Title: GNSS Lab Report

Lab 1: GNSS Positioning

Course:	AAE4203 — Guidance and Navigation
Group:	Group 3
Members:	ZHANG Zhuoya
Member IDs:	23098581d
Instructor:	Prof. Weisong WEN
Department:	Department of Aeronautical and Aviation Engineering
School:	The Hong Kong Polytechnic University
Date:	November 10, 2025

Abstract

This report demonstrates how we collected GNSS data during lab and the analysis of the collected data and the positioning result. It includes data collection progress description, raw data analysis and visualization, SPP_LSE code demonstration, positioning result analysis and visualization, and corresponding citations.

Keywords: GNSS, SPP, least squares

1 Data Collection

Preparation of the data collection process conducted near Block X:

- Hardware setup:

1.U-blox ZED-F9P + Antenna. 2.One laptop

- Soft ware setup:

1.u_center (data collection software for ublox-receivers)

- Environmental conditions:

1.Surrounding: partial obstruction (bulidings). 2.Weather: sunny day. 3.Time: 27 October, 10 a.m.

Photo of the site:

Figure 1: Data collection environment near Block X.

log summary:

The data collection process is successful, we designed a path with altitude change and no overlap. The total time period is about 100 seconds, sufficient for data analysis.

2 Raw Data Analysis

Based on the result of the data process, we have drawn numerous graphs for data visualization, and the quality of the data can be roughly evaluated based on the following three quality indicators:

Table 1: Data collection summary.

Field	Value
Location	Vicinity of Block X
Duration	30 min @ 1 Hz
Receiver	Multi-constellation GNSS
Conditions	Urban campus, partial obstruction

1.Carrier to Noise Ratio (C/N_0)

Observations with $C/N_0 < 30$ dB-Hz usually have high noise levels [1].

At most of the epoch, the C/N_0 of most signals are larger than 30 dB-Hz, meaning most of the signals are strong enough for precise positioning.

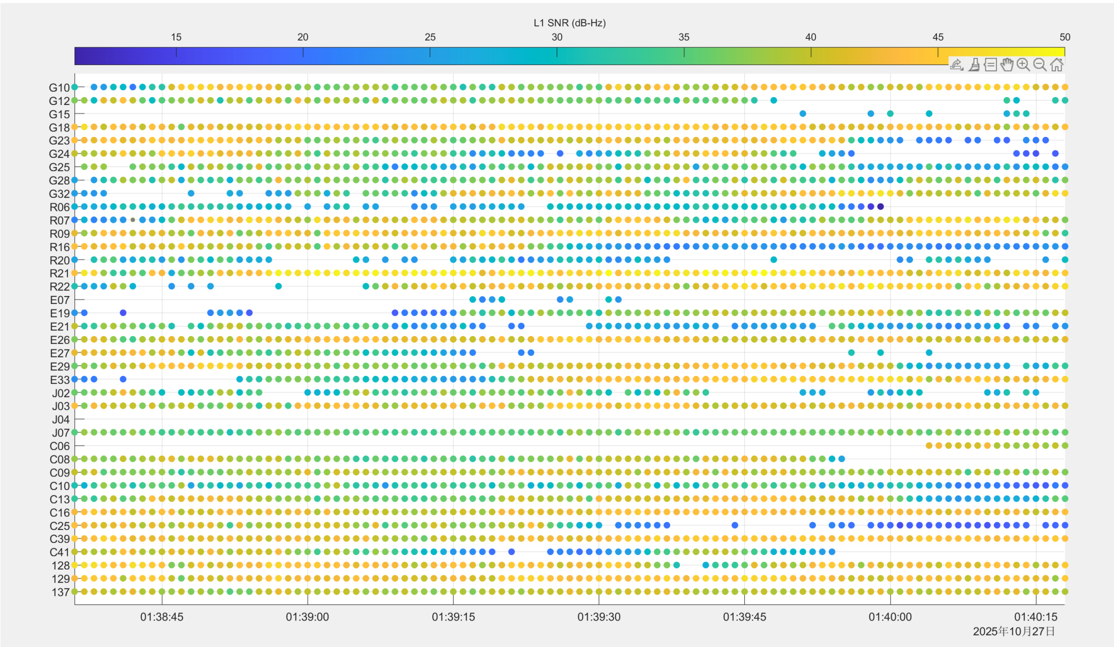

Figure 2: C/N_0 time series.

2.Dilution of Precision (DOP)

Epoches with Position Dilution of Precision (PDOP) lower than 4 is more suitable for high_precision positioning [2].

The PDOP are all under 1.35 which refers to a good satellite geometry.

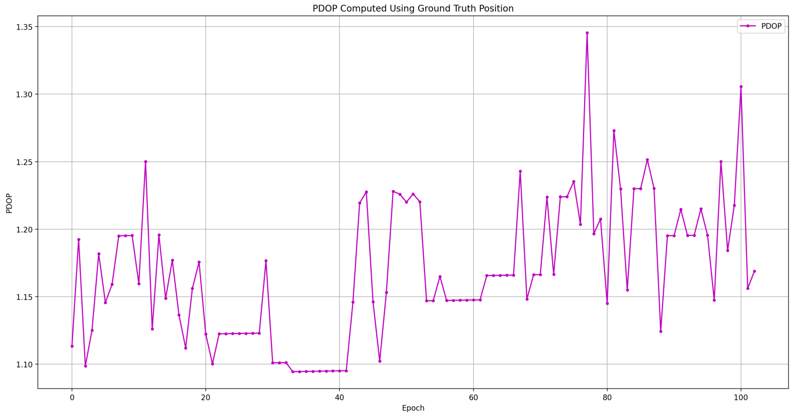

Figure 3: PDOP computed using ground truth position.

3.Number of Satellites Tracked

In a multi-system combination, the higher number of satellites means the higher speed of convergence and higher reliability of positioning [3].

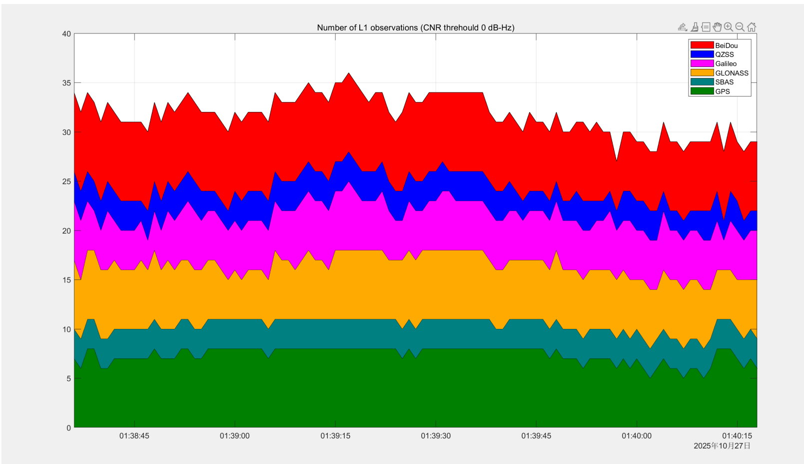

Figure 4: Number of L1 observation of different satellite systems.

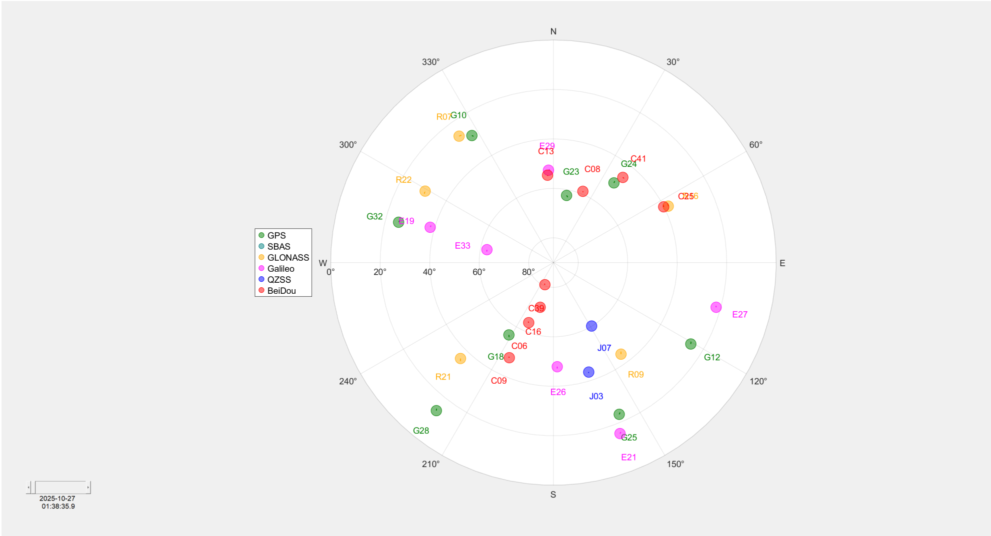

Figure 5: Sky plot of tracked satellites of first epoch.

The number of satellites of our raw data is 24-28, sufficient for positioning, meaning a good sky vision.

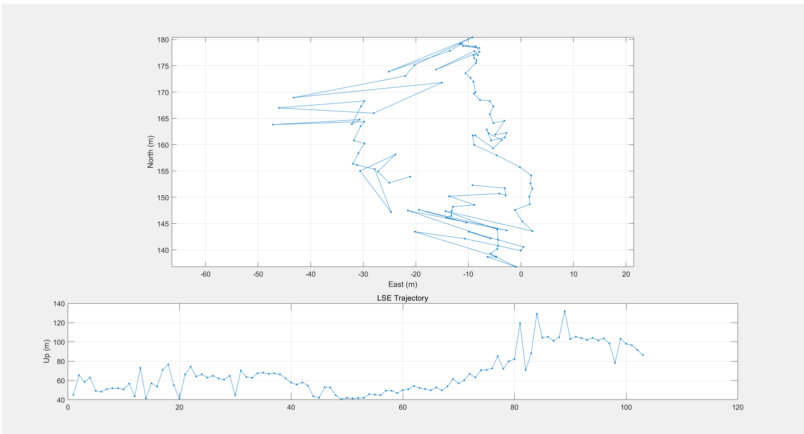

Figure 6: Planar trajectory and height variation LSE by Rinex2CSV.m

In general, the raw data is of good quality.

3 SPP Algorithm (Least Squares)

The Single Point Positioning (SPP) algorithm estimates the user's position and clock bias by solving a system of pseudorange equations.

The fundamental observation model for the i-th satellite is given by:

$$\rho_i = \|\mathbf{x} - \mathbf{s}_i\| + c \cdot \Delta t + \varepsilon_i, \tag{1}$$

where

- ρ_i is the measured pseudorange to satellite i,
- $\mathbf{x} = [x, y, z]^T$ is the unknown user position in ECEF coordinates,
- $\mathbf{s}_i = [s_{ix}, s_{iy}, s_{iz}]^T$ is the known ECEF position of satellite i,
- c is the speed of light,
- Δt is the receiver clock bias (in seconds),
- ε_i represents measurement noise and unmodeled errors.

To solve this nonlinear system, we linearize around an initial guess $\mathbf{x}_0, \Delta t_0$. Define the residual vector $\mathbf{v}$ and the Jacobian matrix $\mathbf{H}$ to calculate the correction vector $\delta = [\delta_x, \delta_y, \delta_z, \delta_t]^T$ obtained by solving the normal equation:

$$\delta = (\mathbf{H}^T\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{W}\mathbf{v}, \tag{2}$$

where $\mathbf{W}$ is the weight matrix.

In this implementation, we use the **unweighted least squares** approach, i.e., $\mathbf{W} = \mathbf{I}$ (identity matrix), assuming equal reliability across all measurements. This simplifies Equation (2) to:

$$\delta = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{v}. \tag{3}$$

The state vector is then updated iteratively:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \delta_x, \quad \Delta t_{k+1} = \Delta t_k + \delta_t,$$

until convergence or maximum iterations are reached [4, 5].

4 Code Listing

In my work, the SPP_LSE fuction is achieved by:

```python
import numpy as np
import pandas as pd
import os

def lse_spp(pseudoranges, sat_positions, initial_pos=None, max_iter=15,
    tol=1e-3):
    N = len(pseudoranges)
    if N < 4:
        return np.array([np.nan, np.nan, np.nan, np.nan]), np.full(N,
            np.nan), False
    if initial_pos is None:
        initial_pos = np.array([0.0, 0.0, 6371000.0])
    state = np.concatenate([initial_pos, [0.0]])  # [x, y, z, b]
```

```
    for _ in range(max_iter):
        pos = state[:3]
        b = state[3]
        geometric_ranges = np.linalg.norm(sat_positions - pos, axis=1)
        predicted_pr = geometric_ranges + b
        residuals = pseudoranges - predicted_pr

        # Jacobian
        J = np.zeros((N, 4))
        for j in range(N):
            dx = pos - sat_positions[j]
            r = geometric_ranges[j]
            if r < 1e-6:
                raise ValueError("Satellite too close to user estimate.
                    ")
            J[j, :3] = dx / r
            J[j, 3] = 1.0
        JTJ = J.T @ J
        JTr = J.T @ residuals
        try:
            delta = np.linalg.solve(JTJ, JTr)
        except np.linalg.LinAlgError:
            return np.array([np.nan, np.nan, np.nan, np.nan]),
               residuals, False
        state += delta
        if np.linalg.norm(delta[:3]) < tol:
            return state, residuals, True

    return state, residuals, False
```

Listing 1: SPP least-squares solver.

5 Results and Discussion

After being processed by my SPP_LSE code, the estimation result is shown below via various visualization. Present positioning outputs, concise error analysis, and urban-environment interpretation. Example figure (position scatter):

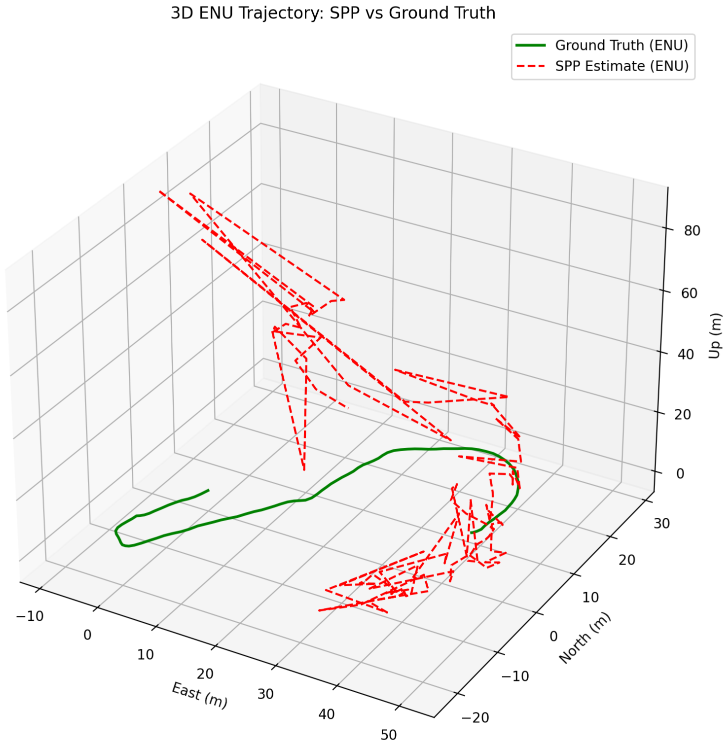

Figure 7: 3D ENU trajectory: SPP vs Ground truth.

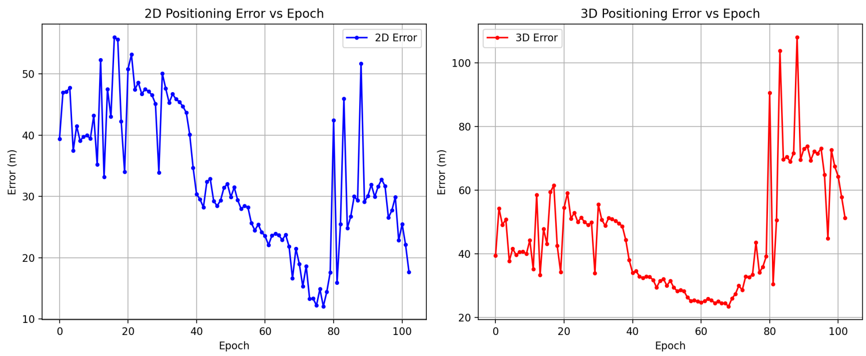

Figure 8: 2D positioning error and 3D positioning error vs epoch

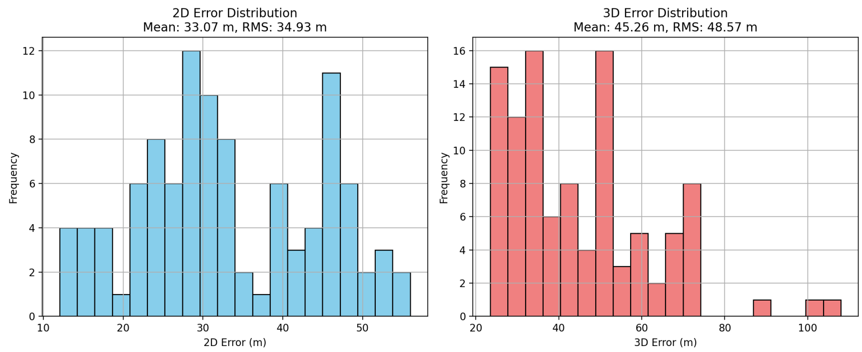

Figure 9: 2D and 3D error distribution

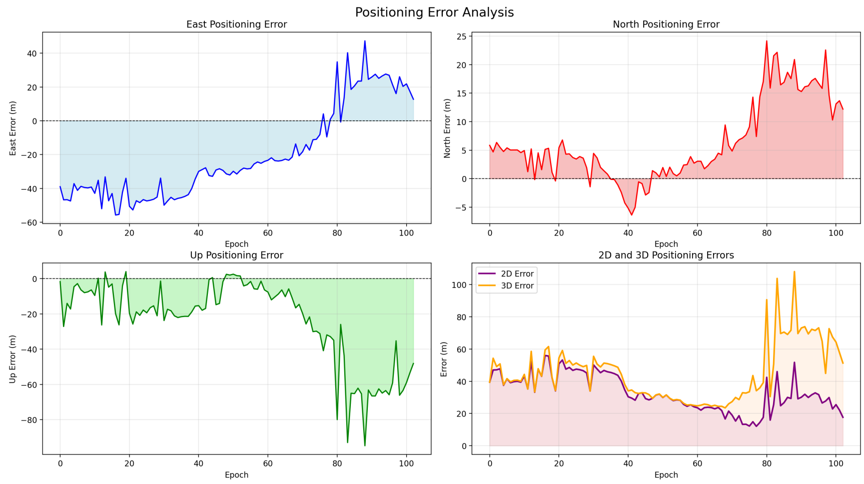

Figure 10: Positioning error analysis in ENU direction and in overall

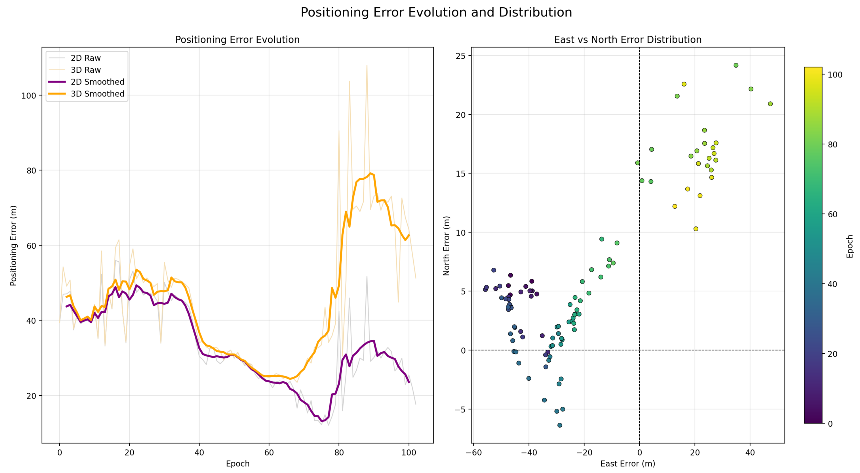

Figure 11: Positioning error evolution and distribution

The 2D average positioning error of my unweighted positioning code is 33.07 meters, and the 3D average positioning error is 45.26 meters. In an urban environment, this level is considered normal for unweighted positioning.

From these visualized images, it is clearly observable that there is a significant fluctuation in my positioning results, and they are more sensitive to changes in altitude. In the later stage, that is, after approximately the 60th time point, there is a considerable height variation in our data collection path. Coupled with the fact that the height is affected more by the influence of buildings, this leads to an increase in three-dimensional errors. This is a common problem in global navigation satellite system positioning (pseudo-range errors are usually more pronounced in the altitude dimension).

Meanwhile, as shown in Figure 10 and Figure 11, we can observe the horizontal error distribution in the ENU coordinate system. This error distribution is mainly influenced by the distribution of surrounding buildings, including non-line-of-sight signals and multipath effects caused by buildings. In our data collection path, the building obstructions in the west and north areas may be more severe, which may cause the positioning result to deviate towards the southeast. This conjecture is roughly consistent with our positioning results (since it is a dynamic path, the influence of buildings changes with our position, and there will be corresponding variations).

References

[1] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, 2nd ed. Norwood, MA, USA: Artech House, 2013.

[2] E. D. Kaplan and C. J. Hegarty, *Understanding GPS/GNSS: Principles and Applications*, 3rd ed. Norwood, MA, USA: Artech House, 2017.

[3] O. Montenbruck, P. Steigenberger, L. Prange, Z. Deng, Q. Zhao, F. Perosanz, I. Romero, C. Noll, A. Stürze, G. Weber, R. Schmid, K. MacLeod, and S. Schaer, "The Multi-GNSS Experiment (MGEX) of the International GNSS Service (IGS) – achievements, prospects and challenges," *Adv. Space Res.*, vol. 59, no. 7, pp. 1671–1697, Apr. 2017.

[4] P. Misra and P. Enge, GPS: Signals, Measurements, and Performance, 2nd ed., Ganga-Jamuna Press, 2011.

[5] P. J. G. Teunissen and O. Montenbruck, Springer Handbook of GNSS, Springer, 2017.