Multi-pass phase-differential GNSS ambiguity fixing for post-processing applications

D. Becker[1], E. v. Hinüber[1]

[1] iMAR Navigation GmbH
Im Reihersbruch 3
66386 St. Ingbert
GERMANY

Inertial Sensors and Systems 2023

Braunschweig, Germany

979-8-3503-4724-1/23/$31.00 ©2023 IEEE

Abstract

Traditional parameter estimation like Kalman filtering is mostly based on the assumption that all states/parameters and observations are defined in *continuous* multi-dimensional space. GNSS ambiguity fixing however is breaking this assumption: the previously float ambiguity estimates are fixed to integer values, for example using the integer-least-squares method. Furthermore, the error of an ambiguity fix is essentially binary: the fix is correct, or not; it is usually of little interest how incorrect a bad fix is. Similarly, outliers of raw GNSS observations are commonly flagged, or not flagged, based on some predefined threshold – also breaking the continuity assumption. The traditional Extended Kalman Filter (EKF) framework in combination with RTS smoothing (also called Kalman smoothing) may therefore fail to produce optimal results, even in the absence of (significant) linearization errors. On the other hand, once a correct fix is successfully established after some convergence time, this is obviously information that should be made available to epochs in the past: as long as a satellite is continuously tracked by the receiver without any loss-of-lock or cycle slip, the ambiguity must always be constant by definition. Therefore, a newly established ambiguity fix not only applies to the *upcoming* epochs, but also to those epochs of the convergence phase *before* the fix.

In this paper, we present results from a GNSS test data evaluation using iMAR's post-processing software suite *iPosCal-SURV*. The test data set consists of a large variety of kinematic scenarios, including very challenging urban canyons. A Kalman Filter is embedded into an iterative process that walks over the entire multi-GNSS multi-base data set iteratively, using multiple passes. An ambiguity memory is used in order to make double-differenced ambiguity fix information available to the upcoming passes. Consider for example three passes: forward-reverse-forward. The second (reverse) pass can use the entire fixing information of the first pass, which will in general allow *additional* fixes during the second pass. The third pass can then use all the accumulated fixing information of the first two passes, enabling *even more* successful ambiguity fixes. Real-world examples indicate, that the fixing rate may even benefit from using *six* or more passes, in particular under very challenging GNSS conditions. Results are shown for both GNSS-only processing, and the tightly-coupled integration of GNSS and a navigation-grade IMU.

1. Introduction

For real-time GNSS applications including time-correlated parameters, it is a common approach to use a Kalman Filter (KF) technique [1], or some of its variants. The KF will implicitly use all information from the past for the parameter estimation of the current epoch. It cannot make use of information that only becomes available in the future.

When *post-processing* GNSS data (rather than processing in real time), it is essential to make use of the *entire* data set's available information for the estimation of *each* epoch's parameters (in particular: the position estimates). One way of doing so is to perform a one-step least-squares estimation comprising *all* measurements and system states at *all* time instances. This is however, in general, difficult to achieve, because a data set may consist of millions of measurements and tens of thousands of states (“parameters”), leading to huge matrices which are difficult to handle with a state-of-the-art workstation computer.

It turns out, that also in the case of *post-processing*, Kalman Filtering can be very helpful, mainly due to its very low computation time and memory requirements. In this case, intermediate results of the regular Kalman Filtering phase (forward in time) will be stored in memory. As soon as the final epoch has been processed by the KF, a so-called Rauch-Tung-Striebel-Smoother (RTS, [2]), also known as Kalman Smoother, can be applied, going over the stored intermediate KF results *backward* in time. This combination of Kalman Filtering and RTS-smoothing generates provenly *optimal* estimates for all epochs *when dealing with linear systems*. The RTS-smoother is an optimal smoother in this case. However, there are two underlying assumptions of the KF and the RTS smoother which *do not apply* to raw GNSS processing including phase ambiguity fixing: (1) the system is commonly *non-linear* (since the measurements mainly deal with distances), and (2) the parameter estimation is designed for *continuous* (real-valued) parameters, but fixed phase ambiguities are *integer* numbers [7].

The well-known Extended Kalman Filter (EKF) is designed for non-linear systems, applying a first-order Taylor approximation to the non-linear system model and measurement models. For many applications in positioning, this is a viable solution, as long as the difference between predicted state and true state does not become too large. For the remainder of this paper, we will not further discuss non-linearity errors, because it is safe to assume here that the effects from the non-linearities are insignificant (keeping the predicted position’s error usually below 20 meters).

When using raw GNSS *phase* observations in the context of Extended Kalman Filtering, it is essential to let the EKF estimate the phase *ambiguity* for a particular satellite and transmission channel as a random constant, while the satellite is being tracked without interruption. The constraint of a *constant* phase ambiguity can greatly improve the positioning accuracy. Individual ambiguity states are added to the EKF state vector for this.

Phase-differential GNSS positioning based on double-differenced code- and phase observations is a well-known and well-established GNSS positioning strategy for high precision applications [8]. The well-known Real-Time Kinematic (RTK) processing is one famous example, allowing phase ambiguity fixing in real-time applications. A double difference is the among-station-difference (between a rover and a master station) of two among-satellite differences (between a satellite and a well-defined pivot satellite).

Therefore, a double-difference consists of four original raw GNSS observations, and this holds both for code and phase observations [8].

The phase ambiguity is, by its nature, an *integer* number, contradicting the EKF’s and RTS’ assumption of continuous parameters. Initially however, phase ambiguities are indeed modelled as being *continuous* quantities (commonly called *float* phase ambiguities). With the underlying assumption of constancy (zero system noise), the float ambiguity estimates will converge over time in the course of the EKF processing and thereby already significantly support positioning. Usually an EKF-based post-processing software will continuously try to find a valid phase ambiguity *integer* solution (also called a “fix” or a “fixed solution”) after each EKF measurement update, either for the whole set of satellites, or for individual or subsets of satellites (depending on the fixing strategy). As long as a particular satellite is being observed without interruption or cycle-slips, a once found integer solution persists over time, and the ambiguity estimate as part of the EKF system state can be deactivated for that particular ambiguity. The phase observations become high-precision range observations in this case (at the centimeter level!), which is of course the underlying fundamental idea of fixing the phase ambiguity in the first place [7].

Real-time applications can only use the additional information (coming from the integer ambiguity fix) for the current and the upcoming epochs. In post-processing, however, one can further exploit this information, because the (constant) ambiguity must have had the same value *already before the integer fix was acquired*, for as long as the satellite has been observed without interruption. The post-processing software design has to make sure, that such knowledge is somehow being transferred to these epochs in the past. The following section will discuss established ways of exploiting the constancy of fixed ambiguities.

2. Exploiting Ambiguity Constancy - State of the art

Definition. We borrow the term *Connected Component* from graph theory here. For our purposes in this paper, it will denote the time interval between any two complete GNSS outages, with zero (valid) GNSS phase observations for at least one epoch. Consider Figure 1 for the clarification of this term.

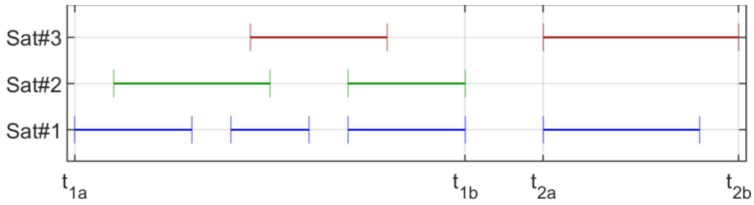

Figure 1. Example satellite visibility plot with two connected components: the first going from t1a to t_{1b}, the second from t_{2a} to t_{2b}. There are three GNSS outages in this example, where no (valid) phase observation were recorded by the receiver: (1) before t_{1a}; (2) between t_{1b} and t_{2a}; (3) after t_{2b}.

When dealing with GNSS phase ambiguities, one seeks to exploit the fact, that the (unknown) ambiguity is constant over time, as long as a satellite is being tracked without interruption or cycle slips. In fact, this constancy is the basis of the enormous potential of using GNSS phase observations in the first place. For time-continuous state estimation (like an EKF), the ambiguity is therefore modelled as a *random constant*.

Figure 2 shows the same situation as in Figure 1, but highlights different time instances. Assume that an EKF works forward in time, and a valid integer ambiguity fix can be found at t_{fix1} for satellites #1 and #2. Because we already know, that the ambiguity is constant while a satellite is tracked continuously by the receiver, we not only know its integer ambiguity from now on (between t_{fix1} and t_2), but for the *whole* interval from t_1 to t_2! In post-processing, it would obviously be a waste of information *not* exploiting this property, and it can be assumed that most post-processing software packages indeed use this constancy property somehow.

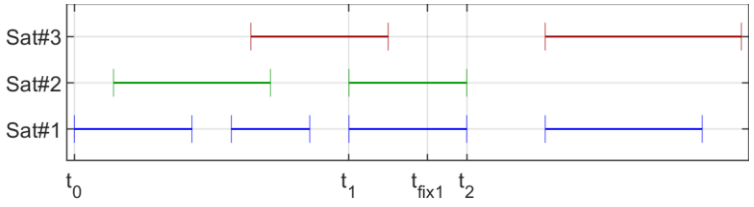

Figure 2. Example satellite visibility plot explaining the general idea of the ambiguity fix constancy.

2.1. Forward-Backward-Combination of Position Estimates

A straight-forward method that does *not* exploit the constancy of integer ambiguities is the simple weighted combination of two EKF runs, one in forward, and one in backward direction. This approach is obviously sub-optimal: consider a satellite that is being tracked for 10 seconds, but it is not before the last (10th) epoch, that the forward run manages to establish an ambiguity fix, and it is not before the first epoch, that the backward run manages to establish the same fix. The naïve forward-backward-combination will not use the knowledge of the ambiguity fix for the eight central epochs!

2.2. Forward-Backward-Combination of Float Ambiguity Estimates

Instead of performing a forward-backward combination in the position domain, one can perform such a combination in the (float) phase ambiguity domain, as presented in [9]. This essentially leads to the following algorithm:

1. Perform two EKF runs (one in forward, the other in backward direction in time), and do *not* fix the phase ambiguities.

2. Combine the float phase ambiguity estimates of the two runs. Thereby, all these estimates can (in general) benefit from *all* available information of the data set.

3. With these combined float phase ambiguity estimates, perform the integer ambiguity fixing.

4. Use the acquired integer fixes for another EKF-run, where all fixed ambiguities are not estimated as float anymore. An RTS-smoother should still be applied afterwards, if some phase ambiguities could not be fixed.

2.3. Generic RTS smoothing with pseudo-integer-observations

Again, consider the example of Figure 2. Another straight-forward approach is the generic implementation of EKF filtering in combination with regular RTS smoothing. Once the ambiguity fix was established at t_{fix1}, *real-valued pseudo-observations* are introduced to the EKF reflecting the new *integer* ambiguity fix. Since in the regular formulation of EKF and RTS, a measurement cannot have zero noise, this pseudo-observation of the integer ambiguity can be introduced with an artificially low noise level “close to zero”, for example with 0.0001 cycles, which is consistent with < 0.02 mm for common GNSS frequencies - and therefore insignificant compared to other errors (e.g. atmospheric errors). From t_{fix1} onwards, the EKF (in its native implementation) can use this fix information to future epochs, and a native RTS smoother will then “take” this information back (reversely in time) all the way back to t_0, instantly producing optimal estimates for *all* epochs between t_0 and t_2.

2.4. Ambiguity Fixing Chain Reactions

The aforementioned strategies indeed help to improve the amount of information that is available for the parameter estimation in post-processing, but there is another aspect, which all these strategies do not exploit optimally: Consider satellite #3 in Figure 2 and assume that this satellite could not be fixed while it was visible. When fixing satellites #1 and #2 at t_{fix1}, satellite #3 has already disappeared and the knowledge of the other two satellite's ambiguity fix *cannot be used for the (potential) fixing of satellite #3*. But let's assume, that satellite #3 could indeed be fixed with this additional knowledge. This is what is denoted as "phase ambiguity fixing chain reaction" for the remainder of this paper, or simply "chain reaction". Now assuming we have found a fix for satellite #3 at (or just after) t_1, this knowledge itself can be used for previous epochs, and new ambiguity fixes of satellites #1 and #2 may become possible before t_1!

We will later in this paper see, that such "chain reactions" can indeed be found in real-world data sets, in particular in very challenging GNSS environments like urban canyons.

None of the aforementioned strategies can optimally exploit the available information in the case of such chain reactions, because *establishing a new fix is not rigorously treated as additional information*, which then also should be made available for the parameter estimation (and more ambiguity fixing) of all other epochs of data set. For example, the generic EKF/RTS approach of Sect. 2.3., the RTS smoother can take information from t_2 back to the past (to t_0) in general, it is *not* generically designed to solve discrete optimization problems, like integer ambiguity fixing, along the way. While in the EKF/RTS approach, an ambiguity fix is used for (directly) improving the positioning results of all epochs, it is *not* used for establishing additional ambiguity fixes.

The aforementioned "chain reactions" ask for new strategies, which *iteratively* allow the usage of additional ambiguity fix information, also for establishing new fixes in other epochs and/or for other satellites.

Note: For GNSS applications *not* performing phase ambiguity fixing, for example float-PPP, the assumption of continuous parameters is fulfilled, and the standard combination of EKF and RTS will indeed produce the optimal solution (apart from linearization errors).

3. EKF Rewinding

The idea of rewinding is to store the entire state of the EKF (including the stochastic estimates, etc.) for each epoch into memory such that the algorithm is able to restore a previous system state.

3.1. Simple Rewinding

Consider Figure 3, which extends the previous example.

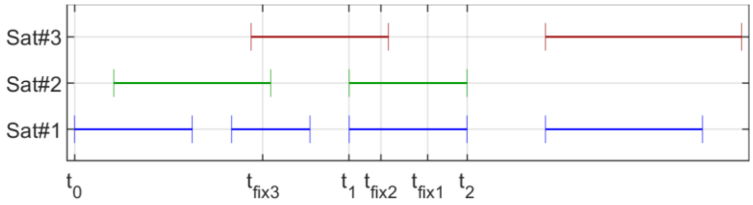

Figure 3. Example satellite visibility plot explaining the concept and limitations of rewinding.

As soon as the fix succeeds at t_{fix1}, the EKF will be rewound to t_1 (by fetching the respective state from memory) and use the fix information *as if* it had already been available at t_1, and then continue as usual (again forward in time). Previously stored states can be overwritten. Obviously, the float estimation and fix determination for satellites #1 and #2 is not required anymore, until the tracking of these satellites is interrupted or a cycle-slip occurs (after t_2).

Note, that this (introductory) method still does not optimally exploit the ambiguity fixing “chain reactions”.

3.2. Iterated Rewinding

We continue with the example of the previous paragraphs. The new fix information gained at t_{fix1} can be helpful for the fixing of *other* satellites from t_1 on. In Figure 3, let’s assume that with the knowledge of the phase ambiguities of satellites #1 and #2, we can now also find a valid integer ambiguity *for satellite #3* at t_{fix2} – the start of a chain reaction. Now assume, that the knowledge of the ambiguity of satellite #3 itself helps to fix the satellites #1 and #2 at t_{fix3}, and so forth. Thus, it can be useful to iteratively continue with this rewind-logic until no further ambiguity fixes can be found. The algorithm is *explicitly* “following” these chain reactions.

Note, that for kinematic *GNSS-only* positioning, the (potentially multiple) rewind steps are always happening *within* a connected component, or in other words, one can process each connected component one by one, as there will not be any relation across them regarding the phase ambiguity estimates. We will later see (Sect. 2.3), that this is not the case when an IMU is present (*tightly coupling*).

Discussion: While the approach may appear simple at first, there are more practical aspects which lead to a much higher implementation complexity. Consider for example an (in practice inevitable) outlier detection routine, that marks an observation as an outlier *after* one or more rewind steps have taken place. This happens in practice, because outliers are easier to detect if the position is better conditioned from the additional ambiguity fixes. If this observation, which was *later* found to be an outlier, has significantly contributed to the original fix itself (that has initiated the rewinds) – should one then undo the rewinds and/or the fixes? Or at least partially? Also, the iterative rewinding strategy as explained above requires a careful storing- and restoring-strategy for the integer fixes. Consider for example satellite #1 and #2: The integer phase ambiguity differs in general at t_{fix1} and t_{fix3}, and so when rewinding further towards t_0, the algorithm has to keep track of all the different integer ambiguities that were found along the way.

3.3. Combination of Rewinding and RTS-smoothing

After performing an EKF run with rewind strategy, it is still useful in general to append an RTS run as usual. For GNSS-only positioning, without any further assumptions regarding the dynamics of a vehicle, the RTS run will reduce the estimates' uncertainty for such epochs, where at least some (or all) satellites could *not* be fixed, i.e. a generic RTS implementation with exploit the constancy of the float ambiguities. On the other hand, if *all* phase ambiguities could be fixed in a particular data set (based on EKF rewinding), the RTS run can be omitted, as all position estimates are effectively just weighted range observations, based on the established ambiguity fixes. The ambiguity states become irrelevant then, and the commonly large system noise for the position state will essentially transform the EKF into a simple *epoch-wise weighted-least-squares* positioning estimator.

3.4. Rewinding in Tightly-Coupled Positioning

The general ideas of exploiting the phase ambiguity constancy, as explained in the previous sections, also holds for tightly-coupled navigation. In fact, already by theoretic considerations, it becomes clear that the relevance of exploiting the constancy intensifies with the quality of the used IMU. Consider the two extreme (theoretic) cases:

1. Tightly-coupled navigation using an IMU with infinitely bad performance. This is obviously equivalent to GNSS-only positioning, as the IMU does not contribute any information.

2. Tightly-coupled navigation using an IMU with *infinitely good* performance.

Let's have a closer look at the second case. The perfect IMU transfers any knowledge regarding the current position into the future without any loss in precision, being consistent with GNSS-only navigation with *zero* system noise for the position states. In terms of observability, this is therefore equal to *static* GNSS-only positioning, drastically increasing the overall measurement redundancy: Consider a case with N epochs and A phase ambiguities. Kinematic positioning requires the estimation of 3*N+A parameters, whereas static positioning only requires 3+A parameters: the (static) position and the ambiguities. In other words: The tight integration of GNSS and a perfect IMU allows (in theory) *kinematic* navigation at the accuracy and precision of *static* GNSS positioning.

Still assuming a perfect IMU, it then becomes obvious, that the effect of exploiting the constancy property intensifies compared to a GNSS-only data set: position information can be transferred *across* several connected components, "bridging" the GNSS outages. Assume the following example with a GNSS outage as shown in Figure 4.

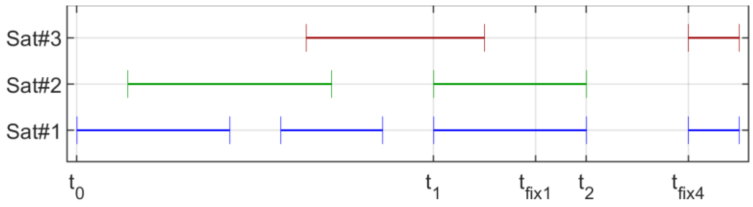

Figure 4. Example satellite visibility plot explaining the additional merit of tightly-coupling for the integer ambiguity resolution in the presence of GNSS outages.

As before, assume the regular EKF run (forward in time) manages to find a fix for satellites #1 and #2 at t_{fix1} (and perhaps it rewinds then to t_1 as explained before). Therefore, we can assume that the position is well-known at t_2 (could be at the sub-centimetre level for real-world scenarios). The perfect IMU will transfer this knowledge over to the next connected component. In the very first epoch of the next connected component, at t_{fix4}, the two visible satellites #1 and #3 can usually *instantly* be fixed (or after very few epochs), relying on the

"IMU-transferred" knowledge regarding the position. This is in particular helpful in situations, where otherwise the limited convergence of the float ambiguities estimates would not allow an integer fix resolution, as is the case for very short connected component starting at t_{fix4}. This paper will demonstrate real-world examples for such situations (using both a commercial-grade and a navigation-grade IMU).

From the above theoretic considerations, it can be seen, how the *entire* data set transforms into a *single* "connected tightly-component", because in general, finding a fix at the very first epoch can affect the very last epoch (and vice versa) by means of the IMU-based information transfer over time, even if full GNSS outages are in between. This then leads to the concept of multi-pass ambiguity resolution, which is the main contribution of this paper.

Note: Real IMUs obviously are neither useless nor perfect, and the degree to which it will help the ambiguity fixing process is related to the grade of the IMU, reflecting how well the IMU can "transfer" position information over time.

4. Multi-Pass ambiguity resolution

With this paper we present a new strategy of exploiting the aforementioned constancy of the phase ambiguity. This new method has been implemented in iMAR's iPosCal-SURV post-processing software package. Phase ambiguity fixing chain reactions are exploited maximally, by making the information of any new ambiguity fix available to both the (continuous) parameter estimation, and the search for *additional* ambiguity fixes

4.1. General idea

It has been shown in the previous section, that for tightly-coupled navigation, the data set has to be seen as a whole (rather than looking at individual connected components). Also, implementing an iterated rewind strategy can become highly complex, as was indicated before. We will see, that the presented method has a moderate implementation complexity while keeping the maximum generality in terms of real-world scenarios.

Multi-pass ambiguity resolution is the idea of iteratively passing over the *entire* data set *multiple* times with an EKF, until no additional ambiguity fixes can be established. Established ambiguity fixes will be stored along the way and will be reused during the next pass. At the end of each pass, the EKF will "turn around" in time, without the need of reinitializing the EKF states. *Examples*: for a 3-pass run starting in forward direction, the passes are forward-backward-forward. For a 5-pass run starting in backward direction (at the very last epoch), the passes are backward-forward-backward-forward-backward.

The implementation complexity of this approach is reduced, because the storage and bookkeeping of the EKF state along the way (including all absolute state estimates etc.) is not required anymore. Also, the aforementioned “chain reactions” are fully covered by this multi-pass approach, while the algorithm design does not have to deal with these explicitly: Once a new ambiguity fix is established, the EKF simply continues with its pass without any rewinding, as it would be the case for a straight-forward real-time EKF implementation operating forward in time.

Only few modifications are required: turning around at the ends, and performing a correct bookkeeping of the found integer ambiguity values after a successful fix was established (cf. Sect. 4.3).

4.2. Computational burden

The method obviously incurs a higher computational burden, because the entire data set will be processed multiple times, no matter if an additional ambiguity fix *significantly* affects most parts of the data set, or not. It can be seen as a brute-force method, which guarantees the maximal ambiguity fixing potential at a low software implementation complexity, but at the same time accepting computational inefficiency to some extent.

In order to put this computational burden into perspective, an example is given here taken from a processing run with iMAR’s iPosCal-SURV software package:

- GNSS-only processing of GPS and Galileo (both 2-frequency double-differenced)
- using a single base station
- duration: 173 minutes; processed at 1 Hz (6535 epochs)
- used hardware: a single core of an Intel i9-12950HX CPU
- *total* computation time using *one* pass: 5.01 seconds.
- *total* computation time using *five* passes (=865 min): 17.30 seconds.

Note: The total computation time does *not* increase by a factor of five in this example, because of the additional overhead of reading in RINEX files, performing RTS smoothing, logging and outputting. The four additional passes apparently require 12.3 seconds more computation time, so the net runtime for a single EKF pass is approximately 3.1 seconds.

So, it can be seen, that with an otherwise efficient software implementation, even five passes take little time on a state-of-the-art workstation PC. Even with *five* passes, the overall computation speed was 600 times higher compared to real time (in other words: it processed 10 minutes of raw data within one second).

4.3. Ambiguity Memory

Similar to the considerations of Section 3.2., the multi-pass ambiguity resolution requires the storage of successfully established integer ambiguity fixes. Since this information is iteratively stored for the *following* EKF pass only, it is sufficient to store a satellite's phase ambiguity fix (*if* an integer fix was established) as soon as it *disappears* (not being tracked anymore). For the next EKF pass in reverse direction over time, it is this epoch where the particular satellite appears again, and it can immediately make use of the stored integer phase ambiguity associated to this epoch in the ambiguity memory. For the current EKF pass, the float ambiguity estimation is inactive for that particular satellite.

4.4. Outlier handling

Any GNSS processing software will usually implement some sort of outlier detection scheme. This subsection does *not* deal with the outlier detection itself, but with the question, who to deal with (already detected) outliers of various types in the context of multi-pass phase ambiguity fixing.

Assume, that an ambiguity fix was already established (before in the current EKF, and/or in a previous EKF pass). Now let's assume, that the outlier detection scheme flags a fixed phase observation as an outlier. This can be due to the following reasons:

1. It is a "regular" outlier: The phase observation of the current epoch has a large error, coming from an undetected cycle slip, from multi-path, or other sources of error.

2. The ambiguity fix is *incorrect*, but still had passed the various validation criteria when the fix was established.

[Note, that the combination of both reasons is possible as well.] In general, we cannot distinguish between these two cases, which have opposite implications regarding the storage of the ambiguity fix: In the first case (regular outlier), one could still store the ambiguity fix and associate it with the *previous* epoch. The next EKF pass can then restore the ambiguity from the memory. In the second case (incorrect fix), the outlier detection was actually very helpful to identify an incorrect ambiguity fix, so in this case, one should *not* store the ambiguity fix into memory! It is then a matter of tuning and implementing further continuous fix validation criteria in order to better cope with such outliers. Note, that in challenging GNSS environments, many (regular) outliers have to be expected in general, as will be seen later in the real-world examples.

In general, it appears useful to enable the outlier detection scheme for *all* EKF passes. For example, assume an outlier was not flagged as such in the first EKF pass, for example, because, the several float ambiguity estimates had not yet converged sufficiently. In the next pass, the EKF might then correctly detect this observation as an outlier, using other/more information in general.

Sometimes, alternating effects were observed in the ambiguity status of real-world data from pass to pass coming from asymmetric outlier detection results (e.g. forward passes repeatedly flagging some observation as an outlier, while the backward passes not flagging it). Note, that the phase ambiguity fixing may never converge in such cases. For the same reason, the fraction of fixed phase observations may *reduce* from one pass to another.

Again, we hereby make the reader aware of such effects, while more elaborate coping strategies might be subject to further research.

4.5. Process Termination

One possible strategy is to terminate the overall process as soon as an EKF pass cannot produce additional fixes. Data sets with very good GNSS signals commonly require two passes (being the minimum number of passes when avoiding information wasting), while then the third pass will not “find” additional ambiguity fixes and the process can be terminated. However, for highly challenging environments (urban canyons with many outliers and severe obstructions), even in the 6th pass new ambiguity fixes may be established, such that the process then terminates after the sixth pass (not finding additional fixes anymore). Real-world examples for such long “chain reactions” will be shown and discussed in Section 7.

4.6. RTS smoothing

As explained in Section 3.3., a single RTS pass is sufficient in order to gain optimal results. The RTS pass should always be performed, if tightly-coupling is performed, or (in the case of GNSS-only), if some phase ambiguities remained float (unfixed) in the final EKF pass.

The final EKF run will always contain the highest level of accumulated information, in particular the phase ambiguity fix information that was gained in previous EKF passes. Therefore, the RTS smoother is *only* applied to this *final* EKF pass. Note that for the other (previous) EKF passes, there is no need to store the intermediate EKF results required by the RTS smoother. If the number of passes is not clear a-priori (cf. Section 4.3), the intermediate results only of the *current* EKF pass are stored, and can immediately be erased once the algorithm (or the operator) decides to perform another EKF pass. [Only the ambiguity memory persists across passes, with very little memory requirements.] This way, the multi-pass strategy has the same computer memory requirement as a single-pass execution.

5. Test data set

On two consecutive days (December 19th and 20th, 2022), the same test in the Saarbrücken area in western Germany was implemented for redundancy, see Figure 5 (top). The route contains a large variety of typical real-world GNSS signal quality scenarios, ranging from an open sky to a highly obstructed urban canyon with a tunnel. The route is split up into several characteristic sections as listed in Table 1.

The urban canyon scenario (section 4) is shown in detail in Figure 5 (bottom). In many epochs within this section, the number of (presumable) code outliers was larger than the number of (presumable) non-outliers, since only the high-elevation satellites were directly visible. Other satellites' signals face (multiple) reflexions along their path to the antenna, and the direct line-of-sight (LOS) signal may be fully obstructed. This was clearly the most challenging scenario (followed by the forest), and we will see in the results, that in particular these kinds of scenarios can benefit from the proposed multi-pass method. New ambiguity fixes can still be found in pass number six in the urban canyon scenario, highlighting the existence of longer ambiguity fixing "chain reactions".

Table 1: Nine selected sections of the Saarbrücken test drive as shown in Figure 5 (top).

#	**Section name and characteristics**	**Duration** *on Dec19* *on Dec20*
1	Industrial Zone: Moderate obstructions, buildings have some distance to the street.	1:44 1:44
2	Autobahn: Occasional loss of *all* tracked satellites from underpasses. Also, obstructions by forest and other vehicles/trucks.	9:04 8:35
3	City outskirts. Some buildings/constructions close to the street, introducing relevant multi-path.	9:57 10:28
4	Urban Canyon. Narrow roads with (on average) 5-storey buildings next to the street on both sides, introducing severe multipath in particular for satellites at lower elevations. Most observations of satellites at less than 50° of elevation are obstructed entirely, only non-LOS signals reaching the antenna.	17:14 13:09
5	City-Autobahn with buildings/constructions and multiple underpasses.	4:53 4:20
6	A small town with moderate obstructions from trees and buildings.	4:42 4:57
7	Rural road along agricultural fields, barely obstructed signals.	9:51 9:59
8	Narrow rural road through a dense forest. Severe obstructions but (presumably) little multi-path.	2:24 2:13
9	Industrial Zone, see 1.	1:45 1:38

The test vehicle, a 2017 KIA Niro, was equipped with the following sensors:

- a Novatel OEM729 receiver;
- an iMAR iNAT-RQT (brief: RQT) navigation-grade IMU with ring-laser-gyroscopes;
- an Epson G320 commercial-grade MEMS IMU (brief: Epson), connected to a Novatel® PwrPak™ 7 SPAN™ receiver.

The two rover GNSS receivers were connected to the same GNSS antenna via an active GNSS splitter. Another Novatel® PwrPak™ 7 receiver (with OEM729 board) was used as a reference station.

The following signals were tracked by both GNSS receivers (both code and phase), and were therefore used for the evaluation of the data set:

- GPS on L1/L2;
- Galileo on E1/E5b;
- GLONASS on R1/R2.

Note that two frequencies are being tracked for each of the GNSSes, allowing the usage of the important linear combinations as ionosphere-free or wide-lane, see Section 6.3. and [6]. The GNSS Beidou is *not* being used here since it was not recorded by the base station receiver.

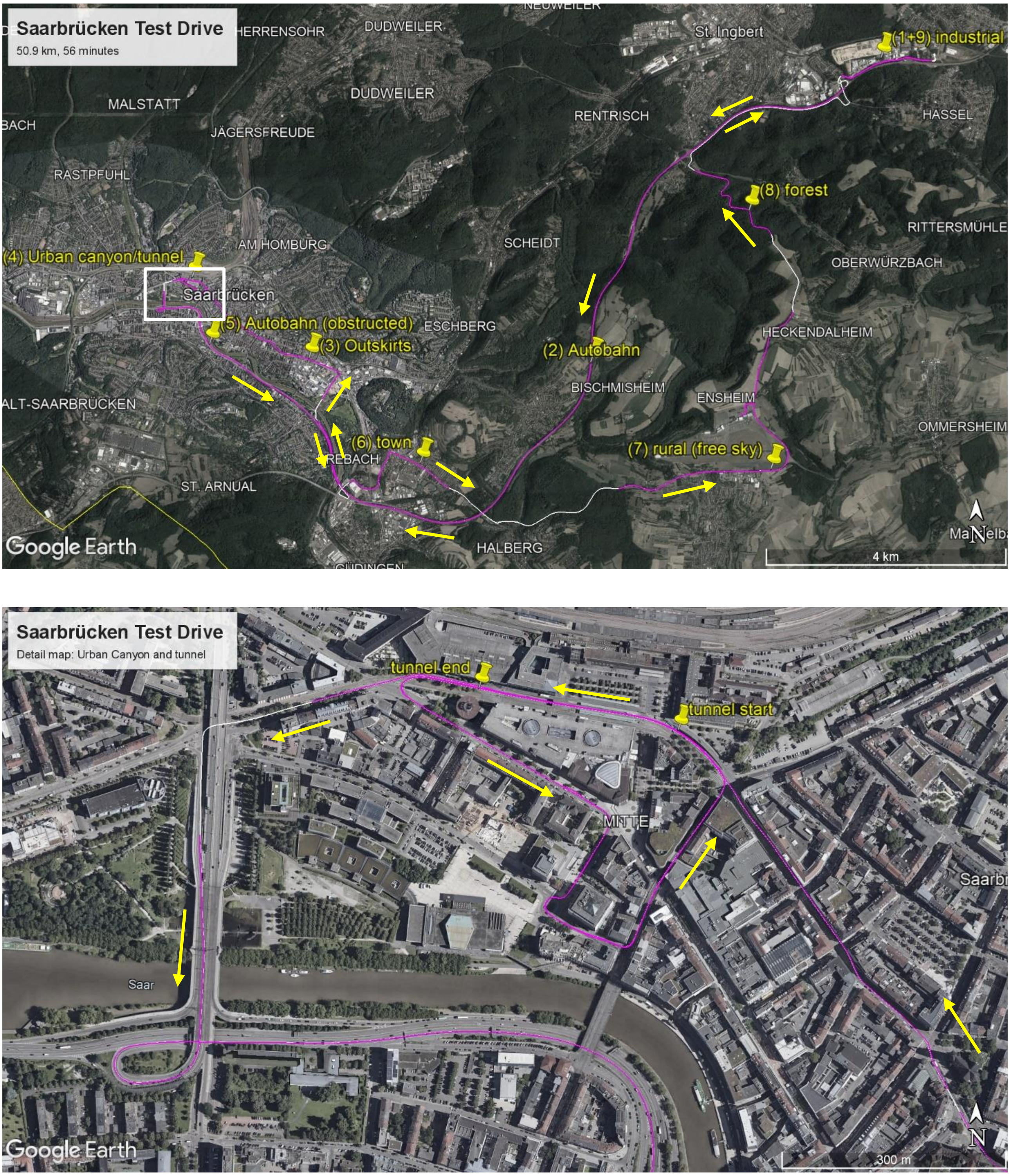

Figure 5. The Saarbrücken test drive as implemented on Dec 19th 2022. Apart from minute variations (as using a different lane), this route was repeated on Dec 20th 2022 (not shown here). ***Top***: Overview, showing all 9 sections (purple). ***Bottom***: Detail map of the urban canyon and tunnel section (section 4) as indicated by the white rectangle in the overview plot. The loop in the inner city was repeated twice in counter-clockwise direction; the tunnel has therefore been passed three times. The lower left of the map shows a part of section 5 (obstructed Autobahn).

6. Processing Details

The test data sets were processed using an Extended Kalman-Filter (EKF) in error-state space formulation, cf. [12], in combination with an RTS optimal smoother applied to the final pass (see Section 4.6). A total of six EKF-passes have been executed for this test, *not* using any termination logic based on the ambiguity fixing success (cf. Section 4.5).

6.1. GNSS-only: System Model

The EKF system state consists of three states for the position error, and additional states for the float ambiguities. Once a float ambiguity is successfully fixed, its state is "deactivated" by functionally separating it from all other states.

6.2. Tightly Coupling: System Model

The typical 15-state system model is used as a basis, consisting of three states each for position, velocity, attitude, accelerometer biases and gyroscope biases, see e.g. [12]. Like in the GNSS-only model, additional states are required for the estimation of the float ambiguities.

6.3. Two-Step Ambiguity Fixing

The LAMBDA-method [3][4][11] is used for the ambiguity fixing, performing an integer least squares estimation. For this experiment, two-frequency signals were available for both GPS (L1/L2) and Galileo (E1/E5b), enabling the computation of the important linear combinations: wide-lane and ionosphere-free. The following two-step fixing approach was used for this experiment:

1. The well-known wide-lane combination of the GNSS observations is formed and the double-differenced wide-lane float ambiguities are introduced as states to the EKF. The algorithm will continuously (in every epoch) try to fix the whole set of wide-lane float ambiguities, or a subset of these, using the LAMBDA method.

2. Once the wide-lane ambiguity is successfully fixed for a particular satellite, the L1 float ambiguity is introduced as a new state to the EKF. This can be formulated based on the ionosphere-free combination, with the already fixed wide-lane ambiguity at hand. Again, raw GNSS observations on two frequencies are required for this. As soon as the EKF manages to fix the ambiguity on L1 as well, we denote the satellite to be "fully fixed".

This two-step approach is in particular helpful for baseline lengths of several kilometres (or more), as the residual ionospheric errors (after double-differencing) can significantly degrade the double differences accuracy. Since the relative error of the ionosphere is much smaller when expressed in cycles for the much longer wide-lane wavelength (e.g. GPS: L1: ~19 cm, Wide-Lane: ~86 cm), the fixing can be performed at a much higher confidence level, although

1. the ionospheric errors are still present in the float estimates, and
2. the wide-lane combination shows a considerably higher noise level [6].

Once the ambiguity is *fully* fixed (i.e. ambiguities on both frequencies are known), the double-differenced phase observations become very precise range measurements, reaching the precision level of 1 cm for a GNSS satellite at high elevation. Again, the ionosphere-free combination is used, eliminating most of the residual (double-differenced) ionospheric errors. Note, that there is no such combination for the *tropospheric* delay, which does not rely on the carrier signal's frequency. It is possible to let the EKF estimate tropospheric parameters such as the double-differenced zenith path delay (ZTD). This can be done in combination with a-priori models for the troposphere. Such an estimation was not performed in this test, as the impact can be expected to be insignificant here, having very little variations in altitude (less than 50 meters) and a maximum baseline length (distance between rover and base station) of less than 14 kilometers.

6.4. GNSS Outlier Detection

A basic outlier detection is performed, only based on the GNSS observations themselves. In other words, in the case of tightly integration, the IMU-predicted EKF-state is *not* used for any outlier detection for this experiment.

Observations (both code and phase) are iteratively removed from the set of available observations at an epoch, as long as the maximum residual is larger than 3.0 times its a-priori standard deviation. This holds for all types of double-differenced observations: code, float phase, wide-lane fixed phase, and fully fixed phase.

If only double-differenced observations to three or less GNSS satellites are remaining (where the pivot-satellite does *not* count), potentially after removing several outliers, no redundancy is left for further detecting more outliers. In this case it can be useful to artificially down-weight these remaining observations significantly (or entirely discard these observations). For the evaluation of the presented test data sets, the a-priori standard deviation was artificially inflated by a factor of 10 for all remaining observations.

6.5. Evaluating ambiguity fix correctness

Two independent solutions are being computed: starting the first EKF-pass (total is six) in forward, or in backward direction, respectively. These two independent solutions can be used for a simple detection scheme for wrong ambiguity fixes: by comparing all fixed ambiguities at an epoch among the two solutions. The ambiguity fixing error rate will be computed as the quotient of the number of used phase observations with a bad fix (differing in the two solutions), and the *total* number of used observations with a fix. The native LAMBDA method [3][4] commonly allows the setting of a parameter, which defines the statistically expectable rate of wrong fixes. This setting was chosen to be 0.1% for this test (equivalent to a theoretic confidence level of 99.9%). The revised ratio-test was used as presented in [11].

If the ambiguity fixes differ among the two solutions, at least one of the two solutions' fixes must be incorrect (or even both). Note however, that the reverse implication does *not* apply: If both solutions *do* show the same ambiguity fix, this is *not* a proof of their correctness, and both can be (equally) wrong. It can be assumed however, that these cases are less probable, and thus the ambiguity fixing error rate can still be a useful measure of the overall fixing correctness, but it must be regarded as a lower bound. Also note, that only those fixes can be evaluated by this method which have been fixed in both solutions.

No further reference was available for this test data set; therefore, the *automated* fixing error detection was limited to the abovementioned strategy. However, the resulting trajectories were also checked *visually* by the testers, confirming the vehicle was positioned (mostly) at the center of a lane of a street, and confirming that no unnatural artifacts can be found in the resulting trajectory. This visual checking is in particular help for the GNSS-only solution, since the positioning is performed epoch-wise and fully unconstrained. Incorrect fixes commonly lead to artifacts like larger jumps, which are visually detectable.

7. Results

7.1. GNSS-only

In this section, we quantitatively evaluate the ambiguity fixing success in GNSS-only mode (no IMU used, and also no other constraints or external information). Table 2a shows the total number of phase observations. These are separated into two groups: phase observations with *at least* a wide-lane fix, and observations which are *fully* fixed. Statistics are shown separately for GPS and Galileo (remember, that no fixing is performed for GLONASS). Note, how the percentage of fixed phase observations with at least a wide-lane fix increases for the first 2-3 passes only, while the amount of fully fixed phase ambiguities still shows a significant increase during pass number 4.

Consistent for both data sets and for both GPS and Galileo, roughly 90% of all phase observations were (at least) wide-lane fixed, and roughly 75% could be fully fixed after the sixth pass.

Table 2.a: Quantitative ambiguity fixing results for **GNSS-only** evaluation, on the basis of the number of processed double-differenced phase observations

			GPS		Galileo	
Data set	**Type**	**After Pass #**	**Total**	**%**	**Total**	**%**
Dec 19th, 2022	**At least wide-lane fix**	1	15959	79.87	15261	78.78
		2	19659	94.8	19217	95.22
		3	19841	95.71	19518	95.88
		4	19907	95.64	19515	96.02
		5	19872	95.82	19532	95.88
		6	**19909**	**95.64**	**19537**	**96.07**
	Fully fixed	1	6843	34.25	6243	32.23
		2	12199	58.82	11492	56.94
		3	14774	71.27	14282	70.16
		4	15678	75.32	14995	73.78
		5	15879	76.57	15242	74.82
		6	**15912**	**76.44**	**15238**	**74.93**
Dec 20th, 2022	**At least wide-lane fix**	1	13318	80.32	11033	80.99
		2	15449	90.53	12811	91.33
		3	16137	94.17	13424	95.06
		4	16182	94.44	13470	95.36
		5	16096	93.81	13555	95.75
		6	**16172**	**94.4**	**13469**	**95.34**
	Fully fixed	1	7566	45.63	6833	50.16
		2	11307	66.26	9642	68.74
		3	12439	72.59	10823	76.64
		4	13100	76.45	11245	79.61
		5	13174	76.78	11628	82.14
		6	**13364**	**78.01**	**11550**	**81.76**

Table 2.b: Quantitative ambiguity fixing results for **GNSS-only** evaluation, on the basis of the number of epochs with certain properties, after the final (6th) pass; **entire data set.**
Note, that rows 4 and 5 sum up to 100%, and the epochs in the 6th row are always a subset of the epochs of row number 5.

row	**Epoch Ambiguity Status**	**Dec 19th, 2022**	**Dec 20th, 2022**
1	Total number of epochs at 1 Hz	4205	4158
2	Number of epochs with at least one used phase obs.	4142	4127
3	Number of epochs without any used phase obs.	63	31
4	Number of epochs with some phase obs., but without any ambiguity fix	383 (9.25%)	346 (8.38%)
5	Number of epochs including at least wide-lane ambiguity fixes	3759 (90.75%)	3781 (91.62%)
6	Number of epochs including fully fixed ambiguities	3282 (79.24%)	3415 (82.75%)

Table 2.c: Quantitative ambiguity fixing results for **GNSS-only** evaluation, on the basis of the number of epochs with certain properties, after the final (6th) pass; **only section 4 (urban canyon)**

Epoch Ambiguity Status	**Dec 19th, 2022**	**Dec 20th, 2022**
Total number of epochs at 1 Hz	797	687
Number of epochs with at least one used phase obs.	780	675
Number of epochs *without* any used phase obs.	17	12
Number of epochs with some phase obs., but without any ambiguity fix	318 (40.77%)	182 (26.96%)
Number of epochs including at least wide-lane ambiguity fixes	462 (59.23%)	493 (73.04%)
Number of epochs including fully fixed ambiguities	370 (47.44%)	491 (72.74%)

Tables 2.b and 2.c show statistics regarding the number of epochs with a particular ambiguity status. For some evaluations, this metric might be more useful, because inherently, epochs with very good signals usually also contain a larger number of satellites, and thus only looking at the number of observations may lead to counter-intuitive statistics.

Table 2.b shows statistics for the entire data sets, while Table 2.c highlights the same figures only for section number 4 (the urban canyon in the Saarbrücken city center, see Figure 5, bottom). According to the results in Table 2c, on Dec 19th (Dec 20th), only 59%

(73%) of all epochs with phase observations contain (at least) a wide-lane fix, or in other words, 41% (27%) of all epochs in section 4 with phase observations have *all* of them only estimated as *float* ambiguities. This highlights the challenges of fixing the ambiguities in a GNSS-only setup under very challenging conditions.

Table 2.d: Quantitative ambiguity fixing results for **GNSS-only** evaluation **by sections and passes.**

Section		**Dec 19th, 2022**						**Dec 20th, 2022**					
	pass # →	1	2	3	4	5	**6**	1	2	3	4	5	**6**
1	# epochs w/obs	98	100	98	100	98	**99**	102	104	102	104	102	**103**
	with WL fix	98	100	98	100	98	**99**	101	104	102	104	102	**103**
	with full fix	0	31	98	100	98	**99**	97	104	102	104	102	**103**
2	# epochs w/obs	500	503	501	503	501	**502**	468	479	473	479	476	**478**
	with WL fix	452	496	494	496	494	**495**	414	428	460	467	463	**466**
	with full fix	74	165	221	223	221	**222**	115	277	309	326	329	**325**
3	# epochs w/obs	565	567	565	567	565	**566**	602	607	603	607	603	**606**
	with WL fix	497	567	565	567	565	**566**	557	598	599	601	599	**600**
	with full fix	303	409	511	513	511	**512**	490	563	561	563	561	**563**
4	# epochs w/obs	763	782	764	782	764	**780**	667	677	674	677	674	**675**
	with WL fix	195	438	462	464	462	**463**	226	365	480	495	480	**494**
	with full fix	0	65	194	342	371	**370**	125	322	435	446	477	**492**
5	# epochs w/obs	237	241	238	241	238	**240**	208	214	212	214	212	**213**
	with WL fix	185	224	222	224	222	**223**	182	214	212	214	212	**213**
	with full fix	68	161	186	188	186	**187**	3	101	126	128	126	**128**
6	# epochs w/obs	267	270	267	270	267	**269**	289	291	289	291	289	**290**
	with WL fix	199	267	266	269	267	**269**	275	291	289	291	289	**290**
	with full fix	180	244	266	268	266	**267**	257	291	289	291	289	**290**
7	# epochs w/obs	569	571	569	571	569	**570**	580	582	580	582	580	**581**
	with WL fix	569	571	569	571	569	**570**	580	582	580	582	580	**581**
	with full fix	569	571	569	571	569	**570**	580	582	580	582	580	**581**
8	# epochs w/obs	103	106	111	106	111	**105**	128	130	128	130	128	**129**
	with WL fix	35	66	64	66	64	**66**	0	0	0	0	0	**0**
	with full fix	15	50	64	66	64	**66**	0	0	0	0	0	**0**
9	# epochs w/obs	97	99	97	99	97	**98**	97	99	97	99	97	**98**
	with WL fix	97	99	97	99	97	**98**	97	99	97	99	97	**98**
	with full fix	97	99	97	99	97	**98**	97	99	97	99	97	**98**

Table 2.d also shows statistics based on the number of epochs with certain characteristics, but this time the results are split up by pass and by section. The table shows, how the number of successfully fixed GNSS observations evolves from pass to pass. For sections with strong GNSS signals, a full L1/L2 ambiguity might have been established already

before the beginning of the section, so that after pass #1, most ambiguities may have been fixed already (sections 1,7,9) and multi-passing will not (relevantly) improve the overall fixing rate. However, in challenging GNSS environments, it can be seen how multi-passing supports the ambiguity fixing (sections 2-6 and 8, in particular section 4), indicating the presence of "ambiguity fixing chain reactions". All passes with significant improvements in terms of the number of epochs with fixes are highlighted in the table using a grey background color.

Note, that on Dec 20th, not a single ambiguity fix was established during section number 8 (dense forest).

Table 2.e shows the percentage of wrong ambiguity fixes, based on the forward/backward evaluation strategy as explained in Section 6.5. The LAMBDA method, which was used for the ambiguity fixing, was configured to use a confidence level of 99.9%, so in other words, a fraction of 0.1% of incorrect ambiguity fixes (on average) had to be expected stochastically. Note, that ambiguity fixing is always a trade-off between confidence and quantity.

Still, in real-world data sets, a higher fraction of incorrect fixes has to be expected, since the underlying assumption of the LAMBDA method of only dealing with normally distributed measurement errors does not hold: Residual ionospheric and tropospheric errors show a strong correlation over time. Also, multi-path errors usually show systematic artifacts and are *not* normally distributed over short periods [14].

Note, that those wrong fixes which could be found by the forward/backward comparison strategy (cf. Section 6.5.) can *automatically* be eliminated by simply not using the (apparently wrong) fixes at all, and reverting to the corresponding float ambiguity instead.

Table 2.e: Ambiguity fixing error rates for **GNSS-only** evaluation (cf. Section 6.5.).

Wrong Fixes	**Dec 19th, 2022**	**Dec 20th, 2022**
After Pass 1	0.78%	0.04%
After Pass 2	0.05%	0.00%
After Pass 3	0.08%	0.02%
After Pass 4	0.16%	0.00%
After Pass 5	0.19%	0.02%
After Pass 6	0.16%	0.00%

Figure 6. Development of ambiguity fixes over 6 EKF-passes for **GNSS-only**; section 4 (urban canyon) with very challenging GNSS conditions. Data rate is 1 Hz. Grey dots: *float* phase observation; colored thin dots: wide-lane fixed phase observation; colored thick dots: fully fixed phase observation; red '+': Outlier.

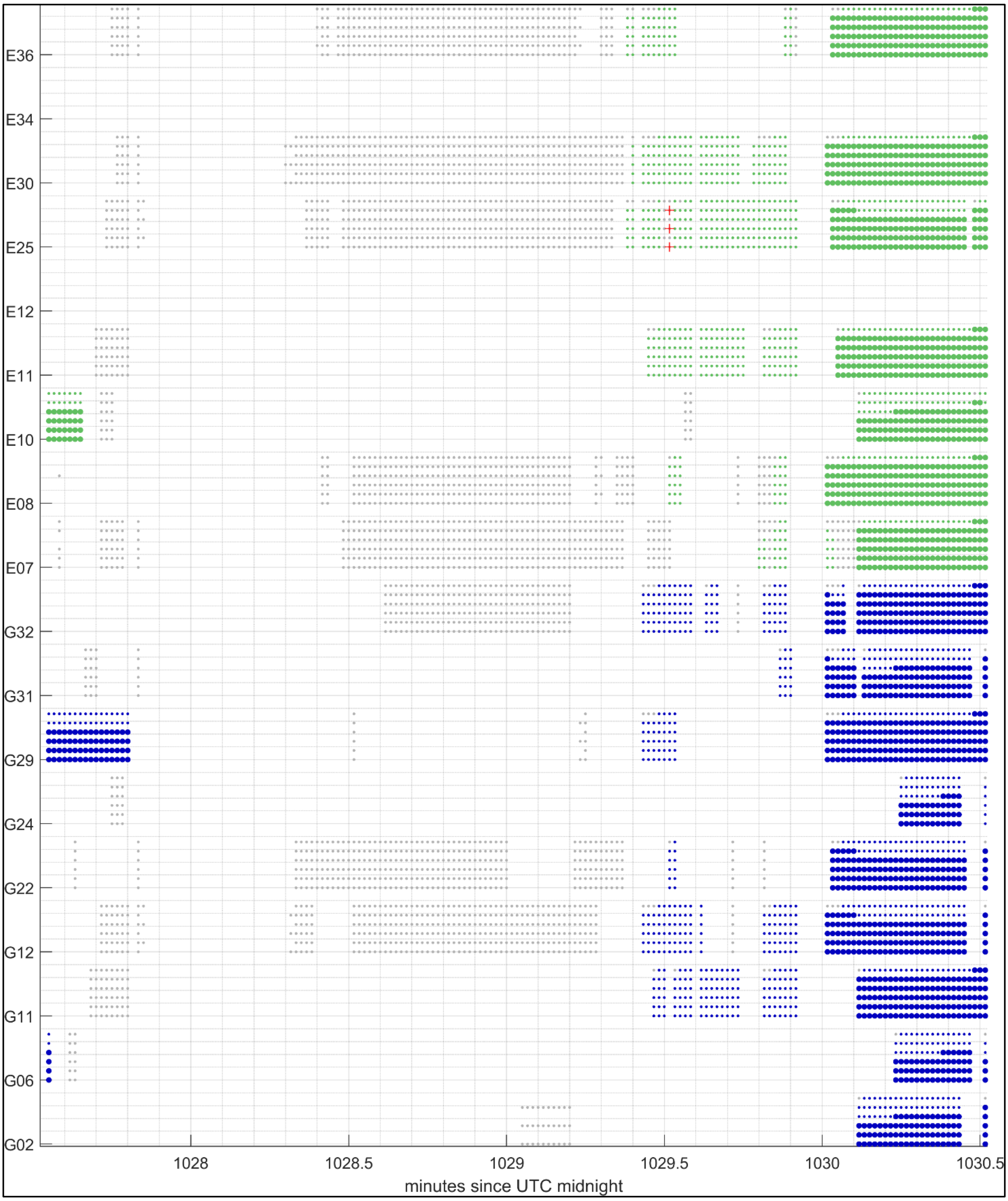

Figure 6 shows the development of the ambiguity status for a selected 3-minutes interval inside section 4 (urban canyon). Only Galileo and GPS satellites are shown for clarity, while GLONASS float phase observations were used as well. For each satellite, 6 lines of ambiguity status information are shown, reflecting (from top to bottom) the six EKF passes performed.

The vehicle was inside the tunnel around minute 1028. When leaving the tunnel, stop-and-go traffic has led to a approx. 1 minute of poor GNSS signals on the ramp of the tunnel, with severe obstructions to both sides from concrete walls and buildings, and also to the back (by the tunnel itself), see minutes 1028.3 – 1029.4. In the GNSS-only evaluation, the LAMBDA method was not able to find an ambiguity fix (at the setting of 99.9% confidence). Note, that the float observations (grey dots in the Figure 6) are still very helpful for the positioning, as the usual constancy assumptions still holds. The RTS smoother is then able to maximize the information from this property for the positioning, even though no ambiguity could be fixed.

First wide-lane fixes then succeed between minutes 1029.5 – 1030, but no full fix can be found. The wide-lane fix allows a positioning uncertainty of roughly 10-15 centimeters.

The signal quality then becomes better (leaving the urban canyon at approx. minute 1030.0). One can see how the multiple passes then successively lead to a much larger fraction of wide-lane-fixes and full fixes.

As an example, we'll discuss the evolution of G24's ambiguity status, starting at minute 1030.25. G24 is only visible for 12 consecutive epochs here (i.e., for 12 seconds). EKF pass 1 is traversing the data *forward* in time. During the first epoch of pass 1, the ambiguity of G24 is "float" (grey dot). But already in the second epoch, the wide-lane fix succeeds, and this is also due to the fact, that several other satellites have already been wide-lane fixed at that time (green thin dots). Then during the *second* pass (now traversing the data *backward* in time!), the wide-lane-fix of G24 can immediately be restored from memory, and there is no need to again fix the wide-lane ambiguity of G24. Throughout the 12 epochs, the algorithm continuously tries to estimate a *full* fix of G24 (among others, of course) using the LAMBDA method. This is however not successful, as can be seen in Figure 6 by the thin blue dots throughout all 12 epochs. Note however, that the full fix succeeds then just "after" that (thinking backwards in time) for several other satellites at minute 1030.1 (G12, G22, and E25). With this additional information, the algorithm then manages to fully fix G24 (and also G06) in the *third* EKF (forward-)pass just before 1030.4.

This fix itself is then fully exploited in EKF pass number *four*, where G24 is now fully fixed for all 12 epochs. In this example, *no* further fixes can be established in passes 5 and 6, so for this particular example, only using a total of *four* EKF passes would have led to the same result.

Note, that already after pass number four, the algorithm was in fact able to fully fix *all visible satellites* in the interval 1030.0 – 1030.5.

7.2. Tightly coupling results

This section summarizes the tightly coupling results, again focusing on the ambiguity fixing success in the scope of the multi-pass strategy.

For reading this section, it is useful to directly compare the shown tables against the corresponding tables in the previously shown GNSS-only results of Section 7.1. (compare Table 3.a against Table 2.a, etc.). Note that for layout reasons, Table 3.d was split into two tables (3.d.1 for Dec 19th, and 3.d.2 for Dec. 20th). Some example findings are highlighted at the end of this subsection, which may also serve as reading examples.

Table 3.a: Quantitative ambiguity fixing results for **tightly-coupled** evaluation, on the basis of the number of phase observations

	Used IMU:		**iMAR RQT**				**Epson G320**			
			GPS		**Galileo**		**GPS**		**Galileo**	
Data set	**Type**	**After Pass #**	**#obs**	**%**	**#obs**	**%**	**#obs**	**%**	**#obs**	**%**
Dec 19th, 2022	**At least wide-lane fix**	1	16528	82.84	15612	80.87	16532	82.57	15668	81.28
		2	19978	96.50	19448	96.02	19935	96.28	19318	95.52
		3	20553	99.07	20114	98.86	20239	97.50	19966	98.28
		4	20578	98.85	20041	98.22	20306	97.61	19898	97.58
		5	20595	99.20	20123	98.81	20339	97.92	19957	98.17
		6	**20632**	**99.00**	**20057**	**98.27**	**20378**	**97.99**	**19931**	**97.74**
	Fully fixed	1	7272	36.45	6474	33.54	7193	35.92	6540	33.93
		2	11946	57.70	11089	54.75	13029	62.92	12391	61.27
		3	14560	70.19	14460	71.07	15056	72.53	14562	71.68
		4	15812	75.96	15260	74.79	15432	74.18	14924	73.19
		5	17500	84.29	17086	83.90	15720	75.68	15265	75.09
		6	**18511**	**88.82**	**17687**	**86.66**	**15855**	**76.24**	**15239**	**74.73**
Dec 20th, 2022	**At least wide-lane fix**	1	13133	79.71	11187	82.82	13654	82.33	11030	81.24
		2	16886	98.82	13877	98.31	16801	98.23	13742	97.54
		3	17114	99.44	14095	99.16	17096	99.34	14092	99.15
		4	17070	99.31	14097	99.05	17107	99.49	14112	99.04
		5	17114	99.44	14136	99.40	17127	99.50	14168	99.47
		6	**17066**	**99.30**	**14099**	**99.06**	**17129**	**99.62**	**14121**	**99.10**
	Fully fixed	1	7296	44.28	6557	48.54	7445	44.89	6820	50.23
		2	11775	68.91	10194	72.22	11925	69.72	10161	72.12
		3	13550	78.73	11709	82.37	13215	76.79	11590	81.55
		4	14384	83.69	12192	85.67	14161	82.36	12081	84.78
		5	14495	84.22	12308	86.54	14250	82.79	12420	87.20
		6	**14597**	**84.94**	**12285**	**86.31**	**14364**	**83.54**	**12352**	**86.69**

Table 3.b: Quantitative ambiguity fixing results for **tightly-coupled** evaluation, on the basis of the number of epochs with certain properties, after the final (6th) pass; **entire data set**

Epoch Ambiguity Status	**Dec 19th, 2022**		**Dec 20th, 2022**	
	RQT	**Epson**	**RQT**	**Epson**
Total number of epochs at 1 Hz	4203		4156	
Number of epochs with at least one used phase obs.	4143	4141	4129	4130
Number of epochs without any used phase obs.	60	62	27	26
Number of epochs with some phase obs., but without any ambiguity fix	28 (0.68%)	200 (4.83%)	27 (0.65%)	7 (0.17%)
Number of epochs including at least wide-lane ambiguity fixes	4115 (99.32%)	3941 (95.17%)	4102 (99.35%)	4123 (99.83%)
Number of epochs including fully fixed ambiguities	3912 (94.42%)	3446 (83.22%)	3903 (94.50%)	3817 (92.44%)

Table 3.c: Quantitative ambiguity fixing results for **tightly-coupled** evaluation, on the basis of the number of epochs with certain properties, after the final (6th) pass; **only section 4 (urban canyon)**

Epoch Ambiguity Status	**Dec 19th, 2022**		**Dec 20th, 2022**	
	RQT	**Epson**	**RQT**	**Epson**
Total number of epochs at 1 Hz	797		687	
Number of epochs with at least one used phase obs.	780	781	680	680
Number of epochs without any used phase obs.	17	16	7	7
Number of epochs with some phase obs., but without any ambiguity fix	11 (1.41%)	157 (20.10%)	21 (3.09%)	2 (0.29%)
Number of epochs including at least wide-lane ambiguity fixes	769 (98.59%)	624 (79.90%)	678 (99.71%)	659 (96.91%)
Number of epochs including fully fixed ambiguities	725 (92.95%)	531 (67.99%)	647 (95.14%)	650 (95.59%)

Table 3.d.1: Quantitative ambiguity fixing results for **tightly-coupled** evaluation **for all sections and passes, for Dec 19th, 2022.** *For a detailed description, see Table 2d.*

Section		iMAR RQT						Epson G320					
	pass # →	1	2	3	4	5	**6**	1	2	3	4	5	**6**
1	# epochs w/obs	98	100	98	100	98	**100**	98	100	98	100	98	**100**
	with WL fix	11	100	98	100	98	**100**	86	100	98	100	98	**100**
	with full fix	0	31	98	100	98	**100**	0	99	98	100	98	**100**
2	# epochs w/obs	500	503	501	503	501	**503**	500	503	501	503	501	**503**
	with WL fix	476	501	499	501	499	**501**	474	501	499	501	499	**501**
	with full fix	67	117	212	223	368	**392**	74	165	221	223	221	**223**
3	# epochs w/obs	565	567	565	567	565	**567**	565	567	565	567	565	**567**
	with WL fix	561	567	565	567	565	**567**	541	567	565	567	565	**567**
	with full fix	250	426	562	564	565	**567**	255	426	511	513	511	**513**
4	# epochs w/obs	763	782	782	781	782	**783**	763	782	771	783	771	**783**
	with WL fix	358	668	756	752	771	**773**	176	584	625	623	621	**626**
	with full fix	37	224	413	598	599	**727**	42	227	402	446	528	**534**
5	# epochs w/obs	237	241	239	241	239	**241**	237	241	239	241	239	**241**
	with WL fix	140	237	239	241	239	**241**	211	241	239	241	239	**241**
	with full fix	28	188	220	222	220	**222**	66	154	166	168	166	**168**
6	# epochs w/obs	267	270	267	270	268	**270**	267	270	267	270	268	**270**
	with WL fix	261	268	266	270	268	**270**	261	268	266	270	268	**270**
	with full fix	219	264	262	268	266	**269**	240	268	266	268	266	**268**
7	# epochs w/obs	569	571	569	571	569	**571**	569	571	569	571	569	**571**
	with WL fix	569	571	569	571	569	**571**	569	571	569	571	569	**571**
	with full fix	569	571	569	571	569	**571**	569	571	569	571	569	**571**
8	# epochs w/obs	103	109	112	109	112	**109**	103	106	111	106	111	**106**
	with WL fix	59	94	92	94	92	**94**	37	66	64	66	64	**66**
	with full fix	19	51	64	66	64	**66**	16	52	64	66	64	**66**
9	# epochs w/obs	97	99	97	99	97	**99**	97	99	97	99	97	**99**
	with WL fix	97	99	97	99	97	**99**	97	99	97	99	97	**99**
	with full fix	97	99	97	99	97	**99**	97	99	97	99	97	**99**

Table 3.d.2: Quantitative ambiguity fixing results for **tightly-coupled** evaluation **for all sections and passes, for Dec 20th, 2022.** *For a detailed description, see Table 2d.*

Section		**iMAR RQT**						**Epson G320**					
	pass # →	1	2	3	4	5	**6**	1	2	3	4	5	**6**
1	# epochs w/obs	102	104	102	104	102	**104**	102	104	102	104	102	**104**
	with WL fix	0	103	102	104	102	**104**	12	104	102	104	102	**104**
	with full fix	0	103	102	104	102	**104**	0	103	102	104	102	**104**
2	# epochs w/obs	469	479	476	478	476	**478**	468	479	476	479	476	**479**
	with WL fix	194	467	474	476	474	**476**	426	477	474	478	474	**478**
	with full fix	0	212	267	355	353	**355**	116	327	329	331	329	**331**
3	# epochs w/obs	602	607	605	607	605	**607**	602	607	605	607	605	**607**
	with WL fix	596	606	605	607	605	**607**	500	584	605	607	605	**607**
	with full fix	540	563	563	569	567	**569**	492	563	561	595	599	**601**
4	# epochs w/obs	673	682	680	682	680	**682**	673	682	679	682	680	**682**
	with WL fix	634	678	678	661	678	**661**	458	674	676	680	678	**680**
	with full fix	427	660	663	649	666	**649**	232	615	649	671	664	**636**
5	# epochs w/obs	208	214	212	214	212	**214**	208	214	212	214	212	**214**
	with WL fix	185	214	212	214	212	**214**	184	214	212	214	212	**214**
	with full fix	13	154	196	200	202	**204**	10	110	143	182	180	**182**
6	# epochs w/obs	289	291	289	291	289	**291**	289	291	289	291	289	**291**
	with WL fix	286	291	289	291	289	**291**	286	291	289	291	289	**291**
	with full fix	159	291	289	291	289	**291**	268	291	289	291	289	**291**
7	# epochs w/obs	580	582	580	582	580	**582**	580	582	580	582	580	**582**
	with WL fix	580	582	580	582	580	**582**	580	582	580	582	580	**582**
	with full fix	580	582	580	582	580	**582**	580	582	580	582	580	**582**
8	# epochs w/obs	128	130	128	130	128	**130**	128	130	128	130	128	**130**
	with WL fix	89	130	111	130	128	**130**	48	129	111	128	128	**128**
	with full fix	37	38	63	61	61	**61**	36	78	111	128	109	**128**
9	# epochs w/obs	97	99	97	99	97	**99**	97	99	97	99	97	**99**
	with WL fix	97	99	97	99	97	**99**	97	99	97	99	97	**99**
	with full fix	97	99	97	99	97	**99**	97	99	97	99	97	**99**

Table 3.e: Ambiguity fixing error rates for **tightly** evaluation (cf. Section 6.5.).

Wrong Fixes	**Dec 19th, 2022**		**Dec 20th, 2022**	
	RQT	**Epson**	**RQT**	**Epson**
After Pass 1	0.77%	0.68%	0.02%	0.02%
After Pass 2	0.05%	0.05%	0.10%	0.00%
After Pass 3	0.07%	0.08%	0.04%	0.01%
After Pass 4	0.05%	0.20%	0.03%	0.01%
After Pass 5	0.14%	0.21%	0.04%	0.01%
After Pass 6	0.18%	0.41%	0.03%	0.01%

Figure 7. Development of ambiguity fixes over 6 EKF-passes for **tightly coupling** with a navigation grade IMU (iMAR RQT); section 4 (urban canyon) with very challenging GNSS conditions. Data rate is 1 Hz. Grey dots: *float* phase observation; colored thin dots: wide-lane fixed phase observation; colored thick dots: fully fixed phase observation; red '+': Outlier.

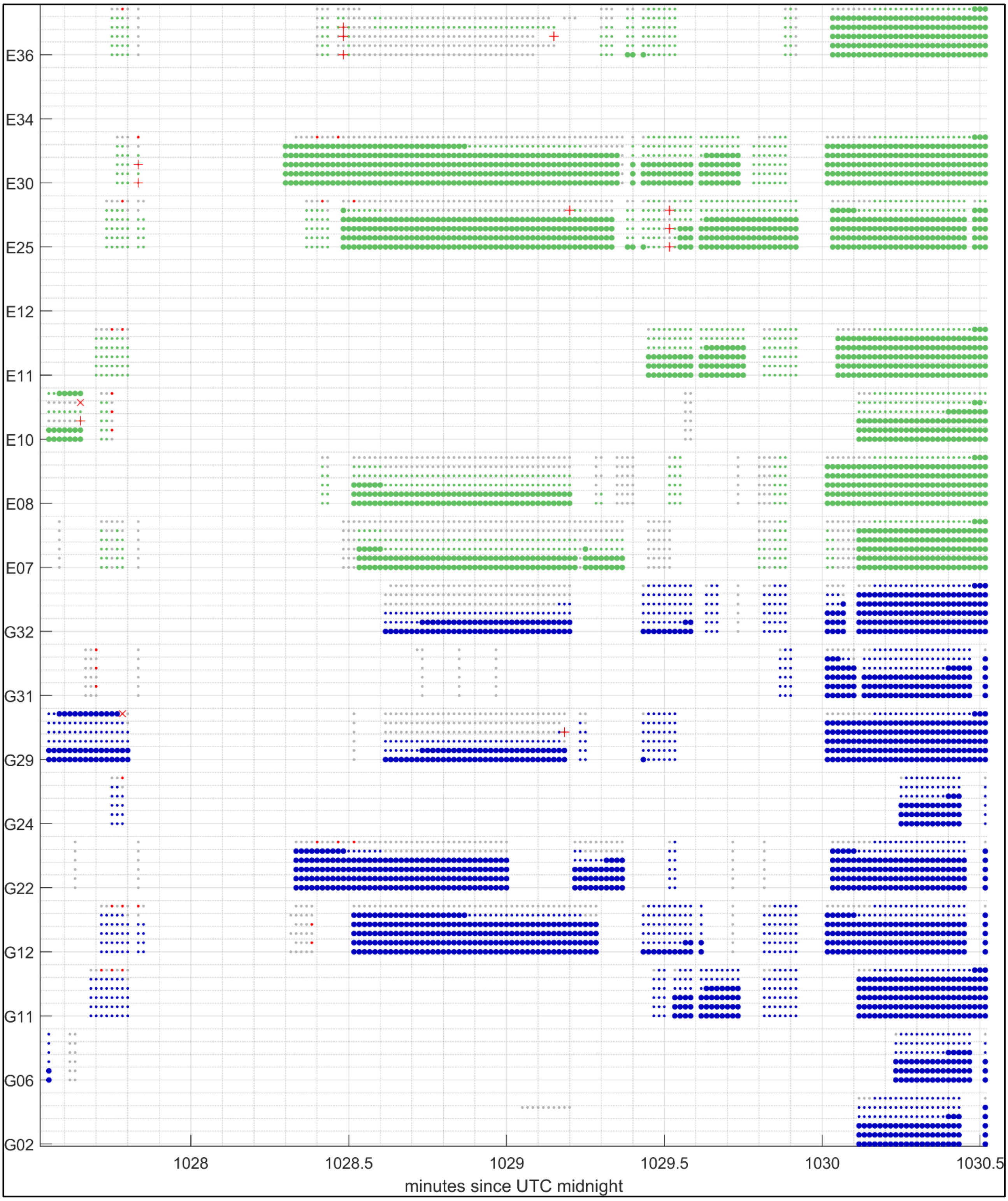

Figure 7 shows the same situation shown in Figure 6, but this time for tightly-coupling using the navigation-grade iMAR RQT as IMU. After leaving the tunnel at minute 1028.4, no wide-lane fixes succeed at first, until minute 1029.4, where a wide-lane fix can be established for multiple satellites (dots changing from grey to colored). It takes until minute 1030.4, until the first satellites can be fully fixed (dots changing from thin to thick). This enables various additional fixes after minute 1030.5 (not shown in the figure) which can be exploited in the *second* pass (again, backward in time), for example E25, E10, G31, G24. Although there is a full GNSS outage of 5 seconds near minute 1029.9, the IMU is helpful in transferring already acquired position knowledge over time (in the backward pass number to: from minute 1030.0 to 1029.9). Thereby, a wide-lane fix can immediately be acquired for most satellites at 1029.9.

Note in the interval from minute 1028.3 to 1029.9, how the various satellites become visible and disappear again multiple times. Not a single satellite can be tracked over the whole interval. What can be seen here, is a real-world ambiguity fixing “chain reaction” of length 5:

- during **pass #1**: no full fixes can be established (but some wide-lane fixes);
- during **pass #2**: some satellites become fully fixed (backward in time; E25, E30, G22 and G12);
- during **pass #3**, other satellites, by exploiting the knowledge of the first two passes, can be fully fixed (forward in time; see for example E30, E25, E11 and G11 just after minute 1029.6);
- during **pass #4**, more fixes can be found, for example, E08, E07 at minute 1028.6;
- during **pass #5**, more satellites can be fully fixed (for example G32 and G12 just before 1029.6, or G29 at minute 1028.7);
- even during **pass #6**, more full fixes can be established, for example E36, E25 and G29 around minute 1029.4.

To summarize the results for the case of tightly coupling, a few findings are listed here, mainly coming from the comparisons against the GNSS-only results shown in the previous section:

- The fraction of phase observations with at least a wide-lane fix has significantly improved from adding an IMU: from roughly 90% to 98%-99% (see Tables 2.a and 3.a). Also, the fraction of fully-fixed phase observations increased significantly, from approx. 75%-80% to ~86% (except for the Dec 19th data set with the Epson IMU, where the fraction remained at ~75%).

- Likewise, the number of epochs with at least a wide-lane or a full fix significantly increased from approximately 90% up to 99% for the entire data set (cf. Tables 2.b and 3.b). An even stronger increase can be observed for the urban canyon (section number 4). For example, the fraction of epochs with at least a wide-lane fix increased from 59.2% (GNSS-only) to 98.6% (using RQT). Similarly, the fraction of epochs with fully fixed ambiguities increased from 47.4% (GNSS-only) to 92.9% (using RQT). The improvements on Dec 19th were more moderate when using the lower grade Epson IMU.

- For both GNSS-only and tightly, it is in particular the urban canyon (section 4) which benefits from multiple passes. Again, the sixth pass has led to significantly more ambiguity fixes. It is just noted here, that the authors have found some examples, where (few) new ambiguity fixes were found in *pass number 8*, which could not be established in any of the first seven passes!

- It can in general be seen in tables 2.d and 3.d.1 / 3.d.2, that the more challenging a scenario is, the more it benefits from the additional passes.

- On some occasions, alternating effects were found, where the ambiguity fixing never seems to converge (see Section 4.4). This was the case for some of the observations on Dec 20th in sections 8 and 9 when using the Epson IMU, see the entries in Table 3.d.2. highlighted with yellow color. Here, the fixed phase observation’s residual was very close to the outlier threshold, however only exceeding it during the forward passes.

8. Conclusions

With the presented experiments we have analyzed, if and to what extent EKF-multi-passing is helpful for the fixing of double-differenced GNSS phase ambiguities. The central findings are summarized here:

- For data sets with very good GNSS satellite visibility (many satellites visible, no or few outages), a simple re-winding, or the combination of position estimates from a forward- and a backward EKF pass can be expected to provide comparable results to the multi-pass strategy. Even modeling the ambiguity fixes as pseudo-EKF-measurements should (in combination with RTS filtering) allow comparable results.

- A simple re-winding (see Section 3.1) performed for a GNSS-only data set is equivalent (in terms of the used information) to using *two* consecutive EKF-passes with the strategy presented here. For challenging environments, with many obstructions, satellite lock losses, cycle slips and outliers e.g. from multi-path, it was shown, that additional EKF-passes lead to an increasing number of ambiguity fixes. Under such conditions, more and longer ambiguity fixing “chain reactions” can be found (cf. Sect. 2.4).

- For tightly-coupled data sets, it was shown based on theoretic considerations, that a data set must be considered as a whole, not only limiting the rewinding to within the current “connected component” (as defined in Sect. 2.1). In this sense, the multi-passing strategy described essentially equals an iterated re-winding over the entire data set. The method shows a low implementation complexity, because “chain reactions” do not have to be handled *explicitly*. Multi-passing implicitly accumulates any ambiguity fix information, thereby iteratively providing the maximum amount of information for the estimation of each epoch’s system state (in particular the position, but also velocity and attitude in the case of a tightly integration) *and* the establishment of additional ambiguity fixes.

- New ambiguity fixes could be found in the 6th EKF pass. There ware few examples in the urban canyon scenarios, where even the 7th and 8th EKF-pass have led to more ambiguity fixes, the number of additional fixes however being insignificant. This shows, that longer “chain reaction” exist in real-world data sets, for both GNSS-only and tightly-coupled evaluations.

- While not being the main focus of this paper, it could be quantitatively shown, how much the usage of an IMU supports the phase ambiguity fixing by effectively transferring position information through time. This was shown for both a navigation-grade, and a commercial-grade IMU.

- Tightly coupling is not only helpful for the ambiguity fixing in situations with GNSS outages: the higher the grade of the IMU, the higher the redundancy of available information for both the position determination and (intrinsically) for the ambiguity fix determination. Therefore, the IMU will also be helpful in challenging situations with many obstructions and a low number of visible satellites, even in the absence of a full GNSS outage.

- The ambiguity fixes established in the later EKF-passes show a slight tendency to be more error-prone. This is not surprising, as these fixes are mostly taking place in challenging environments (like the urban canyon), where the risk of ending up with a wrong ambiguity fix is much higher *in general*. However, the comparison of the number of additional fixes with the low number of additional *erroneous* fixes in the presented results still suggests, that multi-passing can be a useful method in general. For some applications, the usual trade-off between availability and reliability may lead to the refusal of new fixes in the later EKF passes, or, for example, the requirement of a higher fixing confidence level for these passes.

References

[1] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems.

[2] Rauch, H. E., Tung, F., & Striebel, C. T. (1965). Maximum likelihood estimates of linear dynamic systems. AIAA journal, 3(8), 1445-1450.

[3] Teunissen, P. J. G., De Jonge, P. J., & Tiberius, C. C. J. M. (1995, September). The LAMBDA method for fast GPS surveying. In International Symposium “GPS Technology Applications” Bucharest, Romania.

[4] Teunissen, P. J., De Jonge, P. J., & Tiberius, C. C. J. M. (1997). Performance of the LAMBDA method for fast GPS ambiguity resolution. Navigation, 44(3), 373-383.

[5] Teunissen, P. J., & Montenbruck, O. (Eds.). (2017). Springer handbook of global navigation satellite systems (Chapter 8: GLONASS). Cham, Switzerland: Springer International Publishing.

[6] Teunissen, P. J., & Montenbruck, O. (Eds.). (2017). Springer handbook of global navigation satellite systems (Chapter 20: Combinations of Observations). Cham, Switzerland: Springer International Publishing.

[7] Teunissen, P. J., & Montenbruck, O. (Eds.). (2017). Springer handbook of global navigation satellite systems (Chapter 23: Carrier Phase Integer Ambiguity Resolution). Cham, Switzerland: Springer International Publishing.

[8] Teunissen, P. J., & Montenbruck, O. (Eds.). (2017). Springer handbook of global navigation satellite systems (Chapter 24: Differential Positioning). Cham, Switzerland: Springer International Publishing.

[9] Zhang, X., Zhang, Y., & Zhu, F. (2020). A method of improving ambiguity fixing rate for post-processing kinematic GNSS data. Satellite Navigation, 1(1), 1-13.

[10] Li, B., Verhagen, S., & Teunissen, P. J. (2013). GNSS integer ambiguity estimation and evaluation: LAMBDA and Ps-LAMBDA. In China Satellite Navigation Conference (CSNC) 2013 Proceedings: Satellite Navigation Signal System, Compatibility & Interoperability• Augmentation & Integrity Monitoring• Models & Methods (pp. 291-301). Springer Berlin Heidelberg.

[11] Teunissen, P. J., & Verhagen, S. (2009). The GNSS ambiguity ratio-test revisited: a better way of using it. Survey Review, 41(312), 138-151.

[12] Becker, D. (2016). Advanced calibration methods for strapdown airborne gravimetry (No. 51). Technische Universität Darmstadt.

[13] Paul Groves, Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, Second Edition , Artech, 2013.

[14] Xie, P., & Petovello, M. G. (2014). Measuring GNSS multipath distributions in urban canyon environments. IEEE Transactions on Instrumentation and Measurement, 64(2), 366-377.