IOP Publishing Measurement Science and Technology
Meas. Sci. Technol. **31** (2020) 055001 (7pp) https://doi.org/10.1088/1361-6501/ab63e9

Solution for wireless time synchronization using sub-Nyquist sampling rates

Lukas Christensen[1] **and José M G Merayo**

Division of Measurement and Instrumentation Systems, DTU Space, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark

E-mail: lamc@space.dtu.dk

Received 8 September 2019, revised 2 December 2019
Accepted for publication 19 December 2019
Published 31 January 2020

Abstract

This article presents an accurate low-cost solution for wireless time synchronization between different systems. The results are obtained using commercial-off-the-shelf 2.4 GHz transceiver components and a novel approach to the sampling of pseudo-random number code modulated signals that enables highly accurate estimation of signal arrival times using simple hardware. The method consists of sampling incoming signals at a sub-bit-rate frequency and subsequent data processing that estimates the shape of the pulse-form of each individual data bit. Based on this, the arrival time of the signal can be determined, with estimates achieving higher accuracy with increasing code lengths regardless of bit-rate. This means that accurate measurements can be achieved with modest hardware running at sample rates in the low megahertz range. The developed solution is thereby able to achieve sub-nanosecond resolution using a 32767-bit Gold code modulated signal with a chirp-rate of 3.2767 MHz.

Keywords: time synchronization, gold codes, compressive sampling, fractional sampling

(Some figures may appear in colour only in the online journal)

1. Introduction

Many modern day applications require accurate time synchronization between devices not connected by physical means. Examples include distributed measurement systems, navigation systems, communication systems, and many others. The synchronization is generally achieved by measuring the arrival time of a wireless signal, transmitted from some reference station, at the different devices, and then using the deviation from the expected value to adjust the individual clocks.

A commonly used reference is the *Global Positioning System* (GPS) and other global navigation satellite systems. For GPS the arrival time estimation is achieved by having each satellite emit a navigation message that is modulated using a 1023-bit Gold code and then having the receivers perform a cross correlation between the received signal and a local copy of the encoding [1]. This results in a rough estimate of the time delay caused by the distance between the satellites and the receiver, which can then be used to estimate the time offset from the GPS clock if combined with time delay measurements from multiple other satellites [2].

While the code-based method is reliable, it is often not very accurate and needs to be supported by other means such as measuring the phase of the carrier wave of the wireless signal if high accuracy is needed [3, 4]. Phase measurements have one significant disadvantage in that they are associated with ambiguities that need to be resolved before the delay can be accurately determined. Methods for this exist, but they are often complicated and computationally expensive meaning that severe limits are placed upon the hardware performing the calculations [5].

This work presents a technique that allows for precise arrival time estimation of GPS and other Gold code modulated signals without using carrier phase information and with a low computational overhead. The technique uses sampling frequencies very close to the bit-rate of the reference signal and thus well below the Nyquist rate, which means that it can be implemented using simpler hardware than would otherwise be required. Because of this, the method potentially allows for simple and low cost hardware being able to achieve performance levels that are normally reserved for prohibitively expensive equipment.

[1] Author to whom any correspondence should be addressed.

Section 2 presents the details of the method and describes the underlying theory. In section 3 an implementation of the method in actual hardware is presented and its performance is described in section 4. Finally, section 5 provides a discussion on the effectiveness and viability of the method itself as well as the developed hardware solution.

2. Method

2.1. Gold code timing estimation

Gold codes are special pseudo-random binary sequences that have a number of properties that make them interesting for communication purposes [6]. These include the following:

- Being easy to generate in hardware using shift-registers.
- Having well defined single peak auto-correlation functions.
- Individual codes from a single set are approximately orthogonal to each other leading to low levels of cross-correlation.

The codes are constructed by combining the output of two maximum length sequence *linear feedback shift register* (LFSR) configurations. An example would be the sequences described by the following two characteristic polynomials [7]:

$$P_1(x) = 1 + x^8 + x^{15} \tag{1}$$

$$P_2(x) = 1 + x^9 + x^{11} + x^{12} + x^{13} + x^{14} + x^{15}. \tag{2}$$

Combining the output of these results in a 32767-bit Gold code. By changing the initial values of the LFSRs, different codes can be produced using the same hardware with the different codes all being part of the same set. If multiple signals, each modulated using its own Gold code, are present in a given recording it is possible to extract information from each of them individually because of the approximate orthogonality. The arrival time of each signal can also be estimated by performing a cross-correlation between the received signal and the Gold code used for modulation, with the time of maximum correlation corresponding to the arrival time. The accuracy achievable in this way is proportional to the sampling rate, which is commonly set to frequencies significantly higher than the code bit rate [1]. By utilizing upsampling better accuracy can be achieved, but it requires significantly more computations as well as sampling frequencies above the Nyquist rate [8].

2.2. Fractional sampling

Figure 1(a) shows an ideal square wave function representing a binary signal. This can be considered to be the result of a convolution between a series of delta functions (figure 1(b)) and a single square wave pulse-form as represented in figure 1(c), i.e.

$$s(t) = c(t) * p(t) \tag{3}$$

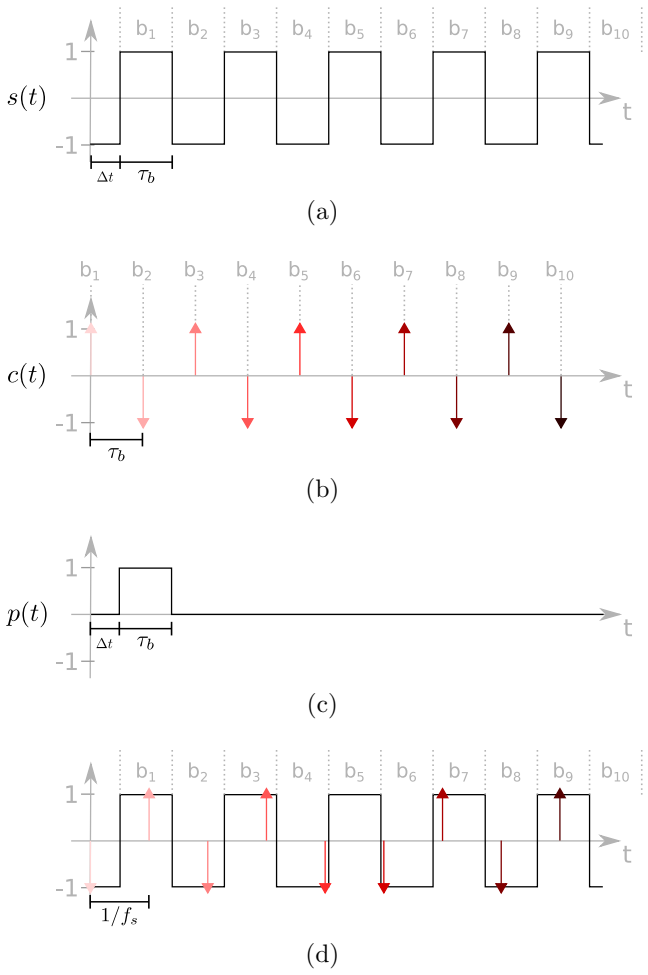

Figure 1. A square signal, $s(t)$. (a) can be considered as the convolution of a digital code represented by delta functions, $c(t)$, (b) and a square pulse-form, $p(t)$, (c). The arrows in (d) correspond to the sampling points when performing fractional sampling and, as can be seen, bit 5 is skipped by the sampling process.

with $s(t)$ being the binary signal, $c(t)$ being the modulation delta functions and $p(t)$ being the pulse-form. Any delay Δt of the binary signal can be described by a corresponding time shift of the pulse-form.

Suppose the 10-bit-long signal of figure 1(a) is transmitted at a bit-rate of $f_b = 1/\tau_b$, where τ_b is the length of each bit of the modulation code, and is digitized with a sampling frequency of $f_s = f_b$. Because the signal can be considered as the sum of 10 individually shifted copies of the pulse-form, each sample contains information about 10 different time positions of the pulse-form. Furthermore, because the sampling frequency is the same as the bit rate, each subsequent sample contains information about the same 10 positions because the time difference between each sample is equal to time shift between each bit of the signal. This is illustrated in figure 2(a) where the brightness of the arrows correspond to when the sample was taken, with brighter tones corresponding to earlier samples, similar to figure 1(b). Because of this sparse sampling of the pulse-form the arrival time of the signal cannot be determined better than to within $\pm\tau_b$. Increasing the sampling rate to integer multiples of f_b will improve the accuracy by an

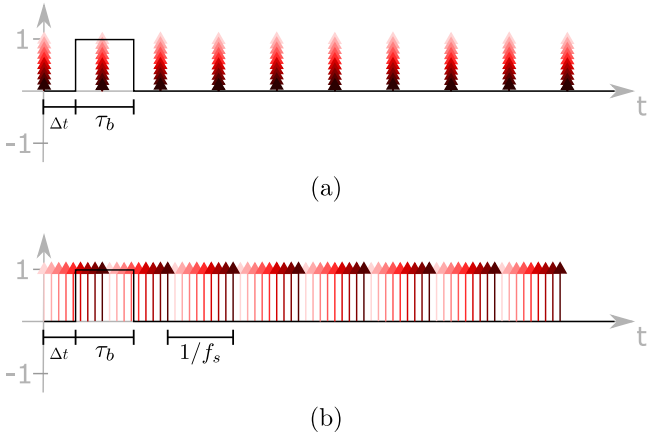

Figure 2. (a) shows the pulse-form information contained in each sample when sampling at a rate of f_b. (b) shows similar information when using a sample rate of $0.9f_b$.

equal factor, but the pulse-form information contained in each sample will still overlap as before.

Now, if the signal is sampled at a fractioned rate of e.g. $f_s = 0.9f_b$ instead, something interesting occurs: as illustrated in figure 1(d), bits 5 and 10 of the signal are skipped by the sampling process while all other bits are sampled in different locations from each other. Similar to before each sample corresponds to 10 equally spaced time locations of the pulse-form, however, because of the fractional sampling frequency there is no overlap of information between the different samples. This is illustrated in figure 2(b). Because of the tighter spacing of the pulse-form samples, the arrival time of the signal can now be determined with an accuracy of $\pm\tau_b/9$. In general, if the code has a length of N bits, by choosing the sampling frequency as

$$f_s = f_b \frac{N-1}{N} \tag{4}$$

a potential temporal accuracy of $\tau_b/(N-1)$ can be attained. Applying this method to GPS ($f_b = 1.023$ MHz, $N = 1023$) leads to a time resolution of $\approx$1 ns resulting a distance resolution of $\approx$ 30 cm. This is a significant improvement over most code-based GPS receivers [1] and it is achieved using a sub-Nyquist rate of sampling.

However, before the arrival time can be estimated it is necessary to first perform a reconstruction of the pulse-form.

2.3. Arrival time estimation

One way to recover the pulse-form is to perform a deconvolution. However, even if this is done in the spectral domain it is a processing intensive procedure [9]. Another possible option is to mix the recorded signal with two versions of the modulation code: one on time, and one delayed by one sample. Using the previous example, the modulation code **c** and recorded signal **r** are respectively equal to

$$\mathbf{c} = [+1,-1,+1,-1,+1,-1,+1,-1,+1,-1]$$

$$\mathbf{r} = [-1,+1,-1,+1,-1,-1,+1,-1,+1]$$

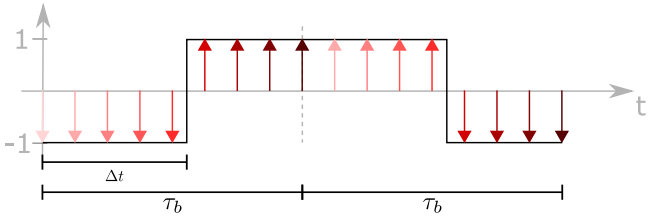

Figure 3. Envelope of the reconstructed pulse-form.

using a non-return-to-zero representation. Figure 1(d) shows the sampling points that have been used to construct the recorded signal. Element-wise multiplication of the signal with the shifted code copies (ignoring the final sample of the code) results in

$$\mathbf{m_0} = [-1,-1,-1,-1,-1,+1,+1,+1,+1]$$

$$\mathbf{m_1} = [+1,+1,+1,+1,+1,-1,-1,-1,-1]$$

respectively for the on-time and the delayed codes. The mixing with the on time code extracts the information of the first τ_b interval of the pulse-form, while the mixing with the delayed copy extracts the information of the next τ_b interval and so on. Concatenating $\mathbf{m_0}$ and $\mathbf{m_1}$ then results in a representation of the first $2\tau_b$ seconds of the pulse-form sampled at $\tau_b/9$ second intervals. This is illustrated in figure 3.

From this it can be seen that the example code arrived at a time between $5/10\tau_b = 0.5\tau_b$ and $6/10\tau_b = 0.6\tau_b$, which is the expected result. In general, defining the first sample index of the mix that is equal to $+1$ as K_f, the delay will lay in the range of

$$\frac{K_f - 1}{N}\tau_b \leqslant \Delta t \leqslant \frac{K_f}{N}\tau_b. \tag{5}$$

The example uses a very simple code of alternating ones and zeroes, which does not represent a realistic situation utilizing *pseudo-random number* (PRN) codes for the modulation. Figure 4(a) shows the pulse-form reconstruction resulting from using a 32767-bit PRN Gold code generated using $P_1(x)$ and $P_2(x)$ as the generating LFSRs.

As can be seen, the pseudo-random nature of the code means that the reconstruction is less clean and alternates between ±1 where the initial example consisted of solely -1. However, since the deviations from the ideal are pseudo-randomly distributed the result can be improved by performing a simple smoothing of the data. This is shown in figure 4(b) where the data has been processed using a moving average filter with a span of 100 samples. This clearly improves the shape considerably. With the pulse-form reconstructed, the delay can be determined by noting when the value of the reconstruction crosses a certain threshold. That is, detecting the rising edge. Accuracy can be improved by noting the falling edge as well.

It should be noted that the proposed method is only able to determine fractional arrival times, i.e. times $\Delta t \leqslant \tau_b$ after the start of recording. If the signal is further delayed it is necessary to first determine the integer arrival time, i.e. how many full bits the signal is delayed, before performing the reconstruction

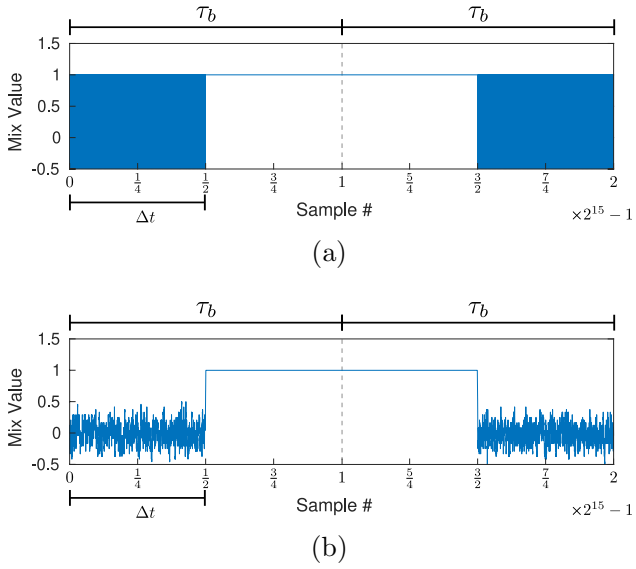

Figure 4. Pulse-form reconstructions using a 32767-bit Gold code. (a) is the raw mix of the recording and the local code copies whereas (b) is the same mix after it has been smoothed using a running average filter 100 samples wide.

and then offsetting the data accordingly. This then leads to the following formulation of the arrival time estimate

$$\Delta t \approx \tau_b \left[K_i + \frac{K_f}{N} \right] \tag{6}$$

where K_i is the integer arrival time and K_f is the fractional arrival time, both in the unit of samples. The integer part can be determined using conventional methods for signal delay estimation such as performing a cross-correlation between the received signal and the modulation code. However, it should be noted that when fractional sampling is performed the singular peak that is normally present in the cross-correlation function will be split up over two adjacent samples. This can be intuitively understood by looking at the code-data products $\mathbf{m_0}$ and $\mathbf{m_1}$: the cumulative sums of these essentially each represent one sample of the cross-correlation, and since both contain part of the signal, the correlation peak will be distributed over the two samples. Determining which of the two peaks represents the actual integer delay can be achieved by, for example, testing both possibilities individually. Another approach is to mix the recording with additionally delayed code copies to extend the time period covered by the pulse reconstruction method.

3. Implementation

To test the time synchronization method, a custom 2.4 GHz radio transceiver board has been designed and constructed from easily accessible low-cost parts. The main components are as follows:

- Maxim integrated MAX2830 analog front-end.
- Maxim integrated MAX5851 *digital-to-analog converter* (DAC).

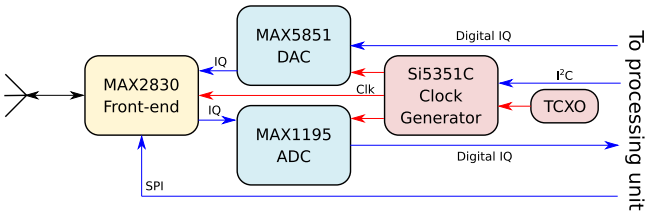

Figure 5. Overview of the transceiver hardware.

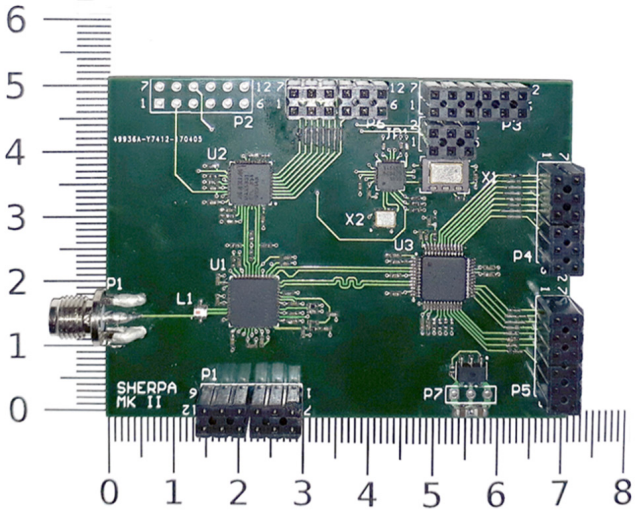

Figure 6. Image of the transceiver hardware. The scale is in units of centimeters.

- Maxim Integrated MAX1195 *analog-to-digital converter* (ADC).
- Silicon labs Si5351C clock generator.
- TXC Corporation 25 MHz *temperature compensated crystal oscillator* (TCXO).

The output of the analog front-end is connected to a standard quarter wave dipole antenna designed for WiFi applications. The antenna gain is 2 dBi and the front-end chip has a noise figure of 4 dB and a transmit power of up to 17.1 dBm [10].

Figure 5 shows an overview of the hardware design whereas figure 6 presents an image of the actual hardware. The combined cost of each transceiver prototype, excluding the processing unit, is less than €40.

The transceiver board is connected to a Xilinx Spartan-6 FPGA that generates the digital baseband signals needed for transmission and works as a data bridge sending the baseband signals to a PC over USB 2.0 when receiving. The FPGA also takes care of the digital configuration of the different transceiver components. The signal processing could in principle also be implemented directly in the FPGA, however, to reduce development time the processing has been implemented in software running on a host PC.

When transmitting, the system uses the LFSRs described by equations (1) and (2) to generate a 32767-bit Gold code which is transmitted at a 3.2767 MHz bit rate using *binary phase-shift keying* (BPSK) modulation. When receiving, the incoming signal is sampled at a rate of 3.2766 MHz (see

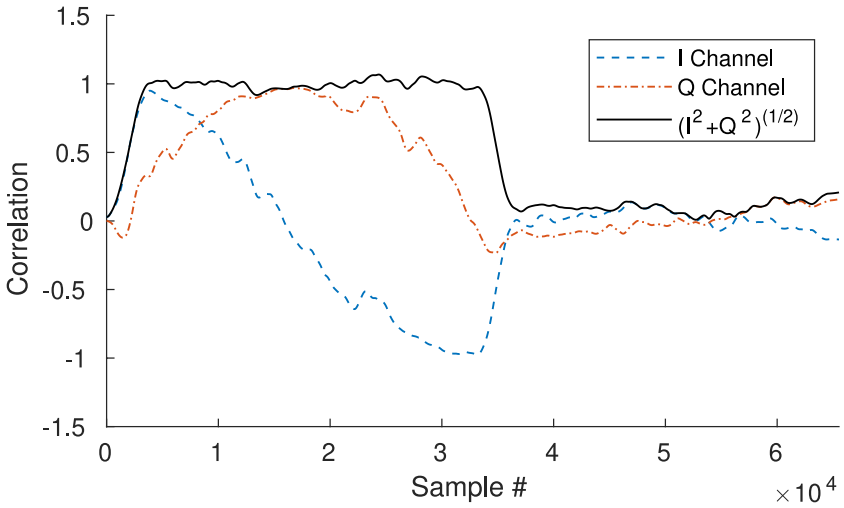

Figure 7. An example of a recovered pulse-form. Receiver and transmitter were separated by 100 cm and each had their own clock source.

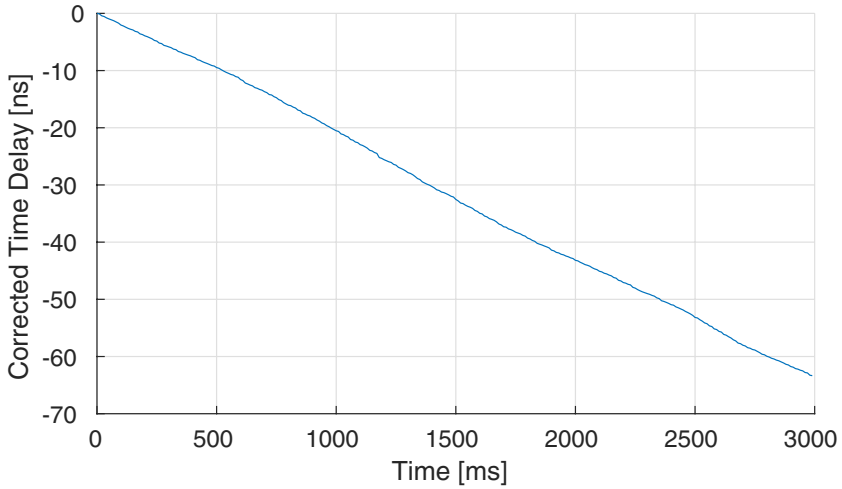

Figure 8. The recovered time delay between a transmitter and a receiver separated by 100 cm, each with their own clock source. Note that the initial delay value has been subtracted from the data to have the graph start at zero delay. The slope of the graph corresponds to an oscillator offset of 21.7 ppb, or 52.1 Hz.

equation (4)) and subsequently mixed with a copy of the Gold code.

When performing tests two transceivers are used, one for transmission and one for reception, each connected to their own FPGA. The transceivers each have an on-board clock source accurate to within 280 ppb [11] in the form of the TCXO which drives both chirp and sample rates as well as the 2.4 GHz modulation/demodulation. However, to allow for a larger variety of setups, the transceivers also allow for bypassing the on-board clock in order to drive the frequency synthesis from an external source.

4. Results

In figure 7 the recovered pulse-form is shown for a test where the emitting and receiving transceivers were separated by a distance of 1 meter. No carrier tracking is performed in order to test the robustness of the method and the pulse-form is therefore generated by individually mixing the I and Q channels with the Gold code, smoothing, and then adding the results in quadrature.

From the I and Q channels it is clear that there is a slight mismatch in carrier frequency between the emitter and the receiver as is to be expected. Since each complete transmission of the Gold code takes 10 ms, the offset is approximately 50 Hz based on the I and Q periodicity.

It is clear that the recovered pulse-form is somewhat distorted compared to the theoretical result of figure 4(b). A significant cause of this is system noise, but an additional contributing factor is the fact that the ADC has been configured to operate in a 1-bit mode. This was selected as it greatly simplifies the hardware design and, in addition, represents a worst-case scenario for the pulse-form recovery and therefore tests the robustness of the fractional sampling method. Regardless, the pulse-form is well defined enough that the rising and falling edges can easily be identified.

Recovering the delay for several code cycles in the same setup results in figure 8. The offset between the transmitter and receiver clock frequencies causes the estimated signal arrival

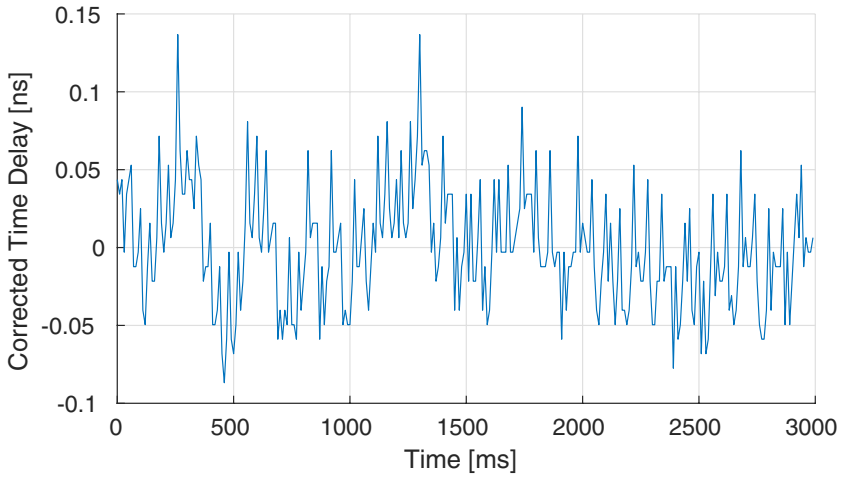

Figure 9. The recovered time delay between a transmitter and a receiver separated by 10 cm using a common clock source. Note that the mean has been subtracted from the data to center it around zero delay. The standard deviation is equal to 37.7 ps with a maximum deviation of 136.7 ps.

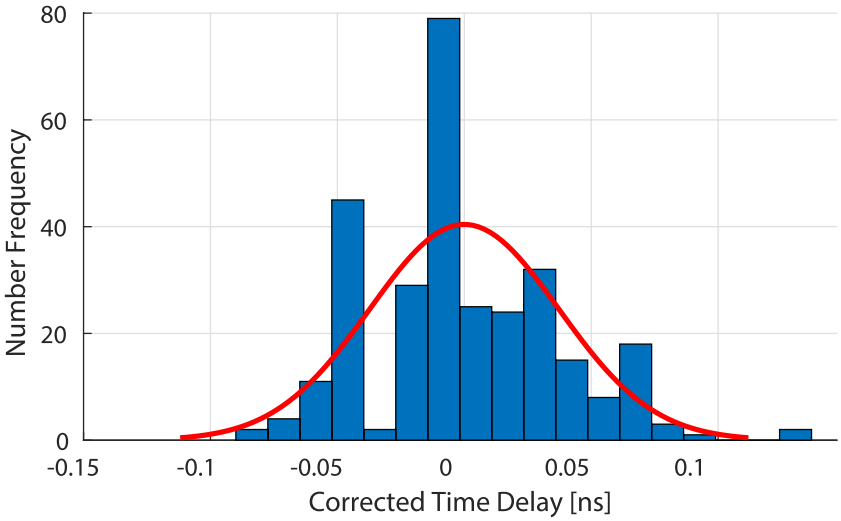

Figure 10. Histogram of the delay data shown in figure 9 together with the best fitting normal distribution.

time to change with each transmission as the two clocks drift farther and farther apart in time. This is evident in the data as a linearly varying time delay.

The graph has a slope of -0.0217 ns ms^{-1} equal to a frequency deviation of 21.7 ppb. For a carrier frequency of 2.4 GHz this corresponds to 52.1 Hz, which matches what was indicated from the raw I and Q channels.

The slope of the plot makes it possible to synchronize clock frequencies across the two transceivers to a high degree of accuracy as even this 21.7 ppb offset shows up as a very clear tendency in the data. However, based on this data alone it is difficult to conclude much regarding the potential accuracy of the fractional sampling method: even if a linear function is fitted to and subsequently subtracted from the data there is still significant variations in the estimated delays caused by short-term oscillator drift.

A second test was therefore made where both transceivers were connected to the same clock source. To maximize clock signal integrity the inter-transceiver distance was reduced to 10 cm so signal lines could be kept as short as possible. The recovered time offset is displayed in figure 9. Note that because of delays and biases in the hardware, the results of this test are not indicative of the absolute time synchronization performance of the system. However, the results can still be used to evaluate noise characteristics of the sampling method and hence the potentially attainable precision given perfectly calibrated hardware.

An almost constant time delay is observed with what appears as randomly distributed noise. A histogram of the variation is presented in figure 10. Clearly, the variations are not Gaussian, but the normal distribution serves as a decent first approximation.

The delay exhibits a standard deviation of 37.7 ps and a maximum deviation of 136.7 ps from the mean. By equation (5) the minimum theoretically measurable time step is 9.3 ps, so the standard and maximum deviation approximately represents offsets of 4 and 15 samples from the ideal, respectively.

5. Discussion

The experiments performed with the developed test platform confirm that the fractional sampling method works and that performance similar to the theoretical accuracy is achievable even on simple hardware. One contributing factor to the observed accuracy being less than the theoretical value is the fact that the pulse-form recovery relies on the modulation code having an approximately equal amount of ones and zeroes: information can only be extracted when there is a change of bit value in the code. This means that if the signal delay corresponds to a section of the PRN code that consists of, for example, 10 logical ones in a row, it is not possible to determine exactly which one of the 10 samples that represents the actual delay. As such, there is an inherent decrease in the accuracy of the method which depends on the local shape of the modulation code. Even with this reduced accuracy, achieving a time resolution similar to that of the single-oscillator test using a simple cross-correlation technique would require a sampling frequency of 26.5 GHz [1]. However, the frequency could be reduced by employing methods such as upconversion at the expense of requiring more processing power [8]. Still, the fractional sampling method represents a significant reduction in the required sampling frequency relative to even just the Nyquist rate.

The fact that the data was only digitized to a 1-bit accuracy results in an reduction of the signal-to-noise ratio of about 2 dB [12]. However, in the performed tests the receiving and transmitting receivers were placed so close together that the only other significant causes of signal degradation would be system noise and potentially multipath effects. As such, the presented data is only representative of high signal-to-noise situations and it therefore cannot be guaranteed that the method will perform as well in all scenarios. Future work therefore entails classification of the system at different noise and signal levels as well as further investigation of the fractional sampling method itself using different hardware with other design parameters. Even given these considerations the hardware performance is still admirable considering that none of the used components were designed for time critical operations, but rather for use in general communications systems.

6. Conclusion

In this article a fractional sampling method has been described that enables highly accurate estimates of the arrival times of PRN code modulated wireless signals without using the phase of the carrier wave. This has been validated by the construction of a low-cost radio transceiver and subsequent hardware testing of the delay estimation method, which has shown to produce results similar to the theoretical performance.

Because the approach is built on sampling rates that are similar to the bit rate of the modulation code it is possible to implement it on simple hardware. This enables the development of simple, low-cost time synchronization systems that can be utilized for a multitude of different purposes ranging from temporal sensor alignment to time of flight-based navigation.

ORCID iDs

Lukas Christensen https://orcid.org/0000-0001-7561-4611

References

[1] Borre K, Akos D M, Bertelsen N, Jensen S H and Rinder P 2007 *A Software-Defined GPS and Galileo Receiver: a Single-Frequency Approach* (Boston, MA: Birkhäuser)
[2] Weiss M A, Petit G and Jiang Z 2005 A comparison of gps common-view time transfer to all-in-view *Proc. 2005 IEEE Int. Frequency Control Symp. and Exposition* p 5
[3] Yao J, Skakun I, Jiang Z and Levine J 2015 A detailed comparison of two continuous GPS carrier-phase time transfer techniques *Metrologia* **52** 666–76
[4] Yao J and Levine J 2016 A study of GPS carrier-phase time transfer noise based on NIST GPS receivers *J. Res. Natl Inst. Stand. Technol.* **121** 372–88
[5] Kim D and Langley R 2000 GPS ambiguity resolution and validation: methodologies, trends and issues *7th GNSS Workshop—Int. Symp. on GPS/GNSS (Seoul, South Korea)*
[6] Gold R 1967 Optimal binary sequences for spread spectrum multiplexing (corresp.) *IEEE Trans. Inf. Theory* **13** 619–21
[7] Peterson W W and Weldon E J 1972 *Error-Correcting Codes Error-Correcting Codes* 2nd edn (Cambridge, MA: MIT Press)
[8] Calvo-Palomino R, Ricciato F, Repas B, Giustiniano D and Lenders V 2018 Nanosecond-precision time-of-arrival estimation for aircraft signals with low-cost sdr receivers *Proc. 17th Acm/ieee Int. Conf.* pp 272–7 (https://arxiv.org/pdf/1802.07016.pdf)
[9] Sherrick J D 2005 *Concepts in Systems and Signals* 2nd edn (Upper Saddle River, NJ: Prentice Hall)
[10] Maxim Integrated 2011 MAX2830 2.4GHz to 2.5GHz 802.11g/b RF Transceiver, PA, and Rx/Tx/Antenna Diversity Switch Rev 2 Data Sheet (San Jose, CA: Maxim Integrated)
[11] TXC Corportion 2015 TXC Corportion 2015 Precise SMD Temperature Compensated Crystal Oscillators 7.0 × 5.0 × 2.0 mm 7N Series (10 pad) *Data Sheet* (Taipei, Taiwan: TXC Corporation)
[12] Stein M, Wendler F, Mezghani A and Nossek J A 2013 Quantization-loss reduction for signal parameter estimation *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing* pp 5800–4