A NOTE ON FRAME SYNCHRONIZATION SEQUENCES

Thokozani Shongwe[1], Victor N. Papilaya[2]

[1]Department of Electrical and Electronic Engineering Science,
University of Johannesburg
P.O. Box 524, Auckland Park, 2006,
Johannesburg, South Africa
e-mail: tshongwe@uj.ac.za

[2]Electronic and Computer Engineering Faculty
Satya Wacana Christian University
Jl. Diponegoro 52-60, Salatiga 50711, Indonesia
e-mail: victor.papilaya@staff.uksw.edu

ABSTRACT

An introduction into binary sequences used for frame synchronization is given, together with criteria used in designing and/or finding optimal sequences that can be used as markers. Two approaches to the choice of optimal sequences for frame synchronization are presented, one approach focuses on the good properties of the marker and the other focuses on both marker and the data used in a frame. To illustrate the importance of "good" marker choice, simulations are presented showing the performance of "good" and "bad" marker choices.

Keywords: Frame synchronization, markers, bounded delay, synchronization codes

1. WHY SYNCHRONIZATION?

In Digital communications, synchronization is essential if reliable communications is to take place between a transmitter of information and a receiver of information. To explain why synchronization is necessary, we explain what synchronization is for digital communications. Consider a simplified digital communications system consisting of a transmitter, a channel and a receiver of information. The basic element of information is considered to be a bit, from which blocks (of bits) can be formed and called sequences/words or frames. For the receiver to "make sense" of the transmitted information, it must know when a bit starts and ends (bit period/duration). This bit period should be agreed upon between the receiver and transmitter, that is, clocks at the receiver and transmitter must be synchronized. Bit synchronization is the first level of synchronization; the next level is to synchronization a group of bits forming a frame, which is called frame synchronization. In frame synchronization the receiver is interested in knowing the beginning and end of every frame sent to it. Obviously, frame synchronization can only work if bit synchronization has already been achieved. There are two well known methods of synchronization for blocks of data bits namely, marker based frame synchronization and bounded synchronization delay (BSD) codes. Discussion in this paper is limited to the design markers used in frame synchronization, only a brief discussion will be given on BSD in Section IV.

2. FRAME SYNCHRONIZATION USING MARKERS

2.1. Frame Synchronization Algorithm

Markers are sometimes called synchronization sequences or Synchronization markers, for brevity, throughout this paper we shall use the word marker(s). The most common approach in frame synchronization using markers is to embed the marker in the transmitted data and at the receiver, a search for the marker is performed in the frame using a sliding window method and Hamming distance comparison. The sliding window of the same length (number of symbols) as the marker, is moved symbol by symbol over the received data and at each window period the data is compared for Hamming distance or correlated with the known marker.

Let the following parameters be defined to enhance the description of a synchronization algorithm.

- S: synchronization word or synchronization sequence or marker, defined as $\{S_1S_2 \ldots S_N\}$.
- N: synchronization sequence length (number of symbols making up a synchronization sequence).
- H: error tolerance threshold of the synchronization algorithm (that is, any N-tuple of received symbols that differ to the marker in at most H positions is declared the marker.)
- L: number of data symbols in a synchronized frame.
- P_e: symbol error rate.

A frame is then made up of L data symbols and N marker symbols as shown in Fig. 1, where three frames are shown to give an idea of a continuous transmission of data with periodically embedded markers.

$S_1\ S_2 \ldots S_N$	$D_1\ D_2 \ldots D_L$	$S_1\ S_2 \ldots S_N$	$D_{L+1}\ D_{L+2} \ldots D_{2L}$	$S_1\ S_2 \ldots S_N$	$D_{2L+1}\ D_{2L+2} \ldots D_{3L}$

Fig. 1. Framed sequences of received symbols with synchronization sequences

There can be various algorithms for the marker search using the sliding window and correlation, but in general the algorithm should look something like this.

1. From the initial received symbol, pick symbols of length equal to the known marker, an N-tuple of symbols called a window.
2. Compare the N-tuple received symbols with the known marker for agreements according to a set error tolerance threshold, H.
   a. If the N-tuple satisfy the error tolerance threshold, the algorithm declares a marker found and a search for the next marker begins.
   b. If the N-tuple does not satisfy the error tolerance threshold, the algorithm continues searching for the marker.

Obviously, with such an algorithm several things can happen, but there are two important events that can determine the success rate of such an algorithm. Either the algorithm declares a marker found in the wrong place in the frame, which we call *false acquisition* or declares a marker found in the right place in the frame, which we call *true acquisition*.

2.2. Aperiodic Autocorrelation Based Markers

Since the pioneering work done by Barker [2] in 1953 on frame synchronization, most sequences that were used as markers have been designed based on aperiodic autocorrelation

and some of these sequences are, Barker [2], Turyn [3], Willard [4] and Maury and Styles [5] sequences. These sequences have good aperiodic autocorrelation functions (ACFs) which make them “optimal” markers, and what that basically means is that the sequence only produces an aperiodic autocorrelation function that has a high *main lobe* and minimal *side lobes* when correlated with zero-shifted version of itself. Using a Barker sequence, the following example illustrates a marker with a good autocorrelation function.

Example 1: Fig. 2 shows the graphical representation of an aperiodic autocorrelation function of a barker sequence of length N = 7, {1110010}. The results in Fig. 2 were found by performing correlation between the barker sequence and the shifted versions of itself in the following manner: Let = x denote the x-shifted version of the sequence, therefore, the aperiodic autocorrelation of {1110010} is,

$$\begin{array}{lll}
S: & 1\,1\,1\,0\,0\,1\,0 & \\
\tau = 0: & 1\,1\,1\,0\,0\,1\,0 & = 7 \\
\tau = 1: & \quad 1\,1\,1\,0\,0\,1\,0 & = 0 \\
\tau = 2: & \quad\quad 1\,1\,1\,0\,0\,1\,0 & = -1 \\
\tau = 3: & \quad\quad\quad 1\,1\,1\,0\,0\,1\,0 & = 0 \\
\tau = 4: & \quad\quad\quad\quad 1\,1\,1\,0\,0\,1\,0 & = -1 \\
\tau = 5: & \quad\quad\quad\quad\quad 1\,1\,1\,0\,0\,1\,0 & = 0 \\
\tau = 6: & \quad\quad\quad\quad\quad\quad 1\,1\,1\,0\,0\,1\,0 & = -1.
\end{array}$$

The values of the ACF at each τ = x result into what is called side lobes for x ≠ 0, and main lobe at x = 0.

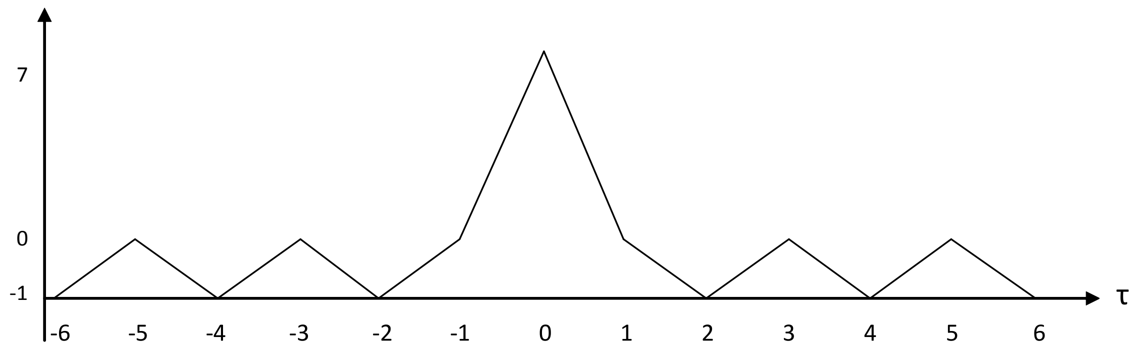

Fig. 2. Graphical representation of an aperiodic auto correlation function of a length seven binary Barker sequence, {1110010}.

The aperiodic autocorrelation function of a binary sequence, C_f , can be presented mathematically as C_f = $\sum_{t=1}^{N-\tau}(-1)^{S_t+S_{t+\tau}},\ S_t \in \{0,1\}$.

For Barker sequences, the aperiodic autocorrelation function satisfies the following conditions, $|Cf_{\tau}| \leq 1,\ 1 < \tau < N$. All sequences therefore described by an autocorrelation function where the side lobes do not exceed a magnitude of one are called Barker sequences, even though they are not the original Barker sequences. The following is a list of known Barker sequence for various lengths up to N = 13. N = 2 : {11}, N = 3 : {110}, N = 4 : {1110}, N = 5 : {11101}, N = 7 : {1110010}, N = 11 : {11100010010} and N = 13 : {1111100110101}.

There is a wide range of other types of sequences that were designed following the breakthrough of Barker sequences, and these sequences were based on the autocorrelation function criteria. The design goal of these sequences is to minimize the side lobes of the ACF. Binary sequences with low values of the ACF (except at τ = 0) are not only suitable for frame synchronization, but for a number of applications as well, some of which are radar pulse-compression, spread spectrum communications and stream ciphers.

While considering the ACF as the criterion for designing/searching for optimal markers is a good guideline, there are other ways of looking at the problem of marker design. One such criterion was presented by Scholtz in 1980 [1], which is consideration of the probability of false acquisition on data, P_{FAD} and probability of true acquisition on the marker, P_{TAM}. We shall not elaborate further on this design criteria in this paper.

3. GOOD AND BAD MARKER CHOICES

In this section, simulation results are presented to show examples of good and bad marker choices. It is further explained why one particular sequence is preferred over another as a marker. A binary symmetric channel (BSC) was used for the simulations.

Figs. 3 and 4 show similar graphs of comparison of two examples of bad markers and two well known optimal markers. The two well known optimal markers used in the simulation are (a) Barker sequence of length seven, {1110010}, and (b) a sequence common to both Turyn sequences, and Maury and Styles sequence, {1011000}. It should be noted that the Turyn, and Maury and Styles sequence in (b), {1011000}, satisfy the Barker condition in the ACF ($|C_f| \leq 1$, $1 < \quad < N$), hence the equal performance to the Barker sequence in (a). In both Figs. 3 and 4, a *same symbol sequence* and *periodic sequence* are used as the two examples of bad markers. In Fig. 3 sequences {0000000} and {1010101} are used, and In Fig. 4 sequences {1111111} and {1100110} are used.

The sequences, {0000000}, {1010101}, {1111111} and {1100110} all perform worse than the Barker ({1110010}), and Turyn, Maury and Styles sequence ({1011000}) because they have larger sidelobes in their ACFs. The same symbol sequences are the worst because they have ACFs with the highest sidelobes compared to all the sequences in question. The procedure introduced in Example 1 can be used to verify this conclusion on the performance of the sequences based on ACF.

In the simulations, a simple correlation rule was used to locate the marker in the frame. In the correlation rule, the entire frame is searched for the marker and the N-tuple with lowest Hamming distance with the marker is declared a marker. When there are more than one N-tuples with lowest Hamming distance the algorithm declares synchronization failure as well as when the Marker is declared in the wrong position.

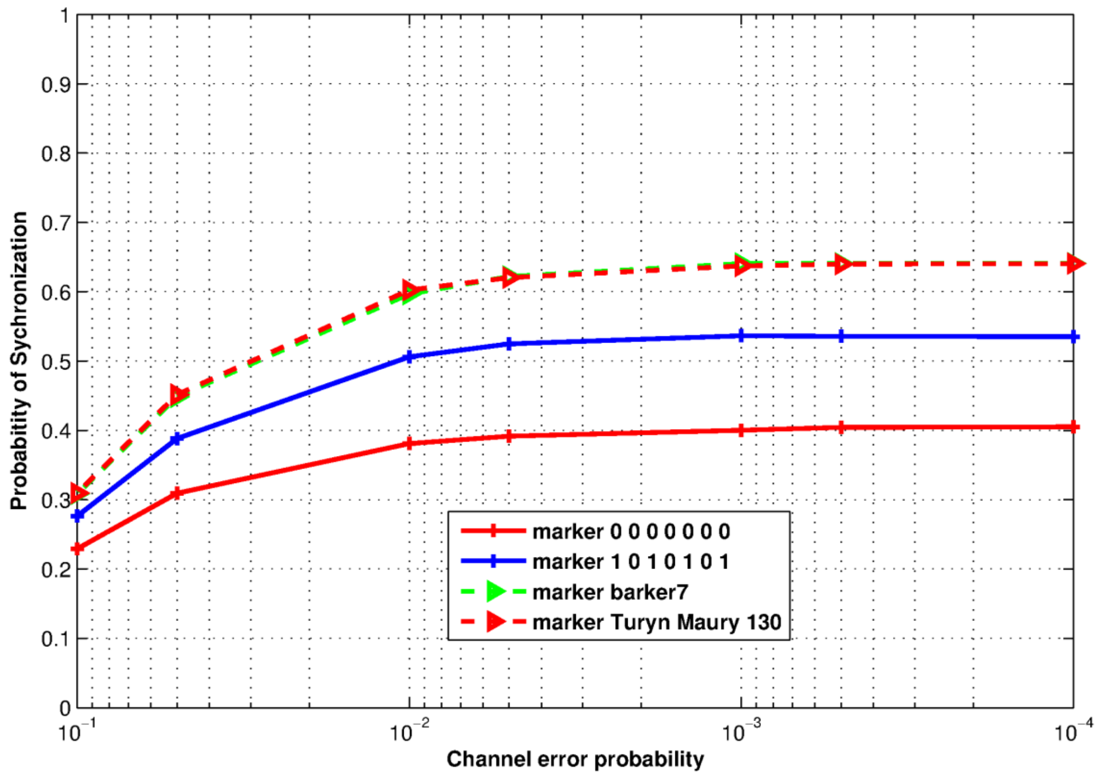

Fig. 3. Comparison of four possible markers for probability of synchronization in a BSC for various channel error probabilities.

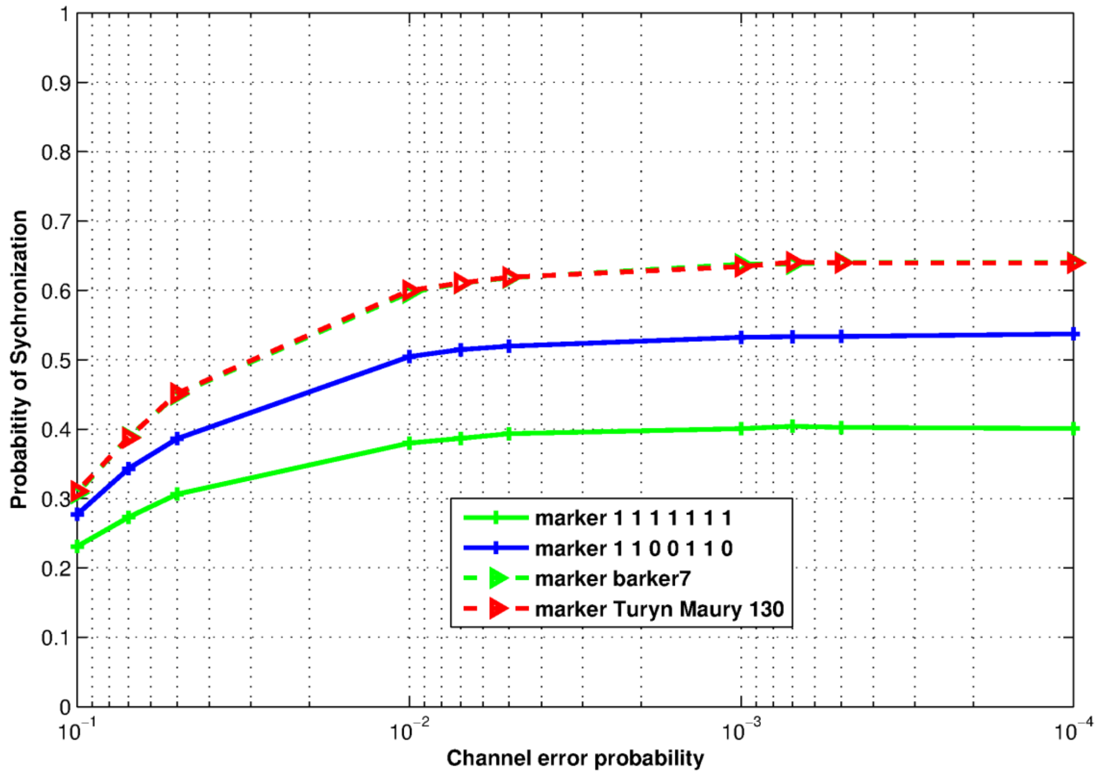

Fig. 4. Comparison of four possible markers for probability of synchronization in a BSC for various channel error probabilities.

4. BOUNDED SYNCHRONIZATION DELAY BLOCK CODES

Apart from using frames composed of data and markers, there are other ways of achieving synchronization in digital transmission. In this section, a class of codes called *Bounded synchronization delay* (BSD) is discussed as another option to the frame synchronization. BSD codes are synchronizable codes with *uniquely decodable* codewords, which when their codewords are sent through a noiseless channel, a receiver can be able to tell the boundaries of the codewords after observing d_l symbols. The parameter d_l is the synchronization delay of the codes and is bounded by some value in relation to the codewords' length, hence the term BSD. BSD codes can be variable length (examples here are source codes) or fixed length. Variable length BSD codes are outside the scope of this paper, so the focus is on BSD codes of fixed length, which are referred to as block codes. BSD block codes can be divided into three main classes, *comma-free codes*, *prefix codes* and *comma codes*.

4.1. Comma-Free Codes

This class of BSD block codes has been extensively studied. It was first introduced by Golomb et. al in 1958 [7], where they formalised it as follows. Given a block code, B and codewords X = $\{x_1x_2 \ldots x_n\}$ and Y = $\{y_1y_2 \ldots y_n\}$ of length n, where X, Y Є B. B is a comma free code if, and only if, for any n-tuple, Z resulting from the overlap of X and Y , Z = $\{x_1 \ldots x_ny_1 \ldots y_{i-1}\}$, Z is not element of B for $2 \leq i \leq n$.

Eastman [8], was the first to present a construction that achieves the maximum cardinality comma-free codes, for n odd.

4.2. Prefix Codes

Prefix codes were first advanced by Gilbert [10] in his article on synchronization of binary messages. These codes are constructed by appending a N-tuple sequence (called a prefix) to every n-symbol sequence such that the (N + n)-symbol sequence is a uniquely decodable codeword of codebook. It should be noted that N-tuple sequence is the same for every n-symbol sequence and it is appended to indicate the start of each codeword, hence allowing synchronization of the codewords. For synchronization to be possible all the time, in the absence of errors, the N-tuple prefix should not be part of any codeword or should not be formed by an overlap between a codeword and the prefix itself.

4.3. Comma Codes

In comma codes, codewords of a particular length say, n, are chosen from a codebook to be transmitted and after k codewords, a special N-tuple sequence is inserted in the transmission as a comma. These codes are more attractive for implementation because the frequency with which the comma is inserted in the transmission can be varied according to the channel properties. The constraints in the choice or design of a comma in the comma codes are, the comma should not be any of the n-tuple codewords, an overlap between any two or more n-tuple codewords or an overlap between the comma and any n-tuple codeword should not produce the comma. Related work in this area can be found in [11] and [12].

5. CONCLUSION

The question of why we need synchronization and how it can be achieved in digital communications was answered using a known technique of frame synchronization, with the assumption that symbol synchronization was already established. Then a study for markers for frame synchronization was presented in this article, where aperiodic correlation function properties of the marker were discussed. It was also shown that another approach to the marker formulation problem is to study the data not containing the marker and in this approach, the probabilities of false and true acquisition needed to be computed. In all the cases, the data was considered to be composed of Q-ary symbols randomly selected from an alphabet of size Q with equal probability. Another more interesting scenario, addressed in the Subsection IV-C on the subject of comma codes, is when the data is chosen as codewords from a codebook and then appending a marker at specific fixed intervals.

REFERENCES

[1] R. A. Scholtz, “Frame synchronization techniques,” IEEE Transactions on Communications, vol. 28, no. 8, pp. 1782–1789, Aug. 1980.

[2] R. H. Barker,“Group synchronizing of binary digital systems,” in Communication Theory, W. Jackson. Ed. New York: Academic-Butterworth, 1953.

[3] R. Turyn, “Sequences with small correlation,” Error Correcting Codes, pp. 195–228, H. B. Mann, Ed. New York: Wiley, 1968.

[4] M. W. Willard, “Optimum code patterns for PCM synchronization,”in Proc. 1962 National Telemetering Conference, Washington, DC. May 1962, paper 5-5.

[5] J. L. Maury and F.J. Styles, “Development of optimum frame synchronization codes for Goddard Space Flight Center PCM Telemetry Standards,” in Proc. 1965 National Telemetering Conference, Los Angeles, CA, Jun. 1964. paper 3-1.

[6] J. J. Stiffler, Theory of Synchronous Communications. New Jersey: Prentic-Hall, Inc., 1971.

[7] S. W. Golomb, B. Gordon, and L. R.Welch,“Comma-free codes,” Canadian Journal of Mathematics, vol. 10, no. 2, pp. 202–209, 1958.

[8] W. L. Eastman, “On the construction of comma-free codes” IEEE Transactions on Information Theory, vol. 11, no. 2, pp. 263–267, Apr. 1965.

[9] B. H. Jiggs, “Recent results in comma-free codes,” Canadian Journal of Mathematics, vol. 15, pp. 178–187, 1963.

[10] E. N. Gilbert, “Synchronization of binary messages,” IRE Transactions on Information Theory, IT-6, pp. 470, 1960.

[11] W. B. Kendall, “Optimum synchronizing words for fixed word-length code dictionaries”, Space Program Summary, 4, Jet Propulsion Lab., calif. Inst. Tech., Pasadena, Calif. pp. 37, 1964.

[12] J. J. Stiffler, “Instantaneously synchronizable block code dictionaries”, Space Program Summary 37–32, 4, Jet Propulsion Lab., calif. Inst. Tech., Pasadena, Calif. pp. 268, 1965