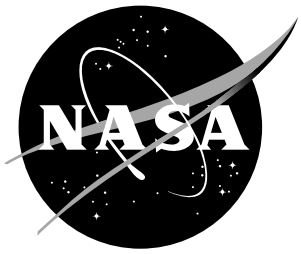
NASA/CR-2003-211937



Preliminary Design and Analysis of the GIFTS Instrument Pointing System

Paul P. Zomkowski
Joint Institute for Advancement of Flight Sciences
The George Washington University
Langley Research Center, Hampton, Virginia

March 2003

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results ... even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at ***http://www.sti.nasa.gov***

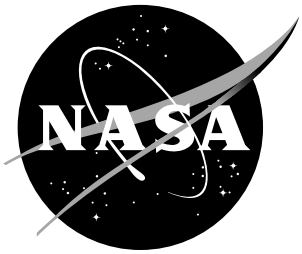
- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA STI Help Desk at (301) 621-0134

- Phone the NASA STI Help Desk at (301) 621-0390

- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076-1320

NASA/CR-2003-211937



Preliminary Design and Analysis of the GIFTS Instrument Pointing System

Paul P. Zomkowski
Joint Institute for Advancement of Flight Sciences
The George Washington University
Langley Research Center, Hampton, Virginia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

Prepared for Langley Research Center
under Contract NCC1-01017

March 2003

Acknowledgments

The author would like to thank the following individuals for making this research possible:
• Dr. Lucas Horta from the NASA Langley Research Center for his guidance and support.
• Dr. Paul Cooper from The George Washington University / Joint Institute for Advancement of Flight Sciences for his counsel and patience.
• Dr. Peiman Maghami from the NASA Goddard Space Flight Center for his guidance and expertise in the areas of spacecraft dynamics modeling and Simulink development.
• Mrs. Kim Kudlinski from the NASA Langley Research Center for support with the Pointing Systems Working Group of the GIFTS Project.
• Benjamin George from The George Washington University / Joint Institute for Advancement of Flight Sciences for his support during model development.

Available from:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 605-6000

Abstract

The Geosynchronous Imaging Fourier Transform Spectrometer (GIFTS) Instrument is the next generation spectrometer for remote sensing weather satellites. The GIFTS instrument will be used to perform scans of the Earth's atmosphere by assembling a series of field-of-views (FOV) into a larger pattern. Realization of this process is achieved by step scanning the instrument FOV in a contiguous fashion across any desired portion of the visible Earth. A 2.3 arc second pointing stability, with respect to the scanning instrument, must be maintained for the duration of the FOV scan. A star tracker producing attitude data at 100 Hz rate will be used by the autonomous pointing algorithm to precisely track target FOV's on the surface of the Earth.

The main objective is to validate the pointing algorithm in the presence of spacecraft disturbances and determine acceptable disturbance limits from expected noise sources. Proof of concept validation of the pointing system algorithm is carried out with a full system simulation developed using Matlab Simulink. Models for the following components function within the full system simulation: inertial reference unit (IRU), attitude control system (ACS), reaction wheels, star tracker, and mirror controller. With the spacecraft orbital position and attitude maintained to within specified limits the pointing algorithm receives quaternion, ephemeris, and initialization data that are used to construct the required mirror pointing commands at a 100 Hz rate.

This comprehensive simulation will also aid in obtaining a thorough understanding of spacecraft disturbances and other sources of pointing system errors. Parameter sensitivity studies and disturbance analysis will be used to obtain "limits of operability" for the GIFTS

instrument. The culmination of this simulation development and analysis will be used to validate the specified performance requirements outlined for this instrument.

Table of Contents

List of Figures

List of Symbols and Abbreviations

Acronym / Symbol	Meaning
ACS	attitude control system
CCD	charge coupled device
C_{dim}	dimension of columns in a pattern
CM	control module
C_{num}	column identifier number for a particular field-of-view
D_{Frac}	fraction of a day past midnight (i.e. 0.5 for noon)
EO-1,3	Earth observing -1,3
f	Earth's flattening factor
FOV	field-of-view
F_{Static}	static force disturbance applied to reaction wheel
f_2, f_3, g_3	extrapolation coefficients
GHA	Greenwich hour angle
GHA_{Act}	Greenwich hour angle at time of actual usage
$GHA_{midnight}$	Greenwich hour angle at midnight
GIFTS	Geosynchronous Imaging Fourier Transform Spectrometer
GOES	Geostationary Operational Environmental Satellite
h	geodetic altitude
H_{RWA}	angular momentum of reaction wheel assembly
H_{SYS}	angular momentum of system
IFOV	instantaneous field of view

IRU	inertial reference unit
I_{SYS}	inertia of system
LEO	low Earth orbit
NOAA	National Oceanographic and Atmospheric Administration
NWS	National Weather Service
PID	proportional-integral-derivative controller
$PMVE$	rotation matrix from prime meridian to vernal equinox
$q_{dynamic}$	quaternion dynamic offset
q_{static}	quaternion static offset
q_{2000}	star tracker data in mean of J2000 frame
R_{dim}	dimension of rows in a pattern
R_{los}	position vector for line-of-sight (correct direction and magnitude)
R_{losi}	line-of-sight position vector in instrument frame
R_{losd}	position vector for new FOV line-of-sight (correct direction only)
R_{losPC}	position vector for pattern center line-of-sight
R_{num}	row identifier number for a particular field-of-view
R_{PC}	position vector for target center
$R_{s/c}$	spacecraft position vector
R_{scM00}	spacecraft position vector in mean of J2000 frame
RSS	root-sum-square
R_{targ}	position vector for target
R_{targPM}	position vector for target in prime meridian frame
$R_{targM00}$	position vector for target in mean of J2000 frame

RWA	reaction wheel assembly
$R_{\oplus}$	radius of the Earth
SM	sensor module
T_{act}	actual time
TDTM00	transformation matrix to mean of Julian 2000 frame
$T_{dynamic}$	dynamic torque disturbance applied to reaction wheel
T_{eph}	time tag for ephemeris
T_{ext}	external torques (i.e. solar pressure)
T_{FOV}	start time for FOV
Vsc	spacecraft velocity vector

α_{offset}	angular offset between field-of-views
δ_{lat}	incremental lateral displacement between field-of-views
δ_{vert}	incremental vertical displacement between field-of-views
$\Delta\varepsilon$	obliquity of the true equator with respect to mean ecliptic
ζ_{gyro}	damping constant for gyro
$\theta_{azimuth}$	commanded gimbal rotation angle about the mirror x-axis
λ	longitude
ϕ	geodetic latitude
$\Delta\psi$	longitude of the true equator with respect to mean ecliptic
$\psi_{elevation}$	commanded gimbal rotation angle about the mirror y-axis

ω_{gyro}	natural frequency for gyro
ω_{M}	measured angular rate vector
ω_{rate}	random rate noise vector
ω_{RW}	angular rate of reaction wheel
ω_{sys}	actual spacecraft angular rate vector
ω_{walk}	random walk noise rate vector

$\cdot$	multiplication symbol
$*$	quaternion multiplication symbol
$\bullet$	dot product symbol

List of Tables

Chapter 1. Introduction

The Geosynchronous Imaging Fourier Transform Spectrometer (GIFTS) instrument is part of the NASA New Millennium EO1 mission. This program's goal is to apply advanced-technologies to spacecraft that are considered high risk.

Several components on the GIFTS mission are new and emerging technologies that will receive on-orbit operational testing. In particular, an advanced star tracker is being developed to supply attitude information at a 100 Hz rate. By testing these emerging technologies in their operational environments, the hope is to lower the risk for future missions, and provide the scientific community with advanced equipment. Other technologies to be demonstrated on this mission include a suite of optical and interferometer scanners capable of producing high resolution images of the earth's atmosphere. The measurements taken will represent a revolutionary step in the area of meteorological forecasting and atmospheric research.

The purpose of this paper is to explore the pointing system performance characteristics involved with carrying out the strict pointing requirements outlined for this mission. Operational methodologies will be presented and discussed to better understand the motivation for using certain techniques. Presented in this thesis is a preliminary design and analysis of the algorithm used to command the GIFTS instrument pointing mirror. Development of the dynamic equations and coordinate systems used to describe various components of GIFTS and the accompanying spacecraft are also presented. Studies pertaining to pointing stability and jitter performance metrics have been performed to prove the operational feasibility of this mission. Other components that are known to have an impact on

the pointing performance will also be studied to obtain maximum limits of disturbance noise operation. In particular, a study pertaining to the GIFTS star tracker performance will be used to justify the added expense of this high cost technology. Also presented will be the cumulative disturbances added to the spacecraft-instrument system from the different components (e.g. gyros, reaction wheels). These studies are performed using a spacecraft simulation developed using the Matlab Simulink general-purpose simulation analysis program. A considerable amount of attention was given to the pointing algorithm and its position within the complete instrument-spacecraft system. This paper discusses the development of the GIFTS instrument pointing algorithm and the spacecraft simulation that is used as a validation tool for the full system.

1.1 Mission Background

The EO3 mission, or GIFTS, is the next step in validating advanced technologies for improving operational weather observing services. The technologies tested here build upon the experiences obtained from the predecessor EO1 mission managed by NASA Goddard Space Flight Center. EO1 was launched in November of 2000 as a follow-on to the Landsat 7 imaging spacecraft. The 705 km circular, sun-synchronous orbit at 98.7 degrees matched to within one arc minute, the Landsat 7 orbit and collected identical images for later comparison (Ref 1). Due to the spacecraft's position in LEO an instrument pointing accuracy of 0.05 degrees was required. The instrumentation aboard the EO1 consisted of an "Advanced Land Imager", "Atmospheric Corrector", and a "Hyperion (Hyperspectral Imager)". The Advanced Land Imager improved upon current spectral imaging by introducing several new bands and reducing mass, power, complexity and cost. The atmospheric corrector provided significant

improvement in the measurement of surface reflectance, which is used to correct surface imagery for atmospheric variability. The Hyperion provided spectral resolution into the hundreds of bands compared to 10 bands for the Landsat 7 instrument. This reflects a dramatic leap in the quality of data that can be used for mining, geology, forestry and agriculture.

Data obtained from the EO3 mission will be used by NOAA and NWS to provide early warning of severe weather. Detailed information on; water-vapor winds, ozone characteristics, and radiative properties of evolving clouds are some of the time-dependant variables of interest. Successive images of clouds and relative humidity will be used to reveal temporal changes in evolving weather conditions. The data will also be used to better understand long-term changes in the earth's atmosphere and water cycle.

Capabilities of GIFTS include:

- Regional to full-Earth visible and multi-spectral infrared imaging with 1- to 5-minute temporal frequency.
- Full-Earth temperature, moisture, and tracer wind sounding with 1- to 3-hour temporal frequency for global numerical weather prediction.
- Regional half-hourly high-resolution sounding for mesoscale intense weather observation and forecasting.
- Soundings of chemical composition for monitoring pollutant and greenhouse gas episode evolution and transport.

The EO3 GIFTS mission focuses on a more thorough utilization of an interferometer instrument, similar to the Hyperion, for imaging of the earth's atmosphere. Interferometers for scanning of the earth's atmosphere have been used in past scientific programs; however, the implementation of such a device on a geosynchronous platform has never been accomplished. It's this orbital location of 36,000 km, in a nominally circular orbit, that represents the next dramatic leap in providing continuous observation of large geographical areas. It's important to realize the benefits of positioning such an instrument at this distance. By demonstrating the measurement capabilities from a geosynchronous orbit the problems associated with aircraft and lower orbits, such as limited field-of-view, residence times, and footprint movement, are eliminated.

The benefit of a spectrometer is that it has the ability to discriminate, identify and quantify material composition at a subpixel level. An imaging spectrometer allows the construction of a picture in which each small element of the picture, known as a pixel, contains information on the spectrum of light; that is, the light is broken into its individual colors, as when you look through a prism (Ref 2). This allows the detection of minute concentrations of different signature spectra in the instruments CCD detector pixels. The imaging spectrometer on GIFTS works in the ultraviolet and infrared wavelength region, and the resulting data allow scientists to determine, among other things, the chemical composition of objects being viewed. Spectrometer technology has been utilized on several missions including the Deep Space 1 advanced technology demonstrator. The flight produced infrared spectral imaging of the Braille (1999) and Borrelly (2001) comets (Ref 2).

An imaging spectrometer operates by acquiring images of the same scene simultaneously in many contiguous spectral bands over a given spectral range (one might think of this as equivalent to a contiguous set of multi-color images) (Ref 3). By adding wavelength to the image as a third dimension, the spectrum of any pixel in the scene can be calculated. This allows the investigator to isolate any part of the target based upon its reflectance spectral signature. Once properly calibrated, these images can be used to obtain the reflectance spectrum for each image pixel, which can then be used to identify constituent elements in the target. The GIFTS instrument is capable of very high spatial and spectral resolution and works by taking an interferogram, a form of time-based multiplexing, of each picture element of an image. A Fourier transform of this signal represents the spectrum of that picture element.

Currently, a scanning instrument of this caliber in geosynchronous orbit does not exist. As a result, several time critical parameters of the atmosphere are not receiving the required attention. Regions of the earth's surface are not being analyzed and important time critical atmospheric data such as wind velocities are being lost. A significant improvement to the current measurement systems can be realized through the implementation of this technology.

1.2 Review of Previous Measurement Platforms

In the past, spectrometer instruments have been operated from manned aircraft over relatively small geographical locations. The method then progressed to high altitude unmanned aircraft and then to LEO satellite systems. An evolutionary approach to mature this technology has brought about significant advances in the gathering and use of these data. The performance metrics applied to this class of sensor have evolved from benchmarked methods that

painstakingly compile the data from many sources. Past options required individual surveying and meticulous compiling of data from sources such as hand sampling of crops for proper fertilizer usage, soil sampling for investigating mineral deposits, and frequent weather balloon sounding for adequate atmosphere analysis. Weather information provided by the GIFTS instrument will be equivalent to launching 100,000 weather balloons every minute at intervals of 2 miles (Ref 4).

In addition, the use of laser radar or lidar has been used extensively to accomplish the task of studying atmospheric particles and cloud composition. This method utilizes a telescope to measure reflected laser radiation similar to a standard radar system that collects information by bouncing radio waves off of clouds and rain. By using different wavelengths of laser radiation the lidar can measure ozone characteristics, aerosols, and clouds. The use of lidar is limited by its ability to acquire data only along the line-of-sight of the laser beam. Data obtained in this manner requires considerably more post processing then the currently envisioned GIFTS instrument and accompanying software. With the advent of the GIFTS instrument the past performance metrics applied to data acquisition, accuracy, computational resources usage, and cost effectiveness will be greatly surpassed.

1.3 GIFTS Instrument Proof-of-Concept Plan

Development of a plan to demonstrate and evaluate the pointing capabilities of the GIFTS instrument is presented here. The plan outlines the scope with which the research is carried out. Details of the individual steps of this plan are more thoroughly discussed in the following chapters.

In this thesis, the dynamical equations of motion for the GIFTS instrument pointing system and spacecraft hardware are assembled into a Matlab Simulink simulation. The simulation equations attempt to accurately replicate the actual dynamics of the coupled system consisting of a gimbaled pointing mirror on a 3-axis stabilized spacecraft. The gimbaled pointing mirror is used to target and track points of interest on the Earth's surface. The image is then reflected off the gimbaled pointing mirror and directed to the interferometer optics. The GIFTS instrument will perform scans by assembling a 512km x 512km instrument FOV into larger patterns. The patterns range in size from a regional 4 FOV x 4 FOV pattern to a larger global scan consisting of a 25 FOV x 25 FOV pattern. The difficulties associated with maintaining the point of interest in the instruments FOV constitutes a significant portion of the research performed. Realization of the GIFTS mission requirement for extended Earth coverage is achieved by step scanning the instrument FOV in a contiguous fashion across any desired portion of the visible earth. Automated tracking techniques will be used to maintain the FOV of interest to within specified tolerances. The pattern of FOV's has a sampling period ranging from minutes to hours, placing tight requirements on pointing knowledge and repeatability. However, the individual FOV's are required to be tracked for a duration of only 0.1 to 10 seconds. This places strict requirements on the pointing stability and jitter of the system.

The specific goal of the current investigation is to advance the state-of-the-art in the analysis of pointing systems. To accomplish this a full system simulation is constructed for design validation and disturbance analysis of the GIFTS pointing algorithm and associated mirror gimbals. The objectives of the present research are:

1) Design and analyze an autonomous pointing system algorithm
2) Construct a full system simulation including spacecraft
3) Investigate component disturbance profiles and validate instrument pointing system performance requirements

The performance metrics used for validation are obtained from specific requirements presented for instrument operation and science return. Chapter 2 presents a more detailed description of the GIFTS instrument pointing system operation and the motivation behind selecting the performance metrics. Specific information about the current system design is also presented to clarify the tasks that are to be accomplished. Chapter 3 begins the technical development of the GIFTS pointing system model simulation. This is followed by a description of the spacecraft model simulation given in Chapter 4. Chapters 5 and 6 then present analyses of disturbances within the system and validation methods for satisfying the pointing requirements, respectively. Concluding remarks on the research performed are given in Chapter 7.

Chapter 2. Description of GIFTS Instrument Tracking Operations

This chapter discusses the operational characteristics of the GIFTS instrument. A summary of the instrument and mechanical operating limits are outlined. Details of typical operations including the scanning methodology and FOV pattern scenarios are given. Finally, the technical requirements for performing instrument scanning are described.

Figure 2.1: GIFTS instrument and spacecraft

These technical requirements will serve as the performance metric for the system. Figure 2.1 shows the location of the GIFTS instrument with respect to the spacecraft.

The GIFTS instrument is comprised of a cryogenic Michelson interferometer in conjunction with a metrology laser to provide multispectral images with very high spatial and spectral resolution. It works by taking an interferogram, a form of time-based multiplexing, of each picture element of an image. The Fourier transform of this signal represents the spectrum of that picture element.

The GIFTS instrument is the primary instrument on the spacecraft. The instrument contains a two-axis gimbaled scan mirror with inductosyn® angular position transducers and brushless DC motors for repositioning. The pointing mirror is gimbaled +/- 12° for full earth scans. The instrument contains a 128 x 128 pixel focal plane detector array in the spectrometer, which translates to a 4km x 4km footprint per pixel providing a 512km x 512km total coverage on

the ground at nadir. A resolution capability of the instrument allows the scan to commence for a period from 0.1 seconds to 10 seconds per FOV. This variable scan duration time is derived from the interferometers ability to acquire varying data fidelity levels given by the minimum data for a 0.1 second scan or the maximum data for a 10 second scan. A number of FOV's can then be assembled into several predefined patterns to allow for larger earth coverage. This is accomplished by stepping the instrument line-of-sight to any desired direction on the earth face. With the spacecraft in a geosynchronous orbit at 77° West longitude, the visible earth-face extends to 13° East longitude and 167° West longitude and then North and South to the poles. The orbital plane of the spacecraft is also given by a 5° inclination, which has the effect of rotating the pattern by this amount.

Other necessary constraints imposed are that the time-to-point and settle to a new FOV be less than 1 second and that the pointing knowledge be better than 0.4 km for wind determination. All of this is carried out with an autonomous pointing system requiring an advanced knowledge of many specific constraints (e.g., pointing stability, instrument alignments).

An important characteristic of this project is the use of feedforward control in providing pointing mirror command angles. Spacecraft and orbit anomalies are characterized and compensated before a target tracking profile is supplied to the pointing mirror.

The system is divided into two subsystems consisting of a Control Module and Sensor Module to allow for flexibility and preserve mass balance.

Control Module (CM)

The CM contains the supporting electronics, payload controller, data processor, telemetry, and power subsystems. This module is configured as a separate package to provide future missions with a compact, high capability, and versatile flight-proven instrument command and data processing unit. The payload controller portion of this module initializes scan pattern parameters, constructs the pointing commands within the pointer profiler, and sets data collection modes. The other subsystems within this module aid in the processing and dissemination of the command and control information.

Sensor Module (SM)

The SM contains the star tracker, optical components, fast-steering mirror, and mechanical subsystems for pointing of the mirror. The components within this module are designed to function as a unit to facilitate ground testing and on-orbit diagnostics. The star tracker feeds quaternion information to the CM pointer profiler. Commands are then received by the SM gimbals to produce the step-stare pointing sequence. The SM's purpose is to sense changes in state and apply the corrective actions that keep the mirror on track.

2.1 Normal Operation Mode

This is the primary operational mode in which science data are collected. It is the only mode currently being applied to the pointing algorithm. Other modes such as the Moon mode, which is used for calibration purposes; and Thermal Unload mode, which flips the spacecraft every 6 months; are not incorporated into the simulation. The normal pointing mode provides the capabilities to hold the spacecraft in position to the specified requirements and begin

GIFTS instrument operation. The normal mode is characterized by the sequencing of several component operations. First, an initialization command is sent to the CM providing a start of scan time, pattern center location in geodetic latitude and longitude, dimensions of scan pattern (rows by columns), and resolution of FOV scan (dwell time). Secondly, a time tag signifying the stepping of the spacecraft solar array drive assemblies is given. This is provided to assure science data collection only between stepping actuations. Thirdly, a time tagged spacecraft ephemeris update is supplied to the CM. This is to provide a current spacecraft position and velocity value to begin the operation. Next, an updated and time tagged attitude quaternion is provided to the CM. All this information is finally sent to the "Pointer Profiler" to develop commands for the SM pointing mirror.

The following two subsections give more detail about the process used for carrying out the GIFTS instrument scans. A description of some of the scanning options is also given.

2.1.1 Scanning Methodology

Several methods for performing the scan operation are presented. In the past, atmospheric scanning operations were carried out from a LEO spacecraft. These operations had the benefit of short orbital periods allowing the spacecraft motion to serve as a scanning platform. This method is shown in Figure 2.2. However, the geosynchronous platform maintains a relatively fixed position in space with respect to the Earth. This is due to the spacecraft having an orbital period equal to the Earth's rotation (24 sidereal hours) (Ref 5). From this type of platform a two-axis gimbaled pointing device is required to orient a line-of-sight vector to the region of interest on the Earth's surface. This technique is shown in Figure 2.3.

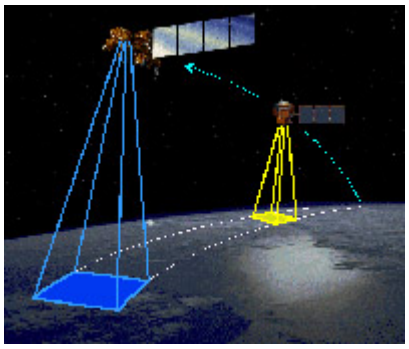


Figure 2.2: EO-1 and Landsat 7

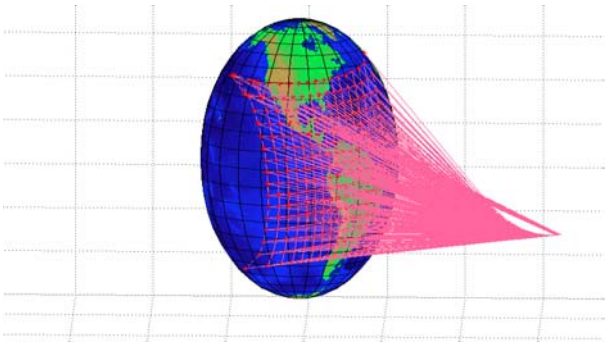


Figure 2.3: EO-3 GIFTS

The scan can then be carried out using a raster method or a step-and-stare method. The raster method is a scanning pattern of parallel lines. The step-and stare method is comprised of many small scan regions or FOV's that are then assembled into a larger pattern to cover the desired area. When there is a choice, step and stare is not usually the preferred scanning method because of the increased stability requirements to obtain the desired sensitivity. This sensitivity plays a dramatic role in the pointing requirements for the instrument. However, the GIFTS instrument uses the step-and-stare method. This method can obtain some unique pattern options, described in the next section, which cannot be obtained using the raster method.

2.1.2 Pattern Options

Utilization of the step-and-stare methodology allows the GIFTS instrument to perform scans of very select portions of the Earth. It also serves the purpose of allowing the instrument to vary its resolution or dwell time, from 0.1 seconds to 10 seconds, to acquire higher resolution data.

Several pattern configurations are available to accomplish unique tasks. A typical zone scan consists of a 4 FOV by 4 FOV pattern. With each FOV covering a 512 km by 512 km area on the surface of the Earth at nadir. The larger regional and area scans are formed using an integral number of FOV's for the rows and columns. A global scan is comprised of a 25 FOV by 25 FOV pattern. The pattern options are displayed in Figure 2.4. The size differences are based on user specified requirements pertaining to different weather systems. For example, a tornado can be better studied with a zone scan, while a hurricane would require a larger Area scan.

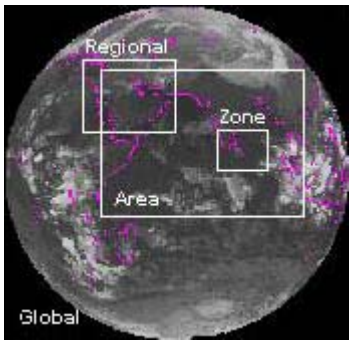


Figure 2.4: Pattern options

Assembly of these types of patterns can be carried out using either a snake or snail scanning techniques. The snake technique begins with a supplied pattern center location given by a latitude and longitude. Equations, to be discussed in section 3.2.3, are then used to position the first FOV in the top left corner of the pattern. The sequence then progresses to the right side of the pattern, then moves down and progresses to the left side as seen in Figure 2.5. The snail technique, like the snake technique, begins with a supplied pattern center location. The pattern then begins at this center point and works its way out in a snail fashion as seen in Figure 2.6.

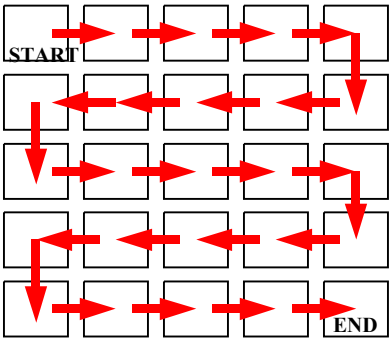


Figure 2.5: Snake pattern method

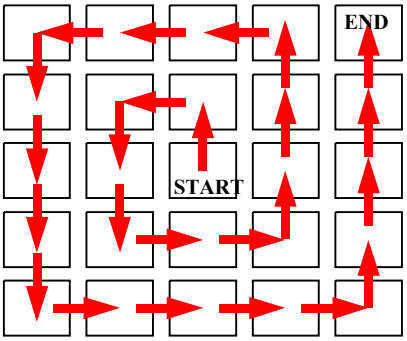


Figure 2.6: Snail pattern method

Several difficulties arise when using the snail technique. This type of pattern requires a greater number of coupled angle movements. Gimbal position encoding and control actuators must process more commands with this method. In contrast, the snake technique has the benefit of commanding gimbal angle moves for just one gimbal in a greater portion of the pattern. It also allows for more accurate FOV to FOV tracking by minimizing the compounding of gimbal movement errors. The snake technique is the method used for the pointing algorithm developed in this document. The remaining issues pertain to the requirement of applying the scanning methodology and pattern options previously described.

2.2 System Technical Requirements

Development of the scan pattern is complicated by inaccuracies in spacecraft and GIFTS instrument components. These inaccuracies are composed of hardware misalignments, data truncation, and incorrect timing that degrade the full system performance. Consequently, the end design objective is to meet the pointing requirements of the science instrument. Therefore, it is of critical importance to develop a pointing algorithm which maintains target pointing for the duration of a FOV scan while in the presence of disturbances from many

sources. This objective is realized by obtaining accurate disturbance profiles from all relevant components of the spacecraft and other obtrusive payload operations.

The GIFTS system requirements provide for establishing a threshold from which individual subsystems must adhere. The system is required to maintain geo-location pointing knowledge to within 10 km. A frame-to-frame relative pointing knowledge of 5.2 arc seconds (25 micro-radians) shall be maintained for the acquired frames over a 30-minute scan period. Pointing stability shall be maintained to within 2.3 arc seconds (11 micro-radians).

The requirements have been broken down into several areas and are described below. The process for analyzing these requirements will be discussed in Chapter 5. The sources involved in satisfying these requirement values are mapped in an error tree, also presented in Chapter 5. The component errors are subsequently simulated and the degree to which the components meet the requirements presented in Chapter 6.

Baseline Geo-Location Knowledge

The requirement pertaining to the geo-location knowledge of the FOV states that the geo-location of the centroids of the instantaneous geometric fields-of-view of each pixel shall be known within 1-km, for nadir pointing. A visualization of this requirement is given below in Figure 2.7.

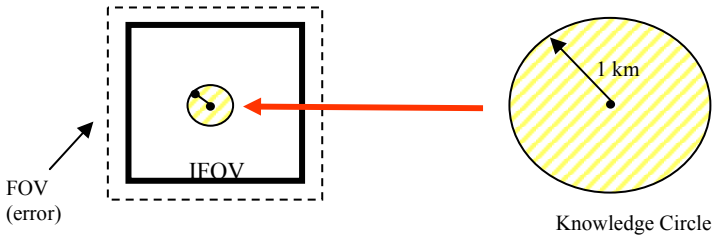


Figure 2.7: Geo-location knowledge

Figure 2.7 illustrates the limits of the requirement and how the instantaneous FOV can be located anywhere within these bounds. This requirement is characterized by sensor operation meaning a better sensor provides better knowledge. In this case it's the fidelity of the spacecraft ephemeris that is of extreme importance. Improvement of this number is based on obtaining better ranging of the spacecraft to provide a more accurate spacecraft position in orbit. This requirement is presented to give a baseline for the types of accuracies involved with this project. Analysis of this requirement involves trade studies associated with spacecraft ephemeris logistics, which are not within the scope of this thesis.

Frame-to-Frame Pointing Knowledge

The locations of the centers of the instantaneous geometric field-of-view of each pixel shall be known to better than or equal to 5.2 arc seconds (25 μrad) (1σ) from frame-to-frame for frames acquired over a period of 30 minutes at nadir pointing. This requirement must be strictly enforced to provide the necessary repeatability for accurate water-vapor wind measurements. This requirement is characterized by sensor operation. In particular, star tracker and IRU data must meet specifications. Instrument dynamics and subsystems must be fully understood to minimize misalignments and error sources.

Instrument Pointing Stability

The pointing stability is defined for frequencies below or equal to 10 Hz with a maximum angular excursion of 2.3 arc seconds (11.2 μrad) (1σ) for a period of 10 seconds, at nadir pointing. This can be further explained as control to within 1/10 of a 4-km instrument pixel footprint for the duration of a 10 second interferometer scans at nadir. This maximum value

for pointing stability will provide for quantitative analysis of data at or near cloud boundaries. The limits were derived to reduce image smearing.

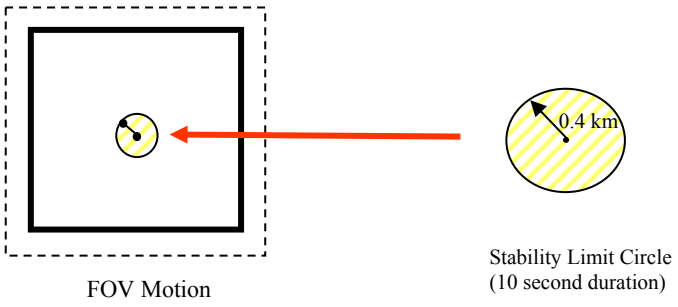


Figure 2.8: Instrument pointing stability

The circle in Figure 2.8 represents the bounded area, for a 10 second interval, in which the centroid of the geometric field-of-view can move. Analysis of this requirement is presented in Chapter 6.

Instrument Pointing Jitter

The pointing jitter is defined for frequencies greater than 10 Hz with a maximum angular excursion of 2.3 arc seconds (11.2 μrad) (1σ) for a period of 10 seconds, at nadir pointing. This can be further explained as control to within 1/10 of a 4-km instrument pixel footprint, for the duration of a 10 second interferometer-scan, for frequencies greater than 10 Hz. The jitter requirement differs from the stability requirement in the frequency range in which disturbances are present. Analysis of the stability and jitter requirements are presented in Chapter 6.

Chapter 3. GIFTS Pointing System Model Realization and Simulation

This chapter begins with a discussion of the breakdown of the full system simulation and the technical development of the GIFTS instrument-pointing algorithm. A model realization and simulation is presented to serve two purposes. First, the development of the algorithm will serve as a design and analysis tool from which future modifications to the mission can be easily simulated and validated. Secondly, the algorithm will serve as a guide for the actual flight software development. An initial description of the components is given and their location within the system is highlighted. The system is separated into two major components; GIFTS instrument pointing algorithm and spacecraft model. A thorough description of the mathematical equations used to construct the GIFTS instrument-pointing algorithm is given in Section 3.3. Chapter 4 completes the development of the full system with a realization and simulation of the spacecraft platform. These two components will be discussed, in their respective sections, and referenced back to Section 3.1 for their proper location within the full system.

3.1 Full System Overview

The simulation was designed in a manner that closely replicates the functional components of the actual system. A block diagram showing the different components of the simulation is given in Figure 3.1. Disturbance arrows are displayed to highlight the different paths through which noise travels. The simulation is separated into 3 major components: the spacecraft platform, Control Module, and the Sensor Module. Each of the modules carries out a specific task that is used to construct the pointing system command that meets the total pointing system requirement. The components in the CM replicate the instrument software and data

processing capabilities. The CM's purpose is to receive data from the initialization module, spacecraft bus, and star tracker; to perform calculations, then provide azimuth and elevation commands to the SM mirror gimbals. This module also houses the Pointer Profiler, which contains the pointing algorithm that is described in this chapter. The SM houses the mechanical equipment used to sense changes in state and carry out mirror pointing. The GIFTS star tracker is given as the system primary with a back up supplied by the spacecraft star trackers. A Kalman filter is used with the back-up system to provide an enhanced position quaternion to the Pointer Profiler than could be obtained with just the spacecraft star tracker.

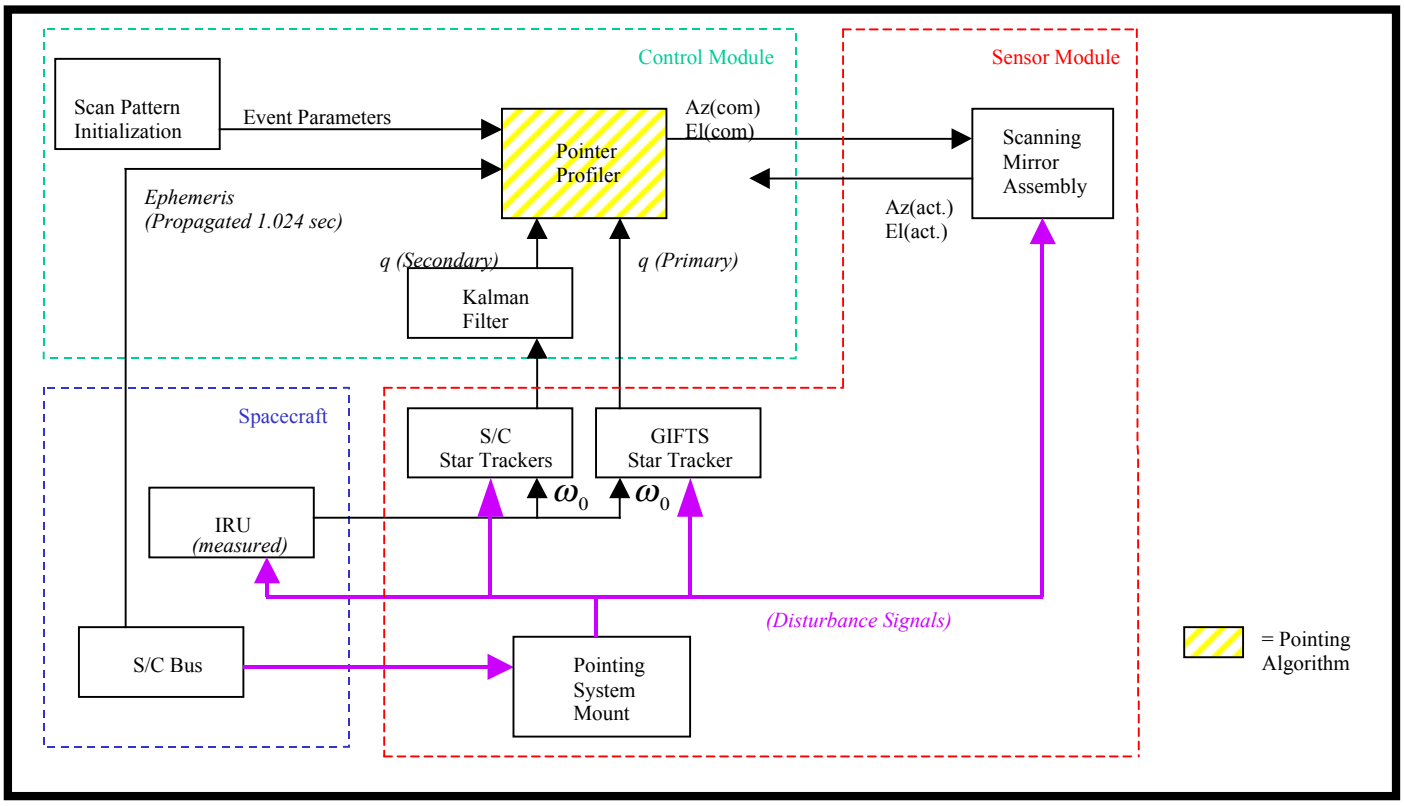


Figure 3.1: Full system block diagram

The next section provides a description of the different coordinate systems used to reference particular component origins. Precise knowledge of these reference frames is critical to properly interpreting data and transforming data from one reference frame to another.

3.2 Coordinate Systems

Coordinate systems are a critical part in developing a complex interrelated simulation. A coordinate system is defined for each component so that input and output can be related to other components. The following subsection defines each of the relevant component coordinate systems. A comprehensive description of the coordinate systems usage is then presented in section 3.3.

The coordinate systems are easier to visualize when applied to the actual operation of the instrument. The first coordinate system is defined by a right-handed system located at the center of mass of the GIFTS instrument-pointing mirror as seen in Figure 3.2.

The axes are defined by;

- X_{PM} directed along the interferometer optics axis (azimuth rotation),
- Y_{PM} a right-handed orthogonal set with X_{PM} and Z_{PM} (elevation rotation) and,
- Z_{PM} directed nominally toward nadir (line-of-sight).

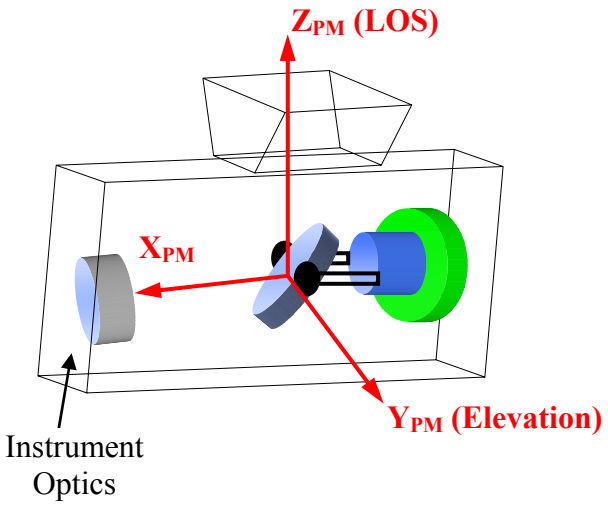


Figure 3.2: GIFTS instrument coordinate system

The second coordinate system relates components to the spacecraft body coordinate system, which has its origin at the center of mass of the spacecraft. The spacecraft body-fixed coordinate system is displayed in Figure 3.3 along with the GIFTS instrument axes.

The axes are defined by;

X_B roll directed along spacecraft velocity vector,

Y_B pitch directed negative normal to orbit plane, and

Z_B yaw directed toward nadir.

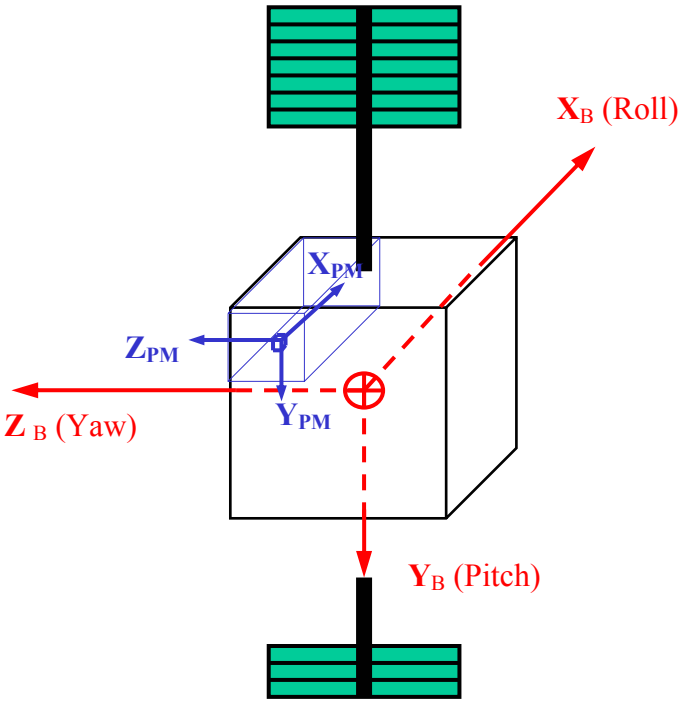


Figure 3.3: Spacecraft body coordinate system

The third coordinate system, displayed in Figure 3.4, gives the relation for the non-inertial local vertical coordinate system located within the orbit plane. The system is used to provide

an instantaneous reference for the development of the line-of-sight pointing vector that extends to the target point on the surface of the earth.

The axes are defined by;

X_{LV} forming right-handed orthogonal set with Y_{LV} and Z_{LV},

Y_{LV} directed opposite to the direction of the spacecraft angular velocity vector, and

Z_{LV} pointing from the spacecraft to the earth's center (nadir).

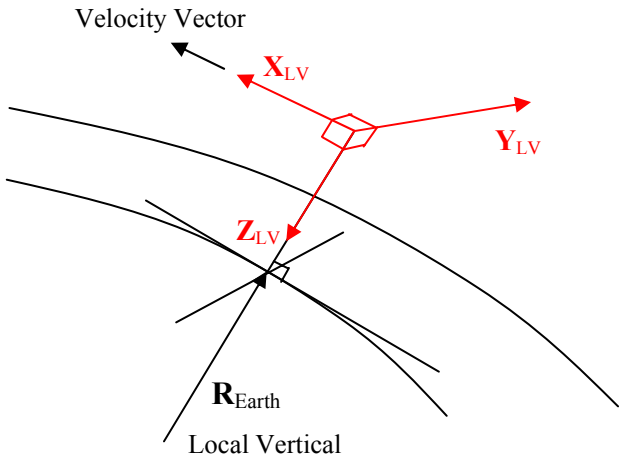


Figure 3.4: Local vertical coordinate system

Figure 3.5 shows the Earth centered rotating coordinate system. This is the reference system that defines the terrestrial latitude and longitude. The Greenwich meridian is the zero point for longitude with 360 degrees to the east for East Longitude and 360 degrees to the west for West Longitude (Ref 5). The Earth's equator is the zero reference point for latitude. The angular distance of the reference meridian from the vernal equinox is the Greenwich Hour Angle. This system is used to define the location of the spacecraft for a specific time of the day. Rotation of the spacecraft to this coordinate system is defined as the true prime meridian of date frame. The use of the GHA is described in section 3.3.3 and is an Earth-fixed coordinate system that rotates with the Greenwich meridian.

The axes are defined by;

X_{GM} lying in the plane of the equator pointing through the Greenwich meridian,

Y_{GM} right-handed orthogonal set with X_{GM} and Z_{GM}, and

Z_{GM} is in the direction of earth's angular velocity.

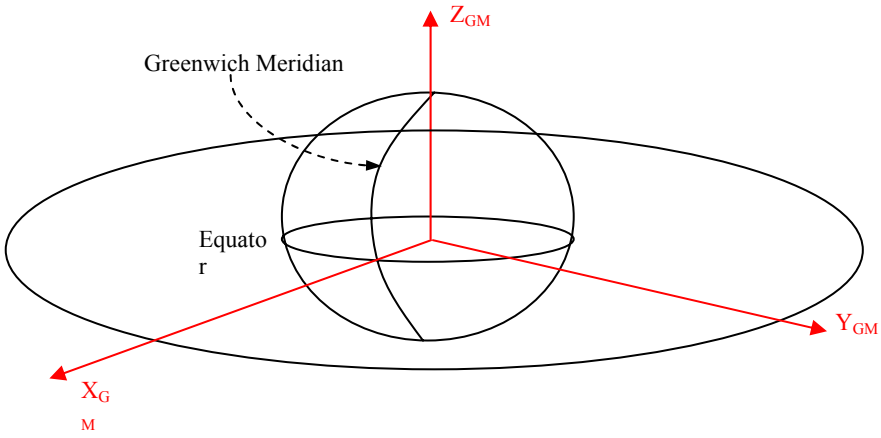


Figure 3.5: Earth centered rotating coordinate system

The last coordinate system is the Earth centered inertial coordinate system. This is the coordinate system of the mean equinox and equator of epoch, called mean of J2000, due to the definition of the vernal equinox in the year 2000. This is the reference frame in which the spacecraft ephemeris (position and velocity) are given. To get to this frame two rotations are used to transform from the true prime meridian of date frame to the true equinox of date frame, then a rotation from the true equinox of date frame to the mean of J2000 frame. The true equinox of date frame is comprised of a transformation matrix that accounts for the nutation and precession of the Earth and is described in section 3.3.4. The vernal equinox is defined as the point where the ecliptic crosses the celestial equator. The first point of Aries, which is actually in Pisces, defines the direction of the x-axis. This is the zero point for

calculating coordinates on the celestial sphere. Figure 3.6 displays the location of the first point of Aries and its relation to the Greenwich hour angle.

The inertial axes are defined by;

X_{ECI} which is in the direction of the vernal equinox,

Y_{ECI} forms a right-handed orthogonal set with X_{ECI} and Z_{ECI}, and

Z_{ECI} is in the direction of earth's angular velocity vector.

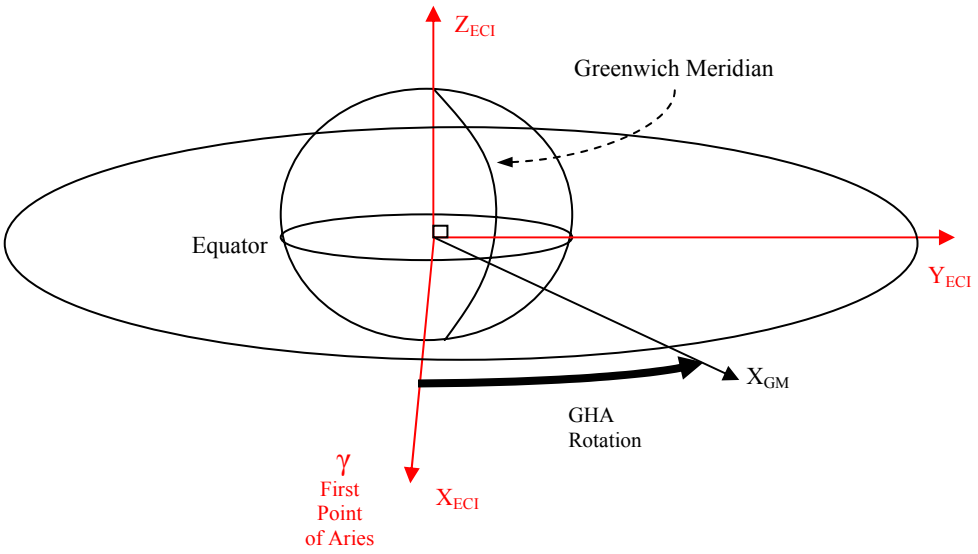


Figure 3.6: Earth centered inertial coordinate system

With the component reference frames properly defined, the development of the algorithm can begin. A description of the GIFTS instrument pointing system operation is discussed first to provide a foundation for the mathematical development.

3.3 Pointing Algorithm Introduction

The GIFTS pointing system consists of a pointing mirror assembly that includes actuating gimbals and a pointing algorithm to supply the angle commands to the gimbals, see Figure 3.7. Given below is a generalized view of the instrument and the mirror control motors.

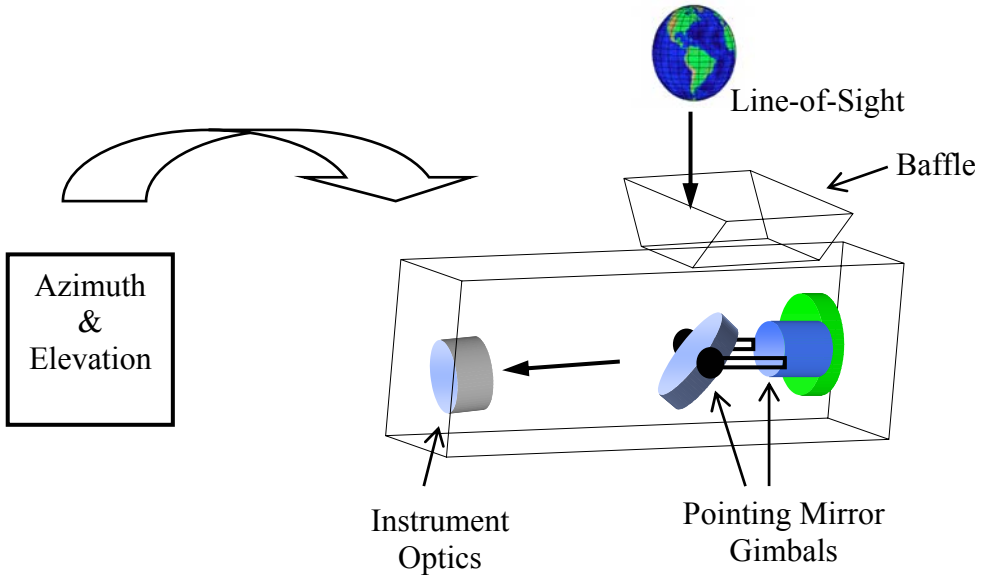


Figure 3.7: GIFTS pointing mirror assembly

The GIFTS Pointing Algorithm described here was developed to give a preliminary design for the problem of orienting the instrument optics with respect to a designated point on the earth through the use of a gimbaled pointing mirror. The commands are computed in an open-loop manner utilizing information supplied at a high data rate. Information is obtained from three different sources to compute the pointing angles for the mirror controller. The first component of information, obtained from the initialization routine, sets up the pattern center location, dimensions of the pattern, pixel overlap, resolution, and start time. The second component, obtained from the GIFTS star-tracker is in the form of quaternions at a delivery rate of 100 Hz. The third component of information is the spacecraft ephemeris (position and velocity) propagated by the spacecraft and delivered every 1.024 seconds. This data is time tagged and

sent to the pointing algorithm event manager. A clock based on spacecraft time is then used within the algorithm to synchronize the input data. More details on the timing sequence are given in section 3.3.2.

The pointing algorithm used in this simulation to provide mirror commands parallels the actual software that will be used for the mission. A considerable amount of development has gone into assembling the pointing algorithm. The task is to generate commands every 0.01-second for the azimuth and elevation gimbals in order to orient the instrument optics with respect to the line-of-sight vector through the baffle, as seen in Figure 3.7. This vector extends to the surface of the earth, as seen in Figure 3.8, correlating to a target location at the specified latitude and longitude of interest. The line-of-sight vector, $\mathbf{R}_{LOS}$, is obtained from a defined target location for the FOV, $\mathbf{R}_{Targ}$, and the spacecraft orbital position vector, $\mathbf{R}_{S/C}$, are given by the spacecraft ephemeris. The current methodology provides for a 10 second scan duration per FOV, with a 1 second settling time when moving from FOV to FOV.

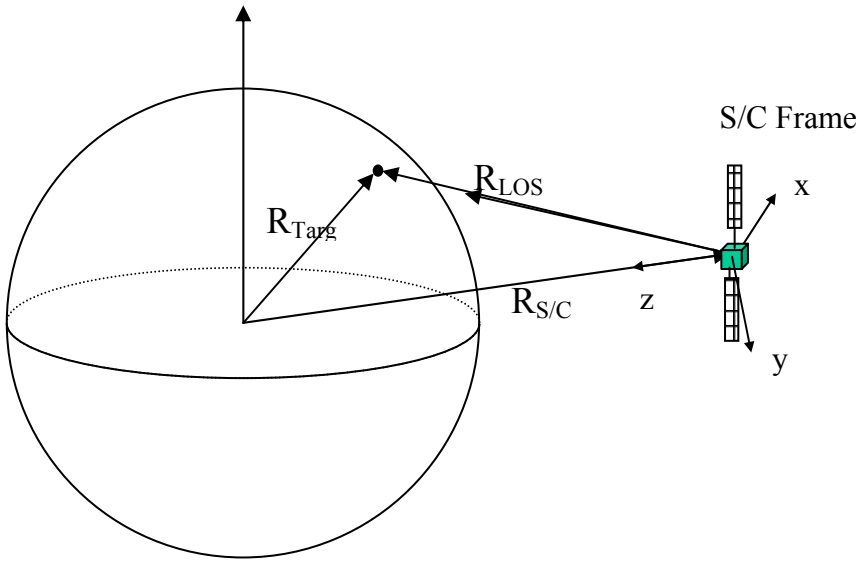


Figure 3.8: Line of sight vector to Earth

The algorithm receives the input data then progresses through a series of calculations, which are programmed for a simulation in Matlab. Each module is given a specific task and is implemented accordingly. First is the computation of the corresponding target vectors, for the entire pattern of FOV's. Second, transformation matrices are then assembled for rotations between several precisely defined coordinate systems. Third, a stepping sequence of commands is then initiated which proceeds through the pattern at a fixed rate based on FOV resolution and pointing mirror settling time. Precise vectors for the spacecraft location and the target location are then used for computation of a vector from the spacecraft to the target through vector subtraction. Instrument gimbal angles are then computed at 100 Hz to orient the instrument line-of-sight with the specified look vector. These commanded angles are then sent to the mirror controller in the Sensor Module.

3.3.1 Simulation Development / Overview

Through a series of modules, shown schematically in Figure 3.9, the data is transformed into tracking commands for specific FOV's within a larger pattern. The modules are located in the CM within the "Pointer Profiler". A block diagram of the routine is given in Figure 3.9. First, the FOV target locations are mapped to the earth with specific time tags using *COMPAT*. Supplemental modules are used within *COMPAT* to compute and apply transformations to the target vectors. A stepping sequence is then initiated to proceed through the pattern at a fixed rate based on FOV resolution and pointing mirror-settling time. The *SETTRG* module computes the transformation matrix to go from true prime meridian of date frame to the mean of Julian date epoch 2000 (mean of J2000) frame. An ephemeris propagator, consisting of f and g series coefficients, is then constructed in *SETSCPOS* to later update the spacecraft

position and velocity to a 100 Hz rate. The 100 Hz operating loop is then accessed for each FOV in the pattern. The preprocessed information is then used within the loop, along with the attitude quaternion from the GIFTS star tracker, to execute a vector subtraction and obtain the precise line-of-sight vector every 0.01 seconds. It is essential that the instrument-pointing algorithm work in conjunction with the spacecraft attitude control system to maintain accurate tracking. The attitude quaternions from the star tracker play a key roll in the development of accurate pointing commands. The pointing mirror control assembly also provides active compensation to remove high frequency jitter outside the bandwidth of the ACS. Therefore the pointing mirror control assembly is a critical component to achieve the desired pointing accuracy for the instrument.

Separate modules are controlled with the *PointDriver* (See Figure 3.9) event manager and carry out specified data conversions and coordinate transformations. The end result is the commanded gimbal angles for the mirror-pointing controller. The following subsections give more detail for the development of each module.

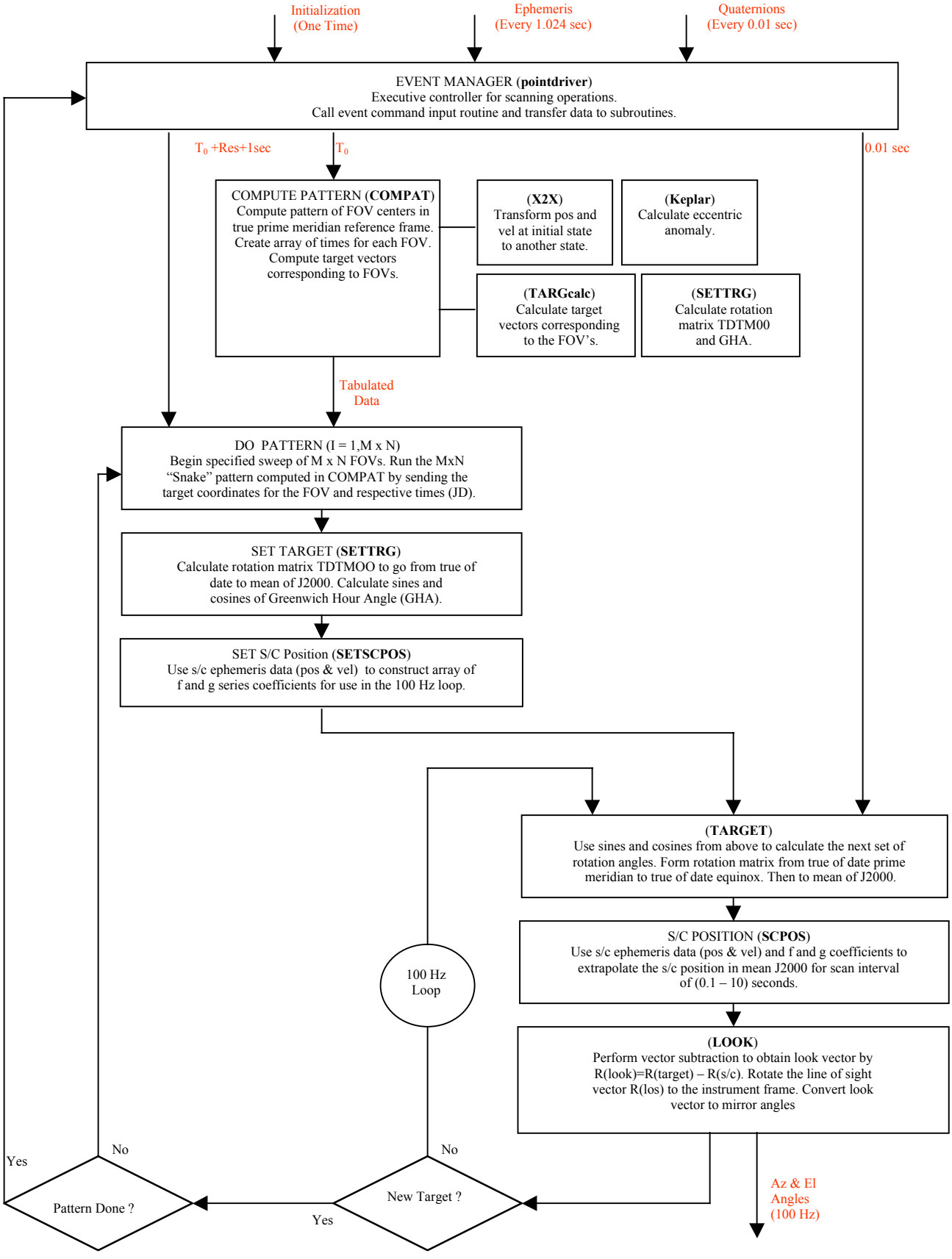


Figure 3.9: GIFTS instrument pointing algorithm block diagram

3.3.2 Pointing Algorithm Driver (PointDriver)

The *PointDriver* acts as an event manager for the scanning operations. Its purpose is to accept the information from the Simulink model and allocate it to the proper module in a specified sequence. The sequence is currently based on fixed intervals and doesn't allow for holds to be placed on the scan process. This is an issue that may be remedied later by either allowing for a skip of a FOV or an extended delay to be implemented before more data is acquired. A precursor to this sequence is the defining of the scan pattern parameters. A one-time parameter initialization of the pattern center in geodetic latitude and longitude, pattern dimensions (rows by columns), scan resolution (0.1 – 10 sec), and FOV pixel overlap allows the routine to begin.

The other inputs to this event manager are supplied as data increments (based on zero-order holds in previous components) and respective time tags. The ephemeris position and velocity in the mean of J2000 reference frame are input from the spacecraft every 1.024 seconds with a corresponding time tag. The time tag, based on the operating accuracy of the spacecraft clock, is given to an accuracy of 1 millisecond. Quaternion data are provided by the GIFTS star tracker at a 100 Hz rate with time tags. Quaternion component values are dimensionless in the range ($-1 <= q_i <= 1$) with the time tags given in Coordinated Universal Time (year, month, day, hour, min, sec, millisecond). A secondary system is provided by the spacecraft star trackers plus Kalman filter and would format the data just as the primary system. Only the primary, GIFTS system, is analyzed in this research. The spacecraft clock is also seen as an input here to provide a basis from which all the events can be interconnected and a timing sequence established.

The timing of the module interrelations is of critical importance. Currently the sequencing is determined by Boolean operators in conjunction with update flags. During operation the event manager is running at a 100 Hz rate to provide the relevant data to the loops. This assures the proper FOV is being targeted and the ephemeris for this FOV is propagated to the correct time. A feature that is being implemented here is that the ephemerides are being held for the duration of one FOV scan. This was done to avoid anomalies in updating of the data. The timing diagram in Figure 3.10, gives a simplified introduction to the events that take place for a typical scan of a pattern.

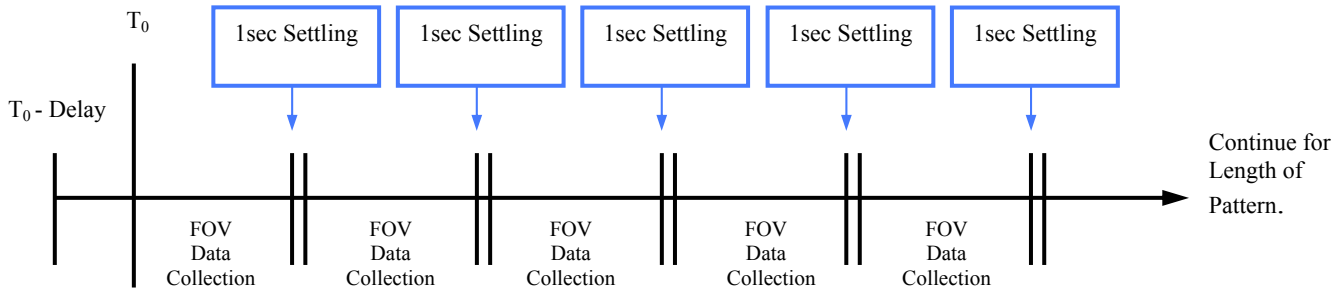


Figure 3.10: Scan pattern timing profile

The figure above shows the duration of events during the course of a pattern scan. The initial delay is shown here to take into account any processing delays that are present in the computation of the FOV target vectors. The scanning process then begins at the T_0 time and proceeds for the duration of the specified FOV resolution. A one second settling time delay is then inserted to allow the mirror control system to stabilize to a certain percent of the commanded value.

3.3.3 Compute Pattern (COMPAT)

Before the algorithm can start computing the mirror pointing angles, a table of information must be assembled using *COMPAT*. The table contains a listing of target vectors to each of the FOV's in the pattern and an associated time tag. This method of pre-computing the pattern target vectors was chosen to limit the number of operations carried out within the "Do Pattern" loop. The pattern is stored in a table in the true prime meridian of date reference frame. This is done to allow the algorithm to perform final coordinate system rotations just before the target vectors are sent to the *LOOK* module.

Equation 3.1 is used to transform the geodetic latitude, longitude, and altitude to Cartesian coordinates (Ref 6).

$$
\begin{aligned}
X &= (R_{\oplus} \cdot C + h\ ) \cdot \cos(\phi) \cdot \cos(\lambda) \\
Y &= (R_{\oplus} \cdot C + h\ ) \cdot \cos(\phi) \cdot \sin(\lambda) \\
Z &= (R_{\oplus} \cdot S + h) \cdot \sin(\phi) \\
&\textit{with}, \\
C &= \left(\cos^2(\phi) + (1-f)^2 \cdot \sin^2(\phi)\right)^{-\frac{1}{2}} \\
S &= (1-f)^2 \cdot C
\end{aligned}
\tag{3.1}
$$

The f term is the flattening factor with a reference value of 1/298.25722 (Ref 6), $R_{\oplus}$ is the equatorial radius with a value of 6378.14 km, ϕ is the geodetic latitude, λ is the geodetic longitude and h is the geodetic altitude. The altitude value is set to zero for this algorithm.

Next, a series of loops are used to assemble the snake pattern. It is here that a fixed value for slewtime is implemented to set up a timing sequence for each of the FOV's in the pattern.

With the snake pattern architecture in place the ephemeris position and velocity must be extrapolated to the start time for each of the FOV's.

The module *TARGcalc* is then used within *COMPAT* to rotate the target vectors to the true prime meridian of date frame. To accomplish this the transpose of the TDTM00 matrix, discussed in section 3.3.4 *SETTRG,* is used to rotate the ephemeris position and velocity to the true equinox of date frame. Another rotation is then preformed using the cosine and sine of the GHA to finally put the ephemeris in the true prime meridian of date frame. Rotation to this coordinate frame is recommended because the final target vector needs to be specified in the true prime meridian of date frame. Figure 3.11 illustrates the position of the vectors used for pattern development.

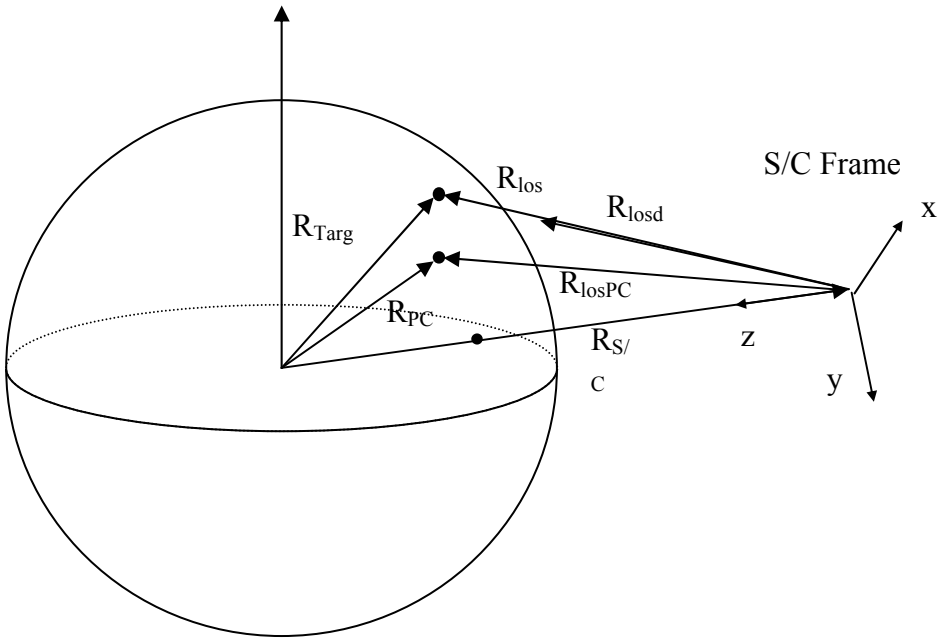


Figure 3.11: Geometry for FOV pattern

The spacecraft position, $\mathbf{R}_{S/C}$, and the pattern center, $\mathbf{R}_{PC}$, are now in the true equator and meridian of date frame. Vector subtraction is then used to calculate the line of sight vector from the spacecraft to the pattern center, $\mathbf{R}_{losPC}$ as such.

$$\vec{R}_{losPC} = \vec{R}_{PC} - \vec{R}_{S/C} \tag{3.2}$$

Assembly of the full scan pattern is accomplished by rotating the R_{losPC} vector a fixed number of degrees around the pitch and roll axes, as seen in Figure 3.12.

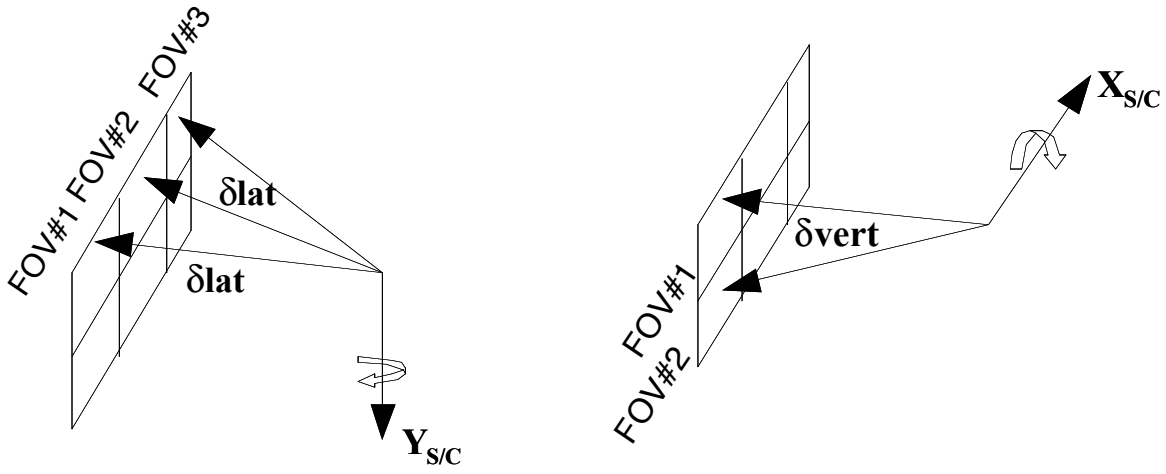


Figure 3.12: Pattern generation technique

The fixed incremental-angle value is based on the angular distance it takes to move from one side of a 512 km FOV to the other. This value is also used to establish the number of overlapping pixels between adjacent FOV's for the pattern. A value of $\theta = 0.0143$ radians is obtained from the equation for the length of an arc $s = \theta r$, with $s = 512$ km and $r = 35{,}786$ km. The offset angle θ, and the dimensions of the pattern are both used in the following equation to determine the lateral and vertical angle movements for each of the FOV's in the pattern.

$$\delta_{lat} = \alpha_{offset} \cdot \left(C_{num} - \frac{C_{\dim}}{2} - \frac{1}{2} \right)$$
$$\delta_{vert} = \alpha_{offset} \cdot \left(- R_{num} + \frac{R_{\dim}}{2} + \frac{1}{2} \right) \quad (3.3)$$

To accomplish these lateral (δ_{lat}) and vertical (δ_{vert}) movements, a proper coordinate system must be defined. The coordinate system used is the spacecraft fixed axis and is defined by taking the cross product of the spacecraft velocity vector and the position vector. This gives the direction of the y-axis. Another cross product is then taken with the spacecraft position vector and the newly defined y-axis to establish the instantaneous coordinate system for the spacecraft. Figure 3.13 gives a visualization of this coordinate system usage.

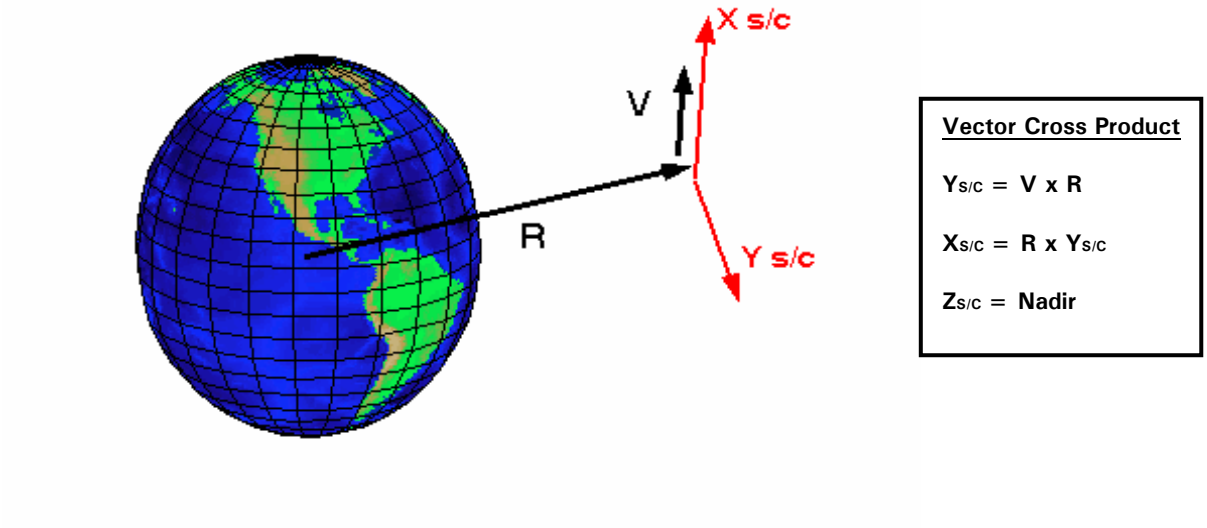


Figure 3.13: Spacecraft coordinate definition

The spacecraft coordinate system is defined by the positive z-axis pointing to the center of the Earth (nadir), the positive y-axis pointing along the negative orbit normal, and the x-axis completing the right handed system pointing roughly in the direction of the velocity for near

completing the right handed system pointing roughly in the direction of the velocity for near circular orbits. The spacecraft fixed coordinate system is yawed 180º every 6 months. This is a specified reorientation of the spacecraft for thermal unloading. The spacecraft coordinate system defined here does not take this reorientation into account therefore it is fixed with the x-axis pointing in the direction of velocity.

The rotation about pitch and roll rotates the line of sight vector from the pattern center to the direction of the center of the new FOV, $\mathbf{R}_{\text{losd}}$, shown in Figure 3.11. However, this rotation only gives the direction to the target and not the distance. The final line of sight vector $\mathbf{R}_{\text{los}}$ is a scalar multiple, K, of this line of sight direction vector, $\mathbf{R}_{\text{losd}}$. Vector summation of the GIFTS position vector and the line of sight vector produces the target vector as seen in equation 3.4 (Ref 7).

$$K\vec{R}_{losd} + \vec{R}_{S/C} = \vec{R}_{T\,\text{arg}} \tag{3.4}$$

To calculate the value of the scalar *K*, the Earth's surface is described using an ellipsoid.

$$X_e^2 + Y_e^2 + \left(\frac{Z_e}{1-f}\right)^2 = \text{Re}^2 \tag{3.5}$$

Where *f* is the flattening coefficient of the earth given by *f*=1/298.25722, and X_e, Y_e, and Z_e are coordinates of the points on the Earths surface. The intersection of the target vector with the ellipsoid defines the *K* values.

Substituting the x, y, and z components of $\mathbf{R}_{\text{Targ}}$ from the equation above yields,

$$(K \cdot X_{losd} + X_{S/C})^2 + (K \cdot Y_{losd} + Y_{S/C})^2 + \left(\frac{K \cdot Z_{losd} + Z_{S/C}}{1-f}\right)^2 = R_{\oplus}^2 \tag{3.6}$$

The above equation can then be expanded to a quadratic form to solve for K. However, we obtain two solutions. One solution is for the point we want and the other is for the point where the $\mathbf{R}_{losd}$ vector intersects the sphere on the backside of the earth. Thus, we discard the larger value and use the lesser of the two. This methodology utilizing the quadratic solution eliminates the need for solving complicated spherical trigonometry equations.

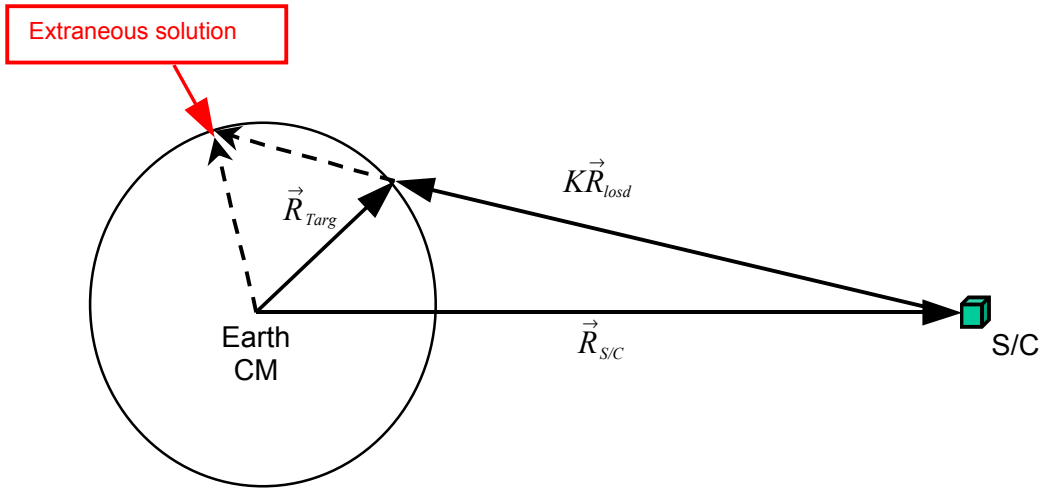


Figure 3.14: Quadratic equation solutions

Finally, the target vector is found by substituting the constant *K* back into equation 3.4.

3.3.4 Setup Target Vectors (SETTRG)

The *SETTRG* module is used within *TARGcalc* and the main "Do Pattern" loop to establish accurate rotation matrices and angle transformations for the coordinate systems. The rotation matrices purposes are to transform the target vector from true prime meridian of date frame to mean of J2000 frame. The angle rotations are actually the sine and cosine of the Greenwich hour angle for the specified Julian date. Also included are transformations for the precession

and nutation of the Earth. The methodology used to construct the matrices is discussed in this section.

The rotation matrix is separated into two parts. The first rotation from true prime meridian of date frame to true equinox of date frame accounts for the rotation of the earth about its axis and varies over the duration of a scan. The second rotation from true equinox of date to mean of J2000 frame accounts for the precession and nutation of the North Pole with respect to the inertial mean of J2000 frame. The full rotation matrix is initialized at the start of a scan and then again before each FOV.

The following equations describe the first rotation from true prime meridian of date frame to true equinox of date frame. The relationship between the vernal equinox, the prime meridian, and the Greenwich hour angle or GHA (at midnight) is given below (Ref 5).

$$GHA_{midnight} = 100.4606184^{o} + 36000.77005T + 0.00038793T^{2} - 2.6\times10^{-7}T^{3} \tag{3.7}$$

With,

$$T = \frac{JD_{midnight} - 2451545.0}{36525} \tag{3.8}$$

Where $JD_{midnight}$ is the Julian date at midnight. To find the actual GHA, calculate the $GHA_{midnight}$ for the previous midnight and add the amount the earth has rotated since the previous midnight, given by the following equation.

$$GHA = GHA_{midnight} + 1.002737909350795 \cdot 360 \cdot D_{Frac} \tag{3.9}$$

Where D_{Frac} is the fraction of the day past midnight (i.e. 0.5 for noon). Finally, convert *GHA* from degrees to radians. The *GHA* is output from this subroutine in the format of a sine and cosine of the computed *GHA*.

The second rotation from true equinox of date frame to mean of J2000 frame is a combination of two rotations, one accounting for the precession and the second accounting for the nutation of the North Pole. Let the nutations in longitude and in obliquity (of the true equator with respect to the mean ecliptic) be called $\Delta\Psi$ and $\Delta\varepsilon$, respectively. Each nutation is comprised of a long-period variation of about 18.6 years and a shorter period variation of about 182.6 days. They are calculated with the following equations (Ref 8).

$$\Delta\Psi = -0.0048^{o}\sin(125.0^{o} - 0.05295^{o}d) - 0.0004^{o}\sin(200.9^{o} + 1.97129^{o}d) \tag{3.10}$$

$$\Delta\varepsilon = 0.0026^{o}\cos(125.0^{o} - 0.05295^{o}d) + 0.0002^{o}\cos(200.9^{o} + 1.97129^{o}d) \tag{3.11}$$

where d = Julian Date – 2451545.0.

The transformation matrix from the true equinox of date frame to the mean equinox of that same date fame is

$$NUTMAT = \begin{bmatrix} 1 & \Delta\Psi\cos\varepsilon & \Delta\Psi\sin\varepsilon \\ -\Delta\Psi\cos\varepsilon & 1 & \Delta\varepsilon \\ -\Delta\Psi\sin\varepsilon & -\Delta\varepsilon & 1 \end{bmatrix} \tag{3.12}$$

With ε= 23.44° and $\Delta\Psi$ and $\Delta\varepsilon$ are in radians.

Next, the transformation matrix from mean equinox of date to mean of J2000, which accounts for the precession of the pole, is given by matrix *P* whose elements are given in equation 3.13 (Ref 8).

$$
\begin{aligned}
P_{11} &= 1.0 - (29724T^2 + 13T^3) \times 10^{-8} \\
P_{12} &= -P_{21} = (2236172T + 667T^2 - 222T^3) \times 10^{-8} \\
P_{13} &= -P_{31} = (971717T - 207T^2 - 96T^3) \times 10^{-8} \\
P_{22} &= 1.0 - (25002T^2 + 15T^3) \times 10^{-8} \\
P_{23} &= P_{32} = (-10865T^2) \times 10^{-8} \\
P_{33} &= 1.0 - (4721T^2) \times 10^{-8}
\end{aligned}
\tag{3.13}
$$

The equation for *T* is described on the previous page. Finally, the transformation matrix from true equinox of date frame to mean of J2000 frame, TDTM00, is found using the matrix multiplication:

$$TDTM00 = P \cdot NUTMAT \tag{3.14}$$

This gives as output from *SETTRG* the TDTM00 matrix and the sine and cosine of the GHA, for the specified input in the form of Julian date. The value for the TDTM00 matrix is held constant for each FOV, throughout the 100 Hz loop, due to the very small changes in these values during a 10 second scan.

3.3.5 Setup Spacecraft Position (SETSCPOS)

To provide for accurate spacecraft position and velocity information, the ephemeris data, which is input to the algorithm every 1.024 sec, must be propagated to the operational rate of

100Hz. This ephemeris propagation is accomplished with the use of a third order power series extrapolation. The coefficients used in this technique are calculated at the start of the scan and before every FOV scan. The coefficient values are held constant for the duration of the scan so as to avoid any jumps in ephemeris data and to limit the number of operations performed within the 100 Hz loop. This module sets up extrapolation coefficients, which are later used in *SCPOS*, utilizing the current spacecraft position and velocity, to calculate future spacecraft positions and velocities. The *SETSCPOS* module reads in the ephemeris position and velocity and returns the coefficients (Ref 8):

$$f_2 = -\frac{\mu}{2\left|\vec{R}\right|^3}; \qquad f_3 = \frac{\mu(\vec{R}\bullet\vec{V})}{2\left|\vec{R}\right|^5}; \qquad g_3 = -\frac{\mu}{6\left|\vec{R}\right|^3} \tag{3.15}$$

where μ is 398600.5 km^3/s^2.

3.3.6 Target Tracking (TARGET)

The *TARGET* module rotates the target vector from true prime meridian of date frame to mean of J2000 using the TDTM00 matrix and GHA calculated in *SETTRG*. The inputs to the function are the target vector in the true prime meridian of date frame (R_{targPM}), the start time of the field of view in seconds (T_{FOV}), the actual time in seconds (T_{act}), the TDTM00 rotation matrix, and the GHA. All these inputs are used to compute the current target vector, which is then sent to the *LOOK* subroutine. The first step is to update the GHA to the actual time by adding the amount the earth rotates from the start of the scan to the actual time, given by equation 3.16.

$$GHA_{Act} = GHA + (T_{Act} - T_{FOV}) \cdot 3.64605792765 \times 10^{-5} \tag{3.16}$$

where the numerical coefficient in equation 3.16 is the rotation rate of the earth in radians per second.

The rotation matrix from prime meridian to vernal equinox is

$$PMVE = \begin{bmatrix} \cos(GHA_{Act}) & \sin(GHA_{Act}) & 0 \\ -\sin(GHA_{Act}) & \cos(GHA_{Act}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.17}$$

Then the following equation rotates the target vector from True Prime Meridian of Date frame to Mean of J2000 frame.

$$\vec{R}_{T\arg M00} = [TDTM00][PMVE]\vec{R}_{T\arg PM} \tag{3.18}$$

3.3.7 Extrapolate Spacecraft Position (SCPOS)

To provide for accurate spacecraft position and velocity information the uploaded ephemeris data must be extrapolated to the operational rate of 100Hz. This is accomplished by the use of the f_2, f_3, and g_3 coefficients previously calculated in *SETSCPOS*. The ephemeris data is supplied as input to the pointing algorithm at an interval of every 1.024 seconds. The spacecraft position module uses the coefficients (f_2, f_3, and g_3) calculated in the set spacecraft position module (*SETSCPOS*) to extrapolate the ephemeris at a rate of 100Hz from the data

provided by the S/C computer. These coefficients along with spacecraft position ($R_{S/C}$), velocity ($V_{s/c}$), ephemeris time (T_{eph}) and actual time (T_{act}) are inputs to this module.

The module then calculates the *f* and *g* functions using the following equations:

$$f(t) = 1 + f_2 \cdot (T_{act} - T_{eph})^2 + f_3 \cdot (T_{act} - T_{eph})^3$$
$$g(t) = (T_{act} - T_0) + g_3 \cdot (T_{act} - T_0)^3 \quad (3.19)$$

The extrapolated position vector in the mean of J2000 frame, R_{SCM00}, is given by,

$$\vec{R}_{SCM\,00} = f(t)\vec{R}_{SC} + g(t)\vec{V}_{SC} \quad (3.20)$$

3.3.8 Line-of-Sight Vector (LOOK)

The *LOOK* module uses the spacecraft position in mean of J2000 frame (R_{SCM00}), and the target position in mean of J2000 frame ($R_{targM00}$), to calculate a line of sight vector (R_{los}), from the spacecraft to the target.

$$\vec{R}_{los} = \vec{R}_{t\,\mathrm{arg}\,M\,00} - \vec{R}_{SCM\,00} \quad (3.21)$$

Then R_{los} is rotated to the instrument frame (R_{losi}), using the quaternions from the star tracker (q_{2000}), and the orientation of the instrument frame to the star tracker frame, which has static (q_{static}) and dynamic ($q_{dynamic}$) components to account for misalignments. The quaternions from the mean of J2000 frame to the instrument frame (q_{ins}), are calculated using the following equation where the tilde "~" over the variable represents a 4 by 1 matrix comprised of a vector plus a scalar term.

$$\tilde{q}_{ins} = \tilde{q}_{2000} * \tilde{q}_{static} * \tilde{q}_{dynamic} \quad (3.22)$$

The line of sight vector (R_{los}) is transformed into the 4 by 1 matrix form by adding a zero term for the scalar rotation,

$$\widetilde{R}_{los} = X_{los}\hat{i} + Y_{los}\hat{j} + Z_{los}\hat{k} + \omega_{los} \tag{3.23}$$

With, $$\omega_{los} = 0$$

The new 4 by 1 matrix representing the line of sight vector in the instrument frame is then calculated using quaternion multiplication in equation 3.24.

$$\widetilde{R}_{losi} = \widetilde{q}_{ins} * \widetilde{R}_{los} * \widetilde{q}_{ins}^{-1} \tag{3.24}$$

Mirror gimbal angles are then calculated from the instrument line of sight vector (R_{losi}). Currently, the instrument coordinate system uses the designated frame, which was assumed in the coordinate system definition section, to compute the azimuth and elevation angles, as such.

$$\theta_{azimuth} = \arccos\left(\frac{Z_{losi}}{\sqrt{X_{losi}^2 + Z_{losi}^2}}\right) \tag{3.25}$$

$$\psi_{elevation} = \arccos\left(\frac{\sqrt{X_{losi}^2 + Z_{losi}^2}\cdot sign(Z_{losi})}{\sqrt{X_{losi}^2 + Y_{losi}^2 + Z_{losi}^2}}\right) \tag{3.26}$$

This completes the description of the GIFTS instrument pointing algorithm. The commands generated here represent the culmination of a complex handling sequence that operates for the duration of a pattern scan. The azimuth and elevation are then output from the “Pointer Profiler”.

3.3.9 Pointing Mirror Controller

The commands generated by the "Pointer Profiler" are then sent to the mirror controller in the CM. An error signal is then computed to serve as input to the pointing mirror control system. The command is then converted to a voltage input for the separate azimuth and elevation gimbal control system loops. Each controller is a proportional-integral-derivative (PID) controller functioning independently of the other. A transformation matrix, in which the spacecraft rates are converted to the instrument frame, provides spacecraft-to-instrument coupling. The rates are then transformed to a disturbance torque with the inverse inertia matrix and applied to the appropriate controller. A schematic block diagram of the system is given in Figure 3.15.

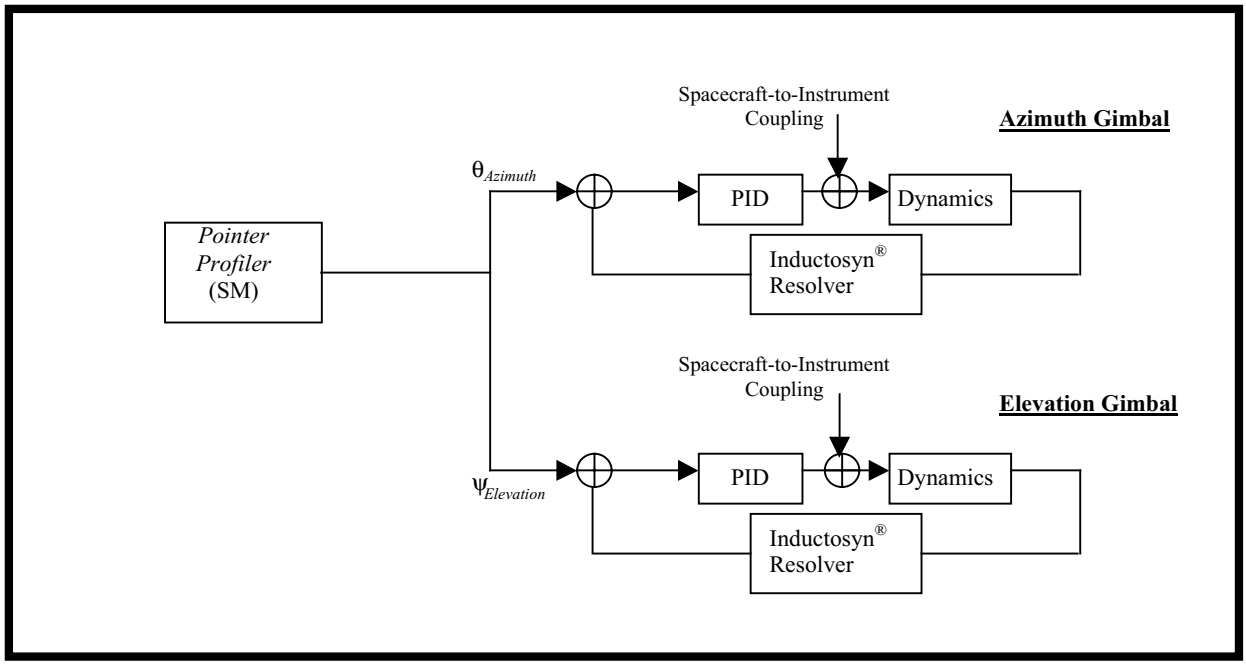


Figure 3.15 Pointing mirror control assembly

3.4 Scan Pattern Examples

Implementation of this algorithm was used to generate the following scan patterns. In the illustrations FOV target vectors emanate from the spacecraft to the location of interest on the surface of the Earth. The squares represent the FOV footprints and all together constitute the full pattern.

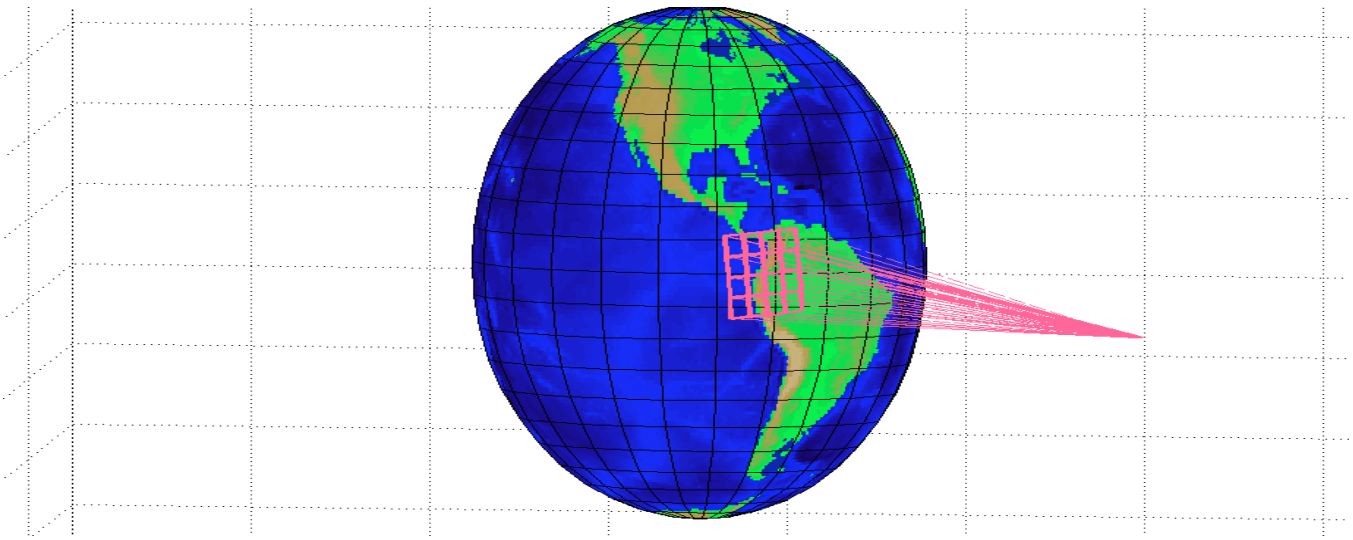


Figure 3.16: Regional scan pattern 4 FOV by 4 FOV

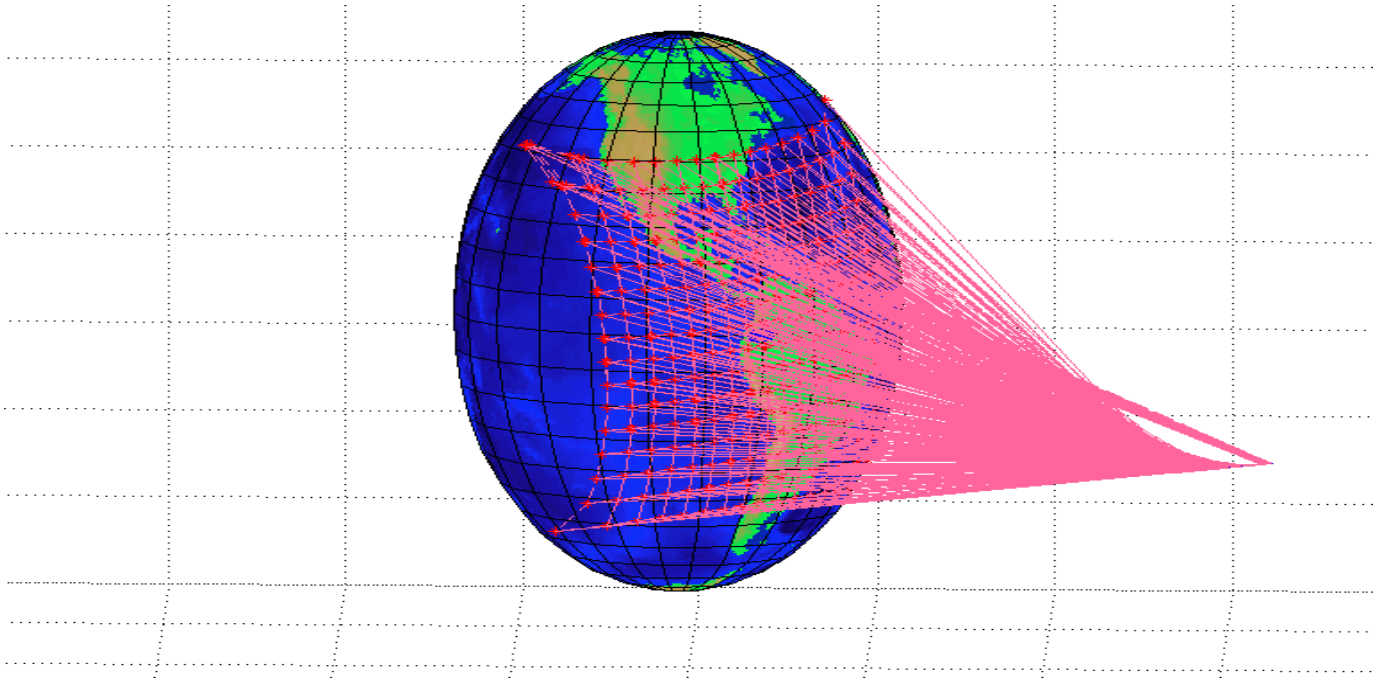


Figure 3.17: Area scan pattern 15 FOV by 15 FOV

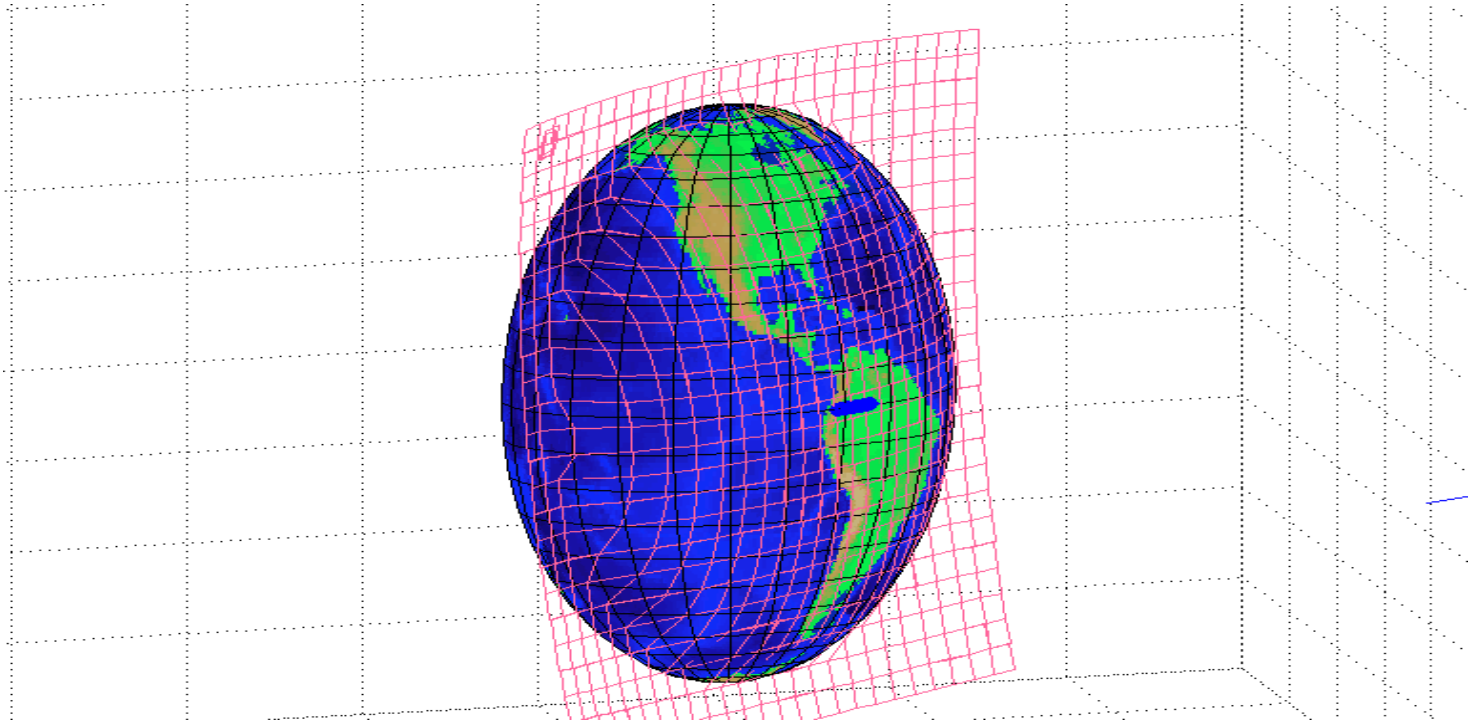


Figure 3.18: Global scan pattern 25 FOV by 25 FOV

The motivation for the analysis discussed is to verify that the pointing algorithms are correct and conform to the requirements set out for proper instrument operation. Shown in Figures 3.16, 3.17 and 3.18 are the graphical representations of the tracking algorithm commands for the development of a complete pattern. Figure 3.16 shows the rotation of the entire pattern due to the 5º inclination of the orbital plane. The individual FOV tracking and assembly of the complete pattern is also demonstrated for larger scan areas. Figure 3.17 shows a regional scan with the pattern center located at nadir. Stretching of the FOV's around the surface of the Earth is predominate. This is a factor that must be considered when targets of interest are near the limb of the Earth. A decrease in the fidelity of the weather data obtained is also experienced, due to the same number of CCD pixels receiving information for a larger surface area. The global scan shown in Figure 3.18 alleviates the loss of data seen in the region scan but adds a considerable amount of scan time to complete the pattern. A pattern generation methodology is under study to allow scans to commence only when a portion of the Earth is within the FOV, therefore eliminating empty space scans.

Chapter 4. Spacecraft Model Realization and Simulation

This chapter presents the dynamic equations used to simulate the spacecraft. Chapter 4 completes the description of the full system simulation by describing the spacecraft model. Together these two model realizations and simulations will be used to validate the finer details outlined in the mission requirements. The development of this simulation was carried out in a compartmentalized fashion. This method allows for easy upgrade of higher fidelity component models (e.g. gyros, star trackers, reaction wheels). The use of multi-payload, independently gimbaled systems, has made it critical for spacecraft designers to have knowledge of how the combined dynamics of all features of a spacecraft influence the line-of-sight pointing of the instrument. The hope here is to identify any adverse payload-spacecraft interactions and provide some form of compensation.

The main objective of the full spacecraft simulation is to validate the pointing algorithm in the presence of spacecraft disturbances and determine acceptable disturbance limits from expected noise sources. The dynamical interactions of the GIFTS instrument pointing mirror and spacecraft platform are examined. Also described are the different components that comprise the spacecraft. Disturbance implementation techniques are also mentioned to establish the scope of later studies. Stochastic inputs to the system are from the star tracker random bias noise, gyro random walk noise, gyro random rate noise, and reaction wheel dynamic and static imbalances.

4.1 Dynamics Model

A rigid body dynamics model using Euler's equations of motion is currently being used for the spacecraft. Since the spacecraft is assumed rigid only the rigid-body translational and rotational dynamics are modeled, no elastic deformations are considered in the dynamic description. This will allow for a mathematically ideal model from which further dynamics models, such as structural vibrations and fluid sloshing, can be appended. Generally, rigid body dynamics models are used to study ACS designs and validate control algorithms (Ref 9). Long-term effects can easily be studied with this ideal model because of the reduced computer run time. Typically ACS bandwidth is much smaller than the lowest natural frequency of the spacecraft structure.

The equations of motion for the spacecraft are given below.

$$\dot{\underset{\sim}{H}}_{SYS} = \underset{\sim}{T}_{EXT} - \left[\omega^*_{SYS}\right] \cdot \underset{\sim}{H}_{SYS} \tag{4.1}$$

Where each quantity with a tilde on its bottom is a three-component column matrix.

with,

$$\omega^*_{SYS} = \begin{bmatrix} 0 & -\omega_{3,SYS} & \omega_{2,SYS} \\ \omega_{3,SYS} & 0 & -\omega_{1,SYS} \\ -\omega_{2,SYS} & \omega_{1,SYS} & 0 \end{bmatrix} \tag{4.2}$$

$$\underset{\sim}{\omega}_{SYS} = \left[I^{-1}_{SYS}\right]\left(\underset{\sim}{H}_{SYS} - \underset{\sim}{H}_{RWA}\right) \tag{4.3}$$

where, $\underset{\sim}{H}_{SYS}$ = Angular momentum of the system

$\underset{\sim}{H}_{RWA}$ = Angular momentum of the reaction wheel assembly

$\underset{\sim}{T}_{EXT}$ = External torques (e.g. solar pressure torques)

$\underset{\sim}{\omega}_{SYS}$ = Angular rate of the system

I_{SYS}^{-1} = Inverse inertia of the system

The equations are implemented in a feedback loop, as seen in Figure 4.1, to allow the angular momentum of the system to update the spacecraft angular rate.

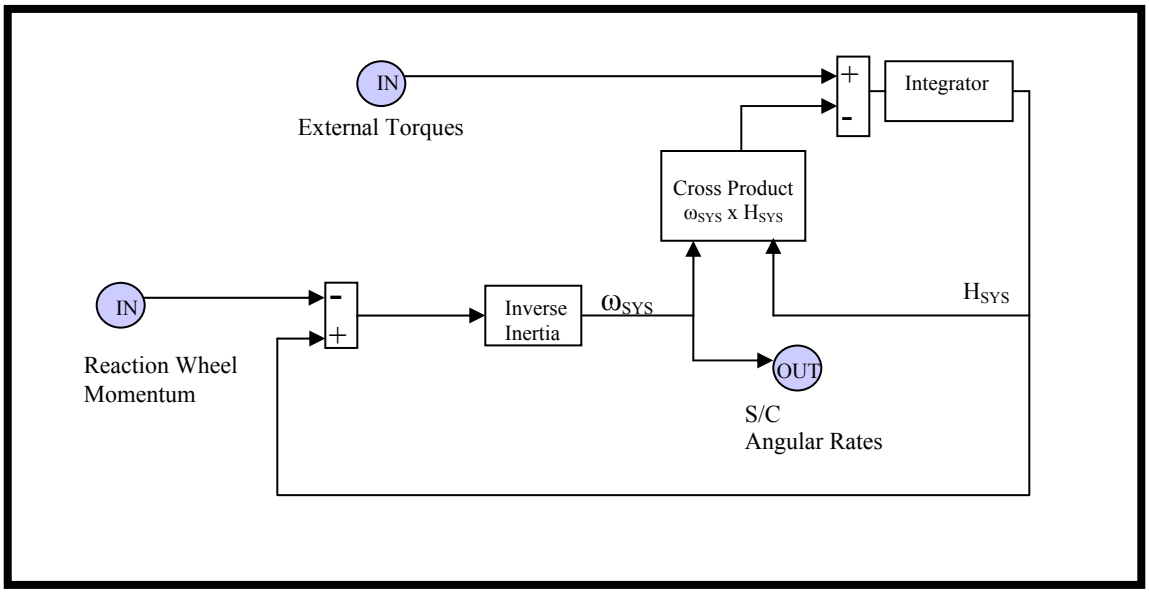


Figure 4.1: Spacecraft rigid body dynamics

The addition of a disturbance momentum from the solar array ratcheting assembly has been neglected since the arrays are assumed held constant for the duration of the 10-second scan period. The disturbances from the stepping motors are known to be a significant disturbance source (Ref. 10). Another part of the research involves the interaction of the spacecraft platform with the instrument-pointing mirror. It was assumed that the coupling of the spacecraft dynamics with the mirror was essential to modeling the complete system.

However, the inertial coupling of the instrument to spacecraft, to evaluate spacecraft motion, is negligible due to the low inertia and small travel of the instrument-pointing mirror.

4.2 Attitude Control System

The attitude control system (ACS) provides a precision pointing platform from which the instrument pointing mirror targets and tracks selected points of interest on the earth. It consists of sensors, actuators, and mechanical device control laws to properly orient the spacecraft with respect to an inertial coordinate frame. An inertial reference unit (IRU) comprising three gyros detects changes of attitude with time. The normal control of the spacecraft attitude is accomplished through angular momentum exchange and storage using four reaction wheels with one wheel along each principal axis and one skew wheel. A Proportional-Integral-Derivative (PID) controller is used to provide the command angular accelerations for the spacecraft. A Simulink block diagram of the ACS is presented in Figure 4.2.

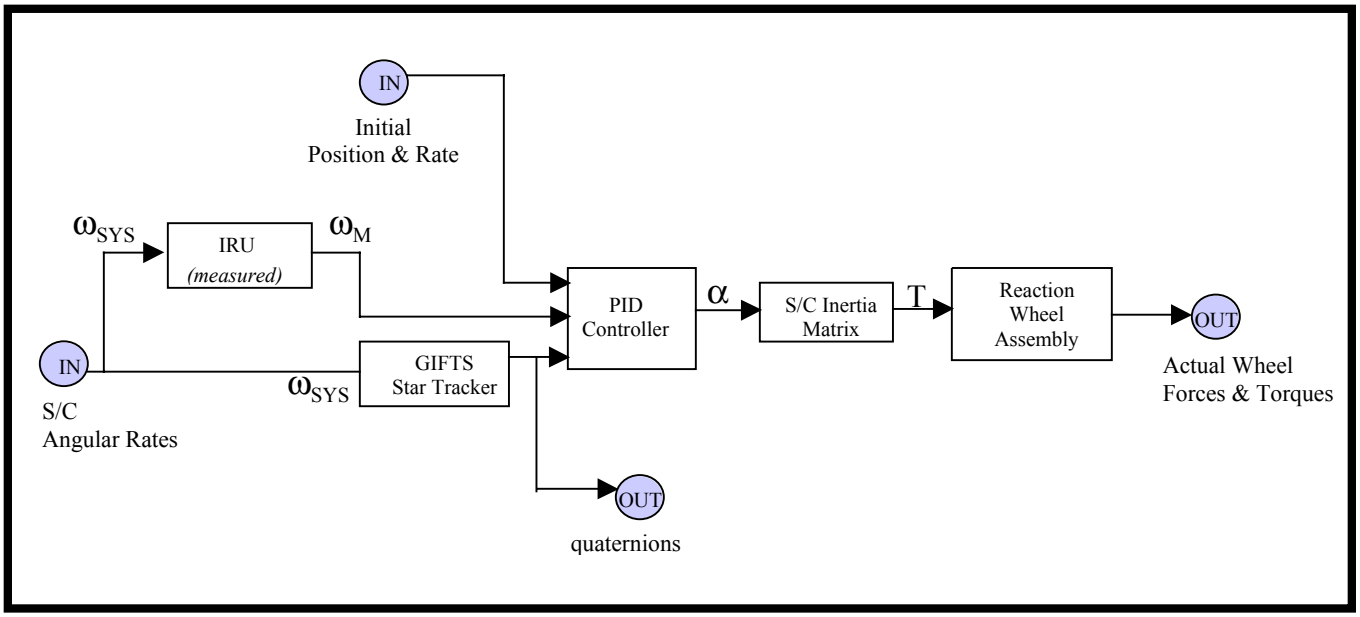


Figure 4.2: Spacecraft attitude control system

This figure shows how the spacecraft system angular rates are processed to form simulated quaternion attitude information, commanded accelerations, and then torques to the vehicle with the use of reaction wheels. The following sections provide further details of various spacecraft components.

4.3 Star Trackers

The star tracker being used is part of a new technology demonstration initiative. With the use of this new technology an attitude quaternion will be delivered to the instrument pointing system every 0.01 seconds. In the case of a line-of-sight type instrument the spacecraft attitude needs to be measured at sufficiently high bandwidth, well beyond what is commonly provided by inertial reference units (Ref. 9). The model accomplishes this task by using the time derivative of the quaternion vector at the desired rate. The true attitude quaternion, a rotation from the inertial reference frame to the body frame, is initialized in the setup file and propagated using the true spacecraft rates. The attitude of the spacecraft is described with respect to the inertial mean of J2000 reference frame. The equation used for the propagation calculation is given below.

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & \omega_{3,SYS} & -\omega_{2,SYS} & \omega_{1,SYS} \\ -\omega_{3,SYS} & 0 & \omega_{1,SYS} & \omega_{2,SYS} \\ \omega_{2,SYS} & -\omega_{1,SYS} & 0 & \omega_{3,SYS} \\ -\omega_{1,SYS} & -\omega_{2,SYS} & -\omega_{3,SYS} & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{4.4}$$

The quaternion is then integrated, normalized, and then a random noise signal is added to obtain the estimated star tracker quaternion every 0.01 seconds. The baseline parameters were

selected so that the star tracker noise culminates in an attitude error of 2.0 arc seconds per axis (Ref. 11). An advantage to using quaternions over Euler angles is that the kinematic equation is linear, requires no trigonometric functions, and is free of singularities. They also have the advantage that successive rotations can be carried out using quaternion multiplication.

4.4 Inertial Reference Unit

The inertial reference unit (IRU) consists of three gyros with spin axes orthogonal to each other providing spacecraft angular rate measurements every 0.1 seconds. The gyroscope accuracy is limited by instrument drift. A good determination of the deterioration of the gyro attitude information (deg per hour) must be known in order to obtain the precision of the ACS. For this purpose the gyro is used in conjunction with a star tracker to smooth the data and provide orientation initialization.

The gyro is modeled with a 2nd order transfer function with an operational rate of 5 Hz and an assumed damping constant equal to 0.707. The dynamic equation used for each of the three gyros is given as such;

$$TF_{gyro} = \frac{\omega_{gyro}^2}{s^2 + 2\zeta_{gyro}\omega_{gyro}s + \omega_{gyro}^2} \tag{4.5}$$

$$\text{with,} \quad \omega_{gyro} = 10 \cdot \pi \quad \text{(rad/s)} \tag{4.6}$$

$$\zeta_{gyro} = \frac{\sqrt{2}}{2} \tag{4.7}$$

The spacecraft angular rates with the addition of gyro random walk noise and gyro random rate noise constitute the output of the simulated gyro. The mathematical model for this spacecraft component is given below in equation 4.8.

$$\omega_M = \omega_{SYS} + \omega_{Walk} + \omega_{Rate} \tag{4.8}$$

with, ω_M = Measured angular rate vector

ω_{SYS} = Actual spacecraft angular rate vector

ω_{Walk} = Random walk noise rate vector

ω_{Rate} = Random rate noise vector

Each component of ω_{Walk} and ω_{Rate} is a normally (Gaussian) distributed random variable. The following baseline parameters were selected for the IRU:

- ω_{Walk} = 0.01 arcsec/sec
- ω_{Rate} = 0.01 arcsec/sec

The impacts of the random rate noise and random walk are described in Chapter 5.

4.5 Reaction Wheels

The reaction wheel assembly (RWA) consists of three reaction wheels with spin axes orthogonal to each other and a redundant fourth wheel added at an equal angle to the other three to minimize wheel speeds and provide control redundancy. The optimum angular orientation of the spin axes relative to the spacecraft's major axis depends on the system's inertial properties and the pointing requirements. For this research the three primary wheels are located along the spacecraft primary axes with the fourth wheel axis located equidistant to the other axes. The vector of commanded wheel torques is converted to a 4-vector with the

extra degree of freedom used to achieve wheel momentum biasing. The momentum biasing is characterized by an angular momentum vector oriented along the negative pitch direction and a coupling of the roll and yaw axes. As is characteristic for pitch momentum biased systems the scheme takes advantage of the reaction wheel gyroscopic coupling of roll and yaw to control yaw without a direct yaw sensor (Ref 12). This coupling of the roll and yaw axes is controlled using the 4th wheel which keeps the wheel speeds within a certain range.

The reaction wheel angular momentum is transformed to the spacecraft body coordinate system and subtracted from the system momentum. This follows the law of conservation of angular momentum by producing an equal torque on the spacecraft body axis, but in the opposite direction. These are classified as internal torques and do not affect the total system momentum.

The RWA is modeled with a 1st order transfer function with an operational rate of 10 Hz. Wheel disturbances of considerable interest pertain to the wheels dynamic and static imbalances. The model assumes that the disturbances consist of discrete harmonics of the reaction wheel speed with amplitudes proportional to the square of the wheel speed (Ref 13). Thus applying disturbance forces and torques as a function of wheel speed. Startup and dynamic friction effects are ignored. The dynamic equation used for each of the four reaction wheels is given by a 1st order transfer function based on a previously developed system and given as such;

$$TF_{RWA} = \frac{\omega_{RWA}}{s + \omega_{RWA}} \tag{4.9}$$

$$\text{with,} \quad \omega_{RWA} = 20 \cdot \pi \quad \text{(rad/s)} \tag{4.10}$$

The following parameters were selected for operation of the RWA, Table 4.1.

Table 4.1: Mechanical specifications for RWA A-15 reaction wheel (Ref 12)

Feature	Value
Angular Momentum	+/- 20 Nms
Speed Range	+/- 2200 RPM
Torque Limit	0.3 Nm
Wheel Inertia	0.0316 kg-m^2

Pointing Control Law

The 3-axis torque commands are computed via a controller consisting of three decoupled, proportional-integral-derivative (PID) controllers. The three-component command vector is separated into four components in the wheel with the fourth wheel used for momentum biasing. Momentum biasing is achieved with the use of a corrective offset torque that allows for control of the mean wheel momentum by having the offset along the null vector providing zero torque in the body frame. The control inputs are the roll, pitch, and yaw position and rate errors from the star tracker and gyro, respectively.

Chapter 5. System Disturbances and Error Sources

Now that the dynamics of the full system have been presented, a description of the types of disturbances and their effects on this full system is presented. This chapter presents information on the different disturbances within the full system and other error sources that may cause degraded performance for the GIFTS instrument pointing system. The objective is to expose areas within the system that will pose a problem for meeting the mission requirements. This will serve as a guide for any future design or operations changes that may impact the project.

Several factors are key to understanding the dominant sources of disturbance and jitter.

- Mass and inertia of moving components
- Torques applied by payload, solar array motors, and spacecraft ACS
- Control Laws for actuators
- Flexible characteristics of spacecraft

The first three factors will be used within the simulation to replicate actual system operation. However, the use of flexible body dynamics is not implemented in this research. The method used for the simulation and analysis of the disturbances within the system is based on defining a profile for each of the components (i.e. gyros, star tracker, reaction wheels). The profile consists of a disturbance characteristic (e.g. random walk, random rate), and an accompanying mathematical method for applying it to the true value.

This chapter presents the effects that spacecraft operations, and other mechanical payloads, have on the GIFTS instrument dynamic response. Other disturbance sources will be discussed with emphasis given to those that will have the greatest influence on the full system. A system

error analysis will also be presented to show the traceablity of the disturbance sources through the full system.

5.1 System Disturbance Sources

The GIFTS instrument in combination with the spacecraft is composed of many moving and/or flexible structures that interactively produce a myriad of disturbance torques. This includes the reaction wheels, solar array drive motors, GIFTS mirror gimbals, GIFTS interferometer, and flexible structure of the entire coupled system. Some of these disturbance sources are negligible and will not be applied to the simulation developed in this research. Others are of great interest to the performance of the instrument-pointing algorithm and will be discussed.

Currently, the spacecraft system is assumed to be a rigid platform with the pointing mirror attached at a gimbaled mount. Solar arrays are also attached to this platform and actuated by stepper motors. Instrument motion, stepper motors, and vibrating components are given as disturbance inputs. Nonlinear dynamics that result in variations of the mass matrix are assumed negligible (Ref. 9). Environmental disturbances such as aerodynamic drag and gravity-gradient torques are negligible at geosynchronous orbit and therefore neglected. The pointing errors considered are exclusive of those contributed by anomalies such as time jumps in the spacecraft ephemeris.

A description of the disturbances emanating from the GIFTS instrument and spacecraft bus is given in the following two subsections.

5.1.1 GIFTS Instrument

The GIFTS instrument contains many interconnected systems that are highly susceptible to stability and jitter disturbances. The model developed thus far is used to predict and study instrument and spacecraft response to a range of disturbance profiles. Sources of error inherent to the instrument can also be profiled and added to the dynamics equations. The driving force for this research is to analyze as many of the disturbances sources to establish acceptable levels and recommend suitable methods for isolation or removal of the adverse vibrations.

Mechanisms internal to the interferometer produce adverse motion and jitter in close proximity to the pointing mirror. Dynamics studies are commencing on the development of the optics substructure to reduce vibrations and deformations. Isolators are being investigated to reduce the amount of disturbances traveling from the spacecraft. Once these tests are completed they will be implemented into the simulation in the form of transfer functions. The transfer function will replicate the dynamical interactions between contact points.

Static and dynamic imbalances are also a critical area of study. Variations in component manufacturing cause permanent abnormalities that send disturbances through the system. Certain types of disturbances are invariant over time and appear as a fixed pattern (bias) over the FOV. The appropriate manufacturer will provide a method for ground testing and characterizing these disturbances. A technique can then be used to correct for these disturbances with a feed-forward control signal. As a heritage program, consisting of proven methodologies, the GOES 8 pointing mirror was tested using a Leica Heerbrugg T3000

theodolite. This is a high precision piece of testing equipment capable of measuring angular positions to an accuracy of +/- 3 micro radians (0.62 arc seconds) (Ref. 14).

Another critical factor involving accuracy pertains to alignment of the GIFTS instrument with respect to the satellite structure. Therefore, monitoring alignment of the instrument coordinate system with respect to the spacecraft, during assembly, is crucial to system performance. Possible misalignments can occur due to vibration, thermal effects, and assembly tolerances. Variable settings for static and dynamic misalignments are include within the simulation and pointer profiler algorithm.

Attitude errors produced by the spacecraft ACS result in a platform motion that changes the payload's line-of-sight pointing. The instrument controller reacts to changes in platform location and produces correcting torques. The torque applied to the instrument has an equal and opposite affect on the spacecraft platform. The effects of erroneous attitude information are presented and discussed in Chapter 6 section 2 in the study involving star tracker errors.

5.1.2 Spacecraft Bus

The spacecraft bus is comprised of the ACS which consists of the inertial reference unit, star tracker, reaction wheels, and accompanying control laws. Disturbances present within the ACS are sensed with the gyro and then corrected with the reaction wheels. The control subsystem must counteract these disturbance torques by applying a corrective control torque. This requires the control system to have ample angular impulse capability to at least equal the

disturbance angular impulse. Proper sizing of the reaction wheels to accommodate varying pointing requirements has a large impact on the disturbance profiles of this component.

A significant disturbance source is due to the operation of the reaction wheels. The reaction wheel disturbances include static and dynamic imbalances. The wheel static and dynamic imbalance induce forces and torques proportional to the square of the wheel speed (ω_{RW}). The static imbalance is a result of the center of mass not coinciding with the wheel spin axis. A sinusoidal force variation at the wheel speed magnitude acting in a direction normal to the spin axis is inherent in these wheels (Ref 13). Dynamic imbalance is a consequence of the spin axis orientation differing from the principal axis of inertia. As a result, sinusoidal torques at the wheel speed frequency acting in a direction normal to the spin axis are produced. Spacecraft dynamic imbalance affects spacecraft attitude and reaction wheel control system through the principal mechanism of altering the nominal angular motions of the spacecraft. The reaction wheel static and dynamic disturbance specifications are given in Table 5.1. These values change throughout the course of operation because of component wear.

Table 5.1: Disturbance specifications of the RWA A-15 reaction wheel (Ref 11)

Feature	Value	Comments
Static Imbalance	**3.6×10^{-6} kg m @ Begin of Life**	**Twice this value @ end of life**
Dynamic Imbalance	**0.92×10^{-6} kg m^2 @ Begin of Life**	**Twice this value @ end of life**

The static force and dynamic torque disturbances are given mathematically as:

$$F_{Static\,X} = I_{Static}(\omega_{RW})^2 \cos(\omega_{RW} \cdot t) \qquad (4.9)$$
$$F_{Static\,Y} = I_{Static}(\omega_{RW})^2 \sin(\omega_{RW} \cdot t)$$

$$T_{Dynamic\,X} = -I_{Dynamic}(\omega_{RW})^2 \sin(\omega_{RW} \cdot t) \qquad (4.10)$$
$$T_{Dynamic\,Y} = I_{Dynamic}(\omega_{RW})^2 \cos(\omega_{RW} \cdot t)$$

In reality the actual vibration caused by the reaction wheel assembly is not limited to pure sinusoidal imbalance but covers a wide frequency band with multiple harmonics and wide band noise (Ref 13). The high frequency disturbances can typically be isolated with the use of damping mounts. Another reaction wheel disturbance is applied due to the wheels located a distance "*r*" from the center of mass of the spacecraft structure. This produces a torque $\tau = Fr$, where F is given by equation 4.9, which in turn produces angular acceleration $\alpha = \tau / I$ where I is the moment of inertia of the spacecraft. This other type of imbalance is given as a specification based on the assumed location of the spacecraft center of mass. In actuality this position changes throughout the course of the spacecraft life due to propellant expenditure. Therefore these disturbances will have a permanent effect on the performance of the attitude control system (ACS).

Another disturbance closely related to the operation of the ACS is sloshing of fluids. Propellant slosh may have a dramatic impact on pointing performance. Fluid in the tanks is used for the station-keeping thrusters. Fluid is also located in the cryo-coolers to provide thermal control for some of the science instruments. This type of disturbance can cause havoc within the control actuators and the sensing gyros, and must be minimized with either vibration absorbers or finely balanced wheels.

The gyros operating within the IRU are now described with respect to their contribution to system disturbances. The gyro drift rate is seen as an increase in angular error over time in deg/hr. Drift limits the accuracy of the instrument, but can be compensated within the attitude control software. The compensation is accomplished by using the gyros in conjunction with another attitude measuring device such as a star tracker.

Gyros are also subject to random walk errors. These errors are given in deg/sqrt(hr) and show the amount of noise measured on the angular rates and angles (Ref 9). Another error in the system is given by the gyro scale factor. This is an indication of the angular error that occurs during rotation. The scale factor can typically be measured by a calibration process and compensated. The simulation computes attitude (star tracker) and rate (gyro) measurements by corrupting the true signals with white noise, random rate, and walk noise. The star tracker also introduces attitude error to the system and is discussed in Section 5.2.

Solar array stepping perturbations on the spacecraft, induced from momentum exchange and possible structural excitation, may be a major disturbance source if actuated during science observations. Attempts are being made to perform the science measurements between periods of solar array actuation. However, the logistics of operation are presented just in case this is not possible.

In geosynchronous/geostationary orbits there are always two symmetrical solar panels with respect to the spacecraft in order to minimize the disturbance torque balance on the spacecraft caused by solar pressure on the panels. One panel is directed toward the positive orbit normal direction and the other is directed toward the negative orbit normal.

Stepping motor disturbance profiles and their effects on pointing performance were investigated. The stepping profile was based on a GOES mission case study in which the panels were incremented 0.007° every 1.69 seconds (Ref. 10). A second profile utilizing a double step option was also suggested in this reference to deadbeat, or perfectly track after a finite number of steps, the solar array oscillations. The aim here is to use alternate stepping configurations to mitigate jitter from the solar arrays.

Solar array orientation plays an important role in disturbance traceability. If an array is positioned such that its panels are parallel to the plane formed by the roll and pitch axes, its transverse vibration will result in vibrations about the spacecraft roll axis (Ref 15). However, if the array is rotated 90 degrees about its drive, with the panel oriented parallel to the yaw and pitch axis, then the vibration takes place about the spacecraft yaw axis. These changes take place every quarter orbit due to the fact that the solar array goes through a full rotation every orbit.

Many of the spacecraft disturbances have fixed points of application and direction that produce a constant disturbance profile that propagates through the system. The solar array response varies with respect to the spacecraft orientation. Therefore a large number of frequencies will be affected by this cyclical solar array positioning due to the many possible configurations.

The effects of high frequency vibration on the optical path are minimized with the use of appropriate absorbing material between the spacecraft and instrument, while the Pointing Control Mirror eliminates low frequency vibrations with the use of a feedback control system.

Other payload disturbances

The torque's produced by other payloads have no direct effect on the other instruments except that they perturb the spacecraft platform. This may be a problem if the disturbances are at such a rate that the spacecraft ACS cannot adequately correct for them. However, within a specified range, the only dynamic coupling between payloads is through the torques that are applied to the instrument gimbals and passed through the spacecraft ACS. A body-fixed reaction wheel attitude control system can be designed so that the dynamic coupling contributes as little as 0.03 to 0.04 degrees to the total pointing error (Ref 12).

5.2 System Error Analysis

This section discusses the error allocation trees that were assembled for the full system. The errors for individual sources are estimated either through testing or actual on-orbit operation. A quantitative value is then assigned to the component describing its performance characteristics.

Another source of errors within the system is introduced when transferring of data from one component to another. If the sampling rate of the signal is not sufficiently fast then the quality of the signal is degraded. If a signal is band-limited with samples taken relatively close together, in relation to the highest frequency present in the signal, then the samples can effectively replicate the true signal.

The exclusion of ephemeris errors from the analysis limits the validation process to the pointing stability and jitter requirements. Requirements such as geolocation knowledge and geolocation accuracy are dependant on the quality of the spacecraft ephemeris data. It is noted

that the ephemeris errors influence the pointing accuracy differently depending on the location of the area of interest. Pointing relative to local vertical is less sensitive to ephemeris errors than pointing to landmarks near the limb of the Earth. Furthermore, space-pointing modes to targets defined on the celestial sphere are independent of ephemeris (Ref 9). This is due to the attitude of the spacecraft having a greater importance than its actual position.
The pointing system performance is considered with reference to the inertial J2000 frame.

Error budgets

The error trees given in Figures 5.1 and 5.2 show the propagation of errors for the different components. GIFTS star tracker accuracies and IRU noise values are estimated to begin the sequence. Subsequent blocks process the attitude data and account for mechanical limitations. The values are root-sum-squared (RSS) to move up the error tree. The figures differ in the dynamic alignment uncertainty value.

Figure 5.1 shows the pointing stability for a single FOV scan duration of up to 10 seconds. The dynamic alignment uncertainty value of 0.62 arc seconds is representative of the small inaccuracies present for this short duration. The instrument and spacecraft pointing stability are then RSS'd to establish the overall system objective of 2.3 arc seconds.

Pointing Stability
For 10 sec @ nadir
For frequencies < 10 Hz
Values are RSS

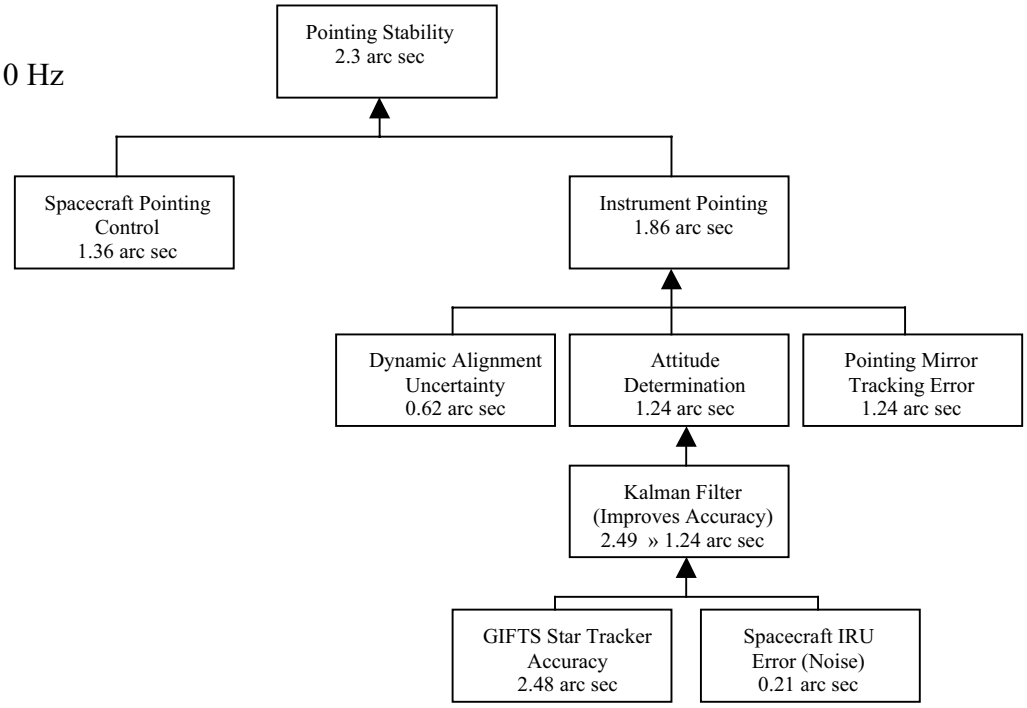


Figure 5.1: Pointing stability error tree

Figure 5.2 shows the relative frame-to-frame pointing knowledge requirement. The 30-minute duration given for this requirement comes from the time it takes to complete a global scan and then repeat the sequence. A dynamic alignment uncertainty of 4.66 arc seconds is the estimate used to represent the thermal, mechanical, and other outside disturbances to the pointing capability. An RSS value of 5.16 arc seconds defines the limit for this mode of operation.

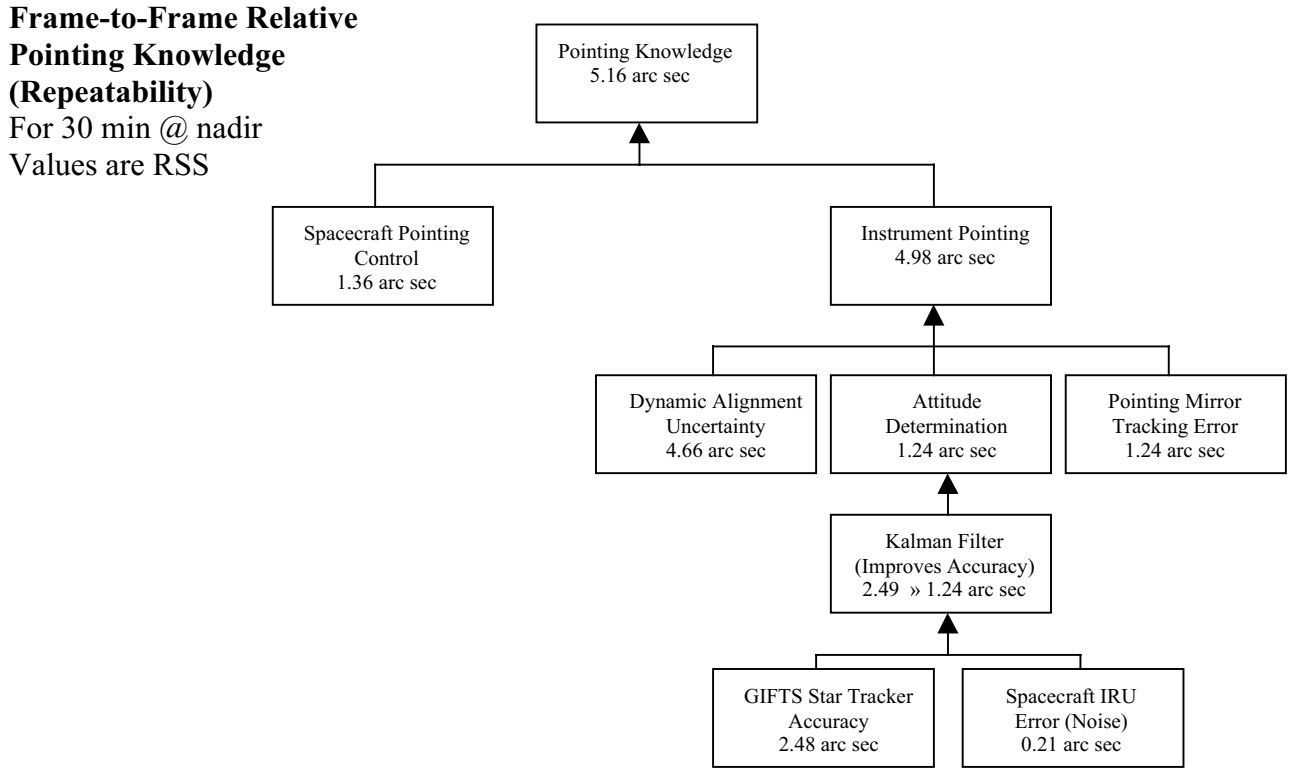


Figure 5.2: Pointing knowledge error tree

The analysis presented in this chapter leads to a better comprehension of the scope and magnitude that disturbances have on instrument operation. Analysis of the individual spacecraft disturbances led to isolation of component disturbance profiles. Optimal parameter values and parameter sensitivities have been found to vary significantly at different noise levels. Most of the pointing errors for the spacecraft are due to earth sensor inaccuracy, instrument pointing system misalignments, and thermal deformations, which are not accounted for in this simulation. Therefore it makes little sense to demand further improvement of the attitude control system unless significant advances can be made in these other areas of disturbance.

In an effort to bound the system (spacecraft and instrument) pointing errors, it is customary to allocate pointing error budgets to each component. However, assembly of a full system simulation allows for the performance of the entire system be defined by the performance of the GIFTS instrument pointing mirror control system.

Chapter 6. System Performance Analysis

In this chapter, the simulation data are presented and discussed. The first section outlines the analysis method used to determine and validate the system pointing stability and jitter requirements. Next, the simulation results for the star tracker accuracy study are presented. Finally, an analysis is performed to investigate the cumulative error effect within the model. This last study is accomplished by adding the disturbance models for the components such as, gyro, reaction wheels, and solar array stepping to the baseline simulation model. Discussions of the results follow, with an emphasis on the behavior each component has on the end performance of the system.

Discussed are the results of several performance studies carried out with the simulation. However, actual parameter values are not available for several components. Educated assumptions have been made for values pertaining to some of the spacecraft hardware. Realistic values for reaction wheel size and magnitudes for static/dynamic imbalance have been selected. In addition, IRU gyro dynamics values for damping, bandwidth, random walk, and random rate noise were obtained from spacecraft with similar mission requirements. The purpose is to begin performance validations of the individual components and their operation within the full system configuration. Once these values are properly defined, they can be applied to this simulation for further evaluation.

6.1 Pointing Stability & Jitter – Case 1.

This section discusses validation of the mission objectives and provides a foundation from which the pointing requirements are assessed. The objective is to show that the controllers and subsystems used can maintain the tracking and pointing of the instrument line-of-sight within science requirement accuracy in the presence of expected payload and spacecraft disturbances.

The pointing algorithm developed thus far meets the requirements set for pattern development and scanning. However, tests are not complete on the line-of-sight stability and jitter performance. This section contains the analysis of some of the interactions between all of the components in the system. The mission requirements are outlined first, and then results utilizing the current simulation parameters are given.

The science portion of the mission requires the spacecraft to accurately maintain a pointing direction at nadir to allow the instrument to target and track specific coordinates on the earth. Specific pointing stability and jitter limits are imposed on the system to assure the science requirements for tracking are met and to avoid smearing of data. For the purpose of the stability and jitter analysis a single FOV positioned directed at nadir is used. The end system requirements with respect to the GIFTS instrument are given in table 6.1.

Table 6.1: GIFTS requirements

Description	Requirement	Frequency	Duration	Location
Pointing Stability	2.3 arcsec	< 10 Hz	10 sec	@ Nadir
Pointing Jitter	2.3 arcsec	≥ 10 Hz	10 sec	@ Nadir

These requirements are derived from the spacecraft pointing capabilities and the GIFTS pointing capabilities. The spacecraft is allocated ± 1.4 arc seconds (1σ) for each axis. The GIFTS instrument is allocated 1.86 arc seconds (1σ) for each axis over a 10 second instrument scan period. This gives a total root-sum-square (RSS) value of 2.3 arc seconds, as seen in Figure 5.1. However, by constructing a full system simulation the spacecraft pointing performance is part of the performance values obtained for the GIFTS instrument. The spacecraft pointing performance values are shown to indicate the degree to which the spacecraft platform can be disturbed without affecting the pointing performance of the instrument. Because the star tracker is located on the same platform as the GIFTS instrument and pointing mirror, this co-location of the sensor and actuator provides an isolated system that is less sensitive to spacecraft motion to within the capabilities of the pointing mirror gimbal controller.

Simulations were performed to study the requirements and allocations above independently and examine the contributing factors involved in meeting these performance values. The simulations used the component parameters for noise previously discussed to establish a baseline operational performance value. This includes reaction wheel noise, gyro noise and a star tracker error of 2 arc seconds per axis. A 10 second portion of the spacecraft attitude angle error data, sampled at 1000 Hz, is shown in Figure 6.1. The left column of plots displays the spacecraft rate data about the roll, pitch, and yaw axes. The center column of plots displays the angle error with the standard deviation value. The standard deviation of the attitude angle is used as a measure of the 1σ error on the axis. The right column of plots in the figure presents the angle magnitude versus frequency. The magnitude versus frequency plot

shows the location of errors with respect to frequency disturbances. Similar results for the GIFTS instrument are given in Figure 6.2.

A listing of the error-per-axis performance is given in Table 6.2 and a RSS value computed to obtain a total system performance value. The values in Table 6.2 for the spacecraft performance are not observable by attitude sensors but are obtained from the dynamics equations to present an absolute positioning of the spacecraft platform.

Table 6.2: Summary of performance data

	x-axis (arcsec)	y-axis (arcsec)	z-axis (arcsec)		RSS (arcsec)
Spacecraft	0.015	0.031	0.014	=	0.037

	Azimuth (arcsec)	Elevation (arcsec)		RSS (arcsec)
GIFTS Inst.	1.725	0.938	=	1.963

		(arcsec)
Requirement Max. Allowable	=	2.3

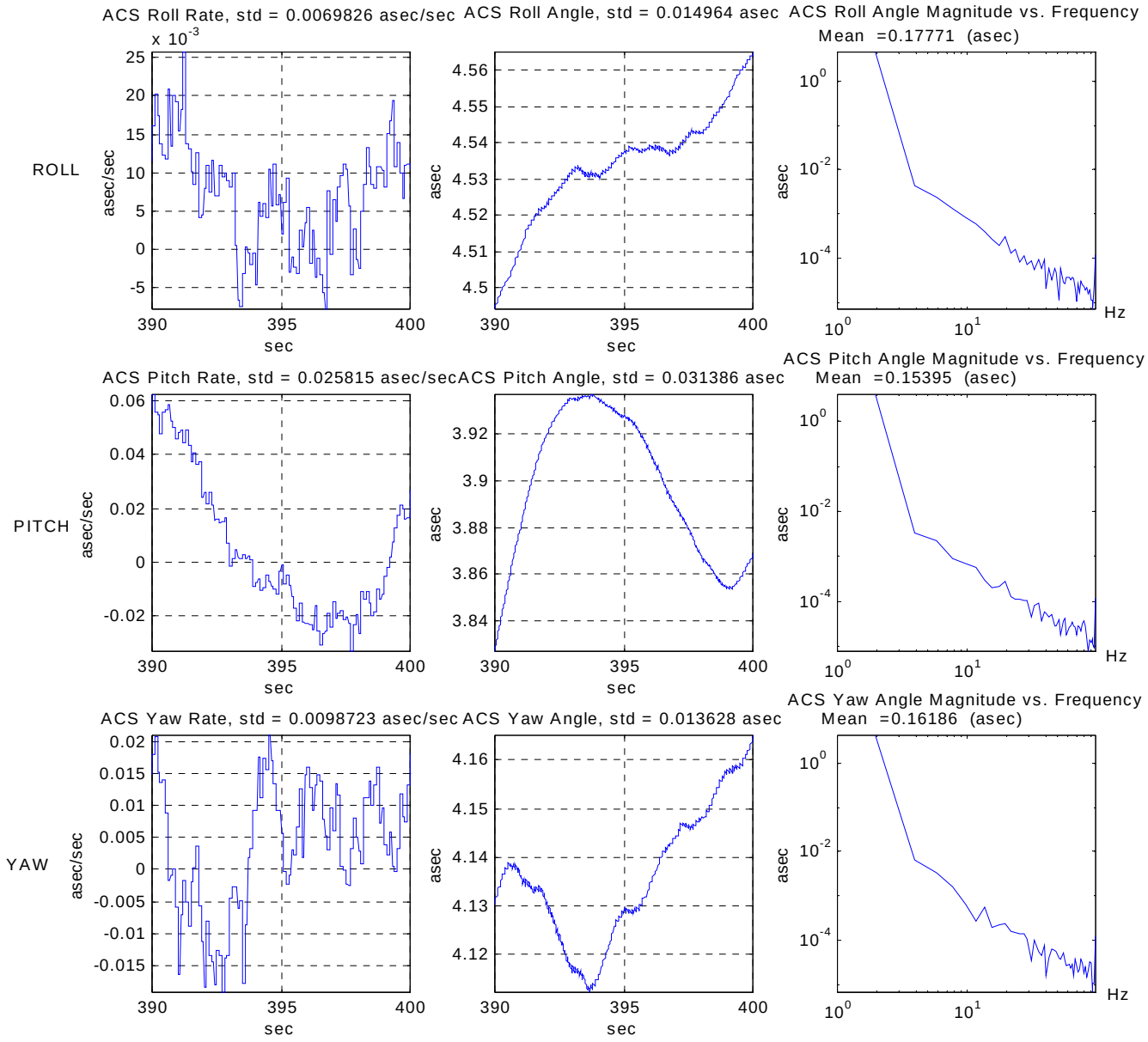


Figure 6.1: Spacecraft performance data

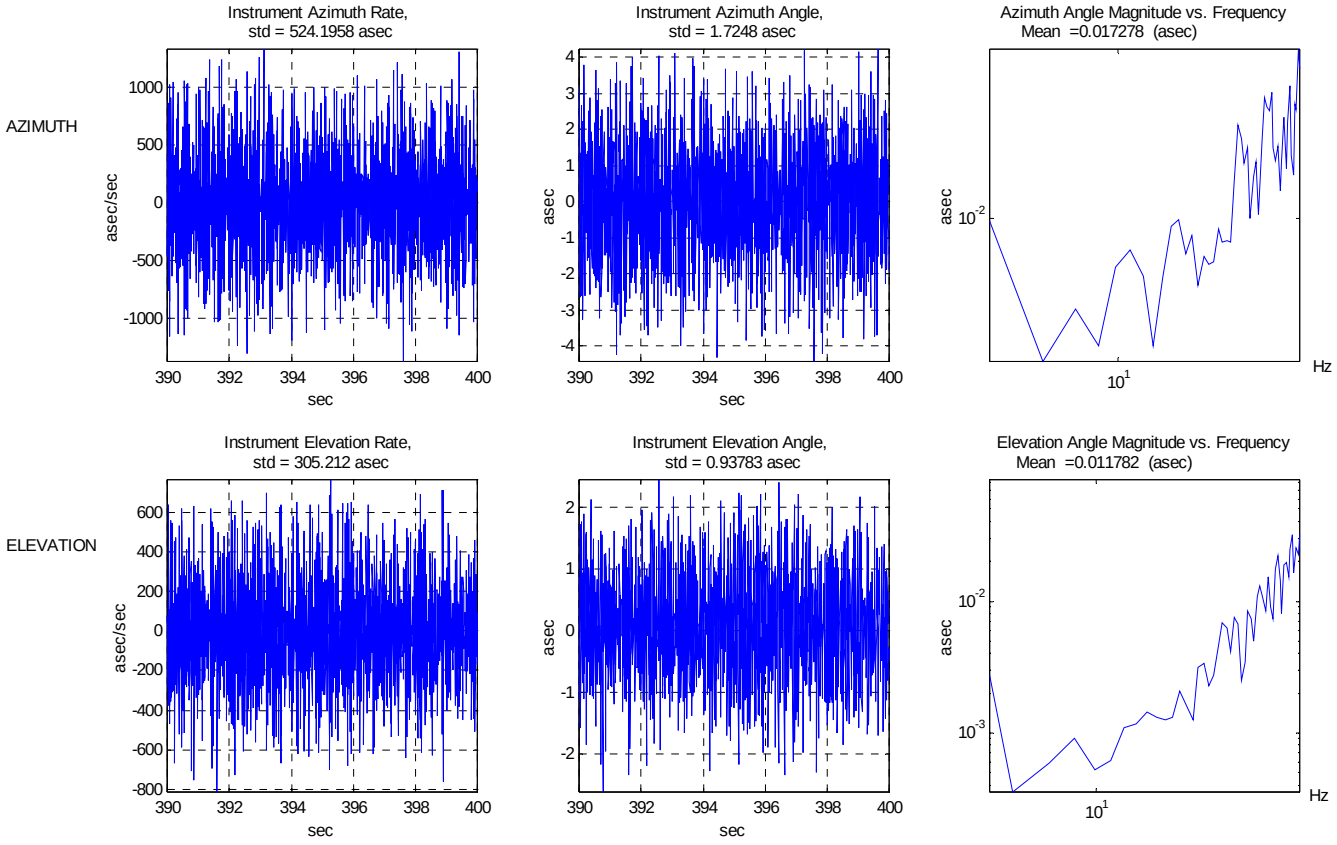


Figure 6.2: GIFTS instrument performance data

The performance value obtained meets the requirement of 2.3 arc seconds. Examination of the magnitude versus frequency data shows several spikes in the frequency. These spikes are due to reaction wheel noise and lie within the outlined requirements for stability and jitter. The pointing performance of the mirror gimbals is linked directly to the accuracy of the star tracker attitude data. The spacecraft disturbances are well within the operational bandwidth of 70 Hz given by the pointing mirror controller. Therefore the current values that were selected for the component disturbances are within the limits of satisfactory operation.

6.2 Star Tracker Accuracy – Case 2.

Since the pointing mirror performance is strongly dependant on the star tracker it was decided to further investigate its performance characteristics. A parametric study on the GIFTS star tracker has been performed to establish operating limits for the GIFTS instrument. To accomplish this task, simulations were run for 6 different cases. The only parameter that was altered was the star tracker accuracy. The accuracy of the star tracker is defined as the angle error obtained from the difference between the true attitude angles and the estimated attitude angles. The star tracker model simulates attitude angular data that has inaccuracies in the angle measurements ranging from 1-arc second to 6 arc seconds, per axis. The data were acquired from a 10 second time duration.

A listing of the error-per-axis performance is given in Table 6.3 and a RSS value computed to obtain a total system performance value. The instrument RSS values provided in Table 6.3 shows the degradation of the system performance as the star tracker attitude data becomes less accurate.

Table 6.3: Summary of RSS errors

Star Tracker Error (arcsec)	S/C x-axis (arcsec)	S/C y-axis (arcsec)	S/C z-axis (arcsec)	S/C RSS (arcsec)		Instrument Azimuth (arcsec)	Instrument Elevation (arcsec)	Instrument RSS (arcsec)
1	0.022	0.021	0.019	0.036		0.877	0.469	0.995
2	0.015	0.031	0.014	0.037		1.725	0.938	1.963
3	0.010	0.058	0.015	0.060		2.537	1.407	2.901
4	0.010	0.086	0.022	0.089		3.423	1.878	3.904
5	0.014	0.115	0.031	0.120		4.362	2.346	4.953
6	0.021	0.145	0.041	0.152		5.330	2.816	6.028

Table 6.3 displays the GIFTS star tracker performance data. Star tracker accuracies below 2.0 arc seconds meet the specified requirements. A steady increase in the pointing accuracy value highlights the importance of the star tracker data. The star tracker errors have a larger influence on the pointing accuracy performance of the GIFTS instrument than on the spacecraft platform. This is due to the direct link between the quaternion attitude information, and the assembly of the pointing commands in an open-loop manner. The spacecraft is not influenced as much by the star tracker angle information because of the added information provided by the spacecraft gyros. The data acquired provides a firm basis for utilizing a Kalman Filter when the star tracker accuracy value is above 2.00 arc seconds.

6.3 Cumulative Component Disturbances – Case 3.

The level of component disturbance effect has been independently investigated and verified against prescribed disturbance values. This analysis was used to quantify the influence of the component disturbance profiles within the make-up of the full system simulation. The level of susceptibility of the entire system to each of the component disturbances was also analyzed to determine the impact of that component. This analysis was carried out with a nominally operating baseline system with the component under investigation being added cumulatively to the simulation run. The table below displays the results of this study. Cases 3 through 4 are plotted in Figure 6.4 to show the torque levels for the pointing mirror gimbals.

Table 6.4: Summary of RSS performance data

Case #	Disturbance Component	S/C RSS (arcsec)	Instrument RSS (arcsec)
1	Baseline	0.0	0.009
2	Baseline + Static Imbalance + Gyro Noise and Drift	1.162	1.975
3	Baseline + Static Imbalance + Gyro Noise and Drift + Dynamic & Static RW	1.367	1.975
4	Baseline + Static Imbalance + Gyro Noise and Drift + Dynamic & Static RW + Solar Array Stepping	31.548	1.975

Figure 6.4 displays the effects of the component disturbance profiles. The elevation gimbal was found to be the worst case due to it rotating about the spacecraft y-axis or pitch. The corrective torques required to maintain nadir pointing are plotted versus the simulation time. The case number designations coincide with those in Table 6.4. Case 2 includes the gyro random walk and random noise along with a static imbalance on the spacecraft body. The pointing performance value of 1.975 arc seconds is maintained in the presence of the disturbances. A mean baseline torque value of –1.238e-6 (Nm) is required for case1. However, an increase in the pointing mirror gimbal torque is required to compensate for the added disturbances within the system. The increase in torque from the baseline case is -2.563e-8 (Nm) and the 70 Hz bandwidth of the gimbal control system can easily compensate for the effects of the disturbances.

Case 3 includes the gyro random walk and rate noise, the static imbalance on the spacecraft body, along with static and dynamic imbalances of the reaction wheels. The torque values

presented for case 3 show a slight increase when compared to those of case 2. The increase is within the capabilities of the pointing mirror controller and does not approach the torque limits placed on the gimbal system. The mean torque value increase over case 2 is –3.043e-8 (Nm). Proper sizing and profiling of these disturbances might increase this value.

Case 4 includes all the disturbances of case 3 with the addition of a solar array stepping disturbance. The profile of this disturbance is discussed in section 5.1.2. The addition of the stepping disturbance has the effect of dramatically increasing the torque on the pointing mirror gimbals by a mean value of –4.764e-4 (Nm). The drift rate shown in case 4 is possibly due to a cyclical variation of the solar array movement that is beyond the capabilities of the control system. However, a more accurate profile of the solar array disturbance is required to properly analyze the impact placed on the full system. The data presented in Table 6.4 and Figure 6.4 show some of the capabilities of the coupled spacecraft-instrument system.

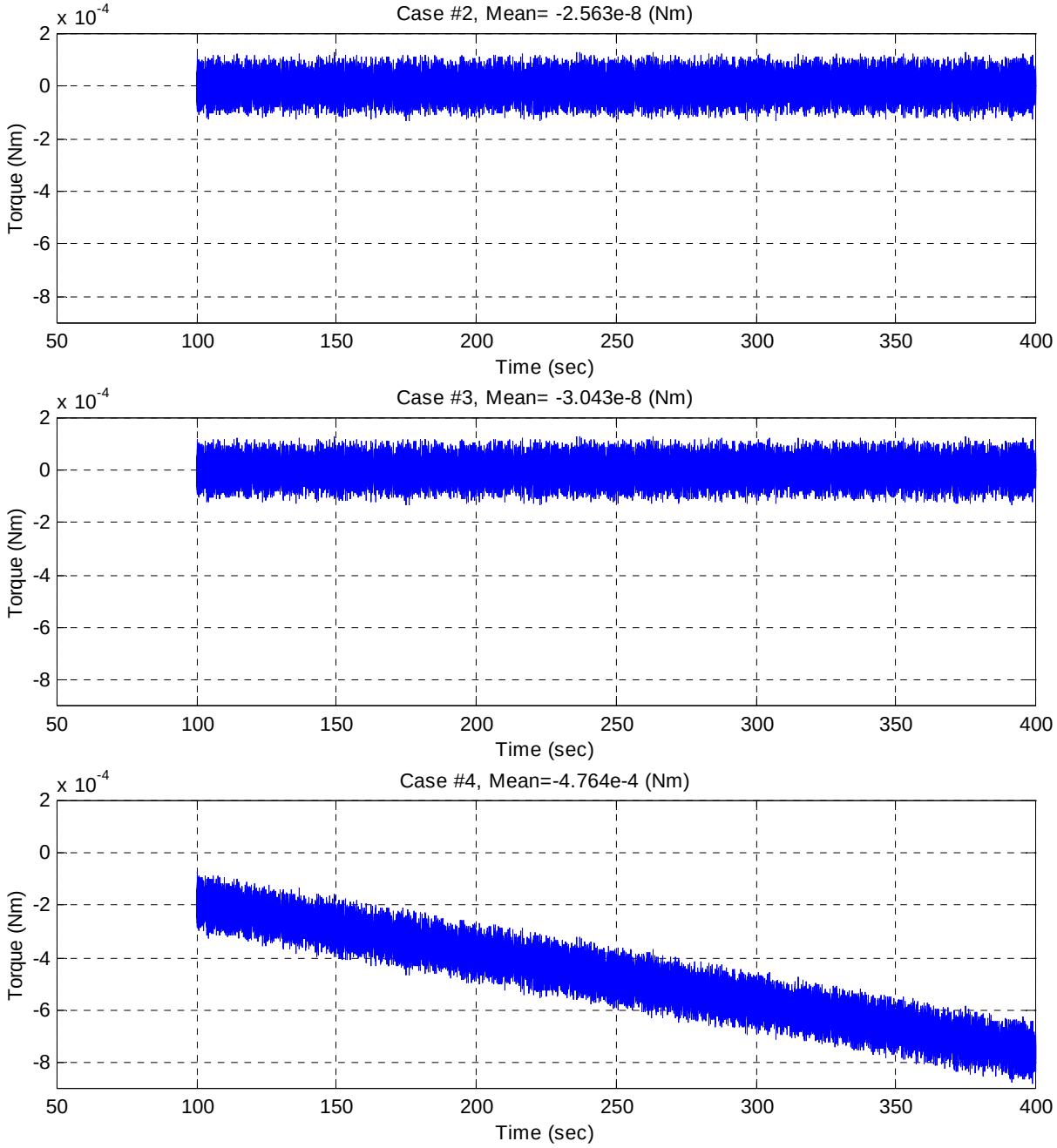


Figure 6.4: Elevation gimbal torque performance for cases 2, 3, and 4

The limiting factors of this system are given by the bandwidth of the gimbal controller for providing sufficient disturbance compensation and the torque capabilities of the gimbal actuators. These parameters are critical to the proper operation of a pointing mirror in the presence of spacecraft disturbances. However, the linearized model currently used to perform the analysis disregards the non-linear effects of friction drag and large angle movements.

A conclusion to be drawn from this cumulative analysis lies in the fact that although the spacecraft performance was degrading the GIFTS instrument pointing system has sufficient control authority to maintain an accurate pointing direction. However, a more thorough investigation is needed to quantify the disturbance magnitudes from each of these separate components. Pointing performance of the GIFTS instrument is mostly a function of star tracker performance provided that the disturbances in the spacecraft are within the bandwidth of the mirror controller.

Chapter 7. Concluding Remarks

The final chapter presents a summary of the results and conclusions obtained throughout the course of this research. First, a summary covering the main topics of each chapter is given. Next, the conclusions of the research are presented. Finally, recommendations for future research are given.

7.1 Summary of Results

The first two chapters present the background information essential to developing the purpose and objective for this research. The purpose being the preliminary design and analysis of the GIFTS instrument pointing algorithm. The heritage behind the current interferometer capabilities was also presented to establish a motivation for continued development in this area of science. A proof-of-concept plan was also discussed to highlight the objectives for this research. Upon assembly of this preliminary algorithm a series of studies were performed to explore the performance capabilities of the full system. From these studies a methodology has been established from which further design iterations can be easily investigated to enhance the capabilities of the mission.

Presented in Chapter 2 are some of the operational specifications of the GIFTS instrument pointing system. Modes of operation and scan pattern options are discussed to provide a functional view of how the instrument will perform its science mission. Requirements for mission success are also presented to provide system metrics.

The development of the simulation tool used to model the GIFTS instrument pointing system is presented in Chapter 3. The development of the pointing system model used to compute the pointing mirror gimbal commands is given. A major portion of the research effort revolved around the development of the GIFTS instrument-pointing algorithm. The models for pointer profiler, mirror dynamics, and controller are introduced and discussed. The introduction of a coupling term to the combined spacecraft-instrument system was required to model the dynamic interaction of the two control systems. To ensure that the performance requirements are achieved, accurate models of all the different components of the full system model are needed.

A description of the spacecraft model is given in Chapter 4. The model serves as a platform for validation of the pointing algorithm and the associated pointing mirror gimbal control system. Dynamics equations and error implementation techniques were presented to establish capabilities and limitations of the current model.

Chapter 5 presents a first order estimate of the pointing performance disturbances and error sources with the spacecraft-instrument structure assumed to be a rigid body. Neglecting the flexible body motion has the effect of reducing jitter information. A model using these more complex flexible body dynamics would surely show an increase in the system jitter.

Presented in Chapter 6 are some results using the simulation as a design tool. The numerical results generated from the simulation model are presented and discussed. A short discussion on the applicability of the studies is presented followed by a discussion of the tendencies of

the system used in this study. Since the instrument design is not complete, many design variables are either not available or have changed throughout the course of development. A thorough understanding of how the model fidelity evolves, as the design improves, is key to properly interpreting the analysis carried out at the different stages.

The results from three case studies are presented in Chapter 6. Results generated by the simulation model using the aforementioned component parameters are presented and compared to the mission requirements. It was found that the spacecraft ACS does not have the control bandwidth to attenuate all the effects of vibrations emanating from the sources such as the reaction wheel assembly, solar array stepper motor, or other possible disturbance sources and noise levels assumed in the study. However, the pointing mirror bandwidth is large enough to correct for disturbances emanating from the spacecraft to a sufficient degree based on current design settings.

7.2 Conclusions

Presented here is a description of the GIFTS instrument pointing system, various components of the spacecraft system, a set of requirements outlining the pointing quality for adequate science return, and three case studies with results of the overall performance of the system.

The driving factors in this research are the need to develop and analyze the pointing capabilities of the GIFTS instrument pointing system and to validate the operation of the pointing algorithm. Although several modifications are due to be implemented in the operational methodology, a considerable amount of work has been performed to begin system

development. However, before such a final study can be conducted, detailed information about the actual performance characteristics of spacecraft-instrument system must be understood. Among the objectives of this investigation are to measure the effective performance of the GIFTS pointing system, construct a simulation model to describe the dynamics, determine acceptable disturbance levels for various components, and implement a methodology to carry out performance studies as the project develops. Overall, this work provided a development tool to the GIFTS project to aid designers in fulfilling mission requirements.

The analysis of the simulation model performance presented in Chapter 6 is important for understanding the validity of the disturbance profiles used in the model. An instrument pointing stability and jitter limit of 2.3 arc seconds indicates that the design does adequately isolate the instrument optics from the disturbances caused by the reaction wheels. GIFTS instrument stability and spacecraft attitude performance had no significant system instabilities resulting from control system interactions. Also highlighted is the range of operability for the star tracker performance. Direct coupling between the pointer profiler command and the star tracker highlights the dependency of the system on the star tracker performance.

An accurate simulation has been developed from which further case studies can be performed on a large range on spacecraft-instrument parameters. It was also found that geosynchronous instrument pointing satellites are complex vehicles, with stringent performance requirements, which must be fulfilled with other concurrent design constraints. Several applications of the simulation include performance verification, evaluation of sensor noise effects, and

determination of parameter sensitivities and optimal parameter values. In addition, the rigid body results obtained can be used to determine the operating bandwidth of the system. The bandwidth can then be used to determine stiffness requirements for the structural designers, which will set a threshold to bring the fundamental frequency above this bandwidth.

7.3 Future Work

The research in this thesis was limited to the investigation of GIFTS instrument pointing system effects with respect to a rigid body spacecraft model. Upon receiving a spacecraft model of adequate fidelity a full system disturbance analysis should be reapplied. It is recommended that future work include a model of higher fidelity with information pertaining to system flexible body motion. The information from the disturbance analyses could be used to correct for any unforeseen interactions and improve the performance of the GIFTS instrument.

Future work should include different disturbance profiles for reaction wheel assemblies and articulating solar arrays. In addition, adding passive isolation to the reaction wheel mounts will attenuate adverse forces and torques emanating from the torque wheels.

A method for updating the component performance characteristics must continue in order to maintain a working analytical model for use with future research endeavors. Finally, the work on the GIFTS project is still progressing. As the project evolves, periodic disturbance studies should be performed on the full system to assure mission success.

References

1. NASA Goddard, "EO1 New Millennium Project," http://eo1.gsfc.nasa.gov

2. JPL "Deep Space 1 New Technology Demonstrator," web site (http://nmp.jpl.nasa.gov/ds1/.)

3. JPL, "Liquid Crystal Tunable Filter Imaging Spectrometer," web site (http://ddl.jpl.nasa.gov/lctf/rtf/)

4. NASA Langley, "Earth Science," http://www.larc.nasa.gov/research/inside_pages/earthscience.htm

5. David A. Vallado, "Fundamentals of Astrodynamics and Applications," 2nd Edition, Microcosm Press 2001

6. James R. Wertz, Wiley J. Larson, "Space Mission Analysis and Design," Microcosm Press and Kluwer Academic Publishers, 1999 Third Ed.

7. Benjamin George, "Changes to GIFTS Compute Pattern Algorithm," George Washington University, Jan 2001

8. Chauncey Uphoff, "Description of Algorithms for GIFTS," Fortune Eight Aerospace Industries Inc., Dec 2000

9. "Precision Pointing Control System," TRW Systems Group, NASA-CR-139132, July 1, 1972

10. S. K. Tadikonda and S. A. Cauffman, "Understanding GOES-8 Dynamics," *The International Society for Optical Engineering (SPIE),* Vol. 2812, No. 64, August 7-9, 1996, Denver Colorado

11. M. C. Algrain and R. M. Powers, “Line-of-Sight Pointing Accuracy-Stability Analysis and Computer Simulation for Small Spacecraft,” *The International Society for Optical Engineering,* Vol. 2739, No. 06, April 1996

12. R. P. Iwens, A. W. Fleming and V. A. Spector, “Precision Attitude Control with a single body-fixed momentum wheel,” AIAA Mechanics and Control of Flight Conference 1974, AIAA Paper No. 74-894

13. Eric Ponslet, “System Level Modeling of the SNAP Instrument and Analysis of Reaction-Wheel-Induced Jitter,” Hytec Inc., Dec. 2000, web site, http://snap.lbl.gov/pubdocs/hytec_jitter.pdf

14. Ramuhalli Krishna, “Improved Pointing Accuracy Using High Precision Theodolite Measurements,” *The International Society for Optical Engineering,* Vol. 2812, Pg. 199, August 1996

15. S. E. Woodard, “Orbital and Configuration Influences on Spacecraft Dynamic Responses,” *Journal of Spacecraft and Rockets*, Vol. 35, Num. 2, March-April 1998, Pg. 177-182

Appendix A. GIFTS Instrument Pointing Algorithm Code

The contents of Appendix A. constitute the Matlab code for the *Pointer Profiler* module of the full system simulation. Figure A.1 on the following page is a block diagram flow chart of the subroutines contained within this appendix. The Matlab code is accessed by the Simulink simulation to calculate the pointing commands for the pointing mirror controller.

Full system initialization parameters are included in this Appendix along with the *Pointer Profiler* initialization parameters. These parameters give initial conditions for components within the system such as gyro characteristics, reaction wheel settings, and pointing mirror controller settings.

The Event Manager subroutine is used as an interface to all the subroutines contained in Figure A.1. Its purpose is to allocate data to specific subroutines based on a timing sequence described in section 3.3.2. Another sequence dependent on pattern dimensions is established by the *COMPAT* module to step through the pattern of FOV's in a contiguous manner.

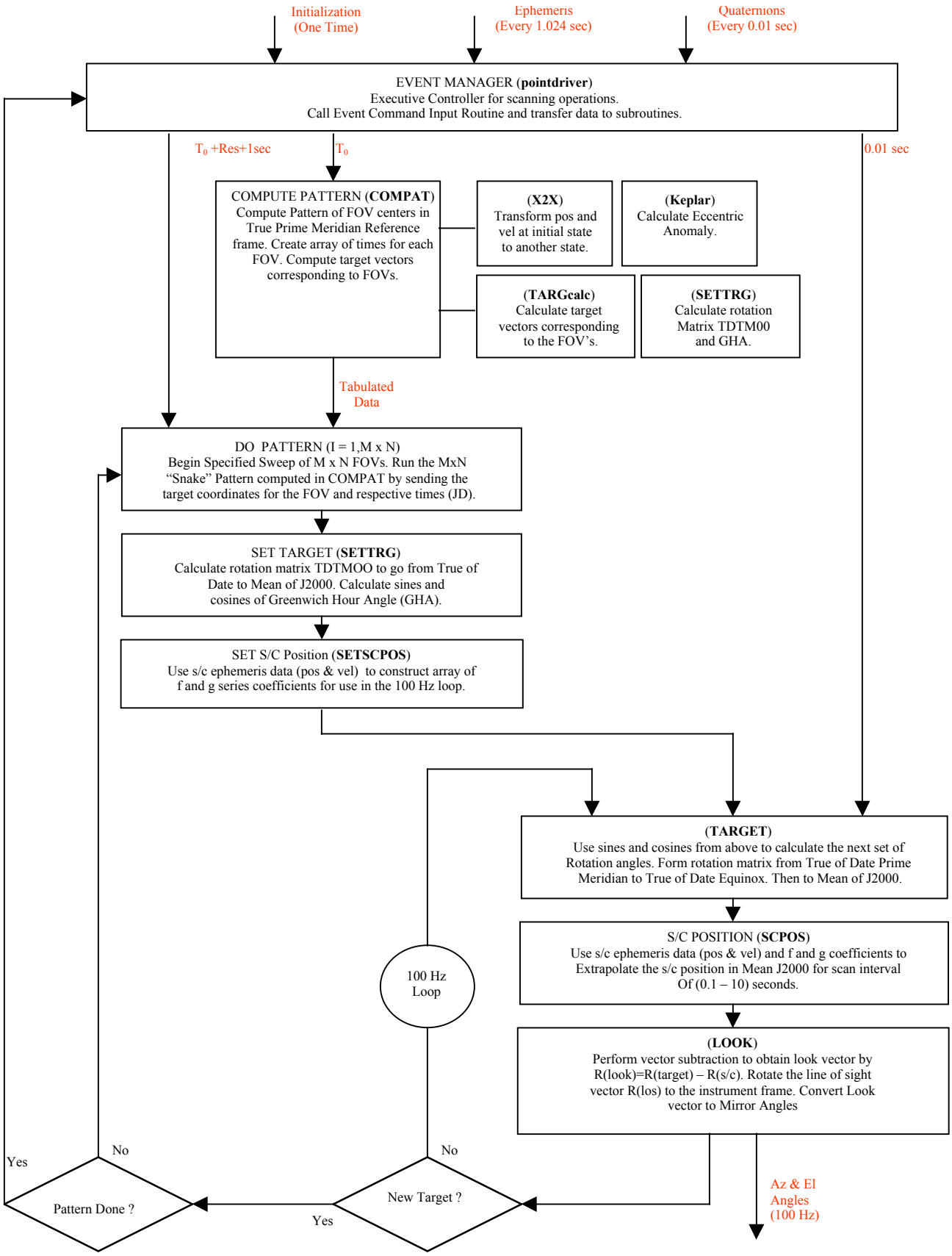


Figure A.1: GIFTS Instrument Pointing Algorithm Block Diagram

```
function [DriverOut] =PointDriver(DriverIn)
% [RtargM00] =TARGETsim(TARGETin)
% This Pointer Driver runs the pointer control system, accesses all of the subfunctions,
% and outputs the angles to the mirror controller
% Input: DriverIn : 12x1 array consisting of
%         Tsim: simulation time in seconds (1)
%         Rsc: Spacecraft position ephemeris (2-4)
%         Vsc: Spacecraft velocity ephemeris (5-7)
%         Teph: Ephemeris time tag - Simulation time in Seconds (8)
%         q2000: Quaternion information from the star tracker (9-12)
%         patcen: The pattern center in geodetic lat and long (rad) and alt (km)(13-15)
%         Dim - Dimension of Pattern - MxN - Rows x Columns (16-17)
%         Resolution - Resolution of FOV (18)
%         Overlap - Overlap in pixels (19)
%         PATstart - Pattern start time - Simulation time (seconds) (20)
% Output        DriverOut : 5x1 array consisting of
%         angles: Mirror Angles - Azimuth, elevation (1-2)
%         Rtarg : Target Vector (3-5)
%Updated 21 Feb 01

global RtargArray RtargArray_ul RtargArray_ur RtargArray_ll RtargArray_lr TargTime
Patind TephStore TDTM00 cGHA sGHA
global RscStore VscStore f2 f3 g3 EphFlag SimStartJD FOVdur PatEndFlag


dr=pi/180;
Tsim=DriverIn(1);
Rsc=DriverIn(2:4);
Vsc=DriverIn(5:7);
Teph=DriverIn(8);
q2000=DriverIn(9:12);
patcen=DriverIn(13:15);
Dim=DriverIn(16:17);
Resolution=DriverIn(18);
Overlap=DriverIn(19);
PATstart=DriverIn(20);
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Compute the Pattern array%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%if new pattern
if isempty(TargTime)|(PATstart~=TargTime(1)&PatEndFlag==2)
   SimStartJD=2451702.31; %Define Julian Date of Simulation start

[RtargArray,RtargArray_ul,RtargArray_ur,RtargArray_ll,RtargArray_lr,TargTime]=CO
MPAT(patcen,Dim,Resolution,Overlap,PATstart,Rsc,Vsc,Teph,SimStartJD);
   %initialize for first point on target
```

```
  FOVdur=Resolution; %Calculate duration of scan for each FOV
  Patind=1; %Set index to first point in pattern
  JD=SimStartJD+TargTime(Patind)/86400; %Calculate the julian date for the FOV
  [TDTM00,sGHA,cGHA,GHA] =SETTRG(JD);  %Initialize the rotation matrices for
the first point
  EphFlag=1; %Update the ephemeris
  PatEndFlag=1;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Target position propagation%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%if still on same scan
if Tsim>=TargTime(Patind)&Tsim<(TargTime(Patind)+FOVdur)
  EphFlag=2; %Do not update ephemeris
%if end of pattern repeat pattern
elseif TargTime(Patind)==TargTime(end)&Tsim==(TargTime(Patind)+FOVdur)
  Patind=1; %Set index to first point in pattern
  TargTime=TargTime+Tsim-TargTime(1); %Adjust pattern time to repeat
  JD=SimStartJD+TargTime(Patind)/86400; %Calculate the julian date for the FOV
  [TDTM00,sGHA,cGHA,GHA] =SETTRG(JD);  %Reinitialize the rotation matrices for
the first point
  EphFlag=1; %Update the ephemeris
  PatEndFlag=2; %Flag that the pattern had ended
%else if time for next FOV
elseif Tsim==(TargTime(Patind)+FOVdur)
  Patind=Patind+1; %Move to next FOV in Pattern
  JD=SimStartJD+TargTime(Patind)/86400; %Calculate the julian date for the FOV
  [TDTM00,sGHA,cGHA,GHA] =SETTRG(JD); %Reinitialize the rotation matrices for
the FOV
  EphFlag=1; %Update the ephemeris
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Ephemeris propagation%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%if start of program or (ephemeris has been updated and not in middle of scan)
if isempty(TephStore)|(Teph~=TephStore&EphFlag==1)
  TephStore=Teph; %Update the Ephemeris time
  [f2,f3,g3] =SETSCPOS(Rsc,Vsc); %Update the f and g series
  RscStore=Rsc; %update the ephemeris position
  VscStore=Vsc; %Update the ephemeris velocity
end

%Calculate the Target vector in J2000
```

```
[RtargM00,RtargM00_ul,RtargM00_ur,RtargM00_ll,RtargM00_lr]=TARGET(RtargArra
y(Patind,:),RtargArray_ul(Patind,:),RtargArray_ur(Patind,:),RtargArray_ll(Patind,:),Rtar
gArray_lr(Patind,:),TargTime(Patind),Tsim,TDTM00,sGHA,cGHA);
%Extrapolate the position of the spacecraft in J2000
[RscM00]=SCPOS(RscStore,VscStore,f2,f3,g3,TephStore,Tsim);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tempRtargM00=[];
%Store the target vectors in an array for plotting
tempRtargM00=[tempRtargM00
  RtargM00];

tempRtargM00_ul=[];
%Store the target vectors in an array for plotting
tempRtargM00_ul=[tempRtargM00_ul
  RtargM00_ul];

tempRtargM00_ur=[];
%Store the target vectors in an array for plotting
tempRtargM00_ur=[tempRtargM00_ur
  RtargM00_ur];
tempRtargM00_ll=[];
%Store the target vectors in an array for plotting
tempRtargM00_ll=[tempRtargM00_ll
  RtargM00_ll];
tempRtargM00_lr=[];
%Store the target vectors in an array for plotting
tempRtargM00_lr=[tempRtargM00_lr
  RtargM00_lr];
%%%%%%%%%%%%%%%%%%%%
%Angle Calculations%
%%%%%%%%%%%%%%%%%%%%
qst=[0 0 0 1];
% 1 deg in +x, 1 deg in -y, and 0 deg in z.
%qst =[-8.7262e-003 -7.6152e-005  8.7262e-003  9.9992e-001];
qdyn=[0 0 0 1];
[Rlosi,Azimuth,Elevation] = LOOK(RscM00,RtargM00,q2000,qst,qdyn);
DriverOut=[Azimuth
  Elevation
  tempRtargM00
  tempRtargM00_ul
  tempRtargM00_ur
  tempRtargM00_ll
  tempRtargM00_lr
  RscM00
  Rlosi];
```

```
function[RtargArray,RtargArray_ul,RtargArray_ur,RtargArray_ll,RtargArray_lr,TIME]=
COMPAT(patcen,Dim,Resolution,Overlap,PATstart,Rsc,Vsc,Teph,JDsim)
%[RtargArray,TIME]=COMPAT(patcen,Dim,Resolution,Overlap,JDstart,Rsc,Vsc,Teph)
%This function calculates the pattern of FOV centers in the True Prime Meridian of date
reference frame.
%Input: patcen - The pattern center in geodetic coordinates (3x1 array of lat,long,alt)(rad
and km)
%       Dim - Dimension of Pattern - MxN - Rows x Columns (2x1 Array)
%       Resolution - Resolution of FOV
%       Overlap - Overlap in pixels
%       PATstart - Pattern start time - Simulation time (sec)
%       Rsc and Vsc - position and velocity of the spacecraft
%       Teph - Time of ephemeris - Simulation time (sec)
%       JDsim - Julian date corrolating to the start of the simulation (sim time = 0)
%Output: RtargArray - Array of FOV centers in True Prime Meridian of Date reference
frame
%                   in cartesian coordinates
%        TIME - Array of start time associated with the FOV.

%Updated 12 Feb 01
%Separated the time array out of the output because of decision to use an m-file driver
%Updated 6 Feb 01
%Changed the times to be in simulation time with an added input of the julian date
% of the start of the simulation
%Updated 31 Jan 01

%Define Constants
Re=6378.14;  %radius of earth(km.)
f=1/298.25722; %Flattening effect on the earth
mu=398600.5; %GM of the earth (km^3/s^2)
dr=pi/180;  % degrees to radian conversion

%Initialize arrays
TIME=zeros(Dim(1)*Dim(2),1);
RtargArray=[];
RtargArray_ul=[];
RtargArray_ur=[];
RtargArray_ll=[];
RtargArray_lr=[];
PC=zeros(3,1);

Scantime = Resolution; %Scan time for each FOV will actually be a function of

%                   resolution but the exact correlation has not been given

lat=patcen(1);
```

```
lon=patcen(2);
alt=patcen(3);

% Convert Pattern Center from Geodetic to Cartesian Coordinates
cosp=cos(lat);
sinp=sin(lat);
C=(cosp^2+(1-f)^2*sinp^2)^(-1/2);
S=(1-f)^2*C;
PCxy=(Re*C+alt)*cosp;
%Pattern center in Cartesian coordinates
PC(1)=PCxy*cos(lon);
PC(2)=PCxy*sin(lon);
PC(3)=(Re*S+alt)*sinp;

%Create an array of times for each FOV
Slewtime=1; %Time for mirror to move from one FOV to another.
          %Slewtime will not be known until after testing
FOVtime=Scantime+Slewtime; %Total time (seconds) for each FOV
%Create the array of FOV times for a Snake Pattern
TIME(1)=PATstart;
PATTERN(1,:)=[1,1];

%  Time=TIME(count-1)+FOVtime;
%Calculate the offset angle
Offset = Overlap;  %Angle between the centers of adjacent FOVs (radians)
%                This will actually be a function of the Overlap
%Create the pattern
index=1;
  %extrapolate the position and velocity to the time of the FOV
  [RscM00,VscM00]=X2X(TIME(index),Teph,Rsc,Vsc,mu,1e-8,1);
  %Calculate the Target Vector for one Field of View
  FOVtime=TIME(index)/86400+JDsim;

[rtarg,rtarg_ul,rtarg_ur,rtarg_ll,rtarg_lr]=TARGcalc(PC,RscM00',VscM00',FOVtime,Di
m,PATTERN(index,:),Offset);
  %Store the target vectors in an array
  RtargArray=[RtargArray
    rtarg];
  RtargArray_ul=[RtargArray_ul
    rtarg_ul];
  RtargArray_ur=[RtargArray_ur
    rtarg_ur];
  RtargArray_ll=[RtargArray_ll
    rtarg_ll];
  RtargArray_lr=[RtargArray_lr
          rtarg_lr];
```

```
function E=Kepler(M,e,tol,ichk)
%E=Kepler(M,e,tol,ichk)
% This function calculates an array of eccentric anomalies, E,
% given an array of mean anomalies, M, eccentricity, and a tolerance
%Input: M - array of mean anomalies
%       e  - the eccentricity of the orbit 0<e<=1
%       tol  - tolerance for the accuracy of E
%       ichk - 1 for input checking other for no checking
%Output: E - An nx1 array of Eccentric Anomalies

if ichk==1                  % do size and range testing
 if min(size(M))>1, error('M must be an array size nx1'),end
 if max(size(e))>1,error('e must be a scalar'),end                  % inputs are scalars
 if max(size(tol))>1,error('tol must be a scalar'),end
 if e<0 | e>1, error('e must be between 0 and 1'), end
 if tol<=0, error('tol must be positive'), end
end      % if ichk==1

E1=zeros(size(M));
E=M+0.85*e*sign(sin(M));

ind = find(abs(E-E1)>tol);
while ind
   E1(ind)=E(ind)+(M(ind)-E(ind)+e*sin(E(ind)))./(1-e*cos(E(ind)));
   ind = find(abs(E-E1)>tol);
   E=E1;
End
```

```
function [r,rdot]=orb2x(t,OrbElem,gm,tol)
% function [r,rdot]=orb2x(t,OrbElem,gm,tol)
% orb2x.m returns the position and velocity vector at time t
% using orbital elements in OrbElem=[z,p,inc,node,omega,tau] array
% t can be a vector of times but the rest of the parameters must be scalars
% tol is convergence criteria for Keplers equation.

% check input
[lent,j]=size(t);
if j>1, error('input time array to orb2x must be a column array'),end
if (max(size(OrbElem)) > 6)|(min(size(OrbElem)) > 1),
 error('input to orb2x must be 6 orbit elements'),end
z=OrbElem(1); p=OrbElem(2); inc=OrbElem(3);
node=OrbElem(4); omega=OrbElem(5); tau=OrbElem(6);
if max(size(gm))>1,error('gm must be scalar in orb2x'),end
if max(size(tol))>1,error('tol must be scalar in orb2x'),end
if p<0,error('p must be non-negative in orb2x'),end
if gm<=0,error('gm must be positive in orb2x'),end
if tol<=eps,error('tol must be greater than eps in orb2x'),end
if (inc<0)|(inc>pi),error('0<=inc<=pi in orb2x'),end

twopi=2*pi; r=zeros(lent,3); rdot=r;   % initialize output arrays

      % for each type of orbit find theta=omega+f

if z > 0                          % ellipse, so solve for E
 n=sqrt(gm*z*z*z);                    % mean motion
 m=n*(t-tau);                       % mean anomaly
 j=find(m<0); k=length(j);            % find m<0
 while k>0                         % add 2pi until all m>0
  m(j)=m(j)+twopi;
  j=find(m<0); k=length(j);
 end   % while k>0
 j=find(m>=twopi); k=length(j);        % find all m>2*pi
 while k>0                         % subtract 2pi until all m<2pi
  m(j)=m(j)-twopi;
  j=find(m>=twopi); k=length(j);
 end   % while k>0
 ecc=sqrt(1-p*z);                  % eccentricity
 E=m+sign(sin(m))*0.85*ecc;       % first guess at eccentric anomaly
 del=2*tol;
 while del > tol
  cosE=cos(E); sinE=sin(E); d=1-ecc*cosE;
  delE=-(E-ecc*sinE-m)./d;
  E=E+delE;
  del=max(abs(delE));            % get max change in E
```

```
 end         % while del>tol
 rho=d/z;                      % radial distance
 rhodot=n/z/z*ecc*sinE./rho;      % radial velocity
 if p>0                        % get true anomaly for theta=omega+f
  cosf=(cosE-ecc)./d;
  sinf=sqrt(p*z)*sinE./d;
  f=atan2(sinf,cosf);
 else
  f=-pi*ones(t);                % degenerate case
  if inc ~= pi/2,error('degenerate ellipse must have inc=90 deg'),end
 end;    % if p>0
 theta=omega+f;                  % end elliptic case
% ----------------------------------------------------------------------
elseif z < 0                   % hyperbola, so solve for F
 n=sqrt(-gm*z*z*z);              % mean motion
 m=n*(t-tau);                  % mean anomaly
 ecc=sqrt(1-p*z);               % eccentricity
 F=asinh(m);                   % first guess at eccentric anomaly
 del=2*tol;
 while del > tol
  coshF=cosh(F); sinhF=sinh(F); d=ecc*coshF-1;
  delF=-(ecc*sinhF-F-m)./d;
  F=F+delF;
  del=max(abs(delF));
 end                           % while del>tol
 rho=-d/z;                      % radial distance
 rhodot=n/z/z*ecc*sinhF./rho;      % radial velocity
 if p>0                        % non-degenerate hyperbola
  cosf=(ecc-coshF)./d;
  sinf=sqrt(-p*z)*sinhF./d;
  f=atan2(sinf,cosf);
 else
  f=-pi*ones(t);                % degenerate case
  if inc ~= pi/2,error('degenerate hyperbola must have inc=90 deg'),end
 end;         % if p>0
 theta=omega+f;               % end hyperbolic case
% ----------------------------------------------------------------------
else                           % parabola, so solve Barker
 if p>0                         % non-degenerate parabola
  n=sqrt(gm/p/p/p);
  B=3*n*(t-tau);                % Battin pg 151
  A=(B+sqrt(1+B.*B)).^(2/3);
  tanf2=(2*A.*B)./(1+A+A.*A);
  f=2*atan(tanf2);
  rho=p./(1+cos(f));                % radial distance
  rhodot=n*p*sin(f);                 % radial velocity
```

```
 else
  f=-pi*ones(t);                        % degenerate case
  if inc ~= pi/2,error('degenerate parabola must have inc=90 deg'),end
  rho=(3*sqrt(gm/2)*abs(t-tau)).^(2/3);   % Battin pg 155 prob 4-3
  rhodot=sqrt(2*gm./rho).*sign(t-tau);    % approaching periapsis if t<tau
 end;             % if p>0
 theta=omega+f;                 % end parabolic case
end    % branch on z for type of orbit
% ------------------- map rho and rhodot into xyz --------------------
 com=cos(omega); som=sin(omega);
 cnode=cos(node); snode=sin(node);
 cthe=cos(theta); sthe=sin(theta);
 ci=cos(inc); si=sin(inc);
 lx=cnode*cthe-snode*sthe*ci;      %  direction cosines
 ly=snode*cthe+cnode*sthe*ci;
 lz=sthe*si;
 eer=[lx ly lz];
 r(:,1)=rho.*lx; r(:,2)=rho.*ly; r(:,3)=rho.*lz;          % position vector
 h=sqrt(gm*p);                  % angular momentum
 rfdot=h./rho;
 mx=-sthe*cnode-cthe*snode*ci;
 my=-sthe*snode+cthe*cnode*ci;
 mz= cthe*si;
 rdot(:,1)=rhodot.*lx+rfdot.*mx;                      % velocity vector
 rdot(:,2)=rhodot.*ly+rfdot.*my;
 rdot(:,3)=rhodot.*lz+rfdot.*mz;
 % end orb2x
```

```
function [r,v]=X2X(t,t1,r1,v1,mu,tol,ichk)
%[r,v]=X2X(t,t1,r1,v1,mu,tol,ichk)
%This function transforms the position and velocity from an initial state at t1 to an array
of times.
%Input: t - time - nx1 array of times
%       t1 - initial time of the position
%       r1  - the position vector at time t1 (3x1 array)
%       v1  - the velocity vector at time t1 (3x1 array)
%       mu  - GM - gravitational constant times the masses of the orbited body
%       tol - the tolerance of the final position and velocity vectors
%       ichk - 1 for input checking other for no checking
%Output: r  - the position vector in cartesian coordinates (3x1 array)
%        v  - the velocity vector in cartesian coordinates (3x1 array)

%make v and r verticle arrays
[n,m]=size(v1);
if m>n
   v1 = v1';
end
[n,m]=size(r1);
if m>n
   r1 = r1';
end

if ichk==1                  % do size and range testing
 if min(size(t))~=1, error('t must be an nx1 array'), end
 if max(size(t1))>1,error('t must be a scalar'),end                % inputs are scalars
 if min(size(v1))~=1, error('v must be a 3x1 array'), end
 if max(size(v1))~=3, error('v must be a 3x1 array'), end
 if max(size(v1)~=size(r1)), error('r must be a 3x1 array'), end
 if max(size(mu))>1,error('mu must be a scalar'),end
 if max(size(tol))>1,error('tol must be a scalar'),end
 if mu<=0, error('mu must be greater than 0'), end
 if tol<=0, error('tol must be greater than 0'), end
end      % if ichk==1

%find magnitude of v and r
mr1=norm(r1);
mv=norm(v1);
z=(2/mr1-mv^2/mu);  %equation 3-5
n=sqrt(z^3*mu);

h=cross(r1,v1);      %calculate angular momentum
e=cross(v1,h)/mu-r1/mr1; %equation 3-6
me=norm(e);          %magnitude of the e vector
```

```
Eo=acos((1-z*mr1)/me);   %equation 'a' on pg 3-9
Eo=sign(dot(r1,v1))*Eo;    %quadrant check for Eo
Mo=Eo-me*sin(Eo);         %equation 3-14
tau=t1-Mo/n;              %equation 3-14
M=n*(t-tau);

E=Kepler(M,me,tol,ichk);

dE=E-Eo;
sE=sin(dE);
cE=cos(dE);

F=1+(cE-1)/(z*mr1);
G=(t-t1)+(sE-dE)/n;

r=r1*F+v1*G;
mr=sqrt(sum(r.*r));
Fd = -n*sE./(z^2*mr*mr1);
Gd = 1+(cE-1)./(z*mr);
v=r1*Fd+v1*Gd;
r=r';
v=v';
```

```
function
[rtarg,rtarg_ul,rtarg_ur,rtarg_ll,rtarg_lr]=TARGcalc(PatCenPM,RscM00,VscM00,FOVti
me,Dim,FOVpos,Offset)
%[rtarg]=TARGcalc(PatCenPM,RSC,VSC,FOVtime,FOVpos)
%This function calculates the target vector corresponding to the FOV's.
%Input: PatCenPM - Pattern Center in True of Date Prime Meridian.
%       RscM00 - Spacecraft position vector in Mean of J2000 frame (3x1 array)
%       VscM00 - Spacecraft velocity vector in Mean of J2000 frame (3x1 array)
%       FOVtime - time of start of FOV scan (JD)
%                  Dim - dimensions of Pattern (2x1 array
%       FOVpos - position of FOV in pattern row, column ie (2,1)
%       Offset - Angle between adjacent FOV centers
%Output: rtarg - Target vector in True Prime Meridian of Date (x,y,z).
%Updated 31 Jan 01

Re=6378.14;  %radius of earth(km.)
f=1/298.25722;  %the flattening coefficient of the Earth
%Initialize arrays
RscPM=zeros(3,1);
VscPM=RscPM;
Xsc=RscPM;
Ysc=RscPM;

%Find rotation matrices
[TDTM00,sGHA,cGHA]=SETTRG(FOVtime);
%Rotate the spacecraft position and velocity from Mean of J2000 to Time of date
Equinox
RscTDT=TDTM00'*RscM00; %Matrix multiplication
VscTDT=TDTM00'*VscM00; %Matrix multiplication
%Rotate the spacecraft position and velocity from True of date Equinox to True of date
Prime meridian
RscPM(1)=RscTDT(1)*cGHA+RscTDT(2)*sGHA;
RscPM(2)=-RscTDT(1)*sGHA+RscTDT(2)*cGHA;
RscPM(3)=RscTDT(3);
VscPM(1)=VscTDT(1)*cGHA+VscTDT(2)*sGHA;
VscPM(2)=-VscTDT(1)*sGHA+VscTDT(2)*cGHA;
VscPM(3)=VscTDT(3);
%Calculate the line of sight vector to the pattern center
RlosCen=PatCenPM-RscPM;
RlosCen=RlosCen/norm(RlosCen);

%Calculate the rotation angles from the pattern center to the position of FOV
%Lateral angle from center of pattern (about pitch axis)
deltalat=Offset*(FOVpos(2)-(Dim(2)/2)-(1/2));
%Vertical angle from center of pattern (about roll axis)
deltavert=Offset*(-FOVpos(1)+(Dim(1)/2)+(1/2));
```

```
%Spacecraft frame y axis,Negative orbit normal, Vsc cross Rsc.
Ysc(1)=RscPM(3)*VscPM(2)-RscPM(2)*VscPM(3);
Ysc(2)=RscPM(1)*VscPM(3)-RscPM(3)*VscPM(1);
Ysc(3)=RscPM(2)*VscPM(1)-RscPM(1)*VscPM(2);
%Normalize vector
Ysc=Ysc/norm(Ysc);
%Spacecraft frame x axis, roughly positive velocity direction
Xsc(1)=RscPM(2)*Ysc(3)-RscPM(3)*Ysc(2);
Xsc(2)=RscPM(3)*Ysc(1)-RscPM(1)*Ysc(3);
Xsc(3)=RscPM(1)*Ysc(2)-RscPM(2)*Ysc(1);
%Normalize vector
Xsc=Xsc/norm(Xsc);

%First Rotation about pitch axis using deltalat and Ysc
%Rp=(RlosCen(1)*Ysc(1)+RlosCen(2)*Ysc(2)+RlosCen(3)*Ysc(3))*(1-
cos(deltalat))*Xsc+RlosCen*cos(deltalat)+sin(deltalat)*cross(Ysc,RlosCen);
Rp=(RlosCen(1)*Ysc(1)+RlosCen(2)*Ysc(2)+RlosCen(3)*Ysc(3))*(1-
cos(deltalat))*Ysc+RlosCen*cos(deltalat)+sin(deltalat)*cross(Ysc,RlosCen);

%Second Rotation about roll axis using deltavert and Xsc
%Giving the direction of the line of sight vector
Rlosd=(Rp(1)*Xsc(1)+Rp(2)*Xsc(2)+Rp(3)*Xsc(3))*(1-
cos(deltavert))*Xsc+Rp*cos(deltavert)+sin(deltavert)*cross(Xsc,Rp);

%Map the Line of sight vector to an elipsoid
%Method described in COMPATmemo2 by Ben George
Rsczf=RscPM(3)/(1-f);
Rloszf=Rlosd(3)/(1-f);
a=Rlosd(1)^2+Rlosd(2)^2+Rloszf^2;
b=2*(Rlosd(1)*RscPM(1)+Rlosd(2)*RscPM(2)+Rloszf*Rsczf);
c=RscPM(1)^2+RscPM(2)^2+Rsczf^2-Re^2;  %magnitude of s/c vector squared minus
the radius of the earth

K=(-b-sqrt(b^2-4*a*c))/(2*a);
%Calculate the target vector
rtarg(1)=RscPM(1)+K*Rlosd(1);
rtarg(2)=RscPM(2)+K*Rlosd(2);
rtarg(3)=RscPM(3)+K*Rlosd(3);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%This section is for calculating the locations of the FOV corners.
%Calculate the rotation angles from the pattern center to the position of FOV
```

```
%Lateral angle from center of pattern (about pitch axis)
del_lat_ul=(Offset)*(FOVpos(2)-Dim(2)/2-1);
%Vertical angle from center of pattern (about roll axis)
del_vert_ul=(Offset)*(-FOVpos(1)+Dim(1)/2+1);
%First Rotation about pitch axis using deltalat and Ysc
Rp_ul=(RlosCen(1)*Ysc(1)+RlosCen(2)*Ysc(2)+RlosCen(3)*Ysc(3))*(1-
cos(del_lat_ul))*Ysc+RlosCen*cos(del_lat_ul)+sin(del_lat_ul)*cross(Ysc,RlosCen);

%Second Rotation about roll axis using deltavert and Xsc
%Giving the direction of the line of sight vector
Rlosd_ul=(Rp_ul(1)*Xsc(1)+Rp_ul(2)*Xsc(2)+Rp_ul(3)*Xsc(3))*(1-
cos(del_vert_ul))*Xsc+Rp_ul*cos(del_vert_ul)+sin(del_vert_ul)*cross(Xsc,Rp_ul);

%Map the Line of sight vector to an elipsoid
%Method described in COMPATmemo2 by Ben George
Rsczf=RscPM(3)/(1-f);
Rloszf=Rlosd_ul(3)/(1-f);
a=Rlosd_ul(1)^2+Rlosd_ul(2)^2+Rloszf^2;
b=2*(Rlosd_ul(1)*RscPM(1)+Rlosd_ul(2)*RscPM(2)+Rloszf*Rsczf);
c=RscPM(1)^2+RscPM(2)^2+Rsczf^2-Re^2;  %magnitude of s/c vector squared minus
the radius of the earth

K=(-b-sqrt(b^2-4*a*c))/(2*a);
%Calculate the target vector
rtarg_ul(1)=RscPM(1)+K*Rlosd_ul(1);
rtarg_ul(2)=RscPM(2)+K*Rlosd_ul(2);
rtarg_ul(3)=RscPM(3)+K*Rlosd_ul(3);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Calculate the rotation angles from the pattern center to the position of FOV
%Lateral angle from center of pattern (about pitch axis)
del_lat_ur=(Offset)*(FOVpos(2)-Dim(2)/2);
%Vertical angle from center of pattern (about roll axis)
del_vert_ur=(Offset)*(-FOVpos(1)+Dim(1)/2+1);
%First Rotation about pitch axis using deltalat and Ysc
Rp_ur=(RlosCen(1)*Ysc(1)+RlosCen(2)*Ysc(2)+RlosCen(3)*Ysc(3))*(1-
cos(del_lat_ur))*Ysc+RlosCen*cos(del_lat_ur)+sin(del_lat_ur)*cross(Ysc,RlosCen);

%Second Rotation about roll axis using deltavert and Xsc
%Giving the direction of the line of sight vector
Rlosd_ur=(Rp_ur(1)*Xsc(1)+Rp_ur(2)*Xsc(2)+Rp_ur(3)*Xsc(3))*(1-
cos(del_vert_ur))*Xsc+Rp_ur*cos(del_vert_ur)+sin(del_vert_ur)*cross(Xsc,Rp_ur);

%Map the Line of sight vector to an elipsoid
%Method described in COMPATmemo2 by Ben George
```

```
Rsczf=RscPM(3)/(1-f);
Rloszf=Rlosd_ur(3)/(1-f);
a=Rlosd_ur(1)^2+Rlosd_ur(2)^2+Rloszf^2;
b=2*(Rlosd_ur(1)*RscPM(1)+Rlosd_ur(2)*RscPM(2)+Rloszf*Rsczf);
c=RscPM(1)^2+RscPM(2)^2+Rsczf^2-Re^2;  %magnitude of s/c vector squared minus
the radius of the earth

K=(-b-sqrt(b^2-4*a*c))/(2*a);
%Calculate the target vector
rtarg_ur(1)=RscPM(1)+K*Rlosd_ur(1);
rtarg_ur(2)=RscPM(2)+K*Rlosd_ur(2);
rtarg_ur(3)=RscPM(3)+K*Rlosd_ur(3);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Calculate the rotation angles from the pattern center to the position of FOV
%Lateral angle from center of pattern (about pitch axis)
del_lat_ll=(Offset)*(FOVpos(2)-Dim(2)/2-1);
%Vertical angle from center of pattern (about roll axis)
del_vert_ll=(Offset)*(-FOVpos(1)+Dim(1)/2);
%First Rotation about pitch axis using deltalat and Ysc
Rp_ll=(RlosCen(1)*Ysc(1)+RlosCen(2)*Ysc(2)+RlosCen(3)*Ysc(3))*(1-
cos(del_lat_ll))*Ysc+RlosCen*cos(del_lat_ll)+sin(del_lat_ll)*cross(Ysc,RlosCen);

%Second Rotation about roll axis using deltavert and Xsc
%Giving the direction of the line of sight vector
Rlosd_ll=(Rp_ll(1)*Xsc(1)+Rp_ll(2)*Xsc(2)+Rp_ll(3)*Xsc(3))*(1-
cos(del_vert_ll))*Xsc+Rp_ll*cos(del_vert_ll)+sin(del_vert_ll)*cross(Xsc,Rp_ll);

%Map the Line of sight vector to an elipsoid
%Method described in COMPATmemo2 by Ben George
Rsczf=RscPM(3)/(1-f);
Rloszf=Rlosd_ll(3)/(1-f);
a=Rlosd_ll(1)^2+Rlosd_ll(2)^2+Rloszf^2;
b=2*(Rlosd_ll(1)*RscPM(1)+Rlosd_ll(2)*RscPM(2)+Rloszf*Rsczf);
c=RscPM(1)^2+RscPM(2)^2+Rsczf^2-Re^2;  %magnitude of s/c vector squared minus
the radius of the earth

K=(-b-sqrt(b^2-4*a*c))/(2*a);
%Calculate the target vector
rtarg_ll(1)=RscPM(1)+K*Rlosd_ll(1);
rtarg_ll(2)=RscPM(2)+K*Rlosd_ll(2);
rtarg_ll(3)=RscPM(3)+K*Rlosd_ll(3);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%Calculate the rotation angles from the pattern center to the position of FOV
%Lateral angle from center of pattern (about pitch axis)
del_lat_lr=(Offset)*(FOVpos(2)-Dim(2)/2);
%Vertical angle from center of pattern (about roll axis)
del_vert_lr=(Offset)*(-FOVpos(1)+Dim(1)/2);
%First Rotation about pitch axis using deltalat and Ysc
Rp_lr=(RlosCen(1)*Ysc(1)+RlosCen(2)*Ysc(2)+RlosCen(3)*Ysc(3))*(1-
cos(del_lat_lr))*Ysc+RlosCen*cos(del_lat_lr)+sin(del_lat_lr)*cross(Ysc,RlosCen);

%Second Rotation about roll axis using deltavert and Xsc
%Giving the direction of the line of sight vector
Rlosd_lr=(Rp_lr(1)*Xsc(1)+Rp_lr(2)*Xsc(2)+Rp_lr(3)*Xsc(3))*(1-
cos(del_vert_lr))*Xsc+Rp_lr*cos(del_vert_lr)+sin(del_vert_lr)*cross(Xsc,Rp_lr);

%Map the Line of sight vector to an elipsoid
%Method described in COMPATmemo2 by Ben George
Rsczf=RscPM(3)/(1-f);
Rloszf=Rlosd_lr(3)/(1-f);
a=Rlosd_lr(1)^2+Rlosd_lr(2)^2+Rloszf^2;
b=2*(Rlosd_lr(1)*RscPM(1)+Rlosd_lr(2)*RscPM(2)+Rloszf*Rsczf);
c=RscPM(1)^2+RscPM(2)^2+Rsczf^2-Re^2;  %magnitude of s/c vector squared minus
the radius of the earth

K=(-b-sqrt(b^2-4*a*c))/(2*a);
%Calculate the target vector
rtarg_lr(1)=RscPM(1)+K*Rlosd_lr(1);
rtarg_lr(2)=RscPM(2)+K*Rlosd_lr(2);
rtarg_lr(3)=RscPM(3)+K*Rlosd_lr(3);
```

```
function [TDTM00,sGHA,cGHA,GHA] =SETTRG(JD)
% [TDTM00,sGHA,cGHA] =SETTRG(JD)
% This program calculates the rotation matrix TDTM00 from the True of Date to Mean
% of J2000 frame and the cosine and sine of the Greenwich Hour Angle for the given
% Julian date specified.
% Input          JD : Julian date
% Output TDTM00 : Rotation Matrix from true of date to mean of J2000
%        sGHA & cGHA: sine and cosine of Greenwich hour angle
%Referenced page numbers are from the pointing algorithm memo by Chauncey Uphoff
%Updated 31 Jan 01
%Updated 18 June 01, corrected "deps" equation to that of the memo.

%error checks
if max(size(JD))>1, error('Julian date is a scalar'), end
if JD<2451545, error('Julian date must be after J2000'), end
dr=pi/180; %define conversion from degrees to radians

%True of date to mean of date calculations to compensate for Nutation
%Calculations on pg.5
d=JD-2451545;
dpsi=(-0.0048*sin((125-0.05295*d)*dr)-0.0004*sin((200.9+1.97129*d)*dr))*dr;
deps=(0.0026*cos((125-0.05295*d)*dr)+0.0002*cos((200.9+1.97129*d)*dr))*dr;

%Calculations on pg.6
ce=cos(23.44*dr);
se=sin(23.44*dr);
NUTMAT=[1 dpsi*ce dpsi*se
  -dpsi*ce 1 deps
  -dpsi*se -deps 1];

%Mean of date to mean of J2000 calculations - pg.6
T=d/36525;

P(1,1)=1-(29724*T^2+13*T^3)*1e-8;
P(1,2)=(-2236172*T-667*T^2+222*T^3)*1e-8;
P(1,3)=(-971717*T+207*T^2+96*T^3)*1e-8;
P(2,2)=1-(25002*T^2+15*T^3)*1e-8;
P(2,3)=(-10865*T^2)*1e-8;
P(3,3)=1-(4721*T^2)*1e-8;
P(2,1)=-P(1,2);
P(3,1)=-P(1,3);
P(3,2)=P(2,3);

%True of date to mean of J2000 rotation matrix - pg.6
TDTM00=P'*NUTMAT;
```

```
%Rotation angle calculations from Prime meridian coordinates to True of Date
coordinates
% This method is the same as found on pg.9 except it uses a polynomial based on the
date
% after JD 2451545 instead of 2433282.5.  Calculations found on pg 52 of the
Explanatory
% supplement to the astronomical almanac.  Also the rotation rate of the earth, OMEGA,
is
% set at a constant value.
% calculate T
Frac=mod(d-.5,1);
T=(d-Frac)/36525;
% calculate GHA in degrees
GHA=100.4606184+36000.77005*T+(0.00038793)*T.^2-(2.6*10^-
7)*T.^3+360*1.002737909350795*Frac;
%Make GHA between 0 and 360;
GHA=mod(GHA,360);
GHA=GHA*dr; %Converts degrees to radians

%Sine and Cosine of GHA
sGHA=sin(GHA);
cGHA=cos(GHA);
```

```
function [f2,f3,g3] =SETSCPOS(Rsc,Vsc)
% [f2,f3,g3] =SETSCPOS(Rsc,Vsc)
% This program finds the f and g series coefficients for use in the 100Hz loop
% This is a change to the algorithm
% Input          Rsc: Position vector of S/C from the S/C bus -km (3x1 Array)
%       Vsc : Velocity vector of the S/C from the spacecraft bus -km/s (3x1 Array)
% Output f2 = t^2 coefficient in f series
%       f3 = t^3 coefficient in f series
%       g3 = t^3 coefficient in g series
%Updated 31 Jan 01

if max(size(Rsc))~=3|min(size(Rsc))~=1, error('Rsc must be a 3x1 array'), end
if max(size(Vsc))~=3|min(size(Vsc))~=1, error('Vsc must be a 3x1 array'), end

%Set constant values
GM=398600.44; %km^3/s^2

mr=sqrt(Rsc(1)^2+Rsc(2)^2+Rsc(3)^2); %Magnitude of the position
mrdot=sqrt(Vsc(1)^2+Vsc(2)^2+Vsc(3)^2); %Magnitude of the velocity

%Calculate the f and g series coefficients as on pg.8
f2=-GM/(2*mr^3);
f3=GM*(Rsc(1)*Vsc(1)+Rsc(2)*Vsc(2)+Rsc(3)*Vsc(3))/(2*mr^5);
g3=-GM/(6*mr^3);
```

```
function [RtargM00,RtargM00_ul,RtargM00_ur,RtargM00_ll,RtargM00_lr]
=TARGET(RtargPM,RtargPM_ul,RtargPM_ur,RtargPM_ll,RtargPM_lr,TFOV,Tact,TD
TM00,sGHA,cGHA)
% [RtargM00] =TARGET(RtargPM,TFOV,Tact,TDTM00,sGHA,cGHA)
% This program takes the position of the center of the FOV in True prime meridian of
date,
% Rotates the position to the True equinox of date frame and then rotates to J2000
% reference frame.  This function occurs within the 100Hz loop.
% Input          RtargPM: Center of FOV in Prime Meridian coordinates (3x1 array)
%       TFOV : Time at start of scan (s)
%       Tact : Actual time (s)
%       TDTM00 : True of date to J2000 rotation found in SETTRG
%       sGHA&cGHA : sine and cosine of Greenwich Hour Angle at start of FOV
% Output RtargM00 : Target Vector in Mean J2000 reference frame
%updated 14 Mar 01 - Corrected error in rotation from True Meridian to Tru Equinox.
%Updated 21 Feb 01 - eleminated creation of Delta cos matrix and calculated cosines and
sines directly
%Updated 31 Jan 01


if max(size(RtargPM))~=3|min(size(RtargPM))~=1, error('RtargPM should be a 3x1
array'), end
%This creates the table of sines and cosines described on pg 9 for up to a 10 second scan.
%This would normally be a hardwired table of numbers but is calculated manually here.
timestep=Tact-TFOV;
delta=timestep*3.64605792765e-5; %timestep multiplied by the rotation rate of the earth
cd=cos(delta);
sd=sin(delta);
ct=cGHA*cd-sGHA*sd;
st=sGHA*cd+cGHA*sd;
%Perform the rotation, RotMat, from True Prime meridian of date to True Equinox of
date
RtargTDT(1,:)=RtargPM(1)*ct-RtargPM(2)*st;
RtargTDT(2,:)=RtargPM(1)*st+RtargPM(2)*ct;
RtargTDT(3,:)=RtargPM(3);

%%Perform the rotation, RotMat, from True Prime meridian of date to True Equinox of
date
%RtargTDT(1,:)=RtargPM(1)*ct+RtargPM(2)*st;
%RtargTDT(2,:)=-RtargPM(1)*st+RtargPM(2)*ct;
%RtargTDT(3,:)=RtargPM(3);

%Rotate the target position from True Equinox of date to Mean of J2000
RtargM00=TDTM00*RtargTDT; %Matrix multiplication
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%For calculation of FOV corner points.
%Perform the rotation, RotMat, from True Prime meridian of date to True Equinox of
date
RtargTDT_ul(1,:)=RtargPM_ul(1)*ct-RtargPM_ul(2)*st;
RtargTDT_ul(2,:)=RtargPM_ul(1)*st+RtargPM_ul(2)*ct;
RtargTDT_ul(3,:)=RtargPM_ul(3);
%Rotate the target position from True Equinox of date to Mean of J2000
RtargM00_ul=TDTM00*RtargTDT_ul; %Matrix multiplication

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%For calculation of FOV corner points.
%Perform the rotation, RotMat, from True Prime meridian of date to True Equinox of
date
RtargTDT_ur(1,:)=RtargPM_ur(1)*ct-RtargPM_ur(2)*st;
RtargTDT_ur(2,:)=RtargPM_ur(1)*st+RtargPM_ur(2)*ct;
RtargTDT_ur(3,:)=RtargPM_ur(3);
%Rotate the target position from True Equinox of date to Mean of J2000
RtargM00_ur=TDTM00*RtargTDT_ur; %Matrix multiplication

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%For calculation of FOV corner points.
%Perform the rotation, RotMat, from True Prime meridian of date to True Equinox of
date
RtargTDT_ll(1,:)=RtargPM_ll(1)*ct-RtargPM_ll(2)*st;
RtargTDT_ll(2,:)=RtargPM_ll(1)*st+RtargPM_ll(2)*ct;
RtargTDT_ll(3,:)=RtargPM_ll(3);
%Rotate the target position from True Equinox of date to Mean of J2000
RtargM00_ll=TDTM00*RtargTDT_ll; %Matrix multiplication

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%For calculation of FOV corner points.
%Perform the rotation, RotMat, from True Prime meridian of date to True Equinox of
date
RtargTDT_lr(1,:)=RtargPM_lr(1)*ct-RtargPM_lr(2)*st;
RtargTDT_lr(2,:)=RtargPM_lr(1)*st+RtargPM_lr(2)*ct;
RtargTDT_lr(3,:)=RtargPM_lr(3);
%Rotate the target position from True Equinox of date to Mean of J2000
RtargM00_lr=TDTM00*RtargTDT_lr; %Matrix multiplication
```

```
function [RscM00] =SCPOS(Rsc,Vsc,f2,f3,g3,Teph,Tact)
% [RscM00] =SCPOS(R0,Rdot0,f2,f3,g3,Teph,Tact)
% This program extrapolates the position of the spacecraft in the 100Hz loop using
% f and g series with the coefficients already calculated
% Input          Rsc: Position vector of S/C at start of scan from the S/C bus - km (3x1
Array)
%       Vsc : Velocity vector of the S/C from the spacecraft bus -km/s (3x1 Array)
%       f2 : t^2 coefficient in f series
%       f3 : t^3 coefficient in f series
%       g3 : t^3 coefficient in g series
%       Teph : Time of ephemeris (sec)
%       Tact : Actual time after (sec)
% Output RscM00 : Position of the spacecraft in Mean J2000
%Updated 31 Jan 01

Time=Tact-Teph;
RscM00=zeros(3,1);

%Calculate the f and g series using precalculated coefficient values - pg.8
f=1+f2*Time^2+f3*Time^3;
g=Time+g3*Time^3;

%Calculate the extrapolated values of the spacecraft position in mean of J2000 frame
RscM00(1)=f*Rsc(1)+g*Vsc(1);
RscM00(2)=f*Rsc(2)+g*Vsc(2);
RscM00(3)=f*Rsc(3)+g*Vsc(3);
```

```
function [Rlosi,Azimuth,Elevation] = LOOK(RscM00,RtargM00,q2000,qst,qdyn)
% [Azimuth,Elevation] = LOOK(RscM00,RtargM00,q2000,qst,qdyn)
% Calculates the roll and pitch angles for the GIFTS 2 gimbal mirror given the spacecraft
and
% the target position in mean of J2000 frame, quaternion information from the star
tracker, and
% the static and dynamic orientation of the star tracker to the mirror
% quaternions are in form (vector, scalar)
% Input          RscM00 : position of the spacecraft in mean of J2000 frame (3x1 array)
%       RtargM00 : position of the target in mean of J2000 frame (3x1 array)
%       q2000 : quaternion from the star tracker (4x1 array)
%       qst : static quaternion relating star tracker and mirror
%       qdyn : dynamic quaternion relating star tracker and mirror
% Output Azimuth : Azimuth of the first gimbal (rad)
%       Elevation : elevation of the second gimbal (rad)
%Updated 21 Feb 01 - changed name of output to Azimuth and Elevation

%Updated 31 Jan 01

%Calculate line of site vector (look vector) in J2000
Rlos(1) = RtargM00(1)-RscM00(1);
Rlos(2) = RtargM00(2)-RscM00(2);
Rlos(3) = RtargM00(3)-RscM00(3);

%Multiply quaterions q2000, qst, and qdyn
qins=qmult(qmult(q2000,qst),qdyn);

%Invert qins
qinsinv(1)=-qins(1);
qinsinv(2)=-qins(2);
qinsinv(3)=-qins(3);
qinsinv(4)=qins(4);

%Turn Rlos into quaternion form
Rlos(4)=0;

%Rotate Rlos to instrument frame
Rlosi=qmult(qmult(qins,Rlos),qinsinv);

Rlosixz=sqrt(Rlosi(1)^2+Rlosi(3)^2);
mRlosi=sqrt(Rlosixz^2+Rlosi(2)^2);  %magnitude of Rlosi

%Find the Pitch and Roll angles
%Note: The way these angles are defined may change depending on how the mirror is set
% in the instrument reference frame but the process of finding the angles using arccosines
% will probably be similar.  Expect this part of the code to change
```

```
Azimuth = acos(Rlosi(3)/Rlosixz);
Elevation = acos(Rlosixz*sign(Rlosi(3))/mRlosi);

%Place pitch and roll between -pi and pi
if Rlosi(1)<0
   Azimuth=-Azimuth;
elseif Rlosi(2)>0
   Elevation=-Elevation;
end
```

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 2003	Contractor Report

4. TITLE AND SUBTITLE
Preliminary Design and Analysis of the GIFTS Instrument Pointing System

5. FUNDING NUMBERS
NCC1-01017
755-03-00-03

6. AUTHOR(S)
Paul P. Zomkowski

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
The George Washington University
Joint Institute for Advancement of Flight Sciences
NASA Langley Research Center
Hampton, VA 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23681-2199

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA/CR-2003-211937

11. SUPPLEMENTARY NOTES
The information presented in this report was submitted to the School of Engineering and Applied Science of The George Washington University in partial fulfillment of the requirements for the Degree of Master Science, August 2002. Langley Technical Monitor: Lucas G. Horta

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified-Unlimited
Subject Category 39 Distribution: Nonstandard
Availability: NASA CASI (301) 621-0390

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
The Geosynchronous Imaging Fourier Transform Spectrometer (GIFTS) Instrument is the next generation spectrometer for remote sensing weather satellites. The GIFTS instrument will be used to perform scans of the Earth's atmosphere by assembling a series of field-of views (FOV) into a larger pattern. Realization of this process is achieved by step scanning the instrument FOV in a contiguous fashion across any desired portion of the visible Earth. The main objective is to validate the pointing algorithm in the presence of spacecraft disturbances and determine acceptable disturbance limits from expected noise sources. Proof of concept validation of the pointing system algorithm is carried out with a full system simulation developed using Matlab/Simulink. Models for the following components function within the full system simulation: inertial reference unit (IRU), attitude control system (ACS), reaction wheels, star tracker, and mirror controller. This comprehensive simulation will also aid in obtaining a thorough understanding of spacecraft disturbances and other sources of pointing system errors. Parameter sensitivity studies and disturbance analysis will be used to obtain "limits of operability" for the GIFTS instrument. The culmination of this simulation development and analysis will be used to validate the specified performance requirements outlined for this instrument.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Spacecraft simulation; GIFTS; Pointing control; Jitter; Multi-body dynamics	133
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL