

Levenberg-Marquardt Filter for Orbit Estimation

a project presented to
The Faculty of the Department of Aerospace Engineering
San José State University

in partial fulfillment of the requirements for the degree
Master of Science in Aerospace Engineering

By

Robert Ziegler

May 2019

approved by

Prof. Jeanine Hunter
Faculty Advisor



ABSTRACT

LEVENBERG-MARQUARDT FILTER FOR ORBIT ESTIMATION

By Robert Ziegler

This paper tests the Levenberg-Marquardt method of least-squares as it is applied to orbit estimation using noisy Doppler data. Doppler data used in the analysis is simulated by calculating range rate at multiple points along the path of a satellite at times the satellite would pass over a real ground station. The paper begins by discussing how real Doppler data would be used for orbit estimation.

Next, a reference frame used for the analysis is defined. Then, the methods used to acquire

simulated data are outlined. Finally, $G > IJETbET(siv)-9(e)-5(nbe)4(r)-6(g)] TJETQq0.00000912 0 612 792 reW*nl$

Acknowledgments

When it came time to begin a project to fulfill the Master of Science in Aerospace Engineering requirements, my desire was to work on a challenging problem that would push my limits as an engineer. This desire was fulfilled when Professor Jeanine Hunter became my project advisor. As I have progressed through this project (hitting every wall that could possibly be found), Professor Hunter has given me guidance and has shown me an extreme amount of patience, and for this she has my sincerest gratitude.

TEST PROGRAM	113
ORBIT AND DATA GENERATION PROGRAM	122
LEVENBERG-MARQUARDT FILTER	125

List of Symbols

Symbol	Definition	Units (SI)
a	semimajor axis length	ft (m)

f	Wavelength	ft (m)
	Topocentric range-rate	ft/s (m/s)
	Sidereal time	deg
	Orbital period	s
	Longitude	deg or rad
	Argument of perigee	deg or rad
	Right ascension of the ascending node	deg or rad
Subscripts		
()	Geocentric	
()	Geodetic	
()	Earth	
()	Equatorial	
()		
()	Polar	
() _r	Receiver	
()	Satellite	
()	Transmitter	
Acronyms		

1. INTRODUCTION

Orbit determination is a tool used by engineers, scientists, and hobbyists to understand the trajectory of objects travelling through space. Using orbital mechanics and observations of the position and velocity of the satellite, observers can predict the position of the object over time. While this prediction was problematic for the first satellite in space, technology now allows precise tracking of satellites using enhanced physical models and computing power. Presently, anyone with access to the internet can either look up the ephemeris of their desired satellite or download sophisticated software that can approximate and display the trajectory of many orbiting satellites; however, a different approach is proposed in this report.

As more satellites are introduced into earth orbit, novel approaches to orbit determination are required. While satellites can send ephemeris data and computer software can approximate the position of the satellite over time, there must be alternative methods which do not rely on such

Figure 1.1 Damped Least Squares Estimation [1].

1.1 DOPPLER ESTIMATION

Many approaches have been taken to use Doppler data to predict the orbit of a satellite. Some of the simpler methods assume a circular orbit, while the more complex variations are flexible with orbital parameters but require more information on the orbital history of the satellite. In this section, texts and articles which discuss the Doppler effect DE, and how DE of a received signal pertains to orbit determination.

As new equipment and software are introduced into space systems, testing these novelties in the field is critical. The analysis of the Precise Range and Range-rate Equipment (PRARE), a satellite track

used to acquire ephemeris data and allow tracking of future passes of the satellite. After processing, the X-band signal is modulated and sent back to the PRARE space segment where range measurements are calculated using the two-way signal time [2]:

$$= \frac{1}{c} (\dots + \dots)$$

— — —

— — — — —

between relative velocity and Doppler shift can be calculated (annotated from [3]):

$$= \frac{0}{\dots} \tag{1.6}$$

where

$$= \frac{(\dots)}{\dots} \tag{1.7}$$

in spherical coordinates, and ω_0 is the carrier frequency. To find \dots , velocities of the satellite and ground station are studied in ECEF coordinates, and perturbing forces are analyzed.

Finding a value for \dots requires an orbit generator with corrections for the following perturbing forces R, S, and W (annotated from [3]):

$$= \dots + \dots + \dots \tag{1.8}$$

$$\dots : \dots \cdot \frac{1}{2} \cdot \frac{1}{4} \tag{1.9}$$

$$= \dots \cdot \sin^2(\dots + \dots) \cdot \sin^2(\dots) \tag{1.10}$$

$$= \dots \cdot \sin(2(\dots + \dots)) \cdot \sin^2(\dots) \tag{1.11}$$

$$= \dots \cdot \sin(\dots + \dots) \cdot \sin(2 \dots) \tag{1.12}$$

student science endeavor. In orbit, the proximity of the satellites made individual identification

rotating coordinate system [5]:

$$= \quad (1.20a)$$

$$= \sin^{-1}(\sin \cdot \sin(\cdot + \theta)) \quad (1.19b)$$

-

$$= \frac{\sin[\theta + \sin^{-1}(\frac{v}{c} \cos \alpha)] \sin[(\omega - \frac{v}{c} \cos \alpha) t]}{\sqrt{2 + 2 \cos[\theta + \sin^{-1}(\frac{v}{c} \cos \alpha)] \cos[(\omega - \frac{v}{c} \cos \alpha) t]}} \quad (1.20)$$

Doppler shift can now be found using [5]:

$$= \frac{f_r}{f_t} = \frac{c}{c - v \cos \alpha} \quad (1.21)$$

where f_t is the transmitted frequency of the downlink signal of the satellite, and c is the speed of light. Using Doppler measurements to improve global positioning system (GPS) performance is the topic of the next article.

As aircraft maneuver through the air, the Doppler shift induced on received signals can be much greater than observed values at a ground station. In their investigation, Agostino, Manzano, and Marucco [6] use a Kalman filter estimator to improve GPS tracking of aircraft using Doppler measurements. In this process, a precise ephemeris of a satellite is used, along with the inherent Doppler shift to calculate the velocity of the aircraft. It is determined that using this Kalman filter estimation reduces errors caused by noisy measurements.

In another experiment, Ialongo [7] uses a cycle counter to read two-way Doppler measurements and produce range rate of a satellite. This method feeds an input frequency f into a counter, where

$$= \frac{2048 \cdot (r_1)^{\frac{1}{2}}}{2048 \cdot (r_2)^{\frac{1}{2}} + 26240 \cdot r_1} \quad (1.23)$$

where

$$= \frac{2}{2048 \cdot r_2 + 26240 \cdot r_1} \quad (1.24)$$

r_1 and r_2 are cycle counts, and

$$(r_1, r_2) = \frac{(r_1 + r_2 \sin \theta + r_2 \cos \theta)}{2} \quad (1.25)$$

Here, r is the distance travelled by the signal.

In this section, methods of measuring the Doppler effect and range rate of a satellite were discussed.

Now, applications to orbit determination will be examined.

1.2 ORBIT DETERMINATION FROM DOPPLER DATA

Having covered a variety of methods used to measure the Doppler shift inherent in satellite communication, orbit determination schemes which use this data are presented in this section, starting with a simple circular orbit-based algorithm.

The experiment presented by Schuch [8] uses observations of the orbital period of a satellite to estimate a circular orbit. Using Doppler measurements, a Time of Closest Approach (TCA) is determined by finding when the received frequency from a satellite is equal to the transmitted

frequency, that is, when there is no Doppler shift present. Noting the TCA, a second pass is evaluated, and a first estimation of Δf can be made. The error incurred from the rotation of the earth is corrected by repeating this process for two successive descending passes. The time elapsed between these passes is an integer multiple of T . This integer can be calculated with the equation (annotated from [8]):

$$= \left(\frac{\Delta f}{f} \right) T \quad (1.26)$$

where

satellite with

$$= (1 \quad -) \tag{1.28}$$

If the distance between the satellite and each of the ground stations is known at two different times, the Keplerian orbit can be formed through methods which will be discussed in the next chapter.

Moving to a higher earth orbit, an experiment performed by Estefan [11] uses differenced Doppler for elliptical orbiters. A method of orbit determination for high-orbit elliptical satellites, Very Long Baseline Interferometry (VLBI), is under investigation [11] for its ability to improve orbit accuracy. The problem with this process is its high cost. Termed “quasi-VLBI,” an alternative differenced (two-way minus three-way) Doppler is proposed. While data measured with differenced Doppler is not as accurate as seen with VLBI, Doppler and range data can be supplied much faster for navigation purposes [11].

Differenced Doppler first relies on extracting range measurements from Orbit Analysis and

subscripts represent different ground stations.

Guier and Weiffenbach [12] use the entirety of a Doppler curve to obtain orbital elements in their article. While many Doppler-based orbit determination schemes include an intermediary process, steps can be taken to maximize Doppler data by directly calculating the six orbital elements from the frequency shift curve. Additional elements, totaling eight, are extracted to account for errors, such as refraction from the ionosphere. Although computational cost is higher using this single-pass method, its results have shown that such calculations are possible. The final article reviewed in the present chapter concerns the use of the Doppler effect in GPS measurements.

The derivation of the GPS relativistic Doppler effects is given by Zhang, et. al [13]. In the GPS observation system, additional changes in frequency are present. These shifts are caused by gravity potential from the geoid shape of the earth, the gravity field of the earth, and the orbital eccentricity of the satellite. To correct the relativistic effects, a special relativity term is added to the equation for received frequency, which will not be included in this paper as these corrections are not desired for the present analysis.

1.3 RESEARCH OBJECTIVES

Sophisticated software has enabled aerospace companies to track satellites with accurate measure. For civilian satellite enthusiasts, there is also satellite tracking software, although these programs take a loss on accuracy. To mitigate this loss, observations of frequency shift from the transmitted radio signal of a satellite can be used to determine the orbital elements of the orbit.

The primary objective of this report is to define an algorithm which uses range rate of a satellite with respect to a ground station to determine the orbital elements of the satellite. To do this, measurements of the signal received from the satellite will be taken as the satellite travels overhead. The collection of these frequency data points will form a Doppler curve which will be used to

calculate the range rate. A least-squares algorithm will then be introduced to an orbit generator for final orbit determination.

In the next chapter, data will be extracted from the described orbit propagator. The data pulled will have contain the mentioned noise needing correction. The LMA will be used to eliminate the noisy data, and an orbit will be reproduced using the range-rate between the simulated orbiting object and a simulated ground station.

2. ORBITAL DETERMINATION FROM GROUND SITE OBSERVATIONS

To establish the trajectory of an object through space, six independent state parameters are necessary. For example, the orbit of a satellite about the earth can be determined by the cartesian state vector, which gives x, y, and z components for the radial and velocity vectors. For this experiment, observable data will be calculated from the orbit propagated from a state vector. One of these observables, range rate, will be calculated using the Doppler signal from the satellites under investigation. For comparison, additional observables will be included in alternative test cases for this orbit determination problem.

The present chapter discusses theory relevant to orbital determination using ground site observations. The first part of this chapter discusses the Julian Date system and sidereal time. These methods of timekeeping simplify later calculations. Next, the Earth-Centered Inertial reference frame is converted to a frame local to the surface of Earth. Then, the theory of orbital determination using the described independent quantities is discussed. Finally, because the trajectory of an object travelling through space can be altered by outside forces, orbital perturbations are briefly discussed.

2.1 TIME MANAGEMENT

As with most problems involving kinetics, time is a necessary component when determining the orbit of a satellite with observational data. Unlike the ubiquitous solar time, which tracks the movement of the Sun through the sky, universal time (UT) monitors the passage of the Sun through the meridian of Greenwich, London, where terrestrial longitude is defined as zero degrees.

Measuring westward from the Greenwich meridian to the local meridian, local standard time is calculated by adding one hour per each time zone passed.

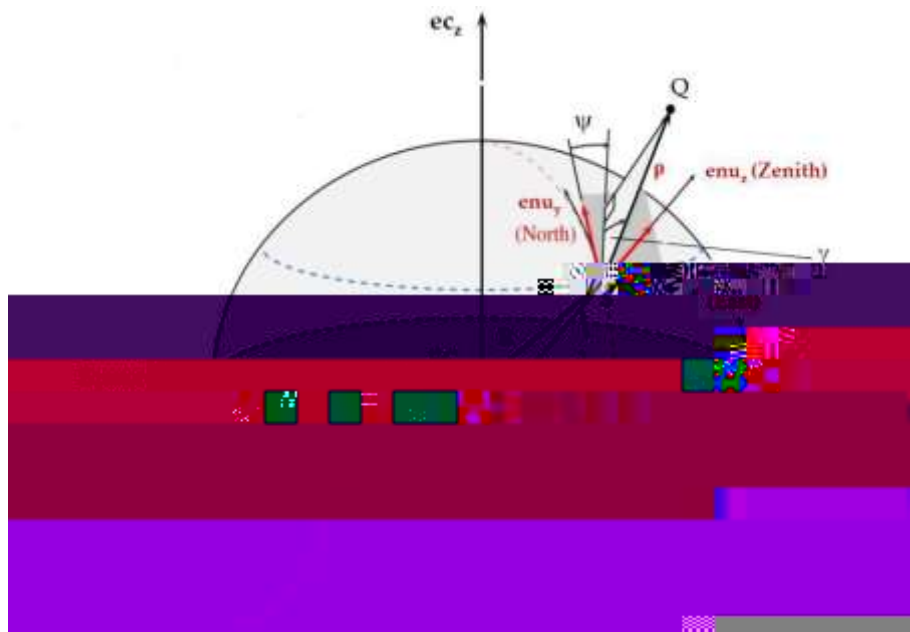


Figure 2.1 Topocentric Coordinate System ([14] as adapted by [17]).

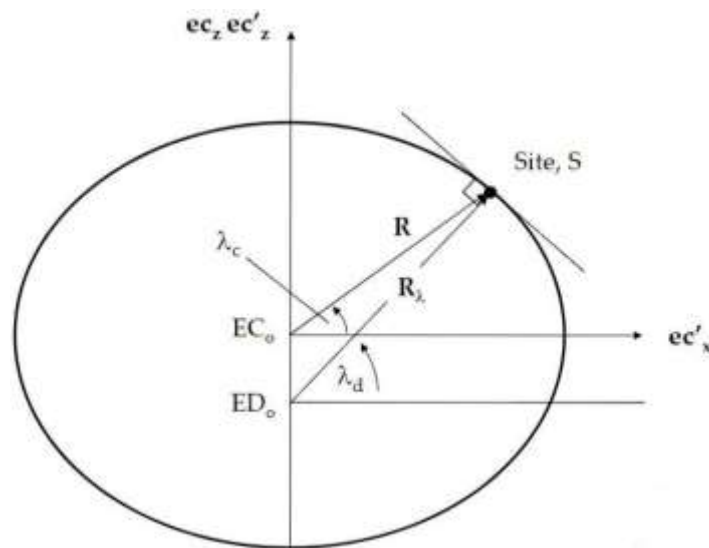


Figure 2.2 Cross-section of the earth [17].

The oblateness and eccentricity of the earth are defined, respectively, by [14]:

$$= \frac{r_0^2 \sin^2 \theta}{2} \quad (2.34)$$

where the relation between e and θ is

$$= 2 \frac{r_0^2 \sin^2 \theta}{2} \quad (2.35)$$

The distance from S_0 to S is $r_0^2 \sin^2 \theta$, where θ is defined as [14]:

$$= \frac{r_0^2 \sin^2 \theta}{2} = \frac{r_0^2 \sin^2 \theta}{(2 - e^2) \sin^2 \theta} \quad (2.36)$$

The position of S with respect to S_0 can now be defined as [14]:

$$= (R + H) \cos \theta \cos \phi + (R + H) \cos \theta \sin \phi + [(\dot{R} + \dot{H})^2 + \dot{\theta}^2] \sin \theta \quad (2.37)$$

where H refers to height of the ground station with respect to the reference ellipsoid.

2.3 THE DOPPLER EFFECT

The Doppler effect is a phenomenon that increases or decreases the observed frequency of a wave due to the relative velocity between the source of the wave and the observer. This frequency shift is most recognizable in the siren of a passing ambulance, but it can also occur in light and radio signal reception. Measuring the frequency of stellar light, astronomers can determine if a star is moving

with respect to the earth. Likewise, a received radio signal from an orbiting satellite will have a frequency shift, though steps are typically taken to correct this effect instantaneously. To attain a better understanding of the Doppler effect, the components of frequency are examined.

The velocity of a wave from a stationary source can be measured in terms of wavelength and transmitted frequency with the equation

$$= \quad (2.38)$$

If the source has a relative velocity with respect to an observer, the observed frequency is shifted from , as per the equation:

$$= (1 + \text{————}) \quad (2.39)$$

if the source is moving at velocity towards an observer, or

$$= (1 \text{ ————}) \quad (2.40)$$

In the case of signal measured from a passing satellite, is hereby referred to as the speed of light c.

With a simulated radio signal and an orbit propagator, equations (2.11) and (2.12) will be used later in this paper to determine the range-rate of the satellite.

2.4 ORBITAL PERTURBATIONS

In a basic two-body problem where the center of mass is dominated by a massive spherical body, an unaltered orbit could be achieved by the secondary body, though, in reality, none of these assumptions hold true. Adding complexity to the two-body problem, there are four forces that alter the orbital elements of a satellite: third-body perturbations, perturbations due to non-spherical planet, atmospheric drag, and solar radiation pressure [18].

The first orbit influencing force, third-body perturbations, is caused by the sun and moon. These bodies induce periodic changes to each orbital element of the satellite. Additionally, secular variations are experienced by the longitude of the ascending node and the mean anomaly due to the gravitational presence of these bodies.

The second force in this list is due to the ellipsoidal shape of the earth. While the planet is typically modelled as having a spherical shape, a better estimate shows that more mass is found along the equator, leaving a flattening effect at the poles. To accurately predict an orbit, zonal coefficients are used to form a geopotential function.

The oblateness of the earth dominates the geopotential expansion. In this expansion, the J_2 term represents perturbations caused by this flattening. This force results in secular changes in the longitude of the ascending node and the argument of perigee [18].

The third orbit perturbing force is atmospheric drag. As a body moves through a fluid, momentum is lost from the body and imparted to particles in the fluid. This exchange of momentum causes a decrease in the velocity of the body. In the case of a satellite, a decrease in velocity means orbital decay. Fluctuations in atmospheric density are caused by varying solar activity. During periods of high solar activity, altitudes in the range of 500 – 800 km can have an atmospheric density around two orders of magnitude greater than seen during low solar activity [18].

The final force in this list is caused by solar radiation pressure. In the lower atmosphere,

atmospheric drag is the most influencing force on orbital elements, but at altitudes greater than 800 km, solar radiation pressure becomes the greater force [18].

3. ORBIT SIMULATION AND DATA ACQUISITION

Paramount to the success of orbital determination is accurate data acquisition. While the data used for the present paper rely on a simulated orbit, realistic scenarios present errors which must be accounted for. The goal of this chapter is to provide the method used to calculate an accurate orbit using noisy data. Reasons for this method choice will be discussed, as well as corrections for the orbital perturbations outlined in the previous chapter.

The first part of the chapter will give a brief history of statistical orbital determination (OD). Next, a discussion on how initial orbital data is obtained using MatLab software is given. The corrections used for orbital perturbations will be examined. To accurately predict the future location of an orbiting body, all perturbations mentioned in chapter 2 must be considered. After this, the sources of data error and the procedures used to reduce these errors will be explained.

A BREIF HISTORY OF ORBITAL DETERMINATION

While astronomers have contemplated motion through space for millennia, it wasn't until Johannes Kepler (c.1610), a German mathematician, astronomer, and astrologer discovered that not all orbits are circular that true statistical OD began [14]. Kepler was a student of a wealthy astronomer, Tycho Brahe, whose belief

attributed to Gauss, who published his own works for OD methods in 1809. While Gauss and Legendre were trying to figure out who discovered LS, another hallmark achievement was accomplished.

In 1801, the Ceres comet was rediscovered after astronomers used observations to predict its location. This was the first time OD was used to locate an orbiting body [20]. Over next two decades, many mathematicians worked to refine the work done by Gauss and Legendre, though the

-

= the frequency received at the ground station

= the frequency transmitted from the satellite

= the line-of-

[20]. Vallado and McClain [20] state biases are “a constant offset from the true value.” In astrodynamics, it is common enough to assume this bias is zero. Drift is known as a slow variation to data over time. The largest contributor to drift is clock instabilities in the satellite, which can be caused by temperature differentials [20]. Due to the short windows of visibility, drift will be negligible in the present study. Noise is a statistical indication, or standard deviation of varying data around the measured average.

Noise errors can stem from several sources. The on-board oscillator can degrade accuracy of the Doppler shift measurements without short-term stability [22]. According to Bart Root (personal communication, 2018), a lecturer at the Delft University of Technology in Delft, Netherlands, this makes tracking smaller satellites (e.g., CubeSats) through one-way Doppler measurements difficult due to the cheap oscillators used. Additional sources of noise may stem from surface radio frequency (RF) emissions or other air/space vehicles.

Mathematical modeling errors happen during data processing. This can mean incorrectly entered data, typos in coding, and general misunderstanding of data field (Bart Root, personal communication, 2018). The best way to avoid modeling errors is to take care in both data recording and coding.

3.2.2 Two-Line Elements (TLEs)

As mentioned in chapter 2, six elements are needed to accurately predict an orbit. While the present study focuses on the use of range-rate information supplied through Doppler data, initial knowledge of the location of the satellite is necessary, according to Gauss, who states in his book, *Theoria Motus* (as translated in [19]), “... this problem [of accurate OD] can only be properly undertaken when an approximate knowledge of the orbit has been already attained.” Gpredict, a real-time satellite tracking application, will be used to supply TLE data to MatLab to form an initial estimate

$$S(\mathbf{x}) = \sum_{i=1}^N [f_i(\mathbf{x})]^2 \quad (3.2)$$

where N is the number of data points and \mathbf{x} is a vector of parameters $x_j, j = 1, 2, 3, \dots, n$. The vector of residuals \mathbf{f} is found by assembling the N functions $f_i(\mathbf{x})$, where $i = 1, 2, 3, \dots, N$, resulting in [24]

$$\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_N(\mathbf{x})]^T. \quad (3.3)$$

Numerous mathematicians have made alterations to Gauss's method for both better understanding of the problem and to decrease computational cost.

Also known as the damped least-squares (DLS) method, the Levenberg-Marquardt algorithm (LMA) is one such modified algorithm that solves curve fitting problems. The LMA is a combination of the steepest descent method (also known as gradient descent method) and Gauss-Newton method.

3.3.1.1 The Steepest Descent Method

Suited for general minimization problems, in the steepest descent method (SDM), parameter values are updated in the "downhill" direction (i.e., towards the minimum). This method is best suited for problems with trivial objective functions [23]. Starting with the gradient $\nabla S(\mathbf{x})$ of $S(\mathbf{x})$, the SDM steps down along the gradient [24]. Using t , the step length along the step path, it is shown that

$$\mathbf{x}(t) = \mathbf{x} - t \nabla S(\mathbf{x})$$

where $f(\mathbf{x})$ was defined in Eqn. (3.2).

The SDM uses $f(\mathbf{x} + \mathbf{t})$ in place of \mathbf{x} and iterates forward from a new position. This process is carried on until a \mathbf{t} no longer exists for Eqn. (3.4), at which point the operation has converged.

3.3.1.2 The Gauss-Newton Method

In the Gauss-Newton method (GNM), a sum-of-squares objective function is minimized. This method assumes the desired function is approximately quadratic near the optimal solution [23]. The GNM allows faster convergence than the gradient descent method when solving moderately sized problems.

The GNM takes advantage of the fact that the gradient $\nabla f(\mathbf{x})$ must be zero at the minimum. That's to say, the functions $f_j(\mathbf{x}), j = 1, 2, 3, \dots, m$, create a nonlinear set of m functions with m unknowns \mathbf{x} such that [24]

$$f_j(\mathbf{x}) = 0. \tag{3.5}$$

The solution to Eq. (3.5) lies on the local minimum or maximum of the function $f(\mathbf{x})$. Further analysis of Eqs. (3.2-3.3) suggests gradient components [24]

$$\frac{\partial f(\mathbf{x})}{\partial x_i} = 2 \sum_{j=1}^m f_j(\mathbf{x}) \frac{\partial f_j(\mathbf{x})}{\partial x_i} \tag{3.6}$$

which leads to

$$\frac{\partial f(\mathbf{x})}{\partial x_j} = \sum_{i=1}^n \frac{\partial f(\mathbf{x})}{\partial y_i} \frac{\partial y_i}{\partial x_j} \quad (3.11)$$

for each element in j . Thus, incorporating Eqns. (3.7) and (3.9),

$$\frac{\partial f(\mathbf{x})}{\partial x_j} = \sum_{i=1}^n \left[\frac{\partial f(\mathbf{x})}{\partial y_i} \frac{\partial y_i}{\partial x_j} + \frac{\partial f(\mathbf{x})}{\partial x_j} \frac{\partial y_i}{\partial x_j} \right] \quad (3.12)$$

The GNM iterates forward, using $(\mathbf{x} + \mathbf{q})$ in place of \mathbf{x} and repeats the process until the value of \mathbf{q} falls below a prescribed tolerance or

$$\|\mathbf{q} + \mathbf{q}\| < \epsilon \quad (3.13)$$

3.3.2 Marquardt's Method

Using both the gradient descent and Gauss-Newton methods, the LMA changes based on the value of an algorithmic parameter λ , as seen in the equation (adapted from [24]):

$$(\mathbf{J}^T \mathbf{J} + \lambda \mathbf{D}) \mathbf{q} = -\mathbf{J}^T \mathbf{f} \quad (3.14)$$

where:

\mathbf{J}^T = transpose of \mathbf{J}

\mathbf{D} = a diagonal matrix with positive diagonal elements

\mathbf{f} = column vector of residuals

should be increased by the same factor followed by repeating Eqn. (3.14).

Marquardt's modified LS algorithm was tested for its curve fitting capability. The decaying exponential function

$$(y, x) = a_1 + a_2 \cdot e^{-x^3}$$

In the next chapter, the Levenberg-Marquardt algorithm, as it is applied to the present study, will be

4. MARQUARDT DAMPED LEAST SQUARES FILTER DESIGN

The Levenberg-Marquardt Algorithm (LMA), also known as the Marquardt algorithm, is acclaimed for its proficiency in orbit estimation. The LMA is robust and allows for a higher degree of error in measured data if your initial estimate of the state vector is reasonable [26]. The Levenberg-Marquardt Filter (LMF) used in the present study borrows from the LMF outlined by Nash [24] with modifications presented by Transtrum and Sethna [27]. The algorithm provides a “best estimate” for state vector \mathbf{x} when provided with noisy data and an initial “estimated” state vector. As it was applied to the present investigation, data was collected from a simulated orbit, then the orbit was perturbed to provide an initial estimate to the system.

4.1 ACQUIRING DATA

Using NORAD Two-Line Element (TLE) data retrieved from Celestrak, the radial and velocity components of a given satellite are calculated using the Simplified General Perturbations 4 (SGP4) propagator, which can be found online in many computing languages. The MatLab version of the SGP4 propagator used in the present paper, written by Mahooti [28], can be found on the MathWorks website. Initially, azimuth and elevation data taken from Gpredict satellite tracking software are used to generate the “truth” state vector.

Gpredict (GP) is a free, downloadable satellite tracking application and was the original source of

problems arose when examining satellites in a larger orbit. Curtis [14] explains the Gauss method in detail and mentions that the time between the measured angles should be small. While it is possible to manipulate the output from GP to an extent, the software did not provide data with short enough time intervals. This became apparent as state vectors generated for larger orbits (e.g., orbits of NAVSTAR and MOLNIYA satellites) were too flawed to provide adequate testbeds. Learning from

—

====

$$= \frac{\dots}{\dots} \tag{4.3}$$

where

is the

brief discussion on the formation of the LMF is given, elaborating on the description given in Chapter Three. Then, modifications proposed by [27] are explained.

In the study, “A Method for the Solution of Certain Non-Linear Problems in Least Squares,” Levenberg [32] proposed damping of parameter increments to improve first-order Taylor series approximations when a flaw was noticed “standard” methods. In past procedures [32] least squares algorithms using linear approximations found updated values for estimated parameters, but the algorithm would fail if the new values were not sufficiently close to the initial estimate. This is because the algorithm may neglect higher order term, which leads to a larger sum of squares of the residuals. Thus, Levenberg [32] determined that finding function residuals under damped conditions was a beneficial alternative. This was done by including a damping parameter

steps to those of previous steps. Should the sum of squares resulting from the corrected set of parameters be greater than the previous sum of squares, the algorithm favors the SDM. Conversely, if the new sum of squares is less than the previous iteration, the method proceeds with the GNM. The algorithm, once supplied with a vector of initial estimates \mathbf{p} and recorded data, iteratively repeats the following steps [27]:

1. Calculate new data and Jacobian values based on the updated parameters.
2. Calculate new Marquardt pa

smaller problems, decreasing and increasing factors of 3 and 2, respectively, work best [27]. Nash [24] suggests making decreasing and increasing factors 0.4 and 10, respectively.

To use the indirect method for determining α , a step size Δ is first determined. The damping parameter that ensures $\| \mathbf{J}^{-1} \mathbf{r} \|$ is then found. As this method for determining α will not be used in this study, greater detail on the matter will not be provided in the present paper. For further information on this indirect method for finding α , see literature such as Moré [34]. Transtrum and Sethna [27] determined that some problems perform better using the direct method for determining α , while others favor the indirect method.

There are several options when choosing the scaling matrix \mathbf{W} . While Levenberg first determined the scaling matrix be the identity matrix \mathbf{I} [27], both Levenberg [32] and Marquardt [33] settled on using the diagonal entries of $\mathbf{J}^T \mathbf{J}$ [24]. Moré [34] determined that the optimal scaling matrix would be a diagonal matrix which updates its entries with the largest diagonal entries of $\mathbf{J}^T \mathbf{J}$ encountered through the duration of the run.

4.2.2 Gain Factor

To ensure a faster convergence using the LMF a gain factor β is used to control which corrections are accepted. This gain factor is formulated as follows (adapted from [35]):

$$\beta = \frac{F(\mathbf{x}) - F(\mathbf{x} + \mathbf{q})}{F(\mathbf{x})} \quad (4.1)$$

where $F(\mathbf{x})$ is the data function evaluated with parameter vector \mathbf{x} and $F(\mathbf{x} + \mathbf{q})$ is the data function evaluated with parameter correction vector \mathbf{q} . The denominator is evaluated as

$$\Delta x = - \frac{1}{2} \left(\frac{\partial^2 f}{\partial x^2} \right)^{-1} \left(\frac{\partial f}{\partial x} \right) \quad (4.2)$$

$$\Delta x = - \frac{1}{2} \left[2 \frac{\partial^2 f}{\partial x^2} + \left(\frac{\partial f}{\partial x} \right)^2 \right] \quad (4.3)$$

$$= \frac{1}{2} \left(\frac{\partial^2 f}{\partial x^2} \right) \quad (4.4)$$

where

$$= \tau \quad (4.5)$$

When τ falls below a predetermined value τ_0 the parameter correction is rejected and τ is increased. Otherwise, the parameter correction is accepted and τ is decreased.

4.2.3 Broyden Rank-1 Jacobian

Each time a correction is accepted to the parameter vector, the Jacobian matrix is updated so new corrections can be determined. This Jacobian matrix is typically evaluated as

$$J_{ij} = \frac{\partial f}{\partial x_j} \quad (4.6)$$

or

$$= \frac{(\mathbf{J} + \mathbf{J}_k) \mathbf{J}_k}{\mathbf{J}_k} \quad (4.7)$$

As calculating \mathbf{J} each time can become computationally expensive, Transtrum and Sethna [27] suggest using an alternative update method set forth by Broyden [36]. Broyden [36] determined that a quasi-Newton root finding method that updates \mathbf{J} with first derivatives on the first iteration, then alternates between reevaluating \mathbf{J} with a rank-1 update. This Broyden rank-1 update is written as (adapted from [36])

$$= \mathbf{J}_{k-1} + \frac{(\mathbf{J}_k - \mathbf{J}_{k-1}) (\mathbf{J}_{k-1}^{-1} \mathbf{J}_k)}{\mathbf{J}_{k-1}^{-1} \mathbf{J}_k} \quad (4.8)$$

where subscript k indicates the current step and $k-1$ represents the previous step.

4.2.4 Convergence and Stopping Criterion

If the software running the least-squares estimator is not told when it is a good place to stop, it may continue iterating indefinitely. This implies that either the parameters have converged to a solution and further iterations cease to produce worthwhile results, or the function is not solvable under the given conditions and further iterations produce worthless results. Convergence and stopping criteria are added to the least-squares program to ensure further calculations are not carried out once the criteria are met. It is suggested to use the following convergence criteria (adapted from [35]):

1. $\|\mathbf{J}_k\| < \epsilon_1$
2. $\frac{\|\mathbf{J}_k\|}{\|\mathbf{J}_k\| + \|\mathbf{J}_{k-1}\|} < \epsilon_2$, where ϵ_2 is greater than zero
3. $\|\mathbf{J}_k - \mathbf{J}_{k-1}\| < \epsilon_3$

The first criterion stops the program should the highest absolute value in the gradient vector be less than a user specified value ϵ_1 . This will be called the gradient convergence criterion. The second criterion stops the program if the highest absolute value of the correction vector divided by its counterpart in the absolute value of the parameter vector plus ϵ_2 , a small number greater than zero, is less than the user specified ϵ_2 . The third criterion stops the program should the iteration count meet or exceed some predetermined value.

4.3 BUILDING THE SOFTWARE

For this study, MatLab was used to write the orbit determination software. To use the LMF, a program was first designed to calculate the “truth” state vector from the TLE set of a given satellite. Next, a program to propagate the state vector for the pass duration of the satellite is used, and simulated data is collected. Then, a vector of perturbing elements is added to the “truth” orbit to simulate an initial estimate of the state vector. Finally, the “estimated” state vector and simulated data are passed to the LMF to find the “best estimate” of the orbit fitting the supplied data. The setup of the LMF and subroutines are set up similarly to many other programs using least squares algorithms.

Three main routines are required when testing a least-squares filter: the least-squares filter, a data acquisition function, and a testing program. The least-squares filter, in this case, the LMF, is built to handle a variety of data fitting applications. Next, a test program is created to declare a vector of initial estimated parameters, system constants, and filter options. Additionally, the test program reads a file consisting of data and times the data was taken. Finally, the data acquisition function uses the vector of parameters and the vector of times corresponding to the times of real data measurement to simulate data.

4.3.1 Test Program

In the algorithm test program, “Orbit100.m”, users can alter testing options before state vectors are produced. Satellites that are currently available for testing are NAVSTAR-77, MOLNIYA 3-50, and the ISS. The test program is broken up into seven sections.

Satellite selection and orbit propagator options can be set in the first section of this program. The TLE set for the selected satellite downloads automatically when the program is started. While only the mentioned satellites are available for the user’s convenience, additional satellites can be tested with a text file containing the TLE of the desired satellite. Also, in the first section, the choice of which observation set to use can be made.

—

this paper is on using Doppler data (range rate) for orbit estimation, these additional cases will serve well for comparison. Methods used to calculate azimuth and elevation were covered in Section 4.1.

Section two of the main program allows changing of options used in the LMF. The options included are:

- bdx - Small perturbation value used for Jacobian calculation ($\delta \mathbf{x}$ in Eqn. 4.7)
- lambda - The Marquardt scaling parameter
- incr - A value to increase lambda
- decr - A value to decrease lambda
- maxIter - Determines maximum iteration count
- eps1 - Gradient convergence criteria ¹
- eps2 - Parameter convergence criteria ²
- eps3 - Root mean square convergence criteria ³
- eps4 - state correction acceptance criteria ⁴

In section three of “Orbit100.m”, the start time for the satellite pass is entered. Satellite tracking software, such as GP, or internet databases can be used to find satellite flyby times. Time is entered in Universal Time (UT). In section four of “Orbit100.m”, the user can change the position of the ground site. Currently, the simulated ground site shares the location of San Jose State University. Constants and coefficients are read into the program from exterior files in sections 5 and 6 of “Orbit100.m.”

In sections 5 and 6 of “Orbit100.m,” files containing constants and coefficients used in the orbit generator are loaded. These files contain Earth Orientation Parameters (EOP), the GRACE gravity model (GGM03S), and NASA JPL Development Ephemerides (DE430). Finally, in section 7 of Orbit100.m, data is generated, and the “estimated” orbit is produced.

4.3.2 Orbit and Data Generator

The orbit propagator used in this study was part of a package put together by Meysam Mahooti. The unaltered version of “High Precision Orbit Propagator” (HPOP) can be found on the MathWorks File Exchange. Mahooti’s HPOP was chosen for its ability to model the variety of forces that act on Earth-orbiting satellites. These forces are:

- Gravity field of the earth
- Gravity of the solar system planets
- Drag effect
- Solar radiation pressure
- Solid Earth tides
- Ocean tides

The ordinary differential equation solver used in HPOP is the Radau IIA, which is derived by Hairer and Wanner [31]. Radau IIA is derived from implicit Runge-Kutta methods that offer step size control and continuous output.

The programs “get_obs.m” and “get_data.m” are called to propagate the state vector \mathbf{x} to times determined by step size and the number of observation sets. The “truth” state vector is propagated in “get_obs.m”, where azimuth, right ascension, and/or range rate data are calculated. White noise is added to this data to simulate data that may be picked up by ground site hardware. Similarly, “get_data.m”, used throughout the LMF, propagates the “estimated” orbit and records data at the times used for observations. As the LMF searches for state vectors with a better fit, “get_data.m” is used to calculate data in the generated orbits.

4.3.3 Least-Squares Filter

Let $\mathbf{f} =$

Calculate $=$

Calculate $= ($

5. RESULTS AND ANALYSIS

In this chapter, the Levenberg-Marquardt Damped Least-Squares algorithm, outlined in previous chapters, is tested for its ability to provide the “best estimate” for the state vector of an orbit at a given epoch. To generate the “truth” orbit of a satellite, the Two-Line Element set of the desired satellite is downloaded into a text file, then it is processed using an SGP4 propagator and the cartesian “truth” state vector is produced. With the state vector acquired, an ephemeris is generated, and simulated observations of range-rate, azimuth, and elevation can be calculated. In the various cases run in this chapter, different combinations of these data are studied. In testing of real data, range-rate would be calculated using the Doppler shift of the carrier signal from the satellite. For the simulated case, it is assumed that the Doppler data has already been processed, giving range rate at each respective point in the orbit. A simulated ground site is used as an observation point. The simulated ground site shares the latitude and longitude of San Jose State University:

37.3352 , 121.8811 . Next, the state vector of the “estimated” orbit is generated.

To generate simulated observations, an initial “estimated” state vector is created by perturbing the initial “truth” state vector. This “estimated” state vector is propagated, and the desired parameters are calculated using methods described in the previous chapter. To this data, white, zero-mean, Gaussian noise is added. The standard deviation for the noise added to each parameter can be seen in Table 4.1. The standard deviation for azimuth and elevation are “realistic” for satellite tracking

Table 5.1 Gaussian Measurement Noise Standard Deviation

km. The larger orbit gives the NAVSTAR satellite an orbital period of about 12 hours. The largest

Table 5.2 Keplerian state vectors of test case satellites.

	ISS	NAVSTAR-77	MOLNIYA 3-50
$h(\text{---})$	52029.77	102898.1	71305.23
e	0.001259		

The “observation” parameters were chosen due to their observability. According to Folcik [29], typical observations used for satellite orbit estimation are made up of angular optical observations and radar observations. The angular optical observations consist of right ascension and declination measurements made against the background of stars. While radar observations also include two angle measurements, azimuth and elevation, they also include range and range-rate measurements. The present analysis assumes that for each case, direct determination of range-rate is not feasible. The number of observations used for each satellite varies based on the size of the orbit. While the smaller orbit of the ISS requires fewer data points (25 were used in the present analysis) to allow the LMF to converge, Hunter [17] advises using an increased number of observations for the larger orbits of NAVSTAR-77 and MOLNIYA 3-50 to better capture their curvature. An increased step size allows the full pass of the satellite to be captured without creating abundant data points to calculate. For the both the NAVSTAR and MOLNIYA satellites, 100 observations were simulated. The output from the LMF is indicative of how well the filter performed. While optimizing a program could entail limiting the amount of function calls, it is expected that running the LMF with increased step size and observation counts will take longer to converge than the more circular ISS case. For the ISS, a step size of 25 seconds was chosen. With the 25 sets of observations, this step size allowed the filter access to the full pass duration of 10.5 minutes. To cover the full pass durations, the NAVSTAR and MOLNIYA orbits were given step sizes of 252 seconds and 360 seconds, respectively.

The output of the LMF is a set of four vectors. These vectors are

Truth - the initial “truth” state vector retrieved from TLE of the satellite

Estimated - “estimated” initial state vector. In the case of this paper, this state vector was

“estimated” by perturbing the “truth” vector. Each satellite initial state vector is perturbed by

As seen in Tables (5.4-5.6), the LMF performed remarkably well. Each of the estimated state vectors were correctly fitted to the noisy data. The data collected from the orbit of the propagated “estimated” state vector, as well as noisy data collected from the “truth” orbit can be found in Appendix A.

5.1.2 Angles Only Case

For the second set of cases, angles of azimuth and elevation (with added noise) will serve as the “observed” data. This case is included in the present analysis to determine if including range rate information has a significant impact on the performance of the LMF. Tables (5.7-5.9) show the results of using angles only with the LMF.

Table 5.7 ISS Results from Azimuth & Elevation.

	Truth	Predicted	Correction	Final
x (km)	-6197.3	-6197.8	-0.5	-6197.3
y (km)	-1366.8	-1368.8	2.0	-1366.8

Table 5.9 MOLNIYA 3-50 Results from Azimuth & Elevation.

	Truth	Predicted	Correction	Final
x (km)	-19347.0	-19341.0	-6.0	-19347.0
y (km)	1824.3	1822.3	2.0	1824.3
z (km)	14893.2	14898.2	-5.0	14893.2
(km/s)	-1.1151	-1.1121	-2.99	-1.1150
(km/s)	-1.6175	-1.6165	-0.99	-1.6175
(km/s)	3.6997	3.6972	2.49	3.6997
t = 2805.202 s				

5.1.1 Range Rate Only Case

For the final set of cases, range rate data will be the only observations used for the LMF. The results of this case are seen in Tables (5.10-5.12).

Table 5.4 ISS Results from only.

	Truth	Predicted	Correction	Final
x (km)	-6197.3			

$$PV = [-4 \quad -2.5 \quad 3 \quad -1.6 \quad 10 \quad 1 \quad 10 \quad -2 \quad \dots],$$

the LMF was able to converge.

Table 5.5 NAVSTAR-77 Results from only (using updated PV).

	Truth	Predicted	Correction	Final
x (km)	14277.2	-14273.2	4.0	-14277.2
y (km)	15437.6	-154351	2.5	-14437.3
z (km)	-16214.8	16216.8	-3.0	16214.8
(km/s)	-3.1686	3.167	1.61	3.1690
(km/s)	0.7177	-0.7067	-0.99	-0.7075

NAVSTAR-77. Through the iterations, it was observed that each new change was rejected, and continued to grow until it hit a max at 10^{-7} . Potential changes were repeatedly rejected even as the potential corrections dwindled down to zero. It was believed that the initial guess may have been too far off for this case (as the literature suggests the filter relies on an estimate close to the solution), so the **PV** was decreased in an attempt to remedy the failure. Indeed, the decrease in **PV** allowed the filter to find the solution. Previous builds of the LMF ran into problems as well.

The results found in this section are the outcome of a fourth generation build of the LMF. Previous iterations proved successful (see Appendix C for results) for the ISS, which converged, and NAVSTAR, which gained successful, albeit lacking, corrections, but it failed to properly correct the MOLNIYA cases. Additionally, even though the ISS and NAVSTAR cases were able to be corrected, there was still room for improvement to the solution, but additional observations only served to exacerbate the error in the final estimate. This failure was prominent in the MOLNIYA case as it was believed that adding data from an additional pass would fix the problem, but the added pass only increased computational cost with zero benefit to the solution. The latest build (used for the results in this chapter) allowed convergence of each of the satellite cases by making several modifications to the program structure.

In previous LMF software designs created in this study, the least-squares algorithm and orbit generator were grouped into one program. While this design can work, as it did for the ISS and NAVSTAR cases, it became difficult to track variable usage. For example, one section of the program split a time of the format HH:MM:SS, where HH, MM, and SS are hours, minutes, and seconds, respectively. The hours, minutes, and seconds were saved to a time vector with variables {hr min sec}. When the output from the program was far from what was expected, it took many hours before the mistake was found. In this case, saving a number under the variable “min” caused

equations that were meant to find the minimum (using the `min.m` function) to output erroneous solutions. The use of “sec” as a variable also caused errors when calling external functions as MatLab uses “sec” to find the secant of an angle. Though MatLab typically spots these types of errors and notifies the user, in this case the only hint of the error was in the output of the program. This was one of many instances that urged the reformatting of the software. Thus, the LMF and orbit/data generator were built into separate functions, leaving the original file to serve as a testing function. Once this procedure was complete, each case was able to converge.

As was expected, the ISS was able to converge much faster than the other two cases. This relatively quick convergence is the product of fewer necessary observations and smaller step sizes. Using this ideology, it was surprising to find the MOLNIYA cases converging about forty minutes faster (averaged) than the NAVSTAR in the range rate only and angles only cases. This may be due to the eccentricity of the Molinaya orbit. Future work will investigate this matter further.

To understand the full capabilities of the LMF used in the paper, future work should involve the testing of real satellite data. While adding noise to simulated data gives a feel for what could be seen at a ground site, real observations would challenge the algorithm and discovering the trajectory of a real satellite with a least-squares algorithm would be all the more rewarding.

included in the analysis. The first additional case had data comprised of range rate, azimuth, and elevation. Because this data set had the most data type, poor resolution in the range rate only case could be quantified. The second added case used only azimuth and elevation angles. Should this case have outperformed the previous case and the range rate only case, an understanding of how the range rate impacted the results could be acquired.

Throughout the building of the LMF software, many iterations of the filter were attempted and failed. Initial builds allowed convergence on the orbit of the ISS and at least some correction for NAVSTAR, but it diverged consistently with the Molinaya orbit. When the addition of data from a subsequent pass only made state vector estimates worse, the program was overhauled and netted positive results. In addition to reformatting the software, several additions were made to improve performance of the filter.

6.1 FUTURE WORK

The successful outcome of this filter begs that it be used with real data. Future work should test the performance of the filter when given processed Doppler data. Initial testing should include data from satellites with known trajectories. If the filter proves successful, more difficult targets could be studied. Should it be desired, several modifications could still be made to the software.

While the filter has seen successful testing with simulated data, the Orbit Propagator has not been tested for its accuracy. Indeed, the OP simulated orbital perturbations, i.e., J_2 , b , J_2 , b , 2 792 reW*nBT/F2 12 Tf1

References

1. Deng, L., Sun, X., and Han, C., Analysis and Comparison on UKF and BLS for Orbit Determination, Proc., *AAS/AIAA 2015 Astrodynamics Specialists Conf.*, American Astronautical Society (AAS), Vail, 9-13 Aug. 2015.
2. Bordi, J.J., Analysis of the Precise Range and Range-Rate Equipment (PRARE) and Application to Precise Orbit Determination, May 1999.
3. Amiri, S., and Mehdipour, M., Accurate Doppler Frequency Shift Estimation for any Satellite Orbit, *RAST 2007 3rd International Conference on Recent Advances in Space Technologies*, Recent Advances in Space Technologies (RAST), Istanbul, Turkey, 14-16 June 2007.
4. Stolarski, M., and Wo niak, G., Estimation of PW-Sat Satellite Orbit Based on Doppler Effect, *Proc. SPIE 8454, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2012, 84540H*, 7 November 2012.
doi: 10.1117/12.2000192
5. Tabakovic, Z., Doppler Effect in Non-GSO Satellite Propagation. *IEEE/AP 2000 Millennium Conference on Antennas and Propagation*, Davos, Switzerland, 9-14 May 2000.
6. Agostino, M. D., Manzano, A., and Marucco, G., Doppler Measurement Integration for Kinematic Real-Time GPS Positioning, *Applied Geomatics*, 2(4), Dec. 2010, pp. 155-162.
doi:10.1007/s12518-010-0031-z
7. Ialongo, G., Method of Doppler Data Processing For Orbit Determination, TR-0066(5110-01)-5, Oct. 1969.

9. Kirschner, S., Samii, M., Broaddus, S., and Doll, C., Preliminary Orbit Determination System for Tracking and Data Relay Satellite System-Tracked Target Spacecraft using the Homotopy

Continuation Method,ETQq0.00000912 0 612 792 reW*nBT/F2 12 Tf. 792 reW*n-7.124 657.94 Tm0.2 g0.2 G

date-system-chosen-advanced>, June 2018.

17. Hunter, J., *Orbital Mechanics Course Reader*, 2018.

18. Michele, R., Position in an Elliptical Orbit, Aerospace Engineering, URL:

28. Mahooti, M., *SGP4*, MathWorks, URL:

htt

APPENDIX A. SIMULATED DATA

ISS MEASUREMENTS FROM TRUTH ORBIT

Least-squares orbit determination

Measurements from "truth" orbit

Date	UTC	Az(deg)	El(deg)	Range rate (km/s)
------	-----	---------	---------	-------------------

ISS MEASUREMENTS WITH ADDED NOISE

Least-squares orbit determination

Measurements with added noise

Date	UTC	Az(deg)
------	-----	---------

NAVSTAR-77 MEASUREMENTS FROM TRUTH ORBIT

Least-

2019/05/12	21:08:34.000	131.744	32.465	0.064
2019/05/12	21:12:46.000	130.072	32.584	0.041
2019/05/12	21:16:58.000	128.387	32.648	0.018
2019/05/12	21:21:10.000	126.690	32.658	-0.004
2019/05/12	21:25:22.000	124.986	32.614	-0.027
2019/05/12	21:29:34.000	123.277	32.518	-0.049
2019/05/12	21:33:46.000	121.566	32.369	-0.070
2019/05/12	21:37:58.000	119.856	32.171	-0.091
2019/05/12	21:42:10.000	118.149	31.922	-0.112
2019/05/12	21:46:22.000	116.448	31.625	-0.132
2019/05/12	21:50:34.000	114.753	31.282	-0.152
2019/05/12	21:54:46.000	113.069	30.893	-0.172
2019/05/12	21:58:58.000	111.395	30.460	-0.190
2019/05/12	22:03:10.000	109.733	29.986	-0.208
2019/05/12	22:07:22.000			

2019/05/13 00:25:58.000	58.091	4.369	-0.247
2019/05/13 00:30:10.000	56.523	3.830	-0.229
2019/05/13 00:34:22.000	54.947	3.333	-0.209
2019/05/13 00:38:34.000	53.363	2.879	-

NAVSTAR MEASUREMENTS WITH ADDED NOISE

Least-squares orbit determination

Measurements with added noise

Date	UTC	Az(deg)	El(deg)	Range rate (km/s)
2019/05/12	18:16:22.000	169.060	-11.407	0.716
2019/05/12	18:20:34.000	168.896	-9.676	0.715
2019/05/12	18:24:46.000	168.700	-7.957	0.714
2019/05/12	18:28:58.000	168.482	-6.268	0.712
2019/05/12	18:33:10.000	168.231	-4.608	0.708
2019/05/12	18:37:22.000	167.937	-2.980	0.703
2019/05/12	18:41:34.000	167.633	-1.360	0.698
2019/05/12	18:45:46.000	167.271	0.214	0.691
2019/05/12	18:49:58.000	166.908	1.771	0.683
2019/05/12	18:54:10.000	166.479	3.291	0.674
2019/05/12	18:58:22.000	166.030	4.800	0.665
2019/05/12	19:02:34.000	165.544	6.269	0.654
2019/05/12	19:06:46.000	165.018	7.714	0.643
2019/05/12	19:10:58.000	164.460	9.114	0.630
2019/05/12	19:15:10.000	163.856	10.499	0.617
2019/05/12	19:19:22.000	163.215	11.847	0.603
2019/05/12	19:23:34.000	162.536	13.171	0.588
2019/05/12	19:27:46.000	161.802	14.445	

2019/05/12	21:12:46.000	130.075	32.579	0.041
2019/05/12	21:16:58.000	128.389	32.642	0.018
2019/05/12	21:21:10.000	126.692	32.660	-0.004
2019/05/12	21:25:22.000	124.991	32.620	-0.027
2019/05/12	21:29:34.000	123.281	32.512	-0.048
2019/05/12	21:33:46.000	121.571	32.364	-0.070
2019/05/12	21:37:58.000	119.851	32.185	-0.091
2019/05/12	21:42:10.000	118.148	31.920	-0.112
2019/05/12	21:46:22.000	116.450	31.621	-0.132
2019/05/12	21:50:34.000	114.756	31.281	-0.152
2019/05/12	21:54:46.000	113.068	30.890	-0.172
2019/05/12	21:58:58.000	111.395	30.455	-0.190
2019/05/12	22:03:10.000	109.738	29.983	-0.208
2019/05/12	22:07:22.000	108.082	29.467	-0.226
2019/05/12	22:11:34.000	106.448	28.928	-0.243
2019/05/12	22:15:46.000	104.833	28.321	-0.259
2019/05/12	22:19:58.000	103.221	27.708	-0.274
2019/05/12	22:24:10.000	101.630	27.054	-0.288
2019/05/12	22:28:22.000	100.056	26.377	-0.302
2019/05/12	22:32:34.000	98.496	25.654	-0.315
2019/05/12	22:36:46.000	96.953	24.917	-0.327
2019/05/12	22:40:58.000	95.418	24.170	-0.338
2019/05/12	22:45:10.000	93.897	23.386	-0.347
2019/05/12	22:49:22.000	92.388	22.580	-0.356
2019/05/12	22:53:34.000	90.894	21.756	-0.364
2019/05/12	22:57:46.000	89.412	20.935	-0.371
2019/05/12	23:01:58.000	87.919	20.089	-0.377
2019/05/12	23:06:10.000	86.435	19.228	-0.382
2019/05/12	23:10:22.000	84.979	18.370	-0.385
2019/05/12	23:14:34.000	83.502	17.511	-0.388
2019/05/12	23:18:46.000	82.038	16.645	-0.389
2019/05/12	23:22:58.000	80.583	15.772	-0.389
2019/05/12	23:27:10.000	79.125	14.903	-0.388
2019/05/12	23:31:22.000	77.657	14.050	-0.386
2019/05/12	23:35:34.000	76.197	13.193	-0.382
2019/05/12	23:39:46.000	74.735	12.347	-0.377
2019/05/12	23:43:58.000	73.255	11.512	-0.371
2019/05/12	23:48:10.000	71.780	10.681	-0.364
2019/05/12	23:52:22.000	70.291	9.881	-0.356
2019/05/12	23:56:34.000	68.802	9.108	-0.346
2019/05/13	00:00:46.000	67.303	8.336	-0.335
2019/05/13	00:04:58.000	65.785	7.609	-0.324
2019/05/13	00:09:10.000	64.268	6.891	-0.310
2019/05/13	00:13:22.000	62.737	6.212	-0.296
2019/05/13	00:17:34.000	61.203	5.560	-0.281
2019/05/13	00:21:46.000	59.646	4.947	-0.264
2019/05/13	00:25:58.000	58.092	4.369	-0.247

2019/05/13 00:30:10.000	56.527	3.843	-0.229
2019/05/13 00:34:22.000	54.953	3.327	-0.209
2019/05/13 00:38:34.000	53.361	2.870	-0.189
2019/05/13 00:42:46.000	51.768	2.478	-0.168

MOLNIYA 3-50 MEASUREMENTS FROM TRUTH ORBIT

Least-squares orbit determination

Measurements

Date	UTC	Az(deg)	El(deg)	Range rate (km/s)
2019/05/13	03:08:26.000			

2019/05/13	07:20:26.000	20.062	58.062	0.129
2019/05/13	07:26:26.000	20.077	58.072	0.077
2019/05/13	07:32:26.000	20.110	58.096	0.026
2019/05/13	07:38:26.000	20.159	58.134	-0.026
2019/05/13	07:44:26.000	20.226	58.187	-0.078
2019/05/13	07:50:26.000	20.310	58.255	-0.130
2019/05/13	07:56:26.000	20.410	58.337	-0.182
2019/05/13	08:02:26.000	20.528	58.433	-0.234
2019/05/13	08:08:26.000	20.661	58.545	-0.286
2019/05/13	08:14:26.000	20.812	58.672	-0.338
2019/05/13	08:20:26.000	20.980	58.814	-0.391
2019/05/13	08:26:26.000	21.165	58.972	-0.443
2019/05/13	08:32:26.000	21.368	59.145	-0.496
2019/05/13	08:38:26.000	21.589	59.334	-0.550
2019/05/13	08:44:26.000	21.830	59.539	-0.603
2019/05/13	08:50:26.000	22.090	59.760	-0.657
2019/05/13	08:56:26.000	22.372	59.998	-0.712
2019/05/13	09:02:26.000	22.676	60.253	-0.767
2019/05/13	09:08:26.000	23.003	60.525	-0.822
2019/05/13	09:14:26.000	23.357	60.815	-0.878
2019/05/13	09:20:26.000	23.737	61.122	-0.934
2019/05/13	09:26:26.000	24.147	61.448	-0.991
2019/05/13	09:32:26.000	24.590	61.793	-1.049
2019/05/13	09:38:26.000	25.069	62.156	-1.108
2019/05/13	09:44:26.000	25.588	62.540	-1.167
2019/05/13	09:50:26.000	26.151	62.943	-1.227
2019/05/13	09:56:26.000	26.763	63.366	-1.288
2019/05/13	10:02:26.000	27.431	63.811	-1.349
2019/05/13	10:08:26.000	28.161	64.276	-1.412
2019/05/13	10:14:26.000	28.963	64.764	-1.476
2019/05/13	10:20:26.000	29.848	65.273	-1.541
2019/05/13	10:26:26.000	30.826	65.804	-1.607
2019/05/13	10:32:26.000	31.915		

2019/05/13	12:02:26.000	87.480	73.134	-2.843
2019/05/13	12:08:26.000	95.617	72.266	-2.919
2019/05/13	12:14:26.000	103.715	70.846	-2.988
2019/05/13	12:20:26.000	111.410	68.764	-3.043
2019/05/13	12:26:26.000	118.423	65.903	-3.076
2019/05/13	12:32:26.000	124.598	62.122	-3.076
2019/05/13	12:38:26.000	129.894	57.236	-3.020
2019/05/13	12:44:26.000	134.343	50.999	-2.879
2019/05/13	12:50:26.000	138.018	43.094	-2.604
2019/05/13	12:56:26.000	140.997	33.164	-2.130
2019/05/13	13:02:26.000	143.350	20.932	-1.389

MOLNIYA 3-50 MEASUREMENTS WITH ADDED NOISE

Least-squares orbit determination

Measurements

Date	UTC	Az(deg)	El(deg)	Range rate (km/s)
2019/05/13	03:08:26.000	76.298	70.308	2.919
2019/05/13	03:14:26.000	69.778		

2019/05/13	07:20:26.000	20.075	58.059	0.121
2019/05/13	07:26:26.000	20.064	58.068	0.074
2019/05/13	07:32:26.000	20.104	58.106	0.031
2019/05/13				

2019/05/13	12:02:26.000	87.479	73.127	-2.841
2019/05/13	12:08:26.000	95.622	72.262	-2.933
2019/05/13	12:14:26.000	103.718	70.862	-2.985
2019/05/13	12:20:26.000	111.411	68.766	-3.038
2019/05/13	12:26:26.000	118.433	65.911	-3.077
2019/05/13	12:32:26.000	124.598	62.123	-3.080
2019/05/13	12:38:26.000	129.899	57.237	-3.013
2019/05/13	12:44:26.000	134.342	51.007	-2.888
2019/05/13	12:50:26.000	138.027	43.086	-2.609
2019/05/13				

MOLNIYA 3-50 MEASUREMENTS FROM ESTIMATED ORBIT

Least-squares orbit determination

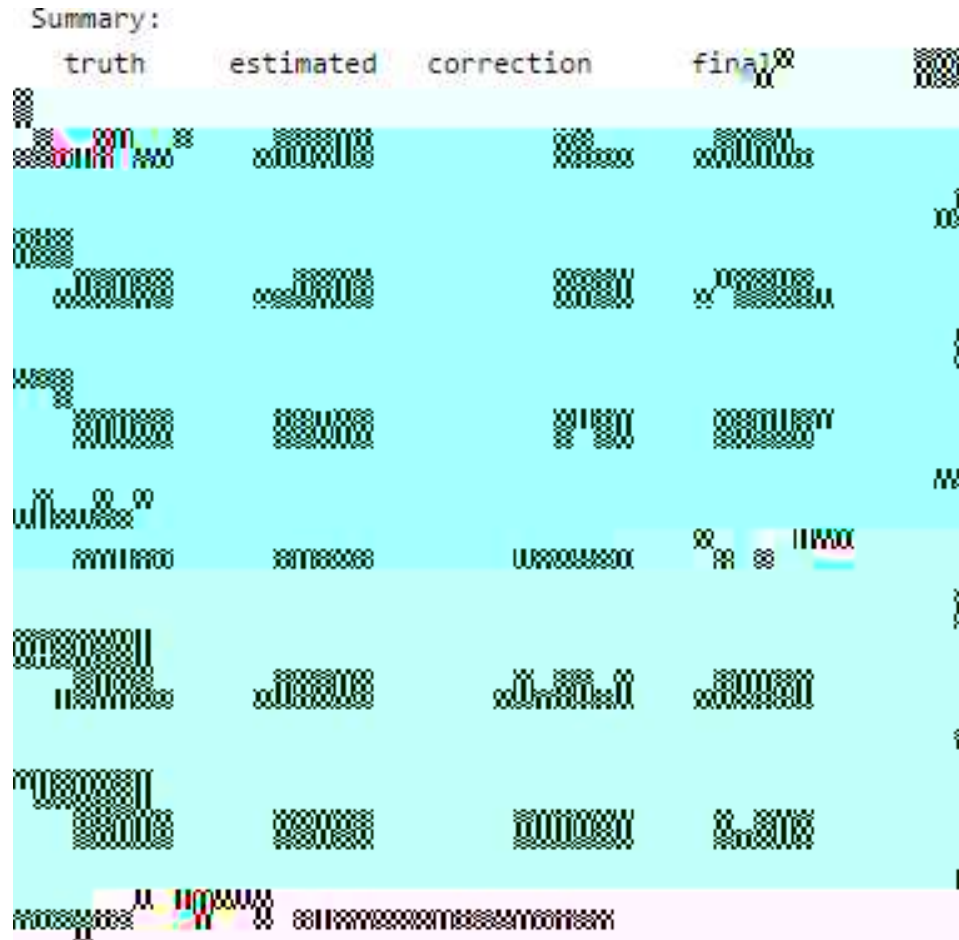
Measurements

Date	UTC	Az(deg)	El(deg)	Range rate (km/s)
2019/05/13	03:08:26.000	76.226	70.299	2.915
2019/05/13	03:14:26.000	69.714		

2019/05/13	07:20:26.000	20.143	58.004	0.121
2019/05/13	07:26:26.000	20.164	58.013	0.068
2019/05/13	07:32:26.000	20.203	58.036	0.016
2019/05/13				

APPENDIX B. SIMULATION RESULTS

ISS RANGE RATE ONLY



ISS ANGLES ONLY

Summary:

	truth	estimated	con	final
x[km]	-6197.3	-6196.8	-0.5	-6
y[km]	-1366.8	-1368.8	2.0	-1
z[km]	2416.8	2417.8	-1.0	2
vxv[km/s]	3.0836	3.0823	1.2900	3
vyv[km/s]	-4.5285	-4.5275	-0.9900	-4
vzv[km/s]	5.3558	5.355	5.3557	

Elapsed time is 90.013438 seconds.

ISS RANGE RATE AND ANGLES

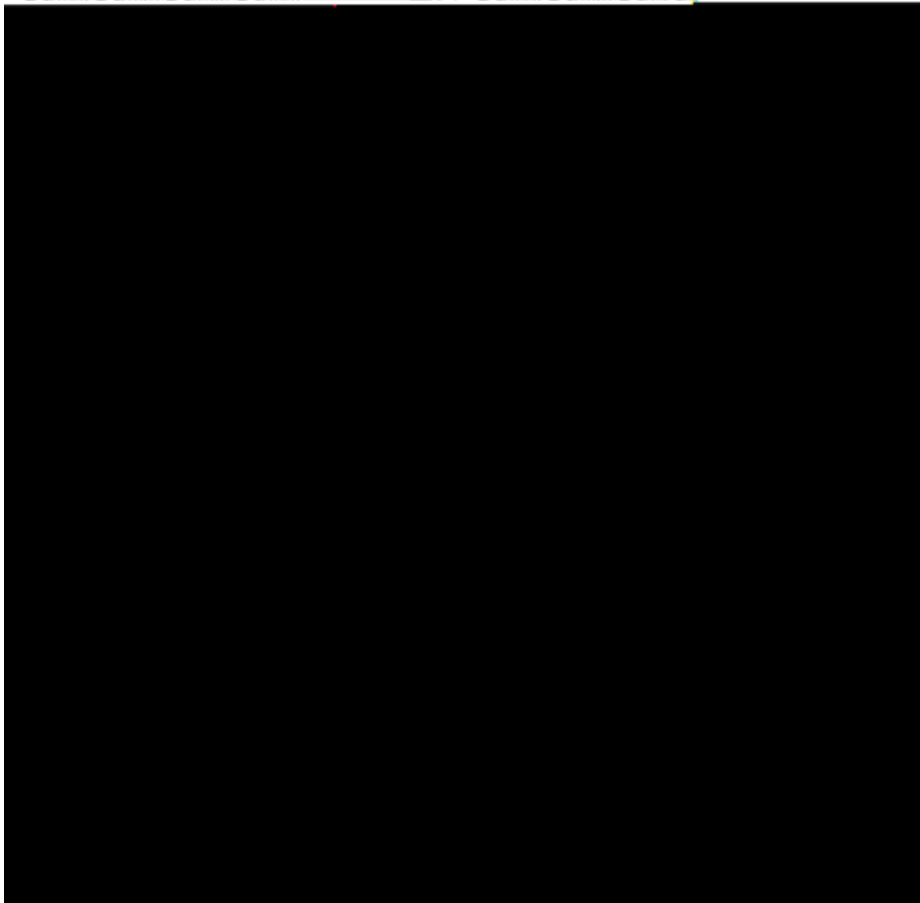
Summary:

	truth	estimated	con	final
x[km]	-6197.6	-6197.1	-0.5	-6
y[km]	-1368.2	-1368.2	2.0	-1
z[km]	2416.3	2417.3	-1.0	2
vxv[km/s]	3.0828	3.0815	0.0013	3
vyv[km/s]	5.3851	5.3856	0.0005	5

Elapse time is 446.626837 seconds.

NAVSTAR RANGE RATE ONLY

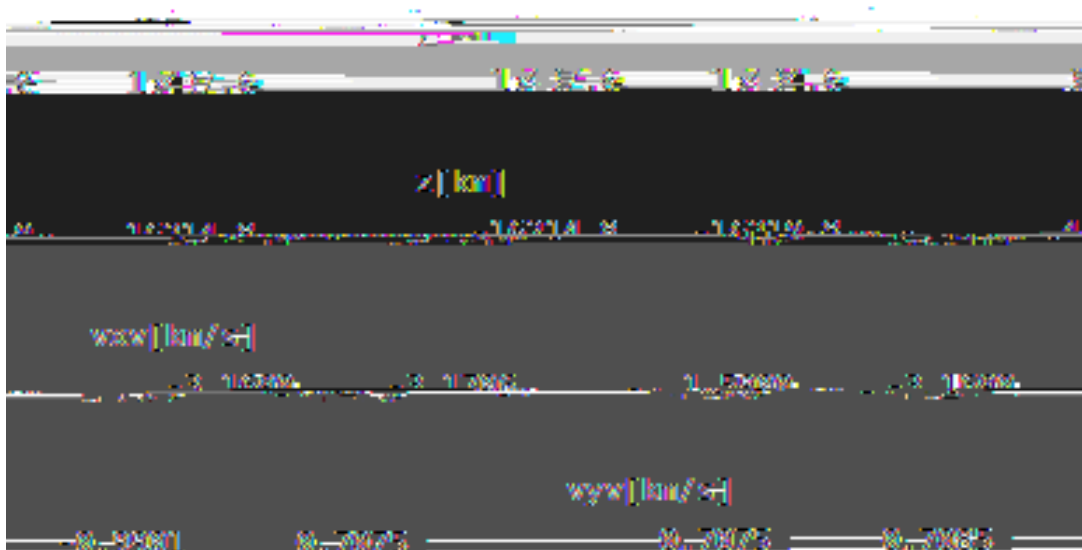
Summary:



NAVSTAR ANGLES ONLY

Summary:

	truth	estimated	correction	final
x[km]	14277.2	14272.2	5.0	14277.2



	truth	estimated	correction	final
vzv[km/s]	2.4900	-2.1149	-2.1149	-2.1174

seconds Flanged time is 5835.171982

NAVSTAR RANGE RATE AND ANGLES

Summary:

	truth	estimated	correction	final
x[km]	16217.8	16218.8	1.0	16217.8
vx[m/s]	0.7875	0.7875	0.0000	0.7875
vyv[m/s]	0.7875	0.7875	0.0000	0.7875
vzv[m/s]	0.7875	0.7875	0.0000	0.7875

MOLNIYA RANGE RATE ONLY

Summary:

x, km				
11.4	19339.9	-6.2	-19341.9	19341.9
y, km				
2.8	1826.6	1826.6	1826.6	1824.6
z, km				
6168	-2.9922	-1.6178	-1.6178	-1.6178
vxv [km/s]				
2.2	1.7213	1.7213	1.6988	2.51
vyv, km/s				
6168	-2.9922	-1.6178	-1.6178	-1.6178
vzv, km/s				
2.2	1.7213	1.7213	1.6988	2.51

MOLNIYA ANGLES ONLY

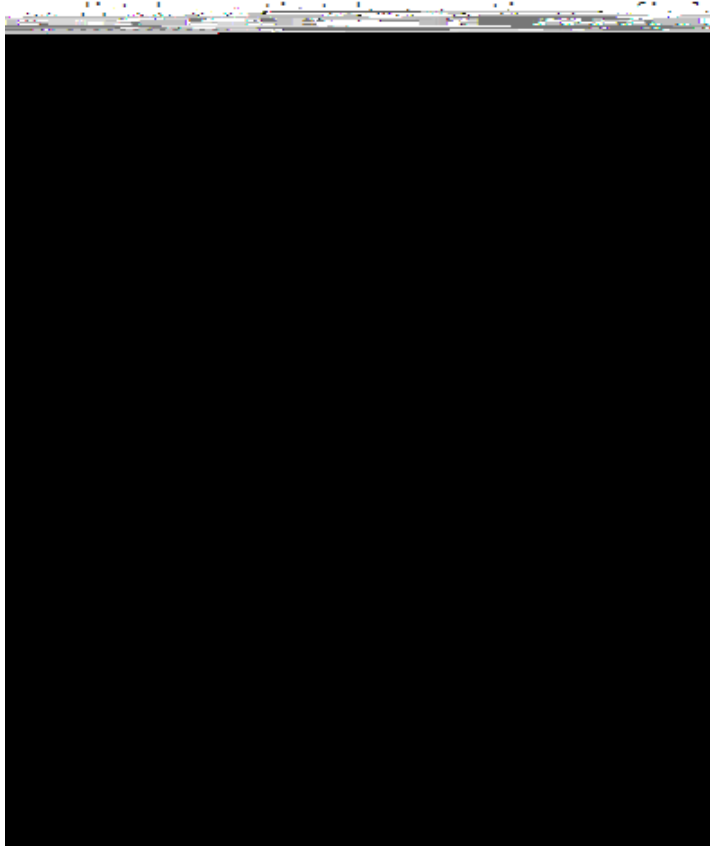


MOLNIYA RANGE RATE AND ANGLES

APPENDIX C. RESULTS FROM PREVIOUS BUILD OF LMF

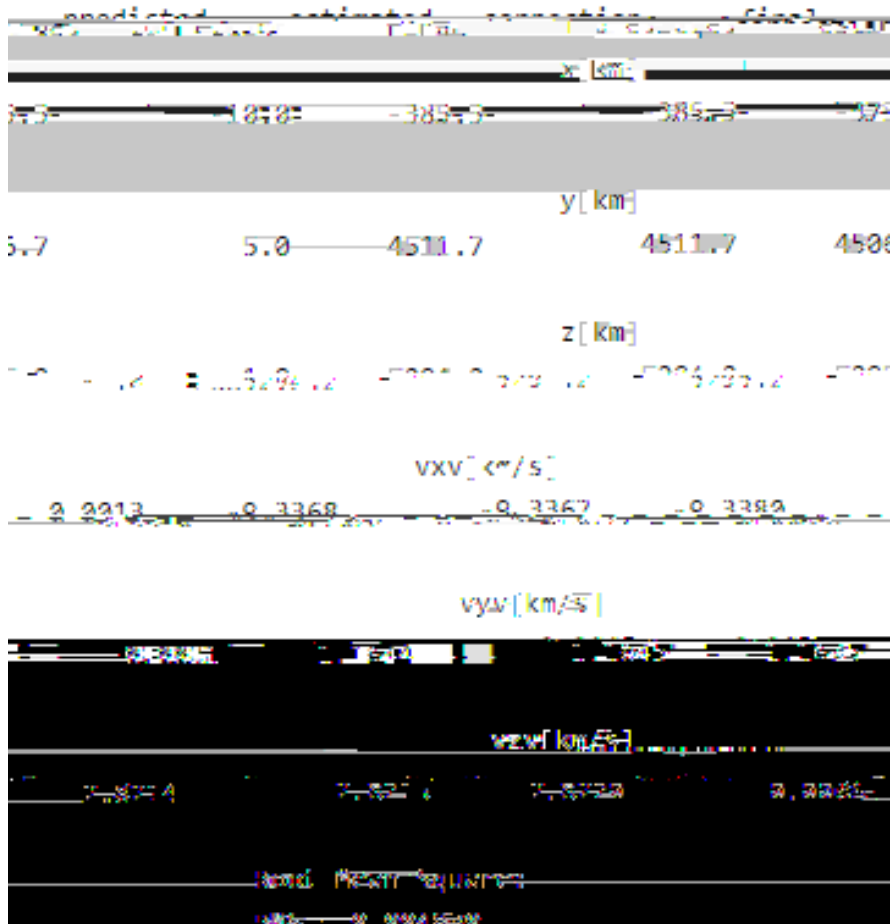
ISS using 10 observations of range rate

Summary:



ISS using 10 observations using range rate, Azimuth and elevation

Summary:

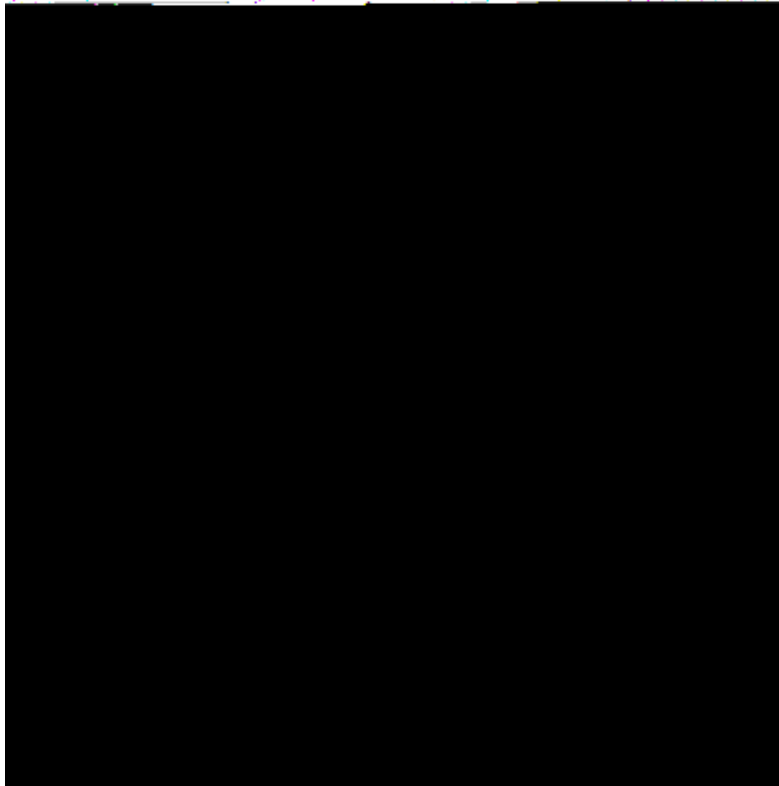


ISS using 20 observations with range rate only

Summary:

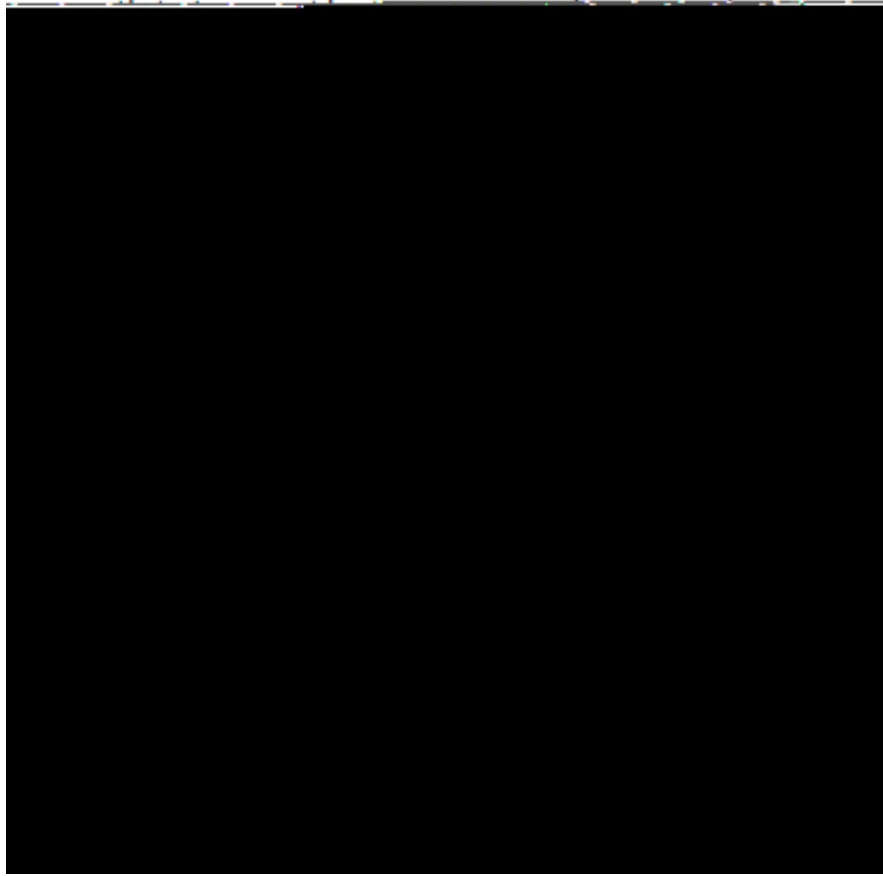
~~predicted, estimated, correction, final~~
~~to km~~
~~5.55 2.71 2.01 2.01~~
~~1514.5 1514.5 1514.5 1514.5~~

y [km]




ISS using 20 observations of range rate, azimuth and elevation

Summary:

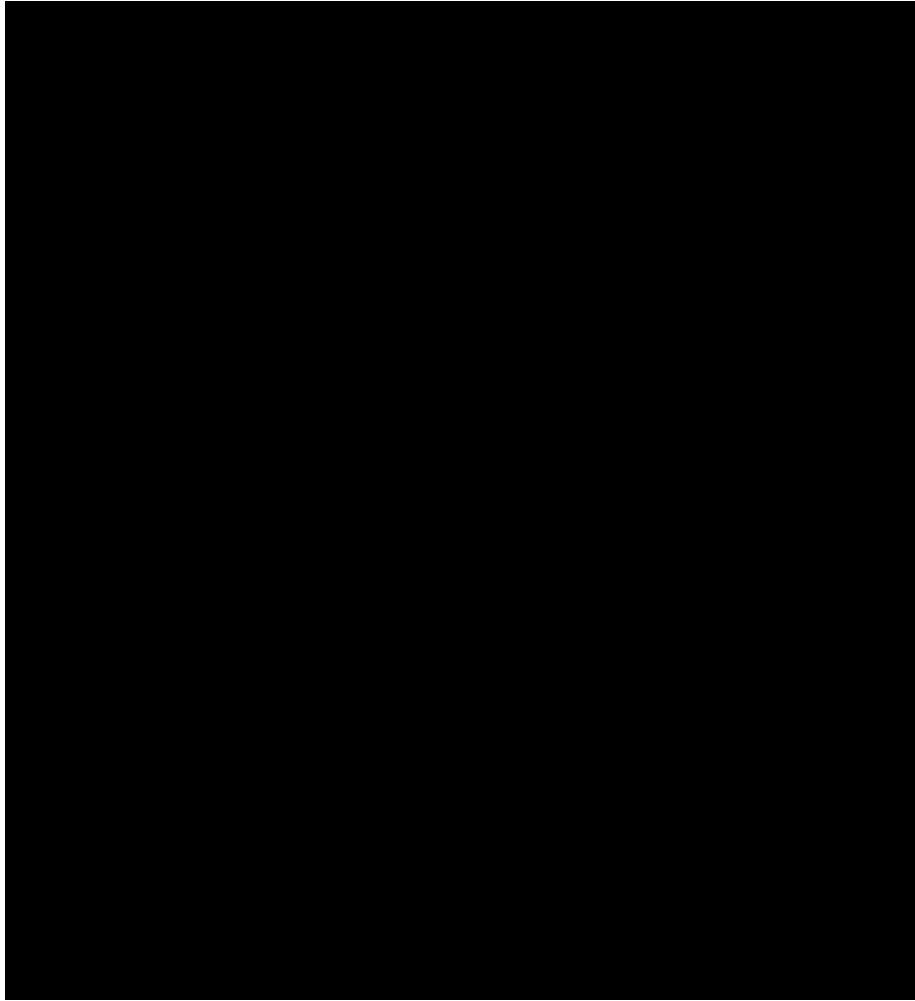


ISS using 40 observations using range rate only

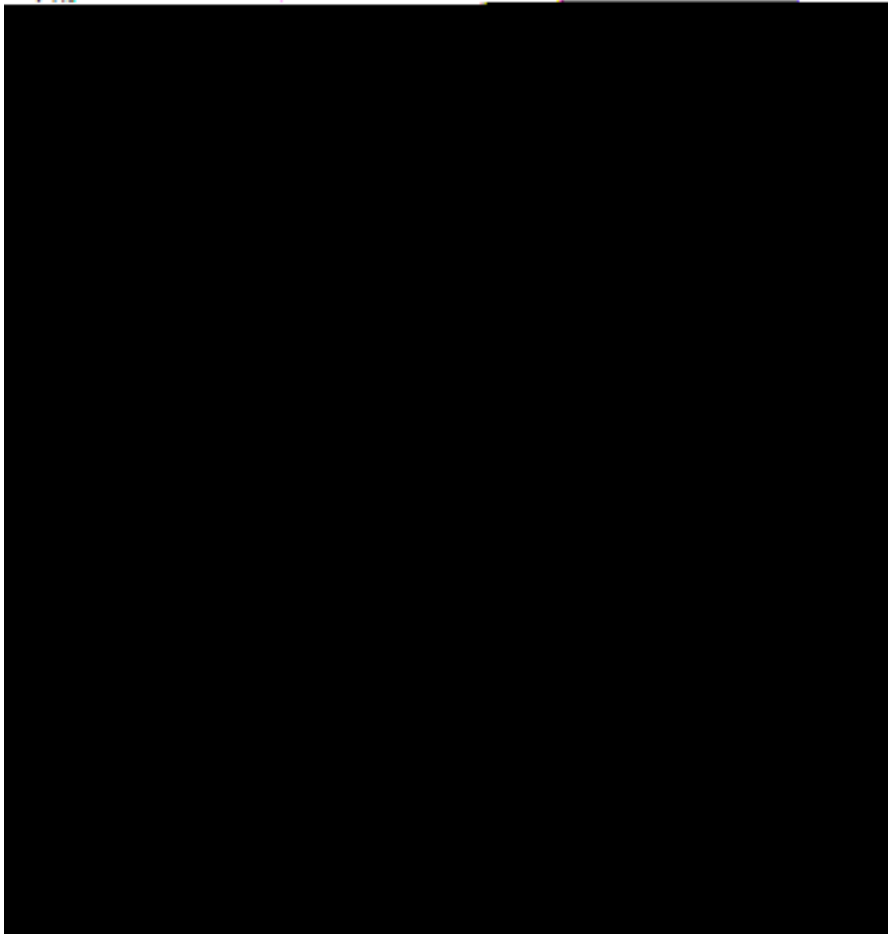
Summary:

predicted	estimated	correction	final
			

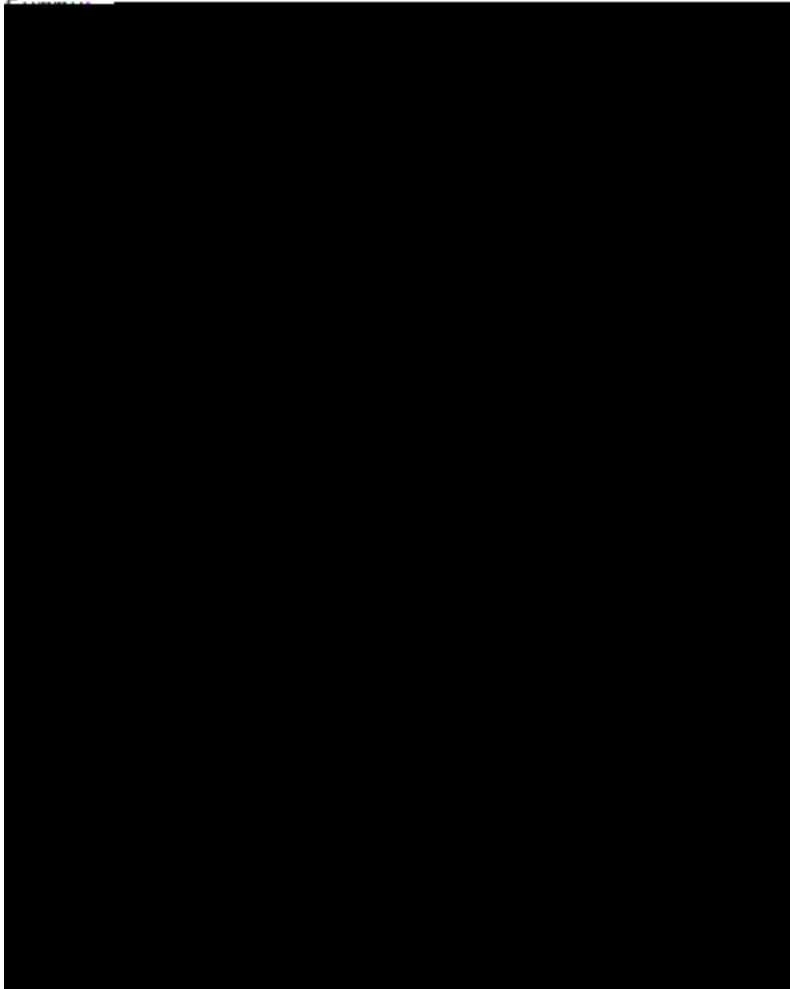
ISS using 40 observations of range rate, azimuth and elevation



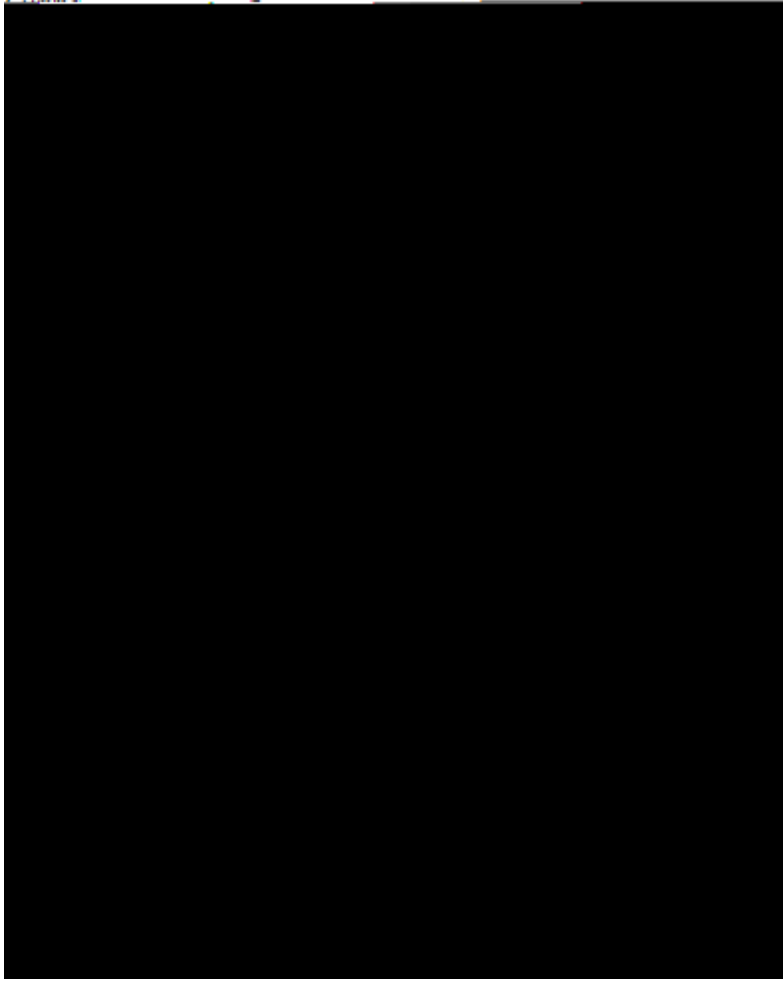
NAVSTAR-77 using 10 observations of range rate, azimuth and elevation



NAVSTAR-77 using 20 observations of range rate only

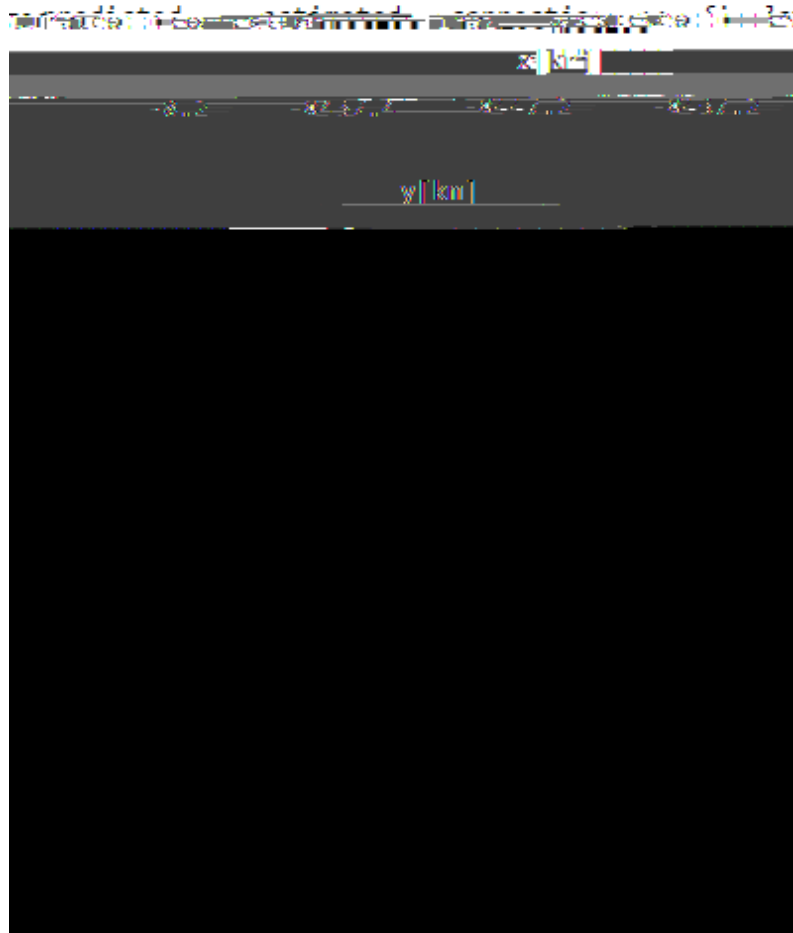


NAVSTAR-77 using 40 observations of range rate only



MOLNIYA 3-50 using 10 observations of range rate

Summary:



MOLNIYA -3-50 using 10 observation of range rate, azimuth and elevation

Summary:

	predicted	estimated	correction	final
x[km]	-8117.2	-8127.2	10.0	-8120.2
y[km]	1111.2	1111.2	0.0	1111.2
z[km]	2531.2	2531.2	0.0	2531.2
vx[m/s]	0.0	0.0	0.0	0.0
vy[m/s]	0.0	0.0	0.0	0.0
vz[m/s]	0.0	0.0	0.0	0.0
ax[m/s ²]	0.0	0.0	0.0	0.0
ay[m/s ²]	0.0	0.0	0.0	0.0
az[m/s ²]	0.0	0.0	0.0	0.0

MOLNIYA -3-50 using 20 observation of range rate

Summary:

predicted	estimated	correction	final
x [km]			
5251.8	5226.3	5221.3	30.5
y [km]			
5251.8	5226.3	5221.3	30.5
z [km]			
-2521.7	-2522.6	-2521.6	-0.1
vxv [km/s]			
6.1067	6.1013	6.1056	0.0011
m/s			
0.0357	0.0387	0.0019	0.0407
vyv [k			
m/s			
4.7314	4.7309	-0.0029	4.7280
vzv [k			
3.89410			

MOLNIYA -3-50 using 20 observation of range rate, azimuth and elevation

Summary:

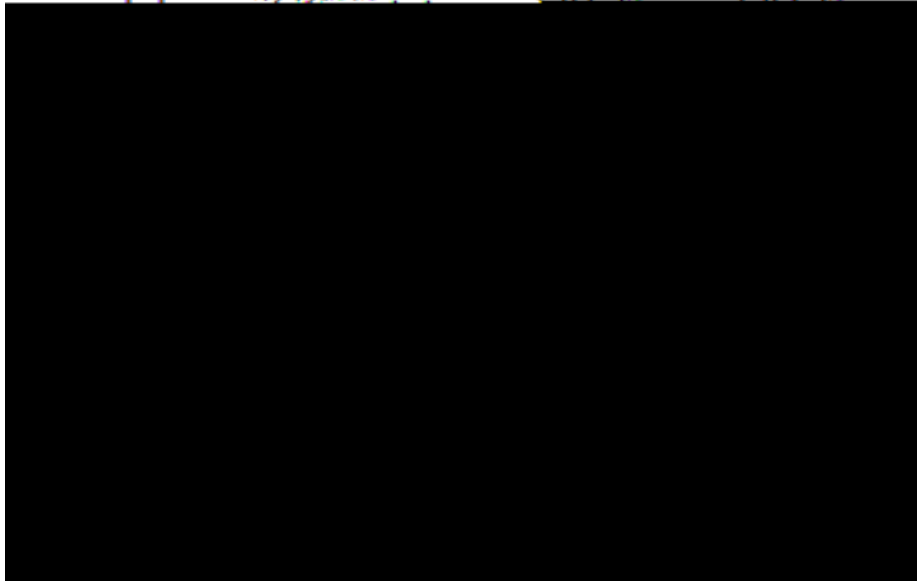
predicted	estimated	correction	final
x[km]			
-8447.2	-8437.2	-2.1	-8439.2

y[km]

5226.7	5224.7	2.0	5210.6
-------------------	-------------------	----------------	-------------------

z[km]

1.1	-2529.4	-2522.6	-2521.6
-----	---------	---------	---------



MOLNIYA -3-50 using 40 observation of range rate, azimuth and elevation

Summary:

	predicted	estimated	correction	final	
x [km]					
y [km]	5234.8	5226.3	5221.3	13.5	
z [km]					
x-dot [km/s]					
y-dot [km/s]	0.0357	0.0387	0.0026	0.0413	RMSE
z-dot [km/s]	4.7314	4.7309	-0.0018	4.7291	RMSE
Root Mean Square:					RMSE
RMSE =	0.015478				

```
= 0.005*pi /180;
```

```
% Noise standard deviation [Az
```

```
= 5;
```

```
% Range rate standard deviation
```



```

Jacobi an)
options.lambda = 0.0001; % Starting point for Marq.
parameter
options.incr = 10; % Factor for increasing lambda
options.decr = 0.4; % Factor for decreasing lambda
options.maxiter = 29; % Maximum amount of iterations
options.eps1 = 1e-4; % Gradient convergence Criteria
options.eps2 = 1e-8; % Parameter convergence criteria
options.eps3 = 1e-6; % RMS criterion
options.eps4 = 1e-20; % Acceptance criteria
if choice == 1 || choice == 3
    options.wts = 1/sig1^2;
elseif choice == 2
    options.wts = [];
    for i = 1:n_obs
        options.wts = [options.wts; 1/sig1^2; 1/sig1^2; 1/sig2^2];
    end
end
end

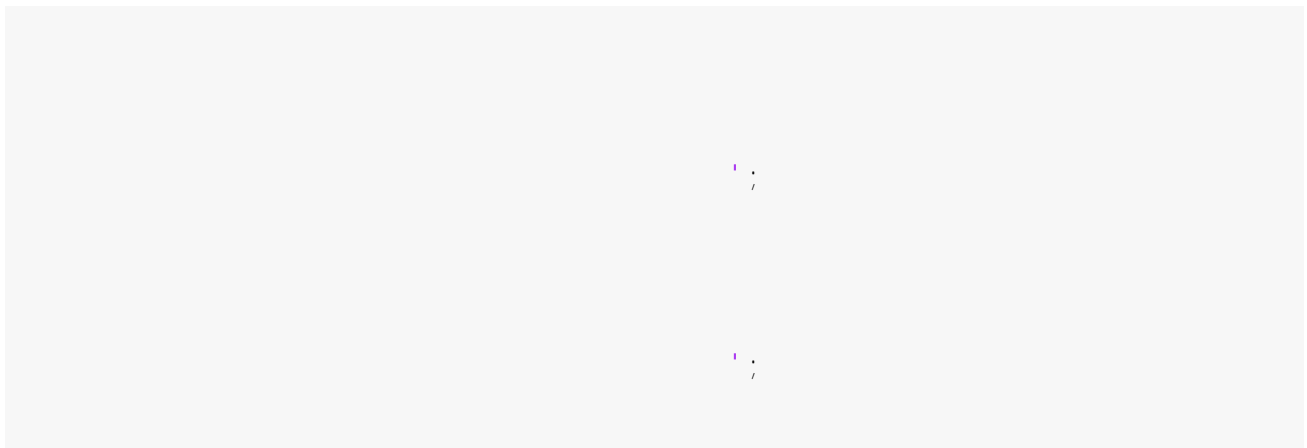
```

3) LOAD SATELLITE DATA AND CALCULATE INITIAL ESTIMATE STATE VECTOR

There are currently three satellites to choose from. Should the user desire studying an alternative satellite, enter the name of the satellite as done below. Note that each name has 24 characters including spaces. Consult [Celestrak](#) for lists of satellites and TLEs.

From here, it is possible to calculate the position of the satellite using Two-Line Element (TLE) data. TLE data from all active satellites is read from [Celestrak](#) and stored in the file TLE_DATA.txt. This text file is scanned for the relevant satellite, then the TLE of that satellite is stored in the file new_tle.txt. The state vector is calculated using sgp4.m.

In the TLE, epoch is represented in days (with fraction of day) since Jan. 1 of the current year. To get time since epoch for an event, first days since Jan. 1 of the event is calculated, then epoch is subtracted from that value. The function sgp4.m is used to extract the state vector from the TLE at the event time.



```
sat.minute1 = passTime1(5);  
sat.second1 = passTime1(6);  
  
sat.UT = sat.hour1 + sat.minute1./60 + sat.second1./(60*60);
```

4) SIMULATE GROUND STATION

Constructing the position of the ground site can be done using Eqn. (5.86) in the Curtis book. For the purposes of this experiment, the latitude, longitude, and altitude of San Jose State University are used for this construction.

5) READ IN EARTH GRAVITY FIELD COEFFICIENTS AND MODEL PARAMETERS


```
end
for i = 1:length(t)
    Obs(i, :) = get_obs(Y0_ref, t(i), sat, 1); % Evaluate model using Y0
end
```

$$\dot{\rho} = \frac{\vec{\rho} \cdot \dot{\vec{\rho}}}{\rho}$$

$\vec{\rho}$

$$\frac{\dot{\rho}}{\rho}$$

f_i

f_r

which is plugged into the $\dot{\rho}$ equation:

This equation can now be solved for

Orbit determination:

The **Collector** function is used to extract the Jacobian **J**, vector of residuals **f**, and the sum of squares of


```
        (Y0(i)-Y0_apr(i)), Y0(i)/1e3);  
    fprintf('\n');  
end  
toc
```


LEVENBERG-MARQUARDT FILTER

