Orbital Visualization Software – MagTrack

Kacper Wardzynski | The University of Manchester

kacper.wardzynski@student.manchester.ac.uk

"Space [...] is big. Really big. You just won't believe how vastly hugely mind-bogglingly big it is. I mean, you may think it's a long way down the road to the chemist, but that's just peanuts to space."
~ D. Adams, The Hitchhiker's Guide to the Galaxy

Motivation

Space is vast. Compared to the expanse of the Universe, the Low Earth Orbit is small and congested with satellites. This introduces a need for accurate tracking data to prevent collisions, which would create even more space debris.

While tracking data is widely and commonly available in the Age of Information, making practical use of it requires specialized knowledge and a series of calculations.

The project aims to bridge the gap between accessing the tracking data and allowing the user to obtain real-time information about the spacecraft's orbit, providing a user-friendly way to display its position as a function of time.

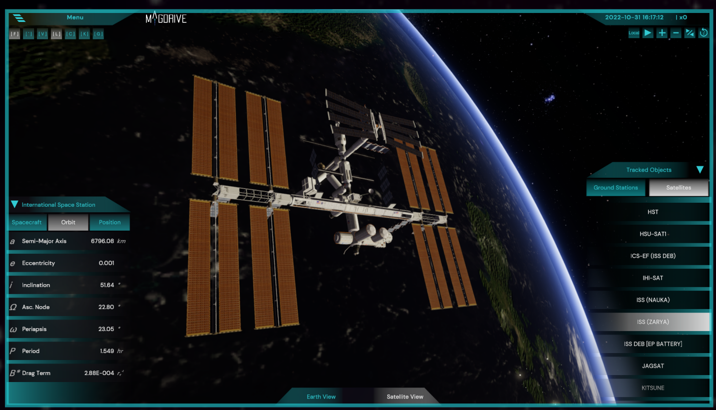

Fig.1 The screen taken from the MagTrack software, showing a close up on the ISS model. Elements of its basic interface can be seen.

Functionality

- Preview spacecraft orbital elements.
- Predict spacecraft position as function of time.
- Support of Low, Medium, High and Polar Earth Orbits.
- Track any number of spacecraft simultaneously.
- Update spacecraft's TLE from the online database.
- Visualize ground station positions.
- Predict time and the duration of satellite's overpass over selected ground station.
- Skip, reverse, and stop the simulation time.

Fig.2 The screenshot taken from the MagTrack software, shows Earth along with several of visualized orbits. The user interface on the right shows the list of currently tracked objects.

Method

The MagTrack was developed within the Unity Engine, making use of its robust graphics engine, and was written in C# programming language.

The software takes in the input of Two-Line Elements, denoted as TLE's – format of data containing the orbital elements of an Earth-orbiting object. The input TLEs are processed through the equations of Keplerian Orbit to propagate the position for a given date time, predicting the satellite's position at that time.

The program allows for requesting the data from the CelesTrak TLE database to obtain the latest, accurate orbital elements to ensure the coherence of the predicted position with the real one.

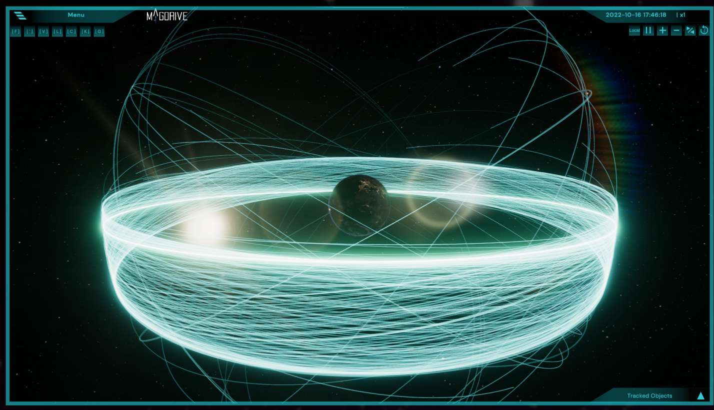

Fig.3 The screenshot taken from the MagTrack software, shows software tracking a large number of satellites with geosynchronous orbits.

Keplerian Orbit

The model employed in the program is the model of a Keplerian Orbit – a simple, non-perturbed elliptical orbit around the central body. It is described through six parameters, containing information about the orbit as well as the satellite position itself.

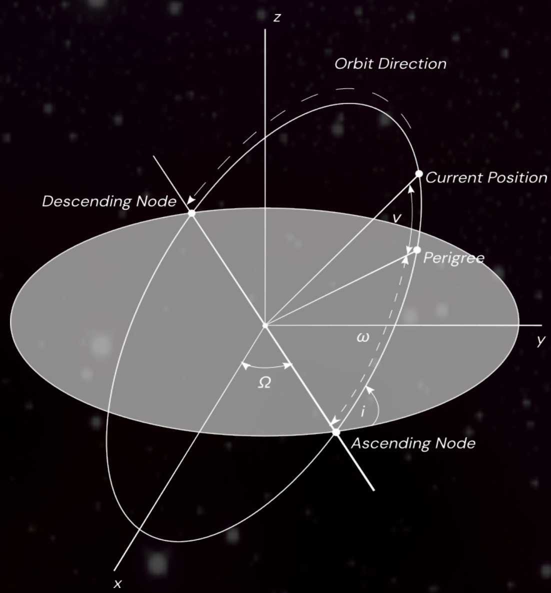

Fig.4 Figure depicting the Keplerian Orbital elements.

Orbital Ellipse

- Semi-Major axis a
- Eccentricity e

Orientation of the ellipse

- Longitude of the ascending node Ω
- Argument of periapsis ω
- Inclination i

Position of the spacecraft is given by True Anomaly ν, which is defined as an angle between periapsis and spacecraft position on the orbit.

Orbit Propagation

The position of the spacecraft as a function of time is obtained by computationally solving the Kepler's Equation.

$$M = E - e \times \sin(E) \qquad (1)$$

Where M is the Mean Anomaly and E is the eccentric anomaly. To accommodate for the time difference, another term is added to the mean anomaly to represent the path that the satellite has traversed since the coordinates were taken.

$$M(t) = M_0 + \Delta t\sqrt{\frac{\mu}{a^3}} \qquad (2)$$

The Δt is the time difference in seconds and μ is the gravitational parameter of Earth.

The Eq. (1) is computationally solved through the use of Newton-Rhapson algorithm, where

$$E_1 = M, \quad E_{k+1} = M + e\sin(E_k). \qquad (3)$$

The Eq. (3) is repeated until convergence is reached, yielding value for eccentric anomaly. Then the true anomaly is calculated from the relation between anomalies.

$$\nu = \arccos\left(\frac{\cos(E) - e}{1 - e\cos(E)}\right) \qquad (4)$$

Ground Stations

Within the program, the user can freely add any ground station by inputting its longitude or latitude or simply creating a database containing a set of ground stations.

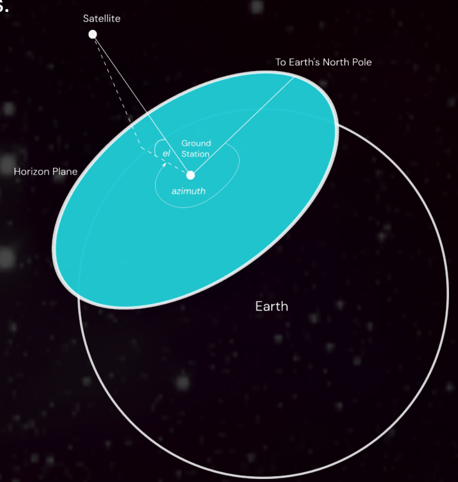

Fig.4 Figure representing the horizon plane geometry. Elevation of the satellite is denoted as *el*. The blue plane represents the horizon plane. The satellite is considered to be overpassing ground station when it is above its horizon plane.

Each ground station allows calculating the time of the overpass of selected satellites, where the overpass is defined as the period when the satellite is above the ground's station horizon angle, meaning a free, unobstructed line of sight from the antennae dish to the satellite.

Fig.5 The screenshot taken from the MagTrack software. Shows the overpass calculation interface, with overpasses calculated for few satellites at ESA Cebreros station.

This allows for free communication with the spacecraft. To visualize the overpass, radar charts are utilized within the software. The program also allows exporting the overpass data to the .csv format should the user desire to do so.

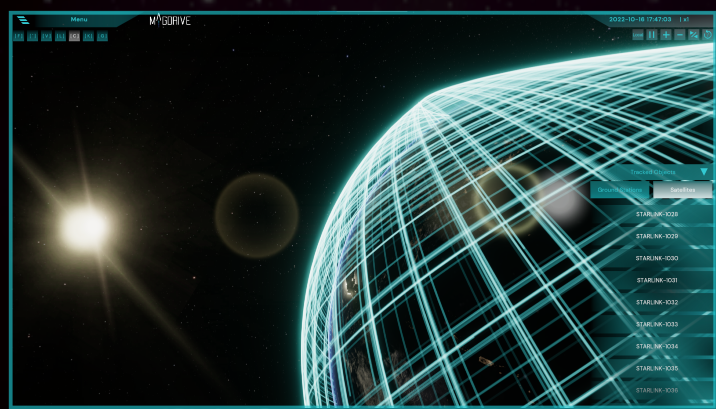

Fig.6 The screenshot taken from the MagTrack software. Shows approximately half of the SpaceX Starlink satellites orbits, as of 21/07/2022, when the data was taken.

Summary

Throughout the project, orbital visualization software was developed that allows tracking of the desired spacecraft, displaying its orbital data. The automatization of updating spacecraft data allows for the long-term operation of the program.

Further work could include integrating the SGP4 Propagator instead of the Keplerian Orbital model to achieve greater accuracy of the tracking.

Project developed under the scheme
Space Placements in Industry 2022
working for Magdrive