

Performance Analysis of Apollo Navigational Starter Routine

Stoyan I. Ivanov¹ and Greg Holt²
NASA Johnson Space Center, Houston, Texas, 77058

The focus of this project is to recreate and analyze the effectiveness of the original Apollo Starter Routine (ASR) which was used to generate the state vector of the Apollo spacecraft based on a series of radiometric observations. The original Apollo navigation software is unavailable in a modern programming language and the original coding has not been preserved. This necessitates its recreation using the original software documentation. Space Shuttle navigation software does not typically use the ASR or an algorithm like it since the Shuttle's state vector is easily deduced from GPS information or other sources. However, this tactic will be ineffective when trying to determine the state vector of a craft approaching, departing or in orbit around the Moon since the GPS network faces the surface of the Earth, not outer space. The recreation of the ASR from the original documentation is therefore vital as a simulation baseline for the navigation software under development for the Constellation program. The algorithms that make up the ASR will be extracted from the original documentation and adapted for and then implemented in a modern programming language; the majority of it will be coded in Matlab. The ASR's effectiveness will then be tested using simulated tracking data. The ability of the ASR to handle realistically noisy data and the accuracy with which it generates state vectors were analyzed. The ASR proved to be robust enough to process data with range and angle noise as large as 10,000 meters and 10^{-6} radians together and 300,000 meters and $5 \cdot 10^{-4}$ radians separately at Lunar distances. The ASR was able to handle marginally more noise at distances closer to the Earth where the angle noise was less significant. The ASR is capable of effectively processing 40-80 data points gathered at a rate of one per 20 seconds at close Earth orbit and up to 28-40 data points gathered at a rate of one per minute at distant Earth orbit and Lunar orbit.

Nomenclature

ρ	= The radiometric measurement of range produced by a ground radar station.
E,A	= Elevation and Azimuth angles of rotation of a ground radar station.
e	= The eccentricity of the Earth reference ellipsoid.
f	= The flattening of the reference ellipsoid used to approximate the true Earth.
N,W	= Intermediate variables used in calculating the Radar station location.
a	= The semimajor radius of the Earth, equatorial radius.
φ	= Geodetic Latitude
λ	= Geodetic Longitude
h	= Height of the radar station above the reference ellipsoid.
Xs, Ys, Zs	= Geocentric Coordinates of the radar station on the surface of the Earth.
θ	= Angle between terrestrial X axis and J2000 mean X axis.
f, g	= Values used to approximate the gravitational force from a single central body.

I. Introduction

The Apollo Starter Routine (ASR) provides an estimate of the state vector of a spacecraft. It is used to generate a guess whenever the state vector of the vehicle in question has been lost or is found to be in considerable error. The guess provided by the ASR is used by the main navigational program as a starting point for batched differential

¹ NASA USRP, Georgia Institute of Technology

² Aerospace Engineer, Descent Analysis Group, NASA Johnson Space Center, Mail Stop – DM42.

correction. It is continually refined based on current observations by the main navigational program and used to propagate the future path of the vehicle using high-fidelity gravitational and force models [7].

The ASR can generate a satisfactory estimate of a state vector with 15 to 30 minutes of tracking data, though it is capable of producing a state vector from as few as 6 points for a craft in close Earth orbit [5].

The ASR is useful now in its application to the future of the Constellation program. The Space Shuttle does not utilize such an algorithm because its state vector could be quickly determined using the GPS satellites. The same method, however, does not work for a spacecraft travelling to the moon or in orbit around the moon because the GPS satellites are oriented to face the Earth's surface. As a result, the ASR, or a ASR like it, is required in order to ensure the ability to calculate useful approximations of the state vector of a spacecraft moving towards or in orbit around the moon. The best use of the ASR is as a baseline for future simulations and development of the Constellation navigational software. Since this ASR was used by Apollo, and Apollo successfully navigated to the moon and back on several occasions, it represents a baseline of performance. Improvements and ideas can be extracted from the ASR in the development of the Constellation counterpart.

The ASR, as implemented in the Apollo navigational software, was written in Fortran and machine code. The ASR reconstruction, however, was implemented in Matlab and Java. The transition, from the Fortran documentation and flowcharts to the Matlab and Java programming languages, required the modification of some parts of the code. However, the overall program flow has not been altered and the functionality has been preserved [5].

II. ASR Overview

The ASR takes input in the form of batched data from a single radiometric source. The batches are limited to a maximum of 80 observations, though multiple batches can be processed at once. Each batch as well as each set of batches, called a superbatch, is sorted by the time of their observations. A batch consists of a sequence of observations from the same radiometric source. Each batch can only have information from one source, though a superbatch can contain information from any number of sources.

Once the data is input, it is converted from the native units, range and angle measurements with respect to the radar station, to rectangular coordinates. Additionally, the reference frame is converted to a J2000 mean inertial reference frame. If the ASR is performing the calculations for a craft or object in Lunar orbit, it converts the observations from geocentric to selenocentric by a simple translational conversion.

Once the conversions are complete, the data is aggregated into a list of converted observations sorted by time. Then a rough estimate of the state vector is calculated prior to the convergence loop using the first two points in the converted list. The position of the first point is taken as the initial position of the state vector. The velocity is estimated as the change in position over the time between the first and second observations.

The ASR then enters the convergence loop. There the variables f and g are calculated using classic universal variable algorithms. The rough estimate of the state vector is propagated and compared to the actual observed position values of the craft and is then iteratively refined. This continues until either the state vector has converged, measured as a tolerance below which no 'change' in the value of the state vector occurs, or the maximum number of iterations have been performed. While the original software documentation limited the number of iterations to 10, this is likely due to the limited computer power at the time. Currently, the maximum number of iterations performed before exiting the loop is 150. For the majority of data, however, the state vector estimation converged long before the limit was reached. The tolerance for the convergence is 1 meter for Lunar orbits and distances and .01 meters for Earth orbits and distances. The trans-Earth case is treated as a Lunar distance. The tolerance is measured component wise, that is each component of the state vector position is examined for a change in its value, if the change in all three components is individually less than the tolerance, then the state vector solution is said to have converged. If, however, the change in even one of the components is greater than the tolerance, even if the other two components have not changed at all, the vector has not converged and another iteration is performed.

After the state vector estimation has converged, it is propagated across the time-arc using the f and g values. The data produced, an estimated trajectory of the craft based on the converged state vector estimation, is placed within a data list. It is then converted from the J2000 geocentric or selenocentric coordinates to the original topocentric,

radiometric coordinates. Residuals are then generated between the estimated positions and the actual measured positions for the range, azimuth and elevation measurements.

The next sections outline the program flow of the ASR, from the format of data to the calculation of the state vector.

A. Batcher

The Batcher is the Matlab implemented function used to process incoming data in the form of '.txt' files. It reads the observations and creates a batch structure which contains the observations, in the 'native' coordinates, the time stamp of the observations and their identification that is the radar station from which they are taken. If there is more than one filename input into the function, it generates a superbatch that is an array of batch structures. The batch array is ordered by time with earlier batches coming before later batches. The data within the different batches is stored in the same manner. Batches are homogenous i.e. the list of observations they hold originates from only one radar station, but superbatches can contain observations from several radar stations.

B. Coordinate Conversion

The observations are taken into the ASR in the radar-station centered topocentric polar coordinate frame. They are taken in initially as polar coordinates, angle and range values, but the observations are subsequently converted to rectangular coordinates $[X, Y, Z]$ by the following equations:

$$X = \rho \cos E \sin A \quad (1)$$

$$Y = \rho \cos E \cos A \quad (2)$$

$$Z = \rho \sin E \quad (3)$$

These topocentric coordinates are then converted into geocentric coordinates. This conversion, however, requires the location of the physical station on the surface of the earth. The following equations are used to determine the location of the station in geocentric coordinates $[X_s, Y_s, Z_s]$. The x axis of this system is at the intersection between the prime meridian and the equator, the z axis is through the pole, and the y axis completes the right-handed system [4].

$$e^2 = 2f - f^2 \quad (4)$$

$$W = (1 - e^2 \sin^2 \varphi)^{\frac{1}{2}} \quad (5)$$

$$N = a/W \quad (6)$$

$$X_s = (N + h) \cos \varphi \cos \lambda \quad (7)$$

$$Y_s = (N + h) \cos \varphi \sin \lambda \quad (8)$$

$$Z_s = [N(1 - e^2) + h] \sin \varphi \quad (9)$$

The positions of the craft relative to the station $[X, Y, Z]$ are then converted into the same reference frame as the station's geocentric coordinates $[X_s, Y_s, Z_s]$. The following equations change the reference frame of the craft from topocentric radar-station $[X, Y, Z]$ to topocentric prime-meridian $[X_{tf}, Y_{tf}, Z_{tf}]$ [5].

$$\begin{bmatrix} X_{tf} \\ Y_{tf} \\ Z_{tf} \end{bmatrix} = \begin{bmatrix} -\sin \lambda & -\sin \varphi \cos \lambda & \cos \varphi \cos \lambda \\ \cos \lambda & -\sin \varphi \sin \lambda & \cos \varphi \sin \lambda \\ 0 & \cos \varphi & \sin \varphi \end{bmatrix} * \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (10)$$

The station positions in rectangular coordinates are then added from the converted craft coordinates to yield the final rectangular coordinates of the craft in a geocentric reference frame. The last transformation rotates the system an angle theta so that it lines up with the J2000 mean reference frame $[X_{gJ2K}, Y_{gJ2K}, Z_{gJ2K}]$. The angle of rotation, about the Z axis, is determined by the length of time between the current time-stamp of the data and the J2000

epoch. It is the angle between the X axis of the Terrestrial Reference frame, whose X axis is from the geocenter through the intersection between the equator and the prime meridian, and the location of the same X axis at the J2000 epoch. This last conversion places the observations in an inertial reference frame.

$$\begin{bmatrix} X_{gJ2K} \\ Y_{gJ2K} \\ Z_{gJ2K} \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} X_{tf} \\ Y_{tf} \\ Z_{tf} \end{bmatrix} + \begin{bmatrix} X_s \\ Y_s \\ Z_s \end{bmatrix} \quad (11)$$

Additionally, if the craft is in Lunar orbit, the coordinates will be modified again. This time the position of the Moon, given by the PISCES ephemeris in J2000 mean rectangular coordinates, will be subtracted from the craft to yield its selenocentric coordinates $[X_{sJ2K}, Y_{sJ2K}, Z_{sJ2K}]$, this final vector allows for orbital analysis.

$$\begin{bmatrix} X_{sJ2K} \\ Y_{sJ2K} \\ Z_{sJ2K} \end{bmatrix} = \begin{bmatrix} X_{gJ2K} \\ Y_{gJ2K} \\ Z_{gJ2K} \end{bmatrix} - \begin{bmatrix} X_{Moon(gJ2K)} \\ Y_{Moon(gJ2K)} \\ Z_{Moon(gJ2K)} \end{bmatrix} \quad (12)$$

C. First Estimation

The first estimation of the state vector is a crude one using only the first two points in the aggregated data list. The list is ordered by time. The following equations are used to make the initial estimate of the state vector:

$$\begin{bmatrix} X_s \\ Y_s \\ Z_s \\ V_{Xs} \\ V_{Ys} \\ V_{Zs} \end{bmatrix} = \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ (X_2 - X_1)/(t_2 - t_1) \\ (Y_2 - Y_1)/(t_2 - t_1) \\ (Z_2 - Z_1)/(t_2 - t_1) \end{bmatrix} \quad (13)$$

D. Iterator

The Iterator takes in the crude estimate of the state vector and refines it through an iterative method. First, the f and g variables are determined with the Kepler routine. A transformation matrix is created and several quantities are computed and summed over the course of the iterations. The method goes through the whole list of observations, calculating f and g from the beginning for each observation and summing the proper quantities. At the end of the loop, a new state vector is calculated and compared to the old one. If there is appreciable difference, determined by the tolerance limit, the cycle is repeated with the new estimation. If there is no appreciable difference, the vector is said to have converged and is set as the final state vector of the observation data.

E. Residuals

The last part of the ASR determines the residuals based on the converged state vector. The converged state vector is propagated over the time of the arc using the f and g variables. The list of propagated position values represents the path of the craft if it were following along the path of the converged state vector. The list is converted from its current reference frame back to the topocentric-radar station reference frame in range and angle measurements. These measurements are then used to create residuals using the original observations. The residuals were designed to be used by the main navigational program in the generation of covariance matrices.

F. Lunar Ephemeris

The program uses a Lunar ephemeris in order to effect the conversions from geocentric to selenocentric coordinates. The Lunar ephemeris is taken from the PISCES Java package³ and used in the position and velocity conversions.

G. Artificial Data

The majority of data used in the testing of the ASR was artificially generated by Matlab-Java hybrid program using the PISCES software package. A known state vector was input in the appropriate reference frame, Earth or Moon, and was then propagated using an Encke propagator from the PISCES software package. The Encke propagator itself is based on the original Fortran implementation [3]. The data was then converted from rectangular coordinates to proper radiometric measurements and collected in a file. At certain times, it was necessary to add uncertainty to the data. This was done with the Java Apollo Nav Filter package⁴ which was able to apply Gaussian noise to the data produced. Noise was applied to the Azimuth and Elevation angles as well as to the measured range. The application of noise was used in the performance testing of the ASR.

III. Data Processing

The ASR was tested in order to determine its ability to handle and accurately process noisy data. To that end, a number of lists of observations were created for three separate cases, Lunar Orbit, Trans Earth and Earth Orbit to test the effectiveness of the ASR. Additionally, for each of these cases a number of different lists of observations were created each having been subjected to a different level of noise, either range noise, angle noise or both. Each case had a total of 65 observation arcs, each generated from the same vector and each processed with a different level of noise.

The level of Gaussian noise ranged from no noise, i.e. ideal data, to significant noise on the order of 1 million meters for range and .1 radians for angle as the maximum applied noise values.

Noise ID	Noise(Range, Azimuth, Elevation)	Noise ID	Noise(Range, Azimuth, Elevation)	Noise ID	Noise(Range, Azimuth, Elevation)
1 (1R)	(0,0,0) (Control)	23 (2A)	(0, 0.00000025, 0.00000025)	45 (3RA)	(10,0.000000075, 0.000000075)
2 (2R)	(10,0,0)	24 (3A)	(0, 0.00000005, 0.00000005)	46 (4RA)	(10,0.00000001, 0.00000001)
3 (3R)	(100,0,0)	25 (4A)	(0, 0.000000075, 0.000000075)	47 (5RA)	(100,0.00000001, 0.00000001)
4 (4R)	(1000,0,0)	26 (5A)	(0, 0.0000001, 0.0000001)	48 (6RA)	(500,0.00000001, 0.00000001)
5 (5R)	(1500,0,0)	27 (6A)	(0, 0.0000005, 0.0000005)	49 (7RA)	(500,0.00000005, 0.00000005)
6 (6R)	(2000,0,0)	28 (7A)	(0, 0.000001, 0.000001)	50 (8RA)	(500,0.00000008, 0.00000008)
7 (7R)	(3000,0,0)	29 (8A)	(0, 0.000002, 0.000002)	51 (10RA)	(500,0.0000001, 0.0000001)
8 (8R)	(5000,0,0)	30 (9A)	(0, 0.000005, 0.000005)	52 (11RA)	(1000,0.0000001, 0.0000001)
9 (9R)	(10000,0,0)	31 (10A)	(0, 0.0000075, 0.0000075)	53 (10RA)	(1500,0.0000001, 0.0000001)

³ The PISCES Java package was developed at the Johnson Space center for use with Navigational studies. It includes ephemerides for solar bodies as well as several different types of integrators, among them the Encke integrator. It was used here for the Lunar ephemeris information.

⁴ The Apollo Nav filter was a Java package similar to the PISCES package. It contained methods which could be used to manipulate observations. Specifically, it contained a method of applying Gaussian noise to information. This was used in tandem with the generation of observations by applying a noise value to them.

10 (10R)	(12000,0,0)	32 (11A)	(0, 0.00001, 0.00001)	54 (13RA)	(2000,0.0000001, 0.0000001)
11 (11R)	(15000,0,0)	33 (12A)	(0, 0.00005, 0.00005)	55 (14RA)	(2000,0.0000005, 0.0000005)
12 (12R)	(20000,0,0)	34 (13A)	(0, 0.00008, 0.00008)	56 (15RA)	(2000,0.0000008, 0.0000008)
13 (13R)	(40000,0,0)	35 (14A)	(0, 0.0001, 0.0001)	57 (16RA)	(2000,0.0000010, 0.0000010)
14 (14R)	(60000,0,0)	36 (15A)	(0, 0.0005, 0.0005)	58 (17RA)	(6000,0.000001, 0.000001)
15 (15R)	(100000,0,0)	37 (16A)	(0, 0.001, 0.001)	59 (18RA)	(10000,0.000001, 0.000001)
16 (16R)	(150000,0,0)	38 (17A)	(0, 0.002, 0.002)	60 (19RA)	(10000,0.000010, 0.000010)
17 (17R)	(200000,0,0)	39 (18A)	(0, 0.008, 0.008)	61 (20RA)	(50000,0.00001, 0.00001)
18 (18R)	(300000,0,0)	40 (19A)	(0, 0.01, 0.01)	62 (21RA)	(100000,0.00001, 0.00001)
19 (19R)	(400000,0,0)	41 (20A)	(0, 0.08, 0.08)	63 (22RA)	(100000,0.00005, 0.00005)
20 (20R)	(500000,0,0)	42 (21A)	(0, 0.1, 0.1)	64 (23RA)	(100000,0.0001, 0.0001)
21 (21R)	(1000000,0,0)	43 (1RA)	(1,0.0000000025, 0.0000000025)	65 (24RA)	(200000,0.0002, 0.0002)
22 (1A)	(0,0.0000000025, 0.0000000025)	44 (2RA)	(10, 0.000000005, 0.000000005)	66 (25RA)	(1000000,0.01, 0.01)

Table 1 Noise values, identification key and magnitude of noise in meters and radians

The above noise values were applied to each of the three cases of interest to produce a list of observations which could be processed by the ASR. They represent Gaussian style data noise produced by the Apollo Nav Java class. Statistics were then collected from the program output. Root Mean Square (RMS) values for the data were calculated and plotted against the noise level of the data. Additionally, the difference between the state vector and the true vector, the vector which was used to generate the observation data, were plotted against the noise level.

The graphs show, as expected, that the noisier data leads to a less accurate estimation of the state vector. More noise in the data also gives higher Root Mean Square values, as fewer and fewer points of the converged and propagated state vector cross or come near those in the observation list. Furthermore, the difference between the estimated state vector and the true state vector increases as the noise increases this is because the ASR becomes unable to estimate a true state vector if the data becomes sufficiently noisy.

While extremely noisy data does render the ASR ineffective, the data in this investigation has shown that the ASR is extremely robust. It is capable of providing an accurate estimation of the state vector even with very noisy data. Indeed, even with an angular noise of .0001 radians at Lunar distances, at which point even a seemingly small error in angle certainty can lead to very large uncertainty in position, the ASR is capable of providing a good estimate of the state vector. The estimations of the state vector in Earth orbit differ from the true state vector by about only 1.4 kilometers in magnitude, for the Trans-Earth and Lunar cases, the difference between the converged solution and the truth increases but only to around 50-100 kilometers.

Analysis using ideal data showed that the ASR was capable of estimating the state vectors to within 10 kilometers, in Earth orbit, and to within 100-200 kilometers in Trans Earth orbit and Lunar orbit. Such estimations are very accurate considering the methods used and would be effective as an input to an orbit determination process if the state vector were lost or found to be in significant error.

A. Case I: Earth Orbit

This case consisted of a spacecraft orbiting the Earth in close proximity, within about 700 kilometers of the Earth's surface. Because the craft was orbiting so closely to the Earth, at least when compared to the other two cases, the angular noise applied to its observations had a smaller effect. Since the three rectangular coordinates, X, Y and Z, are determined trigonometrically from the range and angle measurements of the radar station, the shorter range of the craft while in Earth orbit means that angular noise will not produce an error of equal magnitude as it would if the spacecraft was in Trans-Earth or Lunar orbit.

While angular noise is not as significant in Earth orbit as it is in Trans-Earth and Lunar orbit, the effectiveness of the ASR was impacted by the range noise to a greater degree. Figures 1 and 2 show that range noise created a more significant uncertainty in the data. The cases where both range and angle noise was present showed a similar trend. Following a period of stable performance, the ASR becomes unstable around case RA20, as can be seen in figures 1 and 2. After this point, the RMS and the difference between the converged state vector and the true state vector begin to increase and destabilize.

The RMS plots show a period of relative stability where the uncertainty stays relatively low indicating a good amount of confidence in the result. As the level of noise increases, the confidence begins to decrease and the RMS increases quickly. Again, the range noise is the factor which affects the prediction most severely. At case 15R, the RMS begins to climb rapidly indicating a greater uncertainty in the result. This is consistent with the calculations of the difference between the estimated state vector and the true state vector which begin to diverge increasingly close to that point. The cases for angular noise begin to increase significantly in RMS around case 17A, and the compound cases around noise level 20RA.

B. Case II: Trans-Earth Vector

The cases run with the Trans-Earth Vector are ones at a significant distance from the earth around 100,000 kilometers heading towards Earth to be captured in orbit. These vectors were taken from Design Reference Trajectories for future Constellation missions on their return from Lunar orbit.

Because of the great distance from the Earth, angular noise was more significant for this case than for the Earth orbit case. The general trend remains, however. For the initial noise levels, the ASR produced accurate and reliable data, see figures 3 and 4. The estimates of the state vector were close to the actual vector, and the RMS aggregate was relatively low until the noise became extreme. The ASR continued to produce good estimations until case 20RA for the range and angle noise tests, case 16A for the angle noise tests, and case 13R for the range noise tests. The RMS results were similarly stable until these cases After which they began to grow rapidly, signaling a loss of accuracy and effectiveness of the ASR.

C. Case III: Lunar Orbit

The Lunar orbit cases were run with a vector approximately 300 kilometers from the surface of the moon. The methods of observation generation and the addition of noise followed the steps described in the previous sections. See figures 5 and 6.

Given the distance from the Earth, angle noise played a significant role in the accuracy of the estimated state vector. While the ASR was still able to tolerate a good number of noise levels, the general accuracy of the estimation decreased. Whereas the estimations of the state vector were off by about 10-50,000 meters in Earth orbit, this increased to around 100,000 meters in Lunar orbit. Still, the ASR was stable for noisy angle data until case 15A, for noisy range data until case 19R and for noisy compound data until case 18RA.

Again, the RMS aggregate was a good support statistic. It remained relatively constant before rapidly increasing around the points where too much noise was present.

IV. Standard Noise

In addition to the analysis of the three different orbital vectors with varying degrees of noise, an analysis was conducted on data with standard noise, that is the noise expected to be realistically present in the data collected by actual radiometric observation. This noise amounted to 1 meter range noise and $3.49 \cdot 10^{-4}$ radians of Gaussian angle noise. These values are well within the capabilities of the ASR to effectively, and stably, estimate a state

vector. Additionally, the ASR was able to effectively process data with far greater noise than what is realistically likely [7].

This analysis was focused on finding the best number of data points to be used by the ASR in its calculations. Data was generated as before with the artificial generator, and the standard noise values were applied. However, this data was collected at a certain realistic rate. That is, if the data spanned 40 minutes, not every second would yield a new data point. Rather, the rates for observation collection in Earth orbit were one observation every 20 seconds, and those for Trans Earth and Lunar orbits were one observation every 60 seconds. These are realistic values based on historical mission operation performance.

A list of observations was then generated with these values. The ASR was tested with a different number of points. The fewest possible points that could be input is three, and this value acts as the lower limit. The maximum number of points the ASR can process, however, is limited by the observation rate. While the ASR can take in and process any number of observation points, it cannot carry out any calculations if the time period exceeds 60 minutes. This is a limitation of the function which calculates the f and g values and more fundamentally a limitation of the methods used by the ASR.

Since the ASR depends on a simplistic 2-body estimation of an orbit, it uses the simply calculated f and g variables. These two variables are calculated through an iterative method which takes in position and velocity coordinates as well as the time over which to propagate those values. The greater the period of time, however, the less dependable the result given, since realistically an orbiting spacecraft is under the influence of more forces than just the central body. The limitation here is that the method generating the f and g values is incapable of converging on a result if the time period is greater than 60 minutes. Thus, while any number of points may be entered into the ASR for calculation and analysis, the time over which these observations have been collected cannot exceed 60 minutes.

An additional parameter was used for determining the effectiveness of the ASR. It is a comparison of the semi-major axis of orbit derived from the estimated state vector and compared to the semi-major axis derived from the actual state vector. Since the semi-major axis is dependent on the velocities of the state vector, it adds a valuable dimension of comparison, indicating whether or not the velocities which are being estimated are relatively correct.

A. Case I: Earth Orbit

The plots of the comparison between the true state vector and the estimated state vector showed a relatively unchanging estimation as the number of data points increases. See figures 7-9. However, examining the RMS plot relative to the number of data points, a pattern emerges wherein the RMS is lowest midway between the fewest points and the most points which were used by the ASR. Between 40 and 80 data points, the RMS value reaches its lowest point, meaning there is the greatest confidence in the fit produced by the ASR.

Furthermore, the plot comparing the semi major axes shows a progressive approach of the true semi-major axis as the number of points being processed increases. The fewer points that were processed, the worse the approximation of the semi-major axis, and therefore the worse the approximation of the velocity and position was. However, the approximated semi-major axis is very close to the true semi-major axis at the time when the RMS was also near its lowest point.

The best number of points, and thereby the best length of observation, for the Earth orbit case is between 40 and 80 this translates to between 13 and 27 minutes of observation at the rate of one data point every 20 seconds. Additional points do not show any increase in the accuracy of the estimated state vector, though they do show an increase in the accuracy of the estimated semi-major axis. The value rapidly converges and is reasonably close to the true semi-major axis at around 40 data points.

B. Case II: Trans-Earth Vector

The plots comparing the estimated state vector to the true state vector shows a slow progression to the true vector as the number of observations increases. See figures 10-12. The RMS plot shows a spike in the error as the number of points increases up to 10 points after which the error diminishes somewhat and stabilizes. It increases mildly only at 40 observations and 54 observations. The progression of the semi-major axis plot also shows a progressive trend towards the true semimajor axis. However, this trend is less steady than that of the Earth orbit Case, since the relative noisiness of the data has increased with the distance from the radar station.

However, between 30 and 40 observations, the RMS stays relatively stable. It does not spike sharply or behave unpredictably in that region. Additionally, around the same number of observations the state vector approximation has reached a plateau and is relatively stable. The plot of the semi-major axis shows that after around 30 data points, additional points do not appreciably converge the calculation of the semi-major axis to the truth. Thus the region of stability for the Trans-Earth case in terms of data points is between 30 and 40 data points. In this region, the RMS is stable and lower than the peak RMS. The state vector estimation is stable and the semi-major axis, and therefore the velocity components of the state vector, is stable.

C. Case III: Lunar Orbit

The plots comparing the estimated state vector to the true state vector show a somewhat rapid progression towards a relatively accurate value for the state vector. See figures 13-15. However, the plots eventually reach a stable point where the estimated state vector does not appreciably close to the truth as more points of data are added. A similar trend is seen in the plot of the semi-major axis where, following an initially quick approach to the true axis, the trend becomes almost asymptotic. This shows no great movement towards the true semi-major axis as more points are added. The plot of the RMS, shows a minimum RMS around 20 data points. The area around that RMS, however, is unstable. A stable region in the RMS is found between 28 and 42 data points. It is a region of both stable and relatively low RMS. After this region the RMS begins to rise as with the Earth Orbit case. The state vector difference plot shows a similar region. The addition of new data points yields a better estimation of the state vector until a plateau-like region between 28 and 43 data points, coinciding with the stable region in the RMS plots. The graph of the semi-major axis shows an increase in the accuracy as the number of data points increase until about 30 data points, after which additional data points do not drastically increase the accuracy of the semi-major axis.

Thus the optimal number of data points for a Lunar orbit is again between 30 and 40 data points. In this area, the RMS is low and stable and the state vector approximation is accurate.

D. Iterations to Convergence

The final analysis of the ASR is how quickly it converges on a solution. For the most part, the ASR quickly converges to a solution and exits out of the convergence loop, since it meets the tolerance requirement. The vast majority of tested cases indicate that most cases, including those with high noise or those with significant observations, converge within 40 iterations most of these cases converge around 18-24 iterations. Only a few outliers required iterations above 40 and none required more than 150 iterations, the limit. Thus all calculated estimations of the state vector converged within the maximum allowed iterations, the vast majority of them converging within a fraction of the maximum.

The number of iterations at which many of the cases converged, however, is almost double, the number of iterations allowed in the original Apollo code. Likely, this was because of the limits on computer power and memory. It is likely that a maximum of 10 iterations would suffice for the refinement of most guesses. Because a state vector approximation can be very close to its converged value but still not be considered converged, such a limit was appropriate for the Apollo needs. Current computer technology allows for a more intensive refinement of the state vector approximation, allowing the cap on iterations to remain at approximately 150.

V. Conclusion

The ASR is a robust algorithm, capable of generating accurate and timely approximations of a craft's state vector even with very noisy data and few points. It is effective and stable.

The tradeoff for speed and robustness is, of course, the absolute accuracy of the final solution. After the initial guess was converged to a value, that value would remain a certain distance from the truth even as more data points were added for processing. This phenomenon is a limit of the way in which the ASR calculates the state vector. Since the state vector is calculated initially as a rough approximation which is then iterated into a more accurate approximation, the way in which the approximated state vector is propagated within the ASR is limited. The only force which the ASR takes into account is the point mass gravitational force of the central body. Third body perturbations, drag, solar pressure and the like are not considered in this approximation. Furthermore, the way in which the ASR approximates forces under the influence of gravity and the way in which it propagates a craft under such influences is limited and essentially unreliable as the craft is propagated over longer periods of time with the final limit being 60 minutes. These limitations create a limit to the accuracy of the ASR in determining the state

vector of the craft. However, the ASR was designed only to provide a good approximation of the true state vector. That approximation would then be continually refined using finer force models and including drag, pressure, thrust etc. Despite the limitations of the models used by the ASR, it is effective in carrying out its prescribed function.

VI. Future Work

The implementation and development of the ARS merits further study. An analysis of the performance of different simulations of the gravitational, and perhaps third body forces, would be important. Such an analysis could counterbalance the increase in the accuracy of the ARS with the increase in computational requirements. The original Apollo software had to utilize models of such limited accuracy because their main limit was computer speed and power. Today, however, computers are vastly more powerful and allow for a greater fidelity with regards to gravitational and force models.

Additionally, a more comprehensive analysis of the levels of noise and their relation to the performance of the ARS would be beneficial. Including more noise values in the neighborhood of the ‘critical points’ would resolve the performance boundaries of the ARS.

Appendix

Note: The levels of noise on the plots are listed as integers, usually from 1 to 20 or so. The type of noise is listed at the top of the plots, Range, Angle, Range and Angle, or Standard Noise, use this to determine which entry on the Noise table the plot is using. For example, a plot of RMS versus Range and Angle noise for a Lunar orbit with a critical point at Noise Level 18 will be referring to Noise Level 18RA.

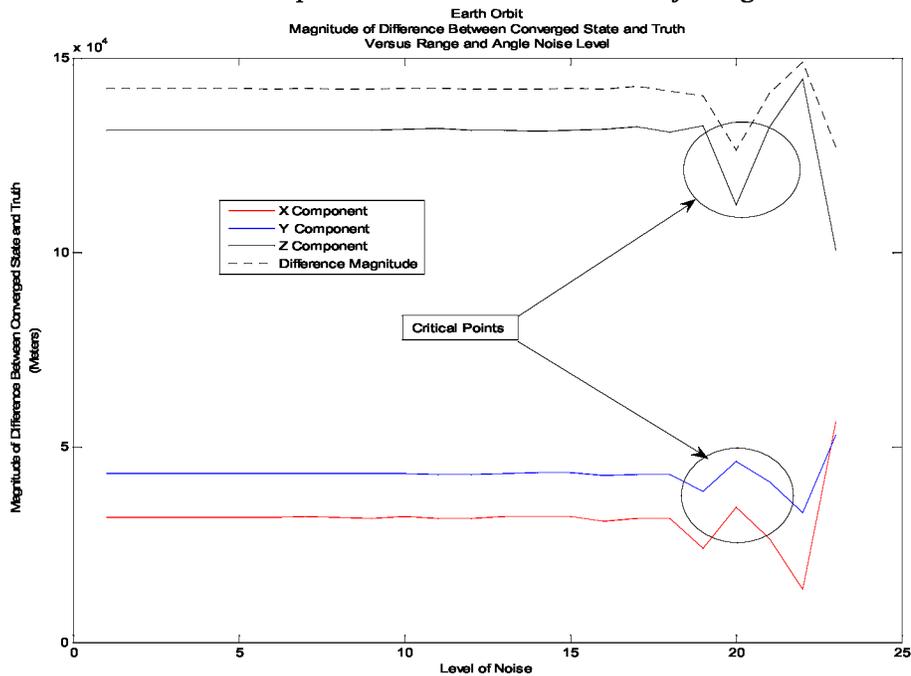


Figure 1: The critical noise level in Earth orbit for the Range and Angle noise case is noise level 20RA. After this point, the State Vector difference begins to ‘destabilize’. This, combined with the increasing RMS, decreases the confidence in the solution and thus marks the limit of the ARS’ capability.

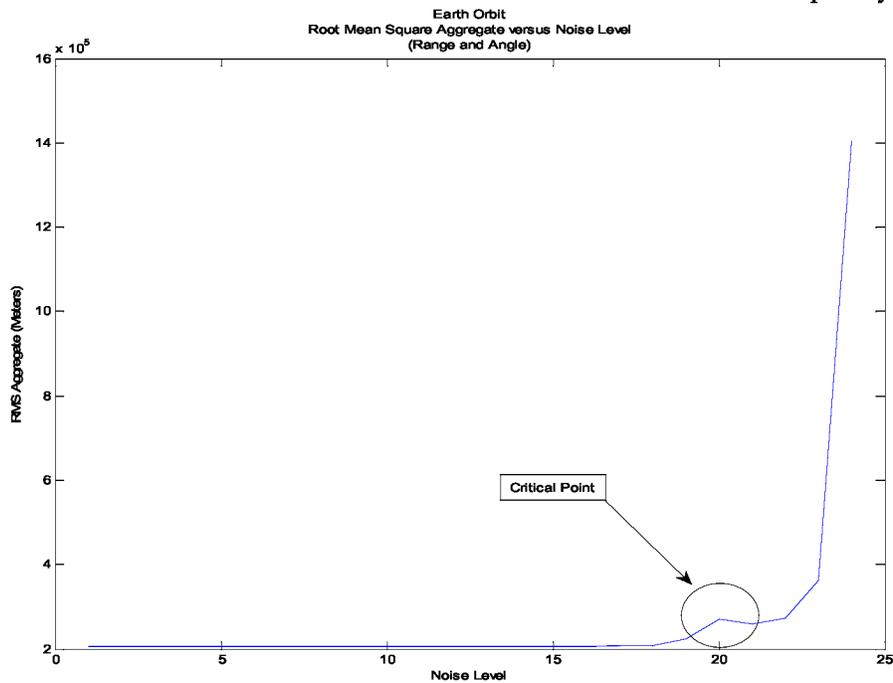


Figure 2: The critical noise level in Earth orbit is illustrated by the RMS plot as well. After noise level 20RA, the RMS rapidly increases. Thus the confidence in the curve fit, and therefore the solution, decreases proportionately. Noise Level 20RA is the critical point for Earth Orbit for Angle and Noise range.

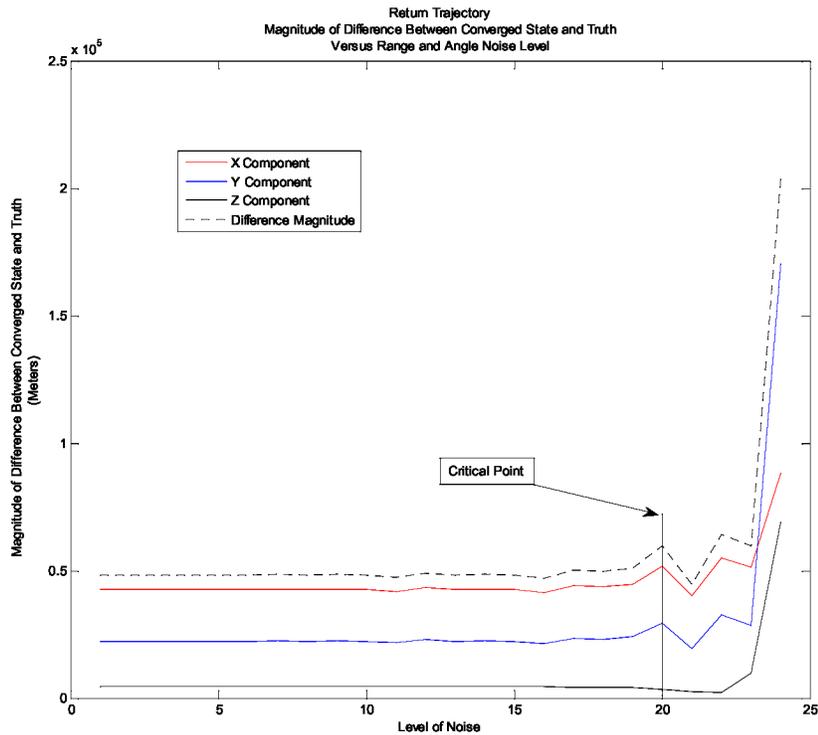


Figure 3: The Return Trajectory critical point for Range and Angle noise is the same as that for Earth Orbit. Above Noise Level 20RA, the state vector estimation begins to greatly diverge from the truth, the plot shows this as an increase in the difference between the converged state vector and the truth.

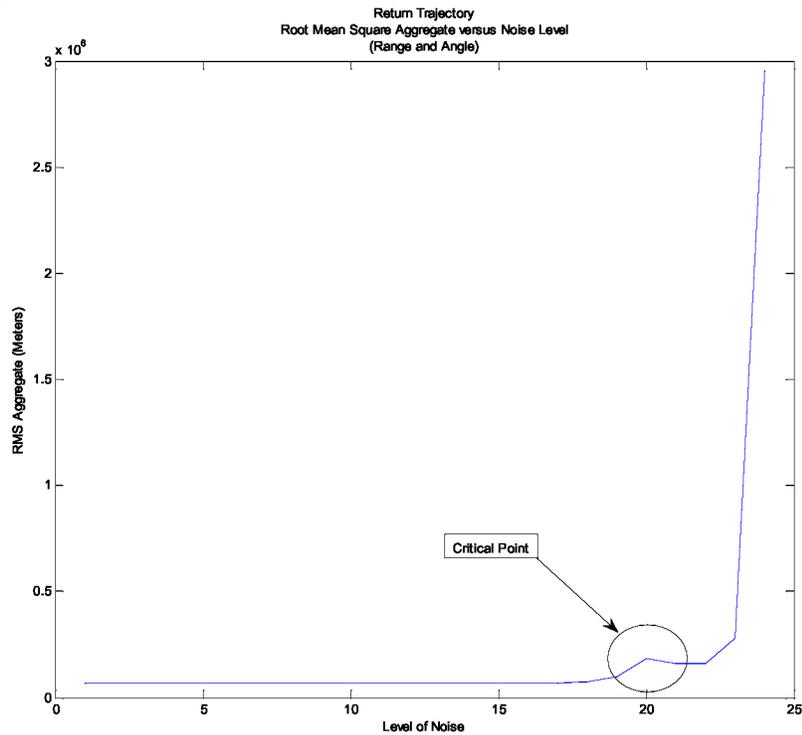


Figure 4: The Return Trajectory critical point for Range and Angle noise is supported by the RMS plot. After the same critical point, 20RA noise, the RMS begins to rapidly climb, and the confidence in the solution deteriorates proportionally.

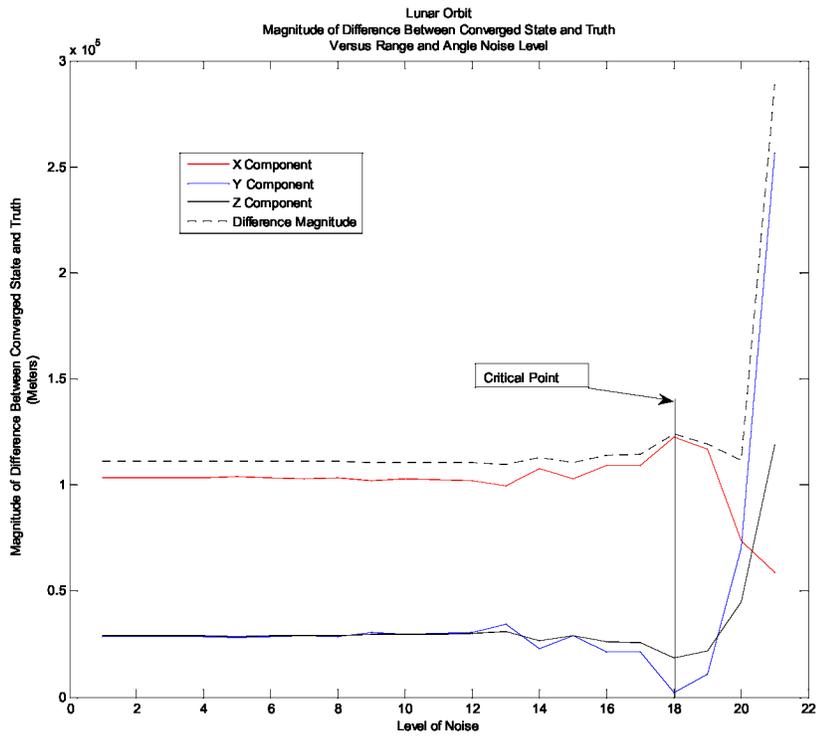


Figure 5: The Lunar Orbit case for Range and Angle noise shows an earlier drop in reliability at 18RA, instead of 20RA. After this critical point, the estimation of the state vector begins to fluctuate wildly.

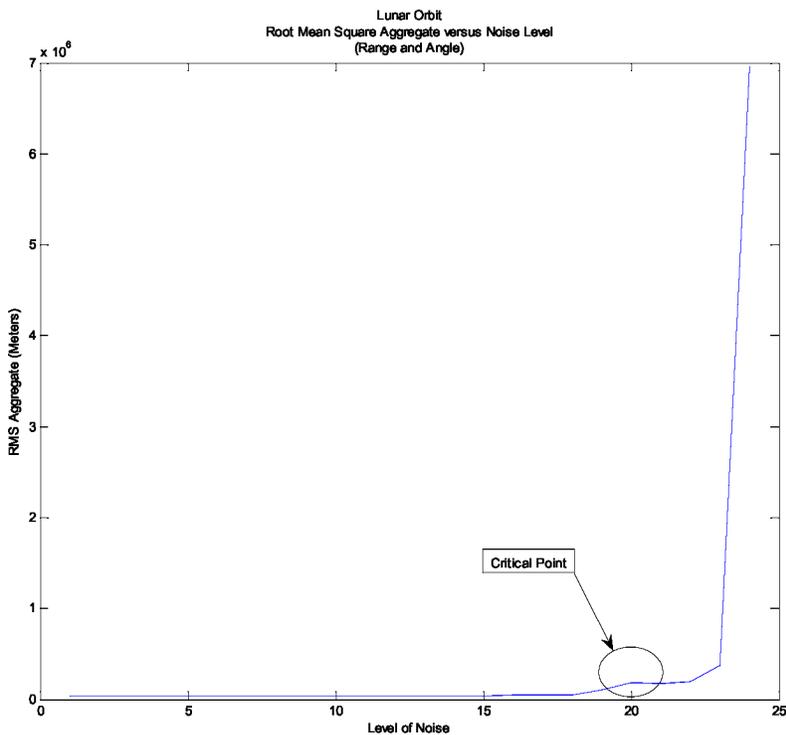


Figure 6: The RMS plot for the Lunar Orbit case, however, shows a similar behavior to the other two cases. The critical point here is still noise level 20RA, though the state vector estimation suffers before this point at noise level 18RA.

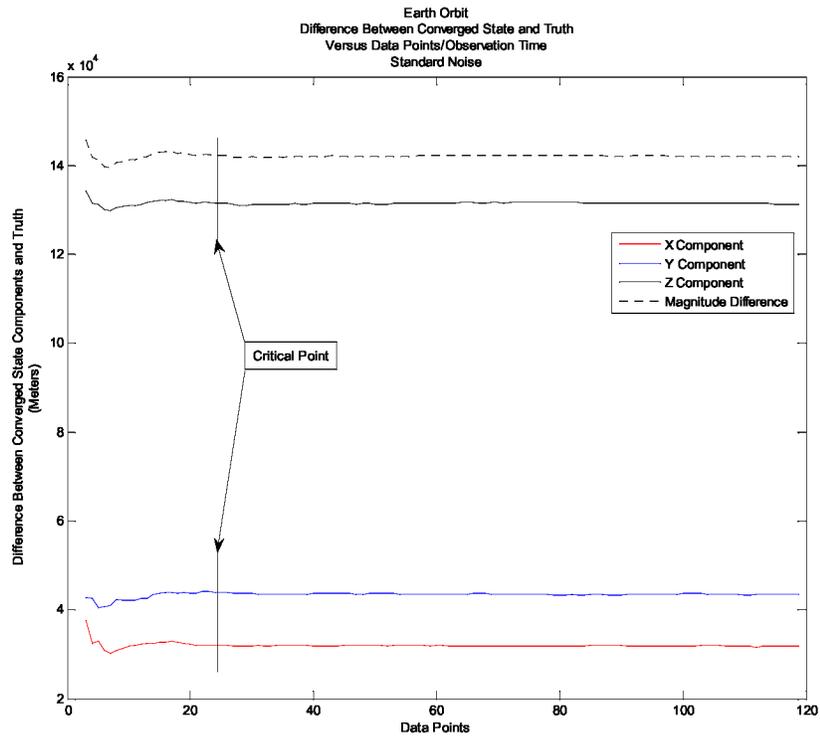


Figure 7: After 22 data points, the accuracy of the state vector estimation ceases to improve. After the critical point at 22 data points, the estimation does not become better, however the confidence in the solution may decrease as more and more data points are used and the RMS climbs.

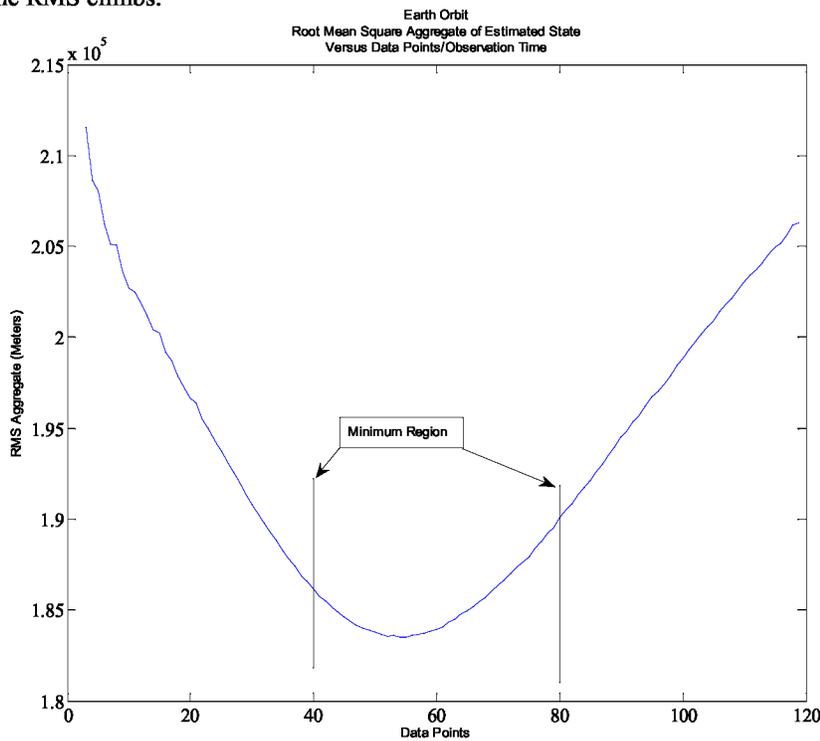


Figure 8: The minimum RMS occurs between 40 and 80 data points. Below 40 data points, there are too few points to make a good estimation of the state vector, which leads to a high RMS. After 80 data points, there are too many data points which act to increase the RMS. Greatest confidence in the solution is between 40 and 80 data points.

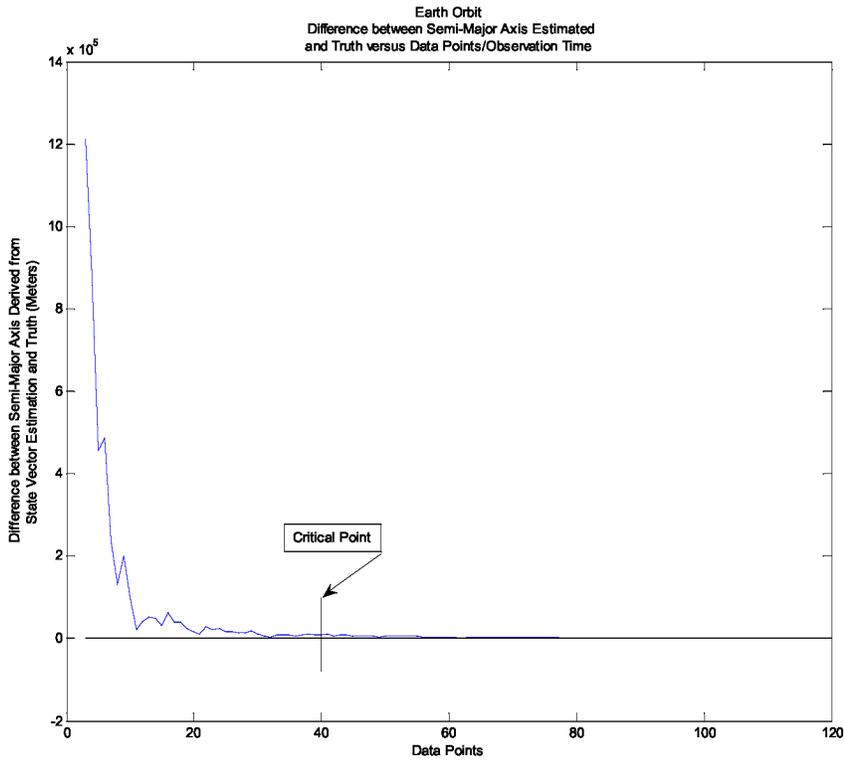


Figure 9: The estimation of the semi-major axis converges rapidly to the truth as the number of data points increase. 40 data points represent a critical point because even if more data points are used, the semi-major axis does not converge as rapidly towards the truth. More data points would decrease the confidence in the solution without providing an increased accuracy in the semi-major axis.

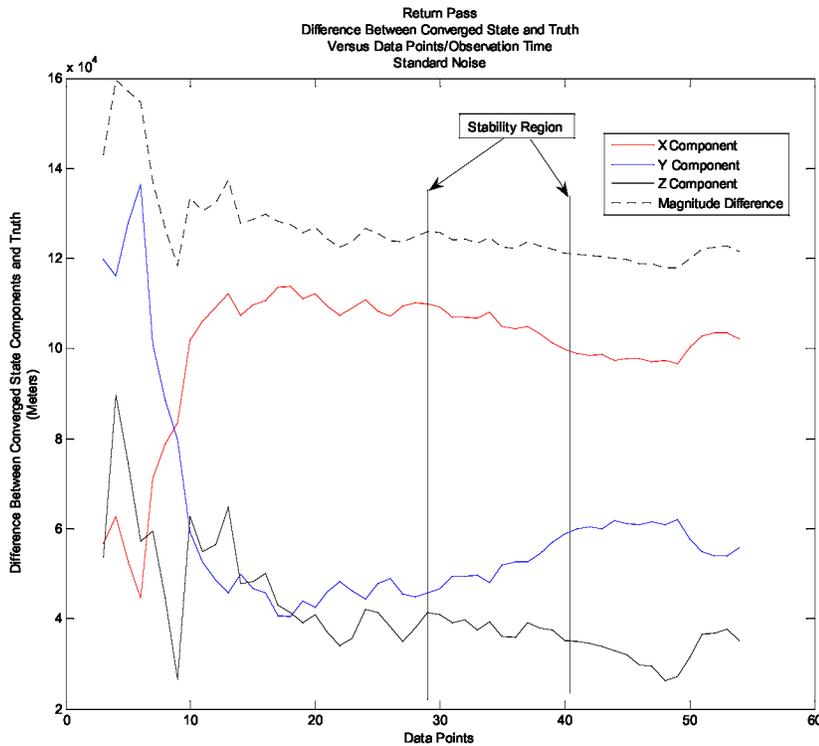


Figure 10: As more data points are used to estimate the state vector for the Return Trajectory case, the estimation accuracy improves. However, between 30 and 40 data points, the estimation has become stable. Fewer data points and the estimation becomes less accurate, any more and the confidence in the solution decreases.

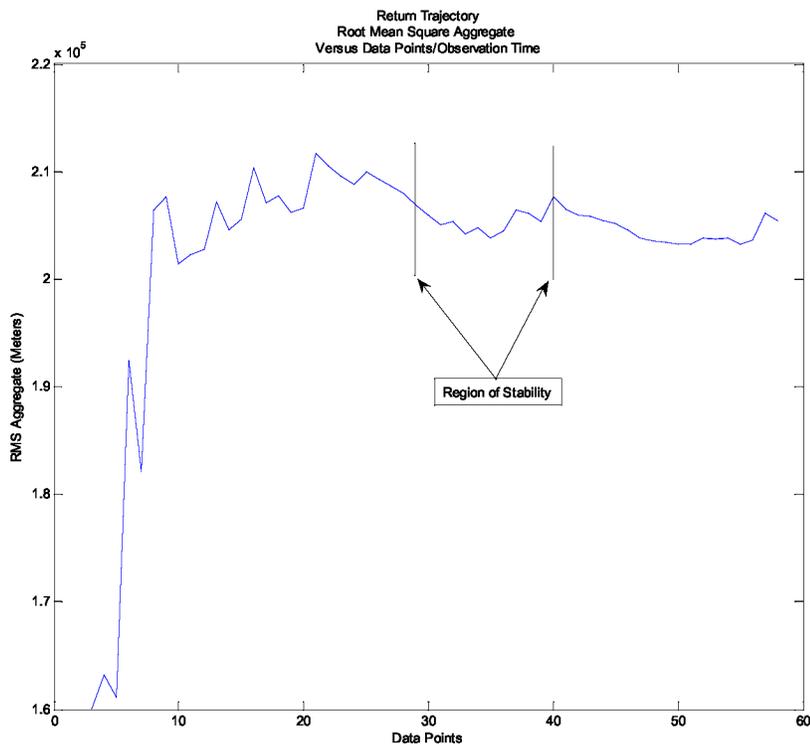


Figure 11: The RMS increases as the number of data points increase until it reaches a maximum. Between 30 and 40 data points, there is a stable region where the RMS does not greatly fluctuate. After this region, the RMS begins to increase again.

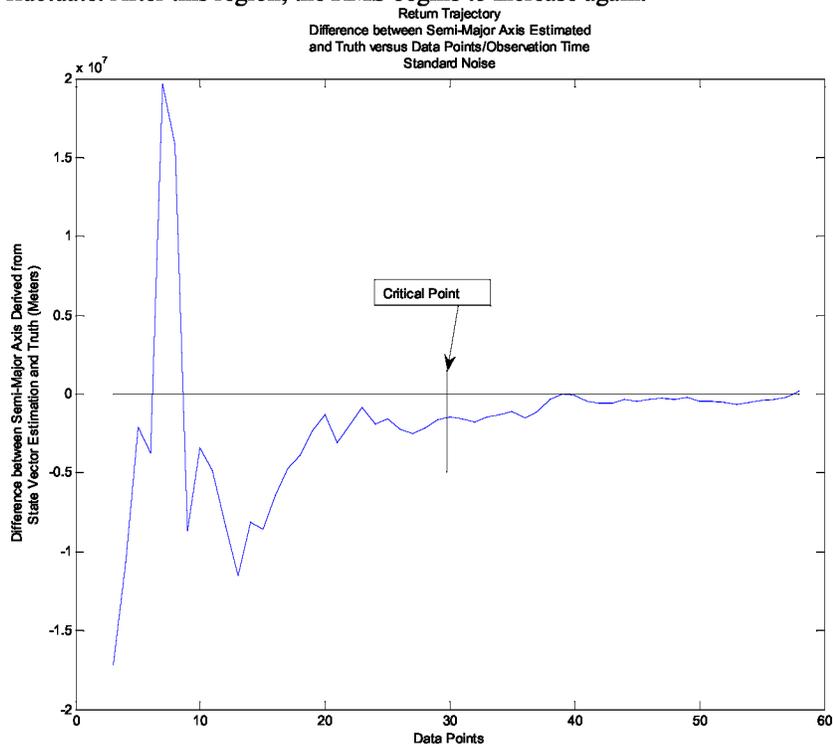


Figure 12: The semi-major axis estimation increases in accuracy as the number of data points increases until 30 data points. At that point, further data points do not produce a significantly more accurate estimation of the semi-major axis.

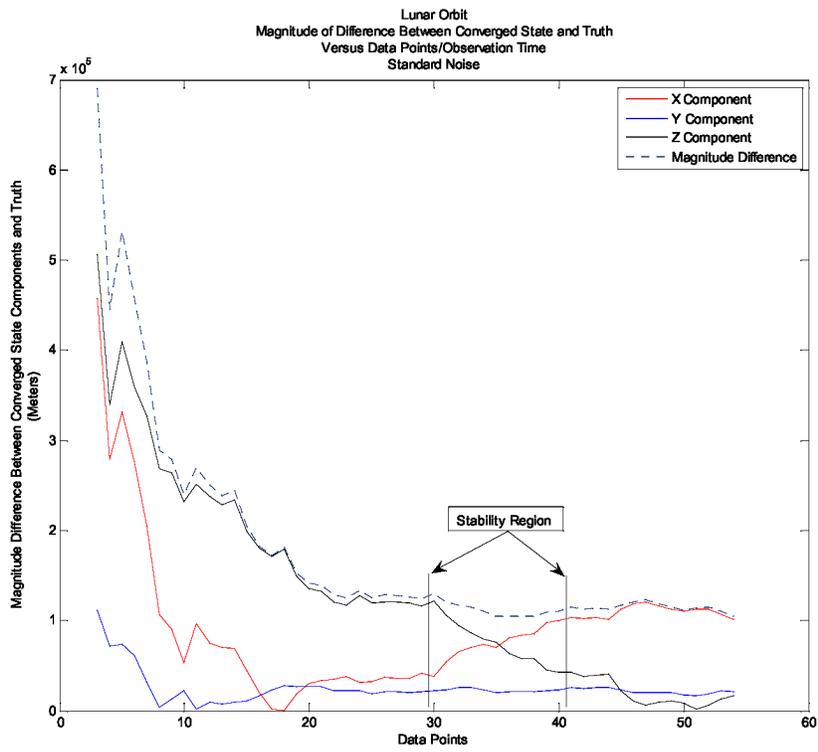


Figure 13: As more data points are used to estimate the state vector for the Lunar Orbit case, the estimation accuracy improves. However, between 30 and 40 data points, the estimation has become stable. Fewer data points and the estimation becomes less accurate, any more and the confidence in the solution decreases.

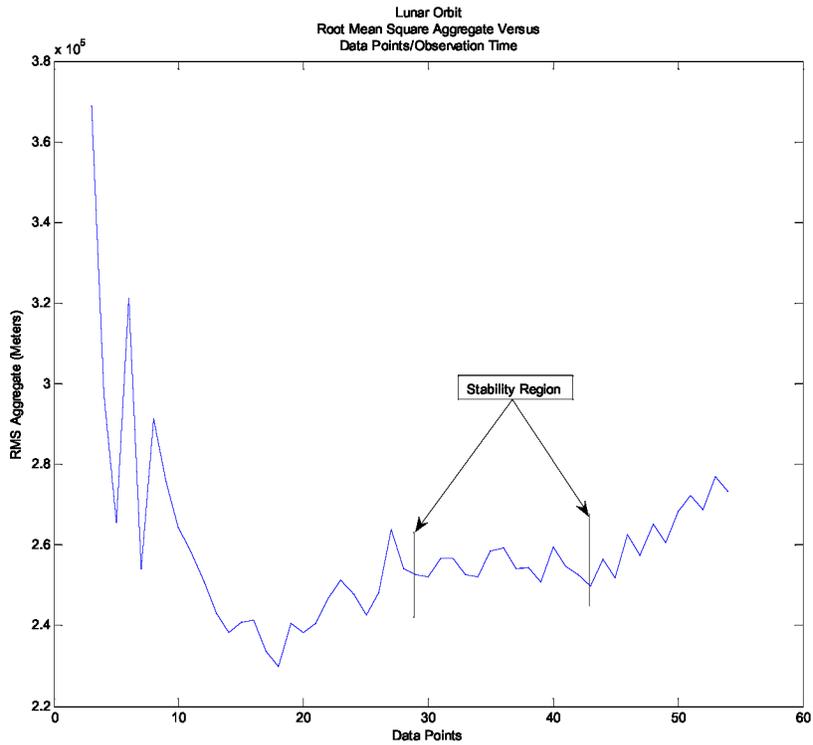


Figure 14: The RMS decreases as the number of data points increase until it reaches an unstable minimum. Between 30 and 40 data points, there is a stable region where the RMS does not greatly fluctuate. After this region, the RMS begins to increase again.

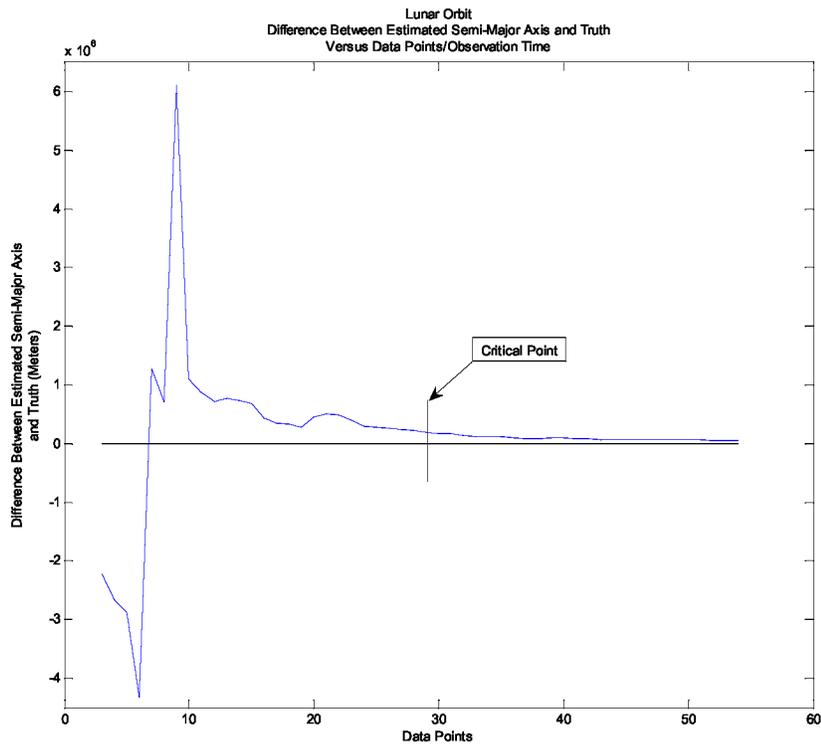


Figure 15 The semi-major axis estimation increases in accuracy as the number of data points increases until 28 data points. At that point, further data points do not produce a significantly more accurate estimation of the semi-major axis.

References

- [1] Bate, R. R., Mueller, D. D., White, J. E., *Fundamentals of Astrodynamics*, Dover Publications, Toronto, Ontario, 1971.
- [2] Lunar Orbit Determination Batch Processor, Software Package, Ver. 1.2, NASA, Johnson Space Center, Houston, TX, 2009.
- [3] PISCES, Platform Independent Software Components for the Exploration of Space, Software Package, Ver. 1.1, NASA, Johnson Space Center, Houston, TX.
- [4] Soler, T., Eisemann, D. W., "Determination of Look Angle to Geostationary Communication Satellites," *Journal of Surveying Engineering*, Vol. 120, No. 3, 1986, pp. 115, 127.
- [5] Schiesser, E. R., deVezin, H. G., Savely, R. T., Oles, M. J., "Basic Equations and Logic for the Real Time Ground Navigation Program for the Apollo Lunar Landing Mission," NASA MSC Internal Note No. 68-FM-100, 1968.
- [6] Vallado, D. A., McClain, W. D., *Fundamentals of Astrodynamics and Applications*, 2nd ed., Microcosm Press, El Segundo, CA, 2004.
- [7] Wertz, J. R., Larson, W. J., *Space Mission Analysis and Design*, 3rd Edition, Microcosm Press, El Segundo, CA, 1999.