EENG 533: Navigation Using GPS
Project 5: Introduction to GPS Processing

Objectives

1. Gain familiarity with GPS data formats
2. Gain familiarity with how to download GPS data from the web
3. Learn how to examine and plot relevant GPS data

Overview

In this lab, you will download real GPS data and start taking a look at what is available with a standard, commercial-grade dual frequency receiver.

Collaboration

This is an individual lab. You are allowed to discuss any aspect of the lab with other students, and you may look at each other's source code for debugging purposes. However, your programming must be your own; you may not copy or transcribe someone else's program, in part or in whole.

Provided Functions

This project will again make use of the georinex (`https://pypi.org/project/georinex/`) Python library. You should have it installed into your Python environment, but reference the project 2 description for installation instructions if needed.

The `datetime2tow()` function will again be used from the provided `helper.py` file to convert from the calendar date and time given in a RINEX observation file to time of week in seconds. The code template file (`project5_template.py`) shows an example of how to convert all the RINEX observation times in a compact way.

Once you have loaded the observation file and perfomed the time conversion, you will have access to the following values

Name	Description	Units
time_tow	time since start of week	sec
sv.data	GPS pseudo-random number	'Gxx'
C1	L1 C/A pseudorange	m
C2	L2C pseudorange	m
L1	L1 carrier phase	cycles
L2	L2 carrier phase	cycles
P1	L1 P-code pseudorange	m
P2	L2 P-code pseudorange	m
S1	L1 signal strength	dB-Hz
S2	L2 signal strength	dB-Hz

Obtaining Data

For this project, you will be downloading data from the Continuously Operating Reference Station (CORS) network. The data file access page for this system can be found at `https://geodesy.noaa.gov/UFCORS/`

Select a 'Start Date' at least one day in the past so a full day of data is available. You can leave 'Start Time' at 00:00. For 'Duration in Hour(s)', pick 24. Select a 'Site ID'. You can use the 'CORS Map' to help or just pick a site from the list. Once you pick a site, the 'Available Satellite Systems' option will populate. Pick 'Legacy Applications / GPS (L1+L2 only)'. Select a 'Sampling Rate' and pick 15 seconds. You do not need to request any Optional Files. Just click 'get CORS data file'.

You should get a download of zip file. Unzip the file and look for a file ending the extension '.21o'. This is the RINEX observation file.

Task A

For this task, you should download 24 hours of data from somewhere in Florida (your choice of the site), on a day of your choosing.

Assuming you have loaded the observations into `obs`, you can access the measurements with `obs['L1']` for example to get all L1 carrier phase measurements for all satellites and times. `obs['L1'].sel(sv='G16')` would give those measurements for just GPS satellite 16. For some times and satellites a measurement may have a value of `nan` which stands for "not a number". This means a specific measurement type was not available for this satellite at a specific time, most likely due to the satellite being below the visible horizon.

You should then generate the following properly labled plots. The non-existent `nan` measurements will be correctly not plotted when using `time_tow` for your x-axis. The complete list of satellites in the observation file are accessible as strings in `obs.sv.data`.

1. A plot of the visible GPS satellite PRNs vs. time in GPS **week seconds** (Hint 1: plot the PRNs vs. time using points. This will prevent extra connecting lines from being added across time segments of `nan`s. Hint 2: a measurement divided by itself times the PRN plotted vs `time_tow` will produce a horizontal line with the value of the PRN, except for when a `nan` is present)
2. A plot of the visible GPS satellite PRNs vs. time in GPS **hour of day**
3. A plot of all L1 C/A code pseudoranges (C1) vs. time in GPS hour of day
4. A plot of all L1 Carrier-Phase measurements (L1) vs. time in GPS hour of day
5. A plot of the time derivative of L1 C/A code pseudorange (C1) vs. time in GPS hour of day for just one satellite. The time derivative should be expressed in units of meters/second (NOT meters/hour). Choose a satellite that is available for a long time. (Hint: use the numpy `diff()` method to calculate the difference between pseudorange measurements and divide by the time interval)
6. A plot of L1 C/A code pseudorange (C1) vs. time in GPS hour of day for the same satellite as plot 5
7. A plot of L1 Carrier-Phase measurement (L1) vs. time in GPS hour of day for the same satellite as plot 5
8. A plot of L2 Carrier-Phase Measurements (L2) vs. time in GPS hour of day for the same satellite as plot 5
9. A plot of L2 P code pseudorange (P2) vs. time in GPS hour of day for the same satellite as plot 5
10. A plot of all L1 C/A code measurements (C1) (converted to units of cycles, using the L1 wavelength) minus all L1 carrier-phase measurements (L1) vs. time. Note that wavelength can be found by dividing the speed of light by the carrier frequency.

Answer the following questions:

A. What is the relationship between L1 C/A code pseudorange (C1) (figure 6) and L2 P code pseudorange (P2) (see figure 9)? (Note: you may need to zoom in or plot the difference in order to compare the two.)

B. What is the relationship between the carrier-phase measurements and the pseudorange measurements? Identify where there is a cycle slip (discontinuity) in plot 10. There should be some relatively “flat” areas in plot 10. Zoom in on one. How much does the code-minus-carrier that you plotted vary when there is no cycle slip?

C. Based on plot 5, what is the range of velocity of the satellite relative to the stationary receiver that you observe?

D. How many unique satellites were visible throughout the day?

Task B: Compare Satellite Visibilities

Now, download data from three additional sites: another site in Florida, a site in Ohio, and a site that is very far away from Florida (Alaska or farther). You now have two Florida data files, an Ohio data file, and one far-away data file.

Generate a single plot that shows visible PRN vs. time in GPS hours for all four locations. Use a different color for each site's data set. You should plot as points, and you should use small vertical offsets in order to be able to see all four receivers. The point is to easily compare when each satellite was visible for all three receivers on a single plot. In one or two sentences, briefly comment on what you see in this plot.

An example of what this figure should look like is shown here:

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

Deliverables

Submit via Canvas your Python script and a pdf file with your plots and answers to the questions from Task A and B.

Grading

You will be graded for completeness and reasonableness of your answers. Points will be deducted for plots that are not properly labeled.