

## ABSTRACT

Title of thesis: **ORBIT DETERMINATION METHODS FOR  
DEEP SPACE DRAG-FREE CONTROLLED  
LASER INTERFEROMETRY MISSIONS**

**Lauren Rose Chung, Master of Science, 2006**

Thesis directed by: Assistant Professor Dr. Ella Atkins  
Department of Aerospace Engineering  
University of Maryland

The objective of this thesis is to determine if different measurement combinations will improve absolute and relative orbit determination (OD) accuracy for the Laser Interferometer Space Antenna (LISA) mission. LISA, a triangular three-satellite heliocentric constellation, measures the relative length changes between spacecraft. The unique contribution of this work is the incorporation of inter-spacecraft range and range rate observations into the OD process, in addition to Deep Space Network (DSN) range and range rate, and Very Long Baseline Interferometry (VLBI) angle tracking data. MATLAB was used for orbit propagation, simulating measurements, and executing the weighted batch least squares algorithm to obtain the best state estimate. Covariance and Monte Carlo analyses were performed to compare the four test cases. The results concluded that cases with inter-spacecraft data had the least absolute and relative OD errors, VLBI eventually becomes ineffectual, and DSN needs at least 20 days of tracking to become observable.

ORBIT DETERMINATION METHODS FOR DEEP SPACE DRAG-FREE  
CONTROLLED LASER INTERFEROMETRY MISSIONS

By

Lauren Rose Chung

Thesis submitted to the Faculty of the Graduate School of the  
University of Maryland, College Park, in partial fulfillment  
of the requirements for the degree of  
Masters of Science  
2006

Advisory Committee:

Assistant Professor Ella Atkins

Dr. Tristram Tupper Hyde

Associate Professor Robert Sanner

© Copyright by  
Lauren Rose Chung  
2006

## Preface

This thesis topic began as a ten-week project during my Summer Institute in Engineering and Computer Applications (SIECA) internship at NASA-Goddard Space Flight Center during the summer of 2004. Under the mentorship of Dr. Tristram Tupper Hyde, the project involved taking a preliminary look at a two-dimensional model of a spacecraft around a flat Earth then simulating three spacecraft orbiting around the Sun in three-dimensions. The only data type considered was range. The batch least squares method was used to estimate the best state. The experience exposed me to brilliant people, got me involved in research, interested me in orbit determination, and made me want to keep on pursuing the complicated problem of tracking spacecraft.

I returned to University of Maryland that fall to finish my Master's coursework and in the spring of 2005, I fully embarked again on this journey of research to complete the final thesis requirement. The topic got deeper and many lessons were learned along the way.

## Dedication

This thesis is dedicated to my Mother: Maxine "Joy" Chung who is *simply mommy*. This is a woman who is always so positive despite it all, who is so giving, who works so hard to provide the best for us, who has an endless supply of crazy expressions that taught me a lot about life, and most of all who always believed in me.

## Acknowledgements

I first want to acknowledge all the members of my very large family, without the support of which, I could not have done this. The most special shout-out is to my sister Kristen who is my twin (okay, we are really one year and nine months apart) - all my quirkiness and humor is not lost on you. You get me and for that I am eternally grateful. Thank you Kristen for counseling me through all the stages of this journey: the pep talks and the straight-talks, and for keeping me hip and down-to-Earth when I thought I was losing my mind. Mom and Dad, I want to thank you for all the wake-up calls. Mom, thanks for always alerting me whenever the words *space*, *shuttle*, *launch* or *NASA* were mentioned on TV. You are that constant force behind me, for better or worse. Dad, thanks for loving science fiction even if I cannot get down with Stargate, Star Trek and Babylon 5. I know: "what kind of aerospace engineer am I?" However we do see eye to eye on our love of the Lord of the Rings and curiously Sex and the City, and thanks for helping me handle those other areas of life: finances and car issues. Thanks to my brother Hugh (an engineer in disguise who also loves Stargate and all those build-it shows) and his wife Shanna for being our get-away out in California. And to my bite-size inspirations, my niece and nephew: Mattea and Austin: the cutest kids in the world, who are so smart, funny and imaginative and who say the darnedest things.

To my local family: my cousin Fran (and her dog Quincy) for helping me get acclimated to MD and Paul, Rona and Aunt Patsy (and Sam the cat) for being my escape in VA and

for listening while I went on and on about school, oh the drama. And thanks for feeding me Dave's Barbeque.

I want to acknowledge my friends for being there for me, supporting me, and somehow always knowing that I could do this. Special shout-outs to my most intelligent and beautiful friends: Keturah, Heneryatta and Angel who are doing their thing by getting their PhDs. You go girls!!! I cherish that we met at Penn State, formed a lifelong bond and had so many fun times in the process. You women continue to inspire me. Angel our friendship deepened here together at Maryland as we tried to navigate through this crazy stage called "Grad school life," and thanks for formally introducing me to "PhD Comics"- how relatable! More Penn Staters that I want to acknowledge are my dear friends Ebony and Brandon (and now baby Jada), Fernando- my ex-ballroom dance partner, Yanira, and Karla. Also, Amy Freeman, Sandra Johnson and Anita Persaud from the MEP, Diane and D'Andre, the II Hype Penn State NSBE Chapter, and my fellow Aerospace Engineers: Jonathan Coleman, Eric Parsons, Erika Mendoza and Dr. Melton for being such a fabulous teacher. Jen DeLecce, my partner in crime in Australia, who is only in VA, but might as well be Australia, since I barely make it to see you. Here is to more free time when this thesis is complete! And to my pal, Damon Bradley, our paths have coincided way too long; Penn State, NSBE ("II Hype!"), NASA internships, Grad school at UMD, and now living and working in Maryland. Damon thank you for always being so positive, smart, for converting me to love glow sticks and techno music, and for deriving crazy equations just for fun and making it okay to be a little (or a lot) geeky.

I also want to acknowledge fellow UMD grad students, including Janisa Henry and my SSL lab mates, who are going through process, as well. And the members of Black Graduate Student Association (BGSA), especially for all the potlucks and deep conversations that went on for hours. Good luck to all. I also want to thank Cynthia Simmons for befriending me from the start in Astrodynamics and moving me to achieve. And thank you Dan Rolf for lending me (err... letting me kidnap) that Vallado textbook.

I want to acknowledge the people in my hobbies/passions: my dance sisters (Malika, Sirena, Leyla, Zuhairah and Gia) and yoga buddies for adding a dose of drama to my life (the former) and for centering me (latter). I am truly thankful and grateful for the journey. I have learned so much more about myself and plan to dance and do yoga until I die.

To all the mentors throughout my life who have inspired me every step of the way: Willard Manning who I credit with being my first true Mentor with a project called "The Environmental Benefits of Earth Observing Satellites," and who is now a life-long friend and unofficial family member. And to Sharae McKay who inspired me to go for a higher degree in Aerospace engineering.

I would like to acknowledge these people from my NASA-GSFC 2004 internship:

-Dr. Russell Carpenter: for answering all my questions, being so generous with your vast knowledge and being an inspiration.



-Dr. Landis Markley: for letting me co-inhabit your office, being the human encyclopedia and pulling any journal article or resource I could possibly need from his files.

From UMD:

-Dr. Healy: though I met you late in the process, I thank you for your help this past summer, and your interest in Cathy Thornton' paper and class notes.

-Dr. Bowden: for always being so positive every time I saw you and for your sense of amazement at everything your mere students were doing.

-Dr. Tolson: for drilling into your student's heads that a solar day is longer than a sidereal day, for assigning those group projects (my group: Dan, Matt and Adam) to make all those MATLAB Astronautics toolkits (though we students were not happy at the time) and for giving us that Cathy Thornton paper.

-Dr. Atkins: for giving me a space in the Space Systems Lab, though I was the odd woman out in the robotics lab working on an orbit determination project! Thanks for your help and counseling on various stages of my thesis. And thank you immensely for your wonderful revisions of the thesis.

-Dr. Sanner: for early conversations on orbit determination.

And I would also like to thank the wonderful employees of a.i.-solutions, Inc. for hiring a girl, not quite done with her Master's thesis. Especially to David Hooshmand, Christy Jeffries, Conrad Schiff, Daryl Carrington who hired me. I won't disappoint you, I am just warming up. And my entertaining officemates: Allan, Dominic and Matt- we do have the hottest (temperature-wise) office at a.i.

And once again, I want to sincerely thank my mentor Dr. Tristram "Tupper" Hyde who went above the call of duty and took on the task mentoring a student through her Master's thesis. You are the one who gave a hopeless grad student a Master's thesis topic in the summer of 2004 and stood by me for the last two years following that summer internship. Thank you for your genuine interest in this research topic. Thanks for the e-mails, the calls, the meetings, helping with the MATLAB codes in all its states of "bugginess," dealing with my endless reading, note taking and "research" that had to finally be channeled into concrete results.

# Table of Contents

Preface.....	ii
Dedication.....	iii
Acknowledgements.....	iv
Table of Contents.....	ix
List of Tables .....	xi
List of Figures.....	xii
List of Abbreviations .....	xv
List of Symbols.....	xix
Chapter 1: Introduction.....	1
1.1. Thesis Objectives & Contributions.....	4
1.2. Overview of Thesis.....	5
Chapter 2: Background .....	6
2.1. Mission & Science Objectives.....	7
2.1.1. Gravitational Waves .....	8
2.1.2. Laser Interferometry .....	10
2.2. LISA Spacecraft Design .....	11
2.3. Instrument Package .....	13
2.3.1. Drag-Free Control .....	15
2.4. LISA Launch & Mission Evolution .....	16
2.5. Orbit Geometry & Configuration .....	17
2.6. Constraints & Technological Challenges.....	20
2.7. Related Missions.....	21
Chapter 3: Methods & Math Techniques.....	26
3.1. The Two-Body Problem .....	26
3.2. Orbital Elements .....	29
3.3. Reference Frames.....	34
3.3.1. Sun-Centered Inertial Coordinate System .....	35
3.3.2. Earth-Centered Inertial Coordinate System.....	36
3.3.3. Satellite Radial Coordinate System .....	37
3.3.4. Time .....	38
3.4. Tracking Systems & Observation Types .....	39
3.4.1. Deep Space Network (DSN).....	40
Range .....	42
Range Rate .....	44
3.4.2. Very Long Baseline Interferometry (VLBI).....	45
3.4.3. Inter-spacecraft Data.....	48
3.5. Tracking Data Error Sources .....	50

Chapter 4: Orbit Determination .....	52
4.1. Orbit Determination History .....	52
4.2. Classical Orbit Determination.....	57
4.3. Orbit Determination Problem Formulation.....	59
4.3.1. Aspects of Estimation .....	61
4.3.2. Linearization of the Non-linear Problem .....	64
4.3.3. The State Transition Matrix .....	67
4.3.4. Relating Observations to an Epoch State .....	68
4.4. Orbit Determination Methods.....	69
4.4.1. Batch Least Squares.....	69
4.4.2. Weighted Batch Least Squares .....	72
4.5. Statistical Methods.....	74
4.5.1. Linear Minimum Variance Estimation (LUMV).....	74
4.5.2. Sequential Methods.....	76
4.5.3. Comparison of OD Methods.....	77
4.6. Perturbation Effects.....	79
 Chapter 5: Problem Description.....	 81
5.1. Assumptions in the Model .....	81
5.2. Batch Processor Algorithm.....	85
5.2.1. Test Cases .....	85
5.2.2. The Nominal Orbit.....	86
5.2.3. The Algorithm.....	97
5.2.4. Simulation of Measurement Noise.....	104
5.3. Validity of the Filter.....	109
5.4. Orbit Accuracy.....	112
 Chapter 6: Results .....	 117
6.1. Comments on Orbit Geometry.....	117
6.2. Measurements Over Time.....	125
6.3. Main Analysis .....	127
6.3.1. Covariance Analysis .....	130
6.3.2. Monte Carlo Runs .....	132
6.4. Discussion of Results.....	145
 Chapter 7: Conclusion.....	 150
7.1. Future Work.....	152
 Appendix A.....	 156
Appendix B .....	157
Bibliography .....	194

## List of Tables

<b>Table 3.1:</b>	Orbital element systems .....	34
<b>Table 3.2:</b>	DSN DSCC locations [35] .....	41
<b>Table 5.1:</b>	Hughes' nominal orbital elements of LISA & Earth .....	89
<b>Table 5.2:</b>	Dhurandhar's nominal orbital elements of LISA & Earth .....	96
<b>Table 5.3:</b>	Noise Levels for each measurement [41] .....	107
<b>Table 6.1:</b>	Abridged version of DSN/VLBI/Relative $H$ matrix at day 20 .....	120
<b>Table 6.2:</b>	DSN/VLBI $H$ matrix for LISA 1 at day 90 .....	123
<b>Table 6.3:</b>	Summary of Mean Absolute Position & Velocity with Standard Deviations .....	135
<b>Table 6.4:</b>	Summary of Mean Relative Position & Velocity with Standard Deviations .....	138
<b>Table 6.5:</b>	Summary of Mean Absolute Position & Velocity with Standard Deviations .....	142
<b>Table 6.6:</b>	Summary of Mean Relative Position & Velocity with Standard Deviations .....	143
<b>Table 6.7:</b>	Parameter Uncertainties ( $PU$ ) for DSN/VLBI/Relative Case at day 7...149	

## List of Figures

<b>Figure 2.1:</b>	LISA orbit geometry relative to the Sun and Earth [1] .....	7
<b>Figure 2.2:</b>	Lasers signals between LISA [1] .....	11
<b>Figure 2.3:</b>	The LISA spacecraft [1] .....	12
<b>Figure 2.4:</b>	Y-shaped structure [1] .....	12
<b>Figure 2.5:</b>	Optical assembly [1] .....	13
<b>Figure 2.6:</b>	Orbit geometry of LISA constellation without angles [1] .....	18
<b>Figure 2.7:</b>	LISA orbit over an entire year [1] .....	19
<b>Figure 3.1:</b>	The two-body problem [47] .....	27
<b>Figure 3.2:</b>	Orbital elements [37] .....	32
<b>Figure 3.3:</b>	Heliocentric-Ecliptic coordinate system ( <i>XYZ</i> ) [35] .....	35
<b>Figure 3.4:</b>	Earth-Centered Inertial coordinate system (ECI) [35] .....	37
<b>Figure 3.5:</b>	Satellite Radial coordinate system ( <i>RSW</i> ) [35] .....	38
<b>Figure 3.6:</b>	Very Long Baseline Interferometry (VLBI) diagram [42] .....	46
<b>Figure 3.7:</b>	Pictorial representation of Earth-based DSN and VLBI measurements ..	48
<b>Figure 3.8:</b>	Pictorial representation of Earth-based and inter-spacecraft measurements .....	49
<b>Figure 4.1:</b>	The Orbit Determination problem .....	60
<b>Figure 4.2:</b>	Along-Track and Cross-Track Errors for LISA [35] .....	62
<b>Figure 5.1:</b>	Angle measurements to spacecraft .....	82
<b>Figure 5.2:</b>	Geocentric parallax .....	84
<b>Figure 5.3:</b>	Initial position of LISA constellation (Equal Axis) .....	90
<b>Figure 5.4:</b>	LISA constellation over 240 days .....	90

<b>Figure 5.5:</b>	LISA position components over 1 year (unperturbed) .....	91
<b>Figure 5.6:</b>	LISA velocity components over 1 year (unperturbed) .....	92
<b>Figure 5.7:</b>	Leg length vs. time for an unperturbed orbit over 1 year .....	93
<b>Figure 5.8:</b>	Percent error in leg lengths for an unperturbed orbit over 1 year .....	94
<b>Figure 5.9:</b>	Relative velocity for unperturbed orbit over 1 year .....	94
<b>Figure 5.10:</b>	Rate of change vs. time of an unperturbed LISA orbit for 1 year .....	95
<b>Figure 5.11:</b>	Dhurandhar magnitude of rate-of-change vs. time for an unperturbed orbit over 1 year .....	97
<b>Figure 5.12:</b>	Weighted Least Squares Batch Processor Algorithm .....	103
<b>Figure 6.1:</b>	LISA constellation at epoch (square axis view) .....	118
<b>Figure 6.2:</b>	LISA orbit geometry over 20 days .....	119
<b>Figure 6.3:</b>	Eyeball views of triangulation .....	122
<b>Figure 6.4:</b>	LISA orbit geometry over 90 days .....	124
<b>Figure 6.5:</b>	Earth-based nominal measurements over 1 year .....	126
<b>Figure 6.6:</b>	Inter-spacecraft nominal measurements over 1 year .....	127
<b>Figure 6.7:</b>	Frequency of observations .....	129
<b>Figure 6.8:</b>	RSS of position sigmas of the Covariance Matrix for LISA 1 .....	130
<b>Figure 6.9:</b>	RSS of velocity sigmas of the Covariance Matrix for LISA 1 .....	131
<b>Figure 6.10:</b>	MC Simulation, NB, Mean Absolute Position Error .....	133
<b>Figure 6.11:</b>	MC Simulation, NB, Mean Absolute Position Error with sigmas .....	134
<b>Figure 6.12:</b>	MC Simulation, NB, Mean Absolute Velocity Error with sigmas .....	134
<b>Figure 6.13:</b>	MC Simulation, NB, Mean Relative Position Error .....	136
<b>Figure 6.14:</b>	MC Simulation, NB, Mean Relative Position Error with sigmas .....	136
<b>Figure 6.15:</b>	MC Simulation, NB, Mean Relative Velocity Error with sigmas .....	137

<b>Figure 6.16:</b>	MC Simulation, noB, Mean Absolute Position Error .....	139
<b>Figure 6.17:</b>	MC Simulation, noB, Mean Absolute Position Error with sigmas .....	140
<b>Figure 6.18:</b>	MC Simulation, noB, Mean Relative Position Error with sigmas .....	140
<b>Figure 6.19:</b>	MC Simulation, noB, Mean Absolute Velocity Error with sigmas .....	141
<b>Figure 6.20:</b>	MC Simulation, noB, Mean Relative Velocity Error with sigmas .....	141
<b>Figure 6.21:</b>	Frequency of Measurements, Absolute Position Error .....	144
<b>Figure 6.22:</b>	Frequency of Measurements, Relative Position Error .....	145
<b>Figure 6.23:</b>	Summary Chart of Days Until Absolute Position Error Equals 1 Kilometer .....	146
<b>Figure 6.24:</b>	Summary Chart of Days Until Relative Position Error Equals 300 meters .....	147
<b>Figure 6.25:</b>	Monte Carlo (Noise, Bias) Parameter Uncertainty of LISA 1 position-components .....	148



## List of Abbreviations in alphabetical order

<b>ALIA</b>	Advanced Laser Interferometer Antenna
<b>ALIAS</b>	Advanced Laser Interferometer Antenna in Stereo
<b>ASTROD</b>	Astrodynamical Space Test of Relativity Using Optical Devices
<b>AU</b>	Astronomical Unit
<b>BBO</b>	Big Bang Observer
<b>CHAMP</b>	CHALLENGING Minisatellite Payload
<b>CLIVAR</b>	Climate Variability Program
<b>cm</b>	Centimeter
<b>CMNT</b>	Colloid Micro-Newton Thruster
<b>CSO</b>	Compensated Sapphire Oscillator
<b>CTE</b>	Coefficient of Thermal Expansion
<b>DECIGO</b>	DECi-hertz Interferometer Gravitational Wave Observatory
<b>DOF(s)</b>	Degree(s) of Freedom
<b>DOR</b>	Delta one-way Range
<b>DORIS</b>	Doppler Orbitography and Radiopositioning Integrated by Satellite
<b>DSCC</b>	Deep Space Communication Centers
	<b>CDSCC</b> Canberra Deep Space Communication Center
	<b>GDSCC</b> Goldstone Deep Space Communication Center
	<b>MDSCC</b> Madrid Deep Space Communication Center
<b>DSN</b>	Deep Space Network
<b>DSS</b>	Deep Space Station
<b>D/R</b>	DSN & Relative data combination

<b>DRS</b>	Disturbance Reduction System
<b>D/V</b>	DSN & VLBI data combination
<b>D/V/R</b>	DSN, VLBI & Relative data combination
<b>ECI</b>	Earth-Centered Inertial, or Earth-based Geocentric Equatorial (ECI), or IJK Coordinate System
<b>EELV</b>	Evolved Expendable Launch Vehicle
<b>EKF</b>	Extended Kalman Filter
<b>EM</b>	Electromagnetic
<b>EOM</b>	Equation of Motion
<b>ESA</b>	European Space Agency
<b>FEED</b>	Field Emission Electric Propulsion
<b>GEO</b>	Geostationary Orbit
<b>GFZ</b>	GeoForschungsZentrum (i.e. National Research Centre for Geosciences in Pottsdam, Germany)
<b>GP-B</b>	Gravity-Probe B
<b>GPS</b>	Global Positioning System
<b>GRACE</b>	Gravity Recovery and Climate Experiment
<b>GRS</b>	Gravitational Reference Sensors
<b>GTO</b>	Geostationary Transfer Orbit
<b>Hz</b>	Hertz
<b>IMS</b>	Interferometry Measurement System
<b>JPL</b>	NASA's Jet Propulsion Laboratory
<b>kg</b>	Kilogram

<b>km</b>	Kilometer
<b>lbs</b>	Pounds
<b>LEO</b>	Low Earth Orbit
<b>LIGO</b>	Laser Interferometer Gravitational-Wave Observatory
<b>LISA</b>	Laser Interferometer Space Antenna
<b>LISAS</b>	Laser Interferometer Space Antenna in Stereo
<b>LITS</b>	Linear Ion Trap Standard
<b>LS</b>	Least Squares
<b>LTP</b>	LISA Technology Package
<b>LUMV</b>	Linear, unbiased, minimum variance
<b>m</b>	Meter
<b>MC</b>	Monte Carlo
<b>MJ2000</b>	Mean of January 1.5, 2000.0 Reference frame
<b>MLE</b>	Maximum Likelihood Estimate
<b>m/s</b>	Meters/second
<b>MTPE</b>	Mission to Planet Earth
<b>NASA</b>	National Aeronautics and Space Agency
<b>NASA/GSFC</b>	NASA Goddard Space Flight Center
<b>NASA/JPL</b>	NASA Jet Propulsion Lab
<b>NB</b>	Noise and Bias
<b>noB</b>	Noise, No Bias
<b>NOCC</b>	Network Operations Control Center
<b>nrad</b>	Nanoradians

<b>OD</b>	Orbit Determination
<b>pdf</b>	Probability density function
<b>P/M</b>	Propulsion Module
<b>PRARE</b>	Precise Range and Range Rate Equipment
<b>rad</b>	Radians
<b>RSW</b>	Satellite-Based Radial Coordinate System (where R-axis = Radial, S-axis = Along track, and W-axis = Cross-track)
<b>SCI</b>	Sun-Centered Inertial, also referred to as Heliocentric (XYZ)
<b>sec</b>	Seconds
<b>SLR</b>	Satellite Laser Ranging
<b>SMART-2</b>	Small Missions for Advanced Research and Technology
<b>STEP</b>	Satellite Test of Equivalence Principle
<b>STM</b>	State Transition Matrix
<b>ST7</b>	Space Technology 7
<b>SVD</b>	Singular Value Decomposition
<b>TCM</b>	Trajectory Correction Maneuver
<b>TDRSS</b>	Tracking & Data Relay Satellite System
<b>TRK</b>	Tracking DSN Data Type
<b>VLBI</b>	Very Long Baseline Interferometry
<b>WBLS</b>	Weighted Batch Least Squares
<b>WLS</b>	Weighted Least Squares
<b>WOCE</b>	World Ocean Circulation Experiment

## List of Symbols in order of appearance

$^{\circ}$	Degree
$s$	Time-varying strain
$l$	Distance
$\delta l$	Change in distance
$c$	Speed of light
$\Delta V$	Change in velocity
$m_1, m_2$	Mass 1 and mass 2
$X, Y, Z$	Position components
$P$	Period
$a$	Semi-major axis
$F_1, F_2$	Forces 1 and 2
$\ddot{R}_1, \ddot{R}_2$	Accelerations of bodies 1 and 2
$\bar{R}_1, \bar{R}_2$	Distance from origin to masses ( $m_1$ and $m_2$ respectively) in inertial three-dimensional space
$\bar{r} = (\bar{R}_1 - \bar{R}_2)$	Relative position vector (i.e. distance between two bodies $m_1$ and $m_2$ )
$r$	Magnitude of relative position vector
$r^3 =  \bar{R}_1 - \bar{R}_2 ^3$	Magnitude cubed of relative position vector
$G$	Universal gravitational constant = $6.6726 \pm 0.0005 \times 10^{-11} \text{ m}^3 / \text{kg} \cdot \text{s}^2$
$M_{total}$	Total mass of system (= $m_1 + m_2$ , )
$\mu$	Gravitational constant = $GM_{total}$

$\ddot{\vec{r}} = \frac{-\mu}{r^3} \vec{r}$	Fundamental equation of motion
$\mu_{SUN}$	Sun's Gravitational constant (= $1.32712428 \times 10^{20} \text{ m}^3/(\text{solar sec})^2$ )
$\mathbf{X}(t)$	State vector composed of position and velocity components
$t$	Time that represents the State
$\dot{X}, \dot{Y}, \dot{Z}$	Velocity components of the state
$\vec{e}$	Eccentricity vector (where $e$ is the eccentricity scalar)
$\vec{h}$	Angular momentum vector (where $h$ is the angular momentum scalar)
$\vec{r}$	Position vector (where $r$ is the magnitude of position)
$\vec{v}$	Velocity vector (where $v$ is the magnitude of velocity)
$r_p$	Periapsis
$r_a$	Apoapsis
$i$	Inclination
$\hat{K}$	Unit vector in the K-direction
$\Omega$	Longitude of the ascending node
$\hat{I}$	Unit vector in the I-direction
$\vec{n}$	Line of nodes
$n_i$	The $i^{\text{th}}$ element of the line of nodes vector, $\vec{n}$
$\omega$	Argument of periapsis
$e_k$	The $k^{\text{th}}$ element of the eccentricity vector, $\vec{e}$
$\nu$	True anomaly
$\tau$	Time of periapsis passage
$M$	Mean anomaly

$E (F)$	Eccentric Anomaly for an ellipse or circle (comparable quantity for a hyperbola)
$p$	Semi-latus rectum
$\tilde{\omega}$	True longitude of periapsis passage
$r$	Orbit radius
$n$	Mean motion
$\phi$	Argument of latitude
$\gamma$	First Point of Aires
$\varepsilon$	Obliquity of ecliptic ( $\sim 23.5^\circ$ )
$I, J, K$	Three-dimensional axes in Geocentric coordinate system
$\alpha$	Right ascension
$\delta$	Declination
$R, S, W$	Three-dimensional axes on the Satellite Radial coordinate system
$\lambda$	Longitude
$\phi_{gd}$	Latitude
$\rho$	Ideal range, magnitude
$\bar{\rho}$	Range vector
$X_{sc}, Y_{sc}, Z_{sc}$	Position components of the spacecraft
$X_{ts}, Y_{ts}, Z_{ts}$	Position components of the tracking station
$\bar{u}_x, \bar{u}_y, \bar{u}_z$	Position unit vector components
$\dot{\rho}$	Range rate magnitude

$\dot{\rho}$	Range rate vector
$\dot{X}_{sc}, \dot{Y}_{sc}, \dot{Z}_{sc}$	Velocity components of the spacecraft
$\dot{X}_{ts}, \dot{Y}_{ts}, \dot{Z}_{ts}$	Velocity components of the tracking station
$\rho L_{ij}$	Inter-spacecraft range, where $i = 1, 2, \text{ or } 3$ and $j = 1, 2, \text{ or } 3$ , and $i \neq j$
$X_{i,j}, Y_{i,j}, Z_{i,j}$	Relative position components of the inter-spacecraft legs
$L_{12}, L_{23}, L_{31}$	Inter-spacecraft legs
$\dot{\rho} L_{ij}$	Inter-spacecraft range rate, where $i = 1, 2, \text{ or } 3$ and $j = 1, 2, \text{ or } 3$ , and $i \neq j$
$\dot{X}_{i,j}, \dot{Y}_{i,j}, \dot{Z}_{i,j}$	Relative velocity components of the inter-spacecraft legs
$\theta$	Elevation angle
$\hat{X}$	Estimated trajectory (or state)
$X$	True trajectory
$X^*$	Nominal trajectory
$O - C$	Residual Error, or literally "Observed minus Calculated"
$n\text{-dim}$	n-dimensional column vector of number of state parameters
$p\text{-dim}$	p-dimensional column vector of observations
$Y_i$	Column vector of observations ( $p$ -dimensional)
$t_i$	Discrete time
$\dot{X}(t)$	Time derivative of the state vector ( $= F(X^*, t)$ )
$G(X_i^*, t_i)$	Observation-state relationship
$\varepsilon_i$	Errors in the observations
$x$	State deviation vector, $n \times 1$
$y$	Observation deviation vector, $p \times 1$



$A(t)$	Partial of the time derivative of the state deviation vector evaluated on the reference trajectory
$\tilde{H}$	Observation-state mapping matrix
$x_k$	Unknown initial state
$\Phi(t_i, t_0)$	State transition matrix
$I$	Identity matrix, $n \times n$ matrix
$y_i$	Observation deviation vector
$f$	The number of rows in a batch of observations
$m$	$= p \times f$ , total number of observations
$H$	Observation sensitivity matrix
$J(x)$	Cost function
$\hat{x}_o$	Least squares correction of the epoch
$w_i$	Elements of the weighting matrix
$W$	Weighting matrix
$P_{cov}$	Covariance matrix
$\bar{x}_k$	A priori estimate
$\bar{w}_k$	A priori weighting
$(R_{cov})_i$	General specified covariance
$B$	Unspecified linear mapping matrix, or Baseline in reference to VLBI
$R$	Observation noise covariance matrix (where $W = R^{-1}$ )
$K_k$	Kalman gain matrix
$\bar{P}_k$	A priori covariance matrix
$P_k$	A posteriori covariance matrix

$\gamma_{\text{GEO}}$	Geocentric parallax
$R_E$	Radius of the Earth
$R_{E\text{-LISA}}$	Distance between Earth and center of LISA constellation
$L_{ij}$	Leg length
$a_E$	Distance from Sun to the Earth
$R_{vs}$	Initial position of center of LISA constellation
$V_{vs}$	Initial velocity of center of LISA constellation
$M$	Information matrix
$N$	Reduced residual vector
$\bar{R}_i$	Position vector
$\bar{V}_i$	Velocity components of the State Vector (i.e. [ $\dot{X}$ , $\dot{Y}$ , $\dot{Z}$ ] )
$X_E, Y_E, Z_E$	Position components of Earth
$M_{SC}$	Scaled information matrix
$N_{SC}$	Scaled reduced residual vector
$W_{sc}$	Scaled weighting matrix
$SC$	Scaling Matrix
$\Delta T$	Time-tag error
$\Delta \dot{T}$	Clock rate offset
$\Delta \rho$	Range error due to clock instability, or delta one-way range in reference to VLBI
$RTL T$	Round trip light time
$\sigma_Y(T)$	Allan Standard deviation
$\Delta \dot{\rho}$	Range rate frequency instability

$\sigma$	Standard deviation, also referred to as sigma
<i>RMS</i>	Root-Mean-Square
<i>RSS</i>	Root-Sum-Square, or normal magnitude of the vector
<i>norm</i>	The normal magnitude of the vector
<i>ABSr</i>	Absolute OD accuracy in position
<i>ABSv</i>	Absolute OD accuracy in velocity
$Li_{rest}$	Estimated distance to LISA where $i = 1, 2, 3$
$Li_{vest}$	Estimated velocity to LISA where $i = 1, 2, 3$
$Li_{tru}$	True (or nominal) distance to LISA where $i = 1, 2, 3$
$Li_{vtru}$	True (or nominal) velocity to LISA where $i = 1, 2, 3$
<i>RELr</i>	Relative OD accuracy in position
<i>RELv</i>	Relative OD accuracy in velocity
$\Delta L13r, \Delta L21r, \Delta L32r$	Differenced norm of the estimated and true spacecraft positions
$\Delta L13v, \Delta L21v, \Delta L31v$	Differenced norm of the estimated and true spacecraft velocities
$\mu_{ij}$	Correlation coefficients (where $i$ and $j$ equals position ( $X, Y, Z$ ) and/or velocity ( $U, V, W$ ) components)
$U, V, W$	Also used to indicate three-dimensional velocity components
$\sigma^2$	Variance
$\Xi$	Number of parameters to be determined from fit
<i>PU</i>	Parameter Uncertainty

## CHAPTER 1: Introduction

The Laser Interferometer Space Antenna (LISA) mission is a joint effort between the National Aeronautics and Space Administration (NASA) and the European Space Agency (ESA) currently scheduled for launch in 2015 [1]. LISA will function for a five year nominal mission, plus an extended mission of at least 3.5 years once it is placed in its operational heliocentric orbit that trails Earth by about 20 degrees. LISA's primary objective is to detect and study gravitational waves in the low frequency range of  $10^{-4}$  to  $10^{-1}$  Hertz (Hz), a region much too low to be perceived by Earth-based observatories. LISA will test Einstein's theory of general relativity and it is also expected to give further insights into astrophysics.

LISA a three-satellite constellation, will measure the relative change in length of adjacent legs of the formation down to picometer ( $\times 10^{-12}$ ) precision. Each leg of the constellation is separated by five million kilometers (km). The leg-length measurement is referenced to a free-floating proof mass that will be shielded from non-gravitational forces inside the optical assembly of each spacecraft using laser interferometry. The relative change in leg length provides information about the direction and polarization amplitude of each gravitational wave.

Precise orbit determination (OD) methods are required to initially track LISA and direct each satellite into its proper orbit. During mission execution, accurate OD is required to

maintain the appropriate orbit configuration precisely given the extreme sensitivity of scientific data quality to leg length and pointing deviations. OD involves deriving an accurate state vector or initial orbit from which to propagate and map the orbit to a future time and then refining the result to model the real world errors.

There are two places in the LISA orbit where OD is particularly important. The first is at the end of cruise phase. At this stage, the Propulsion Modules (P/M) are still actively navigating and correcting the spacecraft orbit thus inter-spacecraft data will not be available. Once the last Trajectory Correction Maneuver (TCM) has occurred and the OD tracking is confident, the propulsion modules can be ejected. The force of the P/M drop-off will give the spacecraft state a "bump." The second crucial OD phase is the commissioning period, roughly a three-month period when LISA will be close to its nominal orbit. Inter-spacecraft lasers will be linked and small adjustments to each satellite's orbit will be made using micro-thrusters. OD continues to track the spacecraft until the OD is confident and the science of the mission can officially begin. The spacecraft will now be passively controlled using Drag-Free control, and no more large corrections can be made to the orbit during the course of the mission. Hence, OD is crucial to track these spacecraft as they precisely initialize their relative positions, so that the mission can be successful for the scientific operations throughout the nominal and any extended portions.

Accurate OD is extremely crucial for successful LISA mission science (i.e. laser interferometry along leg lengths of the constellation) since only passive Drag-Free

control will be employed once in operational orbit. The goal of this thesis is to investigate methods to obtain more accurate absolute (with respect to a tracking station on Earth) and relative (with respect to an adjacent spacecraft) OD. This will be done using a weighted batch least squares processor to simulate different combinations and types of observations. The unique aspect of this research is its incorporation of LISA's inter-spacecraft range and range rate into the OD process that supplements the more standard Deep Space Network (DSN) range, range rate, and directional (right ascension and declination) measurements. Although we study accuracy with the full suite of measurements, we also study accuracy when subsets of all available measurements are used. Such use of "suitable combinations" of observations with a least squares process was studied by researchers as early as Gauss [2]. Currently soft requirements on the mission call for absolute position accuracy to within one kilometer (km) and a relative position accuracy between 300 meters (m) and 20 m.

Hechler and Folkner [4] performed a preliminary study using inter-spacecraft data for LISA with its onboard laser tracking system for exclusively the range data type. They found that the incorporation of laser tracking is crucial in determining the arm lengths and achieving tighter accuracy requirements during the experimental phase of the mission, although ground tracking (i.e. range and Doppler) was found to be sufficiently accurate for the transfer and delivery phases of the mission. Hence, this motivated the need to investigate using inter-spacecraft data as a means to improve the OD accuracy.

## 1.1. Thesis Objectives & Contributions

This thesis examines the benefits of including relative inter-spacecraft laser data as well as traditional ground tracking data in the OD process to improve both the absolute and relative state estimates. In this thesis, "relative data" is defined as inter-spacecraft range and inter-spacecraft range rate, quantities anticipated to be available from the spacecraft-to-spacecraft interferometry. Earth-based range and Doppler data can be provided by the Deep Space Network (DSN) package, while the angle (i.e. right ascension and declination) data simulates Very Long Baseline Interferometry (VLBI).

It is well-known in OD that increasing the amount of data and the time spent observing the target improves accuracy [5]. However, it is not always the case that accuracy improves substantially with the inclusion of new data. This thesis comprehensively evaluates how absolute as well as the relative OD accuracy improves with the inclusion of various measurement combinations. Measurement sets will also be evaluated by their ability to fit within mission constraints and requirements. The effects of longer tracking arcs, greater frequency of measurements, noise, and biases will also be investigated. Our initial hypothesis is that the best OD results will be obtained with the maximum measurement set, including DSN (Earth-based range and range rate), VLBI (angular data), and inter-spacecraft data (inter-spacecraft range and range rate). Specifically, we expect the addition of relative information to yield improvements in both absolute and relative navigation performance.

## 1.2. Overview of Thesis

This thesis first overviews the LISA mission in Chapter 2 with a specific focus on LISA's unique orbit geometry and configuration. During the LISA mission overview, key constraints and technological challenges to this mission will be noted. Related planned and completed missions on which LISA bases elements of its technology will be discussed.

Chapter 3 presents the key math and orbital dynamic topics needed in this research. Also, the measurement types, aspects of estimation and tracking data error sources are presented. Chapter 4 describes the orbit determination method used for this work, the weighted batch least squares method. Other methods such as linear, unbiased, minimum variance (LUMV) and sequential methods will be discussed for completeness. A brief OD history is included to provide perspective and further validate choices of methods used in this work.

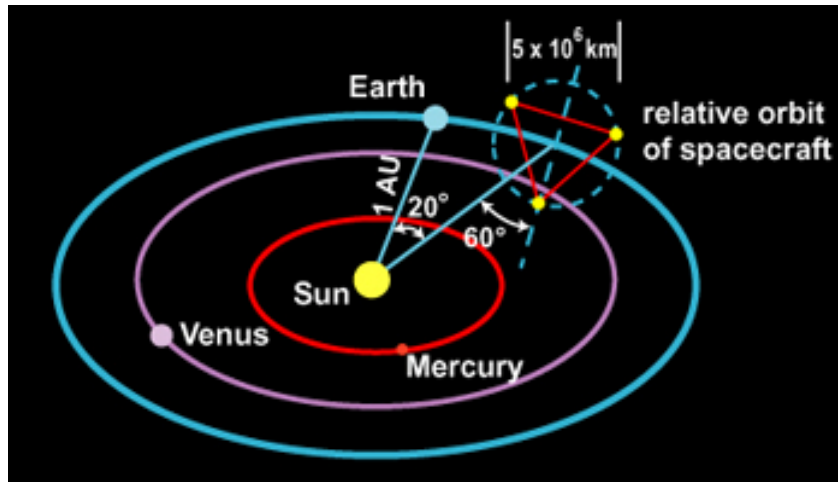
In Chapter 5, the specifics of the nominal LISA orbit, measurement inclusion in the least-squares algorithm, test cases and implementation detail of the simulation are described. The assumptions in the model, validity of the filter, and orbit accuracy test methods will also be defended. Chapter 6 presents the results of the thesis, while Chapter 7 draws conclusions and offers suggestions for future work.



## Chapter 2: Background

The Laser Interferometer Space Antenna (LISA) planned for launch in 2015 is a Cornerstone Mission in ESA's Cosmic Vision Programme, and is a Great Observatory for NASA's Structure and Evolution of the Universe 2003 Roadmap: Beyond Einstein Program (Astronomy and Astrophysics Division) [6]. LISA is managed by NASA-Goddard Space Flight Center (NASA/GSFC) and will be operated by NASA-Jet Propulsion Laboratory (JPL). LISA has a five year nominal mission once in operational orbit, plus a possible 3.5 year extended mission [7].

LISA is a constellation consisting of three identical spacecraft, each of which forms a vertex of an equilateral triangle. The distance, or arm length between each spacecraft in the triangle formation, is five million km. The three spacecraft will be in a heliocentric orbit, centered on a reference orbit of one Astronomical Unit (AU), where 1 AU = 149.6 million km from the Sun in the ecliptic plane. The reference orbit center point will trail the Earth by 20 degrees ( $^{\circ}$ ). Though recently some studies have put this point at 23 $^{\circ}$  degrees behind Earth [8], the reference point will remain at 20 $^{\circ}$  in this thesis. There is a tradeoff between choosing 20 $^{\circ}$  which is more desirable for communications purposes because it is closer to the Earth, while 23 $^{\circ}$  is better for the drag-free control since it is farther from Earth and its perturbations. The actual orbit will be optimized as the mission draws closer. The entire constellation will be tilted 30 $^{\circ}$  away from the Sun, at an inclination of 60 $^{\circ}$ , as illustrated in Figure 2.1.



**Figure 2.1:** LISA orbit geometry relative to the Sun and Earth [1]

## 2.1. Mission & Science Objectives

The scientific objective of LISA is to detect gravitational waves using laser interferometry. In order to measure the changes in distance between two freely-floating proof masses, these masses must be isolated from all non-gravitational forces in each spacecraft [9]. Each spacecraft will track changes in its own motion relative to the other two orbiters [10]. Valuable information concerning the direction of the gravitational wave is gained when comparing the relative motion between two spacecraft of LISA as opposed to following a precise absolute path.

LISA aims to be the first space-based detector to identify and study low-frequency ( $10^{-4}$  to  $10^{-1}$  Hz) gravitational waves from three major binary sources: 1) Coalescences of super massive black holes in distant galaxies [1] which are the most violent events in the universe [9] that result from merging galaxies; 2) Stellar-mass black holes spiraling

toward massive black holes; and 3) Compact binary star systems in the Milky Way Galaxy [11]. Binaries are simply two objects that orbit each other. Objects can be neutron stars, stars, and black holes for example. Black holes are objects whose gravity is so strong that light cannot even escape it; therefore they cannot be seen, but there is "strong indirect observational evidence" [1] of their existence. Black holes are believed to be formed by the collapse of more massive stars after their nuclear fuel has been expended [1]. This phenomenon emits gravitational wave energy that LISA will be able to detect.

A ground-based detector called the Laser Interferometer Gravitational-wave Observatory (LIGO) also views gravitational waves. However, LIGO can only view high-frequency waves from transient phenomena such as supernovae and the final minutes of in-spiraling neutron-star binaries. Low-frequency waves simply cannot be detected with a terrestrial-based detector because of Earth's changing gravitational field due to atmospheric effects and ground motions [11].

### **2.1.1. Gravitational Waves**

Gravitational waves are vibrations of space and time, or "ripples" in space-time caused by accelerations of masses in space, thought to weakly interact with intervening matter [9].

Gravitational waves are essentially propagating gravitational fields. They create time-varying strain,  $s$ , of space-time as defined in Equation 2.1:

$$s = \delta l/l \tag{2.1}$$

Where  $l$  is a distance and  $\delta l$  is a change in distance. For LISA, gravitational waves are detected by monitoring the separation of the proof masses. The motion of the source masses can determine further information such as the waveform and the direction to the source. This motion can be inferred by the amplitude and phase variation caused by LISA's orbital motion [11]. Gravitational waves have been predicted by Einstein's Theory of General Relativity, but have never been directly detected [9]. In the 1970s, indirect evidence became available to confirm Einstein's prediction [12].

Einstein's Theory of General Relativity (1916) proposes matter, space and time are related. The gravity of any mass is the deformation or warping of space-time around it, hence an altering of the path of objects that pass close by. Any variation in movement causes disturbances that become waves in space-time [1]. Mass and gravity are positively correlated: an increase in mass relates to an increase in the gravitational force exerted [13].

LISA aims to test Einstein's theory of General Relativity and to probe the early Universe. Since gravitational waves are thought to weakly interact with intervening matter, there is a theory that waves generated from the beginning of time would have penetrated the heat and the density of the Big Bang to travel onwards [1]. This could possibly reveal clues of the formation of the early Universe through a search for the continuous spectrum (frequency range) of gravitational radiation generated at the Big Bang [1], the so-called beginning of time for our Universe. The early universe was opaque and not transparent to light during the first 380,000 years following the Big Bang, so light alone cannot

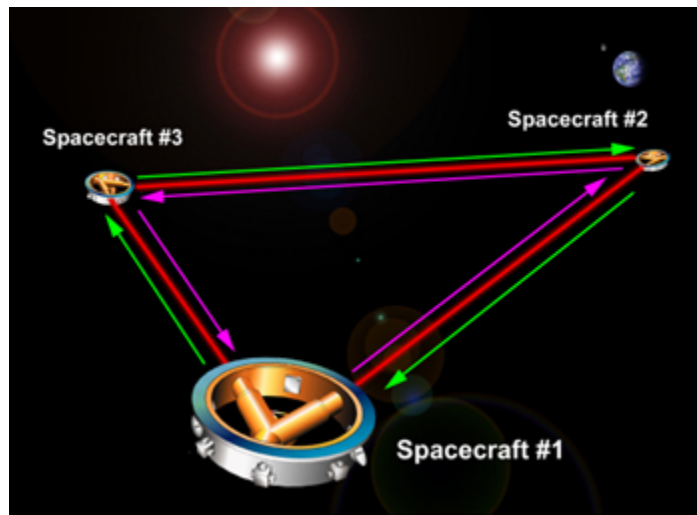
reveal the entire history [9]. Thus, the universe is dark and most of what is known about the universe is through electromagnetic (EM) radiation, which interacts with ordinary matter (i.e. gas and stellar material). However, this "dark" matter interacts with gravity; hence gravitational waves give us additional insight [9].

### **2.1.2. Laser Interferometry**

For LISA to detect gravitational waves, laser interferometry precisely monitors the separation of the three spacecraft proof masses. Interferometers are a popular instrument with the Michelson Interferometer named after American physicist Albert Michelson (1852-1931) a common tool used in astrophysics. The famous Michelson-Morley Experiment demonstrated the absence of "ether" a hypothetical medium which would cause the speed of light to be different in different directions. The experiment found that there is no significant motion of Earth relative to ether. This finding led to Einstein's Theory of Special Relativity in 1906 which states that ether if unobservable simply does not exist. The Michelson-Morley experiment showed that there is no variation in light speed with direction. Laser interferometry takes advantage of the fact that light behaves as a traveling wave that propagates evenly and equally in all directions from its source. Hence, light waves can be identified by their wavelength [1].

Each LISA spacecraft locks the incoming laser signals from the two other LISA spacecraft and transmits them back over the five million km leg length [14]. This is different from the Michelson interferometer, which would reflect the incoming signal.

The "phase lock" on the signal ensures that the phase difference is due only to gravitational waves rather than signal loss due to traveling over the substantial leg length and instrument error. It takes about 32 seconds for the laser to transmit a signal to/from spacecraft to spacecraft (~16 light seconds to, ~16 light seconds back) [14]. It is expected that the changes in arm lengths will be measured as accurately as ten picometers [1]. Figure 2.2 illustrates the laser signal paths as they travel between spacecraft.



**Figure 2.2:** Lasers signals between LISA [1]

## 2.2. LISA Spacecraft Design

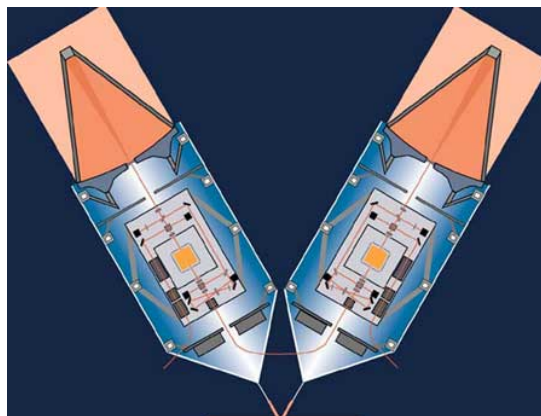
Each 500 kilogram (kg) LISA spacecraft is a short cylinder (2.7 meter (m) diameter, .89 m height) that supports a Y-shaped tubular structure. The Y-shaped structure encases two instruments, which will be discussed below. Solar panels are mounted on a sunshield that extends from the top of the cylinder. The sunshield forms a cover across the top of the cylinder that will prevent sunlight from hitting the Y-shaped structure and the cylinder walls. Figure 2.3 below illustrates one LISA spacecraft. Note the sunshield and solar panels are not included, so that the inside of the spacecraft can be seen. Also, two

X-band 30-centimeter (cm) diameter radio antennae will be mounted on the outside of the spacecraft for communication with Earth.



**Figure 2.3:** The LISA spacecraft [1]

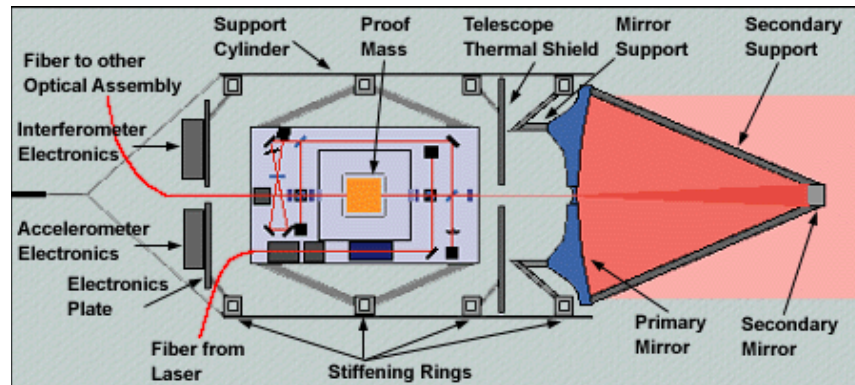
The spacecraft and the Y-shaped structure will be graphite-epoxy composite materials because of their low coefficient of thermal expansion (CTE) [11]. The Y-shaped structure is gold-coated both on the inside and outside and is suspended by stressed-fiberglass bands to thermally isolate it from the spacecraft and to reduce radiative heat transfer as shown in Figure 2.4.



**Figure 2.4:** Y-shaped structure [1]

### 2.3. Instrument Package

Inside the Y-shaped structure is the optical assembly that contains the optical bench, transmit/receive telescope, and low-power electronics pack [15]. There are two optical assemblies per spacecraft. Figure 2.5 details an optical assembly.



**Figure 2.5:** Optical assembly [1]

The optical bench houses the proof mass. Six gold-platinum proof mass cubes (4.6-cm on each side) [10], two per spacecraft, are present in the full LISA constellation. There is a gap between the wall of the structure encasing the proof mass and the proof mass itself. The wall never touches the cube and the spacecraft actively follows the proof mass to maintain the appropriate separation, hence the proof mass continuously "floats" within the spacecraft. For the force on the proof mass to be very low, one needs low stiffness and a low differential displacement of the gap.

The two main systems within LISA are: 1) the Disturbance Reduction System (DRS), which includes the Gravity Reference Sensor (GRS) and 2) the Interferometry Measurement System (IMS). The Disturbance Reduction System works to eliminate non-gravitational forces on the proof mass in the measurement direction and to keep the



spacecraft pointed in the appropriate direction. The DRS reduces disturbance accelerations to the proof masses to less than  $3 \times 10^{-15}(\text{m/s}^2)/\sqrt{\text{Hz}}$  [3]. Included in the DRS is the GRS, with position sensors to measure the proof mass position, while minimizing coupling to spacecraft motion [16]. Two proof masses are in each GRS. The capacitive sensors also are used to locate the proof mass to ensure that the spacecraft is centered upon the masses.

The DRS uses a micro-thrust propulsion system with Micro-Newton ( $\mu\text{N}$ ) thrusters to perform fine control of spacecraft position and attitude, effectively maintaining a "drag-free" flight condition [16] by commanding the spacecraft to follow the motion of the proof masses. Thrusters must counteract all disturbance forces, such as solar pressure with a position accuracy of  $10^{-9}$  m [11]. "Self gravity" between the spacecraft and the proof mass can cause complications in the DRS and create bad measurement noise, the correction of which is an area of current research. In addition, three emerging microthrust propulsion systems are being tested for use on LISA: 1) Field Emission Electric Propulsion (FEEP) using Cesium (Cs-FEEP), or Indium (In-FEEP) propellant, 2) Colloid Micro-Newton Thruster (CMNT), and 3) Precision cold-gas micro-thrusters.

The telescope (30-cm. diameter) is used to transmit and receive laser signals from the other spacecraft. The two proof masses are located in each spacecraft  $60^\circ$  apart from one another. The telescope has an actuator to correct for inter-spacecraft motion, which causes a  $\pm 1^\circ$  variation in look-angle to remote spacecraft. The telescope uses interferometer optics of the Interferometry Measurement System (IMS), which measures

the change in distance between proof masses to a sensitivity of  $4 \times 10^{-11}$  m (averaged over one second) [22]. The interferometer measures the changes in the distances between the proof masses separated with leg lengths of five million km in the different spacecraft to picometer-level ( $\times 10^{-12}$  m) accuracy. The inertial sensor, mounted at the center of each optical bench, senses the level of disturbance imposed on each LISA spacecraft.

### 2.3.1. Drag-Free Control

Drag-free control is a concept initially proposed in 1972 and discussed extensively by Lange [17]. To achieve drag-free control, an unanchored proof mass is enclosed in a spacecraft isolating it from external contact forces such as atmospheric drag and solar radiation pressure. Internal forces can be ignored in ideal conditions, thus leaving gravitational forces as the only disturbance. Then the spacecraft can follow the proof mass using low-thrust propulsion. The propagated trajectory is then easier to follow and remain accurate for longer periods of time with only gravity affecting the spacecraft [17], [6]. To be explicit, the *external forces* from which one must shield the proof mass from include:

- Atmospheric drag
- Solar radiation pressure
- Solar wind
- Photon-pressure
- Higher order gravitational components
- Third-body effects

The *internal forces* also affecting the proof masses including those listed below often cannot be ignored as well:

- Self-gravity

- Differential gravity
- Electric or magnetic field gradients
- Radiation pressure

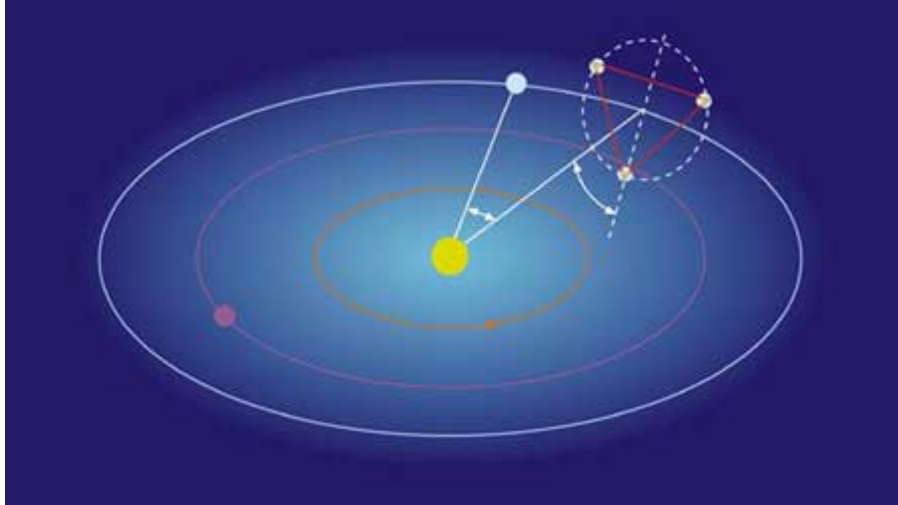
#### **2.4. LISA Launch & Mission Evolution**

The target launch for the LISA mission is 2015 using an Evolved Expendable Launch Vehicle (EELV) to inject all three spacecraft into an Earth escape trajectory. The spacecraft will separate from the launch vehicle and each other soon after launch then will operate independently. After one orbit, the Earth will be 20° ahead of the constellation. During cruise phase: a propulsion module (P/M) on each spacecraft makes the appropriate inclination and energy changes in order to rendezvous with final operations orbit [7]. Two circular solar arrays generate the power for the solar-electric propulsion engines. There will be three deterministic maneuvers for each transfer for each spacecraft [7]. However, maneuver execution errors and OD errors will cause the cruise trajectories to deviate from the planned trajectories such that trajectory correction maneuvers (TCMs) must be administered. The propulsion modules will separate from each spacecraft upon delivery to the final orbit (about 13 months after launch). It is expected to require roughly 27 kg (or 59.5 lbs.) of propellant on each spacecraft to achieve the science orbit [1]. The Micro-Newton thrusters that correct for the solar pressure to maintain drag-free control will use a few kilograms of fuel over the mission.

Following delivery to operations orbit, each spacecraft will be individually evaluated, calibrated and will begin to operate in drag-free mode during the *Element Commissioning Phase*. The *Acquisition Phase* establishes the laser links between spacecraft then the interferometer itself is checked out and calibrated in the *Constellation Commissioning phase* [7]. The science operation begins after this final commission. The spacecraft positions will evolve under gravitational forces only. The spacecraft will be locally controlled in a "drag-free" manner to keep their positions centered about the proof masses contained within them.

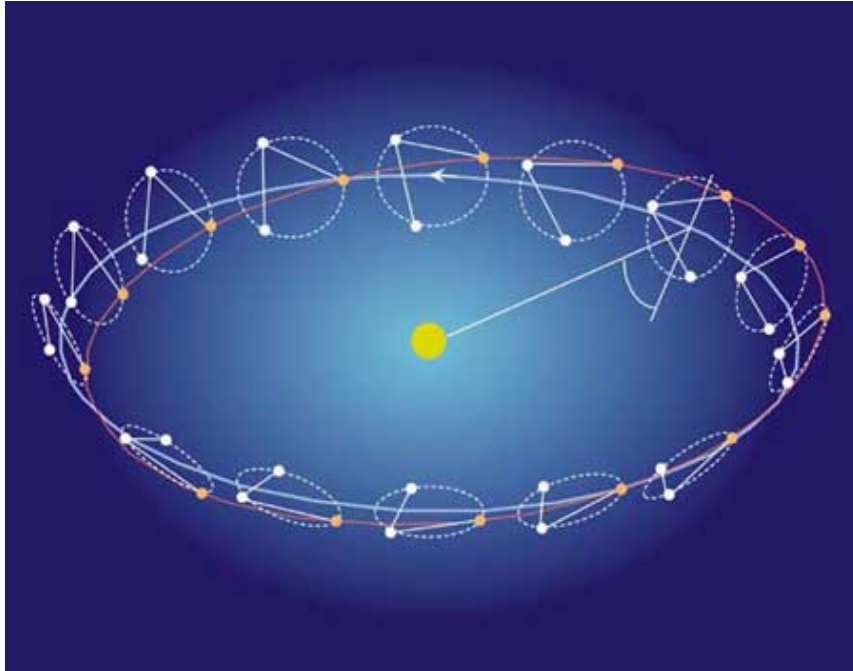
## **2.5. Orbit Geometry & Configuration**

LISA is a constellation of three spacecraft in an equilateral triangle configuration with a  $60^\circ (+/- 1^\circ)$  angle between outward-pointing lasers to relative measurements to the other two spacecraft. The LISA constellation is located  $20^\circ$  behind the Earth. LISA is centered on a reference orbit, a slightly-elliptical Earth-like orbit located one AU from the Sun such that LISA's period, like Earth's, is one year. The reference orbit is in the ecliptic plane to minimize transfer change-in-velocity ( $\Delta V$ ) from Earth [7]. The constellation is inclined  $60^\circ$  with respect to the ecliptic plane of Earth's orbit. The  $60^\circ$  tilt is chosen so that sunlight never enters the interferometer optics; and so that the Sun always illuminates the same part of each spacecraft. Figure 2.6 shows the orbit geometry of the LISA constellation.



**Figure 2.6:** Orbit geometry of LISA constellation without angles [1]

Since each spacecraft traces its own path around the Sun, the triangular formation also changes orientation by performing a complete  $360^\circ$  clockwise rotation once over this same one-year period. Orientation changes are beneficial to help determine the direction of the gravity wave source. The LISA constellation forms an epicycle, or an orbit within an orbit. When viewed from the Sun or Earth, LISA appears to rotate about its center once per year. The rotation is clockwise as viewed from the Sun. Each spacecraft traces an elliptic path as shown in Figure 2.7.



**Figure 2.7:** LISA orbit evolution over an entire year [1]

Sweetser [7] describes the orbital motion pattern shown in Figure 2.7, an orbital mechanics phenomenon that only occurs when a plane is tilted  $60^\circ$ . When a reference point on a circular orbit around a central body is at the center of a plane that is inclined  $\pm 60^\circ$  from the orbital plane; all points on the plane maintain a fixed distance from the center to first order. The plane rotates once around its center point per orbit. Dhurandhar et al [18] mathematically validate this theory using the Clohessy-Wiltshire equations.

The distance between the spacecraft, referred to as the arm length, is to remain uniform to keep the frequency response of the mission at the chosen optimum [19]. The leg length is five million km and is measured between proof masses. This large separation helps to improve the sensitivity required for detecting low-frequency gravitational waves and requires an appropriately sized laser and telescope. Furthermore, constant distances

between the spacecraft minimize problems associated with the inter-spacecraft Doppler shift of the laser signals and allows for optimum frequency response.

## **2.6. Constraints & Technological Challenges**

Criteria for choosing the LISA orbits included satisfying at least ten nonlinear constraints imposed by instrument and communication limitations [14]. The constraints are that the leg lengths be kept within +/- 2% of the nominal, the interior angles of the formation must be kept from 58.5°-61.5°, the formation should be no more than 20° behind the Earth for communication and OD requirements, and all three leg lengths must change at a rate less than 15 meters per second (m/s) [14]. LISA's station at 20° behind Earth is a compromise to: 1) Reduce Earth's gravitational perturbations on the constellation, such as Earth's gravity which reduces with distance [7] according to an inverse square relationship, 2) Satisfy launch vehicle constraints, and 3) Enable communication given power requirements as a function of distance from Earth.

Two requirements on the operational orbit of LISA are that: 1) the leg length is  $5 \times 10^6$  km +/-  $5 \times 10^4$  km [7], the minimum of which allows for the low frequencies that LISA is designed to detect, and 2) the magnitude of the range rates between spacecraft must meet the 15 m/s constraint described above. These constraints on LISA are present because the leg length range (+/-  $5 \times 10^4$  km) signifies orbit design feasibility. Also, the range

rate magnitude specifies angle ranges that onboard interferometers must accommodate [7] throughout the mission (nominal and extended).

## **2.7. Related Missions**

Since the 1960's, attempts have been made to detect high frequency gravitational waves with small interferometers from the ground [20]. The ground-based Laser Interferometer Gravitational-Wave Observatory (LIGO) which was referenced earlier, is a \$365-million dollar Earth-based gravitational wave detector located in Louisiana and Washington. LIGO detects gravitational waves at a higher frequency than LISA, including transient phenomena such as supernovae [9]. However, LIGO is currently in need of an upgrade to be Advanced LIGO because detectors were found to experience 100-1000 times more jitter than blueprints indicated. Other ground-based interferometers include Japan's TAMA, a British and German collaboration GEO600, and an Italy and France partnership known as VIRGO [20].

A set of missions aim to validate subsystems planned for use on LISA and will contribute to the success of future formation-flying interferometer missions. ESA plans to launch the LISA Pathfinder Mission (formerly SMART-2, Small Missions for Advanced Research and Technology, renamed because it will now test technologies that only apply to LISA) in 2009 in a Lagrange point one orbit. LISA Pathfinder will fly two test packages of proof masses and hardware [10]. The payloads will include the complete



DRS/GRS payload of NASA's Space Technology 7 (ST7) to demonstrate the effectiveness of the Gravitational Reference Sensor (GRS) housing two proof masses. This mission will also validate an interferometer design, the Colloid Micro-Newton Thruster (CMNT) by Busek Co., Inc. U.S. [21], and the LISA Technology Package (LTP) [6] also containing two identical proof masses. The LTP drag-free control system consists of an accelerometer (aka inertial sensor), a propulsion system and a control loop [22]. The LTP will instead have the Field Emission Electric Propulsion (FEED) as thrusters and precision cold-gas micro-thrusters as actuators. The goals of the DRS and LTP are similar, but each uses different technology. LISA-Pathfinder's mission objectives include demonstrating drag-free and attitude control with two proof masses, assessing the feasibility of laser interferometry at LISA's accuracy levels, and testing endurance of instrument and hardware in the space environment [22]. The exact design of LISA's proof mass and drag-free control systems will be determined from the studies and feedback of both ST7 and Lisa Pathfinder [22].

Another Beyond Einstein Mission is Constellation-X. Both Constellation-X and LISA are categorized as Great Observatories in NASA's Beyond Einstein Program, a set of high priority missions that will use X-ray spectroscopy and gravitational waves respectively to study black holes [23]. Constellation-X has a possible launch date in 2011. It will consist of four spacecraft each containing a 1.6 m diameter telescope to measure the spectra of cosmic sources of X-rays [24].

Several other missions have demonstrated technologies applicable to LISA. TRIAD launched Sept. 2, 1972, is the only three-axis drag-free satellite ever flown [25]. TRIAD had a spherical gold-platinum proof mass floating inside of it, unaffected by air resistance and sunlight. Electronic sensors on the spacecraft would fire thrusters if the proof mass approached a wall of the spacecraft to keep it undisturbed from outside forces. The inertial sensors and spacecraft control for LISA are based largely on this mission. It was an experimental spacecraft that flew in a low-altitude orbit until it ran out of fuel. However the spacecraft was tracked for many subsequent years to perform magnetometer experiments.

Space Interferometry Mission PlanetQuest (formerly just SIM) developed by NASA-JPL in collaboration with Lockheed Martin Missiles and Space and Northrop Grumman is scheduled for launch in 2011. PlanetQuest will utilize technology related to laser interferometry [26], with a mission of determining the position and distances of stars increased accuracy using optical interferometry. PlanetQuest, like LISA, will also be in an Earth-trailing solar orbit.

Gravity Recovery and Climate Experiment (GRACE) consists of two twin spacecraft flying in formation [13]. The precise speed and distance between the two spacecraft are communicated via a microwave K-band ranging instrument. The scientific objective of GRACE is to obtain a highly accurate gravity field map of the Earth. GRACE is located 500 km above Earth. GRACE fulfills goals of NASA's Mission to Planet Earth (MTPE) - to observe Earth's gravity field, and responds to many other international organizations

concerns such as World Ocean Circulation Experiment (WOCE) and Climate Variability Program (CLIVAR).

Gravity-Probe B (GP-B) is a NASA spacecraft launched on April 20, 2004 with a scientific objective to examine two of Einstein's general theories of relativity: 1) the geodetic effect – the amount to which the Earth warps local space-time in which it resides, and 2) the frame-dragging effect – the amount to which the rotating Earth drags local space-time around with it [27]. It orbits the Earth at an altitude of 400 miles. GP-B features nine new technologies, including drag-free control, cryogenics, and new manufacturing and measuring technologies. It is also the first satellite to ever achieve both three-axis attitude control and three-axis drag-free control (the whole spacecraft flies around one of its science gyros while the spacecraft orbits the Earth) [28]. Thus far, GP-B has successfully acquired one-year of science data, but the results are not yet released.

CHAMP (CHALLENGING Minisatellite Payload) is a German satellite managed by GeoForschungsZentrum (i.e. National Research Centre for Geosciences, GFZ). It is a geopotential research mission that generates highly precise gravity and magnetic field measurements over a five year period. It also uses drag-free spacecraft control [29].

Cassini-Huygens is collaboration between NASA, ESA and Agenzia Spaziale Italiana (i.e. Italian Space Agency) that uses DSN antennas at all three sites continuously [12]. Cassini was launched in 1997, and orbited Saturn on July 1, 2004. Cassini's probe, Huygens, was dropped onto Titan (a moon of Saturn) in January 2005. Cassini's 40-day

scientific mission began on Nov. 26 2005 and was highly successful. The 40-day mission will be repeated twice over the next two years to vary spacecraft's position to sensitize the measurements to different gravitational wave directions. Cassini's overall mission is to measure gravitational waves using radio transmission between Cassini and Earth to reveal velocity changes through the Doppler measurements [30].

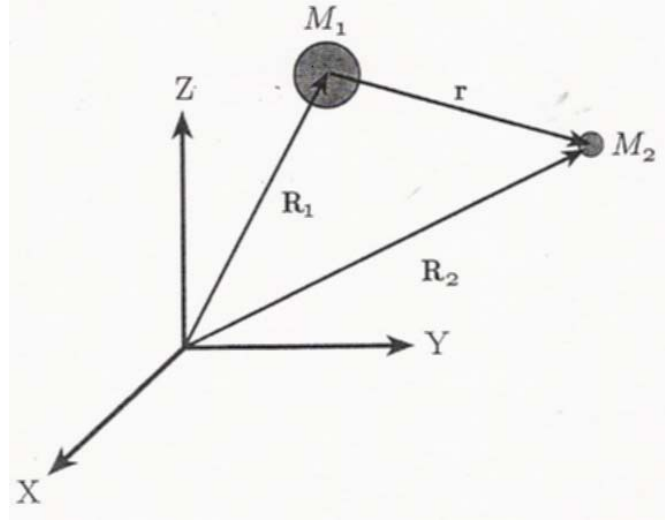
Despite not being launched yet, many successors to LISA are also in the planning stages. Astrodynamical Space Test of Relativity Using Optical Devices (ASTROD) by China with arm lengths of 1-2 AU will search for gravitational waves at even lower frequencies than LISA [31]. The National Astronomical Observatory of Japan is proposing the DECI-hertz Interferometer Gravitational Wave Observatory (DECIGO) with shorter arm lengths than LISA to bridge the frequency gap between the terrestrial and space-based antennas. NASA and ESA will team together again with Satellite Test of Equivalence Principle (STEP) which will further test "drag free" flight. Another NASA: Beyond Einstein Mission, the Big Bang Observer (BBO) also aims to detect gravitational waves to a finer degree than LISA, perhaps reaching toward data from the beginning of the universe. Lastly, direct extensions to the LISA mission have been proposed with shorter arm lengths and lower noise levels, such as Advanced Laser Interferometer Antenna (ALIA) which will detect intermediate mass black holes [32]. Crowder and Cornish [32] have also proposed the addition of a second constellation in a 20-degree Earth-leading orbit to ALIA (the Advanced Laser Interferometer Antenna in Stereo (ALIAS)) and LISA (Laser Interferometer Space Antenna in Stereo (LISAS)).

## CHAPTER 3: Methods & Math Techniques

This chapter begins with a review of two-body motion, deriving the fundamental equation used to generate an ephemeris from Newton's Universal Law of Gravitation. Keplerian orbital elements will be overviewed and well as reference frames used in this work. The measurement types and how they will be calculated are introduced. A description of the methods used by Deep Space Network (DSN) to calculate range and range rate is provided along with the method applied by Very Long Baseline Interferometry (VLBI) to calculate angle measurements. The mathematical formulas used to calculate the inter-spacecraft range and range rate will also be shown. The chapter concludes with a discussion of errors present in each the tracking system.

### 3.1. The Two-Body Problem

The OD problem studied in this work was modeled with respect to two-body dynamics since this "simplified" representation is the only gravitational problem for which a closed-form solution has been found. The two-body problem involves two bodies/masses orbiting under mutual attraction in an inertial reference frame, typically with mass  $m_1$  assumed much larger than  $m_2$  such that  $m_2$  does not affect the motion of  $m_1$ . This is illustrated in the three-dimensional (i.e. with dimensions in the  $X$ ,  $Y$ , and  $Z$  components) inertial coordinate system in Figure 3.1.



**Figure 3.1:** The two-body problem [47]

Note that spherically symmetric bodies, such as planets, behave as point masses. The LISA constellation in this thesis will be treated as a two-body problem between the Sun (the larger mass,  $m_1$ ) and the spacecraft (the negligible mass,  $m_2$ ).

The equations of motion for the two masses are derived from Newton's Universal Law of Gravitation, an inverse square law. Equations 3.1 are based on Kepler's 3<sup>rd</sup> law of planetary motion (i.e. the orbital period is proportional to the semi-major axis to the 3/2 power:  $P^2 \propto a^3$ ).

$$F_1 = m_1 \ddot{\bar{R}}_1 = \frac{-Gm_1m_2}{|\bar{R}_1 - \bar{R}_2|^3} (\bar{R}_1 - \bar{R}_2) \quad (3.1)$$

$$F_2 = m_2 \ddot{\bar{R}}_2 = \frac{-Gm_1m_2}{|\bar{R}_1 - \bar{R}_2|^3} (\bar{R}_2 - \bar{R}_1)$$

$F_1$  and  $F_2$  are the forces and  $\ddot{\bar{R}}_1$  and  $\ddot{\bar{R}}_2$  are the accelerations of the two bodies, where  $m_1$  and  $m_2$  are the masses of body 1 and 2 respectively.  $\bar{R}_1$  and  $\bar{R}_2$  are the distances from the origin to masses  $m_1$  and  $m_2$  respectively in this inertial three-dimensional space,

$\bar{r} = (\bar{R}_1 - \bar{R}_2)$  is the relative position of  $m_1$  with respect to  $m_2$ , while  $r^3 = |\bar{R}_1 - \bar{R}_2|^3$  is the magnitude cubed of the relative position, and  $G$  is the universal gravitational constant, which is equal to  $6.6726 \pm 0.0005 \times 10^{-11} m^3 / kg \cdot s^2$ . Straightforward manipulation of Equations 3.1 then yields:

$$\ddot{\bar{r}} = \frac{-Gm_2\bar{r}}{r^3} - \frac{-Gm_1\bar{r}}{r^3} = \frac{-G\bar{r}(m_2 + m_1)}{r^3} \quad (3.2)$$

Let  $M_{total}$  be defined as  $(m_1 + m_2)$  and  $\mu = GM_{total}$ , yielding Equation 3.3:

$$\ddot{\bar{r}} = -\frac{\mu}{r^3}\bar{r} \quad (3.3)$$

This gravitational equation of motion (EOM) is dependent only on the masses of the two bodies. The EOM in component form represents a set of three, second-order ( $\ddot{\bar{r}}$ ), nonlinear ( $r^3$  term), coupled ( $X, Y, Z$  components of  $r$ ), homogeneous, and autonomous ordinary differential equations [33], where autonomous indicates that a reference epoch for time is not required. The constant  $\mu$ , the product of  $G$  and  $M_{total}$  can be determined to a greater accuracy than  $G$  or  $M_{total}$  alone by accurate Earth satellite tracking [33]. The gravitational constant of the Sun will be defined as,  $\mu_{sun} = 1.32712428 \times 10^{20} m^3/(solar \text{ sec})^2$ . The Equation of Motion will be used to propagate LISA's trajectory, starting with some initial condition, about its heliocentric orbit for one year as will be seen in Section 5.2.2 (Nominal Orbit)

In the least squares problem, the equations of motion must be the exact descriptor of the dynamical system, else dynamical modeling problems may arise [2]. The solution of the EOM requires six independent integrals of constants. The state vector will be defined as

an n-dimensional  $\mathbf{X}(t)$  with dependent variables or constant parameters required to define the time rate of change of the state of the dynamical system [34]. This thesis adopts a typical definition of translational state as a vector of positions and velocities:

$$\mathbf{X}(t) = [X \ Y \ Z \ \dot{X} \ \dot{Y} \ \dot{Z}]$$

composed of position components (X, Y, Z) and velocity components ( $\dot{X}$ ,  $\dot{Y}$ ,  $\dot{Z}$ ), and  $t$  is the specific time that represents this state. The EOM is used to generate an ephemeris representation of motion using the Cartesian state  $\mathbf{X}(t)$  or the Keplerian orbital elements. The most accurate way to generate an ephemeris is by using General Relativity to completely describe the EOM by accounting for many perturbations such as effects from the planets and large asteroids in the solar system (mainly the largest ones i.e. Jupiter, Saturn, etc.). A general relativity accuracy level is usually left to describe a planetary ephemeris over a long period of time (i.e. centuries). The numerically integrated EOM based on classical two-body mechanics as described above without the perturbation terms are adequate for most detailed mission analyses and orbit determination and will be presumed sufficient for this thesis.

### **3.2 Orbital Elements**

Most of the analysis of this thesis is done using Cartesian elements described above to perform Keplerian transformations defined in Vallado [35], Battin [36] and Chobotov [37]. Orbits are defined by a set of six elements. The first is the eccentricity vector,  $\vec{e}$



which distinguishes the shape of the orbit and is defined in Equation 3.4. The eccentricity scalar,  $e$ , is the magnitude of the eccentricity vector.

$$\vec{e} = \frac{\vec{v} \times \vec{h}}{\mu} - \frac{\vec{r}}{r} \quad (3.4)$$

where  $\vec{h}$  is the angular momentum vector and is the cross product of the periapsis-pointing position  $\vec{r}$  and velocity vectors  $\vec{v}$ ,  $\vec{h} = (\vec{r} \times \vec{v})$ ,  $\mu$  is the gravitational constant of the central body. The period,  $P$ , of an orbit is derived from Kepler's 2<sup>nd</sup> (i.e. equal areas swept in equal intervals of time) and 3<sup>rd</sup> (i.e. square of the period of the planet is proportional to the cube of the semi-major axis) laws. The period is defined as:

$$P = 2\pi \sqrt{\frac{a^3}{\mu}} \quad (3.5)$$

where  $a$  is the semi-major axis that determines the size, or altitude of an orbit. The semi-major axis can be calculated many ways, from the vis-viva integral or from using the extreme points of the orbit, i.e. the periapsis,  $r_p$ , and the apoapsis,  $r_a$  as in Equation 3.6:

$$a = \left( \frac{2}{r} - \frac{v^2}{\mu} \right)^{-1} = \frac{r_a - r_p}{2} \quad (3.6)$$

Where,  $r_a = a(1+e)$  and  $r_p = a(1-e)$ ; and  $r$  and  $v$  are the magnitudes of the inertial position and velocity vectors, respectively. Orbits are described as circular ( $e = 0$ ), elliptical ( $e < 1$ ), the hyperbolic ( $e > 1$ ), or parabolic ( $e = 1$ ). Although the LISA orbit is elliptical, some discussion of hyperbolic and parabolic orbits is included for completeness. The inclination,  $i$ , defines the tilt of the orbit as seen in Equation 3.7. It is specifically the

angle between the unit vector,  $\hat{K}$  and the angular momentum vector,  $\vec{h}$  and varies from  $0^\circ$  to  $180^\circ$ .

$$\cos(i) = \frac{\hat{K} \cdot \vec{h}}{|\hat{K}| |\vec{h}|} \quad (3.7)$$

The longitude of the ascending node,  $\Omega$  is the angle measured in the equatorial plane positively from the  $\hat{I}$  unit vector to the location of the ascending node. The ascending node is where the satellite crosses the equatorial plane from South to North, and varies from  $0^\circ$  to  $360^\circ$ . The line of nodes is defined as:  $\vec{n} = \hat{K} \times \vec{h}$ .

$$\cos(\Omega) = \frac{\hat{I} \cdot \vec{n}}{|\hat{I}| |\vec{n}|} \quad (3.8)$$

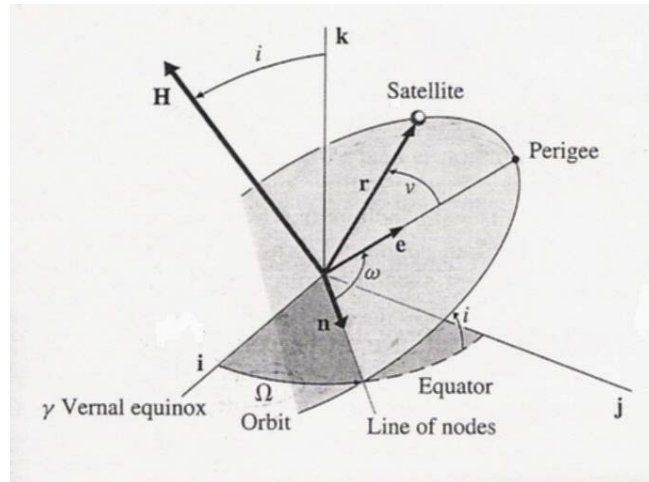
A quadrant check must be employed here that if the  $i^{\text{th}}$  element,  $n_j$ , of the line of node vector,  $\vec{n}$  is greater than 0, then  $\Omega = 360^\circ - \Omega$ . The argument of periapsis,  $\omega$ , locates the closest point of the orbit (periapsis) as measured from the ascending node and varies from  $0^\circ$  and  $360^\circ$ .

$$\cos(\omega) = \frac{\vec{n} \cdot \vec{e}}{|\vec{n}| |\vec{e}|} \quad (3.9)$$

A quadrant check also must be performed on this angle, as it is dependent on the  $k^{\text{th}}$ -component of the eccentricity vector (i.e., if  $e_k$  less than 0, then  $\omega = 360^\circ - \omega$ ). The last orbital element is the true anomaly,  $\nu$ , which is the satellite's current position relative to the location of periapsis.

$$\cos(\nu) = \frac{\vec{e} \cdot \vec{r}}{|\vec{e}| |\vec{r}|} \quad (3.10)$$

Again a quadrant check is required. If the dot product of the position and velocity vectors is greater than zero,  $\vec{r} \cdot \vec{v} < 0$ , then  $\nu = 360^\circ - \nu$ . Sometimes the orbital element time of periapsis passage,  $\tau$ , mean anomaly,  $M$  and even the eccentricity anomaly,  $E$ , can be used instead of true anomaly as the final element. For circular orbits,  $\omega$  and  $\nu$  are undefined and for equatorial orbits both  $\omega$  and  $\Omega$  are undefined. In the case of Earth's special elliptical equatorial orbit, another angle such as the true longitude of periapsis,  $\tilde{\omega}$ , is usually found to remove ambiguity of  $\Omega$  and  $\omega$  [35]. The orbital elements presented thus far are depicted in Figure 3.2, where for a circular orbit the radius,  $r$  to a satellite is equal to the semi-major axis.



**Figure 3.2:** Orbital elements [37]

The reverse procedure involves transforming a set of orbital elements to a state ( $\mathbf{X}(t)$ ). The gravitational parameter,  $\mu_{SUN}$ , for the sun is used in this thesis. The first quantity calculated is the mean motion (rad/sec),  $n$ , Equation 3.11.

$$n = \frac{2\pi}{P} = \sqrt{\frac{\mu}{a^3}} \quad (3.11)$$

Then the mean anomaly,  $M$ , is computed from time of periapsis passage,  $\tau$

$$\tau = t - M/n \quad (3.12)$$

Given the mean anomaly as an initial guess for Kepler's equation, Newton's Method- a common iterative technique- is used to solve for  $E$ , the eccentric anomaly for an ellipse. Note that for a hyperbola, a successive substitution method is used to solve for the comparable quantity,  $F$ . For a parabola, there is no such eccentric anomaly value and it is equal to 0.

$$M = E - e \sin E = n(t - \tau) \quad (3.13)$$

Knowing the eccentricity and the eccentric anomaly; the true anomaly,  $\nu$ , can now be computed if not already given. If the true anomaly was already given as one of the inputs, there is no need to solve Kepler's equation. The eccentric anomaly,  $E$ , (and  $F$ ) can be solved before the mean anomaly,  $M$ , using sine and cosine relations to clarify the quadrant. The magnitude of position,  $r$  in Equation 3.14, is solved for the ellipse using  $E$  and for a hyperbola using  $F$ . For the parabola, the semi-latus rectum,  $p$ , is used instead.

$$r = a(1 - e \cos(E)) \quad (3.14)$$

The argument of latitude, ( $\phi = \omega + \nu$ ) and the angular momentum scalar,  $h$  are computed next. The position and velocity vectors are composed of sine and cosine relations of the longitude of the ascending node,  $\Omega$ ; inclination,  $i$ ; argument of periapsis,  $\omega$ ; argument of latitude,  $\phi$ ; and an expression of the gravitational parameter divided by the angular momentum scalar,  $\mu/h$ . Transformation from elements to state are expressed for position in Equation 3.15 and for velocity in Equation 3.16 [36].

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} r * (\cos(\phi) \cos(\Omega) - \sin(\phi) \sin(\Omega) \cos(i)) \\ r * (\cos(\phi) \sin(\Omega) + \sin(\phi) \cos(\Omega) \cos(i)) \\ r * (\sin(\phi) \sin(i)) \end{bmatrix} \quad (3.15)$$

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = \begin{bmatrix} \frac{-\mu}{h} * (\cos(\Omega) * (\sin(\phi) + e * \sin(\omega)) + \sin(\Omega) * (\cos(\phi) + e * \cos(\omega)) * \cos(i)) \\ \frac{-\mu}{h} * (\sin(\Omega) * (\sin(\phi) + e * \sin(\omega)) - \cos(\Omega) * (\cos(\phi) + e * \cos(\omega)) * \cos(i)) \\ \frac{\mu}{h} * (\cos(\phi) + e * \cos(\omega)) * \sin(i) \end{bmatrix} \quad (3.16)$$

Below, Table 3.1 shows the two orbital element systems used in this thesis.

**Table 3.1:** Orbital element systems

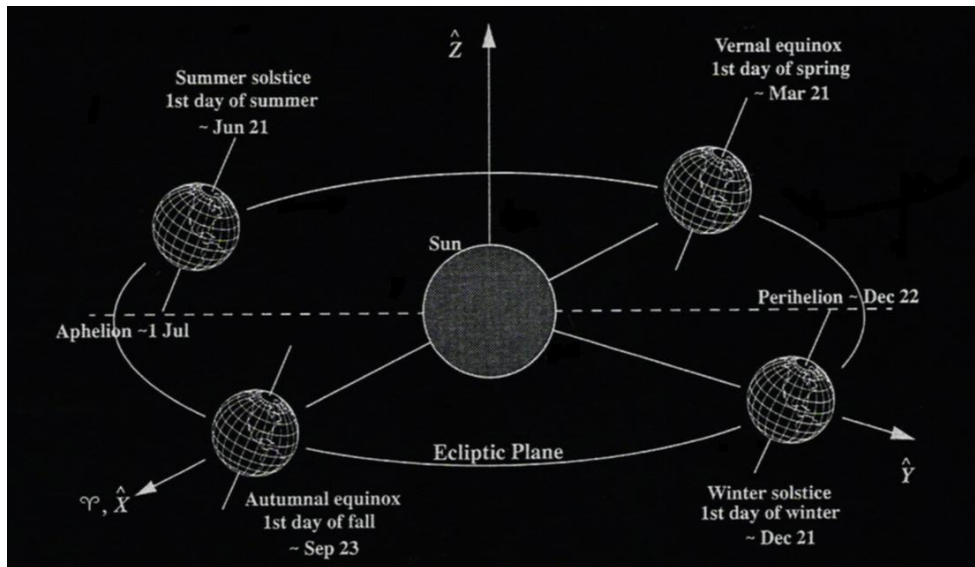
	1	2	3	4	5	6
Classical	$a$	$e$	$i$	$\Omega$	$\omega$	$v(M, \phi, \tau, \text{rarely } E)$
Cartesian	$X$	$Y$	$Z$	$\dot{X}$	$\dot{Y}$	$\dot{Z}$

### 3.3. Reference Frames

The selection of the coordinate reference frames for orbital mechanics are a tradeoff between precision and ease of computation. The main coordinate systems of interest in this thesis are the Heliocentric-Ecliptic ( $XYZ$ ), the Earth-based Geocentric (ECI), and the Satellite-based Radial ( $RSW$ ) systems. An overview of each will be given.

### 3.3.1. Sun-Centered Inertial Coordinate System

Typically, the input ephemeris for the planets is heliocentric and refers to the mean equator and equinox of 1950 [37]. The origin for the Heliocentric-Ecliptic Coordinate System ( $XYZ$ ), also known as Sun-Centered Inertial (SCI), is the center of the Sun as pictured in Figure 3.3, note in the figure that  $\hat{X}, \hat{Y}, \hat{Z}$  is used on the axes instead of  $X, Y,$  and  $Z$ . SCI's fundamental plane (i.e.  $X$ - $Y$  plane) is the ecliptic plane. It uses radial distance, ecliptic latitude, and longitude to locate an object. The principal direction is the vernal (spring) equinox, or first point of Aries,  $\gamma$ [38]. The vernal equinox marks the first day of spring and points in the direction of the  $+X$ -axis. It is when the Sun moves northward through the equatorial plane [37].



**Figure 3.3:** Heliocentric-Ecliptic coordinate system ( $XYZ$ ) [35]

The SCI system is not an inertial frame due to the precession of the Earth [37]. Earth is an oblate spheroid (slightly flattened at the poles) moving in an elliptical orbit which is perturbed by other solar system bodies [38]. The precession causes Earth's spin axis to

wobble and shifts slightly over the centuries, causing the line of intersection of Earth's equatorial plane and ecliptic plane to shift slowly [37].

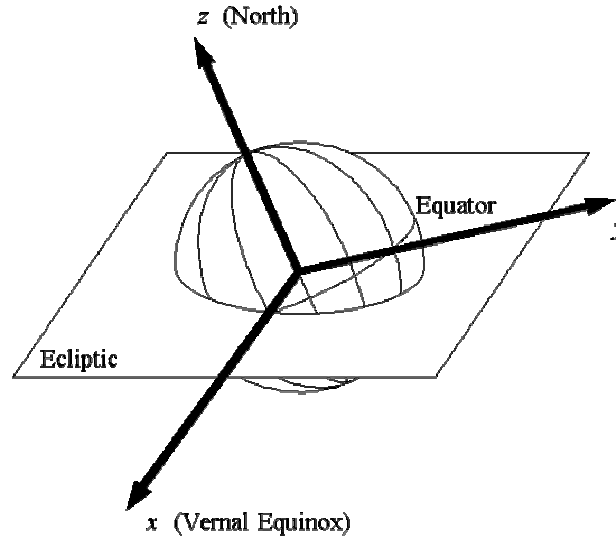
Since integration needs to be conducted in an inertial (non-rotating) frame, the non-inertial heliocentric system is referenced to where the vernal equinox was at a specific epoch, such as the commonly used Mean of January 1.5, 2000.0 frame (i.e. MJ2000) [37].

### **3.3.2. Earth-Centered Inertial Coordinate System**

The general Geocentric Equatorial Coordinate System ( $IJK$ ) is also known as the Earth-Centered Inertial (ECI) system as pictured in Figure 3.4. It is used when collecting relative position and velocity measurements from Earth. ECI's origin is at Earth's center, and its fundamental plane is the equator. The angular difference between where the ecliptic crosses the celestial equator is the obliquity of the ecliptic,  $\varepsilon \sim 23.5^\circ$  [38].

The  $I$ -axis (or  $+X$ -axis) points towards the vernal equinox; the  $J$ -axis (or  $+Y$ -axis) is  $90^\circ$  to the east in the equatorial plane; and the  $K$ -axis (or  $+Z$ -axis) points towards the North pole. This coordinate system is considered inertial, but the equinox and plane of the equator move over time. However, this system becomes inertial by referencing a particular epoch and propagating objects forward in time from this epoch (i.e. the MJ2000 system) [35].

The right ascension,  $\alpha$ , declination,  $\delta$ , and radial distance from the origin,  $r$ , locate an object in this spherical coordinate system.

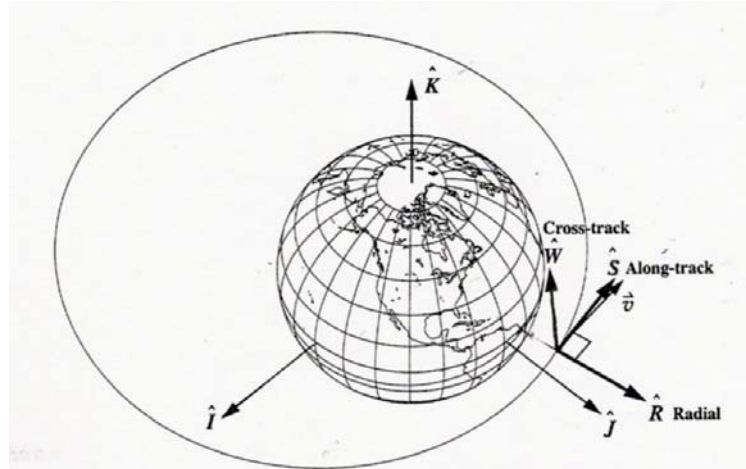


**Figure 3.4:** Earth-Centered Inertial coordinate system (ECI) [35]

### 3.3.3. Satellite Radial Coordinate System

The Satellite Radial coordinate system ( $RSW$ ) moves with the satellite, as it is often used for relative motion. Figure 3.5 illustrates this system. The radial,  $R$ -axis points from Earth's center along the radius vector to the satellite as it moves through an orbit. The along-track  $S$ -axis points in the direction of the velocity vector, and is perpendicular to the radius vector. The cross-track,  $W$ -axis is fixed along the direction normal to the orbital plane [35]. The  $RSW$  system will label graphs of the LISA spacecraft motion.





**Figure 3.5:** Satellite Radial Coordinate System ( $RSW$ ) [35]

Note however that for the purposes of this thesis, this depiction is not entirely accurate. The LISA constellation orbits the Sun, therefore the radial,  $R$ -axis of each spacecraft points from the Sun's center along the radius vector toward each LISA spacecraft as it moves through its orbit. The along-track,  $S$ -axis points in the direction of the velocity vector, which is still perpendicular to the radius vector. But recall that each spacecraft performs a complete revolution about the reference center of the LISA constellation as it orbits the Sun over one year, as will be reflected in the along-track axis. The cross-track,  $W$ -axis completes the right hand coordinate system.

### 3.3.4. Time

The system is propagated over the time frame of one year, specifically a sidereal year which is equivalent to  $\sim 365.25$  solar days. A solar day is longer than a sidereal day by about four minutes. The sidereal day is the time during which the Earth makes a complete rotation relative to a fixed direction (i.e. the vernal equinox), while the mean

solar day is the time during which the Earth makes a complete rotation relative to the mean sun [37]. An epoch is an instant of time or a date selected as a point of reference. An arbitrary epoch of "zero," which is unattached to any specific year and date of the vernal equinox, is used for this study.

### **3.4. Tracking Systems & Observation Types**

The state variables of a spacecraft are not directly observable. Tracking information is required to determine an estimate of true spacecraft motion. The observables will be a nonlinear function of the state variables. Many different types of tracking systems exist to determine instantaneous position or its rate of change. Radio tracking systems, based on radio signals transmitted to or from a ground station are the most commonly used tracking system. These systems are able to perform angle measurements, and obtain distance and velocity information [40]. While radar, used less frequently for spacecraft tracking (i.e. when no active transmitter or transponder is on the spacecraft), is often applied in emergency cases, or for space surveillance tracking [40]. Many tracking systems exist such as: Precise Range and Range Rate Equipment (PRARE), Tracking & Data Relay Satellite System (TDRSS), Global Positioning System (GPS), Satellite Laser Ranging (SLR), Doppler Orbitography and Radiopositioning Integrated by Satellite (DORIS). However, these systems are applicable mainly for low-Earth orbit (LEO) and geosynchronous (GEO) satellites such as TOPEX/Poseidon, the Space Shuttle, and Landsat-4 for tracking and communication support [40] and will not be covered here.

This thesis focuses on orbit determination (OD) based on measurements from the Deep Space Network (DSN), Very Long Baseline Interferometry (VLBI), and the laser interferometry of the LISA spacecraft themselves. Below is a discussion of these observation methods, their typical tracking method, and the equations to obtain the observation. Beyond the scope of this thesis, though related, is the need to address future improvements in radiometric tracking for reasons such as navigation system robustness, reliability, timeliness, accuracy, cost effectiveness, tight targeting requirements and rapid onboard response [41]. In fact, a significant challenge for DSN for example is to dynamically update its current configuration to support the deep space communications demands of the coming decades [42]. Hence, systems such as VLBI, which use arrayed antennas, or a number of smaller antennas rather than a larger single antenna- in order to save operating and production costs, will become more relevant in future missions [42].

#### **3.4.1. Deep Space Network (DSN)**

The Deep Space Network (DSN) is an international telecommunications system managed by JPL at its Network Operations Control Center (NOCC). DSN locates objects in "deep space" loosely classified as locations beyond geosynchronous orbit (about 35,780 km altitude), or simply outside of Earth's Sphere of Influence [35]. NASA primarily uses DSN for interplanetary spacecraft missions. The three Deep Space Communication Centers (DSCC) ground station complexes are located in the sites shown in Table 3.2:

**Table 3.2: DSN DSCC locations [35]**

Site	Antennas	Longitude, $\lambda$ (deg.)	Latitude, $\phi_{gd}$ (deg)	Altitude (m)
Goldstone, CA (GDSCC)	5	-116.848	35.39	1036
Madrid, Spain (MDSCC)	4	-3.685	40.41	670
Canberra, Australia (CDSCC)	3	149.008	-35.32	767

Each antenna or Deep Space Station (DSS) has a schedule of satellites with which it communicates, i.e. transmits and receives data from. Note that in Table 3.2 each ground station is placed approximately  $120^\circ$  apart (longitude) to permit constant observation of the spacecraft as the Earth rotates [43]. The Deep Space Stations are divided into subnets based on their aperture size (26-m, 34-m, and 70-m). The LISA spacecraft will be able to use DSN for tracking.

There are seven DSN data types. The two data types that are the most applicable to LISA are Tracking (TRK) and Very Long Baseline Interferometry (VLBI). TRK uses a closed-loop receiver to determine Doppler, range, predicts, and antenna control. Other DSN data types deal with telemetry, command, and radio science. To examine DSN functionality, consider an Earth-based DSS tracking a satellite. First, at station rise the satellite is visible to the ground station. The tracking station then performs signal acquisition on the spacecraft. During the satellite, or tracking pass, i.e. the period of time in which a satellite flies over the observing site [35], the data is collected and then the satellite "sets" as it falls out of view from the ground station. The spacecraft's trajectory is updated based on angle, Doppler, and range data to guarantee signal acquisition at the second DSN station rise [44].

Communication from Earth to the spacecraft is made within internationally allocated frequency bands [41]. Listed by order of increasing uplink/downlink frequency capabilities are three widely-used bands: S-band, X-band, and Ka-band. Higher frequencies use shorter wavelengths to reduce effects due to charged particles in the ionosphere and solar plasma. This allows for better communication performance and more accurate radiometric measurements [41]. The S-band was developed by DSN for uplink/downlinks in the 1960s; the S/X dual-frequency downlinks were used on spacecraft in the mid-1970s; and by the 1989, the X-band uplink capability was added. By the 1990s, most spacecraft had the capability to transmit and receive X-band frequencies. Now in the 21<sup>st</sup> century Ka-band is typically used. The Cassini spacecraft for example, uses an X-band uplink and a coherent X/Ka-band downlink [41].

### **Range**

The range is defined as the linear distance from the spacecraft to the station along the line of sight. It is computed from the round-trip speed of light travel time of radar signal emitted from ground station antenna to satellite and radiated back to station [40].

Various ranging methods exist (i.e. one-, two-, and three-way), though the most common and accurate ranging and Doppler measurements are obtained via the two-way tracking mode. In two-way tracking, the transmitting and receiving stations and hence the frequency standards are the same [41]. Two-way range is a measurement of the distance based on the round trip light time for a signal sent to the spacecraft to return to ground

station. Thus, the ground station serves as a transmitter and receiver of data. The transit time and frequency, as well as reception time and frequency are determined very accurately using high precision ground station equipment [40]. However, at the extremely large distances needed in some missions two-way tracking is not feasible. One-way ranging uses high precision transit time frequency [40] and just consists of a downlink signal from spacecraft to ground station. The less common three-way range is used with enormous distances. In this case, by the time the signal returns to Earth, the transmission station has rotated out of view, hence a second receiving station is needed [41]. Three-way ranging is a measurement of the distance based on the trip light time of a signal sent to a spacecraft to return to a second ground station [40].

The range data type is computed by taking the norm of the difference of relative position vectors between any two spacecraft as shown in Equations 3.17 below as a magnitude (i.e. ideal range) and a vector. Ideal range  $\rho$  is given by:

$$\rho = \left[ (X_{sc} - X_{ts})^2 + (Y_{sc} - Y_{ts})^2 + (Z_{sc} - Z_{ts})^2 \right]^{\frac{1}{2}} \quad (3.17)$$

The range vector,  $\vec{\rho}$  of satellite with respect to the tracking station, or the instrument is given by:

$$\vec{\rho} = (X_{sc} - X_{ts})\bar{u}_X + (Y_{sc} - Y_{ts})\bar{u}_Y + (Z_{sc} - Z_{ts})\bar{u}_Z$$

Where  $\bar{u}_X$ ,  $\bar{u}_Y$ ,  $\bar{u}_Z$  are the position unit vector terms,  $X_{sc}$ ,  $Y_{sc}$ ,  $Z_{sc}$  are the spacecraft position components, and  $X_{ts}$ ,  $Y_{ts}$ ,  $Z_{ts}$  are the tracking station position components.

## Range Rate

The range-rate or line-of-sight-velocity of the spacecraft relative to the ground station is deduced from the Doppler shift of a radar wave emitted from the ground station transponded by the satellite and received again at the ground station [40].

Similar to range, Doppler also has three techniques of being attained. In one-way Doppler high precision transit time frequency is required, though very few satellites are equipped to perform one-way range and Doppler. The common two-way Doppler is a technique that interprets the phase shift of a signal sent to the spacecraft to determine the speed and direction of the spacecraft [40]. Three-way Doppler interprets the phase shift of a signal sent to the spacecraft, but returns a signal to a second ground station to better determine the speed and direction of the spacecraft [40].

The range-rate data type is attained from Doppler as shown in scalar and vector form in Equations 3.18 below. The range rate depends on the position and velocity of both the satellite and tracking station as shown first with ideal range rate  $\dot{\rho}$ :

$$\dot{\rho} = \frac{\dot{\rho} \cdot \dot{\vec{\rho}}}{\rho} = \frac{[(X_{sc} - X_{ts})(\dot{X}_{sc} - \dot{X}_{ts}) + (Y_{sc} - Y_{ts})(\dot{Y}_{sc} - \dot{Y}_{ts}) + (Z_{sc} - Z_{ts})(\dot{Z}_{sc} - \dot{Z}_{ts})]}{\rho} \quad (3.18)$$

Relative velocity,  $\dot{\vec{\rho}}$  of the satellite with respect to the tracking station, or instrument or other satellite) is described by:

$$\dot{\vec{\rho}} = (\dot{X} - \dot{X}_{ts})\bar{u}_x + (\dot{Y} - \dot{Y}_{ts})\bar{u}_y + (\dot{Z} - \dot{Z}_{ts})\bar{u}_z$$

where  $\bar{u}_x$ ,  $\bar{u}_y$ ,  $\bar{u}_z$  are the position unit vector terms of the satellite with respect to the tracking station,  $\dot{X}_{sc}$ ,  $\dot{Y}_{sc}$ ,  $\dot{Z}_{sc}$  represent the spacecraft velocity, and  $\dot{X}_{ts}$ ,  $\dot{Y}_{ts}$ ,  $\dot{Z}_{ts}$  are the tracking station velocity components. Doppler data is recorded continuously during a tracking pass at DSN. A single Doppler pass determines spacecraft radial velocity, right ascension and declination. The Doppler pass is sensitive to error in declination, which is proportional to  $\sin\delta$ , and vanishes when  $\delta = 0^\circ$ . Therefore, a longer Doppler arc would be needed when passing through  $0^\circ$  declination [41]. However, the angle data in this thesis is assumed to be obtained from VLBI as discussed in the following Section 3.4.2 and not from Doppler. A few days of Doppler data help to deduce the velocity normal to the line of sight and the geocentric range from spacecraft accelerations. Continuous range data has a time signature similar to Doppler and provides spacecraft angular information, geocentric range and range rate. Note for LISA, an error in any spacecraft's heliocentric period leads to increasing oscillations in range rates with respect to the other two spacecraft [8].

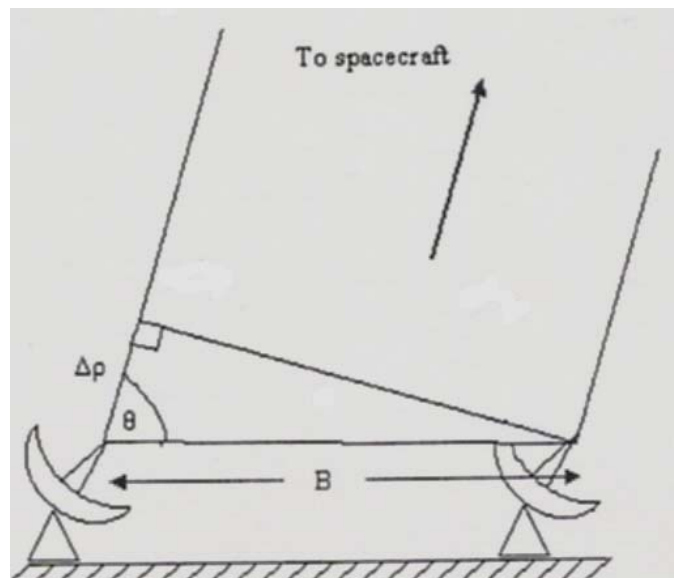
### **3.4.2. Very Long Baseline Interferometry (VLBI)**

Very Long Baseline Interferometry (VLBI) is a more precise method than Doppler and range tracking to directly measure angles and angle rates. Spacecraft angular stations are weakly determined from Doppler and range, as they can be degraded by inaccurate and difficult to model forces such as thruster firings, solar radiation pressure, or spacecraft leaks [41]. Very Long Baseline Interferometry (VLBI) has had long-term use by radio astronomers and was applied to the Voyager Spacecraft (probed Uranus and Neptune) in



the late 1970s. VLBI has since been used on Galileo and the Mars Observer in the 1990s to provide a geometric measure of spacecraft angular position five times more accurate than several days of Doppler and range data [41].

Delta one-way range (DOR,  $D_r$  in Figure 3.6 below) is the measurement of the time difference in seconds that it takes for a signal sent by a spacecraft to reach two different target ground stations. VLBI uses two stations at a very large distance (i.e. baselines,  $B$  in Figure 3.6) apart on Earth to give a DOR time measurement. Case studies from Haeberle et al [42] have found that longer baselines, up to 1000 kilometers apart, increase the accuracy and reliability of the orbit determination, as do the directions of the baselines (i.e. due North, Northeast, East) [42]. The antennas slew to a quasar which is an extragalactic object whose position is known precisely and then back to the spacecraft. Analysis of the data yields precise triangulation that outputs both the angular position and radial distance to the spacecraft.



**Figure 3.6:** Very Long Baseline Interferometry (VLBI) diagram [42]

The angle data used in this simulation is the right ascension,  $\alpha$ , and the declination,  $\delta$ .

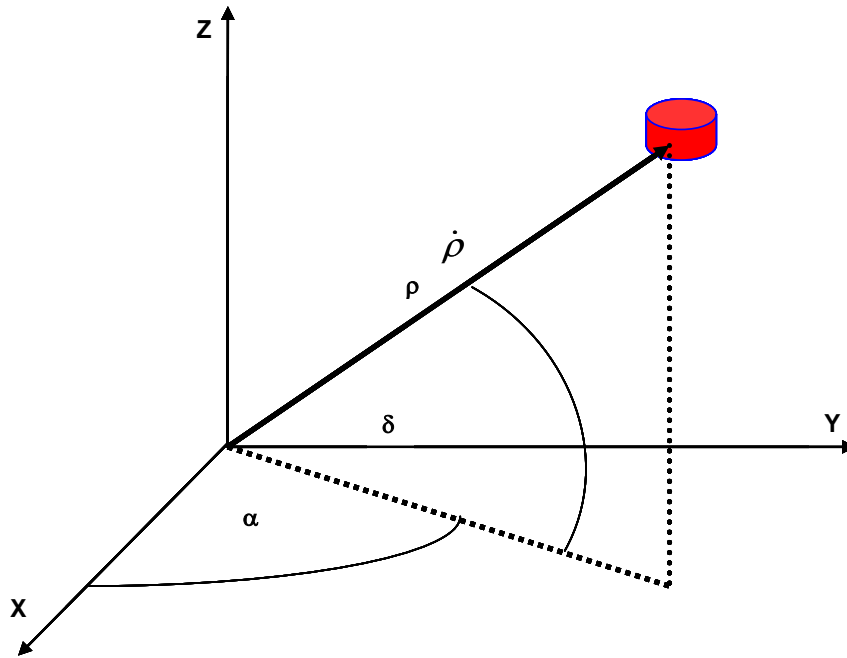
The right ascension,  $\alpha$ , as seen in Equation 3.19 is measured positive to the east in the equator's plane from the vernal equinox direction around the celestial equator in time (0 hours to 24 hours) or degrees ( $0^\circ \leq \alpha \leq 360^\circ$ ) to a plane normal to the equator that contains the object [37].

$$\alpha = \tan^{-1}\left(\frac{X}{Y}\right) \quad (3.19)$$

The declination,  $\delta$ , shown in Equation 3.20 is measured northward ( $0^\circ$  to  $+90^\circ$  at the north celestial pole), and southward ( $0^\circ$  to  $-90^\circ$  at the south celestial pole) from the equator to the object's location  $-90^\circ \leq \delta \leq +90^\circ$  [38].

$$\delta = \sin^{-1}\left(\frac{Z}{\rho}\right) \quad (3.20)$$

Note that the angle information, like range is a function only of vehicle and tracking station position. Figure 3.7 is a summary of the Earth-based DSN and VLBI measurement tracking to a single spacecraft:



**Figure 3.7:** Pictorial representation of Earth-based DSN and VLBI measurements

### 3.4.3. Inter-spacecraft data

The need for spacecraft-to-spacecraft tracking is being prompted by the current and future use of constellation and formation flying missions, close proximity and in situ tracking at Mars and Earth orbit [41]. The science of the mission requires that each LISA spacecraft locks the incoming laser signal from another LISA spacecraft and transmits it back over the five million km leg length. It takes about 32 seconds roundtrip for the laser to transmit a signal to/from spacecraft to spacecraft. LISA measures the distance between two spacecraft (i.e. change in arm length), and the inter-spacecraft range and range rate can be computed as a by-product of the scientific data. The inter-spacecraft data types simulated in this thesis are range and range rate and are obtained using the range and range rate equations of Equations 3.17 & 3.18, replacing the tracking station

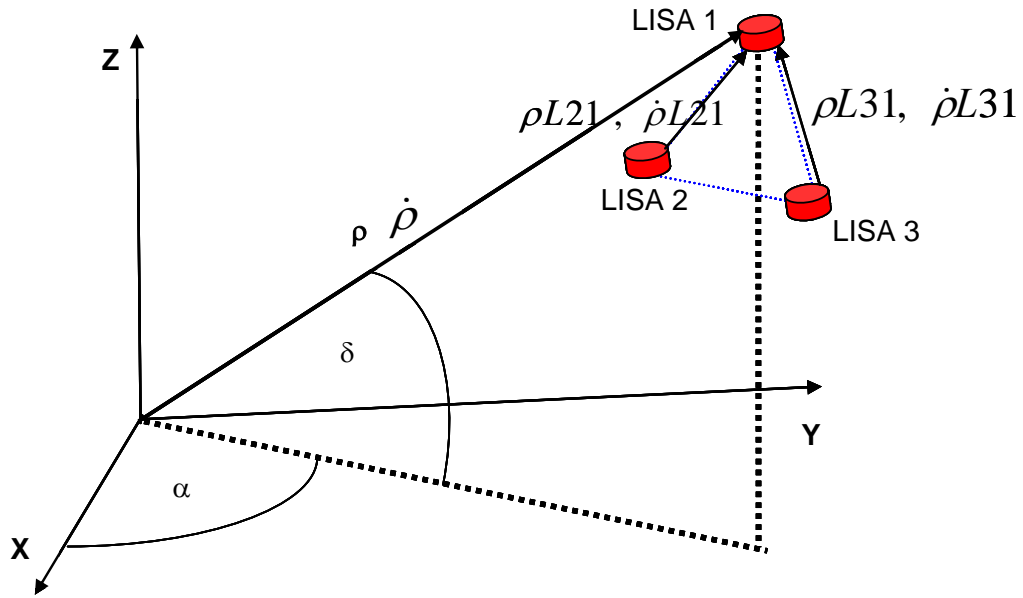
reference with measurements to adjacent spacecraft. Equation 3.21 is the inter-spacecraft range,  $\rho L_{ij}$ .

$$\rho L_{ij} = \left[ (X_i - X_j)^2 + (Y_i - Y_j)^2 + (Z_i - Z_j)^2 \right]^{1/2} \quad (3.21)$$

where  $i = 1, 2, \text{ or } 3$  and  $j = 1, 2, \text{ or } 3$ , and  $i \neq j$ . These reference the leg length. The legs used are  $L_{12}$ ,  $L_{23}$ , and  $L_{31}$ . Inter-spacecraft range rate  $\dot{\rho} L_{ij}$  is seen in Equation 3.22:

$$\dot{\rho} L_{ij} = \frac{[(X_i - X_j)(\dot{X}_i - \dot{X}_j) + (Y_i - Y_j)(\dot{Y}_i - \dot{Y}_j) + (Z_i - Z_j)(\dot{Z}_i - \dot{Z}_j)]}{\rho L_{ij}} \quad (3.22)$$

where the position components are  $X_{i,j}$ ,  $Y_{i,j}$ ,  $Z_{i,j}$  and velocity components are  $\dot{X}_{i,j}$ ,  $\dot{Y}_{i,j}$ ,  $\dot{Z}_{i,j}$ . Figure 3.8 depicts both the Earth-based and inter-spacecraft measurements to LISA 1.



**Figure 3.8:** Pictorial representation of Earth-based and inter-spacecraft measurements

### 3.5. Tracking Data Error Sources

Tracking data error sources limit the accuracy to which the spacecraft orbit can be determined. Observational data is subject to both random and systematic errors [45]. Three main types of errors are: 1) Errors in tracking equipment, 2) Transmission media noise, and 3) Imperfect models of tracking geometry. Errors in tracking equipment can include clock instability, etc. Any errors between frequency reference and the actual transmitted frequency introduce range rate errors. Oscillators onboard the spacecraft help alleviate clock differences by generating a reference signal that is transmitted from the spacecraft to ground. For two-way tracking a hydrogen maser is typically used [41] as an oscillator. However, the Linear Ion Trap standard (LITS) with a cryogenically cooled, compensated sapphire oscillator (CSO) is of an even higher frequency standard than the hydrogen maser. DSN uses observation signals of GPS to measure clock offsets over intercontinental distances to the nanosecond level [41]. Range measurements are also sensitive to clock epoch offsets between transmitting/receiving stations, but they are insensitive to time-tag errors.

Instrument measurement noise is another type of error in tracking equipment. The effects are due to physical instrument accuracy limitations and instrument design trade-offs that introduce random errors, white noise and systematic errors into the system [41].

Likewise, calibration is another aspect of tracking equipment errors, as real-world tracking data is imperfect due to biases and noise. Calibrating the specific instruments used for tracking helps to identify biases and the noise statistics associated with each

measurement [35]. Hence, it is also important to accurately weight the data based on its expected error/noise in order to improve the accuracy of the solution. Since some data types or tracking stations/instruments will be "better" than others. The noises, biases and their weights used in this thesis are presented in Chapter 5.

The next main error type is transmission media, in which charged particles in the interplanetary medium and Earth's ionosphere cause dispersive propagation delays in interplanetary radio signals. Interplanetary data transmission results in group, phase, ionospheric, solar plasma, and charged particle delays and nighttime effects. Group delays in spacecraft and station electronics are a major source of error in ranging systems [41].

Real-world concerns such as sensor site visibility can limit the observable range of azimuth and elevation angles. Visibility limitations are due to mechanical design of the sensor, geographical constraints (i.e. mountains), or political constraints [35]. There are even limitations on range, in that sometimes a satellite really can be too far from the site. It is interesting to note that all sensors combined cannot search the entire volume in which satellites operate and that tracking systems individually can only search a limited amount of space.

## **CHAPTER 4: Orbit Determination**

This chapter begins with a historical overview of the orbit determination (OD) process to give a background to the methods. Then preliminary or classical orbit determination methods will be discussed. The OD problem will then be formulated and the modern statistical OD methods derived from Gauss's least squares theory will be detailed. Emphasis will be placed on the weighted batch least squares method adapted for the LISA OD problem studied in this thesis.

### **4.1. Orbit Determination History**

The earliest evidence of man's interest in the universe was with the Babylonians and Egyptians in 1650 B.C. They used a sexagesimal (base 60) system that is still the standard for modern time-keeping [37]. Aristarchus in 300 B.C. developed the theory that the sun and stars were fixed and the Earth revolved in a circular orbit about the Sun. He was ahead of his time since the prevailing theory then was that the Earth was the fixed center of the universe and the planets revolved around it [37]. Hipparchus (130 B.C.) noted the epicyclical motion of the planets [37]. Claudius Ptolemy A.D. 150 in his book, *Almagest* (translation: Arabic, *The Greatest*) further developed the theories of Hipparchus to create Earth-Centered models to predict the motion of the planets of the inner solar system. The Early Greeks also thought about the size, shape and the composition of the

Earth. However, they fell into two camps: 1) Pythagoras and Aristotle who believed in a spherical Earth; and 2) Anaximenes who believed in a rectangular Earth [37].

By the 16<sup>th</sup> century, a set of European scientists and mathematicians such as Copernicus, Galileo, Kepler, Newton and Leibnitz started making great strides in the fields of mathematics, astronomy and physics that would benefit Orbit Determination. Copernicus (1467-1543) rearranged the Ptolemaic system by placing the Sun at the center, noting that all epicycles of that system had a period of one year [33]. Copernicus also placed the stars in a sphere of very large radius, although his theories were not well received at the time [37]. The first astronomer to use a telescope to observe the 4 moons of Jupiter was the Italian Galileo (1564-1642). Galileo's observations led to the eventual acceptance of Copernicus' earlier heliocentric theory [37]. Interestingly, Galileo was persecuted for believing in such a radical theory, and in 1632 he was forced to recant his beliefs in the Copernican system by the Inquisition and was placed under house arrest for the last ten years of his life [33].

Johannes Kepler (1571-1630) used Tycho Brahe's astronomical observations to deduce that orbits were ellipses with the Sun at one focus to discover his three empirical laws of planetary motion [46]. Kepler's laws of planetary motion are a complete solution to the two-body problem of orbital motion [33]. Isaac Newton (1642-1727) was best known for his development of calculus, a field that Leibnitz claimed that he developed simultaneously and independently [37]. In Newton's *Philosophiae Naturalis Principia Mathematica*, he writes his definitive three laws of motion and law of gravitation, with his assumptions being with respect to an "inertial" coordinate system that is fixed in



space or moving with constant velocity, but not rotating. He showed that Kepler's empirical laws could be deduced from his law of gravitation [46].

In 1801, the rediscovery of the asteroid Ceres marked the first time that astronomers used observations to predict the orbit of an asteroid at a future time [35]. In the 18<sup>th</sup> century, the works of Euler, D'Alembert, Lagrange, Lambert, and Laplace advanced the subject of orbit determination based on astronomical observations by studying the perturbation of planetary orbits [46] and they developed the preliminary OD methods that are still used today. Leonard Euler (1707-1783), author of *Analytical Dynamics* [36], was the first to use mathematical rather than geometric methods to address the problems of dynamics; he studied mathematics under John Bernoulli at the Swiss University of Basel. In 1744, Euler wrote *Theoria Motuum Planetarum et Cometarum* (translation: Theory of the Motion of Planets and Comets), an analytical method for solving a parabolic orbit from three closely spaced observations [35].

The geometrician, Johann H. Lambert (1728-1779), generalized Euler's formulas to include elliptical and hyperbolic orbits; however, his insight sparked analytical solutions from other scientists [35]. In astrodynamics, the solution of the famous "Lambert's problem" allows for the determination of an orbit between any two known position vectors [35]. Joseph-Louis Lagrange's (1736-1813) is known for his contributions to mathematics via theory of numbers, calculus of variation and differential equations just to name a few, however, his main interest was celestial mechanics. Lagrange is best known in astrodynamics for finding stable solutions to the three-body problem (i.e. Lagrange points) [35]. Pierre-Simon de Laplace (1749-1827) created a method of OD based only

on angles that did not need to be close [35], as opposed to Euler's method where small angles are required.

The breakthrough for the field of orbit determination came with the formulation of the least squares method. Both Adrian Marie Legendre (1752 - 1833) and Karl Friedrich Gauss (1777-1855) independently devised least squares. Legendre's work was published in 1806 "Nouvelles methods pour la determination des orbites des cometes" (Translation: "New Methods for Determining the Orbits of Comets"). Gauss devised the method in 1795, but published later in 1809 *Theoria Motus Corporum Coelestium* translation: *Theory of Motion of the Heavenly Bodies*) [2]. Despite, the controversy between Gauss and Legendre, Gauss is credited with the Least Squares Method, which will be explained more in depth in the sections that follow.

Continuing into the 19<sup>th</sup> century, more advances with the use of statistics in estimation processes and refinements of least squares methods were made. Josiah Gibbs's (1839-1903) geometrical method improved on Gauss's method for solving position vectors at the first and third times of the observation [35]. Samuel Herrick (1911-1974) developed a Taylor-Series approach to Gibbs method, allowing problems with short time spans to be solved, a technique highly relevant to modern applications such as closely-spaced radar observations. [35]

R.A. Fisher (1890 - 1962) in 1912 developed the Maximum Likelihood Estimate (MLE) [2], a statistical estimation method that extended Gauss's principle to cover non-Gaussian

error statistics. Though MLE is not a complete theory, his work influenced Kolmogorov and Wiener [35]. Andrei N. Kolmogorov (1903- 1987) and Norbert Wiener (1894 – 1964) independently developed the linear minimum mean-square estimation technique in 1941 and 1942 respectively. The combined Wiener-Kolmogorov filter theory used Gauss' inference that linear equations must be available for the solution of the estimation problem [2].

In 1955, J.W. Follin concocted a recursive approach to derive updated estimates from new measurements [2]. R. E. Kalman (1930- ) published his famous 1960 paper "A New Approach to Linear Filtering and Prediction Problems" on discrete-time, recursive mean-square filtering. Reminiscent of the Legendre-Gauss feud over the ownership of the least-squares method, there was another man named Peter Swerling (1929 - ) who had published a RAND corporation memorandum in 1958. This method was the same as Kalman's except for some more complicated versions of error covariance update matrix equations. However, despite the similarities in their methods, it is Kalman who achieved the fame and is credited with the method [2], though some texts refer to the Kalman filter simply as a Sequential Filter. The recursive algorithms of Swerling and Kalman are a variation of the Least Squares methods of Gauss and Legendre [35]. Richard Bucy partnered with Kalman to work on continuous time versions of the sequential filters [2]. The Kalman-Bucy filter is also known as the Extended Kalman, or Extended Sequential Filter.

Lastly, technology in the forms of better observational instruments and computing power was needed to fully realize the theories of Gauss and the Kalman filter. In the 1960s, positioning of Earth-orbiting satellites was achieved to accuracies of hundreds of meters, due to the limited accuracy of radio frequency and optical measurement systems, force models, analysis techniques and primitive computing technology [47]. The Millstone Hills radar (U.S. by Richard Anderle) used Doppler to determine range rate and to detect Sputnik I and later successfully tracked Sputnik II [35]. In the 1970s, OD accuracy improved to a few tens of meters due to improved force modeling and computing [47]. OD accuracy was in the tens of centimeters by the end of the 1980s, due to significant improvements to Earth's gravity field model, including solid body, ocean tides, surface force effects, and due to further computing improvements [47]. Demands from scientists, especially in the oceanographic and geodetic communities continue to drive more accurate OD. Laser technology is being employed to more precisely track satellites. A highly accurate ephemeris is required to locate the satellite because divergence of the laser is small; hence, an improved ephemeris reduces costly search time. [35].

#### **4.2. Classical Orbit Determination**

Classical orbit determination employs deterministic methods in which observation errors are not considered, and there is equivalence between the number of observations and the number of unknowns. Modern statistical OD methods based on least squares and the sequential algorithms are the focus of this thesis are employed to iteratively refine the

orbit determination (OD) estimate as measurement data is made available. Statistical OD methods recognize the influence of observation error, and therefore try to minimize these errors by using more observations than the number of parameters to be estimated.

Classical OD requires at least six independent measurements to uniquely determine an orbit and derive the set of six orbital elements [40]. The main methods of classical OD fall into two categories: Laplacian and Gaussian. Note that this OD problem is often cast as Lambert's problem, i.e. any problem in which one solves for the orbit given two known position vectors and time-of-flight. Gauss's analytical solution to Lambert's problem works well for closely spaced vectors. Universal Variables is a simple solution for any type of transfer orbit. Battin [36] offers another robust method to improve Gauss's solution to avoid some of the 180 degree transfer difficulties associated with the other two methods [35].

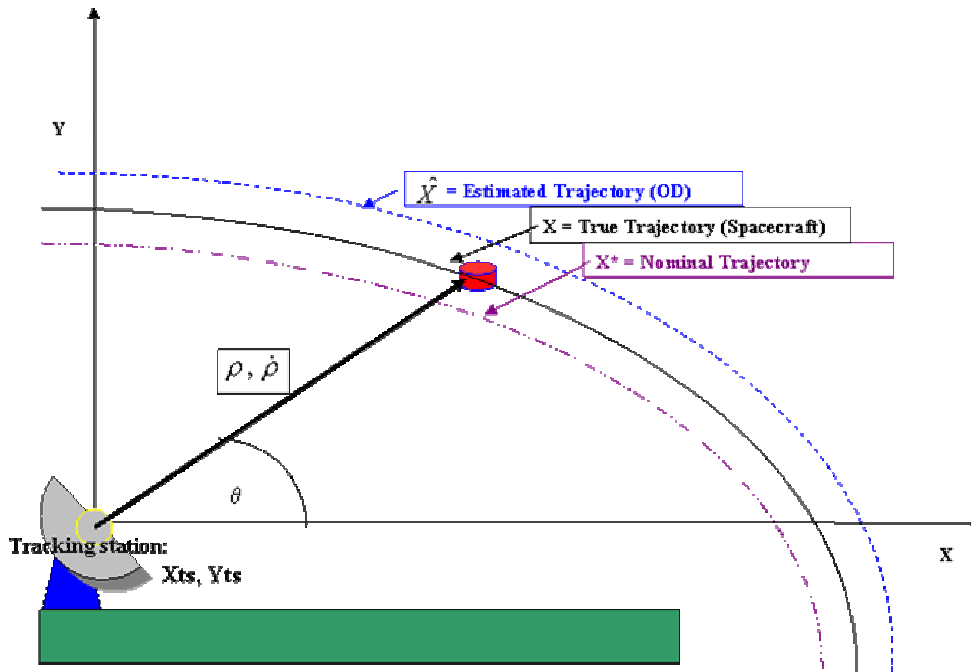
The Laplacian method determines orbits using only optical sightings [35]. Laplace used three sets of angle data to derive the inertial position and velocity of an instant of time in the middle of the observation interval to obtain the orbital elements [40]. The Laplacian method can process data to convergence from different sites at different times [35], though this method is not good for longer tracking arcs if velocity information must be obtained from the interpolation of position measurements [40].

The Gaussian methods work best when angular separation between observations is less than 60 degrees, working extremely well for separations of ten degrees or less, such as

the Angles-only Gauss methods. The double-r Gauss method can be used when observations are far, i.e. days apart [35]. Gibbs, the Herrick-Gibbs, and the Lambert-Universal Variables method can be used in combination with Angles-only Gauss to better parameterize the  $f$  and  $g$  analytic functions needed in the algorithm. The Gibbs and Herrick-Gibbs approach can be used together when given three position vectors and time to find the orbital elements from three sets of widely spaced direction measurements. The Lambert-Euler method, Universal Variables, p-iteration method and the  $f$  and  $g$  series method all are used to solve Gauss' problem. Modern day applications also allows for the ability to solve for position and velocity of the spacecraft given a set of six observations (i.e. range, range rate, elevation, azimuth angles and their rates), such at SITE-TRACK in Vallado [35].

#### **4.3. Orbit Determination Problem Formulation**

The Orbit Determination (OD) problem is to determine the best estimate of the satellite's state at some epoch,  $t_o$ , from observations influenced by random errors using a non-exact mathematical model [45]. The initial state is never known exactly and certain physical constants such as  $\mu$  are only known approximately. The observations of a satellite's motion can only help yield a better estimate of the actual satellite trajectory. Figure 4.1 illustrates the Orbit Determination Problem.



**Figure 4.1:** The Orbit Determination problem

A tracking station located at  $(X_{ts}, Y_{ts})$  in a 2-dimensional  $(X, Y)$  coordinate system makes indirect measurements to a spacecraft at some time with range  $(\rho)$ , range rate  $(\dot{\rho})$  and an elevation angle  $(\theta)$  measurement. The spacecraft orbits along its true trajectory, which is most likely different from the nominal trajectory, the navigation planners have commanded and estimate the spacecraft is following. However, the truth of where the spacecraft is located is never really known- the OD process provides a way to get the best estimate,  $\hat{X}$ , of the true trajectory. The only way to estimate the true state of the spacecraft is with tracking information based on indirect observations of the spacecraft state and the tracking station position. The goal of the OD process is to converge on the indefinite truth.

The calculated values of the observations depend on the nominal state of the spacecraft and the state of the tracking station at the time the observation was made. The OD process computes the difference between the observed and the calculated observations (O-C) which provides the correction needed to better estimate the spacecraft's state. This is also known as the residual error:

$$\text{Residual error} = (\text{O} - \text{C})$$

In an ideal world, one wants these three trajectories to be equal:  $\hat{X} = X = X^*$ . However, the measurements/instruments have noises and biases, and the OD estimation process itself cannot perfectly account for these errors.

#### 4.3.1. Aspects of Estimation

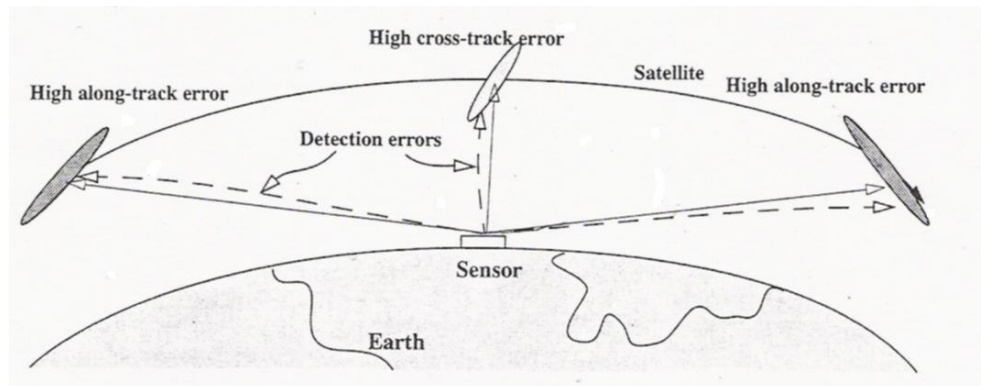
Statistical OD is a very complicated process. First, the data to be used for the OD process may not always be readily available due to sensor maintenance, downtime, and even human interaction that influences the quality of, or even prohibits the observation from reaching the user. Assuming that biases are known and accounted for (i.e. removed) and that real accuracy is known for each data type (i.e. data is weighted accordingly) is [39], the rule *more tracking data = Better Estimate of True Solution* applies. However, continuous data does not exist for all satellites. Data is limited due to visibility of the satellite from the tracking station and the availability of the tracking station sensors to monitor that satellite for example. Having too little data leads to the inability to estimate additional state parameters [35]. Accuracy problems also arise with data accumulated over only a small arc of the satellite orbit. Ideally, one wants *dense*



data on the order of hundreds of data points per satellite pass from just one site, or even from multiple sites. *Sparse* data even from multiple sensors can result in significant error [35].

Viewing constraints or visibility also affects the accuracy of the data the sensor receives. This is due to the satellite physically passing out of view of the ground station, or even geographical (and political) boundaries such as mountains. Viewing constraints are also a factor as the satellite passes directly over the tracking station, a situation with low declination. Given LISA's distance from Earth and the fact that it is centered on Earth's orbit, the constellation will appear at a low declination throughout the mission. In this situation, a small horizontal error could exist, which results in a large plane change or cross-track error. When the satellite is low in the horizon, a small vertical error can result in a large uncertainty over where the satellite is in its orbit, i.e. an along-track error [35].

Figure 4.2 illustrates this phenomenon for LISA:



**Figure 4.2:** Along-track and cross-track errors for LISA [35]

Along-track errors are most detrimental because they affect the ability to get precise timing information; cross-track errors result from sensor misalignment and radial errors are usually small. Fortunately, incorporating data from multiple satellite passes usually reduces these errors in the satellite coordinates [35].

Data types also affect the estimation. The range data type is probably most fundamental and important; it is "crucial to elegantly transforming data [35]." A second general rule that can be inferred is *the more available data you include, the better*. Vallado [39] proposes that ideal availability can be simultaneous range and range rate from multiple locations, or using multiple sensors to obtain observations. The former, though highly accurate due to the triangulation of two simultaneous sites is not practical for most applications (GPS is the exception). The latter of simply using multiple sensors enhances observability and gives geometric diversity to the observations. Sensor calibration, the determination of biases and noise statistics for each individual sensor is crucial.

Implementation is another important aspect of OD, which involves using highly accurate propagators, and high-fidelity force models. The maximum allowable length of the interval between OD updates depends on the quality and quantity of observations, the estimation and propagation schemes, and the types of sensors used. One site alone can do the update by observing, processing and reacquiring the satellite, though this is limiting. Multiple sites allow for numerous observations to be taken and the opportunity to skip observation passes while the data is being processed. Update intervals are usually shorter than the maximum requirement. Ideally they occur when the most stringent requirement for the satellite is no longer met. Update intervals are related to the force model, since accurate force models enables longer update intervals.

Tracking arcs, or fit spans, are the period of time over which the data must be collected for differential correction [35]. The fit span often includes many update intervals and can be adjusted to meet accuracy requirements. Incomplete theories (i.e. those neglecting or

truncating perturbation terms) require longer fit spans to sample enough data to average the inaccurate force models, though longer fit spans potentially introduce error in the propagation routines. Shorter fit spans can cause a bias in the estimate of the initial state because the data may be insufficient to differentiate periodic variations from secular effects. For a deep-space craft a satellite pass may only be 8-10 hours; the update interval can be several hours or days; and the tracking arc is several days long (longer than the update intervals) [35]. A last general rule of OD is that *longer fit spans enable better prediction accuracy, but require a better initial guess and more iteration*. As is true for most estimation processes, a closer first guess provided for the OD filter is best. Although the filter can handle an inaccurate initial estimate, the filter can diverge if the initial state is sufficiently incorrect. A reasonable initial guess is required so that the filter does not diverge.

#### **4.3.2. Linearization of the Non-linear Problem**

The orbit determination problem requires determining the trajectory of the spacecraft in the presence of several complicating factors: (1) only indirect observations of the spacecraft state elements are available, (2) nonlinear equations describe the relationships between observations and state, (3) fewer observations exist at any time epoch than there are state vector components, and (4) errors exist in the observations [47]. Because linear estimation techniques are well-understood and relatively straightforward in this work the nonlinear orbit determination problem is linearized as described below and in [47].

Equations 4.1 and 4.2 describe the dynamics of the  $n$ -dimensional column vector of the state,  $\mathbf{X}(t)$ , at some time,  $t$ , and a  $p$ -dimensional column vector of observations,  $Y_i$  made at discrete times  $t_i$ :

$$\dot{X}(t) = F(X, t); \quad \bar{X}(t_k) \equiv \bar{X}_k \quad (4.1)$$

$$Y_i = G(X_i, t_i) + \varepsilon_i; \quad i = 1, 2, \dots, j \quad (4.2)$$

Where,  $\dot{X}(t)$ , is a continuous nonlinear 1<sup>st</sup> order ordinary differential equation, or simply the change in  $n$ -dimensional state  $\mathbf{X}(t)$ . The actual observation,  $Y_i$ , is assumed to be a nonlinear function of the true observation,  $G(X_i, t_i)$  and the random measurement noise,  $\varepsilon_i$ . Noises are present in each observation because non-linear functions of the state are influenced by random error which occurs in the process of making the observations [34]. Equation 4.1 can be solved via numerical integration. State and observation deviations errors are defined in Equation 4.3 as the difference between the true and the reference (\*) trajectory ( $\mathbf{X}(t)$ ), or observation ( $Y(t)$ ):

$$\begin{aligned} x(t) &= X(t) - X^*(t); & y(t) &= Y(t) - Y^*(t) \\ \dot{x}(t) &= \dot{X}(t) - \dot{X}^*(t) \end{aligned} \quad (4.3)$$

where,  $x = nx1$  state deviation vector  
 $y = px1$  observation deviation vector  
 $X(t), Y(t)$  = true trajectory, true observation  
 $X^*(t), Y^*(t)$  = reference trajectory, reference observation

The linearization of the OD process requires that the reference trajectory,  $X^*$ , be reasonably close to the true trajectory,  $X$ , throughout the time period of interest [47]. To linearize the OD equations, a Taylor Series expansion was performed, and the higher order (nonlinear) terms were neglected, since these terms were small compared to the

first order terms. Expanding  $\dot{X}(t)$  and  $Y_i$  in a Taylor Series about the reference trajectory,  $X^*$ , yields:

$$\begin{aligned}\dot{X}(t) &= F(X, t) = F(X^*, t) + \left[ \frac{\partial F(t)}{\partial X(t)} \right]^* [X(t) - X^*(t)] + H.O.T. \\ Y_i &= G(X_i, t_i) + \varepsilon_i = G(X_i^*, t_i) + \left[ \frac{\partial G(t)}{\partial X(t)} \right]^*_i [X(t_i) - X^*(t_i)]_i + H.O.T. + \varepsilon_i\end{aligned}\tag{4.4}$$

Neglecting the higher-order terms gives:

$$\begin{aligned}\dot{X}^* &= F(X^*, t) \\ Y_i^* &= G(X_i^*, t_i)\end{aligned}\tag{4.5}$$

The linear estimation problem is now:

$$\begin{aligned}\dot{x}(t) &= A(t)x(t) \\ y_i &= H_i x_i + \varepsilon_i\end{aligned}\quad i = 1, 2, \dots, j\tag{4.6}$$

where,

$$\begin{aligned}A(t) &= \left[ \frac{\partial F(t)}{\partial X(t)} \right]^*; \\ H_i &= \tilde{H}_i \Phi(t_i, t_k) \quad \text{where } \tilde{H}_i = \left[ \frac{\partial G}{\partial X} \right]^*_i; \\ x(t) &= X(t) - X^*(t); \quad \text{"* " = evaluated on the reference trajectory} \\ x_i &= X(t_i) - X^*(t_i); \\ y_i &= Y_i - G(X_i^*, t_i)\end{aligned}$$

The observation sensitivity matrix,  $H$  is the product of the observation-state mapping matrix  $\tilde{H}_i$ , which contains the partials of each observation with respect to the reference state component, and the state transition matrix  $\Phi(t, t_k)$ , which will be described below. For the OD problem,  $F(X, t)$  and  $G(X_i, t_i)$  will always be nonlinear in  $X(t)$ . Therefore, deviation vectors and reference trajectories are needed to linearize the system.

### 4.3.3. The State Transition Matrix

While the state deviation and observation error history can be determined by directly solving Equations 4.6, the state transition matrix (STM) provides a simplified formulation to determine the best estimate of  $x$  and for mapping the associated covariance matrices [47]. The STM is essential in relating observations at different times and expresses the solution of  $x(t)$  in terms of the unknown initial state,  $x_k$  [47]. A general solution of  $\dot{x}(t) = A(t)x(t)$  with time-dependent coefficients is:

$$x(t) = \Phi(t, t_k)x_k \quad (4.7)$$

where,  $\Phi(t, t_k)$  is the state transition matrix (STM). The STM reduces and maps observations to a single epoch and has the following properties [34]:

1. Reduces to an  $n \times n$  identity matrix:  $\Phi(t_k, t_k) = I$
2. Unique for Newtonian motion:  $\Phi(t_i, t_k) = \Phi(t_i, t_j) \Phi(t_j, t_k)$
3. Invertible for Newtonian mechanics:  $\Phi(t_i, t_k) = \Phi^{-1}(t_k, t_i)$
4. Satisfies the differential equation:  $\dot{\Phi}(t, t_k) = A(t)\Phi(t, t_k)$ ,  $\Phi(t_k, t_k) = I$  while its inverse satisfies:  $\dot{\Phi}^{-1}(t, t_k) = -\Phi^{-1}(t, t_k)A(t)$ ,  $\Phi^{-1}(t_k, t_k) = I$  the identity matrix,  $I$ .

The STM can be propagated from substituting values for  $\dot{x}(t)$  and then  $x(t)$  into this equation:  $\dot{x}(t) = \dot{\Phi}(t, t_k)x_k$  to yield:

$$\dot{\Phi}(t, t_k) = A(t)\Phi(t, t_k) \quad (4.8)$$

The solution of  $\Phi(t, t_o)$  is obtained by numerical integration Equation 4.8.

#### 4.3.4. Relating Observations to an Epoch State

From Equations 4.6 there is an unknown state vector,  $x_i$ , for each observation  $y_i$ . The state transition matrix is used to express a data set, or collection of observations in terms of the state at a single epoch as follows:

$$\begin{aligned} y_1 &= \tilde{H}_1 \Phi(t, t_k) x_k + \varepsilon_i \\ &\cdot \\ &\cdot \\ &\cdot \\ y_f &= \tilde{H}_f \Phi(t_f, t_k) x_k + \varepsilon_f \end{aligned} \tag{4.9}$$

Equation 4.9 now contains  $m = p \times f$  observations and only  $n$  unknown state components.

This series of equations can be rewritten in the matrix form:

$$y \equiv \begin{bmatrix} y_1 \\ \cdot \\ \cdot \\ \cdot \\ y_f \end{bmatrix}; \quad H \equiv \begin{bmatrix} \tilde{H}_1 \Phi(t_1, t_k) \\ \cdot \\ \cdot \\ \cdot \\ \tilde{H}_f \Phi(t_f, t_k) \end{bmatrix}; \quad \varepsilon \equiv \begin{bmatrix} \varepsilon_1 \\ \cdot \\ \cdot \\ \cdot \\ \varepsilon_f \end{bmatrix} \tag{4.10}$$

or more compactly:

$$\begin{bmatrix} \mathbf{y} \\ \text{mxf} \end{bmatrix} = \begin{bmatrix} \mathbf{H} \\ \text{mxfn} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \text{nxk} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\varepsilon} \\ \text{mxf} \end{bmatrix} \tag{4.11}$$

The H matrix is the observation sensitivity matrix, which is the product of the state transition matrix and observation-state mapping matrix,  $\tilde{H}$ , at a given time. If  $p$  or  $f$  is sufficiently large, then the essential condition  $m > n$  is satisfied. However, the problem lies in the fact that there are still  $m$  unknown observation errors, resulting in  $m+n$  total

unknowns in only  $m$ -equations. The least squares criterion allows a solution to the  $n$ -state variables,  $x$ , at the epoch  $t_k$  by providing conditions on the  $m$ -observation errors.

#### **4.4. Orbit Determination Methods**

Orbit determination methods can be used to predict, filter, smooth, and pre-process the observations. Prediction uses existing observations to compute future states via propagation, while filtering is determining the current state using current (and past) observations. Filtering is typically performed by algorithms such as Kalman Filtering and sequential-batch methods that update state and covariance matrices [35]. Smoothing applies to the least-squares and traditional sequential-batch least squares techniques and improves estimates based on existing data to determine the state at a particular epoch. Pre-processing is applying the smoothing process to the observations [35].

Forces acting on each LISA satellite cannot be presumed fully known, but deterministic equations of motion and state transition matrix can be propagated to approximately predict future system states. Measurements are then corrupted by stochastic noise.

##### **4.4.1. Batch Least Squares**

The least squares criterion as derived by Gauss is applied to approximate the solution of an over-determined ( $m > n$ ) set of linear algebraic equations. The least squares solution selects a state estimate  $x$  that minimizes the sum of the squares of the calculated residuals.



As shown in Equation 4.12,  $x$  is chosen to minimize the performance index, or cost function  $J(x)$ . Note  $\varepsilon$  can be defined from previous Equation 4.11 as  $\varepsilon = y - Hx$ :

$$J(x) \equiv \frac{1}{2} \varepsilon^T \varepsilon = \sum_{i=1}^f \frac{1}{2} \varepsilon_i^T \varepsilon_i = \frac{1}{2} (y - Hx)^T (y - Hx) \quad (4.12)$$

The cost function  $J(x)$  is a squared function because a parabola ( $\varepsilon^2$ ) has a minimum, whereas a line ( $\varepsilon$ ) does not [35].  $J$  is also a quadratic function of  $x$ , it has a unique minimum when these necessary conditions apply:

$$1.) \quad \left( \frac{\partial J}{\partial x} \right) = 0; \quad \text{and} \quad 2.) \quad \delta x^T \frac{\partial}{\partial x} \left( \frac{\partial J}{\partial x} \right) \delta x > 0, \text{ for all } \delta x \neq 0$$

The first condition gives the global minimum/maximum value. The second condition is a second derivative or the Hessian of  $J(x)$ . The Hessian, which is greater than 0, implies that the symmetric matrix is positive definite indicating that  $x$  occurs at a minimum. The first condition can explicitly obtain the normal equation:

$$\left( \frac{\partial J}{\partial x} \right) = 0 = -(y - Hx)^T H = -H^T (y - Hx) \quad (4.13)$$

this, can be rearranged to obtain:

$$H^T y = (H^T H) \hat{x} \quad (4.14)$$

where  $\hat{x}$  is the least squares correction that yields the best estimate of  $x$ . Equation 4.14 is called the normal equation because if  $H^T H$  is an  $m \times m$  symmetric matrix, and if the Hessian is positive definite, then  $H$  has full rank. Therefore, the solution for the best estimate of  $x$  given the linear observation state relationship Equation 4.11 is Equation 4.15:

$$\hat{x} = (H^T H)^{-1} H^T y \quad (4.15)$$

For the solution in Equation 4.15 to exist,  $m \geq n$  and  $H$  must have rank  $n$ . If  $m < n$ ,  $H$  is not of full rank  $n$ , thus there are more unknowns than linearly independent observations.

Therefore a minimum norm solution must be used to determine  $\hat{x}$ , while this is not typically seen in OD problems, it is included for completeness.

The three cases are summarized as follows:

- 1.)  $m = n$  (the dimension of  $y$  is the same as the dimension of  $x$ )
  - $H$  is square ( $m \times m$ ) and invertible
  - $\hat{x} = H^{-1}y$ , trivial solution of  $x$
- 2.)  $m > n$  (the dimension of  $y$  is greater than  $x$ )
  - $H$  is symmetric and positive definite,  $H$  has rank =  $n$
  - $\hat{x} = (H^T H)^{-1} H^T y$ , Least Squares solution for  $\hat{x}$
- 3.)  $m < n$  (more unknowns than equations, but is not typically seen in OD)
  - $H$  is not square or invertible,  $H$  has rank  $< n$
  - $\hat{x} = H^T (H H^T)^{-1} y$ , minimum norm solution

Note that the product  $H^T (H H^T)^{-1}$  and  $(H^T H)^{-1} H^T$  are the pseudo inverses of the  $H$  matrix.

There are disadvantages to the least squares solution that lead to other methods to solve the OD problem. First, each observation error is weighted equally even though the accuracy of the observations may disagree, an issue addressed with the weighted least squares solution described below. The observation errors may be correlated (i.e. not independent), but the least squares solution does not allow for this. Also, the least squares (LS) method does not consider errors as samples from random processes, i.e. there is no use of statistical information. Due to computational problems that arise from inverting the normal equation matrix  $(H^T H)$ , a variety of solution methods exist for the OD problem. The Cholesky decomposition is a more efficient and sometimes more accurate method to take the inverse, though it is applicable only if the normal equation

matrix is symmetric and positive definite [45]. Orthogonal transformation (or the QR factorization method) obtains solutions by applying successive orthogonal transformations to the information array; it yields the same accuracy, but with single-precision computational arithmetic [40]. Given's rotation, or Householder transformation are two orthonormal transformations that can be used to perform QR factorization [40]. The Singular value decomposition (SVD) method is an alternative to the QR decomposition used to solve the LS problem in a numerically stable manner. SVD is good for detecting and overcoming a possible singularity or near singularity, making it well-suited for ill-conditioned problems. However, the computational effort for the SVD method is more substantial than that for the QR factorization [40].

#### 4.4.2. Weighted Least Squares

The weighted least squares (WLS) method is similar to the least squares except there is an associated diagonal weighting matrix,  $w_i$ , for each of the observation vectors. The elements of the weighting matrix are normalized with weighting factor values ranging from 0 -1.0. The smaller the weight the less important the measurement (i.e.  $w_i = 0$ , neglects the observation); while the larger the weighting factor, the more important it is (i.e.  $w_i = 1$ , gives “full” weight to that observation).

The performance index for the weighted least squares problem is given by:

$$J(x_k) \equiv \frac{1}{2} \varepsilon^T W \varepsilon = \sum_{i=1}^f \frac{1}{2} \varepsilon_i^T w_i \varepsilon_i = \frac{1}{2} (y - Hx_k)^T W (y - Hx_k) \quad (4.16)$$

Again, the necessary condition, Equation 4.17 for a minimum  $J(x_k)$  requires that the derivative of  $J$  with respect to  $x_k$  be zero:

$$\left( \frac{\partial J}{\partial x_k} \right) = 0 = -(y - Hx_k)^T WH = -H^T W(y - Hx_k) \quad (4.17)$$

Equation 4.17 is rearranged to obtain the normal equation:

$$H^T W y = (H^T W H) x_k \quad (4.18)$$

Again, if the normal matrix  $H^T W H$  is positive definite then the solution will have an inverse. The weighted least squares estimate,  $\hat{x}_k$ , is given by:

$$\hat{x}_k = (H^T W H)^{-1} H^T W y \quad (4.19)$$

Equation 4.19 can also be expressed as:

$$\hat{x}_k = (P_{cov})_k H^T W y \quad (4.20)$$

where  $(P_{cov})_k = (H^T W H)^{-1}$ . The covariance matrix,  $(P_{cov})_k$ , is a symmetric  $n \times n$  matrix that must be positive definite to have an inverse. Parameter observability is related to the rank of the  $(P_{cov})_k$  matrix. If all parameters in  $x_k$  are observable, then  $(P_{cov})_k$  has full rank and an inverse. This coincides with the condition that there is a greater or equal number of independent observations to the number of parameters being estimated ( $m \geq n$ ). In general, the larger the magnitude of  $(P_{cov})_k$ , the less accurate the estimate, but caution should be exercised when inferring the accuracy of the estimate by the magnitude of  $(P_{cov})_k$  in the WLS estimate because with normalized weights,  $(P_{cov})_k$  is not valid in the statistical sense [45].

If an a priori value  $\bar{x}_k$  is available for  $x_k$  and a corresponding weighting matrix  $W_k$  is available, then the following expression can be used for the estimate of  $x_k$ :

$$\hat{x}_k = \left( H^T W H + \overline{W}_k \right)^{-1} \left( H^T W y + \overline{W}_k \overline{x}_k \right) \quad (4.21)$$

where,  $\overline{x}_k$  is the a priori estimate and  $\overline{W}_k$  is the weighting matrix for the a priori estimate of  $\overline{x}_k$ .

#### 4.5. Statistical Methods

The statistical methods are mentioned here for completeness. As seen in the historical overview all of these methods are applications and extensions of Gauss's least squares method.

##### 4.5.1. Linear Minimum Variance Estimation (LUMV)

While the least squares and weighted least squares estimates provide no information on statistical characteristics of measurement errors and no information on a priori errors, the linear minimum variance estimator (LUMV) does address these two issues. Advantages of using the LUMV estimator are that it is relatively simple to use, since a complete statistical description of all the random errors is not necessary and only the first and second moments of the probability density function (pdf) of the observation errors are required (i.e. the mean and covariance matrix associated with random error respectively) [34]. The observation error,  $\varepsilon_i$ , is assumed to be random with zero mean and a specified covariance,  $(R_{cov})_i$ :

$$E[\varepsilon_i] = 0, \quad E[\varepsilon_i \varepsilon_i^T] = (R_{cov})_i \quad (4.22)$$

The state estimation problem is formulated as:

$$x_i = \Phi(t_i, t_k)x_k \quad (4.23)$$

$$y_i = \tilde{H}_i x_i + \varepsilon_i \quad i = 1, \dots, l$$

where:  $E[\varepsilon_i] = 0$ ,  $E[\varepsilon_i \varepsilon_j^T] = (R_{cov})_i \delta_{ij}$  and  $\delta_{ij}$  = Kronecker delta function. The best linear, unbiased, minimum variance estimate,  $\hat{x}_k$ , of the state  $x_k$  is shown below. The linear condition implies that the estimate is made up of a linear combination of the observations:

$$\hat{x}_k = By \quad (4.24)$$

where,  $B$  is an unspecified linear mapping ( $n \times m$ ) matrix selected to obtain the best estimate [34]. The unbiased condition follows:

$$E[\hat{x}_k] = x = E[By] = E[BHx_k + B\varepsilon_k] = x_k$$

Recall that the expected value, or mean of the error is zero:  $E[\varepsilon_k] = 0$ , so that:

$BHx_k = x_k$ , to arrive at Eqn 4.25:

$$BH = I \quad (4.25)$$

Lastly, the minimum variance condition is the task of selecting the ( $n \times m$ ) matrix  $B$  to minimize the covariance matrix. The covariance matrix,  $(P_{cov})_k$  for an unbiased estimate is:

$$(P_{cov})_k = E[(\hat{x}_k - E[\hat{x}_k])(\hat{x}_k - E[\hat{x}_k])^T]$$

With substitutions of the linear and unbiased conditions and unspecified Lagrange multipliers,  $L$ , the covariance matrix,  $(P_{cov})_k$  is given by:  $(P_{cov})_k = (H^T R^{-1} H)^{-1}$ . And the LUMV estimate of  $x_k$  is Equation 4.26:

$$\hat{x}_k = (H^T R^{-1} H)^{-1} H^T R^{-1} y \quad (4.26)$$

Note that if the weighting matrix,  $W$ , is equal to the inverse of the observation noise covariance matrix,  $R$ , i.e.  $W = R^{-1}$ , the LUMV solution agrees with the WLS solution of Equation 4.21. Similarly to WLS, the LUMV estimate can be modified to propagate the estimate of the state  $x_k$  and its associate covariance matrix. The LUMV estimate with a priori information can be determined from these propagated values [34]. The LUMV estimate of Equation 4.26 also will agree with the maximum likelihood estimate (MLE) developed by Fisher (1912) if the observation errors are assumed to be normally distributed with a zero mean and covariance,  $(R_{cov})_k$  [47].

#### 4.5.2. Sequential Methods

Gauss implied that observations can be inaccurate and of unknown and unknowable errors [2], leading to modern statistical methods such as the Kalman filter. Sequential estimators process observations as soon as they are received. The solution and its associated covariance then can be mapped to another time [47].

A major advantage of this method is that it avoids some of the matrix inversion problems of the least squares methods, in that the matrix to be inverted will be the same dimension as the observation vector [45]. The estimate of the state  $x_k$ , is simply Equation 4.27, which is obtained by recursively solving the supporting equations

$$\hat{x}_k = \bar{x}_k + K_k [y_k - H_k \bar{x}_k] \quad (4.27)$$

where,  $\bar{x}_k = \Phi(t_k, t_j) \hat{x}_j$

$$K_k = \text{weighting matrix (aka Kalman gain matrix)} = \bar{P}_k H_k^T [H_k \bar{P}_k H_k^T + R_k]^{-1}$$

$$\bar{P}_k = \text{a priori covariance matrix} = \Phi(t_k, t_j) P_j \Phi^T(t_k, t_j)$$

$$P_k = \text{a posteriori covariance matrix} = (H_k^T R_k^{-1} H_k + \bar{P}_k^{-1})^{-1}$$

A disadvantage of the sequential processor is that the state estimate covariance matrix will approach zero ( $P_k \rightarrow 0$ ) as the number of observations becomes large. The gain  $K_k$  also approaches zero, and the estimation procedure becomes insensitive to observations and will diverge due to errors that have now been introduced into the linearization procedure, computational errors, and errors due to an incomplete math model [45]. To improve on this disadvantage, two modifications can be used: 1) the Extended Sequential estimator (aka Extended Kalman-Bucy Filter), which minimizes the effects of errors due to the neglect of higher order terms in the linearization process; and 2) the State noise compensation method to combat errors in the dynamic model. Additional modifications such as the 1) Q-Matrix compensated sequential estimation algorithm; and 2) The Dynamic Model Compensation (DMC) estimation algorithm, a sequential estimation method that compensates for the unordered effects in the differential equations which describe the motion of an orbiting vehicle [34].

### **4.5.3. Comparison of OD Methods**

The batch least squares method is often compared to the sequential methods to justify their use, as these are the two basic data processing procedures for tracking data used to estimate a spacecraft's orbit. A disadvantage of both the batch and sequential processors is that if the true and reference state are not close together, the linearization assumptions leading to Equation 4.6 are not valid and the estimation process will diverge [45]. The



batch and sequential algorithms are both iterated until convergence, while the Extended Kalman Filter (EKF) will obtain near convergence in a single iteration [45].

Batch processing allows for simultaneous use of multiple measurements, hence making it most useful for post-flight analysis of very large observation datasets. It provides an estimate of the state at some chosen epoch using an entire batch or set of data. The estimate and its associated covariance are mapped to other times. Process noise in the batch processor complicates the solution of the normal equations, the dimension of the normal matrix increases from  $n$  (#of state parameters) to  $m$  (# of observations) [47]. Without process noise, the sequential algorithm can be shown to be mathematically equivalent to the batch process. Given the same data set, both will produce the same estimates when mapped to the same times [47]. Process noise is added in the EKF however, and this algorithm is not equivalent to the batch processor, but has been shown to be very close [47]. The numerical integrator in the EKF is restarted at each observation time (good for real-time applications), and some state noise is needed to ensure that convergence does not occur. State noise compensates for various error sources in the processing of ground-based or onboard data [47]. As noted before, the covariance matrix,  $(P_{cov})_k$ , may approach zero as the number of observations becomes large, causing filter divergence, though process noise is usually added to deal with this issue. Sequential Filters are best suited for real-time estimation when the mathematical model used to describe the system is accurate, though corrections for unordered force effects can also best be implemented with sequential estimation algorithms [34].

#### 4.6. Perturbation Effects

Major sources of error can be divided into two main groups: satellite force model effects and measurement model inaccuracies. Error sources are mainly *satellite-dependent*, meaning that the satellite configuration, orbit altitude, inclination, and measurement systems used for each individual mission plays a major role in the effects of errors.

Errors affecting the satellite force model are further subdivided into gravitational and nongravitational parameters. Gravitational parameters include: mass of the Earth ( $GM$ ), Geopotential coefficients ( $C_{lm}$ ,  $S_{lm}$ ), solid Earth and ocean tide perturbations, mass and position of the moon and planets, and general relativistic effects. Observations of range/range rate and angular data from various satellites work to improve Earth's gravitational model [47] so that the gravitational potential of Earth, a major error source for LEO satellites, can be better compensated. With deep space orbits like that used for LISA, each satellite is mainly affected by solar-radiation pressure and third-body effects. Gravitational forces from large planets (i.e. Jupiter, Saturn) and asteroids may pose a greater error source on LISA than the gravity of Earth. Non-gravitational parameters include drag, solar and Earth radiation pressure, thrust and other effects such as magnetic forces. This research presumes a two-body force model in which each satellite is only influenced by the Sun's gravitational effects. Appendix A summarizes basic calculations that verify third body effects may be presumed negligible for this preliminary OD work.

Errors in the measurement model include: inertial and terrestrial coordinate system errors and ground-based measurements errors. Inertial and terrestrial coordinate system errors are precession, nutation, and polar motion. Ground-based measurement errors include

tracking station coordinates, atmospheric effects, instrument modeling, tectonic plate motion and clock accuracy [47]. Other appreciable error sources include inaccuracies in the mathematical model of the dynamical (satellite force) and instrument measurement system. Computer errors are due to storage and execution time problems or errors in data processing and the fact that various parameters, such as  $\mu$ , in the math model are only approximately known are all additional sources of error.

## CHAPTER 5: Problem Description

This chapter describes the LISA OD simulation that was developed for this thesis.

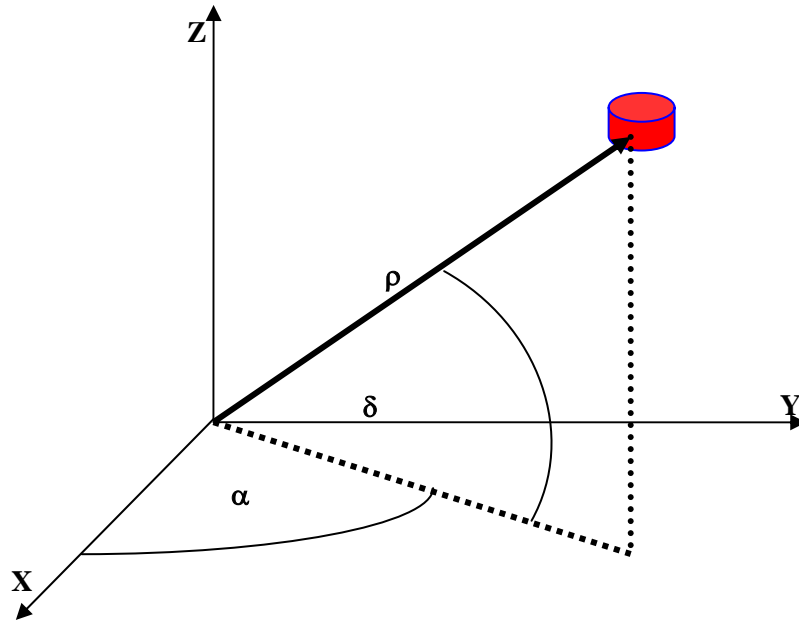
Following a summary of assumptions, the batch processor algorithm will be presented, including the four data type cases tested, the initial condition of the LISA orbit, and data noise sources. Methods used to identify errors in the algorithm and software and a discussion of the equations required to obtain the absolute and relative OD errors are presented.

### 5.1. Assumptions in the Model

The major modeling assumptions in the simulation are:

- 1) The values for parameters  $\mu_{SUN}$  and any other constants are considered to be known exactly.
- 2) Earth and the LISA spacecraft are all considered to be point masses.
- 3) The tracking station(s) are located at the center of this point mass Earth. The rotation of the Earth and DSN tracking stations are not taken into account. A specific Deep Space Station antenna is not modeled.
- 3a.) DSN range and range rate measurements will be two-way tracking made from the single tracking station on the point mass Earth

3b.) VLBI measurements will also be made from that single tracking station, despite the practical requirement for two stations (Section 3.4.2). Ultimately, the angle measurements are the output as illustrated in Figure 5.1. To avoid confusion VLBI refers to the angular measurements right ascension,  $\alpha$ , and declination,  $\delta$ .



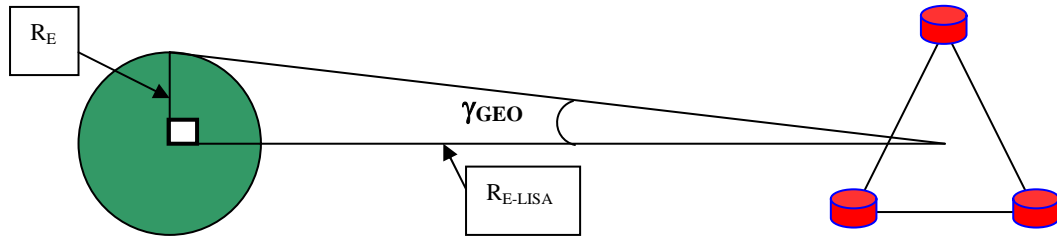
**Figure 5.1:** Angle measurements to spacecraft

4) The system is unperturbed and will be treated as a two-body system with the Sun being the central body force and each spacecraft in the LISA constellation the negligible mass (see Appendix A for a summary of error resulting from this assumption).

4a.) The placement of Earth in the model is considered known and is 20 degrees ahead of the reference center of the LISA constellation. The Earth is also a negligible mass in the two-body system.

- 4b.) The reference orbit is deterministic, a system whose time evolution can be predicted exactly. Therefore, Newtonian acceleration will be integrated and the trajectory of the spacecraft and Earth will be propagated with the Equations of Motion (Section 3.1) for one year from some arbitrary epoch  $t = 0$ .
- 5.) Tracking will be based on X-band uplink/downlink Doppler and range to data to the spacecraft.
  - 5a.) Noise and bias for DSN and VLBI will be based on values for the year 2000 X-band system for DSN using typical error values as listed in Cathy Thornton's monograph [41].
  - 5b.) Values for inter-spacecraft range and range rate random error and biases were made to match equivalent Earth-based range and range rate values.
- 6) The random noises added to the actual measurements will be assumed to follow a Gaussian/Normal distribution.
- 7) A three month (or ~90 day) commissioning period at the end of cruise phase, when LISA is close to its nominal orbit, is the key timespan that this thesis will investigate. During this crucial phase, P/M drop-off has occurred and OD tracking continues until drag-free control and the science of the LISA mission can officially begin.

Calculation of the geocentric parallax,  $\gamma_{\text{GEO}}$ , roughly justifies assumption #3, or the angular difference between observations from the center of Earth and those from a site [35]. A simple, not to scale, drawing as seen in Figure 5.2 illustrates geocentric parallax.



**Figure 5.2:** Geocentric parallax

The geocentric parallax,  $\gamma_{\text{GEO}}$ , is:

$$\begin{aligned} \gamma_{\text{GEO}} &= \arctangent(R_E / R_{\text{E-LISA}}) = \\ &= \arctangent(6,378,135 \text{ meters} / 5.1955 \times 10^{10} \text{ meters}) = \\ &= 1.2276 \times 10^{-4} \text{ radians, or } \sim 0.007 \text{ degrees} \end{aligned}$$

The radius of Earth is 6,378,135 meters, negligible compared to the distance from Earth to the reference center of LISA which is  $5.1955 \times 10^{10}$  m. Hence, the approximation of the tracking station being located at Earth's center is reasonable, since the geocentric parallax is less than a tenth of a degree (= 0.007 degrees).

For assumption # 4, recall that the initial set-up of LISA placed  $20^\circ$  behind the Earth was a compromise to be sufficiently far to reduce the gravitational pull of the Earth and Moon on this constellation, yet still be “close” enough to meet radio communication requirements with Earth, and to meet launch vehicle propellant requirements.

While drag free control shields the proof mass from perturbations such as solar flux, the gravity of other planets still does affect the spacecraft. However, back of the envelope calculations of the force effects of the two major perturbation sources, Jupiter at closest approach to Sun, and Earth at closest approach to LISA. Appendix A uses Newton's Law

of Gravitation to compare these effects to the force of the Sun on LISA to justify the assumption of a two-body system. It was found that the Sun poses the most force on LISA in both cases, which supports the fact that this can be considered a two-body problem for now. Certainly a full perturbation analysis is required prior to launch.

## **5.2. Batch Processor Algorithm**

The computational algorithm for the batch processor used in this thesis is based on the Batch Processing algorithm from Tapley's text *Statistical Orbit Determination* (2004) [47]. The MATLAB 7.0 code that implements the entire simulation and batch processing algorithm is presented in Appendix B. The LISA OD problem requires modeling three spacecraft and four main test cases described below.

### **5.2.1. Test Cases**

Four test cases that utilize different combinations of observation data were investigated:

- Case 1:** DSN & VLBI (D/V)
- Case 2:** DSN, VLBI, & Relative (D/V/R)
- Case 3:** DSN only (DSN)
- Case 4:** DSN & Relative (D/R)

Note that cases such as Angles-only, Relative-only and Angles/Relative observation cases were also considered, however, they were eliminated due to the fact that system state is not fully observable with these measurement sets. Case 3 explores accuracy with strictly the DSN data package, which includes range and range rate measurements. Case



1 uses a standard DSN and VLBI (angles) package. Case 2 incorporates the more novel relative data (i.e. inter-spacecraft range and range rate) available from the LISA spacecraft into DSN and VLBI and is expected perform the best according to the hypothesis that more data types would equal better OD. Case 4 combines DSN and relative data measurements.

### **5.2.2. The Nominal Orbit**

Once a nominal orbit is determined for the baseline and extended missions, proper OD is needed to achieve and maintain the LISA constellation in the specified configuration. An approximate knowledge of the orbit is required for least squares analysis in order to supply the initial "guess" of the state to the algorithm. Section 4.3 stated that the true state of the spacecraft in the OD problem is never known. However, these results were simulated without actual data. It is assumed in this simulation that the truth orbit is known and is the nominal orbit; which will be stated below. The test of how well the weighted batch least squares (WBLS) algorithm estimates is based on how well the final estimate agrees with the truth. Noises and biases will be added into the measurements to simulate real world data obtained from instruments and tracking stations.

Nominal orbits for use on the actual LISA mission have been determined through the works of Steve Hughes [14] and S.V. Dhurandhar et al [18]. Hughes' nominal orbit is found to maintain a nearly equilateral triangle configuration for two-body orbital dynamics and to be unstable when all the planets and Earth's moon are added to the

system. It is also a good guess for numerical optimization as an appropriate orbit can be determined for LISA in a perturbed orbit.

In [14], Hughes creates five performance measures, or cost functions for his orbit optimization problem. The challenge is to "find solutions that minimize a cost function and also satisfy a set of non-linear constraints imposed by instruments and communication limitations." There are ten nonlinear solution constraints for LISA:

- 1) Leg lengths must be +/-2% of the nominal  $5 \times 10^9$  m  
(3 constraints for each of the 3 legs)
- 2) Interior angles of the formation should remain between  $58.5^\circ$ - $61.5^\circ$   
(3 constraints)
- 3) The formation must orbit no more that  $20^\circ$  behind the Earth to meet communication and OD constraints (1 constraint)
- 4) The maximum average rate-of-change must be  $<15\text{m/s}$  for any of the legs  
(3 constraints)

Considering the solution constraints, Hughes numerically optimizes the performance of the formation in the presence of perturbations for these five cost functions ( $cf_n$ , where  $n=1, 2, \dots, 5$ ) formulated based on science and/or practical engineering or OD concerns:

- $cf_1$ : Difference in rate-of-change between two legs in the formation (science).
- $cf_2$ : Difference in lengths of two legs of the formation (science).
- $cf_3$ : Average rate-of-change of all three legs of the formation (engineering).
- $cf_4$ : Maintaining equal leg lengths over time (science).

cf<sub>5</sub>: Deviation of interior angle(s) of the formation from 60° as an alternative method to ensure that all three leg lengths are equal (science).

Hughes improves the nominal orbit, by using it as an initial guess through his optimization approach for each of the five cost functions and checks their feasibility by how well they satisfy the constraints. Since this thesis assumes an unperturbed orbit, Hughes' orbit will be used, though Dhurandhar's will be commented upon.

In the nominal orbit, four orbital elements ( $a, e, i, \omega$ ) are the same for all orbits in the formation. The longitude of the ascending node,  $\Omega$ , and mean anomaly,  $M$ , are different for each orbit. Below is a summary of the nominal orbital elements from [14]:

**Nominal orbit:**

$$a = a_E = 1 \text{ AU} = 149597870691 \text{ m}$$

$$e = \frac{L}{2\sqrt{3}a_E} = 9.6484 \times 10^{-3}, \text{ where } L = \text{leg length} = 5 \times 10^9 \text{ m}$$

$$i = \frac{L}{2a_E} = 1.6711 \times 10^{-2} \text{ radians} = 0.9574956^\circ$$

$$\omega = \pi/2 \text{ or } 3\pi/2 \text{ (} 90^\circ, \text{ or } 270^\circ)$$

for LISA 1: ( $\Omega, M$ )  
 for LISA 2: ( $\Omega + 2\pi/3, M - 2\pi/3$ ), where  $2\pi/3 = 120^\circ$   
 for LISA 3: ( $\Omega - 2\pi/3, M + 2\pi/3$ )

$$P = 1 \text{ year} = 2\pi \sqrt{\frac{r^3}{\mu}} = 3.1558 \times 10^7 \text{ sec.} \sim 365.26 \text{ days}$$

where  $\mu_{Sun} = 1.327124 \times 10^{20} \text{ m}^3/\text{s}^2$

The first four elements ( $a, e, i, \omega$ ) are the same for all spacecraft. The right ascension,  $\Omega$ , and mean anomaly,  $M$ , have a phasing of 120°. The periods,  $P$ , of each LISA are all one year (or  $3.1558 \times 10^7$  sec) - the same as the Earth's period. Table 5.1 below

summarizes the thesis initial condition for all three LISAs and Earth at an epoch unattached to any time system.

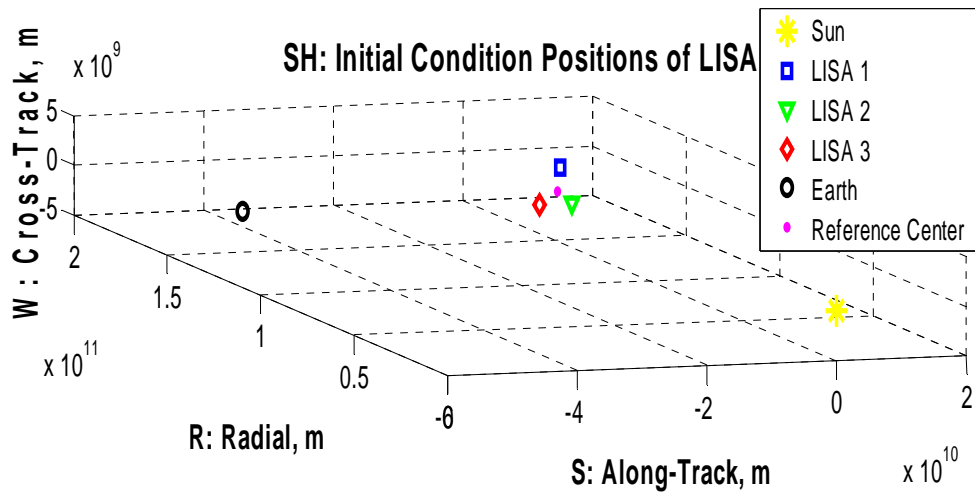
**Table 5.1:** Hughes' nominal orbital elements of LISA & Earth at epoch = 00:00:00

	$a$ (m)	$e$	$i$ (°)	$\Omega$ (°)	$\omega$ (°)	$M$ (°)
<b>LISA 1</b>	$1.4960 \times 10^{11}$	0.0096484	0.95750	0	90	0
<b>LISA 2</b>	$1.4960 \times 10^{11}$	0.0096484	0.95750	120	90	-120
<b>LISA 3</b>	$1.4960 \times 10^{11}$	0.0096484	0.95750	240	90	120
<b>Earth</b>	$1.4960 \times 10^{11}$	$1.3438 \times 10^{-16}$	0	NaN	NaN	0

The right ascension,  $\Omega$ , and the argument of periapsis,  $\omega$ , of Earth are unobservable, indicated as "NaN" (i.e. Not a Number) because the inclination is 0. The work of this thesis was done using Cartesian elements. The nominal orbit derived in Table 5.1 was transformed to a position and velocity using *Orb2X2.m* (see Appendix B). The Cartesian state can be transformed back to the Keplerian orbital elements as a check using *X2Orb.m* (see Appendix B). The position and velocity of the reference center of the LISA orbit was derived to be:

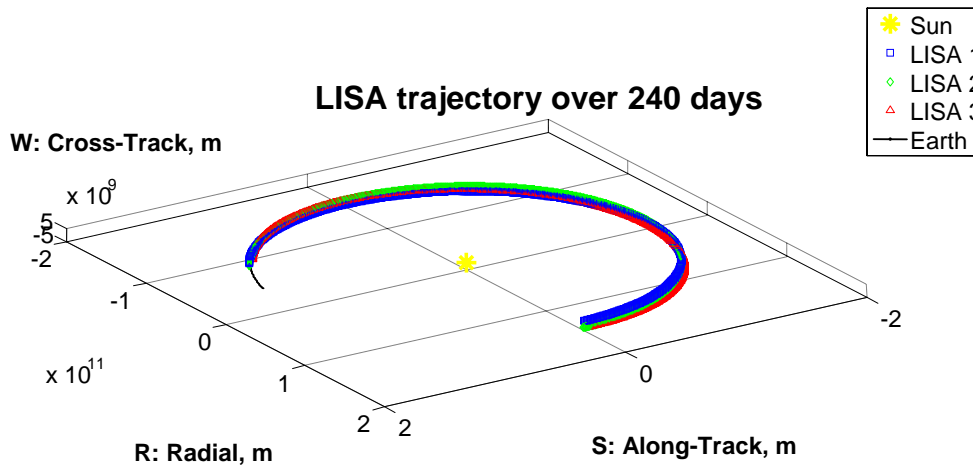
$$\begin{aligned} R_{VS} &= [0 \ a_E \ 0] \\ V_{VS} &= [-v \ 0 \ 0] \end{aligned}$$

Where  $a_E$  is the semi-major axis that is initially all in the + $Y$ -direction (Heliocentric); and  $v$  is the tangential velocity of the reference orbit and is instantaneously in the negative  $X$ -direction. The Earth was rotated to be  $20^\circ$  ahead of the reference center, as illustrated in an "equal-axis" view of Figure 5.3, the initial position for the nominal orbit of the three LISA spacecraft, Earth and the Sun (in *RSW* coordinates). The initial position and velocities of Earth and LISA are then used as the initial condition to propagate the orbit using the *LISA\_EOMderivs.m* (see Appendix B).



**Figure 5.3:** Initial position of LISA constellation (equal axis view)

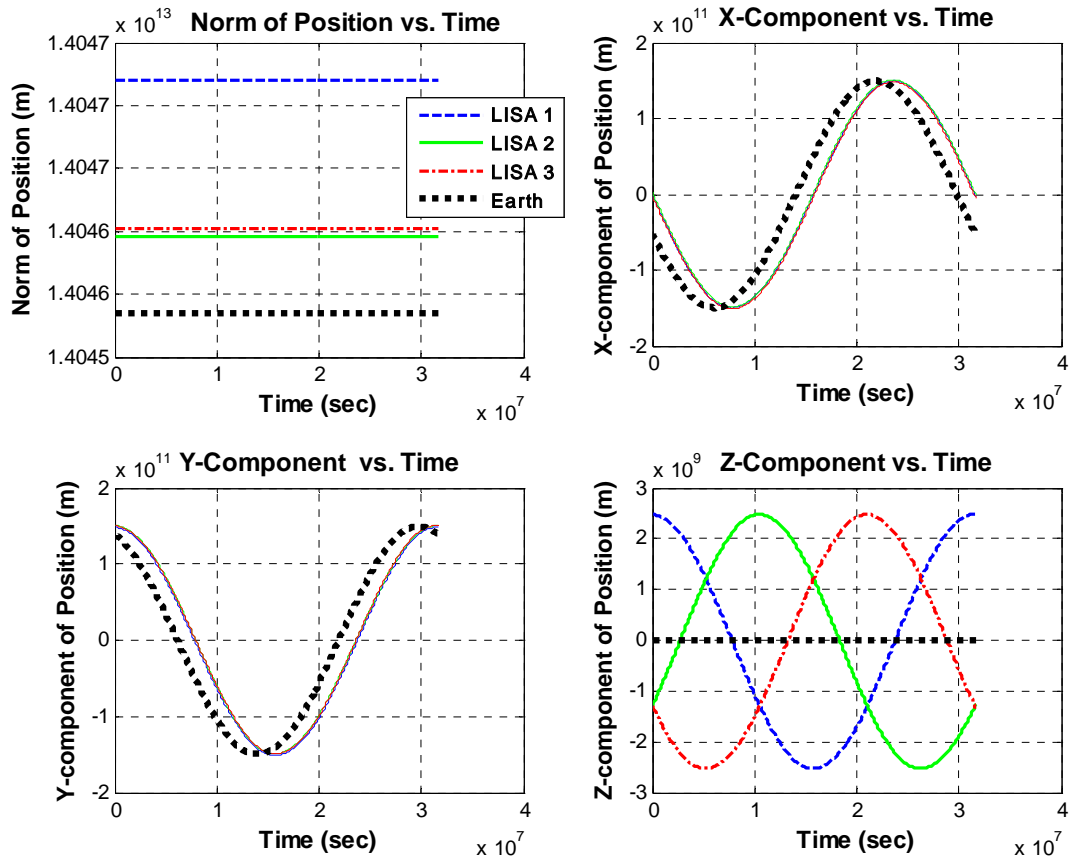
The motion of the constellation around the Sun is carried out for 240 days using the unperturbed equations of motion as illustrated in Figure 5.4.



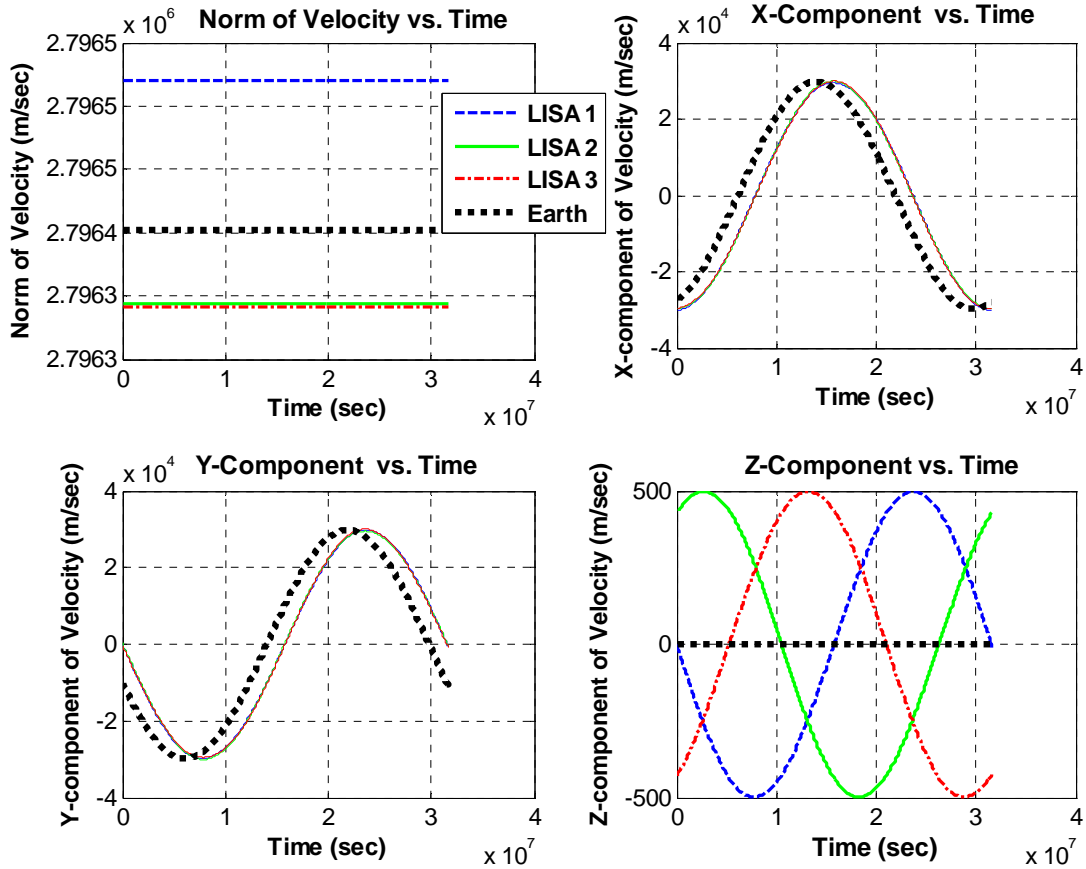
**Figure 5.4:** LISA constellation over 240 days

Note that each LISA rotates around its epicycle and is phased  $120^\circ$  from the other spacecraft as was indicated in the initial orbital elements of Table 5.1. This is clearly

shown in Figures 5.5 and 5.6 which illustrate the sinusoidal motion of the position and velocity components of each LISA.



**Figure 5.5:** LISA position components over 1 year (unperturbed)



**Figure 5.6:** LISA velocity components over 1 year (unperturbed)

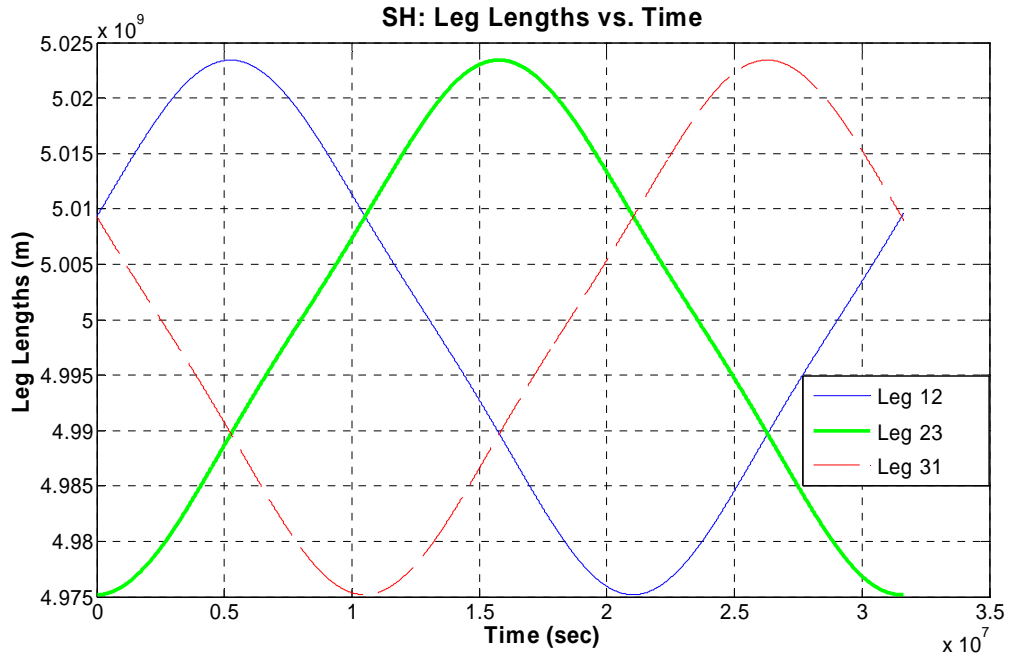
The leg length,  $L_{ij}$ , is a scalar defined as Equation 5.1:

$$L_{ij} = \left\| \bar{r}_i - \bar{r}_j \right\| \text{ where } i \neq j \text{ and } i \ \& \ j = 1, 2, 3 \quad (5.1)$$

The initial leg lengths for the nominal orbit are:

$$[L_{12} \ L_{23} \ L_{31}] = [5.0093 \times 10^9 \quad 4.9752 \times 10^9 \quad 5.0093 \times 10^9] \text{ meters}$$

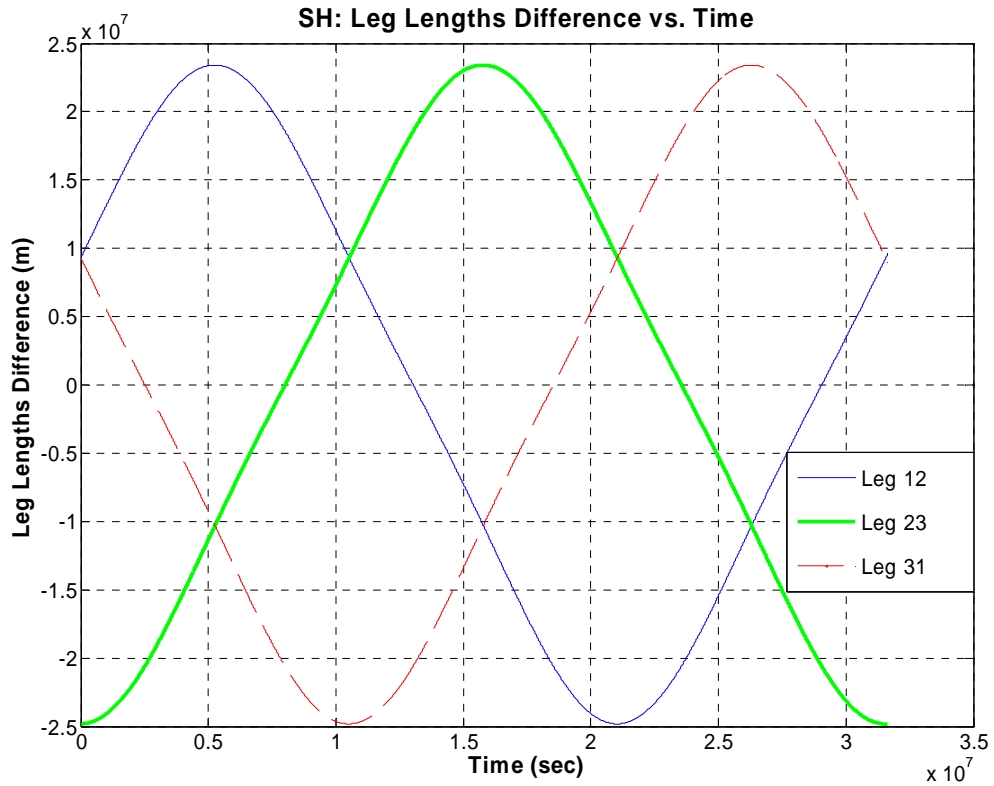
Figure 5.7 illustrates the sinusoidal leg length variation over an unperturbed orbit propagated for one year.



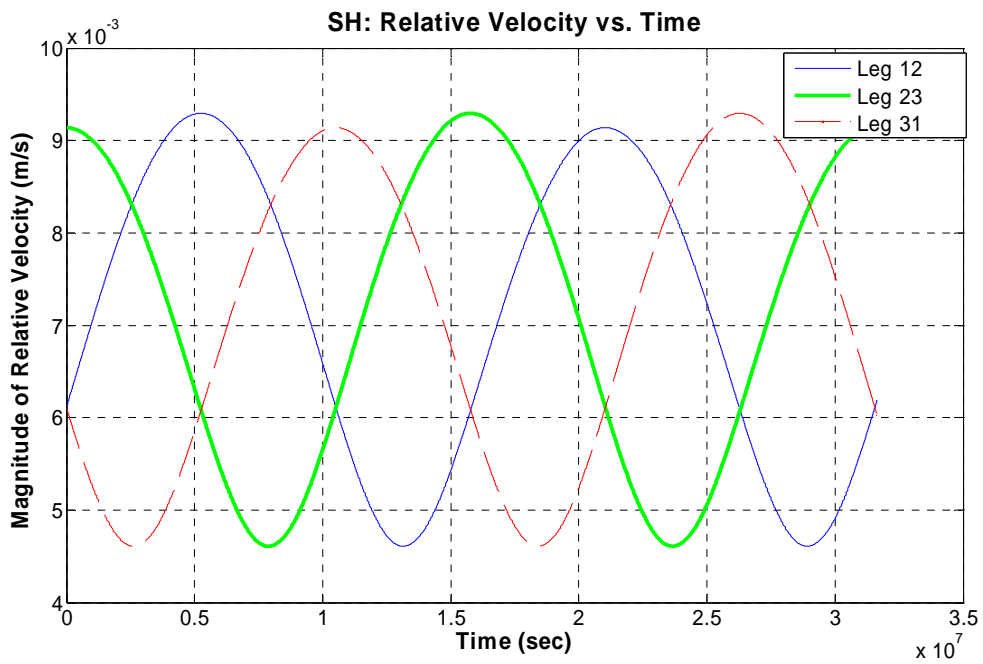
**Figure 5.7:** Leg length vs. time for an unperturbed orbit over 1 year

Figure 5.8 is the differenced leg length graph in which the nominal leg length of exactly  $5 \times 10^9$  m is subtracted from each leg length. The result is a graph that shows the percent error. The leg length varies by  $\pm 0.5\%$ , or  $\pm 25,000$  km. The velocity differences or relative velocities between the legs are shown in Figure 5.9.





**Figure 5.8:** Percent error in leg lengths for an unperturbed orbit over 1 year

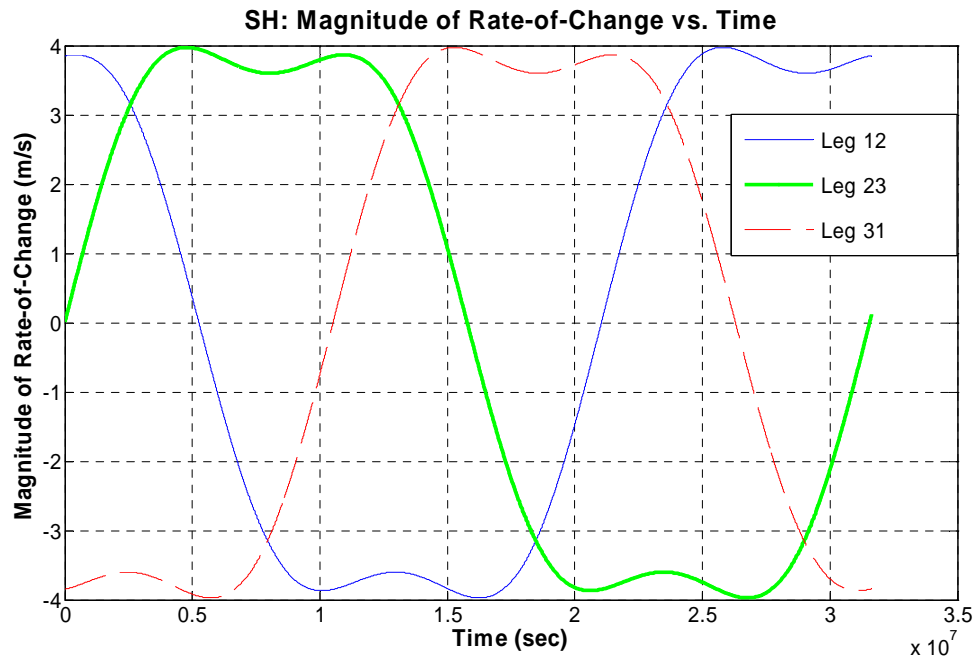


**Figure 5.9:** Relative velocity for unperturbed orbit over 1 year

The magnitude of the rate of change of the leg length is given in Equation 5.2, which is a derivative with respect to time of the position vector.

$$\left\| \frac{d\bar{L}_{ij}}{dt} \right\| = \left\| \frac{d\bar{r}_i}{dt} - \frac{d\bar{r}_j}{dt} \right\| \quad (5.2)$$

The rate-of-change of the LISA orbit vs. time is depicted in Figure 5.10. Note that the mission constraint is satisfied when the rate-of-change does not exceed 15 m/s.



**Figure 5.10:** Rate of change vs. time of an unperturbed LISA orbit for 1 year

The Hughes' nominal orbit is calculated in *NomOrb.m* in Appendix B. Though ultimately not used in this thesis, the nominal orbit of Dhurandhar [18] is shown below. The successful operation of LISA requires the stability of the LISA constellation and constant leg lengths between spacecraft in a perturbed system. Dhurandhar et al. explicitly demonstrate how any configuration of the spacecraft lying in the planes making +/- 60 degree angles with the ecliptic using the Clohessy-Wiltshire equations remains stable, in

that the inter-spacecraft velocities remain constant to first order with respect to the dimensions of the configuration. The trajectory using Dhurandhar's equation is shown in Table 5.2.

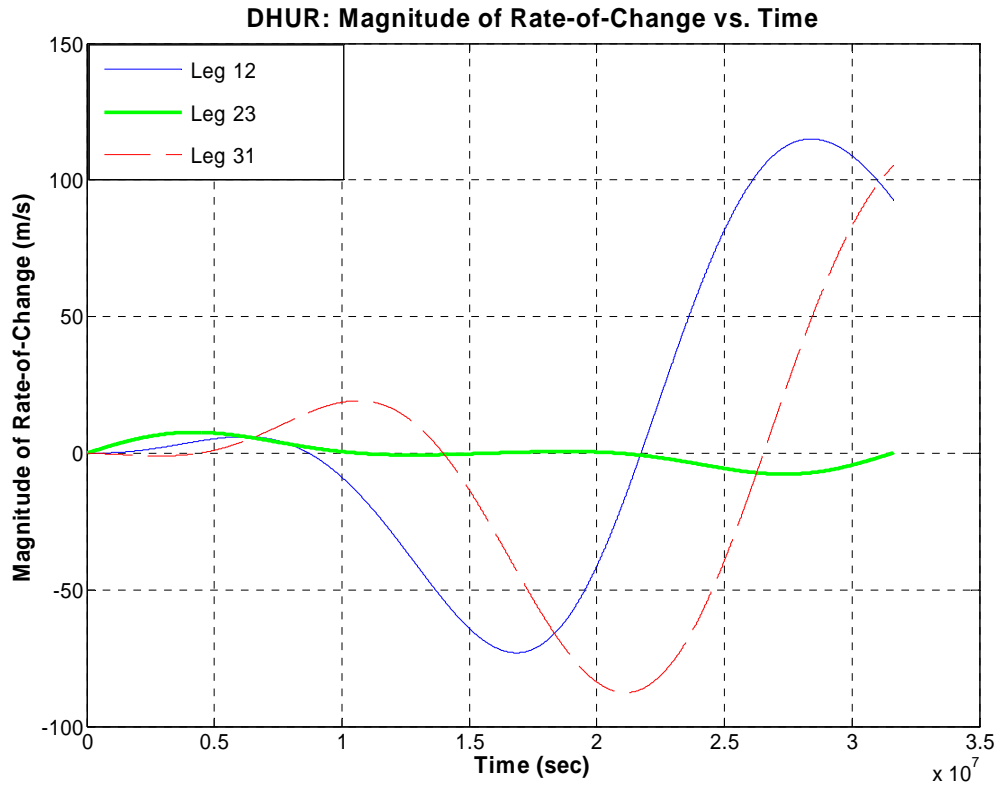
**Table 5.2:** Dhurandhar's nominal orbital elements of LISA & Earth

	$a$ (m)	$e$	$i$ ( $^{\circ}$ )	$\Omega$ ( $^{\circ}$ )	$\omega$ ( $^{\circ}$ )	$M$ ( $^{\circ}$ )
<b>LISA 1</b>	$1.4963 \times 10^{11}$	0.009681	0.96673	0	90	0
<b>LISA 2</b>	$1.4969 \times 10^{11}$	0.0094387	0.95631	118.92	88.349	-117.27
<b>LISA 3</b>	$1.4969 \times 10^{11}$	0.0094387	0.95631	241.08	91.651	117.27
<b>Earth</b>	$1.4960 \times 10^{11}$	$1.3438 \times 10^{-16}$	0	NaN	NaN	0

Dhurandhar proposed using the Clohessy-Wiltshire equations to find a proper orbit for LISA. His initial orbit does not stray too far from nominal guess orbit that Hughes proposed. The argument of periapsis,  $\omega$ , is around  $90^{\circ}$  for all three spacecraft and the phasing on  $M$  and  $\Omega$  are roughly  $120^{\circ}$ . The inclinations are small ( $<1^{\circ}$ ) and the eccentricity is nearly zero, for a nearly perfectly circular orbit. There was some variation found for the period  $P$  of the orbit using Dhurandhar's equations:

$$\begin{aligned}
 P(\text{Earth}) &= 3.1558 \times 10^7 \text{ sec} \\
 P(\text{LISA 1}) &= 3.1567 \times 10^7 \text{ sec} \\
 P(\text{LISA 2 \& 3}) &= 3.1587 \times 10^7 \text{ sec}
 \end{aligned}$$

Figure 5.11 reveals why the Dhurandhar orbit was not used in this thesis. For the unperturbed orbit, Dhurandhar's orbit does not remain constant. Figure 5.11 illustrates the magnitude of rate-of-change extending beyond the 15 m/s mission constraint. In future work, Dhurandhar's orbit would be used in the perturbed system.



**Figure 5.11:** Magnitude of rate-of-change vs. time for Dhurandhar's unperturbed orbit over 1 year

### 5.2.3. The Algorithm

The goal of the batch processor is to estimate the state deviation vector  $x_o$  at a reference time  $t_o$  in order to give the best estimate of the state at that time. Time  $t_o$  is an arbitrary epoch, and all quantities are assumed to be mapped to the epoch using state transition matrix  $\Phi(t_i, t_o)$ .

The input values include initial conditions of reference state  $X^*(t_o)$  and optional a priori state estimates. Although available in the algorithm, the use of a priori estimates will not

be addressed here. Equation 5.3 is the normal equation form of  $\hat{x}_o$  for the weighted least squares. Recall that  $W=R^{-1}$ :

$$(H^TWH)\hat{x}_o = H^TWy \quad (5.3)$$

Equation 5.3 is iterated until  $\hat{x}_o$ , the least squares correction, converges. Note the solution of  $\hat{x}_o$  requires the inversion of information matrix  $M$ :

$$M = (H^TWH) \quad (5.4)$$

The reduced residual vector,  $N$ , is the right-side of Equation 5.3:

$$N = H^TWy$$

The  $M$  and  $N$  matrices are accumulated over the batch of measurements to keep the dimensions of the matrix from growing and to allow for easier inversion as shown below in Equations 5.5 and 5.6:

$$1. \quad H^TWH = \sum_{i=1}^f [\tilde{H}_i\Phi(t_i, t_o)]^T W_i \tilde{H}_i\Phi(t_i, t_o) \quad (5.5)$$

$$2. \quad H^TWy = \sum_{i=1}^f [\tilde{H}_i\Phi(t_i, t_o)]^T W_i y_i \quad (5.6)$$

The state transition matrix  $\Phi(t_i, t_o)$  is attained from integration of  $\dot{\Phi}(t, t_k) = A(t)\Phi(t, t_k)$ .

$A(t)$  is the partial of the time derivative of the state vector  $F(X^*, t)$ , evaluated on the reference trajectory  $X^*$ :

$$A(t) = \frac{\partial F(X^*, t)}{\partial X} \quad (5.7)$$

The code that calculates the state transition matrix (*LISA\_STMderiv.m*) is listed in

Appendix B. The observation-state mapping matrix  $\tilde{H}_i$  is shown below:

$$\tilde{H}_i = \frac{\partial G(X_i^*, t_i)}{\partial X} \quad (5.8)$$

where  $G(X_i^*, t_i)$  is the observation-state relationship evaluated on the nominal, or reference trajectory. The partials are computed by taking the partial derivatives of the observation with respect to the reference trajectory. A listing of the partials is below and will give insight into which measurements give position and velocity information:

#### Range Partial Derivatives

$$\text{Range with respect to position: } \frac{\partial \rho_i}{\partial \bar{R}_i} = \begin{bmatrix} \frac{(x_i - x_E)}{\rho_i} & \frac{(y_i - y_E)}{\rho_i} & \frac{(z_i - z_E)}{\rho_i} \end{bmatrix}$$

$$\text{Range with respect to velocity: } \frac{\partial \rho_i}{\partial \bar{V}_i} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$$

#### Range Rate Partial Derivatives

Range Rate with respect to position:

$$\frac{\partial \dot{\rho}_i}{\partial \bar{R}_i} = \begin{bmatrix} \frac{\dot{X}_i - \left( \frac{X_i - X_E}{\rho_i} \right) \dot{\rho}_i}{\rho_i} & \frac{\dot{Y}_i - \left( \frac{Y_i - Y_E}{\rho_i} \right) \dot{\rho}_i}{\rho_i} & \frac{\dot{Z}_i - \left( \frac{Z_i - Z_E}{\rho_i} \right) \dot{\rho}_i}{\rho_i} \end{bmatrix}$$

$$\text{Range Rate with respect to velocity: } \frac{\partial \dot{\rho}_i}{\partial \bar{V}_i} = \begin{bmatrix} \frac{(x_i - x_E)}{\rho_i} & \frac{(y_i - y_E)}{\rho_i} & \frac{(z_i - z_E)}{\rho_i} \end{bmatrix}$$

#### Right Ascension Angle Partial Derivatives

$$\text{Right Ascension with respect to position: } \frac{\partial \alpha_i}{\partial \bar{R}_i} = \begin{bmatrix} \frac{\sin(\alpha_i) \cos(\alpha_i)}{X_E - X_i} & \frac{-\cos^2(\alpha)}{X_E - X_i} & 0 \end{bmatrix}$$

$$\text{Right Ascension with respect to velocity: } \frac{\partial \alpha_i}{\partial \bar{V}_i} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$$

### Declination Angle Partial Derivatives

Declination with respect to position:

$$\frac{\partial \delta_i}{\partial \bar{R}_i} = \begin{bmatrix} \frac{-\tan(\delta_i) \frac{\partial \rho_i}{\partial X_i}}{\rho_i} & \frac{-\tan(\delta) \frac{\partial \rho_i}{\partial Y_i}}{\rho_i} & \frac{\left(1 - \sin(\delta_i) \frac{\partial \rho_i}{\partial Z_i}\right)}{\rho_i \cos(\delta_i)} \end{bmatrix}$$

$$\text{Declination with respect to velocity: } \frac{\partial \delta_i}{\partial \bar{V}_i} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$$

### Inter-spacecraft Range Partial Derivatives

Inter-spacecraft Range with respect to position (LISA  $i$  range from LISA  $j$ ):

$$\frac{\partial \rho_{Lij}}{\partial \bar{R}_i} = \begin{bmatrix} \frac{(X_i - X_j)}{\rho_{Lij}} & \frac{(Y_i - Y_j)}{\rho_{Lij}} & \frac{(Z_i - Z_j)}{\rho_{Lij}} \end{bmatrix}$$

$$\text{Inter-spacecraft Range with respect to velocity: } \frac{\partial \rho_{Lij}}{\partial \bar{V}_i} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$$

### Inter-spacecraft Range Rate Partial Derivatives

Inter-spacecraft Range Rate with respect to position:

$$\frac{\partial \dot{\rho}_{Lij}}{\partial \bar{R}_i} = \begin{bmatrix} \frac{(\dot{X}_i - \dot{X}_j) + \dot{\rho}_{L13} \frac{(X_i - X_j)}{\rho_{Lij}}}{\rho_{Lij}} & \frac{(\dot{Y}_i - \dot{Y}_j) + \dot{\rho}_{L13} \frac{(Y_i - Y_j)}{\rho_{Lij}}}{\rho_{Lij}} & \frac{(\dot{Z}_i - \dot{Z}_j) + \dot{\rho}_{L13} \frac{(Z_i - Z_j)}{\rho_{Lij}}}{\rho_{Lij}} \end{bmatrix}$$

Inter-spacecraft Range Rate with respect to velocity:

$$\frac{\partial \dot{\rho}_{Lij}}{\partial \bar{V}_i} = \begin{bmatrix} \frac{(X_i - X_j)}{\rho_{Lij}} & \frac{(Y_i - Y_j)}{\rho_{Lij}} & \frac{(Z_i - Z_j)}{\rho_{Lij}} \end{bmatrix}$$

As shown above, the inter-spacecraft range will be computed with respect to the other spacecraft rather than to the Earth tracking station (denoted  $X_E, Y_E, Z_E$ ). For the angle

partial derivatives, MATLAB was used to compute and verify the equations in a compact form. Appendix B contains the MATLAB code (*H\_partials.m*) that calculates the partial derivatives.

The observation sensitivity matrix,  $H_i$ , is given by:

$$H_i = \tilde{H}_i \Phi(t_i, t_o) \quad (5.9)$$

When inverting the accumulated  $M$ , information matrix and the  $N$  reduced residual matrix, a question of observability arises. Observability is crucial for the matrix inversion to work (i.e., sufficient data must exist for all states to be known), since the covariance matrix ( $P_{cov}$ ) is needed to solve the normal equation (i.e.  $(P_{cov}) = M^{-1}$ ). The inverse of a matrix can only be taken if the matrix is *square* ( $m \times m$ ) and *nonsingular*. However, not all square matrices have an inverse. In order to be nonsingular, one of these conditions must be satisfied: 1) no row (column) is a linear combination of the other rows (columns), or 2) the determinant of the matrix *is not* equal to zero [49]. The first condition is related to the *rank* of a matrix. Only a matrix composed of independent equations will be *full rank*, where  $\text{rank}(M) = m$ . Therefore, the rank check is a quick way to determine observability.

Each navigation measurement combination case had to be treated separately in order for the entire program to run cohesively. The  $M$  and the  $N$  matrices were sized depending on the case. For the cases without relative measurements, the initial  $M$  is a 6x6 matrix of zeros, and the  $N$  is a 6x1 vector of zeros, and a separate  $M$  and  $N$  were accumulated for



each spacecraft. For the cases with relative measurements, the additional data requires  $M$  to be an 18x18 vector while  $N$  is an 18x1 vector.

For the relative cases, all three spacecraft states were solved simultaneously. However, in order to obtain observability in all cases, scaling matrices had to be added so that the inverse of  $M$  could be computed. The accumulation of  $M$  and  $N$  is described as:

$$\begin{aligned} M_{SC} &= M_{SC} + ((H_i * SC)^T * W * (H_i * SC)) \\ N_{SC} &= N_{SC} + ((H_i * SC)^T * W * (y_i)) \end{aligned}$$

where,  $SC$  is the scaling vector. The scaling vector,  $SC$  is a square diagonal matrix that does not scale position terms but multiplies velocity terms by  $1 \times 10^{-6}$ .  $SC$  has the same dimensions as  $M$ . In order to remove the scaling vector numerical effects in the results, another scaling matrix was multiplied into the normal equation:

$$\hat{x} = SC * M_{SC}^{-1} * N_{SC} = SC * (P_{cov})_{SC} * N_{SC}$$

or:

$$\hat{x} = M^{-1} * N = P_{cov} * N$$

$X_o^*$  is held constant for the beginning of each iteration. The initial guess orbit,  $X_o^*$  is augmented by the least squares correction value,  $\hat{x}_o$ , after each iteration:

$$(X_o^*)_n = (X_o^*)_{n-1} + (\hat{x}_o)_n \quad (5.10)$$

The recursive nature of the algorithm continues until convergence occurs, when the WBLS algorithm outputs the best estimate of the state,  $\hat{X}$ , which is equivalent to the final corrected  $X_o^*$ . Presented in Figure 5.12 is the weighted batch least squares batch processor algorithm used to estimate the LISA orbit, it is based on an algorithm in [47]. The batch processor code (*LISA\_WBLS.m*) and the code that ran the entire process (*MCRUNbp.m*) is listed in Appendix B.

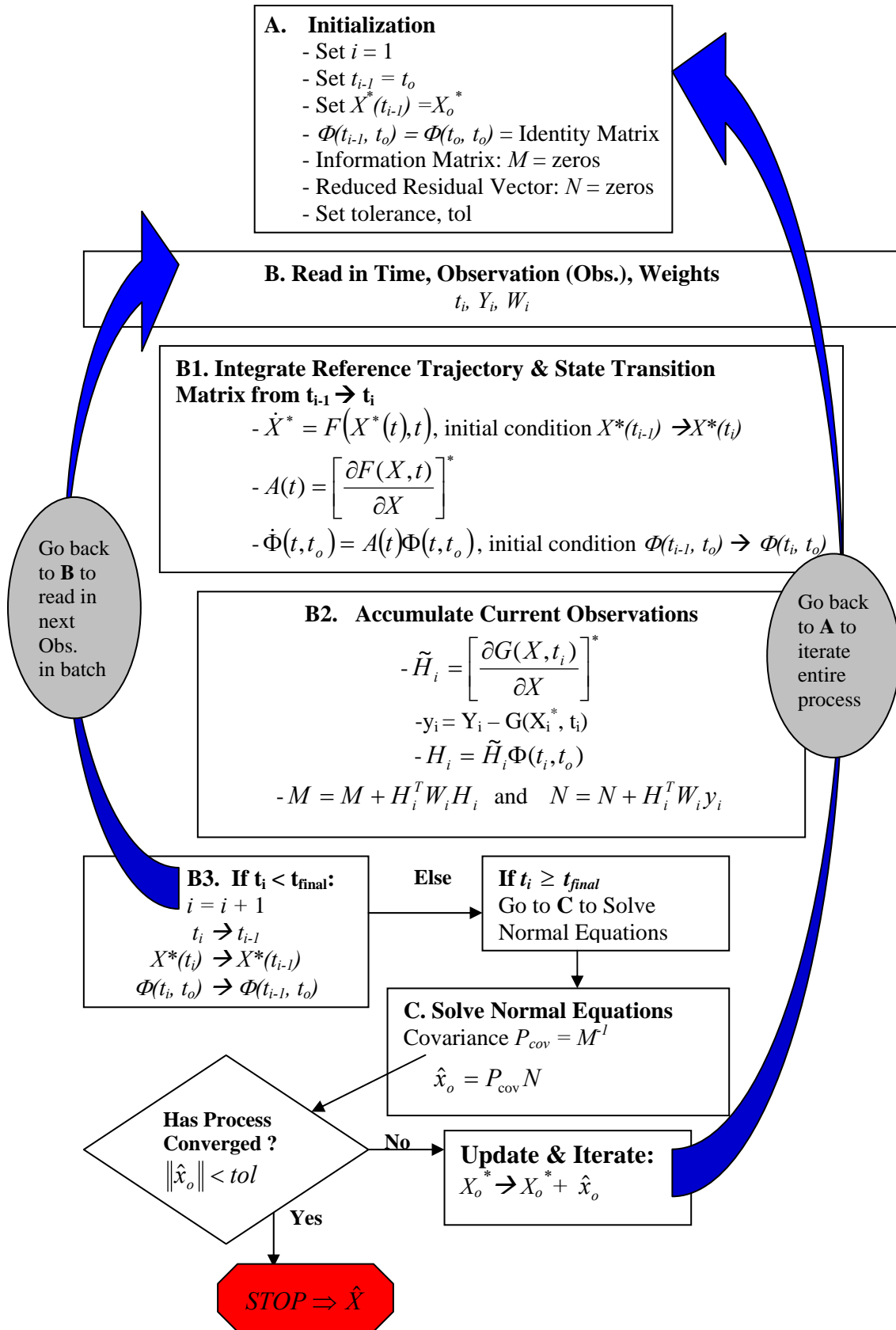


Figure 5.12: Weighted Least Squares Batch Processor Algorithm

#### 5.2.4. Simulation of Measurement Noise

All data is subject to error. Since we nominally generate perfect measurements, noise and bias errors must be simulated in this thesis to study their effects. Noise is defined as a standard deviation ( $\sigma$ ) of the random variation about the measured mean, while biases are constant offsets from a true value [35].

Biases if known can be added or subtracted from each observation. In the test runs with biases, biases were added to the "actual" measurement, but were not subtracted from the calculated measurement. Noises were either set to a specific value or randomly generated for each observation using a normally distributed random number generator (i.e. the "Randn" function in MATLAB) with mean zero, and variance and standard deviation equal to 1.

Time-varying errors such as clock inaccuracies, frequency offsets, ionospheric and tropospheric atmosphere effects, and temperature also affect observations. Some of these errors will be incorporated into the model. Process noise due to error in the mathematical and gravitational modeling of the system dynamics are often not random and are highly correlated with time [35].

The X-band two-way links are employed by the current 2000 DSN system [41] although Ka-band two-way links will be seen in the future. Range is an absolute measure of distance: it is sensitive to measurement biases and random errors, with the range random

error due primarily to thermal noise. The Doppler error is due mainly to fluctuations in the solar plasma along the line-of-sight [41] and clock and frequency offsets. The angle tracking from VLBI will give accuracy up to 50 nanoradians (nrad) or better [41]; therefore it is taken as the assumed accuracy for both the right ascension and declination angle measurements.

The biases due to station location (= 8 cm) and Earth orientation (= 8 cm) are ignored due to the assumptions that the station is located at the center of Earth and that Earth's location is exactly known, as it is not being estimated as was stated in Section 5.0. Biases due to solar plasma are also ignored since LISA is not located near the Sun and Earth's antennas will not have to point at the Sun to track LISA. Additional biases on the range observation are due to an unknown time-tag error ( $\Delta T$ ) and a clock rate offset ( $\Delta \dot{T}$ ) as seen below:

$$\Delta \rho = \Delta T \dot{\rho} \quad (5.11)$$

$$\Delta \rho = c(RTLT)\Delta \dot{T} \quad (5.12)$$

No biases were used on the VLBI angles, though a noise of 50 nrad is a conservative estimate for what would be expected.

In addition to random instrument error, time-varying errors such as range error due to clock instability ( $\Delta \rho$ ) Equation 5.13 and range rate frequency instability ( $\Delta \dot{\rho}$ ) Equation 5.14 were also included.

$$\Delta \rho = \sqrt{2}c(RTLT)\sigma_y(\tau) \quad (5.13)$$

where  $RTLT$  = round trip light time,  $c$  = speed of light and  $\sigma_Y(T)$  = Allan Standard deviation assumed to  $1 \times 10^{-15}$  for a Doppler count time,  $T$ , of 1000 seconds.

Frequency offsets affect the two-way Doppler, or range rate measurement as shown in Equation 5.14:

$$\Delta\dot{\rho} = \sqrt{2 + \log_2(\Pi)} c \sigma_Y(T) \quad (5.14)$$

Where  $\Pi = (RTLT/T)$ , though with hydrogen maser stability, clock instabilities are negligible. Also, the Allan Standard deviation in both equations is negligible compared to other much larger errors in the system such as the troposphere. Although the aforementioned errors are negligible compared to the random instrument noises they are included in the random noise total.

Table 5.3 is a list of the representative error levels that will be incorporated into each simulated measurement. The bullet points below the bold value denote the negligible noise values also incorporated, or breakdown of the bias values used. Many of the random error and bias values listed in Table 5.3 are from Cathy Thornton's "Radiometric Tracking Techniques for Deep-Space Navigation" [41]. The inter-spacecraft range and range rate values were assumed to be the same as the Earth based values.

**Table 5.3:** Random error & bias levels for each measurement [41]

Types	Random Errors ( $\sigma$ )	Biases
<b>Range</b>	<b>60 cm</b> (random error for 60-sec.average) <ul style="list-style-type: none"> <li>▪ clock instability</li> </ul>	<b>2.055 m</b> total bias <ul style="list-style-type: none"> <li>▪ 2 m range instrument bias</li> <li>▪ 1 cm Zenith bias due to troposphere</li> <li>▪ 1 cm line-of-sight fluctuation due to troposphere</li> <li>▪ 3 cm line-of-sight (above 10°) ionosphere bias</li> <li>▪ Time-tag error (small x 10<sup>-4</sup>)</li> <li>▪ Clock rate offset (small x10<sup>-3</sup>)</li> </ul>
<b>Range Rate</b>	<b>0.03 mm/sec</b> (random error for 60-sec. average) <ul style="list-style-type: none"> <li>▪ frequency instability</li> </ul>	_____
<b>Inter-spacecraft Range</b>	<b>60 cm</b> (random error for 60-sec. average) <ul style="list-style-type: none"> <li>▪ clock instability</li> </ul>	<b>2 m</b> <ul style="list-style-type: none"> <li>▪ 2 m range instrument bias</li> <li>▪ Time-tag error</li> <li>▪ Clock rate offset</li> </ul>
<b>Inter-spacecraft Range Rate</b>	<b>0.03 mm/sec</b> (random error for a 60 sec. average)	_____
<b>Angles (VLBI): Right Ascension &amp; Declination</b>	<b>50 nanoradians</b> (for the ~30 min. for once a day maximum)	_____

Noise gets averaged out by the square root, the more time it is averaged over, the less the noise of the measurement is. Measurement noise reduces as the square root of the

number of measurements,  $Num$  increases:  $\frac{\sigma}{\sqrt{Num}}$ . The random errors of Table 5.3 are

averaged over time. For most of the values from Table 5.3, the errors were taken every 60 seconds over a time period of 30 minutes, which means the number of measurements would be equivalent to 30 measurements, or (60 sec/(30 min\*60 sec)). For example, the noise included on the range measurement is:

$$noise = \sigma * \frac{1}{\sqrt{Num}} = 0.6 \text{ m} * (60 \text{ sec}/(30 \text{ min}*60 \text{ sec}))^{1/2} * \text{Randn} = 0.1095 \text{ m} * \text{Randn}$$

Note that a “Randn” is a random number generator in MATLAB that was incorporated into each measurement. The scripts *RealObs.m* and *InterSCD.m* shown in Appendix B respectively tabulate the Earth-based and the inter-spacecraft calculated and observed (i.e. simulated with noise and biases) measurements. As seen in Table 5.3, different accuracies are associated with each measurement type; therefore the residuals will be weighted according to their noises. The weights are the inverse of the observation noise covariance,  $W = R^{-1}$ , where,  $R = \sigma^2 I$ , where  $I$  is the identity matrix. Note the off-diagonal terms are equal to zero:

$$W = \begin{bmatrix} \frac{1}{\sigma_1^2} & \cdot & \cdot & \cdot \\ \cdot & \frac{1}{\sigma_2^2} & \cdot & \cdot \\ \cdot & \cdot & \ddots & \cdot \\ \cdot & \cdot & \cdot & \frac{1}{\sigma_f^2} \end{bmatrix} = \begin{bmatrix} w_1 & \cdot & \cdot & \cdot \\ \cdot & w_2 & \cdot & \cdot \\ \cdot & \cdot & \ddots & \cdot \\ \cdot & \cdot & \cdot & w_f \end{bmatrix}$$

A smaller standard deviation or a larger  $w_i$  will weight the sensor more. The diagonals terms of the  $W$  matrix of each LISA are the values we adopted:

$$\begin{aligned} w_\rho &= 8.327877928620504 \times 10^1 \text{ m}^{-2} \\ w_{\dot{\rho}} &= 3.273113847723235 \times 10^{10} \text{ (m/s)}^{-2} \\ w_\alpha &= 4.0000000000000001 \times 10^{14} \text{ rad}^{-2} \\ w_\delta &= 4.0000000000000001 \times 10^{14} \text{ rad}^{-2} \\ w_{inter-\rho} &= 8.332805331232393 \times 10^1 \text{ m}^{-2} \\ w_{inter-\dot{\rho}} &= 3.3333333333333334 \times 10^{10} \text{ (m/s)}^{-2} \end{aligned}$$

### 5.3. Validity of the Filter

Before the results could be obtained, rigorous testing was done on the algorithm to check that the filter worked, and to give insights into the problem at hand. In orbital mechanics, observability is the ability to apply an estimator to a particular system and obtain a unique estimate for all components of the spacecraft state vector [47]. Unobservability is said to rarely occur for well-formulated problems, hence, the observability criterion tests the validity of the problem formulation and the completeness of the data set. The observability tests the sensitivity of state parameters to the observations. The partial derivatives of the  $\tilde{H}$  Matrix (i.e. the observation-state mapping matrix) give the observability for each measurement; a partial derivative of zero or a small magnitude provides no information to the estimation process. Mathematically, observability and proper conditioning of the matrix allows the inverse of the  $M$  matrix to be taken (i.e.  $(P_{cov}) = M^{-1}$ ). This inversion is crucial since it is needed to solve for  $\hat{x}$ , the least squares correction.

Complete observability requires that the information matrix  $M$  be positive definite. In order to be positive definite, the square  $H$  matrix ( $= \tilde{H}_i \Phi(t_i, t_o)$ ) must be full rank. The mathematical formulation of the problem is made so that only a minimal set of parameters is estimated. Unnecessary parameters cause the solution to be unobservable, and any state variables that are linearly related to each other cannot be uniquely determined [47]. The only parameters being solved for each LISA are the position and velocity (six states). In the cases with relative measurement data, the information matrix



$M$  is 18 x 18 which indicates that the states of all three LISA spacecraft are being solved together. This  $M$  requires a rank of 18 to be observable requiring a complete measurement set over all spacecraft.

As seen in the algorithm, when the fractional change of the root-mean-square (*RMS*) is very small (i.e. equal to some tolerance), the system is considered to have converged thus the iterations terminate. The fractional change is the *RMS* difference between two adjacent iterations. The tolerance drives the outer loop of the algorithm. A final weighted *RMS* of 1 is an indication that the weights used are correct. A weighted *RMS* of less than 1, occurs when errors are overestimated while a greater than 1 value indicates underestimation of the errors [35]. A decreasing trend in the *RMS* is an indicator that the filter is converging. Examining the *RMS* is one way to see how the noises and biases are being treated. One wants a lower *RMS*, ideally  $RMS = 1$  for a weighted case, or  $RMS = 0$  for unweighted *RMS* to get the best fit [35]. The Root-Mean-Square (weighted and unweighted depending on the test) is shown in Equation 5.15:

$$RMS = \left\{ \frac{\sum_{i=1}^f \hat{\epsilon}_i^T W \hat{\epsilon}_i}{m} \right\}^{1/2} \quad (5.15)$$

where weighting matrix  $W = R_i^{-1}$ ,  $m$  is the total number of observations =  $f \times p$ ; and  $\hat{\epsilon}_i$  is the best estimate vector of the observation error:

$$\hat{\epsilon}_i = y_i - H_i \hat{x}_o \quad (5.16)$$

Having a reasonable number of iterations is also a checkpoint. A reasonable filter has been experimentally shown to converge in at most 6 or 7 iterations, regardless of initial

error on the state. More iterations usually indicate an error or an unrealistic tolerance level that is trying to be achieved for that case. In certain cases, a tolerance that is too low may cause filter divergence. The tolerance was set to 0.03 for all cases except DSN (= 0.3), which required higher tolerance for convergence. The higher tolerance also translates into larger standard deviations from the mean when the final statistics are calculated which will be seen for the DSN cases, thereby, making these cases not as reliable as the OD source.

The filter was also tested by inputting an initial state value with no noise, no biases, or guess error. Therefore, what the filter is given is the exact answer and should converge in a single iteration for each case, which the processor was shown to do. In particular, the norm of  $\hat{x}$ , i.e. the least squares correction, is zero in these cases because there is nothing to correct! Other checks to see if the least square correction ( $\hat{x}$ ) is valid includes starting with a guess on LISA 1 being erroneous, for example, only in the X-direction by  $1 \times 10^5$  m, in which case it is expected that only the X-component of  $\hat{X}_1$  requires a negative correction to compensate for the offset. If the Y or Z- position component, or velocity component were to make a huge correction in this case that would indicate a problem with the computation processes.

Each of the four cases tested in the batch processor were tested with variation in noises, and initial guess error (i.e. the state being off in position and/or velocity from the truth). The actual range and angle measurements are sensitive to errors in position, while the range rate is sensitive to both position and velocity errors. An inaccurate initial guess of

position or velocity affects the calculated,  $G_i(X_i^*, t_i)$ , measurements, as well. It also impacts the number of iterations until filter convergence, or in some cases causes the filter to diverge.

A final check was performed on the coding logic. Since this is an iterative process with many parameters, attention has to be paid to make sure that the right variables are updating to the next step for example, the time, the reference and truth trajectory, and state transition matrices within the inner loop. Also, attention was given so that the iterative values are being corrected and properly used as a new initial within the outer loop at the beginning of the next iteration, namely, the updated reference trajectory.

#### **5.4. Orbit Accuracy**

Orbit accuracy is dependent on several factors: amount, type and accuracy of the tracking data, truncation error, round-off error, mathematical model simplifications, and errors in mathematical model of physical system or instrument [47]. The error analysis was performed by comparing solutions of different mixes of tracking data (i.e. cases), weighted with biases and without biases for different tracking arc lengths and number of measurements taken.

OD solution quality will be analyzed two ways. The first analyzes the covariance and parameter uncertainty matrix [41] from single runs of each case over a time interval. The

covariance matrix is a square and symmetrical matrix which estimates error in the computed solution. The variance,  $\sigma^2$ , the diagonal terms describes closeness of the fit to the data. The off diagonal terms are the covariance terms, which contain correlation coefficients,  $\mu_{ij}$  (where  $i$  and  $j$  equals position components  $X, Y, Z$  and/or velocity components denoted  $U, V, W$  here) and the standard deviation,  $\sigma$ . The covariance terms represent the degree of independence among the elements, ideally with small or zeros [35]. The covariance matrix,  $(P_{cov})$  is shown below.

$$(P_{cov}) = (H^TWH)^{-1} = \begin{bmatrix} \sigma_X^2 & \mu_{XY}\sigma_Y\sigma_X & \mu_{XZ}\sigma_Z\sigma_X & \mu_{XU}\sigma_U\sigma_X & \mu_{XV}\sigma_V\sigma_X & \mu_{XW}\sigma_W\sigma_X \\ \mu_{YX}\sigma_X\sigma_Y & \sigma_Y^2 & \mu_{YZ}\sigma_Z\sigma_Y & \mu_{YU}\sigma_U\sigma_Y & \mu_{YV}\sigma_V\sigma_Y & \mu_{YW}\sigma_W\sigma_Y \\ \mu_{ZX}\sigma_X\sigma_Z & \mu_{ZY}\sigma_Y\sigma_Z & \sigma_Z^2 & \mu_{ZU}\sigma_U\sigma_Z & \mu_{ZV}\sigma_V\sigma_Z & \mu_{ZW}\sigma_W\sigma_Z \\ \mu_{UX}\sigma_X\sigma_U & \mu_{UY}\sigma_Y\sigma_U & \mu_{UZ}\sigma_Z\sigma_U & \sigma_U^2 & \mu_{UV}\sigma_V\sigma_U & \mu_{UW}\sigma_W\sigma_U \\ \mu_{VX}\sigma_X\sigma_V & \mu_{VY}\sigma_Y\sigma_V & \mu_{VZ}\sigma_Z\sigma_V & \mu_{VU}\sigma_U\sigma_V & \sigma_V^2 & \mu_{VW}\sigma_W\sigma_V \\ \mu_{WX}\sigma_X\sigma_W & \mu_{WY}\sigma_Y\sigma_W & \mu_{WZ}\sigma_Z\sigma_W & \mu_{WU}\sigma_U\sigma_W & \mu_{WV}\sigma_V\sigma_W & \sigma_W^2 \end{bmatrix}$$

OD accuracy will also be analyzed with a Monte Carlo method, in which the state and measurement equations are simulated using random initial conditions and noises, and OD performance is stored for each case. Such analysis requires numerous OD runs so that a statistically-significant cross section of possible OD scenarios can be analyzed. Each case was run 35 times per day and a random normal distribution was set on the noise. The value of 35 is consistent with what is considered a sufficiently large sample size in the Central Limit Theorem (usually sample size  $>30$ ) however, for a normally distributed sample the larger value of  $n$ , the better the approximation [50]. This allowed for statistics such as the mean and standard deviations of the outputs to be obtained. Monte Carlo is a very time consuming approach because of the number of cases needed to obtain

meaningful statistics, but it does have the capture the effects of non-linearities in the systems; whereas, the covariance analysis is a less time-consuming process, but it ignores any non-linearities in the system.

The summary graphs of Chapter 6 will plot the absolute and relative OD error of position and velocity. The absolute OD accuracy is divided into the position and velocity absolute accuracy, represented by the root-sum-square (*RSS*), or norm expression:

$$RSS = norm = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

The absolute OD accuracies in position (*ABSr*) and velocity (*ABSv*) are taken as the norm of the RSS difference between the estimated LISA state (i.e.  $L1r_{est}$  and  $L1v_{est}$ ) and the nominal LISA state (i.e.  $L1r_{tru}$  and  $L1v_{tru}$ ) as shown below.

$$ABSr = norm\left(\left[ norm(L1r_{est} - L1r_{tru}), norm(L2r_{est} - L2r_{tru}), norm(L3r_{est} - L3r_{tru}) \right]\right)$$

$$ABSv = norm\left(\left[ norm(L1v_{est} - L1v_{tru}), norm(L2v_{est} - L2v_{tru}), norm(L3v_{est} - L3v_{tru}) \right]\right)$$

The relative position (*RELr*) and velocity (*RELv*) error of the constellation is obtained by first taking the norm of the spacecraft estimated leg length ( $L13$ ,  $L21$ ,  $L32$ ) minus the norm of the corresponding truth data leg length: then the norm of all three legs are calculated as shown below.

$$\begin{aligned}
\Delta L13r &= \text{norm}(L1r_{est} - L3r_{est}) - \text{norm}(L1r_{tru} - L3r_{tru}) \\
\Delta L21r &= \text{norm}(L2r_{est} - L1r_{est}) - \text{norm}(L2r_{tru} - L1r_{tru}) \\
\Delta L32r &= \text{norm}(L3r_{est} - L2r_{est}) - \text{norm}(L3r_{tru} - L2r_{tru}) \\
\Delta L13v &= \text{norm}(L1v_{est} - L3v_{est}) - \text{norm}(L1v_{tru} - L3v_{tru}) \\
\Delta L21v &= \text{norm}(L2v_{est} - L1v_{est}) - \text{norm}(L2v_{tru} - L1v_{tru}) \\
\Delta L32v &= \text{norm}(L3v_{est} - L2v_{est}) - \text{norm}(L3v_{tru} - L2v_{tru})
\end{aligned}$$

$$RELr = \text{norm}([\Delta L13r, \Delta L21r, \Delta L32r])$$

$$RELv = \text{norm}([\Delta L13v, \Delta L21v, \Delta L32v])$$

The current inter-spacecraft accuracy requirement of 300 meters [8] may be tightened in the future. Anticipating this constraint, a stricter relative position error of 20 meters will be used in this thesis.

The mean position and velocity error of each LISA is also outputted, along with the final batch processor estimated state vector ( $\hat{X}$ ). This value is used to compute and compare leg length differences and orbital elements.

The sample variance,  $\sigma^2$  is expressed in Equation 5.17:

$$\sigma^2 = \frac{1}{m - \Xi} \sum y_i^T W y_i \quad (5.17)$$

Where,  $\sum y_i^T W y_i$  = sum over all observations (recall that  $y_i$  is the observation deviation vector and  $W$  is the weighting matrix)  
 $m$  = total number of Measurements =  $p \times f$   
 $\Xi$  = number of parameters to be determined from the fit

The parameter uncertainty,  $PU$ , for each state element can be obtained from the sample variance as shown in Equation 5.18:

$$PU = \sqrt{\sigma^2 \times \text{diag}(P_{cov})} \quad (5.18)$$

where, the  $diag(P_{cov})$  term refers to the diagonal elements of the covariance matrix,  $P_{cov}$  corresponding to the state, which was also outputted. The parameter uncertainties are added or subtracted from the estimated state to give a final answer with uncertainties. Other important outputs include *RMS* of each LISA and number of iterations.

The period and semi-major axis are two key orbital elements that will also be assessed.

They are related through:  $P = 2\pi * \sqrt{a^3 / \mu}$ . There are mission constraints on both elements. Maximum period error is 38 seconds from the nominal [8]. Period accuracy is especially crucial since timing affects the range rate measurement. The semi-major axis should be within the range of 0.9 AU and 1.1 AU [8], which indicates that 0.1 AU is the maximum semi-major axis error since 1AU is the nominal. The weighted batch least squares algorithm is expressed in the *LISA\_WBLS.m* (see Appendix B) script which calls the aforementioned scripts for the nominal spacecraft and Earth tracking station trajectory propagated using the EOM, the STM, the simulated observations, and the partials of the observations. *MCRUNbp.m* (see Appendix B) runs the entire process for all test cases and generates comparative statistics in a Monte Carlo analysis.

## CHAPTER 6: Results

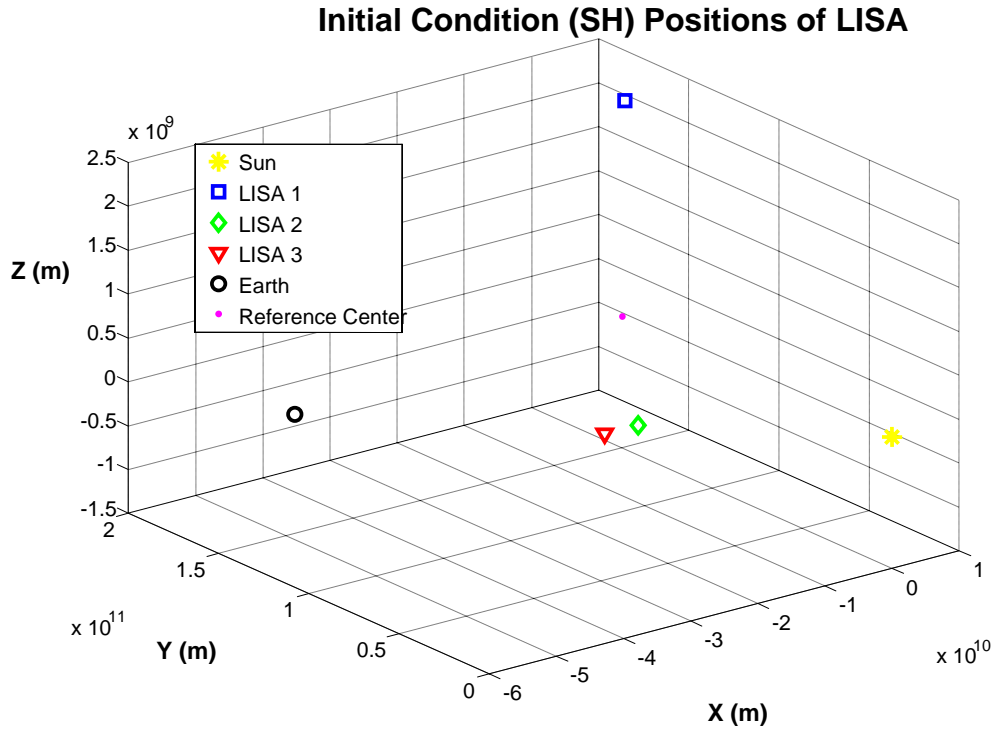
This chapter will provide an overview of the orbit geometry of this problem. Next, it will consider the observation sensitivity matrix ( $H$ ) and the measurements, which will give insights into the main results that follow. All simulation results and some figures were generated in MATLAB.

### 6.1. Comments on Orbit Geometry

A closer look at the LISA simulation orbit geometry is illustrated in the following Figures at the epoch time ( $= 0$ ), 20 days, and 90 days, respectively. Twenty days is where DSN-only becomes effective, assuming a good initial guess. Ninety days is roughly the end of the assumed three-month commissioning period required to put LISA in the proper orbit, so that the drag-free control and the scientific portion of the mission can commence.

Figure 6.1 is a square axis view of the set-up at epoch.



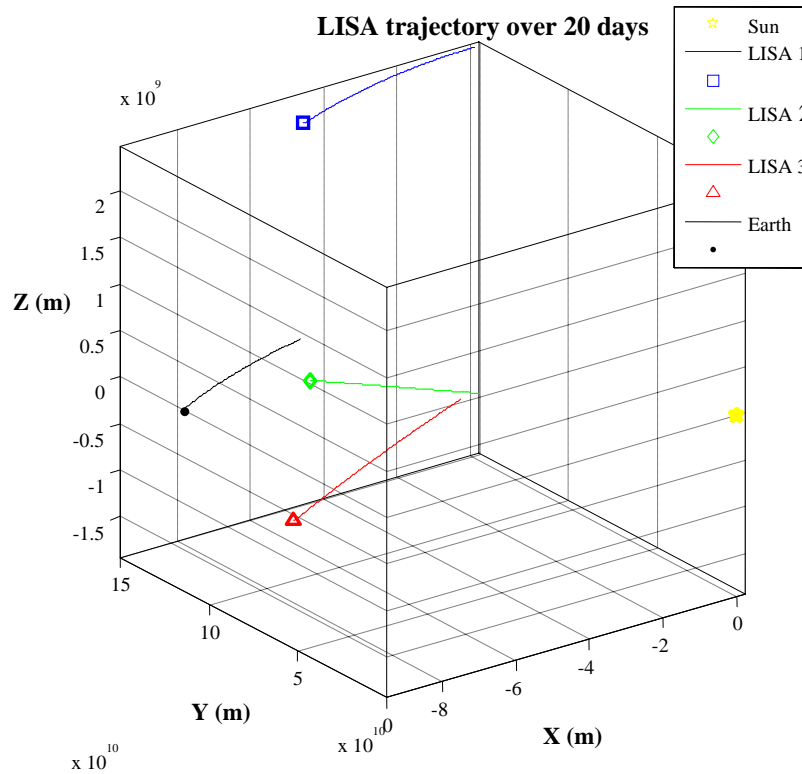


**Figure 6.1:** LISA constellation at epoch (square axis view)

Over LISA’s one year orbital period, the three spacecraft rotate into positions of zero Doppler (i.e. range rate measurement) when viewed from Earth. At these points, the range and Doppler only reveal information about the degrees of freedom (DOFs) along the line to the Earth; hence, the problem is not observable.

This explains why DSN-only (i.e. range and Doppler range rate) measurements are not able to track LISA until a significant arc has been observed. Mathematically, observability is when the  $M$  matrix is full rank. This is found to occur around day 20 when constellation has sufficiently rotated to enable more visibility of all three spacecraft

from Earth as seen in Figure 6.2 below (the *shape* indicates position of the body at day = 20 and the *line* is the path traveled).



**Figure 6.2:** LISA orbit geometry over 20 days

The addition of Relative data (i.e. inter-spacecraft range and range rate) adds information in the directions not observed directly by range and Doppler. For the relative cases, the state of the entire constellation is estimated at once from inverting the 18x18 information matrix,  $M$ . Extra information is obtained about the legs of the constellation, and the location (i.e. range) and velocity (i.e. range rate) of the spacecraft with respect to each other.

The  $M$  matrix is an accumulation over the measurement batch and contains the observation sensitivity, or  $H$  matrix. The partials in the observation-state mapping matrix  $\tilde{H}$  give observability information on each measurement and multiplying it by the STM maps the observability to a specific time. An abridged version (18x6) of the  $H$  partial matrix in Table 6.1, for the DSN/VLBI/Relative case at day 20 takes an explicit look at how the measurements affect the determination of the estimated state, for LISA 1.

**Table 6.1:** Abridged version of DSN/VLBI/Relative  $H$  matrix at day 20

DSN/VLBI/Rel.	LISA 1						LISA 2	LISA 3
Day =20	Position			Velocity				
measurement	$X$	$Y$	$Z$	$\dot{X}$	$\dot{Y}$	$\dot{Z}$	....	....
$\rho L1$	<b>8.8E-01</b>	<b>4.7E-01</b>	<b>4.6E-02</b>	0	0	0	0	0
$\dot{\rho}L1$	<b>-9.6E-08</b>	<b>1.8E-07</b>	<b>-2.9E-09</b>	<b>8.8E-01</b>	<b>4.7E-01</b>	<b>4.6E-02</b>	0	0
$\delta L1$	<b>-7.9E-13</b>	<b>-4.3E-13</b>	<b>2.0E-11</b>	0	0	0	0	0
$\alpha L1$	<b>-9.4E-12</b>	<b>1.7E-11</b>	0	0	0	0	0	0
$\rho L13$	<b>3.3E-01</b>	<b>-4.0E-01</b>	<b>8.5E-01</b>	0	0	0	0	<b>X</b>
$\dot{\rho}L13$	<b>-9.8E-08</b>	<b>-1.7E-08</b>	<b>3.0E-08</b>	<b>3.3E-01</b>	<b>-4.0E-01</b>	<b>8.5E-01</b>	0	<b>X</b>
$\rho L2$	0	0	0	0	0	0	<b>X</b>	0
$\dot{\rho}L2$	0	0	0	0	0	0	<b>X</b>	0
$\delta L2$	0	0	0	0	0	0	<b>X</b>	0
$\alpha L2$	0	0	0	0	0	0	<b>X</b>	0
$\rho L21$	<b>-6.1E-01</b>	<b>-5.6E-01</b>	<b>5.6E-01</b>	0	0	0	<b>X</b>	0
$\dot{\rho}L21$	<b>-3.6E-08</b>	<b>-9.3E-08</b>	<b>-1.3E-07</b>	<b>-6.1E-01</b>	<b>-5.6E-01</b>	<b>5.6E-01</b>	<b>X</b>	0
$\rho L3$	0	0	0	0	0	0	0	<b>X</b>
$\dot{\rho}L3$	0	0	0	0	0	0	0	<b>X</b>
$\delta L3$	0	0	0	0	0	0	0	<b>X</b>
$\alpha L3$	0	0	0	0	0	0	0	<b>X</b>
$\rho L32$	0	0	0	0	0	0	<b>X</b>	<b>X</b>
$\dot{\rho}L32$	0	0	0	0	0	0	<b>X</b>	<b>X</b>

In Table 6.1, each measurement contributes to knowledge of the state at a different level.

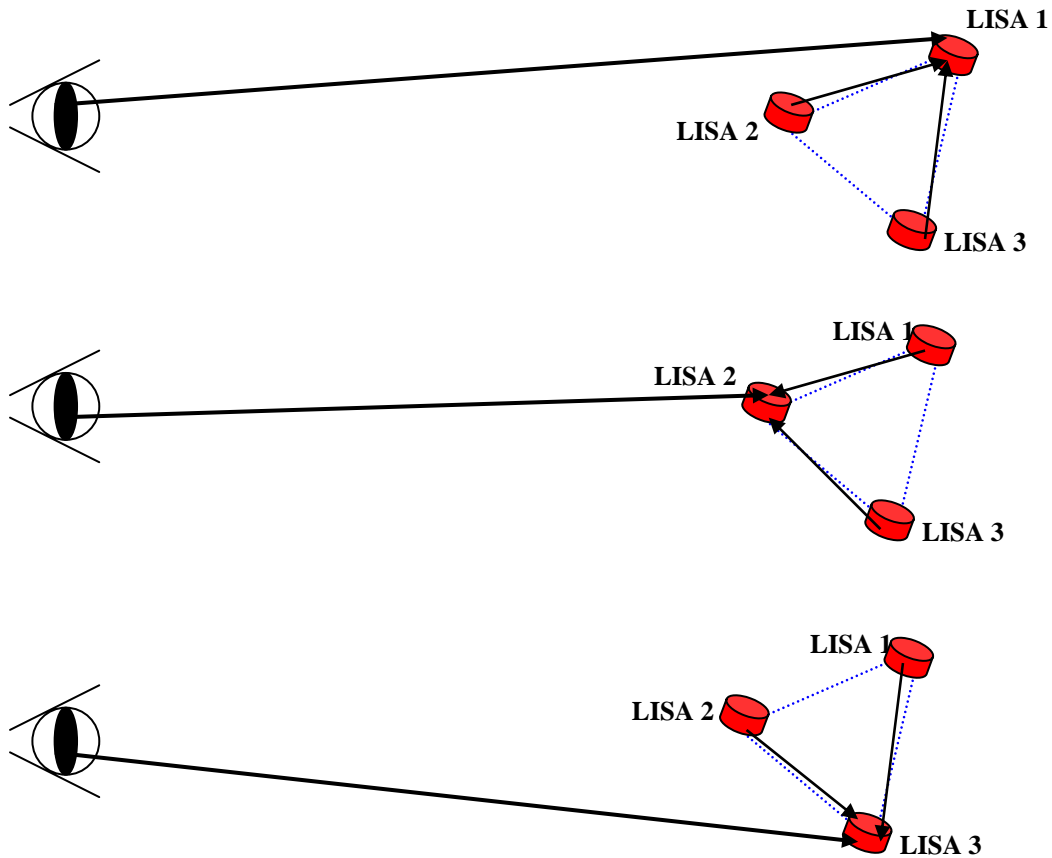
Range (e.g.,  $\rho L1$ ) from Earth only provides information most prominent in the  $X$ - and  $Y$ -

components, whereas, the  $Z$ -component is less discernible because the LISA constellation as viewed from Earth is smallest along the  $Z$ -axis.

Similarly, the range rate from Earth (e.g.,  $\dot{\rho}L1$ ) has the largest partials ( $\times 10^{-1}$ ) in the  $\dot{X}$  - and  $\dot{Y}$  -component of velocity, and it reveals much smaller information about the  $\dot{Z}$  - component. Range rate also gives some position information, albeit, of much lesser value than for velocity. However, for the inter-spacecraft data, range ( $\rho L13$  and  $\rho L21$ ) discloses information of equal magnitude ( $\times 10^{-1}$ ) of all three position components. Moreover, inter-spacecraft range rate (e.g.,  $\dot{\rho}L13$  and  $\dot{\rho}L21$ ) provides data for all three velocity components. Declination (e.g.,  $\delta L1$ ) is most observable in the  $Z$ -component ( $\times 10^{-11}$ ), a magnitude which appears small compared to range, or range rate, but is large for angles measured in the units of radians. Right ascension (e.g.,  $\alpha L1$ ) is most observable in the  $X$  and  $Y$ -component.

Table 6.1 clearly shows that LISA 1 is sensitive to leg 13, but also gives information about leg 21. The same trend is seen for LISA 2, which is set up to measure range and range rate of leg 21, but also contains information on leg 32. LISA 3 includes information on leg 32 in addition to leg 31, where the "X" in Table 6.1 denotes a value for that measurement in the abridged chart. The 60 degree angle of the constellation triangle means that any one spacecraft is well-observed in the plane of the constellation by the other two; since information can still be obtained about the location of the spacecraft despite poor DSN data through this triangulation. Figure 6.3 illustrates this concept of triangulation and how, with the addition of relative range and range rate data,

more information is obtained about a spacecraft. In the illustrations, an "eyeball" is ranging to one LISA (in the order LISA 1, LISA 2 LISA 3), but picks up relative information about the adjacent two legs.



**Figure 6.3:** Eyeball views of triangulation

Figure 6.3 also supports an earlier statement (Chapter 5) that the measurement noise reduces as the square root of the number of measurements. A complete relative inter-spacecraft dataset is similar to having a single spacecraft dataset measured three times as

often. Thus, relative data should effectively be the same as lowering the noise of the range and Doppler data types by a factor of  $\sqrt{3}$ .

In the non-relative cases, the states of the three spacecraft are estimated separately and only information pertaining to that spacecraft is measured, as seen in the  $H$  matrix (6x6) below for LISA 1 for DSN/VLBI at day 90.

**Table 6.2:** DSN/VLBI  $H$  matrix for LISA 1 at day 90

DSN/VLBI	LISA 1					
Day = 90	$X$	$Y$	$Z$	$\dot{X}$	$\dot{Y}$	$\dot{Z}$
$\rho L1$	-1.61E-01	9.87E-01	1.69E-04	0	0	0
$\dot{\rho} L1$	-2.02E-07	-3.29E-08	-1.01E-08	-1.61E-01	9.87E-01	1.69E-04
$\delta L1$	5.51E-16	-3.39E-15	2.04E-11	0	0	0
$\alpha L1$	-2.01E-11	-3.27E-12	0	0	0	0

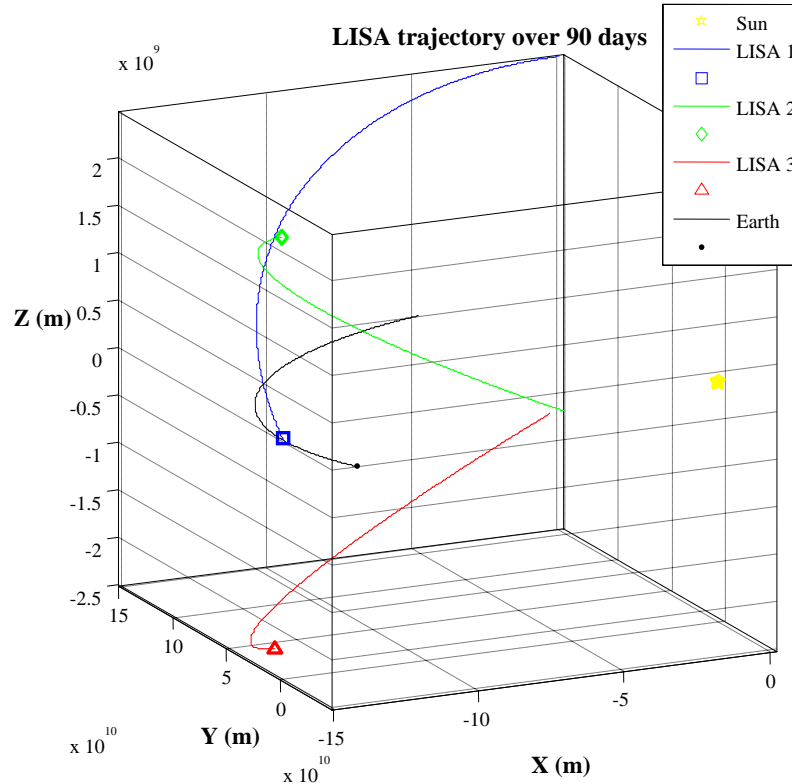
For both  $H$  matrices (Tables 6.1 and 6.2), there is no velocity partial information attained from the range ( $\rho$ ), inter-spacecraft range ( $\rho Lij$ ), or angle measurements ( $\alpha$  and  $\delta$ ).

Velocity information is achieved only with range rate ( $\dot{\rho}$ ) and inter-spacecraft range rate ( $\dot{\rho} Lij$ ). All measurement partials contain position information. However, there is no  $Z$ -component information in right ascension angle ( $\alpha$ ). The components of Table 6.2 retain the same signatures as discussed in Table 6.1, though the direction and size are dissimilar at day 90 due to the different orbit station.

Angle data is shown to help initially by augmenting unobservable DSN, which will be evidenced by obtainable results of DSN/VLBI for small tracking arcs. With a good initial guess, DSN/VLBI values can be achieved for a tracking arc of one day. The angle data given with VLBI is the right ascension ( $\alpha$ ), which locates the spacecraft in the  $X$ - $Y$

plane, and declination ( $\delta$ ), which orients the spacecraft vertically. Knowledge of distances, or positional components are provided by this data. Long arcs of Doppler and range can provide angular position accuracy of 40 nrad or better [41]; hence this could explain why the angular data with accuracy of 50 nrad is not useful compared to range and Doppler after a sufficiently long tracking arc has been traversed.

Figure 6.4 shows the LISA simulation at the end of the three-month commissioning period. Due to the significant tracking arc all spacecraft are observable from Earth.



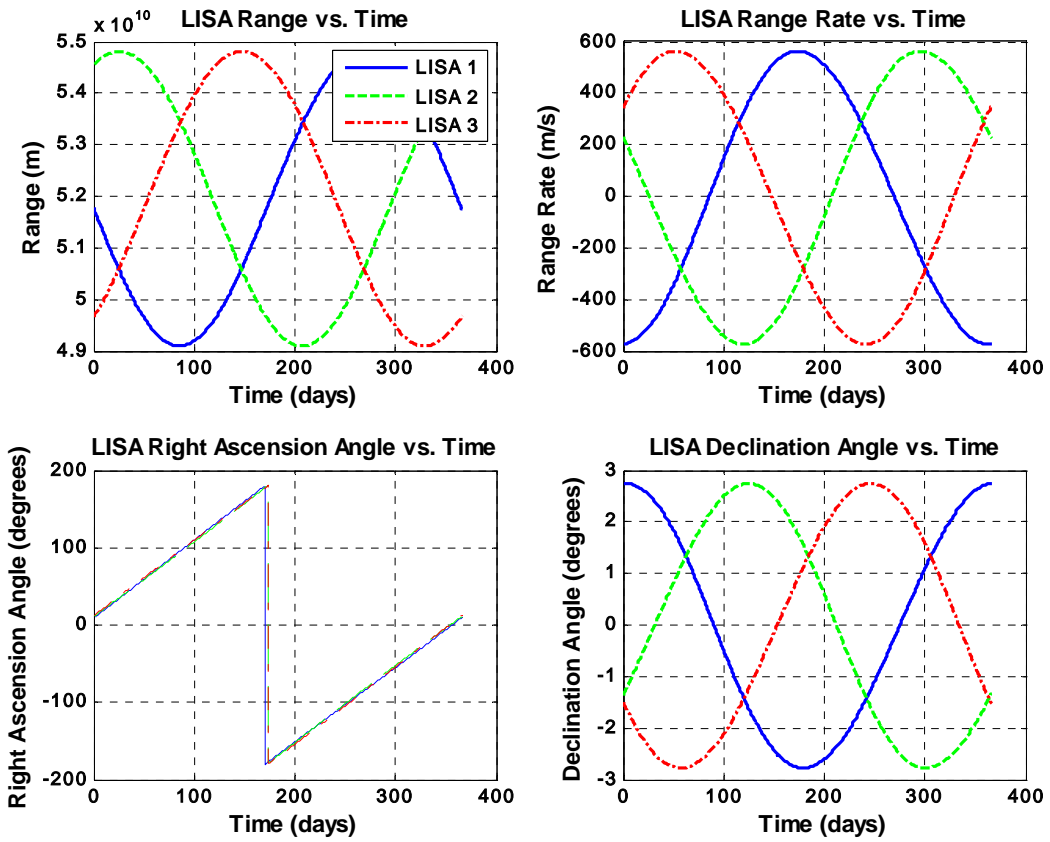
**Figure 6.4:** LISA orbit geometry over 90 days

## 6.2. Measurements Over Time

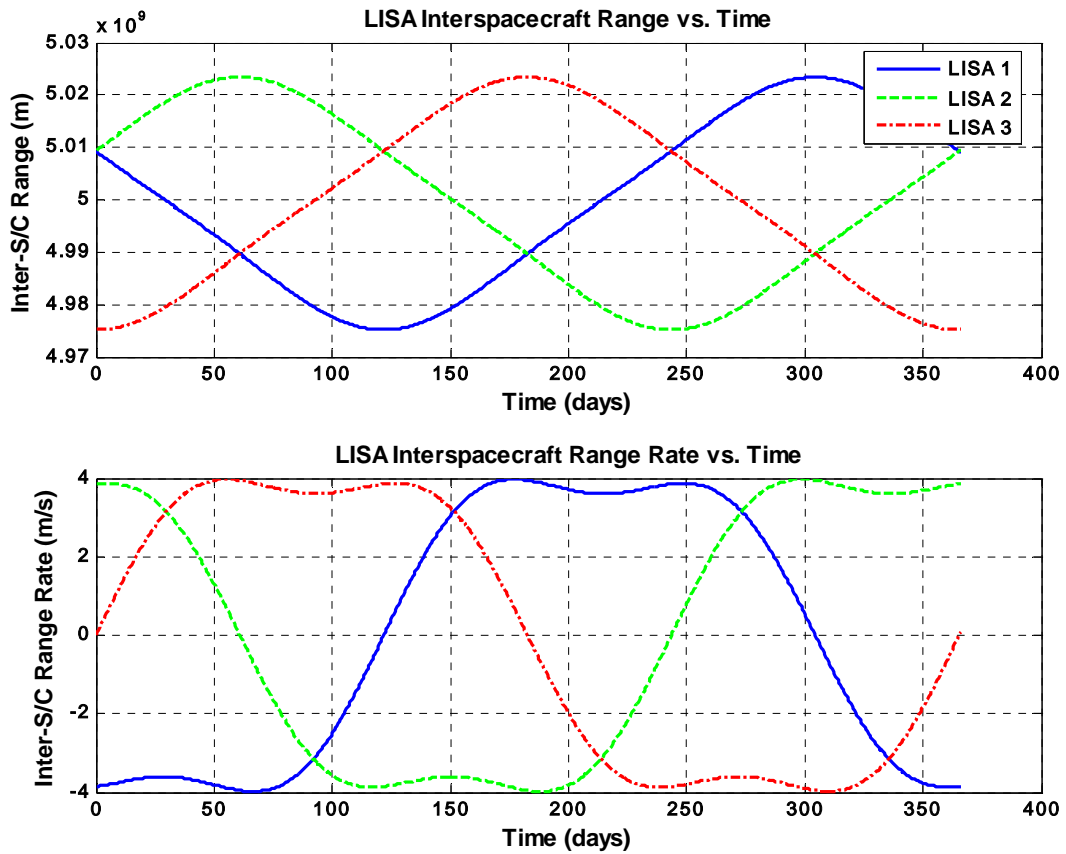
Figures 6.5 and 6.6 examine how measurements from Earth, and the inter-spacecraft measurements, would look over the course of a year. Note the low declination of the LISA orbit, varying from +/- three degrees. The range and range rate follow a sinusoidal pattern.

The inter-spacecraft range and range rate varies in a sinusoidal fashion, hence, recall these values were plotted earlier in the nominal orbit discussion (Section 5.2.2). The inter-spacecraft range varies about its nominal leg length (i.e.  $5 \times 10^9$  m). A mission requirement is that the LISA arm length should be +/- 1% of the  $5 \times 10^9$  m [8] (i.e. between  $4.95 \times 10^9$  m and  $5.05 \times 10^9$  m), even with noisy and biased data, inter-spacecraft range values fall within that constraint in this two-body problem set-up.





**Figure 6.5:** Earth-based nominal measurements over 1 year



**Figure 6.6:** Inter-spacecraft nominal measurements over 1 year

### 6.3. Main Analysis

The main analysis was done on the four main cases:

- 1) DSN/VLBI (D/V)
- 2) DSN/VLBI/Relative (D/V/R)
- 3) DSN only
- 4) DSN/Relative (D/R)

Note that the "Relative" refers to the inter-spacecraft range and range rate data. For the main analysis, a series of 35 randomly-generated simulations for each case were executed

to perform a Monte Carlo analysis. The means and standard deviations of outputs were collected to represent error statistics. The cases were run for the roughly three-month (or 90-day) commissioning period. A summary of the Monte Carlo analysis characteristics are listed below, given noises and biases previously described in Section 5.2.4:

- Random number generated noise
- Bias (Yes and No)
- Random generated Initial Guess Error if indicated
- Each case was run from 1 to 90 days
- Ran 35 times per day per case
- With 1 measurement/day

Two Monte Carlo runs were performed on each case:

- 1) Noise, Bias, Random Initial Guess Errors, 1 measurement/day (NB)
- 2) Noise, No Bias, No Initial Guess Error, 1 measurement/day (noB)

One case of three measurements per day was also done on the noise, bias Monte Carlo (NB) run to compare the effects of more frequent measurements on the estimation.

Prior to this main analysis, single runs were also performed over the entire orbital period (i.e. 365 days) to observe trends. Since each case was run just one time, the single runs were faster than the Monte Carlo runs. A random number generator was not used in the single runs; so in some cases, the noise was turned on at its full value, which is treated as an additional bias since it is now constant; and other cases turned off the noise. While single runs are not the focus of the analysis, they did allow data patterns to be observed, over longer time periods, with larger biases, as well as testing more measurements per day.

The covariance run was done on a case with no noise, no bias, and no initial guess error to generate a covariance plot on each case. In essence, the covariance run reveals the

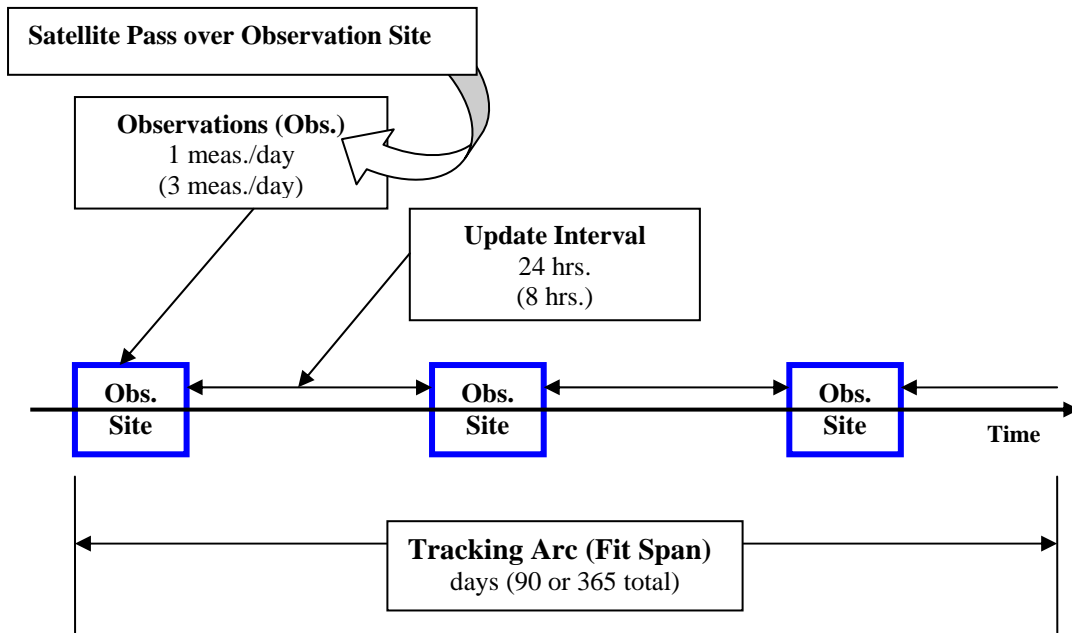
standard deviation of each case. Recall from Chapter 5 that the covariance matrix is a square and symmetrical matrix with the variance (or standard deviations squared) along its diagonal, which discloses the quality of the state estimate. The covariance run described below presumed the case:

- 1) No Noise, No Bias, 1 measurement/day

with

- No Initial Guess Error
- Each case was run from 1 to 365 days
- Each case ran 1 time per day

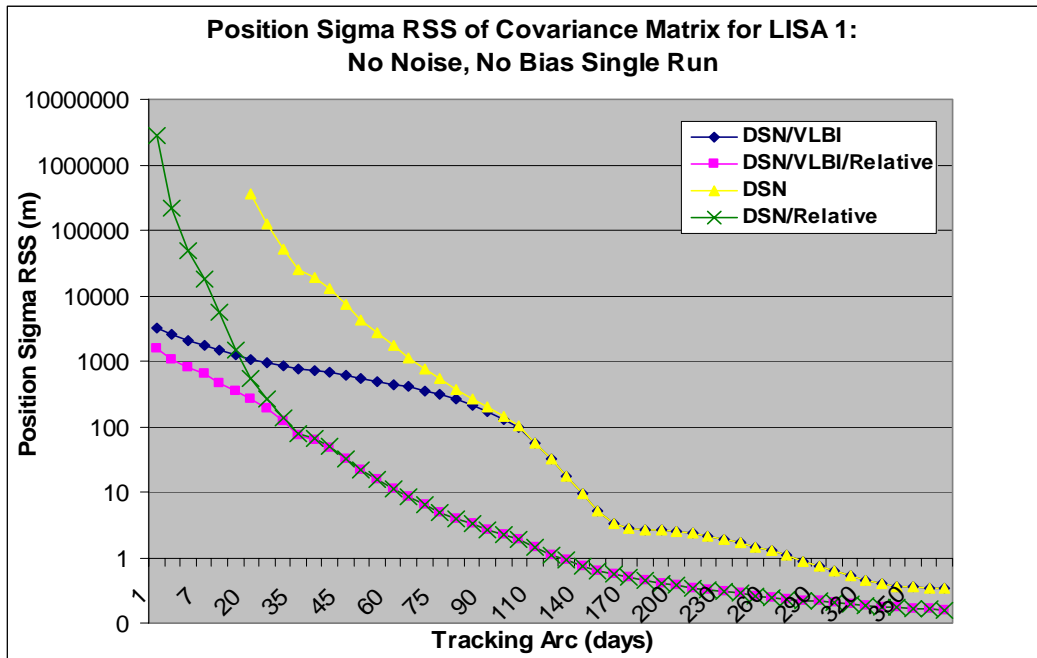
Figure 6.7 illustrates the frequency of observations obtained at each update interval over the entire tracking arc.



**Figure 6.7:** Frequency of observations

### 6.3.1. Covariance Analysis

A no noise, no bias run is really a test of how well the filter works. This case appropriately computes 0 for all the error outputs. The diagonals of the covariance matrix and the estimated state are the only information extracted from this case. The first insight seen early in testing is that the DSN-only case was found to be unobservable until day = 20, therefore, its tracking starts then. For this set-up, it took two iterations for all cases to converge to their tolerance. The root-sum-square (RSS) of the square root of the covariance matrix ( $P_{cov}$ ) diagonals were plotted, i.e. the standard deviations (sigma, or  $\sigma$ ). Figure 6.8 shows the RSS of position sigmas for LISA 1. The covariance graphs for LISA 2 and 3 show similar trends.



**Figure 6.8:** RSS of position sigmas of the Covariance Matrix for LISA 1

Around day 30, DSN/VLBI/Relative and DSN/Relative, overlap and it takes about 90 days for DSN/VLBI and DSN to overlie. These overlaps suggest that VLBI or angle data

eventually becomes ineffectual compared to the other measurement types.

DSN/VLBI/Relative starts out with the least standard deviation, followed by DSN/VLBI.

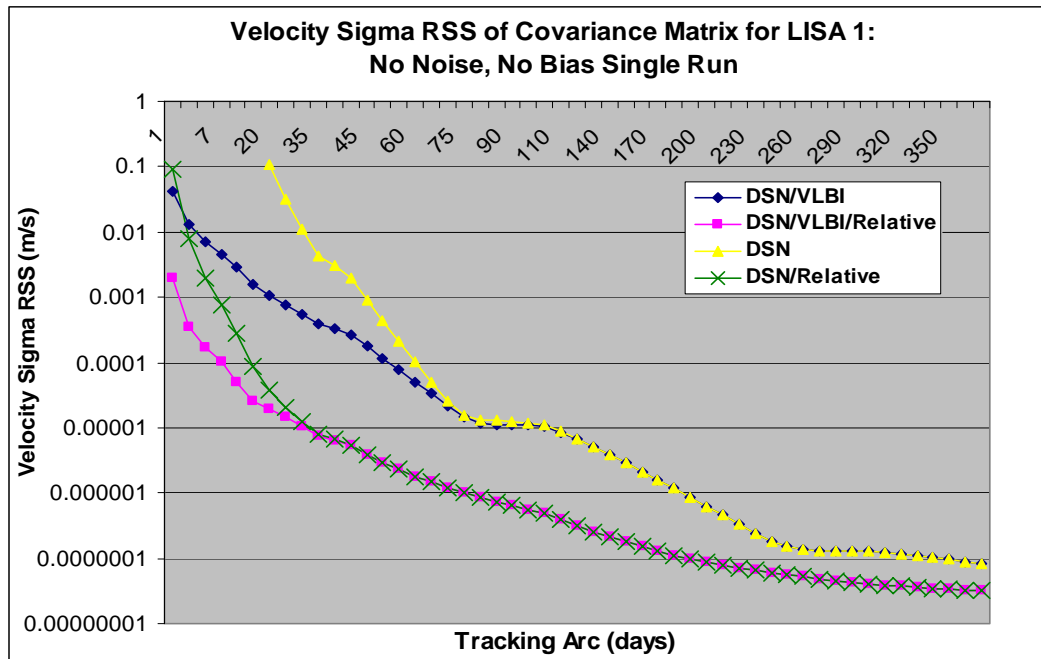
This suggests the importance of VLBI observations early in the mission phase. At 15

days, DSN/Relative surpasses DSN/VLBI. After day 15, the two cases with Relative

information have the least standard deviation for the remainder of the run, concluding

that Relative data gives a clear advantage in accuracy over non-relative data. Figure 6.9

illustrates the RSS of velocity sigmas for LISA 1:



**Figure 6.9:** RSS of velocity sigmas of the Covariance Matrix for LISA 1

In Figure 6.9, the DSN/Relative velocity sigma surpasses DSN/VLBI much earlier by day

3, as indicated in the graph. DSN/VLBI and DSN overlap at day 70. DSN/VLBI/Relative

and DSN/Relative overlie around day 30.

In both Figures 6.8 and 6.9, there is a decreasing trend to all lines. As the tracking arc

and number of measurements increase, the RSS sigmas of the covariance decrease.

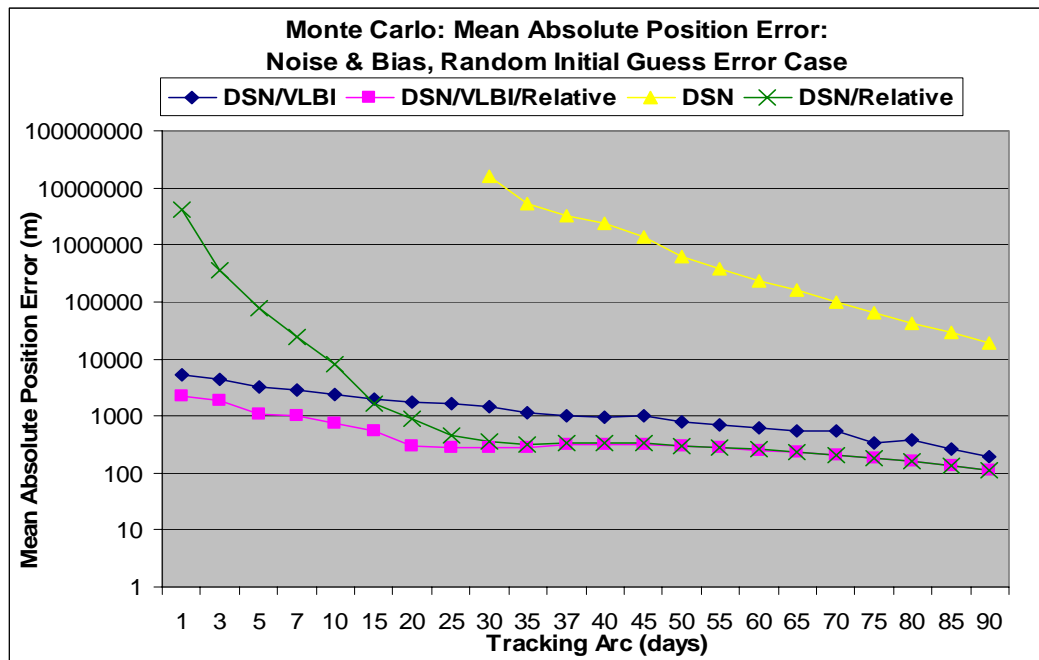
DSN/VLBI and DSN mimic each other; and DSN/VLBI/Relative overlaps DSN/Relative. These figures indicate that with Relative data VLBI adds value for the first 30 days. Without Relative data, it is advantageous to have VLBI for a longer time as indicated in both graphs judging by how long the DSN/VLBI case outperforms DSN. The addition of Relative data improves the overall performance, as well. While VLBI helps initially, it is the presence of Relative data that gives a clear advantage in spacecraft state knowledge for the remainder of the orbital period.

### **6.3.2. Monte Carlo Runs**

The Monte Carlo (MC) runs were performed for three months (or 90 days), the commissioning period as well as a period considered sufficient based on previous runs to capture most measurement/convergence dynamics. Each case was iterated 35 times per day. The noises were randomly generated. The initial guess error was only randomly generated for the Monte Carlo (noise/bias denoted "NB") case. From day 1 to 19, the initial guess error was randomly generated with a maximum of +/-10 km in position and +/-1 x 10<sup>-5</sup> km/sec in velocity. However, after 20 days when the DSN-only case was added the random generator was reduced to 1 km since DSN would often be "close to singular, or badly scaled," thus, unable to converge with even that much initial guess error. No checks were done on the upper limit of how much guess error the other cases could take, yet they were consistently able to converge with the prior stated offset (+/- 10 km) offset. In addition to DSN requiring a sufficiently long tracking arc, the initial guess for DSN cannot be too far from the actual state. For any case, a bad initial guess could

lead to divergence of the solution. However the other cases can compensate better for a bad initial guess because they have additional observation types (angles and/or relative).

Figure 6.10 shows the mean absolute position error. For this simulation, random guess errors were random +/- 10 km, therefore, the DSN measurement was only observable from approximately day 30. Thus, its tracking began there.



**Figure 6.10:** MC Simulation, NB, Mean Absolute Position Error

The 1-sigma standard deviation error bars of Figure 6.11 show that DSN/VLBI/Relative remains distinct until day 15 then it begins to overlap with DSN/Relative. Around day 75, DSN/VLBI/Relative overlaps with DSN/VLBI. By day 10, DSN/Relative and DSN/VLBI are overlapping. DSN remains distinct from all the other data indicating it will never be as accurate over a 90 day tracking arc. Figure 6.12 shows the mean absolute velocity error with 1-sigma ( $1-\sigma$ ) standard deviation error bars.



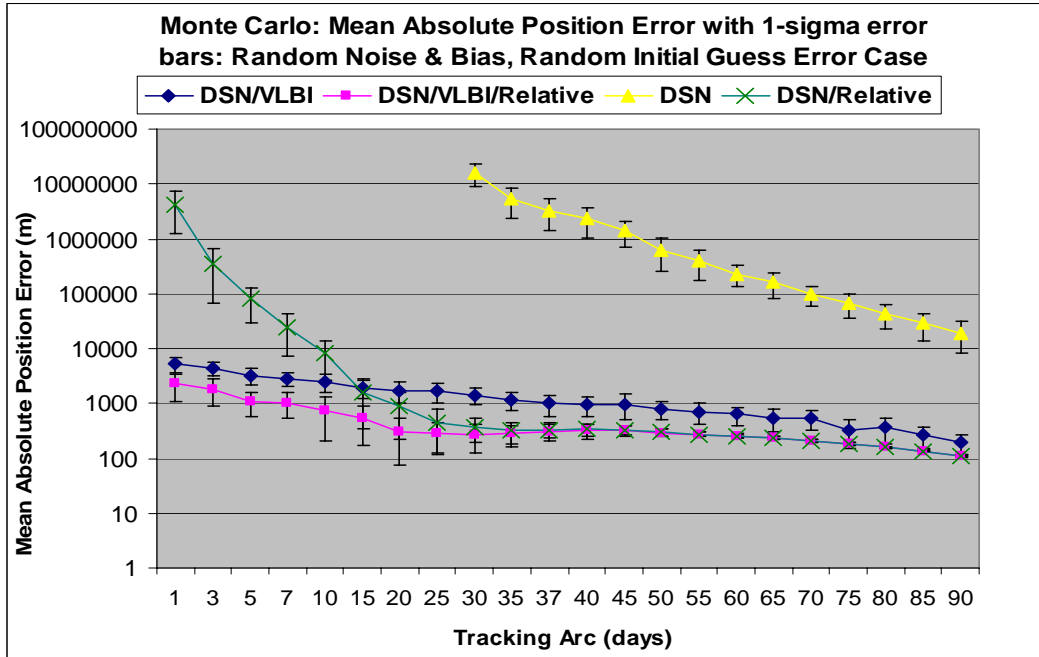


Figure 6.11: MC Simulation, NB, Mean Absolute Position Error with sigmas

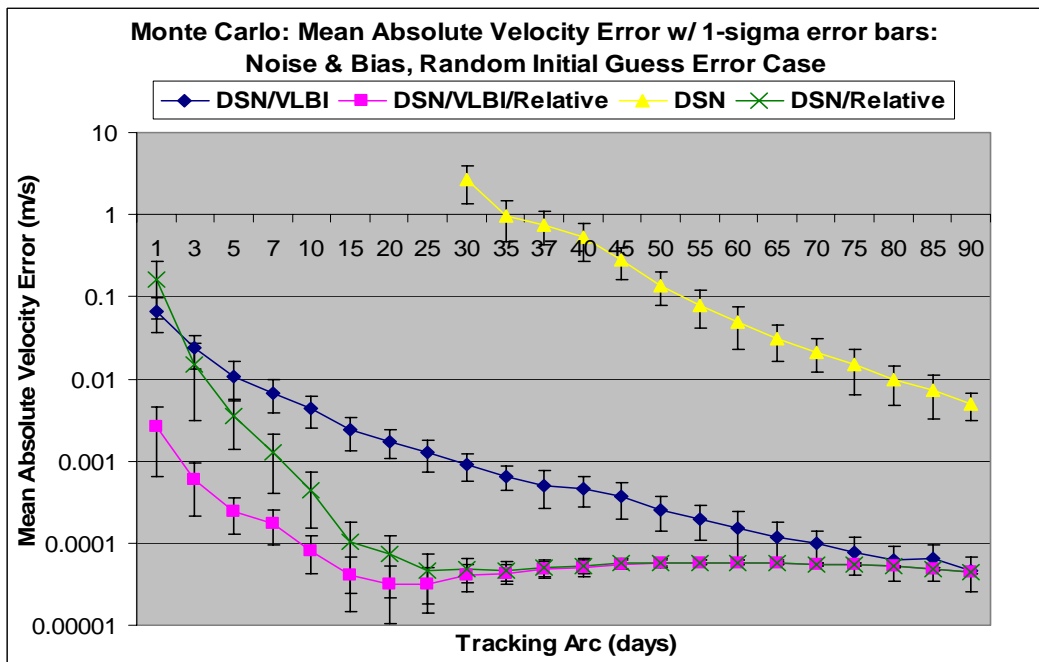


Figure 6.12: MC Simulation, NB, Mean Absolute Velocity Error with sigmas

Table 6.3 is a summary of the raw data of mean absolute position and velocity errors and 1-sigma standard deviations for the cases at 1, 20, 30, 60, and 90 days. There is the expected  $\sqrt{3}$  ( $\sim 1.7$ ) improvement for absolute OD accuracy of the Relative cases. The

absolute error improvements are much greater than  $\sqrt{3}$  for shorter tracking arcs, though the gap between the Relative and non-Relative cases decrease over time.

**Table 6.3:** Summary of Mean Absolute Position & Velocity with Standard Deviations

<b>Mean Absolute Position Error (m) (+/- 1-<math>\sigma</math>)</b>				
<b>days</b>	<b>DSN/VLBI</b>	<b>DSN/VLBI/Relative</b>	<b>DSN</b>	<b>DSN/Relative</b>
<b>1</b>	5294.6 (+/- 1639.9)	2293 (+/- 1195)	-----	$4.26 \times 10^6$ (+/- $3.00 \times 10^6$ )
<b>20</b>	1743.2 (+/- 699.57)	300.99 (+/- 224.83)	-----	881.07 (+/- 665.41)
<b>30</b>	1426.6 (+/- 493.73)	273.03 (+/- 150.03)	$1.60 \times 10^7$ (+/- $7.15 \times 10^6$ )	363.43 (+/- 169.2)
<b>60</b>	632.86 (+/- 229.53)	254.07 (+/- 20.436)	2.30e+05 (+/- 97931)	255.49 (+/- 20.851)
<b>90</b>	188.92 (+/- 86.714)	112.54 (+/- 4.7928)	19378 (+/- 11069)	112.61 (+/- 4.786)
<b>Mean Absolute Velocity Error (m/s) (+/- 1-<math>\sigma</math>)</b>				
<b>days</b>	<b>DSN/VLBI</b>	<b>DSN/VLBI/Relative</b>	<b>DSN</b>	<b>DSN/Relative</b>
<b>1</b>	0.066 (+/- 0.030)	0.00257 (+/- 0.00193)	-----	0.16246 (+/- 0.10881)
<b>20</b>	0.0017 (+/- $6.7 \times 10^{-4}$ )	$3.19 \times 10^{-5}$ (+/- $2.16 \times 10^{-5}$ )	-----	$7.24 \times 10^{-5}$ (+/- $5.11 \times 10^{-5}$ )
<b>30</b>	$8.93 \times 10^{-4}$ (+/- $3.35 \times 10^{-4}$ )	$4.04 \times 10^{-5}$ (+/- $1.52 \times 10^{-5}$ )	2.635 (+/- 1.273)	$4.88 \times 10^{-5}$ (+/- $1.64 \times 10^{-5}$ )
<b>60</b>	$1.5 \times 10^{-4}$ (+/- $9.30 \times 10^{-5}$ )	$5.79 \times 10^{-5}$ (+/- $3.92 \times 10^{-6}$ )	0.049 (+/- 0.025)	$5.81 \times 10^{-5}$ (+/- $3.97 \times 10^{-6}$ )
<b>90</b>	$4.69 \times 10^{-5}$ (+/- $2.10 \times 10^{-5}$ )	$4.43 \times 10^{-5}$ (+/- $1.29 \times 10^{-5}$ )	0.0049 (+/- 0.0018)	$4.43 \times 10^{-5}$ (+/- $1.29 \times 10^{-6}$ )

Note that on day 90, there is still a standard deviation of 11 km for the DSN absolute position error, which is a lot of uncertainty in the answer. While the Relative cases have a mean absolute position of 112 m, which is not too far from 188.92 m for DSN/VLBI, the standard deviation on the relative cases is far less (= +/- 4.79 m). Thus far, the

Absolute data is shown to improve with the presence of Relative data. Figure 6.13 shows that the relative error also improves with the presence of Relative data.

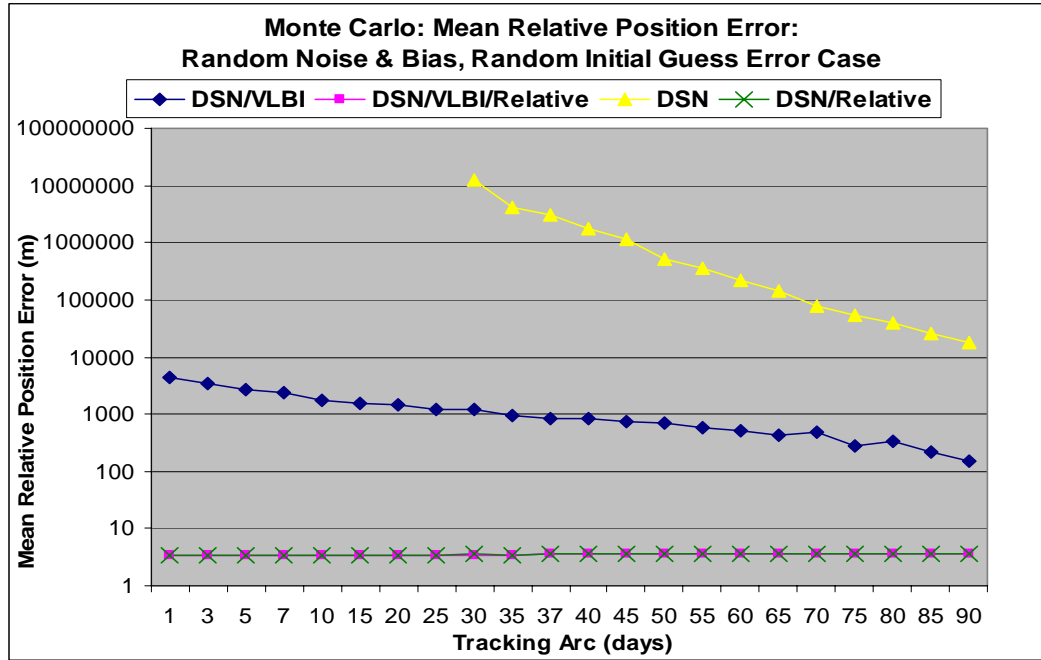


Figure 6.13: MC Simulation, NB, Mean Relative Position Error

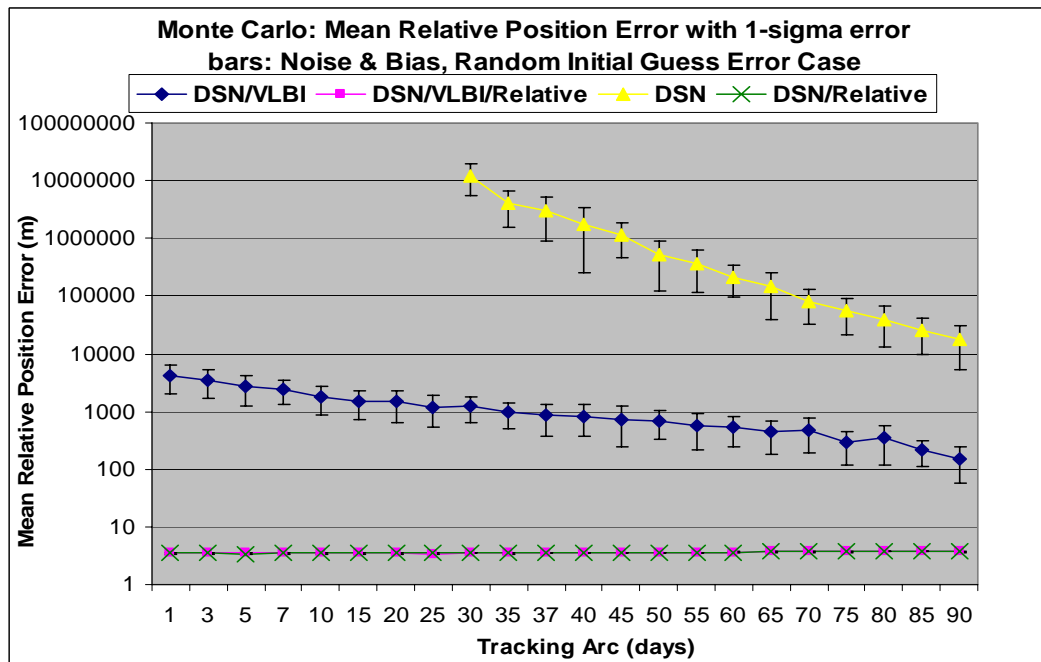
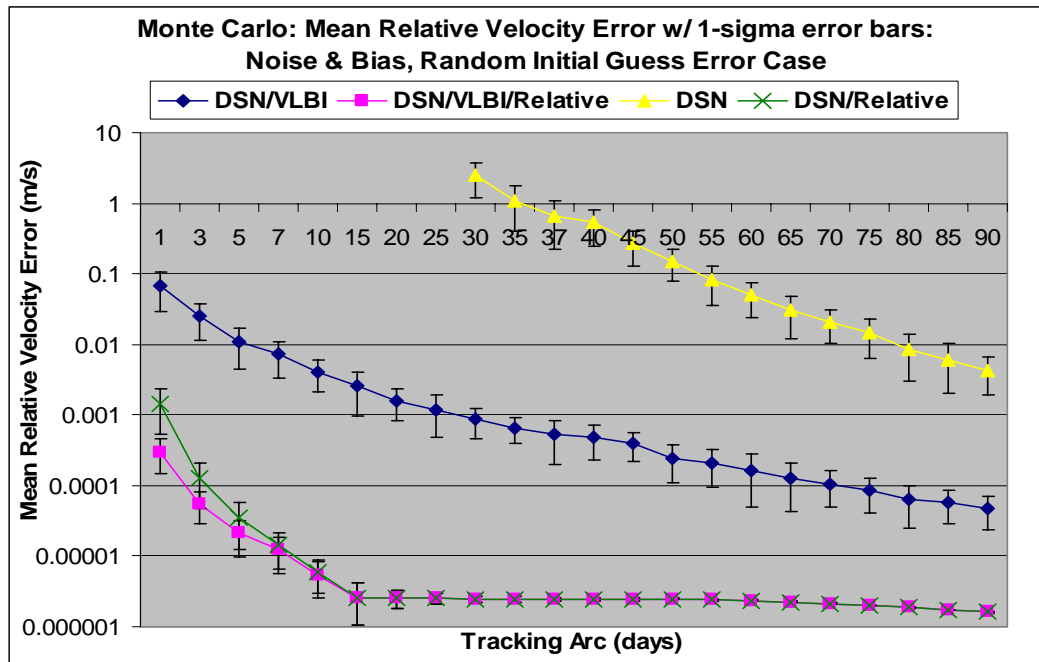


Figure 6.14: MC Simulation, NB, Mean Relative Position Error with sigmas

Figure 6.14 shows that there is an unperceivable standard deviation on the relative cases, whereas, on the non-Relative cases this deviation goes from +/- a few km to a few hundred meters. Figure 6.15 is an error bar plot of the mean relative velocity error with indicating very small standard deviations on the Relative cases after day 20.



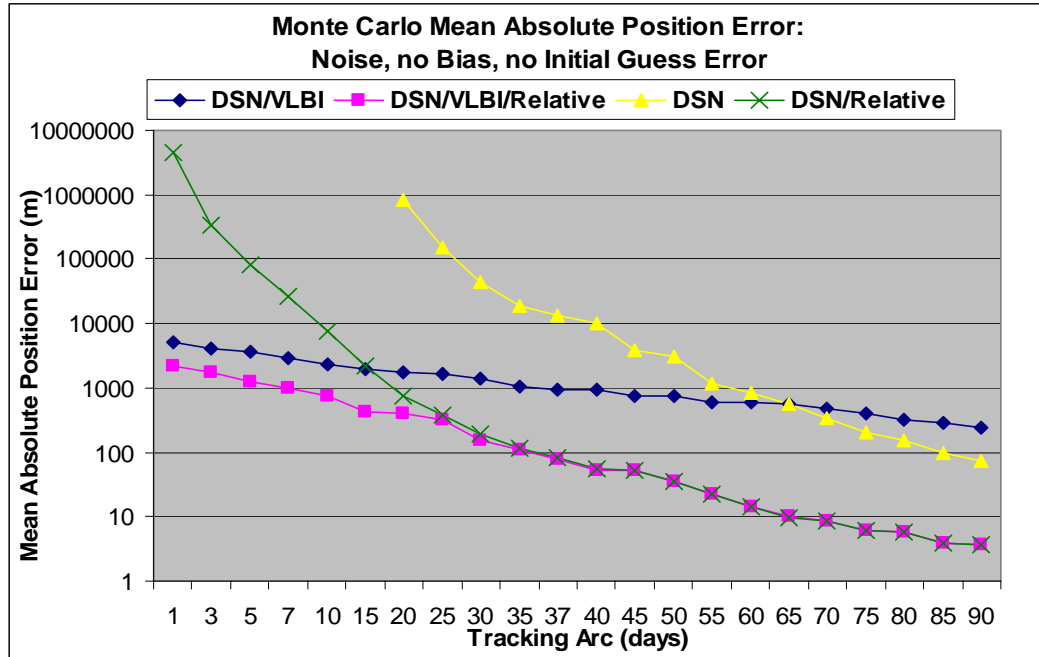
**Figure 6.15:** MC Simulation, NB, Mean Relative Velocity Error with sigmas

The raw data for the mean relative position and velocity with the (+/-) 1-sigma standard deviations are summarized in Table 6.4 below. Of note, DSN has standard deviations of the same magnitude as the measurement itself, with the sigma-error on the order of km by day 90. Currently, there is an inter-spacecraft accuracy requirement of about 300 m [8], which may be tighten to 20 m or less in the future [3]. These findings suggest that it would be impossible to meet this requirement without Relative data cases (D/V/R and D/R). DSN/VLBI could only achieve a 300 meter relative position error accuracy after a ~83 day tracking arc.

**Table 6.4: Summary of Mean Relative Position & Velocity with Standard Deviations**

<b>Mean Relative Position Error (m) (+/- 1-<math>\sigma</math>)</b>				
<b>days</b>	<b>DSN/VLBI</b>	<b>DSN/VLBI/Relative</b>	<b>DSN</b>	<b>DSN/Relative</b>
<b>1</b>	4278.9 (+/- 2224)	3.468 (+/-0.113)	-----	3.469 (+/- 0.111)
<b>20</b>	1475 (+/- 820.29)	3.474 (+/-0.063)	-----	3.485 (+/- 0.081)
<b>30</b>	1209.9 (+/- 580.48)	3.497 (+/- 0.051)	$1.24 \times 10^7$ (+/- $7.06 \times 10^6$ )	3.510 (+/- 0.049)
<b>60</b>	527.01 (+/- 286.1)	3.629 (+/- 0.054)	$2.15 \times 10^5$ (+/- $1.18 \times 10^5$ )	3.631 (+/- 0.054)
<b>90</b>	148.49 (+/-89.641)	3.698 (+/- 0.049)	17996 (+/- 12623)	3.698 (+/- 0.049)
<b>Mean Relative Velocity Error (m/s) (+/- 1-<math>\sigma</math>)</b>				
<b>days</b>	<b>DSN/VLBI</b>	<b>DSN/VLBI/Relative</b>	<b>DSN</b>	<b>DSN/Relative</b>
<b>1</b>	0.06618 (+/- 0.0376)	0.000299 (+/- $1.53 \times 10^{-4}$ )	-----	0.001447 (+/- $9.15 \times 10^{-4}$ )
<b>20</b>	0.001611 (+/- $7.62 \times 10^{-4}$ )	$2.55 \times 10^{-6}$ (+/- $7.21 \times 10^{-7}$ )	-----	$2.54 \times 10^{-6}$ (+/- $7.18 \times 10^{-7}$ )
<b>30</b>	$8.54 \times 10^{-4}$ (+/- $3.92 \times 10^{-4}$ )	$2.46 \times 10^{-6}$ (+/- $2.45 \times 10^{-7}$ )	2.47 (+/- 1.304)	$2.46 \times 10^{-6}$ (+/- $2.46 \times 10^{-7}$ )
<b>60</b>	$1.63 \times 10^{-4}$ (+/- $1.13 \times 10^{-4}$ )	$2.33 \times 10^{-6}$ (+/- $4.97 \times 10^{-8}$ )	$5.00 \times 10^{-2}$ (+/- 0.026)	$2.33 \times 10^{-6}$ (+/- $4.97 \times 10^{-8}$ )
<b>90</b>	$4.79 \times 10^{-5}$ (+/- $2.39 \times 10^{-5}$ )	$1.60 \times 10^{-6}$ (+/- $2.32 \times 10^{-8}$ )	$4.19 \times 10^{-3}$ (+/- 0.002)	$1.60 \times 10^{-6}$ (+/- $2.31 \times 10^{-8}$ )

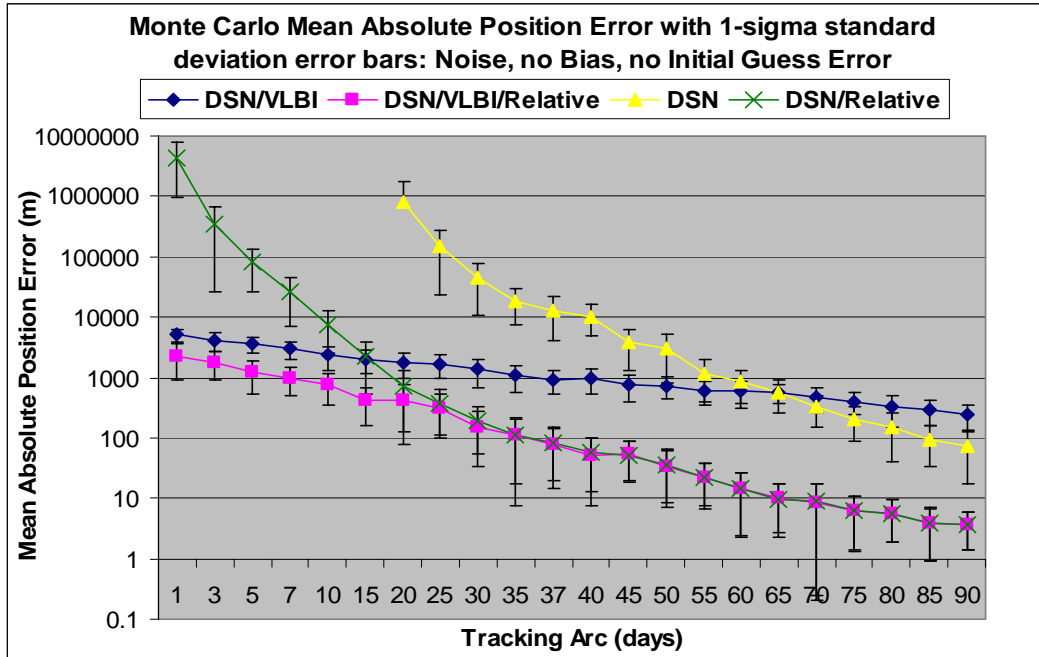
Without biases, (denoted as "noB") and initial guess error there is a dramatic improvement in the already outperforming DSN/VLBI/Relative and DSN/Relative cases; there is an improvement in DSN as well. Just random noise is applied on the simulation in this section. Figure 6.16 is the absolute position error and shows that DSN gets as good as DSN/VLBI by day 65. The Relative case errors continue to decrease rather than remain constant as with biases.



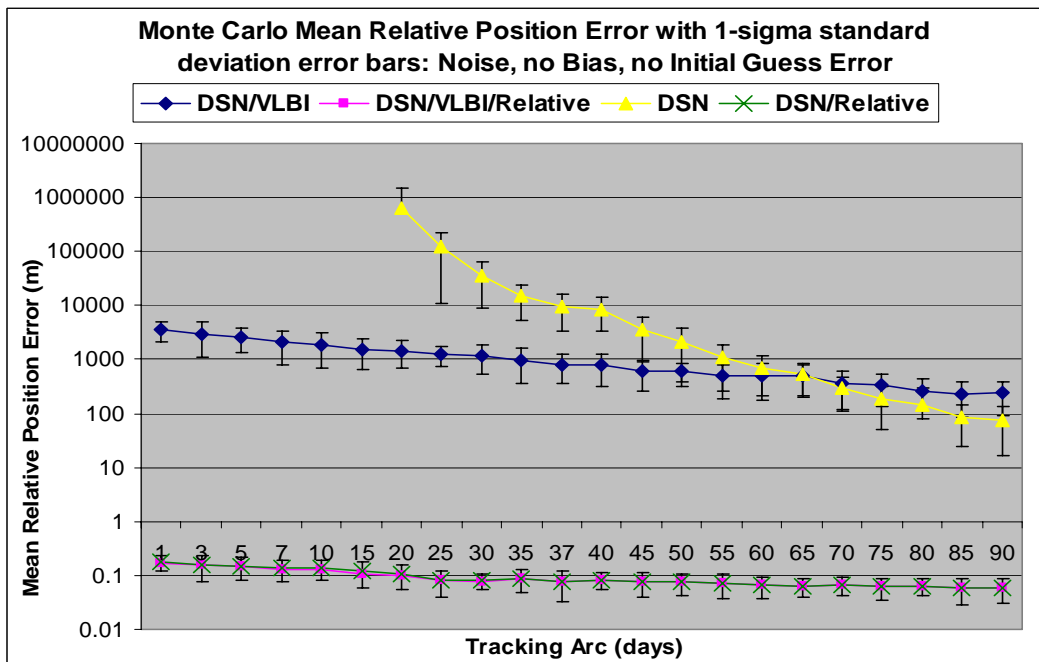
**Figure 6.16:** MC Simulation, noB, Mean Absolute Position Error

Figure 6.17 shows a lot more correlation between the error bars of DSN and DSN/VLBI around day 45. DSN/VLBI and DSN/Relative overlies from days 10-20 and then DSN/Relative and DSN/VLBI/Relative overlap for the rest of the time. This is a strong indicator that VLBI is no longer effective once these two curves overlap. It can also be inferred that VLBI no longer becomes effective once DSN decreases below DSN/VLBI.

Figure 6.18 shows the mean relative position error with standard deviations. VLBI appears to lose effectiveness by day 65 when the DSN curve becomes better than the DSN/VLBI curve. Figures 6.19 and 6.20 show the absolute and relative velocity with their standard deviations.



**Figure 6.17:** MC Simulation, noB, Mean Absolute Position Error with sigmas



**Figure 6.18:** MC Simulation, noB, Mean Relative Position Error with sigmas

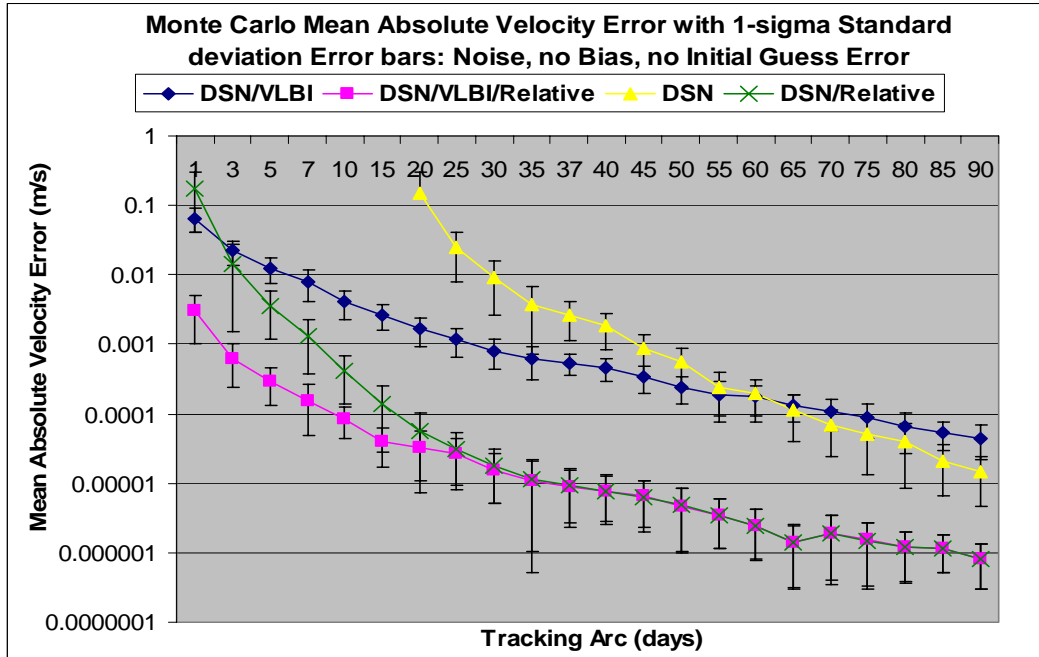


Figure 6.19: MC Simulation, noB, Mean Absolute Velocity Error with sigmas

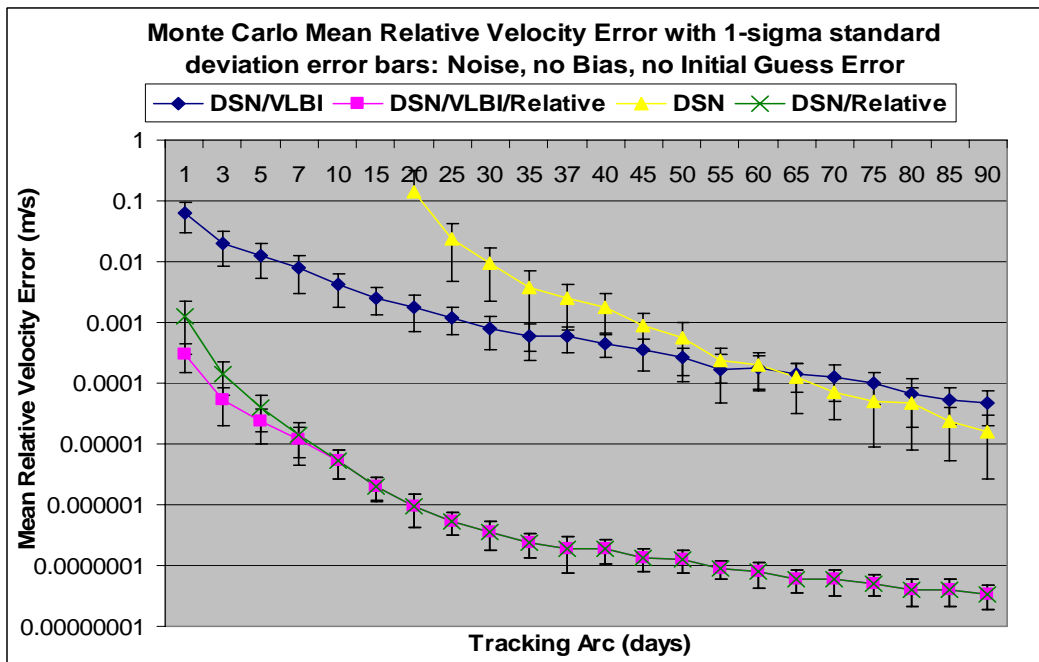


Figure 6.20: MC Simulation, noB, Mean Relative Velocity Error with sigmas

Table 6.5 is a summary of the mean absolute position and velocity for the Monte Carlo simulation with noise and no biases. Note that the Relative-data corrects to much larger than  $\sqrt{3}$  over the entire tracking arc for the Mean absolute position and the errors are



smaller than those seen in Table 6.3. Table 6.6 is a summary of the mean relative position and velocity for the Monte Carlo simulation presented in this section.

**Table 6.5:** Summary of Mean Absolute Position & Velocity with Standard Deviations

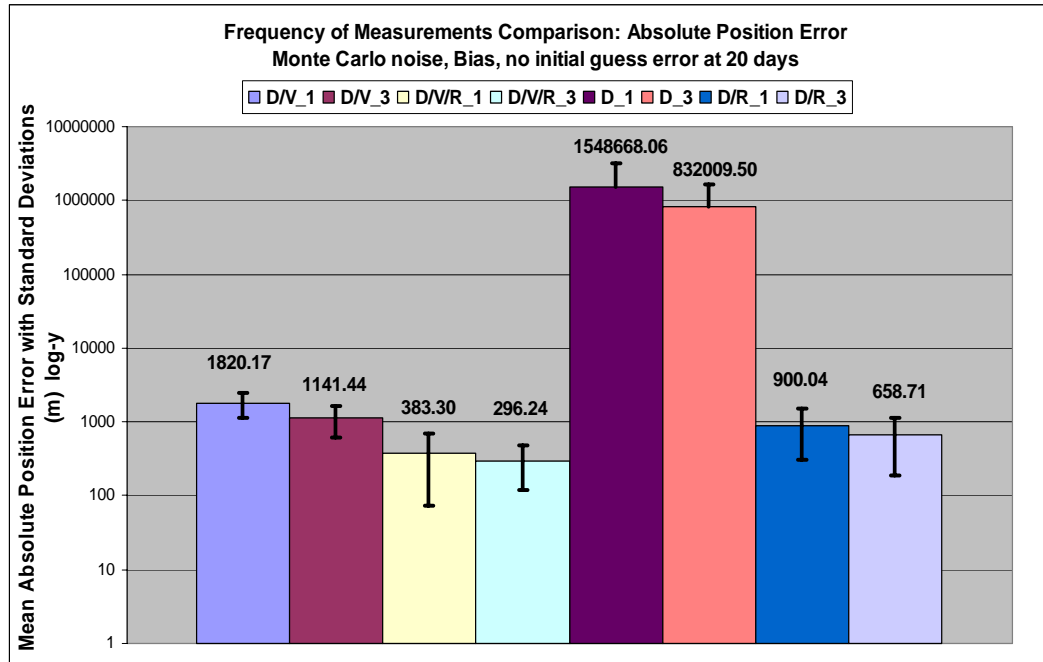
<b>Mean Absolute Position Errors (m) (+/- 1-<math>\sigma</math>)</b>				
<b>Days</b>	<b>DSN/VLBI</b>	<b>DSN/VLBI/Relative</b>	<b>DSN</b>	<b>DSN/Relative</b>
<b>1</b>	5076.3 (+/- 1155.9)	2246 (+/- 1349.7)	-----	4414000 (+/- 3467000)
<b>20</b>	1717 (+/- 773.1)	412.63 (+/- 336.21)	$8.14 \times 10^5$ (+/- $9.85 \times 10^5$ )	732.7 (+/-608.37)
<b>30</b>	1356.5 (+/- 664.78)	154.17 (+/- 120.81)	44549 (+/- 33813)	193.47 (+/- 138.71)
<b>60</b>	590.8 (+/- 287.68)	14.471 (+/- 12.021)	844.9 (+/- 471.73)	14.322 (+/- 12.103)
<b>90</b>	238.61 (+/- 110.54)	3.5992 (+/- 2.1706)	75.061 (+/- 57.118)	3.58 (+/- 2.1834)
<b>Mean Absolute Velocity Errors (m/s) (+/- 1-<math>\sigma</math>)</b>				
<b>Days</b>	<b>DSN/VLBI</b>	<b>DSN/VLBI/Relative</b>	<b>DSN</b>	<b>DSN/Relative</b>
<b>1</b>	0.065252 (+/- 0.024914)	0.003078 (+/- 0.002031)	-----	0.17008 (+/- 0.12875)
<b>20</b>	0.001674 (+/- 0.000748)	$3.24 \times 10^{-5}$ (+/- $2.52 \times 10^{-5}$ )	0.1521 (+/-0.15522)	$5.66 \times 10^{-5}$ (+/- $4.59 \times 10^{-5}$ )
<b>30</b>	0.000812 (+/- 0.000364)	$1.58 \times 10^{-5}$ (+/- $1.06 \times 10^{-5}$ )	0.009395 (+/- 0.006804)	$1.84 \times 10^{-5}$ (+/- $1.33 \times 10^{-5}$ )
<b>60</b>	0.000175 (+/- $8.01 \times 10^{-5}$ )	$2.48 \times 10^{-6}$ (+/- $1.69 \times 10^{-6}$ )	0.000195 (+/- 0.000117)	$2.48 \times 10^{-6}$ (+/- $1.69 \times 10^{-6}$ )
<b>90</b>	$4.49 \times 10^{-5}$ (+/- $2.26 \times 10^{-5}$ )	8.25E-07 (+/- 5.20E-07)	$1.47 \times 10^{-5}$ (+/- $1.00 \times 10^{-5}$ )	$8.23 \times 10^{-7}$ (+/- $5.23 \times 10^{-7}$ )

**Table 6.6: Summary of Mean Relative Position & Velocity with Standard Deviations**

<b>Mean Relative Position Errors (m) (+/- 1-<math>\sigma</math>)</b>				
<b>days</b>	<b>DSN/VLBI</b>	<b>DSN/VLBI/Relative</b>	<b>DSN</b>	<b>DSN/Relative</b>
<b>1</b>	3595.6 (+/- 1430.6)	0.16899 (+/- 0.0584)	-----	0.17643 (+/- 0.055822)
<b>20</b>	1449.1 (+/- 767.26)	0.096109 (+/-0.043723)	$6.31 \times 10^5$ (+/-8.54 x 10 <sup>5</sup> )	0.10383 (+/- 0.050238)
<b>30</b>	1197.8 (+/- 659.26)	0.078791 (+0.027683)	36700 (+/-27589)	0.081075 (+/- 0.02763)
<b>60</b>	503.98 (+/- 327.69)	0.065993 (+/- 0.028862)	699.78 (+/- 487.49)	0.065739 (+/- 0.029286)
<b>90</b>	239.51 (+/- 150.56)	0.057915 (+/- 0.026587)	76.138 (+/- 59.561)	0.057976 (+/- 0.02658)
<b>Mean Relative Velocity Errors (m/s) (+/- 1-<math>\sigma</math>)</b>				
<b>days</b>	<b>DSN/VLBI</b>	<b>DSN/VLBI/Relative</b>	<b>DSN</b>	<b>DSN/Relative</b>
<b>1</b>	0.062469 (+/- 0.032609)	0.00029989 (+/- 0.00014873)	-----	0.001255 (+/- 0.000961)
<b>20</b>	0.001732 (+/- 0.00102)	$9.51 \times 10^{-7}$ (+/-5.21 x 10 <sup>-7</sup> )	0.14373 (+/- 0.16548)	$9.47 \times 10^{-7}$ (+/-5.18 x 10 <sup>-7</sup> )
<b>30</b>	0.000816 (+/- 0.000461)	$3.49 \times 10^{-7}$ (+/-1.67 x 10 <sup>-7</sup> )	0.009325 (+/- 0.007134)	$3.49 \times 10^{-7}$ (+/- 1.67 x 10 <sup>-7</sup> )
<b>60</b>	0.00018 (+/- 9.97 x 10 <sup>-5</sup> )	$7.80 \times 10^{-8}$ (+/- 3.69 x 10 <sup>-8</sup> )	0.000195 (+/- 0.000121)	$7.81 \times 10^{-8}$ (+/- 3.69 x 10 <sup>-8</sup> )
<b>90</b>	$4.81 \times 10^{-5}$ (+/- 2.82 x 10 <sup>-5</sup> )	$3.32 \times 10^{-8}$ (+/- 1.47 x 10 <sup>-8</sup> )	$1.59 \times 10^{-5}$ (+/- 1.33 x 10 <sup>-5</sup> )	$3.32 \times 10^{-8}$ (+/- 1.47 x 10 <sup>-8</sup> )

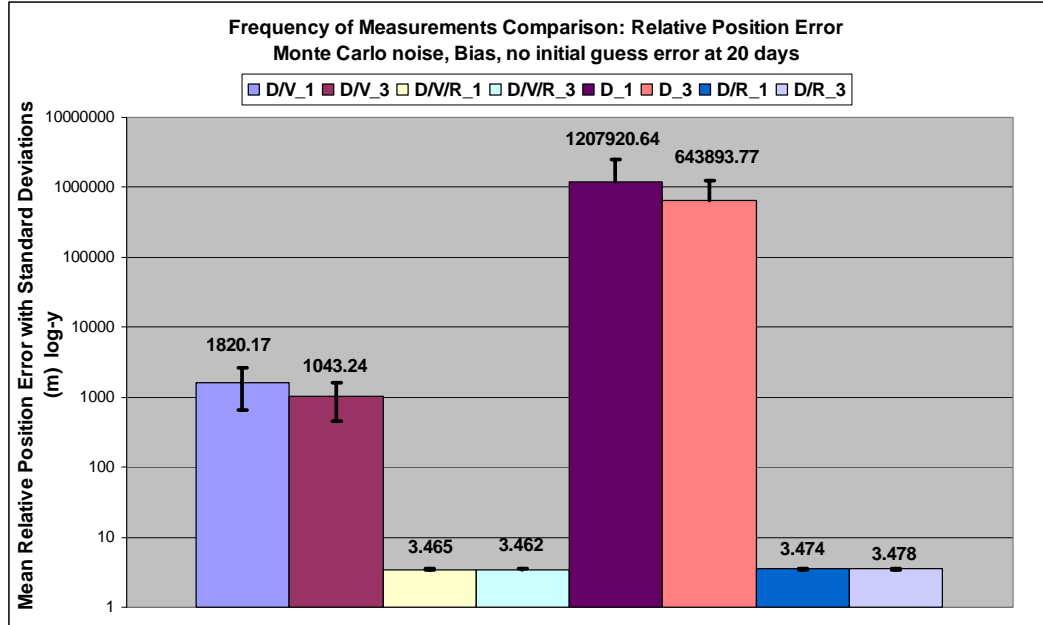
The next two figures predict the effect that increasing the number of measurements per day from 1 to 3 would have on each case. A Monte Carlo dataset was collected for 1 and 3 measurements per day at day 20 with noise and bias and no initial guess errors to determine if more frequent measurements better the OD. Figure 6.21 shows the absolute position error, indicating an error reduction by hundreds of meters for most cases. The least improvement is seen on the DSN/VLBI/Relative case. The greatest improvement is on DSN. The standard deviation bars are also plotted on the columns and the standard

deviation remains comparable for all the cases except DSN/VLBI/Relative in which it decreases for three measurements per day.



**Figure 6.21:** Frequency of Measurements, Absolute Position Error

In Figure 6.22, increasing the measurements per day for mean relative position error has very little effect on the Relative cases, as both bars are nearly equal in value for DSN/VLBI/Relative and DSN/Relative. There is also insignificant standard deviation in both columns. However, for the non-Relative cases, there is a decrease in error with the increased measurement availability.



**Figure 6.22:** Frequency of Measurements, Relative Position Error

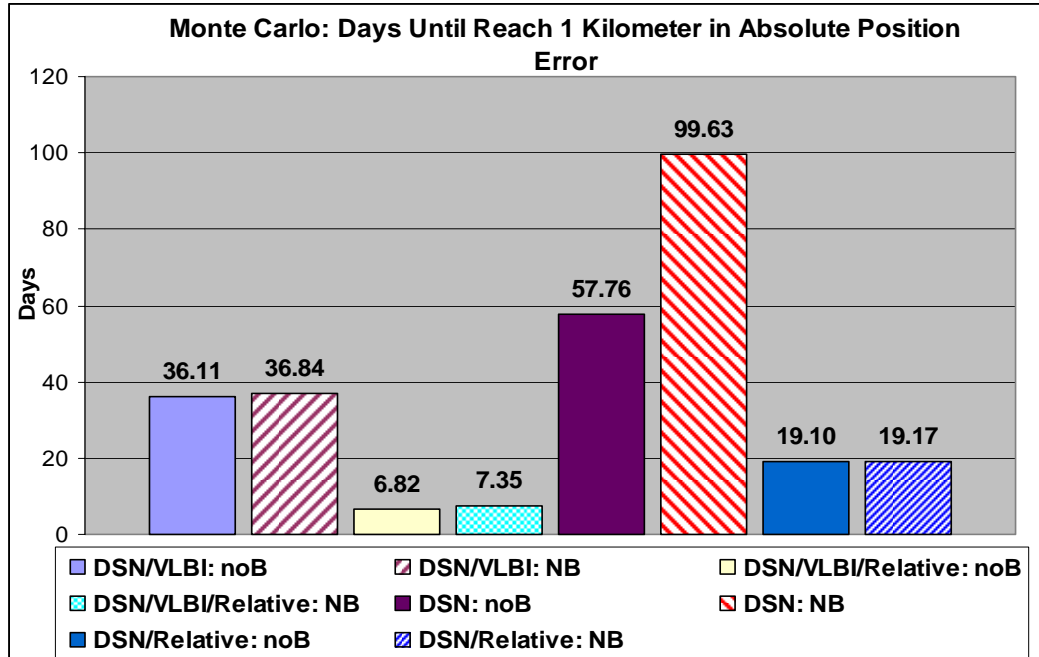
#### 6.4. Discussion of Results

Summary comparisons of the Monte Carlo runs are exhibited in the following bar graphs.

Additional comments will be made concerning various requirement and constraints.

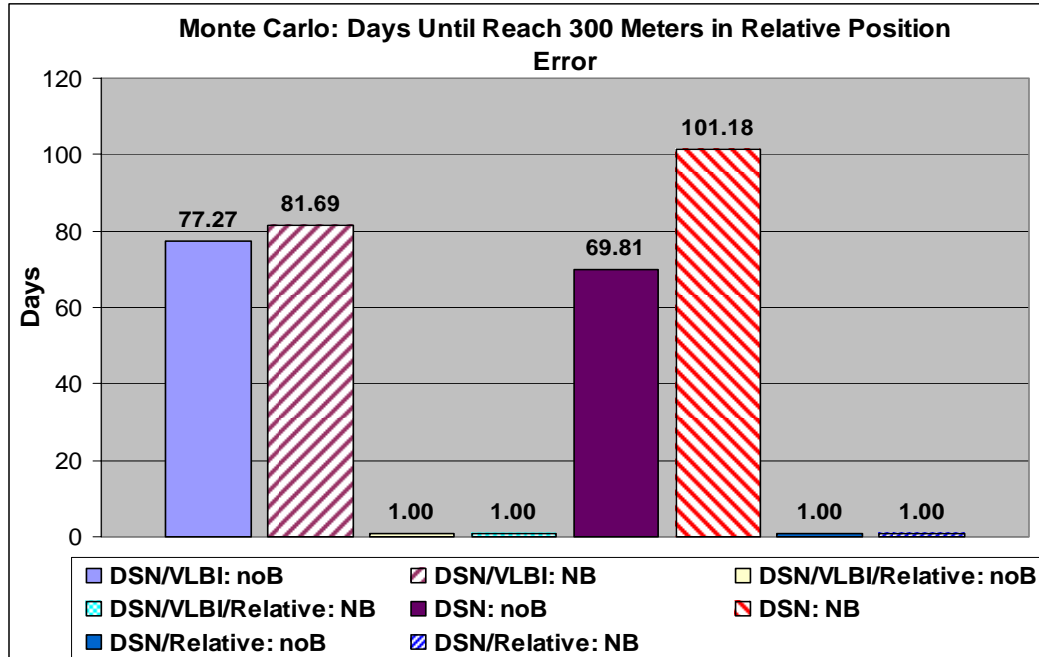
Figure 6.23 illustrates how many days it takes each case to get to an absolute position accuracy of 1 kilometer for both Monte Carlo runs. Linear interpolation was used to determine exactly when each case would reach 1 kilometer. Without biases ("noB"), all cases reduce to 1 kilometer error sooner than with biases ("NB"). There is only a slight improvement for DSN/VLBI and the Relative cases. DSN however, improves greatly, though this can be attributed to no initial guess error since DSN was shown to handle biases quite well. Note that for the Monte Carlo Noise/Bias Run, the DSN-only case never achieved an error as low as 1 kilometer. In order to obtain the projected 99.63 day

bar, linear interpolation was applied to best estimate when this would be achieved beyond the 90 day commissioning period.



**Figure 6.23:** Summary Chart of Days Until Absolute Position Error Equals 1 Kilometer

For relative position error, only the Relative cases (DSN/VLBI/Relative and DSN/Relative) are capable of achieving an accuracy of < 20 meters in the 90 day time period. Figure 6.24 illustrates how many days it takes to obtain a relative position accuracy of the less stringent 300 meters for both Monte Carlo runs. Again linear interpolation was used and some values are projected beyond the 90 day period. Within 1 day, both DSN/VLBI/Relative and DSN/Relative are less than 300 meters.



**Figure 6.24:** Summary Chart of Days Until Relative Position Error Equals 300 meters

A current requirement is that the LISA spacecraft radius, or semi-major axis should be between 0.9 AU ( $= 1.346 \times 10^{11}$  m) and 1.1 AU ( $= 1.646 \times 10^{11}$  m) [8]. This constraint is met as each LISA for all cases stays within  $2.83 \times 10^8$  m of the nominal semi-major axis. There is currently a more stringent requirement on the period, such that each LISA should be within 38 second of nominal heliocentric period before beginning drag free science operations [8]. Since the period is related to the semi-major axis (i.e.  $P =$

$$2\pi \sqrt{\frac{a^3}{\mu}}$$

, the semi-major axis must be controlled to < 60 kilometers of the desired.

The prior stated  $2.83 \times 10^8$  m difference from the nominal semi-major axis translates to a 1.04 day maximum period difference. Since LISA's orbital period is one year, this period difference is merely 0.284% of the nominal period. Curiously, the earlier stated constraint of the semi-major axis being between 0.9 AU and 1.1 AU translates into a much larger period offset of 53-56 days of the nominal, or a 14.6%-15.4% error. This

simulation achieved semi-major axis within the loose constraint, while the periods were different by up to one day. Future work must investigate closing the gap on the tight period requirement through the use of  $\Delta V$  maneuvers prior to dropping the propulsion modules.

Parameter uncertainties decrease with increasing time as is evidenced by Figure 6.25, which shows the parameter uncertainty of the DSN/VLBI/Relative case for all three LISA spacecraft and all three position components. The Z-component, or the cross-track axis of the satellite *RSW* system for all three spacecraft, remains the most uncertain (the top three lines of the Figure 6.25) in low declination orbits. The Y- or along-track components are the middle three lines, while the X- or radial components are the bottom three lines. A longer tracking arc ensures more certainty in all components.

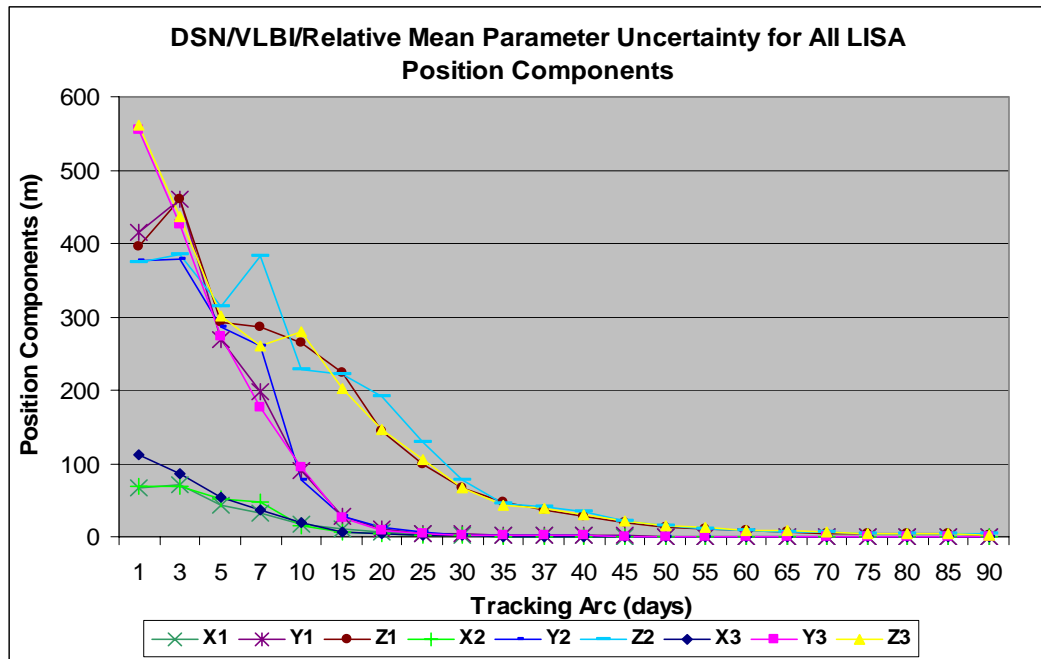


Figure 6.25: MC (NB) Parameter Uncertainty of LISA 1 position-components

The parameter uncertainties ( $PU$ ) and the mean estimated state ( $mean(\hat{X})$ ) at day = 7 for the Monte Carlo (Noise, Bias) case are listed in Table 6.7.

**Table 6.7:** Parameter Uncertainties ( $PU$ ) for DSN/VLBI/Relative Case at day 7

	LISA 1		LISA 2		LISA 3	
	$mean(\hat{X}_1)$	+/- $PU 1$	$mean(\hat{X}_2)$	+/- $PU 2$	$mean(\hat{X}_3)$	+/- $PU 3$
$X$ (m)	5.440	32.411	$2.488 \times 10^9$	47.786	$-2.488 \times 10^9$	35.773
$Y$ (m)	$1.481 \times 10^{11}$	198.210	$1.503 \times 10^{11}$	260.680	$1.503 \times 10^{11}$	177.120
$Z$ (m)	$2.476 \times 10^9$	286.950	$-1.292 \times 10^9$	381.750	$-1.292 \times 10^9$	261.200
$\dot{X}$ (m/s)	$-3.007 \times 10^4$	$3.592 \times 10^{-5}$	$-2.964 \times 10^4$	$4.701 \times 10^{-5}$	$-2.964 \times 10^4$	$3.327 \times 10^{-5}$
$\dot{Y}$ (m/s)	$3.333 \times 10^{-6}$	$2.194 \times 10^{-5}$	$2.476 \times 10^2$	$3.170 \times 10^{-5}$	$-2.476 \times 10^2$	$2.269 \times 10^{-5}$
$\dot{Z}$ (m/s)	$3.624 \times 10^{-6}$	$3.965 \times 10^{-5}$	$4.269 \times 10^2$	$5.454 \times 10^{-5}$	$-4.269 \times 10^2$	$3.851 \times 10^{-5}$



## Chapter 7: Conclusion

This thesis has studied the Orbit Determination (OD) problem for the proposed Laser Interferometer Space Antenna (LISA) mission in which three satellites in heliocentric orbit must maintain a triangular formation. This work investigated the impact of augmenting traditional Earth-based DSN and VLBI OD data with inter-spacecraft range and range rate (also termed "Relative" data) measurements available for the LISA mission. A weighted batch least squares algorithm was used to estimate spacecraft orbital parameters from four test combinations of measurement types: 1) DSN and VLBI, 2) DSN, VLBI and Relative, 3) DSN only, and 4) DSN and Relative. An extensive set of simulation-based experiments were conducted to study OD accuracy statistics. The two main modes of analysis to compare the test cases were the Covariance and Monte Carlo methods.

Very Long Baseline Interferometry (VLBI) has been shown to improve initial data quality, particularly when DSN is the only other measurement available. However, when applied with a noise of 50 nrad, VLBI becomes obsolete after a certain amount of time since long arcs of range and Doppler data can provide angular position accuracy of 40 nrad or better [41]. Though VLBI or some type of angle data is helpful initially, LISA passes through zero declination and has periods of poor observability over the course of its orbit. LISA needs a longer Doppler arc, or tracking must be supplemented with an

alternative technique for measuring angles to become observable. VLBI gives orientation to the range measurement when Doppler is not good initially.

More measurements per day were found to improve the absolute position accuracy. Given LISA's distance from the Earth, its orbital period of one year, and LISA unique geometry with respect to Earth, a sufficient tracking arc length tells more about the spacecraft than denser data. Future work would be needed to investigate the concept of more frequent data further. Confidence in the answer also improves with longer tracking arcs. Parameter uncertainties and standard deviations in the covariance matrix diagonals associated with each element of position and velocity as well as standard deviations associated with measured errors (i.e. absolute position and relative error) all decrease with an increasing tracking arc.

The most major finding of this work is that Relative data is worth including to improve not only the relative knowledge of the spacecraft, but also for absolute position and velocity OD as well. There was a greater than expected  $\sqrt{3}$  improvement seen for absolute OD accuracy for the cases with Relative data. VLBI is also worth incorporating as it does a better job initially of tracking LISA in its low declination orbit. After about 20-35 days, VLBI can be reallocated to other assets, and the tracking can rely on DSN and Relative data without loss of accuracy. DSN-only needs a 20 day tracking arc with a good initial guess; if the initial guess is off by more than 1 km, a 30-35 tracking arc is needed. The other cases could correct for initial guess errors up to 10 km, though no check was done on the upper limit. Noises in the measurements are filtered out with a

sufficiently long tracking arc, but the biases are constant. The biases affected the Relative data more than the non-Relative, which points to a need for the calibration and the subtraction of known biases for better results.

The current inter-spacecraft accuracy requirement of 300 m may be tightened in the future to an accuracy of 20 m or less. DSN/VLBI can barely make that requirement until 85 days of tracking. If the requirement were more stringent, then only the Relative are capable of achieving that level of accuracy. The semi-major axis constraint is met by all cases, but the more stringent 38 second period error constraint prior to propulsion module drop-off is still harder to achieve. This simulation estimates error to be within one day of the nominal period.

The overall best case from this study is DSN/VLBI/Relative, i.e. using all available measurements, including the very crucial inter-spacecraft range and range rate. After 7 days of tracking LISA, DSN/VLBI/Relative is already achieves below one kilometer absolute OD accuracy and less than 3.5 meters in relative OD error.

## **7.1. Future Work**

Future work would include simulating a more realistic model of LISA. The perturbations of planets would need to be added for a more complete and accurate force model of LISA in its space environment. The nominal orbit of Dhurandhar [18], who did substantial

work with the Clohessy-Wiltshire equations on the unique +/- 60 degree tilt of the LISA orbit in a perturbed environment, would most likely be used. The rotation of Earth and the tracking station location on Earth's surface would also have to be modeled. In particular, case studies are required that use the location of specific tracking stations, their visibility of the spacecraft, and sensors that will actually be used. Even more specific simulations would include trade studies to determine which sensors are best, the choice of a Deep Space Station antenna(s) to be used, VLBI antennas, and the selection of the quasar to be used as a reference for angular measurements.

More investigation over the ideal amount of measurements per day could be conducted. It was found that for this orbit increasing the number of measurements to three per day does improve the absolute OD accuracy and to a lesser degree the relative OD at 20 days. However, could the accuracy stay the same for skipping days? For example, consider a three-day rotation where LISA 1 is tracked once per day on day 1, LISA 2 on day 2, etc. The data combination of DSN/VLBI/Relative was found to be the best; however, angle data becomes ineffective after 20-35 days. Further combinations that switch data types on and off when they are or are not effective could be investigated to see if the results are as accurate as the DSN/VLBI/Relative case. For example, using DSN/VLBI/Relative at the beginning and then turning off VLBI to just rely on DSN/Relative when the angles are no longer effective would be an intuitive choice.

Cost related studies are also a spin-off of this thesis. The practical implication of using different measurements as a function of cost can be studied, a total of six measurement

types (range, range rate, right ascension, declination, inter-spacecraft range and range rate) were used here. The potential cost savings of turning VLBI off when no longer effective could be investigated, since this data type requires two antennas on Earth to obtain its measurements. The possibility of using less DSN and relying more on VLBI and inter-spacecraft data is also of interest, since the current configuration of DSN is struggling to support the deep space communications demands of the coming decades. However, VLBI could potentially pose an advantage over DSN since it uses two smaller antennas rather than a larger single antenna, which saves in operating and production costs [42].

Further work with weighting the data, using less or more noise and different biases also could help understand other nuances of this problem, the measurements, and how the error sources interact. Biases were found to have a greater effect on the relative data, therefore, estimating biases and working to better calibrate these biases in the instruments could improve OD.

The 90 day time period of the close to nominal commissioning period was used as the thesis benchmark. Future work could also include looking at other specific OD events around the three month time period such as when the propulsion module drop-off and will "kick" the spacecraft, and how long it will take to return back to the nominal orbit after that disturbance. The ideal tracking arcs should be determined before and after propulsion module drop-off. Lastly, the use of Relative i.e. inter-spacecraft data types could be very beneficial to future constellation missions as a method to get increased OD

accuracy, which is an improvement that cannot be done by Earth-based tracking alone.

Better OD accuracy and navigation means better science and mission success.

## Appendices

### Appendix A: Supporting Calculations

Newton's Law of Gravitation:

$$F_1 = m_1 \ddot{\bar{R}}_1 = \frac{-Gm_1 m_2}{|\bar{R}_1 - \bar{R}_2|^3} (\bar{R}_1 - \bar{R}_2) = \frac{-Gm_1 m_2}{r^2} = \frac{-\mu}{r^2}$$

where,  $\mu = GM_1$

and the mass,  $m_2$  of any LISA spacecraft is negligible compared to the larger mass (i.e. Earth, Jupiter and the Sun in this case). The laws of sines and cosines were used to find the distances between the spacecraft and the planetary body.

Law of sines:  $\frac{a}{\sin(\alpha)} = \frac{b}{\sin(\beta)} = \frac{c}{\sin(\gamma)}$

Law of cosines:  $a^2 = b^2 + c^2 - 2*b*c*\cos(\alpha)$

- Force of Sun on LISA:  $F = \frac{-\mu_S}{r_{SL}^2} = \frac{-1.32712428 \times 10^{11} \text{ km}^3 / \text{s}^2}{(149,597,870 \text{ km})^2} \approx -5.93 \times 10^{-6} \frac{\text{km}}{\text{s}^2}$

where, the distance of LISA to the Sun is:  $r_{SL} = 1$  Astronomical Unit =  
1 AU = 149,597,870 km

- Force of Earth on LISA:

$$F = \frac{-\mu_E}{r_{EL}^2} = \frac{-3.986004415 \times 10^5 \text{ km}^3 / \text{s}^2}{(49,068,043.6708 \text{ km})^2} \approx -1.6555 \times 10^{-10} \frac{\text{km}}{\text{s}^2}$$

where, the distance of the closest LISA from the Earth is:  $r_{EL} = 49,068,043.6708$  km

- Force of Jupiter on LISA:

$$F = \frac{-\mu_J}{r_{JL}^2} = \frac{-1.268 \times 10^8 \text{ km}^3 / \text{s}^2}{(740 \times 10^6 \text{ km} - 152484621.346 \text{ km})^2} \approx -3.6735 \times 10^{-10} \frac{\text{km}}{\text{s}^2}$$

where Jupiter at closest approach to Sun is 740 million km, and the furthest configuration of LISA from the Sun is 152,484,621.346 km.

Compared to the Sun, the force of Earth on LISA is 0.00279 % and the force of Jupiter on LISA is 0.006%.

## Appendix B: MATLAB CODES

### Table of Contents

<b>Orb2X2.m</b> .....	158
-This script transforms a set of Keplerian orbital elements to Cartesian format.	
<b>X2Orb.m</b> .....	160
-This script transforms a set of Cartesian elements to Keplerian orbital elements.	
<b>LISA_EOMderivs.m</b> .....	164
-Propagates the input state using the equations of motion.	
<b>NomOrb.m</b> .....	164
-Determines a nominal trajectory from Steve Hughes initial states. This script generates the ephemeris for one year by using the equations of motion.	
<b>LISA_STMderivs.m</b> .....	166
-Maps the state transition matrix to each time in the propagation	
<b>H_partials.m</b> .....	167
-This program presents the partials of the observation-state mapping matrix for each element for each state component.	
<b>RealObs.m</b> .....	171
-Calculates the Earth-based observations (range, range rate, right ascension and declination) at each measurement time, it also simulates the "observed" measurements by including noises and biases.	
<b>InterSCD.m</b> .....	174
-Calculates the inter-spacecraft observations (inter-spacecraft range and range rate) at each measurement time, it also simulates the "observed" measurements by including noises and biases.	
<b>LISA_WBLS.m</b> .....	176
-This program is the weighted batch least squares algorithm for each of the four cases.	
<b>MCRUNbp.m</b> .....	188
-This program calls LISA_WBLS.m and runs the Monte Carlo simulation and computes relevant statistics.	



## Appendix B: MATLAB CODES

*Orb2X2.m* (from Section 5.2.2.The Nominal Orbit)

```
function [R, V] = Orb2X2(Ephem, t, mu, type)
%*****
%
%INPUT: Ephem1 = [a e i Om w tau], the six classical orbital elements
%      [semi-major axis, eccentricity, inclination, longitude of the
%      ascending node, argument of periapsis, time of periapsis]
%
%      Ephem2 = [a e i Om w nu], the six classical orbital elements
%      [semi-major axis, eccentricity, inclination, longitude of the
%      ascending node, argument of periapsis, true anomaly]
%
%      Ephem3 = [a e i Om ~omega L]
%      ~omega = longitude of perhelion
%      L = mean longitude
%
%      Ephem4 = [a e i Om w M], the six classical orbital elements
%      [semi-major axis, eccentricity, inclination, longitude of the
%      ascending node, argument of periapsis, mean anomaly]
%
%      t = time of epoch associated with the particular ephemeris
%      mu = gravitational parameter
%      type = 1 = Program uses Ephem1 as input
%           = 2 = Program uses Ephem2
%           = 3 = Program uses Ephem3
%           = 4 = Program uses Ephem4
%
%OUTPUT : R - Position Vector
%        V - Velocity Vector
%
%CALLS : newton_meth.m, succ_sub_hyperbola.m
%
%Ephem2 = [149578965033 .01034342962 1.023240970 212.7394780
%90.76228189 136.5015274];[R, V] = Orb2X2(Ephem2,
%2.456262500000000e+006, 389600.4415, 2)
```

```
%
%mu =4*(pi^2);t=.010576712; [R, V] = Orb2X(Ephem,t, mu)
%REFERENCE: Equations from Battin(pp. 123-125), Tolson, Chobotov
%Ephem = [6828973.232519 .0090173388450585 28.474011884869
%35.911822759495 -44.55584705279 43.8860381032208]
%*****
%***INITIALIZATION*****
if nargin < 4
    type = 1 %assume type 1 ephemeris inputs
end

if nargin < 3
    mu = 3.9860044e5 %km^3/s^2, assume gravitational parameter of the
Earth
end

r2d = 180/pi;
d2r = pi/180;

%*****
a = Ephem(1); %semi-major axis (usu. m/s or km/s)

if a > 0
    disp('elliptic orbit')
elseif a < 0
    disp('hyperbolic orbit')
else %z = 1/a = 0
    disp('parabolic orbit')
end
e = Ephem(2); %eccentricity (unitless)
i = Ephem(3)*d2r; %inclination, from deg. to radians
Om = Ephem(4)*d2r; %Right Ascension of the Ascending Node, from deg.
%to radians
if type ~= 3 %thus type 1, 2, or 4
    w = Ephem(5)*d2r; %argument of periapsis, from deg. to radians
    w_bar = Om + w; %longitude of periapsis = argument of periapsis -
%longitude of the ascending node (radians)
```

```

elseif type == 3
    w_bar = Ephem(5)*d2r; %longitude of perihelion
    w = w_bar - Om; %argument of periapsis
end

if e == 0 %circular orbit, no periapsis
    w_bar = NaN; %undefined
end

if e < 1
    mean_mo = sqrt(mu/a^3); %mean motion
elseif e > 1
    mean_mo = sqrt((-1*mu)/a^3); %mean motion
else %e == 1
    p = a*(1-e^2); %semi-latus rectum
    mean_mo = sqrt(mu/p^3); %mean motion
end

p = a*(1-e^2); %semi-latus rectum

%***Solve Kepler's equation for the Eccentric anomalies for type 1 and 3
tol=1e-10;

if type == 1
    tau = Ephem(6); %This Ephemeris uses the time of perapsis passage as
    %the 6th element

    %Find Mean Anomaly
    ep = w_bar - (mean_mo*tau); %the mean longitude at epoch time t=0
    M = (mean_mo * t) + ep - w_bar;

else type == 3
    L = Ephem(6)*d2r;
    M = L - w_bar;
    tau = t - (M/mean_mo);

    if e < 1 %ellipse, solve Kepler's equation for E, Eccentric Anomaly

```

```

        %Newton's iteration method. Call to file "newton_meth.m"
        [E,its] = newton_meth(e,M,tol);
        E
        nu = 2*atan(sqrt((1+e)/(1-e))*tan(E/2)); %radians
    elseif e > 1 %hyperbola, solve Kepler's equation for a hyperbola for F
        %Successive Substitution method, call "Succ_sub_hyperbola.m"
        [F,its] = succ_sub_hyperbola;
        F
    else %e == 1 %parabola
        P = 0 %NO Eccentric Anomaly for Parabolic case
    end
end

if type == 4
    M = Ephem(6)*d2r
    tau = t - (M/mean_mo)
    if e < 1 %ellipse, solve Kepler's equation for E, Eccentric Anomaly
        %Newton's iteration method. Call to file "newton_meth.m"
        [E,its] = newton_meth(e,M,tol);
        E
        nu = 2*atan(sqrt((1+e)/(1-e))*tan(E/2)); %radians
    elseif e > 1 %hyperbola, solve Kepler's equation for a hyperbola for F
        %Successive Substitution method, call "Succ_sub_hyperbola.m"
        [F,its] = succ_sub_hyperbola;
        F
    else %e == 1 %parabola
        P = 0 %NO Eccentric Anomaly for Parabolic case
    end
end

%Solve for true anomaly
%if type == 1|3
% nu = 2*atan(sqrt((1+e)/(1-e))*tan(E/2)); %radians
%end

%***Find the ECCENTRIC and MEAN ANOMALIES for type 2

```

```

if type == 2
    nu = Ephem(6)*d2r; %Otherwise the 6th element is the true anomaly,
convert input into radians
    if e < 1
        cE = (e + cos(nu))/(1+(e*cos(nu)));
        sE = (sqrt(1-e^2)*sin(nu))/(1+e*cos(nu));
        E = atan2(sE,cE)
        M = E - e*sin(E);
    elseif e > 1
        cF = (e + cos(nu))/(1+(e*cos(nu)));
        sF = (sqrt(e^2-1)*sin(nu))/(1+e*cos(nu));
        F = atan2(sF,cF)
        M = e*sinh(F)-F;
    else %e==1
        P = 0;
        M = (1/2)*tan(nu/2) + (1/6)*(tan(nu/2)^3);
    end
end

%ML = w_bar + M; %mean longitude

%M = ML - L; %Mean anomaly

%***CALCULATE POSITION (R) & VELOCITY (V)*****
if e < 1
    rmag = a*(1-e*cos(E)); % magnitude position for an ellipse
elseif e > 1
    rmag = a*(1-e*cosh(F)); % magnitude position for a hyperbola
else % e==0
    rmag = (p/2)*(sec(nu/2)^2);% magnitude position for a parabola
end

theta = (w + nu); % argument of latitude = (w + nu) in radians

sN = sin(Om);
cN = cos(Om);
sT = sin(theta);

```

```

cT = cos(theta);
si = sin(i);
ci = cos(i);
sw = sin(w);
cw = cos(w);
% position vector
R = [rmag*((cT*cN)-(sT*sN*ci)); rmag*((cT*sN)+(sT*cN*ci));
rmag*(sT*si)];

h = sqrt(mu*p); % angular momentum scalar
mu_h = mu/h;
% velocity vector
V = [-mu_h*((cN*(sT + e*sw)) + (sN*(cT + e*cw)*ci));...
(-mu_h*((sN*(sT + e*sw))-(cN*(cT + e*cw)*ci))); (mu_h*((cT +
e*cw)*si))];

```

### *X2Orb.m* (from Section 5.2.2. The Nominal Orbit)

```

function [E] = X2Orb(R, V, t, mu)
% *****
%function [Ephem, Ephem_2, Misc, E] = X2Orb(R, V, t, mu)
%
%Function to obtain the six orbital elements plus some other orbital
%information from an inputted position and velocity vector.
%
%INPUTS : R = Position Vector (m)
%         V = Velocity Vector (m/sec)
%         t = epoch time of inputted R and V vector (sec)
%         mu = inputted gravitational parameter (m/sec^3)
%
%OUTPUTS : Can customize to read out different sets of elements.
%         Ephem = [a e i Om w tau], the six classical orbital elements
%         [semi-major axis, eccentricity, inclination, longitude of the
%         ascending node, argument of periapsis, time of periapsis] in

```

```

%      deg, time and length units
%      Ephem_2 = [f(True anomaly), M(Mean anomaly), ...
%      w_bar_true(true longitude of periapsis),...
%      w_bar(longitude of periapsis), ...
%      uo(argument of latitude at epoch), lo(true longitude at epoch)]
%      Misc = [h(angular momentum), p(semi-latus rectum), ra(apoapsis),
%rp(periapsis),...
%      Period], extra quantities to display
%      E = [a e i w Om M f Period]
%
%REFERENCE: Equations from (1)"Fundamental of Astrodynamics" by
%Bate, Mueller
% & White and Battin; (2)"Statistical Orbit Determination" by Byron
Tapley et.
% al.; (3)"Orbital Mechanics" Chobotov (4) Vallado, ELORB algorithm
% pp.146-147
%*****
%INITIALIZATION
if nargin < 4
    mu = (1.32712428e20); %m3/s2, Sun's gravitational parameter;
%1.327124e11*(1000^3); %m3/s2, Sun's gravitational parameter
    %mu = 3.9860044e5*(1000^3) %m^3/s^2, assume gravitational
%parameter of the Earth
    %error('Must pass in at least R,V, t and mu. Type "help X2Orb" for more
%information.');
```

```

end

rmag = norm(R); %magnitude of position vector
vmag = norm(V); %magnitude of velocity vector
```

```

r2d = 180/pi; %conversion from radians to degrees
d2r = pi/180; %conversion from degrees to radians
```

```

%ANGULAR MOMENTUM VECTOR, hvec
hvec = cross(R,V); %hvec is perpendicular to the plane of the orbit
```

```

hx = (R(2)*V(3)) - (V(2)*R(3)); %the x,y,z,xy components of the h vector
```

```

hy = (R(3)*V(1)) - (V(3)*R(1));
hz = (R(1)*V(2)) - (V(1)*R(2));
hxy = (hx^2 + hy^2)^(1/2);
```

```

h = norm(hvec); %the scalar of the h vector
```

```

%NODE VECTOR, nvec, pointing along the line of nodes in the direction
%of the ascending node
kvec=[0 0 1]; %k-vector
nvec = cross(kvec, hvec); %node vector
n = norm(nvec); %node
```

```

W = cross(R,V)./norm(cross(R,V));
W_check = hvec./norm(hvec);
```

```

%ECCENTRICITY VECTOR, evec, points from focus of orbit to the
%periapsis
```

```

evec = (1/mu)*(((vmag^2 - (mu/rmag))*R) - ((dot(R,V))*V));
e = norm(evec);
Vs = dot(V,V);
vs2 = vmag^2;
Energy = (Vs/2) - (mu/rmag); %Vis-Viva Equation
```

```

evec_check = cross(V,(hvec./mu)) - (R./rmag);
e_check = (1 + ((2*Energy*h^2)/mu^2))^(1/2);
```

```

%SEMI-LATUS RECTUM & SEMI-MAJOR AXIS
```

```

if e~= 0
a = -mu/(2*Energy); %semi-major axis
p = a*(1-e^2); %semi-latus rectum
```

```

rp = p/(1+e); %periapsis
ra = p/(1-e); %apoapsis
a_check = (rp + ra)/2;
```

```

a_check2 = 1/((2/rmag)-(vmag^2/mu));
```

```

else
```

```

a = inf; % semi-major axis equals infinity
p = h^2/mu;%semi-latus rectum
end

%PERIOD
Period = ((2*pi)*(a^(3/2)))/(sqrt(mu)); %seconds
Period_check = ((2*pi)*(a_check^(3/2)))/(sqrt(mu));

%INCLINATION, i
i = acos(dot(hvec,kvec)/h)*r2d; %all 3 methods should yield same value
i_check = acos(hz/h)*r2d;
i_check2 = acos(dot(W,kvec))*r2d;

if i <= 90;
    disp('direct, easterly orbit, inclination between 0 and 90 deg');
elseif i <= 180;
    disp('retrograde orbit, inclination is between 90 and 180 deg. ');
else %i>180, %inclination is always less than 180 deg.
    disp('error, i is always less than 180 deg. ');
end

%LONGITUDE OF THE ASCENDING NODE, Om
ivec = [1 0 0];
N = (cross(kvec,W))./norm(cross(kvec,W));
nvec(2);
if nvec(2) < 0 %if the jth element of the node vector is < 0 then Om =
360deg - Om
    Om_pre = acos(nvec(1)/n)*r2d;
    Om = 360 - Om_pre;
else
    Om = acos(nvec(1)/n)*r2d;
end

%TIME OF PERIAPSIS
z = 1/a;
if z > 0
    %elliptical case

```

```

mean_mo = sqrt(mu*z^3);
cE = inv(e)*(1-z*rmag);
sE = z^2*(dot(R,V))/(mean_mo*e);
E = atan2(sE,cE);
M = (E - (e*sin(E)))*r2d;
M_check = (E - (e_check*sin(E)))*r2d;
tau = t -(E - e*sin(E))/mean_mo;
tau_check = t - (M_check*d2r)/mean_mo;
elseif z < 0,
    %hyperbolic case
    mean_mo = sqrt(mu*-z^3);
    cE = inv(e)*(1-z*rmag);
    sE = z^2*(dot(R,V))/(mean_mo*e);
    E1 = asinh(sE);
    E2 = acosh(cE);
    E = E2;
    if E1 < 0, E = -E2; end
    tau = t -(e*sinh(E) - E)/mean_mo;
else % z == 0
    %parabolic case
    mean_mo = sqrt(mu/p^3);
    cE = p/rmag - 1;
    sE = (dot(R,V))/rmag*sqrt(p/mu);
    E = atan2(sE,cE);
    tau = t -((tan(E/2)^3)/6 +tan(E/2)/2)/mean_mo;
end
%M
%The following 3 elements (nu, uo, lo) can be substituted for tau

%TRUE ANOMALY, the angle in satellite's orbit plane between periapsis
%and satellite's position at a particular epoch, to

if dot(R,V) < 0, %if the dot product of R and V is greater than 0, then the
true anomaly should be less than 180 deg.
    fpre = acos((dot(evec,R))/(e*rmag)) *r2d;
    f = 360 - fpre;
else

```

```

    f = acos((dot(evec,R))/(e*rmag)) *r2d;
end

% ARGUMENT OF PERIAPSIS, w
if e == 0, %circular orbit, no periapsis
    w = undefined;
end

if evec(3) < 0, %if the kth element of the eccentricity vector is >0 then w is
%<180 deg.
    wpre = acos((dot(nvec,evec))/(n*e))*r2d;
    w = 360 - wpre;
else
    w = acos((dot(nvec,evec))/(n*e))*r2d;
end

%LONGITUDE OF PERIAPSIS, w_bar, can be substituted for w
w_bar = Om + w;

if e == 0, %circular orbit, no periapsis
    w_bar = undefined;
end

%TRUE LONGITUDE OF PERIAPSIS, w_bar_true
if evec(2) < 0
    w_bar_true_pre = acos(evec(1)/e)*r2d;
    w_bar_true = 360 - w_bar_true_pre;
else
    w_bar_true = acos(evec(1)/e)*r2d;
end

%ARGUMENT OF LATITUDE, uo, the angle between nvec and R at
%epoch, to
if R(3) < 0, %if the kth element of the position vector is < 0, then uo is <
180 deg.
    uo_pre = acos((dot(nvec,R))/(n*rmag)) *r2d; % = w + nu
    uo = 360 - uo_pre;

```

```

else
    uo = acos((dot(nvec,R))/(n*rmag)) *r2d ;% = w + nu
end

%TRUE LONGITUDE, lo, the angle between I and ro(the radius vector to
%the satellite at epoch, to), measured eastward to the ascending node if it
%exists and then in the orbital plane to ro. If Om, w, and f are all
%defined then lo.

if Om == NaN, %equatorial orbit
    lo = w_bar + f;
elseif w == NaN, %circular orbit
    lo = Om + uo;
else
    lo = acos(R(1)/rmag)*r2d;
    if R(2) < 0
        lo_pre = acos(R(1)/rmag)*r2d;
        lo = 360 - lo_pre;
    end
end

format long e
%***OPTIONAL EPHMERIDES TO DISPLAY*****
%disp('Ephem = [a e i Om w tau]');
%Ephem = [a e i Om w tau];
%disp('Ephem_2 = [f(True anomaly), M(Mean anomaly), w_bar_true(true
%longitude of periapsis), w_bar(longitude of periapsis),uo(argument of
%latitude at epoch), lo(true longitude at epoch)]');
%Ephem_2 = [f M w_bar_true w_bar uo lo];
%disp('Misc =[h(angular momentum), p(semi-latus rectum), ra(apoapsis),
%rp(periapsis),Period]')
%Misc = [h, p, ra, rp, Period];
disp('Ephemeris = [a e i Om w M f Period]')
E = [a e i Om w M f Period];

```

**LISA\_EOMderivs.m** (from Section 5.2.2. The Nominal Orbit)

```
function Xdot = LISA_EOMderivs(t, X)
% *****
% Xdot = LISA_EOMderivs(t, X)
%
% Simulate the orbits of 3 sun-orbiting LISA spacecraft with the given initial
conditions
% using the Equation of Motion (EOM). EOM = -(mu/(norm(r))^3).*r
%
% INPUT: X = initial state of the orbit =
% [S/C #1 position & velocity, S/C #2 position & velocity, S/C #3
position & velocity,
% Tracking Station Position & Velocity] in 3-D (1 x 24)
% t = initial time
%
% OUTPUT: Xdot = the derivative of the inputted state =
% Xdot = [rdot_1; rddot_1; rdot_2; rddot_2; rdot_3; rddot_3; rdot_rts;
% rddot_rts]
%
% Command Window: [t, X] = ode45('LISA_EOMderivs', timespan, XO,
% odeset('RelTol', 1e-12, 'AbsTol', 1e-12));
% *****

mus = (1.32712428e20); % m3/s2, Sun's gravitational parameter

r1 = X(1:3); % m, LISA spacecraft #1 location
rmag_1 = norm(r1);
rdot_1 = X(4:6); % m/sec, LISA spacecraft #1 velocity
rddot_1 = -(mus/(rmag_1^3))*r1; % m/s2, EOM for spacecraft #1

r2 = X(7:9); % m, LISA spacecraft #2 location
rmag_2 = norm(r2);
rdot_2 = X(10:12); % m/sec, LISA spacecraft #2 velocity
rddot_2 = -(mus/(rmag_2^3))*r2; % m/s2, EOM for spacecraft #2

r3 = X(13:15); % m, LISA spacecraft #3 location
```

```
rmag_3 = norm(r3);
rdot_3 = X(16:18); % m/sec, LISA spacecraft #3 velocity
rddot_3 = -(mus/(rmag_3^3))*r3; % m/s2, EOM for spacecraft #3

rts = X(19:21); % m, tracking station location IS Earth (treat Earth as a
% point)
rts_mag = norm(rts);
rdot_rts = X(22:24); % m/sec, Earth velocity
rddot_rts = -(mus/(rts_mag^3))*rts; % m/s2, EOM for spacecraft #3

Xdot = [rdot_1; rddot_1; rdot_2; rddot_2; rdot_3; rddot_3; rdot_rts;
rddot_rts]; % a 24x1 matrix
```

**NomOrb** (from 5.2.2. The Nominal Orbit)

```
function [X_traj, VirtualOrb_Diff, Leg_Length, EE, E1, E2, E3] =
NomOrb()
% *****
% Converts Steve Hughes's set of orbital elements for an equilateral
% formation to position and velocity
%
% REF.: Steve Hughes's paper "Preliminary Optimal ORbit Design for the
% Laser
% Interferometer Space Antenna (LISA)"
%
% CALLS: Orb2X2.m, X2Orb.m, LISA_EOMderivs.m
% *****
% ***CONSTANTS*****

mus = 1.32712428e11*((1e3)^3); % m3/s2, Sun's gravitational parameter
d = 5e6*1000; % m, nominal leg length
r2d = (180/pi); % radians to degrees converter
Sun = [0 0 0]; % Sun is the Center
% ***ORBITAL ELEMENTS*****
```

```

ae = (1.49598023e8)*1000; %m, Semimajor Axis of Earth's orbit around
Sun
e = d/((2*sqrt(3))*ae); %unitless, Eccentricity
i = (d/(2*ae))*r2d; %deg., Inclination (low inclination)
w = 90; %90 deg. = (pi/2) or 270 deg. = (3*pi)/2 Argument of Periapsis
Om1 = 0; %deg., Right Ascension of the Ascending Node of LISA 1
Om2 = Om1 + 120; %deg., 120 deg. = ((2*pi)/3) Right Ascension of the
Ascending Node of LISA 2
Om3 = Om1 - 120; %deg., Right Ascension of the Ascending Node of LISA
3
M1 = 0; %deg., Mean Anomaly of LISA 1
M2 = M1 - 120; %deg., 120 deg = ((2*pi)/3) Mean Anomaly of LISA 2
M3 = M1 + 120; %deg., Mean Anomaly of LISA 3

%***CALL Orb2X2.m to convert to R & V*****

%Ephem4 = [a e i Om w M]
%NOTE: Degrees will be converted to Radians in the code Orb2X2
EphemL1 = [ae e i Om1 w M1];
EphemL2 = [ae e i Om2 w M2];
EphemL3 = [ae e i Om3 w M3];

%[R, V] = Orb2X2(Ephem, t, mu, type)

[R1, V1] = Orb2X2(EphemL1, 0, mus, 4);
[R2, V2] = Orb2X2(EphemL2, 0, mus, 4);
[R3, V3] = Orb2X2(EphemL3, 0, mus, 4);

%***Position & Velocity of Earth Relative to Center of LISA*****
Rvs = [0 ae 0]; % 1x3, 3-D initial Reference position of center of LISA's
%virtual orbit around the Sun, lies all along the y-axis initially

wvs = sqrt(mus/ae^3); %angular rate of virtual orbit
v = wvs*ae; %m/s, tangential velocities of the spacecrafts
Vvs = [-v 0 0]; %initial velocity of the virtual orbit in the x-axis

```

```

th = 20*(pi/180); %Earth is 20 deg. rotation about the z-axis ahead of the
%LISA formation converted to radians
REt = [cos(th) -sin(th) 0; sin(th) cos(th) 0; 0 0 1]*Rvs'; %the transformed
%position of the Earth
VEt = [cos(th) -sin(th) 0; sin(th) cos(th) 0; 0 0 1]*Vvs'; %the transformed
%velocity of the Earth

%***CALL X2Orb.m to GET ORBITAL ELEMENTS OF LISA &
%EARTH*****
%E] = X2Orb(R, V, t, mu)
%E = [a e i Om w M f Period]

[EE] = X2Orb(REt, VEt, 0, mus);
[E1] = X2Orb(R1, V1, 0, mus);
[E2] = X2Orb(R2, V2, 0, mus);
[E3] = X2Orb(R3, V3, 0, mus);

%***LEG LENGTHS*****
L12 = (norm(R2 - R1)); %nx1, leg lengths
L23 = (norm(R3 - R2));
L31 = (norm(R1 - R3));
Leg_Length = [L12, L23, L31];

%***DIFFERENCE FROM VIRTUAL ORBIT TO EACH LISA &
%EARTH*****
L1C = norm(Rvs' - R1);
L2C = norm(Rvs' - R2);
L3C = norm(Rvs' - R3);
EC2 = norm(Rvs' - REt);
VirtualOrb_Diff = [L1C, L2C, L3C, EC2];

%***PROPAGATE THE ORBIT OVER THE YEAR USING THE
%EOM*****
to = 0;
I = (60*60)/2; %sec., 1/2 HOUR Intervals for the entire YEAR
tf = 366.25*24*60*60; %sec, 1 SIDEREAL YEAR
ts = 0:I:tf;

```



```
Xstar = [R1; V1; R2; V2; R3; V3; REt; VEt];
[t, Xs] = ode45('LISA_EOMderivs', ts, Xstar, odeset('RelTol',1e-
8,'AbsTol',1e-8));
X_traj = [ts(:) Xs(:, :)]; %n x 25 matrix
```

**LISA\_STMderivs.m** (from Section 5.2.3 The Algorithm)

```
function stmdot = LISA_stmderive(t, Xstm)
%*****
%function stmdot = LISA_stmderive(t, Xstm)
%
%Function that requires ode45 to integrate the state transition matrix over
%a timespan
%
%INPUTS: t
%      Xstm = the state vector and the state transition matrix
%OUTPUTS: stmdot
%
%Command Window Prompts
%[t, Xstm] = ode45('LISA_stmderive', timespan, Xstm, odeset('RelTol',
% 1e-8,'AbsTol',1e-8))
%*****
mus = (1.32712428e20); %m3/s2, Sun's gravitational parameter

r1 = Xstm(1:3); %m, LISA spacecraft #1 location
rmag_1 = norm(r1);
rdot_1 = Xstm(4:6); %m/sec, LISA spacecraft #1 velocity
rddot_1 = -(mus/(rmag_1^3))*r1; %m/s2, EOM for spacecraft #1

r2 = Xstm(7:9); %m, LISA spacecraft #2 location
rmag_2 = norm(r2);
rdot_2 = Xstm(10:12); %m/sec, LISA spacecraft #2 velocity
rddot_2 = -(mus/(rmag_2^3))*r2; %m/s2, EOM for spacecraft #2

r3 = Xstm(13:15); %m, LISA spacecraft #3 location
```

```
rmag_3 = norm(r3);
rdot_3 = Xstm(16:18); %m/sec, LISA spacecraft #3 velocity
rddot_3 = -(mus/(rmag_3^3))*r3; %m/s2, EOM for spacecraft #3
```

```
rts = Xstm(19:21); %m, tracking station location IS Earth (treat Earth as a
point)
rts_mag = norm(rts);
rdot_rts = Xstm(22:24); %m/sec, Earth velocity
rddot_rts = -(mus/(rts_mag^3))*rts; %m/s2, EOM for spacecraft #3
```

```
Xdot = [rdot_1; rddot_1; rdot_2; rddot_2; rdot_3; rddot_3; rdot_rts;
rddot_rts]; %a 24x1 matrix
```

```
%---A matrix, not constant because the variable r in the equations
I = eye(3,3);
Z3 = zeros(3,3);
```

```
Gamma1 = -(mus/((norm(r1))^3))*I + 3*(((mus*r1)/(norm(r1)^5))*r1');
% 3x3, Gravity gradient matrix
Gamma2 = -(mus/((norm(r2))^3))*I + 3*(((mus*r2)/(norm(r2)^5))*r2');
% 3x3, Gravity gradient matrix
Gamma3 = -(mus/((norm(r3))^3))*I + 3*(((mus*r3)/(norm(r3)^5))*r3');
% 3x3, Gravity gradient matrix
Gammae = -(mus/((norm(rts))^3))*I + 3*(((mus*rts)/(norm(rts)^5))*rts');
% 3x3, Gravity gradient matrix
```

```
A = [Z3 I Z3 Z3 Z3 Z3 Z3 Z3; Gamma1 Z3 Z3 Z3 Z3 Z3 Z3; Z3 Z3
Z3 I Z3 Z3 Z3 Z3;...
Z3 Z3 Gamma2 Z3 Z3 Z3 Z3 Z3; Z3 Z3 Z3 Z3 Z3 I Z3 Z3; Z3 Z3
Z3 Z3 Gamma3 Z3 Z3 Z3;...
Z3 Z3 Z3 Z3 Z3 Z3 Z3 I; Z3 Z3 Z3 Z3 Z3 Z3 Gammae Z3];
%24x24 matrix
```

```
stm = reshape(Xstm(25:600), 24, 24);
stmdot = A*stm;
stmdot = [Xdot; (reshape(stmdot, 576, 1))]; %a 600x1 matrix
```

**H\_partials.m** (from Section 5.2.3. The Algorithm)

```

function [Hi_tilda, Hl_tilda, Hq_tilda] = H_partials7(Gi, GiR, Xstar, obs)
% *****
% *****
X = Xstar(1,:);
Z13 = zeros(1,3);
%***THE PARTIALS
%Range wrt Position

dp1dx1 = (X(1)-X(19))/Gi(1); %m, L1 Range wrt L1 Position
dp1dy1 = (X(2)-X(20))/Gi(1);
dp1dz1 = (X(3)-X(21))/Gi(1);
dp1dR = [dp1dx1 dp1dy1 dp1dz1];

dp2dx2 = (X(7)-X(19))/Gi(2); %m, L2 Range wrt L2 Position
dp2dy2 = (X(8)-X(20))/Gi(2);
dp2dz2 = (X(9)-X(21))/Gi(2);
dp2dR = [dp2dx2 dp2dy2 dp2dz2];

dp3dx3 = (X(13)-X(19))/Gi(3); %m, L3 Range wrt L3 Position
dp3dy3 = (X(14)-X(20))/Gi(3);
dp3dz3 = (X(15)-X(21))/Gi(3);
dp3dR = [dp3dx3 dp3dy3 dp3dz3];

%RANGE wrt VELOCITY

dp1dV = Z13;
dp2dV = Z13;
dp3dV = Z13;

%***RANGE RATE
%LISA 1 RANGE RATE wrt its POSITION

dpdot1dx1 = ((X(4)-X(22))/Gi(1)) - ((Gi(4)*(X(1)-X(19)))/(Gi(1)^2));

dpdot1dy1 = ((X(5)-X(23))/Gi(1)) - ((Gi(4)*(X(2)-X(20)))/(Gi(1)^2));

```

```

dpdot1dz1 = ((X(6)-X(24))/Gi(1)) - ((Gi(4)*(X(3)-X(21)))/(Gi(1)^2));

dpdot1dR = [dpdot1dx1 dpdot1dy1 dpdot1dz1];

%LISA 2 RANGE RATE wrt its POSITION
dpdot2dx2 = ((X(10)-X(22))/Gi(2)) - ((Gi(5)*(X(7)-X(19)))/(Gi(2)^2));

dpdot2dy2 = ((X(11)-X(23))/Gi(2)) - ((Gi(5)*(X(8)-X(20)))/(Gi(2)^2));

dpdot2dz2 = ((X(12)-X(24))/Gi(2)) - ((Gi(5)*(X(9)-X(21)))/(Gi(2)^2));

dpdot2dR = [dpdot2dx2 dpdot2dy2 dpdot2dz2];

%LISA 3 RANGE RATE wrt its POSITION
dpdot3dx3 = ((X(16)-X(22))/Gi(3)) - ((Gi(6)*(X(13)-X(19)))/(Gi(3)^2));

dpdot3dy3 = ((X(17)-X(23))/Gi(3)) - ((Gi(6)*(X(14)-X(20)))/(Gi(3)^2));

dpdot3dz3 = ((X(18)-X(24))/Gi(3)) - ((Gi(6)*(X(15)-X(21)))/(Gi(3)^2));

dpdot3dR = [dpdot3dx3 dpdot3dy3 dpdot3dz3];

%LISA 1,2,3 Range Rate wrt its VELOCITY
dpdot1dV = [dp1dx1 dp1dy1 dp1dz1];
dpdot2dV = [dp2dx2 dp2dy2 dp2dz2];
dpdot3dV = [dp3dx3 dp3dy3 dp3dz3];

%***DECLINATION PARTIALS
%LISA 1 declination wrt its POSITION
dd1dx1 = ((X(3)-X(21))* (-X(1) + X(19)))/...
          ((Gi(1)^3)*(1-(((X(3)-X(21))^2)/(Gi(1)^2)))^(1/2));

dd1dy1 = ((X(3)-X(21))* (-X(2) + X(20)))/...
          ((Gi(1)^3)*(1-(((X(3)-X(21))^2)/(Gi(1)^2)))^(1/2));

```

```

dd1dz1 = ((1/Gi(1)) - (.5*(((X(3)-X(21))*((2*X(3)) -
(2*X(21))))/(Gi(1)^3))))/...
(1-(((X(3)-X(21))^2)/(Gi(1)^2))))^(1/2);

dd1dR = [dd1dx1 dd1dy1 dd1dz1];

%LISA 2 declination wrt its POSITION
dd2dx2 = ((X(9)-X(21))* (-X(7) + X(19)))/...
((Gi(2)^3)*(1-(((X(9)-X(21))^2)/(Gi(2)^2))))^(1/2));

dd2dy2 = ((X(9)-X(21))* (-X(8) + X(20)))/...
((Gi(2)^3)*(1-(((X(9)-X(21))^2)/(Gi(2)^2))))^(1/2));

dd2dz2 = ((1/Gi(2)) - (.5*(((X(9)-X(21))*((2*X(9)) -
(2*X(21))))/(Gi(2)^3))))/...
(1-(((X(9)-X(21))^2)/(Gi(2)^2))))^(1/2);

dd2dR = [dd2dx2 dd2dy2 dd2dz2];
%LISA 3 declination wrt its POSITION
dd3dx3 = ((X(15)-X(21))* (-X(13) + X(19)))/...
((Gi(3)^3)*(1-(((X(15)-X(21))^2)/(Gi(3)^2))))^(1/2));

dd3dy3 = ((X(15)-X(21))* (-X(14) + X(20)))/...
((Gi(3)^3)*(1-(((X(15)-X(21))^2)/(Gi(3)^2))))^(1/2));

dd3dz3 = ((1/Gi(3)) - (.5*(((X(15)-X(21))*((2*X(15)) -
(2*X(21))))/(Gi(3)^3))))/...
(1-(((X(15)-X(21))^2)/(Gi(3)^2))))^(1/2);

dd3dR = [dd3dx3 dd3dy3 dd3dz3];

%***Delta wrt VELOCITY
dd1dV = Z13;
dd2dV = Z13;
dd3dV = Z13;

%RIGHT ASCENSION PARTIALS

```

```

%LISA 1 right ascension wrt its position
da1dx1 = (-1*(X(2)-X(20)))/(((X(1)-X(19))^2)*(1 + ((X(2)-
X(20))^2)/((X(1)-X(19))^2)));
da1dy1 = 1/((X(1)-X(19))*(1 + ((X(2)-X(20))^2)/((X(1)-X(19))^2)));
da1dz1 = 0;
da1dR = [da1dx1 da1dy1 da1dz1];

%LISA 2 right ascension wrt its position
da2dx2 = (-1*(X(8)-X(20)))/(((X(7)-X(19))^2)*(1 + ((X(8)-
X(20))^2)/((X(7)-X(19))^2)));
da2dy2 = 1/((X(7)-X(19))*(1 + ((X(8)-X(20))^2)/((X(7)-X(19))^2)));
da2dz2 = 0;
da2dR = [da2dx2 da2dy2 da2dz2];

%LISA 3 right ascension wrt its position
da3dx3 = (-1*(X(14)-X(20)))/(((X(13)-X(19))^2)*(1 + ((X(14)-
X(20))^2)/((X(13)-X(19))^2)));
da3dy3 = 1/((X(13)-X(19))*(1 + ((X(14)-X(20))^2)/((X(13)-X(19))^2)));
da3dz3 = 0;
da3dR = [da3dx3 da3dy3 da3dz3];

%***Alpha wrt VELOCITY
da1dV = Z13;
da2dV = Z13;
da3dV = Z13;

%***INTERSPACECRAFT RANGE AND RANGE RATE
%*****
%LISA 1 range wrt LISA 3 position
dpL13dx1 = (X(1) - X(13))/GiR(1);
dpL13dy1 = (X(2) - X(14))/GiR(1);
dpL13dz1 = (X(3) - X(15))/GiR(1);
dpL13dR1 = [dpL13dx1 dpL13dy1 dpL13dz1];
dpL13dV1 = Z13;

%LISA 1 range rate wrt LISA 3 position

```

$$\text{dpdotL13dx1} = ((X(4)-X(16))/\text{GiR}(1)) - ((\text{GiR}(4)*(X(1)-X(13)))/(\text{GiR}(1)^2));$$

$$\text{dpdotL13dy1} = ((X(5)-X(17))/\text{GiR}(1)) - ((\text{GiR}(4)*(X(2)-X(14)))/(\text{GiR}(1)^2));$$

$$\text{dpdotL13dz1} = ((X(6)-X(18))/\text{GiR}(1)) - ((\text{GiR}(4)*(X(3)-X(15)))/(\text{GiR}(1)^2));$$

$$\text{dpdotL13dR1} = [\text{dpdotL13dx1} \text{ dpdotL13dy1} \text{ dpdotL13dz1}];$$

$$\text{dpdotL13dV1} = [\text{dpL13dx1} \text{ dpL13dy1} \text{ dpL13dz1}];$$

%LISA 3 range wrt LISA 1 position

$$\text{dpL13dx3} = (-X(1) + X(13))/\text{GiR}(1);$$

$$\text{dpL13dy3} = (-X(2) + X(14))/\text{GiR}(1);$$

$$\text{dpL13dz3} = (-X(3) + X(15))/\text{GiR}(1);$$

$$\text{dpL13dR3} = [\text{dpL13dx3} \text{ dpL13dy3} \text{ dpL13dz3}];$$

$$\text{dpL13dV3} = \text{Z13};$$

%LISA 3 range rate wrt LISA 1 position

$$\text{dpdotL13dx3} = ((-X(4)+X(16))/\text{GiR}(1)) - ((\text{GiR}(4)*(-X(1)+X(13)))/(\text{GiR}(1)^2));$$

$$\text{dpdotL13dy3} = ((-X(5)+X(17))/\text{GiR}(1)) - ((\text{GiR}(4)*(-X(2)+X(14)))/(\text{GiR}(1)^2));$$

$$\text{dpdotL13dz3} = ((-X(6)+X(18))/\text{GiR}(1)) - ((\text{GiR}(4)*(-X(3)+X(15)))/(\text{GiR}(1)^2));$$

$$\text{dpdotL13dR3} = [\text{dpdotL13dx3} \text{ dpdotL13dy3} \text{ dpdotL13dz3}];$$

$$\text{dpdotL13dV3} = [\text{dpL13dx3} \text{ dpL13dy3} \text{ dpL13dz3}];$$

%\*\*\*\*\*

%LISA 2 wrt LISA 1

%LISA 2 range wrt LISA 1 position

$$\text{dpL21dx2} = (X(7) - X(1))/\text{GiR}(2);$$

$$\text{dpL21dy2} = (X(8) - X(2))/\text{GiR}(2);$$

$$\text{dpL21dz2} = (X(9) - X(3))/\text{GiR}(2);$$

$$\text{dpL21dR2} = [\text{dpL21dx2} \text{ dpL21dy2} \text{ dpL21dz2}];$$

$$\text{dpL21dV2} = \text{Z13};$$

%LISA 2 range rate wrt LISA 1 position

$$\text{dpdotL21dx2} = ((X(10)-X(4))/\text{GiR}(2)) - ((\text{GiR}(5)*(X(7)-X(1)))/(\text{GiR}(2)^2));$$

$$\text{dpdotL21dy2} = ((X(11)-X(5))/\text{GiR}(2)) - ((\text{GiR}(5)*(X(8)-X(2)))/(\text{GiR}(2)^2));$$

$$\text{dpdotL21dz2} = ((X(12)-X(6))/\text{GiR}(2)) - ((\text{GiR}(5)*(X(9)-X(3)))/(\text{GiR}(2)^2));$$

$$\text{dpdotL21dR2} = [\text{dpdotL21dx2} \text{ dpdotL21dy2} \text{ dpdotL21dz2}];$$

$$\text{dpdotL21dV2} = [\text{dpL21dx2} \text{ dpL21dy2} \text{ dpL21dz2}];$$

%LISA 1 range wrt LISA 2 position

$$\text{dpL21dx1} = (-X(7) + X(1))/\text{GiR}(2);$$

$$\text{dpL21dy1} = (-X(8) + X(2))/\text{GiR}(2);$$

$$\text{dpL21dz1} = (-X(9) + X(3))/\text{GiR}(2);$$

$$\text{dpL21dR1} = [\text{dpL21dx1} \text{ dpL21dy1} \text{ dpL21dz1}];$$

$$\text{dpL21dV1} = \text{Z13};$$

%LISA 1 range rate wrt LISA 2 position

$$\text{dpdotL21dx1} = ((-X(10)+X(4))/\text{GiR}(2)) - ((\text{GiR}(5)*(-X(7)+X(1)))/(\text{GiR}(2)^2));$$

$$\text{dpdotL21dy1} = ((-X(11)+X(5))/\text{GiR}(2)) - ((\text{GiR}(5)*(-X(8)+X(2)))/(\text{GiR}(2)^2));$$

$$\text{dpdotL21dz1} = ((-X(12)+X(6))/\text{GiR}(2)) - ((\text{GiR}(5)*(-X(9)+X(3)))/(\text{GiR}(2)^2));$$

$$\text{dpdotL21dR1} = [\text{dpdotL21dx1} \text{ dpdotL21dy1} \text{ dpdotL21dz1}];$$

$$\text{dpdotL21dV1} = [\text{dpL21dx1} \text{ dpL21dy1} \text{ dpL21dz1}];$$

%\*\*\*\*\*

```

%LISA 3 wrt LISA 2
%LISA 3 range wrt LISA 2 position
dpL32dx3 = (X(13) - X(7))/GiR(3);
dpL32dy3 = (X(14) - X(8))/GiR(3);
dpL32dz3 = (X(15) - X(9))/GiR(3);
dpL32dR3 = [dpL32dx3 dpL32dy3 dpL32dz3];
dpL32dV3 = Z13;

%LISA 3 range rate wrt LISA 2 position
dpdotL32dx3 = ((X(16)-X(10))/GiR(3)) - ((GiR(6)*(X(13)-
X(7)))/(GiR(3)^2));

dpdotL32dy3 = ((X(17)-X(11))/GiR(3)) - ((GiR(6)*(X(14)-
X(8)))/(GiR(3)^2));

dpdotL32dz3 = ((X(18)-X(12))/GiR(3)) - ((GiR(6)*(X(15)-
X(9)))/(GiR(3)^2));

dpdotL32dR3 = [dpdotL32dx3 dpdotL32dy3 dpdotL32dz3];
dpdotL32dV3 = [dpL32dx3 dpL32dy3 dpL32dz3];

%LISA 2 range wrt LISA 3 position
dpL32dx2 = (-X(13) + X(7))/GiR(3);
dpL32dy2 = (-X(14) + X(8))/GiR(3);
dpL32dz2 = (-X(15) + X(9))/GiR(3);
dpL32dR2 = [dpL32dx2 dpL32dy2 dpL32dz2];
dpL32dV2 = Z13;

%LISA 2 range rate wrt LISA 3 position
dpdotL32dx2 = ((-X(16)+X(10))/GiR(3)) - ((GiR(6)*(-
X(13)+X(7)))/(GiR(3)^2));

dpdotL32dy2 = ((-X(17)+X(11))/GiR(3)) - ((GiR(6)*(-
X(14)+X(8)))/(GiR(3)^2));

dpdotL32dz2 = ((-X(18)+X(12))/GiR(3)) - ((GiR(6)*(-
X(15)+X(9)))/(GiR(3)^2));

dpdotL32dR2 = [dpdotL32dx2 dpdotL32dy2 dpdotL32dz2];
dpdotL32dV2 = [dpL32dx2 dpL32dy2 dpL32dz2];

switch(obs)
case 0 % DSN & VLBI
%Hi, Hl, Hq into 4x18 matrices [1x3, 1x3, 1x3, 1x3, 1x3, 1x3]
Hi_tilda = [dp1dR dp1dV Z13 Z13 Z13 Z13;...
dpdot1dR dpdot1dV Z13 Z13 Z13 Z13;...
dd1dR dd1dV Z13 Z13 Z13 Z13;...
da1dR da1dV Z13 Z13 Z13 Z13];

Hl_tilda = [Z13 Z13 dp2dR dp2dV Z13 Z13;...
Z13 Z13 dpdot2dR dpdot2dV Z13 Z13;...
Z13 Z13 dd2dR dd2dV Z13 Z13;...
Z13 Z13 da2dR da2dV Z13 Z13];

Hq_tilda = [Z13 Z13 Z13 Z13 dp3dR dp3dV;...
Z13 Z13 Z13 Z13 dpdot3dR dpdot3dV;...
Z13 Z13 Z13 Z13 dd3dR dd3dV;...
Z13 Z13 Z13 Z13 da3dR da3dV];

case 1 % DSN, VLBI, RELATIVE
%H 18x18 matrix [1x3, 1x3, 1x3, 1x3, 1x3, 1x3]
Hi_tilda = [dp1dR dp1dV Z13 Z13 Z13 Z13;...
dpdot1dR dpdot1dV Z13 Z13 Z13 Z13;...
dd1dR dd1dV Z13 Z13 Z13 Z13;...
da1dR da1dV Z13 Z13 Z13 Z13;...
dpL13dR1 dpL13dV1 Z13 Z13 dpL13dR3 dpL13dV3;...
dpdotL13dR1 dpdotL13dV1 Z13 Z13 dpdotL13dR3
dpdotL13dV3;...% wrt LISA 1
Z13 Z13 dp2dR dp2dV Z13 Z13;...
Z13 Z13 dpdot2dR dpdot2dV Z13 Z13;...
Z13 Z13 dd2dR dd2dV Z13 Z13;...
Z13 Z13 da2dR da2dV Z13 Z13;...
dpL21dR1 dpL21dV1 dpL21dR2 dpL21dV2 Z13 Z13;...

```

```

    dpdotL21dR1 dpdotL21dV1 dpdotL21dR2 dpdotL21dV2 Z13
Z13;...% wrt LISA 2
    Z13 Z13 Z13 Z13 dp3dR dp3dV;...
    Z13 Z13 Z13 Z13 dpdot3dR dpdot3dV;...
    Z13 Z13 Z13 Z13 dd3dR dd3dV;...
    Z13 Z13 Z13 Z13 da3dR da3dV;...
    Z13 Z13 dpL32dR2 dpL32dV2 dpL32dR3 dpL32dV3;...
    Z13 Z13 dpdotL32dR2 dpdotL32dV2 dpdotL32dR3 dpdotL32dV3];
% wrt LISA 3

    Hl_tilda = [Z13];

    Hq_tilda = [Z13];

case 2 %DSN only
%Hi, Hl, Hq into 2x18 matrices [1x3, 1x3, 1x3, 1x3, 1x3, 1x3]
Hi_tilda = [dp1dR dp1dV Z13 Z13 Z13 Z13;...
    dpdot1dR dpdot1dV Z13 Z13 Z13 Z13];

    Hl_tilda = [Z13 Z13 dp2dR dp2dV Z13 Z13;...
    Z13 Z13 dpdot2dR dpdot2dV Z13 Z13];

    Hq_tilda = [Z13 Z13 Z13 Z13 dp3dR dp3dV;...
    Z13 Z13 Z13 Z13 dpdot3dR dpdot3dV];

case 3 %DSN & RELATIVE
%H 12x18 matrix [1x3, 1x3, 1x3, 1x3, 1x3, 1x3]

    Hi_tilda = [dp1dR dp1dV Z13 Z13 Z13 Z13;...
    dpdot1dR dpdot1dV Z13 Z13 Z13 Z13;...
    dpL13dR1 dpL13dV1 Z13 Z13 dpL13dR3 dpL13dV3;...
    dpdotL13dR1 dpdotL13dV1 Z13 Z13 dpdotL13dR3
dpdotL13dV3;...% wrt LISA 1
    Z13 Z13 dp2dR dp2dV Z13 Z13;...
    Z13 Z13 dpdot2dR dpdot2dV Z13 Z13;...
    dpL21dR1 dpL21dV1 dpL21dR2 dpL21dV2 Z13 Z13;...

```

```

    dpdotL21dR1 dpdotL21dV1 dpdotL21dR2 dpdotL21dV2 Z13
Z13;...% wrt LISA 2
    Z13 Z13 Z13 Z13 dp3dR dp3dV;...
    Z13 Z13 Z13 Z13 dpdot3dR dpdot3dV;...
    Z13 Z13 dpL32dR2 dpL32dV2 dpL32dR3 dpL32dV3;...
    Z13 Z13 dpdotL32dR2 dpdotL32dV2 dpdotL32dR3 dpdotL32dV3];
% wrt LISA 3

    Hl_tilda = [Z13];

    Hq_tilda = [Z13];

    otherwise
        error('Invalid case (0 <= obs <= 3)');
    end

RealObs.m (from Section 5.2.4 Simulation of Measurement Noise)
function [Gi, L1tot, L2tot, L3tot] = RealObs(to, tf, I, X, Xref, sigma_range,
sigma_rrate, sigma_alpha, sigma_delta, sigma_ci, sigma_f, B, plt)
% *****
% This function calculates the "perfect" observations and also calculates
% the corresponding "noisy" measurements.
%
% INPUT : to = Initial time to start propagation (sec)
%         tf = Final time to end propagation (sec)
%         I = Time intervals (sec)
%         XO = An Initial Condition of the State (1x24)
%         sigma_range (m), sigma_rrate(m/s), sigma_alpha(rad),
sigma_delta(rad) = The noises
%         associated with each measurement
%         br(m), brr(m/s), ba(rad), bs(rad) = biases in measurements
%         plt = plot command =0 "off", = 1 "on"
%
% OUTPUT : Gi = 1x12 vector of calculated range, range rate, alpha and
delta

```

```

%           angles of each LISA
%           L1tot, L2tot, L3tot = 4x1 vector of Measurement Deviations
[range, range rate, alpha and delta
%           angles] of each LISA
%*****
%---INITIALIZATION
if nargin < 13,
    plt = 0; %default to turn plot command off
end

%***Parse Truth vector (X)into position & velocity
components*****
R1 = X(1:3); %m (1x3)
R2 = X(7:9);
R3 = X(13:15);
R = [R1 R2 R3]; %(1x9)

V1 = X(4:6); %m/s (1x3)
V2 = X(10:12);
V3 = X(16:18);
V = [V1 V2 V3]; %(1x9)

Rts= X(19:21); %(1x3)
Vts = X(22:24);

%***Parse Reference vector (Xref) into position & velocity
components*****
R1s = Xref(1:3); %m (1x3)
R2s = Xref(7:9);
R3s = Xref(13:15);
Rs = [R1s R2s R3s]; %(1x9)

V1s = Xref(4:6); %m/s (1x3)
V2s = Xref(10:12);
V3s = Xref(16:18);
Vs = [V1s V2s V3s]; %(1x9)

```

```

Rts_ref = Xref(19:21); %(1x3)
Vts_ref = Xref(22:24);

%***CALCULATE THE "ACTUAL" MEASUREMENTS (i.e. Truth plus
noise and biases)**
%RANGE
p1 = norm(R1 - Rts); %m (1x1) range scalar for LISA 1
p2 = norm(R2 - Rts);
p3 = norm(R3 - Rts);

%RANGE RATE
pdot1 = (((R1(1) - Rts(1))*(V1(1) - Vts(1))) + ((R1(2) - Rts(2))*...
(V1(2) - Vts(2))) + ((R1(3) - Rts(3))*(V1(3) - Vts(3))))/p1; %m/s

pdot2 = (((R2(1) - Rts(1))*(V2(1) - Vts(1))) + ((R2(2) - Rts(2))*...
(V2(2) - Vts(2))) + ((R2(3) - Rts(3))*(V2(3) - Vts(3))))/p2;

pdot3 = (((R3(1) - Rts(1))*(V3(1) - Vts(1))) + ((R3(2) - Rts(2))*...
(V3(2) - Vts(2))) + ((R3(3) - Rts(3))*(V3(3) - Vts(3))))/p3;

%ALPHA
alpha1 = atan2((R1(2) - Rts(2)),(R1(1) - Rts(1))); %rad (1x1)
alpha2 = atan2((R2(2) - Rts(2)),(R2(1) - Rts(1)));
alpha3 = atan2((R3(2) - Rts(2)),(R3(1) - Rts(1)));

%DELTA
delta1 = asin((R1(3) - Rts(3))/p1); %rad (1x1)
delta2 = asin((R2(3) - Rts(3))/p2);
delta3 = asin((R3(3) - Rts(3))/p3);

%***ADD NOISE TO THE
TRUTH*****
Yi_1 = p1 + sigma_range(1) + sigma_ci(1) + B(1); %m
Yl_1 = p2 + sigma_range(2) + sigma_ci(2) + B(2);
Yq_1 = p3 + sigma_range(3) + sigma_ci(3) + B(3);

Yi_2 = (pdot1) + sigma_rrate(1) + sigma_f(1); %m/s

```

```

Yl_2 = (pdot2) + sigma_rrate(2) + sigma_f(2);
Yq_2 = (pdot3) + sigma_rrate(3) + sigma_f(3);

Yi_3 = (alpha1) + sigma_alpha(1); %rad
Yl_3 = (alpha2) + sigma_alpha(2);
Yq_3 = (alpha3) + sigma_alpha(3);

Yi_4 = (delta1) + sigma_delta(1); %rad
Yl_4 = (delta2) + sigma_delta(2);
Yq_4 = (delta3) + sigma_delta(3);

%***CALCULATE THE MEASUREMENTS (i.e. Reference
Observations)*****
%RANGE + ADD KNOWN BIASES TO CALCULATED
MEASUREMENT
p1s = norm(R1s - Rts_ref);% + B(1); %m (1x1) range scalar for LISA 1
p2s = norm(R2s - Rts_ref);% + B(2);
p3s = norm(R3s - Rts_ref);% + B(3);

%RANGE RATE
pdot1s = (((R1s(1) - Rts_ref(1))*(V1s(1) - Vts_ref(1))) + ((R1s(2) -
Rts_ref(2))*...
(V1s(2) - Vts_ref(2))) + ((R1s(3) - Rts_ref(3))*(V1s(3) -
Vts_ref(3))))/p1s); %m/s

pdot2s = (((R2s(1) - Rts_ref(1))*(V2s(1) - Vts_ref(1))) + ((R2s(2) -
Rts_ref(2))*...
(V2s(2) - Vts_ref(2))) + ((R2s(3) - Rts_ref(3))*(V2s(3) -
Vts_ref(3))))/p2s);

pdot3s = (((R3s(1) - Rts_ref(1))*(V3s(1) - Vts_ref(1))) + ((R3s(2) -
Rts_ref(2))*...
(V3s(2) - Vts_ref(2))) + ((R3s(3) - Rts_ref(3))*(V3s(3) -
Vts_ref(3))))/p3s);

%ALPHA

```

```

alpha1s = (atan2((R1s(2) - Rts_ref(2)),(R1s(1) - Rts_ref(1))))); %rad (1x1)
alpha2s = (atan2((R2s(2) - Rts_ref(2)),(R2s(1) - Rts_ref(1)))));
alpha3s = (atan2((R3s(2) - Rts_ref(2)),(R3s(1) - Rts_ref(1)))));

```

```

%DELTA
delta1s = (asin((R1s(3) - Rts_ref(3))/p1s)); %rad (1x1)
delta2s = (asin((R2s(3) - Rts_ref(3))/p2s));
delta3s = (asin((R3s(3) - Rts_ref(3))/p3s));

```

```

%*****

```

```

Yi = [Yi_1' Yl_1' Yq_1' Yi_2' Yl_2' Yq_2' Yi_3' Yl_3' Yq_3' Yi_4' Yl_4'
Yq_4']; %Real Measurements (1x12)

```

```

Gi = [p1s' p2s' p3s' pdot1s' pdot2s' pdot3s' alpha1s' alpha2s' alpha3s'
delta1s' delta2s' delta3s']; %Calculated Measurements (1x12)

```

```

diff = Yi-Gi;

```

```

%---COMPUTE MEASUREMENT DEVIATIONS, yi (1x1), AT
TIME ti

```

```

%LISA 1, 2 & 3 measurement deviations; measurement(Yi) -
Reference

```

```

%Measurement(Gi)

```

```

yi_1 = Yi(1) - Gi(1); % (1x1), range measurement deviations

```

```

yl_1 = Yi(2) - Gi(2);

```

```

yq_1 = Yi(3) - Gi(3);

```

```

yi_2 = Yi(4) - Gi(4); %range rate measurement deviations

```

```

yl_2 = Yi(5) - Gi(5);

```

```

yq_2 = Yi(6) - Gi(6);

```

```

yi_3 = Yi(10) - Gi(10); %delta measurement deviations

```

```

yl_3 = Yi(11) - Gi(11);

```

```

yq_3 = Yi(12) - Gi(12);

```

```

yi_4 = Yi(7) - Gi(7); %alpha measurement deviations

```

```

yl_4 = Yi(8) - Gi(8);

```

```

yq_4 = Yi(9) - Gi(9);

```



```

%Total Measurement deviations
L1tot = [yi_1; yi_2; yi_3; yi_4]; %(4x1)
L2tot = [yl_1; yl_2; yl_3; yl_4]; %(4x1)
L3tot = [yq_1; yq_2; yq_3; yq_4]; %(4x1)

%PLOTS (x,y,z)

if plt == 1,

figure
subplot(2,2,1)
plot(timespan, Gi(:,1),'b-', timespan, Gi(:,2),'g--', timespan, Gi(:,3), 'r-.',...
timespan, Yi(:,1),'b*', timespan, Yi(:,2),'g>', timespan, Yi(:,3),'rx');
xlabel('Time (sec)')
ylabel('Calculated / Real(noisy) Range(m)')
legend('LISA 1 (calc.)', 'LISA 2(calc.)', 'LISA 3(calc.)','LISA 1(real)', 'LISA
2(real)', 'LISA 3(real)')
title('LISA Range vs. Time')
grid on

subplot(2,2,2)
plot(timespan, Gi(:,4),'b-', timespan, Gi(:,5), 'g--', timespan, Gi(:,6),'r-.',...
timespan, Yi(:,4),'b*', timespan, Yi(:,5),'b>', timespan, Yi(:,6),'rx');
xlabel('Time (sec)')
ylabel('Calculated / Real(noisy) Range Rate(m/sec)')
%legend('LISA 1', 'LISA 2', 'LISA 3')
title('LISA Range Rate vs. Time')
grid on

subplot(2,2,3)
plot(timespan, Gi(:,7),'b-', timespan, Gi(:,8), 'g--', timespan, Gi(:,9),'r-.',...
timespan, Yi(:,7),'b*', timespan, Yi(:,8),'g>', timespan, Yi(:,9),'rx');
xlabel('Time (sec)')
ylabel('Calculated / Real(noisy) Alpha angle (rad)')
%legend('LISA 1', 'LISA 2', 'LISA 3')
title('LISA Alpha Angle vs. Time')
grid on

```

```

subplot(2,2,4)
plot(timespan, Gi(:,10),'b-', timespan, Gi(:,11), 'g--', timespan, Gi(:,12),'r-
.',...
timespan, Yi(:,10),'b*', timespan, Yi(:,11),'g>', timespan, Yi(:,12),'rx');
xlabel('Time (sec)')
ylabel('Calculated / Real(noisy) Delta angle (rad)')
%legend('LISA 1', 'LISA 2', 'LISA 3')
title('LISA Delta Angle vs. Time')
grid on
else
end

```

**InterSCD.m** (from Section 5.2.4 Simulation of Measurement Noise)

```

function [GiR, L13tot, L21tot, L32tot] = InterSCD(to, tf, I, X, Xref,
sigma_irange, sigma_irate, sigma_ciL, B, plt)
%*****
%Program to generate the interspacecraft data for LISA. The range and
range
%rate information of each spacecraft wrt each other
%
%INPUT : to, tf, I = initial time, final time and Interval (sec)
%       XO = initial condition for state (1x24)
%       sigma_irange, sigma_irate = noises associated with the
interspacecraft range and range rate
%       bir, birr = biases associated with the interspacecraft range and
range rate
%       plt = plot command, if 0 then "no plots", of 1 then "plot"
%
%OUTPUT : T_ISC = [pL13 pL21 pL32 pdotL13 pdotL21 pdotL32]
%           N_ISC = [Yi_5 Yl_5 Yq_5 Yi_6 Yl_6 Yq_6]
%           pL13 = range of LISA 1 wrt 3, with noise(and biases) = Yi_5
%           pL21 = range of LISA 2 wrt 1 = Yl_5
%           pL32 = range of LISA 3 wrt 2 = Yq_5
%           pdotL13 = range rate of LISA 1 wrt 3 = Yi_6

```

```

%      pdotL21 = range rate of LISA 2 wrt 1      = Y1_6
%      pdotL32 = range rate of LISA 3 wrt 2      = Yq_6
%
%*****
%---INITIALIZATION

if nargin < 10,
    plt = 0; %default to turn plot command off
end

%timespan =(to:I:tf)'; %with I sec intervals of time

%***PARSE TRUTH VECTOR(X) into Position & Velocity*****

R1 = X(1:3); %m (1x3)
V1 = X(4:6); %m/s (1x3)
R2 = X(7:9);
V2 = X(10:12);
R3 = X(13:15);
V3 = X(16:18);
RE = X(19:21);
VE = X(22:24);

%***PARSE REFERENCE VECTOR(Xref) into Position & Velocity

R1s = Xref(1:3); %m (1x3)
V1s = Xref(4:6); %m/s (1x3)
R2s = Xref(7:9);
V2s = Xref(10:12);
R3s = Xref(13:15);
V3s = Xref(16:18);
REs = Xref(19:21);
VEs = Xref(22:24);

%***CALCULATE THE ACTUAL MEASUREMENTS(I.e. Truth plus
%noise and biases)***
%RANGE

```

```

pL13 = norm(R1-R3); %m (1x1)
pL13_vec =(R1-R3);
pL21 = norm(R2-R1);
pL21_vec = (R2-R1);
pL32 = norm(R3-R2);
pL32_vec =(R3-R2);

%RANGE RATE
pdotL13 = (((R1(1)-R3(1))*(V1(1)- V3(1))) + ((R1(2)-R3(2))*...
(V1(2) - V3(2))) + ((R1(3) - R3(3))*(V1(3) - V3(3))))/pL13; %m/s

pdotL21 = (((R2(1)-R1(1))*(V2(1)- V1(1))) + ((R2(2)-R1(2))*...
(V2(2) - V1(2))) + ((R2(3) - R1(3))*(V2(3)-V1(3))))/pL21;

pdotL32 = (((R3(1)-R2(1))*(V3(1)- V2(1))) + ((R3(2)-R2(2))*...
(V3(2) - V2(2))) + ((R3(3) - R2(3))*(V3(3)-V2(3))))/pL32;

%***ADD THE NOISE TO THE TRUTH*****
% [GiR, YiR] = InterSCD2(to, ts(i), I, X, Xstar, S(i,13:15), S(i,16:18),
%B(4:6), 0);
%interspacecraft range stays in m
Yi_5 = pL13 + sigma_irange(1) + sigma_ciL(1) + B(1); %m
Y1_5 = pL21 + sigma_irange(2) + sigma_ciL(2) + B(2);
Yq_5 = pL32 + sigma_irange(3) + sigma_ciL(3) + B(3);

%keep in m/s
Yi_6 = (pdotL13) + sigma_irate(1); %mm/s
Y1_6 = (pdotL21) + sigma_irate(2);
Yq_6 = (pdotL32) + sigma_irate(3);

%***CALCULATE THE MEASUREMENTS i.e. Reference Observations
%RANGE PLUS ADD KNOWN BIASES TO CALCULATED
%MEASUREMENT
pL13s = norm(R1s-R3s);%+ B(1); %m; %m (1x1)

pL21s = norm(R2s-R1s);%+ B(2); %m;

```

```

pL32s = norm(R3s-R2s);%+ B(3); %m;

%RANGE RATE
pdotL13s = (((R1s(1)-R3s(1))*(V1s(1)- V3s(1))) + ((R1s(2)-R3s(2))*...
(V1s(2) - V3s(2))) + ((R1s(3) - R3s(3))*(V1s(3) - V3s(3))))/pL13s);
% m/s

pdotL21s = (((R2s(1)-R1s(1))*(V2s(1)- V1s(1))) + ((R2s(2)-R1s(2))*...
(V2s(2) - V1s(2))) + ((R2s(3) - R1s(3))*(V2s(3)-V1s(3))))/pL21s);

pdotL32s = (((R3s(1)-R2s(1))*(V3s(1)- V2s(1))) + ((R3s(2)-R2s(2))*...
(V3s(2) - V2s(2))) + ((R3s(3) - R2s(3))*(V3s(3)-V2s(3))))/pL32s);
% *****
YiR = [Yi_5' Yl_5' Yq_5' Yi_6' Yl_6' Yq_6']; %Real Measurements (1x6)
GiR = [pL13s' pL21s' pL32s' pdotL13s' pdotL21s' pdotL32s']; %Calculated
Measurements (1x6)

%---COMPUTE MEASUREMENT DEVIATIONS, yi (1x1), AT TIME ti
%LISA 1, 2 & 3 measurement deviations; measurement(Yi) - Reference
Measurement(Gi)
    yi_1 = YiR(1) - GiR(1); % (1x1), range measurement deviations
    yl_1 = YiR(2) - GiR(2);
    yq_1 = YiR(3) - GiR(3);

    yi_2 = YiR(4) - GiR(4); %range rate measurement deviations
    yl_2 = YiR(5) - GiR(5);
    yq_2 = YiR(6) - GiR(6);

    %Total Measurement deviations
    L13tot = [yi_1; yi_2]; % (2x1)
    L21tot = [yl_1; yl_2]; % (2x1)
    L32tot = [yq_1; yq_2]; % (2x1)

%PLOTS (x,y,z)
if plt == 1,
figure
subplot(2,1,1)

```

```

plot(timespan, GiR(:,1),'b-', timespan, GiR(:,2),'g--', timespan, GiR(:,3), 'r-
.';...
timespan, YiR(:,1),'b*', timespan, YiR(:,2),'g>', timespan, YiR(:,3),'rx');
xlabel('Time (sec)')
ylabel('Calculated / Real(noisy) Range(m)')
legend('Leg 13', 'Leg 21', 'Leg 32','Leg 13', 'Leg 21', 'Leg 32')
title('LISA Interspacecraft Range vs. Time')
grid on
subplot(2,1,2)
plot(timespan, GiR(:,4),'b-', timespan, GiR(:,5), 'g--', timespan, GiR(:,6),'r-
.';...
timespan, YiR(:,4),'b*', timespan, YiR(:,5),'g>', timespan, YiR(:,6),'rx');
xlabel('Time (sec)')
ylabel('Calculated / Real(noisy) Range Rate(m/sec)')
title('LISA Interspacecraft Range Rate vs. Time')
grid on
else
end

```

#### **LISA\_WBLS.m** (from Section 5.4 Orbit Accuracy)

```

function [Xest, ABS, REL, RMS, var, PU_L1, PU_L2, PU_L3, err, erv,
niter, Pcov] =MCLISA_bp(to_0, tf, I, Xrefstart_0, Xo_0, R1, R2, R3, S, B,
plt, obs)
% *****
% [Xest, DXr, DXv, DLr, DLv] = LISA_bp(to, tf, I, R1, R2, R3, AP, Po,
xo_bar, %plt,obs)
%
% This function is a general Batch Processor that takes in the
% absolute spacecraft data (range, range rate, right ascension and
declination)
% and relative spacecraft data (interspacecraft range and range rate)
% It can also be adapted to take in a priori information and weights. Error
% analysis will also be done and graphed.

```

```

%
%INPUTS : to_0 = initial starting time of data (sec)
%   tf = final time of data (sec)
%   I = time interval to space measurements (sec)
%   Xrefstart_0 = The Initial Guess Vector, or Reference
%       Start Vector (1x18), may be off by some errorstart from the
%       Xo_0
%   Xo_0 = The Nominal Vector (aka Truth) (1x18)
%   R1, R2, R3 = each are 1xn array of weights associated with
%       each observation type for each LISA. Depends on size of
%       observation case
%   S = sigmas associated with each measurement
%   B = read in biases
%   plt= 0, "no" plot; =1, "yes" plot command is on
% Observation Cases:
%   obs = 0, DSN & VLBI
%       = 1, DSN, VLBI & RELATIVE
%       = 2, DSN only
%       = 3, DSN & Relative
%
%OUTPUTS: Xest = The final Batch Processor estimate of the state
%   ABS = The Absolute Error in position & velocity of the LISA
%   constellation
%   REL = The Relative Error in position & velocity of the LISA
%   constellation
%   RMS = The Root-Mean-Square of each LISA
%   var = The Sample Variance of each LISA
%   PU_L1, PU_L2, PU_L3 = Parameter Uncertainities
%   err = Position Error in each LISA
%   erv = Velocity Error in each LISA
%
%CALLS: LISA_EOMderivs.m, LISA_stmderive.m, RealObs.m,
InterSCD.m, %H_partials.m
%
%REFERENCE: Fig. 4.6.1 (pp. 196-197) from B.Tapley's "Statistical Orbit
%Determination"

```

```

% *****
%*****

%---0. INITIALIZATION
format long e

if nargin < 12
    obs = 0; %default, takes into account all absolute (DSN & VLBI)
    observation types
end

if nargin < 11
    plt = 0; %default no plots
end

%---SET UP TOLERANCE FOR CONVERGENCE
switch(obs)
    case 0, %DSN & VLBI
        tol = .03;
    case 1, %DSN, VLBI & RELATIVE
        tol = .03;
    case 2, %DSN
        tol = .3; % .3; % .2; %case 2 won't converge with lower like .09
    case 3, %DSN & RELATIVE
        tol = .03;
    otherwise,
        error('Invalid case (0 <= obs <= 3)');
end

RMSold = [1 1 1];
dRMS = 1;

niter = 0; %set up number of iterations
to = to_0; %initial time
ts = to:I:tf; %(1xn) set up timespan

Xo = Xo_0;

```

```

Xtru = Xo; %Nominal (or Truth) Vector
Xk = Xrefstart_0; %Initial Guess Vector (i.e. Reference Vector)
Xrefstart = Xk; %Will get updated from initial with each iteration in the
program

%---A. OUTER LOOP
while dRMS >= tol

    %---INITIAL CONDITION FOR THE STATE
    Xstaro = Xrefstart; % (1x24) I.C. for integration = Xstar(ti-1)
    to = to_0;
    Xo = Xo_0;
    diff = Xo - Xo_0;

    %---DEFINE & GIVE THE INITIAL CONDITION FOR THE STATE
%TRANSITION MATRIX
%(STM) INTEGRATOR
stmi = eye(24); % (24x24), identity matrix
rstmi = reshape(stmi,576,1); % (576x1)
Xstmi = [Xstaro'; rstmi]; % (600x1) I.C. for STM integration

%---DEFINE THE M = Information Matrix, inverse of the covariance
%      & N = Reduced Residual Vector
%Both matrices are different depending on if a priori knowledge
switch(obs)
    case 0, %DSN & VLBI
        M1 = zeros(6,6); % (6x6) position & velocity of LISA 1
        N1 = zeros(6,1); % (6x1)

        M2 = zeros(6,6); %LISA 2
        N2 = zeros(6,1);

        M3 = zeros(6,6); %LISA 3
        N3 = zeros(6,1);

    case 1, %DSN, VLBI & RELATIVE
        M = zeros(18,18); %LISA 1, 2 & 3 Postion & Velocity (18x18)

```

```

        N = zeros(18,1); % (18x1)

    case 2, %DSN
        M1 = zeros(6,6); % (6x6) position & velocity of LISA 1
        N1 = zeros(6,1); % (6x1)

        M2 = zeros(6,6); %LISA 2
        N2 = zeros(6,1);

        M3 = zeros(6,6); %LISA 3
        N3 = zeros(6,1);

    case 3, %DSN & RELATIVE
        M = zeros(18,18); %LISA 1, 2 & 3 Postion & Velocity (18x18)
        N = zeros(18,1); % (18x1)

    otherwise,
        error('Invalid case (0 <= obs <= 3)');
end

RMS1 = zeros(1);
RMS2 = zeros(1);
RMS3 = zeros(1);

var1 = zeros(1);
var2 = zeros(1);
var3 = zeros(1);

%---B. INNER LOOP
for i = 1:length(ts), %loop over number of measurements for length of
timespan

    %---READ IN TIME (1x1)
    ts(i); %each measurement corresponds to a time in span

    %---INTEGRATE STATE TRANSITION MATRIX, REFERENCE
    TRAJECTORY, & TRUTH

```

```

%TRAJECTORY FROM to->ti
%NOTE: STM & REF. TRAJECTORY INTEGRATED ON THE
NOMINAL ORBIT X*
if i==1,
    Xstm = Xstm_i; %(1x600) no integration required for zero time
change
    Sf = Xstm(end,:); %(1x600)
    Xstar=Xstaro; %(1x24)
    X = Xo;
else
    [t, Xstm] = ode45('LISA_stmderive', [to ts(i)], Xstm_i,
odeset('RelTol', 1e-8,'AbsTol',1e-8));
    [t, Xstar] = ode45('LISA_EOMderivs', [to ts(i)], Xstaro,
odeset('RelTol', 1e-8,'AbsTol',1e-8));
    [t, X] = ode45('LISA_EOMderivs', [to ts(i)], Xo, odeset('RelTol', 1e-
8,'AbsTol',1e-8));

    Sf = Xstm(end,:); %(1x600)
    Xstar = Xstar(end, :); %1x24 Also, Xstar = Sf(1:24)
    X = X(end, :); %1x24
end

%--Xstm_i=Xstar plus the reshaped stm matrix as the I.C. for stmderive
stm = reshape(Sf(25:600),24,24); %(24 x 24)the reshaped State
Transition Matrix is columns 25:600 from Sf

%---CALL RealObs.m & InterSCD.m to READ IN Measurement
Deviations
[Gi(i,:), L1tot(:,i), L2tot(:,i), L3tot(:,i)] = RealObs(to, ts(i), I, X, Xstar,
S(i,1:3), S(i,4:6), S(i,7:9), S(i,10:12), S(i,19:21), S(i,22:24), B(1:3), 0);
[GiR(i,:), L13tot(:,i), L21tot(:,i), L32tot(:,i)] = InterSCD(to, ts(i), I, X,
Xstar, S(i,13:15), S(i,16:18), S(i,25:27), B(4:6), 0);

%Total Measurement deviations
switch(obs)
case 0 %DSN & VLBI

```

```

    yi_tot(:,i) = [L1tot(:,i)]; %(4xi)
    yl_tot(:,i) = [L2tot(:,i)]; %(4xi)
    yq_tot(:,i) = [L3tot(:,i)]; %(4xi)

case 1 %DSN, VLBI & RELATIVE

    yi_tot(:,i) = [L1tot(:,i); L13tot(:,i); L2tot(:,i); L21tot(:,i);
L3tot(:,i); L32tot(:,i)]; %(18xi)

case 2 %DSN only (range and range rate)

    yi_tot(:,i) = [L1tot(1); L1tot(2)]; %(2xi)
    yl_tot(:,i) = [L2tot(1); L2tot(2)]; %(2xi)
    yq_tot(:,i) = [L3tot(1); L3tot(2)]; %(2xi)

case 3, %DSN & RELATIVE

    yi_tot(:,i) = [L1tot(1:2,i); L13tot(:,i); L2tot(1:2,i); L21tot(:,i);
L3tot(1:2,i); L32tot(:,i)]; %(12xi)

otherwise
    error('Invalid case (0 <= obs <= 3)');
end

%---COMPUTE Hi_tilda & Hi
%---H_tilda matrix, Observation Sensitivity (partial derivatives of the
observations
%over the State),or Observation-State Mapping Matrix, G =
observations
switch(obs)
case 0, %DSN & VLBI
    [Hi_tilda(:,i), Hl_tilda(:,i), Hq_tilda(:,i)] = H_partials7(Gi(i,:),
GiR(i,:), Xstar, 0);

    %CALCULATE THE OBSERVATION SENSITIVITY
MATRIX OF EACH LISA
    Hi(:,i) = Hi_tilda(:,1:6,i)*stm(1:6,1:6); %(4x6)

```

```

HL(:, :, i) = HL_tilda(:, 7:12, i)*stm(7:12, 7:12);

Hq(:, :, i) = Hq_tilda(:, 13:18, i)*stm(13:18, 13:18);

%---READ IN Observation Noise Covariances (Ri)
%construct as block diagonal matrices
%---Weighting Matrix
%NOTE: W = inv(Ri), Construct W matrices
W1 = inv(diag(R1)); % (4x4)
W2 = inv(diag(R2));
W3 = inv(diag(R3));
%return
%---Scaled Weights
W1sc = inv(diag(R1));
W2sc = inv(diag(R2));
W3sc = inv(diag(R3));

case 1, %DSN, VLBI, RELATIVE
    [Hi_tilda(:, :, i), HL_tilda(:, :, i), Hq_tilda(:, :, i)] = H_partials7(Gi(i, :),
    GiR(i, :), Xstar, 1);

    Hi(:, :, i) = Hi_tilda(:, :, i)*stm(1:18, 1:18); % (18x18)

    %---Weighting Matrix
    W1 = inv(diag(R1)); % (18x18)

    %---Scaled Weights
    W1sc = inv(diag(R1));

case 2, %DSN only (range and range rate)
    [Hi_tilda(:, :, i), HL_tilda(:, :, i), Hq_tilda(:, :, i)] = H_partials7(Gi(i, :),
    GiR(i, :), Xstar, 2);

    %CALCULATE THE OBSERVATION SENSITIVITY
    MATRIX OF EACH LISA
    Hi(:, :, i) = Hi_tilda(:, 1:6, i)*stm(1:6, 1:6); % (2x6)

```

```

HL(:, :, i) = HL_tilda(:, 7:12, i)*stm(7:12, 7:12);

Hq(:, :, i) = Hq_tilda(:, 13:18, i)*stm(13:18, 13:18);

%---Weighting Matrix
W1 = inv(diag(R1)); % (4x4)
W2 = inv(diag(R2));
W3 = inv(diag(R3));
%---Scaled Weights
W1sc = inv(diag(R1));
W2sc = inv(diag(R2));
W3sc = inv(diag(R3));

case 3, %DSN & RELATIVE
    [Hi_tilda(:, :, i), HL_tilda(:, :, i), Hq_tilda(:, :, i)] = H_partials7(Gi(i, :),
    GiR(i, :), Xstar, 3);

    Hi(:, :, i) = Hi_tilda(:, :, i)*stm(1:18, 1:18); % (18x18)

    %---Weighting Matrix
    W1 = inv(diag(R1)); % (12x12)

    %---Scaled Weights
    W1sc = inv(diag(R1));
    otherwise,
        error('Invalid case (0 <= obs <= 3)');
    end

%ACCUMULATE CURRENT OBSERVATION
switch(obs)
case 0, %DSN & VLBI
    SC=diag([1 1 1 1e-6 1e-6 1e-6]); %Scaling Matrix

    M1 = M1 + ((Hi(:, :, i)*SC)*W1sc*(Hi(:, :, i)*SC)); % 6x6, 2-D
    N1 = N1 + ((Hi(:, :, i)*SC)*W1sc*yi_tot(:, i)); % 6x1, 2-D

```

```

M2 = M2 + ((Hl(:,i)*SC)*W2sc*(Hl(:,i)*SC));
N2 = N2 + ((Hl(:,i)*SC)*W2sc*y1_tot(:,i));

M3 = M3 + ((Hq(:,i)*SC)*W3sc*(Hq(:,i)*SC));
N3 = N3 + ((Hq(:,i)*SC)*W3sc*yq_tot(:,i));

%*****
**
RMS1 = RMS1 + (yi_tot(:,i)*W1sc*yi_tot(:,i)); % 1x1, Weighted
RMS
RMS2 = RMS2 + (y1_tot(:,i)*W2sc*y1_tot(:,i));
RMS3 = RMS3 + (yq_tot(:,i)*W3sc*yq_tot(:,i));
RMSm = [RMS1 RMS2 RMS3];

var1 = var1 + yi_tot(:,i)*W1sc*yi_tot(:,i);
var2 = var2 + y1_tot(:,i)*W2sc*y1_tot(:,i);
var3 = var3 + yq_tot(:,i)*W3sc*yq_tot(:,i);
varm = [var1 var2 var3];

case 1 %DSN, VLBI & RELATIVE
SC = diag([1 1 1 1e-6 1e-6 1e-6 1 1 1 1e-6 1e-6 1e-6...
1 1 1 1e-6 1e-6 1e-6]);
M = M + ((Hi(:,i)*SC)*(W1sc)*(Hi(:,i)*SC)); % 18x18, 2-D
N = N + ((Hi(:,i)*SC)*(W1sc)*yi_tot(:,i)); % 18x1, 2-D

%*****
**
RMS1 = RMS1 + (yi_tot(1:6,i)*(W1sc(1:6,1:6))*yi_tot(1:6,i));
% 1x1, Weighted RMS
RMS2 = RMS2 +
(yi_tot(7:12,i)*(W1sc(7:12,7:12))*yi_tot(7:12,i)); % 1x1, Weighted RMS
RMS3 = RMS3 +
(yi_tot(13:18,i)*(W1sc(13:18,13:18))*yi_tot(13:18,i)); % 1x1, Weighted
RMS
RMSm = [RMS1 RMS2 RMS3];

var1 = var1 + yi_tot(1:6,i)*(W1sc(1:6,1:6))*yi_tot(1:6,i);
var2 = var2 + yi_tot(7:12,i)*(W1sc(7:12,7:12))*yi_tot(7:12,i);
var3 = var3 +
yi_tot(13:18,i)*(W1sc(13:18,13:18))*yi_tot(13:18,i);
varm = [var1 var2 var3];

case 2 %DSN
SC=diag([1 1 1 1e-6 1e-6 1e-6]);
M1 = M1 + ((Hi(:,i)*SC)*(W1sc)*(Hi(:,i)*SC)); % 6x6, 2-D
N1 = N1 + ((Hi(:,i)*SC)*(W1sc)*yi_tot(:,i)); % 6x1, 2-D

M2 = M2 + ((Hl(:,i)*SC)*(W2sc)*(Hl(:,i)*SC));
N2 = N2 + ((Hl(:,i)*SC)*(W2sc)*y1_tot(:,i));

M3 = M3 + ((Hq(:,i)*SC)*(W3sc)*(Hq(:,i)*SC));
N3 = N3 + ((Hq(:,i)*SC)*(W3sc)*yq_tot(:,i));

%*****
**
RMS1 = RMS1 + (yi_tot(:,i)*(W1sc)*yi_tot(:,i)); % 1x1,
Weighted RMS
RMS2 = RMS2 + (y1_tot(:,i)*(W1sc)*y1_tot(:,i));
RMS3 = RMS3 + (yq_tot(:,i)*(W1sc)*yq_tot(:,i));
RMSm = [RMS1 RMS2 RMS3];

var1 = var1 + yi_tot(:,i)*W1sc*yi_tot(:,i);
var2 = var2 + y1_tot(:,i)*W2sc*y1_tot(:,i);
var3 = var3 + yq_tot(:,i)*W3sc*yq_tot(:,i);
varm = [var1 var2 var3];

case 3, %DSN & RELATIVE
SC=diag([1 1 1 1e-6 1e-6 1e-6 1 1 1 1e-6 1e-6 1e-6...
1 1 1 1e-6 1e-6 1e-6]);
M = M + ((Hi(:,i)*SC)*(W1sc)*(Hi(:,i)*SC)); % 18x18, 2-D
N = N + ((Hi(:,i)*SC)*(W1sc)*yi_tot(:,i)); % 18x1, 2-D

```



```

% *****
RMS1 = RMS1 + (yi_tot(1:4,i)*(W1sc(1:4,1:4))*yi_tot(1:4,i));
% 1x1, % Weighted RMS
RMS2 = RMS2 + (yi_tot(5:8,i)*(W1sc(5:8,5:8))*yi_tot(5:8,i));
% 1x1, % Weighted RMS
RMS3 = RMS3 +
(yi_tot(9:12,i)*(W1sc(9:12,9:12))*yi_tot(9:12,i)); % 1x1, Weighted RMS
RMSm = [RMS1 RMS2 RMS3];

var1 = var1 + yi_tot(1:4,i)*W1sc(1:4,1:4)*yi_tot(1:4,i);
var2 = var2 + yi_tot(5:8,i)*W1sc(5:8,5:8)*yi_tot(5:8,i);
var3 = var3 + yi_tot(9:12,i)*W1sc(9:12,9:12)*yi_tot(9:12,i);
varm = [var1 var2 var3];

otherwise
error('Invalid case (0 <= obs <= 3)');
end
if ts(i) < tf,
if ts(i) == ts(1)
to = ts(i); % time ti becomes to
i = i + 1;
Xstaro = Xstaro; % (1x24) the I.C. @ i=1 becomes the I.C. for
next % reading
Xstmi = Xstmi; % (600x1), STM becomes the I.C. for next
reading
Xo = Xo; % the I.C. @ i=1 becomes the I.C. for the next reading
else
to = ts(i); % time ti becomes to
i = i + 1;
Xstaro = Xstar; % (1x24) integrated trajectory becomes the I.C.
for next % reading
Xstmi = Sf; % Sf = (600x1), integrated STM becomes the I.C. for
next % reading
Xo = X; % integrated truth becomes the I.C. for the next reading
end

```

```

%---GO BACK TO THE BEGINNING OF B TO READ NEXT
%OBSERVATION---%

elseif ts(i) >= tf

disp('---All measurements in batch have been read---');

end % PROCEED TO STEP C

end % END OF INNER LOOP B, when all measurements in the batch are
read

disp('done with inner loop B');

% *****
%OBSERVABILITY & MATRIX RANK TEST
%IF FULL RANK & OBSERVABILITY IS GREATER THAN ZERO
(i.e. not %singular
% (det(M) is not equal to zero), to be singular, det(M)=0) then M is
%invertible and observable
%A test for the existence of exact solutions is rank(M) = rank([M N]), if
%not equal then the answer is a LS solution, and is not exact

% *****
%---C. SOLVE THE NORMAL EQUATIONS
%Alternative way to do the inversion of the Normal equations.....
%Here is the straight up way, must pass the observability and matrix
%rank test above to be able to invert
% *****
l = (length(ts)); % (1:length(ts)) % length of batch, i
p = length(yi_tot(:,end)); % length of p-dimensional set of observation
(usu. p<n)
m = p*i; % total # of observations, where m >> n

switch(obs)

```

```
case 0 %DSN & VLBI
```

```
rcond(M1);  
rcond(M2);  
rcond(M3);  
rank(M1);  
rank([M1 N1]);  
rank(M2);  
rank([M2 N2]);  
rank(M3);  
rank([M3 N3]);
```

```
P1 = inv(M1);  
xhat1 = SC*P1*N1;
```

```
P2 = inv(M2);  
xhat2 = SC*P2*N2;
```

```
P3 = inv(M3);  
xhat3 = SC*P3*N3;
```

```
n_xhat1 = norm(xhat1);  
n_xhat2 = norm(xhat2);  
n_xhat3 = norm(xhat3);  
n_xcorr = [n_xhat1 n_xhat2 n_xhat3];
```

```
xhat = [xhat1 xhat2 xhat3];  
n_xhat = norm(xhat);
```

```
%***RMS & Sample Variance  
RMSm;  
RMS = ([RMSm]./m).^(1/2); %(1x3)  
var = (1/(m-6)).*([varm]); %(1x3)
```

```
case 1 %DSN, VLBI & RELATIVE
```

```
rcond(M);  
rank(M);  
rank([M N]);
```

```
P = inv(M);
```

```
xhat = SC*P*N;  
n_xhat = norm(xhat);
```

```
n_xhat1 = norm(xhat(1:6));  
n_xhat2 = norm(xhat(7:12));  
n_xhat3 = norm(xhat(13:18));
```

```
n_xcorr = [n_xhat1 n_xhat2 n_xhat3];
```

```
%***RMS & Sample Variance
```

```
RMS = ([RMSm]./m).^(1/2); %(1x3)
```

```
var = (1/(m-6)).*([varm]); %(1x3) var = (1/(m-18)).*([varm]);  
%(1x3)
```

```
case 2 %DSN
```

```
rcond(M1);  
rcond(M2);  
rcond(M3);  
rank(M1);  
rank([M1 N1]);  
rank(M2);  
rank([M2 N2]);  
rank(M3);  
rank([M3 N3]);
```

```
P1 = inv(M1);  
xhat1 = SC*P1*N1;
```

```
P2 = inv(M2);  
xhat2 = SC*P2*N2;
```

```
P3 = inv(M3);  
xhat3 = SC*P3*N3;
```

```

n_xhat1 = norm(xhat1);
n_xhat2 = norm(xhat2);
n_xhat3 = norm(xhat3);
n_xcorr = [n_xhat1 n_xhat2 n_xhat3];

xhat = [xhat1 xhat2 xhat3];
n_xhat = norm(xhat);

%***RMS & Sample Variance
RMS = ([RMSm]./m).^(1/2); % (1x3)
var = (1/(m-6)).*([varm]); % (1x3)

case 3, %DSN & RELATIVE
    rcond(M);
    rank(M);
    rank([M N]);

    P = inv(M);

    xhat = SC*P*N;
    n_xhat = norm(xhat);

    n_xhat1 = norm(xhat(1:6));
    n_xhat2 = norm(xhat(7:12));
    n_xhat3 = norm(xhat(13:18));

    n_xcorr = [n_xhat1 n_xhat2 n_xhat3];

    %***RMS & Sample Variance
    RMS = ([RMSm]./m).^(1/2); % (1x3)
    var = (1/(m-6)).*([varm]); % (1x3)
    var = (1/(m-18)).*([varm]); % (1x3)
otherwise
    error('Invalid case (0 <= obs <= 3)')
end

%HAS PROCESS CONVERGED?

```

```

niter = niter + 1;
%dRMS = mean((abs(RMSold - RMS)));
dRMS = mean((abs(RMSold - RMS))./RMSold);

if dRMS >= tol, %THEN UPDATE REFERENCE STATE & ITERATE
    switch(obs)
        case 0 %DSN & VLBI
            Xhat1 = Xrefstart(1:6) + xhat1';
            Xhat2 = Xrefstart(7:12) + xhat2';
            Xhat3 = Xrefstart(13:18) + xhat3';

            Xrefstart = [Xhat1 Xhat2 Xhat3 Xtru(1,19:24)]; %Updated
reference/nominal trajectory for next iteration
            RMSold = RMS;

        case 1 %DSN, VLBI & RELATIVE
            Xhat = Xrefstart(1:18) + xhat';

            Xrefstart = [Xhat Xtru(1,19:24)]; %Updated reference/nominal
trajectory for next iteration
            RMSold = RMS;

        case 2 %DSN
            Xhat1 = Xrefstart(1:6) + xhat1';
            Xhat2 = Xrefstart(7:12) + xhat2';
            Xhat3 = Xrefstart(13:18) + xhat3';

            Xrefstart = [Xhat1 Xhat2 Xhat3 Xtru(1,19:24)]; %Updated
reference/nominal trajectory for next iteration
            RMSold = RMS;

        case 3, %DSN & RELATIVE
            Xhat = Xrefstart(1:18) + xhat';

            Xrefstart = [Xhat Xtru(1,19:24)]; %Updated reference/nominal
trajectory for next iteration
            RMSold = RMS;
    end
end

```

```

        otherwise
            error('Invalid case (0 <= obs <= 3)');
        end

%---GO BACK TO A = BEGINNING OF OUTER LOOP with new
values Xstar and
%xo_bar--%

else dRMS < tol, %THE PROCESS HAS CONVERGED
    switch(obs)
        case 0 %DSN & VLBI
            Xhat1 = Xrefstart(1:6) + xhat1'; %updated reference state for
LISA 1
            Xhat2 = Xrefstart(7:12) + xhat2'; %for LISA 2
            Xhat3 = Xrefstart(13:18) + xhat3'; %for LISA 3

            Xup = [Xhat1 Xhat2 Xhat3]; % Updated reference/nominal
trajectory

            Xest = [Xup Xtru(1,19:24)]; % give last Xhat as the resulting
estimate from batch LS

            case 1 %DSN, VLBI & RELATIVE
                Xup = Xrefstart(1:18) + xhat'; %updated reference state for LISA
1,2,& 3

                Xest = [Xup Xtru(1,19:24)]; %give last Xhat as the resulting
estimate from batch LS

            case 2 %DSN
                Xhat1 = Xrefstart(1:6) + xhat1'; %updated reference state for
LISA 1
                Xhat2 = Xrefstart(7:12) + xhat2'; %for LISA 2
                Xhat3 = Xrefstart(13:18) + xhat3'; %for LISA 3

```

```

            Xup = [Xhat1 Xhat2 Xhat3]; % Updated reference/nominal
trajectory

            Xest = [Xup Xtru(1,19:24)]; % give last Xhat as the resulting
estimate from batch LS

            case 3, %DSN & RELATIVE
                Xup = Xrefstart(1:18) + xhat'; %updated reference state for LISA
1,2,& 3

                Xest = [Xup Xtru(1,19:24)]; %give last Xhat as the resulting
estimate from batch LS
            otherwise
                error('Invalid case (0 <= obs <= 3)');
            end
        end %CAN END LOOPS
        %disp('done with outer loop A')
    end %END

%---D. ERROR ANALYSIS
switch(obs)
    case 0 %DSN & VLBI
        for j = 1:length(ts),
            e1(:,j) = yi_tot(:,j) - ([Hi(:,j)]*xhat1); %nxj best estimate of
observation error %nx1, individually
            e2(:,j) = y1_tot(:,j) - ([Hl(:,j)]*xhat2);
            e3(:,j) = yq_tot(:,j) - ([Hq(:,j)]*xhat3);
        end
        obserr = [mean(e1,2) mean(e2,2) mean(e3,2)]; %nx3

        %Unscaled Covariance
        SC=diag([1 1 1 1e-6 1e-6 1e-6]); %Scaling Matrix
        SC2 = (SC'*SC);

        P1_us = (SC2*P1); %(6x6)
        P2_us = (SC2*P2);
        P3_us = (SC2*P3);

```

```

Pcov = [diag(P1_us)' diag(P2_us)' diag(P3_us)']; % (1x18)

sPcov = [sqrt(Pcov)'];
% Parameter Uncertainties
PU_L1 = [sqrt(var(1)*abs(P1_us(1,1))); sqrt(var(1)*abs(P1_us(2,2)));
sqrt(var(1)*abs(P1_us(3,3)));...
sqrt(var(1)*abs(P1_us(4,4))); sqrt(var(1)*abs(P1_us(5,5)));
sqrt(var(1)*abs(P1_us(6,6)))]]; % (6x1) for LISA 1 Position & Velocity

PU_L2 = [sqrt(var(2)*abs(P2_us(1,1))); sqrt(var(2)*abs(P2_us(2,2)));
sqrt(var(2)*abs(P2_us(3,3)));...
sqrt(var(2)*abs(P2_us(4,4))); sqrt(var(2)*abs(P2_us(5,5)));
sqrt(var(2)*abs(P2_us(6,6)))]]; % (6x1) for LISA 2 Position & Velocity

PU_L3 = [sqrt(var(3)*abs(P3_us(1,1))); sqrt(var(3)*abs(P3_us(2,2)));
sqrt(var(3)*abs(P3_us(3,3)));...
sqrt(var(3)*abs(P3_us(4,4))); sqrt(var(3)*abs(P3_us(5,5)));
sqrt(var(3)*abs(P3_us(6,6)))]]; % (6x1) for LISA 3 Position & Velocity

PU = [PU_L1 PU_L2 PU_L3]; % 6x3

% *** Position Errors
err1r = norm(xhat1(1:3));
err2r = norm(xhat2(1:3));
err3r = norm(xhat3(1:3));

err = [err1r err2r err3r];

% *** Velocity Errors
err1v = norm(xhat1(4:6));
err2v = norm(xhat2(4:6));
err3v = norm(xhat3(4:6));
erv = [err1v err2v err3v];

case 1 % DSN, VLBI & RELATIVE
for j = 1:length(ts),

```

```

    e1(:,j) = yi_tot(:,j) - ([Hi(:,j)]*xhat); % nxj best estimate of
observation error % nx1, individually
end
obserr = [e1];

% Unscaled Covariances
SC = diag([1 1 1e-6 1e-6 1e-6 1 1 1 1e-6 1e-6 1e-6...
1 1 1 1e-6 1e-6 1e-6]);
SC2 = (SC'*SC);

P_us = (SC2*P); % (18x18)
Pcov = [diag(P_us)'];

% Parameter Uncertainties
PU_L1 = [sqrt(var(1)*abs(P_us(1,1))); sqrt(var(1)*abs(P_us(2,2)));
sqrt(var(1)*abs(P_us(3,3)));...
sqrt(var(1)*abs(P_us(4,4))); sqrt(var(1)*abs(P_us(5,5)));
sqrt(var(1)*abs(P_us(6,6)))]]; % (6x1) for LISA 1 Position & Velocity

PU_L2 = [sqrt(var(2)*abs(P_us(7,7))); sqrt(var(2)*abs(P_us(8,8)));
sqrt(var(2)*abs(P_us(9,9)));...
sqrt(var(2)*abs(P_us(10,10))); sqrt(var(2)*abs(P_us(11,11)));
sqrt(var(2)*abs(P_us(12,12)))]]; % (6x1) for LISA 2 Position & Velocity

PU_L3 = [sqrt(var(3)*abs(P_us(13,13)));
sqrt(var(3)*abs(P_us(14,14))); sqrt(var(3)*abs(P_us(15,15)));...
sqrt(var(3)*abs(P_us(16,16))); sqrt(var(3)*abs(P_us(17,17)));
sqrt(var(3)*abs(P_us(18,18)))]]; % (6x1) for LISA 3 Position & Velocity

% *** Position Errors
err = [norm(xhat(1:3)) norm(xhat(7:9)) norm(xhat(13:15))];

% *** Velocity Errors
erv = [norm(xhat(4:6)) norm(xhat(10:12)) norm(xhat(16:18))];

case 2 % DSN
for j = 1:length(ts),

```

```

    e1(:,j) = yi_tot(:,j) - ([Hi(:,j)]*xhat1); %nxj best estimate of
observation error %nx1, individually
    e2(:,j) = yl_tot(:,j) - ([Hl(:,j)]*xhat2);
    e3(:,j) = yq_tot(:,j) - ([Hq(:,j)]*xhat3);
end
obserr = [e1 e2 e3];

% Unscaled Covariance
SC = diag([1 1 1 1e-6 1e-6 1e-6]);
SC2 = (SC'*SC);

P1_us = (SC2*P1); % (6x6)

P2_us = (SC2*P2);

P3_us = (SC2*P3);

Pcov = [diag(P1_us)' diag(P2_us)' diag(P3_us)'];

% Parameter Uncertainties
PU_L1 = [sqrt(var(1)*abs(P1_us(1,1))); sqrt(var(1)*abs(P1_us(2,2)));
sqrt(var(1)*abs(P1_us(3,3)));...
sqrt(var(1)*abs(P1_us(4,4))); sqrt(var(1)*abs(P1_us(5,5)));
sqrt(var(1)*abs(P1_us(6,6))]; % (6x1) for LISA 1 Position & Velocity

PU_L2 = [sqrt(var(2)*abs(P2_us(1,1))); sqrt(var(2)*abs(P2_us(2,2)));
sqrt(var(2)*abs(P2_us(3,3)));...
sqrt(var(2)*abs(P2_us(4,4))); sqrt(var(2)*abs(P2_us(5,5)));
sqrt(var(2)*abs(P2_us(6,6))]; % (6x1) for LISA 2 Position & Velocity

PU_L3 = [sqrt(var(3)*abs(P3_us(1,1))); sqrt(var(3)*abs(P3_us(2,2)));
sqrt(var(3)*abs(P3_us(3,3)));...
sqrt(var(3)*abs(P3_us(4,4))); sqrt(var(3)*abs(P3_us(5,5)));
sqrt(var(3)*abs(P3_us(6,6))]; % (6x1) for LISA 3 Position & Velocity

% *** Position Errors
err1r = norm(xhat1(1:3));

```

```

err2r = norm(xhat2(1:3));
err3r = norm(xhat3(1:3));

err = [err1r err2r err3r];

% *** Velocity Errors
err1v = norm(xhat1(4:6));
err2v = norm(xhat2(4:6));
err3v = norm(xhat3(4:6));

erv = [err1v err2v err3v];

case 3, % DSN & RELATIVE
for j = 1:length(ts),
    e1(:,j) = yi_tot(:,j) - ([Hi(:,j)]*xhat); %nxj best estimate of
observation error %nx1, individually
end
obserr = [e1];

% Unscaled Covariances
SC = diag([1 1 1 1e-6 1e-6 1e-6 1 1 1 1e-6 1e-6 1e-6...
1 1 1 1e-6 1e-6 1e-6]);
SC2 = (SC'*SC);

P_us = (SC2*P); % (18x18)
Pcov = [diag(P_us)'];
% Parameter Uncertainties
PU_L1 = [sqrt(var(1)*abs(P_us(1,1))); sqrt(var(1)*abs(P_us(2,2)));
sqrt(var(1)*abs(P_us(3,3)));...
sqrt(var(1)*abs(P_us(4,4))); sqrt(var(1)*abs(P_us(5,5)));
sqrt(var(1)*abs(P_us(6,6))]; % (6x1) for LISA 1 Position & Velocity

PU_L2 = [sqrt(var(2)*abs(P_us(7,7))); sqrt(var(2)*abs(P_us(8,8)));
sqrt(var(2)*abs(P_us(9,9)));...
sqrt(var(2)*abs(P_us(10,10))); sqrt(var(2)*abs(P_us(11,11)));
sqrt(var(2)*abs(P_us(12,12))]; % (6x1) for LISA 2 Position & Velocity

```

```

    PU_L3 = [sqrt(var(3)*abs(P_us(13,13)));
sqrt(var(3)*abs(P_us(14,14))); sqrt(var(3)*abs(P_us(15,15)));...
    sqrt(var(3)*abs(P_us(16,16))); sqrt(var(3)*abs(P_us(17,17)));
sqrt(var(3)*abs(P_us(18,18))]; % (6x1) for LISA 3 Position & Velocity

%***Position Errors
err = [norm(xhat(1:3)) norm(xhat(7:9)) norm(xhat(13:15))];

%***Velocity Errors
erv = [norm(xhat(4:6)) norm(xhat(10:12)) norm(xhat(16:18))];

otherwise
    error('Invalid case (0 <= obs <= 6)');
end

%***ESTIMATE
ERROR*****
fuzz = Xk - Xest;
nfuzz = norm(fuzz);
Xerror = Xest-Xtru;
nXerror = norm(Xerror);
nXest = norm(Xest);
nXtru = norm(Xtru);

%***ABSOLUTE OD
ACCURACY*****
DXr = [norm(Xest(1:3)-Xtru(1:3)), norm(Xest(7:9)-Xtru(7:9)),
norm(Xest(13:15)-Xtru(13:15))]; % position
DXv = [norm(Xest(4:6)-Xtru(4:6)), norm(Xest(10:12)-Xtru(10:12)),
norm(Xest(16:18)-Xtru(16:18))]; % velocity

ABS = [norm(DXr) norm(DXv)];

%***RELATIVE OD
ACCURACY*****
DL13=norm(Xest(1:3)-Xest(13:15))-norm(Xtru(1:3)-Xtru(13:15));
DL21=norm(Xest(1:3)-Xest(7:9))-norm(Xtru(1:3)-Xtru(7:9));

```

```

DL32=norm(Xest(7:9)-Xest(13:15))-norm(Xtru(7:9)-Xtru(13:15));

DL13v=norm(Xest(4:6)-Xest(16:18))-norm(Xtru(4:6)-Xtru(16:18));
DL21v=norm(Xest(10:12)-Xest(4:6))-norm(Xtru(10:12)-Xtru(4:6));
DL32v=norm(Xest(16:18)-Xest(10:12))-norm(Xtru(16:18)-Xtru(10:12));

DL = norm([DL13 DL21 DL32]);
DLv = norm([DL13v DL21v DL32v]);
REL = [DL DLv];

```

***MCRUNbp.m*** (from Section 5.4 Orbit Accuracy)

```

function [Mniter, MXest, Merr, Stderr, Merv, Stderiv, MABSR, StdABSR,
MABSV, StdABSV, MRELR, StdRELR, MRELV, StdRELV, MRMS, Mvar,
MPU_L1, MPU_L2, MPU_L3, M_Pcov] = MCRUNbp()
format long e

x = 1;
days = [1 3 5 7 10 15 20 25 30 35 37 40 45 50 55 60 65 70 75 80 85 90];
for d = [1 3 5 7 10 15 20 25 30 35 37 40 45 50 55 60 65 70 75 80 85 90];
    d;
    x;
    %***SET THE APPROPRIATE START TIME TO THE
%CASE*****
    to_0 = 0; % EPOCH, X_truth(1,1); % start of timespan
    %***SET THE APPROPRIATE END TIME TO THE
%CASE*****
    tf = 60*60*24*d; % days
    %***SET THE APPROPRIATE INTERVAL TIME TO THE
%CASE*****
    I = 60*60*24; % sec, skips every 24 hours for 1 day intervals 1 meas/day

%***TIMESPAN*****
    ts = to_0:I:tf; % timespan for INTERVALS

```

```

% *****
% ***INITIAL CONDITION VECTOR FROM NomOrb_half.mat
% *****
% Xo = [X_traj(1,2:7) X_traj(1, 8:13) X_traj(1, 14:19) X_traj(1,20:25)];
% 1x24
Xo_0 = [9.071855731392364e-006, 1.481339600426210e+011,
2.475763859214967e+009,...
-3.007344842423256e+004, 1.841210487873900e-012,
3.077216312703865e-014,...
2.487592427287221e+009, 1.503039437121128e+011, -
1.292020540364669e+009,...
-2.963517041188058e+004, 2.476377264069181e+002,
4.268667617608362e+002,...
-2.487592427287196e+009, 1.503039437121128e+011, -
1.292020540364669e+009,...
-2.963517041188058e+004, -2.476377264069132e+002, -
4.268667617608363e+002,...
-5.116553726769669e+010, 1.405761582972606e+011, 0,...
-2.798843961236834e+004, -1.018695892245916e+004, 0];

% *****
% ***REFERENCE START VECTOR WITH ERROR aka "FUZZ"
% ADDED ON
% *****
errorstart = 0*[-1 1 1 1e-6 1e-6 1e-6 -1 1 1 1e-6 1e-6 1e-6 -1 1 1 1e-6 1e-
6 1e-6].*randn(1,18); %NO Guess Error
Xrefstart_0 = Xo_0 + [errorstart zeros(1,6)];

% *****
% READ IN SIGMA ERROR, BIASES & WEIGHTS FOR EACH CASE
% *****
c = 299792458; %m/s speed of light
tau = 1000; %sec, Doppler count time
Astd = 1e-15; % Allan Standard deviation for a Doppler Count time
=1000s

```

```

pi1 = norm([Xrefstart_0(1:3) - Xrefstart_0(19:21)]); %m (1x1) range
scalar for LISA 1
pi2 = norm([Xrefstart_0(7:9) - Xrefstart_0(19:21)]);
pi3 = norm([Xrefstart_0(13:15) - Xrefstart_0(19:21)]);
pi13 = norm([Xrefstart_0(1:3) - Xrefstart_0(13:15)]); %m (1x1) range
scalar for LISA Leg 13
pi21 = norm([Xrefstart_0(7:9) - Xrefstart_0(1:3)]);
pi32 = norm([Xrefstart_0(13:15) - Xrefstart_0(7:9)]);

RTL1 = (2*pi1)/c; %round-trip light time for LISA 1
RTL2 = (2*pi2)/c;
RTL3 = (2*pi3)/c;
RTL13 = (2*pi13)/c; %round-trip light time for LISA Leg 13
RTL21 = (2*pi21)/c;
RTL32 = (2*pi32)/c;
%range error due to clock instability
ci1 = ((sqrt(2))*c*RTL1*Astd)*sqrt(60/(1000));
ci2 = ((sqrt(2))*c*RTL2*Astd)*sqrt(60/(1000));
ci3 = ((sqrt(2))*c*RTL3*Astd)*sqrt(60/(1000));
ci = norm([ci1 ci2 ci3]);
ci13 = ((sqrt(2))*c*RTL13*Astd)*sqrt(60/(1000));
ci21 = ((sqrt(2))*c*RTL21*Astd)*sqrt(60/(1000));
ci32 = ((sqrt(2))*c*RTL32*Astd)*sqrt(60/(1000));
cileg = norm([ci13 ci21 ci32]);

M1 = RTL1/tau; %constant M
M2 = RTL2/tau;
M3 = RTL3/tau;

M13 = RTL13/tau; %constant M
M21 = RTL21/tau;
M32 = RTL32/tau;
%range rate frequency instability to range rate error
f1 = ((sqrt(2+ log2(M1)))*c*Astd)*sqrt(60/(1000));
f2 = ((sqrt(2+ log2(M2)))*c*Astd)*sqrt(60/(1000));
f3 = ((sqrt(2+ log2(M3)))*c*Astd)*sqrt(60/(1000));

```



```

f = norm([f1 f2 f3]);

%***SIGMA ERRORS, S
sr = .60*sqrt(60/(30*60)); %m, for 60-sec average
srr = 3e-5*sqrt(60/(30*60)); %m/s, range rate error for 60-sec average
sa = 5e-8*sqrt((30*60)/(30*60)); %rad, right ascension error for 30-min
average
sd = 5e-8*sqrt((30*60)/(30*60)); %rad, declination error for 30-min
average
sir = .60*sqrt(60/(30*60));%3*sqrt(1/60); %m, interspacecraft range
error for 1-sec average
sirr = 3e-5*sqrt(60/(30*60));%1.5e-4*sqrt(1/60); %m/s, interspacecraft
range rate error for 1-sec average

%*****
%Observation Noise Covariances, Ri's
%*****
% With Noise*****

R0a = [(sr + ci1) (srr + f1) sd sa].^2;
R0b = [(sr + ci2) (srr + f2) sd sa].^2;
R0c = [(sr + ci3) (srr + f3) sd sa].^2;
R1 = [(sr+ ci1) (srr + f1) sd sa (sir + ci13) (sirr) (sr+ ci2) (srr+ f2)...
sd sa (sir+ci21) (sirr) (sr+ ci3) (srr+ f3) sd sa (sir+ci32) (sirr)].^2;
R2a = [(sr+ ci1) (srr+ f1)].^2;
R2b = [(sr+ ci2) (srr+ f2)].^2;
R2c = [(sr+ ci3) (srr+ f3)].^2;
R3 = [(sr+ ci1) (srr+ f1) (sir + ci13) (sirr) (sr+ ci2) (srr+ f2)...
(sir + ci21) (sirr) (sr+ ci3) (srr+ f3) (sir + ci32) (sirr)].^2;

%*****
%BIASES
%*****

%RANGE RATE

```

```

pdot1i = (((Xrefstart_0(1) - Xrefstart_0(19))*(Xrefstart_0(4) -
Xrefstart_0(22))) + ((Xrefstart_0(2) - Xrefstart_0(20))*...
(Xrefstart_0(5) - Xrefstart_0(23))) + ((Xrefstart_0(3) -
Xrefstart_0(21))*(Xrefstart_0(6) - Xrefstart_0(24))))/pi1); %m/s

pdot2i = (((Xrefstart_0(7) - Xrefstart_0(19))*(Xrefstart_0(10) -
Xrefstart_0(22))) + ((Xrefstart_0(8) - Xrefstart_0(20))*...
(Xrefstart_0(11) - Xrefstart_0(23))) + ((Xrefstart_0(9) -
Xrefstart_0(21))*(Xrefstart_0(12) - Xrefstart_0(24))))/pi2);

pdot3i = (((Xrefstart_0(13) - Xrefstart_0(19))*(Xrefstart_0(16) -
Xrefstart_0(22))) + ((Xrefstart_0(14) - Xrefstart_0(20))*...
(Xrefstart_0(17) - Xrefstart_0(23))) + ((Xrefstart_0(15) -
Xrefstart_0(21))*(Xrefstart_0(18) - Xrefstart_0(24))))/pi3);

pdot13i = (((Xrefstart_0(1) - Xrefstart_0(13))*(Xrefstart_0(4) -
Xrefstart_0(16))) + ((Xrefstart_0(2) - Xrefstart_0(14))*...
(Xrefstart_0(5) - Xrefstart_0(17))) + ((Xrefstart_0(3) -
Xrefstart_0(15))*(Xrefstart_0(6) - Xrefstart_0(18))))/pi13); %m/s

pdot21i = (((Xrefstart_0(7) - Xrefstart_0(1))*(Xrefstart_0(10) -
Xrefstart_0(4))) + ((Xrefstart_0(8) - Xrefstart_0(2))*...
(Xrefstart_0(11) - Xrefstart_0(5))) + ((Xrefstart_0(9) -
Xrefstart_0(3))*(Xrefstart_0(12) - Xrefstart_0(6))))/pi21);

pdot32i = (((Xrefstart_0(13) - Xrefstart_0(7))*(Xrefstart_0(16) -
Xrefstart_0(10))) + ((Xrefstart_0(14) - Xrefstart_0(8))*...
(Xrefstart_0(17) - Xrefstart_0(11))) + ((Xrefstart_0(15) -
Xrefstart_0(9))*(Xrefstart_0(18) - Xrefstart_0(12))))/pi32);
%range error due to unknown time-tag error
dT = 1e-6; %Station clock epoch error
g1 = dT*pdot1i;
g2 = dT*pdot2i;
g3 = dT*pdot3i;
g13 = dT*pdot13i;
g21 = dT*pdot21i;
g32 = dT*pdot32i;

```

```

%clock rate offset
rT = 5e-14; %Station clock rate offset
h1 = c*RTL1*rT;
h2 = c*RTL2*rT;
h3 = c*RTL3*rT;
    h13 = c*RTL13*rT;
    h21 = c*RTL21*rT;
    h32 = c*RTL32*rT;

br = 2.05; %2 m instrument bias + 5 cm total bias
bir = 2; % 10; % 10 m bias for each LISA

B = [(br+g1+h1) (br+g2+h2) (br+g3+h3) (bir + g13 + h13) (bir + g21 +
h21) (bir + g32 + h32)]; %biases 1x6
%B = [0 0 0 0 0 0]; %No biases 1x6

%*****
%***READ IN OTHER INPUTS
%*****

plt = 0;

%*****
%***CALL LISA_BPtot4.m to RUN DIFFERENT CASES
%*****

%for i = 1:35, % 100 random runs of each case
for i = 1, % 1 run of each for full noise, 100 random runs of each case

    for k = 1:length(ts)
        %S = ([sr sr sr srr srr srr sd sd sd 0 0 0 sir sir sir srr srr srr ci1 ci2
ci3 f1 f2 f3 ci13 ci21 ci32]*zeros(1, k+1)); %No Noise, nx27
        S = ([sr sr sr srr srr srr sd sd sd sa sa sa sir sir sir srr srr srr ci1 ci2
ci3 f1 f2 f3 ci13 ci21 ci32]*ones(1, k+1)); %Full Noise, nx27
        %S = randn(k+1, 27).*([sr sr sr srr srr srr sd sd sd sa sa sa sir sir sir
srr srr srr ci1 ci2 ci3 f1 f2 f3 ci13 ci21 ci32]*ones(1, k+1)); %Random
Noise, nx27

```

```

        %errorstart = randn(k+1, 18).*([-1 1 1 1e-6 1e-6 1e-6 -1 1 1 1e-6 1e-
6 1e-6 -1 1 1 1e-6 1e-6 1e-6]*1000*ones(1,k+1)); %NO Guess Error
        %errorstart = randn(1, 18).*([-1 1 1 1e-6 1e-6 1e-6 -1 1 1 1e-6 1e-6
1e-6 -1 1 1 1e-6 1e-6 1e-6]*1000.*ones(1,1))
        Xrefstart_0 = Xo_0 + [errorstart zeros(1,6)];
    end

    %CASE 0: DSN & VLBI (All absolute data)
    % [Xest, ABS, REL, RMS, var, PU_L1, PU_L2, PU_L3, err, erv, niter,
Pcov] = MCLISA_bp(to_0, tf, I, Xrefstart_0, Xo_0, R1, R2, R3, S, B, plt,
obs)
    [Xest0(i,:), ABS0(i,:), REL0(i,:), RMS0(i,:), var0(i,:),PU0_L1(:,i),
PU0_L2(:,i), PU0_L3(:,i), err0(i,:), erv0(i,:), niter0(i,:), Pcov0(i,:)] =
MCLISA_bp(to_0, tf, I, Xrefstart_0, Xo_0, R0a, R0b, R0c, S, B, plt, 0);

    %CASE 1: DSN, VLBI & RELATIVE
    [Xest1(i,:), ABS1(i,:), REL1(i,:), RMS1(i,:),var1(i,:),PU1_L1(:,i),
PU1_L2(:,i), PU1_L3(:,i), err1(i,:), erv1(i,:), niter1(i,:), Pcov1(i,:)] =
MCLISA_bp(to_0, tf, I, Xrefstart_0, Xo_0, R1, R1, R1, S, B, plt, 1);

    %CASE 2: DSN
    [Xest2(i,:), ABS2(i,:), REL2(i,:), RMS2(i,:), var2(i,:), PU2_L1(:,i),
PU2_L2(:,i), PU2_L3(:,i), err2(i,:), erv2(i,:), niter2(i,:), Pcov2(i,:)] =
MCLISA_bp(to_0, tf, I, Xrefstart_0, Xo_0, R2a, R2b, R2c, S, B, plt, 2);

    %CASE 3: DSN & RELATIVE
    [Xest3(i,:), ABS3(i,:), REL3(i,:), RMS3(i,:), var3(i,:),
PU3_L1(:,i),PU3_L2(:,i), PU3_L3(:,i), err3(i,:), erv3(i,:), niter3(i,:),
Pcov3(i,:)] = MCLISA_bp(to_0, tf, I, Xrefstart_0, Xo_0, R3, R3, R3, S, B,
plt, 3);
    end

%*****
%***ERROR ACCUMULATION & ANALYSIS
%*****
    for j = x,

```

```

%4 Case [case 0(DSN/VLBI),case 1(DSN/VLBI/REL),case 2(DSN),case
3(DSN/REL)]
niter_tot = [niter0(:,), niter1(:,), niter2(:,), niter3(:,)]; %nx4
M_niter(j,:) = mean(niter_tot, 1); % 1x6

Xest = [Xest0(:,), Xest1(:,), Xest2(:,), Xest3(:,)]; %n x 144 (1x24
each)
M_Xest(j,:) = mean(Xest,1); % 1x144

ABSR = [ABS0(:,1), ABS1(:,1), ABS2(:,1), ABS3(:,1)]; %ix6
M_ABSr(j,:) = mean(ABSR,1); % 1x6
std_ABSr(j,:) = std(ABSR, 0, 1); % 1x6

ABSV = [ABS0(:,2), ABS1(:,2), ABS2(:,2), ABS3(:,2)]; %ix6
M_ABSv(j,:) = mean(ABSV,1); % 1x6
std_ABSv(j,:) = std(ABSV, 0, 1); % 1x6

RELR = [REL0(:,1), REL1(:,1), REL2(:,1), REL3(:,1)]; %ix6
M_RELr(j,:) = mean(RELR, 1); % 1x6
std_RELr(j,:) = std(RELR, 0, 1); % 1x6

RELv = [REL0(:,2), REL1(:,2), REL2(:,2), REL3(:,2)]; %ix6
M_RELv(j,:) = mean(RELv,1); % 1x6
std_RELv(j,:)= std(RELv, 0, 1); % 1x6

RMS = [RMS0(:,), RMS1(:,), RMS2(:,), RMS3(:,)]; %ix18 (1x3 for
each case)
M_RMS(j,:) = mean(RMS,1); % 1x18

var = [var0(:,), var1(:,), var2(:,), var3(:,)]; %ix18 (1x3 for each case)
M_var(j,:) = mean(var,1); % 1x18

PU_L1 = [PU0_L1(:,1);...
PU1_L1(:,1);...
PU2_L1(:,1);...
PU3_L1(:,1)]; %ix144(6x1 each)

```

```

M_PU_L1(:,j) = mean(PU_L1,2);

PU_L2 = [PU0_L2(:,1);...
PU1_L2(:,1);...
PU2_L2(:,1);...
PU3_L2(:,1)];

M_PU_L2(:,j) = mean(PU_L2,2);

PU_L3 = [PU0_L3(:,1);...
PU1_L3(:,1);...
PU2_L3(:,1);...
PU3_L3(:,1)];

M_PU_L3(:,j) = mean(PU_L3,2);

err = [err0(:,), err1(:,), err2(:,), err3(:,)]; %ix18 (1x3 each)
M_err(j,:) = mean(err,1); % 1x18
std_err(j,:) = std(err, 0, 1);

erv = [erv0(:,), erv1(:,), erv2(:,), erv3(:,)]; %ix18
M_erv(j,:) = mean(erv,1); % 1x18
std_erv(j,:) = std(erv, 0, 1);

Pcov = [Pcov0(1,:), Pcov1(1,:), Pcov2(1,:), Pcov3(1,)]; %ix(18x3)=
ix54
M_Pcov(j,:) = mean(Pcov,1); % 1x(18x4)

if x < length(days)
x = x + 1
else x = length(days)
end
end
Mniter = M_niter(:,);
MXest = M_Xest(:,);
Merr = M_err(:,);
Stderr = std_err(:,);

```

```
Merv = M_erv(,:);  
Stderv = std_erv(,:);  
MABSR = M_ABSr(,:);  
StdABSR = std_ABSr(,:);  
MABSV = M_ABSv(,:);  
StdABSV = std_ABSv(,:);  
MRELR = M_RELR(,:);  
StdRELR = std_RELR(,:);  
MRELV = M_RELV(,:);  
StdRELV = std_RELV(,:);  
MRMS = M_RMS(,:);  
Mvar = M_var(,:);  
MPU_L1 = M_PU_L1(,:);  
MPU_L2 = M_PU_L2(,:);  
MPU_L3 = M_PU_L3(,:);  
M_Pcov = M_Pcov(,:);  
end
```

## Bibliography

Listed in order of appearance in text

- [1] LISA NASA/JPL Homepage. URL: <http://lisa.jpl.nasa.gov>
  
- [2] Sorenson, H. W., "Least-squares estimation: from Gauss to Kalman," *IEEE Spectrum*, Vol. 7, July 1970, pp. 63-68.
  
- [3] Folkner, W.M., "The Laser Interferometer Space Antenna Mission," AIAA Space 2000 Conference and Exposition, Long Beach, CA, 19-21 September 2000 (AIAA 2000-5129).
  
- [4] Hechler, F. and Folkner, W. M., "Mission Analysis for the Laser Interferometer Space Antenna (LISA) Mission," *Advanced Space Research*, Vol. 32, No.7, Elsevier Ltd., Great Britain, 2003, pp. 1277-1282.
  
- [5] Vallado, David A., and Alfano, Salvatore, "A Future Look at Space Surveillance and Operations," Space Surveillance Workshop, Washington, DC, October 20-23, 1998.
  
- [6] Ziemer, John K., Merkowitz, Stephen M., "Microthrust Propulsion for the LISA Mission," The 40<sup>th</sup> AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Fort Lauderdale, FL., 11-14 July 2004 (AIAA 2004-3439).
  
- [7] Sweetser, Theodore, H., "An End-to-end Trajectory Description of the LISA Mission," *Classical and Quantum Gravity* 22, Institute of Physics Publishing Ltd., United Kingdom, April 2005, pp. 429-435.
  
- [8] Sweetser, Theodore, H., "LISA Mission Description," Internal Memo NASA/JPL, Version 2.0., Oct. 28, 2005.
  
- [9] LISA newsletter Vol. 1, No. 1, LISA NASA/JPL Homepage, URL: [http://www.lisa.nasa.gov/newsletter/newsletter\\_200404.pdf](http://www.lisa.nasa.gov/newsletter/newsletter_200404.pdf), April 2004
  
- [10] Irion, R., "Gravitational Wave Hunters Take Aim at the Sky." *Science*, Vol. 297, August 16, 2002, pp. 1113-1115.

- [11] *LISA Science Brochure*, LISA NASA/JPL Homepage, URL: <http://www.lisa.nasa.gov/LISAbrochure.pdf>
- [12] Webster, Guy, (11/20/2001). *NASA Spacecraft to Hunt for Elusive Gravity Ripples*, Cassini-Huygens News Releases, URL: <http://saturn.jpl.nasa.gov/news/press-releases-01/20011120-pr-a.cfm>
- [13] *GRACE: Gravity Recovery and Climate Experiment* (2/11/2004) GRACE Gravity Measurement, URL: [http://www.csr.utexas.edu/grace/science/gravity\\_measurement.html](http://www.csr.utexas.edu/grace/science/gravity_measurement.html)
- [14] Hughes, Steven P., "Preliminary Optimal Orbit Design for the Laser Interferometer Space Antenna (LISA)," *25<sup>th</sup> Annual AAS Guidance and Control Conference*, Breckenridge CO, Feb. 2002.
- [15] Bender, P.L., "LISA: A proposed joint ESA-NASA gravitational-wave mission": *Gravitational Waves*, Institute of Physics Publishing Ltd., Philadelphia, 2001, Chapter 10, pp. 115-151.
- [16] Laser Interferometer Space Antenna: A cornerstone Mission for the Observation of Gravitational Waves, *Systems and Technology Study (STS) Report*, 11 July, 2000, URL: [ftp://ftp.rzg.mpg.de/pub/grav/lisa/sts/sts\\_1.05.pdf](ftp://ftp.rzg.mpg.de/pub/grav/lisa/sts/sts_1.05.pdf)
- [17] Fleck, Melissa E. and Starin, Scott R., "Evaluation of a Drag-Free Control Concept for Missions in Low Earth Orbit," *AIAA Guidance, Navigation and Control Conference and Exhibit*, Austin, Texas, 11-14 August 2003, (AIAA 2003-5747).
- [18] Dhurandhar, S. V., Rajesh, N.K., Kosti, S, and Vinet, J-Y, "Fundamentals of the LISA Stable Flight Formation," *Classical and Quantum Gravity*, Vol. 22, Institute of Physics Publishing Ltd., United Kingdom, Jan. 2005, pp. 481- 487.
- [19] Folkner, W. M., et al., "LISA orbit selection and stability." *Classical and Quantum Gravity*, Vol. 14, Institute of Physics Publishing Ltd., United Kingdom, 1997, pp.1405-1410.
- [20] *LISA Mission Report from 35<sup>th</sup> COSPAR meeting*, found at LISA ESA Homepage, URL: <http://lisa.esa.int/science-e/www/area/index.cfm?fareaid=27>, pp.113-116.

- [21] NASA/JPL Space Technology 7 (ST7) About ST7 Overview 5/2004 URL:  
[http://nmp.jpl.nasa.gov/st7/ABOUT/About\\_index.html](http://nmp.jpl.nasa.gov/st7/ABOUT/About_index.html).
- [22] ESA: Science and Technology Webpage LISA-Pathfinder Homepage Factsheet:  
5/31/2005, URL: <http://sci.esa.int/science-e/www/object/index.cfm?fobjectid=31714>  
-Instruments: last update 10/21/2004, URL:  
<http://sci.esa.int/science-e/www/object/index.cfm?fobjectid=35589>  
-Summary: 5/31/2005, URL:  
<http://sci.esa.int/sciencee/www/object/index.cfm?fobjectid=31431>
- [23] NASA Beyond Einstein: Great Observatories Homepage URL:  
<http://universe.nasa.gov/program/observatories.html>
- [24] NASA: GSFC Constellation X Homepage last update 1/6/2006 URL:  
<http://constellation.gsfc.nasa.gov/> (checked 3/25/2005)
- [25] The TRIAD Spacecraft last updated 11/25/2001 URL: <http://www-spof.gsfc.nasa.gov/Education/wtriad.html>
- [26] NASA/JPL SIM Planetquest Mission Homepage URL:  
[http://planetquest.jpl.nasa.gov/SIM/sim\\_index.html](http://planetquest.jpl.nasa.gov/SIM/sim_index.html) (checked 3/25/2005)
- [27] Savage, Don, et. al. *NASA Gravity Probe B mission enters science phase, ready to test Einstein's theory* NASA News Release 9/7/2004 URL:  
<http://www.nasa.gov/centers/marshall/news/news/releases/2004/04-228.html>
- [28] NASA Gravity Probe B Homepage URL: <http://www.gravityprobeb.com/>
- [29] GFZ Potsdam Department 1, CHAMP Mission Homepage 11/29/2005 URL:  
[http://www.gfz-potsdam.de/pb1/op/champ/main\\_CHAMP.html](http://www.gfz-potsdam.de/pb1/op/champ/main_CHAMP.html) (checked 3/25/2005)
- [30] NASA/JPL Cassini-Huygens Homepage Overview: URL:  
<http://saturn.jpl.nasa.gov/overview/index.cfm>
- [31] Danzmann, K., & Rudiger, A., "LISA technology- concept, status, prospects,"

*Classical and Quantum Gravity*, Vol. 14, Institute of Physics Publishing Ltd., United Kingdom, 2003, pp. S1-S9.

[32] Crowder, J. & Corning, N. J., "Beyond LISA: Exploring Future Gravitational Wave Missions," 17 October 2005, URL: [http://www.arxiv.org/PS\\_cache/gr-qc/pdf/0506/0506015.pdf](http://www.arxiv.org/PS_cache/gr-qc/pdf/0506/0506015.pdf).

[33] Wiesel, William, E., *Spaceflight Dynamics 2<sup>nd</sup> ed.*, Irwin/McGraw-Hill Companies Inc., New York, NY, 1997.

[34] Tapley, Byron D. "Statistical Orbit Determination Theory," *Recent Advances in Dynamical Astronomy*, D. Reidel Publishing Company, Boston, USA, 1973, pp. 396-425.

[35] Vallado, David A., *Fundamentals of Astrodynamics and Applications*, Space Technology Series ed. Wiley J. Larson, The McGraw-Hill Companies, Inc., New York, NY, 1997.

[36] Battin, Richard, H., *An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition*, American Institute of Aeronautics and Astronautics, Inc., Reston, VA, 1999.

[37] Chobotov, Vladimir, A. editor, *Orbital Mechanics: AIAA Education Series*, American Institute of Aeronautics and Astronautics, Inc., Washington, DC, 1991.

[38] Boulet, Dan L. Jr., *Methods of Orbit Determination for the Micro Computer*, Willmann-Bell, Inc., Richmond, VA., 1991.

[39] Vallado, David A., and Alfano, Salvatore, "A Future Look at Space Surveillance and Operations," *Space Surveillance Workshop*, Washington, DC, October 20-23, 1998.

[40] Montenbruck, O., and Gill, E., *Satellite Orbits- Models, Methods, and Applications*, Springer-Verlag, Berlin, Germany, 2000.

[41] Thornton, Catherine L., Border, James S., "Radiometric Tracking Techniques for Deep Space Navigation", Jet Propulsion Laboratory, California Institute of Technology, Oct. 2000.



- [42] Haerberle, Dennis W., Spencer, David B., and Ely, Todd A., "Preliminary Investigation of Interplanetary Navigation Using Large Antenna Arrays with Varied Baselines." *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*: Providence, RI, 16-19 August 2004, (AIAA 2004-4744) .
- [43] JPL:Radio Science Systems Group Website: "Basics of Spaceflight," <http://www2.jpl.nasa.gov/basics/>.  
-JPL Radio Astronomy DSN [JPL: Basics of Spaceflight "Telecommunications"]  
<http://www2.jpl.nasa.gov/basics/bsf10-1.html>
- [44] Portock, Brian, M., Baird, Darren, T., et. al., "Mars Exploration Rovers Cruise Orbit Determination," (AIAA 2004-4981), *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Providence, RI, 16-19 August 2004 (AIAA 2004-4981).
- [45] Tapley, Byron D., "Fundamentals of Orbit Determination," *Lecture Notes in Earth Sciences, Vol. 25: Theory of Satellite Geodesy and Gravity Field Determination* ed. F. Sanso and R. Rummel, Springer-Verlag, New York, NY, 1989, pp. 235-260.
- [46] Noton, Maxwell, *Spacecraft Navigation and Guidance, Advances in Industrial Control Series*, Springer-Verlag London Limited, Great Britain, 1998.
- [47] Tapley, Byron D., Schutz, Bob E., Born, George H., *Statistical Orbit Determination*, Elsevier Academic Press, New York, NY, 2004.
- [48] Bate, Roger, R., Mueller, Donald, D., and White, Jerry, E., *Fundamentals of Astrodynamics*, Dover Publications, Inc. New York, NY, 1971.
- [49] Gelb, A. ed. and Technical Staff, The Analytic Sciences Corporation, *Applied Optimal Estimation*, The M.I.T. Press, MIT Cambridge, MA, 1974.
- [50] Devore, Jay, L., *Probability and Statistics: For Engineering and the Sciences*, 6<sup>th</sup> ed., Brooks/Cole Thomson Learning, Inc., Belmont, CA, 2004.

