APPENDIX A

OPTIMAL RECURSIVE SYSTEMATIC CONVOLUTIONAL CODES

A.1 APPENDIX OVERVIEW

THE most extensive set of the best rate $R_c = k/n$ RSC coders thus far, has been presented by *Benedetto*, *Garello* and *Montorsi* in [88]. Their search approach for the best codes was based on a minimal encoder description as a finite-state machine, derived from a group-theoretic approach to binary convolutional codes. This appendix summarises the encoder parameters of the optimal rate $R_c = 1/4$, $R_c = 1/3$, $R_c = 1/2$, $R_c = 2/4$, $R_c = 2/3$, $R_c = 3/4$ and $R_c = 4/5$ RSC codes obtained from their exhaustive searches. The minimum free distance d_{free} of each encoder is also given. For illustrative purposes, these configuration parameters are used to construct an optimal 8-state, rate $R_c = 2/3$ RSC code encoder.

A.2 TABLES OF OPTIMAL RSC CODE ENCODER PARAMETERS

Instead of using the classic approach of describing encoder structures by means of their generator polynomials or matrices, *Benedetto*, *Garello* and *Montorsi* [88] opted to describe their set of optimal RSC code encoders using the following parameters:

- υ_a = Number of delay elements in the shift register associated with the a^{th} message word bit in the encoder input vector $\overline{d}_{m,i}$.
- 2^{υ} = Number of states in the code's trellis. The parameter $\upsilon = \sum_{a=0}^{k-1} \upsilon_a$ denotes the total number of delay elements used in the encoder.
- $z(a,b)$ = Output generator polynomial, given in octal form. When converted to a binary sequence, it indicates the tap connections associated with the a^{th} shift register that contribute to the b^{th} non-systematic output bit of the encoder.
- $h(a,b)$ = Feedback generator polynomial, given in octal form. When converted to a binary sequence, it indicates the tap connections associated with the a^{th} shift register that contribute to the input of the b^{th} shift register.

These parameters are used in *Table* A.1, *Table* A.2, *Table* A.3, *Table* A.4, *Table* A.5, *Table* A.6 and *Table* A.7 to define the structures of optimal rate $R_c = 1/4$, $R_c = 1/3$, $R_c = 1/2$, $R_c = 2/4$, $R_c = 2/3$, $R_c = 3/4$ and $R_c = 4/5$ RSC code encoders, respectively.

Table A.1: Encoder Descriptions of Optimal Rate $R_c = 1/4$ RSC Codes

2^v	v_i	**Output Generators** $z(0,b)$	**Feedback Generators** $h(0,b)$	d_{free}
2	1	$1_8, 3_8, 2_8$	3_8	6
4	2	$5_8, 7_8, 6_8$	7_8	10
8	3	$15_8, 17_8, 11_8$	13_8	12
16	4	$35_8, 37_8, 27_8$	23_8	14
32	5	$51_8, 45_8, 71_8$	67_8	15

Table A.2: Encoder Descriptions of Optimal Rate $R_c = 1/3$ RSC Codes

2^v	v_i	**Output Generators** $z(0,b)$	**Feedback Generators** $h(0,b)$	d_{free}
2	1	$3_8, 2_8$	3_8	5
4	2	$7_8, 5_8$	7_8	8
8	3	$15_8, 17_8$	13_8	10
16	4	$37_8, 33_8$	23_8	10
32	5	$51_8, 45_8$	67_8	11
64	6	$131_8, 101_8$	163_8	11

Table A.3: Encoder Descriptions of Optimal Rate $R_c = 1/2$ RSC Codes

2^v	v_i	**Output Generators** $z(0,b)$	**Feedback Generators** $h(0,b)$	d_{free}
2	1	2_8	3_8	3
4	2	5_8	7_8	5
8	3	17_8	13_8	6
16	4	37_8	23_8	6
32	5	17_8	67_8	8
64	6	115_8	147_8	9

Table A.4: Encoder Descriptions of Optimal Rate $R_c = 2/4$ RSC Codes

2^v	v_i	**Output Generators**		**Feedback Generators**		d_{free}
		$z(0,b)$	$z(1,b)$	$h(0,b)$	$h(1,b)$	
2	$1,0$	$1_8, 2_8$	$1_8, 1_8$	$3_8, 0_8$	$1_8, 1_8$	4
4	$1,1$	$3_8, 3_8$	$3_8, 0_8$	$0_8, 3_8$	$3_8, 2_8$	5
8	$2,1$	$3_8, 5_8$	$3_8, 1_8$	$2_8, 5_8$	$3_8, 2_8$	5
16	$2,2$	$1_8, 3_8$	$5_8, 7_8$	$5_8, 4_8$	$2_8, 5_8$	6

Table A.5: Encoder Descriptions of Optimal Rate $R_c = 2/3$ RSC Codes

2^v	v_i	**Output Generators**		**Feedback Generators**		d_{free}
		$z(0,b)$	$z(1,b)$	$h(0,b)$	$h(1,b)$	
2	$1,0$	2_8	0_8	$3_8, 0_8$	$1_8, 1_8$	2
4	$1,1$	0_8	3_8	$2_8, 3_8$	$3_8, 0_8$	3
8	$2,1$	7_8	1_8	$0_8, 5_8$	$3_8, 2_8$	4
16	$2,2$	5_8	3_8	$6_8, 3_8$	$5_8, 4_8$	5
32	$3,2$	15_8	7_8	$0_8, 13_8$	$7_8, 0_8$	6
64	$3,3$	1_8	11_8	$13_8, 12_8$	$16_8, 1_8$	6

Table A.6: Encoder Descriptions of Optimal Rate $R_c = 3/4$ RSC Codes

2^v	v_i	**Output Generators**			**Feedback Generators**			d_{free}
		$z(0,b)$	$z(1,b)$	$z(2,b)$	$h(0,b)$	$h(1,b)$	$h(2,b)$	
2	$1,0,0$	2_8	0_8	0_8	$3_8, 0_8, 0_8$	$1_8, 0_8, 1_8$	$1_8, 1_8, 0_8$	2
4	$1,1,0$	2_8	1_8	1_8	$2_8, 1_8, 0_8$	$2_8, 3_8, 1_8$	$1_8, 0_8, 0_8$	3
8	$2,1,0$	7_8	3_8	1_8	$0_8, 5_8, 0_8$	$3_8, 0_8, 0_8$	$1_8, 1_8, 1_8$	4
16	$2,2,0$	7_8	5_8	1_8	$0_8, 7_8, 0_8$	$7_8, 0_8, 0_8$	$1_8, 1_8, 1_8$	4
32	$2,2,1$	1_8	6_8	3_8	$3_8, 0_8, 5_8$	$6_8, 0_8, 5_8$	$2_8, 3_8, 3_8$	5

Table A.7: Encoder Descriptions of Optimal Rate $R_c = 4/5$ RSC Codes

2^{υ}	υ_i	**Output Generators**				**Feedback Generators**				d_{free}
		$z(0,b)$	$z(1,b)$	$z(2,b)$	$z(3,b)$	$h(0,b)$	$h(1,b)$	$h(2,b)$	$h(3,b)$	
2	$1,0,$ $0,0$	2_8	0_8	0_8	0_8	$3_8,0_8,$ $0_8,0_8$	$1_8,0_8,$ $0_8,1_8$	$1_8,0_8,$ $1_8,0_8$	$1_8,1_8,$ $0_8,0_8$	2
4	$1,1,$ $0,0$	2_8	1_8	1_8	0_8	$2_8,1_8,$ $0_8,0_8$	$2_8,3_8,$ $0_8,1_8$	$0_8,1_8,$ $1_8,0_8$	$1_8,0_8,$ $0_8,0_8$	2
8	$1,1,$ $1,0$	0_8	3_8	1_8	1_8	$2_8,0_8,$ $3_8,0_8$	$0_8,1_8,$ $2_8,0_8$	$2_8,3_8,$ $1_8,1_8$	$1_8,0_8,$ $0_8,0_8$	3

A.3 ENCODER CONSTRUCTION EXAMPLE

Fig. A.1 shows the generic structure of an 8-state, rate $R_c = 2/3$ RSC code. In this figure, $d_{m,i,0}$ and $d_{m,i,1}$ denote the i^{th} pair of input data bits within the m^{th} vector of input bits, whereas $\text{v}_{m,i,0}$ denotes the i^{th} single output parity bit within the m^{th} vector of output bits. Applying configuration parameters such as those specified in *Section* A.2, this generic encoder structure can be altered to obtain several distinctly different 8-state, rate $R_c = 2/3$ RSC code encoders. According to *Table* A.5 (code no. 3), the best rate $R_c = 2/3$ 8-state RSC code encoder is constructed by setting $z(1,1) = 7_8$, $z(2,1) = 1_8$, $h(1,1) = 0_8$, $h(2,1) = 3_8$, $h(1,2) = 5_8$ and $h(2,2) = 2_8$ in *Fig.* A.1. The resultant optimal RSC code encoder is shown in *Fig.* A.2.

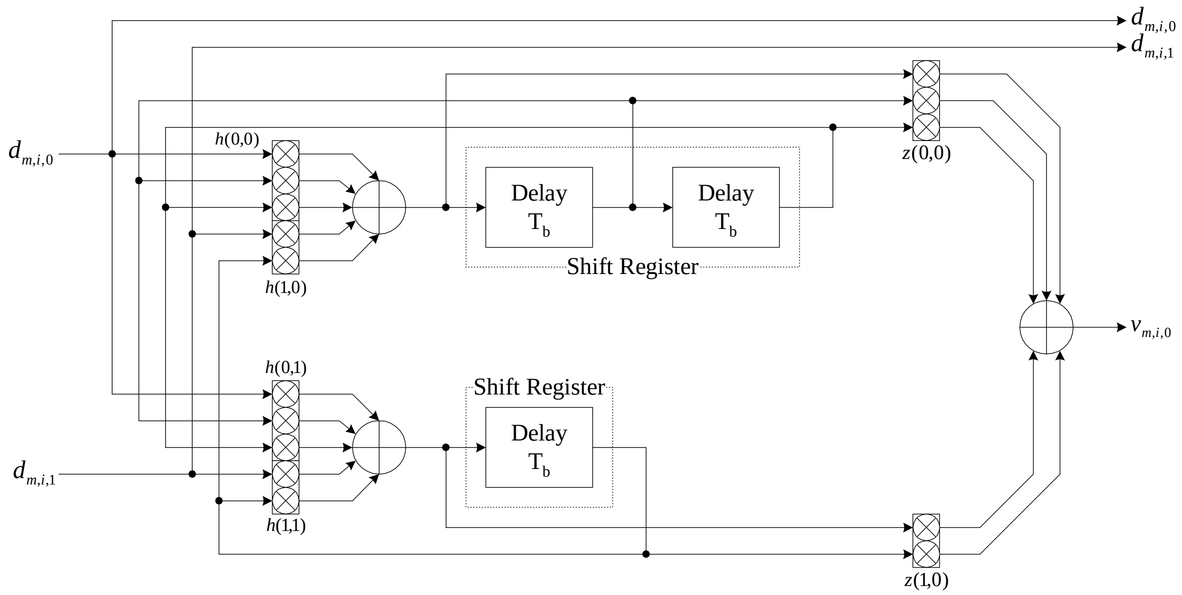

Figure A.1: General Structure of a 8-State, Rate $R_c = 2/3$ Minimal Linear Systematic Convolutional Code Encoder

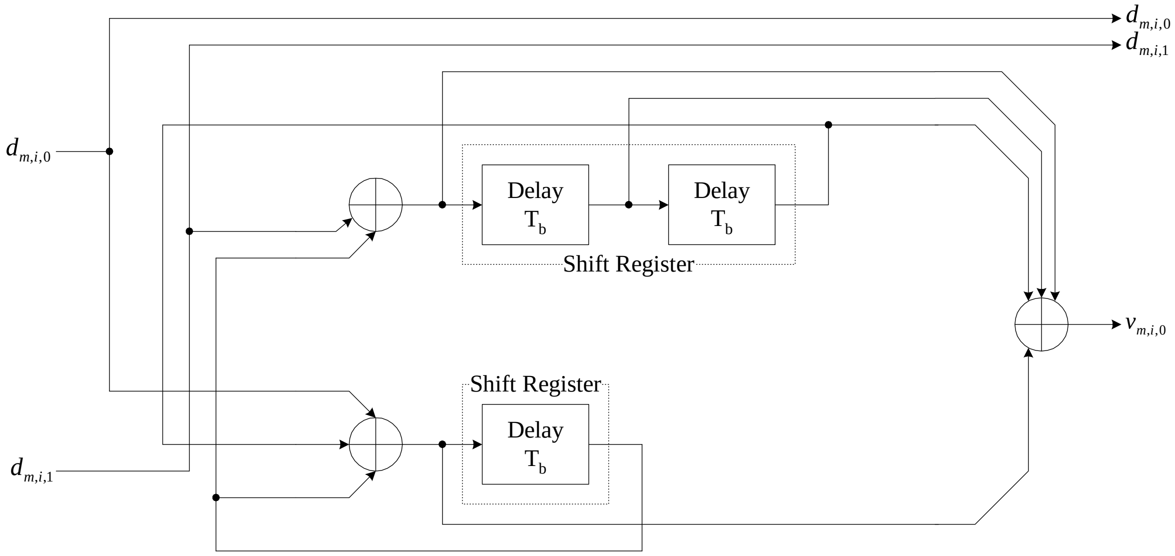

Figure A.2: Optimal 8-State, Rate $R_c = 2/3$ Minimal Linear Systematic Convolutional Code Encoder

APPENDIX B

BERLEKAMP-MASSEY DECODING OF REED-SOLOMON BLOCK CODES

B.1 APPENDIX OVERVIEW

THIS appendix presents a conceptual description of the *Berlekamp-Massey* decoding algorithm [74, 75], frequently employed in the syndrome decoding of classic BCH and RS block codes. Since it falls beyond the scope of this study, the classic *Berlekamp-Massey* algorithm is not described in detail. However, several valuable references that focus on variations of this decoding algorithm are cited for the interested reader.

B.2 THE BERLEKAMP-MASSEY ALGORITHM

The classic approach followed in the decoding of RS block codes entails hard decision syndrome decoding [94], which is described below (all mathematical operations are performed in $GF\left(2^{\xi}\right)$): Assume that $c_m(p) = c_{m,n-1}.p^{n-1} + c_{m,n-2}.p^{n-2} + ... + c_{m,0}$ is the code word polynomial generated by an (n, k, d_{min}) $GF\left(2^{\xi}\right)$ RS block code encoder at encoding instance m, given the message polynomial $d_m(p) = d_{m,k-1}.p^{k-1} + d_{m,k-2}.p^{k-2} + ... + d_{m,0}$. This code word is then transmitted through a non-ideal communication channel. Let $y_m(p) = y_{m,n-1}.p^{n-1} + y_{m,n-2}.p^{n-2} + ... + y_{m,0}$ represent the corrupted code word after hard decisions have been made on the received and demodulated code word symbols. The relationship between this polynomial and the original code word polynomial $c_m(p)$ is as follows [94]:

$$y_m(p) = c_m(p) + e_m(p) \tag{B.1}$$

where $e_m(p) = e_{m,n-1}.p^{n-1} + e_{m,n-2}.p^{n-2} + ... + e_{m,0}$, referred to as the *error polynomial*, describes the alterations made by the channel to the original code word during transmission.

From *Section* 3.2.2.3.3.1 it follows that the number of parity symbols present in each code word, generated by a $t_{correct}$-symbol error correcting RS block code, is $n - k = 2.t_{correct}$. Thus, $2.t_{correct}$ syndromes can be calculated for the received code word. The i^{th} syndrome, with $i = 1, 2, ..., 2.t_{correct}$, is calculated as follows [74, 75, 94, 175]:

$$\$_m^i = y\left(\varphi^i\right) = c_m\left(\varphi^i\right) + e_m\left(\varphi^i\right) = e_m\left(\varphi^i\right) \tag{B.2}$$

where φ is the primitive element of $GF\left(2^{\xi}\right)$. In *Eq.* (B.2) $c_m\left(\varphi^i\right)=0$, because φ^i is a root of the block code's generator polynomial (see *Section* 3.2.2.3.3.2).

Assume that a number of $M_{correct}$, with $M_{correct} \leq t_{correct}$, correctable errors are present in $y_m(p)$, situated at positions $\chi_m^1, \chi_m^2, ..., \chi_m^{M_{correct}}$, where $0 \leq \chi_m^j < n$ for $j=1,2,...,M_{correct}$. For each symbol in error, an *error locater* is defined [74, 75, 94, 175]:

$$\kappa_m^j = (\varphi)^{\chi_m^j} \qquad \text{for } j=1,2,...,M_{correct} \tag{B.3}$$

Noting that only symbols received in error contribute to the syndrome values, it is possible to rewrite *Eq.* (B.2) in terms of the error locators:

$$\$_m^i = \sum_{j=1}^{M_{correct}} e_{m,\chi_m^j}.\left(\kappa_m^j\right)^i \tag{B.4}$$

The *error locator polynomial* polynomial $\Theta_m(p)$, which describes the error pattern present in $y_m(p)$, is defined as a polynomial whose inverse roots are the error locators [74, 75, 94, 175]:

$$\Theta_m(p) = \prod_{j=1}^{M_{correct}} \left(1-\kappa_m^j.p\right) \tag{B.5}$$

The *Berlekamp-Massey* algorithm is an iterative algorithm that computes both the $2.t_{correct}$ syndromes and the error locator polynomial $\Theta_m(p)$. A detail description of this algorithm, which can be performed in both the time and frequency domains, falls beyond the scope of this study. The interested reader is referred to [74, 75, 175] and [176] for descriptions of the time and frequency domain versions of the *Berlekamp-Massey* algorithm, respectively.

In order to decode binary block codes, knowledge of the error positions in the received code words are sufficient information. However, with non-binary block codes, the error magnitudes must also be determined: Assume that the error locater polynomial $\Theta_m(p)$ for the received code word $y_m(p)$ has been successfully constructed. Using the error locator polynomial's coefficients and the $2.t_{correct}$ syndromes, an *error evaluator polynomial* $\nabla_m(p)$ is determined [175]:

$$\nabla_m(p) = \Theta_m(p) + \sum_{j=1}^{M_{correct}} \left[e_{m,\chi_m^j}.\kappa_m^j.p \prod_{\substack{i=1\\ i\neq j}}^{M_{correct}} \left(1-\kappa_m^i.p\right)\right] \tag{B.6}$$

An estimate of the magnitude of the error at position χ_m^j, with $j=1,2,...,M_{correct}$, is then calculated as follows [175]:

$$\hat{e}_{m,\chi_m^j} = -\left.\left(\frac{\kappa_m^j.\nabla_m(p)}{\frac{d\Theta(p)}{dp}}\right)\right|_{p=1/\left(\kappa_m^j\right)} \tag{B.7}$$

Completing the hard decision syndrome decoding process involves constructing an estimate of the original code word polynomial, denoted by $\hat{c}_m(p)$. This is accomplished by subtracting $\hat{e}_{m,\chi_m^j}$ from position χ_m^j in $y_m(p)$, for $j=1,2,...,M_{correct}$.

APPENDIX C

POPULAR BLOCK INTERLEAVERS

C.1 APPENDIX OVERVIEW

SEVERAL prevalent block interleaver structures, frequently encountered in iteratively and non-iteratively decoded concatenated coding schemes, are considered in this appendix. The implementable interleavers considered are divided into two categories: Deterministic and random interleavers. The deterministic interleavers presented include classic block interleavers, *Berrou-Glavieux* interleavers and JPL interleavers. PN generator interleavers, random number generator interleavers and s-random interleavers constitute the random interleavers of interest. The appendix is concluded with a short discussion on the concept of a probabilistic uniform interleaver, frequently encountered in the theoretical performance evaluations of concatenated codes.

C.2 DETERMINISTIC BLOCK INTERLEAVERS

A deterministic block interleaver has a mapping function $\Pi(i)$ that, for a given interleaver depth J and interleaver width F, always produces the same fundamental permutation. Several popular deterministic block interleaver mapping schemes are discussed in the following subsections.

C.2.1 CLASSIC BLOCK INTERLEAVERS

A classic block interleaver [87] consists in essence of a size $J \times F$ memory matrix. The first step in the interleaving process performed by this type of interleaver is to write the interleaver input symbols into this matrix in a row-wise fashion. The second and final step whereby the interleaver output symbols are obtained, is to write out the data stored in the memory matrix in a column-wise fashion. It is easy to see that the interleaver period of this type of interleaver is $N = J.F$.

C.2.2 BERROU-GLAVIEUX INTERLEAVERS

The depth and width of this type of interleaver are restricted to powers of two, i.e. $J = 2^a$ and $F = 2^b$, with a and b positive integer values. Firstly, a set of eight prime numbers are defined: $X(1) = 17$, $X(2) = 37$, $X(3) = 19$, $X(4) = 29$, $X(5) = 41$, $X(6) = 23$, $X(7) = 13$ and $X(8) = 7$. The interleaver mapping function is then defined as follows [87]:

$$\Pi(i) = \text{Ø}(i) + F.\digamma(i) \tag{C.1}$$

where:

$$\begin{aligned}
\digamma_0 &= i \bmod F \\
Ø_0 &= (i - \digamma_0)/F \\
\daleth &= (\digamma_0 + Ø_0) \bmod 8 \\
\digamma(i) &= (X(\daleth + 1) \,.\, (Ø_0 + 1) - 1) \bmod J \\
Ø(i) &= (F/2 + 1) \,.\, (\digamma_0 + Ø_0) \bmod F
\end{aligned} \tag{C.2}$$

for every $0 \leq i < J \cdot F = N$.

C.2.3 JPL INTERLEAVERS

With JPL interleavers, the interleaver depth J must be even. As with *Berrou-Glavieux* interleavers, eight prime values are defined: $X(1) = 31$, $X(2) = 37$, $X(3) = 43$, $X(4) = 47$, $X(5) = 53$, $X(6) = 59$, $X(7) = 61$ and $X(8) = 67$. For every $0 \leq i < J.F = N$, the JPL interleaver mapping function is defined as follows [87]:

$$\Pi(i) = 2.\digamma(i) + J.Ø(i) - \mho(i) + 1 \tag{C.3}$$

where:

$$\begin{aligned}
\mho(i) &= i \bmod F \\
Ø_0 &= (i - \mho(i))/2 \bmod F \\
\digamma_0 &= ((i - \mho(i))/2 - Ø_0)/F \\
\digamma(i) &= (10.\digamma_0 + 1) \bmod J/2 \\
\daleth &= \digamma(i) \bmod 8 \\
Ø(i) &= (X(\daleth + 1).Ø_0 + 21.\mho(i)) \bmod F
\end{aligned} \tag{C.4}$$

C.3 RANDOM BLOCK INTERLEAVERS

A random interleaver can be described as a block interleaver with a mapping function generated from a permutation, based on the outputs of a random noise source [87]. The basic idea behind the design of random block interleavers is to eliminate regular patterns in $G_\pi(D)$, resulting in extremely long interleaver periods. Some of the more popular random interleavers, frequently used in concatenated coding schemes, are discussed in the following subsections.

C.3.1 PN GENERATOR INTERLEAVERS

The generation of the mapping function of this type of interleaver makes use of PN generators. Assuming a maximal length-a PN generator is employed, the interleaver will have a period of $N = 2^a - 1$, since the period of the sequence generated by the PN generator is $2^a - 1$. It is obvious that the interleaver period N will always be an odd number. The interleaver mapping function is determined as follows [87]:

1. Set $i = 0$.
2. At time index i, $\Pi(i)$ is the decimal equivalent of the a-bit binary word stored in the PN generator's shift register.
3. If $i < 2^a - 1$, increment i and return to step (2).

C.3.2 RANDOM NUMBER GENERATOR INTERLEAVERS

Although similar to PN generator interleavers, this type of interleaver's period N need not be an odd number. Using any type of uniform number generator, a period N random number generator interleaver's mapping function is determined as follows [87]:

1. Generate N random numbers. Store these numbers in a length-N vector, denoted by $\overline{U} = \{U_0, U_1, ..., U_{N-1}\}$.

2. Reorder the elements contained in $\overline{U}$ to range from the smallest to the largest value. The result is stored a length-N vector, denoted by $\overline{V} = \{V_0, V_1, ..., V_{N-1}\}$

3. For every i, with $0 \leq i < N$, determine the index j such that $U_i = V_j$. The mapping function of the interleaver is then simply $\Pi(i) = j$.

C.3.3 s-RANDOM INTERLEAVERS

In [177] a simple method to generate a random period-N interleaver that includes a constraint on the spreading factor M_{spread} (see *Section* 3.2.3.2) is presented. The process whereby such an interleaver's mapping function is created, is as follows:

1. Store the numbers 0 to $N-1$ in a length-N vector, denoted by $\overline{U} = \{U_0, U_1, ..., U_{N-1}\}$.

2. Choose an integer value for the spread factor M_{spread}. For a given value of N, it is important to choose $M_{spread} < \sqrt{\frac{N}{2}}$ in order for the interleaver construction method to be successful.

3. Set counter $i = 0$. Repeat the following steps:

   (a) Randomly pick a number from the list of available numbers contained in $\overline{U}$. If the chosen number differs by more than $\pm M_{spread}$ when compared to the previous M_{spread} values stored in $\overline{V}$, store it as element V_i in the vector $\overline{V} = \{V_0, V_2, ..., V_{N-1}\}$ and mark it as unavailable for the next random selection from $\overline{U}$. Otherwise, repeat step (a).

   (b) Increment counter i.

   (c) Repeat (a) to (b) for $0 \leq i < N$.

4. For every i, with $0 \leq i < N$, determine the index j such that $U_i = V_j$. As with random number generator interleavers, the mapping function of the interleaver is then $\Pi(i) = j$.

C.4 UNIFORM INTERLEAVERS

A concept frequently encountered in the derivation of concatenated coding scheme BER performance bounds, is that of a uniform interleaver [100, 101]. A size N uniform interleaver is a probabilistic device that maps any given input word $\overline{\mu}_m^{in}$ of Hamming weight w into all distinct $\binom{N}{w}$ permutations of the input word, each permutation having a probability of occurrence of:

$$\text{Prob.}\left(w_H\left(\overline{\mu}_m^{in}\right) = w\right) = 1/\binom{N}{w} \tag{C.5}$$

APPENDIX D

COMPLEX SPREADING SEQUENCES

D.1 APPENDIX OVERVIEW

THE application of binary sequences in DS/SSMA systems has been exhaustively investigated since introduction of SS. Due to the availability of potentially large sets of sequences that exhibit comparable auto-correlation and improved cross-correlation properties when compared to binary sequences, interest has started to shift towards the use of non-binary and CSSs. There are numerous advantages in using CSSs in future 4G DS/SSMA systems, including the possibility to generate CE and SSB [4, 7, 10] transmitter output signals, etc. This appendix not only summarises some of the important performances measures utilised in the analysis of CSSs, but also gives concise overviews of the filtered and unfiltered CSS families considered in this study.

D.2 IMPORTANT PERFORMANCE MEASURES FOR COMPLEX SPREADING SEQUENCES

D.2.1 SEQUENCE LENGTH AND FAMILY SIZE

The length of a CSS, denoted by M_{seq}, is the number of chips in a single CSS. It is a cardinal factor in the determination of a DS/SSMA system's processing gain (see *Section* D.2.4). Furthermore, it also influences the correlation characteristics of a CSS (see *Section* D.2.2 and *Section* D.2.3), which in turn is the factor determining a DS/SSMA system's capacity.

Inseparably intertwined with the sequence length, is the family size M_{fam} of a CSS. Usually a longer sequence length implies that more sequences, i.e. a larger family, can be generated that exhibit acceptable correlation properties.

D.2.2 PERIODIC AUTO-CORRELATION

Two types of auto-correlation functions can be calculated for spreading sequences, namely periodic and aperiodic. In synchronous DS/SSMA systems, such as the systems considered in this study, the former is of greater importance. The periodic auto-correlation of a continuous length-M_{seq} CSS, $S(t)$, having chips of duration T_{chip} [s], is defined as follows [43, 47, 48]:

$$R_{S(t),S(t)}(\tau) = \int_0^{M_{seq}.T_{chip}} S(t)S^*\left((t+\tau) \bmod (M_{seq}.T_{chip})\right) dt \qquad \text{(D.1)}$$

The periodic auto-correlation function gives an indication of the signal amplitude to be expected at the output of a coherent, perfectly synchronous correlator receiver [47]. As such, this function can also be used for synchronisation purposes in code tracking loops [43].

D.2.3 PERIODIC CROSS-CORRELATION

The periodic cross-correlation measures the periodic similarity between two different CSSs having a relative phase shift of τ seconds. It is defined as follows for the continuous length-M_{seq} CSSs $S_1(t)$ and $S_2(t)$, both consisting of M_{seq} chips of duration T_{chip} [s] [47,48]:

$$R_{S_1(t),S_2(t)}(\tau) = \int_0^{M_{seq}.T_{chip}} S_1(t) S_2^*\left((t+\tau) \bmod (M_{seq}.T_{chip})\right) dt \tag{D.2}$$

The periodic cross-correlation characteristics of the sequences in a CSS family dictate the degradation in performance in a multi-user DS/SSMA system due to MUI. Lower periodic cross-correlation values (especially at $\tau = 0$ in synchronous systems) deliver less MUI, resulting in better BER performances. Also, false code-lock is less probable for sequences with lower periodic cross-correlation values, especially in the range $|\tau| \leq \frac{M_{seq}.T_{chip}}{4}$, centered on $\tau = 0$.

A very popular lower bound on the periodic cross-correlation for the length-M_{seq} sequences $S_1(t)$ and $S_2(t)$ from a family of size-M_{fam}, is the *Welsh*-bound, given by [48]:

$$\max\left\{R_{S_1(t),S_2(t)}(\tau)\right\} \geq M_{seq}\sqrt{\frac{M_{fam}-1}{M_{seq}.M_{fam}-1}} \tag{D.3}$$

Note that for $M_{seq} \to \infty$, the Welsh-bound simplifies to $\max\left\{R_{S_1(t),S_2(t)}(\tau)\right\} \approx \sqrt{M_{seq}}$.

D.2.4 SPREADING FACTOR AND PROCESSING GAIN

Assume a spreading sequence with a chip rate of f_{chip} is used to directly spread a symbol stream with a rate of f_s. The *Spreading Factor* (SF) for this scenario is defined as follows [43,44]:

$$SF = \frac{f_{chip}}{f_s} \tag{D.4}$$

The PG (measured in [dB]) of a DS/SSMA system, which is directly related to its SF, is calculated as follows [43,44]:

$$PG = 10\log_{10}(SF) = 10\log_{10}\left(\frac{f_{chip}}{f_s}\right) \tag{D.5}$$

The SF and PG are important measures that reflect the spreading diversity introduced by the DS/SSMA system in order to combat the detrimental effects of narrowband interferers or jamming signals [43,44,81].

D.2.5 BANDWIDTH EXPANSION FACTOR

When comparing DS/SSMA systems employing filtered (or chip-level pulse shaped) and unfiltered CSSs, parameters such as PG and SF are insufficient, since these measures do not reflect the spectral characteristics of the spreading sequences and/or chip-level pulse shaping filters employed. As such, a new parameter, referred to as the BEF of a DS/SSMA system needs to be defined. The BEF for a DS/SSMA system with a transmitter output signal bandwidth of B_{sig}^{pre} prior to spreading and B_{sig}^{post}

after spreading (and chip-level pulse shaping), is calculated as follows:

$$BEF = \left(\frac{B_{sig}^{post}}{B_{sig}^{pre}} \right) \tag{D.6}$$

As with the SF and PG, the immunity a DS/SSMA system obtains against a narrowband jamming signal by increasing its transmitter output signal's dimensionality [43, 44, 81] through spreading, is reflected by the BEF. However, the BEF also includes the effects of chip-level pulse shaping and other DS/SSMA transmitter filtering on the bandwidth diversity obtained through the spreading process. Furthermore, the spectral characteristics of the spreading sequences used in the DS/SSMA system are also incorporated in the BEF.

D.2.6 SPREADING SEQUENCE LENGTH DIVERSITY

Certain pre-filtered CSS families (or chip-level pulse shaped CSSs), such as the ABC (see *Section* D.3.2.2) and DSB CE-LI-RU filtered GCL CSSs (see *Section* D.3.2.1), are highly bandlimited. Using these CSSs in DS/SSMA communication systems are more bandwidth efficient than using unfiltered CSSs or binary sequences of an equivalent length. As such, employing pre-filtered CSSs, equivalent SFs, but lower BEFs will be obtained. Generally, commercial communication systems are restricted in terms of their transmission bandwidth requirements. Thus, DS/SSMA communication systems employing pre-filtered CSSs support the use of higher data rates and/or longer CSSs in order to deliver permissable BEFs, when compared to systems using unfiltered CSSs. For fixed data rates and BEFs, the SSLD obtained by using bandlimited spreading sequences, is defined as follows:

$$SSLD = \frac{SF}{BEF} \tag{D.7}$$

Thus, the length of the filtered CSSs, denoted by M_{seq}, can be increased SSLD-times in order for the DS/SSMA communication system to occupy the same transmission bandwidth as a DS/SSMA system using unfiltered CSSs of the same length. Moreover, a larger pre-filtered CSS family can be used, possibly supporting more CDMA users.

D.3 IMPORTANT COMPLEX SPREADING SEQUENCE FAMILIES

D.3.1 UNFILTERED SEQUENCES

D.3.1.1 ZADOFF-CHU SEQUENCES

ZC CSSs is a subclass of GCL CSSs [9], which is generated and characterised as follows: Let $\overline{S}_{ZC}^{q} = \left\{ S_{ZC}^{q}[0], S_{ZC}^{q}[1], ..., S_{ZC}^{q}[M_{seq}-1] \right\}$ represent the vector of chips of the q^{th} length-M_{seq} unfiltered continuous-time ZC sequence $S_{ZC}^{q}(t)$. With $j = \sqrt{-1}$, the i^{th} chip in this sequence is determined as follows [4, 5]:

$$S_{ZC}^{q}[i] = \begin{cases} \exp\left(j \frac{\pi . a . i^2}{M_{seq}} \right) & \text{if } M_{seq} \text{ is even} \\ \exp\left(j \frac{\pi . a . i(i+1)}{M_{seq}} \right) & \text{if } M_{seq} \text{ is odd} \end{cases} \tag{D.8}$$

where the sequence number a can only take on integer values relatively prime to M_{seq}. As such, the family size for length-M_{seq} ZC CSSs is calculated as follows:

$$M_{fam} = 1 + \sum_{a=2}^{M_{seq}-1} \begin{cases} 1 & \text{if } M_{seq} \bmod(a) \neq 0 \\ 0 & \text{if } M_{seq} \bmod(a) = 0 \end{cases} \tag{D.9}$$

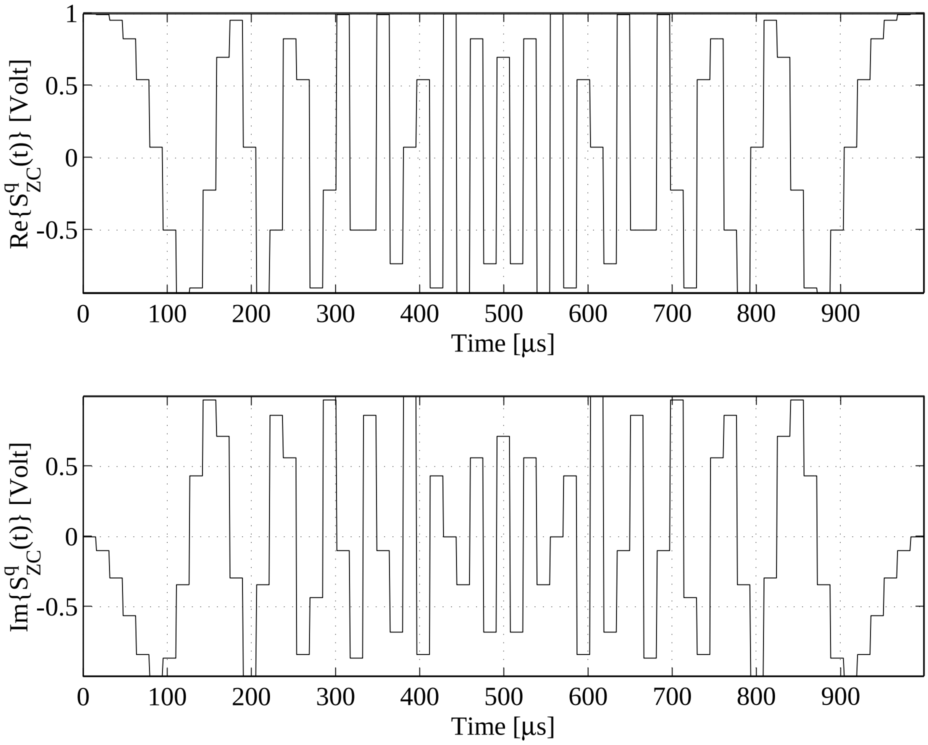

Figure D.1: Real and Imaginary Parts of a Length $M_{seq} = 63$ Unfiltered ZC CSS for $a = 1$, $f_{chip} = 63000$ Hz and 16 Samples per Chip

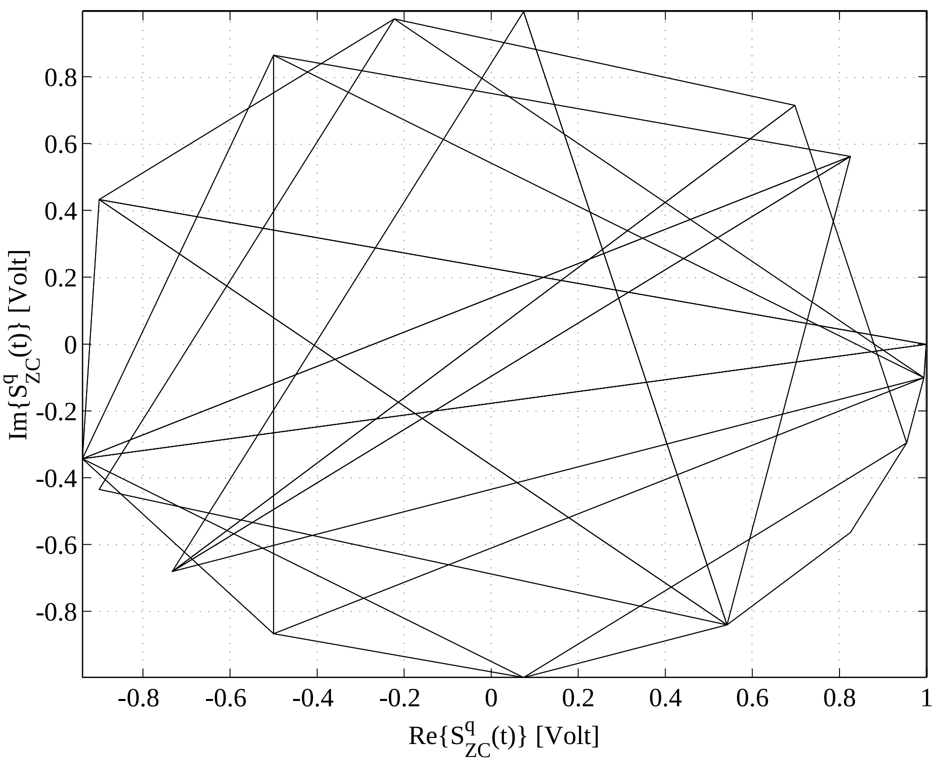

Figure D.2: Envelope of a Length $M_{seq} = 63$ Unfiltered ZC CSS for $a = 1$, $f_{chip} = 63000$ Hz and 16 Samples per Chip

Hence, the largest ZC CSS families are obtained when M_{seq} is an odd prime number. In such a case the family size is $M_{fam} = M_{seq} - 1$ [4]. *Fig.* D.1 shows the real and imaginary parts of the $a = 1$ length-63 ZC CSS with a chip rate of $f_{chip} = 63000$ Hz and 16 samples per chip. Also note that the complex envelope of the sequence, shown in *Fig.* D.2, is not constant.

D.3.1.2 QUADRIPHASE SEQUENCES

QPH sequences are closely related to binary sequences. A length-M_{seq} QPH sequence's chip vector $\overline{S}^q_{QPH} = \left\{S^q_{QPH}[0], S^q_{QPH}[1], ..., S^q_{QPH}[M_{seq}-1]\right\}$ is constructed using two length-M_{seq} binary sequences' chip vectors, denoted by $\overline{S}^q_A = \left\{S^q_A[0], S^q_A[1], ..., S^q_A[M_{seq}-1]\right\}$ and $\overline{S}^q_B = \left\{S^q_B[0], S^q_B[1], ..., S^q_B[M_{seq}-1]\right\}$, respectively. Calculation of the i^{th} chip of the QPH sequence is accomplished as follows [6]:

$$S^q_{QPH}[i] = \frac{1}{2\sqrt{2}}(1+j)S^q_A[i] + \frac{1}{2\sqrt{2}}(1-j)S^q_B[i] \tag{D.10}$$

It follows that each chip in the QPH sequence will have a value from the complex 4-symbol alphabet $\left\{\frac{1}{\sqrt{2}} + \frac{j}{\sqrt{2}}, \frac{1}{\sqrt{2}} - \frac{j}{\sqrt{2}}, -\frac{1}{\sqrt{2}} + \frac{j}{\sqrt{2}}, -\frac{1}{\sqrt{2}} - \frac{j}{\sqrt{2}}\right\}$, if the binary sequences' chip vectors $\overline{S}^q_A$ and $\overline{S}^q_B$ have chips from the antipodal alphabet $\{-1, +1\}$. For *Alltop*-type QPH sequences [4], the family size is given as $M_{fam} = M_{seq} - 1$, with the sequence length M_{seq} limited to prime values. Furthermore, using Gold binary sequences for $\overline{S}^q_A$ and $\overline{S}^q_B$ is a popular approach [48]. *Fig.* D.3 show the real and imaginary parts of user-q's length-63 QPH CSS with a chip rate of $f_{chip} = 63000$ Hz and 16 samples per chip. Since QPH sequences are binary in nature, it follows that their complex envelopes will not be constant. This characteristic is shown in *Fig.* D.4 for the QPH depicted in *Fig.* D.3.

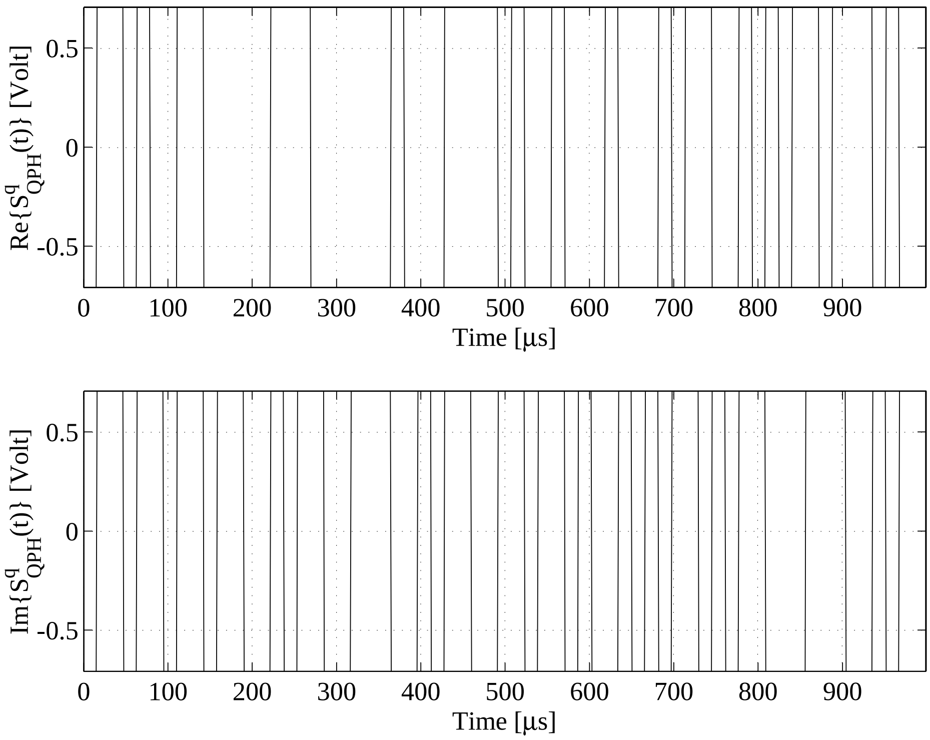

Figure D.3: Real and Imaginary Parts of a Length $M_{seq} = 63$ Unfiltered QPH CSS for $f_{chip} = 63000$ Hz and 16 Samples per Chip

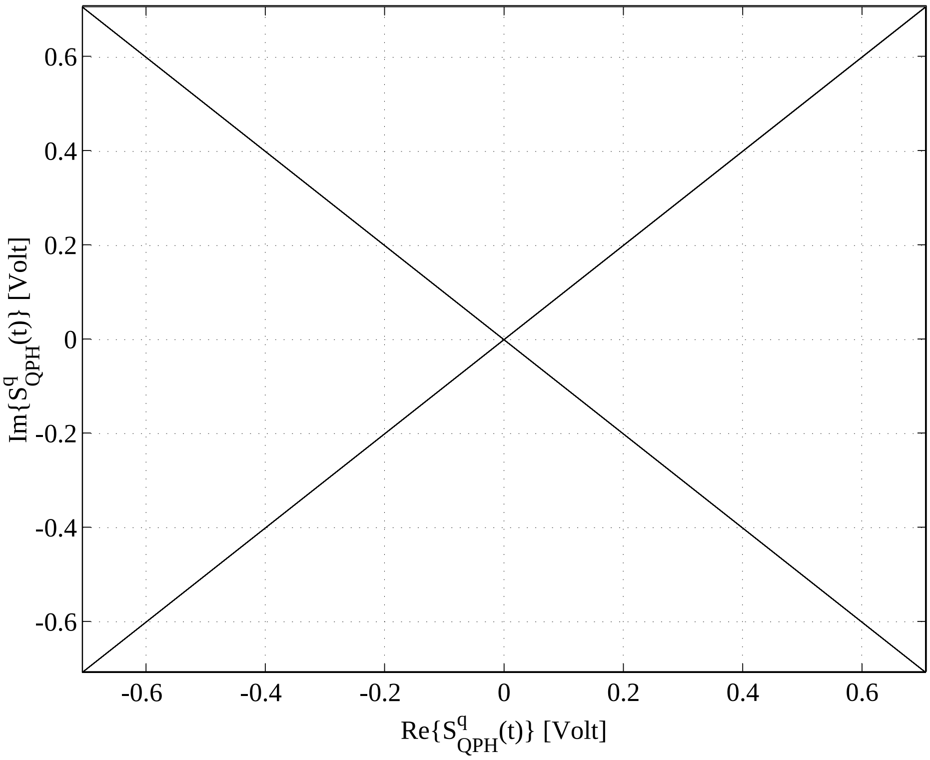

Figure D.4: Envelope of a Length $M_{seq} = 63$ Unfiltered QPH CSS for $f_{chip} = 63000$ Hz and 16 Samples per Chip

D.3.2 FILTERED SEQUENCES

D.3.2.1 DSB CE-LI-RU FILTERED GCL SEQUENCES

It has been shown that ZC sequences contain all the frequencies in the range $[0, M_{fam}/T_{chip})$ [Hz], with T_{chip} the duration of a chip [48]. Thus, the bandwidth of such sequences are a function of the family size. In order to bandlimit $S^q_{ZC}(t)$ and remove its dependency on the sequence index a, a $\mathrm{mod}(2\pi)$ phase constraint can be incorporated, resulting in a *Chu* sequence's chip vector denoted by $\overline{S}^q_{Chu} = \left\{S^q_{Chu}[0], S^q_{Chu}[1], ..., S^q_{Chu}[M_{seq}-1]\right\}$ [4,5]. The i^{th} chip of a *Chu* sequence is determined as follows:

$$S^q_{Chu}[i] = \begin{cases} \exp\left(j\frac{a.\pi.i^2}{M_{seq}}\right) \bmod (2\pi) & \text{if } M_{seq} \text{ is even} \\ \exp\left(j\frac{a.\pi.i.(i+1)}{M_{seq}}\right) \bmod (2\pi) & \text{if } M_{seq} \text{ is odd} \end{cases} \tag{D.11}$$

It has been shown [4] that the bandwidth of *Chu* sequences are $1/T_{chip}$ [Hz]. DSB CE-LI-RU filtered GCL sequences are obtained by filtering $S^q_{Chu}[i]$ with a *linearly interpolating root-of-unity filter* [7,8] in order to achieve the minimum Nyquist bandwidth of $1/(2.T_{chip})$ [Hz]. The family size of such sequences is also given by *Eq.* (D.9). *Fig.* D.5 gives the real and imaginary parts of user-q's length-63 DSB CE-LI-RU filtered GCL CSS with $a = 1$, a chip rate of $f_{chip} = 63000$ Hz and 16 samples per chip. *Fig.* D.6 shows the complex envelope of this sequence, depicting its constant nature. This characteristic alleviates and even eliminates the linearity constraint on power amplifiers used in DS/SSMA systems employing such sequences. Since the instantaneous power of the transmitter output signal will be constant, the communication system engineer no longer needs to be concerned with amplifier back-off. Thus, it will be possible to more efficiently utilise partially linear power amplifiers.

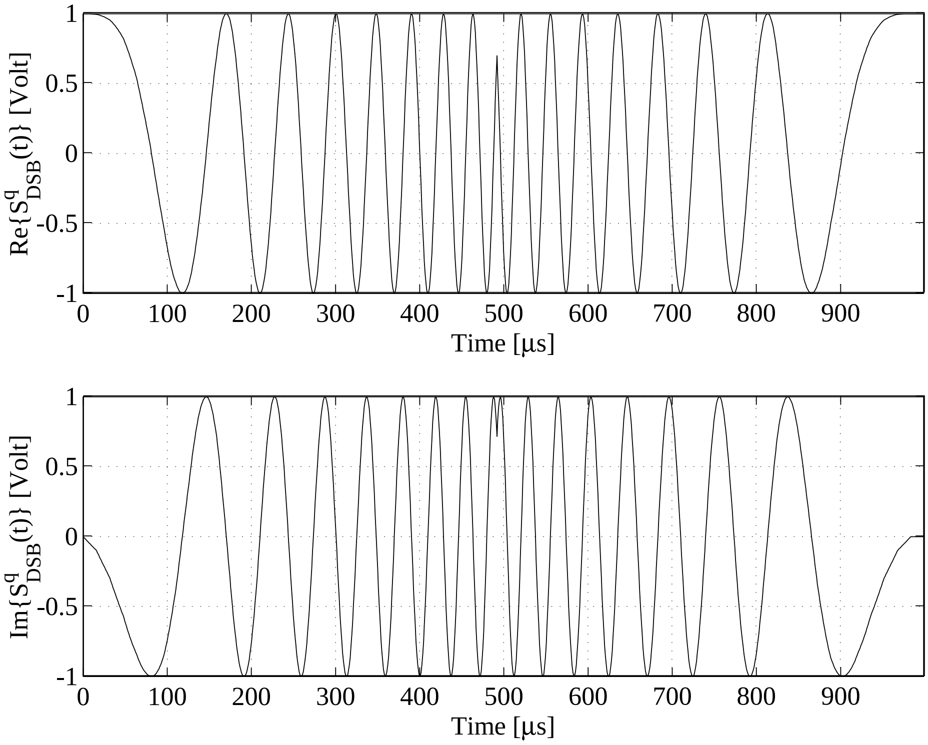

Figure D.5: Real and Imaginary Parts of a Length $M_{seq} = 63$ DSB CE-LI-RU filtered GCL CSS for $a = 1$, $f_{chip} = 63000$ Hz and 16 Samples per Chip

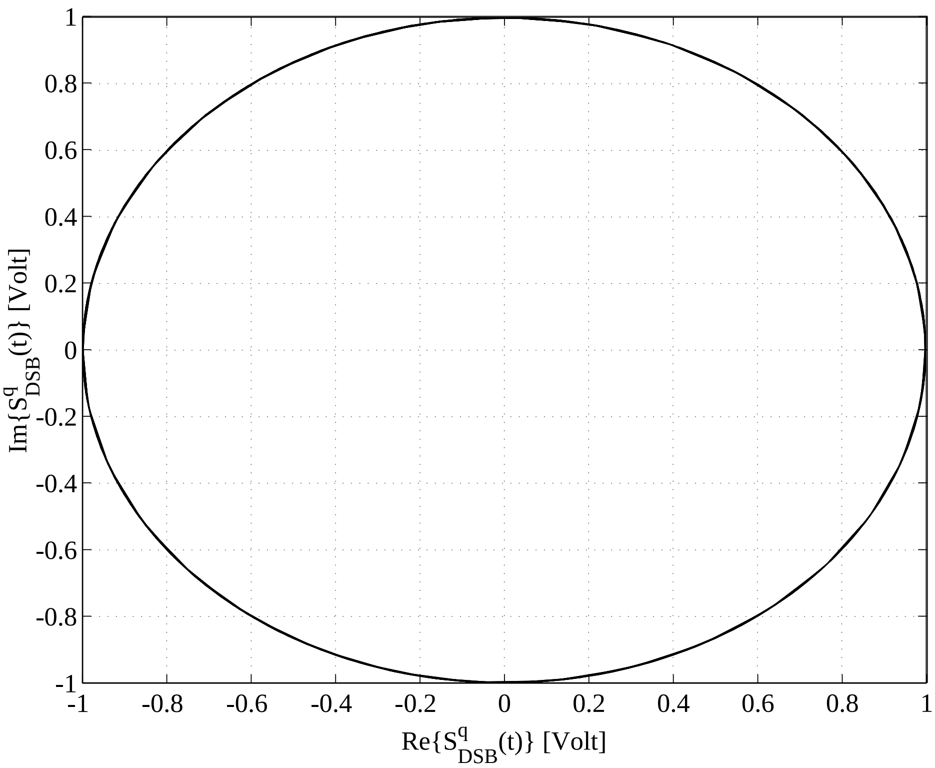

Figure D.6: Envelope of a Length $M_{seq} = 63$ DSB CE-LI-RU filtered GCL CSS for $a = 1$, $f_{chip} =$ 63000 Hz and 16 Samples per Chip

D.3.2.2 ABC SEQUENCES

ABC sequences are generated by appropriately modifying the previously defined DSB CE-LI-RU filtered GCL sequences in order to produce an injective function, as described in [7, 10]. When used in balanced QPSK structures, ABC sequences [7, 10] exhibit analytical properties, i.e. a SSB DS/SSMA signal is obtained after modulation onto in-phase and quadrature carriers (see *Fig.* 6.16 in *Section* 6.4.3) [4]. As with ZC and DSB CE-LI-RU filtered GCL sequences, the family size of ABC sequences is determined using *Eq.* (D.9). *Fig.* D.7 depicts the real and imaginary parts of user-q's length-63 ABC sequence with $a = 1$, a chip rate of $f_{chip} = 63000$ Hz and 16 samples per chip. *Fig.* D.8 shows its constant complex envelope.

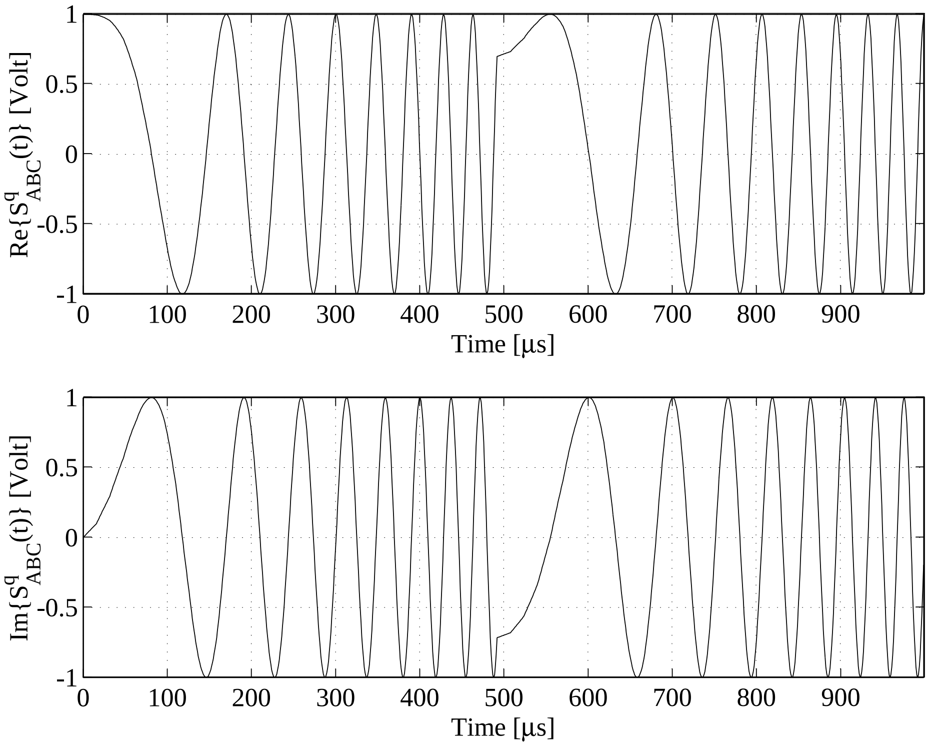

Figure D.7: Real and Imaginary Parts of a Length $M_{seq} = 63$ ABC Sequence for $a = 1$, $f_{chip} = 63000$ Hz and 16 Samples per Chip

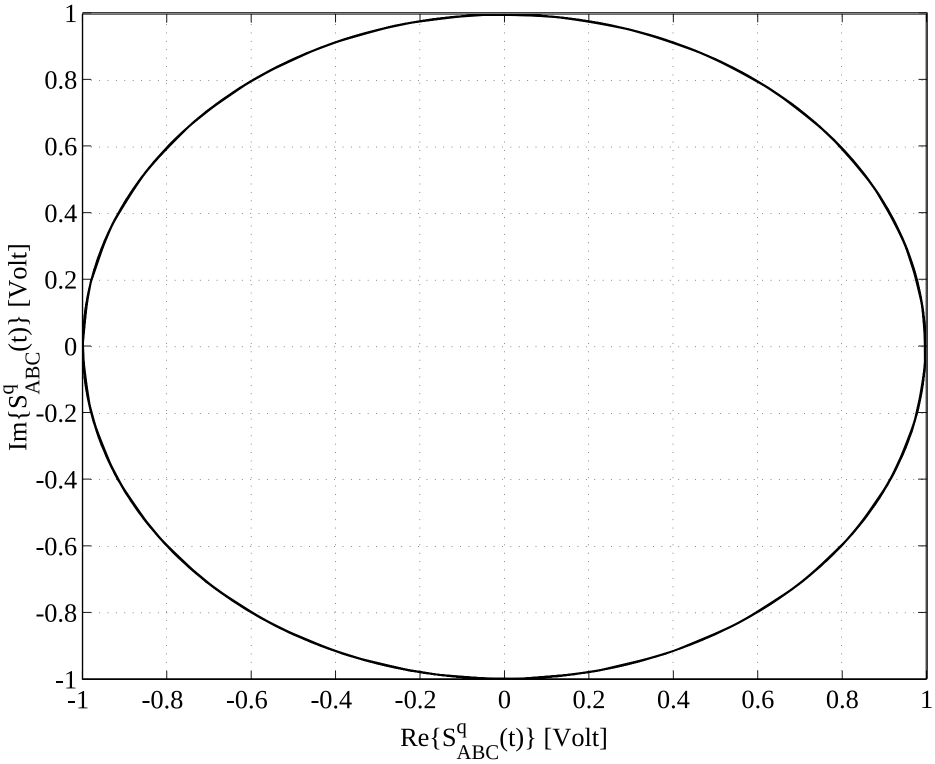

Figure D.8: Envelope of a Length $M_{seq} = 63$ ABC Sequence for $a = 1$, $f_{chip} = 63000$ Hz and 16 Samples per Chip

APPENDIX E

SIMULATION SOFTWARE INDEX

E.1 APPENDIX OVERVIEW

ACQUIRING the results presented in *Chapter* 6 required the development of an extensive set of simulation software tools, ranging from BER performance measurement platforms (see *Chapter* 5), to data analyses and plotting tools. C++ was chosen to implement the simulation platforms presented in *Chapter* 5, whereas Matlab was employed to create the necessary filter design and data analyses/plotting tools.

This appendix firstly presents the labelling conventions applied extensively in the filenames of the Matlab scripts/functions, C++ classes and compiled applications. Next follow detailed indexes, listing the filenames and short descriptions of the simulation software components developed during this study. All of the listed Matlab and C++ software modules are available on the CD-ROM accompanying this dissertation. Note that each of the simulation software tools presented here were created by the author without making use of any prior base code.

E.2 NAMING LABEL CONVENTIONS

The naming label conventions applied in the filenames of the Matlab m-files and compiled executables are given in *Table* E.1.

E.3 MATLAB FUNCTIONS AND SCRIPTS

Table E.2 and *Table* E.3 list the Matlab functions and scripts developed during the course of this study. Note that the Matlab functions and scripts were not employed in the actual performance evaluation of the VA decoded linear block codes under investigation, but rather for the creation of filters and pulse shapes, the processing and plotting of measured simulation results, etc. Although the results presented in *Chapter* 6 were obtained running these scripts and functions on a *Microsoft Windows* platform, they can be used on *Unix* or *Linux* platforms without any alterations.

Table E.1: Matlab Script and C++ Executable Filename Labelling Convention

Naming Label	**Options Available**	**Description**
CODENAME	half_rate_NSC / hr_NSC	4-state, rate $R_c = 1/2$ NSC code
	Hamming_7_4_Classic / Ham_7_4_cl	Hamming $(7, 4, 3)$, classic decoding
	Hamming_7_4_VA / Ham_7_4_cl_VA	Hamming $(7, 4, 3)$, VA decoding
	Interl_Hamming_7_4 / iHam_7_4	Interleaved Hamming $(7, 4, 3)$ code
	Interl_RS_7_5 / iRS_7_5	Interleaved RS $(7, 5, 3)$ code
	Original_5_3 / or_5_3	Cyclic $(5, 3, 2)$ code, original trellis
	Punct_BCH_15_7 / pBCH_15_7	Punctured BCH $(15, 7, 5)$ code
	Punct_half_rate_RSC / phr_RSC	Punctured, rate $R_c = 1/2$ RSC code
	Reduced_5_3 / or_red_5_3	Cyclic $(5, 3, 2)$ code, reduced trellis
	RS_7_5	RS $(7, 5, 3)$ block code
	two_thirds_rate_RSC / ttr_RSC	8-state, rate $R_c = 2/3$ RSC code
	uncoded / uc	Uncoded
SEQNAME	ABC	ABC sequences
	DSB	DSB CE-LI-RU GCL CSSs
	ZC	ZC CSSs
	QPH	QPH CSSs

Table E.2: Description of the Matlab Functions and Scripts - Part I

Matlab Function/Script	**Function/Script Description**
calc_pdf_1D.m	Calculate the 1-dimensional PDF of a set of samples
calc_pdf_2D.m	Calculate the 2-dimensional PDF of 2 sets of samples
create_elliptic_rx_filter.m	Create the numerator and denominator coefficients of an elliptic IIR lowpass filter, plot its amplitude response
create_nyquist_pulse_shape.m	Create a Nyquist pulse shape, plot the pulse shape
create_sqrt_nyquist_pulse_shape.m	Create a square-root Nyquist pulse shape, plot the pulse shape

Table E.3: Description of the Matlab Functions and Scripts - Part II

Matlab Function/Script	**Function/Script Description**
delay_spread.m	Configure the path delays for 10 unique users' multipath fading channels
doppler_filter.m	Create the numerator and denominator coefficients of a Doppler IIR lowpass filter, plot its amplitude response
plot_*CODENAME*_AWGN_ber.m	Plot the BER curves of a *CODENAME* code in AWGN channel conditions
plot_*CODENAME*_*SEQNAME*_ber.m	Plot the BER curves of a *CODENAME* code in multipath fading channel conditions for *SEQNAME* CSSs
plot_*CODENAME*_33Hz_FF_ber.m	Plot the BER curves of a *CODENAME* code in flat fading channel conditions with a 33 Hz Doppler spread
plot_*CODENAME*_100Hz_FF_ber.m	Plot the BER curves of a *CODENAME* code in flat fading channel conditions with a 100 Hz Doppler spread
plot_eye_diagram.m	Plot the eye diagrams of a pulse shaping or matched filter's output
plot_*SEQNAME*_envelope.m	Plot the complex envelope of a *SEQNAME* CSS
plot_*SEQNAME*_PSD.m	Plot the PSD of a *SEQNAME* CSS
plot_*SEQNAME*_time_signals.m	Plot the real and imaginary time signals of a *SEQNAME* CSS
power_delay_profile.m	Create an exponential decay power delay profile
process_sequence_family_files.m	Generation of the user CSS configuration files
random_interleaver.m	Create the interleaver mapping of a random interleaver
Rayleigh_PDF.m	Plot a theoretical Rayleigh PDF
Rician_PDF.m	Plot theoretical Rician PDFs
*SEQNAME*_generate	Generate and store a length-M_{seq} *SEQNAME* CSS family
systematic_cyclic_matrices.m	Convert an (n, k, d_{min}) linear block code's generator matrix to systematic form
classic_doppler_spectrum.m	Plot the classic Doppler spread PSD

E.4 C++ CLASSES

A large number of coding, modulation and channel simulator building blocks are required to construct the AWGN, flat fading and multipath fading channel performance platforms shown in *Fig.* 5.5, *Fig.* 5.8 and *Fig.* 5.9, respectively. The header and *.cpp* files of each C++ class that was created to realise the required performance evaluation platform building blocks are listed in *Table* E.4 and *Table* E.5, with their primary functions. Note that these files are *ANSI C* compliant, ensuring portability to *Microsoft Windows*, *Unix* or *Linux* platforms.

Table E.4: Description of the C++ Classes - Part I

C++ Class	**Class Description**
B_Trellis_Advanced.h B_Trellis_Advanced.cpp	BCJR trellis class for an (n, k, d_{min}) linear block code
B_Viterbi_Advanced.h B_Viterbi_Advanced.cpp	Block-wise VA decoder class that operates on a BCJR trellis
BC_with_Interleaver.h BC_with_Interleaver.cpp	(n, k, d_{min}) linear block code encoder and length-N interleaver combination class
BC_with_Puncturer.h BC_with_Puncturer.cpp	(n, k, d_{min}) linear block code encoder and puncturer combination class
Block_Coder.h Block_Coder.cpp	(n, k, d_{min}) linear block code encoder class
C_Trellis.h C_Trellis.cpp	Rate-n/k convolutional code trellis class
CC_with_Puncturer.h CC_with_Puncturer.cpp	Rate-n/k convolutional code encoder and puncturer combination class
Convolutional.cpp Convolutional.h	Rate-n/k convolutional code encoder class
De-interleaver_with_B_Viterbi.h De-interleaver_with_B_Viterbi.cpp	Length-N de-interleaver and block-wise VA block code decoder combination class
Delay_line.h Delay_line.cpp	General delay line class
De-puncturer_with_B_Viterbi.h De-puncturer_with_B_Viterbi.cpp	De-puncturer and block-wise VA block code decoder combination class
De-puncturer_with_SW_Viterbi.h De-puncturer_with_SW_Viterbi.cpp	De-puncturer and sliding window VA convolutional code decoder combination class

Table E.5: Description of the C++ Classes - Part II

C++ Class	**Class Description**
DSSS_QPSK_RAKE_Rx.h DSSS_QPSK_RAKE_Rx.cpp	Wideband classic and complex DS/SSMA QPSK RAKE receiver class
DSSS_QPSK_Tx.h DSSS_QPSK_Tx.cpp	Wideband classic and complex DS/SSMA QPSK transmitter class
FIR.h FIR.cpp	General FIR filter class
GF_Calculator.h GF_Calculator.cpp	Galois field mathematics calculator class
IIR.h IIR.cpp	General IIR filter class
Int_dump.h Int_dump.cpp	Integrate-and-dump circuit class
Interleaver.h Interleaver.cpp	General block interleaver class
Mapper.h Mapper.cpp	General input-to-output mapper class, used as a block code ML decoder
Multipath_Fading_Channel.h Multipath_Fading_Channel.cpp	Classic and complex multipath fading channel simulator class
Noise.h Noise.cpp	AWGN, uniform noise and Poisson noise generator class
PN_Gen.h PN_Gen.cpp	Length-N PN generator class
Puncturer.h Puncturer.cpp	Block or convolutional code puncturer class
QPSK_Rx.h QPSK_Rx.cpp	Narrowband classic and complex QPSK receiver class
QPSK_Tx.cpp QPSK_Tx.h	Narrowband classic and complex QPSK transmitter class
Rician_Channel.cpp Rician_Channel.h	Classic and complex *Clarke* flat fading channel simulator
SW_Viterbi_Conv.cpp SW_Viterbi_Conv.h	Sliding window VA convolutional code decoder class

E.5 COMPILED EXECUTABLES

Using the C++ classes listed in *Section* E.4, the compiled executable files, listed in *Table* E.6 and *Table* E.7, were created to obtain the simulation results presented in *Chapter* 6. The executables created to test the channel simulators, verify the operation of the narrowband and wideband communication systems, construct block and convolutional code trellises, and creating the mapping functions of random interleaver, were developed using *Borland C++ Builder* 6, since they contain *Microsoft Windows* graphic components, such as forms, buttons, dialog boxes, etc. As such, these executables are not portable to OS platforms other than *Microsoft Windows*.

A command line approach was used for the executables performing the actual BER performance evaluations. These executables were created using *Borland C++ Builder* 6, but compiled using *Intel*'s *ICC* and GNU's *G++* compilers for *Linux* platforms. The BER performance results presented in *Chapter* 6 were obtained in record breaking time by distributing the applications' computational load over the multiple workstations in the *University of Pretoria*'s *I-percube*, donated by *Intel*. The *I-percube* consists of 16 2.4 GHz *Pentium* 4 stations, each station running a *Mandrake Linux* operating system. The 16 stations are linked via *Fast Ethernet* connections. Process migration and message handling between the stations are managed transparently by means of *Open Mosix* for *Linux*.

Table E.6: Description of the Compiled Executables - Part I

Executable	Description/Function	Portability
BC_Trellis_Creator	Creation of an (n, k, d_{min}) linear block code's BCJR trellis	Windows
CC_Trellis_Creator	Creation of a rate-n/k convolutional code's trellis	Windows
Interleaver_Creator	Creation of a random or classic block interleaver	Windows
main_*CODENAME*_QPSK_AWGN	BER performance evaluation of a *CODENAME* code in AWGN channel conditions	Windows Linux DOS
main_*CODENAME*_QPSK_FF	BER performance evaluation of a *CODENAME* code in flat fading channel conditions	Windows Linux DOS
main_*CODENAME*_DSSSMA_QPSK_MPFC	BER performance evaluation of a *CODENAME* code in multipath fading channel conditions	Windows Linux DOS

Table E.7: Description of the Compiled Executables - Part II

Executable	**Description/Function**	**Portability**
Test_flat_fading_channel	Verify the operation of classic and complex flat fading channel simulators	Windows
Test_frequency_selective_fading_channel	Verify the operation of classic and complex multipath fading channel simulators	Windows
Test_narrowband_QPSK	Verify the operation of a narrowband classic or complex QPSK communication system	Windows
Wideband_Correlator	Verify the operation of a classic or complex DS/SSMA QPSK communication system employing a correlator RAKE receiver	Windows
Wideband_Matched_Filter	Verify the operation of a classic or complex DS/SSMA QPSK communication system employing a matched filter RAKE receiver	Windows

REFERENCES

[1] C. Shannon, “A Mathematical Theory of Communications,” *Bell Systems Technical Journal*, vol. 27, October 1948.

[2] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate,” *IEEE Transactions on Information Theory*, pp. 284–287, March 1974.

[3] J. Wolf, “Efficient Maximum Likelihood Decoding of Linear Block Codes Using a Trellis,” *IEEE Transactions on Information Theory*, vol. IT-24, pp. 76–80, January 1978.

[4] M. Jamil, L. P. Linde, J. E. Cilliers, and D. J. van Wyk, “Comparison of Complex Spreading Sequences Based on Filtering Methods and Mean Square Correlation Characteristics,” *Transactions of the SAIEE*, vol. 89, no. 3, pp. 98–112, September 1998.

[5] R. L. Frank and S. A. Zadoff, “Phase Shift Pulse Codes with Good Periodic Correlation Properties,” *IRE Transactions on Information Theory*, vol. IT-7, pp. 381–382, October 1962.

[6] S. M. Korne and D. V. Sarwate, “Quadriphase Sequences for Spread Spectrum Multiple Access Communication,” *IEEE Transactions on Information Theory*, vol. IT-38, no. 3, pp. 1101–1113, May 1992.

[7] M. P. Lötter and L. P. Linde, “Constant Envelope Filtering of Complex Spreading Sequences,” *IEE Electronics Letters*, vol. 31, no. 17, pp. 1406–1407, 17 August 1995.

[8] M. P. Lötter, “A Generalised Linear Root-of-Unity Interpolation Filter,” in *Proc., IEEE COMSIG’95*, (University of Pretoria, Pretoria, South Africa), pp. 43–46, September 1995.

[9] B. M. Popović, “Generalized Chirp-like Polyphase Sequences with Optimum Correlation Properties,” *IEEE Transactions on Information Theory*, vol. 38, July 1992.

[10] M. P. Lötter and L. P. Linde, “A Class of Bandlimited Complex Spreading Sequences with Analytic Properties,” in *Proc., IEEE Int. Symp. on Spread Spectrum Techniques and Applications*, (Mainz, Germany), pp. 662–666, 22-25 September 1996.

[11] G. D. Forney, Jr., “Concatenated Codes.” Cambridge, MA: MIT Press, 1966.

[12] G. Ungerboeck, “Channel Coding with Multilevel/Phase Signaling,” *IEEE Transactions on Information Theory*, vol. IT-25, pp. 55–67, January 1982.

[13] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes,” in *Proc., IEEE Int. Conf. on Communications*, (Geneva, Switzerland), pp. 1064–1070, 23-26 May 1993.

[14] I. S. Reed and G. Solomon, “Polynomial Codes Over Certain Finite Fields,” *SIAM Journal*, vol. 8, pp. 300–304, June 1960.

[15] S. Benedetto and G. Montorsi, "Unveiling Turbo Codes: Some Results on Parallel Concatenated Coding Schemes," *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 409–429, March 1996.

[16] L. Papke and P. Robertson, "Improved Decoding With SOVA in a Parallel Concatenated (Turbo-Code) Scheme," in *Proc., IEEE Int. Conf. on Communications*, (Dallas, TX, USA), pp. 102–106, 23-27 June 1996.

[17] P. Robertson, "Illuminating the Structure of Code and Decoder of Parallel Concatenated Recursive Systematic (Turbo) Codes," in *Proc., IEEE GLOBECOM*, (San Francisco, CA, USA), pp. 1298–1303, 28 November - 2 December 1994.

[18] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Bandwidth Efficient Parallel Concatenated Coding Scheme," *IEE Electronics Letters*, vol. 31, no. 24, pp. 2067–2069, November 1995.

[19] S. Benedetto and G. Montorsi, "Design of Parallel Concatenated Convolutional Codes," *IEEE Transactions on Communications*, vol. 44, no. 5, pp. 591–600, May 1996.

[20] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Parallel Concatenated Trellis Coded Modulation," in *Proc., IEEE Int. Conf. on Communications*, (Dallas, TX, USA), pp. 974–978, 23-27 June 1996.

[21] S. Benedetto and G. Montorsi, "Performance of Continuous and Blockwise Decoded Turbo Codes," *IEEE Communications Letters*, vol. 1, no. 3, pp. 77–79, May 1997.

[22] J. Anderson, "Turbo Coding for Deep Space Applications," in *Proc., IEEE Int. Symp. on Inform. Theory*, (Whistler, British Columbia, Canada), p. 36, 17-22 September 1995.

[23] A. S. Barbulescu and S. S. Pietrobon, "Rate Compatible Turbo-Codes," *IEE Electronics Letters*, vol. 31, pp. 535–536, March 1995.

[24] C. Berrou and M. Jezequel, "Frame-Oriented Convolutional Turbo Codes," *IEE Electronics Letters*, vol. 32, no. 15, pp. 1362–1363, July 1996.

[25] S. Benedetto, G. Montorsi, D. Divsalar, and F. Pollara, "Algorithm for Continuous Decoding of Turbo-Codes," *IEE Electronics Letters*, vol. 32, no. 4, pp. 314–315, February 1996.

[26] W. J. Blackert and S. G. Wilson, "Turbo Trellis Coded Modulation," in *Proc., Conf. on Inform. Systems and Sciences*, (Princeton, NJ, USA), 20-22 March 1996.

[27] S. Benedetto, G. Montorsi, D. Divsalar, and F. Pollara, "Serial Concatenation of Interleaved Codes: Performance Analysis, Design and Iterative Decoding," TDA Progress Report 42-126, JPL, (Pasadena, CA, USA), August 1996.

[28] S. Benedetto and G. Montorsi, "Iterative Decoding of Serially Concatenated Convolutional Codes," *IEE Electronics Letters*, vol. 32, no. 13, pp. 1186–1187, June 1996.

[29] S. Benedetto and G. Montorsi, "Serial Concatenation of Block and Convolutional Codes," *IEE Electronics Letters*, vol. 32, no. 10, pp. 887–888, May 1996.

[30] D. Divsalar and F. Pollara, "Serial and Hybrid Concatenation Codes with Applications," in *Proc., Int. Symp. on Turbo Codes and Related Topics*, (Brest, France), pp. 80–87, 3-5 September 1997.

[31] P. K. Gray, *Serial Concatenated Trellis Coded Modulation*. PhD Thesis, The University of South Australia, March 1999.

[32] P. Adde, R. Pyndiah, and C. Berrou, "Performance of Hybrid Turbo Codes," *IEE Electronics Letters*, vol. 32, no. 24, pp. 2209–2210, November 1996.

[33] D. Divsalar and F. Pollara, "Hybrid Concatenated Codes and Iterative Decoding," TDA Progress Report 42-130, JPL, (Pasadena, CA, USA), August 1997.

[34] R. G. Gallager, "Low-Density Parity-Check Codes," *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, January 1962.

[35] M. Alouini and A. J. Goldsmith, "A Unified Approach for Calculating Error Rates of Linearly Modulated Signals over Generalized Fading Channels," *IEEE Transactions on Communications*, vol. 47, no. 9, pp. 1324–1334, September 1999.

[36] M. K. Simon and M. Alouini, "A Unified Approach to the Performance Analysis of Digital Communication over Generalized Fading Channels," in *Proceedings of the IEEE*, vol. 86, no. 9, September 1998.

[37] B. Sklar, "Rayleigh Fading Channels in Mobile Digital Communication Systems Part I: Characterization," *IEEE Communications Magazine*, pp. 90–100, July 1997.

[38] J. S. Swarts, "Aspects of Multipath Channel Characterization," Master's Thesis, Rand Academic University, November 1998.

[39] A. K. Salkintzis, "Implementation of a Digital Wide-Band Mobile Channel Simulator," *IEEE Transactions on Broadcasting*, vol. 45, no. 1, pp. 122–128, March 1999.

[40] L. E. Vogler and J. A. Hoffmeyer, "A Model for Wideband HF Propagation Channels," *Radio Science*, vol. 28, no. 6, pp. 1131–1142, November-December 1993.

[41] C. C. Watterson, J. R. Juroshek, and W. D. Bensema, "Experimental Confirmation of an HF Channel Model," *IEEE Transactions on Communication Technology*, vol. COM-18, no. 6, pp. 792–803, December 1970.

[42] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Upper Saddle River, NJ, USA: Prentice Hall, First ed., 1996.

[43] R. L. Peterson, R. E. Ziemer, and D. E. Borth, *Introduction to Spread Spectrum Communications*. Upper Saddle River, NJ, USA: Prentice Hall, 1995.

[44] K. Feher, *Wireless Digital Communications - Modulation and Spread-Spectrum Applications*. Upper Saddle River, NJ, USA: Prentice-Hall, First ed., 1995.

[45] F. Adachi, M. Sawahashi, and H. Suda, "Wideband DS-CDMA for Next-Generation Mobile Communications Systems," *IEEE Communications Magazine*, pp. 56–69, September 1998.

[46] IS-95 Interim Standard: *'An Overview of the Application of Code Division Multiple Access (CDMA) to Digital Cellular Systems and Personal Cellular Networks'*, Qualcomm Inc., May 1992.

[47] J. G. Proakis, *Digital Communications*. New York, NY, USA: McGraw-Hill, Third ed., 1995.

[48] M. Jamil, "Comparative Study of Complex Spreading Sequences for CDMA Applications," Master's Thesis, University of Pretoria, May 1999.

[49] T. Li, M. Hatori, and N. Suehiro, “Analysis of Asynchronous Direct-Sequence Spread-Spectrum Multiple-Access Communications with Polyphase Sequences,” in *Proc., IEEE GLOBECOM*, (Phoenix, AZ, USA), 4-8 November 1997.

[50] L. Staphorst, M. Jamil, and L. P. Linde, “Performance of a Synchronous Balanced QPSK CDMA System Using Complex Spreading Sequences in AWGN,” in *Proc., IEEE AFRICON’99*, (Cape Town, South Africa), pp. 215–220, 28 September - 1 October 1999.

[51] L. Staphorst, M. Jamil, and L. P. Linde, “Performance Evaluation of a QPSK System Employing Complex Spreading Sequences in a Fading Environment,” in *Proc., IEEE VTC’99-Fall*, (Asterdam, The Netherlands), pp. 2964–2967, 19-22 September 1999.

[52] W. H. Büttner, L. Staphorst, and L. P. Linde, “Trellis Decoding of Linear Block Codes,” in *Proc., IEEE COMSIG’98*, (Cape Town, South Africa), pp. 171–174, 7-8 September 1998.

[53] L. Staphorst and L. P. Linde, “Performance Evaluation of Viterbi Decoded Binary and Non-binary Linear Block Codes in Flat Fading Channel Conditions,” in *Proc., IEEE AFRICON 2002*, (George, South Africa), pp. 181–186, 2-4 October 2002.

[54] L. Staphorst and L. P. Linde, “On theViterbi Decoding of Linear Block Codes,” *Transactions of the SAIEE*, vol. 94, no. 4, pp. 28–41, December 2003.

[55] L. Staphorst and L. P. Linde, “Performance Evaluation of Viterbi Decoded Reed-Solomon Block Codes in Additive White Gaussian Noise and Flat Fading Channel Conditions,” in *Proc., IEEE WCNC’2002*, (Orlando, Florida, USA), 17-21 March 2002.

[56] D. Chase, “A Class of Algorithms for Decoding Block Codes with Channel Measurement Information,” *IEEE Transactions on Information Theory*, vol. IT-18, pp. 170–181, July 1972.

[57] H. T. Moorthy, S. Lin, and T. Kasami, “Soft-Decision Decoding of Binary Linear Block Codes Based on an Iterative Search Algorithm,” *IEEE Transactions on Information Theory*, vol. 34, pp. 1049–1053, 1988.

[58] S. Lin, T. Kasami, T. Fujiwara, and M. Fossorier, *Trellises and Trellis-Based Decoding Algorithms for Block Codes*. Boston, MA, USA: Kluwer Academic Publishers, First ed., 1998.

[59] R. J. McEliece, “On the BJCR Trellis for Linear Block Codes,” *IEEE Transactions on Information Theory*, vol. 42, pp. 1072–1092, 1996.

[60] J. Hagenauer, “Viterbi Decoding of Convolutional Codes for Fading- and Burst- Channels,” in *Proc., Int. Zurich Seminar on Digital Communications*, (Zurich, Switzerland), pp. G2.1–G2.7, 1980.

[61] J. Hagenauer and P. Hoeher, “A Viterbi Algorithm with Soft-Decision Outputs and its Applications,” in *Proc., IEEE GLOBECOM*, (Dallas, TX, USA), pp. 47.1.1–47.1.7, 27-30 November 1989.

[62] H. Nickel, J. Hagenauer, and F. Burkert, “Approaching Shannon’s Capacity Limit by 0.27 dB Using Simply Hamming Codes,” *IEEE Communications Letters*, November 1997.

[63] S. Benedetto and G. Montorsi, “Average Performance of Parallel Concatenated Block Codes,” *IEE Electronics Letters*, vol. 31, no. 3, pp. 156–157, February 1995.

[64] O. Aitsab and R. Pyndiah, “Performance of Reed-Solomon Block Turbo Code,” in *Proc., IEEE GLOBECOM*, (London, UK), pp. 121–125, 18-22 November 1996.

[65] D. Divsalar and F. Pollara, "Turbo Codes for PCS Applications," in *Proc., IEEE Int. Conf. on Communications*, (Seatle, WA, USA), pp. 54–59, 18-22 June 1995.

[66] S. Benedetto and G. Montorsi, "Performance Evaluation of Parallel Concatenated Codes," in *Proc., IEEE GLOBECOM*, (Singapore, Malaysia), pp. 2273–2277, 13-17 November 1995.

[67] G. Battail, C. Berrou, and A. Glavieux, "Pseudo-Random Recursive Convolutional Coding for Near-Capacity Performance," in *Proc., IEEE Int. Conf. on Communications*, (Geneva, Switzerland), pp. 23–27, 23-26 May 1993.

[68] S. Benedetto and G. Montorsi, "Performance Evaluation of Turbo-Codes," *IEE Electronics Letters*, vol. 31, no. 3, pp. 163–165, February 1995.

[69] E. K. Hall, "Performance and Design of Turbo Codes on Rayleigh Fading Channels," Master's Thesis, University of Virginia, May 1996.

[70] G. E. Moore, "Cramming More Components onto Integrated Circuits," *Electronics*, vol. 38, no. 8, pp. 119 – 122, 19 April 1965.

[71] L. Staphorst, J. Schoeman, and L. P. Linde, "Performance Evaluation of a Joint Source/channel Coding Scheme for DS/SSMA Systems Utilizing Complex Spreading Sequences in Multipath Fading Channel Conditions," in *Proc., IEEE CCECE 2003*, (Montreal, Quebec, Canada), pp. 1679–1682, 4-7 May 2003.

[72] L. Staphorst and L. P. Linde, "Evaluating Viterbi Decoded Reed-Solomon Block Codes on a Complex Spreaded DS/SSMA CDMA System: Part I - Background and Communication System Models," in *Proc., IEEE AFRICON 2004*, (Gabarone, Botswana), pp. 329–334, 15-17 September 2004.

[73] L. Staphorst and L. P. Linde, "Evaluating Viterbi Decoded Reed-Solomon Block Codes on a Complex Spreaded DS/SSMA CDMA System: Part II - Channel Model, Evaluation Platform and Results," in *Proc., IEEE AFRICON 2004*, (Gabarone, Botswana), pp. 335–340, September 2004.

[74] J. L. Massey, "Step-by-Step Decoding of the BCH Codes," *IEEE Transactions on Information Theory*, vol. IT-11, pp. 580–585, October 1965.

[75] J. L. Massey, "Shift-Register Synthesis and BCH Decoding," *IEEE Transactions on Information Theory*, vol. IT-15, January 1969.

[76] R. H. Clarke, "A Statistical Theory of Mobile-Radio Reception," *Bell Systems Technical Journal*, vol. 47, pp. 957–1000, 1968.

[77] M. Lecours, M. Têtu, A. Chefaoui, J. Ahern, and A. Michaud, "Phase Measurements and Characterization of Mobile Radio Channels," *IEEE Transactions on Vehicular Technology*, vol. 45, no. 1, pp. 105–113, February 1996.

[78] G. Marsagalia and T. A. Bray, "A Convenient Method for Generating Normal Variables," *SIAM Rev.*, vol. 6, pp. 260–264, 1964.

[79] R. F. W. Coates, G. J. Janacek, and K. V. Lever, "Monte Carlo Simulation and Random Number Generation," *IEEE Journal on Selected Areas of Communication*, vol. 6, no. 1, pp. 58–66, January 1988.

[80] B. Wichmann and D. Hill, "Building a Random-Number Generator," *Byte Magazine*, pp. 127–128, March 1987.

[81] S. Haykin, *Communications Systems*. New York, NY, USA: John Wiley and Sons, Third ed., 1994.

[82] W. C. Jakes, Jr., *Microwave Mobile Communications*. IEEE Press, 1974.

[83] S. O. Rice, "Mathematical Analysis of Random Noise," *Bell Systems Technical Journal*, vol. 24, pp. 46–156, January 1945.

[84] S. O. Rice, "Statistical Properties of a Sine Wave Plus Random Noise," *Bell Systems Technical Journal*, vol. 27, pp. 109–157, January 1948.

[85] M. J. Gans, "A Power Spectral Theory of Propagation in the Mobile Radio Environment," *IEEE Transactions on Vehicular Technology*, vol. VT-21, pp. 27–38, February 1972.

[86] R. E. Ziemer, B. R. Vojcic, L. B. Milstein, and J. G. Proakis, "Effects of Carrier Tracking in RAKE Reception of Wide-Band DSSS in Rician Fading," *IEEE Transactions on Microwave Theory and Techniques*, vol. 47, no. 6, pp. 681–686, August 1999.

[87] C. Heegard and S. B. Wicker, *Turbo Coding*. Norwell, MA, USA: Kluwer Academic Publishers, First ed., 1999.

[88] S. Benedetto, R. Garello, and G. Montorsi, "A Search for Good Convolutional Codes to be Used in the Construction of Turbo Codes," *IEEE Transactions on Communications*, vol. 46, no. 9, pp. 1101–1105, September 1998.

[89] C. Berrou and A. Glavieux, "Near Optimum Error Correcting Coding and Decoding: Turbo Codes," *IEEE Transactions on Communications*, vol. 44, no. 10, pp. 1261–1271, October 1996.

[90] D. Divsalar and R. J. McEliece, "Effective Free Distance of Turbo Codes," *IEE Electronics Letters*, vol. 32, pp. 445–446, February 1996.

[91] B. Sklar, "A Primer on Turbo Code Concepts," *IEEE Communications Magazine*, pp. 94–102, December 1997.

[92] E. K. Hall and S. G. Wilson, "Design and Performance Analysis of Turbo Codes on Rayleigh Fading Channels," in *Proc., Conf. on Inform. Systems and Sciences*, (Princeton, NJ, USA), 20-22 March 1996.

[93] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*, vol. 16. New York, NY, USA: North-Holland Mathematical Library, First ed., 1977.

[94] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ, USA: Prentice-Hall, First ed., 1983.

[95] R. W. Hamming, "Error Detecting and Error Correcting Codes," *Bell Systems Technical Journal*, vol. 29, pp. 147–160, 1950.

[96] A. Hocquenghem, "Codes Correcteurs d'Erreurs," *Chiffres*, vol. 2, pp. 147–156, 1959.

[97] R. C. Bose and D. K. Ray-Chaudhuri, "On a Class of Error Correcting Binary Group Codes," *Information and Control*, vol. 3, pp. 68–79, March 1960.

[98] R. C. Bose and D. K. Ray-Chaudhuri, "Further Results in Error Correcting Binary Group Codes," *Information and Control*, vol. 3, pp. 279–290, September 1960.

[99] W. W. Pieterson and E. J. Weldon, Jr., *Error-Correcting Codes*. MIT Press, Second ed., 1972.

[100] D. Divsalar and F. Pollara, "On the Design of Turbo Codes," TDA Progress Report 42-123, JPL, (Pasadena, CA, USA), November 1995.

[101] D. Divsalar, S. Dolinar, R. J. McEliece, and F. Pollara, "Transfer Function Bounds on the Performance of Turbo Codes," TDA Progress Report 42-122, JPL, (Pasadena, CA, USA), August 1995.

[102] J. B. Cain, G. C. Clark, Jr., and J. M. Geist, "Punctured Convolutional Codes of Rate (n-1)/n and Simplified Maximum Likelihood Decoding," *IEEE Transactions on Information Theory*, vol. IT-25, no. 1, pp. 97–100, January 1979.

[103] Y. Yasuda, K. Kashiki, and Y. Hirata, "High-Rate Punctured Convolutional Codes for Soft Decision Viterbi Decoding," *IEEE Transactions on Communications*, vol. COM-32, no. 3, pp. 315–319, March 1984.

[104] L. H. C. Lee, "New Rate-Compatible Punctured Convolutional Codes for Viterbi Decoding," *IEEE Transactions on Communications*, vol. 42, no. 12, pp. 3073–3079, December 1994.

[105] M. Öberg and P. H. Siegel, "The Effect of Puncturing in Turbo Encoders," in *Proc., Int. Symp. on Turbo Codes and Related Topics*, (Brest, France), pp. 184–187, 3-5 September 1997.

[106] G. D. Forney, Jr., "The Viterbi Algorithm," in *Proc. of the IEEE*, vol. 61, no. 3, pp. 268–277, March 1973.

[107] R. M. Fano, "A Heuristic Discussion of Probabilistic Coding," *IEEE Transactions on Information Theory*, vol. IT-9, pp. 64–74, April 1963.

[108] M. Jordan and R. Michols, "The Effects of Channel Characterization on Turbo Code Performance," in *Proc., IEEE MILCOM*, (McLean, VA, USA), pp. 17–21, 21-24 October 1996.

[109] D. Hatzinakos and C. Nikias, "Estimation of Multipath Channel Impulse Response in Frequency Selective Channels," *IEEE Journal on Selected Areas of Communication*, vol. S7, no. 1, pp. 12–19, January 1989.

[110] D. I. Laurenson and G. J. R. Povey, "Channel Modelling for a Predictive RAKE Receiver System," in *Proc., IEEE Int. Symp. on Personal, Indoor and Mobile Radio Communications (PIMRC)*, (The Hague, The Netherlands), pp. 715–719, 21-23 September 1994.

[111] M. Missiroli, Y. J. Guo, and S. K. Barton, "Near-Far Resistant Channel Estimation for CDMA Systems Using the Linear Decorrelating Detector," *IEEE Transactions on Communications*, vol. 48, no. 3, pp. 514–524, March 2000.

[112] M. Pätzold, *Mobile Fading Channels:Modelling, Analysis, and Simulation*. Chichester, West Sussex, UK: John Wiley and Sons, First ed., 2002.

[113] P. van Rooyen, "Using MEM to Describe CDMA Multiple Access Interference," *Transactions of the SAIEE*, vol. 89, no. 3, pp. 90–97, September 1998.

[114] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ, USA: Prentice-Hall, Second ed., 1991.

[115] E. Baccarelli and R. Cusani, "Recursive Kalman-Type Optimal Estimation and Detection of Hidden Markov Chains," *Signal Processing EURASIP*, vol. 51, no. 1, pp. 55–64, May 1996.

[116] J. K. Cavers, "An Analysis of Pilot Symbol Assisted Modulation for Rayleigh Fading Channels," *IEEE Transactions on Vehicular Technology*, vol. 40, no. 4, pp. 686–693, November 1991.

[117] A. N. D'Andrea, A. Diglio, and U. Mengali, "Symbol-Aided Channel Estimaton with Nonselective Rayleigh Fading Channels," *IEEE Transactions on Vehicular Technology*, vol. 44, no. 1, pp. 41–48, February 1995.

[118] X. Tang, M. Alouini, and A. J. Goldsmith, "Effect of Channel Estimation Error on M-QAM BER Performance in Rayleigh Fading," *IEEE Transactions on Communications*, vol. 47, no. 12, pp. 1856–1864, December 1999.

[119] K. M. Chugg and A. Polydoros, "MLSE for an Unknown Channel - Part II: Tracking Performance," *IEEE Transactions on Communications*, vol. 44, no. 8, pp. 949–958, August 1996.

[120] R. Raheli, A. Polydoros, and C. Tzou, "Per-Survivor Processing: A General Approach to MLSE in Uncertain Environments," *IEEE Transactions on Communications*, vol. 43, no. 2-4, pp. 354–364, February-April 1995.

[121] G. Auer, G. J. R. Povey, and D. I. Laurenson, "Per-Survivor Processing Applied to Decision Directed Channel Estimation for a Coherent Diversity Receiver," in *Proc., IEEE Int. Symp. on Spread Spectrum Techniques and Applications*, (Sun City, South Africa), pp. 580–584, 2-4 September 1998.

[122] N. Seshadri, "Joint Data and Channel Estimation Using Blind Trellis Search Techniques," *IEEE Transactions on Communications*, vol. 42, no. 2/3/4, pp. 1000–1011, February-April 1994.

[123] M. J. Omidi, P. G. Gulak, and S. Pasupathy, "Parallel Structures for Joint Channel Estimation and Data Detection over Fading Channels," *IEEE Journal on Selected Areas of Communication*, vol. 16, no. 9, pp. 1616–1629, December 1998.

[124] D. K. Borah and B. D. Hart, "Frequency-Selective Fading Channel Estimation with a Polynomial Time-Varying Channel Model," *IEEE Transactions on Communications*, vol. 47, no. 6, pp. 862–873, June 1999.

[125] E. Baccarelli and R. Cusani, "Combined Channel Estimation and Data Detection Using Soft Statistics for Frequency-Selective Fast-Fading Digital Links," *IEEE Transactions on Communications*, vol. 46, no. 4, pp. 424–427, April 1998.

[126] C. Tellambura, Y. J. Guo, and S. K. Barton, "Channel Estimation Using Aperiodic Binary Sequences," *IEEE Communications Letters*, vol. 2, no. 5, pp. 140–142, May 1998.

[127] J. K. Tugnait and U. Gummadavelli, "Blind Equalization and Channel Estimation with Partial Response Input Signals," *IEEE Transactions on Communications*, vol. 45, no. 9, pp. 1025–1031, September 1997.

[128] A. Müller and J. M. H. Elmirghani, "Blind Channel Estimation and Echo Cancellation Using Chaotic Coded Signals," *IEEE Communications Letters*, vol. 3, no. 3, pp. 72–74, March 1999.

[129] Z. Ding, "Characteristics of Band-Limited Channels Unidentifiable from Second-Order Cyclostationary Statistics," *IEEE Signal Processing Letters*, vol. 3, no. 5, pp. 150–152, May 1996.

[130] D. Boss, K. Kammeyer, and T. Petermann, "Is Blind Channel Estimation Feasible in Mobile Communication Systems? A Study Based on GSM," *IEEE Journal on Selected Areas of Communication*, vol. 16, no. 8, pp. 1479–1492, October 1998.

[131] G. Xu, H. Liu, L. Tong, and T. Kailath, “A Least-Squares Approach to Blind Channel Identification,” *IEEE Transactions on Signal Processing*, vol. 43, no. 12, pp. 2982–2993, December 1995.

[132] J. Delmas, H. Gazzah, A. P. Liavas, and P. A. Regalia, “Statistical Analysis of Some Second-Order Methods for Blind Channel Identification/Equalization with Respect to Channel Undermodeling,” *IEEE Transactions on Signal Processing*, vol. 48, no. 7, pp. 1984–1998, July 2000.

[133] E. S. A. Chevreuil, P. Loubaton, and G. B. Giannakis, “Blind Channel Identification and Equalization Using Periodic Modulation Precoders: Performance Analysis,” *IEEE Transactions on Signal Processing*, vol. 48, no. 6, pp. 1570–1586, June 2000.

[134] X. Li and H. Fan, “Linear Prediction Methods for Blind Fractionally Spaced Equalization,” *IEEE Transactions on Signal Processing*, vol. 48, no. 6, pp. 1667–1675, June 2000.

[135] T. Nagayasu, H. Kubo, K. Murakami, and T. Fujino, “A Soft-Output Viterbi Equalizer Employing Expanded Memory Length in a Trellis,” *IEEE Transactions on Communications*, vol. E80-B, no. 2, pp. 381–385, February 1997.

[136] D. Hatzinakos and C. L. Nikias, “Blind Equalization Using a Tricepstrum-Based Algorithm,” *IEEE Transactions on Communications*, vol. 39, no. 5, pp. 669–682, May 1991.

[137] A. J. Weiss and B. Friedlander, “Channel Estimation for DS-CDMA Downlink with Aperiodic Spreading Codes,” *IEEE Transactions on Communications*, vol. 47, no. 10, pp. 1561–1569, October 1999.

[138] E. Aktas and U. Mitra, “Complexity reduction in subspace-based blind channel identification for ds/cdma systems,” *IEEE Transactions on Communications*, vol. 48, no. 8, pp. 1392–1404, August 2000.

[139] M. Benthin and K. D. Kammeyer, “Influence of Channel Estimation on the Performance of a Coherent DS-CDMA System,” *IEEE Transactions on Vehicular Technology*, vol. 46, no. 2, pp. 262–268, May 1997.

[140] M. Stojanovic and Z. Zvonar, “Performance of multiuser detection with adaptive channel estimation,” *IEEE Transactions on Communications*, vol. 47, no. 8, pp. 1129–1132, August 1999.

[141] M. Honary, *Trellis Decoding of Block Codes*. Boston, MA, USA: Kluwer Academic Publishers, First ed., 1998.

[142] D. J. Muder, “Minimal Trellises for Block Codes,” *IEEE Transactions on Information Theory*, vol. 34, pp. 1049–1053, 1988.

[143] S. Dolinar, L. Ekroot, A. Kiely, W. Lin, and R. J. McEliece, “The Permutation Trellis Complexity of Linear Block Codes,” in *Proc., 32nd Allerton Conference on Communication, Control and Computing*, (Monticello, IL, USA), pp. 60–74, 28-30 September 1994.

[144] X. Wang and S. B. Wicker, “Soft Trellis Decoders for Some BCH Codes,” in *Proc., International Symposium on Information Theory and its Applications*, (Victoria, BC, Canada), pp. 700–703, 17-20 September 1996.

[145] G. D. Forney, Jr., “Dimension/Length Profiles and Trellis Complexity of Linear Block Codes,” *IEEE Transactions on Information Theory*, vol. IT-40, pp. 1741–1752, 1994.

[146] G. D. Forney, Jr., “Dimension/Length Profiles and Trellis Complexity of Lattices,” *IEEE Transactions on Information Theory*, vol. IT-40, pp. 1753–1772, 1994.

[147] A. Lafourcade-Jumenbo and A. Vardy, “Lower Bounds on Trellis Complexity of Block Codes,” *IEEE Transactions on Information Theory*, vol. IT-41, pp. 1938–1954, 1995.

[148] G. C. Clark, Jr. and J. C. Bibb, *Error-Correction Coding for Digital Communications*. New York, NY, USA: Plenum Press, 1981.

[149] F. R. Kschischang and G. B. Horn, “A Heuristic for Ordering a Linear Block Code to Minimize Trellis State Complexity,” in *Proc., 32nd Allerton Conference on Communication, Control and Computing*, (Monticello, IL, USA), pp. 75–84, 28-30 September 1994.

[150] T. Kasami, T. Takata, T. Fujiwara, and S. Lin, “On the Optimum Bit Orders with Respect to the State Complexity of Trellis Diagrams for Binary Linear Codes,” *IEEE Transactions on Information Theory*, vol. IT-39, pp. 242–245, 1993.

[151] D. Divsalar and F. Pollara, “Turbo Trellis Coded Modulation with Iterative Decoding for Mobile Satellite Communications,” *International Journal of Satellite Communications*, 1997.

[152] E. Biglieri, “Coding and Modulation for the Fading Channels.” Presented at ISSSTA’98, Tutorial 4, 2-4 September 1998.

[153] F. Gagnon and D. Haccoun, “Bounds of the Error Performance of Coding for Nonindependent Rician-fading Channels,” *IEEE Transactions on Communications*, vol. 40, no. 2, pp. 351–360, February 1992.

[154] N. Sheikholeslami and P. Kabal, “A Family of Nyquist Filters Based on Generalized Raised-Cosine Spectra,” in *Proc.*, 19^{th} *Biennial Symp. Communications*, (Kingston, Ontario, Canada), pp. 131–135, 31 May - 3 June 1998.

[155] F. Marx, , and L. P. Linde, “A Combined Coherent Carrier Recovery and Decision-Directed Delay-Lock-Loop Scheme for DS/SSMA Communication Systems Employing Complex Spreading Sequences,” *Transactions of the SAIEE*, vol. 89, no. 3, pp. 131–139, September 1998.

[156] G. L. Turin, “The Effects of Multipath and Fading on the Performance of Direct-Sequence CDMA Systems,” *IEEE Journal on Selected Areas of Communication*, vol. 2, no. 4, pp. 597–603, July 1984.

[157] R. Price and P. E. Green, Jr., “A Communication Technique for Multipath Channels,” in *Proc., IRE*, vol. 46, pp. 555–570, March 1958.

[158] G. J. R. Povey, P. M. Grant, and R. D. Pringle, “A Decision-Directed Spread Spectrum RAKE Receiver for Fast-Fading Mobile Channels,” *IEEE Transactions on Vehicular Technology*, vol. 45, no. 3, pp. 491–502, August 1996.

[159] S. Benedetto and E. Biglieri, *Principles of Digital Transmission with Wireless Applications*. New York, NY, USA: Plenum Press, First ed., 1998.

[160] J. Bingham, *The Theory and Practice of MODEM Design*. New York, NY, USA: John Wiley and Sons, 1988.

[161] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Englewood Cliffs, NJ, USA: Prentice-Hall, Second ed., 1979.

[162] L. Staphorst and L. P. Linde, "On the Optimal Selection of ABC Sequences for MUI Minimization." Submitted to the IEEE Communications Letters, July 2005.

[163] M. P. C. Fossorier, S. Lin, and D. Rhee, "Bit-Error Probability for Maximum-Likelihood Decoding of Linear Block Codes and Related Soft-Decision Decoding Methods," *IEEE Transactions on Information Theory*, vol. 44, no. 7, pp. 3083–3090, November 1998.

[164] G. D. Forney, Jr., "Burst-Correcting Codes for the Classic Bursty Channel," *IEEE Transactions on Communications*, vol. COM-19, no. 5, pp. 772–781, October 1971.

[165] J. Hagenauer and L. Papke, "Decoding Turbo Codes with the Soft-Output Viterbi Algorithm (SOVA)," in *Proc., IEEE Int. Symp. on Inform. Theory*, (Trondheim, Norway), p. 164, 27 June - 1 July 1994.

[166] J. Hagenauer, P. Robertson, and L. Papke, "Iterative (Turbo) Decoding of Systematic Convolutional Codes with the MAP and SOVA Algorithms," in *Proc., ITG Conference on Source and Channel Coding*, (München, Germany), pp. 21–29, 26-28 October 1994.

[167] A. S. Barbulescu and S. S. Pietrobon, "Terminating the Trellis of Turbo-Codes in the Same State," *IEE Electronics Letters*, vol. 31, pp. 22–23, January 1995.

[168] W. J. Blackert, E. K. Hall, and S. G. Wilson, "Turbo Code Termination and Interleaver Conditions," *IEE Electronics Letters*, vol. 31, pp. 2082–2084, November 1995.

[169] O. Joerssen and H. Meyr, "Terminating the Trellis of Turbo-Codes," *IEE Electronics Letters*, vol. 30, pp. 1285–1286, August 1994.

[170] R. J. McEliece, E. R. Rodemich, and J. Cheng, "The Turbo Decision Algorithm." presented at *33rd Allerton Conf. on Communication, Control and Computing*, 4-6 October 1995.

[171] F. Daneshgaran and M. Mondin, "Design of Interleavers for Turbo Codes Based on a Cost Function," in *Proc., Int. Symp. on Turbo Codes and Related Topics*, (Brest, France), pp. 255–258, 3-5 September 1997.

[172] R. Pyndiah, A. Glavieux, A. Picart, and S. Jacq, "Near Optimum Decoding of Product Codes," in *Proc., IEEE Int. Conf. on Communications*, (New Orleans, LA, USA), pp. 339–343, 1-5 May 1994.

[173] R. Pyndiah, P. Combelles, and P. Adde, "A Very Low Complexity Block Turbo Decoder for Product Codes," in *Proc., IEEE GLOBECOM*, (London, UK), pp. 101–105, 18-22 November 1996.

[174] J. Lodge, R. Young, P. Hoeher, and J. Hagenauer, "Separable MAP Filters for the Decoding of Product and Concatenated Codes," in *Proc., IEEE Int. Conf. on Communications*, (Geneva, Switzerland), pp. 1740–1745, 23-26 May 1993.

[175] G. D. Forney, Jr., "On Decoding BCH Codes," *IEEE Transactions on Information Theory*, vol. IT-11, pp. 549–557, October 1965.

[176] R. E. Blahut, *Theory and Practise of Error Control Codes*. Reading, MA, USA: Addison-Wesley, First ed., 1983.

[177] S. Dolinar and D. Divsalar, "Weight Distributions for Turbo Codes Using Random and Non-Random Permutations," TDA Progress Report 42-121, JPL, (Pasadena, CA, USA), August 1995.