Codeset Overlay for Complementary Code Keying Direct Sequence Spread Spectrum

Farzad Talebi
Electrical Engineering Department
University of Notre Dame, IN 46556
e-mail: ftalebi@nd.edu

Thomas G. Pratt
Electrical Engineering Department
University of Notre Dame, IN 46556
e-mail: tpratt@nd.edu

***Abstract*—A method to improve communications reliability of complementary code keying in IEEE 802.11b WiFi systems is proposed that involves the use of an *overlay* signaling dimension that preserves the underlying data rate and power spectrum characteristics of the WiFi signals. The specific overlay technique expands the number of code sets and selects the codeset in a data-dependent manner to provide redundancies that improve the error rate performance of the underlying communications scheme. Sequential detection is employed at the receiver, where symbols are recovered using the redundancies embedded in the overlay. The maximum likelihood sequential detection criterion vectors adopted for the scheme correspond to different memory depths, and in the best case achieve gains that are in excess of 3 dB without any rate loss or bandwidth expansion relative to to the original complementary code keying scheme, and the method can be implemented with reasonable increases in computational complexity.**

Keywords–WiFi; CCK; Spread Spectrum; IEEE 802.11b.

I. Introduction

IEEE 802.11b [1] is based on a direct sequence spread spectrum (DSSS) scheme which uses complementary code keying (CCK) in the higher data rate modes. In the highest data-rate scheme, which achieves 11 Mb/s, each symbol conveys 8 bits– 2 bits are determined by the QPSK symbol while the other 6 bits are conveyed from a set of 64 spreading sequences which we refer to as a codeset.Besides its use in indoor wireless LAN applications, CCK technology has been used for other applications [2]-[5] because it is more robust to harsh propagation conditions than the 802.11a or 802.11g [3].

In this paper, we consider the use of additional CCK codesets in a data-dependent manner to impart redundancies for the protection of the data associated with the underlying communications. The overlay encoding technique is complemented by a receiver processing that employs sequence estimation techniques to leverage the encoded redundancies for enhanced communications reliability. The technology provides a range enhancement that would improve the reach of mobile ubiquitous systems.

The idea of using different code sets has been proposed before, for example to support interference mitigation [6]. In the presented scheme, an overlay based on multiple code sets is used as a mechanism to convey data redundancies for enhanced reliability of the underlying communications. The codesets used in the scheme are designed using a search technique to find codesets exhibiting large mutual distance profiles between spreading sequences from different codesets. In the encoding scheme, the codesets are selected in a data-dependent manner and the resulting redundancies are exploited at the receiver using Maximum Likelihood (ML) sequence detection. We show that a signal to noise ratio (SNR) gain in excess of 3 dB can be obtained using seven extra codesets, each with 64 codes, when a memory length of 3 symbols is employed, in comparison to the original CCK scheme at the cost of increased computational complexity.

The remainder of the paper is organized as follows. Section II describes the system model and includes a description of the stochastic search method employed to synthesize the extra code sets. Section III discusses the proposed data transmission scheme that uses a recursion for selecting codeset indices in a data dependent manner. Section IV describes the sliding window ML detection that is implemented for different memory depths and sliding window sizes, including memoryless (F), 2-symbol detection (Y), 3-symbol detection (Z), and 4-symbol detection (X). Section V discusses simulation results and analytic performance characterizations for small window detection vectors. Finally, the conclusions of the research are presented in Section VI.

II. System Model

A. CCK Modulation

In its highest data rate mode, IEEE 802.11b uses direct sequence spread spectrum (DSSS) in an 8-chip per symbol spreading scheme where the chip rate is

 ISBN: 978-1-61208-236-3

11Mchips/s and where the code is drawn from a codeset comprising 64 codes. The spreading sequences in this pre-defined set are given by (1).

$$\begin{aligned} C_1 &= [C_1(1), C_1(2), \ldots, C_1(8)] \\ &= [e^{j(\phi_2+\phi_3+\phi_4)}, e^{j(\phi_3+\phi_4)}, e^{j(\phi_2+\phi_4)}, \ldots \\ &\quad -e^{j(\phi_2+\phi_4)}, e^{j(\phi_2+\phi_3)}, e^{j(\phi_2+\phi_3)}, -e^{j(\phi_2)}, 1] \end{aligned} \quad (1)$$

The phases ϕ_2, ϕ_3, and ϕ_4 are selected from the set $\{0,\pi/2,\pi,3\pi/2\}$, and the resulting codes within the set are designated by $C_1(i),\ 1 \leq i \leq 64$.

B. Code Set Expansion

Few design methods for creating additional CCK code sets have been proposed in literature. In one reference, Cotae [8] has proposed a method for designing orthogonal sets of spreading sequences for use in overloaded multi-cell code division multiple access (CDMA) systems. The approach allowed for complex valued spreading sequence designs according to a total weighted square correlation criterion. Furthermore, Xu et al. [9] provides a method to mitigate co-channel interference in cellular communication systems using nearly orthogonal CCK codesets at different cells, but it does not provided any intuition or criteria to help understanding the process of designing the nearly orthogonal codesets. Such methods do not appear to have direct application to our problem and so we proceed with a stochastic search method.

The distance between code sequences will be an important design metric in the synthesis of additional code sets since in additive white Gaussian noise (AWGN) ML detection can be reduced to minimum distance detection [10]. Our goal is to achieve distance profiles comparable to those associated with the original code set. The distance profile between the first and other sequences in the original codeset is plotted in Figure 1.

The method we use to design extra codesets $C_2, \ldots, C_8$ involves a stochastic search to minimize a cost function $f(C_2, \ldots, C_8)$ which is proportional to the probability of error [10]. This cost function is associated with a system using codesets C_2 to C_8, where the cost function is the uniform average of error probabilities associated with all spreading sequences within all codesets:

$$f(C_2, \ldots, C_8) = \sum_{\substack{2 \leq m,n \leq 8 \\ 1 < i,j < 64}} Q(\sqrt{\|c_m(i) - c_n(j)\|^2}). \quad (2)$$

In (2), the function Q represents the normal CDF, and $c_m(i)$ represents the i-th code sequence from codeset C_m.

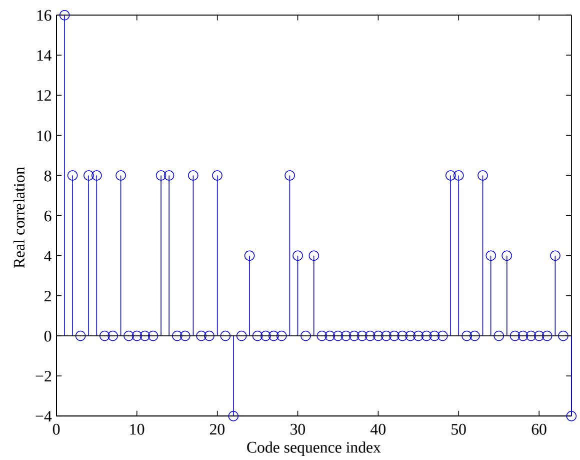

Fig. 1. Cross correlation between the first code set and all 64 original CCK code sequences

Minimization of $f(C_2, \ldots, C_8)$ is achieved via a random search which assumes that permutated and rotated versions of the original CCK code set C_1 can be used as $C_2, \ldots, C_8$. Seven code sets resulting from such a search are given by :

$$\begin{aligned} C_2 &= [C_1(6), -C_1(4), -jC_1(5), -C_1(3), \ldots \\ &\quad -jC_1(8), -jC_1(7), -jC_1(2), -jC_1(1)] \\ C_3 &= [jC_1(5), C_1(8), C_1(7), -jC_1(6), \ldots \\ &\quad -jC_1(2), -jC_1(4), jC_1(1), -jC_1(3)] \\ C_4 &= [-jC_1(2), C_1(8), -C_1(4), -C_1(3), \ldots \\ &\quad -C_1(5), -jC_1(6), C_1(1), -jC_1(7)] \\ C_5 &= [-C_1(6), C_1(4), jC_1(2), -jC_1(3), \ldots \\ &\quad -C_1(1), jC_1(5), -jC_1(7), -jC_1(8)] \\ C_6 &= [-C_1(6), -jC_1(8), -C_1(2), -jC_1(1), \ldots \\ &\quad -C_1(5), -C_1(3), -jC_1(4), jC_1(7)] \\ C_7 &= [-C_1(7), -C_1(4), -jC_1(3), C_1(2), \ldots \\ &\quad C_1(1), -C_1(5), C_1(6), -C_1(8)] \\ C_8 &= [-C_1(7), jC_1(8), -jC_1(6), jC_1(5), \ldots \\ &\quad -C_1(3), -jC_1(4), -C_1(2), -C_1(1)] \end{aligned} \quad (3)$$

III. Data Dependent Code set Selection

Now, we describe the method proposed for adopting extra code sets into the spreading scheme. This approach involves selection of a data-dependent code set index for each transmitted symbol. Let us assume a packet of length $N_p = 1000$ symbols, which corresponds to a medium-length packet length in WLAN systems. Let $S(k)$ designate the code set index for symbol k in the packet, and let $N(k)$ represents the code sequence index for the k-th symbol, where the code sequence is chosen from the members in $S(k)$. Let us assume that the first and second code set and code number indices

 ISBN: 978-1-61208-236-3

are pre-determined for each packet (for the sake of initialization). Subsequent code set indices are selected through a recursion, which in the following example has a memory depth of three symbols:

$$S(k) = \lceil mod(\lceil \frac{N(k-1)}{N_R} \rceil + S(k-1) + \ldots N(k-2) + S(k-2), N_R) \rceil \quad (4)$$

Here, $mod(a,b)$ is the remainder of a divided by b, and N_R is the number of code sets (assumed to be 8 in our analysis). We emphasize that $N(k)$ will only depend on the k-th symbol in the data packet and is not affected by priori code numbers or code set indices. The recursion for choosing the next code set index helps to discriminate against false peaks in the detection process by partitioning indices into groups of N_R, where for a given code sequence (i.e., the data) each group results in the use of a different code set index for the next symbol, thereby achieving data-dependent codeset selection.

IV. MULTI-SYMBOL PROCESSING

To leverage the redundancies achieved through the overlay, multi-symbol detection processing is used at the receiver. The method amounts to a greedy implementation of an exhaustive search to find the best code word index according to a minimum distance (or maximum correlation) criterion which is optimal in a ML detection sense for the case of AWGN, which we assume in our analysis. When the transition between states is unrestricted (as in our case) minimum distance detection of a finite number of possible symbols should be achieved using exhaustive search because a structure (channel code) ruling the sequence of selected indices does not exist. The greedy approach can be implemented in a sliding window manner using different window lengths. Because the inherent memory in the encoding process in (4) is three, we anticipate good performance gain when using detection windows of length 3 or more, which is verified in the simulation results.

Symbol-by-symbol detection corresponds to the conventional detection approach that would be employed in an 802.11b system, and serves as a baseline against which to compare the performance of the overlay approach. Symbol-by-symbol detection employs a metric $F(N_k = j_k)$, which represents the correlation of the received symbol at time k with pre-known symbol N_k-th in code set $S(k)$. At time k the receiver knows $S(k)$ because it depends on previous symbols N_{k-1} , N_{k-2} , $\cdots$. The metric F is computed as follows:

$$F(N_k = j_k) = \sum_{i=1}^{8} 2Re(c_{j_k}(i)c_{rx,k}(i)^*) \quad (5)$$

where the code word c_{j_k} is the j_k-th symbol from the code set defined by the previous symbols, and $c_{rx,k}$ represents the received symbol at time k.

The metric for the two-symbol detection scheme can be written as:

$$Y(j_{k-3}) = F(N_{k-3} = j_{k-3}) + \max_{j_{k-2}}\{F(N_{k-2} = j_{k-2}|N_{k-3} = j_{k-3})\} \quad (6)$$

which reflects the correlation between symbols received at times $k-3$ and $k-2$ based on the decoded estimated of the previous symbols. In (6), $F(N_k = j_k|N_{k-1} = j_{k-1})$ refers to the correlation of the received symbol at time k with the symbol N_k set equal to the j_k-th symbol in $S(k)$ conditioned on the premise that the symbol at time $k-1$, N_{k-1}, is equal to the j_{k-1}-th symbol in code set $S(k-1)$.

By following the same approach, a three-symbol detection metric vector Z can be computed that reflects the correlation between received symbols at times $k-3$,$k-2$ and $k-1$ and is given by (7):

$$Z(j_{k-3}) = F(N_{k-3} = j_{k-3}) + \max_{j_{k-2}}\{F(N_{k-2} = j_{k-2}|N_{k-3} = j_{k-3}) + \max_{j_{k-1}}\{F(N_{k-1} = j_{k-1}|N_{k-2} = j_{k-2}, \ldots N_{k-3} = j_{k-3})\}\} \quad (7)$$

In a like manner, a four symbol detection metric is achieved using (8):

$$X(j_{k-3}) = F(N_{k-3} = j_{k-3}) + \max_{j_{k-2}}\{F(N_{k-2} = j_{k-2}|N_{k-3} = j_{k-3}) + \max_{j_{k-1}}\{F(N_{k-1} = j_{k-1}|N_{k-2} = j_{k-2}, \ldots N_{k-3} = j_{k-3}) + \max_{j_k}\{F(N_k = j_k|N_{k-1} = j_{k-1}, \ldots N_{k-2} = j_{k-2}, N_{k-3} = j_{k-3})\}\}\} \quad (8)$$

Note that in the above derivations, it is assumed that symbols before time $k-3$ are known, which means that symbol errors will propagates through the sequential demodulation of the packet. However, since we are interested in packet error rate (PER) any error is sufficient to result in loss of the packet.

V. PERFORMANCE ANALYSIS AND SIMULATION RESULTS

A. Analytic Performance Analysis

The symbol error rate associated with the symbol-by-symbol detection over additive white Gaussian noise

 ISBN: 978-1-61208-236-3

TABLE I
SQUARED DISTANCE AND MULTIPLICITY

i	**Squared distance** (d_i^2)	**Multiplicity** m_i
1	8	12
2	12	6
3	16	43
4	20	2

TABLE II
SECOND ORDER SQUARED DISTANCE AND MULTIPLICITY

i	**Squared distance** (d_i^2)	**Multiplicity** m_i
1	8	4
2	10	0.05
3	12	0.5
4	14	1.75

(AWGN) channel can be written as [10]:

$$P_e = \sum_i m_i \; Q(\sqrt{d_i^2}) \tag{9}$$

where m_i represents the multiplicity of neighbors at distance d_i, which is given in Table I for the original CCK code set. The corresponding PER for symbol by symbol detection and for two-symbol detection matches results corresponding to an assumption of independent symbols and can be written as (10):

$$\text{PER} = 1 - (1 - P_e)^{N_p} \approx N_p P_e \tag{10}$$

Using the distance-multiplicity data in Table I analytically-determined results for the PER of memory-less detection using F is presented in Figure 2. However, to evaluate the performance of scheme when using two-symbol detection with vector Y as metric, second order distance distributions of codewords are required that can be obtained numerically thanks to the short length and small number of codewords. This distance distribution is given in Table II; by using the distance distribution, the analytic PER performance of schemes Y can be obtained.

B. Simulation Results

Figure 2 shows the PER performance of different simulated detection schemes. The simulation assumed packets consisting of 1000 symbols, where each symbol (i.e., code) carries 6 bits of information where the codes have 8 chips. The SNR is defined as E_c/N_0, where E_c represents chip energy, and is assumed to be unity, and N_0 is the variance of complex Gaussian noise with variance $N_0/2$ on the in-phase and quadrature (I and Q) channels.

TABLE III
RUNTIME AND MEMORY USAGE FOR PACKET LENGTH=1000

Vector	**Runtime**	**Memory Usage (Bytes)**
F	54 msec	512
Y	195 msec	2560
Z	2.14 sec	100352
X	83 sec	6359040

Performance curves are plotted for simulation results associated with detections based on the F, Y, Z and X vectors. We observe that Y, Z, and X yield approximately 0.5 dB, 2.5 dB, and 3 dB gains. As Table II shows, using vector Y (i.e., using two consecutive symbols for detection) reduces the multiplicity of nearest neighbors at distance $d^2 = 8$, so the gain is not expected to be large because the minimum distance has not been reduced. Though we have not reported the distance distribution for the case of using vector Z, simulation results indicate that the minimum distance is increased to $d^2 = 14$, reflected by the SNR gain (about 2.5 dB) relative to F, which represents the original CCK scheme used in IEEE 802.11b. Detection using vector X results in a gain exceeding 3 dB in comparison to original scheme, but is considerably more complex.

The runtime and memory requirements of the described detection schemes are shown in Table III, which indicates a memory usage growth with factors 5, 39 and 64 for each additional memory depth, while the runtimes grow with factors 4, 10 and 40 respectively. Note that the memory usage is expressed in terms of bytes in Table III. Since the runtime and memory usage are dependent on the implementation, the numbers presented are based on our specific implementation using MATLAB on a computer with a 2GHz Dual Core processor. Implementations using FPGA-based hardware that better exploit parallel calculations are anticipated to yield much better computational efficiencies.

Figure 3 compares the required SNR to achieve a PER = 0.01 for different packet lengths. SNR differences between different schemes are seen to be independent of the packet length. However, the required SNR is seen to increase by approximately 1 dB when the packet size changes from 100 symbols to 2000 symbols. This can be of interest for mobile ubiquitous systems where packet sizes as small as 64 bytes might be used [11].

 ISBN: 978-1-61208-236-3

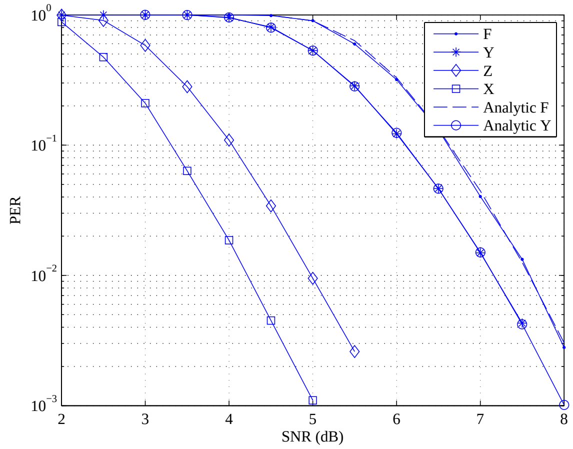

Fig. 2. PER performance of different detection schemes

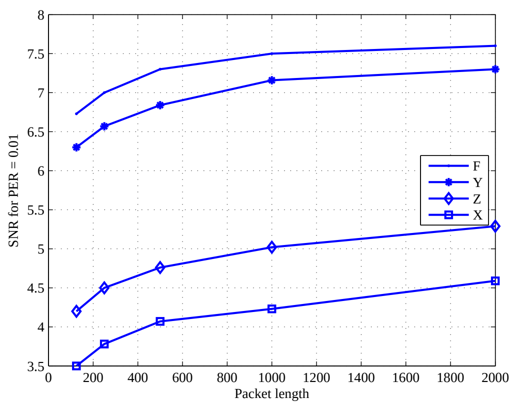

Fig. 3. Required SNR to reach PER = 0.01

VI. Conclusion And Future Work

The use of additional codesets configured as an overlay was shown to improve the error rate performance of an underlying CKK-DSSS signal. The scheme employed the extra code sets that are selected in a data-dependent manner to improve noise immunity without bandwidth expansion or data rate reduction. For encoding involving a recursion that uses three consecutive symbols, we have shown that different gains are achieved depending upon the memory depth assumed at the decoder, where up to more than 3 dB SNR gain is possible when a memory depth of three symbols is employed. The proposed method is seen to exhibit reasonable implementation complexities for most of the processing cases that were considered. Also, it has better performance at shorter block lengths which is achieved without rate loss and forward error correction coding.

A future step in this work is to reduce the computational complexity and memory usage of the detection scheme through algorithm implementation efficiencies and through more efficient numerical representations. For example, our analysis was conducted using double-precision floating point representations. However, other detection/decoding algorithms reported in literature have represented detection vector entries using quantization levels of 4 or 5 bits [12] without significant reduction in performance. Carefully designed quantization schemes can potentially achieve more efficient implementations while simultaneously maintaining a detection performance close to that of the unquantized scheme.

References

[1] Institute of Electrical and Electronics Engineers, "Standard 802.11b-1999, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band," Jan. 2000.

[2] P. Bhagwat, B. Raman, and D. Sanghi "Turning 802.11 inside-out," ACM SIGCOMM Computer Commun. Rev., vol. 34, no. 1, pp. 33–38, Jan. 2004.

[3] K. K. Leung, M. V. Clark, B. McNair, Z. Kostic, L. J. Cimini Jr., and J. H. Winters, "Outdoor IEEE 802.11 cellular networks: radio and MAC design and their performance, IEEE Trans. Veh. Technol., vol. 56, no. 5, pp. 2673–2684, Sept. 2007.

[4] A. Silvennoinen, T. Karhima, M. Hall, and S. G. Haggman, "IEEE 802.11b WLAN capacity and performance measurements in channel with large delay spreads," in Proc. 2004 IEEE Military Communications Conference, vol. 2, pp. 693–696, Oct. 2004.

[5] V. Arneson, K. Ovsthus, O. I. Bentstuen, and J. Sander, "Field trials with IEEE 802.11b-based UHF tactical wideband radio," in Proc. IEEE Military Communications Conference vol. 1, pp. 493–498, Oct. 2005.

[6] X. Xiaohui, Z. Chao and L. Xiaokang, "Using different orthogonal code sets for CCK modulation to mitigate co-channel interference among WLANs,", In Proc. IEEE International Symposium on Communications and Information Technology (ISCIT), vol. 2, pp. 885–888, 2005.

[7] T.-D. Chiueh and S.-M. Li, "Trellis-coded complementary code keying for high-rate wireless LAN system," IEEE Communications Letters, vol. 5, pp. 191–193, May 2001.

[8] P. Cotae, "Multicell Spreading Sequence Design Algorithm for Overloaded S-CDMA," IEEE COMMUNICATIONS LETTERS, VOL. 9, NO. 12, DECEMBER 2005, pp. 1028–1030

[9] Xiaohui Xu, Chao Zhang and Xiaokang Lin, "Using different orthogonal code sets for CCK modulation to mitigate co-channel interference among WLANs," IEEE International Symposium on Information Technology and Communications, 2005. ISCIT vol. 2, pp. 885–888.

[10] Todd K. Moon, "Error Correction Coding Mathematical Methods and Algorithms," John Wiley & Sons, Inc. 2005.

[11] C.-K. Toh, M. Delwar, and D. Allen, "Evaluating the Communication Performance of an Ad Hoc Wireless Network," IEEE transactions on wireless communications, vol. 1, no. 3, July 2002.

[12] I.M. Onyszchuk, K.-M. Cheung, O. Collins, "Quantization loss in convolutional decoding," IEEE Transactions on Communications, vol. 4, no. 2, pp. 261–265, 1993.