DG14™ RTK

Making use of Magellan's BLADE™ GNSS Processing Algorithm for Multipurpose L1 RTK

April 16, 2007

Table of Contents

Foreword
This paper describes how the new Magellan RTK algorithms called BLADE™ (Base Line Accurate Determination Engine) are used in the recently released DG14™ RTK

1 Description of BLADE

BLADE refers to the entire suite of GNSS processing algorithms currently used in Magellan Professional's products for RTK and post processing applications.

- It is the engine of the post-processing package GNSS Solutions, where it is used to process single-frequency GPS+SBAS, dual-frequency GPS, single-frequency GPS+GLONASS, and dual-frequency GPS+GLONASS measurements.
- It is also the engine in Z-Max where it is used to provide Instant RTK and Long Range Kinematic capabilities.
- In ProMark3 it processes range and carrier phase measurements from both GPS and SBAS satellites, thus making the ProMark3 a true GNSS product. The combined processing from two GNSS satellite systems allows for a far more rapid time to fix integer ambiguities thus providing a practical RTK system from an L1-only product (see ProMark3 RTK White Paper for a complete description of this technique and its value in surveying).
- In the DG14™ RTK it provides moving base RTK, heading determination with auto-calibration, GPS+SBAS processing, and Flying RTK.

2 DG14 RTK

This paper describes the standard and advanced features of the new L1 RTK GNSS board from Magellan: DG14™ RTK.

DG14 RTK is the DG14 GNSS board with RTK function incorporated. The board has been on the market for several years and has earned a good reputation for its data/position quality in a variety of OEM applications. Introducing the RTK function extends the DG14 application range by providing decimeter and centimeter position accuracy.

Along with standard RTK function (Float and Fixed RTK) DG14 RTK provides the following extended RTK features:

- Flying RTK™
- RTK with moving base (relative RTK)
- Heading Determination (including self calibration)
- GPS+SBAS RTK Processing

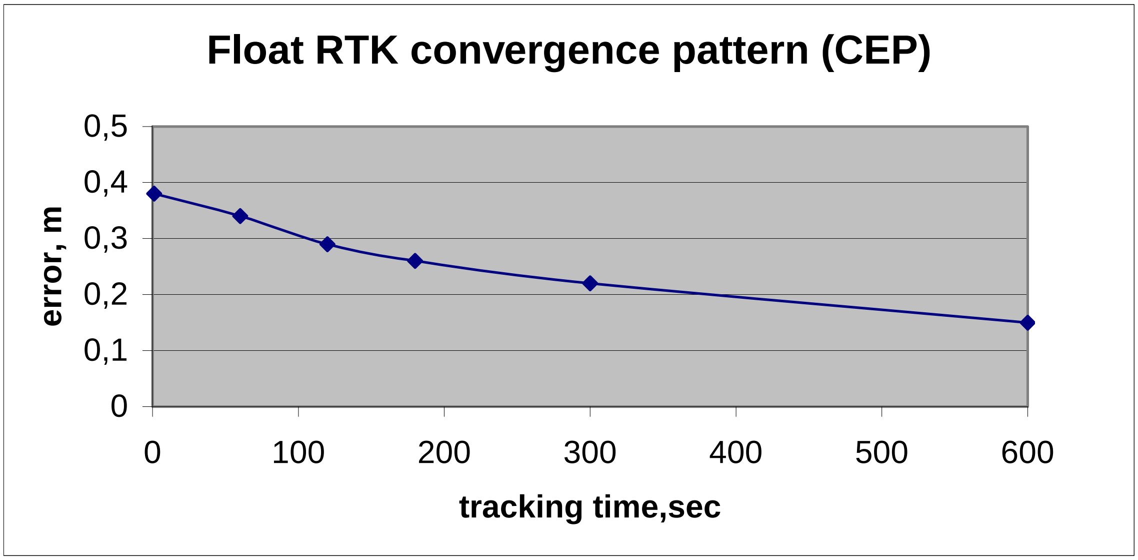
When demonstrating performance, we will focus on statistical figures rather than on presenting particular test results. All the data we used for performance evaluation were collected with static DG14 receivers. However, RTK was running w/o static assumption, with a few exceptions to show the improvement when choosing static option. All performance was evaluated with default DG14 RTK settings.

One very important note must be made. When collecting statistics we used the RTK auto-reset methodology. We always used fixed-length intervals between RTK resets regardless of its current status. Some vendors provide similar auto-reset statistics using the float-length intervals measured when RTK reset occurs, depending on the current RTK status. This float-length interval approach usually gives a more optimistic statistic compared to fixed-length interval statistic. Moreover, the fixed-length interval statistic allows comparing in the same way two different algorithms. That is why we use the fixed-length interval statistic in all cases.

3 Standard RTK Function: Float and Fixed RTK

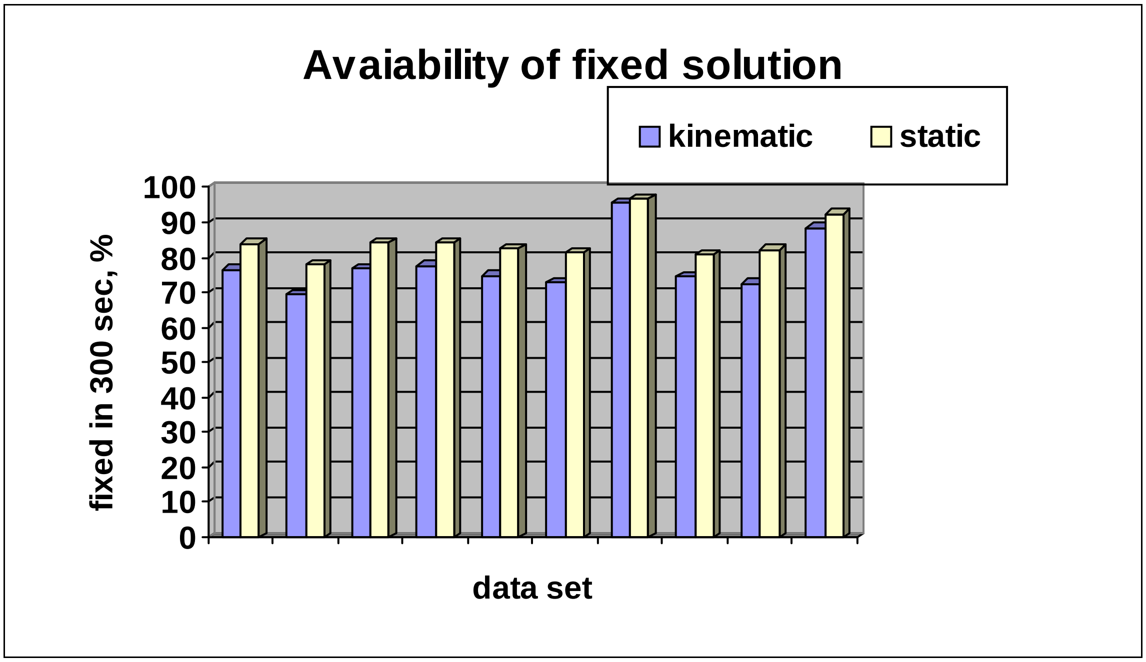
Standard functions in DG14 RTK include float and Fixed ambiguity RTK.

A vital statistic of Float RTK is the CEP convergence pattern *vs.* time elapsed after RTK initialization. The pattern below shows the convergence experienced with many days of data collected in different open and partly shaded environment (against DG14 base) on various baselines from tens of meters to 10 kilometers. Data being collected with static receivers were nevertheless processed in kinematic mode, i.e. no static assumption was done. The overall number of independent 600 sec trials exceeded 1000.



A vital statistic of Fixed RTK is the time required to fix the ambiguity (*i.e.* time to cm) provided that the result meets a pre-set reliability. To obtain time-to-centimeter figures we performed 10 open sky tests (>24 hours each) at different locations (against DG14 base) over various baselines from a few meters to 5 kilometers. For each test, RTK was reset each 300 seconds and the time to fix the ambiguity (time to cm) was measured. For each data set we have more than 250 independent 300-sec trials.

The diagram below shows the percentage of fixed trials (availability of a fixed solution) for each of 10 data sets. Pre-set reliability was 99%, and it was met. Data were processed with and w/o a static assumption.



4 Advanced Function: Flying RTK

To improve standard Float RTK performance, Magellan has integrated a so called Flying RTK solution into DG14 RTK.

4.1 Background: Float and Fixed RTK

.
The goal of Fixed RTK is to deliver centimeter accuracy as fast as possible after RTK start up insuring at the same time a preset reliability. The vital part of Fixed RTK is the correct determination of the carrier phase integer ambiguity without which centimeter accuracy cannot be achieved. Reliability refers to the probability of correct determination (fix) of the integer ambiguity. Usually L1&L2 RTK is fixed-ambiguity RTK, because dual-band data allow getting almost instant reliable centimeter accuracy in open sky conditions for baselines up to 20 km.

On the contrary, the goal of Float RTK is to insure position convergence to decimeter level as fast as possible. Float RTK never tries to fix the ambiguity, so there is no such thing as reliability. Usually, L1 RTK is float-ambiguity RTK, because single-band data (even with open sky and short baseline) cannot guarantee reliable and fast ambiguity fixing. The main disadvantage of Float RTK is the quite slow position convergence to decimeter level.

At the same time, from the point of the core algorithm, the difference between Float and Fixed RTK is not so noticeable. The core part of Fixed RTK is the Float RTK engine. If the Fixed RTK option is enabled, then additionally the special procedure (ambiguity search) runs each processing epoch. It searches for the most probable integer ambiguity candidate corresponding to the current float ambiguity estimate. If a given integer candidate does not pass all validation criteria, then the position stays float and the attempt repeats next epoch. If the integer candidate passes all validation criteria, the current float ambiguity can be fixed to (i.e. substituted by) the most probable integer estimate. If this integer estimate is correct, then RTK delivers centimeter-level position starting from this epoch. If this integer estimate is not correct, then the position error can be as large as a few meters. This is a wrong fix, which can last as long as ten minutes before the RTK engine understands its error.

There is an attractive algorithmic way to improve the convergence performance of the standard Float RTK solution. This is Flying RTK, which can be conceptually placed between standard Fixed RTK and Float RTK. We use word flying because:

- It starts with the same letter F as Fixed and Float
- It means that ambiguity is not constant as in Fixed mode
- It emphasizes that position convergence can be faster than in Float mode

4.2 Background: Flying RTK

Most of the known Float RTK engines treat (and estimate) double-difference (DD) carrier phase ambiguity as a **float** variable, constant with time until a cycle slip is detected. From the theoretical point of view this is incorrect, because it is known *a priori* that the DD carrier ambiguity actually can take only an **integer** value.

Most RTK engines are built as some recurrent data processor, which updates its current estimates each time new data are available. Usually it is a Kalman Filter or some other similar algorithm. With such an approach, the rationale behind considering an integer value as a float value is simple: It is much easier from the point of logic and computation to estimate a float variable rather than to estimate an integer variable. If fixing is needed (Fixed RTK), then the ambiguity search algorithm is called.

There exists a clear theoretical foundation to build a recurrent estimator of vector parameters some of whose states are not float but are integer by definition. It is a kind of adaptive, self-learning filter, which must contain as many parallel filters as needed, each tuned for its own integer ambiguity candidates. From the point of implementation such a multi-channel filter is quite useless. However, some of these theoretical ideas can be used to create Flying RTK without any noticeable additional computations and without changing the existing standard Float RTK processing logic.

On the one hand, the Flying RTK engine processes the ambiguity as float and never fixes it to an integer (i.e. it never can fix the wrong ambiguity); on the other hand it 'keeps in mind' that the actual ambiguity is an integer. The general idea of Flying RTK is to generate some position correction to the standard Float RTK position. The algorithm simply adds this correction and gets the Flying RTK solution.

The slogan of Float RTK is **"always process as float"**
The slogan of Fixed RTK is: **"first process as float, then process as integer"**
The slogan of Flying RTK is: **"always process as float, but always consider as integer"**

Finally, Flying RTK has all the attributes of Float RTK, such as:

- Flat position converged in time
- No wrongly fixed position

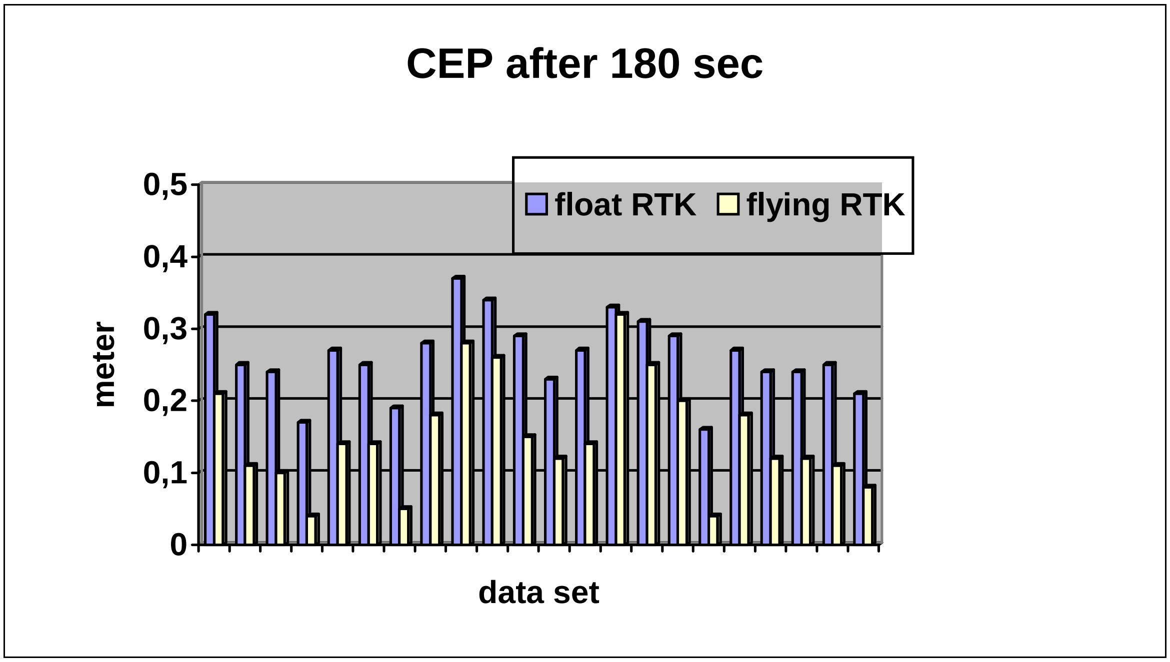
On the other hand, the convergence pattern and steady-state Flying RTK position is statistically better than the Float RTK position.

4.3 Performance of Flying RTK

Here we give test results which demonstrate Float RTK performance improvement after applying the Flying RTK technique. To make the correct comparison we collected raw data from a pair of DG14 receivers (base and rover). We collected the data on different days, with different antennae, for different baselines (10 km maximum) at different locations (US, France, UK, Russia). A total of 22 data sets were chosen. The overall size of the data set is more than 30 full days. Some data sets corresponded to open-sky conditions; and some corresponded to partly blocked sky.

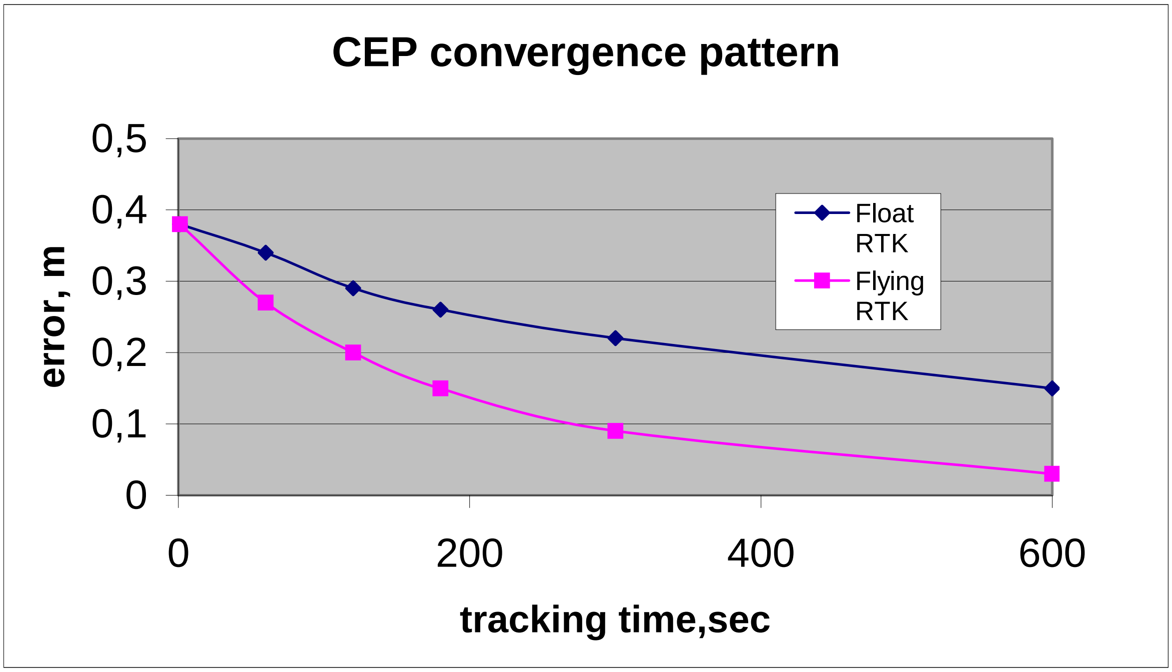
For each data set, a PC version of DG14 RTK was run in a real-time manner. First we ran standard Float RTK, then (with exactly the same data) we ran Flying RTK. In each case RTK was reset each 600 seconds and standard convergence pattern (CEP) was estimated *vs*. time after RTK initialization. In all the cases data were processed in kinematic mode, *i.e*. no static assumption was done.

The diagram below shows CEP experienced at the 180th second after initialization for each of 22 data sets. One can see about two times CEP improvement for each data set thanks to Flying RTK. In the most of the cases Flying RTK delivers sub-20cm CEP in three minutes after Flying RTK initialization.



The pattern below is plotted as the average of 22 patterns for each data set. It shows how CEP converges with time in average. It is seen that:

- Average time to converge to 20 cm is about three times faster with Flying RTK
- One can expect CEP to be less than 5 cm in 10 min after RTK initialization.



Due to the significant performance improvement of Flying RTK solution over standard Float RTK solution, Flying RTK mode supersedes Float RTK mode in the DG14 RTK receiver. It means that the user automatically gets the improved Flying RTK solution when the receiver is commanded to work in float mode.

5 Advanced Function: RTK with Moving Base

In many applications which require decimeter/centimeter accuracy one is interested not in the position itself, but in the relative position between two moving antennae (relative navigation). DG14 RTK allows such a possibility.

5.1 Background: Time-Tagged (TT) and Fast (FST) RTK operation

Two different RTK techniques are usually applied.

To have maximally accurate position the RTK engine must work in the so-called TT mode. To do this RTK must buffer receiver data and wait to decode time-matched correcting data. Since reference data can come with noticeable delay (seconds), then the resulting TT RTK position will be output with comparable latency.

To have position with minimum latency (independent of any data link delay), the so called Fast RTK is applied. Fast RTK uses reference data projected to the current receiver time tag. Due to this base data extrapolation, the accuracy of Fast RTK degrades proportionally to the projected-time interval. However, typically for short (a few seconds) projected-time intervals the inserted error is at the sub-centimeter level only.

The Fast RTK approach is valid only when the reference data are generated by a static receiver. Also, there is no need to send reference data faster than 1Hz when the base is static. The RTK rover can output position at higher rates using Fast RTK technology.

However, when the base is moving and one is still interested in an accurate baseline length estimate, RTK must take into account some specifics of this application.

5.2 Specifics of RTK with moving base

Standard rover RTK usually supposes that the base position is fixed and accurate. That is why base data usually are transmitted at a 1-Hz interval. Also, base position (unchanged in time) can be sent at a slower rate than the base measurement data. For example, the standard RTK base configuration sends measurements once a second, while the reference position can be sent once a minute. The final rover RTK position accuracy is equivalent to the final baseline accuracy and can be as accurate as decimeters for Float RTK and centimeters for Fixed RTK.

When the user wants RTK operation with a moving base, he or she is mainly interested in the baseline estimate rather than in the position estimate. If the base moves, then its position changes in time. To insure that the baseline estimate is as accurate as possible, the base position must be sent each epoch for which the base measurements are sent. Once the rover RTK estimates the baseline against the moving base position, then final rover position accuracy will depend on the transmitted base position accuracy. At the same time, even if the base position is not accurate, the base→rover baseline estimate is still accurate up to the centimeter-level when the ambiguity is fixed.

Standard rover RTK expects a static base. In this case, any change in the base position can be considered as a short or long term failure. It can also be caused by the provider changing the base station position. Usually standard rover RTK integer fix will most likely be reset upon detecting a change in the reference position. However, detecting a change in the reference position is normal when running RTK with a moving base.

To support RTK with a moving base, one must have the moving base itself and the RTK rover both process the moving reference base data adequately. Magellan provides both base and rover software to support this function.

The DG14 RTK base can be told to generate an internally computed position as the (moving) reference position with the same rate as the raw data rate (e.g. 5 Hz). The DG14 RTK rover can be told to consider a changing reference position as the normal (expected) case and not to reset the RTK operation whenever detecting a change in the reference position.

When the base is arbitrarily moving, its data cannot be extrapolated reliably even for 10-100 ms, which is why the Fast RTK technique cannot be applied with a moving base. Only the TT approach is applicable here.

5.3 RTK with moving base: expected performance

When working with a moving base, both Fixed and Flying RTK solutions are available. For baselines less than 20 km, even a few meters base position error does not insert any noticeable error into the baseline estimate. This means that Float/Fixed/Flying RTK performance are **potentially** the same for RTK with either static or moving base (referring to the baseline estimate, not the position estimate). This claim was checked and proved valid with some baseline data which were processed on a PC in both static-base and moving-base modes.

A moving base usually experiences worse GPS tracking, more frequent carrier cycle slips and loss of lock compared to a static base, which one tries to put under an open sky. That is why **actual** performance of an RTK rover working with a moving base may degrade because of poorer data quality coming from the moving base.

6 Advanced Function: Heading Determination

In some precise applications one is interested in attitude determination. In most of these cases, a full attitude solution (heading, pitch and roll) is not required as long as the heading is determined. DG14 RTK provides a heading determination function by enabling the Heading Determination option and using two antennae (and two receivers) mounted on a platform in line with the vehicle heading.

6.1 Specific of Heading Determination function

In general, baseline heading can be estimated for conventional RTK mode (including moving base). However, usually one is interested in vehicle heading, rather than in baseline heading. Standard RTK configuration for vehicle heading determination supposes:

- Base and rover antennae are installed on the vehicle body
- Base and rover receiver ports are connected directly by serial cable (*i.e.*, w/o radio)
- Base→rover baseline is fixed in the vehicle coordinate frame, i.e.:
  - Baseline length is fixed and known with sub-centimeter accuracy
  - Baseline elevation and azimuth against vehicle centerline are fixed and known with sub-degree accuracy

For such a hardware setup, the rover RTK can be put into Heading Determination mode, which allows for estimating vehicle (not baseline) heading for an arbitrary moving vehicle. If the baseline is aligned nearly along the vehicle body, then along with heading we can output vehicle pitch. If the baseline is aligned nearly across the vehicle body than along with heading we can output vehicle roll.

Unlike many existing heading determination systems, baseline length in DG14 RTK is not limited by a few meters and can be as long as tens of meters or more. The only requirement is that the baseline length is fixed and known with sub-centimeter accuracy. The accuracy of the heading and pitch or roll is a function of the antenna separation and improves linearly with every meter of antenna separation.

6.2 Specifics of Heading Calibration

To insure reliable and accurate vehicle heading determination, one must know:

- Baseline length between the two antennae
- Baseline azimuth and elevation offsets against the vehicle's centerline

Baseline length can be determined using GPS w/o any other information. In most existing attitude-determination systems, baseline-length calibration supposes collecting raw data from the base and rover receivers and post-processing in the office. However, with DG14 RTK one can make this calibration in real time and automatically save the calibration results in the receiver' Battery Backed-Up Memory. It is not obligatory for the vehicle to be static during the calibration process. Depending on environmental conditions, the baseline-length calibration process can take from minutes (open sky) up to an hour (noticeable shading). The calibration procedure is fully automatic, insuring 100% availability and reliability and sub-centimeter baseline length accuracy. Once calibration is finished, DG14 RTK automatically starts outputting heading.

DG14 RTK also allows the user to enter the baseline length manually if it is accurately known (sub-cm).

Baseline offsets require more care and are usually calibrated with the help of other tools. The DG 14 RTK software allows entering baseline elevation and heading offsets with the help of newly created serial commands. It is recommended to have the baseline elevation offset as close to zero as possible and to have the baseline azimuth offset as close as possible to a multiple of 90 degrees. If the azimuth offset is close to 0 or 180 degrees, then the vehicle heading and pitch will be estimated; if the azimuth offset is close to 90 or

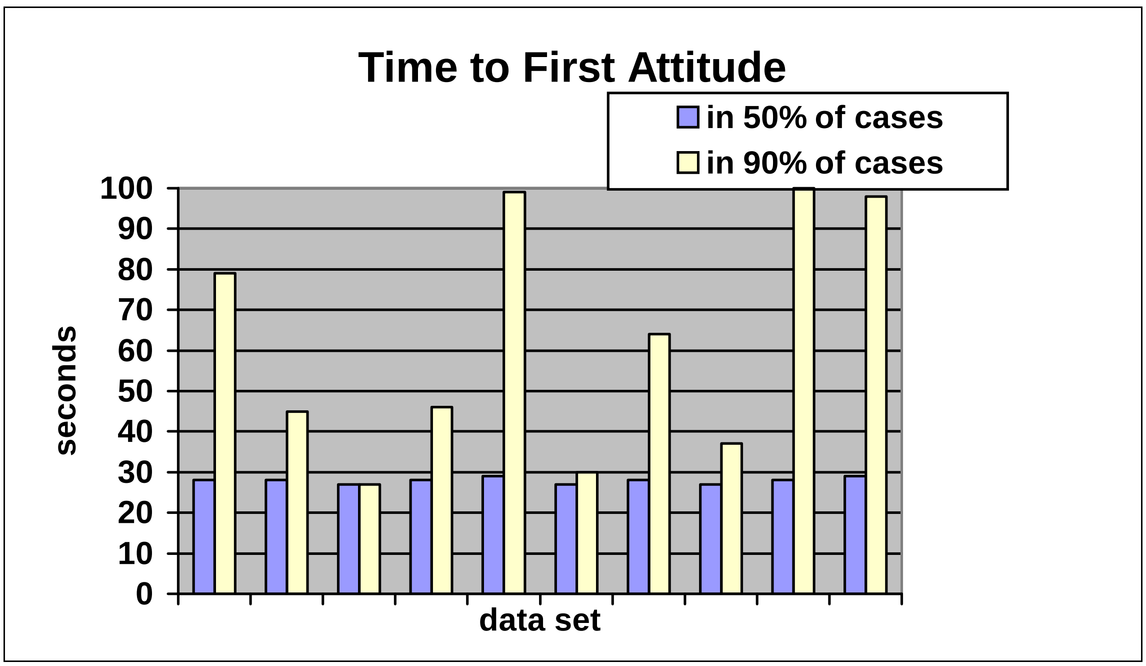
270 degrees, then the vehicle heading and roll will be estimated. It is recommended to have no more than ±5 degrees alignment error. Once the baseline elevation and azimuth offset are specified, DG14 RTK makes the needed re-computations and outputs heading and pitch (or heading and roll) referring not to the baseline, but to the vehicle body.

6.3 Performance of Heading Determination

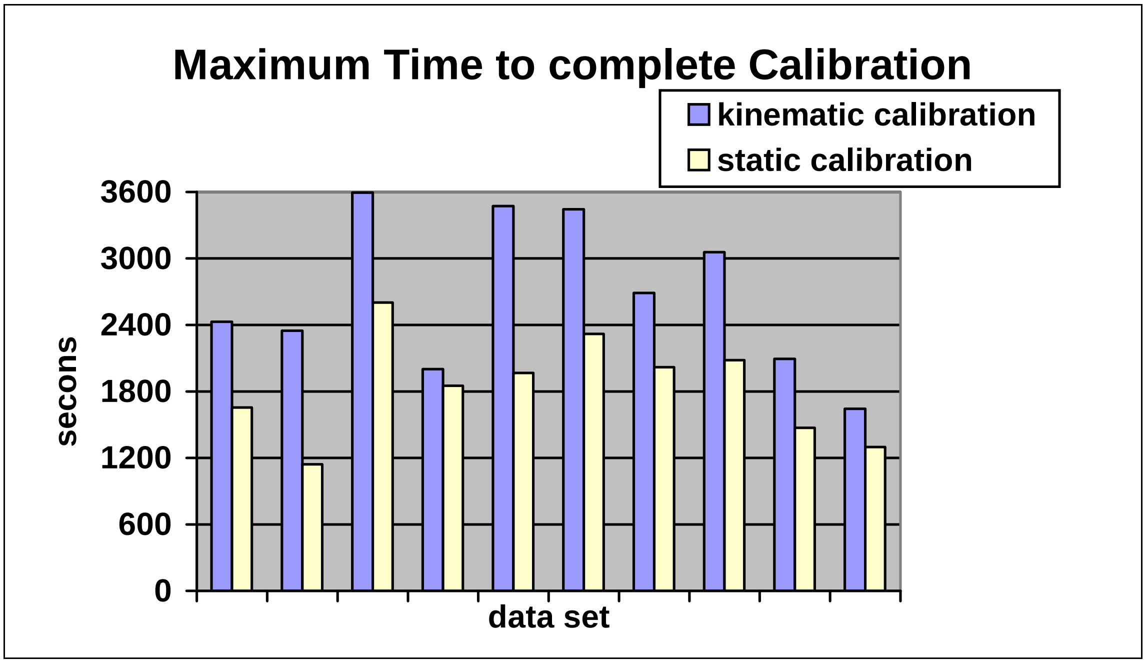
When speaking about heading determination, one is interested in sub-degree accuracy which can be achieved only if the carrier cycle ambiguities are correctly fixed (provided baseline between antennae is longer than about 0.5 meter). Therefore, vital to heading performance is the time needed to fix the ambiguity after Heading Mode initialization, *i.e*. Time To First Attitude (TTFA).

To estimate the TTFA statistic we took 10 sets of raw data (> 24 hours each) collected on different days, with different antennae, for different baselines (in ranges of 1-20 meters) at different locations. Baseline length for each data set was initially calibrated with mm accuracy. A PC version of the heading engine running in real-time mode was reset automatically each 300 seconds, and TTFA and reliability were measured. For each data set we experienced 280-600 independent initializations. The diagram below shows 50% and 90% estimates for TTFA for each data set. Pre-set reliability was set to 99%; however on average we experienced reliability close to 99.9%. One can see that TTFA is quite short (compared to conventional L1 RTK) thanks 2 factors:

- Short baseline
- Known length of baseline



Since the heading function in DG14 RTK allows self-calibration of baseline length, then the Time To complete baseline Calibration (TTC) is also important for the user. Calibration is usually performed only once (after antennae installation) and it is acceptable for this process to take quite a long time. At the same time reliability of the calibration must always be 100%. We used the same data sets as above to estimate TTC, by resetting the calibration process each hour. For each data set we experienced 24-60 independent calibrations. The diagram below demonstrated maximum-experienced (i.e. worst case) TTC for each data set when calibration was performed with and w/o static assumption. Reliability was experienced as 100% in all the tests. One can see that the worst-case TTC was 1 hour. Taking advantage of static calibration (if setup allows this) one can save about 50% of the time required for the complete calibration. However, static calibration is not always possible, e.g. antennae installed on floating vessel.



7 Advanced Function: GPS+SBAS RTK Engine

One of the advantages of the DG14 RTK solution from Magellan is that the DG14 RTK rover can work in GPS+SBAS mode. The DG14 board can track up to 2 SBAS satellites. It is important to note that an SBAS Satellite is not only the source of differential corrections. SBAS satellites also provide a GPS-like signal whose pseudo-range and carrier phase measurements can be used by DG14 RTK. Magellan's DG14 RTK can use SBAS measurements in RTK processing along with GPS measurements, thus making it a true GNSS system.

7.1 Specifics of SBAS Measurements Processing in RTK Engine

Formally speaking, SBAS measurements are equivalent to GPS measurements. The only principal difference is that the algorithms to compute the SBAS orbits are different.

Currently there are three working SBAS constellations:

- WAAS, which includes 4 Satellites covering North and South America and parts of the Atlantic and Pacific oceans.
- EGNOS, which includes 3 Satellites covering Europe and Africa and some nearby countries
- MSAS, which includes 2 Satellites covering Japan and parts of China.

In many areas one can see and track 2 to 4 SBAS satellites.

From the point of signal quality and maturity of orbital information, WAAS Satellites are good. On the contrary the EGNOS signal is not yet stable, and the accuracy of the provided orbital data is currently poor. That is why usage of SBAS (especially EGNOS) measurements in position computation is a challenge.

Fortunately many existing SBAS ranging disadvantages are mitigated when SBAS measurements are used in differential processing. At the same time, some of the negative effects still exist, and when processing SBAS measurements one must take care. The new GPS+SBAS RTK processing technique from Magellan not only uses SBAS ranging and carrier data, but also takes great care that a possible SBAS failure does not spoil RTK behavior.

Instead of ‘mechanical’ usage of SBAS ranging in the RTK processing, DG14 RTK has incorporated the following 3 principal innovations:

- Adaptive SBAS usage
- SBAS data calibration
- SBAS tracking synchronization

In many cases (especially with EGNOS), SBAS data can be bad and under no circumstances must be taken into the RTK processing engine. Adaptive SBAS usage means detecting wrong SBAS measurements and/or orbit and stopping their usage.

SBAS measurements (especially when base and rover data are provided by different receiver types) can have biases which must be accounted for. A special robust procedure estimates the possible SBAS biases in real time and compensates for them in the RTK processing.

Usually a receiver is equipped with only 2 channels to track SBAS (*e.g*., this is the case of DG14), *i.e*. it is not an all-in-view SBAS receiver. In many cases up to 4 SBAS satellites can be seen, so it is desirable to track in the rover those SBAS satellites for which the base transmits data. Such an algorithm has been implemented in DG14 RTK, which allows in many cases escaping asynchronous base and rover SBAS tracking.

The GPS+SBAS RTK technique is similar to GPS+GLONAS L1 RTK technique, which was also invented by Magellan (formerly Ashtech) in 1996. From the point of view of RTK performance, two extra SBAS Satellites do the same job as three extra GLONASS Satellites. With the currently incomplete GLONASS constellation, the ‘power’ of L1 GPS+SBAS RTK and L1 GPS+GLONASS RTK is approximately the same.

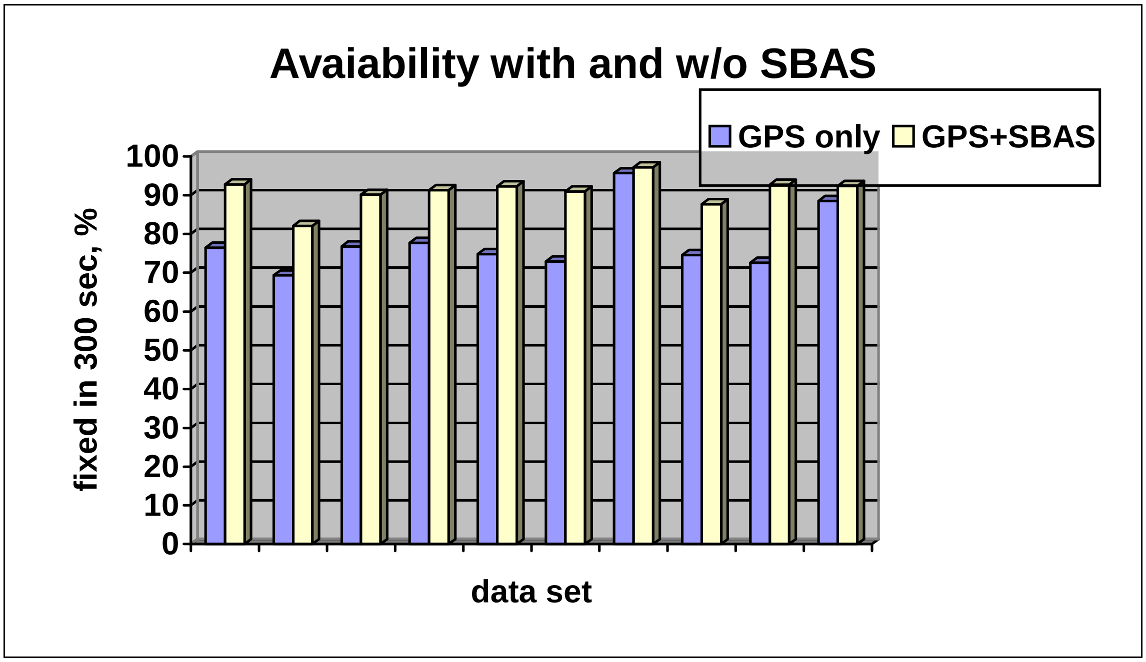
7.2 Transporting SBAS Measurements

Obviously, to enable GPS+SBAS RTK processing, a base station must send SBAS data. This is possible if the base sends data in RTCM 3.0 format where room for SBAS data is reserved. DG14 RTK rover supports this RTCM 3.0 protocol. Note that the DG14 RTK base does not support RTCM 3.0, so it is not currently possible to broadcast SBAS corrections from a DG14 Base to a DG14 rover. Also most of the existing bases supporting RTCM 3.0 do not generate SBAS measurements. However, Magellan offers the ProMark3 entry-level survey receiver that supports the RTCM 3.0 base protocol including the SBAS measurements. Thus it is possible to set up a ProMark3 base and a DG14 RTK rover and take advantage of the GPS+SBAS processing. Working with an RTCM 3.0 base the DG-RTK rover can take advantage of running GPS+SBAS RTK. In the USA this will give up to two extra GPS-like satellites which can noticeably improve DGRTK performance.

7.3 RTK Benefits due to Using SBAS Measurements

The plot below demonstrates the improvement when using SBAS measurements along with GPS measurements.

Ten data sets for open-sky baselines from a few tens of meters to 5 km were used. One or two common SBAS satellites were available to both base and rover. Fixed RTK mode was evaluated. One can see that availability of fixed solution at 300 sec interval is about 15% higher due to the addition of SBAS measurements.



8 Conclusion

DG14 RTK uses Magellan's BLADE, which includes state-of-the art algorithms for processing single-frequency GPS+SBAS data. This new product offers a more rapid time to decimeters than competing products with the new Flying RTK algorithm. It can be used for heading and pitch (or roll) determination, and allows RTK from a moving base. It also fixes ambiguities in shorter times than standard L1 RTK engines owing to the unique way in which Magellan processes both the GPS and SBAS signals. The addition of SBAS to the GPS makes L1 RTK positioning practical, thus lowering the barrier of entry to RTK positioning created by the need for dual-frequency hardware.