2017 International Conference on Military Technologies (ICMT)
May 31 – June 2, 2017, Brno, Czech Republic

Time Difference Measurement Algorithm for TDOA Positioning System using RTL-SDR

Rudolf SCHREIBER*, Josef BAJER*
* Dept. of Aerospace Electrical Systems, University of Defence, Kounicova 65, 662 10 Brno, Czech Republic

Abstract— **The paper is focused on radionavigation systems. The subject of the paper is a design, implementation and experimental verification of an algorithm, which is able to determine time differences between the arrivals of signals from multiple transmitters. The algorithm is a fundamental element for the implementation of the system for position determination using Time Difference Of Arrival (TDOA) method. The implementation is realized in MATLAB environment. The proposed system is intended for position determination on a board of small Unmanned Aerial System (UAS), thus the experimental verification was performed using low-cost software defined receiver RTL-SDR.**

Keywords-Aircraft navigation; radio navigation, software radio.

I. Introduction

Position determination is an essential task to make possible a navigation of the movable object in space no matter what the object is. Various principles for position determination are used, depending on the type and nature of the object and the surrounding environment. To determine the position of the aircraft or UAS (Unmanned Aerial Vehicle) radio or inertial systems are usually used. In case the radio system is used for navigation, various different methods of determining the position can be implemented, such as Time Difference Of Arrival (TDOA), Time Of Arrival (TOA), Angle of Arrival (AOA).

TDOA system in general is based on the principle of measuring the time differences in two basic ways. In the first case, several receivers receive the signal of one transmitter. The time of the signal arrival at each receiver depends on the distance of each receiver to the transmitter. In the latter case, one receiver receive signals from multiple transmitters. The transmitters send signals at the same time. Signals from each of transmitters are received at different time depending on the distance from the receiver to each transmitter. In both cases, time differences of arrival of multiple signals are evaluated. Each pair of signals, and each time difference, consequently corresponds to the position line (surface) in the space. In case of TDOA, positional line (surface) have the shape of hyperbola (hyperboloid). The intersection of the hyperboles (hyperboloids) indicates the exact position of the object [1].

The paper is focused on determining the position of the UAS or aircraft using TDOA system, which consists of multiple transmitters located on the earth and a receiver installed on board the UAS. Onboard receiver is represented by a software-defined receiver RTL-SDR. The subject of the paper is a detailed description of the algorithm calculating time differences and performed practical experiments.

II. Proposed Algorithm

The proposed algorithm consists of two main parts, namely an initialization and a code loading signal from the receiver and determining time differences, which is cyclically repeated. The flowchart is shown in Fig. 1.

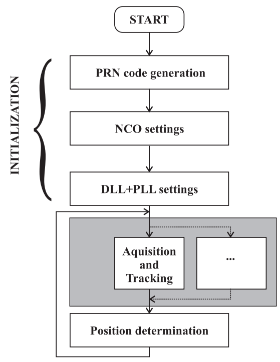

Figure 1. Flow graph of the algorithm

In the first step, samples of the sinusoidal waveforms for a reference Numerically Controlled Oscillator (NCO) are generated. Sinusoidal waveform at the output of the NCO is generated via the so-called Phase accumulator. In this case, the phase accumulator is represented by a variable that contains information about the current position of the sample in the register, where the sample is to be transferred to the output of the NCO. Position of the sample corresponds to the phase of the harmonic signal. Register containing samples of harmonic signal is called Phase to amplitude converter. Systematical omitting some of samples of the harmonic signal during the generation can change frequency of generated harmonic signal [2].

In the next step masks of pseudorandom codes that correspond to pseudo-random sequences emitted by each transmitter are generated. These codes serve to distinguish each transmitter from others. Code properties must enable their simultaneous utilization without mutual interference. This is achieved by using codes with required properties in terms of autocorrelation and cross-correlation. These codes are generated in the receiver in the values of 1 and -1. Generated pseudorandom sequences are sampled, to be used in a similar manner as in the case of NCO. This approach is chosen because of the possibility of future elimination of the frequency shift caused by the Doppler effect, or temperature instability of the receiver [2], [3].

In the last step of initialization, all required parameters, variables, memory registers used in other parts of the algorithm like DLL (Delay Locked Loop) and PLL (Phase Locked Loop) are set.

Initialization phase of the algorithm is followed by cyclically repeating part of the algorithm. Each cycle starts by reading the input signal from the receiver in the form of I (In-phase) and Q (Quadrature) samples. The next part of the algorithm have to be performed for each pseudo-random sequence (each transmitter) separately. For this purpose, it is useful to use parallel structure of the program, as it is shown in Fig. 1. Another option is consecutive running of whole cycle for each pseudo-random sequence. However, this causes slowing down of whole program performance, which is undesirable in the case of requirement for real time processing.

Block diagram of the algorithm for reading signal from the receiver and calculating time differences together with the lock diagram of the RTL SDR receiver is shown in Fig. 2. The function of the whole algorithm is shown on the example of BPSK signal with the use of Gold codes [3]-[7].

The received signal is in the receiver RTL-SDR converted to baseband and fed out from the receiver in the form of I and Q samples. I and Q samples in the time domain are shown in Fig. 3. There can be seen parasitic amplitude modulation, that causes alternating polarity of pseudorandom codes carried by the signal. Parasitic modulation causes some complications during demodulation. From Fig. 3 it is also clear that there is a mutual energy exchange between the I and Q branches [3].

Signal processing algorithm can be divided into two parts, on the tracking code and carrier tracking (see Fig. 2).

The task of the code tracking loop is to detect and follow a specific code in the signal. The result of such a loop is perfectly aligned replica of the reference code. Non-coherent DLL also called early-late tracking loop is used for this purpose. Non-coherent DLL allows to synchronize the generated replica with the received code in a form which includes parasitic modulation (see Fig. 3), since it uses both branches I and Q [3]-[5].

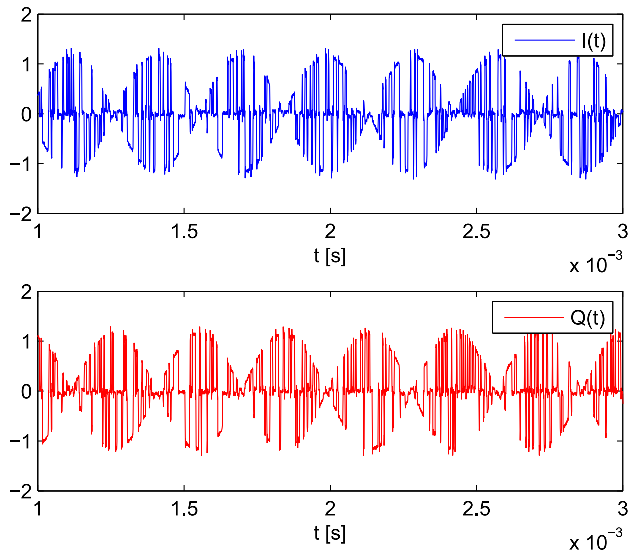

Figure 3. I and Q samples in the time domain

For synchronization is used correlation of the input signal with three replicas on both branches, as can be seen in Fig. 2. After multiplying the input I and Q signals by early, prompt and late replicas, six output signals of the correlator are formed. These signals are supplied to integrate and dump registers and for every N samples averages correlations I_x and Q_x are calculated, where x represents the early, prompt and late branch. I_x and Q_x values indicate how much a given replica code correlate with the received signal. Since the energy flows between the I and Q branches it is necessary to use both branches [3]-[7].

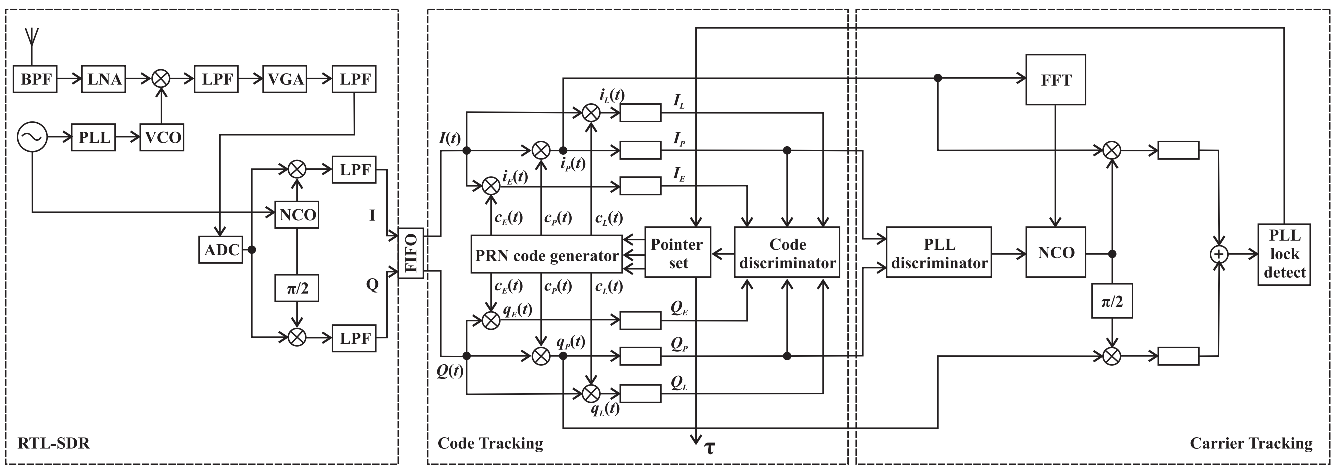

Figure 2. Acquisition and tracking algorithm

$$I_x = \frac{1}{N}\sum_{t=1}^{N} i_x(t)\cdot c_x(t) \ , \tag{1}$$

$$Q_x = \frac{1}{N}\sum_{t=1}^{N} q_x(t)\cdot c_x(t) \ , \tag{2}$$

where

$x \in \{E, P, L\}$ represents early, prompt a late signal,

I_x, Q_x are average correlation values,

$i_x(t)$, $q_x(t)$ are samples I and Q in time t,

$c_x(t)$ are samples of code replica in time t,

t represents time,

N represents the number of samples to calculate average correlation.

Values I_x and Q_x are fed into a DLL discriminator calculating correction value for generated replica D_{DLL} as follows:

$$D_{DLL} = \frac{I_E - I_L}{I_P} + \frac{Q_E - Q_L}{Q_P} \ . \tag{3}$$

In case I_P or Q_P is equal to 0, the calculation of D_{DLL} is not performed.

D_{DLL} correction value is used for time shift of generated replicas. This process is repeated in a feedback loop until the synchronization of PN code carried by the received signal with the generated replica is not achieved. When code synchronization occurs, PN code in received signal is removed by multiplying with synchronized replica and modulated data are obtained. In this case, no data are transmitted, so the output data signal is ideally constant without any changes. Therefore, it can be used to indicate the synchronization status [3].

However, in reality, the signal is modulated by parasitic amplitude modulation (as described above). For correct function of the data output it is therefore necessary to remove the periodically alternating polarity of correlator output by multiplying with the synchronized harmonic waveform. This is accomplished by including a phase locked loop behind the DLL loop (see Fig. 2).

Sinusoidal waveform of desired frequency is generated by the NCO. The frequency is obtained using FFT, which is calculated directly from the prompt correlator output signal $i_P(t)$.

Fig. 4 shows a comparison of the signal $i_P(t)$ with the output signal NCO at the moment, when code synchronization has been reached. It is clear that the code synchronization occurred at the time of approximately 10 ms. Prior to that moment a presence of pseudo-random sequence signal is seen. After achieving synchronization the pseudorandom code sequence is removed, only parasitic modulation remains.

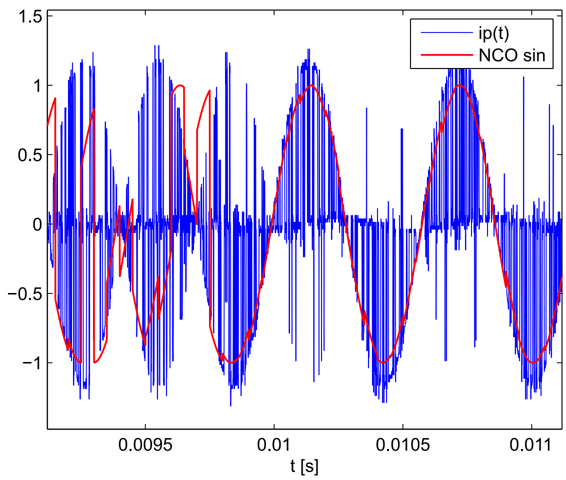

Figure 4. Correlation and NCO output signals

Fig. 5 shows a frequency spectrum of the input signal $I(t)$, the prompt correlator output signal $i_P(t)$ before reaching, and after reaching synchronization. While the synchronization is achieved a dominant spectral line occures. Corresponding frequency is consequently set in the NCO [4]-[7].

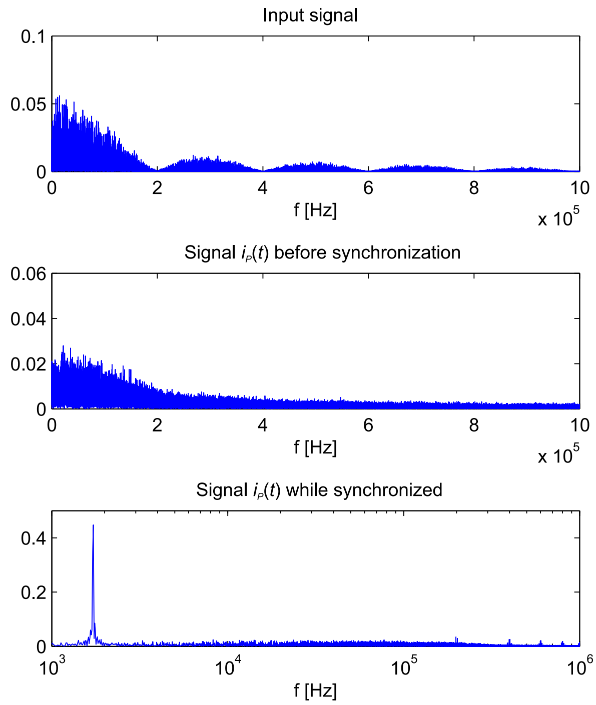

Figure 5. Correlation output in the frequency domain

Correction of NCO run identically to DLL every N samples. Discriminant function of the PLL is defined in a following form:

$$\phi = \operatorname{atan2}(Q_P, I_P) \ , \tag{4}$$

where ϕ is a phase of NCO signal.

The phase of the NCO signal is every N samples is abruptly changed (see Fig. 4). When generating code replica is not synchronous, waveform is not sinusoidal and therefore there are big leaps. After the synchronization is achieved codes began to be generated in sinusoidal shate and corrections are minimized. For the quadrature branch the same procedure is applied. The generated waveform is in this case shifted by $\pi/2$ [3].

Fig. 6 (a) shows the resultant signals obtained by multiplying the signals $i_P(t)$ and $q_P(t)$ with sinusoidal waveform. The result of this multiplication has been labeled as *data* with the index I and Q corresponding to the given branch. To detect synchronization status of the DLL a function *lockdetect* was implemented. It indicates the synchronization by calculating the average value of the sum of the two branches in the interval of N_2 as follows:

$$lockdetect = \frac{1}{N_2}\sum_{t=1}^{N_2}\left(data_I + data_Q\right) . \tag{5}$$

The output signal of function *lockdetect* is shown in Fig. 6 (b).

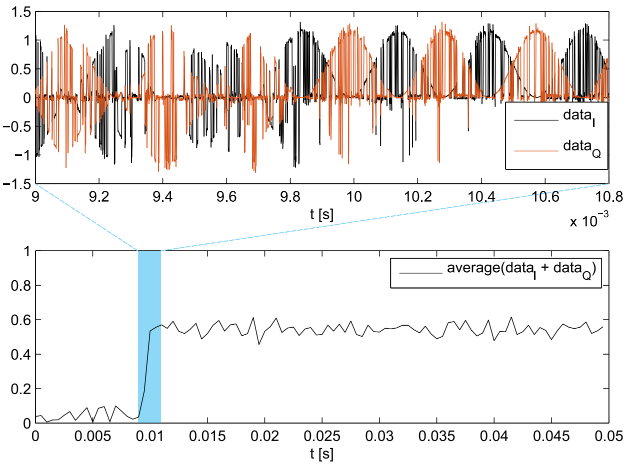

Figure 6. (a) data, (b) lockdetect output signals

If *lockdetect* value exceeds a specified threshold level a signal *DLL_lock* is generated. Otherwise *DLL_not_locked* signal is generated and shift corresponding to the number samples in one chip is executed. Synchronization process is then repeated.

After the successful synchronization is achieved a value $\tau_{TID}(m)$ is computed. It corresponds to the time shift of the received pseudorandom signal from its reference (zero) value. *TID* index indicates an identifier of the transmitter. The value m is incremented with every N samples.

After the synchronization of all transmitters is achieved and all corresponding time offsets $\tau_{TID}(m)$ are obtained a calculation of time differences of signal arrival for each signal (transmitters) pair are calculated. This calculation is possible only under the condition that all transmitters send their signals at the same time. For transmitter identifiers A and B is the time difference calculated as follows:

$$\Delta\tau_{AB}(m) = \tau_A(m) - \tau_B(m) . \tag{6}$$

The obtained value $\Delta\tau_{AB}(m)$ is then averaged in the interval of N_3 in the following way:

$$\Delta T_{AB} = \frac{1}{N_3}\sum_{m=1}^{N_3}\Delta\tau_{AB}(m) , \tag{7}$$

where ΔT_{AB} is the resulting time difference for the pair of transmitters A and B.

The interval N_3 is chosen empirically based on the nature of the movement. For stationary object it is advisable to choose a long interval N_3, which results in more accurate position determination. For moving objects must be interval N_3 shorter.

III. Practical Experiments

Practical experiments were performed using RTL-SDR receivers. It is a software defined receiver designed for a frequency range from approximately 30 MHz to 1.7 GHz. The output signal is provided as I and Q samples with a resolution of 8 bits and a sampling frequency up to 2.8 MHz. During the experiments, the sampling frequency was set to 2.0 MHz.

Transmitters were represented by two channel Arbitrary Waveform Generator (AWG), whose output signals were combined in a RF signal combiner R&S DVS 342.1014.50 and connected to the receiver antenna connector of the RTL-SDR. Each channel represents one transmitter with corresponding PN sequence modulated using Binary Phase Shift Keying (BPSK) to a carrier wave of a frequency f_c = 60 MHz. The time difference between the transmitted waveforms were adjusted using a delay of one of the channels inside the generator. As the pseudo-random sequences Gold codes of length 1023 chips were used. Chip frequency was set to f_{chip} = 200 kHz.

During the experiments different values of time differences corresponding to different distances between transmitters were set in the generator. Proposed algorithm determined time differences based on the method described above. Constants N, N_2 and N_3 were set to the values N = 100, N_2 = 2000 and N_3=20. Subsequently, the set and measured values were compared and measurement errors were calculated. The measured results are shown in Table 1.

Other experiments were carried out to determine the influence of spacing of generated replicas on achieved accuracy and synchronization speed. Measurements were performed for spacing of generated replica of 1/2, 1/5 and 1/10 chip length. Greater separation between the generated replicas provide faster synchronization with the received signal with less precision tracking code. Smaller spacing on the contrary, provides accurate tracking code. However, when dynamically changing signal is processed the synchronization can be easily lost. Data in Table 1 correspond to the spacing of replicas by ½ of chip length.

TABLE I. EXPERIMENTAL RESULTS

Set values [µs]	Measured values [µs]	Errors [µs]
0.000	0.064	0.064
0.333	0.535	0.201
0.667	1.000	0.333
1.000	1.193	0.193
1.333	1.530	0.196
1.667	1.807	0.140
2.000	2.134	0.134
2.333	2.129	-0.205
2.667	2.594	-0.073
3.000	2.792	-0.208
3.333	2.995	-0.338
6.667	6.847	0.180
16.667	16.936	0.269
33.333	33.287	-0.046
50.000	50.084	0.084
66.667	67.030	0.363

Comparison of the average errors of time difference determination and a corresponding distance error for spacing of replicas of 1/2, 1/5 and 1/10 of the chip is shown in Table 2.

TABLE II. INFLUENCE OF REPLICA SPACING

Replica spacing [chip]	1/2	1/5	1/10
Average error [µs]	0.081	0.032	0.033
Average error [m]	24.3	9.686	9.963

IV. CONCLUSIONS

The paper describes the design, implementation and experimental verification of the algorithm intended for software-defined receiver RTL-SDR in a function of radio positioning system based on the method TDOA. The behavior and properties of the proposed algorithm were verified by practical experiments. The measured results show that the average accuracy achievable with the used low-cost receiver is 10 m. However, the measurement was performed in laboratory conditions without any external interferences. Achieved accuracy depends on the mutual spacing of generated early, prompt and late replicas of pseudorandom signal. However, improvement of accuracy results in a degradation of dynamic properties of the system.

The next step is a parallel processing of signals incoming from transmitters. Each transmitters will have corresponding branch in parallel processing algorithm. Implementation is performed using OpenCL (Open Computer Language) together with a GPU (Graphic Processing Unit). This solution leads to the significantly shorter time of position determination process.

ACKNOWLEDGMENT

The work presented in this article has been supported by the Ministry of Defence of the Czech Republic (UoD development program "Research of sensor and control systems to achieve battlefield information superiority") and the Ministry of Education, Youth and Sports of the Czech Republic (UoD student research program “Implementation of modern technologies in avionic systems”).

REFERENCES

[1] D. Dardari, E. Falletti, M. Luise, “Satellite and terrestrial radio positioning techniques: A signal processing perspective,” Academic Press, 2012, p.432.

[2] J. W. M. Rogers, C. Plett, I. Marshland, "Radio frequency system archytecture and design," Artech House Publishers, 2013, p.302.

[3] K. Borre, D. M. Akos, N. Bertelsen, P. Rinder, S. H. Jensen, "A Software-Defined GPS and Galileo Receiver: A Single-Frequency Approach," Birkhäuser, 2007, p.176.

[4] B. W. Parkinson, J. J. Spilker Jr., "Global positioning system: Theory and applications," AIAA, 1996, p.1413.

[5] J. W. Betz, "Engeneering satellite-based navigation and timing: Global navigation sattelite systems, signals, ans recievers,"IEEE PRESS, 2016, p.672.

[6] M. S. Grewal, A. P. Andrews, Ch. G. Bartone, "Global Navigation Satellite Systems, Inertial Navigation, and Integration," 3th edition, Wiley-Interscience, 2013, p.608.

[7] S. Gleason, "GNSS Applications and Methods,"Artech House, 2009, p.508.