

**JET PROPULSION LABORATORY**

**INTERNAL DOCUMENT**

JPL D-24809

June 30, 2004

**SPITZER INSTRUMENT POINTING FRAME (IPF)**

**KALMAN FILTER ALGORITHM**

David S. Bayard and Bryan H. Kang

Autonomy and Control Section (345)  
Jet Propulsion Laboratory  
California Institute of Technology  
4800 Oak Grove Drive  
Pasadena, CA 91109

# SPITZER INSTRUMENT POINTING FRAME KALMAN FILTER ALGORITHM

David S. Bayard and Bryan H. Kang

Autonomy and Control Section (345)  
Jet Propulsion Laboratory  
California Institute of Technology  
4800 Oak Grove Drive  
Pasadena, CA 91109

## ABSTRACT

This paper discusses the Spitzer Instrument Pointing Frame (IPF) Kalman Filter algorithm. The IPF Kalman filter is a high-order square-root iterated linearized Kalman filter, which is parametrized for calibrating the Spitzer Space Telescope focal plane and aligning the science instrument arrays with respect to the telescope boresight. The most stringent calibration requirement specifies knowledge of certain instrument pointing frames to an accuracy of 0.1 arcseconds, per-axis, 1-sigma relative to the Telescope Pointing Frame. In order to achieve this level of accuracy, the filter carries 37 states to estimate desired parameters while also correcting for expected systematic errors due to: (1) optical distortions, (2) scanning mirror scale-factor and misalignment, (3) frame alignment variations due to thermomechanical distortion, and (4) gyro bias and bias-drift in all axes. The resulting estimated pointing frames and calibration parameters are essential for supporting on-board precision pointing capability, in addition to end-to-end “pixels on the sky” ground pointing reconstruction efforts.

During Spitzer’s 3 month In-Orbit Checkout (IOC) period, the space telescope perform a series of repeated calibration maneuvers for each science array, taken on a time interval spanning less than 10 hours, giving rise to a calibration data set for that array. After the calibration data set is available for a given science array, the IPF filter processes the collected attitude history data and instrument centroid data, and produces an estimate of the instrument frame along with estimates of other alignment and calibration parameters. For each array, the calibration will be performed twice. Specifically, a “coarse” calibration will be performed before the optics have fully cooled, and a “fine” calibration will be performed after the telescope is fully operational. The basic philosophy is to combine a high-order Kalman filter with carefully designed on-orbit experiment designs to achieve the overall desired calibration accuracy.

In order to meet requirements, the IPF filter has several novel and important features

including,

1. a gyro pre-processor which allows gyro sensitivities to be pre-computed and stored beforehand. This completely eliminates the need for repeated and time-consuming gyro sensitivity propagation during each iteration of the filter cycle;
2. a parameter “masking” capability which allows one to define the state using any arbitrary subset of parameters. This provides a flexible parametrization which can be used to match different levels of model fidelity to a wide variety of science array types;
3. a formulation based on a square-root iterated linearized Kalman filter for high accuracy and good numerical conditioning;
4. the flexibility to sequentially update prior estimates based on multiple data sets, where certain subsets of parameters are expected to change and not to change from one data set to another;
5. a sandwich-based experiment design concept which starts and ends each calibration maneuver on the same reference sensor. This provides observability of all desired parameters, and allows the same calibration filter to be used for a multitude of different array types (cameras, spectroscopy slits, scanning instruments);
6. the ability to integrate both visible and infra-red sources in the same calibration data set;
7. the ability to process “partial” centroids which only contain information along one axis of the array. This occurs, for example, when calibrating the entrance aperture of a spectroscopy slit by first scanning a source across the narrow slit width, and then along its length at a later time;
8. operation in one of several possible “lite” modes to allow a trade-off between accuracy and robustness. For example, a completely gyroless lite mode can be invoked if there is only minimal or incomplete data.

The Kalman filter performance and all operational modes were benchmarked using simulated data produced by the IPF Filter Unit Test Environment (FLUTE). FLUTE is a unit test environment specifically designed for simulating focal plane survey maneuvers and includes all representative systematic pointing errors, and optical distortions. Based on extensive FLUTE-based testing, the filter was found to meet all operational and performance requirements.

This document includes: (i) The IPF filter functional and performance requirements definitions, (ii) The filter model parameter derivations (i.e. state variables), (iii) The filter formulation, (iv) An overview of the algorithm implementation and (v) The filter verification process.

# Contents

<b>1</b>	<b>INTRODUCTION</b>	<b>10</b>
1.1	Overview of the Spitzer Mission . . . . .	10
1.2	Spitzer Focal Plane Survey and IPF Filter Operations . . . . .	11
1.3	Organization of the Document . . . . .	12
<b>2</b>	<b>IPF REQUIREMENTS</b>	<b>13</b>
2.1	IPF Kalman Filter Design Requirements . . . . .	13
2.2	Experiment Design - Sandwich Maneuvers . . . . .	13
2.3	Performance Requirement and Error Budget . . . . .	15
2.4	Input-Output Data Interface Requirements . . . . .	17
<b>3</b>	<b>PRELIMINARY DEFINITIONS</b>	<b>19</b>
3.1	Overview of Pointing-Relevant Frames . . . . .	19
3.2	Body Frame Definition . . . . .	21
3.3	Telescope Point Frame (TPF) Definition . . . . .	22
3.4	Instrument Pointing Frame (IPF) Definition . . . . .	23
3.5	Spitzer Focal Plane Survey Frame Table Definition . . . . .	25
3.6	Scan-Offset $IPF_{\Gamma}$ Frame Definition . . . . .	30
3.7	Brown Angle Definition . . . . .	30
3.8	Standard Coordinates . . . . .	31
3.9	Oriented Angular Pixel (OAP) Coordinates . . . . .	32
3.10	Mapping OAP to Standard Coordinates . . . . .	33
3.11	Calibration Equation . . . . .	34
3.12	Pixels to Sky Reconstruction . . . . .	36
3.13	Time Axis Definition . . . . .	37
<b>4</b>	<b>IPF FILTER PARAMETERS</b>	<b>38</b>
4.1	Overview . . . . .	38
4.2	Optical Distortion Parameters . . . . .	42
4.3	Scan Mirror Rotation Parameters . . . . .	42
4.4	Telescope Pointing Frame Parameters . . . . .	43

4.5	Thermomechanical Drift Parameters . . . . .	43
4.6	Attitude and Gyro Parameters . . . . .	44
<b>5</b>	<b>FORMULATIONS</b>	<b>45</b>
5.1	IPF Kalman Filter Formulation . . . . .	45
5.1.1	Time Update . . . . .	45
5.1.2	Measurement Update . . . . .	46
5.2	Sensitivity Equations for $p_1$ Parameters . . . . .	50
5.3	Sensitivity Equations for $p_2$ and $\psi$ Parameters . . . . .	51
5.3.1	$\frac{\partial z}{\partial s}$ Derivation . . . . .	52
5.3.2	$\frac{\partial s}{\partial \eta}$ Derivation . . . . .	52
5.3.3	$\frac{\partial \eta}{\partial \lambda}$ Derivation . . . . .	53
5.3.4	$\frac{\partial \lambda}{\partial (\delta p_2)}$ Derivation . . . . .	53
5.4	IPF Gyro Pre-Processor . . . . .	58
5.5	Calculation of $G^\circ$ . . . . .	58
5.6	Calculation of $H_g$ . . . . .	58
<b>6</b>	<b>ALGORITHMS</b>	<b>60</b>
6.1	IPF Filter Algorithm Overview . . . . .	60
6.2	Input Database and User Interface . . . . .	61
6.2.1	User Configuration Parameters . . . . .	61
6.2.2	Merging and Editing Centroid Files . . . . .	63
6.2.3	Gyro Preprocessing . . . . .	64
6.3	Scalable Kalman Filter . . . . .	68
6.4	Least Squares Data Analysis . . . . .	71
6.5	Output Preparation . . . . .	74
6.5.1	Initial Attitude Uncertainty Estimation . . . . .	75
6.6	Inferred Frames . . . . .	76
6.6.1	Inferred Frame Definition . . . . .	77
6.6.2	Explicit Expression for $\tilde{T}$ . . . . .	77
6.6.3	Inferred Frame Estimate $\hat{\tilde{T}}$ . . . . .	80
6.6.4	Inferred Frame Covariance . . . . .	82

6.6.5	Jacobian Expression . . . . .	84
6.7	Special Cases: Partial Centroid Measurements . . . . .	87
6.7.1	Overview of Slit Mode . . . . .	87
6.7.2	Sensitivities for Slit Mode . . . . .	88
6.8	Special Cases: IPF Multi-Run Tool . . . . .	90
<b>7</b>	<b>IPF SOFTWARE VERIFICATION</b>	<b>93</b>
7.1	Software Verification Procedure . . . . .	93
7.2	Description of Test Cases . . . . .	93
<b>8</b>	<b>EXAMPLE FILTER RUN: MIPS</b>	<b>95</b>
8.1	Summary of Results . . . . .	95
8.2	Output Figures . . . . .	97
8.3	IPF OUTPUT DATA (IF MINI FILE) . . . . .	110
8.4	IPF EXECUTION LOG . . . . .	116
<b>9</b>	<b>CONCLUSIONS</b>	<b>119</b>
<b>A</b>	<b>APPENDIX: ACRONYMS</b>	<b>120</b>
<b>B</b>	<b>APPENDIX: IPF Filter Interfaces</b>	<b>121</b>
B.1	Velocity Correction of Star Data . . . . .	121
B.2	Centroid Measurements to Oriented Angular Pixel Coordinates . . . . .	121
B.3	Mirror Encoder Data to Mirror Angle Conversion . . . . .	122
B.4	Inferred Frames from Desired Offsets . . . . .	123
<b>C</b>	<b>APPENDIX: Useful Definitions</b>	<b>124</b>
C.1	Definition: Cross-Product Matrix . . . . .	124
C.2	Definition: Matrix Square-Root . . . . .	124
<b>D</b>	<b>APPENDIX: Useful Lemmas</b>	<b>125</b>
D.1	Kalman Filter Update - Array Square Root Method . . . . .	125
D.2	Gyro Perturbation Lemma . . . . .	127
D.3	Measurement Equation Lemma . . . . .	129

D.4 Attitude Linearization Lemma . . . . .	130
D.5 Angle-Axis Perturbation Lemma . . . . .	131
D.6 Unit Vector Perturbation Lemma . . . . .	136
D.7 Cross Product Operations Lemma: Matrix to Vector I . . . . .	137
D.8 Cross Product Operations Lemma: Matrix to Vector II . . . . .	138
D.9 Cross Product Operations Lemma: Matrix to Outer Product . . . . .	139
D.10 Cross - Outer Product Lemma I . . . . .	139
D.11 Cross - Outer Product Lemma II . . . . .	140
D.12 Cross - Outer Product Lemma III . . . . .	141
D.13 Cross - Outer Product Lemma IIIa . . . . .	141
D.14 Push-Through Lemma . . . . .	142
D.15 Directional Cosine Matrix Exponential Equivalence Lemma . . . . .	142
D.16 Directional Cosine Matrix Integral Lemma . . . . .	144
D.17 Sum Factorization Lemma . . . . .	145
D.18 QR Calculation of $C$ Lemma . . . . .	146
D.19 Differentiation through Inversion Lemma . . . . .	148
D.20 Norm of Mirror-Axis Perturbation Lemma . . . . .	149
D.21 Optical Distortion Vectorization Lemma . . . . .	150
D.22 Matrix Vectorization Lemma . . . . .	152
D.23 Vector-Kronecker Equivalence Lemma . . . . .	153
D.24 Kronecker Product Lemma . . . . .	154

# List of Figures

2.1	Spitzer Sandwich Maneuver for IPF Calibration . . . . .	14
2.2	Spitzer IPF Estimation Error Budget for Fine Focal Plane Surveys . . . . .	17
2.3	Input/Output Description of IPF Filter . . . . .	18
3.1	Spitzer Frames and Transformations . . . . .	19
3.2	Definition of TPF frame in terms of $\beta$ angles, shown projected onto sky, and looking from inside the celestial sphere. . . . .	22
3.3	Spitzer Instrument Prime Frames Definition . . . . .	24
3.4	End-to-End Pointing Transformations . . . . .	35
3.5	Spitzer IPF Time Axis Definition . . . . .	37
5.1	Square-Root, Iterated and Linearized Kalman Filtering Process . . . . .	45
6.1	IPF Functional Block Diagram . . . . .	62
6.2	Merging and Editing Centroid Files . . . . .	65
6.3	Gyro Preprocessor functional block diagram . . . . .	67
6.4	Overview of SRILKF Architecture . . . . .	69
6.5	Overview of SRILKF Architecture (BOX A) . . . . .	70
6.6	Least Squares Filter and Analysis Diagram . . . . .	73
6.7	IPF Output Data Processing And Plotting Routine . . . . .	74
6.8	Telescope Pointing Frame (TPF), Instrument Pointing Frame (IPF), Inferred Frame $\widehat{IPF}$ , and the mappings that relate them . . . . .	76
6.9	Survey Maneuver Through Slit Instrument - Partial Centroid . . . . .	87
7.1	IPF Filter and Test Environment . . . . .	94
8.1	A-priori and a-posteriori IPF frames (ZOOMED) . . . . .	96
8.2	TPF coords of measurements and a-priori predicts . . . . .	97
8.3	Oriented Pixel Coords of measurements and a-priori predicts . . . . .	97
8.4	A-priori prediction error . . . . .	98
8.5	Oriented Pixel Coords of measurements and a-priori predicts (PCRS only) . . . . .	98
8.6	IPF execution convergence, chart 1 . . . . .	99
8.7	IPF execution convergence, chart 2 . . . . .	99
8.8	Parameter uncertainty convergence . . . . .	100
8.9	IPF parameter symbol table . . . . .	100

8.10	KF parameter error sigma plots . . . . .	101
8.11	LS parameter error sigma plot . . . . .	101
8.12	KF and LS parameter error sigma plot . . . . .	102
8.13	Oriented Pixel Coords of meas. and a-posteriori predicts . . . . .	103
8.14	Oriented Pixel Coords of meas. and a-posteriori predicts (attitude corrected) . . . . .	103
8.15	Array plot with (solid) and w/o (dashed) optical distortion corrections . . . . .	104
8.16	KF innovations with (o) and w/o (+) attitude corrections . . . . .	104
8.17	W-axis KF innovations and 1-sigma bound . . . . .	105
8.18	V-axis KF innovations and 1-sigma bound . . . . .	105
8.19	Estimated attitude corrections (Body frame) . . . . .	106
8.20	Estimated attitude error sigma plot (Body frame) . . . . .	106
8.21	Thermo-mechanical boresight drift (equiv. angle in (W,V) coords) . . . . .	107
8.22	Thermo-mechanical boresight drift (equiv. angle in Body frame) . . . . .	107
8.23	Gyro drift bias contribution (equiv. rate in (W,V) coords) . . . . .	108
8.24	Gyro drift bias contribution (equiv. angle in (W,V) coords) . . . . .	108
8.25	Gyro drift bias contribution (equiv. angle in Body frame) . . . . .	109

# 1 INTRODUCTION

## 1.1 Overview of the Spitzer Mission

The Spitzer Space Telescope will provide high-resolution views of the universe in the infrared spectrum, and represents the fourth and final element in NASA's Great Observatory program. The new space telescope is presently scheduled for launch on a Delta II in the August 2003 time frame. Spitzer has an 85 cm telescope aperture and uses a combination of passive cooling and liquid helium to keep its infrared instruments at temperatures of 1.4 degrees Kelvin. It will be launched into an Earth-trailing heliocentric orbit, slowly moving away from Earth with a small drift rate of about 0.1 AU per year [14]. Working in the infrared, Spitzer complements the range of science observations and wavelengths covered by the other three previous NASA Great Observatories (Chandra for X-ray, Hubble for visual, Compton GRO for gamma-rays).

Spitzer is designed to carry three science payload instruments: the Infrared Array Camera (IRAC), the Infrared Spectrograph (IRS) and the Multi-band Imaging Photometer for SIRTf (MIPS). **IRAC** is a general-purpose camera operating at near and mid-infrared wavelengths, which is designed to produce images 5.12 arc-min square in angular size. The images are taken simultaneously at 3.6, 4.5, 5.8 and 8.0 microns using four detector arrays of 256x256 pixels each. **IRS** provides high and low-resolution spectra of astrophysical objects over wavelengths of 5.3 to 40 microns. Separate modules of 128x128 pixels are available to be used for low-resolution spectroscopy in the 5.3-14 micron wavelength (Short-Low), high-resolution spectroscopy at 10-19.5 microns (Short-Hi), low-resolution at 14-40 microns (Long-Low), and high-resolution at 19-37 microns (Long-Hi). The IRS instrument includes peak-up arrays to provide real-time centroids of targeted IR objects, to facilitate their accurate transfer to the spectroscopy slits [1]. **MIPS** provides long wavelength imaging and large area mapping over wavelengths of 20 to 200 microns, and has certain limited spectroscopy capability. The MIPS payload has three detector arrays consisting of a 128x128 pixel array for images at 24 microns, a 32x32 array for images at 70 microns and spectra from 50-100 microns, and a 2x20 array for images at 160 microns. A main distinguishing element of MIPS instrument is its scanning mirror, which moves along a single axis and is coordinated with spacecraft motions to facilitate efficient science observations.

The most stringent calibration requirements are imposed by the IRS Short-Hi spectroscopy slit which is only 4 arcseconds wide. The need to target a celestial source to the center of the slit for high-resolution science places tight requirements on both spacecraft attitude control, knowledge and calibration accuracy.

During the In-Orbit Checkout (IOC) period (from launch to the first 3-4 months into the Spitzer operations), a portion of spacecraft operational time will be dedicated to learning the instrument frame alignments with respect to the Telescope Pointing Frame (TPF). There are up to 128 frames to be calibrated (including those that are inferred or derived), and

the developed Instrument Pointing Frame (IPF) Kalman filter will support these activities by estimating the Prime frames and calculating Inferred frames. The IPF operation will be interactively coordinated with the Spitzer spacecraft planned maneuvers and data links as a combined Focal Plane Survey (FPS) effort.

## 1.2 Spitzer Focal Plane Survey and IPF Filter Operations

There are 17 detectors of the array or slit type (cf., Figure 3.3) located in the circular focal plane associated with the IRAC, IRS and MIPS payload instruments. In order to enable efficient attitude commanding, 128 key frames are defined in the focal plane and stored as quaternions in an on-board frame table. An example frame table is shown in Table 3.6. The on-board frame table indexes these quaternions with numbers from 1 to 128, and lies at the heart of all pointing commands given to the spacecraft. For example, a typical command might be to point frame number 95 to a specified RA and DEC location on the sky. From this information and a sun-avoidance constraint, the attitude is determined uniquely.

Of the 128 frames, about 30 frames are estimated directly and are denoted as “Prime”. Certain other frames are defined in terms of their pixel offset relative to a nearby Prime frame. These frames are denoted as “Inferred” because their alignment is estimated by the IPF filter based on their proximity to the associated Prime frame.

In order to estimate the desired IPFs, the spacecraft must go through series of maneuvers to place a given star at several locations on the instrument array to be calibrated, and also on two reference PCRS detectors. The measurements collected by the spacecraft through these maneuvers must be informative enough to extract the desired parameters. Therefore, prior to any IPF filter execution, the Spitzer mission operations must design a campaign of maneuvers that meets the estimation performance and observability requirements. The type and number of maneuvers are carefully selected such that the desired parameters associated with each frame can be identified and estimated. A typical maneuver consists of placing a source on the first PCRS, moving it to the second PCRS, moving it to several locations on the science array, and then moving it back to the first PCRS. After a completion of a calibration data set, the Spitzer spacecraft downlinks measurements to a designated ground station where each of the IPF input file providers decodes the telemetry and creates necessary IPF input files.

When the IPF input files arrive to the IPF team, the team runs the IPF filter and obtains estimates of the desired Instrument Pointing Frames (IPFs). In addition, the IPF filter also provides corrections to systematic errors including: (1) optical distortion, (2) scan mirror misalignments (i.e., for the MIPS instrument), (3) frame alignment variations due to thermomechanical distortion, and (4) gyro bias and bias-drift. The IPF filter outputs are then uploaded to the Spitzer spacecraft, as they become available, to update the on-board frame table. Also the IPF filter outputs are used by the science and engineering teams for planning and analysis efforts, and for ground-based “pixel on the sky” pointing reconstruction.

Users of the IPF filter should have a working understanding of modern estimation theory. This document lays out the filter derivations, defines necessary terms and variables, and walks through the algorithm in great detail. In combination with the document “*Spitzer Instrument Pointing Frame (IPF) Filter Software Description Document and User’s Guide*” [13], these documents will provide sufficient background to understand the filter and provide guidelines for proper filter operation.

### 1.3 Organization of the Document

This document describes the development of the IPF filter algorithm and is organized as follows. Section 2 summarizes the IPF design requirements. This section includes a discussion of the maneuver design, defines the estimation error budgets, and outlines the filter input-output (data file) requirements. All necessary coordinate frame definitions are given in Section 3. In this section, the time-line for the filtering process is defined along with the frames needed to derive the filter equations (such as TPF, IPF etc.). Section 4 describes the filter state parametrization. This section introduces all models used to derive the filter state variables. The models include various systematic errors such as optical distortion, mirror rotational misalignment, thermomechanical alignment variations, and gyro drift bias. The filter state vector is defined. Section 5 derives the IPF Kalman filter. This section includes the summary of the filter equations, the filter structure and associated sensitivity functions. Special features of the filter such as the derivations for the gyro pre-processor are documented in this section. Section 6 documents the actual implementation of the filter algorithm. The input/output file interface and the MATLAB User Interface (MUI) are discussed. This section explains how the gyro preprocessing and scalable Kalman Filter architecture is implemented. This section also discusses how the IPF filter generates output data and performs error analysis. After filter development was completed, the filter algorithm was tested and verified using FLUTE generated data. Section 7 describes this software verification process. This section briefly summarizes the test cases and the results. Appendix A lists the acronyms used in this document. Appendix B includes the filter interfaces with the input files. This information can be used to relate the mathematical variables with the input file database. While deriving the filter equations, numerous mathematical proofs and ancillary results were necessary. These results are summarized in Appendix C for completeness.

## 2 IPF REQUIREMENTS

### 2.1 IPF Kalman Filter Design Requirements

An overall high-level design requirement is for the IPF filter to automate the focal plane calibration process and support time-constrained IOC operations. More specific design requirements are itemized in Table 2.1. A MATLAB implementation was selected to keep the development cycle time short, and to leverage experiences drawn from the Shuttle Radar Topography Mission (SRTM) where a similar MATLAB-based pipeline was successfully used to process large attitude and calibration data sets [25].

Given the input files, the IPF filter is required to merge the CA and CB files, and permit the removal of bad data by editing. The filter should operate using only a few simple commands and minimal required user interactions. The filter architecture should be flexible enough to support calibration of the entire suite of Spitzer instruments and array types by manipulating only a few software configuration flags and mask vectors. The code should handle partial centroids associated with spectroscopy slit type measurements. The same core filter routines should be callable from the multi-run merge tool which enables estimation of frames using data sets spanning several days. The filter software should self-generate the output files and self-archive all filter execution paths and configurations for each run. The output should include an independent least squares analysis, show plots of the results, and summarize the calculations of Prime and Inferred frames. In addition, the filter should be computationally efficient so that an IPF filter run (input data analysis, filter execution, filter output data analysis and the frame table entry update) can be completed within the IOC time-line allocation of 4 hours. This 4 hour allocation is measured from the time a complete data set is received, to the time the IPF filter outputs are logged into the mission archive.

All of the above design requirements and goals were met by the developed IPF architecture. The use of configuration flags and masking vectors solved the automation and archiving issues, and the gyro-pre-processor was developed to minimize computer memory and CPU intensive calculations.

### 2.2 Experiment Design - Sandwich Maneuvers

For each instrument array to be calibrated, the in-flight experiment design consists of commanding the Spitzer spacecraft through a series of “sandwich” type maneuvers as shown in Figure 2.1. Each sandwich maneuver (i.e., maneuver) consists of the following sequence of steps.

1. Locate a target star on the first PCRS detector, PCRS 1, and take a centroid measurement

2. Move target star to PCRS 2, and take centroid measurement
3. Move target star to several positions on the desired science instrument array, and take a centroid measurement at each location (*for example: a 5 of diamonds pattern, or a 3x3 grid*)
4. Return to PCRS 1 detector, and take centroid measurement

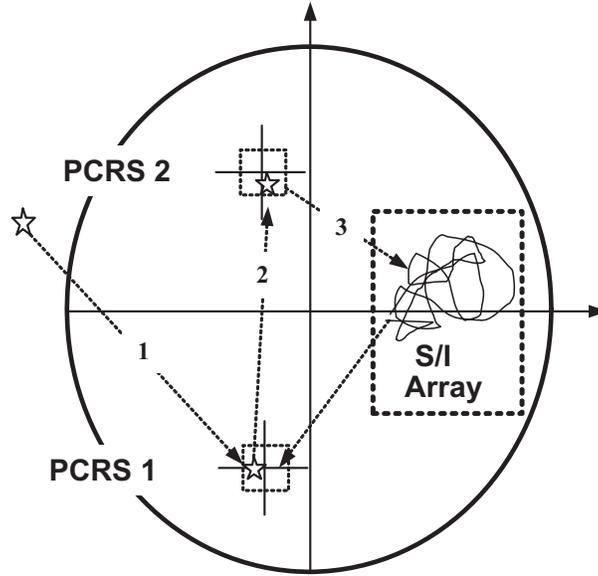


Figure 2.1: Spitzer Sandwich Maneuver for IPF Calibration

The Telescope Pointing Frame (TPF) is defined in terms of the locations of the two PCRS boresight unit vectors (cf., Section 3.3). These boresight unit vectors are defined by the null points at the centers of the two PCRS sensors, and are important because they can be consistently and repeatably established by empirical measurements taken in-flight. By transitioning between PCRS1 and PCRS2, and the science array, the sandwich maneuver becomes informative about the location of the IPF with respect to the TPF, and the TPF with respect to the body frame (defined by the star tracker). Also, by beginning and ending on the same PCRS, the sandwich maneuver is informative about accumulated attitude error due to gyro drift, which can be calibrated out accordingly.

For spectroscopy instruments it is desired to calibrate fiducial points in the entrance aperture rather than in pixel coordinates. In this case the centroid measurements used for calibration must be replaced by a suitable pseudo-measurement. This is done by scanning a source across the entrance aperture, and reporting a centroid at the slit center at the time of peak intensity, and reporting centroids at the slit edges at times when the intensity has fallen to an agreed upon level relative to the peak. The resulting time-tagged list of aperture-relative locations is reported to the IPF filter as if it were a list of time-tagged centroids obtained from an array of real pixels.

For longer wavelength arrays such as the MIPS 70  $\mu\text{m}$ , 160  $\mu\text{m}$ , and SED, the sandwich maneuvers can be designed so that the source used for PCRS centroiding is different from that used for science centroiding. This may be needed due to the general difficulty in finding a single target source which centroids consistently at both visible and long-infrared wavelengths.

Most generally, the science portion of the sandwich maneuver can be designed to include simultaneous centroids taken from a group of stars (e.g., an astrometric cluster) which are offset and/or dithered in any desired pattern to generate a large number of calibration centroids. This approach is useful for calibrating the plate scales and optical distortions, particularly when the centroids are large in number and cover the entire array.

Sandwich maneuvers are typically repeated several times to allow statistical averaging of results, giving a corresponding reduction of errors. The performance and convergence of the IPF filter depends on both the sandwich maneuver design and the total number of maneuvers. The collection of all sandwich maneuvers associated with a single science array is called a “calibration survey”. The IPF filter is run separately on each calibration survey data set.

### 2.3 Performance Requirement and Error Budget

The absolute end-to-end pointing requirement for Spitzer is 5 arcseconds [9] (1-sigma, radial), and 1.4 arcseconds absolute for pointing reconstruction. The most stringent requirement is for relative pointing, where attitude offsets accurate to 0.4 arcseconds are required to place a target star or Infrared source on an IRS slit to within 0.4 arcseconds of its center (.28 arcseconds in the dispersion direction). This in turn requires knowledge of the slit location to better than 0.14 arcseconds (1-sigma, radial). Derived calibration requirements for knowledge of the various instrument frames are tabulated in Table 2.2.

The on-orbit calibration experiment must be designed (centroid location sequence, number of maneuvers, etc.) to meet calibration error budget requirements. There are two sets of error budgets corresponding to Coarse and Fine surveys. Coarse surveys occur earlier in the IOC period, where requirements are more relaxed and the telescope is still cooling. Fine surveys occur later in the IOC period after the telescope has cooled sufficiently and the optics and science arrays become fully operational. A representative error budget for the Fine survey is shown in Figure 2.2. The goal of the IPF filter development is to achieve the required calibration accuracies using the defined experiment designs.

No.	Design Features and Requirements
1	Automated input file extraction capability (Specify names in RN file)
2	Automated data merging capability for CA and CB files
3	Data editing capability for centroid and attitude data files
4	MATLAB based software with minimal required user interactions
5	Single code to operate in multiple modes (different instruments, multi-MIPS)
6	Simple mode configuration via configuration flags and masking vectors
7	Allow specialization to SLIT type of measurements
8	Estimate Inferred frames
9	Comply with performance requirements (see Table 2.2)
10	Computationally efficient code (handle 10 hour data within 1 hour run)
11	Automated output file generation
12	Automated error analysis( based on least squares) and output plotting
13	Automated archiving of run configurations and estimation results

Table 2.1: IPF Filter Design Requirements

Array/Slit	Desired IPF Reconstruction Accuracy [arc-second]	Required TPF to IPF Alignment [arc-second]	
		Coarse	Fine
IRS PeakUp Red	1.346	1	0.25
IRS PeakUp Blue	1.346	1	0.25
IRS PeakUp Red Sweetspot	1.346	1	0.14
IRS PeakUp Blue Sweetspot	1.346	1	0.14
IRS Short-Lo	1.317	1	0.14
IRS Long-Lo	1.338	1	0.28
IRS Short-Hi	1.315	1	0.14
IRS Long-Hi	1.315	1	0.28
IRAC 3.6 $\mu\text{m}$	1.346	1	0.14
IRAC 4.5 $\mu\text{m}$	1.346	1	0.14
IRAC 5.8 $\mu\text{m}$	1.346	1	0.14
IRAC 8.0 $\mu\text{m}$	1.346	1	0.14
MIPS 24 $\mu\text{m}$	1.402	1	0.14
MIPS 70 $\mu\text{m}$ small	1.702	1.12	1.1
MIPS 70 $\mu\text{m}$ large	2.9	2.65	2.6
MIPS 160 $\mu\text{m}$	3.9	3.75	3.7
MIPS SED	1.702	1.15	1.1

Table 2.2: Derived Requirements for Focal Plane Survey [9]

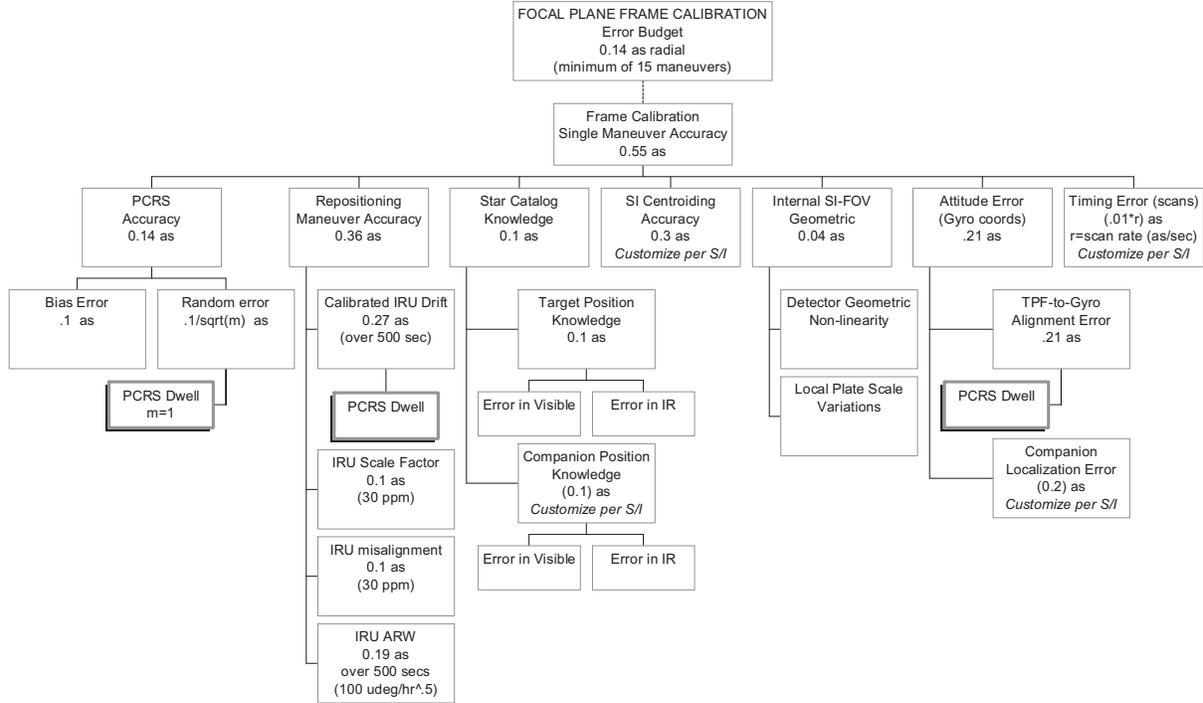


Figure 2.2: Spitzer IPF Estimation Error Budget for Fine Focal Plane Surveys

## 2.4 Input-Output Data Interface Requirements

The overall input and output data interfaces for a standard IPF filter run are summarized in Table 2.3 and are depicted pictorially in Figure 2.3. As shown, the user specifies the six input files (AA, AS, CA, CB, CS and FF Files with their unique 6 digit extensions) and the filter execution configuration choices in the user run file (RN File) prior to each filter execution. The IPF filter is then run with these input files and autonomously produces two output files, namely the IF and LG Files. The 6 digit extension *yyyzzz* is unique to each filter run and identical for RN, IF and LG files. The extension *zzz* signifies the instrument NF number (e.g., 095 for the MIPS 24 micron array) and *yyy* is the file version number that should be incremented each time a new file version of the same type is created.

A special IPF Multi-Run Tool (see Section 6.8), provides the capability to estimate parameters using data sets taken over disparate (i.e., non-contiguous) time intervals. This capability is needed primarily by the MIPS instrument, for which calibration surveys are long and might not be completed in a single observing session. The overall input/output data interfaces for an IPF Multi-Run are shown in Table 2.4. Here, MR is the main run file (rather than RN). As input, the Multi-Run uses the MT and LG files generated from running the IPF filter on each data set individually. The resulting MT files are listed at the beginning of the MR file to direct the automated multi-run processing. The output file is denoted as MF (rather than IF), and is stored into the mission archive.

Name	Description	Type	Format	Source	Destination
RN	Run Configuration	Run	MATLAB (.m)	USER	IPF
AA	Attitude History	Input	Binary (.bin)	FTP	IPF
AS	Attitude Supplemental	Input	MATLAB (.m)	FTP	IPF
CA	Centroid Instrument	Input	MATLAB (.m)	DOM	IPF
CB	Centroid PCRS	Input	MATLAB (.m)	DOM	IPF
CS	Centroid Supplemental	Input	MATLAB (.m)	DOM	IPF
FF	Offset Data	Input	MATLAB (.m)	FTP	IPF
AC	Compact Attitude	Intermediary	MATLAB (.m)	IPF	IPF
AG	Gyro Sensitivity	Intermediary	MATLAB (.m)	IPF	IPF
MT	Workspace	Intermediary	MATLAB (.mat)	IPF	IPF
IF	IPF Output	Output	ASCII (.dat)	IPF	DOM
LG	IPF LOG	Output	ASCII (.dat)	IPF	DOM
TR	Archival	Output	TAR (.tar)	LOCAL	DOM

Table 2.3: IPF Filter Input-Output Data Files

Name	Description	Type	Format	Source	Destination
MR	Multi-Run Configuration	Run	MATLAB (.m)	USER	IPF
MT	Workspace	Input	MATLAB (.mat)	IPF	IPF
FF	Offset Data	Input	MATLAB (.m)	FTP	IPF
LG	IPF LOG	Input	ASCII (.dat)	IPF	IPF& DOM
LG	IPF LOG (multi-run)	Output	ASCII (.dat)	IPF	DOM
MF	IPF Output (multi-run)	Output	ASCII (.dat)	IPF	DOM
TR	Archival	Output	TAR (.tar)	LOCAL	DOM

Table 2.4: Multi-Run Input-Output Data Files

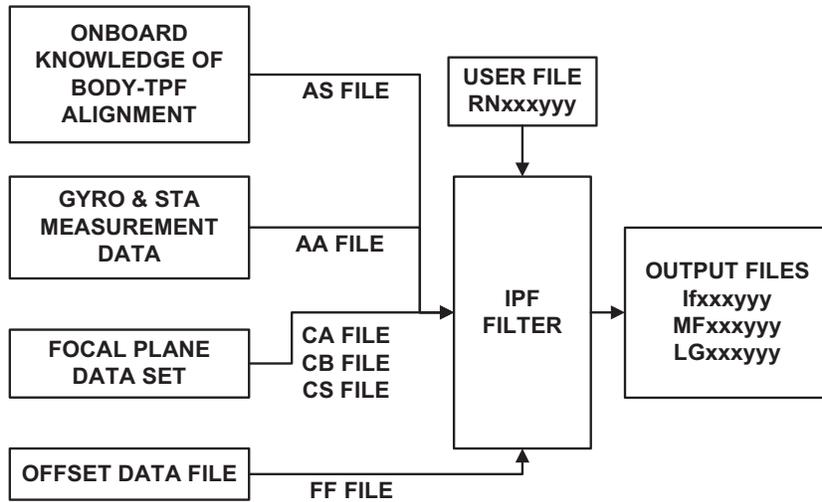


Figure 2.3: Input/Output Description of IPF Filter

### 3 PRELIMINARY DEFINITIONS

#### 3.1 Overview of Pointing-Relevant Frames

The main frames relevant to Spitzer pointing are shown in Figure 3.1. Here the focal plane is shown projected on the sky, as viewed by an observer who is located inside the celestial sphere. The key transformations between these frames are summarized in Table 3.5. For simplicity in presentation, the transformations  $A, R, T, C$  will denote  $3 \times 3$  direction cosine matrices. (This is in contrast to the software implementation which uses quaternions for all numerical computations, but involves less recognizable expressions).

Transformation	Description	From	To
A	Attitude	ICRS	Body
R	Alignment	Body	TPF
T	Instrument	TPF	$IPF_0$
C	Scan Mirror Offset	$IPF_0$	$IPF_\Gamma$

Table 3.5: IPF Filter Transformations

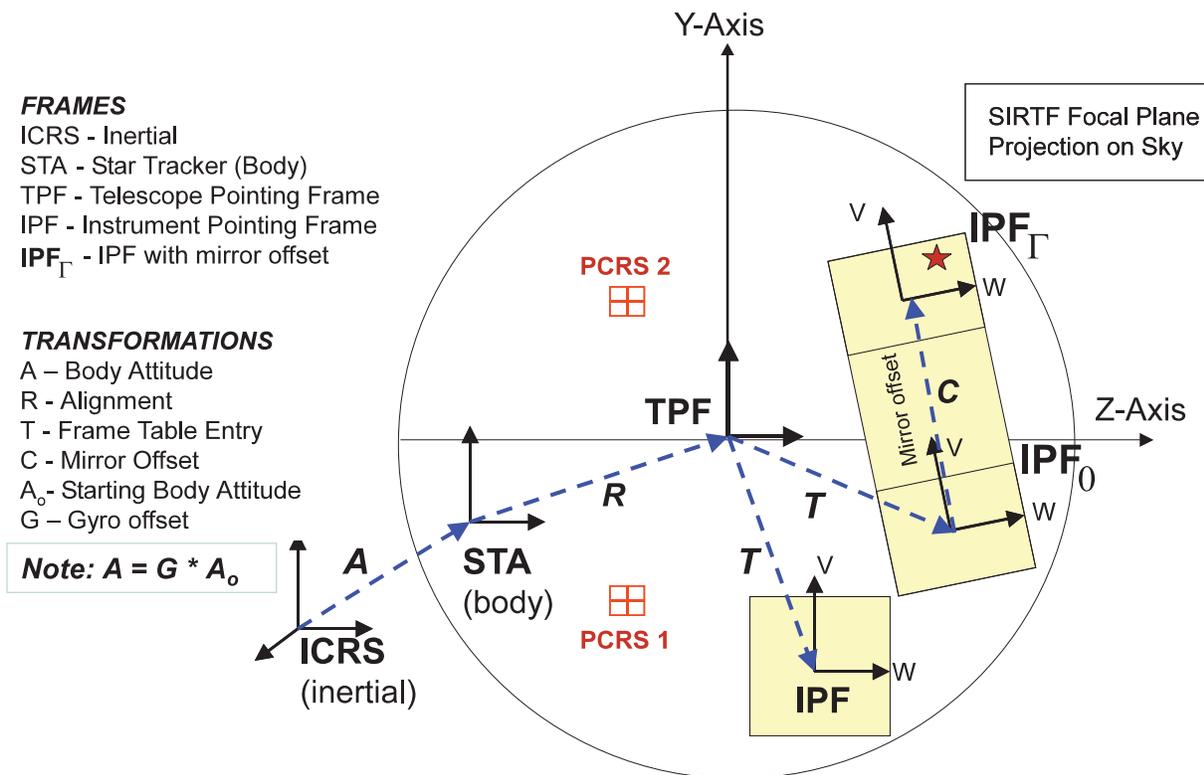


Figure 3.1: Spitzer Frames and Transformations

The International Celestial Reference System (ICRS) frame serves as Spitzer’s principle inertial reference frame. With a suitable relabelling, the star-tracker instrument frame serves as the Spitzer Body frame (i.e., when spelled with its boresight as the  $x$  axis). The mapping from ICRS to the Body Frame is denoted as the spacecraft attitude  $A$ . During each sandwich maneuver, only gyro propagated attitude solutions are used by the IPF filter to reconstruct attitude. The current attitude  $A$  is attained from a gyro propagated offset  $G$  relative to a starting attitude  $A_0$ , i.e.,

$$A = GA_0 \tag{3.1}$$

where  $A_0$  is available from the on-board attitude estimate. The Telescope Pointing Frame (TPF) has the telescope boresight as its  $x$  axis, and is defined rigorously in terms of the null points of the two PCRS sensors in [3]. Specifically, the TPF is defined by a fixed (3,2,1) Euler rotation from the line-of-centers frame (constructed by bisecting and crossing the two PCRS boresight vectors). The mapping from the Body Frame to the TPF is denoted as the alignment matrix  $R$ .

An Instrument Pointing Frame (IPF) is defined by a specific pixel location within a specific science array, such that its coordinate axes adopt the orientation of the corresponding pixel rows and columns of that array. The mapping from the TPF to any specified IPF is denoted generically as  $T$ .

Best estimates of the IPF frames are stored in an on-board “Frame Table” as 128 values for  $T$  (stored as quaternions). The Frame Table is used extensively for commanding purposes. Certain important IPF frames are denoted as **Prime Frames** (typically defined at the center pixel location of each instrument array). Other frames are called **Inferred Frames** and are defined by a pixel offset relative to a nearby Prime frame. The nominal orientations of the science instruments and their associated Prime frames in the telescope focal plane are shown in Figure 3.3.

The  $C$  matrix represents a scan mirror offset from a nominal starting position  $\Gamma = 0$  to its current local offset position  $\Gamma \neq 0$ . For non-MIPS instruments, the  $C$  matrix is set to identity. For MIPS, the frame defined when the scan mirror is offset by angle  $\Gamma$  is denoted as  $IPF_\Gamma$ . Note that as the scan mirror moves there is an entire family of  $IPF_\Gamma$  frames generated as a continuous function of the variable  $\Gamma$ .

The attitude  $A$  is time-varying due to intentional telescope repositioning and unintentional control errors. The alignment matrix  $R$  is time-varying due to thermo-mechanically induced alignment drift. The mapping  $T$  from TPF to IPF is assumed constant due to the fact that the telescope focal plane is actively cooled. The mapping  $C$  is time-varying due to a constantly changing (but nominally known) scan-mirror offset angle  $\Gamma$ .

## 3.2 Body Frame Definition

The Spitzer Body Frame corresponds physically to the star-tracker instrument frame, but labelled such that the star-tracker boresight is the  $x$  axis (in contrast to the star-tracker instrument which labels its own boresight as the  $z$  axis). To distinguish between these two cases, the star tracker instrument frame will be denoted as the STA frame (labelling  $z$  as its boresight), and the star-tracker-defined body frame will be denoted as the STA-defined Body frame, or simply Body frame (labelling the  $x$  axis as its boresight).

Mathematically, the full direction cosine matrix transforming from the Body frame (as used here), to the STA frame (as given in the on-board frame table) is given by,

$$[v]_{STA} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} [v]_{Body} \quad (3.2)$$

where  $v$  is any physical vector of interest, and  $[v]_{Body}$  and  $[v]_{STA}$  denote the resolution of  $v$  in the Body and STA frames, respectively. For example, let  $v$  be the star tracker boresight direction. Then it is seen that the  $x$ -axis boresight label associated with the Body frame (i.e.,  $[v]_{Body} = [1, 0, 0]^T$ ) maps into the  $z$ -axis boresight label associated with the STA frame (i.e.,  $[v]_{STA} = [0, 0, 1]^T$ ). Similarly, the  $y$  and  $z$  axes of the Body frame are relabelled to become the  $x$  and  $y$  axes, respectively, of the STA frame.

### 3.3 Telescope Point Frame (TPF) Definition

The TPF is defined by the null-points of two Pointing Control Reference Sensors (PCRS), which are located in the telescope focal plane to serve as fiducial points for calibration. As shown in Figure 3.2, the null points of these two PCRS detectors (1-A and 2-A) provide two unit vectors. By bisecting these two unit vectors, a Line-Of-Centers (LOC) frame can be defined as shown in this figure. The TPF is defined relative to the LOC frame in terms of the three Euler angles (see (3.7)),

$$\beta_1 = -.2086390720014826 \quad (deg) \quad (3.3)$$

$$\beta_2 = +2.609509909926385 \quad (arcmin) \quad (3.4)$$

$$\beta_3 = -10.96500433037898 \quad (arcsec) \quad (3.5)$$

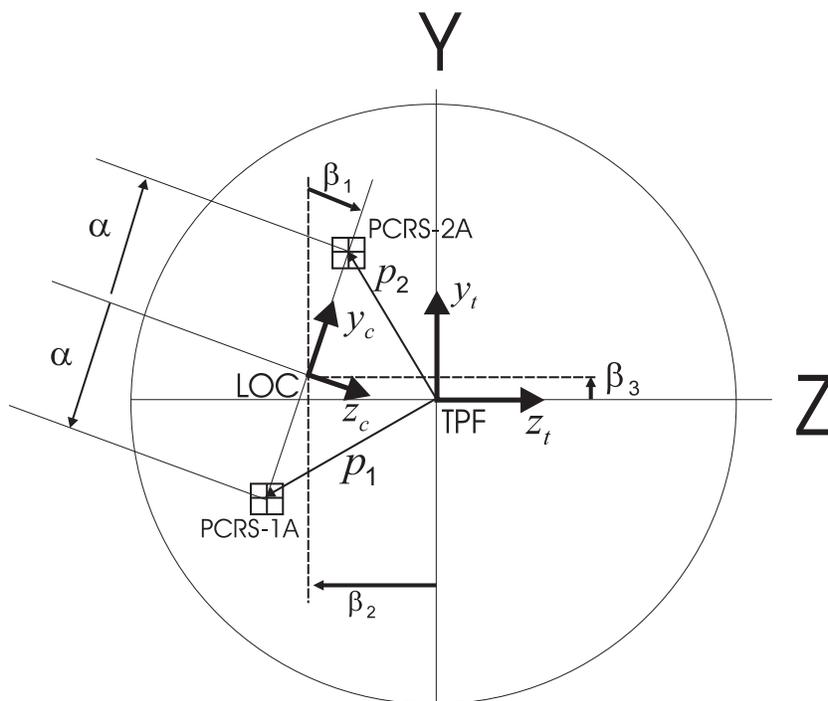


Figure 3.2: Definition of TPF frame in terms of  $\beta$  angles, shown projected onto sky, and looking from inside the celestial sphere.

A more detailed mathematical definition of the TPF is given in the two steps below:

**Step 1:** Let  $p_1$  and  $p_2$  be the Line-of-Sight vectors of PCRS1 and PCRS2, respectively. Define the **Line-of-Centers** (LOC) Frame  $\mathcal{F}_c = \{x_c, y_c, z_c\}$  according to,

$$x_c = (p_1 + p_2)/\|p_1 + p_2\|$$

$$z_c = (p_1 \times p_2)/\|p_1 \times p_2\|$$

$$y_c = z_c \times x_c$$

**Step 2:** Define the **TPF frame**  $\mathcal{F}_t = \{x_t, y_t, z_t\}$  relative to  $\mathcal{F}_c$  such that a 3,2,1 Euler rotation sequence acting on frame  $\mathcal{F}_t$  gives frame  $\mathcal{F}_c$ .

Specifically, if  $[u]_t$  denotes an arbitrary vector resolved in  $\mathcal{F}_c$ , then its resolution  $[u]_c$  in  $\mathcal{F}_c$  is given by,

$$[u]_c = L[u]_t \tag{3.6}$$

$$L = Q_1(\beta_1)Q_2(\beta_2)Q_3(\beta_3) \tag{3.7}$$

where  $\beta_1, \beta_2, \beta_3$  are Euler angles specified in (3.3)(3.4)(3.5), and  $L$  is the direction cosine matrix for the mapping from TPF to the LOC frame.

### 3.4 Instrument Pointing Frame (IPF) Definition

An Instrument Pointing Frame (IPF) is defined by a specific pixel location in the science array which is projected instantaneously onto the sky, and which adopts the orientation of the pixel row and column directions of the array. Spitzer will have up to 128 such IPF frames stored in an on-board frame table. The IPF frames will be stored as quaternion values for  $T$  which define each IPF frame relative to the TPF.

Certain IPF frames which are important enough to be calibrated directly are called **Prime Frames**. Prime frames are typically located at the center of instrument arrays or entrance apertures, and are desired to be known accurately. Other important frames are inferred by their proximity to the Prime frames. Such frames are called **Inferred Frames** and their estimation is discussed in detail in Section 6.6. Best estimates of the IPF frames are stored in an on-board “Frame Table” as 128 values for  $T$  (stored as quaternions). The Frame Table is used extensively for commanding purposes.

The nominal orientation of the science instruments and their associated Prime frames in the telescope focal plane are shown in Figure 3.3. Also shown are the associated  $w$  and  $v$  directions associated with each frame. Angular offsets in the  $(w, v)$  directions define an Oriented Angular Pixel (OAP) coordinate system which is used for calibration (see Section 3.9). It can be seen that many of the arrays are tilted with respect to TPF frame, and that in all cases the  $(w, v)$  directions have been defined such that the  $+w$  axis is within  $\pm 90$  degrees of the TPF  $z_t$  axis.

Also shown are the mission-accepted conventions for the  $w$  and  $v$  directions, defined for each frame. Specifically, each IPF frame is defined by the  $u, v, w$  coordinate axes, where  $v, w$  are shown and  $u = v \times w$  points outward to the sky. The main goal of the IPF Kalman filter (as relevant to supporting on-board pointing capability) is to accurately estimate the IPF frame  $T$  for each of the 128 Prime and Inferred frames listed in the on-board Frame Table.

Currently, there are 19 Prime frames defined on the focal plane. Generally these Prime frames are fixed on the science array (for example, IRAC, IRS Peak-Up Array etc.). The MIPS instrument is slightly more complicated since a Prime frame can only be uniquely defined when its scan mirror is held at some agreed upon angular position. Some instruments (IRS Slits, MIPS SED) have spectroscopy slits whose Prime frames are actually defined by their entrance apertures and not in pixel coordinates. The Prime frames for these instruments are fixed to a point (generally at the center) in the entrance aperture, and are determined by scanning a source across the entrance aperture and recording the time of peak intensity. The objective of the IPF filter is to estimate all Prime frames and inferred frames, or equivalently, to estimate the mappings  $T$  from TPF to each IPF.

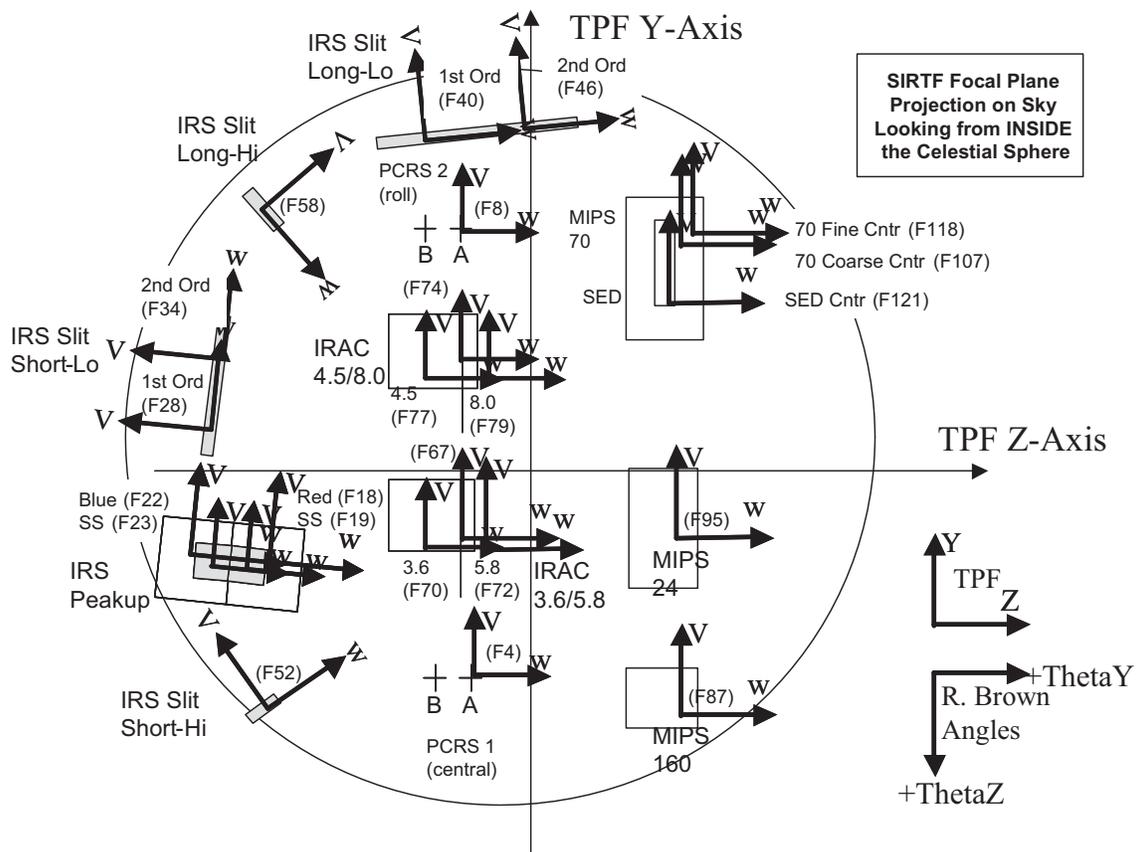


Figure 3.3: Spitzer Instrument Prime Frames Definition

### 3.5 Spitzer Focal Plane Survey Frame Table Definition

A example frame table is shown in Table 3.6.

FOV Index	Frame Type	FOV Name	Brown Angles			Comments
			theta_Y (arcmin)	theta_Z (arcmin)	angle (deg)	
1	NA	Telescope_Boresight	0.0000	0.0000	0.0000	Center of focal plane
2	OET	C7A_frame	0.0000	0.0000	0.0000	Z axis is aligned with the solar array vertex
3	OET	Undefined	0.0000	0.0000	0.0000	
4	OET	PCRS.1.A	-2.5750	9.6570	0.0000	Center of (Central) 1A PCRS Detector
5	OET	PCRS.1.B	-3.5915	9.6585	0.0000	Center of (Central) 1B PCRS Detector
6	OET	Undefined	0.0000	0.0000	0.0000	
7	OET	Undefined	0.0000	0.0000	0.0000	
8	OET	PCRS.2.A	-2.6440	-9.2915	0.0000	Center of (Roll) 2A PCRS Detector
9	OET	PCRS.2.B	-3.6605	-9.2935	0.0000	Center of (Roll) 2B PCRS Detector
10	OET	Undefined	0.0000	0.0000	0.0000	
11	OET	STA1_boresight	0.0000	-5400.0000	90.0000	Center of STA1 boresight
12	OET	STA2_boresight	0.0000	-5400.0000	90.0000	Center of STA2 boresight
13	OET	HGA_boresight	480.0000	10800.0000	0.0000	Center of HGA boresight
14	OET	Plus_Y_LGA_boresight	0.0000	-5400.0000	0.0000	Center of Plus Y LGA boresight
15	OET	Minus_Y_LGA_boresight	0.0000	5400.0000	0.0000	Center of Minus Y LGA boresight
16	OET	Undefined	0.0000	0.0000	0.0000	
17	OET	Undefined	0.0000	0.0000	0.0000	
18	PRIME	IRS_Red_Peak-Up_FOV_Center	-11.7080	2.1380	2.9600	18.5-26 $\mu$ P/U FOV Center Position
19	PRIME	IRS_Red_Peak-Up_FOV_Sweet_Spot	-11.6496	2.1709	2.9600	18.5-26 $\mu$ P/U FOV Sweet Spot Position (found out during IOC)
20	SSC	Undefined	0.0000	0.0000	0.0000	
21	SSC	Undefined	0.0000	0.0000	0.0000	
22	PRIME	IRS_Blue_Peak-Up_FOV_Center	-13.7500	2.0370	2.9600	13-18.5 $\mu$ P/U FOV Center Position
23	PRIME	IRS_Blue_Peak-Up_FOV_Sweet_Spot	-13.8114	2.0640	2.9600	13-18.5 $\mu$ P/U FOV Sweet Spot Position (found out during IOC)
24	SSC	Undefined	0.0000	0.0000	0.0000	
25	SSC	Undefined	0.0000	0.0000	0.0000	
26	INF28	IRS_Short-Lo_Ist_Order_Ist_Position	-12.1800	-2.5140	275.2800	Shortest Wavelength Position on the 7.5-15 $\mu$ Sub-band
27	INF28	IRS_Short-Lo_Ist_Order_2nd_Position	-12.1520	-2.8240	275.2800	Longest Wavelength Position on the 7.5-15 $\mu$ Sub-band
28	PRIME	IRS_Short-Lo_Ist_Order_Center_Position	-12.1660	-2.6690	275.2800	Center Position on the 7.5-15 $\mu$ Sub-band
29	DER28,34	IRS_Short-Lo_Module_Center	-12.1050	-3.3300	275.2800	
30	SSC	Undefined	0.0000	0.0000	0.0000	
31	SSC	Undefined	0.0000	0.0000	0.0000	

Table 3.6: Example Frame Table PART#1 (VER17)

FOV Index	Frame Type	FOV Name	Brown Angles			Comments
			theta_Y (arcmin)	theta_Z (arcmin)	angle (deg)	
32	INF34	IRS_Short-Lo_2nd_Order_1st_Position	-12.0590	-3.8360	275.2800	Shortest Wavelength Position on the 5-7.5 $\mu$ Sub-band
33	INF34	IRS_Short-Lo_2nd_Order_2nd_Position	-12.0300	-4.1460	275.2800	Longest Wavelength Position on the 5-7.5 $\mu$ Sub-band
34	PRIME	IRS_Short-Lo_2nd_Order_Center_Position	-12.0450	-3.9910	275.2800	Center Position on the 5-7.5 $\mu$ Sub-band
35	SSC	Undefined	0.0000	0.0000	0.0000	
36	SSC	Undefined	0.0000	0.0000	0.0000	
37	SSC	Undefined	0.0000	0.0000	0.0000	
38	INF40	IRS_Long-Lo_1st_Order_1st_Position	-4.1640	-13.8990	358.8000	Shortest Wavelength Position on the 21-40 $\mu$ Sub-band
39	INF40	IRS_Long-Lo_1st_Order_2nd_Position	-5.0480	-13.8800	358.8000	Longest Wavelength Position on the 21-40 $\mu$ Sub-band
40	PRIME	IRS_Long-Lo_1st_Order_Center_Position	-4.6060	-13.8890	358.8000	Center Position on the 21-40 $\mu$ Sub-band
41	DER40,46	IRS_Long-Lo_Module_Center	-2.9020	-13.9250	358.8000	
42	SSC	Undefined	0.0000	0.0000	0.0000	
43	SSC	Undefined	0.0000	0.0000	0.0000	
44	INF46	IRS_Long-Lo_2nd_Order_1st_Position	-0.7560	-13.9690	358.8000	Shortest Wavelength Position on the 14-21 $\mu$ Sub-band
45	INF46	IRS_Long-Lo_2nd_Order_2nd_Position	-1.6400	-13.9510	358.8000	Longest Wavelength Position on the 14-21 $\mu$ Sub-band
46	PRIME	IRS_Long-Lo_2nd_Order_Center_Position	-1.1980	-13.9600	358.8000	Center Position on the 14-21 $\mu$ Sub-band
47	SSC	Undefined	0.0000	0.0000	0.0000	
48	SSC	Undefined	0.0000	0.0000	0.0000	
49	SSC	Undefined	0.0000	0.0000	0.0000	
50	INF52	IRS_Short-Hi_1st_Position	-10.7890	10.1540	318.5300	Shortest Wavelength Position
51	INF52	IRS_Short-Hi_2nd_Position	-10.7400	10.1120	318.5300	Longest Wavelength Position
52	PRIME	IRS_Short-Hi_Center_Position	-10.7640	10.1330	318.5300	Center Position
53	SSC	Undefined	0.0000	0.0000	0.0000	
54	SSC	Undefined	0.0000	0.0000	0.0000	
55	SSC	Undefined	0.0000	0.0000	0.0000	
56	INF58	IRS_Long-Hi_1st_Position	-10.5890	-10.2010	43.3400	Shortest Wavelength Position
57	INF58	IRS_Long-Hi_2nd_Position	-10.4990	-10.1150	43.3400	Longest Wavelength Position
58	PRIME	IRS_Long-Hi_Center_Position	-10.5440	-10.1580	43.3400	Center Position
59	SSC	Undefined	0.0000	0.0000	0.0000	
60	SSC	Undefined	0.0000	0.0000	0.0000	
61	SSC	Undefined	0.0000	0.0000	0.0000	
62	SSC	Undefined	0.0000	0.0000	0.0000	could be used by either IRS or IRAC
63	SSC	Undefined	0.0000	0.0000	0.0000	could be used by either IRS or IRAC
64	SSC	Undefined	0.0000	0.0000	0.0000	could be used by either IRS or IRAC
65	SSC	Undefined	0.0000	0.0000	0.0000	could be used by either IRS or IRAC
66	SSC	Undefined	0.0000	0.0000	0.0000	could be used by either IRS or IRAC

Table 3.7: Example Frame Table PART#2 (VER17)

FOV Index	Frame Type	FOV Name	Brown Angles			Comments
			theta_Y (arcmin)	theta_Z (arcmin)	angle (deg)	
67	DER68,69	IRAC_Center_of_3.6&5.8umArray	-3.1200	3.5750	0.0000	Center of the full 256 x 256 pix. 3.6 & 5.8 $\mu$ Array's FOV
68	PRIME	IRAC_Center_of_3.6umArray	-3.1240	3.6340	0.0000	Center of the full 256 x 256 pix. 3.6 $\mu$ Array's FOV
69	PRIME	IRAC_Center_of_5.8umArray	-3.1170	3.5160	0.0000	Center of the full 256 x 256 pix. 5.8 $\mu$ Array's FOV
70	INF68	IRAC_Center_of_3.6umSub-array	-5.1950	5.6490	0.0000	Center of the 32 x 32 pix. 3.6 & 5.8 $\mu$ Sub-array's FOV, whose lower left corner is at pixel (4,4) in the 3.6 & 5.8 $\mu$ Full Array
71	SSC	Undefined	0.0000	0.0000	0.0000	
72	INF69	IRAC_Center_of_5.8umSub-array	-1.0490	5.6460	0.0000	Center of the 32 x 32 pix. 3.6, 5.8 $\mu$ Sub-array's FOV, whose lower left corner is at pixel (4,4) in the 3.6 & 5.8 $\mu$ Full Array
73	SSC	Undefined	0.0000	0.0000	0.0000	
74	DER75,76	IRAC_Center_of_4.5&8.0umArray	-2.9700	-2.9700	0.0000	Center of the full 256 x 256 pix. 4.5 & 8.0 $\mu$ Array's FOV
75	PRIME	IRAC_Center_of_4.5umArray	-2.9600	-2.9500	0.0000	Center of the full 256 x 256 pix. 4.5 $\mu$ Array's FOV
76	PRIME	IRAC_Center_of_8.0umArray	-2.9800	-2.9910	0.0000	Center of the full 256 x 256 pix. 8.0 $\mu$ Array's FOV
77	INF75	IRAC_Center_of_4.5umSub-Array	-5.0410	-0.8980	0.0000	Center of the 32 x 32 pix. 4.5, 8.0 $\mu$ Sub-array, whose lower left corner is at pixel (4,4) in the 4.5 & 8.0 $\mu$ Full Array
78	SSC	Undefined	0.0000	0.0000	0.0000	
79	INF76	IRAC_Center_of_8.0umSub-Array	-0.8970	-0.8970	0.0000	Center of the 32 x 32 pix. 4.5 & 8.0 $\mu$ Sub-array, whose lower left corner is at pixel (4,4) in the 4.5 & 8.0 $\mu$ Full Array
80	SSC	Undefined	0.0000	0.0000	0.0000	
81	DER68, 69, 75, 76	IRAC_FOV_bwn_3.6,5.8,4.5,8.0 for mapping	-3.0450	-0.3020	0.0000	IRAC FOV between 3.6 & 5.8 and 4.5 & 8.0 for mapping
82	SSC	Undefined	0.0000	0.0000	0.0000	could be used by either IRAC or MIPS
83	SSC	Undefined	0.0000	0.0000	0.0000	could be used by either IRAC or MIPS
84	SSC	Undefined	0.0000	0.0000	0.0000	could be used by either IRAC or MIPS
85	SSC	Undefined	0.0000	0.0000	0.0000	could be used by either IRAC or MIPS
86	SSC	Undefined	0.0000	0.0000	0.0000	could be used by either IRAC or MIPS
87	PRIME	MIPS_160um_center_&_large_FOVs	6.6690	11.5170	0.0000	Center of the full 160 $\mu$ Array's FOV, used for SCAN
88	INF87	MIPS_160um_plusY_edge	6.6640	11.9740	0.0000	Position on the +y = +v edge of the FOV, on the z=w=0 median line
89	INF87	MIPS_160um_large_only	6.6690	11.5170	0.0000	160um large, same as center, use for PH/SR visualization
90	SSC	Undefined	0.0000	0.0000	0.0000	
91	INF87	MIPS_160 $\mu$ m_small_FOV1	7.5890	11.5460	0.0000	Position Offset: Y=0, Z=+3.5pixels
92	INF87	MIPS_160um_small_FOV2	5.8820	11.3400	0.0000	Position Offset: Y=+0.5, Z=-3 pixels
93	SSC	Undefined	0.0000	0.0000	0.0000	
94	SSC	Undefined	0.0000	0.0000	0.0000	
95	PRIME	MIPS_24um_center	6.6410	3.9310	0.0000	Center of the full 24 $\mu$ Array's FOV
96	INF95	MIPS_24um_plusY_edge	6.5950	6.7120	0.0000	Position on the +y=+v edge of the FOV, on the z=w=0 median line

Table 3.8: Example Frame Table PART#3 (VER17)

FOV Index	Frame Type	FOV Name	Brown Angles			Comments
			theta_Y (arcmin)	theta_Z (arcmin)	angle (deg)	
97	SSC	Undefined	0.0000	0.0000	0.0000	
98	SSC	Undefined	0.0000	0.0000	0.0000	
99	INF95	MIPS_24um_small_FOV1	7.6750	4.6600	0.0000	Position Offset: y=v=-16.5, z=w=25 pixels, relative to center
100	INF95	MIPS_24um_small_FOV2	5.5620	4.6360	0.0000	Position Offset: y=v=-16.5, z=w=-25.5 pixels, relative to center
101	SSC	Undefined	0.0000	0.0000	0.0000	
102	SSC	Undefined	0.0000	0.0000	0.0000	
103	INF95	MIPS_24um_large_FOV1	6.7460	3.9320	0.0000	Position Offset: y=v=0, z=w=+2.5 pixels, relative to center
104	INF95	MIPS_24um_large_FOV2	6.5580	3.9300	0.0000	Position Offset: y=v=0, z=w= - 2pixels, relative to center
105	SSC	Undefined	0.0000	0.0000	0.0000	
106	SSC	Undefined	0.0000	0.0000	0.0000	
107	PRIME	MIPS_70um_center	6.4450	-8.5920	0.0000	Center of the full 70 $\mu$ Array's FOV
108	INF107	MIPS_70um_minusY_edge	6.3480	-11.2590	0.0000	Position on the -y= -v edge of FOV, on the z=w=0 median line
109	SSC	Undefined	0.0000	0.0000	0.0000	
110	SSC	Undefined	0.0000	0.0000	0.0000	
111	INF107	MIPS_70um_default_small_FOV1	7.4230	-8.6470	0.0000	Position Offset: y=v=0, z=w=+6pixels, relative to center
112	INF107	MIPS_70um_default_small_FOV2	5.3850	-8.5330	0.0000	Position Offset: y=v=0, z=w= - 6.5pixels*, relative to center
113	INF121	MIPS_SED_5	6.9270	-14.2560	0.0000	SED+3 chop FOV 1, offset Y=-13.2, Z=+4 pixels
114	INF121	MIPS_SED_6	5.6150	-14.2500	0.0000	SED+3 chop FOV 2, offset Y=-13.2, Z=-4 pixels
115	INF107	MIPS_70um_default_large_FOV1	6.1010	-8.1550	0.0000	Position Offset: y=v=2.5, z=w=-4.2, relative to center
116	INF121	MIPS_SED_7	6.6630	-10.0500	0.0000	SED-1 chop FOV 1, offset Y=+4.4, Z=+4 pixels
117	INF121	MIPS_SED_8	5.3510	-10.0430	0.0000	SED-1 chop FOV 2, offset Y=+4.4, Z=-4 pixels
118	PRIME	MIPS_70um_fine_center	6.8590	-8.8470	0.0000	center of 70um fine scale field
119	INF118	MIPS_70um_fine_FOV1	6.5460	-8.5790	0.0000	Position Offset: y=v=2.5, z=w=-2.1 pixels, relative to center
120	INF118	MIPS_70um_fine_FOV2	6.9410	-4.1170	0.0000	Position Offset: y=v=26, z=w=-3.6 pixels, relative to center
121	PRIME	MIPS_SED_center	6.0730	-11.0980	0.0000	Y=default 70um, Z=default 70 um + 0.9arcmin
122	INF121	MIPS_SED_1	6.7950	-12.1530	0.0000	SED+1 chop FOV 1, offset Y=-4.4, Z=+4 pixels
123	INF121	MIPS_SED_2	5.4830	-12.1460	0.0000	SED+1 chop FOV2, offset Y=-4.4, Z=-4 pixels
124	INF118	MIPS_70um_fine_FOV3	6.3720	-8.9810	0.0000	Position Offset: y=v=2.5, z=w=-3.6pixels, relative to center
125	INF121	MIPS_SED_3	6.8610	-13.2040	0.0000	SED+2 chop FOV 1, offset Y=-8.8, Z=+4 pixels
126	INF121	MIPS_SED_4	5.5490	-13.1980	0.0000	SED+2 chop FOV 2, offset Y=-8.8, Z=-4 pixels
127	INF118	MIPS_70um_fine_FOV4	6.5210	-11.3780	0.0000	Position Offset: y=v=-+30, z=w=-0.5pixels, relative to center
128	NA	Reserved for PCS DEF FRAME	0.0000	0.0000	0.0000	Defined by latest PCS DEF FRAME CMD and accessed by next PCS POINT CMD

Table 3.9: Example Frame Table PART#4 (VER17)

### 3.6 Scan-Offset $IPF_\Gamma$ Frame Definition

Some instruments (for example, MIPS) employ a single-axis scan mirror that can be freely commanded and rotated relative to a nominal reference position. For notational purposes, the amount of scan mirror rotation beyond its nominal reference position is denoted by  $\Gamma$ . The value  $\Gamma = 0$  corresponds to the case where the scan mirror is at its nominal reference position, for which the IPF frame is denoted as  $IPF_0$ .

The value  $\Gamma \neq 0$  corresponds to a non-zero angular offset of the instrument frame relative to its nominal reference position achieved by rotating the scan mirror a fixed amount. For calibration purposes, the quantity  $\Gamma$  is measured *in units of radians on the sky*. The transformation involved in obtaining  $\Gamma$  from actual encoder units is given in Section B.3. The IPF frame corresponding to a non-zero value of  $\Gamma$  is denoted as  $IPF_\Gamma$ .

As shown in Figure 3.1, the direction cosine matrix  $T$  represents the mapping from TPF to  $IPF_0$ . The direction cosine matrix  $C$  represents the mapping from  $IPF_0$  to  $IPF_\Gamma$ . It is convenient to represent  $C$  using an angle/axis representation. Specifically, given a vector  $u \in \mathcal{R}^3$  resolved in  $IPF_0$ , the same vector  $v \in \mathcal{R}^3$  resolved in  $IPF_\Gamma$  can be written as:

$$v = C(a_m, \Gamma) u \quad (3.8)$$

$$C(a_m, \Gamma) = \cos(\beta\Gamma) \cdot I + (1 - \cos(\beta\Gamma)) a_m a_m^T - \sin(\beta\Gamma) a_m^\times \quad (3.9)$$

where  $a_m = [a_{m1} \ a_{m2} \ a_{m3}]^T$  is the scan mirror spin axis,  $\Gamma$  is the measured scan mirror angle in units of radians, and  $\beta$  is the scale factor parameter associated with  $\Gamma$  (nominally unity).

### 3.7 Brown Angle Definition

Brown angles are used consistently throughout the Spitzer program due to their convenience in converting the CCD Focal Surface to the Telescope Objective Space Field of View (cf., [6],[19] and [20]). Brown angles are simply the (3, 2, 1) Euler angles associated with the transformation from TPF coordinates to IPF coordinates, except for an important difference regarding their signs and units. Due to the large amount of analysis already performed using these conventions, the angle definition was retained and named after the author who first used them (cf., [6]). The relationship of Brown angles to Euler angles will be given in this section.

Consider the direction cosine matrix  $T$  that transforms from TPF to IPF ( $IPF_0$  for frames associated with instruments having scan mirrors). The matrix  $T$  is parameterized using Euler angles  $(\theta_3, \theta_2, \theta_1)$  in a 3-2-1 sequence, i.e.,

$$T(\theta) \triangleq T_1(\theta_1)T_2(\theta_2)T_3(\theta_3) \quad (3.10)$$

where

$$T_1(\theta_1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_1 & \sin \theta_1 \\ 0 & -\sin \theta_1 & \cos \theta_1 \end{bmatrix} \quad (3.11)$$

$$T_2(\theta_2) = \begin{bmatrix} \cos \theta_2 & 0 & -\sin \theta_2 \\ 0 & 1 & 0 \\ \sin \theta_2 & 0 & \cos \theta_2 \end{bmatrix} \quad (3.12)$$

$$T_3(\theta_3) = \begin{bmatrix} \cos \theta_3 & \sin \theta_3 & 0 \\ -\sin \theta_3 & \cos \theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.13)$$

Given the Euler angles  $(\theta_1, \theta_2, \theta_3)$  in units of *rad*, the Brown angles  $(\theta_{1brown}, \theta_{2brown}, \theta_{3brown})$  [6] are defined by,

$$\begin{bmatrix} \theta_{1brown} \\ \theta_{2brown} \\ \theta_{3brown} \end{bmatrix} = \frac{180}{\pi} \begin{bmatrix} 0 & -60 & 0 \\ 0 & 0 & -60 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} \quad (3.14)$$

It is noted that the Brown angles have units of  $[arcmin \ arcmin \ deg]^T$ , respectively, and that they are obtained by reversing signs on the  $\theta_2$  and  $\theta_3$  Euler angles.

### 3.8 Standard Coordinates

Let  $u \in \mathcal{R}^3$  be a unit vector associated with a star location in the ICRS frame, i.e.,

$$u = \begin{bmatrix} \cos(\text{DEC}) * \cos(\text{RA}) \\ \cos(\text{DEC}) * \sin(\text{RA}) \\ \sin(\text{DEC}) \end{bmatrix} \quad (3.15)$$

where RA, DEC denotes the Right Ascension and Declination of the source (in radians).

Let  $\ell \in \mathcal{R}^3$  denote the unit vector after a velocity aberration correction has been applied (consistent with Spitzer's velocity at the time of the observation - see Section B.1). Then (cf., [21]),

$$\ell = \frac{u + \frac{V_{SC}}{c}}{\|u + \frac{V_{SC}}{c}\|} \quad (3.16)$$

where  $c$  denotes the speed of light, and  $V_{SC}$  denotes the spacecraft velocity vector in ICRS.

Define the vector  $s$  as the resolution of  $\ell$  in the  $IPF_T$  frame to give,

$$s = CTRA\ell \quad (3.17)$$

When the current attitude is the result of a gyro offset  $G$  from an initial attitude  $A_0$  one can decompose  $A$  as,

$$A = GA_0 \quad (3.18)$$

Substituting (3.18) into (3.17) gives,

$$s = CTRGA_0\ell \quad (3.19)$$

Let the components of  $s$  be given as,

$$s = \begin{bmatrix} s_x \\ s_y \\ s_z \end{bmatrix} \quad (3.20)$$

Since  $s$  is a unit vector in the  $IPF_\Gamma$  frame, it can be projected into focal plane coordinates to give,

$$z = \begin{bmatrix} z_w \\ z_v \end{bmatrix} = \begin{bmatrix} s_z/s_x \\ s_y/s_x \end{bmatrix} \quad (3.21)$$

The elements of  $z \in \mathcal{R}^2$  will be said to be in **Standard Coordinates**.

Let the matrices  $C, T, R, G$  be parameterized in terms of the elements of the parameter vector  $p_{2f}$  (to be defined in detail in Section 4), and let  $A_0$  be related to an available initial attitude estimate  $\hat{A}_0$  as follows,

$$A_0 = (I - \psi^\times)\hat{A}_0 \quad (3.22)$$

where  $\psi \in \mathcal{R}^3$  denotes the initial attitude error. Then one can write (3.21) in the functional form,

$$z = h_z(p_{2f}, \psi) \quad (3.23)$$

This representation of the target source location in Standard Coordinates will be the starting point for the calibration process.

### 3.9 Oriented Angular Pixel (OAP) Coordinates

Typically, science centroids are obtained in units of pixels. However, calibration is more easily performed if pixel measurements are converted to units of angle (radians), and expressed with respect to an agreed upon origin and orientation. Oriented Angular Pixel (OAP) coordinates serve this purpose.

A pixel coordinate  $(CX, CY)$  (in the instrument  $(x, y)$  coordinate system) is converted to OAP coordinates using the following transformation (see Appendix B, Section B.2 for details),

$$y = \begin{bmatrix} y_w \\ y_v \end{bmatrix} = \begin{bmatrix} D11 & D12 \\ D21 & D22 \end{bmatrix} \begin{bmatrix} \text{PIX2RADX} & 0 \\ 0 & \text{PIX2RADY} \end{bmatrix} \begin{bmatrix} CX-CX0 \\ CY-CY0 \end{bmatrix} \quad (3.24)$$

Here,  $\text{PIX2RADX}, \text{PIX2RADY}$  are nominal plate scales, and the pixel coordinate  $(\text{CX0}, \text{CY0})$  specifies the desired location where the Prime frame is to be embedded. The quantities  $\text{D11}, \text{D12}, \text{D21}, \text{D22}$  are flip parameters (having values  $0, -1, +1$ ), which specify how to map the instrument  $(x, y)$  coordinate directions into the focal plane  $(w, v)$  coordinate directions defined in Figure 3.3.

In summary, it is seen from (3.24) that Oriented Angular Pixel Coordinates are defined by taking the pixel measurements, shifting them to express their location relative to the desired Prime frame, scaling them to get units of angles (radians), and orienting them to coincide with the  $w, v$  directions in the focal plane.

### 3.10 Mapping OAP to Standard Coordinates

Let  $y_{true} \in \mathcal{R}^2$  be a target source as observed in Oriented Angular Pixel Coordinates assuming that there is no centroiding error,

$$y_{true} = \begin{bmatrix} y_{w,true} \\ y_{v,true} \end{bmatrix} \quad (3.25)$$

Generally,  $y_{true}$  will not coincide exactly with  $z$  in (3.23) due to imperfections in the optical system. To accommodate such imperfections, a model which maps  $y_{true}$  in OAP coordinates to  $z$  in Standard coordinates is taken to be of the form,

$$z = \begin{bmatrix} z_w \\ z_v \end{bmatrix} = (I + M(p_1, \Gamma, y_{true})) \begin{bmatrix} y_{w,true} \\ y_{v,true} \end{bmatrix} \quad (3.26)$$

Here  $M \in \mathcal{R}^{2 \times 2}$  is a perturbation matrix which captures the imperfections such as optical distortions, plate scale errors, etc. The exact form of  $M$  will be discussed in Section 4.2 as a function of the distortion parameters  $p_1$ , the scan mirror offset  $\Gamma$ , and the centroid measurement  $y_{true}$ . The scan mirror offset angle  $\Gamma$  is defined in the Appendix, Section B.3.

The relation (3.26) assumes noiseless centroids. To generalize the model, a noisy centroid measurement  $y$  is introduced,

$$y = \begin{bmatrix} y_w \\ y_v \end{bmatrix} \quad (3.27)$$

The noisy centroid  $y$  is used to replace  $y_{true}$  in (3.26) according to the following relation,

$$z = \begin{bmatrix} z_w \\ z_v \end{bmatrix} = (I + M(p_1, \Gamma, y)) \begin{bmatrix} y_w \\ y_v \end{bmatrix} - \nu \quad (3.28)$$

where  $\nu$  denotes the centroiding error in  $y$ . The motivation for choosing this model is that if  $M$  is small (which should always be the case), equation (3.28) is first-order equivalent to the additive noise model,

$$y \simeq y_{true} + \nu \quad (3.29)$$

### 3.11 Calibration Equation

By equating (3.23) and (3.28) the following Calibration Equation is obtained,

$$\boxed{(I + M(p_1, \Gamma, y)) y = h_z(p_{2f}, \psi) + \nu} \quad (3.30)$$

This is the main equation to be used for all Spitzer focal plane calibration. It is an end-to-end relation that maps the source location (known from a star catalog with velocity correction) to the pixel location where the source is observed on the science instrument array. Accordingly, it contains both optical distortions parameterized by  $p_1$  and systematic pointing errors parameterized by  $p_{2f}$ .

The end-to-end pointing transformations associated with the Calibration Equation (3.30) are summarized in Figure 3.4. The top path maps a star position vector in ICRS coordinates to IPF coordinates through the sequence of  $A$ ,  $R$ ,  $T$  and  $C$  matrices. The geometric projection of the resulting vector (in IPF coordinates) into an ideal 2-dimensional image plane defines two angles associated with Standard Coordinates. Moreover, the centroid measurement of this same star on the detector can be brought into Standard Coordinates using the lower path of Figure 3.4. By mapping the centroid measurement to OAP coordinates, and then through the distortion transformation (the  $M$  matrix) one arrives again in Standard Coordinates. Hence, by equating these two paths, one obtains the Calibration Equation (3.30).

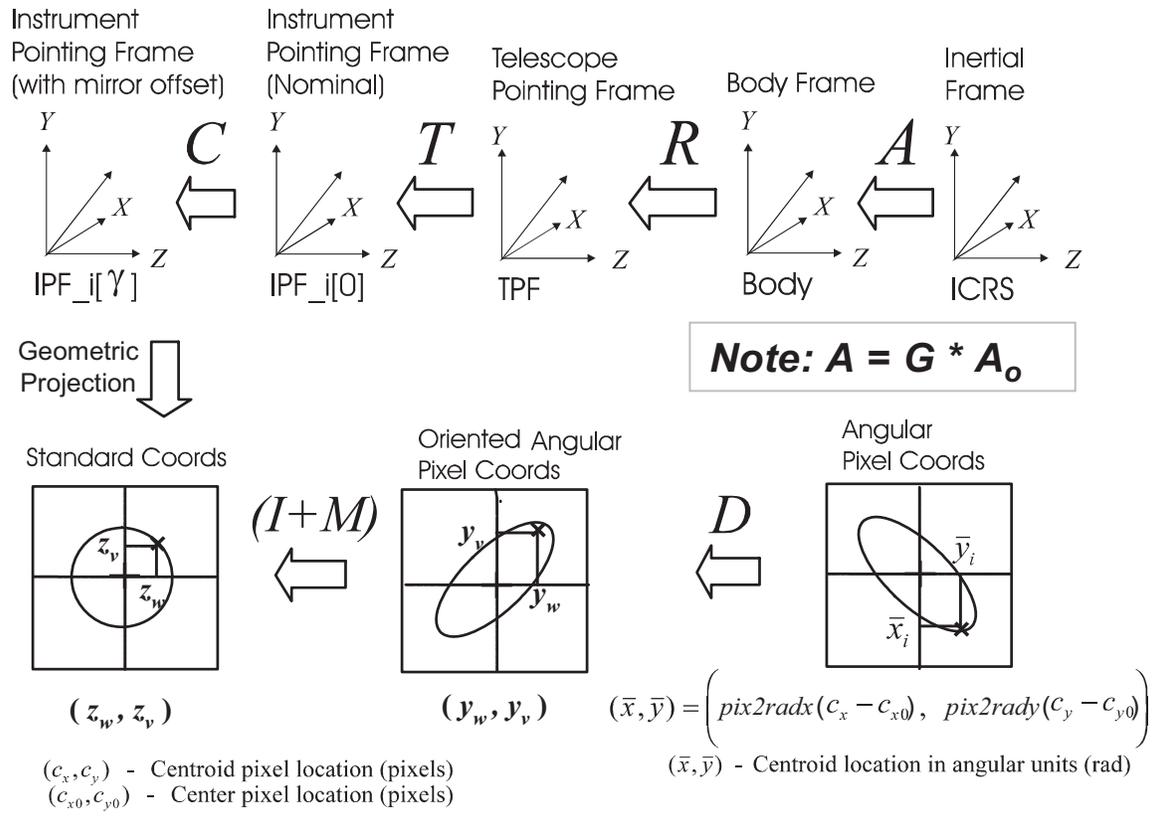


Figure 3.4: End-to-End Pointing Transformations

### 3.12 Pixels to Sky Reconstruction

Instead of equating the upper and lower paths, the transformations shown in Figure 3.4 can be applied in a single direction. This gives an end-to-end parametrization for mapping pixels onto the sky which is useful for addressing the Spitzer reconstruction problem.

Reconstruction can be achieved by the following sequence of calculations. First, a pixel location  $(CX, CY)$  is transformed to  $y$  in Oriented Angular Pixel Coordinates according to (cf., Appendix, Section B.2),

$$y = \begin{bmatrix} y_w \\ y_v \end{bmatrix} = \begin{bmatrix} D11 * (CX - CX0) * PIX2RADX + D12 * (CY - CY0) * PIX2RADY \\ D21 * (CX - CX0) * PIX2RADX + D22 * (CY - CY0) * PIX2RADY \end{bmatrix} \quad (3.31)$$

Second, the mirror encoder data is converted to a mirror offset angle  $\Gamma$  according to (see Appendix, Section B.3),

$$\Gamma = DG * BETA0 * (GAMMA_E - GAMMA_E0) \quad (3.32)$$

Using  $y$  and  $\Gamma$ , the pixel coordinate position is expressed in Standard Coordinates by:

$$z = \begin{bmatrix} z_w \\ z_v \end{bmatrix} = (I + M(p_1, \Gamma, y)) \begin{bmatrix} y_v \\ y_w \end{bmatrix} \quad (3.33)$$

This value of  $z$  is transformed into a unit vector  $s$  in the  $IPF_\Gamma$  frame as follows,

$$s = \frac{1}{\sqrt{1 + z_w^2 + z_v^2}} \cdot \begin{bmatrix} 1 \\ z_v \\ z_w \end{bmatrix} \quad (3.34)$$

The unit vector  $v$  in ICRS associated with  $s$  can be expressed as,

$$v = A^T G^T R^T T^T C^T (p_1, \Gamma) s \quad (3.35)$$

Applying the speed-of-light velocity aberration correction yields,

$$\tilde{v} = \frac{v - \frac{V_{SC}}{c}}{\|v - \frac{V_{SC}}{c}\|} \quad (3.36)$$

Then RA/DEC angles associated with  $\tilde{v}$  can be calculated as,

$$RA = atan2(\tilde{v}(2), \tilde{v}(1)) \quad (3.37)$$

$$DEC = asin(\tilde{v}(3)) \quad (3.38)$$

where the components of  $\tilde{v}$  have been defined as,

$$\tilde{v} = \begin{bmatrix} \tilde{v}(1) \\ \tilde{v}(2) \\ \tilde{v}(3) \end{bmatrix} \quad (3.39)$$

This result captures the complete end-to-end pointing reconstruction transformation from a specified pixel coordinate  $(CX, CY)$  to its RA and DEC location on the sky (as seen by an observer moving with the solar system barycenter).

### 3.13 Time Axis Definition

The IPF reference epoch is marked as  $t = 0$  and is taken at the beginning of each calibration survey data set. As shown in Figure 3.5, there are three different time scales associated with any given calibration data set. The attitude files are time-tagged synchronously at the regular gyro rate of 10Hz, while the centroid data is time-tagged asynchronously at the time instants when the centroids are taken. Specifically, centroid times are marked as  $T_1, T_2, \dots, T_k, \dots$  where  $k$  increases consecutively from the beginning to the end of the data set. A third time scale is created by marking the start time of each sandwich maneuver (occurring approximately every 700 seconds). The start of the  $j$ 'th maneuver is marked by the time tag  $t_j$ .

Time-alignment errors of up to 60 milliseconds are expected between the attitude and centroid data sets.

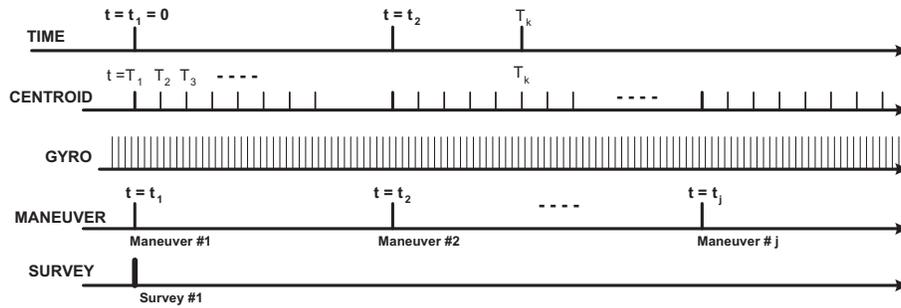


Figure 3.5: Spitzer IPF Time Axis Definition

## 4 IPF FILTER PARAMETERS

### 4.1 Overview

The starting point for the Kalman filter design is the Calibration Equation (3.30) (derived in Section 3.11). A full state vector  $x_f$  is defined as,

$$x_f = \begin{bmatrix} p_1 \\ p_{2f} \end{bmatrix} \quad (4.1)$$

where  $p_1$  are the optical distortion parameters and  $p_{2f}$  are the systematic pointing errors in the Calibration Equation (3.30).

The parameters in  $p_1$  and  $p_{2f}$  are defined such that they are constant with time. A summary of the states  $p_1$  and  $p_{2f}$  is given in Table 4.10 and Table 4.11. The subsets of these parameters that are used for each of the instrument calibrations are indicated in Table 4.12 and Table 4.13.

The next few subsections will be devoted to giving a detailed description of each of these parameters.

$p_1$	Math	Description	$\delta p_1$	Mask
$p_1(1)$	$a_{00}$	Constant Plate Scales	$\delta p_1(1)$	mask1(1)
$p_1(2)$	$b_{00}$		$\delta p_1(2)$	mask1(2)
$p_1(3)$	$c_{00}$		$\delta p_1(3)$	mask1(3)
$p_1(4)$	$a_{10}$	$\Gamma$ Dependent Plate Scales	$\delta p_1(4)$	mask1(4)
$p_1(5)$	$b_{10}$		$\delta p_1(5)$	mask1(5)
$p_1(6)$	$c_{10}$		$\delta p_1(6)$	mask1(6)
$p_1(7)$	$d_{10}$		$\delta p_1(7)$	mask1(7)
$p_1(8)$	$a_{20}$	$\Gamma^2$ Dependent Plate Scales	$\delta p_1(8)$	mask1(8)
$p_1(9)$	$b_{20}$		$\delta p_1(9)$	mask1(9)
$p_1(10)$	$c_{20}$		$\delta p_1(10)$	mask1(10)
$p_1(11)$	$d_{20}$		$\delta p_1(11)$	mask1(11)
$p_1(12)$	$a_{01}$	Linear Plate Scales	$\delta p_1(12)$	mask1(12)
$p_1(13)$	$b_{01}$		$\delta p_1(13)$	mask1(13)
$p_1(14)$	$c_{01}$		$\delta p_1(14)$	mask1(14)
$p_1(15)$	$d_{01}$		$\delta p_1(15)$	mask1(15)
$p_1(16)$	$e_{01}$		$\delta p_1(16)$	mask1(16)
$p_1(17)$	$f_{01}$		$\delta p_1(17)$	mask1(17)

Table 4.10:  $p_1$  State Variables, Perturbations and Mask Vector

$p_{2f}$	Math	Description	$\delta p_2$	Mask
$p_{2f}(1)$	$a_{m1}$	mirror rotation axis unit vector in IPF (x)		
$p_{2f}(2)$	$a_{m2}$	mirror rotation axis unit vector in IPF (y)	$\delta\alpha$	mask2(1)
$p_{2f}(3)$	$a_{m3}$	mirror rotation axis unit vector in IPF (z)		
$p_{2f}(4)$	$\beta$	scan mirror rotation angle scale factor	$\delta\beta$	mask2(2)
$p_{2f}(5)$	$q_{T1}$	T (TPF to IPF) frame quaternion, $q_T(1)$	$\delta\theta_1$	mask2(3)
$p_{2f}(6)$	$q_{T2}$	T (TPF to IPF) frame quaternion, $q_T(2)$	$\delta\theta_1$	mask2(4)
$p_{2f}(7)$	$q_{T3}$	T (TPF to IPF) frame quaternion, $q_T(3)$	$\delta\theta_1$	mask2(5)
$p_{2f}(8)$	$q_{T4}$	T (TPF to IPF) frame quaternion, $q_T(4)$		
$p_{2f}(9)$	$q_{R1}$	R Alignment quaternion, $q_R(1)$	$\delta a_{rx}$	mask2(6)
$p_{2f}(10)$	$q_{R2}$	R Alignment quaternion, $q_R(2)$	$\delta a_{ry}$	mask2(7)
$p_{2f}(11)$	$q_{R3}$	R Alignment quaternion, $q_R(3)$	$\delta a_{rz}$	mask2(8)
$p_{2f}(12)$	$q_{R4}$	R Alignment quaternion, $q_R(4)$		
$p_{2f}(13)$	$b_{rx}$	Linear time varying contribution on alignment x-axis	$\delta b_{rx}$	mask2(9)
$p_{2f}(14)$	$b_{ry}$	Linear time varying contribution on alignment y-axis	$\delta b_{ry}$	mask2(10)
$p_{2f}(15)$	$b_{rz}$	Linear time varying contribution on alignment z-axis	$\delta b_{rz}$	mask2(11)
$p_{2f}(16)$	$c_{rx}$	Quadratic time varying contribution on alignment x-axis	$\delta c_{rx}$	mask2(12)
$p_{2f}(17)$	$c_{ry}$	Quadratic time varying contribution on alignment y-axis	$\delta c_{ry}$	mask2(13)
$p_{2f}(18)$	$c_{rz}$	Quadratic time varying contribution on alignment z-axis	$\delta c_{rz}$	mask2(14)
$p_{2f}(19)$	$b_{gx}$	Delta Gyro Bias from Nominal, x-axis	$\delta b_{gx}$	mask2(15)
$p_{2f}(20)$	$b_{gy}$	Delta Gyro Bias from Nominal, y-axis	$\delta b_{gy}$	mask2(16)
$p_{2f}(21)$	$b_{gz}$	Delta Gyro Bias from Nominal, z-axis	$\delta b_{gz}$	mask2(17)
$p_{2f}(22)$	$c_{gx}$	Gyro Bias Drift Rate, x-axis	$\delta c_{gx}$	mask2(18)
$p_{2f}(23)$	$c_{gy}$	Gyro Bias Drift Rate, x-axis	$\delta c_{gy}$	mask2(19)
$p_{2f}(24)$	$c_{gz}$	Gyro Bias Drift Rate, x-axis	$\delta c_{gz}$	mask2(20)

Table 4.11:  $p_2$  State Variables, Perturbations and Mask Vector

Parameter Description		Const. Plate Scale			$\Gamma$ Dependent			$\Gamma^2$ Dependent			Linear Plate Scale							
$p_1$ Parameters		$a_{00}$	$b_{00}$	$c_{00}$	$a_{10}$	$b_{10}$	$c_{10}$	$d_{10}$	$a_{20}$	$b_{20}$	$c_{20}$	$d_{20}$	$a_{01}$	$b_{01}$	$c_{01}$	$d_{01}$	$e_{01}$	$f_{01}$
Instrument Name	NF	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
IRS_Red_PkUp_Cntr	18	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1
IRS_Red_PkUp_SwtSpot	19	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1
IRS_Blue_PkUp_Cntr	22	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1
IRS_Blue_PkUp_SwtSpot	23	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1
IRS_ShortLo_1stOrd_Cntr	28	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IRS_ShortLo_2ndOrd_Cntr	34	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IRS_LongLo_1stOrd_Cntr	40	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IRS_LongLo_2ndOrd_Cntr	46	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IRS_ShortHi_Cntr	52	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IRS_LongHi_Cntr	58	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IRAC_Cntr_3.6umArray	68	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IRAC_Cntr_5.8umArray	69	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IRAC_Cntr_4.5umArray	75	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IRAC_Cntr_8.0umArray	76	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MIPS_160um_CntrLgFOV	87	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
MIPS_24um_Cntr	95	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
MIPS_70um_Cntr	107	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
MIPS_70um_fine_Cntr	118	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
MIPS_SED_Cntr	121	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.12: Default Mask Vector Assignment for  $p_1$  State Variables (mask1)

Parameter Description	Mirror			IPF (T)			Alignment R						Gyro Drift Bias								
	$\alpha$	$\beta$	NF	$\theta_1$	$\theta_2$	$\theta_3$	$a_{rx}$	$a_{ry}$	$a_{rz}$	$b_{rx}$	$b_{ry}$	$b_{rz}$	$c_{rx}$	$c_{ry}$	$c_{rz}$	$b_{gx}$	$b_{gy}$	$b_{gz}$	$c_{gx}$	$c_{gy}$	$c_{gz}$
IRS_Red_PkUp_Cntr	1	2	NF	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
IRS_Red_PkUp_SwtSpot	0	0	18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
IRS_Blue_PkUp_Cntr	0	0	19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
IRS_Blue_PkUp_SwtSpot	0	0	22	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
IRS_ShortLo_1stOrd_Cntr	0	0	23	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
IRS_ShortLo_2ndOrd_Cntr	0	0	28	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
IRS_LongLo_1stOrd_Cntr	0	0	34	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
IRS_LongLo_2ndOrd_Cntr	0	0	40	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
IRS_ShortHi_Cntr	0	0	46	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
IRS_LongHi_Cntr	0	0	52	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
IRAC_Cntr_3.6umArray	0	0	58	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
IRAC_Cntr_5.8umArray	0	0	68	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
IRAC_Cntr_4.5umArray	0	0	69	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
IRAC_Cntr_8.0umArray	0	0	75	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
MIPS_160um_CntrLgFOV	0	0	76	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
MIPS_24um_Cntr	1	1	87	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
MIPS_70um_Cntr	1	1	95	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
MIPS_70um_fine_Cntr	1	1	107	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
MIPS_SED_Cntr	0	0	118	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	0	0	121	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 4.13: Default Mask Vector Assignment for  $p_2$  State Variables (mask2)

## 4.2 Optical Distortion Parameters

Optical distortion parameters capture imperfections and variations in the telescope and instrument which cause a star image to deviate from its idealized geometric projection. The optical distortions in the calibration equation (3.30) are parameterized in terms of the matrix  $M \in \mathcal{R}^{2 \times 2}$  of the form,

$$M(p_1, \Gamma, y) = M_{00} + \Gamma M_{10} + \Gamma^2 M_{20} + M_{01}(y) \quad (4.2)$$

where,

$$M_{00} = \begin{bmatrix} a_{00} & c_{00} \\ c_{00} & b_{00} \end{bmatrix}; \quad M_{10} = \begin{bmatrix} a_{10} & c_{10} \\ d_{10} & b_{10} \end{bmatrix}; \quad M_{20} = \begin{bmatrix} a_{20} & c_{20} \\ d_{20} & b_{20} \end{bmatrix}; \quad (4.3)$$

$$M_{01}(y) = \begin{bmatrix} a_{01}y_w + c_{01}y_v & b_{01}y_v \\ d_{01}y_w & f_{01}y_w + e_{01}y_v \end{bmatrix}. \quad (4.4)$$

The parameter  $c_{00}$  is repeated symmetrically in  $M_{00}$  to disallow a redundant rotation with  $\theta_1$  of  $T$  (cf., [23]).

## 4.3 Scan Mirror Rotation Parameters

For science arrays having a scan mirror (i.e., MIPS arrays), the scan mirror rotation transformation can be defined by a direction cosine matrix  $C$  which maps the nominal IPF frame (denoted as  $IPF_0$ ) to the IPF frame with a scan mirror offset (denoted as  $IPF_\Gamma$ ). Mathematically,  $C$  is parametrized as an Euler axis rotation of the form,

$$C(p_{2f}, \Gamma) = \cos(\beta\Gamma) \cdot I + (1 - \cos(\beta\Gamma)) a_m a_m^T - \sin(\beta\Gamma) a_m^\times \quad (4.5)$$

Here  $a_m = [a_{m1} \ a_{m2} \ a_{m3}]^T$  is the scan mirror spin axis,  $\Gamma$  is the measured scan mirror angle (see Appendix, Section B.3), and  $\beta$  is the scale factor associated with measured mirror angle. The cross product matrix operation  $a_m^\times$  is defined as,

$$a_m^\times = \begin{bmatrix} 0 & -a_{m3} & a_{m2} \\ a_{m3} & 0 & -a_{m1} \\ -a_{m2} & a_{m1} & 0 \end{bmatrix} \quad (4.6)$$

The vector  $a_m$  is constrained to have unit norm, i.e.,

$$a_{m1}^2 + a_{m2}^2 + a_{m3}^2 = 1 \quad (4.7)$$

and the mirror transformation becomes the identity when the mirror is located in its nominal reference position ( $\Gamma = 0$ ), i.e.,

$$C(p_{2f}, 0) = I. \quad (4.8)$$

For non-MIPS instruments (without a scan mirror), the condition  $C = I$  is enforced.

## 4.4 Telescope Pointing Frame Parameters

The direction cosine matrix  $T$  transforms from TPF to  $IPF_0$  (i.e., the IPF frame associated with  $\Gamma = 0$ ), and can be parameterized with a quaternion  $q_T$  as shown below.

$$T(q_T) = \begin{bmatrix} q_{T1}^2 - q_{T2}^2 - q_{T3}^2 + q_{T4}^2 & 2(q_{T1}q_{T2} + q_{T3}q_{T4}) & 2(q_{T1}q_{T3} - q_{T2}q_{T4}) \\ 2(q_{T1}q_{T2} - q_{T3}q_{T4}) & q_{T2}^2 - q_{T3}^2 + q_{T4}^2 - q_{T1}^2 & 2(q_{T2}q_{T3} + q_{T1}q_{T4}) \\ 2(q_{T1}q_{T3} + q_{T2}q_{T4}) & 2(q_{T2}q_{T3} - q_{T1}q_{T4}) & q_{T3}^2 + q_{T4}^2 - q_{T1}^2 - q_{T2}^2 \end{bmatrix} \quad (4.9)$$

If the  $T$  is described in terms of 3-2-1 Euler sequence, it can be written as:

$$T(p_{2f}) \triangleq T_1(\theta_1)T_2(\theta_2)T_3(\theta_3) \quad (4.10)$$

where

$$T_1(\theta_1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_1 & \sin \theta_1 \\ 0 & -\sin \theta_1 & \cos \theta_1 \end{bmatrix} \quad (4.11)$$

$$T_2(\theta_2) = \begin{bmatrix} \cos \theta_2 & 0 & -\sin \theta_2 \\ 0 & 1 & 0 \\ \sin \theta_2 & 0 & \cos \theta_2 \end{bmatrix} \quad (4.12)$$

$$T_3(\theta_3) = \begin{bmatrix} \cos \theta_3 & \sin \theta_3 & 0 \\ -\sin \theta_3 & \cos \theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (4.13)$$

## 4.5 Thermomechanical Drift Parameters

The directional cosine matrix  $R$  represents the mapping from the STA-defined Body frame to the TPF frame. Since the STA is mounted remotely from the focal plane, a relatively large and time-varying uncertainty can exist between these two frames. The IPF filter estimates this relative misalignment by parameterizing each of the three rotation angles as a quadratic function of time. The resulting model is,

$$R \triangleq \left( I_{3 \times 3} - \left( b_r t + c_r \frac{t^2}{2} \right)^\times \right) R_0(q_R) \quad (4.14)$$

where,

$$b_r = \begin{bmatrix} b_{rx} \\ b_{ry} \\ b_{rz} \end{bmatrix}; \quad c_r = \begin{bmatrix} c_{rx} \\ c_{ry} \\ c_{rz} \end{bmatrix} \quad (4.15)$$

The beginning time  $t = 0$  corresponds to the time tag of the first centroid of the very first maneuver (typically a PCRS centroid). By definition, the quantity  $R_0$  is the static alignment at time  $t = 0$ . The parameters  $b_r$  and  $c_r$  capture the linear and quadratic variations, respectively, in the thermomechanical drift between the Body and TPF frames subsequent to  $t = 0$ .

For notational simplicity, the quaternion equivalent of the initial alignment  $R_0$  is denoted as  $q_R$  (rather than  $q_{R0}$ ).

Since  $R_0$  itself is unknown, it is described in terms of small angular perturbation  $(\delta a_{rx}, \delta a_{ry}, \delta a_{rz})$  about a prior estimate of the alignment which is updated as follows,

$$R_0 \leftarrow \left( I - \begin{bmatrix} \delta a_{rx} \\ \delta a_{ry} \\ \delta a_{rz} \end{bmatrix}^\times \right) R_0 \quad (4.16)$$

It is seen that the parameters  $\delta a_{rx}, \delta a_{ry}, \delta a_{rz}$  only exist incrementally, since they are absorbed onto  $R_0$  in each iteration.

## 4.6 Attitude and Gyro Parameters

The attitude directional cosine matrix  $A$  maps from ICRS to the STA-defined Body frame. This transformation can be divided into two components as:

$$A \triangleq G A_0 \quad (4.17)$$

where  $A_0$  is the initial attitude at the beginning of a specified maneuver, and  $G$  is the gyro-propagated offset attitude which parameterizes the spacecraft body motion at all times from  $t = 0$  to the time of the beginning of the next maneuver.

The true gyro offset  $G$  is defined by integrating the true rate  $\omega$  according to the differential equation,

$$\left( \dot{G} = -\omega^\times G \right) \Big|_{t_j}^{T_k(j)} \quad (4.18)$$

starting with the initial condition  $G(t_j) = I$ .

Since the true rate  $\omega \in \mathcal{R}^3$  is not known exactly, an estimate must be generated. For computational convenience, this is done in two stages. First, the gyro pre-processor produces a nominal rate vector estimate  $\omega_m^\circ \in \mathcal{R}^3$  according a certain construction (to be discussed later). Second, a correction is applied to the nominal rate vector  $\omega_m^\circ$  to get the true rate as,

$$\omega = \omega_m^\circ + b_g + c_g t \quad (4.19)$$

Here it is seen that the parameters  $b_g, c_g \in \mathcal{R}^3$  denote a bias and a bias drift correction to the nominal rate estimate.

# 5 FORMULATIONS

## 5.1 IPF Kalman Filter Formulation

The Spitzer IPF Kalman Filter algorithm is architected as a square-root iterated linearized Kalman filter. The filter operates in block sequential form as shown in Figure 5.1. First, the nominal state estimate to be used for linearization purposes is prescribed at the beginning of the data set (corresponding to  $t = 0$ ). For notation purposes, the start time of the  $j$ th sandwich maneuver is denoted as  $t_j$ , and the individual centroid times are denoted as  $T_k$ . Centroid data from each individual maneuver is then “stacked” into a single tall measurement vector which is used to update the Kalman filter. Accordingly, for a calibration data set having  $N$  sandwich maneuvers, there will be  $N$  vector measurement updates. After processing the entire data set, the estimated state correction is applied and the filter is re-linearized about the updated state estimate. This procedure of processing the entire data set and then re-linearizing is repeated until convergence is obtained.

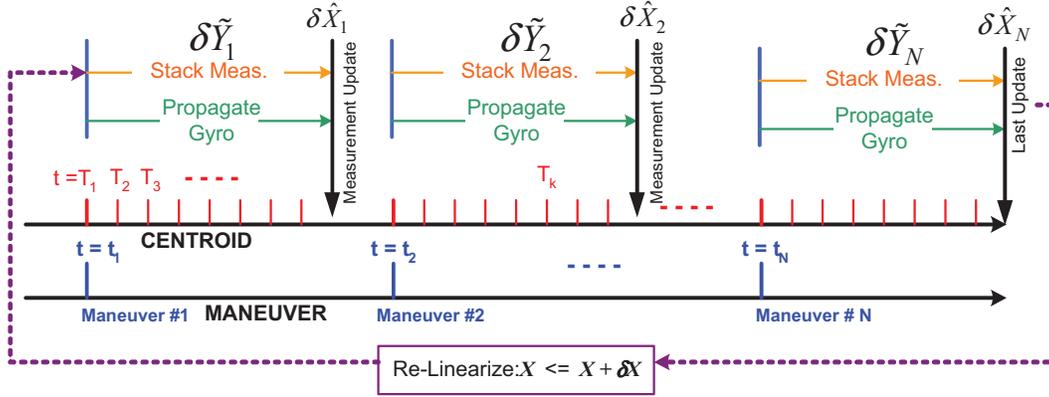


Figure 5.1: Square-Root, Iterated and Linearized Kalman Filtering Process

### 5.1.1 Time Update

Corresponding to the full state vector  $x_f$  defined in (4.1), a state vector perturbation is given as,

$$\delta x = \begin{bmatrix} \delta p_1 \\ \delta p_2 \end{bmatrix} \in \mathcal{R}^{37} \quad (5.1)$$

where the quantities  $p_1, \delta p_1, p_{2f}, \delta p_2$  have been defined in Table 4.10 and Table 4.10. It is noted that  $\delta p_2 \in \mathcal{R}^{20}$  is of smaller dimension than  $p_{2f} \in \mathcal{R}^{24}$  due to constraints on the parameters.

Since the IPF Kalman filter is parameterized by constant coefficients, the state perturbation propagation equation can be written as,

$$\delta\dot{x} = \begin{bmatrix} \delta\dot{p}_1 \\ \delta\dot{p}_2 \end{bmatrix} = 0. \quad (5.2)$$

Accordingly, the discrete form of the state perturbation propagation equation for the mean and the square-root covariance can be written as,

$$\delta\hat{x}_{j+1|j} = \delta\hat{x}_{j|j} \quad (5.3)$$

$$P_{j+1|j}^{\frac{1}{2}} = P_{j|j}^{\frac{1}{2}}. \quad (5.4)$$

where  $j + 1|j$  signifies the predicted value at the start of the  $j + 1$ 'th maneuver, given measurements from the past  $j$  maneuvers.

### 5.1.2 Measurement Update

#### Measurement Perturbation

The Calibration Equation (3.30) will serve as a measurement equation for the Kalman filter. This is done by first rearranging it into the form,

$$y = h(p_1, p_{2f}, \psi, y) + v \quad (5.5)$$

where,

$$h(p_1, p_{2f}, \psi, y) = -M(p_1, y)y + h_z(p_{2f}, \psi) \quad (5.6)$$

Note that the measurement  $y$  is used on both sides of (5.5), which requires a slight deviation from Kalman filter conventions (motivated by the implicit form of  $y$  in (3.30)). Note also that the dependence of  $M$  on  $\Gamma$  has been dropped purely for notational convenience. Interestingly, in the case of slits, the complete  $y$  is not available in any single measurement to evaluate the right-hand side of (5.6). In this case, the linearization is handled differently, as described in Section 6.7.1.

Equation (5.5) can be linearized to obtain the desired Kalman filter update equation. To this effect, a prediction  $\hat{h}$  of  $h$  is constructed using the nominal state estimates  $\hat{p}_1$  and  $\hat{p}_{2f}$  as follows,

$$\hat{h} = h(\hat{p}_1, \hat{p}_{2f}, 0, y) \quad (5.7)$$

Subtracting (5.7) from (5.5) gives the desired measurement perturbation equation as,

$$\delta y \triangleq y - h(\hat{p}_1, \hat{p}_{2f}, 0, y) \quad (5.8)$$

$$= \mathcal{K}_1 \delta p_1 + \mathcal{K}_2 \delta p_2 + H_\psi \psi + \nu \quad (5.9)$$

$$= \begin{bmatrix} \mathcal{K}_1 & \mathcal{K}_2 \end{bmatrix} \begin{bmatrix} \delta p_1 \\ \delta p_2 \end{bmatrix} + H_\psi \psi + \nu \quad (5.10)$$

$$= H \delta x + n \quad (5.11)$$

where use has been made above of the following definitions,

$$H \triangleq [ \mathcal{K}_1 \quad \mathcal{K}_2 ] \quad (5.12)$$

$$\mathcal{K}_1 \triangleq \left. \frac{\partial h}{\partial p_1} \right|_{\hat{p}_1, \hat{p}_{2f}} = \left. \frac{\partial}{\partial p_1} (-M(p_1, y) y) \right|_{\hat{p}_1, \hat{p}_{2f}} \quad (5.13)$$

$$\mathcal{K}_2 \triangleq \left. \frac{\partial h_z}{\partial(\delta p_2)} \right|_{\hat{p}_1, \hat{p}_{2f}} \quad (5.14)$$

$$H_\psi \triangleq \left. \frac{\partial h_z}{\partial \psi} \right|_{\hat{p}_1, \hat{p}_{2f}} \quad (5.15)$$

$$\delta x \triangleq \begin{bmatrix} \delta p_1 \\ \delta p_2 \end{bmatrix} \quad (5.16)$$

$$n \triangleq H_\psi \psi + \nu \quad (5.17)$$

Equation (5.11) is the desired measurement perturbation relation. It is seen from (5.17) that the measurement noise  $n$  is actually composed of two noises - the centroiding error  $\nu$ , and the initial attitude error  $\psi$ . One must be careful to recognize that the initial attitude error  $\psi$  is not independent from centroid to centroid, but rather is a single error which is common to all centroids taken during a single sandwich maneuver.

The perturbation parameter vector  $\delta p_1 \in \mathcal{R}^{17}$  is defined in Table 4.10, and  $\delta p_2 \in \mathcal{R}^{20}$  is defined in Table 4.11.

As mentioned earlier, the perturbation  $\delta p_2 \in \mathcal{R}^{20}$  has a smaller dimension than the vector it perturbs  $p_{2f} \in \mathcal{R}^{24}$ . This is due to the many constraints that exist in the problem. For example, quaternions in  $p_{2f}$  have 4 elements, but only 3 of them are independent and have been carried in the perturbation  $\delta p_2$ .

In general, the derivative of a function  $f(\xi_f)$  with respect to a constrained parameter vector  $\xi_f$ , will be defined in terms of only the independent parameters  $\delta \xi$  (where  $\delta \xi$  can be of lower dimension than  $\xi_f$ ) by the matrix  $\mathcal{K}$  that satisfies,

$$f(\xi_f) = f(\hat{\xi}_f) + \mathcal{K} \delta \xi + \mathcal{O}(\|\delta \xi\|^2) \quad (5.18)$$

Because in this general case the variable  $\xi$  may only exist in its perturbation form  $\delta \xi$ , the following notation will be used throughout the remainder of the report,

$$\mathcal{K} \triangleq \left. \frac{\partial f}{\partial(\delta \xi)} \right|_{\hat{\xi}_f} \quad (5.19)$$

For example, this notation has been used in (5.14) for describing  $\mathcal{K}_2$  which involves a derivative with respect to  $\delta p_2$ .

## Stacked Measurements

Equation (5.11) provides the measurement perturbation  $\delta y$  associated with a single centroid measurement. For filtering purposes, all of the centroids associated with the  $j$ 'th maneuver are stacked into a single measurement vector as follows,

$$\delta \tilde{y}_j = \tilde{H}_j \delta x + \tilde{n}_j \quad (5.20)$$

where,

$$\delta \tilde{y}_j \triangleq \begin{bmatrix} \delta y_1 \\ \vdots \\ \delta y_{m_j} \end{bmatrix}; \quad \tilde{n}_j \triangleq \begin{bmatrix} n_1 \\ \vdots \\ n_{m_j} \end{bmatrix}; \quad \tilde{H}_j \triangleq \begin{bmatrix} H_1 \\ \vdots \\ H_{m_j} \end{bmatrix} \quad (5.21)$$

and where  $m_j$  is the number of centroids in the  $j$ 'th maneuver.

Because of the special structure of the noise  $n$  in (5.17), the stacked noise term  $\tilde{n}_j$  in (5.20) can be broken down into two separate terms as,

$$\tilde{n}_j = \tilde{\nu}_j + \tilde{H}_{\psi,j} \psi_j \quad (5.22)$$

where,

$$\tilde{\nu}_j \triangleq \begin{bmatrix} \nu_1 \\ \vdots \\ \nu_{m_j} \end{bmatrix}; \quad \tilde{H}_{\psi,j} \triangleq \begin{bmatrix} H_{\psi,1} \\ \vdots \\ H_{\psi,m_j} \end{bmatrix} \quad (5.23)$$

Here, we have used the fact that the initial attitude error  $\psi_j \in R^3$  associated with the  $j$ 'th maneuver contributes to all of the measurements  $\tilde{\nu}_j \in R^{2m_j}$  taken during that maneuver.

Assuming independence of centroiding and attitude errors, the covariance of  $\tilde{n}_j$  in (5.22) can be computed as,

$$\tilde{R}_j \triangleq Cov[\tilde{n}_j] = \tilde{V}_j + \tilde{H}_{\psi,j} P_{\psi,j} \tilde{H}_{\psi,j}^T \quad (5.24)$$

where,

$$\tilde{V}_j \triangleq Cov[\tilde{\nu}_j] = \begin{bmatrix} V_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & V_{m_j} \end{bmatrix} \quad (5.25)$$

$$V_i \triangleq Cov[\nu_i]; \quad P_{\psi,j} = Cov[\psi_j] \quad (5.26)$$

It will be convenient to work with square-root covariances. Let the following factorized matrices be defined,

$$\tilde{R}_j \triangleq \tilde{R}_j^{\frac{1}{2}} \tilde{R}_j^{\frac{T}{2}} \quad (5.27)$$

$$P_{\psi,j} \triangleq P_{\psi,j}^{\frac{1}{2}} P_{\psi,j}^{\frac{T}{2}} \quad (5.28)$$

$$\tilde{V}_j \triangleq \tilde{V}_j^{\frac{1}{2}} \tilde{V}_j^{\frac{T}{2}} \quad (5.29)$$

Using these factorized matrix definitions, equation (5.24) can be equivalently written as,

$$\tilde{R}_j^{\frac{1}{2}} \tilde{R}_j^{\frac{T}{2}} = \tilde{H}_{\psi,j} P_{\psi,j}^{\frac{1}{2}} P_{\psi,j}^{\frac{T}{2}} \tilde{H}_{\psi,j}^T + \tilde{V}_j^{\frac{1}{2}} \tilde{V}_j^{\frac{T}{2}} \quad (5.30)$$

Recognizing that this has the general form  $CC^T = AA^T + BB^T$ , the square-root factor  $\tilde{R}_j^{\frac{1}{2}}$  can be determined by using the results of Lemma D.17 and the  $QR$  factorization approach discussed in Section D.18.

### Measurement Update

Using the stacked measurement equation (5.20), the Kalman Filter gain and the square-root covariance update can be obtained. Specifically, given  $\tilde{H}_j$ ,  $P_{j|j-1}^{\frac{1}{2}}$  and  $\tilde{R}_j^{\frac{1}{2}}$  consider the unitary triangularization of the following matrix,

$$\begin{bmatrix} \tilde{R}_j^{\frac{1}{2}} & \tilde{H}_j P_{j|j-1}^{\frac{1}{2}} \\ 0 & P_{j|j-1}^{\frac{1}{2}} \end{bmatrix} = \begin{bmatrix} X & 0 \\ Y & Z \end{bmatrix} \Theta \quad (5.31)$$

where  $\Theta^T$  is an orthogonal matrix (i.e.,  $\Theta^T \Theta = I$ ). The factorization in (5.31) can be performed using a QR factorization. Extracting  $X$ ,  $Y$  and  $Z$ , one can compute the Kalman Filter Gain  $K$  and square-root covariance update equation as (cf., Corollary D.1 and [12]),

$$K_j = Y X^{-1} \quad (5.32)$$

$$P_{j|j}^{\frac{1}{2}} = Z \quad (5.33)$$

Given the Kalman gain in (5.32) the state perturbation update equation is,

$$\delta \hat{x}_{j|j} = \delta \hat{x}_{j|j-1} + K_j \left( \delta \tilde{y}_j - \tilde{H}_j \delta \hat{x}_{j|j-1} \right) \quad (5.34)$$

This completes the discussion of the time and measurement updates of the Kalman filter. The next two subsections will briefly outline the required sensitivity calculations. More details are provided in the Appendix.

## 5.2 Sensitivity Equations for $p_1$ Parameters

The sensitivity  $\mathcal{K}_1$  is defined from (5.13) as,

$$\mathcal{K}_1 = \left. \frac{\partial}{\partial p_1} (-M(p_1, y) y) \right|_{\hat{p}_1, \hat{p}_{2f}} \quad (5.35)$$

Using the result of Lemma D.21 and the Kronecker product identity of Lemma D.22 gives,

$$-M(p_1, y) y = -(M_{00} + \Gamma M_{10} + \Gamma^2 M_{20} + M_{01}(y)) y \quad (5.36)$$

$$= -(y^T \otimes I) \text{Vec}(M) \quad (5.37)$$

$$= -(y^T \otimes I) [ S_{00} \quad \Gamma S_{10} \quad \Gamma^2 S_{20} \quad f_u(y) ] p_1 \quad (5.38)$$

where,

$$S_{00} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad S_{10} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad S_{20} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (5.39)$$

$$f_u(y) = \begin{bmatrix} y_w & 0 & y_v & 0 & 0 & 0 \\ 0 & 0 & 0 & y_w & 0 & 0 \\ 0 & y_v & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & y_v & y_w \end{bmatrix} \quad (5.40)$$

$$y^T \otimes I = \begin{bmatrix} y_w & 0 & y_v & 0 \\ 0 & y_w & 0 & y_v \end{bmatrix} \quad (5.41)$$

Substituting (5.38) into (5.35) gives the desired expression,

$$\mathcal{K}_1 = -(y^T \otimes I) [ S_{00} \quad \Gamma S_{10} \quad \Gamma^2 S_{20} \quad f_u(y) ] \quad (5.42)$$

### 5.3 Sensitivity Equations for $p_2$ and $\psi$ Parameters

This section will discuss how to compute sensitivities taken with respect to  $p_2$  and  $\psi$  as used to define  $\mathcal{K}_2$  and  $H_\psi$  in (5.14) and (5.15), respectively. The main idea is to make use of equations (3.19)-(3.21) derived earlier.

Assume that current estimates  $\hat{C}, \hat{T}, \hat{R}, \hat{G}, \hat{A}_0$  are sufficiently close to  $C, T, R, G, A_0$  so that one can define the perturbations  $c_k, \theta_k, \phi_k, g_k, \psi_k$  according to the following relationships,

$$C = (I - c_k^\times) \hat{C} \quad (5.43)$$

$$T = (I - \theta_k^\times) \hat{T} \quad (5.44)$$

$$R = (I - \phi_k^\times) \hat{R} \quad (5.45)$$

$$G = (I - g_k^\times) \hat{G} \quad (5.46)$$

$$A_0 = (I - \psi_k^\times) \hat{A}_0 \quad (5.47)$$

Substituting (5.43)-(5.47) into (3.19) and rearranging gives,

$$s \triangleq CTRGA_0\ell \quad (5.48)$$

$$\simeq (I - c_k^\times) \hat{C} (I - \theta_k^\times) \hat{T} (I - \phi_k^\times) \hat{R} (I - g_k^\times) \hat{G} (I - \psi_k^\times) \hat{A}_0\ell \quad (5.49)$$

$$\simeq \left[ I - \left( c_k + \hat{C}\theta_k + \hat{C}\hat{T}\phi_k + \hat{C}\hat{T}\hat{R}g_k + \hat{C}\hat{T}\hat{R}\hat{G}\psi_k \right)^\times \right] \hat{C}\hat{T}\hat{R}\hat{G}\hat{A}_0\ell \quad (5.50)$$

$$= (I - \eta^\times) \hat{N}\ell. \quad (5.51)$$

where  $\eta$  denotes the total pointing perturbation given by the expression,

$$\eta = c_k + \hat{C}\theta_k + \hat{C}\hat{T}\phi_k + \hat{C}\hat{T}\hat{R}g_k + \hat{C}\hat{T}\hat{R}\hat{G}\psi_k \quad (5.52)$$

$$\hat{N} = \hat{C}\hat{T}\hat{R}\hat{G}\hat{A}_0 \quad (5.53)$$

It is seen from (5.52) that  $\eta$  can be written as a linear function of the individual perturbations as follows,

$$\eta = \begin{bmatrix} I & \hat{C} & \hat{C}\hat{T} & \hat{C}\hat{T}\hat{R} \end{bmatrix} \begin{bmatrix} c_k \\ \theta_k \\ \phi_k \\ g_k \end{bmatrix} + \hat{C}\hat{T}\hat{R}\hat{G}\psi_k \quad (5.54)$$

$$= H_\eta\lambda_k + L_\psi\psi_k. \quad (5.55)$$

where,

$$\lambda \triangleq [c_k^T \ \theta_k^T \ \phi_k^T \ g_k^T]^T \in \mathcal{R}^{12} \quad (5.56)$$

$$H_\eta \triangleq [I \ \hat{C} \ \hat{C}\hat{T} \ \hat{C}\hat{T}\hat{R}] \quad (5.57)$$

$$L_\psi \triangleq \hat{C}\hat{T}\hat{R}\hat{G} \quad (5.58)$$

Given the above construction, the desired sensitivities can be written in the form,

$$\mathcal{K}_2 = \frac{\partial h_z}{\partial(\delta p_2)} = \begin{bmatrix} \frac{\partial z}{\partial s} \end{bmatrix} \begin{bmatrix} \frac{\partial s}{\partial \eta} \end{bmatrix} \begin{bmatrix} \frac{\partial \eta}{\partial \lambda} \end{bmatrix} \begin{bmatrix} \frac{\partial \lambda}{\partial(\delta p_2)} \end{bmatrix} = H_z H_s H_\eta H_\lambda \quad (5.59)$$

$$H_\psi = \frac{\partial h_z}{\partial \psi} = \begin{bmatrix} \frac{\partial z}{\partial s} \end{bmatrix} \begin{bmatrix} \frac{\partial s}{\partial \eta} \end{bmatrix} \begin{bmatrix} \frac{\partial \eta}{\partial \psi} \end{bmatrix} = H_z H_s L_\psi \quad (5.60)$$

where evaluation on  $\hat{p}_1, \hat{p}_{2f}$  and  $\psi = 0$  is implied. The various terms in these expressions for  $\mathcal{K}_2$  and  $H_\psi$  will be derived in the remainder of the section.

### 5.3.1 $\frac{\partial z}{\partial s}$ Derivation

Starting with equation (3.21), the partial derivative of  $z$  with respect to  $s$  can be calculated as,

$$H_z = \frac{\partial z}{\partial s} = \begin{bmatrix} -\frac{\hat{s}_z}{\hat{s}_x^2} & 0 & \frac{1}{\hat{s}_x} \\ -\frac{\hat{s}_y}{\hat{s}_x^2} & \frac{1}{\hat{s}_x} & 0 \end{bmatrix} \quad (5.61)$$

where,

$$\hat{s} = \hat{N}\ell = \hat{C}\hat{T}\hat{R}\hat{G}\hat{A}_0\ell \quad (5.62)$$

### 5.3.2 $\frac{\partial s}{\partial \eta}$ Derivation

Equation (5.51) can be rearranged to give,

$$s = (I - \eta^\times) \hat{N}\ell = \hat{N}\ell - \eta^\times \hat{N}\ell \quad (5.63)$$

$$= \hat{N}\ell + (\hat{N}\ell)^\times \eta \quad (5.64)$$

where the vector cross product property has been used ( $-a \times b = b \times a$ ). Taking the partial derivative of (5.64) with respect to  $\eta$  gives,

$$H_s = \frac{\partial s}{\partial \eta} = (\hat{N}\ell)^\times = (\hat{C}\hat{T}\hat{R}\hat{G}\hat{A}_0\ell)^\times \quad (5.65)$$

### 5.3.3 $\frac{\partial \eta}{\partial \lambda}$ Derivation

Starting with equation (5.55), the partial derivative of  $\eta$  with respect to  $\lambda$  can be calculated as,

$$H_\eta = \frac{\partial \eta}{\partial \lambda} = [ I \quad \hat{C} \quad \hat{C}\hat{T} \quad \hat{C}\hat{T}\hat{R} ]. \quad (5.66)$$

Equation (5.66) is the most general expression. However, since sensitivities with respect to  $c_k$  are only needed for the MIPS instruments (which have scan mirrors), and since sensitivities with respect to  $\theta_k$  are not needed for the PCRS sensors (which define the TPF frame), the actual value of  $H_\eta$  will be calculated differently depending on which array the specified centroid was taken on, i.e.,

$$H_\eta = \begin{cases} [I, \hat{C}, \hat{C}\hat{T}, \hat{C}\hat{T}\hat{R}] & \text{for MIPS} \\ [0, I, \hat{C}\hat{T}, \hat{C}\hat{T}\hat{R}] & \text{for non-MIPS} \\ [0, 0, \hat{C}\hat{T}, \hat{C}\hat{T}\hat{R}] & \text{for PCRS} \end{cases} \quad (5.67)$$

### 5.3.4 $\frac{\partial \lambda}{\partial (\delta p_2)}$ Derivation

Given  $\lambda = [ c_k \quad \theta_k \quad \phi_k \quad g_k ]^T$ , it is convenient to decompose the sensitivity of  $\lambda$  with respect to  $\delta p_2$  into the following matrix,

$$H_\lambda = \frac{\partial \lambda}{\partial (\delta p_2)} = \begin{bmatrix} H_c & 0 & 0 & 0 \\ 0 & H_\theta & 0 & 0 \\ 0 & 0 & H_\phi & 0 \\ 0 & 0 & 0 & H_g \end{bmatrix} \quad (5.68)$$

where,

$$H_c = \left[ \frac{\partial c_k}{\partial (\delta \alpha)} \vdots \frac{\partial c_k}{\partial (\delta \beta)} \right] \quad (5.69)$$

$$H_\theta = \frac{\partial \theta_k}{\partial (\delta \theta)} \quad (5.70)$$

$$H_\phi = \left[ \frac{\partial \phi_k}{\partial (\delta a_r)} \vdots \frac{\partial \phi_k}{\partial (\delta b_r)} \vdots \frac{\partial \phi_k}{\partial (\delta c_r)} \right] \quad (5.71)$$

$$H_g = \left[ \frac{\partial g_k}{\partial (\delta b_g)} \vdots \frac{\partial g_k}{\partial (\delta c_g)} \right] \quad (5.72)$$

The quantities  $H_c, H_\theta, H_\phi, H_g$  will be calculated next.

#### Scan Mirror Axis Sensitivity $H_c$

Consider the perturbation on the scan mirror rotation given in (5.43) as,

$$C = (I - c_k^\times) \hat{C} \quad (5.73)$$

Based on Lemma D.5 (Angle-Axis Perturbation), the perturbation  $c_k$  in (5.73) can be written as,

$$c_k = \left[ \sin \hat{\beta} \Gamma \cdot I - \left( 1 - \cos \hat{\beta} \Gamma \right) \hat{a}^\times \right] \delta a + \hat{a} \Gamma \delta \beta \quad (5.74)$$

where  $\hat{\beta}$  is the nominal scale factor and  $\hat{a}$  is the nominal mirror axis. The misalignment on the scan mirror axis  $\delta a$  has two degrees of freedom corresponding to in-plane and out-of-plane errors. However, the out-of-plane error can be ignored because it manifests itself as a frame misalignment, and is estimated as part of  $T$ . The remaining degree of freedom is the in-plane misalignment which can be parametrized as,

$$\delta a = h_a(\hat{a}) \delta \alpha \quad (5.75)$$

where,

$$h_a(\hat{a}) = \begin{bmatrix} 0 \\ \hat{a}_3 \\ -\hat{a}_2 \end{bmatrix} \quad (5.76)$$

Substituting (5.75) into (5.74) gives,

$$c_k = \left[ \left( \sin \hat{\beta} \Gamma \cdot I - \left( 1 - \cos \hat{\beta} \Gamma \right) \hat{a}^\times \right) h_a(\hat{a}) : \hat{a} \Gamma \right] \begin{bmatrix} \delta \alpha \\ \delta \beta \end{bmatrix} \quad (5.77)$$

Consequently, the desired partial derivative of  $c_k$  with respect to  $\delta \alpha, \delta \beta$  is,

$$H_c = \left[ \left( \sin \hat{\beta} \Gamma \cdot I - \left( 1 - \cos \hat{\beta} \Gamma \right) \hat{a}^\times \right) h_a(\hat{a}) : \hat{a} \Gamma \right] \quad (5.78)$$

### **IPF<sub>0</sub> Alignment Sensitivity $H_\theta$**

It is seen from (5.44) that the variables  $\theta_k$  and  $\delta \theta$  represent the same physical perturbation of  $T$ . Consequently, the partial  $\frac{\partial \theta_k}{\partial (\delta \theta)}$  is a  $3 \times 3$  identity matrix, i.e.,

$$H_\theta = I_{3 \times 3} \quad (5.79)$$

### **TPF Alignment Sensitivity $H_\phi$**

The direction cosine matrix  $R$  is the mapping from the STA-defined Body frame to the TPF frame. It is parametrized in (4.14) as the following quadratic function of time,

$$R(q_R, b_r, c_r) \triangleq \left( I_{3 \times 3} - \left( b_r t + \frac{c_r t^2}{2} \right)^\times \right) R_0(q_R) \quad (5.80)$$

where  $R_0$  (and its equivalent quaternion  $q_R$ ) corresponds to the initial alignment at time  $t = 0$ , and  $b_r, c_r$  are parameters associated with the time-varying alignment drift.

Let  $R_0, b_r, c_r$  be perturbed about their current nominal estimates  $\hat{R}_0, \hat{b}_r, \hat{c}_r$  by the perturbations  $\delta a_r, \delta b_r, \delta c_r \in \mathcal{R}^3$  according to,

$$R_0(q_R) = (I - \delta a_r^\times) \hat{R}_0 \quad (5.81)$$

$$b_r = \hat{b}_r + \delta b_r \quad (5.82)$$

$$c_r = \hat{c}_r + \delta c_r \quad (5.83)$$

Substituting the perturbations (5.81)-(5.83) into (5.80) and rearranging gives (to first order),

$$R(q_R, b_r, c_r) = \left( I - \left( \left( \hat{b}_r t + \frac{\hat{c}_r t^2}{2} \right) + \left( \delta b_r t + \frac{\delta c_r t^2}{2} \right) \right)^\times \right) (I - \delta a_r^\times) \hat{R}_0 \quad (5.84)$$

$$= \left[ I - \left( \delta a_r + \delta b_r t + \frac{\delta c_r t^2}{2} \right)^\times \right] \left[ I - \left( \hat{b}_r t + \frac{\hat{c}_r t^2}{2} \right)^\times \right] \hat{R}_0 \quad (5.85)$$

$$\triangleq [I - \phi(t)^\times] R(\hat{q}_R, \hat{b}_r, \hat{c}_r) \quad (5.86)$$

where,

$$R(\hat{q}_R, \hat{b}_r, \hat{c}_r) = \left[ I - \left( \hat{b}_r t + \frac{\hat{c}_r t^2}{2} \right)^\times \right] \hat{R}_0 \quad (5.87)$$

$$\phi(t) = \begin{bmatrix} I & t \cdot I & \frac{t^2}{2} \cdot I \end{bmatrix} \begin{bmatrix} \delta a_r \\ \delta b_r \\ \delta c_r \end{bmatrix} \quad (5.88)$$

The variable  $\phi_k$  is defined by evaluating  $\phi(t)$  at the  $k$  centroid time  $T_k$  to give,

$$\phi_k = \begin{bmatrix} I & T_k \cdot I & \frac{T_k^2}{2} \cdot I \end{bmatrix} \begin{bmatrix} \delta a_r \\ \delta b_r \\ \delta c_r \end{bmatrix} \quad (5.89)$$

Hence, the desired sensitivity  $H_\phi$  is,

$$H_\phi = \begin{bmatrix} I & T_k \cdot I & \frac{T_k^2}{2} \cdot I \end{bmatrix} \quad (5.90)$$

It is noted that after the perturbations  $\delta a_r, \delta b_r, \delta c_r$  are estimated, they are applied to update

the nominal parameters as follows,

$$\hat{R}_0 \leftarrow (I - \delta \hat{a}_r^\times) \hat{R}_0 \quad (5.91)$$

$$\hat{b}_r \leftarrow \hat{b}_r + \delta \hat{b}_r \quad (5.92)$$

$$\hat{c}_r \leftarrow \hat{c}_r + \delta \hat{c}_r \quad (5.93)$$

In this manner the estimate of the initial alignment  $\hat{R}_0$  (and its equivalent quaternion  $\hat{q}_R$ ) is kept as a large angle, while the time-varying drift is kept as a small angle. This formulation is used because  $\hat{R}_0$  is expected to be on the order of degrees, while the time-variations are expected to be only a few arcseconds.

### Gyro Attitude Offset Sensitivity $H_g$

The true gyro offset  $G$  can be written as,

$$G(T_k) = (I - \gamma(T_k)^\times) G^\circ(T_k) \quad (5.94)$$

where  $G^\circ$  is a nominal gyro offset provided by the gyro pre-processor. The quantity  $G^\circ$  is computed by the gyro pre-processor by using a certain nominal rate estimate  $\omega_m^\circ$ . (The actual choice of  $\omega_m^\circ$  by the pre-processor will be discussed in Section 5.4).

Clearly the quantity  $\gamma$  is not known. However, given that the true rate  $\omega$  is related to the approximate rate  $\omega_m^\circ$  according to,

$$\omega = \omega_m^\circ + b_g + c_g \quad (5.95)$$

(by definition of  $b_g$  and  $c_g$  in (4.19)) then as shown in the Appendix, Lemma D.2, the quantity  $\gamma$  can be parametrized linearly in  $b_g, c_b$  as,

$$\gamma(T_k) = H_g(T_k) \begin{bmatrix} b_g \\ c_g \end{bmatrix} \quad (5.96)$$

$$H_g(T_k) = [ \Lambda_b(T_k) \quad \Lambda_c(T_k) ] \quad (5.97)$$

where the quantities  $\Lambda_b$  and  $\Lambda_c$  are obtained by integrating the matrix differential equations,

$$\left( \dot{\Lambda}_b = -(\omega_m^\circ)^\times \Lambda_b + I \right) \Big|_{t_j}^{T_k(j)} \quad \text{with I.C.} \quad \Lambda_b(t_j) = 0 \quad (5.98)$$

$$\left( \dot{\Lambda}_c = -(\omega_m^\circ)^\times \Lambda_c + t \cdot I \right) \Big|_{t_j}^{T_k(j)} \quad \text{with I.C.} \quad \Lambda_c(t_j) = 0 \quad (5.99)$$

Here, the notation  $(\cdot)|_a^b$  denotes integration over the time interval  $t \in [a, b]$ , and  $t_j$  denotes the starting time of the  $j$ 'th maneuver.

Since the  $b_g, c_g$  parameters are not known, perturbations  $\delta b_g, \delta c_g$  are defined such that,

$$b_g = \hat{b}_g + \delta b_g \quad (5.100)$$

$$c_g = \hat{c}_g + \delta c_g \quad (5.101)$$

Using the current nominal estimates  $\hat{b}_g, \hat{c}_g$ , an estimate  $\hat{\gamma}$  of  $\gamma$  at the centroid time  $T_k$  is defined by,

$$\hat{\gamma}(T_k) \triangleq H_g(T_k) \begin{bmatrix} \hat{b}_g \\ \hat{c}_g \end{bmatrix} \quad (5.102)$$

A corresponding estimate of  $G$  is defined by,

$$\hat{G}(T_k) \triangleq (I - \hat{\gamma}(T_k)^\times) G^\circ(T_k) \quad (5.103)$$

This estimate is used in the measurement equation to form the prediction. Combining (5.94) and (5.103) yields (to first order),

$$G(T_k) = \left( I - \left( \gamma(T_k) - \hat{\gamma}(T_k) \right)^\times \right) \hat{G} \quad (5.104)$$

By comparing (5.104) with the definition of  $g_k$  in (5.46) it is clear that the relation between  $g_k$  and  $\gamma$  is given by (to first order),

$$g_k \triangleq \gamma(T_k) - \hat{\gamma}(T_k) \quad (5.105)$$

Accordingly,  $g_k$  can be found by subtracting (5.96) from (5.102) to give,

$$g_k(T_k) = H_g(T_k) \begin{bmatrix} \delta b_g \\ \delta c_g \end{bmatrix} \quad (5.106)$$

It is seen from (5.106) that  $H_g$  is the desired sensitivity function and its formula is given by (5.97).

For computational savings, the gyro pre-processor computes the quantities  $\{G^\circ(T_k), H_g(T_k)\}$  once and stores them. These quantities are then used in the prediction equation (5.103) and the sensitivity equation (5.106) at each centroid time  $T_k$  and during each filter cycle.

## 5.4 IPF Gyro Pre-Processor

In order to reduce computation, the gyro pre-processor computes the quantities  $\{G^\circ(T_k), H_g(T_k)\}$  once and stores them. This (with the attitude quaternion associated with the start of each sandwich maneuver), summarizes the attitude information needed by the Kalman filter. With this approach, the large attitude file (AFILE) is reduced by several hundred to the size of the centroid file (CFILE), and no further processing of attitude data is required to generate the sensitivity equations used by the Kalman filter.

## 5.5 Calculation of $G^\circ$

The gyro pre-processor computes the nominal gyro offset  $G^\circ(T_k)$  by integrating the equation,

$$\left(\dot{G}^\circ = -(\omega_m^\circ)^\times G^\circ\right)\Big|_{t_j}^{T_k(j)} \quad (5.107)$$

(conceptual only - the actual computation is performed using quaternions). Here, the nominal rate  $\omega_m^\circ$  is assumed to have the form,

$$\omega_m^\circ = w_m + b_g^\circ + c_g^\circ t \quad (5.108)$$

where  $w_m$  is the gyro rate measurement that has an on-board scale factor and misalignment correction but does not have an on-board bias correction. The quantities  $b_g^\circ, c_g^\circ \in \mathcal{R}^3$  are nominal estimates of the 3-axis gyro bias and bias rate parameters, respectively. Depending on the user's response to a query during operation, the values of  $b_g^\circ, c_g^\circ$  can be chosen one of three possible ways:

1. The earliest on-board GCF gyro bias correction stored in the attitude history file (most common choice)
2. Default values provided in the RN file
3. Previous IPF filter run estimate of bias correction

## 5.6 Calculation of $H_g$

The gyro sensitivity matrix  $H_g(T_k)$  is computed by integrating the matrix differential equations (5.98)(5.99) (see also, Appendix, Lemma D.2).

The integration of the sensitivities (5.98)(5.99) is aided by an analytical result proved in Section D.16, and briefly summarized here. Specifically, the integral of (5.98) over  $\Delta T$  (the

fast gyro sampling period) can be written as,

$$\Lambda_b(k+1) = e^{-(\omega_m^\circ)^\times \Delta T} \Lambda_b(k) + \int_0^{\Delta T} e^{-(\omega_m^\circ)^\times (\Delta T - \tau)} I d\tau \quad (5.109)$$

$$= e^{-(\omega_m^\circ)^\times \Delta T} \left( \Lambda_b(k) + \int_0^{\Delta T} e^{(\omega_m^\circ)^\times \tau} d\tau \right) \quad (5.110)$$

$$= M_1 \left( \Lambda_b(k) + M_2 \right) \quad (5.111)$$

where from Lemma D.15 it follows that  $M_1$  is a direction cosine matrix of the form,

$$M_1 = M_1 \left( \frac{\omega_m^\circ}{\|\omega_m^\circ\|}, \|\omega_m^\circ\| \Delta T \right) \triangleq e^{-(\omega_m^\circ)^\times \Delta T} \quad (5.112)$$

$$= \cos(\|\omega_m^\circ\| \Delta T) \cdot I + \left( 1 - \cos(\|\omega_m^\circ\| \Delta T) \right) \frac{\omega_m^\circ}{\|\omega_m^\circ\|} \frac{(\omega_m^\circ)^T}{\|\omega_m^\circ\|} \\ - \sin(\|\omega_m^\circ\| \Delta T) \frac{(\omega_m^\circ)^\times}{\|\omega_m^\circ\|} \quad (5.113)$$

and by Lemma D.16,  $M_2$  is given by the expression,

$$M_2 = M_2 \left( \frac{\omega_m^\circ}{\|\omega_m^\circ\|}, \|\omega_m^\circ\| \Delta T \right) \triangleq \int_0^{\Delta T} e^{(\omega_m^\circ)^\times \tau} d\tau \quad (5.114)$$

$$= \frac{\sin(\|\omega_m^\circ\| \Delta T)}{\|\omega_m^\circ\|} \cdot I + \left( \Delta T - \frac{\sin(\|\omega_m^\circ\| \Delta T)}{\|\omega_m^\circ\|} \right) \left( \frac{-\omega_m^\circ}{\|\omega_m^\circ\|} \right) \left( \frac{-\omega_m^\circ}{\|\omega_m^\circ\|} \right)^T \\ - \left( \frac{\cos(\|\omega_m^\circ\| \Delta T) - 1}{\|\omega_m^\circ\|} \right) \left( \frac{-\omega_m^\circ}{\|\omega_m^\circ\|} \right)^\times \quad (5.115)$$

Analogous to (5.111), the integration of (5.99) is performed (approximately) as,

$$\Lambda_c(k+1) = M_1 \left( \Lambda_c(k) + t_k M_2 \right) \quad (5.116)$$

where  $t_k$  is the time of the  $k$ 'th sample (taken at the fast gyro sampling rate).

## 6 ALGORITHMS

### 6.1 IPF Filter Algorithm Overview

A functional block diagram of the IPF filter algorithm is shown in Figure 6.1. The IPF filter algorithm is divided into five major components,

- Filter Initialization and Input data preparation
- Gyro Preprocessor
- Kalman Filter Execution and Iterations
- Least Squares Filter Execution and Data Analysis
- Output Data Processing, Write Output Files, Display Output Figures.

**Initialization:** Upon running the IPF filter, the program queries the user for the six-digit extension of the RN file to be processed. For tracability, this same six-digit extension is used in the name of all output files generated by the run. The program reads the specified RN file from the current directory, and loads all necessary database parameters. The IPF filter loads all input files specified within the RN file (i.e., AA, AS, CA, CB, CS, FF, TT and frame-table files) into the MATLAB workspace. Other filter control variables (e.g., the number of initial filter iterations etc.) are also loaded from the RN file. The software is then initialized based on this information, and the CA and CB files are merged chronologically, and edited to produce a cleaned CC File.

**Gyro Preprocessing:** Once the initialization step is completed, the IPF software enters the gyro preprocessing functional block as shown in Figure 6.1. The IPF filter queries the user for the desired method of gyro linearization. The gyro pre-processor compresses the data from a large AA file into two significantly smaller AG and AC files. The AG and AC files contain attitude data and sensitivity function evaluations at each of the centroid times. Hence, the dimension of the AG and AC files are compressed, being on the order of the number of centroids in the merged CC centroid file. With this approach, the computationally intensive gyro propagation and sensitivity calculations are computed only once per run.

**Kalman Filter Execution:** Once the gyro pre-processor step is completed, the IPF filter program executes the IPF Kalman filter subroutine (cf., Figure 6.1). This filter subroutine first initializes the mean and covariance estimates of the filter. The filter subroutine then sweeps through the entire centroid data set, and accumulates a filter correction. At the end of the file, the accumulated filter correction is applied, and the system is relinearized about the new nominal to prepare for the next iteration of the filter cycle. After the filter iterates for a specified number of iterations (as defined in RN file), a convergence plot is provided so that the user can choose to either proceed or stop the iteration process.

**Least Squares Filtering and Analysis:** Upon completion of the main Kalman filter execution, the program starts an independent *Iterated Linearized Least Squares Filter* subroutine. This calculation is optional and can be disabled by setting a flag in the RN file. However, this calculation is very useful for diagnostic purposes,

- i)* It provides an independent state estimate to compare to the Kalman filter estimate
- ii)* It provides an independently calculated covariance estimate
- iii)* It provides a residual-based scale factor which scales the least-squares covariance and makes it more representative of the effective noise in the given data set.

The least squares filter iteration does not have to be monitored by the user since, for simplicity, it is taken to be the same the total Kalman filter iterations used in the earlier step.

**Output Processing and Plotting:** After the Kalman filter and least-squares processing is completed, the program executes the output data processing subroutine. This routine calculates all the necessary variables for output file generation and output diagnostic plots. The program then writes the output files, plots the filter results, and saves all workspace variables without requiring any additional user inputs.

## 6.2 Input Database and User Interface

### 6.2.1 User Configuration Parameters

The IPF filter has several configuration parameters that can be set by the user to control filter execution behavior. These configuration parameters are listed in Table 6.14 (see also, IPF Filter User's Guide D-Document [13] for more details).

As shown in Table 6.14, the user can

- i)* choose the initial gyro bias for linearization
- ii)* select which state parameters should be estimated through masking vectors
- iii)* define the PCRS locations,
- iv)* enable or disable two possible filtering approaches (i.e., Kalman filtering and Least Squares)
- v)* force the filter to operate in "Slit" mode
- vi)* select a LITE mode of operation
- vii)* use previously generated gyro pre-processor data

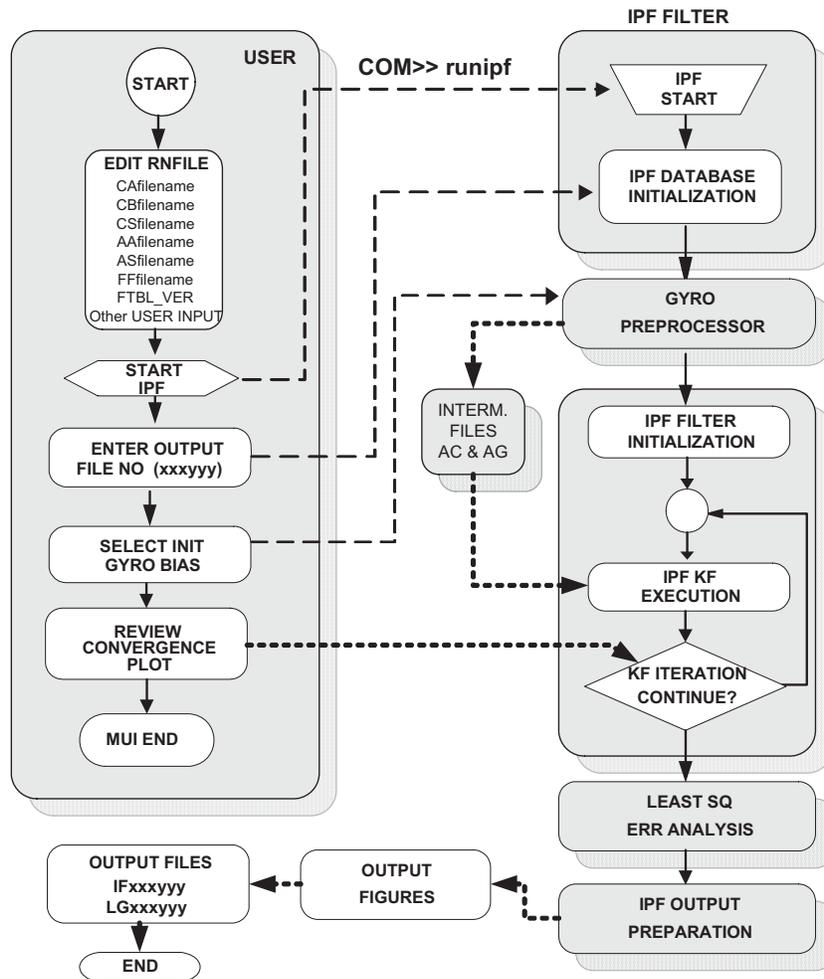


Figure 6.1: IPF Functional Block Diagram

*viii)* use FLUTE test data

*ix)* initialize mean and covariance from user supplied data

*x)* define customized centroid data editing

Note that if the user hasn't specified the mask vectors and the "Slit" mode operations, the IPF filter loads the default database.

Name	Description	Source	Method
bg0user	User defined gyro bias for linearization	RN file	Use Option 2 in MUI
cg0user	User defined gyro bias rate for linearization	RN file	Use Option 2 in MUI
mask1	User defined mask vector for $p_1$	RN file	specify and uncomment
mask2	User defined mask vector for $p_2$	RN file	specify and uncomment
FT_user4-9	User defined PCRS quaternion bypass frametable	RN file	set USR_PCRS_flg = 1
kf_enable	Kalman filter enable	RN file	1 for enable, 0 for disable
lsqrf_enable	Least Squares filter enable	RN file	1 for enable, 0 for disable
Slit_Flag	Slit Mode Manual Selection	RN file	1:Process as a Slit, 0:Not a Slit
IPF_LITE_MODE	LITE Mode Selection	RN file	0: Normal Mode, 1-3 LITE Mode
USE_PREV_AGAC	Bypass Gyro Preprocess and use old data	RN file	Set to 1 and specify AC & AG file name
IPF_TEST_MODE	Use FLUTE data to extract true error (testing only)	RN file	Set to 1 and specify TT file name
Use_RNinit_flg	Use user supplied mean and covariance to initialize filter	RN file	Set to 1 and specify mean and covariance
CEdit_FLG	User Specified Centroid Data Editing Instruction	RN file	Set to 1 and specify edit data

Table 6.14: Special Operational Features of IPF Filter

## 6.2.2 Merging and Editing Centroid Files

There are three different approaches to edit out bad centroids within the IPF filter. These three approaches are summarized in Table 6.15. The purpose of these editing approaches is to i) to detect any bad data glitches autonomously, and ii) allow the user to manually edit out any unwanted centroid data (or complete maneuvers) if necessary.

Instructions for using the RN file approach are described within the RN file. In the CC file approach, the user manually puts 99999's into columns 8 and 9 of the CA and/or CB files to indicate bad centroids. The IPF filter then looks for the 99999's and removes these centroids from consideration. In the AA file approach, the user puts a "1" into column 21 (i.e., the IBAD flag) of the AA file to indicate bad attitude data. When the IPF filter encounters the IBAD indication, it removes the entire maneuver from consideration.

The block diagram associated with the merging and editing subroutine is shown in Figure 6.2. This subroutine loads the supplied CA and CB input files, and then produces a cleaned and chronologically merged CC file. The combined CC file contains both the instrument and PCRS centroid data in chronological order. The CC file is indexed in two different ways: one from ordering the merged raw data set, and the second from ordering the merged cleaned-up data set (i.e., after editing out the indicated bad data).

Method	Trigger	Impacted Data	Comment
RN FILE	CEdit_CX	Remove $C_x$ component of a centroid data	Specified by raw CCFILE row number
	CEdit_CY	Remove $C_y$ component of a centroid data	Specified by raw CCFILE row number
	CEdit_MN	Remove specified Maneuver	Specified by Maneuver number
CC FILE	Col(8)	Remove $C_x$ with detection of 99999's	User can manually edit out unwanted data by placing 99999's (in CA, CB)
	Col(9)	Remove $C_y$ with detection of 99999's	
AA FILE	Col(21)	Removes entire maneuver when IBAD detected	Compares time tags between AA File and CC File

Table 6.15: Three Different Centroid Data Editing Methods

As shown in Figure 6.2, the merging process first loads the CA and CB files into the MATLAB environment as two matrices. These two matrices are then stacked and sorted into chronological order. This produces a stack of time-ordered raw centroid data. Four additional columns are added to this stack as indicated in Table 6.16. The Centroid and Maneuver numbers are added to the file. At this point, the merging tool detects any 99999's in the centroid data field. Once detected, Cx\_valid and Cy\_vaild flags are set to zero, otherwise they are set to 1. Next, the Cx\_valid and Cy\_vaild flags are set based on the user-supplied editing instructions (as specified in RN file, see Table 6.15). With the given Cx\_valid and Cy\_vaild flags, the raw file is trimmed down by eliminating the indicated rows.

The next step is to clean out any unwanted maneuvers. Here, the merging routine applies the user-supplied maneuver editing instructions given in the RN file. Then the attitude data AA file is searched for any IBAD flags, and entire maneuvers are removed which are associated with any bad data. The raw centroid file is screened and trimmed down with these set flags, which provides a cleaned centroid file for use with the IPF filter.

Column	16	17	18	19
	Cx valid	Cy valid	Raw Row number	Final Row number

Table 6.16: Augmented Centroid File Columns

### 6.2.3 Gyro Preprocessing

A standard implementation of an iterated and linearized Kalman filter for this problem requires that the gyro attitude be propagated and gyro sensitivities be calculated at every centroid time and for each iteration. This approach is very time-consuming, considering that

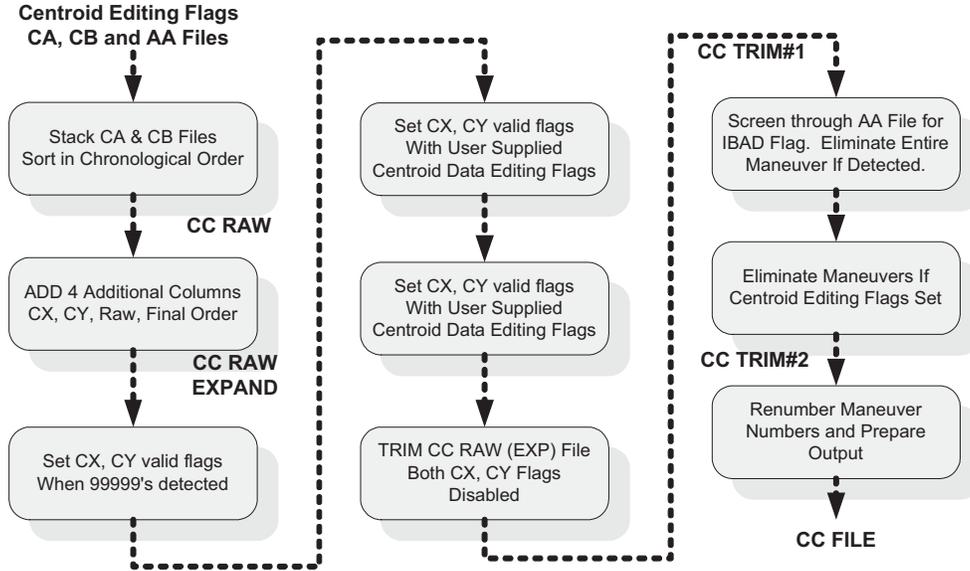


Figure 6.2: Merging and Editing Centroid Files

the attitude history is at 10 Hz and the attitude data file is typically tens of megabytes in size.

The gyro attitude linearization within the iteration loop is avoided by linearizing twice: once globally as a small perturbation about a nominal gyro bias estimate, and a second time locally as a small perturbation to the perturbation. This approach is aided by the fact that the initial on-board estimate of the gyro drift bias from the onboard filter is quite good and serves as a valid initializer from linearizing the entire time history. The first linearization is performed by a gyro pre-processor which propagates the gyro sensitivities about the global linearization point just once-per-run before the Kalman filter iteration starts. The results of these attitude propagation and sensitivity calculations are stored in memory at only the centroid times (of smaller dimension than the attitude history), and are used in all further iterations (i.e., local relinearizations) of the filter.

An overview of the gyro pre-processor is given in Figure 6.3. Functionally, the main objective of the gyro pre-processor is to input the centroid and attitude files and to produce the much more compact AG and AC files <sup>1</sup> for the main filter routines. The AG file contains propagated gyro attitude histories and sensitivities (cf., Section 5.4) and the AC file is the trimmed AA file where each row is time aligned to the corresponding row of the centroid data file.

The gyro pre-processor allows four specialized types of preprocessing modes. These modes are summarized in Table 6.17, and consist of one normal mode and three LITE modes.

<sup>1</sup>The dimension of the compressed file is on the order of the number of centroids in the merged (CA plus CB) centroid files

The normal mode linearizes and propagates attitude using the gyro data corrected by the bias estimates  $b_{g0}$  and  $c_{g0}$ . Sensitivities are also calculated by linearizing about these same nominal values.

In contrast, when any one of the LITE modes is invoked, the gyro bias parameters are not estimated and their sensitivities are not computed <sup>2</sup>. Instead, LITE mode #1 uses the on-board GCF corrected gyro rates to propagate the gyro attitude. LITE mode #2 uses the raw Star Tracker (STA) quaternion to provide the attitude history at each centroid time. Note that the raw tracker data requires a transformation from AST to STA frame (cf., Section 3.3). The last LITE mode #3 uses the attitude history from the onboard attitude observer (smoothed STA solution).

Mode	Type	Linearization	Gyro Attitude	Sensitivities
0	Normal	$b_{g0}, c_{g0}$	Propagate Gyro	YES
1	GCF Corrected	$b_g$	Propagate GCF Corrected Gyro	NO
2	Raw STA	NONE	Use Raw STA quaternion	NO
3	Filtered STA	NONE	Use Onboard Filter Attitude	NO

Table 6.17: LITE Mode Description

As shown in Figure 6.3, the pre-processor starts with the nominal gyro bias to be linearized about, the LITE mode flag and input files (CC and AA). The IPF filter queries the user for the desired method of gyro linearization (selecting appropriate  $b_{g0}$  and  $c_{g0}$ ). The MUI gives the IPF filter user three different choices to select the linearizing gyro drift bias. Option 1 allows the user to specify that the nominal drift biases be taken from the on-board gyro-calibration filter (GCF) (recommended); Option 2 allows one to invoke a user-defined database; and Option 3 allows to the user to choose the best estimated value of gyro drift bias based on the previous IPF filter run. Given these initial conditions, the pre-processor subdivides into two major functional branches, gyro based processing (Mode 0 and 1) and STA based processing (Mode 2 and 3).

For the gyro based modes, initially the attitude data are scrolled down until the attitude data time aligns with the first centroid data. While looping for each centroid data, gyro rates are propagated (GCF corrected gyro data for Mode #1) and the gyro sensitivities are integrated. When the propagated data are available at each centroid time, the data are written in the AG file. Compressed attitude AC file is also obtained. When end of maneuver is detected, the gyro quaternion and sensitivities are reset.

For the two STA based modes, the propagated gyro quaternion is replaced with appropriate attitude quaternion, raw STA quaternion for the mode #2 and the onboard attitude quaternion for the mode #3. No gyro propagation is performed in this path. Similar to the gyro based case, the AG and AC files are created.

This computationally intensive gyro propagation and sensitivity calculations are com-

---

<sup>2</sup>Gyro parameters are masked out by using the mask vector

puted only once per run and the intermediary files (AG and AC) are stored as MATLAB ASCII files and can be reused in future runs on the same data set, without executing the pre-processor if the user specifies such an option in the RN file (it is recommended to do so).

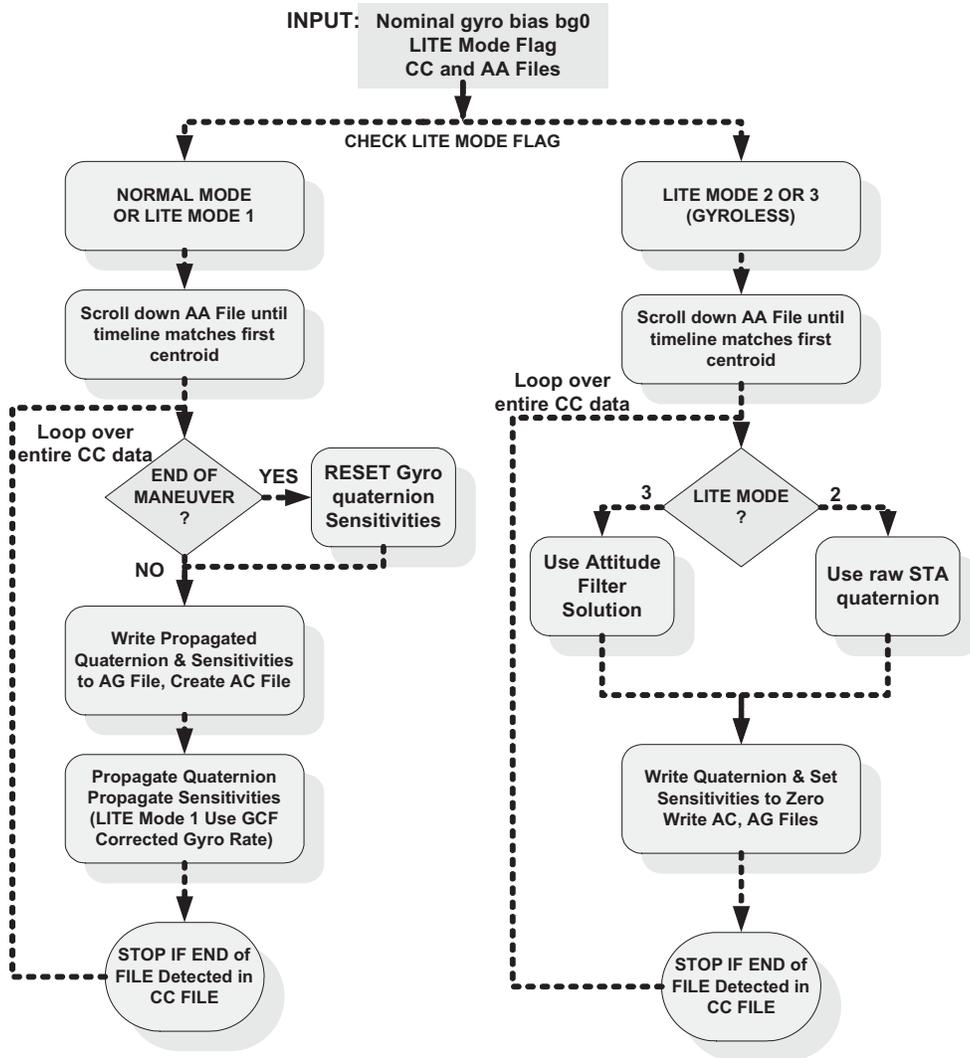


Figure 6.3: Gyro Preprocessor functional block diagram

### 6.3 Scalable Kalman Filter

The core functional block of the IPF filter is the iterated linearized Kalman filter routine which will be discussed in this subsection. The estimates and output data of this routine is used to produce the corresponding IPF filter run output files. Even though the least squares filter that follows the Kalman filter execution also provide the estimates, the least square estimates are only used to provide the independent data quality analysis and bounding the Kalman filter results<sup>3</sup>. As discussed previously, the Kalman filter provides state estimate corrections from maneuver to maneuver and updates the state estimates only once at the end of the survey.

The functional block diagram of the Kalman filter is shown in Figure 6.4 and Figure 6.5. The main Kalman filter routine starts immediately when all input data files and database are obtained and gyro pre-processor completes (or AG and AC files loaded into the MATLAB workspace). As shown in Figure 6.4, the Kalman filter has outer iteration loop that iterates and stops at a given database value as specified in the RN file (`N_KF_Iteration`). Once the iteration reaches `N_KF_Iteration` and the MUI flag `ask4more_iteration_flg` is set to 1, then the MUI displays the filter convergence diagnostics plots and waits for the user to proceed with additional iterations or to terminate. If the user decided with additional runs, the filter iterates `N_KF_Addrun` times as specified in the RN file. Once terminated, the IPF filter software proceeds to the Least Squares data analysis routine.

Within the iteration loop, the detailed Kalman filter operations are described in the functional block diagram of Figure 6.5. As shown in this figure, the data collection occurs at each centroid data points (inner loop), and the correction to the states estimates are calculated every time end of the maneuver reaches. The data collection processes following calculations: i) Obtain a centroid measurement  $y$  and its predict  $\hat{y}$  to form a measurement residual  $y_{res} = y - \hat{y}$ , ii) Calculate sensitivities  $\mathcal{K}_1$  and  $\mathcal{K}_2$  and apply mask to form the linearized observation matrix  $H_{mask}$ . This masking scalability<sup>4</sup> allows the use of same algorithm for various instrument type, iii) Calculate effective noise sigma  $R_{tmp}$  from the centroid measurement noise sigma  $R_c$  and the initial attitude uncertainties  $\Lambda_\psi$  (cf., Sum Factorization Lemma D.17 for detailed calculations), and iv) Stack the measurement residual, the masked sensitivities and the effective noise sigma until the process reaches end of the maneuver.

Whenever end of the maneuver is reached, the stacked vector and matrices of step iv) are used to calculate the Kalman filter gain and a posteriori square-root covariance by using a  $QR$  factorization method (cf., Section 5.1 of the formulation and the Lemma D.1 for detailed calculations). The corrections to the state estimates are accumulated as indicated in the last functional block of Figure 6.5.

---

<sup>3</sup>The Least Squares filter also provides an estimate of the prediction residual noise and provides a scaling of the Kalman filter sigma estimates

<sup>4</sup>The IPF algorithm allows estimation of any sub-vector in any combinations (from  $\mathcal{R}^{37}$  by applying and removing mask vector to the state vector and the square-root covariances.

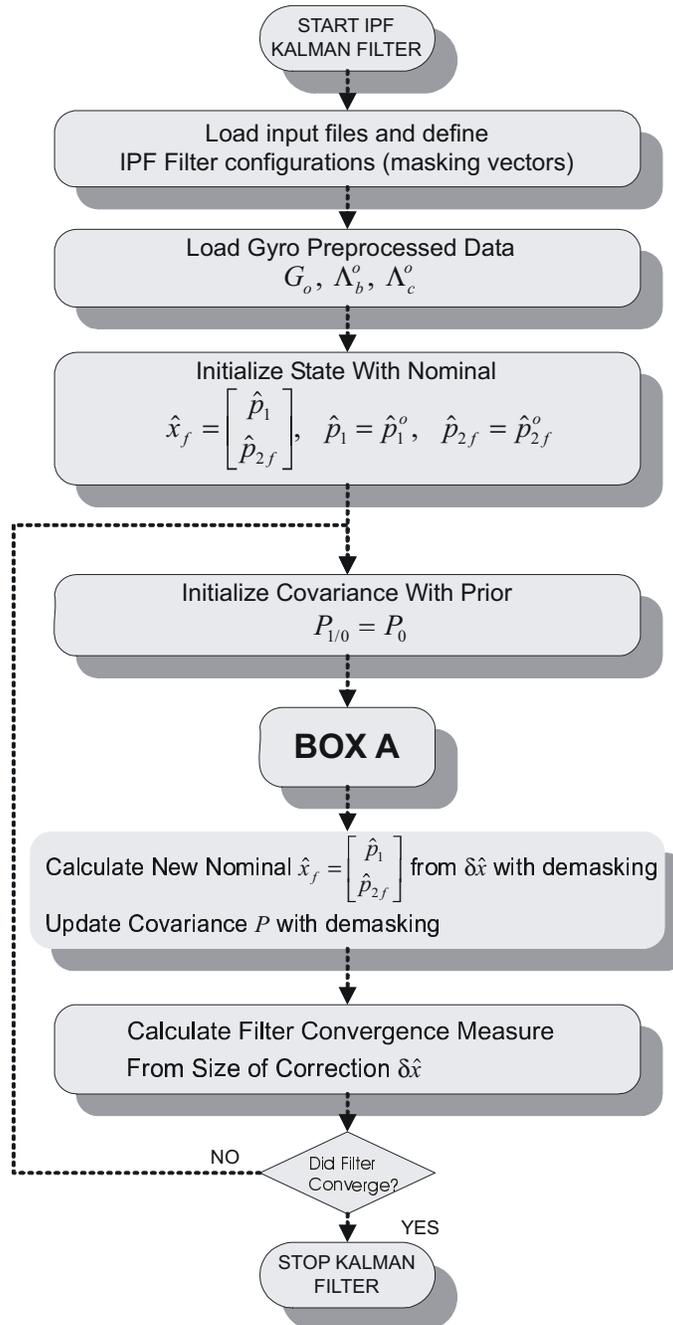


Figure 6.4: Overview of SRILKF Architecture

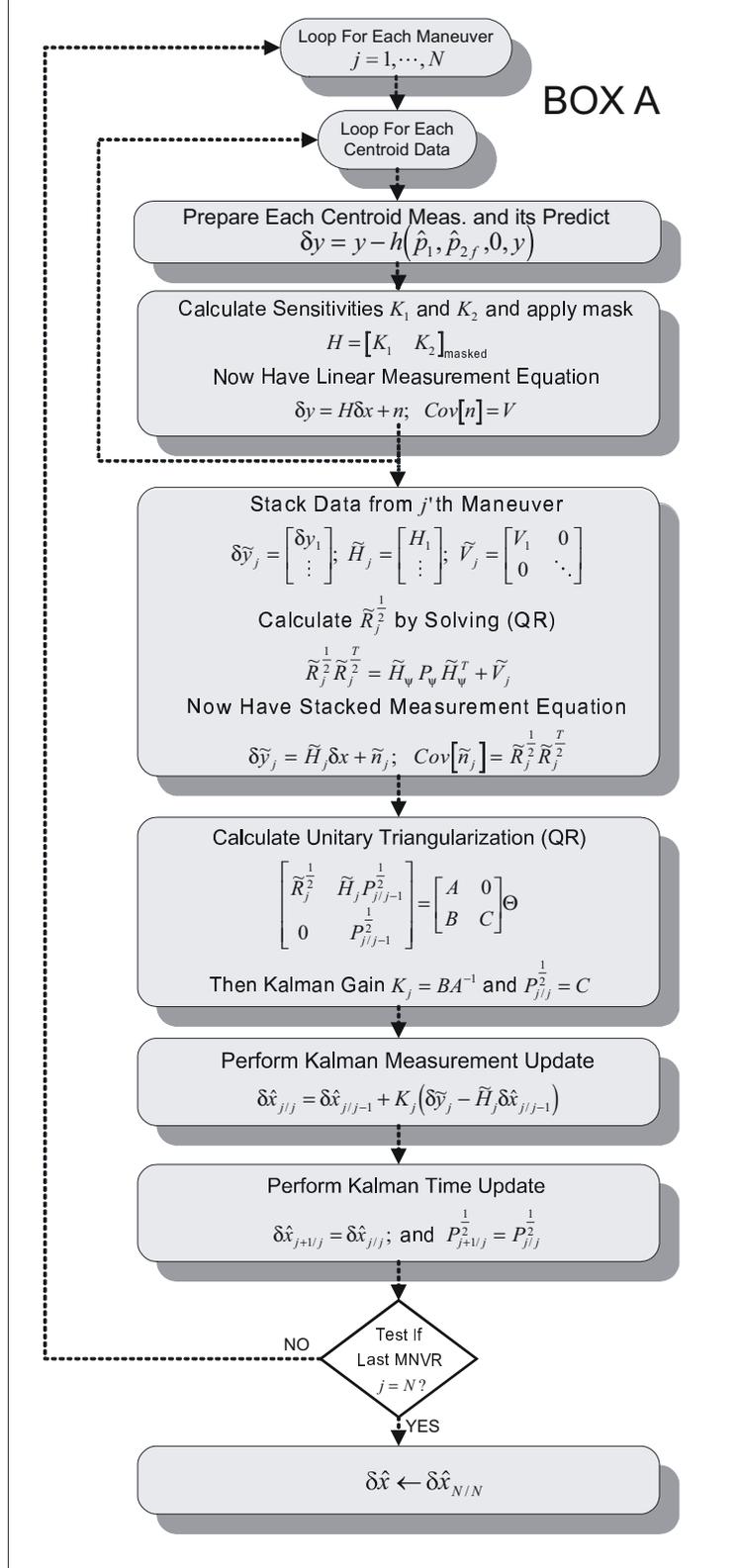


Figure 6.5: Overview of SRILKF Architecture (BOX A)

## 6.4 Least Squares Data Analysis

The IPF software utilizes an independent *Iterated Linearized Least-Squares Filter* for the purpose of post Kalman filter data analysis. The Least-Squares filter execution follows the Kalman filter processing<sup>5</sup> and it provides independent state estimate such that the Kalman filter results can be checked against. Since the Kalman filter incorporates a prior state information (initial estimate of covariance), the Kalman filter should theoretically perform better than the least squares solution. Since the least squares filter algorithm is completely independent from the Kalman filter, it provides a good sanity check and a reasonable covariance bound. In addition, the least squares solution generates an estimate of measurement residual sigma such that the resulting sigma scale factor can be used to scale the estimates with the effective noise of the given data set.

The overall functional block diagram of the least squares data analysis is shown in Figure 6.6. As shown in this figure, the least squares filter starts with initializing variables. Similar to the Kalman filter case, the filter starts with initial estimates of the  $p_1$  and  $p_{2f}$  vectors; however, an initial square-root covariance matrix is not needed or used. The least squares filter has two loops: a re-linearizing outer loop, and centroid data processing inner loop.

The inner centroid data processing loop starts by stacking measurement and sensitivity matrices for the duration of one maneuver. The starting point is equation (5.20) where all the centroids from the  $j$ 'th maneuver are stacked into a single measurement perturbation vector,

$$\delta\tilde{y}_j = \tilde{H}_j\delta x + \tilde{n}_j \quad (6.1)$$

The measurement noise covariance is given by,

$$Cov[\tilde{n}_j] \triangleq \tilde{R}_j = \tilde{R}_j^{\frac{1}{2}}\tilde{R}_j^{\frac{T}{2}} \quad (6.2)$$

where the calculation of  $\tilde{R}_j^{\frac{1}{2}}$  has been described earlier based on factorizing (5.30). The observations are unit-normalized by premultiplying both sides of (6.1) by  $\tilde{R}_j^{-\frac{1}{2}}$ , and then stacked for all  $M$  maneuvers to give the regression equation,

$$\tilde{Y} = \tilde{H}\delta x + \eta \quad (6.3)$$

where,

$$\tilde{Y} \triangleq \begin{bmatrix} \tilde{R}_1^{-\frac{1}{2}}\delta\tilde{y}_1 \\ \vdots \\ \tilde{R}_M^{-\frac{1}{2}}\delta\tilde{y}_M \end{bmatrix}; \quad \tilde{H} \triangleq \begin{bmatrix} R_1^{-\frac{1}{2}}\tilde{H}_1 \\ \vdots \\ R_M^{-\frac{1}{2}}\tilde{H}_M \end{bmatrix}; \quad \eta \triangleq \begin{bmatrix} \tilde{R}_1^{-\frac{1}{2}}\tilde{n}_1 \\ \vdots \\ \tilde{R}_M^{-\frac{1}{2}}\tilde{n}_M \end{bmatrix} \quad (6.4)$$

---

<sup>5</sup>Processing of only one type of filter is also possible via proper RN file specifications; however, an execution of least squares filter does not produce any IPF filter output files.

After the regression equation (6.3) is formed, some columns of  $\tilde{H}$  are scaled for better numerical accuracy (i.e. to provide a better conditioning for matrix inversion). This is done by defining a diagonal scaling matrix  $S$ , and using  $S$  to form the scaled regression matrix  $\tilde{H}_T = \tilde{H}S$ , and scaled parameter vector  $\delta x_T = S^{-1}\delta x$ . The parameters that are scaled by  $S$  relate to the mirror rotation angle  $\Gamma$  and powers of time. A QR factorization method is used to provide the least squares solution  $\delta \hat{x}_T^*$  to the scaled regression equation:

$$\tilde{Y} = \tilde{H}_T \delta x_T + \eta. \quad (6.5)$$

The resulting estimate  $\delta \hat{x}_T^*$  is scaled back to the original physical units to obtain  $\delta \hat{x}$ , and the state estimate is updated using this perturbation. The problem is relinearized about the new state estimate, and the overall process is repeated in an iterative fashion until convergence.

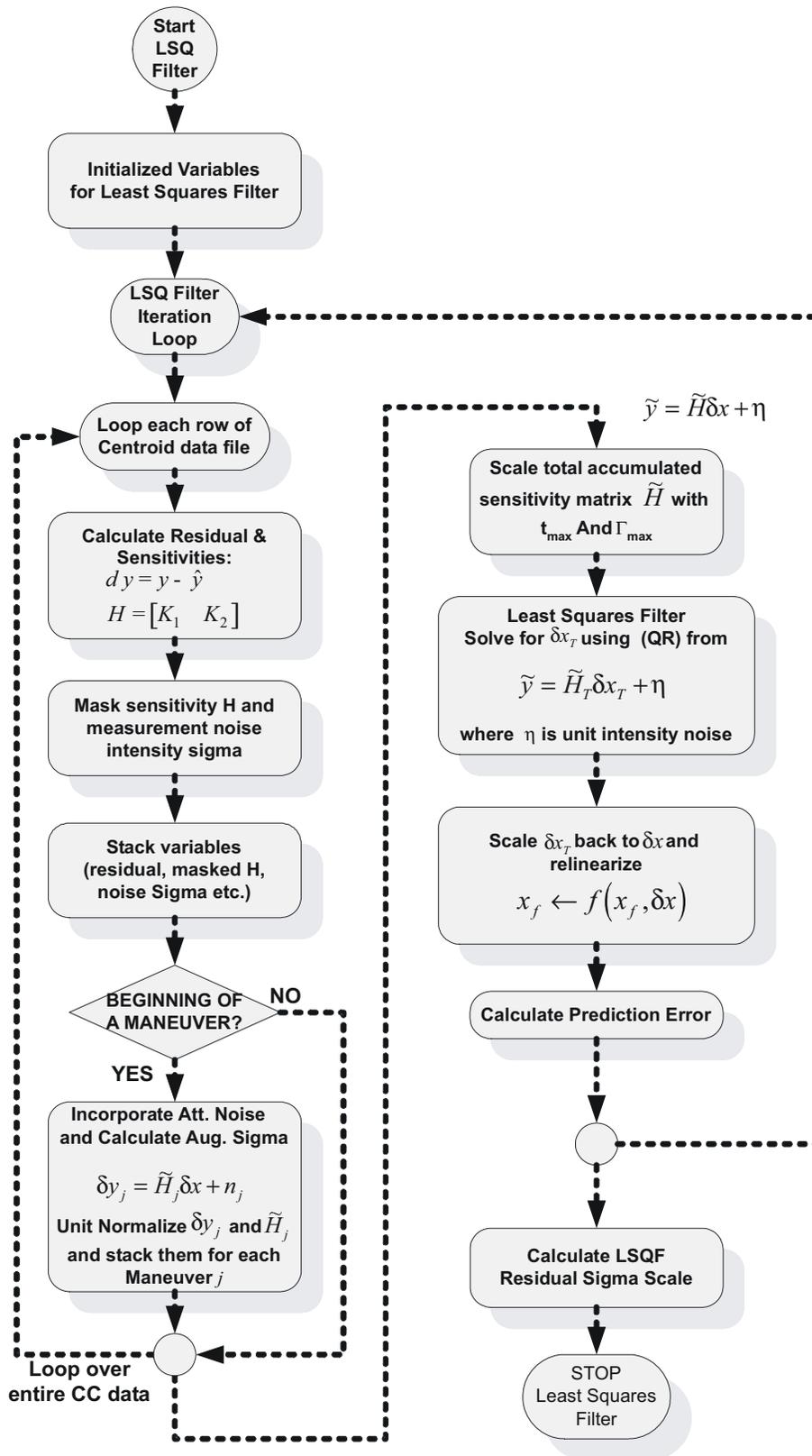


Figure 6.6: Least Squares Filter and Analysis Diagram

## 6.5 Output Preparation

Once the IPF filter is executed, the IPF filter autonomously processes data for output preparation and generates output figures and three output files, namely the IF, LG, and TR Files (cf., Figure 6.7). (In the special case that multiple MIPS data sets are being processed, the IPF filter produces an MF file instead of an IF file). These files are described below,

1. IF file - Output file which summarizes all products (i.e., estimated parameters and covariances) of the IPF filter run. This is the main product used by all IPF filter customers.
2. LG file - A log file containing a record of all aspects of the specified IPF filter execution sequence. This file would allow, for example, the IPF filter to be re-run at a future time and give identical results.
3. TR file - A tar file containing a copy of all input and output files associated with the IPF filter run, and all run products. This is the main product stored on DOM for archival purposes.
4. MF file - Similar to the IF file, but only used when processing multiple MIPS data sets (i.e., the MIPS multi-run tool).

Once the output figures and files are generated, the IPF software saves all workspace variables (as a MT file) and terminates the IPF filter operation.

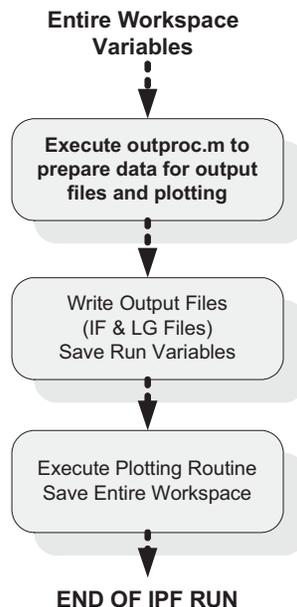


Figure 6.7: IPF Output Data Processing And Plotting Routine

### 6.5.1 Initial Attitude Uncertainty Estimation

Physically, the initial attitude error incurred at the start of each maneuver persists as an additive constant seen in every centroid taken during the same maneuver (cf., (5.11)). Because of this additive and persistent noise structure, initial attitude errors can be estimated in a separate calculation which is made after the filter has converged. This section will outline the details of this separate attitude calculation. The resulting attitude corrections have been found to be very useful for visualization purposes where they allow one to study science centroiding errors in isolation, i.e., without the derogatory contribution of the star tracker errors.

Estimation of the initial attitude correction  $\psi_j$  for the  $j$ 'th maneuver is motivated by substituting (5.22) into (5.20) to give the relation,

$$\delta\tilde{y}_j = \tilde{H}_j\delta x + \tilde{n}_j\tilde{\nu}_j + \tilde{H}_{\psi,j}\psi_j \quad (6.6)$$

where  $\nu_j$  is the centroid noise having covariance,

$$Cov[\tilde{\nu}_j] \triangleq \tilde{V}_j \quad (6.7)$$

As desired, the attitude correction  $\psi_j$  appears explicitly on the right hand side of (6.6). Assuming the quantity  $\delta x$  is driven to zero by the iterative estimation procedure (this will typically occur to within machine precision), equation (6.6) becomes,

$$\delta\tilde{y}_j \simeq \tilde{H}_{\psi,j}\psi_j + \tilde{n}_j \quad (6.8)$$

This equation can be unit normalized with respect to the measurement noise as,

$$V_j^{-\frac{1}{2}}\delta\tilde{y}_j \simeq V_j^{-\frac{1}{2}}\tilde{H}_{\psi,j}\psi_j + \eta \quad (6.9)$$

where the transformed noise  $\eta$  now has unit covariance. Given values for  $\delta\tilde{y}_j$  and  $\tilde{H}_{\psi,j}$  obtained after filter convergence, equation (6.9) is solved for  $\psi_j$  using least squares (via a QR factorization). This results in a minimum-variance estimate  $\hat{\psi}_j$  which serves as the desired attitude correction for the  $j$ 'th maneuver. This calculation is then repeated for each maneuver  $j = 1, \dots, m_j$ .

## 6.6 Inferred Frames

A subset of important frames are denoted as ‘‘Inferred Frames’’ because their location is inferred by their proximity to a nearby Prime frame. These frames are estimated in a post-processing step, only after the IPF filter has been run, and all essential parameter estimates and covariances are available.

Inferred Frames are defined in terms of their angular offsets  $\Delta w$  and  $\Delta v$  (in oriented angular pixel coordinates) relative to the current Prime frame. In order to facilitate this calculation, the user specifies a list of pixel offsets `DELTA_CW`, `DELTA_CV` in the O-file, which are mapped to the quantities  $\Delta w$  and  $\Delta v$  using a formula given in Section B.4 of Appendix B.

Let a desired Inferred Frame be denoted as  $\widetilde{IPF}$ . It is convenient to define the following direction cosine matrices,

$$T = R_1(\theta_1)R_2(\theta_2)R_3(\theta_3) = \text{TPF to IPF} \quad (6.10)$$

$$\tilde{T} = R_1(\tilde{\theta}_1)R_2(\tilde{\theta}_2)R_3(\tilde{\theta}_3) = \text{TPF to } \widetilde{IPF} \quad (6.11)$$

$$L = \tilde{T}T^T = \text{IPF to } \widetilde{IPF} \quad (6.12)$$

These three mappings are shown pictorially in Figure 6.8 with respect to the frames that they relate. The mapping  $\tilde{T}$  defines the desired Inferred Frame  $\widetilde{IPF}$  relative to the TPF frame. Its specification  $\tilde{T}$  as a function of  $\Delta w$  and  $\Delta v$  is given in the next two subsections.

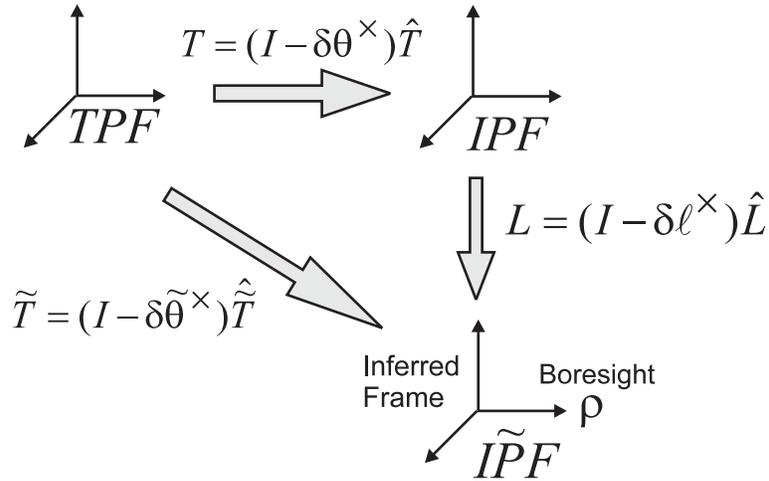


Figure 6.8: Telescope Pointing Frame (TPF), Instrument Pointing Frame (IPF), Inferred Frame  $\widetilde{IPF}$ , and the mappings that relate them

### 6.6.1 Inferred Frame Definition

Let the boresight axis of the Inferred Frame be defined by the vector  $\rho$  (coordinate-free). The resolution of  $\rho$  in IPF is defined by the following construction,

$$s \triangleq \rho \Big|_{IPF} = \frac{\begin{bmatrix} 1 \\ z_v \\ z_w \end{bmatrix}}{1 + z_v^2 + z_w^2} \quad (6.13)$$

where,

$$z \triangleq \begin{bmatrix} z_w \\ z_v \end{bmatrix} = (I + M(y, p_1))y \quad (6.14)$$

$$y = \begin{bmatrix} \Delta w \\ \Delta v \end{bmatrix} \quad (6.15)$$

The boresight vector  $s$  given by (6.13) uniquely determines the Euler angles  $\tilde{\theta}_2$  and  $\tilde{\theta}_3$  of  $\tilde{T}$  in (6.11) (this will be shown explicitly below). The Inferred Frame definition is completed by specifying the last Euler angle as,

$$\tilde{\theta}_1 = \theta_1 \quad (6.16)$$

i.e., it is inherited from the nearby Prime frame.

### 6.6.2 Explicit Expression for $\tilde{T}$

An explicit formula for the Euler angles  $\tilde{\theta}_2, \tilde{\theta}_3$  will now be derived based on the  $\widetilde{IPF}$  boresight vector  $s$  defined in (6.13).

Let the vector  $s$  be resolved in the TPF frame to give the vector  $r$  where,

$$r = \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} = T^T s \quad (6.17)$$

The vector  $r$  can alternatively be calculated by mapping the  $\widetilde{IPF}$  boresight from  $\widetilde{IPF}$  to

TPF as follows,

$$r \stackrel{\Delta}{=} \rho \Big|_{TPF} = \tilde{T}^T \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (6.18)$$

$$= R_3^T(\tilde{\theta}_3)R_2^T(\tilde{\theta}_2)R_1^T(\tilde{\theta}_1) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (6.19)$$

$$= R_3(-\tilde{\theta}_3)R_2(-\tilde{\theta}_2) \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_1 & -s_1 \\ 0 & s_1 & c_1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (6.20)$$

$$= R_3(-\tilde{\theta}_3)R_2(-\tilde{\theta}_2) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (6.21)$$

$$= \begin{bmatrix} c_3 & -s_3 & 0 \\ s_3 & c_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_2 & 0 & s_2 \\ 0 & 1 & 0 \\ -s_2 & 0 & c_2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (6.22)$$

$$= \begin{bmatrix} \cos(\tilde{\theta}_3) \cos(\tilde{\theta}_2) \\ \sin(\tilde{\theta}_3) \cos(\tilde{\theta}_2) \\ -\sin(\tilde{\theta}_2) \end{bmatrix} \quad (6.23)$$

It is noted that equation (6.23) is not a function of  $\tilde{\theta}_1$ . Equating (6.17) and (6.23) gives the relation,

$$\begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} = \begin{bmatrix} \cos(\tilde{\theta}_3) \cos(\tilde{\theta}_2) \\ \sin(\tilde{\theta}_3) \cos(\tilde{\theta}_2) \\ -\sin(\tilde{\theta}_2) \end{bmatrix} \quad (6.24)$$

One can uniquely solve for  $\tilde{\theta}_2, \tilde{\theta}_3$  from the bottom two equalities of (6.24) to give,

$$\tilde{\theta}_2 = \sin^{-1}(-r_z) \quad (6.25)$$

$$\tilde{\theta}_3 = \sin^{-1}\left(\frac{r_y}{\cos(\tilde{\theta}_2)}\right) \quad (6.26)$$

The top equality of (6.24) provides no new information since it is always satisfied when the bottom two equalities are satisfied (i.e., it enforces the unit vector constraint on both sides).

In summary, the Inferred Frame  $\tilde{T}$  is specified by its three Euler angles (6.16)(6.25)(6.26), which in turn can be calculated directly from the angular offsets  $\Delta_w, \Delta_v$  (i.e., via the intermediary vector quantities  $r$  in (6.17) and  $s$  in (6.13)). The angular offsets  $\Delta_w, \Delta_v$ , in turn,

are determined from the user-specified quantities DELTA\_CW, DELTA\_CV in the O-file, using a formula given in Section B.4 of Appendix B.

### 6.6.3 Inferred Frame Estimate $\hat{\tilde{T}}$

Because the true value for the Inferred Frame  $\tilde{T}$  is not known, an estimate  $\hat{\tilde{T}}$  will be constructed in this section.

It is convenient to define the following direction cosine matrix estimates,

$$\hat{T} = R_1(\hat{\theta}_1)R_2(\hat{\theta}_2)R_3(\hat{\theta}_3) = \text{Estimate of } T \quad (6.27)$$

$$\hat{\tilde{T}} = R_1(\hat{\tilde{\theta}}_1)R_2(\hat{\tilde{\theta}}_2)R_3(\hat{\tilde{\theta}}_3) = \text{Estimate of } \tilde{T} \quad (6.28)$$

$$\hat{L} = \hat{\tilde{T}}\hat{T}^T = \text{Estimate of } L \quad (6.29)$$

The quantity  $\hat{T}$  is the Prime Frame estimate which is available from the current IPF Filter run. The quantity  $\hat{\tilde{T}}$  will be constructed in this section by specifying its corresponding Euler angle estimates  $\hat{\tilde{\theta}}_1, \hat{\tilde{\theta}}_2, \hat{\tilde{\theta}}_3$  in (6.28).

Let an estimate of the boresight axis of the Inferred Frame be defined by the vector  $\hat{\rho}$  (coordinate-free). The resolution of  $\hat{\rho}$  in IPF is defined by the following construction,

$$\hat{s} \triangleq \hat{\rho} \Big|_{IPF} = \frac{\begin{bmatrix} 1 \\ \hat{z}_v \\ \hat{z}_w \end{bmatrix}}{1 + \hat{z}_v^2 + \hat{z}_w^2} \quad (6.30)$$

where,

$$\hat{z} \triangleq \begin{bmatrix} \hat{z}_w \\ \hat{z}_v \end{bmatrix} = (I + M(y, \hat{p}_1))y \quad (6.31)$$

$$y = \begin{bmatrix} \Delta w \\ \Delta v \end{bmatrix} \quad (6.32)$$

Note that the estimate  $\hat{s}$  is obtained from its true value  $s$  in (6.13) by simply replacing the optical distortion parameters  $p_1$  by their estimate  $\hat{p}_1$ . The quantity  $y$  remains known exactly since the values for  $\Delta w, \Delta v$  in (6.32) are specified as part of the Inferred Frame definition.

The boresight estimate  $\hat{s}$  given by (6.30) uniquely determines the Euler angle estimates  $\hat{\tilde{\theta}}_2$  and  $\hat{\tilde{\theta}}_3$  of  $\tilde{T}$  (using a similar construction to (6.17)-(6.26) with  $s$  replaced by  $\hat{s}$ ). This will be shown in more detail next.

Resolve the vector  $\hat{s}$  in the TPF frame to give the estimate  $\hat{r}$  where,

$$\hat{r} = \begin{bmatrix} \hat{r}_x \\ \hat{r}_y \\ \hat{r}_z \end{bmatrix} = \hat{T}^T \hat{s} \quad (6.33)$$

Note that  $\hat{r}$  can be calculated since all quantities on the right-hand-side are known or specified.

The vector  $\hat{r}$  can alternatively be calculated by mapping the  $\widetilde{IPF}$  boresight from  $\widetilde{IPF}$  to TPF (using all estimated quantities) as follows,

$$\hat{r} \triangleq \rho \Big|_{TPF} = \hat{T}^T \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos(\hat{\theta}_3) \cos(\hat{\theta}_2) \\ \sin(\hat{\theta}_3) \cos(\hat{\theta}_2) \\ -\sin(\hat{\theta}_2) \end{bmatrix} \quad (6.34)$$

Equating (6.33) and (6.34) gives the relation

$$\begin{bmatrix} \hat{r}_x \\ \hat{r}_y \\ \hat{r}_z \end{bmatrix} = \begin{bmatrix} \cos(\hat{\theta}_3) \cos(\hat{\theta}_2) \\ \sin(\hat{\theta}_3) \cos(\hat{\theta}_2) \\ -\sin(\hat{\theta}_2) \end{bmatrix} \quad (6.35)$$

One can uniquely solve the bottom two equalities of (6.35) simultaneously to give,

$$\hat{\theta}_2 = \sin^{-1}(-\hat{r}_z) \quad (6.36)$$

$$\hat{\theta}_3 = \sin^{-1}\left(\frac{\hat{r}_y}{\cos(\hat{\theta}_2)}\right) \quad (6.37)$$

The Inferred Frame estimate is completed by specifying the last Euler angle estimate as,

$$\hat{\theta}_1 = \hat{\theta}_1 \quad (6.38)$$

i.e., it is inherited from the estimate of the nearby Prime frame.

In summary, the Inferred Frame estimate  $\hat{T}$  is specified by its three Euler angle estimates (6.36)(6.37)(6.38).

It is worth noting that because  $\hat{T}$  is derived by equating (6.33) and (6.34), its defining property is,

$$\hat{T}^T \hat{s} = \hat{T}^T \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (6.39)$$

or equivalently in terms of  $\hat{L}$  (using (6.29)),

$$\hat{L} \hat{s} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (6.40)$$

#### 6.6.4 Inferred Frame Covariance

In this section, an expression is derived for the covariance of the Inferred Frame estimate  $\hat{\tilde{T}}$ .

It is noted that the following relation exists between the direction cosine matrices in (6.10)(6.11)(6.12),

$$\tilde{T} = LT \tag{6.41}$$

It is desired to analyze the relationship between knowledge errors in these quantities. Assuming that the estimates are sufficiently close to their true values, the following small-angle errors  $\delta\tilde{\theta}, \delta\theta, \ell \in R^3$  can be defined,

$$\tilde{T} = (I - \delta\tilde{\theta}^\times)\hat{\tilde{T}} \tag{6.42}$$

$$T = (I - \delta\theta^\times)\hat{T} \tag{6.43}$$

$$L = (I - \ell^\times)\hat{L} \tag{6.44}$$

Substituting (6.42)(6.43)(6.44) into (6.41) and rearranging yields,

$$(I - \delta\tilde{\theta}^\times)\hat{\tilde{T}} = (I - \ell^\times)\hat{L}(I - \delta\theta^\times)\hat{T} \tag{6.45}$$

$$= (I - \ell^\times)(I - (\hat{L}\delta\theta)^\times)\hat{L}\hat{T} \tag{6.46}$$

$$\simeq (I - (\ell + \hat{L}\delta\theta)^\times)\hat{L}\hat{T} \tag{6.47}$$

$$= (I - (\ell + \hat{L}\delta\theta)^\times)\hat{\tilde{T}} \tag{6.48}$$

Here, equation (6.46) follows by the Push-Through Lemma D.14; equation (6.47) follows by dropping the second-order term and rearranging; and (6.48) follows by the definition of  $\hat{L}$  in (6.29).

Multiplying both sides of (6.48) on the right by  $\hat{\tilde{T}}^T$  gives the desired relationship between errors,

$$\delta\tilde{\theta} = \ell + \hat{L}\delta\theta \tag{6.49}$$

Assume that the errors  $\ell$  and  $\delta\theta$  are sufficiently small so that there exists a Jacobian matrix  $J$  such that,

$$\ell + \hat{L}\delta\theta = J\delta x \tag{6.50}$$

where,

$$\delta x = \begin{bmatrix} \delta p_1 \\ \delta p_2 \end{bmatrix} \tag{6.51}$$

It will be shown in the next section that the Jacobian matrix is given by,

$$J = \left[ \begin{array}{c|c} \hline 0, \dots, 0 & \\ \hline \mathcal{S}_{2 \times 3} \hat{L} \begin{bmatrix} 0, \dots, 0 \\ -F \tilde{\mathcal{K}}_1 \end{bmatrix} & \hat{L} \mathcal{S}_\theta \end{array} \right] \quad (6.52)$$

where,

$$\hat{L} = \tilde{T} \hat{T}^T \quad (6.53)$$

$$\mathcal{S}_\theta = [0 | I_{3 \times 3} | 0] \quad (6.54)$$

$$\tilde{\mathcal{K}}_1 = \frac{\partial}{\partial p_1} [-M(y, p_1) y] \quad (6.55)$$

$$\mathcal{S}_{2 \times 3} = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad (6.56)$$

$$F = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (6.57)$$

$$y = \begin{bmatrix} \Delta w \\ \Delta v \end{bmatrix} \quad (6.58)$$

Briefly, the quantity  $\mathcal{S}_\theta$  is a zero-one selection matrix constructed to pick out the entries  $\delta\theta$  from the parameter vector  $\delta p_1$ , i.e.,

$$\delta\theta = \mathcal{S}_\theta \delta p_1 \quad (6.59)$$

The quantity  $\tilde{\mathcal{K}}_1$  is the same sensitivity expression as  $\mathcal{K}_1$  used in the IPF filter derivation, but evaluated on the  $y$  vector defined from the desired offsets (6.58) (here the tilde notation  $\tilde{(\cdot)}$  is used indicate that it is associated with the  $\widetilde{IPF}$  rather than the IPF frame). It is also worth noting that since  $M$  is linear in the  $p_1$  parameters, the sensitivity matrix  $\tilde{\mathcal{K}}_1$  as defined in (6.55) is not a function of  $p_1$ .

Using (6.49) and (6.50), the covariance of the error in the Inferred Frame can be calculated as,

$$Cov[\delta\tilde{\theta}] = J Cov[\delta x] J^T \quad (6.60)$$

In summary, the covariance for the Inferred Frame can be calculated using (6.60), which is expressed in terms of the standard Kalman filter covariance  $P = Cov[\delta x]$  and the Jacobian matrix  $J$ . The expression (6.52) for the Jacobian matrix  $J$  will be derived next.

### 6.6.5 Jacobian Expression

The Jacobian expression (6.52) will be derived in this section. The boresight vector  $\rho$  of the  $\widetilde{IPF}$  frame resolved in (its own)  $\widetilde{IPF}$  frame is simply,

$$\rho \Big|_{\widetilde{IPF}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (6.61)$$

The boresight of the  $\widetilde{IPF}$  frame resolved in the IPF frame is defined by the vector  $s$  as given earlier in (6.13),

$$\rho \Big|_{IPF} = s \quad (6.62)$$

Since the  $L$  matrix maps from the IPF frame to the  $\widetilde{IPF}$  frame, one can write,

$$\rho \Big|_{\widetilde{IPF}} = L \rho \Big|_{IPF} \quad (6.63)$$

Substituting (6.61) and (6.62) into (6.63) gives upon rearranging,

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = Ls \quad (6.64)$$

$$= (I - \ell^\times) \hat{L}(\hat{s} + \delta s) \quad (6.65)$$

$$= \hat{L}\hat{s} + \hat{L}\delta s - \ell^\times \hat{L}\hat{s} - \ell^\times \hat{L}\delta s \quad (6.66)$$

$$\simeq \hat{L}\hat{s} + \hat{L}\delta s - \ell^\times \hat{L}\hat{s} \quad (6.67)$$

Here, equation (6.65) follows by substituting (6.44) and using the relation  $s = \hat{s} + \delta s$ ; equation (6.66) follows by expanding; and the last relation (6.67) follows by dropping a second-order term.

Substituting the defining equation (6.40) into (6.67) gives upon rearranging,

$$\hat{L}\delta s = \ell^\times \hat{L}\hat{s} \quad (6.68)$$

$$= -(\hat{L}\hat{s})^\times \ell \quad (6.69)$$

$$= - \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}^\times \ell \quad (6.70)$$

$$= - \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \ell \quad (6.71)$$

Let the components of vector  $\ell$  be denoted as,

$$\ell = \begin{bmatrix} \ell_1 \\ \ell_2 \\ \ell_3 \end{bmatrix} \quad (6.72)$$

The first component  $\ell_1$  is not constrained by (6.71) and will be arbitrarily set to zero, i.e.,

$$\ell_1 = 0 \quad (6.73)$$

The lower two equations of (6.71) can be solved simultaneously to give,

$$\begin{bmatrix} \ell_2 \\ \ell_3 \end{bmatrix} = \mathcal{S}_{2 \times 3} \hat{L} \delta s \quad (6.74)$$

where,

$$\mathcal{S}_{2 \times 3} = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad (6.75)$$

Consider an approximation to  $s$  in (6.13) given by,

$$s \simeq \begin{bmatrix} 1 \\ z_v \\ z_w \end{bmatrix} \quad (6.76)$$

where,

$$\begin{bmatrix} z_v \\ z_w \end{bmatrix} = F \begin{bmatrix} z_w \\ z_v \end{bmatrix} = F(y - \tilde{\mathcal{K}}_1 p_1) \quad (6.77)$$

Moreover, consider an approximation to  $\hat{s}$  in (6.30) given by,

$$\hat{s} \simeq \begin{bmatrix} 1 \\ \hat{z}_v \\ \hat{z}_w \end{bmatrix} \quad (6.78)$$

where,

$$\begin{bmatrix} \hat{z}_v \\ \hat{z}_w \end{bmatrix} = F \begin{bmatrix} \hat{z}_w \\ \hat{z}_v \end{bmatrix} = F(y - \tilde{\mathcal{K}}_1 \hat{p}_1) \quad (6.79)$$

Then the error in  $\delta s = s - \hat{s}$  can be approximated by subtracting (6.78) from (6.76) to give,

$$\delta s = s - \hat{s} \simeq \begin{bmatrix} 1 - 1 \\ z_v - \hat{z}_v \\ z_w - \hat{z}_w \end{bmatrix} = \begin{bmatrix} 0 \\ -F \tilde{\mathcal{K}}_1 \delta p_1 \end{bmatrix} \quad (6.80)$$

Substituting (6.80) into (6.74) gives,

$$\begin{bmatrix} \ell_2 \\ \ell_3 \end{bmatrix} = \mathcal{S}_{2 \times 3} \hat{L} \begin{bmatrix} 0 \\ -F \tilde{\mathcal{K}}_1 \delta p_1 \end{bmatrix} \quad (6.81)$$

Combining (6.81)(6.73) with (6.59) gives upon rearranging,

$$\ell + \hat{L}\delta\theta = J \begin{bmatrix} \delta p_1 \\ \delta p_2 \end{bmatrix} \quad (6.82)$$

where the Jacobian  $J$  has the form (6.52) as desired.

## 6.7 Special Cases: Partial Centroid Measurements

### 6.7.1 Overview of Slit Mode

The two-component centroid measurement  $y = [y_w, y_v]^T$  is the primary observable used by the IPF filter for calibration purposes. Until now, it has been assumed that the  $y_w$  value is provided simultaneously with the  $y_v$  value to make a complete centroid value  $y$ . The full centroid is then used to linearize the measurement equations.

Unfortunately, slit type instruments often cannot provide a full centroid. This is because centroids must be manufactured artificially by scanning a source across the entrance aperture, and then reporting the centroid to be at the center of the slit at the time instant of the peak response. This defines a useful centroid value in the  $v$  (dispersion) direction, but there may be little or no information in the  $w$  direction. Conversely, when scanning across the length of the slit, one may get a useful centroid value in the  $w$  direction, but little or no information in the  $v$  direction. In this way, a slit type instrument may only provide “partial” centroids  $y$ , where  $y$  contains either a  $y_v$  component or  $y_w$  component, but not both.

Fortunately, sandwich maneuvers for slits are typically designed to involve slit crossings in the dispersion direction at two or three separate locations, and then a crossing along the slit length as shown in Figure 6.9. Clearly, the full calibration information is available over the course of the such an experiment. However, the IPF sensitivity equations must be linearized slightly differently than done previously due to the fact that the information in the  $w$  and  $v$  directions comes in at different times.

A special **slit mode** was developed for the IPF filter to allow proper linearization of the measurement equation using only partial centroids. It will be shown that the slit mode must linearize about the inverse map of the optical distortions to avoid requiring a full centroid. Other than the calculation of the sensitivities and predicts, the filter operation is very similar to the standard IPF mode. In order to simplify the filter operations, an autonomous slit mode detection is built into the IPF code. If the calibrating instrument is of the slit type, the IPF filter autonomously detects this situation from the NF number, and enables slit mode operations. The sensitivities are then calculated in this special way, even if the user provides full centroids.

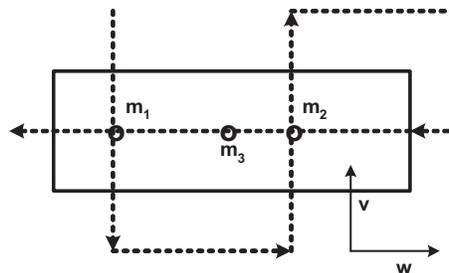


Figure 6.9: Survey Maneuver Through Slit Instrument - Partial Centroid

### 6.7.2 Sensitivities for Slit Mode

**Standard Mode:** Recall from the filter formulation section that the observation equation was defined as,

$$(I + M(p_1, y)) y = z(p_2) \quad (6.83)$$

The measurement equation for the standard approach was defined by rearranging this equation into the following form,

$$y = -M(p_1, y) y + z(p_2) \quad (6.84)$$

Note that the full centroid value  $y$  is required to evaluate the right-hand side of this equation. Furthermore, taking the sensitivities with respect to  $p_1$  and  $p_2$  gives,

$$\frac{\partial}{\partial p_1} (-M(p_1, y) y + z(p_2)) = \frac{\partial}{\partial p_1} (-M(p_1, y) y) \triangleq \mathcal{K}_1 \quad (6.85)$$

$$\frac{\partial}{\partial(\delta p_2)} (-M(p_1, y) y + z(p_2)) = \frac{\partial}{\partial(\delta p_2)} (z(p_2)) \triangleq \mathcal{K}_2. \quad (6.86)$$

Here again, the full centroid  $y$  is required to evaluate the  $p_1$  sensitivity.

**Slit Mode:** For slit-type instruments, it is assumed that the linear plate scale parameters are not estimated. With this simplification, the observation equation (6.83) can be rearranged as,

$$y = (I + M(p_1))^{-1} z(p_2). \quad (6.87)$$

This form has the advantage that the right-hand side does not involve the centroid  $y$  (either full or partial). Furthermore, the top part of this equation serves as a separate measurement equation when  $y_w$  is observed, and the bottom part serves as a measurement equation when  $y_v$  is observed. By using either the top or bottom (for partial centroids) or both (for full centroids) a very general formulation of the measurement update can be made which accomodates either partial or full centroids.

The calculation of the  $p_2$  sensitivity is straightforward,

$$\begin{aligned} \frac{\partial}{\partial(\delta p_2)} ((I + M(p_1))^{-1} z(p_2)) &= (I + M(p_1))^{-1} \frac{\partial}{\partial(\delta p_2)} (z(p_2)) \\ &= (I + M(p_1))^{-1} \mathcal{K}_2 \end{aligned}$$

The calculation of the  $p_1$  sensitivity is more complicated because of the matrix inverse. For this purpose, consider the result from Lemma D.19,

$$\frac{\partial}{\partial p} (A^{-1}) = -A^{-1} \frac{\partial A}{\partial p} A^{-1} \quad (6.88)$$

Then,

$$\begin{aligned}
\frac{\partial}{\partial p_1} ((I + M(p_1))^{-1} z(p_2)) &= -(I + M(p_1))^{-1} \left[ \frac{\partial}{\partial p_1} (I + M(p_1)) \right] (I + M(p_1))^{-1} z(p_2) \\
&= (I + M(p_1))^{-1} \left[ \frac{\partial}{\partial p_1} (-M(p_1)) \right] (I + M(p_1))^{-1} z(p_2) \\
&= (I + M(p_1))^{-1} \mathcal{K}_1 \left\{ (I + M(p_1))^{-1} z(p_2) \right\} \tag{6.89}
\end{aligned}$$

where we have generalized the definition of  $\mathcal{K}_1$  slightly to be a function of an arbitrary vector  $\ell$ , i.e.,

$$\mathcal{K}_1 \{ \ell \} \triangleq \frac{\partial}{\partial p_1} (-M(p_1, y) \ell) \tag{6.90}$$

## 6.8 Special Cases: IPF Multi-Run Tool

In some cases, the number of sandwich maneuvers required for fine calibration of a particular instrument was sufficiently large that it had to be spread across more than one in-flight observing session. In such cases, a capability was needed to combine IPF estimates produced on data sets taken from non-contiguous portions of the mission time-line (typically different days of the mission). The IPF Multi-Run tool was developed to address this need.

The basic idea behind the IPF Multi-Run tool is to combine the results from different IPF runs to form an overall minimum-variance estimate. If the problem were linear, this would involve a straightforward weighted average of the individual estimates. However, because the problem is nonlinear, the problem must be carefully linearized first and the weighted average taken with respect to perturbations expressed relative to a common nominal parameter estimate. Details are given below.

The  $i$ 'th IPF run will produce a state estimate of the form,

$$x_f(i) \triangleq \begin{bmatrix} p_1(i) \\ p_{2f}(i) \end{bmatrix} \in \mathfrak{R}^{37} \quad (6.91)$$

Define a subset of parameters  $\xi_f(i) \subset x_f(i)$  which do not vary from day to day (or observing session to observing session). For example, any or all of the parameters in vector  $p_1$  can be included in  $\xi_f(i)$  since the plate scales and optical distortion parameters are not expected to change with time because the focal plane is actively cooled. Likewise, the IPF frame alignment and scan mirror alignment and scale factor parameters contained in  $p_{2f}$  are not expected to change with time, and their estimates from different days can be combined to advantage. Note that the rest of the  $p_{2f}$  parameters corresponding to thermal distortion parameters and gyro drift biases are not good candidates for including in  $\xi_f(i)$  because of their expected day to day variations.

The main idea for data fusion is to expand all variables  $\xi_f(i)$  about a common nominal  $\xi_f^\circ$  to give the perturbation vector  $\delta\xi(i)$ , and then take a weighted average of the perturbations  $\delta\xi(i)$ . For simplicity, the common nominal will be chosen as the parameter estimate from the very first observing session (i.e.,  $\xi_f^\circ = \xi_f(1)$ ).

If the variables are unconstrained, the perturbation  $\delta\xi(i)$  is defined simply from,

$$\xi_f(i) = \xi_f^\circ + \delta\xi(i) \quad (6.92)$$

If the variables are constrained, the perturbation  $\delta\xi(i)$  consists of only the independent parameters (generally of lower dimension), defined by the expansion,

$$\xi_f(i) = \xi_f^\circ + \mathcal{K}\delta\xi(i) + \mathcal{O}(\|\delta\xi(i)\|^2) \quad (6.93)$$

$$\mathcal{K} \triangleq \left. \frac{\partial \xi_f}{\partial (\delta\xi)} \right|_{\xi_f^\circ} \quad (6.94)$$

By defining the perturbations in this way, the estimates of the mean and square-root covariance from  $n$  different IPF runs can be summarized as,

$$\delta\xi(i), \quad P_{\xi\xi}(i)^{\frac{1}{2}} \quad i = 1, \dots, n \quad (6.95)$$

As an example, the scan mirror alignment estimate  $a(i)$  for the  $i$ 'th observing session can be written in the form (6.93) as follows,

$$a(i) = a^\circ + h_a(a^\circ)\delta\alpha(i) \quad (6.96)$$

where,

$$h_a(a^\circ) = \begin{bmatrix} 0 \\ a_z^\circ \\ -a_y^\circ \end{bmatrix}; \quad a^\circ = \begin{bmatrix} a_x^\circ \\ a_y^\circ \\ a_z^\circ \end{bmatrix}; \quad a(i) = \begin{bmatrix} a_x(i) \\ a_y(i) \\ a_z(i) \end{bmatrix} \quad (6.97)$$

The perturbation  $\delta\alpha(i)$  can then be extracted from (6.96) to give,

$$\delta\alpha(i) = \frac{a_y(i) - a_y^\circ}{a_z^\circ} \quad (6.98)$$

For the IPF frame quaternion estimate  $q_T(i)$  from the  $i$ 'th observing session, the quaternion difference is formed with respect to the nominal quaternion  $q_T^\circ$  to obtain the small angle quaternion  $dq_T(i)$  as,

$$dq_T(i) = q_T(i) \otimes (q_T^\circ)^* \quad (6.99)$$

The perturbation  $\delta\theta(i)$  can then be extracted from (6.99) to give,

$$\delta\theta(i) = \begin{bmatrix} \delta\theta_1(i) \\ \delta\theta_2(i) \\ \delta\theta_3(i) \end{bmatrix} = 2 \begin{bmatrix} dq_{T1}(i) \\ dq_{T2}(i) \\ dq_{T3}(i) \end{bmatrix} \quad (6.100)$$

With these difference operations, the multiple day estimates can be written in terms of the observation model,

$$\delta\xi(i) = \delta\xi + \eta(i) \quad (6.101)$$

where  $\delta\xi$  is the truth perturbation parameter vector and the noise  $\eta(i)$  is defined as,

$$\eta(i) \sim N(0, P_{\xi\xi}(i)) \quad (6.102)$$

The equation (6.101) can be stacked to form an observation equation of the form:

$$\begin{bmatrix} \delta\xi(1) \\ \vdots \\ \delta\xi(n) \end{bmatrix} = \begin{bmatrix} I \\ \vdots \\ I \end{bmatrix} \delta\xi + \begin{bmatrix} \eta(1) \\ \vdots \\ \eta(n) \end{bmatrix} \quad (6.103)$$

Assuming independence of the noise from day to day, equation (6.103) is left-multipled by the inverse square-root noise covariance (scaling with their respective square-root covariance) to obtain a linearized observation equation with unit noise intensity,

$$\begin{bmatrix} P_{\xi\xi}(1)^{-1/2} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & P_{\xi\xi}(n)^{-1/2} \end{bmatrix} \begin{bmatrix} \delta\xi(1) \\ \vdots \\ \delta\xi(n) \end{bmatrix} = \begin{bmatrix} P_{\xi\xi}(1)^{-1/2} \\ \vdots \\ P_{\xi\xi}(n)^{-1/2} \end{bmatrix} \delta\xi + \begin{bmatrix} \tilde{\eta}(1) \\ \vdots \\ \tilde{\eta}(n) \end{bmatrix} \quad (6.104)$$

where,

$$\tilde{\eta}(i) = N(0, I) \quad (6.105)$$

The resulting observation equation (6.104) is in the general regression form,

$$y = H\delta\xi + \tilde{\eta} \quad (6.106)$$

This equation can be solved for  $\delta\xi$  using a *QR* factorization method which is equivalent to the least squares estimate from solving the standard formula,

$$\delta\hat{\xi} = (H^T H)^{-1} H^T y \quad (6.107)$$

The smoothed perturbation  $\delta\hat{\xi}$  is then used to compute the full Multi-Run parameter estimate  $\hat{\xi}_f$  as,

$$\hat{\xi}_f = \xi_f^o + \mathcal{K}\delta\hat{\xi}(i) \quad (6.108)$$

where  $\mathcal{K}$  has been defined earlier in (6.93).

## 7 IPF SOFTWARE VERIFICATION

The IPF Kalman filter has been tested in the IPF Filter Unit Test Environment (FLUTE) for verification of its functionality and performance. FLUTE was developed to create simulated input data files for unit-testing the IPF filter. FLUTE was developed in MATLAB environment with complete command capability to simulate realistic attitude and centroid data. The FLUTE test environment captures all realistic uncertainties, noise sources and parameter errors. With the FLUTE tool, the IPF Kalman filter has been systematically tested for the product verification.

### 7.1 Software Verification Procedure

The FLUTE - IPF interfaces are shown in Figure 7.1. Given a desired survey maneuver, FLUTE produces the appropriate command history. With this command history, the true spacecraft attitude motions are produced. Appropriate truth alignments, thermal drift, disturbances and sensor noises are added to the attitude data. The produced attitude sensor data and instrument centroid data are formatted to produce a mock data set which is representative of an actual Spitzer flight data set.

For each of these FLUTE unit test data sets, the IPF filter was executed under simulated flight data conditions. The corresponding analysis charts were obtained as well as the LG log and IF output files. In addition to the nominal filter execution, the truth data from the FLUTE environment were used to obtain the absolute error of the filter in order to troubleshoot and measure filter performance. Given these results, post execution analysis were performed to identify any unexpected results and obtain physical interpretation of the results.

### 7.2 Description of Test Cases

The IPF Kalman filter has been completed and tested successfully using the FLUTE test data. The tests consisted of four benchmark examples:

1. The IRAC 3.6  $\mu\text{m}$  array for which the experiment design consists of five-of diamonds pattern with dithering
2. The IRS Red Pickup Array for which the experiment design consists of a 3 by 3 grid pattern with dithering
3. The IRS Short-Lo slit for which a horizontal and vertical slit crossing pattern is used
4. The MIPS 24  $\mu\text{m}$  Array using a compound 7x3 grid with gyro offsets and mirror motion.

For all four test cases, the resulting IPF Frame estimates look reasonable, and are accurate to a fraction of an arcsec. For all cases, the actual truth error on state estimates were within reasonable bounds of the final square root covariances.

The detailed test results can be found in D-document *SIRTF Instrument Pointing Frame (IPF) Kalman Filter Unit Test Report* [2].

CASE	INSTRUMENT	DESCRIPTION
CASE 1	IRAC 3.6 $\mu$ m array	Five-of diamonds pattern, includes dithering
CASE 2	IRS Red Peakup Array	3x3 grid pattern
CASE 3	IRS Short-Lo slit	Horizontal and vertical slit crossing pattern
CASE 4	MIPS 24 $\mu$ m	Compound 7x3 grid with gyro offsets and mirror motion

Table 7.18: FLUTE TEST CASES

### IPF Filter and Test Environment

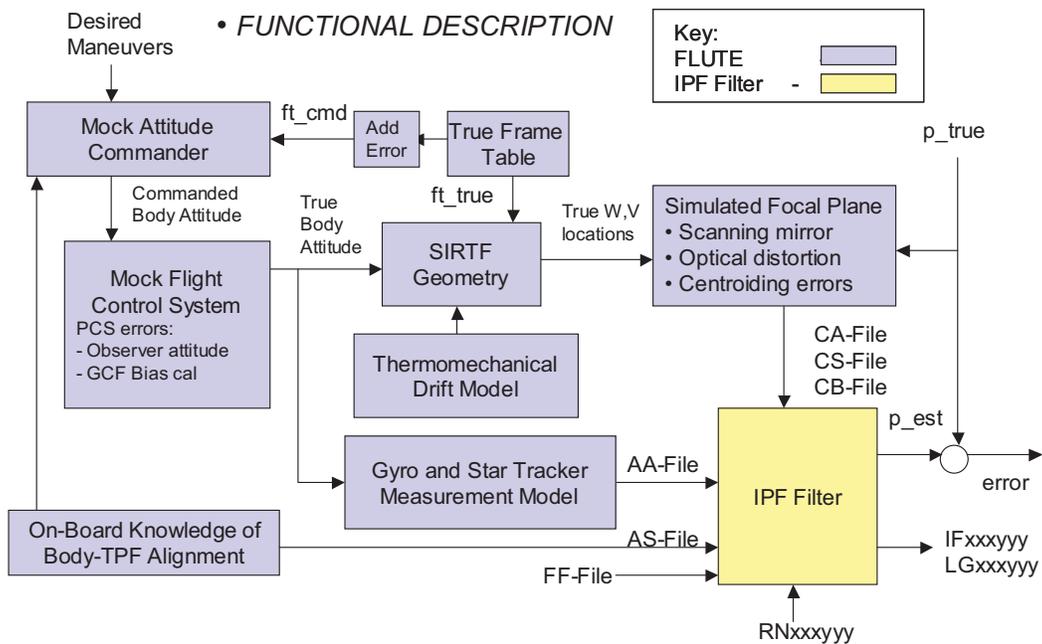


Figure 7.1: IPF Filter and Test Environment

## 8 EXAMPLE FILTER RUN: MIPS

The Focal Plane Survey of the instrument: MIPS\_24um\_center (95), is selected as an example of the IPF Kalman filter execution using the FLUTE generated data.

### 8.1 Summary of Results

**FOCAL PLANE SURVEY ANALYSIS: Covariance Analysis.**

**INSTRUMENT NAME: MIPS\_24um\_center NF: 95**

PIX2RADW: 1.20874169E-005 [rad/pixel] = 2.4932E+000 [arcsec/pixel]

PIX2RADV: 1.20874169E-005 [rad/pixel] = 2.4932E+000 [arcsec/pixel]

FRAME	DESCRIPTION	IPF <sup>6</sup>	SF <sup>7</sup>	ARW <sup>8</sup>	TOTAL	REQ
095(P)	MIPS_24um_center	0.0548	0.0855	0.0703	0.1235	0.14
096(I)	MIPS_24um_plusY_edge	0.0973	0.0855	0.0703	0.1473	N/A
099(I)	MIPS_24um_small_FOV1	0.0547	0.0855	0.0703	0.1234	N/A
100(I)	MIPS_24um_small_FOV2	0.0525	0.0855	0.0703	0.1225	N/A
103(I)	MIPS_24um_large_FOV1	0.0549	0.0855	0.0703	0.1235	N/A
104(I)	MIPS_24um_large_FOV2	0.0547	0.0855	0.0703	0.1234	N/A
TR (C)	CORNERS OF INSTRUMENT	0.1210	0.0855	0.0703	0.1640	N/A
BR (C)	CORNERS OF INSTRUMENT	0.1166	0.0855	0.0703	0.1608	N/A
TL (C)	CORNERS OF INSTRUMENT	0.1127	0.0855	0.0703	0.1579	N/A
BL (C)	CORNERS OF INSTRUMENT	0.1193	0.0855	0.0703	0.1627	N/A

Table 8.19: IPF calibration error summary ([arcsec], 1-sigma, radial)

RMS METRIC	A PRIORI <sup>9</sup>	A POSTERIORI <sup>4</sup>	ATT. CORRECTED <sup>10</sup>	UNITS
Radial	13.7662	1.2918	0.3992	arcsec
Radial	5.5215	0.5181	0.1601	pixels
W-Axis	3.9682	0.3399	0.1131	pixels
V-Axis	3.8393	0.3911	0.1133	pixels

Table 8.20: Measurement prediction error summary (1-sigma)

<sup>6</sup>IPF filter removes systematic pointing errors due to: thermomechanical alignment drift (Body to TPF), gyro bias and bias drift, centroiding error, attitude error, and optical distortion. IPF SIGMA presented here is "NOT Scaled" by the Least Squares Scale factor. The Least Squares Scale Factor was: 1.125028.

<sup>7</sup>Error due to gyro Scale Factor: assumes 95 ppm error over 0.250 degree maneuver.

<sup>8</sup>Error due to gyro Angle Random Walk: assumes  $ARW = 100 \mu\text{deg}/\sqrt{hr}$ , with 960 second Maneuver time (max), and 14 independent Maneuvers.

<sup>9</sup>This can be interpreted as estimate of "pixel to sky" pointing reconstruction error if no science data is used.

<sup>10</sup>This can be interpreted as estimate of achieved S/I centroiding error

IPF BROWN ANGLE SUMMARY						
	WAS			IS		
Frame Number	theta_Y (arcmin)	theta_Z (arcmin)	angle (deg)	theta_Y (arcmin)	theta_Z (arcmin)	angle (deg)
095	+6.641000	+3.931000	+0.000000	+6.641111	+3.932233	+0.016691
096	+6.595000	+6.712000	-0.000000	+6.618045	+6.975361	+0.016691
099	+7.687000	+3.943000	-0.000000	+7.784631	+3.921390	+0.016691
100	+5.574000	+3.918000	+0.000000	+5.475980	+3.941639	+0.016691
103	+6.746000	+3.932000	+0.000000	+6.755408	+3.931220	+0.016691
104	+6.558000	+3.930000	+0.000000	+6.549682	+3.933031	+0.016691
001	+0.000000	+0.000000	+0.000000	+9.608774	+0.843807	+0.016691
001	+0.000000	+0.000000	+0.000000	+9.544149	+6.948933	+0.016691
001	+0.000000	+0.000000	+0.000000	+3.749581	+0.904283	+0.016691
001	+0.000000	+0.000000	+0.000000	+3.699961	+6.991274	+0.016691

Table 8.21: IPF Brown angle summary

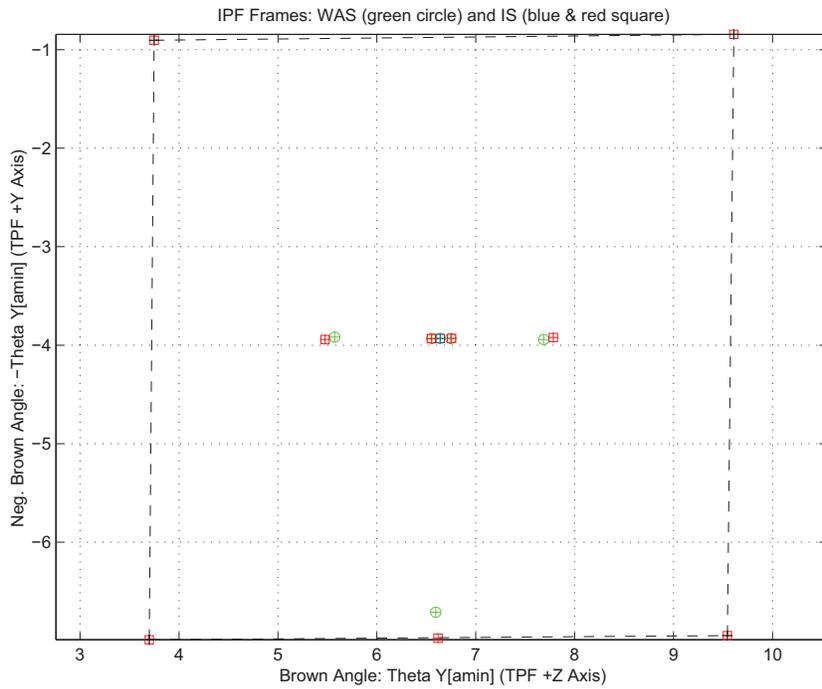


Figure 8.1: A-priori and a-posteriori IPF frames (ZOOMED)

## 8.2 Output Figures

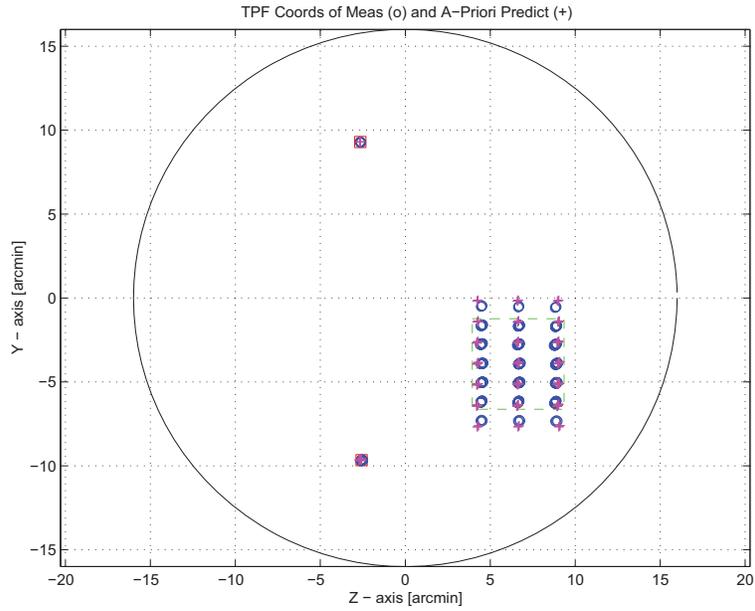


Figure 8.2: TPF coords of measurements and a-priori predicts

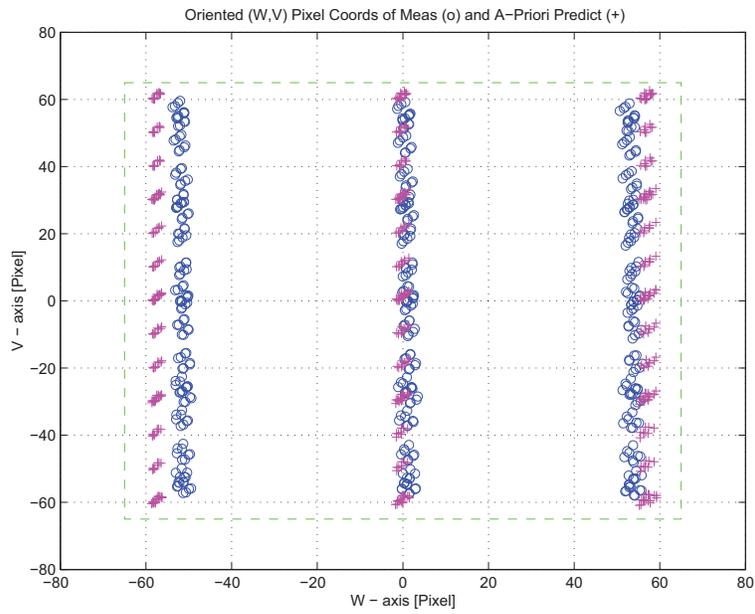


Figure 8.3: Oriented Pixel Coords of measurements and a-priori predicts

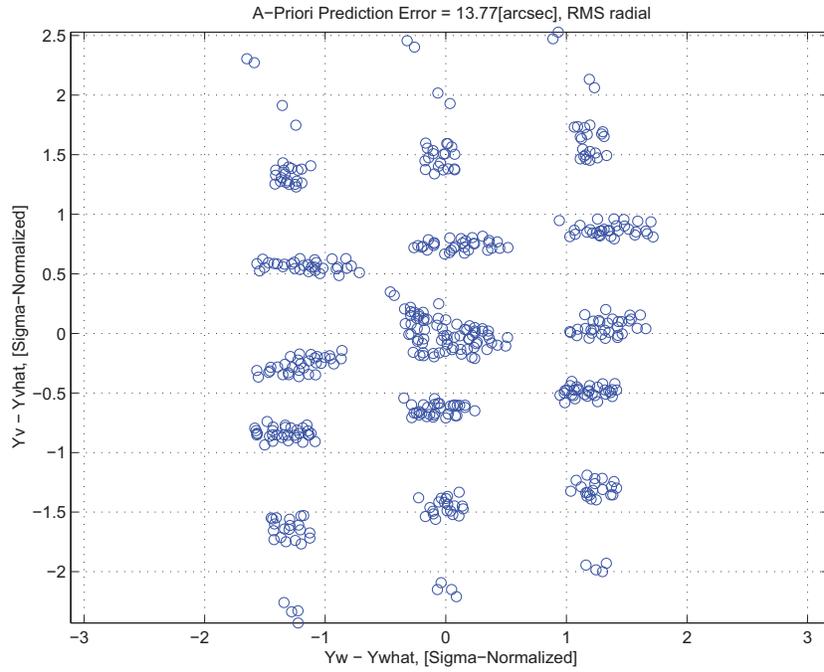


Figure 8.4: A-priori prediction error

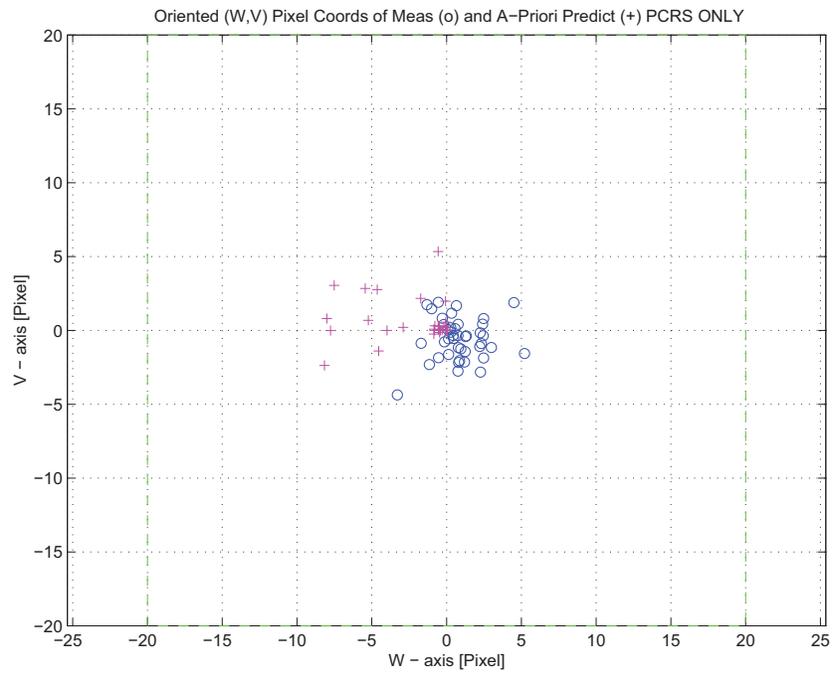


Figure 8.5: Oriented Pixel Coords of measurements and a-priori predicts (PCRS only)

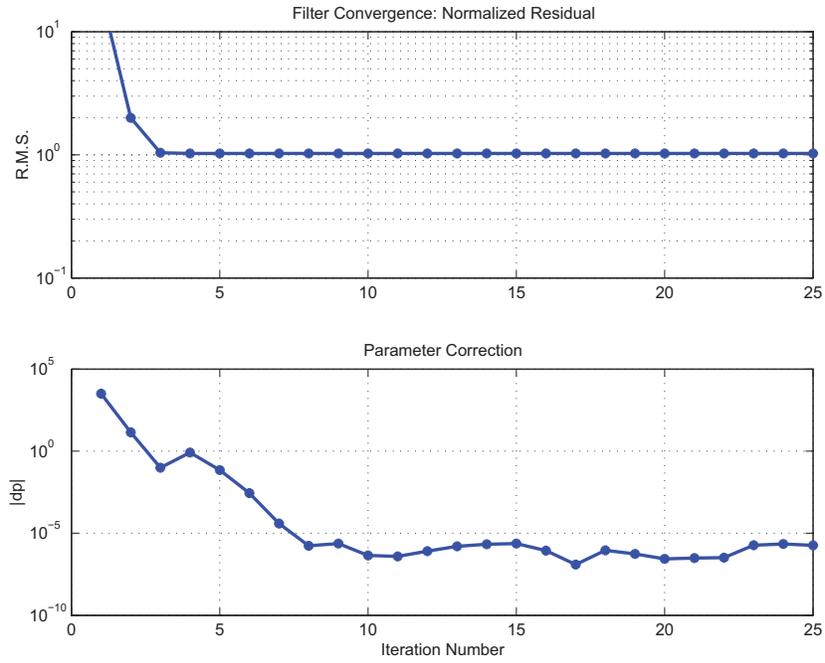


Figure 8.6: IPF execution convergence, chart 1

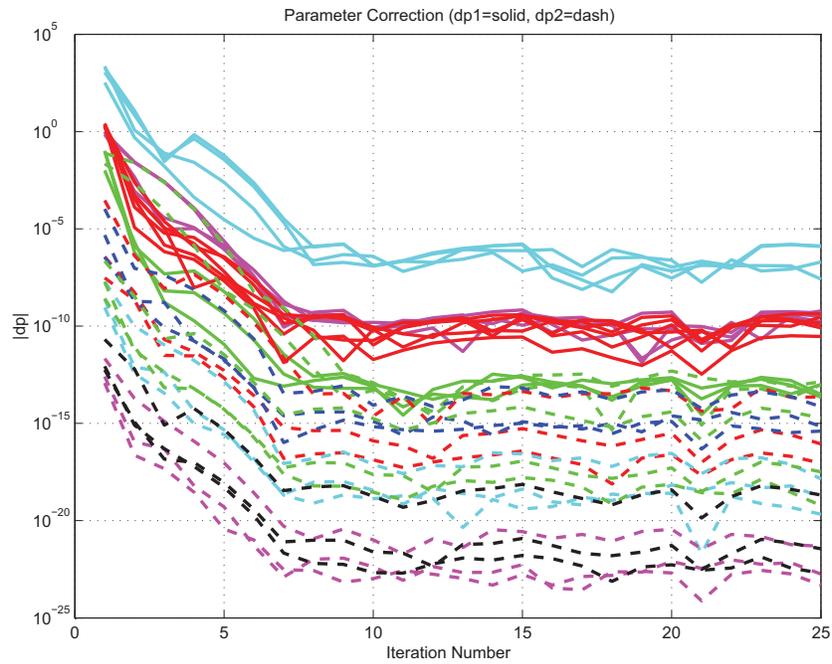


Figure 8.7: IPF execution convergence, chart 2

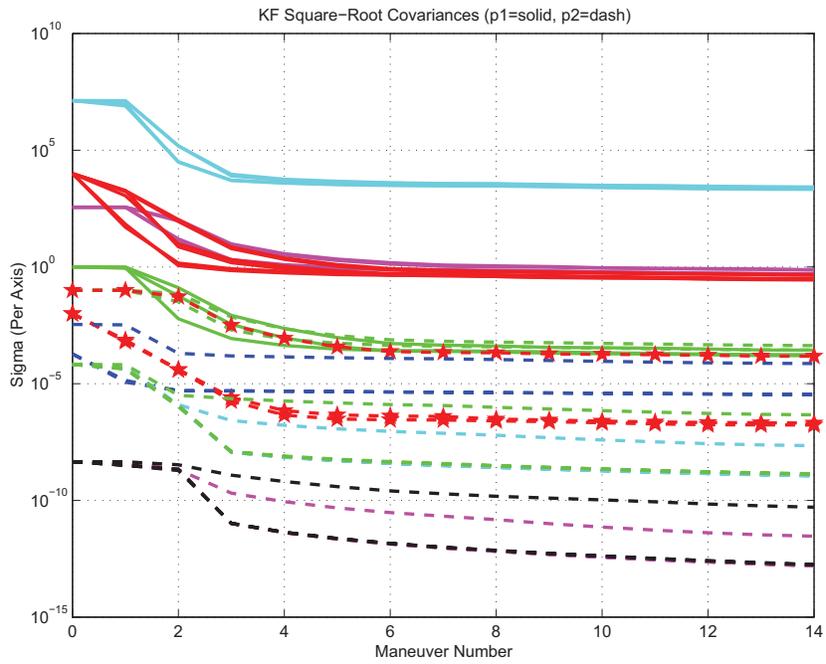
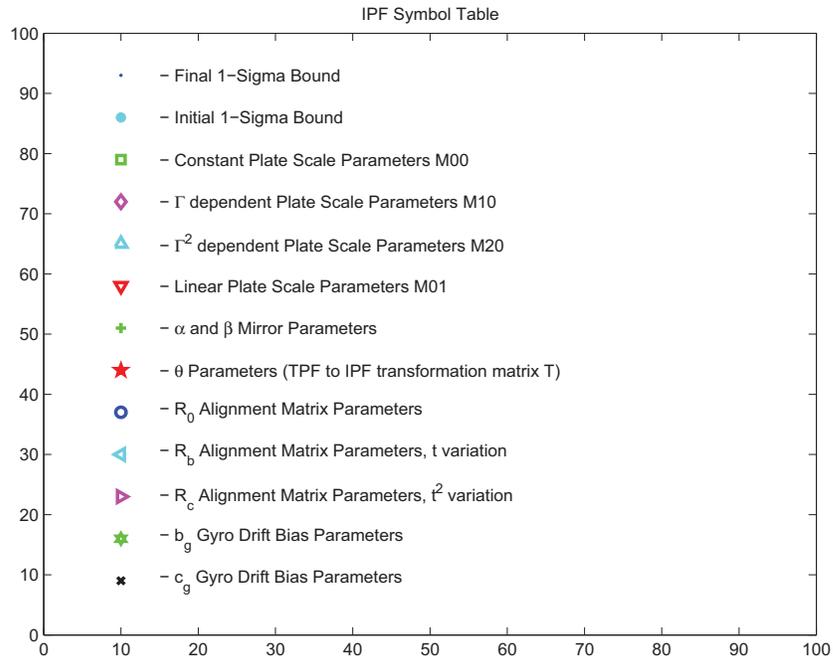


Figure 8.8: Parameter uncertainty convergence



b!]

Figure 8.9: IPF parameter symbol table

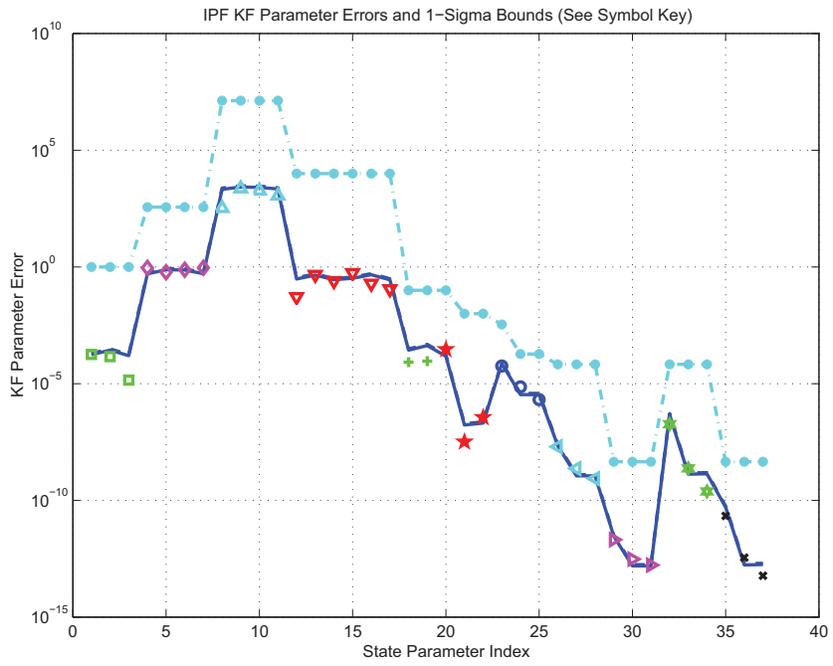


Figure 8.10: KF parameter error sigma plots

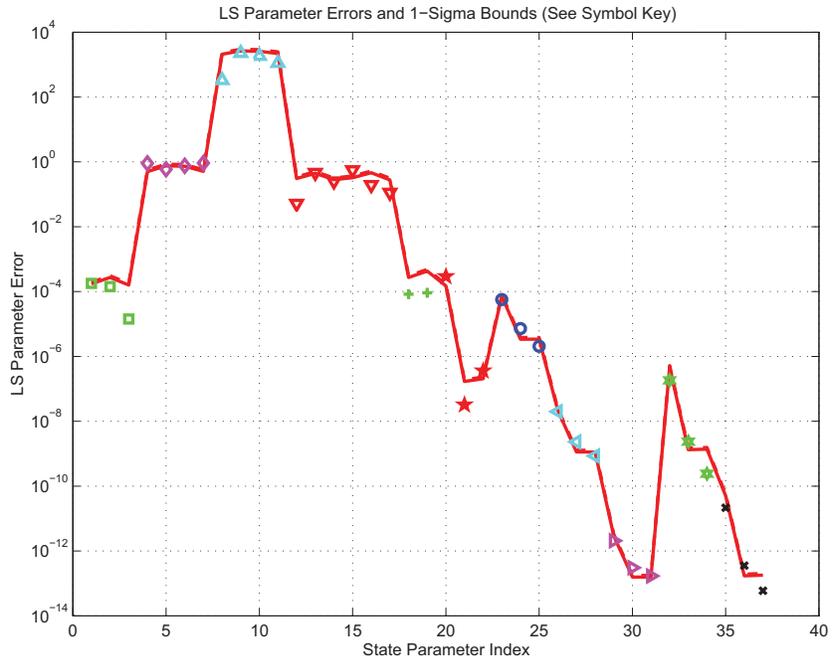


Figure 8.11: LS parameter error sigma plot

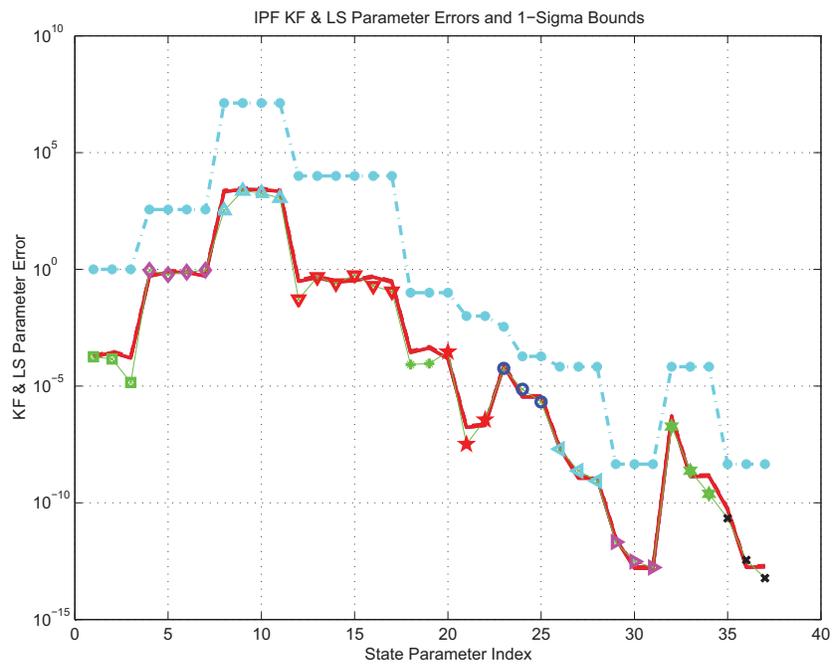


Figure 8.12: KF and LS parameter error sigma plot

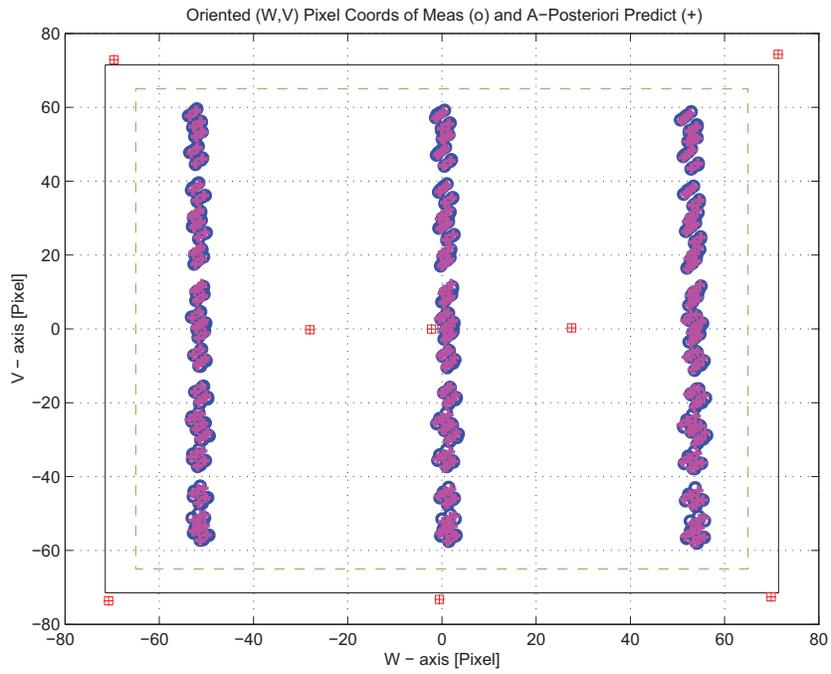


Figure 8.13: Oriented Pixel Coords of meas. and a-posteriori predicts

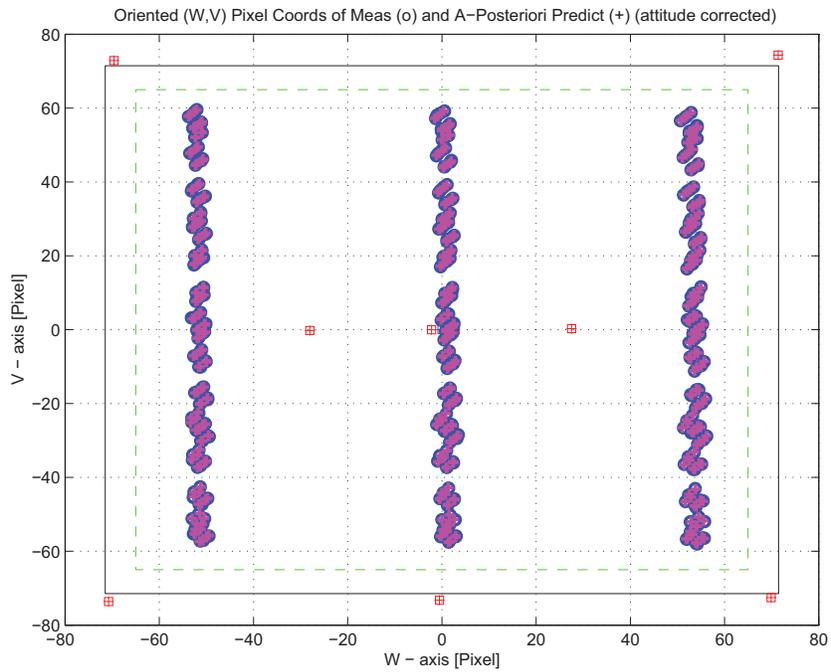


Figure 8.14: Oriented Pixel Coords of meas. and a-posteriori predicts (attitude corrected)

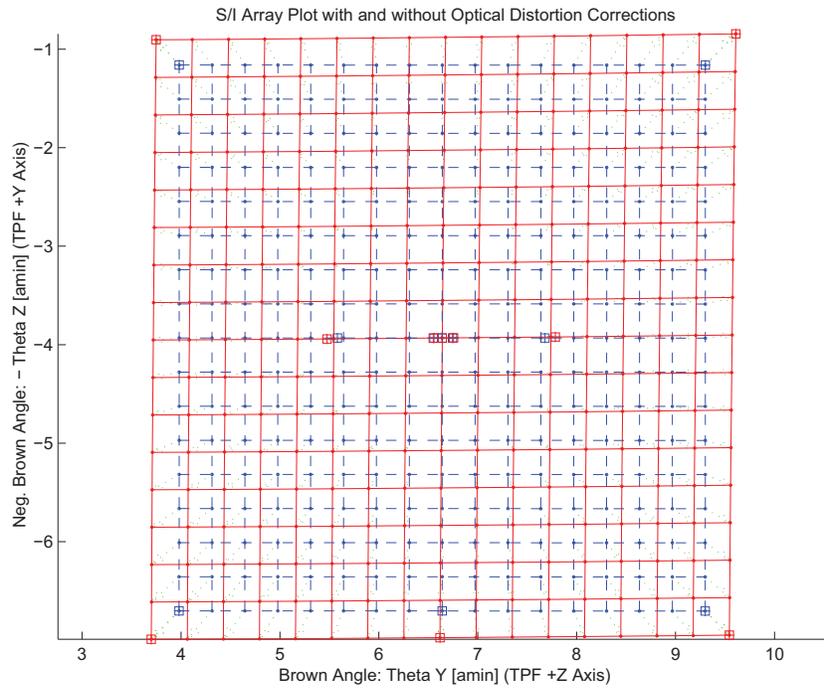


Figure 8.15: Array plot with (solid) and w/o (dashed) optical distortion corrections

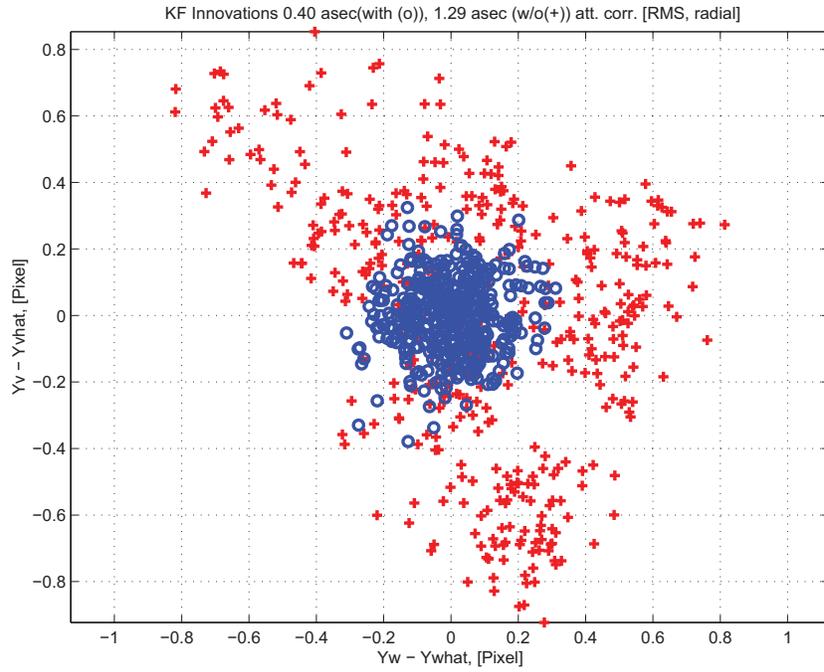


Figure 8.16: KF innovations with (o) and w/o (+) attitude corrections

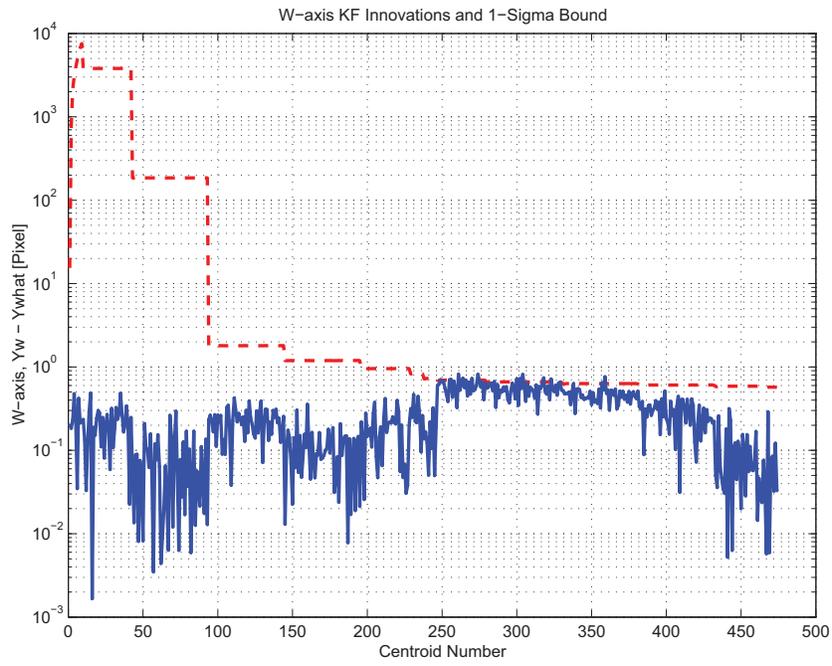


Figure 8.17: W-axis KF innovations and 1-sigma bound

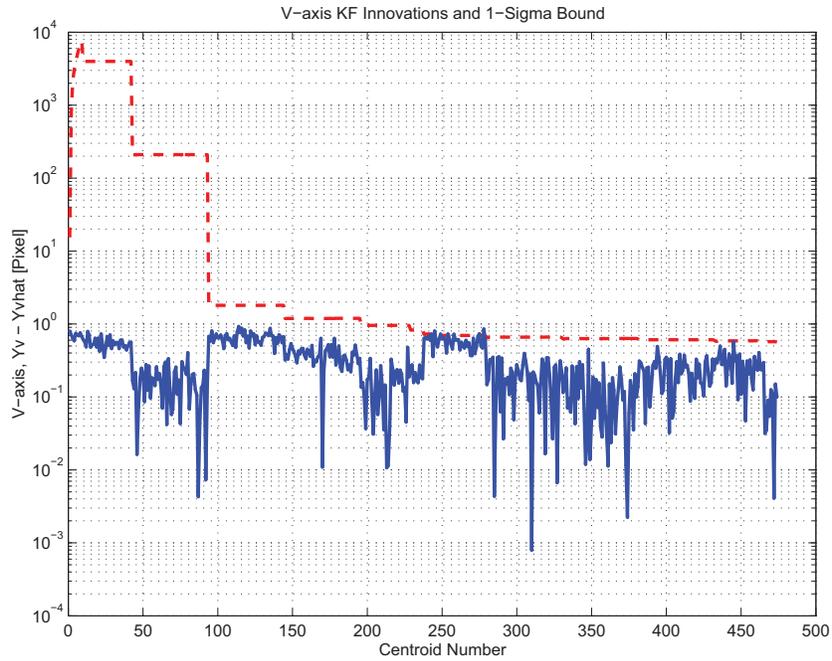


Figure 8.18: V-axis KF innovations and 1-sigma bound

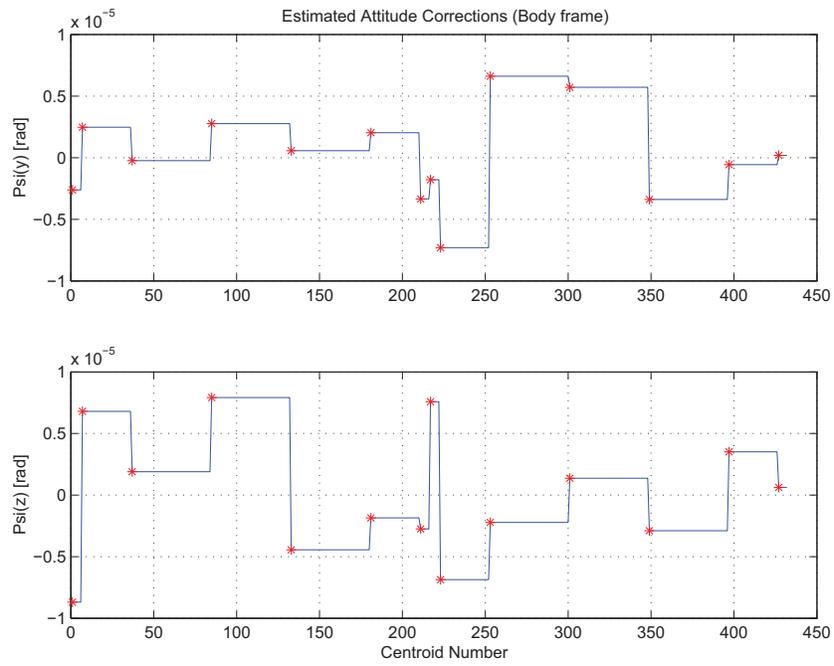


Figure 8.19: Estimated attitude corrections (Body frame)

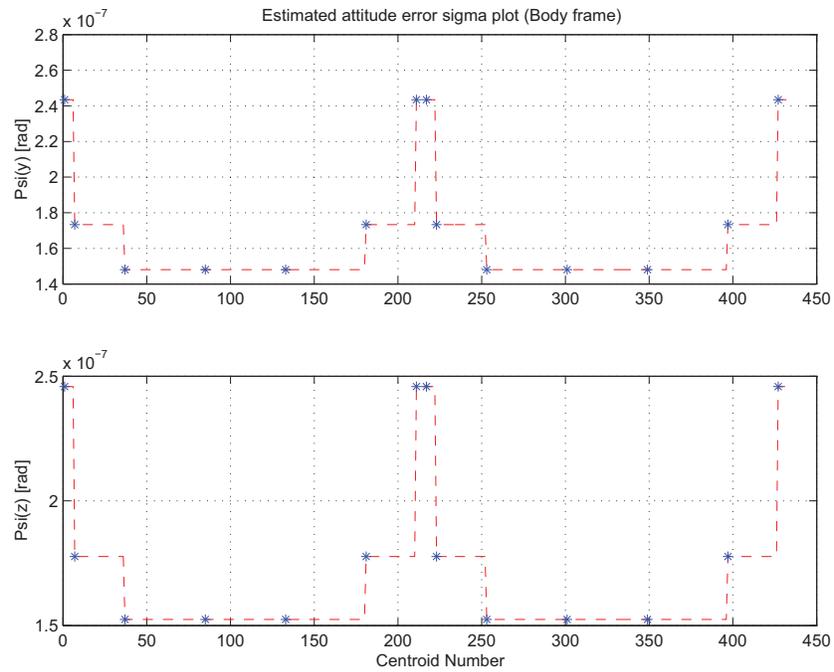


Figure 8.20: Estimated attitude error sigma plot (Body frame)

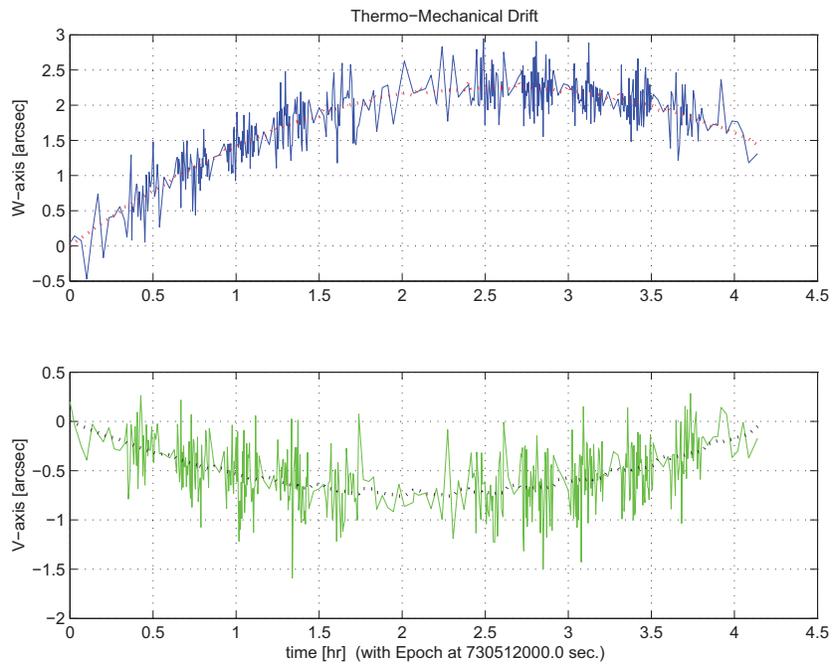


Figure 8.21: Thermo-mechanical boresight drift (equiv. angle in (W,V) coords)

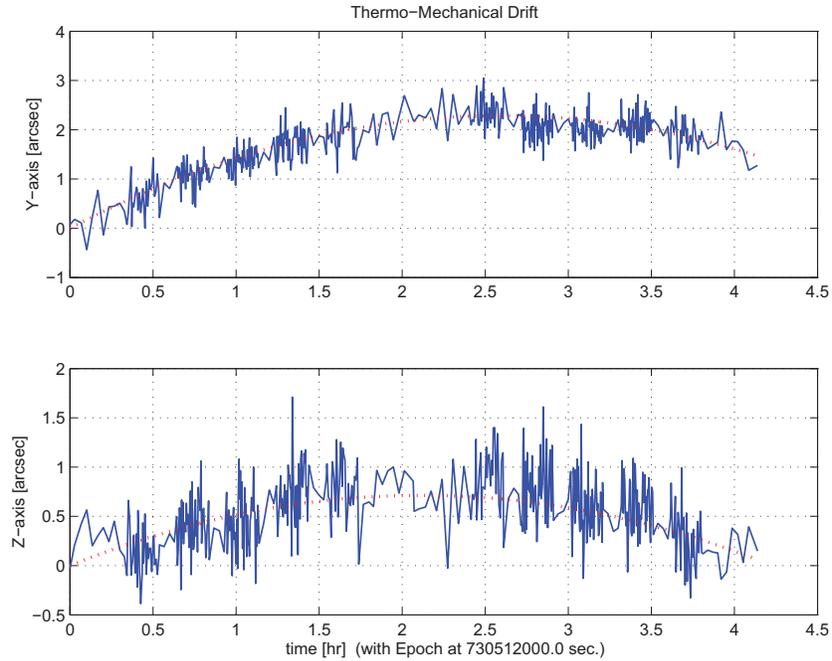


Figure 8.22: Thermo-mechanical boresight drift (equiv. angle in Body frame)

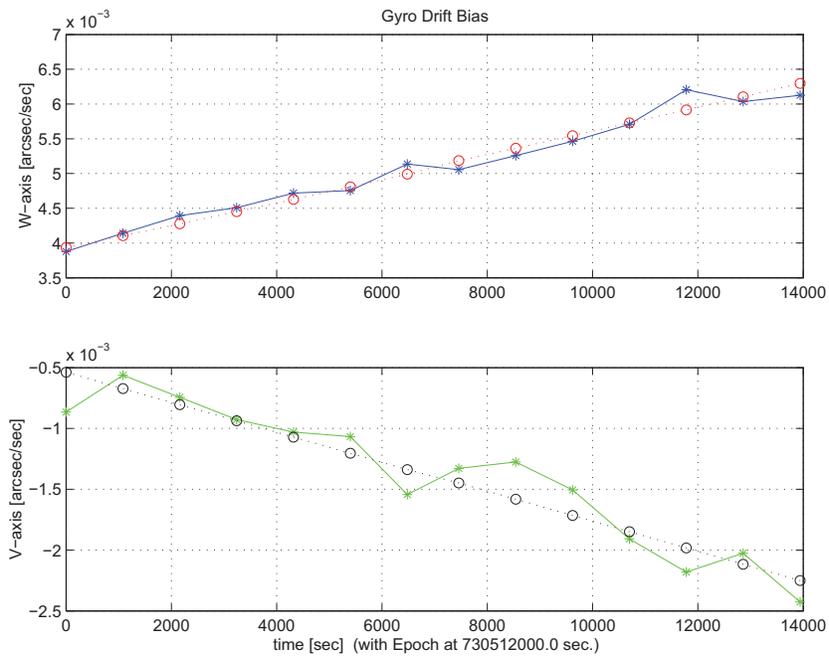


Figure 8.23: Gyro drift bias contribution (equiv. rate in (W,V) coords)

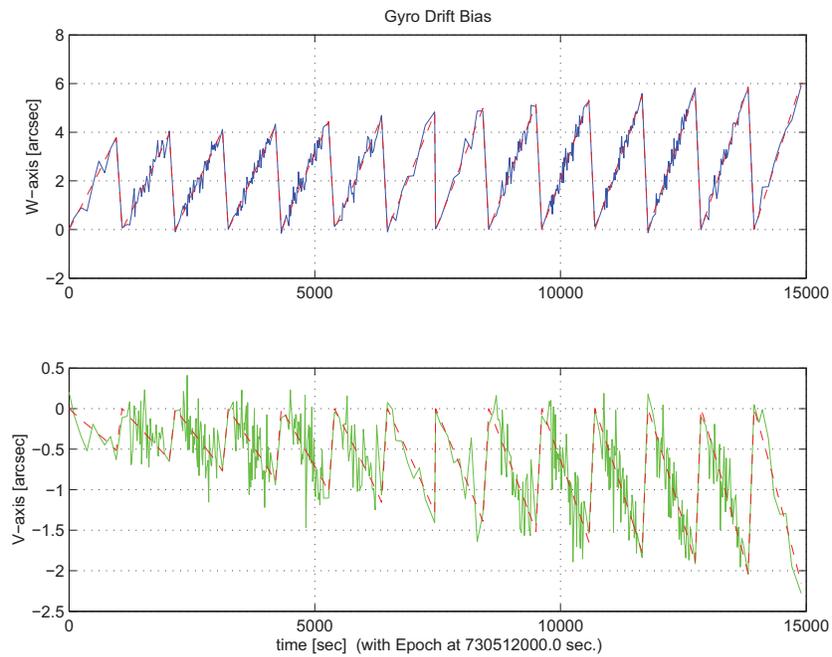


Figure 8.24: Gyro drift bias contribution (equiv. angle in (W,V) coords)

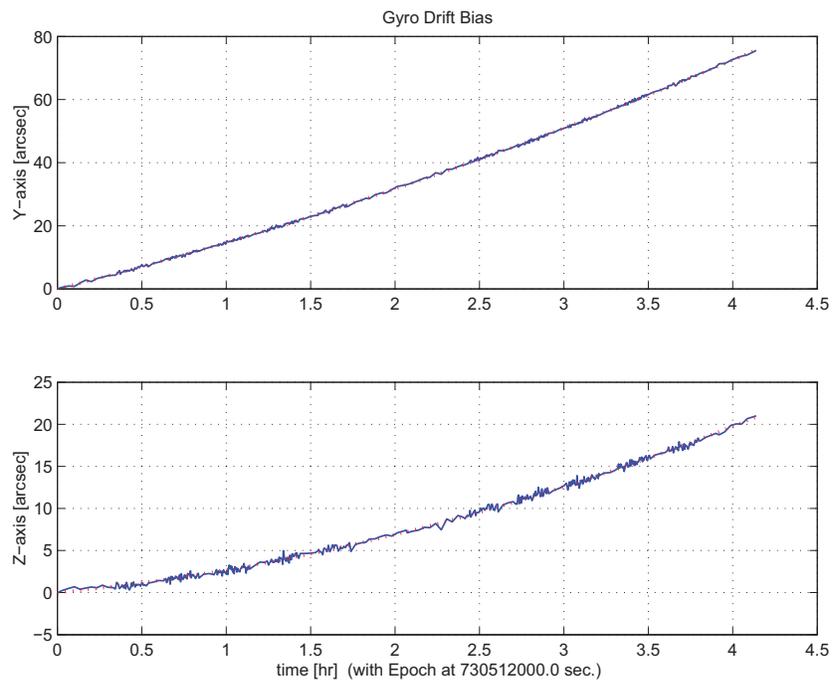


Figure 8.25: Gyro drift bias contribution (equiv. angle in Body frame)

### 8.3 IPF OUTPUT DATA (IF MINI FILE)

OUTPUT FILE NAME: IFmini991095.dat    DATE: 05-Aug-2003    TIME: 21:42  
 INSTRUMENT NAME: MIPS\_24um\_center    NF: 95  
 IPF FILTER VERSION: IPF.V2.0.1Beta    SW RELEASE DATE: August 5, 2003

```

----- IPF BROWN ANGLE SUMMARY -----
-----

```

Frame Number	WAS			IS		
	theta_Y (arcmin)	theta_Z (arcmin)	angle (deg)	theta_Y (arcmin)	theta_Z (arcmin)	angle (deg)
095	+6.641000	+3.931000	+0.000000	+6.641111	+3.932233	+0.016691
096	+6.595000	+6.712000	-0.000000	+6.618045	+6.975361	+0.016691
099	+7.687000	+3.943000	-0.000000	+7.784631	+3.921390	+0.016691
100	+5.574000	+3.918000	+0.000000	+5.475980	+3.941639	+0.016691
103	+6.746000	+3.932000	+0.000000	+6.755408	+3.931220	+0.016691
104	+6.558000	+3.930000	+0.000000	+6.549682	+3.933031	+0.016691
001	+0.000000	+0.000000	+0.000000	+9.608774	+0.843807	+0.016691
001	+0.000000	+0.000000	+0.000000	+9.544149	+6.948933	+0.016691
001	+0.000000	+0.000000	+0.000000	+3.749581	+0.904283	+0.016691
001	+0.000000	+0.000000	+0.000000	+3.699961	+6.991274	+0.016691

```

-----
OFFSET      NF      Delta_CW      Delta_CV
0           95           +0.000           +0.000      pixels
OFFSET FRAME NAME: MIPS_24um_center
Brown Angle  theta_Y(arcmin) theta_Z(arcmin) angle(deg)
WAS(FTB)    +6.641000      +3.931000      +0.000000
IS (EST)    +6.641111      +3.932233      +0.016691
dT_EST      +0.000111      +0.001233      +0.016691
T_sSIGMA    +0.000653      +0.000793      +0.009652
dT_EST/T_sSIGMA +0.169368      +1.554636      +1.729313
-----
OFFSET      NF      Delta_CW      Delta_CV
1           96           +0.000           -64.000     pixels
OFFSET FRAME NAME: MIPS_24um_plusY_edge
Brown Angle  theta_Y(arcmin) theta_Z(arcmin) angle(deg)
WAS(FTB)    +6.595000      +6.712000      -0.000000
IS (EST)    +6.618045      +6.975361      +0.016691
dT_EST      +0.023045      +0.263361      +0.016691
T_sSIGMA    +0.001229      +0.001348      +0.009652
dT_EST/T_sSIGMA +18.757968      +195.324120      +1.729314
-----
OFFSET      NF      Delta_CW      Delta_CV
2           99           +25.000           +0.000     pixels
OFFSET FRAME NAME: MIPS_24um_small_FOV1
Brown Angle  theta_Y(arcmin) theta_Z(arcmin) angle(deg)
WAS(FTB)    +7.687000      +3.943000      -0.000000
IS (EST)    +7.784631      +3.921390      +0.016691
dT_EST      +0.097631      -0.021610      +0.016691
T_sSIGMA    +0.000654      +0.000790      +0.009652
dT_EST/T_sSIGMA +149.346668      -27.351019      +1.729314
-----
OFFSET      NF      Delta_CW      Delta_CV
3          100           -25.500           +0.000     pixels
OFFSET FRAME NAME: MIPS_24um_small_FOV2
Brown Angle  theta_Y(arcmin) theta_Z(arcmin) angle(deg)
WAS(FTB)    +5.574000      +3.918000      +0.000000
IS (EST)    +5.475980      +3.941639      +0.016691
dT_EST      -0.098020      +0.023639      +0.016691
T_sSIGMA    +0.000628      +0.000759      +0.009652
dT_EST/T_sSIGMA -156.096434      +31.152524      +1.729313
-----

```

```

OFFSET      NF      Delta_CW      Delta_CV
  4         103         +2.500         +0.000      pixels
OFFSET FRAME NAME: MIPS_24um_large_FOV1
Brown Angle  theta_Y(arcmin)  theta_Z(arcmin)  angle(deg)
WAS(FTB)    +6.746000         +3.932000         +0.000000
IS (EST)    +6.755408         +3.931220         +0.016691
dT_EST      +0.009408         -0.000780         +0.016691
T_sSIGMA    +0.000654         +0.000794         +0.009652
dT_EST/T_sSIGMA  +14.382693         -0.981727         +1.729313
-----
OFFSET      NF      Delta_CW      Delta_CV
  5         104         -2.000         +0.000      pixels
OFFSET FRAME NAME: MIPS_24um_large_FOV2
Brown Angle  theta_Y(arcmin)  theta_Z(arcmin)  angle(deg)
WAS(FTB)    +6.558000         +3.930000         +0.000000
IS (EST)    +6.549682         +3.933031         +0.016691
dT_EST      -0.008318         +0.003031         +0.016691
T_sSIGMA    +0.000652         +0.000791         +0.009652
dT_EST/T_sSIGMA  -12.764356         +3.829302         +1.729313
-----
OFFSET      NF      Delta_CW      Delta_CV
  6          1         +64.000         +64.000      pixels
OFFSET FRAME NAME: CORNER PIXEL (DIAGNOSTICS)
Brown Angle  theta_Y(arcmin)  theta_Z(arcmin)  angle(deg)
WAS(FTB)    +0.000000         +0.000000         +0.000000
IS (EST)    +9.608774         +0.843807         +0.016691
dT_EST      +9.608774         +0.843807         +0.016691
T_sSIGMA    +0.001582         +0.001628         +0.009652
dT_EST/T_sSIGMA  +6074.913827         +518.409962         +1.729315
-----
OFFSET      NF      Delta_CW      Delta_CV
  7          1         +64.000         -64.000      pixels
OFFSET FRAME NAME: CORNER PIXEL (DIAGNOSTICS)
Brown Angle  theta_Y(arcmin)  theta_Z(arcmin)  angle(deg)
WAS(FTB)    +0.000000         +0.000000         +0.000000
IS (EST)    +9.544149         +6.948933         +0.016691
dT_EST      +9.544149         +6.948933         +0.016691
T_sSIGMA    +0.001496         +0.001595         +0.009652
dT_EST/T_sSIGMA  +6378.896190         +4357.851831         +1.729315
-----
OFFSET      NF      Delta_CW      Delta_CV
  8          1         -64.000         +64.000      pixels
OFFSET FRAME NAME: CORNER PIXEL (DIAGNOSTICS)
Brown Angle  theta_Y(arcmin)  theta_Z(arcmin)  angle(deg)
WAS(FTB)    +0.000000         +0.000000         +0.000000
IS (EST)    +3.749581         +0.904283         +0.016691
dT_EST      +3.749581         +0.904283         +0.016691
T_sSIGMA    +0.001449         +0.001538         +0.009652
dT_EST/T_sSIGMA  +2587.265561         +588.093853         +1.729315
-----
OFFSET      NF      Delta_CW      Delta_CV
  9          1         -64.000         -64.000      pixels
OFFSET FRAME NAME: CORNER PIXEL (DIAGNOSTICS)
Brown Angle  theta_Y(arcmin)  theta_Z(arcmin)  angle(deg)
WAS(FTB)    +0.000000         +0.000000         +0.000000
IS (EST)    +3.699961         +6.991274         +0.016691
dT_EST      +3.699961         +6.991274         +0.016691
T_sSIGMA    +0.001555         +0.001608         +0.009652
dT_EST/T_sSIGMA  +2379.601679         +4349.019953         +1.729315
-----
-----
VARNAME      MEAN      SIGMA      SCALED_SIGMA
a00          +1.0017847104128118E-001  +1.7447930652372362E-004  +1.9629417912081742E-004

```

b00	+9.9857079033987411E-002	+2.7389518853457290E-004	+3.0813987211271567E-004
c00	+9.9857841173256051E-003	+1.5852180433205965E-004	+1.7834153558996170E-004
a10	+9.9963029864107578E-001	+5.0018962900375219E-001	+5.6272754968044203E-001
b10	+6.8984316170071902E-001	+7.5708531007742474E-001	+8.5174249271715896E-001
c10	+7.6884724024437689E-001	+7.3434694730785255E-001	+8.2616118830156837E-001
d10	+9.2496011502265629E-001	+5.1166920647623926E-001	+5.7564240062472449E-001
a20	-3.3054384657566243E+002	+2.0987512242834891E+003	+2.3611547808021746E+003
b20	+2.2710611782966525E+003	+2.6567031466066878E+003	+2.9888665522632950E+003
c20	-1.8379514563129278E+003	+2.5590093642974434E+003	+2.8789582703834981E+003
d20	+1.1167713002555929E+003	+2.1860615870103934E+003	+2.4593814205205540E+003
a01	+1.9484922747036248E+000	+3.0555144454395855E-001	+3.4375405989925356E-001
b01	+2.4604185140619670E+000	+4.4595305179621858E-001	+5.0170985873823226E-001
c01	+1.7492124723813389E+000	+2.8431056582092984E-001	+3.1985746759949657E-001
d01	+2.5560408827877112E+000	+3.2420065273272469E-001	+3.6473494918407734E-001
e01	+2.1944640205947166E+000	+4.6062412452953683E-001	+5.1821523256384949E-001
f01	+2.1128848772350692E+000	+2.7835766572191900E-001	+3.1316028578690902E-001
del_alpha	+9.7101417618344898E-014	+2.7256708575263029E-004	+3.0664571873393633E-004
beta	+1.1000920263285241E+000	+4.2729516658701145E-004	+4.8071920755878261E-004
del_theta1	-2.2871996993838878E-014	+1.4973533883221575E-004	+1.6845651216209327E-004
del_theta2	+9.0660120101444905E-018	+1.6881829681581167E-007	+1.8992538229471094E-007
del_theta3	-8.7474420915015356E-017	+2.0499032761154080E-007	+2.3061994506920910E-007
del_arx	+7.0990322588492319E-015	+7.2282803569624263E-005	+8.1320208533276558E-005
del_ary	+6.8348415973396519E-016	+3.4062034813332439E-006	+3.8320757321205249E-006
del_arz	-4.0414493600256273E-016	+3.4047165321493002E-006	+3.8304028720245521E-006
brx	-1.9819354972179571E-008	+2.1985765498545561E-008	+2.4734611094370642E-008
bry	+2.3780509322415510E-009	+1.1303261347248942E-009	+1.2716490291899543E-009
brz	+9.0751675976722572E-010	+1.1290265487486977E-009	+1.2701869580281818E-009
crx	+2.1178513642499728E-012	+2.9244079535609258E-012	+3.2900420691469208E-012
cry	-2.5502544994697585E-013	+1.5629135512612073E-013	+1.7583221683650293E-013
crz	-1.1917760637055323E-013	+1.5608989658369953E-013	+1.7560557025015791E-013
bgx	+2.246333334022776E-007	+4.6377272785529821E-007	+5.2175750079909937E-007
bgy	+1.8275281016563494E-008	+1.3251018437683919E-009	+1.4907772380367154E-009
bgz	+2.6237650690920771E-009	+1.3742048351481108E-009	+1.5460194990090672E-009
cgx	-2.1063048197094546E-011	+5.0626060898833677E-011	+5.6955757472052268E-011
cgy	+8.5289246807484838E-013	+1.7130973939394889E-013	+1.9272832601019162E-013
cgz	+5.5944436732964080E-013	+1.7947285319163464E-013	+2.0191206105540552E-013

-----  
LSQF RESIDUAL SIGMA SCALE = +1.1250284233226688E+000  
-----

	a_mirror(1)	a_mirror(2)	a_mirror(3)
a_mirror_ipf	+0.0000000000000000E+000	+2.0083432207034409E-002	+9.9979830753546750E-001
a_mirror_tpf	-1.9088008036579826E-003	+1.9794372317572122E-002	+9.9980225009951107E-001
beta	beta_0	beta	beta_total
	+2.8047410000000001E-006	+1.1000920263285241E+000	+3.0854732100166912E-006

	qT(1)	qT(2)	qT(3)	qT(4)
FrmTbl:	-5.5224103706934371E-007	-9.6589398881636961E-004	-5.7174047628006817E-004	+9.9999937008046424E-001
Estim:	+1.4510453874923350E-004	-9.6599336627449701E-004	-5.7177904348510019E-004	+9.9999935943490215E-001
DelTheta	deltheta(1)	deltheta(2)	deltheta(3)	
	+2.9131341622516259E-004	-3.2167338266954286E-008	-3.5852934947754515E-007	[rad]
EulAngT	theta(1)	theta(2)	theta(3)	[rad]
Mean	+2.9131410882893196E-004	-1.9318207610871999E-003	-1.1438400782834333E-003	
SigmaT	+1.4973533883221575E-004	+1.6881829681581167E-007	+2.0499032761154080E-007	

	qR(1)	qR(2)	qR(3)	qR(4)
ASFILE:	+0.0000000000000000E+000	+0.0000000000000000E+000	+0.0000000000000000E+000	+1.0000000000000000E+000
Estim:	+5.2495135971495827E-005	-2.4052361893128234E-006	+1.7717388088944691E-007	+9.9999999861922217E-001
DelThetaR	delthetaR(1)	delthetaR(2)	delthetaR(3)	
	+1.0499027113983199E-004	-4.8104909735359424E-006	+3.5409523489211961E-007	[rad]
EulAngR	angR(1)	angR(2)	angR(3)	[rad]
Mean	+1.0499027113983199E-004	-4.8104909735359424E-006	+3.5409523489211961E-007	
SigmaR	+7.2282803569624263E-005	+3.4062034813332439E-006	+3.4047165321493002E-006	

```

-----
Initial Gyro Bias      Bg0(1)          Bg0(2)          Bg0(3)
+2.4582663655968712E-006 +2.4793742961420354E-006 +2.4976176758578363E-006
Gyro Bias Correction  Bg(1)           Bg(2)           Bg(3)
+2.246333334022776E-007 +1.8275281016563494E-008 +2.6237650690920771E-009
Total Gyro Bias       BgT(1)          BgT(2)          BgT(3)
+2.6828996989370990E-006 +2.4976495771585991E-006 +2.5002414409269283E-006

Initial Gyro Bias Rate Cg0(1)          Cg0(2)          Cg0(3)
+0.0000000000000000E+000 +0.0000000000000000E+000 +0.0000000000000000E+000
Gyro Bias Rate Correction Cg(1)          Cg(2)          Cg(3)
-2.1063048197094546E-011 +8.5289246807484838E-013 +5.5944436732964080E-013
Total Gyro Bias Rate   CgT(1)          CgT(2)          CgT(3)
-2.1063048197094546E-011 +8.5289246807484838E-013 +5.5944436732964080E-013
-----

OFFSET      NF      Delta_CW      Delta_CV
1           96      +0.000        -64.000      pixels
OFFSET FRAME NAME: MIPS_24um_plusY_edge
qT          qT(1)          qT(2)          qT(3)          qT(4)
WAS(FTB)   -9.3639450226028116E-007 -9.5920326392183247E-004 -9.7622022412795586E-004 +9.9999906346070921E-001
IS (EST)   +1.4468037500129608E-004 -9.6270268630143404E-004 -1.0143842495060132E-003 +9.9999901164737204E-001

DelTheta    deltheta(1)      deltheta(2)      deltheta(3)
Units       rad              rad              rad
+2.9116714246919979E-004 -6.7034052089031509E-006 -7.6608415885881338E-005
EulAngT     theta(1)         theta(2)         theta(3)          [rad]
Mean        +2.9131410882893201E-004 -1.9251111357299071E-003 -2.0290502144061232E-003
sSigmaT     +1.6845645244929723E-004 +3.5736277360800847E-007 +3.9221248245078690E-007
SigmaT      +1.4973528575551581E-004 +3.1764777333586837E-007 +3.4862450967454057E-007
-----

OFFSET      NF      Delta_CW      Delta_CV
2           99      +25.000       +0.000       pixels
OFFSET FRAME NAME: MIPS_24um_small_FOV1
qT          qT(1)          qT(2)          qT(3)          qT(4)
WAS(FTB)   -6.4117382846981564E-007 -1.1180284132340469E-003 -5.7348571352373249E-004 +9.9999921056278462E-001
IS (EST)   +1.4501117827978903E-004 -1.1323113892338270E-003 -5.7017768347590169E-004 +9.9999918586971126E-001

DelTheta    deltheta(1)      deltheta(2)      deltheta(3)
Units       rad              rad              rad
+2.9132816512432409E-004 -2.8399845793231449E-005 +6.2861845283510255E-006
EulAngT     theta(1)         theta(2)         theta(3)          [rad]
Mean        +2.9131410882893196E-004 -2.2644575057624533E-003 -1.1406860061260421E-003
sSigmaT     +1.6845649867954388E-004 +1.9016055798045444E-007 +2.2983424313973420E-007
SigmaT      +1.4973532684803018E-004 +1.6902733658837931E-007 +2.0429194354124825E-007
-----

OFFSET      NF      Delta_CW      Delta_CV
3           100     -25.500       +0.000       pixels
OFFSET FRAME NAME: MIPS_24um_small_FOV2
qT          qT(1)          qT(2)          qT(3)          qT(4)
WAS(FTB)   -4.6198041860920160E-007 -8.1070521711647421E-004 -5.6984978266998100E-004 +9.9999950901391088E-001
IS (EST)   +1.4520038901489615E-004 -7.9653229295302314E-004 -5.7317198286516483E-004 +9.9999950796339465E-001

DelTheta    deltheta(1)      deltheta(2)      deltheta(3)
Units       rad              rad              rad
+2.9130295933233721E-004 +2.8512848977866270E-005 -6.8764108508032863E-006
EulAngT     theta(1)         theta(2)         theta(3)          [rad]
Mean        +2.9131410882893196E-004 -1.5928980260865319E-003 -1.1465764211241560E-003
sSigmaT     +1.6845650672371083E-004 +1.8266175731444702E-007 +2.2073394851477005E-007
SigmaT      +1.4973533399822016E-004 +1.6236190439968812E-007 +1.9620299713215465E-007
-----

```

```

OFFSET      NF      Delta_CW      Delta_CV
  4          103      +2.500      +0.000      pixels
OFFSET FRAME NAME: MIPS_24um_large_FOV1
qT          qT(1)          qT(2)          qT(3)          qT(4)
WAS(FTB)   -5.6111515131371191E-007 -9.8116560995659325E-004 -5.7188591179032191E-004 +9.9999935512990956E-001
IS (EST)   +1.4509517353874301E-004 -9.8261716212673790E-004 -5.7162941045143883E-004 +9.9999934332514462E-001

```

```

DelTheta    deltheta(1)          deltheta(2)          deltheta(3)
Units       rad          rad          rad
            +2.9131455381324535E-004 -2.7365614206183226E-006 +2.2676273566199052E-007
EulAngT     theta(1)          theta(2)          theta(3)          [rad]
Mean        +2.9131410882893196E-004 -1.9650684170795267E-003 -1.1435456733001341E-003
sSigmaT     +1.6845651165311556E-004 +1.9026766261452271E-007 +2.3098382925365007E-007
SigmaT      +1.4973533837980255E-004 +1.6912253830226309E-007 +2.0531377204805226E-007

```

```

OFFSET      NF      Delta_CW      Delta_CV
  5          104      -2.000      +0.000      pixels
OFFSET FRAME NAME: MIPS_24um_large_FOV2
qT          qT(1)          qT(2)          qT(3)          qT(4)
WAS(FTB)   -5.4520033785992621E-007 -9.5382213577026277E-004 -5.7159503889057917E-004 +9.9999938175088265E-001
IS (EST)   +1.4511203527994054E-004 -9.5269561073110093E-004 -5.7189706943792625E-004 +9.9999937212345913E-001

```

```

DelTheta    deltheta(1)          deltheta(2)          deltheta(3)
Units       rad          rad          rad
            +2.9131242708483918E-004 +2.4196729888382608E-006 -8.8158025892807909E-007
EulAngT     theta(1)          theta(2)          theta(3)          [rad]
Mean        +2.9131410882893196E-004 -1.9052251994402232E-003 -1.1440722419152306E-003
sSigmaT     +1.6845651243514804E-004 +1.8956482568126830E-007 +2.3021998363057301E-007
SigmaT      +1.4973533907492497E-004 +1.6849781014545915E-007 +2.0463481531483380E-007

```

```

OFFSET      NF      Delta_CW      Delta_CV
  6          1      +64.000      +64.000      pixels
OFFSET FRAME NAME: CORNER PIXEL (DIAGNOSTICS)
qT          qT(1)          qT(2)          qT(3)          qT(4)
WAS(FTB)   +0.0000000000000000E+000 +0.0000000000000000E+000 +0.0000000000000000E+000 +1.0000000000000000E+000
IS (EST)   +1.4548539512439278E-004 -1.3975569702967018E-003 -1.2252306968373375E-004 +9.9999900532781127E-001

```

```

DelTheta    deltheta(1)          deltheta(2)          deltheta(3)
Units       rad          rad          rad
            +2.9131410882893201E-004 -2.7950791491462837E-003 -2.4545350514573538E-004
EulAngT     theta(1)          theta(2)          theta(3)          [rad]
Mean        +2.9131410882893196E-004 -2.7950791491462837E-003 -2.4545350514573538E-004
sSigmaT     +1.6845636436936195E-004 +4.6010185970635906E-007 +4.7347374312805602E-007
SigmaT      +1.4973520746421805E-004 +4.0896909817397342E-007 +4.2085491647374961E-007

```

```

OFFSET      NF      Delta_CW      Delta_CV
  7          1      +64.000      -64.000      pixels
OFFSET FRAME NAME: CORNER PIXEL (DIAGNOSTICS)
qT          qT(1)          qT(2)          qT(3)          qT(4)
WAS(FTB)   +0.0000000000000000E+000 +0.0000000000000000E+000 +0.0000000000000000E+000 +1.0000000000000000E+000
IS (EST)   +1.4425387253851662E-004 -1.3882862082997916E-003 -1.0104779730493976E-003 +9.9999851539214302E-001

```

```

DelTheta    deltheta(1)          deltheta(2)          deltheta(3)
Units       rad          rad          rad
            +2.9131410882893196E-004 -2.7762803302265374E-003 -2.0213626436624585E-003
EulAngT     theta(1)          theta(2)          theta(3)          [rad]
Mean        +2.9131410882893196E-004 -2.7762803302265374E-003 -2.0213626436624585E-003
sSigmaT     +1.6845638288299759E-004 +4.3522895616195560E-007 +4.6384382075936503E-007
SigmaT      +1.4973522392036729E-004 +3.8686040915886067E-007 +4.1229520174205429E-007

```

```

OFFSET      NF      Delta_CW      Delta_CV
  8          1          -64.000          +64.000      pixels
OFFSET FRAME NAME: CORNER PIXEL (DIAGNOSTICS)
qT          qT(1)          qT(2)          qT(3)          qT(4)
WAS(FTB) +0.0000000000000000E+000 +0.0000000000000000E+000 +0.0000000000000000E+000 +1.0000000000000000E+000
IS (EST)  +1.4558530456298621E-004 -5.4537359146879282E-004 -1.3144311288958377E-004 +9.999983204762244E-001

```

```

DelTheta    deltheta(1)          deltheta(2)          deltheta(3)
Units        rad                  rad                  rad
              +2.9131410882893201E-004 -1.0907089436323330E-003 -2.6304513788637989E-004
EulAngT      theta(1)          theta(2)          theta(3)          [rad]
Mean         +2.9131410882893201E-004 -1.0907089436323326E-003 -2.6304513788637989E-004
sSigmaT      +1.6845639152055159E-004 +4.2156822242062697E-007 +4.4728428423462484E-007
SigmaT       +1.4973523159799910E-004 +3.7471784150622941E-007 +3.9757598560364500E-007

```

```

-----
OFFSET      NF      Delta_CW      Delta_CV
  9          1          -64.000          -64.000      pixels
OFFSET FRAME NAME: CORNER PIXEL (DIAGNOSTICS)
qT          qT(1)          qT(2)          qT(3)          qT(4)
WAS(FTB) +0.0000000000000000E+000 +0.0000000000000000E+000 +0.0000000000000000E+000 +1.0000000000000000E+000
IS (EST)  +1.4510975813337586E-004 -5.3828529956594616E-004 -1.0167608463953416E-003 +9.999932769451172E-001

```

```

DelTheta    deltheta(1)          deltheta(2)          deltheta(3)
Units        rad                  rad                  rad
              +2.9131410882893207E-004 -1.0762749992936219E-003 -2.0336791262474466E-003
EulAngT      theta(1)          theta(2)          theta(3)          [rad]
Mean         +2.9131410882893201E-004 -1.0762749992936221E-003 -2.0336791262474466E-003
sSigmaT      +1.6845640070651876E-004 +4.5229208274313220E-007 +4.6761779624702138E-007
SigmaT       +1.4973523976309695E-004 +4.0202724959368474E-007 +4.1564976186642013E-007

```

```

-----
q(1)          q(2)          q(3)          q(4)
PCRS1A: +5.2603126189317361E-007 +3.7451819048221893E-004 -1.4045531552277380E-003 +9.9999894348258311E-001
PCRS2A: -5.1968404336718985E-007 +3.8455385122900486E-004 +1.3513933841514693E-003 +9.9999901292650617E-001

```

```

-----
***** CS-FILE PARAMETERS: ***** ***** AS-FILE PARAMETERS: *****
Row (01) PIX2RADX: +1.2087416876100000E-005 Row (1) TASTART: +7.3051200000000000E+008
Row (02) PIX2RADY: +1.2595908372599999E-005 Row (2) TASTOP: +7.3052690000000000E+008
Row (03) CX0: +6.4500000000000000E+001 Row (3) S/C TIME: +7.3051200000000000E+008
Row (04) CY0: +6.4500000000000000E+001 Row (4) QR1: +0.0000000000000000E+000
Row (05) BETA0: +2.8047410000000001E-006 Row (5) QR2: +0.0000000000000000E+000
Row (06) GAMMA_EO: +2.0070000000000000E+003 Row (6) QR3: +0.0000000000000000E+000
Row (07) D11: -1.0000000000000000E+000 Row (7) QR4: +1.0000000000000000E+000
Row (08) D12: +0.0000000000000000E+000
Row (09) D21: +0.0000000000000000E+000
Row (10) D22: -1.0000000000000000E+000
Row (11) DG: +1.0000000000000000E+000

```

```

-----
INITIAL STA-TO-PCRS ALIGNMENT (R) KNOWLEDGE (1-SIGMA)
SIGMA(X)      SIGMA(Y)      SIGMA(Z)
1.41451738E+001  7.65467990E-001  7.65467990E-001 [arcsec]

```

```

-----
PIX2RADX = 1.208741687610E-005 [rad/pixel]
XPIXSIZE = 2.4932 [arcsec]
PIX2RADY = 1.259590837260E-005 [rad/pixel]
YPIXSIZE = 2.5981 [arcsec]
CX0 = 64.5 [pixel] = 160.81 [arcsec]
CY0 = 64.5 [pixel] = 167.58 [arcsec]

```

```

-----
NOMINAL BETA0 = 2.804741000000E-006 [rad/encoder unit]
ENCODER UNIT SIZE = 0.58 [arcsec]
GAMMA_EO = 2007.00 [encoder unit] = 1161.09 [arcsec]

```

```

-----
      | -1 | +0 |
FLIP MATRIX D = |----|----|   and   DG = +1
      | +0 | -1 |
-----

```

## 8.4 IPF EXECUTION LOG

```

*****
IPF EXECUTION-LOG FILE NAME:  LG991095.dat
INSTRUMENT TYPE:             MIPS_24um_center
IPF FILTER EXECUTION DATE:   05-Aug-2003   TIME: 21:38
IPF FILTER VERSION USED:     IPF.V2.0.1Beta
*****

```

```

----- Loading & Preparing Input Files -----
AAFILE:  AA991095 Loaded!           AAFILE dimension = 149001 X 21
ASFILE:  AS991095 Loaded!
CAFILE:  CA991095 Loaded!           CAFILE dimension = 432 X 15
CBFILE:  CB991095 Loaded!           CBFILE dimension = 42 X 15
CCFILE:  CC991095 Created!          CCFILE dimension = 474 X 19
CSFILE:  CS991095 Loaded!
Loading Input Files Completed!
-----

```

```

----- Selected Mask Vectors -----
index =  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
-----
mask1 = [ 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 ]
mask2 = [ 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 ]
-----

```

```

----- Selected Initial Gyro Bias Parameters -----
IPF>> RNFILE FLAG SET TO USE PREVIOUS AG & AC FILE!
AGFILE USED: AG991095      ACFILE USED: AC991095
IPF Linearized Using Following Nominal Gyro Bias Estimates
bg0 = [+2.4582663655968712E-006 +2.4793742961420354E-006 +2.4976176758578363E-006 ]
cg0 = [+0.0000000000000000E+000 +0.0000000000000000E+000 +0.0000000000000000E+000 ]
-----

```

```

IPF>> GYRO PREPROCESSOR BYPASSED!
IPF>> LOADING PREVIOUS AG & AC FILE!

```

```

FRAME TABLE ENTRIES FOR PCRS LOADED TO TPCRS
q_PCRS4 = [ +5.2603126189317361E-007      q_PCRS5 = [ +7.3379987833742897E-007
           +3.7451819048221893E-004          +5.2236196154513707E-004
           -1.4045531552277380E-003          -1.4047712280184723E-003
           +9.9999894348258311E-001 ];      +9.9999887687698918E-001 ];
q_PCRS8 = [ -5.1968404336718985E-007      q_PCRS9 = [ -7.1963421681856818E-007
           +3.8455385122900486E-004          +5.3239763239987400E-004
           +1.3513933841514693E-003          +1.3516841804518383E-003
           +9.9999901292650617E-001 ];      +9.9999894475050310E-001 ];
-----

```

```

----- Initial Conditions for State -----  ----- Inital Square-Root Cov (diag) -----
p1(01) = a00 = +0.0000000000000000E+000  Sigma_initial(01,01) = 1.0000000000000000E+000
p1(02) = b00 = +0.0000000000000000E+000  Sigma_initial(02,02) = 1.0000000000000000E+000
p1(03) = c00 = +0.0000000000000000E+000  Sigma_initial(03,03) = 1.0000000000000000E+000
p1(04) = a10 = +0.0000000000000000E+000  Sigma_initial(04,04) = 3.6384517097704270E+002
p1(05) = b10 = +0.0000000000000000E+000  Sigma_initial(05,05) = 3.6384517097704270E+002
p1(06) = c10 = +0.0000000000000000E+000  Sigma_initial(06,06) = 3.6384517097704270E+002
p1(07) = d10 = +0.0000000000000000E+000  Sigma_initial(07,07) = 3.6384517097704270E+002
-----

```

p1(08) = a20 = +0.000000000000000E+000 Sigma\_initial(08,08) = 1.3238330844331343E+007  
p1(09) = b20 = +0.000000000000000E+000 Sigma\_initial(09,09) = 1.3238330844331343E+007  
p1(10) = c20 = +0.000000000000000E+000 Sigma\_initial(10,10) = 1.3238330844331343E+007  
p1(11) = d20 = +0.000000000000000E+000 Sigma\_initial(11,11) = 1.3238330844331343E+007  
p1(12) = a01 = +0.000000000000000E+000 Sigma\_initial(12,12) = 1.000000000000000E+004  
p1(13) = b01 = +0.000000000000000E+000 Sigma\_initial(13,13) = 1.000000000000000E+004  
p1(14) = c01 = +0.000000000000000E+000 Sigma\_initial(14,14) = 1.000000000000000E+004  
p1(15) = d01 = +0.000000000000000E+000 Sigma\_initial(15,15) = 1.000000000000000E+004  
p1(16) = e01 = +0.000000000000000E+000 Sigma\_initial(16,16) = 1.000000000000000E+004  
p1(17) = f01 = +0.000000000000000E+000 Sigma\_initial(17,17) = 1.000000000000000E+004

p2f(01) = am1 = +0.000000000000000E+000  
p2f(02) = am2 = +0.000000000000000E+000 Sigma\_initial(18,18) = 1.000000000000001E-001  
p2f(03) = am3 = +1.000000000000000E+000  
p2f(04) = beta = +1.000000000000000E+000 Sigma\_initial(19,19) = 1.000000000000001E-001  
p2f(05) = qT1 = -5.5224103706934371E-007 Sigma\_initial(20,20) = 1.000000000000001E-001  
p2f(06) = qT2 = -9.6589398881636961E-004 Sigma\_initial(21,21) = 1.000000000000000E-002  
p2f(07) = aT3 = -5.7174047628006817E-004 Sigma\_initial(22,22) = 1.000000000000000E-002  
p2f(08) = qT4 = +9.9999937008046424E-001  
p2f(09) = qR1 = +0.000000000000000E+000 Sigma\_initial(23,23) = 3.4288868839039716E-003  
p2f(10) = qR2 = +0.000000000000000E+000 Sigma\_initial(24,24) = 1.8555467699985827E-004  
p2f(11) = qR3 = +0.000000000000000E+000 Sigma\_initial(25,25) = 1.8555467699985827E-004  
p2f(12) = qR4 = +1.000000000000000E+000  
p2f(13) = brx = +0.000000000000000E+000 Sigma\_initial(26,26) = 6.7114093959731537E-005  
p2f(14) = bry = +0.000000000000000E+000 Sigma\_initial(27,27) = 6.7114093959731537E-005  
p2f(15) = brz = +0.000000000000000E+000 Sigma\_initial(28,28) = 6.7114093959731537E-005  
p2f(16) = crx = +0.000000000000000E+000 Sigma\_initial(29,29) = 4.5043016080356737E-009  
p2f(17) = cry = +0.000000000000000E+000 Sigma\_initial(30,30) = 4.5043016080356737E-009  
p2f(18) = crz = +0.000000000000000E+000 Sigma\_initial(31,31) = 4.5043016080356737E-009  
p2f(19) = bgx = +0.000000000000000E+000 Sigma\_initial(32,32) = 6.7114093959731537E-005  
p2f(20) = bgy = +0.000000000000000E+000 Sigma\_initial(33,33) = 6.7114093959731537E-005  
p2f(21) = bgz = +0.000000000000000E+000 Sigma\_initial(34,34) = 6.7114093959731537E-005  
p2f(22) = cgx = +0.000000000000000E+000 Sigma\_initial(35,35) = 4.5043016080356737E-009  
p2f(23) = cgy = +0.000000000000000E+000 Sigma\_initial(36,36) = 4.5043016080356737E-009  
p2f(24) = cgz = +0.000000000000000E+000 Sigma\_initial(37,37) = 4.5043016080356737E-009

----- IPF KALMAN FILTER STARTED -----  
Iteration#001: |dp1| = +3.136219263208E+003 RMS(|Res|) = +3.136219263208E+003  
Iteration#002: |dp1| = +1.379890498077E+001 RMS(|Res|) = +1.379890498077E+001  
Iteration#003: |dp1| = +9.830901580851E-002 RMS(|Res|) = +9.830901580851E-002  
Iteration#004: |dp1| = +8.172273105503E-001 RMS(|Res|) = +8.172273105503E-001  
Iteration#005: |dp1| = +6.964494745514E-002 RMS(|Res|) = +6.964494745514E-002  
Iteration#006: |dp1| = +2.750847999648E-003 RMS(|Res|) = +2.750847999648E-003  
Iteration#007: |dp1| = +3.914500766379E-005 RMS(|Res|) = +3.914500766379E-005  
Iteration#008: |dp1| = +1.728921133155E-006 RMS(|Res|) = +1.728921133155E-006  
Iteration#009: |dp1| = +2.366387844766E-006 RMS(|Res|) = +2.366387844766E-006  
Iteration#010: |dp1| = +4.424385829053E-007 RMS(|Res|) = +4.424385829053E-007  
Iteration#011: |dp1| = +3.897003643676E-007 RMS(|Res|) = +3.897003643676E-007  
Iteration#012: |dp1| = +8.127892945278E-007 RMS(|Res|) = +8.127892945278E-007  
Iteration#013: |dp1| = +1.584513669626E-006 RMS(|Res|) = +1.584513669626E-006  
Iteration#014: |dp1| = +2.119071531128E-006 RMS(|Res|) = +2.119071531128E-006  
Iteration#015: |dp1| = +2.367526536459E-006 RMS(|Res|) = +2.367526536459E-006  
Iteration#016: |dp1| = +8.719548153950E-007 RMS(|Res|) = +8.719548153950E-007  
Iteration#017: |dp1| = +1.231693233202E-007 RMS(|Res|) = +1.231693233202E-007  
Iteration#018: |dp1| = +9.147149207192E-007 RMS(|Res|) = +9.147149207192E-007  
Iteration#019: |dp1| = +5.535627389322E-007 RMS(|Res|) = +5.535627389322E-007  
Iteration#020: |dp1| = +2.742907176399E-007 RMS(|Res|) = +2.742907176399E-007  
Iteration#021: |dp1| = +3.054115708920E-007 RMS(|Res|) = +3.054115708920E-007  
Iteration#022: |dp1| = +3.245014548972E-007 RMS(|Res|) = +3.245014548972E-007  
Iteration#023: |dp1| = +1.869249742040E-006 RMS(|Res|) = +1.869249742040E-006  
Iteration#024: |dp1| = +2.203383761894E-006 RMS(|Res|) = +2.203383761894E-006  
Iteration#025: |dp1| = +1.813571801694E-006 RMS(|Res|) = +1.813571801694E-006  
IPF Kalman Filter Completed with Error |dp1| + |dp2| = +1.8135718016940001E-006

```

-----
----- IPF LEAST SQUARES FILTER STARTED -----
Iteration#001  COND#=#+1.571988770362E+012,  |dp|=+3.135179836901E+003
Iteration#002  COND#=#+1.571989273295E+012,  |dp|=+1.598694123393E+001
Iteration#003  COND#=#+1.571989272655E+012,  |dp|=+2.287588387184E-002
Iteration#004  COND#=#+1.571989277419E+012,  |dp|=+2.116182767216E-003
Iteration#005  COND#=#+1.571989279165E+012,  |dp|=+7.927992189079E-005
Iteration#006  COND#=#+1.571989280408E+012,  |dp|=+1.450568192233E-006
Iteration#007  COND#=#+1.571989277798E+012,  |dp|=+4.795100932354E-008
Iteration#008  COND#=#+1.571989273445E+012,  |dp|=+1.134975818328E-007
Iteration#009  COND#=#+1.571989269168E+012,  |dp|=+6.629818756727E-008
Iteration#010  COND#=#+1.571989274939E+012,  |dp|=+9.155340425099E-008
Iteration#011  COND#=#+1.571989280075E+012,  |dp|=+1.025535672956E-007
Iteration#012  COND#=#+1.571989270783E+012,  |dp|=+1.393199233679E-007
Iteration#013  COND#=#+1.571989278079E+012,  |dp|=+1.538824474536E-008
Iteration#014  COND#=#+1.571989276816E+012,  |dp|=+6.875998764586E-008
Iteration#015  COND#=#+1.571989272449E+012,  |dp|=+3.929231977738E-008
Iteration#016  COND#=#+1.571989276204E+012,  |dp|=+8.871462548943E-008
Iteration#017  COND#=#+1.571989274690E+012,  |dp|=+7.875629633004E-008
Iteration#018  COND#=#+1.571989274914E+012,  |dp|=+3.527463973771E-008
Iteration#019  COND#=#+1.571989275874E+012,  |dp|=+4.807581430322E-008
Iteration#020  COND#=#+1.571989280842E+012,  |dp|=+7.313823132831E-008
Iteration#021  COND#=#+1.571989269595E+012,  |dp|=+9.592392240910E-008
Iteration#022  COND#=#+1.571989276512E+012,  |dp|=+1.235954492925E-007
Iteration#023  COND#=#+1.571989277018E+012,  |dp|=+2.551876437337E-008
Iteration#024  COND#=#+1.571989273122E+012,  |dp|=+4.580802225882E-008
Iteration#025  COND#=#+1.571989278072E+012,  |dp|=+9.542420247573E-008
IPF Least Squares Filter Completed with Error |dp1| + |dp2| = +9.5424202475733262E-008
-----
Total Execution Time: 275 seconds

```

## 9 CONCLUSIONS

This report has summarized the Spitzer Instrument Pointing Frame (IPF) Kalman Filter algorithm. The main novelty of the IPF filter design lies in its ability to handle a large variety of array types (cameras, arrays with scanning mirrors, spectroscopy slits, etc.) in a single formulation, and in its ability to estimate both science and engineering parameters as part of the same filter state. With its companion User's Guide D-Document [13], the formulation, implementation and operation of the IPF filter can be retraced from its basic conceptual level down to its detailed software implementation.

The resulting IPF Kalman filter algorithm was thoroughly tested and all operational modes were validated using the FLUTE unit test environment. In simulation tests, the IPF filter has demonstrated the capability for calibrating frames to an accuracy of 0.1 arcseconds, per-axis, 1-sigma (as needed to meet requirements for Spitzer's most stringent pointing frames).

In addition to demonstrating its nominal capability, the IPF algorithm has been designed to handle various off-nominal cases and specialized operations. For example, the filter accommodates slit type instruments and supports 3 different LITE modes of operation (to allow use with smaller data sets and to allow for troubleshooting). A centroid data editing capability has also been added to allow the user to diagnose data glitches and provide simple fix-and-run capability.

The calibration parameters estimated by the IPF filter will play a key role in supporting Spitzer's in-flight precision pointing capability, and for supporting all ground pointing reconstruction efforts.

## A APPENDIX: ACRONYMS

AA	Attitude History Binary Input File
AC	Compressed Attitude Intermediary File
AG	Gyro-based Attitude Propagation and Linearization Intermediary File
AS	Attitude Supplemental File
CA	Centroid A File (instrument centroids)
CB	Centroid B File (PCRS Centroids)
CC	Merged and Cleaned Centroid Data File
CR	Merged Raw Centroid Data File
CS	Centroid Supplemental File
CVET	C-File Visualization and Editing Tool
DOM	Distributed Object Manager (Spitzer's Main Archival Mission Database)
FEO	Flight Engineering Office
FF	Offset (Inferred Frames) File
FLUTE	Filter Unit Test Environment
IF	IPF Filter Output File
IOC	In-Orbit Checkout
IPF	Instrument Pointing Frame
IPT	Integrated Product Team
IRAC	Infrared Array Camera
IRS	Infrared Spectrograph
KF	Kalman Filter
LG	IPF Filter Log File
MF	Multi-MIPS Run Tool Output File
MIPL	Multi-mission Image Processing Laboratory
MIPS	Multi-band Imaging Photometer for SIRTf
MUI	MATLAB User Interface
NF	Instrument Frame Table Number (from 001 to 128)
OET	Observatory Engineering Team
PCRS	Pointing Control Reference Sensor
RN	IPF Filter Run Configuration File
SBF	Spacecraft Body Frame
SI	Science Instrument
SIS	Software Interface Specification
SSC	Spitzer Science Center
STA	Star Tracker Assembly
SV	Science Verification
TPF	Telescope Pointing Frame
TR	IPF Run TAR file

## B APPENDIX: IPF Filter Interfaces

### B.1 Velocity Correction of Star Data

This section describes the velocity aberration correction (cf., [21]) which is applied to all RA and DEC values.

Note: Star Unit Vector (Interface with Input Data Files as specified in SIS)

$$\text{Star Unit Vector } \ell = \frac{u_{star} + V_{SC}/c}{\|u_{star} + V_{SC}/c\|} \quad \text{where } u_{star} = \begin{bmatrix} \cos(\text{DEC}) * \cos(\text{RA}) \\ \cos(\text{DEC}) * \sin(\text{RA}) \\ \sin(\text{DEC}) \end{bmatrix}$$

$V_{SC}$	Spacecraft Velocity in ICRS [km/sec]	A FILE Col(17-19)
RA	Right Ascension of Source (target star) [rad]	C FILE Col(13)
DEC	Declination of Source (target star) [rad]	C FILE Col(14)
c	Speed of Light [km/sec]	

### B.2 Centroid Measurements to Oriented Angular Pixel Coordinates

Note: Centroid Measurement (Interface with Input Data Files as specified in SIS)

$$y = \begin{bmatrix} y_w \\ y_v \end{bmatrix} = \begin{bmatrix} D11 * (CX - CX0) * PIX2RADX + D12 * (CY - CY0) * PIX2RADY \\ D21 * (CX - CX0) * PIX2RADX + D22 * (CY - CY0) * PIX2RADY \end{bmatrix}$$

CX	X-Axis Centroid (pixels)	C FILE Col(8)
CY	Y-Axis Centroid (pixels)	C FILE Col(9)
PIX2RADX	Nominal angular pixel size in X direction (rad/pixel)	CS FILE Row(1)
PIX2RADY	Nominal angular pixel size in Y direction (rad/pixel)	CS FILE Row(2)
CX0	Center pixel along X axis (pixels)	CS FILE Row(3)
CY0	Center pixel along Y axis (pixels)	CS FILE Row(4)
D11	x-to-w frame flip parameter (D11=0,+1,-1)	CS FILE Row(7)
D12	y-to-w frame flip parameter (D12=0,+1,-1)	CS FILE Row(8)
D21	x-to-v frame flip parameter (D21=0,+1,-1)	CS FILE Row(9)
D22	y-to-v frame flip parameter (D22=0,+1,-1)	CS FILE Row(10)

### B.3 Mirror Encoder Data to Mirror Angle Conversion

Note: Mirror Rotation angle  $\Gamma$  (Interface with Input Data Files as specified in SIS)

$$\Gamma = DG * BETA0 * (GAMMA_E - GAMMA_E0)$$

where,

GAMMA_E	Measured Mirror Encoder Angle [encoder units]	C FILE Col(12)
GAMMA_E0	MIPS Reference Mirror Position [encoder units]	CS FILE Row(6)
BETA0	Nominal Mirror Scale Factor [rad/(encoder unit)]	CS FILE Row(5)
DG	MIPS Mirror Flip Parameter	CS FILE Row(11)

## B.4 Inferred Frames from Desired Offsets

The definition of an Inferred Frame has been given in Section 6.6.1 and Section 6.6.2 in terms of the offset vector,

$$y = \begin{bmatrix} \Delta w \\ \Delta v \end{bmatrix} \quad (\text{B.1})$$

where  $\Delta w$  and  $\Delta v$  are angular offsets in oriented angular pixel coordinates. The values for  $\Delta w$  and  $\Delta v$  are calculated from the quantities DELTA\_CW, DELTA\_CV provided by the user in the O-File according to the following formula,

$$y = \begin{bmatrix} \Delta w \\ \Delta v \end{bmatrix} = D \begin{bmatrix} \text{PIX2RADX} & 0 \\ 0 & \text{PIX2RADY} \end{bmatrix} D^{-1} \begin{bmatrix} \text{DELTA\_CW} \\ \text{DELTA\_CV} \end{bmatrix} \quad (\text{B.2})$$

where,

$$D = \begin{bmatrix} \text{D11} & \text{D12} \\ \text{D21} & \text{D22} \end{bmatrix} \quad (\text{B.3})$$

The formula (B.2) is motivated by the need to correctly apply nominal pixel scales PIX2RADX, PIX2RADY (which are specified in  $x, y$  coordinates) to DELTA\_CW, DELTA\_CV which are specified in  $v, w$  coordinates.

DELTA_CW	Pixel offset in W direction [pixels]	O-FILE Col(2)
DELTA_CV	Pixel offset in W direction [pixels]	O-FILE Col(3)
PIX2RADX	Nominal angular pixel size in X direction (rad/pixel)	CS FILE Row(1)
PIX2RADY	Nominal angular pixel size in Y direction (rad/pixel)	CS FILE Row(2)
D11	x-to-w frame flip parameter (D11=0,+1,-1)	CS FILE Row(7)
D12	y-to-w frame flip parameter (D12=0,+1,-1)	CS FILE Row(8)
D21	x-to-v frame flip parameter (D21=0,+1,-1)	CS FILE Row(9)
D22	y-to-v frame flip parameter (D22=0,+1,-1)	CS FILE Row(10)

## C APPENDIX: Useful Definitions

### C.1 Definition: Cross-Product Matrix

#### DEFINITION C.1 Cross-Product Matrix

Given a vector  $x \in \mathcal{R}^3$  with elements,

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (\text{C.1})$$

the cross product matrix  $x^\times \in \mathcal{R}^3$  is defined as follows,

$$x^\times = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \quad (\text{C.2})$$

■

### C.2 Definition: Matrix Square-Root

#### DEFINITION C.2 Matrix Square-Root

Consider a square positive-definite symmetric matrix  $P \in \mathcal{R}^{n \times n}$ . Let  $P$  be decomposed into the following form,

$$P = XX^T \quad (\text{C.3})$$

where  $X \in \mathcal{R}^{n \times n}$ . Then the matrix  $X$  is said to be the square-root of  $P$ , written as,

$$X = P^{\frac{1}{2}} \quad (\text{C.4})$$

■

It is noted that transposed matrix is on the right in the product (C.3). Substituting (C.4) into (C.3) of Definition C.2 gives  $P = P^{\frac{1}{2}}P^{\frac{T}{2}}$ , where it has been convenient to define  $P^{\frac{T}{2}} \triangleq (P^{\frac{1}{2}})^T$ .

## D APPENDIX: Useful Lemmas

This Appendix contains a summary of various results used in the estimator design, and is included to help make the report self-contained.

### D.1 Kalman Filter Update - Array Square Root Method

#### LEMMA D.1 Array Square-Root Relations

Consider the unitary triangularization of the matrix

$$\begin{bmatrix} R^{\frac{1}{2}} & HM^{\frac{1}{2}} \\ 0 & M^{\frac{1}{2}} \end{bmatrix} = \begin{bmatrix} X & 0 \\ Y & Z \end{bmatrix} \Theta \quad (\text{D.1})$$

where  $R = R^{\frac{1}{2}}R^{\frac{T}{2}} > 0$ ,  $M = M^{\frac{1}{2}}M^{\frac{T}{2}}$ , and  $\Theta$  is an orthogonal matrix (i.e.,  $\Theta^T\Theta = I$ ). Then,

$$\text{(i)} \quad X = (HMH^T + R)^{\frac{1}{2}} \quad (\text{D.2})$$

$$\text{(ii)} \quad Y = MH (HMH^T + R)^{-\frac{T}{2}} \quad (\text{D.3})$$

$$\text{(iii)} \quad YX^{-1} = MH (HMH^T + R)^{-1} \quad (\text{D.4})$$

$$\text{(iv)} \quad Z = \left( M - MH^T (HMH^T + R)^{-1} HM \right)^{\frac{1}{2}} \quad (\text{D.5})$$

**Proof:** Squaring up both sides of (D.1) and using  $\Theta^T\Theta = I$  yields,

$$\begin{bmatrix} R^{\frac{1}{2}} & HM^{\frac{1}{2}} \\ 0 & M^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} R^{\frac{T}{2}} & 0 \\ M^{\frac{T}{2}}H^T & M^{\frac{T}{2}} \end{bmatrix} = \begin{bmatrix} X & 0 \\ Y & Z \end{bmatrix} \begin{bmatrix} X^T & Y^T \\ 0 & Z^T \end{bmatrix} \quad (\text{D.6})$$

$$\begin{bmatrix} R + HMH^T & HM \\ MH^T & M \end{bmatrix} = \begin{bmatrix} XX^T & XY^T \\ YX^T & YY^T + ZZ^T \end{bmatrix} \quad (\text{D.7})$$

Inspection of (D.7) yields,

$$XX^T = HMH^T + R \quad (\text{D.8})$$

$$YX^T = MH^T \quad (\text{D.9})$$

$$YY^T + ZZ^T = M \quad (\text{D.10})$$

The proof of (i) follows by inspection of (D.8). The proof of (ii) follows by post-multiplying (D.9) by  $X^{-T}$  using the form of  $X$  given in (i). The proof of (iii) follows by rearranging (D.10) and using the form of  $Y$  given in (ii) to yield,

$$ZZ^T = M - YY^T \quad (\text{D.11})$$

$$= M - MH^T (HMH^T + R)^{-\frac{T}{2}} (HMH^T + R)^{-\frac{1}{2}} HM \quad (\text{D.12})$$

$$= M - MH^T \left[ (HMH^T + R)^{\frac{1}{2}} (HMH^T + R)^{\frac{T}{2}} \right]^{-1} HM \quad (\text{D.13})$$

$$= M - MH^T (HMH^T + R)^{-1} HM \quad (\text{D.14})$$

### COROLLARY D.1 KF Measurement Update (Array Square Root Method)

Consider the state-space measurement equation,

$$y = Hx + v \quad (\text{D.15})$$

where  $R = \text{Cov}(v)$  is the noise covariance matrix,  $H$  is the observation matrix and  $M = \text{Cov}(x)$  is the a-priori state covariance matrix. Define the square-root Kalman filter quantities  $(K, \Omega^{\frac{1}{2}}, P^{\frac{1}{2}})$  as follows,

$$K \triangleq MH(HMH^T + R)^{-1} \quad (\text{Kalman Gain}) \quad (\text{D.16})$$

$$\Omega^{\frac{1}{2}} \triangleq (HMH^T + R)^{\frac{1}{2}} \quad (\text{Square-root innovations covariance}) \quad (\text{D.17})$$

$$P^{\frac{1}{2}} \triangleq \left( M - MH^T (HMH^T + R)^{-1} HM \right)^{\frac{1}{2}} \quad (\text{Square-root A-posteriori covariance}) \quad (\text{D.18})$$

Let  $(H, R^{\frac{1}{2}}, M^{\frac{1}{2}})$  be used in the unitary triangularization (D.1) of Lemma D.1 to define the square-root factors  $X, Y, Z$ . Then,

$$K = YX^{-1} \quad (\text{D.19})$$

$$\Omega^{\frac{1}{2}} = X \quad (\text{D.20})$$

$$P^{\frac{1}{2}} = Z \quad (\text{D.21})$$

**Proof:** Equation (D.19) follows by inspection of result (iii) of Lemma D.1; equation (D.20) follows by result (i) of Lemma D.1; and equation (D.21) follows by inspection of result (iv) of Lemma D.1.

## D.2 Gyro Perturbation Lemma

Define a nominal rate vector  $\omega_m^\circ \in \mathcal{R}^3$  according to the construction,

$$\omega_m^\circ = w_m + b_g^\circ + c_g^\circ t \quad (\text{D.22})$$

where  $w_m \in \mathcal{R}^3$  denotes the measured rate, and  $b_g^\circ, c_g^\circ \in \mathcal{R}^3$  are nominal estimates of the 3-axis gyro bias and bias rate parameters, respectively. Let the true rate  $\omega \in \mathcal{R}^3$  be related to the nominal rate according to,

$$\omega = \omega_m^\circ + b_g + c_g t \quad (\text{D.23})$$

where the parameters  $b_g, c_g \in \mathcal{R}^3$  denote a correction factor applied to the nominal. Combining (D.22) and (D.23), the total correction to the measured rate is given by,

$$\omega = w_m + (b_g^\circ + b_g) + (c_g^\circ + c_g)t \quad (\text{D.24})$$

which is the sum of the nominal parameter estimates and correction factors. However, for the purpose of this report, it is convenient to keep steps (D.22) and (D.23) separate, and consider the correction as being applied in two distinct stages.

### LEMMA D.2 Gyro Perturbation

Let the nominal gyro offset  $G^\circ$  be defined by integrating the differential equation,

$$\left( \dot{G}^\circ = -(\omega_m^\circ)^\times G^\circ \right) \Big|_{t_j}^{T_k(j)} \quad (\text{D.25})$$

starting with the initial condition  $G^\circ(t_j) = I$ , where  $\omega_m^\circ$  is a nominal rate estimate (e.g., of the form (D.22)). Here, the notation  $(\cdot)|_a^b$  denotes integration over the time interval  $t \in [a, b]$ .

Let the true gyro offset  $G$  be defined by integrating the differential equation,

$$\left( \dot{G} = -\omega^\times G \right) \Big|_{t_j}^{T_k(j)} \quad (\text{D.26})$$

starting with the initial condition  $G(t_j) = I$ , where the true rate  $\omega$  is related to the nominal rate estimate by,

$$\omega = \omega_m^\circ + b_g + c_g t \quad (\text{D.27})$$

If the angle/axis perturbation  $\gamma(t) \in \mathcal{R}^3$  between  $G$  and  $G^\circ$  is sufficiently small such that one can write,

$$G(t) = (I - \gamma(t)^\times) G(t)^\circ \quad (\text{D.28})$$

then,

$$\gamma(T_k) = H_g(T_k) \begin{bmatrix} b_g \\ c_g \end{bmatrix} \quad (\text{D.29})$$

where

$$H_g(T_k) = [ \Lambda_b(T_k) \quad \Lambda_c(T_k) ] \quad (\text{D.30})$$

and

$$\left( \dot{\Lambda}_b = -(\omega_m^\circ)^\times \Lambda_b + I \right) \Big|_{t_j}^{T_k(j)} \quad \text{with I.C.} \quad \Lambda_b(t_j) = 0 \quad (\text{D.31})$$

$$\left( \dot{\Lambda}_c = -(\omega_m^\circ)^\times \Lambda_c + t \cdot I \right) \Big|_{t_j}^{T_k(j)} \quad \text{with I.C.} \quad \Lambda_c(t_j) = 0 \quad (\text{D.32})$$

**Proof:** Given equations (D.25), (D.26), (D.28), the *Attitude Linearization Lemma D.4* states that,

$$\dot{\gamma} = \gamma^\times \omega_m^\circ - \Delta \quad (\text{D.33})$$

where,

$$\Delta = \omega_m^\circ - \omega \quad (\text{D.34})$$

However, in light of (D.27) one has,

$$\Delta = \omega_m^\circ - \omega = -b_g - c_g t \quad (\text{D.35})$$

Combining (D.33) and (D.35) gives upon rearranging,

$$\dot{\gamma} = -(\omega_m^\circ)^\times \gamma + b_g + c_g t \quad (\text{D.36})$$

This is a *Linear Time-Varying* differential equation whose solution is in general the sum of a zero-input response and a zero-state response. Since the initial state is zero (i.e., since  $G(t_j) = G^\circ(t_j) = I \Rightarrow \gamma(t_j) = 0$ ), the solution only depends on the inputs. By *linear superposition* the solution can be written as,

$$\gamma(T_k) = H_g(T_k) \begin{bmatrix} b_g \\ c_g \end{bmatrix} \quad (\text{D.37})$$

where,

$$H_g = [\Lambda_b, \Lambda_c] \quad (\text{D.38})$$

$$\Lambda_b = \frac{\partial \gamma}{\partial b_g}, \quad \Lambda_c = \frac{\partial \gamma}{\partial c_g} \quad (\text{D.39})$$

More specifically, if  $e_i$  denotes a unit vector with a “1” in  $i^{\text{th}}$  place and zeros elsewhere, the  $i^{\text{th}}$  column of the sensitivity matrices  $\Lambda_b$  and  $\Lambda_c$  can be obtained by integrating,

$$\left(\dot{\gamma}_{bi} = -(\omega_m^\circ)^\times \gamma_{bi} + e_i\right)\Big|_{t_j}^{T_k(j)} \quad (\text{D.40})$$

$$\left(\dot{\gamma}_{ci} = -(\omega_m^\circ)^\times \gamma_{ci} + t \cdot e_i\right)\Big|_{t_j}^{T_k(j)} \quad (\text{D.41})$$

with initial conditions  $\gamma_{bi}(t_j) = 0, \gamma_{ci}(t_j) = 0$ . In matrix form, equations (D.40)(D.41) are identical to (D.31) and (D.32).  $\blacksquare$

### D.3 Measurement Equation Lemma

#### LEMMA D.3 Measurement Equation

Let  $q$  denote the true quaternion, let  $q_m$  denote a noisy measurement which is close to  $q$ , and let  $\hat{q}$  be a nominal estimated quaternion which is assumed to be close to both  $q$  and  $q_m$ . Given  $q, q_m, \hat{q}$ , let the three small rotational perturbations  $y, \theta, v \in \mathcal{R}^3$  be defined (to first-order), according to the following expressions,

$$C(q_m) = (I - y^\times) C(\hat{q}) \quad (\text{D.42})$$

$$C(q_m) = (I - v^\times) C(q) \quad (\text{D.43})$$

$$C(q) = (I - \theta^\times) C(\hat{q}) \quad (\text{D.44})$$

(Note that  $y$  can be interpreted as an incremental measurement,  $\theta$  an incremental estimation error, and  $v$  an incremental measurement noise). Then to first order,

$$y = \theta + v \quad (\text{D.45})$$

**Proof:** Equating (D.42), (D.43) and (D.44) gives,

$$(I - y^\times) C(\hat{q}) = (I - v^\times) C(q) = (I - v^\times) (I - \theta^\times) C(\hat{q}) \quad (\text{D.46})$$

Right multiplying (D.46) by  $C(\hat{q})^T$  (noting that  $C(\hat{q}) C(\hat{q})^T = I$  by orthogonality), yields,

$$(I - y^\times) = (I - v^\times) (I - \theta^\times) \quad (\text{D.47})$$

$$= (I - (v + \theta)^\times) + (\text{high-order terms}) \quad (\text{D.48})$$

Subtracting the identity matrix from both sides of (D.48) and rearranging, gives to first-order,

$$y^\times = (v + \theta)^\times \quad (\text{D.49})$$

which is equivalent to (D.45), as desired.  $\blacksquare$

## D.4 Attitude Linearization Lemma

### LEMMA D.4 Attitude Linearization

Let the true attitude matrix  $C$  propagate according to the kinematic equation,

$$\dot{C} = -\omega^\times C \quad (\text{D.50})$$

where  $\omega \in \mathcal{R}^3$  denotes the true angular rate vector. Let an approximation  $\hat{C}$  to  $C$  be constructed according to the following propagation equation,

$$\dot{\hat{C}} = -\hat{\omega}^\times \hat{C} \quad (\text{D.51})$$

where  $\hat{\omega} \in \mathcal{R}^3$  differs from the true angular rate vector  $\omega$  by the amount,

$$\Delta = \hat{\omega} - \omega \quad (\text{D.52})$$

If the angle/axis perturbation  $\theta \in \mathcal{R}^3$  between  $C$  and  $\hat{C}$  is sufficiently small such that one can write,

$$C = (I - \theta^\times) \hat{C} \quad (\text{D.53})$$

then to first order,  $\theta$  propagates according to,

$$\dot{\theta} = \theta^\times \hat{\omega} - \Delta \quad (\text{D.54})$$

**Proof:** Taking the time-derivative of both sides of (D.53) gives,

$$\dot{C} = (I - \theta^\times) \dot{\hat{C}} - \dot{\theta}^\times \hat{C} \quad (\text{D.55})$$

Substituting (D.50) and (D.51) into (D.55) yields,

$$-\omega^\times C = (I - \theta^\times) (-\hat{\omega}^\times \hat{C}) - \dot{\theta}^\times \hat{C} \quad (\text{D.56})$$

Right-multiplying both sides of (D.56) by  $\hat{C}^T$  yields,

$$-\omega^\times C \hat{C}^T = (I - \theta^\times) [-\hat{\omega}^\times] - \dot{\theta}^\times \quad (\text{D.57})$$

Solving for  $\dot{\theta}^\times$  gives upon rearranging,

$$\dot{\theta}^\times = \omega^\times C \hat{C}^T + (I - \theta^\times) [-\hat{\omega}^\times] \quad (\text{D.58})$$

$$= \omega^\times (I - \theta^\times) - (I - \theta^\times) \hat{\omega}^\times \quad (\text{D.59})$$

$$= (\hat{\omega} - \Delta)^\times (I - \theta^\times) - (I - \theta^\times) \hat{\omega}^\times \quad (\text{D.60})$$

$$= \hat{\omega}^\times - \hat{\omega}^\times \theta^\times - \Delta^\times + \Delta^\times \theta^\times - \hat{\omega}^\times + \theta^\times \hat{\omega}^\times \quad (\text{D.61})$$

$$= \theta^\times \hat{\omega}^\times - \hat{\omega}^\times \theta^\times - \Delta^\times + \Delta^\times \theta^\times \quad (\text{D.62})$$

$$= [\theta^\times \hat{\omega}]^\times - \Delta^\times + \Delta^\times \theta^\times \quad (\text{D.63})$$

$$\simeq [\theta^\times \hat{\omega}]^\times - \Delta^\times \quad (\text{D.64})$$

Here, equation (D.59) follows by substituting  $C \hat{C}^T = I - \theta^\times$  (obtained by rearranging (D.53)); equation (D.60) follows by substituting  $\omega = \hat{\omega} - \Delta$  (obtained from (D.52)); equation (D.61) follows by expanding all expressions; equation (D.62) follows by cancelling terms and rearranging; equation (D.63) follows by applying the identity  $\theta^\times \hat{\omega}^\times - \hat{\omega}^\times \theta^\times = [\theta^\times \hat{\omega}]^\times$  from the *Matrix to Vector I Lemma* D.7; and equation (D.64) follows by dropping second-order terms. Equation (D.64) can be re-written in the form,

$$\dot{\theta} = \theta^\times \hat{\omega} - \Delta \quad (\text{D.65})$$

which is the desired expression (D.54). ■

## D.5 Angle-Axis Perturbation Lemma

### LEMMA D.5 Angle-Axis Perturbation

Let the directional cosine matrix  $C$  be parameterized in angle/axis form as follows,

$$C(a, \phi) \triangleq \cos \phi \cdot I + (1 - \cos \phi) aa^T - \sin \phi a^\times \quad (\text{D.66})$$

$$a^T a = 1 \quad (\text{D.67})$$

where  $a \in \mathcal{R}^3$  is the rotation axis and  $\phi$  is the scalar rotation angle. Define a perturbed directional cosine matrix as,

$$\hat{C} = C(\hat{a}, \hat{\phi}) \triangleq \cos \hat{\phi} \cdot I + (1 - \cos \hat{\phi}) \hat{a} \hat{a}^T - \sin \hat{\phi} \hat{a}^\times \quad (\text{D.68})$$

where  $\hat{a}$  and  $\hat{\phi}$  are the perturbed values of the angle/axis parameters, i.e. satisfying,

$$a = \hat{a} + \delta a, \quad (\text{D.69})$$

$$\phi = \hat{\phi} + \delta\phi \quad (\text{D.70})$$

$$\hat{a}^T \hat{a} = 1 \quad (\text{D.71})$$

Then,

(i) the matrix  $C$  can be expanded to first-order as,

$$C(a, \phi) = C(\hat{a} + \delta a, \hat{\phi} + \delta\phi) = \hat{C} + H_\phi + H_a \quad (\text{D.72})$$

where,

$$H_\phi \triangleq s(\hat{a}\hat{a}^T - I)\delta\phi - c\hat{a}^\times \delta\phi \quad (\text{D.73})$$

$$H_a \triangleq (1 - c)(\delta a \hat{a}^T + \hat{a} \delta a^T) - s\delta a^\times \quad (\text{D.74})$$

$$s \triangleq \sin(\hat{\phi}), \quad c \triangleq \cos(\hat{\phi}) \quad (\text{D.75})$$

(ii) If the mapping from  $\hat{C}$  to  $C$  is sufficiently small such that there exists a  $\theta \in \mathcal{R}^3$  satisfying,

$$C = (I - \theta^\times) \hat{C}, \quad (\text{D.76})$$

then,

$$\theta = \left[ \sin \hat{\phi} \cdot I - (1 - \cos \hat{\phi}) \hat{a}^\times \right] \delta a + \hat{a} \delta\phi \quad (\text{D.77})$$

**Proof of (i):** Substituting the perturbations (D.69) (D.70) into (D.66) and expanding gives,

$$C(a, \phi) = C(\hat{a} + \delta a, \hat{\phi} + \delta\phi) \quad (\text{D.78})$$

$$\begin{aligned} &= \cos(\hat{\phi} + \delta\phi) \cdot I + \left(1 - \cos(\hat{\phi} + \delta\phi)\right) (\hat{a} + \delta a)(\hat{a} + \delta a)^T \\ &\quad - \sin(\hat{\phi} + \delta\phi)(\hat{a} + \delta a)^\times \end{aligned} \quad (\text{D.79})$$

$$\begin{aligned} &\simeq (c - s\delta\phi) \cdot I + (1 - c + s\delta\phi)\hat{a}\hat{a}^T + (1 - c + s\delta\phi)(\delta a \hat{a}^T + \hat{a} \delta a^T) \\ &\quad - (s + c\delta\phi)\hat{a}^\times - (s + c\delta\phi)\delta a^\times \end{aligned} \quad (\text{D.80})$$

$$\begin{aligned} &\simeq (c - s\delta\phi) \cdot I + (1 - c + s\delta\phi)\hat{a}\hat{a}^T + (1 - c)(\delta a \hat{a}^T + \hat{a} \delta a^T) \\ &\quad - (s + c\delta\phi)\hat{a}^\times - s\delta a^\times \end{aligned} \quad (\text{D.81})$$

$$\begin{aligned} &= \left[ c \cdot I + (1 - c)\hat{a}\hat{a}^T - s\hat{a}^\times \right] + \left[ s(\hat{a}\hat{a}^T - I)\delta\phi - c\hat{a}^\times \delta\phi \right] \\ &\quad + \left[ (1 - c)(\delta a \hat{a}^T + \hat{a} \delta a^T) - s\delta a^\times \right] \end{aligned} \quad (\text{D.82})$$

$$= \hat{C} + H_\phi + H_a \quad (\text{D.83})$$

Here (D.80) follows from (D.79) by using the first-order trigonometric expansions  $\cos(\widehat{\phi} + \delta\phi) \simeq c - s\delta\phi$ ,  $\sin(\widehat{\phi} + \delta\phi) \simeq s + c\delta\phi$ , and eliminating the second-order term involving  $\delta a \delta a^T$ ; equation (D.81) follows by eliminating second-order terms in (D.80); equation (D.82) follows by regrouping terms in (D.81); and equation (D.83) follows by substituting definitions (D.68) (D.73) (D.74) into (D.82). Equation (D.83) is equivalent to (D.72) which proves the desired relation.

**Proof of (ii):** Expanding (D.76) gives,

$$C = (I - \theta^\times)\widehat{C} = \widehat{C} - \theta^\times\widehat{C} \quad (\text{D.84})$$

Equating (D.84) with (D.72) and rearranging yields,

$$-\theta^\times\widehat{C} = H_\phi + H_a \quad (\text{D.85})$$

or equivalently,

$$\theta^\times = -H_\phi\widehat{C}^T - H_a\widehat{C}^T \quad (\text{D.86})$$

It will be convenient to examine each term on the RHS of (D.86) separately, in the following two cases.

**Case I:** In this case, a simple expression is sought for the term,

$$H_\phi\widehat{C}^T \quad (\text{D.87})$$

The scalar term  $\delta\phi$  in the expression (D.73) for  $H_\phi$  can be brought outside the brackets to give,

$$H_\phi = \left[ s(\widehat{a}\widehat{a}^T - I) - c\widehat{a}^\times \right] \delta\phi \quad (\text{D.88})$$

The transpose of  $\widehat{C}$  can be taken in the expression (D.68) to give,

$$\widehat{C}^T(\widehat{a}, \widehat{\phi}) = c \cdot I + (1 - c)\widehat{a}\widehat{a}^T + s\widehat{a}^\times \quad (\text{D.89})$$

Substituting (D.88) and (D.89) into the expression (D.87) gives,

$$H_\phi \widehat{C}^T = \left[ s(\widehat{a}\widehat{a}^T - I) - c\widehat{a}^\times \right] \delta\phi \left[ c \cdot I + (1-c)\widehat{a}\widehat{a}^T + s\widehat{a}^\times \right] \quad (\text{D.90})$$

$$\begin{aligned} &= \left[ cs(\widehat{a}\widehat{a}^T - I) - c^2\widehat{a}^\times + s(1-c)(\widehat{a}\widehat{a}^T\widehat{a}\widehat{a}^T - \widehat{a}\widehat{a}^T) - c(1-c)\widehat{a}^\times\widehat{a}\widehat{a}^T \right. \\ &\quad \left. + s^2(\widehat{a}\widehat{a}^T\widehat{a}^\times - \widehat{a}^\times) - sc\widehat{a}^\times\widehat{a}^\times \right] \delta\phi \end{aligned} \quad (\text{D.91})$$

$$= \left[ cs(\widehat{a}\widehat{a}^T - I) - c^2\widehat{a}^\times - s^2\widehat{a}^\times - sc\widehat{a}^\times\widehat{a}^\times \right] \delta\phi \quad (\text{D.92})$$

$$= \left[ cs\widehat{a}^\times\widehat{a}^\times - (c^2 + s^2)\widehat{a}^\times - sc\widehat{a}^\times\widehat{a}^\times \right] \delta\phi \quad (\text{D.93})$$

$$= -\widehat{a}^\times \delta\phi \quad (\text{D.94})$$

Here, (D.91) follows by expanding (D.90); equation (D.92) follows by using the vector identities  $xx^Txx^T = xx^T$  and  $x^\times x = 0$  with the choice  $x = \widehat{a}$ ; equation (D.93) follows by using the vector identity  $xx^T - I = x^\times x^\times$  (cf., Lemma D.11) with the choice  $x = \widehat{a}$ ; and (D.94) follows by cancelling terms and using the well-known relation  $s^2 + c^2 = 1$ . In summary,

$$H_\phi \widehat{C}^T = -\widehat{a}^\times \delta\phi \quad (\text{D.95})$$

**Case II:** In this second case, a simple expression is sought for the term,

$$H_a \widehat{C}^T \quad (\text{D.96})$$

Substituting (D.74) and (D.68) into (D.96) yields,

$$H_a \widehat{C}^T = \left[ (1-c)(\delta a \widehat{a}^T + \widehat{a} \delta a^T) - s \delta a^\times \right] \left[ c \cdot I + (1-c) \widehat{a} \widehat{a}^T + s \widehat{a}^\times \right] \quad (\text{D.97})$$

$$\begin{aligned} &= c(1-c)(\widehat{a} \delta a^T + \delta a \widehat{a}^T) - cs \delta a^\times \\ &\quad + (1-2c+c^2) \delta a \widehat{a}^T - s(1-c) \delta a^\times \widehat{a} \widehat{a}^T \\ &\quad + s(1-c)(\delta a \widehat{a}^T \widehat{a}^\times + \widehat{a} \delta a^T \widehat{a}^\times) - s^2 \delta a^\times \widehat{a}^\times \end{aligned} \quad (\text{D.98})$$

$$\begin{aligned} &= c(\widehat{a} \delta a^T + \delta a \widehat{a}^T) - c^2 \widehat{a} \delta a^T - c^2 \delta a \widehat{a}^T - cs \delta a^\times \\ &\quad + \delta a \widehat{a}^T - 2c \delta a \widehat{a}^T + c^2 \delta a \widehat{a}^T \\ &\quad + s(1-c) \left( \widehat{a} \delta a^T \widehat{a}^\times - [\widehat{a} \delta a^T \widehat{a}^\times]^T \right) - s^2 \widehat{a} \delta a^T \end{aligned} \quad (\text{D.99})$$

$$\begin{aligned} &= c[\widehat{a} \delta a^T - \delta a \widehat{a}^T] - (c^2 + s^2) \widehat{a} \delta a^T - cs \delta a^\times + \delta a \widehat{a}^T \\ &\quad + s(1-c) \left( \widehat{a} \delta a^T \widehat{a}^\times - [\widehat{a} \delta a^T \widehat{a}^\times]^T \right) \end{aligned} \quad (\text{D.100})$$

$$= c[-(\widehat{a}^\times \delta a)^\times] + (\delta a \widehat{a}^T - \widehat{a} \delta a^T) - cs \delta a^\times + s(1-c)(-\delta a^\times) \quad (\text{D.101})$$

$$= (1-c)(\widehat{a}^\times \delta a)^\times - cs \delta a^\times - s(1-c) \delta a^\times \quad (\text{D.102})$$

$$= (1-c)(\widehat{a}^\times \delta a)^\times - s \delta a^\times \quad (\text{D.103})$$

$$= [((1-c) \widehat{a}^\times - s \cdot I) \delta a]^\times \quad (\text{D.104})$$

Here, (D.98) follows by expanding terms and using the relations  $\delta a^T \widehat{a} = 0$  (from Lemma D.6) and  $\widehat{a}^T \widehat{a} = 1$  (since  $\widehat{a}$  is a unit vector); equation (D.99) follows by applying  $\widehat{a}^T \widehat{a}^\times = 0$ , and using the relations  $\delta a^\times \widehat{a} \widehat{a}^T = [\widehat{a} \delta a^T \widehat{a}^\times]^T$  (from Lemma D.10), and  $\delta a^\times \widehat{a}^\times = \widehat{a} \delta a^T$  (from Lemma D.9); equation (D.100) follows by cancelling the  $c^2 \delta a \widehat{a}^T$  terms and rearranging the remaining terms; equation (D.101) follows by using the relations  $\widehat{a} \delta a^T - \delta a \widehat{a}^T = -(\widehat{a}^\times \delta a)^\times$  (M2V Lemma D.8) and  $\widehat{a} \delta a^T \widehat{a}^\times - [\widehat{a} \delta a^T \widehat{a}^\times]^T = -\delta a^\times$  (cf., Lemma D.13), and the applying the trigonometric relation  $c^2 + s^2 = 1$ ; equation (D.102) follows by using the relation  $\delta a \widehat{a}^T - \widehat{a} \delta a^T = (\widehat{a}^\times \delta a)^\times$  (M2V Lemma D.8) and rearranging; equation (D.103) follows by cancelling terms; and (D.104) follows by extracting the common term  $\delta a$  to the right, before taking the cross-product.

In summary,

$$H_a \widehat{C}^T = [((1-c) \widehat{a}^\times - s \cdot I) \delta a]^\times \quad (\text{D.105})$$

**Combining Case I and Case II results:**

Substituting Case I result (D.95) and Case II result (D.105) into (D.86) gives,

$$\theta^\times = -H_a \widehat{C}^T - H_\phi \widehat{C}^T \quad (\text{D.106})$$

$$= -\left[ (1-c)\widehat{a}^\times - sI \right] \delta a^\times + \widehat{a}^\times \delta \phi \quad (\text{D.107})$$

$$= \left[ (sI - (1-c)\widehat{a}^\times) \delta a + \widehat{a} \delta \phi \right]^\times \quad (\text{D.108})$$

Dropping the crossproduct from each side of (D.108) (note, this is equivalent to equating the corresponding non-zero elements), gives the relation,

$$\theta = (sI - (1-c)\widehat{a}^\times) \delta a + \widehat{a} \delta \phi \quad (\text{D.109})$$

which is the desired expression (D.77). ■

## D.6 Unit Vector Perturbation Lemma

### LEMMA D.6 Unit Vector Perturbation

Let the unit vectors  $a$  and  $\widehat{a}$  be related to each other by a small perturbation vector  $\delta a$  where,

$$a = \widehat{a} + \delta a \quad (\text{D.110})$$

$$a^T a = 1 \quad (\text{D.111})$$

$$\widehat{a}^T \widehat{a} = 1 \quad (\text{D.112})$$

Then the following relations hold to first-order,

$$\delta a^T \widehat{a} = 0 \quad (\text{D.113})$$

$$\delta a^T a = 0 \quad (\text{D.114})$$

**Proof:**

$$a^T a = (\widehat{a} + \delta a)^T (\widehat{a} + \delta a) \quad (\text{D.115})$$

$$= \widehat{a}^T \widehat{a} + 2\delta a^T \widehat{a} + \delta a^T \delta a \quad (\text{D.116})$$

Substituting the unit vector constraints (D.111)(D.112) into (D.116) gives upon rearranging,

$$2\delta a^T \widehat{a} + \delta a^T \delta a = 0 \quad (\text{D.117})$$

or equivalently,

$$\delta a^T \widehat{a} = 0 + \mathcal{O}(\|\delta a\|^2) \quad (\text{D.118})$$

Hence, relation (D.113) holds to first order, as desired. Expression (D.114) is proved similarly, by expanding  $\widehat{a}^T \widehat{a} = (a - \delta a)^T (a - \delta a)$ . ■

## D.7 Cross Product Operations Lemma: Matrix to Vector I

### LEMMA D.7 Cross Product Operations: Matrix to Vector I

Let  $x, a$  and  $b$  be vectors in  $\mathcal{R}^3$ .

If the following identity holds,

$$x^\times = a^\times b^\times - b^\times a^\times \quad (\text{D.119})$$

then,

$$x = a \times b \quad (\text{D.120})$$

**Proof:** Let  $x, a, b \in \mathcal{R}^3$  have the following components,

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}; \quad a = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}; \quad b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad (\text{D.121})$$

Expanding  $a^\times b^\times - b^\times a^\times$  componentwise gives,

$$a^\times b^\times = \begin{bmatrix} -a_3 b_3 - a_2 b_2 & a_2 b_1 & a_3 b_1 \\ a_1 b_2 & -a_3 b_3 - a_1 b_1 & a_3 b_2 \\ a_1 b_3 & a_2 b_3 & -a_2 b_2 - a_1 b_1 \end{bmatrix} \quad (\text{D.122})$$

$$b^\times a^\times = \begin{bmatrix} -a_3 b_3 - b_2 a_2 & a_1 b_2 & a_1 b_3 \\ a_2 b_1 & -a_3 b_3 - a_1 b_1 & a_2 b_3 \\ a_3 b_1 & a_3 b_2 & -a_2 b_2 - a_1 b_1 \end{bmatrix} \quad (\text{D.123})$$

$$a^\times b^\times - b^\times a^\times = \begin{bmatrix} 0 & -(a_1 b_2 - a_2 b_1) & a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 & 0 & -(a_2 b_3 - a_3 b_2) \\ -(a_3 b_1 - a_1 b_3) & a_2 b_3 - a_3 b_2 & 0 \end{bmatrix} \quad (\text{D.124})$$

Likewise, expanding  $(a \times b)^\times$  componentwise gives,

$$(a \times b)^\times = \begin{bmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{bmatrix}^\times = \begin{bmatrix} 0 & -(a_1 b_2 - a_2 b_1) & a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 & 0 & -(a_2 b_3 - a_3 b_2) \\ -(a_3 b_1 - a_1 b_3) & a_2 b_3 - a_3 b_2 & 0 \end{bmatrix} \quad (\text{D.125})$$

Noting that (D.124) and (D.125) are componentwise identical, they can be equated to give the identity,

$$x^\times = a^\times b^\times - b^\times a^\times \quad (\text{D.126})$$

where  $x = a \times b$ . This is exactly the desired result (D.120). ■

## D.8 Cross Product Operations Lemma: Matrix to Vector II

### LEMMA D.8 Cross Product Operations: Matrix to Vector II

Let  $x, a$  and  $b$  be vectors in  $\mathcal{R}^3$ .

If the following identity holds,

$$x^\times = ab^T - ba^T \quad (\text{D.127})$$

then,

$$x = b \times a \quad (\text{D.128})$$

**Proof:** Let  $x, a, b \in \mathcal{R}^3$  have the following components,

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}; \quad a = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}; \quad b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad (\text{D.129})$$

Expanding  $ab^T - ba^T$  componentwise gives,

$$ab^T - ba^T = \begin{bmatrix} a_1b_1 & a_1b_2 & a_1b_3 \\ a_2b_1 & a_2b_2 & a_2b_3 \\ a_3b_1 & a_3b_2 & a_3b_3 \end{bmatrix} - \begin{bmatrix} a_1b_1 & a_2b_1 & a_3b_1 \\ a_1b_2 & a_2b_2 & a_3b_2 \\ a_1b_3 & a_2b_3 & a_3b_3 \end{bmatrix} \quad (\text{D.130})$$

$$= \begin{bmatrix} 0 & a_1b_2 - a_2b_1 & a_1b_3 - a_3b_1 \\ a_2b_1 - a_1b_2 & 0 & a_2b_3 - a_3b_2 \\ a_3b_1 - a_1b_3 & a_3b_2 - a_2b_3 & 0 \end{bmatrix} \quad (\text{D.131})$$

$$= \begin{bmatrix} 0 & -(a_2b_1 - a_1b_2) & a_1b_3 - a_3b_1 \\ a_2b_1 - a_1b_2 & 0 & -(a_3b_2 - a_2b_3) \\ -(a_1b_3 - a_3b_1) & a_3b_2 - a_2b_3 & 0 \end{bmatrix} \quad (\text{D.132})$$

Likewise, expanding  $(b \times a)^\times$  componentwise gives,

$$(b \times a)^\times = \begin{bmatrix} a_3b_2 - a_2b_3 \\ a_1b_3 - a_3b_1 \\ a_2b_1 - a_1b_2 \end{bmatrix}^\times = \begin{bmatrix} 0 & -(a_2b_1 - a_1b_2) & a_1b_3 - a_3b_1 \\ a_2b_1 - a_1b_2 & 0 & -(a_3b_2 - a_2b_3) \\ -(a_1b_3 - a_3b_1) & a_3b_2 - a_2b_3 & 0 \end{bmatrix} \quad (\text{D.133})$$

Noting that (D.132) and (D.133) are componentwise identical, they can be equated to give the identity,

$$x^\times = ab^T - ba^T \quad (\text{D.134})$$

where  $x = b \times a$ . This is exactly the desired result (D.128). ■

## D.9 Cross Product Operations Lemma: Matrix to Outer Product

### LEMMA D.9 Cross Product Operations: Matrix to Outer Product

If  $a$  and  $b$  are orthogonal vectors (i.e.  $a^T b = 0$ ) in  $\mathcal{R}^3$ , then the following identity holds,

$$a^\times b^\times = b a^T \quad (\text{D.135})$$

**Proof:** Let  $x \in \mathcal{R}^3$  be an arbitrary vector. Construct the vector  $y \in \mathcal{R}^3$  such that,

$$y = a^\times b^\times x \quad (\text{D.136})$$

Then,

$$y = a^\times b^\times x = a \times (b \times x) \quad (\text{D.137})$$

$$= (a^T x) b - (a^T b) x \quad (\text{D.138})$$

$$= (a^T x) b \quad (\text{D.139})$$

$$= b (a^T x) \quad (\text{D.140})$$

where (D.138) follows from the well-known vector identity  $a \times (b \times c) = (a^T c)b - (a^T b)c$ ; equation (D.139) follows from the assumed orthogonality condition  $a^T b = 0$ ; and equation (D.140) follows by writing the scalar term  $a^T x$  on the right instead of the left of the vector  $b$ .

Relation (D.140) can be summarized as,

$$a^\times b^\times x = b a^T x \quad (\text{D.141})$$

Since  $x$  in (D.141) is an arbitrary vector, it follows that,

$$a^\times b^\times = b a^T \quad (\text{D.142})$$

which is the desired result (D.135). ■

## D.10 Cross - Outer Product Lemma I

### LEMMA D.10 Cross - Outer Product Lemma I

If  $a$ ,  $b$  and  $c$  are vectors in  $\mathcal{R}^3$ , then the following identity holds,

$$a^\times b c^T = [c a^T b^\times]^T \quad (\text{D.143})$$

**Proof:** By standard matrix manipulations,

$$[ca^T b^\times]^T = (b^\times)^T (ca^T)^T \quad (\text{D.144})$$

$$= -b^\times (ac^T) \quad (\text{D.145})$$

$$= -(b^\times a)c^T \quad (\text{D.146})$$

$$= (a^\times b)c^T \quad (\text{D.147})$$

which proves the desired identity.  $\blacksquare$

## D.11 Cross - Outer Product Lemma II

### LEMMA D.11 Cross - Outer Product Lemma II

If  $a$  is a unit vector in  $\mathcal{R}^3$  (i.e.,  $a^T a = 1$ ), then the following identity holds,

$$a^\times a^\times = aa^T - I \quad (\text{D.148})$$

**Proof:** Let  $a \triangleq [a_x, a_y, a_z]^T$ . Expanding  $a^\times a^\times$  componentwise gives,

$$a^\times a^\times = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \quad (\text{D.149})$$

$$= \begin{bmatrix} -(a_z^2 + a_y^2) & a_x a_y & a_x a_z \\ a_x a_y & -(a_x^2 + a_z^2) & a_y a_z \\ a_x a_z & a_y a_z & -(a_x^2 + a_y^2) \end{bmatrix} \quad (\text{D.150})$$

Likewise, expanding  $aa^T - I$  componentwise and using the unit vector constraint  $a_x^2 + a_y^2 + a_z^2 = 1$  gives,

$$aa^T - I = \begin{bmatrix} (a_x^2 - 1) & a_x a_y & a_x a_z \\ a_y a_x & (a_y^2 - 1) & a_y a_z \\ a_z a_x & a_z a_y & (a_z^2 - 1) \end{bmatrix} \quad (\text{D.151})$$

$$= \begin{bmatrix} -(a_z^2 + a_y^2) & a_x a_y & a_x a_z \\ a_x a_y & -(a_x^2 + a_z^2) & a_y a_z \\ a_x a_z & a_y a_z & -(a_x^2 + a_y^2) \end{bmatrix} \quad (\text{D.152})$$

Noting that (D.150) and (D.152) are componentwise identical, they can be equated to give the identity (D.148), which is the desired result.  $\blacksquare$

## D.12 Cross - Outer Product Lemma III

### LEMMA D.12 Cross - Outer Product Lemma III

Let  $a$  be a unit vector in  $\mathcal{R}^3$  (i.e.,  $a^T a = 1$ ), and  $b$  be a vector in  $\mathcal{R}^3$ . If the following identity holds,

$$ab^T a^\times - [ab^T a^\times]^T = x^\times \quad (\text{D.153})$$

then,

$$x = (aa^T - I) b \quad (\text{D.154})$$

**Proof:** Construct the quantity

$$c \triangleq -a^\times b \quad (\text{D.155})$$

Taking the transpose of (D.155) gives the useful relation,

$$c^T = -(a^\times b)^T = -b^T (a^\times)^T = b^T a^\times \quad (\text{D.156})$$

Then,

$$ab^T a^\times - [ab^T a^\times]^T = ac^T - ca^T \quad (\text{D.157})$$

$$= (c^\times a)^\times \quad (\text{D.158})$$

$$= -(a^\times c)^\times \quad (\text{D.159})$$

$$= (a^\times (a^\times b))^\times = (a^\times a^\times b)^\times \quad (\text{D.160})$$

$$= [(aa^T - I) b]^\times \quad (\text{D.161})$$

where (D.157) follows by substituting (D.156); equation (D.158) follows by applying the results of Lemma D.8 (the *Matrix to Vector II Lemma*); equation (D.159) follows by reversing the order of the vectors in the crossproduct; equation (D.160) follows by substituting (D.155); and (D.161) follows by applying the results of Lemma D.11 (the *Cross - Outer Product Lemma II*). Equation (D.161) is the desired result. ■

## D.13 Cross - Outer Product Lemma IIIa

### LEMMA D.13 Cross - Outer Product Lemma IIIa

Imposing the orthogonality constraint  $a^T b = 0$  in the result of Lemma D.12 (the Cross - Outer Product Lemma III) gives the following identity,

$$ab^T a^\times - [ab^T a^\times]^T = -b^\times \quad (\text{D.162})$$

**Proof:** Starting with the result of Lemma D.12 gives upon rearranging,

$$ab^T a^\times - [ab^T a^\times]^T = [(aa^T - I)b]^\times \quad (\text{D.163})$$

$$= [aa^T b - b]^\times \quad (\text{D.164})$$

$$= -b^\times \quad (\text{D.165})$$

where (D.165) follows from the orthogonality constraint  $a^T b = 0$ .

## D.14 Push-Through Lemma

### LEMMA D.14 Push-Through Lemma

Let  $A \in \mathcal{R}^{3 \times 3}$  be a direction cosine matrix (i.e.,  $A^T = A^{-1}$  and  $\det(A) = 1$ ), and let  $x \in \mathcal{R}^3$  be an arbitrary vector. Then the following identity holds,

$$A(I - x^\times) = (I - (Ax)^\times)A \quad (\text{D.166})$$

**Proof:** Consider an arbitrary vector  $e \in \mathcal{R}^3$ . Then one has the identity,

$$A(x^\times e) = (Ax)^\times (Ae) \quad (\text{D.167})$$

since the cross product of two vectors is invariant under frame transformation. Continuing,

$$A(I - x^\times)e = Ae - A(x^\times e) \quad (\text{D.168})$$

$$= Ae - (Ax)^\times (Ae) \quad (\text{D.169})$$

$$= (I - (Ax)^\times)Ae \quad (\text{D.170})$$

Here, (D.169) follows from (D.168) by using identity (D.167). The main result (D.166) follows from the fact that (D.170) holds for arbitrary  $e$ . ■

## D.15 Directional Cosine Matrix Exponential Equivalence Lemma

### LEMMA D.15 Directional Cosine Matrix Exponential Equivalence Lemma

Let  $\theta \in \mathcal{R}^3$  be an arbitrary non-zero constant vector. Then the following relationship holds.

$$e^{\theta^\times} = C \left( \frac{-\theta}{\|\theta\|}, \|\theta\| \right) \quad (\text{D.171})$$

where the matrix  $C$  is a direction cosine matrix of the form,

$$C(a, \phi) \triangleq \cos \phi \cdot I + (1 - \cos \phi) aa^T - \sin(\phi) a^\times \quad (\text{D.172})$$

In particular, for an arbitrary nonzero constant vector  $\omega \in \mathcal{R}^3$  and scalar  $\Delta T > 0$ , the following relationship holds,

$$e^{-\omega^\times \Delta T} = C\left(\frac{\omega}{|\omega|}, |\omega| \Delta T\right) \quad (\text{D.173})$$

**Proof:** (Following the approach in [17]). Define the elements of  $\theta$  as,

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} \quad (\text{D.174})$$

Then the characteristic polynomial  $p(\lambda)$  of the matrix  $\theta^\times \in \mathcal{R}^{3 \times 3}$  can be calculated as,

$$p(\lambda) = \det(\lambda \cdot I - \theta^\times) \quad (\text{D.175})$$

$$= \det \begin{bmatrix} \lambda & \theta_3 & -\theta_2 \\ -\theta_3 & \lambda & \theta_1 \\ \theta_2 & -\theta_1 & \lambda \end{bmatrix} \quad (\text{D.176})$$

$$= \lambda(\lambda^2 + \theta_1^2) - \theta_3(-\lambda\theta_3 - \theta_1\theta_2) - \theta_2(\theta_1\theta_3 - \lambda\theta_2) \quad (\text{D.177})$$

$$= \lambda^3 + \lambda\|\theta\|^2 \quad (\text{D.178})$$

The Cayley-Hamilton Theorem ensures that a matrix ( $\theta^\times$  in this case) satisfies its own characteristic equation,

$$p(\lambda) \Big|_{\lambda=\theta^\times} = 0 \quad (\text{D.179})$$

Using (D.178) in (D.179) and rearranging yields,

$$(\theta^\times)^3 = -\|\theta\|^2 \theta^\times \quad (\text{D.180})$$

This key identity is used in the next step.

Consider the power series expansion,

$$e^{\theta^\times} = I + \theta^\times + \frac{(\theta^\times)^2}{2!} + \frac{(\theta^\times)^3}{3!} + \frac{(\theta^\times)^4}{4!} + \frac{(\theta^\times)^5}{5!} + \frac{(\theta^\times)^6}{6!} + \dots \quad (\text{D.181})$$

$$= I + \theta^\times + \frac{(\theta^\times)^2}{2!} - \frac{\|\theta\|^2 \theta^\times}{3!} \quad (\text{D.182})$$

$$- \frac{\|\theta\|^2 (\theta^\times)^2}{4!} + \frac{\|\theta\|^4 \theta^\times}{5!} + \frac{\|\theta\|^4 (\theta^\times)^2}{6!} + \dots \quad (\text{D.183})$$

$$= I + \frac{\theta^\times}{\|\theta\|} \left( \|\theta\| - \frac{\|\theta\|^3}{3!} + \frac{\|\theta\|^5}{5!} - \dots \right) \quad (\text{D.184})$$

$$+ \frac{(\theta^\times)^2}{\|\theta\|^2} \left( \frac{\|\theta\|^2}{2!} - \frac{\|\theta\|^4}{4!} + \frac{\|\theta\|^6}{6!} - \dots \right) \quad (\text{D.185})$$

$$= I + \frac{\sin \|\theta\|}{\|\theta\|} \theta^\times + (1 - \cos \|\theta\|) \frac{(\theta^\times)^2}{\|\theta\|^2} \quad (\text{D.186})$$

$$= I + \frac{\sin \|\theta\|}{\|\theta\|} \theta^\times + (1 - \cos \|\theta\|) \left( \frac{\theta \theta^T}{\|\theta\|^2} - I \right) \quad (\text{D.187})$$

$$= I \cdot \cos \|\theta\| + \frac{1 - \cos \|\theta\|}{\|\theta\|^2} \theta \theta^T + \frac{\sin \|\theta\|}{\|\theta\|} \theta^\times \quad (\text{D.188})$$

$$= C \left( \frac{-\theta}{\|\theta\|}, \|\theta\| \right) \quad (\text{D.189})$$

Here, equation (D.183) follows by using identity (D.180) and rearranging; equation (D.186) follows by recognizing the Taylor expansions for  $\sin x = x - x^3/3! + x^5/5! - \dots$  and  $\cos x = 1 - x^2/2! + x^4/4! - \dots$ ; equation (D.187) follows by applying the result of Lemma D.11 with the choice  $a = \theta/\|\theta\|$ ; equation (D.188) follows by rearranging; and (D.189) follows by applying the definition (D.172). Equation (D.189) is the desired expression (D.171).

Letting  $\theta \triangleq -\omega \Delta T$  in (D.171) gives the desired expression (D.173). ■

## D.16 Directional Cosine Matrix Integral Lemma

### LEMMA D.16 Directional Cosine Matrix Integral

*For an arbitrary nonzero constant vector  $\omega \in \mathcal{R}^3$  and scalar  $\Delta T > 0$ , the following*

relationship holds,

$$\begin{aligned} \int_0^{\Delta T} e^{(\omega\tau)^\times} d\tau &= \frac{\sin(\|\omega\|\Delta T)}{\|\omega\|} \cdot I + \left( \Delta T - \frac{\sin(\|\omega\|\Delta T)}{\|\omega\|} \right) \left( \frac{-\omega}{\|\omega\|} \right) \left( \frac{-\omega}{\|\omega\|} \right)^T \\ &\quad - \left( \frac{\cos(\|\omega\|\Delta T) - 1}{\|\omega\|} \right) \left( \frac{-\omega}{\|\omega\|} \right)^\times \end{aligned} \quad (\text{D.190})$$

**Proof:** Let the following angle/axis variables be defined for notational convenience,

$$a \triangleq \frac{-\omega}{\|\omega\|} \quad \text{and} \quad \varphi \triangleq \|\omega\|. \quad (\text{D.191})$$

Then,

$$\int_0^{\Delta T} e^{(\omega\tau)^\times} d\tau = \int_0^{\Delta T} C \left( \frac{-\omega}{\|\omega\|}, \|\omega\|\tau \right) d\tau = \int_0^{\Delta T} C(a, \varphi) d\tau \quad (\text{D.192})$$

$$= \int_0^{\Delta T} \left[ \cos(\varphi\tau) \cdot I + (1 - \cos(\varphi\tau)) aa^T - \sin(\varphi\tau) a^\times \right] d\tau \quad (\text{D.193})$$

$$= \left[ \frac{\sin(\varphi\tau)}{\varphi} \cdot I + \left( \tau - \frac{\sin(\varphi\tau)}{\varphi} \right) aa^T + \frac{\cos(\varphi\tau)}{\varphi} a^\times \right]_0^{\Delta T} \quad (\text{D.194})$$

$$= \frac{\sin(\varphi\Delta T)}{\varphi} \cdot I + \left( \Delta T - \frac{\sin(\varphi\Delta T)}{\varphi} \right) aa^T - \left( \frac{\cos(\varphi\Delta T) - 1}{\varphi} \right) a^\times \quad (\text{D.195})$$

Here, equation (D.192) follows by equation (D.171) of Lemma D.15 with the choice  $\theta = \omega\Delta T$ , and using the angle/axis quantities defined in (D.191); equation (D.193) follows by the definition of  $C$  in (D.172); equation (D.194) follows by direct integration; and (D.195) follows by evaluating the expression between its lower and upper limits. Applying definitions (D.191) to equation (D.195) gives the desired expression (D.190).

## D.17 Sum Factorization Lemma

### LEMMA D.17 Sum Factorization Lemma

Given matrices  $A \in \mathcal{R}^{n \times m}$  and  $B \in \mathcal{R}^{n \times \ell}$ , let the matrix factor  $C \in \mathcal{R}^{n \times n}$  be defined (non-uniquely) by the following relation,

$$CC^T = AA^T + BB^T \quad (\text{D.196})$$

Then a value for  $C$  can be computed as,

$$C = X \quad (\text{D.197})$$

where  $X$  is determined from the unitary triangularization of the matrix  $\begin{bmatrix} A \\ B \end{bmatrix}$  given as follows,

$$\begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} X \\ 0 \end{bmatrix} \Theta \quad (\text{D.198})$$

where  $\Theta$  is an orthogonal matrix (i.e.,  $\Theta\Theta^T = I$ ).

**Proof:** Both sides of (D.198) can be squared up to give,

$$AA^T + BB^T = \begin{bmatrix} A \\ B \end{bmatrix} \begin{bmatrix} A^T \\ \dots \\ B^T \end{bmatrix} \quad (\text{D.199})$$

$$= \begin{bmatrix} C \\ 0 \end{bmatrix} \Theta\Theta^T \begin{bmatrix} C^T \\ \dots \\ 0 \end{bmatrix} \quad (\text{D.200})$$

$$= \begin{bmatrix} C \\ 0 \end{bmatrix} \begin{bmatrix} C^T \\ \dots \\ 0 \end{bmatrix} \quad (\text{D.201})$$

$$= CC^T \quad (\text{D.202})$$

where the orthogonality property  $\Theta\Theta^T = I$  has been used in (D.201). ■

## D.18 QR Calculation of $C$ Lemma

### LEMMA D.18 QR Calculation of $C$

The matrices  $C$  and  $\Theta$  in the unitary triangularization,

$$\begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} C \\ 0 \end{bmatrix} \Theta \quad (\text{D.203})$$

can be computed as  $\Theta = Q^T$  and  $C = R^T$  where matrices  $Q$  and  $R$  are defined by the following QR decomposition,

$$\begin{bmatrix} A^T \\ B^T \end{bmatrix} = Q \begin{bmatrix} R \\ \dots \\ 0 \end{bmatrix} \quad (\text{D.204})$$

**Proof:** Taking the transpose of both sides of (D.204) gives,

$$\begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} R^T \\ 0 \end{bmatrix} Q^T \quad (\text{D.205})$$

which becomes equivalent to (D.203) under the correspondences  $C = R^T$  and  $\Theta = Q^T$ . ■

**REMARK D.1** Lemma D.18 is useful because it indicates the  $C$  factor defined in (D.196) of Lemma D.17 can be computed by first taking the QR factorization indicated in (D.204) and then setting  $C = R^T$ . Since  $Q$  is not needed, the “economy size” QR decomposition can be used for reducing computation. ■

## D.19 Differentiation through Inversion Lemma

### LEMMA D.19 Differentiation through Inversion Lemma

Let the matrix  $A \in \mathcal{R}^{n \times n}$  be a function of a scalar parameter  $p$ , and let  $A^{-1}$  exist. Then the derivative of  $A^{-1}$  with respect to  $p$  is given by,

$$\frac{\partial}{\partial p} (A^{-1}) = -A^{-1} \frac{\partial A}{\partial p} A^{-1} \quad (\text{D.206})$$

**Proof:** Since  $A$  invertible,

$$A^{-1}A = I \quad (\text{D.207})$$

Differentiating both sides with respect to  $p$  (using the chain rule for matrix products) gives,

$$\frac{\partial}{\partial p} (A^{-1}) A + A^{-1} \frac{\partial A}{\partial p} = 0 \quad (\text{D.208})$$

Multiplying on the right by  $A^{-1}$  and rearranging yields (D.206), as desired. ■

## D.20 Norm of Mirror-Axis Perturbation Lemma

### LEMMA D.20 Norm of Mirror-Axis Perturbation Lemma

Define the perturbation  $\delta a \in \mathcal{R}^3$  on the mirror spin axis (restricted to in-plane motion) as follows,

$$\delta a = h_a(\hat{a}) \delta \alpha \quad (\text{D.209})$$

where,

$$h_a(\hat{a}) = \begin{bmatrix} 0 \\ \hat{a}_3 \\ -\hat{a}_2 \end{bmatrix} \quad (\text{D.210})$$

Then the following identities hold,

$$|\delta \alpha| \triangleq \frac{\|\delta a\|}{\sqrt{\hat{a}_3^2 + \hat{a}_2^2}} \quad (\text{D.211})$$

$$\delta \alpha = \text{sign} \left( \frac{\delta a_2}{\hat{a}_3} \right) \cdot \frac{\|\delta a\|}{\sqrt{\hat{a}_3^2 + \hat{a}_2^2}} \quad (\text{D.212})$$

**Proof:** Since  $\delta \alpha$  is a scalar, one can write (D.209) as,

$$\delta a = \begin{bmatrix} 0 \\ \hat{a}_3 \delta \alpha \\ -\hat{a}_2 \delta \alpha \end{bmatrix} \quad (\text{D.213})$$

Taking the norm of both sides yields,

$$\|\delta a\| = \sqrt{\hat{a}_3^2 \delta \alpha^2 + \hat{a}_2^2 \delta \alpha^2} = |\delta \alpha| \sqrt{\hat{a}_3^2 + \hat{a}_2^2} \quad (\text{D.214})$$

which gives (D.211) upon rearranging. Moreover, using (D.213) the signs can be related as,

$$\text{sign}(\delta \alpha) = \text{sign} \left( \frac{\delta a_2}{\hat{a}_3} \right) = \text{sign} \left( \frac{\delta a_3}{-\hat{a}_2} \right) \quad (\text{D.215})$$

Combining (D.215) and (D.211) gives (D.212) as desired. ■

## D.21 Optical Distortion Vectorization Lemma

### LEMMA D.21 Optical Distortion Vectorization Lemma

Let's consider an optical distortion function  $f_d = M(y, p_1)z$  where  $M$  is a  $2 \times 2$  matrix and  $y, z \in \mathcal{R}^2$ . This optical distortion function is defined and can be expanded as

$$M(y, p_1)z = (M_{00} + \Gamma M_{10} + \Gamma^2 M_{20} + M_{01}(y))z \quad (\text{D.216})$$

where

$$M_{00} = \begin{bmatrix} a_{00} & c_{00} \\ c_{00} & b_{00} \end{bmatrix}; \quad M_{10} = \begin{bmatrix} a_{10} & c_{10} \\ d_{10} & b_{10} \end{bmatrix}; \quad M_{20} = \begin{bmatrix} a_{20} & c_{20} \\ d_{20} & b_{20} \end{bmatrix};$$

$$M_{01}(y) = \begin{bmatrix} a_{01}y_w + c_{01}y_v & b_{01}y_v \\ d_{01}y_w & f_{01}y_w + e_{01}y_v \end{bmatrix}. \quad (\text{D.217})$$

Given the state vector  $p_1$  definition of:

$$p_1 = [a_{00} \ b_{00} \ c_{00} \ a_{10} \ b_{10} \ c_{10} \ d_{10} \ a_{20} \ b_{20} \ c_{20} \ d_{20} \ a_{01} \ b_{01} \ c_{01} \ d_{01} \ e_{01} \ f_{01}]^T, \quad (\text{D.218})$$

the vectorization of  $Mz$  results in a relationship of:

$$-M(p_1, y)z = -(z^T \otimes I) \text{Vec}(M). \quad (\text{D.219})$$

Where the Kronecker Product ( $\otimes$ ) results in:

$$z^T \otimes I = \begin{bmatrix} z_w & 0 & z_v & 0 \\ 0 & z_w & 0 & z_v \end{bmatrix}, \quad (\text{D.220})$$

and the vectorization of  $M$  can be written as:

$$\text{Vec}(M) = [S_{00} \ \Gamma S_{10} \ \Gamma^2 S_{20} \ f_u(y)] p_1 \quad (\text{D.221})$$

where

$$S_{00} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad S_{10} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad S_{20} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad (\text{D.222})$$

and

$$f_u(y) = \begin{bmatrix} y_w & 0 & y_v & 0 & 0 & 0 \\ 0 & 0 & 0 & y_w & 0 & 0 \\ 0 & y_v & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & y_v & y_w \end{bmatrix}. \quad (\text{D.223})$$

**Proof:** Let  $A = I$ ,  $B = M$ , and  $C = z$ . If the Lemma D.22 is applied to  $ABC$ , then

$$\text{vec}(\mathbf{ABC}) = (\mathbf{C}^T \otimes \mathbf{A}) \text{vec}(\mathbf{B}) = (\mathbf{z}^T \otimes \mathbf{I}) \text{vec}(\mathbf{M}). \quad (\text{D.224})$$

Since

$$\text{vec}(M_{00}) = \begin{bmatrix} a_{00} \\ c_{00} \\ c_{00} \\ d_{00} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} a_{00} \\ c_{00} \\ d_{00} \end{bmatrix} \implies \text{vec}(M_{00}) \triangleq S_{00} \begin{bmatrix} a_{00} \\ c_{00} \\ d_{00} \end{bmatrix} \quad (\text{D.225})$$

$$\text{vec}(M_{10}) = \begin{bmatrix} a_{10} \\ d_{10} \\ c_{10} \\ b_{10} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_{10} \\ b_{10} \\ c_{10} \\ d_{10} \end{bmatrix} \implies \text{vec}(M_{10}) \triangleq S_{10} \begin{bmatrix} a_{10} \\ b_{10} \\ c_{10} \\ d_{10} \end{bmatrix} \quad (\text{D.226})$$

$$\text{vec}(M_{20}) = \begin{bmatrix} a_{20} \\ d_{20} \\ c_{20} \\ b_{20} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_{20} \\ b_{20} \\ c_{20} \\ d_{20} \end{bmatrix} \implies \text{vec}(M_{20}) \triangleq S_{20} \begin{bmatrix} a_{20} \\ b_{20} \\ c_{20} \\ d_{20} \end{bmatrix} \quad (\text{D.227})$$

and

$$\text{vec}(M_{01}) = \begin{bmatrix} y_w & 0 & y_v & 0 & 0 & 0 \\ 0 & 0 & 0 & y_w & 0 & 0 \\ 0 & y_v & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & y_v & y_w \end{bmatrix} \begin{bmatrix} a_{01} \\ b_{01} \\ c_{01} \\ d_{01} \\ e_{01} \\ f_{01} \end{bmatrix} \triangleq f_u(y) \begin{bmatrix} a_{01} \\ b_{01} \\ c_{01} \\ d_{01} \\ e_{01} \\ f_{01} \end{bmatrix} \quad (\text{D.228})$$

Thus  $\text{Vec}(M) = [ S_{00} \quad \Gamma S_{10} \quad \Gamma^2 S_{20} \quad f_u(y) ] p_1$

■

## D.22 Matrix Vectorization Lemma

### LEMMA D.22 Matrix Vectorization Lemma

For any  $m \times n$  matrix  $\mathbf{A}$ ,  $n \times p$  matrix  $\mathbf{B}$ , and  $p \times q$  matrix  $\mathbf{C}$ ,

$$\text{vec}(\mathbf{ABC}) = (\mathbf{C}^T \otimes \mathbf{A}) \text{vec}(\mathbf{B}). \quad (\text{D.229})$$

**Proof:** (see Reference [11],[5], and [16] ) Assume  $\mathbf{B}$  can be vectorized and is expressible as

$$\mathbf{B} = \sum_{j=1}^p \mathbf{b}_j \mathbf{u}_j^T. \quad (\text{D.230})$$

where (for  $j = 1, \dots, p$ )  $\mathbf{b}_j$  is the  $j^{\text{th}}$  column of  $\mathbf{B}$  and  $\mathbf{u}_j^T$  is the  $j^{\text{th}}$  row of  $\mathbf{I}_p$ . Thus making use of Lemmas D.24 and D.23, we find that

$$\text{vec}(\mathbf{ABC}) = \text{vec} \left[ \mathbf{A} \left( \sum_{j=1}^p \mathbf{b}_j \mathbf{u}_j^T \right) \mathbf{C} \right] \quad (\text{D.231})$$

$$= \sum_{j=1}^p \text{vec}(\mathbf{A} \mathbf{b}_j \mathbf{u}_j^T \mathbf{C}) \quad (\text{D.232})$$

$$= \sum_{j=1}^p [(\mathbf{C}^T \mathbf{u}_j) \otimes (\mathbf{A} \mathbf{b}_j)] \quad (\text{D.233})$$

$$= \sum_{j=1}^p (\mathbf{C}^T \otimes \mathbf{A}) (\mathbf{u}_j \otimes \mathbf{b}_j) \quad (\text{D.234})$$

$$= \sum_{j=1}^p (\mathbf{C}^T \otimes \mathbf{A}) \text{vec}(\mathbf{b}_j \mathbf{u}_j^T) \quad (\text{D.235})$$

$$= (\mathbf{C}^T \otimes \mathbf{A}) \text{vec} \left( \sum_{j=1}^p \mathbf{b}_j \mathbf{u}_j^T \right) = (\mathbf{C}^T \otimes \mathbf{A}) \text{vec}(\mathbf{B}) \quad (\text{D.236})$$

■

## D.23 Vector-Kronecker Equivalence Lemma

### LEMMA D.23 Vector-Kronecker Equivalence Lemma

For any  $m$ -dimensional column vector  $\mathbf{a} = \{a_i\}$  and  $n$ -dimensional column vector  $\mathbf{b}$ ,

$$\text{vec}(\mathbf{b}\mathbf{a}^T) = \mathbf{a} \otimes \mathbf{b} \quad (\text{D.237})$$

**Proof:**(see Reference [11]) Let  $\mathbf{a} = \{a_i\}$  and  $\mathbf{b} = \{b_i\}$  to represent column vectors of dimensions  $m$  and  $n$ , respectively. The Kronecker Product of  $\mathbf{a}$  and  $\mathbf{b}$  can be written as

$$\mathbf{a} \otimes \mathbf{b} = \begin{pmatrix} a_1 \mathbf{b} \\ a_2 \mathbf{b} \\ \vdots \\ a_m \mathbf{b} \end{pmatrix} \quad (\text{D.238})$$

Note that  $\mathbf{a} \otimes \mathbf{b}$  is an  $mn$ -dimensional partitioned column vector comprising  $m$  subvectors, the  $i$ th of which is  $a_i \mathbf{b}$ . Since the  $i$ th (of the  $m$  columns) of  $\mathbf{b}\mathbf{a}^T$  is  $a_i \mathbf{b}$ , the right hand side of (D.238) equals the vectorization of  $\mathbf{b}\mathbf{a}^T$ .

■

## D.24 Kronecker Product Lemma

### LEMMA D.24 Kronecker Product Lemma

For any  $m \times n$  matrix  $\mathbf{A} = \{a_{ij}\}$ ,  $p \times q$  matrix  $\mathbf{B} = \{b_{ij}\}$ ,  $n \times u$  matrix  $\mathbf{C} = \{c_{ij}\}$ , and  $q \times v$  matrix  $\mathbf{D} = \{d_{ij}\}$ ,

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD}). \quad (\text{D.239})$$

**Proof:** (From Reference [11]) By definition,  $\mathbf{A} \otimes \mathbf{B}$  is a partitioned matrix, comprising  $m$  rows and  $n$  columns of  $p \times q$  dimensional blocks, the  $ij$ th of which is  $a_{ij}\mathbf{B}$ ; and  $\mathbf{C} \otimes \mathbf{D}$  is a partitioned matrix, comprising  $n$  rows and  $u$  columns of  $q \times v$  dimensional blocks, the  $jr$ th of which is  $c_{jr}\mathbf{D}$ . Thus,  $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D})$  is a partitioned matrix, comprising  $m$  rows and  $u$  columns of  $p \times v$  dimensional blocks, the  $ir$ th of which is the matrix

$$\sum_{j=1}^n (a_{ij}\mathbf{B})(c_{jr}\mathbf{D}) = \left( \sum_{j=1}^n a_{ij}c_{jr} \right) \mathbf{BD} \quad (\text{D.240})$$

By way of comparison,  $(\mathbf{AC}) \otimes (\mathbf{BD})$  is a partitioned matrix, comprising  $m$  rows and  $u$  columns of  $p \times v$  dimensional blocks, the  $ir$ th of which is the matrix

$$f_{ir}\mathbf{BD} \quad (\text{D.241})$$

where  $f_{ir}$  is the  $ir$ th element of  $\mathbf{AC}$ . The proof is complete upon observing that:

$$f_{ir} = \sum_{j=1}^n a_{ij}c_{jr} \quad (\text{D.242})$$

and hence that the  $ir$ th block of  $(\mathbf{AC}) \otimes (\mathbf{BD})$  equals the  $ir$ th block of  $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D})$ . ■

## ACKNOWLEDGEMENTS

This research was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

## References

- [1] D.S. Bayard, "Spacecraft pointing control criteria for high resolution spectroscopy," IEEE Transactions on Aerospace and Electronic Systems, Vol. 35, No. 2, pp. 637-644, April 1999.
- [2] D.S. Bayard, B.H. Kang, *SIRTF Instrument Pointing Frame (IPF) Kalman Filter Unit Test Report*. JPL D-Document D-24810, October 31, 2002.
- [3] D.S. Bayard, "SIRTF Telescope Pointing Frame (TPF) Definition," JPL Internal Document, Engineering Memorandum EM 3457-02-006, April 30, 2002.
- [4] G.J. Bierman and M.D. Schuster, "Spacecraft alignment estimation," 27th IEEE Conference on Decision and Control, Austin, TX, Dec. 7-9, 1988.
- [5] J.W. Brewer, "Kronecker Products and Matrix Calculus in System Theory," IEEE Transactions on Circuits and Systems, Vol. CAS-25, No. 9, September 1978.
- [6] R.J. Brown, *Focal Surface to Object Space Field of View Conversion*. Systems Engineering Report SER No. S20447-OPT-051, Ball Aerospace & Technologies Corp., November 23, 1999.
- [7] S. Davis and J. Lai, "Attitude sensor calibration for the Ocean Topography Experiment Satellite (TOPEX)," SPIE Space Guidance, Control and Tracking Conference, Vol. 1949, Society of Photo-Optical Instrumentation Engineers, pp. 80-91, 1993.
- [8] J. Deutschmann and I.Y. Bar-Itzhack, "Extended Kalman Filter for attitude estimation of the Earth Radiation Budget Satellite," Proc. Flight Mechanics/Estimation Theory Symposium, NASA Goddard Space Flight Center, Greenbelt, Maryland, pp. 319-332, May 10-11, 1989.
- [9] *Space Infrared Telescope Facility In-Orbit Checkout and Science Verification Mission Plan*. JPL D-Document D-22622, 674-FE-301 Version 1.4, February 8, 2002.
- [10] A. Gelb (Ed.), *Applied Optimal Estimation*. The MIT Press, Cambridge, Massachusetts, 1984.
- [11] D.A. Harville, *Matrix Algebra From A Statistician's Perspective*. Springer-Verlag New York, 1997.

- [12] T. Kailath, A.H. Sayed, B. Hassibi, *Linear Estimation*. Prentice Hall, Upper Saddle River, New Jersey, 1999.
- [13] B.H. Kang and D.S. Bayard, *Spitzer Instrument Pointing Frame (IPF) Software Description Document and User's Guide*. JPL D-Document D-24808, June 30, 2004.
- [14] T. Kia, D.S. Bayard, F. Tolivar, "A Precision Pointing Control System for the Space Infrared Telescope Facility (SIRTF)," American Astronautical Society, AAS97-067, 1997.
- [15] J.W. Murrell, "Precision attitude determination for multimission spacecraft," Paper No. 78-1248, Proc. AIAA Guidance and Control Conference, Palo Alto, California, August 7-9, 1978.
- [16] H. Neudecker, "Some Theorems on Matrix Differentiation with Special Reference to Kronecker Matrix Products," J. of Amer. Stat. Assoc., VOL. 64, pp.953-963, 1969
- [17] F.C. Park, *The Optimal Kinematic Design of Mechanisms*. Ph.D. Thesis, Applied Mathematics, Harvard University, Cambridge, Massachusetts, 1991.
- [18] M. Pittelkau, "Kalman filtering for spacecraft system alignment calibration," J. Guidance, Control and Dynamics, Vol. 24, No. 6, November-December 2001.
- [19] J. Schwenker, *Focal Surface to Object Space Field of View Conversion, Review and Simplified Math Model*. Systems Engineering Report SER No. S20447-OPT-057, Ball Aerospace & Technologies Corp., September 9, 2000.
- [20] J. Schwenker, *Object Space Field of View, As Built*. Systems Engineering Report SER No. S2447-OPT-072, Ball Aerospace & Technologies Corp., May 15, 2001.
- [21] P.K. Seidelmann, *Explanatory Supplement to the Astronomical Almanac*. University Science Books, Sausalito, California, 1992.
- [22] M.D. Shuster, "Inflight Estimation of Spacecraft Sensor Alignment," Advances in the Astronautical Sciences, 72, 1990.
- [23] M.D. Shuster and R.V.F. Lopes, "Parameter interference in distortion and alignment calibration, AAS Paper AAS 94-186, Feb. 1994.
- [24] R.P. Wishner, J.A. Tabaczynski, M. Athans, "A Comparison of Three Non-Linear Filters," Automatica, Vol. 5, pp.487-496, Pergamon Press, 1969.
- [25] E. Wong, W. Breckenridge, D. Boussalis, P. Brugarolas, D.S. Bayard, J. Spanos, G. Singh, "Post-Flight Attitude Reconstruction for the Shuttle Radar Topography Mission," Proc. AAS/AIAA Astrodynamics Specialists Conference, Paper AAS 01-314, Quebec City, Quebec, Canada, July 30, 2001.