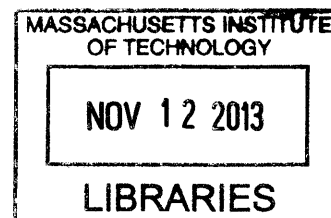


ARCHIVES



**Airborne Protected
Military Satellite Communications:
Analysis of
Open-Loop Pointing and Closed-Loop Tracking
with Noisy Platform Attitude Information**

by

William D. Deike

B.S. Electrical Engineering

United States Air Force Academy, 2008

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2010 *[June 2010]*

© Massachusetts Institute of Technology 2010. All rights reserved.

Author ...

Department of Aeronautics and Astronautics

May 21, 2010

Certified by: ...

[Signature] Timothy Gallagher, Ph.D.

Technical Staff, MIT Lincoln Laboratory

Thesis Supervisor

Certified by: ..

[Signature] Steven R. Hall, Ph.D.

Professor of Aeronautics and Astronautics

[Signature] Thesis Supervisor

Accepted by

[Signature] Eytan H. Modiano, Ph.D.

Associate Professor of Aeronautics and Astronautics

Chair, Committee on Graduate Students

Airborne Protected Military Satellite Communications: Analysis of Open-Loop Pointing and Closed-Loop Tracking with Noisy Platform Attitude Information

by
William D. Deike

Submitted to the Department of Aeronautics and Astronautics
on May 21, 2010, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

U.S. military assets' increasing need for secure global communications has led to the design and fabrication of airborne satellite communication terminals that operate under protected security protocol. Protected transmission limits the closed-loop tracking options to eliminate pointing error in the open-loop pointing solution. In an airborne environment, aircraft disturbances and noisy attitude information affect the open-loop pointing performance. This thesis analyzes the open-loop pointing and closed-loop tracking performance in the presence of open-loop pointing error and uncertainty in the received signal to assess hardware options relative to performance requirements. Results from the open-loop analysis demonstrate unexplained harmonics at integer frequencies while the aircraft is banked, azimuth and elevation errors independent of the inertial pointing vector and aircraft's yaw angle, and uncorrelated azimuth and elevation errors for aircraft pitch and roll angles of $\pm 10^\circ$ and $\pm 30^\circ$, respectively. Several conclusions are drawn from the closed-loop tracking analysis. The distribution of the average noise power has a stronger influence than the distribution of the received isotropic power on the signal-to-noise ratio distribution. The defined step-tracking algorithm reduces pointing error in the open-loop pointing solution for a pedestal experiencing aircraft disturbances and random errors from the GPS/INS. The rate of performance improvement as a function of the number of hops is independent of the antenna aperture size and the GPS/INS unit. Pointing performance relative to the HPBW is independent of the antenna aperture size and GPS/INS unit for on-boresight, but not for off-boresight. With signal-to-noise ratios averaged over 100 hops and pointing biases less than or equal to 0.5 the half-power beamwidth, the step-tracking algorithm reduces the pointing error to within 0.1 the half-power beamwidth of the boresight, for all tested configurations. The overall system performance is bounded by the open-loop pointing solution, which is based on hardware selection. Closed-loop tracking performance is a function of the number of sampled hops and is for the most part independent of the hardware selection.

Thesis Supervisor: Timothy Gallagher, Ph.D.
Title: Technical Staff, MIT Lincoln Laboratory

Thesis Supervisor: Steven R. Hall, Ph.D.
Title: Professor of Aeronautics and Astronautics and MacVicar Faculty Fellow

Acknowledgments

I would like to thank Timothy Gallagher for being my biggest supporter and guide during my past two years here at Lincoln. His knowledge and determination helped me get through the rough patches of my research experience and kept me on track whenever I lost my way. I would also like to thank Professor Steven Hall for his great knowledge, insight, and contributions to this work. A very special acknowledgment goes out to John Kuconis for giving me the opportunity to intern here during the summer of 2007 and to come back for a full Master's Thesis Fellowship. I would like to thank Anthony Hotz and Peter Dolan for their encouragement, advice, and insight as well as patience in answering all my questions. I would also like to thank the following lab employees for making my time here at Lincoln more enjoyable: Kevin Kelly, Neil Mehta, Rajesh Viswanathan, Nagabushan Sivananjaiah, Bruce Hebert, Carmen Petro, and Rosa Figueroa. I would also like to thank the pilots and aircraft maintenance crew at the flight facility for their hard work and dedication. This list is by no means comprehensive, so I would like to express my deepest gratitude to everyone who made my time and experience here in Boston just that much more memorable. Lastly, I would like to thank my family and friends. This would not have been possible without your prayers and support.

Disclaimer

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

Contents

1	Introduction	9
1.1	Motivation for Work	9
1.2	Problem Statement	10
1.3	Contributions	11
1.4	Thesis Overview	11
2	Satellite Communication Terminal Architecture	13
2.1	Antenna Pedestal System	15
2.1.1	Antenna	15
2.1.2	Pedestal	17
2.2	Global Positioning System/Inertial Navigation System	19
2.2.1	Inertial Navigation System	19
2.2.2	Global Positioning System	20
2.2.3	GPS/INS Integration	20
2.3	Satellite Ephemeris/Target Location	22
2.3.1	Satellite Orbital Characteristics	22
2.3.2	Satellite Ephemeris	23
2.3.3	Geostationary Orbits	23
2.4	Pedestal Control Computer	25
2.4.1	Open-Loop Pointing	25
2.4.2	Closed-Loop Tracking	27
3	Open-Loop Pointing	29
3.1	Plant Definition	29
3.1.1	Equations of Motion	29
3.1.2	Elevation Gimbal	34

3.1.3	Azimuth Gimbal	35
3.1.4	Pedestal Dynamics	35
3.1.5	Motor Dynamics	36
3.2	Aircraft Disturbance Spectra Analysis	38
3.2.1	Aircraft Flight Profile Data	38
3.2.2	Aircraft Disturbances	38
3.2.3	Spectral Analysis	40
3.3	Control System Analysis	44
3.3.1	Linearized Plant Model	44
3.3.2	Nonlinear Plant Model	49
3.4	Open-loop Pointing Error Analysis	52
3.4.1	Problem Definition	52
3.4.2	Pointing Error Closed Form Analysis	53
3.4.3	Small Angle Pointing Error Approximation	56
3.4.4	Pointing Error Look Up Tables	60
3.4.5	Open-loop Pointing Error Distribution	64
3.5	Open-loop Antenna Pointing Simulation	68
3.5.1	Racetrack Flight Data	68
3.5.2	Cruise Flight Data	71
4	Closed-Loop Tracking	73
4.1	Effects on Signal-to-Noise Ratio	73
4.1.1	Received Isotropic Power	73
4.1.2	Thermal Noise	78
4.1.3	Receiver Antenna Gain	78
4.1.4	Military Satellite Communications Systems Characteristics . .	78
4.2	Signal-to-Noise Ratio Characterization	79
4.2.1	Thermal Noise Power Characterization	79
4.2.2	Received Isotropic Power Characterization	83
4.2.3	SNR Characterization	87
4.2.4	SNR Analysis	89
4.3	Step-Tracking Algorithm	92
4.3.1	Step-Tracking Theory	92

4.3.2	Test Case Scenarios	96
4.3.3	Step-Tracking SNR Characterization	98
4.4	Closed-Loop Pointing Simulation	99
4.4.1	Ideal Pedestal Simulation	100
4.4.2	Closed-loop Pointing Simulation	103
5	Conclusions and Suggestions for Future Work	109
5.1	Conclusions	109
5.2	Suggestions for Future Work	110
A	Antenna Pedestal Equations of Motion	113
A.1	APS.nb	113
A.2	Final Solution	115
B	Matlab Simulation Code	117
B.1	Spectral Analysis	117
B.1.1	SimSpectra.m	117
B.1.2	FreqAnalysis.m	118
B.1.3	FreqSmooth.m	119
B.2	Linearized Plant Simulation	121
B.2.1	SimLinear.m	121
B.2.2	Linearized Plant Simulink Model	123
B.3	Nonlinear Plant Simulation	127
B.3.1	SimNonlinear.m	127
B.3.2	Nonlinear Plant Simulink Model	128
B.4	INS Error Simulations	131
B.4.1	Pio.m	131
B.4.2	PioAnalysis.m	132
B.4.3	INSErrorSim.m	134
B.4.4	INS Error Simulink Model	136
B.5	Open-Loop Pointing Simulation	137
B.5.1	Simulator.m	137
B.5.2	Open-Loop Pointing Simulink Model	139
B.6	SNR Characterization Simulations	140

B.6.1	SimSNR.m	140
B.6.2	SNR Characterization Simulink Model	142
B.6.3	Dist_Plots.m	142
B.6.4	SNR_Analysis.m	146
B.6.5	SNR_AnalysisII.m	147
B.7	Closed Loop Pointing Simulation	149
B.7.1	AntPatt.m	149
B.7.2	LookUpTable.m	149
B.7.3	FullDitherSquintSim.m	150
B.7.4	Closed-Loop Tracking Simulink Model	154
B.7.5	DitherSquintSim.m	155
C	List of Acronyms and Symbols	157

Chapter 1

Introduction

1.1 Motivation for Work

Satellite communication (SATCOM) systems provide beyond line-of-sight communication with voice, video, and data capabilities. Although SATCOM applications are diverse, the U.S. military has realized the strategic and tactical advantage SATCOM systems can provide to troops in wartime environments, and has utilized this technology in combat zones since the early 1990's [1, 2]. The Military Strategic and Tactical Relay (MILSTAR) program is one constellation of geosynchronous satellites within the Military Satellite Communications (MILSATCOM) system that provides secure beyond line-of-sight communication and enables sensitive information sharing between the President, the Secretary of Defense, and the U.S. Armed Forces around the globe [3].

MILSTAR is a robust “Nuclear Survivable” system with the ability to avoid, repel, and withstand virtually any enemy attack [4]. The MILSTAR satellites operate in the Extremely High Frequency (EHF) band, with center frequencies for downlink and uplink at 20 and 44 GHz, respectively. The system utilizes fast frequency hopping to create low probabilities of interception and detection [5]. The MILSTAR satellites operate in a protected protocol, so there is no tracking beacon for adversaries to locate and jam. At the same time, the lack of a beacon makes it difficult for allies to acquire and track the satellite.

SATCOM terminals point an antenna at an orbiting satellite to secure a communication link. Mobile SATCOM terminals perform the same function within either land-based or airborne vehicles and utilize system feedback to cancel out vehicle motion. An inertially stabilized platform cancels disturbances and keeps a payload within an inertial reference frame. A gimballed pedestal is a type of inertially stabilized platform that stabilizes and points an antenna in a specific direction [6]. The platform uses internal feedback to control the antenna's orientation and point it in the direction commanded by the pedestal control computer commands. The control computer calculates a pointing solution based on the satellite's location and the ter-

minal's location and orientation. This form of control is defined as *open-loop pointing* because the solution incorporates no performance feedback to reduce and eliminate errors within the pointing solution [7]. Errors enter the system and cause inaccuracies in the pointing solution, which reduce the received signal strength and decrease the communication link's performance. More robust systems utilize *closed-loop tracking* to improve the pointing performance by feeding back the received signal strength. Closed-loop tracking reduces bias errors in the pointing solution and improves the communication link.

Commercial terminals rely on feedback using continuous beacons to eliminate pointing errors. The lack of a beacon in protected transmission makes closed-loop tracking more difficult, but still possible. Airborne terminals have difficulty in pointing due to aircraft disturbances and non-ideal system hardware. For protected airborne terminal transmission, closed-loop tracking methods can be modified to reduce uncertainty and eliminate an error from the open-loop pointing solution. If the method is modified improperly, closed-loop tracking may cause more error than reduce the error from open-loop pointing. The modified closed-loop tracking should reduce the uncertainty in feedback and eliminate any open-loop pointing error.

1.2 Problem Statement

MILSATCOM terminals for airborne applications require accurate pointing of the antenna to achieve the best communication link performance. Pointing error biases in the open-loop pointing solution degrade performance and closed-loop tracking attempts to reduce this error. The goal of this thesis is to define relevant parameters that affect the terminal's pointing performance and analyze their impact on a communication link, which is accomplished by the objectives as follows:

1. Defining a nominal, two-axis gimballed antenna pedestal and developing an open-loop pointing controller using state-space control techniques.
2. Obtaining a model for the open-loop pointing error from random errors in the GPS/INS using statistical analysis and simulations.
3. Examining the performance of the open-loop pointing solution through simulation.
4. Characterizing and modeling the received signal-to-noise ratio from the MIL-STAR satellite through simulation.
5. Defining a step-tracking algorithm that accomplishes closed-loop tracking and eliminates pointing error in the presence of uncertainty in the received signal-to-noise ratio.
6. Examining the performance of the closed-loop tracking algorithm through analysis and simulation.

7. Determining the design constraints and system performance for different hardware and their impact on open-loop pointing and closed-loop tracking performance.

1.3 Contributions

This thesis makes the following contributions while accomplishing the objectives outlined in Section 1.2:

1. Analyzes inaccuracies in the open-loop pointing solution caused by random errors in the attitude information from the GPS/INS Euler angle information.
2. Creates an open-loop pointing Simulink model that incorporates the dynamics of the pedestal, the aircraft disturbances, and open-loop pointing solution errors.
3. Characterizes the average signal-to-noise ratio as a function of the number of samples.
4. Creates a closed-loop tracking Simulink model that incorporates the uncertainty of the signal-to-noise ratio and pointing error from the open-loop pointing simulation.

1.4 Thesis Overview

Chapter 2 explains the SATCOM system architecture and the hardware within a MILSATCOM terminal. Chapter 3 focuses on the open-loop pointing portion of the problem by first deriving the pedestal dynamics and an open-loop pointing controller (Objective 1). The chapter then presents analysis on the open-loop pointing error caused by random errors in the GPS/INS Euler angle information (Objective 2). Chapter 3 concludes by presenting simulations of the open-loop pointing performance in the presence of pedestal dynamics, aircraft disturbances, and open-loop pointing errors (Objective 3). Chapter 4 focuses on the closed-loop tracking portion of the problem and begins by defining the signal-to-noise ratio and modeling its uncertainty (Objective 4). The chapter continues by presenting a closed-loop tracking algorithm (Objective 5) and a simulation that tests the closed-loop tracking performance (Objective 6). The chapter concludes with an analysis of the simulation results (Objective 7).

Chapter 2

Satellite Communication Terminal Architecture

The design, fabrication, and implementation of a mobile satellite communication (SATCOM) terminal must balance many different factors including mission constraints and system hardware selection. Mission constraints may limit the terminal to a certain size and weight. System hardware selection determines the level of attainable pointing accuracy, which in turn affects the communication performance. Engineers address all of these issues in mobile SATCOM terminal design. This chapter examines the the issues surrounding the different components of the mobile SATCOM terminal system hardware.

A SATCOM terminal contains all hardware and software required to stabilize and point an antenna at an orbiting satellite and also to transmit and receive data. This thesis focuses on communication performance in the presence of non-ideal stabilization and pointing. Stabilization and pointing in the presence of aircraft disturbances are the functions of the Antenna Positioner System (APS). The APS is comprised of an antenna pedestal system, a Global Positioning System (GPS)/Inertial Navigation System (INS), satellite ephemeris, and a pedestal control computer. The signal processing system is a separate system that performs the terminal's transmit and receive functions. Figure 2-1 is a block diagram of the components of a SATCOM terminal and their connections. The following sections explain each component of the block diagram.

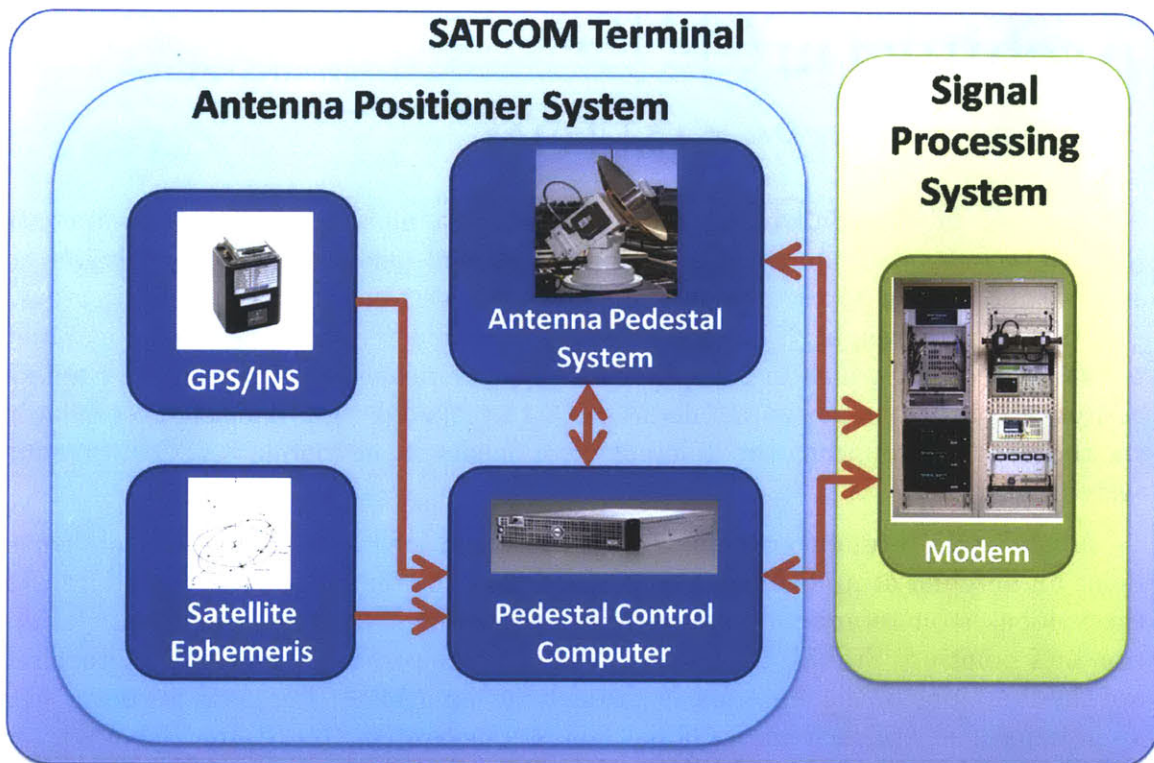


Figure 2-1: SATCOM terminal block diagram defines the Antenna Positioner System, Signal Processing System, their internal subsystems, and the connections between each system.



Figure 2-2: Antenna pedestal system contains the antenna and pedestal. The system passes the received signal to the signal processing system and obeys pointing commands from the pedestal control computer.

2.1 Antenna Pedestal System

The antenna pedestal system consists of the antenna hardware and the pedestal required to point the antenna. Figure 2-2 is a picture of the antenna pedestal system used in this thesis.

2.1.1 Antenna

The antenna collects and directs RF energy between the satellite and the terminal to establish a communication link. In communication theory, an isotropic antenna emits and collects energy uniformly in three-dimensional space. In practice, a directional antenna focuses energy in a specific direction and the direction of maximum gain is defined as the antenna aperture's boresight. Apertures for EHF (30-300 GHz) SATCOM applications are typically highly directional in order to transmit great distances at high frequencies [8].

An antenna beam pattern characterizes the gain with respect to the antenna aperture's boresight and is used to define the pedestal's pointing performance requirement. Figure 2-3 is an example of a antenna beam pattern for a small aperture antenna commonly used in SATCOM systems. The main beam is the region of the beam pattern between the first set of nulls and the side lobes are the successive decreasing lobes on either side of the main beam. The half-power beamwidth (HPBW) is the angle between the two -3dB points and is a function of the aperture size and the transmission frequency. Most SATCOM systems require the pedestal to point at the satellite to within the antenna aperture's HPBW so that the received power does not drop below

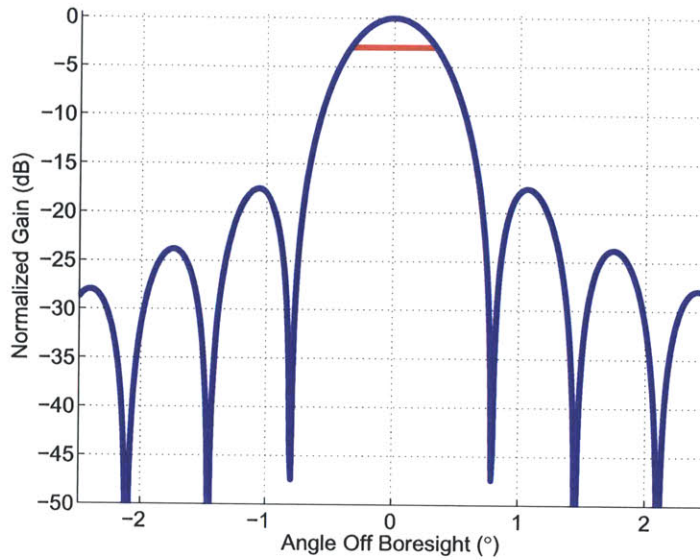


Figure 2-3: A one-dimensional cross-section of a 0.3 m radius aperture operating at 20 GHz. The antenna pattern is normalized to 0 dB and the red line defines the half-power beamwidth region.

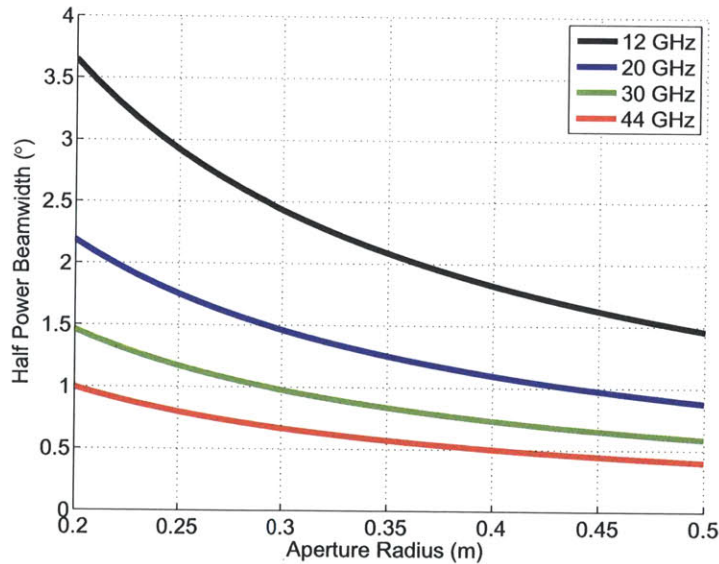


Figure 2-4: Typical half-power beamwidths for various size apertures at different frequencies.

50%. Figure 2-4 demonstrates that as the antenna aperture size and transmission frequency increase, the antenna beam's null-spacing decreases, which translates to tighter pointing requirement [9].

There are two types of antennas that determine the other hardware requirements

in order to steer the aperture's boresight toward the satellite. Fixed boresight antennas, such as a parabolic dish, must be physically moved to steer the boresight. They are typically mounted to gimballed pedestal systems, which enable them to point in any commanded direction. An electrically steerable boresight antenna, such as a phased array, is a much more advanced piece of hardware [9]. The reduction in moving parts makes electrically steerable boresight apertures more attractive to mobile SATCOM applications. However, the design effort is considerably greater, which significantly increases the system cost.

This thesis uses a fixed boresight aperture that transmits and receives at EHF. The transmission frequencies for downlink and uplink are 20 and 44 GHz respectively. An antenna with a fixed aperture is selected over a phased array to reduce the overall system cost. The antenna beam pattern in Figure 2-4 is the normalized one-dimensional beam pattern for a 0.3 m radius antenna aperture at the 20 GHz receive frequency. This thesis focuses on a 0.3 m dish, but also analyzes 0.4 and 0.5 m dishes to understand how performance changes for larger apertures. Each larger aperture has a smaller HPBW, which translates to a tighter pointing performance requirement. Because this thesis uses a fixed boresight aperture, it requires a multi-axis gimballed pedestal to steer the antenna's boresight.

2.1.2 Pedestal

A pedestal stabilizes and points a fixed boresight aperture in a commanded direction, which requires a minimum of two axes of rotation. A gimbal is a collection of motors, bearings, and machined parts that forms a rigid body and allows motion in one axis of rotation [6]. The two-axis gimballed system is the simplest, cheapest, and sturdiest configuration that points an antenna in any direction. The outer gimbal controls the azimuth axis, while the inner gimbal controls the elevation axis. The set of gimbals provide a complete range of motion in a hemispherical field of view from horizon to zenith and steer the antenna's boresight any direction within that range.

The only disadvantage to a two-axis system is the problem of gimbal lock, which occurs at elevation angles approaching zenith. When the elevation gimbal is directly at 90° , the system reaches a singularity, which causes the pedestal to have only one degree of freedom instead of the normal two degrees of freedom. This reduction in axes of rotation imposes no problem in a static tracking problem, but becomes a major concern in a dynamic tracking problem. With dynamic azimuth and elevation commands coming from the pedestal control computer, it becomes increasingly difficult to control the azimuth gimbal at high elevation angles because the required azimuth rate of change approaches infinity.

The problem occurring at zenith is averted by avoiding any elevation angle above 80° , which is commonly referred to as the keyhole region due to the hole in the pedestal's field of view at zenith [9]. Another option, if operation in or near the keyhole region is required, is to have a third gimbal that eliminates the singularity and the gimbal lock constraint. A third gimbal also increases the size, complexity,

and cost of the system. Debruin [9] discusses pedestals with three degrees of freedom and their possible configurations in more detail.

In addition to pointing the antenna in any desired direction, the pedestal also cancels out disturbances and stabilizes the pedestal. Torque disturbances enter the gimballed pedestal and cause unwanted angular accelerations in the axes of rotation resulting in pointing error [6]. These disturbances are caused by coulomb friction, spring torques, imbalance, vehicle motion coupling, intergimbal coupling, internal disturbances, structural flexure, and environmental disturbances [10]. Sensors measure these unwanted rotations and the pedestal then uses the gimbal motors to cancel out the torque disturbances.

The pedestal uses gyroscopic sensors and angular resolvers to sense these unwanted rotations. A gyroscope measures rotational acceleration in inertial space and is typically aligned and mounted with the antenna's reference frame. The pedestal is then able to detect any accelerations the antenna experiences and counteract them with torques in either gimbal. Position resolvers attached to each gimbal feed back orientation data to the pedestal so that any errors between the commanded and actual angles are eliminated. The pedestal system characterized in this thesis utilizes a 2-axis gimballed system to point the antenna because the mission profile does not require keyhole operation based on the aircraft's flight dynamics and area of operation.

Internal to the two-axis pedestal for this thesis, Cleveland Motion Controls (CMC) 2100 series brush servo-motors with F-windings are the steering motors for both gimbals. They have proven to be reliable in other APS projects conducted at Lincoln Laboratory [11]. The EHF SATCOM On-The-Move project mounted an APS on a High Mobility Multipurpose Wheeled Vehicle (HMMWV) and required accurate pointing in the presence of severe vehicle motion in off-road environments [12]. In this thesis, the disturbances from the airborne environment are less, but the operating environment is harsher due to the extreme cold at altitude.

The feedback sensor for the pedestal in this thesis include angular resolvers for each gimbal and a two-axis KVH Industries fiberoptic gyroscope mounted to the inner gimbal to measure rotational accelerations in the pitch and yaw axes [13]. These measurements are fed back internally to the pedestal to ensure that it points where it is commanded. This commanded input comes from the pedestal control computer after it calculates a pointing solution. This calculation requires information on the pedestal's current position, velocity, and orientation. This knowledge comes from the GPS/INS hardware attached to the system.



Figure 2-5: GPS/INS block contains the GPS/INS hardware and transmits the systems location and orientation to the pedestal control computer.

2.2 Global Positioning System/Inertial Navigation System

The Global Positioning System/Inertial Navigation System (GPS/INS) subsystem transmits the system's location and orientation to the pedestal control computer. Figure 2-5 is a picture of this subsystem. This section explains the theory behind both the INS and GPS and why they are combined into one subsystem.

2.2.1 Inertial Navigation System

Many vehicles, including some aircraft and submarines, use an Inertial Navigation System (INS) to provide position and attitude information. To determine positions, an INS measures accelerations using three orthogonal accelerometers. The accelerations are integrated once to obtain inertial velocity, and a second time to obtain position relative to the Earth. Because the vehicle attitude changes over time, the orientation of the accelerometers relative to the navigation frame must be determined. Early versions used an inertially stabilized platform, which uses gyroscopes on the platform and an actuated gimbal system to maintain the accelerometers in a fixed direction in the navigation frame. In more modern strapdown systems, the accelerometers are fixed relative to the vehicle frame, and measured angular rates are integrated to determine the relative orientation of the vehicle and navigation frames [14, 15].

When doing computations in a strapdown INS, several choices of coordinate systems are possible. In this thesis, we will use the North, East, and Down (NED) reference frame. The origin of the NED reference frame is located at a specific point relative to the INS. The Z-axis points down; the X- and Y-axes point North and East, respectively, so that the three axes form an orthogonal basis. Note that the

NED frame is not an inertial frame, because its orientation in inertial space changes as the Earth rotates, and as the vehicle moves over the surface of the Earth. Nevertheless, in some references, the NED frame is treated as an inertial frame, because the angular rate of rotation of the coordinate system with respect to inertial space is significantly smaller than the angular rate experienced by the vehicle [16].

In principle, a strapdown INS with ideal accelerometers and gyroscopes and perfect knowledge of the vehicle initial conditions can determine the vehicle position with perfect accuracy. In practice, the inertial measurements are subject to numerous errors, including scale factor errors, misalignments, and random noise, and the initial conditions are known imperfectly. As a result, the navigation errors of an INS tend to grow over time. The errors can be reduced by incorporating information from other types of navigation systems, such as the Global Positioning System, discussed below.

2.2.2 Global Positioning System

The Global Positioning System (GPS) is an alternative navigation system used in other vehicles such as cars or boats. A constellation of satellites orbiting the Earth make up the GPS network. Each satellite transmits RF energy signals with time-stamped data. Once this data is decoded by a receiver, the data gives the total travel time between satellite and receiver and can then be converted into an estimated range. With a minimum of four satellites in view, the receiver can triangulate its position with a certain degree of precision. Calculating a position requires three range measurements, synchronization errors among the satellites and the receiver cause the measurements to only be approximations of the range (pseudorange). A fourth pseudorange measurement eliminates the synchronization issue [17, 18].

The receiver estimates its position, using a Kalman filter, with higher precision as more satellites are in view. Gaps in the constellation or obstruction in the time-stamped signal can cause the number of measurements to drop below the minimum, resulting in a blackout. The system always stores its last calculated position, but can only extrapolate its next position using calculated velocity vectors prior to the blackout period. The risk of blackout is not an issue for a steady vehicle with little change in orientation. On the other hand, a blackout could be a very serious issue during critical maneuvers that require real-time position and orientation information, such as landing or refueling an aircraft.

2.2.3 GPS/INS Integration

GPS provides an accurate position, but only updates once per second. An INS measures changes in position and orientation at a much higher rate, but accumulates error and drift. Combining the GPS and INS into one system provides accurate position and orientation information. The combination makes up for the shortcomings of each separate system [15]. A Kalman filter optimally blends the two systems in the presence of noise and uncertainty.

A few methods of integration that are commonly used in navigation systems are loosely-coupled, tightly-coupled, and ultra-tightly coupled [19, 20]. Loosely-coupled integration takes the GPS solution and filters it with the INS solution to bound the drift. Loosely-coupled systems utilize two separate Kalman filters in cascade: a Kalman filter in the GPS, and a separate filter that takes the position output and blends it with the INS solution.

Tightly-coupled integration combines the two Kalman filters from the loosely-coupled such that the raw pseudorange measurements from the GPS receiver feed directly into the Kalman filter along with the IMU data. This scheme is more robust in the presence of signal blockage or too few satellites being in view [21, 22].

Ultra-tightly coupled integration is much more advanced than the other two schemes. Strategic navigation systems in wartime environments utilize this integration method in the presence of jamming and interference. The ultra-tightly coupled integration method attempts to mitigate jamming and interference by designing the Kalman filter to utilize the in-phase and quadrature samples from the GPS receiver. Unlike pseudorange measurements, these samples are less susceptible to malicious tampering [23]. Each level of integration increases the precision, but also increases the design complexity and system cost.

The APS within this thesis utilizes a tightly-coupled GPS/INS receiver. The tightly-coupled integration is selected over the other two options because it provides an accurate solution in the presence of blackout and the mission profile does not require the GPS/INS system to operate under jamming and interference. The GPS/INS supplies the APS with accurate knowledge of the system's current location and orientation, so the control computer can calculate a pointing solution.

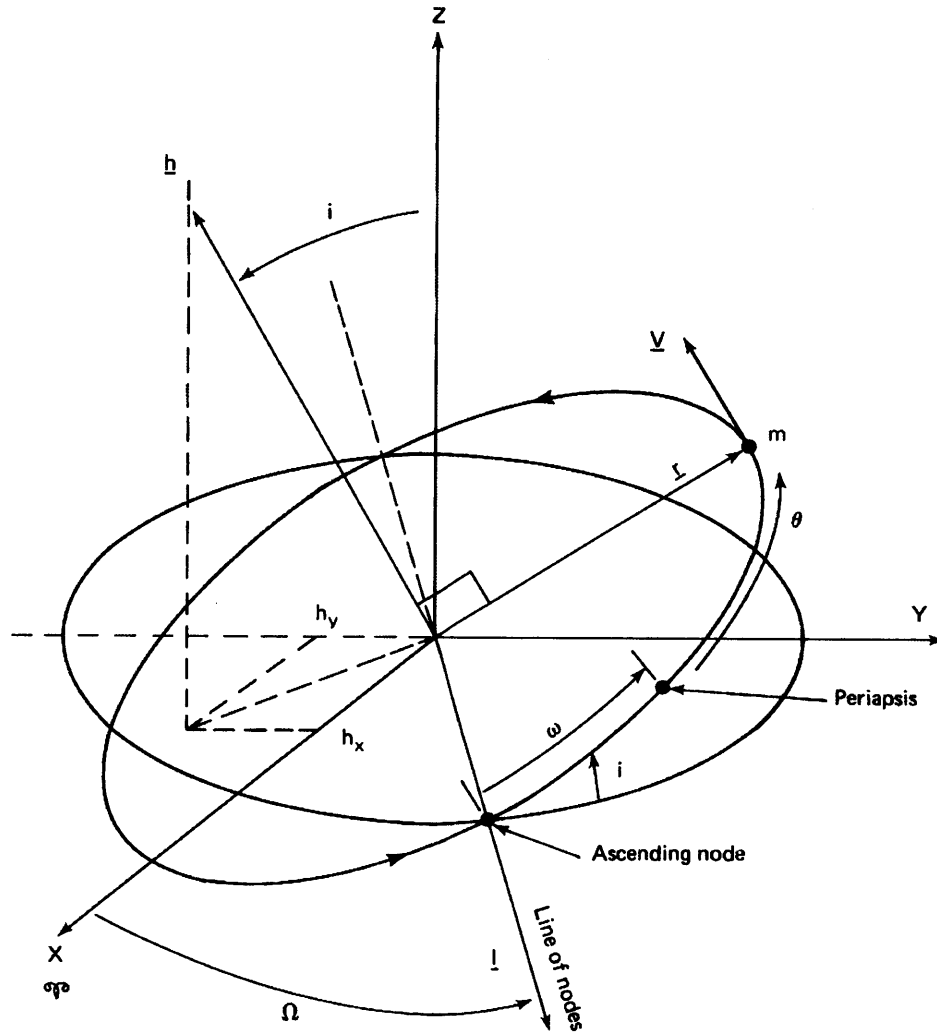


Figure 2-6: Satellite ephemeris contains orbital parameters that define the satellite's location at specific instance in time, which the computer can then use to estimate future target coordinates. Reproduced from Reference 24.

2.3 Satellite Ephemeris/Target Location

The pedestal control computer requires an accurate knowledge of the satellite's location in order to calculate a pointing solution between the terminal and the satellite. Satellite ephemeris pinpoints the satellite's exact location at a given instance in time and can then be used to estimate future positions in time. Figure 2-6 is an illustration of the satellite's location information contained in ephemeris.

2.3.1 Satellite Orbital Characteristics

In order to know the satellite's location, it is useful to understand the physics behind satellite orbits. Equations of motion describe a satellite's trajectory around the earth.

These equations require six orbital elements, which define the satellite's orbit and its location in the orbit. Figure 2-6 presents the six orbital elements. Once these elements are known for a given instance in time, computers use orbital propagation algorithms to estimate the satellite's trajectory and future position.

Over time, orbital perturbations cause the orbital paths of each satellite to change, creating an error between the true and estimated positions. Atmospheric drag, the Earth's oblateness, solar radiation pressure, and third-body gravitational effects are common perturbations experienced by satellites [25]. Physically changing the orbit or updating the orbital parameters to reflect the new, perturbed orbit are two methods to reduce the error between the estimated and true position.

Engineers take orbital perturbations into consideration when designing a satellite for a specific mission and install hardware on the satellite to reduce or eliminate perturbations the satellite may encounter during its life span. Engineers track the satellite's orbit, and can command the satellite to use onboard hardware, such as a propellant tank or stabilization gyroscope, to get it back in its desired position and orientation [25]. If onboard hardware cannot correct for the perturbation, then the satellite's ephemeris is updated to account for this change.

2.3.2 Satellite Ephemeris

The calculation of a satellite's ephemeris uses high fidelity orbital propagation algorithms to account for predictable perturbations, but cannot account for random perturbations. To account for random perturbations, tracking stations observe and record all perturbations in a satellite's orbit and routinely update the ephemeris by transmitting it to the satellite. Updating the ephemeris helps ease acquisition, because the satellite can either broadcast it continuously or transmits it to a user after receiving a request [17]. Uncorrected ephemeris only poses a serious threat if the system is unable to calculate the satellite position well enough to establish a communication link. Once the link is established, the ephemeris can be updated and the terminal will have the best estimate of the satellite's location.

2.3.3 Geostationary Orbits

Satellite orbits are classified based on their shape, direction, or altitude. Satellites fall into one of three altitude ranges: low earth orbit (LEO), medium earth orbit (MEO), and geosynchronous earth orbit (GEO). Satellites are considered LEO if their altitudes are less than 1000 km and satellites with an altitude of exactly 35,786 km are GEO. Satellites with an altitude between 1000 and 35,786 km are categorized as MEO [26].

GEO satellites have the unique trait of a 24 hour orbital period. To an observer on the Earth's surface, a GEO satellite appears to make a figure-eight pattern in the sky. If a GEO satellite has an inclination of 0° , so that it orbits the Earth along

its equatorial plane, then the satellite is in a geostationary orbit, which means the satellite would appear as a fixed point in the sky. Geostationary orbits are commonly used by communication satellites to eliminate the added complexity of pointing at a moving target. MILSTAR satellites operate in geostationary orbits.

Besides the reduction in acquisition and tracking, GEO satellites at such a high altitude have a very wide area of coverage. A GEO satellite can see nearly a fourth of the Earth's surface, and can transmit very wide or narrow beams depending on the mission parameters or even multiple beams to allow for different coverage areas at different transmission rates. Additionally, the Doppler effect is not a significant issue because the satellite is stationary relative to the Earth's surface and there is little to no change in relative motion between the satellite and terminal, even in an airborne application [8].

GEO satellites operate at such a high altitude that signal attenuation and transmission delays can be a problem. Attenuation is a function of the transmission frequency and the distance the signal travels. Section 4.1 discusses this in more detail as it is related to the Received Isotropic Power (RIP). RF signals propagate at the speed of light, which means a signal takes approximately 120 ms to travel between antennas. This propagation delay may present a serious issue if the mission requires real-time communications.

Orbital perturbations cause GEO satellites to move from their fixed position above the Earth. Atmospheric drag does not affect GEO satellites because they operate 25,000 km above the atmosphere. At this extreme distance away from the Earth, the orbit's eccentricity and inclination change over time. Solar-radiation pressure causes long term variations in eccentricity, while third-body gravitational effects from the Sun and Moon cause long-term variations in inclination. In addition, tesseral harmonics induced by the Earth's gravitational field cause longitudinal shifts in the satellite's position over the Earth [27].

Scientists have documented each of these perturbations, and engineers take them into consideration when designing the satellite. Most GEO satellites mitigate these errors and maintain their intended orbit and fixed location in the sky. In addition, satellite operators track the satellite's performance and take the necessary actions to ensure the satellite follows its intended orbit. As stated previously, the ephemeris reflects all perturbations and changes in the satellite's orbit and should give the best possible estimate of the satellite's location.

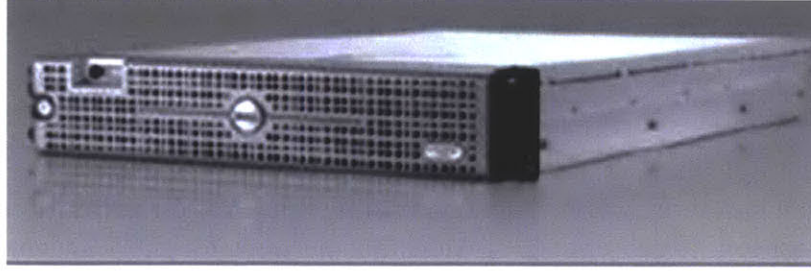


Figure 2-7: Pedestal control computer interacts with all of the other subsystems within the APS as well as the Signal Processing System. The computer is responsible for calculating the pointing solution and performing closed-loop tracking.

2.4 Pedestal Control Computer

The pedestal control computer, presented in Figure 2-7, takes the terminal position and orientation data from the GPS/INS and the satellite ephemeris and calculates a pointing solution. The computer then commands the pedestal to steer the antenna's boresight in the direction of the pointing solution. Feedback internal to the pedestal controls the gimbal orientations. This feedback is defined as open-loop pointing, because the system does not utilize feedback from the terminal to improve the pointing solution. The pedestal control computer performs closed-loop tracking by manipulating the received signal level to eliminate pointing error biases from the open-loop pointing solution.

Open-loop pointing is the simple solution to the pointing problem, but cannot determine the amount of pointing error between the terminal and the satellite. Closed-loop tracking provides a more robust solution that eliminates open-loop pointing errors, but decreases performance temporarily and could potentially decrease overall system performance. The concerns of each are discussed in this section and Chapters 3 and 4 study the trade-offs in more detail.

2.4.1 Open-Loop Pointing

Open-loop pointing receives information on the target's location and the terminal's position and orientation, calculates a pointing solution, and commands the pedestal to point in the calculated direction. Several early SATCOM systems performed open-loop pointing with great success [28, 29]. The fundamental problem with open-loop pointing is that there is no way to eliminate an error in the open-loop pointing solution. This open-loop pointing error is a summation of several smaller errors within the pedestal that impact the final pointing solution. The errors include

1. Aged satellite ephemeris at the terminal.
2. GPS/INS position, orientation, and non-orthogonality error.

3. Misalignment errors between components.
4. Steady-state biasing in pedestal resolvers.

Aged satellite ephemeris causes error in the satellite's estimated position. Perturbations in the satellite's orbit cause the satellite's position to be different from the estimated position. Aged ephemeris does not account for more recent perturbations and could potentially estimate an incorrect satellite location. Aged ephemeris is only an issue in the acquisition phase of the communication link. Once the terminal has acquired the satellite, it can request updated ephemeris, which is then stored in the pedestal computer's memory. The impact of aged ephemeris is reduced for geostationary orbits because the satellite appears as a fixed point in the sky. Additionally, the distance between the terminal and satellite is great, which means errors in the satellite's position translate to infinitesimal error angles.

A much more serious error stems from inaccuracies in the GPS/INS solution. Noisy sensors within the INS cause errors to accumulate over time. Integrating the output with the GPS bounds the drift and reduces the error. GPS/INS hardware specifications define the position and orientation errors as Gaussian random variables with defined variance. Similar to the satellite position error, errors in the terminal's position do not cause serious errors in the pointing solution. On the other hand, orientation errors factor directly into the pointing solution [30]. The amount of pointing error in the open-loop pointing solution is tied directly to orientation error and are analyzed further in Section 3.4. Misalignment between the sensors causes non-orthogonality in the system. If one sensor is misaligned, then it senses acceleration in an axis other than the intended axis, which causes an error in the system's reported orientation. Navigation grade GPS/INS are put through rigorous tests to ensure the sensors are indeed orthogonal.

Unmeasured misalignment errors in any of the components cause an error in the pointing solution. Misalignment errors are either static or dynamic. Mounting misalignment errors cause static biasing errors between components, while bending and flexing of the aircraft frame cause dynamic misalignment errors. Static errors are mitigated by paying extra attention to accurate mounting and alignment. If the GPS/INS is far enough from the pedestal, flexing of the aircraft could cause dynamic errors. Mounting the GPS/INS to the base of the pedestal mitigates this error. Misalignment errors can also exist internal to the pedestal or between the antenna and the pedestal. All of these misalignments are factors directly influencing the pedestal's pointing accuracy and the terminal's communication performance. It is assumed that the careful design of the antenna pedestal minimizes these errors.

The final form of error is steady-state biasing in the pedestal resolvers. This bias results in an error between where the pedestal is commanded to point and where it actually points. This bias will cause a significant error not in the pointing solution, but in the pedestal pointing accuracy. The most effective method to eliminate this error is through careful design and calibration.

2.4.2 Closed-Loop Tracking

The pedestal control computer performs closed-loop tracking by calculating the same pointing solution as before and then using the received signal-to-noise ratio to detect any error in the current pointing solution. Three closed-loop tracking strategies are commonly used in radar and communication systems.

1. Monopulse
2. Conical Scanning
3. Step tracking

Monopulse tracking uses multiple antennas to locate and track a target. The signal levels from the individual antenna feeds are manipulated to determine a pointing offset between the antenna and the target. The pointing solution is updated with the calculated pointing error [31, 32]. When the signal levels from all of the feeds are equal, the main beam is accurately pointed at the target. Monopulse tracking is more commonly found in radar tracking systems than communication systems because they require advanced hardware, with multiple antenna feeds. The antenna within the MILSATCOM terminal has only one feed. For this reason, monopulse tracking is not a viable solution.

Conical scanning, conscan for short, is a technique common to both radar and communication systems. Instead of using multiple feeds as monopulse tracking does, conscan requires only one antenna feed. The antenna aperture's beam is mechanically steered in a circular motion around the estimated pointing angle. The circular motion causes sinusoidal variations in the received signal power, which are then used to estimate the pointing error. This estimated error is fed back into the pedestal as gimballed orientation corrections [33]. The radius of the conscan movement is selected based on the antenna pattern so no significant loss in signal power occurs. The sinusoidal frequency is chosen based on the system's sampling rate. Other more elaborate scanning patterns, such as the Lissajous and rosetta pattern, have been designed and tested and present similar results [33]. Conscan tracking can be very useful, but it may require extra equipment. The pedestal control computer can command the pedestal to manually steer the antenna or the feed within the antenna can be off-centered and rotated. The latter requires more equipment and increases the system's complexity and cost. Furthermore, conscan tracking works best in systems with continuous beacon. For protected systems, uncertainty in the signal-to-noise ratio degrades the performance of the conscan tracking algorithm and the continuous motion of conscan tracking degrades communication performance.

The simplest and least expensive method for closed-loop pointing is step tracking, which has some of the advantages of both monopulse and conscan techniques. Step tracking requires only one feed, so no modification to the antenna is required. In addition, step tracking does not require any augmentation to the pedestal to scan the antenna beam. Step tracking takes SNR readings at specific points in a desired

pattern and then adds and subtracts the samples to estimate the pointing error, which the computer then uses to recalculate the pointing solution. The difference between step tracking and conscan is that step tracking points at a fixed location in the sky and takes enough samples to estimate the SNR rather than continuously scanning the antenna. This thesis focuses on step tracking because it is the most practical form of closed-loop tracking for protected MILSATCOM transmission.

Chapter 3

Open-Loop Pointing

3.1 Plant Definition

The purpose of this section is to define the plant model of the pedestal by characterizing the dynamics of a two-axis gimballed system and the dynamics of the attached motors.

3.1.1 Equations of Motion

The equations of motion describe the system's response to internal and external forces. The response side of the equation, typically the left hand side of the equation, defines the internal interactions within the system, while the moment side, the right hand side, defines the external torques acting on the system. The derivation of the equations of motion is commonly available for a standard rotating rigid body and explicit definitions of both sides of the equation are commonly available [16]. The two-axis gimballed pedestal is not a rigid body because of its two axes of rotation. Therefore a more rigorous derivation is required to solve the equations of motion that govern a two-axis gimballed pedestal.

The antenna pedestal houses a two-axis gimballed system as depicted in Figure 3-1. The two degrees of freedom allow the pedestal to point the antenna in any direction within the pedestal's hemispherical field of view. The azimuth and elevation gimbal angles, represented by ψ and θ respectively, define the orientation of the APS.

Three reference frames describe the orientation of the three pedestal components. These define the rotation transformation from the aircraft frame to the antenna body frame. In addition to the aircraft reference frame, denoted by $[x_k, y_k, z_k]$, the other two frames are fixed to the azimuth and elevation gimbals within the system, and are denoted by $[x_b, y_b, z_b]$ and $[x_a, y_a, z_a]$, respectively. Figure 3-1 graphically defines each reference frame.

A rotational transformation matrix defines the transition from one frame to an-

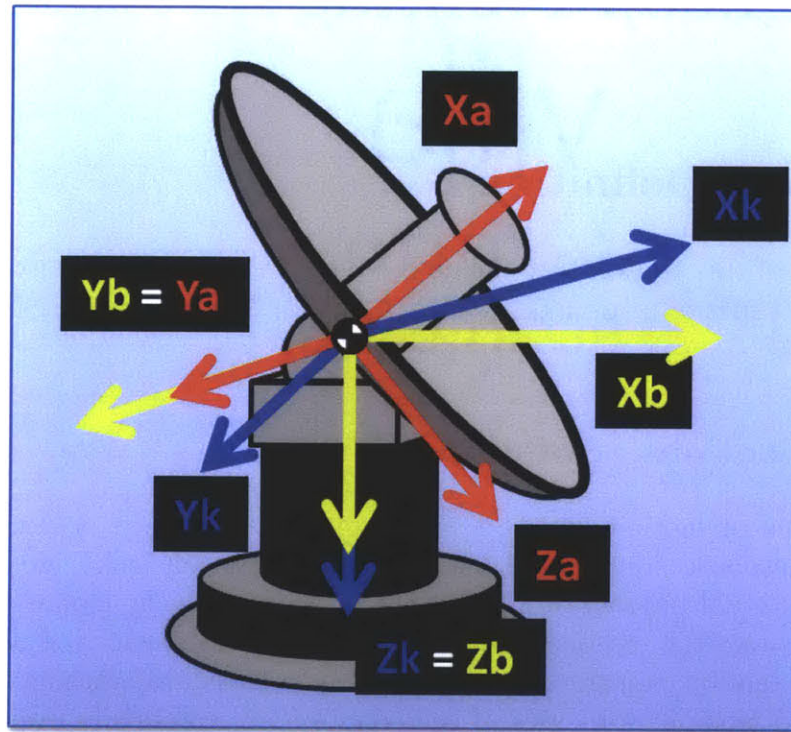


Figure 3-1: The antenna pedestal two-axis gimballed system contains three reference frames. $[x_k, y_k, z_k]$ is fixed to the base of the pedestal and is aligned with the aircraft reference frame. $[x_b, y_b, z_b]$ is fixed to the azimuth gimbal and serves as the transition frame. $[x_a, y_a, z_a]$ is fixed to the elevation gimbal and is aligned with the antenna's reference frame.

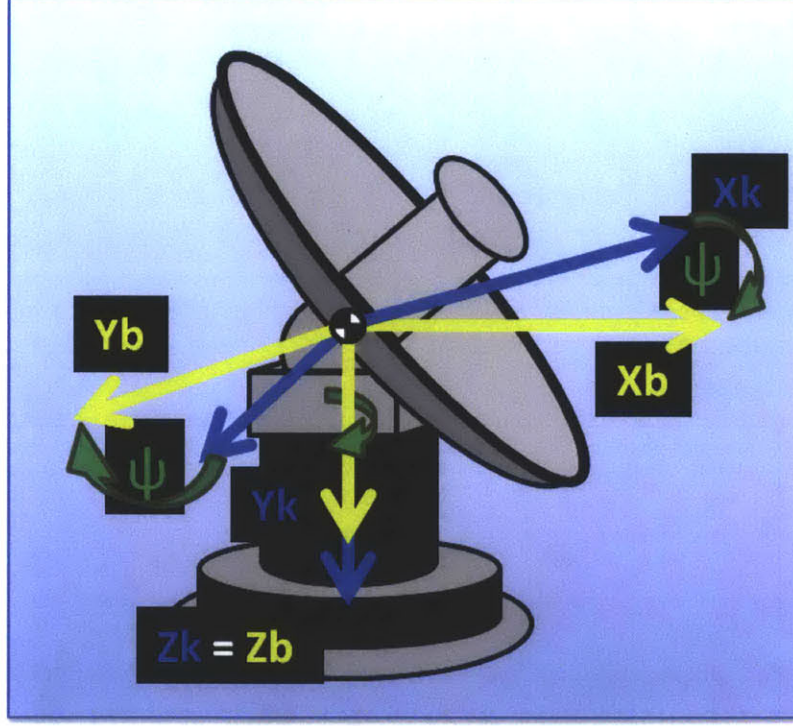


Figure 3-2: The azimuth gimbal rotation about the Z-axis by azimuth angle ψ . The rotation transformation from $[x_k, y_k, z_k]$ to $[x_b, y_b, z_b]$ is the relation between the aircraft frame and the transitional frame.

other. The transformation matrices from the aircraft frame to the azimuth frame by the angle ψ and the azimuth frame to the elevation frame by the angle θ are defined as

$$R_{bk} = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

and

$$R_{ab} = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (3.2)$$

respectively. Figures 3-2 and 3-3 depict the two rotations about angles ψ and θ respectively.

An angular rotation in one frame is related to an angular rotation in another frame by one of the transformation matrices. This angular rotation depends on the azimuth and elevation gimbal orientations as well as the gimbal angular velocities [34]. The angular velocities for the three reference frames are defined as

$$\omega_k = \begin{bmatrix} p_k \\ q_k \\ r_k \end{bmatrix}, \quad (3.3)$$

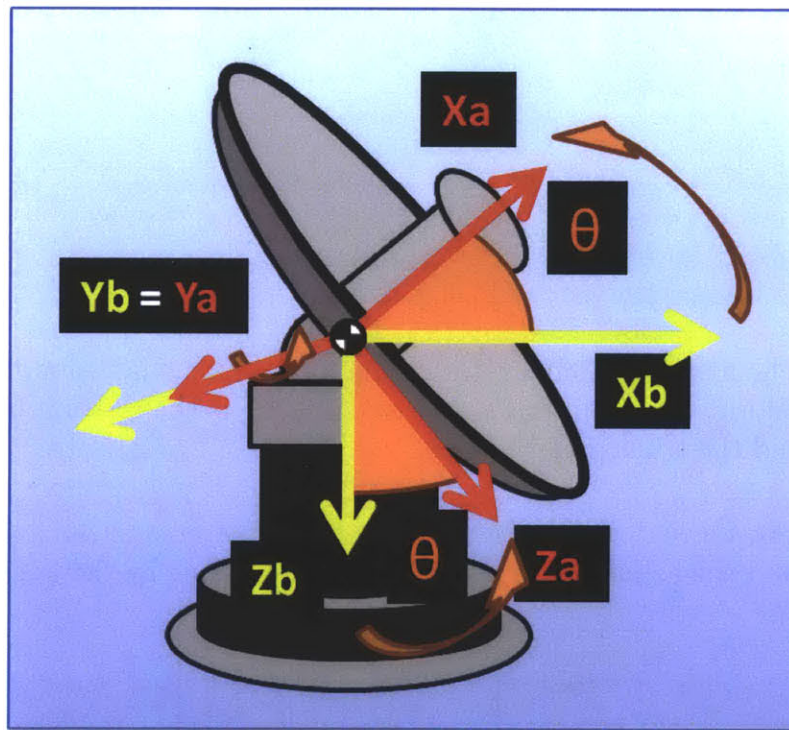


Figure 3-3: The elevation gimbal rotation about the Y-axis by elevation angle θ . The rotation transformation from $[x_b, y_b, z_b]$ to $[x_a, y_a, z_a]$ is the relation between the transitional frame and the antenna frame.

$$\omega_b = \begin{bmatrix} p_b \\ q_b \\ r_b \end{bmatrix}, \quad (3.4)$$

and

$$\omega_a = \begin{bmatrix} p_a \\ q_a \\ r_a \end{bmatrix}, \quad (3.5)$$

where p, q, r represent the roll, pitch and yaw components in each frame, respectively.

The relationships between the angular velocities and their respective reference frames are given by

$$\begin{bmatrix} p_b \\ q_b \\ r_b \end{bmatrix} = \begin{bmatrix} p_k \cos(\psi) + q_k \sin(\psi) \\ -p_k \sin(\psi) + q_k \cos(\psi) \\ r_k + \dot{\psi} \end{bmatrix} \quad (3.6)$$

and

$$\begin{bmatrix} p_a \\ q_a \\ r_a \end{bmatrix} = \begin{bmatrix} p_b \cos(\theta) - r_b \sin(\theta) \\ q_b + \dot{\theta} \\ p_b \sin(\theta) + r_b \cos(\theta) \end{bmatrix}, \quad (3.7)$$

where ψ and $\dot{\psi}$ are the angle and angular rate of the azimuth gimbal and θ and $\dot{\theta}$ are the angle and angular rate of the elevation gimbal [34].

The two axes of concern in this application are the pitch and yaw velocities (q_a and r_a) of the elevation gimbal because unwanted rotations in these axes correspond to pointing error between the antenna and the satellite. Because the antenna aperture is circularly symmetric, no orientation requirement exists between the terminal and satellite, so rotation in the roll axis does not impact performance. Any deviation in either the pitch or yaw axis will result in a pointing error and a loss in signal strength, and so the pedestal is designed to eliminate these rotations.

The standard equations of motion for a rigid body cannot describe the antenna pedestal because it can rotate in two axes. The gimbals within the pedestal are rigid bodies, so the standard equations of motion describe the dynamics of each gimbal. Each gimbal has an inertia matrix associated with it that is defined by

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{xz} & I_{yz} & I_{zz} \end{bmatrix}, \quad (3.8)$$

where (I_{xx}, I_{yy}, I_{zz}) are the moments of inertia and (I_{xy}, I_{xz}, I_{yz}) are the products of inertia. The moments indicate the rigid body's resistance to rotation about each axis and the products indicate the cross-coupling and symmetry of the body [16].

The equations of motion of a rigid body define the angular velocities and accelerations caused by torques entering the system and interacting with the inertia matrix

as indicated by

$$I\dot{\omega} + \omega \times I\omega = T, \quad (3.9)$$

where $\dot{\omega}$, ω , and T are the angular acceleration, angular velocity, and external torque. As defined earlier, the left hand side contains the response and the right hand side contains the external torques. The fully derived equations of motion with all angular velocities and accelerations are

$$\begin{bmatrix} \dot{p}I_x + qr(I_z - I_y) - (q^2 - r^2)I_{yz} - (\dot{r} + pq)I_{xz} + (pr - \dot{q})I_{xy} \\ \dot{q}I_y - pr(I_z - I_x) + (p^2 - r^2)I_{xz} - (rq + \dot{p})I_{xy} + (pq - \dot{r})I_{yz} \\ \dot{r}I_z + pq(I_y - I_x) - (p^2 - q^2)I_{xy} - (pr + \dot{q})I_{yz} + (qr - \dot{p})I_{xz} \end{bmatrix} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}. \quad (3.10)$$

Equation 3.10 is the standard equations of motion for a rotating rigid body [16].

3.1.2 Elevation Gimbal

The elevation gimbal is isolated from the rest of the system and treated as a rigid body that contains both the gimbal and the mounted antenna. The standard equations of motion from Equation 3.10 characterize the dynamics of this subsystem. The derivation makes a simplification that reduces the complex cross-coupling among the three axes. Setting the products of inertia equal to zero cancels out the last three terms on the left hand side of Equation 3.10. This substitution is a reasonable simplification often done in practice because the source of the coupling is understood and engineers design systems with the inertia matrix in mind. This practice is done so that the final system is balanced with minimal cross-coupling between the axes. Pedestal designs follow this practice, so the resulting equations of motion for the elevation gimbal are defined by

$$I_a\dot{\omega}_a + \omega_a \times I_a\omega_a = T_a, \quad (3.11)$$

which simplifies to

$$\begin{bmatrix} \dot{p}_a I_{xa} + q_a r_a (I_{za} - I_{ya}) \\ \dot{q}_a I_{ya} - p_a r_a (I_{za} - I_{xa}) \\ \dot{r}_a I_{za} + p_a q_a (I_{ya} - I_{xa}) \end{bmatrix} = \begin{bmatrix} T_{xa} \\ T_{ya} \\ T_{za} \end{bmatrix}. \quad (3.12)$$

Coupling still exists among the three axes so most texts go a step further and eliminate the cross-coupling by linearizing the equations of motion around an operating point, such as straight and level flight [16]. This derivation incorporates the cross-coupling to more accurately characterize the dynamics and interactions within the pedestal.

The elevation gimbal controls rotation in the Y-axis by angle θ as indicated by Figure 3-3. T_{ya} in Equation 3.12 represents the external torque the gimbal exerts on the system to cause rotation in the pedestal's elevation axis. T_{xa} and T_{za} are external torques the elevation gimbal cannot control, but still affect the gimbal and cause rotations in the yaw and roll axes of the elevation gimbal. All three torques define the interaction between the azimuth and elevation gimbal.

3.1.3 Azimuth Gimbal

The azimuth gimbal is another rigid body, so Equation 3.10 also characterizes the dynamics of the azimuth gimbal, but the right hand side of the equation is a little more complex because of the interaction between the two gimbals. The new right hand side includes the torques imposed directly on the azimuth gimbal as well as the torques from the elevation gimbal after a proper rotation transformation. The equations of motion of the azimuth gimbal are defined by

$$I_b \dot{\omega}_b + \omega_b \times I_b \omega_b = T_b - R_{ab}^{-1} T_a, \quad (3.13)$$

which fully expands to

$$\begin{bmatrix} \dot{p}_b I_{xb} + q_b r_b (I_{zb} - I_{yb}) \\ \dot{q}_b I_{yb} - p_b r_b (I_{zb} - I_{xb}) \\ \dot{r}_b I_{zb} + p_b q_b (I_{yb} - I_{xb}) \end{bmatrix} = \begin{bmatrix} T_{xb} \\ T_{yb} \\ T_{zb} \end{bmatrix} - R_{ab}^{-1} \begin{bmatrix} T_{xa} \\ T_{ya} \\ T_{za} \end{bmatrix}. \quad (3.14)$$

The azimuth gimbal controls rotation in the Z-axis by angle ψ as indicated by Figure 3-2. T_{zb} in Equation 3.14 represents the external torque the gimbal exerts on the system to cause rotation in the pedestal's azimuth axis. T_{xb} and T_{yb} are external torques the azimuth gimbal cannot control. These torques along with the torques from the elevation gimbal cause rotations in the roll and pitch axes of the azimuth gimbal. All of the external torques imposed on the pedestal come from the aircraft's angular velocities and accelerations, p_k, q_k, r_k and $\dot{p}_k, \dot{q}_k, \dot{r}_k$ respectively.

3.1.4 Pedestal Dynamics

Equations 3.11 and 3.13 characterize the dynamics of the pedestal, but do not convey the relationship between the inputs and outputs very well. Equations 3.6 and 3.7 are substituted into Equations 3.11 and 3.13 to solve for the angular accelerations in both gimbals as a function of the other parameters. After some simplification, the equations for the elevation and azimuth gimbals are

$$\ddot{\theta} = \frac{1}{I_{ya}} (T_\theta + (I_{za} - I_{xa}) p_a r_a) - \dot{q}_b \quad (3.15)$$

and

$$\ddot{\psi} = \frac{1}{I_Z} [T_\psi + I_{d1} + I_{d2} + I_{d3}] - \dot{r}_k \quad (3.16)$$

respectively, where

$$\begin{aligned} I_Z &= I_{zb} + I_{xa} \sin^2(\theta) + I_{za} \cos^2(\theta) \\ I_{d1} &= [I_{xb} + I_{xa} \cos^2(\theta) + I_{za} \sin^2(\theta)] p_b q_b \\ I_{d2} &= \frac{1}{2} (I_{xa} - I_{za}) \sin(2\theta) (\dot{p}_b - q_b r_b) \\ I_{d3} &= \theta [(I_{xa} - I_{za}) (p_b \cos(2\theta) - r_b \sin(2\theta)) - I_{ya} p_b] \end{aligned} \quad (3.17)$$

It is important to note that Equations 3.15 and 3.16 are in terms of angular velocities and accelerations in each of the three reference frames defined by Equations 3.6 and 3.7. The true solution is only in terms of the aircraft angular velocities and accelerations, the torques, and the gimbals' states. Appendix A presents the Wolfram Mathematica code to derive the equations of motion. Ekstrand [34] presents an alternative derivation for the two-axis gimballed system and confirms the final equations of motion for the two gimbals.

3.1.5 Motor Dynamics

With the equations of motion defined, the next step is to replace the applied torques in the equations with motor dynamics.

Two DC servomotors mechanically steer the pedestal and apply torques to the azimuth and elevation gimbals. An armature current is applied to govern the rotation of the servomotor's shaft. The torque produced by the servomotor is directly proportional to the armature current as defined by

$$T = Ki_a, \quad (3.18)$$

where T , i_a , and K are the torque, armature current and motor-torque constant respectively. If the armature is rotating, the armature induces a voltage proportional to the product of the flux and angular velocity. If the flux is constant, then the induced voltage is proportional to the angular velocity as defined by

$$e_b = K_b \dot{\theta}, \quad (3.19)$$

where e_b , $\dot{\theta}$, and K_b are the induced voltage, angular velocity, and back emf constant respectively.

A differential equation characterizes the armature current as a function of the armature's inductance and resistance, back emf voltage, and the applied armature voltage as defined by

$$L_a \frac{di_a}{dt} + R_a i_a + e_b = e_a, \quad (3.20)$$

where L_a , R_a , and e_a are the armature's inductance, resistance and applied voltage respectively. Assuming friction is negligible, the torque caused by armature current creates an angular acceleration proportional to the moment of inertia, so that

$$J \frac{d^2\theta}{dt^2} = T = Ki_a, \quad (3.21)$$

where J is the moment of inertia of the motor and its load. Motor dynamics are well documented and can be referenced in dynamics texts [35].

If the armature inductance is neglected, which is reasonable due to its small value,

then Equation 3.20 reduces to

$$i_a = \frac{e_a - e_b}{R_a}. \quad (3.22)$$

This is substituted into Equation 3.21, which is then substituted into Equation 3.19. The resulting equation that defines the applied torques for each gimbal as a function of the motor dynamics is identical for each motor. The DC motor equations for the azimuth and elevation gimbals are

$$T_\psi = \frac{K_\psi}{R_{a_\psi}} \left(e_\psi - K_{b_\psi} \dot{\psi} \right) \quad (3.23)$$

and

$$T_\theta = \frac{K_\theta}{R_{a_\theta}} \left(e_\theta - K_{b_\theta} \dot{\theta} \right), \quad (3.24)$$

respectively.

Equations 3.23 and 3.24 replace the torque terms in Equations 3.15 and 3.16 with applied voltages e_{az} and e_{el} , so that the equations of motion incorporate the inertia properties of the pedestal as well as motor dynamics. These are represented symbolically by

$$\ddot{\psi} = f_1(\psi, \theta, \dot{\psi}, \dot{\theta}, e_\psi, p, q, r, \dot{p}, \dot{q}, \dot{r}) \quad (3.25)$$

and

$$\ddot{\theta} = f_2(\psi, \theta, \dot{\psi}, \dot{\theta}, e_\theta, p, q, r, \dot{p}, \dot{q}, \dot{r}). \quad (3.26)$$

These two equations characterize the internal and external forces of the pedestal. They incorporate the moments of inertia for the entire pedestal and motor dynamics of the gimbals.

3.2 Aircraft Disturbance Spectra Analysis

The purpose of this section is to analyze recorded flight data in order to model the aircraft disturbances. Modeling the disturbances allows the pedestal controller to account for the disturbances and eliminate pointing error from the commanded pointing solution.

3.2.1 Aircraft Flight Profile Data

Lincoln Laboratory operates and maintains operational aircraft to benchmark and rigorously test government sensor and communication systems. The Paul Revere is a heavily modified Boeing 707 that accurately represents an operational Military aircraft that will utilize future MILSATCOM terminals. Lincoln Laboratory employees tested a mobile SATCOM system, on the Paul Revere, during June 2009. The aircraft took off from Hanscom Air Force Base, MA and flew down the Atlantic coast. It then flew several tight racetrack loops just off the coast of New Jersey, while testing the SATCOM system. Figure 3-4 displays the entire flight profile. The onboard GPS/INS recorded the aircraft's position and orientation data for the entire flight. As part of post-flight processing, the flight data was broken into segments of two distinct flight profiles, racetrack and cruise. These are used to characterize system performance during the two distinct aircraft mission profiles. The racetrack data simulates an aircraft performing tight concentric circles over a target area, as in a wartime environment, in order to aid troops on the ground below and enable continuous communication channels to other military assets. The cruise data simulates an aircraft performing steady, level flight traveling from point A to point B while allowing mission essential personnel to stay in contact with forces around the globe.

3.2.2 Aircraft Disturbances

The GPS/INS flight data collected from the flight captures the real aircraft disturbances imposed on all hardware aboard the aircraft, including the pedestal. This recorded data is in an inertial reference frame and must be converted to the aircraft reference frame to reflect the aircraft disturbances accurately. The measurements include the aircraft's yaw, pitch, and roll angles, Ψ , Θ , and Φ respectively, referenced to an inertial reference frame. The derivative of the angles indicate the rate of change in each inertial axis. This rate of angle change is related to the aircraft disturbance by way of the aircraft's current orientation. Describing this relation requires a modified rotational matrix. Each of the three inertial rotation rates is rotated through either zero, one, or two transformations to find the rotational rates of change P , Q , and R , which are the rotation rates in the aircraft's X, Y, and Z-axes respectively. The roll rate is already in the aircraft reference frame, but the pitch rate must be transformed through the aircraft's current roll angle and the yaw rate must be transformed

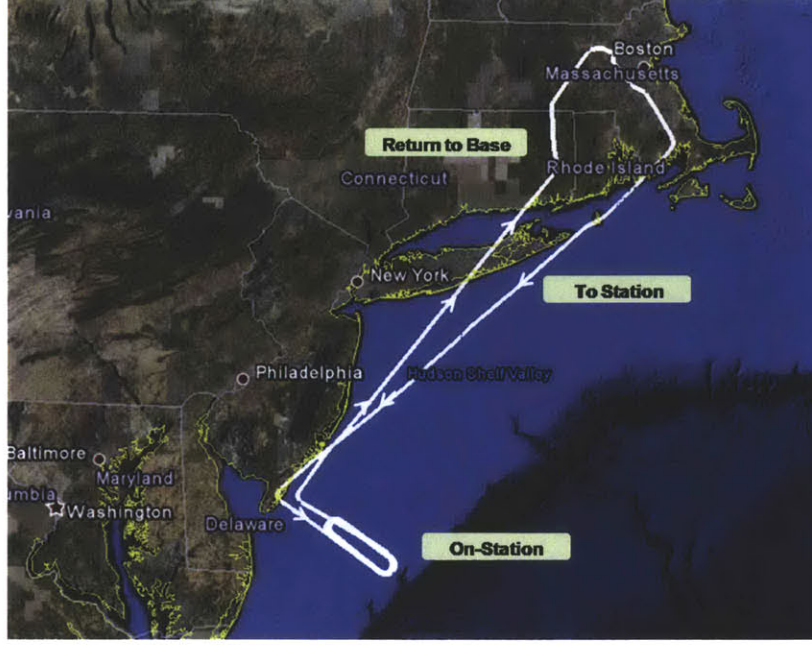


Figure 3-4: Recorded flight data from SATCOM system test. The Paul Revere flew South along the coast and then several racetrack loops “On Station” off the coast of New Jersey. Image created in Google Earth [36].

through the aircraft’s current pitch and roll angles as indicated by

$$\begin{bmatrix} P \\ Q \\ R \end{bmatrix} = R_{\text{roll}}(\Phi)R_{\text{theta}}(\Theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\Psi} \end{bmatrix} + R_{\text{roll}}(\Phi) \begin{bmatrix} 0 \\ \dot{\Theta} \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\Phi} \\ 0 \\ 0 \end{bmatrix}, \quad (3.27)$$

which simplifies to

$$\begin{aligned} P &= -\sin(\Theta)\dot{\Psi} + \dot{\Phi} \\ Q &= \sin(\Phi)\cos(\Theta)\dot{\Psi} + \cos(\Phi)\dot{\Theta} \\ R &= \cos(\Phi)\cos(\Theta)\dot{\Psi} - \sin(\Phi)\dot{\Theta} \end{aligned} \quad (3.28)$$

These three equations represent the rotation rates in the axes fixed to the aircraft’s body and the disturbances the aircraft imposes on the pedestal system [16]. Equation 3.28 converts the GPS/INS flight data into aircraft disturbances in the time domain. With the aircraft disturbances extrapolated from the flight data, the next step is to find the power spectral density of the aircraft disturbances in each axis.

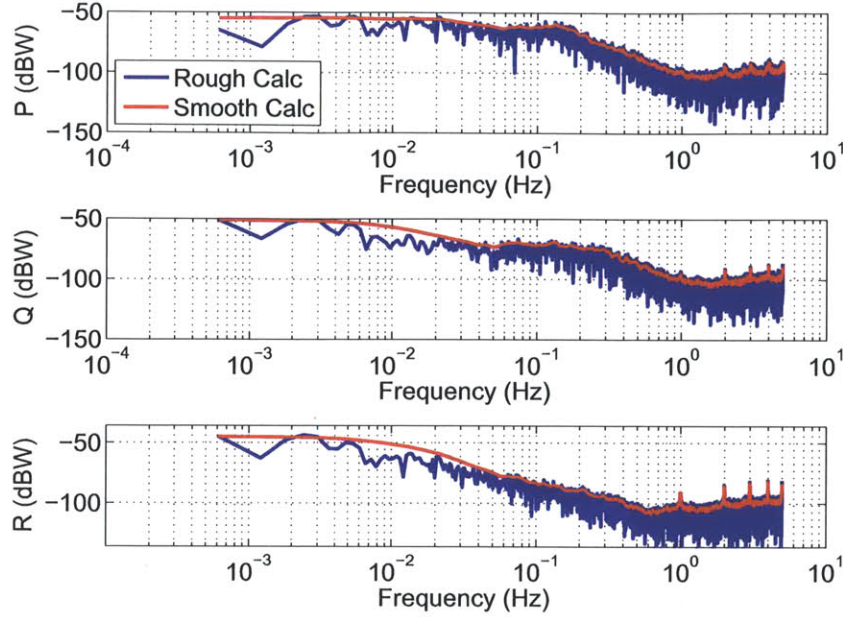


Figure 3-5: PSD of the aircraft disturbances in their respective axes for the racetrack flight data.

3.2.3 Spectral Analysis

The Power Spectral Density (PSD) is the frequency domain representation of a time domain signal's power. The PSD is defined by

$$S_X(j\omega) = \mathfrak{F}[R_X(\tau)] = \int_{-\infty}^{\infty} R_X(\tau) e^{-j\omega\tau} d\tau, \quad (3.29)$$

where $\mathfrak{F}[\cdot]$ indicates the Fourier transform [37], and S_X is the frequency domain interpretation of the signal power in the time domain, which is represented by the autocorrelation function (R_X). The autocorrelation function is defined as

$$R_X(\tau) = E[x(t)x(t+\tau)^T]. \quad (3.30)$$

MATLAB's fast Fourier transform (FFT) function computes the PSD of a discrete time domain signal [38]. Figure 3-5 presents the calculated PSD for the disturbances in each aircraft axis for the racetrack data. Rough calculations are a direct implementation of the FFT and the smooth calculation involve the PSD after passing through a smoothing filter. The important thing to notice is that each disturbance spectra displays impulses at frequencies above 1 Hz. Figure 3-6 is a close-up of this frequency range and clearly demonstrates the impulses at each frequency.

Upon further investigation, it became clear that the harmonics appear when the aircraft is banked and turning in the racetrack pattern. Figures 3-7 and 3-8 verify this

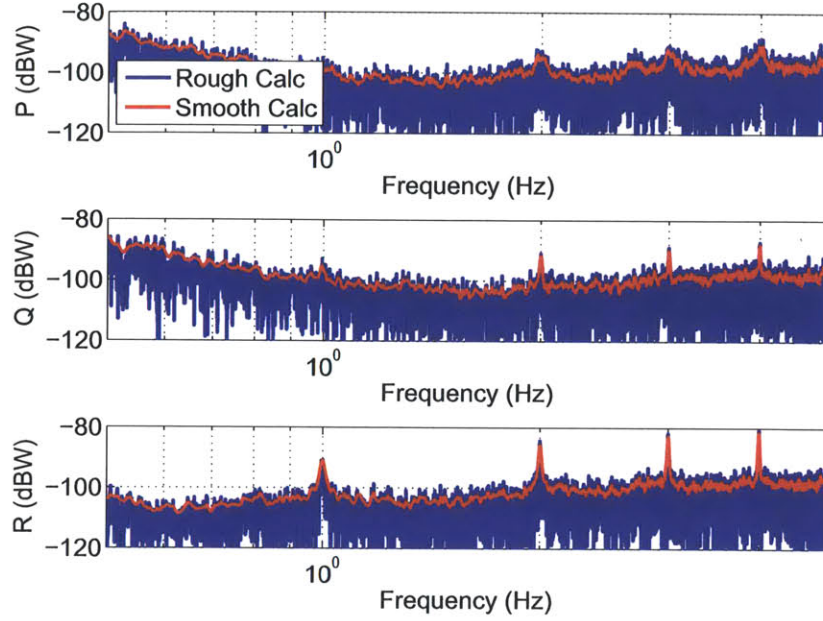


Figure 3-6: A close-up of the aircraft disturbances for the racetrack flight data that demonstrates the harmonics.

conclusion by presenting the disturbance spectra from the cruise flight data, which contain no harmonics.

The goal in calculating the PSD of the aircraft disturbances is to characterize the disturbances such that unity white noise passing through a shaping filter creates a signal that models the disturbances [37]. The shaping filters allow engineers to create simulations of the system without actual flight data. Shaping filters for the cruise flight profile are created and presented in Figure 3-9. The filters for disturbances in each aircraft axis are

$$P_{\text{Filter}} = \left(\frac{2}{5}\right) \frac{(s^2 + 20s + 100)}{(s^2 + 2s + 1)(s^2 + 40s + 400)}, \quad (3.31)$$

$$Q_{\text{Filter}} = \left(\frac{2}{15}\right) \frac{(s^2 + 20s + 100)}{(s^2 + 2s + 1)(s^2 + 40s + 400)}, \quad (3.32)$$

and

$$R_{\text{Filter}} = \left(\frac{1}{1000}\right) \frac{(s^2 + 4s + 4)}{(s^2 + 0.4s + 0.04)} \quad (3.33)$$

These three filters simulate the aircraft disturbance for straight and level flight. The harmonics observed in Figures 3-5 and 3-6 demonstrate that unique disturbances occur in the aircraft while performing a turn and that these filters do not adequately model these disturbances. Although the filters have been defined and are valid for straight, level flight, the recorded flight data is used throughout the rest of this thesis

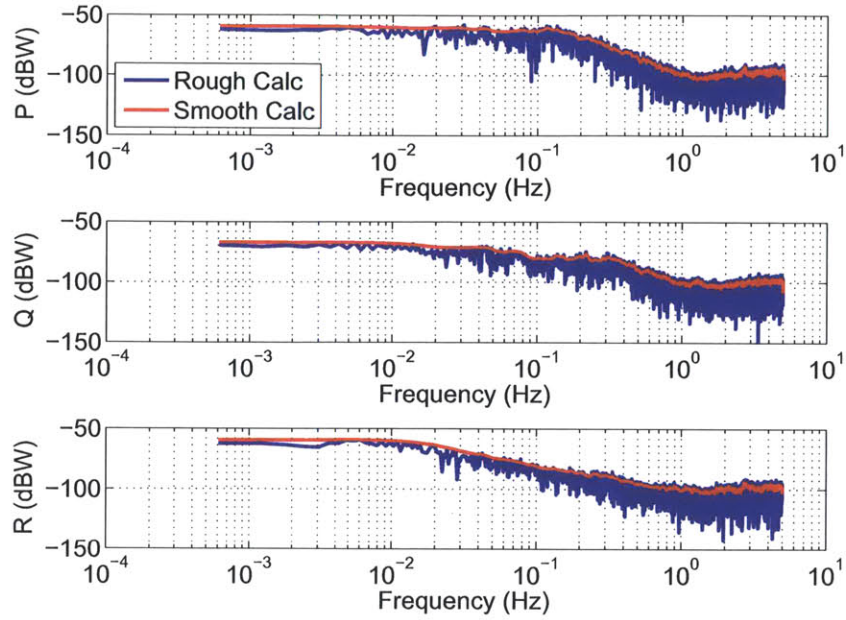


Figure 3-7: PSD of the aircraft disturbances in their respective axes for the cruise flight data.

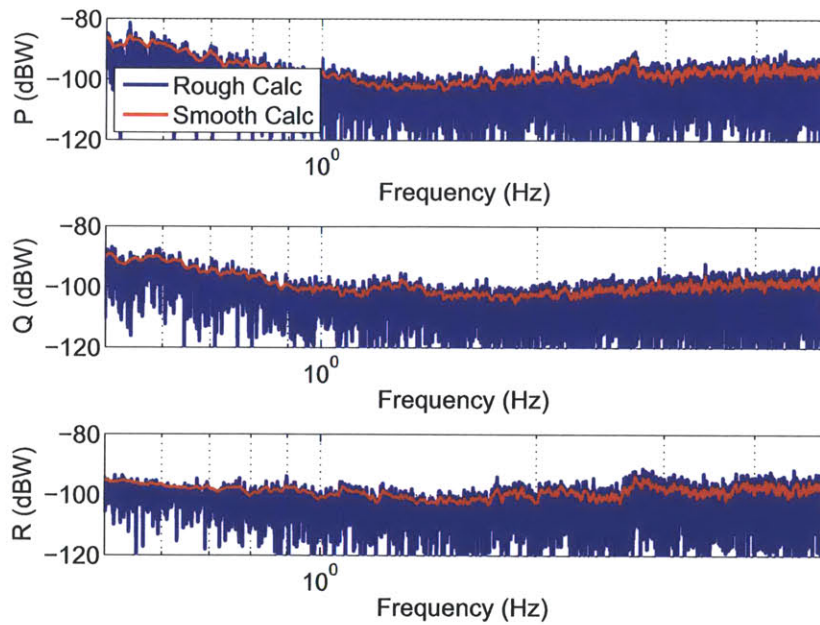


Figure 3-8: A close-up of the aircraft disturbance for the cruise flight data that demonstrates the absence of the harmonics.

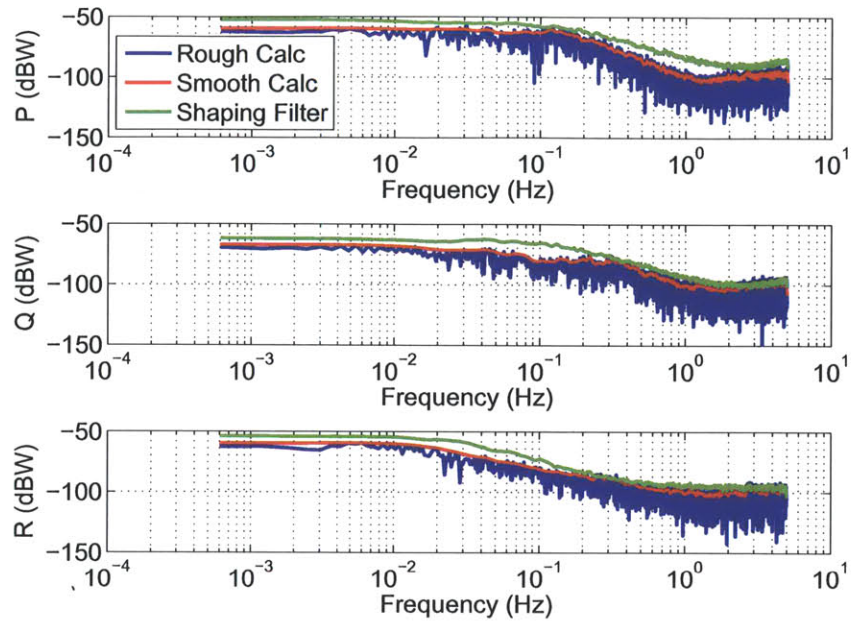


Figure 3-9: Shaping filters to simulate aircraft disturbances during straight, level flight.

to simulate the actual aircraft disturbances. Appendix B.1 presents the MATLAB code used to calculate the rough and smooth PSD for each axis and the shaping filters to simulate the disturbances.

3.3 Control System Analysis

The purpose of this section is to define the control law that allows the pedestal to follow a commanded pointing solution. The derived pedestal and motor dynamics from Section 3.1 are converted into state space form and then linearized around an operating point. The control law maintains the operating point in the presence of aircraft disturbances. The same law is then applied to the nonlinear model and analyzed further.

3.3.1 Linearized Plant Model

The plant model will first be linearized around an operating point to prove that the nonlinear plant can be stabilized to point in a particular direction. Using the techniques described in [39], Equations 3.25 and 3.26 can be formatted into a vector format given by

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad (3.34)$$

where

$$\mathbf{x} = [\psi \quad \dot{\psi} \quad \theta \quad \dot{\theta}]^T \quad (3.35)$$

and

$$\mathbf{u} = [e_{az} \quad e_{el} \quad p \quad q \quad r \quad \dot{p} \quad \dot{q} \quad \dot{r}]^T. \quad (3.36)$$

The first step in linearizing the model is solving for the equilibrium points, which are the set of states that satisfy $\mathbf{f}(\mathbf{x}, \mathbf{u}) = 0$ [40]. This step is accomplished by setting $\dot{\mathbf{x}} = 0$ and $\mathbf{u} = 0$ and solving Equation 3.34 for the azimuth and elevation angles, ψ and θ respectively. After substituting in $\mathbf{u} = 0$ into Equation 3.34, the solution reduces to

$$\ddot{\psi} = \frac{-(I_{xa} - I_{za}) \sin(2\theta) \dot{\theta} \dot{\psi}}{\frac{1}{2}(I_{xa} + I_{za}) + I_{zb} - \frac{1}{2}(I_{xa} - I_{za}) \cos(2\theta)} \quad (3.37)$$

and

$$\ddot{\theta} = \frac{\frac{1}{2}(I_{xa} - I_{za}) \sin(2\theta) \dot{\psi}^2}{I_{ya}}. \quad (3.38)$$

Equation 3.38 demonstrates that after substituting in $\dot{\mathbf{x}} = 0$, both sides of the equation drop to zero. This final result means the set of equilibrium points contains all combinations of azimuth and elevation angles and that the system can be linearized to operate around any given point solution.

The next step is to linearize the model by taking the partial derivative of every equation with respect to each state as indicated by

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{\partial f}{\partial \psi} & \frac{\partial f}{\partial \dot{\psi}} & \frac{\partial f}{\partial \theta} & \frac{\partial f}{\partial \dot{\theta}} \\ 0 & 0 & 0 & 1 \\ \frac{\partial g}{\partial \psi} & \frac{\partial g}{\partial \dot{\psi}} & \frac{\partial g}{\partial \theta} & \frac{\partial g}{\partial \dot{\theta}} \end{bmatrix}_0 \quad (3.39)$$

and

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\partial f}{\partial e_\psi} & \frac{\partial f}{\partial e_\theta} & \frac{\partial f}{\partial p} & \frac{\partial f}{\partial q} & \frac{\partial f}{\partial r} & \frac{\partial f}{\partial \dot{p}} & \frac{\partial f}{\partial \dot{q}} & \frac{\partial f}{\partial \dot{r}} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\partial g}{\partial e_\psi} & \frac{\partial g}{\partial e_\theta} & \frac{\partial g}{\partial p} & \frac{\partial g}{\partial q} & \frac{\partial g}{\partial r} & \frac{\partial g}{\partial \dot{p}} & \frac{\partial g}{\partial \dot{q}} & \frac{\partial g}{\partial \dot{r}} \end{bmatrix}_0. \quad (3.40)$$

A desired operating point is selected and substituted into the previous two matrices, which then form the A and B matrices for the linearized plant model as indicated by

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}. \quad (3.41)$$

For simulation purposes, ψ and θ are both set to 45° to simulate a reasonable commanded operating point for the pedestal to maintain. These state values are inserted into the linearized A and B matrices and the resulting matrix of transfer functions come out to be

$$G(s) = \begin{bmatrix} g_{11} & 0 & 0 & 0 & 0 & g_{16} & g_{17} & g_{18} \\ g_{21} & 0 & 0 & 0 & 0 & g_{26} & g_{27} & g_{28} \\ 0 & g_{32} & 0 & 0 & 0 & g_{36} & g_{37} & 0 \\ 0 & g_{42} & 0 & 0 & 0 & g_{46} & g_{47} & 0 \end{bmatrix}. \quad (3.42)$$

The matrix of transfer functions present two observations. The first is that none of the plant states depend on the aircraft's angular velocities (p , q , and r). The more important realization is that the two gimbals are decoupled and can be separated into individual state vectors as defined by

$$\mathbf{x}_\psi = [\psi \quad \dot{\psi}]^T \quad (3.43)$$

$$\mathbf{u}_\psi = [e_\psi \quad \dot{p} \quad \dot{q} \quad \dot{r}]^T \quad (3.44)$$

$$\dot{\mathbf{x}}_\psi = A_\psi \mathbf{x}_\psi + B_\psi \mathbf{u}_\psi, \quad (3.45)$$

where

$$A_\psi = \begin{bmatrix} 0 & 1 \\ \frac{\partial f}{\partial \psi} & \frac{\partial f}{\partial \dot{\psi}} \end{bmatrix}_0 \quad (3.46)$$

$$B_\psi = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{\partial f}{\partial e_\psi} & \frac{\partial f}{\partial \dot{p}} & \frac{\partial f}{\partial \dot{q}} & \frac{\partial f}{\partial \dot{r}} \end{bmatrix}_0 \quad (3.47)$$

and

$$\mathbf{x}_\theta = [\theta \quad \dot{\theta}]^T \quad (3.48)$$

$$\mathbf{u}_\theta = [e_\theta \quad \dot{p} \quad \dot{q}]^T \quad (3.49)$$

$$\dot{\mathbf{x}}_\theta = A_\theta \mathbf{x}_\theta + B_\theta \mathbf{u}_\theta, \quad (3.50)$$

where

$$A_\theta = \begin{bmatrix} 0 & 1 \\ \frac{\partial g}{\partial \theta} & \frac{\partial f}{\partial \dot{\theta}} \end{bmatrix}_0 \quad (3.51)$$

$$B_\theta = \begin{bmatrix} 0 & 0 & 0 \\ \frac{\partial g}{\partial e_\psi} & \frac{\partial g}{\partial \dot{p}} & \frac{\partial g}{\partial \dot{q}} \end{bmatrix}_0. \quad (3.52)$$

These two sets of state space models model the individual gimbals of the linearized plant model. The method of feedback used to control the models is a simple proportional differential (PD) controller, which calculates the error between a reference command and the actual output. The error and the derivative of the error are fed into the controller, which then sends control inputs into the plant in an effort to control the system's response. The compensator transfer function is defined as

$$G_c(s) = \frac{M(s)}{E(s)} = k_P + k_D s. \quad (3.53)$$

The key design factor is to make the closed-loop pole locations stable. Otherwise, the system runs the risk of becoming unstable if a large disturbance occurs. The proportional and differential gains, k_P and k_D respectively, determine the closed-loop pole locations and the system's response. It is important to note that the two state space models have at least 3 inputs, one of which is the voltage input that controls the motor torque and the other inputs are aircraft disturbances entering the system.

The open-loop transfer functions between the voltage input and the gimbal angular acceleration are defined by

$$G_\psi(s) = \frac{\ddot{\psi}}{e_\psi} = \frac{0.01345}{s^2 + 1.773s} \quad (3.54)$$

and

$$G_\theta(s) = \frac{\ddot{\theta}}{e_\theta} = \frac{0.03677}{s^2 + 4.846s}. \quad (3.55)$$

Both transfer functions are marginally stable due to the pole at zero in the denominator, but the PD controller pulls the closed-loop poles away from the imaginary axis and creates a stable response. As the poles pull further away, the system's commanded response becomes faster, but after some threshold the system cannot respond as fast as the control commands. Therefore the proportional and differential gains must be balanced in order to produce a stable response with good transient and steady state characteristics.

A MATLAB Simulink model simulates the linearized state space models to determine proper gain settings to create a stable system that tracks changes in the reference command. The Simulink model takes the recorded flight data and simulates how well the system will track the commanded trajectory. Figure 3-10 presents the simulation performance between the output and the commanded reference. The figure demonstrates that the linearized plant model tracks the commanded trajectory for the simulation. Figure 3-11 presents the total pointing error, which is the root mean square of the errors in the azimuth and elevation axes in Figure 3-10. The figure demonstrates that the total pointing error between the reference and actual

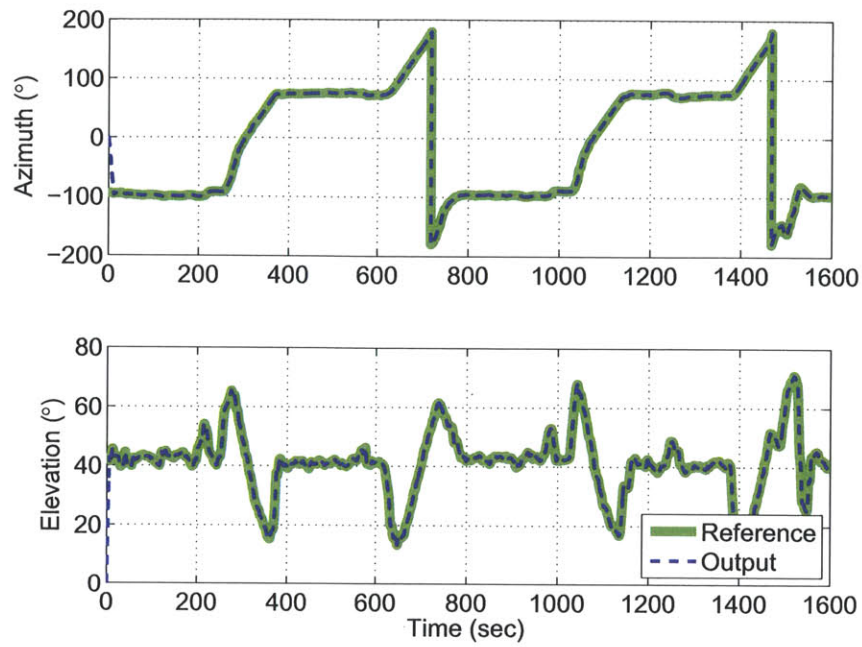


Figure 3-10: Linearized model simulation performance comparison between the commanded and system output. Both the azimuth and elevation gimbal track the commanded inputs accurately.

never exceeds 0.25° , and stays well below 0.1° for the majority of the simulation. Appendix B.2 presents the MATLAB code that simulates the linearized plant model.

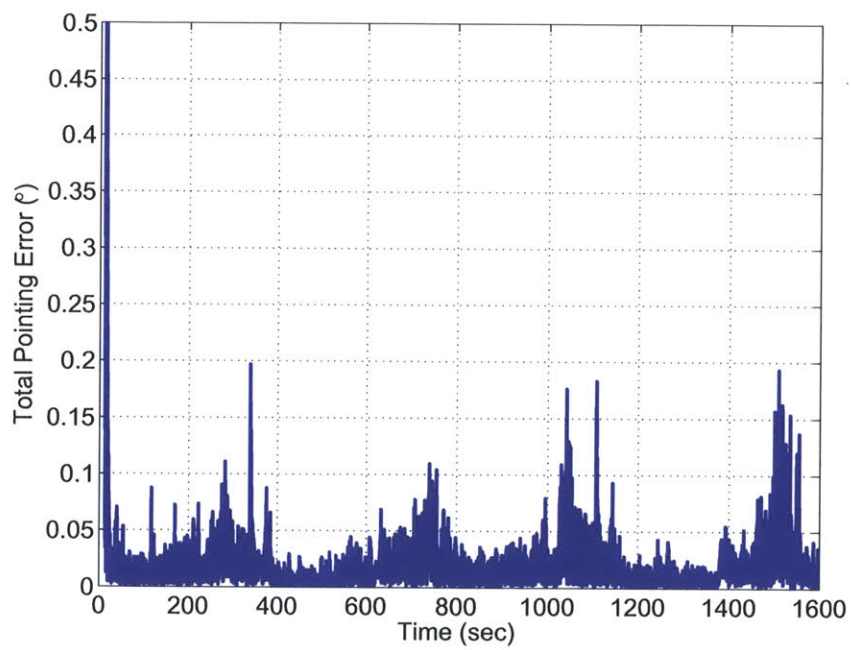


Figure 3-11: Linearized model simulation performance of total pointing error. The error remains within 0.2° during the simulation.

3.3.2 Nonlinear Plant Model

The same PD controller from the linearized model is tested with the full nonlinear plant to see if the pedestal still follows the commanded pointing solution. The nonlinear plant incorporates the coupling dynamics between the two gimbals as well as the nonlinearities caused by trigonometric functions in the derivation. Angular resolver sensors and the gyros mounted to the elevation gimbal are the only feedback mechanisms for the pedestal controller. The angular resolvers feedback the gimbals' orientations (ψ and θ) directly to the pedestal controller. On the other hand, the gimbals' rotational velocity data (q_a and r_a) must be calculated by the GPS/INS yaw and pitch angles (ψ and θ) and the aircraft disturbances (p , q , and r). The transformation is defined by

$$\dot{\psi} = (r_a - \sin(\theta)) \frac{p \cos(\psi) + q \sin(\psi)}{\cos(\theta)} - r \quad (3.56)$$

and

$$\dot{\theta} = q_a - q \cos(\psi) + p \sin(\psi), \quad (3.57)$$

which come from the equations of motion derivation code in Appendix A. The output of this estimator is fed back to the PD controller, which then calculates the control signal sent to the motors.

The results from the simulation demonstrate that the PD controller allows the pedestal to follow a commanded pointing solution despite the complex, nonlinearities within the model. As indicated by [40], complex models can often be stabilized by simple controllers. Figure 3-12 presents the comparison between the output and the commanded reference. Figure 3-13 demonstrates the realtime pointing error of the simulation and reveals a problem within the nonlinear model. The pointing error is similar to the data from Figure 3-11, but the steady state error grows over time. Upon further investigation, the angular acceleration measurements are the root cause of the growing error. Due to the aircraft disturbances, the acceleration measurements are integrated over time and accumulate error, which exaggerates the estimated differential error from Equations 3.56 and 3.57. This inaccurate measurement causes the plant controller to compensate for a nonexistent error.

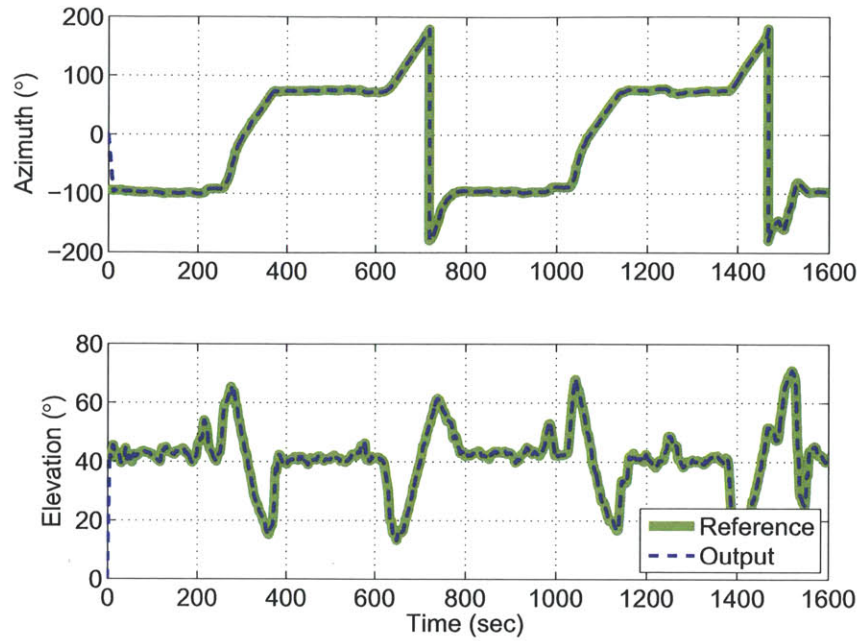


Figure 3-12: Nonlinear model simulation performance comparison between the commanded and system output. As seen in the linearized simulation, both the azimuth and elevation gimbal track the commanded inputs accurately.

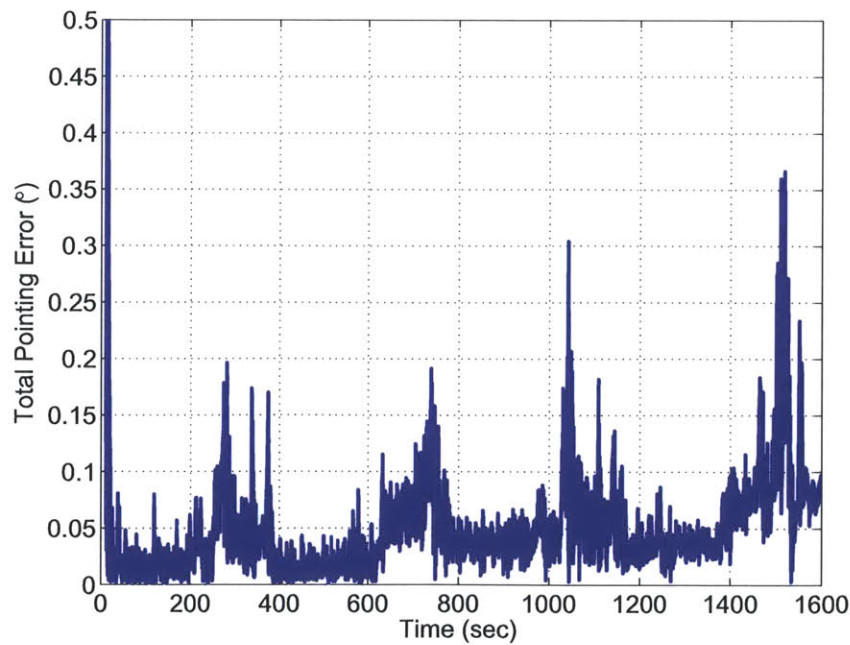


Figure 3-13: Nonlinear model simulation performance of the pedestal's total pointing error with the PD controller. The steady state error increases over time and is not a desirable system response.

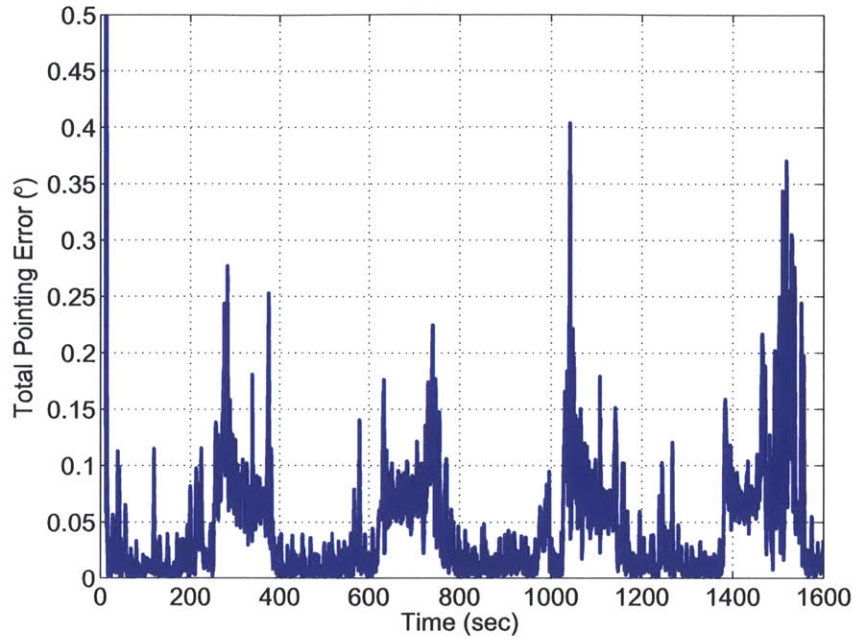


Figure 3-14: Nonlinear model simulation performance of the pedestal's total pointing error with the P controller. The steady state error does not accumulate and the total error stays below 0.35° during the simulation

The solution to this problem is to eliminate the differential gain in the controller and only have a proportional gain. Figure 3-14 presents the new pedestal performance error, which has no accumulating steady state error previously caused by the differential gain. The end result is a pedestal that tracks the commanded trajectory with a total error never exceeding 0.35° and remaining below 0.2° the majority of the time. Appendix B.3 presents the MATLAB code that simulates the nonlinear plant model.



Figure 3-15: Aircraft reference frame is fixed to the aircraft body and points out the nose of the aircraft. The antenna pedestal and GPS/INS are mounted and aligned to the same frame.

3.4 Open-loop Pointing Error Analysis

Section 2.4 defined the major errors associated with open-loop pointing. Careful design and fabrication of the pedestal reduces the majority of these errors. The errors imposed by the GPS/INS cannot be eliminated and must be analyzed to determine their impact on the open-loop pointing error.

3.4.1 Problem Definition

A GPS/INS supplies position and orientation information in a specified reference frame. The position information is used to calculate an inertial pointing vector between the platform and its intended target and then the orientation information is used to translate that pointing vector into the aircraft's reference frame. Figure 3-15 depicts the antenna pedestal and GPS/INS location in reference to the aircraft reference frame. Both are aligned with the aircraft frame.

Position errors are trivial due to the distance between antenna and satellite, but orientation errors increase pointing error directly. Each GPS/INS has a random error in the output of each Euler angle that is modeled as a zero mean, Gaussian random variable with some variance. GPS/INS units are graded on this precision and higher grade GPS/INS units have lower variance, but are more complex and expensive. Table 3.1 identifies typical values for four different grade GPS/INS units.

Table 3.1: Standard deviations ($1\text{-}\sigma$) for 4 different grade GPS/INS packages

	Package 1	Package 2	Package 3	Package 4
Yaw	5 mrad	2 mrad	1 mrad	0.7 mrad
Pitch	2.5 mrad	1 mrad	0.5 mrad	0.35 mrad
Roll	2.5 mrad	1 mrad	0.5 mrad	0.35 mrad

An inertial pointing vector in North, East, Down (NED) reference frame can be expressed in the aircraft's reference frame by transforming the inertial pointing vector by three rotation matrices corresponding to the three Euler angles as defined by

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{AC} = R_{roll}(\Phi)R_{pitch}(\Theta)R_{yaw}(\Psi) \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{NED}. \quad (3.58)$$

With the pointing vector is in the aircraft frame, it is converted into pedestal look angles

$$\psi = \tan^{-1}(y_{AC}/x_{AC}) \quad (3.59)$$

and

$$\theta = \sin^{-1}\left(-z_{AC}/\sqrt{x_{AC}^2 + y_{AC}^2 + z_{AC}^2}\right). \quad (3.60)$$

The pedestal control computer commands the pedestal to the calculated azimuth and elevation angle. The order of the rotation matrices is critical and an error in one angle translates into an error in the final pointing solution.

The GPS/INS unit has random errors in the Euler angle measurements, which are modeled as independent, zero mean, Gaussian random variables. The variance of each random variable corresponds to the values defined in Table 3.1. The purpose of this section is to characterize the open-loop pointing error as a function of the GPS/INS error, defined by

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{AC} = R_{roll}(\Phi + \delta_3)R_{pitch}(\Theta + \delta_2)R_{yaw}(\Psi + \delta_1) \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{NED}. \quad (3.61)$$

3.4.2 Pointing Error Closed Form Analysis

The purpose of this characterization is to understand how random errors (σ_1 , σ_2 , and σ_3) impact the final pointing solution in the aircraft reference frame as defined in Equation 3.61. The first attempt at characterizing the pointing error involved tracking the random variables through the three rotation matrices. This derivation proved intractable due to the nonlinearities involved with each transformation. At present, there appears to be no closed-form, linear transformation solution for this problem.

An alternative method to characterize the errors is to perform Monte Carlo simulations, which are multiple iterations of a simulation with changing parameters. Observations from the simulations are made in an effort to understand what is happening. To simulate and characterize the random errors, a series of random variable simulations were completed with different inertial pointing vector and aircraft orientation configurations. Each simulation calculated the final azimuth and elevation variance and their correlation coefficient in an effort to characterize the open-loop

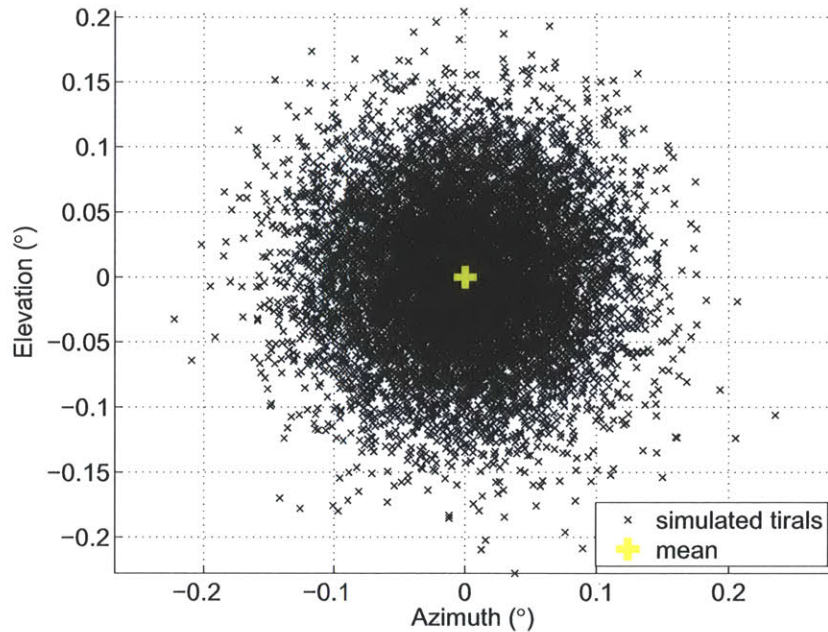


Figure 3-16: Final pointing solution in the aircraft's reference frame with equal variance and zero correlation.

pointing error. No solid conclusions can be made from the simulations, but a few common trends appear among the simulation results.

The first observable trend is that there are instances of the azimuth and elevation errors being uncorrelated and their distribution having equal variance as depicted in Figure 3-16. This first figure is an instance where the final pointing solution depends only on the variances of two of the random variables and the third is ignored because that axis is aligned with the boresight of the antenna and rotation in this axis is irrelevant because the antenna pattern is circularly symmetric.

In other instances the azimuth and elevation errors are uncorrelated and their distributions have unequal variance as demonstrated by Figure 3-17. In contrast to Figure 3-16, the azimuth error distribution is wider, while the elevation error distribution stays the same.

There also are instances of the distributions having unequal variance and the random variables having strong correlation as indicated by Figure 3-18. In contrast to Figure 3-17, the total pointing distribution is rotated by some angle, which corresponds to the correlation between the random variables.

Despite these observable trends and the data collected from the Monte Carlo simulations, the closed form analysis does not characterize the open-loop pointing error as a function of the inertial pointing vector and aircraft's orientation. Clearly there are instances where the azimuth and elevation distributions have equal variance and other times when the two random variables are strongly correlated. Characterizing

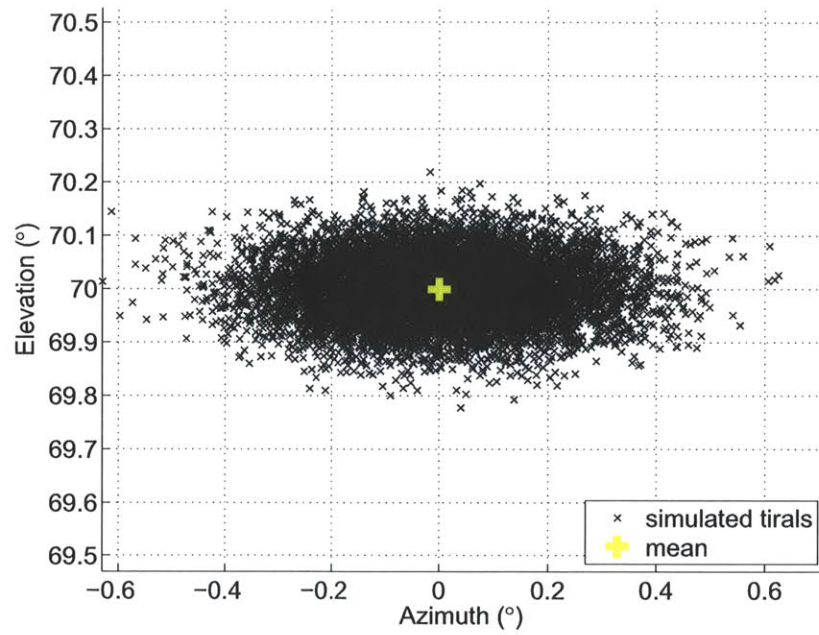


Figure 3-17: Final pointing solution in the aircraft's reference frame with unequal variance and zero correlation.

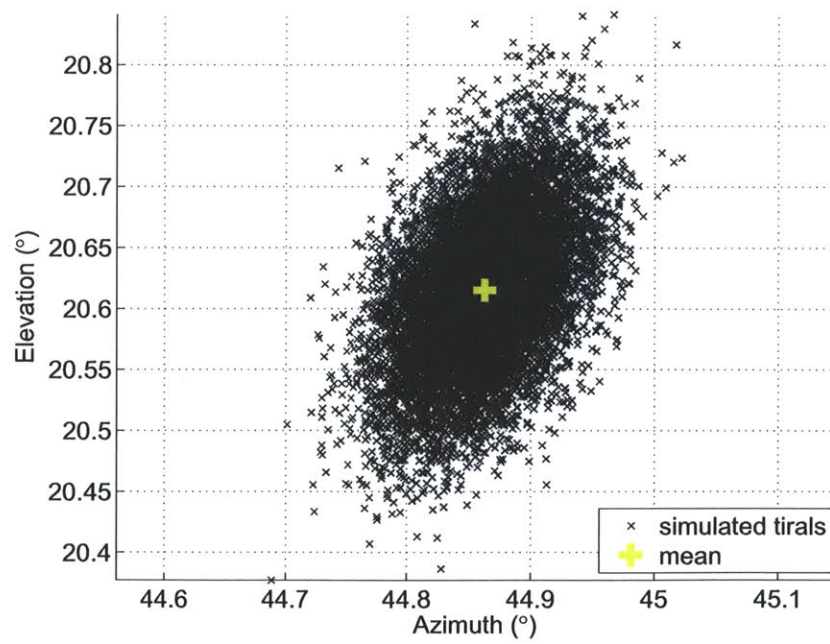


Figure 3-18: Final pointing solution in the aircraft's reference frame with unequal variance and strong correlation.

what conditions directly cause this result is still unclear. Since a closed form solution cannot be found theoretically or through simulation, the next characterization attempt uses small angle approximations to eliminate the complexity imposed by the trigonometric functions.

3.4.3 Small Angle Pointing Error Approximation

Richard Pio analyzed the errors in three consecutive Euler angle transformations [41]. Pio identifies the three previously defined Euler angle rotation errors as well as six additional errors from nonorthogonality existing between the three axes. Through his analysis, he concludes that the errors within each rotation transformation can be pulled out of each matrix and formulated into a separate error matrix defined by

$$R_{\text{error}}(\delta_1, \delta_2, \delta_3, \Phi, \Theta) = \begin{bmatrix} 1 & \epsilon_3 & -\epsilon_2 \\ -\epsilon_3 & 1 & \epsilon_1 \\ \epsilon_2 & -\epsilon_1 & 1 \end{bmatrix} \quad (3.62)$$

where

$$\begin{aligned} \epsilon_1 &= -\delta_1 \sin(\Theta) + \delta_3 \\ \epsilon_2 &= \delta_1 \cos(\Theta) \sin(\Phi) + \delta_2 \cos(\Phi) \\ \epsilon_3 &= \delta_1 \cos(\Theta) \cos(\Phi) + \delta_2 \sin(\Phi) \end{aligned} \quad (3.63)$$

Using Pio's approximation, Equation 3.61 can be rewritten as

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\text{AC}} = R_{\text{error}}(\delta_1, \delta_2, \delta_3, \Phi, \Theta) R_{\text{roll}}(\Phi) R_{\text{pitch}}(\Theta) R_{\text{yaw}}(\Psi) \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\text{NED}} \quad (3.64)$$

Pio assumes the small angle approximation is valid, which lets him linearize the errors within the trigonometric functions and simplify the problem. Pio's approximation is related to the open-loop pointing with Euler angle errors, but the approximation cannot be applied until the small angle approximation is analyzed and simulations demonstrate that the approximation holds for the range of potential errors within the context of this problem.

A series of MATLAB simulations were conducted to test the validity of Pio's approximation. For each simulation, a maximum error was set and inserted into each δ in Equations 3.61 and 3.64. This δ corresponds to the $3\text{-}\sigma$ value of the random variable errors in each axis. For this analysis, the maximum variance of the worst GPS/INS unit is 5 mrad, which corresponds to a $3\text{-}\sigma$ of 0.8594° . Samples across the full range of possible inertial pointing vectors and Euler angle combinations were inserted into each equation to observe the greatest difference between the true error and the Pio's approximation.

After a series of tests, two trends become apparent. The first is that the approximation error in azimuth increases as the true elevation angle increases, while the approximation error in elevation is uniform for all pointing solutions. This observa-

tion means the approximation fails at high elevation angles, but this observation is expected and easily explained due to the keyhole problem explained in Section 2.1. When small orientation errors enter the system at large elevation angles, the errors in azimuth increase significantly. Because the original calculation fails in the keyhole, it makes sense that the approximation also fails. If the keyhole region is avoided, as the pedestal is designed to do, then the approximation holds.

The other trend is that certain initial pointing vector and aircraft orientation combinations cause Pio's approximation to completely fail. Examining a specific case is the best way to understand this rare phenomena. Given an initial pointing vector solely in the positive X -axis, the X -component of the final solution is equal to

$$X = \cos(\Theta + \Delta) \cos(\Psi + \Delta) \quad (3.65)$$

and approximately equal to

$$X \approx \cos(\Theta) \cos(\Psi) - \Delta \sin(\Theta + \Psi). \quad (3.66)$$

This equation is a direct implementation of the small angle approximation, but it clearly demonstrates that the second term, which is the error approximation, becomes zero when certain pitch and yaw combinations occur. If the aircraft is pitched 90° and yawed 90° , then both terms of the approximation go to zero, which cause a serious approximation error if Δ is large. This example demonstrates an instance where the Pio approximation's solution is equal to the solution without errors and the errors essentially fall out of the approximation. This failure occurs in final azimuth calculations at large elevation angles with specific inertial pointing vectors and aircraft orientations combinations. They only occur at large yaw and pitch angles that result in high elevation angles, so the Pio approximation is an acceptable approximation of the actual open-loop pointing solution as long as the aircraft's pitch is restricted to $\pm 80^\circ$ and the final pointing solutions within the keyhole region are ignored.

As validation to the approximation and to prove that the approximation is reasonable for the possible errors, Figure 3-19 presents the maximum observed difference between the actual azimuth error and the approximation.

Figure 3-20 presents the corresponding actual azimuth error caused by the errors in the GPS/INS. Figure 3-21 is the difference normalized by the actual error. The interesting observation is that Figure 3-21 demonstrates the approximation actually gets worse at lower elevation angles, but the absolute error presented in Figure 3-20 is extremely small within this range. The Pio approximation is valid for this open-loop pointing error analysis.

As a final example, Figure 3-22 presents a side-by-side comparison between the full calculation and Pio's approximation of the open-loop pointing error distribution. Clearly, Pio's approximation is a valid simplification of the original calculation.

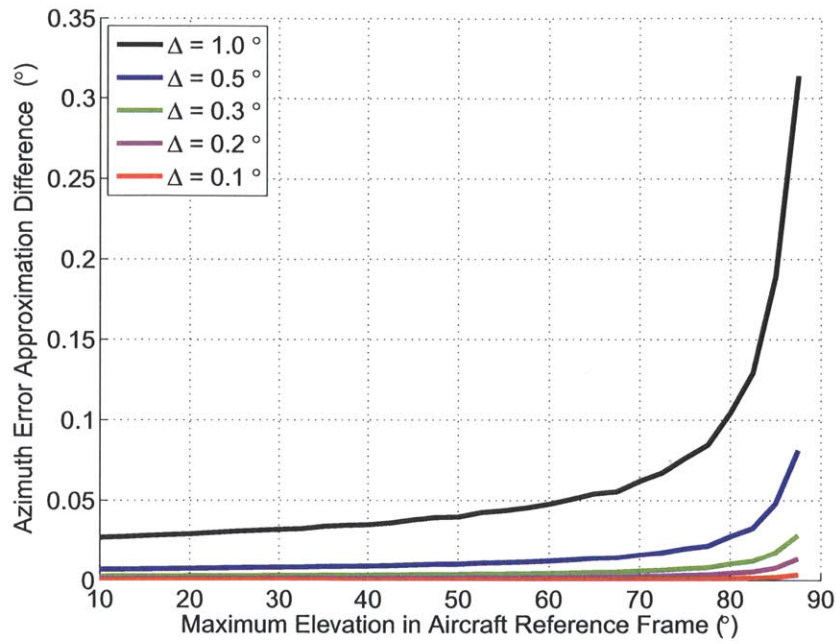


Figure 3-19: Maximum observed difference between the actual azimuth error and the approximation. As the maximum acceptable elevation increases, the azimuth approximation error increases.

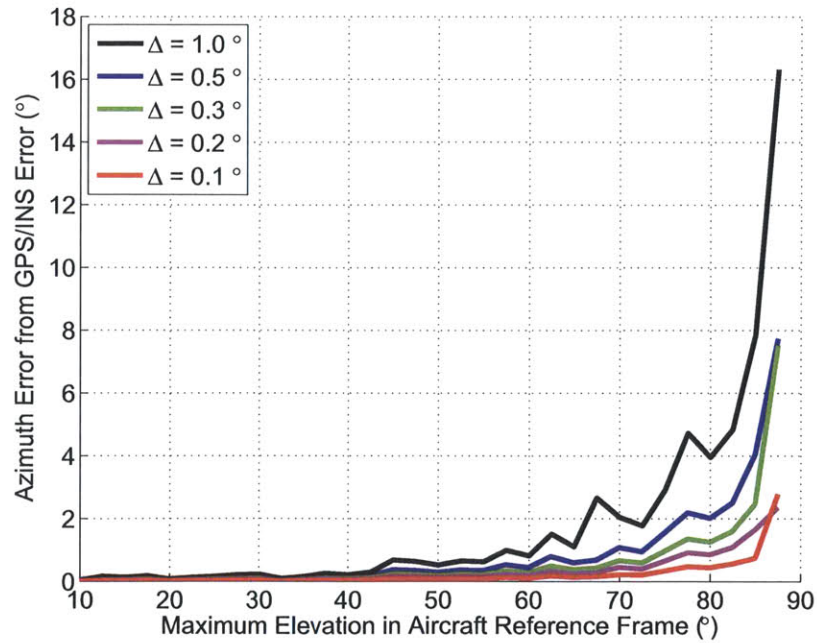


Figure 3-20: Corresponding difference between the azimuth and azimuth with error. This is the true error that increases as the maximum acceptable elevation increases.

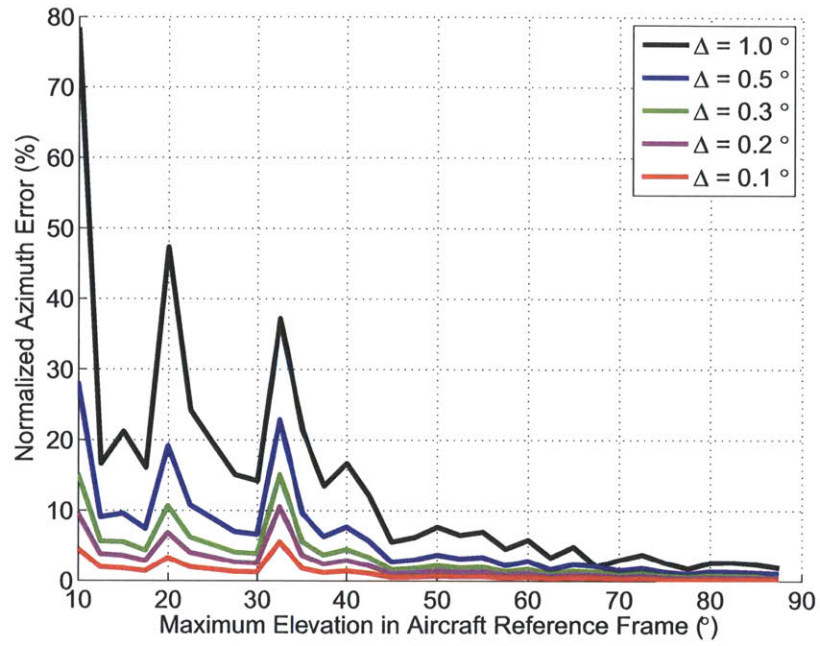


Figure 3-21: Normalized error between the approximation error and the true error. The ratio increases as the maximum acceptable elevation decreases.

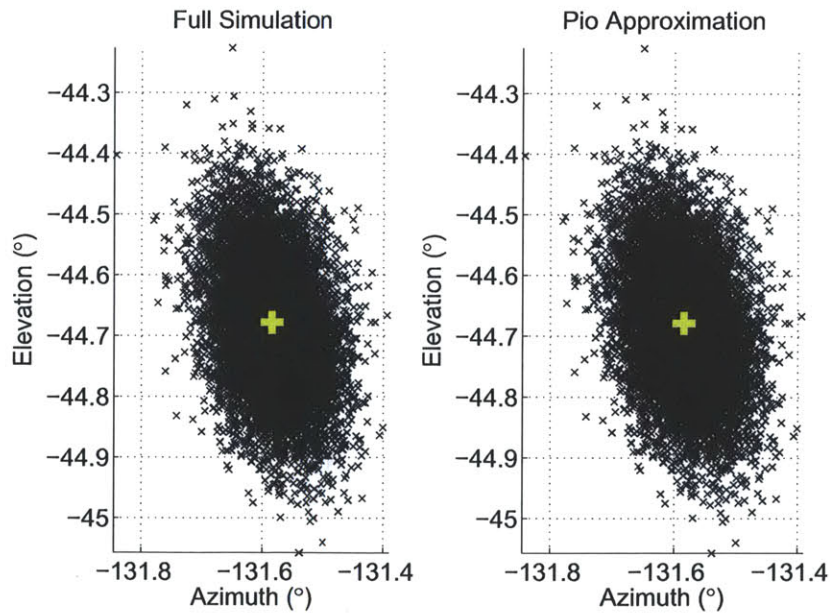


Figure 3-22: Comparison between the complete pointing solution error distribution and the Pio approximation error distribution. The Pio approximation is nearly identical to full computation, but with a simpler computation.

3.4.4 Pointing Error Look Up Tables

Pio's conclusion, as defined in Equation 3.64, states that the errors within the three rotation matrices can be removed from the matrices and combined into a separate error matrix. This conclusion reveals an important fact about the open-loop pointing error distribution: the final distribution is independent of the inertial pointing vector and the yaw angle. The error matrix defined in Equation 3.63 depends only on the noise components in each axis and the current roll and pitch angles. The pointing solution error is found by taking the pointing solution in the aircraft reference frame and then multiplying it through the error matrix. Therefore the distribution on the azimuth and elevation depend on the variance of the noise for each axis, the aircraft's current pitch and roll, and the pointing vector in the aircraft's reference frame.

Another series of Monte Carlo simulations analyzed the pointing distribution with different pitch, roll, and pointing vector combinations. As before, each simulation calculated the azimuth and elevation variances and their correlation coefficient. The results from the simulation are compiled into a look up table defined for a specific pitch and roll combination.

Figures 3-23 and 3-24 present sample contour plots of the look up table data. Each figure displays the azimuth and elevation variances and their correlation coefficient as a function of the final pointing solution in the aircraft reference frame. The difference between the figures is the aircraft's current roll and pitch combination. Figure 3-23 presents the open-loop pointing error when the aircraft is flying straight and level. The upper graph demonstrates that the error in azimuth increases substantially as the elevation angle increases, but is independent of the azimuth angle. The middle graph demonstrates that elevation error is nearly constant for all azimuth and elevation angles. The lower graph shows that there is no clear correlation between the azimuth and elevation errors.

In contrast, Figure 3-24 presents the error distributions when the aircraft is pitched and rolled 80° . This roll/pitch combination is an unrealistic orientation outside the range of motion of a typical aircraft. It is presented to demonstrate that the error distribution does change as a function of the aircraft's pitch and roll. The upper graph shows the azimuth angle error depends on the elevation angle as before, but also depends on the azimuth angle. The middle graph shows that the elevation angle error now depends on the azimuth angle and the lower graph indicates that for given regions within the range of possible azimuth and elevation angles, pockets of positive and negative correlation exist as first observed in Figure 3-18. Upon further investigation of the look up table data, the aircraft's pitch angle heavily influences the changes in the pointing error distribution. In fact, when the pitch is fixed to zero and the aircraft rolls between $\pm 80^\circ$, no real change occurs in the distribution. The difference only occurs when the pitch angle changes and then the aircraft's roll angle also affects the distribution.

The purpose in graphing the look up tables is to understand the open-loop pointing error distribution as a function of the random errors within the Euler angle measure-

ments from the GPS/INS. Pio's approximation separates the random errors from the calculation and demonstrates that the error distribution depends on the aircraft's current pitch and roll angles and the pedestal's pointing vector in the aircraft reference frame. Given the 707's range of in-flight dynamics, the aircraft's pitch and roll never exceed $\pm 10^\circ$ and $\pm 30^\circ$ respectively. Using this range of motion as the standard, the look up table data is graphed and analyzed to understand the range of possible open-loop pointing errors. The data demonstrates that errors in azimuth and elevation are reasonable as long as operation within the keyhole is avoided and that little correlation exists between axes and they can be regarded as independent.

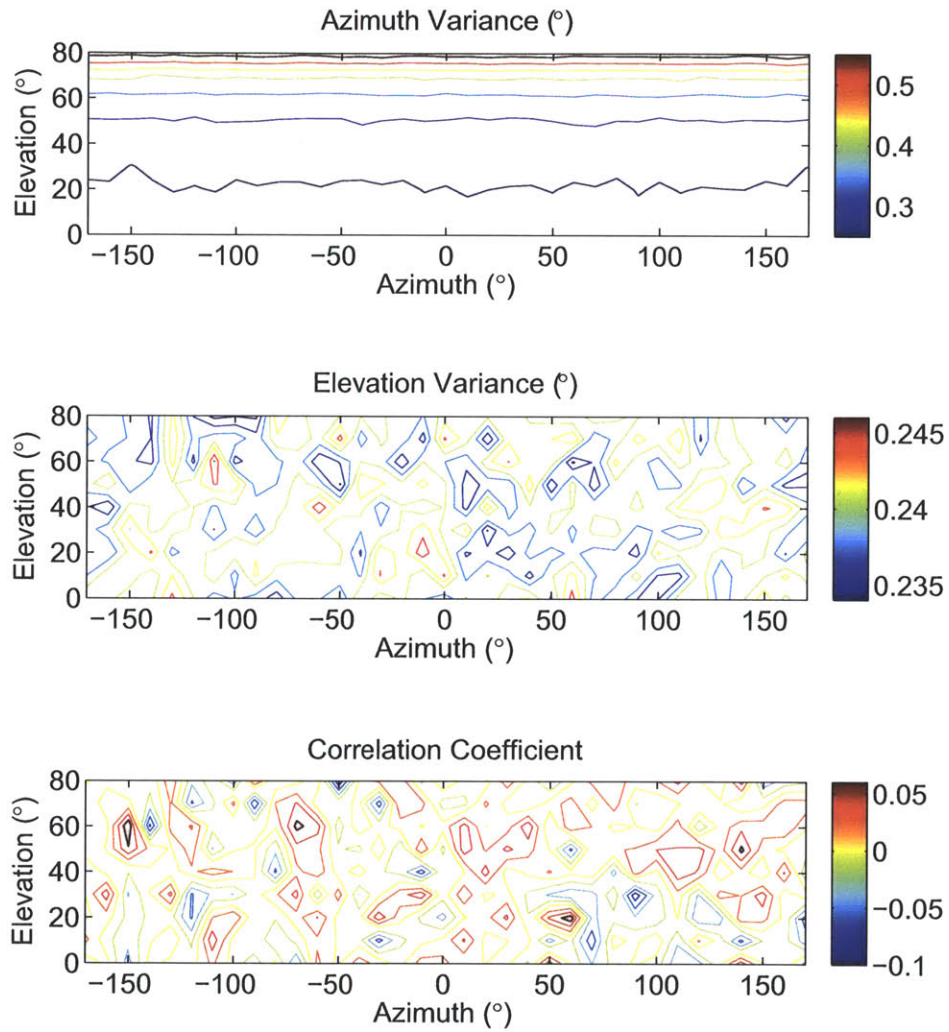


Figure 3-23: Graphical representation of the look up table data of the open-loop pointing error when the aircraft is pitched 0° and rolled 0° . There is very little change in the elevation variance and correlation coefficient and the azimuth variance increases as the elevation increases.

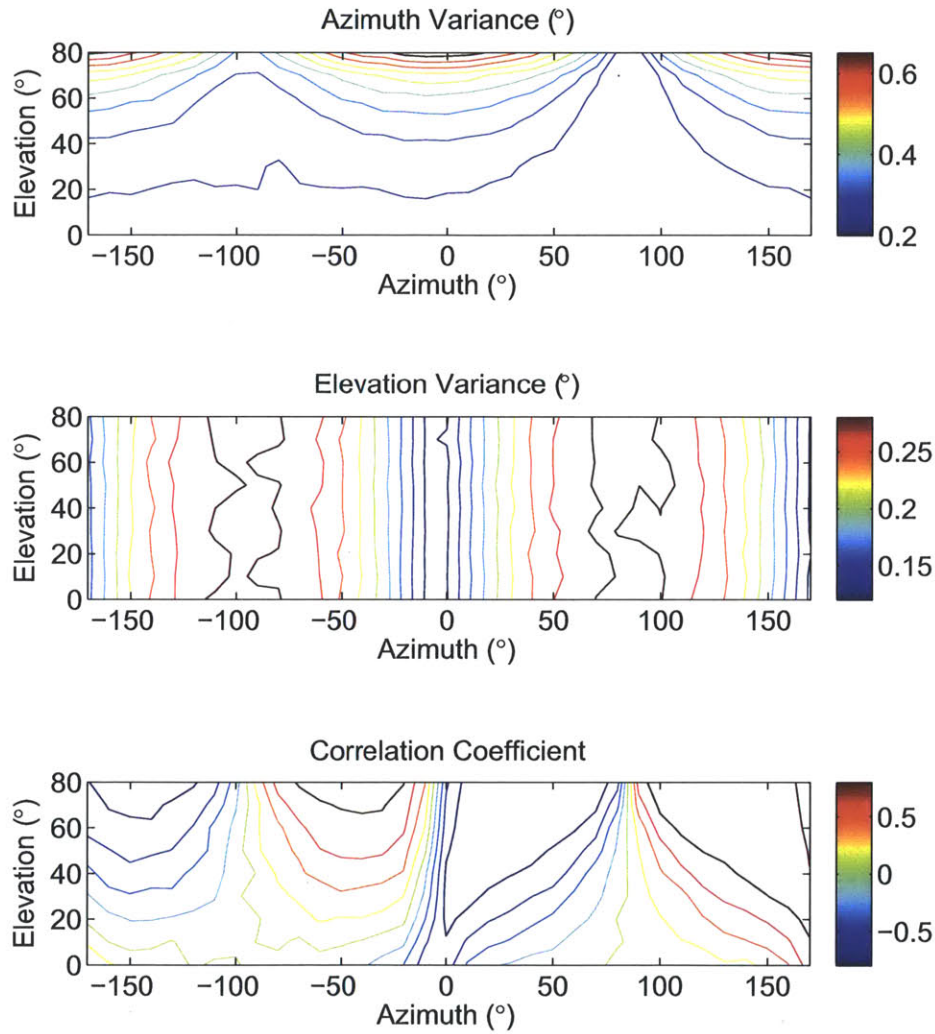


Figure 3-24: Graphical representation of the look up table data of the open-loop pointing error when the aircraft is pitched 80° and rolled 80° . All three graphs demonstrate a dependence on the azimuth angle and the lower graph demonstrates pockets of positive and negative correlation.

3.4.5 Open-loop Pointing Error Distribution

In the analysis presented in [42], Schodorf characterizes the pointing error performance of a land mobile SATCOM system. His analysis encapsulates all of the pointing error factors including disturbances imposed by the environment and pointing errors within the antenna pedestal. The same statistical process can characterize the open-loop pointing error caused by the INS as long as the random variables in each axis are independent, Gaussian distributions. Given the analysis from the look up tables, this assumption is reasonable for the 707's range of motion, but it would not be valid for an aircraft with a greater range in pitch and roll as the distributions demonstrate correlation for certain situations.

The analysis in [42] presents two cases, the first is when both variances are equal and the latter when the variances are unequal. The goal is to determine the PDF of the total pointing error, $f_\rho(\rho)$. Given independent, Gaussian random variables ρ_ψ and ρ_θ , which are the pointing error distributions in each axis, the total pointing error is the root mean square of the two errors ($\rho = \sqrt{\rho_\psi^2 + \rho_\theta^2}$). When the variances of each random variable are equal ($\sigma_\psi = \sigma_\theta$), the resulting distribution is a Rayleigh distribution. For this problem, equal variance is possible yet a rare occurrence. The more practical case is when the distributions have unequal variances ($\sigma_\psi \neq \sigma_\theta$). The derivation presented in [42] involves making a change in random variables, substituting in characteristic functions, taking the inverse of the result, and performing another change in variables. The final distribution is

$$f_\rho(\rho) = \frac{\rho}{\sigma_\psi \sigma_\theta} e^{-\rho^2(\sigma_\psi^2 + \sigma_\theta^2)/4\sigma_\psi^2 \sigma_\theta^2} I_0 \left(\frac{\sigma_\psi^2 - \sigma_\theta^2}{4\sigma_\psi^2 \sigma_\theta^2} \rho^2 \right), \quad (3.67)$$

where $I_0(\cdot)$ is a zeroth order modified Bessel function of the first kind [42]. The calculated variances from Pio's approximation are substituted into Equation 3.67 and the resulting distribution defines the open-loop pointing error for a given pointing vector due to errors in the GPS/INS, for a particular inertial pointing vector and the aircraft attitude combination.

A MATLAB Simulink model verifies the open-loop pointing error distribution. Figure 3-25 presents the final result of the simulation. The simulation performs the complete pointing calculation with GPS/INS errors and plots the set of trials in the top graph in reference to the true pointing solution. The lower graph presents the cumulative distribution function (CDF) of the simulated data and compares it to the theoretical distribution based on Pio's approximation and Schodorf's distribution. A Kolmogorov-Smirnov (KS) test is performed between the two distributions to determine if the theoretical distribution should be rejected. This test defines the error between the two distributions as

$$D_n = \sup |F_n(x) - F_0(x)|. \quad (3.68)$$

The KS test rejects the hypothesis ($H_0 : F(x) = F_0(x)$) with a level of confidence α

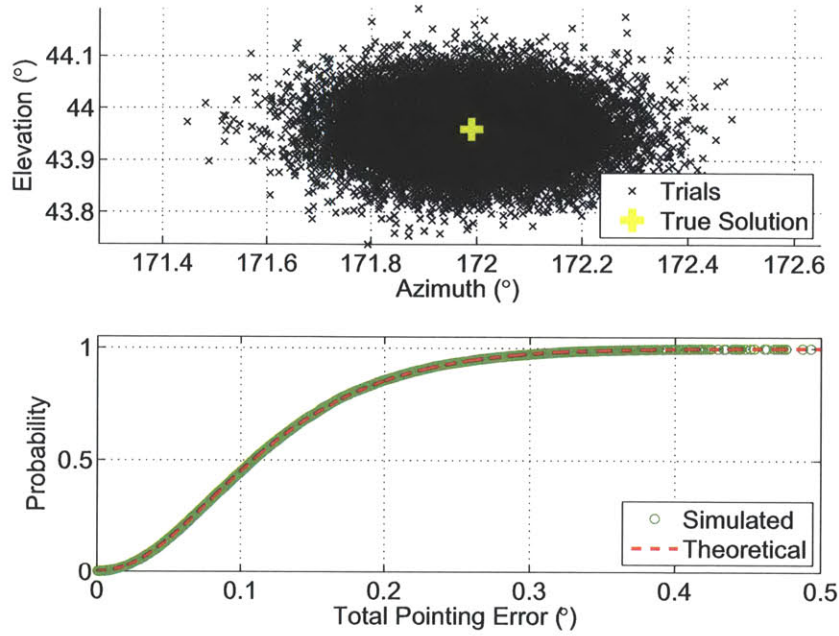


Figure 3-25: Open-loop pointing error simulation that compares the simulated and theoretical distributions.

if the error exceeds a threshold defined by

$$P_{H_0}(D_n > d_{n,\alpha}) = \alpha. \quad (3.69)$$

The threshold ($d_{n,\alpha}$) for an $\alpha = 0.01$ is equal to $1.63/\sqrt{n}$, where n is the number of samples within the distribution [43]. Therefore the threshold for this simulation is approximately 0.013. Figure 3-26 demonstrates the test statistic is below the threshold and the hypothesis is not rejected. Because the hypothesis is not rejected, the theoretical and simulated distributions are considered equal.

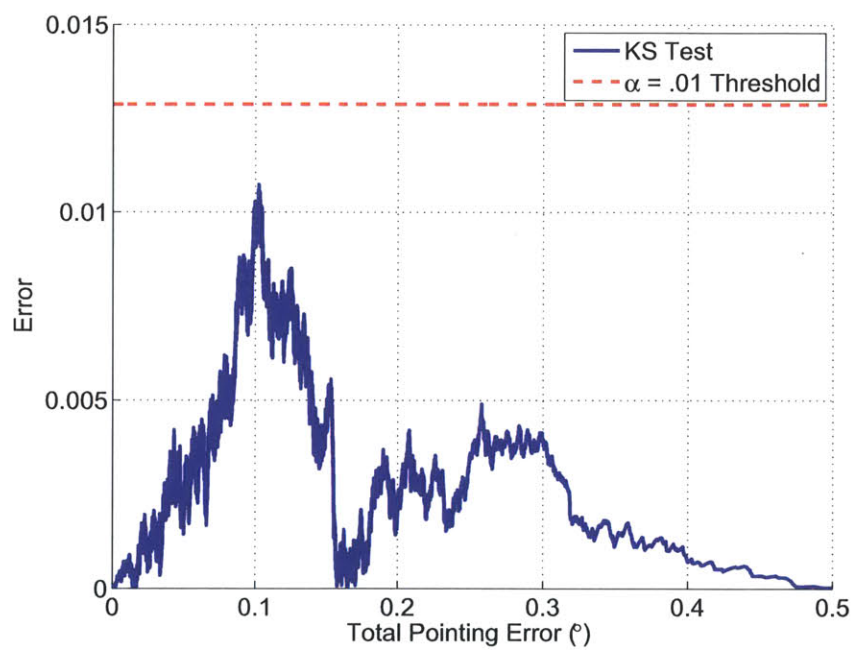


Figure 3-26: Kolmogorov-Smirnov test that compares the error between the two distributions. The error never exceeds the threshold, so the theoretical and simulated distributions are considered equal.

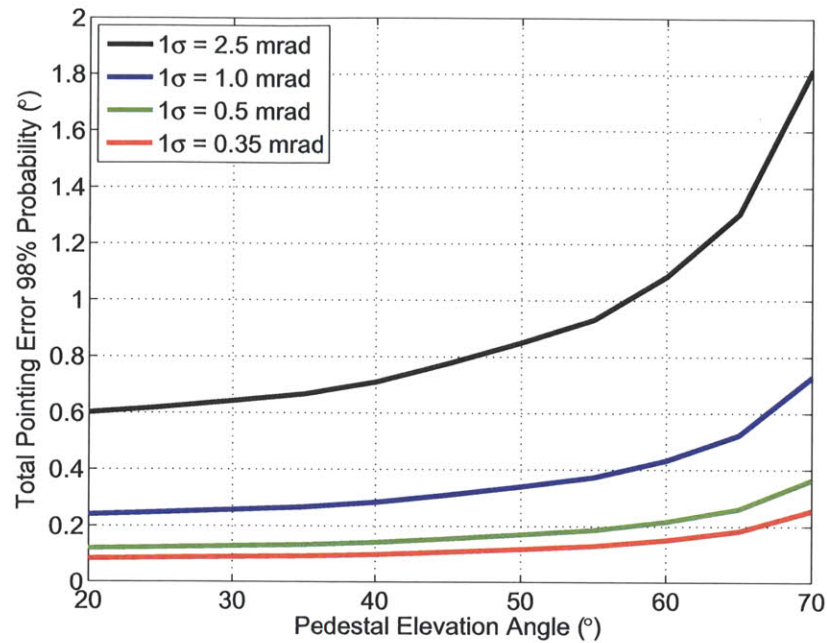


Figure 3-27: INS/GPS pointing error as a function of the pedestal's elevation angle.

Before proceeding to the next section, it is important to note how much of an impact the GPS/INS random errors have on the open-loop pointing error at high elevation angles. Figure 3-27 plots the 98% confidence limit of the total pointing error for increasing elevation angles. The figure demonstrates that the open-loop pointing error increases dramatically at high elevation angles. Section 3.5 combines the analysis from this section with the analysis from Section 3.3 to create one complete simulation of the open-loop pointing error.

3.5 Open-loop Antenna Pointing Simulation

The culmination of this chapter is a simulation that combines the nonlinear pedestal control model from Section 3.3 and the GPS/INS random errors from Section 3.4. The resulting system simulates the pedestal's open-loop pointing performance in the presence of aircraft disturbances and GPS/INS random errors. The model simulates different grade GPS/INS units for different segments of recorded flight data.

3.5.1 Racetrack Flight Data

Figures 3-28, 3-29, and 3-30 present the total pointing error of the racetrack simulations. Figure 3-28 presents the total pointing error in azimuth and elevation coordinates, which demonstrates the difference in variance between the azimuth and elevation distributions. The GPS/INS error causes this inequality and as the figure demonstrates, the difference decreases proportional to the GPS/INS variance. The upper graph of Figure 3-29 presents the total pointing error, which is the root mean square of the azimuth and elevation errors, as a function of time. The lower graph plots the commanded azimuth and elevation angles as a function of time. The two graphs demonstrate that large pointing errors occur at high elevation angles. It is important to note that the magenta data set in the upper plot is identical to the data set in Figure 3-14 at the end of Section 3.3. The other data sets in Figure 3-29 demonstrate the open-loop pointing error caused by random errors in the GPS/INS attitude information. Figure 3-30 presents the CDF of the total pointing error for each GPS/INS. The magenta data set demonstrates the open-loop pointing performance, if a perfect GPS/INS without any random errors existed. The inflection in the line at 0.05° is a result of the pedestal's reaction to dynamic changes in the pointing solution while the aircraft is banked during the racetrack as depicted in Figure 3-14.

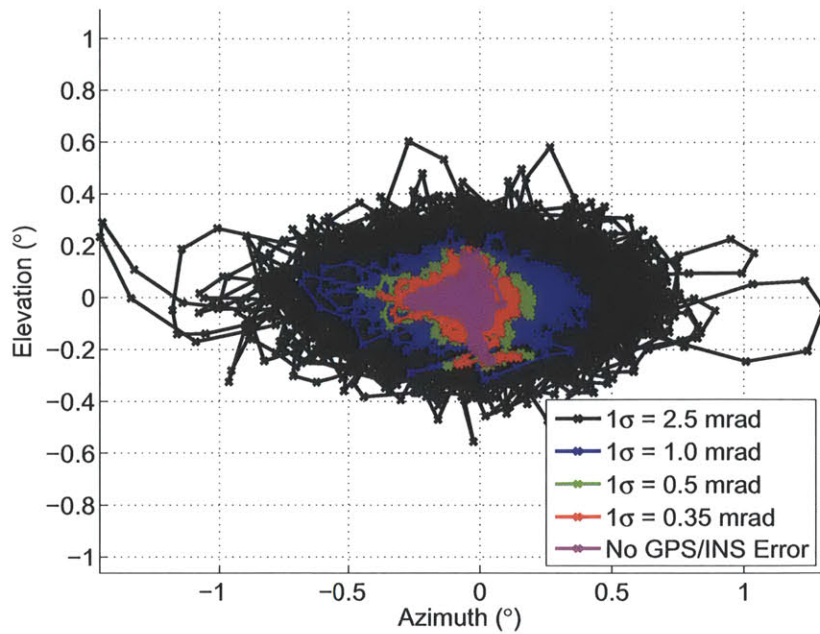


Figure 3-28: Open-loop pointing azimuth and elevation error from simulation of race-track with GPS/INS pointing errors.

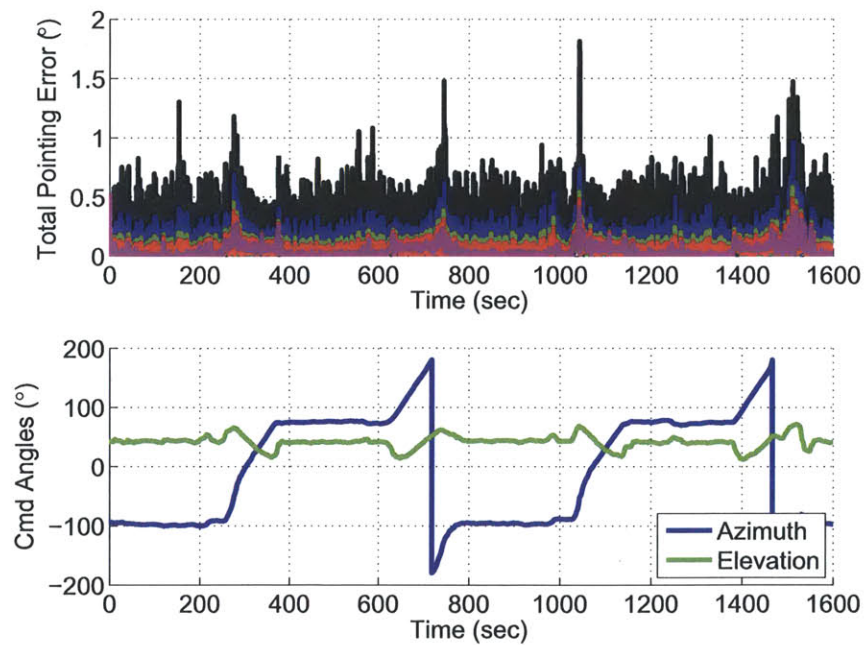


Figure 3-29: Total pointing error from open-loop pointing simulation of the racetrack flight data with GPS/INS pointing errors. The lower graph plots the commanded azimuth and elevation commands. The spikes in error correspond to high elevation angles.

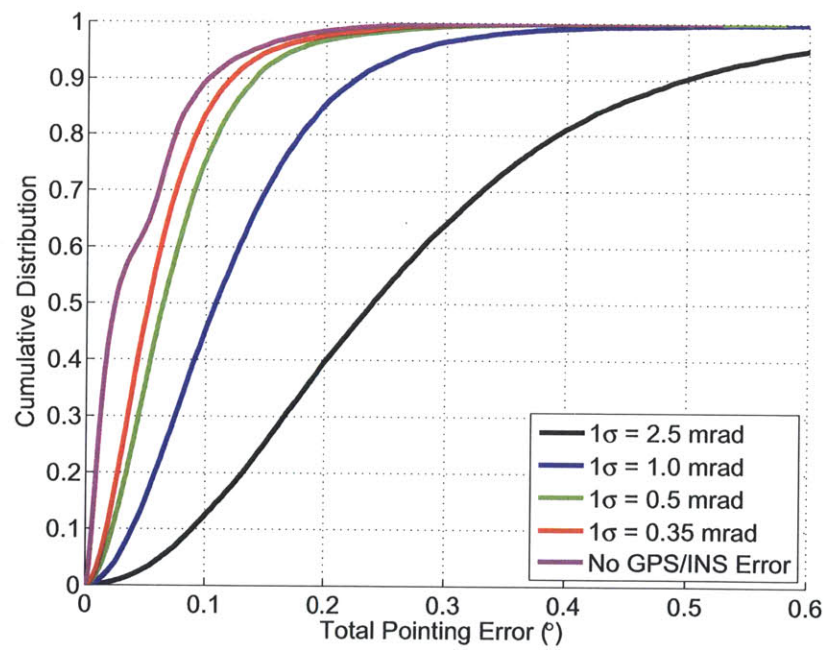


Figure 3-30: Cumulative distribution function of the total pointing error from the open-loop pointing simulation of the racetrack flight data.

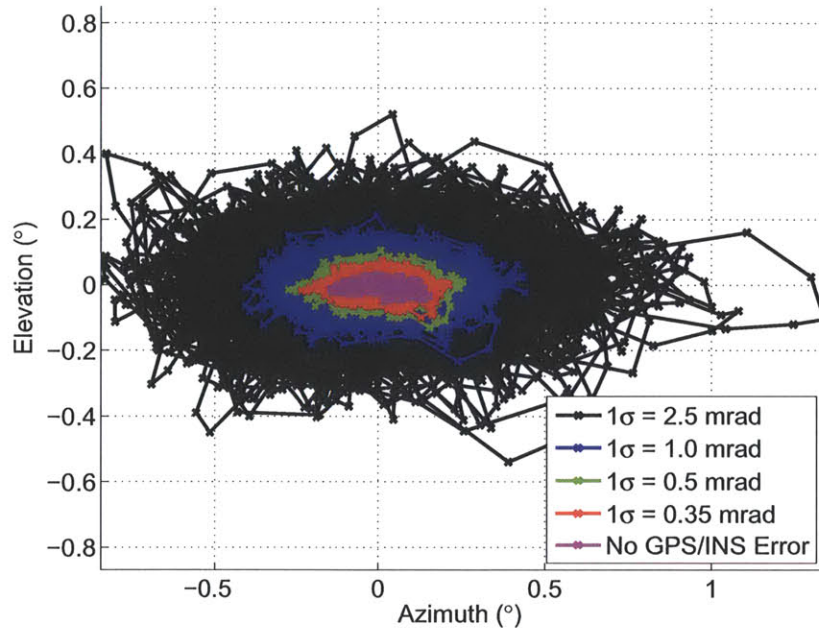


Figure 3-31: Open-loop pointing azimuth and elevation error from simulation of cruise with GPS/INS pointing errors.

3.5.2 Cruise Flight Data

Figures 3-31, 3-32, and 3-33 present the performance results of this simulation. The results demonstrate that the total pointing error is less than racetrack pattern because the aircraft is flying straight and level with very little pitching, rolling, and yawing. This lack of aircraft motion causes a much more constant command input as indicated by the lower graph of Figure 3-32. The GPS/INS random errors still influence the system, but the commanded elevation does not approach the keyhole, so errors in azimuth are not as severe in Figure 3-31 as in Figure 3-28. This restriction in motion also causes the CDF in Figure 3-33 to have lower total pointing error distribution than the distribution in Figure 3-30.

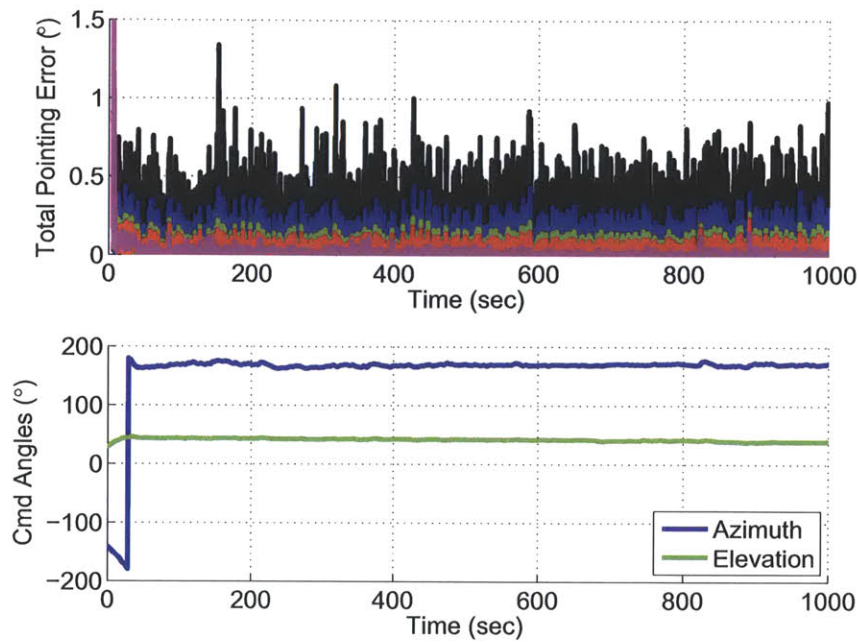


Figure 3-32: Total pointing error from open-loop pointing simulation of the cruise flight data with GPS/INS pointing errors. The lower graph plots the commanded azimuth and elevation commands.

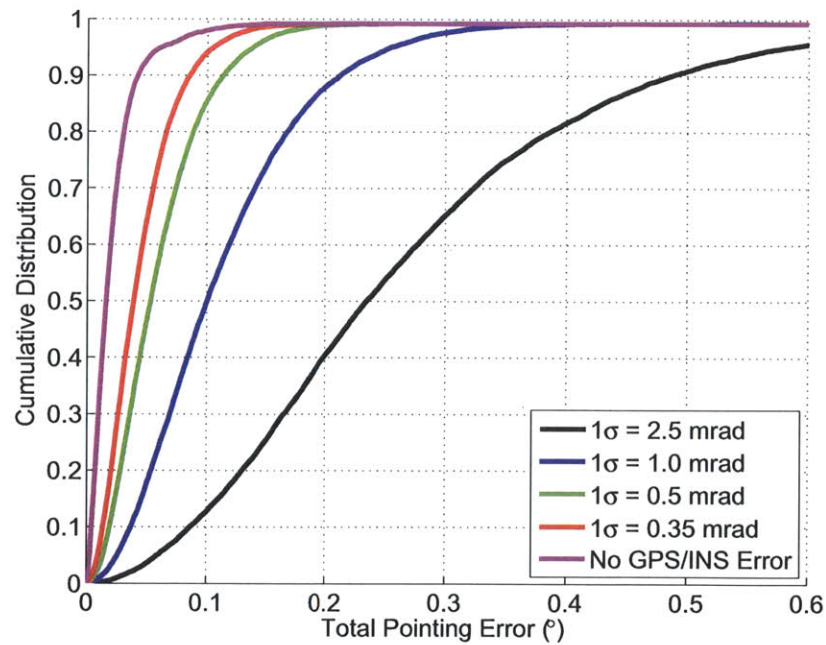


Figure 3-33: Cumulative distribution of the total pointing error from the open-loop pointing simulation of the cruise flight data.

Chapter 4

Closed-Loop Tracking

4.1 Effects on Signal-to-Noise Ratio

The signal processing system is responsible for calculating the signal-to-noise ratio (SNR). The pedestal control computer uses the SNR to estimate antenna pointing errors. The SNR varies over time because it is composed of factors that vary over time. This section examines the variations in SNR to better understand how to implement SNR as a figure of merit for antenna pointing accuracy. The SNR is defined by

$$\text{SNR} = \frac{P_R}{N} = \frac{P_T G_T G_R}{N L_{\text{FSP}} L_0} = \frac{\text{RIP} * G_R}{N}, \quad (4.1)$$

where P_R and P_T are the received and transmitted power respectively, G_R and G_T are the receiver and transmitter antenna gain respectively, N is thermal noise in the receiver, L_{FSP} is the free-space path loss, L_0 is the combination of other losses (atmospheric absorption, rain attenuation, refraction, diffraction, and multipath), and RIP is the received isotropic power. It is convenient to work in terms of decibels, which allow the equation to be rewritten as

$$10 \log_{10} \text{SNR} = 10 \log_{10} \text{RIP} + 10 \log_{10} G_R - 10 \log_{10} N. \quad (4.2)$$

Each component is discussed in more detail to understand its impact to the SNR. Unique characteristics of the MILSATCOM system are also discussed to understand their impact on the SNR components.

4.1.1 Received Isotropic Power

The Received Isotropic Power (RIP) is the power at the receiver antenna, and is defined as

$$\text{RIP} = \frac{P_T G_T}{L_{\text{FSP}} L_0} = \frac{\text{EIRP}}{L_{\text{FSP}} L_0}, \quad (4.3)$$

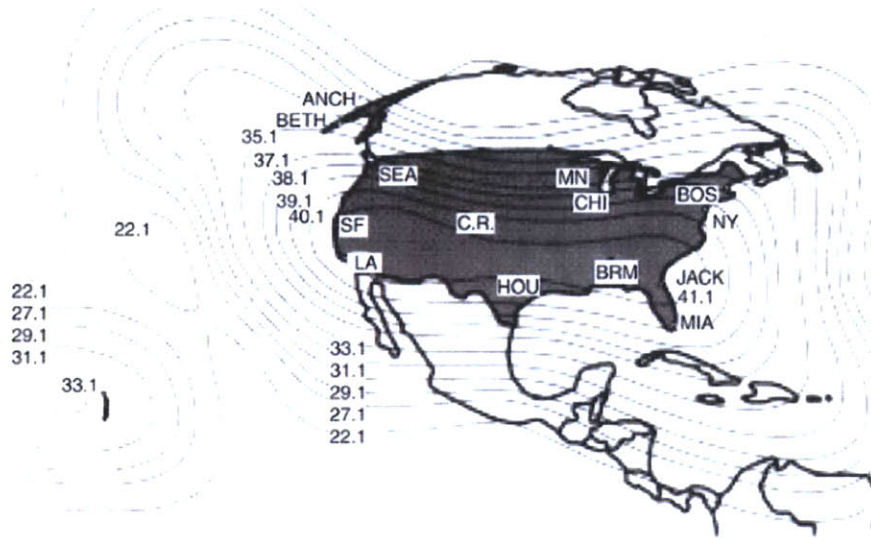


Figure 4-1: Satellite footprint of the effective isotropic radiated power for a beam on a satellite. The power is a function of the receive antenna's geographic location. Reproduced from Reference 44.

where EIRP is the effective isotropic radiated power and the other components are already defined. Each component is discussed in more detail.

Effective Isotropic Radiated Power

The effective isotropic radiated power (EIRP) is the transmitted power times the transmitter gain. Due to the satellite's antenna beam pattern, the EIRP does not radiate uniformly to the Earth's surface, so certain locations have higher EIRPs than others. A satellite footprint map defines the EIRP as a function of geographic location. Figure 4-1 presents a typical satellite footprint that demonstrates the EIRP as a function of the latitude and longitude. Depending on the mission profile and the aircraft's intended flight path, the performance could change throughout the flight.

Friis Free-Space Path Loss

The Friis Free-space path loss equation is defined as

$$L_{\text{FSP}} = \left(\frac{4\pi d}{\lambda} \right)^2, \quad (4.4)$$

where λ and d are the transmission wavelength and distance traveled respectively in meters. Because of the relation

$$c = f\lambda \quad (4.5)$$

where c is the speed of light in meters per second and f is the transmission frequency in hertz, Equation 4.4 can be rewritten as

$$L_{\text{FSP}} = \left(\frac{4\pi df}{c} \right)^2. \quad (4.6)$$

Equation 4.4 is converted to decibels and becomes

$$L_{\text{FSP}}\text{dB} = 20 \log_{10}(f) + 20 \log_{10}(d) - 147.57, \quad (4.7)$$

As seen in Equation 4.7, the loss increases as either the distance or transmission frequency increase. A GEO satellite transmitting at EHF results in a loss of approximately 210 dB [8].

Atmospheric Absorption and Rain Attenuation

Atmospheric absorption and rain attenuation are two other factors common to all communication systems. Oxygen and water molecules in the atmosphere absorb RF energy and convert it into heat as the signals pass through the atmosphere. The level of attenuation varies with the amount of atmosphere the signal travels through, which depends on the terminal's geographic location with respect to the satellite. Atmospheric absorption becomes severe at elevation angles below 10° relative to the Earth's surface [44]. The attenuation increases at higher frequencies, but the most significant attenuation occurs at the resonant frequencies of water vapor and oxygen, 22 and 66 GHz respectively as Figure 4-2 demonstrates.

Rain attenuation is directly related to atmospheric absorption and the two are often combined as one loss term. Rain droplets absorb and scatter RF signals, which cause drops in the received signal. The loss depends on the transmission frequency and angle of incidence, much like atmospheric attenuation, except that the amount of loss depends on the rate of precipitation, as demonstrated in Figure 4-3.

Since the terminal is on an airborne platform flying at cruising altitude above the weather, rain attenuation is not a serious issue, but atmospheric attenuation still causes a minor loss approximately equal to 0.2 dB [44].

Refraction, Diffraction, and Multipath

Refraction, diffraction, and multipath are three impairments that do not impact the received SNR for this application, but are explained to understand why they are insignificant.

Refraction is a distortion inherent to the atmosphere that bends RF signals much like a prism bends light into a color spectrum. The upper portion of the atmosphere defined as the troposphere has decreasing levels of density, which cause RF signals to bend further at lower altitudes. Much like atmospheric absorption, refraction can

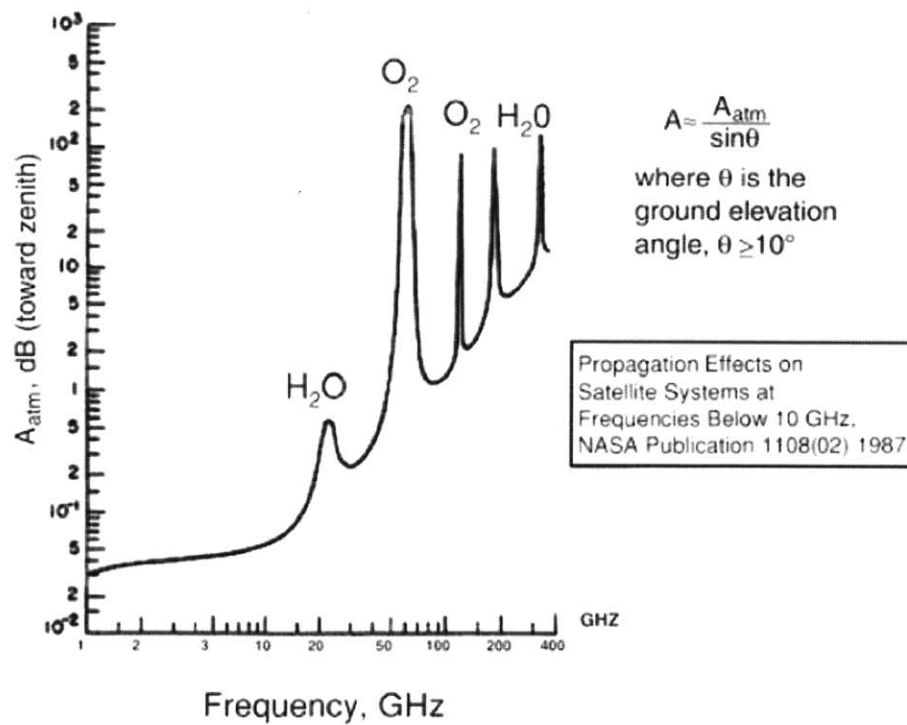


Figure 4-2: Signal attenuation caused by atmospheric absorption. Spikes in attenuation occur at the resonant frequencies of water and oxygen, 22 and 66 GHz respectively. Reproduced from Reference 44.

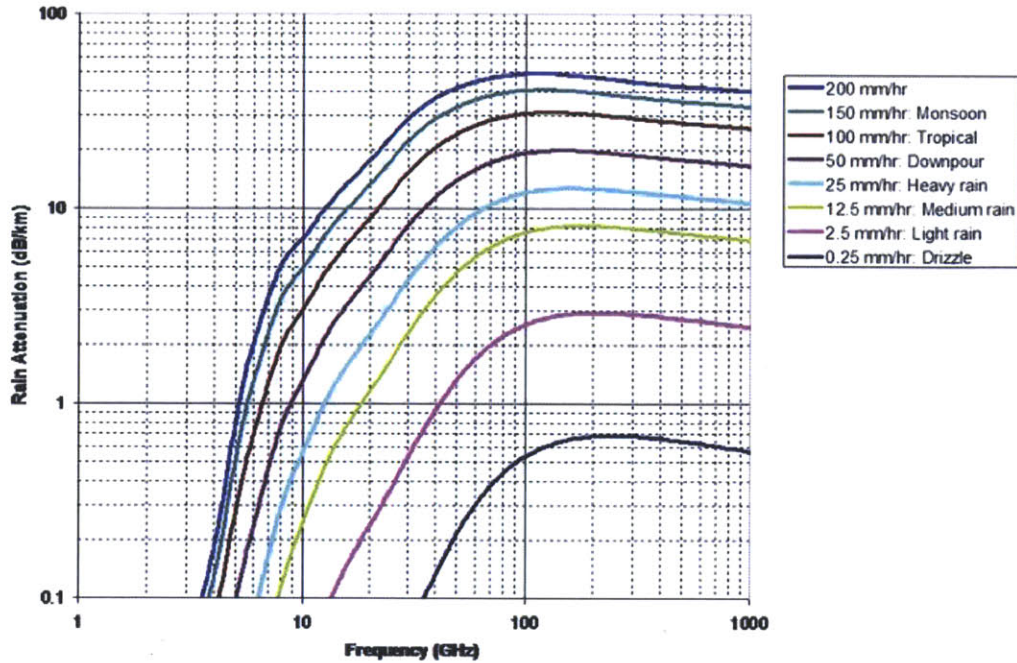


Figure 4-3: Signal Attenuation caused by the rain droplets. The attenuation increases as a function of transmission frequency. Reproduced from Reference 45.

cause fading in the received signal at extremely low elevation angles. For the most part, refraction is not a serious issue because the amount of bending is extremely small relative to beamwidth of the terminal and satellite antennas [44].

Similar to refraction, diffraction is the bending of RF signals around a physical object such as a building or a hill. This bending occurs when the obstruction is the same size or smaller than the wavelength. At 20 and 44 GHz, the wavelengths are too short for the signal to be affected significantly by diffraction [8].

Multipath occurs when the RF signal passing through the ionosphere reflects off of a discontinuity in such a way that the reflection reaches the intended target via an indirect path. The terminal receives both the direct and indirect signals, which are out of phase and cause either constructive or destructive interference. This unstable impairment is very infrequent because the discontinuities in the ionosphere are rare phenomena that occur semiannually during the equinoxes [44]. Multipath can still occur from the RF signal reflecting off of other objects or surfaces, but at cruising altitude, multipath is not a major factor.

4.1.2 Thermal Noise

Thermal noise occurs in the receiver as in any electronic device with a current running through it. The noise power density in a receiver is defined as

$$N_0 = kT, \quad (4.8)$$

where N_0 is the noise power density in watts per hertz of bandwidth, k is Boltzmann's constant in joules per kelvin, and T is the system temperature in kelvin. Thermal noise exists in the terminal and cannot be eliminated from the system. It gives the received signal a random component which limits the performance of the system. The average noise power is the thermal noise density times the bandwidth of the receiver as defined by

$$N = kTB, \quad (4.9)$$

where N is the thermal noise power in watts, B is the receiver bandwidth in hertz, and k and T are the same as in Equation 4.8.

Other forms of noise exist, such as intermodulation and impulse noise. Intermodulation noise between transmitted signals is caused by spacing antennas too close together [44]. Impulse noise is caused by electromagnetic disturbances or flaws in the system [8]. These types of noise do not significantly impact MILSATCOM terminals.

4.1.3 Receiver Antenna Gain

The final component that affects the SNR is the receiver antenna gain defined in Equation 4.1. This gain depends on the receiver antenna's beam pattern at the transmission frequency and the pointing accuracy of the antenna pedestal. As shown in Figures 2-3 and 2-4, the HPBW is a way of comparing different antennas and defining the required pointing accuracy the pedestal must achieve. Pointing errors translate into losses defined by the receiver's antenna beam pattern. This loss is the error in the pointing solution that closed-loop tracking is trying to eliminate in order to improve the communication link performance.

4.1.4 Military Satellite Communications Systems Characteristics

The MILSATCOM Program is a constellation of GEO satellites that provides "Nuclear Survivable" communication to military assets around the globe. Each satellite achieves secure communication links through complex security measures. The transmission bandwidth for uplink and downlink are wide enough that combined with fast frequency hopping it is difficult for an unauthorized user to detect and intercept [12]. The transmitter and antenna have frequency responses, which means the EIRP changes as a function of time. Section 4.2 addresses this change and determines the impact on the sampled SNR.

4.2 Signal-to-Noise Ratio Characterization

Section 4.1 defined the major components of the SNR and the factors that affect these components. This section will concentrate on the components of the SNR that change between samples.

The signal processing system calculates the signal and noise levels during a single hop. These values are fed to the antenna pedestal computer, which assesses the pointing performance. By modeling these properties, this section identifies proper estimators for the SNR.

The RIP, receiver antenna gain, and the receiver's thermal noise are the three independent, random components of the SNR as defined by

$$\text{SNR} = \frac{\text{RIP} * G_R}{N}. \quad (4.10)$$

Of the three components, the receiver gain depends on the pedestal's open-loop pointing performance from Chapter 3. The analysis for this section and Section 4.3 assume the pedestal points perfectly and the receiver antenna gain is constant. Section 4.4 incorporates the open-loop pointing performance and the changing receiver antenna gain. The two remaining components to model are the RIP and the receiver's thermal noise. The analysis in this section analyzes both to determine how best to estimate each component and if one component influences the final SNR more than the other.

4.2.1 Thermal Noise Power Characterization

A zero mean Gaussian distribution with variance (σ^2) models the receiver's thermal noise. This distribution is explicitly defined by

$$f(X_i|\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{X_i}{\sigma}\right)^2}, \quad -\infty \leq X_i \leq \infty \quad \text{for } i = 1, \dots, n. \quad (4.11)$$

The variance of the random variable is the average noise power and is the unknown parameter the pedestal control computer must estimate from the received noise samples. The upper and lower graphs of Figure 4-4 are the probability distribution function (PDF) and cumulative distribution function (CDF) of Equation 4.11 respectively.

In point estimation, such properties as unbiasedness and consistency factor into identifying a suitable estimator for an unknown statistical parameter. The Maximum Likelihood Estimator (MLE) is a statistical estimator with several desirable properties. The MLE is consistent, which means the estimate converges to the true value, and it is asymptotically normal, which means as the number of samples increases, the distribution of the MLE tends toward a Gaussian distribution [43]. The first step in finding the MLE is creating the likelihood function for a set of random variables from a distribution with unknown parameters. The estimator that maximizes the likelihood function is the MLE for that parameter.

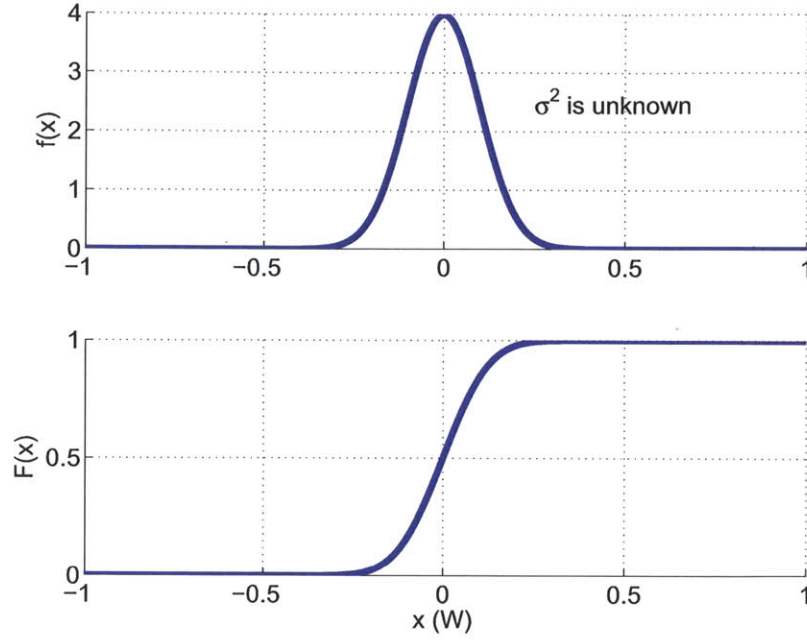


Figure 4-4: Probability distribution function and cumulative distribution function of the zero mean Gaussian distribution with unknown variance.

The likelihood function of the average noise power is defined as

$$L(\sigma^2) = \prod_{i=1}^n \{f(X_i|\sigma^2)\}. \quad (4.12)$$

The estimate that maximizes Equation 4.12 is

$$(\hat{\sigma}^2) = \frac{1}{n} \sum_{i=1}^n (X_i)^2, \quad (4.13)$$

which is MLE of the average noise power [43].

The next step is to define the distribution of the MLE as a function of the number of samples and the mean noise power. The distribution of the MLE is a modified chi-square distribution. A chi-square distribution is the sum of the squares of a normal distribution with zero mean and unity variance. The PDF and CDF for a general chi-square distribution with n degrees of freedom are defined by

$$f(x|n) = \begin{cases} \frac{1}{2^{n/2}\Gamma(n/2)} x^{n/2-1} e^{-x/2}, & \text{if } 0 \leq x < \infty \\ 0, & \text{otherwise} \end{cases} \quad (4.14)$$

and

$$F(x|n) = \begin{cases} \frac{1}{\Gamma(n/2)} \gamma(n/2, x/2), & \text{if } 0 \leq x < \infty \\ 0, & \text{otherwise} \end{cases}, \quad (4.15)$$

respectively, where n is the number of squared and summed normal random variables [43].

Two modifications are made to the distribution. To get a distribution of the MLE, the chi-square distribution is divided by the number of samples (n), and a simple change of variables accommodates the non-unity variance of the random variables. The PDF and CDF for the MLE of the average noise power are

$$f_N(y|n, \sigma^2) = \begin{cases} \frac{1}{2^{n/2} \Gamma(n/2)} \left(\frac{n}{\sigma^2} y\right)^{n/2-1} e^{-\frac{n}{2\sigma^2} y}, & \text{if } 0 \leq y < \infty \\ 0, & \text{otherwise} \end{cases} \quad (4.16)$$

and

$$F_N(y|n, \sigma^2) = \begin{cases} \frac{1}{\Gamma(n/2)} \gamma(n/2, \frac{n}{2\sigma^2} y), & \text{if } 0 \leq y < \infty \\ 0, & \text{otherwise} \end{cases}. \quad (4.17)$$

The MLE distribution defined in Equation 4.16 is converted from a linear to logarithmic scale as the final step. This modification is beyond a simple change of variables and cannot be solved analytically. Instead, a numerical solution is found by taking the CDF of Equation 4.17 and translating the X -axis into a logarithmic scale. The PDF of the new distribution is the derivative of the new CDF. Figures 4-5 and 4-6 are the PDF and CDF of the average noise power estimate as a function of the number of samples.

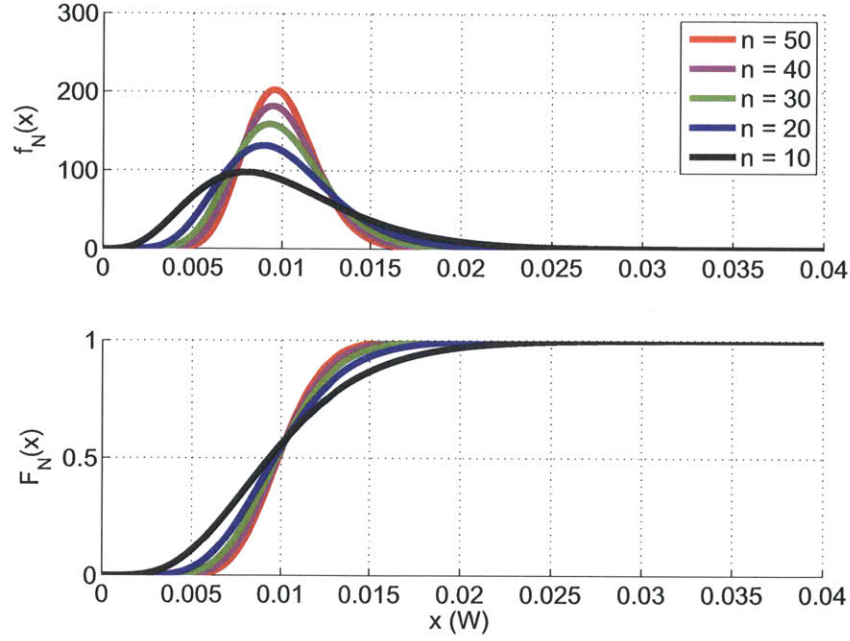


Figure 4-5: Probability and cumulative distribution functions of the estimated average noise power as a function of n samples (Linear Scale).

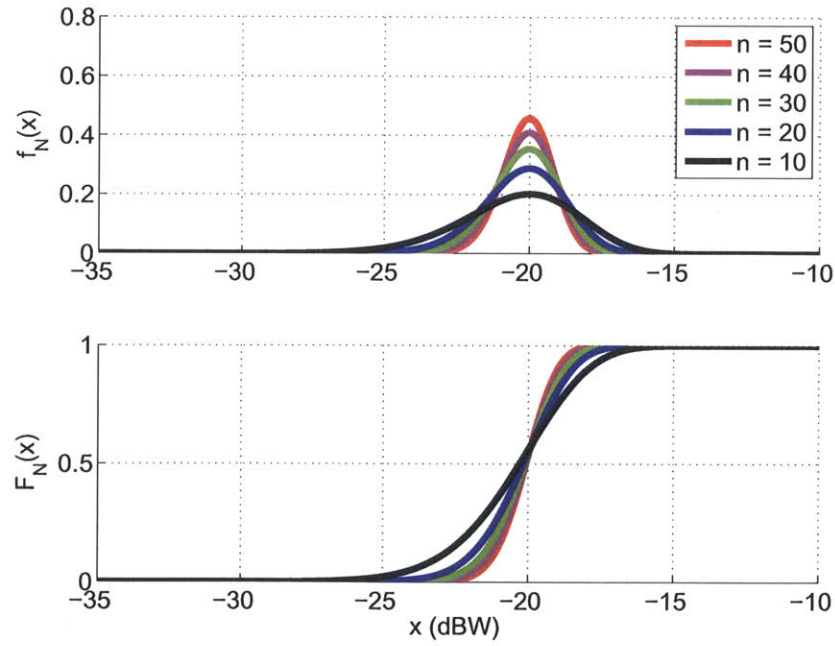


Figure 4-6: Probability and cumulative distribution functions of the estimated average noise power as a function of n samples (Logarithmic Scale).

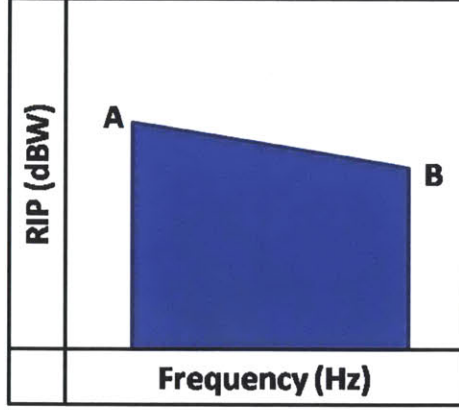


Figure 4-7: Frequency response of the RIP. The slope and the minimum and maximum RIP are unknown.

4.2.2 Received Isotropic Power Characterization

The RIP is a function of the EIRP, free-space path loss, and environmental losses as defined by

$$\text{RIP} = \frac{\text{EIRP}}{L_{\text{FSP}} L_0}. \quad (4.18)$$

Of the components, only the EIRP is considered to be a random variable, because EIRP is a function of frequency and fast frequency hopping places the signal at different locations within the band over time. For this case, the RIP (in dBW) is modeled as a linear function of the transmission frequency, Figure 4-7. The transmitted signal randomly hops over the frequency spectrum and has equal probability of hopping to any frequency. Although the range of possible frequencies is known, the range of possible RIP is unknown. The goal then is to estimate the average RIP.

Because there is equal probability of being at any frequency within the band, the RIP is modeled by a uniform distribution with an unknown range. The unknown range corresponds to the slope of the RIP. For a set of n samples, the transmitted signal level for each sample is defined as

$$f(X_i|r) = \begin{cases} \frac{1}{r}, & \text{if } c \leq X_i \leq r + c \\ 0, & \text{otherwise} \end{cases} \quad \text{for } i = 1, \dots, n. \quad (4.19)$$

The upper and lower graphs of Figure 4-8 are the PDF and CDF, respectively, of the uniform distribution with unknown range.

The goal is to estimate the unknown parameter r to determine the average RIP. It is important to note that Equation 4.19 is explicitly defined for the range $[c, r+c]$. The midrange $r/2 + c$ and range r are the two unknown parameters for this distribution. The location parameter c is merely a bias that impacts the final value of the midrange, but does not effect the estimation theory for either the midrange or range. The parameter c is later set to 0 during simulations.

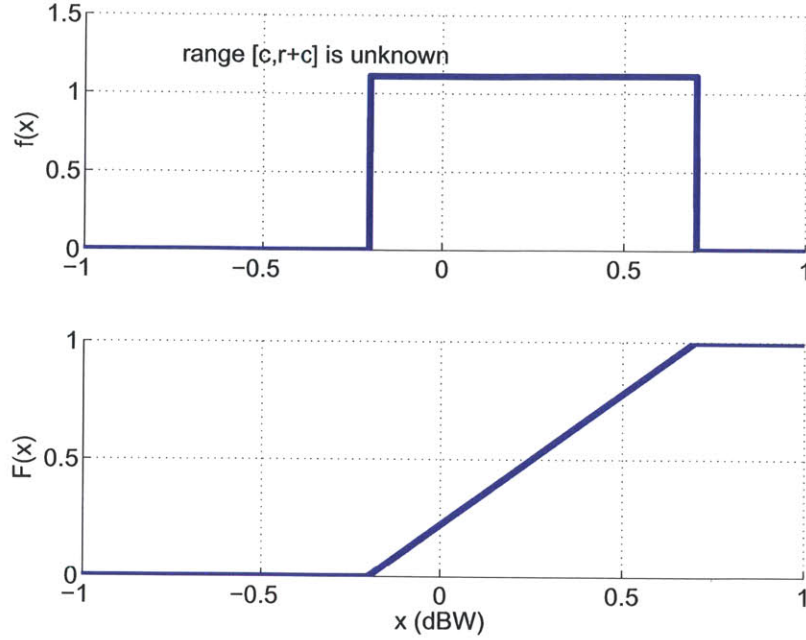


Figure 4-8: RIP distribution is modeled as uniform distribution with unknown range $[c, r + c]$.

Similar to finding the MLE for the noise power, the likelihood function is defined as

$$L(r/2 + c, r) = \begin{cases} \left(\frac{1}{r}\right)^n, & \text{if } c \leq X_1, X_2, \dots, X_n \leq r + c \\ 0, & \text{otherwise} \end{cases} \quad (4.20)$$

The estimate for the midrange and range that maximize Equation 4.20 are

$$\left(\frac{r}{2} + c\right) = \frac{\max(X_1, \dots, X_n) + \min(X_1, \dots, X_n)}{2} \quad (4.21)$$

and

$$\hat{r} = \max(X_1, \dots, X_n) - \min(X_1, \dots, X_n). \quad (4.22)$$

The mean of the maximum and minimum observed samples is the midpoint estimate, while the maximum minus the minimum is the range estimate [43]. Although both are defined, only the midpoint MLE is required to estimate the RIP.

The next step is to define the distribution of the MLE. Equation 4.21 states that the MLE of the midpoint is the average of the maximum and minimum observed samples. Order statistics is concerned with distributions of random variables that have been sorted from least to greatest. Samples X_1, X_2, \dots, X_n from a distribution $f(x)$ are sorted least to greatest such that $X_{(1)} \leq X_{(2)} \leq \dots \leq X_{(n)}$ [43]. Distributions

on each individual order statistic are obtained with

$$f_{X_{(i)}}(x) = \frac{n!}{(i-1)!(n-i)!} [F(x)]^{i-1} f(x) [1 - F(x)]^{n-1}, \quad (4.23)$$

where $f(x)$ and $F(x)$ are the PDF and CDF of the original random variable. Since this problem is concerned with the minimum and maximum observed samples, distributions for the minimum and maximum are defined by

$$f_{X_{(1)}}(x) = n[1 - F(x)]^{n-1} f(x) \quad (4.24)$$

and

$$f_{X_{(n)}}(x) = n[F(x)]^{n-1} f(x) \quad (4.25)$$

respectively. The sum of these two distributions form a joint distribution defined as

$$\begin{aligned} f_{1n}(x, y) &= \text{Prob} [X_{(1)} \leq x \text{ and } X_{(n)} \leq y] \\ &= \begin{cases} n(n-1)f(x)f(y)[F(y) - F(x)]^{n-2} & \text{if } x \leq y \\ 0 & \text{if } x > y \end{cases} \end{aligned} \quad (4.26)$$

To get the distribution on the MLE midrange, Equation 4.26 is divided by 2, which then becomes

$$\begin{aligned} f_M(x) &= 2 \int_{-\infty}^x f_{1n}(t, 2x - t) dt \\ &= 2n(n-1) \int_{-\infty}^x f(t)f(2x - t)[F(2x - t) - F(t)]^{n-2} dt \end{aligned} \quad (4.27)$$

The distributions for the MLE are found by substituting the uniform distribution defined in Equation 4.19 into Equation 4.27. The final distributions are defined as

$$\begin{aligned} f_S(x|n, r) &= \begin{cases} \frac{n}{r} 2^{n-1} \left(\frac{x}{r}\right)^{n-1} & \text{if } c < x \leq c + \frac{r}{2} \\ \frac{n}{r} 2^{n-1} \left(1 - \frac{x}{r}\right)^{n-1} & \text{if } c + \frac{r}{2} \leq x < c + r \end{cases} \\ F_S(x|n, r) &= \begin{cases} 2^{n-1} \left(\frac{x}{r}\right)^n & \text{if } c < x \leq c + \frac{r}{2} \\ 1 - 2^{n-1} \left(1 - \frac{x}{r}\right)^n & \text{if } c + \frac{r}{2} \leq x < c + r \end{cases} \end{aligned} \quad (4.28)$$

These equations define the distribution on the MLE to estimate the signal power [46]. This distribution is only valid when the receiver antenna gain is constant. Figure 4-9 is the PDF and CDF of the estimate as a function of the number of samples for a range of 10 dBW. As the number of samples increases, the variance of the distribution decreases.

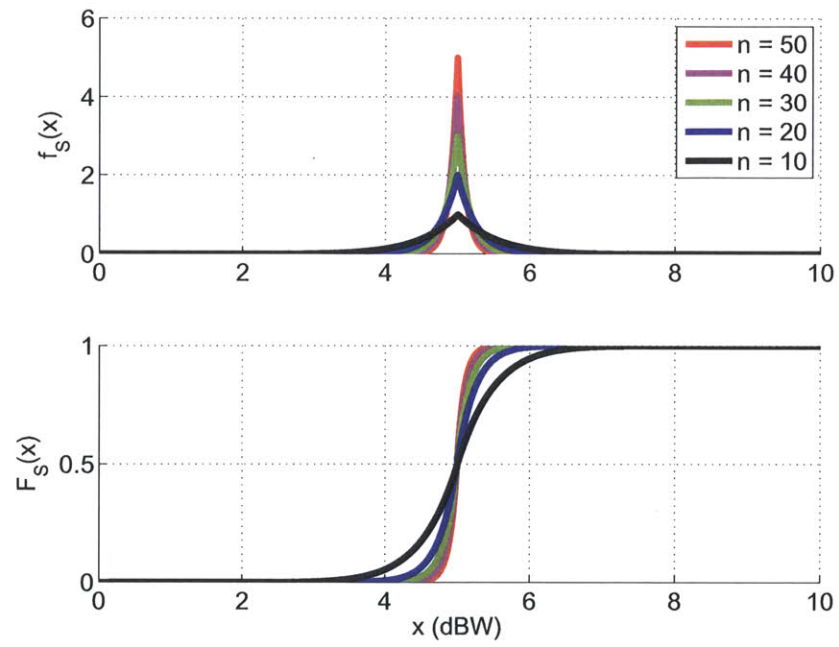


Figure 4-9: Probability and cumulative distribution functions of the estimated RIP as a function of n samples for a range of 10 dBW.

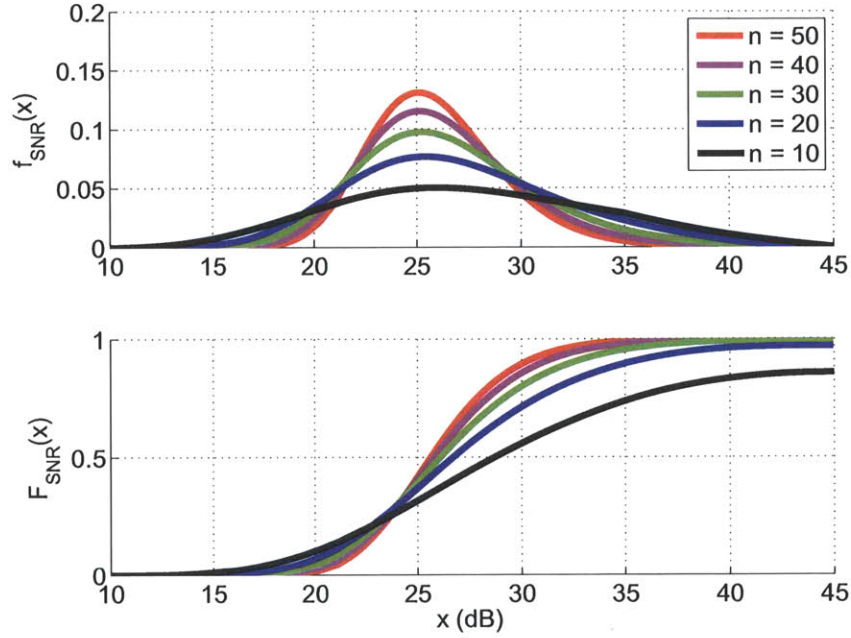


Figure 4-10: Probability and cumulative distribution functions of the estimated SNR as a function of n samples.

4.2.3 SNR Characterization

The estimated SNR is calculated with the MLE for both the thermal noise power and the RIP. Since both estimates are in dBW, the estimate for the SNR is the signal level minus the noise power as defined by

$$\text{SNR}_{\text{dB}} = S_{\text{dBW}} - N_{\text{dBW}}. \quad (4.29)$$

The distribution of the sum of two independent random variables is the convolution of the individual PDFs [43]. The estimated SNR distribution is the convolution of the RIP distribution with the negative of the noise power distribution. Figure 4-10 is the PDF and CDF of the SNR distribution as a function of the number of samples.

Figure 4-11 is the output from a MATLAB Simulink simulation that compares the theoretical and simulated distributions. A KS test, similar to the one presented in Section 3.4, tests the null hypothesis that the simulated and theoretical distributions are equal. Figure 4-12 presents the results from the test, which demonstrates that the test statistic does not reject the hypothesis with a confidence level of $\alpha = 0.01$.

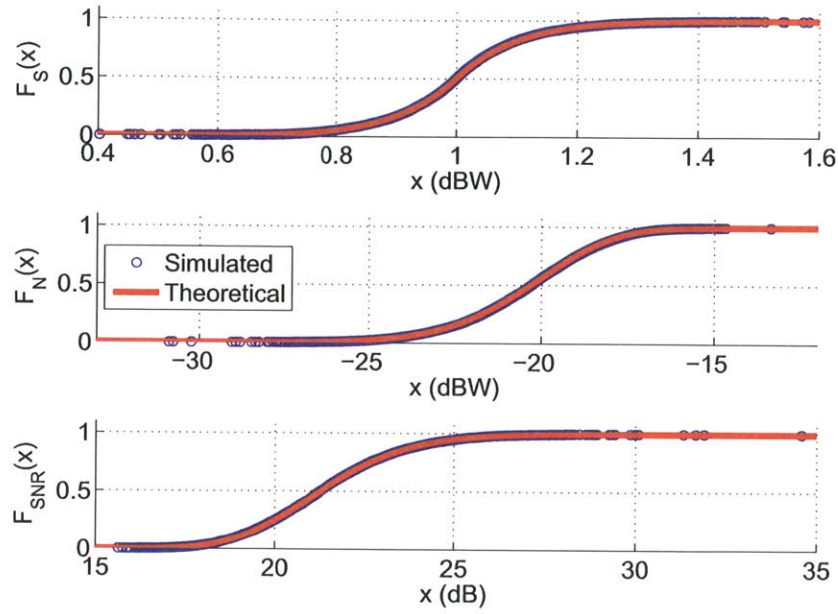


Figure 4-11: Comparison between the simulated and theoretical MLE distributions for 10 samples.

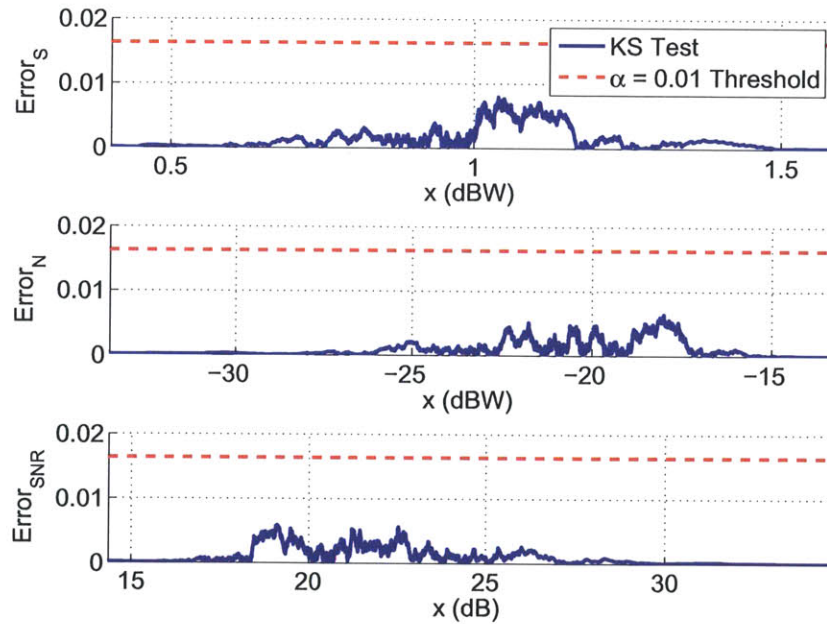


Figure 4-12: Error between simulated and theoretical MLE distributions for 10 samples.

4.2.4 SNR Analysis

The estimated SNR distribution is fully modeled, but must be analyzed to understand how the RIP, thermal noise, and the number of samples impact the estimated SNR distribution. The number of samples directly impacts the probability of being within a certain range of the average value. This analysis shows how the average noise power and slope of the RIP impact the SNR distribution. Both the noise power or RIP have individual distributions, but it is unclear if one has a stronger influence on the SNR distribution than the other.

The average noise power is the variance of the Gaussian random variable that models the receiver thermal noise. The magnitude of the average noise power affects the mean and variance of the MLE distribution. Estimating the average noise power to within $\pm X$ dB of the true value is the main concern and is equivalent to saying $10 \log_{10}(\sigma^2) \pm X$. After a little algebra, the relation becomes $\sigma^2 10^{\pm X/10}$, which means the two critical points around the true value are $\sigma^2(1 \pm 10^{\pm X/10})$. Using the CDF from Equation 4.17, the probability of being within $\pm X$ dB of the average noise power is

$$\text{Prob} \left[\left| \sigma^2 - \hat{\sigma}^2 \right| \leq X_{\text{dB}} \right] = F_N \left(\sigma^2(1 + 10^{X/10}) \mid n, \sigma^2 \right) - F_N \left(\sigma^2(1 - 10^{-X/10}) \mid n, \sigma^2 \right). \quad (4.30)$$

The important observation is that the output of Equation 4.30 is independent of the average noise power. When the critical points are substituted into Equation 4.17 as shown by

$$F_N(\sigma^2(1 + 10^{X/10}) \mid n, \sigma^2) = \frac{1}{\Gamma(n/2)} \gamma(n/2, \frac{n}{2}(1 + 10^{X/10})), \quad (4.31)$$

the σ^2 term cancels out, so that the probability depends only the number of samples n and independent of the variance. The average noise power impacts the average SNR, but the distribution about the mean SNR is independent of the average noise power.

The slope of the RIP affects the probability of estimating the midrange. As the slope increases, the probability of being within ± 1 dB of the signal level decreases drastically. The probability of being within $\pm X$ dB is defined explicitly as

$$\begin{aligned} \text{Prob} \left[\left| \frac{r}{2} - \frac{\hat{r}}{2} \right| \leq X_{\text{dB}} \right] &= F_S \left(\frac{r}{2} + X \mid n, r \right) - F_S \left(\frac{r}{2} - X \mid n, r \right) \\ &= 2 \left(.5 - 2^{(n-1)} \left(\frac{r-2X}{2r} \right)^n \right) \end{aligned} \quad (4.32)$$

It is important to note that the bias imposed on the distribution by c drops out of the equation. Figure 4-13 demonstrates that the probability of being within ± 1 dB of the signal level increases for a larger number of hops (n), but decreases for larger slopes (r). In practice, the slope should not exceed 3 dB, which means the probability of being within ± 1 dB is extremely high. Therefore the probability of being within $\pm X$ dB of the average SNR does not depend heavily on the RIP distribution as long as the slope is relatively small and the pedestal control computer estimates over at

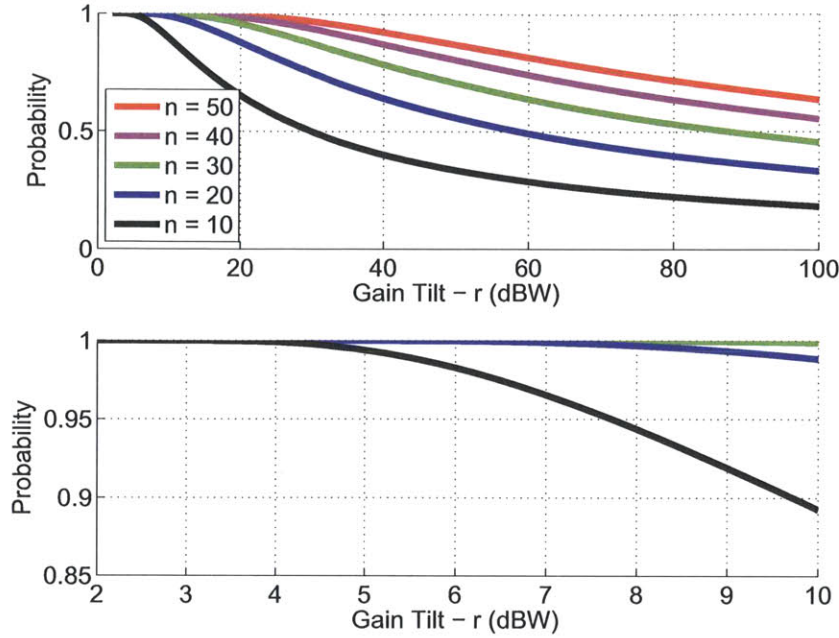


Figure 4-13: Probability of estimating the average RIP within ± 1 dB.

least 10 hops.

Figure 4-13 demonstrates that the distribution of the RIP does not impact the SNR distribution as much as the average noise power distribution. Figure 4-14 compares all three distributions as a function of the number of hops. The upper graph has a much narrower band in the X -axis, which translates into less of an influence on the final distribution of the lower graph.

As a final comparison, the upper two graphs of Figure 4-15 are the probability of being within $\pm X$ dB of the average SNR and the average noise power as a function of the number of hops. The lower graph of Figure 4-15 is the difference between the two probabilities demonstrates that the variance of the RIP is insignificant when compared to the variance in the average noise power.

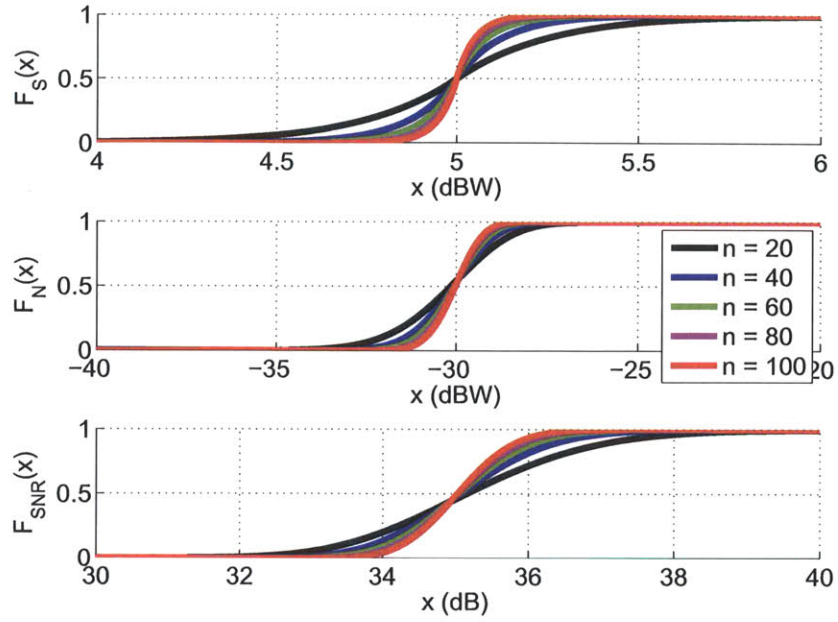


Figure 4-14: Comparison between the three distributions as a function of the number of hops.

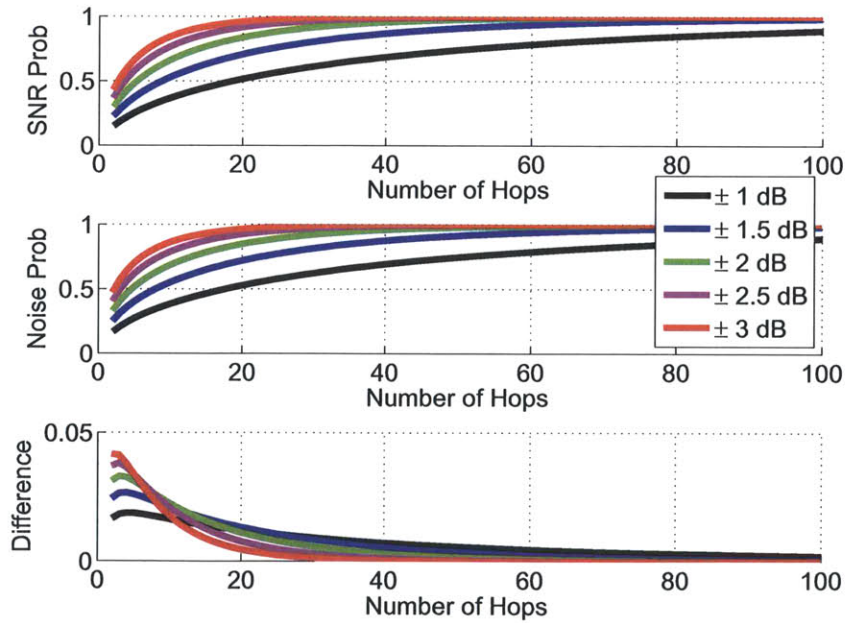


Figure 4-15: Upper two graphs are the probability of being within $\pm X$ dB of the average SNR and the average noise power as a function of the number of hops. The lower graph is the difference between the two probabilities.

4.3 Step-Tracking Algorithm

As defined in Section 2.4, the MILSATCOM terminal has no beacon to track, so step tracking is the most practical closed-loop pointing option. This section explains the theory behind the step-tracking algorithm and then applies the theory with the modeled SNR from Section 4.2 to determine the closed-loop tracking performance for an ideal pedestal.

4.3.1 Step-Tracking Theory

Step tracking is a form of closed-loop tracking used to assess and reduce the pointing error between the terminal's antenna and the satellite. The pedestal control computer commands the pedestal to point the antenna deliberately off-boresight by a predefined angle. The modem processes the received signal, and the computer samples over a certain number of hops to estimate the SNR. The system repeats this process a set number of times. The computer then assesses the pointing error between the pedestal and satellite. If an error exists, then the pointing solution is updated.

Focusing just on the one-dimensional step-tracking problem, a dither in one direction is comparable to shifting the entire beam pattern along the x-axis by the dither angle. The beam pattern is dithered in the opposite direction by same the dither angle. The two new beam patterns are then subtracted from another to create a difference pattern that defines the the difference in gain as a function of the AOB [32]. Figure 4-16 presents two dithered antenna beam patterns and their difference pattern. If a pointing error exists, then the difference pattern is used to correct the pointing error.

This step-tracking method works for the range of pointing errors between the two spikes of the difference pattern. These spikes occur at the nulls of the dithered antenna beam patterns. Using this method for ranges that extend beyond the difference beam pattern's two spikes introduces ambiguity in the AOB because one gain value can map to multiple AOBs.

The difference beam pattern of Figure 4-16 depends on two factors: the antenna beam pattern and the dither angle. The antenna beam pattern is fixed for a particular aperture size and transmission frequency. The dither angle is a configurable parameter in the step-tracking software. The task becomes selecting the proper angle that optimizes the dither scheme. As the dither angle increases, the spikes in the lower graph of Figure 4-16 move closer together, which causes the slope of the linear portion between the spikes to increase. This increase in slope results in small AOBs having larger gain variance, which translates into a better resolution, but also decreases the usable AOB range. Smaller dither angles pull the spikes further apart, which implies a larger AOB range, but also decreases the AOB resolution. Figure 4-17 demonstrates the slope increasing as the dither angle increases. The optimum dither angle depends on the specific application. For communication systems, the optimum dither angle is 0.47δ , where δ is the HPBW of the receive antenna beam pattern [32, 47]. The step-

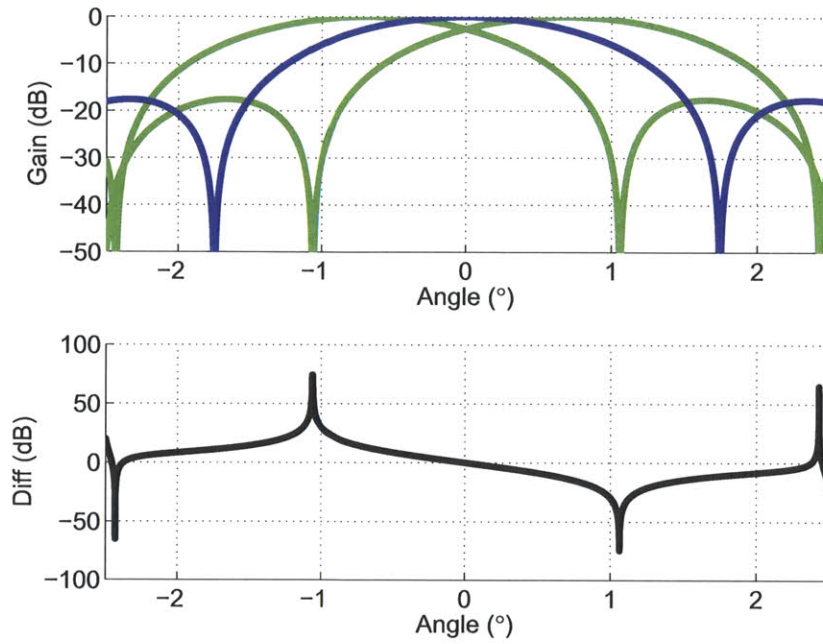


Figure 4-16: One-dimensional antenna beam pattern dither by angle $\pm d\delta$ in either direction. The resulting difference pattern defines the angle off-boresight as a function of the difference.

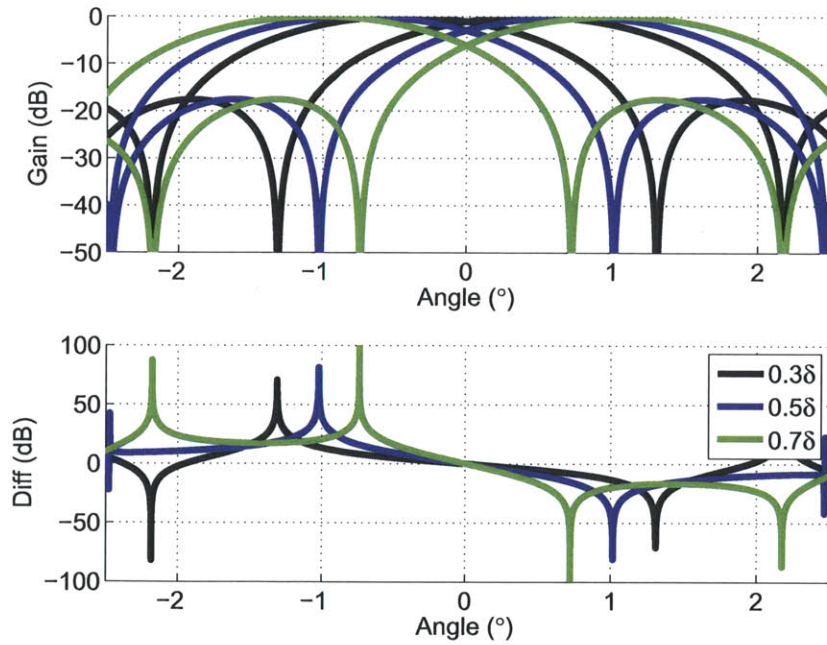


Figure 4-17: One-dimensional antenna beam difference patterns as a function of the dither angle. As the dither increases, the spikes get closer and the slope increases.

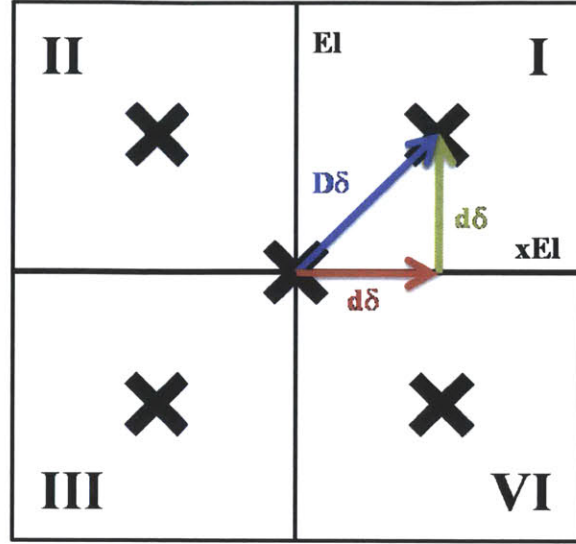


Figure 4-18: Step-tracking algorithm dithers the antenna to four designated points. The origin denotes the current pointing solution and the four dither points are defined for each quadrant.

tracking technique has been restricted to one dimension. In practice, the step-tracking algorithm is concerned with pointing error in two dimensions.

Different two-dimensional step-tracking algorithms have been implemented and studied at Lincoln Laboratory [48]. One project in particular dealt with the same MILSATCOM program in a fixed ground terminal problem instead of an airborne terminal. That step-tracking algorithm used eight sampled SNR values to assess the pointing error. If the error reached a predetermined threshold, then the computer changed the pointing solution by incremental values in either axis. If the threshold was not exceeded, then no change was made to the solution.

This thesis utilizes a similar algorithm. The pedestal control computer commands the pedestal to point in four successive directions equidistant from the open-loop pointing solution. Figure 4-18 shows the four dither points. It is important to note that the coordinates are no longer azimuth and elevation, but cross-elevation and elevation. Cross-elevation, also referred to as traverse, is directly related to azimuth, but also depends on the current elevation angle. As the elevation angle increases, the azimuth gimbal has to sweep larger ranges to dither the antenna by the same AOB as indicated by

$$\theta_{\text{dither}} = \theta \pm d\delta \quad (4.33)$$

and

$$\psi_{\text{dither}} = \psi \pm d\delta / \cos(\theta_{\text{dither}}) \quad (4.34)$$

These two equations define the azimuth and elevation commands for the four dither points of Figure 4-18. This change in reference frame is directly related to the keyhole problem because at an elevation angle of 90° , the azimuth dither command approaches infinity. This reference frame change is not a problem as long as the keyhole is avoided.

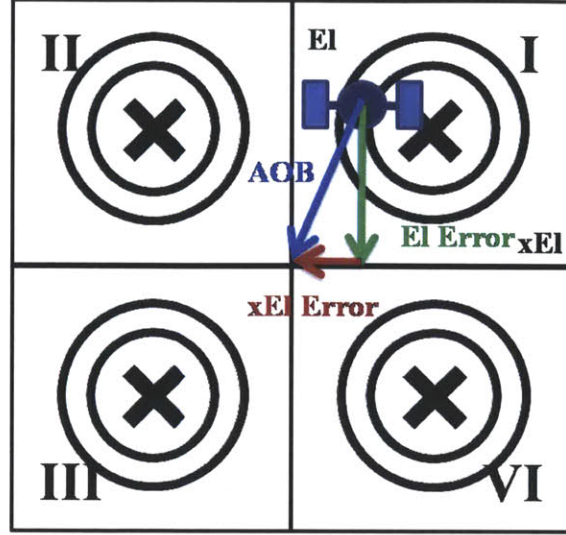


Figure 4-19: Angle-off-boresight between the terminal's pointing solution defined by the origin and the target satellite's position. The concentric circles around the dither points indicate the antenna's beam pattern as it is dithered.

Each dither steers the terminal antenna to a new angle, which changes the point in the terminal's antenna beam pattern through which the signal passes and changes the SNR. This SNR helps estimate the terminal's pointing error as demonstrated in Figure 4-19.

The pedestal control computer compares the four SNR samples and estimates the azimuth and elevation error. The theory presented so far has been restricted to one dimension. The actual problem is in two dimensions, which increases the problem's complexity. Figure 4-18 identifies the four dither points and Figure 4-19 defines the beam patterns at each point and the satellite's location. The SNRs from each quadrant are substituted into sum and difference equations as defined by

$$\Delta xEl = (SNR_I + SNR_{IV}) - (SNR_{II} + SNR_{III}) \quad (4.35)$$

and

$$\Delta El = (SNR_I + SNR_{II}) - (SNR_{III} + SNR_{IV}) . \quad (4.36)$$

The cross-elevation and elevation errors are found using a similar relation to the one-dimensional case. First two antenna beam patterns are dithered by $d\delta$ the dither angle in one axis and added together. The summed beam pattern is then dithered again by $d\delta$ and subtracted. The resulting difference pattern uses Equations 4.35 and 4.36 to approximate the AOB in both axes because the direct calculation of the error is not possible. The difference pattern approximates the AOB and reduces the error by at least 80%. Figure 4-20 presents the actual and normalized AOB approximation error as a function of the AOB magnitude and direction. The reduction in normalized error at 45° , 135° , 225° , and 315° is caused by the peak of the antenna beam approaching the AOB. In the worst case scenario of the pointing error being $0.5 \cdot \text{HPBW}$, the

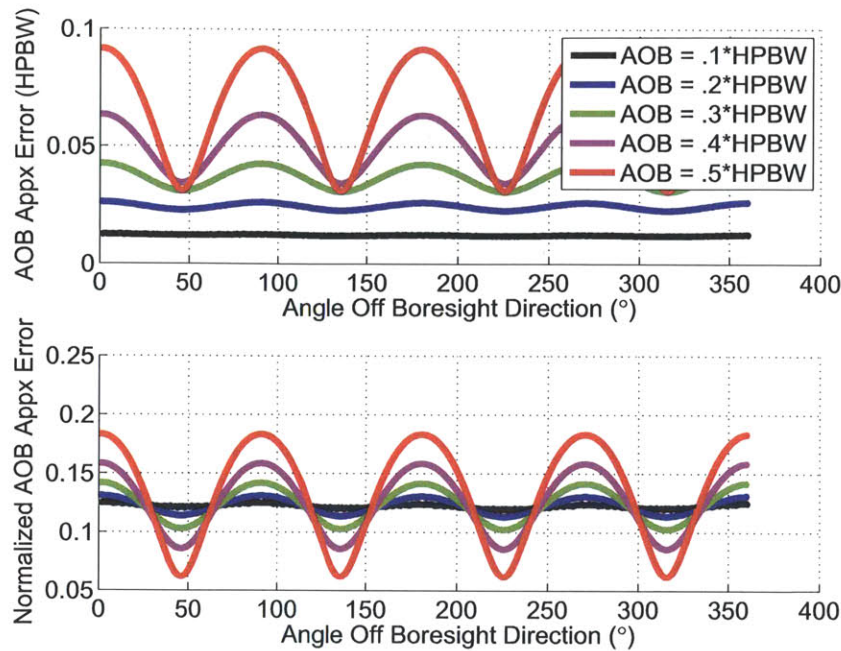


Figure 4-20: Angle off-boresight approximation error as a function of the angle off-boresight magnitude and direction.

correction produces an AOB of $0.14 \cdot \text{HPBW}$. This new AOB reduces the previous 3 dB loss to a mere 0.21 dB. If the step-tracking algorithm is performed periodically, then the pointing solution will continue to improve. This analysis so far assumes ideal SNR dither sample.

4.3.2 Test Case Scenarios

The purpose of performing closed-loop tracking is to track out and eliminate a bias error from the open-loop pointing solution. The two performance criteria that must be assessed are how well closed-loop tracking reduces a pointing bias and how much error exists in the closed-loop tracking process. Two different test case scenarios simulate and assess these criteria as shown in Figure 4-21. The first scenario is setting the pedestal on-boresight and observing the error in the final pointing solution. The second scenario is setting the pedestal off-boresight and observing how well it eliminates the bias. The focus of this thesis is a tracking problem, which means the system has established a link with the satellite and the pedestal is pointing the antenna's main beam at the satellite. The second scenario's initial condition is placing the satellite on the edge of the antenna's HPBW ring. It is important to note that the graphs are for a 0.3 m radius antenna aperture and that only the scaling of the cross-elevation and elevation axes change for a different sized apertures.

This analysis deals with cross-elevation instead of azimuth. Step tracking is con-

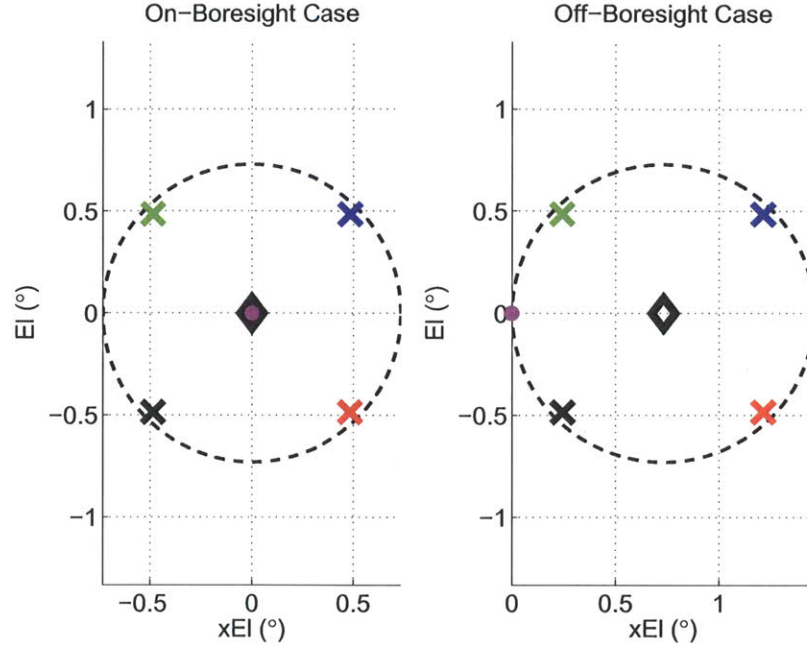


Figure 4-21: Graphical representation of the problem statement test cases. The black diamond represents the antenna's boresight and the dashed circle is the HPBW radius. The magenta dot indicates the satellites location and four X's are the dither points.

cerned with the AOB and not the total pointing error from Chapter 3. The AOB indicates the angle between the terminal and satellite and results in a change in receiver antenna gain. The total pointing error is strictly the root mean square of the azimuth and elevation errors, while the AOB is the pointing error modified based on the current elevation angle as indicated by Equations 4.33 and 4.34.

$$\text{Total Pointing Error} = \sqrt{(\sigma_\psi)^2 + (\sigma_\theta)^2} \quad (4.37)$$

and

$$\text{Angle Off Boresight} = \sqrt{(\sigma_\psi \cos(\theta))^2 + (\sigma_\theta)^2}. \quad (4.38)$$

The analysis in the remainder of this section and Section 4.4 work in cross-elevation/elevation instead of azimuth/elevation. This change of variable means the process is normalized and independent of the current elevation angle as would not be the case in working with azimuth/elevation. The analysis does not change, but the implementation in hardware does change as indicated by Equations 4.33 and 4.34.

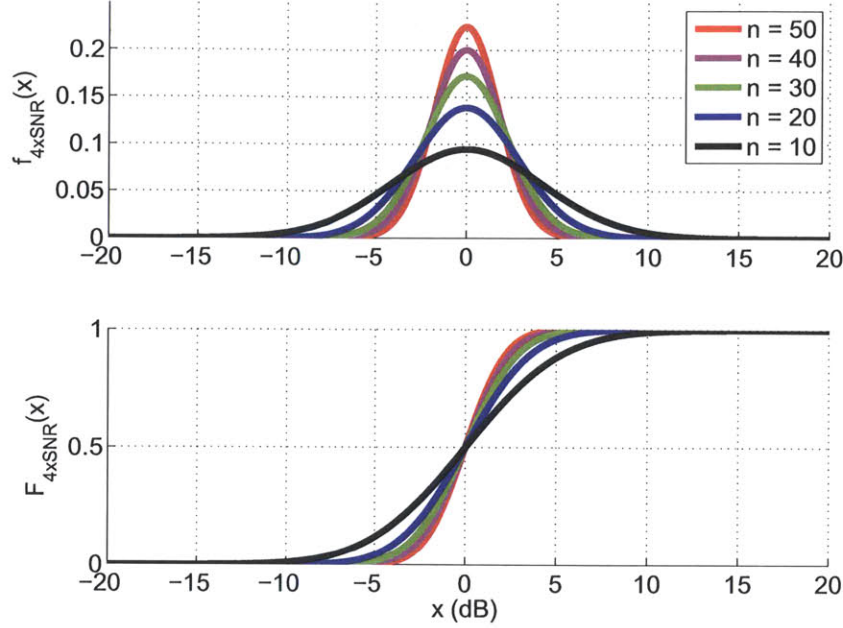


Figure 4-22: Distribution on random variable term of the sum and difference equations. Four independent SNR distributions are convolved together.

4.3.3 Step-Tracking SNR Characterization

Section 4.2 presented the characterization and modeling of the SNR distribution. The distribution defines the probability of estimating the SNR within a certain range of the average value as a function of the number of hops. As the number of hops increases, the estimate of the average SNR improves. In this section, the SNR distribution at an AOB is independent of the dither scheme, that is, G_R is deterministic. Equation 4.1 is separated out into random and deterministic parts as defined by

$$\text{SNR}_{\text{dB}} = 10 \log_{10} \left(\frac{\text{RIP}}{N} \right) + 10 \log_{10}(G_R). \quad (4.39)$$

The first term contains the random components of the SNR, while the second term reflects the bias imposed by the dither scheme. Assuming for the time being that G_R is constant and not changing with each sample, the four SNR distributions previously characterized are convolved to create joint PDFs for Equations 4.35 and 4.36. Figure 4-22 presents the resulting joint distribution centered around zero dBW. This distribution defines the random variable term of both sum and difference equations. As the number of hops increases, the distribution becomes tighter around zero, which translates to a smaller approximation error in calculating the AOB.

In the case where the antenna is pointed on-boresight, the dither bias in Equations 4.35 and 4.36 equals zero, and the PDF and CDF for each equation look like Figure 4-22. Because both distributions have equal variance with zero mean, the

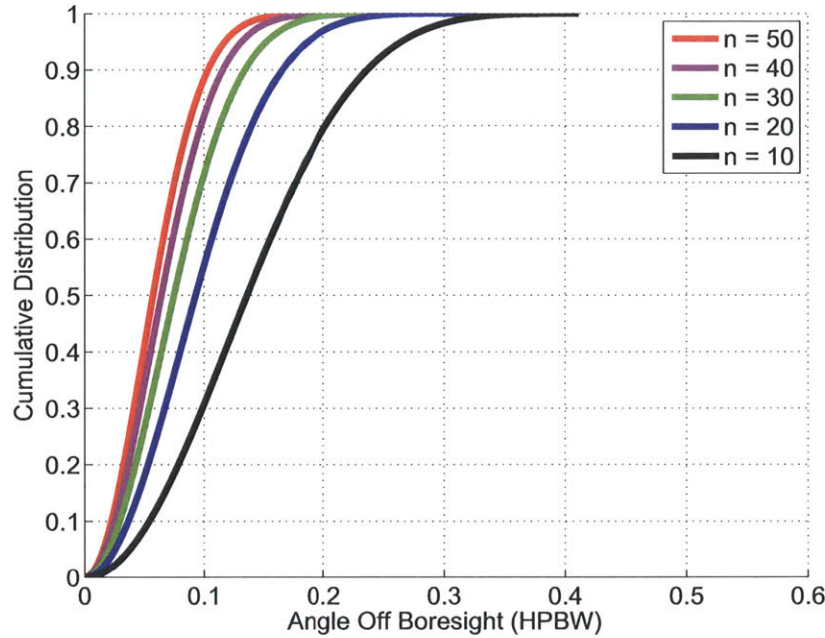


Figure 4-23: Closed-loop tracking pointing error distribution in terms of HPBW as a function of the number of hops.

distribution on the magnitude of the error is a Rayleigh distribution. This Rayleigh distribution is indexed to the difference pattern, which defines the closed-loop tracking pointing error distribution when the pedestal is on-boresight. Figure 4-23 presents the pointing error probability for closed-loop tracking as a function of the number of hops. As long as a reasonable number of hops are sampled, the AOB stays well within the antenna aperture's HPBW. It is important to note that the pointing error distribution as presented in Figure 4-23 is independent of the aperture size.

This analysis is only valid when the AOB between the pedestal and satellite is zero, which means the sum and difference equations have zero mean. In the off-boresight case, the mean pointing error is no longer zero, so the distribution is no longer a Rayleigh distribution and the method is no longer valid. The alternative is to simulate the test case to determine an empirical distribution.

4.4 Closed-Loop Pointing Simulation

The analysis in Section 4.3 relies on the assumption that the receiver gain is constant at each dither point. In other words, the pedestal is ideal and points the antenna perfectly with zero deviation. Because of this assumption, parts of the system performance are analyzed theoretically, but simulations are required to fully model the closed-loop pointing performance. Up to this point, the analysis assumes an ideal pedestal. This assumption is reasonable for a fixed ground station under the right

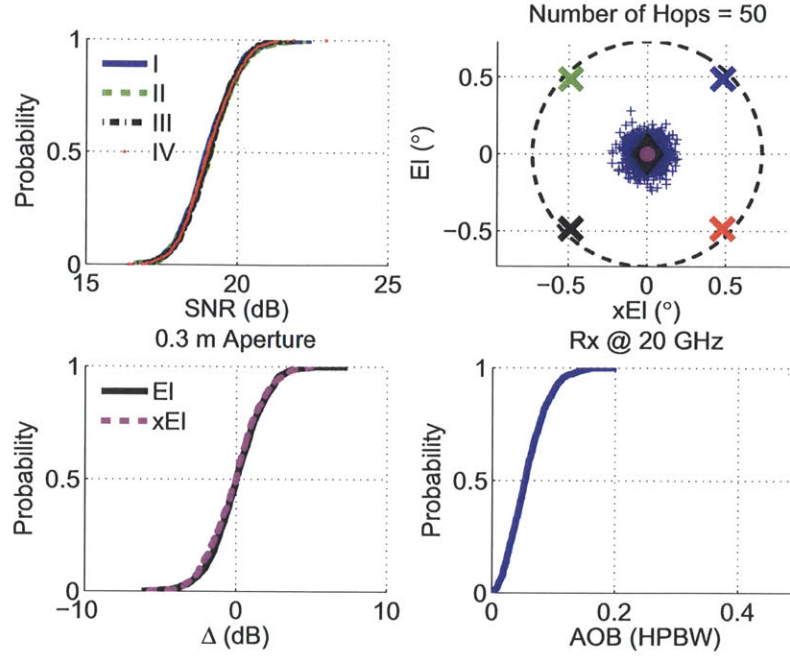


Figure 4-24: Closed-loop pointing simulation of ideal pedestal pointed on-boresight using SNRs averaged over 50 hops. The final pointing solution is within 20% of the antenna's HPBW.

conditions. Chapter 3 demonstrates that in an airborne environment, the aircraft disturbances and GPS/INS random errors cause the pedestal to mispoint during flight. This pointing error adds an order of complexity to the system performance because the receiver antenna gain is no longer deterministic, which influences the SNR distribution. By way of simulation, this section wraps up the unfinished performance modeling for an ideal pedestal and then simulates the complete closed-loop pointing system with aircraft disturbance and GPS/INS errors. It then concludes with interpretations of the results.

4.4.1 Ideal Pedestal Simulation

Figure 4-23 demonstrates the pointing performance of the closed-loop tracking algorithm for an ideal pedestal, which defines the probability of going off-boresight when the antenna is actually pointed on-boresight. A MATLAB Simulink simulation tested both on and off-boresight cases first defined by Figure 4-21. Before proceeding to the off-boresight case, the on-boresight simulation was tested to verify the theoretical calculations from Section 4.3. Figure 4-24 presents the results from the simulation. The figure consists of four graphs, three of which are CDFs of the various calculations and the fourth is a graphical representation of the antenna beam pattern as previously seen in Figure 4-21. The four X's in the upper, right graph are the four dither points, which correspond to the distribution in the upper, left graph. The lower, left graph is

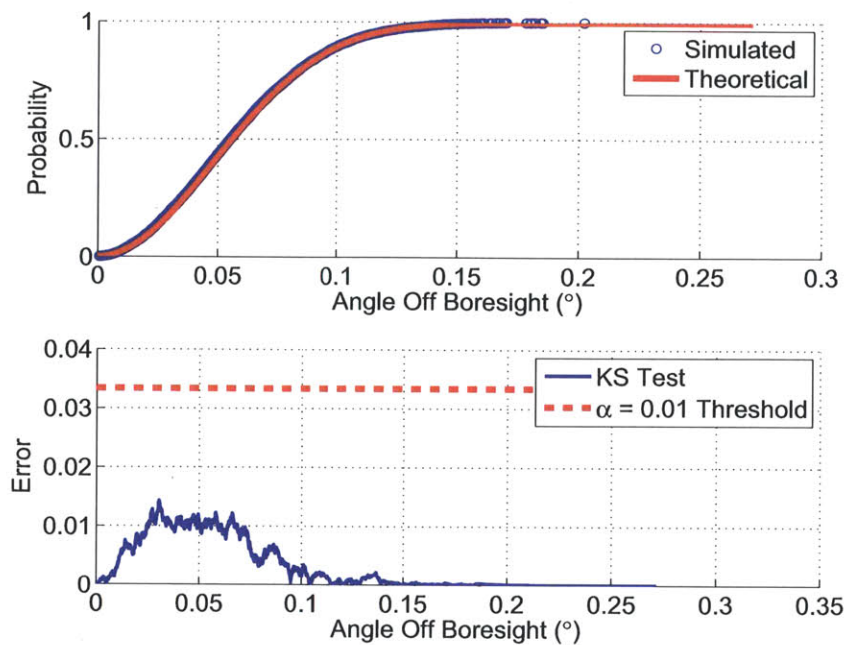


Figure 4-25: Error between simulated and theoretical distributions of closed-loop tracking error using SNRs averaged over 50 hops.

the probability of the sum and difference equations and the lower, right graph is the probability of final pointing solution being within a particular fraction of the HPBW. The blue plus signs in the upper, right graph indicate the calculated final pointing solutions for each iteration. Figure 4-24 demonstrates the system performance using SNRs averaged over 50 hops.

It is important to note that the lower, right graph corresponds to the first data set of Figure 4-23. In an effort to verify that the simulated and theoretical distributions are equal, a KS test is performed with a confidence level of $\alpha = 0.01$. Figure 4-25 is the result and demonstrates that the theoretical and simulated distributions are considered equal.

With the simulated distribution verified, the second test case was simulated and the results are presented in Figure 4-26. The antenna's boresight is pointed off by half of the HPBW. The final pointing solution approaches the satellite's location with a maximum AOB of 20% of the HPBW and a mean of approximately 10% of the HPBW.

Figure 4-27 presents the results from performing this simulation multiple times over increasing numbers of hops. The figure demonstrates that for as few as 10 hops, the final pointing solution reduces the pointing bias below 20% of the HPBW 85% of the time, and therefore closed-loop pointing works for the case of an ideal pedestal.

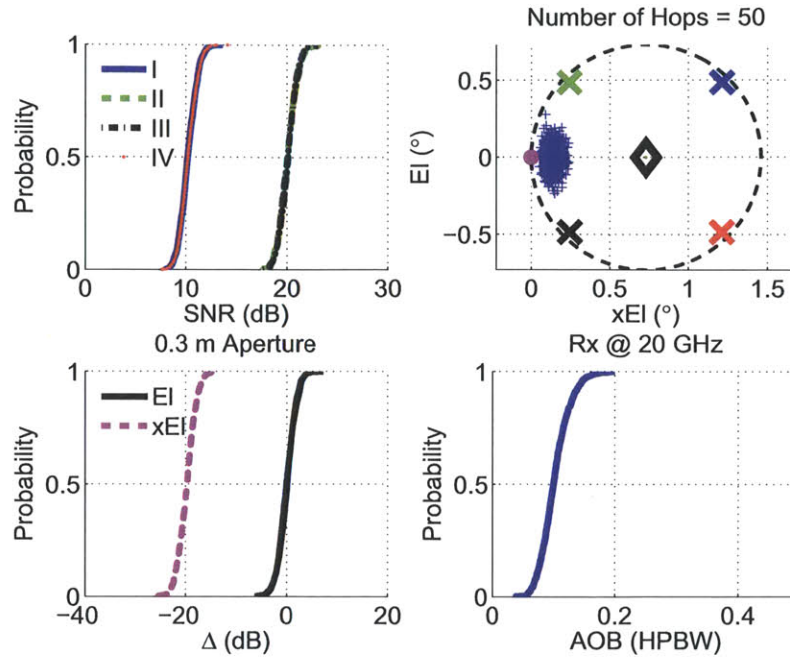


Figure 4-26: Closed-loop pointing simulation of ideal pedestal with the maximum AOB using SNRs averaged over 50 hops. The final pointing solution is within 20% of the antenna's HPBW.

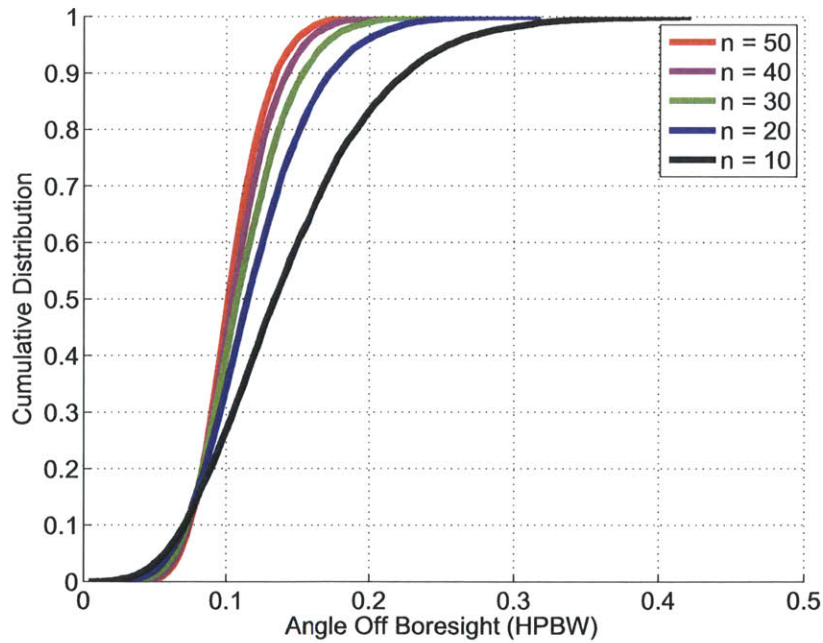


Figure 4-27: Probability of final closed-loop pointing solution as a function of the number of hops.

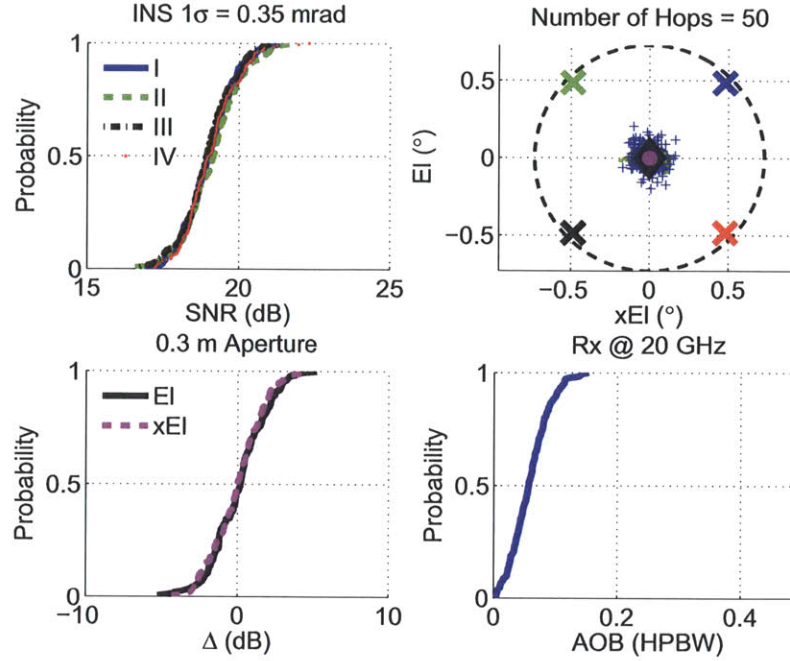


Figure 4-28: Closed-loop pointing simulation of pedestal pointed on-boresight using SNRs averaged over 50 hops. The GPS/INS random error is $1\sigma = 0.35$ mrad.

4.4.2 Closed-loop Pointing Simulation

Pointing errors due to the pedestal cause the receiver antenna gain to change over time, which impacts the SNR distribution. This pointing error is difficult to incorporate into the theoretical model, so simulations are used as an alternative approach. The analysis from Section 3.5 models the pedestal's pointing error as a function of the aircraft dynamics and the GPS/INS random errors.

For this simulation, it is assumed that the step-tracking algorithm is only performed during straight and level flight to reduce the large pointing errors that occur when the aircraft is banked and to avoid step tracking at high elevation angles. The total pointing error output from the simulation are modified to output the AOB as indicated by Equations 4.37 and 4.38. This modification transforms the azimuth error to cross-elevation error. The only other variable is the grade of the GPS/INS.

Figures 4-28 and 4-29 present the simulation results with the highest grade GPS/INS for the on- and off-boresight cases respectively. The green dots within the black diamond are the open-loop pointing errors and the blue, plus signs indicate the final pointing solution.

Figures 4-30 and 4-31 present the performance results from same simulation with the lowest grade GPS/INS. Figure 4-30 demonstrates that despite the increase in open-loop pointing error, the step-track algorithm still gives similar results. The performance varies slightly for the off-boresight case. Figure 4-31 demonstrates that the

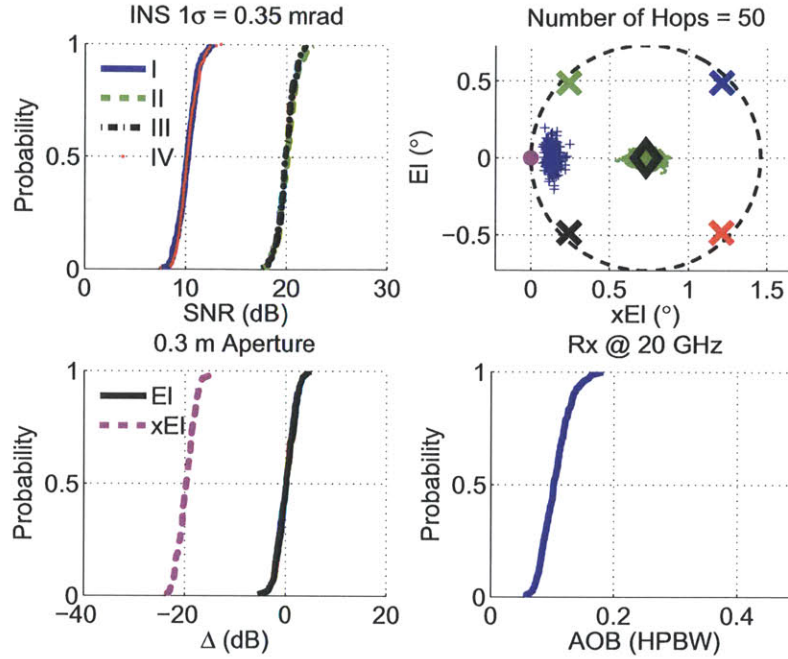


Figure 4-29: Closed-loop pointing simulation of pedestal with the maximum AOB using SNRs averaged over 50 hops. The GPS/INS random error is $1\sigma = 0.35$ mrad.

pointing performance decreases for larger GPS/INS random errors, but still improves the pointing solution and eliminates some portion of the pointing error bias.

Figures 4-32 and 4-33 present the performance results of several simulations as a function of the GPS/INS random errors and the number of sampled hops. Figure 4-32 demonstrates that the pedestal performance is nearly identical for all GPS/INS units and that the performance improves as the number of hops increases. Figure 4-33 demonstrates that for the off-boresight case, the performance varies among GPS/INS units, but regardless of the unit, performance improves at the same rate as the number of hops increases.

It is important to note that for all of the simulations thus far, the antenna aperture has been fixed to a radius of 0.3 m. In the case of an ideal pedestal, simulation performance is identical for different sized apertures. This statement does not hold when open-loop pointing errors enter the simulation. Similar to the result between different grade GPS/INS units, the closed-loop tracking performance decreases as the aperture size increases. The reason for this decrease in performance is because the beam pattern gets tighter and the HPBW gets smaller as the aperture increases in size. The pointing error is independent of the aperture size and therefore impacts the closed-loop tracking performance more for larger antenna apertures. Figures 4-34 and 4-35 present the system performance for different aperture sizes with the GPS/INS random errors set to $1\sigma = 1.0$ mrad. The trends in both figures correspond to the trends in Figures 4-32 and 4-34.

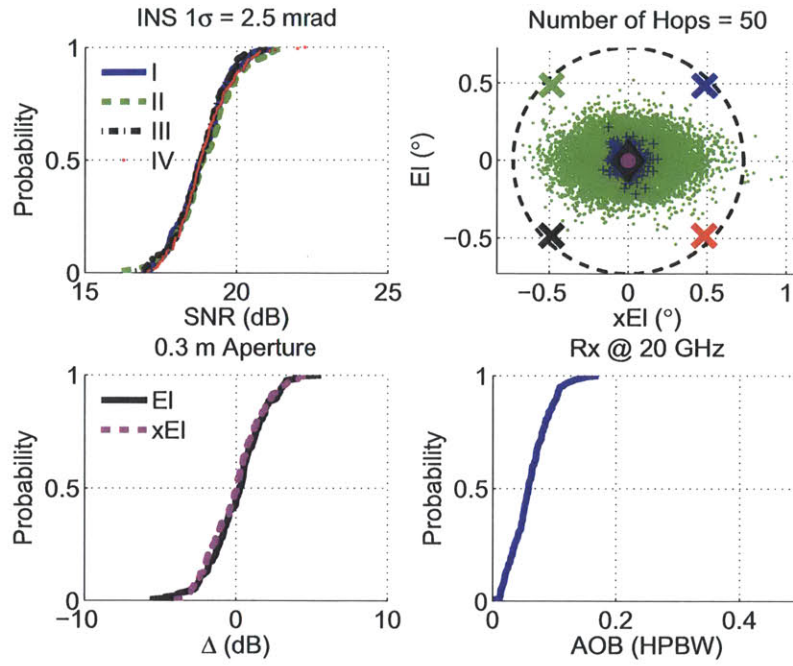


Figure 4-30: Closed-loop pointing simulation of pedestal pointed on-boresight using SNRs averaged over 50 hops. The GPS/INS random error is $1\sigma = 2.5$ mrad.

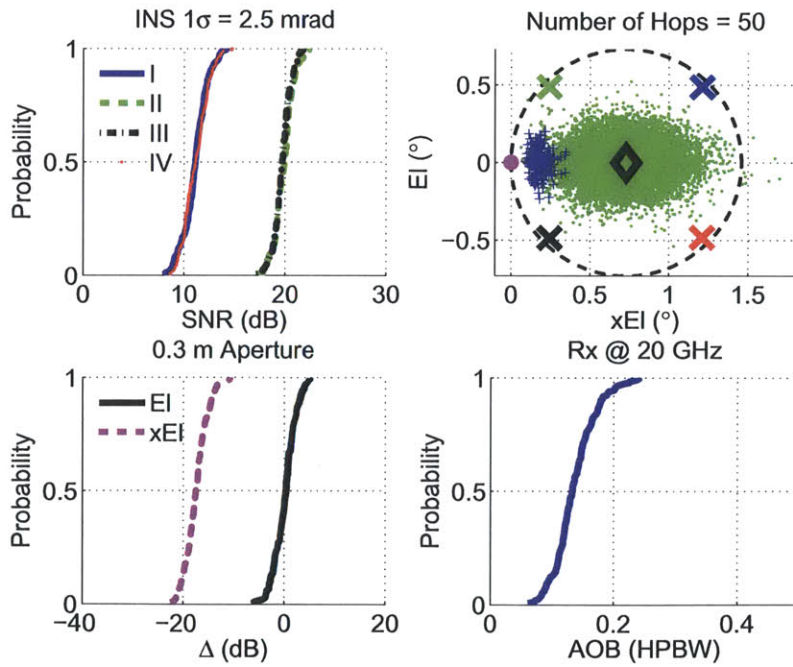


Figure 4-31: Closed-loop pointing simulation of pedestal with the maximum AOB using SNRs averaged over 50 hops. The GPS/INS random error is $1\sigma = 2.5$ mrad.

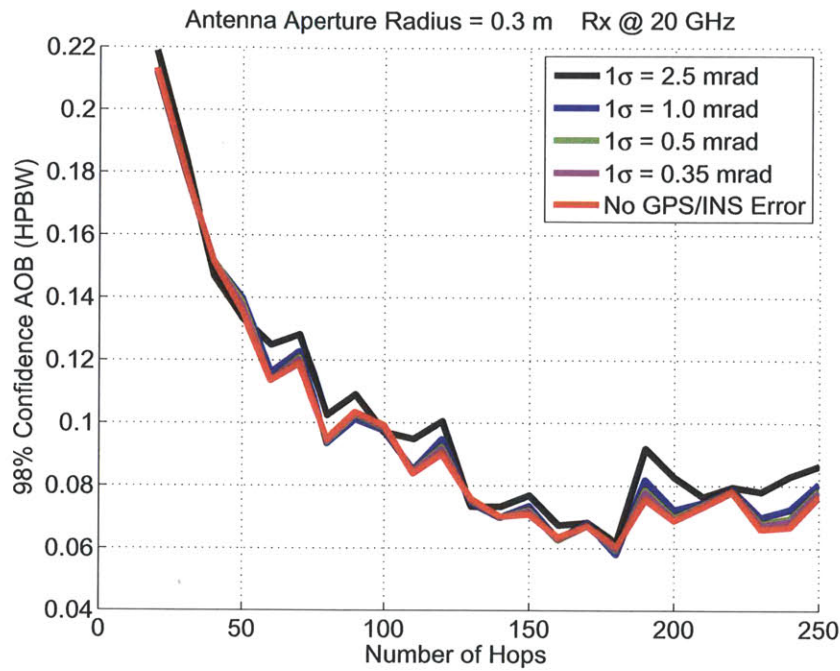


Figure 4-32: 98% confidence interval for final pointing solution when the pedestal is on-boresight as a function of GPS/INS random error and the number of hops.

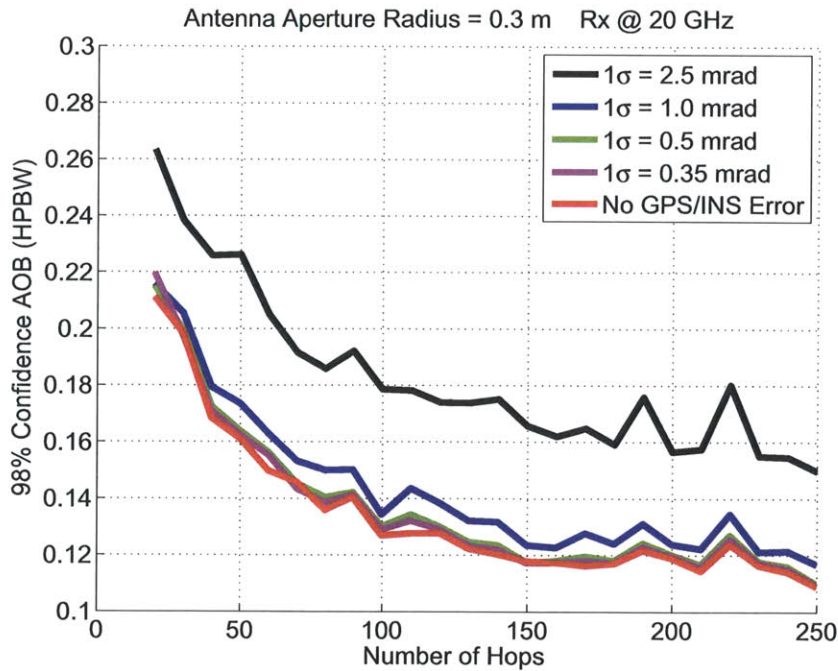


Figure 4-33: 98% confidence interval for final pointing solution when the pedestal is at the maximum AOB as a function of GPS/INS random error and the number of hops.

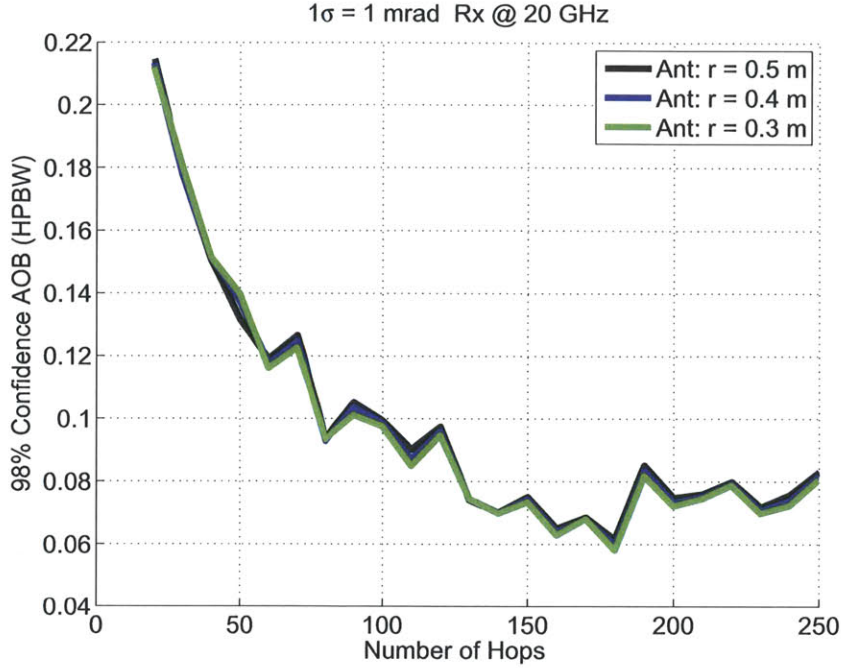


Figure 4-34: 98% confidence interval for final pointing solution when the pedestal is on-boresight as a function of the aperture size and the number of hops.

The four figures indicate that closed-loop tracking performance is nearly identical when the AOB is relatively small. The performance begins to decrease for larger aperture sizes and larger GPS/INS random errors as the AOB increases. Despite this difference in performance, closed-loop tracking in each scenario eliminates a significant portion of the pointing error and improves the communication link. For larger apertures or lower grade GPS/INS units, it may take multiple step-tracking iterations to track out a bias, but the pointing solution improves and tends toward zero AOB.

Because closed-loop tracking works for each grade GPS/INS and antenna aperture, the design tradeoffs are based on the open-loop pointing performance presented in Section 3.5. The uplink antenna beam pattern has a smaller HPBW than the downlink, so the 44 GHz aperture HPBW determines the pointing requirement for reasonable system performance. Figure 4-36 presents the family of CDFs of pointing performance for each grade GPS/INS and compares it to the half of the HPBW of each antenna aperture's uplink antenna beam pattern. The figure shows that if the terminal has a design constraint of pointing within the HPBW 95% of the time for a 0.5 m antenna aperture, then the GPS/INS unit must have a 1σ value no more than 1.0 mrad.

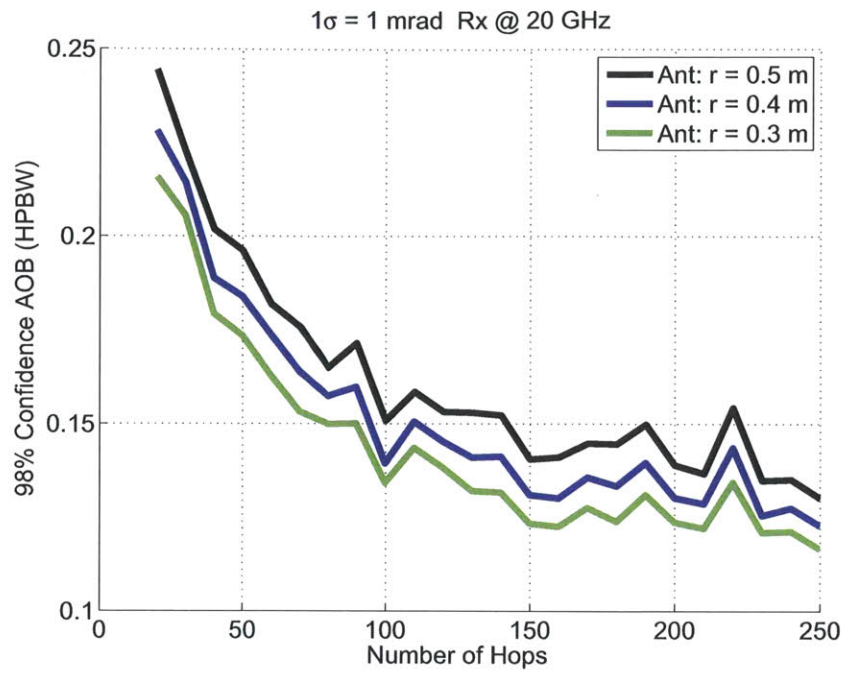


Figure 4-35: 98% confidence interval for final pointing solution when the pedestal is at the maximum AOB as a function of the aperture size and the number of hops.

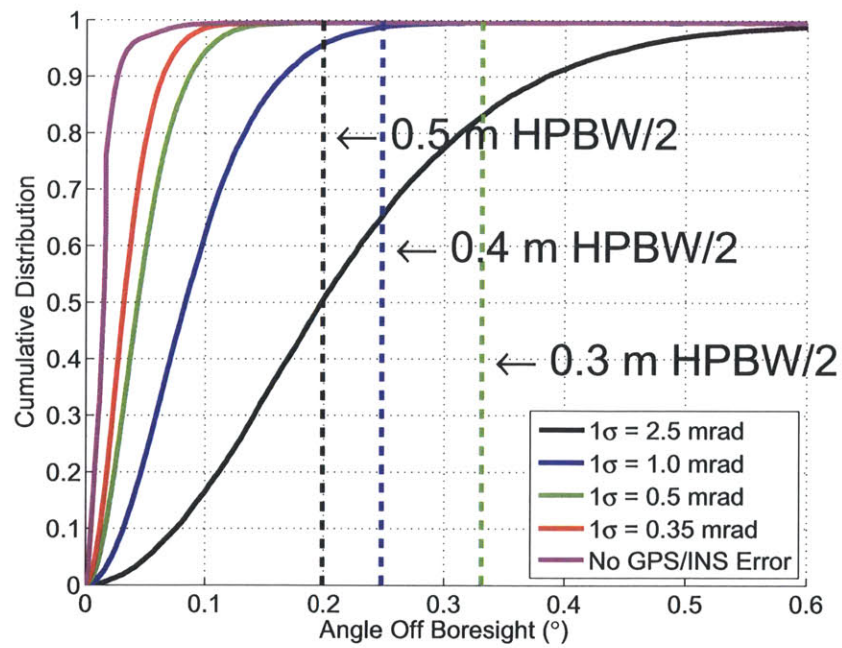


Figure 4-36: Family of cumulative distribution functions of the open-loop pointing performance in reference to uplink 44 GHz antenna aperture HPBW.

Chapter 5

Conclusions and Suggestions for Future Work

This thesis analyzed the pointing and tracking accuracy of an airborne satellite communications terminal. The antenna pedestal system consisted of an antenna mounted on a two-axis gimballed pedestal, with a GPS/INS navigation system that provided position and orientation information. First, the open-loop pointing performance was considered, in the presence of aircraft disturbances and random errors in the GPS/INS attitude information. Next, the closed-loop performance was evaluated. For the closed-loop system, a step-tracking algorithm was developed that estimates the open-loop pointing error using samples of the signal-to-noise ratio. Because the pointing accuracy increases when the signal-to-noise ratio is averaged over a larger number of hops, an analysis was carried out to determine the impact averaging has on the final result. Finally, an assessment of hardware options was performed to understand the relationship between the antenna aperture size and GPS/INS quality, and to analyze overall system performance.

5.1 Conclusions

The following conclusions are reached through the work presented:

1. While the aircraft is banked during the race track portion of the mission, analysis of the power spectral density of the IMU data demonstrates unexplained harmonics at integer frequencies.
2. The open-loop pointing error from the GPS/INS random errors depends only on the pointing solution in the aircraft reference frame, the aircraft's current pitch and roll configuration, and the selected GPS/INS package and is independent of the inertial pointing vector and the aircraft's yaw.
3. The open-loop pointing errors in the azimuth and elevation axes from the

GPS/INS random errors are independent while the aircraft motion is restricted to pitch and roll angles of $\pm 10^\circ$ and $\pm 30^\circ$ respectively.

4. For the received isotropic power distributions investigated in Chapter 4, the distribution of the average noise power has a stronger influence than the distribution of the received isotropic power on the signal-to-noise ratio distribution.
5. The step-tracking algorithm defined in this thesis reduces pointing error in the open-loop pointing solution for both an ideal pedestal and a pedestal experiencing aircraft disturbances and random errors from the GPS/INS.
6. In the ideal pedestal case, closed-loop tracking performance, as measured by the pointing error normalized to the half-power beamwidth, is independent of the antenna aperture size and improves as the number of hops increases.
7. For the airborne pedestal case, the rate of performance improvement as a function of the number of hops is independent of the antenna aperture size and the GPS/INS unit. For the on- and off-boresight case, the 98% confidence level angle off boresight decreases by 0.11 and 0.08 half-power beamwidth, respectively, when the number of hops increases from 20 to 100.
8. For the airborne pedestal pointing on-boresight case, pointing performance relative to the half-power beamwidth is independent of the antenna aperture size and GPS/INS unit.
9. For the off-boresight case, the performance depends on the antenna aperture size and GPS/INS unit. For example, the 98% confidence level angle off boresight decreases by 0.04 half-power beamwidth between the 2.5 and 1.0 mrad GPS/INS units for a 0.3 m antenna aperture.
10. With SNRs averaged over 100 hops and pointing biases less than or equal to 0.5 half-power beamwidth, the step-tracking algorithm reduces the pointing error to within 0.10 half-power beamwidth of the boresight, for all tested configurations.
11. The overall system performance is bounded by the open-loop pointing solution, which is based on hardware selection. Closed-loop tracking performance is a function of the number of sampled hops and is for the most part independent of the hardware selection.

5.2 Suggestions for Future Work

The following issues are identified and presented as suggestions for further work:

1. Transforming correlated open-loop pointing errors. The pointing error characterization of random errors in the GPS/INS assumed that the azimuth and elevation errors were independent. This assumption holds for reasonable ranges

of aircraft motion, but fails at more extreme pitch and roll combinations. Using statistical analysis tools, correlated distributions can be transformed with a change of variables to create an uncorrelated distribution. Transforming correlated open-loop pointing error distributions would allow the open-loop pointing characterization to model GPS/INS random errors for more extreme aircraft orientations.

2. Creating an algorithm that detects and rejects sidelobe measurements. The simulated closed-loop tracking scenarios in this thesis restricted the initial pointing error bias to within the half-power beamwidth region of the antenna aperture. This was done because any larger errors would cause some of the dither points to fall outside of the main beam and on a sidelobe, which the tracking algorithm cannot handle. Creating an algorithm that detects and rejects sidelobe measurement is a possible way to use the step-tracking algorithms for larger pointing errors.
3. Determining the optimum time interval between step-tracking iterations. Overall system performance decreases during the step-tracking algorithm because the pedestal is intentionally pointing off-boresight to assess the pointing error. The step-tracking algorithm should not be performed too frequently to cut down on performance loss, but should be performed periodically to check for any pointing error. Determining the optimum time interval between step-tracking iterations would balance the long- and short-term system performance.
4. Analyzing step-tracking algorithms with thresholds. The step-tracking algorithm presented in this thesis approximated the angle-off-boresight and always updated the pointing solution. The algorithm presented in Figucia's report [48] only corrects for errors above a threshold. These thresholds restrict the pedestal from moving off-boresight when it actually is on-boresight. It is unclear how to define the thresholds based on system performance. Analyzing step-tracking algorithms with thresholds would determine if thresholds improve system performance.
5. Integrating acquisition and tracking phases into one simulation. This thesis focused on the tracking phase of the communication link. The acquisition phase is the process of locating and acquiring the satellite's receive signal. Marsh's thesis [49] focused on analyzing and simulating the acquisition phase of the communication link. Integrating acquisition and tracking phases into one simulation would create a tool that could rigorously test the entire communication link.

Appendix A

Antenna Pedestal Equations of Motion

Overview

The following code was written using the Wolfram Mathematica programming language, Version 7.0.

A.1 APS.nb

```
%%Equations of Motion for APS
Ta = {{Txa},{Tya},{Tza}}
Tb = {{Txb},{Tyb},{Tzb}}
Ja = {{Ixa, 0,0},{0, Iya, 0},{0,0, Iza}}
Jb = {{Ixb, 0,0},{0, Iyb, 0},{0,0, Izb}}
Lbp = {{Cos[\Psi[t]],Sin[\Psi[t]],0},
{-Sin[\Psi[t]],Cos[\Psi[t]],0},{0,0,1}}
Lab = {{Cos[\Theta[t]],0,-Sin[\Theta[t]]},
{0,1,0},{Sin[\Theta[t]],0,Cos[\Theta[t]]}}
Lpb = FullSimplify[Inverse[Lbp]]
Lba= FullSimplify[Inverse[Lab]]
%%Body Rate Conversions
\[Omega] = {{p[t]},{q[t]},{r[t]}};
{{pb[t]},{qb[t]},{rb[t]}}=Lbp.\[Omega]+
{{0},{0},{\Psi'[t]}};
\[Omega]b = {{pb[t]},{qb[t]},{rb[t]}};
{{pa[t]},{qa[t]},{ra[t]}} = Lab.\[Omega]b+
{{0},{\Theta'[t]},{0}};
\[Omega]a={{pa[t]},{qa[t]},{ra[t]}};
\[Alpha] = D[\[Omega],t];
\[Alpha]b = D[\[Omega]b,t];
\[Alpha]a = D[\[Omega]a,t];
Export["vel.txt",{\[Omega],[\[Omega]b,[\[Omega]a]};
Export["acc.txt",{\[Alpha],[\[Alpha]b,[\[Alpha]a]};
%%Antenna Body Frame
```

```

Ha = Ja. \[Omega]a;
Hidot = D[Ha,t];
A = Hidot + Cross[Transpose[{{pa[t]}, {qa[t]}, {ra[t]}}][[1]],
Transpose[Ha][[1]]];
A = FullSimplify[A];
Tya = T_EL;
FullSimplify[Solve[A==Ta, {Txa, Tza}]]
T1 = {{Cos[\[Theta][t]] (- (Ixa+Iya-Iza)
(\[Theta]^\[Prime])[t] (r[t]+(\[Psi]^\[Prime])[t]))+
Cos[\[Psi][t]] (Ixa (p^\[Prime])[t]+q[t] ((-Iya+Iza)
r[t]+(Ixa-Iya+Iza) (\[Psi]^\[Prime])[t]))+Sin[\[Psi][t]]
(Ixa (q^\[Prime])[t]+p[t] (-Ixa (\[Psi]^\[Prime])[t]+
(Iya-Iza) (r[t]+(\[Psi]^\[Prime])[t])))+
1/2 Sin[\[Theta][t]] (2 (-Iya+Iza) Cos[2 \[Psi][t]]
p[t] q[t]+(Iya-Iza) (p[t]^2-q[t]^2) Sin[2 \[Psi][t]]-2
(Ixa+Iya-Iza) Cos[\[Psi][t]] p[t] (\[Theta]^\[Prime])
[t]-2 (Ixa+Iya-Iza) q[t] Sin[\[Psi][t]]
(\[Theta]^\[Prime])[t]-2 Ixa
((r^\[Prime])[t]+(\[Psi]^\[Prime]^\[Prime])[t]))},
{T_EL}, {(Ixa-Iya) Cos[\[Theta][t]] Cos[\[Psi][t]]
p[t]^2 Sin[\[Psi][t]]+p[t] ((-Ixa+Iya) Cos[\[Theta][t]]
Cos[2 \[Psi][t]] q[t]+(-Ixa+Iya+Iza) Cos[\[Theta][t]]
Cos[\[Psi][t]] (\[Theta]^\[Prime])[t]+Sin[\[Theta][t]]
Sin[\[Psi][t]] ((-Ixa+Iya) r[t]-(Ixa-Iya+Iza)
(\[Psi]^\[Prime])[t]))+Sin[\[Theta][t]]
(Iza Sin[\[Psi][t]] (q^\[Prime])[t]-(-Ixa+Iya+Iza)
(\[Theta]^\[Prime])[t] (r[t]+(\[Psi]^\[Prime])[t]))+
Cos[\[Psi][t]] (Iza (p^\[Prime])[t]+q[t] ((Ixa-Iya)
r[t]+(Ixa-Iya+Iza) (\[Psi]^\[Prime])[t])))+1/2
Cos[\[Theta][t]] ((-Ixa+Iya) q[t]^2 Sin[2 \[Psi][t]]+
2 (-Ixa+Iya+Iza) q[t] Sin[\[Psi][t]]
(\[Theta]^\[Prime])[t]+2 Iza ((r^\[Prime])[t]+
(\[Psi]^\[Prime]^\[Prime])[t]))}};
%%Intermediate Body Frame
Hb = Jb. \[Omega]b;
Hb_dot = D[Hb,t];
B = Hb_dot + Cross[Transpose[{{pb[t]}, {qb[t]}, {rb[t]}}][[1]],
Transpose[Hb][[1]]];
B = FullSimplify[B];
Tzb = T_AZ;
TT=FullSimplify[Tb-Lba.T1];
Solve[B==TT, {Txb, Tyb, \[Psi]''[t]}]
Solve[A==Ta, {Txa, Tza, \[Theta]''[t]}]
%%Azimuth/Elevation EOM
TEL = (KEL/RaEL)(eEL-KbEL*(\[Theta]^\[Prime])[t]);
TAZ = (KAZ/RaAZ)(eAZ-KbAZ*(\[Psi]^\[Prime])[t]);
Elevation = 1/Iya (TEL-(Ixa-Iza) Cos[2 \[Theta][t]]
(Cos[\[Psi][t]] p[t]+q[t] Sin[\[Psi][t]]) (r[t]+
(\[Psi]^\[Prime])[t])-Iya Cos[\[Psi][t]] ((q^\[Prime])
[t]-p[t] (\[Psi]^\[Prime])[t])+Iya Sin[\[Psi][t]]
((p^\[Prime])[t]+q[t] (\[Psi]^\[Prime])[t])-1/2
(Ixa-Iza) Sin[2 \[Theta][t]] (Cos[\[Psi][t]]^2 p[t]^2+
q[t]^2 Sin[\[Psi][t]]^2+p[t] q[t] Sin[2 \[Psi][t]]-
(r[t]+(\[Psi]^\[Prime])[t])^2))
El = FullSimplify[Elevation];
Azimuth = -(-TAZ+1/4 (Ixa-2 Iya+Iza+(Ixa-Iza) Cos[2

```

```

\[Theta][t]]) p[t]^2 Sin[2 \[Psi][t]]-1/4 (Ixa-2 Iya+
Iza+(Ixa-Iza) Cos[2 \[Theta][t]]) q[t]^2
Sin[2 \[Psi][t]]+1/2 (Ixb-Iyb) (-2 Cos[2 \[Psi][t]]
p[t] q[t]+(p[t]^2-q[t]^2) Sin[2 \[Psi][t]])+1/2
(Ixa+Iza) (r^\[Prime])[t]+Izb (r^\[Prime])[t]-1/2
(Ixa-Iza) Cos[2 \[Theta][t]] (r^\[Prime])[t]-1/2 p[t]
((Ixa-2 Iya+Iza+(Ixa-Iza) Cos[2 \[Theta][t]])
Cos[2 \[Psi][t]] q[t]+(Ixa-Iza) r[t] Sin[2 \[Theta][t]]
Sin[\[Psi][t]]-2 (Iya+(-Ixa+Iza) Cos[2 \[Theta][t]])
Cos[\[Psi][t]] (\[Theta]^\[Prime])[t])-1/2 q[t]
((-Ixa+Iza) Cos[\[Psi][t]] r[t] Sin[2 \[Theta][t]]-2
(Iya+(-Ixa+Iza) Cos[2 \[Theta][t]]) Sin[\[Psi][t]]
(\[Theta]^\[Prime])[t])-1/2 (Ixa-Iza) Sin[2 \[Theta][t]]
(Cos[\[Psi][t]] (p^\[Prime])[t]+Sin[\[Psi][t]]
(q^\[Prime])[t]-2 (\[Theta]^\[Prime])[t] (r[t]+
(\[Psi]^\[Prime])[t])))/((Ixa+Iza)/2+Izb-1/2 (Ixa-Iza)
Cos[2 \[Theta][t]])
Az = FullSimplify[Azimuth];
Export["ddTheta.txt",El];
Export["ddPsi.txt",Az];
JacobA = FullSimplify[{{0,1,0,0},{D[Az,\[Psi][t]],D[Az,
\[Psi]'[t]],D[Az,\[Theta][t]],D[Az,\[Theta]'[t]]},
{0,0,0,1},{D[El,\[Psi][t]],D[El,\[Psi]'[t]],
D[El,\[Theta][t]],D[El,\[Theta]'[t]]}}]
JacobB = FullSimplify[{{0,0,0,0,0,0,0,0},
{D[Az,eAZ],D[Az,eEL],D[Az,p[t]],D[Az,q[t]],
D[Az,r[t]],D[Az,p'[t]],D[Az,q'[t]],D[Az,r'[t]]},
{0,0,0,0,0,0,0,0},
{D[El,eAZ],D[El,eEL],D[El,p[t]],D[El,q[t]],
D[El,r[t]],D[El,p'[t]],D[El,q'[t]],D[El,r'[t]]}
}]
Export["JacobA.txt",JacobA];
Export["JacobB.txt",JacobB];
Solve[Az==0]
Export["ddTheta.tex",El];
Export["ddPsi.tex",Az];

```

A.2 Final Solution

```

ddPsi = (-4*T_AZ+(Ixa+2*Ixb-2*(Iya+Iyb)+Iza...
+(Ixa-Iza)*cos(2*Theta))*p^2*sin(2*Psi)...
-(Ixa+2*Ixb-2*(Iya+Iyb)+Iza...
+(Ixa-Iza)*cos(2*Theta))*q^2*sin(2*Psi)...
+2*(Ixa+Iza+2*Izb)*dr+2*(-Ixa+Iza)*cos(2*Theta)*dr...
+2*p*(-((Ixa+2*Ixb-2*(Iya+Iyb)+Iza+(Ixa-Iza)*cos(2*Theta))...
*cos(2*Psi)*q)+(-Ixa+Iza)*r*sin(2*Theta)*sin(Psi)...
+2*(Iya+(-Ixa+Iza)*cos(2*Theta))*cos(Psi)*dTheta)...
+2*q*((Ixa-Iza)*cos(Psi)*r*sin(2*Theta)...
+2*(Iya+(-Ixa+Iza)*cos(2*Theta))*sin(Psi)*dTheta)...
-2*(Ixa-Iza)*sin(2*Theta)*(cos(Psi)*dp+sin(Psi)*dq...
-2*dTheta*(r+dPsi)))/(-2*(Ixa+Iza+2*Izb)...
+2*(Ixa-Iza)*cos(2*Theta));
ddTheta = (T_EL-(Ixa-Iza)*cos(2*Theta)*(cos(Psi)*p...

```

```

+q*sin(Psi))*(r+dPsi)+Iya*cos(Psi)*(-dq+p*dPsi)...
+Iya*sin(Psi)*(dp+q*dPsi)-((Ixa-Iza)*sin(2*Theta)...
*(cos(Psi)^2*p^2+q^2*sin(Psi)^2+p*q*sin(2*Psi)...
-(r+dPsi)^2))/2)/Iya;
dra = cos(Theta)*(cos(Psi)*p+q*sin(Psi))*dTheta...
-sin(Theta)*dTheta*(r+dPsi)+sin(Theta)*(cos(Psi)...
*dp+sin(Psi)*dq+cos(Psi)*q*dPsi-p*sin(Psi)*dPsi)...
+cos(Theta)*(dr+ddPsi);
dqa = -(sin(Psi)*dp)+cos(Psi)*dq-cos(Psi)*p*dPsi...
-q*sin(Psi)*dPsi+ddTheta;
dPsi = (ra - sin(Theta)*(cos(Psi)*p+q*sin(Psi)))/cos(Theta)-r;
dTheta = qa - cos(Psi)*q + p*sin(Psi);

```

Appendix B

Matlab Simulation Code

Overview

The following code was written using the MATLAB programming language, Version 7.8.0.347.

B.1 Spectral Analysis

B.1.1 SimSpectra.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SimSpectra
% author: Will Deike
% Created: 8 October 2009 last Modified: 4 May 2010
% Description: matlab file that runs simulink model of
% spectra analysis of Aircraft Disturbances and creates
% shaping filters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Mnum = [1 20 100];
Mden = conv([1 2 1],[1 40 400]);
MG = Mden(end)/Mnum(end)/10;
Nnum = [1 20 100];
Nden = conv([1 2 1],[1 40 400]);
NG = Nden(end)/Nnum(end)/30;
Onum = [1 4 4];
Oden = [1 .4 .04];
OG = Oden(end)/Onum(end)/10;
Fs = 10;
load RT10Hz2.mat
sim('Spectra',1000);
P = logsout.ACVel.P.Data;
Q = logsout.ACVel.Q.Data;
R = logsout.ACVel.R.Data;
M = logsout.MT.Data;
N = logsout.NT.Data;
O = logsout.OT.Data;
figure('Position', [240 212 800 600]);
```



```

subplot(311)
[mp f] = FreqAnalysis(P,Fs);
semilogx(f,10*log10(mp),'b','linewidth',2);
hold on; grid on
[mp f] = FreqSmooth(P,Fs,.005);
semilogx(f,10*log10(mp),'r','linewidth',2);
[mp f] = FreqSmooth(M,Fs,.005);
semilogx(f,10*log10(mp),'g','linewidth',2);
% [mp f] = FreqAnalysis(M,Fs);
% semilogx(f,10*log10(mp),'k');
xlabel('Frequency (Hz)')
ylabel('P (dBW)')
legend('Rough Calc','Smooth Calc',...
'Shaping Filter','location','southwest')
axis([1E-4 10 -150 -50])
subplot(312)
[mp f] = FreqAnalysis(Q,Fs);
semilogx(f,10*log10(mp),'b','linewidth',2);
hold on; grid on
[mp f] = FreqSmooth(Q,Fs,.005);
semilogx(f,10*log10(mp),'r','linewidth',2);
[mp f] = FreqSmooth(N,Fs,.005);
semilogx(f,10*log10(mp),'g','linewidth',2);
% [mp f] = FreqAnalysis(N,Fs);
% semilogx(f,10*log10(mp),'k');
xlabel('Frequency (Hz)')
ylabel('Q (dBW)')
axis([1E-4 10 -150 -50])
subplot(313)
[mp f] = FreqAnalysis(R,Fs);
semilogx(f,10*log10(mp),'b','linewidth',2);
hold on; grid on
[mp f] = FreqSmooth(R,Fs,.005);
semilogx(f,10*log10(mp),'r','linewidth',2);
[mp f] = FreqSmooth(O,Fs,.005);
semilogx(f,10*log10(mp),'g','linewidth',2);
% [mp f] = FreqAnalysis(O,Fs);
% semilogx(f,10*log10(mp),'k');
xlabel('Frequency (Hz)')
ylabel('R (dBW)')
axis([1E-4 10 -150 -50])

```

B.1.2 FreqAnalysis.m

```

function [mx f] = FreqAnalysis(x,Fs)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FreqAnalysis
% author: MathWorks Tech-Notes
% Created: 8 October 2009 last Modified: 10 November 2010
% Description: m-file function computes the PSD of a time
% series x, with time step 1/Fs.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
L = length(x);

```

```

nfft = 2^nextpow2(L);
fftx = fft(x,nfft);
NumUniquePts = ceil((nfft+1)/2);
fftx = fftx(1:NumUniquePts);
mx = abs(fftx);
mx = mx/length(x);
mx = mx.^2;
if rem(nfft, 2)
    mx(2:end) = mx(2:end)*2;
else
    mx(2:end -1) = mx(2:end -1)*2;
end
f = (0:NumUniquePts-1)*Fs/nfft;
%mx = mx/max(mx);

```

B.1.3 FreqSmooth.m

```

function [Y f] = FreqSmooth(x,Fs,W)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FreqSmooth
% author: Will Deike
% Created: 8 October 2009 last Modified: 10 November 2010
% Description: m-file function computes the PSD of a time
% series x, with time step 1/Fs the same way as
% FreqAnalysis.m, except that the result is smoothed by
% convolving the PSD with a smoothing function with
% parameter W.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
L = length(x);
nfft = 2^nextpow2(L);
fftx = fft(x,nfft);
NumUniquePts = floor((nfft+1)/2);
fftx = fftshift(fftx);
mx = abs(fftx);
mx = mx/length(x);
mx = mx.^2;
F = (-NumUniquePts+1:NumUniquePts)*Fs/nfft;
res = F(2)-F(1);
H = exp(-abs(F)/W)/W;
Y = conv(mx,H,'same')*res;
Y = Y(NumUniquePts:end);
if rem(nfft, 2)
    Y(2:end) = Y(2:end)*2;
else
    Y(2:end -1) = Y(2:end -1)*2;
end
f = (0:NumUniquePts)*Fs/nfft;

```

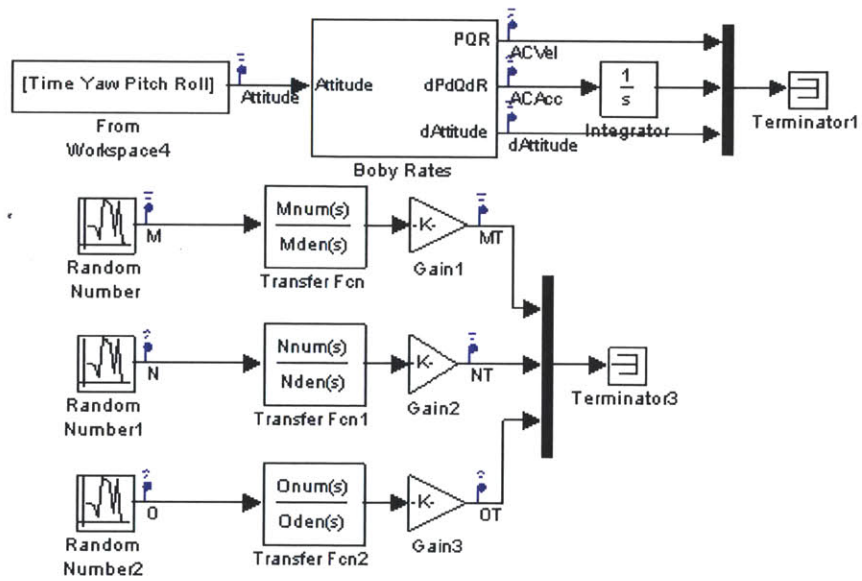


Figure B-1: Spectra.mdl Simulink Model

B.2 Linearized Plant Simulation

B.2.1 SimLinear.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SimLinear.m
% author: William Deike
% Created: 15 January 2010 last Modified: 20 March 2010
% Description: m-file runs the Simulink model to simulate
% the linearized plant model using recorded flight data
% and pedestal moments of Inertia.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all; close all;
K = 18.7; %oz-in/amp
K = K*.007061552; %N-,/amp
Kb = 13.8; %V/RPM
Kb = Kb*60/(2*pi); %V/(rad/s)
Ra = 4.84; %ohms
KAZ = K;
KEL = K;
KbAZ = Kb;
KbEL = Kb;
RaAZ = Ra;
RaEL = Ra;
Ixa = 2712.9273*.4535924*.0254^2;
Iya = 2535.3767*.4535924*.0254^2;
Iza = 3861.0414*.4535924*.0254^2;
Ixb = 3335.6395*.4535924*.0254^2;
Iyb = 2907.7924*.4535924*.0254^2;
Izb = 4217.5204*.4535924*.0254^2;
theta = pi/2; psi = pi/2; eAZ = 0; eEL = 0;
%%
A = [0, 1, 0, 0;...
      0, (2*KAZ*KbAZ)/((-Ixa - Iza - 2*Izb)*RaAZ...
      + (Ixa - Iza)*RaAZ*cos(2*theta)),...
      (-4*eAZ*(Ixa - Iza)*KAZ*sin(2*theta))/(RaAZ...
      *(Ixa + Iza + 2*Izb + (-Ixa + Iza)...
      *cos(2*theta))^2), 0;...
      0, 0, 0, 1;...
      0, 0, 0, -(KbEL*KEL)/(Iya*RaEL)];
B = [0, 0, 0, 0, 0, 0, 0, 0;...
      (2*KAZ)/(RaAZ*(Ixa + Iza + 2*Izb...
      + (-Ixa + Iza)*cos(2*theta))), 0, 0, 0, 0,...
      ((Ixa - Iza)*cos(psi)*sin(2*theta))/...
      (Ixa + Iza + 2*Izb...
      + (-Ixa + Iza)*cos(2*theta)),...
      ((Ixa - Iza)*sin(2*theta)*sin(psi))/...
      (Ixa + Iza + 2*Izb...
      + (-Ixa + Iza)*cos(2*theta)), -1;...
      0, 0, 0, 0, 0, 0, 0;...
      0, KEL/(Iya*RaEL), 0, 0, 0, sin(psi), -cos(psi), 0];
C = eye(4);
D = zeros(4,8);
sys = ss(A,B,C,D);
%%
Ap = [0, 1;...
      0, (2*KAZ*KbAZ)/((-Ixa - Iza - 2*Izb)*RaAZ...
      + (Ixa - Iza)*RaAZ*cos(2*theta))];
```

```

At = [0, 1;...
      0, -((KbEL*KEL)/(Iya*RaEL))];
Bp = [0, 0, 0, 0;...
      (2*KAZ)/(RaAZ*(Ixa + Iza + 2*Izb...
      + (-Ixa + Iza)*cos(2*theta)),...
      ((Ixa - Iza)*cos(psi)*sin(2*theta))...
      /(Ixa + Iza + 2*Izb + (-Ixa + Iza)...
      *cos(2*theta)),...
      ((Ixa - Iza)*sin(2*theta)*sin(psi))...
      /(Ixa + Iza + 2*Izb + (-Ixa + Iza)...
      *cos(2*theta)), -1];
Bt = [0, 0, 0, 0;...
      KEL/(Iya*RaEL), sin(psi), -cos(psi), 0];
Cp = [1 0; 0 1];
Ct = [1 0; 0 1];
Dp = zeros(2,4);
Dt = zeros(2,4);
sysPsi = ss(Ap,Bp,Cp,Dp);
sysTheta = ss(At,Bt,Ct,Dt);
[Kp,S,e] = lqr(Ap,Bp(:,1),...
Cp'*[.001 0; 0 .001]*Cp,.01,0);
K1p = Kp(1);
K2p = Kp(2);
K1t = 50;
K2t = 20;
Kp = [K1p K2p; 0 0; 0 0; 0 0];
Kt = [K1t K2t; 0 0; 0 0; 0 0];
Np = [Ap-Bp*Kp Bp(:,1); Cp(1,:) 0]^(-1)*[0; 0; 1];
Nt = [At-Bt*Kt Bt(:,1); Ct(1,:) 0]^(-1)*[0; 0; 1];
%%
load RT10Hz.mat
sim('LinearPD',1600);
time = logouts.Psi.Time;
psi = logouts.Psi.Data;
theta = logouts.Theta.Data;
AOB = logouts.AOB.Data;
az = logouts.CMD.Data(:,1);
el = logouts.CMD.Data(:,2);
figure('Position', [240 212 800 600]);
subplot(211)
plot(time,az.*180/pi,'g','linewidth',6)
hold on; grid on
plot(time,psi.*180/pi,'b--','linewidth',2)
ylabel('Azimuth (\circ)')
subplot(212)
plot(time,el.*180/pi,'g','linewidth',6)
hold on; grid on
plot(time,theta.*180/pi,'b--','linewidth',2)
ylabel('Elevation (\circ)')
xlabel('Time (sec)')
legend('Reference','Output','location','southeast')
figure('Position', [240 212 800 600]);
plot(time,AOB,'b','linewidth',2)
hold on; grid on
ylabel('Total Pointing Error (\circ)')
xlabel('Time (sec)')
axis([0 1600 0 .5])

```

B.2.2 Linearized Plant Simulink Model

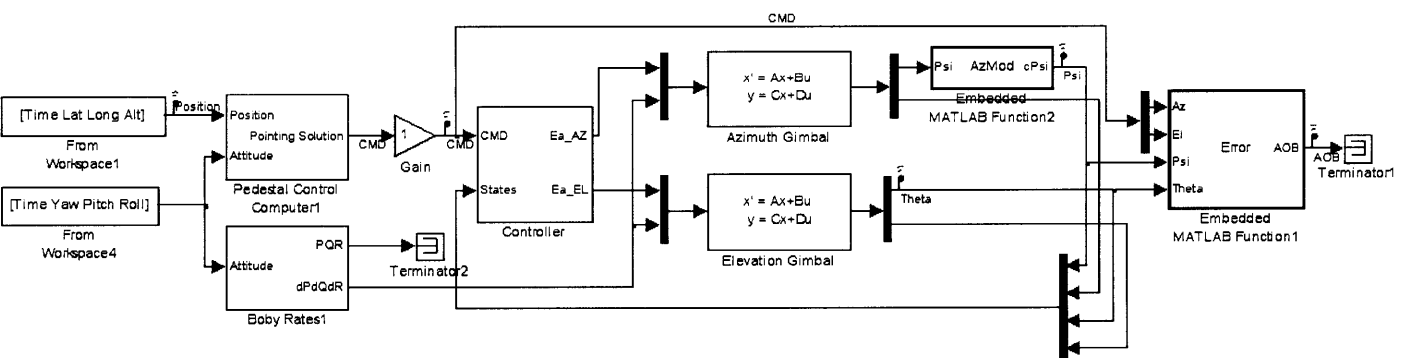


Figure B-2: LinearPD.mdl Simulink Model

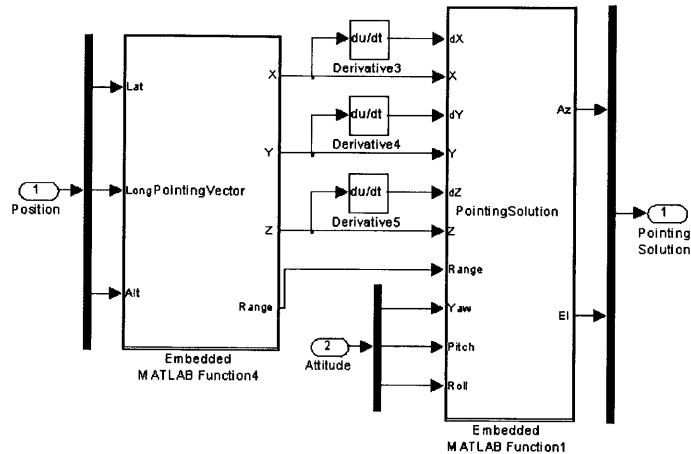


Figure B-3: Pedestal Control Computer Simulink Block

PointingVector.m

```
function [X,Y,Z,Range] = PointingVector(Lat,Long,Alt)
a = 6378137;
b = 6356752.314;
efphi = (a-b)/a;
ecc = sqrt(2*efphi - efphi^2);
% Milstar %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
r = 4120000; %orbital radius (meters)
loc = -90*pi/180; %lambdaitude
Sat = r*[cos(loc) sin(loc) 0];
% 707 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
phi = Lat*pi/180;
lambda = Long*pi/180;
h = Alt;
sinphi = sin(phi);
cosphi = cos(phi);
N = a ./ sqrt(1 - ecc^2 * sinphi.^2);
X = (N + h) .* cosphi .* cos(lambda);
Y = (N + h) .* cosphi .* sin(lambda);
Z = (N*(1 - ecc^2) + h) .* sinphi;
% Pointing Vector %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ecef2ned = [-sin(phi) 0 cos(phi); 0 1 0; -cos(phi) 0 -sin(phi)]*...
    [cos(lambda) sin(lambda) 0; -sin(lambda) cos(lambda) 0; 0 0 1];
XIJ = (Sat-[X Y Z])';
VNED = ecef2ned*XIJ;
nedAz = atan2(VNED(2),VNED(1))*180/pi;
nedEl = asin(-VNED(3)/sqrt(sum(VNED.^2)))*180/pi;
Range = sqrt(sum(VNED.^2));
X = VNED(1); Y = VNED(2); Z = VNED(3);
```

PointingSolution.m

```
function [Az,El] = PointingSolution(dX,X,dY,Y,dZ,Z,Range,Yaw,Pitch,Roll)
```

```

R = [X; Y; Z];
V = [dX; dY; dZ];
Ry = [cos(Yaw) sin(Yaw) 0; -sin(Yaw) cos(Yaw) 0; 0 0 1];
Rp = [cos(Pitch) 0 -sin(Pitch); 0 1 0; sin(Pitch) 0 cos(Pitch)];
Rr = [1 0 0; 0 cos(Roll) sin(Roll); 0 -sin(Roll) cos(Roll)];
T = Rr*Rp*Ry;
Ra = T*R; Va = T*V;
Xa = Ra(1); Ya = Ra(2); Za = Ra(3);
dXa = Va(1); dYa = Va(2); dZa = Va(3);
Az = atan2(Ya,Xa);
El = atan2(-Za,cos(Az)*Xa+sin(Az)*Ya);
dAz = (-sin(Az)*dXa+cos(Az)*dYa)/Range;
dEl = -(cos(El)*dZa+sin(El)*(cos(Az)*dXa+sin(Az)*dYa))/Range;

```

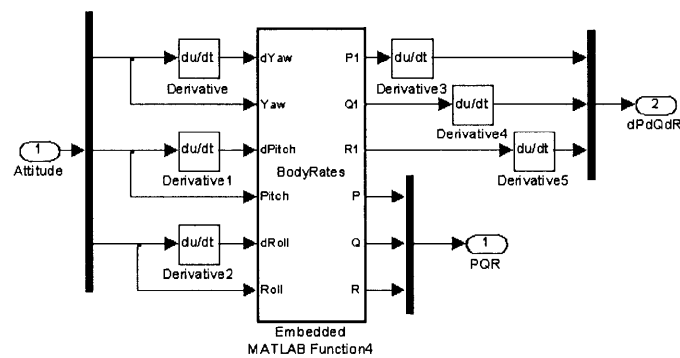


Figure B-4: Body Rates Simulink Block

BodyRates.m

```

function [P1,Q1,R1,P,Q,R] = BodyRates(dYaw,Yaw,dPitch,Pitch,dRoll,Roll)
if dYaw >= 50
    dYaw = dYaw - 2*pi/.1;
elseif dYaw <= -50
    dYaw = dYaw + 2*pi/.1;
end
P = -sin(Pitch)*dYaw + dRoll;
Q = sin(Roll)*cos(Pitch)*dYaw + cos(Roll)*dPitch;
R = cos(Roll)*cos(Pitch)*dYaw - sin(Roll)*dPitch;
P1 = P;
Q1 = Q;
R1 = R;

```

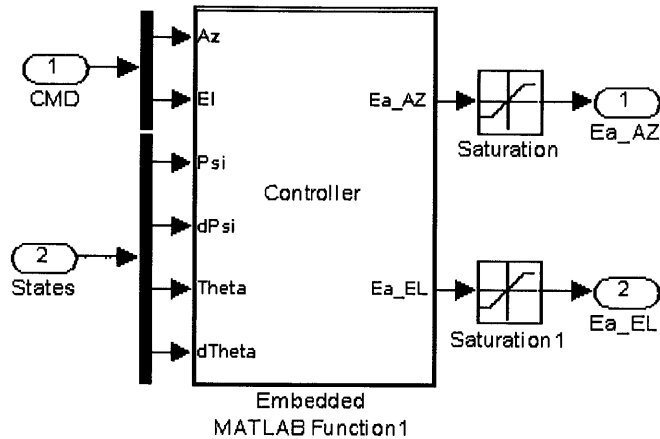


Figure B-5: Controller Simulink Block

Controller.m

```
function [Ea_AZ,Ea_EL] = Controller(Az,El,Psi,dPsi,Theta,dTheta)
KPaz = 2500; KDaz = 100;
KPe1 = 2500; KDe1 = 100;
if Psi-Az >= pi
    Psi = Psi-2*pi;
elseif Psi-Az <= -pi
    Psi = Psi+2*pi;
else
end
Ea_AZ = -(KPaz*(Psi-Az)-KDaz*(dPsi));
Ea_EL = -(KPe1*(Theta-El)-KDe1*(dTheta));
```

B.3 Nonlinear Plant Simulation

B.3.1 SimNonlinear.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SimNonlinear.m
% author: William Deike
% Created: 20 January 2010 last Modified: 20 March 2010
% Description: m-file runs the Simulink model to simulate
% the nonlinear plant model using recorded flight data
% and pedestal moments of Inertia.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all; close all;
load RT10Hz.mat
sim('NonlinearPD',1600);
time = logouts.AOB.Time;
psi = logouts.Sensors.cPsi.Data;
theta = logouts.Sensors.Theta.Data;
AOB = logouts.AOB.Data;
az = logouts.CMD.Az.Data;
el = logouts.CMD.El.Data;
figure('Position', [240 212 800 600]);
subplot(211)
plot(time,az.*180/pi,'g','linewidth',6)
hold on; grid on
plot(time,psi.*180/pi,'b--','linewidth',2)
ylabel('Azimuth (\circ)')
subplot(212)
plot(time,el.*180/pi,'g','linewidth',6)
hold on; grid on
plot(time,theta.*180/pi,'b--','linewidth',2)
ylabel('Elevation (\circ)')
xlabel('Time (sec)')
legend('Reference','Output','location','southeast')
figure('Position', [240 212 800 600]);
plot(time,AOB,'b','linewidth',2)
hold on; grid on
ylabel('Total Pointing Error (\circ)')
xlabel('Time (sec)')
axis([0 1600 0 .5])
```

B.3.2 Nonlinear Plant Simulink Model

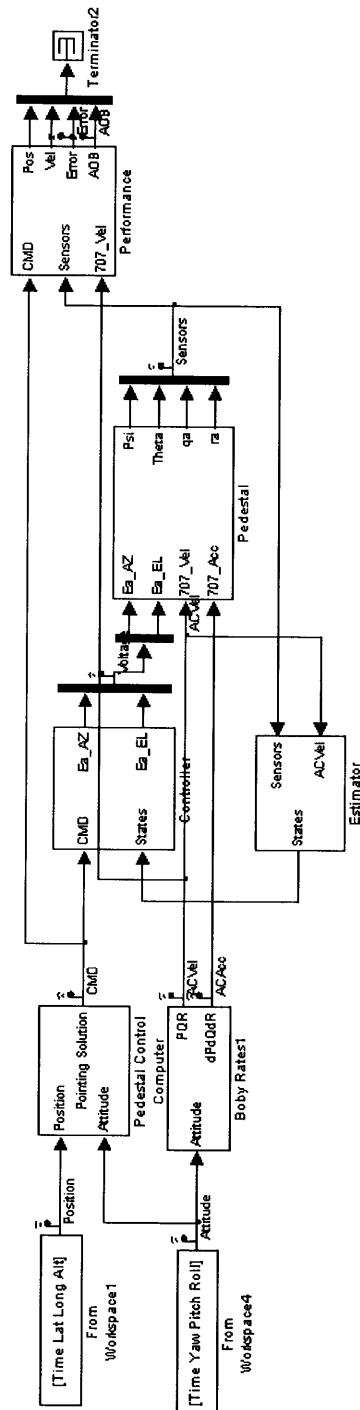


Figure B-6: NonlinearPD.mdl Simulink Model

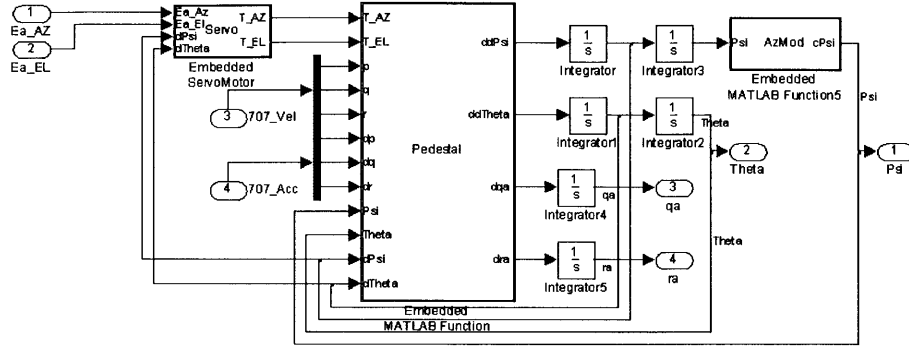


Figure B-7: Pedestal Simulink Block

Pedestal.m

```
function [ddPsi,ddTheta,dqa,dra]= Pedestal(T_AZ,...
    T_EL,p,q,r,dp,dq,dr,Psi,Theta,dPsi,dTheta)
Ixa = 2712.9273*.4535924*.0254^2;
Iya = 2535.3767*.4535924*.0254^2;
Iza = 3861.0414*.4535924*.0254^2;
Ixb = 3335.6395*.4535924*.0254^2;
Iyb = 2907.7924*.4535924*.0254^2;
Izb = 4217.5204*.4535924*.0254^2;
ddPsi = (-4*T_AZ+(Ixa+2*Ixb-2*(Iya+Iyb)+Iza...
    +(Ixa-Iza)*cos(2*Theta))*p^2*sin(2*Psi)...
    -(Ixa+2*Ixb-2*(Iya+Iyb)+Iza...
    +(Ixa-Iza)*cos(2*Theta))*q^2*sin(2*Psi)...
    +2*(Ixa+Iza+2*Izb)*dr+2*(-Ixa+Iza)*cos(2*Theta)*dr...
    +2*p*(-((Ixa+2*Ixb-2*(Iya+Iyb)+Iza+(Ixa-Iza)...
    *cos(2*Theta))*cos(2*Psi)*q)...
    +(-Ixa+Iza)*r*sin(2*Theta)*sin(Psi)...
    +2*(Iya+(-Ixa+Iza)*cos(2*Theta))*cos(Psi)*dTheta)...
    +2*q*((Ixa-Iza)*cos(Psi)*r*sin(2*Theta)...
    +2*(Iya+(-Ixa+Iza)*cos(2*Theta))*sin(Psi)*dTheta)...
    -2*(Ixa-Iza)*sin(2*Theta)*(cos(Psi)*dp...
    +sin(Psi)*dq-2*dTheta*(r+dPsi)))...
    /(-2*(Ixa+Iza+2*Izb)+2*(Ixa-Iza)*cos(2*Theta));
ddTheta = (T_EL-(Ixa-Iza)*cos(2*Theta)*(cos(Psi)*p...
    +q*sin(Psi))*(r+dPsi)...
    +Iya*cos(Psi)*(-dq+p*dPsi)+Iya*sin(Psi)*(dp+q*dPsi)...
    -((Ixa-Iza)*sin(2*Theta)*(cos(Psi)^2*p^2+q^2*...
    sin(Psi)^2+p*q*sin(2*Psi)-(r+dPsi)^2))/2)/Iya;
dra = cos(Theta)*(cos(Psi)*p+q*sin(Psi))*dTheta...
    -sin(Theta)*dTheta*(r+dPsi)+sin(Theta)*...
    (cos(Psi)*dp+sin(Psi)*dq+cos(Psi)*q*dPsi...
    -p*sin(Psi)*dPsi)+cos(Theta)*(dr+ddPsi);
dqa = -(sin(Psi)*dp)+cos(Psi)*dq-cos(Psi)...
    *p*dPsi-q*sin(Psi)*dPsi+ddTheta;
```

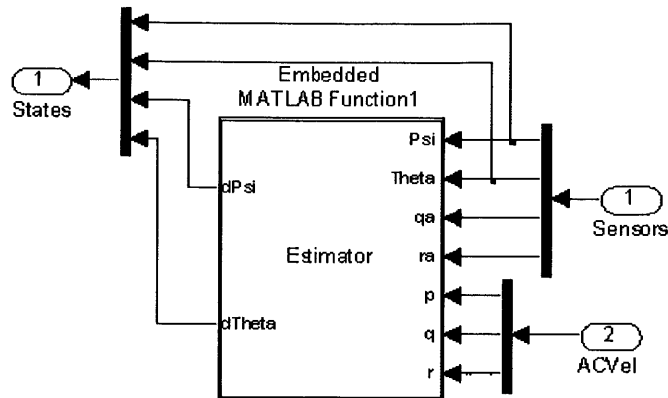



Figure B-8: Estimator Simulink Block

Servo.m

```
function [T_AZ,T_EL] = Servo(Ea_Az,Ea_El,dPsi,dTheta)
K = 18.7; %oz-in/amp
K = K*.007061552; %N-,/amp
Kb = 13.8; %V/RPM
Kb = Kb*60/(2*pi); %V/(rad/s)
% Kb = K;
Ra = 4.84; %ohms
T_AZ = K/Ra*(Ea_Az - Kb*dPsi);
T_EL = K/Ra*(Ea_El - Kb*dTheta);
```

Estimator.m

```
function [dPsi,dTheta] = Estimator(Psi,Theta,qa,ra,p,q,r)
dPsi = (ra - sin(Theta)*(cos(Psi)*p+q*sin(Psi)))/cos(Theta)-r;
dTheta = qa - cos(Psi)*q + p*sin(Psi);
```

B.4 INS Error Simulations

B.4.1 Pio.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Pio.m
% author: William Deike
% Created: 20 January 2010 last Modified: 15 March 2010
% Description: m-file runs the Monte Carlo simulation to
% test the pio approximation with the full range of Euler
% angles and pointing vectors.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
AZ = (-180:10:180)*pi/180;
EL = (0:10:80)*pi/180;
Pitch = (-90:10:90)*pi/180;
Roll = (-90:10:90)*pi/180;
A = length(AZ);
B = length(EL);
C = length(Pitch);
D = length(Roll);
%%
AzstdV = zeros(A,B,C,D);
ElstdV = zeros(A,B,C,D);
CorrV = zeros(A,B,C,D);
for a = 1:A
    Az = AZ(a);
    for b = 1:B
        El = EL(b);
        itt = 1;
        Azstd = zeros(C*D,1);
        Elstd = zeros(C*D,1);
        Corr = zeros(C*D,1);
        for c = 1:C
            Beta = Pitch(c);
            for d = 1:D
                Gamma = Roll(d);
                trials = 1000;
                VAR = .001;
                y = normrnd(0,VAR,trials,1);
                p = normrnd(0,VAR,trials,1);
                r = normrnd(0,VAR,trials,1);
                AzEr = zeros(trials,1);
                ElEr = zeros(trials,1);
                for n = 1:trials
                    s3 = y(n);
                    s5 = p(n);
                    s7 = r(n);
                    e1 = -s3*sin(Beta)+s7;
                    e2 = s3*cos(Beta)*sin(Gamma)+s5*cos(Gamma);
                    e3 = s3*cos(Beta)*cos(Gamma)-s5*sin(Gamma);
                    T44 = [1 e3 -e2; -e3 1 e1; e2 -e1 1];
                    ijk = XYZ(Az,El);
                    IJK = (T44*ijk)';
                    [AzEr(n) ElEr(n)] = AzEl(IJK);
                end
                AzstdV(a,b,c,d) = std(AzEr)*180/pi;
                ElstdV(a,b,c,d) = std(ElEr)*180/pi;
                Azstd(itt,1) = std(AzEr)*180/pi;
            end
        end
    end
end
```

```

        Elstd(itt,1) = std(ElEr)*180/pi;
        cc = corrcoef([AzEr ElEr]*180/pi);
        Corr(itt,1) = cc(2,1);
        CorrV(a,b,c,d) = cc(2,1);
        itt = itt + 1;
    end
end
end
end
save('Pio.mat','AzstdV','ElstdV','CorrV')

```

B.4.2 PioAnalysis.m

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PioAnalysis.m
% author: William Deike
% Created: 20 January 2010 last Modified: 15 March 2010
% Description: m-file analyzes the results from the Monte
% Carlo simulation Pio.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
load PioIII.mat
p = 0;
r = 80;
P = int2str(p);
R = int2str(r);
Az = Azmi(:,2:end-1)*180/pi;
El = Elev*180/pi;
x = find(p*pi/180 <= Pitch,1);
y = find(r*pi/180 <= Roll,1);
figure
subplot(311)
contour(Az,El,sqrt(squeeze(AzstdV(2:end-1,:,x,y))))
title('Azimuth Variance (\circ)')
xlabel('Azimuth (\circ)')
ylabel('Elevation (\circ)')
colorbar
subplot(312)
contour(Az,El,sqrt(squeeze(ElstdV(2:end-1,:,x,y))))
title('Elevation Variance (\circ)')
xlabel('Azimuth (\circ)')
ylabel('Elevation (\circ)')
colorbar
subplot(313)
contour(Az,El,squeeze(CorrV(2:end-1,:,x,y)))
title('Correlation Coefficient')
xlabel('Azimuth (\circ)')
ylabel('Elevation (\circ)')
colorbar
%%
a = 5; b = 7; c = 9;
figure
subplot(331)
contour(Az,El,squeeze(AzstdV(2:end-1,:,c,a)))
subplot(332)
contour(Az,El,squeeze(AzstdV(2:end-1,:,c,b)))
subplot(333)

```

```

contour(Az,El,squeeze(AzstdV(2:end-1,:,c,c)))')
subplot(334)
contour(Az,El,squeeze(AzstdV(2:end-1,:,b,a)))')
subplot(335)
contour(Az,El,squeeze(AzstdV(2:end-1,:,b,b)))')
subplot(336)
contour(Az,El,squeeze(AzstdV(2:end-1,:,b,c)))')
subplot(337)
contour(Az,El,squeeze(AzstdV(2:end-1,:,a,a)))')
subplot(338)
contour(Az,El,squeeze(AzstdV(2:end-1,:,a,b)))')
subplot(339)
contour(Az,El,squeeze(AzstdV(2:end-1,:,a,c)))')
figure
subplot(331)
contour(Az,El,squeeze(ElstdV(2:end-1,:,c,a)))')
subplot(332)
contour(Az,El,squeeze(ElstdV(2:end-1,:,c,b)))')
subplot(333)
contour(Az,El,squeeze(ElstdV(2:end-1,:,c,c)))')
subplot(334)
contour(Az,El,squeeze(ElstdV(2:end-1,:,b,a)))')
subplot(335)
contour(Az,El,squeeze(ElstdV(2:end-1,:,b,b)))')
subplot(336)
contour(Az,El,squeeze(ElstdV(2:end-1,:,b,c)))')
subplot(337)
contour(Az,El,squeeze(ElstdV(2:end-1,:,a,a)))')
subplot(338)
contour(Az,El,squeeze(ElstdV(2:end-1,:,a,b)))')
subplot(339)
contour(Az,El,squeeze(ElstdV(2:end-1,:,a,c)))')
figure
subplot(331)
contour(Az,El,squeeze(CorrV(2:end-1,:,c,a)))')
subplot(332)
contour(Az,El,squeeze(CorrV(2:end-1,:,c,b)))')
subplot(333)
contour(Az,El,squeeze(CorrV(2:end-1,:,c,c)))')
subplot(334)
contour(Az,El,squeeze(CorrV(2:end-1,:,b,a)))')
subplot(335)
contour(Az,El,squeeze(CorrV(2:end-1,:,b,b)))')
subplot(336)
contour(Az,El,squeeze(CorrV(2:end-1,:,b,c)))')
subplot(337)
contour(Az,El,squeeze(CorrV(2:end-1,:,a,a)))')
subplot(338)
contour(Az,El,squeeze(CorrV(2:end-1,:,a,b)))')
subplot(339)
contour(Az,El,squeeze(CorrV(2:end-1,:,a,c)))')

```

B.4.3 INSErrorSim.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% INSErrorSim.m
% author: William Deike
% Created: 29 January 2010 last Modified: 15 March 2010
% Description: m-file runs the Simulink model to simulate
% the GPS/INS error.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
close all
Az_NED = -155*pi/180;
El_NED = 42*pi/180;
Yaw = .6;
Pitch = .03;
Roll = .03;
INSvar = .001; %rad
x = cos(Az_NED)*cos(El_NED);
y = sin(Az_NED)*cos(El_NED);
z = -sin(El_NED);
R = [x y z]';
Ry = [cos(Yaw) sin(Yaw) 0; -sin(Yaw) cos(Yaw) 0; 0 0 1];
Rp = [cos(Pitch) 0 -sin(Pitch); 0 1 0; sin(Pitch) 0 cos(Pitch)];
Rr = [1 0 0; 0 cos(Roll) sin(Roll); 0 -sin(Roll) cos(Roll)];
T = Rr*Rp*Ry;
Ra = T*R;
Xa = Ra(1);
Ya = Ra(2);
Za = Ra(3);
Az = atan2(Ya,Xa);
El = atan2(-Za,cos(Az)*Xa+sin(Az)*Ya);
Beta = Pitch;
Gamma = Roll;
trials = 10000;
VAR = INSvar;
y = normrnd(0,2*VAR,trials,1);
p = normrnd(0,VAR,trials,1);
r = normrnd(0,VAR,trials,1);
AzEr = zeros(1,trials);
ElEr = zeros(1,trials);
for n = 1:trials
    s1 = y(n);
    s2 = p(n);
    s3 = r(n);
    e1 = -s1*sin(Pitch)+s3;
    e2 = s1*cos(Pitch)*sin(Roll)+s2*cos(Roll);
    e3 = s1*cos(Pitch)*cos(Roll)-s2*sin(Roll);
    T44 = [1 e3 -e2; -e3 1 e1; e2 -e1 1];
    ijk = XYZ(Az,El);
    IJK = (T44*ijk)';
    [AzEr(n) ElEr(n)] = AzEl(IJK);
end
EstAzStd = std(AzEr.*180/pi);
EstElStd = std(ElEr.*180/pi);
sim('INSError',1600);
Az = logouts.AzEl.Az.Data*180/pi;
El = logouts.AzEl.El.Data*180/pi;
AzEr = logouts.Error.Data(:,1)*180/pi;
ElEr = logouts.Error.Data(:,2)*180/pi;
```

```

sigAz = std(AzEr);
sigEl = std(ElEr);
CC = corrcoef([AzEr ElEr]);
figure('Position', [240 212 800 600]);
subplot(211)
hold on; grid on
plot(Az+AzEr,El+ElEr,'kx')
plot(Az,El,'r.','markersize',20)
axis equal
xlabel('Azimuth (\circ)')
ylabel('Elevation (\circ)')
legend('Trials','True Solution','location','SouthEast')
subplot(212)
X = sort(sqrt((AzEr).^2+(ElEr).^2));
FX = (1:length(AzEr))/length(AzEr);
plot(X,FX,'go')
hold on; grid on
sigAz = EstAzStd;
sigEl = EstElStd;
x = 0:.0001:2;
fx = x./(sigAz*sigEl).*exp(-(x.^2*(sigAz^2+sigEl^2))/(4*sigAz^2*sigEl^2))...
    .*besseli(0,x.^2*(sigAz^2-sigEl^2)/(4*sigAz^2*sigEl^2));
Fx = cumsum(fx)/sum(fx);
plot(x,Fx,'r--','linewidth',2)
axis([0 .5 0 1.05])
xlabel('Total Pointing Error (\circ)')
ylabel('Probability')
legend('Simulated','Theoretical','location','SouthEast')
KS = zeros(length(X),1);
for t = 1:length(X)
    KS(t) = Fx(find(x >= X(t),1));
end
figure('Position', [240 212 800 600]);
hold on; grid on
plot(X,abs(KS-FX'),'linewidth',2)
plot(X,1.63/sqrt(length(X)).*ones(length(X),1),'r--','linewidth',2)
axis([0 .5 0 .015])
xlabel('Total Pointing Error (\circ)')
ylabel('Error')
legend('KS Test','\alpha = .01 Threshold')

```

B.4.4 INS Error Simulink Model

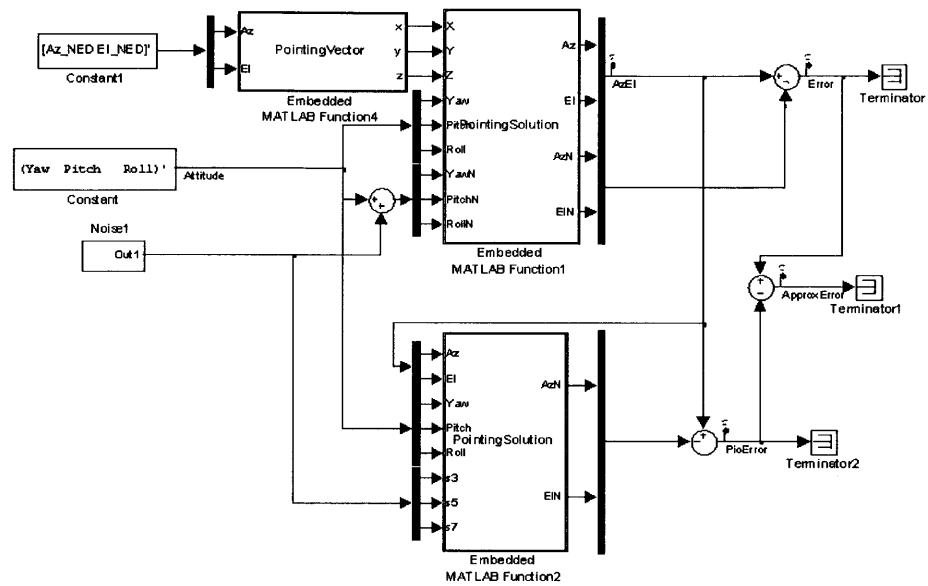


Figure B-9: INSError.mdl Simulink Model

B.5 Open-Loop Pointing Simulation

B.5.1 Simulator.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Simulator.m
% author: William Deike
% Created: 20 March 2010  last Modified: 24 April 2010
% Description: m-file runs the Simulink model to simulate
% the open-loop pointing performance.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sig = [.0025 .001 .0005 .00035 0];
t = 1600; %seconds
color = 'kbgrm';
figure('Position', [240 212 800 600]);
hold on; grid on
figure('Position', [240 212 800 600]);
subplot(211)
hold on; grid on
figure('Position', [240 212 800 600]);
hold on; grid on
axis([0 .6 0 1])
antrad = [.5 .4 .3]; %(m)
load FlightData/RT10Hz2
[theta_deg F_pat1 hpbw1] = AntPatt(antrad(1),44);
[theta_deg F_pat2 hpbw2] = AntPatt(antrad(2),44);
[theta_deg F_pat3 hpbw3] = AntPatt(antrad(3),44);
for x = 1:length(sig)
    INSsig = sig(x);
    sim('OpenLoopPD',t);
    AzEr = logcout.Error.AzEr.data(100:9100).*180/pi;
    ElEr = logcout.Error.ElEr.data(100:9100).*180/pi;
    AOB = logcout.AOB.Data;
    FX = (1:length(AOB))/length(AOB);
    figure(1)
    plot(AzEr,ElEr,[color(x) '-x'],'linewidth',2)
    figure(2)
    subplot(211)
    plot(0:.1:t,AOB,color(x),'linewidth',3)
    figure(3)
    plot(sort(AOB),FX,color(x),'linewidth',3)
end
%%
figure(1)
xlabel('Azimuth (\circ)')
ylabel('Elevation (\circ)')
legend('1\sigma = 2.5 mrad','1\sigma = 1.0 mrad',...
       '1\sigma = 0.5 mrad','1\sigma = 0.35 mrad',...
       'No GPS/INS Error','location','southeast')
axis equal
figure(2)
subplot(211)
xlabel('Time (sec)')
ylabel('Total Pointing Error (\circ)')
figure(3)
```

```

xlabel('Total Pointing Error (\circ)')
ylabel('Cumulative Distribution')
legend('1\sigma = 2.5 mrad','1\sigma = 1.0 mrad',...
       '1\sigma = 0.5 mrad','1\sigma = 0.35 mrad',...
       'No GPS/INS Error','location','southeast')
% plot(hpbw1/2.*ones(2,1),[0 1],'--k','linewidth',3)
% plot(hpbw2/2.*ones(2,1),[0 1],'--b','linewidth',3)
% plot(hpbw3/2.*ones(2,1),[0 1],'--g','linewidth',3)
% text(hpbw1/2,.8,' \leftarrow 0.5 m HPBW/2','FontSize',28)
% text(hpbw2/2,.6,' \leftarrow 0.4 m HPBW/2','FontSize',28)
% text(hpbw3/2,.4,' \leftarrow 0.3 m HPBW/2','FontSize',28)
Az = logsout.CMD.Az.Data.*180/pi;
El = logsout.CMD.El.Data.*180/pi;
figure(2)
subplot(212)
hold on
grid on
plot(0:.1:t,Az,'b','linewidth',3)
plot(0:.1:t,El,'g','linewidth',3)
xlabel('Time (sec)')
ylabel('Cmd Angles (\circ)')
legend('Azimuth','Elevation','location','southeast')

```

B.5.2 Open-Loop Pointing Simulink Model

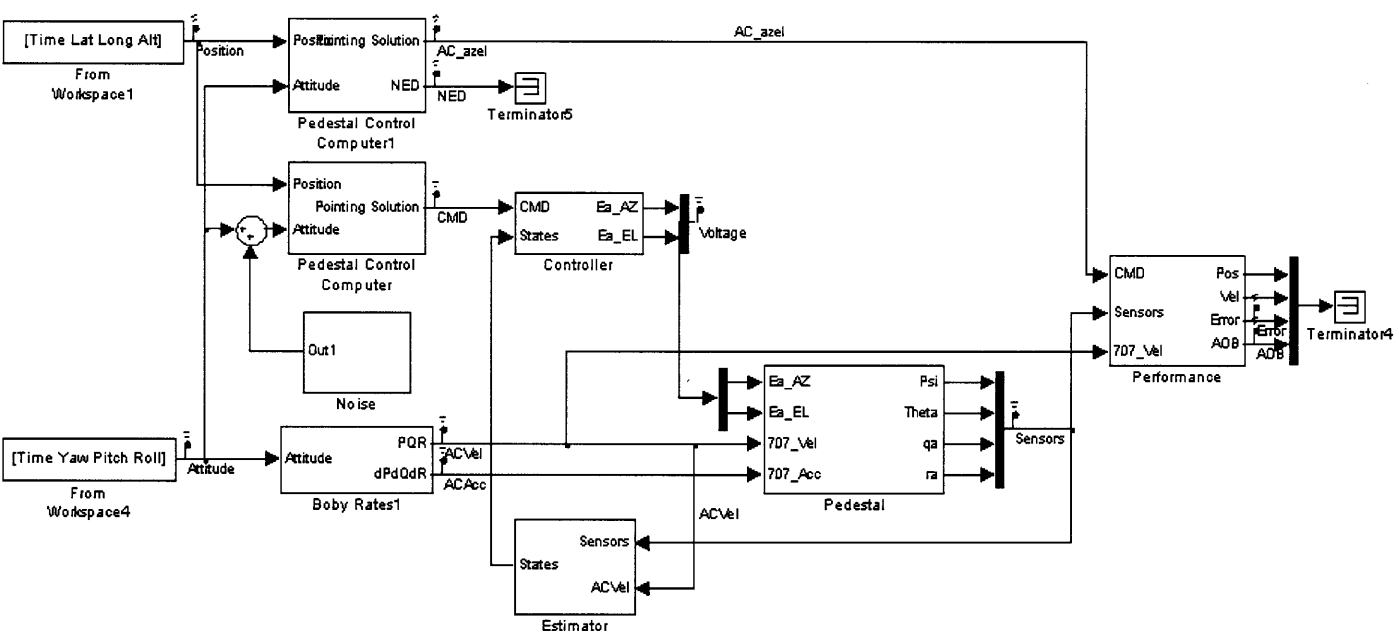


Figure B-10: OpenLoop.mdl Simulink Model

B.6 SNR Characterization Simulations

B.6.1 SimSNR.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SimSNR.m
% author: William Deike
% Created: 20 March 2010  last Modified: 24 April 2010
% Description: m-file runs the Simulink model to simulate
% the signal gain of the Tx and the noise power of Rx.
% These are then combined to get a simulated SNR.  Each
% of these are compared to theoretical calculations
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all; close all;
V = .01; %Sigma^2
R = 2;  %Gain Tilt [0,R]
n = 10; %Number of Samples
%Run Simulink Model
sim('SignalNoise',n*1000);
%Convert output data
S = lognout.Signal.Data(2:end);
N = lognout.Noise.Data(2:end);
NdB = lognout.NdB.Data(2:end);
SNR = lognout.SNR.Data(2:end);
%Sort data to create CDF
S = sort(S);
N = sort(N);
NdB = sort(NdB);
SNR = sort(SNR);
Fx = (1:length(N))/length(N);
%Plot Simulated Data
figure('Position', [240 212 800 600]);
subplot(311)
hold on; grid on
plot(S,Fx,'bo')
ylabel('F_S(x)')
xlabel('x (dBW)')
subplot(312)
hold on; grid on
plot(NdB,Fx,'bo')
ylabel('F_N(x)')
xlabel('x (dBW)')
subplot(313)
hold on; grid on
plot(SNR,Fx,'bo')
ylabel('F_{SNR}(x)')
xlabel('x (dBW)')
%Create Theoretical Distributions
r = .0001;
%Theoretical Signal
X=0:r:R;
p = find(X>=R/2,1);
FX = zeros(1,length(X));
FX(1:p) = 2^(n-1).*((X(1:p))/(R)).^(n);
FX(p:end) = 1-2^(n-1).*(1-(X(p:end))/(R)).^(n);
fX = zeros(1,length(X));
```

```

fX(1:p) = n/R*2^(n-1).*((X(1:p))/(R)).^(n-1);
fX(p:end) = n/R*2^(n-1).*(1-(X(p:end))/(R)).^(n-1);
%Theoretical Noise
Y = -35:r:-10;
FY = chi2cdf(10.^(Y./10).*(n/(V)),n);
fY = (FY(2:end)-FY(1:end-1))/r;
fY = fY(-(-end:-1));
Y1 = Y;
Y = -Y(-(-end:-1));
%Theoretical SNR
fZ = conv(fX,fY)*r;
FZ = cumsum(fZ)*r;
Z = [Y(1:end-1) Y(end)+X(1:end-1)];
%Plot Theoretical
subplot(311)
plot(X,FX,'r','linewidth',4)
axis([.4 1.6 0 1.1])
subplot(312)
plot(Y1,FY,'r','linewidth',4)
axis([-33 -12 0 1.1])
legend('Simulated','Theoretical','location','West')
subplot(313)
plot(Z,FZ,'r','linewidth',4)
axis([15 35 0 1.1])
%%
ST = zeros(length(S),1);
NT = zeros(length(S),1);
SNRT = zeros(length(S),1);
for t = 1:length(S)
    ST(t) = FX(find(X>=S(t),1));
    NT(t) = FY(find(Y1>=NdB(t),1));
    SNRT(t) = FZ(find(Z>=SNR(t),1));
end
figure('Position', [240 212 800 600]);
subplot(311)
hold on; grid on
plot(S,abs(ST-Fx'),'b','linewidth',2)
plot([S(1) S(end)],1.63/sqrt(length(Fx)).*...
ones(2,1),'r--','linewidth',2)
ylabel('Error_S')
xlabel('x (dBW)')
legend('KS Test','\alpha = 0.01 Threshold')
axis([S(1) S(end) 0 .02])
subplot(312)
hold on; grid on
plot(NdB,abs(NT-Fx'),'b','linewidth',2)
plot([NdB(1) NdB(end)],1.63/sqrt(length(Fx)).*...
ones(2,1),'r--','linewidth',2)
ylabel('Error_N')
xlabel('x (dBW)')
axis([NdB(1) NdB(end) 0 .02])
subplot(313)
hold on; grid on
plot(SNR,abs(SNRT-Fx'),'b','linewidth',2)
plot([SNR(1) SNR(end)],1.63/sqrt(length(Fx)).*...
ones(2,1),'r--','linewidth',2)
ylabel('Error_{SNR}')

```

```
xlabel('x (dBW)')
axis([SNR(1) SNR(end) 0 .02])
```

B.6.2 SNR Characterization Simulink Model

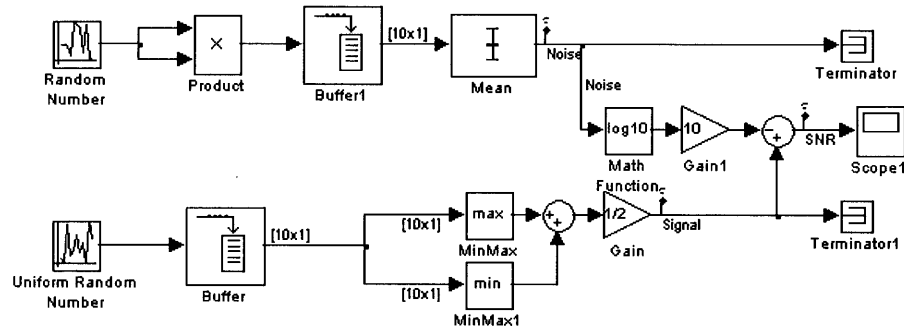


Figure B-11: SignalNoise.mdl Simulink Model

B.6.3 Dist_Plots.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Dist_Plots.m
% author: William Deike
% Created: 15 December 2009 last Modified: 10 March 2010
% Description: m-file runs plots the probability
% distributions for the SNR Characterization.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure('Position', [240 212 800 600]);
subplot(211)
hold on; grid on
X = -1:.001:1;
plot(X,normpdf(X,0,.1),'linewidth',4)
text(.25,2.5,'\sigma is unknown')
ylabel('f(x)')
subplot(212)
hold on; grid on
plot(X,normcdf(X,0,.1),'linewidth',4)
xlabel('x (W)'); ylabel('F(x)')
figure('Position', [240 212 800 600]);
subplot(211)
hold on; grid on
X = -1:.001:1;
plot(X,unifpdf(X,-.2,.7),'linewidth',4)
text(-.75,1.25,' range [c,r+c] is unknown')
ylabel('f(x)')
subplot(212)
```

```

hold on; grid on
plot(X,unifcdf(X,-.2,.7),'linewidth',4)
xlabel('x (dBW)')
ylabel('F(x)')
%%
V = .01;
R = 10;
r = .001;
n = -(-5:1:-1);
c = 'rmgbk';
X=0:r:R;
figure('Position', [240 212 800 600]);
subplot(211)
hold on; grid on
ylabel('f_S(x)')
subplot(212)
hold on; grid on
xlabel('x (dBW)')
ylabel('F_S(x)')
for t = 1:length(n);
    p = find(X>=R/2,1);
    Fx = zeros(1,length(X));
    Fx(1:p) = 2^(n(t)-1).*((X(1:p))/(R)).^(n(t));
    Fx(p:end) = 1-2^(n(t)-1).*(1-(X(p:end))/(R)).^(n(t));
    fx = zeros(1,length(X));
    fx(1:p) = n(t)/R*2^(n(t)-1).*((X(1:p))/(R)).^(n(t)-1);
    fx(p:end) = n(t)/R*2^(n(t)-1).*(1-(X(p:end))/(R)).^(n(t)-1);
    subplot(211)
    plot(X,fx,c(t),'linewidth',4)
    subplot(212)
    plot(X,Fx,c(t),'linewidth',4)
end
subplot(211)
legend('n = 50','n = 40','n = 30','n = 20','n = 10')
%%
X = 0:r/100:.04;
figure('Position', [240 212 800 600]);
subplot(211)
hold on; grid on
ylabel('f_N(x)')
subplot(212)
hold on; grid on
xlabel('x (W)')
ylabel('F_N(x)')
for t = 1:length(n);
    Fx = (chi2cdf(n(t)/(V).*X,n(t)))';
    fx = n(t)/(V).*(chi2pdf(n(t)/(V).*X,n(t)))';
    subplot(211)
    plot(X,fx,c(t),'linewidth',4)
    subplot(212)
    plot(X,Fx,c(t),'linewidth',4)
end
subplot(211)
legend('n = 50','n = 40','n = 30','n = 20','n = 10')
X = -35:r:-10;
Y = 10.^(X./10);
figure('Position', [240 212 800 600]);

```



```

subplot(211)
hold on; grid on
ylabel('f_N(x)')
subplot(212)
hold on; grid on
xlabel('x (dBW)')
ylabel('F_N(x)')
for t = 1:length(n);
    Fx = (chi2cdf(n(t)/(V).*Y,n(t)))';
    fx = (Fx(2:end)-Fx(1:end-1))/r;
    subplot(211)
    plot(X(2:end),fx,c(t),'linewidth',4)
    subplot(212)
    plot(X,Fx,c(t),'linewidth',4)
end
subplot(211)
legend('n = 50','n = 40','n = 30','n = 20','n = 10')
%%
figure('Position', [240 212 800 600]);
subplot(211)
hold on; grid on
ylabel('f_{SNR}(x)')
subplot(212)
hold on; grid on
xlabel('x (dBW)')
ylabel('F_{SNR}(x)')
for t = 1:length(n);
    X=0:r:R;
    p = find(X>=R/2,1);
    fx = zeros(1,length(X));
    fx(1:p) = n(t)/R*2^(n(t)-1).*((X(1:p))/(R)).^(n(t)-1);
    fx(p:end) = n(t)/R*2^(n(t)-1).*(1-(X(p:end))/(R)).^(n(t)-1);
    X1 = X;
    X = -35:r:-10;
    Y = 10.^(X./10);
    Fy = (chi2cdf(n(t)/(V).*Y,n(t)))';
    fy = (Fy(2:end)-Fy(1:end-1))/r;
    fy = fy(-(-end:-1));
    Y1 = X;
    Y1 = -Y1(-(-end:-1));
    fz = conv(fx,fy)*r;
    Fz = cumsum(fz)*r;
    Z = [Y1(1:end-1) Y1(end)+X1(1:end-1)];
    subplot(211)
    plot(Z,fz,c(t),'linewidth',4)
    subplot(212)
    plot(Z,Fz,c(t),'linewidth',4)
end
subplot(211)
legend('n = 50','n = 40','n = 30','n = 20','n = 10')
%%
V = .01;
R = 10;
r = .001;
n = -(-50:10:-10);
c = 'rmgbk';
X=0:r:R;

```

```

STD = zeros(length(n),1);
figure('Position', [240 212 800 600]);
subplot(211)
hold on; grid on
xlabel('x (dBW)')
ylabel('f_{4xSNR}(x)')
subplot(212)
hold on; grid on
xlabel('x (dBW)')
ylabel('F_{4xSNR}(x)')
for t = 1:length(n);
    X=0:r:R;
    p = find(X>=R/2,1);
    fx = zeros(1,length(X));
    fx(1:p) = n(t)/R*2^(n(t)-1).*((X(1:p))/(R)).^(n(t)-1);
    fx(p:end) = n(t)/R*2^(n(t)-1).*(1-(X(p:end))/(R)).^(n(t)-1);
    X1 = X;
    X = -35:r:-10;
    Y = 10.^(X./10);
    Fy = (chi2cdf(n(t)/(V).*Y,n(t)))';
    fy = (Fy(2:end)-Fy(1:end-1))/r;
    fy = fy(-(-end:-1));
    Y1 = X;
    Y1 = -Y1(-(-end:-1));
    fz = conv(fx,fy)*r;
    Fz = cumsum(fz)*r;
    Z = [Y1(1:end-1) Y1(end)+X1(1:end-1)];
    fzz = conv(fz,fz)*r;
    ZZ = [Z(1)+Z(1:end-1) Z(end)+Z(1:end)];
    fz4 = conv(fzz,fzz(-(-end:-1)))*r;
    Z4 = [ZZ(1)-ZZ(-(-end:-1)) ZZ(1:end-1)-ZZ(1)];
    Fz4 = cumsum(fz4)*r;
    STD(t) = -Z4(find(Fz4 >= .02,1))/2;
    subplot(211)
    plot(Z4,fz4,c(t),'linewidth',4)
    subplot(212)
    plot(Z4,Fz4,c(t),'linewidth',4)
end
subplot(211)
axis([-20 20 0 .25])
legend('n = 50','n = 40','n = 30','n = 20','n = 10')
subplot(212)
axis([-20 20 0 1])
%%
r = .5;
fc = 20;
[theta gain hpbw] = AntPatt(r,fc);
[deg dBW] = LookUpTable(r,fc,0);
figure('Position', [240 212 800 600]);
hold on
grid on
xlabel('Angle Off Boresight (HPBW)')
ylabel('Cumulative Distribution')
t = floor(length(deg)/2);
Xaob = deg(t:end);
X = -dBW(t:end);
for t = 1:length(n);

```

```

        FX = raylcdf(X,STD(t));
        plot(Xaob/hpbw,FX,c(t),'linewidth',4)
    end
    legend('n = 50','n = 40','n = 30','n = 20','n = 10')
    axis([0 .6 0 1])

```

B.6.4 SNR_Analysis.m

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SNR_Analysis.m
% author: William Deike
% Created: 10 January 2010 last Modified: 12 March 2010
% Description: m-file runs the analyzes the SNR model.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc; close all; clear all
figure('Position', [240 212 800 600]);
subplot(211)
hold on; grid on
n = -(-50:10:-10);
c = 'rmgbk';
r = 2:100;
for u = 1:length(n)
    P = 2*(.5-2^(n(u)-1).*((r./2-1)./r).^n(u));
    plot(r,P,c(u),'linewidth',4)
end
xlabel('Gain Tilt - r (dBW)')
ylabel('Probability')
legend('n = 50','n = 40','n = 30','n = 20','n = 10',...
'location','Southwest')
subplot(212)
hold on; grid on
r = 2:.1:10;
for u = 1:length(n)
    P = 2*(.5-2^(n(u)-1).*((r./2-1)./r).^n(u));
    plot(r,P,c(u),'linewidth',4)
end
xlabel('Gain Tilt - r (dBW)')
ylabel('Probability')
%%
figure('Position', [240 212 800 600]);
hold on; grid on
for t = 1:length(n)
    np = -100:1:-10;
    P = zeros(1,length(np));
    for u = 1:length(np)
        V = 10^(np(u)/10);
        X = 2*np(u):.001:np(u)/2;
        Y = 10.^(X./10);
        Fx = (chi2cdf(n(t)/(V).*Y,n(t)))';
        P(u) = Fx(find(X>=np(u)+1,1))-Fx(find(X>=np(u)-1,1));
    end
    plot(np,P,c(t),'linewidth',4)
end
xlabel('Noise Power (dBW)')
ylabel('Probability')
axis([-100 -10 0 1])

```

```
legend('n = 50','n = 30','n = 30','n = 20','n = 10',...
'location','Southeast')
```

B.6.5 SNR_AnalysisII.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SNR_AnalysisII.m
% author: William Deike
% Created: 10 January 2010 last Modified: 12 March 2010
% Description: m-file runs the analyzes the SNR model.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
V = .001;
R = 10;
h = 2:100;
SNR1 = zeros(1,length(h));
SNR15 = zeros(1,length(h));
SNR2 = zeros(1,length(h));
SNR25 = zeros(1,length(h));
SNR3 = zeros(1,length(h));
SdB = R/2;
NdB = 10*log10(V);
SNRdB = SdB-NdB;
r = .001;
itt = 0;
c = 'kbgmr';
figure('Position', [240 212 800 600]);
subplot(311)
hold on
grid on
xlabel('x (dBW)')
ylabel('F_S(x)')
subplot(312)
hold on
grid on
xlabel('x (dBW)')
ylabel('F_N(x)')
subplot(313)
hold on
grid on
xlabel('x (dBW)')
ylabel('F_{SNR}(x)')
for t = 1:length(h)
    Y = (NdB*2):r:(NdB-.5*NdB);
    FY = gammainc(10.^(Y./10).*(h(t)/(V*2)),h(t)/2,'lower');
    fY = (FY(2:end)-FY(1:end-1))/r;
    fY = fY(-(-end:-1));
    Y1 = Y;
    Y = -Y(-(-end:-1));
    X1=0:r:R;
    p = find(X1>=R/2,1);
    F1x = zeros(1,length(X1));
    F1x(1:p) = 2^(h(t)-1).*((X1(1:p))/(R)).^(h(t));
    F1x(p:end) = 1-2^(h(t)-1).*(1-(X1(p:end))/(R)).^(h(t));
    fX = (F1x(2:end)-F1x(1:end-1))/r;
    fZ = conv(fX,fY)*r;
    FZ = cumsum(fZ)*r;
```

```

    fz = (FZ(2:end)-FZ(1:end-1))/r;
    Z = [Y(1:end-1) Y(end)+X1(1:end-2)];
    SNR1(t) = FZ(find(Z>=SNRdB+1,1))-FZ(find(Z>=SNRdB-1,1));
    SNR15(t) = FZ(find(Z>=SNRdB+1.5,1))-FZ(find(Z>=SNRdB-1.5,1));
    SNR2(t) = FZ(find(Z>=SNRdB+2,1))-FZ(find(Z>=SNRdB-2,1));
    SNR25(t) = FZ(find(Z>=SNRdB+2.5,1))-FZ(find(Z>=SNRdB-2.5,1));
    SNR3(t) = FZ(find(Z>=SNRdB+3,1))-FZ(find(Z>=SNRdB-3,1));
if rem(h(t),20)==0
    itt = itt + 1;
    subplot(311)
    plot(X1,F1x,c(itt),'linewidth',4)
    subplot(312)
    plot(Y1,FY,c(itt),'linewidth',4)
    subplot(313)
    plot(Z,FZ,c(itt),'linewidth',4)
else
end
end
subplot(311)
axis([4 6 0 1])
subplot(312)
legend('n = 20','n = 40','n = 60','n = 80','n = 100')
axis([-40 -20 0 1])
subplot(313)
axis([30 40 0 1])
%%
figure('Position', [240 212 800 600]);
subplot(311)
hold on; grid on
plot(h,SNR1,'k','linewidth',4)
plot(h,SNR15,'b','linewidth',4)
plot(h,SNR2,'g','linewidth',4)
plot(h,SNR25,'m','linewidth',4)
plot(h,SNR3,'r','linewidth',4)
xlabel('Number of Hops')
ylabel('SNR Prob')
subplot(312)
hold on; grid on
n = 2:100;
P1 = zeros(1,length(n));
P15 = zeros(1,length(n));
P2 = zeros(1,length(n));
P25 = zeros(1,length(n));
P3 = zeros(1,length(n));
for t = 1:length(n)
    np = -10;
    V = 10^(np/10);
    X = -20:.001:-5;
    Y = 10.^(X./10);
    Fx = (chi2cdf(n(t)/(V).*Y,n(t)))';
    P1(t) = Fx(find(X>=np+1,1))-Fx(find(X>=np-1,1));
    P15(t) = Fx(find(X>=np+1.5,1))-Fx(find(X>=np-1.5,1));
    P2(t) = Fx(find(X>=np+2,1))-Fx(find(X>=np-2,1));
    P25(t) = Fx(find(X>=np+2.5,1))-Fx(find(X>=np-2.5,1));
    P3(t) = Fx(find(X>=np+3,1))-Fx(find(X>=np-3,1));
end
plot(n,P1,'k','linewidth',4)

```

```

plot(n,P15,'b','linewidth',4)
plot(n,P2,'g','linewidth',4)
plot(n,P25,'m','linewidth',4)
plot(n,P3,'r','linewidth',4)
legend('\pm 1 dBW','\pm 1.5 dBW','\pm 2 dBW',...
'\pm 2.5 dBW','\pm 3 dBW','location','SouthEast')
xlabel('Number of Hops')
ylabel('Noise Prob')
subplot(313)
hold on; grid on
plot(n,P1-SNR1,'k','linewidth',4)
plot(n,P15-SNR15,'b','linewidth',4)
plot(n,P2-SNR2,'g','linewidth',4)
plot(n,P25-SNR25,'m','linewidth',4)
plot(n,P3-SNR3,'r','linewidth',4)
xlabel('Number of Hops')
ylabel('Difference')

```

B.7 Closed Loop Pointing Simulation

B.7.1 AntPatt.m

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% AntPatt.m
% author: William Deike
% Created: 12 February 2010 last Modified: 22 March 2010
% Description: matlab function that calculates a
% theoretical one-dimensional antenna pattern based on the
% aperture size and transmission frequency.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [theta_deg F_pat hpbw] = AntPatt(a,Fc)
% a is radius of aperture in meters
% bigger a = smaller beamwidth
theta_deg = -5:0.0001:5;
fc = Fc*10^9; % center frequency in Hz
lambda = 3e8/fc; % wavelength
b = 2*pi/lambda; % wave number
theta_rad = deg2rad(theta_deg); % trig using radians
f_pat = 2*besselj(1,b*a*sin(theta_rad))./...
(b*a*sin(theta_rad)); % antenna pattern
F_pat = 20*log10(abs(f_pat)); % antenna pattern in dB
F_pat(ceil(length(theta_deg)/2)) = 0; % replace singularity at 0 deg
hpbw = rad2deg(2*1.6*lambda/(pi*2*a)); % half-power beamwidth (deg)

```

B.7.2 LookUpTable.m

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% LookUpTable.m
% author: William Deike
% Created: 15 March 2010 last Modified: 28 April 2010
% Description: matlab function that calculates a

```

```

% the step-tracking difference antenna pattern to estimate
% the AOB based on dithering.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [deg dBW]=LookUpTable(r,fc,p)
[theta_deg F_pat hpbw] = AntPatt(r,fc);
D = hpbw*.47;
dither = D/sqrt(2);
a = (find(theta_deg-dither >= -4,1));
b = (find(theta_deg+dither >= -4,1));
c = (find(theta_deg-dither >= 4,1));
d = (find(theta_deg+dither >= 4,1));
SumF_pat = F_pat(a:c)+F_pat(b:d);
SumTheta = theta_deg(find(theta_deg >= -4,1):find(theta_deg >= 4,1));
if p >=1
    figure('Position', [240 212 800 600]);
    subplot(211)
    hold on
    grid on
    plot(theta_deg(a:c)-dither,F_pat(a:c),'b','linewidth',4)
    plot(theta_deg(b:d)+dither,F_pat(b:d),'g','linewidth',4)
    plot(SumTheta,SumF_pat,'r','linewidth',4)
    axis([-1 1 -50 0])
    xlabel('Angle Off Boresight (\circ)')
    ylabel('dBW')
else
end
a = (find(SumTheta-dither >= -3,1));
b = (find(SumTheta+dither >= -3,1));
c = (find(SumTheta-dither >= 3,1));
d = (find(SumTheta+dither >= 3,1));
DiffF_pat = SumF_pat(a:c)-SumF_pat(b:d);
DiffTheta = SumTheta(find(SumTheta >= -3,1):find(SumTheta >= 3,1));
if p >=1
    subplot(212)
    hold on; grid on
    plot(SumTheta(a:c)-dither,SumF_pat(a:c),'b','linewidth',4)
    plot(SumTheta(b:d)+dither,SumF_pat(b:d),'g','linewidth',4)
    plot(DiffTheta,DiffF_pat,'r','linewidth',4)
    axis([-1 1 -50 50])
    xlabel('Angle Off Boresight (\circ)')
    ylabel('dBW')
else
end
a = find(DiffTheta>=-hpbw/2,1);
b = find(DiffTheta>=+hpbw/2,1);
deg = DiffTheta(a:b);
dBW = DiffF_pat(a:b);

```

B.7.3 FullDitherSquintSim.m

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FullDitherSquintSim.m
% author: William Deike
% Created: 20 April 2010 last Modified: 27 April 2010
% Description: m-file runs the Simulink model to simulate

```



```

% the step-tracking algorithm after running Simulator.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sig = [.0025 .001 .0005 .00035 0];
t = 1000; %seconds
INSsig = sig(1);
filename = 'FlightData/RTB10Hz';
timeStart = 100;
timeFinish = 9100;
load(filename)
sim('OpenLoopPD',t);
AzEr = logout.Error.AzEr.data(100:9100).*180/pi;
ElEr = logout.Error.ElEr.data(100:9100).*180/pi;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
antrad = [.5 .4 .3]; % (m)
r = antrad(3);
n = 50; %Hops
fc = 20;
[theta_deg F_pat hpbw] = AntPatt(r,fc);
[deg dBW] = LookUpTable(r,fc,0);
D = hpbw*.47;
d = D/sqrt(2);
Az = hpbw*0;
El = hpbw*0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure('Position', [240 212 800 600]);
subplot(221)
hold on; grid on
subplot(222)
hold on; grid on
axis equal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
V = .01; %Sigma^2
R = 3; %Gain Tilt [0,R]
time = floor(length(AzEr)/10);
Time = (0:.1:time)';
AOB1 = sqrt((Az+d+AzEr).^2+(El+d+ElEr).^2);
AOB2 = sqrt((Az-d+AzEr).^2+(El+d+ElEr).^2);
AOB2 = [AOB2(500:end); AOB2(1:499)];
AOB3 = sqrt((Az-d+AzEr).^2+(El-d+ElEr).^2);
AOB3 = [AOB3(1000:end); AOB3(1:999)];
AOB4 = sqrt((Az+d+AzEr).^2+(El-d+ElEr).^2);
AOB4 = [AOB4(1500:end); AOB4(1:1499)];
AOB = [AOB1 AOB2 AOB3 AOB4];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sim('DitherSquint',time); clc
SNR1 = logout.SNR1.Data(2:end);
SNR2 = logout.SNR2.Data(2:end);
SNR3 = logout.SNR3.Data(2:end);
SNR4 = logout.SNR4.Data(2:end);
delAz = logout.delAz.Data(2:end);
delEl = logout.delEl.Data(2:end);
FX = (1:length(SNR1))./length(SNR1);
subplot(221)
hold on; grid on
plot(sort(SNR1),FX,'b','linewidth',4)
plot(sort(SNR2),FX,'g--','linewidth',4)

```

```

plot(sort(SNR3),FX,'k-','linewidth',4)
plot(sort(SNR4),FX,'r.','linewidth',4)
legend('I','II','III','IV')
legend('location','northwest')
legend('boxoff')
xlabel('SNR (dBW)')
ylabel('Probability')
subplot(223)
hold on; grid on
plot(sort(delEl),FX,'k','linewidth',4)
plot(sort(delAz),FX,'m--','linewidth',4)
legend('El','Az')
legend('location','northwest')
legend('boxoff')
xlabel('\Delta (dBW)')
ylabel('Probability')
title([num2str(r) ' m Aperture'])
AzCalc = zeros(length(delAz),1);
ElCalc = zeros(length(delAz),1);
for i = 1:length(delAz)
    if isempty(deg(find(delAz(i) >= dBW,1)))
        if delAz(i) > 0
            AzCalc(i) = min(deg);
        else
            AzCalc(i) = max(deg);
        end
    else
        AzCalc(i) = deg(find(delAz(i) >= dBW,1));
    end
    if isempty(deg(find(delEl(i) >= dBW,1)))
        if delEl(i) > 0
            ElCalc(i) = min(deg);
        else
            ElCalc(i) = max(deg);
        end
    else
        ElCalc(i) = deg(find(delEl(i) >= dBW,1));
    end
end
FinalAz = Az-AzCalc;
FinalEl = El-ElCalc;
FinalError = sort(sqrt(FinalAz.^2+FinalEl.^2));
FX = (1:length(delAz))./length(delAz);
ErrorHops = FinalError(find(FX>=.98,1));
subplot(222)
plot(AzEr+Az,ElEr+El,'b.')
plot(Az-AzCalc,El-ElCalc,'+g')
[x y] = cylinder(hpbw/2,360);
plot(Az+x(1,:),El+y(1:,:), 'k--','linewidth',2)
plot(Az+d,El+d,'xb','markersize',20,'linewidth',4)
plot(Az-d,El+d,'xg','markersize',20,'linewidth',4)
plot(Az-d,El-d,'xk','markersize',20,'linewidth',4)
plot(Az+d,El-d,'xr','markersize',20,'linewidth',4)
plot(Az,El,'dk','markersize',15,'linewidth',5)
plot(0,0,'m.','markersize',30,'linewidth',4)
xlabel('Az (\circ)')
ylabel('El (\circ)')
title(['Number of Hops = ' num2str(n)])

```

```

subplot(224)
hold on; grid on
plot(FinalError/hpbw,FX,'b','linewidth',4)
xlabel('AOB (HPBW)')
ylabel('Probability')
axis([0 .5 0 1])
title(['Rx @ ' num2str(fc) ' GHz' ])
subplot(221)
title(['INS 1\sigma = ' num2str(INSsig*1000) ' mrad'])

```

B.7.4 Closed-Loop Tracking Simulink Model

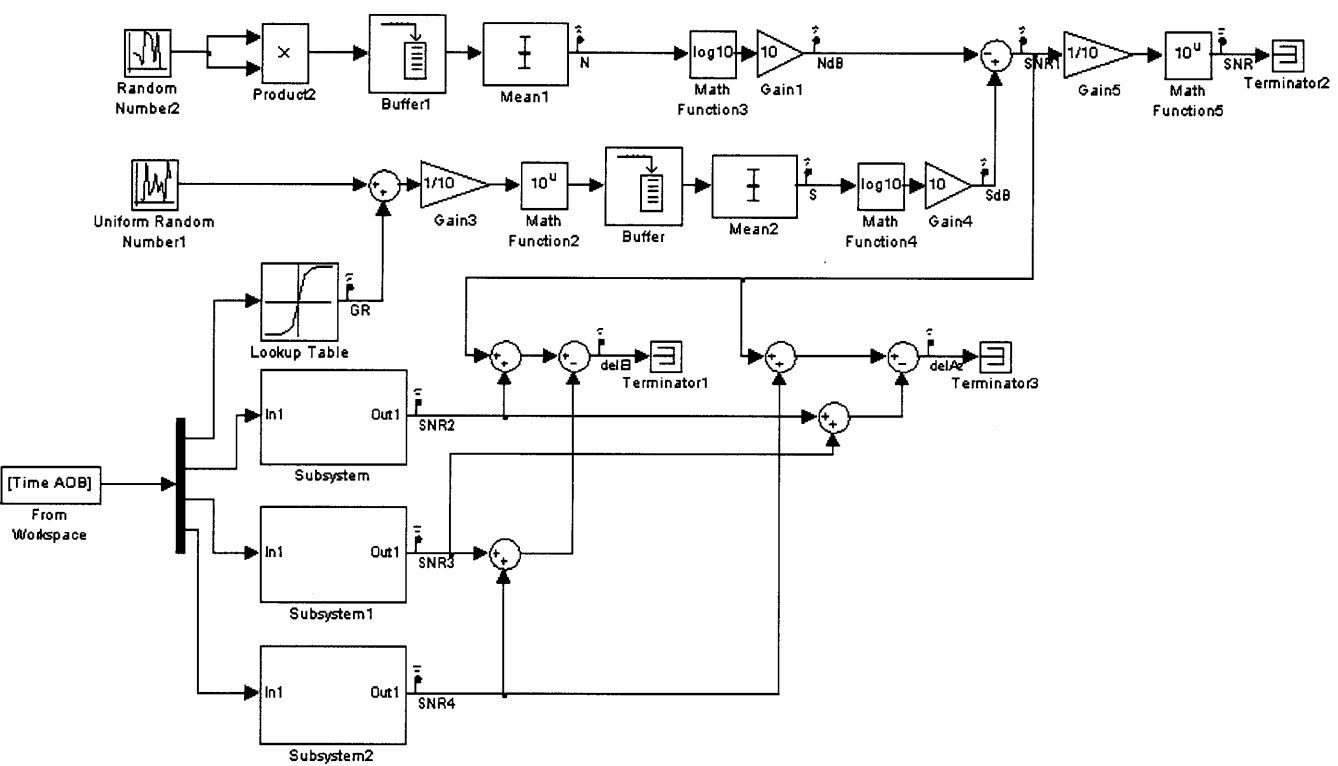


Figure B-12: DitherSquint.mdl Simulink Model

B.7.5 DitherSquintSim.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% DitherSquintSimITT.m
% author: William Deike
% Created: 20 April 2010 last Modified: 27 April 2010
% Description: m-file runs the Simulink model to simulate
% the step-tracking algorithm several times.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
antrad = [.5 .4 .3]; % (m)
r = antrad(3);
N = -(-50:10:-10); %Hops
fc = 20;
[theta_deg F_pat hpbw] = AntPatt(r,fc);
[deg dBW] = LookUpTable(r,fc,0);
D = hpbw*.47;
d = D/sqrt(2);
Az = hpbw*.5;
El = hpbw*0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure('Position', [240 212 800 600]);
hold on; grid on
color = 'rmgbk';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
V = .01; %Sigma^2
R = 3; %Gain Tilt [0,R]
for t = 1:length(N)
    n = N(t);
    time = n*1000;
    AZ = Az.*ones(time*10+1,1);
    EL = El.*ones(time*10+1,1);
    Time = (0:.1:time)';
    AOB1 = sqrt((AZ+d).^2+(EL+d).^2);
    AOB2 = sqrt((AZ-d).^2+(EL+d).^2);
    AOB3 = sqrt((AZ-d).^2+(EL-d).^2);
    AOB4 = sqrt((AZ+d).^2+(EL-d).^2);
    AOB = [AOB1 AOB2 AOB3 AOB4];
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    sim('DitherSquint',time); clc
    delAz = logouts.delAz.Data(2:end);
    delEl = logouts.delEl.Data(2:end);
    AzCalc = zeros(length(delAz),1);
    ElCalc = zeros(length(delAz),1);
    for i = 1:length(delAz)
        if isempty(deg(find(delAz(i) >= dBW,1)))
            if delAz(i) > 0
                AzCalc(i) = min(deg);
            else
                AzCalc(i) = max(deg);
            end
        else
            AzCalc(i) = deg(find(delAz(i) >= dBW,1));
        end
        if isempty(deg(find(delEl(i) >= dBW,1)))
            if delEl(i) > 0
                ElCalc(i) = min(deg);
            else
                ElCalc(i) = max(deg);
            end
        end
    end
end
```

```

        end
    else
        ElCalc(i) = deg(find(delEl(i) >= dBW,1));
    end
end
FinalAz = Az-AzCalc;
FinalEl = El-ElCalc;
FinalError = sort(sqrt(FinalAz.^2+FinalEl.^2));
FX = (1:length(delAz))./length(delAz);
ErrorHops = FinalError(find(FX>=.98,1));
plot(FinalError/hpbw,FX,color(t),'linewidth',4)
end
xlabel('Angle Off Boresight (HPBW)')
ylabel('Cumulative Distribution')
axis([0 .5 0 1])
legend('n = 50','n = 40','n = 30','n = 20','n = 10')

```

Appendix C

List of Acronyms and Symbols

Table C.1: List of acronyms used in this work

Abbreviation	Description
AOB	Angle Off Boresight
APS	Antenna Positioner System
CDF	Cumulative Distribution Function
CMC	Cleveland Motion Controls
EHF	Extremely High Frequency
EIRP	Effective Isotropic Radiated Power
FFT	Fast Fourier Transform
GEO	Geosynchronous Earth Orbit
GPS	Global Positioning System
HMMWV	High Mobility Multipurpose Wheeled Vehicle
HPBW	Half-Power Beamwidth
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
KS	Kolmogorov-Smirnov
LEO	Low Earth Orbit
MEO	Medium Earth Orbit
MILSATCOM	Military Satellite Communications
MILSTAR	Military Strategic and Tactical Relay
MLE	Maximum Likelihood Estimator
NED	North, East, Down
PD	Proportional Differential
PDF	Probability Distribution Function
PSD	Power Spectral Density
RIP	Received Isotropic Power
SATCOM	Satellite Communication
SNR	Signal-to-Noise Ratio

Bibliography

- [1] Hampton, P., “Naval Space Command’s top functional requirements for commercial SATCOM,” *Military Communications Conference Proceedings, 1999. MILCOM 1999. IEEE*, Vol. 32, November 1999.
- [2] Nicol, S., Walton, G., Westbrook, L., and Wynn, D., “Future Satellite Communications to Military Aircraft,” *Electronics and Communication Engineering Journal*, Vol. 12, February 2000.
- [3] Command, A. F. S., *MILSTAR Satellite Communications System*, March 2009, <http://www.af.mil/information/factsheets/factsheet.asp?id=118>.
- [4] DATSD, *Nuclear Surety*, May 2010, <http://www.acq.osd.mil/ncbdp/nm/nuclearweaponssurety.html>.
- [5] Cummings, W. C., Jain, P. C., and Ricardi, L. J., “Fundamental Performance Characteristics That Influent EHF MILSATCOM Systems,” *IEEE Transaction on Communications*, Vol. COM-27, No. 10, October 1979.
- [6] Hilkert, J. M., “Inertially Stabilized Platform Technology,” *IEEE Control Systems Magazine*, February 2008.
- [7] Hilkert, J. M., “Kinematic Algorithms for Line-of-Sight Pointing and Scanning using INS/GPS Position and Velocity Information,” *SPIE Proceedings*, Vol. 5810, March 2005.
- [8] Stallings, W., *Wireless Communications And Networks*, Pearson Prentice Hall, 2nd ed., 2005.
- [9] Debruin, J., “Control Systems for Mobile SATCOM Antennas,” *IEEE Control Systems Magazine*, February 2008.
- [10] Masten, M. K., “Platforms for Optical Imaging Systems,” *IEEE Control Systems Magazine*, February 2008.
- [11] Controls, C. M., *Torquemaster Brush Servo-Motors 2100 Series*, 2002, www.cmcontrols.com/downloads/servo_motors/platform2100.pdf.

- [12] Schodorf, J. B., "EHF Satellite Communications On the Move: Experimental Results," Tech. rep., Massachusetts Institute of Technology Lincoln Laboratory, August 2003.
- [13] KVH Industries, I., *KVH DSP-3000 Fiber Optic Gyro Technical Manual*, 2007.
- [14] King, A. D., "Inertial Navigation-Past, Present, and Future," *IEEE Colloquium on Airborne Navigation Systems Workshop*, Vol. 169, No. 3, February 1997, pp. 1–9.
- [15] Weston, J. L. and Titterton, D. H., "Modern Inertial Navigation Technology and its Application," *Electronics and Communication Engineering Journal*, April 2000.
- [16] Yechout, T. R., *Introduction to Aircraft Flight Mechanics*, AIAA, 2003.
- [17] Hofmann-Wellenhof, B., Lightenegger, H., and Collins, J., *Global Positioning System Theory And Practice*, Springer Wien New York, 5th ed., 2001.
- [18] Herring, T., *Lecture Notes for 12.540: Principles of the Global Positioning System*, Massachusetts Institute of Technology, 2009.
- [19] Schmidt, G. T., "INS/GPS Technology," *Emerging Military Capabilities Enabled by Advances in Navigation Sensors*, Vol. 6815, October 2002.
- [20] Qi, H. and Moore, J. B., "Direct Kalman Filtering Approach for GPS/INS Integration," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 38, No. 2, April 2002, pp. 687–693.
- [21] Knight, D. T., "Tightly-Coupled GPS/INS Systems," *IEEE AES Systems Magazine*, February 1997.
- [22] Wei, W., Zongyu, L., and Rongrong, X., "An Improved Tightly Coupled Approach for GPS/INS Integration," *IEEE Proceedings Conference on Robotics, Automation and Mechatronics*, December 2004.
- [23] Meidel, M. K. W., *GPS signal jamming mitigation through multiple model adaptive estimation applied to ultra-tightly coupled GPS/INS architecture.*, Master's thesis, Air Force Institute of Technology, 2005.
- [24] Griffin, M. D. and French, J. R., *Space Vehicle Design*, AIAA, 2nd ed., 2004.
- [25] Sellers, J. J., *Understanding Space*, McGraw-Hill, 2nd ed., 2000.
- [26] Vallado, D. A., *Fundamentals of Astrodynamics and Applications*, Microcosm Press, 3rd ed., 2007.
- [27] Chao, C.-C., *Applied Orbit Perturbation and Maintenance*, AIAA, Inc., 1st ed., 2005.

- [28] Dodgson, D. T., "Communications-On-the-Move, Mobile Terminal Design," *Military Satellite Communications, IEEE Colloquium*, March 1995.
- [29] Karabinis, P., Egri, R., and Bennet, C., "Antenna Pointing and Scanning Control for a Two Axis Gimbal System in the Presence of Platform Motion," *Military Communications Conference*, Vol. 3, October 1988.
- [30] Fisk, J. W., "Confidence Limits for the Pointing Error of Gimbale Sensors," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-2, No. 6, November 1966.
- [31] Tan, A. E.-C., Chia, M. Y.-W., and Rambabu, K., "Design of Ultra-Wideband Monopulse Receiver," *IEEE Transaction on Microwave Theory and Techniques*, Vol. 54, No. 11, November 2006.
- [32] Skolnik, M. I., *Introduction to Radar Systems*, McGraw-Hill, 3rd ed., 2001.
- [33] Gawronski, W. and Craparo, E. M., "Antenna Scanning Techniques for Estimation of Spacecraft Position," *IEEE Antenna's and Propagation Magazine*, Vol. 44, No. 6, December 2002.
- [34] Ekstrand, B., "Equations of Motion for a Two-Axes Gimbal System," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 37, No. 3, July 2001.
- [35] Ogata, K., *System Dynamics*, Pearson Prentice Hall, 4th ed., 2004.
- [36] Google, *Mid-Atlantic Coast*, 2009, <http://earth.google.com/>.
- [37] Hwang, W. G., "Bandwidth on Demand for Deployed-IP Users," *IT Professional*, Vol. 7, No. 1, January 2005.
- [38] MathWorks, "Using FFT to Obtain Simple Spectral Analysis Plots," *Tech Notes*, Vol. 1702, 2010, <http://www.af.mil/information/factsheets/factsheet.asp?id=118>.
- [39] How, J. P., *Lecture Notes for 16.31: Feedback Control*, Massachusetts Institute of Technology, 2001.
- [40] Slotine, J.-J. E. and Li, W., *Applied Nonlinear Control*, Prentice Hall, 1991.
- [41] Pio, R. L., "Error Analysis of Euler Angle Transformations," *AIAA Journal*, Vol. 11, No. 6, June 1973.
- [42] Schodorf, J. B., "A Probabilistic Mispoining Analysis for Land Mobile Satellite Communications Systems with Directive Antennas," *IEEE 54th Vehicular Technology Conference*, Vol. 1, IEEE, October 2001, pp. 296–300.
- [43] Dudewicz, E. J. and Mishra, S. N., *Modern Mathematical Statistics*, John Wiley and Sons, 1st ed., 1998.

- [44] Elbert, B. R., *Introduction to Satellite Communication*, Artech House, 3rd ed., 2008.
- [45] "Specific attenuation model for rain for use in prediction methods," Rec. P.838-2, International Telecommunication Union, Nov. 2004.
- [46] Kokoska, S. and Zwillinger, D., *CRC Standard Probability and Statistics Tables and Formulae*, Chapman and Hall/CRC, student ed., 2000.
- [47] Berger, H., "On the Optimum Squint Angles of Amplitude Monopulse Radar and Beacon Tracking Systems," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-8, No. 4, November 1972.
- [48] Figucia, R. J., "Downlink Acquisition and Tracking Procedures for the ASCAMP Satellite Communications Terminal," Tech. rep., Massachusetts Institute of Technology Lincoln Laboratory, September 1993.
- [49] Marsh, E. A., *Inertially Stabilized Platforms for SATCOM On-The-Move Applications: A Hybrid Open/Closed-Loop Antenna Pointing Strategy.*, Master's thesis, Massachusetts Institute of Technology, 2008.