In this laboratory report, we will leverage the RTKLIB (GUI) to process the U-Blox RINEX data in the medium urban case of the UrbanNav dataset. As for the coding part, run the main script with the prefix ""main_evaluation_" to generate the evaluated results and the figures. All the used figures in this report have also been include in the folder "Figures" in the repository.

1 RTKLIB Settings

The file path settings are summarized in the following Table 1.

Table 1: RTKLIB GUI settings

File Type	File Name
Observation file of reference station	hksc141g.21o
Observation file of receiver (U-Blox F9P)	20210521.medium-urban.whampoa.ublox.f9p.obs
Ephemeris (GLONASS)	hksc141g.21g
Ephemeris (GALILEO)	hksc141g.21l
Ephemeris (GPS)	hksc141g.21n

2 Parameters and Analysis

Parameters regarding positioning mode, filter settings, and satellite selection criteria will be tuned and analyzed in the following paragraphs. For the convenience of comparison, we configure the following settings to be a benchmark, as summarized in the Table 2. For each tuned parameter, the evaluation will be conducted in the following several

Table 2: Benchmark settings

Parameters	Value
Filter Type	Forward
Elevation Mask	OFF
Positioning Mode	Kinematic
Signal-Noise-Ratio (SNR) Mask	OFF
Frequencies	L1+L2+L5
Included Constellation	GPS, GLONASS, Galileo, QZSS, SBAS, BeiDou, IRNSS

aspects:

- **2D trajectory**: showing the scatter plot of latitude-longitude coordinates throughout all epochs.

- **2D positioning error**: showing the positioning quality throughout all epochs, as well as the corresponding root means square error (RMSE) and standard deviation (STD).

- **Number of valid satellite signals**: the number of satellites signals used for positioning, which reflects the extent of open-sky view of the surroundings and the quality of the signals.

2.1 Filter Type

Foward filter means that the data is processed from the earliest epoch to the latest one, and the physical scenario with forward filter is real-time positioning. The backward filter means the data is reversely processed from the latest epoch to the earliest one, which is applied in post-processing. The backward setting may improve the positioning accuracy by using "future" data to support "historical" data. Additionally, the combined filtering combines both the forward and backward process, which stores all the forward-processed data, then implements a Rauch-Tung-Striebel (RTS) smoother to smoothen the states along the trajectory during the backward filtering.

Since we are using "kinematic" positioning mode by default, the Kalman filter is active, the forward, backward, and combined filtering may generate different positioning results, and we will discuss as follows.
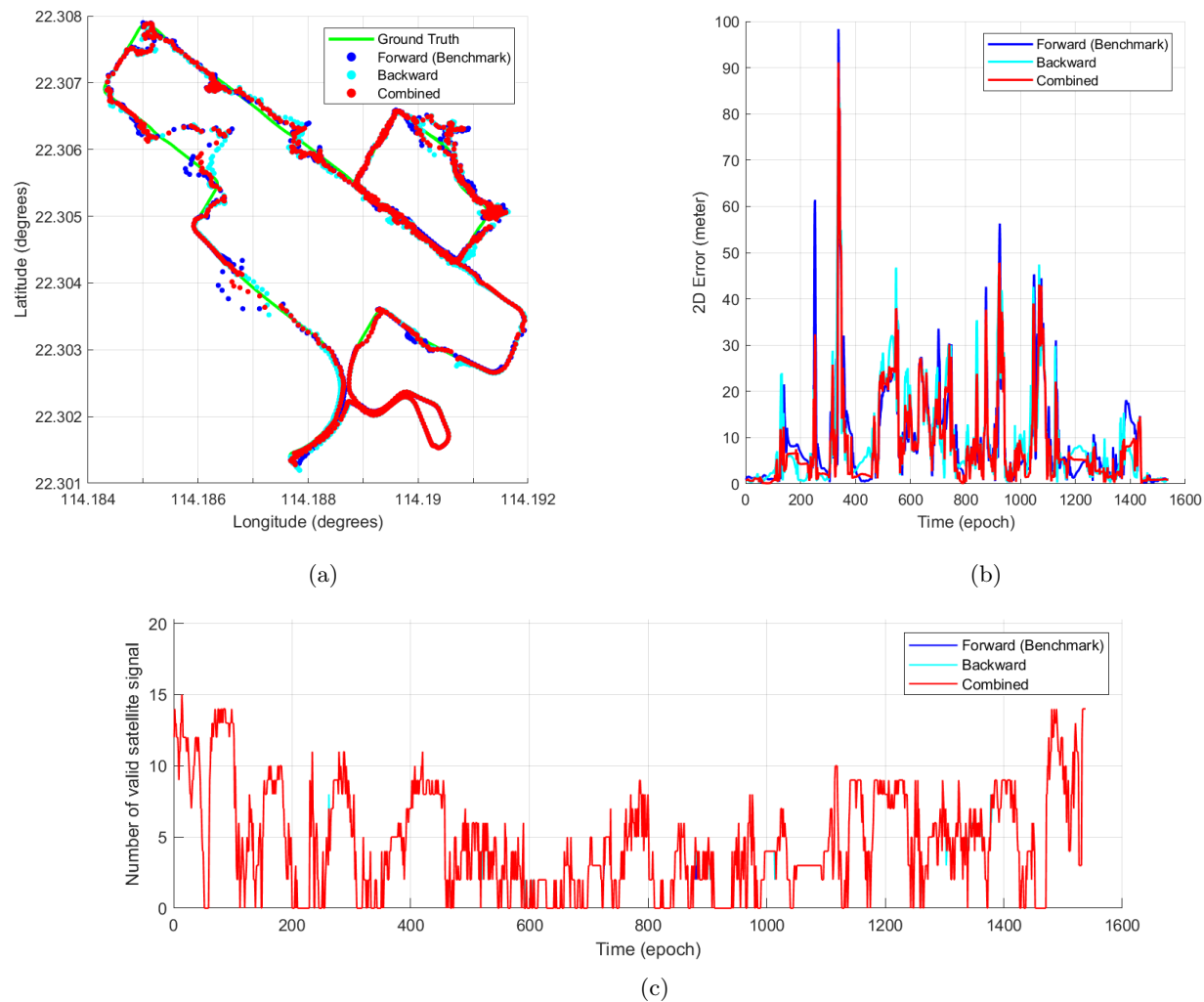

(a)

(b)

(c)

Figure 1: Evaluated results, including (a) 2D trajectories; (b) 2D positioning error; (c) number of received satellites, when tuning Kalman filter type. Three types including "forward (benchmark)", "backward", and "combined" are discussed.

It can be seen from both Figure 1 and Table 3 that the combined filtering type provides the most improvement to the positioning results given by the medium urban GNSS dataset. The combined mode has the lowest RMSE and STD, while the backward mode slightly improves the positioning accuracy and precision compared to the forward

Table 3: Comparison between benchmark and results given by tuned results

Filter type	RMSE (m)	STD (m)	Number of Epochs having positioning solution
Foward (benchmark)	14.28	10.84	1538
Backward	14.26	10.91	1538
Combined	13.23	10.25	1538

mode. Notably, the combined mode takes the longest processing time as it contains both a forward and backward process. Besides, the Kalman filtering mode does not set satellite selection thus the three filtering mode receives approximately the same number of valid satellites, as shown in Figure 1c. Notably, there might be zero number of valid satellites in "kinematic" mode because it set strict selection standards, including continuous phase lock and resolved ambiguities. For these particular epochs, the Kalman filter will propagate the model prediction without the update of GNSS measurements. As a result, positioning solution is available for all epochs.

2.2 Elevation mask

An elevation mask of 30° is applied to all epochs, and only the satellite signals with an elevation angle higher than 30° will be utilized for positioning. This setting is designed to filter out non-line-of-sight (NLOS) or multipath satellites that might get blocked by tall buildings in the urban scenario.

The positionings given by the benchmark settings and the tuned elevation mask will be compared in Figure 2, and the 2D RMSE and STD of the positionings are displayed in Table 4.

Table 4: Comparison between benchmark and results given by tuned results

Elevation mask setting	RMSE (m)	STD (m)	Number of Epochs having positioning solution
No elevation mask (benchmark)	14.28	10.84	1538
With elevation mask	15.35	11.24	1538

Our expected outcome is that the RMSE and STD can decrease because Elevation mask should filter some "bad" satellite with potential NLOS or multipath. However, results in Table 4 shows that this is not necessarily the case. This can be explained that when the degree is fixed to 30°, some functional satellites are also being excluded from the dataset, and this operation contributes to the drop of the positioning performance. An improvement could be setting the skymask which sets the elevation mask based on the azimuth angle. This will avoid the case that LOS satellite signals with low elevation is filtered out.

2.3 Signal-Noise Ratio (SNR) mask

A SNR mask of 25 dB is applied to all epochs, and only the satellite signals with an elevation angle higher than 25 dB will be utilized for positioning. The logic lies that reflected or diffracted satellite signals tend to have lower SNR. A SNR mask is designed to filter out potential NLOS or destructive multipath satellites signals.

The positionings given by the benchmark settings and the tuned SNR mask will be compared in Figure 3, and the 2D RMSE and STD of the positionings are displayed in Table 5.

Same as the elevation mask options, when the SNR mask is applied in the RTKlib on the data, RMSE and STD become slightly larger. Reasons may lie that the SNR mask with a fixed magnitude may filter out some LOS signal with low signal strength, thus decreasing the positioning accuracy. Improvement. A better approach could be replacing the fixed SNR mask to be the adaptive one that is based on the elevation angle.
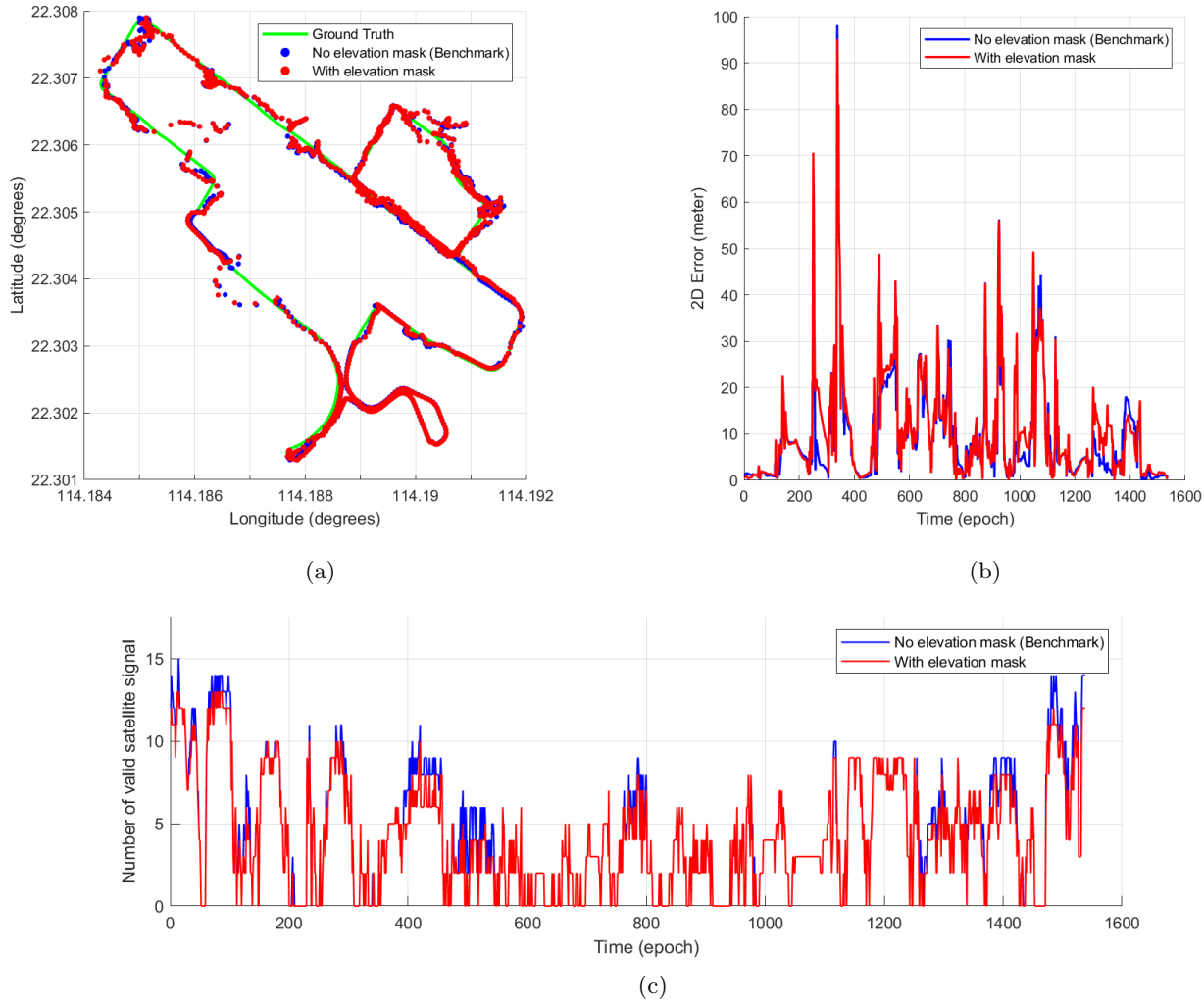
(a)

(b)

(c)

Figure 2: Evaluated results, including (a) 2D trajectories; (b) 2D positioning error; (c) number of received satellites, when tuning elevation mask settings. Two types, including "No elevation mask (benchmark)" and "With elevation mask", are discussed.

Table 5: Comparison between benchmark and results given by tuned results

SNR mask setting	RMSE (m)	STD (m)	textbfNumber of Epochs having positioning solution
No SNR mask (benchmark)	14.28	10.84	1538
With SNR mask	14.10	11.29	1492

2.4 Constellation type

The setting of the single constellation (GPS only) will be explored in this section. The positionings given by the benchmark settings and the tuned constellation settings will be compared in Figure 4, and the 2D RMSE and STD of the positionings are displayed in Table 6.

As is expected, the performance of one single constellation (GPS) is worse than that given by multiple constellations.
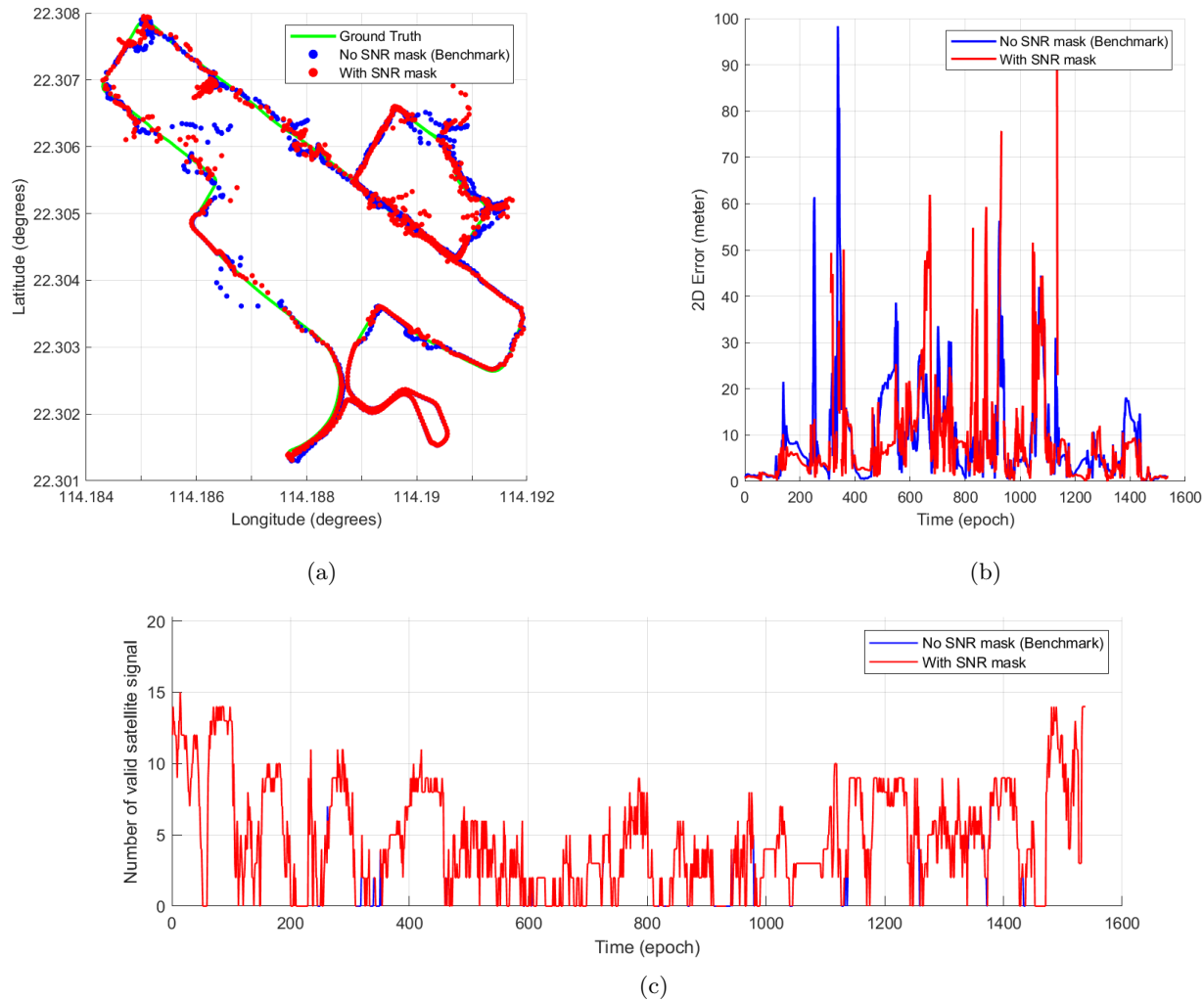
(a)

(b)

(c)

Figure 3: Evaluated results, including (a) 2D trajectories; (b) 2D positioning error; (c) number of received satellites, when tuning SNR mask settings. Two types, including "No SNR mask (benchmark)" and "With SNR mask", are discussed.

Table 6: Comparison between benchmark and results given by tuned results

Constellation setting	RMSE (m)	STD (m)	textbfNumber of Epochs having positioning solution
Multiple Consellation (benchmark)	14.28	10.84	1538
Single (GPS) constellation	32.24	27.99	1456

We can see in Figure 4a that the GPS positionings are more sparsely distributed with lower precision. Figure 4b shows that there can also be large error up to 250m. Due to the reduced number of valid satellites (Figure 4c), both the RMSE and STD given by the single constellation are much higher than those given by multiple constellations. And there are only 1456 epochs having positioning results using GPS only.
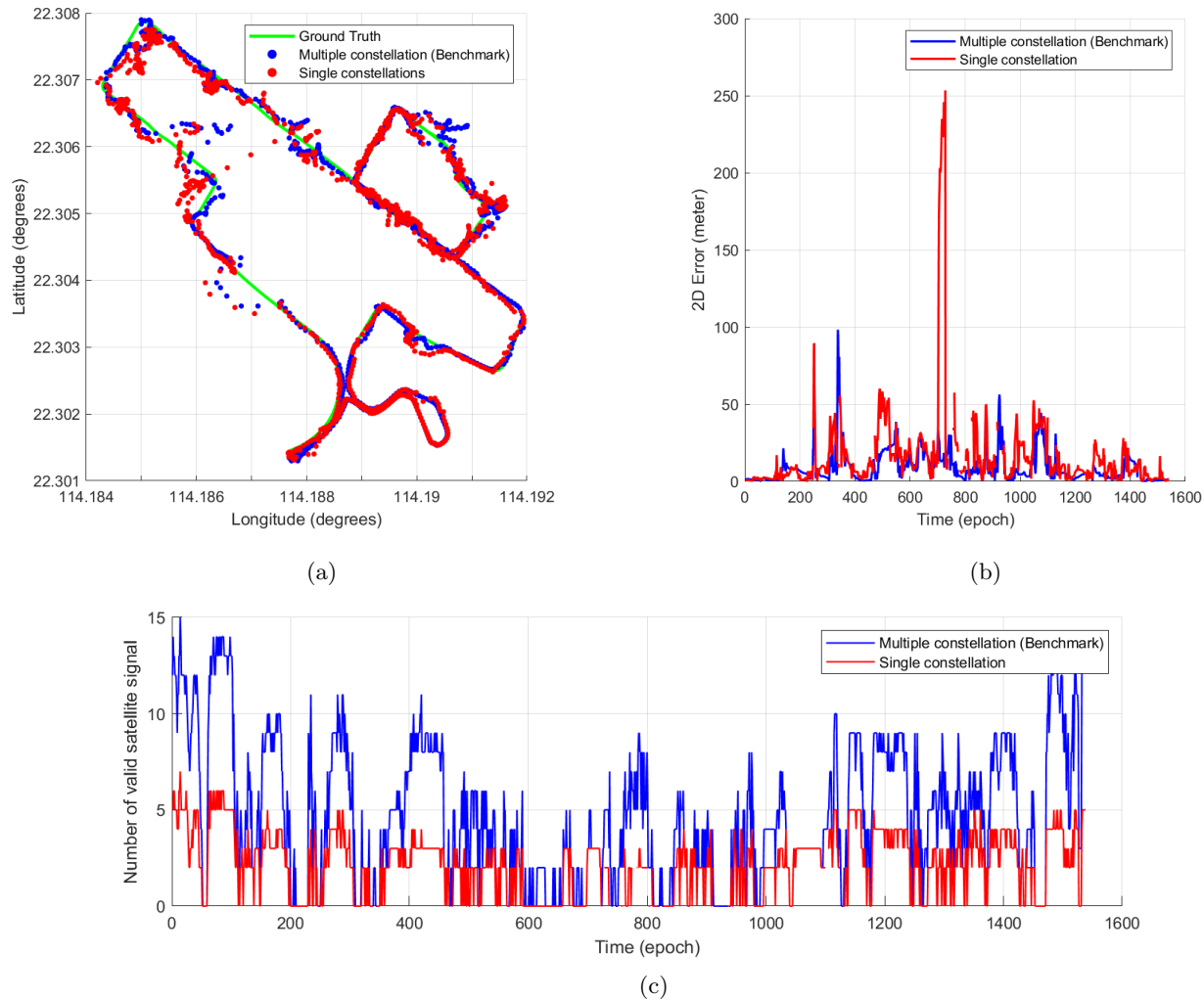
Figure 4: Evaluated results, including (a) 2D trajectories; (b) 2D positioning error; (c) number of received satellites, when tuning SNR mask settings. Two types, including "No SNR mask (benchmark)" and "With SNR mask", are discussed.

2.5 Positioning mode

Suggested discussion on the results.:

1. Might explain the methodology/principle differences between the positioning modes

2. Analyze the results presented by Figure 5 and the Table 7.

As it shows in the Figure 5, there are 4 positioning modes used incomparison. Figure 5a shows the overall positioning performance in a moving scenario, where the green line represents the ground truth, the dark blue dots represents the positioninig trajactory via kinematic mode, the pink dots represents the positioning results of the static mode, the light blue dots are the trajectory of the single mode and the red lines are the results of the differential mode.

- **Kinematic mode**: The algorithm used in the kinematic mode is the combination of carrier phase differential and the real time kinematic (RTK), which estimates position and ambiguity in real-time using the extended kalman filter (EKF). It utilises the data from both the receiver and the reference base station, so that both the clock bias of the satellite and the receiver can be eliminated. The kinematic mode is suitable for dynamic scenarios, with the model incorporating velocity and state transition.

- **Static mode**: For static mode, it shares the similar algorithm as the kinematic mode, and it also requires both the receiver and the reference base station for positioning. Unlike the kinematic mode, the state model does not include velocity terms and the position is typically modeled as constant or slowly varying. Therefore, the static mode is not suitable for dynamic positioning and the use case for the static mode is usually the positioning of the ground truth.

- **Single mode**: The observables for the single point positioning (SPP) are only the pseudorange measurements, and it can derive the user's location directly by solving the positioning equation based on least squares (LS) and the EKF. However, due to the lack of the estimation of other error sources, such as the ionospheric delay, tropospheric delay, and multipath, the positioning accuracy of the SPP is not that satisfactory, compared to the kinematic mode and the static mode.

- **Differential mode**: As the improvement of the SPP, differential mode is enhanced by using the observables from the other reference base station, which utilize both the signals from the satellite and the reference base station to eliminate the error caused by the ionospheric delay and tropospheric delay. However, it requires both the receiver and the base station receive the same satellite signals. Therefore, the limitation of the differential mode is the restricted coverage.

Table 7: Comparison between benchmark and results given by tuned results

Positioning mode setting	RMSE (m)	STD (m)	textbfNumber of Epochs having positioning solution
Kinematic (benchmark)	14.28	10.84	1538
Static	287.69	145.62	932
Single	20.19	12.91	1526
Differential	15.21	11.12	1535

We can observe that the static positioning mode gives the worst performance. It has the highest 2d positioning error, as shown in Figure 5b. It has the least number of valid satellites (reflected in Figure 5d) and has the least number of epochs having positioning solutions (i.e., 932). Accordingly, the RMSE and STD are significantly higher than other modes. As explained in the previous paragraph, the static mode assumes constant or slowly-varying position model without considering the velocity, thus is not suitable for our dynamic dataset.

Single mode represents SPP, which has the least strict selection standards for valid satellites, thus overall presenting the highest level in Figure 5d. It has the second largest RMSE and STD, given that the data is from uBlox F9P only. In contrast, the differential mode use observation dataset from both the reference station and the receiver, thus further reducing RMSE and STD. This mode also provides positioning solutions in almost every epochs (1535).

The benchmark (i.e., kinematic mode) considers both the pseudorange measurement and the carrier phase measurements, thus giving the least RMSE and STD shown in Table 7, as is expected for our dynamic dataset.

The reason why the static mode has the worst positioning performance is obvious, as it is incapable of solving the errors caused by the ionospheric delay and the tropospheric delay. The differential mode outperforms the static mode is that the receiver has a certain mobility, and it is not suitable for static mode.

As for the number of available satellites for different positioning mode, the single mode only requires the pseudorange measurements, therefore, it can have the most satellites for positioning. The number of available satellites for differential mode is a little bit less than the single mode, it is due to the fact that compared to the single mode, the differential additionally requires the reference base station to receive the signals from the same satellites. The

reason why the static and the kinematic mode have less number of available satellites is that they ask for signals at both L1 and L2 frequency.

3 Strengths and limitations of the library

3.1 Strength

The RTKlib GUI has a graphical interface that is very helpful in terms of demonstration usage. It is especially friendly for users who have no coding knowledge. In addition, the graphical interface provides an interactive operation mode, and it is easy for the user to play with different settings. The results can be plotted well, and there are no further actions for the users to visualize the positioning results. Besides, compared to other software receivers, it compiles the given dataset quickly, since it is written in C. For the MATLAB-based software receivers, such as softgnss, the RTKlib saves time for running the code. Also, the RTKlib can cater for developing other embedded operating systems, and the real-time positioning platform can be built based on the RTKlib.

3.2 Limitation

The current RTKLIB GUI version only supports a limited number of functions in the interface. The parameter that can be tuned is fixed and less flexible than the RTKLIB written in programming languages. Besides, the RTKlib is best written in C code, but it is very difficult for users to make any other modifications, such as the modification of the algorithm. It may require the users to have very solid coding capabilities. Therefore, in terms of doing research related to signal processing, other than using the output positioning results, the RTKlib has more limitations compared to the MATLAB-based software receivers.

3.3 Suggestions for improvements

1. The process for selecting GNSS library needs improvement. Currently, when using the RTKlib for positioning, we need to input the original files from the UBLOX, and we need to select the needed types of satellites to do the positioning. Therefore, we have to do our own decision. Further improvements can be made for more automation. For example, the improved version of RTKlib could select the satellite constellations based on the received signal qualities and so on.

2. The parameter tuning process can be further improved. Currently, we tune a parameter and conduct the processing throughout the whole dataset, which is time-consuming. If we deliberately extract a known GNSS-challenging fraction of the dataset to test the performance given by the tuned parameter, the process is expected to be shorter. We can more quickly switch through different values of the parameters, thus obtaining the optimal value more efficiently.
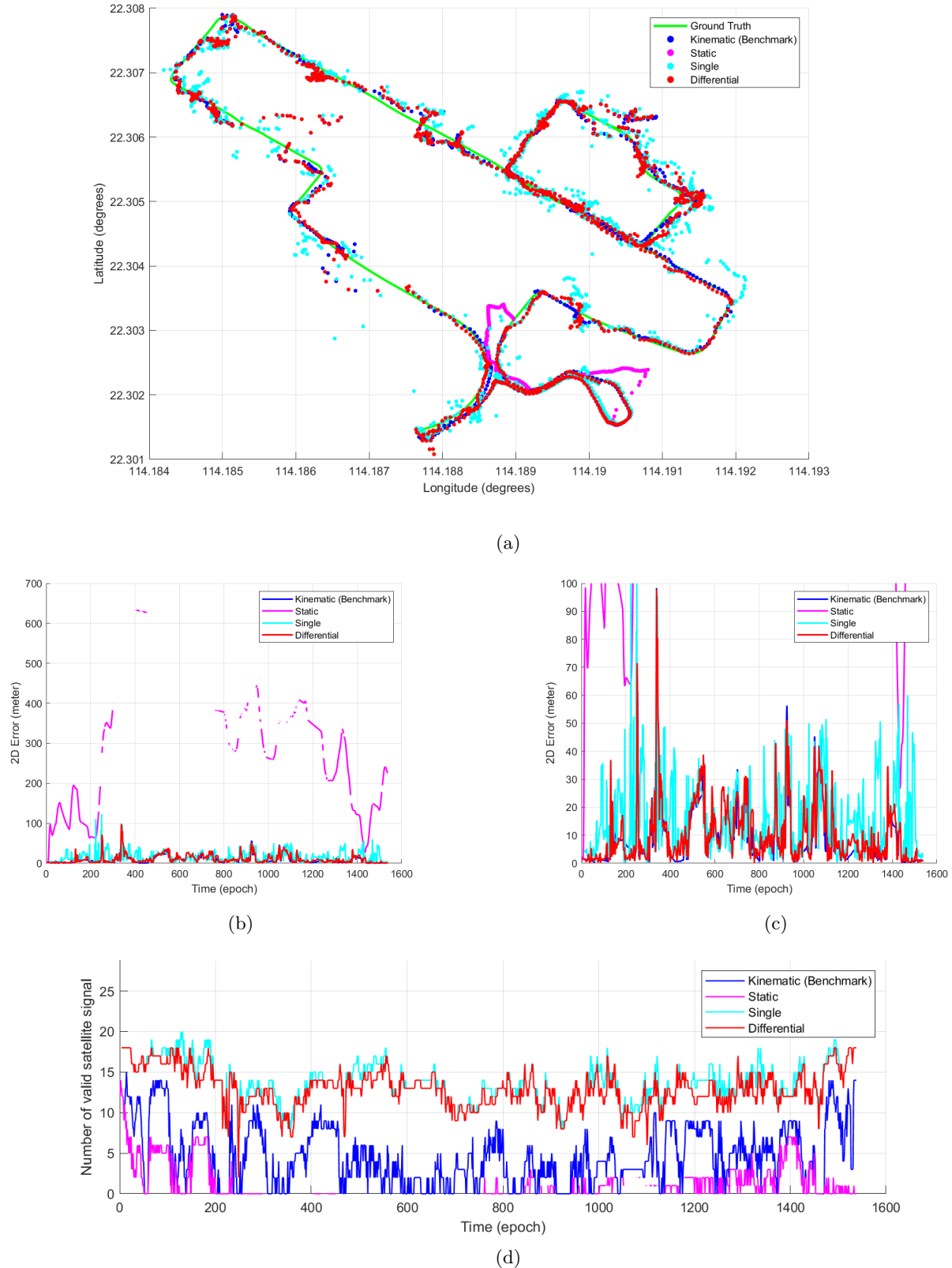
(a)

(b)

(c)

(d)

Figure 5: Evaluated results, including (a) 2D trajectories; (b) 2D positioning error; (c) zoom-in view of 2D positioning error (below 100m); (d) number of received satellites, when tuning SNR mask settings. Four types, including "Kinematic (benchmark)", "Static", "Single", and "Differential" modes, are discussed.