

Military Technical College
Kobry El-Kobbah
Cairo, Egypt



10th International Conference
On Aerospace Sciences &
Aviation Technology

INTERACTIVE VISUALIZATION OF SATELLITE FOOTPRINT IN A SIMULATED ENVIRONMENT

Mohammed* S. A , Ahmed** M. H., Mahmoud*** M.G., Safwat**** H. H.

ABSTRACT

Satellites are complicated systems designed to accomplish different missions in space. Simulating satellite motion using digital computers gives the ability to visualize dynamically its motion in space, as well as the areas of the earth's surface covered by its service.

In this paper fundamentals of astrodynamics were applied to implement a software system that simulates satellite motion in 3D space. In addition, it interactively computes and visualizes the satellite's ground trace and footprint coverage on the rectangular map of the earth.

KEYWORDS

Satellites, Simulation, Visualization, Footprint.

* Dean of Faculty of Computer & information sciences, Ain Shams University, Cairo Egypt
** Vice dean of Faculty of Computer & information sciences, Ain Shams University, Cairo Egypt.
*** Associate professor, Dpt. of Scientific computing, Ain Shams University, Cairo Egypt.

INTRODUCTION

A satellite is known as any object that orbits or revolves around another object. In 1957, the Soviets launched the first "artificial satellite," Sputnik 1. Today, thousands of satellites are in orbit.

One of the characters of the satellite is that it provides a sustained observing platform outside the atmosphere for studying the earth and its environment, as well as outer space. Their applications encompass global communication, navigation, weather reconnaissance, planetary studies, geodesy, studies of biological organisms in a weightless environment, cosmic ray and solar physics, ocean and land surveillance for marine and mineral resources, agricultural surveys, location of sites for archaeological explorations, etc. There is no doubt that the future will see an increasing number of satellites and space probes being used for ever-expanding applications.

Many steps of analysis, design and even building a prototype precede the process of sending a satellite into space in order to accomplish a certain mission. Simulation can be used as a powerful tool to study and compare alternative designs or to troubleshoot existing systems. With simulation models, it's possible to explicitly visualize, how a new system might behave before the prototype is even completed. The ability to easily construct and execute models and to generate statistics and animations about results has always been one of the main attractions of simulation software.

In this paper, we will introduce in section two the parameters that describe the satellite orbit. Then an algorithm for computing the satellite's footprint is introduced in section three. In section four we explore some features of the implemented simulation software. In the last section we will introduce some experimental results of the tested case studies.

SATELLITE ORBIT SPECIFICATION

By definition, the path formed by the satellite movement in space is called its orbit. There are infinite numbers of possible orbits for an earth satellite. While there are special orbits that are designed for specific purposes. Fig. 1 shows some famous classes of orbits including geostationary orbits (GEO), Low Earth Orbits (LEO), Medium Earth Orbits (MEO), and Highly Elliptical Orbits (HEO) [1].

Placing a satellite in a specific orbit in space requires a complete description of the orbit's size, shape, and orientation. *Keplerian elements* or simply *Keps* consist of seven numbers, *Inclination*, *Right Ascension of Ascending Node (RAAN)*, *Argument of Perigee*, *Semi-major axis*, and *Eccentricity* are used to describe the orbit itself. The remaining two elements; *the Epoch and True anomaly*, define the satellite position in space. You can refer to [2], [6] for more detailed information.

Generally, these parameters define a conic section, orient it about the earth, and place the satellite on the ellipse at a particular time. These Keps; as its name says,

are based on the three laws of planetary motion described by the German Astronomer Johan Kepler. Although they describe the motion of planets around the sun, they of course apply also for an artificial satellites moving around the earth. These laws are based on two assumptions:

- 1) The bodies are spherically symmetric. This enables us to treat the bodies as though their masses were concentrated at their centers (point masses).
- 2) There are neither external nor internal forces acting on the system other than the gravitational forces which act along the line joining the centers of the two bodies.

Obviously, describing the orbital elements requires a suitable coordinate system. Various celestial coordinate system can be used, however we have chosen EFC (Earth Fixed Centered) coordinate system as a reference frame to carry out the calculations. In this system, the origin is at the center of mass of the Earth, the Z-axis corresponds to the mean north celestial pole, the X-axis is based on the Greenwich meridian, the Y-axis is the cross product of the Z and X axes, and the equatorial plane is chosen as the fundamental plane[3]. Fig. 2 shows the principle axes forming this coordinate system.

THE SATELLITE'S FOOTPRINT

Definition

The satellite's footprint can be defined as the portion of the earth's surface over which the satellite operates. For a communications satellite, the footprint is the area on the earth where transmissions can be sent to or received from the satellite. For an imaging satellite, the footprint also describes the area of the earth that is imaged by the satellite in addition to its communication footprint. The sensors attached to the satellite has a predefined field of view defined by its *Nadir Angle* (α). Therefore, the footprint (or coverage area) of the satellite can also be defined as the intersection of the sensor's view pattern and the surface of the earth. Fig.3 shows a depiction of the satellite's footprint.

Coverage Geometry

The diagram shown in fig. 4 illustrates the coverage geometry or field-of-view of a satellite in orbit above a spherical planet [4]. Where:

- | | | | |
|-----------|---|---------|-----------------|
| α | = Nadir Angle, | β | = Central Angle |
| θ | = Elevation Angle, | s | = Slant Range |
| r_{sat} | = Distance from center of the earth to satellite position | | |
| r_e | = radius of the Earth | | |

Given the nadir angle α , the central angle, elevation angle and slant range can be determined from the following set of equations [4]:

$$\beta = \sin^{-1} \left(\frac{r_{sat}}{r_e} \sin \alpha \right) - \alpha \quad (1)$$

$$\theta = \cos^{-1}\left(\frac{r_{sat}}{r_e} \sin \alpha\right) \quad (2)$$

$$s = r_{sat} \cos \alpha - \sqrt{r_e^2 - r_{sat}^2 \sin^2 \alpha} \quad (3)$$

Footprint Coverage Computation

The satellite position in space at any moment of time can be determined through a series of complicated computations[6]. Given this position (X, Y, Z), in EFC coordinate system, as well as the value of the Nadir Angle (α), the footprint of the satellite can be calculated. The idea is based on the spherical earth shape and the conic shape of the satellite's sensor's pattern. That is, the intersection between the sphere (the earth) and the cone (satellite's field of view) will form the satellite's footprint.

Consider that the satellite is in a position that is directly over the center of the earth's North Pole. When viewed from above the North Pole, the shape of the intersection plane will be a complete circle. The radius of this circle depends on the value of α .

However, this is not always the case. The satellite moves around the continuously rotating earth resulting in different coverage areas depending on the shape and orientation of its orbit. But, it may be easier if we start computing the footprint of a satellite from the previously illustrated position (the satellite above the North Pole). In this case, the dashed line shown in fig. 4 represents the intersection plane between the satellite's conical field of view and the surface of the spherical earth. Here, z_o denotes the distance between the intersection plane and the center of the earth. Notice that all intersection points lie in the plane defined by the value of z_o .

Given the value of α , calculating β will be straight forward using equation (1). Then, computing the coordinates of the points forming the boundary of the circle requires the value of the radius of the circle r as well as the value of z_o . Using the geometry illustrated in fig. 4, those values can be calculated directly from the value of β using the following equations (4) and (5).

$$z_o = r_e \cos \beta \quad (4)$$

$$r = r_e \sin \beta \quad (5)$$

It's clear now that the z-coordinate of all the intersection points is z_o . We can make use of the circular shape of the intersection plane to compute the values of the other two coordinates (x, y). Considering that the point (x, y) makes a polar angle ω° with the reference axis, then the (x, y) values can be computed using the following couple of equations:

$$x = r \cos \omega \quad (6)$$

$$y = r \sin \omega \quad (7)$$

Still, the previously calculated points represent the footprint of the satellite at the assumed position. So, a transformation operation needs to take place in order to map those points to their right position around the actual satellite position. This can be done using a standard coordinate transformation operation. Assuming that the two coordinate systems have the same origin (the Earth Center) but they differ in the orientation of the z-axis, where the z-axis of the first one points toward the North Pole and the second z-axis points toward the current satellite position. All what we need now is to set up the transformation matrix that brings the two z-axes into alignment [5].

Let the direction of the new z-axis is denoted by the unit vector u , which can be calculated from the following equation, assuming that the two end points of V is the Earth center or (0,0,0) and the satellite position (X,Y,Z).

$$u = \frac{V}{|V|} \tag{8}$$

Now, we can form the set of local unit vectors (u_x, u_y, u_z) for the new coordinate system as follows:

$$u_z = u \tag{9}$$

$$u_y = \frac{u \times u_x}{|u \times u_x|} \tag{10}$$

$$u_x = u_y \times u_z \tag{11}$$

Then, we can use these unit vectors to compose the transformation Matrix (M) expressed in (12).

$$M = \begin{bmatrix} u_{x1} & u_{x2} & u_{x3} \\ u_{y1} & u_{y2} & u_{y3} \\ u_{z1} & u_{z2} & u_{z3} \end{bmatrix} \tag{12}$$

where,

$$u_x = (u_{x1}, u_{x2}, u_{x3})$$

$$u_y = (u_{y1}, u_{y2}, u_{y3})$$

$$u_z = (u_{z1}, u_{z2}, u_{z3})$$

Finally, the newly transformed intersection points (x',y',z') can be mapped to the rectangular Earth map by computing their latitude(ϕ) and longitude(λ) using the following set of equations:

$$\phi = \tan^{-1} \left(\frac{z'}{\sqrt{x'^2 + y'^2}} \right) \tag{13}$$

$$\lambda = \text{atan2}(y', x') \tag{14}$$

The result will be a set of longitude and latitude values that can be drawn directly on the map of the earth in order to visualize the instantaneous satellite's footprint.

Footprint Algorithm

The following algorithm describes a function that may be used for computing the satellite's footprint coverage for a specific satellite position. The output of this function can then be combined with a visualization operation in order to project the computed points on the map of the earth to gain more understanding of the result.

function Compute Satellite Footprint

Input:

α = Nadir Angle.

(X, Y, Z) = Satellite's Position in EFC coordinate system.

Output:

CovPoints = Array of footprint coverage points (longitude, Latitude)

BEGIN

1. Compute the value of r_{sat}

$$r_{sat} = \sqrt{X^2 + Y^2 + Z^2}$$

2. Compute the central Angle (β)

$$\beta = \sin^{-1} \left(\frac{r_{sat} \cdot \sin \alpha}{r_e} \right) - \alpha$$

3. Find the value of z_o

$$z_o = r_e \cos \beta$$

4. Compute the radius of the coverage circle

$$r = r_e \sin \beta$$

5. for $\omega = 1^\circ$ to 360°

- 5.1 Compute the point (x, y) on the circle circumference

$$x = r \cos \omega$$

$$y = r \sin \omega$$

- 5.2 Transform the point (x, y, z_o) to represent the actual coverage point

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} u_{x1} & u_{x2} & u_{x3} \\ u_{y1} & u_{y2} & u_{y3} \\ u_{z1} & u_{z2} & u_{z3} \end{bmatrix} \begin{bmatrix} x \\ y \\ z_o \end{bmatrix}$$

- 5.3 Find the Longitude (λ) and Latitude(φ) values of the new point

$$\varphi = \tan^{-1} \left(\frac{z'}{\sqrt{x'^2 + y'^2}} \right)$$

$$\lambda = \text{atan2}(y', x')$$

- 5.4 Add the point (λ , φ) to CovPoints

$$\text{CovPoints}[\omega] \leftarrow (\lambda, \varphi)$$

End Loop

6. Return CovPoints

END

EXPLORING SOFTWARE CAPABILITIES

The previous Algorithms were used to develop a satellite simulation and visualization software. The Implementation tool used were MS Visual-C++6 along with OpenGL graphics libraries. The entities of the system were designed and implemented using an object oriented approach.

The program introduces the user with a really friendly user interface that enables the user to input the seven orbital elements as well as editing some simulation parameters by means of interactive dialogs. Each time the user changes one of these parameters the program adjust its state according to the new changes.

Once the orbital elements are specified the orbit and the satellite is rendered around the spherical model of the earth (fig. 6). The user then can change the view point easily using the mouse. More and more, zooming in and out also facilitates visualizing the scene in the best way. The user also can switch on the animation capability to view the actual satellite motion in its orbit around the earth.

The satellite's ground track can also be visualized on a rectangular earth map displayed in a separate window, allowing the user to trace the satellite's projection on earth while it is in motion. Fig. 7 shows the ground track computed for the famous Russian Molniya satellite that was designed to observe the northern hemisphere [1].

Fig. 8 shows another capability offered by the developed software, which allows the user to visualize the orthogonal footprint as the satellite moves in its orbit.

EXPERIMENTAL RESULTS

Fig. 9 shows the Footprint of a satellite in a circular orbit with zero inclination and 30° Nadir Angle, while Fig. 10 shows the Footprint of a satellite in a polar orbit (Perigee = 5000 km, Apogee = 8000 Km, 90° inclination) passing just over the north pole. The Nadir Angle is 20°.

Fig. 11 shows the Footprint of a satellite in a LEO orbit (Perigee = 5000 km, Apogee = 10000 Km, 45° inclination) moving near its perigee. The Nadir Angle is also 20°.

CONCLUSIONS

The laws governing the satellite motion are complicated and the relations between them are so dynamic.

We have developed a software system that models the satellite motion in 3D space, and provides a visualization tool to view its track on the earth along with the areas covered by its service as it moves around the earth. The software provides a very user-friendly interface with animation capability. This software is recommended to be used for interactive orbit design.

REFERENCES

- [1] Martin W. Lo, *Satellite Constellation design*, Computing in Science and Engineering journal, January-February 1999.
 - [2] Nick Dargahi, *The Space Simulator Pilot's handbook*, MIS Press/IDG, 2000.
 - [3] *Landsat 7 Image Assessment System (IAS)*; Geometric Algorithm Theoretical Basis Document, version 3.2, 1998.
 - [4] David Eagle, *Orbital Mechanics with Numerit*, Technical Notes, 2002.
 - [5] Donald Hearn, M. Pauline Baker, *Computer Graphics*, 2nd edition, Prentice Hall, New Jersey, 1997.
 - [6] M. Said Abdel-Wahab, Ahmed Mohammed Hamad, Mahmoud El-Gamal, Safwat Helmy M. Hamad, *Satellite Motion Simulation and Visualization*, the 7th IEEE International Conference On Intelligent Engineering Systems (INES), 2003.
-

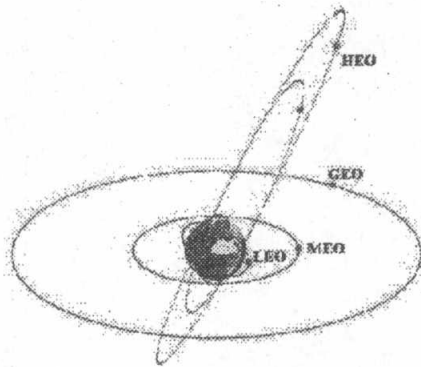


Fig. 1 Special Orbits

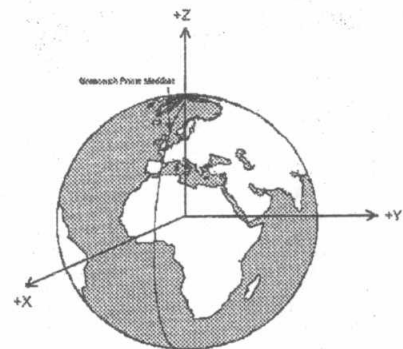


Fig. 2 EFC Coordinate System

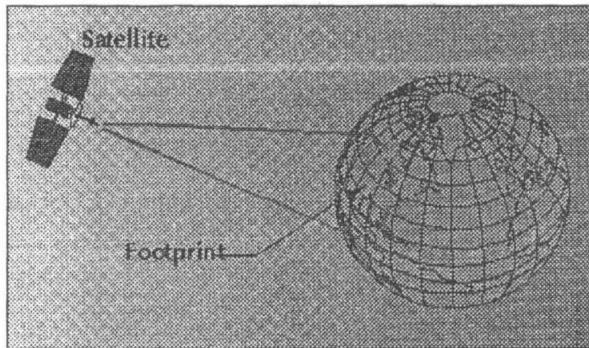


Fig. 3 Satellite's Footprint

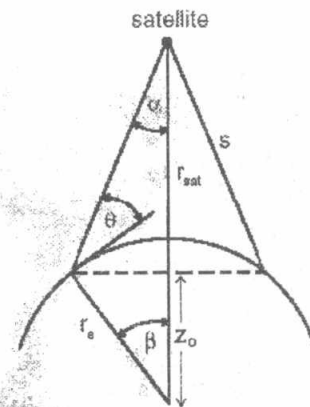


Fig. 4 The Satellite's Field of View

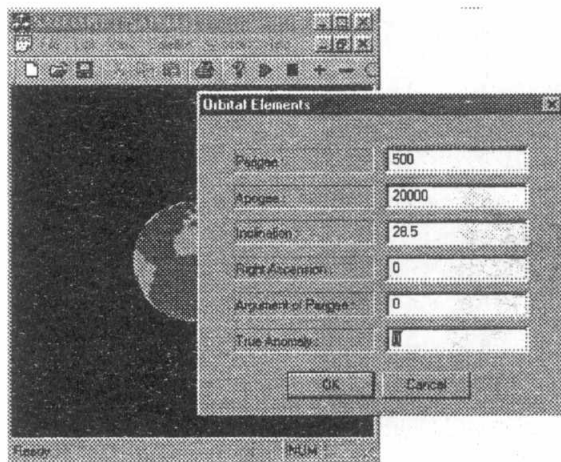


Fig. 5 Orbital Elements Dialog

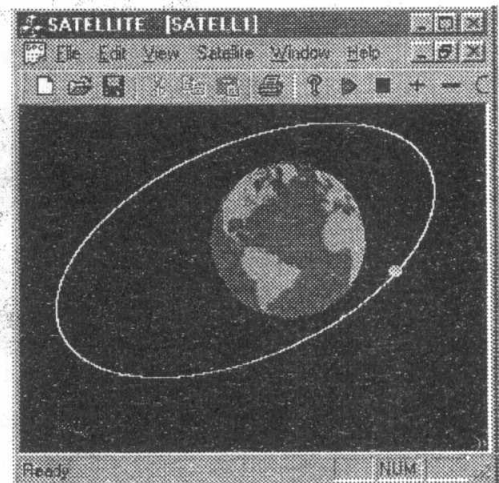


Fig. 6 Satellite Orbit Visualization

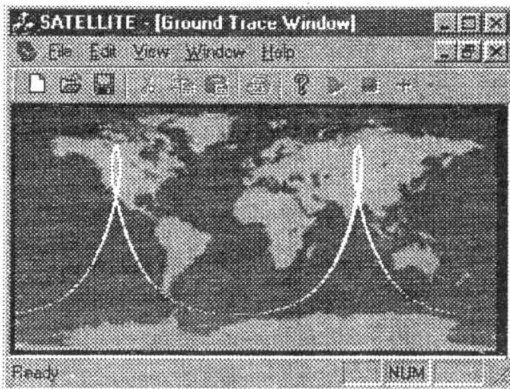


Fig. 7 Molniya ground track

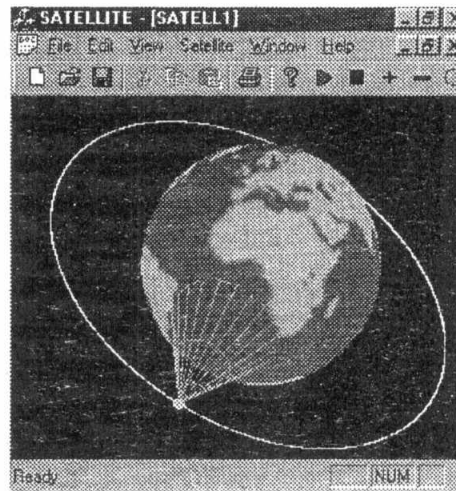


Fig. 8 Orthogonal footprint Visualization

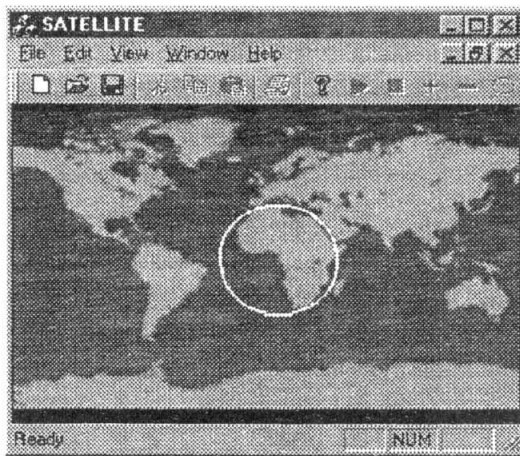


Fig. 9 Circular Orbit Footprint

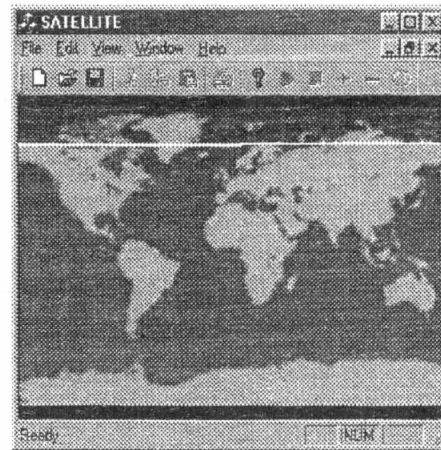


Fig. 10 Polar Orbit Footprint

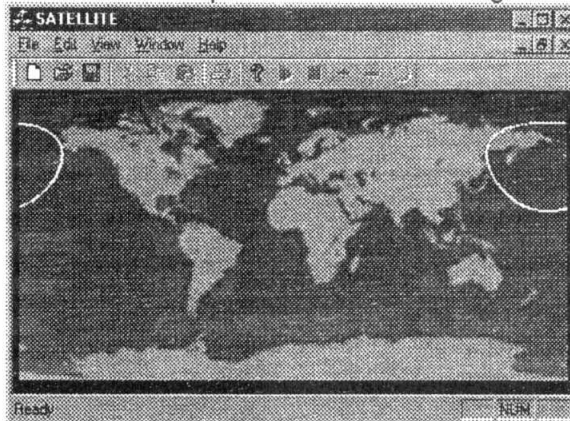


Fig. 11 LEO Orbit Footprint