ENGO 625 – Advanced GNSS Theory and Applications (F2018)

Lab #1 - Single-Point and Differential Positioning

Due Date

Friday October 30, 2020 @ 11:59 PM via D2L dropbox. Please submit a written report and your source code.

Objectives

1. To look at some GPS data (observations and satellite positions)
2. To implement GPS-based positioning solutions.
3. To gain insight into the different processing methods (single point, differential, etc.).
4. To become familiar with analysis based on estimated accuracy, satellite geometry and measurement residuals.
5. To improve programming skills with regards to Geomatics Engineering (C/C++, MATLAB, or Python only, please).

Data Description

Each student will be given three data files.

- A binary file containing 1 Hz GPS observations from a static NovAtel OEMV *remote* receiver.
- A binary file containing 1 Hz GPS observations from a static *base* station. You will be given the true coordinates of both receivers
- A binary file containing the satellite coordinates and velocity components in the Earth-Centred Earth-Fixed (ECEF) frame, also at 1 Hz.

You will be given truth coordinates for both receivers for comparison and executables to view these files. The data has already been corrected for tropospheric error, group delay (not the ionospheric error) and satellite clock offset and drift. You are expected to do the rest of the processing – e.g. additional corrections, satellites dropping in and out of view, synchronizing observations between files. You will use this data for both Labs 1 and 2 in this course.

<u>Observation file</u> – binary format, each element is 64-bit float (double)

Each second of data contains 12 channels. Each channel can hold observations for one satellite. Channels without observations will contain zeroes. The format for one channel is as follows:

<Satellite PRN>
<GPS time of observation (seconds)>
<C/A code pseudorange (m)>
<L1 carrier phase measurement (L1 cycles)>
<Doppler of L1 carrier (Hertz)>
<L2 carrier phase measurement (L2 cycles)>

<u>Satellite file</u> – binary format, each element is 64-bit float (double)

Each second of data contains 12 channels. Each channel can hold coordinates for one satellite. Channels without satellites will contain zeroes. The format for one channel is as follows:

<Satellite PRN>
<GPS time of observation (seconds)>
<X Satellite Coordinate (m)>
<Y Satellite Coordinate (m)>
<Z Satellite Coordinate (m)>
<X Satellite Velocity (m/s)>
<Y Satellite Velocity (m/s)>
<Z Satellite Velocity (m/s)>

To load these files using C/C++ you can read them into structs that contain 6 and 8 doubles respectively (yes everything is stored as a double, including the PRN number). To load them into MATLAB, you can use the `fread` function. Specifically something like :

```
fid = fopen('Satellites.sat')
SatellitePosArray = fread(fid,[8 Inf],'double')'
%reads the entire file into an 8 row, Inf column array then
%transpose the result to get a matrix of doubles where the columns correspond to each
%record
```

Tasks

1. Load the satellite coordinate file and the rover observation file
   - **a.** For the first 300 epoch, plot the coordinates of the satellites in 3D
   - **b.** Discuss: How are the satellites distributed. Is 300 epochs enough time to see the satellite paths? If not, plot some more (possibly all) epochs and discuss.
   - **c.** For the first 300 epochs, plot the pseudorange, Doppler, and L1 carrier phase observed for each of the satellites. You will need to read through the observation array you have loaded from the file row by row, because this receiver has stored the observations for each epoch in the first rows of each set of 12 rows (so the specific row that corresponds to a satellite may change from epoch to epoch).
   - **d.** Discuss: How much does the pseudorange vary from epoch to epoch. Does this match the change in the carrier phase? Does the carrier phase change match what you think it should based on the Doppler and the pseudorange?
2. Use parametric least-squares (PLSQ) and pseudorange measurements to estimate the 3D single point position and clock offset (either X, Y, Z, cdT or geodetic latitude, longitude, height above the ellipsoid and cdT) for each epoch.
   - a. This is non-linear parametric LS, so you will need to iterate the solution. You can start at [0,0,0,0] or at the solution from the last epoch.
   - b. Using the true position provide, determine the position errors in the East-North-Up (ENU) frame. If you have estimated in the ECEF frame you will need to transform your errors to the ENU frame).
   - c. Plot the time series of each of these error components (East, North, Up) **on separate subplots** (use the subplot function in Matlab). On the same graphs, plot the time series of the estimated standard deviations. You must plot both the standard deviation (which is

always positive) and the standard deviation multiplied by negative one (which will then be always negative) in order to get the estimated accuracy "envelope".

   d. Discuss the differences between the true and estimated accuracy. Comment on which assumptions you made may influence any discrepancies between the two. Justify your assumptions if necessary.

3. Compute the HDOP and VDOP.
   a. Plot each DOP as a time series. If you have done the estimation in ECEF, you will need to rotate the XYZ covariance matrix to a local geodetic frame to determine the H and V DOPs.
   b. Plot a time series of the number of satellites used in the solution.
   c. Compare the plots of ~~2a/2b~~ with the accuracy plots from ~~1b/1c~~.
   d. Discuss the role of DOP in analyzing solutions.
4. Compute the residuals
   a. Plot each satellite's residuals as a time series (one plot, with a different colour for each satellite).
   b. Plot all the residuals as a function of satellite elevation angle.
   c. Discuss the trends seen in plots 4a/4b and discuss what you see.

5. Repeat Tasks 1-4 using measurements differenced between the rover and base station. This type of processing is called *between-receiver-single-differencing* mode. You will have to justify any changes in your algorithm.
   a. Comment on the differences between the single-point and receiver-single-differencing modes.
6. From the tasks above, draw an intuitive general conclusion on the quality of the position estimates obtained by a user from GPS based on the number of the satellites in view, their spatial distribution in the sky as well as the quality of the pseudorange.

Tips

- The individual lab tasks are interconnected and you might find that programming separately for each task becomes very tedious. Build your code in modules (e.g. least squares module, design matrix row building module, etc.) so that you can re-use some parts for many different lab tasks or even the next lab. If you are new to programming or are an expert in Matlab, then you may use Matlab. If you are fluent in Python, you may use Python, however I would recommend C++ as it remains an industry favorite.
- Do not design your code to work specifically with your data set (e.g. hard-coding PRNs or GPS times). You may require a different data set for Lab 2, and Lab 2 will use results from Lab 1. For those who are inexperienced with Matlab, there are many tricks that can make your plots look clearer. You can adjust the width and style of the lines, set the color, increase the font size of the labels, and so on. The "set" function in Matlab can be used to set the properties for individual plot lines or entire plot areas. The Property Inspector can be accessed as follows: right click on an object in a figure, click "Show Property Editor" and click "More Properties".
- When plotting time series, showing all six digits of the GPS time can be confusing. What you can do instead is subtract the first time value from all the times, so the time scale starts at zero. Make sure when you do this that the time value that you subtract is the same for all the time series you plot! The time axes should then be labeled sensibly, for example "Time from beginning on test, t (s)"

Report Format

Each report must be professionally written. Be concise but thorough. Explanations should accompany all plots. Assumptions must be clearly documented and justified. All source code must be submitted to the dropbox, not NOT include it in the report. Only C/C++, Matlab or Python code will be accepted.

Grading Criteria (10% of Final Grade)

Technical Accuracy (7/10)

- Correct solution/plots
- Correct analysis of results
- Correct justification and assumptions
- Insight into results
- Graphs correctly labeled (each axis must have a label that includes a description, a symbol and a unit

Source Code (2/10)

- Methodical and logical procession of algorithms
- Commented source code
- If you share software with other classmates, you must acknowledge their contribution both in the comments and in the text of the report.

Report Quality, Neatness and Readability (1/10)

- Title Page
- Neat, clear and effective use of scales on plots
- Logical organization of lab

References

ENGO 625 Course Notes - Advanced GNSS Theory and Applications
ENGO 421 Course Notes (might be useful for coordinate systems)
ENGO 363 Course Notes or Text (for a detailed explanation of Least Squares)