



**Technology Paper**

Flying™ RTK Solution as Effective  
Enhancement of Conventional Float RTK

# Flying™ RTK Solution as Effective Enhancement of Conventional Float RTK

Dmitry Kozlov, Gleb Zyryanov  
*Magellan, Russia*

## BIOGRAPHY

Dmitry Kozlov has been working in Magellan since 1993 as a Senior Scientist. Since 2002 he is Algorithm Group Manager. He received his MS EE degree from Moscow Aviation Institute and his PhD in Signal Processing Theory from the Institute of Automatics, Moscow.

Gleb Zyryanov has been working in Magellan since 2000 as a Software Engineer. Since 2006 he is Senior Software Engineer. He received his MS in Mathematics and Mechanics from Moscow State University.

## ABSTRACT

Float RTK is used to deliver sub-meter to decimeter accuracy for various L1 and L1&L2 RTK applications. Typically Float RTK insures sub-meter accuracy with start-up and decimeter level accuracy after few (or few tens) minutes. For given hardware, the RTK convergence time and steady state accuracy are subject of local environmental conditions (shading, multipath), baseline length and RTK algorithm itself.

In given paper, we describe new RTK algorithm from Magellan: Flying RTK. Being quite a simple in realization and not so time consuming, it demonstrates statistically better performance compared to standard Float RTK. While Flying RTK algorithm can be applied to both L1 and L1/L2 systems, in given article we make accent on L1 RTK systems. We give apple-to-apple performance comparison between Float RTK and Flying RTK with the data collected in different environments and baselines. The results prove that CEP convergence to decimeter accuracy can be achieved 1.5-5 times (depending on conditions) faster when applying Flying RTK algorithm instead of Float RTK

The data used for validation were collected with different Magellan receivers.

## INTRODUCTION

The goal of Fixed RTK is to reach 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 about 20 km (for longer baselines initialization can last up to tens minutes).

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 a 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 [1]. 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 unlike in Fixed mode
- ❑ It emphasizes that position convergence can be faster than in Float mode

Flying RTK algorithm has been implemented in latest Magellan L1 RTK products:

- DG RTK – L1 GPS+SBAS OEM board supporting RTK against static and moving base as well as heading function with baseline auto-calibration
- ProMark3 RTK – L1 GPS+SBAS handheld supporting real time Surveying and Mobile Mapping functions with different connectivity including NTRIP and Direct IP Network connection.

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

## ALGORITHM

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 (KF) 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 how 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 [2],[3], 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 slogans of Float/Fixed/Flying RTKs can be summarized as follows.

- Float RTK: **“always process as float”**
- Fixed RTK: **“first process as float, then process as known integer”**
- Flying RTK: **“always process as float, but always consider as unknown 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.

The Flying RTK algorithm idea can be explained based on KF approach. KF updates its state vector  $X$  and associated state covariance matrix  $P$  each time new pseudo-range and carrier phase data (new epoch) available. Typically for Float RTK applications, state vector  $X$  includes three groups:

- Position & velocity
- DD ambiguity
- Others (e.g. ionosphere, troposphere residuals)

Float RTK processes all these states in one KF using standard update (measurement update) and project (time update) procedures (Fig.1).

If fixed ambiguity option is enabled (Fixed RTK) then  $X$  and  $P$  are both used to construct DD ambiguity search space (Fig.2). Integer ambiguity search is called and its results are validated in correspondence with some statistical criteria. If integer candidate is recognized as reliable, then ambiguity is fixed in which case original state vector  $X$  and associated covariance  $P$  are modified to

reflect new status of DD ambiguity: known integer. Please note that this modification is done inside KF loop and modified X and P are used to project KF state for the next epoch.

It is seen that Fixed RTK algorithm applies 3 different modifications for X and P:

- KF project (each epoch)
- 'Regular' KF update (each epoch)
- 'Occurrence' KF update (after fixing ambiguity)

This 'occurrence' update is what which results to either:

- Cm level position error (if fix was correct)
- Meter level position error (if fix was incorrect)

for sufficiently long time after ambiguity fix.

Note that usually position error is not noticeably changed after regular update, while can be changed by meter level after 'occurrence' update. Position error jump (even thanks to correct ambiguity fix) is what many users do not like. Flying RTK can potentially reach the same accuracy as fixed RTK, but usually does not experience any position jump.

Flying RTK never makes 'occurrence' update, because there is no occurrence related with ambiguity fix. Standard KF principle (update/project) used in Float RTK is saved in Flying RTK as well. The Flying RTK correction is made outside KF loop (Fig.3). The primary idea to compute Flying RTK correction is calling DD ambiguity search based on current X and P. However, unlike Fixed RTK, Flying RTK uses search results in other way. While Fixed RTK is trying to validate the reliability of most probable integer candidate, Flying RTK is trying to build some 'integer ambiguity field' having found a number M of most probable integer vectors (value M can vary from 1 to 100). These M vectors and associated chi square statistic are used to generate Flying RTK correction to be applied to Float position. Since this correction can be quite a noisy, we apply some sort of smoothing of this correction during some appropriate time (e.g. 1 minute). Given smoothed correction is usually flat which guarantees finally flat Flying RTK position.

The qualitative considerations above can be generalized to any basic RTK (not obligatory KF) principle.

The tentative plot below demonstrates typical position error as function of cumulative tracking time for RTK modes under the issue.

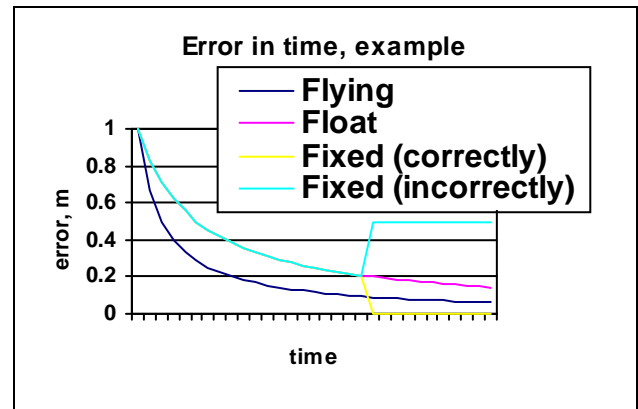


Figure 4. Tentative convergence pattern for different RTK techniques

## PERFORMANCE

Here we give test results which demonstrate Float RTK performance improvement after applying the Flying RTK technique. We have chosen apple-to-apple statistic comparison. Apple-to-apple means that we evaluated Float and Flying RTK algorithm against exactly the same GPS measurements. This means that we used PC version of the algorithm running with pre-collected raw data. Since the PC version is built using exact source code from receiver firmware, this PC derived results 100% correspond to what one will have in a receiver. Statistic approach means that we used a huge number of data to derive statistically sufficient estimates. All the tests results given below were obtained by the same procedure. The algorithm was reset each 600 seconds and all statistical figures were computed by processing sufficient number of independent 600 sec trials.

In all the cases data were processed in **kinematic** mode, *i.e.* no static assumption was done. All the performance figures are expressed in terms of CEP (*i.e.* 50% horizontal error bound) vs. time elapsed after algorithm initialization.

To demonstrate the performance we selected the following 3 cases:

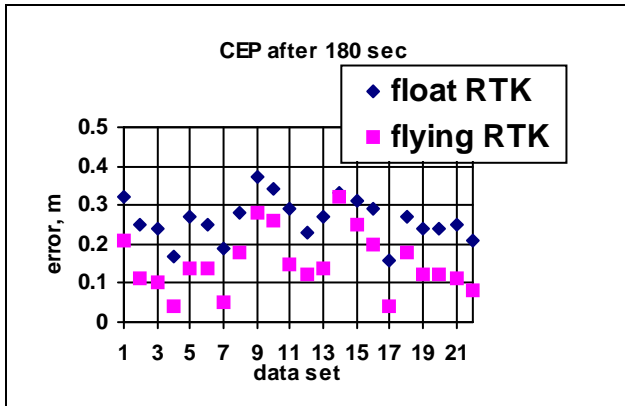
- Data collected with DG RTK receivers covering different baselines from few meters to 10 km in different environment (from open sky to shaded).
- Data collected with Zmax receivers covering open sky baselines from 7 to 50 km.
- Data collected with ProMark3 receiver against Leica Spider NTRIP Network at open sky baseline 10 km.

### A. DG RTK raw data

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.

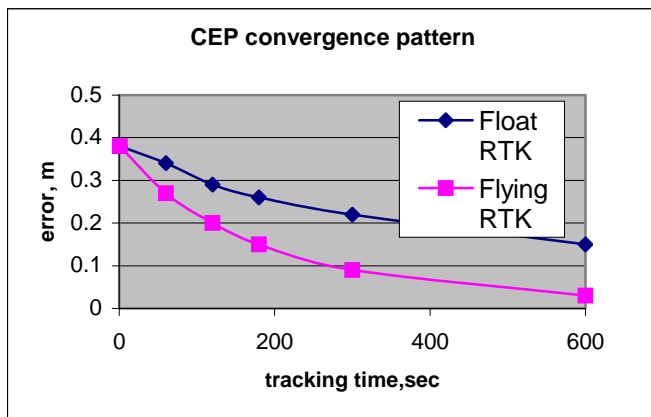
The diagram below shows CEP experienced at the 180th second after initialization for each of 22 data sets. One can see about two times CEP improvement for each data set thanks to Flying RTK. In the most of the cases Flying RTK delivers sub-20cm CEP in three minutes after Flying RTK initialization.



**Figure 5. Accuracy after 3 minutes after initialization for different data sets**

More detailed analysis has shown that Float and Flying RTK performance is mainly environment (shading, multipath) dependent rather than baseline length dependent.

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



**Figure 6. Accuracy convergence vs. time after RTK initialization**

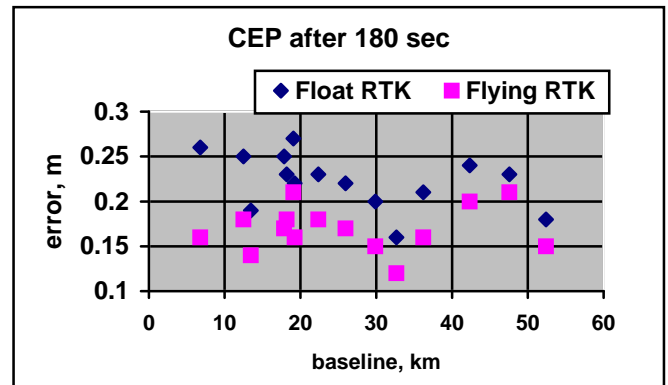
We conclude that:

- In most of the cases one can achieve 20 cm CEP in 3 minutes applying Flying RTK
- Average time to converge to 20 cm CEP can be about three times faster with Flying RTK
- One can expect CEP to be less than 5 cm in 10 min after RTK initialization.

### B. Zmax raw data (L1 only)

To see what one can expect for long baselines, we performed the following recording. Six Zmax receivers were installed on different open sky points and GPS 1Hz raw data were recorded simultaneously during about 2 days. As a result we got  $6*5/2=15$  baselines covering the ranges from about 7 to 52 km. All the baselines were preliminary post processed with resulting sub-cm accuracy. To estimate the convergence we had about  $N=48*3600/600=288$  independent 600 sec trials for each baseline.

While Zmax is L1/L2 receiver, for evaluation we used only L1 CA portion of data, i.e. run Float and Flying RTK algorithms with L1 CA data only. The plot below shows CEP for Float and Flying RTK for all 15 baselines. All the figures correspond to 180 sec (3 minutes) cumulative RTK processing.



**Figure 7. Accuracy after 3 minutes after initialization vs. baseline length**

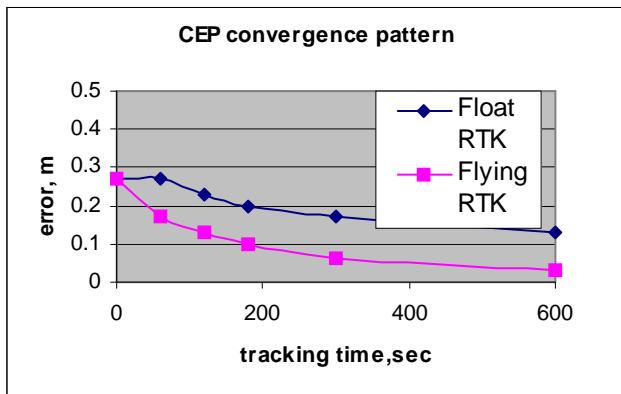
One can make the following conclusions:

- Float convergence behavior is quite the same for baselines varied from 10 to 50 km provided equal environmental conditions.
- Flying RTK algorithm can improve Float RTK performance even for long baselines.

### C. ProMark3 RTK raw data against NTRIP Network

These data were collected simultaneously with ProMark3 RTK receiver working in NRIP rover mode with single reference station in Leica Spider Network near Nantes (Orpheon Network), France. Network data are RTCM-3

Type 1004 stream. Baseline is about 10 km. Data recording lasted 46 hours. Convergence patterns are plotted below.



**Figure 8. Accuracy convergence vs. time after RTK initialization on with Network Ntrip data**

One can see that working with 3<sup>rd</sup> party Network data Flying RTK technique insures noticeable performance enhancement compared to Float RTK.

## CONCLUSIONS

There are many applications requiring sub-foot kinematics position accuracy. This accuracy cannot be achieved by standard DGPS technique, so to meet it one requires more powerful RTK processing. At the same time in many cases using L1/L2 RTK systems is not possible for price reasons.

In this case, L1 RTK solution can be a very good compromise between accuracy requirement and price. With L1 RTK one potentially can get even cm level accuracy by enabling Fixed ambiguity RTK. However this cm accuracy not always can be reached in reasonable time. Fixed ambiguity L1 RTK is limited in range (about 10 km maximum) and very sensitive to local environmental conditions like shading and multipath, which can make ambiguity fix impossible in many situations. Also for L1 RTK is not so ease to guarantee absolute reliability of fixed ambiguity solution.

That is why many users fix their minds on Float L1 RTK systems. Depending on baseline length and (mainly) local environmental conditions, such a system can deliver sub-foot accuracy, but user can be not satisfied with the time it takes to reach sub-foot.

New Flying RTK technology from Magellan is oriented on such users. The performance figures evaluated in given paper demonstrate noticeable improvement in L1 Float RTK performance when applying Flying RTK algorithm.

Flying RTK algorithm is integrated into 2 lately released L1 products: DG RTK OEM board, and ProMark3 RTK Surveyor. Each product can be commanded to work either in Float mode or Fixed mode. Being in Float mode receivers nevertheless output Flying RTK position marked as Float.

DG RTK can be used for many applications including machine control and agriculture. Aside 'standard' RTK function, it can support RTK against arbitrary moving base station. Given mode is used in some machine control applications. Flying RTK solution is applied when working with moving base which results to more accurate baseline estimate between moving receivers.

ProMark3 RTK Surveyor is the first handheld RTK product at quite a low price. Its fixing performance are limited by about 10 km baseline, however for longer baselines it provides accurate dm level solution applying Flying RTK technique.

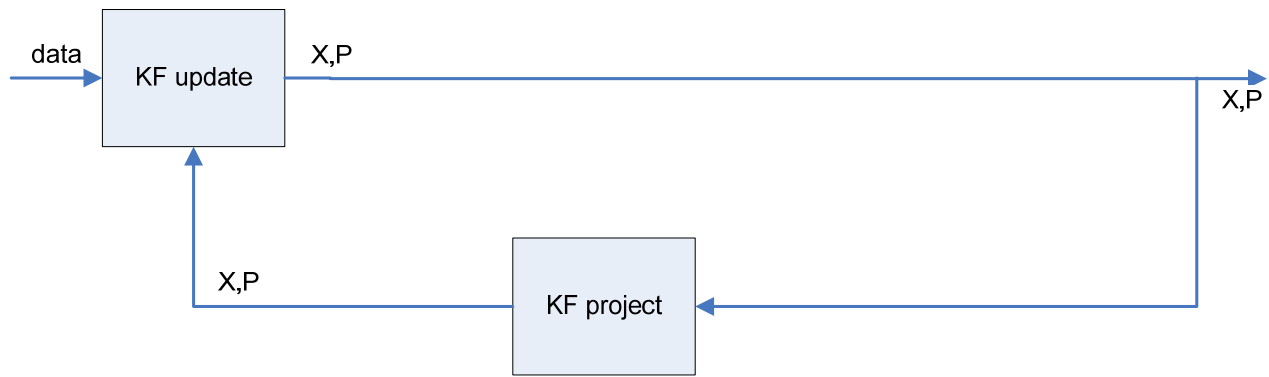
Concluding, Flying RTK technique is the solution which is acceptable for many applications requiring sub-foot accuracy. It joins positive advantages of Fixed and Float RTK, escaping at the same time their possible disadvantages. From point of performance Flying RTK can be placed between Fixed and Float RTK giving thereby to user one extra solution choice.

## ACKNOWLEDGMENTS

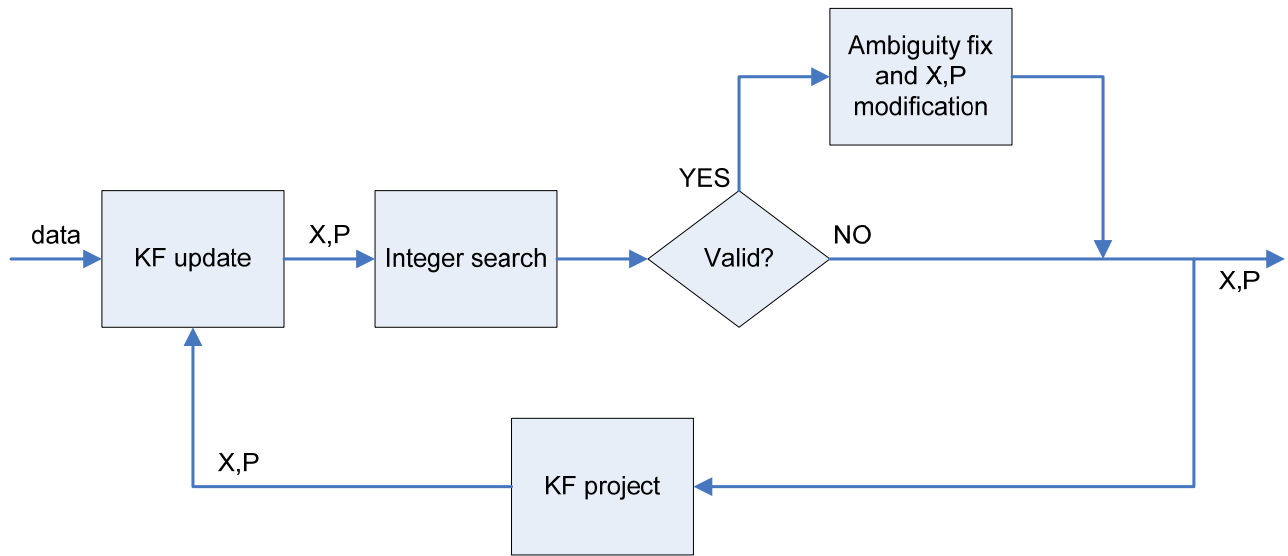
Authors would like to thank Magellan System Test group for their careful testing and validation efforts with release DG RTK and ProMark3 RTK. Our personal thanks for valuable help with data collection and performance tests are to Yves Le Pallec, Eugeny Sunitsky (all Magellan), and Bill Cottrel (Cottrel Navigation Services).

## REFERENCES

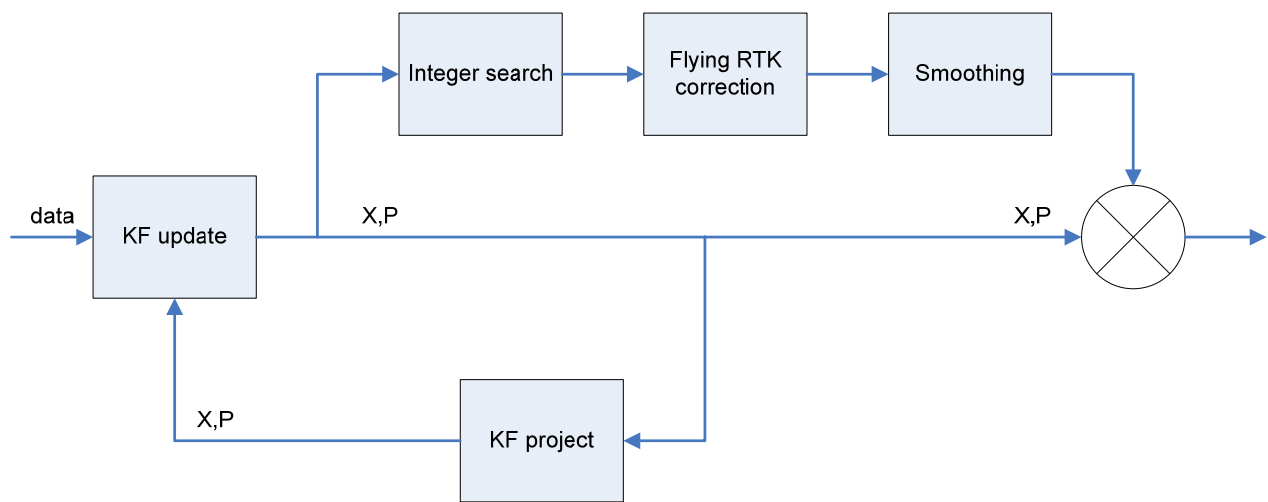
- [1] Leick. A., *GPS Satellite Surveying*, John Wiley & Sons, Inc., 1995, 2<sup>nd</sup> ed.
- [2] D. Magill, *Optimal Adaptive Estimation of Sampled Stochastic Processes*, IEEE Trans. on Automatic Control, AC-10, No.4, 434-439, October 1965.
- [3] R. Brown, Y. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, John Wiley & Sons, Inc., 1995, 2<sup>nd</sup> ed.



**Figure 1. Float RTK diagram**



**Figure 2. Fixed RTK diagram**



**Figure 3. Flying RTK diagram**