A Carrier Synchronization Algorithm for SDR-based Communication with LEO Satellites

Sebastian KOZŁOWSKI

Inst. of Radioelectronics and Multimedia Technology, Warsaw University of Technology, Warsaw, Poland

S.Kozlowski@ire.pw.edu.pl

Submitted May 22, 2017 / Accepted October 19, 2017

Abstract. *In this work a carrier synchronization algorithm for communication with low Earth orbit (LEO) satellites was proposed. The algorithm has a form of a software routine and is intended to be run on a typical personal computer (PC) providing computational resources for a software defined radio (SDR) receiver. Due to limited computational power of PCs in comparison with other devices, such as FPGAs, the work was focused on providing carrier synchronization with minimal processing. The algorithm includes a nonlinear operation for recovering carrier wave from the received signal, a software 2nd order type 2 phase locked loop for tracking the recovered carrier, and a correction block for removing frequency shift from the received signal. Computer simulations were performed to investigate algorithm's behavior. Additionally, execution time was measured to determine maximal symbol rate of a signal to be processed.*

Keywords

Software defined radio, satellite communication, frequency synchronization, phase locked loop

1. Introduction

Software defined radio (SDR) technology provides possibilities for developing radio systems without a need for designing dedicated hardware modules. The key feature of SDR is moving as large part of the entire system as it is possible to the software domain, that is, replacing hardware components by software routines realizing the same tasks. Computational resources can be provided to an SDR system by devices such as personal computers (PC), digital signal processors (DSP), graphics processing units (GPU) or FPGA circuits.

The use of PCs is advantageous for several reasons. They are easily accessible for millions of users. Numerous free programming tools allow for developing new applications with no costs associated with purchase of dedicated hardware or software. On the other hand, even state-of-the-art PC can occur to be too slow to perform real-time reception of certain kinds of signals. This is why efforts are undertaken to measure execution time of particular operations performed in PC-based SDR systems [1], [2], [3], as well as to optimize some computationally demanding routines [4].

Typical PC-based SDR receiver consists of an SDR module responsible for signal amplification, downconversion to baseband, some preliminary filtering, and analog-to-digital conversion. The SDR module is connected via a fast digital interface (USB3, Gigabit Ethernet, PCIe) to a device capable of performing digital signal processing involved in the actual reception process. The device operates on a sample stream being a digital baseband representation of the received signal.

A low Earth orbit (LEO) is defined as the one passing up to 2000 km above the sea level. Because placing a satellite in LEO is relatively cheap and transmission from such satellite does not require high power levels, LEO satellites are often used for low-cost missions. However, high orbital velocity specific for LEO causes a number of technical problems with radio communication. One of them is relatively short visibility time, that is, the time interval during which a satellite is seen above the horizon line by a ground station placed in a fixed point on the Earth. The other problem is high and rapidly changing Doppler shift.

Short visibility time limits amount of data that can be downloaded from a satellite. To solve this problem, a network of many ground stations can be used to exploit geographical diversity [5]. In such a case, when the satellite hides below the horizon for one ground station, it becomes visible for the other one, so the data can be downloaded with no idle periods. However, since this solution has the disadvantage of requiring a lot of ground infrastructure, another approach is taken into consideration for future LEO systems. It consists in adaptive adjusting the data rate during a single satellite's pass. When a satellite passes over a ground station, the distance between them changes and so does signal-to-noise ratio (SNR). Thus it is possible to use robust, but inefficient modulation schemes when the satellite is seen at low elevation angles, and then switch to more efficient schemes when the satellite approaches its highest point in the sky [6], [7]. The consequence of such approach is that the receiver of the ground station should be capable of synchronizing with signals modulated with complex

modulation schemes, usually M-PSK or M-APSK, which besides providing high bit rate are relatively robust to nonlinear distortion and thus allow efficient use of the satellite's onboard power amplifier.

The Doppler shift characteristic of a LEO satellite signal is known to be strongly time-dependent [8]. For low elevation angles the Doppler shift is significant, but almost constant. Contrary, for the highest elevation angle the Doppler shift itself equals zero, but changes rapidly; in other words, its first derivative is high. Such behavior puts demands on carrier synchronization circuitry and algorithms used in both a satellite and a ground station.

This paper concerns a problem of carrier synchronization in a PC-based SDR receiver in a case of rapidly changing frequency and a high-order modulation scheme. This is a situation expected in a LEO satellite system when a satellite is observed at a high elevation angle.

There is a number of publications [9], [10] devoted strictly to the SDR-based communication with LEO satellites, but solutions presented therein are designed for simple modulation techniques. Works concerning high-order modulation schemes [11], [12], [13] are focused mainly on the TV broadcasting system known as DVB-S2 [14]. For geostationary satellites and fixed terminals the Doppler shift is not an issue, therefore the presented results relate to signals affected by Additive White Gaussian Noise and nonlinear distortions, whereas the frequency offset is assumed to be already largely compensated. In consequence, there is a lack of information whether proposed loops are able to track signal whose frequency rapidly changes and to regain lock to such signal after it is lost.

A recognized solution to the problem of varying Doppler shift is the use of PLL coupled with a frequency locked loop (FLL) [15], [16]. In such a structure, commonly referred to as an FLL-assisted PLL, frequency and phase error signals are employed for controlling the receiver's local oscillator. In a general case of digital modulation, the auxiliary frequency error signal can be obtained by means of a bank of band-edge filters having characteristics dependent on the pulse shaping method used [16]. The FLL-assisted PLL can be easily embodied as a computer program and thus it is suitable for PC-based SDR receiver, however, implementation of two appropriately cooperating loops increases computational complexity and consequently limits data rate of a signal to be processed.

A simple carrier synchronization method for an adaptive LEO satellite system was presented in [7]. The authors run a number of tests in a set-up consisted of SDR modules acting as a LEO satellite, a channel simulator and an SDR receiver. Modulation schemes up to 16-APSK were employed. In the implemented software receiver the current value of the carrier frequency was determined in a frame-by-frame manner by searching for a peak indicating a presence of the carrier wave in the spectrum of the received signal subjected to an appropriate nonlinear operation. The phase was synchronized in the subsequent channel equalizer. The main weakness of this method is that two conflicting requirements are to be simultaneously met. The frames have to be short enough to meet the assumption of constant Doppler shift and long enough to provide desired frequency resolution and make the peak clearly higher than other spectrum components.

In this paper a software PLL for carrier synchronization is proposed. Adjective "software" indicates that the loop is actually a piece of a computer program intended to be run on a common PC forming a part of an SDR receiver. To minimize complexity, the loop is not FLL-assisted and resulting limitations of performance are presented in the paper. The loop parameters were optimized to achieve reasonable failure rate in the most demanding conditions, i.e. when the satellite passes through the highest point of its orbit, in which the Doppler shift changes most rapidly. At the same time SNR takes the highest value, so it is possible to transmit data using high-order modulation schemes known to be sensitive to frequency synchronization imperfections. The tests presented in the paper were run for 16-APSK constellation. Finally, execution time of the loop is provided. This factor is not discussed by other authors, although it may be crucial, since too long computations may prevent an otherwise excellent solution from being implemented in an SDR receiver.

2. Operating Conditions

The proposed algorithm is designed to be a separate routine independent of other ones involved in the reception process. In particular, no symbol timing is required and consequently no hard decisions on symbols can be made. The exact structure of the received signal is unknown, that is, no information on the presence of pilots, training sequences, etc., are available. Accordingly, the loop forming the main part of the algorithm can be classified as a non-data-aided (NDA) PLL [17] used for continuous tracking of carrier phase rather than for determining the actual frequency offset.

The signal model is as follows. The receiver's downconverter outputs an analog signal $y(t)$:

$$y(t) = \exp\left(\mathrm{j}2\pi\varphi(t)\right)\cdot r(t) + v(t) \tag{1}$$

where $r(t)$ is the baseband representation of the signal transmitted by the satellite and $v(t)$ is Additive White Gaussian Noise. Channel attenuation is not taken into consideration. $\varphi(t)$ reflects phase change determined by current Doppler shift and depending on the satellite's orbit. Transmitted signal $r(t)$ is given by:

$$r(t) = \sum_k c_k h\left(t - kT_{\mathrm{symb}}\right) \tag{2}$$

where c_k are complex representations of symbols selected from 16-APSK scheme, T_{symb} stands for a symbol duration, and $h(t)$ is a square-root raised cosine (SRRC) pulse.

In the receiver $y(t)$ is passed through the second SRRC filter:

$$y_{16\text{APSK}}(t) = y(t) * h(t) \tag{3}$$

where * represents convolution. The received signal is assumed to be sampled with the oversampling factor of 4:

$$y_{16\text{APSK}}[n] = y_{16\text{APSK}}\left(n \cdot \frac{T_{\text{symb}}}{4} + \tau\right). \tag{4}$$

Simulations were carried out for $\tau = 0$, however, it should be stressed that information about which sample is closest to a symbol and which ones correspond to a transition between consecutive symbols is never used. It was confirmed that for another values of τ the algorithm works in the same manner, but the numbers presented below as performance measures are in general different. SRRC filter has known roll-off factor R and a number of FIR taps large enough to avoid intersymbol interferences.

It is known that any PLL needs a sine wave as a reference signal to lock to. In the digital baseband representation of the modulated signal, $y_{16\text{APSK}}[n]$, there is no such wave. It has to be generated by carrier recovery, which in turn can be performed by a nonlinear operation, in general different for different modulation schemes. In a case of 16-APSK, the carrier can be recovered by raising the modulated signal to the 12th power [11]. Since amplitude information is never used in the PLL and should be removed anyway for more reliable operation of the lock indicator, as it is discussed in the next sections, each sample of the reference signal can be divided by its absolute value. Finally, the reference signal is given by:

$$y_{\text{ref}}[n] = \left(y_{16\text{APSK}}[n]\right)^{12} / \left|\left(y_{16\text{APSK}}[n]\right)^{12}\right|. \tag{5}$$

Figure 1 presents spectra of $y_{\text{ref}}[n]$ for $R = 0.2$ and $R = 0.9$, averaged over 5000 realizations, each computed from 40 000 samples (1000 16-APSK symbols, 4 samples per symbol, symbol duration $T_{\text{symb}} = 1$ μs). SRRC filter was applied twice to simulate filtering at both transmitter and receiver. The plots are normalized to shift the maximum value to 0 dB.

The spectra comprise a wideband component spread over the entire frequency range and three sine wave components. Two of them correspond to the symbol rate ($f_{\text{symb}} = 1$ MHz). The third one represents the recovered carrier wave, but is located at twelve times the carrier frequency – in this case it is 300 Hz since the frequency offset f_c introduced for the simulation purpose equals 25 Hz. This is the signal the PLL is supposed to track.

It is clearly visible that the roll-off factor has a huge impact on the sine wave generation. The power of the carrier component is 0 dB due to the normalization used, whereas the power of the random component can be computed by numerical integration of the area under the spectrum taking into account the FFT gain. The result equals 19 dB and 7 dB for $R = 0.2$ and $R = 0.9$, respectively. These values show that even when no noise is present, the PLL has to lock to a very weak sine wave strongly disturbed by a random signal. The greater is the roll-off factor,

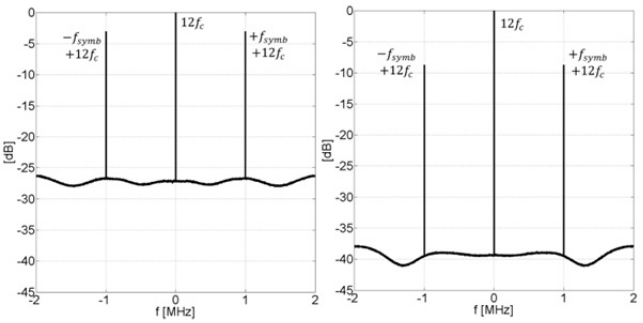

Fig. 1. Spectra of the reference signal for: (a) $R = 0.2$ and (b) $R = 0.9$.

time [s]	**dist.** [km]	$\mathbf{f_{Dopp}}$ [Hz]	$\mathbf{f'_{Dopp}}$ [Hz/s]
0	2000	-49 000	1.4
261	350	0	1 100
522	2000	49 000	1.4

Tab. 1. Orbit parameters.

the better reference signal can be obtained, but at the expense of spectral efficiency. Frequency band corresponding to a case of $R = 0.9$ is approximately 1.6 times wider than the one for $R = 0.2$. For this reason in satellite communication low values of the roll-off factor are preferred, e.g. in DVB-S2 [14] standard R can be changed from 0.2 to 0.35. To investigate demanding conditions, in the paper it is assumed that $R = 0.2$.

The orbit model necessary for numerical simulations was generated using STK Software [18]. Its main parameters are presented in Tab. 1. Radial velocity was converted into the Doppler shift assuming carrier frequency of 2 GHz. From many possible orbits the one enabling adaptive changes of the modulation scheme was selected. One can notice that during a single satellite's pass the distance to the ground station decreases almost 6 times, providing approx. 15 dB SNR improvement, which is enough to switch from BPSK to 16-APSK scheme as long as some forward error correction is employed. However, at the point where the highest SNR is expected and the most complex modulation scheme will be used, the Doppler shift changes more than 1 kHz per second. If a synchronization algorithm used in the receiver fails to track such changing frequency, benefits from high SNR will not be achieved.

3. Structure of the Algorithm

The block diagram of the proposed frequency synchronization algorithm is shown in Fig. 2. An input signal $y_{16\text{APSK}}^{\text{in}}$ is fed to the nonlinear operation block to generate a reference signal y_{ref} provided to the software PLL (SPLL). SPLL feeds a signal from its virtual internal generator to the frequency divider (div) which reduces frequency by a factor of 12. The divider's output signal Ψ^{out}, represented by a stream of phase values, is used for removing frequency shift from the input signal $y_{16\text{APSK}}^{\text{in}}$ in the correction block. After the correction the frequency shift is

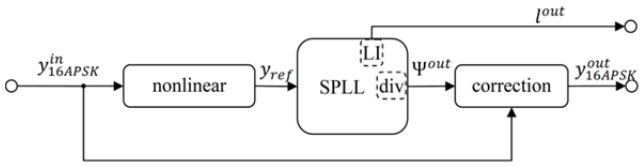

Fig. 2. Block diagram of the proposed frequency synchronization algorithm.

removed, however, the constellation of the output signal may be rotated by a fixed angle. Such behavior is typical for phase-locked loops and in this paper it is assumed that the rotation is eliminated in further channel equalizer block, not shown in the figure. SPLL includes also the lock indicator (LI) providing a signal referred to as l^{out}. It allows to determine whether SPLL is in a transient or locked state.

3.1 Software PLL

It is known that the 3rd order PLL is a preferred solution for space applications due to its ability for tracking a frequency ramp, i.e. a signal with linearly increasing or decreasing frequency. However, in this paper a 2nd order loop is considered for two reasons: lower computational complexity and the simple relationship between the loop's parameters and its impulse response which simplifies the design process. The PLL used is based on typical (known e.g. from GNU radio [19]) set of operations:

$$\begin{aligned}
&1.\ \varepsilon \leftarrow \text{phase_det}\left(\Phi_{\text{ref}}[n]-\Phi_{\text{O}}\right)\\
&2.\ \omega_O \leftarrow \omega_O + \beta\varepsilon\\
&3.\ \Phi_{\text{O}} \leftarrow \left[\Phi_{\text{O}}+\omega_{\text{O}}+\alpha\varepsilon\right]_{\mp\pi}\\
&4.\ n \leftarrow n+1\\
&5.\ \text{goto 1}
\end{aligned} \tag{6}$$

where "phase_det(•)" returns some measure of a phase error ε, "←" is an assignment operator, $\Phi_{\text{ref}}[n]$ is a phase of the n-th sample of the reference signal, $[\bullet]_{\mp\pi}$ is a wrapping operator that puts the phase value within the range $[-\pi,+\pi)$, α and β are loop parameters, and Φ_{O} and ω_{O} describe the current state of the loop. When the loop is locked, ε equals 0, which means that ω_{O} does not change and Φ_{O} changes by a fixed angle ω_{O} in each iteration. Accordingly, ω_{O} can be considered as the circular frequency multiplied by the sampling period T_{s}, and Φ_{O} as the current phase of a virtual internal generator of the loop. It should be stressed that if a coarse frequency of the reference signal $\widetilde{\Delta f}$ is known, one can run the loop with an initial condition $\omega_{\text{O}} = 2\pi\cdot\widetilde{\Delta f}\cdot T_{\text{s}}$ which for large $\widetilde{\Delta f}$ may be a prerequisite for successful locking or at least reduce the locking time.

Assuming the ideal linear phase detector, (6) can be translated into a system of difference equations:

$$\begin{cases}
\varepsilon[n] = \Phi_{\text{ref}}[n]-\Phi_{\text{O}}[n]\\
\omega_{\text{O}}[n+1] = \omega_{\text{O}}[n]+\beta\varepsilon[n]\\
\Phi_{\text{O}}[n+1] = \Phi_{\text{O}}[n]+\omega_{\text{O}}[n+1]+\alpha\varepsilon[n]
\end{cases} \tag{7}$$

or z-domain equations:

$$\begin{cases}
\varepsilon = \Phi_{\text{ref}}-\Phi_{\text{O}}\\
z\omega_{\text{O}} = \omega_{\text{O}}+\beta\varepsilon\\
z\Phi_{\text{O}} = \Phi_{\text{O}}+z\omega_{\text{O}}+\alpha\varepsilon
\end{cases} \tag{8}$$

From (8), the transfer function of the loop is:

$$H(z) = \frac{\Phi_{\text{O}}}{\Phi_{\text{ref}}} = \frac{(\alpha+\beta)z-\alpha}{z^2+z(\alpha+\beta-2)+(1-\alpha)} \tag{9}$$

By subjecting (9) to the bilinear (Tustin) transform:

$$z \approx \frac{2+sT_{\text{s}}}{2-sT_{\text{s}}} \tag{10}$$

where T_{s} is the sampling period, the transfer function of the loop in the s-domain can be obtained:

$$H(s) = \frac{s^2T_{\text{s}}^2(2\alpha+\beta)-s\,4\alpha T_{\text{s}}-4\beta}{s^2T_{\text{s}}^2(2\alpha+\beta-4)-s\,4\alpha T_{\text{s}}-4\beta}. \tag{11}$$

Further, by introducing the substitutions (12), the transfer function of the loop can be converted to the more usual form (13):

$$\begin{cases}
\alpha = \dfrac{4\zeta\theta}{1+2\zeta\theta+\theta^2}\\
\beta = \dfrac{4\theta^2}{1+2\zeta\theta+\theta^2}\\
\theta = \dfrac{\omega_{\text{n}}T_{\text{s}}}{2}
\end{cases} \tag{12}$$

$$H(s) = \frac{s^2\left(\zeta\omega_{\text{n}}T_s+\dfrac{1}{4}\omega_{\text{n}}^2T_{\text{s}}^2\right)-s\cdot 2\zeta\omega_{\text{n}}-\omega_{\text{n}}^2}{-s^2-s\cdot 2\zeta\omega_{\text{n}}-\omega_{\text{n}}^2}. \tag{13}$$

In many practical cases T_{s} is several orders of magnitude less than ζ or ω_{n}, therefore the term comprising s^2 in the nominator can be neglected. Such operation leads to the final form of the transfer function of the loop:

$$H(s) \approx \frac{s\cdot 2\zeta\omega_{\text{n}}+\omega_{\text{n}}^2}{s^2+s\cdot 2\zeta\omega_{\text{n}}+\omega_{\text{n}}^2}. \tag{14}$$

Equation (14) corresponds to the well-known 2nd order type 2 loop with a damping factor ζ and a natural frequency ω_{n}. Both these parameters determine loop characteristics in a manner described in detail e.g. in [20], [21].

3.2 Nonlinear Operation Block

The nonlinear operation block applies formula (5) to generate the reference signal. Raising a complex number to the 12th power requires only 4 complex multiplications when the following scheme is used:

$$x^{12} = \left(\left(x^2\cdot x\right)^2\right)^2. \tag{15}$$

Determining an absolute value usually involves computation of a square root, which takes a lot of time. However, one can notice that:

$$|x^{12}| = |x^6|^2 = \left(\mathrm{Re}\left[x^6\right]\right)^2 + \left(\mathrm{Im}\left[x^6\right]\right)^2. \quad (16)$$

Because the 6th power of a sample is available after the penultimate step of calculating (15), the absolute value is determined with only 2 real multiplications and 1 real addition.

The above operations are performed in a sample-by-sample manner, therefore they can be easily optimized using SIMD instructions [22] supported by processors of substantially all today's PCs. The optimized code is up to 4 times faster when using double-precision and up to 8 times faster for single-precision arithmetic provided that the Advanced Vector Extension (AVX) instruction set is available. For slightly older computers, supporting only SSE sets, the corresponding numbers equal 2 and 4, respectively. The possibility of optimization is the main reason why the nonlinear operation block should be designed as a separate subroutine responsible for preparing the reference signal before the calculations of the actual loop start.

3.3 Lock Indicator

The lock indicator operates according to the following formula:

$$l^{\mathrm{out}}[n] = y_{\mathrm{ref}}[n] \cdot \exp\left(-\mathrm{j}\Phi_{\mathrm{O}}^{n}\right) \quad (17)$$

where Φ_{O}^{n} stands for a value of Φ_{O} in the n-th iteration of SPLL. Because Φ_{O} is overwritten in each iteration, subsequent samples $l^{\mathrm{out}}[n]$ must be computed "on the fly" inside the loop.

The reference signal was prepared so as to obtain $|y_{\mathrm{ref}}[n]| \equiv 1$. This assures that the output of the lock indicator is not influenced by the received signal amplitude and settings of the receiver's gain, which makes the entire synchronization algorithm less dependent on gain control algorithms or circuitry. Assuming that the reference signal is a noiseless sine wave, $l^{\mathrm{out}} \equiv 1$ in the locked state and $\overline{l^{\mathrm{out}}} \approx 0$ in the transient state as long as the averaging is performed over a sufficiently large number of samples. In a real-world scenario $\overline{l^{\mathrm{out}}}$ is a function of time with a step indicating a transition to the locked state. Despite the normalization of the reference signal used, the actual shape of the step still depends on SNR and the number of samples taken for the averaging. For this reason processing of the lock indicator signal applied to retrieve information about the current state of the PLL is considered to be external to the synchronization algorithm and is not discussed in detail in this paper.

3.4 Correction Block

The correction block operates according to the following formula:

$$y_{16\mathrm{APSK}}^{\mathrm{out}}[n] = y_{16\mathrm{APSK}}^{\mathrm{in}}[n] \cdot \exp\left(-\mathrm{j} \cdot \Psi^{\mathrm{out}}[n]\right) \quad (18)$$

where the output signal of the frequency divider is:

$$\Psi^{\mathrm{out}}[n] = \left[\Psi^{\mathrm{out}}[n-1] + \frac{\omega_{\mathrm{O}} + \alpha\varepsilon}{12}\right]_{\mp\pi}. \quad (19)$$

Division by 12 is introduced due to the fact that the loop follows the reference signal with a frequency 12 times larger than the actual Doppler shift to be removed, as it has been already stated in Sec. 2. Consequently, each time the phase of the loop's internal generator Φ_{O} increases by a certain angle ($\omega_{\mathrm{O}} + \alpha\varepsilon$), $\Psi^{\mathrm{out}}[n]$ used for the correction purpose should be increased by the same angle divided by 12.

3.5 The Entire Algorithm

During a single call the synchronization algorithm operates on a block containing N samples. To simulate continuous operation, the internal state of SPLL, that is ω_{O}, Φ_{O}, and $\Psi^{\mathrm{out}}[N]$, is stored after a block is processed and restored immediately after the next call ($\Psi^{\mathrm{out}}[N]$ is restored as $\Psi^{\mathrm{out}}[0]$).

The pseudo-code of the algorithm is as follows:

1. Prepare N samples of the reference signal acc. to (5)
2. FOR EACH $n \le N$
   a. $X_{\mathrm{O}} \leftarrow \exp\left(\mathrm{j} \cdot \Phi_{\mathrm{O}}\right)$
   b. $l^{\mathrm{out}}[n] \leftarrow y_{\mathrm{ref}}[n] \cdot X_{\mathrm{O}}^{*}$
   c. $\varepsilon \leftarrow$ phase_det($l^{\mathrm{out}}[n]$)
   d. $\omega_{\mathrm{O}} \leftarrow \omega_{\mathrm{O}} + \beta\varepsilon$
   e. $\Delta \leftarrow \omega_{\mathrm{O}} + \alpha\varepsilon$
   f. $\Phi_{\mathrm{O}} \leftarrow \left[\Phi_{\mathrm{O}} + \Delta\right]_{\mp\pi}$
   g. $\Psi^{\mathrm{out}}[n] \leftarrow \left[\Psi^{\mathrm{out}}[n-1] + \frac{\Delta}{12}\right]_{\mp\pi}$
3. Perform correction acc. to (18)

Points 1 and 3 reflect the operation of the nonlinear operation block and the correction block, respectively. The FOR loop representing SPLL performs substantially the same task as pseudo-code (6). Two lines were added: in line (b) the lock indicator is implemented and in line (g) the frequency divider prepares a signal for the correction block. The phase detector was slightly redefined and now it takes $l^{\mathrm{out}}[n]$ as an input argument. This has no impact on SPLL operation since from line (b) it is clear that $l^{\mathrm{out}}[n] = \exp(\mathrm{j} \cdot (\Phi_{\mathrm{ref}}[n] - \Phi_{\mathrm{O}}))$, thus it carries all information necessary for retrieving current phase error value.

4. Algorithm Performance

The synchronization algorithm was implemented as a Matlab mex file. Two operation modes were provided: reference (R) and optimized (O) in terms of computational complexity. In the first one a sawtooth phase detector is employed by means of atan2 function:

$$\varepsilon \leftarrow \mathrm{atan2}\left(\mathrm{Im}\left[l^{\mathrm{out}}[n]\right], \mathrm{Re}\left[l^{\mathrm{out}}[n]\right]\right) = \left[\Phi_{\mathrm{ref}}[n] - \Phi_{\mathrm{O}}\right]_{\mp\pi}. \quad (20)$$

In the latter one the imaginary part is taken as a measure of the phase error:

$$\varepsilon \leftarrow \mathrm{Im}\left[l^{\mathrm{out}}[n] \right] = \sin\left(\Phi_{\mathrm{ref}}[n] - \Phi_{\mathrm{O}} \right). \qquad (21)$$

In both cases, Euler's formula is used to express exp(jx) by sine and cosine functions. In the reference mode atan2, sine, and cosine are taken from the standard C math library, whereas in the optimized mode atan2 is not used at all and sine and cosine are implemented using look-up tables (LUT) with resolution 1.2 degree providing both very fine quantization and reasonable memory occupation.

The algorithm performance was tested by computer simulations for the orbit presented in Tab. 1 and SNR = 13 dB measured at the constellation diagram. This can be considered as a limit value since corresponding symbol error rate (SER) equals approximately 0.1 and one can assume that using 16-APSK modulation when lower SNR is observed is pointless.

4.1 Selection of Loop Parameters

The damping factor ζ and the natural frequency ω_{n} determine the loop's tracking capabilities and should be adjusted to a specific orbit. The known way to express tracking capabilities is providing the Mean Time to Lose Lock (MTTL) [17], however, an excessive number of simulations would be required to determine this parameter for numerous pairs of ζ and ω_{n}. Furthermore, the results would be ambiguous since the operating conditions of the loop change continuously as it is presented in Tab. 1. For these reasons in this paper a simpler quality measure was adopted. It was checked whether the loop is able to stay locked when the satellite is seen at high elevation angles. For given ζ and ω_{n} 20 tries were performed, each one starting at a point where the Doppler shift change becomes higher than 1 kHz/s and ending at a point where the change drops below 1 kHz/s. Such fragment of the orbit corresponds to 2400 blocks, each containing N = 40 000 samples or 10 000 symbols oversampled by a factor of 4.

The results are shown in Fig. 3 in a form of the failure percentage indicated by a color code. One can notice that for both modes it is possible to configure the loop so as to obtain a failure-free operating. However, in the reference mode the range of loop's parameters providing correct operation is very limited, whereas in the optimized mode it is much wider in both ζ and ω_{n} dimension. Additionally, in both cases relatively high damping factor is required, in contrast to many applications where $\zeta < 2$ is typically used.

4.2 Capture Range

A ground station is usually equipped with a high-gain dish antenna following the satellite while it travels through the sky. Thus the orbit of the satellite must be known at the receiver and thereby the current Doppler shift value can always be roughly estimated. The estimate can be used to coarsely remove the frequency shift if necessary (e.g. if the shift is large in comparison with the signal bandwidth) and

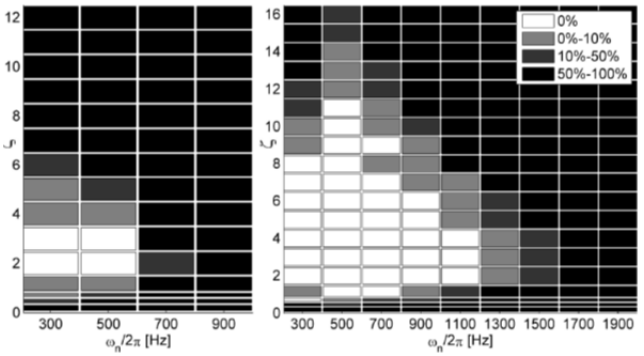

Fig. 3. Tracking failure rate at SNR = 13 dB for various values of loop parameters: mode R (a), mode O (b).

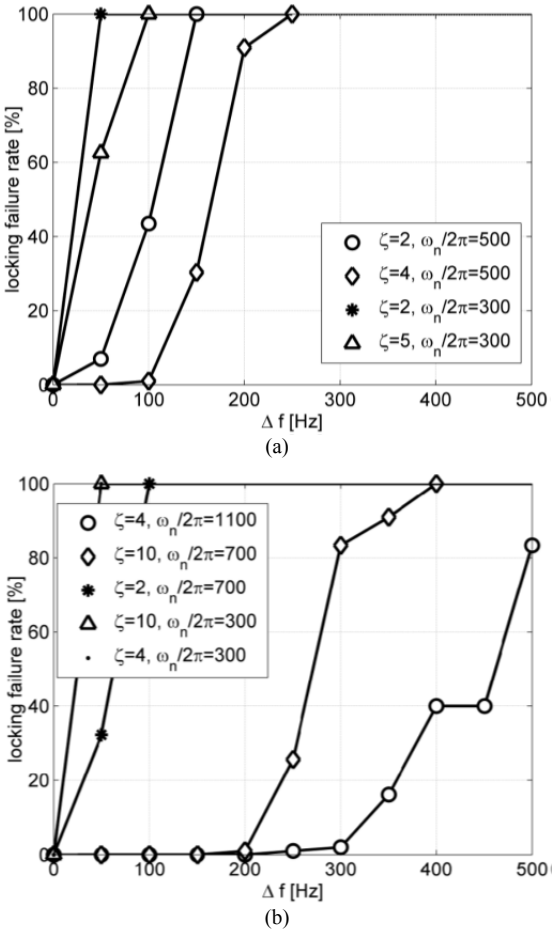

Fig. 4. Locking failure rate at SNR = 13 dB for various values of loop parameters: mode R (a), mode O (b).

to determine the initial value of ω_{O} as it is discussed in Sec. 3.1. However, due to the estimation inaccuracy, perfect initial frequency synchronization can never be obtained and the algorithm always starts with ω_{O} slightly different than the frequency of the reference signal. For this reason it should exhibit a tolerance for such difference.

Figure 4 shows a percentage of locking failures for the algorithm run when the satellite is seen at the highest elevation angle with the initial condition $\omega_{\mathrm{O}} = 2\pi \cdot (f_{\mathrm{Dopp}} - \Delta f) \cdot T_{\mathrm{s}} \cdot 12$, where Δf is a Doppler shift estimation error.

Simulations were performed for several sets of parameters assuring acceptable loop tracking performance (indicated by white or light grey patches in Fig. 3). It is easy to notice that the higher are the damping factor and natural frequency, the wider is the capture range, that is, the value of Δf for which the loop is still able to lock. This is not a surprise since increasing those two parameters increases the loop's bandwidth. Other observations are that the sinusoidal phase detector used in the optimized mode provides much wider capture range that the sawtooth detector.

The simulations also show that the frequency estimation error has to be low. The reasonable values of Δf do not exceed 100 Hz for mode (R) and 300 Hz for mode (O). Taking into account that the Doppler shift changes by 1100 Hz every second, these results put demands on the Doppler shift estimation method employed in the receiver.

4.3 Execution Time

The execution time of the three subroutines constituting the proposed synchronization algorithm was measured for Intel Core i5 processor working at 2.67 GHz. The results are shown in Tab. 2 as the median of time required for processing one symbol (4 samples), where a complex sample is represented by two double-precision real numbers. The reference signal preparation was implemented according to (5), (15), (16) and optimized using SIMD instructions up to SSE4.

As it was expected, the most computationally expensive parts of the algorithm are connected with the trigonometric functions. Removing them speeds up the correction block 8 times. However, the most complex block is SPLL and in practice it is the one determining the performance of the entire algorithm. SPLL can be speed up almost 6 times by removing trigonometric functions. Since the sinusoidal detector used in mode (O) performs just retrieving $\mathrm{Im}[l^{\mathrm{out}}[n]]$ from the memory, which is almost immediate operation, further simplification of the algorithm would be a difficult task. Consequently, the time obtained for mode (O) determines the maximal symbol rate of the signal to be processed. It equals $(140\ \mathrm{ns/symb.})^{-1} \approx 7$ Msymb./s provided that no other processing is involved in the reception or such processing is realized by other processor cores. Of course, one can improve the algorithm performance by simply using more powerful computer.

mode	nonlinear	SPLL	correction	total
R	33	560	170	760
O	33	87	20	140

Tab. 2. Medians of the running time in [ns/symbol].

SNR [dB]	13	14	15	16
***R* = 0.2**	300	550	900	1300
***R* = 0.3**	300	550	900	1300
***R* = 0.9**	450	800	1250	1650

Tab. 3. Capture range in [Hz] under different conditions (mode O, $\zeta = 4$, $\omega_n/2\pi = 1100$ [Hz]).

4.4 Performance under Different Conditions

The loop exhibiting the best performance (sinusoidal detector, $\zeta = 4$, $\omega_n/2\pi = 1100$ Hz) was selected for further tests in order to investigate its behavior under different conditions, namely for various values of SNR and R. It was found that the loop's tracking capability is extremely sensitive to SNR changes. For SNR = 12 dB the tracking failure rate occurred to be almost 100% whereas for 13 dB the same parameter equals zero as it is indicated in Fig. 3. Contrary, the capture range seems to improve much more slowly as SNR increases. This effect is illustrated in Tab. 3 providing values of initial Doppler shift estimation error Δf for which locking failure rate does not exceed 10%. As it can be noticed, in the assumed SNR range Δf increases 1.3 to 1.8 times per decibel.

Another observation is that only a significant increase of the roll-off factor results in capture range improvement. As one can see from Tab. 3, results obtained for $R = 0.2$ and $R = 0.3$ may differ no more than the frequency resolution of 50 Hz used in the simulation. For $R = 0.9$ the improvement is clear, but does not exceed 50%.

5. Conclusions

In this work a carrier synchronization algorithm for communication with LEO satellites was proposed. The algorithm is based on the 2nd order type 2 software PLL with an additional lock indicator and does not require any knowledge of an actual signal structure, channel coding, or symbol timing. The only assumption made is that the modulation scheme is known and the transmission is continuous, that is, no silent periods are present.

The algorithm was tested by means of computer simulations for a 16-APSK-modulated signal received in a low SNR scenario. The simulations have shown that using a sinusoidal phase detector leads not only to execution time reduction due to the lack of need for calculating arctangent, but also to capture range improvement. However, even in this case the algorithm requires relatively accurate initial Doppler shift estimate for correct locking.

Algorithm optimized for the test orbit was investigated under different noise conditions. Obtained results have shown that there is a certain threshold SNR value below which the algorithm does not work. Exceeding this threshold causes very rapid improvement of algorithm tracking performance and much slower yet continuous improvement of locking capabilities.

Measurements of the execution time indicate that the upper limit of the symbol rate of a signal to be processed is 7 Msymb./s when PC equipped with Intel Core i5 processor is used.

The advantage of the algorithm is its versatility. Although tested for a case of 16-APSK, it can be easily adopted for other digital modulations. The only modification required is replacing the exponent in the nonlinear operation block and the divider in SPLL, both equal to 12,

with some other value corresponding to selected modulation scheme. This feature makes the proposed solution specially suitable for adaptive systems in which various modulations are used depending on current radio channel conditions.

References

[1] GE, F., CHIANG, C. J., GOTTLIEB, Y. M., et al. GNU Radio-based digital communications: computational analysis of a GMSK transceiver. In *IEEE Global Telecommunications Conference (GLOBECOM 2011)*. Houston (TX, USA), Dec. 2011, p. 1–6. DOI: 10.1109/GLOCOM.2011.6133692

[2] TANAKA, Y., INAMORI, M., SANADA, Y. Timing synchronization scheme for OFDM signal in general purpose processor based software defined radio receiver. In *IEEE 23rd International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*. Sydney (NSW, Australia), Sept. 2012, p. 2287 to 2290. DOI: 10.1109/PIMRC.2012.6362737

[3] CHEN, Z., WU, J. LTE physical layer implementation based on GPP multi-core parallel processing and USRP platform. In *9th International Conference on Communications and Networking in China (CHINACOM)*. Maoming (China), Aug. 2014, p. 197–201. DOI: 10.1109/CHINACOM.2014.7054285

[4] TSENG, S.-M., KUO, Y.-C., KU, Y.-C., et al. Software Viterbi decoder with SSE4 parallel processing instructions for software DVB-T receiver. In *IEEE International Symposium on Parallel and Distributed Processing with Applications*. Chengdu (China), Aug. 2009, p. 102–105. DOI: 10.1109/ISPA.2009.100

[5] ANDROLEWICZ, J. F., BUFFINGTON, R. L., KIEF, C. J., et al. Software-defined and cognitive radio technology for military space applications. In *Wireless Innovation Forum Conference on Communications Technologies and Software Defined Radio*. Washington (USA), Nov.-Dec. 2011.

[6] GEDNEY, R. Low-cost, high-speed modems for small LEO-to-ground and LEO-to-LEO data links. In *29th Annual AIAA/USU Conference on Small Satellites.* Logan (UT, USA), Aug. 2015.

[7] SKARZYNSKI, J., DARMETKO, M., KOZLOWSKI, S., et al. SDR implementation of the receiver of adaptive communication system. *Radio Science,* April 2016, vol. 51, no. 4, p. 344–351. DOI:10.1002/2015RS005899

[8] ALI, I., AL-DHAHIR, N., HERSHEY, J. E. Doppler characterization for LEO satellites. *IEEE Transactions on Communications,* March 1998, vol. 46, no. 3, p. 309–313. DOI: 10.1109/26.662636

[9] DASCAL, V., DOLEA, P., CRISTEA, O., et al. Low-cost SDR-based ground receiving station for LEO satellite operations. In *11th International Conference on Telecommunication in Modern Satellite, Cable and Broadcasting Services (TELSIKS)*. Nis (Serbia), Oct. 2013, vol. 02, p. 627–630. DOI: 10.1109/TELSKS.2013.6704456

[10] BOSCO, M., TORTORA, P., CINARELLI, D. Alma mater ground station transceiver: a software defined radio for satellite communications. In *IEEE Metrology for Aerospace (MetroAeroSpace)*. Benevento (Italy), May 2014, p. 549–554. DOI: 10.1109/MetroAeroSpace.2014.6865986

[11] GAPPMAIR, W., CIONI, S., VANELLI-CORALLI, A., et al. Exact analysis of different detector algorithms for NDA carrier phase recovery of 16/32-APSK signals. *Transactions on Emerging Telecommunications Technologies*, March 2010, vol. 21, no. 2, p. 154–166. DOI: 10.1002/ett.1353

[12] DE GAUDENZI, R., GUILLEN I FABREGAS, A., MARTINEZ, A. Performance analysis of turbo-coded APSK modulations over nonlinear satellite channels. *IEEE Transactions on Wireless Communications*, Sept. 2006, vol. 5, no. 9, p. 2396–2407. DOI: 10.1109/TWC.2006.1687763

[13] BEITLER, U., AMRANI, O. Real time cycle slip detection and correction for APSK modulation. *IEEE Transactions on Communications*, Feb. 2014, vol. 62, no. 2, p. 736–746. DOI: 10.1109/TCOMM.2013.112913.130206

[14] ETSI EN 302 307 V1.2.1 (2009-08). Digital video broadcasting (DVB); second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications (DVB-S2).

[15] MA, L., SHI, L., WANG, Z. Performance analysis of a second order FLL assisted third order PLL for tracking Doppler rates. *WSEAS Transactions on Communications*, 2014, vol. 13, p. 26–43. E-ISSN: 2224-286

[16] HARRIS, F., VENOSA, E., CHEN, X., DICK, C. Non-data aided symbol and carrier synchronization via band-edge filters. In *46th Asilomar Conference on Signals, Systems and Computers*. Pacific Grove (CA, USA), Nov. 2012.

[17] LANDIS, S., BOBROVSKY, B. Z. Decision directed versus non-data aided PLLs: a comparative review. *IEEE Transactions on Communications*, April 2010, vol. 58, no. 4, p. 1256–1261. DOI: 10.1109/TCOMM.2010.04.080286

[18] *STK*. [Online] Cited May 2017. Available at: http://www.agi.com/products/stk/

[19] *GNU Radio*. [Online] Cited May 2017. Available at: http://gnuradio.org/

[20] KROUPA, V. F. *Phase Lock Loops and Frequency Synthesis.* John Wiley & Sons, 2003. ISBN: 0-470-84866-9

[21] GARDNER, F. M. *Phase Lock Techniques.* 3rd ed. John Wiley & Sons, 2005. ISBN: 978-0-471-43063-6

[22] *Intel Intrinsics Guide*. [Online] Cited May 2017. Available at: https://software.intel.com/sites/landingpage/IntrinsicsGuide/

About the Author ...

Sebastian KOZŁOWSKI was born in Warsaw, Poland. He received the M.Sc. and Ph.D. degrees in Telecommunication from Warsaw University of Technology in 2004 and 2011, respectively. His research interests include propagation in wireless systems, MIMO techniques, software-defined radio, and cognitive radio.