Software-defined Radio GNSS Receiver Signal Tracking Methods

A A Kumarin¹ and I A Kudryavtsev¹

¹ NIL102, Samara University, Samara, Russia

Abstract. Software-defined radio (SDR) - based GNSS receivers are flexible. SDR technology allows changing processing algorithms and adding new signals without any hardware modifications. The main drawback of such a receiver is high computation load. Signal tracking module takes most part of CPU time. This paper considers distributed receiver architectures where most of the processing is performed in a remote PC or a server. The amount of raw data in the SDR receiver is too large for complete transfer. Two methods of reduction are discussed: truncating epochs and disabling tracking for a certain period. The limitations of each method are estimated using simulations. Epochs can be truncated by 50-75% depending on signal-to-noise ratio. The maximum disabling time is about 0.5s for a stationary receiver and 0.1s for a car receiver. Using both methods in combination reduce data volume by 95.4% for a stationary receiver and 83.3% for a car receiver.

1. Introduction

SDR-based global navigation satellite systems (GNSS) receivers are popular nowadays. The main difference in comparison with traditional receivers based on application-specific integrated circuits (ASICs) is that signal is mostly processed in software. Therefore, processing algorithms can be modified without changing hardware. This allows adding new types of signals and processing algorithms without changing any hardware. In addition, higher accuracy can be achieved with simpler hardware. Typical SDR-based GNSS receiver consists of an antenna, a simple radio frequency (RF) frontend and a computational unit. The antenna is usually an active patch antenna. RF front-end moves signal to intermediate frequency (IF) where it can be directly digitized by an ADC. Obtained samples are processed in a computational unit or stored in some data storage.

The data processing usually requires 3 stages: signal acquisition, signal tracking and position computation. On the first stage signals from all possible satellites are searched for and their parameters are estimated. That is a heavy computation burden, however, it is executed once during startup and relatively rarely afterwards to find new available signals or to reacquire lost ones. The second stage is started right after the first one and lasts until signal is lost or the receiver is shut down. Its main purpose is to keep track of the signal: its carrier frequency, code delay and extract navigational data bits. The third stage is executed upon need. Usually it’s executed periodically, e.g. once per second. Its main purpose is to calculate receiver position.

The problem of software data processing approach is heavier computational load and thus, need of a more sophisticated computational unit. In most cases the data are postprocessed by a PC slower than needed in a real-time mode. This limits possible areas of SDR GNSS application. For example, they are not suitable for IoT applications where real-time mode and low power consumption are preferred.

As described in previous research [1], the tracking stage of a software receiver is the most time-consuming. Moreover, normally this load can’t be reduced, because satellites should be tracked all the...
time in order for the receiver to be able to calculate its position. Thus, some architectural solutions should be made to reduce tracking algorithm computation time. One of the approaches uses special platforms such as FPGA+PS systems-on-chip [2]. However, it doesn’t completely solve the problem of power consumption and cost. This paper is focused on simplifying end-user devices in order to achieve full advantage of SDR technology without using sophisticated computational units.

2. Possible approaches
One of the possible approaches of reducing cost, size and power consumption of such a system is considering distributed system architecture (Figure 1). Normally data is processed by some portable computational unit (DSP, FPGA, microprocessor or any combination) or PC. Alternatively, the data can be stored for later processing. However, if a system has suitable data transfer channel, the raw data can be transferred to a remote computational system. It can be a remote PC dedicated for the task or a specialized server. Computations can be shared between local and remote computational units [3] or a remote-only solution can be used.

The main problem of this approach is the need of a stable high-speed data link. For example, if ADC sampling rate is 53 MSPS with 2bit resolution one needs minimum data rate 106 Mbit/s. This is a high network load and can only be achieved in Gbit Ethernet networks. Therefore, such approach is only applicable for a simple and short network or a direct PC connection.

Thus, reduction of the data volume is a crucial task. There are two possible ways to solve it: compressing data before transfer or partial data usage. Compressing SDR data can be done using different algorithms. Testing has been done using several popular algorithms on 10kBytes data chunk. The compression results are shown in table 1.

| Compression method                                      | Result file size, B | Compression ratio, % |
|--------------------------------------------------------|---------------------|----------------------|
| Deflate Normal, 32KB dictionary, 32b word size          | 9 167               | 89.52                |
| Deflate Ultra, 32KB dictionary, 128b word size          | 9 158               | 89.43                |
| LZMA2 Fast, 1MB dictionary, 32b word size, 128MB block size | 9 466               | 92.44                |
| LZMA2 Fast, 1MB dictionary, 32b word size, 4GB block size | 9 356               | 91.37                |

It is evident that, compression does not have significant effect. This is mostly due to the nature of the data, which is close to white noise.

Another method is partial data usage. This can be done in several ways. Using the first one implies truncating each data block (for example to 512 chips instead of original size of 1023 chips [4]). Tracking performance greatly depends on noise conditions. Poor SNR leads to lower correlation peaks. However, that is not a problem for strong signals. Figure 2. shows cross-correlation for strong signal with different
PRN length: 75% (b), 50% (c), 25% (d), 10% (e) of an epoch (1023 chips) compared to full epoch (a). Both correlation and lag are normalized to the PRN length for better comparison.

**Figure 2.** Cross-correlation of ideal PRN with modelled noisy PRN with different length for strong signal. a) full epoch, b) 75%, c) 50%, d) 25%, e) 10%.

Figure 3 shows an example of disappearing peak when PRN length is reduced.

**Figure 3.** Cross-correlation of ideal PRN with modelled noisy PRN with different length for average signal. a) full epoch, b) 75%, c) 50%, d) 25%, e) 10%.

Weak signals may require full length to have a distinguishable peak, like shown in figure 4.

**Figure 4.** Cross-correlation of ideal PRN with modelled noisy PRN with different length for low signal. a) full epoch, b) 75%, c) 50%, d) 25%, e) 10%.
In another method of dataflow reduction tracking is disabled entirely for some time and resumed it again afterwards. The main problems of such approach are carrier frequency drift, growing code misalignment and incorrect filter operation.

3. Estimating Doppler frequency
Another technique that reduces computation load of signal tracking is complete tracking disabling for a more significant time. Signal is tracked using N1 epochs and after that N2 epochs are skipped. Signal parameters can be extrapolated (Figure 5 (b)) or assumed to be constant (Figure 5 (a)). Then tracking is resumed for N1 epochs again and the cycle is repeated.

Figure 5. Tracking signal with partial tracking disable with constant parameters model (a) and linear extrapolation (b).

Such approach requires careful selection of N1 and N2. Common tracking algorithm follows changing frequency and time delay of incoming signal of an acquired satellite. To select an appropriate value of N2 an estimation of frequency changing rate is needed. For that purpose, a model has been created and simulated.

The first step of the simulation is orbit modelling using required GNSS orbit parameters: inclination, altitude. The eccentricity is neglected. The right ascension of the ascending node (RAAN) \( \Omega \) is varied from 0 to \( 2\pi \) with a certain step. Orbit is considered to be a circle (\( e = 0 \)) with the semimajor axis \( r = 26.56 \text{ km} \), inclination \( i = 55^\circ \) – standard GPS orbit. The argument of latitude \( u \) changes over time linearly. Cartesian coordinates of each modelled orbit point is calculated using the following commonly used formula:

\[
R_{ES} = (h + R_e) \begin{pmatrix}
\cos u \cos \Omega - \sin u \sin \Omega \cos i \\
\cos u \sin \Omega + \sin u \cos \Omega \cos i \\
\sin u \sin i
\end{pmatrix}
\]

(1)

where \( R_e \) – radius of the Earth.

The velocity of a satellite is calculated using commonly used formula:

\[
v = \sqrt{\frac{\mu}{R_e + h}} \begin{pmatrix}
-\sin u \cos \Omega - \cos u \sin \Omega \cos i \\
-\sin u \sin \Omega + \cos u \cos \Omega \cos i \\
\cos u \sin i
\end{pmatrix}
\]

(2)

where \( \mu \) – gravitational parameter of the Earth.
At the second step the parts of the orbits visible to a receiver are calculated. The receiver was considered to be placed at zero longitude and variable latitude from 0 to $\pi/2$. Combining it with variable RAAN gives all possible relative positions of the satellite related to the receiver. A positive dot product of normal vector to the Earth in the receiver’s position ($\mathbf{R_{ER}}$) and vector from the receiver to the satellite ($\mathbf{R_{RS}}$) was used as a visibility criteria (3). This corresponds to zero elevation angle for the receiver. In reality, practical elevation angles are about 5–10°. However, this paper considers worst case scenario. Modelling gives a number of orbit parts like shown in figure 6 for receiver latitude 30° and zero RAAN.

\[
\mathbf{R_{ER}} \cdot \mathbf{R_{RS}} > 0 \tag{3}
\]

Next step is Doppler frequency computation. For that purpose, dot product of normalized satellite-receiver vector ($-\mathbf{R_{RS}}$) and satellite velocity ($\mathbf{v}$) are calculated. The final Doppler frequency is obtained multiplying the velocity by radio frequency ($F_{L1}$) divided by speed of light ($c$).

\[
F_{\text{Dopp}} = \frac{-F_{L1}}{c} \left( \frac{\mathbf{R_{RS}} \cdot \mathbf{v}}{||\mathbf{R_{RS}}||} \right) \tag{4}
\]

At the final step time derivative of the Doppler frequency is calculated. The results for afore-mentioned orbit are shown in Figure 7 and Figure 8.

\[\text{Figure 6. Visible orbit}\]

For each combination of RAAN and receiver latitude maximum and minimum change rate have been found. The vertical axis of the surface plots on figures 9 and 10 is maximum Doppler change rate (DDopp), horizontal – RAAN and receiver latitude. The plot is shown in figure 10 for maxima and in figure 9 for minima. It has to be noted that surface for minima is less smooth due to the fact that minima
correspond to minimal elevation angles which are not exactly equal to zero, depending on parameter step and can vary from orbit to orbit. That is a modelling effect with no real physical basis. However, one can see the general tendency. In addition, both surfaces have empty areas for orbits with zero visibility.

It can be seen that Doppler frequency change rate is always less than 1 Hz/s. The overall maximum is about 0.937 Hz/s, overall minimum – about 0.007 Hz/s. The surfaces actually touch each other. It happens at about 0.2 Hz/s. The surfaces show expected Doppler frequency changing rate for a specific receiver and a specific satellite. For example, a polar receiver has almost flat curve. The maximum expected change rate is about 0.71 Hz/s. The greatest variation is observed for an equatorial receiver.

4. The maximum tracking disabling time assessment

Calculated maximum Doppler change rate gives an estimation of maximum time interval, during which tracking can be suspended without following re-acquisition. One of the main limitations is phase difference accumulated during the time when tracking is disabled. Phase difference should not exceed 45° when data is present and 90° when no data transfer occurred during the interval [5]. Estimation of phase changes requires phase over time model. In this paper constant frequency change rate was used:

\[ \dot{\omega} = \omega_0 + \dot{\omega} t, \quad \dot{\omega} = \text{const} \]  

Integrating the equation gives the phase model:

\[ \varphi = \varphi_0 + \omega_0 t + \frac{\dot{\omega} t^2}{2} \]  

Therefore, the difference is:

\[ \Delta \varphi = \dot{\omega} \frac{t^2}{2} = \pi \nu t^2 \]  

This gives phase difference over disabled tracking time for different frequency changing rates (Figure 11). In addition, guaranteed divergence time has been calculated over expected frequency changing rated (Figure 12). It can be seen that even without other disturbances tracking disabling leads to guaranteed divergence after 0.5-7 seconds.
5. **Moving receiver considerations**

The aforementioned numbers are computed for a stationary receiver. Additional considerations were taken into account for a moving receiver. The worst case scenario is when a receiver moves in GNSS satellite orbital plane with speed vector parallel to the direction from the receiver to the satellite (Figure 13). In this case Doppler frequency is maximal by absolute value. At the same time, in this scenario elevation angle is minimal, thus stationary effects described in previous paragraph can be neglected. Doppler change rate can be estimated using the following formula:

\[
F_{\text{Dopp Move}} = \frac{F_{11}}{c} v_c
\]  

and its change rate is given by formula:

\[
\frac{d}{dt} F_{\text{Dopp Move}} = \frac{F_{11}}{c} \dot{v}_c
\]  

Considering previous results, divergence time dependence over receiver acceleration is shown in figure 14. The active acceleration of 3 m/s² was used as a practical maximum for a car [6]. Another
reason why considering higher accelerations is not practical is that they also affect network connection and the system is unable to operate. Considering both satellite motion and receiver acceleration results in a more complex dependence. However, maximum Doppler frequency change rate is a rare case, therefore these plots can be used independently.

6. Discussion
Compression test shows that due to the nature of data it is not practical for SDR GNSS. The 10% reduction of the data size does not change the need of Gbit Ethernet and significant network load. Therefore, full data transfer is only applicable for direct PC connection.

On the other hand, truncating epochs seems a better solution. It can drastically reduce the amount of data used for tracking (by 50-75%) without losing loop lock. However, this approach has positive effect only on relatively strong signals. Weak signals on the contrary require longer periods to distinguish correlation peaks.

The other option – disabling tracking for some time does not have the disadvantage of reducing weak signal visibility. The upper limit is about 2 seconds is a relatively good value. However, it is only applicable for satellites with low elevation angles. The safe value for all the satellites does not exceed 0.5 seconds for a stationary receiver and 0.1s for a receiver in a car.

Considering worst case scenario, tracking algorithm needs about 50ms to get steady PLL lock [7]. For lower accuracy even less time is needed. Therefore, the amount of data can be reduced by 90.1% for stationary receiver and by 66.6% for a car receiver. Additional reduction can be achieved combining this method with in-epoch data reduction. For example, 50% reduction can be considered. The overall reduction can exceed 95% for a stationary receiver and 83% for a car receiver. Therefore, the required data rate for 53 MSPS 2bit frontend is about 4.72 Mbit/s for a stationary receiver and 17.66 Mbit/s for a car receiver.

Another matter to consider is that reduced dataflow allows permanent tracking without losing the lock on signals. However, in this case accuracy can degrade. One of the ways of improving accuracy is to transfer several full epochs just before position computation is to be done. This would not significantly affect data channel load; however, it will restore the accuracy to full-data-processing levels.

There is still a matter of other effects which may shorten possible disable tracking time. This paper is only focused on Doppler frequency change rate as the main limitation. However, there are several other effects such as local oscillator instability, leading to IF error and various nonlinear effects in RF frontend. Additional tests with actual receiver data are required.

The main problem is that the data rate is still high even after all data reductions. One of the possible solutions is designing a simple correlator system using a low power FPGA or an MCU. It would allow greater pause between sending data without losing PLL lock. Another option is based on snapshot approach [8].

Keeping on-board tracking with cloud assistance is relevant for systems with limited network access. An example of such a system is a space vehicle requiring high accuracy positioning before maneuvering without excessive power consumption.

7. Conclusion
SDR-based GNSS receiver is a flexible device with a great potential. However, it requires a sophisticated computational unit. One of possible solutions is a distributed system. The sensor part contains antenna and RF frontend and the processing part is a remote PC or server. The parts could be connected via LAN/WAN.

The main problem of such approach is high network load. One of the methods to reduce data rate implies truncating each integration period. It can reduce data rate by 50-75%. However, it is not acceptable with weak signals. In the proposed method tracking process is suspended for a certain period. Its duration is limited mostly by Doppler frequency changes. Calculations and simulation results show that acceptable intervals are about 0.5s for a stationary receiver and 0.1s for a car receiver. Using this
method with truncation of integration periods reduces data rate by 95% for a stationary receiver and by 83% for a car receiver.

Additional testing is needed for the methods using actual physical experiments to ensure stability of the PLL and DLL filters. Further research is to be conducted on combining simple on-board correlators with cloud processing.

All testing software is available at [9].

Acknowledgments
The work was carried out within the project 0777-2020-0018 financed from the state assignment means given to winners of competition of scientific laboratories of educational organizations of higher education under the authority of Ministry of Science and Higher Education of Russia.

References
[1] Kumarin A A and Kudryavtsev I A 2019 SoC opportunities for boosting SDR GNSS performance, CEUR Workshop Proc. 2416 457-462
[2] Kumarin A A, Kudryavtsev I A and Shafran S V. Implementation of a GNSS receiver signal tracking module, 27th Saint Petersburg Int. Conf. on Integrated Navigation Systems Proc. pp 1-4
[3] Niehoefer B and Schweikowski F 2014 Cloud-aided SDR Solution for Lance-Specific Vehicle Positioning via Local Interference Compensation IEEE Aerospace Conf. pp 1-7
[4] Tsui J B 2000 Fundamentals of Global Positioning System Receivers: A software Approach. (John Willey and Sons, New York) pp 73-105
[5] Kaplan E D and Hegarty C J 2006 Understanding GPS Principles and Applications, Second Edition (Artech house, Boston) pp 153-243
[6] Bokare P S and Maurya A K 2017 Acceleration-Deceleration Behaviour of Various Vehicle Types Transportation Research Procedia 25 4733-4749
[7] Borre K, et. al. 2006 A software-defined GPS and Galileo receiver (Birkhauser, Boston) p 91
[8] Shafran S. Gizatulova E and Kudryavtsev I 2018 Snapshot technology in GNSS receivers 25th Saint Petersburg Int. Conf. on Integrated Navigation Systems pp 1-3
[9] Kumain A A GNSS research public repository github.com/alkysamarapublic/gnss_ddoppler