Android GNSS Measurements under Spoofing and Interference

Andrea Botticella[*,†]
andrea.botticella@studenti.polito.it

Elia Innocenti[*,†]
elia.innocenti@studenti.polito.it

Renato Mignone[*,†]
renato.mignone@studenti.polito.it

Simone Romano[*,†]
simone.romano2@studenti.polito.it

ABSTRACT

This study explores the behavior of GNSS navigation on a commercial Android smartphone by analyzing raw measurements collected in both static and dynamic conditions. Using Google's GNSS Logger and MATLAB-based post-processing, we evaluate positioning accuracy, signal quality, and clock stability under normal operation and under spoofed inputs—where false positions and delays are injected into the data. The results highlight the vulnerability of smartphone GNSS to software spoofing and underline diagnostic features that may aid in anomaly detection and robustness improvement.

1 INTRODUCTION

Global Navigation Satellite Systems (GNSS) are fundamental to modern positioning services, widely embedded in smartphones and critical infrastructure. Despite their ubiquity, GNSS signals are inherently vulnerable to spoofing—an attack technique where counterfeit signals deceive the receiver into computing false location or timing information. To develop robust mitigation strategies, it is essential to understand how smartphone GNSS systems behave under both normal and manipulated conditions.

This work investigates the GNSS measurement behavior of a consumer-grade Android smartphone across different operating scenarios. We analyze baseline performance by collecting and processing raw GNSS data in two real-world environments: a static session on the rooftop of Monte dei Cappuccini and a dynamic session aboard a tram in urban Turin. In addition to this baseline study, we simulate spoofing by injecting false position inputs into the processing pipeline and, separately, by introducing artificial delays.

The remainder of this report is organized as follows.

- **Section 2** describes the experimental setup, including device configuration, data collection procedures, and the processing pipeline.
- **Section 3** presents results and discussion, contrasting static versus dynamic performance, examining spoofed-location impacts, and analyzing delay effects.
- **Section 4** summarizes the key findings and outlines directions for future work.

2 METHODS

2.1 Devices and Software

We used a Samsung Galaxy A51 with Android 11 for this experiment. GNSS Logger v3.1.0.4 was chosen due to its unrestricted access to raw GNSS measurements, compatibility with newer Android APIs, and ability to record detailed GNSS data that is suitable for precise analysis. MATLAB R2024b was employed to handle data because it comes with Google's GNSS toolbox, which accommodates robust analysis and visualization of GNSS measurements and position solutions.

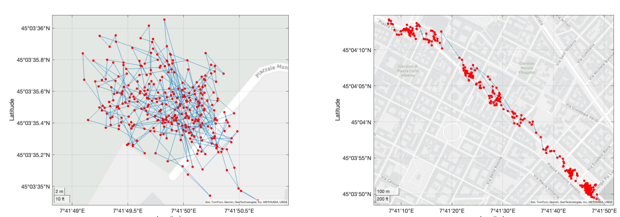
2.2 Data Collection Procedure

Two distinct 5-minute GNSS data logging sessions were conducted on 3 May 2025, under cloudy weather conditions, using the GNSS Logger app configured with the following settings enabled:

- **GNSS Location:** Enabled to capture location data.
- **GNSS Measurements:** Enabled to log raw GNSS measurements.
- **Navigation Messages:** Enabled to capture navigation data.
- **GnssStatus:** Enabled to log GNSS status information.
- **Sensors:** Enabled to capture sensor data.

The sessions were designed to capture both static and dynamic GNSS performance, with the following details:

- **Static Scenario:** Performed on the rooftop of Monte dei Cappuccini, Turin, starting at 10:35:20. The device was stationary throughout the entire session, providing baseline measurements.
- **Dynamic Scenario:** Conducted on tram line 15 from Piazza Castello to Piazza Vittorio Veneto, starting at 10:00:21, simulating a typical urban mobility scenario.



(a) Monte dei Cappuccini. **(b)** Tram Line 15.

Figure 1: Comparison of GNSS data: (a) Static scenario at Monte dei Cappuccini,(b) dynamic scenario along Tram Line 15.

2.3 Processing Pipeline

The raw GNSS data from the GNSS Logger served as the input dataset for MATLAB. Processing involved a scripted workflow via `ProcessGnssMeasScript.m`, where the following steps were executed:

1. **Filtering:** Data points with a carrier-to-noise ratio below 25 dB-Hz or satellite elevations below 15° were excluded to improve accuracy.

[*]The authors collaborated closely in developing this project.
[†]All the authors are students at Politecnico di Torino, Turin, Italy.

2. **Measurement Extraction:** Pseudorange and Doppler measurements were computed from GNSS timestamps and satellite transmission data.
3. **Weighted Least Squares (WLS) Positioning:** Applied to derive precise positioning and clock bias estimates.
4. **Visualization and Comparison:** Output plots from MATLAB, including pseudorange, pseudorange rates, and position solutions, were generated to facilitate comparative analysis of the static and dynamic scenarios.

Results from this processing pipeline provided insights into the differences in GNSS performance under static and dynamic conditions.

2.4 Spoofed-Input Configuration

Spoofing scenarios were emulated by introducing artificial variations to the recorded GNSS data through MATLAB processing. Specifically, mock positions were assigned by adjusting the parameter `spoof.position`, which represents modified latitude, longitude, and altitude coordinates. Additionally, artificial time delays were tested by adjusting the `spoof.delay` parameter, typically in milliseconds, to mimic delayed GNSS signal arrival. Such configurations facilitated evaluation of the impact of spoofing scenarios on position estimation reliability and accuracy.

2.5 Optional Interference Scenario

Even a worst-case interference situation was simulated, replicating conditions in the vicinity of potential interference sources such as broadcasting antennas or communication areas of high density. GNSS data were collected near such sources of interference, and then processed with the same MATLAB procedure. Nominal condition comparison was established to analyze the impact of external interference on GNSS observables such as variation in pseudorange measurements, carrier-to-noise ratio, and positional accuracy overall.

3 RESULTS AND DISCUSSIONS

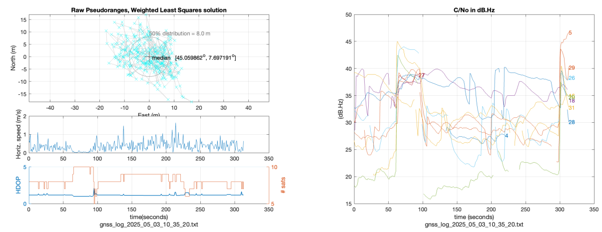
3.1 Baseline Performance: Static vs. Dynamic

3.1.1 Static Case: Monte dei Cappuccini.

The static scenario demonstrated stable GNSS performance. Pseudorange values remained consistent with expected GPS satellite distances (around 2×10^7 m). Position estimates clustered within an 8 m radius of the median, and horizontal speed remained near zero, confirming the device was stationary (Figure 2a). C/No values showed minor fluctuations (Figure 2b), with a few satellites (e.g., SV 27) experiencing temporary signal degradation. HDOP remained generally low, indicating good satellite geometry. Clock bias increased linearly over time (Figure 3), suggesting typical thermal-induced drift.

3.1.2 Dynamic Case: Tram Ride.

The dynamic scenario exhibited greater measurement variability. Pseudoranges changed more rapidly (Figure 4a), reflecting motion relative to the satellites. Position estimates were more dispersed, with northward velocity peaking near 20 m/s (Figure 4b). C/No values fluctuated significantly due to urban obstructions (Figure 5a), and some satellites showed weak or intermittent signals (e.g., SV 18). HDOP remained within acceptable bounds, although the number of tracked satellites varied. Clock bias again increased linearly but was more affected by motion-induced dynamics (Figure 5b).



(a) Static scenario: Position estimates, horizontal speed, HDOP.
(b) Static scenario: Carrier-to-noise ratio over time.

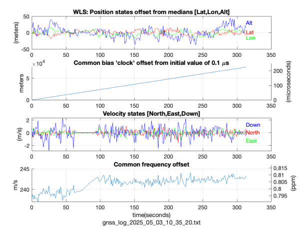
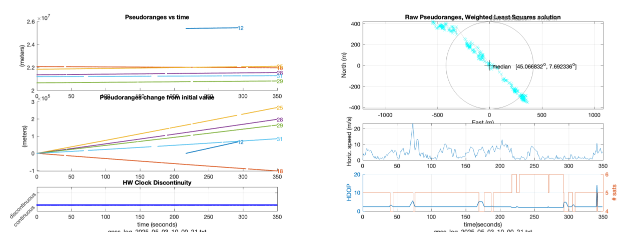
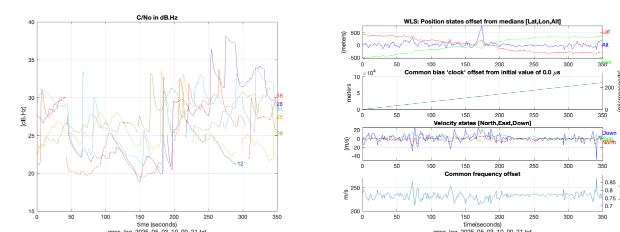


Figure 3: Static scenario: WLS state offsets and clock bias trends.



(a) Dynamic scenario: Pseudoranges and variation over time.
(b) Dynamic scenario: Position estimates, speed, and HDOP.



(a) Dynamic scenario: Signal quality (C/No) per satellite.
(b) Dynamic scenario: WLS offset and velocity states.

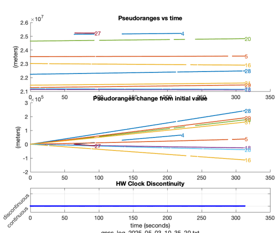
3.1.3 Comparison of Static and Dynamic Performance.

Key differences emerged when comparing static and dynamic scenarios:

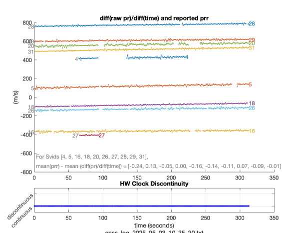
- **Pseudorange Trends:** Static measurements showed flat and stable pseudorange lines with minor steps due to satellite handovers or clock corrections (Figure 6a). In contrast, dynamic measurements exhibited sloped lines, reflecting relative motion to satellites.

- **Doppler Residuals:** In the static case, calculated and reported pseudorange rates aligned closely (Figure 6b). The dynamic case introduced more noise and discrepancies (Figure 7), reflecting movement complexity and potential modeling errors.
- **Signal Quality (C/No):** The static case showed relatively stable C/No values, while the dynamic case had significant fluctuations and lower mean values due to urban interference and multipath effects.
- **HDOP and Satellite Geometry:** HDOP remained low and stable in the static case, while it was slightly higher and more variable in the dynamic scenario due to changing satellite visibility.
- **Error Statistics:** The static case achieved tighter clustering (~8 m radius), whereas the dynamic case showed larger offsets, especially in the latitude/longitude axes (up to ~500 m).
- **Velocity and Clock Behavior:** The dynamic case revealed dominant northward motion and larger variation in velocity states. Clock bias trends were similar in both scenarios but more affected by noise in the tram ride.

These differences reflect the impact of movement, signal obstruction, and urban dynamics on GNSS performance, confirming that dynamic environments introduce more variability and require more robust estimation strategies.



(a) Static scenario: Pseudoranges and variation over time.



(b) Static scenario: Reported vs computed PRR.

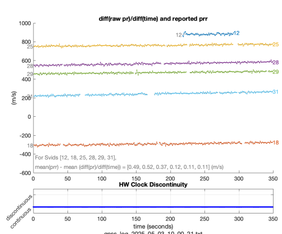


Figure 7: Dynamic scenario: Reported vs computed PRR.

3.2 Impact of Spoofed Position

3.3 Effects of Timing Delays

3.4 Interference Effects

4 CONCLUSIONS

A APPENDIX

"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."