Convolutional Coding Schemes with Convenient Power Spectral Density Characteristics

by

Salvador Villarreal-Reyes

A Doctoral Thesis Submitted in Partial Fulfilment of the Requirements for the Award of Doctor of Philosophy of Loughborough University

May 2007

Abstract

This thesis analyses the effects caused by the introduction of convolutional coding schemes over the power spectral density (PSD) of impulse radio (IR) based ultra wideband (UWB) communication systems. This is an important topic since UWB emissions are expected to comply with spectral masks imposed by various regulatory bodies.

The spectral analysis of convolutionally coded/Markov-driven IR-based UWB signals is carried out by assuming that the data stream at the encoder input is generated by a binary Markov source (BMS). The signal model covers a wide variety of IR-based modulation schemes such as pulse position modulation (PPM), binary phase shift keying (BPSK), pulse amplitude modulation (PAM), biorthogonal PPM (BOPPM), PAM/PPM and pulse shape modulation (PSM), combined with periodic or random time hopping (TH) and/or direct sequence (DS) multiplication. In addition the inclusion of attenuation and random jitter has been accounted for in the signal model. Novel closed form PSD expressions for convolutionally coded/Markov-driven IR-based UWB signals are obtained by performing the spectral analysis of the signal model. The formulas provided clearly identify the relative contributions of different parameters allowing enhanced design of UWB systems.

It is shown that convolutional encoders can be used to generate IR-based UWB signals with advantageous PSD characteristics. The analysis presented in this dissertation focuses on spectral line suppression/elimination. Novel maximum free distance (MFD) binary to M-ary convolutional encoders with spectral line suppression capabilities for M-ary PPM based UWB systems are presented. Moreover, novel MFD binary convolutional encoders which generate BPSK/Q-BOPPM IR-based UWB signals whose PSD is spectral line-free are introduced. Results show that for IR-based UWB systems these convolutional encoders offer improved PSD characteristics when compared to the standard non-coded case and the best convolutional encoders known. As well, it is demonstrated that by using these convolutional encoders improved PSD characteristics and improved bit error rates (BER) can be achieved simultaneously.

A Dios y al Santo niño de Atocha por llevarme cargando durante todo el camino.

A Cecy el amor de mi vida. Gracias por todo tu amor, comprensión y apoyo incondicionales. Durante esta loca aventura tu has sido mi soporte e inspiración. Esta tesis no es solo mía sino de los dos.

A mi Madre. Gracias por estar conmigo apoyándome y escuchándome siempre que lo he necesitado. Siempre has sido mi gran ejemplo y guía. Todo lo que soy y podré ser se los debo a ti y a mi Padre

A mi Padre. Gracias por apoyarme cuando lo he requerido.

A Sergio, Soraya y Chuya. Gracias por que aun en la distancia he sentido que han estado conmigo.

Acknowledgments

I would like to thank my supervisor Dr. Robert M. Edwards for his valuable support and guidance through the realisation of my doctoral studies.

I want to thank the CONACYT (Mexico) and the SEP-DGRI (Mexico) for the financial support of this work.

During this period in the U. K. I have met several people that have had a significant impact in my life and work. Therefore I would like to express my gratitude to Harry Ladas, Demetrios Belis, David Diaz, Lu Family, Remeikis Family, Francesca Raffi, Andy Wang, Erika Sanchez, Ballante Family, Mouratidis Family and Aquino-Rodriguez Family.

I wish to thank as well the students of the C4MCR at the University of Sheffield and the CMCR at Loughborough University who have made this a pleasant time.

Especial thanks to Dr. David Muñoz-Rodriguez for his encouragement and support during a difficult period on my studies.

Finally I would like to thank all my friends and relatives for all the support and encouragement they showed throughout this period especially to the Reyes-Reyes Family, Rodriguez-Serrato Family and Rosy and Tetetas.

Publications

[1] S. Villarreal-Reyes and R. M. Edwards, "Analysis techniques for the power spectral density estimation of convolutionally encoded impulse radio UWB signals subject to attenuation and timing jitter " *Submitted to IEEE Transactions on Vehicular Technology*, Jan. 2007.

[2] S. Villarreal-Reyes and R. M. Edwards, "Maximum free distance binary to *M*-ary convolutional codes for pseudo chaotic type time hopping PPM impulse radio UWB," *IEEE Microwave and Wireless Components Letters*, vol. 17, pp. 250 - 252, April 2007.

[3] S. Villarreal-Reyes and R. M. Edwards, "Power spectral density of convolutionally encoded biorthogonal pulse position modulated time hopping UWB signals," *at IEEE Global Telecommunications Conference 2006, GLOBECOM 2006*, Nov. 2006.

[4] S. Villarreal-Reyes and R. M. Edwards, "New convolutional encoders for spectral line suppresion in pulse position modulated time hopping impulse radio ultra wideband systems," *at The IET Seminar in Ultra Wideband Systems, Technologies and Applications 2006*, pp. 206 - 210, Apr. 2006.

[5] S. Villarreal-Reyes and R. M. Edwards, "On the comparison of measurement and simulation of the power spectral density of PPM TH-IR UWB signals," *at Loughborough Antennas & Propagation Conference, LAPC 2006*, pp. 117 - 120, Apr. 2006.

[6] S. Villarreal-Reyes and R. M. Edwards, "Spectrum shaping of PPM TH-IR based ultra wideband systems by means of the PPM modulation index and pulse doublets," *at IEEE Antennas and Propagation Society International Symposium*, pp. 631-634, July 2005.

[7] S. Villarreal-Reyes, R. M. Edwards, and J. C. Vardaxoglou, "On the use of simulation-DFT based analysis for spectral estimation of PPM TH-IR UWB signals," *at IEEE Vehicular Technology Conference 2005. VTC 2005-Spring*, pp. 1370 -1374, May 2005.

[8] S. Villarreal-Reyes and R. M. Edwards, "On the behaviour of simulation-DFT based analysis for spectral estimation of PPM TH-IR UWB signals," *at Loughborough Antennas & Propagation Conference, LAPC 2005*, pp. 164-167, Apr. 2005.

[9] S. Villarreal-Reyes and R. M. Edwards, "Spectral line suppression in TH-IR ultra wideband systems," *at 5th IEE International Conference on Mobile Communications Technologies (3G 2004)*, pp. 614-618, Oct. 2004.

Table of Contents

List of Abbreviations

AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BMS	Binary Markov Source
BOPPM	Biorthogonal Pulse Position Modulation
BPSK	Binary Phase Shift Keying
BSC	Binary Symmetric Channel
CDMA	Code Division Multiple Access
CEPT/ECC	European Conference of Postal and Telecommunications Administrations/Electronic Communications Committee
CM	Channel Model
dB	decibels
dBm	decibel reference to 1 milliwatt
DFT	Discrete Fourier Transform
DS	Direct Sequence
EDS	Energy Density Spectrum
EIRP	Effective Isotropic Radiated Power
FCC	U. S. Federal Communications Commission
FEC	Forward Error Correction
FFT	Fast Fourier Transform
FSLS	Feedforward Spectral Line Suppressive
FSSM	Finite State Sequential Machine
FT	Fourier Transform
GF	Galois Field
HVD	Hard Viterbi Decoding
i.i.d.	independent identically distributed
IEEE	Institute of Electrical and Electronics Engineers
IPI	Inter-Pulse Interference
IR	Impulse Radio
ISI	Inter-Symbol Interference
IWS	Information Weight Spectrum
lcm	least common multiple
LOS	Line of Sight
LSB	Least Significant Bit
MA	Multiple Access

MAI	Multiple Access Interference
MC	Markov Chain
MCR	Maximal Ratio Combining
MCSK	*M*-ary Code Shift Keying
MFD	Maximum Free Distance
ML	Maximum Likelihood
MSB	Most Significant Bit
NTIA	National Telecommunications and Information Administration
ODS	Optimum Distance Spectrum
OFDM	Orthogonal Frequency Division Multiplexing
OIWS	Optimum Information Weight Spectrum
OOK	On-Off Keying
p.d.f.	probability density function
p.m.f.	probability mass function
PAM	Pulse Amplitude Modulation
PCTH	Pseudo-Chaotic Time Hopping
PN	Pseudo-Noise
PPD	Pulse Position Demodulator
PPM	Pulse Position Modulation
PR	Pseudo-Random
PRF	Pulse Repetition Frequency
PSD	Power Spectral Density
PSM	Pulse Shape Modulation
Q-BOPPM	Quaternary Biorthogonal Pulse Position Modulation
Q-PPM	Quaternary Pulse Position Modulation
QPSK	Quaternary Phase Shift Keying
RBW	Resolution Bandwidth
RMS	Root Mean Square
RNS	Recursive Non-Systematic
RSLF	Recursive Spectral Line Free
RSLS	Recursive Spectral Line Suppressive
SE	Source-Encoder
SLF	Spectral Line Free
SLS	Spectral Line Suppressive
SVD	Soft Viterbi Decoding
TH	Time Hopping
UWB	Ultra Wideband

List of Figures

List of Tables

Nomenclature

$\{A_{d_{free}+i}\}$	distance spectrum of a convolutional encoder
$\{\alpha_l\}$	PAM data stream
$\{\boldsymbol{\alpha}_l\}$	stochastic PAM data stream
$\{a_{lN_w+k}\}$	PR-DS code with period χ_a
$\{\boldsymbol{a}_l\}$	perfectly random (i.i.d. and uniform distributed) DS sequence
$\{B_{d_{free}+i}\}$	weight spectrum of a convolutional encoder
$\{\beta_l\}$	PPM data stream
$\{\boldsymbol{\beta}_l\}$	stochastic PPM data stream
$\{c_{lN_w+k}\}$	PR-TH sequence with period χ_c
$\{\boldsymbol{c}_{lN_w+k}\}$	perfectly random (i.i.d. and uniform distributed) TH sequence
$a(D)$	convolutional encoder's shift register feedforward polynomial
$b(D)$	convolutional encoder's shift register feedback polynomial
$\mathbf{B}_n = [B_{n,ij}]$	next state matrix
$BM(\underline{\mathbf{r}}_i \mid \underline{\mathbf{z}}'_i)$	branch metric
χ_a	PR-DS sequence period
χ_c	PR-TH sequence period
D	delay operator
d_{free}	free distance of a binary convolutional encoder
d_{Mfree}	free distance of binary to M-ary convolutional encoder
$\delta(\cdot)$	Dirac delta function
$\mathrm{E}\{\cdot\}$	expected value
E_b	bit energy
E_s	symbol energy
E_w	pulse energy
$\phi(\cdot)$	FSSM next state function
$\Im\{\cdot\}$	Fourier transform operator
$\gamma(\cdot)$	FSSM output function
$\mathbf{G}(D)$	generator matrix of a convolutional encoder
$g_i^{(j)}(D)$	generator of a convolutional encoder (element of $\mathbf{G}(D)$)

g_{Δ} and $g_{\Delta,\Delta'}$	first order and second order p.d.f.'s of the jitter process, Δ_{lN_w+k}
$G_{\Delta}(f)$	expected value of $e^{-j2\pi f\Delta_{lN_w+k}}$ (equivalent to Fourier transform of the first order probability density function g_{Δ})
K	constraint length of a convolutional encoder
Δ_{lN_w+k}	random jitter process
$n(t)$	additive noise
$N_0/2$	AWGN two sided power spectral density
N_q	number of states of a FSSM/encoder
$N_\sigma = 2N_q$	SE pair Markov model's number of states
N_w	number of transmitted pulses per data symbol
$\overline{S}(f)$	power spectral density (PSD)
$S_C(f)$	PSD's continuous part
$S_D(f)$	PSD's discrete part
$SM(r_i^{(j)} \mid z'^{(j)}_i)$	symbol metric
$P_2(d)$	pairwise error probability between two codewords (code sequences) with Hamming distance d
$p_{\alpha,i} = \Pr\{\alpha = i\}$	probability of $\{\alpha = i\}$
$p_{\beta,j} = \Pr\{\beta = j\}$	probability of $\{\beta = j\}$
P_b	bit error probability
$PM(\mathbf{r} \mid \mathbf{z}')$	path metric
P_s	symbol error probability
$\underline{\pi_y} = [\pi_{y,0} \;\; \pi_{y,1}]$	first order BMS's steady state probabilities vector
$p_{y,ij}$	first order BMS's one step transition probability
$\mathbf{P}_y$	first order BMS's one step transition probabilities matrix
$p_{y,ij}^{(n)}$	first order BMS's n^{th} step transition probability
$\mathbf{P}_y^n$	first order BMS's n^{th} step transition probabilities matrix
$\underline{\pi_\sigma} = [\pi_{\sigma,0},\ldots,\pi_{\sigma,N_\sigma-1}]$, $\underline{\pi} = [\pi_0,\ldots,\pi_{N_\sigma-1}]$	SE pair Markov model's steady state probabilities vector
$p_{\sigma,ij}$, p_{ij}	SE pair Markov model's one step transition probability
$\mathbf{P}_\sigma$	SE pair Markov model's one step transition probabilities matrix
$p_{\sigma,ij}^{(n)}$, $p_{ij}^{(n)}$	SE pair Markov model's n^{th} step transition probability
$\mathbf{P}_\sigma^n$	SE pair Markov model's n^{th} step transition probabilities matrix
Ω	finite state sequential machine (FSSM)

$\mathcal{Q}=\{q_0,q_1,\ldots,q_{N_q-1}\}$	FSSM/encoder's finite set of states
$Q(\cdot)$	the Q-function
q_i	FSSM/encoder's state
θ_l	FSSM/encoder's state at time l
$\mathbf{r}$	received sequence after demodulation
ρ_l	first order BMS's state process
$\mathcal{R}=\{r_0,r_1\}$	first order BMS's state set
$\overline{R_x}(\tau)$	average autocorrelation of continuous time stochastic signal $x(t)$
$R_\xi(m)$	autocorrelation of the wide sense stationary process ξ_l
$\mathcal{S}=\{\mathcal{R}\times\mathcal{Q}\}$	SE pair Markov model's state set
s_i	SE pair Markov model's state
σ_l	SE pair Markov model's Markov chain state process
T_β	PPM modulation index
T_c	nominal shift caused by the PR-TH sequence
T_r	mean repetition time between pulses
T_s	symbol time ($T_s=N_wT_r$)
T_w, T_{wT}, T_{wR}	pulse durations
v	total encoder memory of a convolutional encoder
$w(t)$, $w_{Tx}(t)$, $w_{Rx}(t)$	pulse waveforms
$W(f)$	Fourier transform of $w(t)$
$w_{\sigma_l,k}(t)$	k^{th} pulse used for the transmission of the encoder output vector $z_l=\gamma(\sigma_l)$ at time l
$W_{\sigma_l,k}(f)$	Fourier transform of $w_{\sigma_l,k}(t)$
$x(t)$	continuous time signal (IR-based UWB signal)
$x(t)$	stochastic continuous time signal (IR-based UWB signal)
$x_{Per}(t)$	periodic pulse train
ξ_{lN_w+k}	random attenuation process
$\mathcal{Y}=\{i_0,i_1,\ldots,i_{N_y-1}\}$	set of FSSM input symbols or input symbol vectors
$\mathbf{y}$	information sequence generated by a data source
$\mathbf{y}'$	estimate of the information sequence, $\mathbf{y}$, after decoding
$y^{(i)}$, $y^{(i)}(D)$	input sequence to shift register i of a convolutional encoder
$\underline{\mathbf{y}}(D)$	vector of input sequences to an encoder
y_l	data symbol generated by a stochastic data source at time l

$\underline{\mathbf{y}}_l$	input vector to an encoder at time l
$\mathcal{Z} = \{\zeta_0, \zeta_1, \ldots, \zeta_{N_s-1}\}$	set of FSSM output symbols or output symbol vectors
$\underline{\zeta}_{s_i}$	output vector corresponding to the SE pair Markov model's state s_i
$\mathbf{z}$	code sequence generated by an encoder
$\mathbf{z}'$	estimate of the code sequence, $\mathbf{z}$, after decoding
$z^{(j)}$, $z^{(j)}(D)$	sequence at the j^{th} output of a convolutional encoder
$\underline{\mathbf{z}}(D)$	vector of output sequences from a convolutional encoder
$z_l^{(k)}$	encoder's output code symbol at time l
$\underline{\mathbf{z}}_l$	encoder's output code vector at time l
z_l	SE pair Markov model's output process
$\oplus$	modulo 2 addition

Chapter 1

Introduction

Currently there are several technologies and standards in the market allowing wireless communication among devices and wireless networking (e.g. IEEE 802.11x, Bluetooth, HiperLAN). These technologies are suitable for applications that require low to moderate bit rates over the wireless link (e.g. file transfer, web browsing, e-mail, etc). For example, the IEEE 802.11g standard provides a wireless link bit rate up to 54 Mbps in an operating range of 50 metres while Bluetooth in its low power mode provides speeds up to 1 Mbps in a range of up to 10 metres. Both technologies have achieved worldwide success as evidenced by the increase of wireless networking in private and communal areas such as airports and the large amount of mobile phones equipped with Bluetooth. Nevertheless, due to the dramatic demand growth for new and improved wireless applications and services these technologies can be considered to be rapidly ageing. Thus there is a need for new wireless technologies able to fulfil the new challenges arising in areas such as wireless sensor networks and high-speed wireless personal area networks among others. One such technology that over the past few years has gained a lot of attention is ultra wideband (UWB) radio. UWB promises low power consumption, high capacity, spectrum coexistence and comparatively high data rates, [6-19].

Nevertheless, despite the potential offered by the introduction of UWB, there have been concerns about possible harmful interference from UWB to other established communication deployments, [1-5, 162]. As a consequence regulatory bodies such as the U.S. Federal Communications Commission (FCC) and the European Conference of Postal and Telecommunications Administrations/Electronic Communications Committee (CEPT/ECC) have established power limits in the form of spectral masks for intentionally generated UWB emissions, [2, 4, 5]. In this context the analysis, estimation and shaping of the power spectral density (PSD) of UWB signals is a topic of major interest for the design of compliant UWB systems.

1.1 Ultra Wideband Definition

Over the years the term ultra wideband (UWB) has been applied to several communication and radar systems, [6-11]. It turns out that terms like impulse radio, time domain, carrier-free and sub-nanosecond communications had been frequently used as synonymous of UWB. However, since its introduction in 2002 the definition provided by the U.S. Federal Communications Commission (FCC) in its 2002 Report and Order, [2], regarding UWB transmission systems has been widely adopted, [10-19].

The FCC defines a UWB emission as radio frequency energy intentionally radiated having a fractional bandwidth (to be defined in Chapter 2 – Section 2.2) larger than 20% or a bandwidth wider than 500 MHz, [2]. Thus, the FCC definition is based on spectrum occupancy and is not linked to a particular transmission technology. A consequence of this definition was that, besides the original impulse radio (IR) based UWB systems, [6-8, 20], new UWB approaches compliant with the FCC definition emerged, [13-19]. This research is focused on the IR-based UWB approach.

In brief, IR-based UWB systems convey information by transmitting low power ultra short pulses (on the order of few nanoseconds), [6-8, 20]. Several modulation schemes have been proposed for IR-based UWB systems including pulse position modulation (PPM), pulse amplitude modulation (PAM), on-off keying (OOK), binary phase shift keying (BPSK), biorthogonal PPM (BOPPM) and pulse shape modulation (PSM), [6-8, 13-20]. Traditionally, several pulses are transmitted per data symbol in order to increase the symbol energy and achieve an acceptable bit error rate (BER) performance, [6-8, 13-20]. Alternatively, this basic pulse repetition scheme can be replaced by a rate equivalent coded scheme to increase the system's BER performance without increasing the transmit power, [21-32].

Another important characteristic of typical IR-based UWB signals is that they usually employ pseudo-random (PR) time hopping (TH) and/or PR direct sequence (DS) multiplication for multiple access and/or PSD shaping purposes, [6-8, 13-19]. Basically, a TH-IR-based UWB system transmits pulses with a very low duty cycle. The relative position of each pulse within a specified time frame is then determined by a PR-TH sequence. In a pure DS-IR-based UWB system the relative position of each pulse within a time frame remains unchanged but the pulse amplitude is multiplied by a PR-DS usually taking values of the set $\{-1, +1\}$. Thus the duty cycle of a DS-IR-based UWB signal can be higher than the duty cycle of a TH-IR-based UWB signal. Finally a TH-DS IR-based UWB signal combines both techniques and its duty cycle can be as low as that of a TH-IR-based UWB signal.

1.2 Motivation of this Research

As previously mentioned, various regulatory bodies such as the FCC and the CEPT/ECC have established power limits for UWB emissions in the form of spectral masks, [2, 4, 5]. The aim of these masks is to enable the coexistence of UWB systems with already established narrowband systems and harmonise the use of the frequency spectrum. Therefore, unlike traditional narrowband systems, the UWB emissions are not necessarily limited in terms of spectrum occupancy (as most narrowband systems are) but in terms of the maximum allowed transmit power across an "ultra" wide range of frequencies. Thus one of the main research areas when considering IR-based UWB systems is the maximisation of the signal's transmit power while maintaining compliance with the spectral mask limits imposed by current UWB regulations.

As an example, the spectral mask for average emission limits as defined by the FCC regulation for indoor UWB systems, [2], is shown in Figure 1-1 (the European mask will be introduced in Chapter 2 - Section 2.2.2). This spectral mask is specified in terms of effective isotropic radiated power (EIRP) in dBm as measured with a 1 MHz resolution bandwidth (RBW), root mean square (RMS) average detector and an average time of 1 ms or less, [2].

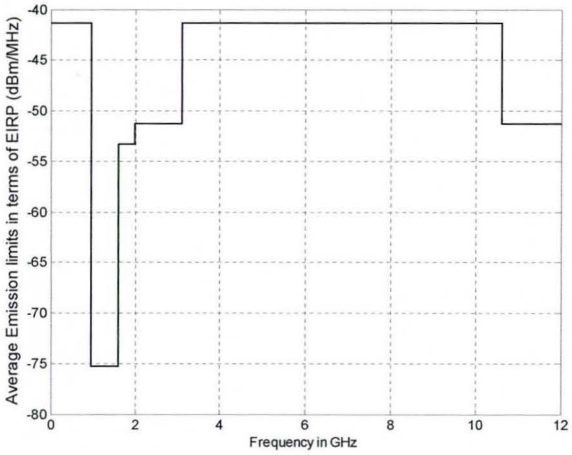


Figure 1-1. FCC's spectral mask for indoor UWB systems (the European mask will be introduced in Chapter 2 - Section 2.2.2).

Note that any non-compliant UWB system can be forced to fit under a spectral mask by properly adjusting its transmit power. However such a method would have the drawback that less energy would be transmitted per data symbol and thus the BER may worsen. A way to circumvent this BER performance loss would be to increase the number of transmitted pulses per symbol. Nevertheless this scheme would imply decreasing the system's data rate. Therefore alternative spectral shaping methods which do not incur such performance penalties can be usefully considered.

As an example let us look at how the simulated PSD of an IR-based UWB signal using binary PPM without TH compares with the FCC' spectral mask for indoor UWB systems shown in Figure 1-1. This comparison is shown in Figure 1-2. It can be seen that this particular signal does not comply with the FCC's spectral mask. Moreover, to make it fit by reducing the overall power a reduction of more than 10 dBm/MHz would be needed.

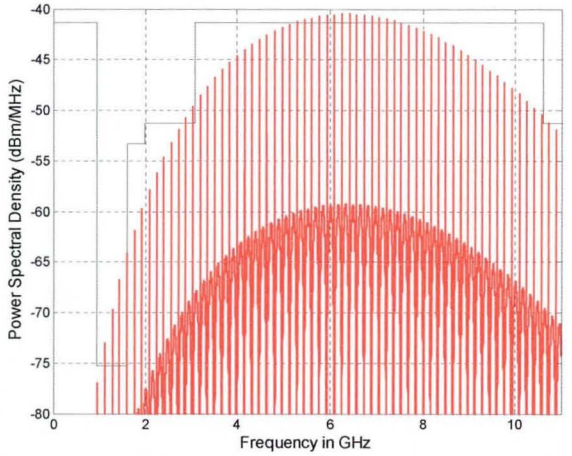


Figure 1-2. Comparison between the simulated PSD (red) of a binary PPM IR-based UWB signal and the FCC's spectral mask for indoor UWB systems (continuous black line). The 3rd derivative Gaussian pulse was used with duration $T_w \approx 0.35$ ns. The transmit power was 0.005 mW. No TH was used in this signal.

Note how the PSD in Figure 1-2 consists of a continuous component and spectral lines spaced at 160 MHz intervals. The spectral lines that appear in the PSD are due to the periodic components present in the UWB signal. These periodic components may arise due to deterministic elements in the signal; the modulation format used, or/and the statistics of the data stream, [13, 83].

Let us now consider what happens when perfectly random (ideal) 16-ary TH is added to the previous binary PPM IR-based UWB signal. The comparison between the simulated PSD for this case and the FCC spectral mask is shown in Figure 1-3. Note that the signal used for this example has the same base parameters (e.g. transmit power and data rate) as the signal used to obtain Figure 1-2. The only difference is the addition of TH in the second signal. It can be seen in Figure 1-2 that the addition of perfectly random TH enables the compliance with the FCC regulation without the need of reducing the signal's transmit power. Furthermore, note how in this case the number of spectral lines is reduced as they now appear spaced at 2.56 GHz intervals.

It is important to highlight that the use of perfectly random TH is an idealisation that in practice is addressed by generating a pseudo-random (PR) TH sequence with extremely long period. Nevertheless, the use PR-TH sequences with extremely long period increases the transmitter and receiver complexity. However, this complexity increase may not be reflected in significant BER performance improvements (when comparing systems with the same transmit power). Thus, there exists the possibility of finding novel PSD shaping mechanisms that enable compliance with the current UWB regulations while providing significant BER performance improvements when compared to traditional UWB systems' implementations.

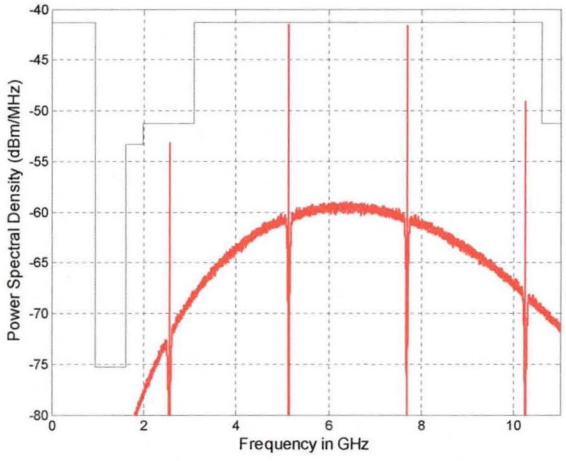


Figure 1-3. Comparison between the simulated PSD (red) of a binary PPM IR-based UWB signal using perfectly random 16-ary TH and the FCC's spectral mask for indoor UWB applications (continuous black line). The 3rd derivative Gaussian pulse was used with duration $T_w \approx 0.35$ ns. The transmit power was 0.005 mW. Note how the signal used for this example has the same base parameters (e.g. transmit power and data rate) as the signal used to obtain Figure 1-2.

Based on the previous discussion it can be said that the analysis, estimation and shaping of the PSD of UWB signals is a topic of major interest for in the design of IR-based UWB systems as a properly designed system will enable compliance with the UWB regulations without incurring significant performance loss. This work deals with these topics in general and in particular proposes ways to simultaneously achieve improved BER performance and convenient PSD

characteristics in IR-based UWB systems through the use of properly designed convolutional encoders. This matter is important due to the likely introduction of error correction mechanisms in UWB systems for BER and/or capacity improvement purposes, [19, 21-40].

1.3 Scope and Objectives of the Thesis

The research presented in this thesis deals with the spectral analysis and spectral shaping of impulse radio (IR) based ultra wideband (UWB) signals. In particular this work studies the effects over the transmitted signal's power spectral density (PSD) caused by the introduction of convolutional coding in IR-based UWB systems.

This thesis aims and objectives can be listed as follows:

- Perform the spectral analysis of convolutionally coded IR-based UWB signals such that different variables inherent to this kind of signals like time hopping (TH), direct sequence (DS) multiplication, attenuation and/or jitter are included.
- Provide an analytical closed form expression for the spectral analysis of convolutionally coded IR-based UWB signals.
- Analyse the spectral shaping effects caused by the introduction of convolutional coding in IR-based UWB systems.
- Identify ways to use the convolutional encoding operation not only for bit error rate (BER) performance improvement but as well for spectral shaping purposes.
- Suggest and find convolutional encoders which provide both convenient PSD characteristics and good BER performance (compared with an equivalent non-coded system) when used on IR-based UWB systems.

The first step for achieving these objectives consists of the definition of a model for the convolutional encoding operation such that the theoretical analysis of the signal's PSD can be performed. The second step consists of obtaining an analytical expression for the PSD of convolutionally coded IR-based UWB signals and verifying the validity of this result through examples and PSD estimates obtained from simulation. The third step involves analysing the previous results to identify the variables which enable the spectral shaping of the signal's PSD. Once these variables have been identified, a fourth step can be made to determine which conditions a convolutional encoder must fulfil to provide advantageous spectral shaping. Next the fifth step is to devise an algorithm to find convolutional encoders that provide both improved PSD characteristics and good BER performance. The sixth and last step consists of comparing the PSD characteristics and BER performance of IR-based UWB systems using the convolutional encoders found, the best convolutional encoders known and no convolutional coding through analysis and simulation.

The literature survey of previous work relevant to the research presented in this thesis is introduced in the next section.

1.4 Literature Survey

As established in Section 1.3, this work deals with two main topics: the PSD of convolutionally coded IR-based UWB signals and the use of convolutional encoders for simultaneous spectral shaping of IR-based UWB signals and BER improvement. Thus a summary of previous research related to the areas of spectral analysis of IR-based UWB signals, spectral shaping of IR-based UWB signals and application of convolutional coding in IR-based UWB systems is presented in the following subsections. An extended review of the main results relevant for this thesis will be provided in Chapter 2.

1.4.1 Previous Work on the Spectral Analysis of IR-Based UWB Signals

The spectral analysis of IR-based UWB signals has been widely addressed in the literature, [13, 18, 41-50]. In these works PSD expressions for non-coded IR-based UWB signals have been reported including with more or less degree different signal variables such as pseudo-random (PR)/random time hopping (TH), pseudo-random (PR) /random direct sequence multiplication (DS), timing jitter and/or attenuation. A brief description of the results reported in these works is provided next.

The PSD expression for non-coded digital pulse based signals subject to timing jitter is reported in [41] for PAM signals. Their model considers the PAM modulating stream to be wide sense stationary. This observation is important due to it determines the applicability range of the PSD expressions introduced in this paper.

The PSD expression for non-coded PAM/PPM TH-IR-based UWB signals subject to timing jitter is reported in [42, 44, 45]. All the signal models adopted in these works assumed mutual independence between the PAM modulating stream, the PPM modulating stream and the timing jitter. In [44] the PAM data stream is assumed to be wide sense stationary while the PPM data stream and the jitter are assumed to be second order stationary in the strict sense. The model used in [45] assumes the PAM and PPM data streams to be stationary and uncorrelated. Lastly in [42] the PAM and PPM data streams are assumed to be independent identically distributed (i.i.d.).

The signal model used in [43] is similar to the model used in [44] but focuses on the effects of deterministic TH codes and therefore do not consider timing jitter. In this paper the PAM and PPM data streams are assumed to be i.i.d. Furthermore, mutual independence between the PAM and PPM data streams is assumed as in the previous papers. The analytical method introduced in [43] was used in [48] to evaluate the PSD of M-ary code shift keying (MCSK) IR-based UWB signals.

The results and analysis presented in [13] and [18] are equivalent to the results and analysis introduced in the previously mentioned works (that is [42-45]).

In [47] the spectral analysis of UWB multiple access (MA) schemes is addressed assuming perfect (ideal) random scrambling, no timing jitter, no attenuation and statistically independent data bits.

A shot noise approach was used in [46, 49, 50] for the spectral analysis of PAM/PPM IR-based UWB signals with TH-DS, timing jitter and attenuation. These works were based on the

assumption of i.i.d. data streams for PPM, i.i.d. and wide sense stationary data streams for PAM, i.i.d. timing jitter and mutual independence of the PAM and PPM data streams.

Even though the general analytical framework presented in some of the previously mentioned papers could be further developed for the analysis of different modulation schemes, most of the analysis actually performed in these works focuses on PAM, PPM and related modulation formats (e.g. PAM/PPM) where all data streams and random variables are assumed to be mutually independent. Furthermore, as most of these papers assume mutual independence of the PAM and PPM data streams, the results presented there are not applicable in a straightforward way to the spectral analysis of IR-based UWB signals where the PAM and PPM data streams are correlated.

An important case of mutually dependent PAM and PPM data streams arises when the input to the IR-based UWB modulator is given by the output of a convolutional encoder. Thus the results presented in the previously mentioned papers do not cover this particular case. This is due to the correlation introduced by the convolutional encoder on the data stream. For example, if the IR-based UWB system uses biorthogonal PPM (BOPPM), [51], in conjunction with a convolutional encoder, then the PAM and PPM data streams will be correlated and a different approach must be used to perform the spectral analysis. Therefore it may be postulated that prior to this thesis there was a lack of studies addressing the spectral analysis of UWB signals when the data input driving the modulator consists of symbols generated by a convolutional encoder.

Spectral analysis of coded signals has been previously addressed in [52-54] for block codes, in [55, 56] for line codes, in [57, 58] for trellis codes and [59, 60] for convolutional codes. All these works assume ideal synchronous data pulse streams. Therefore these results do not consider the effects of variables such as time hopping, DS multiplication, timing jitter and pulse attenuation. Furthermore the spectral analysis presented in [59, 60] for convolutionally coded signals focuses on feedforward encoders and PAM/binary phase shift keying (BPSK) modulation and therefore cases such as PPM, BOPPM and PSM are not covered.

1.4.2 Previous Work on Spectral Shaping of IR-Based UWB Signals

The current approaches used for spectral shaping of IR-based UWB signals can be classified into the categories of pulse shape based spectral shaping, TH code based spectral shaping and DS code based spectral shaping. All these approaches look to shape the transmitted signal's PSD such that compliance with emission limits is achieved and/or interference from/to narrowband users of the spectrum is minimised.

1.4.2.1 Pulse Shape Based Spectral Shaping of IR-Based UWB Signals

The basic pulse shapes used in most of the original papers introducing IR-based UWB were the first and second order derivatives of the Gaussian pulse, [6-8, 20]. However, as these waveforms do not fit well under the FCC's proposed limits, the use of higher order derivatives has been proposed in [61, 62] as they offer a better fit for the FCC's spectral masks. There exist other pulse shapes and pulse shaping mechanisms proposed in the literature, [13, 14, 63-69]. The use of any particular

pulse shape or pulse shaping mechanism depends on the target application and design goals. For example, orthogonal pulses with convenient characteristics for pulse shape modulation (PSM) have been proposed in [14, 64-66], pulse shaping mechanisms for maximum spectral mask utilization have been devised in [13, 67] and pulse shaping mechanisms for narrowband interference suppression have been introduced in [68, 69].

The use of a particular waveform determines the general PSD shape of an IR-based UWB signal. Furthermore, by properly designing the pulse shape, notches at specific frequencies of the signal's PSD can be generated as shown in [68, 69]. However, important PSD characteristics such as the appearance and suppression of spectral lines depend on other variables such as the modulation scheme employed and the use of TH and/or DS multiplication as exemplified in Section 1.2. So we see that once a particular pulse shape has been chosen, additional spectral shaping mechanisms may be needed to further improve the signal's PSD characteristics.

Lastly, it is worth mentioning that a specific pulse shape may be distorted by the antenna used in the system and the channel, [163]. Thus, pulse shape based spectral shaping is further constrained by these extra factors.

1.4.2.2 Time Hopping Code Based Spectral Shaping of IR-Based UWB Signals

From the PSD examples presented in Section 1.2 (Figures 1-2 and 1-3), it can be seen that the introduction of perfectly random (ideal) time hopping (TH) has positive effects on the signal's PSD (Figure 1-3) as a significant number of spectral has been suppressed and compliance with the FCC's spectral mask is achieved. Thus the TH code can be effectively used to shape the PSD.

The topic of spectral shaping of IR-based UWB signals through time hopping has been previously addressed in [70-77]. These works analyse different issues related to use of the TH code for spectral shaping purposes as explained next.

The effects of the time hopping code characteristics over the signal's PSD are addressed in [71, 73] for TH-IR-based UWB systems. Similarly, the effects of the TH code design over the PSD characteristics of *M*-ary code shift keying (MCSK) IR-based UWB signals are investigated in [77]. All these works address the topic of spectral line elimination/attenuation through proper design of the TH code.

In [70] a pseudo-chaotic time hopping (PCTH) scheme has been introduced. As explained in [70], PCTH uses the symbolic dynamics of chaotic maps to generate aperiodic random-like TH sequences and thus eliminates a significant number of spectral lines in the signal's PSD. This scheme aims to resemble ideal perfectly random TH by using the data stream to drive a specifically designed PCTH encoder.

The spectral line suppression issue is addressed as well in [76] where the concept of fluid TH is introduced. Shortly, fluid time hopping uses real-valued TH codes instead of traditional discrete valued codes to perform spectral shaping. The real-valued TH codes are generated by sampling a real-valued signal (or function) at predefined intervals and then using the samples to determine the pulse position on the time axis, [76].

Besides eliminating spectral lines, the TH code can be used to produce notches in the PSD. This approach has been particularly followed in [72, 74, 75] where different algorithms are proposed for the construction of codes generating spectral notches in specific frequency bands of the PSD.

1.4.2.3 Direct Sequence Based Spectral Shaping of IR-Based UWB Signals

The use of a direct sequence (DS) code for spectral shaping purposes of IR-based UWB signals has been previously addressed in [78-81]. All these works propose the multiplication of each transmitted symbol and/or pulse by a pseudo-random (PR) DS code taking values on the set $\{-1,1\}$. This is the reason why this spectral shaping technique is often referred as polarity randomisation, [78, 80, 81].

If an ideal perfectly random DS code taking values on the set $\{-1,1\}$ is assumed, then the IR-based UWB signal's PSD will not show any spectral lines, [78-81]. However, this is an idealisation that in practical terms is addressed by generating a PR-DS code with an extremely long period, [80, 81]. Note that in this approach the DS code may or may not be explicitly used for multiple access or spreading purposes.

1.4.3 Previous Work on the Application of Convolutional Coding in IR-Based UWB Systems

The application of convolutional coding in IR-based UWB systems has been previously addressed in [21, 22, 24, 25, 27, 30-32]. All these works have shown that the use of convolutional codes and related turbo codes in IR-based UWB systems offer great benefits in terms of bit error rate (BER) performance when compared with the original (non-coded) pulse repetition schemes used in [6-8, 20].

The authors in [21] showed that for binary PPM TH-IR-based UWB, a system using convolutional codes can outperform a system using the traditional pulse repetition scheme in terms of BER and/or number of supported users without increasing the transmitted power. Similar results for other modulation schemes have been reported in [22, 24, 25, 27, 30-32, 35]. Particularly the use of superorthogonal convolutional codes, [82], has been proposed in [21, 25, 32] for binary PPM IR-based UWB, in [24, 27, 30] for PAM IR-based UWB, and in [35] for PSM IR-based UWB.

Well known binary convolutional encoders such as superorthogonal codes, [82], maximum free distance (MFD) codes, [83], and rate compatible punctured convolutional codes, [84], have been used in [22, 33, 37, 38] for M-ary PPM IR-based UWB.

Orthogonal convolutional coded modulation schemes for IR-based UWB have been analysed in [31] while a biorthogonal convolutional coded modulated scheme has been proposed in [39].

It is worth mentioning that the IEEE 802.15.4a draft specification for Low-Rate Wireless Personal Area Networks considers the use of a rate 1/2 systematic convolutional encoder in its forward error correction (FEC) layer, [40]. As well the IEEE study group 802.15.3a (now disbanded) considered the use of convolutional codes for FEC purposes, [19].

Most of the papers dealing with the application of convolutional codes in IR-based UWB systems focus on BER analysis without considering the effects of the convolutional encoder over the signal's PSD. The only two exceptions are the schemes proposed in [33, 37] where the use of binary maximum free distance convolutional encoders, [83, 85], combined with *M*-ary PPM is proposed. In fact the goal of these works is to achieve enhanced PSD characteristics similar to those obtained with PCTH, [70], through the use of the binary MFD convolutional encoders. However, the binary convolutional encoders used in these works were not specifically designed for *M*-ary signalling. Thus there exists the possibility of improving the BER while achieving similar PSD characteristics by properly designing binary to *M*-ary convolutional encoders. Finally, it is worth mentioning that the spectral analysis presented in [33] was performed using simulation, while [37] uses a simplified model for the signal's spectral analysis which is based on the assumption of uniform distributed i.i.d. streams at the encoder output (that is the correlation introduced by the convolutional encoders is not considered). As well, neither [33] nor [37] consider the use of other variables such as PAM, TH, DS, timing jitter and attenuation.

1.5 Original Contributions of the Thesis

The effects over the transmitted signal's power spectral density (PSD) produced by the introduction of convolutional coding in impulse radio (IR) based ultra wideband (UWB) systems has been analysed in this thesis. A first order binary Markov source (BMS) has been chosen to perform the analysis. This kind of source enables the modelling of unbalanced (that is non-uniformly distributed) independent identically distributed (i.i.d.) memoryless data streams as well as data streams with memory, thus offering a wide applicability range. From this analysis, the conditions that a particular encoder must fulfil to provide advantageous spectral shaping characteristics have been identified. Furthermore, a method to design and find convolutional encoders with convenient PSD characteristics and good bit error rate (BER) performance has been proposed. The original contributions of the thesis can be outlined as follows:

- Derivation of a source-encoder (SE) pair model that enables us to describe the interaction of the BMS and the convolutional encoder with a single Markov model which can be classified as being of the Moore type (within the definition adopted for finite state sequential machines).
- An original spectral analysis for convolutionally coded IR-based UWB systems. The model used for the analysis and the closed form PSD expressions obtained from it cover different modulation schemes such as pulse amplitude modulation (PAM), binary phase shift keying (BPSK), pulse position modulation (PPM), biorthogonal PPM (BOPPM), PAM/PPM, and pulse shape modulation (PSM), combined with pseudo-random (periodic)/random time hopping (TH) and/or pseudo-random/random direct sequence (DS). In addition the signal model and PSD expressions account for the inclusion of attenuation and/or timing jitter.

- The identification of convolutional encoder's characteristics that have a positive or negative effect in the transmitted signal's PSD.
- The classification of pseudo-chaotic time hopping (PCTH) and related schemes as feedforward binary to *M*-ary convolutionally coded PPM IR-based UWB systems.
- The identification of the variables that enable the design of feedforward binary to *M*-ary convolutional encoders with similar PSD characteristics to those achieved with the PCTH scheme under the same operation constraints (that is i.i.d. binary data streams with uniform distribution are assumed at the encoder input).
- A code search procedure to seek feedforward binary to *M*-binary convolutional encoders with PSD characteristics similar to those achieved with PCTH and improved BER performance.
- A set of new *M*-ary maximum free distance (MFD) feedforward binary to *M*-ary convolutional encoders that have the same spectral line suppressive characteristics as the PCTH scheme (under similar operation conditions) and provide improved BER performance over the PCTH scheme.
- The identification of the variables and conditions that enable the design of binary to *M*-ary convolutional encoders with similar PSD characteristics to those achieved with the PCTH scheme even when the encoder input does not consist of an i.i.d. binary data stream with uniform distribution. In particular the data stream is assumed to be generated by a generic first order BMS.
- The modification of the code search procedure such that spectral line suppressive binary to *M*-ary convolutional encoders for unbalanced (that is, non-uniform distributed) BMS's can be found.
- Two sets of novel *M*-ary MFD spectral line suppressive binary to *M*-ary convolutional encoders which eliminate as many spectral lines as the PCTH scheme but keep its spectral line suppressive characteristics even when the binary stream at the encoder input is generated by an unbalanced BMS (that is, the data stream does not consists of uniform distributed i.i.d. binary symbols).
- The novel proposal of using binary convolutional encoders for spectral line elimination purposes in BPSK and quaternary BOPPM (Q-BOPPM) IR-based UWB systems.
- The identification of the variables and conditions that enable the design of binary convolutional encoders which generate signals with spectral line-free PSDs when used in BPSK/Q-BOPPM IR-based UWB systems with unbalanced BMS inputs.
- A code search procedure aimed to seek spectral line-free binary convolutional encoders with good BER performance.
- The introduction of nine sets of novel MFD binary convolutional encoders which generate a spectral line-free signal even when the binary stream at the encoder input is generated by an unbalanced BMS (that is, the data stream does not consists of uniform distributed i.i.d.

binary symbols). Encoders with rates from 1/2 up to 1/32 and memory orders from 3 up to 9 are provided.

1.6 Outline of the Thesis

Chapter 2 introduces concepts, terminology and definitions that will be used throughout this entire thesis. Two main areas can be identified in Chapter 2. The first area describes the UWB approach adopted in this work where several important topics such as current UWB regulations, the classic IR-based UWB scheme, the PSD of non-coded IR-based UWB signals and spectral shaping mechanisms for IR-based UWB signals are reviewed. The second area addresses concepts related to convolutional coding. In particular an overview of convolutional coding applied to IR-based UWB systems and a review of basic concepts related to convolutional coding is provided.

In Chapter 3 the spectral analysis of convolutionally coded/Markov-driven IR-based UWB signals is performed. The system and signal models used throughout Chapters 4, 5, 6 and 7 are introduced in this chapter. First a source-encoder Markov model is obtained as this is the starting point of the analysis. The corresponding signal model is then introduced followed by the derivation and analysis of the signal's average autocorrelation and average power spectrum. Afterwards, illustrative application examples which enable to assess and identify the effects of the convolutional coding operation over the PSD are provided. Comparisons between analytical and simulated PSD results are provided for validation purposes.

In Chapter 4 we look at the design of feedforward binary to M-ary convolutional encoders with similar PSD characteristics to those obtained with pseudo-chaotic time hopping (PCTH) and improved bit error rate (BER) performance. The constraints that a binary to M-ary convolutional encoder must fulfil to suppress as many spectral lines as PCTH under the same operation conditions (i.i.d. binary data streams with uniform distribution at the encoder input) are identified. Then new feedforward binary to M-convolutional encoders with spectral line suppression capabilities are presented. These encoders were found through a code search procedure introduced in this chapter. Afterwards comparisons between PSD plots obtained when using the new encoders and the PCTH scheme are shown. Next BER performance comparisons between the new encoders and the PCTH scheme are provided. Lastly, conclusions drawn from these comparisons are presented.

Chapter 5 addresses the problem of designing spectral line suppressive binary to M-ary convolutional encoders which eliminate as many spectral lines as PCTH even when the binary stream at the encoder input does not consist of i.i.d. symbols with uniform distribution. Here it is assumed that encoder's input consists of data streams generated by an unbalanced (that is, non-uniform distributed) first order binary Markov source (BMS). In a similar way to Chapter 4, the constraints that a binary to M-ary convolutional encoder must fulfil to achieve the desired spectral line suppressive characteristics are identified. This is followed by the modification of the code search procedure introduced in Chapter 4 to account for the new constraints. Novel binary to M-ary convolutional encoders that satisfy the spectral line suppression condition for unbalanced BMS's

are then introduced. Afterwards comparisons in terms of PSD characteristics and BER performance between the new encoders and the encoders introduced in Chapter 4 are presented. Lastly, conclusions drawn from these comparisons are provided.

The novel idea of using binary convolutional encoders for spectral line elimination purposes in binary phase shift keying (BPSK) and quaternary biorthogonal pulse position modulation (Q-BOPPM) IR-based UWB systems is introduced in Chapter 6. This chapter deals with the design of binary convolutional encoders that generate BPSK/Q-BOPPM IR-based UWB signals with spectral line-free PSDs even when the binary stream at the encoder input consist of symbols generated by an unbalanced BMS. The conditions that the binary convolutional encoders must fulfil to generate such spectral line-free signals are identified and a specific code search procedure is introduced. Novel "spectral line-free" binary convolutional encoders for BPSK/Q-BOPPM IR-based UWB systems are then introduced. Afterwards, comparisons in terms of PSD characteristics and BER performance between BPSK/Q-BOPPM IR-based UWB systems using the newly found convolutional encoders, the best convolutional encoders known and no convolutional coding are presented. Lastly, conclusions drawn from these comparisons are provided.

Finally, Chapter 7 presents the general conclusions of the thesis and outlines the key findings. This chapter finalises with suggestions of future research areas.

Chapter 2

Theoretical Background

2.1 Introduction

The purpose of this chapter is to introduce the concepts, terminology and definitions that will be used in the chapters that follow. A key theme of this research is the effect of the convolutional encoding operation on the power spectral density (PSD) of ultra wideband (UWB) signals. Therefore two main areas can be identified in this chapter. The first area is dealt with in Sections 2.2 to 2.5 and describes the UWB approach adopted for this work. The second area is written in Sections 2.6 and 2.7 and concerns several important concepts related to convolutional coding

In summary, the structure of Chapter 2 is as follows. Section 2.2 introduces definitions for UWB and regulatory issues. A basic non-coded impulse radio (IR) based UWB approach is introduced in Section 2.3 while its power spectral density (PSD) is presented in Section 2.4. In Section 2.5 a description of spectral shaping mechanisms for IR-based UWB signals is provided. Section 2.6 gives an overview of convolutional coding applied to IR-based UWB while Section 2.7 introduces basic concepts related to convolutional codes. Finally conclusions are provided in Section 2.8.

2.2 Ultra Wideband Definition and Regulatory Issues

As mentioned in Chapter 1, since its introduction in 2002 the definition provided by the U.S. Federal Communications Commission (FCC) in its 2002 Report and Order, [2], regarding UWB transmission systems has been widely be adopted, [10-19]. The FCC defines a UWB emission as radio frequency energy intentionally radiated having a fractional bandwidth larger than 20% or a bandwidth wider than 500 MHz, where the fractional bandwidth is defined as

$$\text{Fractional Bandwidth} = 2\frac{(f_H - f_L)}{(f_H + f_L)} \tag{2.1}$$

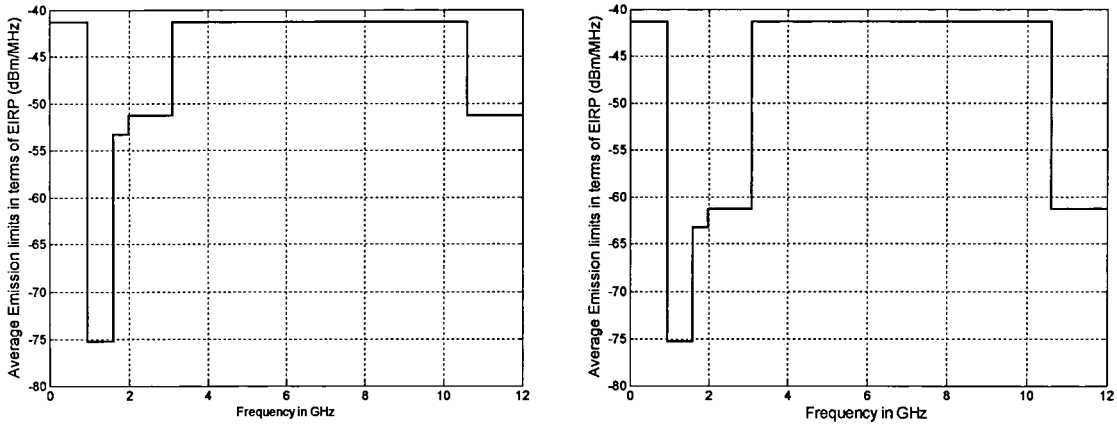
with f_H and f_L set at the -10 dB upper and lower emission points in the signal spectrum. Thus, the FCC definition is based on spectrum occupancy and is not linked to a particular transmission technology. However, as mentioned in the introduction, this work focuses on IR-based UWB implementations.

2.2.1 FCC Emission Limits for UWB Emissions

The FCC's 2002 Report and Order [2] established emission limits for UWB emissions in the form of spectral masks. In Table 2-1 and Figure 2-1 the average emission limits for UWB emissions as specified by the FCC for indoor and outdoor applications are presented, [2].

Table 2-1. Spectral masks for UWB average emission limits in terms of EIRP in dBm/MHz established by the FCC.

Frequency band in GHz	Power levers in terms of EIRP (dBm/MHz)	
	Devices for indoor applications	Devices for outdoor applications
0.000009-.960	−41.3	−41.3
.960-1.610	−75.3	−75.3
1.610-1.990	−53.3	−63.3
1.990-3.100	−51.3	−61.3
3.100-10.600	−41.3	−41.3
10.600-22.000	−51.3	−61.3
22.000-29.000	−51.3	−61.3
Above 29.000	−51.3	−61.3



a) Indoor applications b) Outdoor applications

Figure 2-1. FCC UWB spectral masks for indoor and outdoor applications.

These average emission limits are given in terms of effective isotropic radiated power (EIRP) in dBm as measured with a 1 MHz resolution bandwidth (RBW), root mean square (RMS) average detector and average time of 1 ms or less (in case a spectrum analyser is used to perform the measurements), [2]. In addition the FCC regulation establishes a limit of 0 dBm EIRP on the peak emission level contained within a 50 MHz bandwidth centred on the frequency at which the highest radiated emission occurs.

The FCC set these limits for protection of previously established users of the spectrum, particularly GPS and military systems.

2.2.2 CEPT/ECC Emission Limits for UWB Emissions

In 2006 the European Conference of Postal and Telecommunications Administrations/Electronic Communications Committee (CEPT/ECC) released the decisions ECC/DEC/(06)04 and ECC/DEC/(06)12 regulating the operation of UWB devices in Europe, [4, 5]. In Table 2-2 the emissions limits as specified in the March 2007 decision ECC/DEC/(06)04 are presented. Note the

CEPT/ECC limits only agree with the FCC regulation on the 6 to 8.5 GHz band as extra protection for diverse radiocommunications systems deployed across Europe is provided.

Table 2-2. CEPT/ECC decision ECC/DEC/(06)04 on the emission limits for devices using UWB technology.

Frequency Range in GHz	Maximum mean EIRP density (dBm/MHz)	Maximum peak EIRP density (dBm/50MHz) (Note 1)
Below 1.6	−90	−50
1.6 to 3.8	−85	−45
3.8 to 4.8	−70	−30
4.8 to 6.0	−70	−30
6.0 to 8.5	−41.3	0
8.5 to 10.6	−65	−25
Above 10.6	−85	−45

Note 1: The peak EIRP can be alternatively measured in a 3 MHz bandwidth. In this case, the maximum peak e.i.r.p. limits to be applied is scaled down by a factor of 20log(50/3) = 24.4 dB.

On December 2007 the CEPT/ECC released the decision ECC/DEC/(06)12 relaxing the ECC/DEC/(06)04 limits on the 3.8 to 4.8 GHz band for devices employing Low Duty Cycle (LDC) mitigation mechanisms. This decision allows the operation of LDC UWB devices at a level of −41.3 dBm/MHz in the frequency band 3.8 to 4.8 GHz with the following requirements:

- T_{on} max = 5ms, where T_{on} is defined as the duration of a burst irrespective of the number of pulses contained.
- T_{off} mean $\geq$ 38 ms (averaged over 1 s), where T_{off} is defined as the time interval between two consecutive bursts when the UWB emission is idle.
- The sum of the T_{off} periods must be larger than 950 ms per second.
- The sum of the T_{on} periods must be less than 5% per second and 0.5% per hour.

Further decisions from CEPT/ECC are expected regarding UWB systems using Detect and Avoid (DAA) mechanisms, [4, 5].

2.3 Impulse Radio Based UWB Systems

As previously mentioned this work focuses on impulse radio (IR) based UWB implementations. The IR-based UWB approach uses low power ultra short pulses (on the order of few nanoseconds) to transmit information, [6-8, 13-20]. Several modulation schemes have been proposed for IR-based UWB systems including pulse position modulation (PPM), pulse amplitude modulation (PAM), on-off keying (OOK), binary phase shift keying (BPSK), biorthogonal PPM (BOPPM) and pulse shape modulation (PSM), [6-8, 13-20]. The Figure 2-2 shows BPSK, binary PPM, binary PSM and quaternary BOPPM (Q-BOPPM) with Gray mapping. Each modulation scheme has its own advantages and disadvantages in terms of bit error rate (BER) performance, complexity and PSD characteristics of the signal. As well it is worth noting that due to the pulse's low power the transmission of several pulses per data symbol may be needed in order to achieve an acceptable BER, [6-8, 13-20].

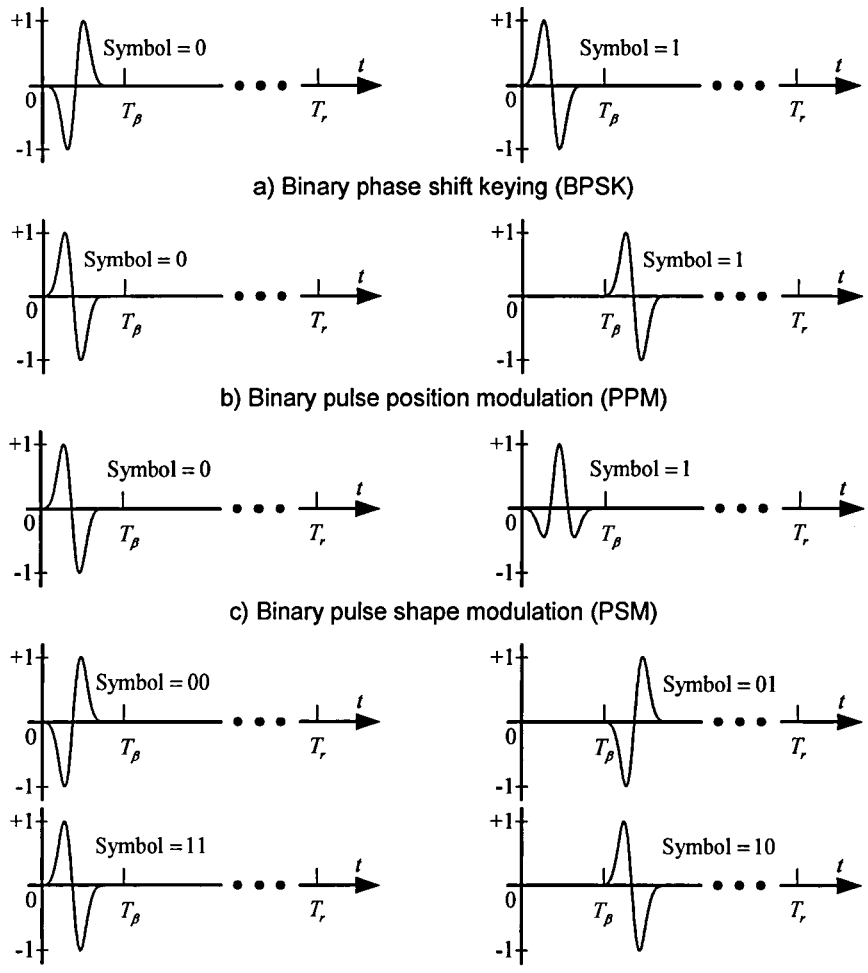


Figure 2-2. Several modulation schemes for IR-based UWB. T_β is the PPM modulation index and T_r is the mean repetition time between pulses.

The next subsections introduce the particular IR approaches relevant for this research.

2.3.1 Time Hopping IR-Based UWB

As originally introduced in [6-8, 20], time hopping (TH) IR UWB used PPM to convey the information. Nevertheless it will be more useful to define a general model covering other modulation formats as well. Bearing this in mind, the transmitted and received signals in a typical single user TH-IR-based UWB system can be described by

$$\begin{aligned} x_{Tx}(t) &= \sum_{l=-\infty}^{\infty} \sum_{k=0}^{N_w-1} \alpha_l w_{Tx}(t - lT_s - kT_r - c_{lN_w+k}T_c - \beta_l T_\beta) \\ x_{Rx}(t) &= \sum_{l=-\infty}^{\infty} \sum_{k=0}^{N_w-1} \alpha_l w_{Rx}(t - lT_s - kT_r - c_{lN_w+k}T_c - \beta_l T_\beta - \tau) + n(t) \end{aligned} \tag{2.2}$$

where $w_{Tx}(t)$ and $w_{Rx}(t)$ are the waveforms of the transmitted and received pulses respectively, $\{\alpha_l\}$ and $\{\beta_l\}$ are the PAM and PPM data streams (not restricted to be binary) and N_w is the number of transmitted pulses per data symbol. The pseudo-random (PR) TH sequence $\{c_{lN_w+k}\}$ takes values on the set $\{0, 1, 2 \ldots, N_c - 1\}$ with period χ_c. The TH sequence $\{c_{lN_w+k}\}$ is normally

used for multiple access purposes and to improve the spectral characteristics of the transmitted signal, [6-8, 20]. T_{wT} and T_{wR} are the durations of the transmitted and received pulses, T_r is the mean repetition time between pulses, T_s is the symbol time ($T_s = N_w T_r$), T_β is the PPM modulation index and T_c is the nominal shift caused by the PR-TH sequence. Usually $T_r >> \max(T_{wT}, T_{wR})$, hence the signal has a very low duty cycle. The basic TH concept for two different PR-TH sequences is shown in Figure 2-3. Finally τ is the propagation delay and $n(t)$ is additive noise. It is worth mentioning that the two pulse shapes model used for the transmitted and received signals was introduced to account for the antennas and channel effects over the shape of the originally transmitted pulse, [7, 8].

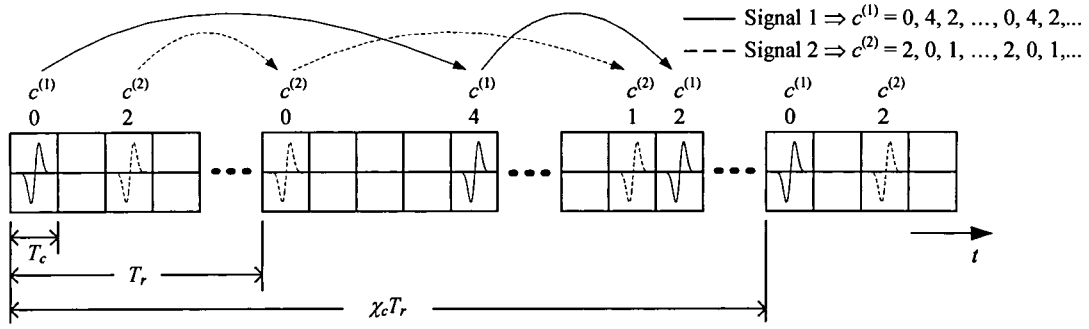


Figure 2-3. Pulse positions for two TH–IR UWB signals using different PR-TH sequences. $c^{(1)}$ and $c^{(2)}$ are two different PR-TH sequences, T_r is the mean repetition time between pulses, T_c is the nominal shift caused by the PR-TH sequence and χ_c is the period of the PR-TH sequence.

The use of α_l and β_l in Equation (2.2) allows the modelling of several modulation schemes such as BPSK, PAM, PPM, OOK and BOPPM. For example, if M-ary PPM is considered, then $\alpha_l = 1$ and β_l will take values on the set {0, 1, 2 ..., $M-1$}. For clarity and unless otherwise stated only orthogonal PPM will be considered in this work. In order to guarantee the M-ary PPM symbols are orthogonal and avoid intersymbol interference (ISI), it will be assumed that $T_\beta \geq \max(T_{wT}, T_{wR})$, $T_c \geq M T_\beta$, and $T_r \geq N_c T_c$. It is worth mentioning that, although not explicitly covered by the signal model defined by Equation (2.2), PSM has been proposed as well for TH-IR-based UWB systems, [14, 64-66].

If BPSK is considered then $\alpha_l \in \{-1,1\}$ and $\beta_l = 0$ whereas for M-ary BOPPM $\alpha_l \in \{-1,1\}$ and $\beta_l \in \{0,1,...,(M/2)-1\}$, [51]. Note BPSK, BOPPM and orthogonal PPM TH-IR-based UWB are in fact carrier free/pulse based implementations of antipodal, biorthogonal and orthogonal signaling. Hence, if $\{c_{lN_w+k}\}$ and τ are known at the receiver, all the data symbols are equiprobable and $n(t)$ is additive white Gaussian noise (AWGN) with two sided power spectral density $N_0/2$, then the optimum receiver is the correlation/matched filter receiver followed by maximum likelihood (ML) detection, [83, 86]. Therefore, the symbol error probability, P_s, for BPSK IR-based UWB is given by

$$P_s = P_b = Q\left(\sqrt{\frac{2E_s}{N_0}}\right) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) = Q\left(\sqrt{\frac{2N_w E_w}{N_0}}\right) \tag{2.3}$$

where P_b is the bit error probability, E_s is the symbol energy, E_b is the energy per bit, E_w is the pulse energy and $Q(\cdot)$ is defined as ([83, 86])

$$Q(\varpi) = \frac{1}{\sqrt{2\pi}} \int_{\varpi}^{\infty} e^{-\lambda^2/2} d\lambda \tag{2.4}$$

For M-ary PPM the symbol error probability can be found to be

$$\begin{aligned} P_s = P_b &= Q\left(\sqrt{\frac{E_s}{N_0}}\right) = Q\left(\sqrt{\frac{E_b}{N_0}}\right) = Q\left(\sqrt{\frac{N_w E_w}{N_0}}\right), && \text{Binary PPM} \\ P_s = P_M &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \left[1 - \left(\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{y} e^{-x^2/2} dx\right)^{M-1}\right] \exp\left[-\frac{1}{2}\left(y - \sqrt{\frac{2E_s}{N_0}}\right)^{M-1}\right] dy \le (M-1) Q\left(\sqrt{\frac{E_s}{N_0}}\right), && M\text{-ary PPM} \end{aligned} \tag{2.5}$$

where the inequality in the left side is a union bound for the error probability. Note for both BPSK and binary PPM the symbol error probability, P_s, and the bit error probability, P_b, are equivalent. It is important to recall as well that the error probability depends on the Euclidean distance between symbols as obtained when the signals are represented as vectors, [83, 86].

Of particular interest for this research is the case of quaternary BOPPM (Q-BOPPM) with Gray code mapping. This modulation format is geometrically equivalent to quaternary phase shift keying (QPSK) with Gray mapping, [83, 87], and therefore its bit error probability can be found to be, ([83, 87, 88])

$$P_b = Q\left(\sqrt{\frac{E_s}{N_0}}\right) = Q\left(\sqrt{\frac{N_w E_w}{N_0}}\right) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \tag{2.6}$$

Thus the bit error probability of BPSK and Q-BOPPM (Gray) IR-based UWB is the same in terms of the spent energy per bit, E_b.

The theoretical formulas introduced in this section for the evaluation of error probabilities will be used in the following chapters to compare the BER performances of non-coded IR-based UWB systems and convolutionally coded IR-based UWB systems.

2.3.2 Direct Sequence IR-Based UWB and Mixed TH-DS IR-Based UWB

In the direct sequence (DS) IR-based UWB approach the pulse amplitude is multiplied by a pseudo-random (PR) sequence combined with the elimination or reduction of time hopping, [16, 89, 90]. Therefore a typical DS-IR-based UWB signal can be described by

$$\begin{aligned} x_{Tx}(t) &= \sum_{l=-\infty}^{\infty} \sum_{k=0}^{N_w-1} a_{lN_w+k} \alpha_l w_{Tx}(t - lT_s - kT_r) \\ x_{Rx}(t) &= \sum_{l=-\infty}^{\infty} \sum_{k=0}^{N_w-1} a_{lN_w+k} \alpha_l w_{Rx}(t - lT_s - kT_r - \tau) + n(t) \end{aligned} \tag{2.7}$$

where $\{a_{lN_w+k}\}$ is a PR-DS sequence with period χ_a usually taking values from the set $\{-1,1\}$. The sequence $\{a_{lN_w+k}\}$ is normally used for multiple access purposes as in traditional code division multiple access (CDMA). Commonly in DS-IR-based UWB T_r is set to be just slightly larger than $\max(T_{wT}, T_{wR})$ (although this is not mandatory). Thus the signal's duty cycle may not be as low as in the TH-IR-based UWB approach, [16]. A typical DS-IR-based UWB signal is shown in Figure 2-4.

The modulation format typically used for DS-IR-based UWB is BPSK, although PAM, OOK and PSM can be implemented as well. Orthogonal and biorthogonal signalling can be achieved by using standard CDMA techniques through the use of Hadamard or Walsh-Hadamard codes, [16, 82, 91].

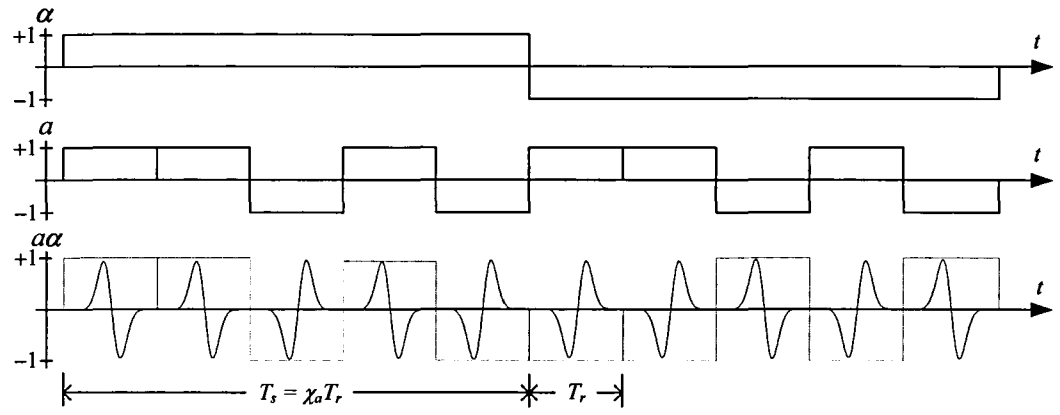


Figure 2-4. Typical DS-IR-based UWB signal where $\{\alpha_l\}$ is the PAM data stream, N_w is the number of transmitted pulses per data symbol, T_r is the mean repetition time between pulses, T_s is the symbol time and χ_a is the period of the PR-DS code $\{a_{lNw+k}\}$. Note $T_s = \chi_a T_r = N_w T_r$.

A mixed TH-DS IR-based UWB signal can be defined as

$$\begin{aligned} x_{Tx}(t) &= \sum_{l=-\infty}^{\infty} \sum_{k=0}^{N_w-1} a_{lN_w+k}\alpha_l w_{Tx}(t - lT_s - kT_r - c_{lN_w+k}T_c - \beta_l T_\beta) \\ x_{Rx}(t) &= \sum_{l=-\infty}^{\infty} \sum_{k=0}^{N_w-1} a_{lN_w+k}\alpha_l w_{Rx}(t - lT_s - kT_r - c_{lN_w+k}T_c - \beta_l T_\beta - \tau) + n(t) \end{aligned} \tag{2.8}$$

In this case each parameter in Equation (2.8) can be set to fulfil a specific design requirement. For example, in, [92, 93], the TH sequence is used for multiple access purposes whereas the DS code is used to smooth the signal's PSD.

2.4 Spectral Analysis of TH-IR-Based UWB Signals

Power spectral density expressions for IR-based UWB signals have been reported in, [41-50]. Most of these works assumed mutual independence between data streams driving different modulation variables such as PAM and PPM.

Consider the signal described by Equation (2.2) and assume that the PAM data stream, $\{\alpha_l\}$, and the PPM data stream, $\{\beta_l\}$, consists of i.i.d. discrete valued random variables (not necessarily uniformly distributed) with probability mass functions (p.m.f.) $p_{\alpha,i} = \Pr\{\alpha_l = i\} = \Pr\{\alpha = i\}$ and $p_{\beta,j} = \Pr\{\beta_l = j\} = \Pr\{\beta = j\}$ respectively. For clarity it will be assumed that the antennas and channel do not significantly distort the pulse shape and therefore $w_{Tx}(t) = w_{Rx}(t) = w(t)$. By assuming that $\{\alpha_l\}$ and $\{\beta_l\}$ are mutually independent the signal's PSD can be found to be

$$\begin{aligned} \overline{S}(f) &= S_{C.1}(f) + S_D(f) \qquad N_w = 1 \\ \overline{S}(f) &= S_{C.1}(f) + S_{C.2}(f) + S_D(f) \qquad N_w > 1 \end{aligned} \tag{2.9}$$

$$S_{C.1}(f) = \frac{1}{T_r}|W(f)|^2 \,\mathrm{E}\{|\alpha|^2\} - \frac{|W(f)|^2|\mathrm{E}\{\alpha\}|^2|G_\beta(f)|^2}{T_s \Lambda_{wc}} \sum_{n=0}^{\Lambda_{wc}-1} \left| \sum_{k=0}^{N_w-1} e^{j2\pi f k T_r} e^{j2\pi f T_c c_{nN_w+k}} \right|^2 \tag{2.10}$$

$$S_{C.2}(f) = \frac{2|W(f)|^2 \mathrm{E}\{|\alpha|^2\}}{T_s \Lambda_{wc}} \sum_{k=0}^{N_w-2} \sum_{k'=k+1}^{N_w-1} \sum_{n=0}^{\Lambda_{wc}-1} \mathrm{Re}\{e^{-j2\pi f(k'-k)T_r} e^{-j2\pi f T_c (c_{nN_w+k'} - c_{nN_w+k})}\} \tag{2.11}$$

$$S_D(f) = \frac{|W(f)|^2 |\mathrm{E}\{\alpha\}|^2 |G_\beta(f)|^2}{(T_s \Lambda_{wc})^2} \left| \sum_{k=0}^{N_w-1} \sum_{n=0}^{\Lambda_{wc}-1} e^{j2\pi f k T_r} e^{j2\pi f n T_s} e^{j2\pi f T_c c_{nN_w+k}} \right|^2 \sum_{r=-\infty}^{\infty} \delta(f - \frac{r}{T_s \Lambda_{wc}}) \quad (2.12)$$

where $S_{C.1}(f)$ and $S_{C.2}(f)$ conform the PSD's continuous part, $S_D(f)$ is the PSD's discrete part, $\Lambda_{wc} = \mathrm{lcm}(\chi_c, N_w)/N_w$ (lcm stands for least common multiple), $W(f)$ is the Fourier transform of $w(t)$, $\delta(\cdot)$ is the Dirac delta function, $\mathrm{E}\{\cdot\}$ is the expected value and

$$|G_\beta(f)|^2 = \mathrm{E}\{e^{-j2\pi f \beta T_\beta}\}\mathrm{E}\{e^{j2\pi f \beta T_\beta}\} \quad (2.13)$$

Equations (2.9) to (2.13) were obtained as a particular case of the spectral analysis presented in Chapter 3. Equivalent PSD expressions can be found in the papers mentioned at the beginning of this section.

From Equations (2.9) to (2.13) it can be seen that the PSD shape is affected by the pulse's energy density spectrum (EDS) $|W(f)|^2$, the TH sequence $\{c_{lN_w+k}\}$ and the statistics of $\{\alpha_l\}$ and $\{\beta_l\}$. Note as well from Equation (2.13) that the appearance of spectral lines can be expected at multiples of $1/(T_s \Lambda_{wc})$. From Equation (2.13) it can be seen that the appearance/suppression of spectral lines is influenced by deterministic elements in the signal (that is the symbol time, $T_s = N_w T_r$, the mean pulse repetition time, T_r, and the characteristics of the PR-TH sequence $\{c_{lN_w+k}\}$), the modulation format used (that is: PAM, PPM or PAM/PPM) and mean statistics of the data stream (that is $|\mathrm{E}\{\alpha\}|$ and $|G_\beta(f)|^2$).

For the PSD plots presented in the rest of this section it will be assumed that α_l and β_l have p.m.f.

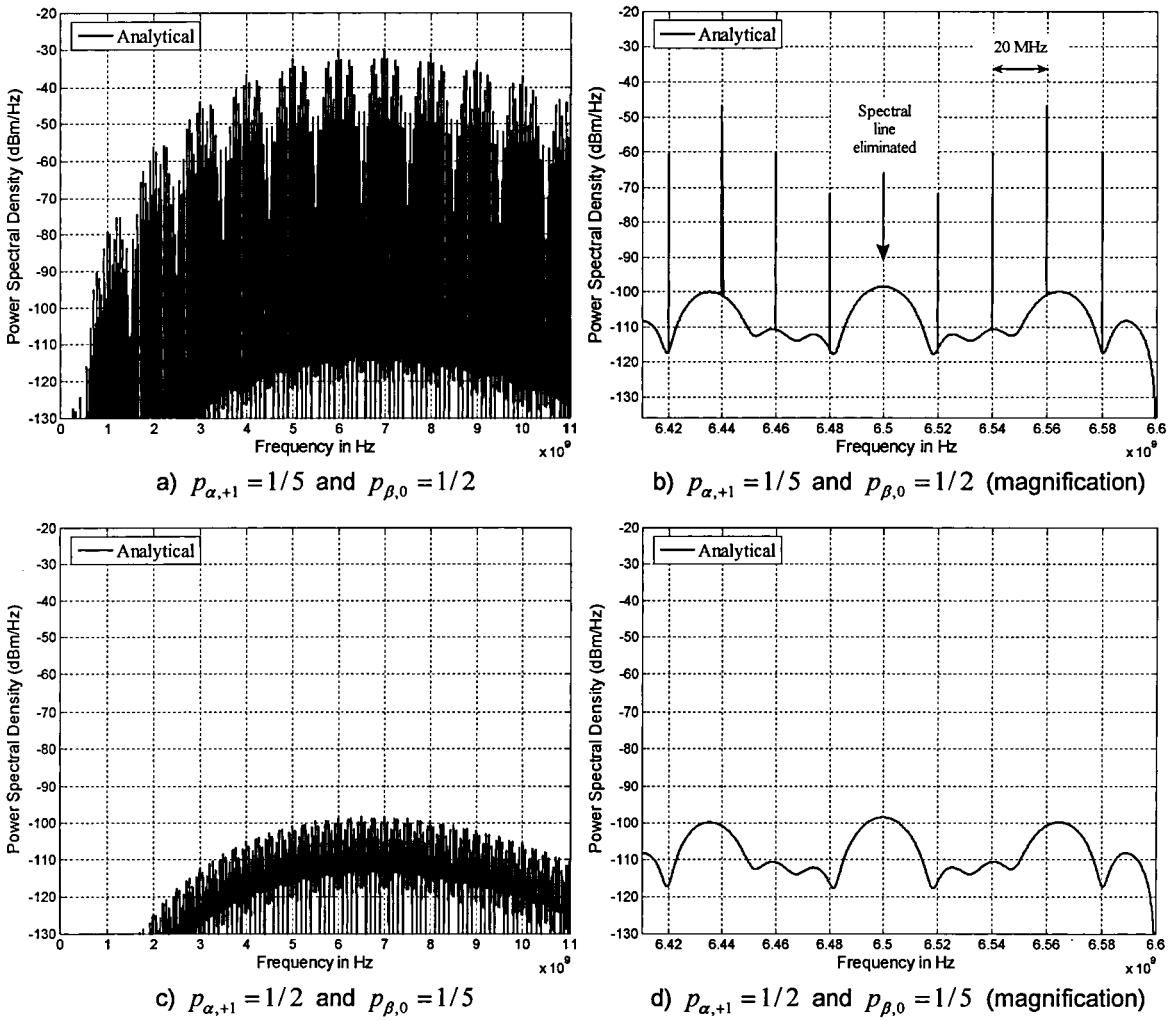
$$\begin{aligned} p_{\alpha,+1} &= \Pr\{\alpha_l = +1\} = \Pr\{\alpha = +1\}; \quad p_{\alpha,-1} = \Pr\{\alpha_l = -1\} = \Pr\{\alpha = -1\} = 1 - p_{\alpha,+1} \\ p_{\beta,0} &= \Pr\{\beta_l = 0\} = \Pr\{\beta = 0\}; \quad p_{\beta,1} = \Pr\{\beta_l = 1\} = \Pr\{\beta = 1\} = 1 - p_{\beta,0} \end{aligned} \quad (2.14)$$

that is, BPSK is used in combination with binary PPM where $0 < p_{\alpha,+1} < 1$ and $0 < p_{\beta,0} < 1$. Note that due to the fact that $\{\alpha_l\}$ and $\{\beta_l\}$ are mutually independent this model does not cover the case in which Q-BOPPM is used for the transmission of quaternary data streams in which α_l and β_l are correlated.

In Figure 2-5 several PSD plots for different values of $p_{\alpha,+1}$ and $p_{\beta,0}$ are shown. These plots were obtained by evaluating Equations (2.9) to (2.13) in MATLAB®. The signal parameters are $N_w = 5$, $T_s = 50$ ns, $T_r = 10$ ns, $T_c = 2$ ns, $T_\beta = 1$ ns. The periodic sequence {0 2 4 1 3} was used for TH. The 4th derivative Gaussian pulse was used for these plots (see Section 2.5) with approximate duration of 0.4 ns. For this set of parameters $\Lambda_{wc} = 1$ as $\chi_c = 5$ and thus spectral lines can be expected at $1/(T_s \Lambda_{wc}) = 20$ MHz intervals.

Note how for $p_{\alpha,+1} = 1/5$ and $p_{\beta,0} = 1/2$ the PSD plots in Figures 2-5a and 2-5b show a large number of spectral lines spaced at 20 MHz intervals as predicted. However, some spectral lines are eliminated due to nulls present in the absolute value of Equation (2.12) as seen in Figure 2-5b. The

spectral lines in Figures 2-5a and 2-5b may cause interference to established communication deployments and may limit the maximum transmit power of the UWB system in order to comply with the emission masks imposed by regulatory bodies such as those introduced in Section 2.2. This can be seen by noting that the maximum spectral line height in Figures 2-5a and 2-5b is about 11 dBm above the –41.3 dBm limit imposed by the FCC.



a) $p_{\alpha,+1}=1/5$ and $p_{\beta,0}=1/2$

b) $p_{\alpha,+1}=1/5$ and $p_{\beta,0}=1/2$ (magnification)

c) $p_{\alpha,+1}=1/2$ and $p_{\beta,0}=1/5$

d) $p_{\alpha,+1}=1/2$ and $p_{\beta,0}=1/5$ (magnification)

Figure 2-5. Analytical PSDs for a TH-IR-based UWB signal using BPSK and binary PPM. The signal parameters are $N_w = 5$, $T_s = 50$ ns, $T_r = 10$ ns, $T_c = 2$ ns and $T_\beta = 1$ ns. The 4th derivative Gaussian pulse was used with duration $T_w \approx 0.4$ ns. The periodic sequence {0, 2, 4, 1, 3} was used for TH.

An improvement on the PSD's characteristics is observed in Figures 2-5c and 2-5d for $p_{\alpha,+1}=1/2$ and $p_{\beta,0}=1/5$. Note how for this case the spectral lines are completely eliminated even when the PPM data stream has a biased p.m.f. ($p_{\beta,0}=1/5$). The spectral lines disappear due to the BPSK data stream is perfectly random ($p_{\alpha,+1}=p_{\alpha,-1}=1/2$) and thus Equation (2.12) becomes zero as $|\mathrm{E}\{\boldsymbol{\alpha}\}|^2=0$ when $p_{\alpha,+1}=p_{\alpha,-1}=1/2$.

In order to better assess the effects of the modulating data streams in the PSD's shape let us now consider a non-modulated periodic pulse train with period T_{Per} defined as

$$x_{Per}(t)=\sum_{k=-\infty}^{\infty} w(t-kT_{Per}) \tag{2.15}$$

Using classic Fourier analysis the PSD of $x_{Per}(t)$ can be found to be ([94])

$$S_{x_{Per}}(f) = \frac{1}{T_{Per}^{\ 2}} \sum_{n=-\infty}^{\infty} |W(\tfrac{n}{T_{Per}})|^2 \, \delta(f - \tfrac{n}{T_{Per}}) \tag{2.16}$$

Thus the PSD of a periodic pulse train consists of spectral lines appearing at multiples of the pulse repetition frequency, PRF = $1/T_{Per}$. The PSD of a periodic pulse train with pulse repetition period $T_{Per} = 10$ ns is shown in Figure 2-6. Note how this PSD consists of spectral lines spaced at PRF = $1/T_{Per}$ = 100 MHz intervals.

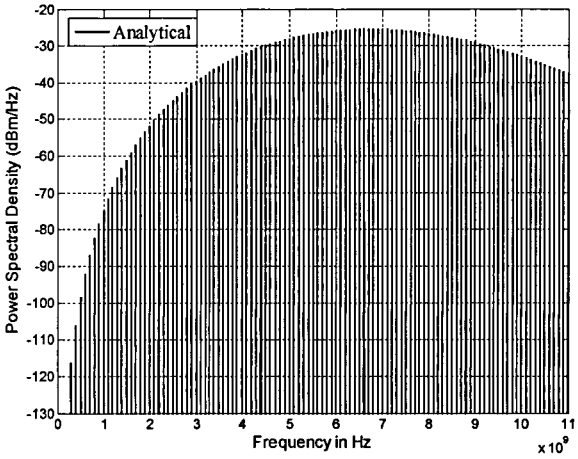
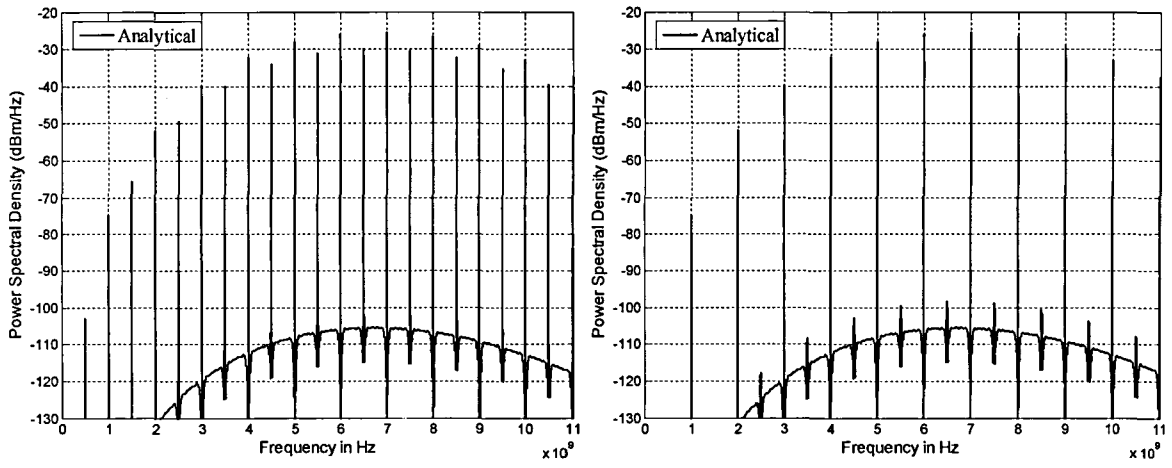


Figure 2-6. Analytical PSD of a periodic pulse train. The signal pulse repetition period is T_{Per} = 10 ns. The 4th derivative Gaussian pulse was used with duration $T_w \approx 0.4$ ns.

Comparing Equation (2.16) with Equations (2.9) to (2.13) it can be seen that the introduction of modulation and time hopping shapes the signal's PSD in relation to the power spectrum of a non-modulated periodic pulse train. That is, upon the introduction of modulation and TH some of the power that otherwise would entirely be distributed over spectral lines (Equation (2.16)) is transformed into a continuous PSD component (Equations (2.10) and (2.11)). In order to further see this consider a binary PPM TH-IR-based UWB system employing a perfectly random TH sequence (that is, the TH sequence is i.i.d. with uniform distribution) with c_{lN_w+k} uniformly distributed over the set {0, 1, 2, 3, 4}. The PSD plots for this system are shown in Figure 2-7 for different values of $p_{\beta,0}$ (note for this case $p_{\alpha,+1} = 1$ as BPSK is not considered).

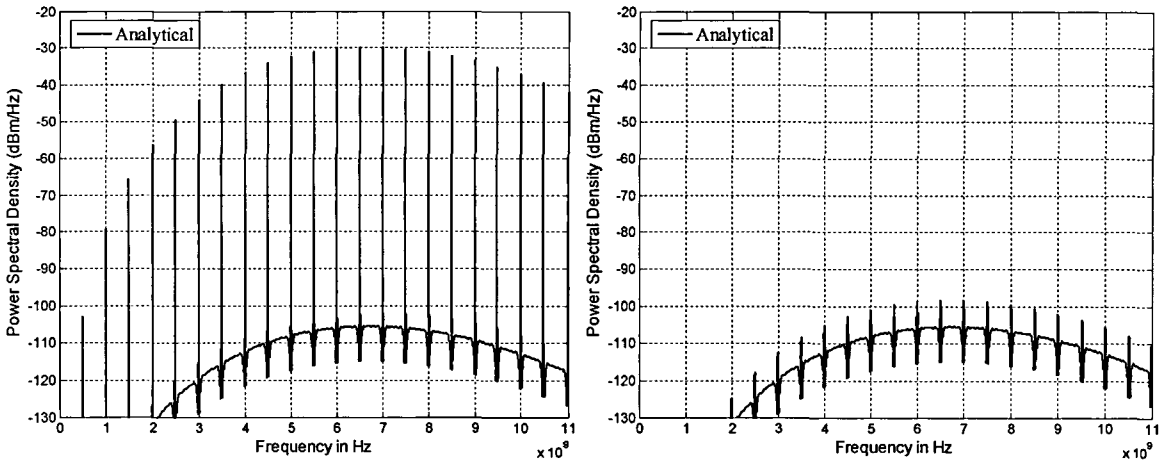
By comparing Figure 2-7 with Figures 2-5a, 2-5b and 2-6 it can be seen that by assuming a perfectly random (ideal) TH sequence the number of spectral lines in the PSD can be reduced. Furthermore, note how for $p_{\beta,0} = 1/5$ spectral lines appear at 500 MHz intervals (Figure 2-7a) while for $p_{\beta,0} = 1/2$ they appear at 1 GHz intervals (Figure 2-7b). This is an important observation which implies that better PSD characteristics can be achieved when the binary PPM data stream is perfectly random (that is, the data stream is an i.i.d sequence of binary random variables with uniform distribution).



a) $p_{\alpha,+1} = 1$ and $p_{\beta,0} = 1/5$ b) $p_{\alpha,+1} = 1$ and $p_{\beta,0} = 1/2$

Figure 2-7. Analytical PSDs for a TH-IR-based UWB system using binary PPM with perfectly random TH. The signal parameters are $N_w = 5$, $T_s = 50$ ns, $T_r = 10$ ns, $T_c = 2$ ns, $T_\beta = 1$ ns and c_{lN_w+k} uniformly distributed over the set {0, 1, 2, 3, 4}. The 4th derivative Gaussian pulse was used with duration $T_w \approx 0.4$ ns.

Now consider a BPSK TH-IR-based UWB system employing a perfectly random TH sequence with c_{lN_w+k} uniformly distributed over the set {0, 1, 2, 3, 4}. The PSD plots for different values of $p_{\alpha,+1}$ are shown in Figure 2-8 (note for this case $p_{\beta,0} = 1$ as PPM is not considered). Similar to the previous example, spectral lines appear at 500 MHz intervals for $p_{\alpha,+1} = 1/5$ (Figure 2-8a). Nevertheless when $p_{\alpha,+1} = 1/2$ the spectral lines are completely eliminated (Figure 2-8b) due to $|\mathrm{E}\{\alpha\}|^2 = 0$ as in Figures 2-5c and 2-5d.



a) $p_{\alpha,+1} = 1/5$ and $p_{\beta,0} = 1$ b) $p_{\alpha,+1} = 1/2$ and $p_{\beta,0} = 1$

Figure 2-8. Analytical PSDs for a TH IR-based UWB system using BPSK with perfectly random TH. The signal parameters are $N_w = 5$, $T_s = 50$ ns, $T_r = 10$ ns, $T_c = 2$ ns, $T_\beta = 1$ ns and c_{lN_w+k} uniformly distributed over the set {0, 1, 2, 3, 4}. The 4th derivative Gaussian pulse is used with duration $T_w \approx 0.4$ ns.

The next section introduces some common spectral shaping mechanisms proposed for IR-based UWB systems.

2.5 Spectral Shaping Methods for IR-Based UWB Signals

This section introduces the main spectral shaping mechanisms proposed for IR-based UWB signals. We can classify current spectral shaping methods into the categories of pulse shape based spectral shaping, TH code based spectral shaping and DS code based spectral shaping.

2.5.1 Pulse Shape Based Spectral Shaping Mechanisms for IR-Based UWB Signals

From Equations (2.9) to (2.12) it can be seen that the pulse's energy density spectrum (EDS), $|W(f)|^2$, dictates the overall form acquired by the IR signal's PSD. The basic waveforms used in most of the original papers introducing IR-based UWB were the first and second order derivatives of the Gaussian pulse, [6-8, 20]. The basic Gaussian pulse can be defined as

$$w_{G,0}(t) = \frac{A}{\sigma\sqrt{2\pi}} e^{-t^2/2\sigma^2} \tag{2.17}$$

where σ can be used to control the pulse duration. The Fourier transform of $w_{G,0}(t)$ is

$$W_{G,0}(f) = Ae^{-(j2\pi f\sigma)^2/2} \tag{2.18}$$

Therefore the Fourier transform for the k^{th} order derivative of the Gaussian pulse will be

$$W_{G,k}(f) = A(j2\pi f)^k e^{-(j2\pi f\sigma)^2/2} \tag{2.19}$$

and the time domain representation can be recursively obtained by using ([61])

$$w_{G,k}(t) = -\frac{k-1}{\sigma^2} w_{G,k-2}(t) - \frac{t}{\sigma^2} w_{G,k-1}(t) \tag{2.20}$$

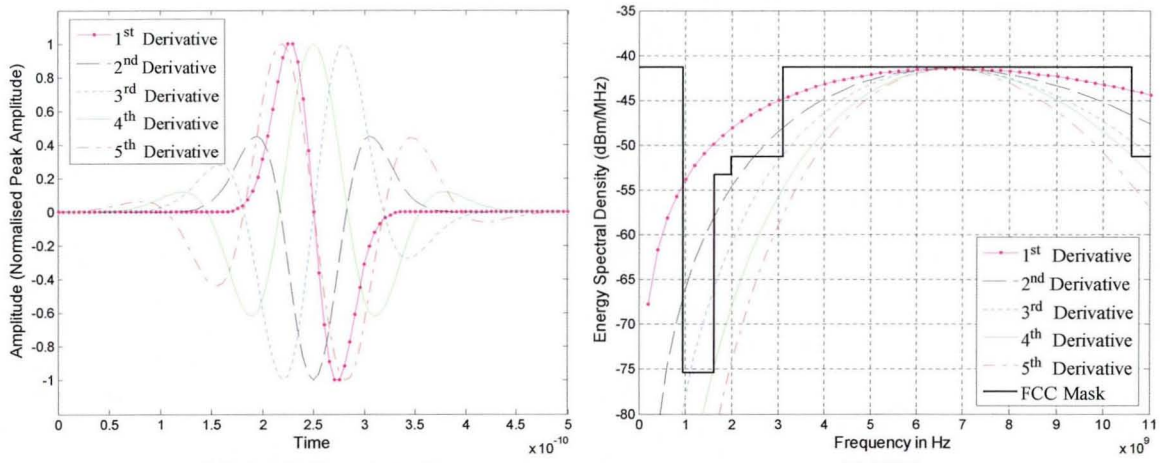
with the 1st derivative Gaussian pulse defined by

$$w_{G,1}(t) = \frac{-At}{\sigma^3\sqrt{2\pi}} e^{-t^2/2\sigma^2} \tag{2.21}$$

Figure 2-9 shows time domain and EDS, $|W_{G,k}(f)|^2$, plots for the 1st to 5th order derivatives of the Gaussian pulse. The FCC indoor spectral mask is plotted together with the EDS plots for comparison purposes. It can be seen from Figure 2-9b that the 1st and 2nd order derivatives of the Gaussian pulse do not fit under the FCC's proposed limits. In contrast the 4th and 5th order derivative offer a better fit.

There exist other pulse shapes and pulse shaping mechanisms proposed in the literature, [13, 14, 62-69]. Similar to the modulation schemes, the use of any particular pulse shape or pulse shaping mechanism depends on the target application and design goals. For example, orthogonal pulses with convenient characteristics for PSM have been proposed in [14, 64-66], pulse shaping mechanisms for maximum spectral mask utilization have been devised in [13, 67] and pulse shaping mechanisms for narrowband interference suppression have been introduced in [68, 69]. However, important PSD characteristics such as the appearance and suppression of spectral lines depend on other variables such as the modulation scheme employed and the use of TH and/or DS multiplication as

exemplified in Section 2.4. So we see that once a particular pulse shape has been chosen, additional spectral shaping mechanisms can be used to further improve the signal's PSD characteristics.



a) Pulse in time domain b) ESD

Figure 2-9. Pulse shape and EDS plots of the 1st to 5th order derivatives of a Gaussian pulse. The value of σ was set such that all EDS plots reach its maximum value at 6.7 GHz.

It is worth mentioning that a specific pulse shape may be distorted by the antenna used in the system and the channel, [163]. Therefore, even if we select a given pulse shape for specific spectral shaping purposes, the radiated signal may show a different PSD due to these extra factors. Hence, the effects of the antenna and channel should be accounted for when using pulse shape based spectral shaping.

2.5.2 Time Hopping Code Based Spectral Shaping Mechanisms for IR-Based UWB Signals

As in the previous case, from Equations (2.9) to (2.12) it can be seen that besides multiple access purposes a pseudo-random (PR) time hopping (TH) sequence can be used to shape the PSD of the IR-based UWB signal. In fact, the topic of spectral shaping of IR-based UWB signals through time hopping has been previously addressed in [70-77]. These works analyse different issues related to use of the TH code for spectral shaping purposes as explained next.

The effects of the time hopping code characteristics over the signal's PSD are addressed in [71, 73] for TH-IR-based UWB systems. Similarly, the effects of the TH code design over the PSD characteristics of *M*-ary code shift keying (MCSK) IR-based UWB signals are investigated in [77]. All these works address the topic of spectral line elimination/attenuation through proper design of the TH code.

The effectiveness of TH code-based spectral shaping can be observed by comparing the PSD plot shown in Figure 2-6, corresponding to a periodic signal, with the PSD plots shown in Figure 2-7 where perfectly random TH was assumed. In particular note how several spectral lines were eliminated when the TH sequence was assumed to be perfectly random. Nevertheless, as previously mentioned in Section 1.2, the use of perfectly random TH is an idealisation that in practice is addressed by generating a pseudo-random (PR) TH sequence with extremely long

period. Bearing this in mind, pseudo-chaotic time hopping (PCTH), [70], and fluid time hopping, [76], have been proposed to generate TH sequences with improved spectral line elimination capabilities. Shortly, fluid time hopping uses real-valued TH codes instead of traditional discrete-valued codes to perform spectral shaping, [76]. The real-valued TH codes are generated by sampling a real-valued signal (or function) at predefined intervals and then using these samples to determine the pulse position on the time axis, [76]. In contrast, PCTH uses the symbolic dynamics of chaotic maps to generate aperiodic random-like TH sequences, [70]. The PCTH scheme will be discussed with more detail in Section 2.5.5.

Besides eliminating spectral lines, the TH code can be used to produce notches in the PSD. This approach has been particularly followed in [72, 74, 75] where different algorithms are proposed for the construction of TH codes generating spectral notches in specific frequency bands of the PSD.

2.5.3 Direct Sequence Code Based Spectral Shaping Mechanisms for IR-Based UWB Signals

As noted in Figures 2-7 and 2-8a some spectral lines remain in the PSD even with the use of perfectly random TH. However, when BPSK is used and the modulating data stream is perfectly random (that is $p_{\alpha,+1} = p_{\alpha,-1} = 1/2$) the spectral lines are eliminated as seen in Figures 2-5c, 2-5d and 2-8b. As previously explained this is due to the multiplicative term $|\mathrm{E}\{\alpha\}|^2$ in Equation (2.12) becomes zero when $p_{\alpha,+1} = p_{\alpha,-1} = 1/2$, and thus the PSD's discrete part, $S_D(f)$, becomes zero. Bearing this in mind, the authors in [78-81] proposed the use of polarity randomisation for spectral shaping of TH IR UWB signals.

Basically the polarity randomisation scheme proposes the multiplication of each pulse (pulse based polarity randomisation) or each PAM/PPM symbol (symbol based polarity randomisation), by a perfectly random (ideal) direct sequence (DS) $\{a_l\}$ (where $a_l \in \{-1,1\}$). Thus this scheme assumes that $\{a_l\}$ is a sequence of i.i.d. random variables with uniform distribution (that is $p_{a,-1} = p_{a,+1} = 1/2$) independent of the PAM, $\{\alpha_l\}$, and PPM, $\{\beta_l\}$, data streams. Using this technique the PAM symbol in Equation (2.2) becomes $\alpha'_l = a_l \alpha_l$ where α_l is the original PAM data symbol and a_l is a uniform distributed random variable over the set $\{-1, +1\}$. Hence the mean of the PAM symbol stream becomes

$$\mathrm{E}\{\alpha'\} = \mathrm{E}\{a\alpha\} = \mathrm{E}\{a\}\mathrm{E}\{\alpha\} = 0 \tag{2.22}$$

since $\mathrm{E}\{a\} = 0$. Therefore the spectral lines are eliminated as $S_D(f) = 0$ (Equation (2.12)) due to $|\mathrm{E}\{\alpha'\}|^2 = 0$.

Generating a DS sequence with infinite period is an idealisation which in practice is addressed by generating a PR-DS code with an extremely long period. However, as it will be shown in Chapter 6, if the period of the PR-DS code is relatively short the height of the spectral lines may be reduced but the actual number of spectral lines in the PSD may be simultaneously increased.

Note that even though the pulse or symbol polarities (as driven by the DS code) can be discarded when using PPM, [81], the same can not be done for PAM, BOPPM or any other modulation format conveying information in the pulse amplitude. Therefore acquisition and synchronisation of the DS code in the receiver side is required for these modulation schemes, [95]. Nevertheless the acquisition and synchronisation of extremely long DS codes becomes a challenge for IR-based UWB systems in terms of large acquisition times and complexity of the acquisition system, [96-98]. Furthermore, [96-98] assume periodic DS codes while dealing with the acquisition problem. Thus the use of polarity randomisation with extremely long period PR-DS sequences may complicate the receiver design, [81].

2.5.4 Pseudo-Chaotic Time Hopping for IR-Based UWB

As previously mentioned, pseudo-chaotic time hopping (PCTH) uses the symbolic dynamics of chaotic maps to generate aperiodic random-like TH sequences, [70]. In particular the PCTH scheme introduces encoders that resemble the dynamics of the Bernoulli and tent maps when its input consists of i.i.d. binary symbols with uniform distribution. The basic structure of the PCTH encoder is shown in Figure 2-10.

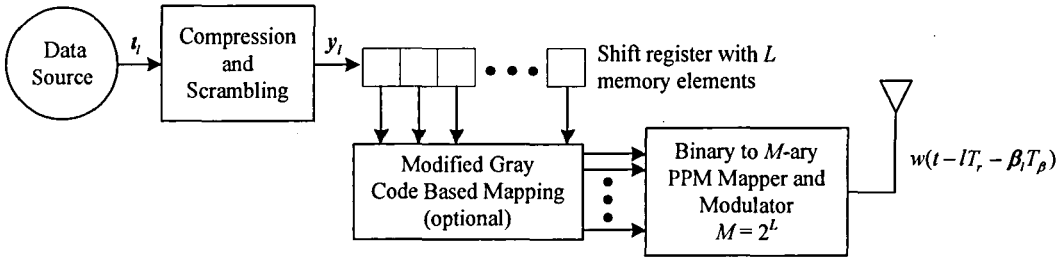


Figure 2-10. Basic block diagram of the PCTH transmitter. M-ary PPM is used as the modulation format with $M = \log_2(L)$ where L is the number of memory elements in the shift register.

The compression and scrambling block in Figure 2-10 is needed in order to feed the shift register with a uniform distributed i.i.d. binary stream. After compression and scrambling each bit, y_l, is fed to the least significant bit (LSB) position of a shift register with L memory elements. The PCTH system state, θ_l, at the l^{th} bit interval is thus defined as

$$\theta_l = 0.\theta_{l,1}\theta_{l,2}...\theta_{l,L} = \sum_{i=1}^{L} 2^{-i}\theta_{l,i} \tag{2.23}$$

where $\theta_{l,i} \in \{0,1\}$ and θ_l is an approximation to the Bernoulli shift map [70]. The modified Gray code based mapper is introduced in order to implement the tent map. In some particular cases this may help to improve the system's BER as well. For Bernoulli shift map PCTH realisations the modified Gray mapper is not used. The shift register content, $\theta_{l,1}\theta_{l,2}...\theta_{l,L}$, (or its modified Gray code based representation) is used to drive the PPM modulator, that is

$$\beta_l = \sum_{i=0}^{L-1} 2^i \theta_{l,i+1} \tag{2.24}$$

and therefore the pulse can occupy any of $M = 2^L$ different positions (M-ary PPM). Finally the mapper and modulator generate the pulse

$$w(t - lT_r - \beta_l T_\beta) \tag{2.25}$$

corresponding to the l^{th} bit interval and where the PPM modulation index is set to $T_\beta = T_r / M$. Note in PCTH all time hopping is driven by the encoder.

The PCTH scheme can also be described by a Markov chain model with $M = 2^L$ states, where each state is related to one M-ary PPM symbol. In [70] it was shown that if the input to the shift register consists of uniform distributed i.i.d. binary symbols then the steady state probabilities, π_i, of the PCTH Markov chain are equal to

$$\pi_i = 1/M; \quad i = 1,2,...,M \tag{2.26}$$

Therefore in steady state the PCTH scheme generates a sequence of *M-ary* symbols, β_k, whose statistics resemble those of a perfectly random TH sequence. Thus by setting $T_\beta = T_r / M$ several spectral lines can be eliminated in a similar fashion to that shown in Figure 2-7 where a perfectly random sequence was assumed.

It is important to note that in the original PCTH system model no additional time hopping or direct sequence multiplication was considered. Hence in [70] the PSD of the PCTH signal is obtained by using the results presented in [99] for the spectral analysis of synchronous Markov-driven signals. An extended analysis of the PCTH Markov chain model and its respective PSD is addressed in Chapter 4.

Finally on this topic, it is worth mentioning that the PCTH scheme can be interpreted as a form of convolutional coding. Particularly, in [70] it was considered that PCTH can be seen as a rate $1/L$ binary convolutional encoder due to each input bit causes an output of L bits. Moreover, this interpretation implies that maximum likelihood decoding using the Viterbi algorithm is feasible as a mean to recover the original data.

2.5.5 Summary of the Spectral Shaping Mechanisms for IR-Based UWB Signals

To finalize this section Table 2-3 (next page) presents a summary of the advantages and disadvantages of each spectral shaping method. The comments regarding the spectral line elimination capabilities of each scheme were made taking as a reference the spectral line content of a periodic signal (see Section 2.4).

Table 2-3. A comparison of spectral shaping mechanisms for IR-based UWB signals.

Spectral Shaping Mechanism	Main Characteristic	Advantages	Disadvantages
Pulse Shape Based Spectral Shaping	The pulse energy density spectrum (EDS) determines the overall PSD shape.	- The pulse EDS can be designed for maximum spectral mask utilization. - Can be used to implement orthogonal signalling. - Can be used to produce notches in the PSD.	- The pulse shape may be distorted by the antenna and channel. Thus the EDS of the radiated pulse may differ from the intended EDS. - Can not be effectively used to eliminate spectral lines in the PSD.
TH Code Based Spectral Shaping	The TH sequence can be used to eliminate spectral lines or generate notches in the PSD.	- If the TH sequence is assumed to be perfectly random, then a significant number of spectral lines can be eliminated in the PSD. - Can be used to generate several notches in the PSD simultaneously.	- To maximise the amount of spectral lines eliminated the TH sequence must be perfectly random. - The generation of a TH code with extremely long period (such that it approaches a perfectly random sequence) complicates the transmitter and receiver design. - For most practical implementations some spectral lines remain in the PSD. - The effectiveness to generate spectral notches in the PSD is adversely affected by timing jitter.
DS Code Based Spectral Shaping	The DS code can be used to eliminate spectral lines.	- If the DS sequence is assumed to be perfectly random, then a spectral line-free PSD can be achieved.	- If the DS sequence is not perfectly random, then spectral lines will appear in the PSD. - The generation of a DS code with extremely long period (such that it approaches a perfectly random sequence) complicates the transmitter and receiver design. - If the period of the DS code is relatively short, then the number of spectral lines in the PSD is increased.
PCTH	Generates random-like TH sequences using a pseudo-chaotic encoder.	- Generates aperiodic random-like TH sequences that eliminate a significant number of spectral lines in the PSD. - If Viterbi decoding is implemented in the receiver, then a significant BER improvement over non-coded binary PPM can be achieved.	- In order to generate the random-like TH sequence, the PCTH encoder must be fed with i.i.d. binary symbols with uniform distribution (e.g. perfectly random data stream). - In order obtain i.i.d. binary symbols with uniform distribution, a compression and scrambling block may be needed between the data source and the PCTH encoder. This increases the complexity of the transmitter and receiver. - If the PCTH encoder is not fed with i.i.d. binary symbols with uniform distribution, then the number of spectral lines eliminated is reduced.

2.6 Application of Convolutional Codes in IR-Based UWB Systems

As previously mentioned, in most of the IR-based UWB approaches several pulses are normally used to transmit one data symbol. This is because of the restrictions imposed over the maximum allowed transmit power. This repetition scheme increases the total symbol power and thus improves the system's bit error rate (BER) performance.

Previously, references [21, 23, 28] have pointed out that such a pulse repetition scheme can be interpreted as a simple repetition block coded system. As an example of this let us consider a basic PPM TH-IR-based UWB system transmitting N_w pulses per data symbol as defined in Section 2.3. A block diagram of a possible implementation for this system is shown in Figure 2-11.

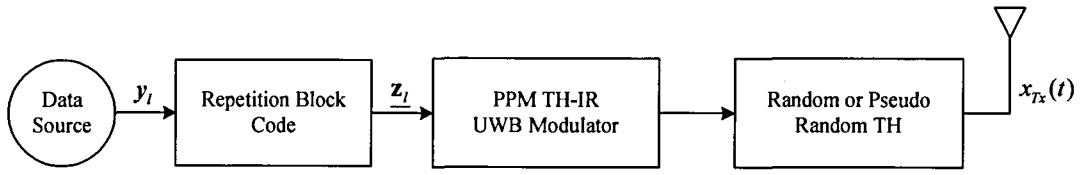


Figure 2-11. Block diagram of an IR-based UWB system where the pulse repetition scheme is interpreted as repetition block code. $\underline{z}_l$ is a code vector with N_w elements all equal to y_l.

In Figure 2-11 the repetition block code is used to map the input data symbol y_l into N_w consecutive pulses. Thus $\underline{z}_l = [y_l, y_l, ..., y_l]$ is a code vector with N_w elements all equal to y_l. Then one pulse is generated for each element of $\underline{z}_l$ and the transmitted signal would be

$$x_{Tx}(t) = \sum_{l=-\infty}^{\infty} \sum_{k=0}^{N_w-1} w_{Tx}(t - lT_s - kT_r - c_{lN_w+k}T_c - y_l T_\beta) \tag{2.27}$$

where $T_s = N_w T_r$ is the input symbol time equal to the code vector time. Note Equation (2.27) defines a typical PPM TH-IR-based UWB signal transmitting N_w pulses per data symbol. Therefore in the following discussion and chapters the terms **"non-coded"** and **"non-convolutionally coded"** IR-based UWB will be used for IR-based UWB systems employing the pulse repetition/repetition block code scheme.

From the coding theory point of view, the repetition block code is not as good as other coding schemes, [83, 100]. Therefore it is possible to improve the system's BER without increasing the transmit power by replacing the "repetition block code" with a better code. One easy way to achieve this is through the use of convolutional codes and in fact this idea has been proposed in several papers in the UWB literature, [21, 22, 24, 25, 27, 30-32]. Thus in the convolutionally coded scheme the repetition block code in Figure 2-11 is replaced by a rate $1/N_w$ convolutional encoder as shown in Figure 2-12.

In a way similar to the repetition block code in Figure 2-11, the rate $1/N_w$ convolutional encoder in Figure 2-12 generates a code vector $\underline{z}_l = [z_l^{(0)}, z_l^{(2)}, ..., z_l^{(N_w-1)}]$ with N_w code symbols, $z_l^{(k)}$, per

input symbol, y_l. However, in this case each element, $z_l^{(k)}$, of $\underline{\mathbf{z}}_l$ is not necessarily equal to the input symbol, y_l. This is explained in Section 2.7. Therefore the transmitted signal now becomes

$$x_{Tx}(t) = \sum_{l=-\infty}^{\infty} \sum_{k=0}^{N_w-1} w_{Tx}(t - lT_s - kT_r - c_{lN_w+k}T_c - z_l^{(k)}T_\beta) \tag{2.28}$$

where again the input symbol time $T_s = N_w T_r$ is equal to the code vector time.

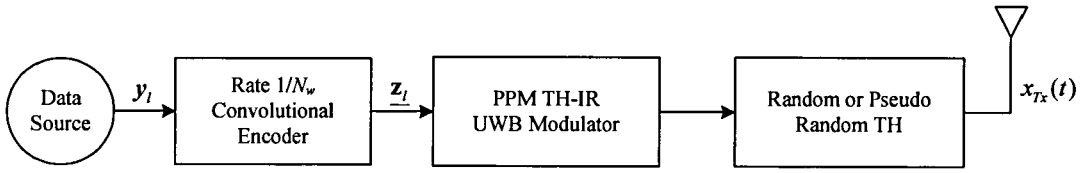
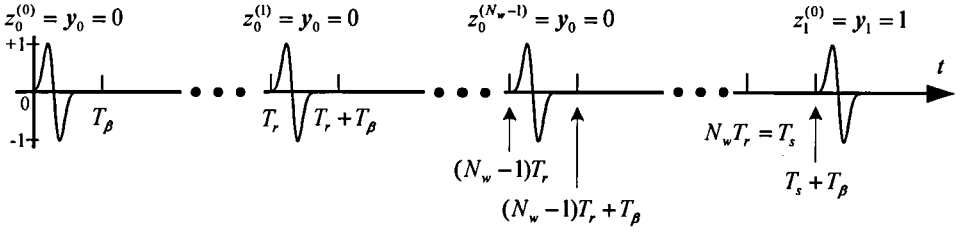
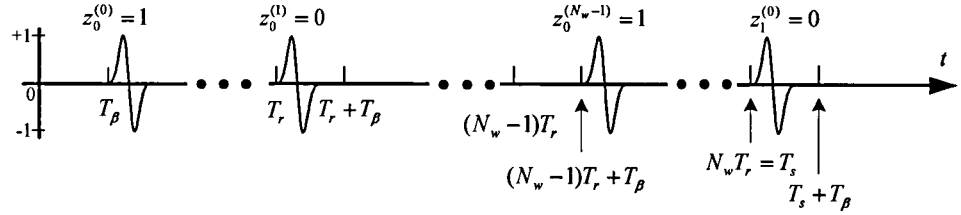


Figure 2-12. Block diagram of a convolutionally coded IR-based UWB system. $\underline{\mathbf{z}}_l$ is a code vector with N_w elements not necessarily equal to y_l.

In Figure 2-13 typical binary PPM TH-IR-based UWB signals for the pulse repetition and convolutionally coded schemes are shown (where for clarity $c_{lN_w+k} = 0$). It can be seen in this figure that the introduction of the convolutional encoder changes the signal mapping used for the transmission of the data stream $\{y_l\}$.



a) Typical signal generated by the repetition block coded system of Figure 2-11 as described by Equation (2.27)



b) Typical signal generated by the convolutionally coded system of Figure 2-12 as described by Equation (2.28)

Figure 2-13. Typical binary PPM TH-IR-based UWB signals for the systems shown in Figures 2-11 and 2-12. For clarity $c_{lN_w+k} = 0$ in these signals.

The authors in [21] have shown that for binary PPM TH-IR-based UWB, a system using convolutional codes can outperform a system using the traditional pulse repetition scheme in terms of BER and/or number of supported users without increasing the transmitted power. Similar results for other modulation schemes have been reported in [22, 24, 25, 27, 31, 32, 35]. Particularly, the use of superorthogonal convolutional codes, [82], has been proposed in [21, 25, 32] for binary PPM IR-based UWB, in [24, 27, 30] for PAM IR-based UWB, and in [35] for PSM IR-based UWB. Well known binary convolutional encoders such as superorthogonal codes, [82], maximum free distance (MFD) codes, [83], and rate compatible punctured convolutional codes, [84], have been

used in [22, 33, 37, 38] for *M*-ary PPM IR-based UWB. Orthogonal convolutional coded modulation schemes for IR-based UWB have been analysed in [31] while a biorthogonal convolutional coded modulated scheme has been proposed in [39]. Finally, the use of turbo codes and turbo-like decoding schemes for IR-based UWB has been addressed in [23, 26, 28, 29, 34, 36]. All these works have shown that the use of convolutional codes and related turbo codes in IR-based UWB systems offer great benefits in terms of BER performance when compared with non-coded pulse repetition schemes. It is worth mentioning that the IEEE 802.15.4a draft specification for Low-Rate Wireless Personal Area Networks considers the use of a rate 1/2 systematic convolutional encoder in its forward error correction (FEC) layer, [40]. As well, the IEEE study group 802.15.3a considered the use of convolutional codes for FEC purposes, [19].

Most of the previously mentioned papers (dealing with the application of convolutional codes in IR-based UWB systems) focus on BER analysis without considering the effects of the convolutional encoder over the signal's PSD. The only two exceptions are the schemes proposed in [33] and [37] where the use of binary MFD encoders combined with *M*-ary PPM is proposed. The scheme in [33] uses constraint length $K = \log_2(M)$, rate $1/K$ MFD binary convolutional encoders with interleaving. Besides, the use of constraint length $K = \log_2(M)+1$, rate $1/\log_2(M)$ MFD binary convolutional encoders without interleaving is proposed in [37]. Both schemes assume uniform distributed i.i.d. data streams at the encoder input. Similar to the PCTH scheme, [70], the output of the interleaver ([33])/encoder ([37]) is set to drive all time hopping without extra PR-TH or DS multiplication. Under these constraints it is shown in [33] and [37] that both schemes provide enhanced PSD characteristics similar to those obtained with PCTH. Nevertheless it is worth noting that the scheme proposed in [33] needs the interleaver, since without it the number of spectral lines in the signal's PSD increases. The spectral analysis presented in [33] was performed using simulation, while [37] uses a simplified model for the signal's spectral analysis which is based on the assumption of uniform distributed i.i.d. streams at the encoder output (that is the correlation introduced by the convolutional encoders is not considered). As well, neither [33] nor [37] consider the use of other variables such as PAM, TH, DS, timing jitter and attenuation. Finally it is important to mention that the binary convolutional encoders used in these works were not specifically designed for *M*-ary signalling. Thus, there exists the possibility of improving the BER while achieving similar PSD characteristics by properly designing binary to *M*-ary convolutional encoders.

A review of basic concepts related to convolutional codes needed for a better understanding of this work will be provided in the next section. Besides the particular system model used in this thesis for convolutional coded IR-based UWB systems is introduced in Chapter 3.

2.7 Convolutional Codes Basics

Convolutional codes are widely used in digital communication systems to improve bit error rates, [83, 100-104]. In effect they are used to protect the transmitted digital data from channel

induced errors. Therefore it is likely that several practical UWB systems will include some sort of convolutional coding for forward error correction (FEC) purposes, [21, 22, 24, 25, 27, 30, 31, 32, 40], as explained in the previous section.

This section introduces some basic definitions and results related to convolutional coding that will be used throughout this thesis.

2.7.1 Basic Representations for Convolutional Encoders

There exist two main parameters related to convolutional encoders. These are the code rate and the total encoder memory. The code rate, υ/κ, defines the number of code symbols, κ, that the encoder generates per υ input symbols. The total encoder memory, ν, is the number of memory elements needed to generate the encoder's output. The total encoder memory provides, as well, a measure of decoding complexity, [100, 101, 104].

A generic diagram for a rate υ/κ convolutional encoder is shown in Figure 2-14. Note the encoder is a sequential circuit consisting of υ inputs, υ shift registers with ν_i memory elements each, κ outputs and several sum points. At each encoder time υ source symbols are fed at once into the encoder and κ code symbols are simultaneously produced through a set of linear operations over the current input and the shift registers' content. The outputs and linear operations are not restricted to be binary, allowing for binary to M-ary encoding operations.

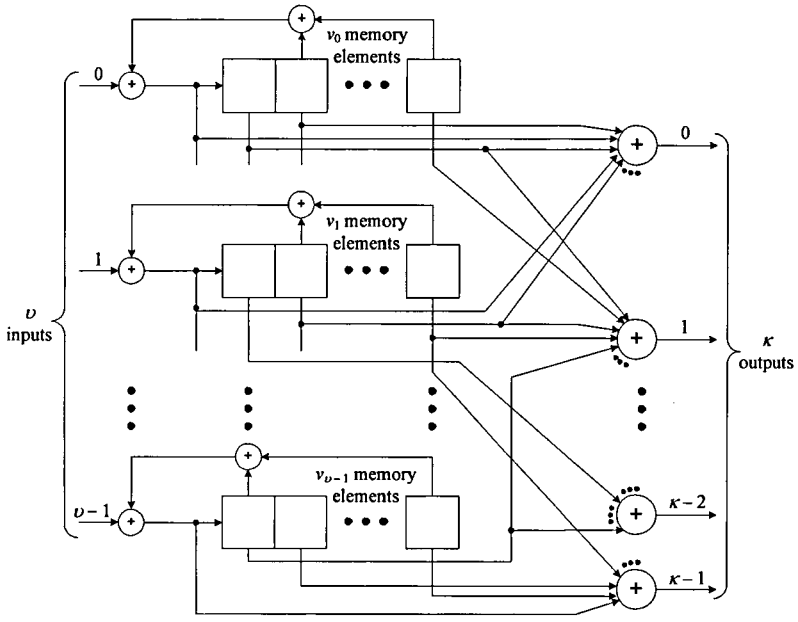
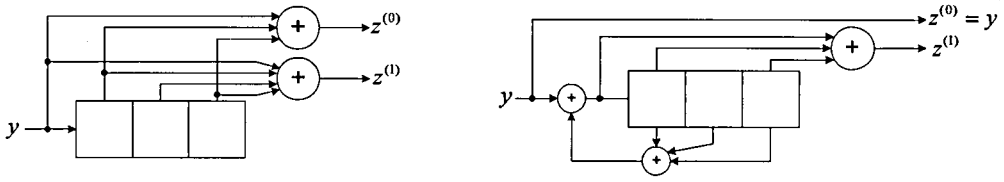


Figure 2-14. Generic diagram of a convolutional encoder.

Note in Figure 2-14 that although all the shift registers have a feedback element this is not mandatory as shown in Figure 2-15a. When no feedback is present the encoder will be referred as a feedforward encoder. When feedback is present then the encoder will be referred as a recursive encoder. If the input sequence appears completely unchanged at the output as in Figure 2-15b then the encoder is referred as a systematic encoder which can be itself feedforward or recursive.



a) Feedforward non-systematic encoder b) Recursive systematic encoder

Figure 2-15. Examples of binary feedforward and recursive systematic convolutional encoders.

Let v be the total number of memory elements used in the encoder, that is

$$v = \sum_{i=0}^{\upsilon-1} v_i \tag{2.29}$$

This is the definition of overall constraint length in [103] and total encoder memory in [100]. The encoder state is usually defined as the content of its v memory elements. Therefore an encoder with binary input symbols can assume up to 2^v different states. Another quantity of interest is the encoder memory (encoder memory order in [100]) defined as

$$m = \max_i v_i \tag{2.30}$$

that is the length of the largest shift register.

The relationship between the encoder's inputs and outputs for a rate υ/κ convolutional encoder can be described by the transfer function matrix, [100], (generator matrix in [103]) defined as

$$\mathbf{G}(D) = \begin{bmatrix} g_0^{(0)}(D) & g_0^{(1)}(D) & \dots & g_0^{(\kappa-1)}(D) \\ g_1^{(0)}(D) & g_1^{(1)}(D) & \dots & g_1^{(\kappa-1)}(D) \\ \vdots & \vdots & & \vdots \\ g_{\upsilon-1}^{(0)}(D) & g_{\upsilon-1}^{(1)}(D) & \dots & g_{\upsilon-1}^{(\kappa-1)}(D) \end{bmatrix} \tag{2.31}$$

where each element, $g_i^{(j)}(D)$, will be referred as a generator. Every single generator has the form

$$g_i^{(j)}(D) = \frac{a_{i,0}^{(j)} + a_{i,1}^{(j)} D + \dots + a_{i,v_i}^{(j)} D^{v_i}}{1 + b_{i,1} D + \dots + b_{i,v_i} D^{v_i}} \tag{2.32}$$

where "D" is the delay operator . As shown in Figure 2-16, the numerator in Equation (2.32) describes the *feedforward* connections from shift register i to output j, and the denominator describes the *feedback* connections of shift register i.

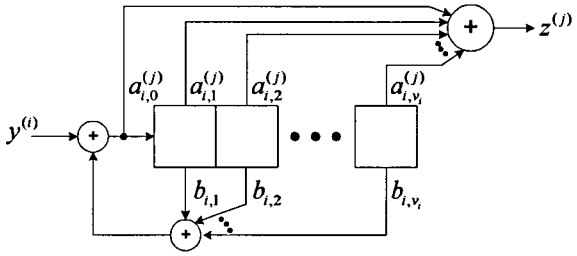


Figure 2-16. Basic structure of a convolutional encoder.

For binary to binary encoders $a_{i,l}^{(j)}$ and $b_{i,l}$ take values from the binary Galois field (GF(2)) with elements {0, 1}. Besides, for binary to M-ary encoders $b_{i,l}$ takes values from GF(2) while $a_{i,l}^{(j)}$

takes values from GF(M), [105, 106]. In the following the binary to binary encoders will be simply referred as binary encoders.

For feedforward encoders the denominator of $g_i^{(j)}(D)$ equals 1 and $\mathbf{G}(D)$ is normally referred as a polynomial generator matrix. In the following the term "rational generator" will be used for generators, $g_i^{(j)}(D)$, with polynomial denominators whereas the term "generator polynomial" will be used for generators with denominator equal to 1. Note for polynomial transfer function matrices (generator matrices) it is customary to represent the generator polynomials in binary or octal notation, e.g. $1+D+D^3 \Rightarrow 1101_2 \Rightarrow 15_8$.

When the input sequence, $y^{(i)} = y_0 y_1 y_2 ... y_k ...$, to shift register i is expressed in terms of the delay operator as

$$y^{(i)}(D) = y_0 + y_1 D + y_2 D^2 + ... + y_k D^k + ... \tag{2.33}$$

the output, $z^{(j)}$, can be obtained by

$$z^{(j)}(D) = y^{(i)}(D) g_i^{(j)}(D) \tag{2.34}$$

assuming $z^{(j)}$ only has connections to shift register i. Therefore the encoding operation for a rate υ/κ convolutional code can be represented as

$$\underline{\mathbf{z}}(D) = \underline{\mathbf{y}}(D)\mathbf{G}(D) \tag{2.35}$$

where $\underline{\mathbf{y}}(D) = [y^{(0)}(D), y^{(1)}(D), ..., y^{(\upsilon-1)}(D)]$ and $\underline{\mathbf{z}}(D) = [z^{(0)}(D), z^{(1)}(D), ..., z^{(\kappa-1)}(D)]$. Note $\mathbf{G}(D)$ defines a one-to-one linear mapping from the input (information) sequence represented by $\underline{\mathbf{y}}(D)$, to the code sequence represented by $\underline{\mathbf{z}}(D)$, [85, 100, 102, 103].

It is important to recall that the *convolutional code* is the set of all possible code sequences (codewords) generated from the linear mapping represented by Equation (2.35). Besides, the *convolutional encoder* is the sequential circuit realization of the linear mapping with specific transfer function matrix (generator matrix) $\mathbf{G}(D)$. Therefore different transfer function matrices (generator matrices) with different input (information) to codeword mappings can generate the same convolutional code, [85, 103].

There exist other description methods for convolutional codes such as the binary/M-ary generator matrices, tree diagrams, trellis diagrams and state transition diagrams, [83, 100-103]; however it is necessary to clarify the use of some of these terms throughout this thesis. For instance the term "generator matrix" is used in [102, 103] to designate both the "binary/M-ary generator matrix" and the "transfer function matrix" as defined in [100, 101, 107]. In order to avoid confusion, the term "binary/M-ary generator matrix" will be explicitly used for matrix representations which do not use the delay operator "D" and whose elements are taken from a binary or M-ary alphabet. Besides, the term "transfer function matrix" and "rational/polynomial generator matrix" will be used for matrices whose elements consists of rational or polynomial expressions of the form defined in Equation (2.32). The term "total encoder memory" will be used to refer v as defined in Equation (2.29) instead of "overall constraint length". Similarly the term

"encoder memory" will be used to designate m in Equation (2.30) instead of the term "encoder memory order". Finally, for rate $1/\kappa$ encoders the constraint length, K, will be defined as

$$K = v+1 = m+1 \tag{2.36}$$

The use of other terms will be set during the subsequent discussion.

2.7.2 Schematic Representations for Convolutional Encoders

Two schematics representations will be of interest for the rest of this thesis: the state transition diagrams and the trellis diagram.

As previously mentioned, a convolutional encoder is a sequential circuit with c inputs, κ outputs and v memory elements. The *state transition diagram* is thus obtained, [83, 100-103], by defining the encoder state as the content of its v memory elements with state transitions caused by the current input vector (consisting of υ input symbols), $\underline{\mathbf{y}}_l = [y_l^{(0)}, y_l^{(1)}, ..., y_l^{(\upsilon-1)}]$, where l is the time index. Therefore the state transition diagram will have 2^v states and there will be 2^υ state transitions (branches) from/to each state. Note the current encoder output vector (consisting of κ code symbols), $\underline{\mathbf{z}}_l = [z_l^{(0)}, z_l^{(1)}, ..., z_l^{(\kappa-1)}]$, is a function of both the current encoder state and the current input. As an example the state transition diagram of the binary encoder depicted in Figure 2-15a, with transfer function matrix given by

$$\mathbf{G}(D) = [1+D+D^3 \quad 1+D+D^2+D^3] \tag{2.37}$$

(or $(15, 17)_8$ in octal form), is shown in Figure 2-17. The states are labelled with the possible binary content of the 3 memory cells where by convention, and unless otherwise stated, the content of the leftmost memory cell is assumed to be most significant bit (MSB) of the binary representation. Each state transition (branch) is labelled following the format $y/z^{(0)}z^{(1)}$, indicating the input causing the transition altogether with its respective output. Note from the state transition diagram and Figure 2-14 that a convolutional encoder is in essence a *finite state sequential machine* (FSSM), [101], as will be further shown in Chapter 3.

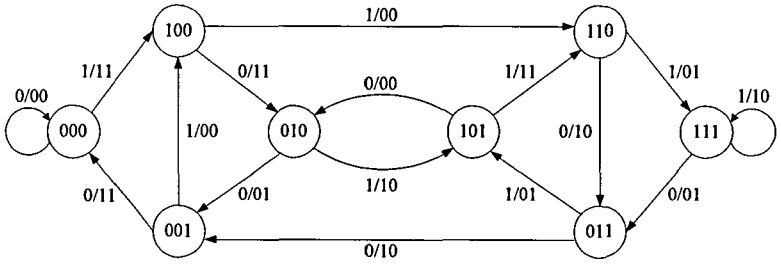


Figure 2-17. State transition diagram for the rate 1/2, $v = m = 3$, binary feedforward encoder $(15, 17)_8$.

The second graphical representation of interest for this work is that of the *trellis diagrams.* A trellis diagram can be interpreted as an expansion of the state transition diagram over time. Therefore, the trellis diagrams provide a graphical representation of the paths defined by the state transitions as produced when specific input sequences are fed to the encoder. For every path a specific codeword is generated and therefore each state transition (branch) has an encoder output

vector (consisting of κ code symbols) associated to it. As an example the trellis diagram corresponding to the encoder defined by Equation (2.37) is shown in Figure 2-18. In this figure the states are labelled as $q_0 \leftrightarrow (000)$, $q_1 \leftrightarrow (001)$,..., $q_7 \leftrightarrow (111)$, where again the content of the leftmost memory cell is regarded to be the MSB. As well, each branch is labelled with the input causing the transition and the output produced by such transition as $y/z^{(0)}z^{(1)}$

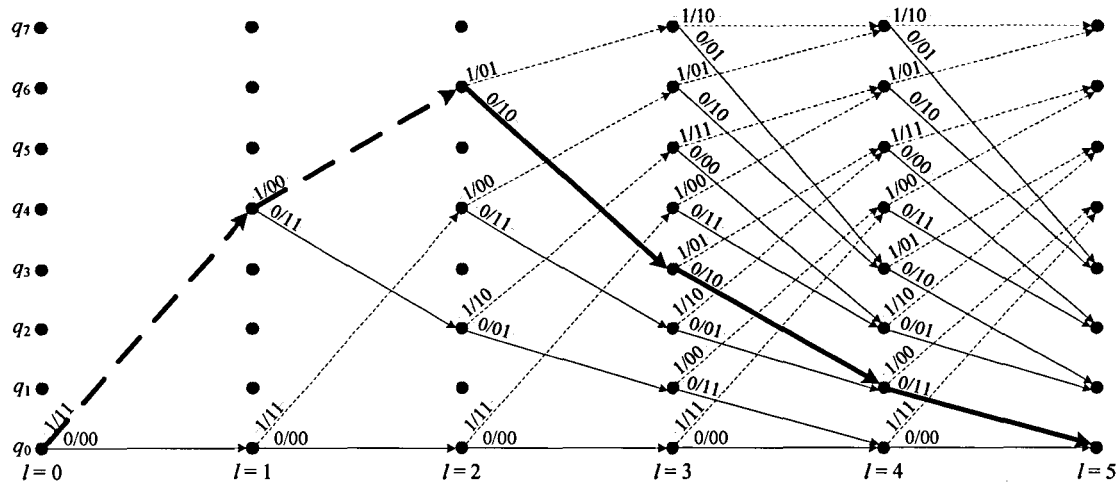


Figure 2-18. Trellis diagram for the rate 1/2, $v = m = 3$, binary feedforward encoder $(15, 17)_8$.

In Figure 2-18 it is assumed that the encoder starts in state q_0, and therefore only transitions emanating from this node are possible at time $l = 0$. By time $l = m = 3$ branches are emanating from all nodes with two branches coming out and two branches going into each node by time $l = 4$. As previously mentioned, a particular input sequence will define a specific path through the trellis and thus a specific sequence of code symbols (formed by several specific encoder output vectors) will be generated. Hence, following a specific path through the trellis will define a specific *codeword*. For example, the input sequence $y = \{1,1,0,0,0,...\}$ defines the path highlighted with bold lines in Figure 2-18 and the code sequence (codeword) $z = \{11,00,10,10,11,...\}$ is generated.

2.7.3 Performance Related Parameters for Convolutional Codes

Before introducing the performance parameters for convolutional codes relevant to this thesis, it is necessary to define the Hamming weight of a sequence and the Hamming distance between sequences. A sequence's *Hamming weight* is defined as the number of non-zero symbols in such sequence. In the case of binary sequences the Hamming weight is just the number of "1's" in the sequence. The *Hamming distance* between two sequences of the same length is defined as the number of positions in which both sequences differ. It is important to mention that for linear codes, such as the convolutional codes, the Hamming distance between codewords is effectively a metric, [100, 103].

There exists several distance measures for convolutional codes/encoders such as the column distance, the distance profile, the minimum distance, the free distance and the row distance, [100, 101, 103]. Of all these measures the most important for the purposes of this thesis is the code *free*

distance, d_{free}. This is due to d_{free} is one of the most important performance parameters when using convolutional codes with binary antipodal or *M*-ary orthogonal signalling and maximum likelihood decoding (such as that achieved with the Viterbi algorithm), [83, 100-103, 105-107]. The free distance of a convolutional code is defined as the minimum Hamming distance between any two codewords, that is:

$$d_{free} = \min\{d_{Hamm}(\mathbf{z},\mathbf{z}') : \mathbf{y} \neq \mathbf{y}'\} \tag{2.38}$$

where $\mathbf{z}$ and $\mathbf{z}'$ are two valid codewords (code sequences) corresponding to the information sequences $\mathbf{y}$ and $\mathbf{y}'$; and d_{Hamm} stands for Hamming distance. It is important to mention that due to the convolutional codes are linear codes, the free distance of a convolutional code is equivalent to the minimum Hamming weight of any valid codeword generated by a path starting and finishing in the zero state of the trellis. For reasons that will become evident in the following chapters, the term d_{free} will be exclusively used to refer the free distance of binary convolutional codes. Besides, for binary to *M*-ary convolutional codes the term d_{Mfree} will be used. This is due to for this kind of codes the free distance is defined in terms of the *M*-ary Hamming weight between codewords.

There is an infinite number of valid codewords (code sequences) in a convolutional code generated by trellis paths which diverge from the zero state and the all-zeros path at some point and later remerge to them without leaving again, [83, 100-103]. Therefore there may be more than one of such paths generating a codeword with Hamming weight equal to d_{free}/d_{Mfree} and an infinite number of such paths generating codewords with Hamming weight larger than d_{free}/d_{Mfree}. Counting the number of such paths producing codewords with specific Hamming weight $d_{free}+i$ defines the sequence

$$\{A_{d_{free}+i}\}, \quad i = 0,1,2,... \tag{2.39}$$

which is called the *distance spectrum* of the code. Besides, adding the total Hamming weight of all information sequences (that is the total number of nonzero information symbols) driving paths with codeword Hamming weight equal to $d_{free}+i$ defines another sequence

$$\{B_{d_{free}+i}\}, \quad i = 0,1,2,... \tag{2.40}$$

commonly called *information weight spectrum* (IWS). It is worth mentioning that while the free distance is code specific, the distance spectrum and information weight spectrum is encoder specific, [100, 103].

The free distance, distance spectrum and information weight spectrum are important because they can be used to obtain upper bounds for the first event error probability and bit error rates achievable with specific convolutional encoders, [83, 100-103, 105-107]. Therefore these distance parameters can be effectively used as metrics to compare different encoders.

An important characteristic of convolutional encoders is that the free distance, distance spectrum, information weight spectrum and related distance properties do not change when interchanging the order of the columns in the transfer function matrix $\mathbf{G}(D)$, [83, 100-103, 105, 107]. This can be readily inferred from Figures 2-14, 2-17 and 2-18 where it can be seen that

changing the output vector order from $\underline{\mathbf{z}}_l = [z_l^{(0)}, z_l^{(1)}, \ldots, z_l^{(\kappa-1)}]$ to $\underline{\mathbf{z}}_l = [z_l^{(\kappa-1)}, z_l^{(2)}, \ldots, z_l^{(\kappa-3)}]$ (or any other permutation) does not affect its Hamming weight. As consequence the output Hamming weight associated to every branch is not altered by interchanging columns in $\mathbf{G}(D)$.

Calculation of the free distance, distance spectrum and information weight spectrum can be done analytically by splitting the zero state of the state transition diagram into an "initial" and "final" states and then computing the generating function of the resulting signal flow graph, [83, 100-103]. Nevertheless, excluding some particular cases such as Viterbi's orthogonal and superorthogonal codes, [82], the signal flow graph approach becomes impractical for convolutional encoders with medium to large total encoder memory, v. Therefore, several computer algorithms for free distance, distance spectrum and information weight spectrum calculation have been developed over the years, [100, 103, 108]. In particular the FAST algorithm proposed by Cedervall and Johannesson, [103, 108], is the most relevant for the purposes of this thesis.

Finally it is important to mention that there exist convolutional encoders that generate an output sequence with finite Hamming weight even when the input sequence has infinite Hamming weight. Such encoders are called *catastrophic* due to a finite number of channel errors can cause an infinite number of decoding errors, [83, 100-103]. An easy way to identify a catastrophic encoder is through its state transition diagram which will have a closed loop (not necessarily to the same state) with non-zero input Hamming weight and zero output Hamming weight [83, 100, 101].

2.7.4 Union Bounds for Error Probabilities Achieved with Convolutional Coding

As previously mentioned the free distance, distance spectrum and information weight spectrum are used in the calculation of upper bounds for the bit and first event error probabilities achieved when using specific convolutional encoders. As well the kind of modulation (signalling) used by the system and the particular channel will affect these error probabilities. This work mainly deals with binary antipodal signalling, quaternary biorthogonal signalling and M-ary orthogonal signalling (such as orthogonal PPM). Therefore an upper bound on the bit error probability achieved with binary and binary to M-ary convolutional encoders for output symmetric memoryless channels can be obtained as ([83, 100-103, 105-107]),

$$P_b \leq \sum_{d=d_{free}}^{\infty} B_d P_2(d) \tag{2.41}$$

where $P_2(d)$ is the pairwise error probability between two codewords (code sequences) with Hamming distance d (that is two codewords differing in exactly d positions). Similarly, the code first event error probability is upper bounded by

$$P_e \leq \sum_{d=d_{free}}^{\infty} A_d P_2(d) \tag{2.42}$$

The form acquired by $P_2(d)$ will depend on the kind of signalling and detector used in the system

and its evaluation for each particular case relevant for this thesis will be addressed in the following chapters.

From Equation (2.41) it is readily seen that for convolutional encoders with the same rate and total encoder memory, v, a good design strategy consists of finding the encoder with the largest free distance and minimum $\{B_{d_{free}+i}\}$. In fact maximising the free distance and optimizing the information weight spectrum has been the main criteria used in works dealing with the search of good convolutional encoders such as [105-107, 109-112]. These encoders are usually referred as maximum free distance (MFD) and optimum distance spectrum (ODS) codes.

2.7.5 Branch Metrics and Maximum Likelihood Decoding for Convolutional Codes

In this work it is assumed that the channel used for the transmission is memoryless and that decoding is performed by means of the Viterbi algorithm. These assumptions are made to enable a straightforward BER performance comparison between the new convolutional codes reported in this thesis and the BER performance of convolutional codes previously reported in the literature, [83, 100-103, 105-107, 109-112]. Note the Viterbi algorithm is a maximum likelihood (ML) decoding technique, [83, 100-103].

Consider the system presented in Figure 2-19 where the data source produces a length N information sequence $\mathbf{y}=\{\underline{\mathbf{y}}_0,\underline{\mathbf{y}}_1,...,\underline{\mathbf{y}}_{N-1}\}=\{y_0^{(0)},...,y_0^{(\upsilon-1)},...,y_{N-1}^{(0)},...,y_{N-1}^{(\upsilon-1)}\}$ which defines a path generating the code sequence $\mathbf{z}=\{\underline{\mathbf{z}}_0,\underline{\mathbf{z}}_1,...,\underline{\mathbf{z}}_{N-1}\}=\{z_0^{(0)},...,z_0^{(\kappa-1)},...,z_{N-1}^{(0)},...,z_{N-1}^{(\kappa-1)}\}$ when fed to the convolutional encoder. Next $\mathbf{z}$ is transmitted over a memoryless noisy channel (using some specific modulation format) and demodulated to produce the received sequence $\mathbf{r}=\{\underline{\mathbf{r}}_0,\underline{\mathbf{r}}_1,...,\underline{\mathbf{r}}_{N-1}\}=\{r_0^{(0)},...,r_0^{(\kappa-1)},...,r_{N-1}^{(0)},...,r_{N-1}^{(\kappa-1)}\}$. The received sequence, $\mathbf{r}$, is then used by the Viterbi decoder to produce an estimate, $\mathbf{z}'$, of the transmitted sequence $\mathbf{z}$ and an estimate, $\mathbf{y}'$, of the information sequence, $\mathbf{y}$. Since the Viterbi algorithm is a ML decoder it selects $\mathbf{z}'$ such that the conditional probability density function (probability mass function for discrete channels) $g(\mathbf{r}\,|\,\mathbf{z}')$ is maximized, [83, 100, 101]. As the channel is assumed to be memoryless we have

$$g(\mathbf{r}\,|\,\mathbf{z}')=\prod_{i=0}^{N-1}g(\underline{\mathbf{r}}_i\,|\,\underline{\mathbf{z}}'_i)=\prod_{i=0}^{N-1}\prod_{j=0}^{\kappa-1}g(r_i^{(j)}\,|\,z_i'^{(j)}) \tag{2.43}$$

or taking the logarithm

$$\log(g(\mathbf{r}\,|\,\mathbf{z}'))=\sum_{i=0}^{N-1}\log(g(\underline{\mathbf{r}}_i\,|\,\underline{\mathbf{z}}'_i))=\sum_{i=0}^{N-1}\sum_{j=0}^{\kappa-1}\log(g(r_i^{(j)}\,|\,z_i'^{(j)})) \tag{2.44}$$

which is known as the log likelihood function, [83, 100, 101]. Note maximizing Equation (2.43) is equivalent to maximizing Equation (2.44).

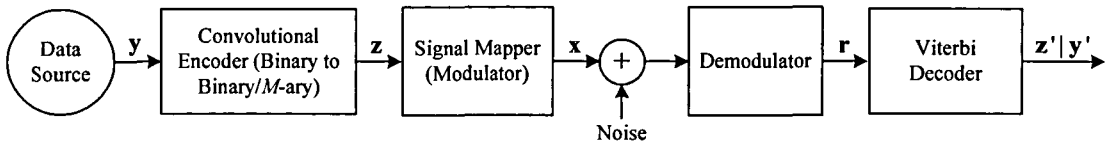


Figure 2-19. Generic block diagram of a convolutionally coded system.

Since $\log(g(\mathbf{r}|\mathbf{z}'))$ depends on $\mathbf{z}'$ and $\mathbf{z}'$ itself defines a specific path in the trellis, the quantity $\log(g(\mathbf{r}|\mathbf{z}')) = PM(\mathbf{r}|\mathbf{z}')$ is referred as a *path metric*. Note $\underline{\mathbf{z}'_i}$ is an encoder output vector associated to a specific branch in the path related to $\mathbf{z}'$, therefore $\log(g(\underline{\mathbf{r}_i}|\underline{\mathbf{z}'_i})) = BM(\underline{\mathbf{r}_i}|\underline{\mathbf{z}'_i})$ is usually called *branch metric*. Similarly the terms $\log(g(r_i^{(j)}|z_i'^{(j)})) = SM(r_i^{(j)}|z_i'^{(j)})$ are normally referred as symbol metrics. Using this notation a partial path metric at time l in the trellis can be defined for $\mathbf{z}'$ as

$$PM_l(\mathbf{r}|\mathbf{z}') = \sum_{i=0}^{l} BM(\underline{\mathbf{r}_i}|\underline{\mathbf{z}'_i}) \tag{2.45}$$

that is, $PM_l(\mathbf{r}|\mathbf{z}')$ is the partial path metric accumulated when transversing the first l branches of path $\mathbf{z}'$.

The concept of path metric, branch metric, symbol metric and partial branch metric are important when performing ML through the Viterbi algorithm. In short the Viterbi algorithm can be resumed as follows ([83, 100, 101]):

1. Assuming the encoder starts in the zero state, q_0, at time $l = 0$, set the metric in this state to be equal to 0.
2. For every state (node) in the trellis at time $l + 1$, calculate the partial path metrics for every branch entering the node by adding the corresponding branch metric, $BM(\underline{\mathbf{r}_{l+1}}|\underline{\mathbf{z}'_{l+1}})$, to the respective surviving partial path metric, $PM_l(\mathbf{r}|\mathbf{z}')$, assigned to each state at time l.
3. Compare all the path metrics entering a given state and store the path with the largest metric, altogether with the respective path metric, while deleting the other paths. Do this for all states in the trellis such that every state will be assigned a surviving partial path and partial path metric, $PM_{l+1}(\mathbf{r}|\mathbf{z}')$. Ties are solved by selecting a path at random.
4. If the end of the trellis has not been reached, increase l and go back to step 2. Otherwise find the state with the largest metric and select its respective path and associated code sequence as the decoded codeword. The information sequence, $\mathbf{y}'$, associated to the decoded codeword, $\mathbf{z}'$, is then the decoded information sequence.

The form acquired by $BM(\underline{\mathbf{r}_i}|\underline{\mathbf{z}'_i})$ depends on the way the received sequence $\mathbf{r}$ is fed to the Viterbi decoder. Furthermore, the selection of an adequate metric is important because of the form acquired by the pairwise error probability $P_2(d)$. Hence, the bit error probability depends on the symbol and branch metrics used, [83, 100, 101]. As an example consider a binary convolutional encoder is used in a system employing binary antipodal signalling over an additive white Gaussian

noise (AWGN) channel with two sided power spectral density $N_0/2$. If the demodulator is able to provide soft decisions to the decoder, that is

$$r_i^{(j)} = x_i^{(j)} + n_i^{(j)} = \sqrt{E_s}(2z_i^{(j)} - 1) + n_i^{(j)} \tag{2.46}$$

where $\sqrt{E_s}$ is the signal energy and $n_i^{(j)}$ is a Gaussian random variable with variance $\sigma^2 = N_0/2$, then

$$g(\mathbf{r} \mid \mathbf{z}') = \prod_{i=0}^{N-1} \prod_{j=0}^{\kappa-1} \frac{1}{\sqrt{\pi N_0}} e^{-(r_i^{(j)} - \sqrt{E_s}(2z_i'^{(j)} - 1))^2 / N_0} \tag{2.47}$$

and

$$\ln(g(\mathbf{r} \mid \mathbf{z}')) = \sum_{i=0}^{N-1} \sum_{j=0}^{\kappa-1} \left\{ -\ln(\sqrt{\pi N_0}) - (r_i^{(j)} - \sqrt{E_s}(2z_i'^{(j)} - 1))^2 / N_0 \right\} \tag{2.48}$$

By discarding terms common to all paths in Equation (2.48) we obtain

$$PM(\mathbf{r} \mid \mathbf{z}') = \sum_{i=0}^{N-1} \sum_{j=0}^{\kappa-1} r_i^{(j)}(2z_i'^{(j)} - 1) \tag{2.49}$$

and therefore the branch metric is given by

$$BM(\underline{\mathbf{r}}_i \mid \underline{\mathbf{z}}'_i) = \sum_{j=0}^{\kappa-1} r_i^{(j)}(2z_i'^{(j)} - 1) \tag{2.50}$$

When using Equation (2.50) as the branch metric in the Viterbi algorithm the decoder is usually referred as *soft Viterbi* decoder (SVD). Note maximising Equations (2.48) and (2.49) is equivalent to finding the path/codeword, $\mathbf{z}'$, which minimises the squared Euclidean distance,

$$d_{Euc}(\mathbf{r}, \mathbf{x}')^2 = \sum_{i=0}^{N-1} \sum_{j=0}^{\kappa-1} (r_i^{(j)} - \sqrt{E_s}(2z_i'^{(j)} - 1))^2 \tag{2.51}$$

If instead of providing soft decisions, the decoder makes a "hard" decision on the received symbol before feeding the Viterbi decoder, then the channel becomes a binary symmetric channel (BSC) with transition probabilities

$$\Pr\{r_i^{(j)} \neq z_i^{(j)}\} = p \quad \text{and} \quad \Pr\{r_i^{(j)} = z_i^{(j)}\} = 1 - p \tag{2.52}$$

and thus ([83, 100, 101])

$$\log(g(\mathbf{r} \mid \mathbf{z}')) = \log(\Pr[\mathbf{r} \mid \mathbf{z}']) = d_{Hamm}(\mathbf{r}, \mathbf{z}') \log \frac{p}{1-p} + N \log(1-p) \tag{2.53}$$

If $p < 1/2$, then $\log(p/(1-p)) < 0$ and maximising Equation (2.53) is equivalent to minimising the Hamming distance. If the Hamming distance is used as the path metric, then the Viterbi algorithm must be modified such that the paths with the smallest partial Hamming distances are saved while discarding the rest. Therefore the branch and partial path metrics are given by

$$BM(\underline{\mathbf{r}}_i \mid \underline{\mathbf{z}}'_i) = \sum_{j=0}^{\kappa-1} d_{Hamm}(r_i^{(j)}, z_i'^{(j)}) \quad \text{and} \quad PM_l(\mathbf{r} \mid \mathbf{z}') = d_{Hamm}(\mathbf{r}, \mathbf{z}') = \sum_{i=0}^{l} BM(\underline{\mathbf{r}}_i \mid \underline{\mathbf{z}}'_i) \tag{2.54}$$

respectively and the *hard Viterbi* decoder (HVD) must find the path, $\mathbf{z}'$, with the smallest Hamming distance to the received vector $\mathbf{r}$.

2.7.6 Representation Format for Binary to *M*-ary Convolutional Encoders for *M*-ary Orthogonal Signalling

It will be useful to define the representation format used for binary to *M*-ary encoders. Unless otherwise stated, it will be assumed that the binary to *M*-ary convolutional encoders considered in this thesis will be used with *M*-ary orthogonal signalling. As well for these encoders the rate will be defined in terms of the number of *M*-ary symbols produced per each encoder input. For clarity assume $M = 2^n$ and feedforward encoders. Then there exist two basic representations (implementations) for binary to *M*-ary convolutional encoders as depicted in Figures 2-20 and 2-21 for rate 1 encoders.

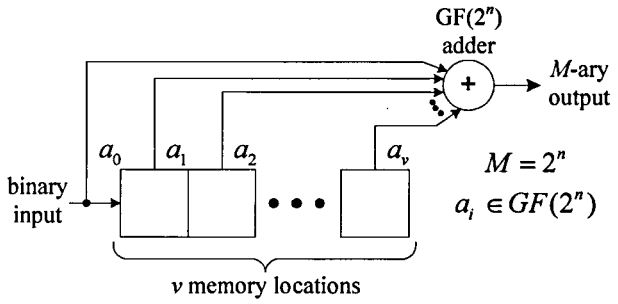


Figure 2-20. Representation over GF(2^n) for rate 1 binary to *M*-ary convolutional encoders.

The binary implementation shown in Figure 2-21 has been used in [107] while the representation over GF(2^n) shown in Figure 2-20 has been used in [105]. Formulas to obtain the representation over GF(2^n) from the binary one are given in [106]. Note the representation over GF(2^n) requires the definition of a primitive polynomial to obtain the code symbols. Therefore in this work the representation depicted in Figure 2-21 will be adopted.

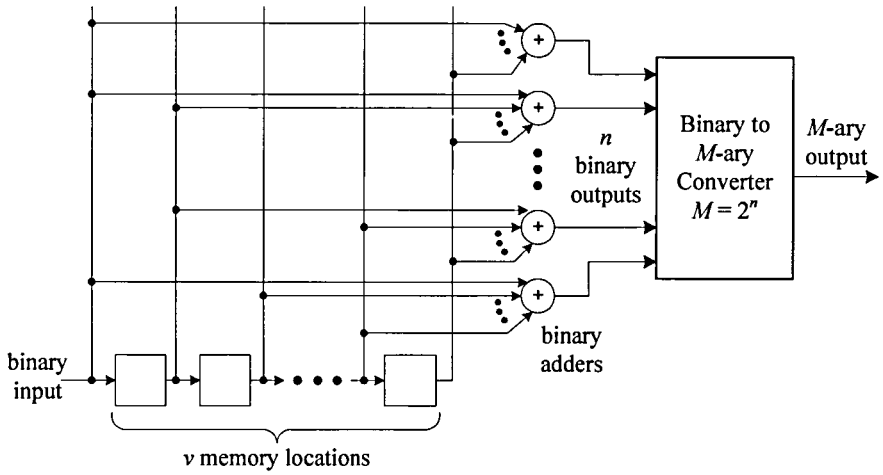


Figure 2-21. Binary representation for rate 1 binary to *M*-ary convolutional encoders.

A binary representation example for a rate 1/2, $v = 3$, binary to 8-ary convolutional encoder is shown in Figure 2-22. This figure illustrates how an *M*-ary output (in this case $M = 8$) is generated from the current memory content and the current input. For example, the binary value of the first set of binary outputs, $\{z_0^{(0)}, z_1^{(0)}, z_2^{(0)}\}$, is calculated as

$$z_0^{(0)} = y \oplus \theta_1 \oplus \theta_2 \oplus \theta_3 = 1 \oplus 1 \oplus 0 \oplus 1 = 1;$$
$$z_1^{(0)} = \theta_2 \oplus \theta_3 = 0 \oplus 1 = 1; \quad (2.55)$$
$$z_2^{(0)} = y \oplus \theta_2 \oplus \theta_3 = 1 \oplus 0 \oplus 1 = 0$$

where $\oplus$ stands for modulo 2 addition. Once the binary value of $\{z_0^{(0)}, z_1^{(0)}, z_2^{(0)}\}$ has been calculated, the corresponding 8-ary output is obtained from

$$z^{(0)} = 4z_0^{(0)} + 2z_1^{(0)} + z_2^{(0)} = 4(1) + 2(1) + 0 = 6 \quad (2.56)$$

as indicated in Figure 2-22. A similar process is followed to calculate the second 8-ary output, $z^{(1)}$.

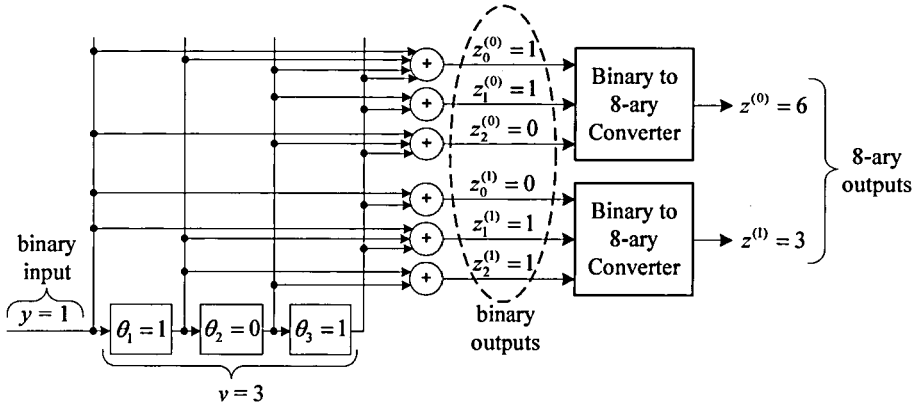


Figure 2-22. Generation of two 8-ary symbols from the current input and memory content of a rate 1/2, $v = 3$, binary to 8-ary convolutional encoder.

Note that when using the binary representation of Figure 2-21 a set of $n = \log_2(M)$ generator polynomials (or rational generators for recursive encoders) over GF(2) must be given to describe the encoder. Similarly for a rate $1/\kappa$ binary to M-ary convolutional code, κ sets of $n = \log_2(M)$ generators must be provided (e.g. 2 sets of 3 generators polynomials each must be provided to describe the encoder shown in Figure 2-22). However, care should be taken in how to interpret and handle the generators sets as they should not be regarded as simple binary convolutional encoders coupled to an M-ary orthogonal modulation scheme.

For instance, it may be tempting to try to use the generators of a rate $1/n$ maximum free distance (MFD) binary convolutional encoder to construct the generator set of a rate 1 binary to 2^n-ary convolutional encoder. Nevertheless this does not necessarily mean that a good rate 1 binary to 2^n-ary convolutional encoder will be obtained with such procedure. In order to see this remember that for M-ary orthogonal signalling the Euclidean distance between every pair of signals is the same regardless of its label, [83, 86]. Therefore, for binary to M-ary convolutional codes used with M-ary orthogonal signalling, the squared Euclidean distance between two signals corresponding to two different M-ary code symbols is proportional to its M-ary Hamming distance. Besides, the squared Euclidean distance is not proportional to the binary Hamming distance as obtained from the binary representations of the signal/symbol labels. As an example consider the quaternary orthogonal signal set expressed in vector form as

$$\mathbf{x}_0 = [\sqrt{E_{si}}, 0, 0, 0]; \quad \mathbf{x}_1 = [0, \sqrt{E_{si}}, 0, 0]; \quad \mathbf{x}_2 = [0, 0, \sqrt{E_{si}}, 0]; \quad \mathbf{x}_3 = [0, 0, 0, \sqrt{E_{si}}] \quad (2.57)$$

where E_{si} is the signal energy. Next assign binary labels as

$$\mathbf{x}_0 \rightarrow 00; \quad \mathbf{x}_1 \rightarrow 01; \quad \mathbf{x}_2 \rightarrow 10; \quad \mathbf{x}_3 \rightarrow 11 \tag{2.58}$$

Therefore the binary Hamming distance between the labels of $\mathbf{x}_0$ and $\mathbf{x}_3$ is 2 while for $\mathbf{x}_0$ and $\mathbf{x}_2$ is 1. Nevertheless the squared Euclidean distance between $\mathbf{x}_0$ and $\mathbf{x}_3$ is identical to the squared Euclidean distance between $\mathbf{x}_0$ and $\mathbf{x}_1$, that is, $2E_{si}$. As a consequence the binary Hamming distance is not proportional to the squared Euclidean distance. In contrast the *M*-ary Hamming distance between every pair of signal/symbol labels is equal to 1, and this is in turn proportional to the squared Euclidean distance between every pair of signals. This is the reason why the design goal in [105-107] was the optimization of the *M*-ary distance properties.

2.8 Conclusions for Chapter 2

In this chapter some useful theoretical background has been presented. Moreover, basic terminology and definitions extensively used in the rest of this thesis for the areas of impulse radio (IR) based ultra wideband (UWB) and convolutional coding have been introduced.

Firstly the UWB definition and current related regulations were reviewed. Then the basic IR-based UWB approaches adopted in this thesis were stated. A general explanation of time hopping (TH) IR-based UWB, direct sequence (DS) IR-based UWB and TH-DS IR-based UWB was presented.

Several important issues can be highlighted for the UWB systems introduced in this chapter. In particular it was established that UWB systems are not confined to operate in a specific narrow frequency band. Instead UWB emissions occupy "ultra" wide portions of the frequency spectrum as far as compliance with well established average and peak emission limits in the form of spectral masks is maintained. Thus UWB emissions are power limited over an ultra-wide frequency band. Therefore one of the main issues for proper UWB operation is the spectral shaping of the transmitted signal. Bearing this in mind the power spectral density of non-coded TH-IR-based UWB signals was introduced in Section 2.4 while the main spectral shaping mechanisms for IR-based UWB signals were introduced in Section 2.5.

In Section 2.6 a literature review of the use of convolutional codes in UWB systems has been provided. It was pointed out that convolutional encoders can improve the BER performance of IR-based UWB systems without increasing the transmitted power of the signal. For example, coding gains of more than 2 dB for bit error rates below 10^{-3} have been reported in references [22, 24, 27, 31, 35, 39]. It is worth mentioning that traditional UWB spectral analysis and most UWB spectral shaping mechanisms do not consider the spectral shaping effects caused by the introduction of a convolutional coding operation in a typical IR-based UWB system. Conversely most of the work done in convolutional coding applied to UWB does not consider the signal's PSD. The only exceptions to this are the PCTH and related schemes which focus on the particular case of *M*-ary PPM TH-IR-based UWB signals where all the time hopping is driven by the encoder.

After the introduction of UWB related issues, a review of basic concepts related to convolutional coding was provided. Basic analytical and schematic representations and a review of the main performance parameters for convolutional encoders were presented. Union bounds for error probabilities and a review of branch metrics and Viterbi decoding were provided. Finally, the representation format used in this thesis for binary to M-ary convolutional encoders was defined. However, coding and more specifically convolutional coding are large topics and it has been assumed that the reader will have some background in this area. More complete details on convolutional coding can be found in [83, 100-103].

Chapter 3

Power Spectral Density of Convolutionally Coded Impulse Radio Based Ultra Wideband Signals

3.1 Introduction

The contribution of this chapter consists of the spectral analysis of convolutionally coded/Markov-driven impulse radio (IR) based ultra wideband (UWB) signals, which is novel. The model used in the analysis and the closed form PSD expressions obtained from it cover different modulation schemes such as pulse amplitude modulation (PAM), binary phase shift keying (BPSK), pulse position modulation (PPM), [8], biorthogonal PPM (BOPPM), PAM/PPM, [51, 113], and pulse shape modulation (PSM), [14, 64, 65], all combined with pseudo-random (periodic)/random time hopping (TH) and/or pseudo-random/random direct sequence (DS) spread spectrum. In addition the signal model and PSD expressions account for the inclusion of attenuation and/or timing jitter.

As mentioned in Chapter 1 and Section 2.2, the current regulations intended for UWB are defined in terms of spectral masks which limit the maximum amount of transmit power over a given frequency range, [2, 4, 5]. Therefore, the analysis estimation, and shaping of the power spectral density (PSD) of UWB signals is a topic of major interest in the design of compliant UWB systems. In this chapter the power spectral density of convolutionally coded/Markov-driven IR-based UWB signals is obtained. This matter is important due to the likely introduction of error correction mechanisms in UWB systems, [21-40], which will have an effect over the PSD of the transmitted signal as shown later in this and subsequent chapters.

The spectral analysis approach presented in this chapter starts with the derivation of a finite state sequential machine (FSSM)-Markov model for the source-encoder pair. Therefore the results presented in spectral analysis of Markov-driven signals in [52, 99, 114] and on several digital communication books, e.g. [83, 87, 115], should be regarded as previous work. Nevertheless, these results are based on the assumption of ideal synchronous data pulse streams. Therefore the effects of variables such as time hopping, DS multiplication, timing jitter and pulse attenuation are not covered by these results.

As introduced in Sections 1.4 and 2.4, PSD expressions for non-coded IR-based UWB signals has been previously reported in the literature, [41-50]. PSD expressions for non-coded digital pulse based signals subject to timing jitter are reported in [41] for PAM signals and in [42, 44, 45] for PAM/PPM TH-IR-based UWB signals. All the signal models adopted in these works assumed mutual independence between the PAM modulating stream, the PPM modulating stream and the timing jitter. In [44] the PAM data stream is assumed to be wide sense stationary while the PPM data stream and the jitter are assumed to be second order stationary in the strict sense. The model used in [45] assumes the PAM and PPM data streams to be stationary and uncorrelated, while in [42] the PAM and PPM data streams are assumed to be independent identically distributed (i.i.d.). The model used in [43] is similar to the one used in [44] but focuses on the effects of deterministic TH codes and does not consider timing jitter. In this paper the PAM and PPM data streams are assumed to be i.i.d and mutually independent. The analytical method introduced in [43] was used in [48] to evaluate the PSD of *M*-ary code shift keying (MCSK) IR-based UWB signals. In [47] the spectral analysis of UWB multiple access (MA) schemes is addressed assuming perfect (ideal) random scrambling, no timing jitter, no attenuation and statistically independent data bits. A shot noise approach was used in [46, 49, 50] for the spectral analysis of PAM/PPM IR-based UWB signals with TH/DS, timing jitter and attenuation. These works were based on the assumption of i.i.d. data streams for PPM, i.i.d. and wide sense stationary data streams for PAM, i.i.d. timing jitter and mutual independence of the PAM and PPM data streams. Although the results presented in [46, 49, 50] cover several of the results previously published for spectral analysis of IR-based UWB signals, the case of convolutionally coded/Markov-driven IR-based UWB signals is not addressed by these works. For example the case of correlated PPM as in the case of convolutionally coded/Markov-driven PPM is not covered.

Note that most of the previously mentioned papers assume mutual independence between the PAM and PPM data streams. Thus the results presented in these works are not applicable in a straightforward way to the spectral analysis of IR-based UWB signals where the modulating variables consists of data streams generated by a convolutional encoder. This is due to the correlation introduced by the encoder. For example, if the IR-based UWB system uses BOPPM, [51, 113], in conjunction with a convolutional encoder, then the PAM and PPM data streams will be correlated and a different spectral analysis approach must be used. Therefore it may be postulated that there is a lack of studies addressing the spectral analysis of UWB signals when the data input driving the modulator consists of symbols generated by a convolutional encoder.

Spectral analysis of coded signals has been previously addressed in [52-54] for block codes, in [55, 56] for line codes, in [57, 58] for trellis codes and [59, 60] for convolutional codes. All these works assume ideal synchronous data pulse streams. Therefore these results do not cover the effects of variables such as time hopping, DS multiplication, timing jitter and pulse attenuation. Furthermore the spectral analysis presented in [59, 60] for convolutionally coded signals focuses on

feedforward encoders and PAM/binary phase shift keying (BPSK) modulation and therefore cases such as PPM, BOPPM and PSM are not covered.

This chapter is structured as follows. Section 3.2 describes the system model used throughout this thesis. Of particular importance is the source-encoder Markov model introduced in Section 3.2.3 as it is the analysis starting point. The signal model is introduced in Section 3.3 while the derivations of its average autocorrelation and average power spectrum are addressed in Section 3.4. Section 3.5 presents some illustrative application examples. Finally conclusions are presented in Section 3.6.

3.2 System Model for Convolutionally Coded Impulse Radio Based Ultra Wideband Signals

The block diagram of the generic system model used in this thesis is shown in Figure 3-1. The convolutional encoder is not restricted to be binary to binary. This allows for the inclusion of binary to M-ary encoders. For each input vector (consisting of υ input symbols), $\underline{\mathbf{y}}_l = [y_l^{(0)}, y_l^{(1)}, ..., y_l^{(\upsilon-1)}]$, a convolutional encoder with rate υ/κ produces a encoder output vector (consisting of κ output symbols), $\underline{\mathbf{z}}_l = [z_l^{(0)}, z_l^{(1)}, ..., z_l^{(\kappa-1)}]$, which is then fed to the UWB modulator. The UWB modulator then generates a pulse or series of pulses (depending on the modulation format) which are further subject to an additional time shift specified by a random or pseudo-random (PR) TH code and/or multiplication by a random or PR DS sequence. Note the UWB modulator block is not specifically detailed to allow for the inclusion of different modulation formats.

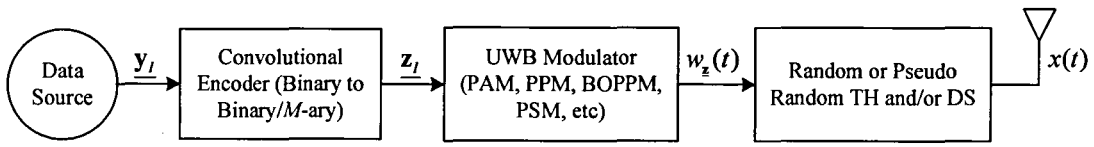


Figure 3-1. Block diagram of a generic convolutionally coded IR-based UWB system.

As this work focuses on the effects that the convolutional encoding operation has on the transmitted signal's PSD, it is assumed throughout this and all other chapters that the antenna does not significantly distort the transmitted and received pulses or that the antenna effects over the pulse shape are perfectly known and have been accounted for in the pulse and overall system design. Note this is an idealisation assumed to enable a fair comparison between different systems and different encoders. For a review of the antenna and channel effects over the pulse shape the reader is referred to [163].

Further specifics on the operation of each block are given in the following sections where a Markov chain model capturing the effects of the source-encoding operation is introduced. This model will then be used in the spectral analysis of the convolutionally coded/Markov-driven IR-based UWB signal.

3.2.1 Data Source Model

It has been customary to assume a source model generating uniform (equally likely) independent identically distributed (i.i.d.) symbols (e.g. $p_0 = 1/2$ and $p_1 = 1/2$ for a memoryless binary source), when analysing the performance of error control coding schemes (such as those involving convolutional coding). Although this model is perfectly suited for bit error rate (BER) performance analysis, it is not the only case of interest when analysing the effects of the coding operation over the signal's PSD. This is particularly true since any unbalance in the data symbol probabilities (e.g. $p_0 \neq 1/2$ and $p_1 = 1 - p_0$) increases the number spectral lines appearing in the PSD of non-coded IR-based UWB signals (see Section 2.4). Therefore a model covering cases of sources generating non-uniform distributed (unbalanced) i.i.d. memoryless data sequences should be considered for a better analysis of the convolutional encoding effects over the IR-based UWB signal's PSD.

A source model covering the unbalanced i.i.d memoryless case as well as information sequences with memory is defined by the first order binary Markov source (BMS) depicted in Figure 3-2, [116-118], where

$$p_{y,ij} = \Pr\{\rho_l = r_j \mid \rho_{l-1} = r_i\} = \Pr\{y_l = j \mid y_{l-1} = i\} \tag{3.1}$$

and

$$p_{y,00} = 1 - p_{y,01}; \quad p_{y,11} = 1 - p_{y,10} \tag{3.2}$$

are the Markov chain (MC) one step transition probabilities, ρ_l is the current state of the Markov chain and y_l is the current symbol produced by the source. Therefore the source model adopted in this chapter consists of the BMS defined by the one step transition probabilities matrix

$$\mathbf{P}_y = \begin{bmatrix} p_{y,00} & p_{y,01} \\ p_{y,10} & p_{y,11} \end{bmatrix} \tag{3.3}$$

and the initial probability mass function $p_{y,i}^{(0)}$.

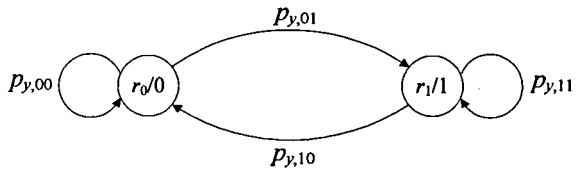


Figure 3-2. First order binary Markov source (BMS) model.

Avoiding trivial degenerate cases, such as $p_{y,ij} = 1$, the BMS model can be classified as an ergodic regular (acyclic) MC, [119-120], (irreducible ergodic in [121-122] notation), and the following properties apply:

$$\pi_{y,j} = \lim_{n\to\infty} p_{y,ij}^{(n)}, \quad \underline{\pi_y} = \underline{\pi_y}\mathbf{P}_y, \quad \lim_{n\to\infty} \mathbf{P}_y^n = \mathbf{P}_y^{\infty} \tag{3.4}$$

where $\pi_{y,j}$ is state r_j steady state probability, $\underline{\pi_y}$ is the vector $[\pi_{y,0}\ \pi_{y,1}]$, $p_{y,ij}^{(n)}$ is the n^{th} step transition probability (that is the probability of being in state r_i and after n transitions reach state r_j),

$\mathbf{P}_y^n$ is the n^{th} step transition probabilities matrix with elements $p_{y,ij}^{(n)}$ and $\mathbf{P}_y^\infty$ is the matrix with identical rows equal to $\underline{\pi}_y$. The corresponding steady state (stationary) probabilities for this Markov chain (MC) can be found to be

$$\underline{\pi}_y = [\pi_{y,0} \;\; \pi_{y,1}] = \left[\frac{p_{y,10}}{p_{y,01}+p_{y,10}} \;\; \frac{p_{y,01}}{p_{y,01}+p_{y,10}}\right] \tag{3.5}$$

where $\pi_{y,0}$ and $\pi_{y,1}$ are the steady state probabilities of generating a "0" and a "1" respectively.

Throughout the rest of the analysis it is assumed that the MC has reached steady state and $p_{y,0}^{(0)} = \pi_{y,0}$ and $p_{y,1}^{(0)} = \pi_{y,1}$. Note the i.i.d. memoryless case is obtained by setting $p_{y,01} = p_{y,11}$ with $p_{y,00} = p_{y,10}$, $\pi_{y,0} = p_{y,10}$ and $\pi_{y,1} = p_{y,01}$.

3.2.2 Convolutional Encoder Model

As explained in Section 2.7 there exist several representations for convolutional encoders such as binary/*M*-ary generator matrices, transfer function matrices, trellis diagrams and state transition diagrams. In fact the spectral analysis of convolutionally coded signals reported in [59] and [60] use the binary and *M*-ary generator (connection) matrices representations of the encoders respectively and focus on feedforward encoders and PAM/PSK modulation. Nevertheless, as it will be shown in subsequent chapters, the feedforward scheme is not the only case of interest for IR-based UWB systems. Therefore the transfer function matrix, **G**(*D*), and state transition diagram representations as defined in Section 2.7 will be adopted in this thesis as they allow for the inclusion of a wider set of encoders such as those with feedback.

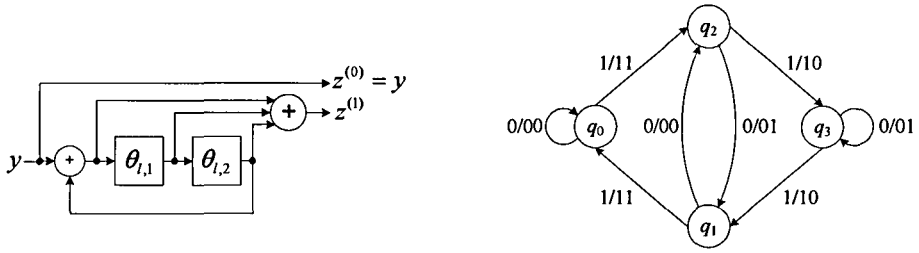
As briefly mentioned in Section 2.7, the convolutional encoder's state transition diagram is in essence finite state sequential machine (FSSM). Formally, an FSSM, **Ω**, is defined by a quintuple $(\mathcal{Q},\mathcal{Y},\mathcal{Z},\phi,\gamma)$ where $\mathcal{Q}=\{q_0,q_1,...,q_{N_q-1}\}$ is a finite set of states (that is the set of all possible memory content combinations of the encoder); $\mathcal{Y}=\{\iota_0,\iota_1,...,\iota_{N_y-1}\}$ is a set of input symbols or input symbol vectors (that is the set of valid symbols/vectors generated by the source); $\mathcal{Z}=\{\zeta_0,\zeta_1,...,\zeta_{N_z-1}\}$ is a set of output symbols or output symbol vectors (that is the set of code symbols/vectors generated by the encoder); $\phi:\mathcal{Y}\times\mathcal{Q}\to\mathcal{Q}$ is the next state function, and $\gamma:\mathcal{Y}\times\mathcal{Q}\to\mathcal{Z}$ is the output function. Note for convolutional encoders the output is usually a vector consisting of all κ binary/*M*-ary code symbols generated by the encoder at each encoder time *l*.

A FSSM is classified as Melay if its output depends on both the current input and the current state, that is, $\theta_{l+1}=\phi(y_l,\theta_l)$ and $z_l=\gamma(y_l,\theta_l)$, where y_l, θ_l and z_l are the current input, the current state and the current output respectively. Besides, a FSSM is classified as Moore if γ is restricted to be a mapping of $\mathcal{Q}$ onto $\mathcal{Z}$, that is, $\gamma:\mathcal{Q}\to\mathcal{Z}$ and $z_l=\gamma(\theta_l)$. Therefore, the encoder's state transition diagram can be classified as a Melay FSSM owing to the fact that the output depends on both the current state and the current input.

As well as state transition diagram representations, there exist other representations for FSSM such as transition matrices which are explained with the following example. Consider the binary to binary systematic recursive convolutional encoder defined by the transfer function matrix (rational generator matrix)

$$\mathbf{G}(D)=\left[1 \quad \frac{1+D+D^2}{1+D^2}\right] \tag{3.6}$$

with state transition diagram shown in Figure 3-3b.



a) Sequential circuit b) State transition diagram (Melay FSSM)

Figure 3-3. Systematic recursive convolutional encoder and its respective state transition diagram.

Remember the encoder state is defined as its memory content. Hence, the states in Figure 3-3b are associated with the encoder's memory content in the following way: $q_0 \leftrightarrow (\theta_{l,1}\theta_{l,2})=(00)$, $q_1 \leftrightarrow (01)$, $q_2 \leftrightarrow (10)$, $q_3 \leftrightarrow (11)$, where by convention the content of the leftmost memory cell, $\theta_{l,1}$,is assumed to be the most significant bit (MSB). Therefore the transition matrix of the encoder Melay machine model is given by

$$\begin{array}{c} \\ \text{Current State} \end{array} \overbrace{\begin{array}{cccc} q_0 & q_1 & q_2 & q_3 \end{array}}^{\text{Next State}} \begin{array}{c} \\ \begin{matrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{matrix} \end{array} \begin{bmatrix} 0/00 & - & 1/11 & - \\ 1/11 & - & 0/00 & - \\ - & 0/01 & - & 1/10 \\ - & 1/10 & - & 0/01 \end{bmatrix} = \boldsymbol{\Omega} \tag{3.7}$$

where the matrix elements are the input causing a transition from state q_i to state q_j and the output produced by such transition, that is $y/z^{(0)}z^{(1)}$.

Two general characteristics of the state transition diagrams of realisable convolutional encoders are of importance for the rest of this chapter. The first is that the encoder's state transition diagram (hence its Melay machine) will have a self loop in the all zeros state, q_0, producing the all-zeros output, $\underline{\mathbf{z}}_l=[0^{(0)},0^{(1)},...,0^{(\kappa-1)}]$, when the all zeros input, $\underline{\mathbf{y}}_l=[0^{(0)},0^{(1)},...,0^{(\upsilon-1)}]$, is applied. In order to prove this consider the basic convolutional encoder structure introduced in Section 2.6 reproduced in Figure 3-4 with some minor modifications. If the encoder is in state zero, then $\theta_{l,k}^{(i)}=u_{l-k}^{(i)}=0$, for $k=1,2,...,v_i$ and $i=0,1,...,\upsilon-1$. Next note that

$$u_l^{(i)}=y_l^{(i)}\oplus\sum_{k=1}^{v_i}\theta_{l,k}^{(i)}b_{i,k}=y_l^{(i)}\oplus\sum_{k=1}^{v_i}u_{l-k}^{(i)}b_{i,k} \tag{3.8}$$

where $\oplus$ stands for modulo 2 addition (that is over the binary Galois field – GF(2)). Thus, if $\underline{y}_l=[0^{(0)},0^{(1)},...,0^{(\upsilon-1)}]$ is the current input then $u_l^{(i)}=0$, and the next state will be q_0 again as

$\theta^{(i)}_{(l+1),1} = u^{(i)}_l = 0$. Similarly, if the encoder is in the all zeros state and the input is $\underline{y_l} = [0^{(0)}, 0^{(1)}, ..., 0^{(\upsilon-1)}]$, then

$$z^{(j)}_l = \sum_{i=0}^{\upsilon-1} \left(u^{(i)}_l a^{(j)}_{i,0} \oplus \sum_{k=1}^{v_i} \theta^{(i)}_{l,k} a^{(j)}_{i,k} \right) = \sum_{i=0}^{\upsilon-1} \sum_{k=0}^{v_i} u^{(i)}_{l-k} a^{(j)}_{i,k} = 0 \tag{3.9}$$

that is $\underline{\mathbf{z}_l} = [0^{(0)}, 0^{(1)}, ..., 0^{(\kappa-1)}]$, where the sums $\oplus$ in Equation (3.9) are performed modulo 2 for binary convolutional encoders and modulo M for binary to M-ary convolutional encoders.

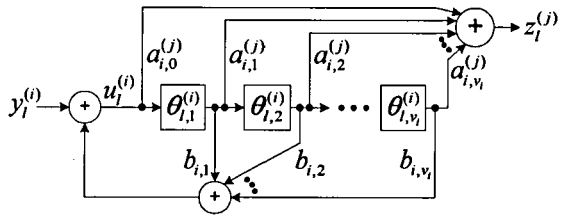


Figure 3-4. Basic structure of a convolutional encoder showing the connections from shift register i to output $z^{(j)}$.

In order to further clarify this property consider the rate 1/3, v =3, binary convolutional encoder shown in Figure 3-5. It can be seen in this figure that if the encoder is in the all zeros state and the current input is zero, then the encoder generates the all zeros output and the next state will be the all zeros state again as $u_l = 0$.

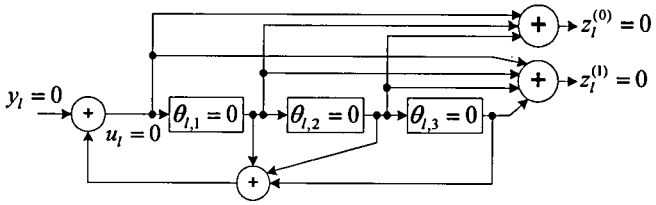


Figure 3-5. Rate 1/2, $v = 3$, binary convolutional encoder producing the all zeros output.

The second characteristic is that the convolutional encoder is a strongly connected machine. A FSSM, Ω, is defined to be strongly connected if for every pair of states q_n, $q_{n'}$ belonging to $\mathcal{Q}$, there exist and input sequence driving Ω from q_n to $q_{n'}$, [123]. For convenience assume that all the encoder's shift registers have the same number of memory cells, $v_i = m$. In order to prove that the encoder's state transition diagram is strongly connected assume that $[\theta^{(i)}_{l,1}, \theta^{(i)}_{l,2}, ..., \theta^{(i)}_{l,v_i}] = [\mu^{(i)}_1, \mu^{(i)}_2, ..., \mu^{(i)}_{v_i}]$, $i = 0,1,...,\upsilon-1$, is the encoder's memory content at time l and that it defines state q_n (n is the decimal representation of the memory content). Similarly, assume $[\theta^{(i)}_{l+v_i,1}, \theta^{(i)}_{l+v_i,2}, ..., \theta^{(i)}_{l+v_i,v_i}] = [\mu'^{(i)}_1, \mu'^{(i)}_2, ..., \mu'^{(i)}_{v_i}]$ defines state $q_{n'}$. By setting $u^{(i)}_l = \mu'^{(i)}_{v_i}$ in Equation (3.8) we can find

$$y'^{(i)}_0 = \mu'^{(i)}_{v_i} \oplus \sum_{k=1}^{v_i} \theta^{(i)}_{l,k} b_{i,k} = \mu'^{(i)}_{v_i} \oplus \sum_{k=1}^{v_i} \mu^{(i)}_k b_{i,k} \tag{3.10}$$

By assuming that the previous value is used as an input in Equation (3.8), that is, $y^{(i)}_l = y'^{(i)}_0$, and

setting, $u_{l+1}^{(i)} = \mu'^{(i)}_{v_i-1}$, we can recursively obtain

$$y'^{(i)}_n = \mu'^{(i)}_{v_i-n} \oplus \sum_{k=1}^{v_i} \theta^{(i)}_{l+n,k} b_{i,k} = \mu'^{(i)}_{v_i-n} \oplus \sum_{k=1}^{n} \mu'^{(i)}_{v_i-n+k}\, b_{i,k} \oplus \sum_{k=n+1}^{v_i} \mu^{(i)}_{k-n} b_{i,k} \tag{3.11}$$

by setting $y^{(i)}_{l+n} = y'^{(i)}_n$ for $n = 1,2,...,v_i - 1$. Note at time $l+(v_i-1)$: $y^{(i)}_{l+(v_i-1)} = y'^{(i)}_{v_i-1}$, $u^{(i)}_{l+(v_i-1)} = \mu'^{(i)}_1$, $[\theta^{(i)}_{l+(v_i-1),1}, \theta^{(i)}_{l+(v_i-1),2}, ..., \theta^{(i)}_{l+(v_i-1),v_i-1}, \theta^{(i)}_{l+(v_i-1),v_i}] = [\mu'^{(i)}_2, \mu'^{(i)}_3, ..., \mu'^{(i)}_{v_i}, \mu^{(i)}_1]$, and the next state will be $q_{n'}$. Therefore by using the sequence, $\{ y'^{(i)}_0\, y'^{(i)}_1 ... y'^{(i)}_{v_i-1} \}$, as defined above, the encoder's FSSM, Ω, can be driven from state q_n to state $q_{n'}$ and the machine is strongly connected.

The convolutional encoder representation used throughout this chapter has now been defined. The source-encoder Markov model will be introduced in the next section.

3.2.3 Source-Encoder Markov Model

The next step in the analysis requires finding a joint source-encoder (SE) stochastic model when the input to the convolutional encoder is given by the BMS introduced in Section 3.2.1. In order to find such model, the methodology introduced in [123] pp. 514 - 518 for the analysis of strongly connected deterministic sequential machines with linearly dependent stochastic inputs will be used.

Assume the Melay FSSM representation of the convolutional encoder, $\Omega = (\mathcal{Q}, \mathcal{Y}, \mathcal{Z}, \phi, \gamma)$, as defined in Section 3.2.2 has been obtained. Define the input to the encoder to be a sequence of binary random variables, $\{y_l\}$, generated by the BMS introduced in Section 3.2.1 with state set $\mathcal{R} = \{r_0, r_1\}$ and corresponding outputs $\{\iota_0, \iota_1\} = \{0,1\}$. Then the SE pair consists of a strongly connected Melay FSSM, Ω, with a linearly dependent input process, y_l, which is Markov itself.

Let the number of encoder states be N_q, that is, $\mathcal{Q} = \{q_0, q_1, ..., q_{N_q-1}\}$. Now define a set of $N_q \times N_q$ "next state" matrices, $\mathbf{B}_n = [B_{n,ij}]$, for each element in $\mathcal{Y} = \{\iota_0, \iota_1\} = \{0,1\}$, $n = 0,1$. Then, $B_{n,ij}$, is defined to be equal to unity if $q_j = \phi(\iota_n, q_i)$, and zero otherwise. For example, the next state matrices of the encoder defined in Equation (3.6) are given by

$$\mathbf{B}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } \mathbf{B}_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{3.12}$$

By using the BMS's transition probabilities matrix, $\mathbf{P}_y$, defined in Equation (3.3), and the next state matrices, a Markov model for the SE pair can be obtained by defining a new Markov chain state process, σ_l, with state set defined as

$$\mathcal{S} = \{\mathcal{R} \times \mathcal{Q}\} = \{(r_0,q_0),(r_0,q_1),...,(r_0,q_{N_q-1}),(r_1,q_0),(r_1,q_1),...,(r_1,q_{N_q-1})\} = \{s_0, s_1, ..., s_{2N_q-1}\} \tag{3.13}$$

and transition probabilities matrix given by

$$\mathbf{P}_\sigma = \begin{bmatrix} p_{y,00}\mathbf{B}_0 & p_{y,01}\mathbf{B}_0 \\ p_{y,10}\mathbf{B}_1 & p_{y,11}\mathbf{B}_1 \end{bmatrix} \tag{3.14}$$

That Equation (3.14) defines indeed a transition matrix for the new state process can be seen from

the fact that there will be a transition from state (r_n, q_i) to state $(r_{n'}, q_{i'})$ only if $q_{i'} = \phi(\iota_n, q_i)$ and $p_{y,nn'} = \Pr\{\rho_l = r_{n'} \mid \rho_{l-1} = r_n\} \neq 0$.

Note from Figure 3-2 and Equation (3.1) that there is a one to one correspondence between the BMS's state, r_n, and the BMS's output symbol, ι_n. Therefore the state (r_n, q_i) can be relabelled as (ι_n, q_i) and the output mapping of the SE pair Markov model can be defined as

$$\underline{\zeta_{s_{nN_q+i}}} = [\zeta^{(0)}_{s_{nN_q+i}}, \zeta^{(1)}_{s_{nN_q+i}}, \ldots, \zeta^{(\kappa-1)}_{s_{nN_q+i}}] = \gamma(r_n, q_i) = \gamma(\iota_n, q_i) = \gamma(s_{nN_q+i}) \tag{3.15}$$

where $n = 0,1$ and $i = 0,1,\ldots,N_q - 1$. Equation (3.15) implies that the system output, $\underline{\zeta}$, depends exclusively on the SE pair MC state, $s_{nN_q+i} = (r_n, q_i) \in \mathcal{S}$, and the SE pair Markov model can be classified as Moore type. For this reason the SE pair Markov model's output process, z_l, is said to be a projection of the MC state process σ_l. Note z_l is not necessarily Markov as one specific output symbol/vector $\underline{\zeta_j}$ can be assigned to more than one state in $\mathcal{S}$. As an example the SE pair Markov model for the convolutional encoder defined by Equation (3.6) has state transition matrix and output mapping

$$\mathbf{P}_\sigma = \begin{array}{c} \\ s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{array} \begin{array}{c} \begin{array}{cccccccc} s_0 & s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 \end{array} \\ \left[\begin{array}{cccccccc} p_{y,00} & 0 & 0 & 0 & p_{y,01} & 0 & 0 & 0 \\ 0 & 0 & p_{y,00} & 0 & 0 & 0 & p_{y,01} & 0 \\ 0 & p_{y,00} & 0 & 0 & 0 & p_{y,01} & 0 & 0 \\ 0 & 0 & 0 & p_{y,00} & 0 & 0 & 0 & p_{y,01} \\ 0 & 0 & p_{y,10} & 0 & 0 & 0 & p_{y,11} & 0 \\ p_{y,10} & 0 & 0 & 0 & p_{y,11} & 0 & 0 & 0 \\ 0 & 0 & 0 & p_{y,10} & 0 & 0 & 0 & p_{y,11} \\ 0 & p_{y,10} & 0 & 0 & 0 & p_{y,11} & 0 & 0 \end{array}\right] \end{array} \quad ; \quad \begin{array}{l} s_0 = (r_0, q_0) = (0,00);\ \gamma(s_0) = [0,0] \\ s_1 = (r_0, q_1) = (0,01);\ \gamma(s_1) = [0,0] \\ s_2 = (r_0, q_2) = (0,10);\ \gamma(s_2) = [0\ 1] \\ s_3 = (r_0, q_3) = (0,11);\ \gamma(s_3) = [0, 1] \\ s_4 = (r_1, q_0) = (1,00);\ \gamma(s_4) = [1, 1] \\ s_5 = (r_1, q_1) = (1,01);\ \gamma(s_5) = [1, 1] \\ s_6 = (r_1, q_2) = (1,10);\ \gamma(s_6) = [1, 0] \\ s_7 = (r_1, q_3) = (1,11);\ \gamma(s_7) = [1, 0] \end{array} \tag{3.16}$$

Avoiding trivial degenerate cases such as $p_{y,00} = 0$ and similar, an important characteristic of the transition probabilities matrix, $\mathbf{P}_\sigma$, is that it does not have transient states since the encoder's Melay FSSM model, Ω, is strongly connected. Furthermore, $\mathbf{P}_\sigma$ has a transition from state $s_0 = (r_0, q_0)$ to itself due to $b_{0,00} = 1$ (because of the self loop in the all zeros state, q_0, on the encoder's Melay FSSM model, Ω) and $0 < p_{y,00} < 1$. The self loop presence means state s_0 has period one. Hence, the MC state process, σ_l, with transition probabilities matrix, $\mathbf{P}_\sigma$, can be classified as an ergodic regular (acyclic) MC, [119, 120], (irreducible ergodic in [121, 122] notation). Therefore the properties described in Equation (3.4) for the BMS apply for σ_l and $\mathbf{P}_\sigma$ as well, that is

$$\pi_{\sigma,j} = \lim_{n\to\infty} p^{(n)}_{\sigma,jj}, \quad \underline{\pi_\sigma} = \underline{\pi_\sigma}\mathbf{P}_\sigma, \quad \lim_{n\to\infty} \mathbf{P}^n_\sigma = \mathbf{P}^\infty_\sigma \tag{3.17}$$

where $\mathbf{P}^n_\sigma = [p^{(n)}_{\sigma,jj}]$ is the n^{th} step transition probabilities matrix, $\underline{\pi_\sigma} = [\pi_{\sigma,0}, \pi_{\sigma,1}, \ldots, \pi_{\sigma,N_\sigma-1}]$ is the steady state probabilities vector, $N_\sigma = 2N_q$ is the number of elements in $\mathcal{S} = \{s_0, s_1, \ldots, s_{2N_q-1}\}$, and

$\mathbf{P}_{\sigma}^{\infty}$ is the matrix with identical rows equal to $\underline{\pi_{\sigma}}$. As an example the steady state probabilities for the MC in Equation (3.16) are

$$\underline{\pi_{\sigma}} = \left[\frac{\pi_{y,0}}{4}, \frac{\pi_{y,0}}{4}, \frac{\pi_{y,0}}{4}, \frac{\pi_{y,0}}{4}, \frac{\pi_{y,1}}{4}, \frac{\pi_{y,1}}{4}, \frac{\pi_{y,1}}{4}, \frac{\pi_{y,1}}{4}\right] \tag{3.18}$$

where $\pi_{y,0}$ and $\pi_{y,1}$ are defined in Equation (3.5).

Another property of regular ergodic MCs related to n^{th} step transition probabilities (useful in further calculations) is defined by ([119, 120])

$$p_{ij}^{(n)} = \pi_j + \vartheta_{ij}^{(n)} \tag{3.19}$$

where $\left|\vartheta_{ij}^{(n)}\right| \leq br^n$ with b constant and $0 < r < 1$. Note Equation (3.19) implies convergence to the steady state probabilities as in Equation (3.17).

Once the SE pair has been defined next sections introduce the signal model and spectral analysis for the convolutionally/Markov-driven IR-based UWB signals.

3.3 Signal Model for Convolutionally Coded/Markov-Driven IR-Based UWB Signals

As explained in Section 2.3, IR-based UWB systems convey the information by transmitting low power ultra short pulses where more than one pulse is usually transmitted per information symbol. Hence an encoder output symbol, z_l, which consists of one or more binary/M-ary code symbols, can be transmitted using one or several consecutive pulses depending on the modulation scheme used. For example, the output of the encoder shown in Figure 3-3, consisting of the two binary symbols $z^{(1)}z^{(2)}$, can be coupled to binary PAM or PPM (two pulses or signals per output symbol), and quaternary PAM, PPM, BOPPM or PSM (one pulse or signal per output symbol). Following the general system model introduced in Section 3.2, once modulation has been performed each pulse can be subject to an additional time shift ruled by a PR-TH sequence and/or multiplication by a PR-DS. The PR-TH and PR-DS sequences can be used for multiple access or spectral shaping purposes. Finally, although ideally each pulse is transmitted with fixed amplitude at regular time intervals of length T_r, imperfections such as jitter and/or attenuation should be considered in the model. A signal model including all these variables can be defined as

$$x(t) = \sum_{l=-\infty}^{\infty} \sum_{k=0}^{N_w-1} \xi_{lN_w+k} a_{lN_w+k} w_{\sigma_l,k}(t - lT_s - kT_r - c_{lN_w+k}T_c - \Delta_{lN_w+k}) \tag{3.20}$$

where σ_l is the SE pair Markov model state process with N_σ states; $w_{\sigma_l,k}(t - lT_s - kT_r)$ is the k^{th} pulse used for the transmission of the encoder output vector $z_l = \gamma(\sigma_l)$ (consisting of κ output symbols) at symbol time l; N_w is the number of pulses used per encoder output vector; T_r is the mean repetition time between pulses; $T_s = N_w T_r$ is the encoder output vector time; $\{a_{lN_w+k}\}$ is a PR-DS sequence usually (but not restricted to) taking values on the sets {-1, 1} or {0, 1} with period χ_a;

$\{c_{lN_w+k}\}$ is a PR-TH sequence taking values on the set {0, 1, 2 ,..., N_c-1} with period χ_c; T_c is the nominal shift caused by the TH sequence; Δ_{lN_w+k} is random jitter (assumed to be second order stationary at least) and ξ_{lN_w+k} is random attenuation (assumed to be wide sense stationary).

Note $w_{\sigma_l,k}(t-lT_s-kT_r)$ depends on the current state of the MC, σ_l, at time l and the kind of modulation used. As the purpose of this chapter is to evaluate the PSD's average statistics, it is assumed the MC has already reached steady state, which is reflected in the fact that the signal model assumes -∞ as starting time. It is further assumed that, σ_l, ξ_{lN_w+k}, Δ_{lN_w+k}, a_{lN_w+k} and c_{lN_w+k} are mutually independent which is the most common case. It will prove useful to further define

$$\chi_{acw} = \text{lcm}(\chi_a, \chi_c, N_w) \quad \text{and} \quad \Lambda = \chi_{acw}/N_w \tag{3.21}$$

where lcm stands for least common multiple.

For example, consider binary PPM is used with the encoder defined by Equation (3.6) with SE pair Markov model defined by Equation (3.16). If the encoder is in state $\sigma_l = s_i$ with output vector $z_l = \underline{\mathbf{z}_l} = [z_l^{(0)}, z_l^{(1)}] = \gamma(\sigma_l = s_i) = \underline{\zeta_{s_i}} = [\zeta_{s_i}^{(0)}, \zeta_{s_i}^{(1)}]$, then the system output at time l (without considering attenuation, jitter, TH or DS) would be given by

$$w_{s_i}(t) = w(t-\zeta_{s_i}^{(0)}T_\beta - lT_s) + w(t-\zeta_{s_i}^{(1)}T_\beta - lT_s - T_r) \tag{3.22}$$

where T_β is the PPM modulation index. Besides, if quaternary BOPPM (Q-BOPPM) is used instead of binary PPM, then the output signal at time l would be given by

$$w_{s_i}(t) = \alpha_i w(t-\beta_i T_\beta) \tag{3.23}$$

where $\alpha_i = (2\zeta_{s_i}^{(0)} - 1)$ and $\beta_i = \zeta_{s_i}^{(1)}$. It is important to note that α_i and β_i *are not mutually independent* as usually assumed but correlated.

Now the signal model has been defined the spectral analysis of the signal model is introduced in the next section.

3.4 Spectral Analysis of the Signal Model for Convolutionally Coded/Markov-Driven IR-Based UWB Signals

Due to the introduction of variables such as attenuation and jitter and to the fact that χ_a, χ_c are usually more than N_w, PSD expressions for MC-driven signals such as those reported in [52, 83, 87 99, 114, 115], can not be straightforwardly used for the spectral analysis of the signal model introduced in the previous section. The analysis presented in this correspondence is based on the derivation of the average autocorrelation and average power spectrum (that is, the PSD) of the non-stationary signal $x(t)$. This is addressed using an approach similar to that introduced in [124] for the analysis of non-stationary processes and therefore the cyclostationary hypothesis is not addressed.

3.4.1 Average Autocorrelation of the Signal Model for Convolutionally Coded/Markov-Driven IR-Based UWB Signals

The first step of the PSD derivation of the signal model is to calculate the signal's average autocorrelation, $\overline{R_x}(\tau)$, which is defined as

$$\overline{R_x}(\tau) = \lim_{T\to\infty} \frac{1}{2T} \mathrm{E}\{ \int_{-T}^{T} x(t)x^*(t+\tau)dt \} = \lim_{T\to\infty} \frac{1}{2T} \mathrm{E}\{R_{xT}(\tau)\} \tag{3.24}$$

where $\mathrm{E}\{\cdot\}$ denotes the expected value, $x^*(t)$ stands for the complex conjugate of $x(t)$ and

$$\mathrm{E}\{R_{xT}(\tau)\} = \int_{-T}^{T} \mathrm{E}\{x(t)x^*(t+\tau)\}dt \tag{3.25}$$

substituting Equation (3.20) in Equation (3.24)

$$\begin{aligned} \mathrm{E}\{R_{xT}(\tau)\} = \int_{-T}^{T} \mathrm{E}\Bigg\{ & \sum_{k=0}^{N_w-1} \sum_{l=-\infty}^{\infty} \xi_{lN_w+k} a_{lN_w+k} w_{\sigma_l,k}(t - lT_s - kT_r - c_{lN_w+k}T_c - \Delta_{lN_w+k}) \\ & \sum_{k'=0}^{N_w-1} \sum_{l'=-\infty}^{\infty} \xi^*_{l'N_w+k'} a^*_{l'N_w+k'} w^*_{\sigma_{l'},k'}(t+\tau - l'T_s - k'T_r - c_{l'N_w+k'}T_c - \Delta_{l'N_w+k'}) \Bigg\} dt \end{aligned} \tag{3.26}$$

By introducing

$$w_{\sigma_l,k}(t) = \int_{-\infty}^{\infty} W_{\sigma_l,k}(\nu) e^{+j2\pi\nu t} d\nu \tag{3.27}$$

Equation (3.26) can be written as

$$\begin{aligned} \mathrm{E}\{R_{xT}(\tau)\} = \int_{-T}^{T} \sum_{k,k'=0}^{N_w-1} \sum_{l,l'=-\infty}^{\infty} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \Big\{ & e^{j2\pi(\nu+\nu')t} e^{j2\pi\nu'\tau} e^{-j2\pi T_s(\nu l+\nu' l')} e^{-j2\pi T_r(\nu k+\nu' k')} \\ & \mathrm{E}\{\xi_{lN_w+k}\xi^*_{l'N_w+k'}\} a_{lN_w+k} a^*_{l'N_w+k'} e^{-j2\pi T_c(\nu c_{lN_w+k}+\nu' c_{l'N_w+k'})} \\ & \mathrm{E}\{e^{-j2\pi(\nu\Delta_{lN_w+k}+\nu'\Delta_{l'N_w+k'})}\} \mathrm{E}\{W_{\sigma_l,k}(\nu) W^*_{\sigma_{l'},k'}(-\nu')\} \Big\} d\nu d\nu' dt \end{aligned} \tag{3.28}$$

where the mutual independence of σ_l, ξ_{lN_w+k}, Δ_{lN_w+k}, a_{lN_w+k} and c_{lN_w+k} has been used to separate the expectations.

Let us now define

$$G_1(l,l',k,k',\nu,\nu') = \mathrm{E}\{W_{\sigma_l,k}(\nu) W^*_{\sigma_{l'},k'}(-\nu')\} \tag{3.29}$$

Under the stationary steady state assumption the following stands for the signal selection process

$$\Pr\{\sigma_l = s_i\} = \Pr\{w_{\sigma_l,k}(t - kT_r) = w_{s_i,k}(t - kT_r)\} = \Pr\{W_{\sigma_l,k}(\nu) = W_{s_i,k}(\nu)\} = \pi_i \tag{3.30}$$

in addition

$$\begin{aligned} \Pr\{\sigma_l = s_i, \sigma_{l'} = s_{i'}\} &= \Pr\{\sigma_l = s_i\} \Pr\{\sigma_{l'} = s_{i'} \mid \sigma_l = s_i\} = \pi_i p_{ii'}^{(l'-l)} \qquad l' > l \\ \Pr\{\sigma_l = s_i, \sigma_{l'} = s_{i'}\} &= \Pr\{\sigma_l = s_i\} \Pr\{\sigma_l = s_i \mid \sigma_{l'} = s_{i'}\} = \pi_{i'} p_{i'i}^{(l-l')} \qquad l > l' \end{aligned} \tag{3.31}$$

and thus

$$G_1(l,l',k,k',\nu,\nu') = \begin{cases} \sum_{i=0}^{N_\sigma-1} W_{i,k}(\nu) W^*_{i,k'}(-\nu')\pi_i & l' = l \\ \sum_{i,i'=0}^{N_\sigma-1} W_{i,k}(\nu) W^*_{i',k'}(-\nu')\pi_i p_{ii'}^{(|l'-l|)} & l' \neq l \end{cases} \tag{3.32}$$

where for clarity the subscript "σ" has been dropped from $\pi_{\sigma,i}$ and $p^{(n)}_{\sigma,ii'}$. Note that G_1 is a function of the difference between l' and l, that is

$$G_1(l,l',k,k',\nu,\nu') = G_1(l'-l,k,k',\nu,\nu') = G_1(m,k,k',\nu,\nu') \tag{3.33}$$

where $m = l'-l$. Next we define

$$G_2(l,l',k,k',\nu,\nu') = \mathrm{E}\{\xi_{lN_w+k}\xi^*_{l'N_w+k'}\}\mathrm{E}\{e^{-j2\pi(\nu\Delta_{lN_w+k}+\nu'\Delta_{l'N_w+k'})}\} \tag{3.34}$$

Note G_2 is also a function of the difference between l' and l since ξ_{lN_w+k} is wide sense stationary and Δ_{lN_w+k} is second order stationary. To illustrate this point we see that

$$\mathrm{E}\{\xi_{lN_w+k}\xi^*_{l'N_w+k'}\} = R_\xi((l'-l)N_w + k'-k) \tag{3.35}$$

and

$$\mathrm{E}\{e^{-j2\pi(\nu\Delta_{lN_w+k}+\nu'\Delta_{l'N_w+k'})}\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-j2\pi(\nu\Delta+\nu'\Delta')} g_{\Delta,\Delta'}(\Delta,\Delta';|(l'-l)N_w + k'-k)|)d\Delta d\Delta' \tag{3.36}$$

for $lN_w + k \neq l'N_w + k'$', with

$$\mathrm{E}\{\xi_{lN_w+k}\xi^*_{lN_w+k}\} = R_\xi(0) \quad \text{and} \quad \mathrm{E}\{e^{-j2\pi(\nu\Delta_{lN_w+k}+\nu'\Delta_{l'N_w+k'})}\} = \int_{-\infty}^{\infty} e^{-j2\pi(\nu\Delta+\nu'\Delta)} g_\Delta(\Delta)d\Delta \tag{3.37}$$

for $lN_w + k = l'N_w + k'$, where g_Δ and $g_{\Delta\Delta'}$ are the first and second order probability density functions (p.d.f.) of the jitter process. Therefore G_2 becomes

$$G_2(l,l',k,k',\nu,\nu') = G_2(l'-l,k,k',\nu,\nu') = G_2(m,k,k',\nu,\nu') \tag{3.38}$$

where again $m = l'-l$.

To continue we can define

$$\begin{aligned} G_3(l,l',k,k',\nu,\nu') &= a_{lN_w+k}a^*_{l'N_w+k'}e^{-j2\pi T_c(\nu c_{lN_w+k}+\nu' c_{l'N_w+k'})} \\ &= a_{lN_w+k}a^*_{(l+m)N_w+k'}e^{-j2\pi T_c(\nu c_{lN_w+k}+\nu' c_{(l+m)N_w+k'})} = G_3(l,m,k,k',\nu,\nu') \end{aligned} \tag{3.39}$$

In Equation (3.39) the following properties apply due to the periodicity of the PR-TH and PR-DS codes

$$\begin{aligned} G_3(l+\Lambda,m,k,k',\nu,\nu') &= G_3(l,m,k,k',\nu,\nu') \\ G_3(l,m+\Lambda,k,k',\nu,\nu') &= G_3(l,m,k,k',\nu,\nu') \\ G_3(l+\Lambda,m+\Lambda,k,k',\nu,\nu') &= G_3(l,m,k,k',\nu,\nu') \end{aligned} \tag{3.40}$$

where Λ was defined in Equation (3.21).

Using G_1, G_2, G_3 and $l' = l + m$ Equation (3.28) can be rewritten as

$$\begin{aligned} \mathrm{E}\{R_T(\tau)\} = \int_{-T}^{T}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \Big\{ & e^{j2\pi(\nu+\nu')t}e^{j2\pi\nu'\tau}\sum_{k,k'=0}^{N_w-1}\sum_{l=-\infty}^{\infty} e^{-j2\pi T_s l(\nu+\nu')}\sum_{m=-\infty}^{\infty} e^{-j2\pi T_s m\nu'}e^{-j2\pi T_r(\nu k+\nu' k')} \\ & G_1(m,k,k',\nu,\nu')\, G_2(m,k,k',\nu,\nu')G_3(l,m,k,k',\nu,\nu')\Big\}d\nu d\nu' dt \end{aligned} \tag{3.41}$$

For notational convenience we can write

$$G_A(m,k,k',\nu,\nu') = e^{-j2\pi T_s m\nu'}e^{-j2\pi T_r(\nu k+\nu' k')}G_1(m,k,k',\nu,\nu')\, G_2(m,k,k',\nu,\nu') \tag{3.42}$$

and by setting $l = u'\Lambda + n$ and using Equations (3.40) and (3.42), the sums in Equation (3.41) become

$$\sum_{k,k'=0}^{N_w-1}\sum_{n=0}^{\Lambda-1} e^{-j2\pi T_s n(\nu+\nu')}\sum_{m=-\infty}^{\infty} G_A(m,k,k',\nu,\nu')\,G_3(n,m,k,k',\nu,\nu')\sum_{u'=-\infty}^{\infty} e^{-j2\pi T_s u'\Lambda(\nu+\nu')} \tag{3.43}$$

By using the Poisson sum formula ([161])

$$\sum_{u'=-\infty}^{\infty} e^{-j2\pi x u' T} = \frac{1}{T}\sum_{u=-\infty}^{\infty}\delta(x+\tfrac{u}{T}) \tag{3.44}$$

in Equation (3.43) and replacing the result in Equation (3.41) we get

$$\begin{aligned}\mathrm{E}\{R_T(\tau)\} = \tfrac{1}{T_s\Lambda}\int_{-T}^{T}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\Big\{e^{j2\pi(\nu+\nu')t}e^{j2\pi\nu'\tau}\sum_{k,k'=0}^{N_w-1}\sum_{n=0}^{\Lambda-1}\sum_{m=-\infty}^{\infty}\sum_{u=-\infty}^{\infty} e^{-j2\pi T_s n(\nu+\nu')}e^{-j2\pi T_s m\nu'}e^{-j2\pi T_r(\nu k+\nu' k')}\\ G_1(m,k,k',\nu,\nu')\,G_2(m,k,k',\nu,\nu')G_3(n,m,k,k',\nu,\nu')\delta(\nu+\nu'-\tfrac{u}{T_s\Lambda})\Big\}d\nu d\nu' dt\end{aligned} \tag{3.45}$$

where $\delta(\cdot)$ is the Dirac delta function.

Evaluating the integrals in Equation (3.45) first in ν and second in t and substituting the result in Equation (3.24) yields

$$\begin{aligned}\bar{R}(\tau) = \lim_{T\to\infty}\tfrac{1}{2T}\tfrac{1}{T_s\Lambda}\sum_{k,k'=0}^{N_w-1}\sum_{n=0}^{\Lambda-1}\sum_{m=-\infty}^{\infty}\sum_{u=-\infty}^{\infty}\frac{\sin(2\pi\frac{u}{T_s\Lambda}T)}{\pi\frac{u}{T_s\Lambda}}\int_{-\infty}^{\infty}\Big\{e^{j2\pi\nu'\tau}\,e^{-j2\pi T_s n\frac{u}{T_s\Lambda}}e^{-j2\pi T_s m\nu'}e^{-j2\pi T_r\frac{u}{T_s\Lambda}k}e^{-j2\pi T_r\nu'(k'-k)}\\ G_1(m,k,k',\tfrac{u}{T_s\Lambda}-\nu',\nu')G_2(m,k,k',\tfrac{u}{T_s\Lambda}-\nu',\nu')G_3(n,m,k,k',\tfrac{u}{T_s\Lambda}-\nu',\nu')\Big\}d\nu'\end{aligned} \tag{3.46}$$

However

$$\lim_{T\to\infty}\frac{\sin(2\pi\frac{u}{T_s\Lambda}T)}{2\pi\frac{u}{T_s\Lambda}T} = \begin{cases}1, & u=0\\ 0, & u\neq 0\end{cases}$$

and therefore Equation (3.46) becomes

$$\begin{aligned}\bar{R}(\tau) = \tfrac{1}{T_s\Lambda}\sum_{k,k'=0}^{N_w-1}\sum_{n=0}^{\Lambda-1}\sum_{m=-\infty}^{\infty}\int_{-\infty}^{\infty}\Big\{e^{j2\pi\nu'\tau}e^{-j2\pi T_s m\nu'}e^{-j2\pi T_r\nu'(k'-k)}G_1(m,k,k',-\nu',\nu')\\ G_2(m,k,k',-\nu',\nu')G_3(n,m,k,k',-\nu',\nu')\Big\}d\nu'\end{aligned} \tag{3.47}$$

which is the desired result. Equation (3.47) provides the average autocorrelation function which next will be used to obtain the signal's PSD.

3.4.2 Power Spectral Density of Convolutionally Coded/Markov-Driven IR-Based UWB Signals

The power spectral density, $\bar{S}(f)$, can now be obtained from the Fourier transform (FT) of Equation (3.47) on τ, that is

$$\begin{aligned}\bar{S}(f) = \Im_\tau\{\bar{R}(\tau)\} = \tfrac{1}{T_s\Lambda}\sum_{k,k'=0}^{N_w-1}\sum_{n=0}^{\Lambda-1}\sum_{m=-\infty}^{\infty}\int_{-\infty}^{\infty}\Big\{\Im_\tau\{e^{j2\pi\nu'\tau}\}e^{-j2\pi T_s m\nu'}e^{-j2\pi T_r\nu'(k'-k)}G_1(m,k,k',-\nu',\nu')\\ G_2(m,k,k',-\nu',\nu')G_3(n,m,k,k',-\nu',\nu')\Big\}d\nu'\end{aligned} \tag{3.48}$$

where $\Im\{\cdot\}$ is the Fourier transform operator. Evaluating Equation (3.48) yields

$$\begin{aligned}\bar{S}(f) = \tfrac{1}{T_s\Lambda}\sum_{k,k'=0}^{N_w-1}\sum_{n=0}^{\Lambda-1}\sum_{m=-\infty}^{\infty}\int_{-\infty}^{\infty}\Big\{\delta(f-\nu')e^{-j2\pi T_s m\nu'}e^{-j2\pi T_r\nu'(k'-k)}G_1(m,k,k',-\nu',\nu')\\ G_2(m,k,k',-\nu',\nu')G_3(n,m,k,k',-\nu',\nu')\Big\}d\nu'\end{aligned} \tag{3.49}$$

and performing the integration on ν'

$$\bar{S}(f)=\frac{1}{T_s\Lambda}\sum_{k,k'=0}^{N_w-1}\sum_{n=0}^{\Lambda-1}\sum_{m=-\infty}^{\infty}e^{-j2\pi fT_s m}e^{-j2\pi f(k'-k)T_r}G_1(m,k,k',-f,f)G_2(m,k,k',-f,f)G_3(n,m,k,k',-f,f) \quad (3.50)$$

where from Equation (3.32)

$$G_1(m,k,k',-f,f)=\begin{cases}\sum_{i=0}^{N_\sigma-1}W_{i,k}(-f)W_{i,k'}^*(-f)\pi_i & m=0\\ \sum_{i,i'=0}^{N_\sigma-1}W_{i,k}(-f)W_{i',k'}^*(-f)\pi_i p_{ii'}^{(|m|)} & m\neq 0\end{cases} \quad (3.51)$$

or using Equation (3.19) in Equation (3.51)

$$G_1(m,k,k',-f,f)=\sum_{i,i'=0}^{N_\sigma-1}W_{i,k}(-f)W_{i',k'}^*(-f)\pi_i(\pi_{i'}+\vartheta_{i,i'}^{(|m|)}) \quad (3.52)$$

for $m\neq 0$. Similarly from Equations (3.34) and (3.37)

$$G_2(0,k,k,-f,f)=\mathrm{E}\{\xi_{lN_w+k}\xi_{lN_w+k}^*\}\mathrm{E}\{e^{-j2\pi f(\Delta_{lN_w+k}-\Delta_{lN_w+k})}\}=R_\xi(0) \quad (3.53)$$

for $mN_w+k'-k=0$, (note l becomes a "dummy" index upon which the result does not depends). From Equations (3.34), (3.35) and (3.36)

$$\begin{aligned}G_2(m,k,k',-f,f)&=\mathrm{E}\{\xi_{lN_w+k}\xi_{(l+m)N_w+k'}^*\}\mathrm{E}\{e^{-j2\pi f(\Delta_{(l+m)N_w+k'}-\Delta_{lN_w+k})}\}\\&=R_\xi(mN_w+k'-k)\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}e^{-j2\pi f(\Delta'-\Delta)}g_{\Delta,\Delta'}(\Delta,\Delta';|mN_w+k'-k|)d\Delta d\Delta'\end{aligned} \quad (3.54)$$

for $mN_w+k'-k\neq 0$. Finally from Equation (3.39)

$$G_3(n,m,k,k',-f,f)=a_{nN_w+k}a_{(n+m)N_w+k'}^*e^{-j2\pi fT_c(c_{(n+m)N_w+k'}-c_{nN_w+k})} \quad (3.55)$$

Although Equation (3.50) could be used in its current form for generic PSD evaluations, it does not differentiate between the PSD's continuous and discrete parts nor provides any insight on the influence of the different variables in the PSD shape. Therefore, in order to obtain the continuous and discrete parts of the PSD let us now separate Equation (3.50) in four terms:

$$\bar{S}(f)=S_{A.1}(f)+S_{A.2}(f)+S_{C.1}(f)+S_{C.2}(f) \quad (3.56)$$

$$S_{A.1}(f)=\frac{1}{T_s\Lambda}\sum_{k,k'=0}^{N_w-1}\sum_{n=0}^{\Lambda-1}\sum_{\substack{m=-\infty\\m\neq 0}}^{\infty}e^{-j2\pi f(k'-k)T_r}\left(\sum_{i,i'=0}^{N_\sigma-1}W_{i,k}(-f)W_{i',k'}{}^*(-f)\pi_i\pi_{i'}\right)e^{-j2\pi fT_s m}$$
$$G_2(m,k,k',-f,f)G_3(n,m,k,k',-f,f) \quad (3.57)$$

$$S_{A.2}(f)=\frac{1}{T_s\Lambda}\sum_{k,k'=0}^{N_w-1}\sum_{n=0}^{\Lambda-1}\sum_{\substack{m=-\infty\\m\neq 0}}^{\infty}e^{-j2\pi f(k'-k)T_r}\left(\sum_{i,i'=0}^{N_\sigma-1}W_{i,k}(-f)W_{i',k'}^*(-f)\pi_i\vartheta_{ii'}^{(|m|)}\right)e^{-j2\pi fT_s m}$$
$$G_2(m,k,k',-f,f)G_3(n,m,k,k',-f,f) \quad (3.58)$$

$$S_{C.1}(f)=\frac{1}{T_s\Lambda}\sum_{k=0}^{N_w-1}\sum_{n=0}^{\Lambda-1}e^{-j2\pi f(k-k)T_r}G_1(0,k,k,-f,f)G_2(0,k,k,-f,f)G_3(n,0,k,k,-f,f) \quad (3.59)$$

$$S_{C.2}(f)=\frac{1}{T_s\Lambda}\sum_{\substack{k,k'=0\\k\neq k'}}^{N_w-1}\sum_{n=0}^{\Lambda-1}e^{-j2\pi f(k'-k)T_r}G_1(0,k,k',-f,f)G_2(0,k,k',-f,f)G_3(n,0,k,k',-f,f) \quad (3.60)$$

where Equation (3.52) was used to obtain Equations (3.57) and (3.58). By replacing Equations (3.51),(3.53) and (3.55) in Equation (3.59) we obtain

$$S_{C.1}(f)=\frac{1}{T_s\Lambda}\sum_{k=0}^{N_w-1}\sum_{n=0}^{\Lambda-1}\sum_{i=0}^{N_\sigma-1}R_\xi(0)\,|\,a_{nN_w+k}\,|^2|\,W_{i,k}(-f)|^2\,\pi_i \tag{3.61}$$

Now note that when $N_w=1$, $S_{C.2}(f)$ becomes zero and when $N_w>1$ Equation (3.60) can be rewritten as

$$S_{C.2}(f)=\tfrac{1}{T_s\Lambda}\sum_{k=0}^{N_w-2}\sum_{k'=k+1}^{N_w-1}\sum_{n=0}^{\Lambda-1}2\,\mathrm{Re}\{e^{-j2\pi f(k'-k)T_r}G_1(0,k,k',-f,f)G_2(0,k,k',-f,f)\,G_3(n,0,k,k',-f,f)\} \tag{3.62}$$

After some algebraic work using the periodicity of $G_3(n,m,k,k',-f,f)$ as described in Equation (3.40), $S_{A.2}(f)$ can be expressed as

$$S_{A.2}(f)=S_{C.3}(f)=\tfrac{2}{T_s\Lambda}\mathrm{Re}\Big\{\sum_{k,k'=0}^{N_w-1}\sum_{n=0}^{\Lambda-1}\sum_{m=1}^{\infty}\sum_{i,i'=0}^{N_\sigma-1}e^{-j2\pi f(k'-k)T_r}W_{i,k}(-f)W^*_{i',k'}(-f)\pi_i\vartheta_{ii'}^{(|m|)}\;e^{-j2\pi fT_sm}\,G_2(m,k,k',-f,f)a_{nN_w+k}a^*_{(n+m)N_w+k'}e^{-j2\pi fT_c(c_{(n+m)N_w+k'}-c_{nN_w+k})}\Big\} \tag{3.63}$$

Next by subtracting and adding

$$\mathrm{E}\{\xi_{lN_w+k}\}\mathrm{E}\{\xi^*_{(l+m)N_w+k'}\}\mathrm{E}\{e^{j2\pi f\Delta_{lN_w+k}}\}\mathrm{E}\{e^{-j2\pi f\Delta_{(l+m)N_w+k'}}\}=\mu_\xi\mu_{\xi^*}\int_{-\infty}^{\infty}e^{j2\pi f\Delta}g_\Delta(\Delta)d\Delta\int_{-\infty}^{\infty}e^{-j2\pi f\Delta}g_\Delta(\Delta)d\Delta=|\mu_\xi|^2\,G_\Delta(f)G^*_\Delta(f)=|\mu_\xi|^2|\,G_\Delta(f)|^2 \tag{3.64}$$

to $G_2(m,k,k',-f,f)$ in Equation (3.57), $S_{A.1}(f)$ it can be separated as

$$S_{A.1}(f)=S_{A.1.1}(f)+S_{A.1.2}(f)$$

$$S_{A.1.1}(f)=\tfrac{1}{T_s\Lambda}\sum_{k,k'=0}^{N_w-1}\sum_{n=0}^{\Lambda-1}\sum_{\substack{m=-\infty\\m\neq0}}^{\infty}e^{-j2\pi f(k'-k)T_r}\sum_{i,i'=0}^{N_\sigma-1}W_{i,k}(-f)W^*_{i',k'}(-f)\pi_i\pi_{i'}e^{-j2\pi fT_sm}\{G_2(m,k,k',-f,f)-|\mu_\xi|^2|\,G_\Delta(f)|^2\}G_3(n,m,k,k',-f,f) \tag{3.65}$$

$$S_{A.1.2}(f)=\tfrac{1}{T_s\Lambda}\sum_{k,k'=0}^{N_w-1}\sum_{n=0}^{\Lambda-1}\sum_{\substack{m=-\infty\\m\neq0}}^{\infty}e^{-j2\pi f(k'-k)T_r}\sum_{i,i'=0}^{N_\sigma-1}W_{i,k}(-f)W^*_{i',k'}(-f)\pi_i\pi_{i'}\,e^{-j2\pi fT_sm}|\mu_\xi|^2|\,G_\Delta(f)|^2\,G_3(n,m,k,k',-f,f) \tag{3.66}$$

In a similar way to $S_{A.2}(f)$, using the periodicity of $G_3(n,m,k,k',-f,f)$, $S_{A.1.1}(f)$ can be expressed as

$$S_{A.1.1}(f)=S_{C.4}(f)=\tfrac{2}{T_s\Lambda}\mathrm{Re}\Big\{\sum_{k,k'=0}^{N_w-1}\sum_{n=0}^{\Lambda-1}\sum_{m=1}^{\infty}\sum_{i,i'=0}^{N_\sigma-1}e^{-j2\pi f(k'-k)T_r}W_{i,k}(-f)W^*_{i',k'}(-f)\pi_i\pi_{i'}e^{-j2\pi fT_sm}\{G_2(m,k,k',-f,f)-|\mu_\xi|^2|\,G_\Delta(f)|^2\}a_{nN_w+k}a^*_{(n+m)N_w+k'}e^{-j2\pi fT_c(c_{(n+m)N_w+k'}-c_{nN_w+k})}\Big\} \tag{3.67}$$

Next by adding and subtracting the term at $m=0$ in Equation (3.66), $S_{A.1.2}(f)$ can be expressed as

$$S_{A.1.2}(f)=S_{A.1.2.1}(f)+S_{A.1.2.2}(f)$$

$$S_{A.1.2.1}(f)=\tfrac{|\mu_\xi|^2|G_\Delta(f)|^2}{T_s\Lambda}\sum_{k,k'=0}^{N_w-1}e^{-j2\pi f(k'-k)T_r}\sum_{i,i'=0}^{N_\sigma-1}W_{i,k}(-f)W^*_{i',k'}(-f)\pi_i\pi_{i'}\sum_{n=0}^{\Lambda-1}\sum_{m=-\infty}^{\infty}e^{-j2\pi fT_sm}G_3(n,m,k,k',-f,f) \tag{3.68}$$

$$S_{A.1.2.2}(f)=-\tfrac{|\mu_\xi|^2|G_\Delta(f)|^2}{T_s\Lambda}\sum_{k,k'=0}^{N_w-1}e^{-j2\pi f(k'-k)T_r}\sum_{i,i'=0}^{N_\sigma-1}W_{i,k}(-f)W^*_{i',k'}(-f)\pi_i\pi_{i'}\sum_{n_1=0}^{\Lambda-1}G_3(n_1,0,k,k',-f,f) \tag{3.69}$$

By substituting Equation (3.55) in $S_{A.1.2.2}(f)$ and reordering we get

$$S_{A.1.2.2}(f)=S_{C.5}(f)=-\frac{|\mu_\xi|^2|G_\Delta(f)|^2}{T_s\Lambda}\sum_{n=0}^{\Lambda-1}\left|\sum_{k=0}^{N_w-1}\sum_{i=0}^{N_\sigma-1}e^{j2\pi fkT_r}a_{nN_w+k}e^{j2\pi fT_cc_{nN_w+k}}W_{i,k}(-f)\pi_i\right|^2 \tag{3.70}$$

Working with the sums on *n* and *m* in Equation (3.68) we can find that

$$\begin{aligned}\sum_{n=0}^{\Lambda-1}\sum_{m=-\infty}^{\infty}G_3(n,m,k,k',-f,f)e^{-j2\pi fmT_s}=&\sum_{n=0}^{\Lambda-1}a_{nN_w+k}e^{j2\pi fT_cc_{nN_w+k}}e^{j2\pi fnT_s}\sum_{l=0}^{\Lambda-1}a^*_{lN_w+k'}e^{-j2\pi fT_cc_{lN_w+k'}}e^{-j2\pi flT_s}\\&\times\sum_{l'=-\infty}^{\infty}e^{-j2\pi fl'\Lambda T_s}\end{aligned} \tag{3.71}$$

using Equation (3.44) in Equation (3.71), substituting the result in Equation (3.68) and simplifying $S_{A.1.2.1}(f)$ becomes

$$\begin{aligned}S_{A.1.2.1}(f)=S_D(f)=&\frac{|\mu_\xi|^2|G_\Delta(f)|^2}{(T_s\Lambda)^2}\left|\sum_{k=0}^{N_w-1}\sum_{i=0}^{N_\sigma-1}\sum_{n=0}^{\Lambda-1}e^{j2\pi fkT_r}W_{i,k}(-f)\pi_ia_{nN_w+k}e^{j2\pi fT_cc_{nN_w+k}}e^{j2\pi fnT_s}\right|^2\\&\times\sum_{r=-\infty}^{\infty}\delta(f-\tfrac{r}{T_s\Lambda})\end{aligned} \tag{3.72}$$

which is the PSD's discrete part.

Therefore, grouping all together, ***the PSD of the convolutionally coded/Markov-driven IR-based UWB signal model*** introduced in Section 3.3 is given by

$$\bar{S}(f)=S_C(f)+S_D(f) \tag{3.73}$$

$$\begin{aligned}S_C(f)&=S_{C.1}(f)+S_{C.3}(f)+S_{C.4}(f)+S_{C.5}(f) \qquad N_w=1\\S_C(f)&=S_{C.1}(f)+S_{C.2}(f)+S_{C.3}(f)+S_{C.4}(f)+S_{C.5}(f) \qquad N_w>1\end{aligned} \tag{3.74}$$

$$S_{C.1}(f)=\frac{1}{T_s\Lambda}\sum_{k=0}^{N_w-1}\sum_{n=0}^{\Lambda-1}\sum_{i=0}^{N_\sigma-1}R_\xi(0)|a_{nN_w+k}|^2|W_{i,k}(-f)|^2\pi_i \tag{3.75}$$

$$S_{C.2}(f)=\frac{2}{T_s\Lambda}\sum_{k=0}^{N_w-2}\sum_{k'=k+1}^{N_w-1}\sum_{n=0}^{\Lambda-1}\mathrm{Re}\{e^{-j2\pi f(k'-k)T_r}G_1(0,k,k',-f,f)G_2(0,k,k',-f,f)G_3(n,0,k,k',-f,f)\} \tag{3.76}$$

$$\begin{aligned}S_{C.3}(f)=\frac{2}{T_s\Lambda}\mathrm{Re}\Big\{&\sum_{k,k'=0}^{N_w-1}\sum_{n=0}^{\Lambda-1}\sum_{m=1}^{\infty}\sum_{i,i'=0}^{N_\sigma-1}e^{-j2\pi fT_sm}e^{-j2\pi f(k'-k)T_r}W_{i,k}(-f)W^*_{i',k'}(-f)\pi_i\vartheta^{(|m|)}_{ii'}\\&G_2(m,k,k',-f,f)G_3(n,m,k,k',-f,f)\Big\}\end{aligned} \tag{3.77}$$

$$\begin{aligned}S_{C.4}(f)=\frac{2}{T_s\Lambda}\mathrm{Re}\Big\{&\sum_{k,k'=0}^{N_w-1}\sum_{n=0}^{\Lambda-1}\sum_{m=1}^{\infty}\sum_{i,i'=0}^{N_\sigma-1}e^{-j2\pi fT_sm}e^{-j2\pi f(k'-k)T_r}W_{i,k}(-f)W^*_{i',k'}(-f)\pi_i\pi_{i'}\\&\{G_2(m,k,k',-f,f)-|\mu_\xi|^2|G_\Delta(f)|^2\}G_3(n,m,k,k',-f,f)\Big\}\end{aligned} \tag{3.78}$$

$$S_{C.5}(f)=-\frac{|\mu_\xi|^2|G_\Delta(f)|^2}{T_s\Lambda}\sum_{n=0}^{\Lambda-1}\left|\sum_{k=0}^{N_w-1}\sum_{i=0}^{N_\sigma-1}e^{j2\pi fkT_r}a_{nN_w+k}e^{j2\pi fT_cc_{nN_w+k}}W_{i,k}(-f)\pi_i\right|^2 \tag{3.79}$$

$$S_D(f)=\frac{|\mu_\xi|^2|G_\Delta(f)|^2}{(T_s\Lambda)^2}\left|\sum_{k=0}^{N_w-1}\sum_{i=0}^{N_\sigma-1}\sum_{n=0}^{\Lambda-1}e^{j2\pi fnT_s}e^{j2\pi fkT_r}a_{nN_w+k}e^{j2\pi fT_cc_{nN_w+k}}W_{i,k}(-f)\pi_i\right|^2\sum_{r=-\infty}^{\infty}\delta(f-\tfrac{r}{T_s\Lambda}) \tag{3.80}$$

where μ_ξ is the mean of ξ_{lN_w+k} and $G_\Delta(f)$ is the Fourier transform of the first order probability density function of Δ_{lN_w+k}.

Equations (3.73) to (3.80) are the closed form expressions for the PSD of convolutionally coded/Markov-driven IR-based UWB signals. Another form for Equation (3.74) (which may be useful for numerical evaluation of the PSD by a computer) is obtained by defining

$$G_4(k,k',f)=\sum_{i,i'=0}^{N_\sigma-1}W_{i,k}(-f)W^*_{i',k'}(-f)\pi_i\pi_{i'} \tag{3.81}$$

and merging $S_{C.3}(f)$ and $S_{C.4}(f)$ into a single term

$$\begin{aligned}S_{C.3}(f)+S_{C.4}(f)=\tfrac{2}{T_s\Lambda}\mathrm{Re}\Big\{&\sum_{k,k'=0}^{N_w-1}\sum_{n=0}^{\Lambda-1}\sum_{m=1}^{\infty}e^{-j2\pi f(k'-k)T_r}e^{-j2\pi fT_sm}a_{nN_w+k}a^*_{(n+m)N_w+k'}e^{-j2\pi fT_c(c_{(n+m)N_w+k'}-c_{nN_w+k})}\\&\{\sum_{i,i'=0}^{N_\sigma-1}W_{i,k}(-f)W^*_{i',k'}(-f)\pi_i(\pi_{i'}+\vartheta_{ii'}^{(|m|)})G_2(m,k,k',-f,f)-G_4(k,k',f)|\mu_\xi|^2|G_\Delta(f)|^2\}\Big\}\end{aligned} \tag{3.82}$$

to obtain

$$\begin{aligned}S_{C.6}(f)=\frac{2}{T_s\Lambda}\mathrm{Re}\Big\{&\sum_{k,k'=0}^{N_w-1}\sum_{n=0}^{\Lambda-1}\sum_{m=1}^{\infty}e^{-j2\pi f(k'-k)T_r}e^{-j2\pi fT_sm}a_{nN_w+k}a^*_{(n+m)N_w+k'}e^{-j2\pi fT_c(c_{(n+m)N_w+k'}-c_{nN_w+k})}\\&\{G_1(m,k,k',-f,f)G_2(m,k,k',-f,f)-G_4(k,k',f)|\mu_\xi|^2|G_\Delta(f)|^2\}\Big\}\end{aligned} \tag{3.83}$$

Then

$$S_C(f)=S_{C.1}(f)+S_{C.2}(f)+S_{C.6}(f)+S_{C.5}(f) \tag{3.84}$$

It is useful to note that while the continuous part of the PSD is a function of the steady state and n^{th} step transition probabilities, ***the discrete part is only a function of the steady state probabilities. This means that, regardless of the n^{th} step transition probabilities' convergence rate to the steady state probabilities, the discrete part of the spectrum can be shaped by the encoder's steady state probabilities.*** Furthermore note that the spectral lines in the PSD are spaced at multiples of $1/(T_s\Lambda)$, showing dependence on the periodicity of χ_a, χ_c and the number of pulses, N_w, per encoder output vector used.

3.4.3 Simplifications for Special Cases of Modulation Formats, Attenuation/Jitter Statistics and Time Hopping/Direct Sequence

1. When only one basic pulse shape is used with PAM, PPM or BOPPM then

$$w_{\sigma_l,k}(t)=\alpha_{\sigma_l,k}w(t-\beta_{\sigma_l,k}T_\beta) \tag{3.85}$$

where T_β is the PPM modulation index. Note $\alpha_{\sigma_l,k}$ and $\beta_{\sigma_l,k}$ are the k^{th} PAM and PPM variables used for the transmission of the l^{th} encoder output vector, $z_l=\gamma(\sigma_l)$, whose value depends on the SE pair MC state at time l. Therefore in the frequency domain

$$\begin{aligned}W_{\sigma_l,k}(f)&=\alpha_{\sigma_l,k}W(f)e^{-j2\pi f\beta_{\sigma_l,k}T_\beta}\\W_{\sigma_l,k}(-f)&=\alpha_{\sigma_l,k}W(-f)e^{j2\pi f\beta_{\sigma_l,k}T_\beta}\end{aligned} \tag{3.86}$$

and as a consequence

$$\begin{aligned}W_{i,k}(-f)W^*_{i',k'}(-f)&=\alpha_{i,k}W(-f)e^{j2\pi f\beta_{i,k}T_\beta}\alpha^*_{i',k'}W^*(-f)e^{-j2\pi f\beta_{i',k'}T_\beta}\\&=|W(f)|^2\,\alpha_{i,k}\alpha^*_{i',k'}e^{-j2\pi f(\beta_{i',k'}-\beta_{i,k})T_\beta}\end{aligned} \tag{3.87}$$

for state s_i. Therefore

$$G_1(m,k,k',-f,f) = |W(f)|^2 \, G_{AP}(m,k,k',-f,f) \tag{3.88}$$

where

$$G_{AP}(m,k,k',-f,f) = \begin{cases} \sum_{i=0}^{N_\sigma-1} \alpha_{i,k}\alpha^*_{i,k'} e^{-j2\pi f(\beta_{i,k'}-\beta_{i,k})T_\beta}\pi_i & m=0 \\ \sum_{i,i'=0}^{N_\sigma-1} \alpha_{i,k}\alpha^*_{i',k'} e^{-j2\pi f(\beta_{i',k'}-\beta_{i,k})T_\beta}\pi_i p_{ii'}^{(|m|)} & m\neq 0 \end{cases} \tag{3.89}$$

2. If the attenuation and jitter are i.i.d. processes then

$$G_2(m,k,k',-f,f) = \begin{cases} \mathrm{E}\{|\xi|^2\}, & mN_w+k'-k=0 \\ |\mu_\xi|^2 |G_\Delta(f)|^2, & mN_w+k'-k\neq 0 \end{cases} \tag{3.90}$$

and $S_{C.4}(f)$ becomes zero.

3. If PR-TH sequence is assumed to be random then our result can be used by "merging" the jitter process Δ_{lN_w+k}, and $c_{lN_w+k}T_c$ into a single random dither variable Δ'_{lN_w+k}, that is

$$\Delta'_{lN_w+k} = \Delta_{lN_w+k} + c_{lN_w+k}T_c \tag{3.91}$$

and

$$\begin{aligned} \chi_c &= 1 \\ \chi_{acw} &= \mathrm{lcm}(\chi_a, N_w) \end{aligned} \tag{3.92}$$

then

$$\begin{aligned} G_2(m,k,k',-f,f) &= \mathrm{E}\{\xi_{lN_w+k}\xi^*_{(l+m)N_w+k}\}\mathrm{E}\{e^{-j2\pi f(\Delta'_{(l+m)N_w+k'}-\Delta'_{lN_w+k})}\} \\ &= \mathrm{E}\{\xi_{lN_w+k}\xi^*_{(l+m)N_w+k}\}\mathrm{E}\{e^{-j2\pi f(\Delta_{(l+m)N_w+k'}+c_{(l+m)N_w+k'}T_c-\Delta_{lN_w+k}-c_{lN_w+k}T_c)}\} \\ &= \mathrm{E}\{\xi_{lN_w+k}\xi^*_{(l+m)N_w+k}\}\mathrm{E}\{e^{-j2\pi f(\Delta_{(l+m)N_w+k'}-\Delta_{lN_w+k})}\}\mathrm{E}\{e^{-j2\pi f(c_{(l+m)N_w+k'}T_c-c_{lN_w+k}T_c)}\} \end{aligned} \tag{3.93}$$

since Δ_{lN_w+k}, and $c_{lN_w+k}T_c$ are independent.

Due to the PR-TH sequence is random, $c_{lN_w+k}T_c$ is a sequence of uniform i.i.d. random variables with distribution

$$g_{cT_c}(\phi) = \sum_{i=0}^{N_c-1} \frac{1}{N_c}\delta(\phi - iT_c) \tag{3.94}$$

and

$$\mathrm{E}\{e^{-j2\pi f(c_{(l+m)N_w+k'}T_c-c_{lN_w+k}T_c)}\} = |G_{cT_c}(f)|^2 = \frac{1}{N_c^2}\left|\sum_{\varsigma=0}^{N_c-1} e^{-j2\pi f\varsigma T_c}\right|^2 = \frac{1}{N_c} + \frac{2}{N_c^2}\sum_{i=1}^{N_c-1}(N_c-l)\cos(2\pi i f T_c) \tag{3.95}$$

for $mN_w+k'-k\neq 0$. A similar approach can be followed for a perfect direct sequence (DS) by merging ξ_{lN_w+k} and a_{lN_w+k}.

3.5 Application Examples for the Spectral Analysis of Convolutionally Coded/Markov-Driven IR-Based UWB Signals

To illustrate the application of Equations (3.73) to (3.80), several PSD plots are introduced in this section for two different encoders coupled with quaternary BOPPM (Q-BOPPM) and quaternary PPM (Q-PPM). First, PSD plots obtained when using the recursive systematic convolutional encoder defined in Equation (3.6) are introduced. Next, results obtained when using the binary data stream generated by the BMS introduced in Section 3.2.1 without coding are presented. Finally, the effects of interchanging the encoder's generators order for a maximum free distance (MFD), rate 1/4, feedforward non-systematic binary convolutional encoder are shown. Note more application examples will be shown throughout the rest of the thesis as the results presented in this chapter will be frequently used in the following chapters.

In order to be able to compare the analytical and simulated results, a double simulation-PSD estimation procedure has been performed for PSDs with continuous and discrete components (that is $S_C(f) \neq 0$ and $S_D(f) \neq 0$) as described in Appendix A. Thus, in several of the PSD plots presented in this and the following chapters, two separate simulations using periodogram-discrete Fourier transform (DFT) based PSD estimation have been performed. In these plots the simulation results labelled as "Sim. $|X_w(0)|^2=1$" were obtained using a data window, $x_w(t)$, with the frequency domain characteristic $|X_w(0)|^2=1$, where $X_w(f)=\Im\{x_w(t)\}$. The results from these simulations have been used to validate the analytical results for the PSD's discrete part (spectral lines). On the other side, the continuous part of the PSD has been validated through the use of simulations with unit energy data windows. The results from these simulations have been labelled as "Sim. U. E.". The reader is referred to Appendix A for more details on these procedures.

3.5.1 Rate 1/2 Recursive Systematic Convolutional Encoder Defined by Equation (3.6)

Consider the encoder defined by Equation (3.6) with SE Markov model defined in Equation (3.16) and steady state probabilities given in Equation (3.18). Using this information the PSD of the convolutionally coded IR-based UWB signal for Q-BOPPM and quaternary PPM will be evaluated for several particular cases.

3.5.1.1 Q-BOPPM with TH, DS and Jitter

For this case only one quaternary biorthogonal signal is needed to transmit one encoder output vector as each vector consists of two bits per state, thus $N_w=1$ and $T_s=T_r$. Therefore, if the encoder is in state $\sigma_l=s_i$ at time l with corresponding encoder output vector

$z_l = \underline{\mathbf{z}_l} = [z_l^{(0)}, z_l^{(1)}] = \gamma(\sigma_l = s_i) = \underline{\zeta_{s_i}} = [\zeta_{s_i}^{(0)}, \zeta_{s_i}^{(1)}]$, then

$$w_{\sigma_l}(t) = w_{s_i}(t) = \alpha_i w(t - \beta_i T_\beta) \tag{3.96}$$

with $\alpha_i = (2\zeta_{s_i}^{(0)} - 1)$ and $\beta_i = \zeta_{s_i}^{(1)}$, and where $\alpha_i \in \{-1,1\}$ and $\beta_i \in \{0,1\}$.

Set the TH and DS sequences to be {0, 2, 4, 1, 3} (prime code over GF(5)) and {-1, 1, -1, 1, -1, -1, 1, 1, -1, 1 } respectively, thus $\chi_c = 5$, $\chi_a = 10$, $\chi_{acw} = 10$ and $\Lambda = 10$. Finally assume there is no pulse attenuation and that the jitter is a sequence of i.i.d. random variables uniformly distributed in the interval (0, T_u), then

$$G_2(m,k,k',-f,f) = \begin{cases} 1, & mN_w + k' - k = 0 \\ \operatorname{sinc}^2(T_u f), & mN_w + k' - k \neq 0 \end{cases} \tag{3.97}$$

Using the results of Section 3.4.3 and substituting Equation (3.97) into the PSD formulas yields

$$S_D(f) = \frac{\operatorname{sinc}^2(T_u f)|W(f)|^2}{100T_s^2} \left| \sum_{i=0}^{7} \sum_{n=0}^{9} e^{j2\pi f n T_s} a_{nN_w} e^{j2\pi f T_c c_{nN_w}} \alpha_i e^{j2\pi f \beta_i T_\beta} \pi_i \right|^2 \sum_{r=-\infty}^{\infty} \delta\left(f - \frac{r}{10T_s}\right) \tag{3.98}$$

and

$$\begin{aligned} S_C(f) = {} & \frac{1}{T_s}|W(f)|^2 - \frac{\operatorname{sinc}^2(T_u f)|W(f)|^2}{10T_s} \sum_{n=0}^{9} \left| \sum_{i=0}^{7} a_{nN_w} e^{j2\pi f T_c c_{nN_w}} \alpha_i e^{j2\pi f \beta_i T_\beta} \pi_i \right|^2 \\ & + \frac{2\operatorname{sinc}^2(T_u f)|W(f)|^2}{10T_s} \operatorname{Re}\left\{ \sum_{n=0}^{9} \sum_{m=1}^{\infty} e^{-j2\pi f T_s m} a_{nN_w} a^*_{(n+m)N_w} e^{-j2\pi f T_c (c_{(n+m)N_w} - c_{nN_w})} \right. \\ & \left. \times \sum_{i,i'=0}^{7} \alpha_i \alpha_{i'}^* e^{-j2\pi f (\beta_{i'} - \beta_i) T_\beta} \pi_i \{ p_{ii'}^{(|m|)} - \pi_{i'} \} \right\} \end{aligned} \tag{3.99}$$

The PSD plots obtained by using Equations (3.98) and (3.99) are shown in Figure 3-6 for $p_{y,10} = 3/5$ and different values of $p_{y,01}$ with T_u = 0 ns. For $p_{y,01} = 3/5$ (Figures 3-6a and 3-6b) the SE pair MC steady state probabilities are

$$\underline{\pi_\sigma} = [\pi_0, \pi_1, \ldots, \pi_7] = \left[\tfrac{1}{8}, \tfrac{1}{8}, \tfrac{1}{8}, \tfrac{1}{8}, \tfrac{1}{8}, \tfrac{1}{8}, \tfrac{1}{8}, \tfrac{1}{8}\right] \tag{3.100}$$

and therefore the discrete part, $S_D(f)$, and the negative term in $S_C(f)$ becomes zero. This means that asymptotically (that is in steady state) the encoder's output resembles a sequence of uniform distributed i.i.d. symbols. Nevertheless the shape of $S_C(f)$ is not an exact replica of the pulse's energy density spectrum (ESD) due to the jitter and the MC convergence rate to steady state as evidenced by the third term in Equation (3.99) and the small ripples in the PSD plot of Figure 3-6a (magnification in Figure 3-6b).

Power spectral density (PSD) plots obtained when setting $p_{y,01} = 2/5$ and $p_{y,01} = 1/5$ are shown in Figures 3-6c to 3-6f. For these cases spectral lines appear at $1/(\Lambda T_s) = 10$ MHz intervals. The appearance of spectral lines when setting $p_{y,01} = 2/5$ (Figures 3-6c and 3-6d) and $p_{y,01} = 1/5$ (Figures 3-6d and 3-6f) is due to the change in the stationary probabilities from Equation (3.100) to

$$\underline{\pi_\sigma} = \tfrac{1}{4}\left[\tfrac{3}{5}, \tfrac{3}{5}, \tfrac{3}{5}, \tfrac{3}{5}, \tfrac{2}{5}, \tfrac{2}{5}, \tfrac{2}{5}, \tfrac{2}{5}\right] \text{ and } \underline{\pi_\sigma} = \tfrac{1}{4}\left[\tfrac{3}{4}, \tfrac{3}{4}, \tfrac{3}{4}, \tfrac{3}{4}, \tfrac{1}{4}, \tfrac{1}{4}, \tfrac{1}{4}, \tfrac{1}{4}\right] \tag{3.101}$$

respectively, which are no longer uniform like distributed.

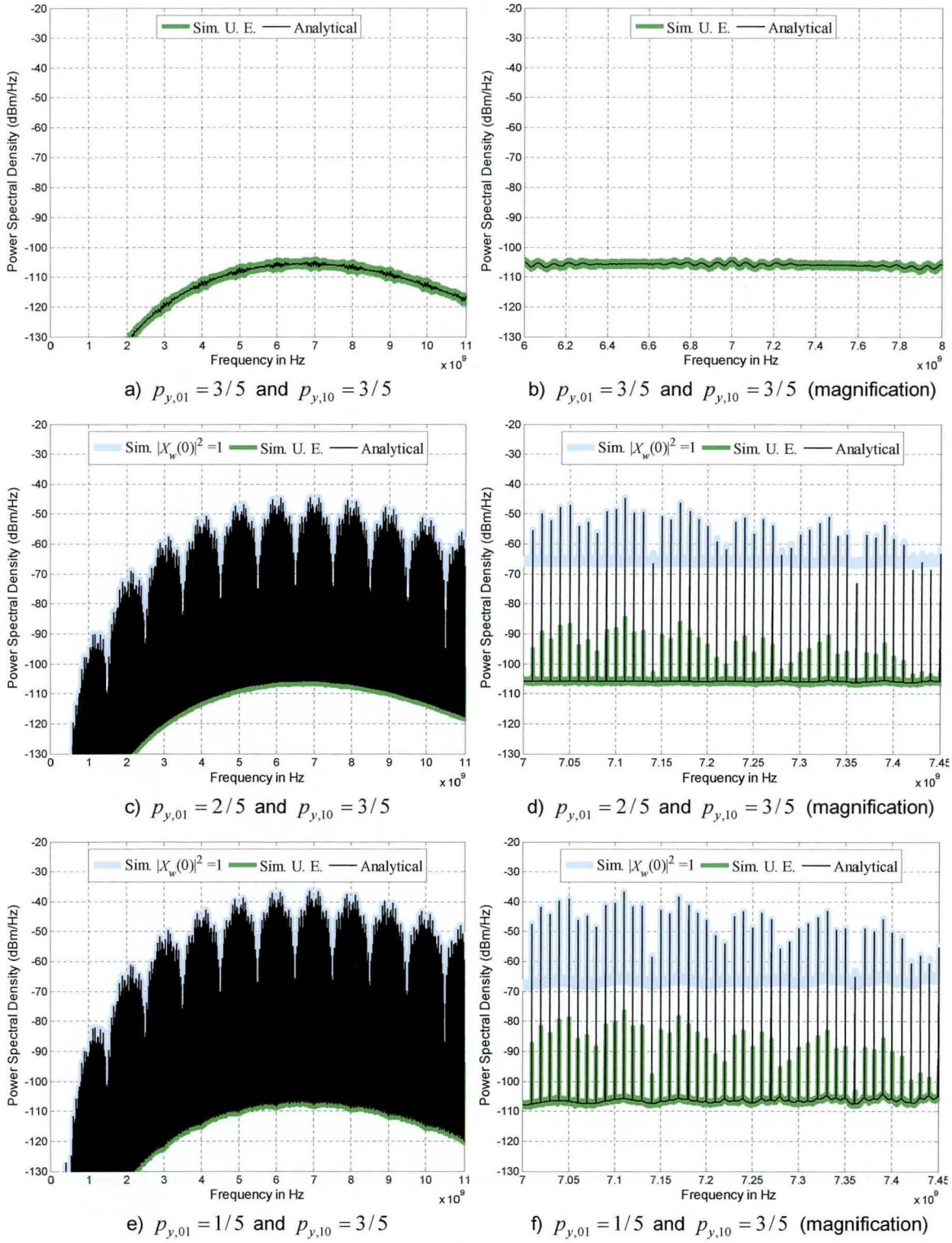


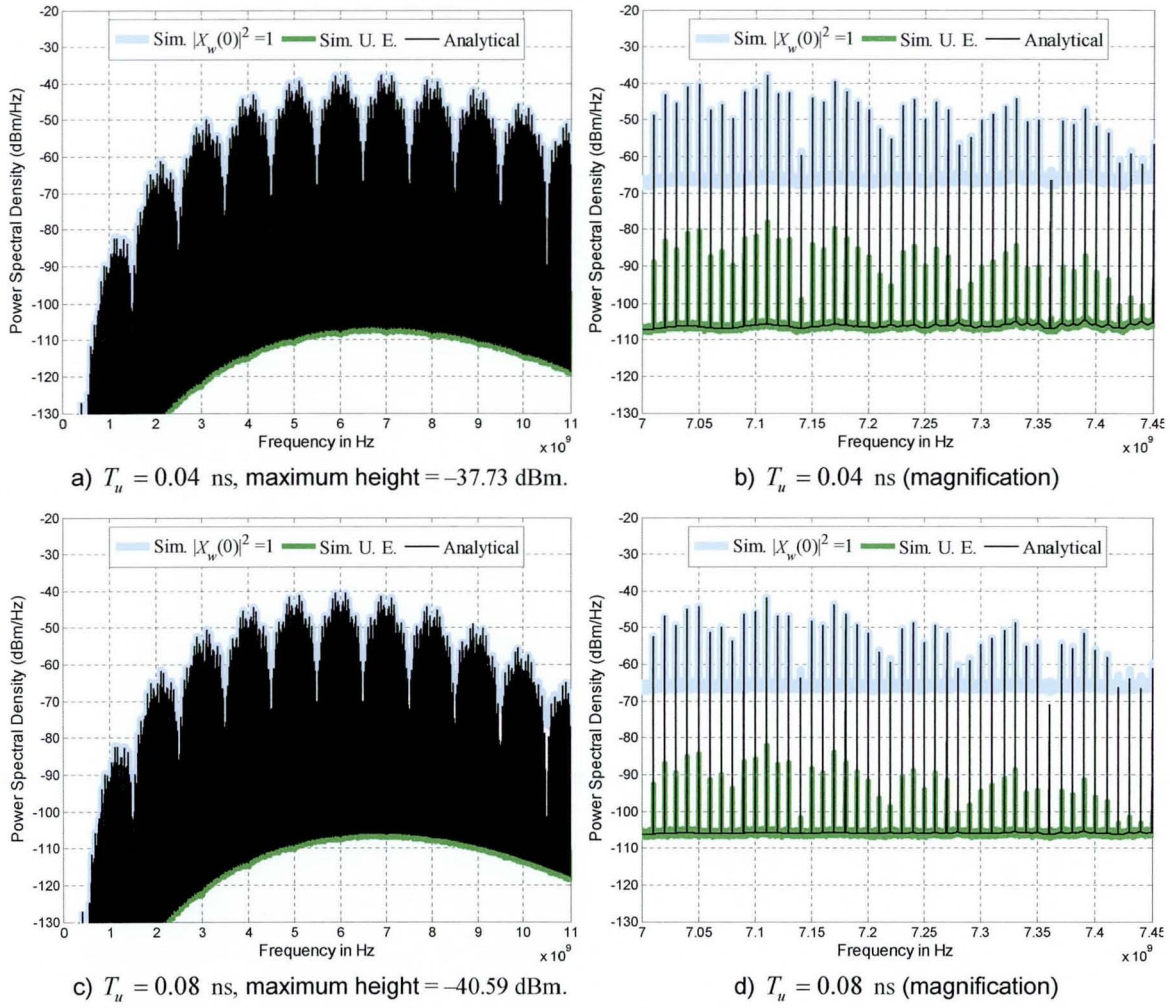
Figure 3-6. Analytical and simulated PSDs obtained when using the recursive systematic binary convolutional encoder defined by Equations (3.6) and (3.16) in a Q-BOPPM IR-based UWB system. The signal parameters are $T_s = 10$ ns, $T_r = 10$ ns, $T_c = 2$ ns, $T_\beta = 1$ ns and no jitter. The 4th derivative Gaussian pulse is used with duration $T_w \approx 0.4$ ns. The sequences {-1, 1, -1, 1, -1, -1, 1, 1, -1, 1 } and {0, 2, 4, 1, 3} were used for DS and TH respectively.

In order further clarify the appearance of spectral lines, note that the absolute value in Equation (3.98) for $p_{y,01} = 2/5$ (Figures 3-6c and 3-6d) becomes

$$\left| \sum_{i=0}^{7} \sum_{n=0}^{9} e^{j2\pi f n T_s} a_{nN_w} e^{j2\pi f l T_c c_{nN_w}} \alpha_i e^{j2\pi f \beta_i T_\beta} \pi_i \right|^2 = \left| \sum_{n=0}^{9} e^{j2\pi f n T_s} a_{nN_w} e^{j2\pi f l T_c c_{nN_w}} \left\{ -\tfrac{1}{10} - \tfrac{1}{10} e^{j2\pi f l T_\beta} \right\} \right|^2 \qquad (3.102)$$

which is no longer equal to zero for all values of $f = r/(10T_s)$, $r = 0, \pm 1, \pm 2, \ldots$. It is worth mentioning that the appearance of spectral lines was expected due to the encoder is systematic (that is the input data stream appears unchanged in one of the outputs) and the BMS' stationary probabilities are $\underline{\pi_y} = [3/5, 2/5]$ for $p_{y,01} = 2/5$ and $\underline{\pi_y} = [3/4, 1/4]$ for $p_{y,01} = 1/5$, meaning that the BMS generates more "zeros" than "ones".

The effects of timing jitter uniform distributed in the interval (0, T_u) are shown in Figure 3-7 for $T_u = 0.04$ ns and $T_u = 0.08$ ns with $p_{y,01} = 1/5$ and $p_{y,10} = 3/5$. By comparing Figure 3-6e (maximum spectral line height equal to –36.62 dBm) with Figure 3-7a (maximum spectral line height equal to –37.73 dBm) and Figure 3-7c (maximum spectral line height equal to –40.59 dBm) it can be seen that the jitter helps to reduce the spectral lines height. Nevertheless the use of timing jitter is not advisable for spectral shaping purposes. This is due to the presence of timing jitter has adverse effects in the system's bit error rate (BER) performance as explained in [125].



a) $T_u = 0.04$ ns, maximum height = –37.73 dBm.

b) $T_u = 0.04$ ns (magnification)

c) $T_u = 0.08$ ns, maximum height = –40.59 dBm.

d) $T_u = 0.08$ ns (magnification)

Figure 3-7. Analytical and simulated PSDs obtained when using the recursive systematic binary convolutional encoder defined in Equations (3.6) and (3.16) with a Q-BOPPM IR-based UWB system. The BMS has one step transition probabilities $p_{y,01}$ = 1/5 and $p_{y,10}$ = 3/5. The signal parameters are T_s = 10 ns, T_r = 10 ns, T_c = 2 ns, T_β = 1 ns and jitter uniform distributed in the interval (0, T_u). The 4th derivative Gaussian pulse is used with duration $T_w \approx 0.4$ ns. The sequences {-1, 1, -1, 1, -1, -1, 1, 1, -1, 1 } and {0, 2, 4, 1, 3} were used for DS and TH respectively.

Finally note how there is good agreement between the analytical and simulated results for both the PSD's continuous part (Sim. U.E.) and the PSD's discrete part (Sim. $|X_w(0)|^2 = 1$) in all the previous figures.

3.5.1.2 Q-PPM with Random TH

In this case one PPM symbol is needed per encoder output vector, thus $N_w = 1$ and $T_s = T_r$. Therefore, if the encoder is in state $\sigma_l = s_i$ at time l with corresponding encoder output vector $z_l = \gamma(\sigma_l = s_i) = \underline{\zeta_{s_i}} = [\zeta_{s_i}^{(0)}, \zeta_{s_i}^{(1)}]$, then

$$w_{\sigma_l}(t) = w_{s_i}(t) = w(t - \beta_i T_\beta) \tag{3.103}$$

with $\beta_i = \zeta_{s_i}^{(0)} + 2\zeta_{s_i}^{(1)}$, and therefore $\beta_i \in \{0,1,2,3\}$.

It is assumed the TH sequence is perfectly random with c_{lN_w+k} uniform distributed over the set {0, 1, 2, 3, 4}. For this example DS multiplication, timing jitter and pulse attenuation are not considered. Therefore using the results of Section 3.4.3 we get

$$G_2(m,k,k',-f,f) = |G_{cT_c}(f)|^2 = \tfrac{1}{5} + \tfrac{2}{25}\sum_{l=1}^{4}(5-l)\cos(2\pi f l T_c) \tag{3.104}$$

and $\chi_a = 1$, $\chi_c = 1$ and $\Lambda = 1$. Then replacing in the PSD formulas yields

$$\begin{aligned} S_C(f) = {} & \tfrac{1}{T_s}|W(f)|^2 + \tfrac{2|G_{cT_c}(f)|^2|W(f)|^2}{T_s}\mathrm{Re}\Big\{\sum_{m=1}^{\infty}\sum_{i,i'=0}^{7} e^{-j2\pi f T_s m} e^{-j2\pi f(\beta_{i'}-\beta_i)T_\beta} \pi_i \{p_{ii'}^{(|m|)} - \pi_{i'}\}\Big\} \\ & - \tfrac{|G_{cT_c}(f)|^2|W(f)|^2}{T_s}\left|\sum_{i=0}^{7} e^{j2\pi f \beta_i T_\beta}\pi_i\right|^2 \end{aligned} \tag{3.105}$$

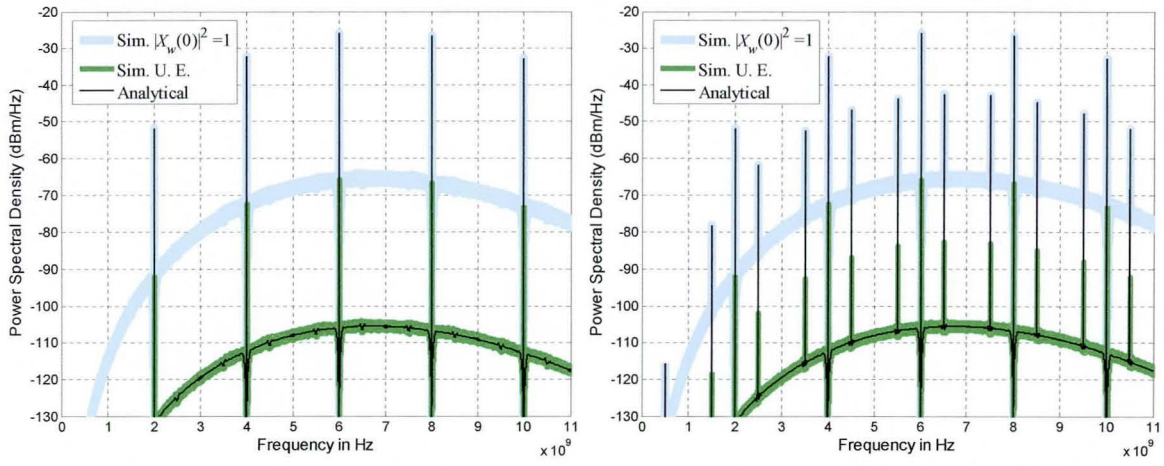
and

$$S_D(f) = \tfrac{|G_{cT_c}(f)|^2|W(f)|^2}{T_s^2}\left|\sum_{i=0}^{7} e^{j2\pi f \beta_i T_\beta}\pi_i\right|^2 \sum_{r=-\infty}^{\infty}\delta(f - \tfrac{r}{T_s}) \tag{3.106}$$

Power spectral density (PSD) plots obtained by using Equations (3.105) and (3.106) for $p_{y,10} = 3/5$ and different values of $p_{y,01}$ are shown in Figure 3-8.

From Equation (3.106) it can be seen that spectral lines could be expected at $1/T_s$ = 100 MHz intervals in Figure 3-8. However, for $p_{y,01} = 3/5$ (Figure 3-8a) spectral lines appear at 2 GHz intervals. This is due to the time hopping sequence, $\{c_{lN_w+k}\}$, is assumed to be perfectly random over the set {0, 1, 2, 3, 4}; $\underline{\pi_\sigma}$ is like in Equation (3.100) and T_β and T_c were set to eliminate as many spectral lines as possible. Note how as soon as $p_{y,01}$ is changed (Figure 3-8b) the number of spectral lines in the PSD trebles as a consequence of the change on the MC steady state probabilities from Equation (3.100) to Equation (3.101).

Again note the good agreement between the analytical and simulated results for both the PSD's continuous part (Sim. U.E.) and the PSD's discrete part (Sim. $|X_w(0)|^2 = 1$) in Figure 3-8.



a) $p_{y,01}=3/5$ and $p_{y,10}=3/5$ b) $p_{y,01}=2/5$ and $p_{y,10}=3/5$

Figure 3-8. Analytical and simulated PSDs obtained when using the recursive systematic binary convolutional encoder defined by Equations (3.6) and (3.16) in a quaternary PPM IR-based UWB system. The signal parameters are $T_s=10$ ns, $T_r=10$ ns, $T_c=2$ ns, $T_\beta=0.5$ ns, c_{lNw+k} uniform distributed over the set {0, 1, 2, 3, 4} and no jitter. The 4th derivative Gaussian pulse is used with duration $T_w\approx 0.4$ ns.

3.5.2 Non-Coded Q-BOPPM with Pulse Repetition

Now consider the BMS introduced in Section 3.2.1 is coupled directly to a Q-BOPPM IR-based UWB system. In this case two information bits can be transmitted per quaternary signal. Nevertheless, in order to provide a meaningful comparison with the previous example, where two quaternary signals are required to transmit two information bits because of the code rate, it will be assumed that each quaternary signal is repeated in two consecutive frames. Thus, this is a pulse repetition scheme where every couple of information bits, $\underline{\mathbf{y}}_l=[y_l^{(0)},y_l^{(1)}]$, is transmitted twice, that is

$$z_l=\underline{\mathbf{z}}_l=[z_l^{(0)},z_l^{(1)},z_l^{(2)},z_l^{(3)}]=[y_l^{(0)},y_l^{(1)},y_l^{(0)},y_l^{(1)}] \tag{3.107}$$

Therefore a Markov model for this pulse repetition scheme can be defined as

$$\mathbf{P}_\sigma=\begin{array}{c} \\ s_0 \\ s_1 \\ s_2 \\ s_3 \end{array}\begin{array}{c} \begin{array}{cccc} s_0 & s_1 & s_2 & s_3 \end{array} \\ \begin{bmatrix} p_{y,00}p_{y,00} & p_{y,00}p_{y,01} & p_{y,01}p_{y,10} & p_{y,01}p_{y,11} \\ p_{y,10}p_{y,00} & p_{y,10}p_{y,01} & p_{y,11}p_{y,10} & p_{y,11}p_{y,11} \\ p_{y,00}p_{y,00} & p_{y,00}p_{y,01} & p_{y,01}p_{y,10} & p_{y,01}p_{y,11} \\ p_{y,10}p_{y,00} & p_{y,10}p_{y,01} & p_{y,11}p_{y,10} & p_{y,11}p_{y,11} \end{bmatrix} \end{array};\quad \begin{array}{ll} s_0=(00); & \gamma(s_0)=[0,0,0,0] \\ s_1=(01); & \gamma(s_1)=[0,1,0,1] \\ s_2=(10); & \gamma(s_2)=[1,0,1,0] \\ s_3=(11); & \gamma(s_3)=[1,1,1,1] \end{array} \tag{3.108}$$

with steady state probabilities given by

$$\underline{\boldsymbol{\pi}}_\sigma=\left[\pi_{y,0}p_{00},\pi_{y,0}p_{01},\pi_{y,1}p_{10},\pi_{y,1}p_{11}\right] \tag{3.109}$$

As previously mentioned two quaternary signals will be used per encoder output vector, thus $N_w=2$ and $T_s=2T_r$. Therefore, if the encoder is in state $\sigma_l=s_i$ at time l with corresponding output vector $z_l=\gamma(\sigma_l=s_i)=\underline{\zeta}_{s_i}=[\zeta_{s_i}^{(0)},\zeta_{s_i}^{(1)},\zeta_{s_i}^{(2)},\zeta_{s_i}^{(3)}]$, then

$$w_{\sigma_l,k}(t)=w_{s_i,k}(t)=\alpha_{i,k}w(t-\beta_{i,k}T_\beta),\qquad k=0,1 \tag{3.110}$$

with $\alpha_{i,0}=\alpha_{i,1}=(2\zeta_{s_i}^{(0)}-1)=(2\zeta_{s_i}^{(2)}-1)\in\{-1,1\}$ and $\beta_{i,0}=\beta_{i,1}=\zeta_{s_i}^{(1)}=\zeta_{s_i}^{(3)}\in\{0,1\}$.

Set the TH and DS sequences to be as in Section 3.5.1.1, that is {0, 2, 4, 1, 3} (prime code over GF(5)) and {-1, 1, -1, 1, -1, -1, 1, 1, -1, 1 } respectively, thus $\chi_a = 10$, $\chi_c = 5$, $\chi_{acw} = 10$ and $\Lambda = 5$. Finally assume there are no pulse attenuation and no jitter.

Under the previous assumptions the discrete part of the PSD can be found to be

$$S_D(f) = \frac{|W(f)|^2}{25T_s^2}\left|\sum_{k=0}^{1}\sum_{i=0}^{3}\sum_{n=0}^{4} e^{j2\pi fkT_r} e^{j2\pi fnT_s} a_{nN_w+k} e^{j2\pi fT_c c_{nN_w+k}} \alpha_{i,k} e^{j2\pi f\beta_{i,k}T_\beta} \pi_i\right|^2 \sum_{r=-\infty}^{\infty}\delta(f-\tfrac{r}{5T_s}) \tag{3.111}$$

while the PSD's continuous part is given by

$$\begin{aligned} S_C(f) = {} & \tfrac{2}{T_s}|W(f)|^2 + \tfrac{2|W(f)|^2}{5T_s}\sum_{n=0}^{4}\mathrm{Re}\{e^{-j2\pi fT_r}G_{AP}(0,0,1,-f,f)G_3(n,0,0,1,-f,f)\} \\ & + \tfrac{2|W(f)|^2}{5T_s}\mathrm{Re}\Big\{\sum_{k,k'=0}^{1}\sum_{n=0}^{4}\sum_{m=1}^{\infty}\sum_{i,i'=0}^{3} e^{-j2\pi fT_s m} e^{-j2\pi f(k'-k)T_r}\alpha_{i,k}\alpha_{i',k'}^{*} e^{-j2\pi f(\beta_{i',k'}-\beta_{i,k})T_\beta} \\ & \qquad \times \pi_i\{p_{ii'}^{(|m|)} - \pi_{i'}\}G_3(n,m,k,k',-f,f)\Big\} \\ & - \tfrac{|W(f)|^2}{5T_s}\sum_{n=0}^{4}\left|\sum_{k=0}^{1}\sum_{i=0}^{3} e^{j2\pi fkT_r} a_{nN_w+k} e^{j2\pi fT_c c_{nN_w+k}} \alpha_{i,k} e^{j2\pi f\beta_{i,k}T_\beta}\pi_i\right|^2 \end{aligned} \tag{3.112}$$

where $G_{AP}(m,k,k',-f,f)$ was defined in Equation (3.89) as

$$G_{AP}(m,k,k',-f,f) = \begin{cases} \sum_{i=0}^{N_\sigma-1}\alpha_{i,k}\alpha_{i,k'}^{*} e^{-j2\pi f(\beta_{i,k'}-\beta_{i,k})T_\beta}\pi_i & m=0 \\ \sum_{i,i'=0}^{N_\sigma-1}\alpha_{i,k}\alpha_{i',k'}^{*} e^{-j2\pi f(\beta_{i',k'}-\beta_{i,k})T_\beta}\pi_i p_{ii'}^{(|m|)} & m\neq 0 \end{cases} \tag{3.113}$$

Power spectral density (PSD) plots obtained by using Equations (3.111) and (3.112) with $p_{y,10} = 3/5$ and different values of $p_{y,01}$ are shown in Figure 3-9.

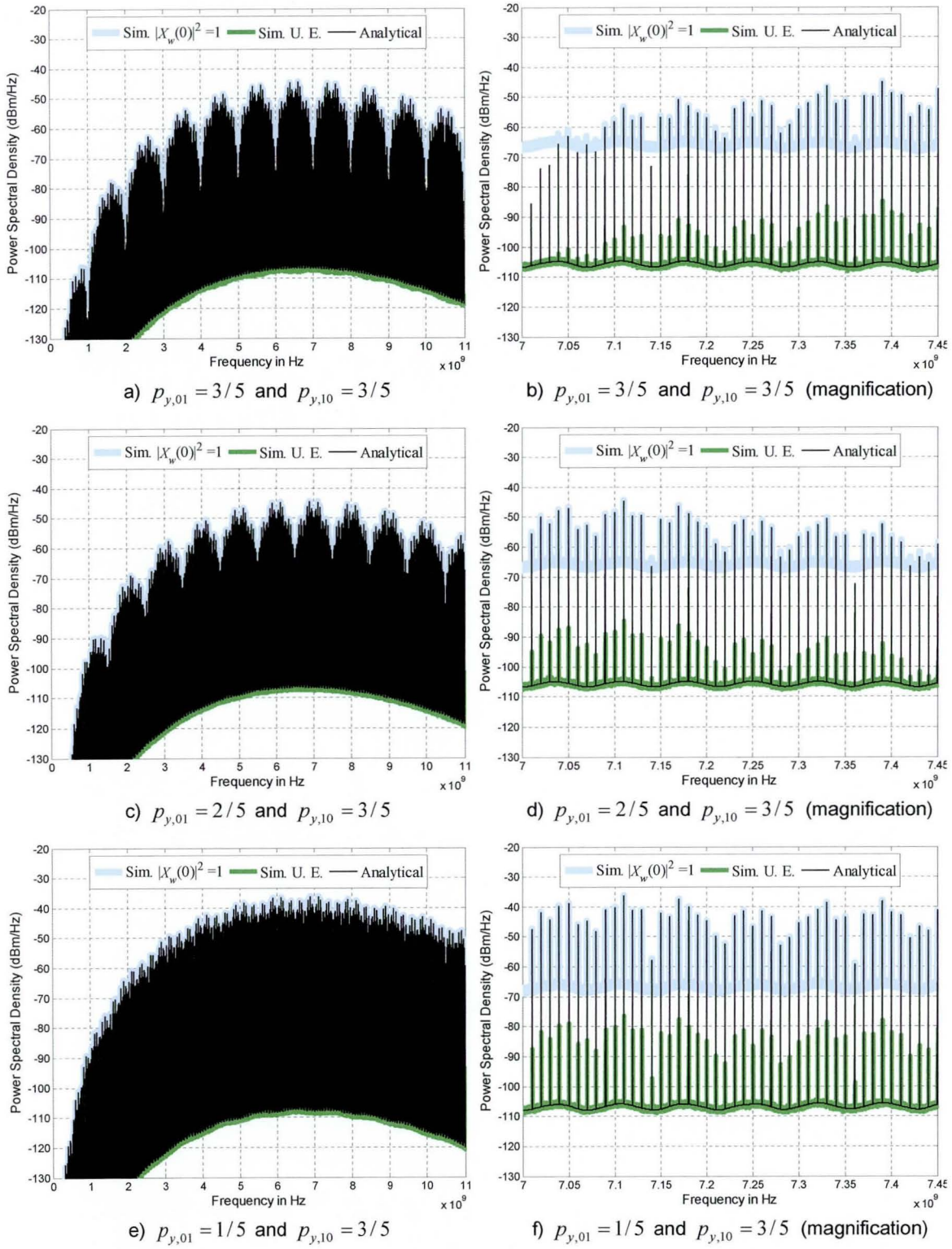
Note how for this system spectral lines spaced at $1/(\Lambda T_s) = 10$ MHz intervals appear in the PSD even for the case with $p_{y,01} = 3/5$ and $p_{y,10} = 3/5$ as shown in Figures 3-9a and 3-9b. This is due to for this case the SE pair MC steady state probabilities are

$$\underline{\pi_\sigma} = \left[\pi_{y,0}p_{00}, \pi_{y,0}p_{01}, \pi_{y,1}p_{10}, \pi_{y,1}p_{11}\right] = \left[\tfrac{1}{5}, \tfrac{3}{10}, \tfrac{3}{10}, \tfrac{1}{5}\right] \tag{3.114}$$

which is not uniform like distributed.

By comparing the plots in Figure 3-9 with the plots in Figure 3-6 the effects of the encoding operation over the PSD shape can be seen. These effects are particularly acute for the $p_{y,10} = 3/5$, $p_{y,01} = 3/5$ case, as evidenced by the absence of spectral lines in Figure 3-6a and the presence of spectral lines in Figure 3-9a. Therefore, it can be said that, for the particular case of the encoder defined by Equations (3.6) and (3.16), the introduction of the encoding operation had positive "spectral shaping" effects compared to the non-coded pulse repetition scheme.

Finally note the good agreement between the analytical and simulated results for both the PSD's continuous part (Sim. U.E.) and the PSD's discrete part (Sim. $|X_w(0)|^2 = 1$) in Figure 3-9.



a) $p_{y,01} = 3/5$ and $p_{y,10} = 3/5$

b) $p_{y,01} = 3/5$ and $p_{y,10} = 3/5$ (magnification)

c) $p_{y,01} = 2/5$ and $p_{y,10} = 3/5$

d) $p_{y,01} = 2/5$ and $p_{y,10} = 3/5$ (magnification)

e) $p_{y,01} = 1/5$ and $p_{y,10} = 3/5$

f) $p_{y,01} = 1/5$ and $p_{y,10} = 3/5$ (magnification)

Figure 3-9. Analytical and simulated PSDs for a BMS coupled to a Q-BOPPM IR-based UWB system with pulse repetition and no convolutional coding. The signal parameters are $T_s = 20$ ns, $T_r = 10$ ns, $T_c = 2$ ns, $T_\beta = 1$ ns and no jitter. The 4th derivative Gaussian pulse is used with duration $T_w \approx 0.4$ ns. The sequences {-1, 1, -1, 1, -1, -1, 1, 1, -1, 1} and {0, 2, 4, 1, 3} were used for DS and TH respectively.

3.5.3 Rate 1/4 Maximum Free Distance Feedforward Convolutional Encoder $(5, 5, 7, 7)_8$ – Effects of the Generators Order

Consider now the rate 1/4, total encoder memory $v=2$, convolutional encoder defined by the following polynomial transfer function matrix

$$\mathbf{G}(D)=[1+D^2 \quad 1+D^2 \quad 1+D+D^2 \quad 1+D+D^2] \Rightarrow (5,5,7,7)_8 \tag{3.115}$$

This encoder has maximum free distance and optimum distance spectrum, [83, 111].

The SE pair Markov model for this encoder is defined by

$$\mathbf{P}_\sigma = \begin{array}{c} \\ s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{array} \begin{array}{c} \begin{array}{cccccccc} s_0 & s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 \end{array} \\ \begin{bmatrix} p_{y,00} & 0 & 0 & 0 & p_{y,01} & 0 & 0 & 0 \\ p_{y,00} & 0 & 0 & 0 & p_{y,01} & 0 & 0 & 0 \\ 0 & p_{y,00} & 0 & 0 & 0 & p_{y,01} & 0 & 0 \\ 0 & p_{y,00} & 0 & 0 & 0 & p_{y,01} & 0 & 0 \\ 0 & 0 & p_{y,10} & 0 & 0 & 0 & p_{y,11} & 0 \\ 0 & 0 & p_{y,10} & 0 & 0 & 0 & p_{y,11} & 0 \\ 0 & 0 & 0 & p_{y,10} & 0 & 0 & 0 & p_{y,11} \\ 0 & 0 & 0 & p_{y,10} & 0 & 0 & 0 & p_{y,11} \end{bmatrix} \end{array} \begin{array}{l} \\ s_0=(0,00);\ \gamma(s_0)=[0,0,0,0] \\ s_1=(0,01);\ \gamma(s_1)=[1,\ 1,\ 1,\ 1] \\ s_2=(0,10);\ \gamma(s_2)=[0\ 0,\ 1,\ 1] \\ s_3=(0,11);\ \gamma(s_3)=[1,\ 1,0,0] \\ s_4=(1,00);\ \gamma(s_4)=[1,\ 1,\ 1,\ 1] \\ s_5=(1,01);\ \gamma(s_5)=[0,0,0,0] \\ s_6=(1,10);\ \gamma(s_6)=[1,\ 1,0,0] \\ s_7=(1,11);\ \gamma(s_7)=[0,0,\ 1,\ 1] \end{array} \tag{3.116}$$

with steady state probabilities given by

$$\underline{\boldsymbol{\pi}_\sigma}=\left[\pi_{y,0}p_{00}p_{00},\pi_{y,1}p_{10}p_{00},\pi_{y,0}p_{01}p_{10},\pi_{y,1}p_{11}p_{10},\pi_{y,0}p_{00}p_{01},\pi_{y,1}p_{10}p_{01},\pi_{y,0}p_{01}p_{11},\pi_{y,1}p_{11}p_{11}\right] \tag{3.117}$$

In order to see the effects of the generators order in the PSD shape, a Q-BOPPM IR-based UWB system will be considered with perfectly random TH uniformly distributed over the set {0, 1, 2, 3, 4, 5}. Timing jitter, attenuation and DS multiplication will not be considered for this example. For this case two quaternary biorthogonal signals are needed to transmit one encoder output vector due to each vector consists of four bits per state, thus $N_w=2$ and $T_s=2T_r$. Therefore, if the encoder is in state $\boldsymbol{\sigma}_l=s_i$ at time l with corresponding encoder output vector $\boldsymbol{z}_l=\gamma(\boldsymbol{\sigma}_l=s_i)=\underline{\zeta_{s_i}}=[\zeta_{s_i}^{(0)},\zeta_{s_i}^{(1)},\zeta_{s_i}^{(2)},\zeta_{s_i}^{(3)}]$, then

$$w_{\sigma_l,k}(t)=w_{s_i,k}(t)=\alpha_{i,k}w(t-\beta_{i,k}T_\beta), \qquad k=0,1 \tag{3.118}$$

with $\alpha_{i,0}=(2\zeta_{s_i}^{(0)}-1)$, $\alpha_{i,1}=(2\zeta_{s_i}^{(2)}-1)\ \in \{-1, 1\}$ and $\beta_{i,0}=\zeta_{s_i}^{(1)}$, $\beta_{i,1}=\zeta_{s_i}^{(3)}\ \in \{0, 1\}$.

Under these assumptions the discrete part of the PSD can be found to be

$$S_D(f)=\frac{|G_{eT_c}(f)|^2|W(f)|^2}{T_s^2}\left|\sum_{k=0}^{1}\sum_{i=0}^{7}e^{j2\pi fkT_r}\alpha_{i,k}e^{j2\pi f\beta_{i,k}T_\beta}\pi_i\right|^2\sum_{r=-\infty}^{\infty}\delta(f-\tfrac{r}{T_s}) \tag{3.119}$$

while the PSD's continuous part is given by

$$\begin{aligned} S_C(f)=&\tfrac{2}{T_s}|W(f)|^2-\frac{|G_{eT_c}(f)|^2|W(f)|^2}{T_s}\left|\sum_{k=0}^{1}\sum_{i=0}^{7}e^{j2\pi fkT_r}\alpha_{i,k}e^{j2\pi f\beta_{i,k}T_\beta}\pi_i\right|^2 \\ &+\frac{2|G_{eT_c}(f)|^2|W(f)|^2}{T_s}\Bigg(\mathrm{Re}\{e^{-j2\pi fT_r}\sum_{i=0}^{7}\alpha_{i,0}\alpha_{i',1}^{*}e^{-j2\pi f(\beta_{i',1}-\beta_{i,0})T_\beta}\pi_i\} \\ &+\mathrm{Re}\Big\{\sum_{k,k'=0}^{1}\sum_{m=1}^{\infty}\sum_{i,i'=0}^{7}e^{-j2\pi f(k'-k)T_r}e^{-j2\pi fT_sm}\alpha_{i,k}\alpha_{i',k'}^{*}e^{-j2\pi f(\beta_{i',k'}-\beta_{i,k})T_\beta}\pi_i\{p_{ii'}^{(|m|)}-\pi_{i'}\}\Big\}\Bigg) \end{aligned} \tag{3.120}$$

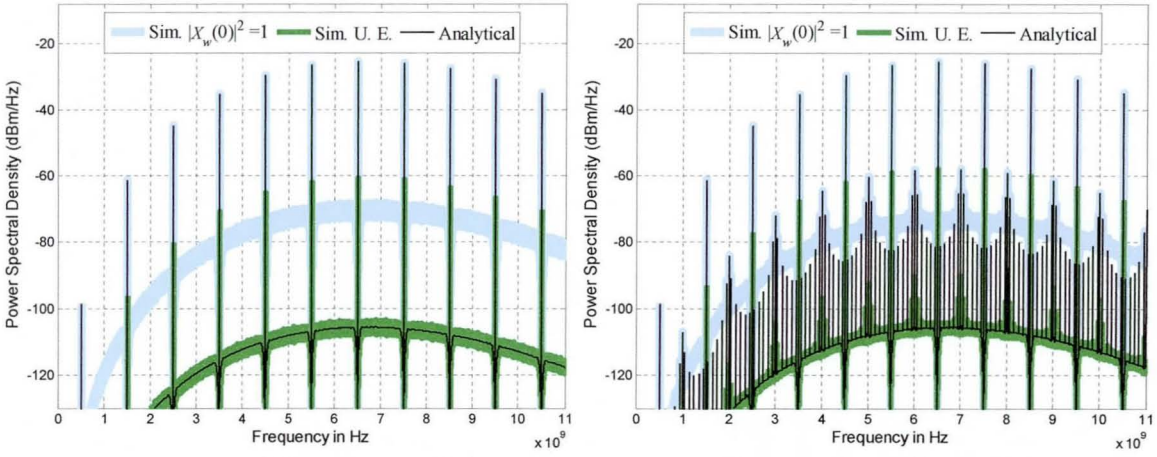
where $|G_{cT_c}(f)|^2$ is defined as

$$|G_{cT_c}(f)|^2 = \tfrac{1}{5} + \tfrac{2}{25}\sum_{l=1}^{4}(5-l)\cos(2\pi f l T_c) \tag{3.121}$$

Power spectral density (PSD) plots for different values of $p_{y,01}$ and $p_{y,10}$ are shown in Figure 3-10. It is important to highlight the appearance of spectral lines in Figure 3-10a even when $p_{y,01}=1/2$ and $p_{y,10}=1/2$ with corresponding SE pair MC steady state probabilities $\underline{\pi}_\sigma = \left[\tfrac{1}{8},\tfrac{1}{8},\tfrac{1}{8},\tfrac{1}{8},\tfrac{1}{8},\tfrac{1}{8},\tfrac{1}{8},\tfrac{1}{8}\right]$. To further see this note that for $\underline{\pi}_\sigma = \left[\tfrac{1}{8},\tfrac{1}{8},\tfrac{1}{8},\tfrac{1}{8},\tfrac{1}{8},\tfrac{1}{8},\tfrac{1}{8},\tfrac{1}{8}\right]$ the PSD's discrete part becomes

$$S_D(f) = \frac{|G_{cT_c}(f)|^2|W(f)|^2}{4T_s^2}\left|-1+e^{j2\pi fT_\beta}+e^{j2\pi fT_r}\{-1+e^{j2\pi fT_\beta}\}\right|^2\sum_{r=-\infty}^{\infty}\delta(f-\tfrac{r}{T_s}) \tag{3.122}$$

which is not equal to zero for all values of $f = r/T_s$, $r = 0, \pm1, \pm2, \ldots$.



a) $p_{y,01}=1/2$ and $p_{y,10}=1/2$ b) $p_{y,01}=2/5$ and $p_{y,10}=3/5$

Figure 3-10. Analytical and simulated PSDs obtained when using the maximum free distance feedforward binary convolutional encoder $(5, 5, 7, 7)_8$ in a Q-BOPPM IR-based UWB system. The signal parameters are $T_s = 20$ ns, $T_r = 10$ ns, $T_c = 2$ ns, $T_\beta = 1$ ns and c_{lNw+k} uniform distributed over the set $\{0, 1, 2, 3, 4\}$. The 4th derivative Gaussian pulse is used with duration $T_w \approx 0.4$ ns.

Remember from Section 2.7.3 that the distance and bit error correction (BER) properties of a convolutional encoder do not change when interchanging columns in its transfer function matrix. Nevertheless consider what happens with the signal's PSD when interchanging the second and last columns in Equation (3.115) to

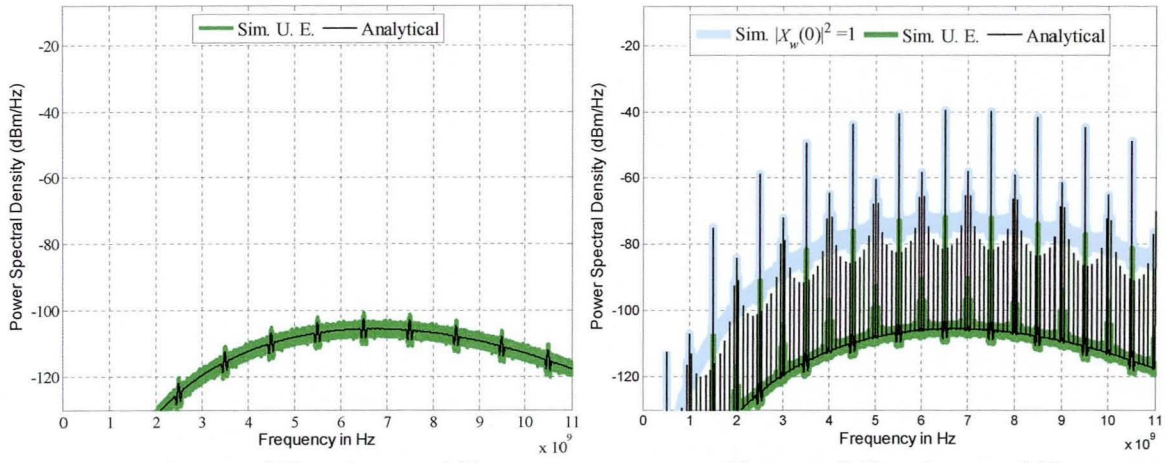
$$\mathbf{G}(D) = [1+D^2 \quad 1+D+D^2 \quad 1+D+D^2 \quad 1+D^2] \Rightarrow (5,7,7,5)_8 \tag{3.123}$$

The corresponding SE pair Markov model has the same transition probabilities matrix, $\mathbf{P}_\sigma$, given in Equation (3.116) but the output mapping, $\gamma(\cdot)$, changes to

$$\begin{aligned} &\gamma(s_0)=[0,0,0,0] \quad \gamma(s_1)=[1,1,1,1] \quad \gamma(s_2)=[0,1,1,0] \quad \gamma(s_3)=[1,0,0,1] \\ &\gamma(s_4)=[1,1,1,1] \quad \gamma(s_5)=[0,0,0,0] \quad \gamma(s_6)=[1,0,0,1] \quad \gamma(s_7)=[0,1,1,0] \end{aligned} \tag{3.124}$$

This simple change has a significant impact in the signal's PSD as shown in Figure 3-11. Note how for $p_{y,01}=1/2$ and $p_{y,10}=1/2$ the PSD's discrete part becomes zero as shown in

Figure 3-11a. Furthermore, by comparing Figure 3-10b with Figure 3-11b it can be seen that the height of the spectral lines has been reduced in Figure 3-11b. Therefore, it can be concluded that even though interchanging the generators has no effects on the error correction capabilities of the encoder, such change can have advantageous effects on the signal's PSD.



a) $p_{y,01}=1/2$ and $p_{y,10}=1/2$ b) $p_{y,01}=2/5$ and $p_{y,10}=3/5$

Figure 3-11. Analytical and simulated PSDs obtained when using the maximum free distance feedforward binary convolutional encoder $(5, 7, 7, 5)_8$ in a Q-BOPPM IR-based UWB system. The signal parameters are $T_s = 20$ ns, $T_r = 10$ ns, $T_c = 2$ ns, $T_\beta = 1$ ns and c_{lNw+k} uniform distributed over the set $\{0, 1, 2, 3, 4\}$. The 4th derivative Gaussian pulse is used with duration $T_w \approx 0.4$ ns.

Again note how there is good agreement between the analytical and simulated results for both the PSD's continuous part (Sim. U.E.) and the PSD's discrete part (Sim. $|X_w(0)|^2=1$) in Figures 3-10 and 3-11.

3.6 Conclusions for Chapter 3

In this chapter the spectral analysis of convolutionally coded/Markov-driven IR-based UWB signals with binary Markov sources has been presented. Particularly, a first order binary Markov source (BMS) was assumed as it enables the modelling of unbalanced (that is non-uniform distributed) independent identically distributed (i.i.d.) memoryless binary data streams as well as binary data streams with memory often found in practical applications. The analysis was performed by introducing a source-encoder pair Markov model (of the Moore type) which describes the statistics of the convolutional encoder when driven by the BMS. In order to obtain the SE pair Markov model, it was shown that the encoder's state diagram representation is essentially a Melay FSSM which is strongly connected and has a self loop in state zero.

Based on the SE pair Markov model a general convolutionally coded/Markov-driven IR-based UWB signal model was then introduced. This signal model covers a wide variety of IR-based modulation schemes such as PPM, PAM, BPSK, BOPPM (BPSK/PPM), PAM/PPM and PSM, combined with periodic or random TH and/or DS multiplication. Furthermore, the model accounts for factors such as attenuation and timing jitter. Then the spectral analysis of the signal model was

performed by using the properties of the SE pair Markov model and a non-cyclostationary based approach.

From the spectral analysis, novel closed form PSD formulas for convolutionally coded/Markov-driven IR-based UWB signals were obtained. Such formulas allow the spectral analysis of complicated systems in a straightforward way by providing explicit expressions for the continuous and discrete PSD components and by clearly differentiating the contributions of each variable (e.g. timing jitter, attenuation, TH, etc.) to the PSD. Furthermore, as shown in Section 3.5, such expressions can be easily applied for the analysis of non-coded systems and any other scheme that can be described by an ergodic regular Moore Markov model.

Application examples of the formulas were presented for several cases of interest. With these examples it was shown that the introduction of the convolutional encoding operation can be used for spectral shaping purposes. It was demonstrated that for some cases the introduction of convolutional coding has positive effects in the PSD shape of IR-based UWB signals when compared with the non-convolutionally coded case, as spectral lines were attenuated or even eliminated by the coded system. Nevertheless, it was shown as well that the introduction of a specific convolutional encoder for spectral shaping purposes can not be trivially done, as even a simple change in the generators order may have a significant impact in the PSD shape. Finally note how the analytical and simulation results presented in all application examples of this chapter show a good agreement for both the PSD's continuous part (Sim. U.E.) and the PSD discrete part (Sim. $|X_w(0)|^2=1$) which validates the closed form PSD expressions derived in Section 3.4.

The idea of using convolutional encoders for spectral shaping in IR-based UWB systems will be addressed in the following chapters where convolutional encoders specifically designed to achieve both improved PSD characteristics and bit error rate performance are introduced.

Chapter 4

Maximum Free Distance Binary to *M*-ary Convolutional Encoders for Pseudo-Chaotic Time Hopping Type Spectral Shaping of IR-Based UWB Signals

4.1 Introduction

In this chapter novel maximum free distance (MFD) binary to *M*-ary convolutional encoders with convenient power spectral density (PSD) characteristics for non-interleaved pulse position modulated (PPM) time hopping (TH) impulse radio (IR) based ultra wideband (UWB) systems are introduced. The PSD characteristics of the new encoders are similar to those obtained with the pseudo-chaotic time hopping (PCTH) scheme introduced in [70] (Section 2.5.4) for *M*-ary PPM TH-IR-based UWB systems.

The new encoders provide improved bit error rate (BER) performance over the non-coded and PCTH schemes when used in UWB systems employing *M*-ary orthogonal signalling (not necessarily restricted to be PPM based). Feedforward encoders with rates 1, 1/2 and 1/3, for 16-ary, 32-ary, 64-ary and 128-ary orthogonal modulation schemes are introduced. For these encoders the free distance is defined in terms of the *M*-ary Hamming distance between codewords and the rate is given as a function of the number of *M*-ary symbols produced per each input bit (Sections 2.7.3 and 2.7.6).

The design of convolutional codes for *M*-ary (where *M* is a power of 2) orthogonal signalling schemes has been addressed in [105-107, 126, 127]. A construction method for rate 1 binary to *M*-ary convolutional encoders with constraint length $K = \log_2(M)$ was introduced in [126, 127]. Maximum free distance (MFD) binary to *M*-ary convolutional encoders are reported in [105, 107] for 4-ary, 8-ary and 16-ary orthogonal signalling. These encoders were found by computer search with the constraint length, *K*, not limited to be equal to $\log_2(M)$, [105, 107]. An analytical method

for the design of rate 1 binary to M-ary convolutional codes is presented in [106] where maximum M-ary free distance rate 1 binary to 16-ary encoders are introduced. The use of orthogonal convolutional codes in code division multiple access (CDMA) systems is discussed in [82, 91, 128, 129] while its use in optical channels with PPM has been addressed in [130-133]. All of these works are focused on BER analysis and do not consider the PSD of the transmitted signal.

The use of binary convolutional codes in TH-IR-based UWB systems with M-ary PPM signalling has been proposed in [22, 33, 37, 38]. Nevertheless none of these papers consider the use of binary to M-ary convolutional codes. Instead, the use of low rate binary superorthogonal codes, [82], and high rate binary punctured convolutional codes, [84], without interleaving is proposed in [22]. Binary rate compatible punctured convolutional codes with interleaving are employed in [38]. The use of rate $1/\log_2(M)$, $K=\log_2(M)$, MFD binary convolutional encoders with interleaving is proposed in [33] for M-ary PPM. Besides, rate $1/\log_2(M)$, $K=\log_2(M)+1$, MFD binary convolutional encoders without interleaving are used in [37] for M-ary PPM. It is important to highlight that none of the convolutional encoders used in these works were designed for M-ary orthogonal alphabets or interpreted as binary to M-ary convolutional encoders. This interpretation is important due to, as it will be shown in this chapter, the BER performance can be improved upon the introduction of properly designed binary to M-ary convolutional encoders.

A form of orthogonal convolutional modulation for IR-based UWB systems has been proposed in [31]. This scheme does not use traditional orthogonal M-ary PPM signalling. Instead, an orthogonal signal set is constructed by using M non-overlapping TH sequences which define a specific train of pulses for each M-ary symbol. Therefore, this scheme is similar to the implementation of orthogonal convolutional coding through the use of binary Hadamard codes in direct sequence (DS) CDMA systems, [82, 91, 128, 129].

As mentioned in Section 2.6, both [33] and [37] address the effects of the convolutional coding operation on the signal's PSD. Remember that, in a similar fashion to PCTH, in these schemes all time hopping is driven by the interleaver/encoder output with no additional TH or DS multiplication. The spectral analysis presented in [33] was performed using simulation, while [37] uses a simplified model for the signal's spectral analysis which is based on the assumption of independent identically distributed (i.i.d.) streams with uniform distribution at the encoder output (that is the correlation introduced by the convolutional encoders is not considered). The analysis presented in both [33] and [37] showed that spectral characteristics similar to those obtained with PCTH are feasible when using these schemes under similar operation constraints, that is uniform distributed i.i.d. binary data streams are assumed at the encoder input. Nevertheless, the scheme proposed in [33] relies on the interleaver to obtain such PSD characteristics and therefore a delay in the decoding operation is introduced when compared to non-interleaved schemes.

Based on the previous discussion it can be said that the use of binary to M-ary convolutional encoders in M-ary PPM IR-based UWB systems to achieve both PSD characteristics similar to those obtained with the PCTH scheme and improved BER has not been previously addressed.

Therefore the focus here is to suggest, identify and find good feedforward binary to M-ary convolutional encoders with convenient PSD characteristics without interleaving.

This chapter is structured as follows. Section 4.2 describes the proposed system model. Section 4.3 introduces the signal model and its respective PSD. The code search procedure is introduced in Section 4.4 altogether with the MFD binary to M-ary convolutional encoders found with such procedure. Comparisons in terms of PSD characteristics between the new encoders and the PCTH scheme are presented in Section 4.5. The BER performance analysis for the new encoders and BER comparisons with the PCTH scheme are provided in Section 4.6. Finally conclusions are presented in Section 4.7.

4.2 System Model for Binary to M-ary Convolutionally Coded TH-IR-Based UWB Signals

The block diagram of the system model used for discussion in this chapter is shown in Figure 4-1. This model is similar to the PCTH scheme reviewed in Section 2.5.4, with the difference that the PCTH encoder in Figure 2-10 (consisting of the shift register and the optional modified Gray based mapper) is replaced with a binary to M-ary convolutional encoder.

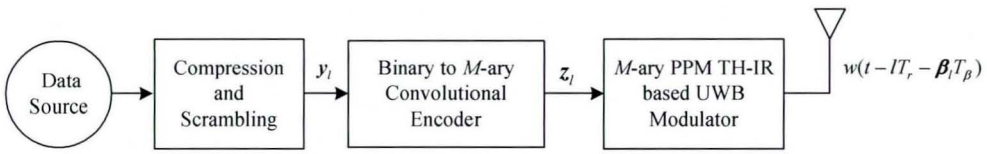


Figure 4-1. Binary to M-ary convolutionally coded M-ary PPM TH-IR-based UWB system.

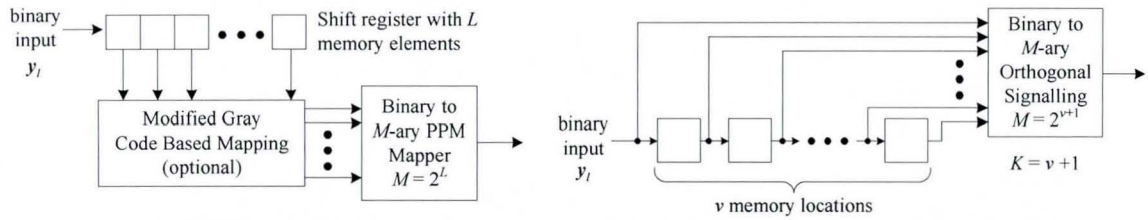
Throughout this chapter it will be assumed that M is a power of 2 and that $\{y_l\}$ is a uniformly distributed i.i.d. binary stream as in the PCTH scheme, [70]. That is, y_l has probability mass function (p.m.f)

$$p_{y,0} = \Pr\{y_l = 0\} = 1/2; \quad \text{and} \quad p_{y,1} = \Pr\{y_l = 1\} = 1 - p_{y,0} = 1/2 \tag{4.1}$$

4.2.1 Interpretation of PCTH as a Rate 1 Binary to M-ary Convolutionally coded Scheme

The encoders used in [70] and [37] were interpreted as binary encoders. However, these encoders can be interpreted as binary to M-ary encoders as well. This interpretation is important due to PCTH can be seen as a set of rate 1 binary to M-ary convolutional encoders with constraint length $K = L = \log_2(M)$ and total encoder memory $v = \log_2(M) - 1$. One can see that is interpretation is correct by comparing the basic PCTH encoder structure in Figure 4-2a (see also Section 2.5.4 – Figure 2-10) with the binary representation for rate 1 binary to M-ary convolutional encoders introduced in Section 2.7.6 – Figure 2-21. We can further confirm this interpretation by comparing the basic structure of the PCTH encoder with that of the binary to M-ary orthogonal convolutional encoders introduced by Viterbi in [126] as shown in Figure 4-2b. The main

difference between both encoders is that in Figure 4-2b the modulation format has not been defined while in the original PCTH scheme (Figure 4-2a) the modulation format used is M-ary PPM

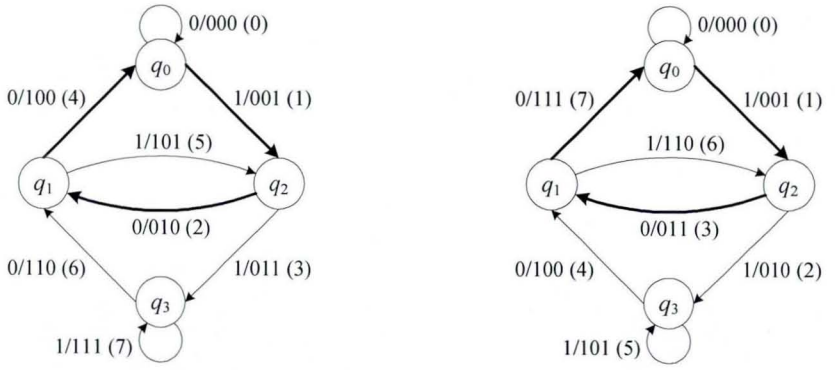


a) PCTH encoder b) Viterbi's orthogonal convolutional encoder

Figure 4-2. Basic diagrams of the PCTH encoder and Viterbi's orthogonal convolutional encoder.

As mentioned in Section 2.5.4, the modified Gray code based mapper in Figures 2-10 and 4-2a is needed for tent map PCTH implementations. However, for Bernoulli shift map PCTH realisations the Gray code based mapper is discarded. The tent map implementation offers improved BER performance for binary hard decision Viterbi decoding (HVD) compared to the Bernoulli shift map realization, [70]. This is because of the larger binary free distance, d_{free}, of the tent map encoder, which is the main performance parameter when using binary HVD (Sections 2.7.4 and 2.7.5) as in [70]. However, both PCTH realizations have the same M-ary free distance, d_{Mfree}, when interpreted as binary to M-ary convolutional encoders. This is due to changing a non-zero label with another non-zero Gray code based label does not change the M-ary Hamming weight of a given M-ary symbol/signal (Section 2.7.6) as far as all changes are performed in a one to one basis (such as with Gray code based labelling).

As an example consider Figure 4-3 where state transition diagrams equivalent to the Bernoulli shift and tent maps encoders for 8-ary PCTH are shown. Starting in q_0 the path that defines the free distance is highlighted with bold lines. It can be seen that the binary free distance is larger for the tent map encoder. However, the M-ary free distance is the same for both encoders as the sequences {1, 2, 4} (Bernoulli shift map) and {1, 3, 7} (tent map) have the same M-ary Hamming weight.



a) Bernoulli shift map PCTH encoder, $d_{free} = 3$ b) Tent map PCTH encoder, $d_{free} = 6$

Figure 4-3. State transition diagrams equivalent to the Bernoulli shift and tent map encoders for 8-ary PCTH. Both encoders have different binary free distance, d_{free}, but the same M-ary free distance, $d_{Mfree} = 3$.

From Figure 4-3 it is clear that the M-ary Hamming weight of every branch remains unchanged when using the modified Gray code based labelling. Therefore, performance related parameters

(see Section 2.7.3) such as the distance spectrum and information weight spectrum (IWS), are the same for both encoders when calculated in terms of the M-ary Hamming weight. As a consequence the performance of both encoders should be similar when interpreted as binary to M-ary convolutional encoders. This can be inferred from the union bounds introduced in Section 2.7.4 for bit and first event error probabilities since they would be identical for both binary to M-ary encoders.

4.2.2 Source-Encoder Markov Model for the Binary to *M*-ary Convolutionally Coded Time Hopping Scheme

The source-encoder (SE) model introduced in Section 3.2 will be used for the signal's spectral analysis. Therefore the binary Markov source (BMS) equivalent to the data source and compression-scrambling blocks in Figure 4-1 is defined by the following one step transition probabilities matrix

$$\mathbf{P}_y = \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix} \tag{4.2}$$

where all the transitions have the same probability due to the uniform i.i.d. assumption for the binary data stream.

This chapter focuses on feedforward binary to M-ary convolutional encoders. The total encoder memory is not restricted to $v = L-1 = \log_2(M)-1$ or $v = L = \log_2(M)$ as in [70] and [37] respectively and the code rate, $1/\kappa$, can be smaller than 1. Therefore, the encoders considered in this chapter have the basic structure shown in Figure 4-4 (see as also Section 2.7.6 – Figure 2-21). For clarity the operations defined by the encoder's generator polynomials are generically represented by the binary operations set blocks in Figure 4-4.

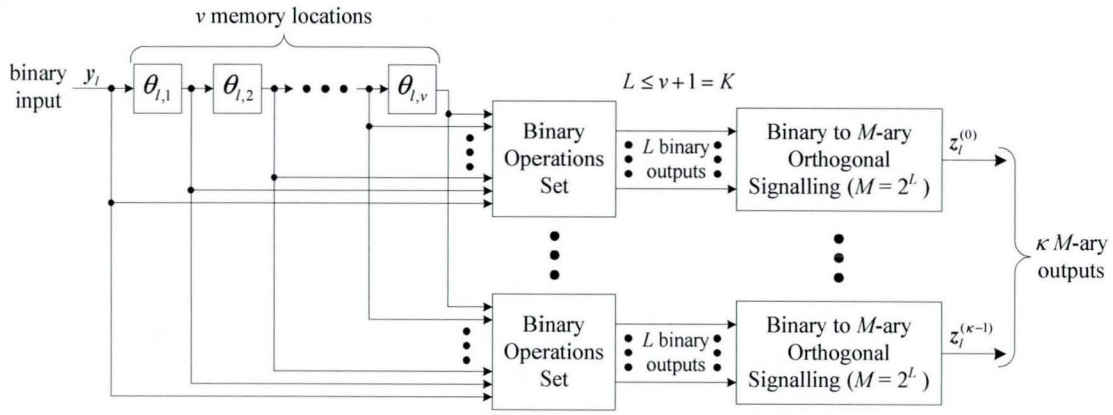


Figure 4-4. Generic diagram of a rate $1/\kappa$ feedforward binary to M-ary convolutional encoder.

From Section 3.2.3 the state process, σ_l, defined by the SE pair Markov model has transition probabilities matrix given by

$$\mathbf{P}_\sigma = \begin{bmatrix} \frac{1}{2}\mathbf{B}_0 & \frac{1}{2}\mathbf{B}_0 \\ \frac{1}{2}\mathbf{B}_1 & \frac{1}{2}\mathbf{B}_1 \end{bmatrix} \tag{4.3}$$

where $\mathbf{B}_0$ and $\mathbf{B}_1$ are the $N_q \times N_q$ the next state matrices defined in Section 3.2.3. Remember $N_q = 2^v$ is the number of encoder states, q_i, as defined by the encoder's state transition diagram representation (see Sections 2.7.2 and 3.2.2). Now assume the encoder memory content at time l to be $[\theta_{l,1}, \theta_{l,2}, ..., \theta_{l,v}] = [\mu_1, \mu_2, ..., \mu_v]$ with corresponding state label q_i, where by convention $i = \mu_1 2^{v-1} + \mu_2 2^{v-2} + ... + \mu_v$ as in the previous chapter. When the current input is $y_l = 0$ the next state is $[\theta_{l+1,1}, \theta_{l+1,2}, ..., \theta_{l+1,v}] = [0, \mu_1, \mu_2, ..., \mu_{v-1}]$ or using Section 3.2.3 notation:

$$\theta_{l+1} = \phi(y_l, \theta_l) = \phi(0, q_i) = q_{\lfloor i/2 \rfloor} \tag{4.4}$$

where $\lfloor \cdot \rfloor$ stands for the floor function and $\phi(\cdot)$ is the next state function of the encoder's Melay finite state sequential machine model (FSSM) equivalent to the encoder's state transition diagram. Therefore each element, $B_{0,ij}$, of $\mathbf{B}_0$ is equal to

$$B_{0,ij} = \begin{cases} 0, & j \neq \lfloor i/2 \rfloor \\ 1, & j = \lfloor i/2 \rfloor \end{cases} \tag{4.5}$$

Similarly when the current input is $y_l = 1$ the next state is given by

$$\theta_{l+1} = \phi(y_l, \theta_l) = \phi(1, q_i) = q_{2^{v-1} + \lfloor i/2 \rfloor} \tag{4.6}$$

since $[\theta_{l+1,1}, \theta_{l+1,2}, ..., \theta_{l+1,v}] = [1, \mu_1, \mu_2, ..., \mu_{v-1}]$, and each element, $B_{1,ij}$, of $\mathbf{B}_1$ is equal to

$$B_{1,ij} = \begin{cases} 0, & j \neq 2^{v-1} + \lfloor i/2 \rfloor \\ 1, & j = 2^{v-1} + \lfloor i/2 \rfloor \end{cases} \tag{4.7}$$

Finally the state set for the SE pair Markov model is defined as

$$\begin{aligned} \mathcal{S} &= \{(r_0, q_0), (r_0, q_1), ..., (r_0, q_{Nq-1}), (r_1, q_0), (r_1, q_1), ..., (r_1, q_{Nq-1})\} \\ &= \{(0, q_0), (0, q_1), ..., (0, q_{Nq-1}), (1, q_0), (1, q_1), ..., (1, q_{Nq-1})\} = \{s_0, s_1, ..., s_{2Nq-1}\} \end{aligned} \tag{4.8}$$

with corresponding output mapping

$$\underline{\zeta_{s_{nN_q+j}}} = [\zeta^{(0)}_{s_{nN_q+j}}, \zeta^{(1)}_{s_{nN_q+j}}, ..., \zeta^{(\kappa-1)}_{s_{nN_q+j}}] = \gamma(s_{nN_q+j}) = \gamma(n, q_j) \tag{4.9}$$

where $\gamma(\cdot)$ is the output mapping function, $n = 0,1$ and $j = 0,1,..., N_q - 1$.

From Equations (4.2) to (4.7) it can be seen that for the particular case of uniform distributed i.i.d. binary streams the entries in each column of the transition probabilities matrix, $\mathbf{P}_\sigma$, sum to unity. Thus $\mathbf{P}_\sigma$ is a double stochastic matrix and therefore all the steady state probabilities are equal, [134], that is

$$\underline{\pi_\sigma} = [\pi_{\sigma,0}, \pi_{\sigma,1}, ..., \pi_{\sigma,2N_q-1}] = [\tfrac{1}{2N_q}, \tfrac{1}{2N_q}, ..., \tfrac{1}{2N_q}] \tag{4.10}$$

As an example the next state matrices for feedforward encoders with $v = 2$ (e.g. the Bernoulli shift and tent map encoders shown in Figure 4-3) are given by

$$\mathbf{B}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \text{ and } \mathbf{B}_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.11}$$

Hence the SE pair Markov model for the 8-ary Bernoulli shift map encoder is defined by

$$\mathbf{P}_\sigma = \begin{array}{c} \\ s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{array} \begin{array}{c} \begin{array}{cccccccc} s_0 & s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 \end{array} \\ \left[\begin{array}{cccccccc} 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 \end{array}\right] \end{array} ; \quad \begin{array}{l} s_0 = (0, q_0);\ \gamma(s_0) = 0 \\ s_1 = (0, q_1);\ \gamma(s_1) = 4 \\ s_2 = (0, q_2);\ \gamma(s_2) = 2 \\ s_3 = (0, q_3);\ \gamma(s_3) = 6 \\ s_4 = (1, q_0);\ \gamma(s_4) = 1 \\ s_5 = (1, q_1);\ \gamma(s_5) = 5 \\ s_6 = (1, q_2);\ \gamma(s_5) = 3 \\ s_7 = (1, q_3);\ \gamma(s_7) = 7 \end{array} \tag{4.12}$$

with steady state probabilities vector equal to

$$\underline{\boldsymbol{\pi}_\sigma} = [\pi_{\sigma,0}, \pi_{\sigma,1}, ..., \pi_{\sigma,7}] = [\tfrac{1}{8}, \tfrac{1}{8}, ..., \tfrac{1}{8}] \tag{4.13}$$

It is important to recall that for binary to M-ary encoders the rate is defined in terms of the number of M-ary symbols produced per input bit. This is the reason why the decimal representation of the Bernoulli shift map encoder's output is used for the output mapping defined by $\gamma(\cdot)$ in Equation (4.12). As another example consider the rate 1/2, $v = 2$, MFD binary to 8-ary encoder with transfer function matrix

$$\mathbf{G}(D) = [(1 \quad D \quad D^2)^{(0)} \quad (1 \quad D + D^2 \quad D^2)^{(1)}] \tag{4.14}$$

The two sets of generators polynomials corresponding to the two M-ary outputs are clearly differentiated by the parenthesis in Equation (4.14). This encoder is equivalent to the one reported in [105] with representation over GF(8). The corresponding SE pair Markov model is defined by

$$\mathbf{P}_\sigma = \begin{array}{c} \\ s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{array} \begin{array}{c} \begin{array}{cccccccc} s_0 & s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 \end{array} \\ \left[\begin{array}{cccccccc} 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 \end{array}\right] \end{array} ; \quad \begin{array}{l} s_0 = (0, q_0);\ \gamma(s_0) = [0,0] \\ s_1 = (0, q_1);\ \gamma(s_1) = [1,3] \\ s_2 = (0, q_2);\ \gamma(s_2) = [2,2] \\ s_3 = (0, q_3);\ \gamma(s_3) = [3,1] \\ s_4 = (1, q_0);\ \gamma(s_4) = [4,4] \\ s_5 = (1, q_1);\ \gamma(s_5) = [5,7] \\ s_6 = (1, q_2);\ \gamma(s_6) = [6,6] \\ s_7 = (1, q_3);\ \gamma(s_7) = [7,5] \end{array} \tag{4.15}$$

with steady state probabilities given in Equation (4.13).

4.3 Signal Model and Power Spectral Density of the Binary to *M*-ary Convolutionally Coded Time Hopping Scheme for PPM IR-Based UWB

In this section the signal model of the binary to M-ary convolutionally coded time hopping PPM IR-based UWB scheme is introduced. The power spectral density of the signal model is then provided as obtained when using the results from Section 3.4. Finally, some PSD examples obtained when using known convolutional encoders are presented.

4.3.1 Signal Model for the Binary to *M*-ary Convolutionally Coded Time Hopping Scheme

The signal model for the binary to *M*-ary convolutionally coded TH-IR-based UWB scheme analyzed in this chapter is defined as

$$x(t) = \sum_{l=-\infty}^{\infty} \sum_{k=0}^{N_w-1} w(t - lT_s - kT_r - \beta_{\sigma_l,k} T_\beta) \tag{4.16}$$

where $w(t)$ is the waveform used; σ_l is the SE pair Markov model's state process with $N_\sigma = 2N_q$ states; $\beta_{\sigma_l,k} \in \{0,1,...,M-1\}$ is the k^{th} *M*-ary symbol transmitted when the encoder is in state σ_l; N_w is the number of *M*-ary symbols transmitted per state as defined by the encoder rate $1/\kappa = 1/N_w$; T_β is the PPM modulation index; T_r is the mean repetition time between pulses and $T_s = N_w T_r$ is the encoder's output vector time. As in Chapter 3 the notation $\pi_i = \pi_{\sigma,i}$ and $p_{ii'}^{(n)} = p_{\sigma,ii'}^{(n)}$ will be used for the steady state and the n^{th} step transition probabilities of the state process, σ_l, respectively.

If the rate $1/\kappa$ encoder is in state $\sigma_l = s_i$ at time l with corresponding output vector $z_l = \underline{z}_l = [z_l^{(0)}, z_l^{(1)}, ..., z_l^{(\kappa-1)}] = \gamma(\sigma_l = s_i) = \underline{\zeta}_{s_i} = [\zeta_{s_i}^{(0)}, \zeta_{s_i}^{(1)}, ..., \zeta_{s_i}^{(\kappa-1)}]$, then

$$\beta_{\sigma_l,k} = z_l^{(k)} = \zeta_{s_i}^{(k)} = \beta_{i,k}, \qquad k = 0,1,...,N_w - 1 \tag{4.17}$$

where $N_w = \kappa$ and $\beta_{i,k} = \zeta_{s_i}^{(k)} \in \{0,1,...,M-1\}$.

4.3.2 Power Spectral Density of the Binary to *M*-ary Convolutionally coded Time Hopping Signal

Using the results presented in Section 3.4 the PSD of the signal described by Equation (4.16) can be found to be

$$\begin{aligned} \overline{S}(f) &= S_{C.1}(f) + S_{C.3}(f) + S_D(f) && N_w = 1 \\ \overline{S}(f) &= S_{C.1}(f) + S_{C.2}(f) + S_{C.3}(f) + S_D(f) && N_w > 1 \end{aligned} \tag{4.18}$$

$$S_{C.1}(f) = \frac{1}{T_r}|W(f)|^2 - \frac{1}{T_s}|W(f)|^2 \left| \sum_{k=0}^{N_w-1} \sum_{i=0}^{N_\sigma-1} e^{j2\pi fkT_r} e^{j2\pi f\beta_{i,k}T_\beta} \pi_i \right|^2 \tag{4.19}$$

$$S_{C.2}(f) = \frac{2|W(f)|^2}{T_s} \sum_{k=0}^{N_w-2} \sum_{k'=k+1}^{N_w-1} \mathrm{Re}\left\{ e^{-j2\pi f(k'-k)T_r} \sum_{i=0}^{N_\sigma-1} e^{-j2\pi f(\beta_{i,k'} - \beta_{i,k})T_\beta} \pi_i \right\} \tag{4.20}$$

$$S_{C.3}(f) = \frac{2|W(f)|^2}{T_s} \mathrm{Re}\left\{ \sum_{k,k'=0}^{N_w-1} \sum_{m=1}^{\infty} \sum_{i,i'=0}^{N_\sigma-1} e^{-j2\pi f(k'-k)T_r} e^{-j2\pi fT_s m} e^{-j2\pi f(\beta_{i',k'} - \beta_{i,k})T_\beta} \pi_i (p_{ii'}^{(|m|)} - \pi_{i'}) \right\} \tag{4.21}$$

$$S_D(f) = \frac{1}{(T_s)^2}|W(f)|^2 \left| \sum_{k=0}^{N_w-1} \sum_{i=0}^{N_\sigma-1} e^{j2\pi fkT_r} e^{j2\pi f\beta_{i,k}T_\beta} \pi_i \right|^2 \sum_{r=-\infty}^{\infty} \delta(f - \tfrac{r}{T_s}) \tag{4.22}$$

where $S_C(f)$ is the PSD's continuous part, $S_D(f)$ is the PSD's discrete part and $W(f) = \Im\{w(t)\}$. Note spectral lines in the PSD are expected to appear at multiples of $1/(T_s)$.

Now we will focus on the PSD's discrete part, $S_D(f)$. If the steady state probabilities, π_i, and output mapping function, $\gamma(\cdot)$, of the state process, σ_l, are such that

$$\sum_{i=0}^{N_\sigma-1} e^{j2\pi f \beta_{i,k} T_\beta} \pi_i = \frac{1}{M} \sum_{\varsigma=0}^{M-1} e^{j2\pi f \varsigma T_\beta}; \quad k = 0,1,...,N_w - 1 \tag{4.23}$$

for every value of k and where $\beta_{i,k} \in \{0,1,...,M\}$, then Equation (4.22) becomes

$$S_D(f) = \frac{|W(f)|^2}{(T_s)^2} \sum_{r=-\infty}^{\infty} \left| \frac{1}{M} \sum_{\varsigma=0}^{M-1} e^{j2\pi (r/T_s)\varsigma T_\beta} \right|^2 \left| \sum_{k=0}^{N_w-1} e^{j2\pi (r/N_w)k} \right|^2 \delta(f - \tfrac{r}{T_s}) \tag{4.24}$$

where some of the absolute values were brought into the sum. By setting

$$T_\beta = \frac{T_r}{M} \tag{4.25}$$

and assuming Equation (4.23) is satisfied it can be proved that

$$\left| \sum_{k=0}^{N_w-1} e^{j2\pi (r/N_w)k} \right|^2 \left| \frac{1}{M} \sum_{\varsigma=0}^{M-1} e^{j2\pi (r/T_s)\varsigma T_\beta} \right|^2 = \begin{cases} (N_w)^2, & r = 0, \pm MN_w, \pm 2MN_w, ... \\ 0, & \text{otherwise} \end{cases} \tag{4.26}$$

and as a consequence the number of spectral lines in Equation (4.24) is reduced. Thus the space between successive spectral lines is increased from $1/T_s$ to $M/T_r = MN_w/T_s$ and it can be concluded that when the SE pair Markov model is such that Equation (4.23) is satisfied the number of spectral lines in the PSD can be reduced.

4.3.3 Power Spectral Density of a Non-Coded Binary PPM IR-Based UWB Signal with Random and Pseudo Random TH

It is convenient to define a signal model for a non-coded TH-IR-based UWB signal for comparison purposes. For meaningful comparisons the rate of the non-coded system must be equivalent to the convolutionally coded one. Similarly, the number of TH positions available within a pulse frame (as defined by the mean repetition time between pulses T_r) must be the same for both systems. Therefore the model adopted in this chapter for non-coded (uncoded) PPM TH-IR-based UWB signals is defined by

$$\boldsymbol{x}_u(t) = \sum_{l=-\infty}^{\infty} \sum_{k=0}^{N_w-1} w(t - lT_s - kT_r - \boldsymbol{y}_l T_y - c_{lN_w+k} T_c) \tag{4.27}$$

where $\boldsymbol{y}_l \in \{0,1\}$ is the l^{th} data bit; N_w is the number of transmitted pulses per bit; T_y is the binary PPM modulation index; $\{c_{lN_w+k}\}$ is a pseudo-random (PR) TH sequence taking values on the set $\{0, 1, 2, ..., (M/2)-1\}$ with period χ_c; T_c is the nominal shift caused by the TH sequence; T_r is the mean repetition time between pulses and $T_s = N_w T_r$ is the bit time.

It is assumed that $\{\boldsymbol{y}_l\}$ is a uniform distributed (balanced) i.i.d. binary sequence with p.m.f. as defined by Equation (4.1). Moreover, T_y and T_c are set to

$$T_c = \frac{T_r}{M/2} \quad \text{and} \quad T_y = \frac{T_c}{2} \tag{4.28}$$

Thus the PR-TH sequence, $\{c_{lN_w+k}\}$, combined with binary PPM defines M different pulse positions which are equivalent to the M different symbols used in the binary to M-ary convolutionally coded system. Furthermore, as in the rate $1/\kappa$ binary to M-ary convolutionally coded scheme κ M-ary symbols are generated per input bit, the number of transmitted pulses per bit, N_w, in the non-coded system will be set to $N_w = \kappa$.

Using Section 3.4 results the PSD of the signal described by Equation (4.27) can be found to be

$$\begin{aligned} \overline{S_u}(f) &= S_{u.C.1}(f) + S_{u.D}(f) \qquad N_w = 1 \\ \overline{S_u}(f) &= S_{u.C.1}(f) + S_{u.C.2}(f) + S_{u.D}(f) \quad N_w > 1 \end{aligned} \tag{4.29}$$

$$S_{u.C.1}(f) = \tfrac{1}{T_r} |W(f)|^2 - \tfrac{|W(f)|^2 |G_y(f)|^2}{T_s \Lambda_u} \sum_{n=0}^{\Lambda_u - 1} \left| \sum_{k=0}^{N_w - 1} e^{j2\pi fkT_r} e^{j2\pi fT_c c_{nN_w+k}} \right|^2 \tag{4.30}$$

$$S_{u.C.2}(f) = \tfrac{2|W(f)|^2}{T_s \Lambda_u} \sum_{k=0}^{N_w - 2} \sum_{k'=k+1}^{N_w - 1} \sum_{n=0}^{\Lambda_u - 1} \mathrm{Re}\{e^{-j2\pi f(k'-k)T_r} e^{-j2\pi fT_c (c_{nN_w+k'} - c_{nN_w+k})}\} \tag{4.31}$$

$$S_{u.D}(f) = \tfrac{|W(f)|^2 |G_y(f)|^2}{(T_s \Lambda_u)^2} \left| \sum_{k=0}^{N_w - 1} \sum_{n=0}^{\Lambda_u - 1} e^{j2\pi fkT_r} e^{j2\pi fnT_s} e^{j2\pi fT_c c_{nN_w+k}} \right|^2 \sum_{r=-\infty}^{\infty} \delta(f - \tfrac{r}{T_s \Lambda_u}) \tag{4.32}$$

where $S_{u.C}(f)$ is the PSD's continuous part, $S_{u.D}(f)$ is the PSD's discrete part, $\Lambda_u = \mathrm{lcm}(\chi_c, N_w)/N_w$ and

$$|G_y(f)|^2 = \mathrm{E}\{e^{-j2\pi fyT_y}\}\mathrm{E}\{e^{j2\pi fyT_y}\} = \left| \sum_{\varsigma=0}^{1} e^{j2\pi\varsigma T_y} p_{y,i} \right|^2 = \left| \frac{1}{2} \sum_{\varsigma=0}^{1} e^{j2\pi\varsigma fT_y} \right|^2 \tag{4.33}$$

Note how for this case spectral lines in the PSD are expected to appear at multiples of $1/(T_s \Lambda_u)$.

Let us focus on the PSD's discrete part, $S_{u.D}(f)$. Assuming the TH sequence is perfectly random (that is i.i.d. with uniform distribution) with c_{lN_p+k} uniformly distributed over the set $\{0, 1, 2, ..., (M/2)-1\}$ Equation (4.32) becomes

$$S_{u.D}(f) = \tfrac{|W(f)|^2}{(T_s)^2} \sum_{r=-\infty}^{\infty} \left| \frac{1}{2} \sum_{\varsigma=0}^{1} e^{j2\pi(r/T_s)\varsigma T_y} \right|^2 \left| \frac{1}{(M/2)} \sum_{l=0}^{(M/2)-1} e^{-j2\pi(r/T_s)lT_c} \right|^2 \left| \sum_{k=0}^{N_w - 1} e^{j2\pi(r/T_s)kT_r} \right|^2 \delta(f - \tfrac{r}{T_s}) \tag{4.34}$$

where $|G_y(f)|^2$ was substituted using Equation (4.33). Then for this ideal case

$$\left| \frac{1}{2} \sum_{\varsigma=0}^{1} e^{j2\pi(r/T_s)\varsigma T_y} \right|^2 \left| \frac{1}{(M/2)} \sum_{l=0}^{(M/2)-1} e^{-j2\pi(r/T_s)lT_c} \right|^2 \left| \sum_{k=0}^{N_w - 1} e^{j2\pi(r/T_s)kT_r} \right|^2 = \begin{cases} (N_w)^2, & r = 0, \pm MN_w, \pm 2MN_w, \ldots \\ 0, & \text{otherwise} \end{cases} \tag{4.35}$$

and therefore the number of spectral lines in Equation (4.34) is less than in Equation (4.32) where non-ideal TH is considered.

By comparing Equations (4.34) and (4.35) with Equations (4.24) and (4.26) it can be concluded that for PPM TH-IR-based UWB systems with uniform distributed i.i.d. binary data streams at the input the number of spectral lines suppressed in the PSD is equal for:

a) A binary to M-ary convolutionally coded scheme with state process, σ_l, and output

mapping function, $\gamma(\cdot)$, such that Equation (4.23) is strictly fulfilled. All TH is driven by the encoder and therefore M-ary PPM is implemented.

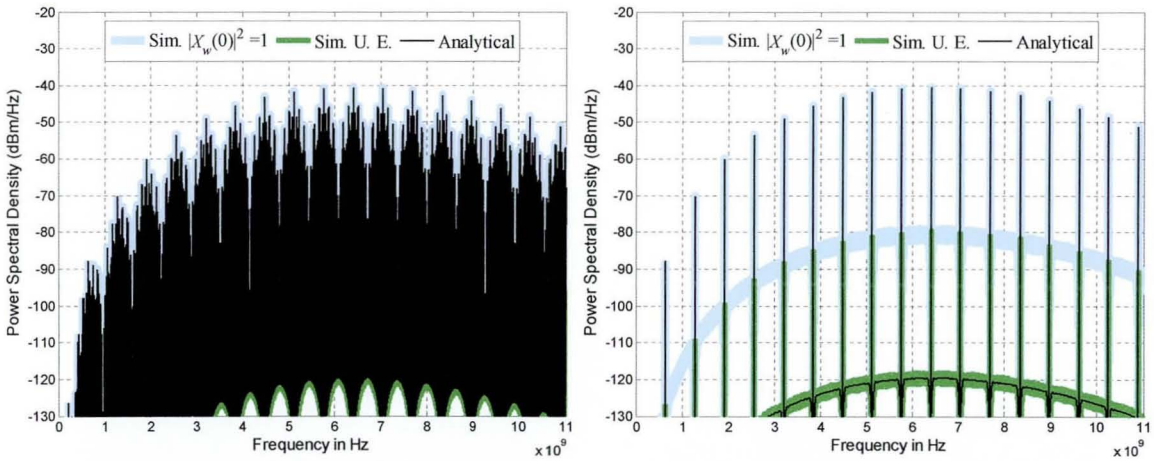
b) A system using binary PPM combined with perfectly random (ideal) TH over the set $\{0, 1, 2, ..., (M/2)-1\}$.

Therefore, in the subsequent Equation (4.23) will be referred as the ***spectral line suppression condition*** for binary to M-ary convolutional encoders.

4.3.4 PSD Examples of Binary to *M*-ary Convolutionally Coded and Non-Coded TH-IR-Based UWB Signals

This section presents PSD examples of TH-IR-based UWB signals using: non-coded binary PPM combined with random and pseudo-random (PR) 4-ary TH; the 8-ary Bernoulli shift map PCTH scheme; and the rate 1/3 MFD feedforward binary convolutional encoder $(5, 7, 7)_8$, [83], coupled to 8-ary PPM. In order to be able to compare the analytical results with results obtained from simulation, a double simulation-estimation procedure was performed for all the plots in this chapter as described in Appendix A.

Power spectral density plots for a TH-IR-based UWB signal employing binary PPM combined with pseudo-random and perfectly random 4-ary TH are shown in Figure 4-5. The sequence {0, 2, 1, 3, 2, 1, 0, 3} with period $\chi_c = 8$ was used for PR-TH in Figure 4-5a. Note the significant amount of spectral lines in this PSD plot appearing at $1/(\Lambda T_s) = 10$ MHz intervals. In contrast, when the TH sequence is assumed to be perfectly random (ideal case) over the set {0, 1, 2, 3}, the number of spectral lines in the PSD is considerably reduced as seen in Figure 4-5b where the spectral lines appear at 640 MHz intervals.



a) PR-TH driven by the periodic sequence {0, 2, 1, 3, 2, 1, 0, 3}

b) Ideal perfectly random TH with c_{lNw+k} uniform distributed over the set {0, 1, 2, 3}

Figure 4-5. Analytical and simulated PSDs for a TH-IR-based UWB system using binary PPM with pseudo-random and perfectly random 4-ary TH. The signal parameters are $N_w = 1$, $T_s = 12.5$ ns, $T_r = 12.5$ ns, $T_c = T_r/4$ and $T_\beta = T_r/8$. The 3rd derivative Gaussian pulse was used with duration $T_w \approx 0.35$ ns.

The Figure 4-6 shows that the spectral line suppression capabilities of the 8-ary Bernoulli shift map PCTH scheme are equivalent to those obtained with a system employing binary PPM

combined with perfectly random (ideal) TH over the set {0, 1, 2, 3}. These can be seen by noting that in this PSD plot the spectral lines appear at 640 MHz intervals as in Figure 4-5b.

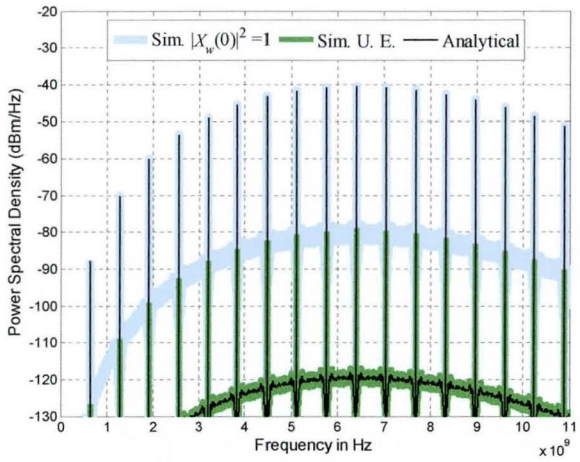


Figure 4-6. Analytical and simulated PSDs for a TH-IR-based UWB system using the 8-ary Bernoulli shift map PCTH scheme. The signal parameters are $N_w = 1$, $T_s = 12.5$ ns, $T_r = 12.5$ ns and $T_\beta = T_r/8$. The 3rd derivative Gaussian pulse was used with duration $T_w \approx 0.35$ ns.

Finally Figure 4-7 shows that good spectral line suppression capabilities are not necessarily obtained when combining good rate $1/\log_2(M)$ binary convolutional encoders with M-ary PPM. This plot was obtained by using the rate 1/3 MFD binary convolutional encoder $(5, 7, 7)_8$ with 8-ary PPM. The SE pair Markov model for this encoder has transition probabilities matrix, $\mathbf{P}_\sigma$, identical to the one defined in Equation (4.12) but with output mapping changed to

$$\gamma(s_0)=0 \quad \gamma(s_1)=7 \quad \gamma(s_2)=3 \quad \gamma(s_3)=4 \quad \gamma(s_4)=7 \quad \gamma(s_5)=0 \quad \gamma(s_6)=4 \quad \gamma(s_7)=3 \tag{4.36}$$

A consequence of this change is that Equation (4.23) is not satisfied, that is

$$\sum_{i=0}^{N_\sigma-1} e^{j2\pi f\beta_i T_\beta}\pi_i = \tfrac{1}{4}\{1+e^{j2\pi f3T_\beta}+e^{j2\pi f4T_\beta}+e^{j2\pi f7T_\beta}\} \neq \tfrac{1}{8}\sum_{\varsigma=0}^{M-1} e^{j2\pi f\varsigma T_\beta} \tag{4.37}$$

Therefore the signal's PSD for this system (shown in Figure 4-7) has 3 times more spectral lines than the PSD of signals generated when using the PCTH scheme or the system employing perfectly random TH (Figures 4-5b and 4-6).

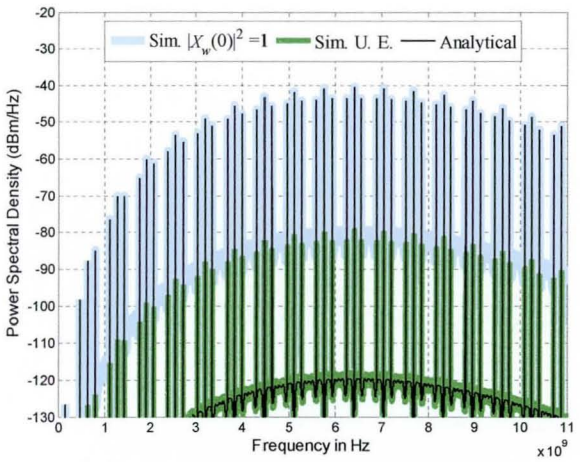


Figure 4-7. Analytical and simulated PSDs for a TH-IR-based UWB system using the rate 1/3 MFD feedforward binary convolutional code $(5, 7, 7)_8$ coupled to 8-ary PPM. The signal parameters are $N_w = 1$, $T_s = 12.5$ ns, $T_r = 12.5$ ns and $T_\beta = T_r/8$. The 3rd derivative Gaussian pulse was used with $T_w \approx 0.35$ ns.

4.4 Code Search Procedure for Feedforward Spectral Line Suppressive Binary to *M*-ary Convolutional Encoders

In this section a code search procedure for feedforward "spectral line suppressive" binary to *M*-ary convolutionally encoders with uniform distributed i.i.d. binary data streams is introduced. First the two main criteria used in the algorithm to discard and compare encoders are introduced. Then the code search procedure is described by using a flowchart and lastly the results of the code search are presented.

4.4.1 Verification of the Spectral Line Suppression Condition

From the results presented in the previous sections it can be seen that a binary to *M*-ary convolutional encoder can be effectively used to eliminate spectral lines in *M*-ary PPM IR-based UWB systems. In order to do so the encoder must satisfy Equation (4.23). For the case under study (uniform distributed i.i.d. binary data streams at the encoder input), all steady state probabilities are of the form $\pi_i = 1/(2N_q) = 1/N_\sigma$ as shown in Section 4.2.2 and therefore Equation (4.23) becomes

$$\frac{1}{N_\sigma}\sum_{i=0}^{N_\sigma-1} e^{j2\pi f \beta_{i,k} T_\beta} = \frac{1}{M}\sum_{\varsigma=0}^{M-1} e^{j2\pi f \varsigma T_\beta}; \quad k = 0,1,...,N_w - 1 \tag{4.38}$$

where $N_\sigma = 2N_q = 2^{v+1} = \eta M$ with $\eta \geq 1$.

Now assume k has been fixed and that the states, s_i, are relabelled to, $s_{i'}$, such that the first η_0 *M*-ary symbols, $\beta_{i',k} = \zeta_{s_{i'}}^{(k)}$, are equal to 0, the following η_1 *M*-ary symbols are equal to 1, and so on until the last η_{M-1} *M*-ary symbols are equal to $M-1$. Using this new ordering Equation (4.38) can be expanded as

$$\frac{1}{N_\sigma}\sum_{i=0}^{N_\sigma-1} e^{j2\pi f \beta_{i,k} T_\beta} = \frac{1}{N_\sigma}\left\{\sum_{i'=0}^{\eta_0-1} e^{0} + \sum_{i'=\eta_0}^{\eta_0+\eta_1-1} e^{j2\pi f T_\beta} + ... + \sum_{i'=N_\sigma-\eta_{M-1}-1}^{N_\sigma-1} e^{j2\pi f (M-1) T_\beta}\right\} = \frac{1}{M}\sum_{\varsigma=0}^{M-1} e^{j2\pi f \varsigma T_\beta} \tag{4.39}$$

Therefore if $N_\sigma = \eta M$ Equations (4.38) and (4.39) only hold when $\eta_0 = \eta_1 = ... = \eta_{M-1} = \eta$.

As a consequence a code search procedure aimed to find binary to *M*-ary convolutional encoders with good spectral line suppression characteristics must verify that the state process, $\boldsymbol{\sigma}_l$, and output mapping function, $\gamma(\cdot)$, are such that Equation (4.39) is satisfied or equivalently $\eta_0 = \eta_1 = ... = \eta_{M-1} = \eta = N_\sigma / M$. In fact this is the first discarding criteria used in the code search procedure introduced later in Section 4.4.3.

4.4.2 Superior Information Weight Spectrum Criteria

If an encoder complies with the spectral line suppression property, then an assessment of its error correction capabilities must be performed for comparison and selection purposes. In this work

the superior information weight spectrum (IWS) criteria, [105-107, 109, 111], will be adopted as a performance measure when comparing different encoders with the same basic characteristics such as total encoder memory, v, rate and output alphabet (e.g. binary, 4-ary, M-ary).

The superior IWS criteria is based on the form acquired by the upper bound on the bit error probability achieved when using binary and binary to M-ary convolutional encoders in output symmetric memoryless channels as introduced in Section 2.7.4. Remember this bound has the form ([83, 100-103, 105-107])

$$P_b \leq \sum_{d=d_{Mfree}}^{\infty} B_d P_2(d) \tag{4.40}$$

where $P_2(d)$ is the pairwise error probability between two codewords with M-ary Hamming distance d, and $\{B_d\}$ is the IWS introduced in Section 2.7.3. Thus, as explained in Section 2.7.4, a good design strategy consists of finding the encoder with the largest M-ary free distance, d_{Mfree} and minimum $\{B_d\}$. As an example assume Code 1 has M-ary free distance $d_{Mfree}^{(1)}$ and IWS $\{B_d^{(1)}\}$. Similarly assume Code 2 has free distance $d_{Mfree}^{(2)}$ and IWS $\{B_d^{(2)}\}$. Then Code 1 has superior IWS compared to Code 2 if $d_{Mfree}^{(1)} > d_{Mfree}^{(2)}$. Alternatively if $d_{Mfree}^{(1)} = d_{Mfree}^{(2)}$, then Code 1 has superior IWS compared to Code 2 if $B_{d_{Mfree}}^{(1)} = B_{d_{Mfree}}^{(2)}$, $B_{d_{Mfree}+1}^{(1)} = B_{d_{Mfree}+1}^{(2)}$,..., $B_{d_{Mfree}+i}^{(1)} = B_{d_{Mfree}+i}^{(2)}$ and $B_{d_{Mfree}+i+1}^{(1)} < B_{d_{Mfree}+i+1}^{(2)}$.

In the code search procedure used in this work the M-ary free distance and IWS are calculated using the FAST algorithm introduced by Cedervall and Johannesson, [103, 108].

4.4.3 Code Search Procedure for Feedforward Spectral Line Suppressive Binary to *M*-ary Convolutional Encoders

Once the two main encoder discarding/comparison criteria have been defined we can introduce the code search procedure. Based on the previous discussion, there exists the possibility to find feedforward binary to *M-ary* convolutional encoders with better BER performance than PCTH and the same spectral line suppression capabilities. Therefore an extended search to find this kind of spectral line suppressive (SLS) convolutional encoders has been performed.

The flowchart of the code search procedure is shown in Figure 4-8. The goal is to find the encoders with spectral line suppression capabilities and the best possible IWS. This code search procedure is similar to that introduced in [105, 106] with some modifications. Briefly, the main difference in our code search procedure is the verification the encoder's spectral line suppression capabilities, which is novel. Thus encoders with inferior or no spectral line suppression capabilities are discarded before calculating the encoder's IWS.

The code search procedure has not been restricted to rate 1 binary to M-ary convolutional codes as rate 1/2 and rate 1/3 encoders have been included. Similarly the total encoder memory, v, can be

larger than $\log_2(M)-1=L-1$. Remember from Sections 2.7.6 and 4.2.2 that for a rate $1/\kappa$ binary to M-ary convolutional code κ sets of $L=\log_2(M)$ generators must be provided, therefore the following notation will be used for the transfer function matrix

$$\mathbf{G}(D)=[\{g_0^{(0)}(D)\ g_1^{(0)}(D)...g_{L-1}^{(0)}(D)\}^{(0)}\ \ ...\ \ \{g_0^{(\kappa-1)}(D)\ g_1^{(\kappa-1)}(D)...g_{L-1}^{(\kappa-1)}(D)\}^{(\kappa-1)}] \tag{4.41}$$

The i^{th} generator of the k^{th} set corresponding to the k^{th} M-ary output has the form

$$g_i^{(k)}(D)=a_{i,0}^{(k)}+a_{i,1}^{(k)}D+...+a_{i,v}^{(k)}D^v \tag{4.42}$$

where D is the delay operator $a_{i,j}^{(k)}\in\{0,1\}$, $i=0,1,...,L-1$ and $k=0,1,...,\kappa-1$. If the input sequence, $y=\{y_0y_1y_2...y_l...\}$, is expressed in terms of the delay operator as

$$y(D)=y_0+y_1D+y_2D^2+...+y_lD^l+... \tag{4.43}$$

then the M-ary symbols produced by the k^{th} generator set can be obtained by using

$$z^{(k)}(D)=2^{(L-1)}y(D)g_0^{(k)}(D)+2^{(L-2)}y(D)g_1^{(k)}(D)+...+y(D)g_{L-1}^{(k)}(D) \tag{4.44}$$

Thus the encoder output mapping function, $\gamma(\cdot)$, is defined by the generator sets in Equation (4.41) and the binary to M-ary conversion defined by Equation (4.44).

For a better understanding of the code search procedure several main tasks have been identified in the flowchart of Figure 4-8. *Task 1* consists of defining the main parameters of the code search such as the M-ary alphabet, the total encoder memory, v, and the code rate, $1/\kappa$. In order to provide rate adaptability a nested code search approach has been adopted, [112]. This means that first the best generator set (in terms of spectral line suppression capabilities and IWS) for rate 1 SLS encoders must be found. Afterwards this set is used as the first generator set of a rate 1/2 encoder while the second generator set is searched in such a way that the best possible rate 1/2 SLS encoder is formed. This approach is then repeated by searching the additional generator set needed to form the best possible rate 1/3 SLS encoder from the rate 1/2 SLS encoder.

During *Task 2* the target M-ary free distance is set. Initially the following upper bound on the M-ary free distance of rate $1/\kappa$ binary to M-ary convolutional codes is used ([105])

$$d_{Mfree}\le d_{Mtar}=(v+1)\kappa \tag{4.45}$$

If no MFD spectral line suppressive encoder is found when using the current target free distance then d_{Mtar} is decreased until an encoder is found.

Note the total number of transfer function matrices to be searched in every stage of the nested code search procedure is $2^{L(v+1)}$. Nevertheless this number is reduced in *Task 3* by identifying and discarding encoders with identical IWS. Three main equivalence criteria have been used in Taks 3 ([105, 107]):

1. Generator sets with interchanged generator polynomials are equivalent. This is due to a change in the generators order in the k^{th} set from $\{g_0^{(k)}(D)\ g_1^{(k)}(D)...g_{L-1}^{(k)}(D)\}^{(k)}$ to $\{g_3^{(k)}(D)\ g_{L-2}^{(k)}(D)...g_4^{(k)}(D)\}^{(k)}$ (or any other permutation) does not modify the M-ary Hamming weight of the M-ary symbols generated.

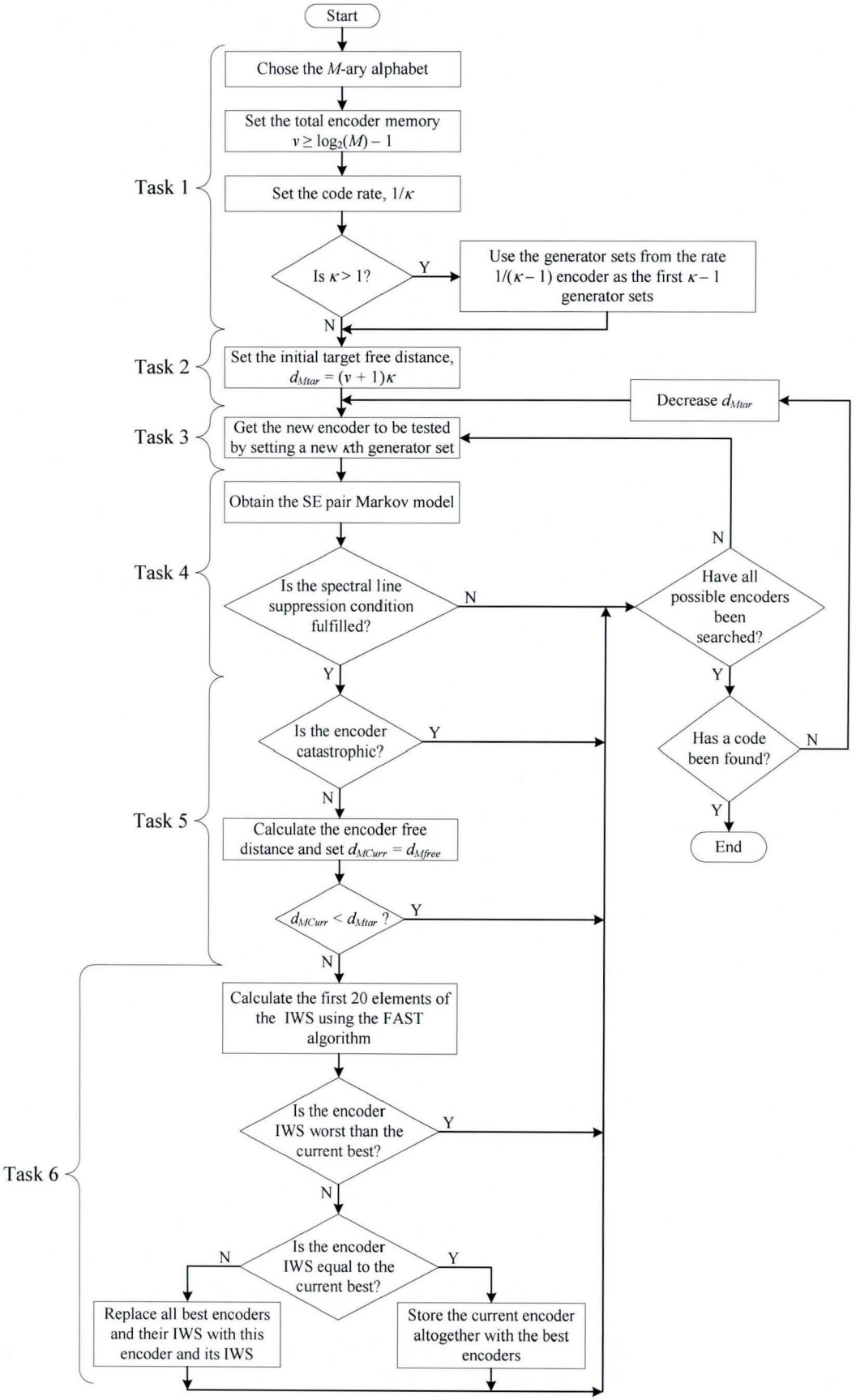


Figure 4-8. Flowchart diagram of the code search procedure for feedforward SLS binary to M-ary convolutional encoders.

2. Any rate $1/\kappa$ binary to M-ary code is equivalent to one with

$$g_{L-1}^{(k)}(D) = 1 + a_{L-1,1}^{(k)} D + \ldots + a_{L-1,v}^{(k)} D^v \quad \text{and} \quad g_i^{(k)}(D) = 0 + a_{i,1}^{(k)} D + \ldots + a_{i,v}^{(k)} D^v \tag{4.46}$$

for $i = 0,1,\ldots,L-2$. This equivalence rule halves the number of encoders to search.

3. Two generators sets are equivalent (in the IWS sense) if one can be obtained by performing a column operation in the other.

Task 4 verifies the encoder's spectral line suppression capabilities. The encoder is discarded if it does not satisfy the spectral line suppression condition introduced in Sections 4.3.2 and 4.4.1.

Task 5 verifies if the current encoder is catastrophic (see Section 2.7.3). If the encoder is not catastrophic then its free distance, $d_{MCurr} = d_{Mfree}$, is calculated and compared with the target free distance, d_{Mtar}.

Task 6 is performed if the current free distance, d_{MCurr}, is at least equal to the target free distance, d_{Mtar}. If this task is reached the current encoder's IWS is calculated and compared to the best IWS. The encoder is discarded if its IWS is worst than the current best. If the encoder's IWS is equal to the current best it is saved in a file containing all encoders found with the same IWS. Finally if the encoder's IWS is better than the current best all previously stored encoders are discarded, the current encoder is saved and its IWS is set as the current best. The algorithm ends when an encoder with spectral line suppressive capabilities and the best possible IWS is found.

All different tasks involved in of the code search procedure flowchart have been programmed in MATLAB®.

4.4.4 Code Search Results for Feedforward Spectral Line Suppressive Binary to *M*-ary Convolutional Encoders

New maximum free distance (MFD) feedforward binary to M-ary convolutional encoders with the spectral line suppression property have been found using the code search procedure introduced in the previous section. Table 4-1 presents the code search results for feedforward spectral line suppressive (FSLS) binary to 16-ary, 32-ary, 64-ary and 128-ary convolutional encoders with rates 1, 1/2 and 1/3.

The results in Table 4-1 are interpreted as follows: for each M-ary alphabet and total encoder memory, v, the first line gives the set of generators $\{g_0^{(0)}(D)\ g_1^{(0)}(D) \ldots g_{L-1}^{(0)}(D)\}^{(0)}$ (in octal form), the M-ary free distance, d_{Mfree}, and the first 18 components of the information weight and distance spectrums for rate 1 FSLS convolutional encoders. The second and third lines give the second and third generator sets needed to form the rate 1/2 and 1/3 FSLS convolutional encoders with their respective spectrums. All these encoders have optimum IWS within the FSLS set while the binary to 16-ary encoders presented here have better IWS than the encoders reported in [105-107].

Table 4-1. Best feedforward spectral line suppressive binary to M-ary convolutional encoders with the first 18 elements of its IWS and distance spectrum.

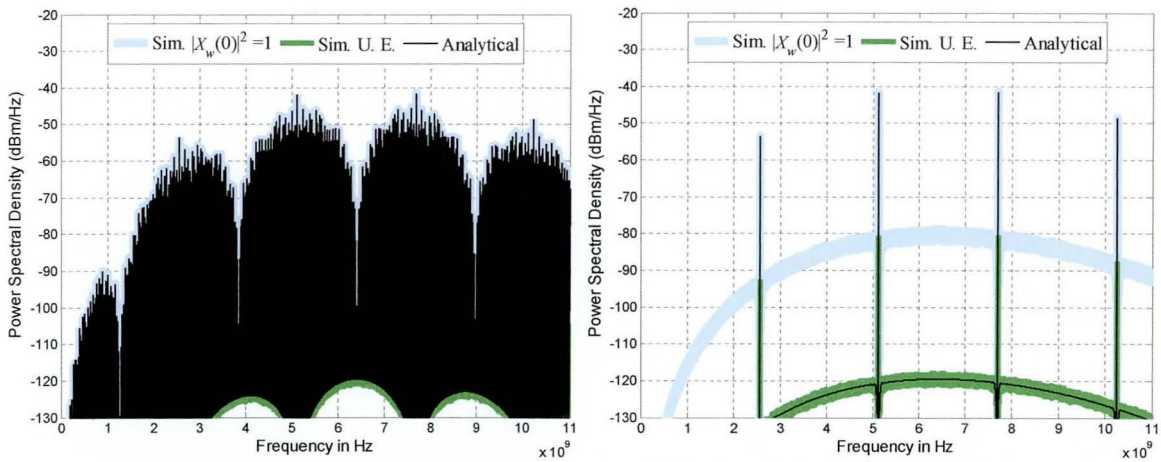
M	v	Generator sets in octal form	d_{MSfree}	Information weight spectrum (IWS): $[B_{dbSfree}, B_{dbSfree+1}, \ldots, B_{dbSfree+18}]$ Distance spectrum: $\{A_{dbSfree}, A_{dbSfree+1}, \ldots, A_{dbSfree+18}\}$
16	3	(17, 15, 13, 12)	4	[1, 2, 5, 12, 26, 56, 118, 244, 499, 1010, 2027, 4040, 8004, 15776, 30956, 60504, 117845, 228818] {1, 1, 2, 4, 7, 13, 24, 44, 81, 149, 274, 504, 927, 1705, 3136, 5768, 10609, 19513}
		(15, 13, 12, 17)	8	[1, 0, 2, 0, 5, 0, 12, 0, 26, 0, 56, 0, 118, 0, 244, 0, 499, 0] {1, 0, 1, 0, 2, 0, 4, 0, 7, 0, 13, 0, 24, 0, 44, 0, 81, 0}
		(13, 17, 15, 12)	12	[1, 0, 0, 2, 0, 0, 5, 0, 0, 12, 0, 0, 26, 0, 0, 56, 0, 0] {1, 0, 0, 1, 0, 0, 2, 0, 0, 4, 0, 0, 7, 0, 0, 13, 0, 0}
	4	(21, 10, 4, 2)	5	[1, 2, 5, 14, 32, 74, 171, 382, 847, 1864, 4060, 8788, 18917, 40506, 86361, 183426, 388248, 819294] {1, 1, 2, 5, 9, 18, 37, 73, 146, 293, 585, 1170, 2341, 4681, 9362, 18725, 37449, 74898}
		(22, 10, 4, 3)	10	[1, 0, 2, 0, 5, 0, 12, 5, 23, 14, 48, 41, 95, 112, 198, 260, 412, 602] {1, 0, 1, 0, 2, 0, 4, 2, 6, 4, 11, 10, 19, 24, 36, 48, 68, 100}
		(24, 21, 10, 2)	15	[1, 0, 0, 2, 0, 0, 5, 0, 0, 12, 0, 8, 20, 0, 22, 40, 4, 56] {1, 0, 0, 1, 0, 0, 2, 0, 0, 4, 0, 3, 5, 0, 6, 9, 1, 13}
	5	(42, 21, 11, 4)	6	[1, 2, 5, 17, 35, 87, 219, 474, 1121, 2570, 5669, 12879, 28605, 63035, 139739, 306254, 670001, 1464450] {1, 1, 2, 6, 9, 21, 46, 86, 189, 389, 786, 1662, 3409, 7029, 14630, 30134, 62357, 129141}
		(50, 32, 13, 4)	12	[1, 0, 2, 0, 5, 0, 12, 11, 17, 24, 44, 63, 84, 177, 207, 327, 501, 795] {1, 0, 1, 0, 2, 0, 4, 4, 4, 6, 11, 14, 18, 35, 41, 54, 88, 129}
		(43, 22, 12, 4)	18	[1, 0, 0, 2, 0, 0, 5, 0, 0, 12, 0, 19, 9, 0, 36, 28, 9, 91] {1, 0, 0, 1, 0, 0, 2, 0, 0, 4, 0, 6, 2, 0, 9, 7, 2, 20}
	6	(124, 66, 21, 10)	7	[1, 2, 5, 20, 36, 95, 247, 521, 1252, 2993, 6532, 15150, 34443, 76167, 172361, 384126, 849663, 1891486] {1, 1, 2, 7, 9, 23, 53, 95, 215, 466, 921, 2008, 4225, 8708, 18608, 38973, 81492, 172380}
		(143, 61, 44, 10)	14	[1, 0, 2, 0, 5, 0, 12, 19, 9, 30, 37, 81, 108, 145, 254, 342, 629, 773] {1, 0, 1, 0, 2, 0, 4, 6, 2, 8, 9, 18, 23, 30, 48, 62, 110, 125}
		(164, 114, 106, 21)	21	[1, 0, 0, 2, 0, 0, 5, 0, 0, 12, 2, 26, 0, 3, 40, 24, 34, 59] {1, 0, 0, 1, 0, 0, 2, 0, 0, 4, 1, 7, 0, 1, 10, 6, 7, 13}
	7	(210, 161, 122, 104)	8	[1, 2, 5, 23, 41, 113, 293, 584, 1629, 3499, 8228, 19429, 43371, 100244, 226734, 512425, 1156579, 2596939] {1, 1, 2, 8, 10, 26, 58, 106, 262, 515, 1122, 2426, 5059, 10926, 23161, 49388, 105309, 224199}
		(305, 122, 113, 46)	16	[1, 0, 2, 0, 5, 0, 12, 28, 0, 38, 65, 49, 130, 187, 273, 369, 764, 888] {1, 0, 1, 0, 2, 0, 4, 8, 0, 10, 14, 11, 27, 35, 53, 62, 130, 138}
		(326, 136, 42, 21)	24	[1, 0, 0, 2, 0, 0, 5, 0, 0, 12, 11, 17, 0, 10, 45, 22, 38, 68] {1, 0, 0, 1, 0, 0, 2, 0, 0, 4, 4, 4, 0, 3, 11, 5, 8, 15}
32	4	(33, 27, 25, 17, 35)	5	[1, 2, 5, 12, 28, 62, 136, 294, 628, 1328, 2787, 5810, 12043, 24840, 51016, 104380, 212848, 432732] {1, 1, 2, 4, 8, 15, 29, 56, 108, 208, 401, 773, 1490, 2872, 5536, 10671, 20569, 39648}
		(25, 17, 35, 33, 27)	10	[1, 0, 2, 0, 5, 0, 12, 0, 28, 0, 62, 0, 136, 0, 294, 0, 628, 0] {1, 0, 1, 0, 2, 0, 4, 0, 8, 0, 15, 0, 29, 0, 56, 0, 108, 0}
		(27, 17, 25, 35, 33)	15	[1, 0, 0, 2, 0, 0, 5, 0, 0, 12, 0, 0, 28, 0, 0, 62, 0, 0] {1, 0, 0, 1, 0, 0, 2, 0, 0, 4, 0, 0, 8, 0, 0, 15, 0, 0}
	5	(41, 20, 10, 4, 2)	6	[1, 2, 5, 12, 30, 68, 154, 344, 763, 1670, 3631, 7844, 16856, 36040, 76740, 162800, 344245, 725770] {1, 1, 2, 4, 9, 17, 34, 68, 137, 273, 546, 1092, 2185, 4369, 8738, 17476, 34953, 69905}
		(50, 20, 11, 4, 2)	12	[1, 0, 2, 0, 5, 0, 12, 0, 28, 5, 59, 14, 128, 41, 271, 112, 566, 292] {1, 0, 1, 0, 2, 0, 4, 0, 8, 2, 14, 4, 27, 10, 51, 24, 96, 56}
		(60, 21, 10, 4, 2)	18	[1, 0, 0, 2, 0, 0, 5, 0, 0, 12, 0, 0, 28, 0, 8, 56, 0, 22] {1, 0, 0, 1, 0, 0, 2, 0, 0, 4, 0, 0, 8, 0, 3, 13, 0, 6}
	6	(102, 44, 20, 10, 5)	7	[1, 2, 5, 12, 33, 71, 167, 377, 860, 1886, 4197, 9188, 20101, 43568, 94407, 203331, 437037, 935829] {1, 1, 2, 4, 10, 17, 37, 74, 154, 304, 629, 1269, 2590, 5240, 10686, 21673, 44089, 89526}
		(141, 103, 20, 10, 4)	14	[1, 0, 2, 0, 5, 0, 12, 0, 28, 11, 53, 22, 122, 67, 251, 180, 520, 456] {1, 0, 1, 0, 2, 0, 4, 0, 8, 4, 12, 6, 26, 16, 47, 38, 88, 86}
		(111, 50, 20, 13, 4)	21	[1, 0, 0, 2, 0, 0, 5, 0, 0, 12, 0, 0, 28, 0, 18, 46, 0, 34] {1, 0, 0, 1, 0, 0, 2, 0, 0, 4, 0, 0, 8, 0, 6, 10, 0, 9}
	7	(205, 41, 107, 20, 10)	8	[1, 2, 5, 12, 36, 72, 175, 388, 930, 1989, 4485, 9947, 21952, 47695, 104436, 226879, 491513, 1059996] {1, 1, 2, 4, 11, 17, 39, 77, 169, 323, 681, 1396, 2871, 5822, 12021, 24606, 50453, 103231}
		(341, 145, 120, 103, 10)	16	[1, 0, 2, 0, 5, 0, 12, 0, 28, 18, 46, 32, 116, 82, 255, 198, 538, 503] {1, 0, 1, 0, 2, 0, 4, 0, 8, 6, 10, 8, 25, 18, 49, 40, 93, 91}
		(303, 111, 104, 41, 20)	24	[1, 0, 0, 2, 0, 0, 5, 0, 0, 12, 0, 0, 28, 0, 30, 34, 0, 48, 96, 8, 124, 188] {1, 0, 0, 1, 0, 0, 2, 0, 0, 4, 0, 0, 8, 0, 9, 7, 0, 12, 20, 2, 26, 36}
64	5	(76, 75, 71, 67, 55, 53)	6	[1, 2, 5, 12, 28, 64, 142, 312, 678, 1460, 3120, 6624, 13987, 29394, 61515, 128264, 266568, 552384] {1, 1, 2, 4, 8, 16, 31, 61, 120, 236, 464, 912, 1793, 3525, 6930, 13624, 26784, 52656}
		(67, 55, 53, 76, 75, 71)	12	[1, 0, 2, 0, 5, 0, 12, 0, 28, 0, 64, 0, 142, 0, 312, 0, 678, 0] {1, 0, 1, 0, 2, 0, 4, 0, 8, 0, 16, 0, 31, 0, 61, 0, 120, 0}
		(55, 53, 76, 75, 71, 67)	18	[1, 0, 0, 2, 0, 0, 5, 0, 0, 12, 0, 0, 28, 0, 0, 64, 0, 0] {1, 0, 0, 1, 0, 0, 2, 0, 0, 4, 0, 0, 8, 0, 0, 16, 0, 0}
	6	(101, 40, 20, 10, 4, 2)	7	[1, 2, 5, 12, 28, 66, 148, 330, 728, 1592, 3459, 7462, 16015, 34212, 72788, 154308, 326072, 687060] {1, 1, 2, 4, 8, 17, 33, 66, 132, 264, 529, 1057, 2114, 4228, 8456, 16913, 33825, 67650}
		(140, 41, 20, 10, 4, 2)	14	[1, 0, 2, 0, 5, 0, 12, 0, 28, 0, 64, 5, 139, 14, 304, 41, 655, 112] {1, 0, 1, 0, 2, 0, 4, 0, 8, 0, 16, 2, 30, 4, 59, 10, 115, 24}
		(120, 40, 21, 10, 4, 2)	21	[1, 0, 0, 2, 0, 0, 5, 0, 0, 12, 0, 0, 28, 0, 0, 64, 0, 8] {1, 0, 0, 1, 0, 0, 2, 0, 0, 4, 0, 0, 8, 0, 0, 16, 0, 3}
	7	(202, 104, 40, 20, 10, 5)	8	[1, 2, 5, 12, 28, 69, 151, 343, 761, 1676, 3680, 7974, 17277, 37172, 79705, 170316, 362604, 770139] {1, 1, 2, 4, 8, 18, 33, 69, 138, 278, 564, 1128, 2285, 4597, 9270, 18692, 37656, 75942}
		(301, 203, 40, 20, 10, 4)	16	[1, 0, 2, 0, 5, 0, 12, 0, 28, 0, 64, 11, 133, 22, 298, 67, 635, 180] {1, 0, 1, 0, 2, 0, 4, 0, 8, 0, 16, 4, 28, 6, 58, 16, 111, 38}
		(320, 122, 101, 40, 10, 4)	24	[1, 0, 0, 2, 0, 0, 5, 0, 0, 12, 0, 0, 28, 0, 0, 64, 0, 17] {1, 0, 0, 1, 0, 0, 2, 0, 0, 4, 0, 0, 8, 0, 0, 16, 0, 6}
128	6	(173,140,117,136,67,41,10)	7	[1, 2, 5, 12, 28, 64, 144, 318, 696, 1510, 3252, 6960, 14816, 31392, 66243, 139282, 291915, 610056] {1, 1, 2, 4, 8, 16, 32, 63, 125, 248, 492, 976, 1936, 3840, 7617, 15109, 29970, 59448}
		(173,140,117,136,67,41,10)	14	[1, 0, 2, 0, 5, 0, 12, 0, 28, 0, 64, 0, 144, 0, 318, 0, 696, 0] {1, 0, 1, 0, 2, 0, 4, 0, 8, 0, 16, 0, 32, 0, 63, 0, 125, 0}
		(173,140,117,136,67,41,10)	21	[1, 0, 0, 2, 0, 0, 5, 0, 0, 12, 0, 0, 28, 0, 0, 64, 0, 0] {1, 0, 0, 1, 0, 0, 2, 0, 0, 4, 0, 0, 8, 0, 0, 16, 0, 0}
	7	(201, 100, 40, 20, 10, 4, 2)	8	[1, 2, 5, 12, 28, 64, 146, 324, 714, 1560, 3384, 7296, 15651, 33414, 71055, 150564, 318036, 669888] {1, 1, 2, 4, 8, 16, 33, 65, 130, 260, 520, 1040, 2081, 4161, 8322, 16644, 33288, 66576}
		(240, 100, 41, 20, 10, 4, 2)	16	[1, 0, 2, 0, 5, 0, 12, 0, 28, 0, 64, 0, 144, 5, 315, 14, 688, 41] {1, 0, 1, 0, 2, 0, 4, 0, 8, 0, 16, 0, 32, 2, 62, 4, 123, 10}
		(300, 101, 40, 20, 10, 4, 2)	24	[1, 0, 0, 2, 0, 0, 5, 0, 0, 12, 0, 0, 28, 0, 0, 64, 0, 0] {1, 0, 0, 1, 0, 0, 2, 0, 0, 4, 0, 0, 8, 0, 0, 16, 0, 0}

The next sections show that improved bit error rate (BER) performance over the PCTH scheme can be obtained by using the new encoders reported in Table 4-1 while preserving the same spectral line suppression properties.

4.5 PSD Comparisons of the FSLS Binary to 32-ary Convolutional Encoders and 32-ary PCTH

This section presents PSD examples of TH-IR-based UWB signals obtained when using non-coded binary PPM with random and pseudo-random 16-ary TH, the 32-ary Bernoulli map PCTH scheme, and several of the rate 1 FSLS binary to 32-ary convolutional encoders introduced in Table 4-1.

The PSD plots for non-coded binary PPM combined with pseudo-random and perfectly random 16-ary TH are shown in Figure 4-9. In Figure 4-9a the periodic sequence {0, 14, 1, 5, 13, 6, 3, 15, 7, 11, 8, 12, 9, 2, 10, 4} with $\chi_c = 16$ was used for TH. It can be seen that when the TH sequence is not assumed to be perfectly random the PSD has a significant amount of spectral lines as shown in Figure 4-9a. Particularly, the spectral lines appear at $1/(\Lambda T_s) = 5$ MHz intervals in this figure. When the TH sequence is assumed to be perfectly random (ideal case) over the set {0, 1,..., 15}, the number of spectral lines is considerably reduced as shown in Figure 4-9b. Note how in this figure the spacing between successive spectral lines increases to 2.56 GHz. As reference, recall from Section 2.4 that the power spectrum of a periodic pulse train consists of spectral lines spaced at multiples of the pulse repetition frequency $1/T_{Per}$. Thus the power spectrum of a periodic pulse train with the same set of parameters consists of spectral lines appearing at $1/T_{Per} = 80$ MHz intervals as shown in Figure 4-10.



a) PR-TH driven by the periodic sequence {0, 14, 1, 5, 13, 6, 3, 15, 7, 11, 8, 12, 9, 2, 10, 4}

b) Ideal perfectly random TH with c_{lNw+k} uniform distributed over the set {0, 1, ..., 15}

Figure 4-9. Analytical and simulated PSDs for a TH-IR-based UWB system using binary PPM with pseudo-random and perfectly random 16-ary TH. The signal parameters are $N_w = 1$, $T_s = 12.5$ ns, $T_r = 12.5$ ns, $T_c = T_r/16$ and $T_\beta = T_r/32$. The 3rd derivative Gaussian pulse was used with duration $T_w \approx 0.35$ ns.

The Figures 4-11 and 4-12 show that by using the 32-ary Bernoulli shift map PCTH scheme or the binary to 32-ary convolutional encoders from Table 4-1 the same amount of spectral lines can be eliminated as when using binary PPM with perfectly random 16-ary TH. This is because the spectral line suppression property is satisfied, that is

$$\frac{1}{N_\sigma}\sum_{i=0}^{N_\sigma-1} e^{j2\pi f \beta_i T_\beta} = \frac{1}{32}\sum_{\varsigma=0}^{31} e^{j2\pi f \varsigma T_\beta} \tag{4.47}$$

for these particular encoders.

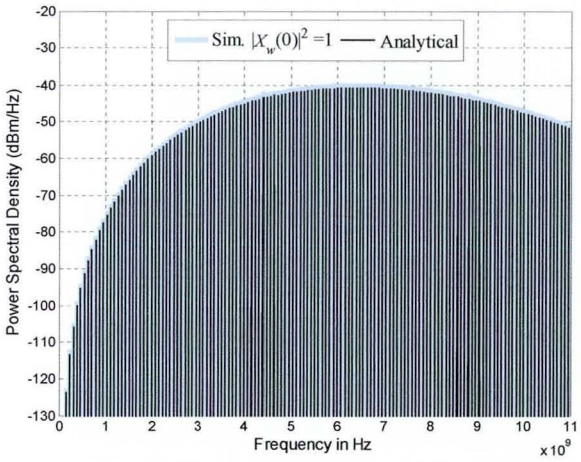


Figure 4-10. Analytical and simulated PSDs of a periodic pulse train. The signal pulse repetition period is $T_{Per} = 12.5$ ns. The 3rd derivative Gaussian pulse was used with duration $T_w \approx 0.35$ ns.

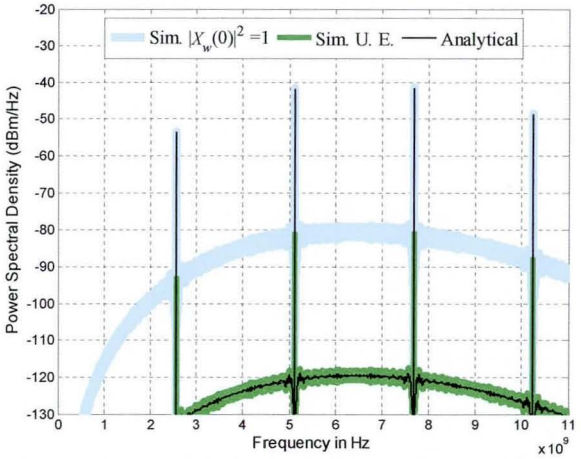


Figure 4-11. Analytical and simulated PSDs for a TH-IR-based UWB system using the 32-ary Bernoulli shift map PCTH scheme. The signal parameters are $N_w = 1$, $T_s = 12.5$ ns, $T_r = 12.5$ ns and $T_\beta = T_r/32$. The 3rd derivative Gaussian pulse was used with duration $T_w \approx 0.35$ ns.

By comparing Figures 4-9b, 4-11 and 4-12 it can be seen that even though the FSLS binary to 32-ary convolutional encoders, the 32-ary Bernoulli shift map PCTH scheme and the system employing binary PPM with perfectly random 16-ary TH eliminate the same amount of spectral lines, some differences can be detected on the PSD's continuous part. The differences are explained by the dependence of the continuous PSD on the steady state and n^{th} step transition probabilities of the SE pair Markov model for convolutionally coded systems (including the PCTH scheme), as evidenced by Equations (4.18) to (4.21). Nevertheless, note how for the binary to 32-ary convolutional encoders given in Table 4-1 the continuous PSD becomes smoother as v increases.

Finally note how there is good agreement between the analytical and simulated results for both the PSD's continuous part (Sim. U.E.) and the PSD's discrete part (Sim. $|X_w(0)|^2=1$) in the all the figures presented in this and previous sections.

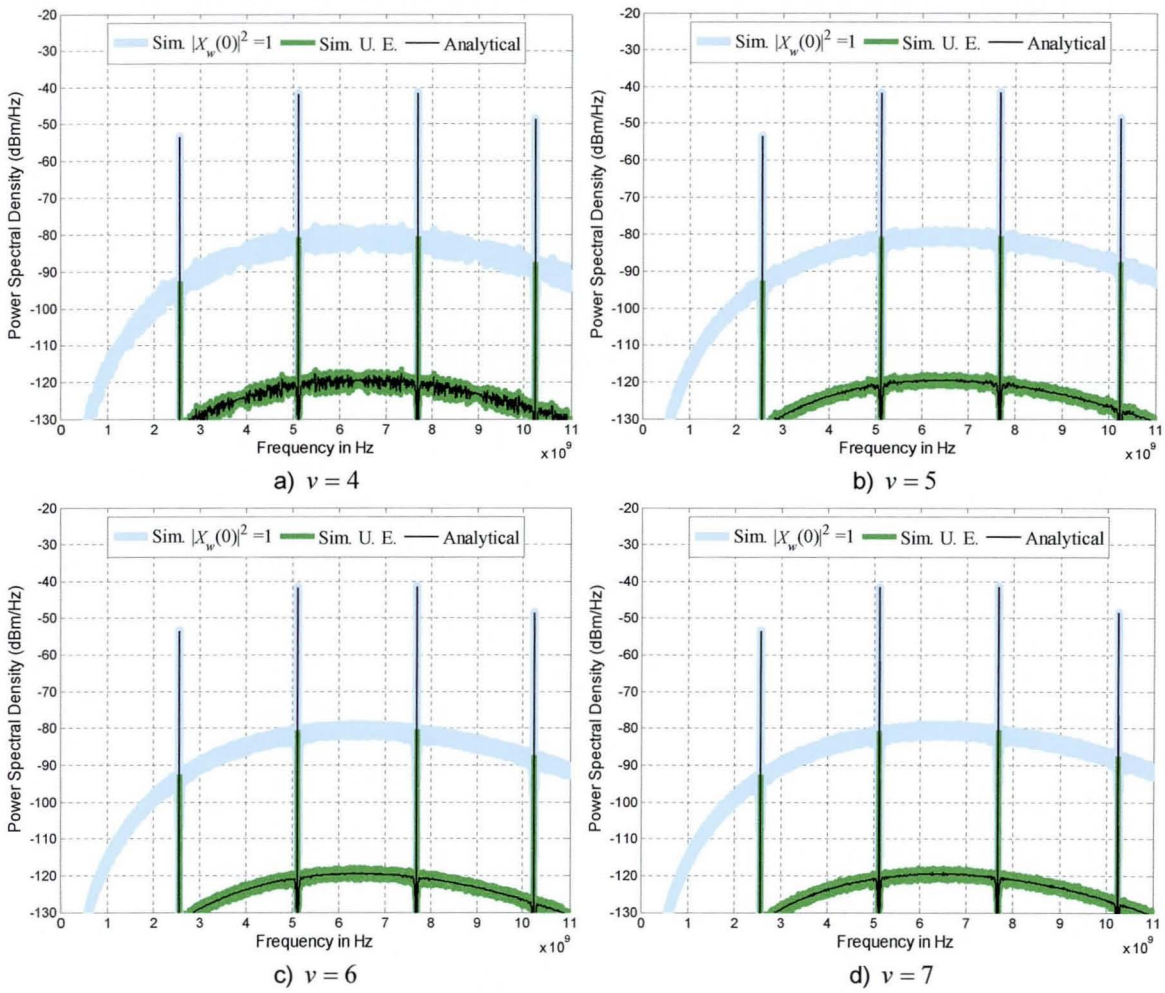


Figure 4-12. Analytical and simulated PSDs for a TH-IR-based UWB system using several of the FSLS binary to 32-ary convolutional encoders from Table 4-1. The signal parameters are $N_w = 1$, $T_s = 12.5$ ns, $T_r = 12.5$ ns and $T_\beta = T_r/32$ ns. The 3rd derivative Gaussian pulse was used with duration $T_w \approx 0.35$ ns.

4.6 BER Performance Comparisons of FSLS Binary to *M*-ary Convolutional Encoders and the PCTH Scheme

In this section comparisons in terms of BER performance between FSLS binary to *M*-ary convolutional encoders and the PCTH scheme are provided for hard Viterbi decoding (HVD) and soft Viterbi decoding (SVD). It is shown that performing HVD based on the *M*-ary Hamming distances improves the BER performance compared to HVD decoding based on binary Hamming distances as in the PCTH scheme.

As explained in Section 2.3, orthogonal PPM TH-IR-based UWB is a particular implementation of an orthogonal signalling system. Therefore an IR-based UWB system using binary to *M*-ary convolutional encoders can be described by the block diagram shown in Figure 4-13.

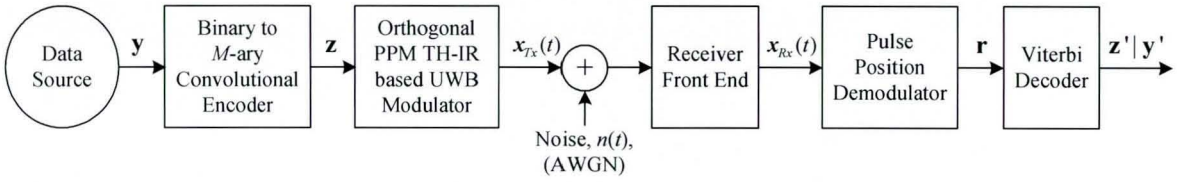


Figure 4-13. Block diagram of a binary to M-ary convolutionally coded TH-IR-based UWB system.

Using the same notation as in Section 2.7.5 the data source produces a binary sequence $\mathbf{y} = \{y_0, y_1, ..., y_{N-1}\}$ of length N which defines a path generating the code sequence $\mathbf{z} = \{\underline{\mathbf{z}_0}, \underline{\mathbf{z}_1}, ..., \underline{\mathbf{z}_{N-1}}\} = \{z_0^{(0)}, ..., z_0^{(\kappa-1)}, ..., z_{N-1}^{(0)}, ..., z_{N-1}^{(\kappa-1)}\}$ where $z_l^{(k)} \in \{0,1,...,M-1\}$. Next $\mathbf{z}$ is transmitted over a memoryless noisy channel using an M-ary orthogonal PPM TH-IR-based UWB system. The transmitted signal, $x_{Tx}(t)$, for this sequence is defined by

$$x_{Tx}(t) = \sum_{l=0}^{N-1} \sum_{k=0}^{N_w-1} w(t - lT_s - kT_r - \beta_{\sigma_l,k} T_\beta) \tag{4.48}$$

where $w(t)$ has energy E_w and $\beta_{\sigma_l,k} = z_l^{(k)} \in \{0,1,...,M-1\}$. In vector form $x_{Tx}(t)$ can be represented as

$$\mathbf{x} = [\mathbf{x}_0^{(0)}, ..., \mathbf{x}_0^{(\kappa-1)}, ..., \mathbf{x}_{N-1}^{(0)}, ..., \mathbf{x}_{N-1}^{(\kappa-1)}] \tag{4.49}$$

with

$$\mathbf{x}_l^{(k)} = [x_{l,0}^{(k)}, x_{l,1}^{(k)}, ..., x_{l,M-1}^{(k)}] = [0, 0, ..., \underbrace{\sqrt{E_w}}_{\beta_{\sigma_l,k} \text{ Position}}, ..., 0] \tag{4.50}$$

Note $\mathbf{x}_l^{(k)}$ is an M dimensional vector.

The received signal can be described as

$$x_{Rx}(t) = \sum_{l=0}^{N-1} \sum_{k=0}^{N_w-1} w(t - lT_s - kT_r - \beta_{\sigma_l,k} T_\beta + \tau) + n(t) \tag{4.51}$$

where τ is the delay and $n(t)$ is additive noise. For comparison purposes it will be assumed that τ is known at the receiver and that $n(t)$ is additive white Gaussian noise (AWGN) with zero mean and power spectral density $N_0/2$. After signal reception the pulse position demodulator (PPD) produces the received sequence

$$\mathbf{r} = \{\underline{\mathbf{r}_0}, \underline{\mathbf{r}_1}, ..., \underline{\mathbf{r}_{N-1}}\} = \{\mathbf{r}_0^{(0)}, ..., \mathbf{r}_0^{(\kappa-1)}, ..., \mathbf{r}_{N-1}^{(0)}, ..., \mathbf{r}_{N-1}^{(\kappa-1)}\} \tag{4.52}$$

where $\mathbf{r}_l^{(k)}$ is itself a vector. Due to orthogonal signalling is being used, and assuming perfect synchronisation between receiver and transmitter has been achieved (coherent detection), $\mathbf{r}_l^{(k)}$ can be expressed in vector form as ([83, 86]):

$$\mathbf{r}_l^{(k)} = [r_{l,0}^{(k)}, r_{l,1}^{(k)}, ..., r_{l,M-1}^{(k)}] = [n_0, n_1, ..., \sqrt{E_w} + n_{\beta_{\sigma_l,k}}, ..., n_{M-1}] \tag{4.53}$$

where n_0, n_1,..., n_{M-1} are zero mean mutually independent Gaussian random variables with variance $N_0/2$. Note the PPD can be implemented by using a set of M correlators (the actual number can be reduced) such that

$$r_{l,i}^{(k)} = \int_{lT_s+kT_r}^{lT_s+(k+1)T_r} \boldsymbol{x}_{Rx}(t)w(t-lT_s-kT_r-iT_\beta+\tau)dt$$
$$= \int_{lT_s+kT_r}^{lT_s+(k+1)T_r} \{w(t-lT_s-kT_r-\boldsymbol{\beta}_{\sigma_l,k}T_\beta)+n(t)\}w(t-lT_s-kT_r-iT_\beta)dt \quad (4.54)$$

for $i=0,1,...,M-1$.

The received sequence, $\mathbf{r}$, is then used by the Viterbi decoder to produce an estimate, $\mathbf{z}'=\{\underline{\mathbf{z}}_0,\underline{\mathbf{z}}_1,...,\underline{\mathbf{z}}_{N-1}\}=\{z'^{(0)}_0,...,z'^{(\kappa-1)}_0,...,z'^{(0)}_{N-1},...,z'^{(\kappa-1)}_{N-1}\}$, of the transmitted sequence $\mathbf{z}$ and an estimate, $\mathbf{y}'=\{\boldsymbol{y}'_0,\boldsymbol{y}'_1,...,\boldsymbol{y}'_{N-1}\}$, of the information sequence, $\mathbf{y}$. Hard Viterbi decoding (HVD) or soft Viterbi decoding (SVD) can be implemented depending on the way $\mathbf{r}_l^{(k)}$ is fed to the decoder as explained in the following sections.

4.6.1 Bit Error Rate Performance for Hard Viterbi Decoding

In this section the metrics used for hard Viterbi decoding of the FSLS binary to M-ary convolutional encoders are introduced. A union bound on the bit error probability for M-ary Hamming distance based HVD over memoryless AWGN channels is derived. Also examples of BER plots for FSLS binary to 32-ary convolutional encoders are provided.

4.6.1.1 Branch Metrics for Hard Viterbi Decoding

When the PPD makes a hard decision on the received pulse position hard Viterbi decoding must be implemented. In this case $r_l^{(k)}$ is no longer a vector but an M-ary symbol. For the coherent PPD detector under consideration the effect of performing a hard decision before decoding is the transformation of the AWGN channel into an M-ary symmetric discrete channel with transition probabilities ([95])

$$\Pr\{r_l^{(k)} \mid z_l^{(k)}\} = \begin{cases} 1-P_M & r_l^{(k)} = z_l^{(k)} \\ \frac{P_M}{M-1} & r_l^{(k)} \neq z_l^{(k)} \end{cases} \quad (4.55)$$

where P_M is the error probability for M-ary orthogonal signals given by (see Section 2.3)

$$P_s = P_M = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \left[1-\left(\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{y} e^{-x^2/2}dx\right)^{M-1}\right] \exp\left[-\frac{1}{2}\left(y-\sqrt{\frac{2E_w}{N_0}}\right)^{M-1}\right]dy \le (M-1)Q\left(\sqrt{\frac{E_w}{N_0}}\right) \quad (4.56)$$

where $Q(\cdot)$ was defined in Section 2.3.1.

Assume the M-ary Hamming distance between the estimated sequence $\mathbf{z}'$ and the received sequence to be $d_{MHamm}(\mathbf{r},\mathbf{z}')$. Then the conditional probability mass function (p.m.f.), $\Pr\{\mathbf{r}\mid\mathbf{z}'\}$, can be found to be

$$\Pr\{\mathbf{r}\mid\mathbf{z}'\} = (1-P_M)^{N-d_{MHamm}(\mathbf{r},\mathbf{z}')}\left(\frac{P_M}{M-1}\right)^{d_{MHamm}(\mathbf{r},\mathbf{z}')} \quad (4.57)$$

Remember the Viterbi algorithm maximizes the log-likelihood function $\log(g(\mathbf{r}\mid\mathbf{z}'))$ introduced in Section 2.7.5. Therefore substituting Equation (4.57) in $\log(g(\mathbf{r}\mid\mathbf{z}'))$ yields

$$\log(g(\mathbf{r}\mid\mathbf{z}')) = \log(\Pr\{\mathbf{r}\mid\mathbf{z}'\}) = d_{MHamm}(\mathbf{r},\mathbf{z}')\log\frac{P_M}{(M-1)(1-P_m)} + N\log(1-P_M) \quad (4.58)$$

For most cases of interest $P_M < 1/2$, hence maximising the log-likelihood function is equivalent to minimising the M-ary Hamming distance $d_{MHamm}(\mathbf{r},\mathbf{z}')$ since $N\log(1-P_M)$ is a constant for all $\mathbf{z}'$. Therefore the path and branch metrics are given by

$$PM(\mathbf{r}\,|\,\mathbf{z}') = d_{MHamm}(\mathbf{r},\mathbf{z}') = \sum_{i=0}^{N-1} BM(\underline{\mathbf{r}}_i\,|\,\underline{\mathbf{z}}'_i) \text{ and } BM(\underline{\mathbf{r}}_i\,|\,\underline{\mathbf{z}}'_i) = \sum_{j=0}^{\kappa-1} d_{MHamm}(r_i^{(j)}, z_i'^{(j)}) \tag{4.59}$$

where $r_i^{(j)}, z_i'^{(j)} \in \{0,1,...,M-1\}$ and the hard Viterbi decoder (HVD) must find the path, $\mathbf{z}'$, with the smallest M-ary Hamming distance to the received vector $\mathbf{r}$. It is important to note that the metrics defined in Equation (4.59) are different to the hard metrics proposed in [70] for the PCTH scheme which are based on the binary Hamming distance.

Figure 4-14 shows BER plots for PPM TH-IR-based UWB systems using the rate 1 FSLS binary to 32-ary convolutional encoders from Table 4-1 with M-ary Hamming distance based HVD, 32-ary PCTH with binary Hamming distance based HVD, and binary PPM. From Figure 4-14 it can be seen that the system's BER performance can be improved by using the encoders reported in Table 4-1.

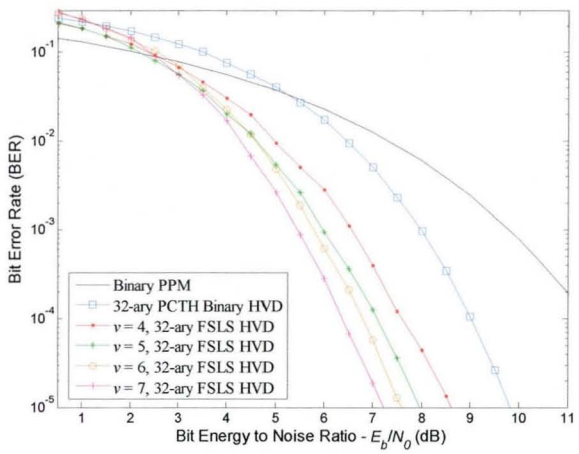


Figure 4-14. Bit error rate versus bit energy to noise ratio for hard Viterbi decoding (HVD) in PPM TH-IR-based UWB systems using the rate 1 FSLS binary to 32-ary convolutional encoders from Table 4-1, 32-ary PCTH, and binary orthogonal PPM. All the plots were obtained by simulation for AWGN channel. The total encoder memory, v, of each code is indicated in the legend. The branch and path metrics used with FSLS encoders were based on the M-ary Hamming distances. For the PCTH scheme HVD was performed by using branch and path metrics based on binary Hamming distances.

4.6.1.2 Upper Bound on the Bit Error Probability for Hard Viterbi Decoding

An upper bound for the bit error probability can be obtained by calculating the pairwise error probability, $P_2(d)$, and substituting it in Equation (4.40). Without loss of generality (due to the linearity property of convolutional codes) assume the transmitted M-ary sequence to be the all zeros sequence and name it $\mathbf{z}_C$. As well consider a symbol sequence, $\mathbf{z}_E$, that diverges from $\mathbf{z}_C$ at some point and later remerges with it after accumulating an M-ary Hamming weight d. Consider the d positions at which $\mathbf{z}_C$ and $\mathbf{z}_E$ differ and label them as $z_{C,i}$ and $z_{E,i}$ with $i = 0,1,...,d-1$. Similarly label the corresponding received symbols at these positions as r_i. Define φ_C to be the number of positions at which $r_i = z_{C,i} \neq z_{E,i}$ and φ_E to be the number of positions at which

$r_i = z_{E,i} \neq z_{C,i}$. As well define $\varphi_I = d - \varphi_C - \varphi_E$, that is the number of positions at which $r_i \neq z_{C,i}, z_{E,i}$. By noting that

$$\Pr\{r_i \mid z_{C,i}\} = \begin{cases} 1 - P_M & r_i = z_{C,i} \\ \frac{P_M}{M-1} & r_i = z_{E,i} \\ \frac{P_M(M-2)}{M-1} & r_i \neq z_{C,i}, z_{E,i} \end{cases} \tag{4.60}$$

we can find the following p.m.f.

$$\Pr\{\boldsymbol{\varphi}_C = \varphi_C, \boldsymbol{\varphi}_E = \varphi_E, \boldsymbol{\varphi}_I = d - \varphi_C - \varphi_E \mid \mathbf{z}_C\} = \frac{d!}{\varphi_C!\varphi_E!(d-\varphi_C-\varphi_E)!}(1-P_M)^{\varphi_C}\left(\frac{P_M}{M-1}\right)^{\varphi_E}\left(\frac{P_M(M-2)}{M-1}\right)^{d-\varphi_C-\varphi_E} \tag{4.61}$$

Due to the fact that hard Viterbi decoding is equivalent to finding the path with the smaller M-ary Hamming distance to the received sequence, $\mathbf{r}$, the pairwise error probability between the sequences $\mathbf{z}_C$ and $\mathbf{z}_E$ can be found as

$$\begin{aligned} P_2(d) = P(\mathbf{z}_C \to \mathbf{z}_E) = \Pr\{d_{MHamm}(\mathbf{r},\mathbf{z}_E) < d_{MHamm}(\mathbf{r},\mathbf{z}_C) \mid \mathbf{z}_C\} + \\ + \tfrac{1}{2}\Pr\{d_{MHamm}(\mathbf{r},\mathbf{z}_E) = d_{MHamm}(\mathbf{r},\mathbf{z}_C) \mid \mathbf{z}_C\} \end{aligned} \tag{4.62}$$

where $P(\mathbf{z}_C \to \mathbf{z}_E)$ stands for the pairwise probability of erroneously selecting $\mathbf{z}_E$ when transmitting $\mathbf{z}_C$. Using $\boldsymbol{\varphi}_C$, $\boldsymbol{\varphi}_E$ and $\boldsymbol{\varphi}_I$ Equation (4.62) becomes

$$P_2(d) = P(\mathbf{z}_C \to \mathbf{z}_E) = \Pr\{\boldsymbol{\varphi}_C < \boldsymbol{\varphi}_E \mid \mathbf{z}_C\} + \tfrac{1}{2}\Pr\{\boldsymbol{\varphi}_C = \boldsymbol{\varphi}_E \mid \mathbf{z}_C\} \tag{4.63}$$

hence

$$\begin{aligned} P_2(d) = P(\mathbf{z}_C \to \mathbf{z}_E) = \sum_{\varphi_E=1}^{d} \sum_{\varphi_C=0}^{\min(\varphi_E-1,\, d-\varphi_E)} \frac{d!}{\varphi_C!\varphi_E!(d-\varphi_C-\varphi_E)!}(1-P_M)^{\varphi_C}\left(\frac{P_M}{M-1}\right)^{\varphi_E}\left(\frac{P_M(M-2)}{M-1}\right)^{d-\varphi_C-\varphi_E} + \\ + \frac{1}{2}\sum_{\varphi_C=\varphi_E=0}^{\lfloor d/2 \rfloor} \frac{d!}{\varphi_C!\varphi_E!(d-\varphi_C-\varphi_E)!}(1-P_M)^{\varphi_C}\left(\frac{P_M}{M-1}\right)^{\varphi_E}\left(\frac{P_M(M-2)}{M-1}\right)^{d-\varphi_C-\varphi_E} \end{aligned} \tag{4.64}$$

By replacing Equation (4.64) in Equation (4.40):

$$P_b \leq \sum_{d=d_{Mfree}}^{\infty} B_d P_2(d) \tag{4.65}$$

and upper bound on the bit error probability for HVD based on the M-ary Hamming distances is obtained. It is worth mentioning that a different looser upper bound for HVD of binary to M-ary encoders is provided in [95]. This bound has the form

$$P_b \leq \sum_{d=d_{Mfree}}^{\infty} B_d \mathcal{D}^d \tag{4.66}$$

where $\mathcal{D}$ is given by

$$\mathcal{D} = \sqrt{\frac{4P_M(1-P_M)}{M-1}} + \frac{M-2}{M-1}P_M \tag{4.67}$$

Figure 4-15 shows plots of the bounds defined by Equations (4.65) and (4.66) compared with simulation results for the rate 1, $v = 6$, FSLS binary to 32-ary encoder. These bounds were calculated by considering the first 20 elements of the IWS (as reference consider that the upper bounds for BER performance of convolutional encodes reported in [135], [110] and [111] were calculated using 8, 13 and 15 elements of the IWS respectively). The plots in Figure 4-15 show that the upper bound "A" defined by Equations (4.64) and (4.65) is tighter than the upper bound "B"

defined by Equations (4.66) and (4.67).

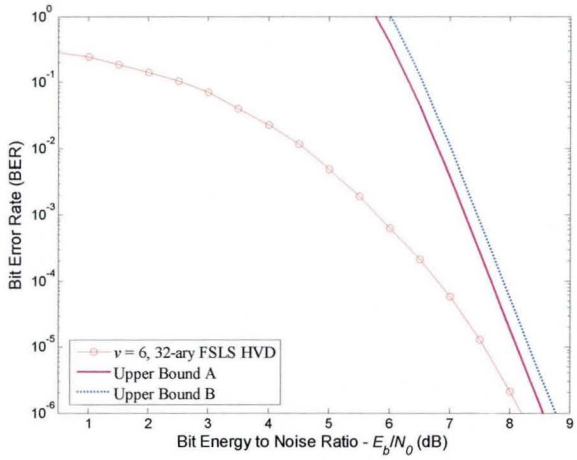


Figure 4-15. Upper bounds on the bit error probability for a PPM TH-IR-based UWB system using the rate 1, $v = 6$, FSLS binary to 32-ary convolutional encoder from Table 4-1 with HVD. The upper bound A was calculated using Equations (4.64) and (4.65) whereas for upper bound B Equations (4.66) and (4.67) were used. The first 20 elements of the IWS were used to calculate both bounds.

4.6.2 Bit Error Rate Performance for Soft Viterbi Decoding

In this section the metrics used for soft Viterbi decoding of the FSLS binary to *M*-ary convolutional encoders are introduced. Also the bit error probability union bound for SVD over memoryless AWGN channels with coherent detection is provided. Examples of BER plots for feedforward spectral line suppressive binary to 32-ary convolutional encoders are presented.

4.6.2.1 Branch Metrics for Soft Viterbi Decoding

When the received vector $\mathbf{r}_l^{(k)}$ is fed to the decoder as a measure of the Euclidean distance between the received signal and each one of the possible *M*-ary orthogonal PPM signals, soft Viterbi decoding can be implemented. Remember from Section 2.7.5 that for soft decoding the Viterbi algorithm maximizes the log-likelihood function

$$\ln(g(\mathbf{r} \mid \mathbf{z}')) = \sum_{i=0}^{N-1} \ln(g(\underline{\mathbf{r}_i} \mid \underline{\mathbf{z}'_i})) = \sum_{i=0}^{N-1} \sum_{j=0}^{\kappa-1} \ln(g(\mathbf{r}_i^{(j)} \mid z_i'^{(j)})) = \sum_{i=0}^{N-1} \sum_{j=0}^{\kappa-1} \ln(g(\mathbf{r}_i^{(j)} \mid \mathbf{x}_i'^{(j)})) \tag{4.68}$$

with

$$\mathbf{x}_i'^{(j)} = [x_{i,0}^{(j)}, x_{i,1}^{(j)}, ..., x_{i,M-1}^{(j)}] = [0, ..., \underbrace{\sqrt{E_w}}_{z_i'^{(j)} \text{ Position}}, ..., 0, 0] \tag{4.69}$$

and where $g(\mathbf{r} \mid \mathbf{z}')$ is the conditional probability density function of $\mathbf{r}$ given $\mathbf{z}'$. For *M*-ary orthogonal signalling we have ([83, 127])

$$g(\mathbf{r}_i^{(j)} \mid \mathbf{x}_i'^{(j)}) = \frac{1}{(\pi N_0)^{M/2}} \exp\left(-\sum_{\varsigma=0}^{M-1} \frac{(r_{i,\varsigma}^{(j)} - x_{i,\varsigma}'^{(j)})^2}{N_0}\right) \tag{4.70}$$

and therefore

$$\ln(g(\mathbf{r} \mid \mathbf{z}')) = \sum_{i=0}^{N-1} \sum_{j=0}^{\kappa-1} \left\{-\ln\left((\pi N_0)^{M/2}\right) - \sum_{\varsigma=0}^{M-1} \frac{(r_{i,\varsigma}^{(j)} - x_{i,\varsigma}'^{(j)})^2}{N_0}\right\} \tag{4.71}$$

Maximising Equation (4.71) is equivalent to maximise

$$PM(\mathbf{r} \mid \mathbf{z}') = \sum_{i=0}^{N-1} BM(\underline{\mathbf{r}}_i \mid \underline{\mathbf{z}}'_i) = \sum_{i=0}^{N-1} \sum_{j=0}^{\kappa-1} \sum_{\varsigma=0}^{M-1} r_{i,\varsigma}^{(j)} x'^{(j)}_{i,\varsigma} = \sum_{i=0}^{N-1} \sum_{j=0}^{\kappa-1} \mathbf{r}_i^{(j)} \cdot \mathbf{x}'^{(j)}_i \tag{4.72}$$

where terms common to all paths have been discarded. Due to $x'^{(j)}_{i,\varsigma} = 0$ for $\varsigma \neq z'^{(j)}_i$ we have

$$\mathbf{r}_i^{(j)} \cdot \mathbf{x}'^{(j)}_i = \begin{cases} \sqrt{E_w}(\sqrt{E_w} + n_{\beta_{\sigma_l},k}) & z'^{(j)}_i = \beta_{\sigma_l,k} \\ \sqrt{E_w} n_{z'^{(j)}_i} & z'^{(j)}_i \neq \beta_{\sigma_l,k} \end{cases} \tag{4.73}$$

and therefore the use of $\mathbf{r}_l^{(k)}$ to form the path and branch metrics is straightforward. The BER plots obtained when implementing SVD by using these metrics are shown in Figure 4-16.

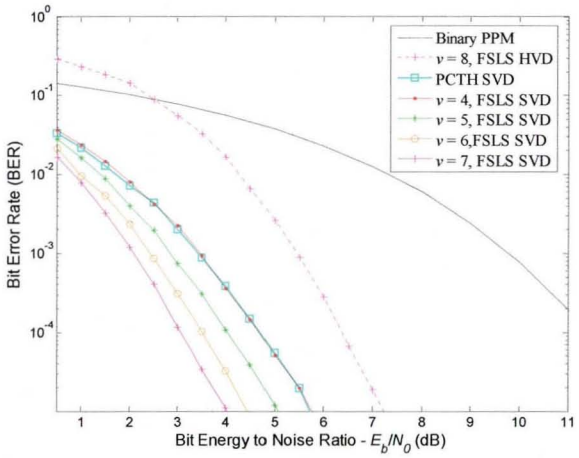


Figure 4-16. Bit error rate versus bit energy to noise ratio for soft Viterbi decoding (SVD) in PPM TH-IR-based UWB systems using the rate 1 FSLS binary to 32-ary convolutional encoders presented in Table 4-1 and 32-ary PCTH. The BER plots for rate 1, $v = 6$, FSLS binary to 32-ary convolutional encoders with HVD and binary orthogonal PPM are provided as a reference. All the plots were obtained by simulation for the AWGN channel. The total encoder memory, v, of each code is indicated in the legend.

The BER plots in Figure 4-16 are for PPM TH-IR-based UWB systems using the rate 1 FSLS binary to 32-ary convolutional encoders from Table 4-1 with SVD, 32-ary PCTH with SVD, and binary PPM. From this figure it is clear that the BER is further improved when using SVD compared to HVD as expected.

Finally note that maximising Equations (4.71) and (4.72) is equivalent to finding the path/codeword, $\mathbf{z}'$, which minimises the squared Euclidean distance:

$$d_{Euc}(\mathbf{r}, \mathbf{x}')^2 = \sum_{i=0}^{N-1} \sum_{j=0}^{\kappa-1} \sum_{\varsigma=0}^{M-1} (r_{i,\varsigma}^{(j)} - x'^{(j)}_{i,\varsigma})^2 \tag{4.74}$$

4.6.2.2 Upper Bound on the Bit Error Probability for Soft Viterbi Decoding

As in the previous case, the pairwise error probability, $P_2(n)$, must be calculated in order to obtain an upper bound on the bit error probability for SVD. Assume again that the all zeros sequence, $\mathbf{z}_C$, is transmitted and name its respective signal sequence as $\mathbf{x}_C$. Now assume the symbol sequence, $\mathbf{z}_E$, with respective signal sequence, $\mathbf{x}_E$, diverges from $\mathbf{z}_C$ at some point and later remerges with it after accumulating an M-ary Hamming weight d. Denote $d_{Euc}(\mathbf{x}_C, \mathbf{x}_E)$ the Euclidean distance between the signal sequences $\mathbf{x}_C$ and $\mathbf{x}_E$. As the channel is assumed to be AWGN with two sided power spectral density $N_0/2$ and soft Viterbi decoding is equivalent to

finding the signal sequence with the smaller Euclidean distance to the received sequence, **r**, the pairwise error probability between the sequences $\mathbf{x}_C$ and $\mathbf{x}_E$ is given by ([83, 101])

$$P_2(d) = P(\mathbf{x}_C \to \mathbf{x}_E) = Q\left(\sqrt{\frac{(d_{Euc}(\mathbf{x}_C, \mathbf{x}_E))^2}{2N_0}}\right) \tag{4.75}$$

where $P(\mathbf{x}_C \to \mathbf{x}_E)$ stands for the pairwise probability of erroneously selecting $\mathbf{x}_E$ when transmitting $\mathbf{x}_C$. Because of the symmetry of the orthogonal signal set:

$$(d_{Euc}(\mathbf{x}_C, \mathbf{x}_E))^2 = \left((\sqrt{E_w} - 0)^2 + (0 - \sqrt{E_w})^2\right)d = 2E_w d \tag{4.76}$$

and therefore

$$P_2(d) = Q\left(\sqrt{\frac{E_w}{N_0} d}\right) = Q\left(\sqrt{\frac{E_b}{\kappa N_0} d}\right) \tag{4.77}$$

where E_b is the information bit energy and κ is the number of M-ary symbols produced per data bit.

Therefore for SVD the probability of error of a binary to M-ary convolutional code is upper bounded by

$$P_b \le \sum_{d=d_{Mfree}}^{\infty} B_d P_2(d) = \sum_{d=d_{Mfree}}^{\infty} B_d Q\left(\sqrt{\frac{E_b}{\kappa N_0} d}\right) \tag{4.78}$$

In Figure 4-17 the plots of the bound defined by Equation (4.78) compared to simulation results for the rate 1, $v = 6$, FSLS binary to 32-ary encoder are shown. The upper bound was obtained by considering the first 20 element of the IWS whereas for the lower bound only the first element was used. The plots in Figure 4-17 show there is a good agreement between the bounds and the simulated results.

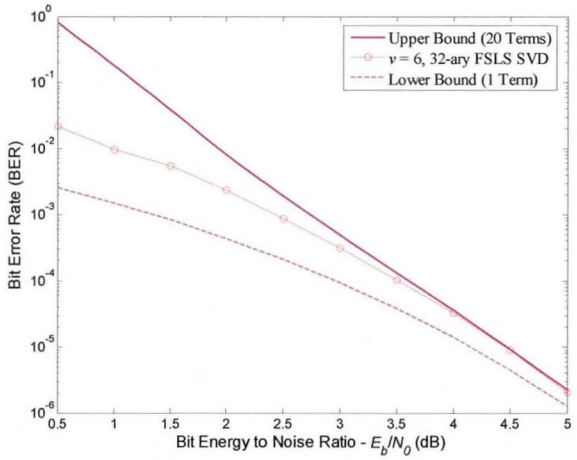


Figure 4-17. Upper and lower bounds on the bit error probability for SVD in a PPM TH-IR-based UWB system employing the rate 1, $v = 6$, FSLS binary to 32-ary convolutional encoders presented in Table 4-1. For the upper bound the first 20 elements of the IWS were used in Equation (4.78) whereas for the lower bound only the first element of the IWS was considered.

4.7 Conclusions for Chapter 4

In this chapter feedforward spectral line suppressive (FSLS) binary to M-ary convolutional encoders have been introduced. First, it was shown that the PCTH scheme can be interpreted as a binary to M-ary convolutionally coded system using M-ary orthogonal PPM signalling. Therefore it

was concluded that there existed the possibility to find good binary to M-ary convolutional encoders with similar PSD characteristics and better BER than the PCTH encoders. Based on this premise the condition needed for a binary to M-ary convolutional encoder to eliminate as many spectral lines as the PCTH scheme was obtained through analysis. It has also been shown that binary to M-ary convolutional encoders complying with the spectral line suppression condition can eliminate as many spectral lines as a TH-IR-based UWB system employing binary PPM and perfectly random M/2-ary TH.

Based on the spectral line suppression analysis a code search procedure for rate 1, 1/2 and 1/3 FSLS binary to M-ary convolutional encoders with the best possible information weight spectrum (IWS) was introduced. The code search procedure verifies that the spectral line suppression condition is fulfilled before comparing the IWS of the encoder under test. Using this code search procedure maximum free distance FSLS binary to M-ary convolutional encoders with the best possible IWS were found. A table of the best FSLS convolutional encoders for 16-ary, 32-ary, 64-ary and 128-ary orthogonal signalling has been provided. These encoders are able to eliminate as many spectral lines as the M-ary PCTH scheme and binary PPM with perfectly random M/2-ary TH under the same operation constraints (uniform distributed i.i.d. binary inputs).

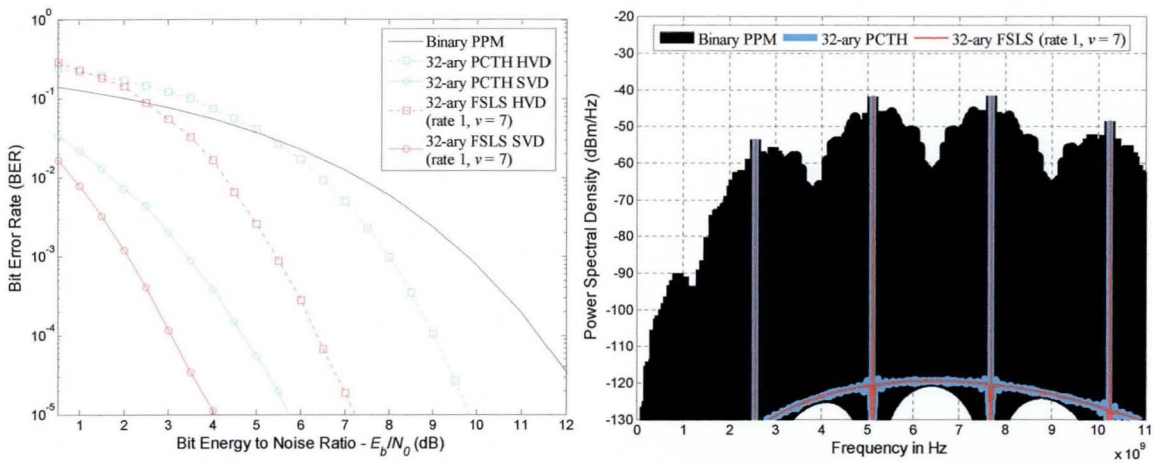
Power spectral density examples of PPM TH-IR-based UWB signals using FSLS binary to 32-ary convolutional encoders, 32-ary PCTH and binary PPM with perfectly random and pseudo-random (PR) 16-ary TH have been provided. These examples show that all these systems have the same spectral line suppression capabilities with the exception of the system employing binary PPM with periodic PR-TH.

Lastly the bit error rate performance of the proposed convolutionally coded system has been addressed. A derivation of the branch and path metrics used for hard and soft Viterbi decoding has been provided. Upper bounds on the bit error probability for hard and soft Viterbi decoding have been introduced and BER plots examples have been given for 32-ary encoders. The BER plots examples showed that improved BER performance over PCTH and binary PPM can be achieved by using the FSLS binary to M-ary convolutional encoders introduced in Table 4-1 for both HVD and SVD. Furthermore, by using the M-ary Hamming distance to compute the branch and path metrics needed for HVD a significant improvement on the BER performance over PCTH and binary PPM can be achieved with the FSLS binary to M-ary convolutional encoders. This characteristic is important if such encoders are intended to be used in low rate low complexity systems where soft Viterbi decoding may not be feasible. It is worth mentioning that the use of the FSLS binary to M-ary convolutional encoders is not restricted to orthogonal M-ary PPM as they can be used in any system employing M-ary orthogonal signalling.

It is clear that the use of binary to M-ary convolutional encoders in TH-IR-based UWB systems may increase the system's complexity compared with non-coded binary PPM TH-IR-based UWB systems. However, in order to achieve similar PSD characteristics to those obtained with the PCTH and the FSLS binary to M-ary convolutionally coded schemes, the non-coded system must employ

a perfectly random (ideal) TH sequence. In practice the generation of this ideal TH sequence may be approximated by generating a PR-TH sequence with extremely long period which would increase the system's complexity as well (note this PR-TH sequence must be generated in both the transmitter and the receiver in order to be able to recover the original information). In any case the complexity difference between the binary to M-ary convolutionally coded scheme and the non-coded scheme would reside in the receiver side and particularly in the implementation of the Viterbi decoder. Nevertheless, this possible increase in complexity is compensated by the improved BER performance achieved when using the binary to M-ary convolutional encoders as shown in Section 4.6.

It is worth mentioning that the decoding complexity of the FSLS binary to M-ary convolutional encoders introduced in this chapter can be regarded to be low to medium as their total encoder memory ranges (a measure of decoding complexity, [100, 101, 104]) from 3 to 7 and the industry de facto standard binary convolutional encoder has total encoder memory equal to 6, [104]. As well, the decoding complexity of the encoders reported in Table 4-1 can be equal to or larger than the decoding complexity of the PCTH encoders, as several FSLS binary to M-ary convolutional encoders with different total encoder memory were reported for the same M-ary alphabet. However, this increase in complexity is compensated by the improved BER performance achieved when increasing the total encoder memory as evidenced by the results presented in Section 4.6.



a) BER performance comparison b) PSD shape comparison

Figure 4-18. Side by side BER performance and PSD shape (analytical) comparisons between PPM TH-IR-based UWB systems employing: the rate 1, $v = 7$, binary to 32-ary FSLS convolutional encoder from Table 4-1, the 32-ary Bernoulli shift map PCTH scheme and non-coded orthogonal binary PPM with pseudo-random 16-ary TH. The input data stream is assumed to consist on i.i.d. binary symbols with uniform distribution. The periodic sequence {0, 14, 1, 5, 13, 6, 3, 15, 7, 11, 8, 12, 9, 2, 10, 4} was used for TH in the system employing non-coded orthogonal binary PPM. The BER performance plots were obtained by simulation for soft Viterbi decoding (SVD) and hard Viterbi decoding (HVD) in the AWGN channel. For the PCTH scheme HVD was performed by using branch and path metrics based on binary Hamming distances. The signal parameters are $N_w = 1$, $T_s = 12.5$ ns, $T_r = 12.5$ ns and $T_\beta = T_r/32$ ($T_c = T_r/16$ for the system employing pseudo-random 16-ary TH). The 3rd derivative Gaussian pulse was used with duration $T_w \approx 0.35$ ns.

Chapter 5

Maximum Free Distance Spectral Line Suppressive Binary to *M*-ary Convolutional Encoders for Unbalanced Binary Markov Sources

5.1 Introduction

In this chapter novel maximum free distance (MFD) recursive spectral line suppressive (SLS) binary to *M*-ary convolutional encoders for pulse position modulated (PPM) time hopping (TH) impulse radio (IR) based ultra wideband (UWB) systems with first order binary Markov sources (BMS) are introduced. These convolutional encoders eliminate as many spectral lines as the feedforward SLS (FSLS) convolutional encoders introduced in the previous chapter and the pseudo-chaotic time hopping (PCTH) scheme *even when the binary stream at the encoder input does not consist of independent identically distributed (i.i.d.) symbols with uniform distribution.* In particular it is assumed that the data stream is generated by an unbalanced (that is non-uniform distributed) binary Markov source.

The main characteristic of the SLS convolutional encoders presented in this chapter is the introduction of feedback, which is novel for binary to *M*-ary encoders. It is shown that through the introduction of recursive structures the spectral line suppression condition introduced in Section 4.3.2 can be satisfied even when the data stream is generated by an unbalanced first order BMS.

Good encoders with rates 1, 1/2 and 1/3 for 16-ary, 32-ary, 64-ary and 128-ary orthogonal modulation schemes (not restricted to be PPM based) are reported. It is shown that these recursive SLS (RSLS) convolutional encoders provide enhanced power spectral density (PSD) characteristics compared to the FSLS convolutional encoders with a slight degradation on the system's bit error rate (BER) performance.

One of the main limitations of the PCTH scheme and the FSLS binary to *M*-ary convolutional encoders introduced in the previous section is that they strictly require the encoder input to consist of uniform distributed i.i.d binary streams in order to satisfy the spectral line suppression condition. As explained in Sections 2.5 and 4.2 in [70, 33, 37] it is assumed that these uniform distributed

i.i.d. binary streams can be obtained through a compression and scrambling operation over the original data stream generated by the source. Nevertheless, in order to guarantee this constraint is fulfilled, the compression algorithm must be optimal meaning that it should be able to completely eliminate all the redundancy on the source's data stream, [136]. However achieving this in practice may prove difficult excluding a few specific cases. Furthermore, most real world source encoders are suboptimal or asymptotically optimal and therefore some residual redundancy, in the form of non-uniformity and/or memory, is usually left in the data stream after compression, [117, 137-141]. Therefore it can be asserted that for many cases of interest the assumption of uniform distributed i.i.d. binary streams at the encoder input can not be sustained, even with the inclusion of compression and scrambling operations.

As previously mentioned, the source model adopted in this chapter is the first order BMS introduced in Section 3.2. This is due to the first order BMS enables the modelling of unbalanced i.i.d. memoryless binary data streams as well as binary data streams with memory, [116-118]. Note these kinds of data streams can be generated by the combination of data sources with suboptimal source encoders (e.g. suboptimal compression algorithms) or by stand alone data sources. Therefore, the SLS convolutional encoders introduced in this chapter have a much wider applicability range than the PCTH scheme and the FSLS encoders introduced in Chapter 4.

This chapter is structured as follows. Section 5.2 describes the system model. Section 5.3 shows that a recursive structure for binary to *M*-ary convolutional encoders is advantageous for the design of SLS encoders with unbalanced first order BMS inputs. A modification of the code search procedure introduced in Section 4.4 is presented in Section 5.4 altogether with the novel recursive SLS (RSLS) binary to *M*-ary convolutional encoders found with such procedure. Comparisons in terms of PSD characteristics and BER performance between the RSLS, FSLS and PCTH encoders are provided in Section 5.5. Lastly conclusions are presented in Section 5.6.

5.2 System Model for Binary to *M*-ary Convolutionally Coded TH-IR-Based UWB Signals with First Order Binary Markov Sources

The block diagram of the system model assumed in this chapter is shown in Figure 5-1. This model is similar to the one introduced in Section 4.2 with the difference that this time the data source is assumed to be the first order binary Markov Source (BMS) introduced in Section 3.2.1.

Remember this BMS source model is defined by the state set $\mathcal{R}=\{r_0,r_1\}$ with corresponding outputs $\{t_0,t_1\}=\{0,1\}$ and the one step transition probabilities matrix defined by

$$\mathbf{P}_y=\begin{bmatrix} p_{y,00} & p_{y,01} \\ p_{y,10} & p_{y,11} \end{bmatrix} \tag{5.1}$$

where

$$p_{y,ij} = \Pr\{\rho_l = r_j \mid \rho_{l-1} = r_i\} = \Pr\{y_l = \iota_j \mid y_{l-1} = \iota_i\} = \Pr\{y_l = j \mid y_{l-1} = i\} \tag{5.2}$$

with

$$p_{y,00} = 1 - p_{y,01} \quad \text{and} \quad p_{y,11} = 1 - p_{y,10} \tag{5.3}$$

The corresponding steady state probabilities for this Markov chain (MC) are then given by

$$\underline{\pi_y} = [\pi_{y,0} \;\; \pi_{y,1}] = \left[\frac{p_{y,10}}{p_{y,01}+p_{y,10}} \;\; \frac{p_{y,01}}{p_{y,01}+p_{y,10}}\right] \tag{5.4}$$

where $\pi_{y,0}$ and $\pi_{y,1}$ are the steady state probabilities of generating a "0" and a "1" respectively. In order to avoid trivial degenerate cases such as $p_{y,ij} = 1$, it will be assumed that

$$0 < p_{y,ij} < 1 \tag{5.5}$$

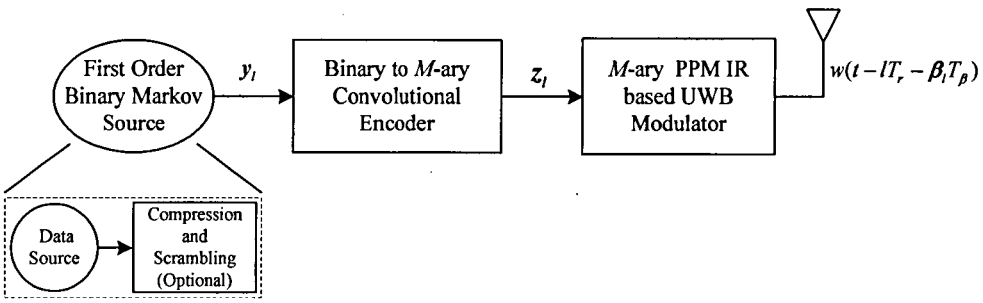


Figure 5-1. Block diagram of the binary to M-ary convolutionally coded M-ary PPM TH-IR-based UWB system with first order binary Markov source (BMS).

In Section 3.4 it was shown that the PSD's discrete part of a convolutionally coded – Markov driven IR-based UWB signal is a function of the steady state probabilities of the source encoder (SE) pair Markov model. Therefore it is pertinent to analyse the SE pair Markov model steady state behaviour when using feedforward and feedback structures to construct binary to M-ary convolutional encoders. This analysis will be provided in the next sections where, as in the previous chapter, it will be assumed that M is a power of 2.

Before starting the analysis it is convenient to remember from Section 3.2.3 that the state process, σ_l, defined by the SE pair Markov model has transition probabilities matrix given by

$$\mathbf{P}_\sigma = \begin{bmatrix} p_{y,00}\mathbf{B}_0 & p_{y,01}\mathbf{B}_0 \\ p_{y,10}\mathbf{B}_1 & p_{y,11}\mathbf{B}_1 \end{bmatrix} \tag{5.6}$$

where $\mathbf{B}_0$ and $\mathbf{B}_1$ are the $N_q \times N_q$ the next state matrices defined in Section 3.2.3 and $N_q = 2^v$ is the number of encoder states, q_i, as defined by the encoder's state transition diagram representation (see Sections 2.7.2 and 3.2.2). Recall the state set for the SE pair Markov model is defined as

$$\mathcal{S} = \{(0,q_0),(0,q_1),...,(0,q_{Nq-1}),(1,q_0),(1,q_1),...,(1,q_{Nq-1})\} = \{s_0, s_1, ..., s_{2Nq-1}\} \tag{5.7}$$

with corresponding output mapping

$$\underline{\zeta_{s_{nN_q+j}}} = [\zeta^{(0)}_{s_{nN_q+j}}, \zeta^{(1)}_{s_{nN_q+j}}, \ldots, \zeta^{(\kappa-1)}_{s_{nN_q+j}}] = \gamma(s_{nN_q+j}) = \gamma(n, q_j) \tag{5.8}$$

where $\gamma(\cdot)$ is the output mapping function, $n = 0,1$ and $j = 0,1,...,N_q - 1$.

5.2.1 Source-Encoder pair Markov Model Analysis for Feedforward Convolutional Encoders

Consider the feedforward structure for binary to M-ary convolutional encoders introduced in Section 4.4.2 – Figure 4-4 and reproduced for convenience in Figure 5-2.

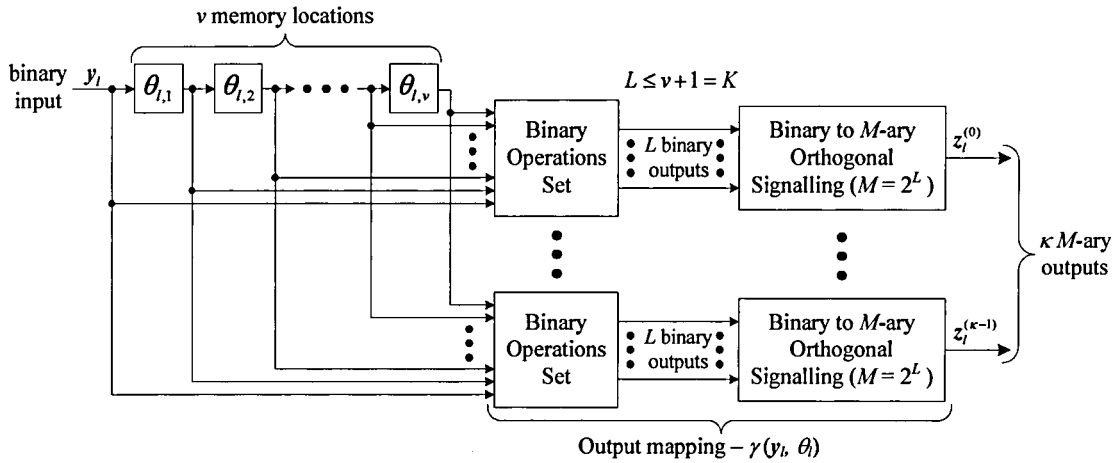


Figure 5-2. Generic diagram of a rate $1/\kappa$ feedforward binary to M-ary convolutional encoder.

Recall from Section 4.2.2 that for feedforward convolutional encoders each element $B_{n,jj'}$ of the $N_q \times N_q$ the next state matrices, $\mathbf{B}_n = [B_{n,jj'}]$, is equal to

$$B_{0,jj'} = \begin{cases} 0, & j' \neq \lfloor j/2 \rfloor \\ 1, & j' = \lfloor j/2 \rfloor \end{cases} \quad \text{and} \quad B_{1,ij} = \begin{cases} 0, & j' \neq 2^{v-1} + \lfloor j/2 \rfloor \\ 1, & j' = 2^{v-1} + \lfloor j/2 \rfloor \end{cases} \tag{5.9}$$

where $\lfloor \cdot \rfloor$ stands for the floor function. Since the BMS' one step transition probabilities $p_{y,00}$ are not restricted to be 1/2 as in the previous chapter (see Sections 4.1 and 4.2), the entries in each column of the transition probabilities matrix, $\mathbf{P}_\sigma$, do not necessarily sum to unity. Thus, it can not be guaranteed that $\mathbf{P}_\sigma$'s steady state probabilities are all equal to $1/(2N_q)$ as in Section 4.2.

Before providing a closed form expression for $\mathbf{P}_\sigma$'s steady state probabilities it is necessary to analyse the state transitions defined by the SE pair Markov model. Assume the state of the SE pair MC process at time l to be $\sigma_l = (\rho_l, \theta_l) = (y_l, \theta_l) = (n', q_{j'}) = s_{n'N_q + j'}$ with corresponding encoder memory content $[\theta_{l,1}, \theta_{l,2}, ..., \theta_{l,v}] = [\mu'_1, \mu'_2, ..., \mu'_v]$, where $j' = \mu'_1 2^{v-1} + \mu'_2 2^{v-2} + ... + \mu'_v$ with $\mu'_i \in \{0,1\}$ as in the previous chapters. Remember from Section 3.2.3 that this SE pair MC state can only be reached from a previous SE pair MC state $\sigma_{l-1} = (\rho_{l-1}, \theta_{l-1}) = (y_{l-1}, \theta_{l-1}) = (n, q_j) = s_{nN_q + j}$ if

$$q_{j'} = \phi(n, q_j) \quad \text{and} \quad p_{y,nn'} = \Pr\{y_l = n' \mid y_{l-1} = n\} \neq 0 \tag{5.10}$$

where $\phi(\cdot)$ is the next state function of the encoder's Melay finite state sequential machine (FSSM) model equivalent to the encoder's state transition diagram. From Figure 5-2 it can be seen that the

encoder state $q_{j'}$ can only be reached from two previous encoder states q_{k+0} and q_{k+1} with respective memory content at time $l-1$ given by

$$[\theta_{l-1,1},\theta_{l-1,2},...,\theta_{l-1,v}]=[\mu'_2,...,\mu'_v,0] \text{ and } [\theta_{l-1,1},\theta_{l-1,2},...,\theta_{l-1,v}]=[\mu'_2,...,\mu'_v,1] \tag{5.11}$$

where $k=\mu'_2 2^{v-1}+...+\mu'_v 2=(2j' \bmod 2^v)$ and mod stands for the modulus operation. Furthermore, in order to reach $q_{j'}$ from any of these states it is needed that

$$y_{l-1}=\mu'_1 \tag{5.12}$$

Thus the SE pair MC state $(n',q_{j'})=s_{n'N_q+j'}$ can only be accessed from the states

$$(\mu'_1,q_k)=s_{\mu'_1 N_q+k} \text{ and } (\mu'_1,q_{k+1})=s_{\mu'_1 N_q+k+1} \tag{5.13}$$

As a consequence the $i'=(n'N_q+j')^{\text{th}}$ column corresponding to state $(n',q_{j'})=s_{n'N_q+j'}$ in the one step transition probabilities matrix $\mathbf{P}_\sigma$ has row entries equal to

$$p_{\sigma,ii'}=\begin{cases} p_{y,\mu'_1 n'}, & i=\mu'_1 N_q+k,\mu'_1 N_q+k+1 \\ 0, & \text{otherwise} \end{cases} \tag{5.14}$$

where $k=(2j' \bmod 2^v)$ and $i=nN_q+j$ is the row index corresponding to state $(n,q_j)=s_{nN_q+j}$ with $n=0,1$ and $j=0,1,...,N_q-1$. This property will now be used to prove the following theorem.

Theorem 5.1: *Consider a feedforward convolutional encoder with first order BMS input defined by Equations* (5.1) *to* (5.5). *Then the steady state probabilities of the SE pair Markov model state process,* σ_l, *with one step transition probabilities matrix,* $\mathbf{P}_\sigma$, *and state set,* $\mathcal{S}$, *as defined by Equations* (5.6) *and* (5.7) *respectively, are given by*

$$\pi_{\sigma,nN_q+j}=\pi_{\sigma,(n,q_j)}=\pi_{\sigma,s_{nN_q+j}}=p_{y,\mu_1 n}p_{y,\mu_2\mu_1}\cdots p_{y,\mu_{v-1}\mu_{v-2}}p_{y,\mu_v\mu_{v-1}}\pi_{y,\mu_v} \tag{5.15}$$

where $n=0,1$, $\pi_{y,t}$ *and* $p_{y,tt'}$ *are BMS' steady state and one step transition probabilities, and* $[\mu_1,\mu_2,...,\mu_v]$ *is the encoder's memory content defining the encoder state* q_j *with* $j=\mu_1 2^{v-1}+...+\mu_v$.

Proof: Remember from Section 3.2 that the steady state probabilities satisfy ([119, 120])

$$\underline{\pi_\sigma}=\underline{\pi_\sigma}\mathbf{P}_\sigma \tag{5.16}$$

or equivalently ([122])

$$\pi_{\sigma,i'}=\sum_{i=0}^{2N_q-1}\pi_{\sigma,i}p_{\sigma,ii'} \tag{5.17}$$

where $\underline{\pi_\sigma}=[\pi_{\sigma,0},\pi_{\sigma,1},...,\pi_{\sigma,2N_q-1}]$. Substituting Equation (5.14) in Equation (5.17) yields

$$\pi_{\sigma,i'}=\pi_{\sigma,n'N_q+j'}=\pi_{\sigma,\mu'_1 N_q+k}p_{y,\mu'_1 n'}+\pi_{\sigma,\mu'_1 N_q+k+1}p_{y,\mu'_1 n'} \tag{5.18}$$

where $i'=n'N_q+j'=n'N_q+\mu'_1 2^{v-1}+...+\mu'_v$ and $k=\mu'_2 2^{v-1}+...+\mu'_v 2=(2j' \bmod 2^v)$. Thus, in order to proof Theorem 5.1 it is necessary to demonstrate that Equation (5.18) holds when replacing $\pi_{\sigma,i'}$ by Equation (5.15), that is

$$p_{y,\mu'_1 n} p_{y,\mu'_2 \mu'_1} \cdots p_{y,\mu'_{v-1}\mu'_{v-2}} p_{y,\mu'_v \mu'_{v-1}} \pi_{y,\mu'_v} = \pi_{\sigma,i'} = \pi_{\sigma,n' N_q + j'} = \pi_{\sigma,\mu'_1 N_q + k} p_{y,\mu'_1 n'} + \pi_{\sigma,\mu'_1 N_q + k+1} p_{y,\mu'_1 n'} \quad (5.19)$$

Substituting Equation (5.15) in the right side of Equation (5.18) gives

$$\pi_{\sigma,\mu'_1 N_q + k} p_{y,\mu'_1 n'} + \pi_{\sigma,\mu'_1 N_q + k+1} p_{y,\mu'_1 n'} = p_{y,\mu'_2 \mu'_1} p_{y,\mu'_3 \mu'_2} \cdots p_{y,\mu'_v \mu'_{v-1}} (p_{y,0\mu'_v} \pi_{y,0} + p_{y,1\mu'_v} \pi_{y,1}) p_{y,\mu'_1 n'} \quad (5.20)$$

However note that

$$p_{y,0\mu'_v} \pi_{y,0} + p_{y,1\mu'_v} \pi_{y,1} = p_{y,0\mu'_v} \frac{p_{y,10}}{p_{y,01}+p_{y,10}} + p_{y,1\mu'_v} \frac{p_{y,01}}{p_{y,01}+p_{y,10}} = \begin{cases} \frac{p_{y,10}(p_{y,00}+p_{y,01})}{p_{y,01}+p_{y,10}} = \pi_{y,0}, & \mu'_v = 0 \\ \frac{p_{y,01}(p_{y,10}+p_{y,11})}{p_{y,01}+p_{y,10}} = \pi_{y,1}, & \mu'_v = 1 \end{cases} \quad (5.21)$$

where Equation (5.4) was used. Replacing Equation (5.21) into Equation (5.20) yields

$$\pi_{\sigma,\mu'_1 N_q + k} p_{y,\mu'_1 n'} + \pi_{\sigma,\mu'_1 N_q + k+1} p_{y,\mu'_1 n'} = p_{y,\mu'_1 n'} p_{y,\mu'_2 \mu'_1} p_{y,\mu'_3 \mu'_2} \cdots p_{y,\mu'_v \mu'_{v-1}} \pi_{y,\mu'_v} \quad (5.22)$$

which is identical to the leftmost side of Equation (5.19). Therefore for feedforward convolutional encoders the steady state probabilities of the SE pair Markov model state process, $\boldsymbol{\sigma}_l$, are given by Equation (5.15).

It is worth mentioning that a distribution equivalent to Equation (5.15) has been reported in [142] where the correlation of the output of feedforward binary convolutional encoders is analysed. Nevertheless in [142] no formal proof of Equation (5.15) is provided and the analysis is performed in a different way.

5.2.2 Source-Encoder pair Markov Model Analysis for Recursive Convolutional Encoders

Now consider the recursive structure for binary to M-ary convolutional encoders shown in Figure 5-3. For reasons that will become clear in the following analysis, this work focuses on structures with $b_v = 1$ while the rest of the feedback coefficients, b_i, can be freely set to be 0 or 1.

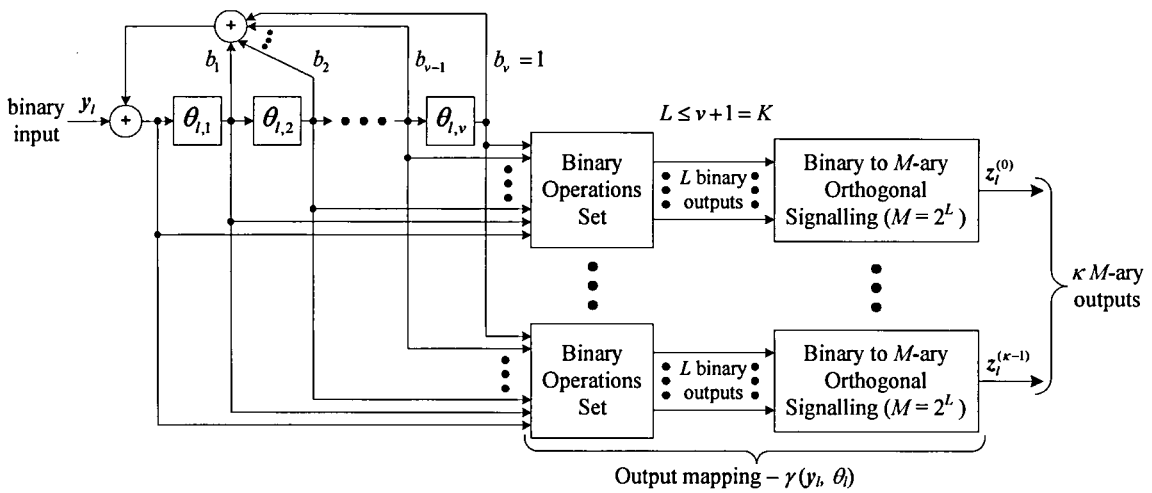


Figure 5-3. Generic diagram of a rate $1/\kappa$ recursive binary to M-ary convolutional encoder.

The transition probabilities matrix of the state process, $\boldsymbol{\sigma}_l$, defined by this encoder's SE pair Markov model can be found by using Equation (5.6). However this time the $N_q \times N_q$ next state matrices, $\mathbf{B}_0$ and $\mathbf{B}_1$, acquire a completely different form. In order find $\mathbf{B}_0$ and $\mathbf{B}_1$ consider the

encoder memory content at time l to be $[\theta_{l,1},\theta_{l,2},...,\theta_{l,v}]=[\mu_1,\mu_2,...,\mu_v]$ with corresponding state label q_j where $j=\mu_1 2^{v-1}+\mu_2 2^{v-2}+...+\mu_v$. Two different cases regarding the encoder's feedback structure and the current memory content can now be identified namely:

$$\text{(1). } b_1\mu_1 \oplus b_2\mu_2 \oplus ... \oplus b_{v-1}\mu_{v-1}=0 \text{ and (2). } b_1\mu_1 \oplus b_2\mu_2 \oplus ... \oplus b_{v-1}\mu_{v-1}=1 \tag{5.23}$$

where $\oplus$ stands for modulo 2 addition (that is over the binary Galois field – GF(2)). Let us focus first on case with $b_1\mu_1 \oplus ... \oplus b_{v-1}\mu_{v-1}=0$. When the current input is $y_l=0$ and $\mu_v=0$ (that is j is even) the next state is $[\theta_{l+1,1},\theta_{l+1,2},...,\theta_{l+1,v}]=[0,\mu_1,\mu_2,...,\mu_{v-1}]$ due to $y_l \oplus b_1\mu_1 \oplus ... \oplus b_{v-1}\mu_{v-1} \oplus b_v\mu_v=0$. In contrast, when $\mu_v=1$ (that is j is odd) the next state is $[\theta_{l+1,1},\theta_{l+1,2},...,\theta_{l+1,v}]=[1,\mu_1,\mu_2,...,\mu_{v-1}]$ since $y_l \oplus b_1\mu_1 \oplus ... \oplus b_{v-1}\mu_{v-1} \oplus b_v\mu_v=1$. Therefore using Section 3.2.3 notation

$$\theta_{l+1}=\phi(y_l,\theta_l)=\phi(0,q_j)=\begin{cases} q_{j/2} & j \text{ even} \\ q_{2^{v-1}+\lfloor j/2 \rfloor} & j \text{ odd} \end{cases} \tag{5.24}$$

for case (1) with $b_1\mu_1 \oplus ... \oplus b_{v-1}\mu_{v-1}=0$. Using a similar analysis when the current input is $y_l=1$ we can find

$$\theta_{l+1}=\phi(y_l,\theta_l)=\phi(1,q_j)=\begin{cases} q_{2^{v-1}+j/2} & j \text{ even} \\ q_{\lfloor j/2 \rfloor} & j \text{ odd} \end{cases} \tag{5.25}$$

for case (1).

Consider now case (2) with $b_1\mu_1 \oplus ... \oplus b_{v-1}\mu_{v-1}=1$. When the current input is $y_l=0$ and j is even (that is $\mu_v=0$) the next state is $[\theta_{l+1,1},\theta_{l+1,2},...,\theta_{l+1,v}]=[1,\mu_1,\mu_2,...,\mu_{v-1}]$ due to $y_l \oplus b_1\mu_1 \oplus ... \oplus b_{v-1}\mu_{v-1} \oplus b_v\mu_v=1$. In contrast, when j is odd (that is $\mu_v=1$) the next state is $[\theta_{l+1,1},\theta_{l+1,2},...,\theta_{l+1,v}]=[0,\mu_1,\mu_2,...,\mu_{v-1}]$ since $y_l \oplus b_1\mu_1 \oplus ... \oplus b_{v-1}\mu_{v-1} \oplus b_v\mu_v=0$. Hence

$$\theta_{l+1}=\phi(y_l,\theta_l)=\phi(0,q_j)=\begin{cases} q_{2^{v-1}+j/2} & j \text{ even} \\ q_{\lfloor j/2 \rfloor} & j \text{ odd} \end{cases} \tag{5.26}$$

for case (2) with $b_1\mu_1 \oplus ... \oplus b_{v-1}\mu_{v-1}=1$. Using a similar analysis when the current input is $y_l=1$ we can find

$$\theta_{l+1}=\phi(y_l,\theta_l)=\phi(1,q_j)=\begin{cases} q_{j/2} & j \text{ even} \\ q_{2^{v-1}+\lfloor j/2 \rfloor} & j \text{ odd} \end{cases} \tag{5.27}$$

for case (2).

Therefore each element $B_{0,jj'}$ of $\mathbf{B}_0=[B_{0,jj'}]$ is equal to

$$B_{0,jj'}=\begin{cases} 1, & \text{for } j \text{ even, } j'=j/2 \text{ and } b_1\mu_1+...+b_{v-1}\mu_{v-1}=0 \\ 1, & \text{for } j \text{ odd, } j'=2^{v-1}+\lfloor j/2 \rfloor \text{ and } b_1\mu_1+...+b_{v-1}\mu_{v-1}=0 \\ 1, & \text{for } j \text{ even, } j'=2^{v-1}+j/2 \text{ and } b_1\mu_1+...+b_{v-1}\mu_{v-1}=1 \\ 1, & \text{for } j \text{ odd, } j'=\lfloor j/2 \rfloor \text{ and } b_1\mu_1+...+b_{v-1}\mu_{v-1}=1 \\ 0, & \text{otherwise} \end{cases} \tag{5.28}$$

whereas each element $B_{1,jj'}$ of $\mathbf{B}_1=[B_{1,jj'}]$ is equal to

$$B_{1,jj'}=\begin{cases}1, & \text{for } j \text{ even, } j'=2^{v-1}+j/2 \text{ and } b_1\mu_1+\ldots+b_{v-1}\mu_{v-1}=0\\ 1, & \text{for } j \text{ odd, } j'=\lfloor j/2\rfloor \text{ and } b_1\mu_1+\ldots+b_{v-1}\mu_{v-1}=0\\ 1, & \text{for } j \text{ even, } j'=j/2 \text{ and } b_1\mu_1+\ldots+b_{v-1}\mu_{v-1}=1\\ 1, & \text{for } j \text{ odd, } j'=2^{v-1}+\lfloor j/2\rfloor \text{ and } b_1\mu_1+\ldots+b_{v-1}\mu_{v-1}=1\\ 0, & \text{otherwise}\end{cases} \tag{5.29}$$

Using these results the behaviour of the SE pair Markov model's state transitions can now be analysed. Assume the state of the SE pair MC state process at time l to be $\sigma_l=(\rho_l,\theta_l)=(y_l,\theta_l)=(n',q_{j'})=s_{n'N_q+j'}$ with corresponding encoder memory content $[\theta_{l,1},\theta_{l,2},\ldots,\theta_{l,v}]=[\mu'_1,\mu'_2,\ldots,\mu'_v]$. From Figure 5-3 it can be seen that the encoder state $q_{j'}$ can only be reached from two previous encoder states q_k and q_{k+1} with respective memory content at time $l-1$ given by

$$[\theta_{l-1,1},\theta_{l-1,2},\ldots,\theta_{l-1,v}]=[\mu'_2,\ldots,\mu'_v,0] \text{ and } [\theta_{l-1,1},\theta_{l-1,2},\ldots,\theta_{l-1,v}]=[\mu'_2,\ldots,\mu'_v,1] \tag{5.30}$$

where $k=\mu'_2 2^{v-1}+\ldots+\mu'_v 2=(2j'\bmod 2^v)$. From Equation (5.10) and the previous analysis it can be seen that in order to reach $q_{j'}$ from any of these states it is needed that

$$y_{l-1}=\mu'_1\oplus b_1\mu'_2\oplus\ldots\oplus b_{v-1}\mu'_v=n^{(k)} \text{ and } y_{l-1}=\mu'_1\oplus b_1\mu'_2\oplus\ldots\oplus b_{v-1}\mu'_v\oplus b_v=n^{(k+1)} \tag{5.31}$$

for the states q_k and q_{k+1} respectively. Note $n^{(k)}\neq n^{(k+1)}=n^{(k)}\oplus 1$ as far as $b_v=1$. Thus the SE pair MC state $(n',q_{j'})=s_{n'N_q+j'}$ can only be accessed from the states

$$(n^{(k)},q_k)=s_{n^{(k)}N_q+k} \text{ and } (n^{(k+1)},q_{k+1})=s_{n^{(k+1)}N_q+k+1} \tag{5.32}$$

Therefore the $i'=(n'N_q+j')^{\text{th}}$ column in the one step transition probabilities matrix $\mathbf{P}_\sigma$, corresponding to state $(n',q_{j'})=s_{n'N_q+j'}$ with encoder memory content $[\mu'_1,\mu'_2,\ldots,\mu'_v]$, has row entries equal to

$$p_{\sigma,ii'}=\begin{cases}p_{y,n^{(k)}n'}, & i=n^{(k)}N_q+k\\ p_{y,n^{(k+1)}n'}, & i=n^{(k+1)}N_q+k+1\\ 0, & \text{otherwise}\end{cases} \tag{5.33}$$

where $k=(2j'\bmod 2^v)$, $n^{(k)}=\mu'_1\oplus b_1\mu'_2\oplus\ldots\oplus b_{v-1}\mu'_v$, $n^{(k+1)}=\mu'_1\oplus b_1\mu'_2\oplus\ldots\oplus b_{v-1}\mu'_v\oplus b_v$, and $i=nN_q+j$ is the row index corresponding to state $(n,q_j)=s_{nN_q+j}$ with $n=0,1$ and $j=0,1,\ldots,N_q-1$. Now Theorem 5.2 can be introduced.

Theorem 5.2: *Consider a recursive convolutional encoder with $b_v=1$ and first order BMS input defined by Equations* (5.1) *to* (5.5). *Then the steady state probabilities of the SE pair Markov model state process, σ_l, with one step transition probabilities matrix, $\mathbf{P}_\sigma$, and state set, $\mathcal{S}$, as defined by Equations* (5.6) *and* (5.7) *respectively, are given by*

$$\pi_{\sigma,nN_q+j}=\pi_{\sigma,(n,q_j)}=\pi_{\sigma,s_{nN_q+j}}=\frac{\pi_{y,n}}{N_q} \tag{5.34}$$

where $n=0,1$, $\pi_{y,n}$ *is the BMS' steady state probability corresponding to state* r_n *with BMS' output* n, *and* $j=\mu_1 2^{v-1}+...+\mu_v$ *is defined by the encoder's memory content* $[\mu_1,\mu_2,...,\mu_v]$ *corresponding to the encoder state* q_j.

Proof: It is only needed to prove that

$$\pi_{\sigma,i'}=\pi_{\sigma,n'N_q+j'}=\pi_{\sigma,(n',q_{j'})}=\sum_{i=0}^{2N_q-1}\pi_{\sigma,i}p_{\sigma,ii'} \tag{5.35}$$

holds for every $n'=0,1$ and $j'=0,1,...,N_q-1$. First note $i'=n'N_q+j'$ and assume that $j'=\mu'_1 2^{v-1}+...+\mu'_v$ corresponds to the encoder memory content $[\mu'_1,\mu'_2,...,\mu'_v]$. Substituting Equation (5.33) in Equation (5.35) gives

$$\pi_{\sigma,i'}=\pi_{\sigma,n'N_q+j'}=\pi_{\sigma,n^{(k)}N_q+k}p_{y,n^{(k)}n'}+\pi_{\sigma,n^{(k+1)}N_q+k+1}p_{y,n^{(k+1)}n'} \tag{5.36}$$

where $n^{(k)}=\mu'_1\oplus b_1\mu'_2\oplus...\oplus b_{v-1}\mu'_v$, $n^{(k+1)}=\mu'_1\oplus b_1\mu'_2\oplus...\oplus b_{v-1}\mu'_v\oplus b_v$ and $k=(2j'\bmod 2^v)$ (note $k=\mu'_2 2^{v-1}+...+\mu'_v 2$). Replacing $\pi_{\sigma,i'}$ in Equation (5.36) by Equation (5.34) yields

$$\frac{\pi_{y,n'}}{N_q}=\pi_{\sigma,i'}=\pi_{\sigma,n'N_q+j'}=\pi_{\sigma,n^{(k)}N_q+k}p_{y,n^{(k)}n'}+\pi_{\sigma,n^{(k+1)}N_q+k+1}p_{y,n^{(k+1)}n'} \tag{5.37}$$

Substituting Equation (5.34) in the right side of Equation (5.36) gives

$$\pi_{\sigma,n^{(k)}N_q+k}p_{y,n^{(k)}n'}+\pi_{\sigma,n^{(k+1)}N_q+k+1}p_{y,n^{(k+1)}n'}=\frac{\pi_{y,n^{(k)}}}{N_q}p_{y,n^{(k)}n'}+\frac{\pi_{y,n^{(k+1)}}}{N_q}p_{y,n^{(k+1)}n'} \tag{5.38}$$

Replacing Equation (5.4) in the right side of Equation (5.38) yields

$$\pi_{\sigma,n^{(k)}N_q+k}p_{y,n^{(k)}n'}+\pi_{\sigma,n^{(k+1)}N_q+k+1}p_{y,n^{(k+1)}n'}=\frac{1}{N_q}\frac{p_{y,(n^{(k)}\oplus 1)n^{(k)}}p_{y,n^{(k)}n'}+p_{y,(n^{(k+1)}\oplus 1)n^{(k+1)}}p_{y,n^{(k+1)}n'}}{(p_{y,01}+p_{y,10})} \tag{5.39}$$

Nevertheless $n^{(k+1)}=n^{(k)}\oplus 1$ as $b_v=1$ and therefore

$$\frac{p_{y,(n^{(k)}\oplus 1)n^{(k)}}p_{y,n^{(k)}n'}+p_{y,(n^{(k+1)}\oplus 1)n^{(k+1)}}p_{y,n^{(k+1)}n'}}{(p_{y,01}+p_{y,10})}=\begin{cases}\dfrac{p_{y,n^{(k+1)}n^{(k)}}(p_{y,n^{(k)}n^{(k)}}+p_{y,n^{(k)}n^{(k+1)}})}{p_{y,01}+p_{y,10}}=\pi_{y,n^{(k)}}, & n'=n^{(k)}\\ \dfrac{p_{y,n^{(k)}n^{(k+1)}}(p_{y,n^{(k+1)}n^{(k)}}+p_{y,n^{(k+1)}n^{(k+1)}})}{p_{y,01}+p_{y,10}}=\pi_{y,n^{(k+1)}}, & n'=n^{(k+1)}\end{cases} \tag{5.40}$$

Hence Equation (5.38) becomes

$$\pi_{\sigma,n^{(k)}N_q+k}p_{y,n^{(k)}n'}+\pi_{\sigma,n^{(k+1)}N_q+k+1}p_{y,n^{(k+1)}n'}=\frac{\pi_{y,n^{(k)}}}{N_q}p_{y,n^{(k)}n'}+\frac{\pi_{y,n^{(k+1)}}}{N_q}p_{y,n^{(k+1)}n'}=\begin{cases}\dfrac{\pi_{y,n^{(k)}}}{N_q}, & n'=n^{(k)}\\ \dfrac{\pi_{y,n^{(k+1)}}}{N_q}, & n'=n^{(k+1)}\end{cases} \tag{5.41}$$

which is equivalent to the leftmost side of Equation (5.37). Therefore for recursive convolutional encoders with $b_v=1$ the steady state probabilities of the SE pair Markov model state process, σ_l, are given by Equation (5.34). Furthermore, it is important to highlight that $b_v=1$ is a necessary condition owing to the fact that otherwise $n^{(k)}=n^{(k+1)}$ and Equations (5.37) to (5.41) would not be valid for this case.

5.3 Spectral Line Suppression Capabilities of the Binary to *M*-ary Convolutional Encoders with First Order BMS Inputs for PPM IR-Based UWB Systems

The signal model considered in this chapter for the binary to M-ary convolutionally coded time hopping (TH) scheme with first order BMS for PPM IR-based UWB is that introduced in Section 4.3.1. The main difference is that this time the SE pair Markov model is defined by Equations (5.6), (5.7), (5.8), (5.14) and (5.15) for feedforward encoders, and Equations (5.6), (5.7), (5.8), (5.28), (5.29), (5.33) and (5.34) for recursive encoders. Therefore the power spectral density formulas introduced in Section 4.3.2 can be used for these encoders as well, that is

$$\begin{aligned} \overline{S}(f) &= S_{C.1}(f) + S_{C.3}(f) + S_D(f) \qquad N_w = 1 \\ \overline{S}(f) &= S_{C.1}(f) + S_{C.2}(f) + S_{C.3}(f) + S_D(f) \quad N_w > 1 \end{aligned} \tag{5.42}$$

$$S_{C.1}(f) = \frac{1}{T_r}|W(f)|^2 - \frac{1}{T_s}|W(f)|^2 \left| \sum_{k=0}^{N_w-1} \sum_{i=0}^{N_\sigma-1} e^{j2\pi fkT_r} e^{j2\pi f\beta_{i,k}T_\beta} \pi_i \right|^2 \tag{5.43}$$

$$S_{C.2}(f) = \frac{2|W(f)|^2}{T_s} \sum_{k=0}^{N_w-2} \sum_{k'=k+1}^{N_w-1} \mathrm{Re}\left\{ e^{-j2\pi f(k'-k)T_r} \sum_{i=0}^{N_\sigma-1} e^{-j2\pi f(\beta_{i,k'}-\beta_{i,k})T_\beta} \pi_i \right\} \tag{5.44}$$

$$S_{C.3}(f) = \frac{2|W(f)|^2}{T_s} \mathrm{Re}\left\{ \sum_{k,k'=0}^{N_w-1} \sum_{m=1}^{\infty} \sum_{i,i'=0}^{N_\sigma-1} e^{-j2\pi f(k'-k)T_r} e^{-j2\pi fT_s m} e^{-j2\pi f(\beta_{i',k'}-\beta_{i,k})T_\beta} \pi_i (p_{ii'}^{(|m|)} - \pi_{i'}) \right\} \tag{5.45}$$

$$S_D(f) = \frac{1}{(T_s)^2}|W(f)|^2 \left| \sum_{k=0}^{N_w-1} \sum_{i=0}^{N_\sigma-1} e^{j2\pi fkT_r} e^{j2\pi f\beta_{i,k}T_\beta} \pi_i \right|^2 \sum_{r=-\infty}^{\infty} \delta\left(f - \frac{r}{T_s}\right) \tag{5.46}$$

where $S_C(f)$ is the PSD's continuous part, $S_D(f)$ is the PSD's discrete part, $W(f)$ is the Fourier transform of the transmitted pulse shape $w(t)$; $\beta_{i,k} = \zeta_{s_i}^{(k)} \in \{0,1,...,M-1\}$ is the k^{th} M-ary symbol generated when the encoder is in state s_i; $N_\sigma = 2N_q$ is the number of states in the SE pair Markov model; N_w is the number of M-ary symbols transmitted per state as defined by the encoder rate $1/\kappa = 1/N_w$; T_β is the PPM modulation index; T_r is the mean repetition time between pulses and $T_s = N_w T_r$ is the encoder's output vector time. As in Section 4.3.2 the notation $\pi_i = \pi_{\sigma,i}$ and $p_{ii'}^{(n)} = p_{\sigma,ii'}^{(n)}$ has been used for the steady state and the n^{th} step transition probabilities of the SE pair Markov model's state process, σ_l, respectively.

Remember we are interested in binary to M-ary convolutional encoders whose SE pair Markov model satisfies the spectral line suppression condition

$$\sum_{i=0}^{N_\sigma-1} e^{j2\pi f\beta_{i,k}T_\beta} \pi_i = \frac{1}{M} \sum_{\varsigma=0}^{M-1} e^{j2\pi f\varsigma T_\beta}; \quad k = 0,1,...,N_w - 1 \tag{5.47}$$

as defined in Sections 4.3.2 and 4.3.3. Therefore in the next sections the feasibility of fulfilling Equation (5.47) when using feedforward and recursive convolutional encoders will be assessed.

5.3.1 Failure of the Spectral Line Suppression Condition for Feedforward Binary to *M*-ary Convolutional Encoders with First Order BMS Inputs

This section analyses the spectral line suppression condition when feedforward implementations for binary to *M*-ary convolutional encoders are considered. Theorem 5.1 defines the steady state behaviour of the SE pair Markov model for this set of convolutional encoders. By analysing Equation (5.15) in Theorem 5.1, it can be seen that satisfying the spectral line suppression condition defined by Equation (5.47) may be difficult to achieve for most combinations of BMS' transition probabilities. This is due to every steady state probability $\pi_i = \pi_{\sigma,i}$ is a function of at least two probabilities from the following set:

$$0 < p_{y,01} < 1, \quad 0 < p_{y,10} < 1, \quad p_{y,00} = 1 - p_{y,01}, \quad p_{y,11} = 1 - p_{y,10}, \quad \pi_{y,0} = \frac{p_{y,10}}{p_{y,01} + p_{y,10}}, \quad \pi_{y,1} = \frac{p_{y,01}}{p_{y,01} + p_{y,10}} \tag{5.48}$$

Moreover, even though the fulfilment of Equation (5.47) for particular combinations of $p_{y,01}$ and $p_{y,10}$ can be achieved through proper specific designs of the output mapping function (like in Chapter 4 for $p_{y,ii'} = 1/2$), there exist combinations for which the spectral line suppression condition can not be satisfied no matter what (valid) output mapping function, $\gamma(\cdot)$, is used. In order to see this note that it is not difficult to find values of $p_{y,01}$ and $p_{y,10}$ such that

$$\pi_i = \pi_{nN_q+j} = \pi_{s_{N_q+j}} = p_{y,\mu_1 n} p_{y,\mu_2 \mu_1} \cdots p_{y,\mu_{v-1}\mu_{v-2}} p_{y,\mu_v \mu_{v-1}} \pi_{y,\mu_v} > \frac{1}{M} \tag{5.49}$$

which concludes to the no fulfilment of Equation (5.47). Therefore it can be stated that feedforward structures do not provide a good construction framework for spectral line suppressive (SLS) binary to *M*-ary convolutional encoders with unbalanced first order BMS inputs.

5.3.2 Spectral Line Suppression Condition Analysis for Recursive Binary to *M*-ary Convolutional Encoders with First Order BMS Inputs

Now consider recursive implementations for binary to *M*-ary convolutional encoders with $b_v = 1$ (see Figure 5-3). The SE pair Markov model for this kind of convolutional encoders has steady stated probabilities defined by Theorem 5.2. From Equation (5.34) in Theorem 5.2 these probabilities can only assume one of two values namely

$$\pi_i = \pi_{s_{0N_q+j}} = \frac{\pi_{y,0}}{N_q}, \text{ for } i = 0,1,...,N_q - 1 \quad \text{and} \quad \pi_i = \pi_{s_{N_q+j}} = \frac{\pi_{y,1}}{N_q}, \text{ for } i = N_q, N_q + 1,...,2N_q - 1 \tag{5.50}$$

where $j = 0,1,...,N_q - 1$ .Therefore the recursive structures with $b_v = 1$ provide a better framework for the construction of SLS encoders as shown next.

Let the number of encoder states to be $N_q = 2^v = \eta M$ with $\eta \geq 1$. Thus the number of states in the SE Markov model will be $N_\sigma = 2\eta M$. Now assume k has been fixed in Equation (5.47) and

focus on the first set of states $s_i = s_{0N_q+j}$ with $i = j = 0,1,...,N_q - 1$ and $\pi_i = \pi_{y,0}/N_q$. Change the label of these states to s_{i_0} such that the k^{th} M-ary output symbol, $\beta_{i_0,k} = \zeta_{s_{i_0}}^{(k)}$, of the first $\eta_0^{(i_0)}$ states is equal to 0; then the following $\eta_1^{(i_0)}$ states will have k^{th} M-ary output symbol equal to 1, and so on until the last $\eta_{M-1}^{(i_0)}$ states have k^{th} M-ary output symbol equal to $M-1$. Similarly change the label of the set of states $s_i = s_{N_q+j}$ with $i = N_q + j = N_q,...,2N_q - 1$ and $\pi_i = \pi_{y,1}/N_q$ to s_{i_1} such that the first $\eta_0^{(i_1)}$ states have k^{th} M-ary output symbol, $\beta_{i_1,k} = \zeta_{s_{i_1}}^{(k)}$, equal to 0, and so on until the last $\eta_{M-1}^{(i_1)}$ states have k^{th} M-ary output symbol equal to $M-1$. Using this new labelling the left hand side of Equation (5.47) can be expanded as

$$\begin{aligned}\sum_{i=0}^{N_\sigma-1} e^{j2\pi f\beta_{i,k}T_\beta}\pi_i = \frac{\pi_{y,0}}{N_q}\Bigg\{\sum_{i_0=0}^{\eta_0^{(i_0)}-1} e^0 + \sum_{i_0=\eta_0^{(i_0)}}^{\eta_0^{(i_0)}+\eta_1^{(i_0)}-1} e^{j2\pi fT_\beta} + ... + \sum_{i_0=N_q-\eta_{M-1}^{(i_0)}}^{N_q-1} e^{j2\pi f(M-1)T_\beta}\Bigg\} + \\ + \frac{\pi_{y,1}}{N_q}\Bigg\{\sum_{i_1=0}^{\eta_0^{(i_1)}-1} e^0 + \sum_{i_1=\eta_0^{(i_1)}}^{\eta_0^{(i_1)}+\eta_1^{(i_1)}-1} e^{j2\pi fT_\beta} + ... + \sum_{i_1=N_q-\eta_{M-1}^{(i_1)}}^{N_q-1} e^{j2\pi f(M-1)T_\beta}\Bigg\}\end{aligned} \tag{5.51}$$

for this kind of recursive convolutional encoders. Note Equation (5.51) can be simplified to

$$\sum_{i=0}^{N_\sigma-1} e^{j2\pi f\beta_{i,k}T_\beta}\pi_i = \sum_{\varsigma=0}^{M-1}\left(\frac{\pi_{y,0}}{N_q}\eta_\varsigma^{(i_0)} + \frac{\pi_{y,1}}{N_q}\eta_\varsigma^{(i_1)}\right)e^{j2\pi f\varsigma T_\beta} \tag{5.52}$$

Therefore in order to satisfy the spectral line suppressive condition the output mapping function, $\gamma(\cdot)$, must designed such that

$$\sum_{i=0}^{N_\sigma-1} e^{j2\pi f\beta_{i,k}T_\beta}\pi_i = \sum_{\varsigma=0}^{M-1}\left(\frac{\pi_{y,0}}{N_q}\eta_\varsigma^{(i_0)} + \frac{\pi_{y,1}}{N_q}\eta_\varsigma^{(i_1)}\right)e^{j2\pi f\varsigma T_\beta} = \frac{1}{M}\sum_{\varsigma=0}^{M-1} e^{j2\pi f\varsigma T_\beta}; \tag{5.53}$$

is satisfied.

For Equation (5.53) to be true it is needed that

$$\frac{\pi_{y,0}}{N_q}\eta_\varsigma^{(i_0)} + \frac{\pi_{y,1}}{N_q}\eta_\varsigma^{(i_1)} = \frac{1}{M} \tag{5.54}$$

for $\varsigma = 0,1,...,M-1$. However $0 < \pi_{y,0} < 1$ with $\pi_{y,1} = 1 - \pi_{y,0}$ for every combination of $0 < p_{y,01} < 1$ and $0 < p_{y,10} < 1$ (see Equation (5.4)). Therefore if $\gamma(\cdot)$ is such that

$$\eta_\varsigma^{(i_0)} = \eta_\varsigma^{(i_1)} = \eta \tag{5.55}$$

then

$$\frac{\pi_{y,0}}{N_q}\eta_\varsigma^{(i_0)} + \frac{\pi_{y,1}}{N_q}\eta_\varsigma^{(i_1)} = \frac{\pi_{y,0}}{\eta M}\eta + \frac{\pi_{y,1}}{\eta M}\eta = \frac{1}{M} \tag{5.56}$$

and Equation (5.53), thus the spectral line suppression condition, is satisfied for every combination of $0 < p_{y,01} < 1$ and $0 < p_{y,10} < 1$.

Consequently it can be concluded that by using recursive structures with total encoder memory $v \geq \log_2(M) = L$ and feedback loop with $b_v = 1$ (see Figure 5-3), spectral line suppressive (SLS) binary to M-ary convolutional encoders for unbalanced (that is $p_{y,ii'} \neq 1/2$) binary Markov sources (BMS) can be constructed.

5.4 Code Search Procedure for Recursive Spectral Line Suppressive Binary to *M*-ary Convolutional Encoders with Unbalanced First Order BMS Inputs

The code search procedure used in this chapter for recursive spectral line suppressive (RSLS) binary to *M*-ary convolutional encoders is based on that introduced in Section 4.4.3 for feedforward spectral line suppressive encoders (FSLS). The main difference is that this time the main goal is to find RSLS convolutional encoders which eliminate as many spectral lines as the FSLS convolutional encoders and the PCTH scheme even when the input consists of binary data streams generated by an unbalanced (that is $p_{y,ii'} \neq 1/2$) BMS. As in Chapter 4, the second aim of the code search procedure is the optimisation of the information weight spectrum (IWS) so that an RSLS binary to *M*-ary convolutional encoder with the best possible IWS is found. The modified flowchart of the code search procedure is shown in Figure 5-4 where the changes/additions to the flowchart introduced in Section 4.4.3 – Figure 4-8 have been highlighted with bold lines.

In a similar way to that shown in Section 4.4.3, the code search procedure has not been restricted to rate 1 binary to *M*-ary convolutional codes as rate 1/2 and rate 1/3 encoders have been included. Furthermore the notation introduced in Section 4.4.3 will be used in this section and thus for a rate $1/\kappa$ binary to *M*-ary convolutional code κ sets of $L = \log_2(M)$ generators will be provided. However, this time each element of the transfer function matrix

$$\mathbf{G}(D) = [\{g_0^{(0)}(D)\ g_1^{(0)}(D)...g_{L-1}^{(0)}(D)\}^{(0)}\ \ ...\ \ \{g_0^{(\kappa-1)}(D)\ g_1^{(\kappa-1)}(D)...g_{L-1}^{(\kappa-1)}(D)\}^{(\kappa-1)}] \tag{5.57}$$

has the form

$$g_i^{(k)}(D) = \frac{a_i^{(k)}(D)}{b(D)} = \frac{a_{i,0}^{(k)} + a_{i,1}^{(k)}D + ... + a_{i,\nu}^{(k)}D^{\nu}}{1 + b_1 D + ... + b_{\nu-1}D^{\nu-1} + b_{\nu}D^{\nu}} \tag{5.58}$$

where D is the delay operator, $a_{i,j}^{(k)} \in \{0,1\}$, $b_j \in \{0,1\}$, $i = 0,1,...,L-1$, $k = 0,1,...,\kappa-1$, and $g_i^{(k)}(D)$ is the i^{th} rational generator of the k^{th} set corresponding to the k^{th} *M*-ary output. The numerator in Equation (5.58)

$$a_i^{(k)}(D) = a_{i,0}^{(k)} + a_{i,1}^{(k)}D + ... + a_{i,\nu}^{(k)}D^{\nu} \tag{5.59}$$

describes the generator's feedforward connections (feedforward polynomial) while the denominator

$$b(D) = 1 + b_1 D + ... + b_{\nu-1}D^{\nu-1} + b_{\nu}D^{\nu} \tag{5.60}$$

is the feedback polynomial describing the shift register's feedback connections in Figure 5-3 (see Section 2.7).

If the input sequence, $y = \{y_0 y_1 y_2 ... y_l ...\}$, is expressed in terms of the delay operator as

$$y(D) = y_0 + y_1 D + y_2 D^2 + ... + y_l D^l + ... \tag{5.61}$$

then the *M*-ary symbols produced by the k^{th} generator set can be obtained by using

$$z^{(k)}(D) = 2^{(L-1)} y(D) g_0^{(k)}(D) + 2^{(L-2)} y(D) g_1^{(k)}(D) + \ldots + y(D) g_{L-1}^{(k)}(D) \quad (5.62)$$

and therefore the encoder output mapping function, $\gamma(\cdot)$, is defined by the rational generator sets in Equation (5.57) and the binary to M-ary conversion defined by Equation (5.62).

5.4.1 Modifications to the Code Search Procedure Introduced in Section 4.4 for the Search of RSLS Binary to M-ary Convolutional Encoders with Unbalanced First Order BMS Inputs

As previously mentioned, the code search procedure described by the flowchart in Figure 5-4 is based on the flowchart introduced in Section 4.4.3 – Figure 4-8. Therefore only an explanation of the main flowchart's modifications will be provided referring the reader to Section 4.4.3 for the explanation of the unmodified tasks.

The main modification on *Task 1* is the addition of the feedback polynomial selection block. The feedback polynomials' selection pool for an encoder with total encoder memory $v \geq \log_2(M) = L$ was chosen to consist of the simple $b(D) = 1 + D^v$ polynomial plus all known primitive polynomials of degree v. The heuristic reason for the inclusion of the primitive polynomials is based on the fact that the pseudo noise (PN) linear-feedback shift register sequences with the longest period (m-sequences), hence best noise-like properties, are generated when the feedback connections are defined by a primitive polynomial, [82, 83, 95, 136]. Furthermore, it has been shown that for self-synchronizing scrambled sequences the selection of primitive scrambling polynomials is the best in terms of the signal's PSD characteristics when used in pulse amplitude modulation (PAM) based systems, [143]. Therefore it is expected that RSLS convolutional encoders with primitive polynomial based feedback will provide better PSD characteristics than encoders with $b(D) = 1 + D^v$ based feedback.

Task 2 has not been modified while *Task 3* was slightly modified to account for the use rational generators instead of polynomial ones. Thus the new generator sets are obtained by defining new feedforward polynomials (numerator in Equation (5.58)) each time while the feedback polynomial remains unchanged.

Task 4 is modified to obtain the SE pair Markov model from an unbalanced BMS with stationary probabilities $\pi_{y,0} \neq \pi_{y,1} \neq 1/2$. One or two specific numeric values for $p_{y,01} \neq 1/2$ and $p_{y,10} \neq p_{y,01}$ are used at this stage. The encoder is discarded if it does not satisfy Equation (5.53) and hence does not satisfy the spectral line suppression condition defined by Equation (5.47) (see Section 5.3).

Task 5 has not been modified as it deals with standard encoder's properties.

Finally in *Task 6* the operations related to the comparison of the encoder's IWS are the same as in Figure 4-8 (Section 4.4.3). Nevertheless two blocks have been added before storing the encoder under test in case its IWS is as good as or better than the current best. These blocks further verify

that Equations (5.47), (5.53) and (5.56) are satisfied by the encoder under test for all combinations of $p_{y,01} = \{0.1, 0.2, ..., 0.9\}$ and $p_{y,10} = \{0.1, 0.2, ..., 0.9\}$.

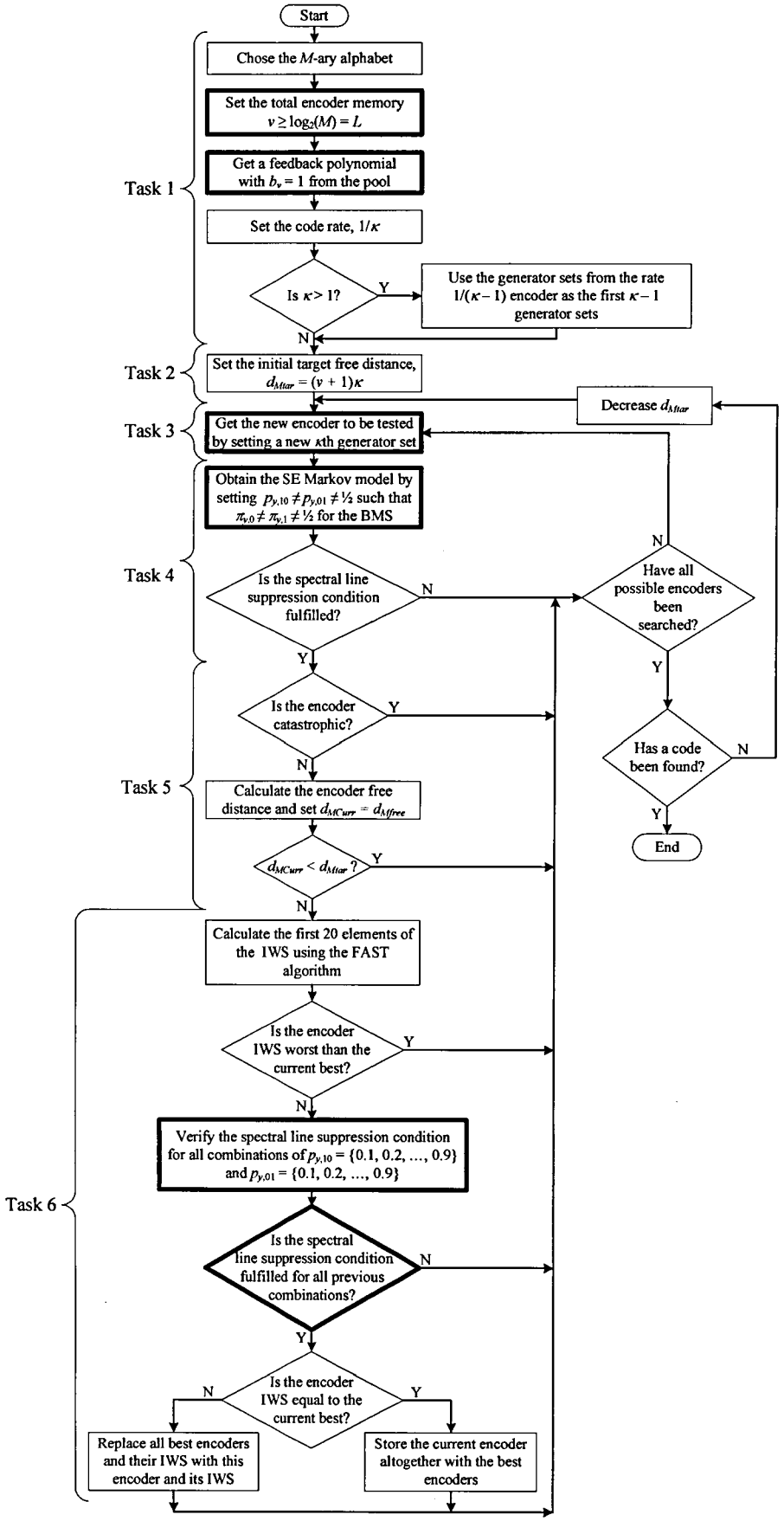


Figure 5-4. Flowchart diagram of the code search procedure for RSLS binary to *M*-ary convolutional encoders.

The algorithm ends when an encoder with spectral line suppressive capabilities for unbalanced BMS and the best possible IWS is found.

5.4.2 Code Search Results of RSLS Binary to *M*-ary Convolutional Encoders for Unbalanced First Order BMS Inputs

Novel maximum free distance (MFD) recursive spectral line suppressive (RSLS) binary to *M*-ary convolutional encoders for unbalanced first order BMS inputs have been found using the code search procedure introduced in the previous section. RSLS encoders with the best possible IWS for 16-ary, 32-ary, 64-ary and 128-ary orthogonal signalling are reported in Table 5-1 for recursive structures with feedback polynomials of the form $b(D)=1+D^{v}$ and in Table 5-2 for recursive structures with primitive feedback polynomials.

Table 5-1. Best recursive spectral line suppressive (RSLS) binary to *M*-ary convolutional encoders with feedback polynomial $b(D) = 1 + D^{v}$. The feedback and feedforward polynomials are given in octal form. The first 15 elements of the IWS and distance spectrum are reported in the last column.

M	v	Feedback polynomial	Feedforward polynomials sets in octal form	d_{Mfree}	Information weight spectrum (IWS): $[B_{d_{Mfree}}, B_{d_{Mfree}+1}, \ldots, B_{d_{Mfree}+15}]$ Distance spectrum: $\{A_{d_{Mfree}}, A_{d_{Mfree}+1}, \ldots, A_{d_{Mfree}+15}\}$
16	4	21	(22, 10, 4, 1)	5	[2, 4, 14, 24, 60, 116, 280, 566, 1276, 2626, 5744, 11942, 25616, 53408, 113160] {1, 1, 3, 4, 11, 18, 42, 75, 163, 307, 642, 1243, 2546, 5004, 10137}
			(20, 14, 2, 11)	10	[2, 0, 4, 0, 10, 10, 14, 10, 30, 38, 76, 58, 132, 176, 312] {1, 0, 1, 0, 2, 2, 2, 2, 6, 6, 12, 8, 21, 26, 40}
			(20, 4, 12, 11)	15	[2, 0, 0, 4, 0, 0, 10, 0, 16, 8, 0, 16, 24, 8, 44] {1, 0, 0, 1, 0, 0, 2, 0, 3, 1, 0, 3, 5, 1, 7}
	5	41	(21, 10, 44, 42)	6	[2, 4, 10, 34, 64, 132, 348, 692, 1554, 3580, 7488, 16718, 36636, 78514, 172118] {1, 1, 2, 6, 9, 21, 46, 86, 189, 389, 786, 1662, 3409, 7029, 14630}
			(42,23,10,5)	12	[2, 0, 4, 0, 10, 0, 24, 22, 34, 36, 68, 98, 130, 274, 312] {1, 0, 1, 0, 2, 0, 4, 4, 4, 6, 11, 14, 18, 35, 41}
			(61, 10, 45, 43)	18	[2, 0, 0, 4, 0, 0, 10, 0, 0, 24, 0, 38, 18, 0, 54] {1, 0, 0, 1, 0, 0, 2, 0, 0, 4, 0, 6, 2, 0, 9}
	6	101	(142, 124, 10, 105)	7	[2, 4, 10, 40, 72, 166, 406, 818, 1898, 4388, 9358, 21166, 47176, 102448, 227920] {1, 1, 2, 7, 9, 23, 53, 95, 215, 466, 921, 2008, 4225, 8708, 18608}
			(122, 10, 44, 143)	14	[2, 0, 4, 0, 10, 0, 24, 38, 18, 60, 72, 126, 172, 242, 402] {1, 0, 1, 0, 2, 0, 4, 6, 2, 8, 9, 18, 23, 30, 48}
			(164, 70, 62, 21)	21	[2, 0, 0, 4, 0, 0, 10, 0, 0, 24, 4, 52, 0, 6, 80] {1, 0, 0, 1, 0, 0, 2, 0, 0, 4, 1, 7, 0, 1, 10}
	7	201	(254, 150, 42, 21)	8	[2, 4, 10, 46, 82, 216, 510, 968, 2568, 5302, 12026, 27916, 60566, 137370, 305528] {1, 1, 2, 8, 10, 26, 58, 106, 262, 515, 1122, 2426, 5059, 10926, 23161}
			(351, 161, 144, 112)	16	[2, 0, 4, 0, 10, 0, 24, 56, 0, 76, 130, 92, 238, 304, 442] {1, 0, 1, 0, 2, 0, 4, 8, 0, 10, 14, 11, 27, 35, 53}
			(362, 172, 104, 21)	24	[2, 0, 0, 4, 0, 0, 10, 0, 0, 24, 22, 34, 0, 20, 90] {1, 0, 0, 1, 0, 0, 2, 0, 0, 4, 4, 4, 0, 3, 11}
32	5	41	(40, 21, 10, 4, 2)	6	[2, 4, 10, 28, 56, 110, 256, 536, 1144, 2472, 5168, 10968, 23144, 48482, 101860] {1, 1, 2, 5, 8, 18, 36, 69, 143, 282, 563, 1134, 2256, 4520, 9045}
			(20, 10, 44, 42, 1)	12	[2, 0, 4, 0, 10, 0, 24, 10, 46, 10, 86, 38, 182, 106, 380] {1, 0, 1, 0, 2, 0, 4, 2, 6, 2, 14, 6, 25, 14, 50}
			(60, 10, 4, 22, 1)	18	[2, 0, 0, 4, 0, 0, 10, 0, 0, 24, 0, 16, 40, 0, 16] {1, 0, 0, 1, 0, 0, 2, 0, 0, 4, 0, 3, 5, 0, 3}
	6	101	(102, 60, 10, 4, 41)	7	[2, 4, 10, 24, 66, 136, 260, 604, 1346, 2822, 6006, 13080, 27842, 59116, 125836] {1, 1, 2, 4, 10, 17, 37, 74, 154, 304, 629, 1269, 2590, 5240, 10686.}
			(142, 120, 10, 4, 103)	14	[2, 0, 4, 0, 10, 0, 24, 0, 56, 22, 106, 36, 188, 114, 394] {1, 0, 1, 0, 2, 0, 4, 0, 8, 4, 12, 6, 26, 16, 47}
			(110, 20, 44, 52, 41)	21	[2, 0, 0, 4, 0, 0, 10, 0, 0, 24, 0, 0, 56, 0, 36] {1, 0, 0, 1, 0, 0, 2, 0, 0, 4, 0, 0, 8, 0, 6}
	7	201	(204, 146, 20, 10, 103)	8	[2, 4, 10, 24, 72, 144, 326, 632, 1528, 3144, 6906, 14738, 32262, 68376, 147134] {1, 1, 2, 4, 11, 17, 39, 77, 169, 323, 681, 1396, 2871, 5822, 12021}
			(254, 144, 20, 12, 41)	16	[2, 0, 4, 0, 10, 0, 24, 0, 56, 36, 92, 64, 230, 128, 410] {1, 0, 1, 0, 2, 0, 4, 0, 8, 6, 10, 8, 25, 18, 49}
			(222, 160, 10, 104, 121)	24	[2, 0, 0, 4, 0, 0, 10, 0, 0, 24, 0, 0, 56, 0, 60, 68] {1, 0, 0, 1, 0, 0, 2, 0, 0, 4, 0, 0, 8, 0, 9}
64	6	101	(100, 41, 20, 10, 4, 2)	7	[2, 4, 10, 24, 60, 128, 238, 538, 1156, 2440, 5114, 10872, 22736, 47686, 99718] {1, 1, 2, 4, 9, 16, 34, 67, 135, 267, 539, 1073, 2150, 4295, 8599}
			(40, 20, 10, 104, 102, 1)	14	[2, 0, 4, 0, 10, 0, 24, 0, 56, 10, 118, 10, 214, 38, 470] {1, 0, 1, 0, 2, 0, 4, 0, 8, 2, 14, 2, 30, 6, 57}
			(140, 20, 10, 4, 102, 1)	21	[2, 0, 4, 0, 10, 0, 24, 0, 56, 10, 118, 10, 214, 38, 470] {1, 0, 1, 0, 2, 0, 4, 0, 8, 2, 14, 2, 30, 6, 57}
	7	201	(202, 140, 20, 10, 4, 101)	8	[2, 4, 10, 24, 56, 138, 296, 548, 1244, 2658, 5700, 11994, 25134, 53480, 112190] {1, 1, 2, 4, 8, 18, 33, 69, 138, 278, 564, 1128, 2285, 4597, 9270}
			(302, 142, 20, 10, 4, 101)	16	[2, 0, 4, 0, 10, 0, 24, 0, 56, 0, 128, 22, 266, 36, 476] {1, 0, 1, 0, 2, 0, 4, 0, 8, 0, 16, 4, 28, 6, 58}
			(202, 40, 22, 10, 104, 101)	24	[2, 0, 0, 4, 0, 0, 10, 0, 0, 24, 0, 0, 56, 0, 0] {1, 0, 0, 1, 0, 0, 2, 0, 0, 4, 0, 0, 8, 0, 0}
128	7	201	(202, 100, 40, 20, 10, 4, 1])	8	[2, 4, 10, 24, 56, 132, 288, 526, 1178, 2494, 5268, 11066, 23074, 48496, 100968] {1, 1, 2, 4, 8, 17, 32, 66, 131, 262, 525, 1047, 2099, 4193, 8389}
			(200, 140, 20, 10, 4, 2, 101)	16	[3, 0, 4, 0, 11, 0, 26, 0, 60, 0, 136, 4, 236, 13, 531, 36, 1110, 101, 2311, 266] {1, 0, 1, 0, 2, 0, 4, 0, 8, 0, 16, 1, 31, 2, 62, 5, 122, 12, 241, 28}
			(200, 40, 20, 10, 4, 102, 101)	24	[2, 0, 0, 4, 0, 0, 10, 0, 0, 24, 0, 0, 56, 0, 0] {1, 0, 0, 1, 0, 0, 2, 0, 0, 4, 0, 0, 8, 0, 0}

Table 5-2. Best recursive spectral line suppressive (RSLS) binary to M-ary convolutional encoders with primitive feedback polynomials. The feedback and feedforward polynomials are given in octal form. The first 15 elements of the IWS and distance spectrum are reported in the last column.

M	v	Feedback polynomial	Feedforward polynomials sets in octal form	d_{Mfree}	Information weight spectrum (IWS): $[B_{d_{Mfree}}, B_{d_{Mfree}+1}, \ldots, B_{d_{Mfree}+15}]$ Distance spectrum: $\{A_{d_{Mfree}}, A_{d_{Mfree}+1}, \ldots, A_{d_{Mfree}+15}\}$
16	4	23	(30, 4, 22, 21)	5	[3, 4, 11, 24, 52, 118, 255, 556, 1199, 2582, 5523, 11784, 25049, 53096, 112238] {1, 1, 2, 5, 9, 19, 37, 76, 151, 305, 612, 1231, 2474, 4973, 9997}
			(24, 14, 6, 5)	10	[3, 0, 4, 0, 11, 0, 18, 11, 39, 21, 78, 55, 181, 115, 375] {1, 0, 1, 0, 2, 0, 4, 2, 7, 3, 13, 8, 27, 15, 52}
			(10, 4, 22, 21)	15	[3, 0, 0, 4, 0, 0, 11, 0, 0, 18, 0, 18, 26, 6, 28] {1, 0, 0, 1, 0, 0, 2, 0, 0, 4, 0, 3, 5, 1, 4}
	5	45	(51, 30, 15, 12)	6	[3, 6, 11, 31, 59, 160, 292, 763, 1531, 3579, 7599, 16886, 36652, 79623, 173386] {1, 1, 2, 6, 9, 23, 40, 96, 181, 395, 794, 1664, 3427, 7085, 14696}
			(64, 50, 42, 5)	12	[3, 0, 6, 0, 11, 0, 20, 22, 26, 40, 92, 67, 189, 183, 366] {1, 0, 1, 0, 2, 0, 4, 4, 4, 6, 14, 9, 27, 24, 47}
			(62, 10, 46, 43)	18	[3, 0, 0, 6, 0, 0, 11, 0, 0, 20, 0, 34, 14, 7, 48] {1, 0, 0, 1, 0, 0, 2, 0, 0, 4, 0, 6, 2, 1, 7}
	6	103	(162, 10, 104, 143)	7	[3, 4, 11, 46, 68, 154, 460, 788, 2038, 4738, 9496, 22860, 50634, 107778, 247857] {1, 1, 2, 7, 10, 21, 57, 95, 224, 489, 941, 2117, 4462, 9106, 19844}
			(122, 70, 24, 11)	14	[3, 0, 4, 0, 11, 0, 26, 43, 17, 46, 79, 109, 186, 249, 366] {1, 0, 1, 0, 2, 0, 4, 6, 2, 8, 11, 14, 24, 31, 44}
			(164, 134, 126, 21)	21	[3, 0, 0, 4, 0, 0, 11, 0, 0, 26, 6, 54, 0, 5, 64] {1, 0, 0, 1, 0, 0, 2, 0, 0, 4, 1, 7, 0, 1, 10}
	7	301	(306, 142, 127, 116)	8	[3, 4, 11, 53, 79, 219, 429, 1015, 2327, 5171, 11592, 26102, 58129, 128802, 287803] {1, 1, 2, 8, 10, 26, 54, 111, 245, 513, 1085, 2327, 4912, 10416, 22211}
			(264, 50, 22, 105)	16	[3, 0, 4, 0, 11, 0, 26, 60, 0, 93, 93, 97, 256, 240, 507] {1, 0, 1, 0, 2, 0, 4, 8, 0, 11, 11, 13, 31, 28, 57}
			(311, 142, 30, 104)	24	[3, 0, 0, 4, 0, 0, 11, 0, 0, 26, 27, 33, 0, 17, 92] {1, 0, 0, 1, 0, 0, 2, 0, 0, 4, 4, 4, 0, 2, 11}
32	5	51	(50, 20, 14, 2, 11)	6	[3, 6, 11, 20, 54, 112, 250, 518, 1139, 2392, 5153, 10806, 23003, 48214, 101784] {1, 1, 2, 4, 9, 17, 35, 68, 140, 276, 560, 1112, 2245, 4478, 9011}
			(44, 24, 14, 6, 5)	12	[3, 0, 6, 0, 11, 0, 20, 0, 48, 12, 99, 22, 206, 62, 431] {1, 0, 1, 0, 2, 0, 4, 0, 8, 2, 15, 3, 29, 8, 57}
			(50, 20, 14, 12, 11)	18	[3, 0, 0, 6, 0, 0, 11, 0, 0, 20, 0, 0, 48, 0, 19] {1, 0, 0, 1, 0, 0, 2, 0, 0, 4, 0, 0, 8, 0, 3}
	6	141	(104, 102, 41, 20, 10)	7	[3, 4, 11, 26, 69, 111, 279, 601, 1322, 2744, 6079, 12862, 27621, 58622, 125269] {1, 1, 2, 4, 10, 17, 37, 74, 154, 304, 629, 1269, 2590, 5240, 10686}
			(60, 110, 104, 102, 21)	14	[3, 0, 4, 0, 11, 0, 26, 0, 60, 20, 97, 10, 233, 89, 467] {1, 0, 1, 0, 2, 0, 4, 0, 8, 4, 14, 2, 30, 13, 55}
			(121, 41, 11, 4, 23)	21	[3, 0, 0, 4, 0, 0, 11, 0, 0, 26, 0, 0, 60, 0, 33] {1, 0, 0, 1, 0, 0, 2, 0, 0, 4, 0, 0, 8, 0, 6}
	7	203	(204, 146, 20, 10, 107)	8	[3, 4, 11, 26, 80, 144, 280, 704, 1519, 3138, 6913, 15160, 32214, 68714, 148526] {1, 1, 2, 4, 11, 18, 36, 82, 165, 325, 682, 1411, 2858, 5842, 12071}
			(303, 141, 123, 113, 105)	16	[3, 0, 4, 0, 11, 0, 26, 0, 60, 44, 92, 48, 205, 138, 404] {3, 0, 4, 0, 11, 0, 26, 0, 60, 44, 92, 48, 205, 138, 404}
			(243, 141, 22, 10, 44)	24	[3, 0, 0, 4, 0, 0, 11, 0, 0, 26, 0, 0, 60, 0, 70] {1, 0, 0, 1, 0, 0, 2, 0, 0, 4, 0, 0, 8, 0, 9}
64	6	103	(140, 20, 10, 4, 102, 101)	7	[3, 4, 11, 26, 60, 110, 248, 534, 1140, 2400, 5083, 10692, 22485, 47120, 98596] {1, 1, 2, 4, 8, 17, 33, 66, 132, 265, 529, 1059, 2118, 4238, 8477}
			(120, 60, 10, 4, 22, 21)	14	[3, 0, 4, 0, 11, 0, 26, 0, 60, 0, 104, 13, 227, 28, 478] 1, 0, 1, 0, 2, 0, 4, 0, 8, 0, 16, 2, 30, 4, 59{}
			(40, 20, 10, 4, 102, 101)	21	[3, 0, 0, 4, 0, 0, 11, 0, 0, 26, 0, 0, 60, 0, 0] {1, 0, 0, 1, 0, 0, 2, 0, 0, 4, 0, 0, 8, 0, 0}
	7	301	(101, 40, 20, 10, 204, 202)	8	[3, 4, 11, 26, 60, 145, 247, 583, 1241, 2656, 5658, 11832, 25279, 53078, 111853] {1, 1, 2, 4, 8, 18, 33, 69, 138, 278, 564, 1128, 2285, 4597, 9270}
			(140, 20, 210, 204, 202, 41)	16	[3, 0, 4, 0, 11, 0, 26, 0, 60, 0, 136, 20, 233, 12, 527] {1, 0, 1, 0, 2, 0, 4, 0, 8, 0, 16, 4, 30, 2, 61}
			(340, 260, 10, 4, 202, 241)	24	[3, 0, 0, 4, 0, 0, 11, 0, 0, 26, 0, 0, 60, 0, 0] {1, 0, 0, 1, 0, 0, 2, 0, 0, 4, 0, 0, 8, 0, 0}
128	7	301	(300, 40, 20, 10, 4, 102, 101)	8	[3, 4, 11, 26, 60, 136, 246, 552, 1174, 2484, 5240, 10968, 23019, 48100, 100437] {1, 1, 2, 4, 8, 16, 33, 65, 130, 260, 520, 1041, 2081, 4163, 8326}
			(204, 104, 40, 20, 10, 6, 5)	16	[3, 0, 4, 0, 11, 0, 26, 0, 60, 0, 136, 0, 240, 13, 531] {1, 0, 1, 0, 2, 0, 4, 0, 8, 0, 16, 0, 32, 2, 62}
			(300, 40, 20, 10, 4, 2, 101)	24	[3, 0, 0, 4, 0, 0, 11, 0, 0, 26, 0, 0, 60, 0, 0] {1, 0, 0, 1, 0, 0, 2, 0, 0, 4, 0, 0, 8, 0, 0}

The results in Tables 5-1 and 5-2 are interpreted as follows: for each M-ary alphabet, total encoder memory, v, and feedback polynomial $b(D)$ (presented in octal form), the first line gives the set of feedforward polynomials $\{a_0^{(0)}(D)\, a_1^{(0)}(D) \ldots a_{L-1}^{(0)}(D)\}^{(0)}$ (in octal form), the M-ary free distance, d_{Mfree}, and the first 15 components of the information weight and distance spectrums for rate 1 RSLS convolutional encoders. The second and third lines give the second and third sets of feedforward polynomials needed to form the rate 1/2 and 1/3 RSLS convolutional encoders with their respective spectrums.

It is important to note that the RSLS convolutional encoders reported in Tables 5-1 and 5-2 have slightly inferior IWS compared to the FSLS convolutional encoders reported in Section 4.4.4 – Table 4-1. Therefore it is expected that the RSLS convolutional encoders will have slightly inferior bit error rate (BER) performance compared to the FSLS convolutional encoders previously

introduced. However, as it will be demonstrated in the next sections, this minor decrease in BER performance is compensated by the superior spectral line suppressive characteristics shown by the RSLS convolutional encoders, especially when the data stream is generated by an unbalanced binary Markov source.

Another interesting result is that in general convolutional encoders with feedback polynomials of the form $b(D)=1+D^{v}$ as reported in Table 5-1, have better IWS than convolutional encoders with primitive feedback polynomials as reported in Table 5-2. Nevertheless as it will be shown next, the RSLS convolutional encoders with primitive feedback polynomials have better PSD characteristics than encoders with $b(D)=1+D^{v}$ feedback polynomial.

In the next sections comparisons between RSLS binary to 32-ary convolutional encoders, FSLS binary to 32-ary convolutional encoders and 32-ary PCTH in terms of PSD characteristics and BER performance will be provided.

5.5 Comparisons between RSLS Binary to 32-ary Convolutional Encoders, FSLS Binary to 32-ary Convolutional Encoders and 32-ary PCTH

This section introduces comparisons between PPM TH-IR-based UWB systems employing RSLS binary to 32-ary convolutional encoders, FSLS binary to 32-ary convolutional encoders and 32-ary PCTH in terms of both PSD characteristics and BER performance.

5.5.1 PSD Comparisons of RSLS Binary to 32-ary Convolutional Encoders, FSLS Binary to 32-ary Convolutional Encoders and 32-ary PCTH for Unbalanced First Order BMS Inputs

This section presents PSD examples of PPM TH-IR-based UWB signals obtained when using the 32-ary Bernoulli map PCTH scheme, the rate 1 FSLS binary to 32-ary convolutional encoder introduced in Table 4-1 (Section 4.4.4) with $v=6$, and the rate 1 RSLS binary to 32-ary convolutional encoders reported in Tables 5-1 and 5-2 with $v=6$. Furthermore, PSD examples of a non-coded binary PPM TH-IR-based UWB signal with perfectly random (ideal) 16-ary TH are provided as well for comparison purposes.

The input binary data stream is assumed to be generated by an unbalanced first order BMS for two different combinations of $p_{y,01}$ and $p_{y,10}$:

1). $p_{y,01}=2/5$ and $p_{y,10}=3/5$ thus $\pi_{y,0}=3/5$ and $\pi_{y,1}=2/5$

2). $p_{y,01}=1/5$ and $p_{y,10}=3/5$ thus $\pi_{y,0}=3/4$ and $\pi_{y,1}=1/4$

Note the source's steady state statistics are unbalanced for both cases.

The PSD plots for non-coded binary PPM combined with ideal perfectly random 16-ary TH are shown in Figure 5-5. Remember from Section 4.5 – Figure 4-9b that spectral lines appear at 2.56 GHz intervals when ideal perfectly random 16-ary TH is assumed and the data stream consists of i.i.d. binary symbols with uniform distribution. However in Figure 5-5 it can be seen that when the source's stationary distribution is not uniform like the number of spectral lines in the PSD is doubled (the spectral lines appear at 1.28 GHz intervals in Figures 5-5a and 5-5b). It is worth pointing out that if a periodic pseudo-random (PR) sequence had been used instead of an ideal one, then the PSD would have shown a larger number of spectral lines (see Section 4.5–Figure 4-9a).

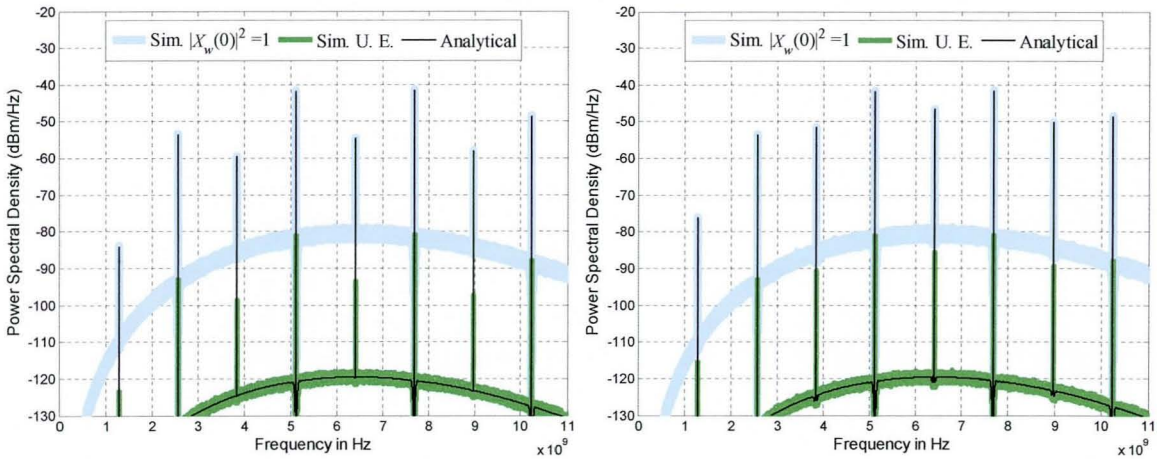
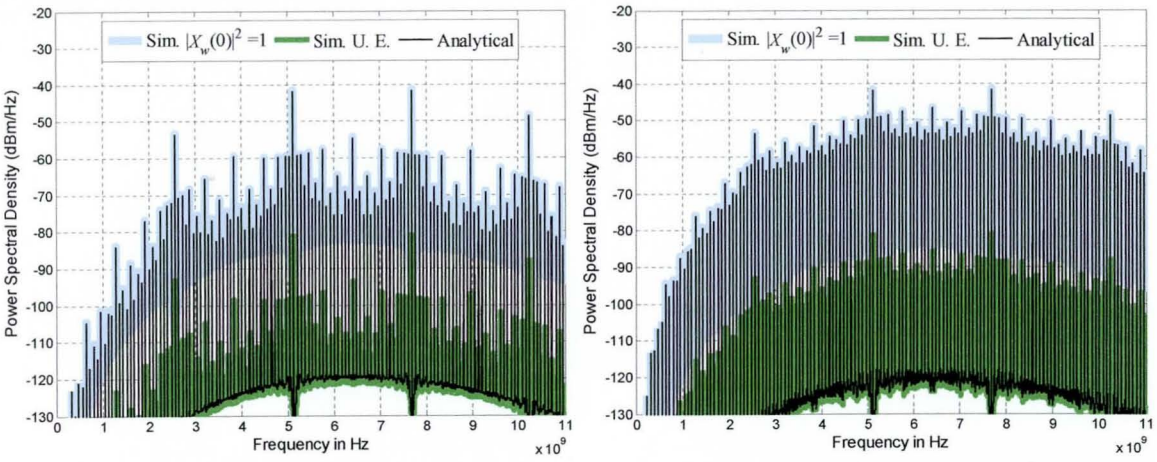


Figure 5-5. Analytical and simulated PSDs for a TH-IR-based UWB system using binary PPM with ideal perfectly random 16-ary TH. The signal parameters are $N_w = 1$, $T_s = 12.5$ ns, $T_r = 12.5$ ns, $T_c = T_r/16$, $T_\beta = T_r/32$ ns and c_{lNw+k} uniform distributed over the set $\{0, 1, \ldots, 15\}$. The 3rd derivative Gaussian pulse was used with duration $T_w \approx 0.35$ ns.

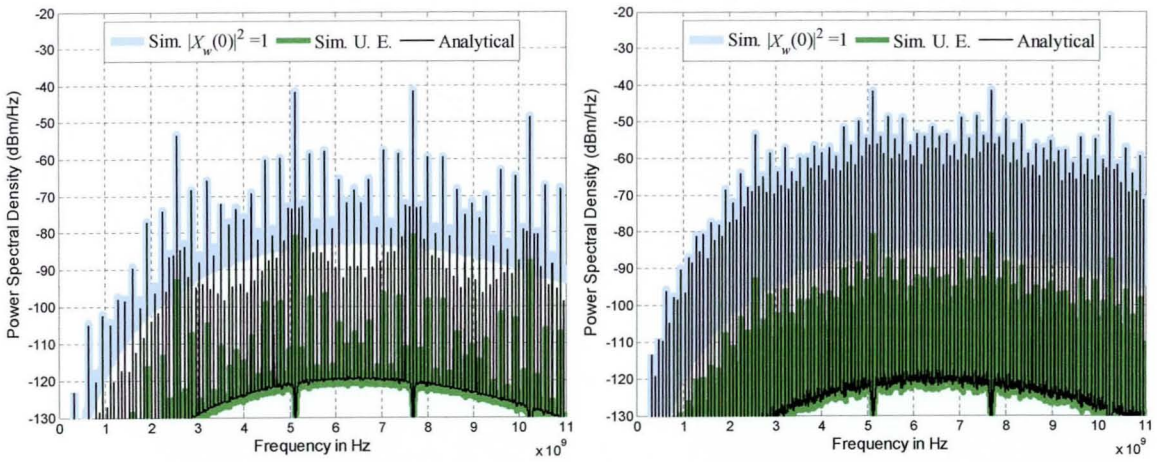
The PSD plots obtained when using the 32-ary Bernoulli shift map PCTH encoder and the rate 1, $v = 6$, FSLS binary to 32-ary convolutional encoder from Table 4-1 (Section 4.4.4) are shown in Figures 5-6 and 5-7 respectively. Remember from Section 4.5 – Figures 4-11 and 4-12 that spectral lines appear at 2.56 GHz intervals when the data input to these encoders consists of a uniform distributed i.i.d. binary stream. Nevertheless, in Figures 5-6 and 5-7 spectral lines appear at 80 MHz intervals. Thus, fewer spectral lines are eliminated by these encoders when the source generates a non-uniform distributed binary stream. Furthermore, it can be seen that the number of spectral lines generated by these encoders is larger than the number of spectral lines produced by the system employing ideal perfectly random 16-ary TH for both combinations of BMS' one step transition probabilities.

To conclude Figures 5-8 and 5-9 show that the best PSD characteristics are obtained when using the rate 1, $v = 6$, RSLS binary to 32-ary convolutional encoders introduced in Tables 5-1 and 5-2. It can be seen in Figures 5-8 and 5-9 that these encoders eliminate the same amount of spectral lines regardless of the values acquired by the BMS' one step transition probabilities. In fact the spectral lines in these figures are spaced at 2.56 GHz intervals. This is the same spectral line spacing achieved when assuming ideal perfectly random 16-ary TH is used in a non-coded binary PPM system with uniform distributed i.i.d. binary data streams (Section 4.5 – Figure 4-9b).



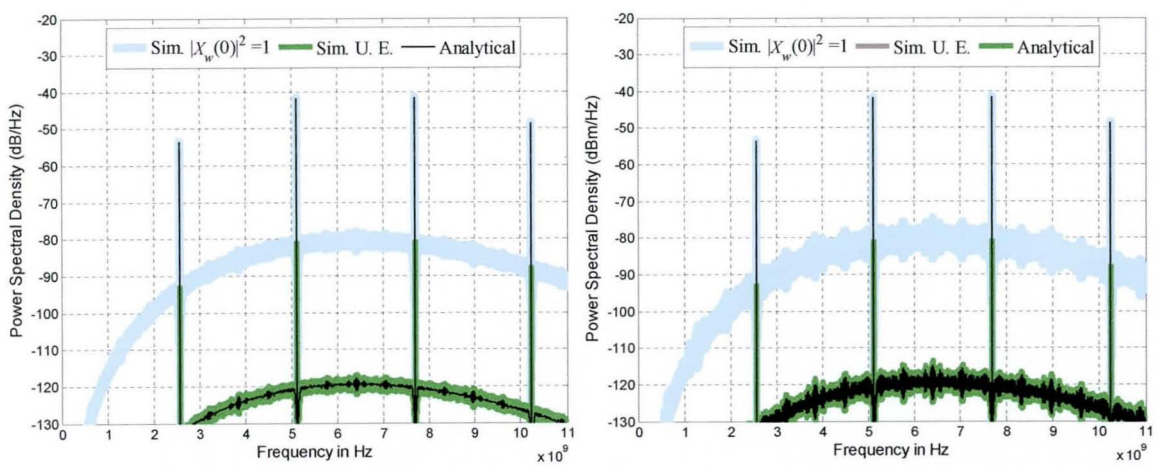
a) $p_{y,01} = 2/5$ and $p_{y,10} = 3/5$ b) $p_{y,01} = 1/5$ and $p_{y,10} = 3/5$

Figure 5-6. Analytical and simulated PSDs for a TH-IR-based UWB system using the 32-ary Bernoulli shift map PCTH scheme. The signal parameters are $N_w = 1$, $T_s = 12.5$ ns, $T_r = 12.5$ ns and $T_\beta = T_r/32$. The 3rd derivative Gaussian pulse was used with duration $T_w \approx 0.35$ ns.



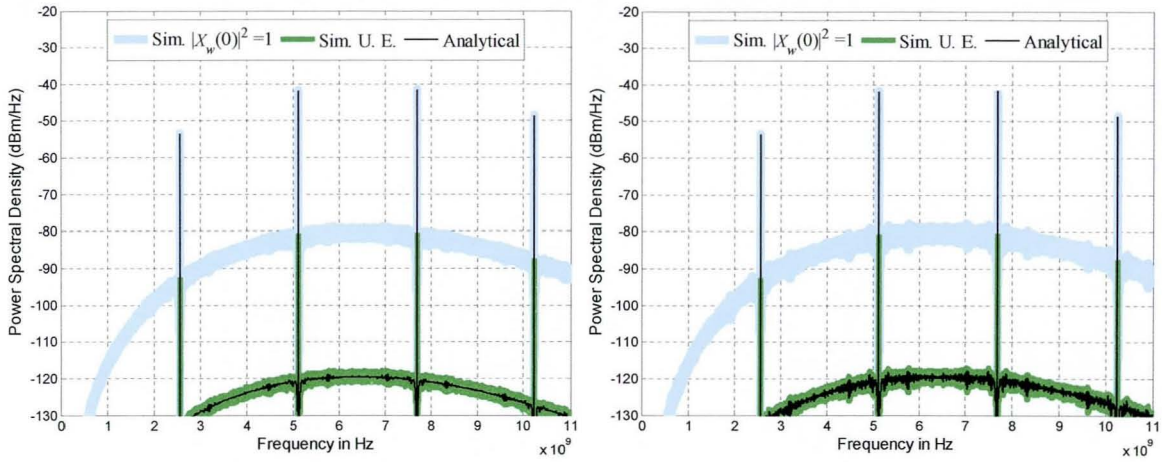
a) $p_{y,01} = 2/5$ and $p_{y,10} = 3/5$ b) $p_{y,01} = 1/5$ and $p_{y,10} = 3/5$

Figure 5-7. Analytical and simulated PSDs for a TH-IR-based UWB system using the rate 1, $v = 6$, FSLS binary to 32-ary convolutional encoder from Table 4-1 (Section 4.4.4). The signal parameters are $N_w = 1$, $T_s = 12.5$ ns, $T_r = 12.5$ ns and $T_\beta = T_r/32$. The 3rd derivative Gaussian pulse was used with duration $T_w \approx 0.35$ ns.



a) $p_{y,01} = 2/5$ and $p_{y,10} = 3/5$ b) $p_{y,01} = 1/5$ and $p_{y,10} = 3/5$

Figure 5-8. Analytical and simulated PSDs for a TH-IR-based UWB system using the rate 1, $v = 6$, RSLS binary to 32-ary convolutional encoder from Table 5-1 with feedback polynomial $b(D) = 1 + D^6 \Rightarrow 101_8$. The signal parameters are $N_w = 1$, $T_s = 12.5$ ns, $T_r = 12.5$ ns and $T_\beta = 0.39$ ns. The 3rd derivative Gaussian pulse was used with duration $T_w \approx 0.35$ ns.



a) $p_{y,01}=2/5$ and $p_{y,10}=3/5$ b) $p_{y,01}=1/5$ and $p_{y,10}=3/5$

Figure 5-9. Analytical and simulated PSDs for a TH-IR-based UWB system using the rate 1, $v=6$, RSLS binary to 32-ary convolutional encoder from Table 5-2 with primitive feedback polynomial $b(D)=1+\mathrm{D}+D^6 \Rightarrow 141_8$. The signal parameters are $N_w=1$, $T_s=12.5$ ns, $T_r=12.5$ ns and $T_\beta=0.39$ ns. The 3rd derivative Gaussian pulse was used with duration $T_w \approx 0.35$ ns.

By comparing Figures 5-8 and 5-9 it can be seen that even though the RSLS binary to 32-ary convolutional encoder with $b(D)=1+D^6$ feedback polynomial and the RSLS binary to 32-ary convolutional encoder with $b(D)=1+D+D^6$ primitive feedback polynomial eliminate the same amount of spectral lines, there exist noticeable differences on the PSD's continuous part. Therefore the RSLS convolutional encoders with primitive feedback polynomials reported in Table 5-2 must be used if getting a continuous PSD component as smooth as possible is required.

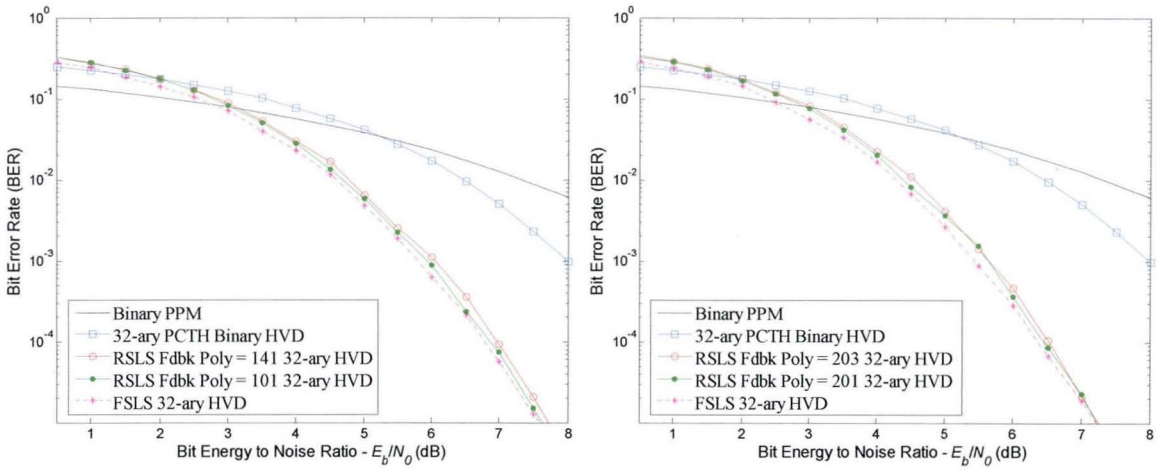
Finally note how there is good agreement between the analytical and simulated results for both the PSD's continuous part (Sim. U.E.) and the PSD's discrete part (Sim. $|X_w(0)|^2=1$) in the all the figures presented in this section.

5.5.2 BER Comparisons of RSLS Binary to 32-ary Convolutional Encoders, FSLS Binary to 32-ary Convolutional Encoders and 32-ary PCTH

In this section comparisons in terms of bit error rate (BER) performance between RSLS and FSLS binary to 32-ary convolutional encoders in additive white Gaussian noise (AWGN) channel are provided for hard Viterbi decoding (HVD) and soft Viterbi decoding (SVD). The implementation particularities for each system are described in Section 4-6 and therefore only results in terms of BER plots will be presented in this section.

Simulated BER plots for a PPM IR-based UWB system employing rate 1 RSLS (Tables 5-1 and 5-2) and FSLS (Table 4-1) binary to 32-ary convolutional encoders are shown in Figure 5-10 for *M*-ary Hamming distance based HVD. From this figure it can be seen that for HVD the BER performance loss of both RSLS convolutional encoders is negligible compared to the BER performance of the FSLS convolutional encoders introduced in Chapter 4. Furthermore, the BER performance of RSLS convolutional encoders with primitive feedback polynomials (Fdbk Poly = 141

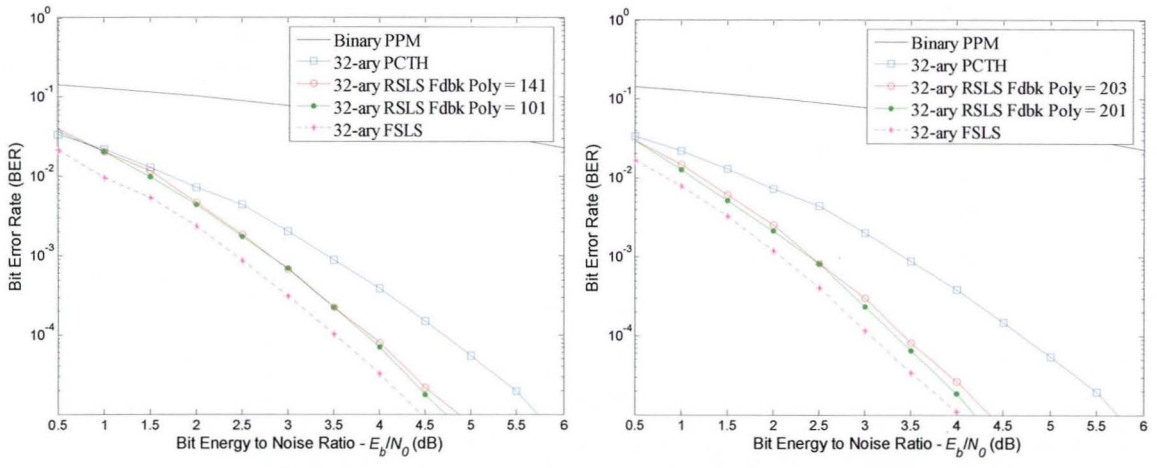
and 203 – see Table 5-2) is nearly identical to the BER performance achieved with RSLS convolutional encoders with feedback polynomials of the form $b(D)=1+D^{v}$ (Fdbk Poly = 101 and 201 – see Table 5-1).



a) $v=6$ b) $v=7$

Figure 5-10. Bit error rate versus bit energy to noise ratio for hard Viterbi decoding (HVD) in PPM TH-IR-based UWB systems using 32-ary PCTH, the rate 1 RSLS binary to 32-ary convolutional encoders presented in Tables 5-1 (Fdbk Poly = 101 and 201) and 5-2 (Fdbk Poly = 141 and 203 – primitive polynomials), and the rate 1 FSLS binary to 32-ary convolutional encoders introduced in Table 4-1 (Section 4.4.4). The BER plot for binary orthogonal PPM is provided as a reference. All the plots were obtained by simulation for the AWGN channel. The total encoder memory, v, of each code is indicated in the legend.

Simulated BER plots for a PPM IR-based UWB system employing rate 1 RSLS (Tables 5-1 and 5-2) and FSLS (Table 4-1) binary to 32-ary convolutional encoders are shown in Figure 5-11 for SVD. In this case the performance loss when using RSLS convolutional encoders is not as negligible as for HVD. Nevertheless this performance loss can still be considered to be acceptable as the bit energy to noise ratio difference between the FSLS convolutional encoders and both RSLS convolutional encoders is less than 0.5 dB for BER values below 10^{-2}. Finally note that the BER performance of both RSLS convolutional encoders is nearly identical for SVD as in the HVD case.



a) $v=6$ $v=7$

Figure 5-11. Bit error rate versus bit energy to noise ratio for soft Viterbi decoding (SVD) in PPM TH-IR-based UWB systems using 32-ary PCTH, the rate 1 RSLS binary to 32-ary convolutional encoders presented in Tables 5-1 (Fdbk Poly = 101 and 201) and 5-2 (Fdbk Poly = 141 and 203 – primitive polynomials), and the rate 1 FSLS binary to 32-ary convolutional encoders presented in Table 4-1 (Section 4.4.4). The BER plot for binary orthogonal PPM is provided as a reference. All the plots were obtained by simulation for the AWGN channel. The total encoder memory, v, of each code is indicated in the legend.

5.6 Conclusions for Chapter 5

In this chapter recursive spectral line suppressive (RSLS) binary to M-ary convolutional encoders for unbalanced (non-uniform distributed) binary Markov sources (BMS) have been introduced. First it was shown that the traditional feedforward structures do not provide a good framework for the construction of SLS encoders when the binary data stream is generated by an unbalanced first order BMS. Next it was shown that by introducing recursive structures a better framework is provided, as the distribution acquired by the steady state probabilities of the source encoder (SE) pair Markov model only depends on the BMS's steady state probabilities.

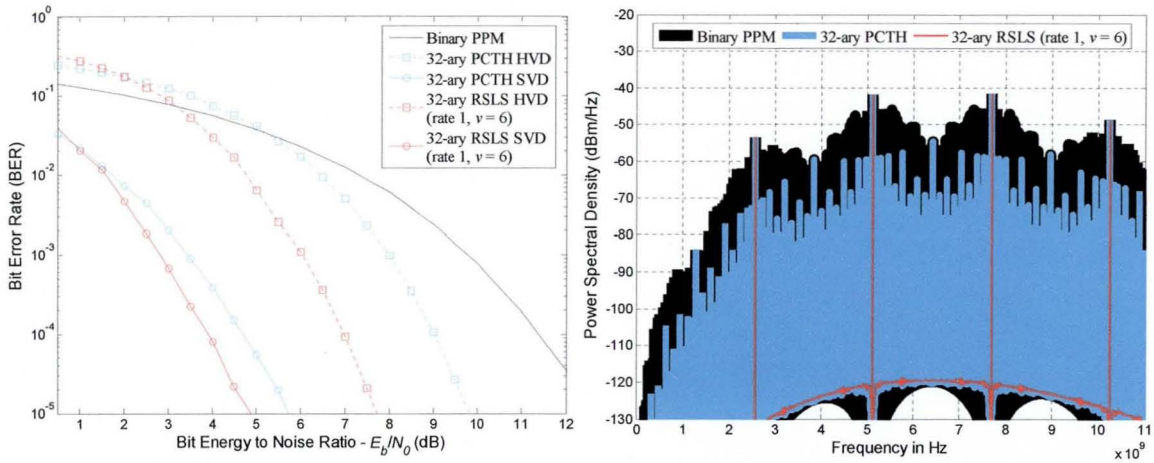
It was shown analytically that binary to M-ary convolutional encoders based on feedforward structures do not satisfy the spectral line suppression condition for all combinations of BMS' one step transition probabilities, $0 < p_{y,ij} < 1$. It was also demonstrated that by using binary to M-ary encoders with recursive structures the spectral line suppression condition can be satisfied for all combinations of $0 < p_{y,ij} < 1$ as far as the output mapping function, $\gamma(\cdot)$, is properly designed. Based on this analysis a code search procedure for rate 1, 1/2 and 1/3 RSLS binary to M-ary convolutional encoders with the best possible information weight spectrum (IWS) was introduced. The code search procedure verifies that the spectral line suppression condition is fulfilled for unbalanced first order BMS's. Using this code search procedure maximum free distance RSLS binary to M-ary convolutional encoders have been found. Two tables with the best RSLS convolutional encoders found for 16-ary, 32-ary, 64-ary and 128-ary orthogonal signalling have been provided: one for recursive structures with $b(D) = 1 + D^v$ feedback polynomials and another for recursive structures with primitive feedback polynomials. These encoders are able to eliminate as many spectral lines as PCTH and the FSLS binary to M-ary convolutional encoders introduced in Chapter 4 even when the data stream at the encoder input consists of binary symbols generated by an unbalanced first order BMS.

Power spectral density examples of PPM TH-IR-based UWB signals using RSLS and FSLS binary to 32-ary convolutional encoders, 32-ary PCTH and binary PPM with ideal perfectly random 16-ary TH have been provided. These examples showed that compared to all the other schemes the RSLS encoders offer superior PSD characteristics when the data stream is generated by an unbalanced first order BMS. Moreover, based on these examples it can be concluded that if the main constraint in the design of spectral line suppressive convolutionally coded PPM TH-IR-based UWB systems is maximising the number of spectral lines suppressed while keeping the continuous PSD component as smooth as possible, then the RSLS convolutional encoders with primitive feedback polynomials reported in Table 5-2 must be preferred over the encoders with $b(D) = 1 + D^v$ feedback polynomials reported in Table 5-1.

Lastly comparisons of the bit error rate (BER) performance achievable with RSLS and FSLS binary to 32-ary convolutional encoders for hard and soft decision Viterbi decoding were

introduced. The BER plots presented showed that the RSLS binary to 32-ary convolutional encoders provide negligible performance loss compared to equivalent FSLS binary to 32-ary convolutional encoders for hard Viterbi decoding. Furthermore, these plots showed that for SVD the performance loss is limited to no more than 0.5 dB for bit error rates below 10^{-2}. Nevertheless, this slight BER performance loss is compensated by the superior PSD characteristics achieved with the RSLS binary to *M*-ary convolutional encoders when the data streams are generated by an unbalanced binary Markov source.

Finally it is worth mentioning that the complexity of the RSLS binary to *M*-ary convolutional encoders introduced in this chapter is equivalent to the complexity of the FSLS binary to *M*-ary convolutional encoders introduced in Chapter 4 (as far a both encoders have the same total encoder memory). Moreover, similar to the FSLS binary to *M*-ary convolutional encoders, the use of the RSLS binary to *M*-ary convolutional encoders is not restricted to orthogonal *M*-ary PPM as they can be used in any system employing *M*-ary orthogonal signalling.



a) BER performance comparison b) PSD shape comparison

Figure 5-12. Side by side BER performance and PSD shape (analytical) comparisons between PPM TH-IR-based UWB systems employing: the rate 1, $v = 6$, binary to 32-ary FSLS convolutional encoder from Table 5-2 (primitive feedback polynomial: $b(D) = 1 + D + D^6 \Rightarrow 141_8$), the 32-ary Bernoulli shift map PCTH scheme and non-coded orthogonal binary PPM with pseudo-random 16-ary TH. The input binary data stream is assumed to be generated by an unbalanced BMS with $p_{y,01} = 2/5$ and $p_{y,01} = 3/5$. The periodic sequence {0, 14, 1, 5, 13, 6, 3, 15, 7, 11, 8, 12, 9, 2, 10, 4} was used for TH in the system employing non-coded orthogonal binary PPM. The BER performance plots were obtained by simulation for soft Viterbi decoding (SVD) and hard Viterbi decoding (HVD) in the AWGN channel. For the PCTH scheme HVD was performed by using branch and path metrics based on binary Hamming distances. The signal parameters are $N_w = 1$, $T_s = 12.5$ ns, $T_r = 12.5$ ns and $T_\beta = T_r/32$ ($T_c = T_r/16$ for the system employing pseudo-random 16-ary TH) . The 3rd derivative Gaussian pulse was used with duration $T_w \approx 0.35$ ns.

Chapter 6

Maximum Free Distance Spectral Line Free Binary Convolutional Encoders for BPSK/Q-BOPPM IR-Based UWB Systems with Unbalanced Binary Markov Sources

6.1 Introduction

In this chapter new binary convolutional encoders with maximum free distance (MFD) are introduced. The main characteristic of this set of new convolutional encoders is that they generate a spectral line-free (SLF) signal when used in binary phase shift keying (BPSK) and quaternary biorthogonal pulse position modulation (Q-BOPPM) impulse radio (IR) based ultra wideband (UWB) systems. Furthermore, *the power spectral density (PSD) of the signals driven by these encoders is spectral line-free even when the binary stream at the encoder input does not consist of independent identically distributed (i.i.d.) symbols with uniform distribution.*

The binary convolutional encoders introduced in this chapter are based on recursive non-systematic (RNS) structures and were specifically designed to generate signals with spectral line-free PSDs even when the data stream is generated by an unbalanced first order binary Markov source (BMS). These recursive spectral line-free (RSLF) convolutional encoders provide enhanced PSD characteristics when used in BPSK/Q-BOPPM IR-based UWB systems compared to the non-coded pulse repetition scheme and traditional feedforward and recursive-systematic binary convolutional encoders.

In common with the recursive spectral line suppressive (RSLS) binary to M-ary convolutional encoders introduced Chapter 5, the RSLF binary convolutional encoders introduced in this chapter have a wide applicability range, as the first order BMS can be used to model both unbalanced (that is non-uniform distributed) i.i.d. memoryless binary data streams and binary data streams with memory, [116-118].

As introduced in Sections 1.4 and 2.6, the application of binary convolutional and related codes (for example turbo codes) in binary IR-based UWB systems has been previously addressed in [21, 23-30, 32]. However, these works do not consider the effects of the convolutional coding operation over the signal's PSD and focus their analysis on the improvement of the system's BER performance. Therefore the idea of using binary convolutional encoders to achieve both a spectral line-free PSD and improved BER performance in IR-based UWB systems has not been previously addressed.

As previously explained in Sections 1.4.2 and 2.5.3 the main approach proposed for the elimination of the PSD's spectral lines is the polarity randomisation technique, [78-81]. This technique requires the multiplication of each transmitted symbol (symbol based polarity randomisation) or pulse (pulse based polarity randomisation) by an ideal perfectly random direct sequence (DS). In practice it is assumed that this ideal case can be approximated by the use of a pseudo-random (PR) DS code with an extremely long period, [81]. Nevertheless, as it will be shown later in this chapter, when the length of the PR-DS code is relatively short the height of the spectral lines may be reduced but the actual number of spectral lines in the PSD may be simultaneously increased. Although the analysis presented in this chapter focuses in single user scenarios, it is worth mentioning that in systems employing time hopping (TH) – code division multiple access (CDMA) the polarity randomisation technique may help to improve the system's BER performance in the presence of multiple access interference (MAI), compared to TH-IR-based UWB systems where no DS multiplication is used, [92]. However, in single user scenarios the polarity randomisation technique does not necessarily provide significant BER performance improvements when compared with TH-IR-based UWB systems in which no DS multiplication is used.

The RSLF binary convolutional encoders introduced in this chapter provide a novel alternative method for spectral line elimination when the input symbols are generated by an unbalanced first order BMS. It will be shown that these encoders provide a spectral line-free PSD while simultaneously improving the system's BER performance for single user scenarios.

It is worth mentioning that the use of convolutional codes for spectral shaping purposes with error correction capabilities has been previously addressed in [144-150]. However the main objective in all these papers is to obtain a DC-free signal which is a different goal from the one pursued in this chapter. The schemes proposed in these papers are based on feedforward structures which are modified through the addition of specific mechanisms that increase both the encoder and decoder complexity compared to normal convolutionally coded schemes.

This chapter is structured as follows. Section 6.2 describes the system model. Section 6.3 introduces the signal model and its respective PSD. Section 6.4 shows that a recursive non-systematic structure for binary convolutional encoders enables the design of spectral line-free encoders. The code search procedure is introduced in Section 6.5 altogether with the RSLF binary convolutional encoders found with such procedure. Comparisons in terms of PSD characteristics between the RSLF binary convolutional encoders, the best binary convolutional encoders known and a rate equivalent non-coded scheme are presented in Section 6.6 while comparisons in terms of

BER performance are provided in Section 6.7. Finally conclusions are presented in Section 6.8.

6.2 System Model for Binary Convolutionally Coded BPSK/QBOPPM IR-Based UWB Signals with First Order Binary Markov Sources

The block diagram of the system model assumed in this chapter is shown in Figure 6-1. This model assumes the data source to be the first order binary Markov Source (BMS) introduced in Section 3.2.1. However, unlike Chapters 4 and 5 this time the analysed convolutional encoder is set to be binary and the modulation format is assumed to be BPSK or Q-BOPPM (with natural or Gray based mapping).

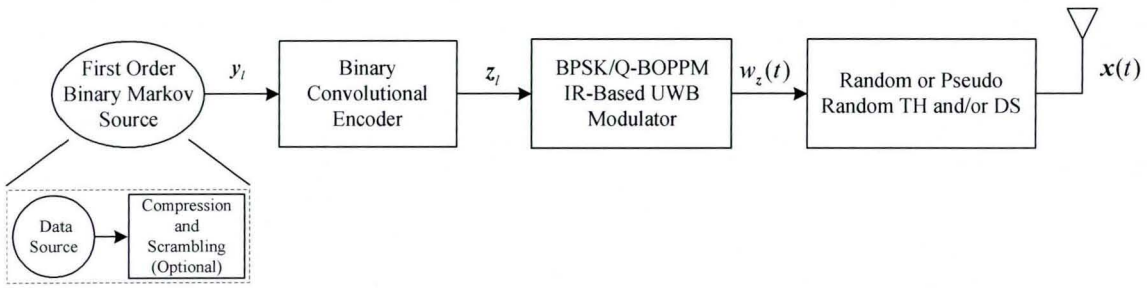


Figure 6-1. Block diagram of the binary convolutionally coded BPSK/Q-BOPPM IR-based UWB system with first order binary Markov Source (BMS).

Recollect that this BMS source model is defined by the state set $\mathcal{R} = \{r_0, r_1\}$ with corresponding outputs $\{\iota_0, \iota_1\} = \{0,1\}$ and the one step transition probabilities matrix defined by

$$\mathbf{P}_y = \begin{bmatrix} p_{y,00} & p_{y,01} \\ p_{y,10} & p_{y,11} \end{bmatrix} \tag{6.1}$$

with corresponding steady state probabilities given by

$$\underline{\boldsymbol{\pi}_y} = [\pi_{y,0} \;\; \pi_{y,1}] = \left[\frac{p_{y,10}}{p_{y,01}+p_{y,10}} \;\; \frac{p_{y,01}}{p_{y,01}+p_{y,10}} \right] \tag{6.2}$$

where $\pi_{y,0}$ and $\pi_{y,1}$ are the steady state probabilities of generating a "0" and a "1" respectively. As in the previous chapters it will be assumed that

$$0 < p_{y,ij} < 1 \tag{6.3}$$

in order to avoid trivial degenerate cases.

Remember from Section 3.4 that for convolutionally coded/Markov driven IR-based UWB signals the discrete part of the PSD is a function of the steady state probabilities of the source encoder (SE) pair Markov model. Recall as well that the state process, σ_l, defined by the SE pair Markov model has state set defined by

$$\mathcal{S} = \{(0,q_0),(0,q_1),...,(0,q_{Nq-1}),(1,q_0),(1,q_1),...,(1,q_{Nq-1})\} = \{s_0, s_1, ..., s_{2Nq-1}\} \tag{6.4}$$

with corresponding transition probabilities matrix given by

$$\mathbf{P}_{\sigma} = \begin{bmatrix} p_{y,00}\mathbf{B}_0 & p_{y,01}\mathbf{B}_0 \\ p_{y,10}\mathbf{B}_1 & p_{y,11}\mathbf{B}_1 \end{bmatrix} \tag{6.5}$$

where $\mathbf{B}_0$ and $\mathbf{B}_1$ are the $N_q \times N_q$ the next state matrices defined in Section 3.2.3 and $N_q = 2^v$ is the number of encoder states, q_l, as defined by the encoder's state transition diagram representation (see Sections 2.7.2 and 3.2.2). Furthermore, remember that the output mapping is defined as

$$\underline{\zeta_{s_{nN_q+j}}} = [\zeta^{(0)}_{s_{nN_q+j}}, \zeta^{(1)}_{s_{nN_q+j}}, \ldots, \zeta^{(\kappa-1)}_{s_{nN_q+j}}] = \gamma(s_{nN_q+j}) = \gamma(n, q_j) \tag{6.6}$$

where $\gamma(\cdot)$ is the output mapping function, $n = 0,1$ and $j = 0,1,\ldots,N_q - 1$.

As in Section 5.2, it is convenient to analyse separately the steady state behaviour of the SE pair Markov model when feedforward and feedback structures are used to construct binary convolutional encoders.

6.2.1 Source-Encoder pair Markov Model Behaviour for Rate $1/\kappa$ Feedforward Binary Convolutional Encoders

Consider the feedforward structure for rate $1/\kappa$ binary convolutional encoders shown in Figure 6-2. Note how the base shift register structure shown in this figure is essentially the same as that introduced in Section 5.2.1 – Figure 5-2 for feedforward binary to *M*-ary convolutional encoders. The main difference between both encoders resides in their output mapping function, $\gamma(\cdot)$, owing to the fact that in Figure 6-2 the output consists of binary symbols while in Figure 5-2 the output consists of *M*-ary symbols.

The output mapping is defined by the encoder's particular generator polynomials and it is a function of the current SE pair Markov chain (MC) state (see Section 3.2.3 and Equation (6.6)). Furthermore, remember from Section 3.2.3 that the SE pair output process, z_l, is said to be a projection of the MC state process, σ_l, that is

$$z_l = \gamma(\sigma_l = s_i) \tag{6.7}$$

Therefore by definition a change in the output mapping function, $\gamma(\cdot)$, does not change the MC state process, σ_l, of the SE pair Markov model. This means that the state process, σ_l, of feedforward binary convolutional encoders and feedforward binary to *M*-ary convolutional encoders with the same base shift register structure are equivalent. As a consequence the SE pair Markov model analysis presented in Section 5.2.1 for feedforward binary to *M*-ary convolutional encoders applies to binary convolutional encoders whose base structure is equivalent to that shown in Figure 6-2.

Therefore, based on the previous discussion and according to Theorem 5.1, for feedforward encoders with base structure as shown in Figure 6-2 the steady state probabilities of the SE pair Markov model state process, σ_l, are given by

$$\pi_{\sigma,nN_q+j} = \pi_{\sigma,(n,q_j)} = \pi_{\sigma,s_{nN_q+j}} = p_{y,\mu_1 n} p_{y,\mu_2\mu_1} \cdots p_{y,\mu_{v-1}\mu_{v-2}} p_{y,\mu_v\mu_{v-1}} \pi_{y,\mu_v} \tag{6.8}$$

where $n=0,1$, $\pi_{y,t}$ and $p_{y,tt'}$ are BMS' steady state and one step transition probabilities, and $[\mu_1,\mu_2,...,\mu_v]$ is the encoder's memory content defining the encoder state q_j with $j=\mu_1 2^{v-1}+...+\mu_v$.

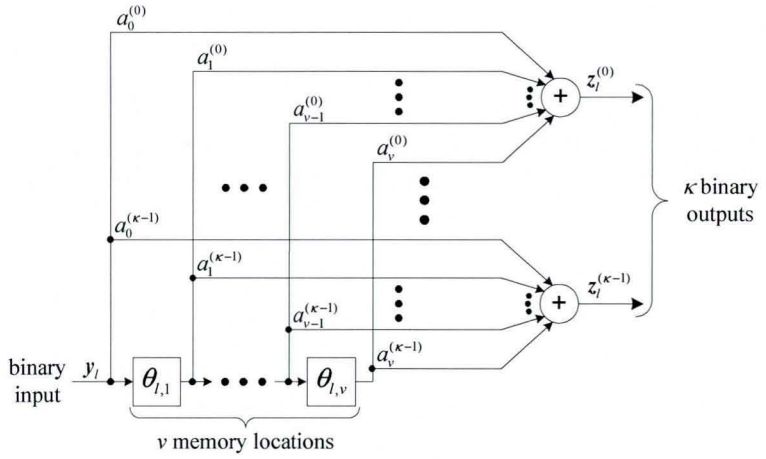


Figure 6-2. Generic diagram of a rate $1/\kappa$ feedforward binary convolutional encoder.

6.2.2 Source-Encoder pair Markov Model Behaviour for Rate $1/\kappa$ Recursive Binary Convolutional Encoders

Now consider the recursive structure for rate $1/\kappa$ binary convolutional encoders shown in Figure 6-3. As in Section 5.2.2 the feedback coefficients, b_i, can be freely set to 0 or 1 with the exception of the last one which is set to 1, that is $b_v=1$.

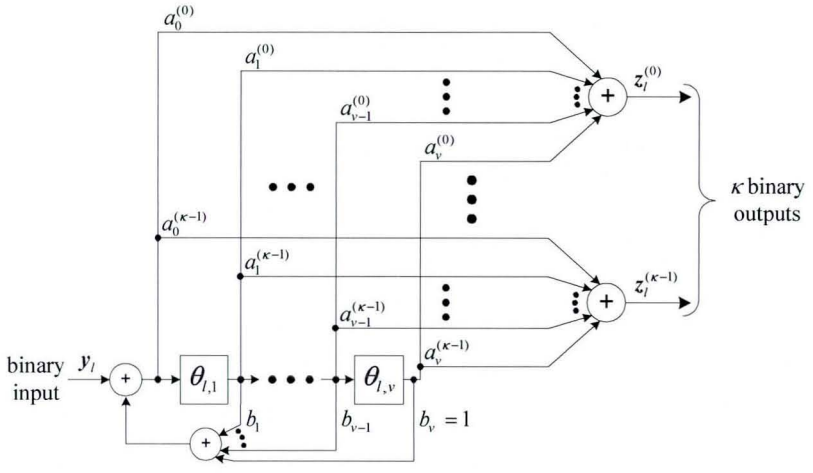


Figure 6-3. Generic diagram of a rate $1/\kappa$ recursive binary convolutional encoder.

Similar to the previous case, the base recursive structure used in Section 5.2.1 – Figure 5-2 for recursive binary to M-ary convolutional encoders is equivalent to the recursive structure introduced in Figure 6-2 for rate $1/\kappa$ recursive binary convolutional encoders. The main difference between both encoders resides in the output mapping function, $\gamma(\cdot)$. Therefore Theorem 5.2 applies as well for recursive binary convolutional encoders with base structure as shown in Figure 6-2. This means

that SE pair Markov model state process, σ_l, for this kind of recursive convolutional encoders has steady state probabilities given by

$$\pi_{\sigma,nN_q+j} = \pi_{\sigma,(n,q_j)} = \pi_{\sigma,s_{nN_q+j}} = \frac{\pi_{y,n}}{N_q} \tag{6.9}$$

where $n = 0,1$, $\pi_{y,n}$ is the BMS' steady state probability corresponding to state r_n with respective output n, and $j = \mu_1 2^{v-1} + ... + \mu_v$ is defined by the encoder's memory content $[\mu_1, \mu_2, ..., \mu_v]$.

The next section introduces the corresponding signal model and PSD obtained when using these binary convolutional encoders in BPSK/Q-BOPPM IR-based UWB systems.

6.3 Signal Model and Power Spectral Density of the Binary Convolutionally coded Scheme for BPSK/Q-BOPPM IR-Based UWB

In this section the signal model of the binary convolutionally coded BPSK/Q-BOPPM IR-based UWB scheme is introduced. The power spectral density of the signal model is then provided as obtained when using the results from Section 3.4. Finally the required conditions to obtain a spectral line-free PSD through the convolutional encoding operation are derived.

6.3.1 Signal Model for the Binary Convolutionally coded BPSK/Q-BOPPM IR-Based UWB Scheme

The signal model for the binary convolutionally coded BPSK/Q-BOPPM IR-based UWB scheme analysed in this chapter is defined as

$$x(t) = \sum_{l=-\infty}^{\infty} \sum_{k=0}^{N_w-1} a_{lN_w+k} \alpha_{\sigma_l,k} w(t - \beta_{\sigma_l,k} T_\beta - lT_s - kT_r - c_{lN_w+k} T_c) \tag{6.10}$$

where σ_l is the SE pair Markov model state process with $N_\sigma = 2N_q$ states; $\alpha_{i,k} \in \{-1,1\}$ and $\beta_{i,k} \in \{0,1\}$ are the k^{th} PAM and PPM symbols used for the transmission of the lth encoder output vector, $z_l = \gamma(\sigma_l)$, whose value depends on the SE pair MC state at time l; N_w is the number of pulses used per output vector; T_β is the PPM modulation index; T_r is the mean repetition time between pulses; $T_s = N_w T_r$ is the encoder's output vector time; $\{a_{lN_w+k}\}$ is a pseudo-random (PR)-DS sequence taking values on the set {-1, 1} with period χ_a; $\{c_{lN_w+k}\}$ is a PR-TH sequence taking values on the set {0, 1, 2 ,..., $N_c - 1$} with period χ_c and T_c is the nominal shift caused by the TH sequence. If BPSK is used then $\alpha_{i,k} \in \{-1,1\}$, $\beta_{i,k} = 0$ and $T_\beta = 0$. In contrast for Q-BOPPM $\alpha_{i,k} \in \{-1,1\}$, $\beta_{i,k} \in \{0,1\}$ and $T_\beta \neq 0$.

Assume that the rate $1/\kappa$ binary convolutional encoder is in state $\sigma_l = s_i$ with corresponding

encoder output vector $\mathbf{z}_l = \underline{\mathbf{z}_l} = [z_l^{(0)}, z_l^{(1)}, ..., z_l^{(\kappa-1)}] = \gamma(\sigma_l = s_i) = \underline{\zeta_{s_i}} = [\zeta_{s_i}^{(0)}, \zeta_{s_i}^{(1)}, ..., \zeta_{s_i}^{(\kappa-1)}]$. If BPSK is used then $N_w = \kappa$ and

$$\alpha_{\sigma_l,k} = (2z_l^{(k)} - 1) = (2\zeta_{s_i}^{(k)} - 1) = \alpha_{i,k} \quad \text{and} \quad \beta_{\sigma_l,k} = 0, \qquad k = 0,1,...,N_w - 1 \tag{6.11}$$

where $\alpha_{i,k} \in \{-1,1\}$. If Q-BOPPM with natural mapping is used then $N_w = \kappa/2$ and thus

$$\alpha_{\sigma_l,k} = (2z_l^{(2k)} - 1) = (2\zeta_{s_i}^{(2k)} - 1) = \alpha_{i,k} \quad \text{and} \quad \beta_{\sigma_l,k} = z_l^{(2k+1)} = \zeta_{s_i}^{(2k+1)} = \beta_{i,k}, \qquad k = 0,1,...,N_w - 1 \tag{6.12}$$

where $\alpha_{i,k} \in \{-1,1\}$ and $\beta_{i,k} \in \{0,1\}$. Lastly, if Gray mapping based Q-BOPPM is used then

$$\begin{aligned} &[\zeta_{s_i}^{(2k)}, \zeta_{s_i}^{(2k+1)}] = [0,0] \Rightarrow (\alpha_{i,k}, \beta_{i,k}) = (-1,0); \quad [\zeta_{s_i}^{(2k)}, \zeta_{s_i}^{(2k+1)}] = [0,1] \Rightarrow (\alpha_{i,k}, \beta_{i,k}) = (-1,1); \\ &[\zeta_{s_i}^{(2k)}, \zeta_{s_i}^{(2k+1)}] = [1,0] \Rightarrow (\alpha_{i,k}, \beta_{i,k}) = (+1,1); \quad [\zeta_{s_i}^{(2k)}, \zeta_{s_i}^{(2k+1)}] = [1,1] \Rightarrow (\alpha_{i,k}, \beta_{i,k}) = (+1,0); \end{aligned} \tag{6.13}$$

Note how the code bits to signal assignment is different for Q-BOPPM with natural mapping and Q-BOPPM with Gray mapping.

6.3.2 Power Spectral Density of the Binary Convolutionally coded BPSK/Q-BOPPM IR-Based UWB Scheme

Using the results presented in Section 3.4 the PSD of the signal described by Equation (6.10) can be found to be

$$\overline{S}(f) = S_C(f) + S_D(f) \tag{6.14}$$

$$\begin{aligned} S_C(f) &= S_{C.1}(f) + S_{C.3}(f) \qquad N_w = 1 \\ S_C(f) &= S_{C.1}(f) + S_{C.2}(f) + S_{C.3}(f) \qquad N_w > 1 \end{aligned} \tag{6.15}$$

$$S_{C.1}(f) = \frac{1}{T_r}|W(f)|^2 - \frac{|W(f)|^2}{T_s\Lambda}\sum_{n=0}^{\Lambda-1}\left|\sum_{k=0}^{N_w-1}\sum_{i=0}^{N_\sigma-1} e^{j2\pi fkT_r} a_{nN_w+k} e^{j2\pi fT_c c_{nN_w+k}} \alpha_{i,k} e^{j2\pi f\beta_{i,k}T_\beta} \pi_i\right|^2 \tag{6.16}$$

$$\begin{aligned} S_{C.2}(f) = \frac{2|W(f)|^2}{T_s\Lambda}\sum_{k=0}^{N_w-2}\sum_{k'=k+1}^{N_w-1}\sum_{n=0}^{\Lambda-1} \mathrm{Re}\Big\{ & e^{-j2\pi f(k'-k)T_r}\Big(\sum_{i=0}^{N_\sigma-1}\alpha_{i,k}\alpha_{i,k'}^{*} e^{-j2\pi f(\beta_{i,k'}-\beta_{i,k})T_\beta}\pi_i\Big) \\ & \times a_{nN_w+k} a_{nN_w+k'}^{*} e^{-j2\pi fT_c(c_{nN_w+k'}-c_{nN_w+k})}\Big\} \end{aligned} \tag{6.17}$$

$$\begin{aligned} S_{C.3}(f) = \frac{2|W(f)|^2}{T_s\Lambda}\mathrm{Re}\Big\{ & \sum_{k,k'=0}^{N_w-1}\sum_{n=0}^{\Lambda-1}\sum_{m=1}^{\infty}\sum_{i,i'=0}^{N_\sigma-1} e^{-j2\pi fT_s m} e^{-j2\pi f(k'-k)T_r}\alpha_{i,k}\alpha_{i',k'}^{*} e^{-j2\pi f(\beta_{i',k'}-\beta_{i,k})T_\beta}\pi_i(p_{ii'}^{(m)} - \pi_{i'}) \\ & \times a_{nN_w+k} a_{(n+m)N_w+k'}^{*} e^{-j2\pi fT_c(c_{(n+m)N_w+k'}-c_{nN_w+k})}\Big\} \end{aligned} \tag{6.18}$$

$$S_D(f) = \frac{|W(f)|^2}{(T_s\Lambda)^2}\left|\sum_{k=0}^{N_w-1}\sum_{n=0}^{\Lambda-1}\sum_{i=0}^{N_\sigma-1} e^{j2\pi fnT_s} e^{j2\pi fkT_r} a_{nN_w+k} e^{j2\pi fT_c c_{nN_w+k}} \alpha_{i,k} e^{j2\pi f\beta_{i,k}T_\beta}\pi_i\right|^2 \sum_{r=-\infty}^{\infty}\delta(f - \tfrac{r}{T_s\Lambda}) \tag{6.19}$$

where $S_C(f)$ is the PSD's continuous part, $S_D(f)$ is the PSD's discrete part, $W(f) = \Im\{w(t)\}$ and $\Im\{\cdot\}$ is the Fourier transform operator. Remember

$$\chi_{acw} = \mathrm{lcm}(\chi_a, \chi_c, N_w) \quad \text{and} \quad \Lambda = \chi_{acw}/N_w \tag{6.20}$$

where lcm stands for least common multiple. Note how spectral lines in the PSD are expected to appear at frequencies multiples of $1/(\Lambda T_s)$.

6.3.3 Spectral Line Free Condition for Binary Convolutionally coded BPSK/Q-BOPPM IR-Based UWB

Let us now focus on the PSD's discrete part, $S_D(f)$. By reordering the sums in Equation (6.19) we get

$$S_D(f)=\frac{|W(f)|^2}{(T_s\Lambda)^2}\left|\sum_{k=0}^{N_w-1}\sum_{n=0}^{\Lambda-1}e^{j2\pi fnT_s}e^{j2\pi fkT_r}a_{nN_w+k}e^{j2\pi fT_c c_{nN_w+k}}\sum_{i=0}^{N_\sigma-1}\alpha_{i,k}e^{j2\pi f\beta_{i,k}T_\beta}\pi_i\right|^2\sum_{r=-\infty}^{\infty}\delta(f-\tfrac{r}{T_s\Lambda}) \quad (6.21)$$

Now note how all spectral lines can be eliminated by making the absolute value in Equation (6.21) equal to zero. Thus all spectral lines can be eliminated if

$$\sum_{i=0}^{N_\sigma-1}\alpha_{i,k}e^{j2\pi f\beta_{i,k}T_\beta}\pi_i=0 \quad (6.22)$$

for all values of $k=0,1,...,N_w-1$. Therefore Equation (6.22) will be referred as the ***spectral line-free condition*** (SLF) for binary convolutional encoders.

6.4 Spectral Line Elimination Capabilities of Binary Convolutional Encoders with First Order BMS Inputs for BPSK/Q-BOPPM IR-Based UWB Systems

In this section the spectral line elimination capabilities of feedforward and recursive convolutional encoders will be assessed. Particularly the feasibility of fulfilling the spectral line-free condition defined by Equation (6.22) when using these structures is analysed.

6.4.1 Failure of the Spectral Free Suppression Condition for Rate $1/\kappa$ Feedforward Binary to *M*-ary Convolutional Encoders with First Order BMS Inputs

As explained in Section 6.2.1, the Theorem 5.1 describes the SE pair Markov model's steady state behaviour for rate $1/\kappa$ feedforward convolutional encoders. Thus from Equation (6.8) the steady state probabilities $\pi_i=\pi_{\sigma,i}$ are a function of at least two probabilities from the following set:

$$0<p_{y,01}<1,\quad 0<p_{y,10}<1,\quad p_{y,00}=1-p_{y,01},\quad p_{y,11}=1-p_{y,10},\quad \pi_{y,0}=\frac{p_{y,10}}{p_{y,01}+p_{y,10}},\quad \pi_{y,1}=\frac{p_{y,01}}{p_{y,01}+p_{y,10}} \quad (6.23)$$

Therefore although the output mapping function, $\gamma(\cdot)$, defined by the encoder's generator polynomials can be designed to satisfy the spectral line-free condition for particular cases, there exist combinations for which the spectral line suppression condition can not be fulfilled no matter what (valid) output mapping function is used. In order to see this, note that it is not difficult to find values of $p_{y,01}$ and $p_{y,10}$ such that

$$\pi_i = \pi_{nN_q+j} = \pi_{s_{N_q+j}} = p_{y,\mu_1 n} p_{y,\mu_2\mu_1} \cdots p_{y,\mu_{v-1}\mu_{v-2}} p_{y,\mu_v\mu_{v-1}} \pi_{y,\mu_v} > \frac{1}{2} \tag{6.24}$$

which concludes to the no fulfilment of Equation (6.22) as $\alpha_{i,k} \in \{-1,1\}$. Therefore it can be stated that feedforward structures do not provide a good construction framework for spectral line-free (SLF) binary to *M*-ary convolutional encoders with first order BMS inputs.

6.4.2 Spectral Line Suppression Condition Analysis for Rate $1/\kappa$ Recursive Binary Convolutional Encoders with First Order BMS Inputs

Now consider recursive implementations for rate $1/\kappa$ binary convolutional encoders with $b_v = 1$ (see Figure 6-3). As introduced in Section 6.2.2, the SE pair Markov model for this kind of convolutional encoders has steady stated probabilities defined by Theorem 5.2. Hence, from Equation (6.9) these probabilities can only assume one of two values namely

$$\pi_i = \pi_{s_{0N_q+j}} = \frac{\pi_{y,0}}{N_q}, \text{ for } i = 0,1,...,N_q - 1 \text{ and } \pi_i = \pi_{s_{N_q+j}} = \frac{\pi_{y,1}}{N_q}, \text{ for } i = N_q, N_q+1,...,2N_q - 1 \tag{6.25}$$

where $j = 0,1,...,N_q - 1$. Therefore, the recursive structures with $b_v = 1$ provide a better framework for the design of spectral line-free (SLF) encoders. Next an analysis similar to that introduced in Section 5.3.2 for recursive binary to *M*-ary convolutional encoders will be performed for rate $1/\kappa$ recursive binary convolutional encoders coupled to Q-BOPPM and BPSK IR-based UWB systems.

6.4.2.1 Analysis for Q-BOPPM IR-Based UWB Systems

Assume that k has been fixed in Equation (6.22) and that the number of encoder states is $N_q = 2^v \geq 4$ corresponding to $N_\sigma = 2^{v+1} \geq 8$ states in the SE pair Markov model. Note there are four different valid Q-BOPPM symbols:

$$(\alpha_{i,k}, \beta_{i,k}) = (-1,0);\ (\alpha_{i,k}, \beta_{i,k}) = (-1,1);\ (\alpha_{i,k}, \beta_{i,k}) = (+1,0);\ (\alpha_{i,k}, \beta_{i,k}) = (+1,1) \tag{6.26}$$

corresponding to the binary outputs [0, 0], [0, 1], [1, 0] and [1, 1] respectively and where $k = 0,1,...,N_w - 1$. Remember for this case $N_w = \kappa/2$.

Now define $\eta_0^{(n,k)}$ to be the number of states $s_i = s_{nN_q+j}$ whose k^{th} Q-BOPPM symbol is equal to $(\alpha_{i,k}, \beta_{i,k}) = (-1,0)$ where $n = 0,1$. Note from Equation (6.25) that these states' steady state probability is $\pi_i = \pi_{s_{nN_q+j}} = \pi_{y,n}/N_q$. Similarly define $\eta_1^{(n,k)}$, $\eta_2^{(n,k)}$ and $\eta_3^{(n,k)}$ to be the number of states $s_i = s_{nN_q+j}$ whose k^{th} Q-BOPPM symbol is equal to $(\alpha_{i,k}, \beta_{i,k}) = (-1,1)$, $(\alpha_{i,k}, \beta_{i,k}) = (+1,0)$ and $(\alpha_{i,k}, \beta_{i,k}) = (+1,1)$ respectively. Using these variables the spectral line-free condition defined by Equation (6.22) becomes

$$\begin{aligned} \sum_{i=0}^{N_\sigma - 1} \alpha_{i,k} e^{j2\pi f \beta_{i,k} T_\beta} \pi_i = {} & \frac{\pi_{y,0}}{N_q} \{-\eta_0^{(0,k)} - \eta_1^{(0,k)} e^{j2\pi f T_\beta} + \eta_2^{(0,k)} + \eta_3^{(0,k)} e^{j2\pi f T_\beta}\} + \\ & + \frac{\pi_{y,1}}{N_q} \{-\eta_0^{(1,k)} - \eta_1^{(1,k)} e^{j2\pi f T_\beta} + \eta_2^{(1,k)} + \eta_3^{(1,k)} e^{j2\pi f T_\beta}\} = 0 \end{aligned} \tag{6.27}$$

for this kind of recursive convolutional encoders.

Therefore our goal will be to find convolutional encoders with output mapping (defined by the rational generators) such that Equation (6.27) holds for every possible combination of $0 < p_{y,01} < 1$ and $0 < p_{y,10} < 1$ (or equivalently for every $0 < \pi_{y,0} < 1$ with $\pi_{y,1} = 1 - \pi_{y,0}$) and every value of k. Note a way to do this is by looking for encoders with output mapping, $\gamma(\cdot)$, such that

$$\eta_0^{(0,k)} = \eta_2^{(0,k)}; \quad \eta_1^{(0,k)} = \eta_3^{(0,k)}; \quad \eta_0^{(1,k)} = \eta_2^{(1,k)}; \quad \eta_1^{(1,k)} = \eta_3^{(1,k)} \tag{6.28}$$

holds for every value of k.

Consequently it can be concluded that by using recursive structures with $b_v = 1$ (see Figure 6-3) the construction of spectral line-free binary convolutional encoders with unbalanced (that is $p_{y,ii'} \neq 1/2$) BMS inputs for Q-BOPPM IR-based UWB systems is feasible.

Before continuing, it is important to mention that the verification of the spectral line-free condition must be performed separately for Q-BOPPM with natural mapping and Q-BOPPM with Gray mapping. This is due to the code bits to signal assignment is different for each mapping (see Equations (6.12) and (6.13)). Therefore the k^{th} Q-BOPPM symbol, $(\alpha_{i,k}, \beta_{i,k})$, corresponding to state, s_i, may be different for Gray mapping based Q-BOPPM and natural mapping based Q-BOPPM.

6.4.2.2 Analysis for BPSK IR-Based UWB Systems

For this case there are only two different symbols: $\alpha_{i,k} = -1$ and $\alpha_{i,k} = +1$. Therefore the spectral line suppression condition defined by Equation (6.22) becomes

$$\sum_{i=0}^{N_\sigma - 1} \alpha_{i,k} \pi_i = 0 \tag{6.29}$$

Performing an analysis similar to the one presented in the previous section Equation (6.29) can be rewritten as

$$\sum_{i=0}^{N_\sigma - 1} \alpha_{i,k} \pi_i = \frac{\pi_{y,0}}{N_q} \{-\eta'^{(0,k)}_{-1} + \eta'^{(0,k)}_{+1}\} + \frac{\pi_{y,1}}{N_q} \{-\eta'^{(1,k)}_{-1} + \eta'^{(1,k)}_{+1}\} = 0 \tag{6.30}$$

where $\eta'^{(n,k)}_{-1}$ and $\eta'^{(n,k)}_{+1}$ are the number of states $s_i = s_{nN_q + j}$ whose k^{th} BPSK symbol is equal to $\alpha_{i,k} = -1$ and $\alpha_{i,k} = +1$ respectively. Remember for this case $N_w = \kappa$ and thus $k = 0, 1, ..., \kappa - 1$.

Therefore, for BPSK IR-based UWB systems a rate $1/\kappa$ recursive binary convolutional encoder with $b_v = 1$ will be spectral line-free for every possible combination of $0 < p_{y,01} < 1$ and $0 < p_{y,10} < 1$ (or equivalently for every $0 < \pi_{y,0} < 1$ with $\pi_{y,1} = 1 - \pi_{y,0}$) if it has output mapping, $\gamma(\cdot)$, such that

$$\eta'^{(0,k)}_{-1} = \eta'^{(0,k)}_{+1} \quad \text{and} \quad \eta'^{(1,k)}_{-1} = \eta'^{(1,k)}_{+1} \tag{6.31}$$

holds for every value of k.

6.5 Code Search Procedure to Seek Recursive Spectral Line Free Binary Convolutional Encoders for BPSK/Q-BOPPM IR-Based UWB Systems with Unbalanced First Order BMS Inputs

The code search procedure introduced in this section for recursive spectral line-free (RSLF) binary convolutional encoders is similar to that introduced in Section 5.4 for recursive spectral line suppressive (RSLS) binary to M-ary convolutional encoders. The principal differences are that this time the encoders are binary and the main goal is to find spectral line-free encoders for BPSK/Q-BOPPM IR-based UWB systems with unbalanced BMS inputs.

Before continuing it is convenient to recall from Section 2.7.1 that a rate $1/\kappa$ recursive binary convolutional encoder can be described by the transfer function matrix

$$\mathbf{G}(D) = [g^{(0)}(D) \quad g^{(1)}(D) \ \ldots \ g^{(\kappa-1)}(D)] \tag{6.32}$$

where D is the delay operator. Each rational generator, $g^{(k)}(D)$, of $\mathbf{G}(D)$ has the form

$$g^{(k)}(D) = \frac{a^{(k)}(D)}{b(D)} = \frac{a_0^{(k)} + a_1^{(k)} D + \ldots + a_v^{(k)} D^v}{1 + b_1 D + \ldots + b_{v-1} D^{v-1} + b_v D^v} \tag{6.33}$$

in which $a_j^{(k)} \in \{0,1\}$, $b_j \in \{0,1\}$ and $k = 0,1,\ldots,\kappa-1$. The numerator in Equation (6.33) describes the generator's feedforward connections (feedforward polynomial) while the denominator is the feedback polynomial describing the shift register's feedback connections (see Figure 6-3). Therefore to define a particular rate $1/\kappa$ recursive binary convolutional encoder it is sufficient to specify one feedback polynomial, $b(D)$, and κ feedforward polynomials $a^{(k)}(D)$.

6.5.1 Superior Information Weight Spectrum Criteria

As in Chapters 4 and 5, another important aim of the code search procedure is to find the RSLF binary convolutional encoders with the best possible information weight spectrum (IWS). Therefore, the superior IWS criteria introduced in Section 4.4.2 will be used for comparison and selection purposes. Remember from Sections 4.4.2 and 2.7.4 that the superior IWS criteria is based on the form acquired by the upper bound on the bit error probability achieved when using binary convolutional encoders in output symmetric memoryless channels, [83, 100-103, 105-107]. This bound is given by

$$P_b \leq \sum_{d=d_{free}}^{\infty} B_d P_2(d) \tag{6.34}$$

where $\{B_d\}$ is the IWS introduced in Section 2.7.3 and $P_2(d)$ is the pairwise error probability between two codewords with binary Hamming distance d. It is worth noting that for this case the IWS is based on the binary Hamming distance properties of the encoder (see Sections 2.7.3). Thus, as explained in Sections 2.7.4 and 4.4.2, the superior IWS criteria consists of finding the encoder

with the largest binary free distance, d_{free}, and minimum $\{B_d\}$.

In the code search procedure introduced in this chapter the binary free distance and IWS are calculated using the FAST algorithm introduced by Cedervall and Johannesson, [103-108].

6.5.2 Code Search Procedure for Recursive Spectral Line Free Binary Convolutional Encoders

The flowchart of the code search procedure is shown in Figure 6-4. The goal is to find spectral line-free encoders with the best possible IWS. Note how this flowchart is similar to the one introduced in Section 5.4 for recursive spectral line suppressive binary to M-ary convolutional encoders. As in Section 5.4, several main tasks have been identified in the flowchart of Figure 6-4 for a better understanding of the code search procedure.

Task 1 consists of defining the main parameters of the code search procedure such as the total encoder memory, the feedback polynomial used and the code rate $1/\kappa$. As in Section 5.4 the feedback polynomials' selection pool for an encoder with total encoder memory, v, was chosen to consist of the simple $b(D)=1+D^v$ polynomial plus all known primitive polynomials of degree v. Again it is expected that RSLF binary convolutional encoders with primitive polynomial based feedback will provide better PSD characteristics than encoders with $b(D)=1+D^v$ based feedback.

In order to provide rate adaptability a nested code search approach has been adopted for convolutional encoders with rates $1/\kappa<1/4$ as in [112]. Due to the target here is to find RSLF convolutional encoders for both BPSK and Q-BOPPM, κ is always set to be a multiple of 2 as each Q-BOPPM symbol can transmit 2 code bits. Therefore in order to search for a rate $1/\kappa<1/4$ RSLF convolutional encoder, the generators of the best $1/(\kappa-2)$ RSLF convolutional encoder are used as the first $\kappa-2$ generators of the new encoder. Thus only two new generators are searched each time. The best rate 1/4 RSLF binary convolutional encoders are used as the parent encoders of RSLF convolutional encoders with rates $1/\kappa<1/4$.

During *Task 2* the target binary free distance, d_{tar}, is set. Initially the free distance of the best rate $1/\kappa$ convolutional encoders found up to date as reported in [107, 109, 111, 112] is used as the target free distance. For rate $1/\kappa<1/4$ encoders with total encoder memory $v=6$ the improved Heller upper bound on the binary free distance of rate $1/\kappa$ binary convolutional codes reported in [135] is used. If no maximum free distance (MFD) spectral line-free convolutional encoder is found when using the current target free distance, then target free distance is decreased until an encoder is found.

Note the total number of transfer function matrices to be searched for rate 1/4 encoders is $2^{4(v+1)}$ while for rate 1/2 and nested codes this total is $2^{2(v+1)}$. However this number is reduced in *Task 3* by identifying and discarding encoders with identical IWS.

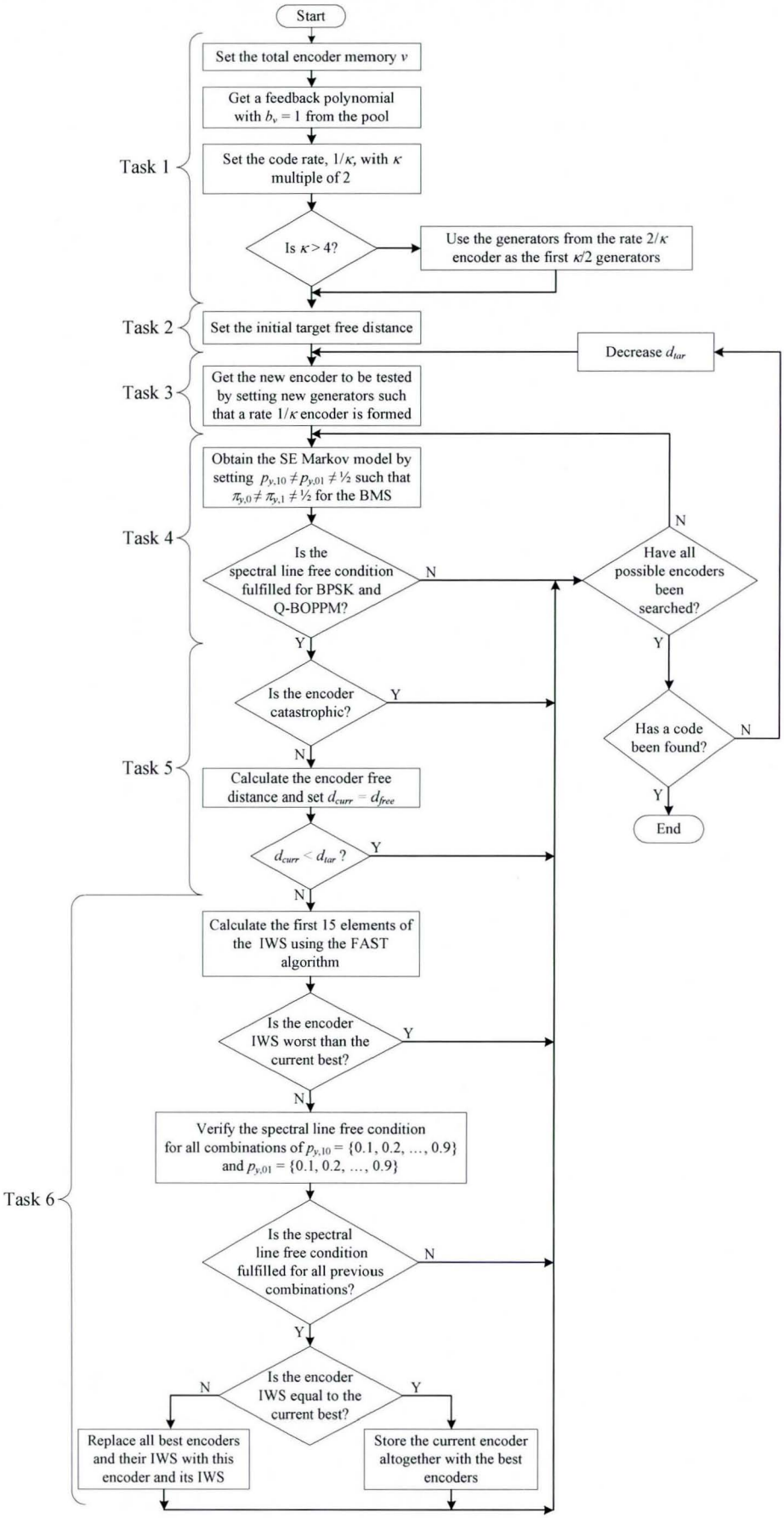


Figure 6-4. Flowchart diagram of the code search procedure for RSLF binary convolutional encoders.

As mentioned in Section 2.7.3 an important characteristic of binary convolutional encoders is that the free distance, distance spectrum, information weight spectrum and related distance properties do not change when interchanging the order of the columns in the transfer function matrix, $\mathbf{G}(D)$, [83, 100-103, 107]. Therefore this property is used in Task 3 to discard equivalent polynomials before advancing to the next task in the flowchart.

During *Task 4* the SE pair Markov model is obtained by assuming the source to be an unbalanced BMS with stationary probabilities $\pi_{y,0} \neq \pi_{y,1} \neq 1/2$. One or two specific numeric values for $p_{y,01} \neq 1/2$ and $p_{y,10} \neq p_{y,01}$ are used at this stage. The encoder is discarded if it does not satisfy the spectral line-free condition for BPSK, Q-BOPPM with natural mapping and Q-BOPPM with Gray mapping. Hence the encoder must satisfy both Equations (6.27) (Q-BOPPM natural and Gray mapping) and (6.30) (BPSK) or equivalently Equations (6.28) and (6.31).

Task 5 verifies if the current encoder is catastrophic (see Section 2.7.3). If the encoder is not catastrophic then its binary free distance, $d_{Curr} = d_{free}$, is calculated and compared with the target free distance, d_{tar}.

Task 6 is performed if the current free distance, d_{Curr}, is at least equal to the target free distance, d_{tar}. If this task is reached the current encoder IWS is calculated and compared to the best IWS. The encoder is discarded if its IWS is worst than the current best. If the encoder's IWS is at least equal to the current best, then the fulfilment of the spectral line-free condition for BPSK and Q-BOPPM (natural and Gray mapping) is verified for all combinations of $p_{y,01} = \{0.1, 0.2, ..., 0.9\}$ and $p_{y,10} = \{0.1, 0.2, ..., 0.9\}$. If the spectral line-free condition is satisfied and the encoder's IWS is equal to the current best it is saved in a file containing all encoders found with the same IWS. Finally if the encoder's IWS is better than the current best all previously stored encoders are discarded, the current encoder is saved and its IWS is set as the current best. The algorithm ends when an encoder with spectral line-free capabilities and the best possible IWS is found.

All different tasks in the code search procedure flowchart have been programmed in MATLAB®.

6.5.3 Code Search Results for RSLF Binary Convolutional Encoders for BPSK/Q-BOPPM IR-Based UWB Systems with Unbalanced First Order BMS Inputs

Using the code search procedure previously introduced, new maximum free distance (MFD) recursive binary convolutional encoders showing the spectral line-free property for BPSK/Q-BOPPM IR-based UWB with unbalanced BMS inputs have been found.

The best rate 1/2 recursive spectral line-free (RSLF) convolutional encoders are reported in Table 6-1 for recursive structures with feedback polynomials of the form $b(D) = 1 + D^v$ and in Table 6-2 for recursive structures with primitive feedback polynomials. Similarly, the best rate 1/4

RSLF convolutional encoders are reported in Table 6-3 for recursive structures with feedback polynomials of the form $b(D)=1+D^v$ and in Table 6-4 for recursive structures with primitive feedback polynomials. The first column in Tables 6-1 to 6-4 gives the total encoder memory v. The second and third columns give the feedback polynomial $b(D)$ and the feedforward polynomials $a^{(k)}(D)$ (represented in octal form) of the best RSLF convolutional encoders found. The last two columns give the code's binary free distance, d_{free}, and the first 15 components of the information weight (IWS) and distance spectrums.

Table 6-1. Best rate 1/2 recursive spectral line-free (RSLF) binary convolutional encoders with feedback polynomial $b(D) = 1 + D^v$. The feedback and feedforward polynomials are given in octal form. The first 15 elements of the IWS and distance spectrum are reported in the last column.

v	Feedback polynomial	Feedforward polynomials	d_{free}	Information weight spectrum (IWS): $[B_{d_{free}}, B_{d_{free}+1}, \ldots, B_{d_{free}+15}]$ Distance spectrum: $\{A_{d_{free}}, A_{d_{free}+1}, \ldots, A_{d_{free}+15}\}$
3	11	(13, 17)	6	[4, 12, 28, 70, 174, 430, 1046, 2520, 6032, 14348, 33946, 79946, 187522, 438284, 1021124] {1, 3, 5, 11, 25, 55, 121, 267, 589, 1299, 2865, 6319, 13937, 30739, 67797}
4	21	(23, 35)	7	[8, 18, 30, 114, 296, 628, 1742, 4568, 10670, 26702, 67660, 163510, 398728, 983612, 2388524] {2, 3, 4, 16, 37, 68, 176, 432, 925, 2156, 5153, 11696, 26868, 62885, 145085}
5	41	(53, 75)	8	[4, 54, 52, 88, 440, 920, 2980, 6964, 15342, 43692, 105970, 265998, 672438, 1620268, 4074604] {1, 8, 7, 12, 48, 95, 281, 605, 1272, 3334, 7615, 18131, 43197, 99210, 237248}
6	101	(117, 155)	10	[68, 0, 340, 0, 1990, 0, 15650, 0, 98842, 0, 614666, 0, 3954342, 0, 24756766] {11, 0, 38, 0, 193, 0, 1331, 0, 7275, 0, 40406, 0, 234969, 0, 1337714}
7	201	(237, 345)	10	[4, 40, 108, 244, 540, 1454, 3720, 9418, 24804, 64028, 164172, 409724, 1034640, 2613160, 6553858] {1, 6, 12, 26, 52, 132, 317, 730, 1823, 4446, 10739, 25358, 60773, 146396, 350399}
8	401	(435, 657)	12	[66, 0, 490, 0, 3424, 0, 21776, 0, 144868, 0, 919460, 0, 5909052, 0, 37371990] {11, 0, 50, 0, 286, 0, 1630, 0, 9639, 0, 55152, 0, 320782, 0, 1859184}
9	1001	(1151, 1753)	12	[4, 42, 180, 324, 780, 2278, 5432, 14208, 37146, 93452, 234394, 600822, 1524076, 3826906, 9640382] {1, 7, 19, 28, 69, 185, 411, 1010, 2492, 5963, 14192, 34584, 83567, 200343, 483393}
10	2001	(2473, 3217)	14	[108, 0, 1130, 0, 5864, 0, 40366, 0, 260952, 0, 1650952, 0, 10471980, 0, 66180302] {14, 0, 92, 0, 426, 0, 2595, 0, 15221, 0, 87694, 0, 509876, 0, 2975097}

Table 6-2. Best rate 1/2 recursive spectral line-free (RSLF) binary convolutional encoders with primitive feedback polynomials. The feedback and feedforward polynomials are given in octal form. The first 15 elements of the IWS and distance spectrum are reported in the last column.

v	Feedback polynomial	Feedforward polynomials	d_{free}	Information weight spectrum (IWS): $[B_{d_{free}}, B_{d_{free}+1}, \ldots, B_{d_{free}+15}]$ Distance spectrum: $\{A_{d_{free}}, A_{d_{free}+1}, \ldots, A_{d_{free}+15}\}$
3	15	(13, 17)	6	[4, 13, 28, 71, 180, 439, 1064, 2563, 6126, 14555, 34406, 80965, 189778, 443281, 1032192] {1, 3, 5, 11, 25, 55, 121, 267, 589, 1299, 2865, 6319, 13937, 30739, 67797}
4	23	(27, 31)	7	[8, 20, 28, 104, 309, 638, 1668, 4556, 10741, 26236, 66907, 163078, 394832, 972948, 2371517] {2, 3, 4, 16, 37, 68, 176, 432, 925, 2156, 5153, 11696, 26868, 62885, 145085}
5	51	(53, 75)	8	[4, 54, 46, 80, 436, 911, 2926, 6731, 14922, 42966, 104164, 261315, 659994, 1592486, 4012396] {1, 8, 7, 12, 48, 95, 281, 605, 1272, 3334, 7615, 18131, 43197, 99210, 237248}
6	103	(117, 155)	10	[72, 0, 335, 0, 2012, 0, 15779, 0, 98797, 0, 617558, 0, 3973717, 0, 24839366] {11, 0, 38, 0, 193, 0, 1331, 0, 7275, 0, 40406, 0, 234969, 0, 1337714}
7	253	(255, 363)	10	[4, 22, 106, 202, 472, 1237, 3234, 8233, 22228, 55753, 142118, 364648, 920678, 2312367, 5837064] {1, 4, 13, 24, 50, 117, 285, 689, 1724, 4089, 9764, 23638, 56531, 134963, 324501}
8	551	(557, 751)	12	[64, 69, 288, 505, 1716, 3848, 10914, 25539, 71834, 172755, 454420, 1130823, 2909024, 7245902, 18427122] {10, 9, 30, 51, 156, 340, 875, 1951, 5127, 11589, 28740, 68191, 166304, 396120, 961154}
9	1257	(1055, 1657)	12	[4, 55, 132, 359, 700, 1953, 5500, 12916, 34022, 88432, 222430, 565157, 1441878, 3629621, 9138132] {1, 7, 15, 37, 65, 167, 446, 974, 2408, 5930, 14101, 34021, 82502, 198077, 476528}
10	2475	(2473, 3217)	14	[110, 0, 1020, 0, 5370, 0, 37539, 0, 244463, 0, 1558017, 0, 9926179, 0, 63022211] {14, 0, 92, 0, 426, 0, 2595, 0, 15221, 0, 87694, 0, 509876, 0, 2975097}

Table 6-3. Best rate 1/4 recursive spectral line-free (RSLF) binary convolutional encoders with feedback polynomial $b(D) = 1 + D^v$. The feedback and feedforward polynomials are given in octal form. The first 15 elements of the IWS and distance spectrum are reported in the last column.

v	Fdbk poly	Feedforward polynomials	d_{free}	Information weight spectrum (IWS): $[B_{d_{free}}, B_{d_{free}+1}, \ldots, B_{d_{free}+15}]$ Distance spectrum: $\{A_{d_{free}}, A_{d_{free}+1}, \ldots, A_{d_{free}+15}\}$
3	11	(13, 15, 13, 17)	13	[8, 4, 0, 16, 4, 24, 50, 26, 104, 110, 138, 360, 326, 638, 1058] {2, 1, 0, 3, 1, 4, 8, 4, 15, 16, 18, 45, 40, 73, 119}
4	21	(25, 27, 33, 37)	16	[16, 0, 14, 0, 24, 0, 96, 0, 204, 0, 418, 0, 982, 0, 2222] {4, 0, 2, 0, 4, 0, 15, 0, 30, 0, 54, 0, 115, 0, 252}
5	41	(45, 55, 73, 77)	18	[10, 0, 18, 0, 60, 0, 100, 0, 210, 0, 504, 0, 1110, 0, 2868] {3, 0, 3, 0, 9, 0, 13, 0, 26, 0, 66, 0, 127, 0, 311}
6	101	(117, 127, 155, 171)	20	[6, 0, 34, 0, 62, 0, 110, 0, 212, 0, 556, 0, 1304, 0, 2778] {2, 0, 6, 0, 7, 0, 15, 0, 25, 0, 62, 0, 138, 0, 294}
7	201	(231, 273, 327, 375)	22	[4, 8, 8, 12, 36, 64, 86, 134, 144, 150, 314, 486, 746, 1152, 1650] {1, 2, 2, 2, 5, 8, 10, 14, 14, 18, 32, 50, 74, 107, 159}
8	401	(473, 513, 671, 765)	24	[2, 0, 30, 0, 66, 0, 216, 0, 364, 0, 772, 0, 1694, 0, 3862] {11, 0, 50, 0, 286, 0, 1630, 0, 9639, 0, 55152, 0, 320782, 0, 1859184}
9	1001	(1137, 1255, 1571, 1663)	27	[14, 20, 0, 56, 104, 114, 108, 162, 268, 500, 740, 818, 1416, 2390, 3342, 4874, 6920, 11548, 18480, 25726] {3, 4, 0, 5, 12, 12, 10, 17, 26, 46, 69, 70, 120, 199, 271, 391, 540, 880, 1371, 1881}

Table 6-4. Best rate 1/4 recursive spectral line-free (RSLF) binary convolutional encoders with primitive feedback polynomials. The feedback and feedforward polynomials are given in octal form. The first 15 elements of the IWS and distance spectrum are reported in the last column.

v	Fdbk poly	Feedforward polynomials	d_{free}	Information weight spectrum (IWS): $[B_{d_{free}}, B_{d_{free}+1}, \ldots, B_{d_{free}+15}]$ Distance spectrum: $\{A_{d_{free}}, A_{d_{free}+1}, \ldots, A_{d_{free}+15}\}$
3	13	(11, 15, 15, 17)	12	[3, 0, 11, 0, 17, 0, 45, 0, 113, 0, 250, 0, 568, 0, 1317] {1, 0, 2, 0, 4, 0, 7, 0, 17, 0, 34, 0, 71, 0, 149}
4	23	(25, 33, 35, 37)	16	[18, 0, 9, 0, 23, 0, 104, 0, 212, 0, 409, 0, 953, 0, 2287] {18, 0, 9, 0, 23, 0, 104, 0, 212, 0, 409, 0, 953, 0, 2287}
5	67	(47, 65, 73, 57)	18	[12, 0, 29, 0, 38, 0, 88, 0, 164, 0, 491, 0, 1369, 0, 2552] {3, 0, 5, 0, 6, 0, 12, 0, 23, 0, 67, 0, 157, 0, 283}
6	103	(123, 171, 175, 133)	20	[7, 0, 37, 0, 83, 0, 86, 0, 321, 0, 625, 0, 1346, 0, 3353] {2, 0, 6, 0, 11, 0, 11, 0, 39, 0, 69, 0, 149, 0, 346}
7	203	(273, 327, 375, 231)	22	[4, 8, 12, 14, 36, 64, 84, 118, 134, 166, 300, 484, 740, 1085, 1646] {1, 2, 2, 2, 5, 8, 10, 14, 14, 18, 32, 50, 74, 107, 159}
8	607	(467, 635, 533, 771)	24	[4, 10, 16, 23, 24, 78, 104, 153, 108, 173, 414, 518, 900, 1213, 1712] {1, 2, 2, 3, 3, 10, 14, 17, 12, 19, 42, 52, 87, 117, 160}
9	1333	(1137, 1255, 1571, 1663)	27	[17, 28, 0, 56, 108, 114, 100, 172, 274, 476, 729, 756, 1394, 2370, 3255, 4844, 6814, 11484, 18277, 25302] {3, 4, 0, 5, 12, 12, 10, 17, 26, 46, 69, 70, 120, 199, 271, 391, 540, 880, 1371, 1881}

All the encoders reported in Tables 6-1 to 6-4 have maximum free distance with the exception of the $v=3$ encoder in Table 6-4 (no MFD rate 1/4 RSLF convolutional code with primitive feedback polynomial was found for $v=3$). Note as well that not all the encoders with $b(D)=1+D^v$ feedback polynomial have better IWS than the encoders with primitive feedback polynomials. It is worth mentioning that the IWS of the spectral line-free encoders reported in Tables 6-1 to 6-4 is slightly inferior than the IWS of the best rate 1/2 and 1/4 binary convolutional encoders known reported in [107, 109, 111]. However it is important to highlight that all the convolutional encoders reported in [107, 109, 111] are feedforward and thus they do not fulfil the spectral line-free condition for all combinations of $0<p_{y,01}<1$ and $0<p_{y,10}<1$ as shown in Section 6.4.1, (commonly the spectral line-free condition is only satisfied for $p_{y,10}=p_{y,01}=1/2$). Therefore, the possible decrease in bit error rate (BER) performance due to the inferior IWS of the RSLF convolutional encoders would be compensated by the superior PSD characteristics achieved when using these encoders.

As with the recursive spectral line suppressive (RSLS) binary to *M*-ary convolutional encoders introduced in Chapter 5, the RSLF encoders with primitive feedback polynomials offer better PSD characteristics than the RSLF encoders with $b(D)=1+D^v$ polynomials (see Section 6.6). Therefore the rate 1/4 convolutional encoders in Table 6-4 have been used as parent codes for the search of rate $1/\kappa<1/4$ RSLS convolutional encoders. The results of this code search are presented in Tables 6-5 to 6-9 for rates ranging from 1/6 to 1/32 and total encoder memory ranging from $v=5$ to $v=9$. These tables are interpreted as follows: the first column gives the code rate $1/\kappa$; the second column gives the two additional generators needed to form the $1/\kappa$ encoder from the $1/(\kappa-2)$ encoder; the third column gives the code's binary free distance, d_{free}, and the last two columns the first 15 components of the information weight (IWS) and distance spectrums. Note the additional generators are given in pairs as the goal is to find spectral line-free convolutional encoders for both BPSK (one code bit per signal) and Q-BOPPM (two code bits per signal) IR-based UWB systems. All the encoders in Tables 6-5 to 6-9 have maximum free distance.

Table 6-5. MFD rate 1/6 to 1/32 RSLF binary convolutional encoders with total encoder memory $v = 5$ and primitive feedback polynomial. The rate 1/4 RSLF convolutional encoder with feedback polynomial 67_8 and feedforward polynomials $(47, 65, 73, 57)_8$ reported in Table 6-4 was used as parent code. The additional feedforward polynomials are given in octal form. The first 15 elements of the IWS and distance spectrum are reported in the last two columns.

Rate	Additional Feedforward polynomials	d_{free}	Information weight spectrum (IWS) $[B_{d_{free}}, B_{d_{free}+1}, \ldots, B_{d_{free}+15}]$	Distance spectrum $\{A_{d_{free}}, A_{d_{free}+1}, \ldots, A_{d_{free}+15}\}$
1/6	(53, 75)	27	[8, 10, 0, 0, 10, 14, 0, 14, 26, 10, 24, 60, 32, 40, 95]	{2, 2, 0, 0, 2, 2, 0, 2, 4, 2, 4, 9, 4, 6, 13}
1/8	(57, 71)	36	[9, 0, 14, 0, 0, 0, 5, 0, 6, 0, 37, 0, 59, 0, 13]	{2, 0, 3, 0, 0, 0, 1, 0, 1, 0, 5, 0, 9, 0, 3}
1/10	(55, 75)	45	[5, 10, 3, 0, 0, 0, 10, 0, 0, 0, 16, 22, 0, 6, 5]	{1, 2, 1, 0, 0, 0, 2, 0, 0, 0, 2, 3, 0, 1, 1}
1/12	(57, 73)	54	[4, 8, 6, 0, 0, 5, 0, 0, 0, 5, 0, 0, 8, 16, 14]	{1, 2, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 2, 2}
1/14	(55, 75)	64	[18, 0, 0, 0, 0, 0, 0, 0, 10, 0, 0, 0, 32, 0, 0]	{4, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 4, 0, 0}
1/16	(57, 71)	72	[4, 5, 6, 3, 0, 0, 0, 5, 0, 0, 0, 5, 0, 0, 8]	{1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1}
1/18	(65, 73)	82	[12, 0, 6, 0, 0, 0, 0, 0, 0, 0, 10, 0, 0, 0, 0]	{3, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0}
1/20	(57, 65)	91	[8, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 10, 0, 0]	{2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0}
1/22	(47, 57)	100	[9, 0, 9, 0, 0, 0, 0, 0, 0, 0, 5, 0, 0, 0, 5]	{2, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1}
1/24	(55, 75)	109	[5, 10, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 10]	{1, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2}
1/26	(57, 73)	118	[4, 8, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 0]	{1, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0}
1/28	(55, 75)	128	[18, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	{4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
1/30	(57, 71)	136	[4, 5, 6, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	{1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
1/32	(65, 73)	146	[12, 0, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	{3, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}

Table 6-6. MFD rate 1/6 to 1/32 RSLF binary convolutional encoders with total encoder memory $v = 6$ and primitive feedback polynomial. The rate 1/4 RSLF convolutional encoder with feedback polynomial 103_8 and feedforward polynomials $(123, 171, 175, 133)_8$ reported in Table 6-4 was used as parent code. The additional feedforward polynomials are given in octal form. The first 15 elements of the IWS and distance spectrum are reported in the last column.

Rate	Additional Feedforward polynomials	d_{free}	Information weight spectrum (IWS) $[B_{d_{free}}, B_{d_{free}+1}, \ldots, B_{d_{free}+15}]$	Distance spectrum $\{A_{d_{free}}, A_{d_{free}+1}, \ldots, A_{d_{free}+15}\}$
1/6	(135, 157)	30	[4, 8, 6, 0, 6, 30, 12, 0, 30, 33, 40, 59, 68, 80, 98]	{1, 2, 1, 0, 1, 4, 2, 0, 4, 5, 5, 7, 9, 10, 12}
1/8	(113, 165)	40	[3, 0, 15, 0, 0, 0, 20, 0, 21, 0, 29, 0, 26, 0, 45]	{1, 0, 3, 0, 0, 0, 3, 0, 3, 0, 4, 0, 4, 0, 6}
1/10	(137, 153)	51	[8, 10, 0, 0, 0, 0, 7, 12, 0, 8, 21, 0, 14, 14, 16]	{2, 2, 0, 0, 0, 0, 1, 2, 0, 1, 3, 0, 2, 2, 2}
1/12	(117, 165)	61	[3, 10, 5, 0, 0, 0, 0, 0, 14, 6, 0, 14, 7, 0, 28]	{1, 2, 1, 0, 0, 0, 0, 0, 2, 1, 0, 2, 1, 0, 4}
1/14	(137, 155)	72	[18, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 26, 0, 0, 0, 22]	{4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 3}
1/16	(153, 171)	82	[12, 0, 6, 0, 0, 0, 0, 0, 0, 0, 6, 0, 20, 0, 0]	{3, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 3, 0, 0}
1/18	(137, 151)	92	[7, 0, 11, 0, 0, 0, 0, 0, 0, 0, 0, 12, 0, 14]	{2, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 2}
1/20	(135, 173)	102	[4, 8, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 6]	{1, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1}
1/22	(127, 151)	112	[3, 0, 15, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	{1, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
1/24	(153, 175)	123	[8, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	{2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
1/26	(127, 171)	133	[3, 10, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	{1, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
1/28	(133, 175)	144	[18, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	{4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
1/30	(123, 157)	154	[12, 0, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	{3, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
1/32	(113, 175)	164	[7, 0, 11, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	{2, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}

Table 6-7. MFD rate 1/6 to 1/32 RSLF binary convolutional encoders with total encoder memory $v = 7$ and primitive feedback polynomial. The rate 1/4 RSLF convolutional encoder with feedback polynomial 203_8 and feedforward polynomials $(273, 327, 375, 231)_8$ reported in Table 6-4 was used as parent code. The additional feedforward polynomials are given in octal form. The first 15 elements of the IWS and distance spectrum are reported in the last column.

Rate	Additional Feedforward polynomials	d_{free}	Information weight spectrum (IWS) $[B_{d_{free}}, B_{d_{free}+1}, \ldots, B_{d_{free}+15}]$	Distance spectrum $\{A_{d_{free}}, A_{d_{free}+1}, \ldots, A_{d_{free}+15}\}$
1/6	(265, 317)	34	[12, 0, 12, 0, 20, 0, 39, 0, 66, 0, 172, 0, 144, 0, 210]	{3, 0, 2, 0, 3, 0, 5, 0, 8, 0, 20, 0, 16, 0, 24}
1/8	(251, 357)	45	[3, 10, 5, 0, 14, 6, 0, 6, 0, 0, 39, 30, 41, 62, 34]	{1, 2, 1, 0, 2, 1, 0, 1, 0, 0, 5, 4, 5, 7, 4}
1/10	(233, 275)	56	[3, 0, 15, 0, 0, 0, 20, 0, 6, 0, 7, 0, 25, 0, 37]	{1, 0, 3, 0, 0, 0, 3, 0, 1, 0, 1, 0, 3, 0, 5}
1/12	(313, 375)	68	[7, 0, 11, 0, 0, 0, 12, 0, 14, 0, 0, 0, 22, 0, 7]	{2, 0, 2, 0, 0, 0, 2, 0, 2, 0, 0, 0, 3, 0, 1}
1/14	(265, 367)	80	[18, 0, 0, 0, 0, 0, 0, 0, 26, 0, 0, 0, 0, 0, 22]	{4, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 3}
1/16	(257, 331)	91	[8, 10, 0, 0, 0, 0, 0, 0, 7, 12, 0, 0, 7, 0, 0]	{2, 2, 0, 0, 0, 0, 0, 0, 1, 2, 0, 0, 1, 0, 0}
1/18	(235, 337)	102	[4, 8, 6, 0, 0, 0, 0, 0, 0, 0, 6, 14, 6, 0, 0]	{1, 2, 1, 0, 0, 0, 0, 0, 0, 0, 1, 2, 1, 0, 0}
1/20	(255, 363)	114	[12, 0, 6, 0, 0, 0, 0, 0, 0, 0, 6, 0, 20, 0, 0]	{3, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 3, 0, 0}
1/22	(257, 351)	125	[3, 10, 5, 0, 0, 0, 0, 0, 0, 0, 0, 14, 6, 0]	{1, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 2, 1, 0}
1/24	(233, 275)	136	[3, 0, 15, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 20]	{1, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3}
1/26	(263, 375)	148	[7, 0, 11, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 12]	{2, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2}
1/28	(265, 367)	160	[18, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	{4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
1/30	(257, 331)	171	[8, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	{2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
1/32	(337, 351)	182	[4, 8, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	{1, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}

It is important to highlight that although changing the generators order of a particular encoder does not affect its distance properties (d_{free}, IWS, etc. – see Section 2.7.3) such a change can affect the PSD characteristics of the transmitted signal (see Section 3.5.3 for an example). Therefore care should be taken when interchanging columns of the encoders reported in Tables 6-1 to 6-9. This is

particularly important for Q-BOPPM systems where the outputs of two consecutive generators correspond to one Q-BOPPM symbol. This is the reason why the generators in Tables 6-1 to 6-9 are not necessarily numerically ordered.

Table 6-8. MFD rate 1/6 to 1/32 RSLF binary convolutional encoders with total encoder memory $v = 8$ and primitive feedback polynomial. The rate 1/4 RSLF convolutional encoder with feedback polynomial 607_8 and feedforward polynomials $(467, 635, 533, 771)_8$ reported in Table 6-4 was used as parent code. The additional feedforward polynomials are given in octal form. The first 15 elements of the IWS and distance spectrum are reported in the last column.

Rate	Additional Feedforward polynomials	d_{free}	Information weight spectrum (IWS) $[B_{d_{free}}, B_{d_{free}+1}, \ldots, B_{d_{free}+15}]$	Distance spectrum $\{A_{d_{free}}, A_{d_{free}+1}, \ldots, A_{d_{free}+15}\}$
1/6	(457, 725)	37	[5, 12, 19, 6, 0, 16, 11, 26, 86, 62, 66, 42, 41, 110, 210]	{1, 2, 3, 1, 0, 2, 1, 3, 10, 8, 8, 6, 5, 12, 22}
1/8	(575, 663)	50	[14, 0, 14, 0, 24, 0, 8, 0, 26, 0, 84, 0, 29, 0, 99]	{3, 0, 2, 0, 3, 0, 1, 0, 3, 0, 10, 0, 4, 0, 12}
1/10	(551, 767)	62	[4, 10, 8, 0, 10, 14, 6, 0, 0, 0, 8, 0, 16, 52, 32]	{1, 2, 1, 0, 1, 2, 1, 0, 0, 0, 1, 0, 2, 6, 4}
1/12	(513, 657)	75	[10, 12, 0, 0, 7, 16, 0, 0, 7, 0, 0, 0, 16, 16, 0]	{2, 2, 0, 0, 1, 2, 0, 0, 1, 0, 0, 0, 2, 2, 0}
1/14	(515, 737)	88	[22, 0, 0, 0, 0, 0, 30, 0, 0, 0, 0, 0, 7, 0, 25]	{4, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 1, 0, 3}
1/16	(711, 753)	100	[9, 0, 13, 0, 0, 0, 14, 0, 16, 0, 0, 0, 0, 0, 0]	{2, 0, 2, 0, 0, 0, 2, 0, 2, 0, 0, 0, 0, 0, 0}
1/18	(531, 677)	112	[4, 5, 8, 5, 0, 0, 0, 7, 16, 0, 0, 7, 0, 0, 0]	{1, 1, 1, 1, 0, 0, 0, 1, 2, 0, 0, 1, 0, 0, 0}
1/20	(473, 725)	125	[5, 12, 5, 0, 0, 0, 0, 0, 14, 6, 0, 10, 0, 0, 0]	{1, 2, 1, 0, 0, 0, 0, 0, 2, 1, 0, 1, 0, 0, 0}
1/22	(555, 637)	138	[14, 0, 8, 0, 0, 0, 0, 0, 6, 0, 24, 0, 0, 0, 0]	{3, 0, 1, 0, 0, 0, 0, 0, 1, 0, 3, 0, 0, 0, 0}
1/24	(575, 671)	150	[4, 10, 8, 0, 0, 0, 0, 0, 0, 0, 10, 14, 6, 0, 0]	{1, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 1, 0, 0}
1/26	(515, 753)	163	[10, 12, 0, 0, 0, 0, 0, 0, 0, 0, 7, 16, 0, 0, 7]	{2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 0, 0, 1}
1/28	(453, 767)	176	[22, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 30, 0, 0, 0]	{4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0}
1/30	(571, 647)	188	[9, 0, 13, 0, 0, 0, 0, 0, 0, 0, 0, 0, 14, 0, 16]	{2, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 2}
1/32	(465, 773)	200	[4, 5, 8, 5, 0, 0, 0, 0, 0, 0, 0, 0, 7, 16]	{1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2}

Table 6-9. MFD rate 1/6 to 1/32 RSLF binary convolutional encoders with total encoder memory $v = 9$ and primitive feedback polynomial. The rate 1/4 RSLF convolutional encoder with feedback polynomial 1333_8 and feedforward polynomials $(1137, 1255, 1571, 1663)_8$ reported in Table 6-4 was used as parent code. The additional feedforward polynomials are given in octal form. The first 15 elements of the IWS and distance spectrum are reported in the last column.

Rate	Additional Feedforward polynomials	d_{free}	Information weight spectrum (IWS) $[B_{d_{free}}, B_{d_{free}+1}, \ldots, B_{d_{free}+15}]$	Distance spectrum $\{A_{d_{free}}, A_{d_{free}+1}, \ldots, A_{d_{free}+15}\}$
1/6	(1135, 1517)	40	[7, 0, 41, 0, 17, 0, 58, 0, 136, 0, 117, 0, 273, 0, 344]	{1, 0, 6, 0, 2, 0, 6, 0, 14, 0, 13, 0, 28, 0, 34}
1/8	(1375, 1663)	54	[8, 10, 12, 18, 8, 0, 0, 9, 22, 63, 40, 14, 80, 63, 46]	{1, 2, 2, 2, 1, 0, 0, 1, 3, 7, 4, 2, 8, 7, 4}
1/10	(1237, 1531)	68	[15, 0, 23, 0, 18, 0, 0, 0, 0, 0, 68, 0, 52, 0, 62]	{2, 0, 4, 0, 2, 0, 0, 0, 0, 0, 8, 0, 5, 0, 7}
1/12	(1365, 1633)	82	[18, 0, 16, 0, 22, 0, 0, 0, 0, 0, 0, 48, 0, 40]	{3, 0, 2, 0, 3, 0, 0, 0, 0, 0, 0, 6, 0, 4}
1/14	(1267, 1751)	96	[26, 0, 0, 0, 30, 0, 0, 0, 0, 0, 0, 0, 0, 57]	{4, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 7}
1/16	(1145, 1727)	109	[7, 16, 3, 0, 18, 8, 0, 4, 0, 0, 0, 0, 0, 0]	{1, 2, 1, 0, 2, 1, 0, 1, 0, 0, 0, 0, 0, 0}
1/18	(1337, 1545)	123	[10, 16, 0, 0, 7, 12, 0, 0, 11, 0, 0, 0, 0, 0, 0]	{2, 2, 0, 0, 1, 2, 0, 0, 1, 0, 0, 0, 0, 0, 0}
1/20	(1117, 1655)	136	[7, 0, 19, 0, 0, 0, 22, 0, 8, 0, 0, 0, 0, 0, 0]	{1, 0, 3, 0, 0, 0, 3, 0, 1, 0, 0, 0, 0, 0, 0}
1/22	(1355, 1763)	150	[8, 10, 8, 0, 0, 0, 4, 18, 8, 0, 0, 0, 0, 0, 0]	{1, 2, 1, 0, 0, 0, 1, 2, 1, 0, 0, 0, 0, 0, 0}
1/24	(1237, 1531)	164	[15, 0, 11, 0, 0, 0, 12, 0, 18, 0, 0, 0, 0, 0, 0]	{2, 0, 2, 0, 0, 0, 2, 0, 2, 0, 0, 0, 0, 0, 0}
1/26	(1365, 1473)	178	[18, 0, 8, 0, 0, 0, 8, 0, 22, 0, 0, 0, 0, 0, 0]	{3, 0, 1, 0, 0, 0, 1, 0, 3, 0, 0, 0, 0, 0, 0}
1/28	(1127, 1575)	192	[26, 0, 0, 0, 0, 0, 0, 0, 30, 0, 0, 0, 0, 0, 0]	{4, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0}
1/30	(1365, 1623)	205	[7, 16, 3, 0, 0, 0, 0, 0, 18, 8, 0, 4, 0, 0, 0]	{1, 2, 1, 0, 0, 0, 0, 0, 2, 1, 0, 1, 0, 0, 0}
1/32	(1337, 1545)	219	[10, 16, 0, 0, 0, 0, 0, 0, 7, 12, 0, 0, 11, 0, 0]	{2, 2, 0, 0, 0, 0, 0, 0, 1, 2, 0, 0, 1, 0, 0}

6.6 PSD Comparisons of RSLF Binary Convolutional Encoders with the Best Binary Convolutional Encoders Known and the Non-Coded Scheme with Pulse Repetition

Remember that the examples presented in Chapter 3 – Section 3.5 showed that for Q-BOPPM IR-based UWB systems the non-coded pulse repetition scheme, the scheme using a systematic recursive convolutional encoder and the scheme employing a feedforward convolutional encoder do not have a spectral line-free PSD when the binary input stream is generated by an unbalanced BMS. In this section, further PSD examples for Q-BOPPM (natural mapping) IR-based UWB

systems are introduced in order to compare the PSD characteristics of the RSLF binary convolutional encoders introduced in Section 6.5.3. As well examples of the peak to average ratio achieved when using the encoders reported in Tables 6-1 to 6-4 are provided in Section 6.6.2.

6.6.1 PSD Comparisons of the Rate 1/4, $v = 4$, RSLF Binary Convolutional Encoders; the Best Rate 1/4, $v = 4$, Feedforward Binary Convolutional Encoder Known and a Rate Equivalent Non-Coded Pulse Repetition Scheme for Q-BOPPM IR Based UWB

In this section PSD examples obtained when using the rate 1/4, $v=4$, RSLF binary convolutional encoders from Tables 6-3 and 6-4, the best rate 1/4, $v=4$, binary convolutional encoder known reported in [83, 111] and a rate equivalent non-coded pulse repetition scheme are presented. The modulation scheme has been chosen to be Q-BOPPM with natural mapping and therefore two consecutive pulses are used to transmit an encoder output vector $z_l = \underline{\mathbf{z}}_l = [z_l^{(0)}, z_l^{(1)}, z_l^{(2)}, z_l^{(3)}]$ (see Section 6.3.1).

The binary data stream at the system input is assumed to be generated by an unbalanced first order BMS for three different combinations of $p_{y,01}$ and $p_{y,10}$:

1). $p_{y,01} = 4/5$ and $p_{y,10} = 3/5$ thus $\pi_{y,0} = 3/7$ and $\pi_{y,1} = 4/7$

2). $p_{y,01} = 1/2$ and $p_{y,10} = 1/2$ thus $\pi_{y,0} = 1/2$ and $\pi_{y,1} = 1/2$

3). $p_{y,01} = 1/5$ and $p_{y,10} = 3/5$ thus $\pi_{y,0} = 3/4$ and $\pi_{y,1} = 1/4$

Therefore the source's steady state statistics are unbalanced for the first and third cases. Note that the second case corresponds to a perfectly random (uniform distributed) i.i.d. binary data stream.

All the convolutionally coded schemes considered in this section transmit two information bits per four pulses. Therefore, in order to provide a meaningful comparison with the convolutionally coded case, four pulses must be used to transmit two information bits for the non-coded pulse repetition scheme. Hence, it will be assumed that each quaternary signal is repeated in four consecutive frames for this case. As in Section 3.5.2, this scheme can be interpreted as a repetition code where every couple of information bits, $\underline{\mathbf{y}}_l = [y_l^{(0)}, y_l^{(1)}]$, is transmitted four times, that is,

$$z_l = \underline{\mathbf{z}}_l = [z_l^{(0)}, z_l^{(1)}, z_l^{(2)}, z_l^{(3)}, z_l^{(4)}, z_l^{(5)}, z_l^{(6)}, z_l^{(7)}] = [y_l^{(0)}, y_l^{(1)}, y_l^{(0)}, y_l^{(1)}, y_l^{(0)}, y_l^{(1)}, y_l^{(0)}, y_l^{(1)}] \tag{6.35}$$

Note the SE pair Markov model for this scheme is equivalent to the one introduced in Section 3.5.2 with the output mapping accordingly changed to consider the repetition of four pulses instead of two.

The PSD plots obtained when the non-coded pulse repetition scheme is used in a Q-BOPPM IR-based UWB system are shown Figure 6-5. For these plots no TH or DS has been considered and therefore $\Lambda = 1$. Note the large amount of spectral lines appearing in Figures 6-5a and 6-5b. This is due to the unbalance in the one step and steady state probabilities of the BMS. It is worth mentioning that although spectral lines are expected at $1/(\Lambda T_s) = 25\text{MHz}$ intervals, they actually

appear at 100 MHz intervals. This is due to nulls present in the absolute value of Equation (6.19) (discrete PSD) for these cases. Finally observe that even though no spectral lines appear for $p_{y,01}=1/2$ and $p_{y,10}=1/2$ in Figure 6-5c, large ripples can be seen in this continuous PSD as shown in the magnification of Figure 6-5c presented in Figure 6-5d.

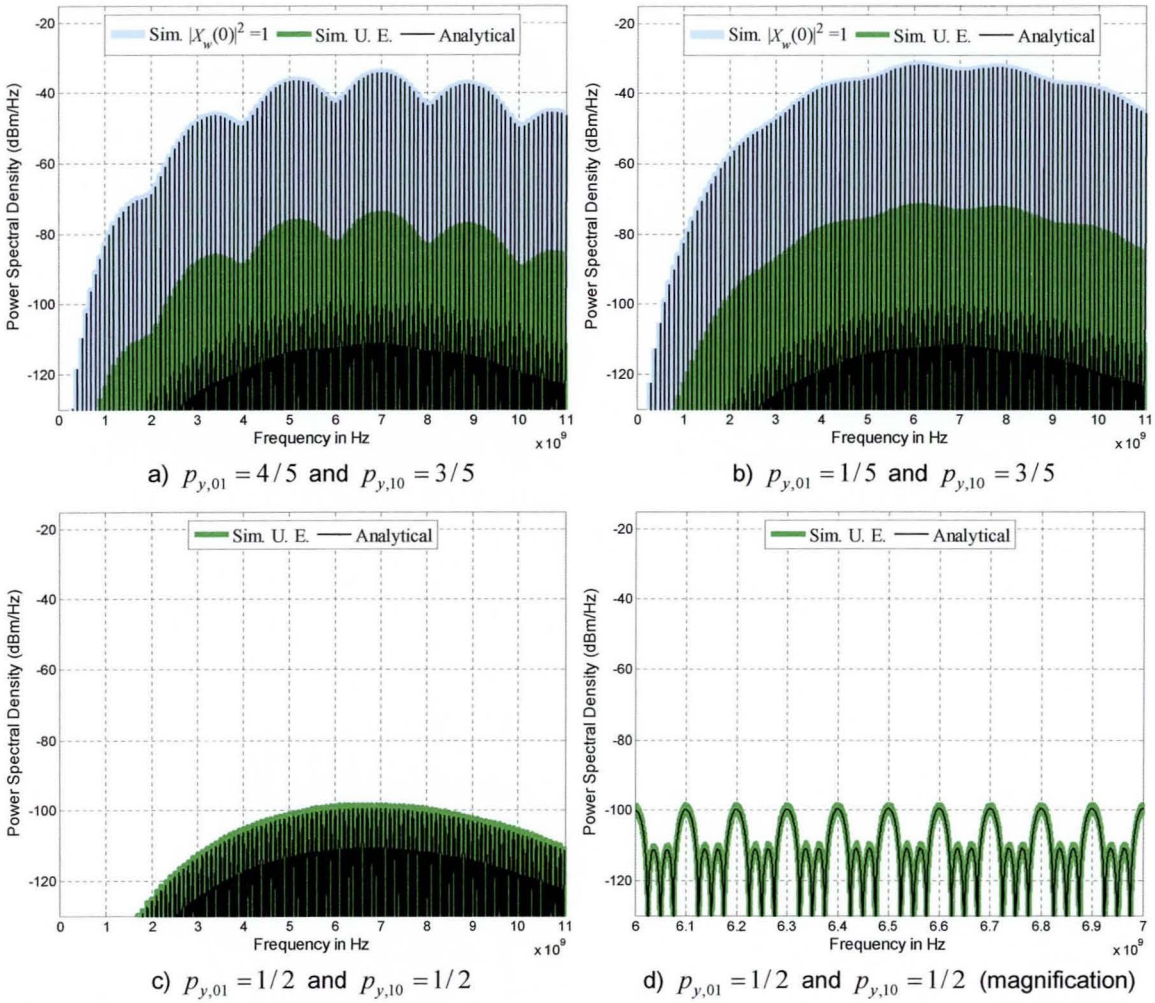


Figure 6-5. Analytical and simulated PSDs for a BMS coupled to a Q-BOPPM IR-based UWB system with four pulses repetition and no convolutional coding. The signal parameters are $T_s=40$ ns, $T_r=10$ ns and $T_\beta=0.5$ ns. The 4th derivative Gaussian pulse is used with duration $T_w\approx 0.4$ ns. No TH or DS is considered.

Now consider the PSD plots shown in Figure 6-6. This plots were obtained by using the best rate 1/4, $v=4$, binary convolutional encoder known. This encoder has generator matrix ([83, 111])

$$\mathbf{G}(D)=[1+D^2+D^4 \quad 1+D^2+D^3+D^4 \quad 1+D+D^3+D^4 \quad 1+D+D^2+D^3+D^4]\Rightarrow(25,27,33,37)_8 \qquad (6.36)$$

and thus is feedforward. As previously mentioned, all the best rate 1/2 and 1/4 binary convolutional encoders known reported in [107, 109, 111] are feedforward and thus do not fulfil the spectral line-free condition but for few specific cases (commonly $p_{y,10}=p_{y,01}=1/2$). This can seen in Figures 6-6a and 6-6b which have a significant amount of spectral lines spaced at 50 MHz intervals.

Even though the number of spectral lines increases in Figure 6-6 compared with Figure 6-5 the overall PSD shape improves upon the introduction of the convolutional encoding operation. This can be seen in Figures 6-6a and 6-6c as the maximum PSD height in these plots is less than in

Figures 6-5a and 6-5c (remember however that in Figures 6-5a and 6-6a the maximum height corresponds to a spectral line whereas in Figures 6-5c and 6-5c corresponds to continuous PSD).

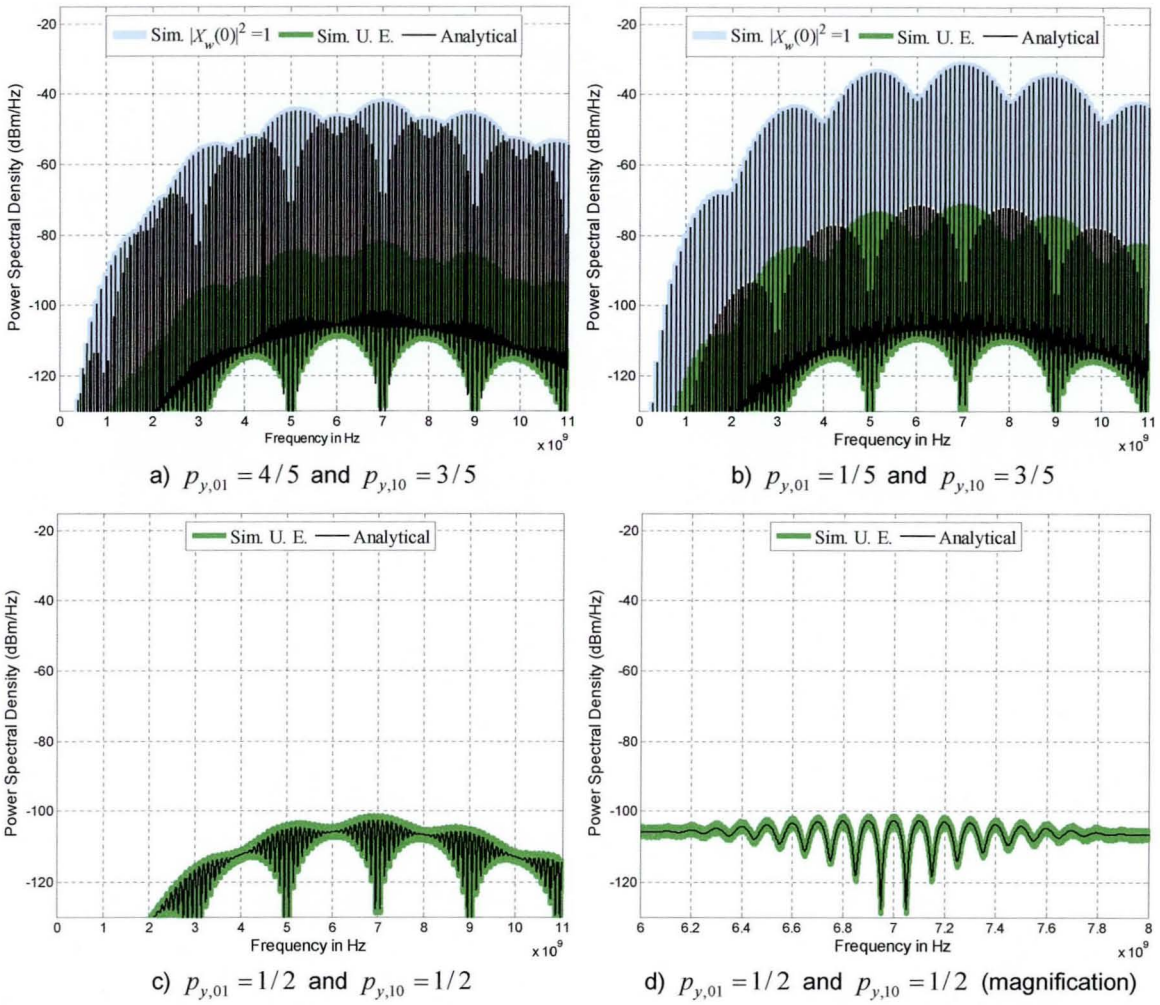


Figure 6-6. Analytical and simulated PSDs obtained when using the optimum distance spectrum feedforward $(25, 27, 33, 37)_8$ binary convolutional encoder in a Q-BOPPM IR-based UWB system. The signal parameters are $T_s = 20$ ns, $T_r = 10$ ns and $T_\beta = 0.5$ ns. The 4th derivative Gaussian pulse is used with duration $T_w \approx 0.4$ ns. No TH or DS is considered.

Note the significant improvement achieved in the PSD characteristics when using the MFD rate 1/4, $v = 4$, RSLF binary convolutional encoders reported in Tables 6-3 and 6-4 as shown in Figures 6-7 and 6-8 respectively. It can be seen in Figure 6-7 that when using the RSLF encoder with feedback polynomial of the form $b(D) = 1 + D^4$ (given in Table 6-3) a spectral line-free PSD is obtained for all three cases. Therefore this system outperforms the non-coded pulse repetition scheme and the system employing the best rate 1/4, $v = 4$, binary convolutional encoder known in terms of PSD characteristics. However, from Figure 6-8 it can be seen that the RSLF encoder with primitive feedback polynomial $b(D) = 1 + D^3 + D^4$ (given in Table 6-4) outperforms all other schemes. Furthermore, even the PSD with the most biased BMS ($p_{y,01} = 1/5$ and $p_{y,10} = 3/5$) in Figure 6-8b is smoother than the PSDs in Figures 6-5c, 6-6c and 6-7c which were obtained considering a balanced BMS generating uniform distributed i.i.d. binary streams.

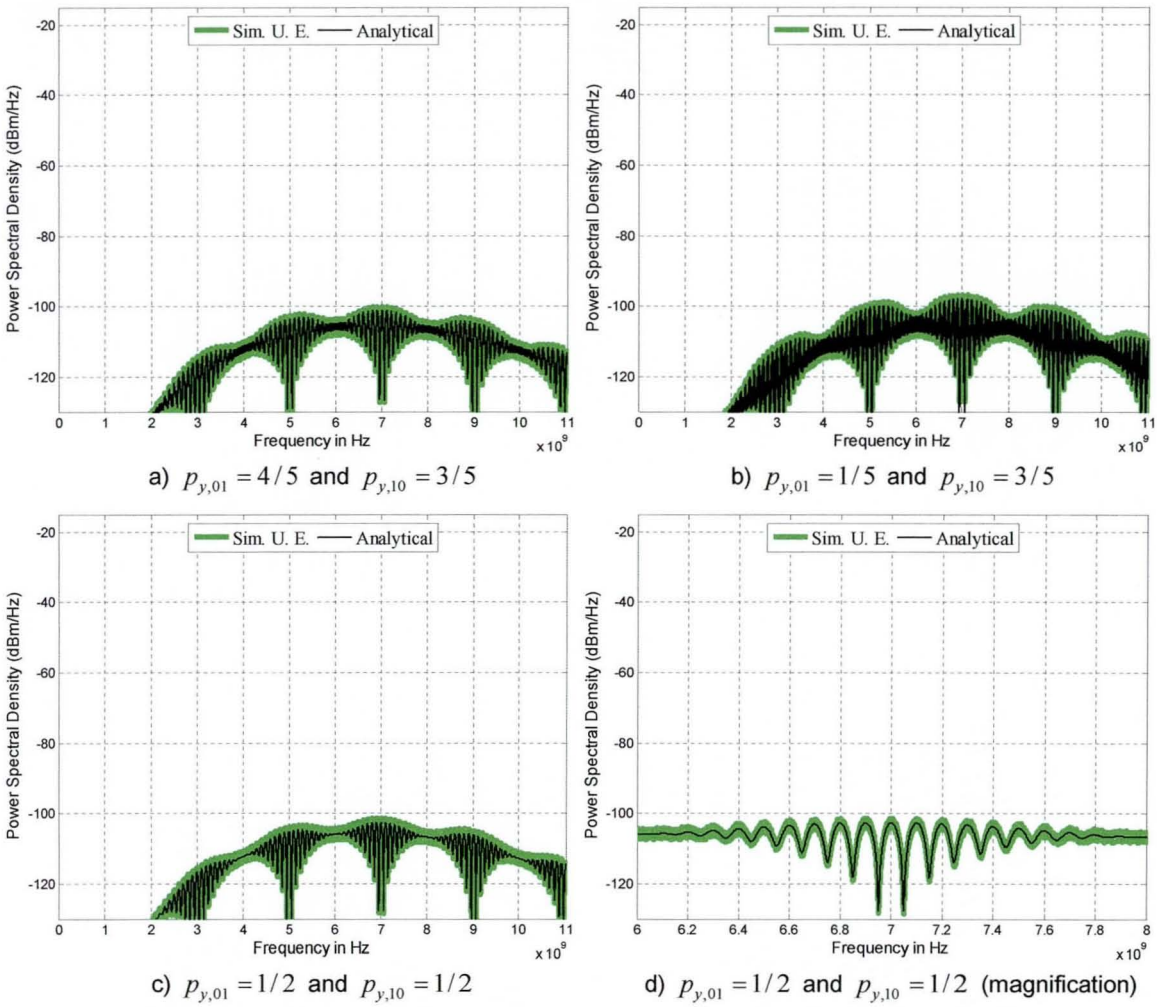
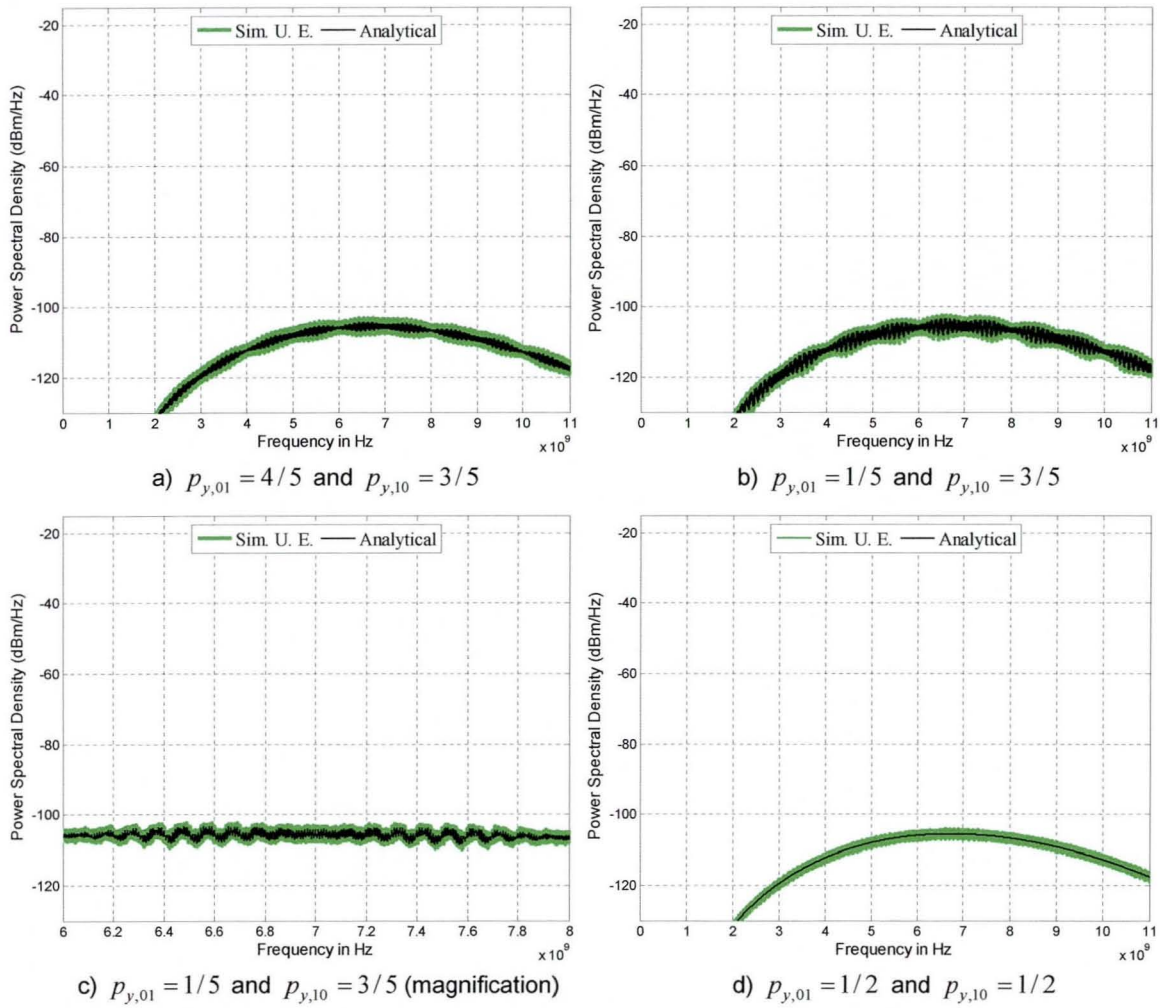


Figure 6-7. Analytical and simulated PSDs for a Q-BOPPM IR-based UWB system using the MFD rate 1/4, $v=4$, RSLF binary convolutional encoder from Table 6-3 with feedback polynomial $b(D)=1+D^4 \Rightarrow 21_8$. The signal parameters are $T_s=20$ ns, $T_r=10$ ns and $T_\beta=0.5$ ns. The 4th derivative Gaussian pulse is used with duration $T_w \approx 0.4$ ns. No TH or DS is considered.

Let us now consider what happens when a short PR-DS sequence is used for pulse polarity randomisation purposes. In this case the Barker sequence {+1, -1, +1, +1, -1, +1, +1, +1, -1, -1, -1} with period $\chi_a=11$ used in the IEEE 802.11 standard, [151], will be considered. Figure 6-9 shows the PSD plots obtained when using the non-coded pulse repetition scheme in a Q-BOPPM DS-IR-based UWB system with an unbalanced BMS with $p_{y,01}=1/5$ and $p_{y,10}=3/5$ ($\pi_{y,0}=3/4$ and $\pi_{y,1}=1/4$). It can be seen that although the maximum spectral line height has been reduced in Figure 6-9 compared to Figure 6-5c, the number of spectral lines has increased upon the introduction of the PR-DS as they are now spaced at 9.09 MHz intervals. Moreover, note that even though the use of the Barker sequence helps to improve the PSD characteristics for the non-coded scheme, the PSD in Figure 6-9 is still far from the spectral line-free PSDs shown in Figures 6-7b and 6-8b which were obtained using the RSLF convolutional encoders reported in Tables 6-3 and 6-4.

In comparison Figure 6-10 shows the PSD plot obtained when using the MFD rate 1/4, $v=4$, RSLF binary convolutional encoder reported in Table 6-4 in a Q-BOPPM DS-IR-based UWB system with an unbalanced BMS with $p_{y,01}=1/5$ and $p_{y,10}=3/5$. Comparing Figure 6-8b (where

no DS is used) and Figure 6-10 (where the Barker PR-DS is used) it can be seen that the use of the Barker sequence helps to further smooth the PSD.



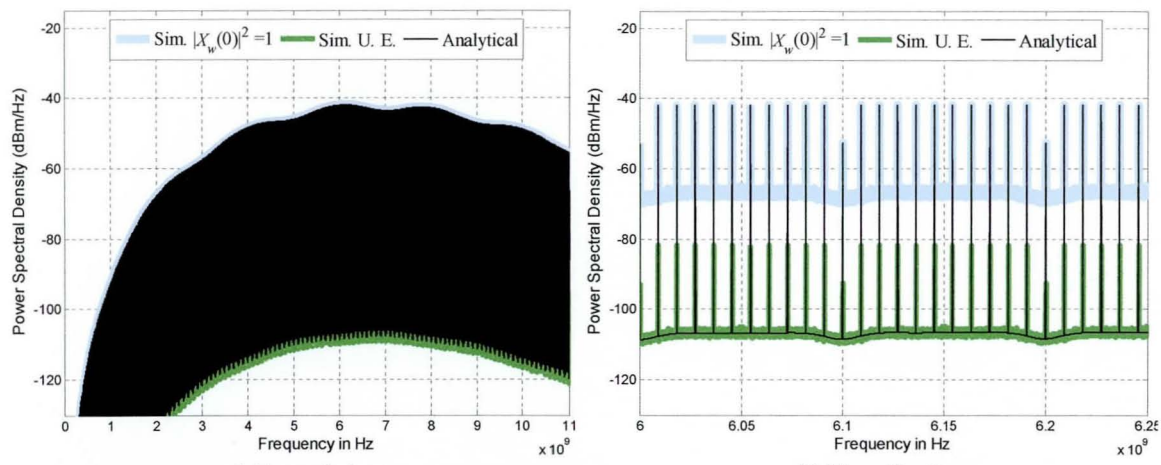
a) $p_{y,01} = 4/5$ and $p_{y,10} = 3/5$

b) $p_{y,01} = 1/5$ and $p_{y,10} = 3/5$

c) $p_{y,01} = 1/5$ and $p_{y,10} = 3/5$ (magnification)

d) $p_{y,01} = 1/2$ and $p_{y,10} = 1/2$

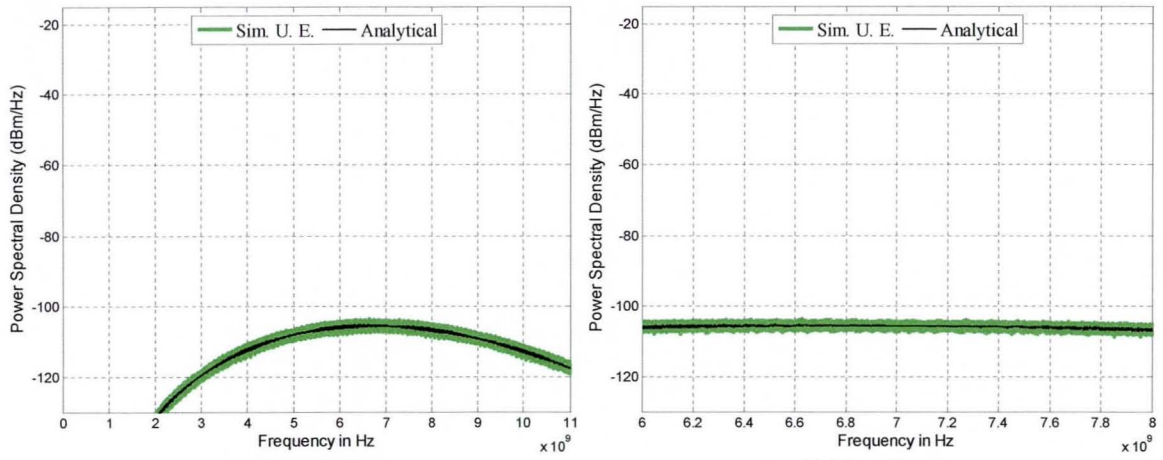
Figure 6-8. Analytical and simulated PSDs for a Q-BOPPM IR-based UWB system using the MFD rate 1/4, $v = 4$, RSLF binary convolutional encoder from Table 6-4 with feedback polynomial $b(D) = 1 + D^3 + D^4 \Rightarrow 23_8$. The signal parameters are $T_s = 20$ ns, $T_r = 10$ ns and $T_\beta = 0.5$ ns. The 4th derivative Gaussian pulse is used with duration $T_w \approx 0.4$ ns. No TH or DS is considered.



a) Normal view

b) Magnification

Figure 6-9. Analytical and simulated PSDs for a BMS coupled to a Q-BOPPM IR-based UWB system with four pulses repetition and no convolutional coding. The BMS probabilities are $p_{y,01} = 1/5$ and $p_{y,01} = 3/5 \Rightarrow \pi_{y,0} = 3/4$ and $\pi_{y,1} = 1/4$. The signal parameters are $T_s = 40$ ns, $T_r = 10$ ns and $T_\beta = 0.5$ ns. The 4th derivative Gaussian pulse is used with duration $T_w \approx 0.4$ ns. The periodic Barker sequence {+1, -1, +1, +1, -1, +1, +1, +1, -1, -1, -1} is used for DS pulse polarity randomisation purposes. No TH is considered.



a) Normal view b) Magnification

Figure 6-10. Analytical and simulated PSDs for a Q-BOPPM IR-based UWB system using the MFD rate 1/4, $v = 4$, RSLF binary convolutional encoder from Table 6-4 with feedback polynomial $b(D) = 1 + D^3 + D^4 \Rightarrow 23_8$. The BMS probabilities are $p_{y,01} = 1/5$ and $p_{y,01} = 3/5 \Rightarrow \pi_{y,0} = 3/4$ and $\pi_{y,1} = 1/4$. The signal parameters are $T_s = 20$ ns, $T_r = 10$ ns and $T_\beta = 0.5$ ns. The 4th derivative Gaussian pulse is used with duration $T_w \approx 0.4$ ns. The periodic Barker sequence {+1, -1, +1, +1, -1, +1, +1, +1, -1, -1, -1} is used for DS pulse polarity randomisation purposes. No TH is considered.

Finally note how there is good agreement between the analytical and simulated results for both the PSD's continuous part (Sim. U.E.) and the PSD's discrete part (Sim. $|X_w(0)|^2 = 1$) in all the figures presented in this section.

6.6.2 Peak to Average Ratio for Some of the RSLF Binary Convolutional Encoders from Tables 6-1 to 6-4

Comparing Figures 6-7 and 6-8 it can inferred that better PSD characteristics can be obtained when using the RSLF binary convolutional encoders with primitive feedback polynomials reported in Tables 6-2 and 6-4. In order further to see this, the peak to average ratio in dB calculated by using Equations (6.14) to (6.19) is reported in Tables 6-10 and 6-11 for the first five RSLF convolutional encoders introduced in Tables 6-1 to 6-4.

The peak to average ratio was calculated for the case of an unbalanced BMS with $p_{y,01} = 1/5$ and $p_{y,10} = 3/5$ corresponding to $\pi_{y,0} = 3/4$ and $\pi_{y,1} = 1/4$. For generality no specific UWB pulse shape, $w(t)$, has been considered. Instead a flat shape for $|W(f)|^2$ has been assumed (e.g. $w(t)$ is an impulse function $A\delta(t)$) such that the PSD's average level is normalised to unity. For Table 6-10 the modulation format used was BPSK while for Table 6-11 the modulation format used was Q-BOPPM. The general signal parameters are $T_r = 10$ ns, $T_\beta = 0.5$ ns, no TH and no DS. The peak to average ratio was calculated by using

$$\text{Peak to Average Ratio (dB)} = 10\log_{10}\left(\frac{\max(\bar{S}(f))}{\text{mean}(\bar{S}(f))}\right) \tag{6.37}$$

where $\bar{S}(f)$ is the analytical PSD defined by Equations (6.14) to (6.19). In practical terms the peak to average ratio was obtained by calculating the analytical PSD, $\bar{S}(f)$, in the range of 4 GHz to 8 GHz with a resolution of 200 kHz.

The peak to average ratios calculated for the first 5 rate 1/2 RSLF binary convolutional encoders reported in Tables 6-1 and 6-2 are presented in Table 6-10 for BPSK signalling. It can be seen that the RSLF convolutional encoders with primitive feedback polynomials (Table 6-2) have lower peak to average ratio than the encoders with $b(D)=1+D^v$ feedback polynomials (Table 6-1). Furthermore, note how in general by increasing the total encoder memory, v, the peak to average ratio is reduced. As a reference, the last column of Table 6-10 reports the maximum spectral line eight of PSDs obtained when using the best rate 1/2 binary convolutional encoders known reported in [111] under the same operation conditions.

Table 6-10. Peak to average ratios (dB) for the first 5 rate 1/2 RSLF binary convolutional encoders reported in Tables 6-1 and 6-2. It is assumed that BPSK is used and that the data stream is generated by an unbalanced BMS with $p_{y,01} = 1/5$ and $p_{y,01} = 3/5$.

v	Peak to average ratio (dB) of the encoders reported in ***Table 6-1*** $b(D) = 1 + D^v$	Peak to average ratio(dB) of the encoders reported in ***Table 6-2*** (primitive feedback polynomial)	Maximum spectral line height (dB) of the best rate 1/2 encoders known reported in [111]
3	2.404	1.977	65.933
4	2.113	0.535	64.690
5	1.129	1.003	60.688
6	0.915	0.044	59.129
7	0.661	0.000	56.752

The peak to average ratios calculated for the first 5 rate 1/4 RSLF binary convolutional encoders reported in Tables 6-3 and 6-4 are presented in Table 6-11 for Q-BOPPM signalling with natural mapping. As in the previous case the RSLF convolutional encoders with primitive feedback polynomials (Table 6-4) have lower peak to average ratio than ones with $b(D)=1+D^v$ feedback polynomials (Table 6-3). *This is the reason why the encoders from Table 6-4 were used as the parent codes for the nested code search of rate 1/6 to 1/32 RSLF convolutional encoders* (Tables 6-5 to 6-9). Note from Table 6-11 how the peak to average ratio is reduced as the total encoder memory, v, increases.

Table 6-11. Peak to average ratios (dB) for the first 5 rate 1/4 RSLF binary convolutional encoders reported in Tables 6-3 and 6-4. It is assumed that Q-BOPPM is used and that the data stream is generated by an unbalanced BMS with $p_{y,01} = 1/5$ and $p_{y,01} = 3/5$.

v	Peak to average ratio (dB) of the encoders reported in ***Table 6-3*** ($b(D) = 1 + D^v$ polynomial) with natural mapping	Peak to average ratio(dB) of the encoders reported in ***Table 6-4*** (primitive feedback polynomial) with natural mapping	Peak to average ratio(dB) of the encoders reported in ***Table 6-4*** (primitive feedback polynomial) with Gray mapping
3	6.397	4.760	4.312
4	7.678	2.055	1.754
5	7.550	1.576	1.356
6	3.021	0.592	0.405
7	1.745	0.177	0.190
8	1.962	0.000	0.053

The last column in Table 6-11 reports the peak to average ratios obtained when using Gray mapping based Q-BOPPM and the RSLF convolutional encoders from Table 6-4. This column was included due to the fact that binary convolutional codes are often used with Gray mapping based quaternary phase shift keying (QPSK), [83, 85], which is geometrically equivalent to Q-BOPPM

with Gray mapping as explained in Section 2.3.1. Note from the last two columns in Table 6-11 that the change from natural to Gray mapping has little or no significant effects on peak to average ratio achieved with the RSLF encoders reported in Table 6-4.

6.7 BER Performance Comparisons of RSLF Binary Convolutional Encoders with the Best Binary Convolutional Encoders Known and the Non-Coded Pulse Repetition Scheme

In this section comparisons in terms of bit error rate (BER) performance are provided for soft Viterbi decoding (SVD) when using the rate 1/4, $v=4$ RSLF binary convolutional encoder from Table 6-4, the best rate 1/4, $v=4$ binary convolutional encoder known reported in [83, 111] (defined by Equation (6.36)) and a rate equivalent non-coded scheme. Bit error rate (BER) results are presented for additive white Gaussian noise (AWGN) channel and multipath channel. The base system used in this section for BER comparison purposes is described by the block diagram shown in Figure 6-11.

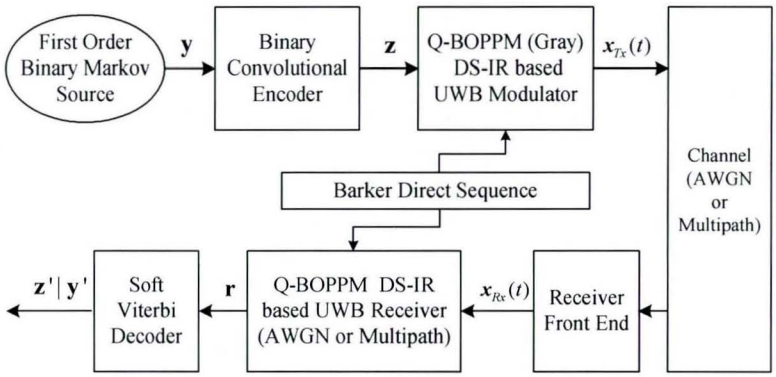


Figure 6-11. Block diagram of a binary convolutionally coded Q-BOPPM (Gray) IR-based UWB system.

Note the system shown in Figure 6-11 uses Q-BOPPM with Gray mapping. This kind of signalling was chosen due to Q-BOPPM with Gray mapping is geometrically equivalent to QPSK with Gray mapping, [83, 87], and the BER performances of convolutionally coded BPSK and convolutionally coded QPSK with Gray mapping in AWGN channel are similar, [83, 85]. Therefore the results presented in this section for AWGN channel are representative of the BER performance achievable with both convolutionally coded BPSK and convolutionally coded Q-BOPPM (Gray) IR-based UWB systems.

Using the same notation as in Section 2.7.5 the data source produces a binary sequence $\mathbf{y}=\{y_0, y_1, ..., y_{N-1}\}$ of length N which defines a path generating the code sequence $\mathbf{z}=\{\underline{\mathbf{z}_0}, \underline{\mathbf{z}_1}, ..., \underline{\mathbf{z}_{N-1}}\}=\{z_0^{(0)}, ..., z_0^{(3)}, ..., z_{N-1}^{(0)}, ..., z_{N-1}^{(3)}\}$ where $z_l^{(i)} \in \{0,1\}$, $i=0,1,2,3$ and $l=0,...,N-1$. Next every couple of code bits $[z_l^{(2k)}, z_l^{(2k+1)}]$ (where $k=0,1$) is transmitted using a Q-BOPPM IR-

based UWB system with Gray mapping. Thus the transmitted signal of this system has the following general form

$$x_{Tx}(t) = \sum_{l=0}^{N-1} \sum_{k=0}^{N_w-1} a_{lN_w+k} \alpha_{\sigma_l,k} w(t - \beta_{\sigma_l,k} T_\beta - lT_s - kT_r) \tag{6.38}$$

where all the variables have been defined in Section 6.3.1 and $w(t)$ has energy E_w. Due to Gray mapping based Q-BOPPM is assumed, $\alpha_{i,k} \in \{-1,1\}$ and $\beta_{i,k} \in \{0,1\}$ are defined according to Equation (6.13). Note that no PR-TH sequence has been considered and $\{a_{lN_w+k}\}$ is set to be Barker PR-DS {+1, -1, +1, +1, -1, +1, +1, +1, -1, -1, -1} with period χ_a = 11. The general signal parameters are as follows: $T_s = 20$ ns, $T_r = 10$ ns and $T_\beta = 0.5$ ns. The 4th derivative Gaussian pulse is used with approximate duration $T_w \approx 0.4$ ns as in the previous section.

The equations describing $x_{Rx}(t)$ for AWGN and multipath channels are introduced in Sections 6.7.1 and 6.7.2 respectively. The BER performance results presented in these sections were obtained through simulation assuming coherent detection and perfect DS synchronisation between transmitter and receiver.

For each particular channel the receiver should be able to fed the decoder with a received sequence $\mathbf{r} = \{\underline{\mathbf{r}_0}, \underline{\mathbf{r}_1}, ..., \underline{\mathbf{r}_{N-1}}\} = \{r_0^{(0)}, ..., r_0^{(3)}, ..., r_{N-1}^{(0)}, ..., r_{N-1}^{(3)}\}$ which consists of the soft metrics to be used by the soft Viterbi decoder (SVD), [83]. Using the received sequence, **r**, the SVD produces an estimate, $\mathbf{z}' = \{z'^{(0)}_0, ..., z'^{(\kappa-1)}_0, ..., z'^{(0)}_{N-1}, ..., z'^{(\kappa-1)}_{N-1}\}$, of the transmitted sequence **z** and an estimate, $\mathbf{y}' = \{\mathbf{y}'_0, \mathbf{y}'_1, ..., \mathbf{y}'_{N-1}\}$, of the information sequence, **y**, which is then used to compute the BER performance of the system.

6.7.1 BER Comparisons of the Rate 1/4, v = 4, RSLF Binary Convolutional Encoder with the Best Rate 1/4, v = 4, Binary Convolutional Encoder Known and Non-Coded Signals in AWGN Channel

In the AWGN channel the transmitted signal is only affected by the additive noise an therefore the received signal can be described by

$$x_{Rx}(t) = \sum_{l=0}^{N-1} \sum_{k=0}^{N_w-1} a_{lN_w+k} \alpha_{\sigma_l,k} w(t - \beta_{\sigma_l,k} T_\beta - lT_s - kT_r + \tau) + n(t) \tag{6.39}$$

where τ is the delay and $n(t)$ is AWGN with zero mean and two sided power spectral density $N_0/2$. As previously mentioned, it will be assumed that τ is known at the receiver (coherent reception) and that perfect DS synchronisation has been achieved between transmitter and receiver.

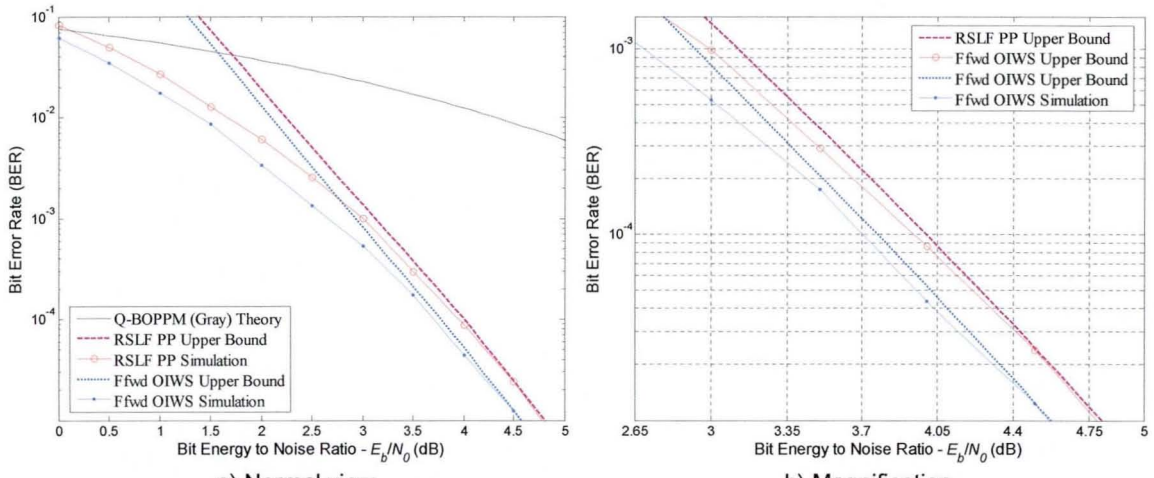
Under the previous assumptions the results and analysis presented in the literature ([83, 101, 103]) for BER performance of binary convolutional codes with BPSK and QPSK (Gray) signalling in AWGN channels are straightforwardly applicable to the analysis of the convolutionally coded

Q-BOPPM (Gray) IR-based UWB system under study. An upper bound on the bit error probability achieved with convolutionally coded BPSK/QPSK (Gray) in AWGN channels with coherent detection and soft Viterbi decoding (SVD) is given by [83]:

$$P_b \leq \sum_{d=d_{free}}^{\infty} B_d P_2(d) = \sum_{d=d_{free}}^{\infty} B_d Q\left(\sqrt{\frac{2E_b R_c}{N_0} d}\right) \tag{6.40}$$

where E_b is the information bit energy, $R_c = 1/\kappa$ is the code rate and $\{B_d\}$ is the encoder's IWS. Thus the upper bound defined by Equation (6.40) applies for convolutionally coded Q-BOPPM (Gray) IR-based UWB systems as previously explained. Note that for a rate 1/4 encoder with Q-BOPPM signalling $E_b = 2E_w$ as two Q-BOPPM signals are needed to transmit the four code bits generated by the encoder per each information bit.

Simulated BER plots for the system described by Figure 6-11 in AWGN channel are shown in Figure 6-12 altogether with the corresponding upper bounds calculated by using Equation (6.40). The plots labelled as "RSLF PP" were obtained by using the rate 1/4, $v = 4$, RSLF binary convolutional encoder with primitive feedback polynomial from Table 6-4. The BER plots corresponding to the best rate 1/4, $v = 4$, binary convolutional encoder known (defined by Equation (6.36)) are labelled as "Ffwd OIWS" (as this encoder has optimum IWS). The upper bounds were calculated considering the first 20 elements of the IWS which give a good estimate for BER values below 10^{-3} as seen in Figure 6-12 (as reference consider that the upper bounds reported in [135], [110] and [111] were calculated using the first 8, 13 and 15 elements of the IWS respectively).



a) Normal view b) Magnification

Figure 6-12. BER performance in AWGN channel for SVD of Q-BOPPM DS-IR-based UWB systems using the rate 1/4, $v = 4$, RSLF convolutional encoder with primitive feedback polynomial (RSLF PP) and the best rate 1/4, $v = 4$, convolutional encoder known with optimum IWS (Ffwd OIWS). The theoretical BER plot for non-coded Q-BOPPM DS-IR-based UWB systems is provided as a reference. All the plots were obtained assuming coherent reception and perfect DS synchronisation.

Remember from Section 2.3.1 that the theoretical bit error probability for non-coded Q-BOPPM IR-based UWB systems with Gray code mapping can be found to be ([83, 87, 88])

$$P_b = Q\left(\sqrt{\frac{E_s}{N_0}}\right) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) = Q\left(\sqrt{\frac{N_p E_p}{N_0}}\right) \tag{6.41}$$

Thus Equation (6.41) was used to obtain the plot labelled as "Q-BOPPM (Gray) Theory" in Figure 6-12a.

It can be seen in Figure 6-12b that the performance loss when using the RSLF convolutional encoder instead of the best convolutional encoder known is less than 0.35 dB for bit error rates below 10^{-3}. This performance loss is expected as the encoder defined by Equation (6.36) has optimum IWS which is superior to the IWS of the RSLF encoder. However, in systems with unbalanced BMS this slight performance loss is compensated by the improved PSD characteristics obtained with the RSLF encoders, as shown in Section 6.6.

Note how both convolutionally coded schemes outperform the non-coded scheme by approximately 2.5 dB at BERs below 10^{-2}. Obviously this increase in BER performance is achieved at a cost of increased system complexity for the convolutionally coded schemes. However note how the complexity of both convolutionally coded systems is the same as both encoders have the same total encoder memory, $v = 4$, and hence have the same decoding complexity (the Viterbi decoder has 16 states for both encoders). Furthermore, this increase in the system complexity can be regarded to be acceptable as the de facto industry standard convolutional encoder with generator matrix $(133, 171)_8$, [104], has total encoder memory $v = 6$ and therefore the Viterbi decoder for this encoder has 64 states.

Finally it is worth mentioning that through the use of the upper bound defined by Equation (6.40), it was verified that the BER performance loss of the RSLF binary convolutional encoders reported in Tables 6-2 and 6-4 compared to the best rate 1/2 and rate 1/4 binary convolutional encoders known reported in [111] should be less than 0.4 dB for BER values between 10^{-3} and 10^{-6} (the upper bounds were calculated by considering the first 15 elements of the IWS of each code as in [111]).

6.7.2 BER Comparisons of the Rate 1/4, $v = 4$, RSLF Binary Convolutional Encoder with the Best Rate 1/4, $v = 4$, Binary Convolutional Encoder Known and Non-Coded Signals in Multipath Channel

A channel commonly used in the literature (e.g. [13, 15, 17, 19]) as an example of a typical UWB multipath channel model is the IEEE 802.15.3a UWB channel model, [152] [153]. Thus this model will be used in this section to provide a BER performance comparison in a multipath affected channel between the RSLF binary convolutional encoders and the best binary convolutional encoders known.

The impulse response of the IEEE 802.15.3a channel model is described by ([13, 17, 152, 153]):

$$h_i(t) = X_i \sum_{l=1}^{L^i} \sum_{k=1}^{K_l^i} \varphi_{k,l}^i \delta(t - T_l^i - \tau_{k,l}^i) = \sum_{n=1}^{N_{MPath}} h_n^i \delta(t - \tau_n^i) \tag{6.42}$$

where the subscript i stands for the channel realisation; X_i represents log-normal shadowing; L^i is the number of clusters; K_l^i is the number of multipath components received in the l^{th} cluster;

N_{MPath} is the total number of multipath components; $\varphi_{k,l}^{i}$ are the multipath coefficients of the k^{th} multipath component of the l^{th} cluster; T_l^i is the l^{th} cluster arrival time and $\tau_{k,l}^{i}$ is the delay of the k^{th} multipath component within the l^{th} cluster. The channel model considers four different indoor scenarios and thus defines four different sets of statistics for each one of them. This work considers channel model 1 (CM1) corresponding to a line of sight (LOS) scenario within 0-4 m, [13, 152, 153]. The reader is referred to [13, 15, 17, 19, 152, 153] for a detailed explanation of this channel model and its implementation in a discrete simulation. A summary of the model's main parameters provided in [153] is reproduced in Appendix B.

In the simulations a quasi-static scenario was considered, [13, 29, 154], meaning that the channel's multipath statistics remain unchanged during the transmission of an information block consisting of $N=1024$ information bits or equivalently 4096 code bits. Thus if the transmitted signal is described by Equation (6.38) then for the i^{th} channel realisation the received signal can be expressed as:

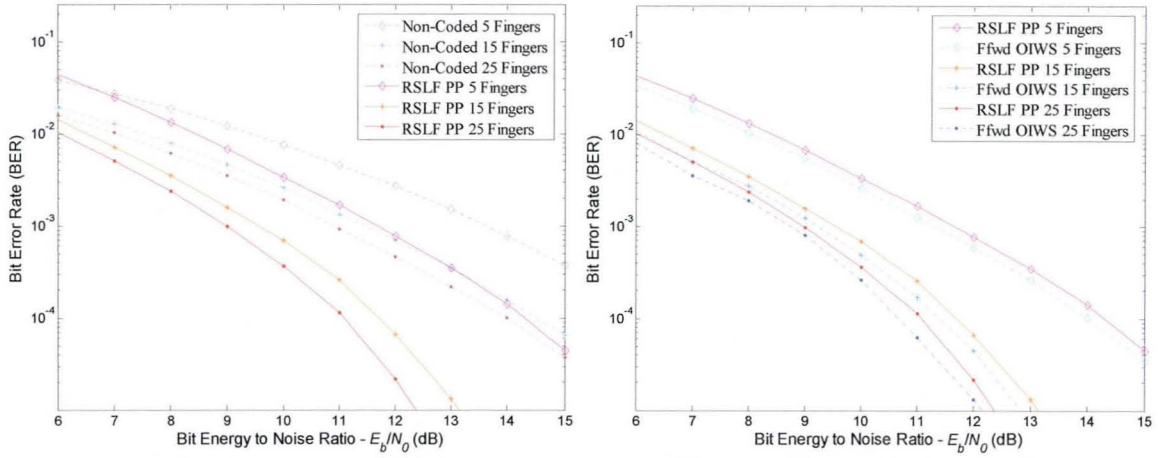
$$x_{Rx}(t)=\sum_{n=1}^{N_{MPath}}\sum_{l=0}^{N-1}\sum_{k=0}^{N_w-1} a_{lN_w+k}\alpha_{\sigma_l,k}h_n^i w(t-\beta_{\sigma_l,k}T_\beta-lT_s-kT_r-\tau_n^i)+n(t) \tag{6.43}$$

where $w(t)$ has energy E_w and $n(t)$ is AWGN with zero mean and two sided power spectral density $N_0/2$. Note that the transmitted bit energy is $E_b=2E_w$ as in the previous case.

Note from Equation (6.43) that for this kind of channels the transmitted pulse energy, E_w, is distributed over N_{MPath} multipath arrivals, [13, 15, 17, 19, 152, 153]. Thus in order to collect as much energy as possible a RAKE receiver must be used, [13, 15, 17, 19, 152, 153]. An ideal RAKE receiver would combine the energy from all N_{MPath} multipath components through a set of N_{MPath} correlators (usually referred as "fingers"), one per multipath arrival [13, 15, 17, 152, 153]. However this ideal all-RAKE (ARAKE) receiver may be too complicated to implement in practice due to the large number of multipath components. Thus simplified structures collecting energy from a fixed number, N_{fin}, of multipath components must be considered, [13, 15, 17 153]. A practical RAKE implementation which only collects energy from the N_{Sfin} strongest paths is called a selective RAKE (SRAKE) whereas a RAKE implementation that collects energy from the first arriving N_{Pfin} paths is called a partial RAKE (PRAKE). For the simulation results presented in this section the SRAKE approach has been adopted as in general it provides better performance than PRAKE implementations, [13, 15]. As well it has been assumed that the receiver has perfect knowledge of the channel's impulse response (perfect channel side information - CSI), $h_i(t)$, during the transmission of an information block. Finally the strategy adopted for the combination of the energy captured by the SRAKE fingers has been chosen to be maximal ratio combining (MRC), which is optimal when the noise is AWGN and the inter-symbol interference (ISI) is negligible, [13, 15, 17, 83]. Briefly, MRC applies a weight to every received multipath component before combining, such that the stronger signals have larger weight than the weak signals, [13, 83].

In our simulations 1000 different realisations of the IEEE 802.15.3a UWB line of sight (LOS) channel model 1 (CM1) were generated with a sampling time of 0.2 ns. The average characteristics of these realisations are: mean excess delay = 5.12 ns; rms delay spread = 5.56 ns; number of paths within 10 dB of the strongest path = 11.87 and number of paths that capture 85% of channel energy = 15.73. At least one information block is transmitted per channel realisation for BER values above 10^{-3}. For BER values below 10^{-3} more than one information block was transmitted per channel realization (the same amount of information blocks were transmitter for every channel realisation) such that a minimum of 500 bit errors were counted per BER point. Finally note that due to $T_r = 10$ ns, $T_\beta = 0.5$ ns and $T_w \approx 0.4$ there may exist inter-pulse interference (IPI) and ISI, [17, 154].

Simulated BER plots for the system described by Figure 6-11 in the IEEE 802.15.3a UWB line of sight (LOS) channel model 1 are shown in Figure 6-12 for different SRAKE configurations. The plots are presented in terms of the originally transmitted bit energy $E_b = 2E_w$ to noise ratio (E_b / N_0). The plots labelled as "RSLF PP" where obtained by using the rate 1/4, $v = 4$, RSLF binary convolutional encoder with primitive feedback polynomial from Table 6-4. The plots corresponding to the best rate 1/4, $v = 4$, binary convolutional encoder known (defined by Equation (6.36)) with optimum IWS are labelled as "Ffwd OIWS". The plots labelled as "Non-Coded" correspond to a non-coded rate equivalent Q-BOPPM DS-IR-based UWB system. The number of SRAKE fingers used for every BER plot is indicated in the legend.



a) Non-Coded vs. RSLF Code b) Feedforward OIWS Code vs. RSLF Code

Figure 6-13. BER performance in IEEE 802.15.3a UWB channel model 1 (CM1) for SVD of Q-BOPPM DS-IR-based UWB systems using the rate 1/4, $v = 4$, RSLF convolutional encoder with primitive feedback polynomial (RSLF PP), the best rate 1/4, $v = 4$, convolutional encoder known with optimum IWS (Ffwd OIWS) and a rate equivalent non-coded Q-BOPPM DS-IR-based UWB system. Results are presented for several SRAKE receivers with different numbers of fingers (N_{Sref}). Perfect channel side information and DS synchronisation was assumed.

It can be seen in Figure 6-13a that the RSLF convolutional encoder outperforms the non-coded system by about 2 dB at BERs below 10^{-3} and by more than 3 dB at a BERs below 10^{-4}. Again the price for this BER improvement over the non-coded scheme is the increased complexity of the convolutionally coded system. As well from Figure 6-13b it can be verified that the performance

loss when using the RSLF convolutional encoder instead of the best convolutional encoder known is less than 0.5 dB for BER values below 10^{-2}.

6.8 Conclusions for Chapter 6

In this chapter recursive spectral line-free (RSLF) binary convolutional encoders for unbalanced (non-uniform distributed) binary Markov sources (BMS) have been introduced.

First it was proved that traditional feedforward structures do not provide a good framework for the design of spectral line-free (SLF) encoders when the binary input stream is generated by an unbalanced first order BMS. Particularly, it has been demonstrated analytically that binary convolutional encoders with feedforward structures do not fulfil the spectral line-free condition for all combinations of BMS' one step transition probabilities, $0 < p_{y,ij} < 1$.

In contrast, it was shown that recursive structures are better suited for the design of SLF encoders as the steady state probabilities of the source encoder (SE) pair Markov model have a quite regular distribution which only depends on the BMS's steady state probabilities. In fact these structures enable the fulfilment of the spectral line-free condition for all combinations of $0 < p_{y,ij} < 1$ as long as the output mapping function, $\gamma(\cdot)$, is properly designed.

Based on the analysis a code search procedure to seek RSLF binary convolutional encoders with the best possible information weight spectrum (IWS) was introduced. The code search procedure verifies that the spectral line-free condition is satisfied when the binary data stream is generated by an unbalanced first order BMS and the modulation format used is BPSK, Q-BOPPM with natural mapping or Q-BOPPM with Gray mapping. Using this code search procedure novel maximum free distance RSLF binary convolutional encoders were found. Tables of the best RSLF encoders found were provided for recursive structures with $b(D) = 1 + D^v$ feedback polynomials and recursive structures with primitive feedback polynomials for encoders with data rates equal to 1/2 and 1/4. For encoders with data rates $1/\kappa < 1/4$ a nested code search approach was adopted where the parent codes used were the rate 1/4 encoders with primitive feedback polynomials. All these encoders provide a spectral line-free PSD when used in BPSK or Q-BOPPM (natural and Gray mappings) IR-based UWB systems even when the input consists of a binary data stream generated by an unbalanced first order BMS.

Power spectral density examples of Q-BOPPM IR-based UWB signals using rate 1/4, v = 4, RSLF convolutional encoders, the best rate 1/4, v = 4, convolutional encoders known and a rate equivalent non-coded system were provided. These examples showed that the RSLF encoders produce spectral line-free PSD even when the binary data stream is generated by an unbalanced BMS. In contrast, the other two schemes generate signals with a significant amount of spectral lines in the PSD under the same operating conditions (the exception was for $p_{y,01} = 1/2$ and $p_{y,10} = 1/2$ as expected). Furthermore, it was shown that for a biased BMS with $p_{y,01} = 1/5$ and

$p_{y,10} = 3/5$ (thus $\pi_{y,0} = 3/4$ and $\pi_{y,1} = 1/4$) the use of the Barker DS with period 11 for polarity randomisation purposes does not eliminate the spectral lines for the non-coded pulse repetition scheme (although it does help to decrease the maximum spectral line height whilst increasing the number of spectral lines). In contrast the use of the same sequence combined with the rate 1/4, $v = 4$, RSLF encoder with primitive feedback polynomial helps to further smooth the signal's PSD.

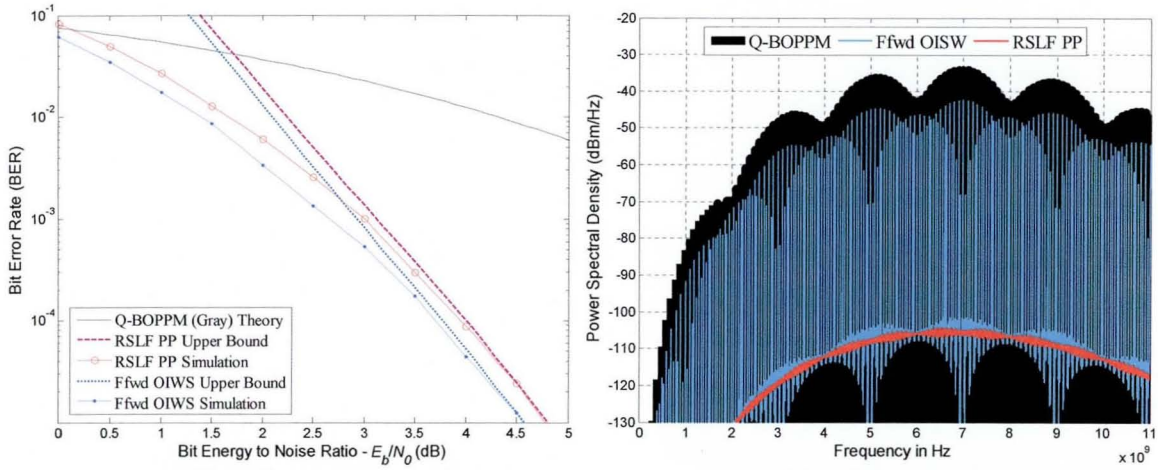
From the PSD examples and the peak to average ratio introduced in Section 6.6.2, it can be concluded that the RSLF convolutional encoders with primitive feedback polynomials should be preferred over the RSLF convolutional encoders with $b(D) = 1 + D^v$ feedback polynomials if the design priority is to obtain a PSD as smooth as possible.

Lastly comparisons of the bit error rate (BER) performance achieved with RSLF binary convolutional encoders in AWGN channel and in a typical UWB multipath channel have been provided for soft Viterbi decoding. These BER plots showed that the performance loss of the rate 1/4, v = 4, RSLF binary convolutional encoder compared to the best rate 1/4, v = 4, binary convolutional encoder known, is less than 0.5 dB for both channels at bit error rates below 10^{-2}. It was also verified that in a typical UWB multipath channel both convolutionally coded schemes outperform the non-coded pulse repetition scheme by more than 2 dB at BERs below 10^{-3} and by more than 3 dB at BERs below 10^{-4}. For AWGN channel the BER performance gain is larger for both convolutionally coded schemes.

It is worth noting that the maximum transmit power of a system employing RSLF binary convolutional encoders will be limited in terms of a continuous PSD even when the data input is generated by an unbalanced BMS. In contrast the maximum transmit power of a non-coded system or a system employing the best binary convolutional encoders known will be limited in terms of the maximum spectral line height when the data input is generated by an unbalanced BMS. Thus the later systems would need to decrease their transmit power or use an additional mechanism to reduce or eliminate the spectral lines. The consequence of implementing such solutions would be a loss in the BER performance for the first option or an increase in the system complexity for the second option. Thus it can be concluded that the use of RSLF binary convolutional encoders offer significant advantages in terms of both PSD characteristics and BER performance.

It is evident that the introduction of convolutional coding (RSLF or non-RSLF) in any UWB communication system will increase the system's complexity compared with the non-coded case. Nevertheless, the inclusion of error correction mechanisms has become standard in actual communication systems. For example both the IEEE 802.15.4a standard draft and the IEEE 802.15.3a study group (now disbanded), which use UWB systems in its physical layer, consider the use of convolutional codes, [40, 19]. Note how the use of the RSLF convolutional encoders introduced in this chapter does not increase the system's complexity compared with the use of equivalent traditional feedforward convolutional encoders with optimum IWS (OIWS – the best binary convolutional encoders known). Furthermore, when using the RSLF convolutional encoders the overall system design can be simplified as even the use of a relatively short PR-DS helps to

have a smoother spectrum when the BMS is highly biased. Finally it is worth mentioning that the complexity of the RSLF convolutional codes reported here can be classified to be low to medium as the total encoder memory ranges (a measure of decoding complexity, [100, 101, 104]) from 3 to 10 and good convolutional encoders have been reported with total encoder memory above 20, [103] (the industry de facto standard has total encoder memory equal to 6, [104]).



a) BER performance comparison b) PSD shape comparison

Figure 6-14. Side by side BER performance and PSD shape (analytical) comparisons between Q-BOPPM IR-based UWB systems employing: the rate 1/4, $v = 4$, RSLF convolutional encoder with primitive feedback polynomial (RSLF PP), the best rate 1/4, $v = 4$, convolutional encoder known with optimum IWS (Ffwd OIWS) and a rate equivalent non-coded Q-BOPPM IR-based UWB system. The input binary data stream is assumed to be generated by an unbalanced BMS with $p_{y,01} = 4/5$ and $p_{y,01} = 3/5$. The signal parameters are $T_s = 20$ ns, $T_r = 10$ ns and $T_\beta = 0.5$ ns. The 4th derivative Gaussian pulse is used with duration $T_w \approx 0.4$ ns. No TH or DS is considered.

Chapter 7

Conclusions

This chapter concludes this research by summarising the contributions and findings presented throughout this thesis. In addition, areas of possible future research are presented at the end of this chapter.

7.1 Summary of Main Results

This thesis has studied the effects on the transmitted signal's power spectral density (PSD) caused by the introduction of convolutional coding in impulse radio (IR) based ultra wideband (UWB) systems. It has been shown that properly designed convolutional encoders can achieve both improved PSD characteristics and improved bit error rates (BER) when used on IR-based UWB systems.

The literature review of previous work relevant for this thesis was provided in Chapter 1. In this chapter it was determined that the spectral analysis of general convolutionally coded IR-based UWB signals had not been previously addressed. As well, it was noted that only pseudo-chaotic time hopping (PCTH) and related schemes addressed the issue of simultaneous BER improvement and spectral line suppression for the particular case of pulse position modulated (PPM) time hopping (TH) IR-based UWB signals. In these schemes all the TH is driven by the encoder.

In Chapter 2 some useful concepts, terminology and definitions extensively used throughout this thesis were introduced. Two main areas were dealt with in this chapter. The first area described the UWB model used and further reviewed previous work on the spectral analysis and spectral shaping of IR-based UWB signals. The second area provided a brief review of the application of convolutional coding in IR-based UWB systems and introduced several important concepts related to convolutional coding in general.

The spectral analysis of convolutionally coded/Markov-driven IR-based UWB signals was carried out in Chapter 3. The analysis started by deriving a source-encoder (SE) pair Markov model (of Moore type) which describes the statistics of the convolutional encoder when driven by a binary Markov source. Particularly, a first order binary Markov source (BMS) was used as it enables the modelling of unbalanced (that is non-uniform distributed) independent identically distributed (i.i.d.) memoryless binary data streams as well as binary data streams with memory often found in practical systems. Once the SE pair model was introduced, a signal model that covers a wide variety of IR-based modulation schemes such as PPM, pulse amplitude modulation (PAM), binary phase shift keying (BPSK), biorthogonal PPM (BOPPM), PAM/PPM and pulse shape modulation

(PSM), combined with periodic or random TH and/or direct sequence (DS) multiplication was introduced. In addition the signal model accounts for the inclusion of attenuation and/or timing jitter. By performing the spectral analysis of the signal model through a non-cyclostationary based approach, novel closed form expressions for the power spectral density (PSD) of convolutionally coded/Markov-driven IR-based UWB signals were obtained. Application examples of the derived formulas were presented for several cases of interest. With these examples it was proved that the introduction of convolutional coding may have positive effects on the PSD shape of IR-based UWB signals. Nevertheless, it was shown as well that in order to achieve such positive effects care should be taken when using a convolutional encoder for spectral shaping purposes. This is due to even a simple change in the generators order may have a significant impact in the PSD shape of IR-based UWB signals.

Once it was determined that the introduction of convolutional coding may have positive effects over the PSD's shape, the search for convolutional encoders that can achieve both improved PSD characteristics and improved bit error rates (BER) when used in IR-based UWB systems started in Chapter 4. First the PCTH scheme was classified as a binary to M-ary convolutionally coded system using orthogonal M-ary PPM signalling. Then the condition needed for a binary to M-ary convolutional encoder to eliminate as many spectral lines as the PCTH scheme was identified through analysis. Making use of this condition (referenced as the spectral line suppression condition) new M-ary maximum free distance (MFD) feedforward spectral line suppressive (FSLS) binary to M-ary convolutional encoders were found through a computer based code search procedure. The new encoders were designed to eliminate as many spectral lines as the PCTH scheme under the same operation conditions, that is, independent identically distributed (i.i.d.) binary data streams with uniform distribution are assumed at the encoder's input. The spectral line suppression capabilities of the new encoders were further demonstrated through several examples of PSD plots. Moreover, it was shown that the new FSLS binary to M-ary convolutionally coded scheme can outperform the PCTH and non-coded schemes in terms of BER performance for both hard and soft Viterbi decoding.

As discussed in the introduction of Chapter 5, not all the data sources or compression algorithms found in practical systems generate uniform distributed i.i.d. binary streams. Thus the goal in this chapter was to find binary to M-ary convolutional encoders that satisfy the spectral line suppressive condition even when the binary data stream at the encoder input does not consist on uniformly distributed i.i.d. binary symbols. In particular it was assumed that the data stream was generated by an unbalanced BMS. This source enables the modelling of unbalanced i.i.d. memoryless binary data streams as well as binary data streams with memory often found in practical systems. First it was proved that for an unbalanced BMS the spectral line suppression condition may not be satisfied when using encoders with traditional feedforward structures such as those used by the PCTH and the FSLS binary to M-ary convolutional encoders introduced in Chapter 4. Afterwards it was demonstrated that properly designed binary to M-ary convolutional

encoders with recursive structures enable the fulfilment of the spectral line suppression condition even when the data stream is generated by an unbalanced first order BMS. Using a modified version of the code search procedure introduced in Chapter 4, novel M-ary MFD recursive spectral line suppressive (RSLS) binary to M-ary convolutional encoders for unbalanced BMS's were found. These encoders are able to eliminate as many spectral lines as PCTH and the FSLS binary to M-ary convolutional encoders introduced in Chapter 4 even when the data stream at the encoder input consists of binary symbols generated by an unbalanced BMS. Through PSD plots examples it was further shown that the RSLS binary to M-ary convolutional encoders offer superior PSD characteristics when the data stream is generated by an unbalanced BMS compared to the PCTH scheme, the FSLS binary to M-ary convolutional encoders and binary PPM with ideal perfectly random TH. Lastly it was shown that there may exist a minimal BER performance loss when using the RSLS binary to M-ary convolutional encoders instead of the FSLS binary to M-ary convolutional encoders. Nevertheless it is important to highlight that this possible BER performance loss is compensated by the superior PSD characteristics achieved with the RSLS binary to M-ary convolutional encoders. As well it is important to note that both convolutionally coded systems outperform the non-coded binary PPM system in terms BER performance by a significant amount.

In Chapter 6 the novel idea of using binary convolutional encoders for spectral line elimination purposes in BPSK and quaternary BOPPM (Q-BOPPM) IR-based UWB systems was proposed. The goal in this chapter was to design and find binary convolutional encoders that generate signals whose PSD is spectral line-free even when the binary data stream at the encoder input is generated by an unbalanced first order BMS. The condition needed for a binary convolutional encoder to generate such a spectral line-free signal was obtained through analysis. In a similar way to Chapter 5, it was proven that the traditional feedforward structures which are used by the best binary convolutional encoders known may not produce a spectral line-free PSD when the data input is generated by an unbalanced BMS. In contrast it was shown that for this kind of sources recursive structures are best suited for the design of spectral line-free binary convolutional encoders. Thus novel MFD recursive spectral line-free (RSLF) binary convolutional encoders were found by using a new code search procedure. These encoders provide a spectral line-free PSD when used in BPSK/Q-BOPPM (natural and Gray) IR-based UWB systems even when the input consists on a binary data stream generated by an unbalanced BMS. Through several PSD plots examples it was demonstrated that for unbalanced BMS's the RSLF binary convolutional encoders generate Q-BOPPM IR-based UWB signals with a spectral line-free PSD. In contrast, it was shown that for Q-BOPPM IR-based UWB systems using the best binary convolutional encoders known or a rate equivalent non-coded pulse repetition scheme (with/without pseudo-random pulse polarity randomisation), the signal's PSD shows a significant amount of spectral lines when the data stream is generated by an unbalanced BMS. Lastly it was shown that there may exist some sligth BER performance loss when using the RSLF binary convolutional encoders instead of the best binary

convolutional encoders known. However, when the data stream does not consist of uniformly distributed i.i.d. binary symbols, the possible BER performance loss is compensated by the spectral line-free characteristic offered by the RSLF binary convolutional encoders. Note as well that both convolutionally coded systems outperform the rate equivalent non-coded pulse repetition scheme in terms of BER performance by a significant amount. It is worth highlighting that the maximum transmit power of a system employing RSLF binary convolutional encoders will be limited in terms of a continuous PSD even when the data input is generated by an unbalanced BMS. In contrast the maximum transmit power of a non-coded pulse repetition based system or a system employing the best binary convolutional encoders known will be limited in terms of the maximum spectral line height when the data input is generated by such a source. Therefore the later systems would need to decrease their transmit power or use an additional mechanism to reduce or eliminate the spectral lines. The consequence of implementing such solutions would be a loss on BER performance for the first case or an increase in the system complexity for the second case. Thus it can be concluded that the use of RSLF binary convolutional encoders offer significant advantages in terms of both PSD characteristics and BER performance. This is particularly important for UWB communications as they are PSD limited

Based on the previous discussion it can be asserted that the use of FSLS binary to M-ary convolutional encoders, RSLS binary to M-ary convolutional encoders and RSLF binary convolutional encoders in IR-based UWB systems offer significant advantages in terms of both PSD characteristics and BER performance compared to IR-based UWB systems employing no coding or standard convolutional coding. Thus it can be concluded that the objectives of this thesis introduced in Chapter 1 have been reached.

7.2 Conclusions

In this thesis it has been demonstrated that the introduction of convolutional coding in impulse radio (IR) based ultra wideband (UWB) systems can have significant effects over the power spectral density (PSD) shape of the transmitted UWB signal. Therefore by properly designing the convolutional encoder two key issues arising in IR-based UWB communications can be addressed simultaneously, namely, bit error rate (BER) performance improvement and PSD shaping. In particular this thesis dealt with the design of convolutional coding schemes for spectral line elimination (suppression) in IR-based UWB systems, which is a topic of major importance in the design of compliant UWB systems.

Traditionally, feedforward structures have been preferred over recursive structures when designing stand alone convolutional encoders, [83, 100-103]. This is due to in general feedforward implementations offer better BER performance. Nevertheless, the feedforward structures are not best suited for the design of convolutional encoders with good spectral shaping capabilities. This is especially true when the data stream at the input of the system is generated by an unbalanced binary Markov source (BMS). Instead, in this thesis it has been proved that by using recursive

structures the design of convolutional encoders that eliminate significant numbers of spectral lines in pulse position modulated (PPM) time hopping (TH) IR-based UWB systems is feasible. Furthermore, for IR-based UWB systems employing binary phase shift keying (BPSK) or biorthogonal PPM (BOPPM) the design of recursive convolutional encoders that generate a spectral line-free PSD is feasible.

Here new recursive non-systematic convolutional encoders with convenient PSD characteristics for IR-based UWB systems have been introduced. These encoders effectively shape the signal's PSD when the data stream at the input of the encoder is generated by a balanced/unbalanced BMS. Therefore, these encoders offer significant advantages in terms of PSD characteristics compared with traditional feedforward convolutional encoders. However, it is worth noting that a BER performance degradation of up to 0.5 dB can be observed when using the new spectral shaping convolutional encoders introduced in this work instead of the best convolutional encoders known. Nevertheless, as UWB emissions are currently limited in terms of spectral masks instead of bandwidth occupancy, the possible performance degradation is compensated by the superior PSD characteristics offered by the new spectral shaping convolutional encoders.

As previously mentioned, the introduction of convolutional coding will increase the system's complexity compared to the non-coded case. However, the inclusion of error correction mechanisms has become standard in actual communication systems. The advantage offered by the spectral shaping encoders introduced in this thesis is that they conveniently shape the spectrum and provide significant coding gains over non-coded systems. Furthermore, note that a system using traditional convolutional encoders or no convolutional coding will need extra mechanisms to effectively shape the PSD and achieve compliance with the current regulations. Therefore, in terms of implementation, an IR-based UWB system employing the spectral shaping encoders reported in this thesis may be less complex than a system employing traditional convolutional encoders. On the other side, a non-coded system may be simpler to implement but it will show a significant BER performance degradation compared to the convolutionally coded system (for the all examples presented in this thesis that performance degradation is of more than 2 dB at BERs below 10^{-3} and more than 3 dB at BERs below 10^{-4}).

It is worth mentioning that for the convolutional encoders reported in this thesis the BER performance and PSD characteristics are improved upon using encoders with larger memory. However, the number of states in the decoder increases as the encoder memory grows and therefore the overall system's complexity is increased as well. Nevertheless, as the industry de facto standard has total encoder memory equal to 6, it can be said that a good compromise between complexity, BER performance and PSD characteristics is obtained by using the spectral shaping convolutional encoders introduced in Chapters 5 and 6 with total encoder memories equal to 6 and 7. Furthermore, from all the encoders introduced in these two chapters the ones using primitive feedback polynomials must be preferred as they provide better PSD characteristics. *Therefore, it can be concluded that the convolutional encoders reported in Tables 5-2, 6-2 and 6-4 with total*

encoder memory, v, equal to 6 and 7 offer a good compromise between complexity, BER performance and PSD characteristics.

Finally, it is worth commenting that the main limitation in the use of the spectral shaping convolutional encoders introduced in this thesis resides in the kind of data source feeding the system. For instance, the feedforward spectral line suppressive (FSLS) convolutional encoders introduced in Chapter 4 require the data stream to strictly consist on independent identically distributed (i.i.d.) binary symbols with uniform distribution. Note this assumption limits the applicability range of these encoders. On the other side, the recursive spectral line suppressive (RSLS) and recursive spectral line-free (RSLF) convolutional encoders introduced in Chapters 5 and 6 were designed by assuming that the data stream is generated by an unbalanced/balanced BMS. Although this assumption covers a wide variety of data sources, there still may exist data sources with a completely different set of statistics. For these cases the encoders reported on Chapters 5 and 6 may not show the expected PSD characteristics, in which case different convolutional encoders designs may be needed to be explored.

7.3 Original Contributions of the Thesis

The contributions of the thesis can be outlined as follows:

- Derivation of a source-encoder (SE) pair model that enables one to describe the interaction of the BMS and the convolutional encoder with a single Markov model which can be classified as being of the Moore type (within the definition adopted for finite state sequential machines).
- An original spectral analysis for convolutionally coded IR-based UWB systems. The model used for the analysis and the closed form PSD expressions obtained from it cover different modulation schemes such as pulse amplitude modulation (PAM), binary phase shift keying (BPSK), pulse position modulation (PPM), biorthogonal PPM (BOPPM), PAM/PPM, and pulse shape modulation (PSM), combined with pseudo-random (periodic)/random time hopping (TH) and/or pseudo-random/random direct sequence (DS). In addition the signal model and PSD expressions account for the inclusion of attenuation and/or timing jitter.
- The identification of convolutional encoder's characteristics that have a positive or negative effect in the transmitted signal's PSD.
- The classification of pseudo-chaotic time hopping (PCTH) and related schemes as feedforward binary to *M*-ary convolutionally coded PPM IR-based UWB systems.
- The identification of the variables that enable the design of feedforward binary to *M*-ary convolutional encoders with similar PSD characteristics to those achieved with the PCTH scheme under the same operation constraints (that is i.i.d. binary data streams with uniform distribution are assumed at the encoder input).
- A code search procedure to seek feedforward binary to *M*-binary convolutional encoders with PSD characteristics similar to those achieved with PCTH and improved BER

performance.

- A set of new *M*-ary maximum free distance (MFD) feedforward binary to *M*-ary convolutional encoders that have the same spectral line suppressive characteristics as the PCTH scheme (under similar operation conditions) and provide improved BER performance over the PCTH scheme.
- The identification of the variables and conditions that enable the design of binary to *M*-ary convolutional encoders with similar PSD characteristics to those achieved with the PCTH scheme even when the encoder input does not consist of an i.i.d. binary data stream with uniform distribution. In particular the data stream is assumed to be generated by a generic first order BMS.
- The modification of the code search procedure such that spectral line suppressive binary to *M*-ary convolutional encoders for unbalanced (that is, non-uniform distributed) BMS's can be found.
- Two sets of novel *M*-ary MFD spectral line suppressive binary to *M*-ary convolutional encoders which eliminate as many spectral lines as the PCTH scheme but keep its spectral line suppressive characteristics even when the binary stream at the encoder input is generated by an unbalanced BMS (that is, the data stream does not consists on uniform distributed i.i.d. binary symbols).
- The novel proposal of using binary convolutional encoders for spectral line elimination purposes in BPSK and quaternary BOPPM (Q-BOPPM) IR-based UWB systems.
- The identification of the variables and conditions that enable the design of binary convolutional encoders which generate signals with spectral line-free PSDs when used in BPSK/Q-BOPPM IR-based UWB systems with unbalanced BMS inputs.
- A code search procedure aimed to seek spectral line-free binary convolutional encoders with good BER performance.
- The introduction of nine sets of novel MFD binary convolutional encoders which generate a spectral line-free signal even when the binary stream at the encoder input is generated by an unbalanced BMS (that is, the data stream does not consists on uniform distributed i.i.d. binary symbols). Encoders with rates from 1/2 up to 1/32 and memory orders from 3 up to 9 are provided.

7.4 Areas of Future Research

This thesis has studied the use of convolutional coding in IR-based UWB systems to simultaneously achieve both improved PSD characteristics and improved BER performance. In that sense the following areas have been identified to continue the research carried out in this thesis

- *Bit error rate performance in multiple user scenarios*. The first extension of this work would be to evaluate the BER performance achieved with the convolutional encoders introduced in this dissertation when two or more IR-based UWB systems are transmitting

simultaneously. Although it has been shown that the introduction of convolutional coding helps to improve the BER performance and the number of supported users in multiple user scenarios, [21, 24, 25, 32], some variations may be worth researching. In particular it would be interesting to evaluate if the improved PSD characteristics of the convolutional encoders reported in this thesis help to reduce the multiple access (MA) interference in multiuser IR-based UWB systems. Variations include the presence of users with unbalanced data inputs and the use of short PR-TH and/or PR-DS sequences for MA purposes.

- *Generation of spectral notches in the PSD.* Another area worth exploring would be the generation of spectral notches in the PSD for narrowband interference suppression through the convolutional coding operation. This issue has been previously addressed by means of pulse shaping, [68, 69], and TH sequence design [72, 74, 75]. Thus further research can be done in the feasibility of finding convolutional encoders that provide both spectral notches in the PSD and improved bit error rates.
- *Analysis and design of turbo codes with convenient PSD characteristics.* An important feature of the convolutional encoders introduced Chapters 5 and 6 is that they were constructed based on recursive structures. Commonly, turbo codes are constructed using systematic recursive convolutional codes, [85], while little research has been done in the use of non-systematic recursive structures in the construction of good turbo codes, [155]. Thus it would be interesting to design turbo codes based in non-recursive convolutional encoders which provide convenient PSD characteristics.
- *Performance in multipath channels.* Although some results were presented in Chapter 6 – Section 6.7.2 for the IEEE 802.15.3a UWB channel model 1 (CM1), more research is needed about the performance of the convolutional encoders introduced in this thesis in multipath channels. In particular it would be interesting to evaluate if the improved PSD characteristics help to reduce the effects of interpulse and interframe interference.
- *Encoders matched to specific antennas.* It would be worth to explore the feasibility of designing convolutional encoders whose signal's PSD matches the antenna's frequency response.
- *Impact of interleavers.* When a system is expected to work in multipath and fading channels the BER performance can be sometimes improved upon the introduction of interleaving after channel coding (providing some time delay can be tolerated), [83]. Therefore, it is would be worth to analyse the PSD characteristics and BER performance of the convolutional encoders introduced in this thesis when interleaving is introduced after the convolutional coding operation.
- *Analysis and design of low density parity check (LDPC) codes with convenient PSD characteristics.*

References

[1] L. K. Brunson, J. P. Camacho, W. M. Doolan, R. L. Hinkle, G. F. Hurt, M. J. Murray, F. A. Najmy, P. C. Roosa, and R. L. Sole, *Assessment of Compatibility Between Ultrawideband Devices and Selected Federal Systems*, NTIA Special Publication 01-43, U.S. Department of Commerce, Jan. 2001. http://www.ntia.doc.gov/osmhome/reports/uwb/uwb.pdf

[2] Federal Communications Commission, *First Report and Order, revision of part 15 of the Commission's rules regarding ultra-wideband transmission systems*, ET Docket 98-153, FCC 02-48, 2002. http://ftp.fcc.gov/oet/dockets/et98-153

[3] Manson Communications Ltd, *Impact of UWB on third-generation telecommunications (3G) Final Report*, Office of Communications U.K. (OFCOM), Feb. 2003. http://www.ofcom.org.uk/static/archive/ra/topics/research/topics/uwb/3gfinalreport.zip

[4] CEPT/ECC, *ECC Decision of 24 March 2006 on the harmonised conditions for devices using Ultra-Wideband (UWB) technology in bands below 10.6 GHz*, CEPT/ECC- ECC/DEC/(06)04, March 2006, http://www.ero.dk/documentation/docs/doc98/official/Word/ECCDEC0604.DOC

[5] CEPT/ECC, *ECC Decision of 1 December 2006 on the harmonised conditions for devices using Ultra-Wideband (UWB) technology with Low Duty Cycle (LDC) in the frequency band 3.4 - 4.8 GHz*, CEPT/ECC-ECC/DEC/(06)12, Dec. 2006. http://www.ero.dk/documentation/docs/doc98/official/Word/ECCDEC0612.DOC

[6] R. A. Scholtz, "Multiple access with time-hopping impulse modulation," *at IEEE Military Communications Conference 1993, MILCOM 1993,* pp. 447 - 450, Oct. 1993.

[7] M. Z. Win and R. A. Scholtz, "Impulse radio: how it works," *IEEE Communications Letters*, vol. 2, pp. 36-38, Feb. 1998.

[8] M. Z. Win and R. A. Scholtz, "Ultra-wide bandwidth time-hopping spread-spectrum impulse radio for wireless multiple-access communications," *IEEE Transactions on Communications*, vol. 48, pp. 679-689, April 2000.

[9] P. Siwiak, "Ultra-wide band radio: introducing a new technology," *at IEEE Vehicular Technology Conference 2001. VTC 2001-Spring*, pp. 1169 - 1172, May 2001.

[10] G. R. Aiello and G. D. Rogerson, "Ultra-wideband wireless systems," *IEEE Microwave Magazine*, vol. 4, pp. 36 - 47, June 2003.

[11] B. Allen, M. Ghavami, A. Armogida, and H. Aghvami, "The holy grail of wire replacement?," *IEE Communications Engineer*, vol. 1, pp. 14 - 17, Oct. 2003.

[12] S. Roy, J. R. Foerster, V. S. Somayazulu, and D. G. Leeper, "Ultrawideband radio design: the promise of high-speed, short-range wireless connectivity," *Proceedings of the IEEE*, vol. 92, pp. 295 - 311, Feb. 2004.

[13] M. G. Di Benedetto and G. Giancola, *Understanding Ultra Wide Band Radio Fundamentals*, 1 ed. Prentice Hall, 2004.

[14] M. Ghavami, L. B. Michael, and R. Kohno, *Ultra wideband signals and systems in communication*

engineering. Wiley, 2004.

[15] I. Oppermann, M. Hamalainen, and J. Linatti, *UWB Theory and Applications*. John Wiley & Sons, 2004.

[16] R. Aiello and A. Batra, *Ultra Wideband Systems: Technologies and Applications*. Newnes, 2006.

[17] H. Arslan, Z. N. Chen, and M. G. Di Benedetto, *Ultra Wideband Wireless Communication*. Wiley - Interscience, Oct. 2006.

[18] X. Shen, M. Guizani, R. C. Qiu, and T. Le-Ngoc, *Ultra-Wideband Wireless Communications and Networks*. John Wiley &Sons, April 2006.

[19] M. Ghavami, L. B. Michael, and R. Kohno, *Ultra Wideband Signals and Systems in Communications Engineering*, 2 ed. John Wiley & Sons, 2007.

[20] F. Ramirez-Mireles, "Performance of ultra-wideband SSMA using time hopping and M-ary PPM," *IEEE Journal on Selected Areas in Communications*, vol. 19, pp. 1186-1196, June 2001.

[21] A. R. Forouzan, M. Nasiri-Kenari, and J. A. Slehi, "Performance analysis of time-hopping spread-spectrum multiple-access systems: uncoded and coded schemes," *IEEE Transactions on Wireless Communications*, vol. 1, pp. 671-681, Oct. 2002.

[22] A. R. Forouzan and M. Abtahi, "Application of convolutional error correcting codes in ultrawideband M-ary PPM signaling," *IEEE Microwave and Wireless Components Letters*, vol. 13, pp. 308-310, Aug. 2003.

[23] N. Yamamoto and T. Ohtsuki, "Adaptive internally turbo-coded ultra wideband-impulse radio (AITC-UWB-IR) system," *at IEEE International Conference on Communications 2003, ICC 2003*, pp. 3535 - 3539, May 2003.

[24] Z. Bai, W. Liu, S. Xu, W. Zhang, and K. Kwak, "DS-BPAM UWB system-coded and uncoded scheme," *at IEEE International Symposium on Communications and Information Technology 2005. ISCIT 2005*, pp. 689 - 692, Oct. 2005.

[25] M. Nasiri-Kenari and M. G. Shayesteh, "Performance analysis and comparison of different multirate TH-UWB systems: uncoded and coded schemes," *IEE Proceedings in Communications*, vol. 152, pp. 833 - 844, Dec. 2005.

[26] U. Riaz, C.-C. J. Kuo, and M.-O. Pun, "Ultra-wide band impulse radio (UWB-IR) with super-orthogonal turbo codes (SOTC)," *at 39th Asilomar Signals, Systems and Computers Conference* 2005, pp. 981 - 985, Nov. 2005.

[27] M. M. Pietrzyk and J. H. Weber, "Performance of UWB-IR with polarity randomization and interleaved coding-modulation on multipath fading channels," *at IEEE Vehicular Technology Conference 2005, VTC 2005-Spring*, pp. 1365 - 1369, May 2005.

[28] F. Chiti, R. Fantacci, D. Marabissi, L. Innocenti, H. H. Chen, and M. Guizani, "Punctured hopping CDMA techniques: fundamentals and application to UWB communications," *IEEE Journal on Selected Areas in Communications*, vol. 24, pp. 731 - 737, April 2006.

[29] K. Li, X. Wang, G. Yue, and L. Ping, "A low-rate code-spread and chip-interleaved time-hopping UWB system," *IEEE Journal on Selected Areas in Communications*, vol. 24, pp. 864 - 870, April 2006.

[30] M. M. Pietrzyk, K. Popovsky, T. A. Wysocki, B. J. Wysocki, and J. H. Weber, "Scarcely populated UWB-IR systems with interleaved coding-modulation on multipath fading channels," *at IEEE International Conference on Ultra-Wideband 2006, (ICUWB2006)*, pp. 55 - 60, Sept. 2006.

[31] L. Reggiani and G. M. Maggio, "Coherent vs. non-coherent detection for orthogonal convolutional modulation: a trade-off analysis," *at IEEE International conference on ultra-wideband 2006*, pp. 43 - 48, Sept. 2006.

[32] Z. Zeinalpour-Yazdi and M. Nasiri-Kenari, "Performance analysis and comparisons of different ultra-wideband multiple access modulation schemes," *IEE Proceedings in Communications*, vol. 153, pp. 705 - 718, Oct. 2006.

[33] D. Laney, G. M. Maggio, F. Lehmann, and L. Larson, "BER performance and spectral properties of interleaved convolutional time hopping for UWB impulse radio," *at IEEE Global Telecommunications Conference 2003, GLOBECOM 2003*, pp. 1994 - 1998, Dec. 2003.

[34] K. Takizawa and R. Kohno, "Combined iterative demapping and decoding for coded UWB-IR systems," *at IEEE Conference on Ultra Wideband Systems and Technologies 2003*, pp. 423 - 427, Nov. 2003.

[35] T. Matsumoto, H. Ochiai, and R. Kohno, "Super-orthogonal convolutional coding with orthogonal pulse waveform for ultra wideband communications," *at Conference on Ultrawideband Systems and Technologies and International Workshop on Ultra Wideband Systems 2004*, pp. 202-206, May 2004.

[36] K. Takizawa and R. Kohno, "Combined iterative demapping and decoding for coded M-ary BOK DS-UWB systems," *at Conference on Ultrawideband Systems and Technologies 2004, Joint UWBNST & IWUWBS*, pp. 207 - 211, May 2004.

[37] Y. P. Hong and H. Y. Song, "Line spectrum analysis of impulse radio UWB systems using a pulse position modulation," *at IEEE International Conference on Communications 2005, ICC 2005*, pp. 2877 - 2880, May 2005.

[38] R. Merz and J.-Y. L. Boudee, "Effect of interfering users on the modulation order and code rate for UWB impulse-radio bit-interleaved coded M-ary PPM," *at IEEE International Conference on Broadband Networks 2005, BROADNETS 2005*, pp. 1156 - 1162, Oct. 2005.

[39] Y. Tian and C. Yang, "Coded modulation in UWB communications," *at IEEE International Symposium on Personal, Indoor and Mobile Radio Communications 2005, PIMRC 2005.*, pp. 491 - 495, Sept. 2005.

[40] P. Martigne, "UWB for low data rate applications: technology overview and regulatory aspects," *at IEEE International Symposium on Circuits and Systems 2006, ISCAS 2006*, pp. 2425 - 2428, May 2006.

[41] M. Z. Win, "On the power spectral density of digital pulse streams generated by M-ary cyclostationary sequences in the presence of stationary timing jitter," *IEEE Transactions on Communications*, vol. 46, pp. 1135-1145, Sept. 1998.

[42] W. A. Kissick, *The Temporal and Spectral Characteristics of Ultrawideband Signals*, NTIA Report 01-383, U.S. Department of Commerce, Jan. 2001.
http://www.its.bldrdoc.gov/pub/ntia-rpt/01-383/

[43] J. Romme and L. Piazzo, "On the power spectral density of time-hopping impulse radio," *at IEEE Conference on Ultra Wideband Systems and Technologies 2002*, pp. 241-244, May 2002.

[44] M. Z. Win, "A unified spectral analysis of generalized time-hopping spread-spectrum signals in the presence of timing jitter," *IEEE Journal on Selected Areas in Communications*, vol. 20, pp. 1613-1627, Dec. 2002.

[45] N. H. Lehmann and A. M. Haimovich, "The power spectral density of a time hopping UWB signal: a

survey," *at IEEE Conference on Ultra Wideband Systems and Technologies 2003*, pp. 234-239, Nov. 2003.

[46] P. Bremaud and A. Ridolfi, "Power spectra related to UWB communications," *at IEEE International Conference on Communications 2004, ICC 2004*, pp. 292 - 294 June 2004.

[47] A. Cassioli and R. Mazzenga, "Spectral analysis of UWB multiple access schemes using random scrambling," *IEEE Transactions on Wireless Communications*, vol. 3, pp. 1637-1647, Sept. 2004.

[48] S. Erkucuk and D. I. Kim, "Power spectral density characteristics of MCSK based impulse radios in UWB communications," *at Vehicular Technology Conference 2005, VTC 2005-Spring*, pp. 1391 - 1395, May 2005.

[49] A. Ridolfi and M. Z. Win, "UWB signals as shot noises," *at IEEE International Conference on Ultra-Wideband 2005*, pp. 690-694, Sept. 2005.

[50] A. Ridolfi and M. Z. Win, "Ultrawide bandwidth signals as shot noise: a unifying approach," *IEEE Journal on Selected Areas in Communications*, vol. 24, pp. 899 - 905, Apr. 2006.

[51] H. Zhang and T. A. Gulliver, "Biorthogonal pulse position modulation for time-hopping multiple access UWB communications," *IEEE Transactions on Wireless Communications*, vol. 4, pp. 1154-1162, May 2005.

[52] G. L. Cariolaro and G. P. Tronca, "Spectra of block coded digital signals," *IEEE Transactions on Communications*, vol. 22, pp. 1555-1974, Oct. 1974.

[53] J. H. Gilchrist and J. B. Thomas, "Power spectral densities of modulated error-correcting coded sequences," *IEEE Transactions on Communications*, vol. 23, pp. 1207 - 1214, Nov. 1975.

[54] E. Biglieri and G. Caire, "Power spectrum of block-coded modulation," *IEEE Transactions on Communications*, vol. 42, pp. 1580-1585, Feb/March/April 1994.

[55] J. Justesen, "Information rates and power spectra of digital codes," *IEEE Transactions on Information Theory*, vol. 28, pp. 457 -472, May 1982.

[56] J. Justesen, "Calculation of power spectra for block coded signals," *IEEE Transactions on Communications*, vol. 49, pp. 389-392, March 2001.

[57] E. Biglieri, "Ungerboeck codes do not shape the signal power spectrum," *IEEE Transactions on Information Theory*, vol. 32, pp. 595-596, July 1986.

[58] E. Biglieri, D. Divsalar, P. J. McLane, and M. K. Simon, *Introduction to Trellis - Coded Modulation with Applications*. Macmillan, 1991.

[59] D. Divsalar and M. K. Simon, "Spectral characteristics of convolutionally coded digital signals," *IEEE Transactions on Communications*, vol. 28, pp. 173-185, Feb. 1980.

[60] J. B. Bezerra and D. S. Arantes, "Spectral analysis of Q-ary digital signals encoded by P-ary convolutional codes," *IEEE Transactions on Information Theory*, vol. 32, pp. 313-319, Mar. 1986.

[61] H. Sheng, P. Orlik, A. M. Haimovich, L. J. Cimini, and J. Zhang, "On the spectral and power requirements for ultra-wideband transmission," *at IEEE International Conference on Communications 2003, ICC 2003*, pp. 738 - 742, May 2003.

[62] B. Hu and N. C. Beaulieu, "Pulse shaping in UWB communication systems," *at IEEE Vehicular Technology Conference 2005, VTC 2004-Fall*, pp. 5175 - 5179, Sept. 2004.

[63] B. Parr, B. L. Cho, K. Wallace, and Z. Ding, "A novel ultra-wideband pulse design algorithm," *IEEE Communications Letters*, vol. 7, pp. 219 - 221, May 2003.

[64] R. S. Dilmaghani, M. Ghavami, B. Allen, and H. Aghvami, "Novel UWB pulse shaping using prolate

spheroidal wave functions," *at IEEE Personal, Indoor and Mobile Radio Communications 2003, PIMRC 2003*, pp. 602 - 606, Sept. 2003.

[65] B. Allen, S. A. Ghorashi, and M. Ghavami, "A review of pulse design for impulse radio," *at IEE Seminar on Ultra Wideband Communications Technologies and System Design 2004*, pp. 93 - 97, July 2004.

[66] Y. Kim, B. Jang, C. Shin, and B. F. Womack, "Orthonormal pulses for high data rate communications in indoor UWB systems," *IEEE Communications Letters*, vol. 9, pp. 405-407, May 2005.

[67] X. Wu, Z. Tian, T. N. Davidson, and G. B. Giannakis, "Optimal waveform design for UWB radios," *IEEE Transactions on Signal Processing*, vol. 54, pp. 2009 - 2021, June 2006.

[68] Z. Luo, H. Gao, Y. Liu, and J. Gao, "A new UWB pulse design method for narrowband interference suppression," *at IEEE Global Telecommunications Conference 2004, GLOBECOM 2004*, pp. 388 - 3492, Dec. 2004.

[69] Y. Wang, X. Dong, and I. J. Fair, "A method for spectrum shaping and NBI suppression in UWB communications," *at IEEE International Conference on Communications 2006, ICC 2006*, pp. 1476 - 1481, June 2006.

[70] G. M. Maggio, N. Rulkov, and L. Reggiani, "Pseudo-chaotic time hopping for UWB impulse radio," *IEEE Transactions on Circuits and Systems Part I*, vol. 48, pp. 1424-1434, Dec. 2001.

[71] C. Muller, S. Zeisberg, H. Seidel, and A. Finger, "Spreading properties of time hopping codes in ultra wideband systems," *at IEEE International Symposium on Spread Spectrum Techniques and Applications 2002, ISSSTA 2002*, pp. 64 - 67, 2002.

[72] L. Piazzo and J. Romme, "Spectrum control by means of the TH code in UWB systems," *at IEEE Vehicular Technology Conference 2003, VTC 2003-Spring*, pp. 1649 - 1653, Apr. 2003.

[73] W. D. Wu, C. H. Wang, and C. C. Chao, "Effects of hopping codes in TH-SS UWB signals," *at Vehicular Technology Conference 2003, VTC 2003-Fall*, pp. 1308 - 1312, Oct. 2003.

[74] J. Bellorado, S. S. Ghassenzadeh, A. Kavcic, B. Tarokh, and V. Tarokh, "Time-hopping sequence design for narrowband interference suppression," *at IEEE Vehicular Technology Conference 2004, VTC 2004-Fall*, pp. 3925 - 3929, Sept. 2004.

[75] J. I. Jamp and L. E. Larson, "A coding technique for spectral shaping ultra-wideband pulse position modulated signals," *at IEEE Vehicular Technology Conference 2004, VTC 2004-Fall*, pp. 1188 - 1191, Sept. 2004.

[76] D. Domenicali, G. Giancola, and M. G. di Benedetto, "Fluid coding and coexistence in ultra wide band networks," *Mobile Networks and Applications*, vol. 11, pp. 501 - 508, Aug. 2006.

[77] S. Erkucuk and D. I. Kim, "Effects of code design on the spectral characteristics of MCSK based impulse radios," *at IEEE International Conference on Communications 2006, ICC 2006*, pp. 475 - 4762, June 2006.

[78] N. Lehmann and A. M. Haimovich, "New approach to control the power spectral density of a time hopping UWB signal," *at Conference on Information Sciences and Systems 2003*, Mar. 2003.

[79] S. S. Mo, A. D. Gelman, and J. Gopal, "Frame synchronization in UWB using multiple SYNC words to eliminate line frequencies," *at IEEE Wireless Communications and Networking Conference 2003, WCNC 2003*, pp. 773 - 778, March 2003.

[80] Y. P. Nakache and A. F. Molisch, "Spectral shape of UWB signals - influence of modulation format, multiple access scheme and pulse shape," *at IEEE Vehicular Technology Conference 2003 - Spring*,

pp. 2510-2514, April 2003.

[81] Y. P. Nakache and A. F. Molisch, "Spectral shaping of UWB signals for time-hopping impulse radio," *IEEE Journal on Selected Areas in Communications*, vol. 24, pp. 738 - 744, Apr. 2006.

[82] A. J. Viterbi, *CDMA: Principles of Spread Spectrum Communication*. Addison-Wesley, 1995.

[83] J. G. Proakis, *Digital Communications*, 3 ed. McGraw Hill, 1995.

[84] J. Hagenauer, "Rate-compatible punctured convolutional codes (PCPC codes) and their performance," *IEEE Transactions on Communications*, vol. 36, pp. 389 - 400, Apr. 1988.

[85] C. B. Schlegel and L. C. Perez, *Trellis and Turbo Coding*. IEEE/Wiley-Interscience, 2004.

[86] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*, 1 ed. John Wiley & Sons, 1965.

[87] M. K. Simon, S. M. Hinedi, and W. C. Lindsey, *Digital Communication Techniques: Signal Design and Detection*. Prentice-Hall, 1995.

[88] P. J. Lee, "Computation of the bit error rate of coherent M-ary PSK with Gray code bit Mapping," *IEEE Transactions on Communications*, vol. 34, pp. 488 - 491, May 1986.

[89] N. Boubaker and K. B. Letaief, "Ultra wideband DSSS for multiple access communications using antipodal signaling," *at IEEE International Conference on Communications 2003, ICC 03*, pp. 2197 - 2201, May 2003.

[90] J. R. Foerster, "The performance of a direct-sequence spread ultrawideband system in the presence of multipath, narrowband interference, and multiuser interference," *IEEE Conference on Ultra Wideband Systems and Technologies 2002*, pp. 87 -91, May 2002.

[91] A. J. Viterbi, "Very low rate convolutional codes for maximum theoretical performance of spread-spectrum multiple-access channels," *IEEE Journal on Selected Areas in Communications*, vol. 8, pp. 641 - 649, May 1990.

[92] S. Gezici, H. Kobayashi, H. V. Poor, and A. F. Molisch, "Performance evaluation of impulse radio UWB systems with pulse-based polarity randomization," *IEEE Transactions on Signal Processing*, vol. 53, pp. 2537 - 2549, July 2005.

[93] A. F. Molisch, Y. G. Li, Y. P. Nakache, P. Orlik, M. Miyake, Y. Wu, S. Gezici, H. Sheng, S. Y. Kung, H. Kobayashi, H. V. Poor, A. M. Haimovich, and J. Zhang, "A low-cost time-hopping impulse radio system for high data rate transmission," *EURASIP Journal on Applied Signal Processing*, vol. 2005, pp. 397 - 412, Oct. 2005.

[94] S. Haykin, *Communication Systems*, 2 ed. John Wiley & Sons, 1983.

[95] M. K. Simon, J. K. Omura, R. A. Scholtz, and B. K. Levitt, *Spread Spectrum Communications Handbook*. McGraw-Hill, 2002.

[96] I. Ramachndran and S. Roy, "Acquisition of direct-sequence ultra-wideband signals," *at IEEE Wireless Communications and Networking Conference 2005*, pp. 752 - 757, Mar. 2005.

[97] S. R. Aedudodla, S. Vijayakumaran, and T. F. Wong, "Ultra-wideband signal acquisition with hybrid DS-TH spreading," *IEEE Transactions on Wireless Communications*, vol. 5, pp. 2504 - 2515, Sept. 2006.

[98] J. Ibrahim and R. M. Buehrer, "Two-stage acquisition for UWB in dense multipath," *IEEE Journal on Selected Areas in Communications*, vol. 24, pp. 801 - 807, Apr. 2006.

[99] P. Galko and S. Pasupathy, "The mean power spectral density of Markov chain driven signals," *IEEE Transactions on Information Theory*, vol. 27, pp. 746-754, Nov. 1981.

[100] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*. Prentice-Hall, 1983.

[101] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*, 1 ed. Prentice-Hall, 1995.

[102] L. H. C. Lee, *Convolutional Coding: Fundamentals and Applications*, 1 ed. Artech House, 1997.

[103] R. Johannesson and K. S. Zigangirov, *Fundamentals of Convolutional Coding*. IEEE Press, 1999.

[104] D. J. Costello, J. Hagenauer, H. Imai, and S. B. Wicker, "Applications of error-control coding," *IEEE Transactions on Information Theory*, vol. 44, pp. 2531 - 2560, Oct. 1998.

[105] W. E. Ryan and S. G. Wilson, "Two classes of convolutional codes over GF(q) for q-ary orthogonal signaling," *IEEE Transactions on Communications*, vol. 39, pp. 30-40, Jan. 1991.

[106] S. L. Miller, "Design and analysis of trellis codes for orthogonal signal sets," *IEEE Transactions on Communications*, vol. 43, pp. 821-827, Feb-Mar-Apr 1995.

[107] J. J. Chang, D. J. Hwang, and M. C. Lin, "Some extended results on the search for good convolutional codes," *IEEE Transactions on Information Theory*, vol. 43, pp. 1682-1697, Sept. 1997.

[108] M. Cedervall and R. Johannesson, "A fast algorithm for computing distance spectrum of convolutional codes," *IEEE Transactions on Information Theory*, vol. 35, pp. 1146-1159, Nov. 1989.

[109] I. E. Bocharova and B. D. Kudryashov, "Rational rate punctured convolutional codes for soft-decision Viterbi decoding," *IEEE Transactions on Information Theory*, vol. 43, pp. 1305-1313, July 1997.

[110] P. Frenger, P. Orten, T. Ottoson, and A. V. Svensson, "zRate-compatible convolutional codes for multirate DS-CDMA systems," *IEEE Transactions on Communications*, vol. 47, pp. 828 - 836, June 1999.

[111] P. Frenger, P. Orten, and T. Ottoson, "Convolutional codes with optimum distance spectrum," *IEEE Communications Letters*, vol. 3, pp. 317 - 319, Nov. 1999.

[112] P. Frenger, P. Orten, and T. Ottoson, "Code-spread CDMA using maximum free distance low rate convolutional codes," *IEEE Transactions on Communications*, vol. 48, pp. 135 - 144, Jan. 2000.

[113] H. Zhang and T. A. Gulliver, "Pulse position amplitude modulation for time-hopping multiple-access UWB communications," *IEEE Transactions on Communications*, vol. 53, pp. 1269-1273, Aug. 2005.

[114] G. Bilardi, R. Padovani, and G. L. Pierobon, "Spectral analysis of functions of Markov chains with applications," *IEEE Transactions on Communications*, vol. 31, pp. 853-861, July 1983.

[115] S. Benedetto, E. Biglieri, and V. Castellani, *Digital Transmission Theory*. Prentice-Hall, 1987.

[116] N. Abramson, *Information Theory and Coding*. McGraw-Hill, 1963.

[117] F. I. Alajaji, N. C. Phamdo, N. Farvardin, and T. E. Fuja, "Detection of binary markov sources over channels with additive markov noise," *IEEE Transactions on Information Theory*, vol. 42, pp. 230 - 239, Jan. 1996.

[118] J. M. Kroll and N. Phamdo, "Source-channel optimized trellis codes for bitonal image transmission over AWGN channels," *IEEE Transactions on Image Processing*, vol. 8, pp. 899 -912, July 1999.

[119] J. G. Kemeny and J. L. Snell, *Finite Markov Chains*. Van Nostrand, 1960.

[120] V. I. Romanovsky, *Discrete Markov Chains*. Wolters-Noordhoff, 1970.

[121] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering*, 2 ed. Addison-Wesley, 1994.

[122] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, 2002.

[123] T. L. Booth, *Sequential Machines and Automata Theory*. Wiley, 1967.

[124] A. Papoulis, *Probability, Random Variables and Stochastic Processes*. McGraw Hill, 1965.

[125] I. Guvenc and H. Arslan, "Performance evaluation of UWB systems in the presence of timing jitter," *at IEEE Conference on Ultra Wideband Systems and Technologies*, pp. 136 - 141, Nov. 2003

[126] A. J. Viterbi, "Orthogonal tree codes for communication in the presence of white Gaussian noise," *IEEE Transactions on Communications*, vol. 15, pp. 238 - 242, Apr. 1967.

[127] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*. McGraw-Hill, 1979.

[128] F. L. Chang, "Comparison of two convolutional orthogonal coding techniques for CDMA radio communications systems," *IEEE Transactions on Communications*, vol. 43, pp. 2028 - 2037, June 1995.

[129] R. F. Ormondroyd and J. J. Maxey, "Performance of low-rate orthogonal convolutional codes in DS-CDMA applications," *IEEE Transactions on Vehicular Technology*, vol. 46, pp. 320 - 328, May 1997.

[130] G. E. Atkin and H. P. Corrales, "Orthogonal convolutional coding for the PPM optical channel," *IEEE Journal of Lightwave Technology*, vol. 7, pp. 731 - 1007, Apr. 1989.

[131] E. Forestieri, R. Gangopadhyay, and G. Prati, "Performance of convolutional codes in a direct-detection optical PPM channel," *IEEE Transactions on Communications*, vol. 37, pp. 1303 - 1317, Dec. 1989.

[132] H. Park, "Convolutional coded pulse-position modulation on wireless infrared link," *IEE Proceedings in Optoelectronics*, vol. 148, pp. 199 - 203, Aug. 2001.

[133] H. Park, "Convolutional coded pulse-position modulation on wireless optical communication," *at IEEE Vehicular Technology Conference 2003. VTC 2003-Spring*, pp. 1293 - 1297, Apr. 2003.

[134] W. Feller, *An Introduction to Probability Theory and Its Applications*, 3rd ed. John Wiley & Sons, Inc., 1968.

[135] D. G. Daut, J. W. Modestino, and L. D. Wismer, "New short constraint length convolutional code constructions for selected rational rates," *IEEE Transactions on Information Theory*, vol. 28, pp. 794 - 1982, Sept. 1982.

[136] G. Wade, *Coding Techniques: An Introduction to Compression and Error Control*. Palgrave, 2000.

[137] F. I. Alajaji, N. C. Phamdo, and T. E. Fuja, "Channel codes that exploit the residual redundancy in CELP-encoded speech," *IEEE Transactions on Speech and Audio Processing*, vol. 4, pp. 325 - 336, Sep. 1996.

[138] T. Fingscheidt, T. Hindelang, R. V. Cox, and N. Sechadri, "Joint source-channel (de-)coding for mobile communications," *IEEE Transactions on Communications*, vol. 50, pp. 200 - 212, Feb. 2002.

[139] T. Fazel and T. Fuja, "Robust transmission of MELP-compressed speech: an illustrative example of joint source-channel decoding," *IEEE Transactions on Communications*, vol. 51, pp. 973 - 982, June 2003.

[140] H. Nguyen and P. Duhamel, "Estimation of redundancy in compressed image and video data for joint source-channel decoding," *at Global Telecommunications Conference 2003, GLOBECOM 2003*, pp. 2198 - 2202, Dec. 2003.

[141] M. Jeanne, J. C. Carlach, and P. Siohan, "Joint source-channel decoding of variable-length codes for convolutional codes and turbo codes," *IEEE Transactions on Communications*, vol. 53, pp. 10 - 15, Jan. 2005.

[142] M. Braff and D. L. Schilling, "The correlation of the output of a binary convolutional encoder," *IEEE Transactions on Communications*, vol. 28, pp. 371 - 375, 1980.

[143] I. J. Fair, V. K. Bhargava, and Q. Wang, "On the power spectral density of self-synchronizing scrambled sequences," *IEEE Transactions on Information Theory*, vol. 44, pp. 1687 - 1693, July 1998.

[144] R. H. Deng, Y. X. Li, and M. A. Herro, "DC-free error-correcting convolutional codes," *IEE Electronic Letters*, vol. 29, pp. 1910 - 1911, Oct. 1993.

[145] S. I. Lee, J. G. Kim, and E. K. Joo, "Class of DC-free error correcting codes based on convolutional codes," *IEE Electronic Letters*, vol. 30, pp. 1588 - 1590, Sept. 1994.

[146] S. I. Lee, "DC- and Nyquist-free error correcting convolutional codes," *IEE Electronic Letters*, vol. 32, pp. 2196 - 2197, Nov. 1996.

[147] M. Nasiri-Kenari and C. K. Rushforth, "A class of DC-free subcodes of convolutional codes," *IEEE Transactions on Communications*, vol. 44, pp. 1389 - 1391, Nov. 1996.

[148] M. C. Chiu, "DC-free error-correcting codes based on convolutional codes," *IEEE Transactions on Communications*, vol. 49, pp. 609 - 619, April 2001.

[149] T. Wadayama and A. J. Han-Vinck, "DC-free binary convolutional coding," *IEEE Transactions on Information Theory*, vol. 48, pp. 162 - 173, Jan. 2002.

[150] F. Zhai, Y. Xin, and I. J. Fair, "Dc-free convolutional codes and dc-free turbo codes," *at IEEE International Conference on Communications 2005, ICC 2005*, pp. 632 -636, May 2005.

[151] B. Bing, *High-Speed Wireless ATM and LANs*. Artech House, 2000.

[152] J. R. Foerster, M. Pendergrass, and A. F. Molisch, "A channel model for ultrawideband indoor communication," *at International Symposium on Wireless Personal Multimedia Communication*, pp. 116 - 120, Oct. 2003.

[153] A. F. Molisch, J. R. Foerster, and M. Pendergrass, "Channel models for ultrawideband personal area networks," *IEEE Wireless Communications*, vol. 10, pp. 1536-1284, Dec. 2003.

[154] X. Chu and R. Murch, "Performance analysis of DS-MA ultra-wideband communications incorporating channel-induced pulse overlap," *IEEE Transactions on wireless communications*, vol. 5, pp. 948 - 959, Apr. 2006.

[155] A. Banerjee, F. Vatta, B. Scanavino, and D. J. Costello, "Nonsystematic turbo codes," *IEEE Transactions on Communications*, vol. 53, pp. 1841 - 1849, Nov. 2005.

[156] M. Weisenhorn and W. Hirt, "Impact of the FCC Average and Peak Power Constraints on the Power of UWB Radio Signals," Integrated Project PULSERS Contract No: 506897, Deliverable No: D3b4b (Annex 1), Sept. 2004.

[157] N. C. Geckinli and d. Yavuz, *Discrete Fourier Transformation and Its Applications to Power Spectral Estimation*. Elsevier, 1983.

[158] S. L. Marple, *Digital Spectral Analysis: with Applications*. Prentice-Hall, 1987.

[159] D. E. Newland, *A Introduction to Random Vibrations, Spectral and Wavelet Analysis*, 3 ed. Pearson Education Limited, 1993.

[160] J. G. Proakis, *Digital Signal Processing*, 3 ed. Prentice Hall, 1996.

[161] A. Papoulis, *Signal Analysis*. McGraw Hill, 1977.

[162] M. Hamalainen, V. Honiven, R. Tesi, J. Iinatti, and M. Latva-aho, "On the UWB system coexistence with GSM900, UMTS/WCDMA, and GPS," *IEEE Journal on Selected Areas in Communications*, vol. 20, pp. 1712-1721, Dec 2002.

[163] W. Q. Malik, D. J. Edwards and C. J. Stevens, "Angular-spectral antenna effects in ultra wideband communications links," *IEE Proceedings on Communications*, Vol. 153, No. 1, pp. 99-105, Feb. 2006.

Appendix A

The Use of Simulation-Periodogram-DFT Methods for the Spectral Analysis of IR-Based UWB Signals

A.1 Introduction

This appendix deals with the application of simulation-periodogram-discrete Fourier transform (DFT) methods for the spectral analysis of ultra wideband (UWB) signals. The methodology used throughout this thesis for the validation of analytical (theoretical) power spectral density (PSD) expressions for impulse radio (IR) based UWB signals is explained here. Furthermore, comparisons of simulation-periodogram-DFT based PSD estimates with the FCC's (U.S. Federal Communications Commission) spectral mask for indoor UWB systems, [2], are provided at the end of the appendix.

Typically, the PSD properties of IR-based UWB signals have been studied by means of theoretical analysis [13, 18, 41-50] and/or simulation [18, 42, 43, 45, 80, 81, 156]. Although [43, 45, 80, 81] presented simulation results based on simulation-periodogram-DFT methods, only [43] directly compares the analytical PSD with the simulated PSD. Nevertheless, the authors in [43] adjust the continuous part of the analytical PSD to enable the comparison with the simulated PSD. Comparisons between analytical and simulated results have also been reported in [18, 42] where the simulated results were obtained by simulating the stages of a swept spectrum analyzer.

The comparison of analytical and simulated results is important since a misinterpretation of simulated results could lead to erroneous conclusions about the actual levels of a signal's PSD. Please note that the objective of this appendix is not to provide novel results regarding simulation-periodogram-DFT based PSD estimation, but to define a methodology which enables the use known digital spectral analysis results (presented in [157-160]) for the validation of the theoretical PSD results presented in this thesis. Shortly, it will be shown that when using simulation-periodogram-DFT based spectral analysis, the characteristics and effects of the data window used in the simulation should be considered for an adequate comparison between simulated PSD estimates and analytical PSD results for IR-based UWB signals. This is particularly true when both the analytical and simulated results are plotted together for inspection.

After introducing the simulation-periodogram-DFT based procedure, comparisons of simulated PSD estimates with the FCC's spectral mask for indoor UWB systems, [2], are provided. The objective of this comparison is to further show the spectral shaping capabilities of the convolutional encoders introduced in Chapters 5 and 6. Before introducing the comparison plots, the expected behaviour of actual power spectrum measurements of IR-based UWB signals as reported in [1, 2, 42] is reviewed. Based on this review the simulation-periodogram-FFT based procedure that closest resembles the expected behaviour of actual power spectrum measurements is defined.

A.2 Signal Models and Corresponding Analytical Power Spectral Densities of Typical Non-Coded IR-Based UWB Signals

Three kinds of signals models are used in next section's introduction of the simulation-periodogram-DFT based PSD estimation procedure. The first one is the non-modulated periodic pulse train with period T_{Per} defined in Section 2.4 as

$$x_{Per}(t) = \sum_{k=-\infty}^{\infty} w(t - kT_{Per}) \tag{A.1}$$

where $w(t)$ is the particular waveform used. Recall as well from Section 2.4 that this signal has PSD given by

$$S_{x_{Per}}(f) = \frac{1}{T_{Per}^{\ 2}} \sum_{n=-\infty}^{\infty} |W(\tfrac{n}{T_{Per}})|^2 \, \delta(f - \tfrac{n}{T_{Per}}) \tag{A.2}$$

where $W(f)$ is the Fourier transform of $w(t)$. Note how $S_{x_{Per}}(f)$ consists exclusively on spectral lines appearing at multiples of the pulse repetition frequency, PRF = $1/T_{Per}$.

The second signal used in Section A.3 is the non-coded (uncoded) PPM TH-IR-based UWB signal introduced in Section 4.3.3 defined by

$$x_{PPM}(t) = \sum_{l=-\infty}^{\infty} \sum_{k=0}^{N_w-1} w(t - lT_s - kT_r - y_l T_y - c_{lN_w+k} T_c) \tag{A.3}$$

where $y_l \in \{0,1\}$ is the l^{th} data bit; N_w is the number of transmitted pulses per bit; T_y is the binary PPM modulation index; $\{c_{lN_w+k}\}$ is a pseudo-random (PR) time hopping (TH) sequence taking values on the set $\{0, 1, 2, ..., (M/2)-1\}$ with period χ_c; T_c is the nominal shift caused by the TH sequence; T_r is the mean repetition time between pulses and $T_s = N_w T_r$ is the bit time.

In this and the following section it will be assumed that y_l is an independent identically distributed (i.i.d.) binary random sequence with uniform distribution. As well it will be assumed that the TH sequence is perfectly random (that is i.i.d. with uniform distribution) and thus c_{lN_w+k} is uniformly distributed over the set $\{0, 1, 2, ..., (M/2)-1\}$. Under these assumptions the PSD of the

signal defined by Equation (A.3) will show continuous (noise like) and discrete (spectral lines) components and is given by (see Section 4.3.3)

$$\begin{aligned} \overline{S_{x_{PPM}}}(f) &= S_{x_{PPM}.C.1}(f) + S_{x_{PPM}.D}(f) \qquad N_w = 1 \\ \overline{S_{x_{PPM}}}(f) &= S_{x_{PPM}.C.1}(f) + S_{x_{PPM}.C.2}(f) + S_{x_{PPM}.D}(f) \qquad N_w > 1 \end{aligned} \tag{A.4}$$

$$S_{x_{PPM}.C.1}(f) = \frac{1}{T_r}|W(f)|^2 - \frac{|W(f)|^2}{T_s}\left|\frac{1}{2}\sum_{\varsigma=0}^{1} e^{j2\pi\varsigma f T_y}\right|^2 \left|\frac{1}{(M/2)}\sum_{l=0}^{(M/2)-1} e^{-j2\pi l f T_c}\right|^2 \left|\sum_{k=0}^{N_w-1} e^{j2\pi k f T_r}\right|^2 \tag{A.5}$$

$$S_{x_{PPM}.C.2}(f) = \frac{2|W(f)|^2}{T_s}\left|\frac{1}{(M/2)}\sum_{l=0}^{(M/2)-1} e^{-j2\pi f l T_c}\right|^2 \sum_{k=0}^{N_w-2}\sum_{k'=k+1}^{N_w-1} \mathrm{Re}\{e^{-j2\pi f(k'-k)T_r}\} \tag{A.6}$$

$$S_{x_{PPM}.D}(f) = \frac{|W(f)|^2}{(T_s)^2}\sum_{r=-\infty}^{\infty}\left|\frac{1}{2}\sum_{\varsigma=0}^{1} e^{j2\pi(r/T_s)\varsigma T_y}\right|^2 \left|\frac{1}{(M/2)}\sum_{l=0}^{(M/2)-1} e^{-j2\pi(r/T_s)lT_c}\right|^2 \left|\sum_{k=0}^{N_w-1} e^{j2\pi(r/T_s)kT_r}\right|^2 \delta(f-\tfrac{r}{T_s}) \tag{A.7}$$

where $S_{x_{PPM}.C}(f)$ is the PSD's continuous part and $S_{x_{PPM}.D}(f)$ is the PSD's discrete part.

The third signal considered is the non-coded (uncoded) BPSK TH-IR-based UWB signal defined by

$$x_{BPSK}(t) = \sum_{l=-\infty}^{\infty}\sum_{k=0}^{N_w-1} \alpha_l w(t - lT_s - kT_r - c_{lN_w+k}T_c) \tag{A.8}$$

where $\alpha_l \in \{-1,1\}$ is the BPSK data stream, c_{lN_p+k} is a pseudo-random (PR) time hopping (TH) sequence taking values on the set $\{0, 1, 2, ..., (M/2)-1\}$ with period χ_c, and the other variables are as in Equation (A.3). In this and the following section it will be assumed that α_l is an uniform distributed i.i.d. sequence, that is $p_{\alpha,+1} = \Pr\{\alpha_l = +1\} = 1/2$ and $p_{\alpha,-1} = \Pr\{\alpha_l = -1\} = 1/2$. Thus using the formulas introduced in Sections 2.4 and 3.4 the PSD of $x_{BPSK}(t)$ can be found to be

$$\overline{S_{x_{BPSK}}}(f) = \frac{1}{T_r}|W(f)|^2 + \frac{2|W(f)|^2}{T_s\Lambda_{x_{BPSK}}}\sum_{k=0}^{N_w-2}\sum_{k'=k+1}^{N_w-1}\sum_{n=0}^{\Lambda_{x_{BPSK}}-1} \mathrm{Re}\{e^{-j2\pi f(k'-k)T_r} e^{-j2\pi f T_c(c_{nN_w+k'}-c_{nN_w+k})}\} \tag{A.9}$$

where $\Lambda_{x_{BPSK}} = \mathrm{lcm}(\chi_c, N_w)/N_w$. Note how the PSD defined by Equation (A.9) does not have any discrete (spectral line) component.

A.3 A Simulation-Periodogram-DFT Based Spectral Estimation Methodology for the Validation of Analytical PSD Expressions of IR-Based UWB Signals

There exist several implementations of periodogram based spectral analysis such as the Bartlett, the Welch and the Blackman-Tukey methods, [158, 160]. However, their common aim is to reduce the variance of the periodogram based spectral estimate when only a single data record is available, [158, 160]. This is not a problem with simulations in which several data records of the same length

can be generated. Thus the variance reduction problem in this thesis is addressed by averaging non-overlapping periodograms (Bartlett method), [157, 158, 160], and this topic will not be further addressed in this appendix.

A.3.1 The Periodogram Based PSD Estimator

The periodogram is an estimator of a random process' PSD from a finite segment of a single realisation of such process, [157, 161]. In terms of a rectangular "data window" of length T, the periodogram is defined as

$$P_w(f) = \frac{1}{T}\left|\int_{-T/2}^{T/2} \boldsymbol{x}(t)e^{-j2\pi ft}dt\right|^2 \tag{A.10}$$

where $\boldsymbol{x}(t)$ is the random process whose PSD will be estimated. Similarly, in terms of a generic data window the periodogram is given by

$$P_w(f) = \left|\int_{-\infty}^{\infty} \boldsymbol{x}(t)x_w(t)e^{-j2\pi ft}dt\right|^2 \tag{A.11}$$

where $x_w(t)$ is a real function such that $x_w(t) = 0$ for $-\infty < t \le 0$ and $T \le t < \infty$. It turns out that $x_w(t)$ is usually restricted to have unit energy, [157-161], that is

$$\int_{-\infty}^{\infty} x_w{}^2(t)dt = \int_{-\infty}^{\infty} |X_w(f)|^2 df = 1 \tag{A.12}$$

where $X_w(f)$ is the Fourier transform of $x_w(t)$.

Note that the time domain multiplication of the functions $\boldsymbol{x}(t)$ and $x_w(t)$ implies convolution in the frequency domain. Thus, it can be shown that the expected value of the estimator, $\mathrm{E}\{P_w(f)\}$, is related to the true spectrum, $\overline{S_x}(f)$, of the random process by ([157, 161])

$$\mathrm{E}\{P_w(f)\} = \overline{P_w}(f) = \overline{S_x}(f) * |X_w(f)|^2 \tag{A.13}$$

From Equation (A.13) can be seen that the periodogram is a biased estimator of the actual signal spectrum, the bias being given by,

$$B_w(f) = \overline{S_x}(f) * |X_w(f)|^2 - \overline{S_x}(f) \tag{A.14}$$

Note how the bias is reduced as $|X_w(f)|^2$ approaches an impulse function with unit energy, which would require the use a data window with infinite length, [157, 159, 161].

A.3.2 Issues Arising in the Application of Periodogram Based PSD Estimation Methods for Periodic and Noise Like Signals

Before introducing the simulation-periodogram-DFT based spectral estimation procedure used in this thesis, the behaviour of the periodogram estimator will be assessed for two extreme cases namely a noise like signal (whose PSD does not have any discrete component) and a periodic signal (whose PSD consists exclusively on spectral lines).

First assume $x(t)$ is white noise with two sided power spectral density given by

$$S_{x_{AWGN}}(f) = N_0/2 \tag{A.15}$$

When a data window fulfilling Equation (A.12) is used the expected value of the periodogram estimate is given by

$$\overline{P_w}(f) = S_{x_{AWGN}}(f) * |X_w(f)|^2 = \int_{-\infty}^{\infty} \frac{N_0}{2} |X_w(f-\lambda)|^2 \, d\lambda = N_0/2 \tag{A.16}$$

Thus it can be concluded that the periodogram estimator gives good results for signals with noise like behaviour.

Now consider the periodic signal defined by Equation (A.1) with respective PSD given by Equation (A.2). When using a data window fulfilling Equation (A.12) the expected value of the periodogram estimator yields

$$\overline{P_w}(f) = S_{x_{Per}}(f) * |X_w(f)|^2 = \frac{1}{T_{Per}^2} \sum_{n=-\infty}^{\infty} |W(\tfrac{n}{T_{Per}})|^2 \, \delta(f - \tfrac{n}{T_{Per}}) = \frac{1}{T_{Per}^2} \sum_{n=-\infty}^{\infty} |W(\tfrac{n}{T_{Per}})|^2 |X_w(f - \tfrac{n}{T_{Per}})|^2 \tag{A.17}$$

for this signal. Thus the spectral lines are replaced by shifted replicas of $|X_w(f)|^2$ located at frequencies multiples of $1/T_{Per}$.

This means that a periodogram based PSD plot may not show the actual power contained in a spectral line located at $f = k/T_{Per}$. Instead a periodogram plot will "display" the power given by

$$\overline{P_w}(\tfrac{k}{T_{Per}}) = \frac{1}{T_{Per}^2} |W(\tfrac{k}{T_{Per}})|^2 |X_w(0)|^2 + \frac{1}{T_{Per}^2} \sum_{\substack{n=-\infty \\ n \neq k}}^{\infty} |W(\tfrac{n}{T_{Per}})|^2 |X_w(\tfrac{k}{T_{Per}} - \tfrac{n}{T_{Per}})|^2 \tag{A.18}$$

at $f = k/T_{Per}$. Therefore for a spectral line located at $f = k/T_{Per}$ a periodogram based PSD plot will display the actual power of the spectral line multiplied by $|X_w(0)|^2$ plus an aliasing component given by the rightmost sum in Equation (A.18). Thus inspecting such a plot may not be a sensible way for the assessment of the spectral lines' power unless the effects of the data window are considered. This is particularly true for unit energy data windows (that is data windows fulfilling Equation (A.12)) which are not necessarily constrained in terms of the value acquired by $|X_w(0)|^2$.

As an example of this consider the unit energy rectangular data window defined by

$$x_{w,ue}(t) = \begin{cases} 1/\sqrt{T} & 0 \le t \le T \\ 0 & \text{elsewhere} \end{cases} \tag{A.19}$$

For this data window $|X_w(f)|^2$ can be found to be

$$|X_{w,ue}(f)|^2 = T\,\mathrm{sinc}^2(Tf) = T\left(\frac{\sin(\pi Tf)}{\pi Tf}\right)^2 \tag{A.20}$$

where sinc is the cardinal sine function. Note how for this data window $|X_{w,ue}(0)|^2 = T$. Therefore if $T \neq 1\,\mathrm{s}$ a periodogram based PSD plot will not display the actual power of the spectral lines as shown by Equation (A.18).

An better method for periodogram based PSD estimation of periodic signals would be to ignore the unit energy constraint and instead use a data windows that satisfy

$$|X_w(0)|^2 = 1 \tag{A.21}$$

Thus if Equation (A.21) is fulfilled, a periodogram based PSD plot will show a closer approximation to the actual power of discrete PSD components, being any difference caused by the rightmost sum on Equation (A.18). As an example a "$|X_w(0)|^2 = 1$" rectangular data window can be defined as

$$x_{w,nue}(t) = \begin{cases} 1/T & 0 \le t \le T \\ 0 & \text{elsewhere} \end{cases} \tag{A.22}$$

with

$$|X_{w,nue}(f)|^2 = \text{sinc}^2(Tf) \tag{A.23}$$

Note how the data window defined by Equation (A.22) and the unit energy rectangular data window defined by Equation (A.19) are equivalent for $T = 1$ s.

It is worth noting that although the use of a $|X_w(0)|^2 = 1$ data window allows us to obtain periodogram based PSD plots that show a good approximation to the actual power of periodic PSD components, a large error in the estimation of the continuous PSD level may be produced. As an example, when using the $|X_w(0)|^2 = 1$ rectangular data window defined by Equation (A.22) the expected value of the periodogram of white noise with PSD defined by Equation (A.15) becomes

$$\overline{P_w}(f) = S_{x_{AWGN}}(f) * |X_{w,nue}(f)|^2 = \int_{-\infty}^{\infty} \frac{N_0}{2} \text{sinc}^2(T\lambda) d\lambda = \frac{N_0}{2}\frac{1}{T} \tag{A.24}$$

which is a weighted version of the actual continuous PSD level.

A.3.3 Discrete Periodogram Formulas

Until this point the discussion has been presented in terms of continuous signals. However the previous results and observations also apply when the estimation is based on a discrete record $\boldsymbol{x}(nT_{samp})$ of the continuous signal $\boldsymbol{x}(t)$, where T_{samp} is the sampling period. In fact the discrete equivalent of the periodogram (as defined in Equation (A.11)) for a unit energy rectangular data window (Equation (A.19)) is defined by ([157-159])

$$\widehat{P_{w,ue}}\left[\frac{k}{NT_{samp}}\right] = \frac{T_{samp}}{N}\left|\sum_{n=0}^{N-1} \boldsymbol{x}(nT_{samp}) e^{-j2\pi nk/N}\right|^2; \quad k = 0,1,...,N \tag{A.25}$$

where N is number of samples taken from $\boldsymbol{x}(t)x_{w,ue}(t)$. Thus, Equation (A.25) gives the (possibly aliased) unit energy rectangular data window periodogram estimate, $P_{w,ue}(f)$, of the continuous time signal, $\boldsymbol{x}(t)$, at frequencies multiples of $1/NT_{samp}$. Note the relationship between the sampling period, T_{samp}, the number of samples, N, and the data window length in continuous time, T, is given by

$$T = T_{samp} N \tag{A.26}$$

By following [157] or [158] the discrete equivalent of the periodogram (as defined in Equation (A.11)) for a $|X_w(0)|^2=1$ rectangular data window (Equation (A.22)) can be found to be

$$\widehat{P_{w,nue}}\left[\frac{k}{NT_{samp}}\right]=\left|\frac{1}{N}\sum_{n=0}^{N-1}\boldsymbol{x}(nT_{samp})e^{-j2\pi nk/N}\right|^2 \tag{A.27}$$

which gives the (possibly aliased) $|X_w(0)|^2=1$ rectangular data window periodogram estimate, $P_{w,nue}(f)$, of the continuous time signal, $\boldsymbol{x}(t)$, at frequencies multiples of $1/NT_{samp}$.

A.3.4 The Simulation-Periodogram-DFT Based Spectral Estimation Procedure Used to Validate the Analytical PSD Results

The first issue arising when using simulation-periodogram-DFT based spectral analysis of IR-based UWB signals is that the PSD may show both continuous (noise like) and discrete (periodic like) components as evidenced by Equations (A.4) to (A.7). Thus, in order to simultaneously validate the continuous and discrete parts of analytical PSD expressions with a single periodogram based PSD estimate the data window used must satisfy both the unit energy and the $|X_w(0)|^2=1$ conditions.

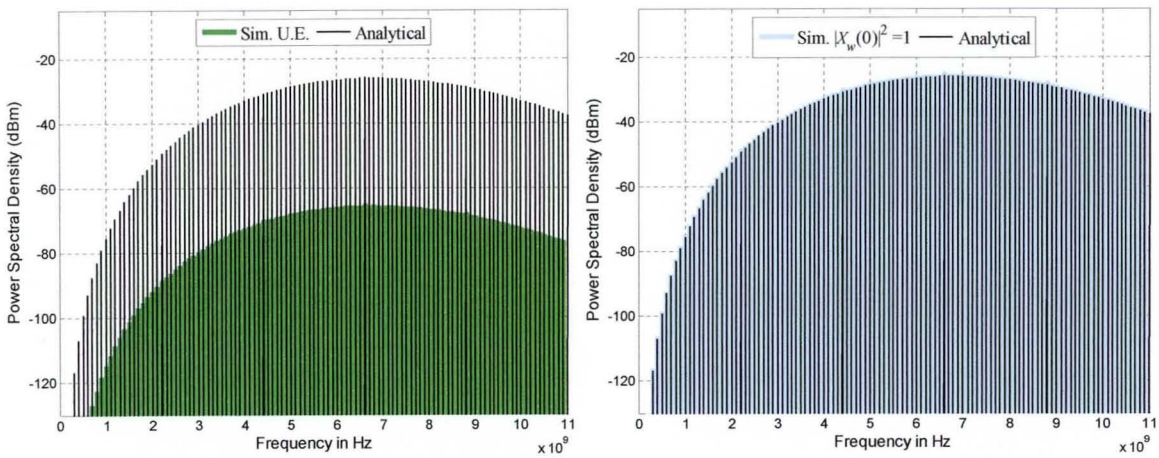
However, from the previous discussion it can be inferred that for signals with continuous and discrete PSD parts it may be advantageous to implement a double simulation-estimation procedure when the PSD estimation is periodogram based. Therefore the following methodology was used to validate the analytical PSD results presented in the previous chapters:

- For signals with discrete PSD and no continuous PSD the simulation-periodogram-DFT based PSD estimate was calculated using a $|X_w(0)|^2=1$ rectangular data window. These plots were correspondingly labelled as "Sim. $|X_w(0)|^2=1$" throughout the thesis.
- For signals with continuous PSD and no discrete PSD the simulation-periodogram-DFT based PSD estimate was calculated using a unit energy rectangular data window. These plots were correspondingly labelled as "Sim. U.E.".
- For signals with both continuous and discrete PSD parts two simulation-periodogram-DFT based PSD estimates were obtained, one using a $|X_w(0)|^2=1$ rectangular data window (labelled as "Sim. $|X_w(0)|^2=1$") and the other using a unit energy rectangular data window (labelled as "Sim. U.E."). Usually both estimates and the analytical result were plotted altogether in the same figure adequately labelled.

All the simulations were implemented in MATLAB®. Parameters such as the simulation's resolution and the data window length were adjusted in accordance with the computational resources available. Although MATLAB® provides predefined spectral estimation functions such as the periodogram, these were not used. Instead a different routine which uses MATLAB®'s basic fast Fourier transform (FFT) function was programmed to calculate the discrete periodogram estimates introduced in Section A.3.3.

A.3.4.1 Examples of Simulation-Periodogram-DFT based PSD Estimates of Periodic Signals

In Figure A-1 the analytical PSD defined by Equation (A.2) is plotted with simulation-periodogram-DFT based PSD estimates obtained by using a unit energy rectangular data window (Figure A-1a) and a $|X_w(0)|^2=1$ rectangular data window (Figure A-1b). Note that the analytical PSD defined by Equation (A.2) consists of discrete components exclusively.



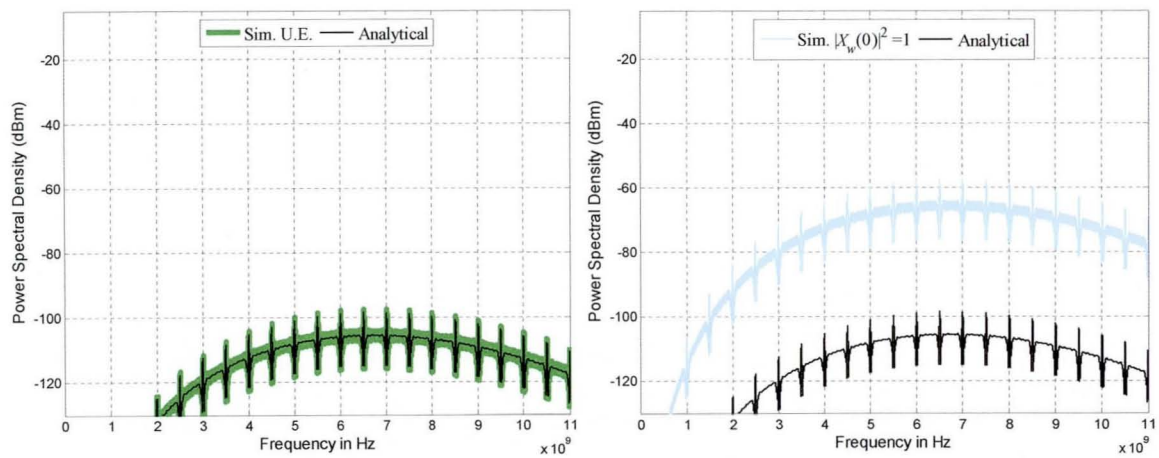
a) Unit energy rectangular data window b) $|X_w(0)|^2=1$ rectangular data window

Figure A-1. Analytical and simulated power spectrum of a periodic pulse train. The signal pulse repetition period is $T_{Per}=10$ ns. The 4th derivative Gaussian pulse was used with duration $T_w \approx 0.4$ ns..

It can be seen in Figure A-1 that for a PSD consisting of spectral lines exclusively, the use of $|X_w(0)|^2=1$ rectangular data windows provides a better PSD estimate.

A.3.4.2 Examples of Simulation-Periodogram-DFT based PSD Estimates of Signals with Continuous PSD and No Spectral Lines

In Figure A-2 the analytical PSD defined by Equation (A.9) is plotted with simulation-periodogram-DFT based PSD estimates obtained by using a unit energy rectangular data window (Figure A-2a) and a $|X_w(0)|^2=1$ rectangular data window (Figure A-2b).



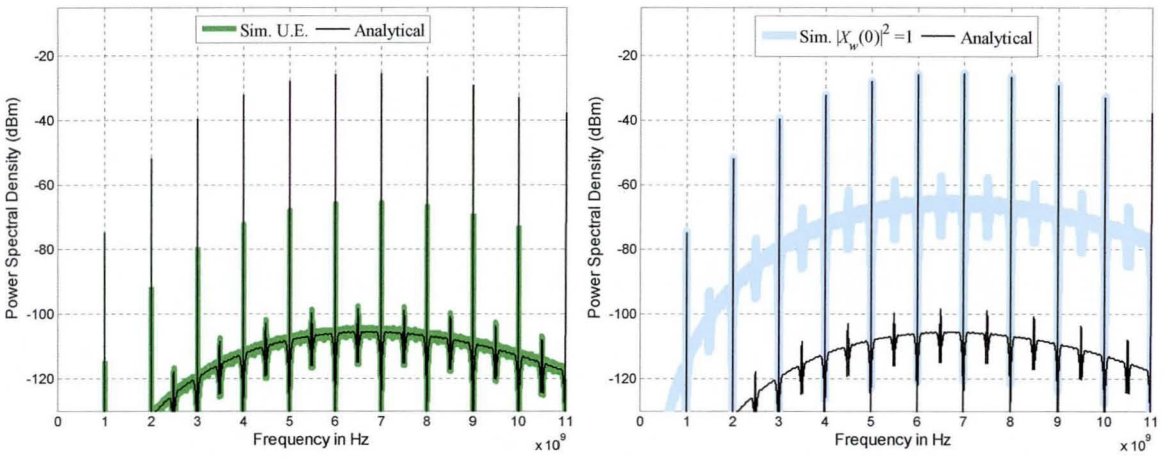
a) Unit energy rectangular data window b) $|X_w(0)|^2=1$ rectangular data window

Figure A-2. Analytical and simulated PSDs of the BPSK TH-IR-based UWB signal defined by Equation (A.8). The signal parameters are $N_w=5$, $T_s=50$ ns, $T_r=10$ ns, $T_c=2$ ns, $T_\beta=1$ ns and c_{lNw+k} uniform distributed over the set $\{0, 1, 2, 3, 4\}$. The 4th derivative Gaussian pulse is used with duration $T_w \approx 0.4$ ns.

It can be seen in Figure A-2 that for a PSD consisting on a continuous (noise like) part without any spectral line content the unit energy rectangular data window provides a better PSD estimate.

A.3.4.3 Examples of Simulation-Periodogram-DFT based PSD Estimates of Signals with both Continuous and Discrete PSD Components

In Figure A-3 the analytical PSD defined by Equations (A.4) to (A.7) is plotted with simulation-periodogram-DFT based PSD estimates obtained by using a unit energy rectangular data window (Figure A-3a) and a $|X_w(0)|^2=1$ rectangular data window (Figure A-3b). Note that the analytical PSD defined by these equations shows both continuous (noise like) and discrete (periodic like) components.



a) Unit energy rectangular data window

b) $|X_w(0)|^2=1$ rectangular data window

Figure A-3. Analytical and Simulated PSDs of the PPM TH-IR-based UWB signal defined by Equation (A.3). The signal parameters are $N_w = 5$, $T_s = 50$ ns, $T_r = 10$ ns, $T_c = 2$ ns, $T_\beta = 1$ ns and c_{lNw+k} uniform distributed over the set $\{0, 1, 2, 3, 4\}$. The 4th derivative Gaussian pulse was used with duration $T_w \approx 0.4$ ns.

It can be seen in Figure A-3 that for a PSD showing continuous and discrete components both data windows must be used to validate the analytical result as expected.

A.4 Comparisons of Simulation-Periodogram-DFT Based PSD Estimates of Convolutionally Coded Signals with the FCC's Spectral Mask for Indoor Applications

The simulation-periodogram-DFT based PSD plots presented in this section are intended to further show the spectral shaping capabilities of the recursive spectral line suppressive (RSLS – Chapter 5) and the recursive spectral line-free (RSLF – Chapter 6) convolutional encoders introduced in Chapters 5 and 6.

The FCC defines its average emission limits in terms of effective isotropic radiated power (EIRP) in dBm as measured with a 1 MHz resolution bandwidth (RBW), [2]. Thus, in order to obtain a simulation-periodogram-DFT based PSD estimate that can be compared to the FCC's

spectral mask, the length of the data window used in the simulation must be such that the simulation estimate has equivalent resolution bandwidth equal to 1 MHz. For rectangular data windows such as those defined by Equations (A.19) and (A.22), the data window's length should be set to $T = 1/\text{RBW} = 1/10^6 = 1$ μs in order to obtain simulation estimates with equivalent RWB equal to 1 MHz, [158, 159].

Once the rectangular data window length has been defined, it is necessary to decide which type of window should be used for the estimation: a unit energy rectangular data window or a $|X_w(0)|^2 = 1$ rectangular data window. In order to do this the next section reviews the expected behaviour of actual power spectrum measurements of IR-based UWB signals as reported in [1, 2, 42].

A.4.1 Review of Results Reported in [1, 2, 42] Concerning the Behaviour of PSD Measurements of UWB Signals

As explained in Section A.3.2 any periodogram based PSD estimate will show different behaviour for the continuous and discrete PSD components depending on the type of data window used (unit energy or $|X_w(0)|^2 = 1$). Thus defining which data window to use is an important issue if simulation-periodogram-DFT based PSD estimates are to be compared with the FCC's spectral masks.

The first point to note in the FCC regulation ([2] page 72), is that it clarifies that for periodic signals the measured spectral line power will be the same regardless of the RBW used for the measurement as far as the RBW of the measurement device contains only one spectral line at a time. As well, the FCC regulation clarifies that the measured power of noise like emissions will vary as

$$10\log_{10}\left(\frac{\text{RBW}_{new}}{\text{RBW}_{ref}}\right) \quad \text{(dB)} \tag{A.28}$$

where RBW_{ref} is the reference resolution bandwidth and RBW_{new} is the new resolution bandwidth. This means that for UWB signals with discrete and continuous PSD components, the measured power of the spectral lines should remain unchanged when varying the RBW (as far as only one spectral line is resolved within the specified RBW) while the measured power of the continuous PSD should vary as described by Equation (A.28).

The U. S. Department of Commerce – National Telecommunications and Information Administration (NTIA) has released two special publications [1, 42], which deal with the power spectrum measurement and interference assessment of UWB signals. In particular [1] introduces formulas that describe the behaviour of actual UWB power spectrum measurements as a function of the RBW of the measurement instrument (see Section 3 on [1]). These formulas are given referenced to the average power spectrum measured with a 1 MHz RBW.

If we assume a periodic IR-based UWB signal with $T_{Per} > 1$ μs (see Equation (A.1)), then the results in [1] and [42] tell us that the measured average power spectrum of this signal will consist of spectral lines whose height remains constant even when varying the RBW. This is so as far as

the RBW is less than $1/T_{Per}$. In contrast, when the IR-based UWB signal shows noise like behaviour the measured average power spectrum will change with any variation in the measurement's RBW as described by Equation (A.28).

A.4.2 Data Window Used in the Simulations

From the review presented in Section A.4.1, it can be concluded that in order to compare simulation-periodogram-DFT based PSD estimates with the FCC's spectral masks, the unit energy rectangular data window may not be the best option. In order to see this note from Equation (A.16) how for this window the periodogram based PSD estimate of noise like signals does not vary with a change on the data window length T (equivalent to change on the RWB). Moreover, when varying the data window length the estimated power of any spectral line will change as a function of T. Thus the behaviour of the simulation-periodogram-DFT based PSD estimate will not resemble the expected behaviour of actual power spectrum measurements as reported in [1, 2, 42] (see the previous section) when a unit energy data window is used.

Now consider the behaviour of simulation-periodogram-DFT based PSD estimates when the $|X_w(0)|^2=1$ rectangular data window is used. It can be concluded from Equations (A.18), (A.21) and (A.23) that for this window the estimated power of the spectral lines does not vary with a change on the data window length T (equivalent to a change in the RBW). Nevertheless, from Equation (A.24) it can be seen that when using this kind of data window the periodogram based PSD estimate of noise-like signals will display the actual PSD level multiplied by $1/T$ (which is the equivalent RBW of the estimate). Thus, when changing the data window length from T_{ref} to T_{new} the periodogram based PSD estimated will vary following

$$10\log_{10}\left(\frac{1/T_{new}}{1/T_{ref}}\right) \quad \text{(dB)} \tag{A.29}$$

which is equivalent to the behaviour described by [1, 2, 42] (see Equation (A.28)). Thus it can be concluded that when comparing simulation-periodogram-DFT based PSD estimates with the FCC's spectral masks the better option would be to use $|X_w(0)|^2=1$ data windows.

A.4.3 Comparisons of Simulation-Periodogram-DFT Based PSD Estimates of Binary to *M*-ary Convolutionally Coded Signals with the FCC's Spectral Mask for Indoor Applications

This section presents simulation-periodogram-DFT based PSD estimates for the same set of signals used in Chapter 5 – Section 5.5.1. Thus simulated PSD estimates are obtained for PPM TH-IR-based UWB signals using the 32-ary Bernoulli map pseudo-chaotic time hopping (PCTH) scheme, the rate 1 feedforward spectral line suppressive (FSLS) binary to 32-ary convolutional encoder with $v=6$ introduced in Chapter 4 – Table 4-1 (Section 4.4.4), and the rate 1 recursive spectral line suppressive (RSLS) binary to 32-ary convolutional encoders with $v=6$ reported in Chapter 5 – Tables 5-1 and 5-2. Furthermore, PSD examples of a non-coded binary PPM TH-IR-

based UWB signal with periodic and perfectly random (ideal) 16-ary TH are provided as well for comparison purposes.

In order to further assess the spectral shaping capabilities of the RSLF binary to M-ary convolutional encoders introduced in Chapter 5, a more biased (compared to the two used in Section 5.5.1) binary Markov source (BMS) will be assumed. Particularly the BMS used to obtain all the results presented in this section has one step transition probabilities given by

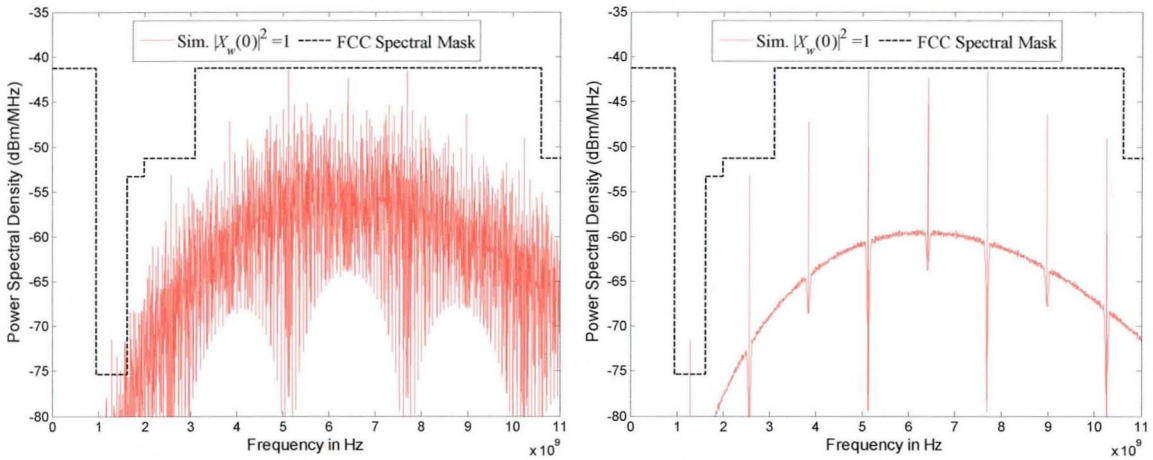
$$p_{y,01} = 9/10 \quad \text{and} \quad p_{y,10} = 1/10 \tag{A.30}$$

with corresponding steady state probabilities

$$\pi_{y,0} = 1/10 \quad \text{and} \quad \pi_{y,1} = 9/10 \tag{A.31}$$

Thus both, the steady state and one step transition probabilities are highly biased.

The comparisons of simulated PSDs with the FCC's spectral mask for the non-coded PPM TH-IR-based UWB signal are shown in Figure A-4. In Figure A-4a the periodic sequence {0, 14, 1, 5, 13, 6, 3, 15, 7, 11, 8, 12, 9, 2, 10, 4} with $\chi_c = 16$ was used for TH while in Figure A-4b perfectly random (ideal) TH uniformly distributed over the set {0, 1,..., 15} was assumed. Note that none of the of the PSDs plotted in Figure A-4 fits under the FCC's spectral mask, specifically in the range covering from 0.96 GHz to 1.610 GHz.



a) PR-TH driven by the periodic sequence {0, 14, 1, 5, 13, 6, 3, 15, 7, 11, 8, 12, 9, 2, 10, 4}

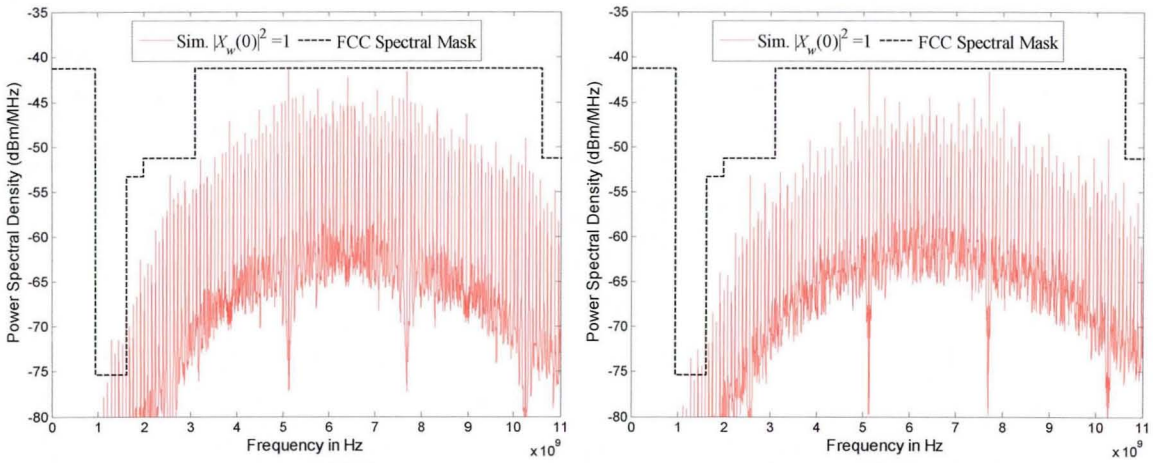
b) Ideal perfectly random TH with c_{lNw+k} uniform distributed over the set {0, 1, ..., 15}

Figure A-4. Simulated PSDs for a TH-IR-based UWB system using binary PPM with pseudo-random and perfectly random 16-ary TH. The simulation used a $|X_w(0)|^2 = 1$ rectangular data window with duration $T = 1$ μs (equivalent to a 1 MHz RBW). The signal parameters are $N_w = 1$, $T_s = 12.5$ ns, $T_r = 12.5$ ns, $T_c = 0.78125$ ns and $T_\beta = 0.39$ ns. The 3rd derivative Gaussian pulse was used with duration $T_w \approx 0.35$ ns.

In Figure A-5 the comparisons of simulated PSDs with the FCC's spectral mask for the schemes employing 32-ary PCTH and the rate 1, $v = 6$, FSLS binary to 32-ary convolutional encoder introduced in Table 4-1 (Section 4.4.4) are shown. Note that for the highly biased BMS with one step transition probabilities given by Equation (A.30), the PSD of the signals generated by these encoders does not fit under the FCC's spectral mask in the range covering from 0.96 GHz to 1.610 GHz.

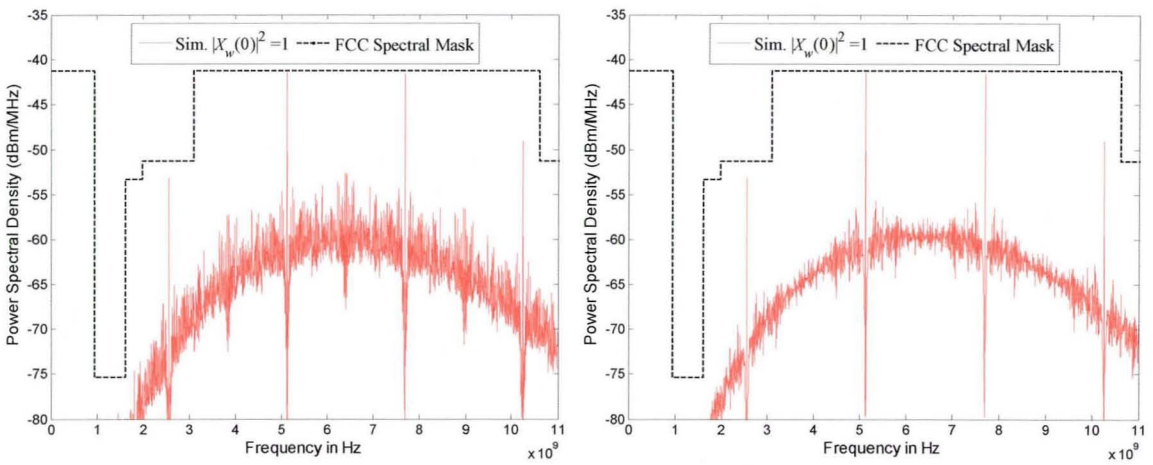
The comparisons of simulated PSDs with the FCC's spectral mask for the schemes employing the rate 1, $v = 6$, RSLS binary to 32-ary convolutional encoders introduced in Tables 5-1 and 5-2 (Section 5.4.2) are shown in Figure A-6. It can be seen in Figure A-6 that the generated signal's

PSD fits under the FCC's spectral mask when using the RSLS encoders. Thus, compliance with the FCC's spectral mask can be achieved even when the data stream at the encoder input is generated by the highly biased BMS with one step transition probabilities defined by Equation (A.30).



a) 32-ary PCTH b) 32-ary FSLS encoder

Figure A-5. Simulated PSDs for a TH-IR-based UWB system using 32-ary PCTH and the rate 1, $v = 6$, FSLS binary to 32-*ary* convolutional encoder from Table 4-1 (Section 4.4.4). The simulation used a $|X_w(0)|^2 = 1$ rectangular data window with duration $T = 1$ μs (equivalent to a 1 MHz RBW). The signal parameters are $N_w = 1$, $T_s = 12.5$ ns, $T_r = 12.5$ ns and $T_\beta = 0.39$ ns. The 3rd derivative Gaussian pulse was used with duration $T_w \approx 0.35$ ns.



a) 32-ary RSLS encoder with feedback polynomial $b(D) = 1 + D^6 \Rightarrow 101$ (Table 5-1)

b) 32-ary RSLS encoder with primitive feedback polynomial $b(D) = 1 + D + D^6 \Rightarrow 141_8$ (Table 5-2)

Figure A-6. Simulated PSDs for a TH-IR-based UWB system using the rate 1, $v = 6$, RSLS binary to 32-ary convolutional encoders reported in Tables 5-1 and 5-2 (Section 5.4.2). The simulation used a $|X_w(0)|^2 = 1$ rectangular data window with duration $T = 1$ μs (equivalent to a 1 MHz RBW). The signal parameters are $N_w = 1$, $T_s = 12.5$ ns, $T_r = 12.5$ ns and $T_\beta = 0.39$ ns. The 3rd derivative Gaussian pulse was used with duration $T_w \approx 0.35$ ns.

A.4.4 Comparisons of Simulation-Periodogram-DFT Based PSD Estimates of Binary Convolutionally Coded Signals with the FCC's Spectral Mask for Indoor Applications

This section presents simulation-periodogram-DFT based PSD estimates for the some of the rate 1/4 binary convolutional encoders reported in Chapter 6 – Table 6-4 (Section 6.5) when used in quaternary biorthogonal PPM (Q-BOPPM) IR-based UWB systems with natural mapping. As

well, simulated PSD estimates are provided for some the best rate 1/4 binary convolutional encoders known reported in [111] and the rate equivalent non-coded scheme introduced in Chapter 6 – Section 6.6 as a reference.

All the figures introduced in this chapter were obtained under the constraint that the binary data stream at the encoder's input is generated by the BMS with one step transition probabilities defined by Equation (A.30) and corresponding steady state probabilities given in Equation (A.31).

The comparisons between the FCC's spectral mask and simulated PSD estimates obtained when using the rate equivalent non-coded pulse repetition scheme are shown in Figure A-7. In Figure A-7b the Barker pseudo-random (PR) direct sequence (DS) {+1, -1, +1, +1, -1, +1, +1, +1, -1, -1, -1} with period $\chi_a = 11$ (used in the IEEE 802.11 standard, [151]) was used for polarity randomisation purposes (see Sections 2.5 and 6.6). It can be seen in Figure A-7 that none of the simulated PSD estimates fits under the FCC's spectral mask. Nevertheless note how the maximum PSD level is reduced upon the introduction of PR-DS multiplication (Barker sequence).

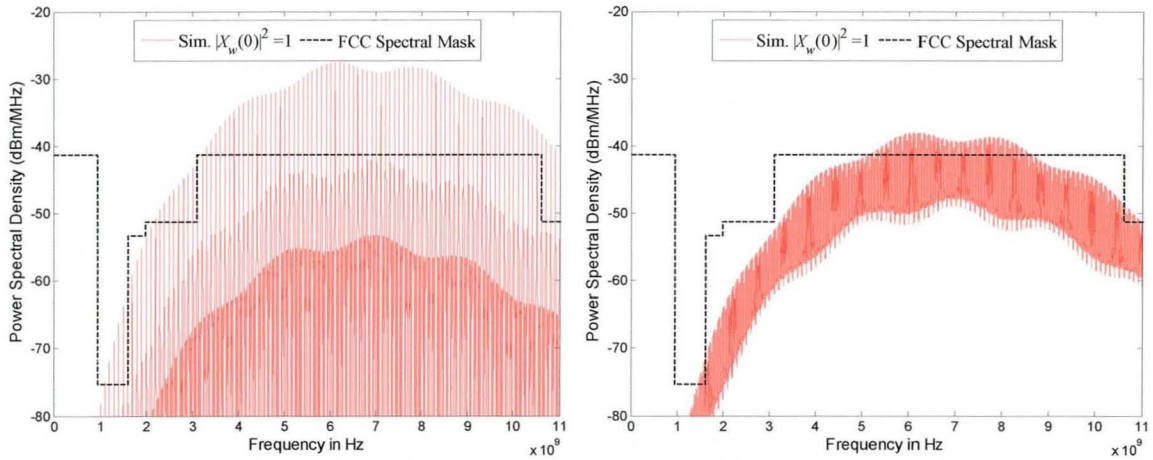
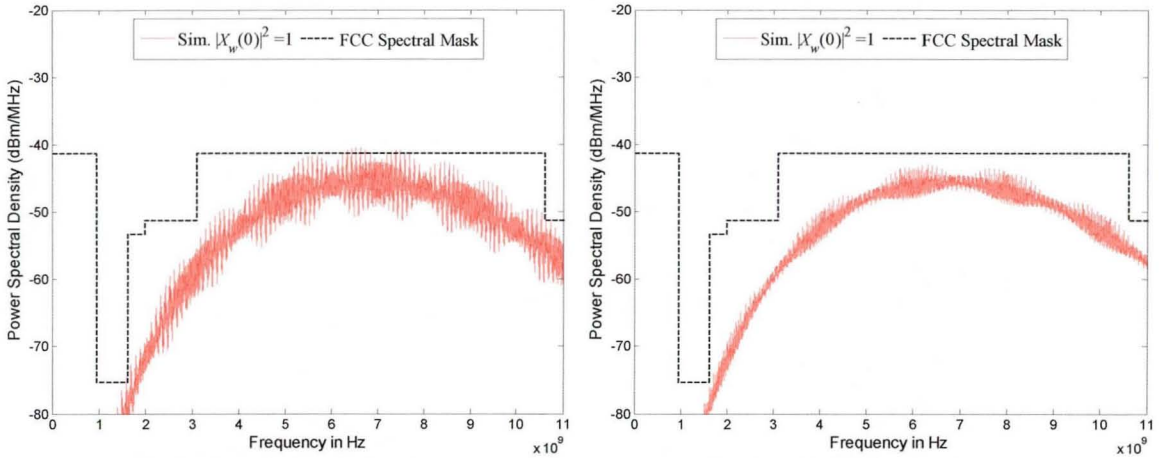


Figure A-7. Simulated PSDs for a non-coded Q-BOPPM IR-based UWB system with four pulses repetition. The simulation used a $|X_w(0)|^2 = 1$ rectangular data window with duration $T = 1$ μs (equivalent to a 1 MHz RBW). The signal parameters are $T_s = 40$ ns, $T_r = 10$ ns and $T_\beta = 0.5$ ns. The 4th derivative Gaussian pulse is used with duration $T_w \approx 0.4$ ns. No TH is considered. The periodic Barker sequence {+1, -1, +1, +1, -1, +1, +1, +1, -1, -1, -1} is used for DS pulse polarity randomisation purposes in Figure A-7b.

In Figure A-8 simulated PSD estimates obtained when using the rate 1/4, v = 4, RSLF binary convolutional encoder reported in Table 6-4 (Section 6.5) are shown. Note how the maximum PSD level of the plots shown in Figure A-8 is less than the maximum PSD level of the plots shown in Figure A-7. Furthermore, it can be seen in Figure A-8b that compliance with the FCC's spectral mask is achieved upon the introduction of PR-DS multiplication by the Barker DS sequence {+1, -1, +1, +1, -1, +1, +1, +1, -1, -1, -1}.

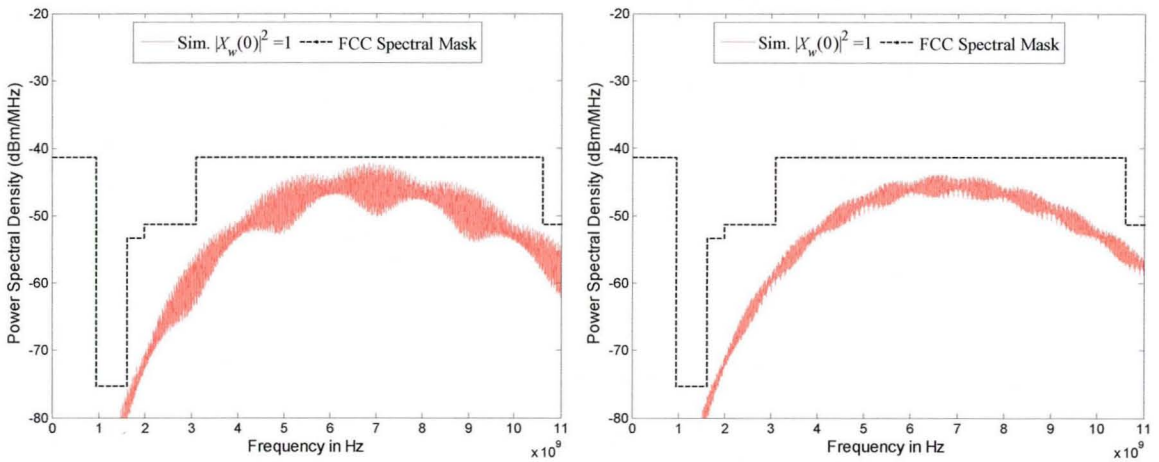
The comparisons of simulated PSDs with the FCC's spectral mask obtained when using the rate 1/4 RSLF binary convolutional encoders with $v = 5$ and $v = 6$ reported in Table 6-4 (Section 6.5) are shown in Figure A-9. It is important to highlight that neither TH nor DS multiplication was assumed for these plots. It can be seen in Figure A-9 that for this highly biased BMS (defined by

Equation (A.30)) compliance with the FCC's spectral mask can be achieved by using these RSLF binary convolutional encoders even without using any polarity randomisation.



a) No PR-DS multiplication b) PR-DS multiplication (Barker sequence)

Figure A-8. Simulated PSDs for a Q-BOPPM IR-based UWB system using the rate 1/4, $v = 4$, RSLF binary convolutional encoder from Table 6-4 (Section 6.5). The simulation used a $|X_w(0)|^2 = 1$ rectangular data window with duration $T = 1$ μs (equivalent to a 1 MHz RBW). The signal parameters are $T_s = 20$ ns, $T_r = 10$ ns and $T_\beta = 0.5$ ns. The 4th derivative Gaussian pulse is used with duration $T_w \approx 0.4$ ns. No TH is considered. The periodic Barker sequence {+1, -1, +1, +1, -1, +1, +1, +1, -1, -1, -1} is used for DS pulse polarity randomisation purposes in Figure A-8b.



a) $v = 5$ b) $v = 6$

Figure A-9. Simulated PSDs for a Q-BOPPM IR-based UWB system using the rate 1/4 RSLF binary convolutional encoders from Table 6-4 (Section 6.5). The simulation used a $|X_w(0)|^2 = 1$ rectangular data window with duration $T = 1$ μs (equivalent to a 1 MHz RBW). The signal parameters are $T_s = 20$ ns, $T_r = 10$ ns and $T_\beta = 0.5$ ns. The 4th derivative Gaussian pulse is used with duration $T_w \approx 0.4$ ns. No TH or DS is considered.

Lastly, the comparisons of simulated PSDs with the FCC's spectral mask obtained when using the best rate 1/4 binary convolutional encoders known with $v = 5$ and $v = 6$ reported [111] are shown in Figure A-10. Similar to Figure A-9 neither TH nor DS multiplication was assumed for these plots. It can be seen in Figure A-10 that none of the simulated PSD estimates fits under the FCC's spectral mask.

As a concluding remark note that the comparison plots presented here further prove the spectral shaping capabilities of the RSLS binary to *M*-ary convolutional encoders introduced in Chapter 5 and the RSLF binary convolutional encoders introduced in Chapter 6. In fact it can be seen that

compliance with the FCC's regulation can be achieved by using these encoders compared with rate equivalent non-coded systems and convolutionally coded systems using the best binary convolutional encoders known.

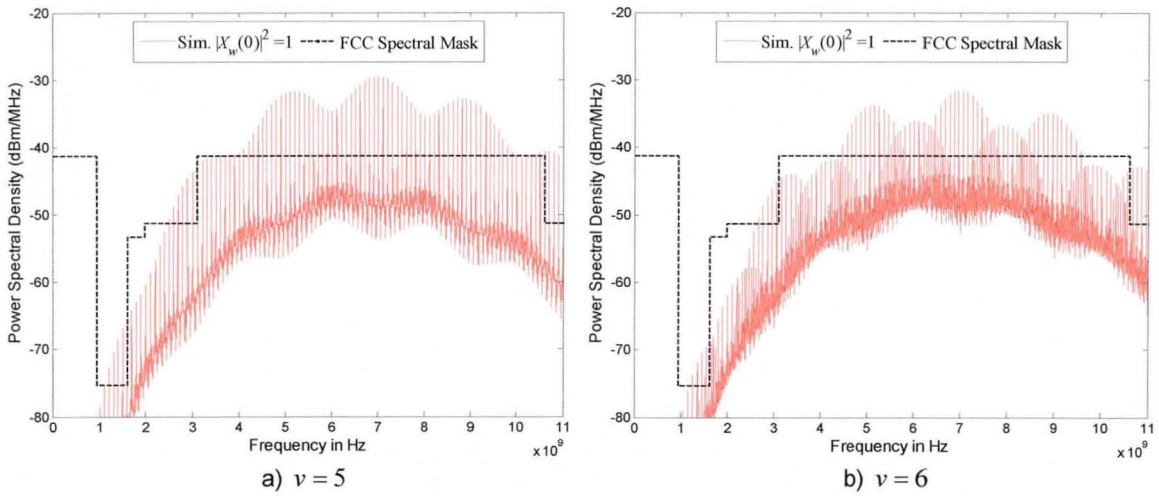


Figure A-10. Simulated PSDs for a Q-BOPPM IR-based UWB system using the best rate 1/4 RSLF binary convolutional encoders known reported [111]. The simulation used a $|X_w(0)|^2 = 1$ rectangular data window with duration $T = 1$ μs (equivalent to a 1 MHz RBW). The signal parameters are $T_s = 20$ ns, $T_r = 10$ ns and $T_\beta = 0.5$ ns. The 4th derivative Gaussian pulse is used with duration $T_w \approx 0.4$ ns. No TH or DS is considered.

Appendix B

IEEE 802.15.3a Multipath Channel Model for UWB Communications (from [153])

The impulse response of the IEEE 802.15.3a multipath channel model for ultra wideband (UWB) communications is described by ([13, 15, 17, 153]):

$$h_i(t) = X_i \sum_{l=1}^{L^i} \sum_{k=1}^{K_l^i} \varphi_{k,l}^i \delta(t - T_l^i - \tau_{k,l}^i) \tag{B.1}$$

where the subscript i stands for the channel realization; X_i represents log-normal shadowing; L^i is the number of received clusters; K_l^i is the number of multipath components received in the l^{th} cluster; $\varphi_{k,l}^i$ are the multipath coefficients of the k^{th} multipath component of the l^{th} cluster; T_l^i is the l^{th} cluster arrival time and $\tau_{k,l}^i$ is the delay of the k^{th} multipath component within the l^{th} cluster.

The proposed model uses the following definitions:

- T_l = the arrival time of the first path of the l^{th} cluster
- Λ = cluster arrival rate
- λ = ray arrival rate; that is, the arrival rate of path within each cluster.

By definition, we have $\tau_{0,l}^i = 0$. The distribution of cluster arrival time and the ray arrival time are given by

$$\begin{aligned} P(T_l \mid T_{l-1}) &= \Lambda \exp\{-\Lambda(T_l - T_{l-1})\}, \qquad l > 0 \\ P(\tau_{k,l} \mid \tau_{(k-1),l}) &= \lambda \exp\{-\lambda(\tau_{k,l} - \tau_{(k-1),l})\}, \quad k > 0 \end{aligned} \tag{B.2}$$

The channel coefficients are defined as follows:

$$\begin{aligned} &\varphi_{k,l}^i = p_{k,l} \xi_l \zeta_{k,l} \\ &20\log 10(\xi_l \zeta_{k,l}) \propto \text{Normal}(\mu_{k,l}, \sigma_1^2 + \sigma_2^2) \end{aligned} \tag{B.3}$$

or

$$|\xi_l \zeta_{k,l}| = 10^{(\mu_{k,l} + n_1 + n_2)/20)} \tag{B.4}$$

where $n_1 \propto \text{Normal}(\mu_{k,l}, \sigma_1^2)$ and $n_2 \propto \text{Normal}(\mu_{k,l}, \sigma_2^2)$ are independent and correspond to the fading on each cluster and ray, respectively,

$$\text{E}\{|\xi_l \zeta_{k,l}|\}^2 = \Omega_0 e^{-T_l/\Gamma} e^{-\tau_{k,l}/\gamma} \tag{B.5}$$

where T_l is the excess delay of bin l and Ω_0 is the mean energy of the first path of the first cluster, and $p_{k,l}$ is equiprobable ± 1 to account for signal inversion due to reflections. Γ and γ are the cluster decay factor and ray decay factor, respectively. The $\mu_{k,l}$ are thus given by

$$\mu_{k,l} = \frac{10\ln(\Omega_0) - 10T_l/\Gamma - 10\tau_{k,l}/\gamma}{\ln(10)} - \frac{(\sigma_1^2 + \sigma_2^2)\ln(10)}{20} \tag{B.6}$$

In the above equations, ξ_l reflects the fading associated with the l^{th} cluster, and $\zeta_{k,l}$ corresponds to the fading associated to the k^{th} ray of the l^{th} cluster. Note that a complex tap model was not adopted here. The complex baseband model is a natural fit for narrowband systems to capture channel behaviour independent of carrier frequency, but this motivation breaks down for UWB systems where a real-valued simulation at RF may be more natural.

Since the log-normal shadowing of the total multipath energy is captured by the term X_i, the total energy contained in the terms $\varphi_{k,l}^i$ is normalized to unity for each realization. This shadowing term is characterized by the following

$$20\log 10(X_i) \propto \text{Normal}(0, \sigma_X^2) \tag{B.7}$$

As mentioned in Chapter 6 the channel model considers four different indoor scenarios and thus it defines four different sets of statistics for each one of them. The parameters for each channel model are given in Table B-1.

Table B-1. IEEE 802.15.3a UWB multipath channel model parameters

Model Parameters	CM 1[1]	CM 2[2]	CM 3[3]	CM 4[4]
Λ [1/nsec] (cluster arrival rate)	0.0233	0.4	0.0667	0.0667
λ [1/nsec] (ray arrival rate)	2.5	0.5	2.1	2.1
Γ (cluster decay factor)	7.1	5.5	2.1	2.1
γ (ray decay factor)	4.3	6.7	7.9	12
σ_1 [dB] (stand. dev. of cluster lognormal fading term in dB)	3.4	3.4	3.4	3.4
σ_2 [dB] (stand. dev. of cluster lognormal fading term in dB)	3.4	3.4	3.4	3.4
σ_X [dB] (stand. dev. of lognormal fading term for total multipath realizations in dB)	3	3	3	3

[1] This model is based on LOS (0-4 m) channel measurements
[2] This model is based on NLOS (0-4 m) channel measurements
[3] This model is based on NLOS (4-10 m) channel measurements
[4] This model was generated to fit a 25 ns RMS delay spread to represent an extreme NLOS multipath channel