

ABSTRACT

Title of Document: DESIGN AND APPLICATION OF PAN AND
TILT SERVO GIMBALS IN POINTING,
ACQUISITION, AND TRACKING

John Robertson Rzasa
Master of Science in Electrical Engineering
2007

Directed By: Professor Christopher C. Davis
Electrical and Computer Engineering

Directional wireless communications systems are fast becoming an essential part of the world's broadband network infrastructure. When using these types of transceivers in reconfigurable networks, it becomes necessary to point them rapidly and accurately to different locations, or even to targets that may be in motion. The most efficient way of doing this is through the use of two-axis pan and tilt motion stages, also known as gimbals. This paper presents the motivation for, design and construction of, and testing of a pair of multipurpose servo gimbals, usable for both RF and laser transceivers. The gimbals are tested in terms of pointing error, movement speed, and response time. For the network portion, relink times as a function of angular rotation are examined, as well as the angular offset vs. data rate. The gimbal is also tested as part of a remote surveillance network, evaluating its ability to track moving objects.

DESIGN AND APPLICATION OF PAN AND TILT SERVO GIMBALS IN POINTING,
ACQUISITION, AND TRACKING

By

John Robertson Rzasa

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of

Master of Science
2007

Advisory Committee:
Professor Christopher C. Davis, Chair
Professor Gilmer L. Blankenship
Professor Julius Goldhar

© Copyright by
John Robertson Rzasa
2007

Dedication

To my parents and brother, to whom I owe all.

Acknowledgements

After working in the Maryland Optics Group for so long, the list of people I'm indebted to could go on for ages. I owe so much to Professor Chris Davis, who gave me a job as a sophomore undergrad and then let me continue as a grad student. It is from him that I learned what it is like to be a true engineer, not to mention a cheerful one! I owe much to Dr. Milner, Dr. Balzano, and Dr. Hodzic for their advice and guidance over the years, and some good food as well! I will never forget the endless stream of help from Dr. Shawn Ho, who formed a great engineering tag-team with me. I'll always appreciate Dr. Sugi Trisno's patient advice, as well as Yuju Hung's entertaining conversations. I am also deeply indebted to John 'J' Pyle, the IREAP machine supervisor, who taught me how to be a machinist, something I've grown to love very much. Finally, I thank everyone in MOG and the school in general, who've helped me along the way and made this a fun place to work.

Table of Contents

Dedication.....	ii
Acknowledgements.....	iii
Table of Contents.....	iv
List of Tables.....	v
List of Figures.....	vi
Chapter 1: Introduction and Background.....	1
1.1 Introduction.....	1
1.2 Mechanical Overview of Servo Motors.....	3
1.3 Radial vs. Axial Flux Servo Motors.....	5
1.4 Electrical Overview of Servo Motors.....	6
1.5 Servo Motor Control Methods.....	10
1.5.1 Proportional-Integral-Differential (PID) Control.....	12
1.5.2 Proportional-Integral-Velocity (PIV) Control.....	13
1.5.3 Feed Forward Control.....	15
1.6 Servos vs. steppers in PAT applications.....	16
1.7 Motor Selection.....	17
1.8 Controller Selection.....	18
Chapter 2: Experimental Design.....	20
2.1 Design Considerations.....	21
2.2 Design specifications.....	23
2.3 PIV Control Loop Gains.....	23
2.4 Motor Control through RS232.....	26
2.5 Mechanical Design.....	27
Chapter 3: Experimental Results.....	32
3.1 Introduction.....	32
3.2 Angular Rotation Time and Positioning Error.....	33
3.3 Reconfigurable Network Experiments.....	35
3.3.1 Transceiver Angular Offset vs. Packet Loss.....	35
3.3.2 Make-Break-Make Tests.....	37
3.3.2.1 RAD RF Antenna Results.....	39
3.3.2.2 Canobeam Results.....	42
3.4 Gross Motion Stabilization.....	44
3.5 Surveillance Applications.....	47
Chapter 4: Future Work.....	49
Appendix A: Gimbal CAD Design Drawings.....	52
Bibliography.....	55

List of Tables

Table 1: Comparison of servo and stepper motors.....	17
Table 2: Specifications of the Bodine E-Torq Motor.....	18
Table 3: Gimbal Design criteria.....	23
Table 4: Loop gains and their associated effects.....	25

List of Figures

Figure 1.1: Cutaway of a DC brushless servo [1].....	5
Figure 1.2: Radial and Axial flux configurations for servo motors.....	6
Figure 1.3: Hall-Effect Sensor [2]	7
Figure 1.4: Hall-effect Sensor Alignment vs. motor current phase [1]	8
Figure 1.5: Rotary optical encoder [4].....	9
Figure 1.6: Limit cycling behavior in a servo motor	11
Figure 1.7: Typical servo move profile (current, velocity and position).....	11
Figure 1.8 PID control loop diagram [6]	13
Figure 1.9: PIV control loop [6]	14
Figure 1.10: Copley Xenus Control Method [7].....	14
Figure 1.11 Feed Forward Control [7].....	16
Figure 1.12: Copley Xenus Servo Drive [8].....	19
Figure 2.1: Velocity loop filter [7].....	21
Figure 2.2: Rotation with and without overshoot	22
Figure 2.3: PIV tuning method [11].....	24
Figure 2.4: Adjustable Friction Bushing.....	29
Figure 2.5: Conceptual Design of gimbal holding a Canobeam.....	30
Figure 2.6: Finished Canobeam Gimbal	30
Figure 2.7: Conceptual design of gimbal holding a RF antenna.....	31
Figure 2.8: Finished RAD antenna gimbal	31
Figure 3.1: Angular move vs. time, azimuth motor on left, elevation on right	33

Figure 3.2: Standard Deviation of the difference between commanded and steady state angular rotations, azimuth motor.....	34
Figure 3.3: Block diagram of experimental setup.....	35
Figure 3.4: Angular offset vs. average data rate (RAD).....	36
Figure 3.5: Angular offset vs. average data rate (Canobeam)	36
Figure 3.6: Angular move vs. Reconnect time	40
Figure 3.7: Time vs. data rate and motor rotation for a 180° move.....	40
Figure 3.8: Angular move vs. Data Rate.....	42
Figure 3.9: Time vs. data rate and motor rotation for a 180° move.....	43
Figure 3.10: Azimuthal/Elevation motor velocity and gyro velocity	45
Figure 3.11: Residue of Motor and Gyro Angular Velocities.....	45
Figure 3.12: Still capture from autonomous tracking experiment	48
Figure 4.1: Conceptual design of light-weight RF servo gimbal.....	51
Figure A.1: Unloaded Gimbal Top View	52
Figure A.2: Gimbal side view	53
Figure A.3: Gimbal Front View.....	54
Bibliography	55

Chapter 1: Introduction and Background

1.1 Introduction

In the field of directional wireless communications, being able to point a transceiver precisely is the most important aspect in establishing a link. When the two ends are fixed, this becomes a relatively simple process of coarsely aligning the two units by eye or beacon, and then using some type of signal strength indicator to fine-tune the transceivers' orientations. However, this situation becomes markedly more difficult when one has multiple nodes that must be reconfigured automatically, or if one or more of the nodes are moving. While these situations may seem far-fetched at first, there are many areas where they can be found, from networks between skyscraper rooftops, between first-responder vehicles and a command center, between military vehicles and planes, or amongst ships at sea. Knowing how to acquire, point to, and track these nodes is a very complicated process, which has been investigated by other members of our group, so this paper will touch only lightly on the theory involved. Once one knows where to point, the issue becomes how to move the transceiver as fast and as accurately as possible to the new location, or in moving cases, to track the moving node closely. With directional wireless units regularly exceeding 100 Mbps, even a few seconds long break in a link can cause a large loss of data. Therefore, it is essential to develop a platform that can rotate these units to anywhere over a hemisphere or more, while at the same time doing so reliably and quickly enough to minimize the mechanical delay reconnect times. The most common method is to employ a two-axis pan and tilt gimbal,

which utilizes two motors to provide azimuthal and elevation rotation. Starting from this simple idea the variations become enormous. From the gimbal on a toy to the half million dollar one on a helicopter, there is a large range of technology used. This paper focuses on the use of a relatively new type of motor, namely the direct-drive low-profile (pancake) DC brushless servo motor. Amongst other things, these motors provide very high peak torque, smooth low velocity control, as well as milli-radian positioning. Chapter 1 of this paper details the mechanical and electrical theory behind the motors, the differences between servo and stepper motors, and the types of controllers currently in use. Chapter 2 explains the design and fabrication of the gimbal used in this paper, as well as the tuning of the motor controllers. Also included are the gimbals' limitations and general specifications. Chapter 3 lists the experimental results which tested the units' positioning performance, response time, and ability to reconfigure a network link using two types of directional wireless transceivers. Chapter 4 brings the paper to a conclusion by suggesting future work and other potential areas of interest.

1.2 Mechanical Overview of Servo Motors

Servo motors have several distinct characteristics that separate them from their stepper counterparts. The biggest is the lack of direct gearing between the rotor and the output shaft. This eliminates the backlash and cogging behaviors found in steppers, where there is a period of slop between the gear teeth before movement actually begins, and where the shaft continues to move after the motor has stopped. This can lead to jerky starts and stops, as well as a time delay in movement. This does not impede static positioning performance markedly, but it presents major issues when on-the-fly velocity changes or hard starts/stops are needed.

A model of a typical radial brushless DC servo motor is shown below in figure 1.1. For a long time, servo motors used brushes to transfer current from the static winding to the rotor, but this would lead to wear on the brushes, in turn shortening the lifespan of the motor. With the advent of electronic motor controllers, the brushless design was adopted, which uses control electronics to vary the currents phases to the motor's windings in the same way the brushes do. For the rest of this paper, all mention of servo motors will be of the brushless type.

Looking at figure 1.1 below, there are several objects of interest. First are the armature windings (held by the stator), which create a magnetic field that travels through the air gap to the permanent magnets on the rotor. Even though there are normally no gears in a servo motor, cogging can still exist, as there are gaps between the magnets on the rotor where the flux decreases, though this only becomes noticeable at low speeds. This type of cogging in servos is perhaps more accurately termed 'detent torque.' There are two ways to minimize this type of cogging, the most common being the addition of

some gearing to the drive shaft. This allows the motor to run at a higher speed out of its cogging region, but does not compromise power output or precision, though it can induce some backlash. The other way of minimizing cogging is to skew the magnets on the rotor so that a radial line from the center of the rotor always intersects a magnet at least once. When using a motor without gearing, it is known as a direct drive motor. This allows for the best transfer of power to the load, and avoids any of the negative aspects of gearing previously mentioned. A feature in newer servo motors (including the Bodine models used in this thesis) is the use of an ironless stator, which eliminates iron saturation, a situation where the magnetic properties of the iron limit how much current can be applied to the windings. Inducing iron saturation too often will cause overheating and possibly damage the windings or magnets. With an ironless stator, rotor magnet skewing is not necessary, as the magnetic fields aren't influenced by the material of the stator. Also, since the only mechanical connection between the shaft and the body is through the bearings, friction is very low (especially when using ball bearings).

In high torque motors such as the ones used in this thesis, the rotor actually consists of two plates of permanent magnets sandwiching the stator, which allows for a major increase in torque. This feature only exists in axial flux motors, due to the design where the stator lies in between the rotors, whereas in radial flux servos, the rotor is completely enclosed by the stator. The majority of the heat dissipated from a servo motor comes from the stator, so its outside location aids in cooling. In fact, the main limiting factor in the power of a servo motor is the heat capacity of the stator and the armature windings.

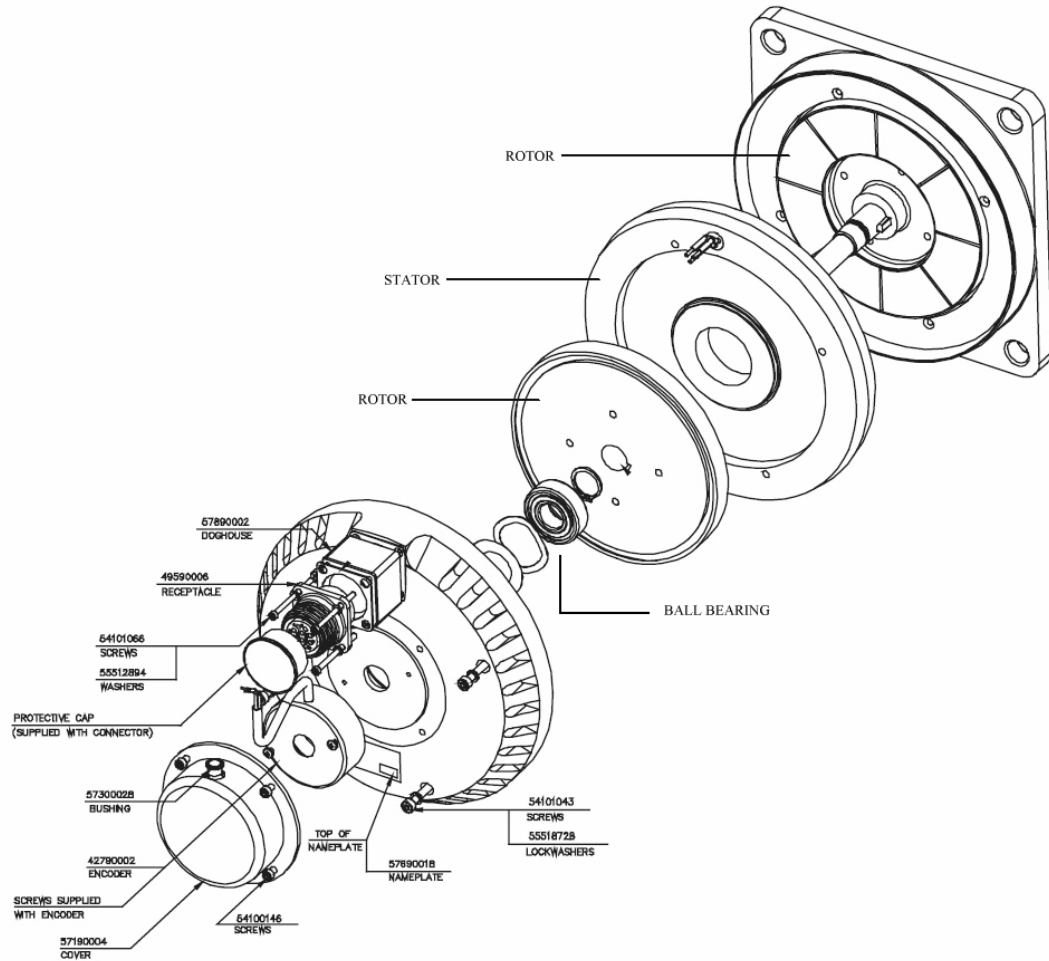


Figure 1.1: Cutaway of a DC brushless servo [1]

1.3 Radial vs. Axial Flux Servo Motors

In servo motor design, there are two types of stator/rotor configurations: radial and axial. These refer not to the mechanical configuration but to the flow of magnetic flux. The most common type is the radial servo, which has a smaller radius but longer length. Normally, the stator is on the outside, completely enclosing the permanent magnet rotor. In the case of axial motors, the rotor is a thin disk, with the stator on one side. In high torque applications, two rotors are used, essentially sandwiching the stator.

Figure 1.2 below shows the mechanical configuration of radial and axial servo motors. Each type of motor design has its advantages, namely axial flux motors allowing for higher peak torque, while radial motors perform more smoothly at very low velocities.

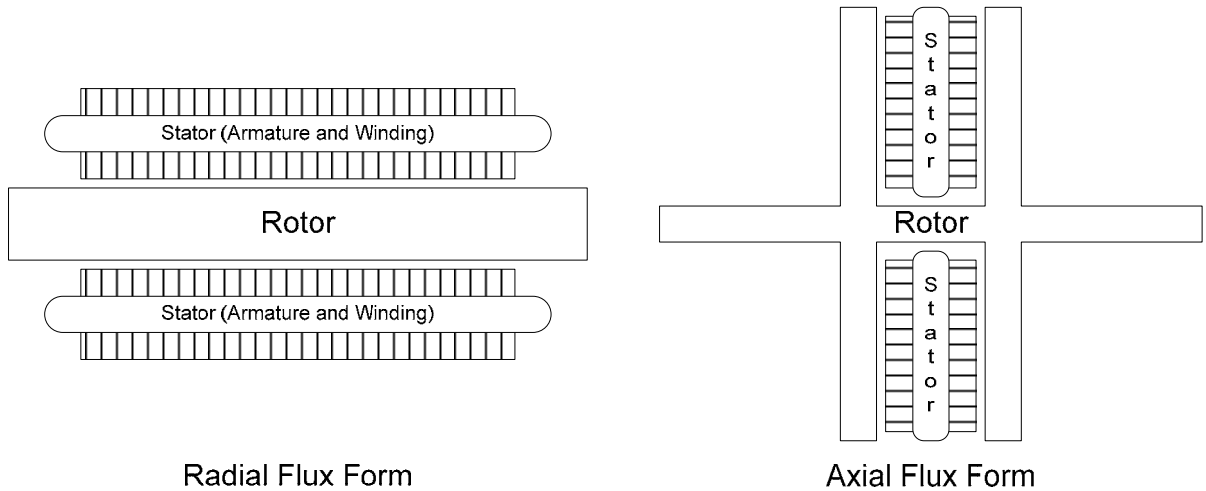


Figure 1.2: Radial and Axial flux configurations for servo motors

1.4 Electrical Overview of Servo Motors

In older brushed servo motors, the phasing of the current supplied to the rotor was determined by the orientation of the commutator, a plate on the rotor which interacted with the brushes to determine where the current should flow. In this manner, the motor's operation could be controlled with a few potentiometers. With this mechanical connection removed in brushless motors, control became more complicated. All commutation became electrically controlled, requiring the use of motor controllers that employ some type of feedback sensor to determine the rotor's position relative to the stator. The most popular sensors in use are Hall-effect sensors, optical rotary encoders, and resolvers.

Hall-effect sensors operate by changing their voltage in response to a varying magnetic field. Thus, as the magnets on the rotor move, the voltage will change proportionally. Using several sensors spaced evenly around the rotor, one can determine the rotor position as well as its velocity. Hall-effect sensors have the added benefit of having no mechanical connection to the rotor, and are usually integrated into the armature windings. In most motors, they are used as the primary sensor to determine the required current phase to send to the motor, given their fast response and lack of additional control circuitry. A diagram of three Hall sensors in a servo motor is shown in figure 1.3, while a chart of the Hall sensor alignment and current phase for the Bodine E-Torq motor is shown in figure 1.4.

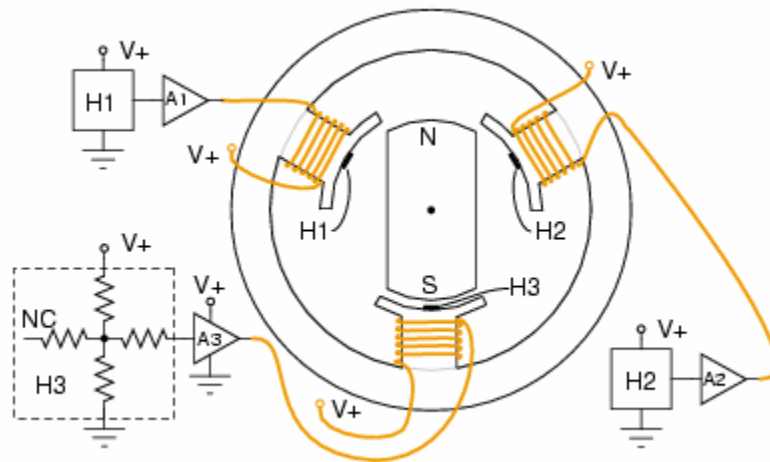


Figure 1.3: Hall-Effect Sensor [2]

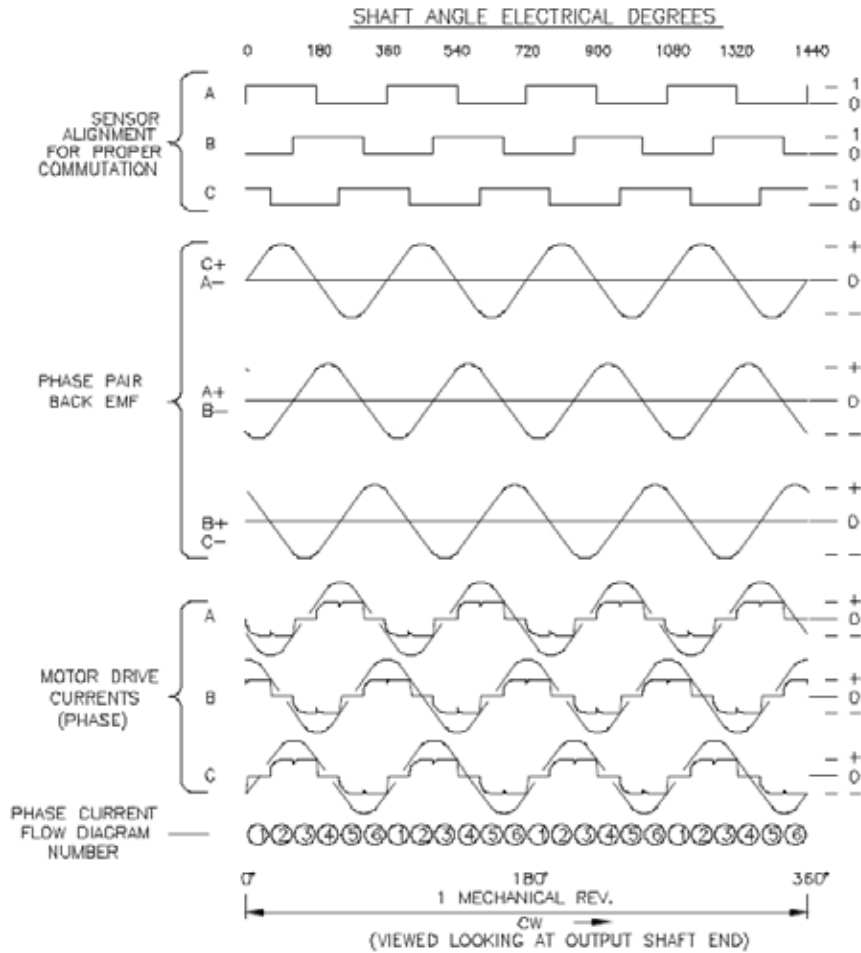


Figure 1.4: Hall-effect Sensor Alignment vs. motor current phase [1]

In some cases, such as for stepping motors, velocity measurements are not required, but instead position. In this instance, the optical rotary encoder is the primary sensor, since no commutation control is needed. It uses a slotted disk attached to the rotor with either a binary or gray code cut into it. An IR signal is passed through the code wheel, and the resulting pulses are processed in the controller to determine the rotor's position. Of course, as the resolution goes up, so does the encoder's complexity and cost. Current encoders allow for milli-radian precision [3]. The two main types are absolute and incremental encoders. The former measures the position based on an etched pattern

in the wheel, while the latter measures interrupt pulses through a slotted disk. Absolute encoders are the most popular, since they can also be used in an incremental fashion.

Figure 1.5 shows a diagram of an optical encoder.

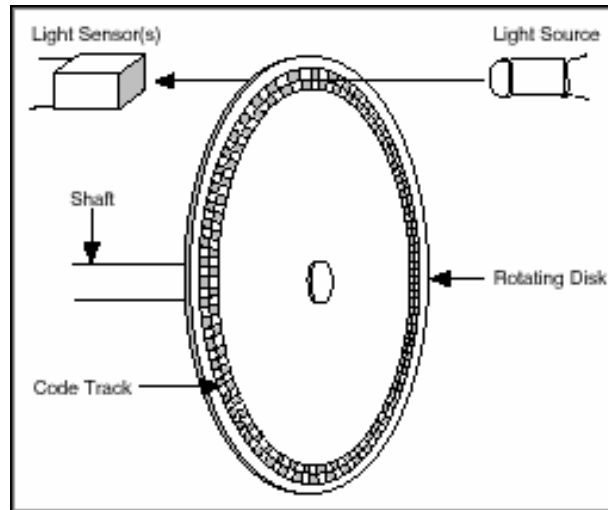


Figure 1.5: Rotary optical encoder [4]

An angular resolver uses a special rotary transformer to determine the rotor's position. The theory behind these is relatively complicated, so the reader is referred to [5] for more detail. They can provide accuracy on par with an optical encoder, and their signals can be passed through a slip-ring, something useful for reducing cabling.

Resolvers tend to be complicated devices requiring sophisticated control circuitry as well as an additional rotary transformer to provide power to the resolver itself, so they do not appear often in positioning applications, or situations with size and power constraints.

1.5 Servo Motor Control Methods

When using a servo motor in a velocity control mode, the encoder's position is used to form a velocity estimate, which in turn adjusts the underlying current loop to achieve the desired motion. It is worth noting that since the velocity used by the controller is an estimate, the velocity error is bounded by the resolution of the encoder. Thus, for positioning and constant velocity applications it is essential to have a high resolution encoder, otherwise the controller won't be able to keep the motor moving steadily (including at zero velocity). At zero velocity, the motor could bounce between encoder pulses, a phenomenon known as limit cycling (figure 1.6 with a frequency of ~2Hz). The primary method of eliminating limit cycling in servo motors is to introduce some static friction into the drive train, usually with an external bearing. Most servo motors have more than enough torque to overcome the friction of the bearing without a noticeable loss in performance.

Many servo controllers use a trapezoidal velocity profile to move to set positions, with a linear acceleration phase, a constant velocity phase, and a linear deceleration phase, hence the title 'trapezoidal.' The control loop must be tightly tuned for this to occur properly, since any mistake can cause the motor to overshoot its target, move unsteadily during the constant velocity portion, arrive too slow, or even oscillate once reaching the position. A typical trapezoidal profile is shown in figure 1.7, comparing commanded and actual current, velocity, and position.

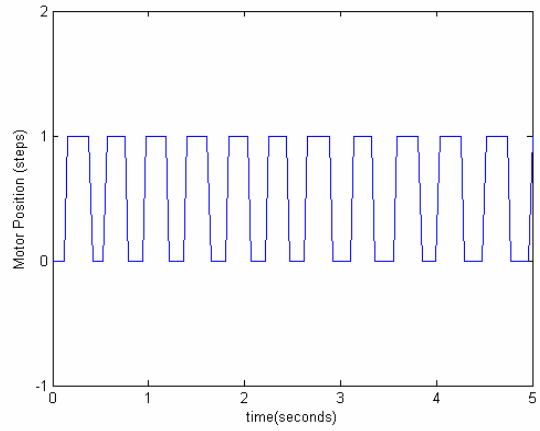


Figure 1.6: Limit cycling behavior in a servo motor

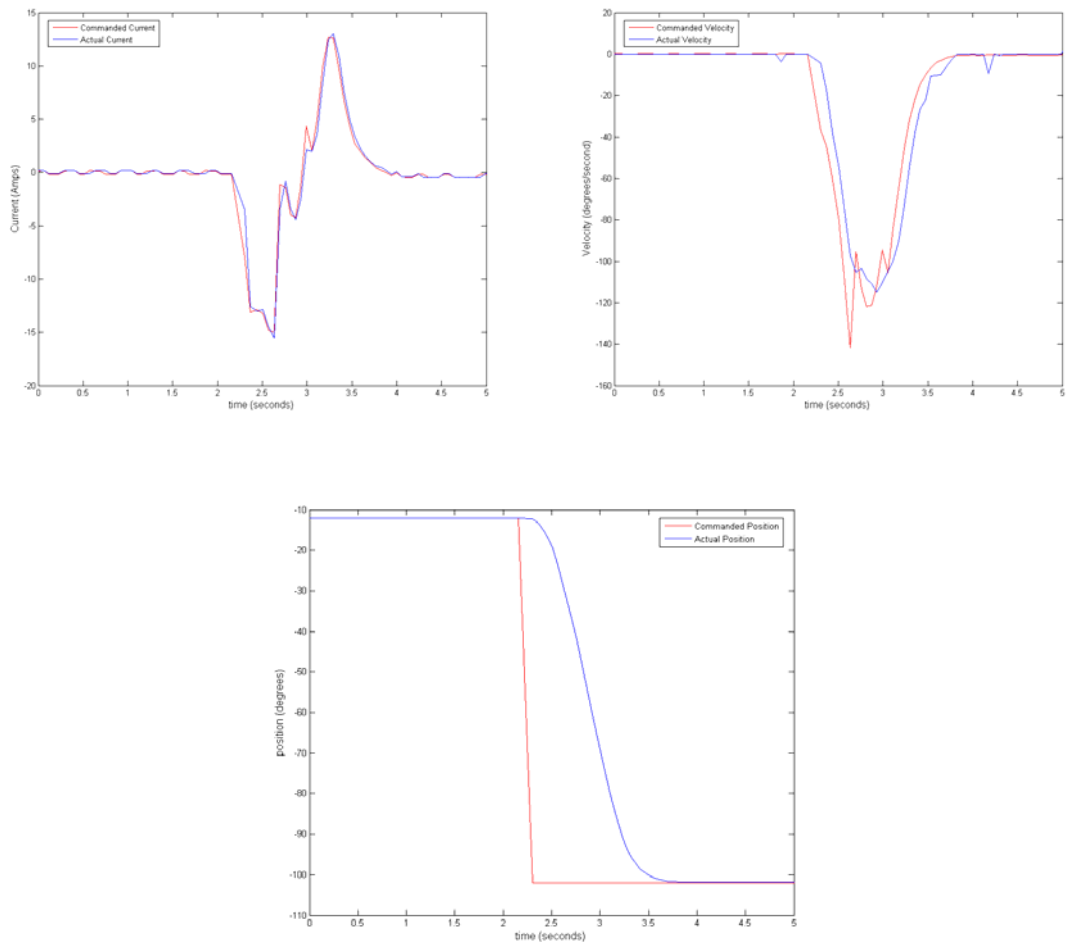


Figure 1.7: Typical servo move profile (current, velocity and position)

To ensure that a motor is moving in the proper fashion, some type of feedback control loop must be used, with the rotary optical encoder as the feedback source. There are two types of these loops used in motor control, the first (PID) being common to many different types of control, not just motors. The second (PIV) is used almost exclusively in motor control, velocity being the most important factor.

1.5.1 Proportional-Integral-Differential (PID) Control

In a PID control algorithm (figure 1.8), the system operates to minimize the error between the output signal, $\Theta(s)$, and the commanded signal, $\Theta^*(s)$. Each of the three parts (proportional, integral, and differential) play an important part in the control process. The proportional gain K_p adjusts the overall level of the output signal, driving it closer to the ideal output. However, this signal is rather slow and also has the disadvantage of never being able to drive the error signal to zero. Thus, if K_p is set too high, the output signal will never reach a steady state in static operations, such as a motor in stand still. To compensate for this, the integral factor K_i is introduced. This takes a running integral of the error over a set period of time and feeds it back into the output signal, helping to zero the steady state error. Using this, one can achieve a steady state signal with no error, as would be expected in a still motor. Because the integral term adjusts for error accumulated over time, it cannot always eliminate rapidly occurring error. There are two ways to compensate for this, namely adding a differentiation component to the error signal, or using feed forward control, which will be described in section 1.3.3. Adding the differentiation term, K_d , allows for a quicker control loop, since it contributes the rate of change of the error to the output signal. One must

carefully use this term to avoid introducing high frequency noise. The most common way of eliminating this noise is to add a low pass filter.

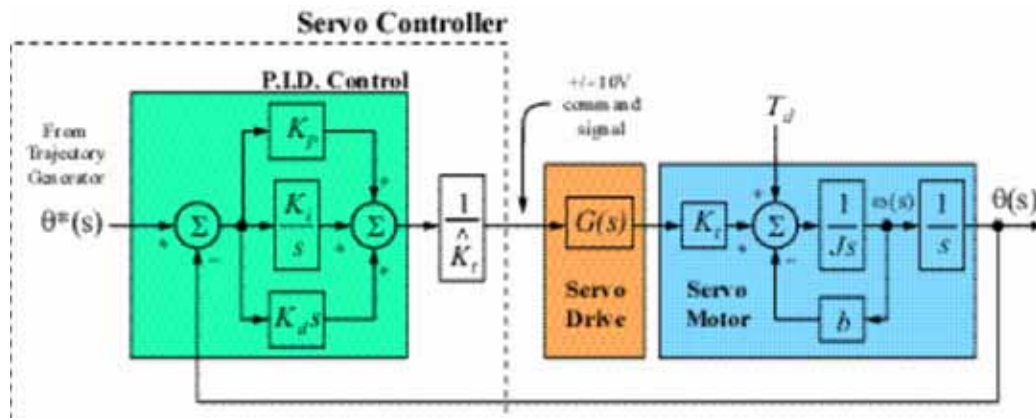


Figure 1.8 PID control loop diagram [6]

1.5.2 Proportional-Integral-Velocity (PIV) Control

A derivative of the PID controller is the PIV controller, which uses the velocity of a sensor as part of the error correction algorithm. This is especially useful in the field of servo motors, since controlling velocity is extremely important (eg. Electric trains, industrial automation) and can be easily measured using optical encoders or tachometers. A formulaic diagram of a generic PIV controller is shown in figure 1.9. A diagram of the related control loop used in the Copley Xenus drive is shown below in figure 1.10. Looking at figure 1.9, we see two new gain factors, K_v and K_t , K_t being the integral gain. K_v is known as the velocity proportional gain, and allows the control loop to react quickly to velocity changes. That is, when one wants to accelerate or decelerate a motor rapidly, K_v should be high. However, leaving this value high for slow movements will cause overshoot. In situations where a steady velocity is required, K_i should be increased, as it will attempt to reduce the velocity error over time.

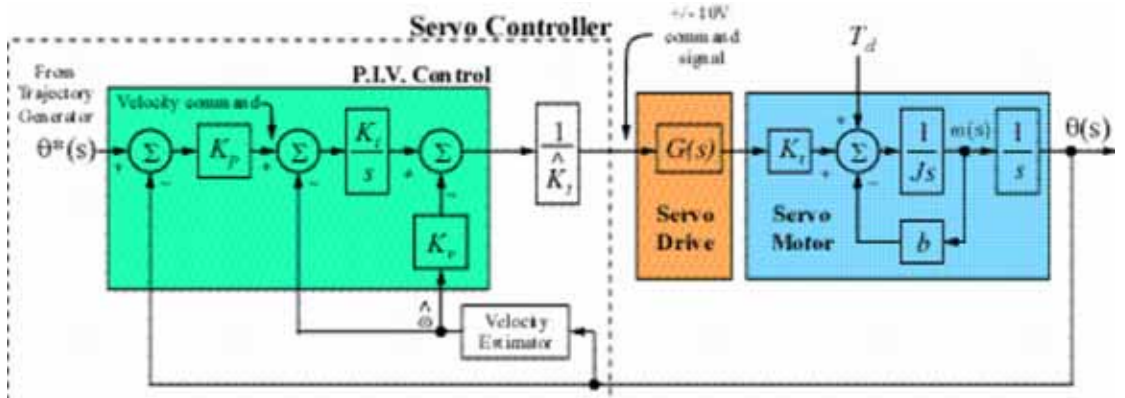


Figure 1.9: PIV control loop [6]

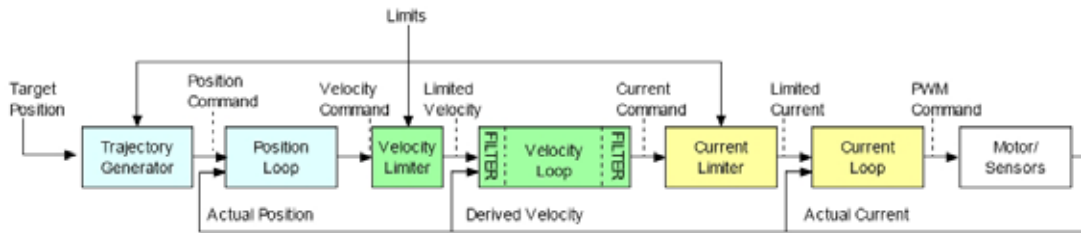


Figure 1.10: Copley Xenus Control Method [7]

1.5.3 Feed Forward Control

In some cases it may be necessary to eliminate following errors or overshoot in servo control systems. In some applications, it doesn't matter if the output control signal lags behind the commanded input, or if the motor overshoots a little before coming to a stop. However, in precision positioning or velocity matching applications, these two issues can become big problems. When a positioning command is issued, the goal is for the gimbal to move to that location as fast as possible and then come to a dead stop. If there is overshoot, this could cause the data transceiver to lose its connection before re-linking successfully, or a camera to initially miss its target. This would also potentially cause the network topology control to malfunction, if it sees nodes intermittently connected. To induce faster response times in the control loop, the most common method is that of feed forward control, which directly couples the commanded velocity and acceleration to the control loop. The main requirement of feedforward control is the availability of acceleration and velocity commands from the motor controller, something usually found only in new units. Use of feed forward control will minimize the tracking error, which is the difference between the commanded and actual velocity. If this error is low, the effective response time of the control loop becomes faster, which in turn decreases the chance of overshoot. When tuning a motor using feed forward control, one adjusts the velocity feed forward gain (V_{ff}) first to achieve the desired overall response time, that is, V_{ff} is usually set as the value that causes about 10% overshoot. This small overhead is needed because the acceleration feed forward will decrease the response time of the velocity feed forward somewhat. Finally, the acceleration feed forward gain (A_{ff}) is adjusted in order to eliminate any overshoot caused by V_{ff} [8]. In the end, feed

forward predicts the motion based off of a predetermined profile, while P.I.V adjusted for unexpected disturbances to that profile. A diagram of a PIV control loop using feed forward is shown in figure 1.11.

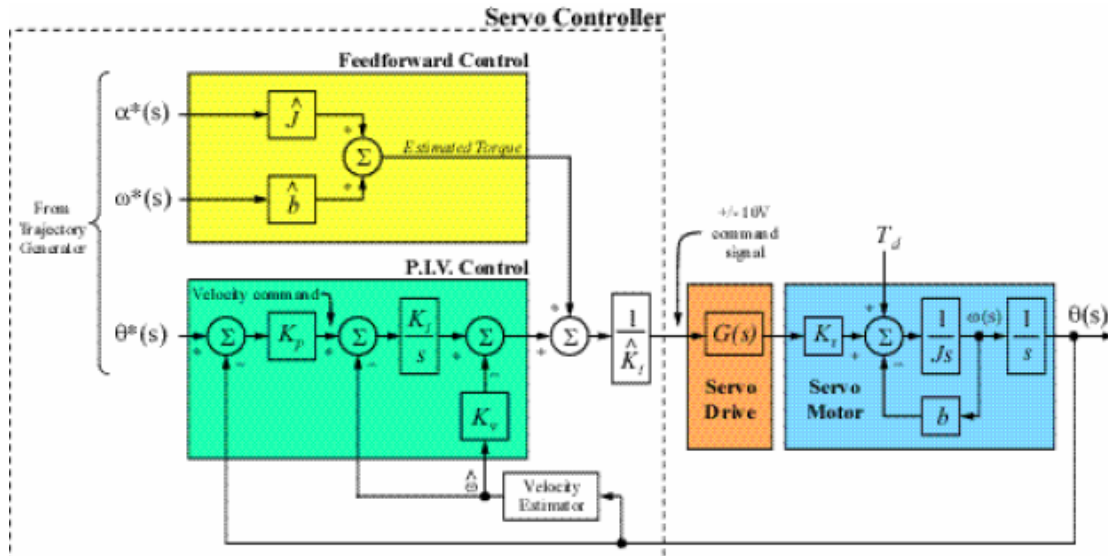


Figure 1.11 Feed Forward Control [7]

1.6 Servos vs. steppers in PAT applications

In our group's research, gimbals are utilized in pointing, acquisition, and tracking applications. For different tasks, different motors are needed. When developing the homography and radial trifocal tensor algorithms for image-based acquisition and pointing, Dr. T-H Ho used a gimbal with micro-stepping motors that have a 0.0072° step resolution. This type of stepper gimbal can point to a location with great accuracy and repeatability, but cannot track moving objects nor change direction mid-flight. In the experiment described later in section 3.5, a servo motor was equipped with a camera to track moving vehicles from a remote location. While it used the same algorithms as in [9], its pointing accuracy was only about 20% as good compared to the stepper motor

gimbal. However, for aiming a camera at large objects such as cars, this did not become an issue. What was more important was its ability to match velocity with an object so that it was always in the FOV of the camera, something which stepper motors cannot do smoothly. A comparison of servos vs. steppers (of similar continuous torque ratings) is shown below in table 1.

	Brushless Servo Motor [1]	Micro Stepping Motor[4]
Continuous torque	2.3 N·m	3.5 N·m
Peak torque	>23 N·m	8.3 N·m
Pulses per revolution	8,192	50,000
Power consumption	590W (continuous use)	380W (continuous use)
Gearing	None (direct-drive)	Harmonic (50:1)
Feedback	Optical encoder and Hall-effect sensors	Optical encoder
Operating modes	Current, velocity, position	Velocity, position

Table 1: Comparison of servo and stepper motors

1.7 Motor Selection

Before deciding what motors to use for the various planned experiments, it was important to select the appropriate qualities needed. First and foremost is the motor's ability to function at slow (<30rpm) velocities with a small tracking error. After this came the requirement of being able to change velocity and/or final position while already in motion. In other words, if a motor is traveling to point A, the controller can send an updated movement profile mid-flight to redirect the motor. In the case of a stepper motor, one must wait until one motion is finished before starting another. Even with fast acceleration and deceleration, this command delay causes jerky motion in stepper motors when used in tracking applications.

Finally, the azimuthal motor must be able to swing a mass of 30kg 180° in under two seconds. This requires a large peak current value, which in experiments was around 35A_{DC}.

After all these considerations were taken into account, the motor selected was a 7” diameter DC brushless servo motor from Bodine-Electric, model number 07-EKEP-00.

A small list of relevant specifications is shown below in table 2.

Maximum Continuous Current	5.5 A _{DC}
Motor Power	590W
Maximum speed	6000 rpm
Maximum continuous speed	2500 rpm
Maximum continuous torque	2.3 N·m
Peak torque	>23 N·m
Motor efficiency	88%
Number of poles	8
Angular resolution (8192 ppr)	0.0439°

Table 2. Specifications of the Bodine E-Torq Motor (adapted from [1])

1.8 Controller Selection

There are a myriad of servo motor controllers on the market today, some of the most popular being from AMC, Copley, Parker Motion, and Oriental Motor. All provide the same basic function: supplying three-phase DC current to the motor in the proper pattern to induce the desired motion in the motor. Most controllers offer different control modes, feedback options, and output current levels, so they are generally chosen for a specific application. In our case, the controllers for this project needed to provide at least these five features: High peak current, optical encoder with milli-radian resolution, feed forward control, trajectory position mode, and the ability to control the motors and gains in real-time through RS232 or CAN.

From these considerations and the recommendations of the motor distributor, the Copley Xenus driver was chosen, as it fulfilled the minimum requirements for our project in addition to its affordability. Figure 1.12 shows the Xenus drive, the XSL-230-36



Figure 1.12: Copley Xenus Servo Drive [8]

One interesting safety feature of this driver is its I^2T algorithm [8], which prevents too much current being supplied to the motor. This algorithm averages the power over a certain time interval, which for our setup was 1.1s. It constantly monitors the motor's actual current draw and checks whether the motor is operating above its continuous current limit. If it is, it increases an internal accumulator, and if it not, the accumulator is decreased, though never becoming negative. If the accumulator goes higher than a certain value determined by the time interval, continuous current limit, and peak current limit, the controller shuts the motor off. The I^2T time constant for this project was set intentionally low to protect the motors and drivers.

Chapter 2: Experimental Design

The central piece of hardware in this paper is a two-axis gimbal which can carry either a directional RF antenna or FSO optical transceiver. The FSO transceiver (a Canon Canobeam DT-110) weighs 8kg and is significantly larger than the RF antenna, so the gimbal was designed with this unit in mind. Most gimbals follow the same design rules, with one motor providing azimuthal rotation and the other elevation rotation. A payload cradle is coupled between the elevation motor and high-speed ball bearing to hold the transceiver. Because the elevation motor contributes a great deal to the moment of inertia of the entire upper section, a counterweight is used to balance the mass of the elevation motor, which sits far out from the center of azimuthal rotation. While the counterweight also adds to the moment of inertia, it helps the motor maintain constant velocity as well as smooth acceleration and deceleration, as the moment of inertia is now balanced. It is worth noting that this is a prototype, not having the great advantage of slip-ring connectors to transfer power and control signals to the elevation motor and payload without the need for cables, which can add uneven forces to the elevation motor, in turn causing errors in the control loop.

2.1 Design Considerations

When designing a gimbal, one must take several issues into consideration. First is weight; namely, the maximum payload mass which will determine the maximum acceleration and deceleration values that can be achieved without inducing instability. This worst-case approach works well when tuning servo motors, as the controller can easily provide less current for a smaller moment of inertia. If one tunes the motor for a lower inertial payload and then puts something heavier on, the motors will perform unevenly, and in some cases even cause the controller to shut off.

The second consideration is that of the maximum velocity. For a device that rotates always at a constant velocity, how well it accelerates is not really an issue. All that matters with a conveyor belt is that it moves steadily, not how fast it spins up in the morning. However, when the motor makes rapid starts and stops, velocity stability is essential in creating a smooth motion profile. Therefore, the velocity feedback loop takes prominence in the design. A diagram of the velocity loop used in the Xenus amplifier is shown below in figure 2.1.

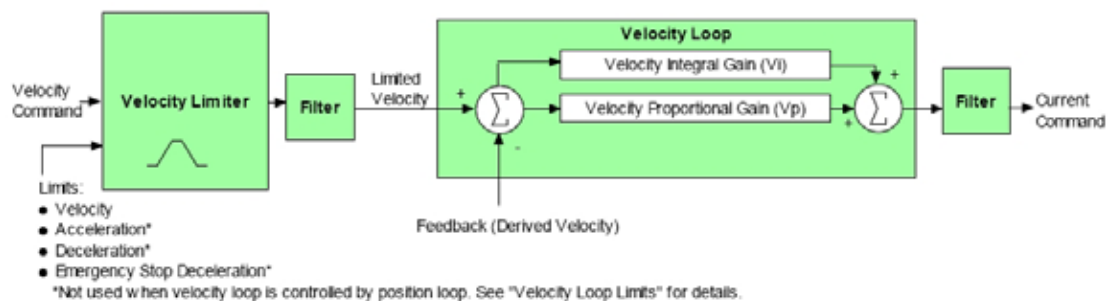


Figure 2.1: Velocity loop filter [7]

Lastly, one must decide whether stopping on a dime is more important than speed. With such a large moment of inertia, it is quite difficult to make a gimbal stop suddenly with no overshoot. This can be eliminated by moving slower, but this can become

undesirable in certain situations that will be discussed in chapter 3. However, in practical situations seen in experiments with directional RF antennas, some overshoot can be tolerated, as the beam is wide enough to still cover the receiver even if there is some misalignment for a short period. Figure 2.2 below shows a comparison of movement profiles of the same step count, part a) moving fast enough to cause overshoot, while b) moves slower.

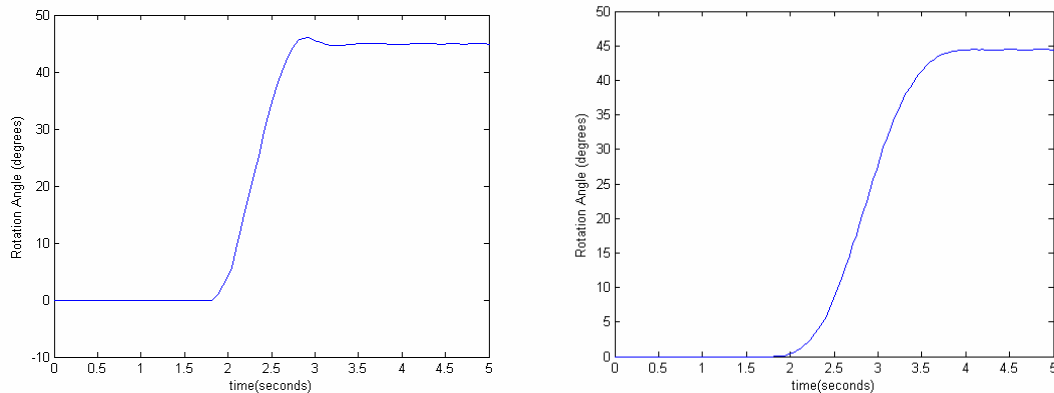


Figure 2.2: Rotation with and without overshoot

Naturally, one must also think about things like power, reliability, rigidity, weight, cabling vs. slings, and many other issues when designing such a complicated system. Since this is a prototype to test various PAT concepts, the above concerns will be left to the next generation of gimbals.

2.2 Design specifications

As mentioned briefly in Chapter 1, the specifications for this gimbal are based on its usage as a platform for rotating a directional transceiver. Another factor was the ability for the gimbal to be used as a platform to compensate for large platform motions, say from the rocking of a ship. This way, an FSO transceiver could use rapid fine-angle pointing, acquisition, and tracking to lock on and follow another transceiver. A prototype FPAT system utilizing piezo actuators has already been designed and tested in [9].

Table 3 below shows the target specifications for the gimbal.

Time to rotate 180° azimuthally	~1.5s
Field of Regard	3π Steradians
Payload Capacity	10kg
Angular resolution	$<0.5^\circ$
Weight (with max. payload)	Less than 45kg

Table 3: Gimbal Design criteria

2.3 PIV Control Loop Gains

To achieve the above specifications, robust tuning of the motor's controller is required. This is a complicated process, exacerbated by the fact any change in the weight, size, or moment of inertia will necessitate a retuning. However, there is a well-established method for tuning the PIV controller, which is of the form current loop => velocity loop => position loop. In total, there are eight gain factors that must be adjusted, and also the velocity filter cutoff frequency. Figure 2.3 shows a standard flow chart for tuning the velocity loop of a PIV controller [10]. After one has tuned the velocity loop, the position loop can be tuned, usually by commanding a step movement, viewing the motor response, and adjusting the position proportional gain, feed forward gains and gain multiplier[11].

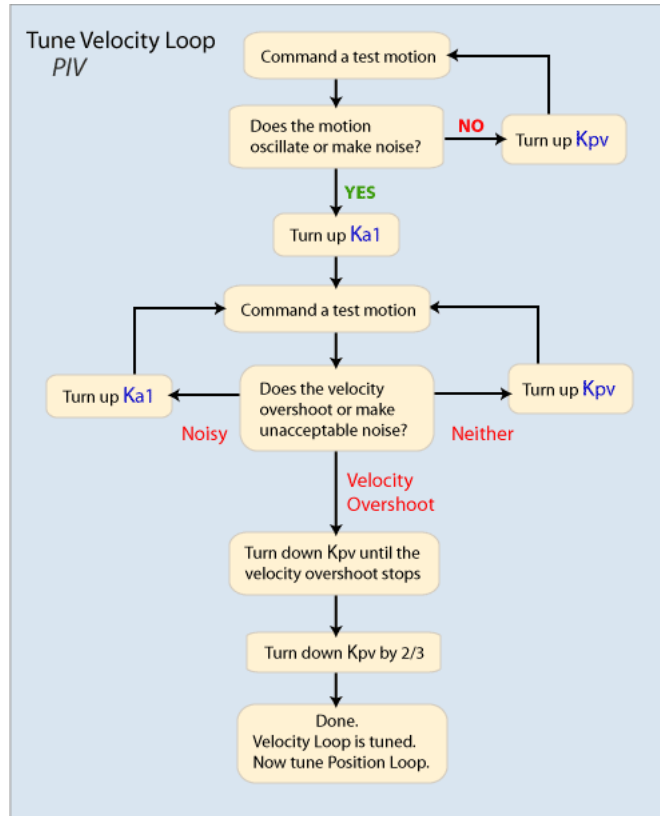


Figure 2.3: PIV tuning method [11]

After this process has been completed, the cutoff frequency of the velocity loop filter must be adjusted. It essentially tells the motor feedback to ignore certain amounts of noise. If this value is set too high, the controller will try to compensate for motion that is in fact not there. When this happens, the motor will begin to oscillate or make a grinding noise as it rapidly changes current polarity in an attempt to compensate for the perceived noise. In our system, the appropriate value was found to be 3Hz, using a single-pole 20db/dec low-pass filter.

With regard to the other eight loop gains, table four below explains the effect each of them has on the system.

Gain Type	Associated Effect
Current Proportional	Allows for faster current changes
Current Integral	Smooths output current, reduces steady state error
Velocity Proportional	Allows for faster velocity changes
Velocity Integral	Damps out velocity overshoot, reduces steady state error
Position proportional	Determines how fast motor tries to get to a position, too high a value causes overshoot
Velocity feed forward	Decreases following error during constant velocity moves
Acceleration feed forward	Reduces overshoot caused by using Velocity Feed Forward
Gain multiplier	Affects overall response time of the position loop. Too high a value causes oscillations when settling

Table 4: Loop gains and their associated effects

When one tunes a motor, it assumes the motor will only be used in that way (same move distance, speed, acceleration, etc). Ideally, the underlying current loop should be controlled by a DSP which actively monitors the motor's position, velocity, and acceleration, and then dynamically adjusts the current. This is essentially the same as adjusting the gain values continuously. However, for the types of movements described in this paper, the two main gains that need to be adjusted are V_p and P_p , which are changed adaptively through serial port control based on the acceleration value selected. To find the best values, movements were commanded using various acceleration values, and V_p and P_p were adjusted to achieve the best performance. These values were then curve-fitted to a linear slope, which was in turn used as a calibration for adjusting the gains for any other motion. The velocity value was set based on the rotation angle, in order to ensure a triangular profile (i.e, always accelerating or decelerating). For the

surveillance applications described later in 3.5, the gains and velocities were adjusted dynamically, but were limited to 10Hz by the motion detection program.

2.4 Motor Control through RS232

While there are several different control schemes available for the controllers used in this project, the simplest to employ rapidly is the RS232 serial command interface. Through a series of ASCII commands sent to the controller, the user can operate the motors in current, velocity, or position modes, as well as being able to change the loop gains. In some ways it mimics what a DSP would do, but at a far slower rate. For these experiments, all control came from a custom-designed Labview interface, using the ASCII commands from Copley. In experiments, movement commands could be sent at about 20Hz. This is adequate for the surveillance tracking applications described later, but it is not enough for rapid stabilization experiments. There is also an inherent serial communications delay of 5ms, so in the case of the gyro test in section 3.5, a delay can be seen between the gyro sensors and the motors' movement. For a general positioning application, figure 2.4 shows the command sequence to complete a motor movement. The section boxed in blue must be resent for each new command, assuming the values for velocity, acceleration/deceleration, and move distance have changed.

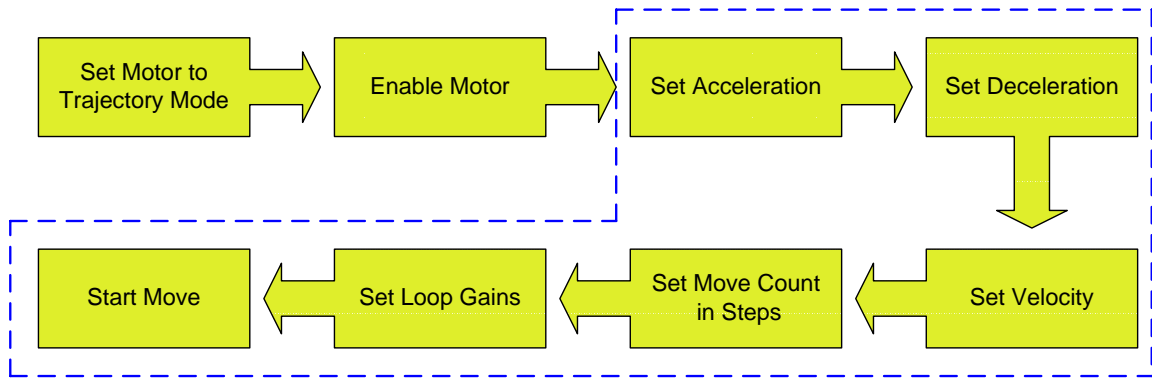


Figure 2.4: Command sequence for a position-trajectory move

2.5 Mechanical Design

Most gimbals in use today use a basic U-shaped structure, with the azimuthal motor underneath, and the payload cradle supported by the elevation motor and an opposing bearing on top. There are many small variations on this, but the author's design was simple enough to be made in the IREAP machine shop. The motor controllers were attached to the azimuthal motor support to allow for short cabling, and to make the unit as compact as possible. The overall dimensions of the finished FSO gimbal are 71.12x48.26x45.72cm. Weight reduction was achieved by using mostly aluminum, and slotting out parts of the structure. A brass counterweight was added opposite the elevation motor in order to provide a balanced moment of inertia. Even though this added mass slightly decreased the maximum velocity of the azimuthal motor, the balanced load greatly improved positioning performance and velocity stability, not to mention simplifying the tuning of the PIV loop. The motors were operated in a direct drive configuration, due to the large moment of inertia involved. While using gearing could give even better low-speed performance, the gears would have to be specially designed to handle the large peak torques generated by the azimuthal motor. To eliminate limit

cycling, a sealed ball bearing was used for the elevation stage, while a custom copper-backed bushing was used for the azimuthal motor. This design was employed because it allowed the static friction to be adjusted in order to provide just as much friction needed to stop the steady state oscillatory behavior. Figure 2.4 below shows a schematic of the adjustable friction bushing. Figures 2.5 and 2.6 show the author's conceptual design of the unit holding a FSO transceiver and RF antenna, respectively, while figures 2.7 and 2.8 show the actual units holding a FSO transceiver and RF antenna, respectively.

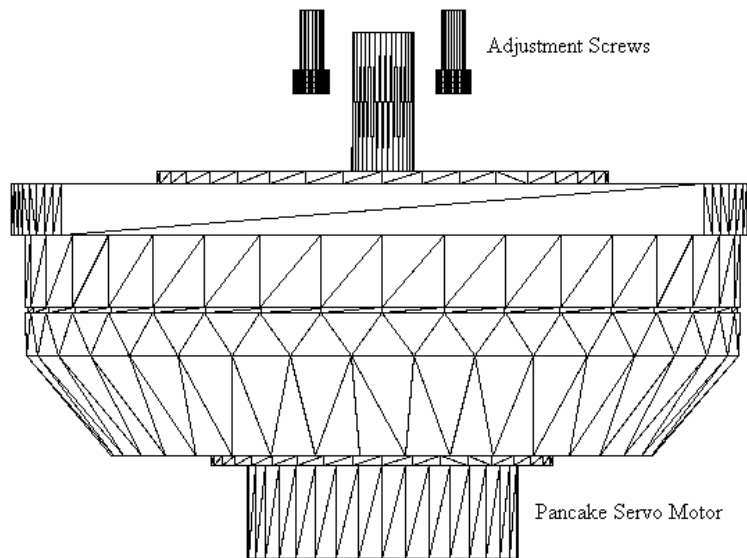
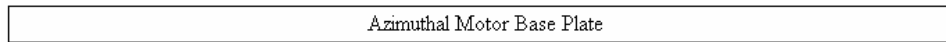
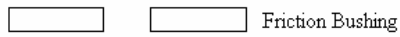
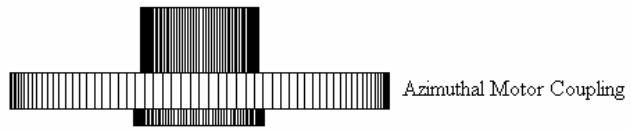
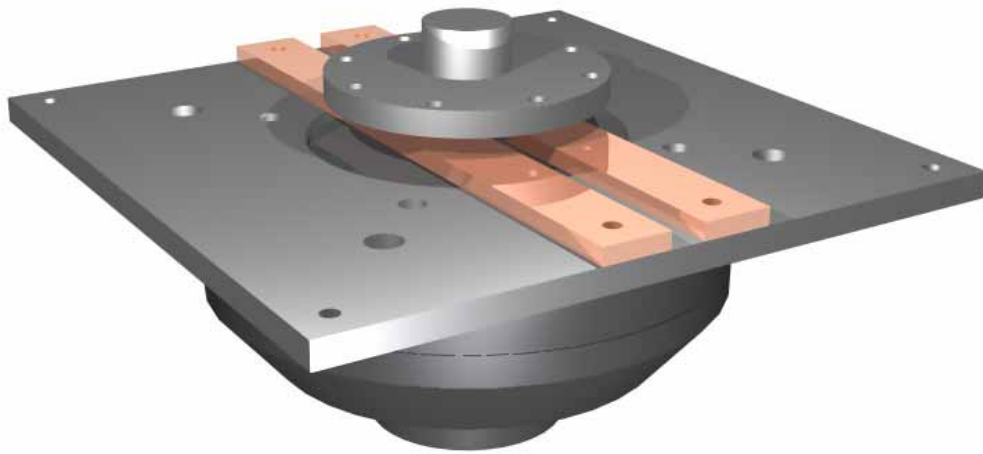


Figure 2.4: Adjustable Friction Bushing

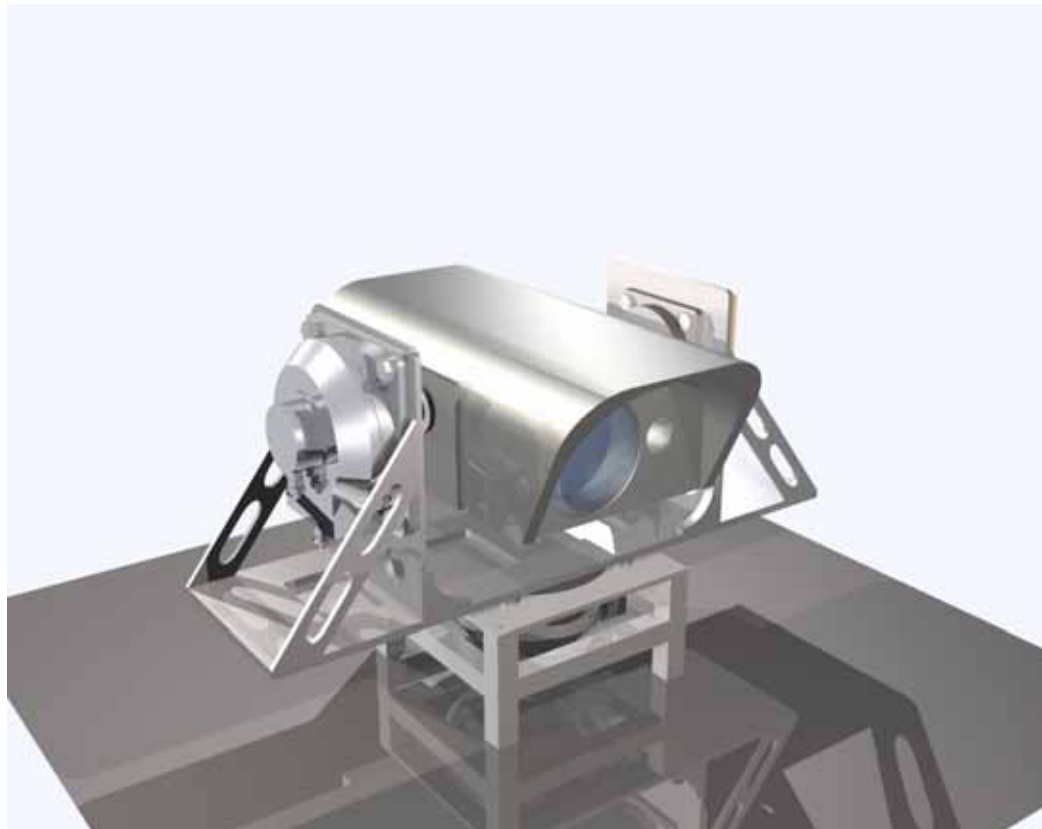


Figure 2.5: Conceptual Design of gimbal holding a Canobeam



Figure 2.6: Finished Canobeam Gimbal

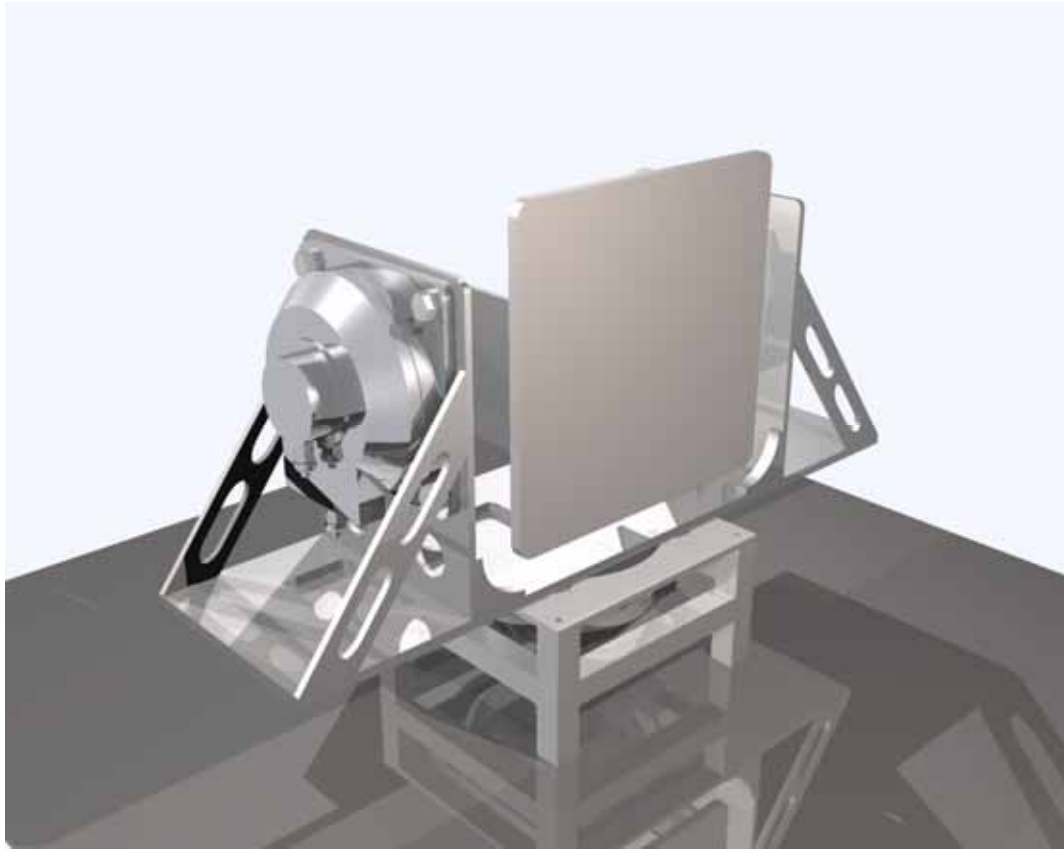


Figure 2.7: Conceptual design of gimbal holding a RF antenna



Figure 2.8: Finished RAD antenna gimbal

Chapter 3: Experimental Results

3.1 Introduction

After all construction was complete, the end result was a pair of gimbals, each capable of holding either a RAD RF antenna [11] or a Canobeam FSO transceiver [12]. Before using these units to test the gimbals' performance as part of a reconfigurable wireless network, it is essential to see how they perform in basic positioning tasks, measured in terms of movement times and positioning error. Next, the gimbals were evaluated as communications nodes, measuring the average data rate during a make-break-make test, which will be described later. Another experiment was also performed, in which the gimbals' capability as a gross-motion stabilization platform was evaluated. Finally, a small demonstration of object tracking using automatic speed control was conducted. In the latter two experiments, the main limiting factor was the use of serial port commands to control the motors, which induced a 60ms delay (the total delay from all the commands seen in figure 2.4). This prevented the gimbals from quickly compensating for rapid changes in platform orientation, and also caused some unevenness when tracking moving objects. The use of DSP would drastically reduce the delay and command time, however the amount of development time was not feasible for the scope of this phase of the project. In a mature system, the encoder feedback would be directed straight into the DSP (which would also calculate the velocity and acceleration estimates), thus eliminating the various delays incurred from interfacing with the motor controller.

3.2 Angular Rotation Time and Positioning Error

The first performance test involved moving the gimbal as rapidly as possible in both elevation and azimuthal directions, and monitoring how long the total move took. In the case of the azimuthal motor, the movement range was tested between 0 and 180°, and -30° to 30° for the elevation, both limited by the mechanics of the system. Figure 3.1 shows the results for both motors, with each rotation angle tested ten times, with the Canobeam as a payload. Figure 3.2 show the standard deviation of the final position, showing how well the motors can move to a commanded position, averaged over 10 runs. The gains for the amplifier were dynamically adjusted in this test to maximize speed and minimize overshoot.

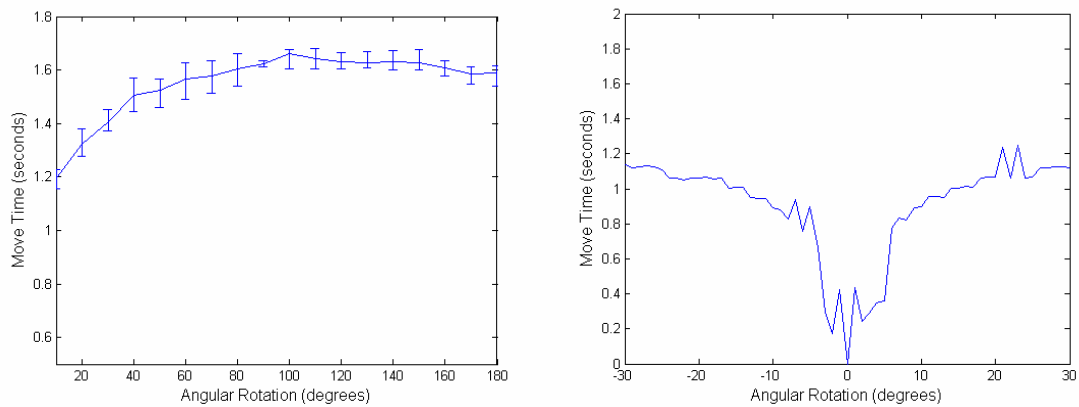


Figure 3.1: Angular move vs. time, azimuth motor on left, elevation on right

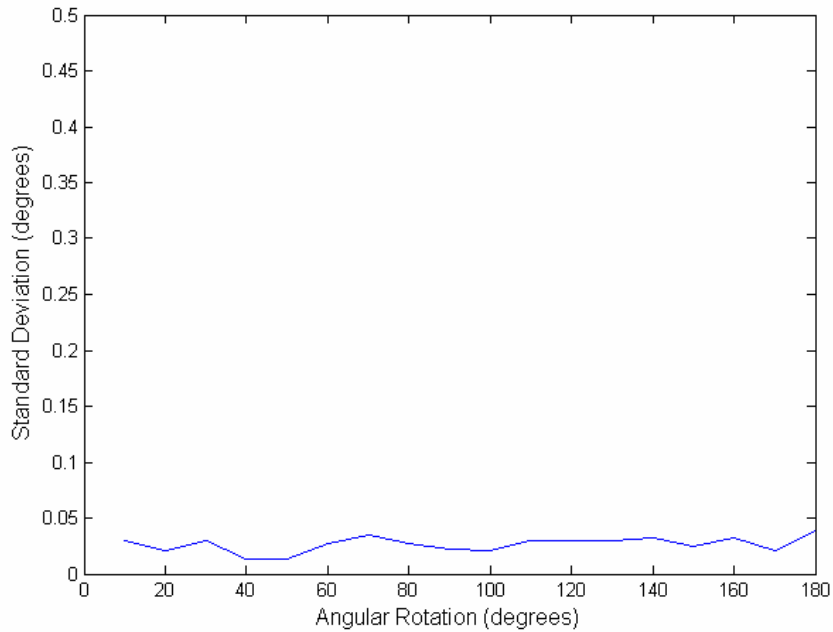


Figure 3.2: Standard Deviation of the difference between commanded and steady state angular rotations, azimuth motor

Looking at the above graphs, a few interesting observations can be made. First, the move times for the azimuthal motor get faster above a certain angle, due to the momentum of the gimbal once it gets going. As for the elevation motor, the move times increase with the rotational angle as expected, however for very small angles, it also increases. This is primarily due to the static friction of the ball bearing on the elevation motor assembly. Looking at the graphs of the position standard deviation, the azimuthal motor performance remains nearly flat across the movement range, and shows a very small ($<0.05^\circ$) deviation. No graph was included for the elevation motor, as the standard deviation of the angular rotation (defined as the difference between the commanded and steady state position) was 0 over the entire angular range. Since the elevation motor has much less inertia to deal with, the position loop was able to converge to the command value.

3.3 Reconfigurable Network Experiments

The test setup for the experiments in this section was located on the roof of the Kim building, with the remote node on Martin Hall, 254m away. Figure 3.3 below shows a diagram of the test bed and hardware. Two transceivers were tested, a Canon Canobeam FSO unit [13], which uses a 7mW laser operating at 765nm and has an effective range of 500m, and a RAD Airmux-200 Radio unit [12], which operates at 5.81GHz and has an effective range of 3 miles at its full data rate using a 30cm x 30cm patch antenna. Both of these units are designed for fixed outdoor links, to the following experiments also tested whether they could function in a morphing physical network.

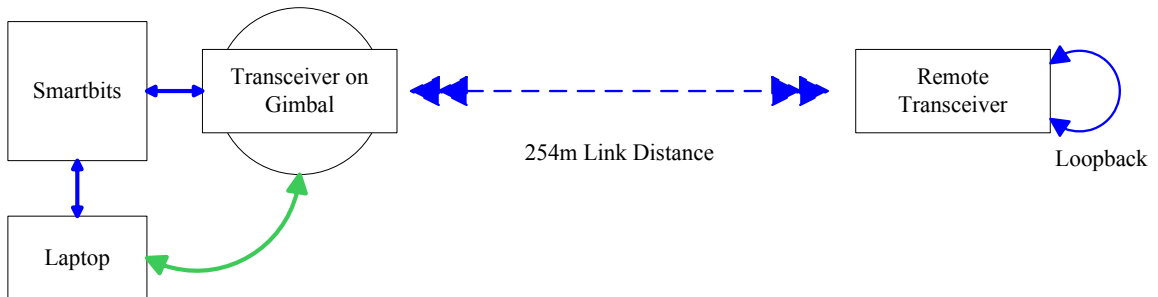


Figure 3.3: Block diagram of experimental setup

3.3.1 Transceiver Angular Offset vs. Packet Loss

Each of the transceivers described in this thesis has a certain angular mismatch it can tolerate between two nodes. This is very important to quantify, as it can reveal how high the pointing resolution must be for the motors, as well as the unit's actual beamwidth. To begin with, each of transceivers was aligned manually to achieve the strongest signal strength, based on the manufacturer's alignment procedure. This

position was then set as the starting location. The azimuthal motor was then rotated incrementally until the signal was lost, with a Smartbits test rack recording the average data rate over a one minute interval at each angle. The same procedure was then repeated for the elevation motor, with the azimuthal motor recentered. Figures 3.3 and 3.4 below show the results for both transceivers.

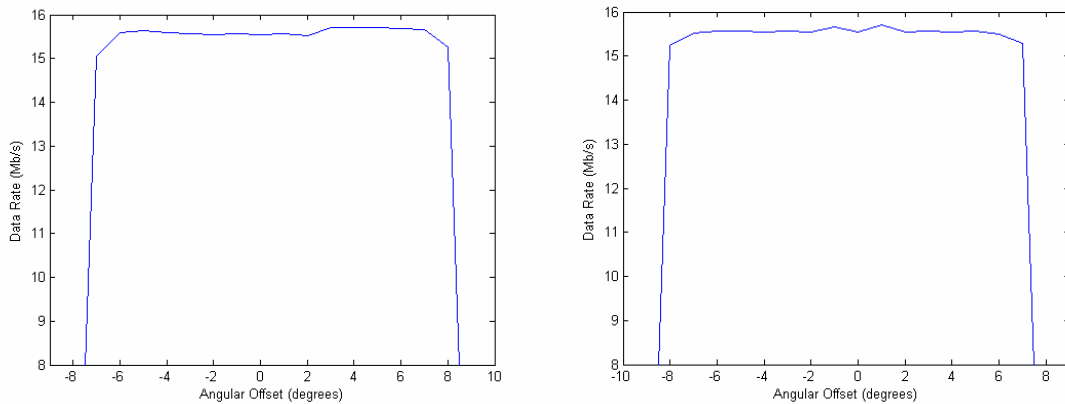


Figure 3.4: Angular offset vs. average data rate (RAD), azimuthal and elevation

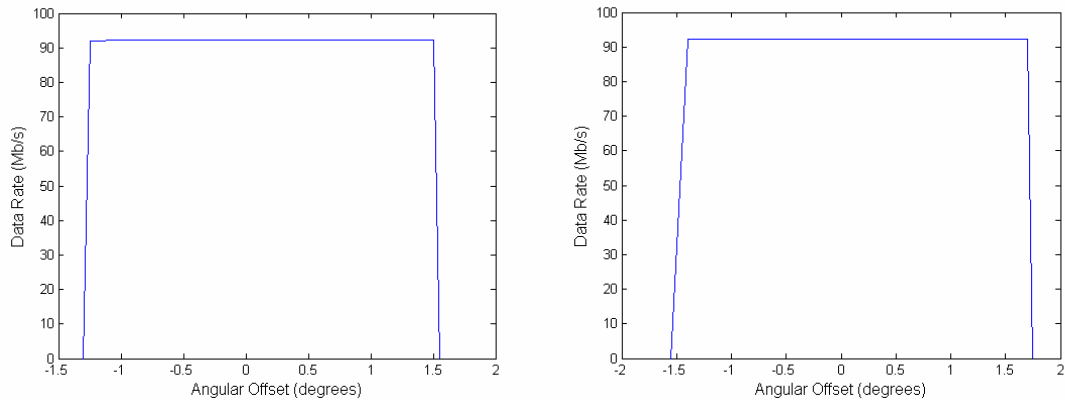


Figure 3.5: Angular offset vs. average data rate (Canobeam), azimuthal and elevation

We can make several observations from the graphs above. The effective beamwidth of the RAD antenna is about 15°, compared to the specification of 9°, which was based off a measurement along the antenna’s diagonal. These radios also display a sharp cutoff in usable data rate; that is, there is a critical angle where the data rate goes

almost immediately to 0 Mb/s. Also, in order for the RAD antenna to achieve its full capacity, all 24 sub-channels must be synchronized. As the signal strength fades, these sub-channels start to fail, leading to an immediate drop off in data rate.

As for the graphs from the Canobeam, the first thing that can be noticed is the flatness of the data rate in the usable region. Not surprisingly, the use of a single wavelength to transmit data leads to a tighter received signal, and thus a steadier data rate. The Canobeam was found to have a 2.8° horizontal and 3.1° vertical beamwidth, with the signal fading to nothing within 0.05° past these limits. These numbers are much wider than the “ $<1^\circ$ ” beamwidth specified by Canon, however the autotracking feature was enabled in this test, which can compensate for $\pm 1.2^\circ$ of angular mismatch. This is actually a very useful feature, since it makes the initial alignment much easier. It also acts like a stabilizer for very slow platform motions. It is feasible that this feature could be used on a ship to compensate for some of the very low frequency oscillations (i.e. a ship in calm waters), thus saving gimbal motor power. One interesting observation from the next section was that the gimbal rotated fast enough so that the autotracking did not begin searching for the beam, implying that its response time is greater than ~ 2.5 s.

3.3.2 Make-Break-Make Tests

The end result of all these tests was to optimize the gimbals so that they could function in a reconfigurable network. To that end, they must be able to rotate to a position fast enough that the network can relink in a time so that little data is lost. Of course, ‘little data’ is a relative term given buffering and other tricks, but for our purposes we used the Smartbits packet generator to test the links unbuffered. The idea is to simulate

the gimbal linking to a node that would be the farthest away (180° from the starting position). To do this properly would require three transceivers of each type, which we did not have. To bypass this, we moved the gimbal up to 90° away and then 90° back to the original position. This method will result in slower relink times due to the pair of accelerations and decelerations, but it is reasonable in obtaining a measure of the worst-case relink time. The test methodology is as follows: packets are sent from the Smartbits at the fixed location and then looped back at the mobile node. The link is then broken in the above manner while the data rate is measured. This is then repeated 10 times for each rotation angle. Since each of these units operates at different data rates and use different modulation schemes, it would be improper to directly compare their performance. Also, these units are designed for different areas, so a group using a FSO transceiver wouldn't necessarily want to use a radio unit. It is therefore advisable to look at these tests independently, given the units' fundamental differences.

For both experiments, the test range was between Martin Hall and the Kim Building, which is a distance of 254m. Tests were performed on a clear day with winds less than 10kph. The make-break-make tests were conducted by rotating the azimuthal motor, since in the case of the RAD antenna the vertical polarization must remain the same, and for the Canobeam because the unit is too big to do a vertical flip, so neither of these situations is likely to occur in real topologies.

3.3.2.1 RAD RF Antenna Results

For the RAD antenna units, the radios were first manually aligned to find the maximum signal between the two. After this was set as the zero position, the unit was rotated azimuthally in the manner prescribed above, and the data rate was recorded. When used at its maximum signaling rate, the RAD antenna uses 24 sub-channels to transmit and receive data. Each of these takes at most 300ms to reconnect, so if they all do not re-sync on the first pass, the system must try again for the link to be reestablished. Because of this, reconnects can take up to eight seconds if all the subchannels connect on the first pass. Figure 3.5 below shows the results of the make-break-make test, plotting total angular move (away and back) vs. reconnect time (at 15.5Mbps, slightly below the maximum capacity of the unit, in order to allow some cushion). Figure 3.6 shows a graph of time vs. data rate for a typical move of 180°, as well as the movement of the motor. In figure 3.6, the motor doesn't actually accelerate as sharply as shown, this was a data logging issue caused by the program controlling the motors. The position looks like a steep triangle if recorded properly.

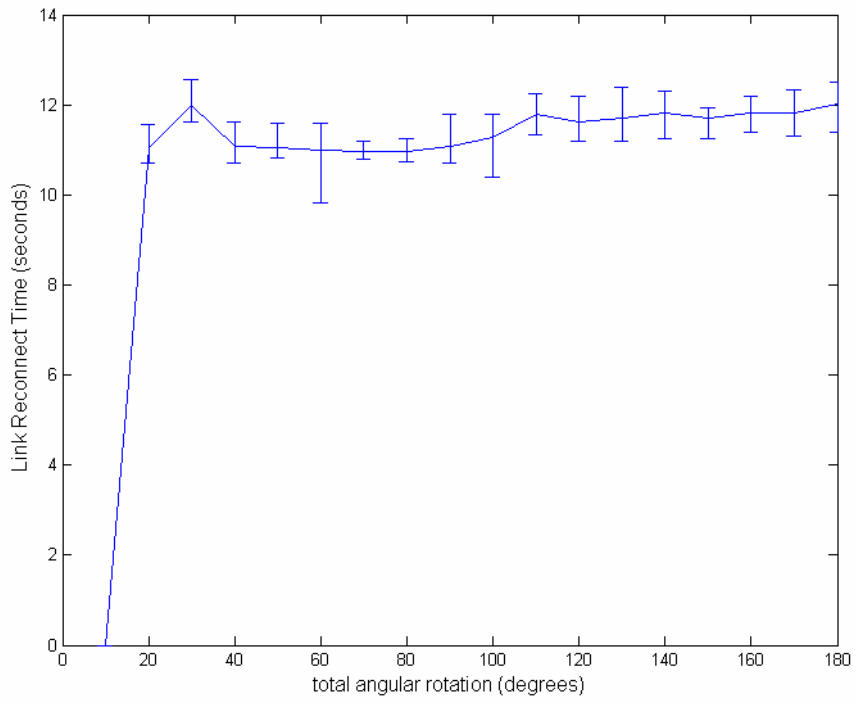


Figure 3.6: Angular move vs. Reconnect time

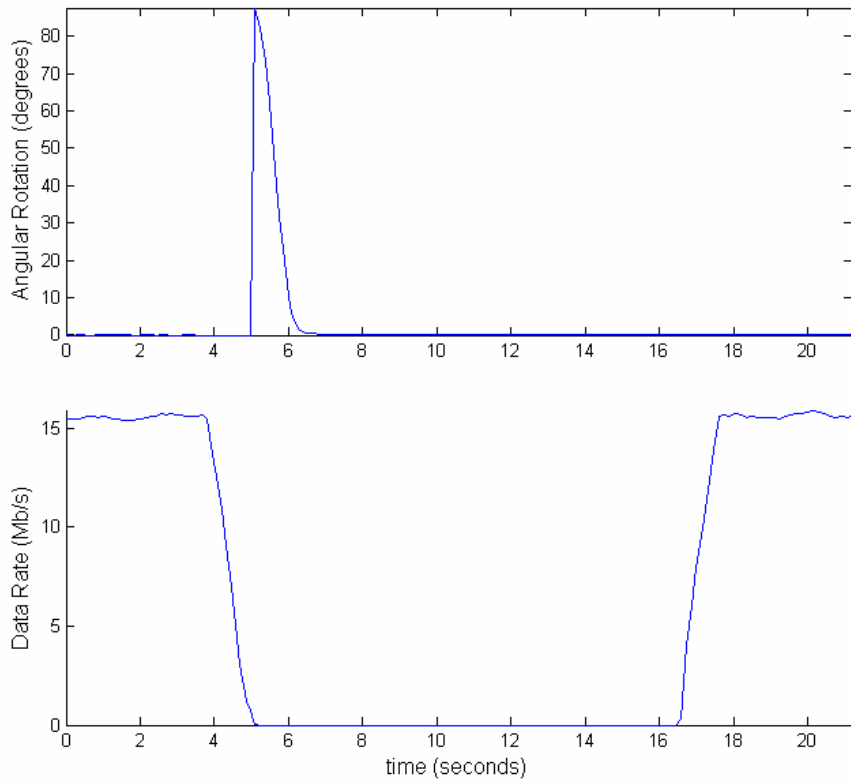


Figure 3.7: Time vs. data rate and motor rotation for a 180° move

Looking at the previous plots, one can note several interesting properties of the radios. First is that the relink time is almost unrelated to the time that the motor takes to move away and back. Of course, for a longer mechanical break the reconnect time would be longer, but these tests show that the gimbals moves fast enough at all angles that it doesn't contribute significantly to the downtime. The manufacturer stated that the maximum reconnect time is around 8 seconds, however in these experiments it was found to be around 11.45 seconds. This disparity is most likely from the sub-channels not all disconnecting and then reconnecting at the same time, somewhat akin to pulling taffy apart. If the radio cannot reconnect all the sub-channels at once, it rescans them all again, which leads to a longer overall reconnect time. Given that the make-break-make times for the motor ranged from 1.1s (10° rotation) to 2.9s (180° rotation), the gimbal movement represents between 9.6% and 25.3% of the link reconnect time. It should also be noted that for rotations less than 10°, the radio did not lose its connection at all. The use of lower frequency directional RF transceivers offers a significant saving in terms of cost and alignment difficulty, but at the cost of reconnect time. However, these units were never intended to be used in situations involving rapid reconnects, so slow re-linking times are somewhat to be expected. Utilizing a higher frequency radios would speed up reconnect times significantly, as they only use two channels, one for Rx and one for Tx.

3.3.2.2 Canobeam Results

While the Canobeam doesn't have the multi-channel connection issues of the RAD unit, it does have a much tighter beam, which can make any motor overshoot or beam misalignment a problem. However, when swinging such a large mass around, there is a definite tradeoff between speed and settling time. Figure 3.7 below shows the results of the make-break-make test, plotting angular move vs. reconnect time (at 95 Mbps, near the maximum single channel capacity of the unit). Figure 3.8 shows a graph of time vs. data rate for a typical move of 180°.

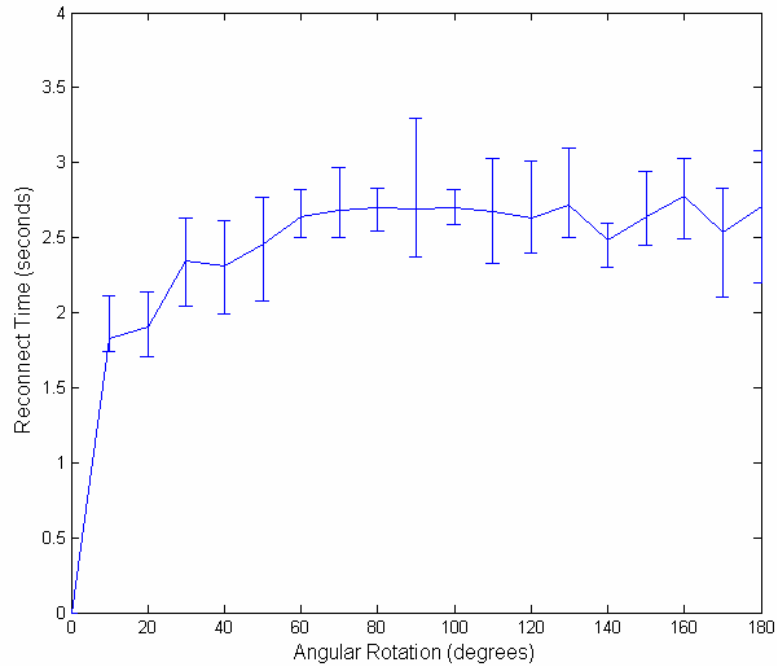


Figure 3.8: Angular move vs. reconnect time

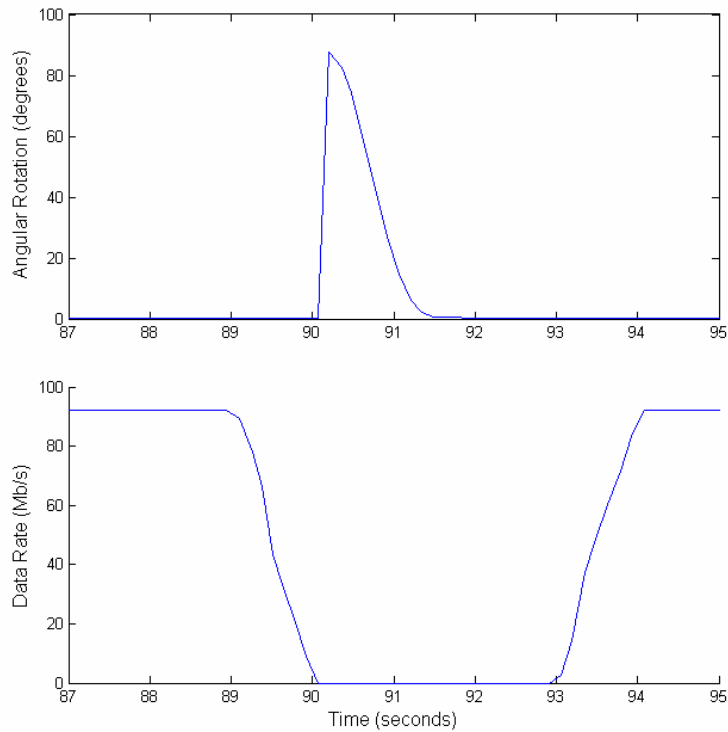


Figure 3.9: Time vs. data rate and motor rotation for a 180° move

Looking at the results from the Canobeam, we can see several striking differences from those of the RAD antenna. First, is that the reconnect times do increase as a function of the angular rotation before leveling off. Because the reconnect time of the Canobeam itself is relatively short (~1.5s), the mechanical rotation time does become a major factor in the overall reconnect time. This can be seen in figure 3.8, which shows that the gimbal's movement for a 180° break accounts for about half of the link's downtime. Thus, in applications involving the reconfiguration of FSO nodes, the mechanical rotation time is a dominant factor in reconnect delay. When using FSO transceivers, the received data rate is very sensitive to platform jitter, and because of the lack of buffering inside the unit, the data rate can drop dramatically for short periods of time when the platform moves by even less than a degree.

3.4 Gross Motion Stabilization

Another part of this project was the requirement that these gimbals could operate as stabilization platforms, in addition to being transceiver pointing devices. If, for example, two ships wished to communicate using directional wireless transceivers, the rocking of the ships would need to be compensated for in order to give the transceivers a stable platform to work from. In modern aircraft, a gyroscope is used to collect angular velocity measurements and then adjust the control surfaces appropriately to produce a stable airplane. In this experiment, a 6 DOF gyro from O-Navi was used, with ± 300 °/s angular velocity sensitivity [13]. The basic premise for these tests was how well the motors could match a commanded waveform, which is essentially what it would have to do to compensate for platform motion.

The test was a somewhat realistic setup, where the gimbal had to move in the opposite velocity to a quasi-random input waveform, courtesy of the previously mentioned gyro. The gyro was moved by hand, and the gimbal had to compensate for it. The ultimate test would be to mount the gimbal and gyro on either a hydraulic motion platform or a boat, which would provide adequate large-amplitude motions for the system to compensate for. The signals from the elevation motor and gyro Y sensor may look jerkier than those of the azimuthal motor, however both have the same amount of jitter, the graphs are merely scaled differently to show the movements better.

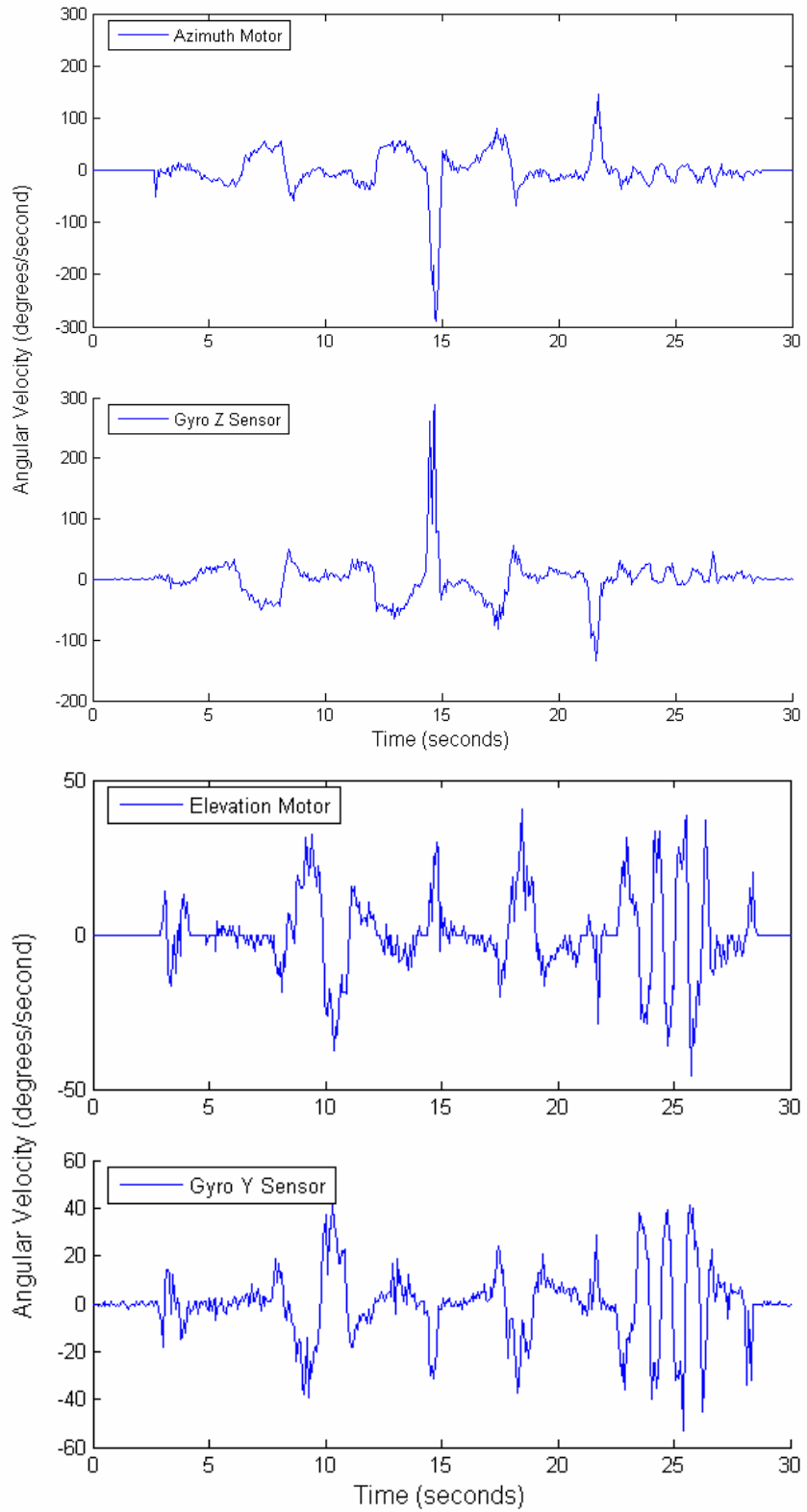


Figure 3.10: Azimuthal/Elevation motor velocity and gyro velocity

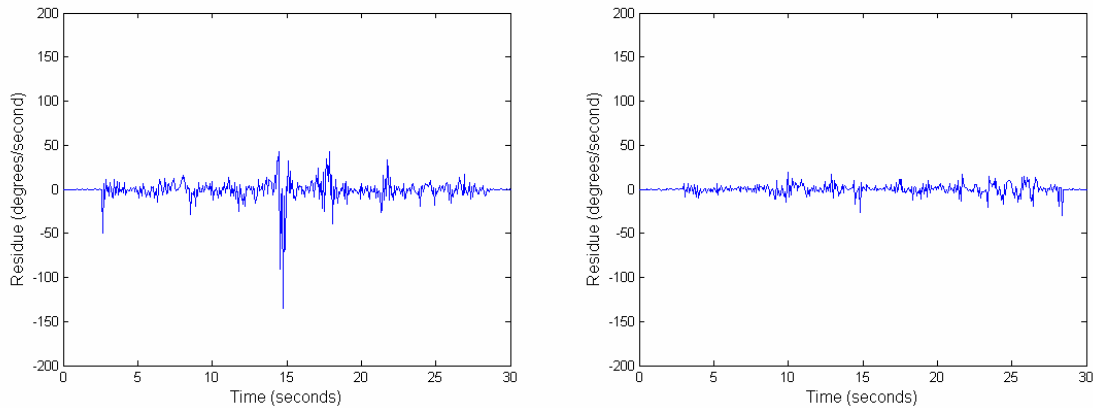


Figure 3.11: Residue of Motor and Gyro Angular Velocities, Azimuth on left, Elevation on right

The residues of the velocities (defined as the sum of the motor and gyro velocities) show that both motors are able to match the gyro's output well, with the elevation motor performing exceptionally well, due to its lighter payload. Some of the error seen above arose from uneven sampling due to the serial port communications delay. As mentioned earlier, the average delay for each command sequence is 60ms, but this can vary by ± 10 ms. The large spike in the residue of the azimuthal motor was a result of jerking the gyro to its sensor's limit, but the motor was not able to accelerate quickly enough with such a large load attached. However, it is unlikely that a stabilization platform would have to accelerate to $150^\circ/\text{s}$ over 600ms.

From this small test of the motors and controllers, it is apparent that they have the capacity to function as part of a stabilization platform. The use of an analog gyroscope with its output connected directly to a DSP controlling the motors would form the basis for a high performance stabilized platform. Preliminary tests from a colleague have shown the communications delay from a similar system using the CAN bus to be around 1ms, so digital control is still possible in the same fashion as the RS232 here.

3.5 Surveillance Applications

Another area of research in the Maryland Optics Group is autonomous remote real-time surveillance systems. With this, a camera can analyze a scene and determine if motion is present, whether the object is a person, car, truck, etc..., if an object was abandoned, and all kinds of other processes. Currently, a HDTV camera is mounted on a stepper motor gimbal to provide zoomed-in follow-up on events. However, these motors cannot provide smooth tracking of moving objects, which leads to blurry video and the inability to get clean image data. Stepper motors generate a current profile before actually moving, so when in motion the movement cannot be changed. In contrast, certain servo controllers can operate in a trajectory position mode [7]. This mode also creates a current profile to move the motor, but it can be changed at any time during the move, which, when tracking moving objects, allows the motor to act like it's in velocity mode, but keep its precise positioning capability. In this experiment, one of the servo gimbals described in this paper was fitted with a HDTV camera and told to automatically follow the first person that entered the FOV of a fixed camera next to the gimbal. A still image from the video is shown in figure 3.12 as an example of the image quality available. A video of the gimbal in action can be seen at [15]. This application is particularly encouraging because of the potential for multiple camera-mounted gimbals to track the same object over a long range, handing off from camera to camera, regardless of the movement behavior of the object. While this has yet to be fully implemented, it is of high interest to the campus police department.



Figure 3.12: Still capture from autonomous tracking experiment

Looking at the above screen capture from [15], the image is quite clear, even though it is from a 24fps camera outputting compressed video while the camera itself is in motion. Several license plates can be readily identified from $>50\text{m}$ away without image enhancement. In viewing the video, we see that the gimbal tracks the person smoothly, with the exception of a few spots. The motor receives movement commands from the motion detection program, which will take longer between frames when there are more objects in the scene, so the motor will then receive unevenly spaced commands, leading to changes in the velocity. This can be mitigated by intentionally slowing down the motion detection program to a period that encompasses most detection times. The gimbal will then move more smoothly, but at the expense of being able to follow objects that are rapidly changing direction. However, cars and people generally do not change direction very rapidly or move so far that they would fall out of the FOV of the camera, given the motion detection program's update rate ($\sim 10\text{fps}$).

Chapter 4: Future Work

While the experiments and devices described in this thesis are varied, there is much left to be worked on. Since they are only prototypes, certain areas were not developed because of time and cost constraints. The system described in this thesis has the definite potential to become a robust centerpiece of a reconfigurable wireless network, provided several improvements are made.

One of the first improvements would be the use of a higher resolution optical encoder which would increase the precision of positioning moves. Encoders of up to 20,000ppr are available, but again at a higher cost. While not necessary for RF applications, these would be essential for positioning FSO transceivers at long distances, or for stabilizing rapidly moving platforms. The use of higher resolution encoders can also mitigate limit cycling, where the motor oscillates between encoder pulses while at a stand still. Since the pancake motors used in this paper are not mass-produced units, upgrades like the ones above will add time to the production of a motor, the ones we used taking five weeks to construct.

Another upgrade that would be very time consuming, but not very expensive, would be controlling the motors using a DSP chip. The Xenus controllers currently allow for full motor control using analog voltage inputs. While not critical for discrete positioning, the use of a DSP would vastly improve performance in real-time applications like object tracking and platform stabilization, the latter of which is described next.

One of the areas of interest in FSO is the use of fine angle pointing and tracking, which would use micro positioning devices to control a laser beam. These allow for precise angular control of an outgoing laser, being able to compensate for platform jitter

or other small and rapid disturbances. The disadvantage of these FPAT systems is that their overall angular range is very small, on the order $\pm 1.5^\circ$ [9]. So if the whole system needs to compensate for more than that, the FPAT couldn't work. To mitigate this, a FPAT unit can be mounted onto one of the gimbals found in this paper, also known as a CPAT, or coarse angle pointing and tracking unit. The CPAT unit would then cancel out the gross motion, allowing the FPAT to have a steady platform to work from. To achieve this performance, a tilt sensor (or gyroscope) paired with a DSP board and the motors' encoders would calculate the required rotation to keep the platform level and then move the motors to do so. In order to cancel out gross motion greater than 2 Hz (based from preliminary experiments), a DSP approach would have to be used. The C6713 Evaluation Board from TI has the capabilities to do this, and is reasonably priced for prototyping. For applications that require the use of only the RF radios, a much smaller and lighter gimbal could be fabricated, which would reduce cost and increase reconnect speeds. The author has actually begun development of these units, but they will not be finished in time to be described in this thesis. They employ standard package DC brushless servo motors from Parker Motion with 20,000ppr encoders, with the entire gimbal only weighing about 10kg. Figure 4.1 below shows the conceptual design of the unit.

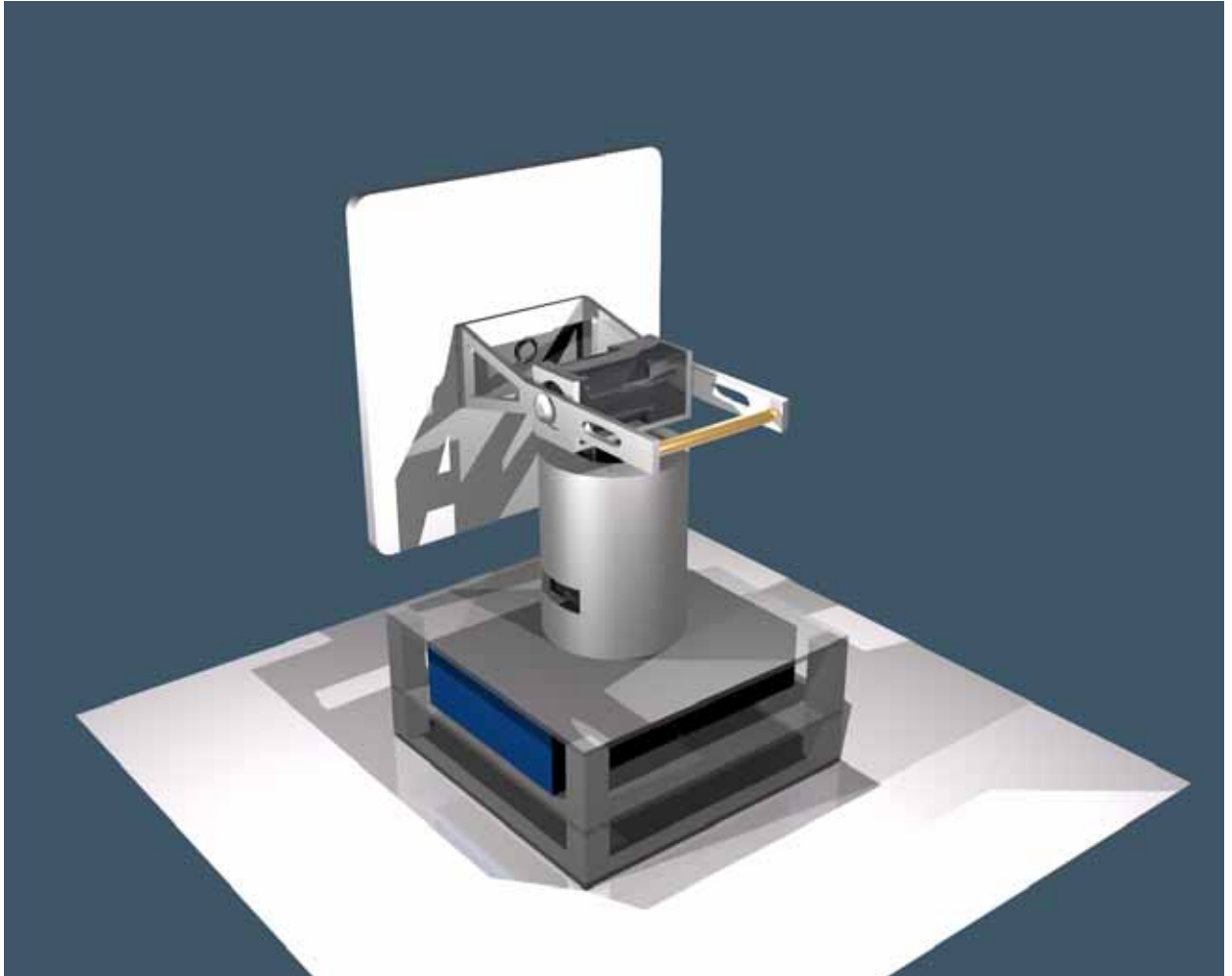


Figure 4.1: Conceptual design of light-weight RF servo gimbal

There are many other modifications that can be made to these units (sliprings, clear domed covers, honeycomb frame, etc), but the ones listed above are those that will cause the most significant increases in performance. While these gimbals are only prototypes, they will be very useful for proving concepts developed by other researchers. Directional communications are fast becoming an important part of the broadband universe, and these agile gimbals will open up new opportunities in reconfigurable networks.

Appendix A: Gimbal CAD Design Drawings

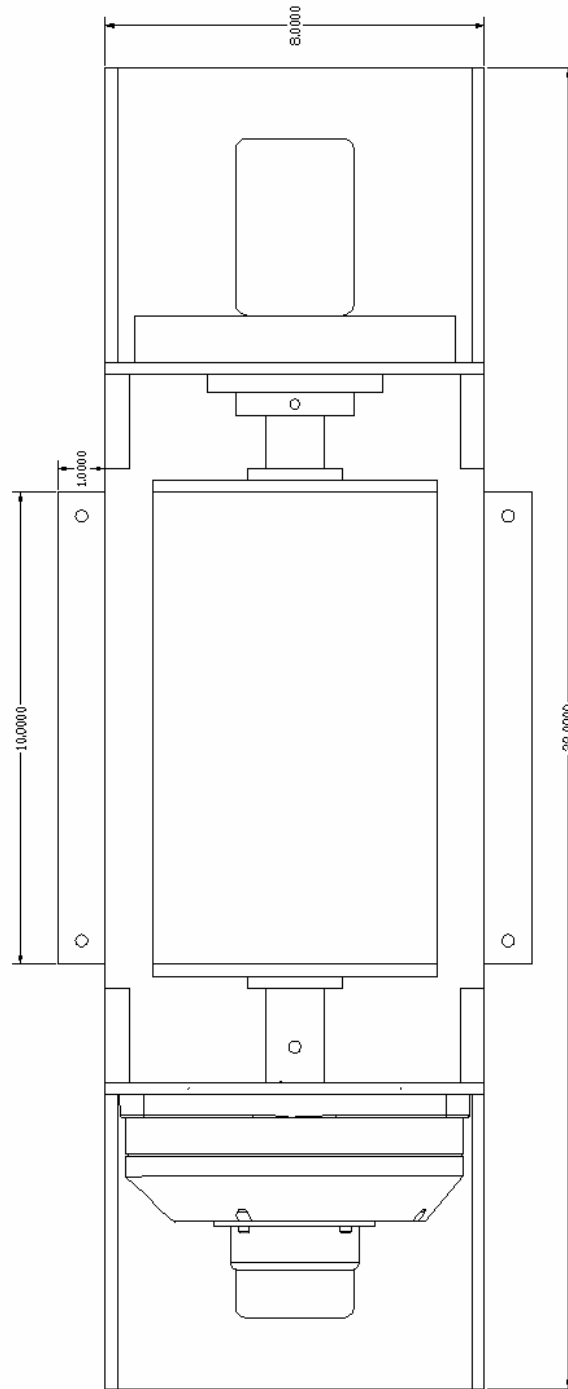


Figure A.1: Unloaded Gimbal Top View

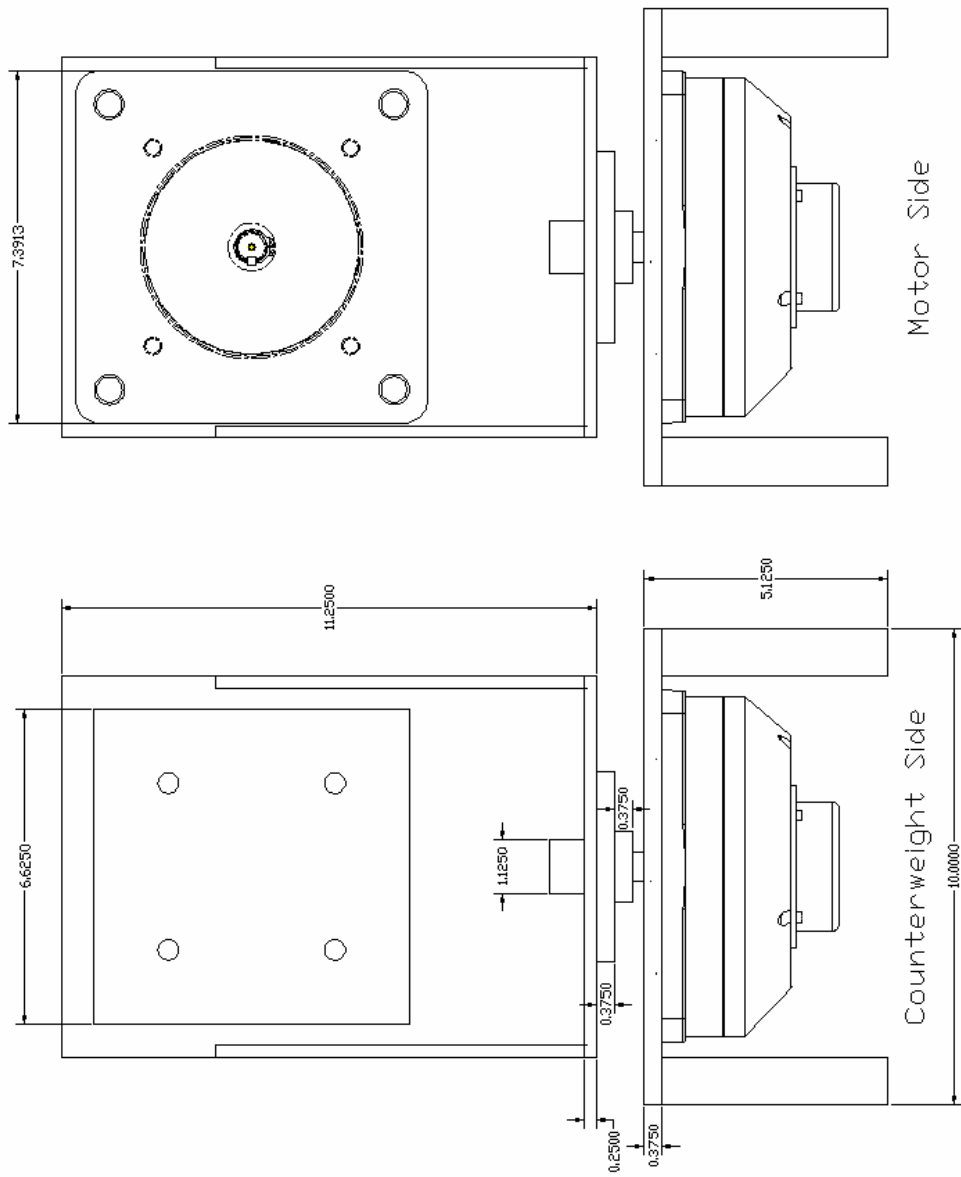


Figure A.2: Gimbal side view

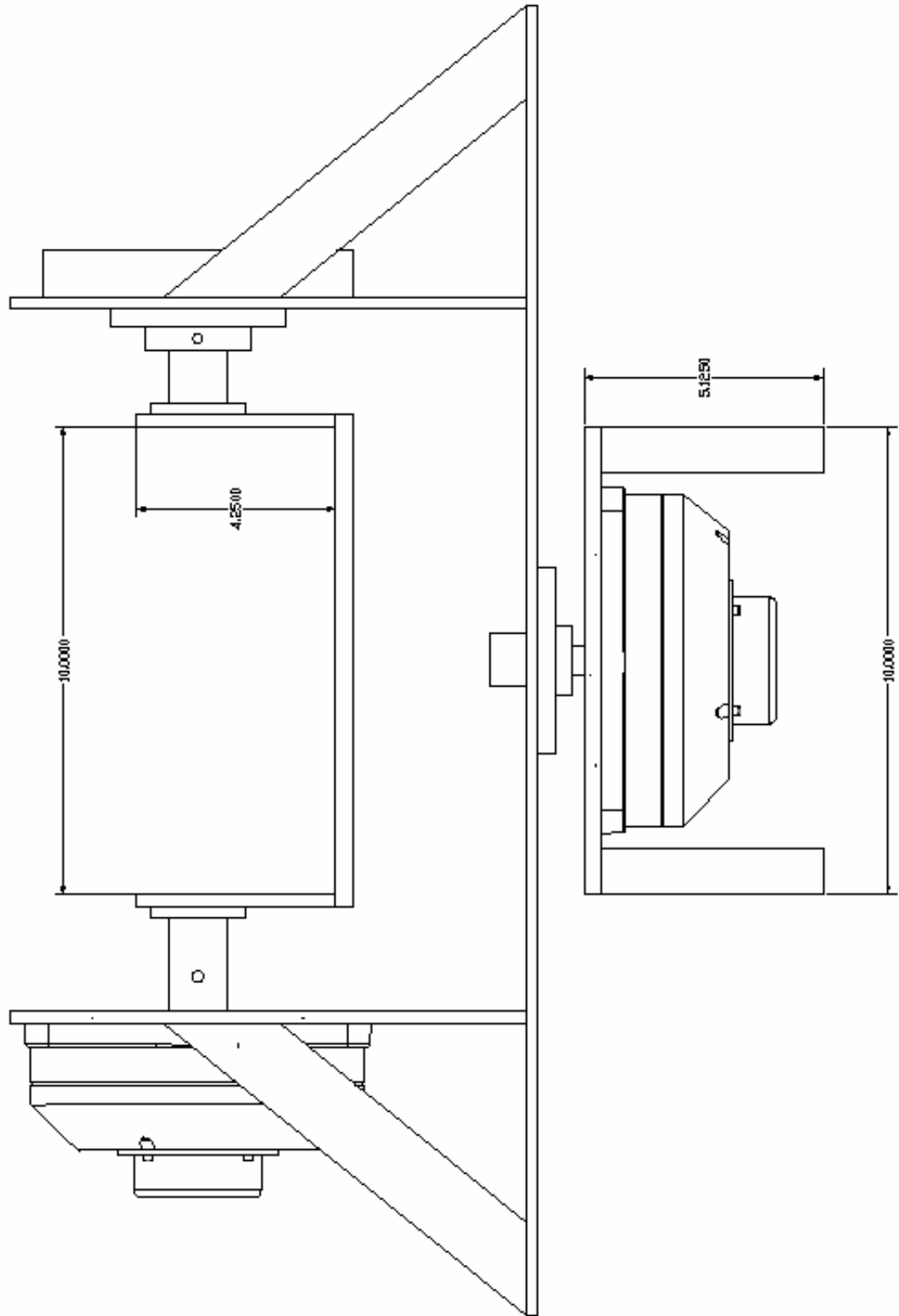


Figure A.3: Gimbal Front View

Bibliography

- [1] <http://www.bodine-electric.com/etorq>
- [2] <http://www.web-books.com/eLibrary/Engineering/Circuits/AC/02459.png>
- [3] www.orientalmotor.com/products/pdfs/C_VEXTA/AS-ASC_Series_Brochure.pdf
- [4] <http://zone.ni.com/devzone/cda/tut/p/id/4623>
- [5] Crowder, R. "Electric Drives and Their Controls," Clarendon Press, 1983
- [6] Kaiser, D. "Fundamentals of Servo Motion Control," Parker Compumotor, 2007
- [7] <http://www.copleycontrols.com/motion/downloads/zip/XenusUserGuide.zip>
- [8] PT Design, "Feed-forward in Position-Velocity Loops," PT Design, 2000
- [9] Ho, T-H. "Pointing, Acquisition, and Tracking Systems for Free-Space Optical Communications Links", University of Maryland, 2007
- [10] http://support.motioneng.com/Downloads-Notes/Tuning/piv_tune_vel_loop.htm
- [11] PT Design, "Cascaded Position-Velocity Loops," PT Design, 2000
- [12] <http://www.rad.com>
- [13] <http://www.canobeam.com>
- [14] <http://www.o-navi.com/falcongx.htm>
- [15] <http://www.ece.umd.edu/mog/tracking.avi>