ENG 4350 P6: Simulated Software Tracking

Abdul Cisman and Michael Stewart

October 31 2020

Purpose

The Purpose of this laboratory was to work on testing the code for the ARO virtual visit in mid-December. The code should output an Ephemeris (.e) file that once imported into STK properly tracks the movements of the designated Satellite . Additionally, the code will need to output a Sensor Pointing (.sp) file that will contain the pointing angle predicted by the code to angle the ARO dish at a given satellite. Screenshots and results of test runs were provided in the lab report.

Procedure

Work Distribution

Michael	Abdul
Coding & Debugging	STK
%50 of Documentation	50% Documentation

The goal of this lab was to continue testing the code and ensure that the Ephemeris and Sensor Pointing file is outputted.
The Sensor Pointing file is to contain the:

- Version of STK used
- Number of Attitude Points
- Azimuth and Elevation angles
- Time since Satellite Epoch in seconds

The results gathered from the **range_topo2lookangle()**, that is the Azimuth and Elevation angles are required to properly produce the Sensor Pointing file.

The **STKout()** and **STKsp()** functions from Laboratory 2 were salvaged and implemented into this iteration of code with minor changes. As explained further on, their outputs were first displayed in a .txt Text File format to enable readability and observation of any obvious issues. The functions would eventually output their corresponding data in their extension formats to be read in by STK .

Sensor pointing files contain attitude data which describes a rotation from the parent body coordinate frame to the sensor body frame. This is useful because you can import external sensor attitude information into STK so you can model certain circumstances. In a pointing file, you will find the following, the stk version, begin and end attitude, number of attitude points, time in seconds from epoch. A sequence is defaulted and will be supplied. If you enter 323, the sensor will rotate about this boresight. There are also only 2 valid sequence values for

Azimuth/Elevation angles and they are sequence 323 and 213. So in STK, a 323 sequence is going to rotate about the boresight for a fixed sensor pointing. For example, for a pointing targeted type sensor, the boresight type will be tracked and rotated about the boresight. A 213 sequence is equivalent to using a fixed boresight type and instead of rotating, it will hold about the boresight. All these can be found under the basic properties of a sensor in STK.

The Procedure can be roughly described in three steps: Implementation of Functions, Debugging, and Comparison of Values.

Implementation of Code Functions

The Following Functions were added or editing in this iteration of code:

- **STKout()**
    - Creates a Ephemeris File for a designated Satellite
        - The Original Output of the **STKout()** function was a .txt file that could be easily read and fixed. The file extension was then changed to .e to ensure that STK could read the file.
        - The format and output of this function is vital for STK comparisons
            - Multiple minor bugs were fixed from the last iterations of code, including using time in seconds from Epoch, and formatted print statement
- **STKsp()**
    - Creates a Pointing File for a designated Satellite.
        - Similar to that of the Ephemeris File, the Pointing File was first outputted as a .txt file to identify any noticeable mistakes. It was then changed to output a .sp extension to import into STK.
            - Multiple bugs were also fixed in the implementation of this code including: numpy.rad2deg(), and ensuring only the specific satellite chosen was outputted
- **Output_function()**
    - Output_funciton is a new function which formats the values produced by **sat_ECI()**, **sat_ECF(), Range_topo2look_angles()** for a specifically defined satellite. This function then calls upon **STKout()** and **STKsp()** to output the Pointing and Ephemeris files for that Satellite.

Debugging

All Major Changes are Recorded on Github commits at the following site:
https://github.com/mstewart2525/ENG4350Lab03Draft2/commits/Lab04/MainCall.py

The following functions were edited to output more accurate values:

- **Poitining()**
    - Fixed Repeating AOS values
        - Created a Unique_AOS_LOS variable which ensures only one AOS to LOS pair is outputted
            - To explain this further. The original output in the AOS_LOS.csv file looked something like this:
            -

Sat	AOS Time	LOS Time	Power Loss
BIIR-2	16:00:00	16:02:00	-1
BIIR-2	16:00:00	16:04:00	-1
BIIR-2	16:00:00	16:09:00	-1

            - In the new version of the code, it ensures that only the pair AOS 16:00:00- LOS 16:02:00 shows. And that another AOS is established for the next LOS, say AOS 16:02:00- LOS 16:04:00.
            -

Sat	AOs Time	LOS Time	Power Loss
BIIR-2	16:00	16:02	-1
BIIR-2	16:02	16:04	-1
BIIR-2	16:05	16:06	-1

            -
    - Fixed Availability logical Statement
        - Wrapping `(AZ_list[j-Satnum_iteration] < float(AZ_muth_min) or AZ_list[j-Satnum_iteration] > float(AZ_muth_max) or float(EL_lim_max) < EL_list[j-Satnum_iteration] or EL_list[j-Satnum_iteration] < float(EL_lim_min))` in brackets ensured that the right availability would be found
- **sat_ECF()** and **sat_ECI()** functions
    - These functions relied on a matrix multiplication which would output the wrong answer. The solution was to get rid of scipy.spatial.transform Rotation and introduce numpy.matmul. This created a more accurate answer for ECI and ECF over the scipy.

- General
    - Multiple “quality of life” code pieces were added.
        - np.deg2rad() replacing math.pi/180
        - SatListPropagate()
            - Output Satellite Name next to Satellite index to easily pick desired Satellite.

Comparison of Values

For this lab we chose to use the **BIIR-2 (PRN 13)** satellite for testing, this one was most convenient since it was the first satellite read in from the TLE file and thus it's values were always first in our lists (which made it easy to check to see if our values were close using the variable explorer) outputted in our code. The Test Case included the Time 27 October 2020 16:00:00 to 20:00:00 with a time step of 60 seconds. This time was just enough to see deviations with somewhat high accuracy, while keeping computation time down.
The Tables created below, show snippets of information comparing our Test Satellite calculated from the code and calculated from STK.
The tables below show the comparison between the perifocal position/velocity from STK and the code (master file). Additionally, we also talk about the discrepancies between both data results. See below for a manual comparison and automatic comparison.

In this lab we compared the values of our chosen Test Satellite **BIIR-2 (PRN 13)**. We approached this in two methods. First method was to output the values of each important step into a .csv file which could be easily filtered to show the data of a Satellite at a time step, this .csv file came to be known as **Master.csv,** and comparisons referred to as "Manual comparison". The Comparison of STK and Master.csv values started at the Anomalies of the Orbit and continued until the Topocentric system. Further investigation may be done by the reader by using the Master.csv file and filtering data for a specific Satellite. The second method involved outputting Ephemeris and Pointing files from the code to import into STK, and has come to be known colloquially as "Automatic" comparisons .

This Master .csv file contains the following information for a given satellite:

- Perifocal Position and Velocity
- ECI position and velocity
- ECF Position and Velocity
- Azimuth and Elevation from Ground Station and their Rates
- Topocentric Position and Velocity
- Satellite Name
- Availability
- AOS/LOS booleans

STK Generated data from the Satellite TLE files was created in STK and Reports for the Reference Systems, and Orbital Anomalies (Mean, True, and Eccentric) were generated. These reports are found in an attached folder for **BIIR-2(PRN 13)**. The file is saved as GPS_BIIR-02_24876_STK_anomalies.txt

The Mean, Eccentric and True Anomalies were compared manually by creating global variables inside the Main.py code. All global variables that were used for comparison started with the name "zTest_". The constructed code calculates all Anomaly values for all satellites at a given time step then continues to the next satellites for that time step. Identifying the length of our Satellite List "SatList" and looking at the values of the ZTest values every length of SatList gives us the points for a particular satellite. The resultant table was created manually through this method.

Results and Discussions

Results for Manual Comparison

Anomalies

Time	Mean Anomaly	Eccentric Anomaly	True Anomaly
27 Oct 16:00:00	16.555329	16.481426	13.0718859
27 Oct 16:01:00	17.056734	16.980668	13.473613899335106
27 Oct 16:02:00	17.558138	17.479991	13.875866831481053

Table 1: Values Gathered Manually for Satellite GPS-BIIR-02 (PRN 13)

Time	Mean Anomaly	Eccentric Anomaly	True Anomaly
27 Oct 16:00:00	15.074	15.144	15.214
27 Oct 16:01:00	15.581	15.653	15.726
27 oCT 16:02:00	16.093	16.168	16.242

Table 2: Values of the Anomalies generated from STK report for Satellite GPS-BIIR-02 (PRN 13)

Perifocal

Time (UTCG)	Perifocal x(km)	Perifocal y(km)	Perifocal z(km)
27 Oct 2020 16:00:00	-25516.650438	-6924.36500	~0
27 Oct 2020 16:01:00	-25455.348837	-7147.492493	~0
27 Oct 2020 16:02:00	-25391.549363	-7372.003345	~0

Table 3: Values of the of the Perifocal Frame of Reference for Satellite GPS-BIIR-02 (PRN 13) obtained from the STK

Time	Perifocal x(km)	Perifocal y(km)	Perifocal z(km)
27 Oct 2020 16:00:00	25758.64785	5980.90411922	0

27 Oct 2020 16:01:00	25716.26610	6161.40506886	0
27 Oct 2020 16:02:00	25672.56688	6341.84356670	0

Table 4: Values of the Perifocal Frame of Reference for Satellite GPS-BIIR-02 (PRN 13) obtained from the code.

ECI

Time	**ECI x**	**ECI y**	**ECI z**
27 Oct 2020 16:00:00	-8548.6900	-14094.6325	20677.0497
27 Oct 2020 16:01:00	-8373.10602242031	-14130.3394	20724.7161
27 Oct 2020 16:02:00	-8196.8782	-14165.4007	20771.4294

Table 5: Values of the ECI for Satellite GPS-BIIR-02 (PRN 13) obtained from the code.

Time	**ECI X**	**ECI y**	**ECF z**
27 Oct 2020 16:00:00	-6996.873078	-14341.842978	21080.83512.
27 Oct 2020 16:01:00	-6773.5000041	-14379.732595	21128.285396
27 Oct 2020 16:02:00	-6547.665886	-14416.839531	21174.514049

Table 6: Values of the ECI for Satellite GPS-BIIR-02 (PRN 13) obtained from STK.

ECF

Time	**ECF x**	**ECF y**	**ECF z**
27 Oct 2020 16:00:00	13041.0207	-10083.1813	20677.0497
27 Oct 2020 16:01:00	13052.7987	-9969.9469	20724.7161
27 Oct 2020 16:02:00	13065.2535	-9856.2944	20771.4294

Table 7: Values of the ECF for Satellite GPS-BIIR-02 (PRN 13) obtained from the code.

Time	**ECF x**	**ECF y**	**ECF z**
27 Oct 2020 16:00:00	13496.895502	-8548.005334	21066.839318
27 Oct 2020 16:01:00	13631.901420	-7742.926790	21291.9243254
27 Oct 2020 16:02:00	13782.322642	-6932.775479	21475.697371

Table 8: Values of the ECF for Satellite GPS-BIIR-02 (PRN 13) obtained from STK.

Topocentric

Time	Topocentric x	Topocentric y	Topocentric z
27 Oct 2020 16:00:00	13400.5220	-6263.6662	7272.1340
27 Oct 2020 16:01:00	13432.6374	-6211.9782	7386.8742
27 Oct 2020 16:02:00	13465.3457	-6158.8403	7501.2571

Table 9: Values of the Topocentric for Satellite GPS-BIIR-02 (PRN 13) obtained from the code.

Automatic Comparison-STK .e and .sp files

Ephemeris

Two Different Types of Comparison were created using “Automatic” methods to compare orbits. The First was to import the ECI Reference System into STK using an ephemeris file. The results are as follows:

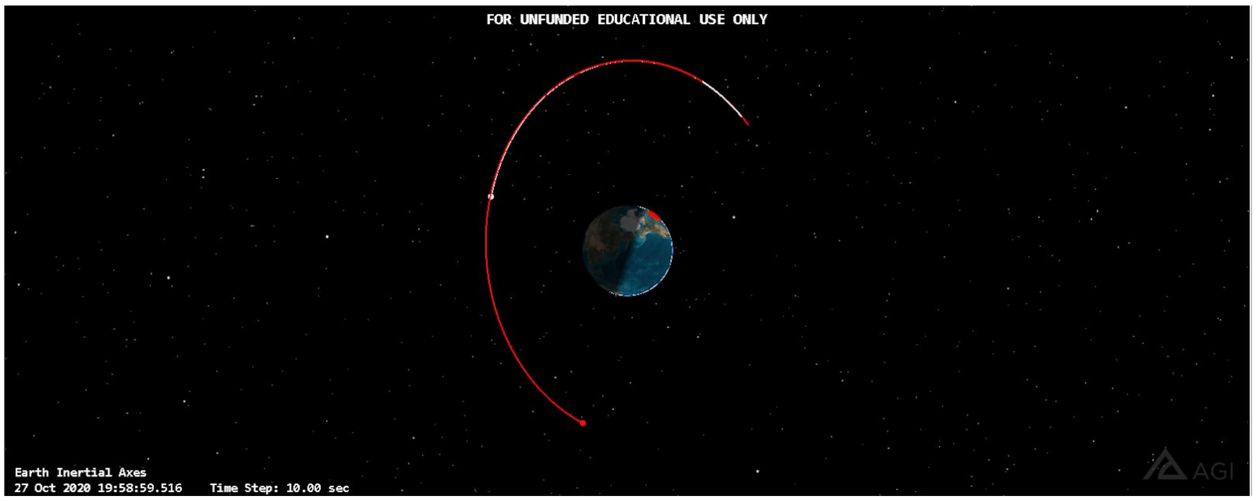
FOR UNFUNDED EDUCATIONAL USE ONLY
Earth Inertial Axes
27 Oct 2020 19:58:59.516 Time Step: 10.00 sec
AGI

Figure 1: Orbit comparison for satellite BIIR-2 (Inertial)

In figure 1, the red orbit is the Ephemeris Inertial imported orbit while white is the original Orbit for **BIIR-2 (PRN 13)**. Orbit was recorded from 27 October 16:00:00 to 20:00:00 for both orbits.

The Second way of Comparing the Code to STK can be from comparing the outputted Fixed Ephemeris File from the code to the original orbit.

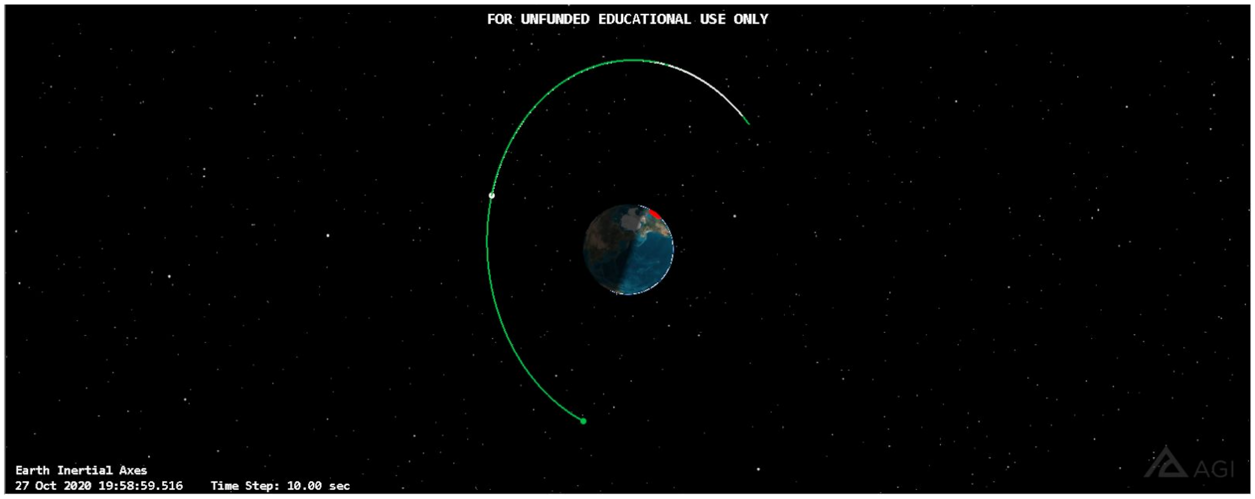
FOR UNFUNDED EDUCATIONAL USE ONLY
Earth Inertial Axes
27 Oct 2020 19:58:59.516 Time Step: 10.00 sec
AGI

Figure 2: Orbit comparison for satellite BIIR-2 (Fixed)

In figure 2, the green orbit is the Ephemeris Fixed Imported Orbit while the white is the original Orbit of **BIIR-2(PRN13)**. Orbit was recorded from 27 October 16:00:00 to 20:00:00 for both orbits.

Pointing

A function called STKsp was implemented into the code. This function generated a pointing file which was then imported into STK. Figure 3 below shows the output and the cone angle.

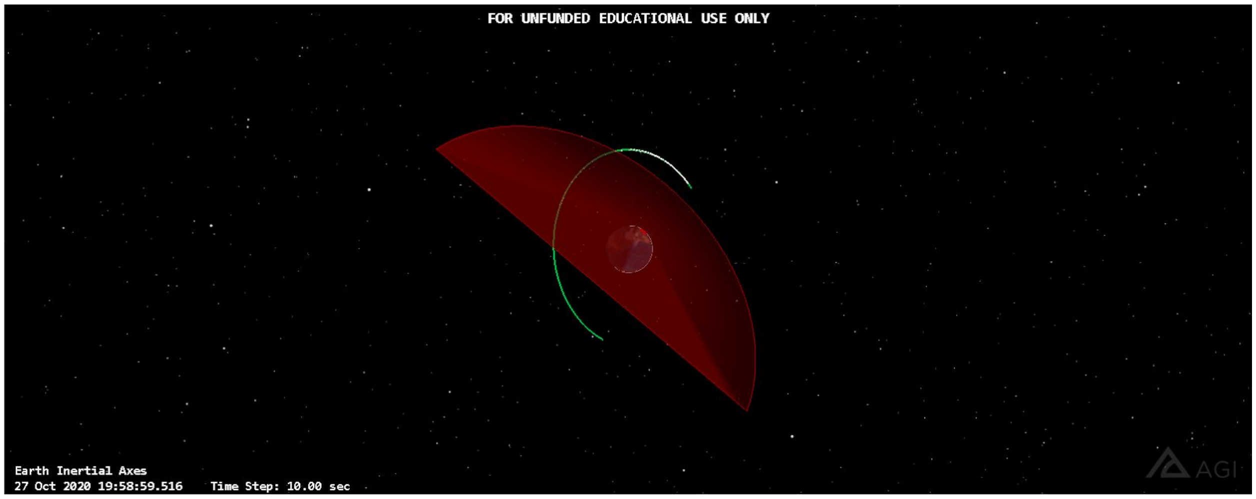
FOR UNFUNDED EDUCATIONAL USE ONLY
Earth Inertial Axes
27 Oct 2020 19:58:59.516 Time Step: 10.00 sec
AGI

Figure 3: Pointing angle for the sensor pointing file

Discussion for Manual Comparison

From the manual comparison of values using the Master.csv file and STK Reports show the following discrepancies:

- Mean Anomaly if off by ~1^0
- Eccentric Anomaly is off by by ~1.5^0
- True Anomaly is off by ~2^0
- Semi-major Axis is off by around ~1km
- Perifocal X is off by ~1000km

    - Note: the signs are switched for Perifocal Frame of Reference but this does not affect further calculations
        - It is thought that the Perifocal Frame of Reference is initialized incorrectly in STK
    - This ~1000km offset is passed on throughout the next calculations in ECI, ECF, Topocentric and eventually Azimuth and Elevation Angles

Discussion for Simulated Software Tracking

Ephemeris

Comparing the Red and Green Orbit of the “Automatic Comparison” Section above to the original orbit of **BIIR-2 (PRN 13)** we note several differences:

- The Fixed and Inertial Imported from Ephemeris data orbit appear to be further along their orbit than the original orbit
    - This may have resulted from the Anomaly deviation compounding over the 4 hours of orbit used.
- The Imported Inertial and Fixed Orbits are very similar to each other. This proves that the calculation between the inertial and Fixed frames (for the positions anyways) is correct.

Pointing

- On our own, we imported the pointing file into STK. To do so, we selected the sensor properties, changed the pointing type to external and selected the sensor pointing file the code outputted. In this file, it contains the pointing angles predicted by the code to a point at a given spacecraft.
    - However, there was an error and the angle was incorrect. This could be due to the reasons listed in the discussion for the manual comparison section of the lab.

Conclusions

In this lab,we ran into a multitude of issues surrounding the Pointing and Ephemeris file outputs but persevered and were able to output a somewhat reasonable answer, of course more refinement will be needed! The STK portion from the previous lab helped us with data collection.Similar to that of the previous lab, the Master.csv file served us well to identify and isolate issues to a specific calculation, it gave us a sense of accomplishment seeing our values get closer and closer to the Reports generated . The coding of the Pointing and Ephemeris file in the .txt file format provided a clear way to identify obvious mistakes in the coding of them , such as not representing the chosen satellite . There was an abundant amount of close or correct data being output this session of the lab compared to that of P5.The most notable values of ECI position, ECF position and Perifocal position are now relatively close compared to what we submitted for lab P5, where the values had huge discrepancies. We wanted to compare the values of the topocentric range to the one in STK but couldn’t really find a report manager for the topocentric range. It would be beneficial if we could have been provided a set of bench-mark values to compare each operation to, this would reduce the time spent debugging.

Lessons Learned

This session reinforced the concept of team-work and preparation. As we were not prepared to go over the code with Professor Chesser, we could not get good advice on how to fix our issues.