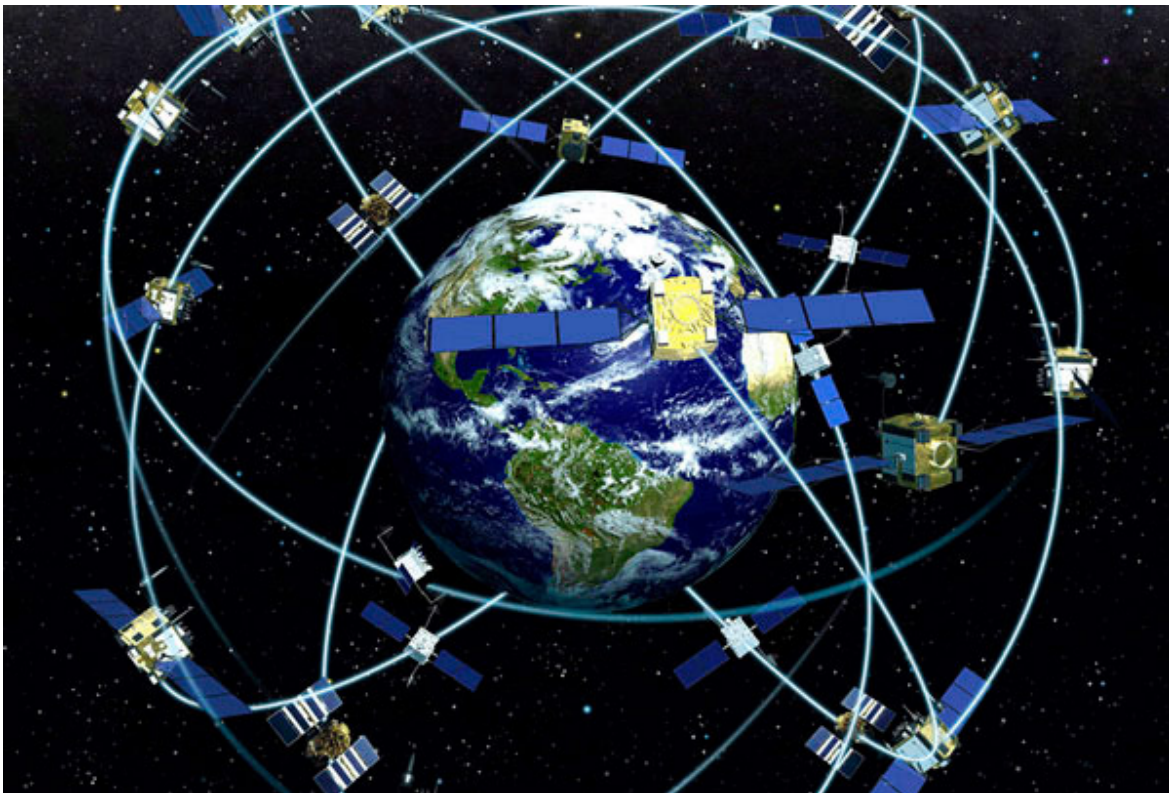


ASEN 5070-Statistical Orbit Determination-Final Project Report

Zach Dischner
University of Colorado at Boulder
Department of Aerospace Engineering

12-2-2012



[2]

1 Introduction

This report summarizes an investigation of various methods of statistical orbit determination, as outlined in ASEN 5070. All programming was performed in Matlab, using a combination of built in functions, self-defined functions, and ones created in collaboration with others. I will examine the results and implications of various filter methodologies including:

- Batch Processor
- Conventional Kalman (Sequential) Filter
- Extended Kalman Filter
- State Noise Compensation
- Alternative Methods for Determining P , the Covariance Matrix

Henceforth, any mention of *textbook*, *5070 textbook* or simply, *the book* will refer to the text written by Byron D. Tapley, Bob E. Schutz, and George H Born, titled "Statistical Orbit Determination" [3]. All equations, methodologies, and definitions were obtained from this text¹.

¹Unless otherwise mentioned, all equations are from this text and will not be cited individually

Contents

1	Introduction	2
2	Background	4
2.1	Orbit Determination Process	4
2.1.1	Overview of Orbit Determination Process	4
2.1.2	Linearization	4
2.1.3	State Transition Matrix	5
2.2	OD Solutions	6
2.2.1	Least Squares Solution	6
2.2.2	Weighted Least Squares	7
2.2.3	Weighted Least Squares with <i>a-priori</i>	7
2.2.4	Minimum Variance Estimate	7
2.3	Orbit Determination Filters	8
2.3.1	The Batch Processor	8
2.3.2	Sequential Filter	9
3	My system	10

List of Figures

2 Background

2.1 Orbit Determination Process

2.1.1 Overview of Orbit Determination Process

The orbit determination process is, at the fundamental level, one which determines a celestial body's motion relative to another. Typically, this process is used to determine the motion of Earth-launched satellites relative to Earth. Though the problem can and is often applied to a variety of systems, but this paper will concern itself with Earth-centered satellites and their dynamical states. Even so, the same process described here can easily be applied to any dynamical system. From tracking trains and automobiles, to basic control systems for pointing mechanical linkages, the theory and procedures are generally the same.

The state of a satellite is "a set of parameters required to predict future motion of the system"[3]. These parameters include the position and velocity vectors of the satellite, and often includes other information relating to the dynamical model. Other information can include atmospheric drag, solar wind, gravity terms, tracking station information, or other system dynamics. Fundamentally, anything can be included in the state that the operator would wish to track and model.

The process of determining a satellite state at a given Epoch involves convolving information about its present and past state, in a mathematically optimized manner. Present state information comes from both physical observations of the system, as well as from a dynamical system model. Observations often comes from range, range-rate, azimuth, elevation, angel, and other physically observable quantities, provided by tracking ground station or other celestial bodies. The dynamical model is a purely mathematical approximation of the satellite's state in time. Information about the satellite's past state (*a-priori*) information) comes from the navigator's historical data.

Important to note is the fact that all information about the state is imperfect. Observations have biases and accuracy in measurements, any model will have errors and unknown factors at play, and *a-priori* information is a result of similar imperfections obtained from past information. Basically, the true state of the satellite is never known. The OD process is one which uses all of the imperfect information to generate a statistical "best" estimate of the satellite state at a given epoch.

Another key note is that the relation between observation of the state and the state itself are highly nonlinear in most applications.

2.1.2 Linearization

One of the most crucial elements of the OD process is the ability to model it in a linear manner, even when both the dynamics and observation relationships are highly nonlinear. Typical OD problems can be formulated in the following manner [3].

$$\dot{X} = F(X, t), \quad X(t_k) \equiv X_k \quad (2.1)$$

$$Y_i = G(X_i, t_i) + \epsilon_i; \quad i = 1, 2, \dots, \ell \quad (2.2)$$

X_k is the n -dimensional state vector we wish to track at time t_k , and Y_i the observation set used to obtain the best estimate of the state at a given time. In this problem, the system is comprised of indirect observations of a satellite state, with inherent observation errors and biases, and a nonlinear mapping between the state and observation vector. The problem of using said nonlinear observation maps to determine the state with nonlinear dynamics at a given time is very difficult one.

The key in this OD process comes at this point. If the state X , and the reference state X^* (obtained from numerical integration) can be assumed to be reasonably close during the time under examination, the deviation between the two states can be assumed linear. This involves setting a Taylor Series expansion of the true state about the reference state, and neglecting higher order terms. Upon doing so, the full nonlinear OD problem is simplified into solving for the *state deviation vector*, x , which involves solving a simpler set of time-dependent ODEs. In this, the problem of estimating an orbit's trajectory and properties at a point in time becomes one in which we find the deviation of the observed trajectory from a reference one. For a fuller explanation of this procedure, reference the textbook [3, section 4.2].

As a result of the linearization, the OD problem described in (2.1) and (2.2) becomes a simpler one .

$$\dot{x}(t) = A(t)x(t) \quad (2.3)$$

$$y_i = \tilde{H}_i x_i + \epsilon_i; \quad i = 1, 2, \dots \ell \quad (2.4)$$

With:

$$A(t) = \left[\frac{\partial F(t)}{\partial X(t)} \right] \quad \tilde{H}_i = \left[\frac{\partial G}{\partial X} \right]$$

The A and \tilde{H} matrices are critical in the solving of the OD problem. They are a result of the linearization process. This paper will discuss how to find these specifically **later**.

In general, it can be seen that A is found by taking the partial derivative $\frac{\partial F}{\partial X}$. Which is the partial of the system's dynamical model with respect to the system state. In essence, this describes how the system's dynamics change as the state changes itself. The fact that this is a linear relation for most well formulated problems is what allows the OD problem to be solved in the way we do. In the same way, \tilde{H} is found by taking the partial derivative $\frac{\partial G}{\partial X}$. G is the relationship between the satellite state and its observations. This partial represents how the state-observation relation changes with changes in the state. Again, this partial assumes the deviations are linear over the time in question. This holds true for well formulated problems.

2.1.3 State Transition Matrix

Equation (2.3) is a linear system of equations with time-dependent terms. A general solution to this system is presented as [3]:

$$x(t) = \Phi(t, t_k)x_k \quad (2.5)$$

This equation presents Φ , the *state transition matrix*. This matrix can be used to map the state or state deviation vector forwards and backwards in time. In this general case, it maps x at t_k to x at some arbitrary time t . This matrix has several useful and unique properties. All of which are outlined thoroughly in the *5070 textbook*. In summary, these properties lead to the formulation of:

$$\dot{x}(t) = \dot{\Phi}(t, t_k)x_k \quad (2.6)$$

Equation (2.6) is of the same form as 2.3. That is, it describes a linear set of differentiable equations. In practice, this leads the operator to include a re-formed Φ with the state vector during numerical integration, giving a numerical solution for the state transition matrix at each state reference time. In practice, a numerical solution for Φ is all that will be obtained, as an analytical one only arises from conditions such as linearity rarely seen in real operations.

While seeming unnecessary immediately, the state transition matrix is powerful in that it can project the state or state deviation forwards or backwards in time. Often times, an OD filter must be iterated; a process in which observations and state information must be related to some epoch time. The state transition matrix is one of the simplest ways to perform this mapping.

For example, to map all observations back to a reference epoch at t_k , equation (2.4) may be formulated as:

$$\begin{aligned} y_1 &= \tilde{H}_1 \Phi(t_1, t_k) x_k + \epsilon_1 \\ y_2 &= \tilde{H}_2 \Phi(t_2, t_k) x_k + \epsilon_2 \\ &\vdots \\ y_\ell &= \tilde{H}_\ell \Phi(t_\ell, t_k) x_k + \epsilon_\ell \end{aligned} \tag{2.7}$$

In this, ℓ separate observation deviation sets are all mapped to the epoch state deviation vectors at x_k . This is desirable for iterative filtering, as well as reducing the number of equations to be solved in the system.

2.2 OD Solutions

Now, with the formulation of a workable OD problem in section 2.1, the statistical 'best' solution must be determined. Conceptually, there are an infinite number of measurable metrics, or *performance indices*, which can be design against. This is one of the most important steps in solving the OD problem. What is the best way to combine an erroneous reference trajectory and noisy/biased observations to obtain an estimate of our state that is closest to the truth trajectory? This simple question is what drives the design of OD filters and determination algorithms. Several of the basic realizations of the answer to this question will be provided here, though there are countless more.

2.2.1 Least Squares Solution

One of the most common approaches is to attempt to minimize the sum of the squares of the observation residuals. The performance index for this approach is [3]:

$$J(x) = \frac{1}{2} \epsilon^T \epsilon \tag{2.8}$$

Choosing x to minimize eq(2.8) is a natural choice. By design, this procedure is robust to the sign of the residual. If large residuals exist, but have opposite sign, the minimization attempt could yield calculated observation errors of zero. **This method is also sensitive to outliers.**

A solution for x must be found that minimizes the least squares index. From eq(2.4), ϵ may be solved for, and substituted into eq(2.8).

$$J(x) = \frac{1}{2} (y - Hx)^T (y - Hx) \tag{2.9}$$

Minimizing J is done by finding a zero in its partial derivative, where the second derivative is positive.

$$\frac{\partial J}{\partial x} = 0 \text{ Where } \delta x^T \frac{\partial^2 J}{\partial x^2} \delta x > 0$$

After some algebra, a final formulation for x that minimizes the least squares criterion is

$$\hat{x} = (H^T H)^{-1} H^T y \tag{2.10}$$

By solving for the state deviation vector in this manner, the sum of the square of the observation residuals is minimized. A full derivation of this solution can be found in the *5070 textbook* [3]. While this is a sufficient measure for some applications, more powerful realizations exist which can help obtain a more accurate prediction of the satellite.

2.2.2 Weighted Least Squares

The standard least squares formulation can be modified by adding weighting to each observation vector. In reality, not all observations have equal precision or accuracy, and the circumstances surrounding the observation may cause certainty to change. For example, a laser range finder will obtain more precise results as the satellite flies directly overhead, as opposed to near the horizon, where atmospheric turbulence will distort measurements. With the inclusion of a weighting matrix, eq(2.8) becomes

$$J(x) = \frac{1}{2} \epsilon^T W \epsilon \quad (2.11)$$

W is a diagonal weighting matrix, with size $\ell x \ell$. Each diagonal element corresponds to a weighting value to be applied to observation vector. After more algebra, the solution found in eq(2.10) with weighting included becomes

$$\hat{x} = (H^T W H)^{-1} H^T W y \quad (2.12)$$

The implications of this are the same. By determining \hat{x} in this manner, we wish to minimize a performance criteria which now includes weighting factors.

2.2.3 Weighted Least Squares with *a-priori*

In the same vein as adding a weighting to the least squares solution, an even better state deviation estimate can be found by including information about the state's history, known as *a-priori* information. This information comes in the form of \bar{x} , the last known state deviation vector, and \bar{W} , the associated previous weighting matrix associated with that state. When these terms are added into the weighted least squares solution, eq(2.12) becomes

$$\hat{x} = (H^T W H + \bar{W}_k)^{-1} (H^T W y + \bar{W}_k \bar{x}_k) \quad (2.13)$$

This solution now takes advantage of the most commonly available information about a given state/state deviation vector. With it, comes the most accurate estimate for \hat{x} that has been discussed yet.

2.2.4 Minimum Variance Estimate

All above formulations of the least squares solutions makes intuitive sense when considering the physical system, however, they lack in the fact that they do not account for statistical characteristics in observation errors and *a-priori* information. One other common estimator is the *Minimum Variance* estimator. This method tries to utilize any statistical information about observations in time to generate the optimal estimate of \hat{x} . The derivation of this method won't be explored, but it is in the course textbook.

$$\hat{x} = P_k H^T R^{-1} y \quad (2.14)$$

Where

$$P = E[(\hat{x}_k - x_k)(\hat{x}_k - x_k)^T] \quad R = E[\epsilon \epsilon^T] = W^{-1}$$

P is the system's covariance matrix. It holds the variances (certainty) of each state element on its diagonal, and the off-diagonal elements contain the linear correlation factors between state elements. It is a symmetric, positive definite matrix by definition (when properly conditioned), and is a crucial statistical measure of a filter's certainty in its solution at a given time. **R is a matrix which holds information about the observation errors, and their relation to each-other.**

Again, adding *a-priori* information to eq(2.14) can help to yield a more robust estimation.

$$\hat{x} = (\tilde{H}_k R^{-1} \tilde{H}_k + \bar{P}_k^{-1})^{-1} (\tilde{H}_k R^{-1} y_k + \bar{P}_k^{-1} \bar{x}_k) \quad (2.15)$$

Theoretically, there are an infinite number of *best* estimate solutions. From a 'least cubed' or 'least 4/3' estimates, to an often used *Maximum Likelihood and Bayesian Estimation* solution, there are dozens of implemented and derived OD solutions. The general process is to first pick a performance index, $J(x)$. This should be a function which has some basis in statistical reasoning. Next, solve for a value of x which minimizes the performance index.

2.3 Orbit Determination Filters

With assumptions about the linearity of the deviation between satellite truth and reference states, and statistically derived criteria for solving for this state deviation, the process of orbit determination can begin. There are several standard approaches to solving the OD problem. This paper will only concern itself with the basic ones.

2.3.1 The Batch Processor

The Batch Processor is one formulation of the OD process. As the name implies, its aim is to process a set of observations and a reference trajectory at one time, in a single batch computation. Typically, the entire batch of computation is done in order to get a new best estimate for the state deviation vector at an epoch t_0 . In operation, the satellite's current trajectory is not the only state of interest. For science missions, a very accurate estimation of the state at a given historical instance is needed for high quality analysis of data. The batch processor takes information about a time before the epoch of interest, and information afterwards to generate the best estimate of the satellite state at epoch. Using the weighted least squares with *a-priori* information, the eq(2.13) equation for the state deviation vector for a full batch of observations becomes

$$\hat{x}_0 = (H^T R^{-1} H + \bar{P}_0^{-1})^{-1} (H^T R^{-1} y + \bar{P}_0^{-1} \bar{x}_0) \quad (2.16)$$

Here, t_0 is the epoch time of interest. Since eq(2.16) is supposed to solve for \hat{x}_0 , all quantities must be mapped to that epoch using the state transition matrices, described in section 2.1.3.

The course text book features an in-depth procedure for using the batch processor in an application, as well as insights into key terms and quantities in Chapter 4.7. It will not be discussed here for the sake of brevity. The general process is to [1]:

- Collect Observations and Residuals Over an Arc
- Process All Using Weighted Least Squares with *a-priori*
- Generate Best Estimate
- Iterate

The batch processor will yield an updated result for the state deviation vector at epoch, however it must be iterated in order to obtain the absolute best solution. The iteration process is straightforward, and is done by re-running the filter after updating the initial conditions with the new best estimates. Again, this process is outlined in the textbook.

One problem with the batch processor is the fact that a large matrix inversion is required. This is evident in eq(2.16), and this portion is called the *information matrix*, eq(2.17). In practice, Cholesky decompositions or other methods would be utilized to increase the accuracy of this inversion, but matrix inversions are still computationally heavy.

$$\Lambda_0 = (H^T R^{-1} H + \bar{P}_0^{-1}) \quad (2.17)$$

The other potential detriment to the batch processor is the fact that since it uses future information to get the best estimate of some past epoch, it is not ideally suited for live tracking situations. This is where the sequential processor comes in.

2.3.2 Sequential Filter

The conventional sequential filter, commonly referred to as a Kalman filter, KF, or CKF, is a reformulation of the batch processor described in 2.3.1. With perfect numerical precision, the two yield identical results.

The advantage of the CKF is that it processes observations live. As soon as a new observation is available, it will be processed. The other advantage is that not only can it process observation sets one at a time, but it can process just single observations sequentially. Even when a set is observed at the same time. This is advantageous because it cuts down on computational requirements by just involving scalar divisions [3]. The reformulation of the batch solution is as follows

$$\hat{x}_k = \bar{x}_k + K_k[y_k - \tilde{H}_k \bar{x}_k] \quad (2.18)$$

Where the covariance matrix, P at new time t_k is

$$P_k = [I - K_k \tilde{H}_k] \bar{P}_k \quad (2.19)$$

And K is found at each observation time as

$$K_k \equiv \bar{P}_k \tilde{H}_k^T [\tilde{H}_k \bar{P}_k \tilde{H}_k^T + R_k]^{-1} \quad (2.20)$$

K is referred to as the Kalman Gain. It is a matrix which weighs the certainty of the observations against the certainty of the reference trajectory. Equation (2.18) shows the computation of the best estimate state deviation vector at time t_k . It is a balance between the *a-priori* state deviation vector and the newly observed difference in observed and computed observation deviation vectors. The *balancing* is performed by the Kalman gain. The computation uses weighting, as well as *a-priori* information.

As with the batch processor, the CKF should be iterated to obtain the best solution. Instead of computing the state deviation at the epoch time, here it is computed at the latest update time. To iterate, this solution must be mapped back to the epoch time using the state transition matrix (section 2.1.3).

The full procedural outline is found in Chapter 4.7 of the textbook, but is summarized here.

- Collect Observations and Residuals Over an Arc
- Process All Using Weighted Least Squares with *a-priori*
- Generate Best Estimate
- Iterate

3 My system

References

- [1] G. B. Jeffry Parker. Asen 5070 f2012 lecture 11. <http://ccar.colorado.edu/ASEN5070/lectures.htm>.
- [2] A. Name. A thing. somesite.orgnet.
- [3] B. Tapley, B. Schutz, and G. Born. *Statistical Orbit Determination*. Elsevier Acad. Press, 2004.