ARL-TN-1140 • Oct 2022

Performance of Zadoff–Chu Sequences for Time Synchronization

by Timothy J Garner, William H Davis, and David A Wikner

NOTICES

Disclaimers

ARL-TN-1140 ● OCT 2022

Performance of Zadoff–Chu Sequences for Time Synchronization

by Timothy J Garner and David A Wikner
DEVCOM Army Research Laboratory

William H Davis
University of Florida

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED (From - To)**
October 2022	Technical Note	March 2022–August 2022

4. TITLE AND SUBTITLE
Performance of Zadoff–Chu Sequences for Time Synchronization

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Timothy J Garner, William H Davis, and David A Wikner

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
DEVCOM Army Research Laboratory
ATTN: FCDD-RLS-EA
Adelphi, MD 20783-1138

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TN-1140

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
primary author's email: <timothy.j.garner.civ@army.mil>.

14. ABSTRACT
Distributed coherent beamforming requires close alignment of transceivers in frequency, time, and phase. Common methods for time synchronization rely on knowledge of transmission and reception times of a known signal. The arrival time is estimated by computing the correlation of the received signal with the known signal. An interpolation method, such as quadratic or higher-order polynomial interpolation, is used to estimate the location of the correlation peak with sub-sample precision. The choice of signal, the signal-to-noise ratio (SNR), and the interpolation method all affect the accuracy of the synchronization. The auto-correlation properties of Zadoff–Chu sequences make them good candidates for synchronization signals. The accuracy of the arrival-time estimation may be improved by increasing the length of the sequence or increasing the signal-to-noise ratio. At high SNRs, the accuracy may also be improved by using higher-order polynomial interpolation.

15. SUBJECT TERMS
synchronization, Zadoff–Chu sequences

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Timothy J Garner
Unclassified	Unclassified	Unclassified	UU	18	**19b. TELEPHONE NUMBER (Include area code)** 301-394-2705

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

1. Introduction

Distributed, phase-synchronous beam-forming arrays consist of a number of transceivers that set their transmit or receive phases in such a way that the transmitted signals arrive in a target direction in phase, or that the received signals from a specified direction are added in phase. Wireless time synchronization is needed for phase-synchronous operation of distributed radio nodes. Phase-synchronous operation means that nodes can generate and receive signals with a chosen phase offset between them. This requires that the offset between the clocks be known with an accuracy better than one-tenth of the carrier period, that is

$$|\Delta t - \Delta t_{\mathrm{est}}| < \frac{1}{10 f_{\mathrm{c}}}, \tag{1}$$

where Δt is the actual offset between the clocks, Δt_{est} is the estimated offset, and f_{c} is the nodes' carrier frequency. Timing accuracy with the Global Positioning System (GPS) is about 10 ns.[1] At carrier frequencies above 10 MHz, better accuracy than GPS provides is required for phase-synchronous operation. Figure 1 shows the accuracy requirement for phase-synchronous operation as a function of frequency. Note that if other sources of error are present, even better time accuracy may be required.

One method of remote clock synchronization is two-way time transfer.[2] In this method, nodes measure the transmission and arrival times of known synchronization signals according to their own clocks. The offset between the two clocks can be calculated from these times, as long as the delay is the same in both directions. A node correlates its received signal with the known synchronization signal to estimate the arrival time of the signal based on the correlation peak according to its own clock. Interpolation methods such as quadratic or higher-order polynomial interpolation are used to estimate the location of the correlation peak with subsample precision.[3] The accuracy of the estimation depends on the characteristics of the channel between the transmitter and receiver, the synchronization signal, and the interpolation method used to estimate the location of the correlation peak.

We studied the accuracy of the arrival-time estimation using Zadoff–Chu sequences. We used both computer simulation and benchtop testing with software-defined radios to measure the effect of sequence length, signal-to-noise level, and interpolation method on the accuracy and precision of the arrival time estimation. This is a

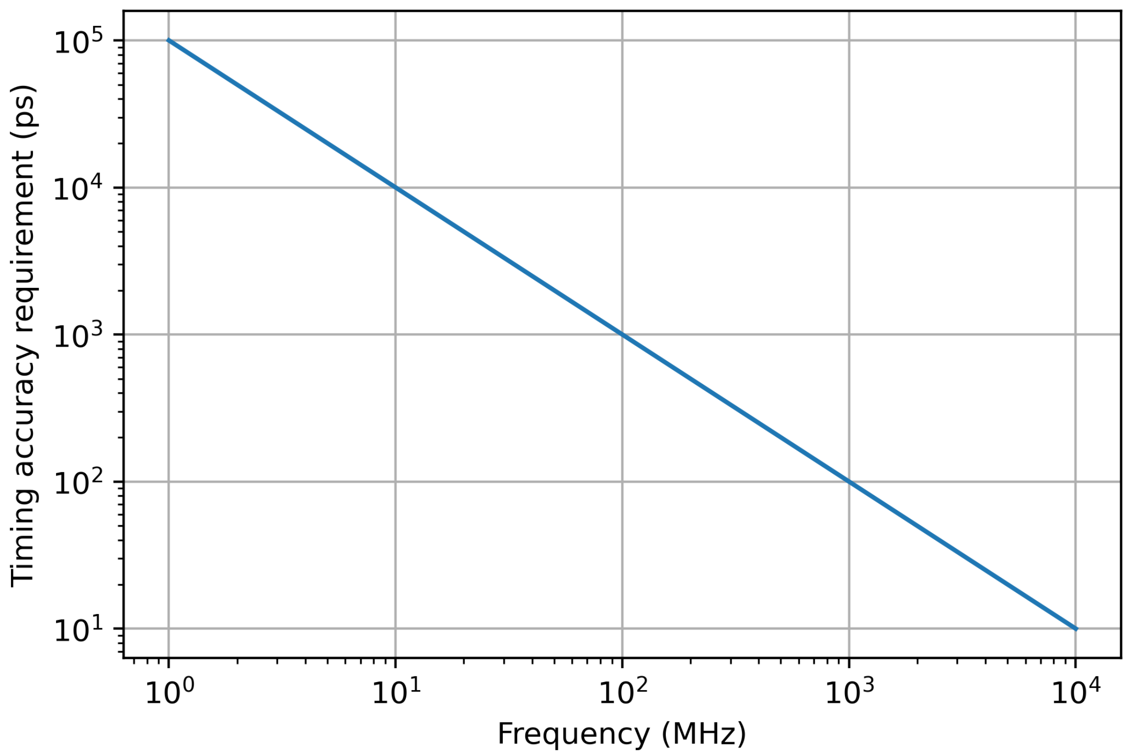

Fig. 1 Minimum timing accuracy requirement for phase-synchronous operation vs. frequency

key aspect of two-way time transfer, but we did not implement full two-way time transfer in this work. Section 2 explains the theory underlying our work, Section 3 presents results from simulations and hardware experiments, and Section 4 gives concluding remarks and suggests direction for future research.

2. Theory

2.1 Arrival-Time Estimation Method

Consider a transmitter and receiver pair whose clocks differ by Δt but run at the same rate. Let the transmitter's clock time be t, and let the receiver's clock time be $t' = t + \Delta t$. The transmitter sends a known, finite-length, discrete-time sequence $x[n]$ at time $t = 0$. Let the length of the sequence be N. Assume that the continuous-time version of the signal is given by Whittaker–Shannon interpolation as

$$x(t) = \sum_{n=0}^{N-1} x[n] \mathrm{sinc}\left(\frac{\mathrm{t} - \mathrm{nT}}{\mathrm{T}}\right), \tag{2}$$

where T is the interval or symbol rate of the sequence.

When the signal arrives at the receiver, it will be delayed by τ, thus the received signal according to the transmitter's clock is

$$x(t-\tau) = \sum_{n=0}^{N-1} x[n] \mathrm{sinc}\left(\frac{\mathrm{t} - \tau - \mathrm{nT}}{\mathrm{T}}\right). \tag{3}$$

If the signal is transmitted at $t = 0$ according to the transmitter's clock, it will arrive at the receiver at $t'_{\mathrm{rx}} = \tau + \Delta t$. According to the receiver's clock, the received signal plus noise is therefore

$$y(t') = x(t' - \Delta t - \tau) = \sum_{n=0}^{N-1} x[n] \mathrm{sinc}\left(\frac{\mathrm{t}' - \Delta \mathrm{t} - \tau - \mathrm{nT}}{\mathrm{T}}\right) + \mathrm{n}(\mathrm{t}'), \tag{4}$$

where $n(t')$ is the noise of the receiver and channel.

We assume that the receiver has an estimate of the expected arrival time of the synchronization pulse. This may be from an external time reference like a global navigation system or from a signal previously sent to the receiver for synchronization. The estimate is $t'_{\mathrm{rx,est}}$, according to the receiver's clock. The receiver starts taking samples at

$$t'_{\text{start}} = t'_{\text{rx,est}} - T_{\text{pad}} \tag{5}$$

and continues sampling until

$$t'_{\text{stop}} = t'_{\text{rx,est}} + (N-1)T + T_{\text{pad}}. \tag{6}$$

This accounts for the signal's duration of $(N-1)T$ and adds extra time T_{pad} at the start and end of the sampling to handle errors in the estimation of $t'_{\text{rx,est}}$. Increasing T_{pad} will increase the probability that the entire synchronization signal is captured in the sampling window at the expense of increased computation.

The receiver takes samples with period T_{rx}, which may differ from the transmitted symbol rate T. The discrete-time version of the synchronization sequence at sample period T_{rx} is generated by evaluating $x(t)$ from Eq. 2 at times $t = t'_{\text{start}} + mT_{\text{rx}}$, where m is the index and T_{rx} is sampling interval of the upsampled sequence. This yields the up-sampled sequence $x_{\text{rx}}[m]$. The received sequence is

$$y_{\text{rx}}[m] = y(t'_{\text{start}} + mT_{\text{rx}}), m \in 0, 1, 2, ...\lceil \frac{2T_{\text{pad}} + (N-1)T}{T_{\text{rx}}} \rceil. \tag{7}$$

The receiver correlates the received signal with the expected synchronization signal. The cross-correlation is

$$(y \star x)[k] = \sum_{m=-\infty}^{\infty} x^*[m] y_{\text{rx}}[m+k]. \tag{8}$$

The arrival time is estimated by locating the maximum value of $(y \star x)[k]$ and using polynomial interpolation with the adjacent points to estimate the true peak location.[3,4]

2.2 Zadoff–Chu sequences

We experimented with Zadoff–Chu sequences in complex baseband representation. The autocorrelation of Zadoff–Chu sequences is zero for nonzero offsets of an integer number of samples.[5] This gives the cross correlation in Eq. 8 a single unambiguous peak. We generated Zadoff–Chu sequences with the Commpy library for

Python.[6] The definition of the Zadoff–Chu sequence used by Commpy is

$$\begin{aligned} x_{\mathrm{ZC}}[n] &= \exp\left[-i\frac{\pi u n(n + N_{\mathrm{ZC}} \pmod 2) + 2q}{N_{\mathrm{ZC}}})\right] \\ n &\in 0, 1, ...N_{\mathrm{ZC}} - 1 \\ 0 &< u < N_{\mathrm{ZC}} \\ \mathrm{gcd}(\mathrm{N_{ZC}}, \mathrm{u}) &= 1 \\ q &\in \mathbb{Z} \end{aligned} \tag{9}$$

where q is a cyclic shift of the Zadoff–Chu sequence. We used exclusively $q = 0$ and $u = 7$ for this project.

3. Results

We computed estimated arrival times of Zadoff–Chu sequences of various lengths at different signal-to-noise levels. We sampled the received signal at 10 times the symbol rate of the transmitted sequence. The lengths of the Zadoff–Chu sequences were

$$N_{\mathrm{ZC}} \in 64, 256, 1024, 4096, 16386, 65536. \tag{10}$$

The longest two sequences were not evaluated computationally, because the program needed too much memory to run them. Future work could improve the memory requirements of the program and allow longer sequences to be evaluated.

For both computational and laboratory experiments, the transmit symbol rate was 1 MSps, and the receive sampling rate was 10 MSps. The carrier frequency was 1 GHz, and quadratic interopolation was used to estimate the peak location.

We used a Python program to conduct computational studies of the accuracy of the arrival-time estimation of Zadoff–Chu sequences. In the computational studies, we assumed that the transmitter and receiver have clocks that are offset by a fixed amount but run at the same rate. For each case, 31 trials were conducted, each with a random true arrival time. For each trial, the program computed an estimated arrival time and computed the difference between the true arrival time and the estimated time. We took the standard deviation of the arrival-time error to represent the expected error level of the estimation.

The laboratory experiments were done as loopback tests, so the transmit and receive clocks were the same, leaving only the delay between transmitter and receiver as unknown. A length of cable was connected from the Tx/Rx A port to the Rx A port of a National Instruments X310 software-defined radio. A 30-dB attenuator was also connected in line with the cable to protect the receiver. This gave a very high SNR at the receiver. The exact ratio was not measured, but it was greater than 40 dB. For each case, we ran 31 trials of the experiment and took the standard deviation of the arrival times to represent the error level of the estimation. This method differs from that used for the computational studies because the true arrival times in the experiment were unknown.

Figure 2 shows the standard deviation of the estimated arrival times as a function of sequence lengths from both computations and experiments. The computational results were for signal-to-noise ratios of $20, 30,$ and 40 dB, along with a special case with no added noise. The noiseless case is listed as SNR = ∞ dB. All cases show decreasing error level with increasing sequence length. The experimental results match closely with the noiseless computational results.

In general, the error decreased with increasing sequence length, but a roughly 1000-fold increase in sequence length was necessary for a 10-fold reduction in error. The results also showed strong sensitivity to SNR. A 10-dB increase in the SNR yielded a roughly five-fold decrease in error. There was a limit to the performance, even with no noise (or infinite SNR). This is likely caused by the approximations used in estimating the arrival time. The arrival time was estimated by fitting the points around the peak correlation to a quadratic. A higher-order polynomial interpolation yields further improvement in the minimum error. The performance of quadratic interpolation may also be improved by using a precomputed lookup table based on the autocorrelation of the synchronization signal.[3]

Figure 3 shows the standard deviation of arrival time error vs. SNR for second-, fourth-, and eighth-order polynomial interpolation. The error decreases for all three orders with increasing SNR. At lower SNRs, the second-order polynomial has the lowest error. At higher SNRs, eighth-order polynomial interpolation performs best. It is likely that the higher-order interpolation provides a better match to the actual shape of the correlation function of the Zadoff–Chu sequence, but the advantage is lost at higher noise levels.

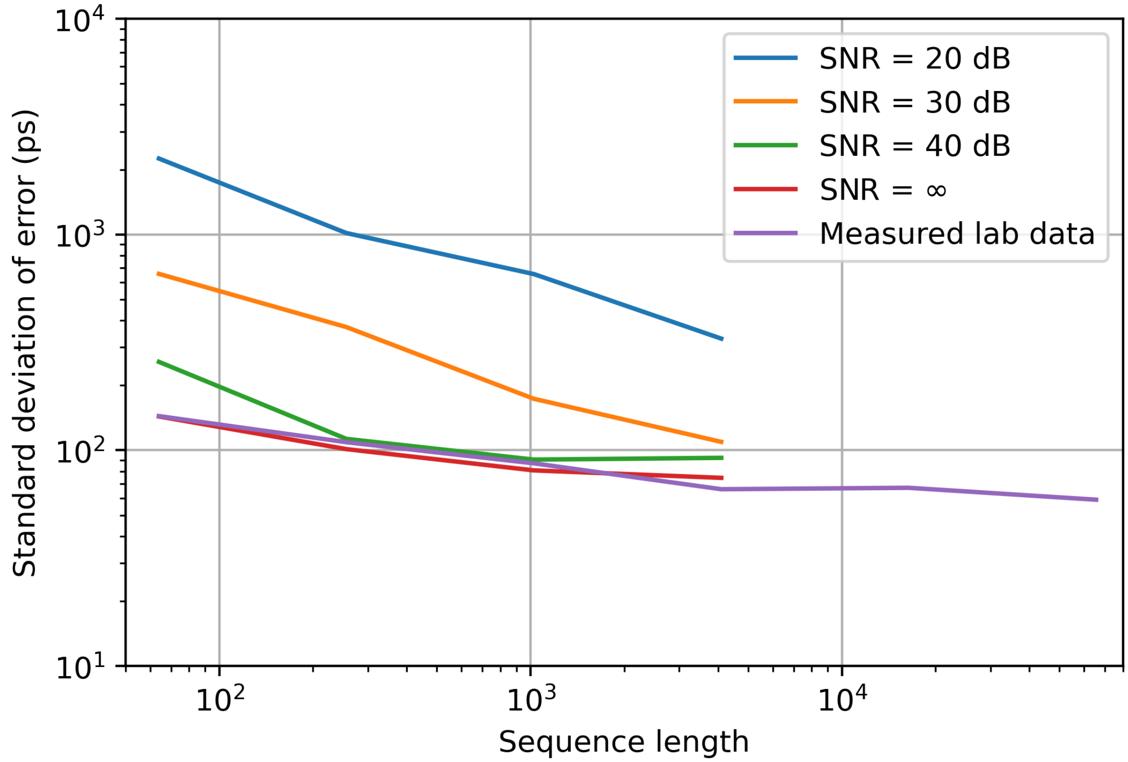

Fig. 2 Standard deviation of arrival time error vs. sequence length for Zadoff–Chu sequences. The first four traces (SNR = 20 dB, SNR = 30 dB, SNR = 40 dB, and SNR = ∞ dB) are computational results. The final trace shows results of a laboratory experiment.

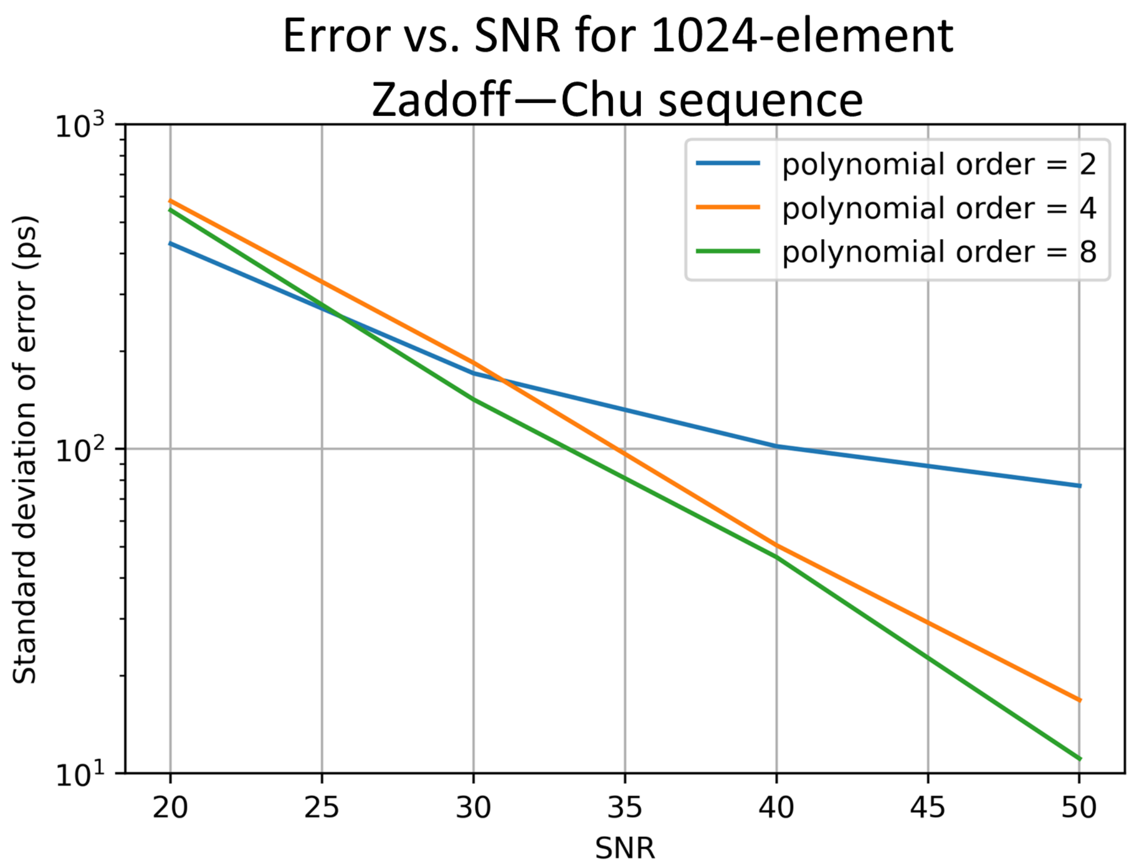

Fig. 3 Standard deviation of arrival time error vs. signal-to-noise in decibel ratio for 1024-element Zadoff–Chu sequence. The three traces show computational results with different orders of polynomial interpolation.

4. Conclusion

Zadoff–Chu sequences may give enough accuracy in time synchronization to allow for distributed beamforming. However, they will require a high SNR. A simple quadratic interpolation method is at least as accurate as higher-order polynomial interpolations at SNRs up to about 25 dB. The simplicity of this interpolation method makes it suitable for implementation on real-time field-programmable gate array hardware.

Future research may be done to investigate other types of sequences, like pseudo-random binary sequences transmitted using binary phase shift keying. The hardware needed to transmit these sequences may be simpler than that needed to transmit Zadoff–Chu sequences, because it would only need to generate two discrete phases. In contrast, transmission of a Zadoff–Chu sequence requires generation of an arbitrary phase.

Some sources of error were neglected in this work including clock frequency offsets, multipath channels, and interference that is not additive white Gaussian noise. Errors due to analog-to-digital and digital-to-analog conversions were also omitted from the computational studies. The performance shown in this report should therefore be viewed as a best-case limit. Further experimentation is necessary to determine how close real hardware operating in real over-the-air channels can come to this ideal performance.

5. References

1. Lewandowski W, Petit G, Thomas C. Precision and accuracy of gps time transfer. IEEE transactions on instrumentation and measurement. 1993;42(2):474–479.

2. Hanson DW. Fundamentals of two-way time transfers by satellite. In: Proceedings of the 43rd annual symposium on frequency control; p. 174–178.

3. Khader I, Bergeron H, Sinclair LC, Swann WC, Newbury NR, Deschênes JD. Time synchronization over a free-space optical communication channel. Optica. 2018;5(12):1542–1548.

4. (U) Prager S, Haynes MS, Moghaddam M. Wireless subnanosecond rf synchronization for distributed ultrawideband software-defined radar networks. IEEE Transactions on Microwave Theory and Techniques. 2020;68(11):4787–4804.

5. Chu D. Polyphase codes with good periodic correlation properties. IEEE Transactions on information theory. 1972;18(4):531–532.

6. Taranalli V, Trotobas B. Commpy: Digital Communication with Python. https://github.com/veeresht/CommPy; 2022. Accessed 2022-08-31.

List of Symbols, Abbreviations, and Acronyms

GPS	Global Positioning System
SNR	signal-to-noise ratio

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
1 (PDF)	DEVCOM ARL FCDD RLD DCI TECH LIB
1 (PDF)	DEVCOM ARL FCDD RLS EA T GARNER D WIKNER
1 (PDF)	UNIV FLORIDA W DAVIS
1 (PDF)	DEVCOM C5ISR J DEROBA E BROWN
1 (PDF)	DEVCOM AVMC M HEIMBECK