Performance analysis and simulations of 32-ary cyclic code-shift keying

Kao, Chi-Han; Robertson, Clark; Kragh, Frank; Lin, Kyle Y.

Wiley

http://hdl.handle.net/10945/47549

Downloaded from NPS Archive: Calhoun
Performance analysis and simulations of 32-ary cyclic code-shift keying

Chi-Han Kao$^1$, Clark Robertson$^2$, Frank Kragh$^{2,*}$,† and Kyle Lin$^3$

$^1$Electrical Engineering Department, Chinese Naval Academy, Kaohsiung, 81300 Taiwan
$^2$Electrical and Computer Engineering Department, Naval Postgraduate School, Monterey, CA 93943-5121, U.S.A.
$^3$Operations Research Department, Naval Postgraduate School, Monterey, CA 93943-5121, U.S.A.

SUMMARY
Cyclic code-shift keying (CCSK) is the baseband 32-ary symbol modulation scheme used by the Joint Tactical Information Distribution System (JTIDS), the communication terminal for Link-16. CCSK is not orthogonal and an analytic expression for the probability of symbol error for CCSK has thus far been elusive. In this paper, an analytic upper bound on the probability of symbol error of CCSK is derived for the 32-chip CCSK starting sequence chosen for JTIDS. The analytically obtained probability of symbol error is compared with two different Monte Carlo simulations for additive white Gaussian noise. The results of both simulations match the analytic results very well and show that the analytic method yields a tight upper bound. A new 32-chip CCSK starting sequence which has a smaller maximum off-peak cross-correlation value than the current JTIDS starting sequence is proposed and evaluated both analytically and by simulation. The results obtained for the new CCSK starting sequence compare favorably with the CCSK starting sequence chosen for JTIDS. Published in 2010 by John Wiley & Sons, Ltd.

Received 22 May 2009; Revised 2 December 2009; Accepted 19 February 2010

KEY WORDS: cyclic code-shift keying (CCSK); joint tactical information distribution system (JTIDS); link-16

1. INTRODUCTION
Cyclic code-shift keying (CCSK) is the baseband 32-ary symbol modulation scheme used in Link-16, a widely used tactical data link (TDL) [1]. The militaries of North American Treaty Organization (NATO) member nations are the primary users of Link-16, but militaries of other NATO-friendly nations have also purchased this capability [2]. The communication terminals for Link-16 are the Joint Tactical Information Distribution System (JTIDS) and the Multifunction Information Distribution System (MIDS), with the latest terminal being the MIDS-Joint Tactical Radio System (JTRS) [3]. TDLs are terrestrial digital radio communication systems designed for military use, including in combat conditions [4]. Therefore, the reliability of the communications is a critical performance criterion. This paper analyzes a fundamental reliability criterion, the symbol error ratio (SER). The authors furthermore suggest a potential modification to Link-16 that would improve the performance, as shown herein. This potential upgrade could be important to Link-16 users since the software-defined nature of MIDS-JTRS [3] offers the potential for relatively easy software upgrades.

$^*$Correspondence to: Frank Kragh, Electrical and Computer Engineering Department, Naval Postgraduate School, Monterey, CA 93943-5121, U.S.A.
$^1$E-mail: frank.kragh@ieee.org
$^2$This article is a U.S. Government work and is in the public domain in the U.S.A.

Published in 2010 by John Wiley & Sons, Ltd.
As CCSK is non-orthogonal, an analytic expression for the probability of symbol error for CCSK has thus far been elusive. Previously, the evaluation was done by simulation [5–7]. The only exception is [8]. Several CCSK cross-correlation properties are formulated in [8], and an analytic expression for the performance of the JTIDS system in additive white Gaussian noise (AWGN) was first given. The results presented in [8], however, are based on the overly optimistic assumption that the cross-correlation values of the CCSK symbols are independent. In fact, as is shown in this paper, these cross-correlation values are not independent. Based on this new finding and the CCSK cross-correlation properties from [8], we use a different approach to derive an analytic upper bound on the probability of symbol error for CCSK with the 32-chip CCSK sequence chosen for JTIDS. Consistent with [9], a two-step maximum-likelihood (ML) detection process is considered rather than a one-step ML chip-sequence detection process since the former represents a more practical assumption for a JTIDS-type signal. The two-step ML detection process includes ML chip detection, which detects each chip separately, followed by ML sequence detection that chooses the 32-chip sequence closest to the ML-chip detector’s output.

The probability of symbol error obtained with the analytic method is compared with that obtained via two different Monte Carlo simulations. One simulation estimates the average probability of symbol error for CCSK using the 32-chip CCSK sequence chosen for JTIDS, whereas the other estimates the conditional probabilities of symbol error for CCSK using the same sequence to obtain the average probability of symbol error for CCSK in AWGN. In addition to the 32-chip CCSK sequence chosen for JTIDS, a new 32-chip CCSK sequence with a smaller maximum off-peak cross-correlation value is presented and evaluated via both analysis and simulation.

The remainder of this paper is organized as follows. The system model of a JTIDS-type transceiver is introduced in Section 2. The fundamentals of CCSK including modulation, demodulation, cross-correlation properties, and conditional probabilities of symbol error for CCSK are discussed in Section 3. An analytic upper bound on the probability of symbol error for the 32-chip CCSK sequence chosen for JTIDS is derived and evaluated in Section 4. Two different Monte Carlo simulations and their results are presented in Section 5. A new 32-chip CCSK sequence is proposed in Section 6. Finally, the important results and findings are summarized in Section 7.

2. SYSTEM MODEL DESCRIPTION

2.1. JTIDS-type transceiver

A schematic system model of a JTIDS-type transceiver is shown in Figure 1. The top section is the transmitter, and the bottom section is the receiver. As is indicated, the transmitter consists of a Reed-Solomon (RS) encoder, a symbol interleaver, a CCSK 32-ary baseband symbol modulator, a data chip scrambler, a minimum shift keying (MSK) chip modulator, a frequency-hopping circuit, and a transmitting antenna [9].

2.2. Transmission process

When the Link-16 message bit stream arrives at the JTIDS transmitter, it is first mapped onto 5-bit symbols. The seven symbol message header is encoded using a (16, 7) shortened and/or punctured RS code. The 15 symbols of the message data are encoded with a (31, 15) RS code. After encoding, the data and header symbols are interleaved for the first layer of transmission security. Next, these 5-bit interleaved symbols are modulated with 32-ary CCSK, where each 5-bit symbol is represented by one of the cyclic-shifts of a 32-chip starting sequence. To obtain the second layer of transmission security, each 32-chip CCSK sequence is scrambled with a 32-chip pseudo-noise (PN) sequence. The resulting 32-chip sequence is MSK modulated to generate an analog pulse. Each pulse is then up-converted to one of the 51 possible carrier frequencies, which contributes a third layer of transmission security. After up-conversion, the signal is amplified, filtered, and transmitted over the channel [10].
2.3. Reception process

The reception process is the reverse of the transmission process. After frequency de-hopping, MSK chip demodulation, and descrambling, each 5-bit coded symbol is recovered by a 32-ary CCSK symbol demodulator. After de-interleaving, the coded message symbols are decoded by a (31, 15) RS decoder. If the decoding is successful, the data symbols are converted into a bit stream which is sent to the upper layer [10].

3. CYCLIC CODE-SHIFT KEYING

3.1. CCSK symbol modulation

In a 32-ary CCSK symbol modulator, each 5-bit symbol is represented by one of the cyclic-shifts of a 32-chip CCSK starting sequence. The 32-chip CCSK starting sequence \( b_0 \) chosen for JTIDS is shown in Table I. As can be seen, 32 CCSK sequences \( b_i \) are derived by cyclically shifting \( b_0 \) to the left \( i \) times where \( i \in \{0, 1, 2, \ldots, 31\} \) to obtain a unique sequence for each 5-bit symbol.

3.2. CCSK symbol de-modulation

As mentioned, this paper considers a two-step ML detection process: ML chip detection and ML sequence detection. It is the second step that is discussed in this subsection. The ML sequence detector is equivalent to a minimum distance detector which, in this case, is equivalent to a maximum correlation detector because

\[
d^2(s_i, s_j) = \langle s_i - s_j, s_i - s_j \rangle = 64 - 2\langle s_i, s_j \rangle
\]

where \( s_i \) is the antipodal equivalent of \( b_i \), i.e. \( s_i, k = 2b_{i,k} - 1 \) for \( k = 0, 1, \ldots, 31 \); \( d(s_i, s_j) \) is the Euclidean distance between \( s_i \) and \( s_j \); and \( \langle s_i, s_j \rangle = \sum_{k=0}^{31} s_{i,k} s_{j,k} \) is the correlation between symbol
Table I. 32-chip CCSK sequences chosen for JTIDS (from [9]).

<table>
<thead>
<tr>
<th>5-bit symbol</th>
<th>32-chip CCSK sequences chosen for JTIDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>b_0 = 01111100111010010001010111101100</td>
</tr>
<tr>
<td>00001</td>
<td>b_1 = 01111100111010010001010111101100</td>
</tr>
<tr>
<td>00010</td>
<td>b_2 = 111100111010010001010111101001</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td>11111</td>
<td>b_{31} = 00111110011101000100110111011101100</td>
</tr>
</tbody>
</table>

Using (5), (6), and (7) we find that

\[ P_s = \sum_{j=0}^{31} \Pr(\text{symbol error}|N = j) \Pr(N = j) \]  

(3)

where \( N \) is the number of the symbol’s chips that are received in error during the first step of de-modulation. \( N \) is a binomial random variable with probability mass function

\[ \Pr(N = j) = \binom{32}{j} P_c^j (1 - P_c)^{32-j} \]  

(4)

where each chip is received in error with probability \( P_c \) and the binomial coefficient \( \binom{n}{k} \) is the coefficient of the term \( x^k \) in \((1+x)^n\).

The 31 symbols \( s_i \) for \( i \in \{1, 2, 3, \ldots, 31\} \) can be grouped into three sets according to their correlation with symbol \( s_0 \). As \( \langle s_0, s_7 \rangle = \langle s_0, s_{16} \rangle = \langle s_0, s_{25} \rangle = 4 \), the first set is defined as \( S_{\text{close}} = \{s_7, s_{16}, s_{25}\} \), and

\[ \Pr(s_7 \text{ chosen}|N = j) = \Pr(s_{16} \text{ chosen}|N = j) = \Pr(s_{25} \text{ chosen}|N = j) \]  

(5)

We define the second group as \( S_{\text{mid}} \equiv \{s_i|i \in \{1, 2, 4, 5, 8, 11, 12, 13, 14, 18, 19, 20, 21, 24, 27, 28, 30, 31\}\} \) since \( \langle s_0, s_i \rangle = 0 \) for all symbols in \( S_{\text{mid}} \). We note that

\[ \Pr(s_i \text{ chosen}|N = j) = \Pr(s_1 \text{ chosen}|N = j) \quad \forall s_i \in S_{\text{mid}} \]  

(6)

We define the third and final group as \( S_{\text{far}} \equiv \{s_i|i \in \{3, 6, 9, 10, 15, 17, 22, 23, 26, 29\}\} \) since \( \langle s_0, s_i \rangle = -4 \) for all symbols in \( S_{\text{far}} \). We note that

\[ \Pr(s_i \text{ chosen}|N = j) = \Pr(s_3 \text{ chosen}|N = j) \quad \forall s_i \in S_{\text{far}} \]  

(7)

Using (5), (6), and (7) we find that

\[ \Pr(\text{symbol error}|N = j) = \sum_{i=1}^{31} \Pr(s_i \text{ chosen}|N = j) \]

\[ = 3 \Pr(s_7 \text{ chosen}|N = j) \]

\[ + 18 \Pr(s_1 \text{ chosen}|N = j) \]

\[ + 10 \Pr(s_3 \text{ chosen}|N = j) \]  

(8)
Symbol errors will occur when the received chips \( r \) correlate at least as good with other symbols as they do with the sent \( s_0 \). Therefore
\[
\Pr(s_i \text{ chosen} | N = j) = \Pr(s_i \text{ chosen} | \langle r, s_i \rangle < \langle r, s_0 \rangle, N = j) \Pr(\langle r, s_i \rangle < \langle r, s_0 \rangle | N = j) \\
+ \Pr(s_i \text{ chosen} | \langle r, s_i \rangle = \langle r, s_0 \rangle, N = j) \Pr(\langle r, s_i \rangle = \langle r, s_0 \rangle | N = j) \\
+ \Pr(s_i \text{ chosen} | \langle r, s_i \rangle > \langle r, s_0 \rangle, N = j) \Pr(\langle r, s_i \rangle > \langle r, s_0 \rangle | N = j)
\]
\[
\leq 0 + 0.5 \Pr(\langle r, s_i \rangle = \langle r, s_0 \rangle | N = j) + 1 \Pr(\langle r, s_i \rangle > \langle r, s_0 \rangle | N = j)
\]
(9)
where the factor 0.5 is included because ties are assumed to be decided randomly, and the first inequality is included because another symbol other than \( s_0 \) or \( s_i \) may correlate as well or better with \( r \).

Assuming \( s_0 \) is sent and it is received with \( N = j \) chip errors, then
\[
\langle r, s_0 \rangle = 32 - 2j
\]
(10)
If \( Q_i \) of the \( j \) chip errors occur in chip positions where the chip in \( s_0 \) equals the chip in \( s_i \), then \( (N - Q_i) \) chip errors occur in chip positions where the chip in \( s_0 \) does not equal the chip in \( s_i \) and
\[
\langle r, s_i \rangle = \langle s_0, s_i \rangle - 2Q_i + 2(N - Q_i)
\]
\[
= \langle s_0, s_i \rangle - 4Q_i + 2N
\]
(11)
From (10) and (11) we can conclude
\[
\Pr(\langle r, s_i \rangle = \langle r, s_0 \rangle | N = j) = \Pr[Q_i = L(j, i) | N = j]
\]
(12)
where we define
\[
L(j, i) \equiv j - 8 + 0.25 \langle s_0, s_i \rangle
\]
(13)
Similarly,
\[
\Pr(\langle r, s_i \rangle > \langle r, s_0 \rangle | N = j) = \sum_{k=0}^{L(j, i) - 1} \Pr(Q_i = k | N = j)
\]
(14)
Combining (9), (12), and (14), we have
\[
\Pr(s_i \text{ chosen} | N = j) \leq 0.5 \Pr[Q_i = L(j, i) | N = j] \\
+ \sum_{k=0}^{L(j, i) - 1} \Pr(Q_i = k | N = j)
\]
(15)
Furthermore, \( Q_i \) is a hypergeometric distributed random variable [11] with probability mass function
\[
\Pr(Q_i = q | N = j) = \begin{cases} 
\binom{n_i}{q} \binom{32 - n_i}{j - q} & \text{if } 0 \leq q \leq j \\
\binom{32}{j} & \text{otherwise}
\end{cases}
\]
(16)
Table II. Conditional probabilities of channel symbol error: the CCSK sequence chosen for JTIDS versus the new CCSK sequence.

<table>
<thead>
<tr>
<th>j</th>
<th>$\zeta_{UB,j}$</th>
<th>$\zeta_{SIM,j}$</th>
<th>$\zeta_{UB,j}^*$</th>
<th>$\zeta_{SIM,j}^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0.0015</td>
<td>0.0015</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0.0207</td>
<td>0.0194</td>
<td>0.0147</td>
<td>0.0143</td>
</tr>
<tr>
<td>9</td>
<td>0.1166</td>
<td>0.1126</td>
<td>0.1040</td>
<td>0.1025</td>
</tr>
<tr>
<td>10</td>
<td>0.4187</td>
<td>0.3669</td>
<td>0.4023</td>
<td>0.3550</td>
</tr>
<tr>
<td>11</td>
<td>1.0</td>
<td>0.7093</td>
<td>1</td>
<td>0.7140</td>
</tr>
<tr>
<td>12</td>
<td>1.0</td>
<td>0.9351</td>
<td>1</td>
<td>0.9367</td>
</tr>
<tr>
<td>13</td>
<td>1.0</td>
<td>0.9953</td>
<td>1</td>
<td>0.9956</td>
</tr>
<tr>
<td>14</td>
<td>1.0</td>
<td>1.0</td>
<td>1</td>
<td>0.9999</td>
</tr>
<tr>
<td>15</td>
<td>1.0</td>
<td>1.0</td>
<td>1</td>
<td>1.0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>32</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

where $s_0$ and $s_i$ are equal in $n_i$ chip positions and

$$n_i = 0.5(s_0, s_i) + 16 = \begin{cases} 32 & \text{if } i = 0 \\ 18 & \text{if } s_i \in S_{\text{close}} \\ 16 & \text{if } s_i \in S_{\text{mid}} \\ 14 & \text{if } s_i \in S_{\text{far}} \end{cases}$$

Combining (8), (13), (15), (16), and (17), we calculate $\zeta_{UB,j}$, a tight upper bound on $\Pr(\text{symbol error} | N = j)$,

$$\Pr(\text{symbol error} | N = j) \leq \zeta_{UB,j} = 3 \left\{ 0.5\Pr(Q_7 = L(j, 7)|N = j) + \sum_{k=0}^{L(j, 7)-1} \Pr(Q_7 = k|N = j) \right\}$$

$$+ 18 \left\{ 0.5\Pr(Q_1 = L(j, 1)|N = j) + \sum_{k=0}^{L(j, 1)-1} \Pr(Q_1 = k|N = j) \right\}$$

$$+ 10 \left\{ 0.5\Pr(Q_3 = L(j, 3)|N = j) + \sum_{k=0}^{L(j, 3)-1} \Pr(Q_3 = k|N = j) \right\}$$

(18)

The numerical results are shown in the second column of Table II. Note that when $N \geq 11$, the upper bound produced by this method exceeds one, in which case the upper bound is given as one.

4. PROBABILITY OF SYMBOL ERROR FOR CCSK

Now, combining Equations (3), (4), and (18), we obtain an upper bound on the probability of symbol error for CCSK as

$$P_S < \sum_{j=7}^{32} \zeta_{UB,j} \left( \begin{array}{c} 32 \\ j \end{array} \right) P_c^j (1-P_c)^{32-j}.$$  

(19)
where $P_c$ is the probability of chip error at the output of the MSK chip demodulator. MSK can be considered as a special case of offset quadrature phase-shift keying (OQPSK) with sinusoidal pulse shaping. When a coherent matched filter or correlator is used to recover the chips, MSK has the same performance as BPSK, QPSK, and OQPSK [12]:

$$P_c = Q\left(\frac{\sqrt{2E_c}}{N_0}\right)$$  \hspace{1cm} (20)

As $E_s = 5E_b = 32E_c$, we can rewrite (20) as

$$P_c = Q\left(\frac{\sqrt{10E_b}}{32N_0}\right)$$  \hspace{1cm} (21)

Note that the actual JTIDS waveform is received non-coherently at the chip level, but in this paper the performance of a JTIDS-type waveform with coherent detection is evaluated to ascertain the performance possible if coherent chip demodulation were practical. The analysis presented in this paper can easily be modified to evaluate the performance with non-coherent chip demodulation.

Substituting (21) into (19), we obtain an analytic upper bound on the probability of channel symbol error for the 32-chip CCSK sequence chosen for JTIDS in AWGN. As this is an upper bound, it provides a guarantee that a properly designed and built Link-16 receiver should perform this good or better in AWGN. The results are shown in Figure 2 along with the probability of channel symbol error for 32-ary orthogonal signaling and that from [8], which assumes that CCSK cross-correlation values are conditionally independent. As expected, the performance of 32-chip CCSK is inferior to that of 32-ary orthogonal signaling by about 2 dB at $P_s = 10^{-5}$; however, the advantage of using CCSK is that only one detector branch is required to recover the original symbol instead of thirty-two detector branches. In addition, the results also show that, when $E_b/N_0$ is small, it is overly optimistic to treat CCSK cross-correlation values as if they were conditionally independent.

5. SIMULATIONS

In addition to the analytic upper bound derived in the last section, two different Monte Carlo simulations and their results are presented in this section. Simulations are important when dealing
with a bound because they allow us to quantify the tightness of the bound. Like the above analysis, the simulations do not consider the effects of RS coding, interleaving, scrambling, and frequency hopping. The first simulation, a Monte Carlo simulation with stratified sampling [13], is implemented to estimate the conditional probabilities of symbol error of CCSK case-by-case for $N = 7, 8, \ldots, 32$. The second simulation, a CCSK Monte Carlo Simulation, is written to estimate the average probability of symbol error of CCSK for the 32-chip CCSK sequence chosen for JTIDS in AWGN. In what follows, the major steps of these two simulations are introduced and their results are compared with that of the analytic upper bound.

5.1. Monte Carlo simulation with stratified sampling

This simulation is implemented in a manner similar to that of finding the analytic upper bound discussed in Section 3; that is, the simulation is conducted by stratified sampling based on the value of $N$, the number of chip errors. For $N = 7$ and given that symbol 0 is sent, the major steps of the simulation are as follows. First, in each iteration, (i) randomly generate a 32-chip sequence with seven chip errors relative to the original 32-chip sequence for symbol 0 to obtain a noisy 32-chip sequence $r$, (ii) cross-correlate the noisy 32-chip sequence with all of the 32 local sequences to yield $\langle r, s_i \rangle$ for $i \in \{0, 1, \ldots, 31\}$, (iii) calculate the probability of symbol error based on the following rules: if $\langle r, s_i \rangle > \langle r, s_0 \rangle$ for any $i \in \{1, 2, \ldots, 31\}$, the conditional probability of symbol error is one; if $\langle r, s_i \rangle = \langle r, s_0 \rangle$ for $\tau$ values of $i \in \{1, 2, \ldots, 31\}$, the conditional probability of symbol error is $\tau/(\tau+1)$; if $\langle r, s_i \rangle < \langle r, s_0 \rangle$ for all $i \in \{1, 2, \ldots, 31\}$, the conditional probability of symbol error is zero. Second, repeat the above iteration 10,000 times and calculate the average conditional probability of symbol error. Last, repeat the above process for $N = 8$ through $N = 32$. The overall simulation results, denoted as $\zeta_{\text{SIM}}$, are shown in the third column of Table II. Modifying (19), we obtain the simulation results for the probability of symbol error of CCSK in AWGN from

$$P_S = \sum_{j=0}^{32} \zeta_{\text{SIM}} \left( \begin{array}{c} 32 \\ j \end{array} \right) P_c^j (1-P_c)^{32-j}. \quad (22)$$

5.2. CCSK Monte Carlo simulation

A flow chart of this simulation is shown in Figure 3. As can be seen, this simulation consists of a transmitter with a CCSK symbol modulator and a MSK chip modulator, an AWGN channel, and a receiver with a coherent MSK chip demodulator and a CCSK symbol demodulator. The input to the CCSK symbol modulator $S_i$ for $0 \leq i \leq 31$ is a decimal number which represents a 5-bit symbol. For example, symbol 0 is denoted as $S_0 = 0 = 00000$ and symbol 1 is denoted as $S_1 = 1 = 00001$. The output of the CCSK symbol demodulator $\hat{S}_i$ is the estimate of the symbol received.

The processes of the CCSK Monte Carlo simulation are as follows: (i) in each iteration, 100 random symbols (between symbol 0 and 31) are generated and modulated with CCSK, following which the chips are modulated with MSK for transmission, (ii) the transmitted signal is added to white Gaussian noise in the channel, (iii) the noisy signal is received and de-modulated with coherent MSK at the chip level and with CCSK at the symbol level to obtain an estimate of the

![Figure 3. Flow chart of the CCSK Monte Carlo simulation.](image-url)
received symbol, (iv) the transmitted and received symbols are compared to determine if a symbol error has occurred, (v) the above process is repeated enough times to ensure sufficient accuracy of the simulation. Then the error ratio is calculated for each $E_b/N_0$.

To compare the difference between the analytic upper bound and the two simulations, all results are shown in Figure 4. As is seen, both simulation results match very well and the analytic result given in (19) is a tight upper bound. In addition to giving us good reason to have confidence in the results, this shows the upper bound is tight to within a small fraction of a dB over a wide range of $E_b/N_0$ and SER.

6. A NEW CCSK SEQUENCE

Recall that in the absence of chip errors, the 32-chip CCSK starting sequence chosen for JTIDS has a maximum off-peak cross-correlation value $\max_{1 \leq i \leq 31}(\langle s_i, s_0 \rangle) = 4$. Intuitively, the performance of CCSK can be improved if the maximum off-peak cross-correlation value is smaller than four. Based on this idea, a simple search algorithm was created and a new 32-chip CCSK starting sequence was found. This new starting sequence is 1011 1010 0011 1101 0010 0000 0110 0110. Given that symbol 0 is sent with no chip errors, the off-peak cross-correlation of this new starting sequence has two discrete values: 0 and $-4$; that is, the maximum off-peak cross-correlation for this new starting sequence is zero instead of four.

With the same approach used to evaluate the starting sequence chosen for JTIDS, this new starting sequence is evaluated both analytically and by Monte Carlo simulation with stratified sampling to obtain the conditional probabilities of symbol error $\zeta^*_{UB,j}$ and $\zeta^*_{SIM,j}$, respectively. The results are shown in the fourth and the fifth columns of Table II. As is seen in Table II, the new starting sequence allows for seven chip errors instead of six chip errors in the received sequence without making a symbol error.

Now, replacing $\zeta_{SIM,j}$ with $\zeta^*_{SIM,j}$ in (22), we obtain the probability of symbol error for the new CCSK sequence in AWGN. To compare the difference between the starting sequence chosen for JTIDS and the new CCSK starting sequence, both simulation results are shown in Figure 5. As can be seen, the results obtained with the new CCSK starting sequence are slightly better than those obtained with the original JTIDS starting sequence since the ultimate performance is determined.
Figure 5. Probability of channel symbol error in AWGN (simulation results): the new CCSK sequence versus the CCSK sequence chosen for JTIDS.

at the symbol level rather than at the chip level. In essence, for practical values of $P_c$, the first non-zero term in (22) is not dominant.

7. CONCLUSION

In this paper, the cross-correlation properties of CCSK were formulated, and an analytic upper bound on the probability of symbol error of CCSK was derived for the CCSK starting sequence chosen for JTIDS. The probability of symbol error obtained with the analytic upper bound was compared to the probability of symbol error obtained by two different Monte Carlo simulations for AWGN. The results show that the analytic method yields a tight upper bound. In addition to the CCSK starting sequence chosen for JTIDS, a new CCSK starting sequence with a smaller maximum off-peak cross-correlation value is introduced and evaluated both analytically and by Monte Carlo simulation with stratified sampling. The probability of symbol error obtained for this new CCSK starting sequence compares favorably with that of the starting sequence chosen for JTIDS.

REFERENCES


**AUTHORS’ BIOGRAPHIES**

**Chi-Han Kao** Assistant Professor. Received his PhD from the United States Naval Postgraduate School in 2008. Since then, he has been a faculty member in the Electrical Engineering Department of the R.O.C. Naval Academy, Taiwan. He teaches in digital communications, digital signal processing, and radar principles. His interests include military communications systems, interference mitigation, forward error correction coding, and WiMAX.

**Clark Robertson** is Professor and Chair in the Electrical and Computer Engineering Department at the Naval Postgraduate School (NPS) in Monterey, California, where he has served since 1989. His teaching and research is in the areas of digital modulation and forward error correction coding, spread spectrum systems, and interference mitigation. Dr Robertson earned the BS degree from Texas Tech University and the MS and PhD degrees from the University of Texas at Austin. He has a long and distinguished history of publishing and service in electrical engineering, including six years as a member of the Military Communication (MILCOM) Conference Board and as Chair of the unclassified technical program for MILCOM 2000.

**Frank Kragh** Associate Professor. Has been with the Electrical and Computer Engineering Department at the Naval Postgraduate School (NPS) in Monterey, California since 2003. His chief research interests are digital communications, software defined radio, spectrally efficient modulation and coding, multi-antenna systems, military communications systems, and satellite communications. Dr Kragh holds a BS degree from Caltech, a MS from the University of Central Florida, and a PhD from NPS.

**Kyle Lin** is an Associate Professor of Operations Research at the Naval Postgraduate School. His research interests include stochastic modeling, decision theory, and game theory. He received his PhD in Industrial Engineering and Operations Research from U.C. Berkeley in 2000, and was an Assistant Professor at Virginia Tech before joining the Naval Postgraduate School in 2004.