

Robust Orbit Determination via Convex Optimization

Bob Wilson
rwilson4@stanford.edu

June 4, 2014

1 Background

Given the initial position and velocity of an object, such as a satellite in orbit around the Earth, it is possible to determine its position and velocity at any other time by integrating the equations of motion. This prediction method is called propagating. Specifying the position and velocity of the satellite at a specific time, called the epoch, is a common way of representing an orbit. In this context, the position and velocity are known as the Cartesian orbital elements. Different methods of representing an orbit are in common use and it is rare to use one representation in isolation. Classical elements are another common representation that are easily interpreted and propagated. For this reason, classical elements were used in this project. While providing a thorough background in orbital mechanics is beyond the scope of this paper, an excellent reference is [PC12], which discusses orbital elements and propagation in detail.

Given the orbit of a satellite, it is straightforward to compute its distance from a point on the Earth, such as a ground station, at any time. Ground stations can measure this distance, called the range, by sending a signal to the satellite. The satellite sends back a signal after a known delay, which is finally received by the ground station. By measuring the total transmission time and factoring in atmospheric effects, the range can be estimated with high accuracy. A ground station may also measure the elevation and azimuth of a satellite. The issues involved in accurate estimation of these parameters, collectively called tracking data, are discussed in [TSB04, §3]. For simplicity, in this project we focused on range measurements only. Accommodating other measurement types is straightforward.

Let $f_r(x; t)$ be the range measured by a particular ground station at a particular time t , provided the orbital elements are given by x and the measurements have no noise. Informally, f_r answers the question: at time t , how far is the satellite from the ground station? Evaluating f_r is nontrivial. We must propagate the orbital elements to the desired time and account for signal delays to determine the range a ground station would measure at time t . High fidelity propagators will result in an f_r for which analytical formulas are not possible, and in practice, even simple forms are too complex to differentiate explicitly. Instead, numerical methods are used to compute gradients approximately.

Suppose a ground station collects N range measurements at times t_1, \dots, t_N . Without noise, the range measurements would be $\{f_r(x, t_i)\}_{i=1}^N$. Real measurements will be corrupted by noise; for this reason, $\{f_r(x, t_i)\}_{i=1}^N$ are called the ideal measurements. Given real measurements y_1, \dots, y_N at times t_1, \dots, t_N , we can estimate x by solving the problem

$$\text{minimize } \sum_{i=1}^N \phi_i(y_i - f_r(x, t_i)), \quad (1)$$

with data $\{t_i, y_i\}$, variable x , and unspecified penalty functions ϕ_i . Solving (1) is called orbit determination. The solution, x^* , is a representation of the estimated orbit. References for orbit determination include [TSB04] and [MG11].

Since f_r is not convex, the problem cannot easily be solved globally; however, iterative methods may be used to find a local minimum based on an initial guess. In Newton's method, we repeatedly linearize f_r about the current estimate x^k and solve (1) to get the next estimate x^{k+1} . The industry standard is to use $\phi_i(r) = w_i r^2$, where w_i is a weight that may depend on considerations such as the type of tracking data (range, azimuth, or elevation). In this approach, we solve a sequence of weighted least squares problems to estimate the orbit.

While least squares methods have several practical benefits, robustness is not one of them. While the method of least squares is optimal, in a sense, when the noise is Gaussian, real noise sources often have fatter tails than Gaussian distributions. We have at least three noise sources: the tracking data, the model parameters implicit in f_r , and the linearization of f_r . If these sources are non-Gaussian, robust methods may outperform least squares. We therefore set out to solve

$$\text{minimize } \sum_{i=1}^N \phi(y_i - \hat{f}_r^k(x; t_i)) + \lambda \psi(x - x^k), \quad (2)$$

where $\hat{f}_r^k(x; t_i)$ is an affine approximation of f_r centered at x^k . In this problem, x is the optimization variable; y_i , t_i and x^k are data. We have two penalty functions: ϕ is an unspecified robust penalty on the residuals, and ψ is an unspecified penalty on the estimate update. We use λ to trade off between competing objectives. We want to find an orbit estimate that fits the data, but we also do not want to deviate too far from the region where linearization is a good approximation for the original nonlinear function f_r .

When ϕ and ψ are convex, this is a convex optimization problem. The solution of (2) forms the linearization point, x^{k+1} , for the next iteration. Note that if this process converges, the linearization error and update penalty go to zero, which shows the limit of this process results in a local minimum of (1). In our approach, we solve a sequence of convex optimization problems to estimate the orbit.

2 Modeling

For simplicity, we used a propagator that assumes the Earth has a perfectly spherical mass distribution. All effects from other celestial bodies, such as solar radiation pressure and lunar gravity, were ignored. Since nothing in our approach depends on the form of f_r , a higher

Semimajor axis (km)	20000	20000	20000	40000
Eccentricity	0.4	0.4	0.01	0.8
Inclination	40°	40°	40°	40°
Right ascension of the ascending node	0°	90°	0°	0°
Argument of perigee	0°	0°	0°	0°
Mean anomaly	0°	0°	0°	0°

Table 1: Orbits considered, specified in classical elements, with an epoch of 2014/01/01 00:00 UTC

Los Angeles	34°03' N	118°15' W
Washington D.C.	38°54' N	77°02' W
Athens	37°58' N	23°43' E

Table 2: Ground station locations

fidelity propagator could easily be substituted. Atmospheric effects and light propagation delay were ignored, which greatly simplifies the calculation of f_r .

We examined orbits consistent with commercial satellite injection orbits, plus one highly eccentric orbit we suspected would be challenging to estimate. The classical orbital elements (see [PC12]) for these orbits are shown in Table 1. We assumed three ground stations, in Los Angeles, Washington D.C., and Athens. Their coordinates are shown in Table 2. These locations do not correspond to real ground stations, but have latitudes corresponding to the inclinations of the orbits considered in this project, and are distributed across the globe, both realistic features of ground stations used in practice.

While we have access to real tracking data from transfer orbits, it is Boeing proprietary and unfortunately is not available for this project. For this reason, tracking data was generated using the nonlinear function f_r . Specifically, we propagated the orbits in Table 1 in one minute increments and computed the distance to the ground stations to generate ideal range measurements. Range measurements were recorded only when the satellite was at least 10° above the horizon from the perspective of the ground station. Since the period of the orbit—the time needed to make one complete revolution—is a natural unit of time in orbital mechanics, we considered measurements spanning up to three revolutions, in increments of half a revolution. Realistic data is mostly low noise, with some outliers. To model this, we used a mixture of Gaussians approach [Ng13]. Gaussian noise with standard deviation 10 meters, 20 meters, and 1 kilometer were added to 50%, 40%, and 10% of the measurements, respectively.

Linearization of f_r was accomplished using a central difference method. We used a Huber penalty on the data and an ℓ_2 penalty on the update. The Huber penalty has well-known robustness properties [BV04, §6], and a Euclidean norm is perfect for penalizing deviations from the linearization point. Thus in the robust problem (2), the penalty functions were

chosen as

$$\phi(u) = \begin{cases} u^2 & |u| \leq M \\ M(2|u| - M) & |u| > M, \end{cases}$$

$$\psi(x) = \|(J^T J)^{1/2} x\|_2^2.$$

In the update penalty, ψ , we included a weight matrix based on the Jacobian. This penalizes solutions that deviate from the linearization point in directions where the function is changing rapidly. This approach was inspired by the Levenberg-Marquardt algorithm [Lev44], [Mar63], and may be thought of as a trust region penalty [Boy14].

3 Adaptive parameter selection

Adaptive methods for selecting the Huber penalty parameter, M , and the trust region weight, λ , were considered. As we near convergence, the residuals mimic the measurement noise. The Huber parameter should be chosen as the threshold beyond which the noise ceases to appear Gaussian. A Q-Q plot is useful for visually assessing whether a distribution is Gaussian [WG68]. We took a different approach, based on a mixture of Gaussians model [Ng13]. We implemented an Expectation-Maximization procedure in which the residuals were clustered into two groups based on an estimate of two Gaussian distributions. The parameters for those distributions were then estimated based on the residual points in each group. This clustering-then-estimating procedure was applied iteratively until convergence. The Huber parameter was chosen to be the standard deviation of the lower-noise distribution.

To select λ , we compared the expected improvement in residuals based on the affine approximation to the actual improvement observed using the nonlinear model. Let

$$\hat{\delta} = \sum_{i=1}^m \phi(y_i - f_r(x^k; t_i)) - \sum_{i=1}^m \phi(y_i - \hat{f}_r^k(x^{k+1}; t_i)),$$

$$\delta = \sum_{i=1}^m \phi(y_i - f_r(x^k; t_i)) - \sum_{i=1}^m \phi(y_i - f_r(x^{k+1}; t_i)).$$

Here, $\hat{\delta}$ is the predicted improvement in the residual penalty, and δ is the actual improvement seen. If $\delta \geq \alpha \hat{\delta}$, for a value of α reasonably close to 1, then the prediction was accurate, implying the affine approximation was valid. In this case, we may try to increase the effective size of the trust region by decreasing λ to $\lambda/\beta^{\text{succ}}$ for some $\beta^{\text{succ}} > 1$. On the other hand, if $\delta < \alpha \hat{\delta}$, that implies the affine approximation was invalid. In this case, we decrease the size of the trust region by increasing λ to $\lambda/\beta^{\text{fail}}$ for some $\beta^{\text{fail}} < 1$. We used $\alpha = 0.1$, $\beta^{\text{succ}} = 1.1$, and $\beta^{\text{fail}} = 0.5$.

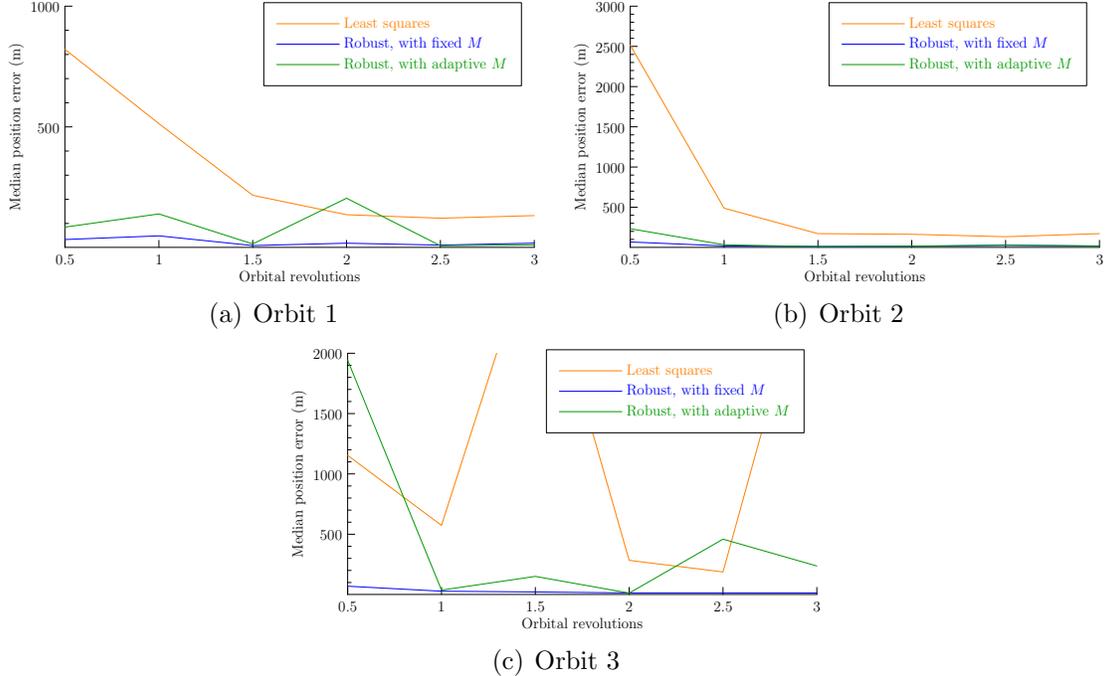


Figure 1: Accuracy of least squares and robust methods.

4 Results

We carried out the above program for the orbits shown in Table 1, with ten IID sets of measurements for each orbit, method, and timespan considered. A robust method using a fixed Huber penalty parameter, $M = 0.015$, was used, as well as a method in which this parameter was adaptively selected as described in §3. Adaptive trust region weighting was used in both methods. Least squares was also used for comparison. In summary, we examined three methods of orbit determination on four different orbits using six different data sets (differing in the number of orbital revolutions) for ten different measurement realizations.

Figure 1 shows the median of the position errors, defined as the distance between the estimated and actual satellite positions at the time of the orbit epoch. The median is over the measurement realizations. Median was used instead of mean since there was a single outlying result (third orbit, fixed M) that skewed the average. The figures suggest that least squares methods have a knee in the curve: it seems data spanning at least one orbital revolution is needed for an accurate estimate. Indeed, it is popularly believed in industry that accurate orbit determinations are impossible when using less than one full revolution of data. On the contrary, robust methods do well even with data spanning only half an orbital revolution, and do not significantly benefit from longer timespans. It would be interesting to investigate even shorter timespans.

In the cases examined, the performance of the adaptive strategy for selecting the Huber penalty parameter, M , was sporadic. It often worked well, but when it failed, results were significantly worse than when using a fixed $M = 0.015$. This is clearly seen in the figures.

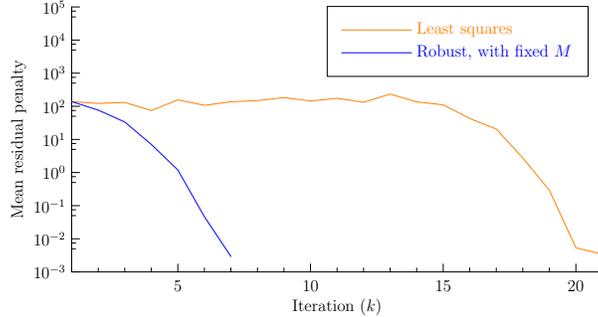


Figure 2: Convergence rates of least squares and a robust method.

We are optimistic our approach may be modified to provide more consistent results.

Figure 1(c) shows a few instances where least squares converged to a local optimum far from the real orbit. Robust methods still performed well in this case—except in a single instance not reflected in the median—perhaps due to the trust region penalty.

The results for the fourth orbit are not shown. This orbit was selected as an extreme case that would be difficult to estimate. In every case examined, least squares failed to converge, while the robust methods converged to points thousands of kilometers from the true orbit. This shows that our method is not immune to the effects of non-global local minima.

Figure 2 shows the convergence rates of least squares and the robust method using a fixed value $M = 0.015$, for a representative case. In this case, the methods converged to estimates with similar residual penalties, but the robust method converged more quickly. We suspect this is due to the trust region penalty. Without this, the solutions of the initial iterations of the least squares method are free to wander far from the linearization point. The resulting estimates have small residuals, but only relative to the linearized objective, which is a poor approximation of the true objective far from the linearization point. Consequently, the solutions found by least squares in the early iterations are untrustworthy. The algorithm jumps from untrustworthy estimate to untrustworthy estimate until, by luck, an estimate is found close enough to the true solution that the linearization is valid. At that point, the algorithm begins to make progress. In contrast, the trust region penalty ensures the linear approximation is valid at the solution of each robust iteration, and convergence is fast.

5 Future work

Our next step will be to incorporate a high-fidelity propagator. We will then be able to apply our method to real data. In practice, an initial orbit determination is used to identify outlying measurements, which are then removed before performing a final orbit determination. Considering our results, robust methods eliminate the need for this step; however, to give least squares fair consideration, we will implement automatic data culling. If real data is Gaussian-plus-outliers, least squares methods with data culling will likely outperform our robust methods, since least squares is optimal for Gaussian measurements. Thus, the real test of our approach is yet to come.

References

- [Boy14] S. Boyd. Sequential convex programming. http://www.stanford.edu/class/ee364b/lectures/seq_slides.pdf, 2014. Accessed May 15, 2014.
- [BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [Lev44] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2:164–168, 1944.
- [Mar63] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441, 1963.
- [MG11] O. Montenbruck and E. Gill. *Satellite Orbits: Models, Methods and Applications*. Springer, 2011.
- [Ng13] A. Ng. Mixtures of Gaussians and the EM algorithm. <http://cs229.stanford.edu/notes/cs229-notes7b.pdf>, 2013. Accessed May 15, 2014.
- [PC12] J. Prussing and B. Conway. *Orbital Mechanics*. Oxford University Press, second edition, 2012.
- [TSB04] B. Tapley, B. Schutz, and G. Born. *Statistical Orbit Determination*. Elsevier Academic Press, 2004.
- [WG68] M. Wilk and R. Gnanadesikan. Probability plotting methods for the analysis of data. *Biometrika*, 55(1):1–17, 1968.