Geosynchronous Satellite Maneuver Detection and Orbit Recovery Using Ground Based Optical Tracking
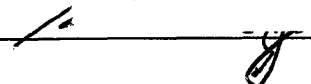
by

Benjamin S. Aaron

B.S. Aerospace Engineering
United States Naval Academy, 2004

SUBMITTED TO THE DEPARTMENT OF AEROSPACE ENGINEERING IN
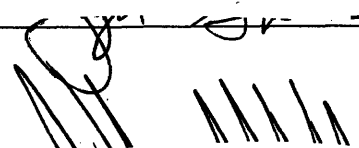PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN AEROSPACE ENGINEERING
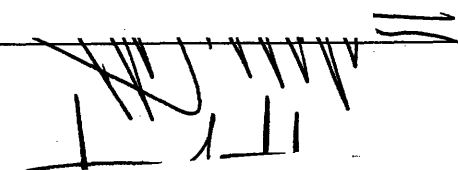AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JUNE 2006
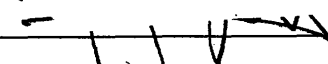
Signature of Author:_____
Department of Aerospace Engineering
May 26, 2006

Certified by:_____
Dr. Richard I. Abbot
Technical Staff, the MIT Lincoln Laboratory
Thesis Supervisor

Certified by:_____
Dr. Jayant Sharma
Technical Staff, the MIT Lincoln Laboratory
Thesis Supervisor

Certified by:_____
Dr. Jeffrey A. Hoffman
Professor of Aerospace Engineering
Thesis Supervisor

Accepted by:_____
Jaime Peraire
Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students
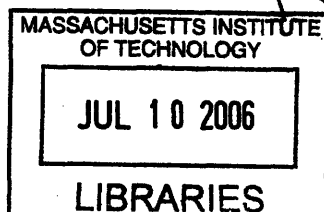
[This page intentionally left blank.]

Geosynchronous Satellite Maneuver Detection and Orbit Recovery Using Ground Based Optical Tracking

by

Benjamin S. Aaron

Submitted to the Department of Aerospace Engineering
on May 26, 2006 in partial fulfillment of the
requirements for the degree of Master of Science in
Aerospace Engineering

ABSTRACT

Geosynchronous satellite orbit maintenance is a very important issue. Satellites maneuver frequently requiring the ability to detect unknown maneuvers for target satellites and quickly recover an accurate orbit. This study uses angles only ground based optical tracking to detect maneuvers and recover orbits for geosynchronous satellites. Using the Analytical Graphics Inc. Orbit Determination Tool Kit sequential estimation software, a sequential estimation filter was "tuned" and validated in various ways. Then, a parametric study of maneuver size and time required to detect a maneuver was done via simulation. Simulated maneuvers ranging from Δv's of 0.01 m/s to 1.0 m/s are discussed. Also examined are multiple methods to recover the orbit after such maneuvers are detected. Orbits are recovered for simulated maneuvers and for a real data case of unknown maneuver size. This work is important towards developing more automatic methods of detecting maneuvers for a large population of active geosynchronous satellites.

Specific contributions made by this thesis include the following: the process and results of "tuning" of the sequential filter for a geosynchronous satellite using high accuracy ground based optical tracking data, the methods of orbit and covariance validation including an orbit overlap analysis and a statistical method using measurements, a parametric study for maneuver detection, and exploring methods for recovering post maneuver orbits quickly and accurately. With the tuned filter and optical tracking, simulations showed that a Δv of 1.0 m/s could be detected as soon as 15 minutes after the maneuver, a Δv of 0.1 m/s could be discernible within 6 hours and easily detected by 12 hours, and a Δv of 0.01 m/s took from 12 to 24 hours to detect with confidence. The best demonstrated means for post-maneuver orbit recovery utilized a method of estimating the approximate maneuver time using the pre-maneuver filter orbit and an approximate post maneuver orbit followed by the "sprinkling" of a number of maneuvers over that period with Δv's of 0 m/s but with finite covariances, and then filtering through that period.

Thesis Supervisor: Dr. Richard I. Abbot
Title: Technical Staff, the MIT Lincoln Laboratory

Thesis Supervisor: Dr. Jayant Sharma
Title: Technical Staff, the MIT Lincoln Laboratory

Thesis Supervisor: Dr. Jeffrey A. Hoffman
Title: Professor of Aerospace Engineering

Disclaimer

[This page intentionally left blank.]

Acknowledgements

First and foremost I would like to express my gratitude to Dr. Richard Abbot for his expertise, time and care in his role as my primary thesis advisor. Your guidance, knowledge and patience certainly made this all possible. I hope you are able to continue this work as you move on in the future.

I would like to thank Dr. Jeffrey Hoffman for taking the time to read and approve this thesis for MIT.

I would also like to thank Dr. Jayant Sharma for his role in advising me for this thesis. His help in framing the problem at hand was instrumental for this work.

I would like to acknowledge Dr. Richard Hujsak of AGI for his correspondence and guidance to help solve the different questions I had concerning STK and ODTK. His previous work and knowledge made the use of the sequential filter for this problem, not only feasible, but essential for this work.

Next, I would like to thank Dr. Jefferson Kommers for providing his expertise and insight into much of the analysis for tuning the sequential filter and evaluating the success of this endeavor.

I would also like to thank Zach Folcik for providing the TDRS data used in the thesis, Zeke Kaufman for spending time teaching me how to use ODTK, and Alana Sette for the many different technical tasks she assisted me with over the past year.

I would like to thank the Group 93 senior staff, primarily Dr. Dave Chan, Dr. Forrest Hunsberger, and Dr. Curt von Braun, for giving me the opportunity to not only earn my masters degree, but to hopefully contribute something that can help in your efforts toward better space situational awareness.

I would like to thank the Lincoln Laboratory for funding my research and giving me the chance to earn my masters degree.

I would also like to thank Bonnie Tuohy for assisting me with any administrative tasks I have encountered over the past two years.

Finally I would like to thank my family and my bride to be for their support.

[This page intentionally left blank.]

Table of Contents

[This page intentionally left blank.]

List of Figures

[This page intentionally left blank.]

List of Tables

[This page intentionally left blank.]

1. Introduction and Motivation

Current space assets are vital to the way of life of hundreds of millions of people. Communications, research, surveillance and reconnaissance are the primary uses of space today. Protecting satellites from the natural environment of space is a difficult job on its own. Satellites are in constant contact with dangers ranging from solar flares to close crossings with nearby satellites. From a commercial standpoint, an important aspect of satellite protection is focusing on close orbit crossings (or conjunctions) primarily in the GEO belt. Satellites in the GEO belt are constantly correcting orbits to maintain their relative position to the earth. However, some satellites maneuver for other reasons. Other growing space powers are exploring the possibilities of microsatellites that could be maneuvered to possibly collide or interfere with important and expensive space assets. The Rumsfeld Commission report from January 2001 [Ref 1] warns of a possible "space Pearl Harbor" that could be a series of catastrophic events that potentially could occur in the near future. The possibilities are endless; however, it is important to look at one aspect of these events. It is vital to be able to detect satellite maneuvers quickly and recover orbits as rapidly as possible in order to prevent any of the above mentioned scenarios.

This thesis primarily focuses on maneuver detection and post maneuver orbit recovery. In order to maintain orbits for satellites that are being tracked, one has to be able to detect maneuvers and quickly and accurately recover the subsequent orbit. If the only information known about a satellite is the tracking data that are acquired, then maneuver detection can be a difficult task. For this purpose, the Analytical Graphics Inc. (AGI) Orbit Determination Tool Kit (ODTK) is used. ODTK uses a sequential filter that enables the user to update a satellite orbit as each new track of data is encountered by the filter. This feature allows for a "real-time" approach that shows how an orbit and its associated covariance (orbit error) changes over the orbit fit span. As new data is introduced to the filter, the filter determines whether the data are good enough to update the orbit fit. This helps with maneuver detection because data that follows a maneuver will be rejected by the filter if the maneuver is large enough to detect. As more and more

data is summarily rejected by the filter, the operator will know that the target satellite has maneuvered. At this point, the operator must be able to quickly recover the orbit so that they may continue tracking the target satellite. This process is very useful for those interested in both commercial and tactical operations. In commercial operations, the user is primarily concerned with ensuring that satellites do not accidentally collide, especially in the overcrowded GEO belt. For tactical purposes, the user wants to avoid intentional rendezvous with hostile satellites that could "disrupt, disable or destroy the target" satellite [Ref 1].

This thesis will explore the use of high accuracy ground based optical tracking data from a single ground station together with a high accuracy sequential estimator for orbit determination. A single tracking station was used to not only simplify the analysis, but also to demonstrate its capability. For this study, first the sequential estimator will be "tuned" in order to provide the best possible orbit and covariance. Further methods will then be presented to confirm the orbit accuracy and covariance. This will be followed by a parametric study used to determine if a maneuver has occurred as a function of maneuver size, time and amount of tracking after the maneuver, and how well the orbit was maintained prior to the maneuver. Following this, three methods will be presented to see how quickly and accurately each can recover the post maneuver orbit and return to pre-maneuver accuracy. Two of these methods will use filter information of the orbit before the maneuver. This will also be done in a parametric manner as a function of timeliness, density and distribution of the post maneuver tracking data. This work will primarily concentrate on methodology with examples. The summary section will discuss more in depth study and future work.

2. Background on Sequential Estimation and AGI ODTK

For this work, the AGI ODTK is used. The purpose of this section is to provide some background information on sequential estimation and the ODTK as is relevant. The mathematics of sequential estimation is given by Vallado [Ref 2] or Tapley (et. al) [Ref 3]. The mathematics as developed and applied to the ODTK is described by Wright [Ref 4, 5].

2.1. Background on Sequential Estimation

A sequential estimator is a "technique for computing the best estimate of the state of a time-varying process" [Ref 2]. All that is needed to start a sequential estimator is an initial estimate of the state vector, the covariance matrix, the process noise, and the measurement error. The sequential estimator updates the epoch, the state vector and covariance at each measurement time (or observation). The constant updating of the filter allows for real time orbit state and covariance updates. Measurements are weighted so that newer measurements are more relevant than older measurements.

The differences between a sequential estimator (such as a Kalman Filter) and batch methods (such as least squares) are important for the problem at hand. A sequential estimator allows for real time orbit updates, while least squares must be recalculated each time there is a new measurement. A sequential estimator also will estimate the state and covariance matrix in real time; once again, least squares does not. Another major difference is that a sequential estimator can provide a realistic covariance estimate with dynamic model errors while least squares cannot. A sequential estimator can also estimate some force model coefficients that are time varying (drag and solar radiation pressure). Least squares cannot do so as quickly or easily as a sequential estimator.

Some disadvantages of the sequential estimator approach include the fact that the filter needs an initial estimate of the state, covariance matrix, process noise, the second moment of the process noise (also called the power-density matrix which is a "covariance of acceleration errors induced by the mathematical modeling of the system dynamics"

[Ref 2]) and measurement error. A least squares solution does not require so many parameters to start. Also, an issue can arise as to when a sequential estimator has enough data to cause the covariance to converge. As the covariance converges, new data may be ignored by the filter. This would of course not allow the orbit to be updated. For this reason, the process noise model is extremely important.

2.2. Covariance

The ODTK sequential estimation process is given an initial covariance for the orbit state. This initial covariance is given in terms of errors in the radial, intrack (along the velocity vector), and crosstrack (perpendicular to the orbit plane) components and their corresponding rates (see figure 2.1). The covariance is determined by including error due to the uncertainty associated with the measurements, the dynamical models, and any other factor that could affect the orbit.
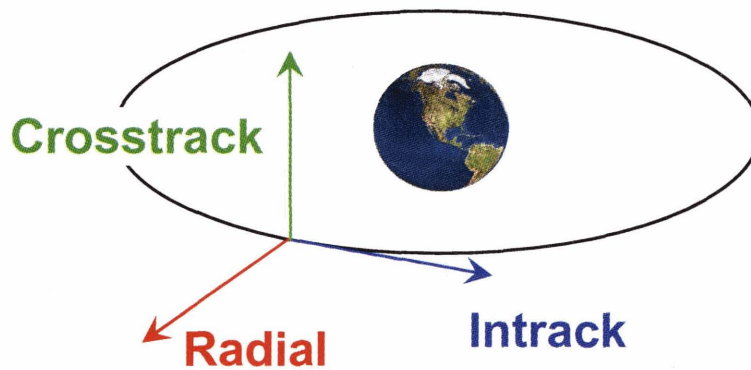
Figure 2.1. RIC coordinate plane.

2.3. Process Noise

The orbit covariance also includes what is known as state or process noise. Two useful references to describe process noise are Tapley [Ref 3] and Vallado [Ref 2]. Basically, due to nonlinearities and effects of errors in the dynamical modeling, the estimate of the orbit state can diverge. This occurs because for a sufficiently large

number of observations, the elements of the covariance matrix will asymptotically approach zero. This will cause the sequential estimator algorithm to be insensitive to further observations. One way to compensate for the fact that the linearized equations and dynamical models for propagating the estimate of the state are in error, is to assume that the errors can be approximated by process noise. Process noise can be constant, piecewise constant, correlated or white noise. For the Kalman filter sequential estimation technique, for example, one refers to the Q process noise covariance matrix. The ODTK software uses a different approach where each physical model (e.g. radiation pressure, gravity field, drag, etc.) has a process noise model [Ref 4, 5].

2.4. Smoother

It is often desirable to perform what is known as a smoothing operation when using a sequential filter. In this case, one is looking for the best estimate of the state and covariance at some time, t_1, based on all the observations through some time, t_2, where $t_2 > t_1$. For this purpose, ODTK has a fixed interval sequential smoother. The smoother uses the last filter output as its first input. As the sequential filter runs forward in time, the sequential smoother runs backward in time, from the final filter output to the first filter output [Ref 3, 4].

2.5. Simulator

When one encounters a situation where no real tracking data are available, ODTK can simulate tracking data. The simulator takes into account all error sources (force model, maneuvers, etc.) that one inputs into the system. After the simulated data are created, one can easily run the filter to fit the data in a similar manner as if real data were used. This powerful tool enables an ODTK user to create many different scenarios to seek solutions to different orbit determination questions that may arise. In this study, the Monte Carlo method is referred to in that certain quantities of the model are randomly selected – specifically the measurement error of the tracking data. A full scale Monte

Carlo analysis requires a significant number of trials to be run, using a different random number sequence for each trial. As this work focuses heavily on methodology, the simulations will be performed using a single trial to illustrate an indicative solution for each problem as opposed to a full Monte Carlo statistical analysis. The method of performing the full Monte Carlo with the ODTK will be discussed more in chapter 6. The simulator will be used extensively in this work.

3. Tuning the Filter

3.1. Filter Tuning Background

The purpose of this chapter is to "tune" the filter and show that the resultant ODTK sequential estimator will produce accurate orbits and realistic orbit errors (covariances). In order for the filter to initialize, it must start with an initial orbit state and with an estimate of the a priori error associated with that orbit. The filter also requires information on all relevant error sources (primarily the force models and measurement errors) in order to obtain realistic covariances.

To tune the filter for this project, real tracking data were used for a geosynchronous satellite for which a very high accuracy reference orbit is available. The satellite used is a component of the Tracking Data and Relay Satellite System (TDRSS), specifically TDRS 5. The real tracking data are from the optical telescope tracking system Ground-Based Electro-Optical Deep Space Surveillance (GEODSS). The data used are from the Maui GEODSS site. The high accuracy reference orbit for TDRS 5 was obtained from the Goddard Space Flight Center (GSFC).

For the tuning process, two months of Maui GEODSS tracking data and GSFC ephemeredes were used. The process began using the first two weeks of each (the tracking data and the GSFC ephemeredes) in order to obtain the best resulting tuning of the filter possible. Since the filter has so many settings, it can be very difficult to find the most ideal set. Once an acceptable set of filter tuning parameters was found, these were tested on a subsequent two week filter run. Theoretically, once a filter has been set up for a specific satellite and the filter initiated, it can maintain a satellite orbit for long periods of time, in some cases even years. For this work, shorter two week runs were suitable.

3.2. TDRSS background

The TDRS 5 satellite is one of "six first generation" satellites that make up the TDRSS. The TDRSS provides communication links for low earth orbit (LEO) satellites. Since LEO satellites change their position relative to earth, in order to have real-time data transfer, the data must be relayed by a satellite that has line-of-sight of both the satellite gathering the information and the user located on the ground. TDRSS provides this capability. In order to ensure that this complicated data link can be achieved, the LEO satellite transmitting the data must know exactly where the TDRSS satellite is. Also, in order for the TDRSS satellite to relay the data received, it must be positioned and oriented properly toward the ground station (White Sands Ground Terminal) located near Las Cruces, New Mexico. Since the location of each TDRS is so important, it is vital that an extremely accurate orbit for each TDRS is maintained. The GSFC maintains a high precision orbit for each TDRS that is updated daily. This orbit is precise to 30 meters [Ref 6]. Since this orbit is so accurate, we are able to compare the results of our filter run to the orbits maintained by GSFC for TDRS 5 [Ref 7].

3.3. GEODSS background

The Ground-Based Electro-Optical Deep Space Surveillance (GEODSS) system tracks objects in deep space. GEODSS "can track objects as small as a basketball more than 20,000 miles in space." GEODSS is responsible for tracking objects like TDRS 5 that are located in the GEO belt. GEODSS ground station sites are located in Socorro, New Mexico; Maui, Hawaii; and Diego Garcia, British Indian Ocean Territories [Ref 8]. TDRS 5 is tracked by three electro-optical sites in Maui that are part of the GEODSS. The three sites are Maui A (MA), Maui B (MB) and Maui C (MC). On average, each GEODSS site at Maui generates one track of data per day for TDRS 5 that contains between 6 and 12 measurements of each right ascension and declination. This data is accurate to approximately 2 arcsecs in both right ascension and declination.

3.4. Single Optical Ground Station Observability

A question may arise as to the capability of a single optical ground station to make the necessary observations in order to provide adequate information to fit an orbit for a GEO satellite. Observability is the ability of the estimation process to uniquely determine the orbit state from the observations, or how sensitive the state parameters are to the observations. This sensitivity is expressed by the partial derivatives of the state with respect to the observations in what is known as the information matrix from the estimation process.

This thesis is concerned with geosynchronous satellites that can also be geostationary. A geosynchronous satellite is by definition, a satellite that has the same rotation period as the earth. It may have a non-zero inclination and eccentricity that would provide relative motion with respect to a ground station. A geostationary satellite by definition should be stationary, or have no relative motion, with respect to a ground station. However, a real world geostationary satellite orbit, with the proper semimajor axis and zero inclination and eccentricity, is difficult to achieve. Even if one is able to place a satellite in a geostationary orbit, maintaining that orbit is nearly impossible due to the natural perturbations that affect satellite orbits. The geostationary satellite would be subject to a nonspherical gravity field causing the longitude to drift towards equilibrium points. The sun and moon would apply torques on the satellite increasing its inclination. Solar radiation pressure would also change the satellite's eccentricity.

To demonstrate observability from a single optical ground station a satellite was simulated that had an eccentricity of ($\sim10^{-4}$) and an inclination angle of 0.1°. With two observation tracks per day, using a 2 arcsec measurement error for the simulated data, the single optical ground station was able to fit an orbit within a 1 km intrack position error within 2 days. This is an expected and sufficient result.

For this study, the TDRS 5 case we are using, the inclination is sufficiently larger (i = 6.47°) while the eccentricity is on the same order ($\sim 10^{-4}$). This means that there will be sufficient relative motion with respect to the ground station; and therefore, there will be no aliasing problems with observability.

3.5. Initial Filter Tuning

The tracking data set for TDRS 5 chosen to "tune" the sequential estimator was from calendar days 175 to 190, 2005. Table 3.1 shows the relevant settings (i.e. those settings that required more specific attention) for the filter run based on a priori errors and the force models. Guidelines for the initial values for the settings were provided by Massachusetts Institute of Technology Lincoln Laboratory (MIT LL) experience of calculating orbits for geosynchronous satellites [Ref 9]. The settings were modified to their final values by examining the following plots and iteration of the tuning process. Table 3.1 focuses on the force and measurement models specific for geosynchronous satellites and for TDRS 5. The complete list of filter settings applied can be found in Appendix A. These settings will be most important for future follow on work.

3.5.1. Position and Velocity Uncertainty

The ODTK software provides a wide range of diagnostic plots. For the initial filter tuning process we will show all the relevant plots that will be presented throughout this thesis. The position and velocity uncertainty plots are the first to be examined in this section. The following plots show these from a filter run. Figures 3.1 and 3.2 show the 2-σ orbit error or (95% confidence level) covariance (position uncertainty) resulting from running the filter over the first two week period (days 175 to 190 of 2005) for TDRS 5 with all the Maui GEODSS data.

ODTK position and velocity output are displayed in RIC (radial, intrack and crosstrack) components. RIC components are derived from ECI (Earth-centered inertial) components that are subject to a transformation matrix [Ref 5].

Scenario Parameter	Setting Change	Comment/Description
Satellite>PhysicalProperties>Mass	1658.243 kg	Input mass of satellite as necessary
Satellite>OpticalProperties> ReferenceFrame	TEME of date	
Satellite>RangingMethod	Skin Track	Best setting for optical observations
Satellite>ForceModel>Gravity> DegreeAndOrder	8	
Satellite>ForceModel>Gravity> ProcessNoise>Use	Yes	Uses default Process Noise when modeling gravity
Satellite>ForceModel>Gravity> ThirdBodies>UseInVariationalEquations	true	
Satellite>ForceModel>SolarPressure> CPNominal	1.37	
Satellite>ForceModel>SolarPressure> Area	40 m^2	TDRS 5 approx. surface area
Satellite>ForceModel>SolarPressure> CPSigma	0.1	Set between 5-10 % of CPNominal
Satellite>ForceModel>SolarPressure> CPHalfLife	14400 min	Time it takes for the Gauss Markov process noise error estimate to decay to one half its value in the absence of measurements
Satellite>ForceModel>SolarPressure> ReflectionModel	Sphere with Perfect Absorption	For compatibility with GSFC ephemeris
Satellite>ForceModel>SolarPressure> AddProcessNoise	true	Uses Process Noise when modeling SRP
Satellite>OrbitUncertainty>R_sigma	250 m	Initial Radial Pos. Covariance
Satellite>OrbitUncertainty>I_sigma	2000 m	Initial Intrack Pos. Covariance
Satellite>OrbitUncertainty>C_sigma	350 m	Initial Cross. Pos. Covariance
Satellite>OrbitUncertainty>Rdot_sigma	0.5 m*sec^{-1}	Initial Radial Vel. Covariance
Satellite>OrbitUncertainty>Idot_sigma	0.05 m*sec^{-1}	Initial Intrack Vel. Covariance
Satellite>OrbitUncertainty>Cdot_sigma	0.05 m*sec^{-1}	Initial Cross. Vel. Covariance
TrackingSystem>Facility> MeasurementStatistics> RightAscension>Type>Bias	0 arcsec	No bias introduced into measurements. Same for Declination.
TrackingSystem>Facility> MeasurementStatistics> RightAscension>Type>BiasSigma	2 arcsec	Measurment accuracy. Same for Declination.
TrackingSystem>Facility> MeasurementStatistics> RightAscension>Type>WhiteNoiseSigma	2 arcsec	Measurment accuracy. Same for Declination.

Table 3.1. Relevant settings for filter force model.

27

RIC components are used in ODTK because they are more intuitive than ECI. As figure 2.1 shows, the radial component points from the center of the satellite directly away from the Earth, the intrack component points from the satellite in the direction of motion, and the crosstrack component points from the satellite at a right angle out of the plane of motion.

As is displayed in figure 3.1, the filter covariance begins at a very high uncertainty (especially in the intrack direction). As the sequential filter runs, more data are introduced that updates the orbit state and the covariance. As more acceptable data are introduced into the sequential filter run, the covariance begins to converge to a certain 2-σ value.
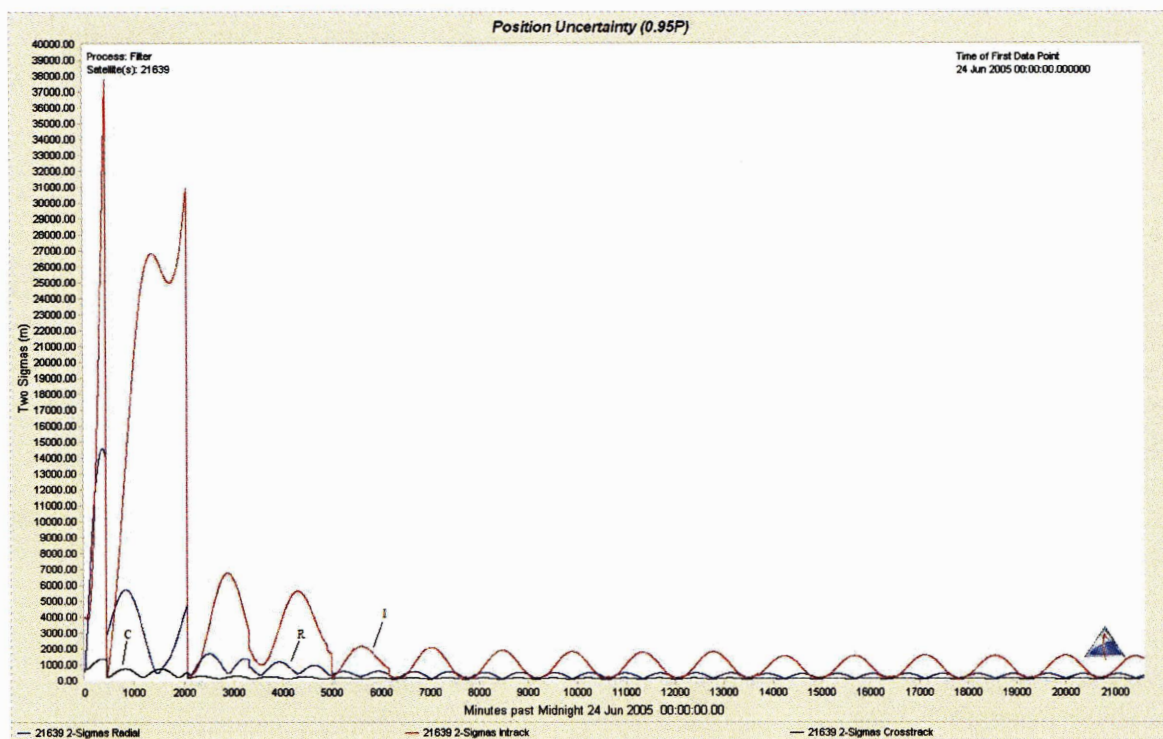
Figure 3.1. Position uncertainty for days 175 to 190, 2005.

One is able to zoom in on the final few days of the filter run in Figure 3.2. This shows that the 2-σ covariance converges to a maximum value of approximately 1600 meters in the intrack direction. With optical data, the orientation of the orbit (for RIC, the crosstrack component) is better determined than the intrack position. The radial position

28

is approximately 400 m at maximum and the crosstrack position is approximately 150 m at maximum.
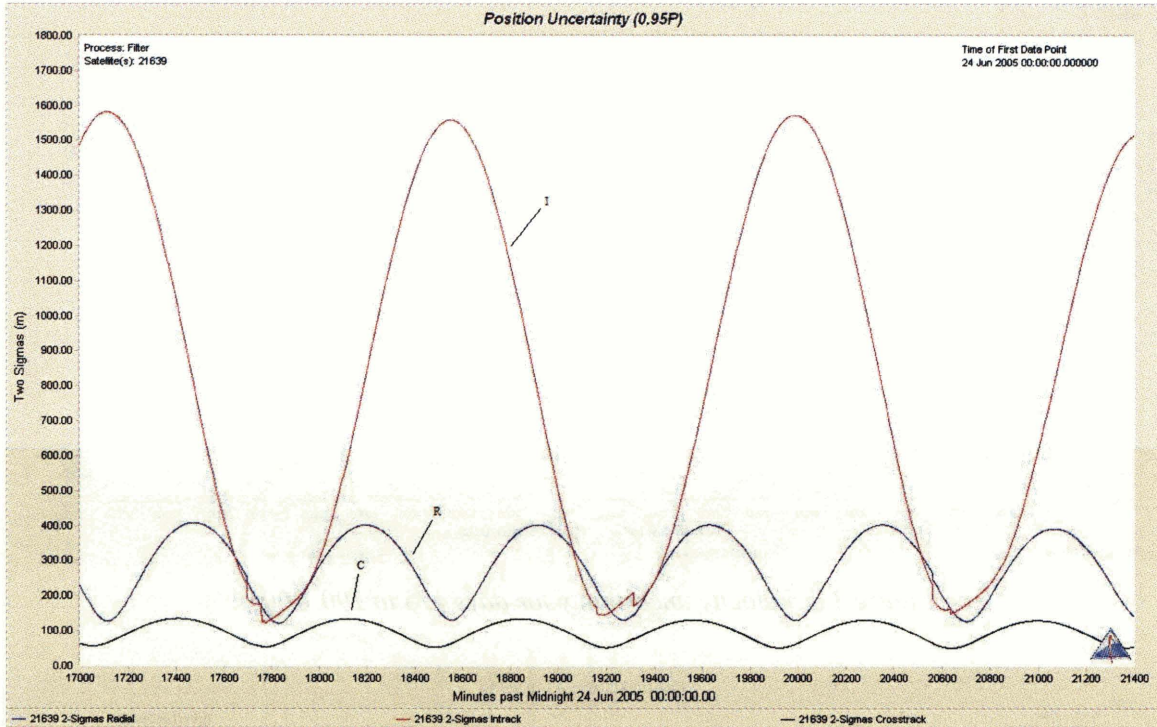
Figure 3.2. Position uncertainty zoomed in approx. for days 187 to 190, 2005.

The next two figures (figures 3.3 and 3.4) display the velocity uncertainty. Figure 3.3 shows the velocity uncertainty for the entire filter run while figure 3.4 once again shows a close-up of the last few days. As before in the position uncertainty graphs, the 2-σ covariance of the velocity uncertainty begins to converge as more data is introduced into the filter. The largest velocity uncertainty is in the radial direction (as optical data is least able to determine rate in that direction) and converges to approximately 9 cm/s (figure 3.4).
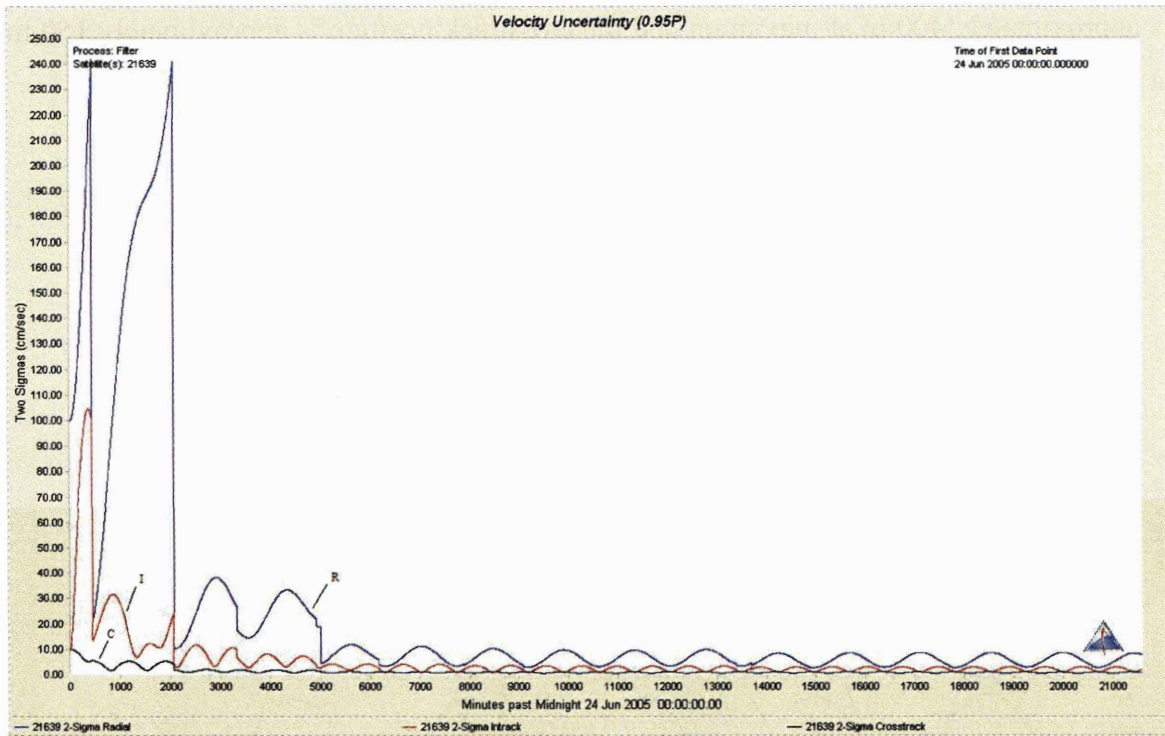
Figure 3.3. Velocity uncertainty for days 175 to 190, 2005.
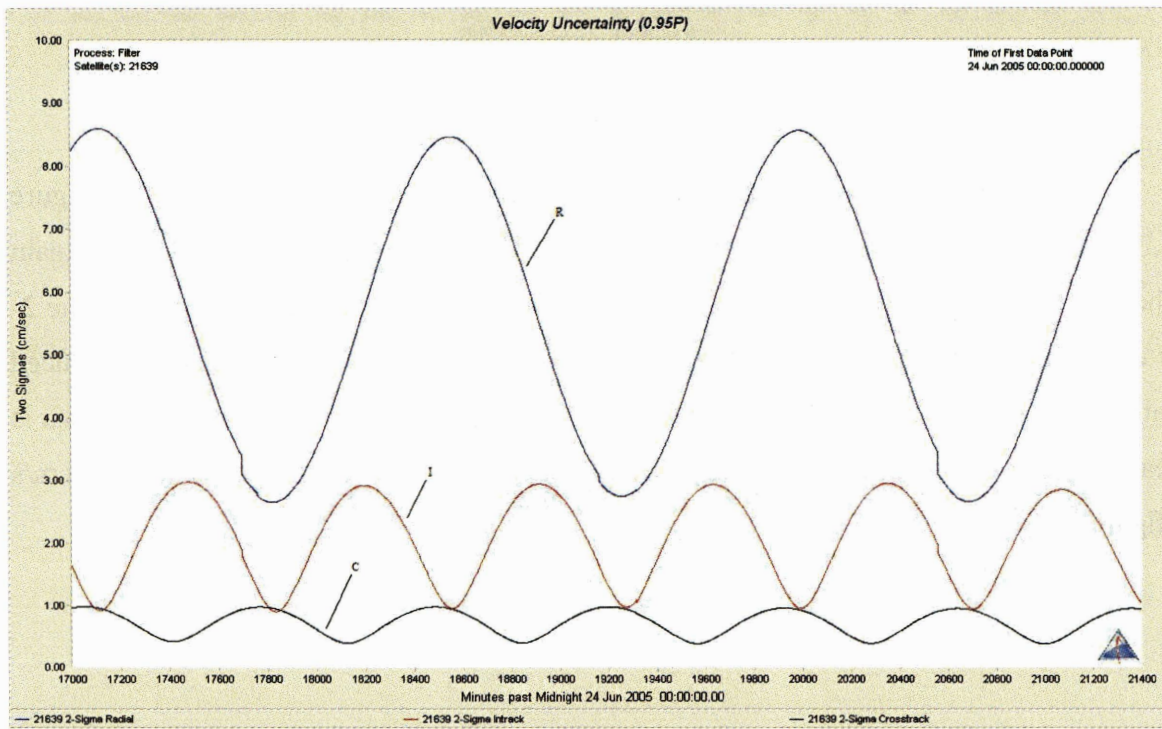
Figure 3.4. Velocity uncertainty zoomed in approx. for days 187 to 190, 2005.

3.5.2. Residual Ratios

The next plot to be shown is the residual ratios plot. A residual is the difference between the actual measurement and the measurement that would be computed from the estimated orbit. This shows how the estimated orbit matches the measurements.

Figure 3.5 displays the residual ratios graph created by ODTK after the filter run. The residual ratios are for the right ascension and declination measurements for each of the three optical tracking sites (MA, MB, and MC). The ratio is that of each residual to its associated σ which is derived by mapping the orbit covariance into the measurement space. If the individual residual ratio is greater than 3.0 or less than -3.0, then this residual will be "thrown out" by the orbit fit (the σ rejection tolerance, -3.0 to 3.0, can be changed in the settings of the filter), as ODTK considers this measurement to be too inaccurate to be used. This feature of ODTK is very helpful when one is using very inaccurate data or as will be discussed later, when one is looking to find an unknown maneuver in a data set.
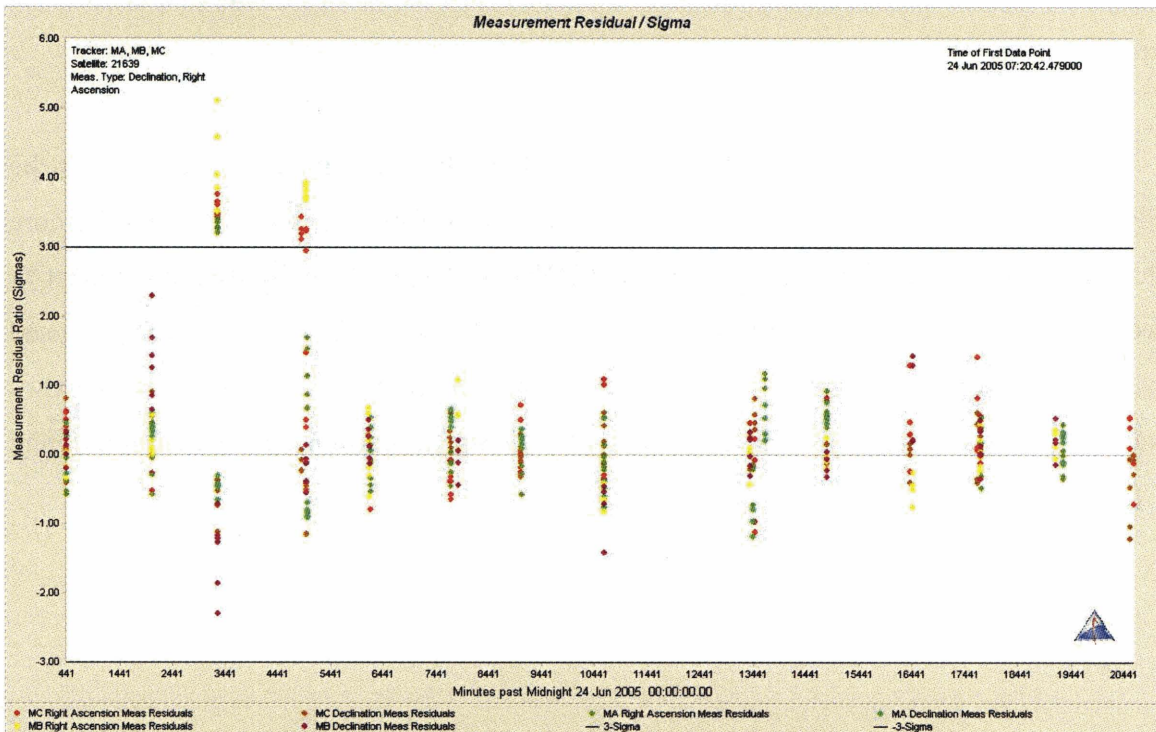
Figure 3.5. Residual ratios for days 175 to 190, 2005.

One can see from figure 3.5 that the initial residuals were large but the initial covariance permitted them to be kept. As the orbit improves with more data, the measurement residuals generally fall within their a priori errors.

3.5.3. Solar Radiation Pressure

Since Solar Radiation Pressure (SRP) is a time-varying parameter, SRP can be estimated in ODTK. SRP is modeled in ODTK as a sphere that requires an effective surface area that is oriented perpendicular to the sun, a satellite mass and an average value of the satellite reflectivity (considering both specular and diffuse reflectivity). ODTK contains a radiation pressure model scale factor (*CPNominal*, see table 3.1) that accounts for the lack of a complex and detailed satellite configuration model. The error term for the SRP model is *CPSigma* (see table 3.1). This allows a selection of *CPNominal* that can change as the SRP model is assumed to vary. As the sequential estimator encounters new data, the SRP is updated just as the orbit state and covariance are.

Figure 3.6 displays the SRP estimate for the filter run. This graph shows how the SRP parameter changes throughout the orbit run. If this parameter varies too much (more than a few percent of *CPNominal*) then the *CPNominal* and *CPSigma* used in the orbit fit were poor choices. The *CPNominal* used for this orbit was 1.37 and the *CPSigma* used was 0.1. Looking at the graph, the CP (Solar Radiation Pressure Coefficient) did not vary more than 0.05. Since this is less than the *CPSigma* chosen, the *CPNominal* and *CPSigma* seem to be adequate.

32
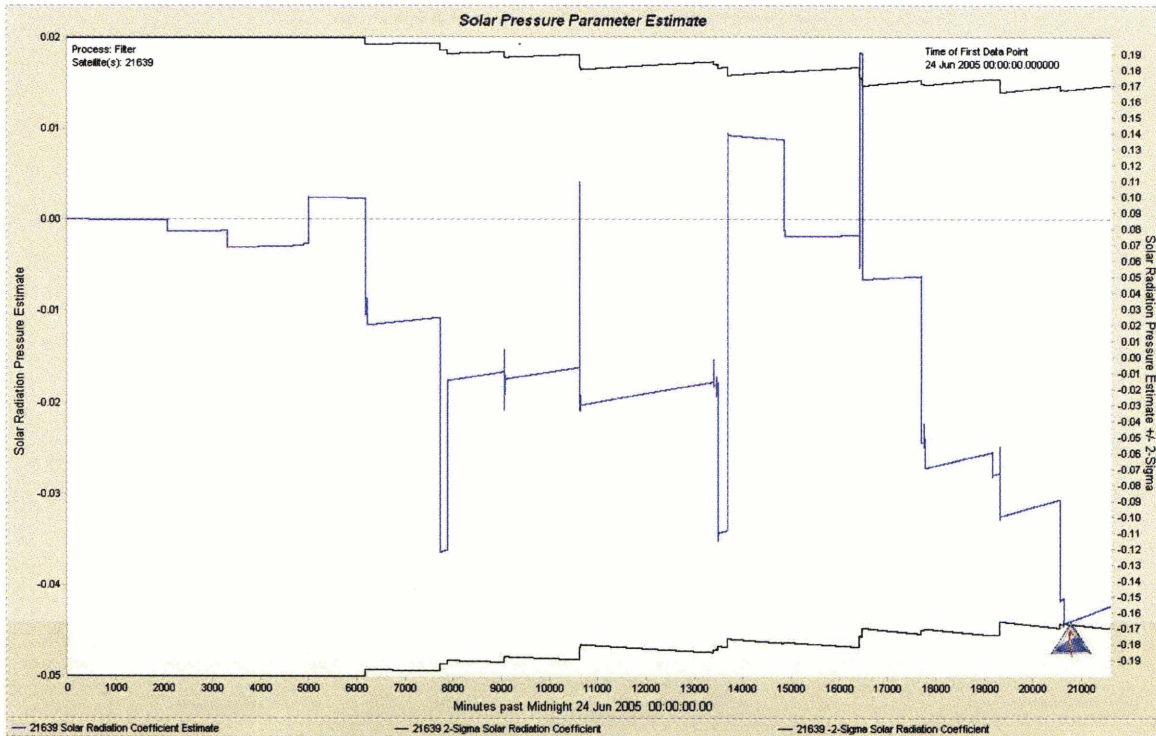
Figure 3.6. Solar radiation pressure scale factor estimate with associated covariance for days 175 to 190, 2005.

3.5.4. Filter run vs. TDRS ephemeris

The next set of graphs show, for each component, the orbit fit of TDRS 5 differenced from the TDRS ephemeris acquired from GSFC. The purpose of this is to evaluate how well the covariance matches reality. The components are again radial, intrack, crosstrack, radial velocity, intrack velocity, and crosstrack velocity. This is overlaid with the 2-σ covariance from the filter run. Only the last few days of the fit span are shown in figures 3.7-3.12. Essentially, figures 3.7-3.12 show that the orbit fit with respect to the TDRS ephemeris is within the bounds of the covariance. This helps to determine that the settings used to "tune" the filter are good enough to provide a realistic accuracy assessment of the orbit fit.

Figure 3.7. Radial position differences approx. for days 187 to 190, 2005.

Figure 3.8. Intrack position differences approx for days 187 to 190, 2005.

Figure 3.9. Crosstrack position differences approx. for days 187 to 190, 2005.

Figure 3.10. Radial velocity differences approx. for days 187 to 190, 2005.

35

Figure 3.11. Intrack velocity differences approx. for days 187 to 190, 2005.
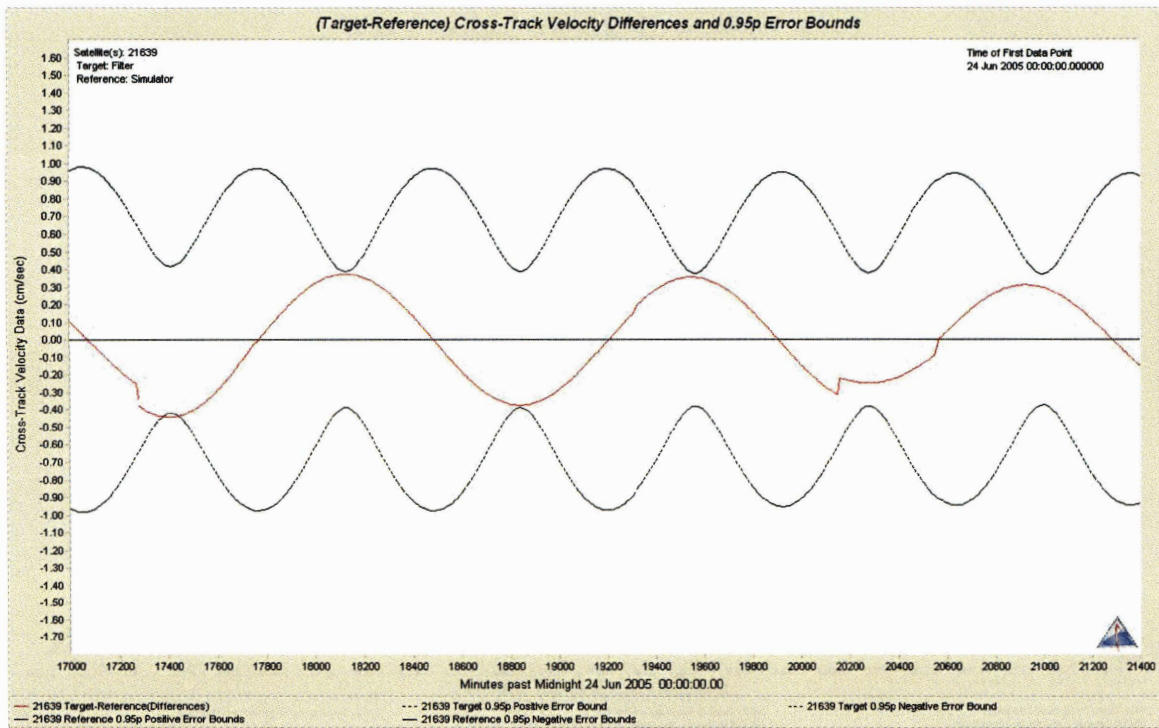
Figure 3.12. Crosstrack velocity differences approx. for days 187 to 190, 2005.

When figures 3.8 and 3.10 are examined, a bias is noticed in the along track position difference with the TDRS 5 standard and in the radial velocity difference. This is noted at the moment and addressed further in section 3.6.

From the above figures, the following is noted. For optical tracking, the crosstrack position component is determined with the best accuracy, while the intrack position component is determined the worst. Intrack errors are suspected to be largest when optical tracking is used as it is weakest in determining size and shape of an orbit. For the velocity components, the crosstrack rate is determined the best, while the radial rate is the worst. When comparing the filter run with the TDRS "truth" ephemeris, it is seen that the covariances agree well with the differences of the orbit against the "truth." Again, the a priori errors used were tweaked somewhat based on comparing with the TDRS "truth." A high negative correlation was observed between the intrack and radial rate errors. Also, the radiation pressure scale factor estimate (figure 3.6) is starting to wander towards the end of the filter period and possibly this is leading to the intrack error.

3.5.5. Smoothed Graphs

As discussed earlier in section 2.4, ODTK includes the ability to "smooth" orbit fits. Once a filter run is complete, one can smooth the fit to show the orbit fit using all the tracking data. The result is a smoothed display of the orbit covariance over the entire fit time span. Figures 3.13-3.18 show the smoothed graphs for each directional component (position and velocity). These graphs are again the differenced result of the filter run vs. the TDRS ephemeris and the 2-σ covariance from the filter run. Figures 3.13-3.18 show the entire fit span.
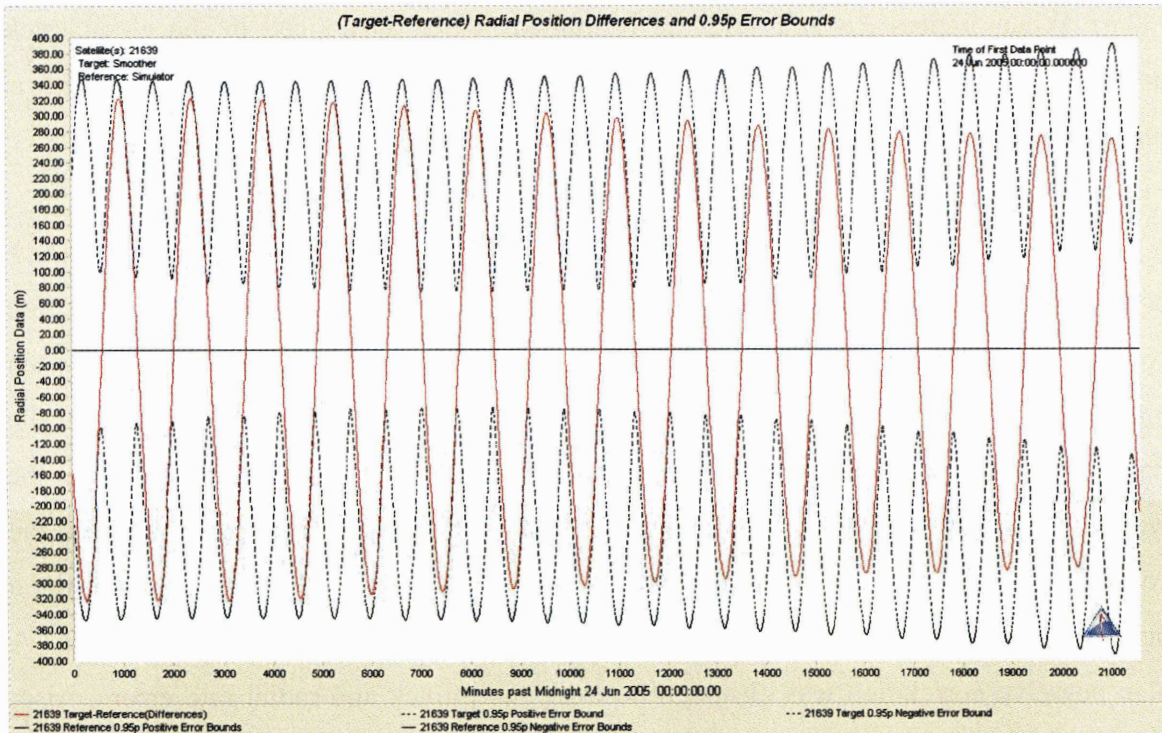
Figure 3.13. Smoothed radial position differences days 175 to 190, 2005.
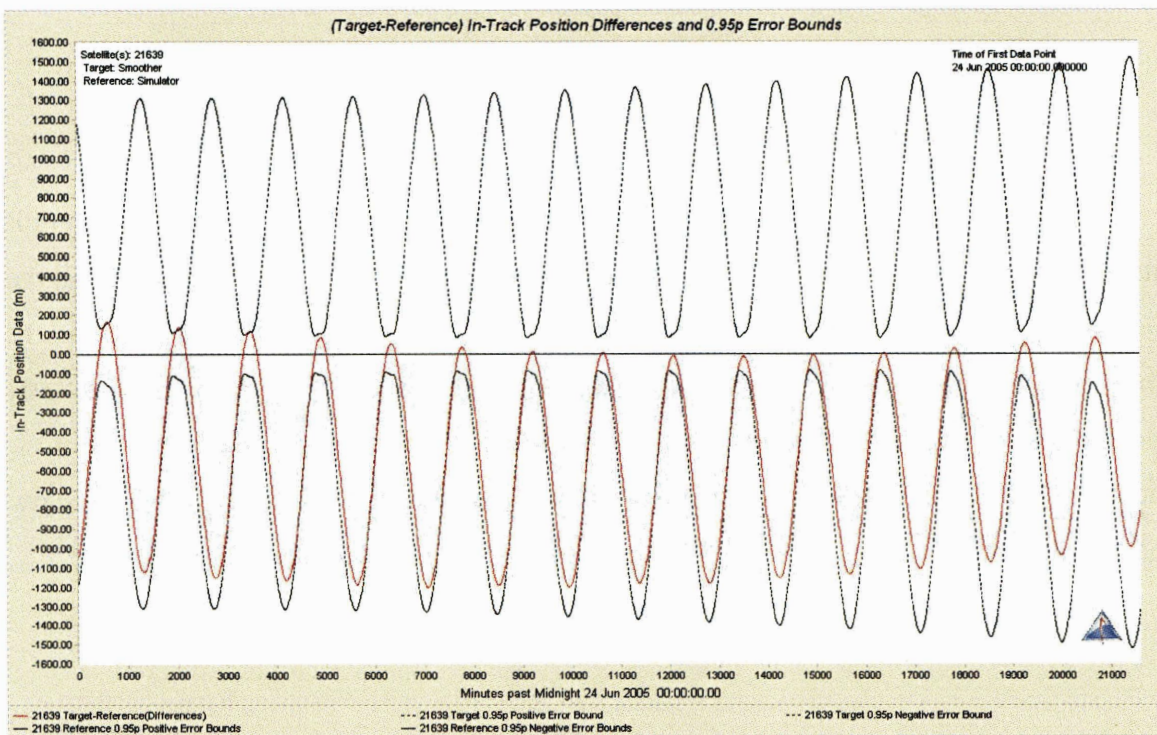
Figure 3.14. Smoothed intrack position differences days 175 to 190, 2005.

38
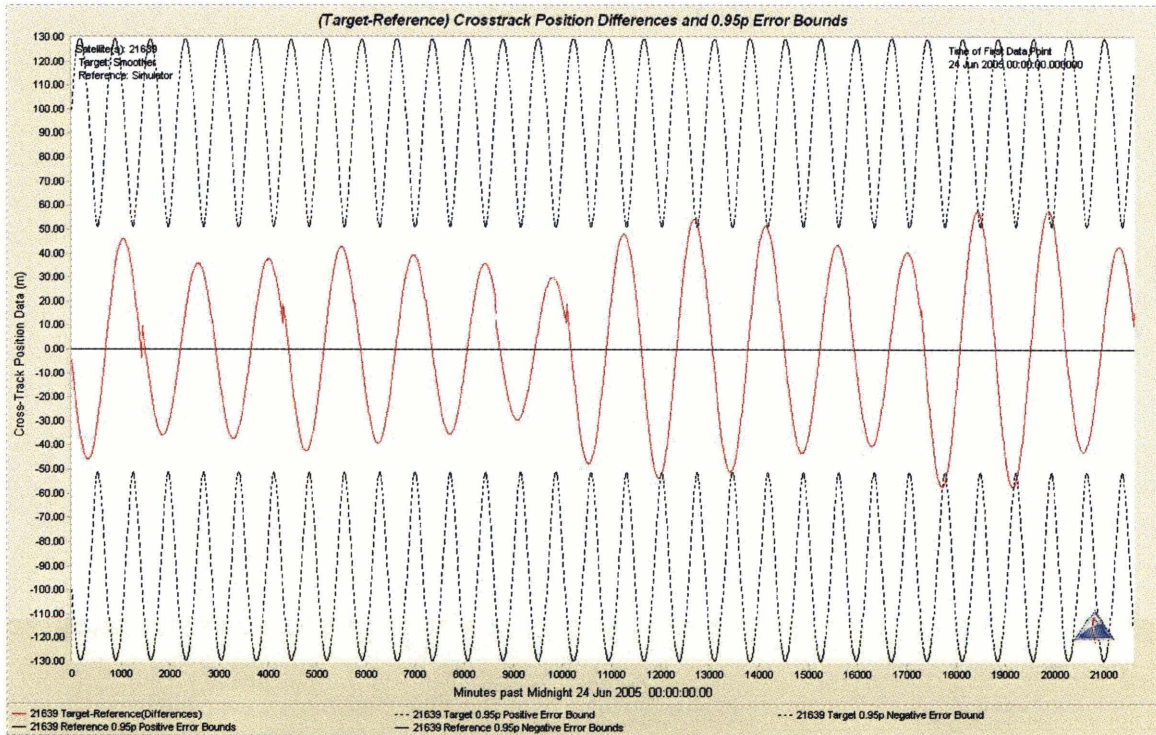
Figure 3.15. Smoothed crosstrack position differences days 175 to 190, 2005.
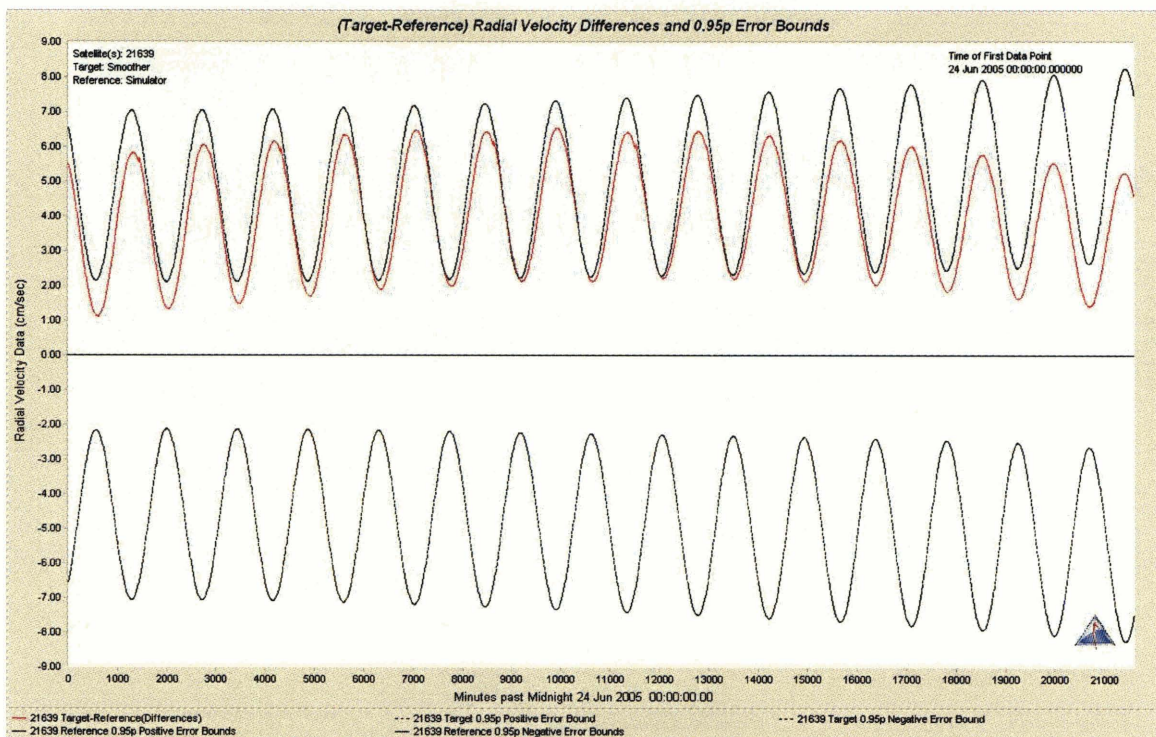
Figure 3.16. Smoothed radial velocity differences days 175 to 190, 2005.

39
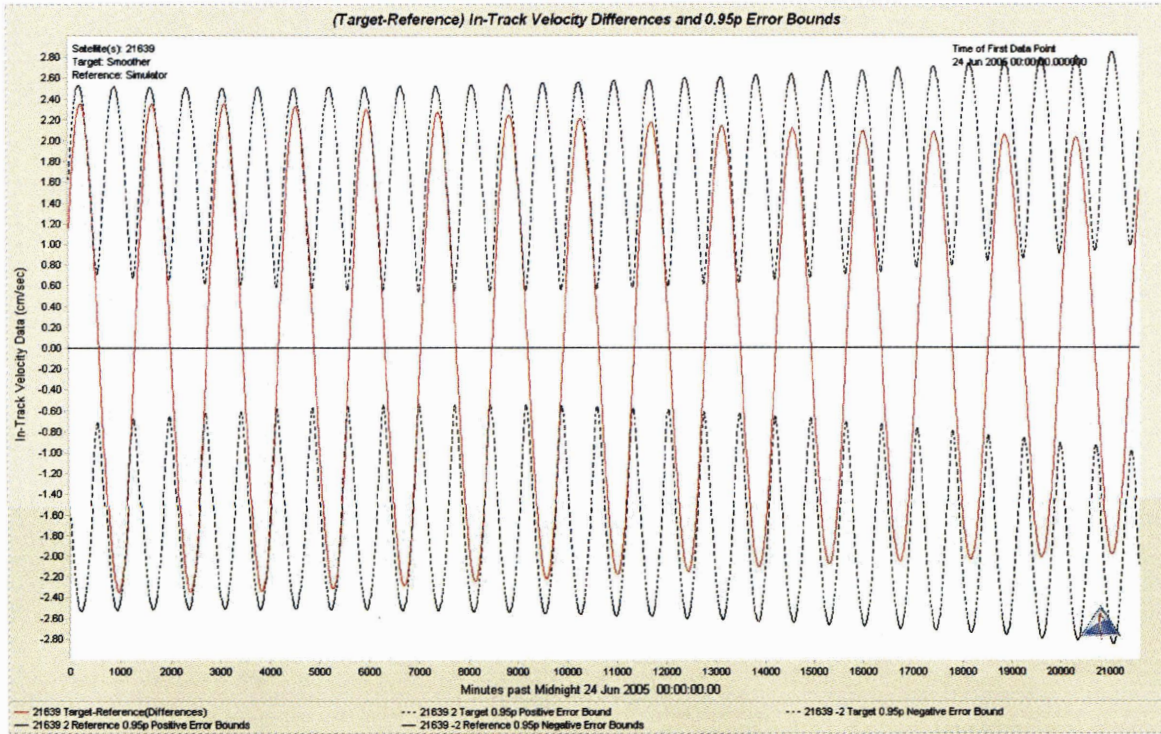
Figure 3.17. Smoothed intrack velocity differences days 175 to 190, 2005.
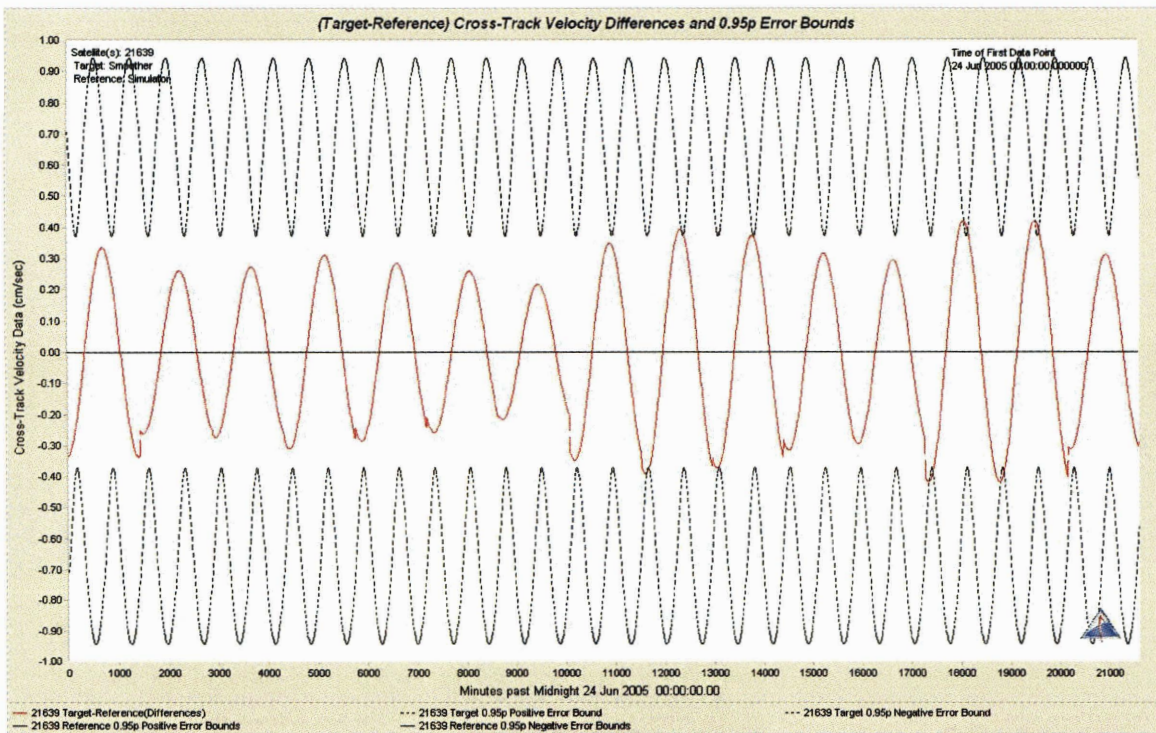
Figure 3.18. Smoothed crosstrack velocity differences days 175 to 190, 2005.

3.6. Validation of Tuned Filter

This section discusses the application of the settings for the sequential estimator to a subsequent set of Maui GEODSS tracking data. In order to gain confidence that the sequential estimator settings for the tracking data from days 175 to 190 of 2005 can be applied to future work, the settings need to be applied to another set of Maui GEODSS data. The second orbit fit was again for TDRS 5; however the fit was from days 189 to 204 of 2005. No ODTK settings were changed except for the necessary change to the time for the start and end of the filter and the initial orbit state for the satellite. Figures 3.19-3.24 are the counterparts to figures 3.7-3.12 for the second orbit fit. Once again, each of these figures shows each individual directional component (position and velocity in the RIC coordinate frame) for the filter run. Each graph shows the orbit fit of TDRS 5 differenced from the TDRS ephemeris acquired from GSFC. This again is overlaid with the 2-σ covariance from the filter run.
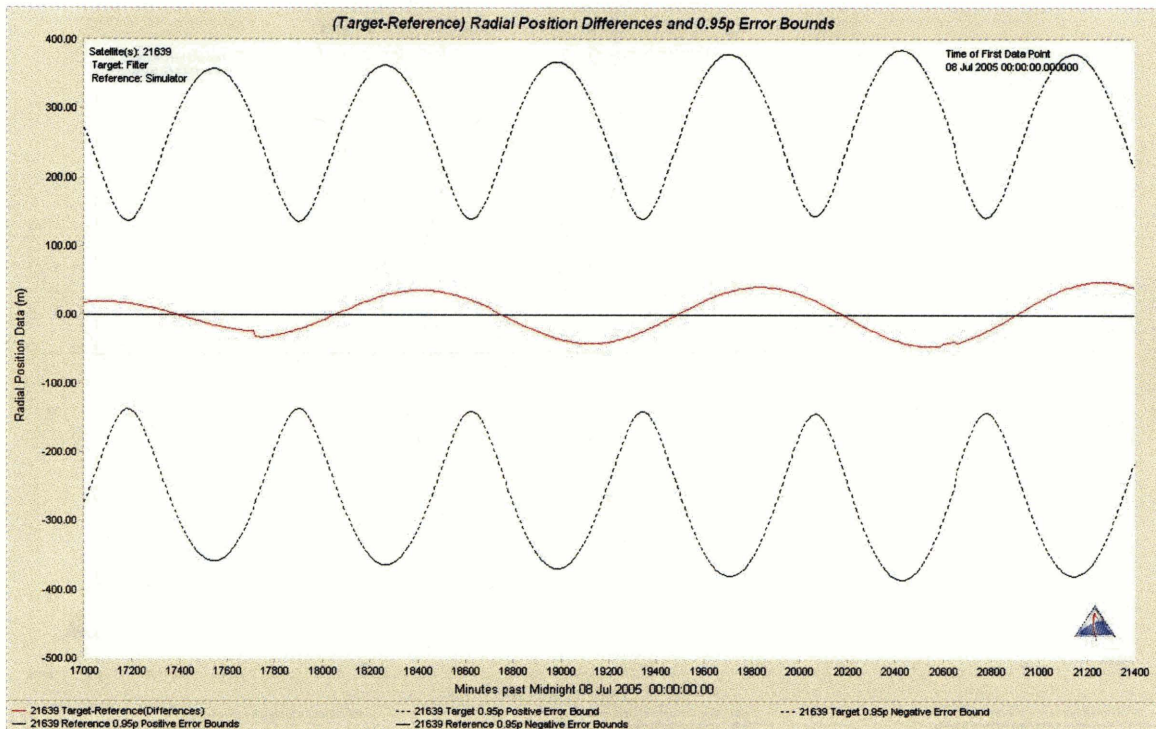
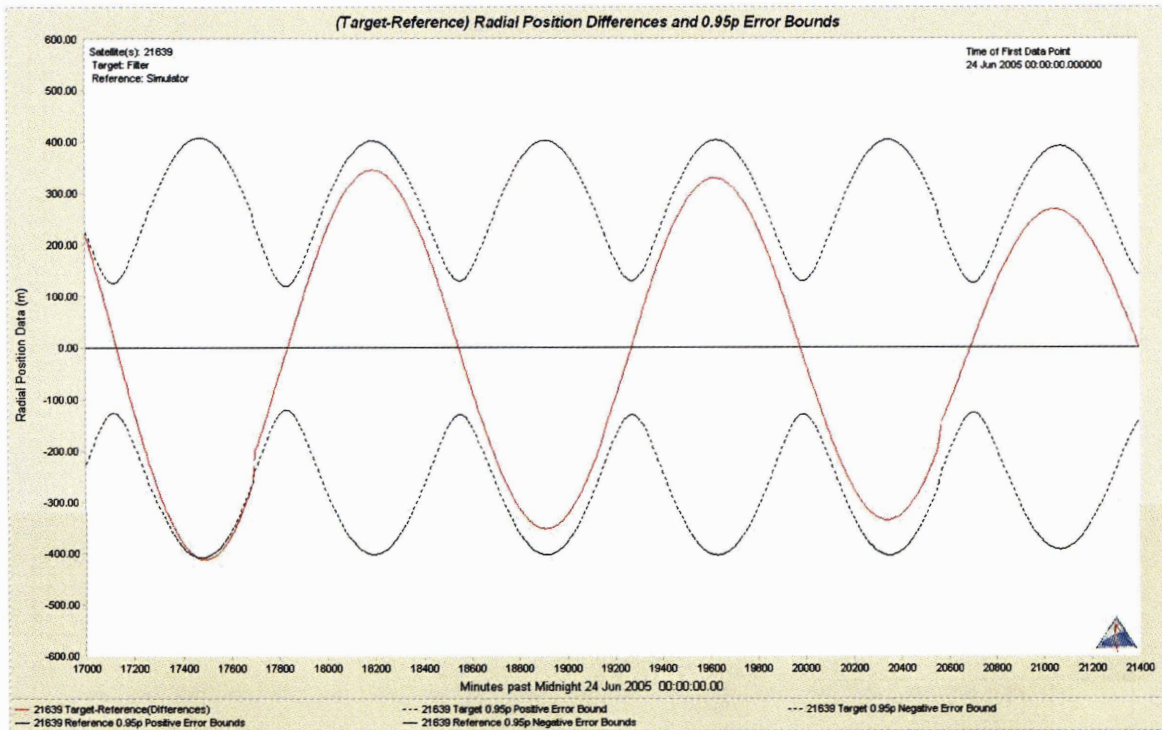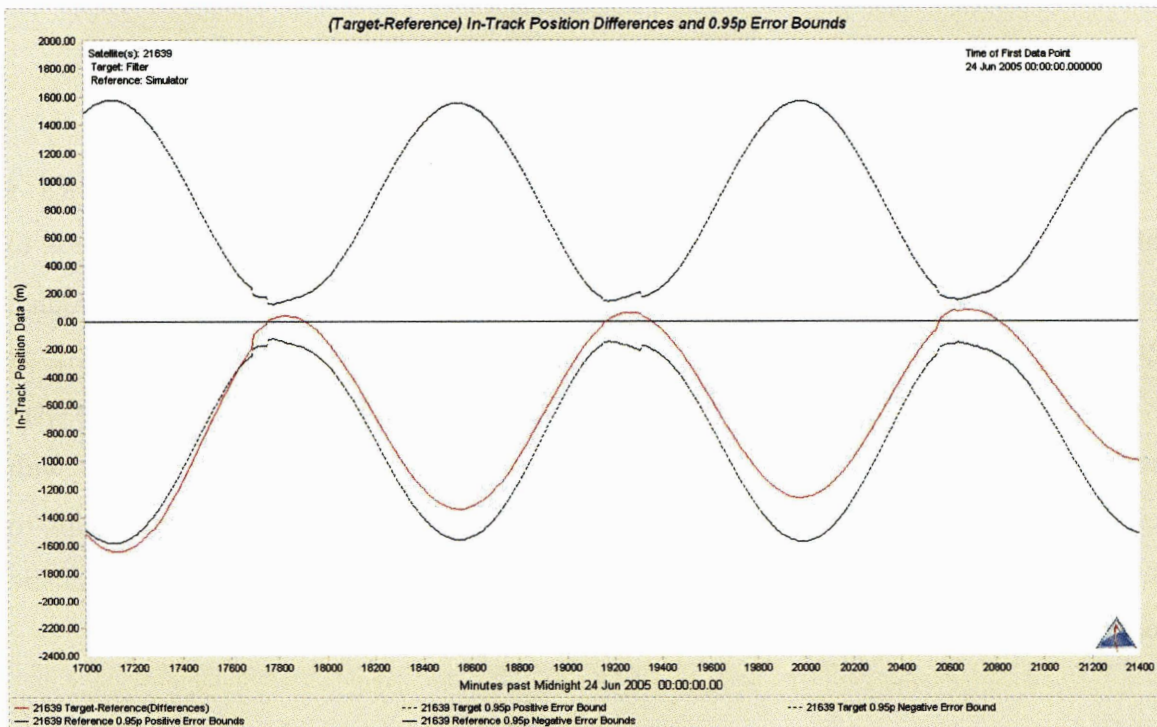*Figure 3.19. Radial position differences approx. for days 201 to 204, 2005.*
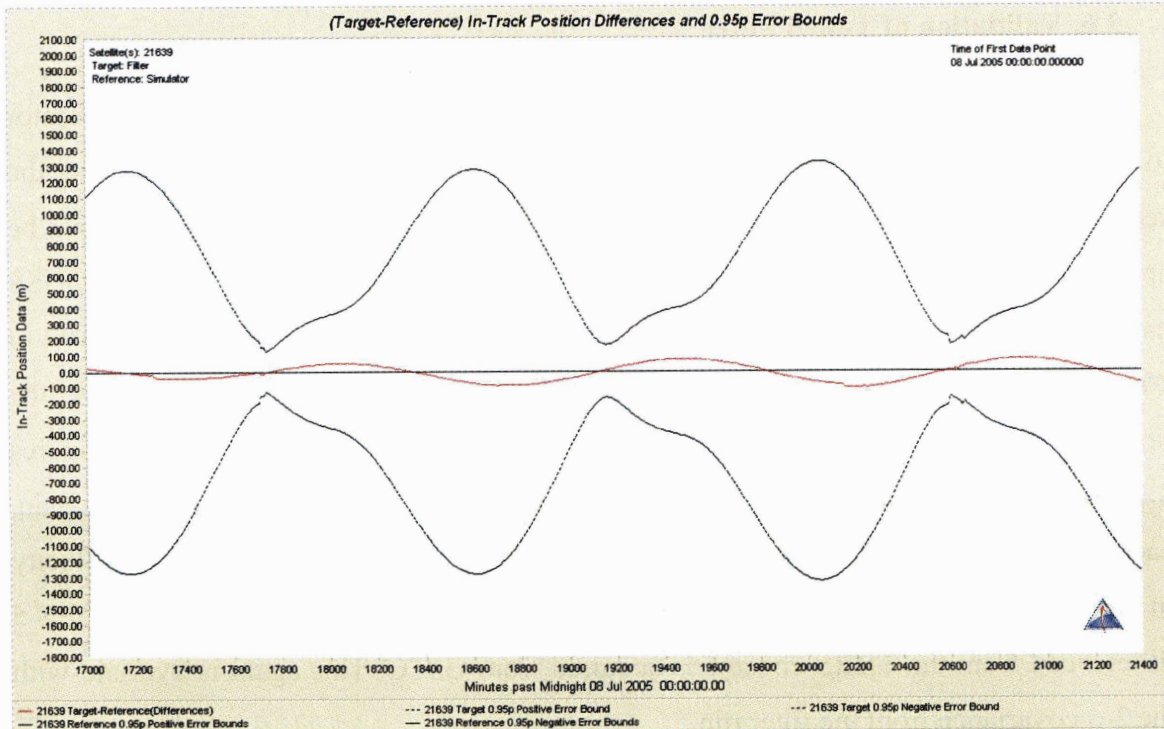
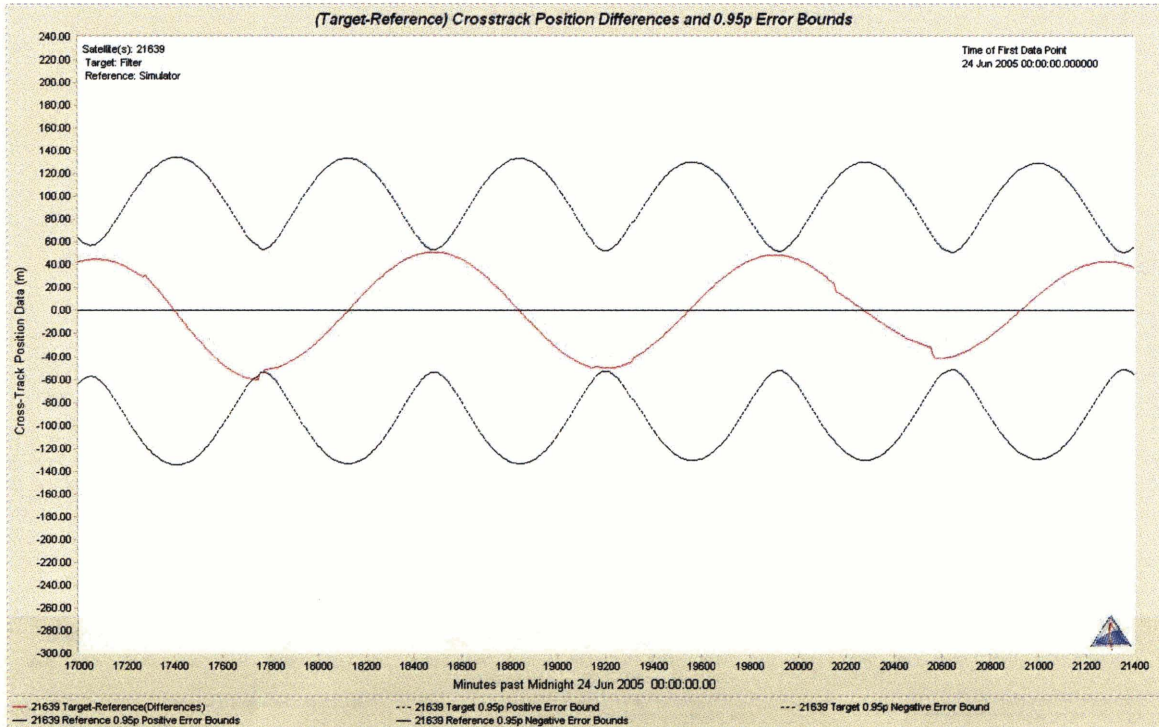*Figure 3.20. Intrack position differences approx. for days 201 to 204, 2005.*
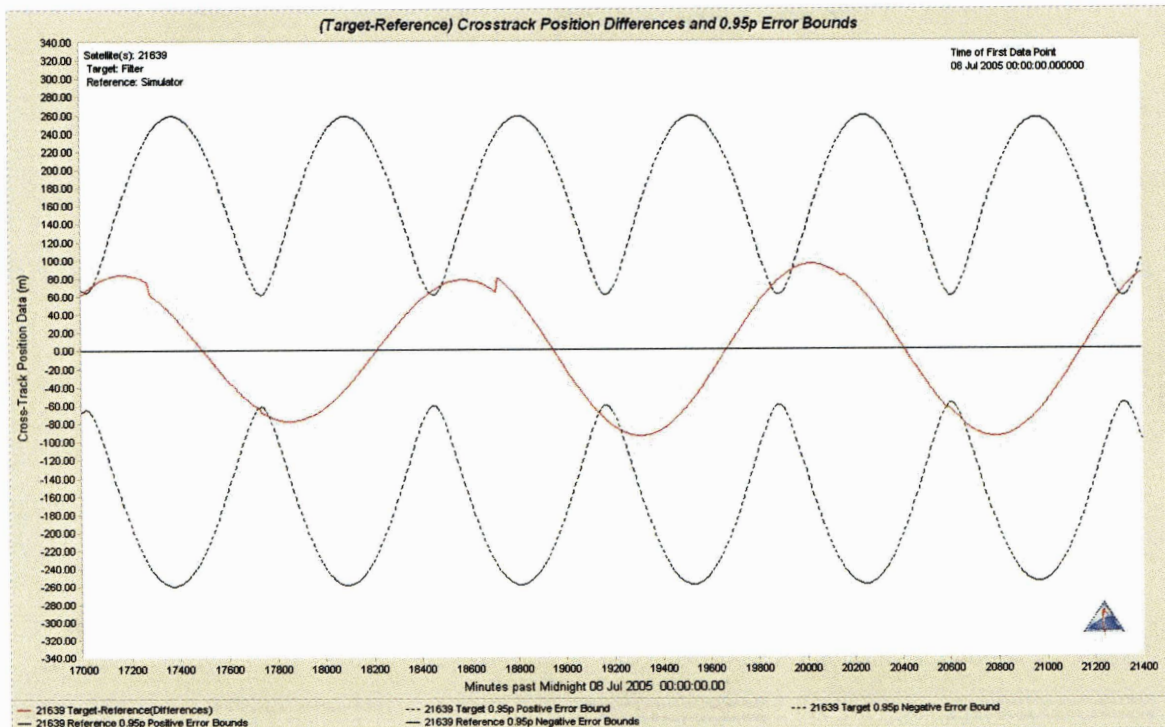
Figure 3.21. Crosstrack position differences approx. for days 201 to 204, 2005.
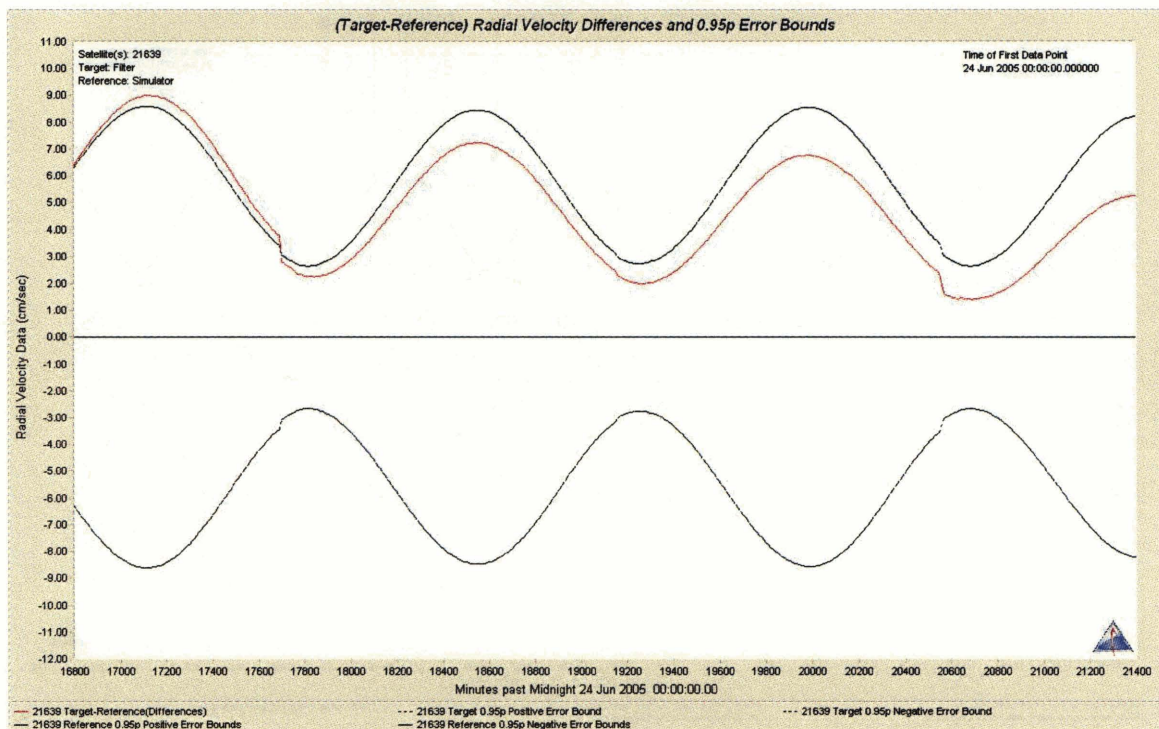
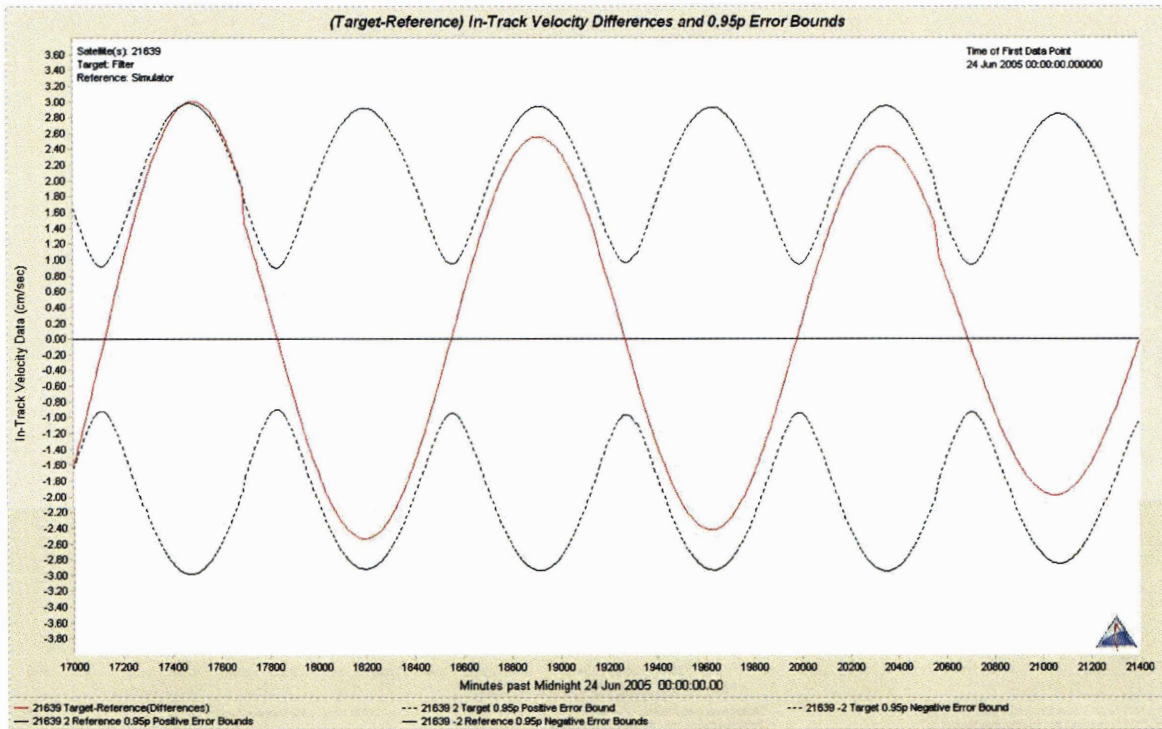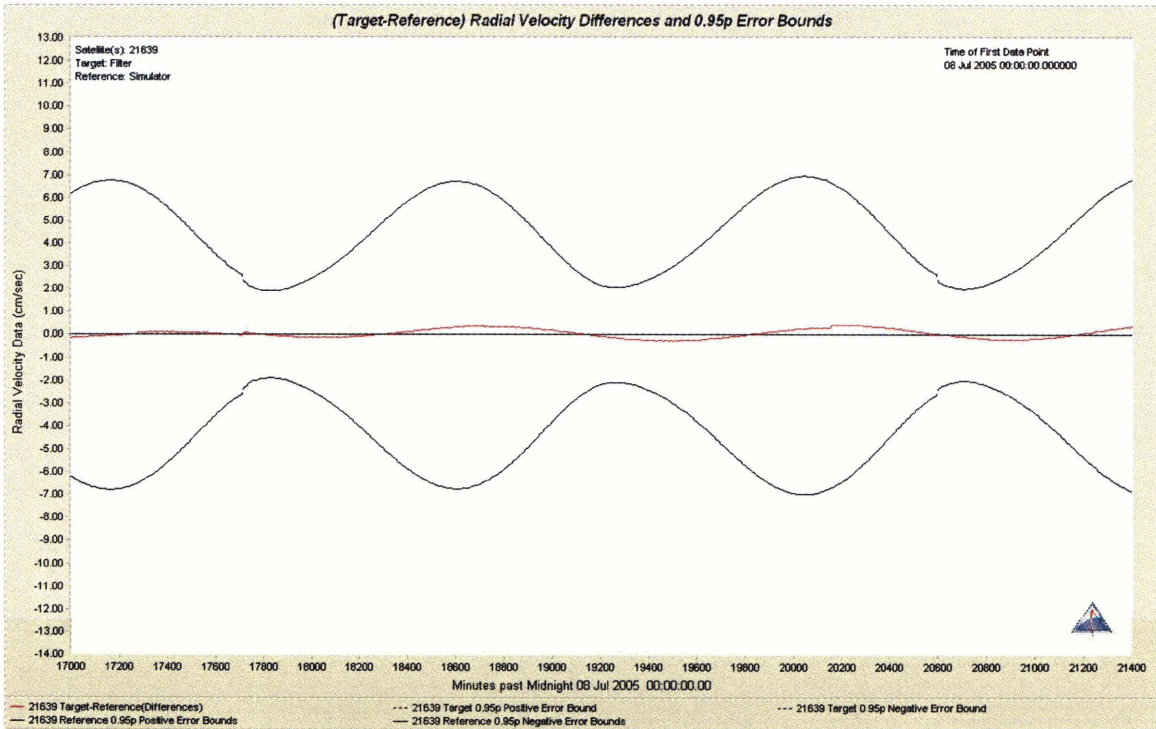*Figure 3.22. Radial velocity differences approx. for days 201 to 204, 2005.*
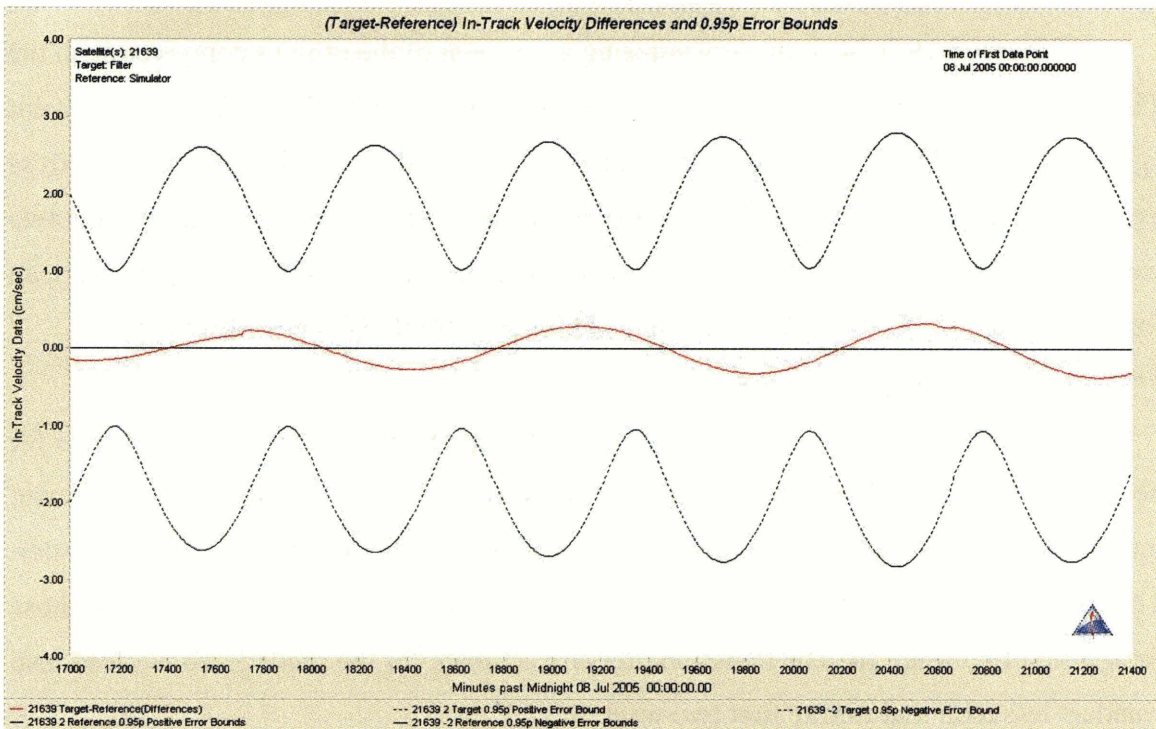
Figure 3.23. Intrack velocity differences approx. for days 201 to 204, 2005.
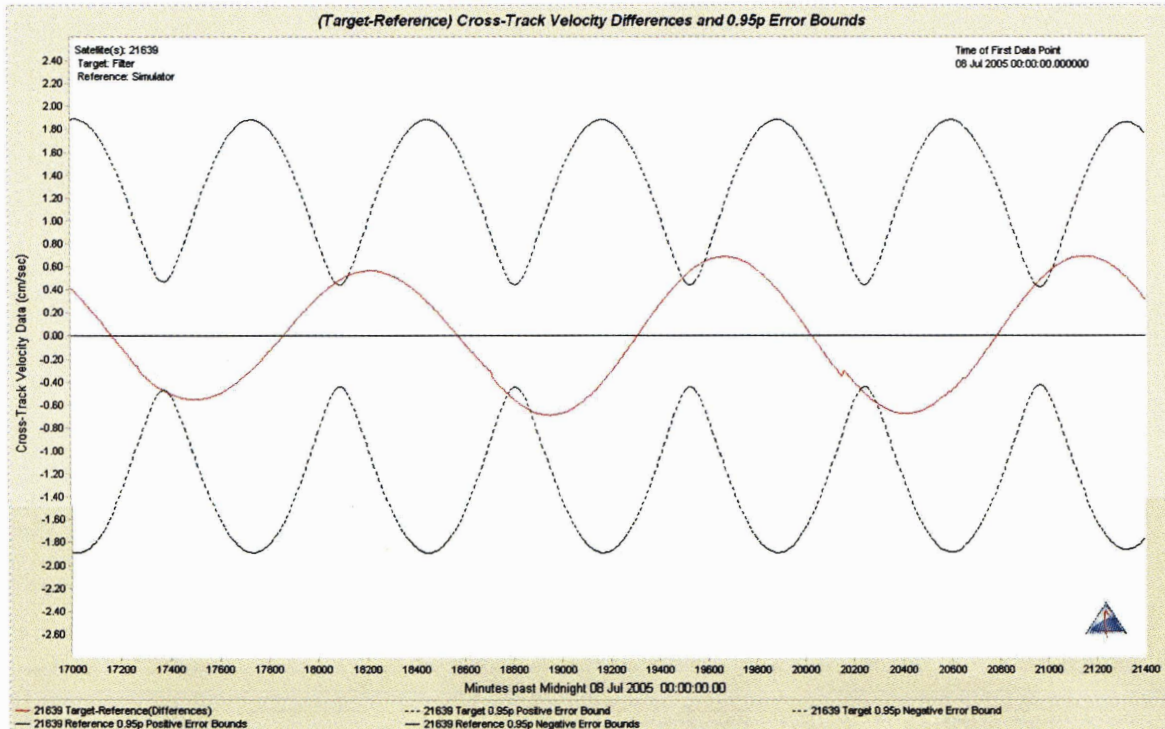
Figure 3.24. Crosstrack velocity differences approx. for days 201 to 204, 2005.

Figures 3.19-3.24 show very amazing agreement of the orbit as compared with the TDRS "truth" ephemeris. In all position components (figures 3.19-3.21), the fit orbit agrees with the TDRS "truth" to better than 100 m. The radial and along track orbit errors as evaluated by the "truth" comparison gained the largest improvement from the first two week fit (figures 3.7-3.12). Also, as opposed to the first two week fit, no biases are seen for the intrack position or radial rate components, nor correlations between the two.

First we must assume that the TDRS "truth" ephemeris is consistently accurate for both two week fits. The second filter fit (days 189 to 204) actually had 23% fewer measurements. It is possible that the time distribution for the optical tracking was better. The right ascension residuals for the second two week fit had less RMS scatter than those for the first two week fit. The radiation pressure estimate for the second two week fit also wanders less than that for the first two week fit.

Orbit fits made by DYNAMO (an MIT LL batch least squares orbit determination program) also confirmed all the above results. Permitting radiation pressure to be

estimated in both the two week fits, DYNAMO was within 6% of the 1.37 nominal solar radiation pressure scale factor value for the second two week fit and only within 13% of the 1.37 nominal value for the first two weeks.

General experience [Ref 9] has shown that the errors for the days 175 to 190 fit are more typical for an orbit derived from optical tracking only.

3.7. Overlap Study

Another method often used to assess orbit accuracy is a method known as overlap. This method uses two orbits that are computed which have a percentage of tracking data in common that permits the orbits to overlap but otherwise to have been independently calculated [Ref 9]. The typical rule of thumb is to allow less than 15% of the tracking data to be common between the two orbits. For this study, the accuracy assessment by the overlap method is compared with the covariance. Orbits are compared on day 189 which is in common over the two week orbit fits, days 175 to 190 and days 189 to 204. The plots included are that of the position overlap (figure 3.25) and velocity overlap (figure 3.26) for the two orbit fits on day 189.

The overlap assessment of position errors (figure 3.25) matches well with the error assessment of the fits from days 175 to 190 (figures 3.7-3.9). The same can be seen for the overlap assessment of velocity errors (figure 3.26) and the error assessment from the days 175 to 190 (figures 3.10-3.12). The error as shown by the overlap of the days 175 to 190 and days 189 to 204 is dominated by the error of the first fit span. This exemplifies a weakness of the overlap analysis which generally assumes that half of the overlap difference is due to error in each orbit. Yet, the overlap test seems to be able to provide an indicative error in the absence of a truth standard or realistic covariance.
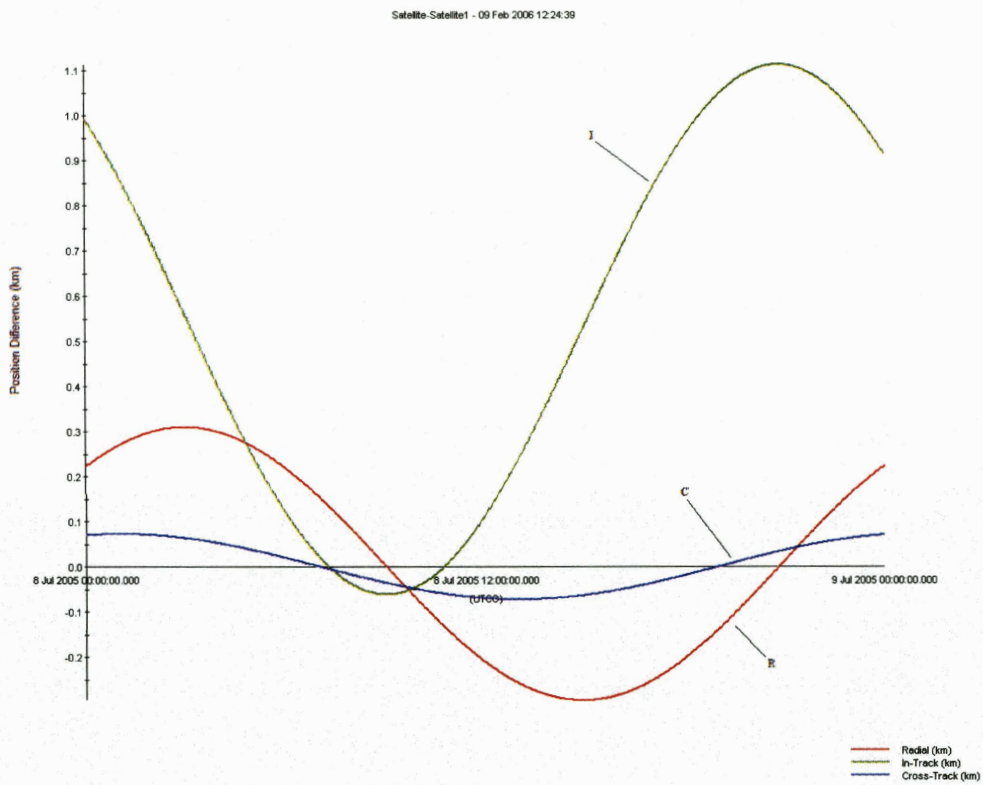
Figure 3.25. Position overlap (km) for day 189.
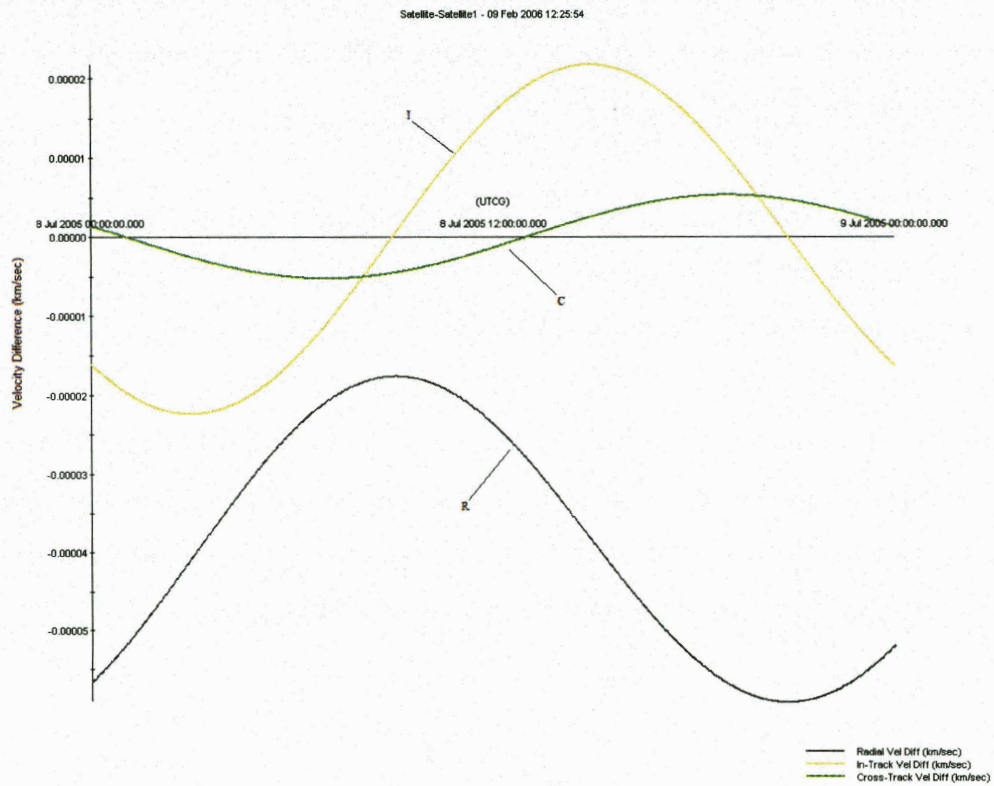
Figure 3.26. Velocity overlap (km/sec) for day 189.

3.8. Validating Covariances Using Measurements

Another method of evaluating orbit error is by using the actual measurements. The orbit error determined by this method can then be used to test the realism of the covariance to estimate the orbit error. Of course, an orbit can only be evaluated to the accuracy of those measurements.

A process to assess the quality of estimated covariance statistics from an orbit determination system was proposed by Dr. Jefferson Kommers of MIT LL [Ref 10]. The purpose of the following procedure is to evaluate the covariance estimates and possibly to help tune the a priori errors of the filter and to consequently improve the covariance accuracy. The method will be presented here with some application and discussion.

The data necessary to conduct the following procedure is optical tracking data that covers "on the order of 50 to 100 [orbit] revolutions with no more than 10 revolutions between any two tracking measurements and at least 200 line-of-sight measurements over the entire time span." Since the tracking data used in this study is from a geosynchronous satellite, there must be between 50 and 100 days of tracking data and no more than 10 days between data points [Ref 10].

The data set used is 6 weeks in length, with the first 4 weeks used to "tune" the filter, so it can provide the best orbit state and covariances, and the following two weeks to validate the covariances. The resulting orbit and covariance that are used are from the days 189 to 204 fit. These are then propagated forward and mapped to future observations of the following two weeks of data. For each optical right ascension and declination measurement time, a residual is computed from the resulting mapped observation by differencing the observed measurement from the measurement the propagated orbit would compute.

The next step is to scale the actual differences between the observations and the predicted measurements. This is done by computing the statistical distance (also known as the Mahalanobis distance [Ref 10]) given by:

$$d_i^2 = (\mathbf{z_i} - \mathbf{h(x(t_i))})^T \mathbf{H(t_i)}^{-1}(\mathbf{z_i} - \mathbf{h(x(t_i))}) \qquad \text{[eq. 1]}$$

where d_i^2 is the Mahalanobis distance, $\mathbf{z_i}$ is the observation vector, $\mathbf{h(x(t_i))}$ is the expected observation vector from the orbit estimation and $\mathbf{H(t_i)}$ is the associated covariance matrix.

If the measurement errors are Gaussian, the quantity $d^2 = \Sigma d_i^2$ should follow a chi-squared distribution with as many degrees of freedom as the measurement vector. As chi-squared has a distribution, one can compute the probability that d^2 is less than or equal to some value chi-squared as given by:

$$P(\chi^2|v) = \frac{1}{2^{v/2}\Gamma(\frac{v}{2})} \int_0^{\chi^2} \exp(-\tfrac{t}{2})\, t^{\frac{v}{2}-1} dt \qquad \text{[eq. 2]}$$

Its complement,

$$Q(\chi^2|v) = 1 - P(\chi^2|v) \qquad \text{[eq. 3]}$$

is the probability that the observed d^2 exceeds some value of chi-squared. For example, in the optical case with $v=2$ degrees of freedom (right ascension and declination), the probability that a random value of chi-squared drawn from its distribution will equal or exceed 0.0201 is 99 out of 100. The probability that it will equal or exceed 5.991 is 5 out of 100 [Ref 11]. Functions for computing these distributions are available in most mathematical and statistical software packages, including MATLAB.

In another approach, the cumulative distribution of d_i^2 can also be compared graphically with the chi-square theoretical distribution using the Kolmogorov-Smirnov (K-S) statistical test [Ref 12]. The K-S test evaluates how closely the cumulative and theoretical distributions follow each other or more technically, assesses the hypothesis that the d_i^2 values are drawn from the expected distribution. Practically, the K-S test determines whether or not the measurement data followed a Gaussian distribution and that the other filter model errors are properly accounted for. If the K-S Test is satisfied,

the covariances are accurate. If the K-S Test is not satisfied, then there is something wrong with the data (possible biases or error assignment) or the models used to estimate the orbit and covariances. The K-S statistic is the difference, D, between the two cumulative distributions at their maximum absolute difference. The K-S test also generates a statistic that measures the significance of any observed non-zero value of D or the probability, p, that the distributions are different – small p-values indicate that the distributions are significantly different. The ideal case would be for the K-S statistic to be 0.

3.8.1. Procedure

With the filter settings that have been verified, the filter run over days 189 to 203 is propagated out another two weeks (days 203 to 217 of 2005). This is a propagation and not a fit to any measurement data. During the propagation, the ODTK software maps the orbit and covariance to the right ascension and declination measurement space at the time of the real tracking measurements. It compiles these into a measurement residuals report and they can be appropriately graphed.

For this study we first show the results of this test if the measurement data were only computed with Gaussian distributed errors and no other error sources. Simulated data are created for the same time period covered by the real measurement data. Table 3.2 is a detailed method for this process with the ODTK.

After results are shown through the controlled simulation, real data will be processed. The method to create the measurement residuals report using real data is the same as in table 3.2, however, the simulator is not used, so those portions of the method must be ignored.

Once the measurement residuals report has been exported from ODTK, this can be read into a software program to graph and evaluate the data. MATLAB was used for this purpose. The MATLAB program written to evaluate this data first reads in the

49

measurement residuals report file. The program then makes a column of the residual ratios and sorts it in ascending order. The residual ratios are then indexed from 0 to 1 in ascending order to create their actual cumulative distribution. An imbedded MATLAB command ("normcdf") is then used to create the residual ratios theoretical standard normal cumulative distribution. The two cumulative distributions are then plotted together. The program also creates a bar graph histogram of the residual ratios for their actual probability distribution function. This is graphed with the theoretical standard normal probability distribution function that is built by the imbedded MATLAB command ("normpdf"). The program also performs the K-S Test which computes both K-S statistic or "D" value, and the probability statistic "p." The program can be found in Appendix B.

Method for Simulated Data
Create Scenario: Satellite, Filter, Simulator, Tracking System
Set Simulator>StartTime
Set Simulator>StopTime
Set Simulator>CustomTrackingInterval>Enabled = True
Set Simulator>CustomTrackingInterval>Schedule
Run Simulator
Set Filter>ProcessControl>StartMode = Initial
Set Filter>ProcessControl>StartTime
Set Filter>ProcessControl>StopMode = TimeSpan
Set Filter>ProcessControl>TimeSpan = length of time desired
Run Filter
Set Filter>ProcessControl>StartMode = Restart
Set Filter>ProcessControl>TimeSpan = length of time desired
Set Satellite>MeasurementProcessing>ResidualEditing>NominalSigma = 0
Run Filter
Using final .filrun file, create "measurement residuals" report
Export "measurement residuals" report as a file that can be used by software to graph and evaluate

Table 3.2. Method to create simulated data "measurement residuals" report that can be manipulated to evaluate the cumulative distribution of measurement residuals against the theoretical distribution of measurement residuals.

3.8.2. Results

The following plots demonstrate both simulated and real data applied to the K-S Test. The first plot (figure 3.27) was created using simulated data with 2 arcsec random error. Again, we want to see the results of the test if no other error source was introduced but Gaussian or random error measurement errors. Figure 3.27 shows the cumulative distribution function (CDF) for the simulated data against the expected CDF for the simulated data. Figure 3.28 shows the probability distribution function (PDF) for the same set of data against the expected Gaussian curve. Figure 3.28 demonstrates how well the data approximates a Gaussian distribution. This is indicated by the relatively high p-value of 0.9183 for figure 3.27. That value means that there is a 0.9183 probability that the simulated data were drawn from a Gaussian distribution. A reason that the probability was not 1.0 can possibly be seen from figure 3.28. Even with 1680 samples of measurements (as we have in this case), there is some skewness of the distribution compared with the theoretical.
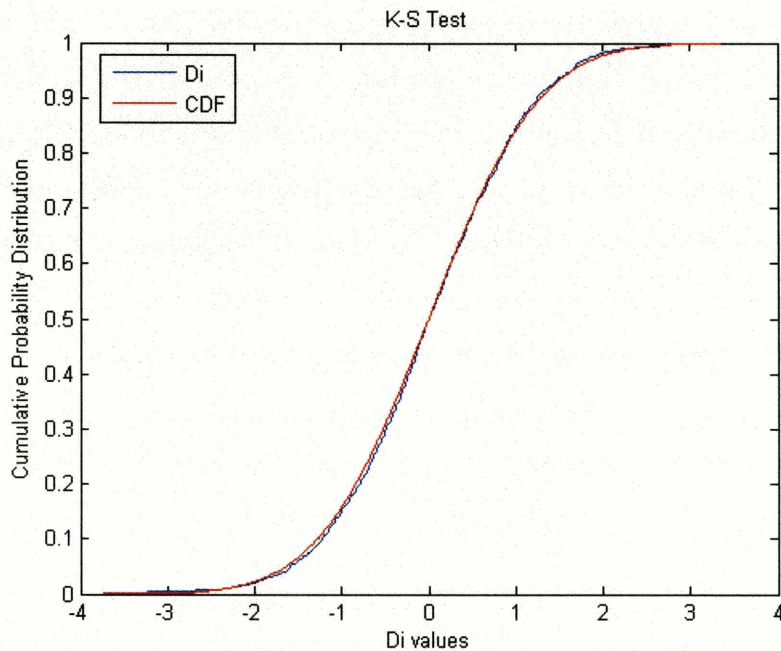
Figure 3.27. K-S test with 2 arcsec simulated data and CPSigma =0.04. Resulting K-S statistic =0.0190, P-value =0.9183. "Di" refers to the expected CDF of the data. "CDF" refers to the theoretical CDF of the simulated data.
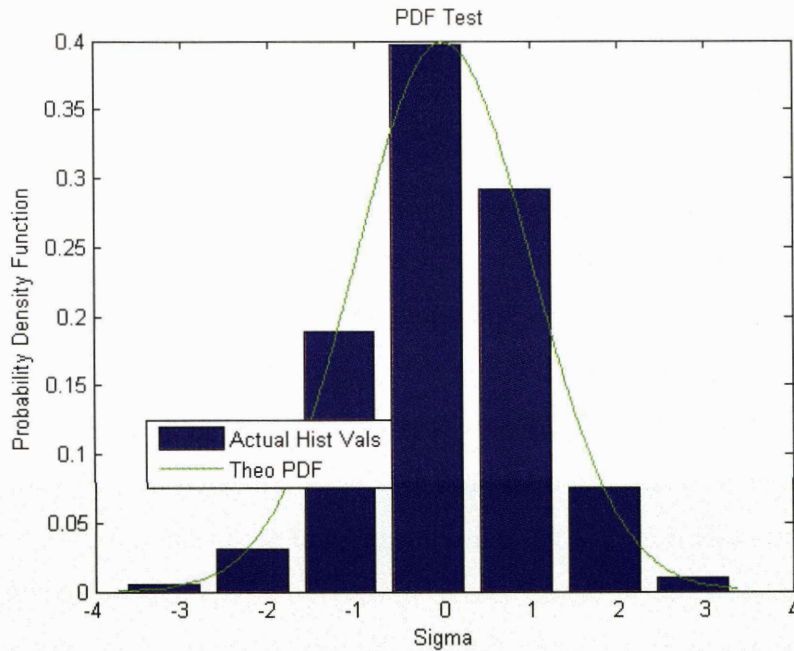
Figure 3.28. PDF histogram of Figure 3.27. "Actual Hist Vals" refers to the PDF for the data. "Theo PDF" refers to the theoretical PDF of the simulated data.

It was also interesting to make the following test. The measurement errors were as before, Gaussian 2 arcsec. However, if the filter were also given the wrong radiation pressure via the solar radiation pressure scale factor (*CPNominal*), and this error was also appropriately given to the filter via an a priori sigma for the radiation pressure model, what would the K-S test show? Therefore, the data were simulated with a solar radiation pressure scale factor of 1.37, but was changed to 1.41 for the filter run to attempt to reflect the possible error associated with the simulation. The *CPSigma* of the filter run was set to 0.04 to account for the error introduced to the radiation pressure offset. Figure 3.29 shows the results for this change in the simulation. The D or maximum distance between distributions was 0.0333 with a p-value of 0.3033.
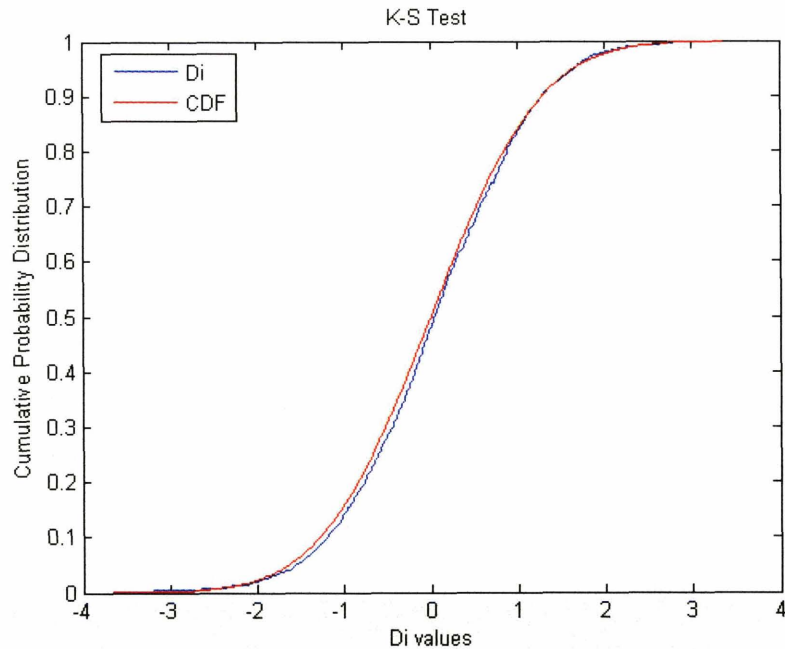
Figure 3.29. K-S test with 2 arcsec simulated data and CPSigma = 0.04. Simulation CPNominal = 1.37. Filter CPNominal = 1.41. Resulting K-S statistic =0.0333, P-value =0.3033. "Di" refers to the expected CDF of the data. "CDF" refers to the theoretical CDF of the simulated data.

Although the difference is subtle, the data from the figure 3.27 matches better with the expected value than does the data from the figure 3.29. More simulations along this line are suggested for future study to understand the difference between the orbit error introduced by the mid-model of the radiation pressure and the effect on the covariance given some information about the error, and also the sensitivity of this test to that difference.

The next set of plots demonstrates the K-S Test when real data are used. For the filter run with real data, a 2 arcsec measurement error for right ascension and declination is first used. Also, a *CPSigma* of 0.025 is used for the error of the *CPNominal* of 1.37. Figure 3.30 shows the result of the K-S test. The maximum "D" distance between the observed and theoretical distribution was 0.2335 with a near zero significance ($\sim 10^{-10}$). This definitively indicates a mismatch between the actual error and that error expressed by the covariance.
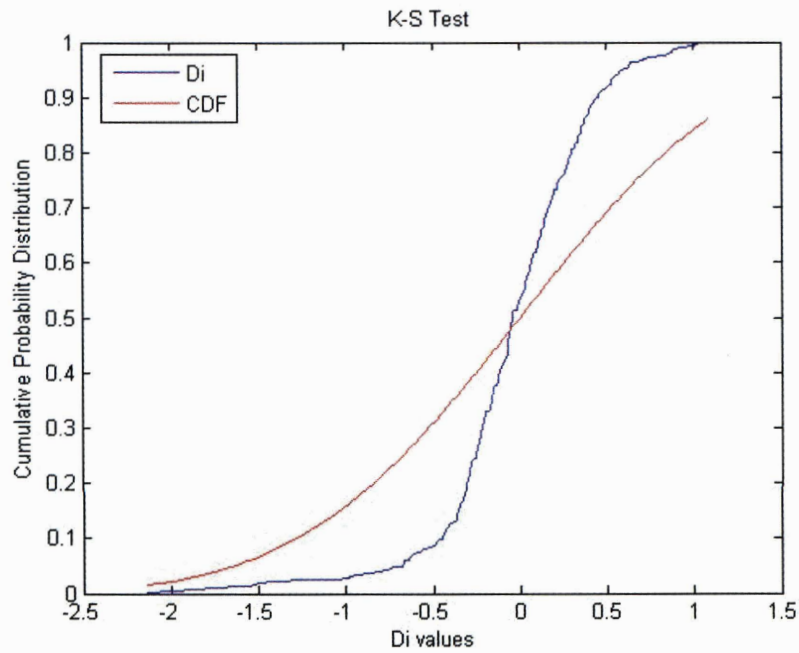
Figure 3.30. K-S test with 2 arcsec real data and CPSigma = 0.025. K-S statistic = 0.2335. P-value = 1.2173 x 10^(-10). "Di" refers to the expected CDF of the data. "CDF" refers to the theoretical CDF of the real data.
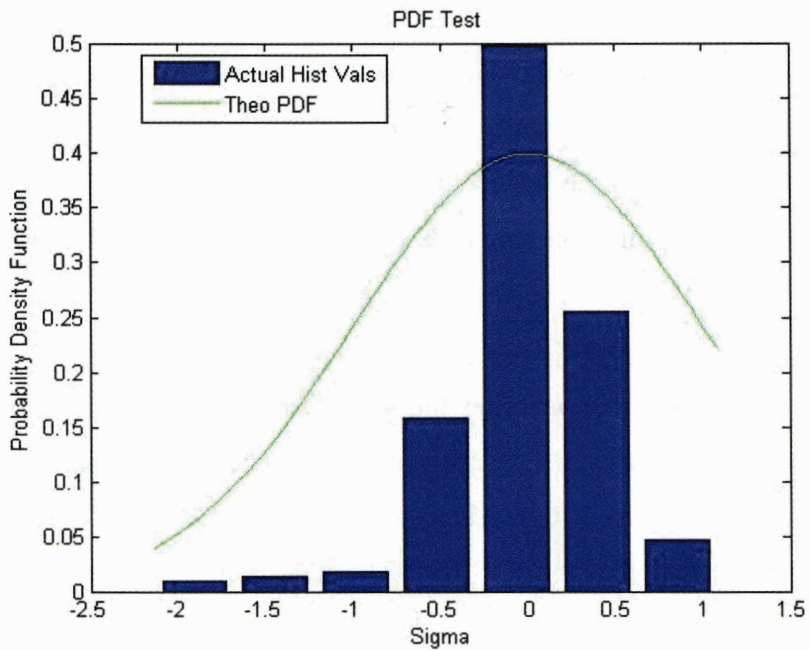
Figure 3.31. PDF histogram of figure 3.30. "Actual Hist Vals" refers to the PDF for the data. "Theo PDF" refers to the theoretical PDF of the real data.
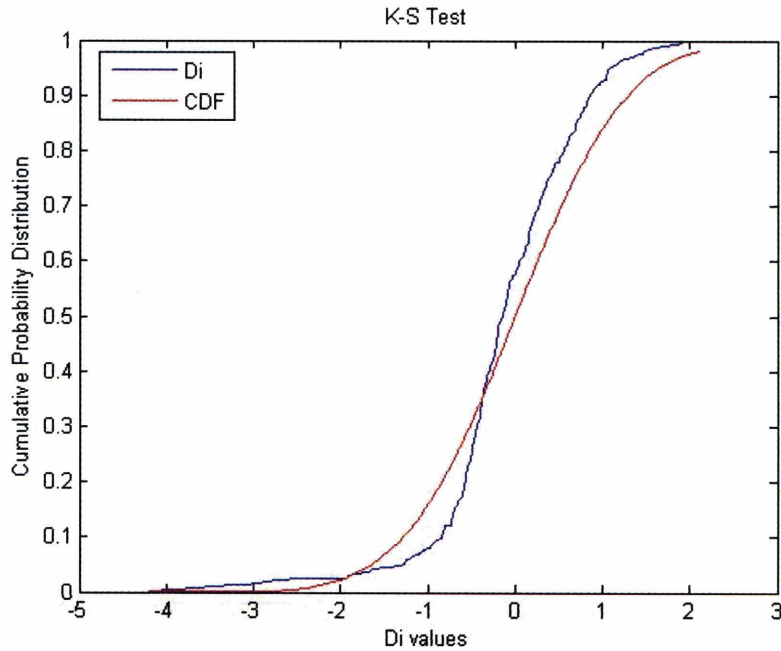
Figure 3.32. K-S test with 1 arcsec real data and CPSigma = 0.025. K-S statistic = 0.1085. P-value = 0.0125. "Di" refers to the expected CDF of the data. "CDF" refers to the theoretical CDF of the real data.

Figure 3.31 shows the PDF histogram of the real data with the theoretical PDF. Since most of the data points in figure 3.31 have a near zero "Sigma" value, the histogram appears to be primarily constrained around -0.5-σ and 0.5-σ. This means that our assumption of 2 arcsec error for the data is probably too pessimistic. The filter was rerun using measurement error of 1 arcsec. Figure 3.32 displays the results.

As can be seen, the 1 arcsec data better adheres to the distribution than does the 2 arcsec data (p-value of 0.0125). This does confirm that the assumption of 2 arcsec measurement error is pessimistic and that the real data are more accurate than that. However, this is the stated accuracy for the data; and for the remainder of this thesis we will stay with this assumption. Also, the remainder of this thesis will be based on simulated data where it will be assumed that the a priori errors are considered and modeled by the filter.
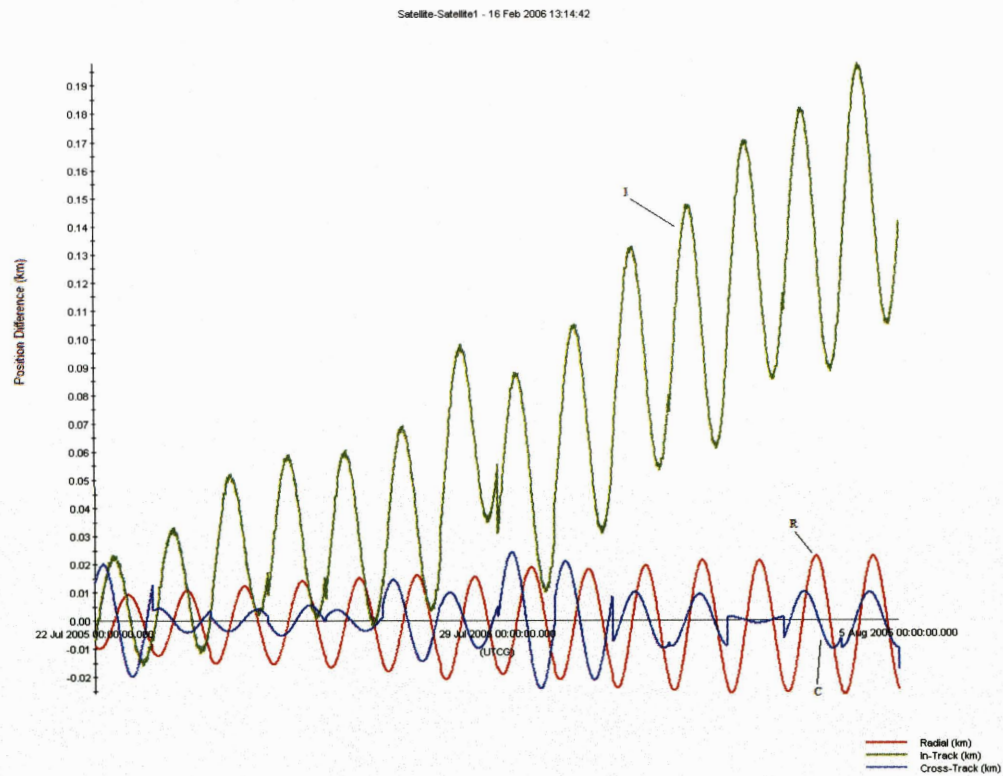
Figure 3.33. TDRS ephemeris vs. the propagated ephemeris (km).

Kommers also suggested a variation on the procedure that would use a truth ephemeris for the reference state in place of the propagated state vector $\mathbf{x}(t_i)$ [Ref 10]. Here the covariance would be evaluated against the best possible "truth" determination of the orbit. Let $\mathbf{x'}(t_i)$ denote state derived from the truth ephemeris. Then the same statistical tests outlined above could be performed using eq. 1. Note that the covariance used is still that based on the propagation of the estimated orbit $\mathbf{x}(t_0)$. Practically, this method would require finding a way to replace the propagated state vector with the truth ephemeris and mapping it to the measurements. It is assumed that the differences between the measurements and the truth ephemeris may need to be calculated by an external program to ODTK. This is left for future work. Figure 3.33 shows the comparison of the propagated covariance with the orbit differenced with the TDRS 5 truth ephemeris. The difference never exceeds more than 200 meters for the entire two week fit, so we might not see much difference in the K-S test using this truth ephemeris.

The purpose of the inclusion of this method of evaluating covariance accuracy was to develop methodology and test it to some extent. Clearly more study is required. As Richard Hujsak noted [Ref 13], this test only measures two dimensions of a multidimensional covariance; although a realistic covariance still must satisfy it. As Hujsak also noted, for example, predicted time varying biases tend to damp to zero over time while covariances diverge. The covariance diverges while correlations between biases and the orbit tend to zero. This is one motivation for doing what was discussed in the previous paragraph. Finally, our experience with this test is that is has extreme sensitivity to improper error modeling.

[This page intentionally left blank.]

4. Maneuver Detection

4.1. Introduction

The purpose of this section is to show the capability of ODTK to detect a single general maneuver and to parameterize the size of a maneuver that is detectable along with the time needed after the maneuver to detect. The parameterization of maneuver detection was done in two ways. The first was to examine whether a maneuver can be detected following a period where a converged orbit has been obtained. In order to achieve this, two weeks of data were used with a realistic tracking density (2 tracks per day for the entire two weeks). The maneuvers input at the end of the two weeks were of three different sizes: 0.01 m/s, 0.1 m/s, and 1.0 m/s. All the maneuvers were Δv in the intrack direction (as is common for GEO satellites that are performing station keeping or relocation maneuvers). This work will not consider North-South maneuvers which involve changing the orbit plane and require a large Δv on the order of 1 m/s. Following the maneuver, one more track was observed at 12 hours, 6 hours, 3 hours or 1 hour, post maneuver. Using ODTK and its capability to simulate measurements, this process can be done quite simply. All that is needed to simulate the maneuver is to add the maneuver located under *Satellite>ForceModel>InstantManeuvers*. In order to "detect" that maneuver, once the simulated measurements have been created, remove the maneuver and run the filter without the maneuver. The full process can be seen in table 4.1.

Simulated Maneuver Detection Method
Create: Scenario, Filter, Satellite, Simulator and Tracking System
Create Tracking Schedule: Simulator>CustomTrackingInterval>Schedule
Create Maneuver: Satellite>ForceModel>InstantManeuvers> Add… InstantManDeltaV
Specify InstantManDeltaV>Epoch and InstantManDeltaV>DeltaV
Run Simulator
Remove Maneuver from Satellite: Satellite>ForceModel>InstantManeuvers> Enabled = false
Run Filter
View "Residual Ratios" plot

Table 4.1. Detect maneuver process.

This allows ODTK to create a maneuver, but not know that it exists when running the orbit filter. After running the filter, the residual ratio report, as discussed previously, will show the residuals from the filter run. Unless the orbit covariance going into the maneuver is vary large, the final track (after the maneuver) will not fit as the other tracks did, and this is evidence that "something happened" or, more precisely, a maneuver occurred.

Before proceeding, the choice of the Δv sizes of 0.01, 0.1 and 1.0 m/s is discussed. If one is given the initial orbit state, a destination orbit and a time to achieve the desired destination orbit, Lambert's rendezvous problem can be used [Ref 14]. The Lambert problem can be both an exercise in IOD and targeting (rendezvous and intercept). The known parameters are two position vectors, r_1 and r_2, and the time-of-flight between them. This information can be used to determine an initial velocity, v_1, in the IOD problem, or the Δv required to maneuver from one position to another in the desired amount of time for targeting [Ref 2].

For this work, we use a parametric range of the Δv's used for satellite station keeping as well as satellite relocation maneuvers. The ODTK has various options for modeling maneuvers. For this study all maneuvers will be considered to occur instantaneously or impulsively. In practice, an acceleration is given over a specified time interval to give a resultant Δv. Table 4.2 shows the resulting change in semimajor axis and resulting longitude drift rate for the three Δv's used in this study.

Δv (m/s)	Δa (Δkm)	Longitude Drift Rate (°/day)
1.0	27.5	-0.353
0.1	2.75	-0.0353
0.01	0.275	-0.00353

Table 4.2. Parameterization of Δv's and resulting changes in semimajor axis and longitude drift rate.

Longitude drift rate was calculated using the following approximation:

$$d = K\Delta a \qquad \text{[eq. 4]}$$

where d is the longitude drift rate (a positive value indicates drift in the eastward direction), K is -0.0128416 °/day/km, and Δa is the change in semimajor axis due to the maneuver [Ref 15]. This helps describe the problem parametrically without giving specific missions or intentions.

4.2. Dense Tracking Maneuver Detection

The following plots are the residual ratio plots taken after two weeks of dense data and with a maneuver modeled impulsively (or instantaneously in these runs) of varying sizes and track times after the maneuver.

The first figures (figures 4.1-4.4) show the resulting residual ratio plot for a Δv of 1.0 m/s and for each a track of 12, 6, 3, or 1 hour post maneuver. As can be clearly seen, this size of a maneuver can be easily detected in this manner. For figures 4.1-4.2, both right ascension and declination are affected by the maneuver. However, in figures 4.3-4.4, only the right ascension residual reflects the maneuver.
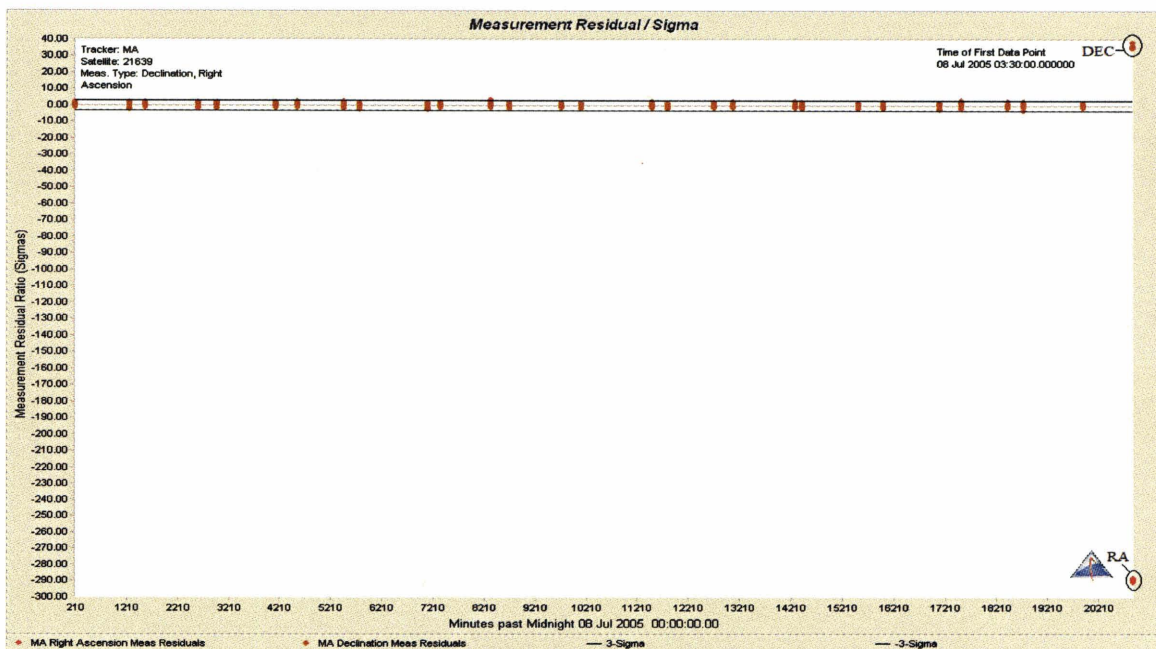
Figure 4.1. $\Delta v = 1.0$ m/s. Track 12 hours after maneuver.
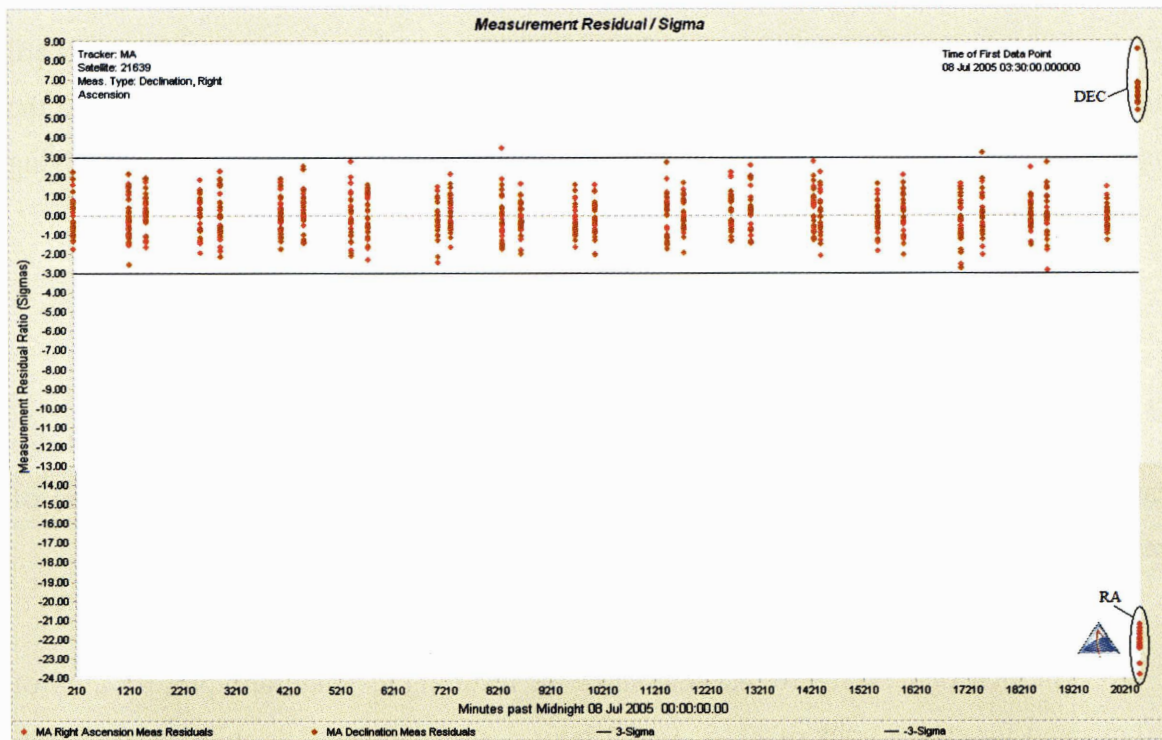
Figure 4.2. Δv = 1.0 m/s. Track 6 hours after maneuver.
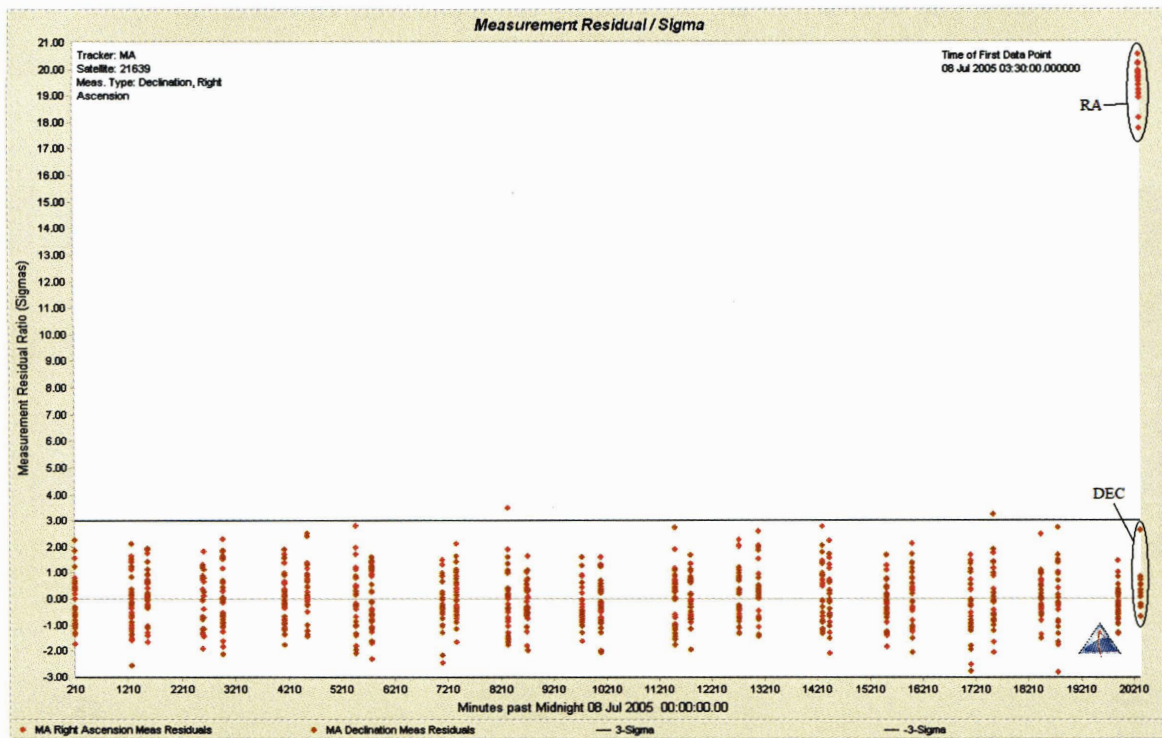
Figure 4.3. Δv = 1.0 m/s. Track 3 hours after maneuver.
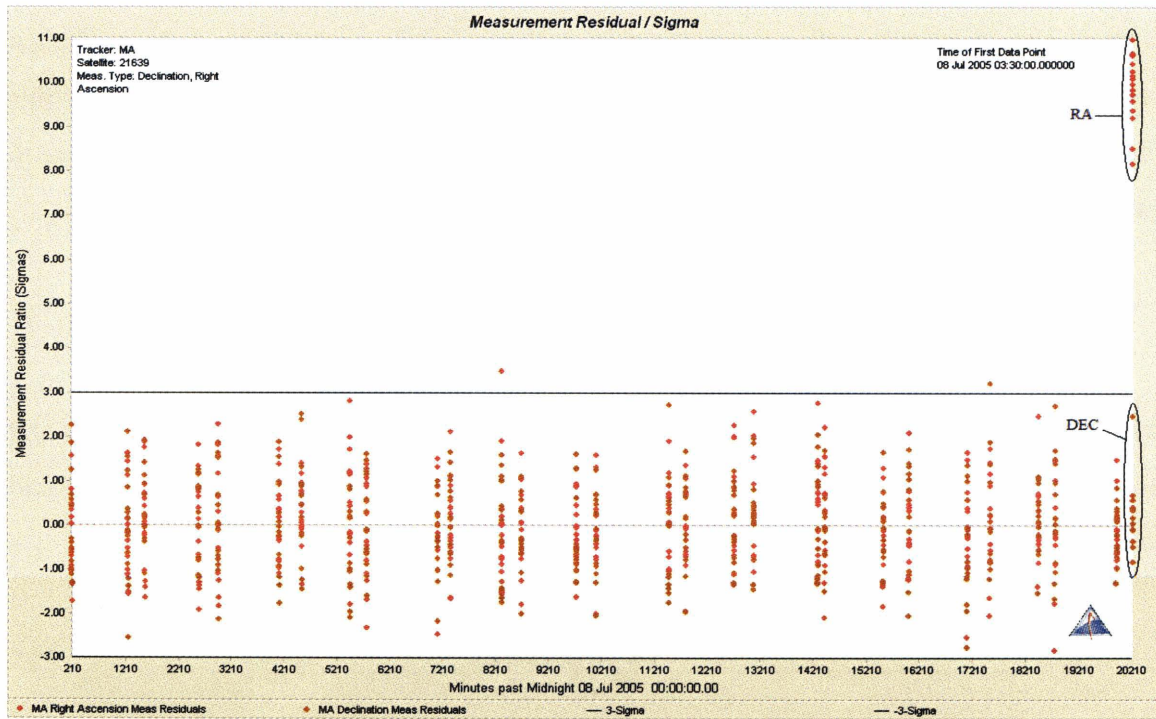
Figure 4.4. Δv = 1.0 m/s. Track 1 hour after maneuver.
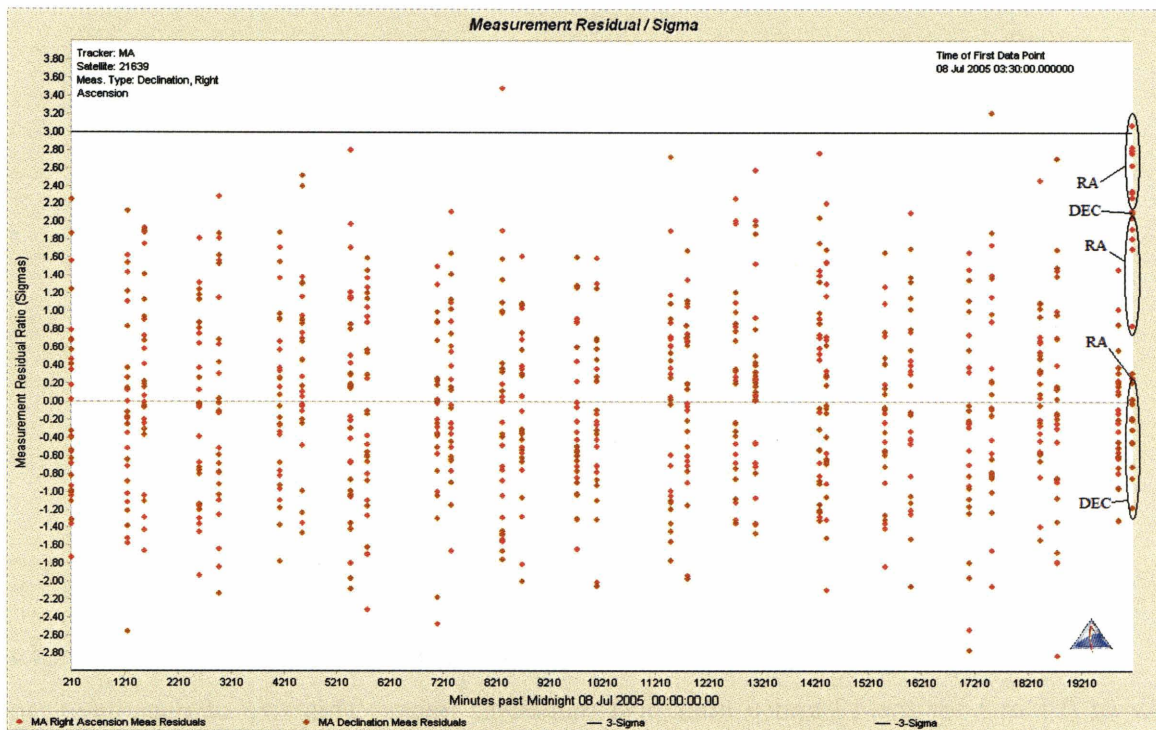
Figure 4.5. Δv = 1.0 m/s. Track 15 minutes after maneuver.

Even a track just 1 hour after the 1.0 m/s maneuver is still obvious (figure 4.4), so a track just 15 minutes after the maneuver is shown in figure 4.5. The right ascension residuals are barely beyond the 2-σ range predicted by the covariance. To show how this may be enhanced, a running mean of the right ascension residuals would appear as in Figure 4.6. The equation used to calculate each point of the running mean is:

$$\bar{a} = (a_1 + 2a_2 + a_3)/4 \qquad \text{[eq. 5]}$$

where $\bar{a}$ is the average, a_1 is the previous measurement, a_2 is the current measurement and a_3 is the next measurement. If the covariance was indeed realistic, then this would indicate a maneuver has occurred with that level of confidence.
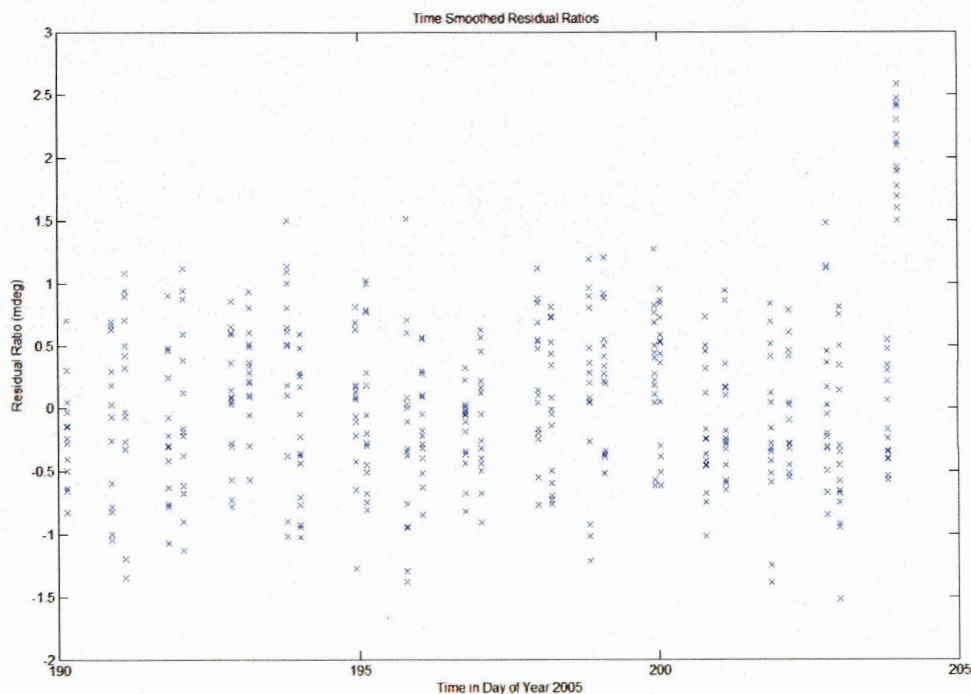
Figure 4.6. Running mean of right ascension residual ratios.

The next two plots (figures 4.7 and 4.8) show the resulting residual ratio plot for a Δv of 0.1 m/s for the 12 and 6 hour post maneuver tracks. This size of maneuver can clearly be seen after 12 hours. However, for the 6, 3, and 1 hour cases, the maneuver results in residual ratios that clump the declination and right ascension, separately,

64

together. Results are only shown for the 6 hour case in figure 4.8. This is once again some evidence that something must have happened.
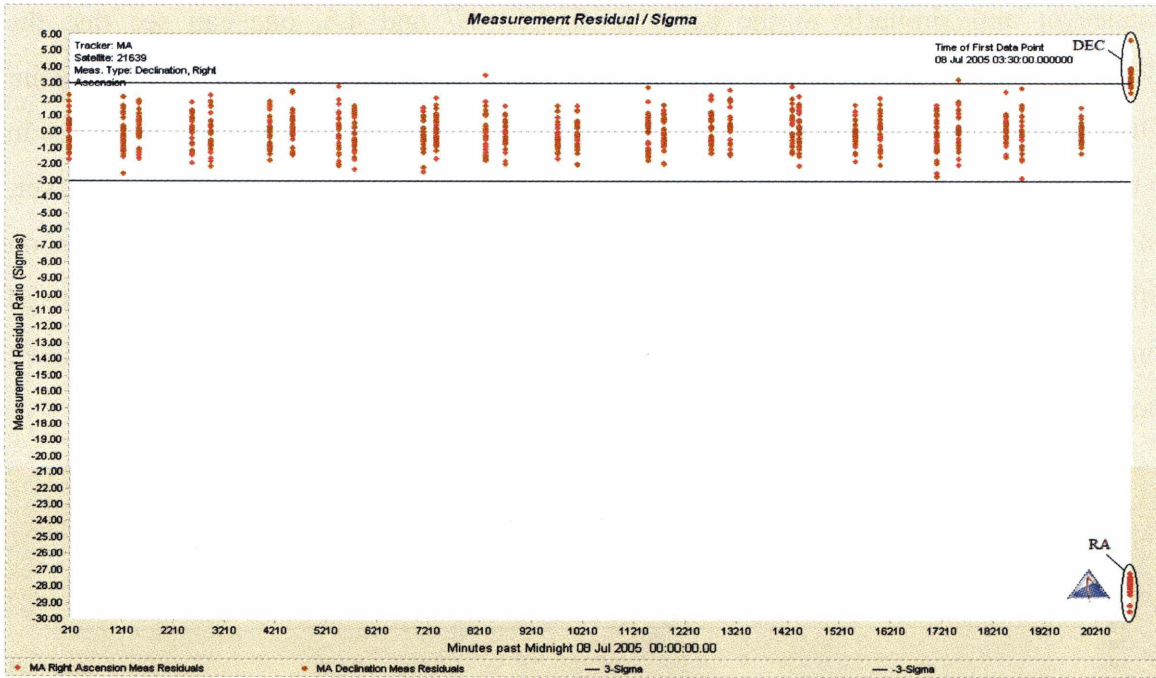
Figure 4.7. Δv = 0.1 m/s. Track 12 hours after maneuver.
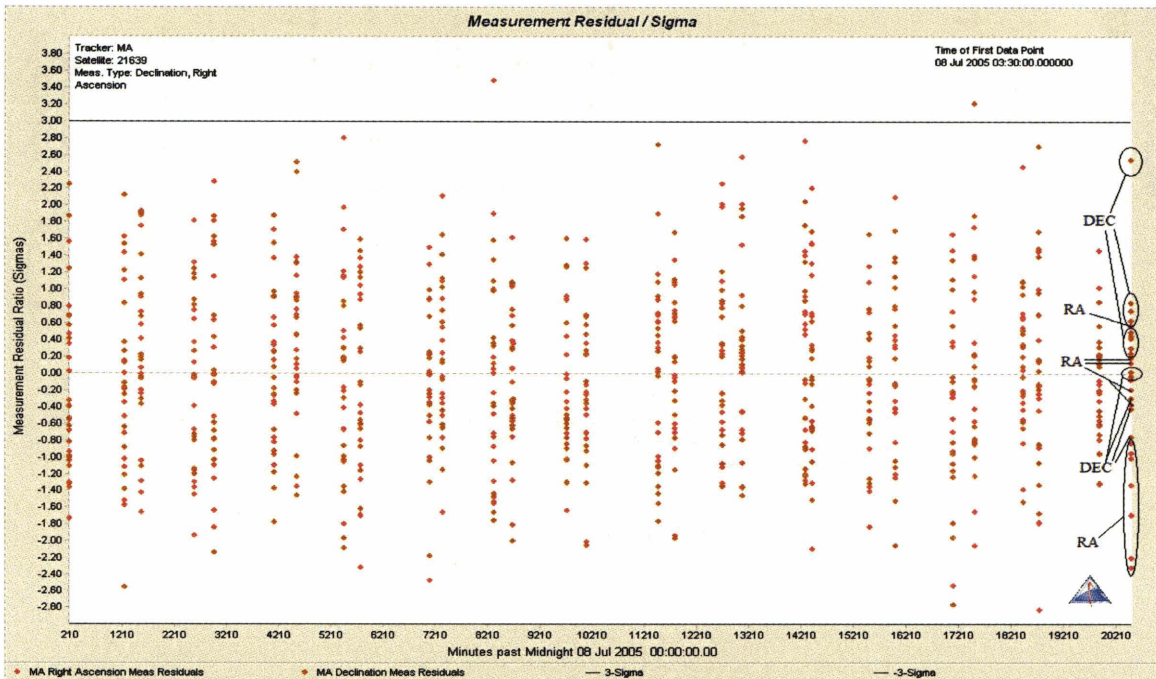
Figure 4.8. Δv = 0.1 m/s. Track 6 hours after maneuver.

For the final case for Δv of 0.01 m/s (figures 4.9 and 4.10), it is much more difficult to detect the maneuver. Figure 4.9 shows the case of a track 12 hours after the maneuver and similarly to the cases in figures 4.7 and 4.8, one can see that the declination and right ascension residual ratios have separately clumped together again. Once again, this is evidence of some change in the orbit. If one were to encounter this in a real situation, it may be more difficult to conclude that a maneuver occurred, However, in this case, the maneuver size is so small (0.01 m/s), that the orbit is hardly affected soon after the maneuver. It would take more time (24 hours) to see that the satellite has moved further from its expected location (figure 4.10). The 6, 3, and 1 hour plots are not shown because the maneuver is indistinguishable just as can be seen in figure 4.9.
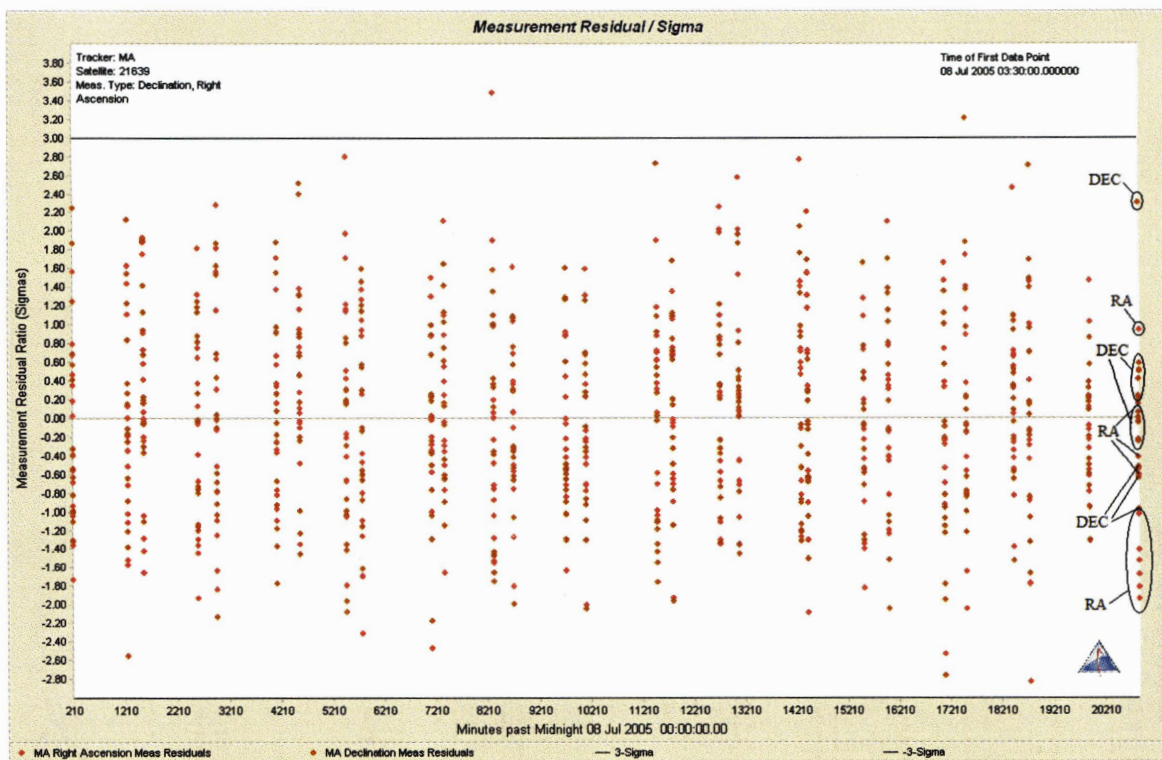
Figure 4.9. Δv = 0.01 m/s. Track 12 hours after maneuver.
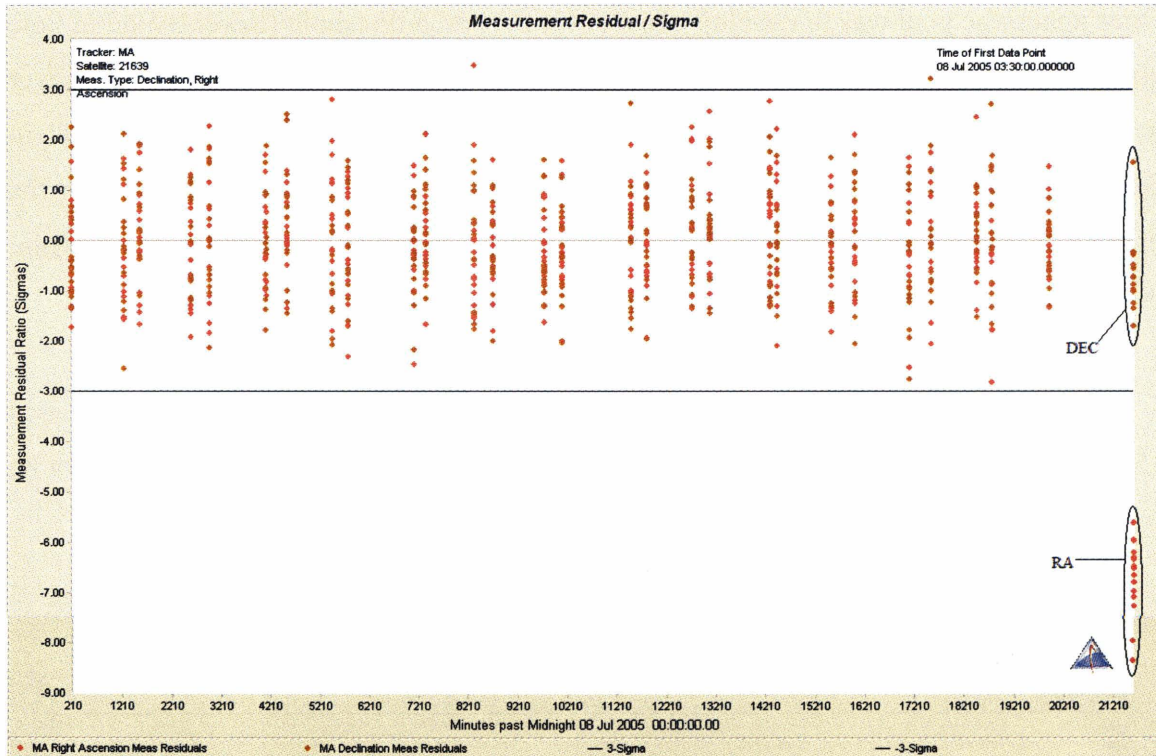
Figure 4.10. Δv = 0.01 m/s. Track 24 hours after maneuver.

As can be seen, the larger the maneuver and the more time allowed to elapse before a subsequent track, the easier it is to pick out the existence of the maneuver. As the maneuvers simulated are smaller (0.01 m/s), the residuals are still within the bounds of the 3 sigma covariance for each the 12, 6, 3, and 1 hour post maneuver tracks. However, upon closer review of the 12 hour case, one can see that the declination residuals and separately, the right ascension residuals have clumped together. This is evidence that something must have happened to the orbit. By 24 hours, however, even this small maneuver can be detected.

4.3. Maneuver Detection Threshold

The next maneuver detection process evaluated was to take a much smaller set of data (three tracks over the period of just one day) and then maneuver and see if detection was possible. The scenario here was a satellite undergoing a series of maneuvers over days where one may only have one day of tracking before the satellite maneuvers again.

Once again, the goal was to develop a maneuver detection threshold (lower bound in time after the maneuver) for each Δv that was tried. If one can detect a certain size maneuver in a certain time, then a larger maneuver or longer time period would also result in detection. The method used to simulate and filter the maneuver was the same as the previous two week cases, only the amount of tracking and time was smaller. Once again, the maneuver was simulated and the then filter was run without the maneuver.

For a Δv of 1.0 m/s, detection is very rapid. Figure 4.11, shows the residual ratios 15 minutes after the maneuver. It looks as if something has happened, but one cannot be sure.
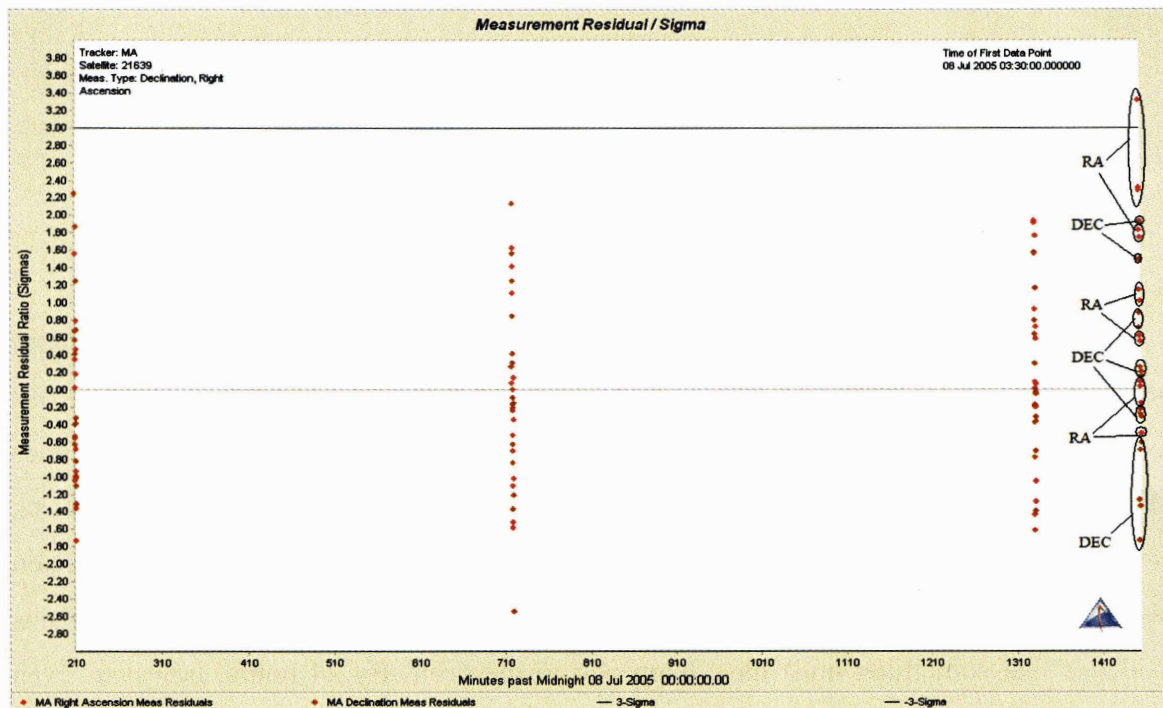
Figure 4.11. Δv = 1.0 m/s. Track 15 minutes after maneuver.

However, for figure 4.12, after 30 minutes it is clear that a maneuver has occurred. This shows that one could easily detect a 1.0 m/s or greater intrack maneuver for a GEO satellite within a half hour after it has occurred.

For the Δv of 0.1 m/s case, one would have a difficult time detecting prior to 12 hours after the maneuver. Figure 4.13 shows the 0.1 m/s intrack maneuver after 6 hours.

The maneuver is not easily detectable. Figure 4.14, however, shows that the 0.1 m/s maneuver is easily detectable after 12 hours.
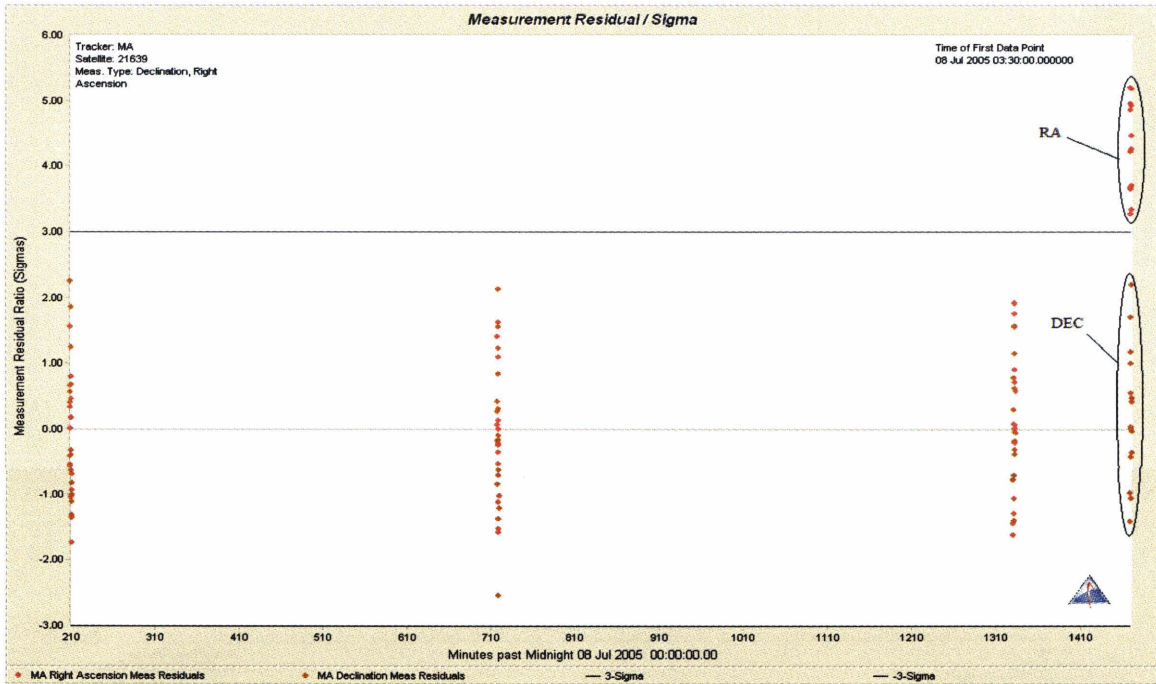
Figure 4.12. Δv = 1.0 m/s. Track 30 minutes after maneuver.
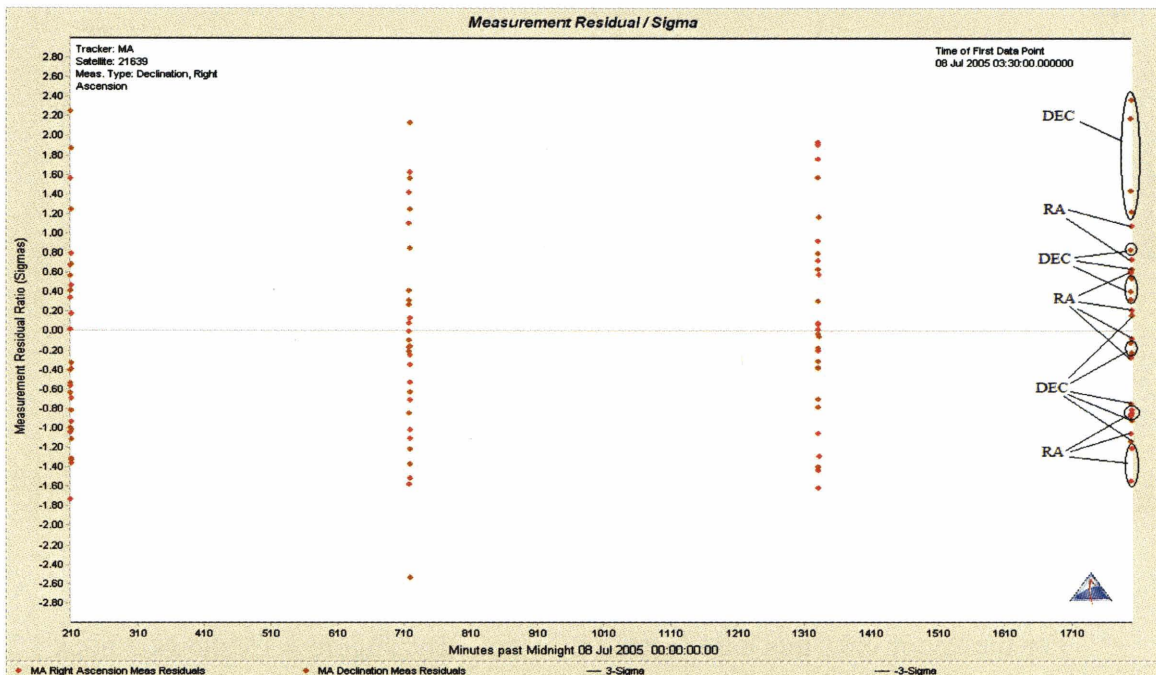
Figure 4.13. Δv = 0.1 m/s. Track 6 hours after maneuver.
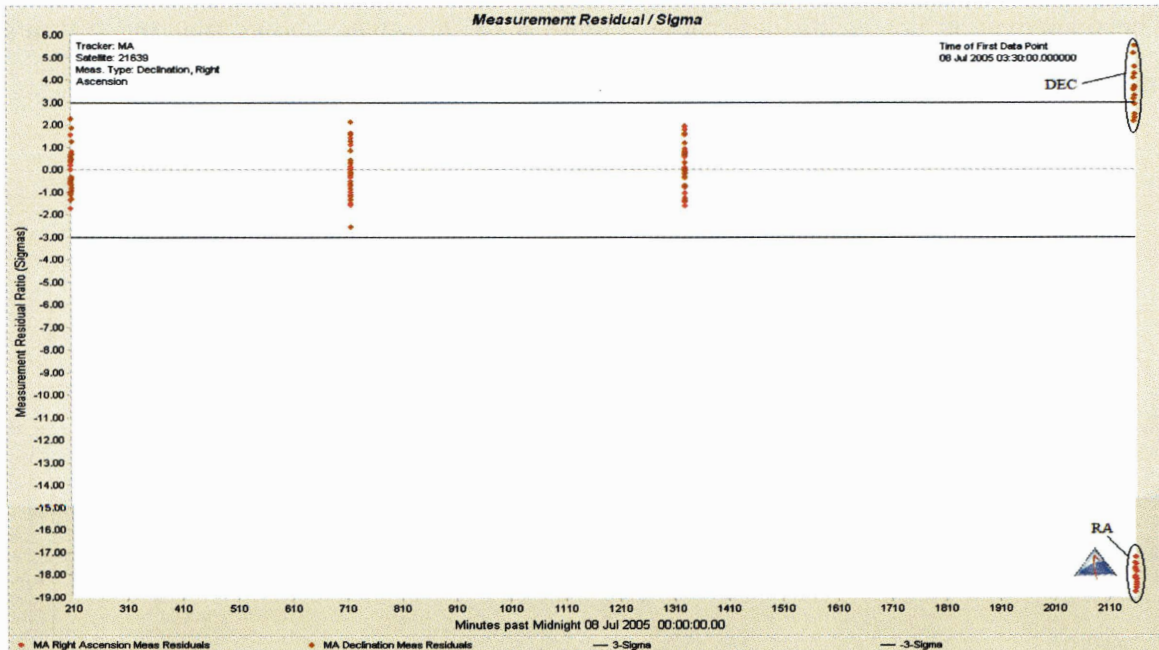
Figure 4.14. Δv = 0.1 m/s. Track 12 hours after maneuver.
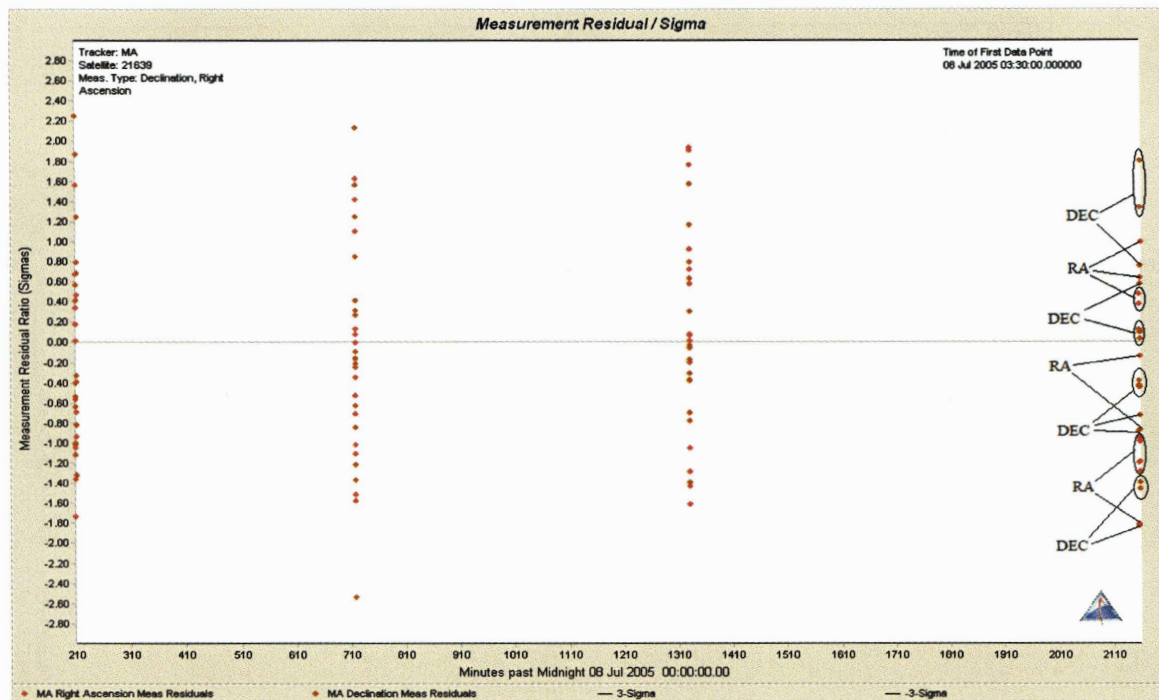
Figure 4.15. Δv = 0.01 m/s. Track 12 hours after maneuver.

For the Δv of 0.01 m/s case, detection is not easy. Figure 4.15 shows a track 12 hours after the maneuver and certainly no definite conclusion can be made. It just looks

like any other measurement. The maneuver is small, and it barely affects the orbit after such a short time period.

Figure 4.16 however, shows tracking at 12, 24, 36 and 48 hours after the 0.01 m/s maneuver. The last three tracks seem to show a trend of the something happening within the measurements. This, as we know, is the maneuver finally affecting the orbit. However, if we did not have such information (say in a real world situation), we may just interpret this data as poor data or that something may have happened. For example, we also tested to see if the track at 24 hours would better show the presence of the maneuver if the track at 12 hours was not present. It appeared that the lack of the 12 hour track did not significantly influence the ability of subsequent tracks to show the maneuver or not.
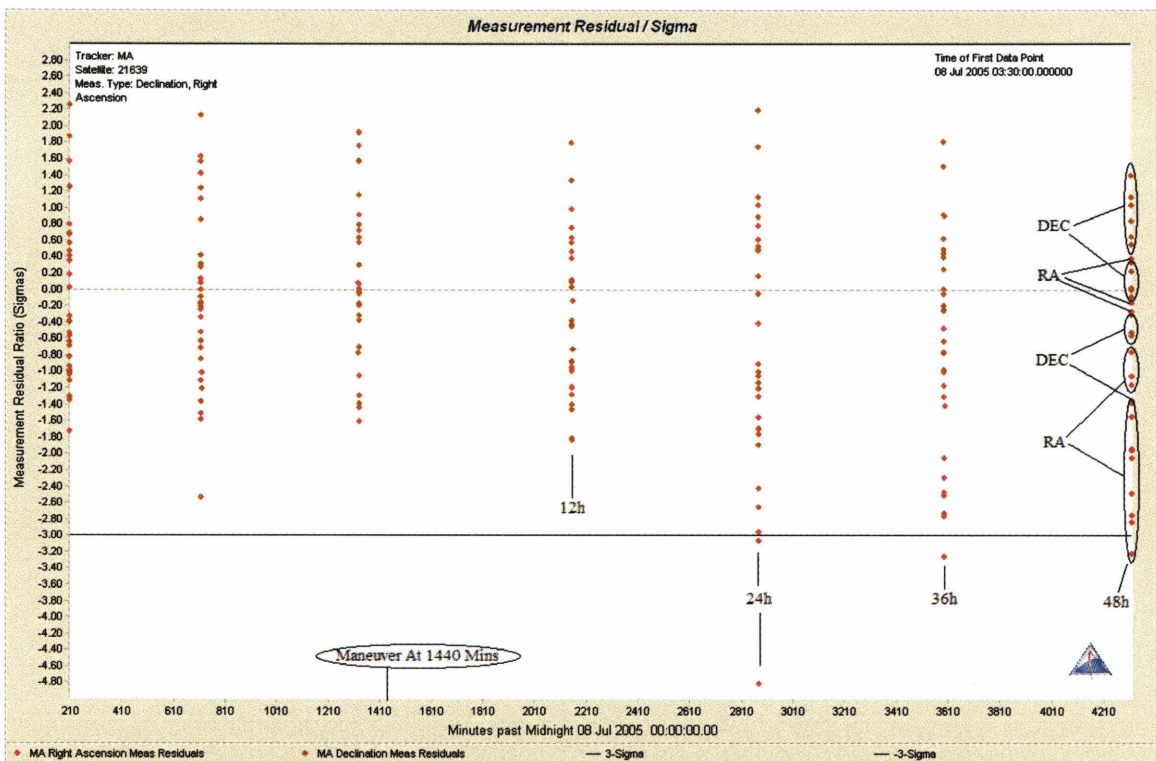
Figure 4.16. Δv = 0.01 m/s. Tracks 12, 24, 36 and 48 hours after maneuver.

A Δv of 0.01 m/s is so small that it almost looks as if the orbit has not really changed within two days. A lower bound threshold for such maneuvers will always be difficult to truly determine. What we do know now is that maneuvers of 0.1 m/s and greater can be easily detected within 12 hours, provided there is adequate tracking.

71

After a maneuver has been detected, the next step is orbit recovery. This is addressed in Chapter 5.

5. Post Maneuver Orbit Recovery

The purpose of this section is to demonstrate how quickly and accurately one can determine an orbit after a maneuver given either an optical sensor that can be dedicated to track for some time or just routine tracking data. Three methods will be examined. The first method asks whether the ongoing filter can just process through a maneuver without any information of the maneuver occurring. The second method will just involve initial orbit determination after the maneuver with no previous knowledge of the maneuver. The third method will involve initial orbit determination followed by a batch least squares and then followed by an attempt to determine maneuver time. Then, after the maneuver time has been determined, the filter orbit will model the maneuver and proceed through it. Finally, a real data case will be attempted for the TDRS 5 satellite that was used in chapter 3.

5.1. Filter Through a Maneuver (Brute Force Method)

In keeping with having a continually running filter (in the sense of using previous tracking data), the first way of attempting to maintain an accurate orbit after a maneuver is by forcing the filter to accept all measurements before and after the satellite has maneuvered. The method as illustrated in the following was suggested and formulated by Richard Hujsak of AGI [Ref 13, 16]. For this, a two week fit, with a tracking density of two tracks per day was simulated. A 1.0 m/s maneuver occurs and 4 tracks spaced 12 hours apart follow the maneuver (essentially trying this method with a fairly relaxed scenario as far as quick orbit recovery is concerned). The filter is provided with the measurements, but no knowledge of the maneuver itself. After the maneuver is detected, as in chapter 4, the filter is run again. This time it is provided with maneuver process noise given only between the last good track (before the maneuver) and the first bad track (after the maneuver). With this process noise, the filter is being told about an active process that increases the covariance. Ordinarily, the filter will reject measurements that are beyond the 3-σ covariance acceptance level (as those after the maneuver are). If the filter is set to accept measurements to say, the 50000-σ covariance level, then for a

reasonably sized maneuver (e.g. 1.0 m/s), all the measurements should be accepted. This σ, 50000-σ, was chosen in order to ensure that no data would be thrown out of the orbit fit. This σ can be any large number that will allow the fit to accept the data (e.g. for this case 1000-σ would even be acceptable as all the data fits within that σ level, see figure 5.1). It will not affect the end result. One fear though, of this large a priori covariance, is that bad measurements could be used.

In practice, the filter is run to the observation of the last track before those tracks that indicated a maneuver. A restart record is created which is used to restart the filter with the maneuver process noise to run the filter to the post maneuver track that showed the maneuver. The maneuver process noise given is set to 0.6 cm/s for all RIC components in *Satellite>ForceModel>UnmodeledAccelerations>ProcessNoise.* The process noise update interval is set to 1 min [Ref 13]. The maneuver process noise is then turned off. A residual ratios plot is first shown in figure 5.1 where no maneuver process noise model had been added. Clearly a maneuver has occurred. In figure 5.2, after the maneuver process noise was added, the residual ratios are accepted by the 3-σ covariance.
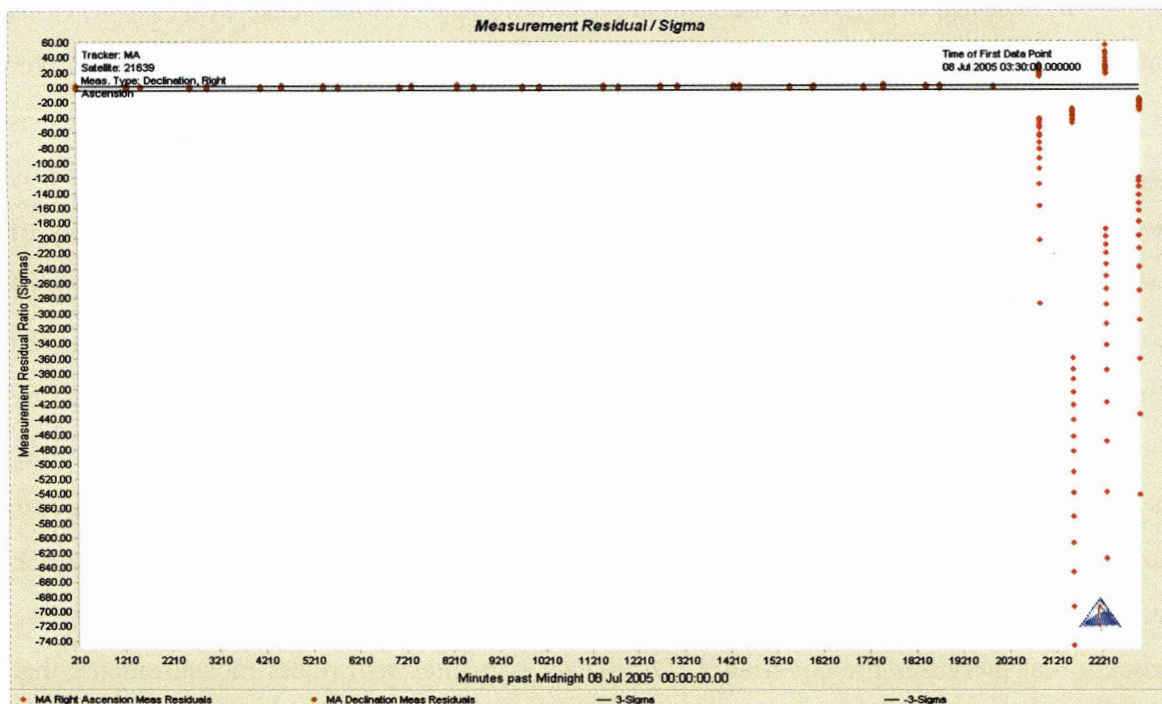
Figure 5.1. Residual ratios. Maneuver detected.
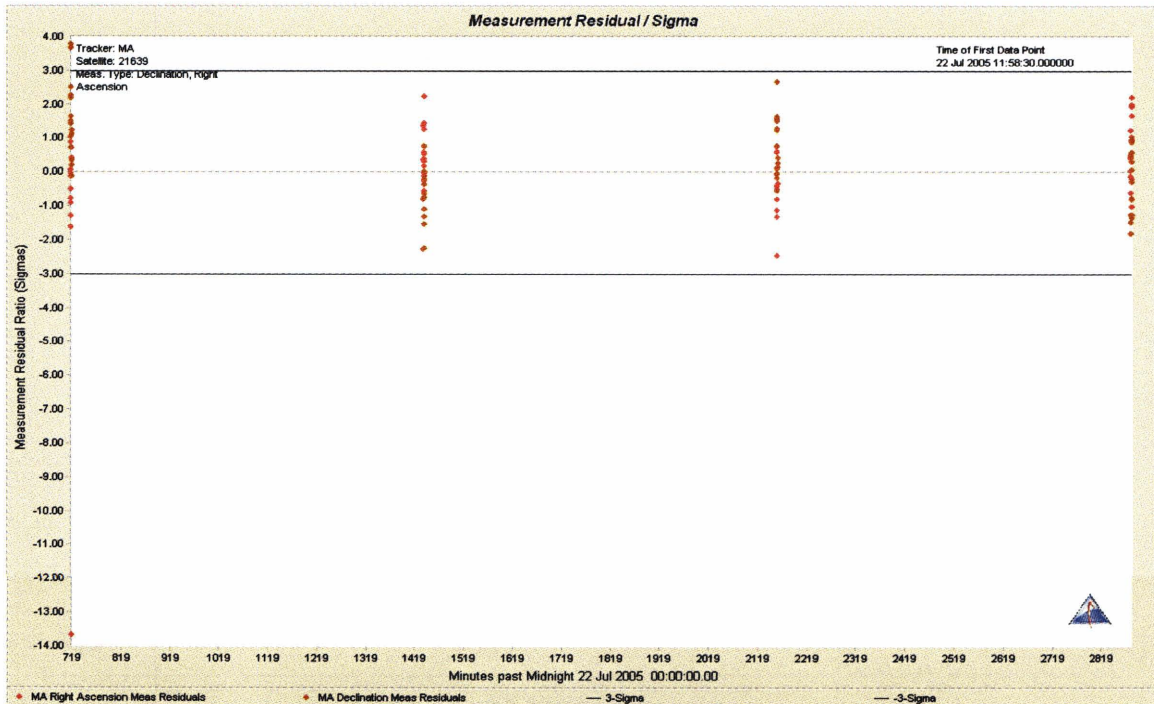
Figure 5.2. Residual ratios. Post maneuver residuals are accepted.

Since the measurements have now been incorporated into the orbit fit, the calculated covariance is of interest (figure 5.3). After the four post maneuver measurements, the covariance indicates a 2-σ confidence level that the orbit error is approximately 2 km intrack and 1 km in the radial direction. But the question remains as to whether this covariance is optimistic or pessimistic.

By checking the simulated truth against the results of the filter fit (figure 5.4) it is seen that the covariance from figure 5.3 is too optimistic. Figure 5.4 shows the position differences between the simulated truth and the filter run after the maneuver to be as much as 5 km in the radial direction.

One can continue exploring this method for future work. The post maneuver smoothed orbit could be predicted backwards and compared with the pre-maneuver orbit with their intersection defining the time of Δv of the maneuver. Then the process could be iterated now knowing more exactly when to add the process noise and how large to make it.

75
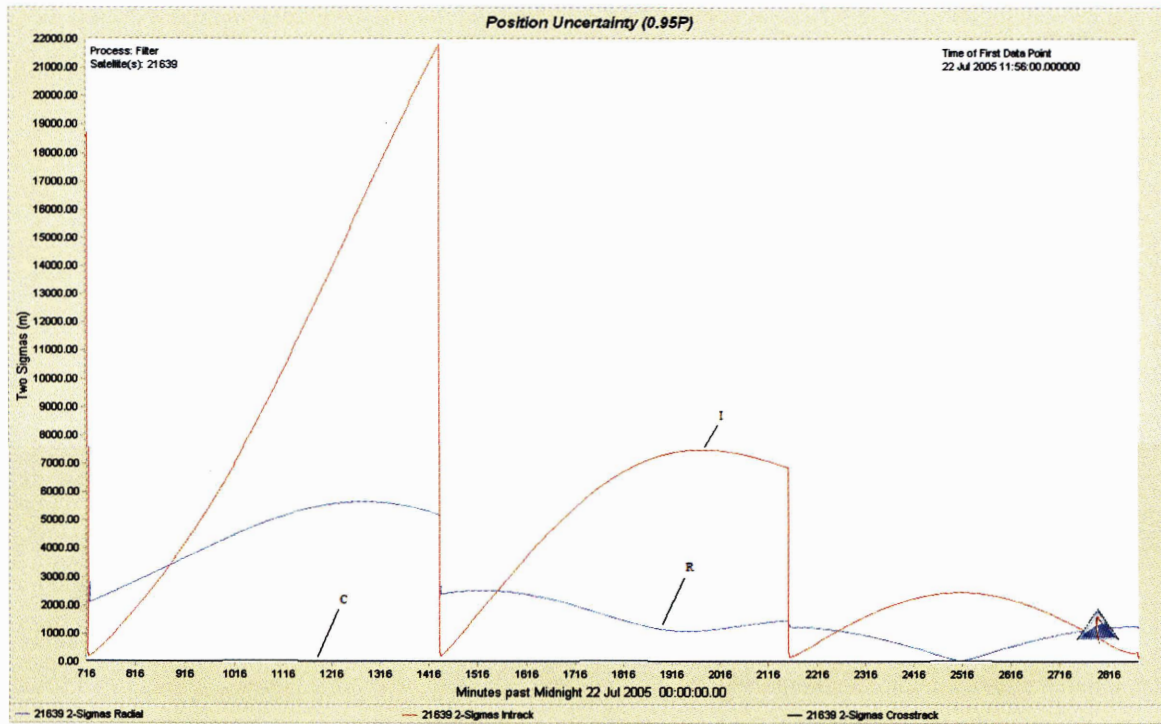
Figure 5.3. Position uncertainty post maneuver.
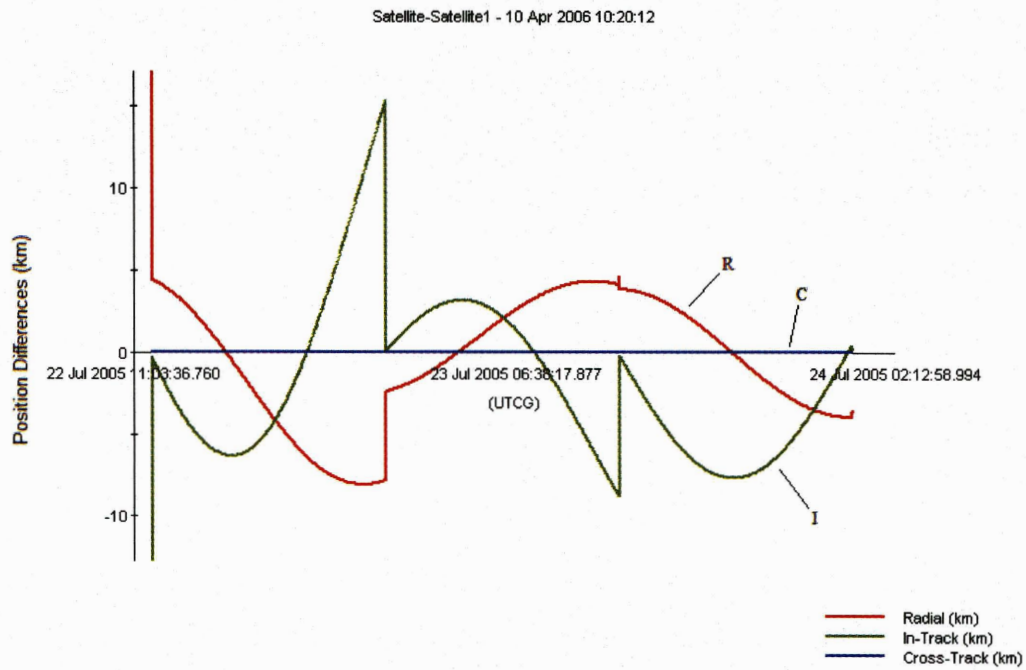
Figure 5.4. Simulated truth differenced from filter run post maneuver (km).

5.2. Initial Orbit Determination (IOD) and Post Maneuver Orbit Recovery

The second and third methods require initial orbit determination to help determine the post maneuver orbit. Initial orbit determination in ODTK requires three observations that are adequately spaced over a period of time. Optical tracking requires more spacing than RADAR because the range resolution is more difficult to determine. For this study, the observations used are angles only measurements (right ascension and declination). The initial orbit determination (IOD) method that is available with ODTK is the Gooding Angles Only IOD method. This method uses two position vector estimates, r_1 and r_3, and their associated times, t_1 and t_3. These positions and times are applied to the Lambert problem to find an estimate of the velocity vector, v_1. Since an estimate of r_1 and v_1 are known at a certain time, one can propagate to the desired position and velocity r_2 and v_2 at t_2. A number of iterations may be made. This process is further detailed in Wright, 2004 [Ref 4].

For this study, we parametrically spread out three or more optical tracks over time periods ranging from 3 to 12 hours. After performing the Gooding Angles Only IOD on three measurements (each measurement from a different track in order to provide adequate spacing), a least squares fit was performed on all the tracking data to get an orbit. The resultant least squares orbit was compared to the expected orbit found from propagating the satellite to the same epoch as the least squares epoch. Table 5.1 shows the expected truth orbits (propagation) for a satellite after maneuvering with a Δv of 1.0 or 0.1 m/s. The orbital elements included in tables 5.1-5.3 are: semimajor axis (a), eccentricity (e), inclination (i), right ascension of ascending node (Ω), argument of perigee (ω), and true argument of latitude (u).

Truth Orbits	a (km)	e	i (°)	Ω (°)	ω (°)	u (°)
Δv = 1.0 (m/s)	42191.9	0.000646	6.476	71.610	325.360	319.000
Δv = 0.1 (m/s)	42167.2	0.000236	6.476	71.610	30.081	319.000

Table 5.1. Expected truth orbits after maneuvers.

Tables 5.2 and 5.3 show the number of tracks, the time elapsed, the size of the maneuver and the least squares solution after IOD was performed on existing tracking.

# of Tracks	Time Elapsed (h)	Δv (m/s)	a (km)	e	i (°)	Ω (°)	ω (°)	u (°)
3	3	1.0	42116.4	0.000735	6.474	71.599	165.858	319.000
3	6	1.0	42171.8	0.000438	6.476	71.606	291.316	319.000
3	9	1.0	42183.2	0.000612	6.476	71.607	309.981	319.000
3	12	1.0	42187.6	0.000669	6.476	71.608	317.618	318.995
5	5	1.0	42183.7	0.00051	6.476	71.608	319.021	319.001
6	6	1.0	42191.8	0.000646	6.476	71.610	324.965	318.999

Table 5.2. Least squares solutions after a Δv = 1.0 m/s.

# of Tracks	Time Elapsed (h)	Δv (m/s)	a (km)	e	i (°)	Ω (°)	ω (°)	u (°)
3	3	0.1	42091.9	0.001232	6.474	71.599	147.390	319.000
3	6	0.1	42147.1	0.000184	6.476	71.606	155.057	319.000
3	9	0.1	42158.5	0.000068	6.476	71.607	14.2907	319.000
3	12	0.1	42163	0.000172	6.476	71.608	11.6397	318.999
6	6	0.1	42167.1	0.000232	6.476	71.610	29.625	318.999

Table 5.3. Least squares solutions after a Δv = 0.1 m/s.

Tables 5.2 and 5.3 indicate the amount of tracking necessary to achieve orbits that resemble those in table 5.1. Depending on the level of accuracy one desires to determine an orbit after a maneuver, the number a tracks and time needed may be inferred.

The Δv = 0.01 m/s case was not included in this study based on the fact that the maneuver is difficult to detect within a 48 hour time frame (see chapter 4). Since the data shows that the orbit is only slightly affected to this point, one could continue to track the satellite and recover a good orbit as more data is incorporated into the fit. Solving for the maneuver at this point is not necessary.

78

5.3. Shotgun Method

This section examines the third approach to post maneuver orbit recovery. This method begins similarly to the second method by first performing an IOD with three measurements followed by a least squares using all the remaining post maneuver data. Then, it tries to correctly predict the maneuver time so that the data and filter information before the maneuver can be used. As will be explained below, the method is called the "shotgun" approach and was suggested by Richard Hujsak [Ref 13].

To implement the "shotgun" method the filter is run with two weeks of simulated observations of normal tracking density (2 tracks per day), followed by a maneuver ($\Delta v =$ 1.0 m/s in the intrack direction on 21 Jul 2005 16:30:00.000) and six tracks within six hours after the maneuver. Six tracks in six hours after the maneuver was the necessary amount of tracking to arrive at a good IOD and least squares solution in order to proceed. This was determined from Tables 5.1-5.3.
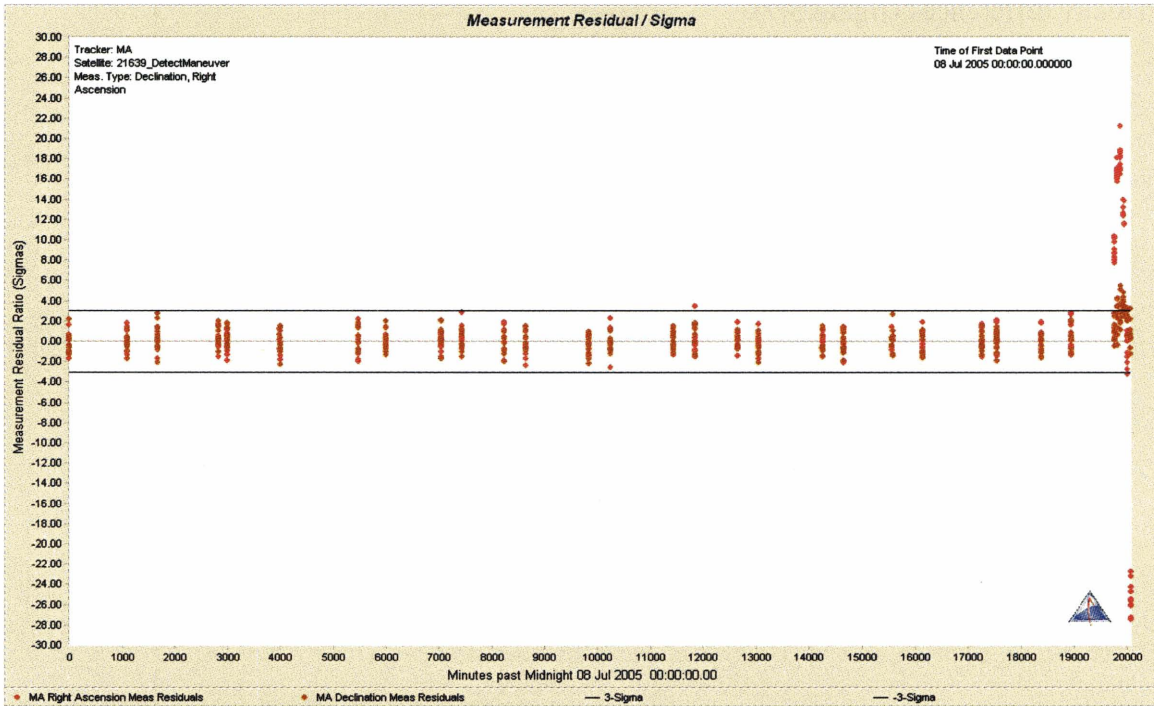
Figure 5.5. Residual ratios plot. $\Delta v = 1.0$ m/s followed by 6 optical tracks in 6 hours.

79

The observations were simulated and the filter was run without knowledge of the maneuver. Figure 5.5 first shows the resulting residual ratios plot. It is clear that a maneuver has happened.

Now that a maneuver has clearly happened, the goal is to know exactly when the maneuver happened so we can continue the filter through the maneuver and see if we can quickly recover an accurate orbit. First an IOD is computed using three observations from within the tracks that are obtained after the maneuver. The IOD is followed by a least squares orbit solution involving all the measurements from the subsequent tracks after the maneuver. The least squares orbit solution is then propagated and compared to (differenced from) the filter orbit solution developed previously. This comparison is done in Satellite Tool Kit (STK) by inputting the least squares solution propagation ephemeris into a satellite (*Scenario1>Satellite1*) and the filter orbit solution ephemeris into another satellite (*Scenario1>Satellite2*). The two satellites can be differenced and graphed. The relevant plots are comparing the RIC position differences (figure 5.6) and the RIC velocity differences (figure 5.7).

Figure 5.6. RIC position differences (km) used to estimate the time of the maneuver.

80
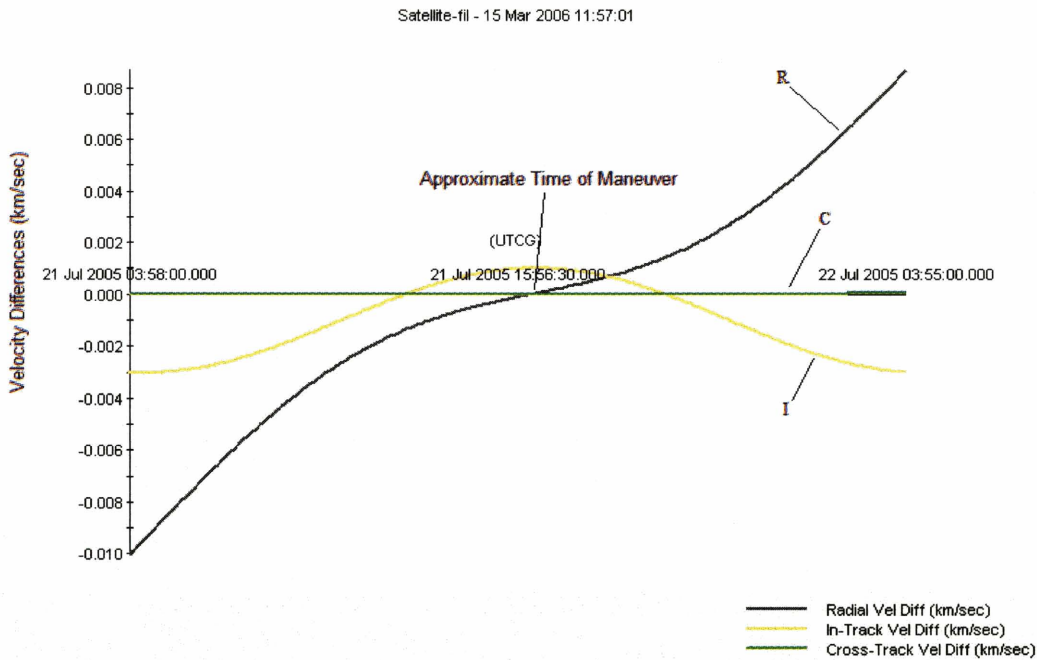
Figure 5.7. RIC velocity differences (km/sec) used to estimate the time of the maneuver.

Figure 5.6 shows that the two ephemeredes are closest together between approximately 21 Jul 2005 16:00:00.000 and 21 Jul 2005 17:00:00.000. Since the only time the two ephemeredes should have in common is the time of the maneuver, this shows that the maneuver must have happened within this hour.

Now that the approximate time of the maneuver is known, the "shotgun" method can be applied. In order to "shotgun" the maneuver for the hour in question, the filter is run with 10 possible maneuver covariances ($\Delta\sigma v_r$, $\Delta\sigma v_i$, and $\Delta\sigma v_c$) during the hour. The ODTK has a utility called "*InstantManeuverSet.htm*" (on the ODTK Start Page) that permits the user to do this easily. Note that the components of the Δv (Δv_r, Δv_i, and Δv_c) are set to zero, but the maneuver covariances are set to roughly the magnitude of the expected maneuver. Hujsak [Ref 13] suggested as a rule of thumb to look at the first track following the maneuver. If the maneuver covariances are too large, and data are being rejected, try smaller values for the maneuver uncertainties. Presumably, one of the sprinkled ("shotgunned") maneuver hypotheses will be close to the actual maneuver. This does not actually instruct the filter that the satellite is maneuvering at a certain time, but if the satellite does maneuver, the filter will be able to adapt and include the measurements.

81

This allows the filter to refine the orbit through each subsequent measurement after the maneuver to continue generating an accurate orbit solution. This results in the residual ratios plot in figure 5.8.
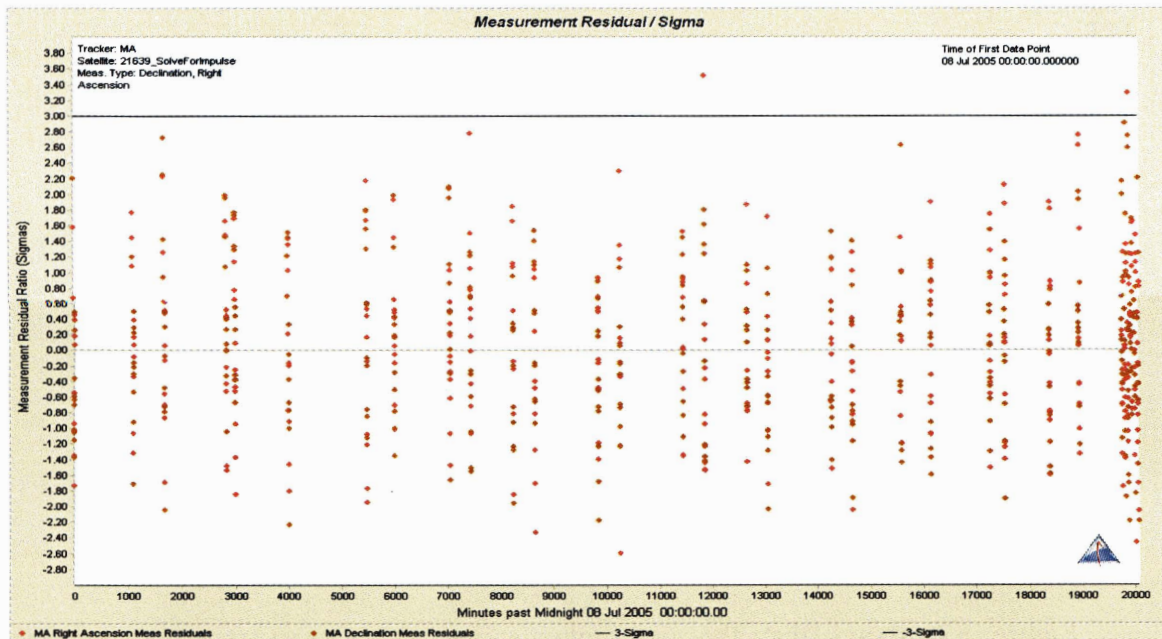
Figure 5.8. Residual ratios plot for "shotgun" method.
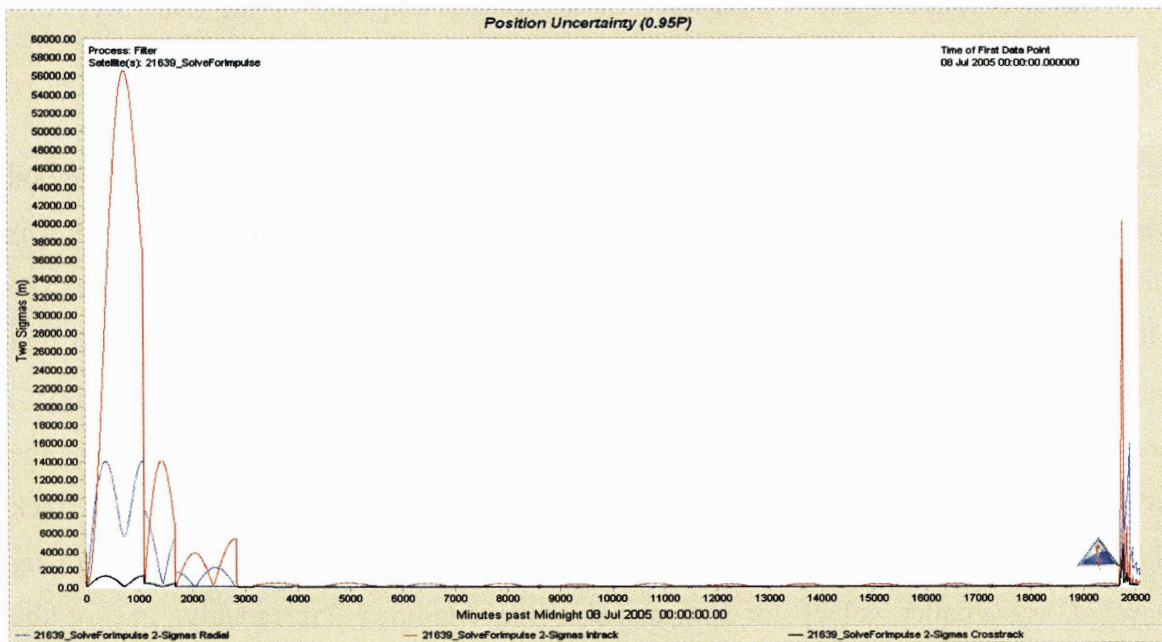
Figure 5.9. Position uncertainty plot. Sharp jump at right is the maneuver.
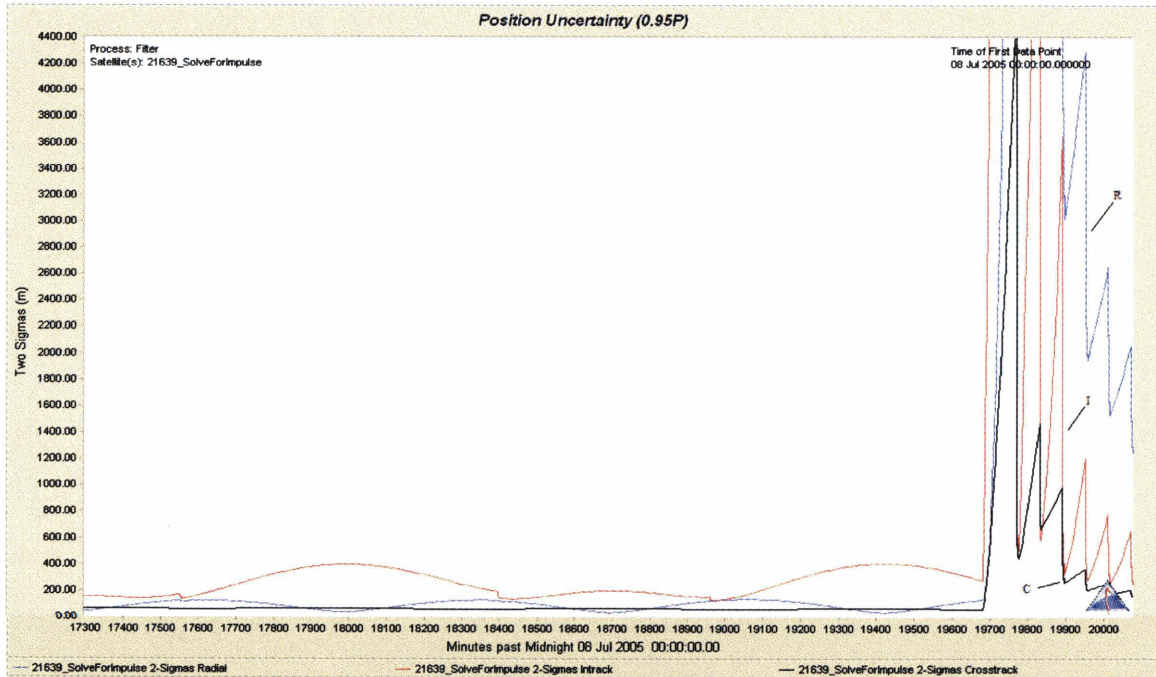
Figure 5.10. Position uncertainty plot zoomed in on maneuver.

The effects of the maneuver can also be seen in the position uncertainty plot. This plot (figure 5.9) shows that the orbit had a good solution until something happened, and then the orbit was recovered. A closer view of the maneuver and recovery is also seen in figure 5.10.

Here (figure 5.10) it is seen that the post maneuver orbit 2-σ error is approximately 1.5 km in the radial direction, 0.4 km for the intrack direction and 0.2 km for the crosstrack direction. The intrack and crosstrack errors are nearly at their pre-maneuver values. The radial error is still coming down. This is within 6 hours of the maneuver. Figure 5.11 is the simulated truth ephemeris differenced from the filter ephemeris for the duration of the fit span. Figure 5.12 is the same fit, only the last few days have been enlarged in order to see the effect of the maneuver. As can be seen by examining the comparable figures (figures 5.9 with 5.11 and figures 5.10 and 5.12) the covariances are similar to the differences.
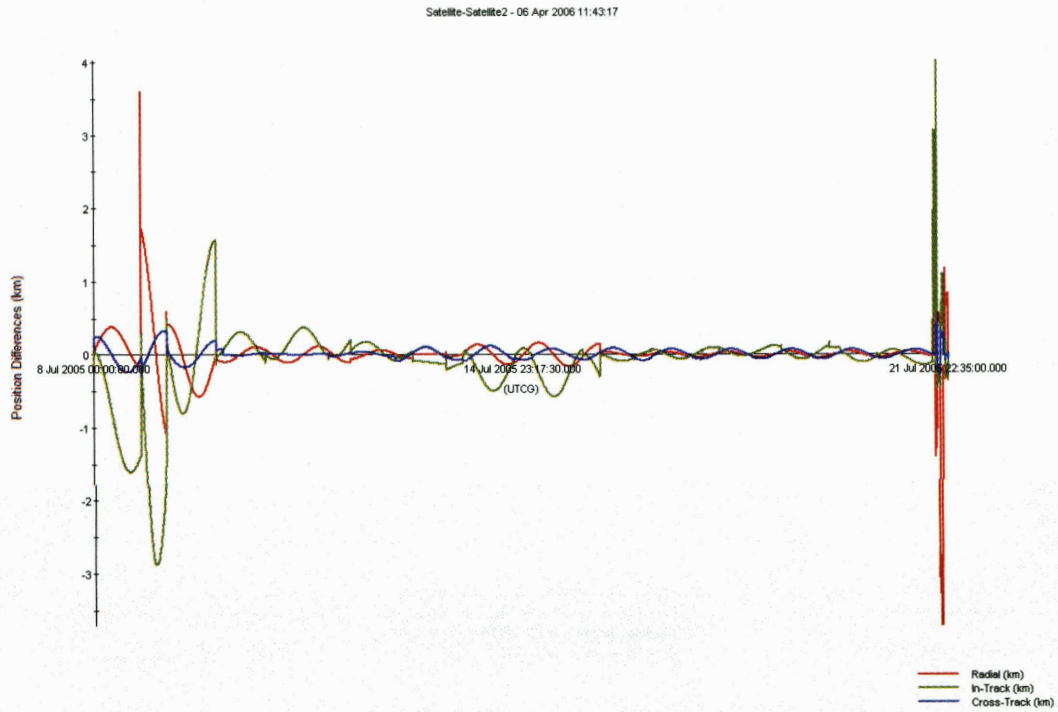
Figure 5.11. Simulated truth ephemeris differenced from filter ephemeris for entire fit span (km).
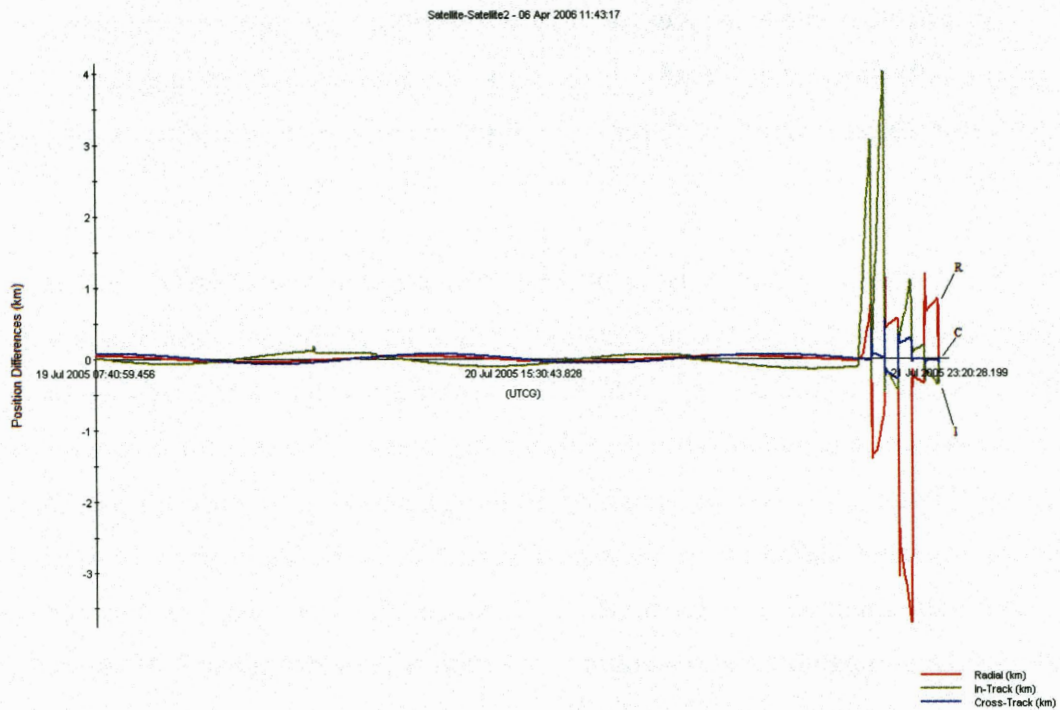
Figure 5.12. Simulated truth ephemeris differenced from filter ephemeris zoomed in on last few days of fit span (km).

Another example of the "shotgun" method is to take a much smaller set of tracking data (3 tracks in one day) again considering the case of a satellite that is constantly maneuvering with just 3 tracks since the previous maneuver. With just 3 tracks, the covariance prior to the maneuver may not be as good as the previous example with more tracking data. The maneuver ($\Delta v = 1.0$ m/s in the intrack direction on 09 Jul 2005 06:00:00.000) was followed by six tracks in six hours and we aim to see if the orbit can be recovered as swiftly. The method is the same, the measurements were simulated with a maneuver and the filter was run without knowledge of the maneuver. The resultant residual ratios plot is shown in figure 5.13.
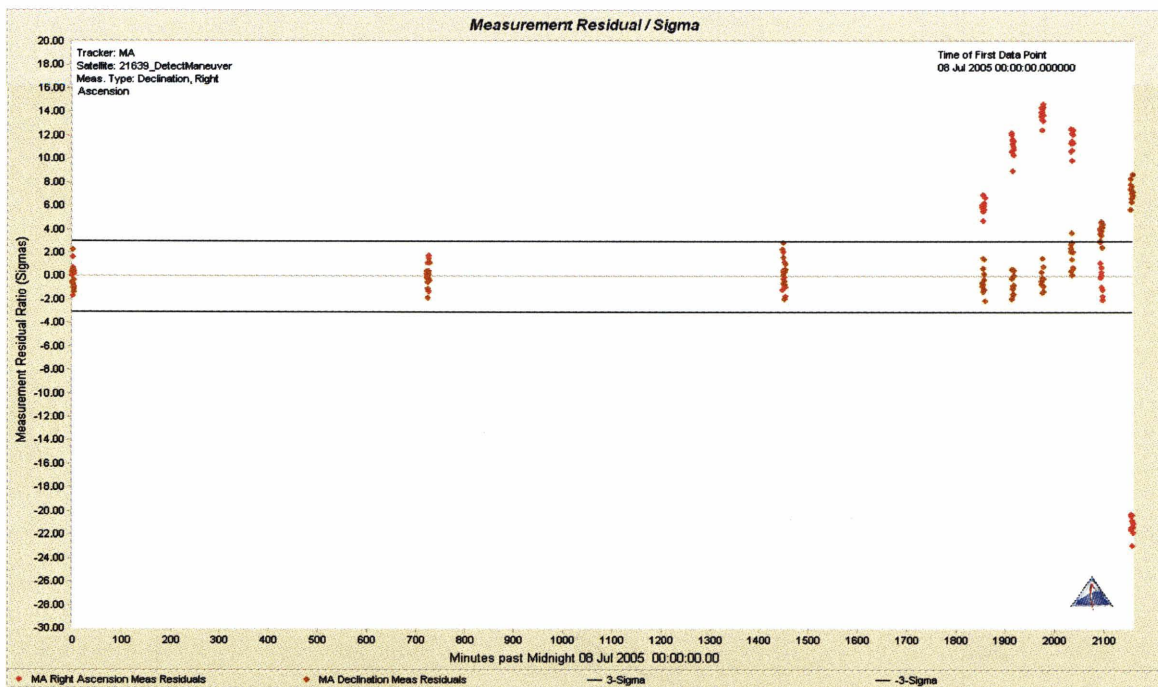
Figure 5.13. Residual ratios plot. $\Delta v = 1.0$ m/s followed by 6 optical tracks in 6 hours.

The maneuver can be easily seen. As before, one must compute an IOD and least squares from the measurements after the maneuver. The least squares orbit solution is propagated and the ephemeris is compared to the ephemeris of the filter run. The results are graphed in STK. Figures 5.14 and 5.15 are the result.
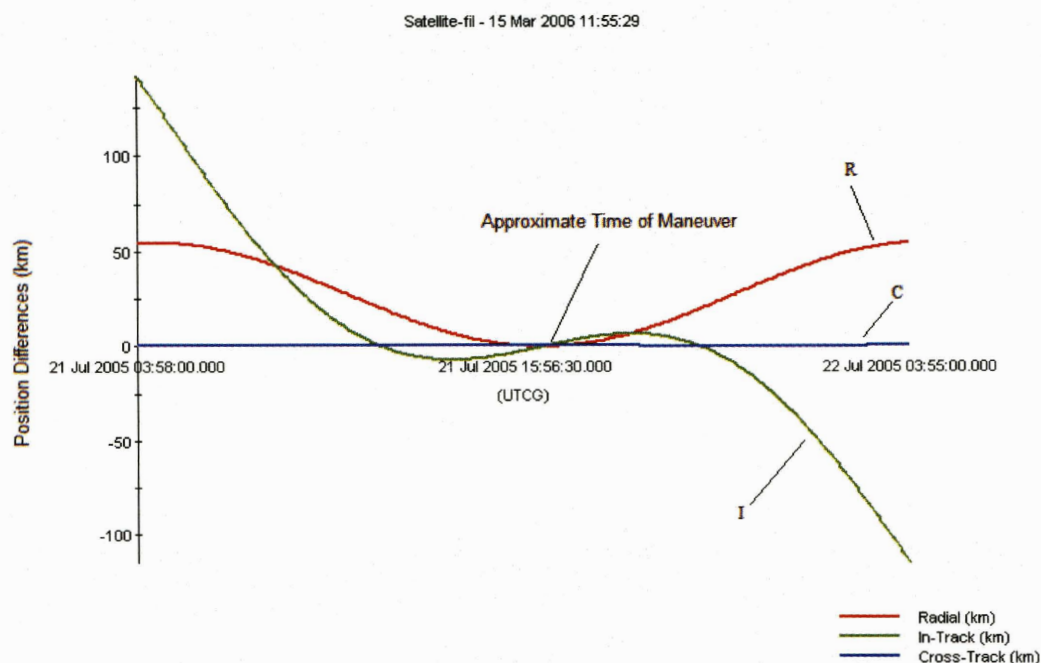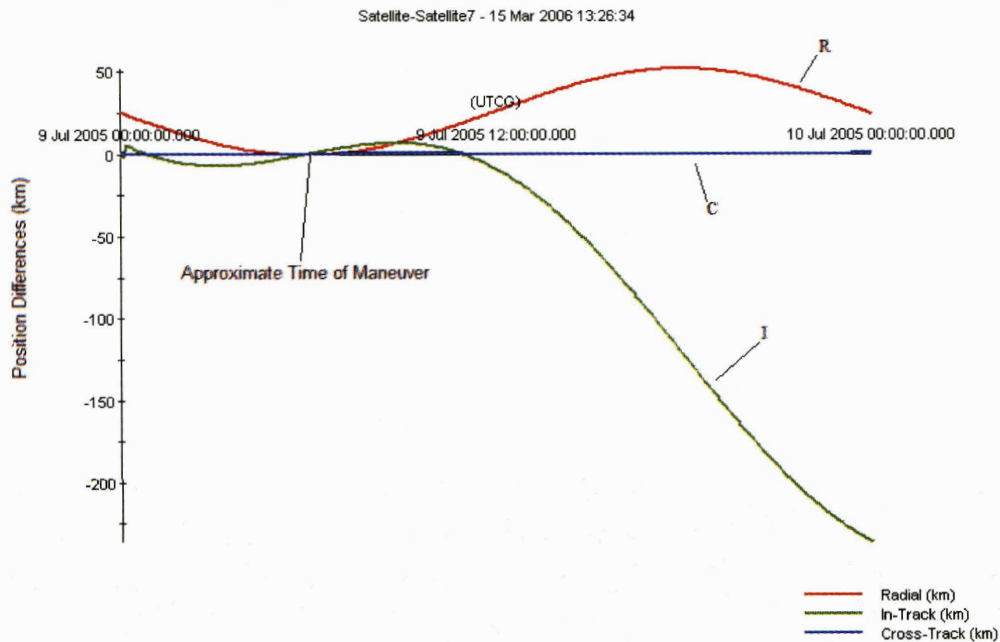
Figure 5.14. RIC position differences (km) used to estimate the time of the maneuver.
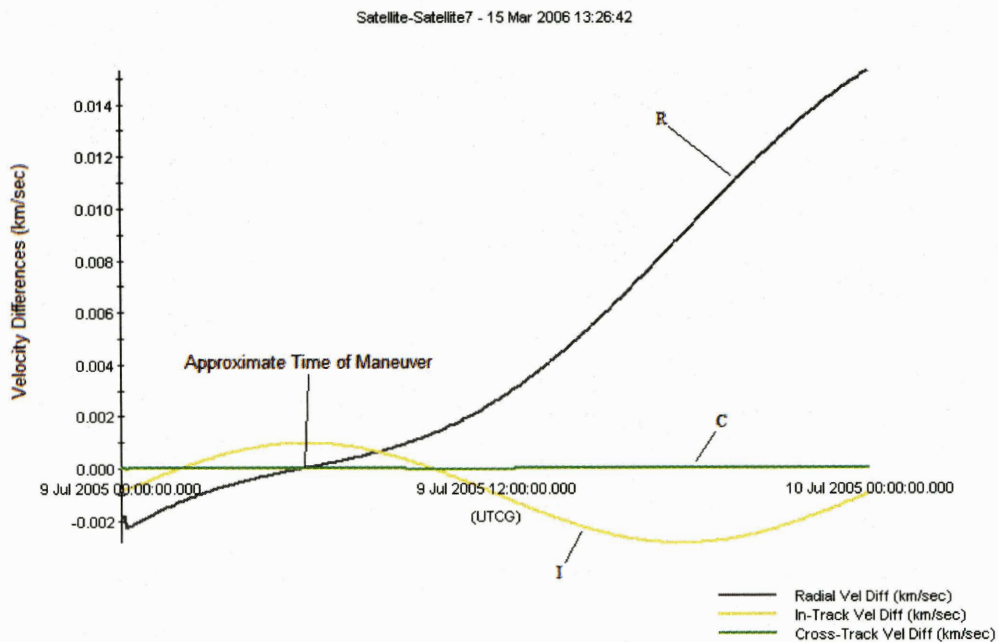
Figure 5.15. RIC velocity differences (km/sec) used to estimate the time of the maneuver.

As before, an approximate time of maneuver can be determined by finding the range of smallest differences in the RIC position and rate differences (figures 5.14 and 5.15). For this case, this time appears to be between 09 Jul 2005 05:30:00.000 and 09 Jul 2005 06:30:00.000. One must create the 10 maneuver covariances and run the filter.

Figure 5.16 is the resulting "shotgun" residual ratios plot. Figure 5.17 is the resulting position uncertainty plot that shows the effect of the maneuver.
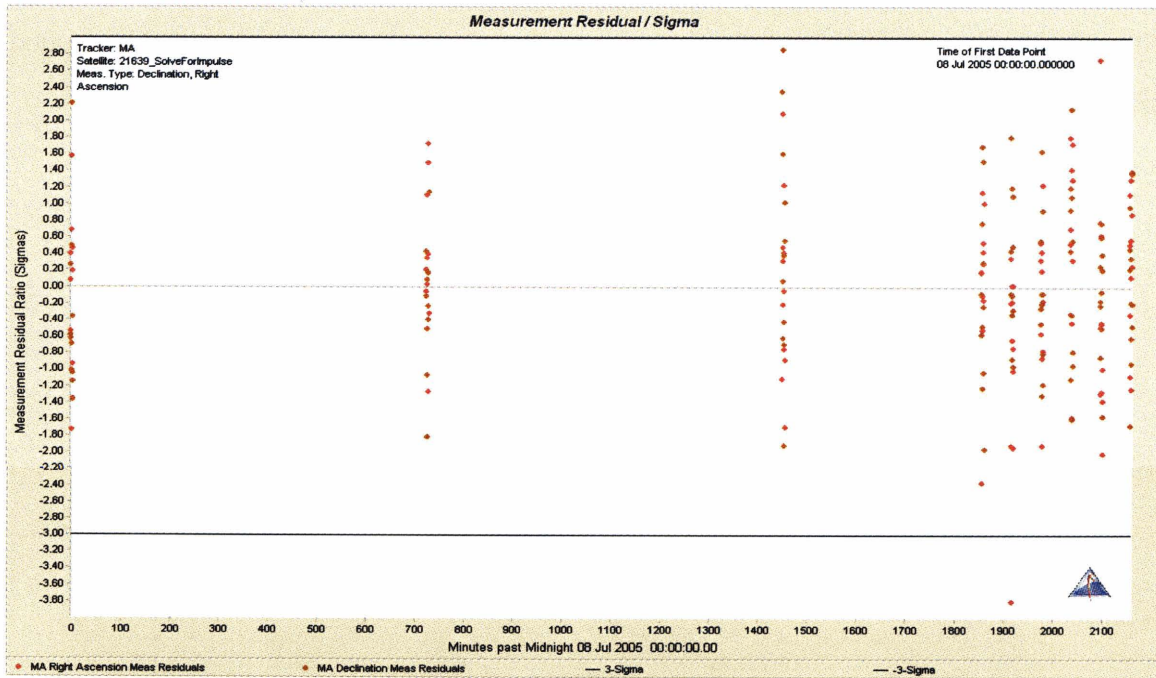
Figure 5.16. Residual ratios plot for "shotgun" method.
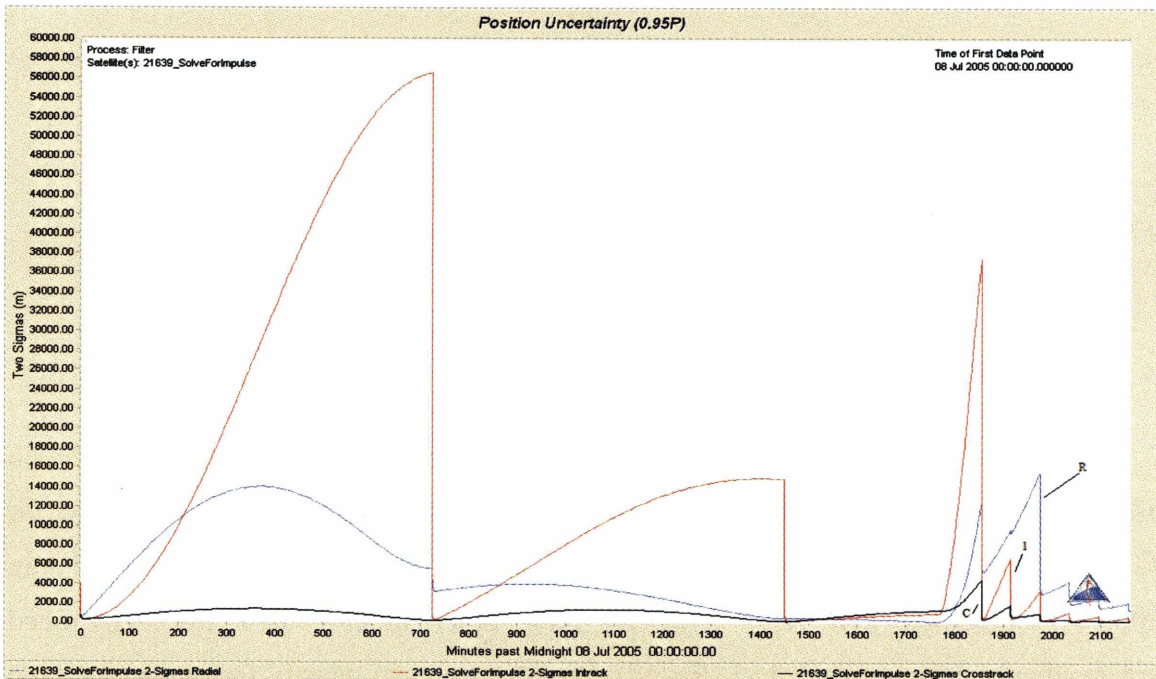
Figure 5.17. Position uncertainty plot. Sharp jump at 1800 minutes is the maneuver.

From figure 5.17, we see that after the six tracks within six hours the radial error is approximately 2.0 km, the intrack error is approximately 0.5 km and the crosstrack error is approximately 0.5 km. Once again, we examine the difference between the simulated truth ephemeris and the filter ephemeris (figure 5.18). This again shows good agreement indicating that the covariance is realistic after the 6 post maneuver tracks.
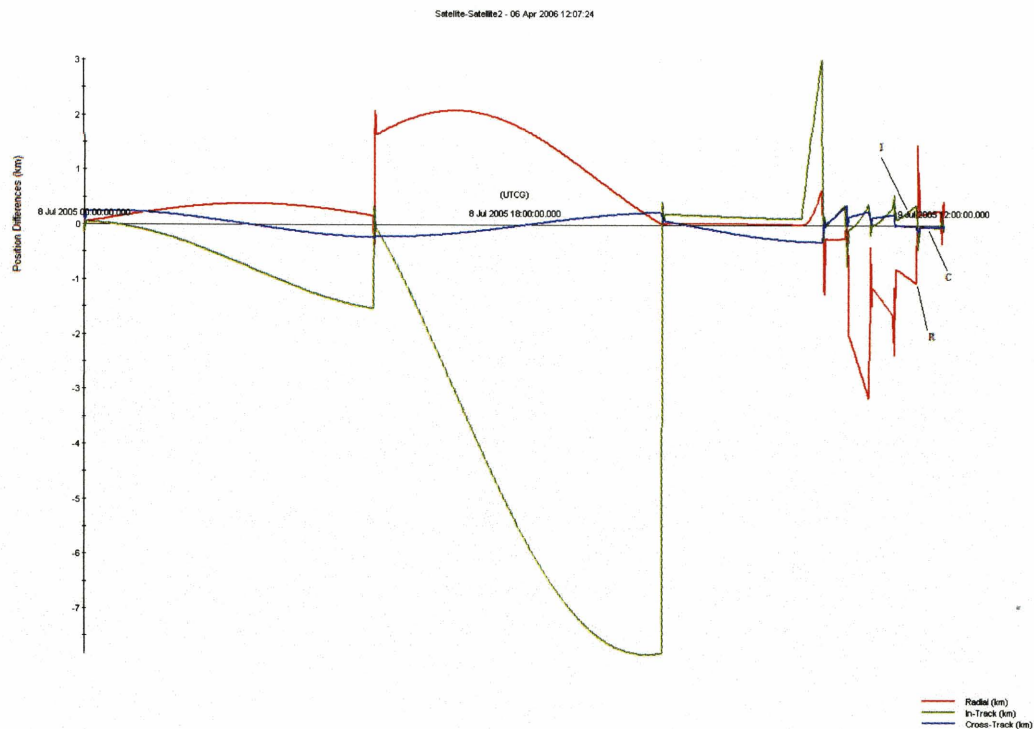
Figure 5.18. Simulated truth ephemeris differenced from filter ephemeris for entire fit span (km).

5.4. A TDRS 5 Real Data Case

Previously, all attempts to detect and process through a maneuver, then recover an orbit were done with simulated data. The following are the results for a real data case for TDRS 5. TDRS 5 is known to have maneuvered at some point between (day 237) 25 Aug 2005 14:00:00.000 and (day 238) 26 Aug 2005 06:00:00.000 (as determined independently with tracking data [Ref 9]). We had a typical density for optical tracking for TDRS 5 from day 224 to day 240 of 2005. This maneuver was ideal to examine because the tracking stations fortunately had 3 tracks on 26 Aug 2005 from 06:00:00.000

to 08:00:00.000 (after the maneuver) in addition to 5 tracks on day 239. The first 3 tracks provided enough information to perform an IOD, but not enough information to get a good least squares solution. The total of 8 tracks after the maneuver were required to provide enough information to get a good least squares solution. The least squares solution propagation was compared to the filter ephemeris from fitting the good data (the pre-maneuver data) and resulted in figure 5.19.
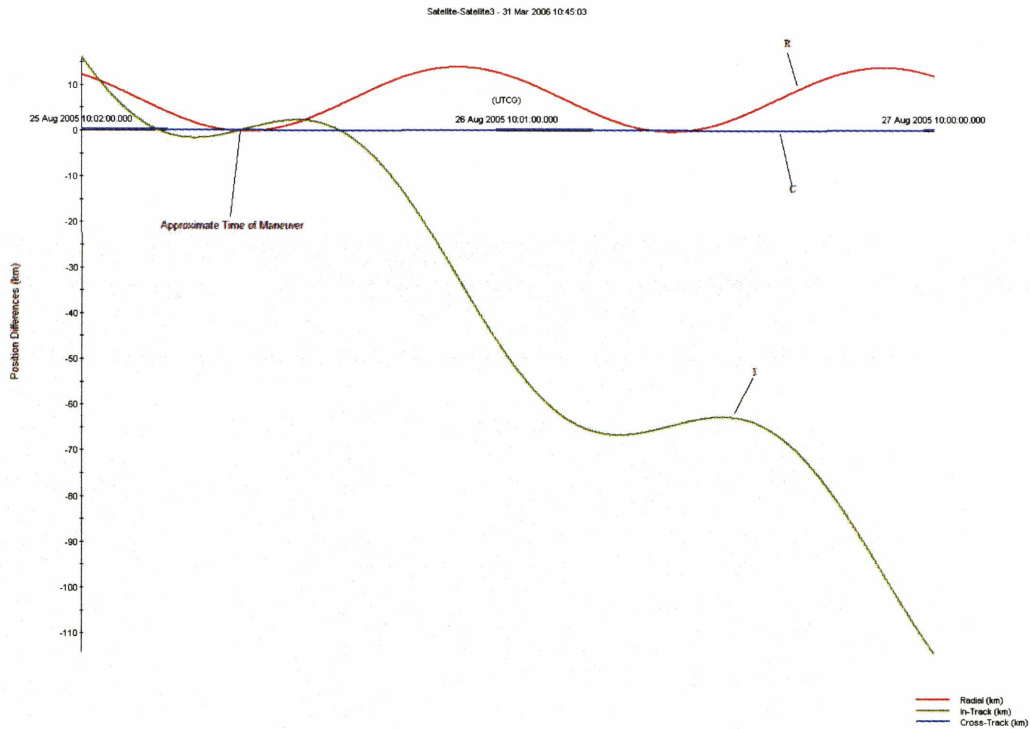
Figure 5.19. Least squares solution vs. filter run for real data case.

This plot (figure 5.19) shows that the approximate time when the filter and the least squares solution difference was closest to zero is the hour around 25 Aug 2005 19:00:00.000. This seems to be the best guess for the maneuver. Since we do not know the actual time of the maneuver, there is no way to absolutely check this assumption. Therefore, in ODTK, a set of 10 shotgun maneuver covariances of typical magnitude were input from 25 Aug 2005 18:30:00.000 to 25 Aug 2005 19:30:00.000. By running the filter with the shotgun maneuvers, the result was the residual ratios plot shown in figure 5.20. As can be seen, the shotgun maneuver covariances have fit the maneuver into the filter run.
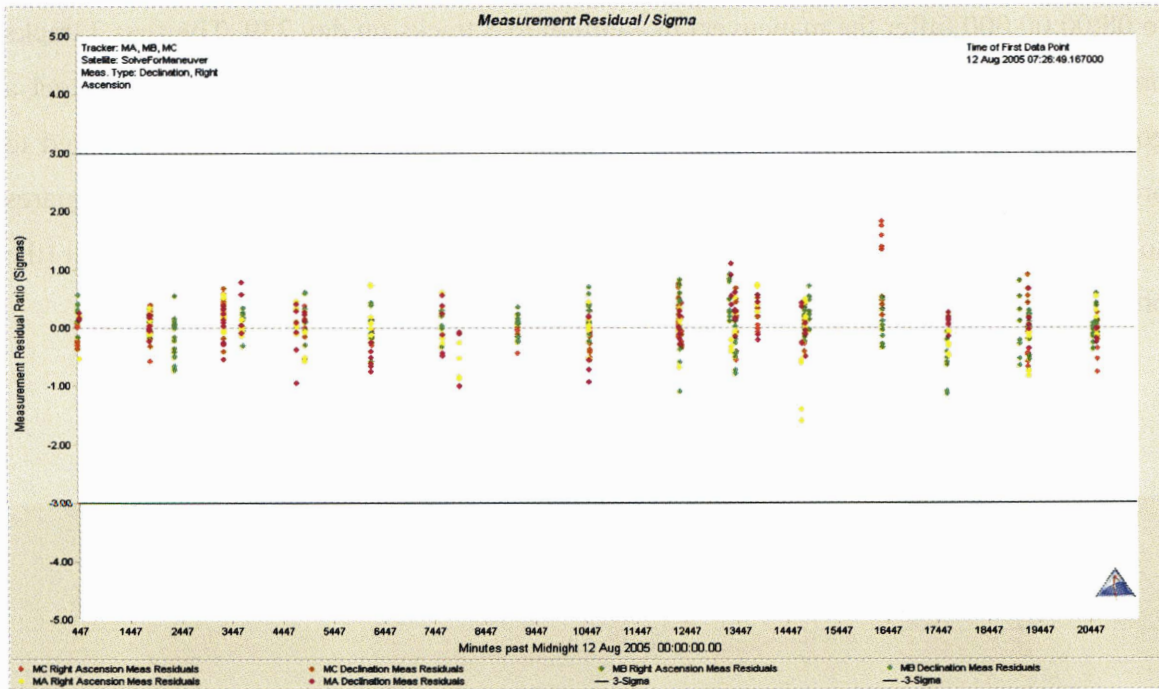
Figure 5.20. Residual ratios of shotgun solution for real data case.
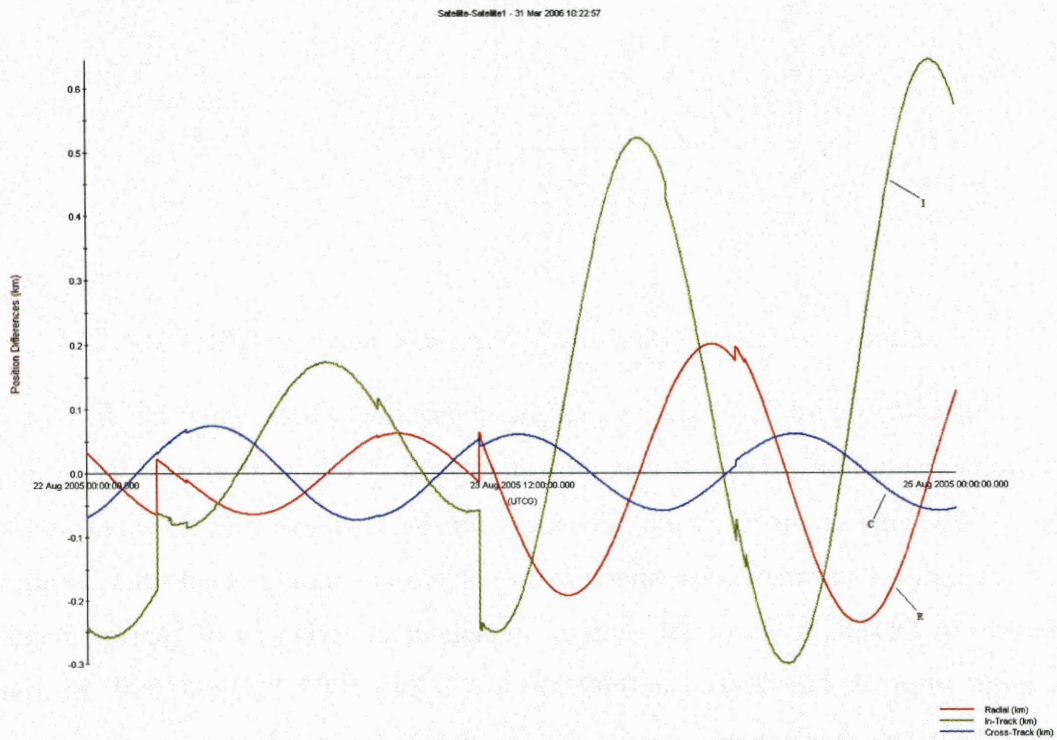
Figure 5.21. TDRS 5 comparison to truth before maneuver (km).

Finally, by comparing the resulting filter run and the TDRS 5 truth ephemeris from GSFC we have the following plots (figure 5.21 and 5.22). The GSFC ephemeris

does not include day 236 because the satellite maneuvered on that day. However, figure 5.21 is the before the maneuver comparison and figure 5.22 is the after maneuver comparison.

Figure 5.21 shows that in the days prior to the maneuver, the filter run was only a maximum of 700 m intrack different than the TDRS 5 truth ephemeris. Figure 5.22 shows the filter and truth differences again, but this time after the maneuver. After the first few tracks on day 238 (26 Aug 2005), the difference reaches a maximum of approximately 4.5 km in the intrack direction. By the time more tracking is available on day 239 (27 Aug 2005), the difference is down to less than 1 km intrack. This shows what can be accomplished with this type of regimen of post maneuver tracking.
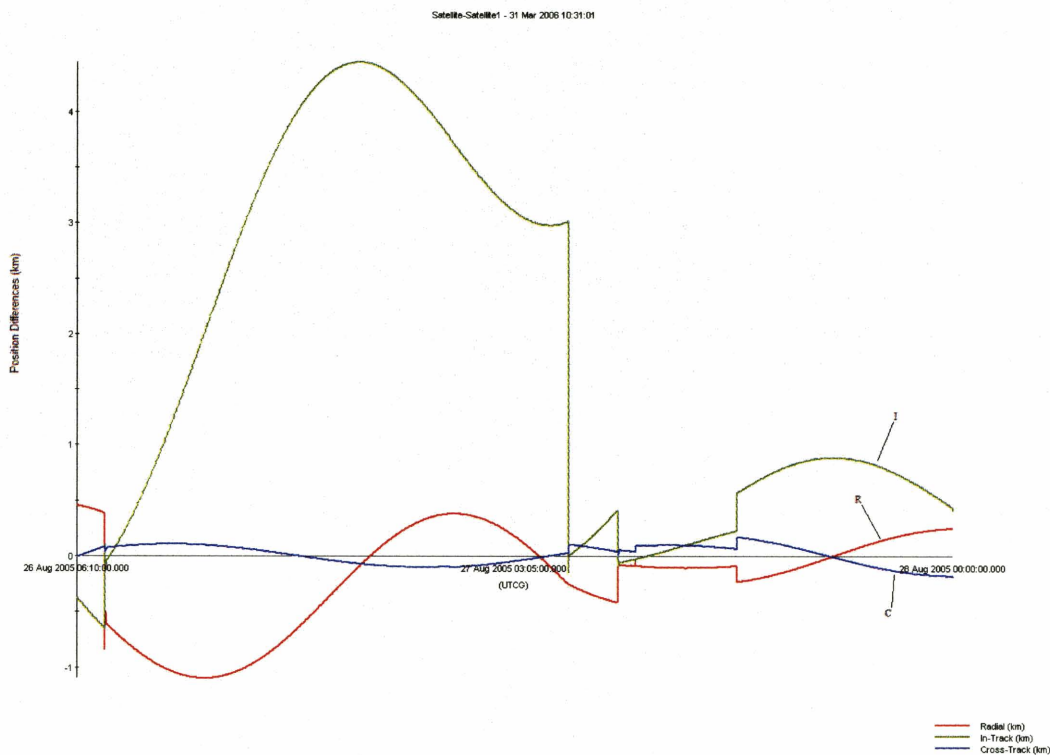
Figure 5.22.TDRS 5 comparison to truth after maneuver (km).

5.5. Summary

In this chapter, three methods of post maneuver orbit recovery were discussed. The three methods were the brute force method, IOD followed by a least squares solution

91

only, and the shotgun approach. Of the three methods, the shotgun approach is the most effective in fitting an orbit after a maneuver. However, a major factor in orbit recovery for the shotgun approach is that it requires a dedicated period of tracking after the maneuver has been detected. Cases were also explored that examined orbit recovery with less dense and more typical tracking. These cases required more time to recover an orbit.

6. Summary, Conclusions and Future Work

6.1. Summary and Conclusions

The primary focus of this thesis was to use a sequential filter to detect maneuvers and recover orbits for geosynchronous satellites. This process began by learning how to tune a sequential filter using the ODTK software. The tuning process involved tweaking the many settings available and comparing the orbit error (covariance) to the known satellite orbit (TDRS 5). After tuning the filter, these same settings were applied to other TDRS 5 orbits to check their validity. The orbit error and covariance were also evaluated by other means. They were compared using an overlap comparison and by a statistical method using future measurements.

After the covariance validity was established, the filter was tested to see how well it could detect maneuvers. A parametric study was performed given different Δv's, tracking intervals and tracking density prior to the maneuver. This testing would help determine what is required to detect a maneuver. With the tuned filter and optical tracking, simulations showed that a Δv of 1.0 m/s could be detected as soon as 15 minutes after the maneuver, a Δv of 0.1 m/s could be discernible within 6 hours and easily detected by 12 hours, and a Δv of 0.01 m/s took from 12 to 24 hours to detect with confidence.

Finally, three different methods of post maneuver recovery were attempted. The first method was to force the filter to accept all the measurements after a maneuver, to see if the filter could recover an orbit quickly based on new information while still utilizing the pre-maneuver orbit information. The second method was to attempt to see how much tracking and time was needed to do IOD and a least squares solution to recover the expected orbit. The final and best method is referred to as the "shotgun" method, as suggested by Hujsak [Ref 13], and its first implementation using optical tracking was tested here. Orbit accuracy post maneuver was quickly regained to the pre-maneuver level and a realistic covariance was also maintained. This method is sensitive to the ability to get a good enough post maneuver orbit with the IOD and least squares. In the

simulation it was found that for a Δv of 1.0 m/s, the quickest an accurate orbit could be regained required 6 post maneuver tracks in 6 hours. This method was also applied to a real data case. The density of tracking was just as available and another day of tracking was required to obtain the pre-maneuver orbit accuracy.

This work helped to confirm that a sequential estimator is the best approach to detect maneuvers and determine accurate orbits thereafter. The batch least squares approach to orbit estimation does not provide realistic covariances or lend itself to the methodology used here of modeling through unknown maneuver periods.

The specific contributions of this thesis were to "tune" the sequential filter for a geosynchronous satellite using ground based optical tracking, test and develop methods for validating the covariance (orbit overlap and a statistical method using measurements), perform a parametric study for maneuver detection, and to explore various methods of recovering accurate orbits quickly following a maneuver. Another important contribution of this work is to familiarize users of ODTK in how to operate the ODTK filter for this work and for work in future.

6.2. Suggestions for Future Work

There are many other projects that would be interesting to undertake for future work. One could do a similar analysis to this, but using combinations of radar and optical data from multiple sensors. There are many different types of maneuver scenarios that were not examined by this thesis. An important example would be Xenon-Ion Propulsion System (XIPS) maneuvers that are of much smaller size, but occur in longer duration and multiple times per day. This has already been initially explored by Hujsak [Ref 16]. Another important aspect of this study that could be improved upon would be to use Lambert's method to try and see if the size of the maneuver can be found more exactly and then be used to set the maneuver sigmas and process noise in the "shotgun" and brute force methods for having the filter process through a maneuver. In our case, we used prior knowledge of geosynchronous satellite maneuvering strategies to make a good

approximation for maneuver size. Also, for our simulations, we only performed the indicative Monte Carlo case (a.k.a. one trial). It would be important to perform a full Monte Carlo analysis where enough trials are performed to gain more confidence in the statistics of the results.

One immediate application of this work would be for close conjunction analysis of geosynchronous satellites [Ref 17] where the maneuvers of one or both satellites are unknown. An ultimate goal would be to have an automated system that would detect the maneuver and utilize one of the approaches for accurate orbit recovery. A Bayesian network possibly could be developed to add to the maneuver detection confidence by utilizing past history such as maneuver frequency and station keeping bounds.

[This page intentionally left blank.]

Appendix A. ODTK Filter Tuning Settings

The following is the scenario used in chapter 3. This shows all the ODTK settings used to tune the ODTK sequential estimator.

SCENARIO: 21639_Day175-190_tunefilter
ODTK.Ver = "4.0"
Description = ""
OverrideRunName = false
NewRunFilename = ""
DefaultTimes [-]

Processes [-]

StartMode = "EarliestSatEpoch"
StartTime = 1 Jun 2004 12:00:32.000
StopMode = "TimeSpan"
StopTime = 2 Jun 2004 12:00:32.000
TimeSpan = 24 hr

Intervals [-]

TimeSpan = 4 hr

Measurements [-]

Files [-]

TrackingFile [-]

Enabled = true
Filename = "C:\Documents and Settings\aaronb\My Documents\ODTK 4\21639-05145-05245.geosc"

RoundTripRepresentation = "OneWay"
EmbeddedWNSigmas [-]

Use = false
OnInputError = "UseDefaultWNSigma"

EditingMethods [-]

ThinningTime = 0 sec
GrossRawData [-]
InclusionTimes [-]
ExclusionTimes [-]

StateControls [-]

MinApogeeAltForSRP = 400 km
MaxPerigeeAltForDrag = 1000 km

EarthDefinition [-]

```
GravityModel = "C:\Program Files\AGI\ODTK 4\STKData\CentralBodies\Earth\EGM96.grv"
EOPFilename = "C:\Program Files\AGI\ODTK 4\DynamicEarthData\EOP.dat"
NutationMethod = "IERS1996"
EclipsingAtmosAlt = 23.862995940238 km
FluxData [-]

        InputMethod = "Read From File"
        InputFluxFile = "C:\Program Files\AGI\ODTK
        4\DynamicEarthData\stkFluxGeoMag.5yr.fxm"
        F10 = 150
        F10bar = 150
        KpUpdate = "3-Hourly"
        KpSubSamplingRatio = 1
        ApVersusKp = "Kp"
        Ap = 8.000000000000005
        Kp = 1.966301126216909

Ionosphere [-]

        IRI2001DataDirectory = "C:\Program Files\AGI\ODTK
        4\DynamicEarthData\IRI2001ModelData\"
```

Wizard [-]

```
StartTime = 1 Jun 2004 12:00:32.000
StopTime = 2 Jun 2004 12:00:32.000
```

Satellite[21639] [-]

```
Description = ""
OrbitState "Keplerian" [-]

        CoordFrame = "J2000"
        Epoch = 24 Jun 2005 00:00:32.000
        XPosition = -8238.153151184753 km
        YPosition = 41276.43319030222 km
        ZPosition = 2331.698784328843 km
        XVelocity = -2.998672645753322 km*sec^-1
        YVelocity = -0.6149840812101222 km*sec^-1
        ZVelocity = 0.2978762555286986 km*sec^-1

        ~ Approximate Keplerian ~
        SemiMajorAxis = 42170.18 km = 6.612 Re
        Eccentricity = 0.00037
        TrueArgofLat = 29.73 deg
        Inclination = 6.4 deg
        RAAN = 71.71 deg
        ArgofPerigree =13.28 deg

EstimateOrbit = true
EphemerisSource = "Reference Trajectory"
ReferenceTrajectory [-]

        EphemerisFileFormat = "STK Ephemeris"
        Filename = "C:\Documents and Settings\aaronb\My Documents\ODTK
        4\TDRS5_160_228_posvelall_J2000.e"

PhysicalProperties [-]
```

Mass = **1658.243 kg**

Attitude [-]

> Source = **"AlignedConstrained"**
> BodyAlignmentVec = **"MinusZ"**
> InertialAlignmentVec = **"Radial"**
> BodyConstraintVec = **"X"**
> InertialConstraintVec = **"Intrack"**
> Filename = **""**
> CenterOfMassInBodyFrame [-]
>
>> X = **0 m**
>> Y = **0 m**
>> Z = **0 m**

MeasurementProcessing [-]

> TrackingID = **21639**
> MeasTypes [-]
>
>> Measurement Type = **"Right Ascension" "Declination"**
>
> ResidualEditing [-]
>
>> NominalSigma = **3**
>> Dynamic [-]
>>
>>> Enabled = **false**
>>> HighSigma = **100**
>>> NumRejectToStart = **10**
>>> NumAcceptToStop = **3**
>>> InitialHighSigmaDuration = **0 min**
>
> ThinningTime = **0 sec**
> MinPassDelta = **20 min**

MeasurementStatistics [-]
MinGrazingAlt = **0 km**
OpticalProperties [-]

> PolarExclusion = **1 deg**
> ReferenceFrame = **"TEME Of Date"**
> AberrationCorrections = **"None"**

RangingMethod = **"Skin Track"**
IonosphereModel [-]

> Enabled = **false**
> Model = **"IRI2001"**
> TransmitFreq = **2267.5 MHz**
> ReceiveFreq = **1815.7715 MHz**

ForceModel [-]

> Gravity [-]
>
>> DegreeAndOrder = **8**

Tides [-]

SolidTides = **false**
TimeDependentSolidTides = **false**
minSolidTideAmplitude = **0**
OceanTides = **false**
MaxDegreeOrderOceanTides = **4**
minOceanTideAmplitude = **0**

GeneralRelativityCorrection = **false**
VariationalEquations [-]

Degree = **2**
Order = **0**

ProcessNoise [-]

Use = **"Yes"**
OmissionErrorModeling [-]

Enabled = **true**
Scale = **1**

CommissionErrorModeling [-]

Enabled = **true**
Scale = **1**

ThirdBodies [-]

Sun = **true**
Moon = **true**
Planets [-]
UseInVariationalEquations = **true**

Drag [-]

Use = **"No"**
AtmDensityModel = **"CIRA 1972"**
EstimateDensity = **false**
SpecMethod = **"Mass Area Cd"**
CD = **2**
Area = **40 m^2**
DensityCorrHalfLife = **180 min**
DensityCorrInitialEstimate = **0**
DensityCorrSigmaScale = **1**
DensityRatioRoot = **2**
DensityRatioIncreaseThreshold = **1**
EstimateBallisticCoeff = **false**
BallisticCoeffCorrInitialEstimate = **0**
BallisticCoeffSigma = **0.1**
BallisticCoeffCorrHalfLife = **10080 min**
SunPosMethod = **"True"**
Initialization [-]

Duration = **0 min**

DensityCorrSigma = 10
DensityCorrHalfLife = 28800 min

SolarPressure [-]

Use = "Based On Orbit"
EstimateSRP = true
CPNominal = 1.37
Area = 40 m^2
CPInitialEstimate = 0
CPSigma = 0.1
CPHalfLife = 14400 min
ReflectionModel = "Sphere with perfect absorption"
SunPosMethod = "Apparent"
UseInVariationalEquations = true
AddProcessNoise = true
EclipticNorthFraction = 0.3
EclipticPlaneFraction = 0.1

Plugin [-]

Use = false
PluginID = "AgAsHpopPlugin4.AgAsHpopPluginEngine"
PluginConfig [-]

UnmodeledAccelerations [-]

ProcessNoise [-]

RadialVelocitySigma = 0 cm*sec^-1
IntrackVelocitySigma = 0 cm*sec^-1
CrosstrackVelocitySigma = 0 cm*sec^-1
TimeInterval = 2 min

InstantManeuvers [-]
PermanentManeuverStates [-]
FiniteManeuvers [-]
OrbitErrorTransitionMethod = "Numerical"

PropagatorControls [-]

IntegrationMethod = "RKF 7(8)"
StepSize [-]

Time = 6 min
TrueAnomaly = 2 deg
EccentricityThreshold = 0.04

EphemerisGeneration [-]

StartTime = 24 Jun 2005 00:00:32.000
Span = 20160 min
TimeStep = 10 min
CreateSTKFile = true
STKFilename = "C:\Documents and Settings\aaronb\My Documents\ODTK
4\Ephemeris\21639_05175_05189.e"

SendToScript = false
ScriptFilename = ""
Acceleration = false

OrbitUncertainty [-]

R_sigma = 250 m
I_sigma = 2000 m
C_sigma = 350 m
Rdot_sigma = 0.5 m*sec^-1
Idot_sigma = 0.05 m*sec^-1
Cdot_sigma = 0.05 m*sec^-1
RI_correlation = 0
RC_correlation = 0
RRdot_correlation = 0
RIdot_correlation = 0
RCdot_correlation = 0
IC_correlation = 0
IRdot_correlation = 0
IIdot_correlation = 0
ICdot_correlation = 0
CRdot_correlation = 0
CIdot_correlation = 0
CCdot_correlation = 0
RdotIdot_correlation = 0
RdotCdot_correlation = 0
IdotCdot_correlation = 0

FilterEvents [-]

MeasurementRejectThreshold [-]

NumForWarning = 0
NumForAlert = 0
OnWarning = "C:\Program Files\AGI\ODTK
4\ODTK\AppData\Scripts\DoNothing.vbs"
OnReturnFromWarning = "C:\Program Files\AGI\ODTK
4\ODTK\AppData\Scripts\DoNothing.vbs"
OnAlert = "C:\Program Files\AGI\ODTK
4\ODTK\AppData\Scripts\DoNothing.vbs"

MeasurementAcceptTimer [-]

TimeGapForWarning = 0 min
TimeGapForAlert = 0 min
OnWarning = "C:\Program Files\AGI\ODTK
4\ODTK\AppData\Scripts\DoNothing.vbs"
OnReturnFromWarning = "C:\Program Files\AGI\ODTK
4\ODTK\AppData\Scripts\DoNothing.vbs"
OnAlert = "C:\Program Files\AGI\ODTK
4\ODTK\AppData\Scripts\DoNothing.vbs"

TrackingSystem[Tracking_Stations] [-]

Description = ""
TroposphereModel [-]

Enabled = false

Model = "SCF"

IonosphereModel [-]

Enabled = false
Model = "IRI2001"
TransmitFreq = 2267.5 MHz
ReceiveFreq = 1815.7715 MHz

Facility[MA] [-]

Description = ""
Position "Cartesian" [-]

Lat = 0.3614237710491347 rad
Lon = -2.727212961248004 rad
Alt = 3059.628797585624 m

TrackingID = 231
Estimate = "Nothing"
LocationErrors [-]

SouthSigma = 30 m
EastSigma = 30 m
AltitudeSigma = 30 m
SouthEastCorrelation = 0
SouthAltCorrelation = 0
EastAltCorrelation = 0

MeasurementStatistics [-]

Right Ascension [-]

Type "Right Ascension" [-]

Bias = 0 arcSec
BiasSigma = 2 arcSec
BiasHalfLife = 1440 min
WhiteNoiseSigma = 2 arcSec
EstimateBias = false

Declination [-]

Type "Declination" [-]

Bias = 0 arcSec
BiasSigma = 2 arcSec
BiasHalfLife = 1440 min
WhiteNoiseSigma = 2 arcSec
EstimateBias = false

MinElevation = 5 deg
MaxElevation = 90 deg
RangingMethod = "Skin Track"
AntennaType = "Optical"
BoresightAzimuth = 0 deg

BoresightElevation = 0 deg
OpticalProperties [-]

> PolarExclusion = 1 deg
> ReferenceFrame = "TEME Of Date"
> AberrationCorrections = "None"

TroposphereModel [-]

> Enabled = "Based on Tracking System"
> Model = "SCF"

TroposphereData [-]

> SurfaceRefractivity = "Constant"
> Value = 340
> Polynomial1 = 340
> Polynomial2 = 0
> Polynomial3 = 0
> Polynomial4 = 0
> Polynomial5 = 0
> Polynomial6 = 0
> Polynomial7 = 0
> Polynomial8 = 0
> Polynomial9 = 0
> Polynomial10 = 0
> Polynomial11 = 0

IonosphereModel [-]

> Enabled = "Based on Tracking System"
> Model = "IRI2001"
> TransmitFreq = 2267.5 MHz
> ReceiveFreq = 1815.7715 MHz

Facility[MB] [-]

Description = ""
Position "Cartesian" [-]

> Lat = 0.3614237899338263 rad
> Lon = -2.72720755161025 rad
> Alt = 3059.622836373999 m

TrackingID = 232
Estimate = "Nothing"
LocationErrors [-]

> SouthSigma = 30 m
> EastSigma = 30 m
> AltitudeSigma = 30 m
> SouthEastCorrelation = 0
> SouthAltCorrelation = 0
> EastAltCorrelation = 0

MeasurementStatistics [-]

Right Ascension [-]

Type "Right Ascension" [-]

Bias = 0 arcSec
BiasSigma = 2 arcSec
BiasHalfLife = 1440 min
WhiteNoiseSigma = 2 arcSec
EstimateBias = false

Declination [-]

Type "Declination" [-]

Bias = 0 arcSec
BiasSigma = 2 arcSec
BiasHalfLife = 1440 min
WhiteNoiseSigma = 2 arcSec
EstimateBias = false

MinElevation = 5 deg
MaxElevation = 90 deg
RangingMethod = "Skin Track"
AntennaType = "Optical"
BoresightAzimuth = 0 deg
BoresightElevation = 0 deg
OpticalProperties [-]

PolarExclusion = 1 deg
ReferenceFrame = "TEME Of Date"
AberrationCorrections = "None"

TroposphereModel [-]

Enabled = "Based on Tracking System"
Model = "SCF"

TroposphereData [-]

SurfaceRefractivity = "Constant"
Value = 340
Polynomial1 = 340
Polynomial2 = 0
Polynomial3 = 0
Polynomial4 = 0
Polynomial5 = 0
Polynomial6 = 0
Polynomial7 = 0
Polynomial8 = 0
Polynomial9 = 0
Polynomial10 = 0
Polynomial11 = 0

IonosphereModel [-]

Enabled = "Based on Tracking System"

Model = "IRI2001"
TransmitFreq = 2267.5 MHz
ReceiveFreq = 1815.7715 MHz

Facility[MC] [-]

Description = ""
Position "Cartesian" [-]

Lat = 0.3614321634709198 rad
Lon = -2.727207075160802 rad
Alt = 3059.654057970593 m

TrackingID = 233
Estimate = "Nothing"
LocationErrors [-]

SouthSigma = 30 m
EastSigma = 30 m
AltitudeSigma = 30 m
SouthEastCorrelation = 0
SouthAltCorrelation = 0
EastAltCorrelation = 0

MeasurementStatistics [-]

Right Ascension [-]

Type "Right Ascension" [-]

Bias = 0 arcSec
BiasSigma = 2 arcSec
BiasHalfLife = 1440 min
WhiteNoiseSigma = 2 arcSec
EstimateBias = false

Declination [-]

Type "Declination" [-]

Bias = 0 arcSec
BiasSigma = 2 arcSec
BiasHalfLife = 1440 min
WhiteNoiseSigma = 2 arcSec
EstimateBias = false

MinElevation = 5 deg
MaxElevation = 90 deg
RangingMethod = "Skin Track"
AntennaType = "Optical"
BoresightAzimuth = 0 deg
BoresightElevation = 0 deg
OpticalProperties [-]

```
            PolarExclusion = 1 deg
            ReferenceFrame = "TEME Of Date"
            AberrationCorrections = "None"

    TroposphereModel [-]

            Enabled = "Based on Tracking System"
            Model = "SCF"

    TroposphereData [-]

            SurfaceRefractivity = "Constant"
            Value = 340
            Polynomial1 = 340
            Polynomial2 = 0
            Polynomial3 = 0
            Polynomial4 = 0
            Polynomial5 = 0
            Polynomial6 = 0
            Polynomial7 = 0
            Polynomial8 = 0
            Polynomial9 = 0
            Polynomial10 = 0
            Polynomial11 = 0

    IonosphereModel [-]

            Enabled = "Based on Tracking System"
            Model = "IRI2001"
            TransmitFreq = 2267.5 MHz
            ReceiveFreq = 1815.7715 MHz
```

Filter[21639_Filter] [-]

```
Description = ""
SatelliteList = "21639"
TrackerList = "Tracking_Stations.MA" "Tracking_Stations.MB" "Tracking_Stations.MC"
MeasTypes [-]

        Measurement Type = "Right Ascension" "Declination"

RejectMeasTypes [-]
ProcessControl [-]

        StartMode = "Initial"
        StartTime = 24 Jun 2005 00:00:32.000
        SelectedRestartTime = "09 Jul 2005 00:00:00.000"
        AutoSelectedRestartTime = "09 Jul 2005 00:00:00.000"
        StopMode = "StopTime"
        StopTime = 9 Jul 2005 00:00:32.000
        TimeSpan = 360 hr
        ProcessNoiseUpdateInterval = 2 min

CurrentTime = 9 Jul 2005 00:00:32.000
Restart [-]
```

SaveRecordsToFile = true
MaxRecordsInFile = 720
SaveFrequency = 60 min
GaussMarkovUpdates [-]

Enabled = false
GaussMarkovList [-]

OnStateSizeChange [-]

StateReductionMethod = "SimpleTruncation"
IfCovGoesNegative [-]

Action = "PromptUser"
PercentSigmaDeweighting = 10

OptionalSolveForParms [-]

MeasBiases = false

Output [-]

DataArchive [-]

Filename = "C:\Documents and Settings\aaronb\My Documents\ODTK
4\DataArchive\21639_Day175-190_tunefilter.filrun"
OutputStateHistory = "AllTimes"
EveryNSteps = 1
SaveOnlyLastMeasPerStep = false
OutputMeasHistory = true
OutputManeuvers = true
OutputSummary = true
OutputHistograms = true
HistogramSize = 3
NumberHistogramBins = 22

Display [-]

EveryNMeasUpdates = 1
EveryNTimeUpdates = 1
ShowPassTimes = true

ClearMsgLog = false
SmootherData [-]

Generate = true
Filename = "C:\Documents and Settings\aaronb\My Documents\ODTK
4\Smoother\22927.rough"
TimeMode = "FilterSpan"
StartTime = 19 Sep 2002 00:00:32.000
StopTime = 30 Mar 2008 00:00:32.000

STKEphemeris [-]

DuringProcess [-]

Generate = true
TimeGrid = "Uniform"

Predict [-]

Generate = false
TimeStep = 2 min
StopMode = "TimeSpan"
TimeSpan = 1440 min
StopTime = 3 Jan 2006 00:00:32.000

Acceleration = false
Covariance = true
CovarianceType = "Position Velocity 6x6 Covariance"
OutputDirectory = "C:\Documents and Settings\aaronb\My Documents\ODTK 4\Ephemeris"
FileNamingOption = "ProcessStart"
Files [-]

EphemInfo [-]

Satellite = "21639"
Filename = "C:\Documents and Settings\aaronb\My Documents\ODTK 4\Ephemeris\Sat_21639_Fil_20050624_000000.e"

Events [-]

OnInternalError = "C:\Program Files\AGI\ODTK 4\ODTK\AppData\Scripts\DoNothing.vbs"
OnResume = "C:\Program Files\AGI\ODTK 4\ODTK\AppData\Scripts\DoNothing.vbs"
OnComplete = "C:\Program Files\AGI\ODTK 4\ODTK\AppData\Scripts\DoNothing.vbs"
OnHalt = "C:\Program Files\AGI\ODTK 4\ODTK\AppData\Scripts\DoNothing.vbs"
OnNoMoreMeas = "C:\Program Files\AGI\ODTK 4\ODTK\AppData\Scripts\DoNothing.vbs"

DebugOutput [-]

Generate = false
Filename = ""

Simulator[21639_Sim] [-]

Description = ""
SatelliteList = "21639"
TrackingStrandList [-]

Tracking Strand = "Tracking_Stations.MA - *" "Tracking_Stations.MB - *" "Tracking_Stations.MC - *"

MeasTypes [-]

Measurement Type = "Right Ascension" "Declination"

StartTime = 24 Jun 2005 00:00:32.000
StopTime = 9 Jul 2005 00:00:32.000

```
TimeStep = 10 min
CustomTrackingIntervals [-]

        Enabled = false
        IntervalFileLinks [-]
        Schedule [-]

ErrorModeling [-]

        NoDeviations = false
        DeviateOrbits = false
        DeviateDensity = false
        DeviateBCoeff = false
        DeviateSolarP = false
        DeviateTranspDelay = false
        DeviateMeasBiases = false
        DeviateManeuvers = false
        AddProcessNoise = false
        AddManeuverProcessNoise = false
        AddGPSReceiverClockProcessNoise = false
        AddMeasWhiteNoise = true
        DeviateStationLocations = false
        RandomSeed = 1

ErrorScaling [-]

        Orbits = 1
        Density = 1
        BCoeff = 1
        SolarP = 1
        TranspDelay = 1
        MeasBiases = 1
        Maneuvers = 1
        StationLocations = 1

UpdateFilterTimes = true
Output [-]

        Measurements [-]

                Filename = "C:\Documents and Settings\aaronb\My Documents\ODTK
                4\TrackingData\21639_Day175-190_tunefilter.geosc"
                Files [-]

                        Simulated Measurement File = "C:\Documents and
                        Settings\aaronb\My Documents\ODTK
                        4\TrackingData\21639_Day175-190_tunefilter.geosc"

        DataArchive [-]

                Filename = "C:\Documents and Settings\aaronb\My Documents\ODTK
                4\DataArchive\21639_Day175-190_tunefilter.simrun"
                OutputStateHistory = "AllTimes"
                EveryNSteps = 1
                OutputMeasHistory = true
                OutputManeuvers = true
                OutputSummary = true
                OutputHistograms = true
                HistogramSize = 3
                NumberHistogramBins = 22
                OutputPerturbations = true
```

STKEphemeris [-]

DuringProcess [-]

Generate = **true**
TimeGrid = **"Uniform"**

Predict [-]

Generate = **false**
TimeStep = **1 min**
StopMode = **"TimeSpan"**
TimeSpan = **720 min**
StopTime = **27 Mar 2004 12:00:32.000**

Acceleration = **false**
Covariance = **false**
CovarianceType = **"Position 3x3 Covariance"**
OutputDirectory = **"C:\Documents and Settings\aaronb\My Documents\ODTK 4\Ephemeris"**
FileNamingOption = **"ProcessStart"**
Files [-]

EphemInfo [-]

Satellite = **"21639"**
Filename = **"C:\Documents and Settings\aaronb\My Documents\ODTK 4\Ephemeris\21639_Truth_20050624_000000.e"**

Events [-]

OnInternalError = **"C:\Program Files\AGI\ODTK 4\ODTK\AppData\Scripts\DoNothing.vbs"**
OnStart = **"C:\Program Files\AGI\ODTK 4\ODTK\AppData\Scripts\DoNothing.vbs"**
OnComplete = **"C:\Program Files\AGI\ODTK 4\ODTK\AppData\Scripts\DoNothing.vbs"**
OnHalt = **"C:\Program Files\AGI\ODTK 4\ODTK\AppData\Scripts\DoNothing.vbs"**

Smoother[Smoother1] [-]

Description = **""**
Input [-]

Files [-]

RoughFile [-]

Enabled = **true**
Filename = **"C:\Documents and Settings\aaronb\My Documents\ODTK 4\Smoother\22927.rough"**
StartTime = **24 Jun 2005 00:00:32.000**
StopTime = **9 Jul 2005 00:00:32.000**

Remove = **false**

ProcessControl [-]

StartMode = "LatestFilterTime"
StopMode = "EarliestFilterTime"
TimeSpan = 24 hr
OutputLag = 0 min
IntervalLength = 86400 min
IntervalOverlap = 720 min

Output [-]

DataArchive [-]

Filename = "C:\Documents and Settings\aaronb\My Documents\ODTK 4\DataArchive\21639_Day175-190_tunefilter.smtrun"
OutputStateHistory = "AllTimes"
EveryNSteps = 1
OutputManeuvers = true

STKEphemeris [-]

DuringProcess [-]

Generate = true
TimeGrid = "Filter"

Predict [-]

Generate = false
TimeStep = 1 min
StopMode = "TimeSpan"
TimeSpan = 720 min
StopTime = 30 Jun 2000 23:59:27.816

Acceleration = false
Covariance = true
CovarianceType = "Position Velocity 6x6 Covariance"
OutputDirectory = "C:\Documents and Settings\aaronb\My Documents\ODTK 4\Ephemeris"
FileNamingOption = "ProcessStart"
Files [-]

EphemInfo [-]

Satellite = "21639"
Filename = "C:\Documents and Settings\aaronb\My Documents\ODTK 4\Ephemeris\Sat_21639_Smooth_20050624_00 0000.e"

Appendix B. MATLAB Program

```matlab
function OD_Error_Statistics2(func1)

%This is a program that Reads an STKOD "Measurment Residuals" report
%and graphs both the CDF with the theoretical CDF of the residual
%ratios and the PDF with the theoretical PDF of the file.
%This program also performs the K-S Test.

%This reads a .txt (Notepad) file that was created by Exporting a
%"Tab Separated Text" file from a "Measurement Residuals" report in
%STKOD. The file should be opened with (Notepad) and saved as .txt file
%to be input into this program.

disp(' ')
disp('The file being graphed and evaluated is:')
disp(func1)
disp(' ')

%This line reads in the file. Each column is read into its own column.
%"'headerlines', 8" starts the textread on the 9th line.
[Time, Tracker, MeasType, Value, Unit1, Residual,Unit2, Sigma,
Ratio,...
    Flag] =
textread(func1,'%29c%4q%q%10f%q%10f%q%f%f%q','headerlines',8);

Di_unsorted = Ratio;          %The Residual Ratios
%Next line Sorts the residual ratios in ascending order.
Di_temp = sort(Di_unsorted);
%Next line can be used to dump an element if desired
Di = Di_temp(1:length(Di_temp));

%Now we have the ratio in a column vector that is sorted in
%ascending order.

NumDi = length(Di);   %Number of residuals in file
max(Di);              %Maximum Residual

%Need an "index" from 0 to 1 for each Di.
Y = [1/length(Di):1/length(Di):1]';   % the ' makes this a column vector

plot(Di, Y)
hold;
%abcissa for cdf is Di.

%build the Normal Cumulative Distribution Function
P1 = normcdf(Di, 0 ,1);
plot(Di, P1, 'r')

%The following are labels for x-axis, y-axis, title and legend
xlabel('Di values')
ylabel('Cumulative Probability Distribution')
title('K-S Test')
```

```matlab
legend('Di','CDF', 'location', 'Best')


%This part is to create a PDF form for the data. This is just to get
%another feel for how the data is distributed.

figure;
n = 7;                     %Number of bins for the data.
N = hist(Di,n);            %Puts all the data into a n-bin Histogram form
NN = sum(N);
Norm_Hist = N/NN;          %Creates a Normalized Histogram
z = abs(min(Di)) + abs(max(Di));
YY = min(Di):(z)/(n):max(Di);   %Creates the array of bin end points.
cc = z/(2*n);              %Finds the middle of the bin.
YYY = YY(1:n)+cc;          %This puts the points in the middle of the bins.

%begin kstest
[H, P, KSSTAT] = kstest2(Y, P1, 0.001, 'unequal')
%P is the p-value - how closely the two distributions are related
%KSSTAT is the K-S Statistic - The maximum absolute difference between
%the two distributions.
%end kstest

%begin Q lamda test
NumDi;
KSSTAT;
Qks = 0;

for i = 1:200
    Qks = Qks + (-1)^(i-1)*exp(-2*(i^2)*(sqrt(NumDi)*KSSTAT)^2);
end
Qks = 2*Qks
%end Q lamda test

bar(YYY, Norm_Hist)        %Plots the bins in a bar form.
hold;
Q = normpdf(Di, 0 ,1);     %This is the S-N PDF of the data
plot(Di, Q, 'g')           %Plots the Gaussian theoretical curve.

%The following are labels for x-axis, y-axis, title and legend
xlabel('Sigma')
ylabel('Probability Density Function')
title('PDF Test')
legend('Actual Hist Vals', 'Theo PDF', 'location', 'Best')
```

List of References

1. Rumsfeld, Donald H., et al. "Report to the Commission to Assess United States National Security Space Management and Organization." Pursuant to Public Law 106-65. January 11, 2001. Available at http://www.defenselink.mil/pubs/space20010111.html.

2. Vallado, David A. Fundamentals of Astrodynamics and Applications. Microcosm Press, 2001. Pgs. 445-485, 685, 718, 723.

3. Tapley, Byron D., Bob E. Schutz, and George H. Born. Statistical Orbit Determination. Academic Press, 2004.

4. Wright, James R. "Orbit Determination Tool Kit: Theorems & Algorithms." Analytical Graphics Inc., 2004.

5. Wright, James R. and Kevin O'Brien. "STK/OD: A Technical Summary." Analytical Graphics Inc., 2002.

6. Folcik, Zack J. Private Communication, 2006.

7. Tracking and Data Relay Satellite Program Webpage. Mission and Spacecraft Library: Jet Propulsion Laboratory. Available at http://msl.jpl.nasa.gov/Programs/tdrss.html.

8. Ground-Based Electro-Optical Deep Space Surveillance Webpage. Federation of American Scientists. Available at http://www.fas.org/spp/military/program/track/geodss.htm.

9. Abbot, Richard I. Private Communication, 2006.

10. Kommers, Jefferson M. Private Communication, 2006.

11. Mills, Frederick C. Statistical Methods. Holt, Rinehart and Winston, 1955. Pgs. 522-529.

12. Press, William H., Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. Numerical Recipes: The Art of Scientific Computing. Cambridge University Press, 1986. Pgs. 469-475.

13. Hujsak, Richard S. Private Communication, 2006.

14. Bate, Roger R., Donald D. Mueller, and Jerry E. White. Fundamentals of Astrodynamics. Dover Publications, 1971. Pgs. 228-275.

15. Campan, G., Alby, F., and Gautier, H. Station-Keeping Techniques for Geostationary Satellites. Cepadues-Editions, 1995. Pgs. 1008-1009.

16. Hujsak, Richard S. "Orbit Determination During High Thrust and Low Thrust Maneuvers," AAS 05-136. Analytical Graphics Inc., Exton, PA, 2002.

17. Abbot, R. I., R. Clouser, E. W. Evans, J. Sharma, R. Sridharan, and L. E. Thornton. "Close Geostationary Satellite Encounter Analysis: 1997-2001," Spaceflight Mechanics, Vol. 112, Advances in the Astronautical Sciences, 2002.