FalconSAT 4 Orbital Estimation Using Kalman and Least-Squares Techniques

Todd V. Small
United States Air Force Academy, Colorado Springs, CO, 80841

This paper investigates the position and velocity estimation capabilities of Kalman and Least-Squares filters during small satellite orbital maneuvers. This research topic originated with the United States Air Force Academy’s orbital maneuvering goals for FalconSAT 4, which was designed to carry four cold gas thrusters with enough propellant to provide 150-350 m/s of ΔV. Kalman and Least-Squares algorithms are developed and then applied to orbital estimation. A truth orbit is designed using gravitational effects, the $J_2$, $J_3$, and $J_4$ zonal harmonics, and a commanded change in RAM-direction velocity. The filters then assume a slightly simpler model including only gravitational effects and the $J_2$ zonal harmonic, which simulates the real-world disparity between true motion and modeled motion. White Gaussian Noise is added to the truth model propagation data to imitate satellite position and velocity sensor measurements. Both filters then process the measurement data, and the performance results are compared to the truth orbit in a series of estimation error plots. The Kalman filter demonstrates several benefits over the Least-Squares filter including greater tuning flexibility, better performance with accurate sensor measurements, superior convergence after poor initial guesses, and significantly less required computing power. The Least-Squares filter demonstrates several advantages over the Kalman filter such as better precision with accurate modeling assumptions, superior propagation accuracy when sensor measurements cease, and much quicker settling times after orbital maneuvers.

Nomenclature

\begin{itemize}
\item $X$ = state vector
\item $x$ = satellite position in the $x$ direction (Earth-Centered Inertial coordinate frame)
\item $y$ = satellite position in the $y$ direction (Earth-Centered Inertial coordinate frame)
\item $z$ = satellite position in the $z$ direction (Earth-Centered Inertial coordinate frame)
\item $\dot{x}$ = satellite velocity in the $x$ direction (Earth-Centered Inertial coordinate frame)
\item $\dot{y}$ = satellite velocity in the $y$ direction (Earth-Centered Inertial coordinate frame)
\item $\dot{z}$ = satellite velocity in the $z$ direction (Earth-Centered Inertial coordinate frame)
\item $r$ = distance from the center of Earth to the satellite
\item $J_2$ = second zonal harmonic of Earth’s gravitational force
\item $J_3$ = third zonal harmonic of Earth’s gravitational force
\item $J_4$ = fourth zonal harmonic of Earth’s gravitational force
\item $\mu$ = Earth’s gravitational parameter
\item $R_e$ = radius of the Earth
\item $\sigma_1$ = standard deviation of satellite position measurement
\item $\sigma_2$ = standard deviation of satellite velocity measurement
\item $R$ = measurement noise covariance matrix
\item $\bar{Q}$ = process noise covariance matrix
\item $q$ = arbitrary tuning constant within $Q$ matrix
\item $dt$ = propagation time step for filter
\item $F$ = matrix of state partial derivatives (Jacobian matrix)
\item $\bar{X}$ = propagated state vector
\item $\hat{X}$ = updated state vector
\item $\Phi$ = state transition matrix
\end{itemize}

1 Undergraduate Student, USAFA Dept. of Astronautics, 2354 Faculty Drive, AIAA Member
\[ I = \text{identity matrix} \]
\[ P = \text{covariance matrix} \]
\[ K = \text{Kalman gain matrix} \]
\[ H = \text{observation matrix} \]

I. Introduction

The topic for this research project originated with the orbital maneuvering goals for FalconSat-4 (FS-4). Before funding was cut for this mission, FS-4 was designed to carry four cold gas thrusters with enough propellant to provide 150-350 m/s of ΔV. Using the thrusters, the satellite would have performed commanded orbital maneuvers, changing the COEs of the orbit and altering the satellite’s time over target (TOT) by 10-30 minutes. Although orbital maneuvering predictions can be calculated from simulations and thruster testing on the ground, many unmodeled factors can cause actual in-orbit performance to deviate from tested or theoretical performance. For instance, inaccurate mounting procedures or severe launch conditions can cause thruster misalignments that alter the thrust vector and change the outcome of attempted orbital maneuvers. In order to accurately execute commanded maneuvers, the following question requires answering – “How can we ensure that our thrusters accomplish the orbital maneuvers as planned?” Orbital estimation is the first step toward the solution.

II. Orbital Estimation Conceptual Background

In order to characterize the true in-flight effects of a thruster burn, the position and velocity of the satellite must be determined before, during, and after the burn. Two basic tools exist which contribute knowledge of a satellite’s position and velocity: orbital mechanics and sensor measurements. Equations of orbital motion are derived from the laws of Newtonian physics, and these equations describe the basic motion of a satellite as it moves through its orbit. However, as with any system model, these equations can never incorporate every factor or disturbance affecting satellite motion. Using a simple illustration, a falling object accelerates downward according to this basic equation:

\[ h = \frac{1}{2} at^2 + vt + h_0 \]  

(1)

However, if this model only accounts for gravity, using 9.81 m/s\(^2\) as the value for acceleration, this equation would not exactly predict a falling object’s height. Many other factors would affect the trajectory, such as drag, wind, and even minute forces such as solar pressure or magnetism. The complexity of the space environment prohibits perfect modeling in much the same way.

Satellite position and velocity can also be determined from on-board sensor measurements such as GPS, as well as ground-based measurements from radar tracking stations. However, sensor measurements contain inherent error from various noise sources. In addition, ground-based tracking information such as NORAD TLEs cannot provide the continuous real-time orbital data necessary to track a thruster burn, unless the satellite happens to be overhead. Since both models and measurements provide orbital information with limited accuracy, Kalman and Least-Squares filters combine data from each to significantly increase the total accuracy of orbital estimation.

A. Kalman Filtering Basics

The design of a Kalman filter begins with the selection of the states to be estimated. For orbital estimation, the states are the six components comprising the satellite position and velocity vectors. A set of governing equations and guesses for the initial conditions...
of each state must then be chosen. Kalman filters take the initial conditions of the states, propagate them forward in
time according to the assumed model, compare the propagated states to measurements, calculate and store the error
between the state propagation and the state measurements in a covariance matrix, and then update the states by
accounting for this error. Once the states are updated, they are propagated again according to the model until
another measurement is taken, when the estimated states are updated once more. Figure 1 provides a graphical
representation of this process. Kalman filtering loops through this process and tracks satellite motion as
measurements are taken, requiring minimal data storage compared to other orbital estimators. The filter uses a
measurement once, updates the covariance matrix calculations, and then discards that measurement.

Kalman estimation demonstrates a high degree of flexibility because Kalman filters can be “tuned” to rely more
heavily on either the model or the measurements during the state update process. The Kalman filter algorithm
includes two matrices designated as R and Q which account for measurement error and system modeling error,
respectively. In a real system, true values for sensor noise error and modeling error will never be known, but by
“tuning” the R and Q matrices to these sources of error as precisely as possible, the accuracy of the Kalman
estimation can be optimized.

B. Least-Squares (Batch) Filtering Basics

The design of a Least-Squares (LSQ) filter also begins with the selection of the states to be estimated. Again, for
orbital estimation, the states are the six components comprising the satellite position and velocity vectors. Next, the
governing model and the initial conditions of each state must be chosen. Unlike Kalman filters, the performance of
LSQ filters always hinges heavily on the accuracy of the governing model. For this reason, it is imperative that the
equations of motion include as many disturbance factors as possible. While Kalman filters continuously update and
take into account new measurements as they arise, LSQ filters take an entire batch of measurements that have
already been collected and fit them to the defined model. Essentially, an LSQ filter works like a powerful regression
line tool. It simply takes a set of initial conditions, propagates each state forward according to the model, calculates
the error between the propagated model and the batch of measurements, and then adjusts the initial conditions until
the model best fits the measurements.

To illustrate this process with a simple example, consider a batch of measurements roughly arranged in a
sinusoidal shape. If an LSQ filter was developed assuming a linear model, the filter would converge when it finds
the straight line which best fits the data. The filter would minimize the estimation error, but the poor model choice
would still create large inaccuracies. However, if a sinusoidal model was assumed, the LSQ filter would converge
on a function with a much closer fit, simply because the original model was more appropriate.

III. Orbital Estimation Algorithms

Before designing the Kalman and LSQ estimators, it was first necessary to compile data for a truth orbit, against
which to compare the estimator performance and from which to simulate measurements. Even though the “truth”
data will never actually be known in a real application, filter strength can be tested by numerically integrating one
particular model to simulate “real” data, and then using a slightly different model within the filter algorithms to
simulate modeling error. For the following scenarios, the Kalman and LSQ filters assumed an orbital model that
accounts for two-body gravitational effects and the J₂ zonal harmonic. The states and the filtering model were
defined as follows:

\[
\begin{align*}
\dot{x} &= u \\
\dot{y} &= v \\
\dot{z} &= w \\
\dot{\dot{x}} &= -\frac{\mu}{r^3} x - \frac{1}{2} J_2 \left( \frac{\mu}{r^3} \right)^2 \left( 1 - \frac{5}{2} \frac{z^2}{r^2} \right) x \\
\dot{\dot{y}} &= -\frac{\mu}{r^3} y - \frac{1}{2} J_2 \left( \frac{\mu}{r^3} \right)^2 \left( 1 - \frac{5}{2} \frac{z^2}{r^2} \right) y \\
\dot{\dot{z}} &= -\frac{\mu}{r^3} z - \frac{1}{2} J_2 \left( \frac{\mu}{r^3} \right)^2 \left( 3 - \frac{5}{2} \frac{z^2}{r^2} \right) z
\end{align*}
\]

The measurement and truth files were generated from equations which also modeled J₃ and J₄ zonal harmonics. For
the sake of comparison with Dr. Hashida’s own data, the truth orbit had an altitude of 800km, an inclination of
98.6°, and an eccentricity of 0.0015. The data points were developed through Runge-Kutta 4 numerical integration,
and sensor measurement error was imitated by adding white Gaussian noise to the truth data points.
A. Kalman Filter

Designing the Kalman filter began with initializing the covariance matrix with a guess for initial error and then setting the R and Q matrices. Since the measurement files for this test were artificially created with known standard deviations, a perfectly tuned R matrix was assumed for simplicity’s sake. The Q matrix was defined with a format provided by Dr. Hashida.

\[
R = \begin{bmatrix}
\sigma_1^2 & 0 & 0 & 0 & 0 & 0 \\
0 & \sigma_1^2 & 0 & 0 & 0 & 0 \\
0 & 0 & \sigma_1^2 & 0 & 0 & 0 \\
0 & 0 & 0 & \sigma_2^2 & 0 & 0 \\
0 & 0 & 0 & 0 & \sigma_2^2 & 0 \\
0 & 0 & 0 & 0 & 0 & \sigma_2^2 \\
\end{bmatrix}
\]

\[
Q = \begin{bmatrix}
\frac{1}{2} q^2 dt^2 & 0 & 0 & \frac{1}{2} q^2 dt^3 & 0 & 0 \\
0 & \frac{1}{2} q^2 dt^3 & 0 & 0 & \frac{1}{2} q^2 dt^2 & 0 \\
0 & 0 & \frac{1}{2} q^2 dt^2 & 0 & 0 & \frac{1}{2} q^2 dt^2 \\
\frac{1}{2} q^2 dt^2 & 0 & 0 & q^2 dt & 0 & 0 \\
0 & 0 & \frac{1}{2} q^2 dt^2 & 0 & 0 & q^2 dt \\
0 & 0 & 0 & \frac{1}{2} q^2 dt^2 & 0 & q^2 dt \\
\end{bmatrix}
\]  (3)

In the R matrix, \( \sigma_1 \) and \( \sigma_2 \) are the standard deviations of the position and velocity measurement errors, respectively. In the Q matrix, \( q \) can be tuned to adjust for the size of the system modeling error.

Next, it was assumed that the position and velocity vector components were measured directly, so that no coordinate transformation was necessary. This allowed the observation matrix \( H \) to be a \( 6 \times 6 \) identity matrix. Finally, the partial derivative \( F \) matrix was calculated to provide a linear approximation of the relationship between the rates of change of each state and the states themselves:

\[
F = \frac{\partial (x, y, z, \dot{x}, \dot{y}, \dot{z})}{\partial (x, y, z, \dot{x}, \dot{y}, \dot{z})}
\]  (4)

With the estimator initializations complete, the Kalman filtering algorithm was followed as shown in Table 1. After the covariance updates, the filter repeats the algorithm loop starting with state propagation.

B. Least-Squares Filter

Just like with a Kalman filter, the \( H \) and \( F \) matrices must be calculated, and an initial guess for the states must be made. However, unlike a Kalman filter, there is no “tuning” that takes place prior to running the simulation. The weighting gain matrix is set to \( W = I \), and then the algorithm in Table 2 is applied. The LSQ algorithm actually contains two loops, one nested within the other. The inner loop consists of all the steps from state propagation to temporary buffer accumulation. Within the inner loop, the states, state transition matrix, and temporary buffers must be propagated through the entire batch of data so that a
correction factor can be calculated at the end, which will change the initial conditions to improve the fit. The entire batch of data must be processed every time an optimization is made. The outer loop includes the correction factor calculation and initial condition update. The LSQ filter runs until the correction factors become sufficiently small, indicating that the filter has converged on an optimized estimation.

IV. Kalman and LSQ Filter Results

As discussed above, the Kalman and LSQ filters were tested against truth data numerically generated from equations of motion that included two-body gravitational effects as well as $f_2$, $f_3$, and $f_4$ zonal harmonics. The measurements were then created by adding white Gaussian noise with a standard deviation of 300 meters. The following plots demonstrate the Kalman filter estimation accuracy as the filter is tuned with different $Q$ values. The vertical axis of each plot shows the difference between the filter’s estimated $x$ component of position and the $x$ component of the truth orbit at that time, in kilometers. Results were similar for each position and velocity component, so only the $x$ component plots are shown here for simplicity. The horizontal axes plot time over one 24 hour period.

Figure 2 demonstrates Kalman filter performance when $q = 1e^{-6}$. As shown, the estimated position remains within 100 meters of the true position even when the standard deviation of the measurement noise is 300 meters and the filter’s model does not account for $f_3$ and $f_4$ zonal harmonics.

However, when the $q$ value is changed to $1e^{-8}$, reducing the value of the assumed system model error, the estimation accuracy improves to within 50 meters of the true position, as demonstrated in Fig. 3.

![Figure 2. Kalman Filter Estimation Error. When $q = 1e^{-6}$](image)

![Figure 3. Kalman Filter Estimation Error. When $q = 1e^{-8}$](image)
Then, when the $q$ value is lowered even more to $1e^{-10}$, causing the Kalman filter to rely even more heavily on the system model, the estimation converges to within 10 meters of the true position, as shown in Fig. 4.

![Figure 4. Kalman Filter Estimation Error.](image)

While the progression of these three plots demonstrates how tuning a Kalman filter can greatly improve the estimator’s accuracy, it also illustrates a crucial tradeoff inherent in the tuning process. Although the accuracy improves, the settling time increases, meaning that the filter takes much longer to converge to steady-state accuracy. Nevertheless, Kalman filters have the ability to accurately converge on the true position even when the initial guesses for each state are wildly wrong. Figure 5 shows that even when the filter is initialized with position components that are 2000 km from the truth and velocity components that are 3 km/s off, the filter still converges to an estimation accuracy of 100 meters within one day’s time.

![Figure 5. Kalman Filter Estimation Error.](image)
When the LSQ processed the same measurement file, its $\tau$ component estimation error produced Fig. 6 below.

![Figure 6. Least-Squares Filter Estimation Error.](image)

Since the truth model and the filter’s assumed model were very similar, most of the error came from measurement noise. The LSQ filter thrives on accurate modeling, so it achieved greater accuracy than the Kalman filter in this case, estimating position within 6 meters of the truth.

A. Fitting vs. Propagation Accuracy

Next, the Kalman and LSQ filters were investigated in a scenario designed to demonstrate their ability to provide both propagation and fitting accuracy. In this scenario, the filters processed a measurement file with 24 hours of data, and then continued to propagate the states for an additional 2 days without measurements, using only the assumed model. Fitting accuracy is defined as the estimator’s ability to converge on the true position during the period of measurement processing, while propagation accuracy is defined as the filter’s ability to track the true orbit during the period without measurement updates. Figure 7 exhibits the Kalman filter’s performance when the $Q$ matrix was tuned to $q = 1 \cdot 10^{-7}$.

![Figure 7. Kalman Filter Fitting and Propagation Accuracy.](image)

When $q = 1 \cdot 10^{-7}$
Figure 8 below demonstrates the same Kalman filter’s performance when $q = 10^{-10}$. By tuning the Kalman filter, the propagation accuracy was increased by 200 meters in the second simulation, but clearly at the expense of fitting accuracy.

When the LSQ filter was run through the same simulation, it demonstrated superior propagation accuracy than the Kalman filter, but worse fitting accuracy, as exhibited in Fig. 9.

While the Kalman filter can be tuned to provide outstanding fitting accuracy, the LSQ filter matches or exceeds the Kalman filter’s best propagation accuracy.

B. Orbital Maneuver

The last scenario of the project used the estimators to process position and velocity data that included a simple orbital maneuver. To simulate an orbital maneuver, an orbit propagator was developed that included a change in velocity magnitude at a commanded time. The velocity change was programmed to be instantaneous, thus assuming an impulsive burn. The associated measurement file for filter processing was generated by adding noise with 100 meters standard deviation.
The Kalman filter required minimal changes to accommodate the orbital maneuver data. It processed the measurements normally both before and after the burn. The following three graphs show the Kalman filter estimation results as it was tuned. In Figure 10, when $q = 10^{-5}$, the filter relied heavily on measurements and was able to recover quickly from the maneuver. The estimator converged to 60 meters of accuracy 700 seconds after the burn.

![Figure 10. Kalman Filter Estimation Error. This scenario includes a simulated orbital maneuver when $q = 10^{-5}$](image)

When the filter was tuned to $q = 10^{-7}$, the filter took much longer to recover from the burn, but achieved higher accuracy once it settled. The filter reached 30 meters of accuracy 18000 seconds after the maneuver, as shown in Fig. 11.

![Figure 11. Kalman Filter Estimation Error. This scenario includes a simulated orbital maneuver when $q = 10^{-7}$](image)
However, when the Q matrix was tuned to rely too heavily on an inaccurate model, the filter became unstable after the maneuver and failed to converge on the true position. The following plot shows the results when $q = 10^{-9}$.

![Figure 12. Kalman Filter Estimation Error. This scenario includes a simulated orbital maneuver when $q = 10^{-9}$](image)

Unlike the Kalman filter, the LSQ estimator had to be significantly remodeled before it could process this orbital maneuver scenario. The velocity change caused the satellite to enter a different orbit, so the LSQ filter had to process the measurements before and after the burn separately. Otherwise, it would erroneously try to fit the entire batch of measurements to two different orbits. Once the filter was coded to split the measurement file into two segments at the time of the orbital maneuver, as well as include a buffer zone of time around the orbital maneuver so that the thrusters could settle into steady-state, the filter processed the scenario and developed the following plot.

![Figure 13. Least-Squares Filter Estimation Error. This scenario includes a simulated orbital maneuver](image)

Although the LSQ filter achieved an accuracy of 80 meters, which was significantly less accurate than the Kalman filter, the LSQ estimator requires no recovery time to re-converge upon the true position.
V. Conclusions and Future Work

The preceding research demonstrated several advantages and disadvantages for each type of orbital estimator. The Kalman filter’s principle strength lies in its flexibility, since it can be tuned to rely more heavily on measurements or models. If a satellite carries accurate sensors, then the Kalman filter has an advantage. However, if a satellite possesses poor sensors or malfunctioning equipment, then the LSQ filter can provide better estimation with a good orbital model. Additionally, Kalman filters can handle very poor initial guesses, while LSQ filters fail to converge if the initial guess is not accurate enough.

Particularly concerning orbital maneuvers, LSQ filters surpass Kalman filters in the arenas of propagation accuracy and settling time. Nevertheless, Kalman filters ultimately use much less computing power, and so they can be implemented much more feasibly into onboard computing systems.

While this project serves as a good start towards the solution of the original problem concerning how to ensure that thrusters accomplish orbital maneuvers as planned, many steps remain. Continuing work would likely include: 1) Checking COE changes due to maneuvers, 2) investigating Kalman tuning variance during burn phases, 3) implementing more complex ΔV scenarios, 4) processing GPS measurements, and 5) developing an orbital controller. Tracking COE changes would be especially important for characterizing the thruster performance.

Acknowledgments

Todd V. Small would like to thank Dr. Yoshi Hashida for his tremendous help and mentorship during this research project. Dr. Hashida provided much of the basis for this work through his personal knowledge and published documents. Todd Small would also like to thank the rest of the staff at the University of Surrey, including the Surrey Space Center and Surrey Satellite Technology, Ltd., who volunteered their time and resources to make this research possible. In addition, Todd Small extends his gratitude to Lt Col Tim Lawrence and the rest of the U.S. Air Force Academy Department of Astronautics for their support throughout this endeavor.

References

1Hale, M. J., Hashida Y., and Vergez P., “Kalman Filtering and the Attitude Determination and Control Task,” USAF Academy, pp. 3