of the two degree of freedom system, we will assume that the tail is a point mass with length $l_t$ shown in Fig. 4.3. Since we will assume that for any practical tailed system the roll axis is least effective in terms of inertia, we will also restrict our actuation to just two degrees of freedom. Therefore the robot has 5 total DOF. Three for the body orientation and two for the relative tail orientation.

A brief notational review consistent with [52] is made here for rotation matrices and angular velocities. The rotation matrix, $\mathbf{R}^i_j$, represents the rotation between frame $j$ relative to frame $i$, where the columns of $\mathbf{R}$ specify the direction cosines. Another function of the rotation matrix is to map a vector expressed in frame $j$ to one expressed in frame $i$ as

$$\mathbf{r}^i = \mathbf{R}^i_j \mathbf{r}^j. \tag{4.1}$$

An additional current frame rotation becomes

$$\mathbf{r}^i = \mathbf{R}^i_j \mathbf{R}^j_k \mathbf{r}^k \tag{4.2}$$

where

$$\mathbf{R}^i_k = \mathbf{R}^i_j \mathbf{R}^j_k. \tag{4.3}$$

Notice the right hand multiplication of $\mathbf{R}_j^k$, which may be counterintuitive, however this is due to the fact that we are using current rotations as opposed to rotations about a fixed basis [52]. In general, $\boldsymbol{\omega}_{ij}^k$ represents the derivative of $\mathbf{R}_j^i$ expressed in frame $k$. The angular velocities of the body frame, $\mathbf{R}_b^O$, and relative tail frame, $\mathbf{R}_t^b$, expressed in $b$ are then $\boldsymbol{\omega}_{Ob}^b$ and $\boldsymbol{\psi}_{bt}^b$ respectively. For simplicity and with abuse of notation we will refer to these angular velocities as just $\boldsymbol{\omega}$ and $\boldsymbol{\psi}$ respectively.

Note that other texts may define the rotation matrix by the form

$$\begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \mathbf{e}_3 \end{bmatrix} = \mathbf{R} \begin{bmatrix} \mathbf{E}_1 \\ \mathbf{E}_2 \\ \mathbf{E}_3 \end{bmatrix}, \tag{4.4}$$

which is a Vetrix notation [53], where $\mathbf{e}_i$ is the basis in the rotated frame and $\mathbf{E}_i$ is the basis in the fixed frame. This is not to be confused with our definition of the rotation matrix, which would be equivalent to the transpose of Vetrix version.

We define the inertial reference frame (denoted by $O$) with Cartesian basis $\{\mathbf{E_i}\}$, where i = 1, 2, 3, and origin $c$ to be located at the COM of the system. The vector $\mathbf{E}_3$ is oriented downward, for consistency with the aerial vehicle convention. From Fig. 4.3 the set of body fixed orthonormal basis vectors, $\{\mathbf{t}_i\}$ and $\{\mathbf{b}_i\}$, are fixed to the COM of the body and tail. The body fixed axis numbers corresponds to the first being roll, the second pitch, and the third as yaw. This also, conveniently sets the first axis running along the length of the body pointing away from the pivot. A 2-3-1 Euler set is chosen to parameterize the matrix so that the order of rotations from the Cartesian basis is first $\theta_1$ about $\mathbf{b}_2$ (pitch), then $\theta_2$ about $\mathbf{b}_3$ (yaw), and finally $\theta_3$ about $\mathbf{b}_1$ (roll). The details of the 2-3-1 matrix is

$$\mathbf{R}_b^O = \begin{bmatrix} c_{\theta_1}c_{\theta_2} & s_{\theta_1}s_{\theta_3} - c_{\theta_3}s_{\theta_2}c_{\theta_1} & c_{\theta_3}s_{\theta_1} + c_{\theta_1}s_{\theta_2}s_{\theta_3} \\ s_{\theta_2} & c_{\theta_2}c_{\theta_3} & -c_{\theta_2}s_{\theta_3} \\ -s_{\theta_1}c_{\theta_2} & c_{\theta_1}s_{\theta_3} + s_{\theta_1}s_{\theta_2}c_{\theta_3} & c_{\theta_1}c_{\theta_3} - s_{\theta_1}s_{\theta_2}s_{\theta_3} \end{bmatrix} \tag{4.5}$$

where $c_{\theta_i} = \cos(\theta_i)$ and $s_{\theta_i} = \sin(\theta_i)$. The tail rotation matrix is chosen relative to the body frame again with a 2-3-1 Euler set, $\{\phi_1, \phi_2, \phi_3\}$. This provides a clear mathematical description for the physical placement of the actuators located along the relative $\mathbf{t}_2$ (pitch) and $\mathbf{t}_3$ (yaw) axis. Therefore, the motors directly affect the shape velocities $[\dot{\phi}_1, \dot{\phi}_2]^T$ [54]. The last rotation about the $\mathbf{t}_1$ (roll) axis is not actuated and fixed. Therefore, $\phi_3 = 0$ and the tail rotation matrix is simply

$$\mathbf{R}_t^b = \begin{bmatrix} c_{\phi_1}c_{\phi_2} & -c_{\phi_1}s_{\phi_2} & s_{\phi_1} \\ s_{\phi_2} & c_{\phi_2} & 0 \\ -s_{\phi_1}c_{\phi_2} & s_{\phi_1}s_{\phi_2} & c_{\phi_1} \end{bmatrix}. \tag{4.6}$$

With the generalized coordinates $\mathbf{q} = [\theta_1, \theta_2, \theta_3, \phi_1, \phi_2]^T$ defined from the absolute and relative Euler angles we can derive the EOM for the two-link system in robotic mass manipulator form $\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \boldsymbol{\tau}$. The input $\boldsymbol{\tau}$ is the vector of torques, $\mathbf{M}(\mathbf{q})$ is the mass matrix, and $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is the Coriolis matrix. Since the motors are located along the $\phi_1$ and $\phi_2$ coordinates, $\boldsymbol{\tau} = [0, 0, 0, \tau_1, \tau_2]^T$. The details are shown in Appendix C.

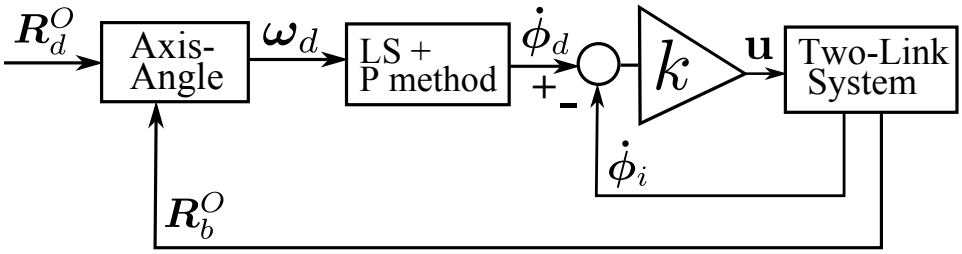



Figure 4.4: Feedback diagram of the proposed nonlinear orientation controller. The outer loop computes a desired body angular velocity $\boldsymbol{\omega}_d$, which is enforced by a simple proportional law on the inner loop shape velocities.

## 4.3 Nonlinear Orientation Control

Our objective is to stabilize the orientation of the robot's body during transient aerial phases (ie, a short fall). Assuming no external forces or torques during this phase, angular momentum is conserved, resulting in a non-holonomic constraint relating the angular velocities of the body and tail. We can control the shape velocities, and hence a component of the total angular momentum; in the case of zero total angular momentum, the other component is strictly a function of body angular velocity $\boldsymbol{\omega}$. Thus our general control strategy is to find the shape velocities that drives $\boldsymbol{\omega}$ to some desired rate. Since the robot is underactuated, this relationship is not perfectly invertible and we must rely on a minimum norm (least squares) solution augmented with a proposed perpendicular method. This feedback strategy is diagrammed in Fig. 4.4 where the outer loop computes a desired $\boldsymbol{\omega}$, which is enforced by a simple proportional law on the inner loop shape velocities.

For simplicity in the synthesis and implementation of the controller we will assume that the pivot is located at the COM of the body, which implies that $l_b = 0$. We justify this by the fact that $l_t \gg l_b$ for most tailed lizard species and our tailed designs will have similar morphometrics. With this assumption, the expression for angular momentum simplifies significantly. For example, when the angular momentum is expressed in the body frame, all the terms are dependent on shape variables and not the body configuration. We will later demonstrate that this controller performs well on a system with $l_b \neq 0$.

### 4.3.1 Angular Momentum

The angular momentum of the 2DOF system, expressed in the inertial reference frame has the compact representation

$$\mathbf{H}^O = \mathbf{J}^O \boldsymbol{\omega}^O + M \boldsymbol{\zeta}^O \times \dot{\boldsymbol{\zeta}}^O \tag{4.7}$$

where $M = \frac{m_b m_t}{m_b + m_t}$, $\mathbf{J}^0$ is the body inertia tensor, $\boldsymbol{\zeta}^O$ is the vector pointing from body COM to tail COM, and $\boldsymbol{\omega}^O$ is the body angular velocity all expressed in the inertial frame [7] (for derivation,

see the Appendix D). Substituting the following relationships

$$\begin{aligned} \boldsymbol{\zeta}^O &= \mathbf{R}_b^O \boldsymbol{\zeta}^b && (4.8)\\ \dot{\boldsymbol{\zeta}}^O &= \dot{\mathbf{R}}_b^O \boldsymbol{\zeta}^b + \mathbf{R}_b^O \dot{\boldsymbol{\zeta}}^b && (4.9)\\ \mathbf{J}^O \boldsymbol{\omega}^O &= \mathbf{R}_b^O \mathbf{J}^b \boldsymbol{\omega} && (4.10) \end{aligned}$$

into (4.7) leads to

$$\mathbf{H}^O = \mathbf{R}_b^O \mathbf{J}^b \boldsymbol{\omega} + M(\mathbf{R}_b^O \boldsymbol{\zeta}^b) \times (\dot{\mathbf{R}}_b^O \boldsymbol{\zeta}^b) + M(\mathbf{R}_b^O \boldsymbol{\zeta}^b) \times (\mathbf{R}_b^O \dot{\boldsymbol{\zeta}}^b) \tag{4.11}$$

To express 4.11 in the body frame we left multiply by $(\mathbf{R}_b^O)^T$ and also use the two properties;

$$\mathbf{R}(\mathbf{a} \times \mathbf{b}) = (\mathbf{R}\mathbf{a}) \times (\mathbf{R}\mathbf{b}) \tag{4.12}$$

for vectors $\mathbf{a}$ and $\mathbf{b}$ and rotation matrix $\mathbf{R}$ and that the transpose of a rotation matrix is its inverse. This results in

$$\mathbf{H}^b = \mathbf{J}^b \boldsymbol{\omega} + M\boldsymbol{\zeta}^b \times (\mathbf{R}_b^O)^T \dot{\mathbf{R}}_b^O \boldsymbol{\zeta}^b + M\boldsymbol{\zeta}^b \times \dot{\boldsymbol{\zeta}}^b. \tag{4.13}$$

Using the identity [55]

$$\hat{\boldsymbol{\omega}}_{On}^n = (\mathbf{R}_n^O)^T \dot{\mathbf{R}}_n^O \tag{4.14}$$

where the skew symmetric matrix operator is

$$\hat{\mathbf{a}} = \mathbf{a}\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}, \tag{4.15}$$

the second right hand term in (4.13) can be rewritten as

$$\begin{aligned} M\boldsymbol{\zeta}^b \times (\mathbf{R}_b^O)^T \dot{\mathbf{R}}_b^O \boldsymbol{\zeta}^b &= M(\boldsymbol{\zeta}^b\times)\hat{\boldsymbol{\omega}}\boldsymbol{\zeta}^b && (4.16)\\ &= M\hat{\boldsymbol{\zeta}}^b \hat{\boldsymbol{\omega}} \boldsymbol{\zeta}^b && (4.17)\\ &= M\hat{\boldsymbol{\zeta}}^b (-\hat{\boldsymbol{\zeta}}^b \boldsymbol{\omega}) && (4.18)\\ &= -M\hat{\boldsymbol{\zeta}}^b \hat{\boldsymbol{\zeta}}^b \boldsymbol{\omega}. && (4.19) \end{aligned}$$

The final term of (4.13) can be decomposed into the product of a matrix $\mathbf{F}_{3\times 2}$ and the shape velocity vector along with (4.19) leads to the representation

$$\mathbf{H}^b = (\mathbf{J}^b - M\hat{\boldsymbol{\zeta}}^b \hat{\boldsymbol{\zeta}}^b)\boldsymbol{\omega} + \mathbf{F} \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \end{bmatrix}. \tag{4.20}$$

In the body frame, $\mathbf{J}^b$ is assumed diagonal and with $l_b = 0$, $\boldsymbol{\zeta}^b$ is simply

$$\begin{aligned} \boldsymbol{\zeta}^b &= \mathbf{R}_t^b [l_t, 0, 0]^T && (4.21)\\ &= l_t \begin{bmatrix} \cos(\phi_1)\cos(\phi_2) \\ \sin(\phi_2) \\ -\sin(\phi_1)\cos(\phi_2) \end{bmatrix}. && (4.22) \end{aligned}$$

Its derivative is

$$\dot{\boldsymbol{\zeta}}^b = \dot{\mathbf{R}}_t^b [l_t, 0, 0]^T \tag{4.23}$$

$$= l_t \begin{bmatrix} -s_{\phi_1} c_{\phi_2} \dot{\phi}_1 - c_{\phi_1} s_{\phi_2} \dot{\phi}_2 \\ c_{\phi_2} \dot{\phi}_2 \\ -c_{\phi_1} c_{\phi_2} \dot{\phi}_1 + s_{\phi_1} s_{\phi_2} \dot{\phi}_2 \end{bmatrix}. \tag{4.24}$$

The matrix $\mathbf{F}$, where

$$\mathbf{F} \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \end{bmatrix} = M \hat{\boldsymbol{\zeta}}^b \dot{\boldsymbol{\zeta}}^b \tag{4.25}$$

$$= M l_t^2 \begin{bmatrix} 0 & s_{\phi_1} c_{\phi_1} & s_{\phi_2} \\ -s_{\phi_1} c_{\phi_2} & 0 & -c_{\phi_1} c_{\phi_2} \\ -s_{\phi_2} & c_{\phi_1} c_{\phi_2} & 0 \end{bmatrix} \begin{bmatrix} -s_{\phi_1} c_{\phi_2} \dot{\phi}_1 - c_{\phi_1} s_{\phi_2} \dot{\phi}_2 \\ c_{\phi_2} \dot{\phi}_2 \\ -c_{\phi_1} c_{\phi_2} \dot{\phi}_1 + s_{\phi_1} s_{\phi_2} \dot{\phi}_2 \end{bmatrix} \tag{4.26}$$

$$= M l_t^2 \begin{bmatrix} -c_{\phi_1} c_{\phi_2} s_{\phi_2} \dot{\phi}_1 + s_{\phi_1} \dot{\phi}_2 \\ c_{\phi_2}^2 \dot{\phi}_1 \\ s_{\phi_1} c_{\phi_2} s_{\phi_2} \dot{\phi}_1 + c_{\phi_1} \dot{\phi}_2 \end{bmatrix} \tag{4.27}$$

$$= M l_t^2 \begin{bmatrix} -\cos(\phi_1)\cos(\phi_2)\sin(\phi_2) & \sin(\phi_1) \\ \cos(\phi_2)^2 & 0 \\ \sin(\phi_1)\cos(\phi_2)\sin(\phi_2) & \cos(\phi_1) \end{bmatrix} \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \end{bmatrix}, \tag{4.28}$$

is dependent solely on the shape angles and is a mapping from the shape velocities to their contribution to total angular momentum. Notice that when the shape is constant, the total angular momentum only depends on $\boldsymbol{\omega}$; hence the term $(\mathbf{J}^b - M\hat{\boldsymbol{\zeta}}^b \hat{\boldsymbol{\zeta}}^b)$ is the configuration-dependent moment of inertia of the body-tail system about its center of mass, which we will denote $\mathbf{J}_c$.

By adding the introduced terms to (4.20), we arrive at an elegant expression for the total angular momentum in the body frame:

$$\mathbf{H}^b = \mathbf{J}_c \boldsymbol{\omega} + \mathbf{F} \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \end{bmatrix} \tag{4.29}$$

where

$$\mathbf{J}_c = \begin{bmatrix} J_1 & 0 & 0 \\ 0 & J_2 & 0 \\ 0 & 0 & J_3 \end{bmatrix} - M l_t^2 \begin{bmatrix} c_{\phi_1}^2 c_{\phi_2}^2 - 1 & c_{\phi_1} c_{\phi_2} s_{\phi_2} & -s_{\phi_1} c_{\phi_1} c_{\phi_2}^2 \\ c_{\phi_1} c_{\phi_2} s_{\phi_2} & -c_{\phi_2}^2 & -s_{\phi_1} c_{\phi_2} s_{\phi_2} \\ -s_{\phi_1} c_{\phi_1} c_{\phi_2}^2 & -s_{\phi_1} c_{\phi_2} s_{\phi_2} & s_{\phi_1}^2 c_{\phi_2}^2 - 1 \end{bmatrix} \tag{4.30}$$

$$= M l_t^2 \begin{bmatrix} \frac{J_1}{M l_t^2} + (1 - c_{\phi_1}^2 c_{\phi_2}^2) & -c_{\phi_1} s_{\phi_2} c_{\phi_2} & s_{\phi_1} c_{\phi_1} c_{\phi_2}^2 \\ -c_{\phi_1} s_{\phi_2} c_{\phi_2} & \frac{J_2}{M l_t^2} + c_{\phi_2}^2 & s_{\phi_1} s_{\phi_2} c_{\phi_2} \\ s_{\phi_1} c_{\phi_1} c_{\phi_2}^2 & s_{\phi_1} s_{\phi_2} c_{\phi_2} & \frac{J_3}{M l_t^2} + (1 - s_{\phi_1}^2 c_{\phi_2}^2) \end{bmatrix}. \tag{4.31}$$

### 4.3.2 Orientation Control

We pose the problem of seeking a desired orientation $\mathbf{R}_d^O$ for the body link, while at a current orientation $\mathbf{R}_b^O$ for a zero angular momentum maneuver, $\mathbf{H}^b = \mathbf{0}$. With slight abuse of notation, the error rotation, $\mathbf{R}_e = \mathbf{R}_d^b$. With this and the identity in (4.3) the two frames are related by

$$\mathbf{R}_d^O = \mathbf{R}_b^O \mathbf{R}_e. \tag{4.32}$$

Solving for $\mathbf{R}_e$ is simply

$$\mathbf{R}_e = (\mathbf{R}_b^O)^T \mathbf{R}_d^O. \tag{4.33}$$

Converting $\mathbf{R}_e$ into an axis and angle representation likens the magnitude of error to the angle $\theta_e$ and the error direction to the axis $\mathbf{v}_e$. These quantities are parameterized by

$$\theta_e = \cos^{-1}\left(\frac{\mathrm{Tr}(\mathbf{R}_e) - 1}{2}\right) \tag{4.34}$$

$$\mathbf{v}_e = \frac{1}{2\sin(\theta_e)} \begin{bmatrix} \mathbf{R}_{e3,2} - \mathbf{R}_{e2,3} \\ \mathbf{R}_{e1,3} - \mathbf{R}_{e3,1} \\ \mathbf{R}_{e2,1} - \mathbf{R}_{e1,2} \end{bmatrix}. \tag{4.35}$$

Note that any body angular velocity, $\boldsymbol{\omega}$, parallel to $\mathbf{v}$ will act to reduce orientation error; hence a simple outer loop proportional controller defined by

$$\boldsymbol{\omega}_d = k_e \theta_e \mathbf{v}_e \tag{4.36}$$

should suffice to drive the error towards zero; where $k_e > 0$ is a tuned constant.

Since the formulation of (4.34) and (4.35) contain a singularity at $\theta_e = 0$, but with a finite limit, the quantity $\theta_e \mathbf{v}_e$ can be rewritten as

$$\theta_e \mathbf{v}_e = f(\theta_e) \cdot \frac{1}{2} \begin{bmatrix} \mathbf{R}_{e3,2} - \mathbf{R}_{e2,3} \\ \mathbf{R}_{e1,3} - \mathbf{R}_{e3,1} \\ \mathbf{R}_{e2,1} - \mathbf{R}_{e1,2} \end{bmatrix} \tag{4.37}$$

where $f(\theta_e)$ is the Taylor expansion of $\theta_e / \sin(\theta_e)$. With three Taylor expansion terms the approximation only has an error of 2.6% at $\theta_e = \pi/2$ *rad*.

### 4.3.3 Shape Control

We now derive our inner loop controller by finding a set of shape velocities $[\dot{\phi}_{1LS}, \dot{\phi}_{2LS}]^T$ such that $\boldsymbol{\omega}$ in (4.29) is driven to $\boldsymbol{\omega}_d$. Since $\mathbf{F}$ has more rows than columns, it cannot be inverted exactly. Instead, we seek the shape velocities that result in the best fit to our desired angular velocity, $\boldsymbol{\omega}_d$:

$$\begin{bmatrix} \dot{\phi}_{1LS} \\ \dot{\phi}_{2LS} \end{bmatrix} = \mathbf{F}^{\dagger} \mathbf{H}_s = \mathbf{F}^{\dagger} (\mathbf{H}^b - \mathbf{J}_c \boldsymbol{\omega}_d) \tag{4.38}$$

where $\mathbf{F}^\dagger = (\mathbf{F}^T\mathbf{F})^{-1}\mathbf{F}^T$ is the generalized left least squares inverse or pseudoinverse and can be written fully as

$$\mathbf{F}^\dagger = \left((Ml_t^2)^2 \begin{bmatrix} c_{\phi_2}^2 & 0 \\ 0 & 1 \end{bmatrix}\right)^{-1} Ml_t^2 \begin{bmatrix} -c_{\phi_1}c_{\phi_2}s_{\phi_2} & c_{\phi_2}^2 & s_{\phi_1}c_{\phi_2}s_{\phi_2} \\ s_{\phi_1} & 0 & c_{\phi_1} \end{bmatrix} \tag{4.39}$$

$$= \frac{1}{Ml_t^2} \begin{bmatrix} -\cos(\phi_1)\tan(\phi_2) & 1 & \sin(\phi_1)\tan(\phi_2) \\ \sin(\phi_1) & 0 & \cos(\phi_1) \end{bmatrix}. \tag{4.40}$$

For zero angular momentum falls $\mathbf{H}^b = 0$ and the vector $\mathbf{H}_s$ is the desired contribution of angular momentum from the shape variable velocities expressed in $b$ that will drive the body to the desired configuration. It is written fully as

$$\mathbf{H}_s = -\mathbf{J}_c k_e \theta_e \mathbf{v}_e. \tag{4.41}$$

Pre multiplying $\mathbf{F}^\dagger$ with $\mathbf{J}_c$ becomes

$$\mathbf{F}^\dagger\mathbf{J}_c = \frac{1}{Ml_t^2} \begin{bmatrix} -c_{\phi_1}t_{\phi_2}(J_1+Ml_t^2) & (J_2+Ml_t^2) & s_{\phi_1}t_{\phi_2}(J_3+Ml_t^2) \\ s_{\phi_1}(J_1+Ml_t^2) & 0 & c_{\phi_1}(J_3+Ml_t^2) \end{bmatrix} \tag{4.42}$$

which is essentially a column scaled version of $\mathbf{F}^\dagger$. This significantly eases the ability to implement and tune the controller.

### Error Properties in Least Squares Method

The error in the LS method from (4.38) is the difference between the actual $\mathbf{H}_s$ and the one computed by the LS method, which is written as

$$\mathbf{e}_{LS} = \mathbf{H}_s - \mathbf{F}\mathbf{F}^\dagger\mathbf{H}_s \tag{4.43}$$

$$= (\mathbf{I}_{3\times3} - \mathbf{F}\mathbf{F}^\dagger)\mathbf{H}_s \tag{4.44}$$

where

$$\mathbf{I}_{3\times3} - \mathbf{F}\mathbf{F}^\dagger = \mathbf{I}_{3\times3} - \begin{bmatrix} 1-c_{\phi_1}^2c_{\phi_2}^2 & -c_{\phi_1}c_{\phi_2}s_{\phi_2} & c_{\phi_1}s_{\phi_1}c_{\phi_2}^2 \\ -c_{\phi_1}c_{\phi_2}s_{\phi_2} & c_{\phi_2}^2 & s_{\phi_1}c_{\phi_2}s_{\phi_2} \\ c_{\phi_1}s_{\phi_1}c_{\phi_2}^2 & s_{\phi_1}c_{\phi_2}s_{\phi_2} & 1-s_{\phi_1}^2c_{\phi_2}^2 \end{bmatrix} \tag{4.45}$$

$$= \begin{bmatrix} c_{\phi_1}^2c_{\phi_2}^2 & c_{\phi_1}c_{\phi_2}s_{\phi_2} & -c_{\phi_1}s_{\phi_1}c_{\phi_2}^2 \\ c_{\phi_1}c_{\phi_2}s_{\phi_2} & s_{\phi_2}^2 & -s_{\phi_1}c_{\phi_2}s_{\phi_2} \\ -c_{\phi_1}s_{\phi_1}c_{\phi_2}^2 & -s_{\phi_1}c_{\phi_2}s_{\phi_2} & s_{\phi_1}^2c_{\phi_2}^2 \end{bmatrix}. \tag{4.46}$$

Since the error is dependent on tail configuration with respect to $\mathbf{H}_s$, we are interested in the set of tail configurations that nullify or maximize the error.

The unity norm tail vector expressed in the body frame is $\mathbf{t}_1^b$ (from here we will just denote as $\mathbf{t}$). This vector can be written as

$$\begin{aligned} \mathbf{t} &= \mathbf{R}_t^b [1,0,0]^T && (4.47) \\ &= \begin{bmatrix} \cos(\phi_1)\cos(\phi_2) \\ \sin(\phi_2) \\ -\sin(\phi_1)\cos(\phi_2) \end{bmatrix} && (4.48) \end{aligned}$$

and it can be easily confirmed through inspection that

$$\mathbf{I}_{3\times 3} - \mathbf{F}\mathbf{F}^{\dagger} = \mathbf{t}\mathbf{t}^T. \tag{4.49}$$

Therefore the least squares error can be interpreted as

$$\begin{aligned} \mathbf{e}_{LS} &= \mathbf{t}\mathbf{t}^T \mathbf{H}_s && (4.50) \\ &= \mathbf{t}(\mathbf{t}\cdot\mathbf{H}_s). && (4.51) \end{aligned}$$

Hence, for a nontrivial $\mathbf{H}_s$, $\mathbf{e}_{LS} = \mathbf{0}$ when the dot product $\mathbf{t}\cdot\mathbf{H}_s = 0$. This physically implies that the error in the LS method is nullified when $\mathbf{t}$ and $\mathbf{H}_s$ are orthogonal.

In fact, $\mathbf{e}_{LS}$ is maximized when $\mathbf{t}$ and $\mathbf{H}_s$ are aligned, which also corresponds to $\mathbf{H}_s$ existing in the nullspace of $\mathbf{F}^{\dagger}$ and would render the LS method from (4.38) inactive. This is shown in Appendix E.

As a result, we seek to augment the LS method with a control scheme that not only avoids the situation when $\mathbf{t}$ and $\mathbf{H}_s$ are aligned, but also drives them to a perpendicular configuration (where the LS method is most effective).

### 4.3.4 Perpendicular Method

Appendix F derives a useful property that the LS method computes an angular velocity that shares the plane containing $\mathbf{t}$ and $\mathbf{H}_s$. Based on this fact and the conclusion of the previous section we propose to inject a new desired angular velocity vector $\boldsymbol{\psi}_{PD}$ perpendicular to the plane containing $\mathbf{t}$ and $\mathbf{H}_s$. This can be accomplished by

$$\boldsymbol{\psi}_{PD} = \gamma \frac{\mathbf{t}\times\mathbf{H}_s}{||\mathbf{t}\times\mathbf{H}_s||}, \tag{4.52}$$

which will not only improve the LS method's effectiveness, but also avoid competing with it. To drive the new angular velocity so that the tail and $\mathbf{H}_s$ are perpendicular we set the value of $\gamma$ as proportional to the difference between their included angle and $\pi/2$ *rad* as

$$\gamma = -k_P \left( \frac{\pi}{2} - \cos^{-1}\left( \frac{\mathbf{t}}{||\mathbf{t}||} \cdot \frac{\mathbf{H}_s}{||\mathbf{H}_s||} \right) \right) \tag{4.53}$$

where $k_P > 0$ is a tuned constant. Finally, the new desired shape velocities can be computed from

$$\begin{bmatrix} \dot{\phi}_{1P} \\ \dot{\phi}_{2P} \end{bmatrix} = \mathbf{G}^T \boldsymbol{\psi}_{PD} \tag{4.54}$$

where $\mathbf{G}^T$ is equivalent to the pseudo-inverse of $\mathbf{G}$ in (F.6). A general example is diagrammed in Fig. 4.5, where the dashed line represents the instantaneous orientation where the tail is most effective. From (4.52), the direction of $\boldsymbol{\psi}_{PD}$ is pointed toward the reader resulting in a clockwise rotation. By inspection of the vectors and signs of $\gamma$ in (4.53), the sign is left unchanged and $\boldsymbol{\psi}_{PD}$ drives the tail towards the dashed line as desired.

Both (4.52) and (4.53) contain a singularity. When the error angle $\theta_e = 0$, implying $||\mathbf{H}_s|| = 0$, then the desired orientation has been achieved and we can set

$$\boldsymbol{\psi}_{PD} = [0,0,0]^T. \tag{4.55}$$

However, when $||\mathbf{H}_s|| \neq 0$ and $||\mathbf{t} \times \mathbf{H}_s|| = 0$, implies that $\mathbf{H}_s$ is nontrivial and parallel to $\mathbf{t}$. Without any conditioning, $\boldsymbol{\psi}_{PD}$ will be poorly defined since $||\mathbf{t} \times \mathbf{H}_s||$ appears in the denominator of (4.52), however if this singularity case is detected we can escape it by setting

$$\boldsymbol{\psi}_{PD} = [0,-1,0]^T. \tag{4.56}$$

This initiates motion in the negative $\phi_1$ direction (positive $\theta_1$ direction), since pure motion in the $\phi_2$ direction will lead to a physical singularity at $\pi/2$ *rad*.

A simple example of this scheme's utility arises when the robot is simply rolled upside-down with the tail initially aligned parallel to the body. In this special case, the pure roll error will create a nonzero $\mathbf{H}_s$ that is parallel with both the body and the tail, producing no response in the LS-method (see simulation results).

While the error in the P-method is in general nonzero, we show in Appendix G that the desired angular velocity from the P-method, $\boldsymbol{\psi}_P$, is perpendicular to $\boldsymbol{\psi}_{LS}$.

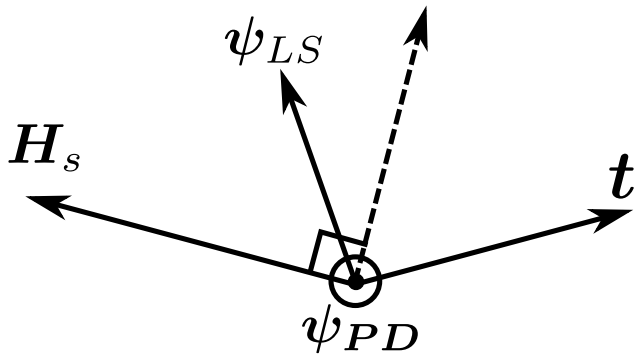



Figure 4.5: Diagram of LS and P method angular velocities relative to $\mathbf{H}_s$ and $\mathbf{t}$. The dashed line is the instantaneous configuration where the tail would be most effective and is perpendicular to $\mathbf{H}_s$. The desired angular velocity vector from the P method, $\boldsymbol{\psi}_{PD}$, is pointing directly out of the page and intends to drive the tail towards the dashed line.

### 4.3.5 Control Law

Using the previously obtained result that the LS and P method will produce perpendicular desired angular velocities, we conclude that we can conveniently obtain the desired shape velocities by adding $\dot{\phi}_{iP}$ to $\dot{\phi}_{iLS}$ directly:

$$\begin{bmatrix} \dot{\phi}_{1d} \\ \dot{\phi}_{2d} \end{bmatrix} = \begin{bmatrix} \dot{\phi}_{1LS} \\ \dot{\phi}_{2LS} \end{bmatrix} + \begin{bmatrix} \dot{\phi}_{1P} \\ \dot{\phi}_{2P} \end{bmatrix}. \tag{4.57}$$

The actual shape variable velocities, which are estimated from the time differentiation of the joint position sensors, are driven towards the desired shape variable velocities, $[\dot{\phi}_1, \dot{\phi}_2]^T$, through a simple proportional feedback law

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = k \left( \begin{bmatrix} \dot{\phi}_{1d} \\ \dot{\phi}_{2d} \end{bmatrix} - \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \end{bmatrix} \right) \tag{4.58}$$

where $k > 0$ is a proportional gain. A more robust controller may be required in the future, but this proportional one has been found to be currently sufficient. With $\mathbf{H}_s = -\mathbf{J}_c k_e \theta_e \mathbf{v}_e$ from (4.41), the combined control law is

$$\mathbf{u} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = k(\mathbf{F}^{\dagger}\mathbf{H}_s + \mathbf{G}^T \boldsymbol{\psi}_{PD} - [\dot{\phi}_1, \dot{\phi}_2]^T) \tag{4.59}$$

where

$$\boldsymbol{\psi}_{PD} = \begin{cases} [0,0,0]^T & \text{if } \theta_e = 0; \\ [0,-1,0]^T & \text{if } ||\mathbf{H}_s|| \neq 0 \ \& \ ||\mathbf{t} \times \mathbf{H}_s|| = 0; \\ \gamma \frac{\mathbf{t} \times \mathbf{H}_s}{||\mathbf{t} \times \mathbf{H}_s||} & \text{otherwise.} \end{cases} \tag{4.60}$$

$$\gamma = -k_P \left( \frac{\pi}{2} - \cos^{-1} \left( \frac{\mathbf{t}}{||\mathbf{t}||} \cdot \frac{\mathbf{H}_s}{||\mathbf{H}_s||} \right) \right). \tag{4.61}$$

**Summary of Control Strategy**

- Though gyroscope integration, compute the quaternion, $\mathbf{q}$, that parameterizes the orientation.
- Convert the quaternion to the current configuration, $\mathbf{R}_b^O$.
- Compare the current configuration and desired configuration, $\mathbf{R}_d^O$, and compute the error axis, $\mathbf{v}_e$, and error angle, $\theta_e$.
- Set the desired angular velocity, $\boldsymbol{\omega}_d$, of the body proportional to the error axis and angle.
- The combined LS and P-method produces a desired relative tail rate, $\dot{\phi}_{id}$. The P-method accounts for when the desired angular velocity is improperly aligned.
- Obtain the relative velocity, $\dot{\phi}_i$, from the derivative of the tail joint or motor position.
- Enforce the desired velocity, $\dot{\phi}_{id}$, using a simple proportional feedback loop.

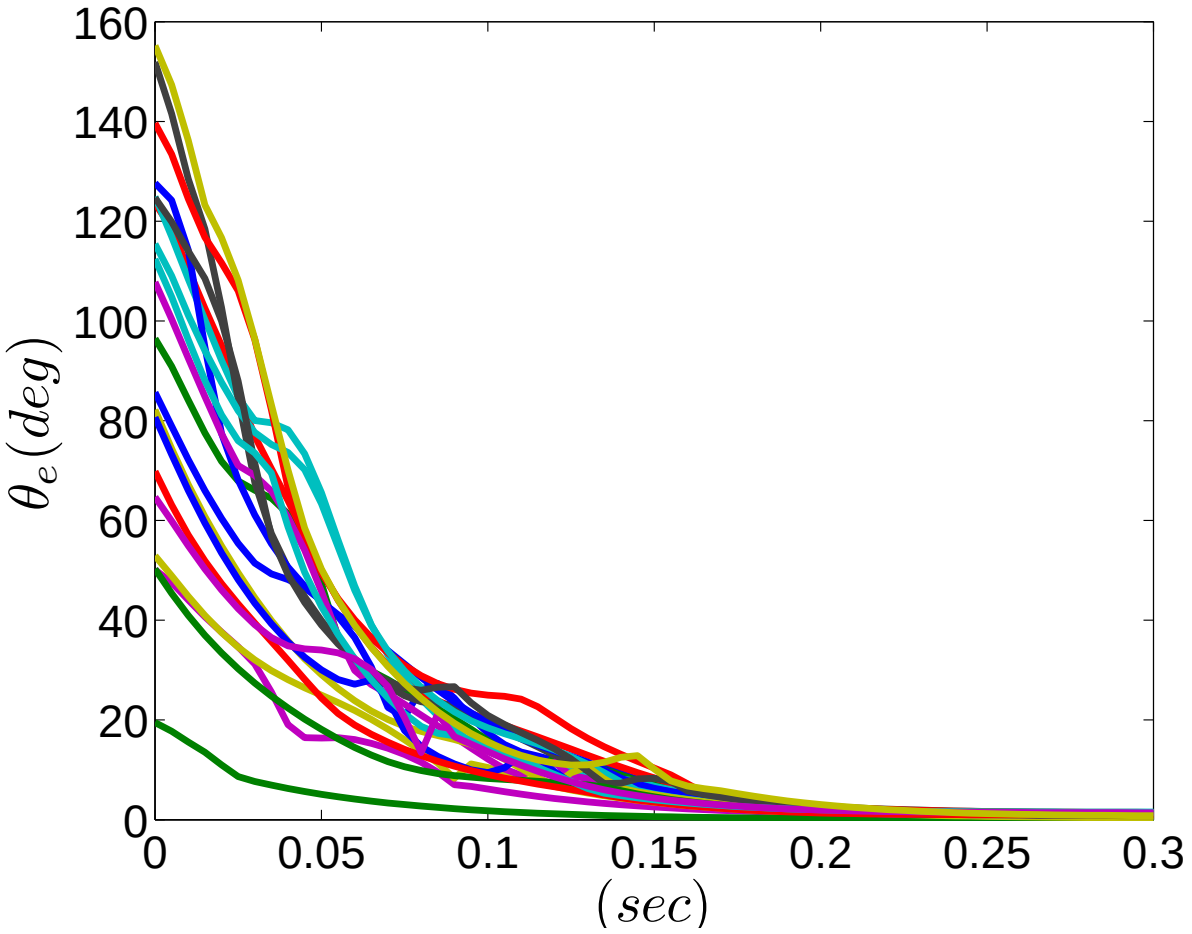



Figure 4.6: Simulation results of the error angle at 20 trials starting at random initial conditions.

## 4.4 SIMULATION RESULTS

Here the controller was implemented on the two link model with the parameters and gains listed in Table 4.1. The desired reference configuration was set to the identity matrix. For each trial a random initial orientation for the body was computed. The first two initial Euler angles varied within $\pm 0.9 \cdot \pi/2$ $(rad)$ and the third initial Euler angle varied within $\pm 0.9 \cdot \pi$ $(rad)$. Figure 4.6 is a plot of the error angle for each trial computed from (4.34). Twenty trials are presented and it is clear from the figure that all trials converge to within $5^\circ$ after 200 $(ms)$. Figure 4.7 is a time framed sequence of a representative simulation where the system is able to simultaneously correct in all three axes.

To test that the P-method was effective we initialized the simulation to have only a roll offset. This results in a nonzero desired $\mathbf{H}_s$ along the axis of the robot, which is also parallel to the tail

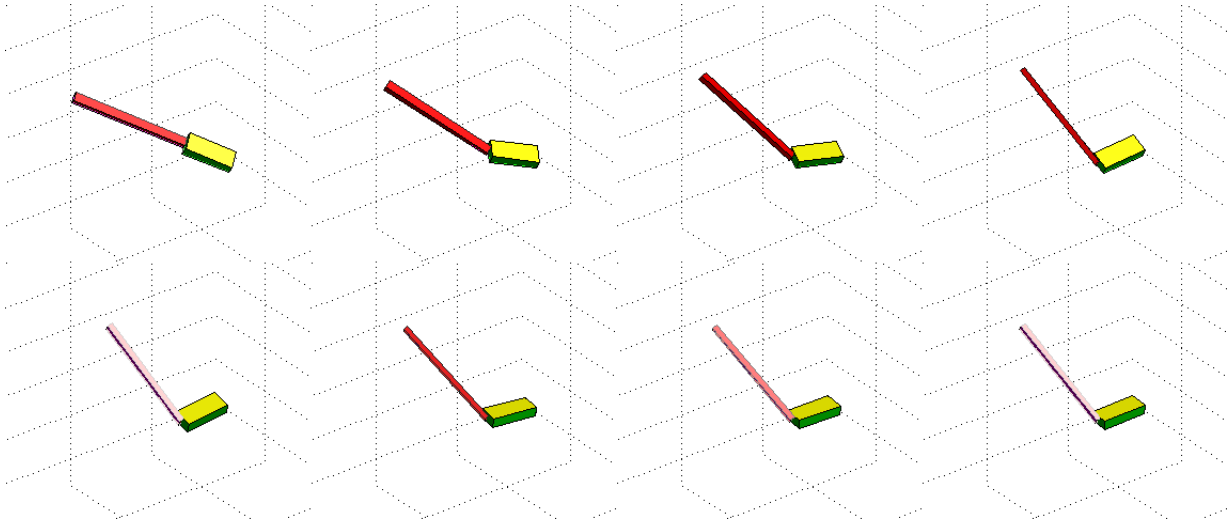


Figure 4.7: Sequence of frames during a correction in pitch, yaw, and roll in simulation. The sequence starts at the upper left and continues to the lower right. The desired body orientation is horizontal with the yellow face up and pointed towards the right.

Table 4.1: Table of Simulation Parameters

| Component | Pitch | Yaw | Roll | Units |
|---|---|---|---|---|
| Body | 0.7 | 2.5 | 2.6 | $(\times 10^{-4})[\text{kg} \cdot m^s]$ |
| $m_b$ | 113 | | | [g] |
| $m_t$ | 57 | | | |
| $l_b$ | 5.2 | | | [cm] |
| $l_t$ | 12 | | | |
| $k_e$ | 25 | | | [none] |
| $k_P$ | 1000 | | | |
| $k$ | 1 | | | |

and initiates the condition in (4.56). Without the P-method, the LS-method would have computed zero torque. Figure 4.8 shows that the P-method is able to steer the tail out of the null space of the LS-method, perturbing the first and second Euler angles away from zero. Then the combined control is able to steer all of the Euler angles to the desired orientation of approximately $0^\circ$. Figure 4.9 is a time framed sequence of a simulation where the system is able to correct in roll when the tail is initially aligned with the body.

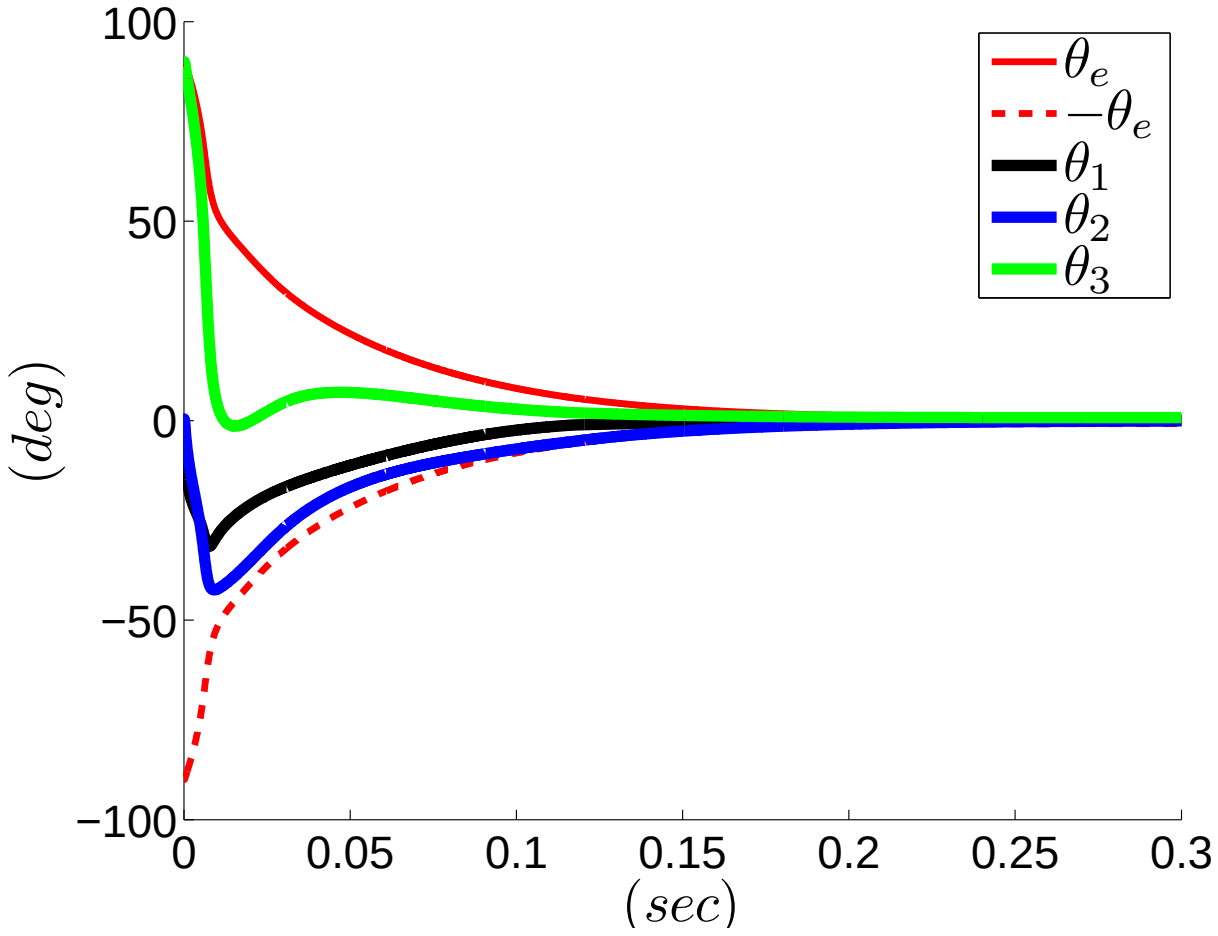



Figure 4.8: Simulation results for a pure roll and no tail offset. Notice that the first and second Euler angles are perturbed away from zero, but corrected in the steady state.

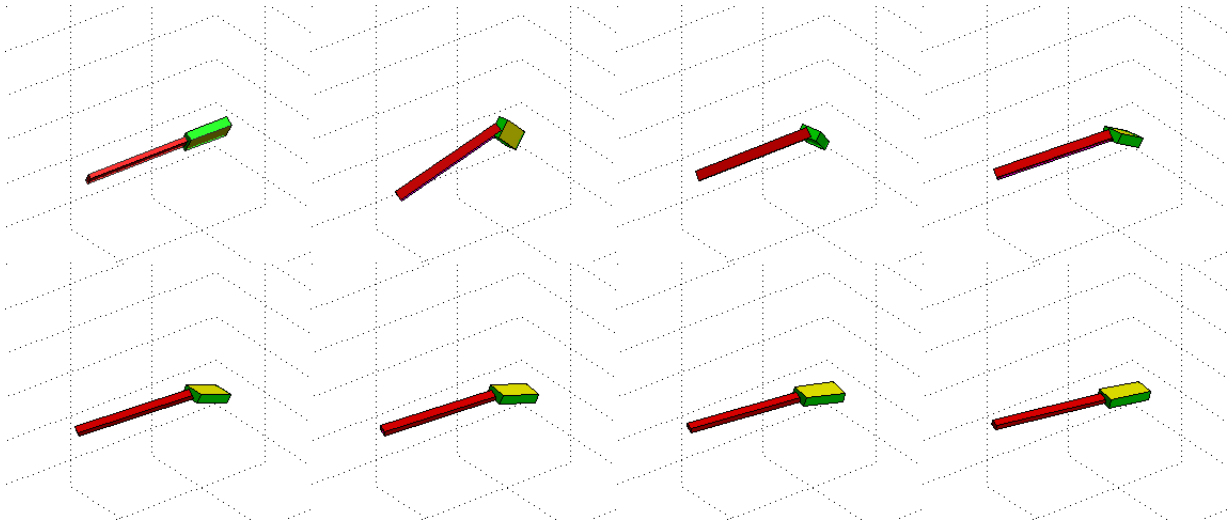


Figure 4.9: Sequence of frames during a roll correction with tail initially aligned with the body in simulation. This would be impossible without the perpendicular method. The sequence starts at the upper left and continues to the lower right. The desired body orientation is horizontal with the yellow face up and pointed towards the right.

Table 4.2: Inertial Measurements of the Robot $(\times 10^{-4})[\mathrm{kg}\cdot\mathrm{m}^2]$

| Component | Pitch | Yaw | Roll | Units |
|---|---|---|---|---|
| Tail | 4.79 | 4.782 | 0.4593 | $(\times 10^{-4})[\mathrm{kg}\cdot\mathrm{m}^2]$ |
| Body | 2.071 | 2.209 | 1.259 | |
| $m_b$ | 105 | | | [g] |
| $m_t$ | 70 | | | |
| $l_b$ | 5.2 | | | [cm] |
| $l_t$ | 7.3 | | | |

## 4.5 ROBOT DESIGN

To verify the simulation results, we created the two link robot shown in Fig. 4.2. The actuated two DOF tail joint was inspired by a compact universal joint used in a snake robot [56]. Connecting the two links is a center gear that allows orthogonal rotation. One motor, fixed to the body link drives one side of the center gear and the second motor fixed to the tail link drives the other side of the center gear allowing actuation along the $\mathbf{t}_2$ and $\mathbf{t}_3$ axis.

The robot was created using a *ProJet HD3000* 3D printer. The design intent was to ensure durability throughout the tests.

Tail effectiveness is defined as the ratio of tail to body rates for a two link planar system with angular momentum conservation [37]. We chose an effectiveness of 1.3, which is the effectiveness needed to move the body approximately 90° by moving the tail through its full stroke.

Similar to the 1DOF *Tailbot*, this 2DOF version contains an onboard *Arduino* pro-mini microprocessor 3.3 $(V)$, 8 $(MHz)$, a *Nordic* wireless transceiver, IMU, *Pololu* FNG-6612 motor driver board and LED indicators. It is also powered by two Lithium Polymer 7.4$V$ 138 $(mAh)$ batteries. The relative tail angle was measured with a *Murata*, multi turn potentiometer.

The inertial properties of the robot are verified through the pendulum method [40]. The parameters of the robot are listed in Table 4.2.

## 4.6 SENSING

Quaternions are a convenient way to parameterize the orientation of the robot body. By numerical integration via the measurements from a triple-axis gyroscope, we can estimate the orientation quaternion [57]. The process is detailed here. The quaternion contains a scalar and vector compo-

nent as

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 i \\ q_3 j \\ q_4 k \end{bmatrix}. \tag{4.62}$$

The quaternion derivative is

$$\dot{\mathbf{q}} = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}) \otimes \mathbf{q} \tag{4.63}$$

where $\otimes$ represents quaternion multiplication and

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix}. \tag{4.64}$$

The body angular velocity, $\boldsymbol{\omega}$, expressed in the body frame can be directly obtained from the gyroscope sensor measurements. The full derivative is then written as

$$\begin{aligned}
\dot{\mathbf{q}} &= \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}) \otimes \mathbf{q} && (4.65)\\
&= \frac{1}{2}\begin{bmatrix} 0 & -\boldsymbol{\omega}^T \\ \boldsymbol{\omega} & -\hat{\boldsymbol{\omega}} \end{bmatrix}\mathbf{q} && (4.66)\\
&= \frac{1}{2}\begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix}\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} && (4.67)\\
&= \frac{1}{2}\begin{bmatrix} -\omega_x q_2 - \omega_y q_3 - \omega_z q_4 \\ \omega_x q_1 + \omega_z q_3 - \omega_y q_4 \\ \omega_y q_1 - \omega_z q_2 + \omega_x q_4 \\ \omega_z q_1 + \omega_y q_2 - \omega_x q_3 \end{bmatrix}. && (4.68)
\end{aligned}$$

Notice that this is just a rearranged form from [53] since the quaternion has been defined with the scalar as the first entry instead of the last.

The estimate of the quaternion is then obtained using forward Euler integration in the discrete time where the sampling time, $dt$, is that of the control loop

$$\mathbf{q}(k) = \mathbf{q}(k-1) + \dot{\mathbf{q}}dt. \tag{4.69}$$

To convert from the quaternion estimate to a rotation matrix we can use the following formula

$$\mathbf{R}_b^O = \begin{bmatrix} q_1^2+q_2^2-q_3^2-q_4^2 & 2q_2q_3-2q_1q_4 & 2q_2q_4+2q_1q_3 \\ 2q_2q_3+2q_1q_4 & q_1^2-q_2^2+q_3^2-q_4^2 & 2q_3q_4-2q_1q_2 \\ 2q_2q_4-2q_1q_3 & 2q_3q_4+2q_1q_2 & q_1^2-q_2^2-q_3^2+q_4^2 \end{bmatrix}. \tag{4.70}$$

Problems associated with using only gyroscopes for estimation are mitigated by our experimental techniques: bias is measured and accounted for by initializing the robot at rest for every trial, and since the trials last less than one second, drift becomes negligible.

When the estimation algorithm starts, the initial quaternion is set to

$$\mathbf{q} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{4.71}$$

which is equivalent to setting $\mathbf{R}_c^O = \mathbf{I}_{3\times 3}$, enforcing that any further rotations are with respect to the initial orientation.

## 4.7 EXPERIMENTAL RESULTS

For each trial, the orientation was first initialized at the desired final orientation and then rotated in the initial falling orientation. The robot was then released from rest with the intent to impose no initial angular momentum. Through the onboard accelerometers, the robot is able to detect free-fall and initiate the control scheme. High speed video was recorded at 500 $Hz$ and the IMU data was wirelessly streamed for data collection.

To verify that the combined LS and P-method controller in a simplified setting the first two tests were performed in the isolated pitch and yaw axes respectively pitch or yaw axis respectively. Figure 4.10 is a time framed sequence of pitch correction where the desired body orientation is horizontal and point towards the right. Figure 4.11 is time framed sequence of yaw correction where the desired body orientation is pointing towards the reader.

We then performed the roll experiment that was explored in simulation with the robot. The robot was oriented 90° in roll from the desired orientation and the tail was initially aligned with the body. As shown in Fig. 4.12, the robot pitches the body to move the tail away from the null space of $\mathbf{F}^{\dagger}$ and then proceeds to roll the body while still minimizing error in pitch and yaw. It eventually is able to drive the body near the desired orientation as shown in the last frame.

Finally, the robot's ability to reorient in all three axes is tested. Figure 4.13 is a representative trial showing the robot at the initial orientation in the left most frame. Throughout the time frame sequence it is able to rapidly reorient to the desired orientation in the right most frame. Figure 4.14 is a plot of the error angle for six trials. All the trials converge to within 20° of the desired orientation in approximately 150 $(ms)$. The steady state error maybe attributed to many factors. Some of the factors may be the simplified model that was used in the controller synthesis, a lack of integrator action in the low level feedback controller, or even friction in the inexpensive motors, which completely resist motion for small input signals. For precision motion control this error may be too large, but for rapid paced terrestrial robots, a 20° steady state error may be acceptable.

Figure 4.10: Sequence of frames during a pure pitch correction. The desired orientation is horizontal and pointing towards the right.

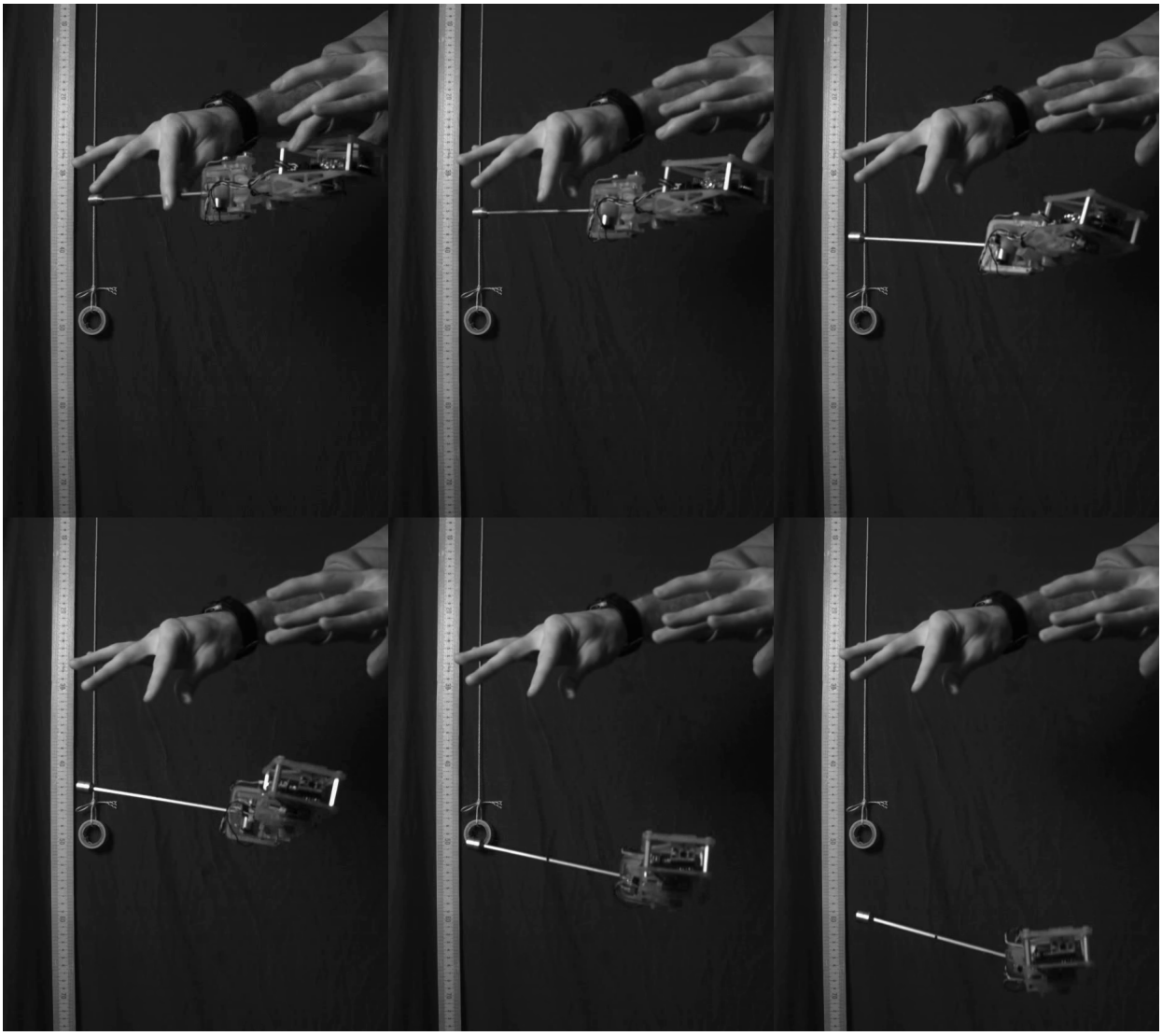


Figure 4.11: Sequence of frames during a pure yaw correction. The desired orientation is pointing towards the reader.

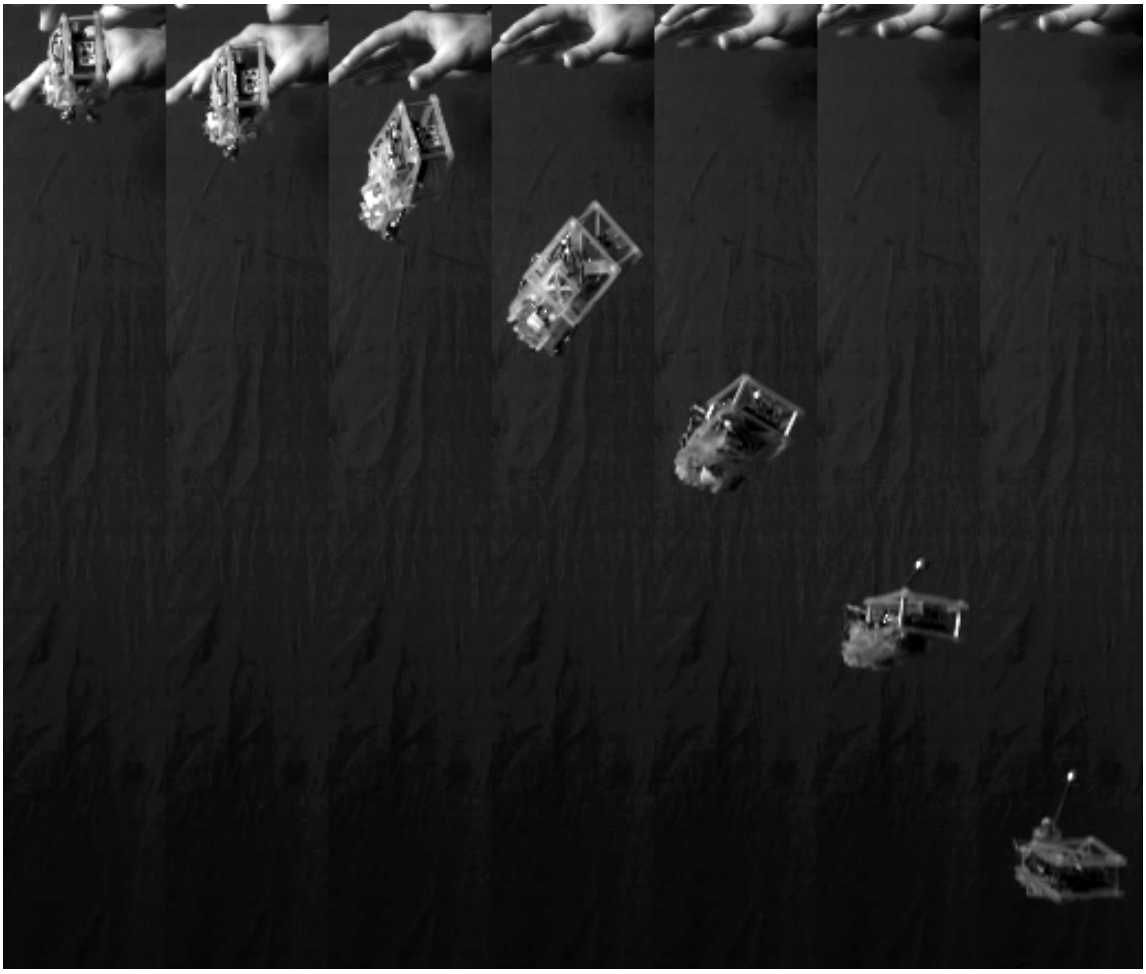


Figure 4.12: Sequence of frames during a roll correction with tail initially aligned with the body. This would be impossible without the perpendicular method. The desired orientation is pointing towards the reader.

Figure 4.13: Sequence of frames during a correction in pitch, yaw, and roll. The desired orientation is pointing towards the reader.

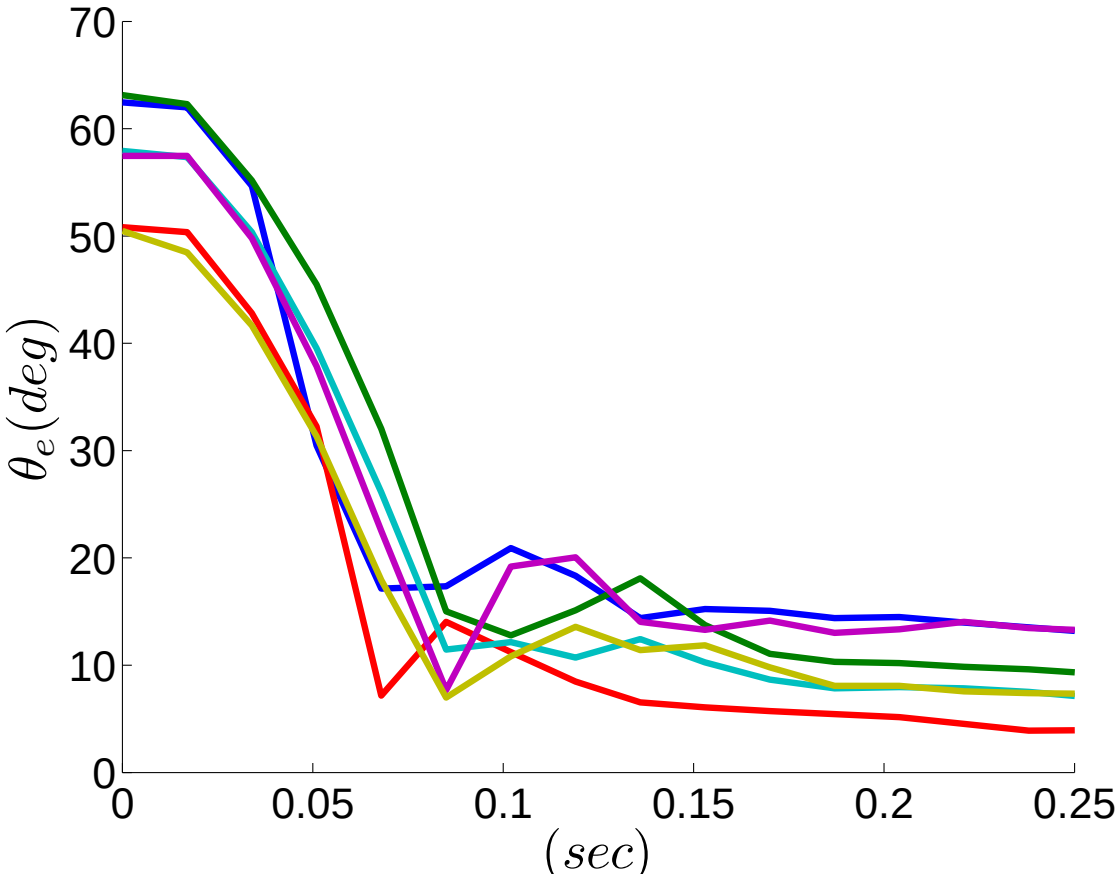



Figure 4.14: Plot of error angle as a function of time for six trials.

## 4.8 SUMMARY

The main contribution of this chapter is to provide a tractable feedback control scheme for orientation control in SO(3) with a two-DOF tailed robot. The benefit of this approach over other proposed methods is its lightweight implementation, which we have demonstrated on a relatively slow and inexpensive microcontroller. While the controller is based on approximate inversion of the total angular momentum constraint using a vastly simplified model, it has demonstrated impressive ability to reorient the robot from arbitrary initial conditions, with low final error.

Unfortunately, we currently lack a tool for analyzing stability and convergence properties of this controller. The field of geometric mechanics [54] may provide some insight into our method's subtleties. In the meanwhile, it has proven remarkably robust in implementation on our testbed, which certainly violates the assumptions of $l_b = 0$ and a perfect point-mass tail.

The nonlinear controller has two major drawbacks as it stands. First, it has no mechanism to avoid joint limits or other undesirable body shapes. For example, the Least Squares inverse requires $\mathbf{F}$ to be full column rank or rank 2. From (4.28) a $\cos(\phi_2)$ appears in every entry of the first column. Therefore, it is clear that $\mathbf{F}^\dagger$ contains a singularity when $\phi_2 = n\pi/2\ rad,\ n \in 2\mathbb{Z}+1$. This corresponds to a physical singularity when the tail is perpendicular to the body in yaw. Second, the nonlinear controller is still sensitive to non-zero angular momentum. We hope to address these issues with modifications to the present structure.

Beyond these challenges, we see a host of potential benefits to robotics, ultimately leading to the next generation of highly mobile search and rescue robots.

# Chapter 5

# CONCLUSION and FUTURE DIRECTIONS

This dissertation has examined the benefits of adding an active tail to improve the performance of mobile robots. We have shown that a robot with an appendage designed for both inertial and contact torques could enable rapid air-reorientation, improve fall survivability, and increase mobility and stability over difficult terrain.

Our 1DOF robot can reorient 90° in one body length of vertical fall, survive perturbations that would flip a tailless robot, and effect rapid angle changes to smooth transitions between surfaces. Our bio-inspired tail-like design has several advantages over alternative technologies such as reaction wheels: low weight, high rotational inertia, and the capability to apply external torques via substrate contact. We found that contact torques and inertial torques can be modulated with the same simple controller to produce predictable, directed body rotation.

A new approach for a single DOF attitude estimation was proposed using a TVCF method. Significant improvements were achieved over the conventional CF method while still maintaining ease of implementation. The effectiveness has been verified with experimental results. This work may help a general set of applications that require effective yet tractable attitude estimates in pitch and roll.

Observations in lizards indicate that by adding an actuated yaw DOF to the tail, we can effect body rotations in not only yaw, but roll as well. This orientation control in SO(3) has been accomplished through a tractable nonlinear feedback control scheme with an active two-DOF tailed robot. The benefit of this approach over other proposed methods is its lightweight implementation, which we have demonstrated on a relatively slow and inexpensive microcontroller. While the controller is based on approximate inversion of the total angular momentum constraint using a vastly simplified model, it has demonstrated impressive ability to reorient the robot from arbitrary initial postures, with low final error. This is possible even with the violation of the assumptions that $l_b = 0$ and that the tail is a perfect point-mass.

### 5.0.1 Future Directions

To improve the maneuverability and reduce the problem of limited tail stroke it will be valuable to develop a tail that can dynamically alter its flexibility and MOI. A more flexible tail can perform alternative motions when the tail stroke would normally be saturated in a rigid tail design or when singularities are reached in a lower DOF design. In the case of landing when the tail is underneath the robot, the tail can also avoid damaging itself and the robot if it is flexible.

Unfortunately, we currently lack a tool for analyzing the stability and convergence properties of this controller. A rigorous stability analysis perhaps involving Lyapunav theory and/or differential geometry should prove useful for not only analysis, but also synthesis of more capable controllers. Robustness guarantees to model uncertainty, noise, and environmental disturbances should also prove useful to bound the design of future robots. The field of geometric mechanics [54] may provide some insight into our method's subtleties.

The nonlinear controller has two major drawbacks as it stands. First, it has no mechanism to avoid joint limits or other undesirable body shapes. For example, the Least Squares inverse requires $\mathbf{F}$ to be full column rank or rank 2. From (4.28) a $\cos(\phi_2)$ appears in every entry of the first column. Therefore, it is clear that $\mathbf{F}^{\dagger}$ contains a singularity when $\phi_2 = n\pi/2\ rad,\ n \in 2\mathbb{Z}+1$. This corresponds to a physical singularity when the tail is perpendicular to the body in yaw. Second, the nonlinear controller is still sensitive to non-zero angular momentum. With the improvement and decreasing cost of IMU sensors it will be valuable to add multiple IMU's throughout the entire system, namely the tail, to obtain a better estimate of the tail's angular velocity. This may lead to an overall better estimate of the systems angular momentum, which can lead to controllers that can compensate for nonzero angular momentum settings. We hope to address these issues with modifications to the present structure.

As with biological systems, learning has a great potential and supplementing the existing controller with well known reinforcement learning and artificial intelligence methods should greatly expand the performance of future tailed robots [58, 59].

Beyond these challenges, we see a host of potential benefits to robotics. Our hope is that the addition of an active tail to the next generation of dynamic legged robots could increase their ability to perform in environments too inaccessible or dangerous for humans, such as in search-and-rescue missions.

# Bibliography


[1] J. Casper and R. R. Murphy. "Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center". In: *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 33.3 (2003), pp. 367–385.

[2] P. Birkmeyer, K. Peterson, and R. S. Fearing. "DASH: A dynamic 16g hexapedal robot". In: *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. 2009, pp. 2683–2689.

[3] Uluc Saranli, Martin Buehler, and Daniel E. Koditschek. "RHex: A Simple and Highly Mobile Hexapod Robot". In: *The International Journal of Robotics Research* 20.7 (2001), pp. 616–631.

[4] M. J. Spenko et al. "Biologically inspired climbing with a hexapedal robot". In: *Journal of Field Robotics* 25.4-5 (2008), pp. 223–242.

[5] C. Menon, M. Murphy, and M. Sitti. "Gecko Inspired Surface Climbing Robots". In: *Robotics and Biomimetics, 2004. ROBIO 2004. IEEE International Conference on*. 2004, pp. 431–436.

[6] Alan T. Asbeck et al. "Scaling Hard Vertical Surfaces with Compliant Microspine Arrays". In: *The International Journal of Robotics Research* 25.12 (2006), pp. 1165–1179.

[7] A. Jusufi et al. "Righting and turning in mid-air using appendage inertia: reptile tails, analytical models and bio-inspired robots". In: *Bioinspiration & Biomimetics* 5.4 (2010), p. 045001.

[8] Thomas Libby et al. "Tail-assisted pitch control in lizards, robots and dinosaurs". In: *Nature* 481.7380 (). 10.1038/nature10710, pp. 181–184.

[9] T. R. Kane and M. P. Scher. "A dynamical explanation of the falling cat phenomenon". In: *International Journal of Solids and Structures* 5.7 (1969). doi: DOI: 10.1016/0020-7683(69)90086-9, pp. 663–666.

[10] Richard Montgomery. "Guage Theory of the Falling Cat". In: *M.J. Enos(Ed.), Dynamics and Control of Mechanical Systems, American Mathematical Society* (1993), pp. 193–218.

[11] E. Papadopoulos, I. Fragkos, and I. Tortopidis. "On Robot Gymnastics Planning with Non-zero Angular Momentum". In: *Robotics and Automation, 2007 IEEE International Conference on*. 2007, pp. 1443–1448.

[12] Xin Xin, T. Mita, and M. Kaneda. “The posture control of a two-link free flying acrobot with initial angular momentum”. In: *Automatic Control, IEEE Transactions on* 49.7 (2004), pp. 1201–1206.

[13] Ardian Jusufi et al. “Active tails enhance arboreal acrobatics in geckos”. In: *Proceedings of the National Academy of Sciences* 105.11 (2008), pp. 4215–4219.

[14] E. Papadopoulos and S. Dubowsky. “On the nature of control algorithms for free-floating space manipulators”. In: *Robotics and Automation, IEEE Transactions on* 7.6 (1991), pp. 750 –758.

[15] K. Yoshida and Y. Umetani. “Control of space free-flying robot”. In: *Decision and Control, 1990., Proceedings of the 29th IEEE Conference on*. 1990, 97–102 vol.1.

[16] Chris Fernandes, Leonid Gurvits, and Li Zexiang. “Attitude Control of a Space Platform Manipulator System Using Internal Motion”. In: *The International Journal of Robotics Research* 13.4 (1994), pp. 289–304.

[17] Richard W. Longman, Robert E. Lindbergt, and Michael F. Zedd. “Satellite-Mounted Robot Manipulators New Kinematics and Reaction Moment Compensation”. In: *The International Journal of Robotics Research* 6.3 (1987), pp. 87–103.

[18] F. Saito, T. Fukuda, and F. Arai. “Swing and locomotion control for two-link brachiation robot”. In: *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*. 1993, 719–724 vol.2.

[19] Z. Li and R. Montgomery. “Dynamics and optimal control of a legged robot in flight phase”. In: *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*. 1990, 1816–1821 vol.3.

[20] T. Rokusho et al. “Landing Control of Acrobat Robot by RHC -Experimental Evaluation”. In: *SICE-ICASE, 2006. International Joint Conference*. 2006, pp. 4808–4813.

[21] K. Yamafuji et al. “Elucidation of Twisting Motion of a Falling Cat and Realization of the CAT-TURN Motion by a Robot”. In: *INTERNATIONAL SYMPOSIUM ON ROBOTICS* 30 (1999), pp. 527–534.

[22] L. Keo and M. Yamakita. “Controlling balancer and steering for bicycle stabilization”. In: *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. 2009, pp. 4541–4546.

[23] M. Yamakita and A. Utano. “Automatic control of bicycles with a balancer”. In: *Advanced Intelligent Mechatronics. Proceedings, 2005 IEEE/ASME International Conference on*. 2005, pp. 1245–1250.

[24] *Murata Boy. http://www.murataboy.com/en/index.html*. 2011.

[25] S. Floyd et al. “A Novel Water Running Robot Inspired by Basilisk Lizards”. In: *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. 2006, pp. 5430–5436.

[26] A. J. Baerveldt and R. Klang. “A low-cost and low-weight attitude estimation system for an autonomous helicopter”. In: *Intelligent Engineering Systems, 1997. INES ’97. Proceedings., 1997 IEEE International Conference on*. 1997, pp. 391–395.

[27] A. El Hadri and A. Benallegue. “Attitude estimation with gyros-bias compensation using low-cost sensors”. In: *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*. 2009, pp. 8077–8082.

[28] Robert Grover Brown. *Introduction to random signal analysis and Kalman filtering*. New York: Wiley, 1983.

[29] M. Euston et al. “A complementary filter for attitude estimation of a fixed-wing UAV”. In: *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. 2008, pp. 340–345.

[30] S. Lohmeier et al. “Sensor system and trajectory control of a biped robot”. In: *Advanced Motion Control, 2004. AMC ’04. The 8th IEEE International Workshop on*. 2004, pp. 393–398.

[31] W. T. Higgins. “A Comparison of Complementary and Kalman Filtering”. In: *Aerospace and Electronic Systems, IEEE Transactions on* AES-11.3 (1975), pp. 321–325.

[32] Yizhou Wang et al. “”Three Dimensional Attitude Estimation Via the TRIAD Algorithm and a Time-Varying Complementary Filter””. In: *DSCC 2012’*. 2012.

[33] Yang Yi-Ling, P. C. P. Chao, and Sung Cheng-Kuo. “Landing posture control via trajectory planning for a generalized twin-body system”. In: *Autonomous Robots and Agents, 2009. ICARA 2009. 4th International Conference on*. 2009, pp. 97–102.

[34] T. W. Mather and M. Yim. “Modular configuration design for a controlled fall”. In: *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. 2009, pp. 5905–5910.

[35] C. Fernandes, L. Gurvits, and Li Zexiang. “Near-optimal nonholonomic motion planning for a system of coupled rigid bodies”. In: *Automatic Control, IEEE Transactions on* 39.3 (1994), pp. 450–463.

[36] T. Kai. “A model predictive control approach to attitude stabilization and trajectory tracking control of a 3D universal joint space robot with an initial angular momentum”. In: *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pp. 3547–3552.

[37] Aaron Johnson et al. “Tail Assisted Dynamic Self Righting”. In: *Proceedings of the International Conference on Climbing and Walking Robots, 2012*.

[38] Evan Chang-Siu et al. “A lizard-inspired active tail enables rapid maneuvers and dynamic stabilization in a terrestrial robot”. In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 1887–1894.

[39] T. Kohlsdorf, T. Garland, and C. A. Navas. "Limb and tail lengths in relation to substrate usage in Tropidurus lizards". In: *Journal of Morphology* 248.2 (2001), pp. 151–164.

[40] James J. Dowling, Jennifer L. Durkin, and David M. Andrews. "The uncertainty of the pendulum method for the determination of the moment of inertia". In: *Medical Engineering & Physics* 28.8 (2006). doi: DOI: 10.1016/j.medengphy.2005.11.007, pp. 837–841.

[41] T. Libby, E. Chang-Siu, and R J. Full. "Dynamic stabilization of rapid transitions in a lizard and a robot with an active tail". In: *In Prep* (2011).

[42] E. Chang-Siu, K. Kong, and M. Tomizuka. "Time-Varying Complementary Filtering for Attitude Estimation". In: *In Prep* (2011).

[43] M. Cameron Jonathan and C. Arkin Ronald. "Survival of falling robots". In: ed. by J. Wolfe William and H. Chun Wendell. Vol. 1613. Mobile Robots VI 1. SPIE, 1992, pp. 91–102.

[44] M. Kovac et al. "A miniature jumping robot with self-recovery capabilities". In: *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. 2009, pp. 583–588.

[45] Soo Jeon, Masayoshi Tomizuka, and Tetsuaki Katou. "Kinematic Kalman Filter (KKF) for Robot End-Effector Sensing". In: *Journal of Dynamic Systems, Measurement, and Control* 131.2 (2009), pp. 021010–8.

[46] Soo Jeon and Masayoshi Tomizuka. "Benefits of acceleration measurement in velocity estimation and motion control". In: *Control Engineering Practice* 15.3 (2007). doi: DOI: 10.1016/j.conengprac.2005.10.004, pp. 325–332.

[47] Lee Dong-Jun and M. Tomizuka. "Multirate optimal state estimation with sensor fusion". In: *American Control Conference, 2003. Proceedings of the 2003*. Vol. 4. 2003, 2887–2892 vol.4.

[48] Gene F. Franklin and J. David Powell. *Digital control of dynamic systems*. Reading, Mass.: Addison-Wesley Pub. Co., 1980.

[49] Yoan D. Landau, Gianluca Zito, and SpringerLink. *Digital control systems design, identification and implementation*. (Online service).

[50] Patrick F. Dunn. *Measurement and data analysis for engineering and science*. Boston: McGraw-Hill Higher Education, 2005.

[51] Evan Chang-Siu et al. "A nonlinear feedback controller for aerial self-righting by a tailed robot". In: *Robotics and Automation, 2013. ICRA '13. IEEE International Conference on*. 2013.

[52] M. W. Spong. *Robot Dynamics And Control*. Wiley India Pvt. Limited, 2008.

[53] J. L. Crassidis and J. L. Junkins. *Optimal Estimation of Dynamic Systems*. Chapman and Hall/CRC, 2004.

[54] Ross L. Hatton and Howie Choset. “Geometric motion planning: The local connection, Stokes theorem, and the importance of coordinate choice”. In: *The International Journal of Robotics Research* 30.8 (), pp. 988–1014.

[55] B. Siciliano, L. Sciavicco, and L. Villani. *Robotics: modelling, planning and control*. Springer Verlag, 2009.

[56] S. Hirose and H. Yamada. “Snake-like robots [Tutorial]”. In: *Robotics & Automation Magazine, IEEE* 16.1 (2009), pp. 88–98.

[57] A. Sabatini. “Quaternion-based strap-down integration method for applications of inertial sensing to gait analysis”. In: *Medical and Biological Engineering and Computing* 43.1 (2005), pp. 94–101.

[58] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. Bradford Book, 1998.

[59] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotic*. Mit Press, 2005.

[60] Thomas Mark. Libby. “Tail assisted pitch control in lizard, robots and dinosaurs”. Masters Thesis. UC Berkeley, 2012.

[61] D. T. Greenwood. *Advanced Dynamics*. Cambridge University Press, 2006.

# Appendix A

# Equations of Motion for 1DOF Tailbot

Here we derive the equations of motion of the single degree of freedom in state space form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})u \tag{A.1}$$

where the states, $\mathbf{x} = [\theta_b, \dot{\theta}_b, \theta_t, \dot{\theta}_t]^T$, consist of the angular position and velocity of the links and the control input, $u = \tau$, is the relative torque.

We assume that the only external force acting on the two-link system is gravity and therefore the COM of the system is accelerating down at one $g$ with respect to the fixed world frame. To conveniently remove the effect of gravity in the equations of motion and without loss of generality, we select the center of mass (COM) of this free-falling system as the origin of the Cartesian coordinate system, $O$, as shown in Fig. 4.3 where $\boldsymbol{\rho}_i$ is the vector from the $O$ to the COM of each link, and $m_i$ is the mass of each link. This results in the relationship

$$\boldsymbol{\rho}_t = -\frac{m_b}{m_t}\boldsymbol{\rho}_b. \tag{A.2}$$

To obtain the equations of motion we must analyze each link separately. Thus, the individual angular momenta for each link, $\mathbf{H}_{io}$, about $O$ can then be defined as

$$\mathbf{H}_{io} = I_i\boldsymbol{\omega}_i + m_i\boldsymbol{\rho}_i \times \dot{\boldsymbol{\rho}}_i \tag{A.3}$$

where $\boldsymbol{\omega}_i = \dot{\theta}_i\mathbf{E}_3$ is the angular velocity of each link and $I_i$ is the moment of inertia (MOI) about the COM of each link. Valid alternative methods to acquire the EOM can take the angular momentum about the COM of each link [60] or even about the pin joint.

Since $O$ is a noninertial reference frame, the derivative of each link's angular momenta can then be equated to not only the total moment acting on the link, but also the pseudo moment, $-\boldsymbol{\rho}_i \times m_i\ddot{\mathbf{r}}_{COM}$ [61], where $\mathbf{r}_{COM}$ is the vector from the world fixed frame to $O$. From Fig. A.2, the total moment on link-one is due to the relative torque from the motor and the torque due to the pin force at the revolute joint and the gravity force. This is fully written as

$$\dot{\mathbf{H}}_{bo} = I_b\dot{\boldsymbol{\omega}}_b + m_b\dot{\boldsymbol{\rho}}_b \times \dot{\boldsymbol{\rho}}_b + m_b\boldsymbol{\rho}_b \times \ddot{\boldsymbol{\rho}}_b \tag{A.4}$$

$$= \boldsymbol{\tau} + (\boldsymbol{\rho}_b - \boldsymbol{\pi}_b) \times \mathbf{F}_{pin} + \boldsymbol{\rho}_b \times \mathbf{F}_{gravity} - \boldsymbol{\rho}_b \times m_b\ddot{\mathbf{r}}_{COM} \tag{A.5}$$

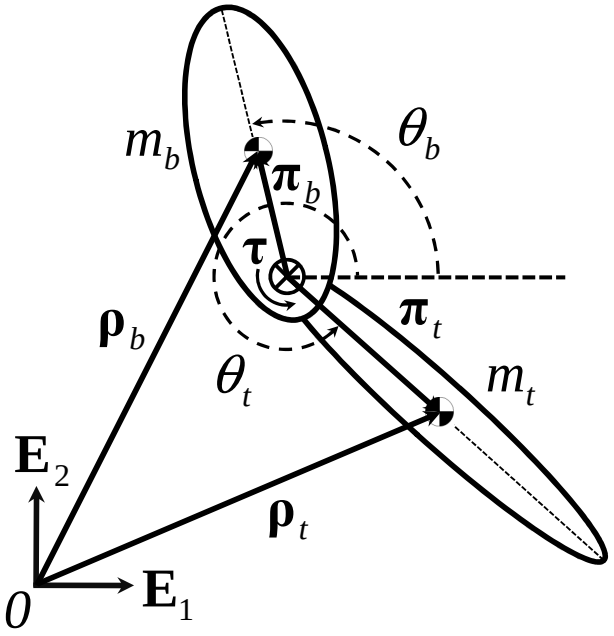



Figure A.1: Diagram of two link planar model for aerial reorientation. The motor torque, $\boldsymbol{\tau} = \tau \mathbf{E}_3$ is positive relative to link-one.

where $\boldsymbol{\pi}_b = l_b \mathbf{b}_1$ and $\boldsymbol{\pi}_t = l_t \mathbf{t}_1$ are the vectors from the pivot to the COM of each link.

With no other external forces on the individual links other than gravity and the pin force and since $\ddot{\mathbf{r}}_{COM} = -g\mathbf{E}_2$ we can equate the sum of forces to the total acceleration in link-one as

$$\mathbf{F}_{pin} + \mathbf{F}_{gravity} = m_b(\ddot{\boldsymbol{\rho}}_b + \ddot{\mathbf{r}}_{COM}) = m_1(\ddot{\boldsymbol{\rho}}_b - g\mathbf{E}_2). \tag{A.6}$$

With $\mathbf{F}_{gravity} = -m_b g \mathbf{E}_2$ and the COM constrain (A.2) we get

$$\mathbf{F}_{pin} = m_b \ddot{\boldsymbol{\rho}}_b = -m_t \ddot{\boldsymbol{\rho}}_t. \tag{A.7}$$

Substituting (A.7) into (A.5), the gravity terms cancel, and with $\dot{\boldsymbol{\rho}}_b \times \dot{\boldsymbol{\rho}}_b = \mathbf{0}$ results in

$$I_b \dot{\boldsymbol{\omega}}_b + m_b \boldsymbol{\rho}_b \times \ddot{\boldsymbol{\rho}}_b = \boldsymbol{\tau} + (\boldsymbol{\rho}_b - \boldsymbol{\pi}_b) \times m_b \ddot{\boldsymbol{\rho}}_b, \tag{A.8}$$

which simplifies to the first equation of motion

$$I_b \dot{\boldsymbol{\omega}}_b + \boldsymbol{\pi}_b \times m_b \ddot{\boldsymbol{\rho}}_b = \boldsymbol{\tau}. \tag{A.9}$$

For link-two there are sign changes on $\boldsymbol{\tau}$ and $\mathbf{F}_{pin}$ as shown in Fig. A.3 and the derivative of its angular momentum is

$$\begin{aligned} \dot{\mathbf{H}}_{to} &= I_t \dot{\boldsymbol{\omega}}_t + m_t \dot{\boldsymbol{\rho}}_t \times \dot{\boldsymbol{\rho}}_t + m_t \boldsymbol{\rho}_t \times \ddot{\boldsymbol{\rho}}_2 & \text{(A.10)} \\ &= -\boldsymbol{\tau} + (\boldsymbol{\rho}_t - \boldsymbol{\pi}_t) \times (-\mathbf{F}_{pin}) + \boldsymbol{\rho}_t \times \mathbf{F}_{gravity} - \boldsymbol{\rho}_t \times m_t \ddot{\mathbf{r}}_{COM}. & \text{(A.11)} \end{aligned}$$

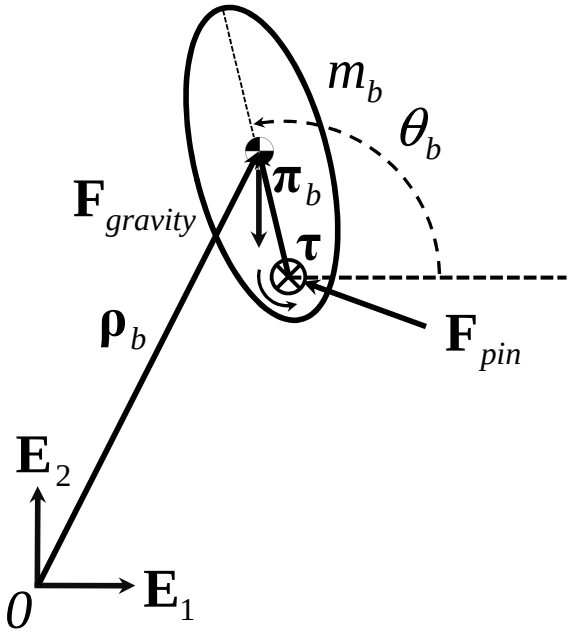



Figure A.2: Link-one isolated.

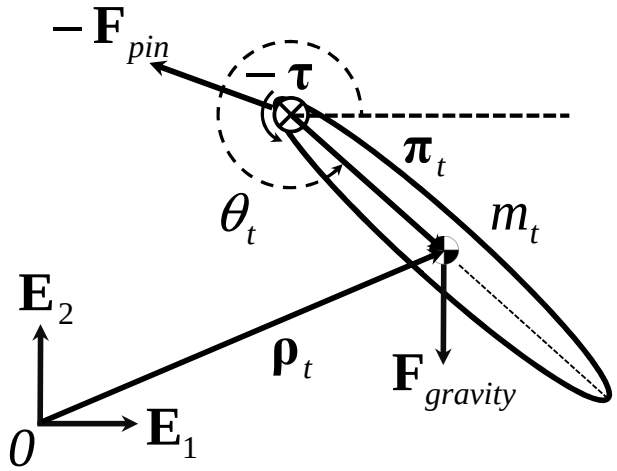



Figure A.3: Link-two isolated.

Following the same process as link-one results in

$$I_t \dot{\boldsymbol{\omega}}_t + \boldsymbol{\pi}_t \times m_t \ddot{\boldsymbol{\rho}}_t = -\boldsymbol{\tau}, \tag{A.12}$$

which is the second equation of motion.

The following transformations and identities will be useful for computing the second left hand term in (A.9) and (A.12):

$$\begin{aligned}
\mathbf{b}_1 &= \cos(\theta_b)\mathbf{E}_1 + \sin(\theta_b)\mathbf{E}_2 && \text{(A.13)}\\
\mathbf{b}_2 &= -\sin(\theta_b)\mathbf{E}_1 + \cos(\theta_b)\mathbf{E}_2 && \text{(A.14)}\\
\mathbf{t}_1 &= \cos(\theta_t)\mathbf{E}_1 + \sin(\theta_t)\mathbf{E}_2 && \text{(A.15)}\\
\mathbf{t}_2 &= -\sin(\theta_t)\mathbf{E}_1 + \cos(\theta_t)\mathbf{E}_2 && \text{(A.16)}\\
\mathbf{b}_1 \times \mathbf{t}_1 &= (-\sin(\theta_b)\cos(\theta_t) + \cos(\theta_b)\sin(\theta_t))\mathbf{E}_3 && \text{(A.17)}\\
&= -\sin(\theta_b - \theta_t)\mathbf{E}_3 && \text{(A.18)}\\
\mathbf{b}_1 \times \mathbf{t}_2 &= (\cos(\theta_b)\cos(\theta_t) + \sin(\theta_b)\sin(\theta_t))\mathbf{E}_3 && \text{(A.19)}\\
&= \cos(\theta_b - \theta_t)\mathbf{E}_3 && \text{(A.20)}\\
\mathbf{t}_1 \times \mathbf{b}_1 &= (\sin(\theta_b)\cos(\theta_t) - \cos(\theta_b)\sin(\theta_t))\mathbf{E}_3 && \text{(A.21)}\\
&= \sin(\theta_b - \theta_t)\mathbf{E}_3 && \text{(A.22)}\\
\mathbf{t}_1 \times \mathbf{b}_2 &= (\cos(\theta_b)\cos(\theta_t) + \sin(\theta_b)\sin(\theta_t))\mathbf{E}_3 && \text{(A.23)}\\
&= \cos(\theta_b - \theta_t)\mathbf{E}_3. && \text{(A.24)}
\end{aligned}$$

Summing the vector loop in Fig. 4.3 provides the equation

$$\boldsymbol{\rho}_b - \boldsymbol{\pi}_b + \boldsymbol{\pi}_t - \boldsymbol{\rho}_t = 0. \tag{A.25}$$

With the COM constraint (A.2) we can then solve for $\boldsymbol{\rho}_b$ and $\boldsymbol{\rho}_t$ as

$$\begin{aligned}
\boldsymbol{\rho}_b &= \boldsymbol{\pi}_b - \boldsymbol{\pi}_t - \frac{m_b}{m_t}\boldsymbol{\rho}_b && \text{(A.26)}\\
\boldsymbol{\rho}_b &= \frac{m_t}{m_b + m_t}(\boldsymbol{\pi}_b - \boldsymbol{\pi}_t) && \text{(A.27)}\\
\boldsymbol{\rho}_b &= M_2(\boldsymbol{\pi}_b - \boldsymbol{\pi}_t) && \text{(A.28)}
\end{aligned}$$

where $M_2 = m_t/(m_b + m_t)$ and

$$\boldsymbol{\rho}_t = -M_1(\boldsymbol{\pi}_b - \boldsymbol{\pi}_t) \tag{A.29}$$

where $M_1 = m_b/(m_b + m_t)$. The double derivatives of $\boldsymbol{\rho}_i$ depend on the double derivatives of $\boldsymbol{\pi}_i$, which can be obtained from

$$\boldsymbol{\pi}_b = l_b \mathbf{b}_1 \tag{A.30}$$

$$\dot{\boldsymbol{\pi}}_b = l_b \dot{\theta}_b \mathbf{b}_2 \tag{A.31}$$

$$\ddot{\boldsymbol{\pi}}_b = -l_b\dot{\theta}_b^2\mathbf{b}_1 + l_b\ddot{\theta}_b\mathbf{b}_2 \tag{A.32}$$

and by the same process

$$\ddot{\boldsymbol{\pi}}_t = -l_t\dot{\theta}_t^2\mathbf{t}_1 + l_t\ddot{\theta}_t\mathbf{t}_2. \tag{A.33}$$

For link-one with the above identities (A.18), (A.20), and (A.32) and (A.33) we can solve for the second term in (A.9) as

$$\begin{aligned}
\boldsymbol{\pi}_b \times m_b\ddot{\boldsymbol{\rho}}_b &= l_b\mathbf{b}_1 \times m_bM_2(\ddot{\boldsymbol{\pi}}_b - \ddot{\boldsymbol{\pi}}_t) && \text{(A.34)}\\
&= l_bm_bM_2\mathbf{b}_1 \times (-l_b\dot{\theta}_b^2\mathbf{b}_1 + l_b\ddot{\theta}_b\mathbf{b}_2 + l_t\dot{\theta}_t^2\mathbf{t}_1 - l_t\ddot{\theta}_t\mathbf{t}_2) && \text{(A.35)}\\
&= l_bm_bM_2(l_b\ddot{\theta}_b\mathbf{b}_1 \times \mathbf{b}_2 + l_t\dot{\theta}_t^2\mathbf{b}_1 \times \mathbf{t}_1 - l_t\ddot{\theta}_t\mathbf{b}_1 \times \mathbf{t}_2) && \text{(A.36)}\\
&= l_bm_bM_2(l_b\ddot{\theta}_b - l_t\dot{\theta}_t^2\sin(\theta_b-\theta_t) - l_t\ddot{\theta}_t\cos(\theta_b-\theta_t))\mathbf{E}_3. && \text{(A.37)}
\end{aligned}$$

For link-two with the above identities (A.22), (A.24), and (A.32) and (A.33) we can solve for the second term in (A.12) as

$$\begin{aligned}
\boldsymbol{\pi}_t \times m_t\ddot{\boldsymbol{\rho}}_t &= l_t\mathbf{t}_1 \times (-m_tM_1(\ddot{\boldsymbol{\pi}}_b - \ddot{\boldsymbol{\pi}}_t)) && \text{(A.38)}\\
&= -l_tm_tM_1\mathbf{t}_1 \times (-l_b\dot{\theta}_b^2\mathbf{b}_1 + l_b\ddot{\theta}_b\mathbf{b}_2 + l_t\dot{\theta}_t^2\mathbf{t}_1 - l_t\ddot{\theta}_t\mathbf{t}_2) && \text{(A.39)}\\
&= -l_tm_tM_1(-l_b\dot{\theta}_b^2\mathbf{t}_1 \times \mathbf{b}_1 + l_b\ddot{\theta}_b\mathbf{t}_1 \times \mathbf{b}_2 - l_t\ddot{\theta}_t\mathbf{t}_1 \times \mathbf{t}_2) && \text{(A.40)}\\
&= -l_tm_tM_1(-l_b\dot{\theta}_b^2\sin(\theta_b-\theta_t) + l_b\ddot{\theta}_b\cos(\theta_b-\theta_t) - l_t\ddot{\theta}_t)\mathbf{E}_3. && \text{(A.41)}
\end{aligned}$$

Now (A.9) and (A.12) can be written completely as

$$I_b\ddot{\theta}_b + l_bm_bM_2(l_b\ddot{\theta}_b - l_t\dot{\theta}_t^2\sin(\theta_b-\theta_t) - l_t\ddot{\theta}_t\cos(\theta_b-\theta_t)) = \tau \tag{A.42}$$

$$I_t\ddot{\theta}_t - l_tm_tM_1(-l_b\dot{\theta}_b^2\sin(\theta_b-\theta_t) + l_b\ddot{\theta}_b\cos(\theta_b-\theta_t) - l_t\ddot{\theta}_t) = -\tau. \tag{A.43}$$

To obtain the state space form we can first write the equations in robot manipulator form as

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{B}\tau \tag{A.44}$$

where $\mathbf{q} = [\theta_b, \theta_t]^T$. Solving for $\ddot{\mathbf{q}}$ results in

$$\ddot{\mathbf{q}} = \mathbf{M}(\mathbf{q})^{-1}(-\mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{B}\tau) \tag{A.45}$$

where by inspection and the inverse property of a 2x2 matrix

$$\mathbf{M} = \begin{bmatrix} I_b + l_b^2m_bM_1 & -l_bl_tm_bM_2\cos(\theta_b-\theta_t) \\ -l_bl_tm_tM_1\cos(\theta_b-\theta_t) & I_t + l_t^2m_tM_1 \end{bmatrix}, \tag{A.46}$$

$$\mathbf{M}^{-1} = \frac{1}{\Delta}\begin{bmatrix} I_t + l_t^2m_tM_1 & l_bl_tm_bM_2\cos(\theta_b-\theta_t) \\ l_bl_tm_tM_1\cos(\theta_b-\theta_t) & I_b + l_b^2m_bM_1 \end{bmatrix}, \tag{A.47}$$

$$\Delta = (l_b^2m_bm_t + I_b(m_b+m_t))(l_t^2m_bm_t + I_t(m_b+m_t)) - l_b^2l_t^2m_b^2m_t^2\cos(\theta_b-\theta_t)^2, \tag{A.48}$$

$$\mathbf{C} = \begin{bmatrix} -l_b m_b M_2 l_t \sin(\theta_b - \theta_t)\dot{\theta}_t^2 \\ l_t m_t M_1 l_b \sin(\theta_b - \theta_t)\dot{\theta}_b^2 \end{bmatrix}, \tag{A.49}$$

and

$$\mathbf{B} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}. \tag{A.50}$$

The rows of $\ddot{\mathbf{q}}$ allow us to write the full state space form where the details of $\mathbf{f}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ are

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} \dot{\theta}_b \\ \frac{-l_b^2 l_t^2 m_b^2 m_t^2 \sin(\theta_b-\theta_t)\cos(\theta_b-\theta_t)\dot{\theta}_b^2 + (l_t^2 m_b m_t + I_t(m_b+m_t))l_b l_t m_b m_t \sin(\theta_b-\theta_t)\dot{\theta}_t^2}{\Delta} \\ \dot{\theta}_t \\ \frac{l_b^2 l_t^2 m_b^2 m_t^2 \sin(\theta_b-\theta_t)\cos(\theta_b-\theta_t)\dot{\theta}_t^2 - (l_b^2 m_b m_t + I_b(m_b+m_t))l_b l_t m_b m_t \sin(\theta_b-\theta_t)\dot{\theta}_b^2}{\Delta} \end{bmatrix} \tag{A.51}$$

$$\mathbf{g}(\mathbf{x}) = \begin{bmatrix} 0 \\ \frac{(l_t^2 m_b m_t + I_t(m_b+m_t) - l_b l_t m_b m_t \cos(\theta_b-\theta_t))(m_b+m_t)}{\Delta} \\ 0 \\ \frac{(-l_b^2 m_b m_t - I_b(m_b+m_t) + l_b l_t m_b m_t \cos(\theta_b-\theta_t))(m_b+m_t)}{\Delta} \end{bmatrix}. \tag{A.52}$$

# Appendix B

# 1DOF Tailbot Design Details

Figure B.1 and B.2 show the schematic and board layout for the electronics in the 1DOF *Tailbot*. The components of the board include an 8Mhz Arduino micro processor, a 2 axis gyroscope, 1 axis accelerometer, LED indicator lights, battery connections, and wireless capability.

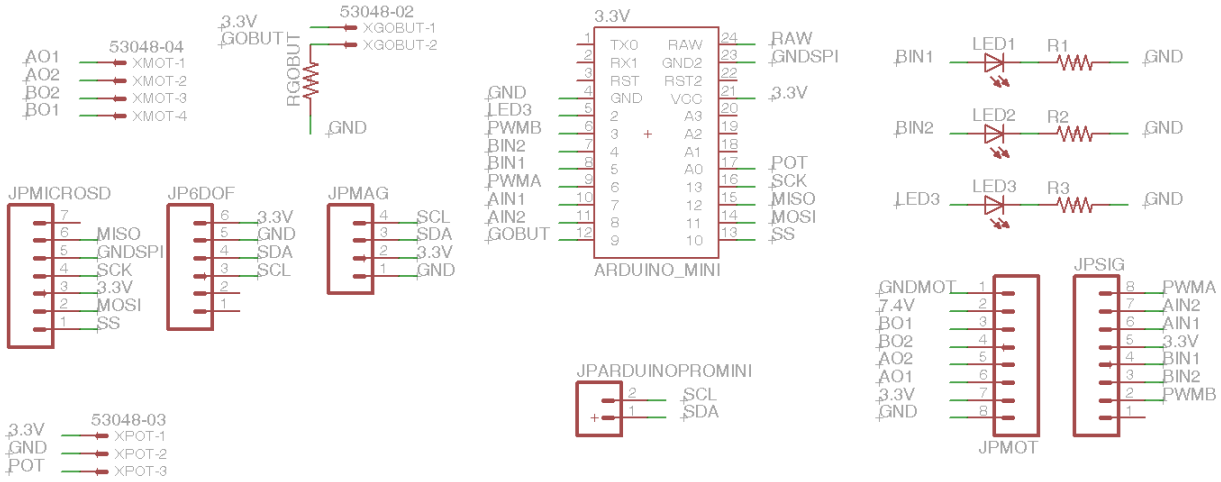



Figure B.1: 1DOF circuit board schematic.

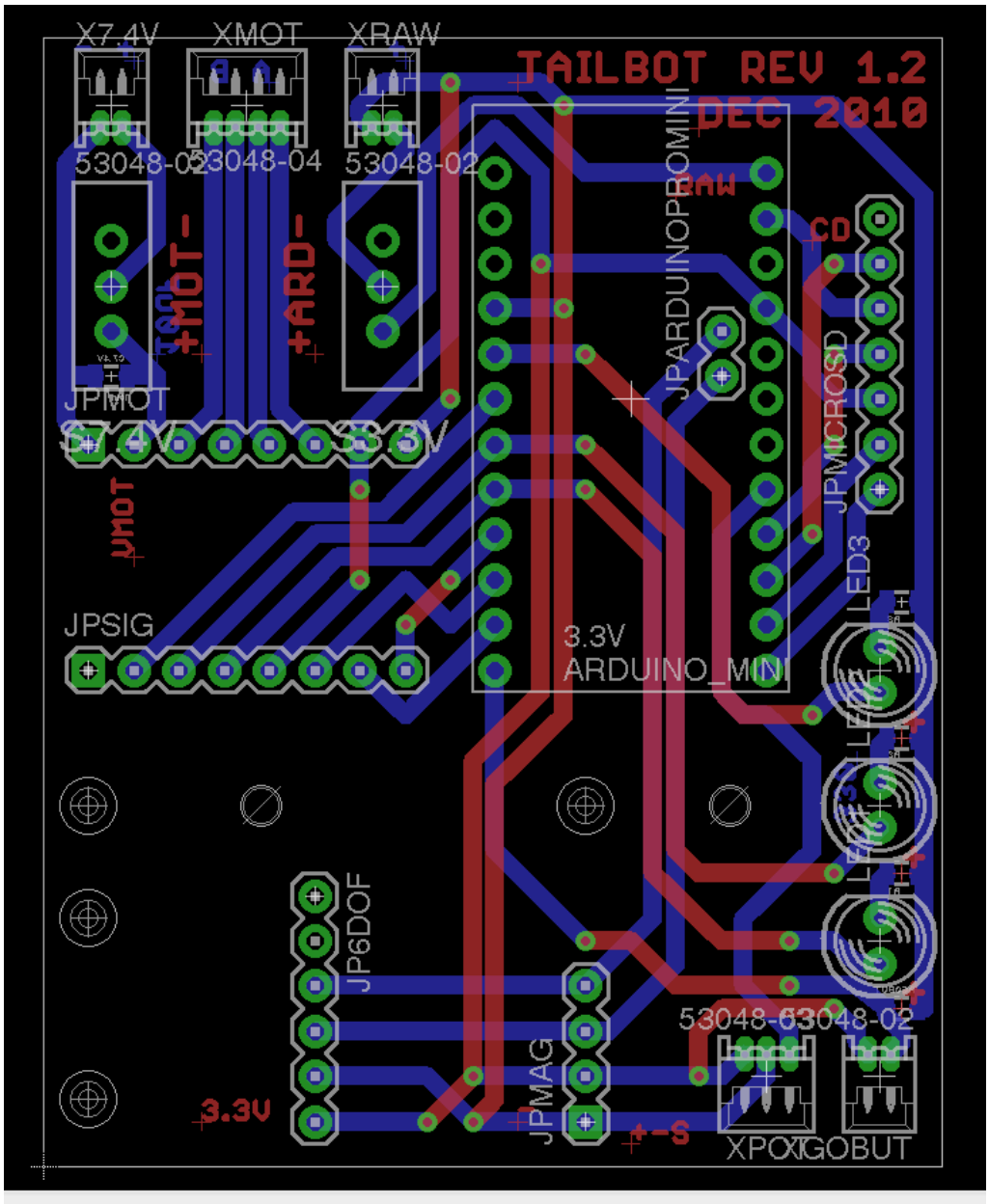



Figure B.2: 1DOF circuit board.

# Appendix C

# Equations of Motion for 2DOF Tailbot

Here we derive the equations of motion for the two-link system in robotic mass manipulator form $\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} = \boldsymbol{\tau}$, where $\boldsymbol{\tau}$ is the vector of input torques, $\mathbf{M}(\mathbf{q})$ is the mass matrix, and $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$ is the Coriolis matrix. The generalize coordinates comprise the absolute and relative Euler angles, $\mathbf{q} = [\theta_1, \theta_2, \theta_3, \phi_1, \phi_2]^T$. Since the motors are located along the $\phi_1$ and $\phi_2$ coordinates, $\boldsymbol{\tau} = [0,0,0,\tau_1,\tau_2]^T$. An alternative way to write the equations of motion from the robotic manipulator form is

$$\sum_{j=1}^{n} m_{ij}\ddot{q}_j + \sum_{j=1}^{n} c_{ij}\dot{q}_j = \tau_i. \tag{C.1}$$

We can compute the components of the Coriolis matrix, $c_{ij}$, from the components of the mass matrix $m_{ij}$ by taking advantage of the relationship that the Christoffel symbols of the first type provides [55] written as

$$\sum_{j=1}^{n} c_{ij}\dot{q}_j = \sum_{j=1}^{n}\sum_{k=1}^{n} \left(\frac{\partial m_{ij}}{\partial q_k} - \frac{1}{2}\frac{\partial m_{jk}}{\partial q_i}\right)\dot{q}_k\dot{q}_j. \tag{C.2}$$

The components $m_{ij}$ can be obtained from taking the double partial derivative of the kinetic energy with respect $\dot{q}$ since the kinetic energy can be written in terms of the mass matrix [55] as

$$T = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} m_{ij}(\mathbf{q})\dot{q}_i\dot{q}_j = \frac{1}{2}\dot{\mathbf{q}}^T\mathbf{M}(\mathbf{q})\dot{\mathbf{q}} \tag{C.3}$$

and

$$\mathbf{M}(\mathbf{q}) = \frac{\partial^2(T)}{\partial\dot{\mathbf{q}}\partial\dot{\mathbf{q}}} = \frac{\partial^2(\frac{1}{2}\dot{\mathbf{q}}^T\mathbf{M}(\mathbf{q})\dot{\mathbf{q}})}{\partial\dot{\mathbf{q}}\partial\dot{\mathbf{q}}}, \tag{C.4}$$

which is equivalent to

$$m_{ij} = \frac{\partial^2(\frac{1}{2}\dot{\mathbf{q}}^T\mathbf{M}(\mathbf{q})\dot{\mathbf{q}})}{\partial\dot{q}_i\partial\dot{q}_j}. \tag{C.5}$$

Therefore, the total equations of motion can be obtained from the kinetic energy of the system.

With the assumption of a point mass tail, there is no rotational tail inertia and the kinetic energy has the representation

$$T = \frac{1}{2}(m_b \dot{\boldsymbol{\rho}}_b^T \dot{\boldsymbol{\rho}}_b + m_t \dot{\boldsymbol{\rho}}_t^T \dot{\boldsymbol{\rho}}_t + \mathbf{J}^b \boldsymbol{\omega}^T \boldsymbol{\omega}) \tag{C.6}$$

where $\mathbf{J}^b = diag(J_1, J_2, J_3)$ is the body rotational inertia tensor expressed in the body frame and $m_i$ represents the mass of each link.

The linear velocities, $\dot{\boldsymbol{\rho}}_b$ and $\dot{\boldsymbol{\rho}}_t$ are computed from the time differentiation of the vectors $\boldsymbol{\rho}_b^O$ and $\boldsymbol{\rho}_t^O$. These can be obtained from the vector sum starting at $\boldsymbol{\rho}_b^O$ in Fig. 4.3

$$\boldsymbol{\rho}_b^O - \boldsymbol{\pi}_b^O + \boldsymbol{\pi}_t^O - \boldsymbol{\rho}_t^O = \mathbf{0}. \tag{C.7}$$

The coordinate vector from $P$ to $b$ is

$$\boldsymbol{\pi}_b^O = \mathbf{R}_b^O \boldsymbol{\pi}_b^b \tag{C.8}$$

where $\boldsymbol{\pi}_b^b$ is along $\mathbf{b}_1$, expressed in the body frame as

$$\boldsymbol{\pi}_b^b = [l_b, 0, 0]^T. \tag{C.9}$$

The coordinate vector from $P$ to $t$ is

$$\boldsymbol{\pi}_t^O = \mathbf{R}_b^O \mathbf{R}_t^b \boldsymbol{\pi}_t^t \tag{C.10}$$

where $\boldsymbol{\pi}_t^t$ is along $\mathbf{t}_1$, expressed in the tail frame as

$$\boldsymbol{\pi}_t^t = [l_t, 0, 0]^T. \tag{C.11}$$

Notice that $\boldsymbol{\pi}_t^t$ is first rotated into the body frame by $\mathbf{R}_t^b$ and then rotated into the reference frame by $\mathbf{R}_b^0$.

Using the constraint that the $c$ and $O$ are located at the same position we have

$$m_b \boldsymbol{\rho}_b^O + m_t \boldsymbol{\rho}_t^O = \mathbf{0}, \tag{C.12}$$

which allows $\boldsymbol{\rho}_b^O$ and $\boldsymbol{\rho}_t^O$ to be written dependently as

$$\boldsymbol{\rho}_t^O = -\frac{m_b}{m_t} \boldsymbol{\rho}_b^O. \tag{C.13}$$

Combining (C.7), (C.10), and (C.13) we obtain the two following coordinate vectors for time differentiation

$$\boldsymbol{\rho}_b^O = \frac{m_t}{m_b + m_t}(\boldsymbol{\pi}_b^O - \boldsymbol{\pi}_t^O) \tag{C.14}$$

$$= \frac{m_t}{m_b + m_t}(\mathbf{R}_b^O [l_b, 0, 0]^T - \mathbf{R}_b^O \mathbf{R}_t^b [l_t, 0, 0]^T). \tag{C.15}$$

With the assumption that $l_b = 0$

$$\boldsymbol{\rho}_b^O = -\frac{m_t}{m_b + m_t} \mathbf{R}_b^O \mathbf{R}_t^b [l_t, 0, 0]^T \tag{C.16}$$

and the derivative is

$$\dot{\boldsymbol{\rho}}_b^O = -\frac{m_t}{m_b+m_t}(\dot{\mathbf{R}}_b^O\mathbf{R}_t^b + \mathbf{R}_b^O\dot{\mathbf{R}}_t^b)[l_t,0,0]^T. \tag{C.17}$$

By the COM constraint (C.13), the tail position and velocity can be written as

$$\boldsymbol{\rho}_t^O = \frac{m_b}{m_b+m_t}\mathbf{R}_b^O\mathbf{R}_t^b[l_t,0,0]^T \tag{C.18}$$

and

$$\dot{\boldsymbol{\rho}}_t^O = \frac{m_b}{m_b+m_t}(\dot{\mathbf{R}}_b^O\mathbf{R}_t^b + \mathbf{R}_b^O\dot{\mathbf{R}}_t^b)[l_t,0,0]^T. \tag{C.19}$$

The angular velocity is computed from the identity [55]

$$\hat{\boldsymbol{\omega}}_{On}^n = \mathbf{R}_n^{OT}\dot{\mathbf{R}}_n^O \tag{C.20}$$

where the skew symmetric matrix operator

$$\hat{\mathbf{a}} = \mathbf{a}\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}. \tag{C.21}$$

Hence, the body angular velocity $\boldsymbol{\omega}$ is simply

$$\hat{\boldsymbol{\omega}} = \mathbf{R}_b^{OT}\dot{\mathbf{R}}_b^O. \tag{C.22}$$

The equations (C.17), (C.19), and (C.22) provide the necessary terms to define the kinetic energy. The equations of motion can then be analytically computed in Mathematica and exported to MATLAB. The rows of $\ddot{\mathbf{q}}$ can be used to obtain the form necessary for simulation in MATLAB where

$$\ddot{\mathbf{q}} = \mathbf{M}(\mathbf{q})^{-1}(-\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \boldsymbol{\tau}). \tag{C.23}$$

The details of $\mathbf{M}$ and $\mathbf{C}$ are excessive and are not included in this text.

# Appendix D

# 2DOF Tailbot Angular Momentum

To prove that (4.7), restated here as

$$\mathbf{H}^O = \mathbf{J}^O \boldsymbol{\omega}^O + M \boldsymbol{\zeta}^O \times \dot{\boldsymbol{\zeta}}^O, \tag{D.1}$$

is a valid expressions of angular momentum and can be obtained from first principles we start with the fundamental equation for angular momentum for the two link system as

$$\mathbf{H}^O = \mathbf{J}^O \boldsymbol{\omega}^0 + m_b \boldsymbol{\rho}_b^O \times \dot{\boldsymbol{\rho}}_b^O + m_t \boldsymbol{\rho}_t^O \times \dot{\boldsymbol{\rho}}_t^O \tag{D.2}$$

where $\mathbf{J}^O$ is the inertia tensor of the body and since the tail is assumed to be a point mass, its inertia tensor is the zero matrix. Substituting in (C.13) and its derivative results in

$$\begin{aligned} \mathbf{H}^O &= \mathbf{J}^O \boldsymbol{\omega}^O + m_b \boldsymbol{\rho}_b^O \times \dot{\boldsymbol{\rho}}_b^O + \frac{m_b^2}{m_t} \boldsymbol{\rho}_b^O \times \dot{\boldsymbol{\rho}}_b^O & \text{(D.3)} \\ &= \mathbf{J}^O \boldsymbol{\omega}^O + \frac{m_b(m_t + m_b)}{m_t} \boldsymbol{\rho}_b^O \times \dot{\boldsymbol{\rho}}_b^O. & \text{(D.4)} \end{aligned}$$

With the definition of $\boldsymbol{\zeta}^O$ as

$$\boldsymbol{\zeta}^O = \boldsymbol{\rho}_t^O - \boldsymbol{\rho}_b^O \tag{D.5}$$

we can substitute (C.13) into (D.5) to compute

$$\boldsymbol{\rho}_b^O = -\frac{m_t}{m_t + m_b} \boldsymbol{\zeta}^O. \tag{D.6}$$

Substituting (D.6) and its derivative into (D.4) finalizes to

$$\mathbf{H}^O = \mathbf{J}^O \boldsymbol{\omega}^O + \frac{m_b m_t}{m_b + m_t} \boldsymbol{\zeta}^O \times \dot{\boldsymbol{\zeta}}^O. \tag{D.7}$$

With the definition

$$M = \frac{m_b m_t}{m_b + m_t} \tag{D.8}$$

(D.7) matches (4.7).

# Appendix E

# LS Method Error

Using the property for two vectors $\mathbf{a}$ and $\mathbf{b}$ that

$$\mathbf{a}\cdot\mathbf{b} = \sqrt{|\mathbf{a}|^2|\mathbf{b}|^2 - |\mathbf{a}\times\mathbf{b}|^2}, \tag{E.1}$$

$\mathbf{e}_{LS}$ can be rewritten as

$$\mathbf{e}_{LS} = \mathbf{t}\sqrt{|\mathbf{t}|^2|\mathbf{H}_s|^2 - |\mathbf{t}\times\mathbf{H}_s|^2}. \tag{E.2}$$

With $|\mathbf{t}| = 1$ and taking the magnitude results in

$$|\mathbf{e}_{LS}| = 1\left|\sqrt{1|\mathbf{H}_s|^2 - |\mathbf{t}\times\mathbf{H}_s|^2}\right| \tag{E.3}$$

$$= \left|\sqrt{|\mathbf{H}_s|^2 - |\mathbf{t}\times\mathbf{H}_s|^2}\right|. \tag{E.4}$$

From here $|\mathbf{e}_{LS}| = |\mathbf{H}_s|$ is maximized when $|\mathbf{t}\times\mathbf{H}_s| = 0$ or physically when $\mathbf{t}$ and $\mathbf{H}_s$ are aligned. Coincidentally, from (4.44), $\mathbf{e}_{LS}$ is also maximized when $\mathbf{H}_s$ is in the nullspace of $\mathbf{F}^{\dagger}$, which would render the LS method from (4.38) inactive.

# Appendix F

# Properties of Least Squares Method

Here we will now show that the computed angular velocity of the tail by the LS method, $\boldsymbol{\psi}_{LS}$, shares the plane containing $\mathbf{t}$, $\mathbf{H}_s$, which is proved if the vector triple product

$$(\mathbf{t} \times \boldsymbol{\psi}_{LS}) \cdot \mathbf{H}_s = 0. \tag{F.1}$$

In general, the relationship between the angular velocity of the tail expressed in the body frame, $\boldsymbol{\psi}$, and $[\dot{\phi}_1, \dot{\phi}_2]^T$ can be derived from the skew symmetric property [55]

$$\hat{\boldsymbol{\psi}} = \dot{\mathbf{R}}_t^b \mathbf{R}_t^{bT}. \tag{F.2}$$

Expanding terms results in

$$\hat{\boldsymbol{\psi}} = \begin{bmatrix} -s_{\phi_1}c_{\phi_2}\dot{\phi}_1 - c_{\phi_1}s_{\phi_2}\dot{\phi}_2 & s_{\phi_1}s_{\phi_2}\dot{\phi}_1 - c_{\phi_1}c_{\phi_2}\dot{\phi}_2 & c_{\phi_1}\dot{\phi}_1 \\ c_{\phi_2}\dot{\phi}_2 & -s_{\phi_2}\dot{\phi}_2 & 0 \\ -c_{\phi_1}c_{\phi_2}\dot{\phi}_1 + s_{\phi_1}s_{\phi_2}\dot{\phi}_2 & c_{\phi_1}s_{\phi_2}\dot{\phi}_1 + s_{\phi_1}c_{\phi_2}\dot{\phi}_2 & -s_{\phi_1}\dot{\phi}_1 \end{bmatrix} \begin{bmatrix} c_{\phi_1}c_{\phi_2} & s_{\phi_2} & -s_{\phi_1}c_{\phi_2} \\ -c_{\phi_1}s_{\phi_2} & c_{\phi_2} & s_{\phi_1}s_{\phi_2} \\ s_{\phi_1} & 0 & c_{\phi_1} \end{bmatrix} \tag{F.3}$$

$$= \begin{bmatrix} 0 & -\cos(\phi_1)\dot{\phi}_2 & \dot{\phi}_1 \\ \cos(\phi_1)\dot{\phi}_2 & 0 & -\sin(\phi_1)\dot{\phi}_2 \\ -\dot{\phi}_1 & \sin(\phi_1)\dot{\phi}_2 & 0 \end{bmatrix}. \tag{F.4}$$

With the cross product identity, we can conveniently decompose $\boldsymbol{\psi}$ into

$$\boldsymbol{\psi} = \begin{bmatrix} \sin(\phi_1)\dot{\phi}_2 \\ \dot{\phi}_1 \\ \cos(\phi_1)\dot{\phi}_2 \end{bmatrix} \tag{F.5}$$

$$= \mathbf{G} \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \end{bmatrix} \tag{F.6}$$

where

$$\mathbf{G} = \begin{bmatrix} 0 & \sin(\phi_1) \\ 1 & 0 \\ 0 & \cos(\phi_1) \end{bmatrix}. \tag{F.7}$$

Therefore, by (4.38), the LS method results in an angular velocity of

$$\boldsymbol{\psi}_{LS} = \mathbf{G}\mathbf{F}^{\dagger}\mathbf{H}_s. \tag{F.8}$$

The triple product in (F.1) can now be rewritten as

$$\begin{aligned} (\mathbf{t} \times \boldsymbol{\psi}_{LS}) \cdot \mathbf{H}_s &= (\hat{\mathbf{t}}\mathbf{G}\mathbf{F}^{\dagger}\mathbf{H}_s) \cdot \mathbf{H}_s & \text{(F.9)} \\ &= \mathbf{H}_s^T \mathbf{F}^{\dagger T} \mathbf{G}^T \hat{\mathbf{t}}^T \mathbf{H}_s & \text{(F.10)} \end{aligned}$$

where

$$\begin{aligned} \mathbf{F}^{\dagger T}\mathbf{G}^T\hat{\mathbf{t}}^T &= \begin{bmatrix} -c_{\phi_1}t_{\phi_2} & s_{\phi_1} \\ 1 & 0 \\ s_{\phi_1}t_{\phi_2} & c_{\phi_1} \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ s_{\phi_1} & 0 & c_{\phi_1} \end{bmatrix} \begin{bmatrix} 0 & -s_{\phi_1}c_{\phi_2} & -s_{\phi_2} \\ s_{\phi_1}c_{\phi_2} & 0 & c_{\phi_1}c_{\phi_2} \\ s_{\phi_2} & -c_{\phi_1}c_{\phi_2} & 0 \end{bmatrix} & \text{(F.11)} \\ &= \begin{bmatrix} -c_{\phi_1}t_{\phi_2} & s_{\phi_1} \\ 1 & 0 \\ s_{\phi_1}t_{\phi_2} & c_{\phi_1} \end{bmatrix} \begin{bmatrix} s_{\phi_1}c_{\phi_2} & 0 & c_{\phi_1}c_{\phi_2} \\ c_{\phi_1}s_{\phi_2} & -c_{\phi_2} & -s_{\phi_1}s_{\phi_2} \end{bmatrix} & \text{(F.12)} \\ &= \begin{bmatrix} 0 & -s_{\phi_1}c_{\phi_2} & -s_{\phi_2} \\ s_{\phi_1} - c_{\phi_2} & 0 & c_{\phi_1}c_{\phi_2} \\ s_{\phi_2} & -c_{\phi_1}c_{\phi_2} & 0 \end{bmatrix}. & \text{(F.13)} \end{aligned}$$

It is well known that given any vector $\mathbf{x}$ and skew symmetric matrix $\mathbf{A}$

$$\mathbf{x}^T\mathbf{A}\mathbf{x} = 0. \tag{F.14}$$

Since $\mathbf{F}^{\dagger T}\mathbf{G}^T\hat{\mathbf{t}}^T$ is skew symmetric, the triple product in (F.10) is zero and we can conclude that (F.1) is true. This guarantees that all three vectors share the same plane. Again, this physically implies that the LS method computes a desired velocity of the tail perpendicular to the plane containing $\mathbf{t}$ and $\mathbf{H}_s$.

# Appendix G

# Error and Properties of the Perpendicular Method

Similar to the LS method, the error in the perpendicular controller is the difference between the desired, $\boldsymbol{\psi}_{PD}$, and actual angular velocity $\boldsymbol{\psi}_P = \mathbf{G}\mathbf{G}^T\boldsymbol{\psi}_{PD}$ described by

$$\begin{aligned} \mathbf{e}_P &= \boldsymbol{\psi}_{PD} - \boldsymbol{\psi}_P && \text{(G.1)} \\ &= \boldsymbol{\psi}_{PD} - \mathbf{G}\mathbf{G}^T\boldsymbol{\psi}_{PD} && \text{(G.2)} \\ &= (\mathbf{I}_{3\times 3} - \mathbf{G}\mathbf{G}^T)\boldsymbol{\psi}_{PD}. && \text{(G.3)} \\ & && \text{(G.4)} \end{aligned}$$

Since $\boldsymbol{\psi}_{PD} = \alpha(\mathbf{t}\times\mathbf{H}_s)$, where $\alpha = \gamma/||\mathbf{t}\times\mathbf{H}_s||$ from (4.52), the error can be reduced to

$$\begin{aligned} \mathbf{e}_P &= (\mathbf{I}_{3\times 3} - \mathbf{G}\mathbf{G}^T)\alpha\hat{\mathbf{t}}\mathbf{H}_s && \text{(G.5)} \\ &= \alpha \begin{bmatrix} c_{\phi_1}^2 & 0 & -s_{\phi_1}c_{\phi_1} \\ 0 & 0 & 0 \\ -s_{\phi_1}c_{\phi_1} & 0 & s_{\phi_1}^2 \end{bmatrix} \begin{bmatrix} 0 & s_{\phi_1}c_{\phi_2} & s_{\phi_2} \\ -s_{\phi_1}c_{\phi_2} & 0 & -c_{\phi_1}c_{\phi_2} \\ -s_{\phi_2} & c_{\phi_1}c_{\phi_2} & 0 \end{bmatrix} \mathbf{H}_s && \text{(G.6)} \\ &= \alpha s_{\phi_2} \begin{bmatrix} s_{\phi_1}c_{\phi_1} & 0 & c_{\phi_1}^2 \\ 0 & 0 & 0 \\ -s_{\phi_1}^2 & 0 & -s_{\phi_1}c_{\phi_1} \end{bmatrix} \mathbf{H}_s. && \text{(G.7)} \end{aligned}$$

In general $\mathbf{e}_P \neq \mathbf{0}$, however a few special cases exist when it is $\mathbf{0}$ e.g. when $\mathbf{H}_s = [c_{\phi_1}, free, -s_{\phi_1}]^T$ or $[0, free, 0]^T$ or due to the sine dependency when $\phi_2 = 0\ rad$. The last case corresponds to when the tail and body are aligned in yaw and may prove as a useful nominal position.

Even with this error present we can show that the resulting angular velocity $\boldsymbol{\psi}_P$ is still perpendicular to the resulting angular velocity from the LS method, $\boldsymbol{\omega}_{LS} = \mathbf{G}\mathbf{F}^{\dagger}\mathbf{H}_s$ from (F.8), which

implies that they will not compete. Taking their dot product is equivalent to

$$\begin{aligned} \boldsymbol{\psi}_P \cdot \boldsymbol{\psi}_{LS} &= (\mathbf{G}\mathbf{G}^T\boldsymbol{\psi}_{PD}) \cdot (\mathbf{G}\mathbf{F}^\dagger\mathbf{H}_s) && \text{(G.8)} \\ &= (\mathbf{G}\mathbf{G}^T\boldsymbol{\psi}_{PD})^T(\mathbf{G}\mathbf{F}^\dagger\mathbf{H}_s) && \text{(G.9)} \\ &= \boldsymbol{\psi}_{PD}^T\mathbf{G}\mathbf{G}^T\mathbf{G}\mathbf{F}^\dagger\mathbf{H}_s. && \text{(G.10)} \end{aligned}$$

Adding our definition of $\boldsymbol{\psi}_{PD}$ in (4.52) along with $\mathbf{G}^T\mathbf{G} = \mathbf{I}_{2x2}$ leads to

$$\begin{aligned} &= \alpha(\hat{\mathbf{t}}\mathbf{H}_s)^T\mathbf{G}\mathbf{F}^\dagger\mathbf{H}_s && \text{(G.11)} \\ &= \alpha\mathbf{H}_s^T\hat{\mathbf{t}}^T\mathbf{G}\mathbf{F}^\dagger\mathbf{H}_s && \text{(G.12)} \\ &= \alpha\mathbf{H}_s^T \begin{bmatrix} 0 & -s_{\phi_1}c_{\phi_2} & -s_{\phi_2} \\ s_{\phi_1}c_{\phi_2} & 0 & c_{\phi_1}c_{\phi_2} \\ s_{\phi_2} & -c_{\phi_1}c_{\phi_2} & 0 \end{bmatrix} \begin{bmatrix} 0 & s_{\phi_1} \\ 1 & 0 \\ 0 & c_{\phi_1} \end{bmatrix} \frac{1}{Ml_t^2} \begin{bmatrix} -c_{\phi_1}t_{\phi_2} & 1 & s_{\phi_1}t_{\phi_2} \\ s_{\phi_1} & 0 & c_{\phi_1} \end{bmatrix} \mathbf{H}_s && \text{(G.13)} \\ &= \frac{\alpha}{Ml_t^2}\mathbf{H}_s^T \begin{bmatrix} 0 & -s_{\phi_1}c_{\phi_2} & -s_{\phi_2} \\ s_{\phi_1}c_{\phi_2} & 0 & c_{\phi_1}c_{\phi_2} \\ s_{\phi_2} & -c_{\phi_1}c_{\phi_2} & 0 \end{bmatrix} \mathbf{H}_s. && \text{(G.14)} \end{aligned}$$

Since the matrix enclosed by $\mathbf{H}_s$ is skew symmetric, the dot product is zero, which concludes the orthogonality.