Chaotic Direct Sequence Spread Spectrum Software Defined Radio System Model Using LabView

Leonard Novosel, Gordan Šišul
University of Zagreb, Faculty of Electrical Engineering and Computing,
Unska 3, 10 000 Zagreb, Croatia
leonard.novosel@fer.hr

Abstract—Paper presents a simple mathematical construction for generating chaotic sequences from piecewise linear tent map. A method for generating maximum length pseudo-noise sequences is explained. Direct sequence-spread spectrum system model is built in LabView for software defined radio USRP transceiver system. Performance of multiple chaotic sequences is compared against maximum length sequences on software defined radio transceiver chain and shown that chaotic sequences have equal, in some cases, even better performance.

Keywords—Chaotic sequence, DS-SS, Spread spectrum, m-sequence, SDR, LabView

I. INTRODUCTION

Applicability of chaotic signals in analog and digital communication systems has been researched during the past few decades [1–4]. First use of chaotic signals in communication systems was with the discovery of chaotic behaviour in Chua’s circuit [5]. An important milestone in chaotic communications was with work of Pecora and Carrol [6] that confirmed synchronization of chaotic carriers. Discovery of chaotic synchronization led to development of different chaotic modulation techniques [7–10]. The research in digital chaotic communications was specially prolific in field of spread spectrum communications where chaotic systems led to research in new methods for generating better spreading codes [11], [12].

Direct sequence-spread spectrum (DS-SS) systems use spreading sequences to spread the information signal. The main purpose of spreading a baseband signal is to hide the signal from unwanted interception. Spreading sequence encodes the information signal so that only a receiver with a valid code can decode the data. An additional benefit from spreading is in increased interference mitigation and improved performance in multipath propagation [13].

Direct sequence-spread spectrum (DS-SS) systems use spreading sequences to spread the information signal. The main purpose of spreading a baseband signal is to hide the signal from unwanted interception. Spreading sequence encodes the information signal so that only a receiver with a valid code can decode the data. An additional benefit from spreading is in increased interference mitigation and improved performance in multipath propagation [13].

Direct sequence-spread spectrum modulation can be explained through two modulation stages. First, the data sequence modulates a spreading, wideband code. The wideband code is used to spread the narrowband data stream into a wideband signal with low power-spectral density and noise-like properties. Wideband signal is, then, modulated to a carrier using phase-shift keying modulation [14]. Resulting wideband signal will have bandwidth that is almost equal to spreading code’s bandwidth.

By multiplying the data signal with the spreading signal, each information bit is spread to $N_c$ chips where $N_c$ is number of spreading sequence samples. Main characteristic of DS-SS signal is that the power-spectral density of the spread signal can lie below channel noise power-spectral density level that allow for stealthy communications and provide protection against interference or jamming [13].

Wideband spreading is usually done with pseudo-noise (PN) sequences that are deterministic periodic signals known to the transmitter and receiver, or the method of generation of such sequences is known to both. Because pseudo-noise sequences are periodic, there are only a limited number of sequences of given length. Also, there exists a finite number of uncorrelated pseudo-noise sequences that set the limit on number of users in multiple access schemes.

Chaotic sequences derived from linear chaotic maps have similar pseudo-noise characteristics. Previous work in comparing pseudo-noise and chaotic sequences was performed [15] that shows that chaotic sequences can be used as a replacement for pseudo-noise sequences. Chaotic signals can be used in spread spectrum systems due to their wide spectrum and random appearance and noise-like behaviour in time domain. With minor changes of chaotic system parameters a large number of uncorrelated sequences can be generated thus increasing the number of users in multiple access systems [15].

This paper focuses on comparing a class of pseudo-noise sequences called maximum length ($m$-sequences) sequences against chaotic sequences of equal length on a real-world spread-spectrum communication system built on software defined radio transceiver chain with signal processing part running on LabView.

This paper is organized as follows: Section II explains how pseudo-noise and chaotic sequences are generated. Section III gives the overview of spread spectrum system model built in LabView for software defined radio transceiver chain. Section IV discusses results obtain from software defined radio system. Section V concludes this paper.

II. SPREADING SEQUENCES FOR DIRECT SEQUENCE SPREAD SPECTRUM

Most spread spectrum systems use some form of pseudo-random sequence for spreading purposes. Mostly used sequences are pseudo-noise sequences that are periodic sequences with noise-like properties. In multiple access systems, special spreading sequences, that are constructed with desired properties, are used for spreading. All spreading sequences...
need to have good autocorrelation properties that are used for code synchronization and detection.

A. Generating pseudo-noise sequences

Spread spectrum communications systems require spreading sequences that have good random properties and can be generated using simple methods. Most pseudo-noise sequences can be generated using shift register systems [13].

Shift register is usually made of \( n \) flip-flops and a logic gate that form a feedback circuit [14]. State of the flip-flops is altered in accordance to a clocking signal where, on each clock pulse, the state of each flip-flop is shifted to its neighbour. Logic circuit, after modulo 2 operation, produces a symbol that forms a pseudo-random sequence [14]. Generated symbol is, then, fed back to the first flip-flop. It is obvious that sequences generated using shift register systems have periodic behaviour where the same sequence will be generated after a number of clock pulses. Using known generator architecture, security of spread-spectrum system is compromised where the attacker can generate all spreading sequences and detect those with best autocorrelation value. Maximal length sequences have the property that, for a \( n \)-stage shift register, the sequence has a repetition period \( p = 2^n - 1 \) [16]. More on generating pseudo-noise sequences can be found in [14] and [16]. Maximal length or \( m \)-sequences will be used as spreading codes in this paper and compared to sequences derived from chaotic signals.

B. Generating chaotic sequences

Chaotic oscillations come from nonlinear system elements that cause unwanted behaviour, sometimes leading to complete system failure. Those chaotic signals, or noise, have been identified as a component in all man-made or natural complex systems. Engineers have battled against noise by simplifying nonlinear systems and ignoring or filtering the noise. Research has found that these chaotic oscillations can be reproduced using relatively simple mathematical objects called chaotic maps and chaotic flows.

Sample based chaotic signals with noise-like properties [15] can be generated from chaotic maps that can be generally given as:

\[
X_{n+1} = F(X_n)
\]

where \( X_n \) is a state variable. Most chaotic system have a system or bifurcation parameter that alters the dynamics of the system to generate chaotic behaviour [15]. Changing the state variable leads to significant differences in generated chaotic signals that can. There exist a number of chaotic maps that can be used as chaotic sequence generators [17], but in this paper a tent map will be used to generate chaotic signals because of its simple model and easily observable behaviour.

Tent map is a logistic, piecewise linear, one-dimensional map on the interval \([0,1]\) exhibiting chaotic dynamics and given by [17]:

\[
X_{n+1} = \mu(1 - 2|x_n - \frac{1}{2}|)
\]

Using (2) different chaotic signals of different lengths are generated. A threshold function is applied to the chaotic signals to obtain a bipolar chaotic spreading sequences that are used to spread signals in DS-SS system model built in LabView. Discretization of chaotic signals leads to loss of information and randomness but only dichotomous values in spreading signals can be used in DS-SS systems.

By changing the bifurcation parameter \( \mu \) by a small amount, an entirely different chaotic signal can be generated. In this paper two chaotic signals were generated using \( \mu = 1.99 \) and \( \mu = 1.9899999 \). The bifurcation parameter (or any other parameter) can be known to transmitter and receiver in multiple access system so that they can locally generate chaotic spreading sequence [15]. Storing only one parameter and not the whole sequence can lead to increased security and less system resource consumption.

III. SOFTWARE DEFINED RADIO SPREAD-SPECTRUM SYSTEM MODEL IN LABVIEW

DS-SS communications system model is built and tested in LabView with different spreading codes on an Universal Software Radio Peripheral (USRP) transceiver chain. A receiver and a transmitter program are built separately for two USRP software defined radio devices. USRP’s are connected using a coaxial cable, on the receiver side a 30 dB attenuator is fixed so that the transmitted signal does not damage receiver side circuits.

The receiver is consists of a random number generator that generates data bits. Frame is constructed using padding head bits, Barker code of length 13, generated data bits and tail padding bits. Padding bits are necessary due to filter delay in LabVIEW blocks. Barker code is used for phase synchronization during phase shift keying demodulation as well for frame synchronization after bit recovery. Constructed frame is then BPSK modulated and spread using different codes. The generated spread signal is upsampled to 1 million samples per second and transmitted through software defined radio transmitter. Output port of the transmitter and input port of the receiver device are connected using a coaxial cable so the channel can, in some sense, be considered as an AWGN channel.

The receiver side software defined radio device receives the raw IQ data. IQ data is then despread using same spreading code as in the transmitter and a threshold energy detector is used to detect the start of valid signal. Before despread procedure can be performed, spreading code has to be synchronized. Code synchronization is performed using a correlation detector over the whole chip. Naturally, longer sequences will have better correlation properties, but due to longer code synchronization takes longer. After the synchronization is obtained (on maximum detected correlation value), the despread operation is performed on the whole chip length.

After despreading the signal is equalized using a feed forward equalizer and BPSK demodulated where phase ambiguity in demodulation process is resolved using Barker sequence in preamble. The start of the recovered frame is
then synchronized and aligned using included Barker code. Received data bits are extracted and compared against sent bits so that bit error rate can be calculated.

Because USRP platform can not measure power in absolute values but uses a relative amplitude value and uses a finite precision upsampled signal, the signal to noise ratio, or any other channel parameter, is hard to calculate with satisfying precision. Therefore the transmitter amplifier gain is used as measure of signal level at receiver end. Receiver gain is fixed to 10 dB and transmitter gain is alternated between 5 dB and 15 dB.

IV. RESULTS

Results were obtained until $10^6$ data bits were received and recovered.

Comparing performance of chaotic sequences and m-sequences from Fig. 1, 2 and 3 it can be seen that chaotic sequences generated from simple map function such as tent map, have the same performance, even outperform m-sequences in some cases, especially in noisier environments in high gain levels.

Observing results in Fig. 1, 2, 3 it may appear strange that raising transmitter gain, and therefore power, leads to increase in error rate. The cause of this strange behaviour is in USRP architecture where the driving amplifier is amplifying all signals, including noise. In time slots where the transmitter is not transmitting or the signal is attenuated below channel noise levels, the receiver still has some signal on the input. This problem has been somewhat mitigated using a energy threshold detector, but that threshold detector can not alter it’s threshold value dynamically with input signal. That is why the BER curve slopes up at higher gain levels where detector is triggered even by the amplified noise.

Comparing m-sequences of different lengths shows that they have the same performance in less noiser environment but longer sequences outperform short sequences in high noise environments at higher amplifier gains. Such behaviour is expected as longer sequences have better autocorrelation, and therefore, better synchronization properties.

Chaotic sequences perform differently depending on the used bifurcation value $\mu$, as was expected. Both shortest chaotic sequences of length 15 outperform the m-sequence, but longer sequences show different behaviour depending on bifurcation parameter and sequence length that shows performance dependence on used bifurcation parameter. Such dependence can be used to generate sequences that will have excellent performance in DS-SS or CDMA systems.

Where one chaotic sequence with parameter $\mu = 1.99$ has better performance, as seen on Fig. 3, the other chaotic sequence with $\mu = 1.9899999$ has the same or worse performance than m-sequences, especially in high noise environment. On the other hand, second chaotic sequence of length 31 has better performance than the first chaotic sequence, and, in most cases, than m-sequence of same length.

These diverse results arise from nonlinearities in used dynamical system and may be due to the method of generating chaotic sequences where more study is needed to generate chaotic sequences with specific properties. There yet does not exist a proven method to generate chaotic sequences with desired correlation properties, rather the sequences are generated and their properties are compared post generation [18].

V. CONCLUSION

Software defined radio systems gives a researcher a simple and reconfigurable tool for system modelling and parameter observation with an easy insight to system behaviour. Such reconfigurability is especially interesting in proving new system elements or parameters. The direct sequence-spread spectrum system model based on LabView and software defined radio enabled insight into behaviour of chaotic sequences in DS-SS system and shows that these sequences can be used as spreading sequences in spread spectrum systems. Additional testing is needed with different chaotic maps and sequence
generators to determine their characteristics and performance in real systems. In this paper, where only a few sequences were tested, it was shown that chaotic sequences behave differently with small changes to system parameters. Further research should be made in statistical analysis of chaotic signals and comparison to pseudo-noise sequences with varying system parameters. Generating chaotic signals with desired properties is also an interesting research topic, especially with implementation in multiple access systems where performance can be tested in real systems on software defined radio systems.

REFERENCES


