Ballistic Coefficient Estimation Based on ODTK

Xiaobin Huang1*, Yan Zhang1, Li LU1, Rui Xiao1, Yan OuYang1 and Binbin Shi1

1 Air Force Early Warning Academy, Wuhan 430019, China
*Corresponding author’s e-mail: 532454132@qq.com

Abstract. Accurate ballistic coefficients play an important role in orbit determination and target recognition of space targets. In order to quickly and effectively calculate the target's ballistic coefficient, AGI’s ODTK (Orbit Determination Tool Kit) software is introduced, detailed operation steps and parameter settings are designed, and simulation is build to verify the effectiveness of executing the task of ballistic coefficient estimation by the ODTK software. The results of this paper have reference value for researchers engaged in space target surveillance.

1. Introduction

Atmospheric drag is the main factor that affects the precise orbit determination of LEO target, and its accurate calculation is determined by the target’s ballistic coefficient and atmospheric density [1, 2]. Obtaining accurate ballistic coefficients plays an important role in the orbit determination and target recognition of space targets.

The ballistic coefficient of LEO target is defined as $B=CD\cdot A/m$, where $m$ is the mass of target, $CD$ is the atmospheric drag coefficient, and $A$ is the windward cross-sectional area of target. For cooperative space targets, the materials, mass, and on-orbit attitude are known, and the ballistic coefficients can be calculated by models and historical observation data. However, for non-cooperative targets, only the observation data of the target can be used to estimate the approximate value of the target’s ballistic coefficient under certain criteria. In recent years, some scholars have proposed TLE-based method to estimate ballistic coefficient [3, 4]. This method first calculates the average semi-major axis change caused by atmospheric drag from TLE, and then estimates ballistic coefficient using drag perturbation equation. However, this method is post-processing and is not suitable for real-time computing. It is also a common method to combine the ballistic coefficient with the satellite’s orbit state (position and velocity) into a state vector and estimate it by the least-squares orbit determination processing [5]. The advantage of this method is that it has a wide range of applications and has real-time characteristics. Therefore, ballistic coefficient estimation theory based on orbit improvement method is explained in this paper, an advanced orbit determination and analysis software - Orbit Determination Tool Kit (ODTK) developed by Analytical Graphics Inc. (AGI) is introduced, operation method of ballistic coefficient estimation using this software is designed, and the effectiveness of the operation method is verified with simulation data.

The content of this paper is organized as follows: The first part describes the principle of least-squares orbit improvement; the second part explains the calculation method of partial derivatives in the model; the third part introduces the ODTK software; the fourth part gives the ODTK operation steps for ballistic coefficient estimation, and verifies the effectiveness of the operation method with simulation data. The results of this paper have reference value for researchers engaged in space target surveillance.
2. The principle of least-squares orbit improvement

The basic principle of least-squares orbit improvement is to find an orbit that minimizes the residual between computed observations and actual observations, that is, solving $x_0$ to minimize the value of equation (1).

$$J(x_0) = [z - h(x_0)]^T[z - h(x_0)]$$

where $x_0 = [r(t_0); v(t_0); p(t_0)]$ is a $6+n_p$ dimensional state vector of the satellite at time $t_0$, including satellite position vector $r$, velocity vector $v$, and $n_p$ solve-for parameters $p$. A reference trajectory can be obtained by orbit prediction using this epoch state vector: $z = [z_1; z_2; \cdots; z_n]$ is $N$ times actual observation vector; $h(x_0)$ is the theoretical observation value computed from the reference trajectory.

Since $h$ is a nonlinear function, Eq. (1) describes a nonlinear least-squares problem, which is extremely difficult to solve. Usually Taylor expansion is used to transform it into a linear least-squares problem, and then its approximate linear solution is solved iteratively as follows [1]

$$x_0^{(i+1)} = x_0^{(i)} + [H^{(i)^T}W^{(i)}]^{-1} H^{(i)^T}W[z - h(x_0^{(i)})]$$

(2)

The initial iteration value is $x_0^{(0)} = x_0^{\text{ref}}$, which can usually be determined by the initial orbit determination [2]. The so-called orbit improvement is to iteratively correct the initial value. $W$ is a diagonal square matrix composed of observation errors, which plays a weighting role. The iterative process is terminated until the relative change of the residual for two consecutive times is less than a given threshold. The Jacobian matrix $H$ is

$$H^{(i)} = \frac{\partial h(x_0)}{\partial x_0} \bigg|_{x_0=x_0^{(i)}}$$

(3)

It should be noted that the solve-for parameters (such as force model parameters related to satellite motion, observation model parameters, etc.) are usually combined with the satellite’s orbital state $(r, v)$ to form a state vector for estimation.

3. Calculating partial derivatives of model

The key to the calculation of Eq. (2) lies in the solution of the partial derivative Jacobian matrix $H$, which is very difficult to directly calculate. To facilitate the derivation, we make the following conventions without loss of generality:

- Take the radar observation at a certain moment (including distance, azimuth and elevation);
- For the convenience of writing, the time index is omitted at other times except $t_0$;
- The ballistic coefficient related to atmospheric drag is selected as solve-for parameter.

Usually the derivative chain rule is used to convert $H$ to

$$H = \frac{\partial h}{\partial x} \frac{\partial x}{\partial x_0} = A\Phi$$

(4)

In above equation, matrix $A$ is the partial derivative matrix of the current observation with respect to the current state vector, and $\Phi$ is the partial derivative matrix of the current state vector with respect to the initial state vector, also called the error state transition matrix.

The acceleration caused by atmospheric drag is [5]

$$a_{\text{drag}} = \frac{1}{2} C_D \frac{A}{m} \rho v_i v = -\frac{1}{2} B \rho v_i v_i$$

(5)

where $B$ is the ballistic coefficient having $B = 0$; $\rho$ represents the atmospheric density at the altitude of the satellite orbit, which is determined by the earth atmosphere model; $v_i$ is the satellite’s relative velocity with respect to the local atmosphere, and $v_i$ is the modulus of the velocity.

The relative velocity $v_i$ can be written approximately as

$$v_i = v - \omega_0 \times r$$

(6)

with the inertial satellite velocity vector $v$, the position vector $r$, and the Earth’s angular velocity vector $\omega_0$ of size $0.7292 \times 10^4$ rad/s.
3.1. Calculating partial derivative matrix

After ignoring some factors such as optical aberration, radar observations are only related to the current position of the satellite, and have nothing to do with its velocity and ballistic coefficient. Therefore, the $A$ matrix can be split into

$$A = \frac{\partial h}{\partial x} = \begin{bmatrix} \frac{\partial h}{\partial r_{\text{ECI}}} & 0_{3\times4} \end{bmatrix}$$  \hfill (7)

In the equation, $r_{\text{ECI}}$ is the position vector of the satellite in the inertial coordinate system (ECI coordinate system).

Radar observations are based on the topocentric horizon coordinate system, such as SEZ (south-east-zenith) coordinate system. It is difficult to directly calculate the partial derivative of the observations with respect to the position vector in the ECI coordinate system. Therefore, the calculation of the $A$ matrix requires further use of the derivation chain rule to decompose.

$$\frac{\partial h}{\partial r_{\text{ECI}}} = \frac{\partial h}{\partial r_{\text{SEZ}}} \cdot \hat{r}_{\text{SEZ}} + \frac{\partial h}{\partial r_{\text{ECEF}}} \cdot \hat{r}_{\text{ECEF}}$$  \hfill (8)

where $\rho_{\text{SEZ}}$ is the cartesian position vector of the satellite in the SEZ coordinate system, represented by $(\rho_x, \rho_y, \rho_z)$; $\rho_{\text{ECEF}}$ is a range vector from radar to satellite represented in the Earth-Fixed coordinate system (ECEF); $r_{\text{ECEF}}$ is the satellite’s position vector in ECEF coordinate system.

After derivation [5], there is

$$\frac{\partial h}{\partial r_{\text{ECI}}} = \begin{bmatrix} \rho_x & \rho_y & \rho_z \\ \rho_x^2 + \rho_E^2 & \rho_x^2 + \rho_E^2 & \rho_x^2 + \rho_E^2 \\ -\rho_x \rho_E & -\rho_x \rho_E & \rho_x^2 + \rho_E^2 \\ \rho^2 \sqrt{\rho_x^2 + \rho_E^2} & \rho^2 \sqrt{\rho_x^2 + \rho_E^2} & \rho^2 \sqrt{\rho_x^2 + \rho_E^2} \end{bmatrix} \begin{bmatrix} \rho_x \\ \rho_y \\ \rho_z \end{bmatrix} \begin{bmatrix} 1 \\ \rho_x^2 + \rho_E^2 \\ \rho^2 \sqrt{\rho_x^2 + \rho_E^2} \end{bmatrix}$$  \hfill (9)

where $\begin{bmatrix} \text{SEZ} \\ \text{ECI} \end{bmatrix}$ is the transformation matrix from SEZ coordinate system to ECI coordinate system.

3.2. Calculating error state transition matrix

Suppose the motion equation of satellite is

$$\frac{d\mathbf{x}(t)}{dt} = \dot{x} = f(t, \mathbf{x}) = \begin{bmatrix} \mathbf{v}(t) \\ a(t, r, v, \beta) \end{bmatrix}$$  \hfill (10)

Then

$$\Phi = \frac{d}{dt} \begin{bmatrix} \dot{x} \\ \dot{x}_0 \end{bmatrix} = \frac{\partial f(t, \mathbf{x})}{\partial x} = \frac{\partial f(t, \mathbf{x})}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial x} = F\Phi$$  \hfill (11)

Where

$$\Phi = \begin{bmatrix} \dot{r} \\ \dot{\mathbf{x}}_0 \end{bmatrix}_{3\times 4} \begin{bmatrix} \dot{v} \\ \dot{\mathbf{x}}_0 \end{bmatrix}_{3\times 4} \begin{bmatrix} \dot{B} \\ \dot{\mathbf{x}}_0 \end{bmatrix}_{3\times 4}$$

$$F = \begin{bmatrix} 0_{3\times 3} & 1_{3\times 3} & 0_{3\times 1} \\ \frac{\partial a}{\partial \mathbf{r}} & \frac{\partial a}{\partial \mathbf{r}} & \frac{\partial a}{\partial \mathbf{r}} \\ 0_{3\times 3} & 0_{3\times 3} & 0_{3\times 1} \end{bmatrix}$$  \hfill (12)

Due to the iterative mode is used to solve the least-squares problem, accuracy requirements for the partial derivatives $\Phi$ and $F$ are generally more relaxed than that for the trajectory itself, it is common to
apply a simplified force model in the solution of the Eq. (11). The incorporation of the lowest-order zonal gravity field perturbation \((C_{2,0})\) already provides an acceptable minimum model.

3.3. Calculating partial derivative of the earth’s gravitational acceleration with respect to the state vector

The acceleration of the earth’s gravitational force (including non-spherical perturbation) is only related to position, and has nothing to do with velocity and ballistic coefficients, and its calculation is usually carried out in the ECEF coordinate system. The partial derivative matrix of the earth's gravity with respect to position \(\frac{\partial a}{\partial r}\) has symmetry, and the sum of the diagonal elements is zero, so that the independent elements of the matrix are reduced from 9 to 5 in the calculation. In the ECEF coordinate system, these elements can be easily obtained from Eq. (13). Specifically, the expression derived by Cunningham [1] can be used, which will not be repeated here due to content capacity limitation.

\[
\frac{\partial a}{\partial r} = \sum_{n,m} \frac{\partial a_{nm}}{\partial r}
\]  

(13)

Finally, the partial derivative matrix in the ECEF coordinate system needs to be converted to the ECI coordinate system. In the case of ignoring Coriolis force and centrifugal force, there is the following relationship

\[
\left(\frac{\partial a}{\partial r}\right)_{\text{eci}} = \left[\text{ECI}^{-1} \left(\frac{\partial a}{\partial r}\right)_{\text{ecef}} \text{ECI}\right]
\]  

(14)

3.4. At the current time, after the matrix \(F\) is calculated, \(P\) matrix can be solved by the numerical differential equation method. Calculating partial derivative of atmospheric drag acceleration with respect to state vector

According to Eq. (5), the partial derivative of the atmospheric drag acceleration with respect to the ballistic coefficient is

\[
\frac{\partial a}{\partial B} = -\frac{1}{2} \rho v_i v_T
\]  

(15)

The partial derivative with respect to the velocity vector is

\[
\frac{\partial a}{\partial v} = -\frac{1}{2} B \rho \left(\frac{\partial v_i}{\partial v} v_i + v_i v_T, 1\right) = -\frac{1}{2} B \rho \left[ v_i v_T^T + v_i, 1 \right]
\]  

(16)

Note that the 1 in the equation is a diagonal matrix, not the number 1.

The partial derivative with respect to the position vector is

\[
\frac{\partial a}{\partial r} = -\frac{1}{2} B v_i \frac{\partial v_i}{\partial r} - \frac{1}{2} B \rho \left(\frac{\partial v_i}{\partial r} v_i + v_i \frac{\partial v_i}{\partial r}\right) = -\frac{1}{2} B v_i \rho \frac{\partial \rho}{\partial r} - \frac{\partial a}{\partial r} X(\omega_b)
\]  

(17)

Where

\[
X(\omega_b) = \begin{bmatrix}
0 & -\omega_z & +\omega_y \\
+\omega_y & 0 & -\omega_x \\
-\omega_z & +\omega_x & 0
\end{bmatrix}
\]  

(18)

\(\frac{\partial \rho}{\partial r}\) differs with different atmospheric models. It only has analytical expressions under some simpler models. For example, the analytical expressions under the Jacchia71 atmospheric model can be referred to [7]. Because the expressions are too complicated, I will not repeat them here. For the case where there is no analytical expression, the \(H\) matrix can be solved by the finite difference method [5].
4. Introduction to ODTK software
ODTK is a commercial software for spacecraft orbit determination developed by AGI. The software is mainly used for space mission orbit determination simulation and orbit measurement data processing. It has the following commonly functions: measurement data simulation; acquisition and process satellite measurement data; parameter estimation (satellite status, equipment system error, environmental parameters, etc.); provide satellite ephemeris and its covariance, etc.

ODTK software adopts a layered structure, and its composition structure is shown in Fig. 1. The ODTK engine contains all dynamics-related modules and data, and shares the dynamics module with STK (STK is another excellent satellite modeling software of AGI); the AGI technology platform provides the component framework and extended applications of ODTK; The ODKT graphical user interface is built on the AGI technology platform and contains the necessary components to realize the ODTK function. The main software interface is shown in Fig. 2. This structure keeps the mathematical models of ODTK and STK products consistent; at the same time, it maintains the separation of graphical user interface and engine software, thereby reducing the complexity of the software; and enhancing the user interface to make it look consistent with other AGI products.

The advantages of ODTK include: simultaneous processing of multi-satellite data; ability to process tracking data during thrust (including limited thrust and pulsed thrust) and to provide deviations in actual conditions, including measurement elements, atmospheric drag; automatic and efficient execution of operations.

5. Ballistic coefficient estimation with ODTK
The basic idea of using ODTK software to estimate ballistic coefficient is: first use ODTK software to simulate a segment of radar observation data to a satellite, including the ballistic coefficient settings related to the discussion in this paper; then use ODTK software to implement orbit determination for the simulation data; finally compare the estimated ballistic coefficient with the set value during data simulation to verify the correctness of the operation process.

The operation method of ODTK software for ballistic parameter estimation is introduced as follows.

5.1. Data simulation
Step 1: Establish a data simulation task scenario
Click the scene button on the main interface to create a new data simulation scene, named Simulator1. Then click the satellite button from the shortcut bar to add a satellite, named Sat1; click the tracking system button and the station button to add a station (representing radar), named Fac1; click the data simulator button to add a data simulation task, named Sim1. The scene configuration after execution is shown in Fig. 3.
Step 2: Set object properties

Double-click the station component Fac1 to set its geodetic coordinate attributes as latitude=40°, longitude=130°, height=100m. Double-click the satellite component Sat1, set its orbital epoch and orbital elements, as shown in Fig. 4; set its atmospheric drag properties as follows, the ForceModel.Drag.Model.SpecMethod property is set to "Ballistic Coeff", ForceModel.Drag.Model.BallisticCoeff property is set to 0.08 (the default value is 0.044, the purpose of modification is to compare with the later estimation), as shown in Fig. 5. Double-click the data simulation component (Sim1), set its ProcessControl properties, including simulation start, end and interval time, as shown in Fig. 6; set its Simulator1.Output.Measurements.Filename property, which is the storage path of the simulation data; set its MeasTypes property, select distance, azimuth, and elevation parameters; set the Simulatorl.ErrorModeling.NoDeviations property to true.

Step 3: Implement data simulation tasks

Select the data simulation component, click the run button on the shortcut bar, observe the simulation operation from the "Message Viewer" column at the bottom of the software, and call the "Measurement Data Summary" report to view the simulation data overview after execution, as shown in Fig. 7. There are 49 data points are generated. The detailed content of the simulation data can be viewed by clicking the data preview button on the shortcut bar.

5.2. Ballistic coefficient estimation

Step 1: Establish a ballistic coefficient estimation task scenario

Click the scene button on the main interface to create a new ballistic coefficient estimation scene, named Scenario1. Then click the satellite button in the shortcut bar to add a satellite, named Sat1; click the tracking system button and the station button to add a station (representing radar) Fac1; click the initial orbit determination button and the least-squares orbit improvement button to add

![Figure 3. Configuration of data simulation scenario](image)

![Figure 4. Settings of satellite’s orbit elements](image)

![Figure 5. Settings of atmospheric drag perturbation](image)

![Figure 6. Settings of simulation time parameters](image)

![Figure 7. Simulation data report](image)
a initial orbit determination task (Iod1) and an orbit improvement task (Ls1). The purpose of adding the initial orbit determination task is to provide an initial value for the orbit improvement task. The scene configuration after execution is shown in Fig. 8.

**Step 2: Set object properties**

Double-click the scenario component Scenario1, set the Scenario1.Measurements.Files property to the data file generated by the previous data simulation. Double-click the station component Fac1, and set its location attribute to be consistent with the station location attribute in the data simulation. Double-click the initial orbit determination component Iod1, set the SelectMeasurements property, that is, select 3 observation records for the initial orbit determination. Double-click the orbit improvement component Ls1, edit its Ls1.Stages attribute, and add an orbit improvement operation, as shown in Fig. 9. Pay attention to setting the EstimateBCoeff attribute to true, indicating that the satellite’s ballistic coefficient is to be estimated in this orbit improvement operation.

**Step 3: Implement orbit improvement**

First select the initial orbit determination component, click the run button on the shortcut bar to get the initial orbit elements of the satellite, and use this value as the initial value for the least-squares orbit improvement, then select the orbit improvement component, and click the run button again. The "Message Viewer" column shows that the orbit elements have been improved after 6 iterations. Double-click the Ls1.Output.ForceModelList property to view the estimated ballistic coefficient from the pop-up dialog box, as shown in Fig. 10. The value is 0.0802605, which is very close to the simulation setting value of 0.08, indicating that the parameter estimation is effective.

6. Conclusions

This article introduces the ballistic coefficient estimation theory based on the orbit improvement method, and describes in detail the operation steps and results of the ballistic coefficient estimation using ODTK software. The results show that the ODTK software can quickly and accurately calculate the satellite’s ballistic coefficient. In the next step, the influence of different atmospheric density models on the estimation of ballistic coefficients will be studied.

References

[1] O. Montenbruck, E. Gill. (2000) Satellite Orbits - Models, Methods and Applications. Springer-Verlag Inc., Berlin.
[2] David J. G, Roberto A, Aleksander A. L. (2017) Ballistic coefficient estimation for reentry prediction of rocket bodies in eccentric orbits based on TLE data. Mathematical Problems in Engineering, Volume 2017, Article ID 7309637.
[3] J. Y. Cheng, S. Li, P. Zhang. (2016) Low earth orbit space debris ballistic coefficients and their applications. Infrared and Laser Engineering, 45(11): 1-6.
[4] Sang J, Bennett J C, Smith C H. (2013) Estimation of ballistic coefficients of low altitude debris objects from historical two line elements. Advances in Space Research, 52(1): 117-124.
[5] Vallado D. A. (2013) Fundamentals of Astrodynamics and Applications (4th ed). McGraw-Hill, New York.
[6] Vallado D. A, Hujsak R. S, Johnson T. M, etal. (2010) Orbit determination using odtk version 6. ESA/ESAC astronomy centre, Madrid, Spain, 3-6.

[7] W. Wei, Z. J, Yu. (2007) Spacecraft Orbit Determination-Models and Algorithms. National Defense Industry Press, Bei jin. pp. 202-219.