

# Wide-Lane Ambiguity Support as an Enabler of Low Cost, Dual-Band Precision GNSS Performance in Modern, Large Scale Correction Networks

Daniel Burr, *Swift Navigation*  
Mike Horton, *HYFIX*

## BIOGRAPHY

**Daniel Burr** Daniel Burr is a Principal Application Engineer at Swift Navigation who has worked with GNSS applications for nearly 20 years in the marine, agriculture and automotive industries. Prior to joining Swift in 2019, he worked as a System Architect for autonomous driving systems at Intel and as a Team Lead at Topcon Precision Agriculture. Daniel holds a BSc (Computer Science) from the University of Sydney and a BE (Computer Systems) from the University of Queensland.

**Mike Horton** Mike Horton is the co-founder and Chief Technology Officer of ANELLO Photonics, a developer of silicon photonic inertial sensors. He is also the co-founder and CEO of HYFIX.AI, a developer of IoT GNSS solutions. Prior to these roles, he founded Crossbow Technology in 1995 and served as CEO until it was successfully sold to Moog Aerospace in 2011. He earned a BS/MS in Electric Engineering from UC Berkeley.

## ABSTRACT

This paper demonstrates how the implementation of support for wide-lane ambiguity resolution in low cost, dual-band GNSS receivers can enable seamless, high-accuracy applications at a continental scale. Results are detailed using correction data from Swift Navigation's Skylark™ Precise Positioning Service in combination with different positioning engines.

Many different approaches for GNSS corrections have been devised during the past few decades, each having their own strengths and weaknesses. For mass-market applications, the ideal correction technology is one which converges quickly, is applicable across a wide area and reliably provides the desired level of accuracy. A number of commercial services available today, such as Skylark, promise centimetre-level accuracy at continental scale with convergence times measured in seconds.

Complementing the availability of continental-scale correction services, rapid growth can be observed in cost-effective, dual-band GNSS chipsets over the previous few years. These devices are frequently used in small, industry-standard footprint modules and offer a path to achieving levels of performance previously associated with much higher cost hardware. With mass-market deployment of chipsets capable of consuming corrections, it is to be expected that many new verticals will benefit from higher accuracy positioning.

Testing results establish a performance baseline with RTKLIB and analyse the performance improvement which can be obtained through the use of a positioning engine with support for wide-lane ambiguity resolution. The wide-lane capable positioning engines tested were: a modified version of RTKLIB, EmbedCM™ from HYFIX and Starling® from Swift Navigation.

High-level results using RTKLIB with Skylark corrections demonstrate an improvement in  $1\sigma$  Horizontal Accuracy from 29.6 cm (baseline) to 17.0 cm (wide-lane). Similar results were demonstrated with the EmbedCM and Starling positioning engines and will be presented in the paper.

## I. INTRODUCTION

A *wide-lane combination* is a linear combination of measurements from the same satellite, at the same time, on two different frequencies (Collins, 1999). The result is a signal which has a longer wavelength than the two original signals. Instead of attempting to resolve integer ambiguities using the original code/carrier observations, a GNSS positioning engine may instead attempt to perform ambiguity resolution on a wide-lane combination of carrier-phase measurements using traditional ambiguity resolution techniques. The longer wavelength of this combination allows for more resilience against GNSS errors such as ionospheric error (Subirana et al., 2011) and allows the positioning engine to increase the probability of achieving a correct ambiguity resolution solution with poorer quality input observations (at the cost of a small increase in noise in the final position solution) (Urquhart, 2009).

Correction data from Swift Navigation's Skylark correction service has been optimised for the use of linear combination. The

use of linear combinations such as the wide-lane combination has many advantages that are well understood by the positioning community (Collins, 1999; Cocard et al., 2008; Urquhart, 2009), but Skylark is the first correction service to apply these techniques to achieve uniform performance on a continental scale, giving the user centimetre-level precision with fast convergence times.

Section II of this paper describes the modifications made to RTKLIB to introduce support for wide-lane ambiguity resolution. It also includes results from static test scenarios demonstrating the accuracy improvement obtained when using a low cost GNSS receiver with and without these changes.

Section III details the horizontal accuracy observed when using a low cost receiver in combination with Skylark corrections in real-world drive test scenarios. Results are shown with three wide-lane capable positioning engines, namely, RTKLIB, EmbedCM from HYFIX and Starling from Swift Navigation.

## 1. Low Cost GNSS Receivers

The price of dual-frequency, RTK-capable GNSS receivers has decreased dramatically in the past 10 years. Likewise, the definition of what constitutes a "low cost" receiver has evolved over time—as recently as 2018, this included receivers ranging in price from USD200 - USD540 (Janos and Kuras, 2021; Jackson et al., 2018). Today it is possible to purchase equivalent receivers for less than USD100 (for single units), with prices reaching into the single digit USD range for larger volumes.

For the purposes of this paper, a *low cost, dual-band* GNSS receiver is considered to be one which satisfies the following criteria:

1. Is available for USD20 or less (at single unit pricing)
2. Supports two or more GNSS frequency bands (L1/L2, L1/L5 or L1/L2/L5)
3. Supports three or more GNSS constellations
4. Includes an application processor which is capable of self-hosting an RTK engine
5. Optionally: includes additional sensors (e.g. IMU) which can be used to improve the availability of positioning in challenging GNSS environments

The supported frequencies and constellations have important implications in terms of the achievable accuracy and availability. For example, the more modern GPS L5 signal has higher transmission power and improved signal structure compared to previous generations, resulting in better reception in challenging GNSS environments (Leclère et al., 2018). However, since GPS L5 is not available on all active GPS SVs (National Coordination Office for Space-Based Positioning, Navigation, and Timing, 2022), an L1/L5 receiver typically receives fewer dual-frequency GPS signals in comparison with an L1/L2 receiver. For this reason, when selecting an L1/L5 receiver, it is desirable to use one which also supports the Galileo and BeiDou-3 constellations since all of their SVs support signals on the respective L5-equivalent bands (Naciri et al., 2021). In particular, the modernised BeiDou-3 constellation significantly increases the number of available L5 signals across the globe.

The GNSS receiver selected for the following evaluation was the AG3335 from Airoha. This module is based upon the MT3335 from MediaTek and includes a host processor which is capable of running an RTK engine. It is also available in different variants, some of which include an IMU sensor which can be used for Dead Reckoning purposes. However, for the purposes of this paper, a variant without an IMU was used since not all of the selected positioning engines were capable of performing Dead Reckoning.

An important factor which influences the performance of a GNSS receiver is the antenna to which it is connected. Higher quality antennas result in superior code and phase positioning performance (Hamza et al., 2020) but also carry a higher price tag. Given the stated objective of enabling low cost positioning solutions, it is also necessary to make some compromises in terms of antenna quality. For the evaluation in Section III, an automotive-grade patch antenna was used, which is representative of a real-world low cost system but severely limits the achievable performance.

## II. WIDE-LANE CORRECTIONS WITH RTKLIB

RTKLIB (Takasu, 2020) is an Open Source suite of tools by Tomoji Takasu which can be used for capturing, converting and processing GNSS data. It includes a number of GUI (Graphic User Interface) applications (e.g. `rtknavi`, `rtkpost`) and CLI (Command Line) utilities (e.g. `rtkrcv`, `rnx2rtkp`, `str2str`). The `rtkrcv` utility is used as the RTK Engine in a number of GNSS receivers, and the remaining tools are extremely popular for a variety of off-line processing tasks.

Swift Navigation maintains a fork of the upstream RTKLIB 2.4.3 b34 repository (Swift Navigation, 2022). This fork includes the following changes in comparison with the upstream version:

1. Support for handling files in the Swift Binary Protocol (SBP) format (Swift Navigation, 2021)
2. Support for reading base station positions from RINEX header and new site occupancy events, which is needed to handle possible changes in base station location when post-processing data
3. Changes which allow RTKLIB to achieve integer fixes using wide-lane corrections from Swift's Skylark correction service

This repository also includes a few minor bug fixes to the upstream version of RTKLIB.

## 1. Wide-lane Support

The correction data output from the Skylark correction service requires that the Positioning Engine uses wide-lane ambiguity resolution. This means that a linear transformation (namely, subtracting L2/L5 ambiguities from L1 ambiguities) is required on the float ambiguities before performing the search of the closest integer vector. Once resolved to integer, wide-lane ambiguities can be used to constrain the position to get an RTK integer fixed position.

As documented in Appendix E.7 of Takasu (2013), the default implementation of integer ambiguity resolution in RTKLIB transforms the estimated float state vector ( $\hat{x}_k$ ) and its associated covariance matrix ( $\hat{P}_k$ ) to double difference form using the following equations:

$$\hat{x}'_k = G\hat{x}_k = \left( \hat{r}_r^T, \hat{v}_r^T, \hat{N}^T \right)^T \quad (1)$$

$$P'_k = GP_kG^T \quad (2)$$

In these equations,  $G$  denotes the single differenced to double differenced transform matrix, which is defined as:

$$G = \begin{bmatrix} I_{6 \times 6} & & & & \\ & D_{1,1} & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & D_{n,f} \end{bmatrix} \quad (3)$$

where  $D_{i,j}$  is the single differencing matrix (for constellation  $i$  on frequency  $j$ ,  $n$  is the number of constellations and  $f$  is the number of frequencies):

$$D = \begin{bmatrix} 1 & -1 & 0 & \dots & 0 \\ 1 & 0 & -1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & -1 \end{bmatrix} \quad (4)$$

The state vector is shown on the right hand side of Equation 1, where  $\hat{r}_r$  and  $\hat{v}_r$  represent the rover antenna position and velocity respectively and  $\hat{N}$  contains the integer ambiguities.  $\hat{r}_r$  and  $\hat{v}_r$  are 3x3 matrices, so the purpose of the 6x6 identity matrix in the top left of  $G$  is to leave these values unchanged when  $G$  is multiplied by  $\hat{x}_k$ . The elements of  $\hat{N}$  are ordered by constellation/frequency pairs, i.e.  $\hat{N} = \left( \hat{A}_{0,0}, \dots, \hat{A}_{n,f} \right)$  where  $\hat{A}_{i,j}$  is a vector containing the integer ambiguities for constellation  $i$  on frequency  $j$ . The total number of rows in  $G$  is therefore  $6 + \sum_{i=1}^n \sum_{j=1}^f |\hat{A}_{i,j}|$  (5), or for a dual frequency case,  $6 + \sum_{i=1}^n |\hat{A}_{i,1} + \hat{A}_{i,2}|$  (6).

To generate the satellite-side single differencing matrix, RTKLIB selects the satellite with the maximum elevation angle from each constellation on an epoch-by-epoch basis as the reference satellite. Each  $\hat{A}_{i,j}$  vector is sorted such that the first element contains the ambiguity for the selected reference satellite and are then joined with  $\hat{r}_r$  and  $\hat{v}_r$  to create the estimated state vector  $\hat{x}_k$ . The construction of  $G$  is such that when it is multiplied by the state vector, the result (excluding the rows containing the identity matrix) will be a vector where each element corresponds to the difference between an estimated ambiguity and the estimated ambiguity for the corresponding reference satellite, i.e.  $\alpha_{i,j} - \beta_{i,j}$  where  $\alpha_{i,j}$  is the ambiguity for the reference satellite for constellation  $i$  on frequency  $j$  and  $\beta_{i,j}$  is an estimated ambiguity.

Note that in practice—for the sake of efficiency—the RTKLIB implementation does not include every row shown above in  $G$ . Instead, rows are excluded unless the corresponding signals satisfy specific criteria, including lock time validity, half cycle ambiguity validity and elevation above the configured threshold. However, the resulting single difference to double difference transformation matrix (again, excluding the rows containing the identity matrix) still satisfies the property that multiplication will result in the subtraction of an estimated float ambiguity from the appropriate reference ambiguity. The Least Squares Ambiguity Decorrelation Adjustment (LAMBDA) search is then performed on the uncombined ambiguities for each frequency.

The same basic method may be used for integer ambiguity resolution with wide-lane corrections. In this case, the desired result is for  $\hat{x}'_k$  to have elements equal to  $(\alpha_{i,1} - \beta_{i,1}) - (\alpha_{i,2} - \beta_{i,2})$  where the  $i,1$  values are from L1 signals and the  $i,2$  values are from L2 or L5 signals. Therefore the single difference to double difference transform matrix must be adjusted to produce wide-lane signal combinations, as follows:

$$G' = \begin{bmatrix} I_{6 \times 6} & & & & & \\ & D_{1,1} & & & & \\ & & \ddots & & & \\ & & & D_{n,1} & & \\ & & & & -D_{1,2} & \\ & & & & & \ddots \\ & & & & & & -D_{n,2} \end{bmatrix} \quad (7)$$

The `dd_wl_mat()` function in Swift Navigation (2022) provides a working example of how to generate the  $G'$  matrix.

The number of rows in  $G'$  is  $6 + \sum_{i=1}^n (|\hat{A}_{i,1}| - 1)$ . By comparing this with Equation (6), it can be seen that this value is approximately half of the number of rows in the uncombined single difference to double difference transform matrix  $G$ . This means that there are far fewer ambiguities to resolve when using wide-lane corrections, resulting in reduced computational search time for ambiguity resolution (Urquhart, 2009).

Following the resolution of integer ambiguities, an additional modification to RTKLIB is required to correctly restore the single-differenced ambiguities when using wide-lane corrections. In the default (single frequency) implementation, the estimated float state is subtracted from the corresponding fixed state for each signal to compute the ambiguity for the next iteration. However, in the case of wide-lane corrections, there are two signals which contribute to each ambiguity, meaning that this technique cannot be used. Instead, the approach used is to restore the original float ambiguity for the L1 signal and to subtract the fixed ambiguities from both signals from the original L2/L5 float ambiguity. This implementation can be found in the `restamb()` function in Swift Navigation (2022).

## 2. Performance Evaluation

To evaluate the performance impact of introducing support for wide-lane ambiguity resolution into RTKLIB, two 12-hour static tests were performed using a Quectel LC29HA module (containing an Airoha AG3335) connected to a geodetic-grade antenna. RTCM output was logged simultaneously from the Quectel module and the Skylark CRS mountpoint. This output was then converted to RINEX and post-processed with different settings using a version of `rnx2rtkp` from Swift Navigation's fork of RTKLIB (Swift Navigation, 2022). The truth position of the antenna was established by a long-term survey of the fixed antenna position.

For the *baseline* tests, RTKLIB was configured to use the `fix-and-hold` ambiguity resolution mode (`pos2-armode`). For the *widelane* tests, the ambiguity resolution mode was set to `wide-lane` and `fix-and-hold` mode was enabled by setting `pos2-wlarmode` to `on`.

Table 1 shows the RTK fix rates and Horizontal Position Error (HPE) measured during Test Run 1. Figure 1 shows the Cumulative Distribution Functions (CDF) of the Horizontal Position Error for the baseline (dashed line) and widelane (solid line) configurations. Table 2 and Figure 2 show the equivalent results from Test Run 2.

**Table 1:** Fix rates and Horizontal Position Error during Test Run 1

Test Run	RTK Float (%)	RTK Fixed (%)	1 $\sigma$ HPE (m)	2 $\sigma$ HPE (m)	3 $\sigma$ HPE (m)
Baseline	97.56	2.44	0.315	0.473	0.564
Widelane	15.40	84.60	0.102	0.259	0.354

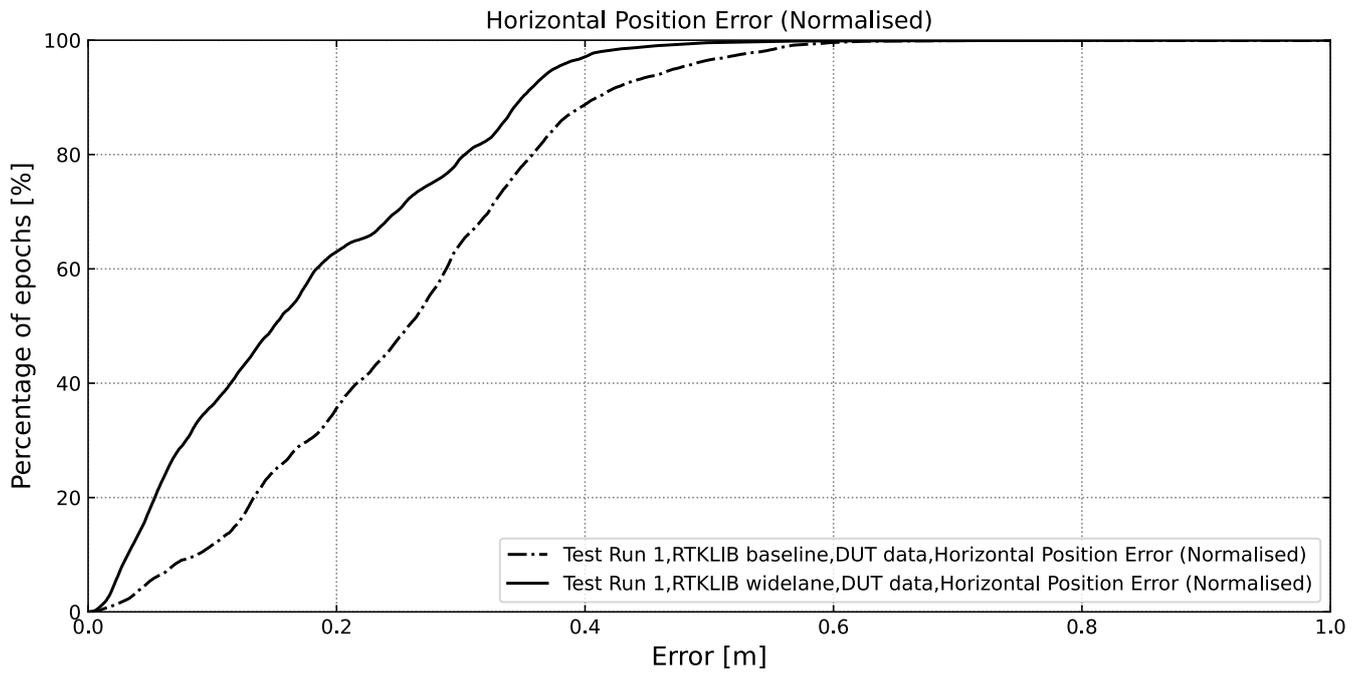
**Table 2:** Fix rates and Horizontal Position Error during Test Run 2

Test Run	RTK Float (%)	RTK Fixed (%)	1 $\sigma$ HPE (m)	2 $\sigma$ HPE (m)	3 $\sigma$ HPE (m)
Baseline	98.39	1.61	0.277	0.424	0.515
Widelane	34.56	65.44	0.238	0.375	0.456

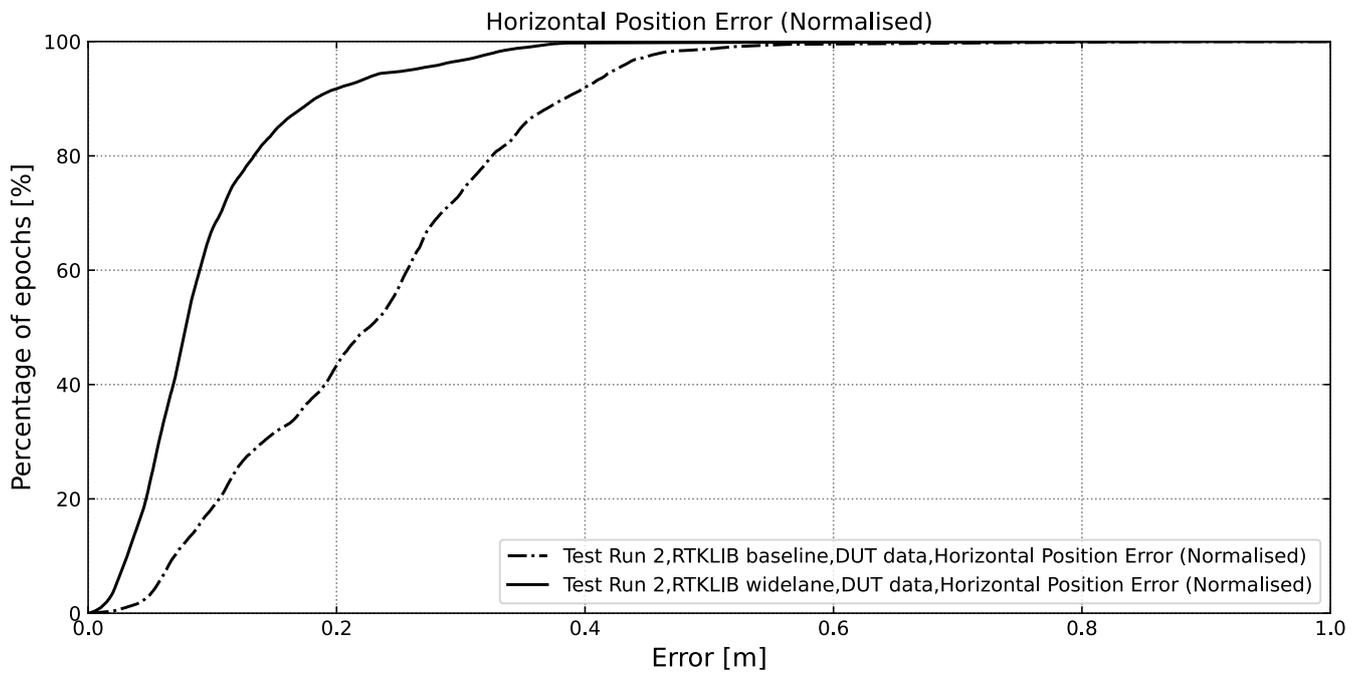
It can be seen from these results that enabling support for wide-lane ambiguity resolution significantly increased the RTK Integer Fix rate, which in turn results in a significant reduction in Horizontal Position Error. Without support for wide-lane ambiguity resolution, RTKLIB produced positions with an average 1 $\sigma$  Horizontal Position Error of 29.6 cm versus an average of 17.0 cm when wide-lane support was enabled. Similar improvements can be seen at the 2 $\sigma$  and 3 $\sigma$  percentiles.

## III. POSITIONING ENGINE COMPARISON

To evaluate the accuracy performance that can be achieved using a low cost receiver with wide-lane corrections in real-world dynamic scenarios, data was collected across a five month period in various environments in the San Francisco Bay Area. This data was then reprocessed with different wide-lane capable positioning engines.



**Figure 1:** Horizontal Position Error for Test Run 1



**Figure 2:** Horizontal Position Error for Test Run 2

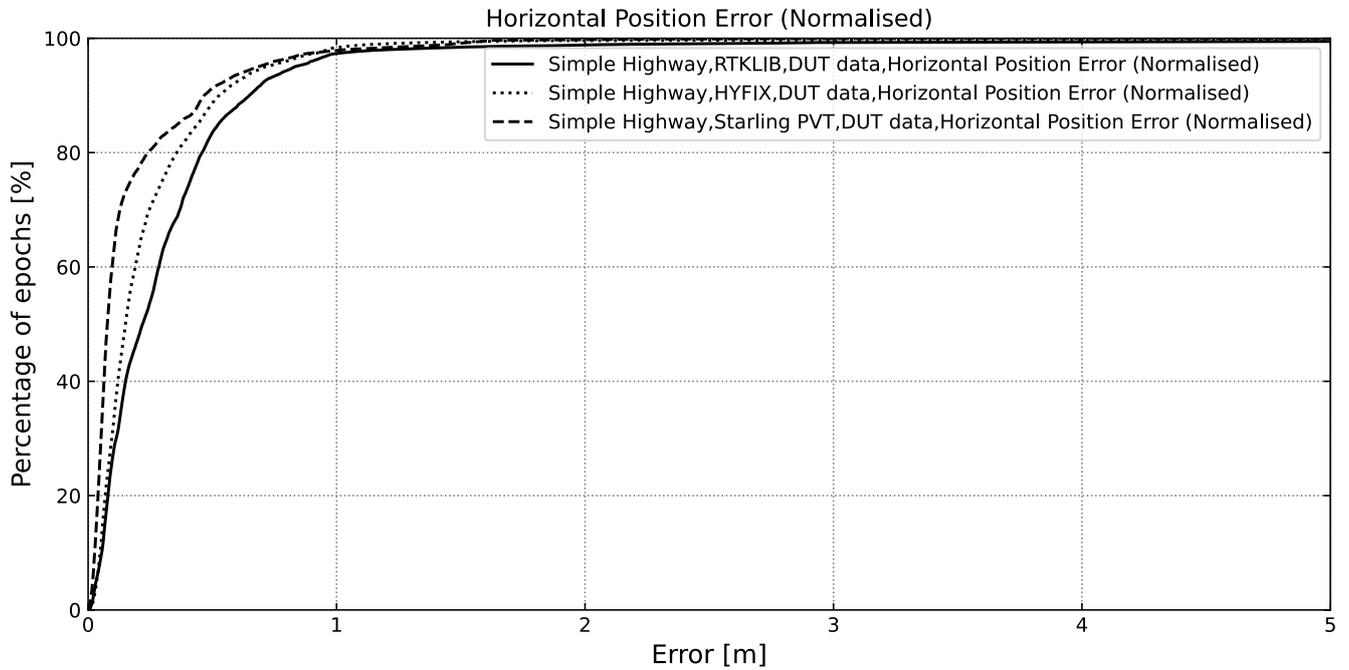


**Table 3:** Drive tests performed on the Simple Highway route

Date	Duration	Distance
2022-06-02	5410 s (1h 30m 10s)	130162.9 m (130.2 km)
2022-06-22	5263 s (1h 27m 43s)	130181.4 m (130.2 km)
2022-07-08	5592 s (1h 33m 12s)	130174.3 m (130.2 km)
2022-07-26	4698 s (1h 18m 18s)	130152.9 m (130.2 km)
Total	20963 s (5h 49m 23s)	520671.5 m (520.671 km)

**Table 4:** Horizontal Position Error from Simple Highway drive tests

Drive	RTKLIB, 1 $\sigma$ (m)	RTKLIB, 2 $\sigma$ (m)	HYFIX, 1 $\sigma$ (m)	HYFIX, 2 $\sigma$ (m)	Starling PVT, 1 $\sigma$ (m)	Starling PVT, 2 $\sigma$ (m)
2022-06-02	0.413	0.766	0.262	0.905	0.114	0.739
2022-06-22	0.406	0.994	0.319	0.589	0.235	0.855
2022-07-08	0.236	0.657	0.174	0.650	0.108	0.557
2022-07-26	0.274	0.775	0.132	0.694	0.055	0.604
Aggregated	0.348	0.830	0.232	0.711	0.118	0.681



**Figure 4:** Aggregated Horizontal Position Error from Simple Highway Drive Tests

It can be seen from the aggregated results that Starling running in GNSS-only mode provided the best horizontal accuracy performance (11.8 cm at  $1\sigma$ ), followed by EmbedCM (23.2 cm at  $1\sigma$ ) and then the wide-lane enabled version of RTKLIB (34.8 cm at  $1\sigma$ ).

b) *Complex Highway*



**Figure 5:** Complex Highway Drive Test Route

Five drive tests were performed on different days along the *Complex Highway* test route using the same test setup as in the previous section. This route (shown in Figure 5) is somewhat more challenging than the Simple Highway scenario since it includes a multi-level covered bridge (San Francisco - Oakland Bay Bridge) as well as short tunnel sections. The durations and driven distance for each drive test are shown in Table 5.

**Table 5:** Drive tests performed on the Complex Highway route

Date	Duration	Distance
2022-05-26	6840 s (1h 54m 0s)	167340.3 m (167.3 km)
2022-06-07	7802 s (2h 10m 2s)	167328.7 m (167.3 km)
2022-06-28	5564 s (1h 32m 44s)	167277.2 m (167.3 km)
2022-07-19	5894 s (1h 38m 14s)	167294.8 m (167.3 km)
2022-08-09	6865 s (1h 54m 25s)	167344.4 m (167.3 km)
Total	32965 s (9h 9m 25s)	836585.4 m (836.585 km)

The Horizontal Position Error values obtained after reprocessing the data from each test drive with different positioning engines are summarised in Table 6. The results obtained from aggregating the results from all five drives are also shown graphically as a CDF in Figure 6.

Again, it can be seen from the aggregated results that Starling running in GNSS-only mode provided the best horizontal accuracy performance (15.8 cm at  $1\sigma$ ), followed by EmbedCM (26.2 cm at  $1\sigma$ ) and then the wide-lane enabled version of RTKLIB (39.9 cm at  $1\sigma$ ). These values are approximately 5 cm worse than the values measured during the Simple Highway drive tests.

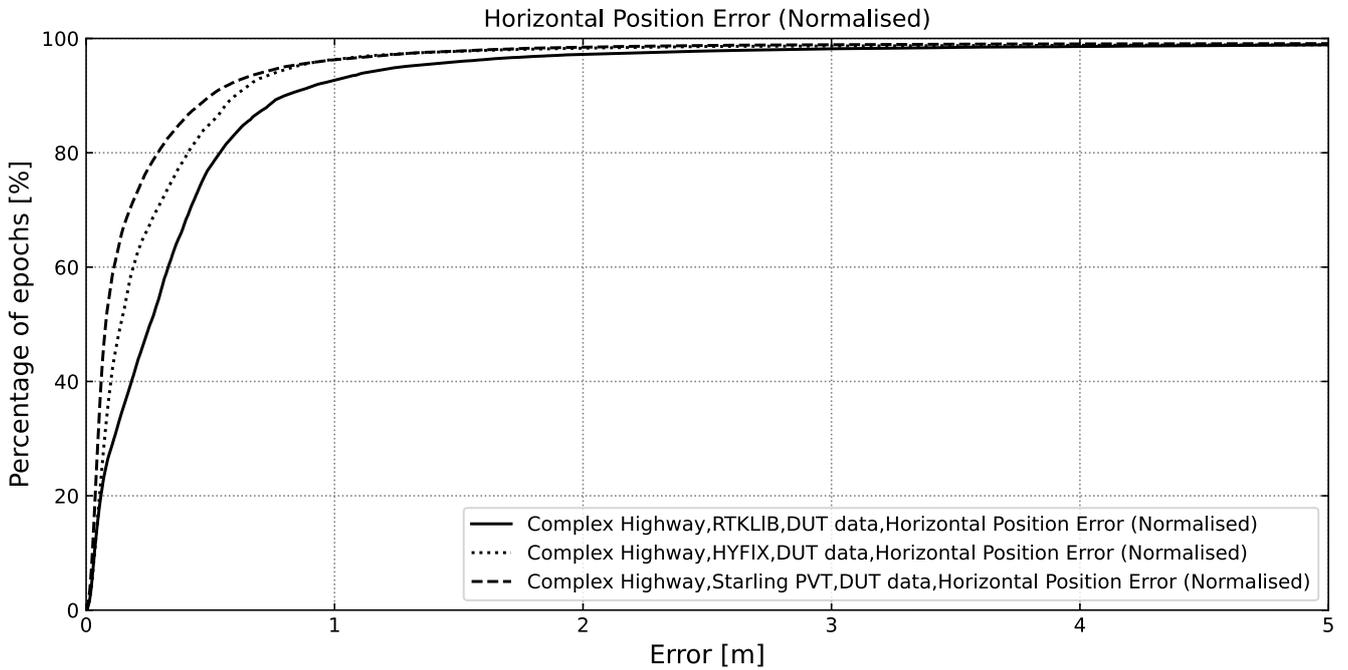
**2. Sensor Fusion**

A number of drive tests were performed in challenging GNSS environments to investigate the limits of the accuracy achievable with low cost GNSS receivers. The logged data was then reprocessed with the following positioning engines:

1. *RTKLIB*: Swift Navigation’s fork of RTKLIB with wide-lane support (Swift Navigation, 2022)
2. *Starling PVT*: Starling Positioning Engine operating in GNSS-only mode (with advanced features such as dead reckoning

**Table 6:** Horizontal Position Error from Complex Highway drive tests

Drive	RTKLIB, 1 $\sigma$ (m)	RTKLIB, 2 $\sigma$ (m)	HYFIX, 1 $\sigma$ (m)	HYFIX, 2 $\sigma$ (m)	Starling PVT, 1 $\sigma$ (m)	Starling PVT, 2 $\sigma$ (m)
2022-05-26	0.447	1.547	0.230	0.943	0.260	0.887
2022-06-07	0.310	0.986	0.286	0.851	0.108	0.813
2022-06-28	0.440	1.534	0.194	0.792	0.144	0.843
2022-07-19	0.483	1.477	0.445	0.798	0.214	0.745
2022-08-09	0.292	0.994	0.167	0.733	0.076	0.694
Aggregated	0.399	1.264	0.262	0.836	0.158	0.793



**Figure 6:** Aggregated Horizontal Position Error from Complex Highway Drive Tests

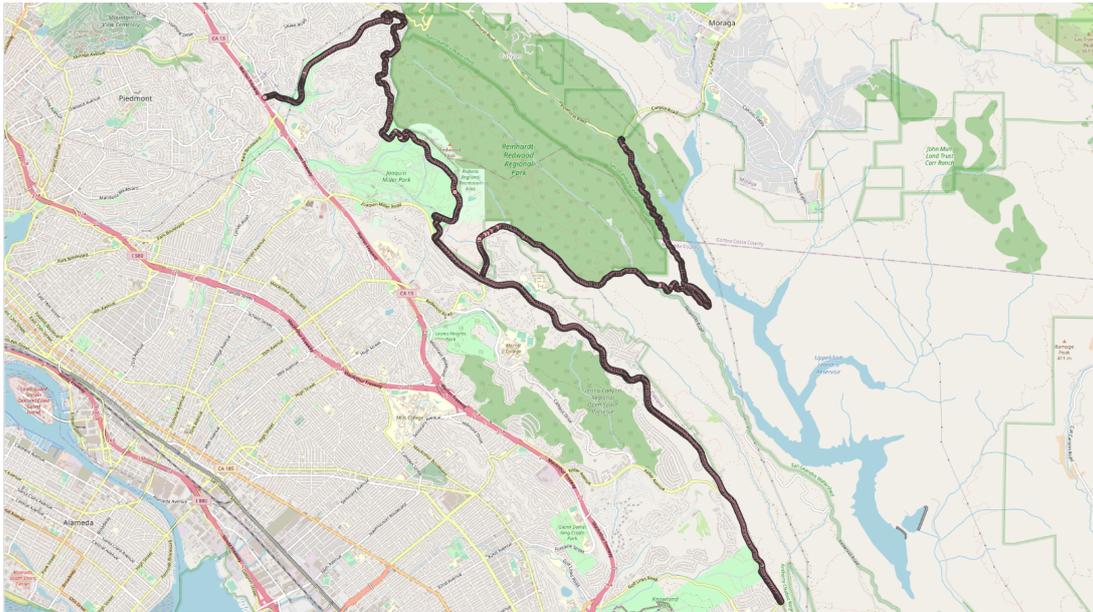
and fusion-aided outlier detection disabled)

3. *Starling*: Starling Positioning Engine with IMU and wheel odometry data

A cheap, automotive-grade MEMS IMU (STMicroelectronics ASM330) was used for the third configuration. The wheel odometry was logged directly from the CAN bus of the test vehicle.

Note that the EmbedCM Positioning Engine was excluded from these tests since it is still under development as of the time of writing.

a) *Foliage*



**Figure 7:** Foliage Drive Test Route

Two drive tests were performed on different days along the *Foliage* test route using the same setup as tests described in the previous sections. This route (shown in Figure 7) traces the boundary of the Reinhardt Redwood Regional Park and includes sections of dense foliage and natural canyons. Dense foliage and tree canopy significantly degrades GNSS tracking performance, particularly for carrier phase measurements. The durations and driven distance for each drive test are shown in Table 7.

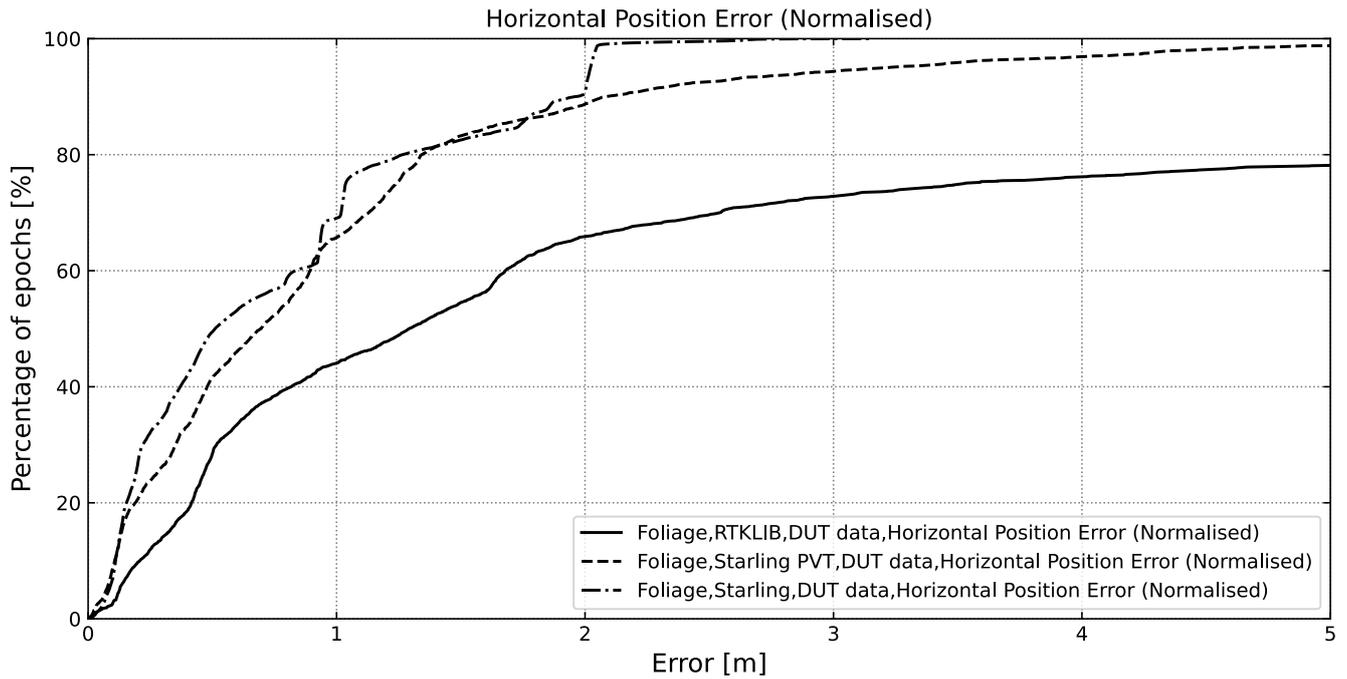
**Table 7:** Drive tests performed on the Foliage route

Date	Duration	Distance
2022-06-09	4836 s (1h 20m 36s)	50972.0 m (51.0 km)
2022-08-04	5200 s (1h 26m 40s)	50955.3 m (51.0 km)
Total	10036 s (2h 47m 16s)	101927.3 m (101.927 km)

The Horizontal Position Error values obtained after reprocessing the data from each test drive with different positioning engines are summarised in Table 8. The results obtained from aggregating the results from both drives are also shown graphically as a CDF in Figure 8.

**Table 8:** Horizontal Position Error from Foliage drive tests

Drive	RTKLIB, 1σ (m)	RTKLIB, 2σ (m)	HYFIX, 1σ (m)	HYFIX, 2σ (m)	Starling PVT, 1σ (m)	Starling PVT, 2σ (m)
2022-06-09	2.735	-	1.270	4.172	1.030	1.865
2022-08-04	1.510	127.738	0.789	1.719	0.461	2.034
Aggregated	2.264	-	1.073	3.217	0.947	2.021



**Figure 8:** Aggregated Horizontal Position Error from Foliage Drive Tests

It can be seen from the aggregated results that Starling operating with sensor fusion input provided the best horizontal accuracy performance (94.7 cm at  $1\sigma$ ), followed by Starling PVT (107.3 cm at  $1\sigma$ ) and then the wide-lane enabled version of RTKLIB (226.4 cm at  $1\sigma$ ). Due to the absence of sensor fusion input, RTKLIB was only able to produce output for 94.27% of the epochs, meaning that no  $2\sigma$  value is available.

*b) Dense Urban*

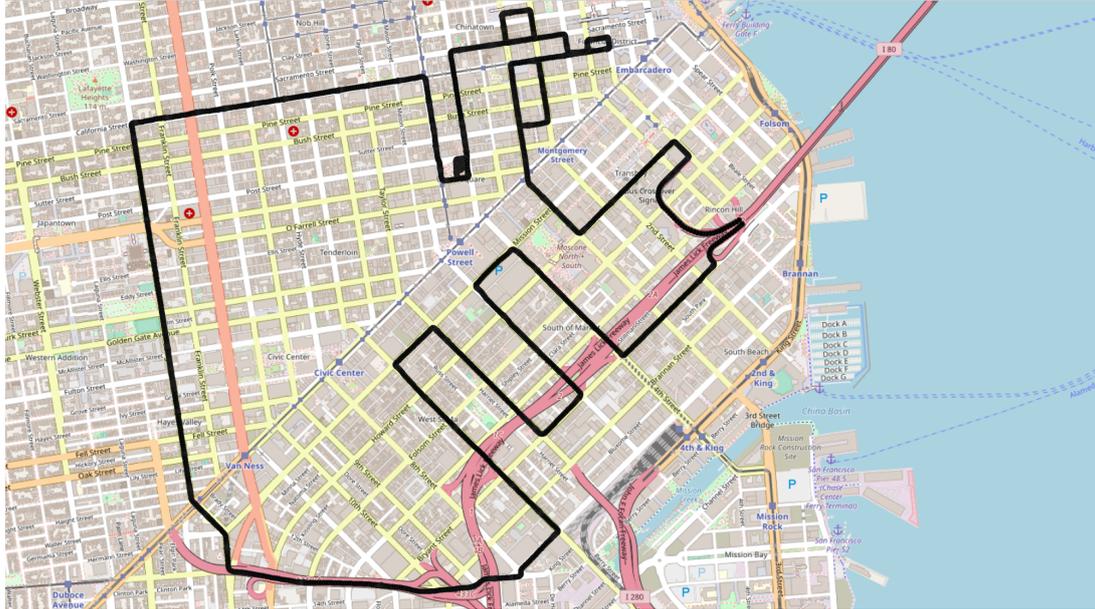
Five drive tests were performed on different days along the *Dense Urban* test route using the same setup as tests described in the previous sections. This route (shown in Figure 9) includes dense urban canyons in the San Francisco financial district as well as an underground parking garage. The durations and driven distance for each drive test are shown in Table 9.

**Table 9:** Drive tests performed on the Dense Urban route

Date	Duration	Distance
2022-06-01	5767 s (1h 36m 7s)	19939.0 m (19.94 km)
2022-06-23	5257 s (1h 27m 37s)	19864.0 m (19.9 km)
2022-07-12	5415 s (1h 30m 15s)	20027.9 m (20.0 km)
2022-07-28	5603 s (1h 33m 23s)	20081.3 m (20.1 km)
2022-08-11	4909 s (1h 21m 49s)	17795.2 m (17.8 km)
Total	26951 s (7h 29m 11s)	97707.4 m (97.707 km)

The Horizontal Position Error values obtained after reprocessing the data from each test drive with different positioning engines are summarised in Table 10. The results obtained from aggregating the results from all five drives are also shown graphically as a CDF in Figure 10.

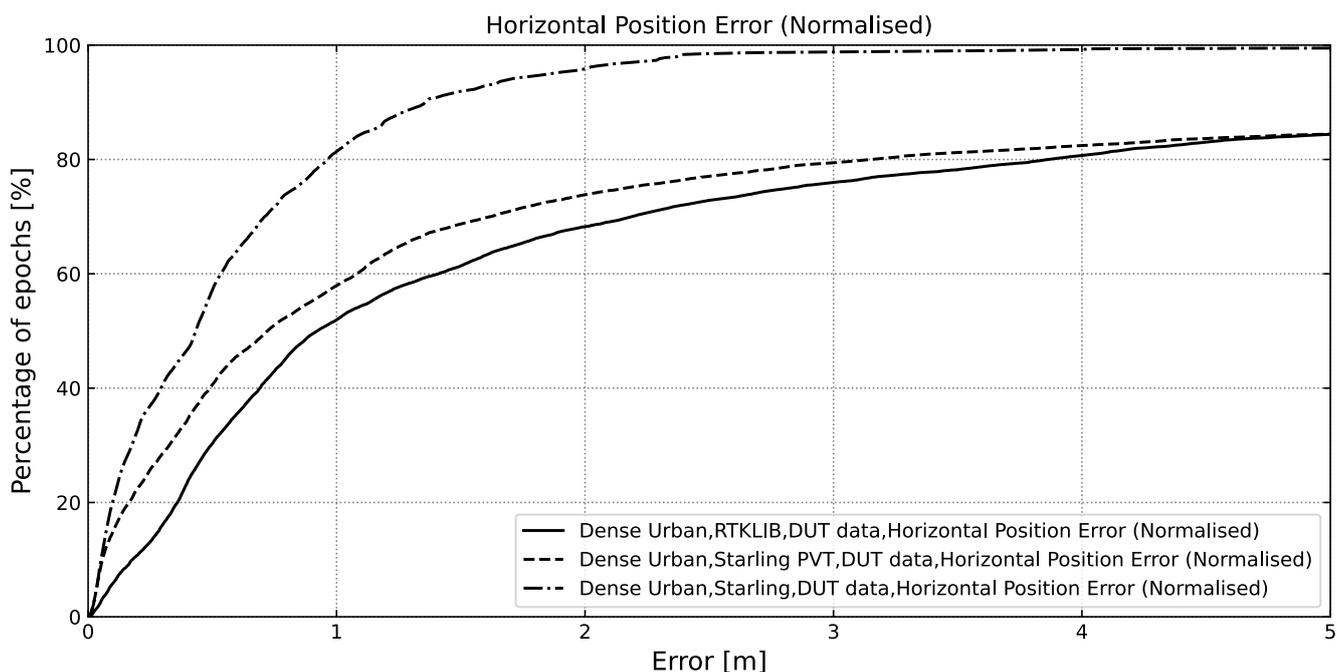
It can be seen from the aggregated results that Starling operating with sensor fusion input provided the best horizontal accuracy performance (67.5 cm at  $1\sigma$ ), followed by Starling PVT (144.3 cm at  $1\sigma$ ) and then the wide-lane enabled version of RTKLIB (197.3 cm at  $1\sigma$ ). Due to the absence of sensor fusion input, Starling PVT was only able to produce output for 92.48% of the epochs, meaning that no  $2\sigma$  value is available.



**Figure 9:** Dense Urban Drive Test Route

**Table 10:** Horizontal Position Error from Dense Urban drive tests

Drive	RTKLIB, 1 $\sigma$ (m)	RTKLIB, 2 $\sigma$ (m)	HYFIX, 1 $\sigma$ (m)	HYFIX, 2 $\sigma$ (m)	Starling PVT, 1 $\sigma$ (m)	Starling PVT, 2 $\sigma$ (m)
2022-06-01	1.190	9.294	1.023	-	0.566	1.371
2022-06-23	2.648	-	1.653	-	0.918	2.390
2022-07-12	2.379	14.060	2.491	-	0.873	2.019
2022-07-28	2.492	5324.785	1.711	-	0.601	1.625
2022-08-11	1.491	10.108	0.721	-	0.237	1.094
Aggregated	1.973	70.093	1.443	-	0.675	1.860



**Figure 10:** Aggregated Horizontal Position Error from Dense Urban drive tests

#### IV. CONCLUSIONS

The results outlined in this paper demonstrate the positioning accuracy which can be achieved through the utilisation of a large scale, continental-wide corrections network with a low cost, dual-band chipset. All of the positioning engines tested were capable of achieving decimetre-level accuracy at  $1\sigma$  using GNSS-only data in static and dynamic highway conditions. For more challenging GNSS reception environments, this level of accuracy cannot be achieved by GNSS alone, but is demonstrably feasible when using a modern positioning engine with sensor fusion capabilities. There are many real-world applications which can benefit from the ability to provide this level of performance with such low cost hardware.

#### V. FUTURE WORK

As a follow-up to the results demonstrated in this paper, Swift Navigation will continue to work with positioning engine vendors to integrate support for wide-lane corrections. As these positioning engines become integrated into low cost GNSS receivers, this will allow end users to benefit from the advantages of continental-scale, wide-area corrections.

Additional work may also be needed to optimise the implementation of the `resamb_WL()` function in Swift Navigation’s fork of RTKLIB. The upstream implementation of `resamb_LAMBDA()` uses an optimisation whereby the  $G$  matrix referred to Section II.1 (named  $D$  in the source code comments) is not completely constructed—given the knowledge that each row can only result in the subtraction of exactly one value from another, an array is constructed where each pair of elements represents the indices of the values to be subtracted. This reduces memory usage as well as computational requirements, which are both highly desirable traits to have in an implementation intended to be embedded in low cost GNSS receivers with limited compute resources. The current implementation of `resamb_WL()` was written with the objective of being easily understood by positioning engine vendors who wish to add wide-lane support rather than being optimised for performance. However this does not preclude an alternate optimised code path from being added in the future.

#### ACKNOWLEDGEMENTS

The authors would like to acknowledge the following people for their contributions to this paper: YuDan Yi (HYFIX) for reprocessing the data with the EmbedCM Positioning Engine, Ivan Smolyakov (Swift Navigation) for his assistance with the Starling Positioning Engine and Alex Parkins for his technical input regarding wide-lane corrections.

## REFERENCES

- Cocard, M., Bourgon, S., Kamali, O., and Collins, P. (2008). A systematic investigation of optimal carrier-phase combinations for modernized triple-frequency GPS. *Journal of Geodesy*, 82(9):555–564.
- Collins, J. P. (1999). An Overview of GPS Inter-frequency Carrier Phase Combinations. Technical report, Geodetic Survey Division, University of New Brunswick, New Brunswick, Canada.
- Hamza, V., Stopar, B., Ambrožič, T., Turk, G., and Sterle, O. (2020). Testing Multi-Frequency Low-Cost GNSS Receivers for Geodetic Monitoring Purposes. *Sensors*, 20(4375).
- Jackson, J., Saborio, R., Ghazanfar, S. A., Gebre-Egziabher, D., and Davis, B. (2018). Evaluation of Low-Cost, Centimeter-Level Accuracy OEM GNSS Receivers. Technical Report MnDOT 2018-10, Minnesota Department of Transportation.
- Janos, D. and Kuras, P. (2021). Evaluation of Low-Cost GNSS Receiver under Demanding Conditions in RTK Network Mode. *Sensors (Basel, Switzerland)*, 21.
- Leclère, J., Landry, R., and Botteron, C. (2018). Comparison of L1 and L5 Bands GNSS Signals Acquisition. *Sensors (Basel, Switzerland)*, 18.
- Naciri, N., Hauschild, A., and Bisnath, S. (2021). Exploring Signals on L5/E5a/B2a for Dual-Frequency GNSS Precise Point Positioning. *Sensors*, 21(2046).
- National Coordination Office for Space-Based Positioning, Navigation, and Timing (2022). GPS.gov: Space Segment. <https://www.gps.gov/systems/gps/space/>. Accessed: 2022-07-22.
- NovAtel Inc. (2020). *Waypoint Software 8.90 User Manual*. NovAtel Inc., Calgary, Canada.
- Subirana, J. S., Zornoza, J. J., and Hernández-Pajares, M. (2011). Detector based in code and carrier phase data: The Melbourne-Wübbena combination. [https://gssc.esa.int/navipedia/index.php/Detector\\_based\\_in\\_code\\_and\\_carrier\\_phase\\_data:\\_The\\_Melbourne-W%C3%BCbena\\_combination](https://gssc.esa.int/navipedia/index.php/Detector_based_in_code_and_carrier_phase_data:_The_Melbourne-W%C3%BCbena_combination). Accessed: 2022-03-03.
- Swift Navigation (2021). Swift Binary Protocol, Protocol Specification 4.1.1. <https://support.swiftnav.com/support/solutions/articles/44001850782-swift-binary-protocol>. Accessed: 2022-04-12.
- Swift Navigation (2022). swift-nav/RTKLIB. <https://github.com/swift-nav/RTKLIB>. Accessed: 2022-08-01.
- Takasu, T. (2013). *RTKLIB ver. 2.4.2 Manual*. Tokyo University of Marine Science and Technology, Tokyo, Japan.
- Takasu, T. (2020). RTKLIB: An Open Source Program Package for GNSS Positioning. <https://www.rtklib.com/>. Accessed: 2022-02-02.
- Urquhart, L. (2009). An Analysis of Multi-frequency Carrier Phase Linear Combinations for GNSS. Technical report, Department of Geodesy and Geomatics Engineering, University of New Brunswick, New Brunswick, Canada.