

# TIS (Tracking interface for satellites): Rastreo satelital y predicción de órbitas a base de modelos de perturbaciones simplificadas en Python

Arnulfo de Jesús Montoya Delgado<sup>a</sup>, Víktor Iván Rodríguez Abdalá<sup>a</sup>, Jorge Flores Troncoso<sup>a</sup>, Remberto Sandoval Arechiga<sup>a</sup>, Salvador Ibarra Delgado<sup>a</sup>, José Ricardo Gómez Rodríguez<sup>a</sup>

<sup>a</sup>Universidad Autónoma de Zacatecas, Unidad Académica de Ingeniería Eléctrica,  
Centro de Investigación, Innovación y Desarrollo en Telecomunicaciones  
Av. López Velarde 801, Col. Centro, Zacatecas, Zac., México, 98000.  
[abdala@uaz.edu.mx](mailto:abdala@uaz.edu.mx),

2018 Published by *DIFU*<sub>100ci</sub>@ <http://difu100cia.uaz.edu.mx>

---

## Resumen

El desarrollo de aplicaciones científicas en lenguajes de programación de alto nivel como Python se ha dado por la integración de librerías que facilitan el análisis y el procesamiento numérico de datos, lo que permite el desarrollo de código científico y de ingeniería en tiempos relativamente cortos para ser utilizado en el campo de trabajo. Las aplicaciones para rastreo satelital y predicción de órbitas suelen ser propietarias, por lo que generar una alternativa en software con licencia libre y abierta es una alternativa viable para fomentar la investigación científica a diferentes niveles, permitiendo al usuario final la personalización de la aplicación de acuerdo a requerimientos específicos.

*Palabras clave:* Rastreo Satelital, predicción de órbita, modelos de perturbaciones.

---

## 1. Introducción

Hoy en día el desarrollo de proyectos de software para rastreo satelital y predicción de órbitas es realizable en lenguajes de programación de alto nivel de código abierto a través de librerías especializadas, tal es el caso de PyEphem, el cual es una librería de Python que provee herramientas de cálculo astronómico de grado científico [1].

Dada una fecha y ubicación en la superficie de la Tie-

rra, esta librería puede calcular la posición de diversos cuerpos celestes incluidos satélites artificiales con la ayuda de elementos orbitales, lo que permite la predicción de órbitas en una determinada fecha. El cálculo numérico detrás de PyEphem se basa en la aplicación de astronomía XEphem [2].

Los satélites artificiales tienen diferentes tipos de órbitas de acuerdo al propósito de la misión, para ello se utiliza un formato de conjunto de elementos en dos líneas llamado TLE (Two-Line Element Set), este es

un formato utilizado por NORAD (North American Aerospace Defence Command) para la identificación de elementos orbitales de los satélites [3] [4], este formato cuenta con tres líneas de datos, tal como se muestra en la Tabla 1.

La primera línea (línea 0) se compone de 24 caracteres, en ella se indica el nombre del satélite. Las dos líneas siguientes contienen los elementos orbitales y valores para la identificación del satélite, tal como se observa en las tablas 2 y 3 [5].

Los elementos orbitales se actualizan dependiendo del tipo de órbita, en el caso de las órbitas bajas los datos deben actualizarse una vez por semana y para órbitas altas una vez por mes [6].

### 1.1. Modelos matemáticos

Para el cálculo de la posición de un satélite en el espacio, el proyecto TIS utiliza los modelos matemáticos de NORAD SGP4/SDP4 [7]. Estos modelos utilizan el formato TEL para determinar las coordenadas de los objetos que orbitan la Tierra, considerando que las coordenadas tiene su origen en el centro de la masa terrestre.

Para ello, SDP4 determina primero la perturbación en el espacio profundo considerando los efectos de la gravedad y el efecto secular de arrastre en la longitud de nodo ascendente, para el cálculo de las predicciones agrega el espacio profundo lunar y la predicción solar de los elementos orbitales que son la velocidad orbital media, excentricidad, inclinación, argumento de perigeo, longitud del nodo ascendente y anomalía media.

## 2. Entorno de programación

El programa de rastreo satelital se basa en el lenguaje de programación Python, el cual permite la integración de múltiples librerías creando un entorno interactivo capaz de ser modificado con diferentes funciones, las librerías utilizadas para el desarrollo del proyecto son:

- PyEphem.
- PyQt.
- Matplotlib.
- Bokeh.
- Folium.
- Hamlib.
- Socket.

### 2.1. Estructura del programa

La interfaz gráfica del usuario consiste en una clase principal la cual controla los diferentes módulos que conforman el proyecto, estos módulos consisten en clases o funciones que realizan tareas simples.

En la Figura 1 se muestra la interfaz gráfica del sistema que consta de:

1. Un mapa con la posición del observador y los satélites que se desean observar.
2. Una tabla con la fecha y hora de la próxima línea de vista de cada satélite.
3. La hora y fecha del observador en formato UTC.
4. Los datos del satélite seleccionado.

El mapa cuenta con diferentes tipos de marcadores: el marcador "X" representa la ubicación del observador que puede ser una Estación Terrena, los rombos azules son los satélites que se desean observar y el rombo rojo es el satélite seleccionado.

Una vez seleccionado el satélite a observar se pueden generar las siguientes gráficas y tablas:

- Gráfica de ángulos: muestra las gráficas de azimut y elevación del satélite seleccionado.
- Ruta del satélite: muestra la ruta del satélite seleccionado.
- Tabla de ángulos: muestra una tabla con los ángulos de elevación y azimut del satélite seleccionado.

Para simular el movimiento de los satélites en el mapa se utilizó la función `basemaps` de la librería `Matplotlib`, la cual genera cada 3 segundos un nuevo mapa con las posiciones actuales de los satélites.

### 2.2. Módulo para la actualización de la base de datos de satélites

Los parámetros TEL para el cálculo de los elementos orbitales se actualizan periódicamente dependiendo de la época del satélite, para ello se desarrolló una función la cual consulta la página web <http://celestrak.com/NORAD/elements/>, tal como se observa en la Figura 2

La actualización se realiza utilizando una técnica llamada `scraping`, esta técnica consiste en simular el comportamiento de una persona al navegar en una página web, en este caso, a una persona buscando los formatos TEL, esto generará un archivo para cada satélite encontrado en la página web y estos archivos se registrarán en un diccionario donde se indique el nombre del satélite y la ubicación del archivo generado.

Tabla 1. Formato TEL.

AAAAAAAAAAAAAAAAAAAAAAAAAAAA
1 NNNNNU NNNNNAAA NNNNN.NNNNNNNNN +.NNNNNNNN +NNNNN-N +NNNNN-N N NNNNN
2 NNNNN NNN.NNNN NNN.NNNN NNNNNNNN NNN.NNNN NNN.NNNN NN.NNNNNNNNNNNNNNN

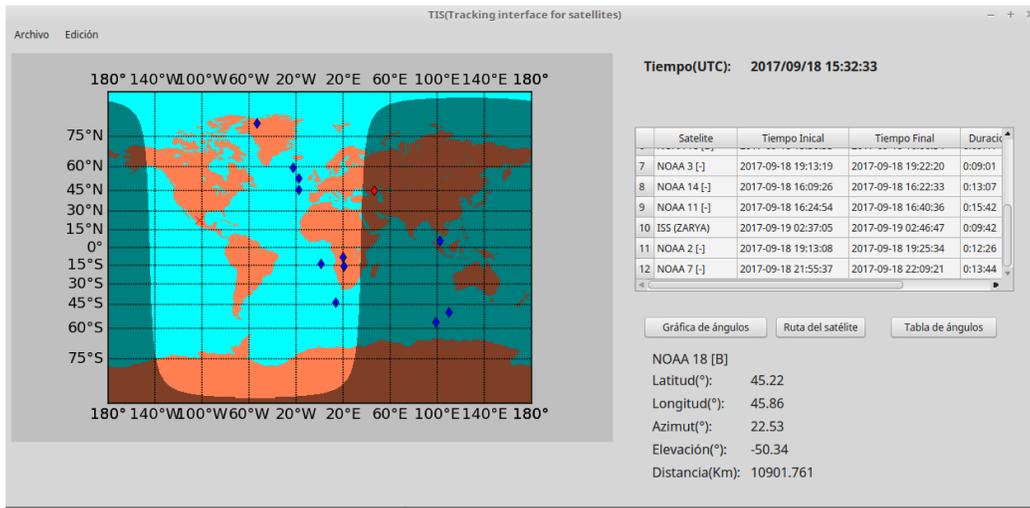


Figura 1. Interfaz gráfica del usuario

Tabla 2. Descripción de la línea 1 de TEL

Columna	Descripción
01	Número de línea
03-07	Número de satélite
08	Clasificación
10-11	Dos últimas cifras del año de lanzamiento
12-14	Número de lanzamiento del año
15-16	Identificación del componente del lanzamiento
19-20	Época - Dos últimas cifras del año
21-32	Época - Día del año y fracción del día
34-43	Primera derivada de la velocidad orbital media
45-52	Segunda derivada de la velocidad orbital media
54-61	Término de resistencia
63	Número 0
65-68	Número de elemento
69	Suma de verificación

Tabla 3. Descripción de la línea 2 de TEL

Columna	Descripción
01	Número de línea
03-07	Número de satélite
09-16	Inclinación (grados)
18-25	Ascensión recta del nodo ascendente (grados) $\Omega$
27-33	Excentricidad
35-42	Argumento del perigeo (grados)
44-51	Anomalía media (grado)
53-63	Velocidad orbital media ( $\frac{revoluciones}{día}$ )
64-68	Número de revoluciones en la época
69	Suma de verificación

### 2.3. Cálculo de los ángulos de azimut y elevación

las gráficas de azimut y elevación se generan a partir de dos funciones: la primera función obtiene los tiempos de línea de vista calculados por el módulo de predicción de pyephem, adicionalmente se calculan los ángulos de azimut, elevación, latitud y longitud; la segunda función toma los ángulos de azimut y elevación junto con el tiempo de línea de vista para construir las gráficas de la Figura 3.

El módulo que calcula los datos de rastreo de los satélites también guarda los datos para generar las gráficas cuando sean llamados por el usuario, en caso de que las gráficas no sean válidas, el programa volverá a calcular los ángulos del satélite.

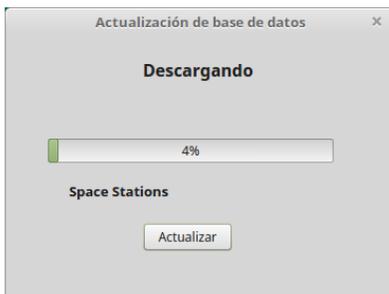
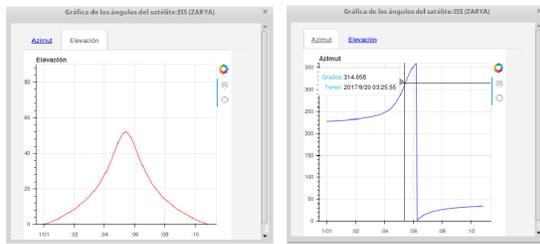


Figura 2. Módulo para la actualización de la base de datos de los satélites.



(a) Ángulo de elevación (b) Ángulo de azimut

Figura 3. Gráfica de los ángulos de elevación y azimut.

**2.4. Cálculo de la ruta del satélite**

Este módulo se basa en la librería folium la cual permite trazar la ruta que seguirá el satélite de acuerdo a los ángulos de latitud y longitud previamente calculados, esta librería permite mostrar los puntos de manera precisa junto con el tiempo de paso del satélite con respecto a la ubicación del observador, tal como se muestra en la Figura 4.

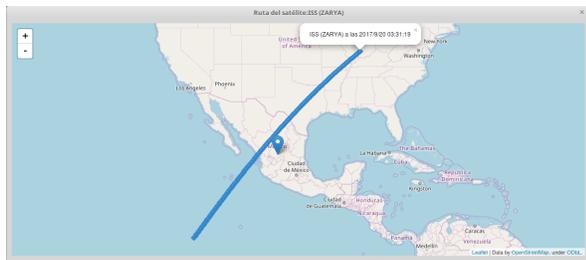


Figura 4. Ruta del satélite.

Para marcar la ruta del satélite que se muestra en el mapa se generaron todos los puntos con un intervalo de tiempo determinado, de este modo se simula una línea que es la ruta que seguirá el satélite durante el tiempo de vista del observador.

**2.5. Módulo de control de los rotores de la antena**

Además se integró un módulo para el control de rotores en caso de conectarse a una Estación Terrena vía IP o USB, esto permitirá al usuario realizar el rastreo satelital a través de una sola interfaz gráfica, tal como se muestra en la Figura 5.

Este módulo de control se basa en dos librerías: una que permite la comunicación con los rotores mediante el puerto USB y otra a través del protocolo TCP/IP.

El módulo USB utiliza la librería hamlib, esta muestra la posición en grados de cada rotor, fija a un ángulo en específico cada rotor y programa la ruta a seguir del satélite seleccionado en tiempo real, además inicializa los rotores al ángulo cero al terminar el recorrido programado.



Figura 5. Interfaz para el control de los rotores de una Estación Terrena.

El módulo TCP/IP se basa en la librería socket la cual permite la comunicación con la librería hamlib vía Ethernet a través de un puerto en específico.

**3. Resultados**

Para validar los cálculos realizados por el proyecto TIS, estos se compararon con los del proyecto Gpredict ya que es la plataforma que comúnmente se utiliza para rastreo satelital, para ello se generaron las gráficas de los ángulos de azimut y elevación para la estación espacial internacional (ISS) además de la ubicación geográfica en ambos proyectos.

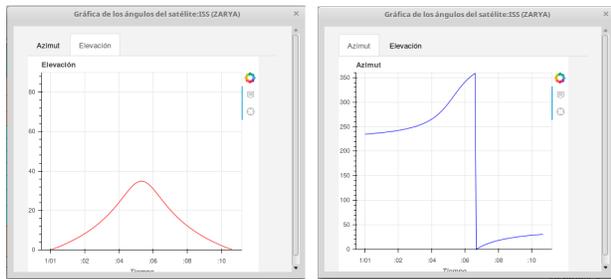
**3.1. Cálculo de ángulo de azimut y elevación**

La ISS se encuentra a una altura de 408 km sobre el nivel del mar, esta se desplaza a una velocidad de 7.66 km/s, este satélite se elevó sobre el horizonte del observador el día 01/02/2018 a las 21:53:44 horas para terminar a las 22:04:24 horas, se calculó y graficó los ángulos de azimut y elevación descritos por el satélite en el proyecto TIS y se compararon los resultados con la gráfica del proyecto Gpredict.

En la Figura 6 se puede observar una comparación de las gráficas calculadas por TIS y Gpredict, las subfiguras 6(a) y 6(b) corresponden al ángulo de elevación y azimut calculado por TIS respectivamente y la subfigura 6(c) corresponde a los cálculos obtenidos por Gpredict, se observa que los resultados son similares entre ambos proyectos puesto que los valores son cercanos tanto en tiempo como en ángulo y solo se diferencian en los valores del eje x ya que el proyecto TIS muestra el tiempo de duración y el proyecto Gpredict muestra el tiempo cuando se encuentra en esa posición.

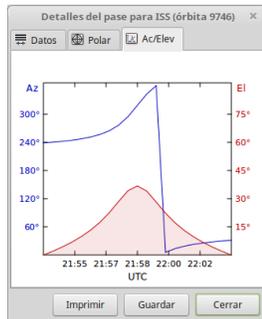
**3.2. Cálculo de la posición de los satélites**

Adicionalmente se comparó la posición de los satélites calculados por TIS y la posición de los satélites mostrados en el proyecto Gpredict.



(a)

(b)



(c)

Figura 6. Comparación de las gráficas de azimut y elevación del satélite ISS.

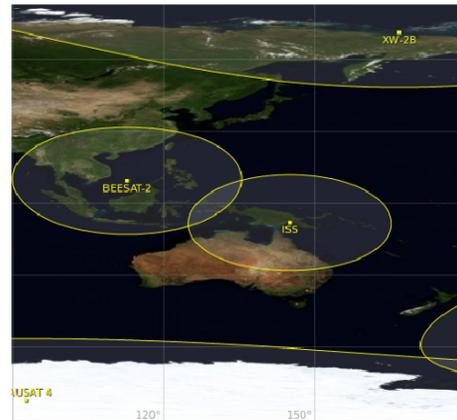
En la Figura 7 se muestra la posición del satélite ISS tanto en el proyecto Gpredict como en el proyecto TIS, se puede observar como la posición del satélite ISS es relativamente parecida ya que las escalas del proyecto Gpredict y el proyecto TIS son diferentes.

#### 4. Conclusión

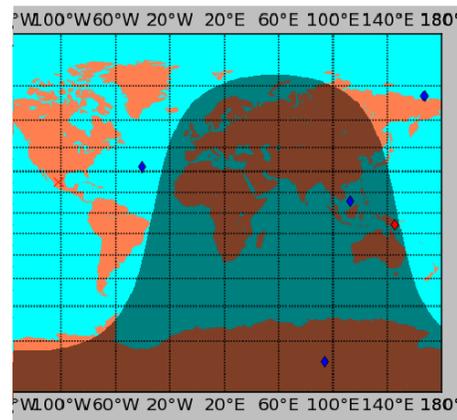
La interfaz desarrollada es capaz de dar seguimiento a varios satélites a la vez, muestra ciertos datos específicos del satélite seleccionado y actualiza la posición de los satélites en el mapa, es capaz de calcular la ruta que tomará cuando pase cerca del observador y puede estimar los parámetros de ajuste para una Estación Terrena.

#### Referencias

- [1] Perez, F., Granger, B. E., & Hunter, J. D. (2011). Python: an ecosystem for scientific computing. *Computing in Science & Engineering*, 13(2), 13-21.
- [2] Rhodes, B. C. (2011). PyEphem: astronomical ephemeris for python. *Astrophysics Source Code Library*.
- [3] Kelso, T. S., & Alfano, S. (2006, May). Satellite orbital conjunction reports assessing threatening encounters in space (SO-CRATES). In *Modeling, Simulation, and Verification of Space-based Systems III* (Vol. 6221, p. 622101). International Society for Optics and Photonics.
- [4] Osweiler, V. P. (2006). Covariance estimation and autocorrelation of NORAD two-line element sets (No. AFIT/GSS/ENY/06-



(a) Posición en Gpredict



(b) Posición en TIS

Figura 7. Comparación de la posición del satélite ISS.

- M09). AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING AND MANAGEMENT.
- [5] Kelso, T. S. (2011). NORAD two-line element sets current data. 2013, accessed.
- [6] Vallado, D. A., Virgili, B. B., & Flohrer, T. (2013, April). Improved SSA through orbit determination of two-line element sets. In *ESA Space Debris Conference*.
- [7] Kelso, T. S., Hoots, F., & Roehrich, R. (1988). Spacetrack report no. 3-models for propagation of norad element sets.