Sequency-ordered complex hadamard transforms and their applications to communications and signal processing

Aung, Aye

2009

Aung, A. (2009). Sequency-ordered complex hadamard transforms and their applications to communications and signal processing. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/18688>

<https://doi.org/10.32657/10356/18688>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

**SEQUENCY-ORDERED COMPLEX HADAMARD
TRANSFORMS AND THEIR APPLICATIONS TO
COMMUNICATIONS AND SIGNAL PROCESSING**

AYE AUNG

SCHOOL OF ELECTRICAL & ELECTRONIC ENGINEERING

2009

Sequency-Ordered Complex Hadamard Transforms And Their Applications To Communications And Signal Processing

Aye Aung

School of Electrical and Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2009

Acknowledgements

Firstly, I would like to express my sincere gratefulness to my supervisors, Assoc. Prof. Ng Boon Poh and Dr. Susanto Rahardja for their guidance, valuable advices, suggestions and understanding throughout the entire research journey of my doctorate. Particularly, I am cordially thankful to Professor Ng for his guidance to become an independent researcher as well as for his great concern in matters not related to academic research. Every technical discussion with him was fruitful and really helpful to embark on the next stages. I am also deeply indebted to Dr. Rahardja for his valuable and helpful comments on my papers as well as for his great suggestions to my research. I was inspired by most of his research papers which actually provided me a prominent and countless assistance in writing papers as well as to my technical knowledge to complete this journey. What I learned from them would be valuable for my whole life especially when I embark on my research career.

I also wish to thank Professor Bi Guoan for his sharing of valuable ideas and numerous help in writing an excellent technical paper. My sincere gratitude also goes to all the examiners during my Ph.D. qualifying examination for their useful comments and constructive remarks, which really gave me a great aid for my research.

I wish to express my cordial gratitude to all my fellow graduate students and friends who have given me emotional support as well as the valuable inputs and advices to my research. They are Dr. Joni Polili Lie, Mr. Syed Muhammad Ghazanfar Monir, Ms. Li Xiumei, Mr. Thein Thant Tun, Mr. Bo Bo Wai Maung, Mr. Wen Fuxi, Mr.

Vinod Veera Reddy and Ms. Zhang Ying. My appreciation is also extended to all the technicians and staffs in Information Systems Research Lab for their help, provisions of equipments, and their efforts in providing a friendly and conducive research environment.

This research is made possible and funded by Institute of Infocomm Research (I²R), Singapore. I wish to extend my appreciation to the Institute as well as to the Nanyang Technological University for the various facilities I enjoyed during my journey.

I have not been able to complete this academic journey without the support of my family in Myanmar and the encouragement of my dear wife, Mrs. Phyu Mar Htun. I am particularly indebted to my lovely wife for her endless support, everlasting kindness and cheering me up whenever I was down.

Finally, I would like to thank all the people who have been providing helps and supports throughout the conduct of this research.

Table of Contents

Acknowledgements	i
Table of Contents	vi
Summary	vii
List of Figures	xiv
List of Tables	xvi
List of Notations	xvii
List of Abbreviations	xxi
1 Introduction	1
1.1 General Background	1
1.2 Motivation	3
1.3 Objectives	5
1.4 Major Contributions of the Thesis	6
1.5 Organization of the Thesis	8
2 Review on Hadamard Transforms	11
2.1 Introduction	11
2.2 Hadamard Matrices	11
2.3 Real Hadamard Transforms	13
2.3.1 BIFORE or Hadamard Transform	15
2.3.2 Fast BIFORE Transform Algorithm	17
2.3.3 Orderings of Real Hadamard Transforms	18
2.3.4 Rademacher Functions	20
2.3.5 Sequency-Ordered Walsh Transform	21
2.4 Complex Hadamard Transforms	22
2.4.1 Complex BIFORE Transform	23
2.4.2 Fast Complex BIFORE Transform Algorithm	24
2.4.3 Unified Complex Hadamard Transforms (UCHTs)	25

2.4.4	Fast Algorithms for the UCHT	28
2.5	Conclusion	31
3	Complex Hadamard Transforms with Sequency and Natural Orderings	33
3.1	Introduction	33
3.2	Sequency-Ordered Complex Hadamard Transform (SCHT)	34
3.2.1	Complex Rademacher Functions and their Matrices	34
3.2.2	SCHT Matrices	37
3.2.3	Exponential Form of the SCHT Matrix	38
3.2.4	Properties of the SCHT	41
3.2.5	Fast Algorithm for the SCHT	45
3.2.6	Applications of the SCHT	52
3.2.7	Relationship of the SCHT with the FFT and the UCHT	56
3.2.8	The DFT via the SCHT	61
3.3	Natural-Ordered Complex Hadamard Transform (NCHT)	69
3.3.1	Basic Definitions	69
3.3.2	Properties of the NCHT	74
3.3.3	Fast NCHT Algorithm	79
3.3.4	Relationship of the NCHT with the SCHT	82
3.4	Conclusion	84
4	Pipelined Hardware Structure for the SCHT	86
4.1	Introduction	86
4.2	Fast SCHT Algorithms	88
4.2.1	Decimation-in-Time Fast SCHT	88
4.2.2	Decimation-in-Sequency Fast SCHT	89
4.3	Implementation	92
4.3.1	Delay and Commutator Unit	93
4.3.2	Modified Adder/Subtractor	96
4.4	Discussion	97
4.5	Conclusion	99
5	Application of the SCHT in DS-CDMA Systems	100
5.1	Introduction	100
5.2	Literature Study	102

5.2.1	Spread Spectrum Techniques	102
5.2.2	Direct Sequence Spread Spectrum Technique	104
5.2.3	Performance of a DS-CDMA System	106
5.2.4	Spreading Sequences for DS-CDMA Systems	109
5.3	A Proposed Spreading Sequence for DS-CDMA Systems	114
5.4	Performance Measurement Parameters of the Sequence Set	116
5.4.1	Periodic Mean-Squared Correlation	116
5.4.2	Aperiodic Mean-Squared Correlation	120
5.5	Synchronous DS-CDMA System	121
5.5.1	BER Performance	121
5.5.2	Simulation Results	125
5.6	Asynchronous DS-CDMA System	131
5.6.1	BER Performance over the Multipath Fading Channel	131
5.6.2	Simulation Results	135
5.7	Conclusion	145
6	Application of the SCHT in Image Watermarking	147
6.1	Introduction	147
6.2	Review of Previous Works	150
6.3	Watermark Embedding Algorithm	152
6.3.1	Pseudo-random Permutation of the Watermark	155
6.3.2	Spread Spectrum	156
6.3.3	Low Amplitude Block Selection	156
6.3.4	Amplitude Boost	158
6.3.5	PSK Modulation	161
6.4	Watermark Detection Algorithm	161
6.4.1	PSK Demodulation	163
6.4.2	Inverse Spread Spectrum	163
6.5	Experimental Results	163
6.5.1	Robustness of the Watermarking Scheme	167
6.5.2	Comparison between the SCHT and DFT Based Watermarking Schemes	176
6.5.3	Comparison with the Chen's Scheme	177
6.5.4	Comparison with the Guo's Scheme	178
6.6	Conclusion	180

7	Conjugate Symmetric Sequency-Ordered Complex Hadamard Transform (CS-SCHT)	183
7.1	Introduction	183
7.2	Basic Definitions of the CS-SCHT	185
7.3	Interpretation of Sequency	190
7.4	Properties of the CS-SCHT	194
7.5	Fast Algorithm of the CS-SCHT	202
7.6	Applications of the CS-SCHT	205
7.6.1	Spectrum Estimation	205
7.6.2	Image Compression	212
7.7	Conclusion	219
8	Conclusions and Recommendations	221
8.1	Conclusions	221
8.2	Recommendations for Future Research	226
	Appendix A: Fast Algorithms for a Real Transform Derived from the CS-SCHT	228
	Author's Publications	234
	Bibliography	235

Summary

Complex Hadamard transforms have attracted a lot of research attention throughout the past years due to their simplicity and usefulness to several applications in signal processing, communications and logic design, especially dealing with quadri-phase modulation and complex-valued or multiple-valued functions. Like their real integer-valued counterparts (i.e., real Hadamard transforms), they preserve much of the simplicity as well as the orthogonality, and it does not incur any extra computational cost by introducing the complex elements $\{\pm j\}$ in their transform matrices where $j = \sqrt{-1}$. This thesis aims to introduce a new class of complex Hadamard transforms called sequency-ordered complex Hadamard transforms (SCHTs), which are useful to be employed in various applications of signal processing and communications. As they are named, these particular transforms exhibit sequency ordering which is analogous to frequency in the discrete Fourier transform (DFT). Two versions of the complex Hadamard transforms are introduced in this thesis, one is the SCHAT and the other is the conjugate symmetric SCHAT called the CS-SCHAT.

At first, this thesis reviews some of the existing real and complex Hadamard transforms in the literature. Much effort is spent on their constructions, properties, fast algorithms and applications. Different types of orderings for real Hadamard transforms are also mentioned. Then, SCHAT is introduced along with its construction, properties, fast algorithm and applications. The construction of the SCHAT is based on the complex Rademacher functions which are the extended complex version of the

conventional Rademacher functions. Some of its properties are derived and analyzed. The exponential form of the SCHAT is further derived and the proof of the unitary property of the SCHAT is given. Using the sparse matrix factorization, the fast and efficient algorithm to compute the SCHAT is developed, and its computational complexity is examined. Relations of the SCHAT with the fast Fourier transform (FFT) and the unified complex Hadamard transform (UCHT) are discussed. The fast DFT computation via the SCHAT (with the use of their relationship matrix) is also developed. The applications of the SCHAT in spectrum estimation, direct sequence code division multiple access (DS-CDMA) systems and digital image watermarking are presented in detail. The simulation results are discussed and explained with comparisons. The pipelined hardware structure which is to implement the SCHAT is also proposed in this thesis to support the high-speed real-time applications of the SCHAT. The structure is implemented based on the decimation-in-sequence (DIS) fast SCHAT algorithm which is derived based on the decimation-in-time (DIT) fast SCHAT algorithm and the symmetry property of an SCHAT matrix.

This thesis also introduces the natural-ordered complex Hadamard transform called the NCHAT whose ordering corresponds to that of the Walsh Hadamard transform (WHT). Unlike the SCHAT, the NCHAT is constructed based on the WHT matrix and through the direct block matrix operation (defined in Section 3.3). Some of its properties, including the exponential property of the NCHAT and the shift invariant property of the NCHAT power spectrum, are derived and analyzed. A fast algorithm to compute the NCHAT is also developed using the sparse matrix factorization method. The relationship of the NCHAT with the SCHAT is also discussed.

Lastly, the conjugate symmetric SCHAT (CS-SCHAT) is presented in this thesis. Unlike the SCHAT, the CS-SCHAT possesses a unique property called conjugate symmetry property as the DFT. Therefore, the CS-SCHAT spectrum is conjugate symmetric and only half of the spectral coefficients are necessary to consider for the processing of

the transform, which makes it special as compared to the aforementioned SCHAT and other transforms. This transform is constructed based on the WHT and the direct block matrix operator. Some of its useful properties are discussed and analyzed. A fast algorithm to compute an N -point CS-SCHAT is also derived and its computational complexity is examined as compared with that of the SCHAT. Sequency defined in the CS-SCHAT is interpreted as compared to frequency in the DFT. Consequently, the applications of the CS-SCHAT in spectrum estimation and image compression are discussed. The simulation results reveal that the CS-SCHAT is promising to be employed in such applications.

List of Figures

2.1	(a) The sine and cosine functions with harmonic frequency, and (b) a set of Walsh functions arranged in sequency order.	16
2.2	Signal flow graph of fast forward BT algorithm ($N=8$).	18
2.3	The Hadamard or natural ordered Walsh function series for $N = 8$	19
2.4	The Walsh functions positioned in Paley order for $N = 8$	20
3.1	Zero crossings of sequency in the unit circle of a complex plane.	43
3.2	Flow graph of the final decomposition of fast inverse SCHAT algorithm ($N = 8$).	50
3.3	Magnitude spectra of the DFT, SCHAT and SOWT for the sinusoidal wave of 20 Hz.	53
3.4	The SCHAT and SOWT magnitude spectra with 90° circular phase shift in the time domain.	54
3.5	The flow graph of fast DFT algorithm using the SCHAT.	65
3.6	Magnitude response of the SCHAT matrix ($N = 8$).	67
3.7	Magnitude spectra of the DFT and SCHAT for the periodic square wave of 20 Hz.	68
3.8	Different orderings of the Hadamard transforms.	69
3.9	Signal flow graph of the fast forward NCHAT decomposition ($N = 8$).	81
3.10	Relationship between the sequency and natural orderings of complex Hadamard matrices.	83
4.1	The signal flow graph of the DIT fast SCHAT algorithm ($N = 8$).	90
4.2	The signal flow graph of the DIS fast SCHAT algorithm ($N = 8$).	92
4.3	Pipelined hardware structure for an N -point SCHAT based on the DIS fast algorithm.	93
4.4	Implementation of the DAC unit at stage i : (a) Implementation with shift registers, and (b) implementation with RAMs.	94
4.5	Timing and operations of the delay and commutator unit at stage 1 ($N = 8$).	95

4.6	Modified adder/subtractor.	96
4.7	Waveforms of the control signals for the stage 1.	97
5.1	The block diagram of a DS-SS system model with binary phase modulation: (a) Transmitter, and (b) receiver.	105
5.2	A simplified diagram for a DS-CDMA system with K mobile users: (a) Transmitter model of K users in a CDMA spread spectrum system, and (b) receiver structure for User 0.	107
5.3	Generation of m -sequence using the m -stage shift registers with linear feedback.	110
5.4	Generation of a set of Gold sequences for $m = 5$	111
5.5	A DS-CDMA receiver using the SCHT spreading sequence.	115
5.6	Algorithm for subsection selection of the sequences.	118
5.7	A DS-CDMA system model for K users using orthogonal complex spreading sequences.	122
5.8	BER performances of various spreading sequences in the synchronous DS- CDMA system ($K = 8$).	127
5.9	BER performances of various spreading sequences in the synchronous DS- CDMA system ($K = 20$).	128
5.10	Theoretical BER for different spreading sequences in the synchronous DS- CDMA system ($K = 8$).	129
5.11	Theoretical BER for different spreading sequences in the synchronous DS- CDMA system ($K = 20$).	130
5.12	BER performances of various spreading sequences in the asynchronous DS- CDMA system over the multipath channel ($K = 8$ and $L = 10$).	140
5.13	BER performances of various spreading sequences in the asynchronous DS- CDMA system over the multipath channel ($K = 20$ and $L = 10$).	141
5.14	BER performances of various spreading sequences in the asynchronous DS- CDMA system over the multipath channel ($K = 30$ and $L = 10$).	142
5.15	BER versus the number of users under fixed E_b/N_0 at 8 dB.	143
5.16	BER versus the number of users under fixed E_b/N_0 at 10 dB.	144
6.1	The flow chart of watermark embedding process.	154
6.2	The watermark transformed by random permutation, (a) the original water- mark, and (b) the permuted watermark.	155
6.3	The selected SCHT coefficient to embed the watermark within an 8×8 block.	157

6.4	The phase change of a phasor $Ae^{j\phi}$ subjected to a two-dimensional additive Gaussian noise of standard deviation σ	158
6.5	The signal constellation for the PSK modulation.	161
6.6	The flow chart of watermark extraction process.	162
6.7	(a) The original image of Lena, the watermarked images of Lena using (b) the SCHAT with PSNR = 39.95 dB, (c) the DFT with PSNR = 37.72 dB, and (d) the UCHT with PSNR = 26.41 dB.	165
6.8	(a) The original image of Peppers, the watermarked images of Peppers using (b) the SCHAT with PSNR = 40.31 dB, (c) the DFT with PSNR = 39.47 dB, and (d) the UCHT with PSNR = 27.90 dB.	166
6.9	The SCHAT watermarked images subjected to various attacks: (a) The scaled, (b) the rotated, (c) the cropped (a quarter image), (d) the cropped (central portion remained), (e) the painted, (f) the filtered, (g) the sharpened, (h) the noised, and (i) the phase perturbation attacked.	170
6.10	The corresponding extracted watermarks: (a) The scaled, (b) the rotated, (c) the cropped (a quarter image), (d) the cropped (central portion remained), (e) the painted, (f) the filtered, (g) the sharpened, (h) the noised, and (i) the phase perturbation attacked.	171
6.11	The JPEG encoded versions of the SCHAT watermarked image of Lena and the corresponding extracted watermarks: (a) at 50% quality with NC $\cong$ 1.00, (b) at 40% quality with NC $\cong$ 1.00, (c) at 30% quality with NC=0.98, (d) at 21% quality with NC=0.80, (e) at 20% quality with NC=0.78, and (f) at 10% quality with NC=0.56.	173
6.12	The extracted watermarks from the attacked SCHAT watermarked images of Peppers: (a) The scaled, (b) the rotated, (c) the cropped (a quarter image), (d) the cropped (central portion remained), (e) the painted, (f) the filtered, (g) the sharpened, (h) the noised, and (i) the phase perturbation attacked. .	175
6.13	The extracted watermarks from the JPEG encoded versions of the SCHAT watermarked image of Peppers: (a) at 50% quality with NC $\cong$ 1.00, (b) at 40% quality with NC $\cong$ 1.00, (c) at 30% quality with NC=0.98, (d) at 21% quality with NC=0.81, (e) at 20% quality with NC=0.78, and (f) at 10% quality with NC=0.55.	176

6.14	The extracted watermarks corresponding to the MSE values under various attacks using the introduced scheme: (a) The cropped (1/4), (b) the cropped (1/16), (c) the filtered (4×4), (d) the filtered (3×3), and the noised with (e) $\sigma = 43.39$, (f) $\sigma = 36.16$, (g) $\sigma = 28.93$, (h) $\sigma = 21.69$, (i) $\sigma = 14.46$, (j) $\sigma = 7.23$	179
7.1	The sine and cosine waveforms in the DFT with increasing harmonic order (frequencies).	192
7.2	Approximation of the sinusoidal waves in the DFT using the corresponding waveforms from the CS-SCHT.	194
7.3	Signal flow graph illustrating the computation of an 8-point CS-SCHT. . . .	204
7.4	Magnitude spectra of the DFT, CS-SCHT and SCHT for the sinusoidal wave of 1000 Hz.	208
7.5	The highlighted portion in Figure 7.4.	208
7.6	Magnitude spectra of the DFT, CS-SCHT and SCHT for the sinusoidal wave of 2000 Hz.	209
7.7	The highlighted portion in Figure 7.6.	209
7.8	Magnitude spectra of the DFT, CS-SCHT and SCHT for the sinusoidal wave of 1375 Hz.	210
7.9	The highlighted portion in Figure 7.8.	210
7.10	Magnitude spectra of the DFT, CS-SCHT and SCHT for the sinusoidal wave of 1650 Hz.	211
7.11	The highlighted portion in Figure 7.10.	211
7.12	Signal flow graph of the fast forward algorithm for the real transform derived from the CS-SCHT ($N = 8$).	214
7.13	(a) The original image of Lena, the decompressed images of Lena using (b) the DCT with PSNR = 34.63 dB, (c) the proposed transform derived from the CS-SCHT with PSNR = 31.17 dB, and (d) the WHT with PSNR = 31.85 dB using 16 coefficients.	217
7.14	MSE comparison among the DCT, the proposed transform derived from the CS-SCHT and the WHT with respect to the number of coefficients selected during reconstruction for the Lena image.	217
7.15	(a) The original image of Peppers, the decompressed images of Peppers using (b) the DCT with PSNR = 36.62 dB, (c) the proposed transform derived from the CS-SCHT with PSNR = 32.17 dB, and (d) the WHT with PSNR = 32.81 dB using 16 coefficients.	218

7.16	MSE comparison among the DCT, the proposed transform derived from the CS-SCHT and the WHT with respect to the number of coefficients selected during reconstruction for the Peppers image.	218
A.1	Signal flow graph of the fast inverse algorithm for the real transform derived from the CS-SCHT ($N = 8$).	233

List of Tables

3.1	Binary Notations for $\tilde{p}_r$ and $\hat{p}_r$	39
3.2	Elements for $(-1)^{\sum_{r=0}^{n-1} \tilde{p}_r k_r}$	39
3.3	Elements for $(j)^{\sum_{r=0}^{n-1} \hat{p}_r k_r}$	40
3.4	Illustration for the elements of $(-1)^{\sum_{r=0}^{n-1} \tilde{p}_r k_r} \cdot (j)^{\sum_{r=0}^{n-1} \hat{p}_r k_r}$	40
3.5	Comparison of computational complexity of the DFT via the SCHAT and the radix-2 FFT.	66
3.6	Binary notations for $\check{p}_r$	75
3.7	Elements generated from $(-1)^{\sum_{r=0}^{n-1} p_r k_r}$	75
3.8	Elements generated from $(j)^{\sum_{r=0}^{n-1} \check{p}_r k_r}$	76
3.9	Elements generated from $(-1)^{\sum_{r=0}^{n-1} p_r k_r} \cdot (j)^{\sum_{r=0}^{n-1} \check{p}_r k_r}$	76
5.1	Periodic MS correlation parameters for the SCHAT, UCHAT and WHT sequences with the length of 64.	119
5.2	Periodic MS correlation parameters for the Gold, small set of Kasami and m -sequences with the length of 63.	119
5.3	Aperiodic MS correlation parameters for the SCHAT, UCHAT and WHT sequences with the length of 64.	137
5.4	Aperiodic MS correlation parameters for the Gold, small set of Kasami and m -sequences with the length of 63.	137
6.1	The PSNR values of the SCHAT, DFT and UCHAT watermarked Lena images arranged according to the ascending threshold values.	160
6.2	Comparison of the NC values corresponding to the number of error bits for the attacked watermarked Lena images under various operations between the SCHAT and the DFT based schemes.	171
6.3	Comparison of the NC values corresponding to the number of errors for the JPEG compressed watermarked Lena images between the SCHAT and DFT based schemes.	172

6.4	Comparison of the NC values corresponding to the number of error bits for the attacked watermarked images of Peppers under various operations between the SCHAT and the DFT based schemes.	174
6.5	Comparison of the NC values corresponding to the number of errors for the JPEG compressed watermarked images of Peppers between the SCHAT and DFT based schemes.	175
6.6	Performance comparison between the proposed scheme and the Chen's scheme under different image manipulation attacks for the Lena image.	177
6.7	Comparison of the MSE values between the proposed scheme and the Guo's scheme under different kinds of attacks for the Lena image.	178
7.1	Binary representations for g_r and f_r	195
7.2	Elements generated from $(-1)^{\sum_{r=0}^{n-1} g_r k_r}$	196
7.3	Elements generated from $(-j)^{\sum_{r=0}^{n-1} f_r k_r}$	196
7.4	Elements generated from $(-1)^{\sum_{r=0}^{n-1} g_r k_r} \cdot (-j)^{\sum_{r=0}^{n-1} f_r k_r}$	197
7.5	The truth table for $g_r(k_r \oplus m_r)$ and $g_r k_r \oplus g_r m_r$	199
7.6	The truth table for $(-1)^{k_r \oplus m_r}$ and $(-1)^{k_r + m_r}$	200
7.7	The truth table for $(-j)^{f_r k_r \oplus f_r m_r}$ and $(-j)^{f_r k_r + f_r m_r}$	200
7.8	Comparison of computational complexity to compute the fast CS-SCHAT and fast SCHAT.	205
A.1	Conversion through a bit reversal and converted gray code ($N = 8$).	229

List of Notations

N	the dimension of the transform matrix	2
j	$\sqrt{-1}$; complex element in complex Hadamard transform	3
$\mathbf{H}$	the real or complex Hadamard transform matrix	11
$\mathbf{I}$	the identity matrix	12
$\otimes$	the right hand side Kronecker product	12
$\mathbf{W}$	the Walsh Hadamard transform matrix	13
$w(p, k)$	the element of a WHT matrix	14
$\mathbf{x}$	the input data vector	15
$\mathbf{X}$	the transform coefficient vector	15
t	the normalized time; $0 \leq t \leq 1$	16
$\otimes^n$	the n -time multiple Kronecker product	17
RAD	the Rademacher function	21
sign	the signum function	21
$\mathbf{W}^s$	the sequency-ordered Walsh transform matrix	21
g_r	the r th bit of the binary Gray code	21
$\odot$	the element by element vector multiplication	22
det	the determinant operator of a matrix	23
$ \cdot $	the modulus operator	23
$\{\}^*$	the complex conjugate transpose operation	23
$\mathbf{M}$	the complex BIFORE transform matrix	24
$\mathbf{U}$	the unified complex Hadamard transform matrix	26
$\diamond$	the direct matrix or block matrix operator	26
τ	the elements contained in a basic UCHT matrix; $\{\pm 1, \pm j\}$	26
$u(p, k)$	the element of a UCHT matrix	27
p	the row index of a matrix	27
k	the column index of a matrix	27
p_r	the r th bit of binary representation of the decimal p	28

k_r	the r th bit of binary representation of the decimal k	28
$\mathbf{D}$	the diagonal or block diagonal matrix	30
CRAD	the complex Rademacher function	35
$\mathbf{R}_n$	the complex Rademacher matrix	36
n	$\log_2 N$	36
$\overline{\{ \}}$	the complex conjugate operation	36
b_r	the r th bit of binary representation of the decimal b	37
$h(p, k)$	the element of a complex Hadamard transform matrix	38
$\{ \}^T$	the transpose of a matrix	44
$\mathbf{X}_e$	the coefficient vector with the entries having even indices	46
$\mathbf{X}_o$	the coefficient vector with the entries having odd indices	46
$\mathbf{S}$	the identity matrix which also consists of the elements j	46
$\oplus$	the direct sum operator	46
$\mathbf{P}_b$	the bit reversal permutation matrix	48
$\mathbf{F}$	the discrete Fourier transform matrix	56
W_N	$\exp(-j\frac{2\pi}{N})$; the twiddle factor in $\mathbf{F}$	56
$\boldsymbol{\omega}$	the diagonal matrix comprised of the twiddle factors	57
$\mathbf{x}_e$	the input data vector with the entries having even indices	57
$\mathbf{x}_o$	the input data vector with the entries having odd indices	57
$\mathbf{P}$	the permutation matrix	58
$\mathbf{T}$	the relationship matrix between the DFT and the SCHAT	106
$\gamma(N)$	the number of complex multiplications for an N -point transform ..	66
$\mu(N)$	the number of complex additions for an N -point transform	66
$\mathbf{x}^m$	the data sequence cyclically shifted to the right by m places	77
$\mathbf{R}^m$	the block diagonal unitary matrix	78
$\mathbf{X}^m$	the transformed vector obtained from $\mathbf{x}^m$	78
$P(k)$	the power spectral points	78
$b(p)$	the decimal number obtained by bit-reversed operation of p	83
D_{total}	the total number of complex data stores in the pipeline	98
T_{MAS}	the processing latency for the MAS	98
G_p	the processing gain	105
BW_t	the transmission bandwidth	105
BW_i	the information bandwidth	105
T	the data symbol period	106
T_c	the chip period	106

$s_k(t)$	the transmitted signal from the k th user	106
P_k	the received power from the k th user	106
$b_k(t)$	the binary data sequence from the k th user	106
$a_k(t)$	the spreading sequence from the k th user	106
f_c	the carrier frequency	106
ϕ_k	the carrier phase offset from the k th user relative to referenced user	106
K	the total number of users in the system	107
$r(t)$	the received signal	107
τ_k	the time delay parameter of the k th user relative to referenced user	107
$n(t)$	the additive white Gaussian noise	107
$Y^{(0)}$	the decision statistic variable for User 0	108
I_0	the desired contribution to $Y^{(0)}$	108
I_k	the interference from the k th user	108
η	the Gaussian random variable with zero mean and variance σ^2 ...	108
σ^2	$(N_0T/4)$; the noise variance	109
$N_0/2$	the two-sided noise power spectral density	109
θ_{ac}	the periodic autocorrelation value	110
θ_{cc}	the periodic crosscorrelation value	112
$x_{p,k}$	the element of an SCHT spreading sequence	115
$\theta_{x,y}$	the periodic crosscorrelation function	117
φ_{cc}	the periodic mean-squared crosscorrelation	117
φ_{ac}	the periodic mean-squared autocorrelation	117
R_{cc}	the aperiodic mean-squared crosscorrelation	120
R_{ac}	the aperiodic mean-squared autocorrelation	120
$C_{x,y}$	the aperiodic crosscorrelation function	120
C_x	the aperiodic autocorrelation function	120
$\Re\{\cdot\}$	the real part of a complex number	121
ω_c	$2\pi f_c$; the angular carrier frequency	121
T_b	the bit duration	123
ξ	the multiple access interference	123
P_e	the probability of error	125
A	the signal energy in volts	125
$Q(\cdot)$	the Q -function associated with the Gaussian distribution tail	125
E_b	the energy per bit	125
L	the number of multipaths	132

$b_l(t)$	the previous bit sequence in the l th multipath component	132
θ_{kl}	the phase delay parameter of l th multipath component for User k	132
ζ	the intersymbol interference	132
$b_{k,0}$	the current received bit from the k th user	134
$b_{k,-1}$	the previous received bit from the k th user	134
$R_{k,0}$	the partial crosscorrelation function	134
τ_{kl}	$\tau_{kl} = \tau_k + lT_c$	135
$\mathbf{Z}_N$	the $N \times N$ original real image block or matrix	153
$\mathbf{Y}_N$	the 2-D SCHAT coefficient matrix of $\mathbf{Z}_N$	153
W	the 2-D binary pattern watermark	153
W_p	the scrambled noise-liked pattern watermark	153
d	the binary watermark bit sequence	154
m	the spread bipolar watermark bit sequence	154
R	the watermarked image	154
Δ	the increment	158
$p(e_1, e_2)$	the joint probability density function of two Gaussian variables ..	158
th	the threshold value	159
W_R	the extracted watermark	162
$\Im\{\cdot\}$	the imaginary part of a complex number	192
$\mathbf{F}_R$	the real transform matrix derived from the DFT matrix	191
$\mathbf{H}_R$	the real transform matrix derived from the CS-SCHAT matrix	193
$\oplus$	the modulo-2 addition	198
$\mathbf{H}_P$	the row permuted matrix of $\mathbf{H}_R$	230

List of Abbreviations

DFT	Discrete Fourier transform	1
DCT	Discrete cosine transform	1
DWT	Discrete wavelet transform	1
WHT	Walsh Hadamard transform	1
HT	Haar transform	1
KLT	Karhunen-Loeve transform	1
BIFORE	Binary Fourier representation	3
CBT	Complex BIFORE transform	3
UCHTs	Unified complex Hadamard transforms	3
DS-CDMA	Direct sequence code division multiple access	4
SCHT	Sequency-ordered complex Hadamard transform	4
CS-SCHT	Conjugate symmetric SCHT	6
NCHT	Natural-ordered complex Hadamard transform	6
DIS	Decimation-in-sequency	7
DIT	Decimation-in-time	7
BER	Bit-error-rate	7
CS-NCHT	Conjugate symmetric NCHT	8
FFT	Fast Fourier transform	8
DIF	Decimation-in-frequency	9
SS	Spread spectrum	9
MC-CDMA	Multicarrier code division multiple access	13
BT	BIFORE transform	15
IBT	Inverse BIFORE transform	16
SOWT	Sequency-ordered Walsh transform	21
HSP	Half spectrum property	27
LSB	Least Significant Bit	75
VLSI	Very large scale integration	86

DAS	Delay and commutator	93
MAS	Modified adder/subtractor	93
RAM	Random access memory	93
DS-SSMA	Direct sequence spread spectrum multiple access	100
MAI	Multiple access interference	100
WCDMA	Wideband code division multiple access	101
S-CDMA	Synchronous code division multiple access	101
TWSC	Total weighted squared correlation	101
AWGN	Additive white Gaussian noise	101
FDMA	Frequency division multiple access	103
TDMA	Time division multiple access	103
FH-SS	Frequency-hopped spread spectrum	104
TH-SS	Time-hopped spread spectrum	104
BPSK	Binary phase-shift-keying	106
SNR	Signal-to-noise ratio	106
AC	Autocorrelation	116
CC	Crosscorrelation	116
MS-CC	Mean-squared crosscorrelation	117
MS-AC	Mean-squared autocorrelation	117
ISI	Intersymbol interference	134
LOS	Line of sight	138
DFRNT	Discrete fractional random transform	149
AB	Amplitude boost	149
LABS	Low amplitude block selection	149
CHT	Complex Hadamard transform	151
COV	Coefficients of variance	151
MSE	Mean square error	152
PSNR	Peak signal-to-noise ratio	160
ISS	Inverse spread spectrum	163
NC	Normalized correlation	164
DSI	Dyadic shift invariant	198

Chapter 1

Introduction

1.1 General Background

With the widespread availability of digital device technology, discrete orthogonal transforms have been used extensively in the fields of digital signal processing, communications and logic design. The commonly used orthogonal transforms which are frequently found in the literature are the discrete Fourier transform (DFT), the discrete cosine transform (DCT), the discrete wavelet transform (DWT), the Walsh Hadamard transform (WHT), the Haar transform (HT) and the Karhunen-Loeve transform (KLT).

The concept of orthogonal transformation is to represent a signal vector by the linear combination of a few simple vectors. In other words, it transforms the original signal into another domain in which the characteristics of the signal can be interpreted more easily. For example, the DFT transforms the signal from the time domain into the frequency domain as it is more meaningful to express signal in frequency domain for some of the applications. Based on the properties, different transforms are suited to be employed in different particular applications. For instance, the DCT is popular for its superior energy compaction property, that is, a large fraction of the average energy

of the signal is packed into a relatively few components of the DCT coefficients [1]. It is often used in data and image compression. The DWT is well-known for its multiresolutional analysis for image signal processing.

The HT is useful in signal and image processing applications in which the real-time application is considered as a crucial factor [2, 3]. It has the lowest computational cost among the existing discrete orthogonal transforms. The computational complexity can be reduced to $O(\log_2 N)$ to compute the transform of an N -point data sequence [4]. The KLT is known to be a statistically optimal transform at the price of costly computational requirement. But it is a data-dependent transform since its transformation matrix is defined by the eigenvectors of the covariance matrix of the input sequence [1]. Related applications of such transform include statistical pattern matching, source coding, and certain lossy data compression [5, 6].

The WHT is a fairly simple orthogonal transform whose transform matrix can be conveniently expressed by a real Hadamard matrix. A real Hadamard matrix is a square matrix composed of the entries $\{\pm 1\}$ and whose row vectors are orthogonal to each other. Due to its simplicity in transform implementation with the availability of many fast algorithms, it has been used widely in the fields of digital signal processing and communications. The row vectors of a WHT matrix is interchangeable while preserving its orthogonality property. The arrangement of the row vectors of Hadamard matrices forms certain ordering. Three main orderings which are conventionally used are the Walsh or sequency order, the Hadamard order and the Paley order. Others have been proposed and employed for the special purpose. Which order to be chosen is very much application-oriented. For instance, sequency order is favored for communications and signal processing works such as spectral analysis and filtering [7].

1.2 Motivation

A complex Hadamard transform is a more generalized form of Hadamard transforms and its transformation matrix consists of the elements confined to a set $C_4 = \{\pm 1, \pm j\}$ where $j = \sqrt{-1}$. Even though the transform involves the complex elements, it does not incur any extra computational cost to compute the transform as compared to that of real Hadamard transform. Only additions and subtractions are required to perform such transforms. Much of simplicity and the orthogonality of real Hadamard transform are retained while providing more flexibility to several applications especially dealing with complex-valued functions or signals in communications and digital signal processing.

In the literature, there exists an orthogonal transform called BIFORE or Hadamard transform. It is introduced in [8] and several of its properties had been developed. The transform is based on the Walsh functions which form an ordered set of rectangular waveforms taking two amplitude values $\{\pm 1\}$. It has been found in many useful applications such as signal representation and classification, spectral analysis and synthesis of voice signals, etc [8]. But due to its limitation to the real-valued data sequences, complex version of BIFORE transform called complex BIFORE transform (CBT) which can process the complex-valued input functions was developed in [9]. This transform is based on four-valued complex Walsh functions, which consists of the elements $\{\pm 1, \pm j\}$. Basically, the CBT is defined based on a recursive formula which assigns one class of complex Hadamard matrices involving the diagonalization of higher order matrices and multiple Kronecker products.

In [10, 11], Rahardja and Falkowski proposed a set of complex Hadamard transforms named unified complex Hadamard transforms (UCHTs), which are constructed based on the direct matrix operator, and recursively generated to the higher dimensional matrices by a single Kronecker product. Among the family of the UCHT matri-

ces, half of them possess a unique property called half spectrum property. Therefore, they are efficient to implement the problems with complex integer-valued functions since only half of the spectral coefficients are required for synthesis and analysis. Besides, it has a constant geometry fast algorithm which is well suited to VLSI hardware implementation. The UCHTs have been employed in many applications of digital signal processing, multiple-valued logic design and communications. For instance, the UCHTs are used as spectral techniques in logic design [12, 13], as complex spreading sequences in direct sequence (DS) CDMA communication system [14] and downlink multi-carrier DS-CDMA systems [15]. In addition, the UCHTs are also utilized in image watermarking [13, 16].

Therefore, it should be pointed out that the complex Hadamard transforms have a lot of implications in research. However, most of the abovementioned complex Hadamard transforms do not exhibit certain ordering as in their real integer-valued counterparts (e.g., real Hadamard transforms with different orderings). No work has been presented so far mainly focusing on the ordering of complex Hadamard matrices and no applications are directly related to the ordering of complex Hadamard matrices. Hence, it is an interest to investigate complex Hadamard transform with certain order for some particular applications in research. This is the motivation to propose a new class of complex Hadamard transforms with certain orderings in the literature.

In this thesis, complex Hadamard transforms with sequency ordering called sequency-ordered complex Hadamard transforms (SCHTs) are introduced together with their properties, fast algorithms, pipelined hardware structure for transform implementation as well as their applications in spectrum analysis, CDMA communication systems, image watermarking and image compression. In fact, SCHT is a discrete orthogonal transform which consists of the elements $\{\pm 1, \pm j\}$ and the row vectors of the transform matrix are arranged in an ascending order of sequencies. Sequency describes the number of times that the row vector of a matrix crosses the imaginary axis in the

unit circle over a normalized time base $0 \leq t \leq 1$. It is analogous to frequency in the DFT where the Fourier components are arranged in an increasing harmonic number in terms of sine and cosine waves. Hence, the SCHAT may have the same potential to be employed in some applications as the DFT. One obvious advantage is that the implementation of the SCHAT is much simpler than that of the DFT in terms of computational complexity and hardware structure. Therefore, it motivates us to explore various applications of the SCHAT, where the DFT has been applied.

1.3 Objectives

In most of the applications in signal processing and communications, the need of real-time operations is an indispensable factor in order to achieve the necessary processing throughput and performance. As mentioned in Section 1.2, the SCHAT exhibits sequency ordering which is analogous to frequency in the DFT, hence, the SCHAT may be considered as a good candidate to replace the DFT in some applications while providing faster computational time and significant hardware savings. In this thesis, the main focus is to propose a new class of complex Hadamard transforms and to explore the applications of the proposed transforms, which are mostly suited to sequency ordering. The overall objectives of this research study are as follows:

- To propose a new class of complex Hadamard transform called sequency-ordered complex Hadamard transform (SCHAT) for some applications in communications and signal processing.
- To derive the properties of the SCHAT to reinforce the theoretical background of the SCHAT in the literature.
- To develop the fast SCHAT algorithm and propose pipelined hardware architecture in practical consideration of the transform.

- To explore the applications of the SCHAT where the DFT has been applied by making use of sequency ordering and its simplicity.
- To evaluate the bit-error-rate performance of the SCHAT spreading sequences in direct sequence (DS) CDMA systems.
- To investigate the performance of the SCHAT in image watermarking and spectrum estimation.
- To propose the conjugate symmetric version of the SCHAT known as the conjugate symmetric SCHAT (CS-SCHAT) which is advantageous over previous introduced SCHAT in terms of memory savings in processing of the transform, and to evaluate its performance in the applications of spectrum estimation and image compression.

1.4 Major Contributions of the Thesis

The first major contribution of the thesis is proposing the complex Hadamard transform with sequency ordering (called the SCHAT) which is to be used in communications and signal processing. Besides, the natural-ordered complex Hadamard transform (NCHAT) whose order corresponds to that of the WHT is also introduced. Various properties of these transforms are derived and analyzed. The fast SCHAT and NCHAT algorithms are also developed using the sparse matrix factorization method. The complete theoretical analysis for both transforms are given. The computational algorithm to compute the DFT via the SCHAT is also proposed by using the relationship matrix between the DFT and the SCHAT. Moreover, the literature study on various orthogonal transforms (the WHT and different types of complex Hadamard transforms) is conducted. More emphasis is given to their constructions, properties, fast algorithms and applications.

The second major contribution of the thesis is a proposed pipelined hardware structure for the implementation of the SCHAT (which is suitable to implement the NCHAT as well). The decimation-in-sequence (DIS) radix-2 fast SCHAT algorithm is first derived from the decimation-in-time (DIT) fast SCHAT algorithm so that the algorithm is to be efficiently mapped into the pipelined hardware structure. As motivated by the necessity of real-time operations, this pipelined hardware architecture is able to achieve a full hardware utilization and require only $\log_2 N$ complex adder/subtractors and $2(N - 1)$ complex data stores for an N -point SCHAT.

The third major contribution of the thesis is proposing a new set of complex spreading sequences known as the SCHAT sequences in order to identify the particular users in DS-CDMA systems. These introduced sequences are derived from the row vectors of an SCHAT matrix and they consist of the elements $\{\pm 1, \pm j\}$. It is motivated by the fact that the row vectors of the SCHAT matrices exhibit very low periodic and aperiodic crosscorrelation values and they can be used as the complex spreading sequences in the multiple access environment to suppress the multiple access interference. Firstly, the mean-squared (MS) correlation parameters of a set of the SCHAT sequences are defined and analyzed. The bit-error-rate (BER) performances of the introduced sequences are examined in both synchronous and asynchronous DS-CDMA systems as compared with the other well-known spreading sequences, such as the UCHAT, WHT, Gold, small set of Kasami and m -sequences.

The fourth major contribution of the thesis is proposing a robust transform domain watermarking algorithm based on the SCHAT. It is motivated by sequence ordering of the SCHAT as well as its simpler transform implementation with the use of fast algorithm. The SCHAT is employed as a main processing tool to develop a robust phase watermarking scheme for digital still images. Then, the proposed scheme is compared with the other schemes (especially which employ the DFT) in terms of peak-signal-to-noise ratio of the watermarked image with respect to the original image, and

robustness to various types of image attacking operations.

The last major contribution of the thesis is proposing a new transform known as conjugate symmetric SCHAT (CS-SCHAT) which is derived from conjugate symmetric NCHAT (CS-NCHAT). It is motivated by the conjugate symmetric property of the DFT matrix, which requires only half of the spectrum for processing of signals. Therefore, the CS-SCHAT reduces the memory requirement (as compared to previous introduced SCHAT) for the processing of the transform in the applications while maintaining various useful properties such as orthogonality, sequency property, linearity and so on. Besides, it has been proved that this transform possesses the dyadic shift invariant property, and the CS-SCHAT power spectrum is dyadic shift invariant. The CS-SCHAT is also applied in spectrum analysis and image compression.

1.5 Organization of the Thesis

The rest of the thesis is organized as follows. In Chapter 2, the literature study on the real and complex Hadamard transforms is conducted, especially focusing on the WHT and the UCHAT. Different orderings of the Hadamard matrices are also mentioned.

In Chapter 3, the sequency-ordered complex Hadamard transform is presented. It is generated based on complex Rademacher function which is the extended form of conventional real Rademacher function. Some of its properties are derived and described. The exponential formula to generate the SCHAT matrices is also given. Due to the sparseness of the SCHAT matrix, the fast algorithm to compute an N -point SCHAT is developed and its computational complexity is examined. The relationships of the SCHAT with the fast Fourier transform (FFT) and the UCHAT are discussed. The DFT computation algorithm via the SCHAT is introduced by deriving the relationship matrix between the DFT and SCHAT matrices. Natural-ordered complex Hadamard transform (NCHAT) is also mentioned in that chapter together with its construction,

properties and fast algorithm. The relationship of the NCHT with the SCHT is also explained.

In Chapter 4, the pipelined hardware structure to implement the SCHT is presented. At first, the concepts of the decimation-in-sequence (DIS) and the decimation-in-time (DIT) decompositions of the SCHT transform matrix are introduced, which are analogous to the concepts of the decimation-in-frequency (DIF) and DIT radix-2 FFT algorithms. Then, the DIS radix-2 fast SCHT algorithm is derived from the DIT fast SCHT algorithm with the use of symmetry property of an SCHT matrix since the DIS decomposition method is used to develop the single-path pipelined hardware structure. Then, the pipelined hardware structure is explained in detail. This structure is useful in the practical usage of the transform, especially dealing with high-speed real-time applications.

Chapter 5 proposes a new set of complex spreading sequences for DS-CDMA systems, which are derived from the SCHT. Firstly, the background theory and concepts of spread spectrum (SS) techniques and the models for DS-CDMA systems are described. Various well-known spreading sequences used in DS-CDMA systems are reviewed. The motivation for proposing the SCHT sequences is also mentioned in the beginning of that chapter. The mean-squared correlation values of various sets of the SCHT sequences are computed and analyzed. The bit-error-rate performances of the introduced sequences are examined in both synchronous and asynchronous DS-CDMA systems by comparing it with the other well-known existing sequences. The simulation results are presented and discussed.

In Chapter 6, the SCHT is used in image watermarking to propose a robust watermarking scheme for digital still images. At first, various transform domain watermarking techniques using different orthogonal transforms are reviewed. Then, the details of proposed watermark embedding and detecting procedures are explained. Different types of image attacking operations on the watermarked images are performed in or-

der to evaluate the robustness of the proposed watermarking scheme. Comparisons of the experimental results between the proposed scheme and the other schemes are also provided, and the results reveal that the proposed scheme demonstrates more robustness than the others.

In Chapter 7, the conjugate symmetric versions of both SCHAT and NCHAT are presented. The constructions of these transform matrices are mentioned together with some of their properties, definitions and fast algorithms. The proof for the dyadic shift invariant property of the CS-SCHAT power spectrum is also given. The applications of the CS-SCHAT in spectrum analysis and image compression are presented. The simulation results are discussed and the conclusions are given.

Finally, this thesis is concluded in Chapter 8 together with the suggestions for future works.

Chapter 2

Review on Hadamard Transforms

2.1 Introduction

Discrete orthogonal transforms have been considered as a practical tool for processing and analysis of digital signals and images over the last few decades. The simplest form of orthogonal transforms is the type of Hadamard transforms which are widely used in communications and signal processing applications [7, 17]. There are two main categories of Hadamard transforms, which are (1) real Hadamard transforms, and (2) complex Hadamard transforms. The former transform matrices consist of the elements $\{\pm 1\}$ and the latter is based on the mapping four-valued integers into the unit circle of a complex plane with the elements strictly confined to be $\{\pm 1, \pm j\}$. The subsequent sections will review some of the well-known Hadamard transforms, their constructions, properties, fast algorithms, applications, and different orderings of Hadamard matrices.

2.2 Hadamard Matrices

A Hadamard matrix, $\mathbf{H}_N$, is defined as a square matrix of dimension $N \times N$ in which

1. all entries of a matrix are ± 1 ,

2. any two distinct rows of the matrix are orthogonal, that is,

$$\mathbf{H}_N^T \mathbf{H}_N = \mathbf{H}_N \mathbf{H}_N^T = N \mathbf{I}_N \quad (2.1)$$

where $\mathbf{H}_N^T$ is the transpose of $\mathbf{H}_N$ and $\mathbf{I}_N$ is the identity matrix of dimension $N \times N$. In fact, a Hadamard matrix is a symmetric matrix whose row/column vectors are orthogonal to each other. It is possible to interchange row/column vectors or to change the sign of every element in a row/column without affecting the orthogonality property.

A real Hadamard matrix of size $N \times N$ where $N = 2^n$ is constructed through a recursive Kronecker product. This type of construction is called the *Sylvester* construction and the resultant matrices are known as the *Sylvester-Hadamard* matrices after *Sylvester* (1867). Let $\mathbf{H}_N$ be a Hadamard matrix of dimension $N \times N$, it can be generated as [17]

$$\mathbf{H}_N = \mathbf{H}_2 \otimes \mathbf{H}_{N/2} \quad (2.2)$$

where $\otimes$ denotes a right hand side Kronecker product operator, $N = 2^n$ and

$$\mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Let $\mathbf{A}$ be $m \times n$ matrix and $\mathbf{B}$ be $k \times l$ matrix, then the right hand side Kronecker product of $\mathbf{A}$ and $\mathbf{B}$ is an $mk \times nl$ block matrix which is defined as follows:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{0,0}\mathbf{B} & a_{0,1}\mathbf{B} & \cdots & a_{0,n-1}\mathbf{B} \\ a_{1,0}\mathbf{B} & a_{1,1}\mathbf{B} & \cdots & a_{1,n-1}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m-1,0}\mathbf{B} & a_{m-1,1}\mathbf{B} & \cdots & a_{m-1,n-1}\mathbf{B} \end{bmatrix} \quad (2.3)$$

where $a_{i,j}$ is the (i th, j th) element of the matrix $\mathbf{A}$.

2.3 Real Hadamard Transforms

The real Hadamard transform is a linear transform which could represent a data vector by linear combination of the column vectors of a transformation matrix. The transformation matrix is a square matrix which can be conveniently expressed by a Hadamard matrix. One of the most well-known real Hadamard transforms is the Walsh Hadamard transform (WHT) which is widely used in many areas of digital signal processing applications such as digital filtering, face recognition, image watermarking [18–20], and communications, for example, multicarrier CDMA (MC-CDMA) system as the spreading sequences [21] and multiband OFDM (MB-OFDM) ultrawideband system [22]. Its transform matrix is a square and symmetric matrix defined as

$$\mathbf{W}_N = \begin{bmatrix} \mathbf{W}_{N/2} & \mathbf{W}_{N/2} \\ \mathbf{W}_{N/2} & -\mathbf{W}_{N/2} \end{bmatrix} \quad (2.4)$$

where $\mathbf{W}_1 = 1$. Especially, if $N = 2^n$,

$$\mathbf{W}_N = \mathbf{W}_2 \otimes^n = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes^n \quad (2.5)$$

where $\otimes^n$ denotes the right hand side Kronecker product applied n times. For example, if $n = 1$, $\mathbf{W}_2 \otimes^1 = \mathbf{W}_2$.

Each element $w(p, k)$ of a WHT matrix can also be constructed using the Boolean synthesis as follows:

$$w(p, k) = (-1)^{\sum_{r=0}^{n-1} p_r k_r} \quad (2.6)$$

where $w(p, k)$ is the (p th, k th) element of a WHT matrix, p_r and k_r are the r th bits of the binary representations of decimal integers p and k respectively, that is, $(p)_{10} = (p_{n-1}, p_{n-2}, \dots, p_0)_2$ and $(k)_{10} = (k_{n-1}, k_{n-2}, \dots, k_0)_2$. Since $(-1)^{p_r k_r} = 1 - 2p_r k_r$, we can write $w(p, k)$ in another form as

$$w(p, k) = \prod_{r=0}^{n-1} (1 - 2p_r k_r). \quad (2.7)$$

Besides, the WHT matrix can be factorized into the product of n identical matrices as follows:

$$\mathbf{W}_N = \mathbf{A}^n \quad (2.8)$$

where $n = \log_2 N$ and

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 1 & \cdots & 0 & 0 \\ & & & \vdots & & & \\ 1 & -1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & -1 & \cdots & 0 & 0 \\ & & & \vdots & & & \\ 0 & 0 & 0 & 0 & \cdots & 1 & -1 \end{bmatrix}.$$

This structure is useful to implement the transform using the constant geometry iterative hardware architecture in which the hardware can be reduced to only one stage,

and the output can be fed back to the input n times.

2.3.1 BIFORE or Hadamard Transform

In [8], BIFORE which stands for *Binary Fourier Representation* was introduced and several of its properties are developed. They are also called the Hadamard transform because the transformation matrices are defined from the Hadamard matrices which are constructed through a recursive Kronecker product. BIFORE is similar to the Fourier harmonic analysis in both geometrical and analytical characteristics except that the Fourier bases are the sinusoidal functions with harmonic frequencies whereas the BIFORE bases are the Walsh functions. Since the Walsh functions are the square waves, their functional values can be $\{\pm 1\}$. Hence, the advantage of BIFORE over the Fourier analysis is its simplicity of square waves relative to the sinusoidal waves or sampled sinusoidal waves. Besides, as the transformation only involves additions and subtractions, the BIFORE transform is computationally efficient.

The first eight of the Fourier sinusoids are shown in Figure 2.1-a together with the corresponding Walsh functions in Figure 2.1-b. It should be pointed out that the Walsh functions are infinitely clipped versions of the Fourier sinusoidal waves. The total number of zero crossings for both the sinusoids and corresponding Walsh functions are the same.

The BIFORE or Hadamard transform (BT) of an N -point data column vector $\mathbf{x}_N = [x(0), x(1), \dots, x(N-1)]^T$ where T stands for the transpose is defined as [8]

$$\mathbf{X}_N = \frac{1}{N} \mathbf{H}_N \mathbf{x}_N \quad (2.9)$$

where N is a power of 2, $\mathbf{X}_N = [X(0), X(1), \dots, X(N-1)]^T$ is the transformed column vector called the BT coefficient vector, $\mathbf{H}_N$ is the BIFORE transform matrix of size $N \times N$ (symmetric and orthogonal matrix). The signal can be recovered uniquely

Figure 2.1: (a) The sine and cosine functions with harmonic frequency, and (b) a set of Walsh functions arranged in sequency order.

from the inverse BIFORE transform (IBT), that is,

$$\mathbf{x}_N = \mathbf{H}_N \mathbf{X}_N. \quad (2.10)$$

Sampling the Walsh function series at $N = 2^n$ equally spaced sampled points over the normalized time base $t \in [0, 1)$ results in an $N \times N$ matrix of the elements $\{\pm 1\}$. The row vectors of the matrix generated by this method can be rearranged to form a particular order, giving a matrix which is similar to a WHT matrix mentioned earlier. Therefore, the BIFORE transform matrix is nothing but just a Hadamard matrix which can be generated recursively based on $\mathbf{H}_2$ as follows:

$$\mathbf{H}_N = \mathbf{H}_2 \otimes^n = \underbrace{\mathbf{H}_2 \otimes \cdots \otimes \mathbf{H}_2}_{n \text{ times}} \quad (2.11)$$

where $\otimes^n$ denotes the n -time multiple Kronecker product, and

$$\mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

2.3.2 Fast BIFORE Transform Algorithm

The BIFORE transform matrix is the Hadamard matrix constructed through a successive Kronecker product, hence, several well-known techniques in developing fast algorithms for this type of transform are applicable [7,17]. In this subsection, we shall mention the fast BIFORE algorithm using the matrix factorization approach. Let us, for example, consider $N = 8$. Then (2.11) can be written as

$$\mathbf{H}_8 = \begin{bmatrix} \mathbf{H}_4 & \mathbf{H}_4 \\ \mathbf{H}_4 & -\mathbf{H}_4 \end{bmatrix}. \quad (2.12)$$

This can be factorized as

$$\mathbf{H}_8 = \begin{bmatrix} \mathbf{H}_4 & \\ & \mathbf{H}_4 \end{bmatrix} \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{I}_4 & -\mathbf{I}_4 \end{bmatrix} \quad (2.13)$$

where $\mathbf{I}_4$ is the identity matrix of size 4×4 and the blank spaces in the matrix represent the zero elements. $\mathbf{H}_4$ can be further factorized in terms of $\mathbf{H}_2$, that is,

$$\mathbf{H}_8 = \begin{bmatrix} \mathbf{H}_2 & & & \\ & \mathbf{H}_2 & & \\ & & \mathbf{H}_2 & \\ & & & \mathbf{H}_2 \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & \mathbf{I}_2 & & \\ \mathbf{I}_2 & -\mathbf{I}_2 & & \\ & & \mathbf{I}_2 & \mathbf{I}_2 \\ & & \mathbf{I}_2 & -\mathbf{I}_2 \end{bmatrix} \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{I}_4 & -\mathbf{I}_4 \end{bmatrix}. \quad (2.14)$$

This will result in fast BT algorithm. The signal flow graph for fast forward BT is shown in Figure 2.2.

Figure 2.2: Signal flow graph of fast forward BT algorithm ($N=8$).

2.3.3 Orderings of Real Hadamard Transforms

The Walsh functions are an orthonormal set of functions which form an ordered set of square waveforms taking only two amplitude values $\{\pm 1\}$. A Walsh function series can be arranged in a number of ways to obtain different orderings. As mentioned earlier, the row vectors of a Hadamard matrix are a sampled version of a Walsh functions series. Hence, there are different ordered versions of the Hadamard matrices based on the arrangement of the Walsh functions. Three main ordering conventions are commonly used [7, 23] and others have been proposed and employed for different purposes. The commonly used orderings are as follows:

1. *Sequency order*: The row vectors of a matrix are arranged in an ascending order of zero crossings. Sequency is defined as one half of the average number of zero crossings per unit time interval. It describes a periodic repetition rate which is independent of the type of waveforms used. It is directly related to frequency of the DFT in which

the Fourier sinusoids are arranged in increasing harmonic number. It can be seen that frequency is merely regarded as a special measurement of sequency, but which is only applicable to the sinusoidal waveforms. This order is also called the *Walsh order*. The Walsh functions arranged in sequency order is illustrated in Figure 2.1-b.

2. *Hadamard order*: The real Hadamard matrices constructed through the successive Kronecker product follow this ordering. For example, the WHT matrix exhibits the Hadamard ordering. This order is also known as the *natural order* and its corresponding Walsh functions are shown in Figure 2.3.

3. *Paley order*: This is the order obtained from the Hadamard matrix generated by the successive Rademacher functions. The Paley order has certain computational and analytical advantages. Figure 2.4 shows the Paley ordered Walsh functions. This order is also called the *dyadic order*.

Figure 2.3: The Hadamard or natural ordered Walsh function series for $N = 8$.

Figure 2.4: The Walsh functions positioned in Paley order for $N = 8$.

Each order has its own particular advantages in applications. The choice of orderings depends on the particular applications [24]. In general, the dyadic and natural orderings are used in mathematical work, in image transmission and for computational efficiency whereas the sequency order is favored to be employed in communications and signal processing work such as spectral analysis and filtering [7, 25].

2.3.4 Rademacher Functions

Rademacher functions are an important, incomplete but orthogonal set of functions, which represent a series of rectangular pulses or square waves [7]. Although they form an incomplete series having odd symmetry, it is possible to generate the other function series which exhibit either odd or even symmetry. Therefore, a complete series can be derived from an incomplete set of functions. The Rademacher functions have two

arguments r and t such that $\text{RAD}(r, t)$ has 2^{r-1} periods of rectangular waves over a normalized time base $t \in [0, 1)$. The function values can be $+1$ and -1 based on the values of t . They are derived from the sinusoidal functions which have the identical zero crossings followed by hard-limiting using the *signum* function as follows:

$$\text{RAD}(r, t) = \text{sign}(\sin 2^r \pi t) \quad (2.15)$$

where $r = 0, 1, 2, \dots$, and $0 \leq t < 1$ and

$$\text{sign}(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases} \quad (2.16)$$

for any real number x . They are important principally because other complete series, such as the Walsh function series, can be derived from them.

2.3.5 Sequency-Ordered Walsh Transform

The real Hadamard matrices complying the harmonics of sine-cosine functions of the DFT are referred to as the sequency-ordered Walsh transform (SOWT) matrices. As illustrated in Figure 2.1, the SOWT is an approximation of sine and cosine waves in the DFT by using the square waves. The construction of the SOWT is based on a product series of the Rademacher functions through a Gray code [7] as follows:

$$\mathbf{W}_N^s(p, k) = \prod_{r=0}^{n-1} \text{RAD}(r, k)^{g_r} \quad (2.17)$$

where $N = 2^n$, $\mathbf{W}_N^s(p, k)$ is the (p th, k th) element of the SOWT matrix $\mathbf{W}_N^s$, $\text{RAD}(r, k)$ is the (r th, k th) element of the discrete Rademacher function sampled at N equally spaced points over a normalized time $0 \leq t \leq 1$, g_r is the r th bit of the

binary Gray code of the decimal p and $g_r = 0$ or 1 . Let us, for example, consider $N = 8$. To find $\mathbf{W}_N^s(6, k)$, we first express 6 in the binary code as 110 and then convert it into the Gray code as 101. Hence,

$$\mathbf{W}_8^s(6, k) = \text{RAD}(2, k) \odot \text{RAD}(0, k) \quad (2.18)$$

where $\odot$ is the operator for the element by element vector multiplication. Since, the binary Grade code is 101, the index values 2 and 0 refer to the ones positioned in the Grade code. As such the whole 8×8 SOWT matrix will be

$$\mathbf{W}_8^s = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix} \quad (2.19)$$

The SOWT matrix can also be obtained by rearranging the row vectors of the WHT matrix through a Gray code and bit reversal operation [7]. Since SOWT is a sequency-ordered version of the WHT, several well-known techniques to develop its fast algorithm are available [7, 25, 26].

2.4 Complex Hadamard Transforms

The complex Hadamard transform is also a type of linear transforms. Its transformation matrix is conveniently expressed by the complex Hadamard matrix. In fact, the

complex Hadamard matrix is a more generalized form of the real Hadamard matrix by introducing the complex elements $\{\pm j\}$ in the transform matrix. It is defined by a square matrix which is composed of the elements from a set $C_4 = \{\pm 1, \pm j\}$ and its row vectors are orthogonal to each other in the complex domain. Let $\mathbf{H}_N$ be the complex Hadamard transform matrix of dimension $N \times N$ where $N = 2^n$ then $\mathbf{H}_N$ is orthogonal in the complex domain [10] as

$$|\det \mathbf{H}_N| = N^{N/2} \quad (2.20)$$

and

$$\mathbf{H}_N \mathbf{H}_N^* = \mathbf{H}_N^* \mathbf{H}_N = N \mathbf{I}_N \quad (2.21)$$

where $\mathbf{H}_N^*$ denotes the complex conjugate transpose of $\mathbf{H}_N$ and $\mathbf{I}_N$ is the identity matrix of order $N \times N$. This kind of transforms is useful to be employed in the applications dealing with complex-valued functions and quadri-phase modulation [27–29]. The following subsections will mention two classes of complex Hadamard transforms which are the complex BIFORE transform and the unified complex Hadamard transforms.

2.4.1 Complex BIFORE Transform

Complex BIFORE transform is introduced in [9] and several of its properties are developed [9, 30]. It is based on four complex integer elements $\{\pm 1, \pm j\}$. The complex BIFORE transform (CBT) of an N -point input column vector $\mathbf{x}_N = [x(0), x(1), \dots, x(N-1)]^T$ where $N = 2^n$ is defined as

$$\mathbf{X}_N = \frac{1}{N} \mathbf{M}_N \mathbf{x}_N \quad (2.22)$$

where $\mathbf{X}_N = [X(0), X(1), \dots, X(N-1)]^T$ is the transformed complex column vector called the CBT coefficient vector, and $\mathbf{M}_N$ is the CBT transformation matrix which is orthogonal in the complex domain and defined as [9]

$$\mathbf{M}_N = \left[\begin{array}{c|c} \mathbf{M}_{N/2} & \mathbf{L}_2 \otimes \mathbf{W}_{N/4} \\ \hline \mathbf{M}_{N/2} & -\mathbf{L}_2 \otimes \mathbf{W}_{N/4} \end{array} \right] \quad (2.23)$$

where $\otimes$ is the right hand side Kronecker product, $\mathbf{W}_{N/4}$ is the $(\frac{N}{4} \times \frac{N}{4})$ WHT matrix, and

$$\mathbf{L}_2 = \begin{bmatrix} 1 & 1 \\ -j & j \end{bmatrix}.$$

The signal can be recovered uniquely from the inverse CBT as

$$\mathbf{x}_N = \mathbf{M}_N^* \mathbf{X}_N \quad (2.24)$$

where $\{ \}^*$ denotes the complex conjugate transpose operation.

2.4.2 Fast Complex BIFORE Transform Algorithm

The fast CBT algorithm can be developed by either matrix partitioning or matrix factorization [31]. For example, we consider $N = 8$, (2.23) becomes

$$\mathbf{M}_8 = \left[\begin{array}{c|c} \mathbf{M}_4 & \mathbf{L}_2 \otimes \mathbf{W}_2 \\ \hline \mathbf{M}_4 & -\mathbf{L}_2 \otimes \mathbf{W}_2 \end{array} \right]. \quad (2.25)$$

But since

$$\mathbf{L}_2 \otimes \mathbf{W}_2 = \left[\begin{array}{c|c} \mathbf{W}_2 & \mathbf{W}_2 \\ \hline -j\mathbf{W}_2 & j\mathbf{W}_2 \end{array} \right],$$

(2.25) can be further factorized as

$$\mathbf{M}_8 = \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{I}_4 & -\mathbf{I}_4 \end{bmatrix} \left[\begin{array}{c|c} \mathbf{M}_2 & \mathbf{L}_2 \\ \hline \mathbf{M}_2 & -\mathbf{L}_2 \end{array} \right] \begin{bmatrix} \mathbf{W}_2 & \mathbf{W}_2 \\ \hline -j\mathbf{W}_2 & j\mathbf{W}_2 \end{bmatrix} \quad (2.26)$$

where

$$\mathbf{M}_2 = \mathbf{W}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Finally, we can factorize it into a product of sparse matrices, which will be shown as

$$\mathbf{M}_8 = \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{I}_4 & -\mathbf{I}_4 \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & \mathbf{I}_2 & & \\ \mathbf{I}_2 & -\mathbf{I}_2 & & \\ & & \mathbf{I}_2 & \mathbf{I}_2 \\ & & -j\mathbf{I}_2 & j\mathbf{I}_2 \end{bmatrix} \begin{bmatrix} \mathbf{M}_2 & & & \\ & \mathbf{L}_2 & & \\ & & \mathbf{M}_2 & \\ & & & \mathbf{M}_2 \end{bmatrix}. \quad (2.27)$$

Substituting (2.27) into (2.22) for $N = 8$ will result in fast forward CBT algorithm. The computational cost can be reduced to $O(N \log_2 N)$ for this type of transforms and the multiplications by the factor j can be performed by the swapping and negating operations (which will be mentioned in Chapter 4).

2.4.3 Unified Complex Hadamard Transforms (UCHTs)

Unified complex Hadamard transforms called UCHTs are proposed in [10, 11] by Rashedja and Falkowski. These transforms are defined using a single unifying mathematical formula, hence, the word unified is used in the name of family of these transforms. They consist of the elements confined to four complex integer values $\{\pm 1, \pm j\}$ and are suitable to process the problems with complex or multiple-valued functions. The

UCHTs are especially utilized in multiple-valued logic systems [12, 13] and CDMA communication systems as complex spreading sequences [14, 32].

The construction of the UCHT transformation matrices is based on the WHT matrix and the direct matrix operator. The higher dimensional matrices are recursively generated by using a single Kronecker product. The transformation matrices describing the elements of the UCHTs are the complex Hadamard matrices. Let $\mathbf{U}_N$ be the transformation matrix for an arbitrary element of the UCHT and $N = 2^n$. First, the basic transformation core matrix of size 2×2 is defined by the unifying formula as [11]

$$\mathbf{U}_2 = \mathbf{W}_2 \diamond \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \diamond \begin{bmatrix} 1 & \tau_3 \end{bmatrix} \quad (2.28)$$

where $\mathbf{W}_2$ is the WHT matrix of size 2×2 , $\diamond$ denotes the direct matrix operator as defined in [10, 11], τ_1, τ_2 and τ_3 are the elements from a set $C_4 = \{\pm 1, \pm j\}$. The higher dimensions are generated recursively based on the core matrix as

$$\mathbf{U}_N = \mathbf{U}_2 \otimes^n \quad (2.29)$$

where $n = \log_2 N$ and $\otimes^n$ is the right hand side Kronecker product applied n times. It is obvious from (2.28) that there are altogether $4^3 = 64$ members of the UCHT matrices with different combinations of τ_1, τ_2 and τ_3 in one single dimension N . The WHT is simply one of them. Some important and useful properties of the UCHTs are described next.

Unitary Property: The UCHT is an orthogonal transform whose transformation matrices are orthogonal in the complex domain as

$$\mathbf{U}_N \mathbf{U}_N^* = \mathbf{U}_N^* \mathbf{U}_N = N \mathbf{I}_N \quad (2.30)$$

where $\mathbf{U}_N^*$ is the complex conjugate transpose of $\mathbf{U}_N$ and $\mathbf{I}_N$ is the identity matrix of order N .

Definition: Let $\mathbf{x}_N = \{x(i)\}_{i=0}^{N-1}$ be the coded data sequence which may be real or complex number based on a particular coding of the data. Then, the UCHT of the data sequence is defined in matrix form as

$$\mathbf{X}_N = \mathbf{U}_N \mathbf{x}_N \quad (2.31)$$

where $\mathbf{X}_N = [X(0), X(1), \dots, X(N-1)]^T$ is the transformed complex column vector. Since the transform is orthogonal, the data sequence can be uniquely recovered by the inverse transform, that is,

$$\mathbf{x}_N = \frac{1}{N} \mathbf{U}_N^* \mathbf{X}_N. \quad (2.32)$$

Half Spectrum Property: Among the 64 members of the UCHT matrices, half of them possess a unique property called half spectrum property (HSP), which implies that only half of the spectral coefficients are required for analysis and synthesis. The necessary condition to have such property is that τ_3 in (2.28) must be imaginary [10]. 32 matrices out of 64 UCHT matrices satisfy the condition for the HSP. The elements $u(p, k)$ of the UCHT matrices with the HSP are mathematically related by

$$u(N-1-p, k) = (\tau_1 \tau_2)^\gamma \overline{u(p, k)} \quad (2.33)$$

where $\gamma = (n \bmod 4)$, $n = \log_2 N$ and $\overline{u(p, k)}$ is the complex conjugate of $u(p, k)$. With the N -point data transformed using (2.31), the HSP-UCHT coefficients $X(k)$ can be expressed as

$$X(N-1-k) = (\tau_1 \tau_2)^\gamma \overline{X(k)} \quad (2.34)$$

where $k = 0, 1, \dots, (N/2) - 1$. The proof is given in [10]. Therefore, only half of the

spectral coefficients are necessary for data analysis and synthesis. This will save the computational cost of the UCHT by half.

Exponential Property of the UCHT: The UCHT matrices can be expressed using the exponential formula as the WHT. Let $u(p, k)$ be an element at row p and column k of any UCHT matrix where $0 \leq p, k \leq N - 1$. Then, [33]

$$u(p, k) = (-1)^{\sum_{r=0}^{n-1} p_r k_r} \cdot (\tau_1)_{r=0}^{n-1} \cdot (\tau_2)_{r=0}^{n-1} \cdot (\tau_3)_{r=0}^{n-1} \cdot (\tau_4)_{r=0}^{n-1} \quad (2.35)$$

where $n = \log_2 N$, $(p_{n-1}, p_{n-2}, \dots, p_0)$ and $(k_{n-1}, k_{n-2}, \dots, k_0)$ are the binary representations of the decimal integers p and k respectively, and $\tau_1, \tau_2, \tau_3 \in \{1, -1, j, -j\}$. This form of the UCHT is useful to derive some other parameters such as correlation properties of the UCHT sequences in CDMA communication system [14], the relationship between the UCHT and the WHT, etc.

2.4.4 Fast Algorithms for the UCHT

Since the construction of the UCHTs is based on the Kronecker product, the techniques used to develop fast WHT algorithm may be applied to derive fast algorithms for the UCHT as well. In particular, a constant geometry fast butterfly [34] and sparse matrix factorization or matrix partitioning of the UCHT matrices leads to the fast algorithms. In this subsection, we shall present the fast UCHT algorithm using matrix factorization based on the interpretation of the Kronecker products applied n times [35]. As in (2.29), the UCHT matrices of dimension $2^n \times 2^n$ can be obtained by

$$\mathbf{U}_{2^n} = \mathbf{U}_2 \otimes \mathbf{U}_{2^{n-1}}. \quad (2.36)$$

From (2.28), $\mathbf{U}_2$ can be expressed as

$$\mathbf{U}_2 = \begin{bmatrix} \tau_1 & \tau_1 \tau_3 \\ \tau_2 & -\tau_2 \tau_3 \end{bmatrix} \quad (2.37)$$

where $\tau_1, \tau_2, \tau_3 \in \{1, -1, j, -j\}$. Let

$$\mathbf{U}_2 = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad (2.38)$$

where a, b, c, d represent the corresponding elements in (2.37). Equation (2.36) can be further expressed using (2.38) as

$$\mathbf{U}_N = \mathbf{U}_2 \otimes \mathbf{U}_{N/2} = \left[\begin{array}{c|c} a\mathbf{U}_{N/2} & b\mathbf{U}_{N/2} \\ \hline c\mathbf{U}_{N/2} & d\mathbf{U}_{N/2} \end{array} \right] \quad (2.39)$$

where $N = 2^n$. By factorizing the above matrix, we will have

$$\mathbf{U}_N = \begin{bmatrix} a\mathbf{I}_{N/2} & b\mathbf{I}_{N/2} \\ c\mathbf{I}_{N/2} & d\mathbf{I}_{N/2} \end{bmatrix} \begin{bmatrix} \mathbf{U}_{N/2} & \\ & \mathbf{U}_{N/2} \end{bmatrix}. \quad (2.40)$$

Subsequently, $\mathbf{U}_{N/2}$ can be further reduced to $\mathbf{U}_{N/4}$, that is,

$$\mathbf{U}_N = \begin{bmatrix} a\mathbf{I}_{N/2} & b\mathbf{I}_{N/2} \\ c\mathbf{I}_{N/2} & d\mathbf{I}_{N/2} \end{bmatrix} \cdot \begin{bmatrix} a\mathbf{I}_{N/4} & b\mathbf{I}_{N/4} & & \\ c\mathbf{I}_{N/4} & d\mathbf{U}_{N/4} & & \\ & & a\mathbf{I}_{N/4} & b\mathbf{U}_{N/4} \\ & & c\mathbf{I}_{N/4} & d\mathbf{U}_{N/4} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{U}_{N/4} & & & \\ & \mathbf{U}_{N/4} & & \\ & & \mathbf{U}_{N/4} & \\ & & & \mathbf{U}_{N/4} \end{bmatrix} \quad (2.41)$$

which can be written in terms of the Kronecker products as

$$\mathbf{U}_N = (\mathbf{U}_2 \otimes \mathbf{I}_{N/2})(\mathbf{I}_2 \otimes \mathbf{U}_2 \otimes \mathbf{I}_{N/4})(\mathbf{I}_{2^2} \otimes \mathbf{U}_{N/4}). \quad (2.42)$$

Continuing this factorization procedure, finally we can have

$$\mathbf{U}_N = \mathbf{D}_0 \mathbf{D}_1 \cdots \mathbf{D}_{n-1} \quad (2.43)$$

where

$$\begin{aligned} \mathbf{D}_0 &= \overbrace{\mathbf{U}_2 \otimes \mathbf{I}_2 \otimes \cdots \otimes \mathbf{I}_2 \otimes \mathbf{I}_2}^n \\ \mathbf{D}_1 &= \mathbf{I}_2 \otimes \mathbf{U}_2 \otimes \cdots \otimes \mathbf{I}_2 \otimes \mathbf{I}_2 \\ &\vdots \\ \mathbf{D}_{n-1} &= \mathbf{I}_2 \otimes \mathbf{I}_2 \otimes \cdots \otimes \mathbf{I}_2 \otimes \mathbf{U}_2 \end{aligned} \quad (2.44)$$

since

$$\mathbf{I}_{2^{n-r}} = \overbrace{\mathbf{I}_2 \otimes \mathbf{I}_2 \otimes \cdots \otimes \mathbf{I}_2}^{n-r}, \quad 1 \leq r \leq n-1. \quad (2.45)$$

Substituting (2.43) into (2.31) will lead to the fast forward UCHT algorithm. Hence, to compute an N -point UCHT, the fast algorithm decomposes the UCHT transformation matrix into a product of n sparse matrices and the computational complexity can be reduced to $O(N \log_2 N)$.

2.5 Conclusion

In this chapter, some of the well-known Hadamard transforms, particularly, the Walsh Hadamard transform and the BIFORE transform are reviewed. The BIFORE transform is based on the Walsh functions which are the approximations of the sinusoidal waves in the DFT. The transformation matrices describing the elements of the BIFORE transform are the Hadamard matrices. Hence, several well-known methods to develop the fast algorithms for this type of transform are available. The fast BIFORE transform algorithm using the matrix factorization approach is presented in that chapter. Certain orderings of the real Hadamard transforms are mentioned in detail in Section 2.3.3. The Rademacher functions are also discussed and it has been found that the SOWT matrix is obtained from a product series of the sampled Rademacher functions.

Section 2.4 reviews a more generalized form of Hadamard transforms, which are called complex Hadamard transforms. Like their real integer-valued counterparts, they are also orthogonal transforms except that the transformation matrices consists of the entries which are strictly confined to four complex integer values $\{\pm 1, \pm j\}$. They are more suitable to process the problems with complex valued functions. The complex BIFORE transform and the UCHTs are discussed together with their constructions, properties and their fast algorithms. The CBT is constructed through a recursive formula defining one class of complex Hadamard matrices, whereas the UCHTs can be obtained from the WHT matrix and the direct matrix operator. Some of their

applications are also described. But the existing complex Hadamard transforms do not exhibit certain orderings like in real Hadamard transforms.

Chapter 3

Complex Hadamard Transforms with Sequency and Natural Orderings

3.1 Introduction

As mentioned in Chapter 2, the row vectors of real Hadamard matrices can be arranged in a number of ways to obtain the ordered Hadamard matrices. Among the different orderings, sequency order is mostly preferred due to its analogy to frequency in the Fourier transform and its potential to be employed in digital signal processing and communications [7]. Sequency measures how fast the elements of a specific row vector of the Hadamard transform matrix vary over a normalized time base $t \in [0, 1)$. Hence, it is also interesting to explore the orderings of complex Hadamard transforms for some particular applications. In this chapter, complex Hadamard transform with sequency ordering known as sequency-ordered complex Hadamard transform (SCHT) is presented. The construction of the SCHT matrices is based on the complex Rademacher functions which are the extended complex version of conventional Rademacher func-

tions. The row vectors of an SCHAT matrix are arranged in an increasing order of zero crossings in the unit circle of a complex plane within the normalized time t , that is, $0 \leq t \leq 1$. As such the SCHAT may have certain significance in signal processing as the discrete Fourier transform (DFT). Complex Hadamard transform with natural ordering called natural-ordered complex Hadamard transform (NCHAT) is also introduced in this chapter. The NCHAT may be seen as another ordered version of complex Hadamard transform where its ordering follows the WHT matrix. The NCHAT is related to the SCHAT in some manner.

This chapter is organized as follows. Section 3.2 describes the SCHAT in detail followed by the NCHAT in Section 3.3. Section 3.4 concludes this chapter.

3.2 Sequency-Ordered Complex Hadamard Transform (SCHAT)

Firstly, we shall present the construction of the SCHAT matrices followed by their properties, fast algorithms and relationship with other transforms together with the discussion of some potential applications. Before going directly to the generation of the SCHAT, we first introduce the complex Rademacher functions which are extended from the conventional real Rademacher functions, and their matrices.

3.2.1 Complex Rademacher Functions and their Matrices

As described in Section 2.3.4, the Rademacher functions are an incomplete orthogonal set of functions having odd symmetry and they are being used to form other function series which exhibit either odd or even symmetry or both. In this subsection, we extend the real Rademacher functions and define complex Rademacher functions (CRAD)

over a normalized time base $0 \leq t \leq 1$ as follows:

$$\text{CRAD}(0, t) = \begin{cases} 1 & , \quad t \in [0, \frac{1}{4}) \\ j & , \quad t \in [\frac{1}{4}, \frac{1}{2}) \\ -1 & , \quad t \in [\frac{1}{2}, \frac{3}{4}) \\ -j & , \quad t \in [\frac{3}{4}, 1) \end{cases} \quad (3.1)$$

and

$$\text{CRAD}(0, t+1) = \text{CRAD}(0, t) \quad (3.2)$$

where period $T = 1$. Another simpler approach to define complex Rademacher functions over a normalized time base $t \in [0, 1)$ is as follows:

$$\text{CRAD}(0, t) = e^{-j\pi/4} \frac{\text{csign}(e^{j2\pi t})}{\sqrt{2}} \quad (3.3)$$

which maps the real-valued t to the element from the set $\{\pm 1, \pm j\}$ where $\text{csign}(x)$ is defined for any nonzero complex number $x = x_1 + jx_2$ as follows:

$$\text{csign}(x) = \text{sign}(x_1) + j\text{sign}(x_2). \quad (3.4)$$

where $\text{sign}(\cdot)$ represents the *signum* function. But if x is a real number, it can be shown that $\text{csign}(x) = \text{sign}(x)$. The definition of *signum* function is given in (2.16).

Half of the elements of the real Rademacher functions are introduced by the complex element j where $j = \sqrt{-1}$. This holds the orthogonality property of the Rademacher functions. For non-negative integer r , the complex Rademacher function series can be defined as follows:

$$\text{CRAD}(r, t) = \text{CRAD}(0, 2^r t) \quad (3.5)$$

where $r = 0, 1, 2, \dots$. This mean that $\text{CRAD}(r, t)$ can be obtained by compressing $\text{CRAD}(0, t)$ in the horizontal direction by a factor of 2^r .

The complex Rademacher matrices are the discrete versions of the complex Rademacher function series. By sampling the complex Rademacher function series at $N = 2^n$ equally spaced locations within the normalized time t , complex Rademacher matrices can be obtained. Let $\mathbf{R}_n$ be the complex Rademacher matrix of dimension $n \times 2^n$. Then, the r th row of $\mathbf{R}_n$ is defined as

$$\mathbf{R}_n(r, k) = \text{CRAD}\left(r, \frac{k}{2^n} + \frac{1}{2^{n+2}}\right) = \text{CRAD}\left(r, \frac{4k+1}{2^{n+2}}\right) \quad (3.6)$$

where $r = 0, 1, \dots, n-1$ and $k = 0, 1, \dots, 2^n - 1$. Let us, as an example, consider $N = 8$. Then, the 3×8 complex Rademacher matrix is

$$\mathbf{R}_3 = \begin{bmatrix} 1 & 1 & j & j & -1 & -1 & -j & -j \\ 1 & j & -1 & -j & 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}. \quad (3.7)$$

Orthogonal Property: The row vectors of the complex Rademacher matrices are orthogonal to each other in the complex domain as

$$\sum_{k=0}^{N-1} \mathbf{R}_n(r, k) \overline{\mathbf{R}_n(p, k)} = \begin{cases} N, & \text{for } r = p \\ 0, & \text{for } r \neq p \end{cases} \quad (3.8)$$

where $N = 2^n$, $\overline{\mathbf{R}_n(p, k)}$ is the complex conjugate of $\mathbf{R}_n(p, k)$, $k = 0, 1, 2, \dots, N-1$ and $r, p = 0, 1, 2, \dots, n-1$.

Like the real Rademacher functions, the complex Rademacher functions are also an incomplete and orthogonal set of functions in the complex domain. But they are useful to construct other complete set of function series, which exhibits both even and

odd symmetries.

3.2.2 SCHT Matrices

With the complex Rademacher matrices defined, the SCHT matrices are generated based on the products of the row vectors of complex Rademacher matrices. Let $\mathbf{H}_N$ be the SCHT matrix of size $N \times N$ where $N = 2^n$. Then, it is defined as

$$\mathbf{H}_N(m, k) = \prod_{r=0}^{n-1} \mathbf{R}_n(r, k)^{b_r} \quad (3.9)$$

where $\mathbf{R}_n(r, k)$ is the $(r\text{th}, k\text{th})$ element of the complex Rademacher matrix,

$$m = b_{n-1}2^{n-1} + \dots + b_12^1 + b_02^0 \quad (3.10)$$

and $b_r = 0$ or 1 . Let $\mathbf{R}_n(r)$, for $r = 0, 1, \dots, n-1$, be the $r\text{th}$ row vector of the complex Rademacher matrix, and also, let $\odot$ be the operator for the element by element vector multiplication. For example, $\mathbf{H}_8$ can be expressed using (3.9) as

$$\mathbf{H}_8(m, k) = \begin{bmatrix} \mathbf{1} \\ \mathbf{R}_3(0) \\ \mathbf{R}_3(1) \\ \mathbf{R}_3(1) \odot \mathbf{R}_3(0) \\ \mathbf{R}_3(2) \\ \mathbf{R}_3(2) \odot \mathbf{R}_3(0) \\ \mathbf{R}_3(2) \odot \mathbf{R}_3(1) \\ \mathbf{R}_3(2) \odot \mathbf{R}_3(1) \odot \mathbf{R}_3(0) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & j & j & -1 & -1 & -j & -j \\ 1 & j & -1 & -j & 1 & j & -1 & -j \\ 1 & j & -j & 1 & -1 & -j & j & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & j & -j & -1 & 1 & -j & j \\ 1 & -j & -1 & j & 1 & -j & -1 & j \\ 1 & -j & -j & -1 & -1 & j & j & 1 \end{bmatrix}. \quad (3.11)$$

The row index values 2, 1 and 0 refer to the ones found in binary bit positions of the decimal value m . The transformation matrices describing the entries of the SCHT are

the SCHAT matrices.

3.2.3 Exponential Form of the SCHAT Matrix

This subsection will present the construction of the SCHAT matrix using the exponential formula. This form of generation is useful in considering the hardware implementation because this direct implementation of the mathematical expression can be made by using the logic circuits. Besides, it is also convenient to derive other properties such as correlation parameters of the SCHAT sequences in direct sequence (DS) CDMA communication system. Let $h(p, k)$ be the (p th, k th) element of any SCHAT matrix of size $N \times N$ where $N = 2^n$, then it is an element from a set $C_4 = \{\pm 1, \pm j\}$ defined by

$$h(p, k) = (-1)^{\sum_{r=0}^{n-1} \tilde{p}_r k_r} (j)^{\sum_{r=0}^{n-1} \hat{p}_r k_r} \quad (3.12)$$

where $p, k = 0, 1, \dots, N - 1$, $n = \log_2 N$ and p_r, k_r are the binary representations of integers p and k respectively where $r = 0, 1, \dots, n - 1$ represents the indices of the binary bit positions. We also define $\tilde{p}_r$ to be the binary code of p_r in a bit-reversed order and $\hat{p}_r$ to be the binary representation of $(p \bmod (N/2))$ in a bit-reversed order (expressed in $n - 1$ bits). For example, we consider $N = 8$, hence, $n = \log_2 8 = 3$. The binary representations for $\tilde{p}_r$ and $\hat{p}_r$ are shown in Table 3.1. The detailed generation (element by element) for the SCHAT matrix of dimension 8×8 is illustrated in Tables 3.2, 3.3 and 3.4. The resultant matrix in Table 3.4 is an 8×8 SCHAT matrix.

p	Binary	Bit Reversal, $\tilde{p}_r$	Binary of $(p \bmod N/2)$ in $(n-1)$ bits	Bit Reversal	$\hat{p}_r$
0	000	000	00	00	000
1	001	100	01	10	010
2	010	010	10	01	001
3	011	110	11	11	011
4	100	001	00	00	000
5	101	101	01	10	010
6	110	011	10	01	001
7	111	111	11	11	011

Table 3.1: Binary Notations for $\tilde{p}_r$ and $\hat{p}_r$.

		k_r							
		000	001	010	011	100	101	110	111
$\tilde{p}_r$	000	1	1	1	1	1	1	1	1
	100	1	1	1	1	-1	-1	-1	-1
	010	1	1	-1	-1	1	1	-1	-1
	110	1	1	-1	-1	-1	-1	1	1
	001	1	-1	1	-1	1	-1	1	-1
	101	1	-1	1	-1	-1	1	-1	1
	011	1	-1	-1	1	1	-1	-1	1
	111	1	-1	-1	1	-1	1	1	-1

Table 3.2: Elements for $(-1)^{\sum_{r=0}^{n-1} \tilde{p}_r k_r}$.

		k_r							
		000	001	010	011	100	101	110	111
$\hat{p}_r$	000	1	1	1	1	1	1	1	1
	010	1	1	j	j	1	1	j	j
	001	1	j	1	j	1	j	1	j
	011	1	j	j	-1	1	j	j	-1
	000	1	1	1	1	1	1	1	1
	010	1	1	j	j	1	1	j	j
	001	1	j	1	j	1	j	1	j
	011	1	j	j	-1	1	j	j	-1

Table 3.3: Elements for $(j)^{\sum_{r=0}^{n-1} \hat{p}_r k_r}$.

		k_r							
		000	001	010	011	100	101	110	111
$\hat{p}_r$	000	1	1	1	1	1	1	1	1
	010	1	1	j	j	-1	-1	$-j$	$-j$
	001	1	j	-1	$-j$	1	j	-1	$-j$
	011	1	j	$-j$	1	-1	$-j$	j	-1
	000	1	-1	1	-1	1	-1	1	-1
	010	1	-1	j	$-j$	-1	1	$-j$	j
	001	1	$-j$	-1	j	1	$-j$	-1	j
	011	1	$-j$	$-j$	-1	-1	j	j	1

Table 3.4: Illustration for the elements of $(-1)^{\sum_{r=0}^{n-1} \tilde{p}_r k_r} \cdot (j)^{\sum_{r=0}^{n-1} \hat{p}_r k_r}$.

The smaller dimensions using (3.12) are

$$\mathbf{H}_1 = [1], \mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

and

$$\mathbf{H}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}.$$

Therefore, the smallest dimension of the SCHAT matrix is 4×4 as $\mathbf{H}_2$ is similar to the 2×2 WHT matrix. The complex conjugate of an SCHAT matrix can be generated easily by changing j to $-j$ in the previous formula stated in (3.12).

3.2.4 Properties of the SCHAT

In this subsection, we shall present the properties of the SCHAT.

Property 1: Unitary

Let $\mathbf{H}_N$ be any SCHAT matrix of order $N = 2^n$ and $h(p, k)$ be the element of $\mathbf{H}_N$ at row p and column k . Then, it is a square and symmetric matrix which is said to be orthogonal in the complex domain as

$$\sum_{k=0}^{N-1} h(p, k) \overline{h(q, k)} = \begin{cases} N, & \text{for } p = q \\ 0, & \text{for } p \neq q \end{cases} \quad (3.13)$$

where $p, q = 0, 1, \dots, N-1$ and $\overline{\{\}} \}$ denotes the complex conjugate operation.

Proof: As described in Section 3.2.3, any element $h(p, k)$ of an SCHAT matrix can

be expressed in the exponential form as follows:

$$h(p_{n-1}, p_{n-2}, \dots, p_0; k_{n-1}, k_{n-2}, \dots, k_0) = \prod_{r=0}^{n-1} (-1)^{\tilde{p}_r k_r} (j)^{\hat{p}_r k_r} \quad (3.14)$$

where $(p_{n-1}, p_{n-2}, \dots, p_0)$ and $(k_{n-1}, k_{n-2}, \dots, k_0)$ are the row and column indices of $\mathbf{H}_N$ expressed in binary notations. The sum of the products of any two row vectors of $\mathbf{H}_N$ can be given as the binary summation as follows:

$$\begin{aligned} \sum_{k=0}^{N-1} h(p, k) \overline{h(q, k)} &= \sum_{k_{n-1}=0}^1 \sum_{k_{n-2}=0}^1 \cdots \sum_{k_0=0}^1 h(p_{n-1}, p_{n-2}, \dots, p_0; k_{n-1}, k_{n-2}, \dots, k_0) \\ &\quad \times \overline{h(q_{n-1}, q_{n-2}, \dots, q_0; k_{n-1}, k_{n-2}, \dots, k_0)}. \end{aligned} \quad (3.15)$$

Substituting (3.14) into (3.15), we have

$$\sum_{k=0}^{N-1} h(p, k) \overline{h(q, k)} = \sum_{k_{n-1}=0}^1 \sum_{k_{n-2}=0}^1 \cdots \sum_{k_0=0}^1 \prod_{r=0}^{n-1} (-1)^{(\tilde{p}_r + \tilde{q}_r)k_r} (j)^{\hat{p}_r k_r} (-j)^{\hat{q}_r k_r}. \quad (3.16)$$

The sum of products can be written as the product of sums. Therefore,

$$\sum_{k=0}^{N-1} h(p, k) \overline{h(q, k)} = \prod_{r=0}^{n-1} \sum_{k_r=0}^1 (-1)^{(\tilde{p}_r + \tilde{q}_r)k_r} (j)^{\hat{p}_r k_r} (-j)^{\hat{q}_r k_r}. \quad (3.17)$$

But if $k_r = 0$, then $(-1)^{(\tilde{p}_r + \tilde{q}_r)k_r} (j)^{\hat{p}_r k_r} (-j)^{\hat{q}_r k_r} = 1$. Hence,

$$\sum_{k=0}^{N-1} h(p, k) \overline{h(q, k)} = \prod_{r=0}^{n-1} \{1 + (-1)^{(\tilde{p}_r + \tilde{q}_r)} (j)^{\hat{p}_r} (-j)^{\hat{q}_r}\}. \quad (3.18)$$

If we consider the case that is $p = q$, then $p_r = q_r$. But the values of p_r and q_r only can be zero or one. Then, it becomes

$$\prod_{r=0}^{n-1} (1 + 1) = 2^n = N. \quad (3.19)$$

If $p \neq q$, whenever $\hat{p}_r = \hat{q}_r$, there is at least one term in (3.18) giving the zero sum (i.e., $\tilde{p}_r \neq \tilde{q}_r$). It will lead to a zero product. Therefore, (3.13) is proved. The proof of the unitary property for the SCHAT matrices is completed here.

Property 2: Sequency Property

The SCHAT matrices show the form of order that is sequency ordering. The row vectors of an SCHAT matrix $\mathbf{H}_N$ are arranged in an ascending order of zero crossings in the unit circle of a complex plane. Zero crossings are shown in Figure 3.1. If we imagine that the unit vector corresponding to each element of one row of an SCHAT matrix rotates anti-clockwise in the unit circle. Then, sequency describes the number of times that each row vector of the SCHAT matrix crosses the imaginary axis in the unit circle over a normalized time base $0 \leq t \leq 1$. It can be defined as one half of the number of zero crossings (i.e., crossings of imaginary axis). For example, sequency for the p th row of the SCHAT matrix is p as the number of zero crossings is $2p$.

Figure 3.1: Zero crossings of sequency in the unit circle of a complex plane.

Sequency is analogous to frequency of the DFT in which the Fourier components are arranged in an increasing harmonic number in terms of cosine and sine waves. Although the phase shifts are not the same each time a row vector of an SCHAT matrix rotates in the unit circle of a complex plane, the total phase shifts of the row

vectors of the SCHAT matrix are the same as that of the DFT matrix having the same size. In fact, with such property, SCHAT can be used as a tool for signal analysis and synthesis.

Property 3: Symmetry

Let $\mathbf{H}_N$ be any SCHAT matrix of order $N = 2^n$. Then $\mathbf{H}_N$ is a square and symmetrical matrix, that is,

$$\mathbf{H}_N = \mathbf{H}_N^T \quad (3.20)$$

where $\mathbf{H}_N^T$ is the transpose matrix of $\mathbf{H}_N$. The inverse of the SCHAT matrix, $\mathbf{H}_N^{-1}$, is related as its complex conjugate transpose matrix, $\mathbf{H}_N^*$, as follows:

$$\mathbf{H}_N^{-1} = \frac{\mathbf{H}_N^*}{N}. \quad (3.21)$$

This property enables us to implement the complex Hadamard transform more easily as both the forward transform and the inverse transform can share the same hardware architecture.

Definition 1: The N -point SCHAT of a signal vector $\mathbf{x}_N = [x(0), x(1), \dots, x(N-1)]^T$ is defined as

$$\mathbf{X}_N = \frac{1}{N} \mathbf{H}_N^* \mathbf{x}_N \quad (3.22)$$

where N is a power of two, $\mathbf{X}_N = [X(0), X(1), \dots, X(N-1)]^T$ is the transformed column vector and $*$ represents the complex conjugate transpose operation. The values of $\mathbf{X}_N$ are the complex numbers. Since, it is an orthogonal transform, the data sequence can be recovered uniquely from the inverse transform, that is,

$$\mathbf{x}_N = \mathbf{H}_N \mathbf{X}_N. \quad (3.23)$$

Property 4: Linearity

Let $x_1(k)$ and $x_2(k)$ be two data vectors of the length $N = 2^n$ where $k = 0, 1, \dots, N - 1$. They are assumed to have the same length. Then, SCHT is a linear transformation that can be expressed as

$$\text{SCHT}[ax_1(k) + bx_2(k)] = a\text{SCHT}[x_1(k)] + b\text{SCHT}[x_2(k)] \quad (3.24)$$

where a and b are constants. Instead of transforming the two sequences individually, the two sequences can be combined together before taking the transform. It will help to reduce the computational time as this technique requires one SCHT transformation in contrast to the use of two N -point SCHTs to compute $X_1(k)$ and $X_2(k)$ individually and combine them.

3.2.5 Fast Algorithm for the SCHT

The fast algorithm is necessary for every type of transformations to compute the transform efficiently due to heavy computational load for direct computation in practice. In this subsection, the fast SCHT algorithm using the sparse matrix factorization approach is derived. Factorization of the SCHT transform matrix into a number of sparse matrices contributes directly to the speed and efficiency of the computation. Since SCHT is a complex Hadamard transform which is composed of the elements $\{\pm 1, \pm j\}$, the transform computation involves only additions and subtractions. Multiplications by the elements j with complex input can be performed by swapping and negating operations (which is to be explained in Chapter 4). For the real signals, $N(N - 1)$ real addition/subtractions are required to perform the SCHT without using the fast algorithm. The inverse SCHT transform matrix in (3.23) is the same as the SCHT matrix itself so we start with the factorizing of the SCHT matrix for the fast inverse SCHT algorithm.

From (3.23), the inverse SCHT of an N -point data sequence is $\mathbf{x}_N = \mathbf{H}_N \mathbf{X}_N$. Let the column vector $\mathbf{X}_N$ be divided into two vectors, the one with the entries having even indices, $\mathbf{X}_e$, and the other with odd indices, $\mathbf{X}_o$. That is, $\mathbf{X}_e = [X(0), X(2), \dots, X(N-2)]^T$ and $\mathbf{X}_o = [X(1), X(3), \dots, X(N-1)]^T$ where $N = 2^n$. Then, the inverse transform can be written by partitioning the matrix as follows.

$$\mathbf{x}_N = \begin{bmatrix} \mathbf{H}_{N/2} & \mathbf{S}_{2^{n-1}} \mathbf{H}_{N/2} \\ \mathbf{H}_{N/2} & -\mathbf{S}_{2^{n-1}} \mathbf{H}_{N/2} \end{bmatrix} \begin{bmatrix} \mathbf{X}_e \\ \mathbf{X}_o \end{bmatrix} \quad (3.25)$$

where $\mathbf{x}_N = [x(0), x(1), x(2), \dots, x(N-1)]^T$,

$$\mathbf{S}_{2^{n-1}} = \begin{bmatrix} \mathbf{I}_{2^{n-2}} & \mathbf{0} \\ \mathbf{0} & j\mathbf{I}_{2^{n-2}} \end{bmatrix} \quad (3.26)$$

and $\mathbf{I}_{2^{n-2}}$ is the identity matrix of dimension $2^{n-2} \times 2^{n-2}$. From (3.25), we can factorize $\mathbf{x}_N$ as

$$\mathbf{x}_N = \begin{bmatrix} \mathbf{I}_{2^{n-1}} & \mathbf{I}_{2^{n-1}} \\ \mathbf{I}_{2^{n-1}} & -\mathbf{I}_{2^{n-1}} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{2^{n-1}} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{2^{n-1}} \end{bmatrix} \begin{bmatrix} \mathbf{H}_{N/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_{N/2} \end{bmatrix} \begin{bmatrix} \mathbf{X}_e \\ \mathbf{X}_o \end{bmatrix} \quad (3.27)$$

which can be written in the compact form as

$$\mathbf{x}_N = \mathbf{B}_n (\mathbf{I}_{2^{n-1}} \oplus \mathbf{S}_{2^{n-1}}) (\mathbf{H}_{N/2} \oplus \mathbf{H}_{N/2}) \begin{bmatrix} \mathbf{X}_e \\ \mathbf{X}_o \end{bmatrix} \quad (3.28)$$

where $\oplus$ is the direct sum operator [17] and

$$\mathbf{B}_n = \begin{bmatrix} \mathbf{I}_{2^{n-1}} & \mathbf{I}_{2^{n-1}} \\ \mathbf{I}_{2^{n-1}} & -\mathbf{I}_{2^{n-1}} \end{bmatrix}.$$

Now we define

$$\mathbf{C}_{n-k} = \begin{bmatrix} \mathbf{I}_{2^{n-k}} & \mathbf{I}_{2^{n-k}} \\ \mathbf{I}_{2^{n-k}} & -\mathbf{I}_{2^{n-k}} \end{bmatrix}, \quad 1 \leq k \leq n \quad (3.29)$$

and

$$\begin{aligned} \mathbf{B}_n &= \mathbf{C}_{n-1} \\ \mathbf{B}_{n-1} &= \mathbf{C}_{n-2} \oplus \mathbf{C}_{n-2} \\ &\vdots \\ \mathbf{B}_1 &= \mathbf{C}_0 \oplus \mathbf{C}_0 \oplus \cdots \oplus \mathbf{C}_0. \end{aligned} \quad (3.30)$$

From (3.27), knowing that $\mathbf{H}_{N/2}$ is the SHT matrix of size $\frac{N}{2} \times \frac{N}{2}$, we can take the inverse SHT of the data vectors, $\mathbf{X}_e$ and $\mathbf{X}_o$, separately. That is,

$$\mathbf{H}_{N/2} \mathbf{X}_e = \mathbf{C}_{n-2}(\mathbf{I}_{2^{n-2}} \oplus \mathbf{S}_{2^{n-2}})(\mathbf{H}_{N/4} \oplus \mathbf{H}_{N/4}) \begin{bmatrix} \mathbf{X}_{ee} \\ \mathbf{X}_{eo} \end{bmatrix} \quad (3.31)$$

and

$$\mathbf{H}_{N/2} \mathbf{X}_o = \mathbf{C}_{n-2}(\mathbf{I}_{2^{n-2}} \oplus \mathbf{S}_{2^{n-2}})(\mathbf{H}_{N/4} \oplus \mathbf{H}_{N/4}) \begin{bmatrix} \mathbf{X}_{oe} \\ \mathbf{X}_{oo} \end{bmatrix} \quad (3.32)$$

where

$$\begin{aligned} \mathbf{X}_{ee} &= [X(0), X(4), X(8), \dots, X(N-4)]^T \\ \mathbf{X}_{eo} &= [X(2), X(6), X(10), \dots, X(N-2)]^T \\ \mathbf{X}_{oe} &= [X(1), X(5), X(9), \dots, X(N-3)]^T \\ \mathbf{X}_{oo} &= [X(3), X(7), X(11), \dots, X(N-1)]^T. \end{aligned} \quad (3.33)$$

The above two equations (3.31) and (3.32) can be combined in the compact form as

follows:

$$\begin{bmatrix} \mathbf{X}_e \\ \mathbf{X}_o \end{bmatrix} = \mathbf{B}_{n-1}(\mathbf{I}_{2^{n-2}} \oplus \mathbf{S}_{2^{n-2}} \oplus \mathbf{I}_{2^{n-2}} \oplus \mathbf{S}_{2^{n-2}})(\mathbf{H}_{N/4} \oplus \mathbf{H}_{N/4} \oplus \mathbf{H}_{N/4} \oplus \mathbf{H}_{N/4}) \begin{bmatrix} \mathbf{X}_{ee} \\ \mathbf{X}_{eo} \\ \mathbf{X}_{oe} \\ \mathbf{X}_{oo} \end{bmatrix} \quad (3.34)$$

Substituting (3.34) into (3.28), we have

$$\begin{aligned} \mathbf{x}_N &= \mathbf{B}_n(\mathbf{I}_{2^{n-1}} \oplus \mathbf{S}_{2^{n-1}})\mathbf{B}_{n-1}(\mathbf{I}_{2^{n-2}} \oplus \mathbf{S}_{2^{n-2}} \oplus \mathbf{I}_{2^{n-2}} \oplus \mathbf{S}_{2^{n-2}}) \\ &\quad \cdot (\mathbf{H}_{N/4} \oplus \mathbf{H}_{N/4} \oplus \mathbf{H}_{N/4} \oplus \mathbf{H}_{N/4}) \begin{bmatrix} \mathbf{X}_{ee} \\ \mathbf{X}_{eo} \\ \mathbf{X}_{oe} \\ \mathbf{X}_{oo} \end{bmatrix}. \end{aligned} \quad (3.35)$$

We can continue this factorization procedure until the SCHAT matrix is reduced to the dimension of 2×2 which involves no more multiplications by the elements j since $\mathbf{H}_2$ is an 2×2 WHT matrix. Finally, this will result in

$$\mathbf{x}_N = (\mathbf{B}_n \mathbf{H}_{11})(\mathbf{B}_{n-1} \mathbf{H}_{22})(\mathbf{B}_{n-2} \mathbf{H}_{33}) \cdots (\mathbf{B}_2 \mathbf{H}_{n-1,n-1}) \mathbf{B}_1 (\mathbf{P}_b \mathbf{x}_N) \quad (3.36)$$

where

$$\begin{aligned} \mathbf{H}_{11} &= (\mathbf{I}_{2^{n-1}} \oplus \mathbf{S}_{2^{n-1}}) \\ \mathbf{H}_{22} &= (\mathbf{I}_{2^{n-2}} \oplus \mathbf{S}_{2^{n-2}} \oplus \mathbf{I}_{2^{n-2}} \oplus \mathbf{S}_{2^{n-2}}) \\ &\vdots \\ \mathbf{H}_{n-1,n-1} &= (\mathbf{I}_2 \oplus \mathbf{S}_2 \oplus \cdots \oplus \mathbf{I}_2 \oplus \mathbf{S}_2) \end{aligned} \quad (3.37)$$

and $\mathbf{P}_b$ is a bit reversal permutation matrix which shuffles the vector $\mathbf{X}_N$ into a bit-reversed order. Let us, for example, consider $N = 8$. Hence, $n = \log_2 8 = 3$ and (3.36) will become

$$\begin{aligned}\mathbf{x}_8 &= \mathbf{B}_3 \mathbf{H}_{11} \mathbf{B}_2 \mathbf{H}_{22} \mathbf{B}_1 \mathbf{P}_b \mathbf{X}_8 \\ &= \mathbf{B}_3 (\mathbf{I}_4 \oplus \mathbf{S}_4) \mathbf{B}_2 (\mathbf{I}_2 \oplus \mathbf{S}_2 \oplus \mathbf{I}_2 \oplus \mathbf{S}_2) (\mathbf{H}_2 \oplus \mathbf{H}_2 \oplus \mathbf{H}_2 \oplus \mathbf{H}_2) \mathbf{P}_b \mathbf{X}_8 \quad (3.38)\end{aligned}$$

since $\mathbf{B}_1 = \mathbf{C}_0 \oplus \mathbf{C}_0 \oplus \mathbf{C}_0 \oplus \mathbf{C}_0$ and $\mathbf{C}_0 = \mathbf{H}_2$ which is

$$\mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

The vectors $\mathbf{X}_8$ and $\mathbf{P}_b \mathbf{X}_8$ are given by

$$\begin{aligned}\mathbf{X}_8 &= [X(0), X(1), X(2), X(3), X(4), X(5), X(6), X(7)]^T \\ \mathbf{P}_b \mathbf{X}_8 &= [X(0), X(4), X(2), X(6), X(1), X(5), X(3), X(7)]^T. \quad (3.39)\end{aligned}$$

The factorization in (3.38) will lead to the fast inverse SCHT algorithm for $N = 8$, which decomposition is illustrated by the signal flow graph in Figure 3.2. To get better understanding of becoming the flow graph, (3.38) can be written in the matrix form

as follows:

$$\mathbf{x}_8 = \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{I}_4 & -\mathbf{I}_4 \end{bmatrix} \begin{bmatrix} \mathbf{I}_4 \\ \mathbf{S}_4 \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & \mathbf{I}_2 \\ \mathbf{I}_2 & -\mathbf{I}_2 \\ & \mathbf{I}_2 & \mathbf{I}_2 \\ & \mathbf{I}_2 & -\mathbf{I}_2 \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 \\ \mathbf{S}_2 \\ & \mathbf{I}_2 \\ & & \mathbf{S}_2 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{H}_2 \\ & \mathbf{H}_2 \\ & & \mathbf{H}_2 \\ & & & \mathbf{H}_2 \end{bmatrix} \mathbf{P}_b \mathbf{X}_8 \quad (3.40)$$

where

$$\mathbf{S}_4 = \begin{bmatrix} \mathbf{I}_2 & \mathbf{0} \\ \mathbf{0} & j\mathbf{I}_2 \end{bmatrix}, \quad \mathbf{S}_2 = \begin{bmatrix} 1 & 0 \\ 0 & j \end{bmatrix}$$

and the black spaces in the matrices of the above expression represent that the corresponding elements are zeros. The equation (3.40) is converted to the signal flow graph as shown in Figure 3.2 which consists of 12 butterflies. The last matrix (which consists

Figure 3.2: Flow graph of the final decomposition of fast inverse SCHT algorithm ($N = 8$).

of $\mathbf{H}_2$) in (3.40) corresponds to the first four butterflies in the figure. Likewise, the matrix comprised of eight $\mathbf{I}_2$ forms four butterflies in the middle, and the last four butterflies are created by the matrix with four $\mathbf{I}_4$.

As shown in Figure 3.2, an 8-point SCHAT can be decomposed into three stages, and each stage needs eight addition/subtractions. In each stage, two complex multiplications by j are required except for the first stage and four complex multiplications are required to perform the 8-point SCHAT operation. Hence, it is easy to conclude that for an N -point SCHAT where $N = 2^n$, the fast algorithm decomposes into $n = \log_2 N$ stages and the total computational requirements are $N \log_2 N$ complex addition/subtractions and $(N/4) \log_2(N/2)$ complex multiplications for the trivial twiddle factors j . In fact, the trivial complex multiplications can be performed by using the swapping and negating operations in the hardware implementation of the transform. Since, the SCHAT transformation matrices are symmetric, the forward SCHAT has the same signal flow graph and computational operations as that of the inverse SCHAT. But in contrast to the latter, the former needs the conjugate of the transform matrix as shown in (3.22) and the scaling factor of $(1/N)$ which can be easily computed by the binary shifting operations without requiring any arithmetic operations.

The WHT matrices following the frequencies of sinusoidal waves in the DFT are referred to as the sequency-ordered Walsh transform (SOWT) matrices as described in Subsection 2.3.5 of Chapter 2. They can be obtained by rearranging the row vectors of a WHT matrix through a Gray code and bit reversal operation [7]. While the Fourier bases are sinusoids with harmonic frequencies, the SOWT bases are the Walsh functions. Actually, SOWT is an approximation of sine and cosine waves of the DFT by using the rectangular waves. It can be realized that SCHAT is an extended complex version of the SOWT and it is another close approximation to the sinusoidal waves in Fourier analysis. Since SOWT is a sequency-ordered version of the WHT, several

well-known techniques to develop the fast algorithms for this type of transform are available [7]. In general, for an N -point SOWT, the fast algorithm reduces the computational complexity to $O(N \log_2 N)$. This is similar to the computational load of an N -point fast SCHAT except that the latter needs the trivial complex multiplications by factor j , which do not require any arithmetic operations. While both the signal and the transform are restricted to real numbers in the SOWT, the SCHAT is suitable to analyze and process the complex-valued signals.

The fast SCHAT algorithm is very much similar to the radix-2 fast Fourier transform (FFT) algorithm because they have almost the same signal flow graphs. Unlike the radi-2 FFT, however, the former does not need any nontrivial twiddle factor multiplications, which results in significant saving in computational time. Similar to the radix-2 FFT, this fast SCHAT algorithm is suitable to adopt a pipelined hardware structure to efficiently map the arithmetic operations into the hardware (this will be detailed in Chapter 4). Therefore, at the permissible expense of some loss in parameter estimation accuracy, one can replace the DFT with the SCHAT in some potential applications with simpler implementation and consequent savings in computational cost. The following section will suggest some useful applications of the SCHAT.

3.2.6 Applications of the SCHAT

As mentioned in Subsection 3.2.4, the SCHAT matrix exhibits sequency ordering which is analogous to frequency in the DFT. Therefore, the SCHAT may have the potential to be employed in different applications as the DFT. In this subsection, the application of the SCHAT in signal analysis and synthesis will be presented together with the supporting simulation results. Other applications will be suggested and further discussed in the subsequent chapters, Chapter 5 and Chapter 6.

A. Signal Analysis and Synthesis

SCHT is applied in spectrum estimation of a sinusoidal signal as an example. It is expected that the spectrum transformed by the SCHT is close to the DFT spectrum. Figure 3.3 shows the normalized magnitude spectra of 64-point DFT, SCHT and SOWT of a sine wave of 20 Hz using 64 samples per second. It can be seen from the figure that the SCHT magnitude spectrum is closely matched with that of the DFT with a few side lobes as compared to that of the SOWT. The desired peaks occur in the same locations, that is, the frequency indices 20 and 44 as that of the DFT even though there are extra spurious spikes in the SCHT spectrum. Therefore, it is worth considering the SCHT in spectral analysis when it is necessary to achieve significant hardware savings and reduced computational time at the permissible expense of some losses in parameter estimation performance.

Besides, it is also found that the SCHT magnitude spectrum is more invariant to the circular phase shift of input signals in the time domain than that of the SOWT.

Figure 3.3: Magnitude spectra of the DFT, SCHT and SOWT for the sinusoidal wave of 20 Hz.

Figure 3.4: The SCHT and SOWT magnitude spectra with 90° circular phase shift in the time domain.

Figure 3.4 illustrates the SCHT and SOWT normalized magnitude spectra due to 90° circular phase shift of the input signal in the time domain. The circular phase shift in the time domain does not change the SCHT magnitude spectrum, but the change is apparent in the SOWT spectrum as shown in the figure. The SCHT of a real signal generates a series of complex numbers in the transformed domain. Hence, it can be concluded that the effect of the circular phase shift mostly changes the phase angles of these complex numbers in the sequency domain rather than their magnitudes. On the other hand, the circular phase shift affects the magnitudes of the SOWT spectrum as its spectral values are the real numbers. But it should be noted that the SCHT does not possess shift invariant property whereas the DFT magnitude spectrum is shift invariant to the circular shifting of the data sequence.

B. Communications

There has been of great interest to use the orthogonal spreading sequences in direct sequence (DS) CDMA systems over the past few decades due to the spectrum efficiency and low multiple access interference. For example, the WHT sequence is one of the most popular orthogonal sequences, which is derived from the row vector of a WHT matrix. It is pointed out in [28] that the complex sequences can provide a 3-dB performance improvement as compared to the real binary sequences in the maximum periodic correlation parameters. With quadri-phase properties, it is more efficient to use the 4-phase sequences over the real binary sequences when dealing with quadri-phase modulation. Therefore, SCHAT is a suitable candidate to be employed in such applications as the complex spreading sequences. In this thesis, the complex 4-phase sequences which are derived from the row vectors of an SCHAT matrix are proposed in DS-CDMA systems. Their bit-error-rate (BER) performance is examined in both synchronous and asynchronous DS-CDMA systems as compared with other well-know spreading sequences such as the WHT, UCHT, Gold, small set of Kasami and m -sequences. The details will be presented in Chapter 5.

C. Image Watermarking

SCHAT is also suitable for use in the transform domain digital image watermarking due to its sequency ordering. The images can be transformed using two dimensional (2-D) SCHAT into the sequency domain based on the 8×8 image blocks. One obvious advantage is its simplicity to compute the transform with the availability of its fast algorithm. Using the analogy of sequency with frequency in the DFT, the watermark bits are inserted in the lower sequency components of the image to make the watermarking scheme robust to the attacks. Chapter 6 will propose a robust phase watermarking scheme for digital still images based on the SCHAT. The detailed procedures will be discussed in that chapter.

3.2.7 Relationship of the SCHT with the FFT and the UCHT

In [33], it has been shown that the radix-2 FFT can be obtained recursively from the WHT and some matrices with the twiddle factors. In this subsection, we discuss the relations of the SCHT with the radix-2 FFT and the UCHT. Firstly, we define the decimation-in-time radix-2 FFT. The DFT of the data sequence $\{x(m)\}_{m=0}^{N-1}$ of length N where N is a power of 2 is defined as [36]

$$\begin{aligned} \mathbf{X}_N &= \mathbf{F}_N \mathbf{x}_N \\ X(k) &= \sum_{m=0}^{N-1} x(m) W_N^{mk}, \quad k = 0, 1, \dots, N-1 \end{aligned} \quad (3.41)$$

where $\mathbf{X}_N = [X(0), X(1), \dots, X(N-1)]^T$ is the DFT coefficient vector in the frequency domain, $\mathbf{F}_N$ is the DFT matrix of dimension $N \times N$ and $W_N^{mk} = \exp(-j\frac{2\pi mk}{N})$ is the twiddle factor of the DFT.

Let the data vector $\mathbf{x}_N$ be divided into two vectors, the one with elements having odd indices and the other with even indices. Then from (3.41), we have the following FFT form [33, 36], that is,

$$X(k) = \sum_{m=0}^{(N/2)-1} x(2m) W_{N/2}^{mk} + W_N^k \sum_{m=0}^{(N/2)-1} x(2m+1) W_{N/2}^{mk} \quad (3.42)$$

$$X(k + (N/2)) = \sum_{m=0}^{(N/2)-1} x(2m) W_{N/2}^{mk} - W_N^k \sum_{m=0}^{(N/2)-1} x(2m+1) W_{N/2}^{mk} \quad (3.43)$$

where $k = 0, 1, \dots, (N/2) - 1$. In order to express the relationship with the SCHT, (3.42) and (3.43) can be rewritten in the matrix form as follows. [17]

$$\mathbf{X}_N = \begin{bmatrix} \mathbf{I}_{2^{n-1}} & \boldsymbol{\omega}_1 \\ \mathbf{I}_{2^{n-1}} & -\boldsymbol{\omega}_1 \end{bmatrix} \begin{bmatrix} \mathbf{F}_{N/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_{N/2} \end{bmatrix} \begin{bmatrix} \mathbf{x}_e \\ \mathbf{x}_o \end{bmatrix} \quad (3.44)$$

where $\boldsymbol{\omega}_1 = \text{diag}\{1, W, W^2, \dots, W^{(N/2)-1}\}$ is the diagonal matrix which consists of the twiddle factors, $\mathbf{x}_e = [x(0), x(2), \dots, x(N-2)]^T$, $\mathbf{x}_o = [x(1), x(3), \dots, x(N-1)]^T$ and $\mathbf{F}_{N/2}$ is the DFT matrix of dimension $(N/2) \times (N/2)$. From (3.44), $\mathbf{F}_{N/2}$ can be further factorized and in general, the radix-2 FFT can be written as

$$\mathbf{F}_N = (\mathbf{I}_{2^0} \otimes \boldsymbol{\Omega}_{n-1})(\mathbf{I}_{2^1} \otimes \boldsymbol{\Omega}_{n-2}) \cdots (\mathbf{I}_{2^{n-1}} \otimes \boldsymbol{\Omega}_0) \mathbf{P}_b \quad (3.45)$$

where $\otimes$ denotes the right hand side Kronecker product as defined in (2.3),

$$\boldsymbol{\Omega}_k = \begin{bmatrix} \mathbf{I}_{2^k} & \boldsymbol{\omega}_{n-k} \\ \mathbf{I}_{2^k} & -\boldsymbol{\omega}_{n-k} \end{bmatrix}, \quad k = 0, 1, \dots, n-1$$

with

$$\begin{aligned} \boldsymbol{\omega}_1 &= \text{diag}\{1, W^1, W^2, \dots, W^{(N/2)-1}\} \\ \boldsymbol{\omega}_2 &= \text{diag}\{1, W^2, W^4, \dots, W^{2((N/4)-1)}\} \\ \boldsymbol{\omega}_3 &= \text{diag}\{1, W^4, W^8, \dots, W^{4((N/8)-1)}\} \\ &\vdots \\ \boldsymbol{\omega}_{n-1} &= \text{diag}\{1, W^{(N/4)}\} \end{aligned} \quad (3.46)$$

$\boldsymbol{\omega}_n = [1]$ where $\text{diag}\{\cdot\}$ denotes the diagonal matrix, and $\mathbf{P}_b$ is a permutation matrix for the radix-2 FFT, which scrambles the data vector into the bit reversal order. Since,

$$\boldsymbol{\omega}_{n-1} = \begin{bmatrix} 1 & 0 \\ 0 & -j \end{bmatrix},$$

we can express the radix-2 FFT using the SHT matrix as follows.

$$\mathbf{F}_N = \begin{cases} (\mathbf{I}_{2^0} \otimes \boldsymbol{\Omega}_{n-1})(\mathbf{I}_{2^1} \otimes \boldsymbol{\Omega}_{n-2}) \cdots (\mathbf{I}_{2^{n-3}} \otimes \boldsymbol{\Omega}_2)(\mathbf{I}_{2^{n-2}} \otimes \mathbf{H}_{p4})\mathbf{P}_b, & \text{if } N > 4 \\ \mathbf{H}_{p4}\mathbf{P}_b & \text{if } N = 4 \end{cases} \quad (3.47)$$

where $N = 2^n$, $\mathbf{H}_{p4} = \mathbf{H}_4\mathbf{P}$,

$$\mathbf{H}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}$$

is the SHT matrix of dimension 4×4 and

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

is a permutation matrix to permute the columns of $\mathbf{H}_4$. Let us, for instance, consider $N = 8$, hence, $n = 3$. Then (3.47) becomes

$$\begin{aligned} \mathbf{F}_8 &= (\mathbf{I} \otimes \boldsymbol{\Omega}_2)(\mathbf{I}_2 \otimes \mathbf{H}_{p4})\mathbf{P}_b \\ \mathbf{F}_8 &= \begin{bmatrix} \mathbf{I}_4 & \boldsymbol{\omega}_1 \\ \mathbf{I}_4 & -\boldsymbol{\omega}_1 \end{bmatrix} \begin{bmatrix} \mathbf{H}_{p4} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_{p4} \end{bmatrix} \mathbf{P}_b \end{aligned} \quad (3.48)$$

where

$$\mathbf{H}_{p^4} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & -j & j \\ 1 & 1 & -1 & -1 \\ 1 & -1 & j & -j \end{bmatrix}.$$

Hence, the radix-2 FFT can be computed using the SHT matrix with the same computational load by the sparse matrix factorization.

Subsequently, we will mention that the SHT matrix can be obtained from the WHT and UHT matrices. Let $\mathbf{H}_N$ be any SHT matrix of dimension $N \times N$. By using the matrix factorization technique, $\mathbf{H}_N$ can be expressed as

$$\begin{aligned} \mathbf{H}_N &= \mathbf{B}_n (\mathbf{I}_1 \otimes \mathbf{A}_{n-1}) (\mathbf{I}_{2^1} \otimes \mathbf{A}_{n-2}) \cdots (\mathbf{I}_{2^{n-2}} \otimes \mathbf{A}_1) \mathbf{P}_b \\ &= \mathbf{B}_n \left[\prod_{r=0}^{n-2} (\mathbf{I}_{2^r} \otimes \mathbf{A}_{n-r-1}) \right] \mathbf{P}_b \end{aligned} \quad (3.49)$$

where

$$\begin{aligned} \mathbf{B}_n &= \begin{bmatrix} \mathbf{I}_{2^{n-1}} & \mathbf{I}_{2^{n-1}} \\ \mathbf{I}_{2^{n-1}} & -\mathbf{I}_{2^{n-1}} \end{bmatrix}, \\ \mathbf{A}_k &= \begin{bmatrix} \mathbf{I}_{2^{k-1}} & \mathbf{I}_{2^{k-1}} & \mathbf{0} & \mathbf{0} \\ \mathbf{I}_{2^{k-1}} & -\mathbf{I}_{2^{k-1}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{2^{k-1}} & \mathbf{I}_{2^{k-1}} \\ \mathbf{0} & \mathbf{0} & j\mathbf{I}_{2^{k-1}} & -j\mathbf{I}_{2^{k-1}} \end{bmatrix}, \quad n-1 \leq k \leq 1 \end{aligned}$$

and $\mathbf{P}_b$ is the bit reversal permutation matrix. As an example, an 8×8 SHT matrix

for $N = 8$ where $n = \log_2 8 = 3$ is

$$\begin{aligned}
\mathbf{H}_8 &= \mathbf{B}_3 (\mathbf{I}_1 \otimes \mathbf{A}_2) (\mathbf{I}_2 \otimes \mathbf{A}_1) \mathbf{P}_b \\
\mathbf{H}_8 &= \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{I}_4 & -\mathbf{I}_4 \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & \mathbf{I}_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{I}_2 & -\mathbf{I}_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_2 & \mathbf{I}_2 \\ \mathbf{0} & \mathbf{0} & j\mathbf{I}_2 & -j\mathbf{I}_2 \end{bmatrix} \begin{bmatrix} \mathbf{W}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{W}'_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{W}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{W}'_2 \end{bmatrix} \mathbf{P}_b
\end{aligned} \tag{3.50}$$

where

$$\mathbf{W}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \text{ and } \mathbf{W}'_2 = \begin{bmatrix} 1 & 1 \\ j & -j \end{bmatrix}.$$

Therefore, the SCHAT matrix $\mathbf{H}_N$ can be expressed using the WHT and UCHT matrices as

$$\begin{aligned}
\mathbf{H}_N &= \mathbf{B}_n \left(\mathbf{I}_1 \otimes \begin{bmatrix} \mathbf{W}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_2 \end{bmatrix} \otimes \mathbf{I}_{2^{n-2}} \right) \left(\mathbf{I}_{2^1} \otimes \begin{bmatrix} \mathbf{W}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_2 \end{bmatrix} \otimes \mathbf{I}_{2^{n-3}} \right) \cdots \\
&\quad \left(\mathbf{I}_{2^{n-2}} \otimes \begin{bmatrix} \mathbf{W}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_2 \end{bmatrix} \otimes \mathbf{I}_1 \right) \mathbf{P}_b \\
\mathbf{H}_N &= \mathbf{B}_n \left[\prod_{r=0}^{n-2} \left(\mathbf{I}_{2^r} \otimes \begin{bmatrix} \mathbf{W}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_2 \end{bmatrix} \otimes \mathbf{I}_{2^{n-r-2}} \right) \right] \mathbf{P}_b
\end{aligned} \tag{3.51}$$

where $\mathbf{W}_2$ is the 2×2 WHT matrix, and

$$\mathbf{U}_2 = \begin{bmatrix} 1 & 1 \\ j & -j \end{bmatrix}$$

is a 2×2 UCHT matrix.

The SCHAT matrices are constructed based on the complex Rademacher matrices as mentioned in Section 3.2 while the generation of the UCHT is based on the WHT and successive Kronecker products [11]. Even though they are generated differently, the SCHAT can be obtained recursively from the WHT and UCHT matrices of dimension 2×2 , and some sparse matrices which include the identity matrices. Both transforms are suitable for problems with complex-valued functions since they are complex integer-valued transforms. The UCHTs with the half spectrum property are suitable and advantageous for use in the classification and synthesis/analysis of logic circuits [13]. The SCHAT is more favored to be used in spectrum analysis, filtering, image signal processing and communications due to its sequency ordering.

3.2.8 The DFT via the SCHAT

As mentioned in Subsection 3.2.4, the concept of sequency is analogous to frequency where we find in the DFT in which the Fourier components are arranged in increasing harmonic number. By making use of this analogy, we can find the relationship matrix between the DFT and the SCHAT. Let $\mathbf{T}_N$ be the relationship matrix of dimension $N \times N$ where $N = 2^n$. Then we define $\mathbf{T}_N$ as

$$\mathbf{T}_N = \mathbf{F}_N \mathbf{H}_N \tag{3.52}$$

where $\mathbf{H}_N$ is the SHT matrix of dimension $N \times N$ as defined in Subsection 3.2.2 and $\mathbf{F}_N$ is the DFT matrix of size $N \times N$ which is expressed by

$$\mathbf{F}_N = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W & W^2 & \cdots & W^{N-1} \\ 1 & W^2 & W^4 & \cdots & W^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W^{N-1} & W^{2(N-1)} & \cdots & W^{(N-1)(N-1)} \end{bmatrix} \quad (3.53)$$

where $W^{mk} = \exp(-j\frac{2\pi mk}{N})$ is the twiddle factors and $m, k = 0, 1, \dots, N-1$. With the $\mathbf{T}_N$ matrix defined, the DFT coefficients can be computed using the SHT coefficients as follows.

The DFT of the data column vector $\mathbf{x}_N = [x(0), x(1), \dots, x(N-1)]^T$ is given by

$$\mathbf{X}_{1,N} = \mathbf{F}_N \mathbf{x}_N \quad (3.54)$$

where $\mathbf{X}_{1,N} = [X_1(0), X_1(1), \dots, X_1(N-1)]^T$ is the complex DFT coefficient vector. Subsequently, the above equation can be expressed using the $\mathbf{T}_N$ matrix as

$$\mathbf{X}_{1,N} = \frac{1}{N} \mathbf{T}_N \mathbf{H}_N^* \mathbf{x}_N \quad (3.55)$$

where $\{ \}^*$ represents the complex conjugate transpose and $\mathbf{H}_N \mathbf{H}_N^* = N \mathbf{I}_N$. Substituting (3.22) into (3.55), we have

$$\mathbf{X}_{1,N} = \mathbf{T}_N \mathbf{X}_N \quad (3.56)$$

where $\mathbf{X}_N = [X(0), X(1), \dots, X(N-1)]^T$ is the complex SHT coefficient vector as defined in (3.22). As such the DFT coefficients can be obtained from the SHT

coefficients through their relationship matrix $\mathbf{T}_N$. In other words, the signal can be converted from the sequency domain to the frequency domain easily by representing the Fourier coefficients as the linear combination of the SCHAT coefficients. In fact, $\mathbf{T}_N$ can be interpreted as the DFT transformation of each row/column vector of the SCHAT matrix. Let us, for example, consider $N = 8$. Then $\mathbf{T}_8$ is found to be

$$\mathbf{T}_8 = \begin{bmatrix} 8 & & & & & & & \\ & 4 + 4W_8^1 & & & 4 - 4W_8^1 & & & \\ & & 8 & & & & & \\ & & & 4 + 4W_8^1 & & & 4 - 4W_8^1 & \\ & & & & 8 & & & \\ & 4 - 4W_8^1 & & & & 4 + 4W_8^1 & & \\ & & & & & & 8 & \\ & & & & & & & 4 - 4W_8^1 \\ & & & & & & & & 4 + 4W_8^1 \end{bmatrix} \quad (3.57)$$

where $W_8^1 = e^{-j2\pi/8} = e^{-j\pi/4}$ and the blank spaces in the matrix denote the zero elements. The first column in this matrix represents the frequency contents of the first row/column vector of the SCHAT matrix and likewise, for the second column and so on. It can be observed from this matrix that sequency is closely matched to frequency in the DFT. In general, $\mathbf{T}_N$ can be defined using the matrix partitioning method as follows:

$$\mathbf{T}_N = \begin{bmatrix} \mathbf{I}_{N/2} & \boldsymbol{\omega}_1 \\ \mathbf{I}_{N/2} & -\boldsymbol{\omega}_1 \end{bmatrix} \begin{bmatrix} \mathbf{T}_{N/2} & \mathbf{T}_{N/2} \\ \mathbf{T}_{N/2}\mathbf{S}_{N/2} & -\mathbf{T}_{N/2}\mathbf{S}_{N/2} \end{bmatrix} \quad (3.58)$$

where $\mathbf{I}_{N/2}$ is the $N \times N$ identity matrix, $\boldsymbol{\omega}_1$ and $\mathbf{S}_{N/2}$ are as defined in (3.46) and (3.26). We can see from the above equation that $\mathbf{T}_N$ can be obtained recursively from $\mathbf{T}_{N/2}$ and some matrices with the twiddle factors. The smallest dimension of the $\mathbf{T}_N$

matrix will be 4×4 which is $\mathbf{T}_4 = 4\mathbf{I}_4$. Due to the sparseness of the $\mathbf{T}_N$ matrix and its symmetric nature, $\mathbf{T}_N$ can be factored into a product of sparse matrices, which will lead to the fast algorithm to compute the $\mathbf{T}_N$ transformation. For example, $\mathbf{T}_8$ can be factorized as follows:

$$\mathbf{T}_8 = \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{I}_4 & -\mathbf{I}_4 \end{bmatrix} \begin{bmatrix} \mathbf{I}_4 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\omega}'_1 \end{bmatrix} \begin{bmatrix} \mathbf{T}_4 & \mathbf{T}_4 \\ \mathbf{T}_4 & -\mathbf{T}_4 \end{bmatrix} \quad (3.59)$$

where $\boldsymbol{\omega}'_1 = \text{diag}\{1, W_8^1, 1, W_8^1\}$. For $N = 16$, (3.58) will become

$$\mathbf{T}_{16} = \begin{bmatrix} \mathbf{I}_8 & \mathbf{I}_8 \\ \mathbf{I}_8 & -\mathbf{I}_8 \end{bmatrix} \begin{bmatrix} \mathbf{I}_8 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\omega}_1 \end{bmatrix} \begin{bmatrix} \mathbf{T}_8 & \mathbf{T}_8 \\ \mathbf{T}_8 \mathbf{S}_8 & -\mathbf{T}_8 \mathbf{S}_8 \end{bmatrix}. \quad (3.60)$$

Substituting $\mathbf{T}_8$ into (3.60), we have

$$\begin{aligned} \mathbf{T}_{16} = & \begin{bmatrix} \mathbf{I}_8 & \mathbf{I}_8 \\ \mathbf{I}_8 & -\mathbf{I}_8 \end{bmatrix} \begin{bmatrix} \mathbf{I}_8 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\omega}_1 \end{bmatrix} \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 & & \\ & \mathbf{I}_4 & -\mathbf{I}_4 & \\ & & \mathbf{I}_4 & \mathbf{I}_4 \\ & & \mathbf{I}_4 & -\mathbf{I}_4 \end{bmatrix} \begin{bmatrix} \mathbf{I}_4 & \mathbf{0} & & \\ & \mathbf{0} & \boldsymbol{\omega}'_2 & \\ & & \mathbf{I}_4 & \mathbf{0} \\ & & \mathbf{0} & \boldsymbol{\omega}'_2 \end{bmatrix} \\ & \cdot \begin{bmatrix} \mathbf{T}_4 & \mathbf{T}_4 & & \\ & \mathbf{T}_4 & -\mathbf{T}_4 & \\ & & \mathbf{T}_4 & \mathbf{T}_4 \\ & & \mathbf{T}_4 & -\mathbf{T}_4 \end{bmatrix} \begin{bmatrix} \mathbf{I}_8 & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_8 \end{bmatrix} \begin{bmatrix} \mathbf{I}_8 & \mathbf{I}_8 \\ \mathbf{I}_8 & -\mathbf{I}_8 \end{bmatrix} \end{aligned} \quad (3.61)$$

where $\boldsymbol{\omega}_1 = \text{diag}\{1, W_{16}^1, W_{16}^2, \dots, W_{16}^7\}$ and $\boldsymbol{\omega}'_2 = \text{diag}\{1, W_{16}^2, 1, W_{16}^2\}$. In general, $\mathbf{T}_N$ can be factored into a sequence of matrices as follows:

$$\mathbf{T}_N = 4 \left[\prod_{k=1}^{n-3} \mathbf{B}_{n-k+1} \mathbf{D}_{kk} \right] \mathbf{B}_3 \mathbf{D}_{n-2, n-2} \mathbf{B}_3 \left[\prod_{k=1}^{n-3} \mathbf{H}_{n-k-2, n-k-2} \mathbf{B}_{k+3} \right] \quad (3.62)$$

where $n = \log_2 N$, $\mathbf{B}_{n-k+1}$ and $\mathbf{H}_{n-k-2, n-k-2}$ where $k = 1, 2, \dots, n-3$ are as defined in (3.30) and (3.37),

$$\begin{aligned}
\mathbf{D}_{11} &= (\mathbf{I}_{2^{n-1}} \oplus \boldsymbol{\omega}_1) \\
\mathbf{D}_{22} &= (\mathbf{I}_{2^{n-2}} \oplus \boldsymbol{\omega}_2 \oplus \mathbf{I}_{2^{n-2}} \oplus \boldsymbol{\omega}_2) \\
&\vdots \\
\mathbf{D}_{n-2, n-2} &= (\mathbf{I}_4 \oplus \boldsymbol{\omega}'_{n-2} \oplus \dots \oplus \mathbf{I}_4 \oplus \boldsymbol{\omega}'_{n-2})
\end{aligned} \tag{3.63}$$

where $\oplus$ is the direct sum operator [17], $\boldsymbol{\omega}_k$ are also as defined in (3.46) and $\boldsymbol{\omega}'_{n-2} = \text{diag}\{1, W_8^1, 1, W_8^1\}$. This factorization results in the fast T transform algorithm to compute (3.56), which can provide the DFT spectrum from the SCHAT spectrum.

As an example, the signal flow graph for $N = 8$ is illustrated in Figure 3.5 (the equations are shown in (3.56) and (3.59)). It can be seen from the figure that an 8-point T transform can be decomposed into two stages using the fast algorithm. Each stage needs eight complex addition/subtractions so 16 complex addition/subtractions and two complex multiplications for the twiddle factors are needed to compute an 8-

Figure 3.5: The flow graph of fast DFT algorithm using the SCHAT.

Table 3.5: Comparison of computational complexity of the DFT via the SCHAT and the radix-2 FFT.

N	DFT via SCHAT			Radix-2 FFT		
	Stages	$\gamma(N)$	$\mu(N)$	Stages	$\gamma(N)$	$\mu(N)$
8	2	2	16	3	2	24
16	4	10	64	4	10	64
32	6	34	192	5	34	160
64	8	98	512	6	98	384
128	10	256	1280	7	256	896
256	12	638	3072	8	638	2048

$\gamma(N)$ - The number of complex multiplications.

$\mu(N)$ - The number of complex addition/subtractions.

Note: Multiplications involving the twiddle factors of $\{\pm 1, \pm j\}$ are not counted.

point DFT from the SCHAT coefficients. In general, the fast algorithm is decomposed into $2(n-2)$ stages and each stage computes N complex addition/subtractions. Table 3.5 shows the comparison between the DFT via the SCHAT using the fast T transform and the radix-2 FFT algorithm in order to compute an N -point DFT.

As shown in Table 3.5, the number of complex multiplications for both methods are the same for any transform dimension N . For $N = 8$, it is observed that the first method (i.e., the DFT via the SCHAT) requires less number of complex addition/subtractions than the radix-2 FFT. But when N becomes larger, that is $N > 16$, the first method needs more number of complex addition/subtractions than the radix-2 FFT to compute an N -point DFT. This is because the former contains more stages (which is $2(n-2)$ stages) than the latter (which has only n stages) in decomposition. Hence, the fast T transform algorithm needs more additions than the radix-2 FFT algorithm to compute the DFT even though they have the same number of complex

multiplications. Therefore, this introduced algorithm for computation of the DFT via the SCHAT is not superior to the radix-2 FFT in terms of the number of complex addition/subtractions. But the DFT computation via the SCHAT ought to be considered in some applications where both DFT and SCHAT coefficients are desired, especially when the DFT coefficients are needed after the SCHAT computation. It is also noted that $\mathbf{T}_N$ derived in (3.52) is an orthogonal matrix in the complex domain since it holds $\mathbf{T}_N \mathbf{T}_N^* = N^2 \mathbf{I}_N$ where $\{ \}^*$ represents the complex conjugate transpose operation and $\mathbf{I}_N$ is the identity matrix of size $N \times N$. Therefore, $\mathbf{T}_N^{-1} = (1/N^2) \mathbf{T}_N^*$ and by using (3.56), the SCHAT coefficients can be obtained from the DFT coefficients as well.

Figure 3.6 shows the magnitude response of each row vector of the SCHAT matrix for $N = 8$. This is in fact corresponding to the $\mathbf{T}_8$ matrix derived in (3.57). It can be seen from the figure that sequency is closely related to frequency in the DFT. But it is noted that some energy in frequency domain is distributed over different sequency

Figure 3.6: Magnitude response of the SCHAT matrix ($N = 8$).

Figure 3.7: Magnitude spectra of the DFT and SHT for the periodic square wave of 20 Hz.

coefficients as shown in Figure 3.6-b, d, f and h. This is because the DFT bases are sinusoids with harmonic frequencies whereas the SHT is based on the (complex) square waves with increasing sequences.

Figure 3.7 illustrates the magnitude spectra of 64-point DFT and SHT of a square wave of 20 Hz using 64 samples per second. This simulation setting is similar to the setting used in Subsection 3.2.6-A which produces Figure 3.3 except that a square wave is used for this simulation. $\text{signum}(\sin 2\pi ft)$ was used in order to get a corresponding square wave. Obviously, some unwanted spurious spikes are introduced in the DFT spectrum as shown in the figure. But it can be seen from the figure that the SHT magnitude spectrum is more closely matched with that of the DFT as compared to the spectra obtained in Figure 3.3. That is, the changes of the DFT spectra are more apparent than that of the SHT spectra as shown in Figures 3.3 and 3.7. Therefore, sequency spectrum may be useful to some extent for such estimation while providing an advantage of very less computational complexity as compared to frequency spectrum obtained by the DFT.

3.3 Natural-Ordered Complex Hadamard Transform (NCHT)

In this section, we shall introduce another ordered version of complex Hadamard transform called natural-ordered complex Hadamard transform (NCHT). Its order corresponds to that of the WHT matrices (i.e., natural or Hadamard order). Figure 3.8 shows the different orderings of the complex Hadamard transforms in parallel to three staple orderings of the real Hadamard transforms. It can be seen that NCHT is a complex counterpart of the WHT while SCHT is an extended complex version of the SOWT as mentioned earlier. Generally, NCHT is a discrete orthogonal transform whose elements are confined to four complex integer values $\{\pm 1, \pm j\}$, and its row vectors are orthogonal to each other in the complex domain.

Figure 3.8: Different orderings of the Hadamard transforms.

3.3.1 Basic Definitions

The construction of the NCHT is based on the WHT and the direct block matrix operation (which is to be defined later in this subsection). Before going directly

into the generation of the NCHT, we first describe the construction of the WHT. As mentioned in Section 2.3 of Chapter 2, the WHT matrix $\mathbf{W}_N$ is a square and symmetric matrix which is defined as

$$\mathbf{W}_N = \left[\begin{array}{c|c} \mathbf{W}_{N/2} & \mathbf{W}_{N/2} \\ \hline \mathbf{W}_{N/2} & -\mathbf{W}_{N/2} \end{array} \right] \quad (3.64)$$

where $\mathbf{W}_1 = 1$ and $N = 2^n$. The NCHT matrix is also constructed using a similar way of generating a WHT matrix but the multiplication via the $\mathbf{S}$ matrices are introduced in the lower half of the matrix. Let $\mathbf{H}_N$ be any NCHT matrix of size $N \times N$. Then, it is a square matrix defined by

$$\mathbf{H}_N = \left[\begin{array}{c|c} \mathbf{H}_{N/2} & \mathbf{H}_{N/2} \\ \hline \mathbf{H}_{N/2}\mathbf{S}_{N/2} & -\mathbf{H}_{N/2}\mathbf{S}_{N/2} \end{array} \right] \quad (3.65)$$

where $N = 2^n$,

$$\mathbf{S}_{2^{n-1}} = \left[\begin{array}{cc} \mathbf{I}_{2^{n-2}} & \mathbf{0} \\ \mathbf{0} & j\mathbf{I}_{2^{n-2}} \end{array} \right]$$

and $\mathbf{I}_{2^{n-2}}$ is the identity matrix of order $2^{n-2} \times 2^{n-2}$. In this way, an $N \times N$ NCHT matrix can be constructed using the NCHT matrices of size $(N/2) \times (N/2)$, and the smallest dimension of the NCHT matrix is 4×4 which is expressed as

$$\mathbf{H}_4 = \left[\begin{array}{c|c} \mathbf{H}_2 & \mathbf{H}_2 \\ \hline \mathbf{H}_2\mathbf{S}_2 & -\mathbf{H}_2\mathbf{S}_2 \end{array} \right] = \left[\begin{array}{cccc} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \\ 1 & -j & -1 & j \end{array} \right] \quad (3.66)$$

where

$$\mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad \text{and} \quad \mathbf{S}_2 = \begin{bmatrix} 1 & 0 \\ 0 & j \end{bmatrix}.$$

Therefore, the NCHT matrix can be constructed using the WHT matrix and the direct block matrix operator as follows:

$$\mathbf{H}_N = \mathbf{W}_N \diamond \mathbf{A}_{n-1,n-1} \diamond \cdots \diamond \mathbf{A}_{1,1} \quad (3.67)$$

where $\mathbf{W}_N$ is the WHT matrix of dimension $N \times N$, $\mathbf{A}_{k,k}$ where $k = 1, 2, \dots, n-1$ are the matrices which are defined as

$$\begin{aligned} \mathbf{A}_{1,1} &= [\mathbf{I}_{2^{n-1}}, \mathbf{S}_{2^{n-1}}]^T \\ \mathbf{A}_{2,2} &= [\mathbf{I}_{2^{n-2}}, \mathbf{S}_{2^{n-2}}, \mathbf{I}_{2^{n-2}}, \mathbf{S}_{2^{n-2}}]^T \\ &\vdots \\ \mathbf{A}_{n-1,n-1} &= [\mathbf{I}_2, \mathbf{S}_2, \cdots, \mathbf{I}_2, \mathbf{S}_2]^T \end{aligned} \quad (3.68)$$

and $\diamond$ denotes the direct block matrix operator which is defined as follows.

Let $\mathbf{B}$ be a block matrix such that

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} & \cdots & \mathbf{B}_{1N} \\ \mathbf{B}_{21} & \mathbf{B}_{22} & \cdots & \mathbf{B}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{B}_{M1} & \mathbf{B}_{M2} & \cdots & \mathbf{B}_{MN} \end{bmatrix},$$

$\mathbf{R}$ be a row block matrix, that is,

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 & \mathbf{R}_2 & \cdots & \mathbf{R}_N \end{bmatrix}$$

and $\mathbf{C}$ be a column block matrix which is

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_1 & \mathbf{C}_2 & \cdots & \mathbf{C}_M \end{bmatrix}^T$$

where the submatrices $\mathbf{B}_{mn}$, $\mathbf{R}_k$ and $\mathbf{C}_l$ are the square matrices which have the same dimensions, and m, n, k and l are positive integers such that $1 \leq m, l \leq M$ and $1 \leq n, k \leq N$. Then the direct block matrix operator $\diamond$ is defined as

$$\mathbf{B} \diamond \mathbf{R} = \begin{bmatrix} \mathbf{B}_{11}\mathbf{R}_1 & \mathbf{B}_{12}\mathbf{R}_2 & \cdots & \mathbf{B}_{1N}\mathbf{R}_N \\ \mathbf{B}_{21}\mathbf{R}_1 & \mathbf{B}_{22}\mathbf{R}_2 & \cdots & \mathbf{B}_{2N}\mathbf{R}_N \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{B}_{M1}\mathbf{R}_1 & \mathbf{B}_{M2}\mathbf{R}_2 & \cdots & \mathbf{B}_{MN}\mathbf{R}_N \end{bmatrix} \quad (3.69)$$

and

$$\mathbf{B} \diamond \mathbf{C} = \begin{bmatrix} \mathbf{B}_{11}\mathbf{C}_1 & \mathbf{B}_{12}\mathbf{C}_1 & \cdots & \mathbf{B}_{1N}\mathbf{C}_1 \\ \mathbf{B}_{21}\mathbf{C}_2 & \mathbf{B}_{22}\mathbf{C}_2 & \cdots & \mathbf{B}_{2N}\mathbf{C}_2 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{B}_{M1}\mathbf{C}_M & \mathbf{B}_{M2}\mathbf{C}_M & \cdots & \mathbf{B}_{MN}\mathbf{C}_M \end{bmatrix} \quad (3.70)$$

If the dimensions of the above submatrices are 1×1 , the operation of the direct block matrix operator is similar to that of the direct matrix operator defined in [10]. Let us, for example, consider $N = 8$. Then $n = \log_2 8 = 3$ and (3.67) will become

$$\mathbf{H}_8 = \mathbf{W}_8 \diamond \mathbf{A}_{2,2} \diamond \mathbf{A}_{1,1}. \quad (3.71)$$

Substituting the corresponding values defined in (3.68) into (3.71), we have

$$\mathbf{H}_8 = \left[\begin{array}{cc|cc} \mathbf{W}_2 & \mathbf{W}_2 & \mathbf{W}_2 & \mathbf{W}_2 \\ \mathbf{W}_2 & -\mathbf{W}_2 & \mathbf{W}_2 & -\mathbf{W}_2 \\ \hline \mathbf{W}_2 & \mathbf{W}_2 & -\mathbf{W}_2 & -\mathbf{W}_2 \\ \mathbf{W}_2 & -\mathbf{W}_2 & -\mathbf{W}_2 & \mathbf{W}_2 \end{array} \right] \diamond \left[\begin{array}{c} \mathbf{I}_2 \\ \mathbf{S}_2 \\ \mathbf{I}_2 \\ \mathbf{S}_2 \end{array} \right] \diamond \left[\begin{array}{c} \mathbf{I}_4 \\ \mathbf{S}_4 \end{array} \right] \quad (3.72)$$

where $\mathbf{W}_2$ is the 2×2 WHT matrix. Thus,

$$\mathbf{H}_8 = \left[\begin{array}{cccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j & 1 & j & -1 & -j \\ 1 & -j & -1 & j & 1 & -j & -1 & j \\ 1 & 1 & j & j & -1 & -1 & -j & -j \\ 1 & -1 & j & -j & -1 & 1 & -j & j \\ 1 & j & -j & 1 & -1 & -j & j & -1 \\ 1 & -j & -j & -1 & -1 & j & j & 1 \end{array} \right]. \quad (3.73)$$

Since the NCHT is corresponding to the WHT, its construction is based on the WHT and recursive operation as well. The NCHT, however, is a complex Hadamard transform which is useful to deal with the complex-valued functions as compared to the WHT. With the NCHT matrix developed, the NCHT of a complex signal vector $\mathbf{x}_N = [x(0), x(1), \dots, x(N-1)]^T$ is defined as

$$\mathbf{X}_N = \frac{1}{N} \overline{\mathbf{H}}_N \mathbf{x}_N \quad (3.74)$$

where $N = 2^n$, $\mathbf{X}_N = [X(0), X(1), \dots, X(N-1)]^T$ is the transformed complex column vector and $\overline{\mathbf{H}}_N$ is the complex conjugate of the NCHT matrix $\mathbf{H}_N$. The data sequence

can be retrieved uniquely from the inverse transform which is given by

$$\mathbf{x}_N = \mathbf{H}_N^T \mathbf{X}_N \quad (3.75)$$

where $\{ \}^T$ denotes the transpose and $\mathbf{H}_N^T \overline{\mathbf{H}}_N = \mathbf{H}_N^* \mathbf{H}_N = N \mathbf{I}_N$ (unitary property) where $\{ \}^*$ represents the complex conjugate transpose.

3.3.2 Properties of the NCHT

This subsection presents the properties of the NCHT.

Property 1: Exponential Property of the NCHT

Let $\mathbf{H}_N$ be an NCHT matrix of the dimension $N \times N$ where $N = 2^n$ and $h(p, k)$ be any element of $\mathbf{H}_N$ at p th row and k th column, where $p, k = 0, 1, \dots, N-1$. Then,

$$h(p, k) = (-1)^{\sum_{r=0}^{n-1} p_r k_r} (j)^{\sum_{r=0}^{n-1} \check{p}_r k_r} \quad (3.76)$$

where $n = \log_2 N$ and p_r, k_r are the r th bits of the binary representations of the decimal integers p and k respectively, that is, $(p)_{10} = (p_{n-1}, p_{n-2}, \dots, p_0)_2$ and $(k)_{10} = (k_{n-1}, k_{n-2}, \dots, k_0)_2$. $\check{p}_r$ represents the $(n-1)$ least significant binary bits (LSB) of the bit reversal of the decimal integer p in a bit-reversed order where $r = 0, 1, \dots, n-1$. Table 3.6 shows the binary representations of $\check{p}_r$ for $N = 8$. The construction for an 8×8 NCHT matrix in the exponential form is further elaborated in Tables 3.7, 3.8 and 3.9. The complex conjugate of the NCHT matrix can be generated by just changing j to $-j$ in (3.76).

Property 2: Unitary Property

The NCHT is a unitary transform whose row vectors are orthogonal in the complex domain just like the SCHT. Let $\mathbf{H}_N$ be any NCHT matrix of order $N \times N$ where $N = 2^n$, then $\mathbf{H}_N$ is said to be orthogonal in the complex domain as

p	Binary	Bit Reversal	$(n - 1)$ LSB bits	Bit Reversal	$\check{p}_r$
0	000	000	00	00	000
1	001	100	00	00	000
2	010	010	10	01	001
3	011	110	10	01	001
4	100	001	01	10	010
5	101	101	01	10	010
6	110	011	11	11	011
7	111	111	11	11	011

Table 3.6: Binary notations for $\check{p}_r$.

		k_r							
		000	001	010	011	100	101	110	111
p_r	000	1	1	1	1	1	1	1	1
	001	1	-1	1	-1	1	-1	1	-1
	010	1	1	-1	-1	1	1	-1	-1
	011	1	-1	-1	1	1	-1	-1	1
	100	1	1	1	1	-1	-1	-1	-1
	101	1	-1	1	-1	-1	1	-1	1
	110	1	1	-1	-1	-1	-1	1	1
	111	1	-1	-1	1	-1	1	1	-1

Table 3.7: Elements generated from $(-1)^{\sum_{r=0}^{n-1} p_r k_r}$.

		k_r							
		000	001	010	011	100	101	110	111
$\check{p}_r$	000	1	1	1	1	1	1	1	1
	000	1	1	1	1	1	1	1	1
	001	1	j	1	j	1	j	1	j
	001	1	j	1	j	1	j	1	j
	010	1	1	j	j	1	1	j	j
	010	1	1	j	j	1	1	j	j
	011	1	j	j	-1	1	j	j	-1
	011	1	j	j	-1	1	j	j	-1

Table 3.8: Elements generated from $(j)^{\sum_{r=0}^{n-1} \check{p}_r k_r}$.

		k_r							
		000	001	010	011	100	101	110	111
p_r	000	1	1	1	1	1	1	1	1
	001	1	-1	1	-1	1	-1	1	-1
	010	1	j	-1	$-j$	1	j	-1	$-j$
	011	1	$-j$	-1	j	1	$-j$	-1	j
	100	1	1	j	j	-1	-1	$-j$	$-j$
	101	1	-1	j	$-j$	-1	1	$-j$	j
	110	1	j	$-j$	1	-1	$-j$	j	-1
	111	1	$-j$	$-j$	-1	-1	j	j	1

Table 3.9: Elements generated from $(-1)^{\sum_{r=0}^{n-1} p_r k_r} \cdot (j)^{\sum_{r=0}^{n-1} \check{p}_r k_r}$.

$$|\det \mathbf{H}_N| = N^{N/2} \quad (3.77)$$

and

$$\mathbf{H}_N \mathbf{H}_N^* = \mathbf{H}_N^* \mathbf{H}_N = N \mathbf{I}_N \quad (3.78)$$

where $\mathbf{H}_N^*$ represents the complex conjugate transpose of the matrix $\mathbf{H}_N$.

Property 3: Linearity

The NCHT is a linear operator just like the SCHT.

Property 4: Parseval's Theorem

The NCHT obeys the Parseval's Theorem in which the energy of the signal in the time and transform domains is related as

$$\frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2 = \sum_{k=0}^{N-1} |X(k)|^2. \quad (3.79)$$

Proof: Using equation (3.74),

$$\begin{aligned} \mathbf{X}_N^* \mathbf{X}_N &= \frac{1}{N} [\overline{\mathbf{H}_N \mathbf{x}_N}]^* \frac{1}{N} \overline{\mathbf{H}_N \mathbf{x}_N} \\ &= \frac{1}{N^2} \mathbf{x}_N^* \mathbf{H}_N^T \overline{\mathbf{H}_N \mathbf{x}_N} \\ &= \frac{1}{N} \mathbf{x}_N^* \mathbf{x}_N \end{aligned}$$

where $\mathbf{H}_N^T \overline{\mathbf{H}_N} = N \mathbf{I}_N$. The proof is completed here.

Property 5: Shift Invariant Power Spectrum

The NCHT is invariant to the cyclic shift of the data sequence $\{x(p)\}_0^{N-1}$ of length N . Let $\mathbf{x}_N^m$ be the data sequence, $x(p)$, which is cyclically shifted to the right by m places, that is, $\mathbf{x}_N^m = [x(N-m), \dots, x(N-1), x(0), \dots, x(N-m-1)]^T$ where

$m = 1, 2, \dots, N-1$. Then, the NCHT of the shifted data sequence, $\mathbf{x}_N^m$, is related to $\mathbf{X}_N$ as

$$\mathbf{X}_N^m = \frac{1}{N} \bar{\mathbf{H}}_N \mathbf{x}_N^m = \mathbf{R}_N^m \mathbf{X}_N \quad (3.80)$$

where

$$\mathbf{R}_N^m = \frac{1}{N} \bar{\mathbf{H}}_N \mathbf{P}_N^m \mathbf{H}_N^T \quad (3.81)$$

is the cyclic transformation matrix which expresses the relation of $\mathbf{X}_N^m$ with $\mathbf{X}_N$ and $\mathbf{P}_N^m$ is the permutation matrix of size $N \times N$, defined by the identity matrix whose rows are cyclically shifted to the bottom by m places. $\mathbf{R}_N^m$ has a block diagonal unitary structure which leads to the shift invariant NCHT power spectra which can be defined as

$$\begin{aligned} P(k) &= |X(k)|^2 = |X^m(k)|^2, \quad k = 0, 1, 2, 3 \\ P(2l+2) &= \sum_{k=2^{l+1}}^{3 \cdot 2^l - 1} |X(k)|^2 = \sum_{k=2^{l+1}}^{3 \cdot 2^l - 1} |X^m(k)|^2, \\ P(2l+3) &= \sum_{k=3 \cdot 2^l}^{2^{l+2} - 1} |X(k)|^2 = \sum_{k=3 \cdot 2^l}^{2^{l+2} - 1} |X^m(k)|^2 \end{aligned} \quad (3.82)$$

where $l = 1, 2, \dots, n-2$ and $n = \log_2 N$. Therefore, the NCHT has $2n$ power spectral points. For instance, we consider $N = 8$ and $m = 1$. Equation (3.81) becomes

$$\begin{aligned} \mathbf{R}_8^1 &= \frac{1}{8} \bar{\mathbf{H}}_8 \mathbf{P}_8^1 \mathbf{H}_8^T \\ \mathbf{R}_8^1 &= \text{diag} \left\{ 1, -1, j, -j, \begin{bmatrix} \frac{1}{2} - \frac{1}{2}j & \frac{1}{2} + \frac{1}{2}j \\ -\frac{1}{2} - \frac{1}{2}j & -\frac{1}{2} + \frac{1}{2}j \end{bmatrix}, \begin{bmatrix} -\frac{1}{2} - \frac{1}{2}j & \frac{1}{2} - \frac{1}{2}j \\ -\frac{1}{2} + \frac{1}{2}j & \frac{1}{2} + \frac{1}{2}j \end{bmatrix} \right\} \end{aligned} \quad (3.83)$$

where $\text{diag}\{\cdot\}$ represents the diagonal matrix. Since $n = \log_2 8 = 3$, the NCHT has 6

power spectral points and $\mathbf{R}_8^1$ will result in the following shift invariant power spectra

$$\begin{aligned} P(k) &= |X(k)|^2 = |X^m(k)|^2, \quad k = 0, 1, 2, 3 \\ P(4) &= |X(4)|^2 + |X(5)|^2 = |X^1(4)|^2 + |X^1(5)|^2 \\ P(5) &= |X(6)|^2 + |X(7)|^2 = |X^1(6)|^2 + |X^1(7)|^2. \end{aligned} \quad (3.84)$$

Besides, it is also observed that the total energy of the signal is conserved, that is,

$$\sum_{k=0}^{N-1} |X^m(k)|^2 = \sum_{k=0}^{N-1} |X(k)|^2 = \frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2. \quad (3.85)$$

3.3.3 Fast NCHT Algorithm

In this subsection, an efficient algorithm for computing the NCHT is developed using the sparse matrix factorization approach. In order to conceptualize the factorization method, let us, for example, consider $N = 8$. Then, (3.65) will become

$$\mathbf{H}_8 = \begin{bmatrix} \mathbf{H}_4 & \mathbf{H}_4 \\ \mathbf{H}_4 \mathbf{S}_4 & -\mathbf{H}_4 \mathbf{S}_4 \end{bmatrix} \quad (3.86)$$

where

$$\mathbf{S}_4 = \begin{bmatrix} \mathbf{I}_2 & \mathbf{0} \\ \mathbf{0} & j\mathbf{I}_2 \end{bmatrix}.$$

Equation (3.86) can be factorized as

$$\mathbf{H}_8 = \begin{bmatrix} \mathbf{H}_4 & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_4 \end{bmatrix} \begin{bmatrix} \mathbf{I}_4 & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_4 \end{bmatrix} \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{I}_4 & -\mathbf{I}_4 \end{bmatrix}. \quad (3.87)$$

$\mathbf{H}_4$ can be further factorized and expressed in terms of $\mathbf{H}_2$. That is,

$$\mathbf{H}_8 = \begin{bmatrix} \mathbf{H}_2 & & & \\ & \mathbf{H}_2 & & \\ & & \mathbf{H}_2 & \\ & & & \mathbf{H}_2 \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & & & \\ & \mathbf{S}_2 & & \\ & & \mathbf{I}_2 & \\ & & & \mathbf{S}_2 \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & \mathbf{I}_2 & & \\ \mathbf{I}_2 & -\mathbf{I}_2 & & \\ & & \mathbf{I}_2 & \mathbf{I}_2 \\ & & \mathbf{I}_2 & -\mathbf{I}_2 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I}_4 & \\ & \mathbf{S}_4 \end{bmatrix} \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{I}_4 & -\mathbf{I}_4 \end{bmatrix} \quad (3.88)$$

where the blank spaces in the matrices of the above equation represents the zero elements,

$$\mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad \text{and} \quad \mathbf{S}_2 = \begin{bmatrix} 1 & 0 \\ 0 & j \end{bmatrix}.$$

The factorization of $\overline{\mathbf{H}}_8$ can be obtained by changing j to $-j$ in the $\mathbf{S}$ matrices of (3.88). This factorization results in fast forward NCHT algorithm, since $\overline{\mathbf{H}}_N$ is the forward NCHT transformation matrix as defined in (3.74). The scaling factor of $(1/N)$ can be added at either the start or the end of the computation. The decomposition of fast forward 8-point NCHT is illustrated by the signal flow graph in Figure 3.9. The multiplication by the scaling factor $(1/N)$ can be done by the binary shifting operations without involving any arithmetic computation. For simplicity, the scaling factor is omitted in the signal flow diagram. As shown in Figure 3.9, the flow graph contains three stages and each stage performs eight addition/subtractions. In each stage, two trivial multiplications by the factor $-j$ are required except for the first stage. In general, for an N -point NCHT, the fast algorithm needs $n = \log_2 N$ stages and each stage requires N addition/subtractions. Therefore, the total computational requirements are $N \log_2 N$ addition/subtractions and $(N/4) \log_2(N/2)$ multiplications by the trivial twiddle factors. Similar to that of fast SCHT algorithm reported in

Figure 3.9: Signal flow graph of the fast forward NCHT decomposition ($N = 8$).

Section 3.2.5, this fast NCHT algorithm reduces the computational cost significantly as compared to the direct computation. Since the NCHT matrix is constructed through the recursive operation like that of the WHT, the transform is simple and easy to derive the fast algorithm.

The NCHT matrices are not symmetric, hence, the inverse NCHT has different signal flow graph as compared to that of the forward NCHT. But they have the same computational requirements. The inverse NCHT algorithm is derived as follows. As defined in (3.75), the inverse transformation matrix is

$$\mathbf{H}_N^T = \begin{bmatrix} \mathbf{H}_{N/2} & \mathbf{H}_{N/2} \\ \mathbf{H}_{N/2}\mathbf{S}_{N/2} & -\mathbf{H}_{N/2}\mathbf{S}_{N/2} \end{bmatrix}^T. \quad (3.89)$$

Let us consider for $N = 8$ to conceptualize the factorization method. Hence, (3.89) becomes

$$\mathbf{H}_8^T = \begin{bmatrix} \mathbf{H}_4^T & \mathbf{S}_4^T \mathbf{H}_4^T \\ \mathbf{H}_4^T & -\mathbf{S}_4^T \mathbf{H}_4^T \end{bmatrix} \quad (3.90)$$

which can be expressed as

$$\mathbf{H}_8^T = \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{I}_4 & -\mathbf{I}_4 \end{bmatrix} \begin{bmatrix} \mathbf{I}_4 & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_4 \end{bmatrix} \begin{bmatrix} \mathbf{H}_4^T & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_4^T \end{bmatrix} \quad (3.91)$$

where $\mathbf{S}_4^T = \mathbf{S}_4$. Finally, we can have

$$\mathbf{H}_8^T = \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{I}_4 & -\mathbf{I}_4 \end{bmatrix} \begin{bmatrix} \mathbf{I}_4 & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_4 \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & \mathbf{I}_2 & & \\ \mathbf{I}_2 & -\mathbf{I}_2 & & \\ & & \mathbf{I}_2 & \mathbf{I}_2 \\ & & \mathbf{I}_2 & -\mathbf{I}_2 \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & & & \\ & \mathbf{S}_2 & & \\ & & \mathbf{I}_2 & \\ & & & \mathbf{S}_2 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{H}_2 & & & \\ & \mathbf{H}_2 & & \\ & & \mathbf{H}_2 & \\ & & & \mathbf{H}_2 \end{bmatrix} \quad (3.92)$$

since

$$\mathbf{H}_4^T = \begin{bmatrix} \mathbf{H}_2^T & \mathbf{S}_2^T \mathbf{H}_2^T \\ \mathbf{H}_2^T & -\mathbf{S}_2^T \mathbf{H}_2^T \end{bmatrix}, \quad (3.93)$$

$\mathbf{H}_2^T = \mathbf{H}_2$ and $\mathbf{S}_2^T = \mathbf{S}_2$. Equation (3.92) will result in the fast inverse NCHT algorithm. This factorization is similar to the method developed for the fast inverse SCHT algorithm except that the former does not need the input or output sequences to be in bit-reversed order. Therefore, it has the same signal flow graph as shown in Figure 3.2 without a bit-reversed ordering of the input or the output.

3.3.4 Relationship of the NCHT with the SCHT

As mentioned earlier, the SCHT matrices are generated from the complex Rademacher functions, and their row vectors are arranged in an ascending order of sequences. The

row vectors of both SCHAT and NCHAT matrices are the same except that the ways they are positioned in the transform matrices are different. In fact, this is similar to what is happening in the WHT and the SOWT where the row vectors of both transform matrices are the same but they are positioned in different ordered manners.

The natural order and sequency order of complex Hadamard matrices are related through a bit reversal conversion. Let $\mathbf{H}_N^{SCHAT}$ be a complex Hadamard matrix of size $N \times N$ with sequency ordering and also $\mathbf{H}_N^{NCHAT}$ be an NCHAT matrix of dimension $N \times N$ where $N = 2^n$. Then, they are related as

$$\mathbf{H}_N^{SCHAT}(p, k) = \mathbf{H}_N^{NCHAT}(b(p), k) \quad (3.94)$$

where $p, k = 0, 1, \dots, N - 1$ and $b(p)$ represents the decimal number obtained by the bit reversal conversion of p . A list of converted values for $N = 8$ is given in Figure 3.10. Hence, an SCHAT matrix can be obtained by rearranging the row vectors of an NCHAT matrix in bit reversal ordering and vice versa.

Figure 3.10: Relationship between the sequency and natural orderings of complex Hadamard matrices.

The SCHAT matrices are constructed based on the complex Rademacher functions while the generation of the NCHAT is based on the WHT and the direct block matrix operation as defined in Subsection 3.3.1. Even though they are generated in separate

ways, their row vectors are the same, however, the arrangements they are positioned in the transform matrices are different. Hence, the NCHT can also be employed in DS-CDMA systems as the complex spreading sequences to identify the particular users as described in Subsection 3.2.6-*B*. Since, the NCHT is a bit-reversed version of the SCHT, the applications which are suited to the SCHT could possibly employ the NCHT as well.

3.4 Conclusion

The sequency-ordered complex Hadamard transform is discussed in this chapter and some of its important properties are described and analyzed with the discussion of some potential applications. It has been shown in Section 3.2 that the SCHT matrices can be constructed based on the complex Rademacher matrices. The real Rademacher functions are extended to the complex Rademacher functions which form an orthogonal set of rectangular waveforms taking the four complex integer values $\{\pm 1, \pm j\}$. The exponential property of the SCHT is presented in detail using an example for $N = 8$ in Subsection 3.2.3. The proof of the unitary property of the SCHT matrices is given in Subsection 3.2.4 together with some important properties in that subsection.

Subsection 3.2.5 derives the fast SCHT algorithm based on the sparse matrix factorization method. Its computational complexity is examined as compared to that of the SOWT and the radix-2 FFT. Some promising applications of the SCHT are indicated in Subsection 3.2.6. The SCHT is shown to have certain significance in signal analysis and synthesis with the simplicity of the transform implementation using the fast algorithm. The SCHT is also suggested in communications as the complex spreading sequences in DS-CDMA systems, and in image watermarking to develop a robust watermarking scheme (which are to be presented in detail in Chapter 5 and 6). The radix-2 FFT is also mentioned in this chapter and it has been found that the radix-

2 FFT can be obtained from the SCHAT matrix of dimension 4×4 . The relation of the SCHAT with the UCHT is also described in Subsection 3.2.7. It has been investigated that the SCHAT can be computed recursively from the UCHT and WHT matrices of size 2×2 and some of the sparse matrices. Due to the analogy of sequency to frequency of the DFT, the relationship matrix between the DFT and SCHAT matrices is derived in Subsection 3.2.8. It has been shown that the sequency spectrum of the signal can be converted conveniently to the frequency spectrum through this relationship matrix.

Another major contribution in this chapter is introducing the natural-ordered complex Hadamard transform whose order corresponds to that of the WHT matrices. Section 3.3 outlines the basic definitions and some of its properties followed by the fast NCHT algorithm and its relation to the SCHAT. In fact, the NCHT is another ordered form of complex Hadamard matrix while the SCHAT is a complex Hadamard transform with sequency ordering. It has been shown that the NCHT matrices can be generated based on the WHT and through the direct block matrix operator. The exponential formula to construct the NCHT matrix is also presented with a detailed example using $N = 8$. The NCHT power spectra which are invariant to the cyclic shift of the sampled data sequences are presented in Subsection 3.3.2. In Subsection 3.3.3, the fast algorithms for both forward and inverse NCHT are deduced using the matrix decomposition approach. A detailed example is given for an 8-point NCHT and it is illustrated in the signal flow graph in that subsection. Finally, Subsection 3.3.4 presents the relationship between the NCHT and SCHAT matrices. Since both transforms are the complex integer-valued transforms, they are well suited to be employed in the problems with complex-valued functions. But in this thesis, we focus on the sequency ordering which is normally preferred in practical applications especially in communications and signal processing. The applications of the SCHAT will form the topics of subsequent chapters.

Chapter 4

Pipelined Hardware Structure for the SCHAT

4.1 Introduction

Discrete orthogonal transforms have been used widely in the areas of digital signal processing and communications. Most applications of such transforms, particularly those concerned with communications or online signal processing, require real-time operations to achieve the necessary processing throughput and performance. Hence, special-purpose hardware is needed to implement the transforms efficiently based on the concept of fast algorithms. For example, the constant geometry iterative hardware architecture is used to implement the unified complex Hadamard transform [11] and the Walsh Hadamard transform [7] to produce a useful VLSI integrated circuit.

In this chapter, the hardware implementation of the SCHAT will be presented to support high-speed real-time applications. The decimation-in-sequence (DIS) fast SCHAT algorithm is first proposed to efficiently map the arithmetic operations into the hardware in order to accommodate sequentially presented input/output data streams for a pipelined hardware structure. As pointed out in Chapter 3, the SCHAT matrices

show the sequency ordering and they are promising to be applied in various applications as the DFT. The SCHAT is shown to have certain significance in signal analysis and synthesis as reported in Subsection 3.2.6. Besides, the DFT spectrum can be obtained from the sequency spectrum which is transformed using the SCHAT. In [37], the SCHAT is employed in direct sequence CDMA systems in which each row vector of the SCHAT matrix is used as a complex spreading sequence assigned to a particular user. For detection at the receiver, the SCHAT of the received signal could be used to identify the spreading sequence that was assigned to that particular user. For example, if the i th coefficient is the largest value in the SCHAT transformed vector, the i th row vector of an SCHAT matrix is used as a complex spreading sequence to identify the individual user. Therefore, real-time implementation of the SCHAT is essential.

Besides, in [38], the SCHAT is also utilized in image watermarking to develop a robust watermarking algorithm in which the images are transformed into the sequency domain using the SCHAT. Due to the widespread usage of multimedia applications and high-speed internet technologies, real-time operations are demanded for such applications [39–41]. Furthermore, the data streams originated from or sent to the distant users, for the reason of the transmission economy, normally arrive in bit and/or word sequential format, which should be duly considered when the SCHAT is implemented in hardware.

In the literature, several architectures for the pipelined FFT processor were proposed to associate with sequentially presented data streams [42–44], which can be adopted for the implementation of the SCHAT. In general, the scheme with feed forward shift registers based on the radix-4 fast algorithm [42] requires fewer arithmetic operators which lead to better hardware utilization. Whereas the scheme with feedback shift registers [44] uses a smaller number of data stores (less memory requirement) but more arithmetic operators. They both have a processing latency about N word cycles and are capable of interfacing sequentially presented data streams with a single-path

pipelined structure of cascaded stages.

This chapter will present a pipelined hardware architecture for an N -point SCHAT based on the decimation-in-sequence (DIS) radix-2 fast algorithm. The rest of the chapter is organized as follows. The decimation-in-time (DIT) fast SCHAT algorithm is first mentioned in Section 4.2 and then the DIS fast SCHAT algorithm is derived. By adopting the concept that was used for the implementation of the radix-4 FFT algorithm in [42], the pipelined hardware structure based on the radix-2 DIS fast SCHAT algorithm is presented in Section 4.3. The merits of the proposed pipelined structure are to be discussed in Section 4.4. Finally, this chapter is concluded in Section 4.5.

4.2 Fast SCHAT Algorithms

The N -point SCHAT of a complex signal vector $\mathbf{x}_N = [x(0), x(1), \dots, x(N-1)]^T$ is

$$\mathbf{X}_N = \frac{1}{N} \mathbf{H}_N^* \mathbf{x}_N \quad (4.1)$$

where N is a power of two, $\mathbf{X}_N = [X(0), X(1), \dots, X(N-1)]^T$ is the transformed complex column vector, and $\mathbf{H}_N^*$ is the complex conjugate transpose of the SCHAT matrix $\mathbf{H}_N$ as defined in Subsection 3.2.2.

4.2.1 Decimation-in-Time Fast SCHAT

It is pointed out in Subsection 3.2.5 that $\mathbf{H}_N$ can be factored into a sequence of sparse matrices which contains the identity matrices and the 2×2 WHT matrices. Hence, $\mathbf{H}_N^*$ in (4.1) can be decomposed by the sparse matrix factorization as

$$\mathbf{X}_N = \frac{1}{N} (\mathbf{B}_n \overline{\mathbf{H}}_{11}) (\mathbf{B}_{n-1} \overline{\mathbf{H}}_{22}) \cdots (\mathbf{B}_2 \overline{\mathbf{H}}_{n-1, n-1}) \mathbf{B}_1 (\mathbf{P}_b \mathbf{x}_N) \quad (4.2)$$

where $n = \log_2 N$, $\overline{\{\}} \}$ represents the complex conjugate operation, $\mathbf{P}_b$ is a bit reversal permutation matrix which shuffles $\mathbf{x}_N$ into a bit-reversed order, and

$$\begin{aligned}\overline{\mathbf{H}}_{11} &= (\mathbf{I}_{2^{n-1}} \oplus \overline{\mathbf{S}}_{2^{n-1}}) \\ \overline{\mathbf{H}}_{22} &= (\mathbf{I}_{2^{n-2}} \oplus \overline{\mathbf{S}}_{2^{n-2}} \oplus \mathbf{I}_{2^{n-2}} \oplus \overline{\mathbf{S}}_{2^{n-2}}) \\ &\vdots \\ \overline{\mathbf{H}}_{n-1,n-1} &= (\mathbf{I}_2 \oplus \overline{\mathbf{S}}_2 \oplus \cdots \oplus \mathbf{I}_2 \oplus \overline{\mathbf{S}}_2)\end{aligned}\tag{4.3}$$

where $\oplus$ is the direct sum operator, $\mathbf{I}_{2^{n-r}}$ is the identity matrix of size $2^{n-r} \times 2^{n-r}$,

$$\overline{\mathbf{S}}_{2^{n-r}} = \begin{bmatrix} \mathbf{I}_{2^{n-r-1}} & \mathbf{0} \\ \mathbf{0} & -j\mathbf{I}_{2^{n-r-1}} \end{bmatrix}, \quad 1 \leq r \leq n-1 \tag{4.4}$$

and $\mathbf{B}_k$ where $k = 1, 2, \dots, n$ are the $N \times N$ sparse matrices as defined in (3.29) and (3.30). The above-mentioned fast algorithm, i.e., the implementatin of (4.1) by (4.2) is referred to as the DIT fast SCHAT algorithm as the algorithm is structured to reveal the concept of sample rate reduction (which is similar to the DIT radix-2 FFT algorithm). The signal flow graph for $N = 8$ is shown in Figure 4.1.

4.2.2 Decimation-in-Sequency Fast SCHAT

In this subsection, we will derive the DIS fast SCHAT algorithm that is to be efficiently mapped into the pipelined hardware structure. The term DIS is analogous to decimation-in-frequency (DIF) used to describe the decomposition method for deriving FFT algorithms. As mentioned in (4.1) and (4.2), it is found that $\mathbf{H}_N^*$ can be written as

$$\mathbf{H}_N^* = (\mathbf{B}_n \overline{\mathbf{H}}_{11})(\mathbf{B}_{n-1} \overline{\mathbf{H}}_{22}) \cdots (\mathbf{B}_2 \overline{\mathbf{H}}_{n-1,n-1}) \mathbf{B}_1 \mathbf{P}_b. \tag{4.5}$$

Figure 4.1: The signal flow graph of the DIT fast SCHT algorithm ($N = 8$).

Since the SCHT matrix is symmetric (see Subsection 3.2.4), that is, $\{\mathbf{H}_N^*\}^T = \mathbf{H}_N^*$, $\mathbf{B}_i = \mathbf{B}_i^T$, and $\bar{\mathbf{H}}_{ii} = \bar{\mathbf{H}}_{ii}^T$, it follows that

$$\mathbf{H}_N^* = \mathbf{P}_b^T \mathbf{B}_1 (\bar{\mathbf{H}}_{n-1,n-1} \mathbf{B}_2) \cdots (\bar{\mathbf{H}}_{22} \mathbf{B}_{n-1}) (\bar{\mathbf{H}}_{11} \mathbf{B}_n). \quad (4.6)$$

Substituting (4.6) into (4.1), we have

$$\mathbf{X}_N = \frac{1}{N} \mathbf{P}_b^T \mathbf{B}_1 (\bar{\mathbf{H}}_{n-1,n-1} \mathbf{B}_2) \cdots (\bar{\mathbf{H}}_{22} \mathbf{B}_{n-1}) (\bar{\mathbf{H}}_{11} \mathbf{B}_n) \mathbf{x}_N. \quad (4.7)$$

Multiplying both sides of (4.7) by $\mathbf{P}_b$, it becomes

$$\mathbf{P}_b \mathbf{X}_N = \frac{1}{N} \mathbf{B}_1 (\bar{\mathbf{H}}_{n-1,n-1} \mathbf{B}_2) \cdots (\bar{\mathbf{H}}_{22} \mathbf{B}_{n-1}) (\bar{\mathbf{H}}_{11} \mathbf{B}_n) \mathbf{x}_N \quad (4.8)$$

where $\mathbf{P}_b \mathbf{P}_b^T = \mathbf{I}_N$. The SCHT transformed vector is shuffled into a bit-reversal order in the sequency domain, which results in the DIS fast SCHT algorithm. Let us, for

instance, consider $N = 8$. Then (4.8) becomes

$$\begin{aligned}\mathbf{P}_b \mathbf{X}_8 &= \frac{1}{8} \mathbf{B}_1 \overline{\mathbf{H}}_{22} \mathbf{B}_2 \overline{\mathbf{H}}_{11} \mathbf{B}_3 \mathbf{x}_8 \\ &= \frac{1}{8} \mathbf{B}_1 (\mathbf{I}_2 \oplus \overline{\mathbf{S}}_2 \oplus \mathbf{I}_2 \oplus \overline{\mathbf{S}}_2) \mathbf{B}_2 (\mathbf{I}_4 \oplus \overline{\mathbf{S}}_4) \mathbf{B}_3 \mathbf{x}_8\end{aligned}\quad (4.9)$$

where $\mathbf{B}_1 = \mathbf{C}_0 \oplus \mathbf{C}_0 \oplus \mathbf{C}_0 \oplus \mathbf{C}_0$ and $\mathbf{C}_0 = \mathbf{H}_2$ which is

$$\mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

To appreciate the concept of matrix factoring and becoming signal flow graph, (4.9) can be rewritten in the matrix form as follows:

$$\begin{aligned}\mathbf{P}_b \mathbf{X}_8 &= \frac{1}{8} \begin{bmatrix} \mathbf{H}_2 & & & \\ & \mathbf{H}_2 & & \\ & & \mathbf{H}_2 & \\ & & & \mathbf{H}_2 \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & & & \\ & \overline{\mathbf{S}}_2 & & \\ & & \mathbf{I}_2 & \\ & & & \overline{\mathbf{S}}_2 \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & \mathbf{I}_2 & & \\ \mathbf{I}_2 & -\mathbf{I}_2 & & \\ & & \mathbf{I}_2 & \mathbf{I}_2 \\ & & \mathbf{I}_2 & -\mathbf{I}_2 \end{bmatrix} \\ &\cdot \begin{bmatrix} \mathbf{I}_4 & \\ & \overline{\mathbf{S}}_4 \end{bmatrix} \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{I}_4 & -\mathbf{I}_4 \end{bmatrix} \mathbf{x}_8\end{aligned}\quad (4.10)$$

where

$$\overline{\mathbf{S}}_4 = \begin{bmatrix} \mathbf{I}_2 & \\ & -j\mathbf{I}_2 \end{bmatrix}, \text{ and } \overline{\mathbf{S}}_2 = \begin{bmatrix} 1 & 0 \\ 0 & -j \end{bmatrix}.$$

Figure 4.2 shows the signal flow graph of fast SCHT algorithm based on the DIS decomposition as in (4.10) for $N = 8$. Multiplication by $(1/8)$ can be performed by a binary shifting operation without requiring any arithmetic operation. For simplicity, this scaling factor is excluded in the signal flow graph. As shown in Figure 4.2, the fast algorithm decomposes an 8-point SCHT into three stages and each stage re-

Figure 4.2: The signal flow graph of the DIS fast SCHT algorithm ($N = 8$).

quires eight addition/subtractions. It is easy to conclude that for an N -point SCHT, the DIS decomposition contains $n = \log_2 N$ stages and each stage performs N addition/subtractions. Similar to the DIT fast algorithm reported in Subsection 3.2.5, the presented DIS fast SCHT algorithm also significantly reduces the computational complexity. In general, a fast algorithm for an N -point SCHT requires $N \log_2 N$ complex addition/subtractions and $(N/4) \log_2(N/2)$ complex multiplications for the trivial twiddle factors. Compared to the direct implementation of the SCHT, the fast algorithm reduces the number of complex addition/subtractions by $(N - 1)/\log_2 N$ times.

4.3 Implementation

Based on the DIS fast SCHT algorithm, Figure 4.3 illustrates a pipelined hardware structure that is particularly suited to bit-serial word sequential input/output data streams. While the input is ordered in an increasing manner, the output is in a bit-reversed order as shown in Figure 4.2. For a full hardware utilization, each stage uses only one adder/subtractor within the N word cycles during which the input data

Figure 4.3: Pipelined hardware structure for an N -point SHT based on the DIS fast algorithm.

are sequentially collected. For such an implementation, the data must be available to the inputs of the adder/subtractor with a correct timing. As observed in Figure 4.3, the pipeline has n stages and each stage includes a delay and commutator (DAC) unit for distribution of input data, and a modified adder/subtractor (MAS) for addition/subtractions and the multiplication with the trivial twiddle factors, i.e., $-j$. The subsequent subsections will explain the DAC unit and the MAS in detail.

4.3.1 Delay and Commutator Unit

One observation from Figure 4.2 is that the input indices of each addition/subtraction are separated by $N_i = (N/2^i)$ word cycles, where $i = 1, 2, \dots, n$ represents the stage index. For an easy description, let us consider the distribution of input data in the first stage of Figure 4.2. In this case, a pair of sequences $x(m)$ and $x(m + N_1)$ for $m = 0, 1, 2, 3$ and $N_1 = 4$, are needed to compute the N_1 complex additions in the first N_1 cycles. Within the next N_1 cycles, the same data sequences are used for the computation of N_1 subtractions.

The details of the DAC unit for the stage i of the pipelined structure are shown in Figure 4.4. It consists of two shift registers, each having the N_i word delays. The appropriate data are selected by using the $2 : 1$ complex multiplexers with a control

signal c_1 . The timing and operations of the DAC unit at stage 1 for an 8-point SCHT are shown in Figure 4.5. In the figure, m represents the arrival index of the input data sequence with the assumption that $x(0)$ is the first data available at the input of the DAC unit, and each input data occupies one word cycle. The eight input data are sequentially shifted into the two shift registers within the first eight word cycles. During the cycles $m = 4$ to 7, the 2 : 1 complex multiplexers are at position 0 so that the pair of data sequences $x(m)$ and $x(m + 4)$ are available at the two outputs of the DAC unit. In the next four cycles, the 2 : 1 complex multiplexers are at position 1 so that the same output sequences are obtained. At the same time, the input sequence for the next transform is shifted into the first shift register. During the next four cycles, that is, starting from $m = 12$, the output sequences for the next transform are provided. Hence, we do not need any extra cycles for resetting operations of subsequent SCHTs, and successive SCHTs can be accomplished continuously.

When the transform size is large, for example, $N > 64$, the use of shift registers

Figure 4.4: Implementation of the DAC unit at stage i : (a) Implementation with shift registers, and (b) implementation with RAMs.

Figure 4.5: Timing and operations of the delay and commutator unit at stage 1 ($N = 8$).

to implement the DAC unit may require too much chip area and power consumption since the data are actively shifted all the time. One alternative implementation is to use RAMs to replace the shift registers, as shown in Figure 4.4(b), particularly for the front stages of the pipeline. Let us consider the stage 1 of the pipelined structure for a length- N input sequence, and also assume that the RAMs can write and read once in each word cycle, respectively. The N input data are written into the two RAMs sequentially with the first N_1 data in RAM 1 and the next N_1 data in RAM 2. From the cycle N_1 , the 2 : 1 complex multiplexer is at position 0 so that $x(m)$ and $x(m + N_1)$, where $m = 0, 1, \dots, N_1 - 1$, are available at the two outputs of Figure 4.4(b), respectively. In the next N_1 cycles, the 2 : 1 complex multiplexer is at position 1 to obtain the same data sequence as that in the previous N_1 cycles. At the same time, the first N_1 input data for the next transform are sequentially written into RAM 1 after the data for the current transform are sequentially read out for computation.

4.3.2 Modified Adder/Subtractor

The modified adder/subtractor (MAS), which is shown in Figure 4.6, is to perform the complex addition/subtractions for the butterfly elements in the signal flow graph. With the 2 : 1 complex multiplexers at its input and output, the multiplication with the trivial twiddle factor, $-j$, can also be accomplished. Because the computation of $-j[(a + jb) - (c + jd)]$ (refer to Figure 4.2) is the same as that of $(b - d) + j(c - a)$, swapping and subtraction operations are jointly used to achieve the required operation. Therefore, swapping operations at the input and the output are needed only when subtraction and multiplication with the trivial twiddle factor are performed. Since, the required data pairs are available sequentially in terms of word cycles for computation, the MAS produces N outputs within N word cycles. Therefore, a full hardware utilization is achieved.

At the next stage of computation, an N -point SCHT is decomposed into two $(N/2)$ -point SCHTs with the same input sequence order, as shown in Figure 4.2. It means that for the next stage, the same operations as those for the first stage are needed, except that the size of the shift registers (or RAMs) is reduced to be $(N/4)$. This extension can be continued until the last stage which deals with the two-point

Figure 4.6: Modified adder/subtractor.

SCHTs.

Each stage of the proposed structure requires simple control signals that are c_1 and c_2 . As an example, Figure 4.7 shows the waveforms of these control signals for the first stage of the pipelined structure. The signal c_1 is needed for the control of $2 : 1$ complex multiplexers in the DAC unit, and for the addition/subtractions in the MAS. Therefore, it has the values of 1 and 0 alternatively within a period of N as shown in the figure. The control signal c_2 is used for the swapping operations in the MAS. The control signals for the other stages can be similarly formulated as far as the period of c_1 and the duration of c_2 are reduced appropriately. For example, the period of c_1 should be $(N/2)$ while the duration of c_2 is $(N/8)$ if it is used for the second stage of the pipelined structure.

Figure 4.7: Waveforms of the control signals for the stage 1.

4.4 Discussion

The proposed pipelined hardware structure is very regular and requires only simple control signals at each stage. The number of complex data stores in the DAC unit for the stage i is $2N_i$ as shown in Figure 4.4 where $N_i = (N/2^i)$ with $i = 1, 2, \dots, n$. The

total number of complex data stores in the whole pipeline is shown to be

$$\begin{aligned}
D_{\text{total}} &= 2 \sum_{i=1}^n \frac{N}{2^i} \\
&= N \left[1 + \frac{1}{2} + \frac{1}{2^2} + \cdots + \frac{1}{2^{n-1}} \right] \\
&= 2N \left[1 - \left(\frac{1}{2} \right)^n \right] \\
&= 2(N - 1)
\end{aligned} \tag{4.11}$$

where $N = 2^n$. Hence, the entire pipeline needs $2(N - 1)$ complex data stores in the DAC units and $\log_2 N$ complex modified adder/subtractors. In general, four 2 : 1 complex multipliers are needed for the DAC and MAS used in each stage. The MAS at the last stage, however, does not need any 2 : 1 complex multiplexer as the last stage does not require any trivial multiplications by the factor $-j$ (see Figure 4.2). Therefore, the entire pipeline needs $4(\log_2 N) - 2$ units of 2 : 1 complex multiplexers.

It can easily be seen that the entire pipelined structure achieves a full hardware utilization for the computation of arithmetic operations. Besides, it is able to deal with the continuous data streams without requiring any extra word cycles for resetting operations between any two consecutive SHTs. It is also found that the proposed structure has a processing latency of $(N - 1)$ cycles + $(\log_2 N)T_{\text{MAS}}$, where T_{MAS} is the processing latency for the MAS. It is also possible to use the delay feedback shift registers [44] in the pipelined structure. In comparison, the delay feedback method can achieve a saving of $(N - 1)$ complex data stores, however, it requires $(\log_2 N)$ more complex adder/subtractors.

4.5 Conclusion

This chapter derives the decimation-in-sequence fast SCHAT algorithm (which is analogous to the decimation-in-frequency radix-2 FFT decomposition method) based on the decimation-in-time fast SCHAT algorithm to compute an N -point SCHAT. Similar to the DIT fast algorithm, it is found that the DIS fast SCHAT algorithm also reduces the computational cost significantly as compared to a direct computation. By making use of the concept of the DIS fast decomposition method, a pipelined hardware structure for the implementation of the SCHAT is presented to directly associate with sequentially presented input/output data streams to support high-speed real-time operations of the SCHAT. It is observed that this pipelined architecture can operate in real-time and able to achieve a 100% hardware utilization. Besides, the proposed structure avoids using any complex multipliers for the twiddle factors. The modified adder/subtractor is proposed to accomplish the tasks for the multiplications with the trivial twiddle factors, where only the swapping and subtraction operations are necessary to compute those operations. The delay and commutator unit is introduced for the distribution of input data to get the correct pairs for computation with the simple control signals. Another prompt plus point is that no extra word cycles are needed for resetting the operations between any two successive SCHAT computations.

This pipelined structure is also well suited to implement the NCHAT since the signal flow graph for fast forward NCHAT decomposition (see Figure 3.9 and Subsection 3.3.3) is similar to that of the DIS fast SCHAT algorithm shown in Figure 4.2 except that the latter needs the bit reversal ordering of the output data sequence.

Chapter 5

Application of the SCHAT in DS-CDMA Systems

5.1 Introduction

Direct sequence spread spectrum multiple access (DS-SSMA) is an attractive multiple access technique which allows dynamic sharing of a common pool of communication channels by a large number of users. The signal from each user is spread over a very wide bandwidth by using a specific codeword called a spreading sequence. One of the most significant limitations in DS-SSMA systems is the interference caused by competing and simultaneous users, called *multiple access interference* (MAI), due to common sharing of all the users within the same bandwidth. The MAI, which is mainly determined by the spreading codes, reduces the channel capacity and leads to degraded performance resulting in poor bit-error-rate (BER) for the individual receivers [45]. The orthogonal spreading sequences have ideally zero crosscorrelation values for zero delays, which is greatly helpful to suppress the MAI and leads to improved bandwidth efficiency of a SSMA system. Therefore, there has been great interest to employ the orthogonal spreading sequences in DS-SSMA systems. For example, the orthogonal

Walsh Hadamard transform (WHT) sequences are used in the downlink of wideband code division multiple access (WCDMA) system (3G mobile communications) as the spreading sequences to distinguish the different users within the same cell.

The results in [28] indicated that the complex 4-phase sequences can provide a 3-dB performance improvement as compared to the binary Gold sequences in the maximum periodic correlation parameters. As such it is more efficient to use the 4-phase sequences over the real binary sequences when dealing with quadrature-phase modulation. Therefore, the use of complex spreading sequences in DS-CDMA systems has motivated a lot of research interests. New algorithm for designing the overloaded complex spreading sequence sets is proposed in [46] for the uplink of multicell synchronous CDMA (S-CDMA) system. The algorithm is based on minimizing the total weighted squared correlation (TWSC), which is used to measure all the interferences in S-CDMA systems. This algorithm benefits the S-CDMA systems by enhancing the user capacity. However, they are nonorthogonal complex spreading sequences.

In [14,47], a set of orthogonal complex sequences named unified complex Hadamard transform (UCHT) sequences which are derived from the row vectors of a UCHT matrix [10] is introduced in DS-CDMA systems. Therein, their BER performance is analyzed in both synchronous and asynchronous DS-CDMA systems over the additive white Gaussian noise (AWGN) channels together with the investigation on their correlation parameters. It is found that the UCHT sequences perform better than the other well-known sequences, such as the Gold, small set of Kasami and m -sequences. But as the UCHT is generated based on the WHT, their BER performance does not outperform the binary real orthogonal WHT sequences in DS-CDMA systems. This motivates the research in finding the sequences which may be better than the WHT sequences in CDMA communication system. This chapter proposes a new set of orthogonal complex spreading sequences called the SCHAT sequences which are derived from sequency-ordered complex Hadamard transform in DS-CDMA systems.

The rest of the chapter is organized as follows:

- Section 5.2 reviews the spread spectrum multiple access techniques especially focusing on the DS-CDMA systems and various existing spreading sequences which are employed in the DS-CDMA systems.
- Section 5.3 mentions about the proposed SCHAT spreading sequence which is to be used in DS-CDMA systems.
- Section 5.4 provides the performance measurement parameters to analyze the performance of a sequence set in synchronous and asynchronous CDMA systems.
- Section 5.5 investigates the BER performance of the proposed sequence in the synchronous DS-CDMA system over the AWGN channel. The simulation results and the performance comparisons with other well-known spreading sequences are discussed in detail in that section.
- The same is applied to Section 5.6 for the asynchronous DS-CDMA system over the multipath fading channel.
- Finally, this chapter is concluded in Section 5.7.

5.2 Literature Study

In this section, we will review the concepts of spread spectrum techniques, various multiple access methods and the existing spreading sequences found in the literature.

5.2.1 Spread Spectrum Techniques

Spread spectrum technique is a means of transmission in which the data occupies a larger bandwidth than necessary. The spreading of the bandwidth is accomplished

before the transmission through the use of a code which is determined by the spreading waveform. The same code is applied to demodulate the data at the receiving end. The advantage of spread spectrum is that many users can simultaneously use the same bandwidth without significant interference with each other by choosing the suitable spreading sequences assigned for the different users. Hence, spread spectrum systems become very bandwidth efficient in the multiuser environment [48].

There are three major multiple access techniques. They are frequency division multiple access (FDMA), time division multiple access (TDMA) and code division multiple access (CDMA). Out of these three techniques, CDMA uses the concept of *spread spectrum* which employs transmission of information signal over a radio channel that is several orders of magnitude greater than necessary to transmit the original signal. Because the signal from one user is spread over a very wide bandwidth using a signature sequence, the interference caused by other users within that bandwidth is minimized by the de-spreading process using the same spreading sequence assigned to the transmitted user. This allows the multiple users to share the same radio channel simultaneously. In the receiving end, the received signal is correlated with a synchronously generated replica of the spreading code to recover the original information-bearing signal. This implies that the receiver must know the spreading code to recover the user data. If not, it will only receive a noise-like signal. Therefore, CDMA was first introduced for the military applications due to its secured and undetectable design, and its robustness against intentional/unintentional jamming.

There exists a number of modulation techniques that generate the spread spectrum signals. They are as follows:

- Direct sequence spread spectrum (DS-SS): The information-bearing signal is directly multiplied by a fast code signal (which may be pseudo-noise (PN) code sequence) for the spreading operation.

- Frequency-hopped spread spectrum (FH-SS): The carrier frequency which conveys the information-bearing signal is rapidly changed according to the code signal.
- Time-hopped spread spectrum (TH-SS): The information-bearing signal is not transmitted continuously. Instead, the signal is transmitted in short bursts where the timings of the bursts are decided by the code signal.
- Chirp modulated spread spectrum: It is a spread spectrum technique that uses the wideband linear frequency modulated chirp pulses to encode the information signal. A chirp is a sinusoidal signal whose frequency increases or decreases linearly over a certain amount of time.
- Hybrid modulated spread spectrum: Two or more of the abovementioned spread spectrum techniques can be used to combine their advantages and to combat their disadvantages.

Among them, we shall focus on using the DS-SS protocol which is to be detailed in the following subsection.

5.2.2 Direct Sequence Spread Spectrum Technique

Direct sequence is the best known spread spectrum technique. As mentioned earlier, in the DS-SS system, the data signal from each user is spread over a very large bandwidth by multiplying the data signal with a PN code (which is produced by a PN code generator). A single pulse or a symbol of the spreading waveform is called a chip. Actually, a PN code is a sequence of chips which have the values ± 1 (polar) and 0 or 1 (non-polar). The main parameter in spread spectrum analysis is the *processing gain*, which is the ratio of the transmission bandwidth over the information bandwidth, that

is,

$$G_p = \frac{BW_t}{BW_i} \quad (5.1)$$

where BW_t and BW_i represent the transmission and information bandwidths respectively. G_p is sometimes referred to as the *spreading factor*. The processing gain determines the number of users which can be allowed in a system as well as the amount of multipath effect reduction and the difficulty to jam or detect a signal. Therefore, it is advantageous to have a processing gain as high as possible for spread spectrum systems. In a DS-SS system, the length of the spreading code is the same as the processing gain or the spreading factor. Figure 5.1 shows a function block diagram of a DS-SS system with binary phase modulation. As shown in Figure 5.1(a), the binary

Figure 5.1: The block diagram of a DS-SS system model with binary phase modulation: (a) Transmitter, and (b) receiver.

data is multiplied with the PN code sequence and then it is modulated by the binary phase-shift-keying (BPSK) scheme to transmit the signal into the wireless channel. At the receiver which is shown in Figure 5.1(b), it uses coherent BPSK demodulation to recover the original signal. Assuming that code synchronization has been achieved at the receiver, the received DS-SS signal is multiplied by a local replica of the PN code sequence. Synchronization must be accomplished at the beginning of the reception and maintained until the whole signal is received, then the original signal is recovered by using the corresponding demodulation.

5.2.3 Performance of a DS-CDMA System

In digital communication systems, one of the most common performance measurements is the bit-error-rate (BER) that the individual user will experience. The BER performance of a communication system in a multiple access environment is dependent on the modulation techniques used and the amount of interference incurred by the individual receiver, and constantly it is closely related to the signal-to-noise ratio (SNR) available at the receiver. In this subsection, we will mention the BER performance of a DS-CDMA system over the additive white Gaussian noise (AWGN) channel. In fact, a DS-CDMA system is an extension of a DS-SS system, which employs the spread spectrum technique incorporated with the multiple access method. Figure 5.2 shows the DS-CDMA system model with multiple users over the AWGN channel [48].

Assuming that each user has a PN sequence with N chips per data symbol period (which is denoted as T) such that $NT_c = T$ where T_c is the chip period. The transmitted signal of k th user can be expressed as

$$s_k(t) = \sqrt{2P_k}b_k(t)a_k(t)\cos(2\pi f_c t + \phi_k) \quad (5.2)$$

where $b_k(t)$ is the binary data sequence from User k , $a_k(t)$ is the spreading (or chip)

Figure 5.2: A simplified diagram for a DS-CDMA system with K mobile users: (a) Transmitter model of K users in a CDMA spread spectrum system, and (b) receiver structure for User 0.

sequence for User k , P_k is the received power at the base station from User k , f_c is the carrier frequency and ϕ_k is the carrier phase offset of User k relative to the referenced User 0. As illustrated in Figure 5.2(a), the received signal consists of the sum of K different transmitted signals (one desired user and $K - 1$ undesired users) plus additive white Gaussian noise (AWGN), which can be expressed as

$$r(t) = \sum_{k=0}^{K-1} s_k(t - \tau_k) + n(t) \quad (5.3)$$

where K is the total number of users in the system, τ_k is the time delay parameter of User K relative to the referenced User 0 and $n(t)$ is the additive white Gaussian noise.

Reception is accomplished by correlating the received signal with the appropriate signature sequence to produce a decision variable. The decision statistic variable for i th transmitted bit of User 0 can be obtained as follows:

$$Y_i^{(0)} = \int_{(i-1)T+\tau_0}^{iT+\tau_0} r(t)a_0(t-\tau_0) \cos(2\pi f_c(t-\tau_0) + \phi_0) dt. \quad (5.4)$$

If the DS-CDMA system is completely synchronized, the time delay parameter τ_k in (5.3) can be ignored. This will result in a synchronous DS-CDMA system in which all the user signals are properly aligned/synchronized and combined before being transmitted. It represents a downlink (hub-to-user) mobile communication channel. But this is not feasible for the uplink (user-to-hub) channel as the mobile transmitters are not time-synchronized (any transmitter can transmit at any time), which leads to an asynchronous DS-CDMA system.

From (5.4), we can conclude that if $b_{0,i} = -1$, then the bit will receive in error if $Y_i^{(0)} > 0$. The probability of error can now be calculated as $Pr\{Y_i^{(0)} > 0 \mid b_{0,i} = -1\}$. Since the received signal $r(t)$ is a linear combination of user signals plus the AWGN noise, (5.4) can be rewritten as

$$Y_i^{(0)} = I_0 + \sum_{k=1}^{K-1} I_k + \eta \quad (5.5)$$

where I_0 is the desired contribution to the decision statistic Y from User 0, that is

$$I_0 = \int_0^T s_0(t)a_0(t) \cos(2\pi f_c t) dt, \quad (5.6)$$

and η is a Gaussian random variable representing the noise with zero mean and vari-

ance σ^2 ,

$$\sigma^2 = E[\eta^2] = \frac{N_0 T}{4} \quad (5.7)$$

where $(N_0/2)$ is the two-sided noise power spectral density, T is the period of the data symbol. For detailed derivation of σ^2 , the readers may refer to Appendix E of [48].

And

$$I_k = \int_0^T s_k(t - \tau_k) a_0(t) \cos(2\pi f_c t) dt \quad (5.8)$$

represents the interference created by User k . As described above, we can investigate the BER performance of a DS-CDMA system by doing a Monte Carlo computer simulation.

5.2.4 Spreading Sequences for DS-CDMA Systems

We can see from (5.8) that the interference caused by one user upon the desired user is dependent on the crosscorrelation of the two spreading sequences used to assign the corresponding users. Hence, the characteristics of the spreading codes provide a crucial effect on the performance of a DS-CDMA system and much care should be taken on the selection of appropriate signature sequences to suppress the MAI effectively. In particular, it is desirable to have the spreading sequences with low crosscorrelation values in the multiuser environment for a DS-CDMA system. In this subsection, we shall present some of the well-known spreading sequences that exist for practical DS-SS systems as well as in the literature. They are the maximum-length sequences (m -sequences), the Gold codes, the Kasami codes, the WHT sequences and the UCHT sequences.

m-Sequences

One of the most important classes of PN sequences is the binary maximum-length sequences or *m*-sequences for short. They are non-orthogonal PN sequences and can be generated using the *m*-stage linear feedback shift registers and the modulo-2 adders (XOR gates). Figure 5.3 shows the design of such a sequence. The maximum length of such PN sequences is determined by the size of the shift register, and it is normally $N = 2^m - 1$ bits generated as demonstrated in the figure. The feedback connection is obtained by a primitive polynomial. The sequence is periodic with the period N and each period has a sequence of 2^{m-1} ones and $(2^{m-1} - 1)$ zeros. The autocorrelation values of *m*-sequence is as follows: [49]

$$\theta_{ac}(l) = \begin{cases} N, & l = 0 \\ -1, & l = 1, 2, \dots, N - 1. \end{cases} \quad (5.9)$$

where l is the amount of offset indicated by integers. The *m*-sequence offers a good autocorrelation property. That is, the *m*-sequence occupies the impulse-like periodic autocorrelation characteristics, which make it ideal for the synchronization purpose. But it has poor crosscorrelation property, that is, there are many large-valued peaks

Figure 5.3: Generation of *m*-sequence using the *m*-stage shift registers with linear feedback.

instead of the ideal zero values for all shifts.

Gold Sequences

Gold sequences are another important class of binary PN sequences which can provide a larger set of sequences with uniform and bounded crosscorrelation values. Therefore, they are preferred in practical due to vast number of codes they can supply by achieving the goal of multiple access systems to accommodate as many users as possible. They can be constructed by the modulo-2 addition of two selected m -sequences known as the *preferred pair* as shown in Figure 5.4. $f(x)$ in the figure represents the primitive polynomial to generate the two m -sequences. The crosscorrelation values of a preferred pair of two m -sequences are -1 , $-t(m)$ and $t(m) - 2$ where

$$t(m) = \begin{cases} 1 + 2^{(m+1)/2}, & m = \text{odd} \\ 1 + 2^{(m+2)/2}, & m = \text{even}. \end{cases} \quad (5.10)$$

The period of the Gold sequences is $N = 2^m - 1$ and there are $2^m + 1 = N + 2$ Gold sequences in the same set. The crosscorrelation values of other Gold sequences in the same set are the same as that of the preferred pair. Hence, the Gold sequences have

Figure 5.4: Generation of a set of Gold sequences for $m = 5$.

only three crosscorrelation values for all values of shifts (the amount of offset between two sequences). Except for the preferred pair, the off-peak autocorrelation of the Gold sequences also has three values which are the same as the three crosscorrelation values. The preferred pair has the off-peak autocorrelation values of -1 since they are the m -sequences.

Kasami Sequences

Kasami sequences are another class of PN sequences which can be obtained by decimating the m -sequences. In the generation of Kasami sequences, every $(2^{(m/2)} + 1)$ -th bit of an m -sequence is selected [49]. This method of construction yields a smaller set of PN sequences as compared to the Gold sequences, but their crosscorrelation is better than that of the Gold sequences, which is $\theta_{cc,max} = \sqrt{N}$ where N is the period of the sequence.

WHT Sequences

All of the aforementioned sequences are the non-orthogonal sequences but the WHT sequences are the orthogonal real spreading sequences which have ideally zero crosscorrelation values for zero delays. Therefore, they are more suited to DS-CDMA system in multiple access environment. Such sequences are constructed based on the row vectors of the Walsh Hadamard matrix, hence, they can provide a large number of codes available to assign the users. Besides, the WHT matrices are easy to generate by using a simple Kronecker product as mentioned in Section 2.3. The WHT sequence has been considered as an ideal candidate in the downlink of IS-95 communication system, and the current generation of wideband CDMA system (3G mobile communications) also uses the WHT codes to separate the user signals within the same cell.

UCHT Sequences

As contrast to the WHT sequence, the UCHT sequences are the orthogonal complex spreading sequences which consist of the elements $\{\pm 1, \pm j\}$ and they are introduced in DS-SSMA systems [14, 32, 47]. They are derived from the row vectors of the UCHT matrix, hence, they are called the UCHT sequences. The generation of the UCHT sequences is as follows [14].

Let $h(p, k)$ be an element of any UCHT matrix at p th row and k th column, where $p, k = 0, 1, \dots, N - 1$ and $N = 2^n$. Then

$$h(p, k) = \prod_{r=0}^{n-1} h_r \quad (5.11)$$

where

$$h_r = \tau_1 + (\tau_2 - \tau_1)p_r + \tau_1(\tau_3 - 1)q_r + (\tau_1 - \tau_1\tau_3 - \tau_2 - \tau_2\tau_3)p_rq_r, \quad (5.12)$$

$(p_{n-1}, p_{n-2}, \dots, p_0)$ and $(k_{n-1}, k_{n-2}, \dots, k_0)$ denote the binary representations of the decimals p and k respectively, and τ_1, τ_2, τ_3 can be any elements from a set $C_4 = \{\pm 1, \pm j\}$. In this way, all the elements of any UCHT matrix of dimension N can be generated and each row vector can be used as an orthogonal complex spreading sequence to identify a particular user in DS-CDMA systems. This form of generating the UCHT sequences is useful to derive and analyze various correlation properties of the sequences [14] as the formulation is dealing with the binary values 0 and 1, and the formula can define the sequence element by element. Note that the UCHT matrices can also be constructed based on the WHT and successive Kronecker products as described in Subsection 2.4.3.

Generally, the UCHT matrices can be categorized into two groups comprising of 32 matrices each based on the existence of the half spectrum property (HSP). They

are the HSP-UCHT matrices and the NHSP-UCHT matrices where the former show the HSP and the latter do not possess the HSP. Similarly, the UCHT sequences can also be classified into two groups of sequences which are the HSP-UCHT sequences and the NHSP-UCHT sequences.

5.3 A Proposed Spreading Sequence for DS-CDMA Systems

As mentioned earlier, in a DS-CDMA system, each user is assigned a specific spreading sequence to spread the signal over an assigned frequency band. To suppress the MAI effectively, the CDMA spreading sequences need to have very low crosscorrelation values. For this purpose, the SCHAT sequences are introduced as the complex spreading sequences in DS-CDMA systems for the first time. Similar to the WHT and UCHT sequences, the SCHAT sequences are also derived from the row vectors of an SCHAT matrix. Therefore, they can provide a large set of spreading sequences for assigning to the different users in the DS-CDMA system. Besides, the SCHAT matrices can simply be generated based on the complex Rademacher matrices (as shown in Subsection 3.2.2) so the SCHAT sequences can be obtained easily for use. It is noted that the SCHAT sequences can also be derived from the row vectors of an NCHAT matrix since both SCHAT and NCHAT matrices have the same row vectors except that the way they are positioned in the matrices is different. In this section, we will define the SCHAT sequences using the exponential formula of the SCHAT.

Let $x_{p,k}$ be the element of any SCHAT sequence of length N where $N = 2^n$, p is the sequence's order of one SCHAT set (or matrix) and k is the element's order of the sequence. Then,

$$x_{p,k} = \prod_{r=0}^{n-1} (-1)^{\tilde{p}_r k_r} (j)^{\hat{p}_r k_r} \quad (5.13)$$

where $n = \log_2 N$, N is the length of the sequence and $(k_{n-1}, k_{n-2}, \dots, k_0)$ denotes the binary representation of the decimal k . $\tilde{p}_r$ and $\hat{p}_r$ are the binary representations as defined in Subsection 3.2.3 where $r = 0, 1, \dots, n-1$ are the indices of binary bit positions and $j = \sqrt{-1}$. The SCHAT sequences are orthogonal to each other as they are derived from the orthogonal SCHAT matrices and they have zero periodic crosscorrelation value for zero delay. For a DS-CDMA transmitter, each data bit can be spread by an SCHAT sequence. The processing gain is thus equal to N .

Let us consider $b_k(t)$ to be the transmitted binary bit sequence from User k in a DS-CDMA system and see how the intended signal can be recovered from the mobile station of User 0 using the SCHAT sequences. And also let us assume that there are K users in the system, then the transmitted spread signal can be expressed as

$$s_{total}(t) = \sum_{k=0}^{K-1} \sqrt{2P_k} b_k(t) a_k(t) \quad (5.14)$$

where $a_k(t)$ is the SCHAT spreading sequence of User k and P_k is the transmitted power of User k . At the mobile station of User 0, the receiver correlates the incoming signal $s_{total}(t)$ with the conjugate of its assigned SCHAT sequence $a_0^*(t)$ as shown in Figure 5.5.

According to the characteristics of the SCHAT sequences, the i th transmitted bit

Figure 5.5: A DS-CDMA receiver using the SCHAT spreading sequence.

for User 0 can be recovered by integrating over the one bit period T as

$$\begin{aligned} \int_{(i-1)T}^{iT} s_{total}(t) a_0^*(t) dt &= \int_{(i-1)T}^{iT} \sqrt{2P_0} b_0^i(t) a_0(t) a_0^*(t) dt \\ &= \begin{cases} +\sqrt{2P_0} NT, & \text{if } b_0^i = +1 \\ -\sqrt{2P_0} NT, & \text{if } b_0^i = -1. \end{cases} \end{aligned} \quad (5.15)$$

5.4 Performance Measurement Parameters of the Sequence Set

In DS-CDMA systems, it leads to the dual requirement of minimal autocorrelation (AC) values to achieve the wideband sequence characteristics, and minimal crosscorrelation (CC) values among the spreading sequences to reduce the MAI. The desirable AC and CC properties may be well viewed as mutual exclusive criterion and at best, a compromise must be sought when selecting the sequence set for the DS-CDMA system. In this section, we shall mention the parameters to measure the performance of the sequence set in a DS-CDMA system. They are periodic mean-squared (MS) correlation and aperiodic MS correlation parameters.

5.4.1 Periodic Mean-Squared Correlation

The periodic correlation function is one of the most important indicators to measure the performance of a sequence set in a DS-CDMA system especially when the user code sequences are being transmitted repetitively. Let $\theta_{x,y}$ be the periodic crosscorrelation function for any pair of the sequences x and y of length N , then it can be defined as

$$\theta_{x,y}(l) = \sum_{k=0}^{N-1} x_{p,k} y_{q,k+l}^* \quad (5.16)$$

where $\{ \}^*$ represents the complex conjugate operation, p and q are the row's orders in one SCHAT sequence set (or matrix), k is the element's order of the sequence, and $l = 0, 1, \dots, N-1$. If $x = y$, it will become the periodic autocorrelation function which is denoted as θ_x . Based on these functions, the periodic MS-CC (φ_{cc}) and MS-AC (φ_{ac}) parameters are defined as follows: [47]

$$\varphi_{cc} = \frac{1}{K(K-1)N^2} \sum_{\substack{x=1 \\ x \neq y}}^K \sum_{y=1}^K \sum_{l=0}^{N-1} |\theta_{x,y}(l)|^2 \quad (5.17)$$

$$\varphi_{ac} = \frac{1}{KN^2} \sum_{x=1}^K \sum_{l=1}^{N-1} |\theta_x(l)|^2 \quad (5.18)$$

which can be considered as a key parameter to evaluate the average performance of the sequence set in the synchronous DS-CDMA system, where K is the size of the subset of the spreading sequences.

To analyze the performance of the SCHAT sequence set in the synchronous DS-CDMA system, the values of φ_{cc} and φ_{ac} are first computed for the subsets of sequences. To select the subsets of sequences (with different sizes) to be employed in a CDMA system, we utilize the selection procedure based on the algorithm shown in Figure 5.6 [14]. Firstly, the sum of the periodic CC values between any two sequences in each subset is computed. In order to reduce the MAI effectively, the subsets with smaller overall sum are selected for computer simulation (which is to be explained in the next Section 5.5). By using this selection criterion, some subsets with different sizes of the SCHAT, HSP-UCHT, NHSP-UCHT, WHT, Gold, small set of Kasami and m -sequences are chosen for computation. The periodic MS-CC and MS-AC values of various sequences for different subset sizes (which are denoted as K) are tabulated in Tables 5.1 and 5.2. The size of the subset represents the user capacity in the system. It can be observed from the Tables that the SCHAT sequences produce very low

Figure 5.6: Algorithm for subsection selection of the sequences.

crosscorrelation values as compared with other spreading sequences especially when the number of users increases in the system. But their periodic autocorrelation values are the highest among different sequences. Therefore, they are more suitable to be employed in the multiple access environment to minimize the MAI but not over the multipath fading channel in the synchronous CDMA system.

Family	SHT		WHT		HSP-UCHT		NHSP-UCHT	
K	φ_{cc}	φ_{ac}	φ_{cc}	φ_{ac}	φ_{cc}	φ_{ac}	φ_{cc}	φ_{ac}
6	1.0667	31.1250	4.8000	13.8125	2.0063	7.2177	4.8000	13.8125
8	1.1423	30.3438	4.0000	12.9844	1.7188	7.2188	4.0000	12.9844
10	1.1556	29.7750	3.0667	12.4875	1.2229	7.2188	3.0667	12.4875
12	1.0909	29.3125	2.4848	12.1563	1.1013	7.2188	2.4848	12.1563
14	0.8462	28.7858	2.0495	11.8750	1.0999	7.1295	2.0495	11.8750
16	0.6833	28.2188	1.7417	11.6250	1.0870	6.9844	1.7417	11.6250
18	0.5441	27.6528	1.5074	11.4306	1.0300	6.8715	1.5074	11.4306
20	0.4658	27.0875	1.4118	11.2750	0.9658	6.7813	1.4118	11.2750

Table 5.1: Periodic MS correlation parameters for the SHT, UCHT and WHT sequences with the length of 64.

Family	K	φ_{cc}	φ_{ac}
Gold	6	1.1699	1.0476
	8	1.0355	1.0476
	10	0.9889	1.0476
	12	0.9591	1.0476
	14	0.9354	1.0476
	16	0.9251	1.0476
	18	0.9399	0.9903
	20	0.9535	0.944
m	6	1.0475	1.2570
Small set of Kasami	8	0.8252	0.7737

Table 5.2: Periodic MS correlation parameters for the Gold, small set of Kasami and m -sequences with the length of 63.

5.4.2 Aperiodic Mean-Squared Correlation

Aperiodic CC and AC functions are the main indicators of performance measurement of the spreading sequences in the asynchronous DS-CDMA system over the multipath fading channel. Therefore, the performance of a set of sequences for this type of DS-CDMA system may be evaluated using the aperiodic MS-CC (R_{cc}) and MS-AC (R_{ac}) parameters. They can be defined as [14, 45]

$$R_{cc} = \frac{1}{K(K-1)N^2} \sum_{\substack{x=1 \\ x \neq y}}^K \sum_{\substack{y=1 \\ y \neq x}}^K \sum_{l=1-N}^{N-1} |C_{x,y}(l)|^2 \quad (5.19)$$

$$R_{ac} = \frac{1}{KN^2} \sum_{x=1}^K \sum_{\substack{l=1-N \\ l \neq 0}}^K |C_x(l)|^2 \quad (5.20)$$

where K is the number of users in the system, N is the length of the sequences, $C_{x,y}$ is the aperiodic CC function of any two sequences in the set

$$C_{x,y}(l) = \begin{cases} \sum_{k=0}^{N-1-l} x_k y_{k+l}^*, & 0 \leq l \leq N-1 \\ \sum_{l=0}^{N-1+l} x_{k-l} y_k^*, & 1-N \leq l < 0 \\ 0, & |l| \geq N \end{cases} \quad (5.21)$$

and $C_x(l)$ is the aperiodic AC function which is defined as the CC function between the complex sequence x and its shifted version. The aperiodic CC function is closely related to the periodic CC function as

$$\theta_{x,y}(l) = C_{x,y}(l) + C_{x,y}(l-N). \quad (5.22)$$

The results on the aperiodic MS-CC and MS-AC parameters of various sequence sets are to be discussed in the subsequent Section 5.6.

5.5 Synchronous DS-CDMA System

Since the downlink is a hub-to-user communication channel, all the user signals can be properly aligned/synchronized and combined before being transmitted. If there exists a dominant signal propagation path from the transmitter to every user without major reflections, then all the users receive a synchronous CDMA signal. Figure 5.7 [48] shows a DS-CDMA system model for K users using the BPSK modulation and orthogonal complex spreading sequences. For the synchronous DS-CDMA signal, the time delay parameters τ_k where $k = 0, 1, \dots, K - 1$ can be neglected and Figure 5.7 will become the model for the synchronous DS-CDMA channel model.

5.5.1 BER Performance

In a synchronous DS-CDMA system using binary signaling with K users assuming equal power for each user, the radio signal transmitted from the mobile station to the k th mobile user can be expressed as

$$s_k(t) = \Re \left\{ \sqrt{2P_k} b_k(t) a_k(t) e^{j\omega_c t + \phi_k} \right\} \quad (5.23)$$

where $b_k(t)$ is the binary data sequence which consists of the elements $\{\pm 1\}$ from User k , $a_k(t)$ is the orthogonal complex spreading sequence for User k , P_k is the received power of User k from the base station, ω_c is the carrier frequency and ϕ_k is the carrier phase offset of User k relative to the referenced User 0. ϕ_k is assumed to be uniformly distributed on $[0, 2\pi]$.

At the receiver corresponding to User 0 (shown in Figure 5.7), the signal available at the input to the correlation receiver (assuming no fading or multipath) in the

Figure 5.7: A DS-CDMA system model for K users using orthogonal complex spreading sequences.

synchronous environment can be written as

$$r(t) = \sum_{k=0}^{K-1} s_k(t) + n(t) \quad (5.24)$$

where K is the number of users in the system and $n(t)$ is the channel noise (AWGN).

Assuming that the phase of the signal is synchronized with User 0 in the receiver,

the decision statistic $Y^{(0)}$ for User 0 is given by

$$Y^{(0)} = \int_0^{T_b} r(t) a_0^*(t) e^{-j\omega_c t} dt. \quad (5.25)$$

Substituting (5.24) into (5.25), $Y^{(0)}$ can be expressed as

$$Y^{(0)} = I_0 + \xi + \eta \quad (5.26)$$

where

$$\begin{aligned} I_0 &= \sqrt{2P_0} \int_0^{T_b} b_0(t) \Re \{ a_0(t) e^{j\omega_c t} \} a_0^*(t) e^{-j\omega_c t} dt \\ &= \sqrt{2P_0} \int_0^{T_b} b_0(t) (\cos^2 \omega_c t + j \sin^2 \omega_c t) dt \\ &\approx \pm \sqrt{\frac{P_0}{2}} T_b \pm j \sqrt{\frac{P_0}{2}} T_b \end{aligned} \quad (5.27)$$

is the useful User 0 signal and

$$\begin{aligned} \eta &= \int_0^{T_b} n(t) a_0^*(t) e^{-j\omega_c t} dt \\ &= \eta_r + j\eta_i \end{aligned} \quad (5.28)$$

is the AWGN random variable with zero mean and variance $\sigma_{\eta_r}^2 = \sigma_{\eta_i}^2 = E[\eta_r^2] \approx \frac{N_0 T_b}{4}$ [48]. ξ represents the multiple access interference (MAI) among the simultaneous

users,

$$\begin{aligned}\xi &= \sum_{k=1}^{K-1} I_k \\ &= \sum_{k=1}^K \sqrt{2P_k} \int_0^{T_b} b_k(t) a_0^*(t) \Re\{a_k(t) e^{j\omega_c t + \phi_k}\} e^{-j\omega_c t} dt,\end{aligned}\quad (5.29)$$

which can be approximated as (see Subsection 5.6.1)

$$\xi \approx T_b \sqrt{\frac{P_0}{2}} \sum_{k=1}^{K-1} (\cos \phi_k + \sin \phi_k) b_k \Re\{\theta_{k,0}(\tau_k = 0)\} \quad (5.30)$$

where $P_0 = P_1 = \dots = P_{K-1}$, $\theta_{k,0}$ is the periodic crosscorrelation between User 0 and User k . The following trigonometric equations are used to simplify the above expressions. They are as follows:

$$\begin{aligned}\cos(\omega_c t + \phi_k) &= \cos(\omega_c t) \cos(\phi_k) - \sin(\omega_c t) \sin(\phi_k) \\ \sin(\omega_c t + \phi_k) &= \sin(\omega_c t) \cos(\phi_k) + \cos(\omega_c t) \sin(\phi_k) \\ \cos^2(\omega_c t) &= \frac{(1 + \cos 2\omega_c t)}{2} \\ \sin^2(\omega_c t) &= \frac{(1 - \cos 2\omega_c t)}{2}.\end{aligned}\quad (5.31)$$

Since all the users signals are properly synchronized for the synchronous CDMA system, theoretically $\theta_{k,0}(\tau_k = 0)$ is zero for the orthogonal spreading sequences, which results in the zero MAI. As shown in Figure 5.7, we consider $\Re\{Y^{(0)}\}$ to make a decision for the output data. Hence, $Y^{(0)}$ in (5.26) can be considered as a Gaussian random variable with mean $\pm \sqrt{\frac{P_0}{2}} T_b$ and variance $\sigma_{\eta_r}^2$, and the probability of error is

given by

$$\begin{aligned}
P_e &= Q\left(\frac{A}{\sigma}\right) \\
&= Q\left(\sqrt{\frac{P_0 T_b^2}{2(\sigma_\eta^2 + \sigma_\xi^2)}}\right) \\
&= Q\left(\sqrt{\frac{4P_0 T_b^2}{2N_0 T_b}}\right) \\
&= Q\left(\sqrt{\frac{2P_0 T_b}{N_0}}\right) \\
&= Q\left(\sqrt{\frac{E_b}{N_0}}\right)
\end{aligned} \tag{5.32}$$

where A is the signal energy in volts, $A = \sqrt{\frac{P_0}{2}T_b}$, $\sigma_\xi^2 = 0$ for the orthogonal spreading sequences, σ_η^2 is the variance of noise, $E_b = 2P_0 T_b$ is the energy per bit, and

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{t^2}{2}} dt \tag{5.33}$$

is the Q -function associated with the Gaussian distribution tail.

5.5.2 Simulation Results

The Monte Carlo simulation is performed to evaluate the BER performances of various spreading sequences in the synchronous DS-CDMA system over the additive white Gaussian noise (AWGN) channel. It can be concluded from the derivation of P_e (see (5.30)) that the orthogonal spreading sequences perform better than non-orthogonal sequences due to their low crosscorrelation. In the simulation, it is assumed that all the users are transmitting the data randomly in the system, and equal transmission power is assumed for each user and $P_0 = P_1 = \dots = P_{K-1}$. The model of synchronous DS-CDMA systems is shown in Figure 5.7 by neglecting the time delay parameters

τ_k , that is, $\tau_k = 0$ where $k = 0, 1, \dots, K - 1$. It is also assumed that the interference is only caused by the other users and the AWGN with zero mean and unit variance.

Figures 5.8 and 5.9 show the BER performances of different sequences in the synchronous CDMA system with the number of users $K = 8$ and $K = 20$, respectively. The corresponding theoretical BER performances are illustrated in Figure 5.10 and Figure 5.11 for $K = 8$ and $K = 20$ and the results are consistent with the simulation results. It can be seen from the figures that the orthogonal spreading sequences outperform the non-orthogonal spreading sequence (i.e, the Gold sequence) in terms of bit-error-rate especially when E_b/N_0 increases in the system. Therefore, providing a particular BER performance in the system, the orthogonal spreading sequences such as the SCHAT, UCHAT and WHT sequences are able to support more users than the Gold sequence. It can also be concluded from Figures 5.8 and 5.9 that the number of users does not influence the BER performances of the orthogonal spreading sequences significantly whereas the bit-error-rate is deteriorated by increasing the user capacity for the Gold sequence. At low E_b/N_0 , the errors are mainly introduced by the AWGN and the advantage of having good CC properties is minimal. When E_b/N_0 increases, the errors are mainly caused by the interference from other users, hence, the effect of good CC properties distinguishes the various spreading sequences as shown in the figures. In the synchronous CDMA system, the BER performances among the orthogonal spreading sequences are not much different.

Figure 5.8: BER performances of various spreading sequences in the synchronous DS-CDMA system ($K = 8$).

Figure 5.9: BER performances of various spreading sequences in the synchronous DS-CDMA system ($K = 20$).

Figure 5.10: Theoretical BER for different spreading sequences in the synchronous DS-CDMA system ($K = 8$).

Figure 5.11: Theoretical BER for different spreading sequences in the synchronous DS-CDMA system ($K = 20$).

5.6 Asynchronous DS-CDMA System

In contrast to the downlink communication channel, the uplink is a user-to-hub communication channel. All the users transmit the data asynchronously and their signals arrive at the mobile base station not only with different time delays, but also with various carrier phases. Such communication link is modeled as an asynchronous DS-CDMA system model as shown in Figure 5.7 [48].

5.6.1 BER Performance over the Multipath Fading Channel

In an asynchronous DS-CDMA system using binary signaling with K users assuming perfect power control, the radio signal transmitted from the k th mobile user to the base station can be expressed as

$$s_k(t - \tau_k) = \Re \left\{ \sqrt{2P_k} b_k(t - \tau_k) a_k(t - \tau_k) e^{j\omega_c t + \phi_k} \right\} \quad (5.34)$$

where $b_k(t)$ is the binary data sequence which consists of the elements $\{\pm 1\}$ from User k , $a_k(t)$ is the spreading sequence for User k , P_k is the received power at the mobile base station from User k , ω_c is the carrier frequency, τ_k is the time delay parameter of User k relative to the referenced User 0, and ϕ_k is the carrier phase offset of User k relative to the referenced User 0. Since, τ_0 and ϕ_0 are the relative terms, we consider these parameters as zero assuming that User 0 is the referenced user.

At the receiver of base station corresponding to the desired User 0 (as shown in Figure 5.7), the signal available at the input to the correlation receiver in the multipath environment can be expressed as

$$r(t) = \sum_{k=0}^{K-1} s_k(t - \tau_k) + \Re \left\{ \frac{\sqrt{2P_0}}{L} \sum_{k=0}^{K-1} \sum_{l=2}^L b_l(t - \tau_k - lT_c) a_k(t - \tau_k - lT_c) e^{j\omega_c t + \theta_{kl}} \right\} + n(t) \quad (5.35)$$

where $P_0 = P_1 = \dots = P_{K-1}$, L is the number of multipaths, $b_l(t)$ is the previous bit sequence in the l th multipath component, T_c is the chip duration, θ_{kl} is the phase delay parameter of the l th multipath component for User k , and $n(t)$ is the AWGN noise with variance σ^2 . The received signal contains both the desired User 0 and $(K - 1)$ undesired users, and the multipath components of all the users, which is actually mixed down to the baseband after multiplied by the spreading sequence of the desired User 0.

Assuming that the receiver is delayed and phase synchronized with User 0, the decision statistic $Y^{(0)}$ for User 0 is given by integrating over one bit period T_b as

$$\begin{aligned} Y^{(0)} &= \int_0^{T_b} r(t) a_0^*(t) e^{-j\omega_c t} dt \\ &= I_0 + \xi + \zeta + \eta \end{aligned} \quad (5.36)$$

where I_0 is the desired contribution to the decision statistic $Y^{(0)}$ from User 0, (which is the same as (5.27) since τ_0 and ϕ_0 are the relative terms and considered as zero reference) defined as

$$\begin{aligned} I_0 &= \sqrt{2P_0} \int_0^{T_b} b_0(t) \Re \{ a_0(t) e^{j\omega_c t} \} a_0^*(t) e^{-j\omega_c t} dt \\ &\approx \pm \sqrt{\frac{P_0}{2}} T_b \pm j \sqrt{\frac{P_0}{2}} T_b, \end{aligned} \quad (5.37)$$

and η is the AWGN random variable as defined in (5.28). And the term ξ represents

the MAI among the simultaneous users defined as

$$\begin{aligned}\xi &= \sum_{k=1}^{K-1} I_k \\ &= \sum_{k=1}^K \sqrt{2P_k} \int_0^{T_b} b_k(t - \tau_k) \Re \{ a_k(t - \tau_k) e^{j\omega_c t + \phi_k} \} a_0^*(t) e^{-j\omega_c t} dt\end{aligned}\quad (5.38)$$

where $a_k(t - \tau_k)$ can be $\{\pm 1, \pm j\}$ as τ_k is assumed to be an integer multiple of T_c and $\Re \{ a_k(t - \tau_k) e^{j\omega_c t + \phi_k} \}$ is $\mp \sin(\omega_c t + \phi_k)$ if $a_k(t - \tau_k)$ is $\pm j$ whereas it is $\pm \cos(\omega_c t + \phi_k)$ if $a_k(t - \tau_k)$ is ± 1 . Now let us consider the demodulation part only. If we consider the sine term, $\sin(\omega_c t + \phi_k)$, then

$$\int_0^{T_b} \sin(\omega_c t + \phi_k) e^{-j\omega_c t} dt = \int_0^{T_b} [\sin(\omega_c t + \phi_k) (\cos \omega_c t - j \sin \omega_c t)] dt. \quad (5.39)$$

Using the trigonometric equations expressed in (5.31), (5.39) becomes

$$\begin{aligned}& \int_0^{T_b} [\sin(\omega_c t + \phi_k) (\cos \omega_c t - j \sin \omega_c t)] dt \\ &= \int_0^{T_b} \left[\frac{1}{2} \sin 2\omega_c t \cos \phi_k - j \sin^2 \omega_c t \cos \phi_k + \cos^2 \omega_c t \sin \phi_k - j \frac{1}{2} \sin 2\omega_c t \sin \phi_k \right] dt \\ &= \left(-\frac{j}{2} \cos \phi_k + \frac{1}{2} \sin \phi_k \right) T_b.\end{aligned}\quad (5.40)$$

Likewise,

$$\begin{aligned}
& \int_0^{T_b} [\cos(\omega_c t + \phi_k) (\cos \omega_c t - j \sin \omega_c t)] dt \\
&= \int_0^{T_b} \left[\cos^2 \omega_c t \cos \phi_k - j \frac{1}{2} \sin 2\omega_c t \cos \phi_k - \frac{1}{2} \sin 2\omega_c t \sin \phi_k + j \sin^2 \omega_c t \sin \phi_k \right] dt \\
&= \left(\frac{1}{2} \cos \phi_k + j \frac{1}{2} \sin \phi_k \right) T_b.
\end{aligned} \tag{5.41}$$

Therefore, (5.38) can be approximated as

$$\xi \approx T_b \sqrt{\frac{P_0}{2}} \sum_{k=1}^K (\cos \phi_k + \sin \phi_k) \Re \{ b_{k,-1} R_{k,0}(\tau_k) + b_{k,0} \hat{R}_{k,0}(\tau_k) \} \tag{5.42}$$

since we consider $\Re\{Y^{(0)}\}$ in decision making of output data, and where $b_{k,0}$ and $b_{k,-1}$ are the current and previous received bits from User k , $R_{k,0}$ and $\hat{R}_{k,0}$ are the partial crosscorrelation functions defined by

$$\begin{aligned}
R_{k,0}(\tau_k) &= \sum_{n=0}^{\tau_k-1} a_k(n - \tau_k) a_0^*(n) \\
\hat{R}_{k,0}(\tau_k) &= \sum_{n=\tau_k}^{N-1} a_k(n - \tau_k) a_0^*(n)
\end{aligned} \tag{5.43}$$

where τ_k is the integer multiple of chip time period such that $0 \leq \tau_k \leq N - 1$ and N is the length of the spreading sequence. Finally, the term ζ in (5.36) symbolizes the intersymbol interference (ISI) due to the multipaths, which is defined by

$$\zeta = \frac{\sqrt{2P_0}}{L} \sum_{k=0}^{K-1} \sum_{l=2}^L \int_0^{T_b} b_l(t - \tau_k - lT_c) \Re \{ a_k(t - \tau_k - lT_c) e^{j\omega_c t + \theta_{kl}} \} a_0^*(t) e^{-j\omega_c t} dt. \tag{5.44}$$

By adopting the same demodulation procedure used to derive ξ , (5.44) can be given by

$$\zeta \approx \frac{T_b}{L} \sqrt{\frac{P_0}{2}} \sum_{k=0}^{K-1} \sum_{l=2}^L (\cos \theta_{kl} + \sin \theta_{kl}) \Re\{b_{k,-1} R_{k,0}(\tau_{kl}) + b_{k,0} \hat{R}_{k,0}(\tau_{kl})\} \quad (5.45)$$

where $\tau_{kl} = \tau_k + lT_c$ such that $0 < \tau_{kl} \leq N + L - 1$. The probability of error is given by

$$P_e = Q \left(\sqrt{\frac{P_0 T_b^2}{2 (\sigma_\eta^2 + \sigma_\xi^2 + \sigma_\zeta^2)}} \right) \quad (5.46)$$

Therefore, the low values of aperiodic autocorrelation and crosscorrelation values are both desired in the asynchronous DS-CDMA system over the multipath fading channel. The bit-error-rate performance is evaluated through the Monte Carlo simulation in the next subsection, and the simulation results and discussions are also provided.

5.6.2 Simulation Results

The performances of various spreading sequences in an asynchronous DS-CDMA system is evaluated via computer simulation. For the asynchronous case, the aperiodic correlation is the key parameter to examine the performance of spreading sequences as mentioned in Subsection 5.4.2. To achieve the usable sequence set, the selection of the sequence subset is performed for each category of sequences based on the algorithm as described in Figure 5.6. First, the sums of the aperiodic CC values between any sequences in each subset are calculated. In order to suppress the MAI effectively, the subset with smaller sum is selected from the same set of the sequences for the analysis. Due to the existence of the tradeoff between the AC and CC properties, the subsets having smallest MS-CC are avoided for the analysis. The length of spreading sequences is considered as $N = 2^6 = 64$ whereas the length of the Gold, small set of Kasami and m -sequences is 63. For the UCHT sequences, as pointed out in Subsection

5.2.4, there are two groups of sequences: the HSP-UCHT sequences and the NHSP-UCHT sequences, having 32 sequences in each group. Among 32 complex spreading sequences, one of them from each group is selected for the simulation. This is because the correlation properties of the 32 sets of those complex sequences in each group are similar [14]. We choose the parameters to be $\tau_1 = j$, $\tau_2 = 1$, $\tau_3 = j$, for the HSP-UCHT sequence, and $\tau_1 = -j$, $\tau_2 = -1$, $\tau_3 = -1$, for the NHSP-UCHT sequence. For the Gold sequences, they are produced by a preferred pair of two m -sequences u and v , where u is obtained by using the primitive polynomial $f_1(x) = 1 + x^5 + x^6$ and v is obtained by using $f_2(x) = 1 + x^2 + x^3 + x^5 + x^6$. Hence, the period of resultant Gold sequences is $N = 63$ and it has 65 different Gold sequences belonging to a set $\{u, v, u \oplus v, u \oplus \text{all shifted versions of } v\}$ where $\oplus$ is the modulo-2 addition operator. The Kasami sequences are generated by decimating an m -sequence, hence, it provides a smaller set of sequences as compared to the Gold sequences.

The aperiodic MS-CC and MS-AC values of different spreading sequences for various subset sizes are calculated and listed in Tables 5.3 and 5.4. It is observed from the tables that the SCHAT sequence occupies the smallest CC values as compared with the other existing sequences such as the HSP-UCHT, NHSP-UCHT, WHT, Gold, small set of Kasami and m -sequences. Therefore, the SCHAT sequences is expected to perform well in the multiuser environment. However, when a set of sequences offers good CC properties, generally the AC properties of the same set are poor [14, 45]. Hence, the SCHAT sequence exhibits relatively poor AC properties as compared with the other sequences. That is, the SCHAT sequence provide larger aperiodic MS-AC (R_{ac}) values than the others although it has smaller MS-CC (R_{cc}) values as shown in Tables 5.3 and 5.4.

For the simulation, it is assumed that all the users in the system are transmitting the data randomly using the same power (perfect power control) from their transmitters, i.e., $P_0 = P_1 = \dots = P_{K-1}$. The corresponding DS-CDMA system model is

Family	SHT		WHT		HSP-UCHT		NHSP-UCHT	
K	R_{cc}	R_{ac}	R_{cc}	R_{ac}	R_{cc}	R_{ac}	R_{cc}	R_{ac}
6	0.2448	33.3802	2.1865	22.1615	0.9865	18.1823	2.3323	27.5885
8	0.1311	25.0352	1.1713	16.6211	0.5285	13.6367	1.2517	20.6836
10	0.0816	20.0281	0.7288	13.2969	0.3288	10.9094	0.7788	16.5469
12	0.0556	16.6901	0.4969	11.0807	0.2242	9.0911	0.4998	13.7109
14	0.0404	14.3058	0.3604	9.4978	0.1029	7.6585	0.3625	11.7522
16	0.0306	12.5176	0.2733	8.3105	0.0780	6.7012	0.2749	10.2637
18	0.0240	11.1267	0.2144	7.3872	0.0612	5.9566	0.2814	8.9010
20	0.0193	10.0141	0.1726	6.6484	0.0493	5.3609	0.2266	8.0109

Table 5.3: Aperiodic MS correlation parameters for the SHT, UCHT and WHT sequences with the length of 64.

Family	K	R_{cc}	R_{ac}
Gold	6	1.1142	2.1151
	8	0.5719	1.5657
	10	0.3632	1.2524
	12	0.2432	1.0646
	14	0.1782	0.9062
	16	0.1388	0.7963
	18	0.1055	0.6968
	20	0.0860	0.6302
m	6	1.0903	0.3529
Small set of Kasami	8	0.8971	0.7846

Table 5.4: Aperiodic MS correlation parameters for the Gold, small set of Kasami and m -sequences with the length of 63.

as shown in Figure 5.7 using the correlation receiver. Without loss of generality, the time delay parameter which accounts for the propagation delay and the lack of synchronism between the desired signal and the interfering signal is the integer multiple of the chip time period. The integer is randomly chosen between 1 and N . The phase angles of the interfering signals and the multipath signals are assumed to be uniformly distributed between 0 and 2π . The phase angle for the desired user is assumed to be zero as the reference. Besides, it is also assumed that the interference is only coming from the other users and Gaussian noise. For the case of the multipath components, the interference caused by the multipaths of different users and the desired user itself is considered. Therefore, the intersymbol interference comes into consideration in the simulation. The line of sight (LOS) is assumed to be present and the system is considered for the indoor environment. The signal strength belonged to the indirect path is considered to have some percentage of the original signal strength. The number of multipaths including the LOS is $L = 10$ in this simulation.

Figures 5.12 to 5.14 illustrate the BER performances of various spreading sequences in the asynchronous DS-CDMA system over the multipath fading channel for the active users $K = 8, 20$ and 30 , respectively. Under the assumption of perfect power control in the simulation, the number of users can be increased to the length of the spreading sequences, N . In a practical system, the number of users per cell is not only determined by the processing gain of the system but also depends on the desired SNR and several other factors. The modest number of a CDMA system with omnidirectional antennas is 32 conversations per mobile base station. In our computer simulation, the length of spreading sequences is considered as 64.

It is known that the lower values of R_{cc} and R_{ac} are both desired in the DS-CDMA system over the multipath channel. As mentioned above, the SCHT sequence has relatively poor AC properties as compared with the HSP-UCHT and WHT sequences. Therefore, when the number of active users is 8, the BER performance of the SCHT

sequence is poorer than that of the WHT and HSP-UCHT sequences as shown in Figure 5.12. The HSP-UCHT sequence has relatively good AC properties [14], hence, this results in better performance in such multiple access (MA) environment (over the multipath fading channel). But while the number of users increases in the system, the errors caused by the MAI also increases. The effect of good CC properties contributes more in the BER performance. Hence, we can see from Figure 5.13 and Figure 5.14 that when the number of active users are 20 and 30, the BER performance of the SCHAT sequence is better than that of the other sequences which are the HSP-UCHT, NHSP-UCHT, WHT and Gold sequences under such circumstances. It can also be observed from Figures 5.12, 5.13, and 5.14 that at low E_b/N_0 , the BER is mostly affected by the additive white Gaussian noise. The contribution of good CC properties is minimal. When E_b/N_0 becomes larger, however, the effect of small CC values contributes more in the BER performance as the errors due to the MAI increase. Therefore, the differences in the BER performances become significant as demonstrated in the figures.

Figures 5.15 and 5.16 present the BER performances of various spreading sequences with respect to the increment of the number of simultaneous users in the system at a fixed input $E_b/N_0 = 8$ dB and 10 dB. As shown in the figures, the BER performance of each spreading sequence degrades when the number of active users increases in the system. But it is observed from the figures that the BER performance of the SCHAT sequence varies and deteriorates slightly as compared with the others. This points out that the SCHAT sequence has very good crosscorrelation properties, which can suppress the MAI effectively. When the number of active users becomes more than a certain number (which is 16 in Figure 5.15, and 12 in Figure 5.16), the SCHAT sequence offers better results than that of the WHT sequence and the others as illustrated in the figures. As pointed out earlier, the effect of good CC properties contributes more in the BER performance when E_b/N_0 becomes larger, hence, the SCHAT sequence performs

better than that of the WHT even when the number of active users is more than 12 at higher E_b/N_0 (which is 10 dB) as shown in Figure 5.16. Therefore, it should be noted that the SCHAT sequence is a better choice to be considered among the sequences in the asynchronous DS-CDMA system over the multipath fading channel with the presence of the MAI and the AWGN especially if the system wants to accommodate larger user capacity at a particular BER, and/or considers high E_b/N_0 .

Figure 5.12: BER performances of various spreading sequences in the asynchronous DS-CDMA system over the multipath channel ($K = 8$ and $L = 10$).

Figure 5.13: BER performances of various spreading sequences in the asynchronous DS-CDMA system over the multipath channel ($K = 20$ and $L = 10$).

Figure 5.14: BER performances of various spreading sequences in the asynchronous DS-CDMA system over the multipath channel ($K = 30$ and $L = 10$).

Figure 5.15: BER versus the number of users under fixed E_b/N_0 at 8 dB.

Figure 5.16: BER versus the number of users under fixed E_b/N_0 at 10 dB.

5.7 Conclusion

A set of new orthogonal complex spreading sequences known as the SCHAT sequences is proposed for use in the DS-CDMA systems in this chapter. Since these sequences are derived from the row vectors of an SCHAT matrix, they are orthogonal to each other and they can provide a large number of sequences available to accommodate the maximum number of users in the CDMA systems. Besides, it has been found that the SCHAT sequences have very good CC properties as compared with other spreading sequences such as the HSP-UCHT, NHSP-UCHT, WHT, Gold, small set of Kasami and m -sequences. Due to the tradeoff between the CC and AC properties, the SCHAT sequences exhibit relatively poorer AC properties than the WHT and UCHT sequences. Therefore, a careful choice to achieve a suitable SCHAT sequence set is critical. An algorithm to select the appropriate subsets of sequences is also mentioned in this chapter.

The average performance indicators of a sequence set in the DS-CDMA systems are explained for both synchronous and asynchronous cases in Section 5.4. The periodic MS-CC (φ_{cc}) and MS-AC (φ_{ac}) parameters are used to evaluate the performance of the sequence set in the synchronous CDMA system whereas the aperiodic MS-CC (R_{cc}) and MS-AC (R_{ac}) parameters are considered for the asynchronous case. The mathematical derivations for probability of error of both synchronous and asynchronous DS-CDMA systems using the orthogonal spreading sequences are given in detail in Section 5.5 and 5.6. Then, the Monte Carlo simulation is performed to measure the BER performance of the proposed sequence in both synchronous and asynchronous DS-CDMA systems with the presence of the MAI and the AWGN, and the results are compared with other spreading sequences. In the synchronous DS-CDMA system, it has been found that the orthogonal SCHAT sequences perform better than the non-orthogonal Gold sequences in terms of bit-error-rate. Moreover, it has also

been investigated that when the number of active users increases in the system, the proposed sequence demonstrates much better BER performance than the traditional Gold sequence. Therefore, it has been concluded that providing a particular value of BER, the system using the SCHAT spreading sequences is able to support more users than that using the Gold sequences.

In the asynchronous DS-SS system, the aperiodic MS-CC and MS-AC values are first calculated and analyzed as the parameters for performance measurement of the sequence set. It has been observed that the SCHAT sequences offer the smallest MS-CC values for various subset sizes among different spreading sequences, especially when the size of subsets becomes larger. Consequently, the simulation results convey that when the number of active users increases in the system, the SCHAT sequences demonstrate better BER performance than the UCHAT, WHT and Gold sequences in the asynchronous DS-SS system. It can be concluded that the SCHAT sequences are suited to be employed in the multiuser environment to combat the MAI over the multipath fading channel, especially when the number of the multipath signals is relatively small due to its relatively poor AC properties. Under such conditions, the system with the proposed sequence can provide higher user capacity than that of the other existing sequences.

Chapter 6

Application of the SCHT in Image Watermarking

6.1 Introduction

Due to the rapid and extensive development of high-speed internet and communication technologies, the use of multimedia applications is becoming more and more widespread. All the electronic documents are accessible to a large number of people in the world via the World Wide Web in a very cost-efficient way. Unfortunately, the digital information is quite easy to be duplicated and intentionally/unintentionally attacked by the unauthorized users. This motivates a significant interest for the researchers to find a way to discourage unauthorized copying and effortless distributing of electronic documents, especially to protect the copyright. Consequently, digital watermarking emerges as one potential and popular solution to identify the copyright ownership and track the usage of digital multimedia works. A digital watermark is embedded in the digital data permanently to identify the source of information for claiming the ownership, and it is still remained within the data after the decryption process. An effective image watermark embedded into the images should have the

following characteristics: [50–52]

- (a) Imperceptible: It should be perceptually invisible and the original image should not be visibly degraded by the presence of the watermark.
- (b) Robust: It should be difficult to remove the watermark. The mark should also be robust to common signal processing attacks such as resizing, sharpening, filtering, cropping, lossy compression and so on. Even though those operations damage the watermarked image, the watermark should not be removed and should be able to be detected during the extraction process.
- (c) Secure and reliable: It should be undetectable by the hackers and be strongly resistant to the unauthorized detecting and decoding. This is determined by the watermarking embedding algorithm and watermark structure.

In the literature, several methods have been developed for image watermarking. In general, they can be classified into two categories, one is processed in the spatial domain and the other in the transform domain. But the watermarks embedded in the spatial domain are not quite robust to tampering as compared to that concealed in the transform domain [53]. One major disadvantage is that a watermark embedded in the spatial domain may be eliminated easily by a common picture cropping operation [54]. On the other hand, the transform domain watermarking schemes have several desirable features. For example, one can insert the watermarks into the host image based on the least perceptually significance of different transform coefficients. This will enhance the imperceptibility of the marks and lead to less visual degradation of the watermarked image. Besides, since the watermark is distributed over the transform coefficients irregularly, it is difficult for the hackers to decode the watermark. Various discrete transforms such as the discrete cosine transform (DCT) [51, 54], the discrete Fourier transform (DFT) [55–58], the discrete wavelet transform (DWT) [58–60], the

Walsh Hadamard transform (WHT) [19, 52], the unified complex Hadamard transform [52, 61], and the discrete fractional random transform (DFRNT) [53] have been introduced so far to transform the host image into an alternative domain. After transforming, the watermark is embedded using some robust algorithms. Finally, the inverse transformation is performed to convert back to the spatial domain to obtain the watermarked images.

This chapter aims to propose a robust phase domain image watermarking algorithm which is processed in the sequency domain using the SCHAT. As mentioned earlier, an SCHAT matrix is an orthogonal complex Hadamard transform matrix whose row vectors are ordered in an ascending number of sequencies. In Chapter 5, the SCHAT was employed in direct sequence CDMA systems, in which each row vector of the SCHAT matrix is used as a complex spreading sequence for a particular user. In this chapter, due to its sequency ordering (which is analogous to frequency in the DFT), plus its simpler transform implementation with the availability of fast algorithm [62], the SCHAT is introduced in image watermarking to develop a technique for embedding and extracting watermarks. Besides, the SCHAT coefficients are complex numbers comprised of both amplitudes and phases so they are well suited to adopt phase modulation techniques such as the phase-shift-keying (PSK) modulation to encode the watermark information. In the proposed scheme, two dimensional binary pattern watermark is used. First, the watermark is pre-permuted into noises to disperse the spatial relationship. Then, each watermark bit is expanded by spread spectrum, and the expanded watermark bits are embedded by the PSK modulation. This scheme also adopts the amplitude boost (AB) and low amplitude block selection (LABS) methods to further improve the robustness and to preserve the imperceptibility of the watermark.

The rest of this chapter is organized as follows: Section 6.2 reviews the previous works done by the others. Section 6.3 presents the proposed embedding algorithm

and Section 6.4 details the watermark extraction process. Section 6.5 provides the experimental results together with simulation comparisons with the other schemes. Finally, Section 6.6 concludes this chapter.

6.2 Review of Previous Works

In [54], the transform domain approach is implemented based on the 8×8 block-based DCT. The DCT coefficients are altered to convey the watermark information. First, the watermark is generated as binary pattern and then it is permuted to disperse the spatial relationship. Due to the trade off between the perceptual invisibility and robustness to the lossy compression operation, the middle frequency range of the DCT coefficients of the image is selected to encode the watermark bits. This scheme is able to sustain common image processing operations and the JPEG compression. In [51], Cox et al. introduced spread spectrum communication technique in multimedia watermarking to ensure a large measure of security against intentional or unintentional attacks to the watermarked image.

In [59], the DWT was used to conceal the watermark effectively. First, the host image is decomposed using the wavelet multiresolutional structure [58] to construct the image frequency components. For the security purpose, the watermark is pre-permuted to become chaotic in the spatial domain by using toral automorphism with a user's key. Then, the scattered secret bits are inserted by modifying the selected DWT coefficients. Ref. [19] utilized the WHT as the passage from spatial to frequency domain to embed the watermark bits due to its simplicity in implementation.

In order to enhance the robustness of watermarking schemes, phase modulation techniques in the transform domain are introduced in [52, 55, 56]. The reasons for selecting the phase components rather than the amplitudes are that a watermark embedded in the phase domain of the transform coefficients is more robust to tampering,

and besides, it is well known from the communication theory that modulating the phase of carrier can achieve better performance and superior noise immunity than amplitude modulation. The general approach in these papers is that the host image is segmented into the 8×8 blocks and each block is mapped into the transform domains. Only the portions which are most significant to the image integrity are marked. Chen et al. [55] proposed the amplitude boost and the low amplitude block selection methods to further enhance the robustness of the watermarking scheme. The phases of the selected DFT coefficients convey the binary watermark information. During the watermark extraction process, the original host image and the watermark are not needed. The scheme achieves robustness which can sustain the JPEG compression and most common signal processing attacks, but it is vulnerable to the phase perturbation attack [63].

Falkowski [52] presented a new concept for a robust phase watermarking scheme using multi-polarity Walsh Hadamard and complex Hadamard transform (CHT). In that paper, at first, the multiresolutional WHT is employed to decompose the host image into the pyramid structure which consists of various bands to determine the most significant components of the image. Then, the multi-polarity CHT is performed on each selected 8×8 block. The phases of the selected CHT coefficients are marked based on the pseudo-random watermark sequence and the coefficients of variance (COV) of the pixels in the 8×8 blocks before the CHT. The inverse procedures are applied to get the watermarked image. The similarity of the original phases and the extracted phases is determined based on certain predefined threshold during the extraction process.

In [53], the DFRNT was used to propose a novel watermarking algorithm. In contrast to the previous methods listed above, the whole image is segmented into several blocks only after the DFRNT. The blocks are chosen randomly for the embedding process according to the size of the watermark. The high amplitude selection and phase-shift-keying methods are adopted to reinforce the robustness of the watermark-

ing scheme. The similarity between the original and extracted watermarks is measured based on their mean square error (MSE) during the detection process. The advantage of this scheme is the randomness of the kernel DFRNT matrix, which leads to high security in information hiding applications.

6.3 Watermark Embedding Algorithm

The design and implementation of a novel transform domain based watermarking algorithm for grey-scaled images is presented in this section. Transforming the image to the sequency domain involves the two dimensional (2-D) SCHT, hence, we first define the 2-D SCHT as follows.

As mentioned in Subsection 3.2.4, an N -point one dimensional (1-D) SCHT of a signal vector $\mathbf{x}_N$ is defined as

$$\mathbf{X}_N = \frac{1}{N} \mathbf{H}_N^* \mathbf{x}_N \quad (6.1)$$

where $\mathbf{x}_N = [x(0), x(1), \dots, x(N-1)]^T$ and $\mathbf{X}_N = [X(0), X(1), \dots, X(N-1)]^T$ is the transformed complex column vector. The data sequence can be recovered uniquely from the inverse transform, that is,

$$\mathbf{x}_N = \mathbf{H}_N \mathbf{X}_N. \quad (6.2)$$

A 2-D SCHT of an segmented $N \times N$ image block can be defined by applying 1-D SCHT on the rows of the block followed by the corresponding columns. The 2-D SCHT is defined as

$$\mathbf{Y}_N = \mathbf{C}_N \mathbf{Z}_N \mathbf{C}_N^T \quad (6.3)$$

where $\mathbf{C}_N^T$ represents the transpose of $\mathbf{C}_N$,

$$\mathbf{C}_N = \frac{1}{\sqrt{N}} \mathbf{H}_N^* \quad (6.4)$$

where $\mathbf{H}_N^*$ is the complex conjugate transpose of the SHT matrix $\mathbf{H}_N$, $\mathbf{Z}_N$ is the $N \times N$ real image block, and $\mathbf{Y}_N$ is the 2-D SHT coefficient matrix of dimension $N \times N$. The SHT coefficients of a real image are the complex values which consist of both magnitudes and phases, hence, the PSK modulation technique can be utilized to embed the watermark in the SHT transformed images. The inverse process can be performed to transform the images back to the original spatial domain.

A robust watermarking scheme must sustain all kinds of attacks, plus it should preserve the visual quality of the original image after embedding the watermark. Various methods are employed to improve the performance of proposed watermarking scheme. The overall watermarking procedure to conceal the watermark is shown in Figure 6.1. The proposed scheme is outlined step by step as follows:

- (1) At first, the 2-D binary pattern watermark, W , is scrambled into noise-like pattern, W_p , by applying pseudo-random permutation using a pseudo-random sequence (PN_1) to disperse the spatial relationship and to enhance the security. We use a visible image of 32×32 pixels as the watermark to be embedded. Such kinds of visually recognizable patterns are more natural to represent one's identity than a sequence of random numbers [54].
- (2) It is then transformed into a bipolar sequence of bits (denoted as d) and expanded by spread spectrum to become a spread bipolar bit sequence m .
- (3) On the other hand, the 8-bit grey-scaled image of 512×512 pixels is evenly divided into the 8×8 non-overlapping blocks.
- (4) The 2-D SHT is performed for each 8×8 image block to transform the whole

Figure 6.1: The flow chart of watermark embedding process.

image from the spatial domain into the sequency domain.

- (5) The low amplitude block selection (LABS) strategy [55] is adopted in our watermarking scheme to preserve the imperceptibility of the watermark as much as possible. The LABS strategy selects a certain number of blocks, B_i , which is equal to the length of the watermark bit sequence m as each secret bit is to be inserted into each transformed 8×8 block. The block selection is made over the whole image globally.
- (6) Subsequently, the amplitude boost (AB) method [55] is hired to enhance the robustness. The amplitude, A , of each selected SHT coefficient, $Ae^{j\phi}$, from each selected block, B_i , is raised to A' to combat the attacks.
- (7) The phase, ϕ , of each selected coefficient is altered to ϕ' according to the spread bipolar watermark bit sequence using the PSK modulation. The embedding

procedure is repeated for each secret bit in m until we get Y' .

- (8) Finally, the inverse 2-D SCHT is performed on the modified transformed image Y' to obtain the respective watermarked image, R .

The details of some steps are presented in the following.

6.3.1 Pseudo-random Permutation of the Watermark

In our approach, the watermark is first pre-permuted into noises using a pseudo-random sequences (denoted as PN_1 which will be kept for watermark extraction in future) for the security purpose. The shape of the watermark becomes chaotic after permuting the image matrix as shown in Figure 6.2. Hence, it will protect the watermark from being stolen by the others. The permutation is performed as follows:

Figure 6.2: The watermark transformed by random permutation, (a) the original watermark, and (b) the permuted watermark.

- (1) First, transform the binary $M \times M$ watermark, W , into a binary bit sequence, u_w , of the size $1 \times n$ where $n = M \times M$.
- (2) Generate a sequence of indices (which is PN_1) by pseudo-random permuting the integers from 1 to n using the randperm MATLAB command.
- (3) Permute the watermark sequence, u_w , based on the sequence of indices obtained from Step 2 to become the scrambled sequence, u_p .
- (4) Then, rearrange the sequence of u_p into the two-dimensional $M \times M$ matrix to obtain a permuted image, W_p .

6.3.2 Spread Spectrum

The pseudo-noise (PN) sequence based spread spectrum (SS) method [51] is used in this introduced scheme. SS ensures a large measure of security against various attacks. In the PN-based spread spectrum system, each information bit is expanded into several bits using the PN sequence. To accomplish the job, first W_p (obtained from Subsection 6.3.1) is converted into a bipolar bit sequence, $d(i)$, where $i = 1, 2, \dots, n$ by mapping 0 to 1 and 1 to -1 . Second, we generate a total of n different PN sequences, $p_i(j) \in \{\pm 1\}$, with all having the same length, l , where $j = 1, 2, \dots, l$ using a user's secret key (which is saved as PN_2 for future watermark detection). By multiplying $d(i)$ with $p_i(j)$, each information-bearing bit is spread into l bits and the whole expanded bipolar bit sequence m is expressed by

$$m_i(j) = d(i) \cdot p_i(j) \quad (6.5)$$

where l is the length of the PN sequences, n is the length of the watermark bit sequence, and $j = 1, 2, \dots, l$ and $i = 1, 2, \dots, n$. The total length of the spread watermark bit sequence will be $l \times n$.

6.3.3 Low Amplitude Block Selection

The LABS method is employed to minimize the difference between the visual qualities of the two images, before and after the embedding process as much as possible. The number of the transformed 8×8 blocks in the sequency domain is more than the size of spread watermark bit sequence. We sum all the amplitudes in each 8×8 transformed blocks and the LABS strategy selects the 8×8 blocks having lower summations of amplitudes out of the total number of blocks available in order to insert the secret bits. The selection is done over the whole image globally. Consequently, the selected

blocks are recorded for future detection purpose using a user's key (which is expressed as Key in Figure 6.1).

To make the watermarking scheme robust, the watermark bits should be placed in the most significant components of an image because this portion of the host data is robust to the attacks and highly sensitive to alteration [55]. For most natural images, the energy is concentrated at the lower frequency range [51, 54] and the information hidden in the high or middle frequency bands is easily removed or lost after some operations such as low pass filtering and quantization of lossy compression. Therefore, the secret bits are embedded in the lower sequency components in this scheme using the analogy with the frequency-based watermarking scheme. Therefore, $B(1,1)$ of each 8×8 coefficient block is selected as the locations to embed the secret bits, which is shown in Figure 6.3. The amplitude and phase of $B(1,1)$ are denoted as A and ϕ respectively, which are as illustrated in Figure 6.1.

	$B(1,1)$						

Figure 6.3: The selected SCHAT coefficient to embed the watermark within an 8×8 block.

6.3.4 Amplitude Boost

The purpose of the AB strategy is to enhance the robustness. In the PSK modulation, the watermark information is inserted into the phases of the selected SCHAT coefficient. When retrieving the watermark, the attack on the embedded image will alter the magnitudes and phases of the SCHAT coefficients, which causes the distortion to the recovered watermark. Figure 6.4 illustrates the distortion $\Delta\phi$ of an SCHAT coefficient $Ae^{j\phi}$ due to the 2-D additive Gaussian noise of standard deviation σ . Referring to the figure, we obtain [53, 56]

$$\begin{aligned} A'' \cos \Delta\phi &= A + e_1 \\ A'' \sin \Delta\phi &= e_2 \end{aligned} \quad (6.6)$$

where e_1 and e_2 are the two independent one-dimensional Gaussian noise of standard deviation σ , A'' denotes the magnitude of the distorted phasor and $\Delta\phi$ represents the amount of distortion to the phasor. The joint probability density function $p(e_1, e_2)$ is

Figure 6.4: The phase change of a phasor $Ae^{j\phi}$ subjected to a two-dimensional additive Gaussian noise of standard deviation σ .

given by

$$p(e_1, e_2) = \frac{1}{2\pi\sigma^2} \exp \left[-\frac{e_1^2 + e_2^2}{2\sigma^2} \right]. \quad (6.7)$$

Substituting (6.6) into (6.7), it is easy to express $p(e_1, e_2)$ in terms of the random variables A'' and $\Delta\phi$ as

$$p(A'', \Delta\phi) = \frac{1}{2\pi\sigma^2} \exp \left[-\frac{A^2 - 2AA'' \cos \Delta\phi + A''^2}{2\sigma^2} \right]. \quad (6.8)$$

The joint density $p(A'', \Delta\phi)$ can be integrated with respect to A'' to obtain the probability density function $p(\Delta\phi)$. Assuming $A \gg \sigma$, it can be shown to be [56]

$$p(\Delta\phi) = \frac{1}{\sqrt{2\pi(\sigma/A)^2}} \exp \left[-\frac{(\Delta\phi)^2}{2(\sigma/A)^2} \right]. \quad (6.9)$$

Thus, the standard deviation of $\Delta\phi$ is equal to (σ/A) , which indicates that the distortion of phase distribution of the SCHAT coefficient due to the additive Gaussian noise is inversely proportional to the amplitude of that coefficient. It can be seen that the phase distortion effects of additive noise can be greatly reduced for those coefficients having large amplitudes.

Hence, in order to keep the phase distortion below a certain level, the amplitudes of the SCHAT coefficients should be maintained at one defined level. Subsequently, the AB method is hired to reinforce the robustness of the watermarking scheme. In the AB strategy, the amplitude of a selected SCHAT coefficient is boosted to a fixed threshold value th if its value is below th . That is,

$$A' = \begin{cases} th, & \text{if } A \leq th \\ A, & \text{if } A > th. \end{cases} \quad (6.10)$$

It has been observed that the SCHAT has certain advantages over the DFT (which

is employed in [55]) and the UCHT by incorporating the AB skill in the image watermarking algorithm. To illustrate this, we embed the watermark using the proposed scheme as shown in Figure 6.1 utilizing the DFT, the SCHAT and the UCHT by varying the values of th . Then, the peak signal-to-noise ratios (PSNRs) are calculated for each watermarked image at any particular value of th . The watermark is embedded in the phases of the transform coefficients in the proposed scheme, hence, the WHT and the DCT are not considered for comparison as their coefficients are real numbers. Table 6.1 shows the corresponding peak signal-to-noise ratio (PSNR) values of the SCHAT, DFT and UCHT watermarked images with respect to the original image based on the increasing values of th using the PSK modulation for the Lena image. The PSNR values of the SCHAT and DFT watermarked images are reduced when the values of th become larger as shown in Table 6.1 whereas the th values do not affect the PSNR value of the UCHT watermarked image. It can be seen from the table that at any particular value of th , the PSNR value of the SCHAT watermarked image is much higher than that of the DFT and UCHT watermarked images while maintaining the robustness at one specific level. Therefore, the value of th can be increased as large as possible to enhance the robustness for the SCHAT watermarked image while maintaining the image quality at one acceptable level.

th		12	16	20	24	28	32
PSNR (dB)	SCHAT	44.31	43.45	42.41	41.45	40.46	39.58
	DFT	37.72	36.91	36.02	35.11	34.20	33.34
	UCHT	26.41	26.41	26.41	26.41	26.41	26.41

Table 6.1: The PSNR values of the SCHAT, DFT and UCHT watermarked Lena images arranged according to the ascending threshold values.

6.3.5 PSK Modulation

In the PSK modulation, the phase ϕ is modified into ϕ' according to the following rule:

$$\phi' = \begin{cases} \pi/2, & \text{if } m(r) = 1 \\ 3\pi/2, & \text{if } m(r) = -1 \end{cases} \quad (6.11)$$

where $m(r)$ is the spread bipolar watermark bit sequence and $r = 1, 2, \dots, l \times n$ represents the sequence index. Figure 6.5 shows the signal constellation of the PSK modulation. The maximum phase perturbation due to the PSK modulation is $\pm\pi$. But the visual quality is still preserved in the watermarked image after embedding the watermark (which is shown in Figures 6.7 and 6.8).

Figure 6.5: The signal constellation for the PSK modulation.

6.4 Watermark Detection Algorithm

Figure 6.6 shows the flow chart for the watermark extraction process. It should be noted that the original host image and the watermark are not needed during the recovering process. But it is important that the watermark bits are decoded from the same positions where they were embedded previously. Therefore, the set of secret

Figure 6.6: The flow chart of watermark extraction process.

keys (Key , PN_1 , PN_2) used in the embedding process is to be reused in the extraction process as well. The detailed steps to extract the watermark are outlined as follows:

- (1) Firstly, the watermarked image R is segmented into the 8×8 non-overlapping blocks and the 2-D SCHAT is applied on those 8×8 blocks to transform the image R to Y'_R .
- (2) The same Key which was used in the embedding process is reused to locate the respective embedded blocks in Y'_R for the PSK demodulation.
- (3) During the PSK demodulation, the watermark bits $m_R(r)$ are extracted from the phases of the selected SCHAT coefficients from the located blocks in Y'_R .
- (4) After all the secret bits are decoded, the extracted bipolar bit sequence is de-spread by the inverse spread spectrum (using PN_2) to obtain a contracted bipolar bit sequence. It is then transformed back to the binary bit sequence by mapping 1 to 0 and -1 to 1 and rearranged into the two-dimensional $M \times M$ image W'_R .
- (5) Finally, W'_R is reverse-permuted to get the recovered watermark W_R according to the predefined pseudo-random sequence PN_1 .

Some steps are elaborated further in the following.

6.4.1 PSK Demodulation

Let ϕ'' be the angle of interest to extract the secret bit from each selected SCHAT coefficient block. The minimum distance decision rule is applied in the process of PSK demodulation. Hence, $m_R(r)$ is detected as

$$m_R(r) = \begin{cases} 1, & \text{when } |\phi'' - \frac{\pi}{2}| \geq |\phi'' - \frac{3\pi}{2}| \\ -1, & \text{when } |\phi'' - \frac{\pi}{2}| < |\phi'' - \frac{3\pi}{2}| \end{cases} \quad (6.12)$$

where $r = 1, 2, \dots, l \times n$. In this way, all the secret bits are decoded, which results in the recovered spread watermark bit sequence comprising of $\{\pm 1\}$.

6.4.2 Inverse Spread Spectrum

After all the secret bits are recovered from PSK demodulation, ISS is performed to reconstruct the bipolar watermark bit sequence d_R . By using the PN sequences $p_i(j)$ (generated by PN_2) which were used in the spreading process, each bit can be determined from m_R as follows:

$$d_R(i) = \begin{cases} 1, & \text{if } \sum_{j=1}^l m_{R,i}(j) \cdot p_i(j) \geq 0 \\ -1, & \text{if } \sum_{j=1}^l m_{R,i}(j) \cdot p_i(j) < 0 \end{cases} \quad (6.13)$$

where $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, l$.

6.5 Experimental Results

In this watermarking scheme, peak signal-to-noise (PSNR) is used to measure the degree of transparency of the watermarked image. For an 8-bit grey-scaled watermarked

image R , PSNR is defined as

$$\text{PSNR} = 10 \log_{10} \frac{255^2}{\frac{1}{N \times N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} [Z(i, j) - R(i, j)]^2} \quad (6.14)$$

where $Z(i, j)$ and $R(i, j)$ represent the values of grey level at the $(i\text{th}, j\text{th})$ position of the original image Z and the watermarked image R of $(N \times N)$ pixels, respectively. In this proposed scheme, the visually recognizable pattern is used as the watermark, hence, the observers can compare the extracted watermark with the reference watermark subjectively. Therefore, a quantity measurement is required to provide an objective judgement for the extracted fidelity. In this scheme, normalized correlation (NC) [54] is used as the similarity measurement between the original and recovered watermarks. It is defined as

$$\text{NC} = \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{M-1} W(i, j) W_R(i, j)}{\sum_{i=0}^{M-1} \sum_{j=0}^{M-1} [W(i, j)]^2}, \quad (6.15)$$

which is normalized by the energy of the referenced watermark to give unity as peak correlation, where W and W_R represent the original watermark and the extracted watermark of size $(M \times M)$, respectively.

In order to evaluate our proposed watermarking scheme, we use the original grey-scaled Lena and Peppers images of (512×512) pixels as the test images as shown in Figures 6.7 and 6.8, and a watermark image of (32×32) pixels, which represents a symbol of star as shown in Figure 6.2 in our simulation. We have also tested on the other images using different watermark symbols, and the results are consistent. The values of parameters used in this simulation are: $l = 3$ and $th = 30$ for the SHT watermarked images whereas $th = 12$ for the DFT and UCHT watermarked images in order to preserve the visual quality of the images. For the UCHT transform matrix, we choose the parameters to be $\tau_1 = j$, $\tau_2 = 1$, $\tau_3 = j$ (see Subsection 2.4.3).

Figure 6.7: (a) The original image of Lena, the watermarked images of Lena using (b) the SHT with $\text{PSNR} = 39.95$ dB, (c) the DFT with $\text{PSNR} = 37.72$ dB, and (d) the UHT with $\text{PSNR} = 26.41$ dB.

Figure 6.8: (a) The original image of Peppers, the watermarked images of Peppers using (b) the SCHT with $\text{PSNR} = 40.31$ dB, (c) the DFT with $\text{PSNR} = 39.47$ dB, and (d) the UCHT with $\text{PSNR} = 27.90$ dB.

The threshold for the NC values is set to 0.8 to decide the extracted fidelity of the watermark. The block size used is 8×8 . Hence, the total number of blocks available is $(512 \times 512) \div (8 \times 8) = 4096$. The length of watermark bit sequence is $n = 32 \times 32 = 1024$ and after spreading, it becomes $1024 \times 3 = 3072$. Therefore, the LABS selects 3072 blocks from a total of 4096 blocks available as one secret bit is to be inserted into each 8×8 SCHT transformed blocks.

Figures 6.7 and 6.8 shows the original Lena and Peppers images and their watermarked versions of images using the SCHT, the DFT and the UCHT. The PSNR values of the corresponding watermarked images with respect to the original image are 39.95 dB, 37.72 dB and 26.41 dB respectively for the Lena images. Likewise, for the Peppers image, the respective PSNR values are 40.31 dB, 39.47 dB and 27.90 dB. We can see from both figures that despite the presence of the watermarked, the SCHT watermarked images do not contain any visible artifacts as the DFT watermarked images. One could hardly perceive the difference between the original and watermarked images. But the visual degradation occurs in the UCHT watermarked images as shown in Figures 6.7-d and 6.8-d. Therefore, the UCHT is not suitable for inserting watermark in the direct transformed coefficients, whereas the multiresolutional WHT is first used to decompose the host image into pyramid structure before taking the UCHT to perform watermarking [52]. The following subsections will demonstrate the robustness of the proposed scheme followed by comparisons with the other watermarking schemes.

6.5.1 Robustness of the Watermarking Scheme

Various attacks are considered to demonstrate the robustness of our watermarking scheme where the Lena and Peppers images are used as the test images again. The robustness of the watermarked images depends directly on the threshold value th used

for the AB method, which in turn influences the visual degradation of the images as shown in Table 6.1. Based on the tradeoff between the robustness and the transparency, we chose a suitable th value for an acceptable level of visual degradation for each watermarked image. Attacks include scaling, rotating, cropping, painting, low-pass filtering, sharpening, noising, phase perturbation [63] and JPEG compression. The details of each attack are shown below.

- (a) Image scaling: The watermarked image is scaled down from its original size of 512×512 pixels to 192×192 pixels. Then the reduced image is re-scaled back to the original dimension before the extraction process.
- (b) Image rotation: The image is rotated 90° clockwise followed by 90° anti-clockwise before proceeding the decoding process.
- (c) Image cropping I: For the cropping attack I, a quarter of the embedded image is discarded after the cropping operation.
- (d) Image cropping II: For the cropping attack II, only the central required portion of the marked image (56.25% of the original size) is remained. In order to extract the watermark, the missing portions are replaced with the black-colored pixels which are zeros. It is shown in Figure 6.9-d.
- (e) Painting: Several paints are added to the watermarked Lena image as shown in Figure 6.9-e.
- (f) Low-pass filtering: The filter is a rotationally symmetric Gaussian low-pass filter of the size 3×3 with the standard deviation σ of 1.4. The embedded image is filtered before proceeding the detection process.
- (g) Sharpening: The contrast of the image is adjusted to enhance the image quality before extracting the watermark. The sharpened version of the image is shown

in Figure 6.9-g.

- (h) Noising: Gaussian noises with zero mean and normalized variance 0.01 are introduced to the marked image. The noisy version of the watermarked Lena image is shown in Figure 6.9-h.
- (i) Phase perturbation: As mentioned in [63], first, a normally distributed noise with mean $\pi/4$ and variance 0.01 is generated and then the noise is added to all the middle sequency and high sequency phases of the SCHT coefficients in the sequency domain.

The figures for the corresponding attacked SCHT watermarked images under various kinds of image signal operations for the Lena image are illustrated in Figure 6.9. Their respective extracted watermarks using the proposed scheme are shown in Figure 6.10. The corresponding NC values (together with the number of error bits) for each attacked SCHT watermarked image are shown in Table 6.2 as compared with the results for the attacked DFT watermarked image. It can be seen from Table 6.2 that for the attacks of rotation, sharpening and phase perturbation, the watermarks are fully recovered from the embedded images as the value of NC is 1 and the number of errors is zero. The NC values which are relevant to the other attacks are also above the threshold value of 0.8 (acceptable number of error bits) and the corresponding extracted watermarks are visually recognized by the human vision as shown in Figure 6.10. But it has been observed that for the attack of scaling, more significant error bits are introduced for the SCHT based scheme as compared to the DFT based scheme as shown in Table 6.2. We have known that re-scaling the image in fact introduces high frequency components in the frequency domain while weakening the low and middle frequency components of the image. Since the watermark bits are embedded in the phases of low frequency/sequency components of the image, re-scaling will lead to corruption of the watermark inserted in the phase domain. As a result, the error bits

Figure 6.9: The SCHT watermarked images subjected to various attacks: (a) The scaled, (b) the rotated, (c) the cropped (a quarter image), (d) the cropped (central portion remained), (e) the painted, (f) the filtered, (g) the sharpened, (h) the noised, and (i) the phase perturbation attacked.

are detected when the watermark bits are decoded from the re-scaled DFT and SCHT watermarked images as shown in Item (a) of Table 6.2. But it should be noted that the effect of re-scaling changes mostly the phase angles of the SCHT coefficients in the sequency domain rather than that of the DFT coefficients in the frequency domain.

Figure 6.10: The corresponding extracted watermarks: (a) The scaled, (b) the rotated, (c) the cropped (a quarter image), (d) the cropped (central portion remained), (e) the painted, (f) the filtered, (g) the sharpened, (h) the noised, and (i) the phase perturbation attacked.

Attack type	Parameters	SCHT		DFT	
		eBits ^a	NC	eBits ^a	NC
(a) Scaling	512×512 to 192×192	45	0.95	4	0.99
(b) Rotating	90° rotated clockwise	0	1.00	0	1.00
(c) Cropping I	Cropping 1/4	76	0.92	79	0.92
(d) Cropping II	Central portion remained	164	0.84	155	0.85
(e) Painting	Painting 3 bars	2	0.99	2	0.99
(f) Low-pass filtering	Size 3×3 , $\sigma = 1.4$	14	0.99	0	1.00
(g) Sharpening	Sharpening the image	0	1.00	0	1.00
(h) Noising	Gaussian noise added	123	0.88	133	0.87
(i) Phase perturbation	Mean= $\pi/4$, var=0.01	0	1.00	0	1.00

^a The number of error bits detected.

Table 6.2: Comparison of the NC values corresponding to the number of error bits for the attacked watermarked Lena images under various operations between the SCHT and the DFT based schemes.

For the JPEG compression attacks, the watermarked image is compressed using the standard JPEG encoding at various quality levels. Table 6.3 summarizes the results for JPEG lossy compression tests from 10% to 100% quality factor using the

JPEG quality level(%)	SCHT		DFT	
	eBits ^a	NC	eBits ^a	NC
100	0	1.00	0	1.00
90	0	1.00	0	1.00
80	0	1.00	0	1.00
70	1	0.99	0	1.00
60	2	0.99	0	1.00
50	2	0.99	0	1.00
40	2	0.99	11	0.99
30	18	0.98	135	0.87
21	203	0.80	293	0.72
20	228	0.78	304	0.71
10	445	0.56	433	0.58

^a The number of error bits detected.

Table 6.3: Comparison of the NC values corresponding to the number of errors for the JPEG compressed watermarked Lena images between the SCHT and DFT based schemes.

proposed scheme based on both SCHT and DFT. It can be seen from Table 6.3 that as the quality level decreases, the NC value reduces accordingly. Figure 6.11 shows the JPEG compressed versions of the watermarked Lena image based on the SCHT together with their extracted watermarks at quality levels 50%, 40%, 30%, 21%, 20% and 10%, respectively. At 21% quality level, the NC value (0.80 which is at the threshold) is still high enough to recognize the extracted watermark as shown in Figure 6.11-d for the SCHT based scheme. The watermark is visible and we are still able to recover it. Below 21%, the quality of the JPEG encoded watermarked image is degraded severely and the extracted watermarks become indiscernible as shown in Figures 6.11-e and 6.11-f.

Figure 6.11: The JPEG encoded versions of the SCHAT watermarked image of Lena and the corresponding extracted watermarks: (a) at 50% quality with $NC \cong 1.00$, (b) at 40% quality with $NC \cong 1.00$, (c) at 30% quality with $NC=0.98$, (d) at 21% quality with $NC=0.80$, (e) at 20% quality with $NC=0.78$, and (f) at 10% quality with $NC=0.56$.

The same set of simulation is applied to the Peppers image as well to test the robustness of the introduced watermarking scheme. Table 6.4 shows the NC values corresponding to the number of error bits detected for various kinds of attacked watermarked images of Peppers using the SCHAT and DFT based watermarking schemes. The corresponding recovered watermarks from the SCHAT watermarked images under various attack operations are shown in Figure 6.12. Likewise, Table 6.5 lists the NC values for the JPEG compressed watermarked Peppers images at various quality levels using the SCHAT and DFT based watermarking schemes. The respective recovered watermarks from the JPEG encoded versions of the SCHAT watermarked images at selected quality levels are shown in Figure 6.13. We have observed that the results are consistent with the previous findings.

Attack type	Parameters	SCHAT		DFT	
		eBits ^a	NC	eBits ^a	NC
(a) Scaling	512×512 to 192×192	49	0.95	4	0.99
(b) Rotating	90° rotated clockwise	0	1.00	0	1.00
(c) Cropping I	Cropping 1/4	36	0.96	35	0.96
(d) Cropping II	Central portion remained	121	0.87	122	0.87
(e) Painting	Painting 3 bars	3	0.99	1	0.99
(f) Low-pass filtering	Size 3×3 , $\sigma = 1.4$	8	0.99	0	1.00
(g) Sharpening	Sharpening the image	0	1.00	0	1.00
(h) Noising	Gaussian noise added	121	0.88	125	0.87
(i) Phase perturbation	Mean= $\pi/4$, var=0.01	0	1.00	0	1.00

^a The number of error bits detected.

Table 6.4: Comparison of the NC values corresponding to the number of error bits for the attacked watermarked images of Peppers under various operations between the SCHAT and the DFT based schemes.

Figure 6.12: The extracted watermarks from the attacked SCHT watermarked images of Peppers: (a) The scaled, (b) the rotated, (c) the cropped (a quarter image), (d) the cropped (central portion remained), (e) the painted, (f) the filtered, (g) the sharpened, (h) the noised, and (i) the phase perturbation attacked.

JPEG quality level(%)	SCHT		DFT	
	eBits ^a	NC	eBits ^a	NC
100	0	1.00	0	1.00
90	0	1.00	0	1.00
80	0	1.00	0	1.00
70	0	1.00	0	1.00
60	2	0.99	0	1.00
50	2	0.99	0	1.00
40	6	0.99	6	0.99
30	23	0.98	129	0.87
21	191	0.81	329	0.67
20	222	0.78	362	0.64
10	462	0.55	466	0.55

^a The number of error bits detected.

Table 6.5: Comparison of the NC values corresponding to the number of errors for the JPEG compressed watermarked images of Peppers between the SCHT and DFT based schemes.

Figure 6.13: The extracted watermarks from the JPEG encoded versions of the SCHT watermarked image of Peppers: (a) at 50% quality with $NC \cong 1.00$, (b) at 40% quality with $NC \cong 1.00$, (c) at 30% quality with $NC = 0.98$, (d) at 21% quality with $NC = 0.81$, (e) at 20% quality with $NC = 0.78$, and (f) at 10% quality with $NC = 0.55$.

6.5.2 Comparison between the SCHT and DFT Based Watermarking Schemes

The comparison of the NC values corresponding to the extracted watermarks between the proposed methods using the SCHT and the DFT is listed in Tables 6.2 and 6.3 for the Lena image. It can be seen from Table 6.2 that the NC values for both schemes are close to each other under various kinds of attacks and they are all above the threshold value 0.8. The extracted watermarks are clearly identified by the human perception as shown in Figure 6.10. For the JPEG lossy compression attacks, the scheme using the SCHT is able to provide higher NC value (which is at the threshold value 0.8) than that employing the DFT at 21% quality level as shown in Table 6.3. This indicates that the SCHT based watermarking scheme is robust to the JPEG compression up to 21% quality factor (according to the threshold defined) whereas the DFT based scheme cannot sustain. The same conclusion can also be applied for the simulation results where Peppers is used as the test image (See Tables 6.4, 6.5, and Figures 6.12, 6.13). Besides, the PSNR values of the SCHT watermarked images (which are 39.95 dB for the Lena image and 40.31 dB for the Peppers image) are even higher than the

corresponding values of the DFT watermarking images (which are 37.72 dB for the Lena image and 39.47 dB for the Peppers image) as shown in Figures 6.7 and 6.8. Therefore, the DFT can be replaced with the SCHT for such applications with simpler implementation plus consequent savings in computational cost [62].

6.5.3 Comparison with the Chen's Scheme

Table 6.6 shows the comparison of the robustness of the proposed scheme as compared with the Chen's scheme [55] under various image manipulation attacks. As mentioned in the table, both schemes are able to sustain different attacks such as scaling, rotating, image cropping, painting and blurring. But the Chen's scheme is vulnerable to the phase perturbation attack which is described in Ref. [63] whereas our scheme is able to withstand such attacks. In addition, the PSNR value of the watermarked image using the proposed method with respect to the original image is 39.95 dB which is larger than the corresponding value in [55], which is 35.34 dB for the Lena image.

Attack item	Our scheme	The Chen's scheme [55]
Scaling	Pass	Pass
Rotating	Pass	Pass
Cropping I	Pass	Pass
Painting	Pass	Pass
Blurring	Pass	Pass
Phase perturbation [63]	Pass	Fail

Table 6.6: Performance comparison between the proposed scheme and the Chen's scheme under different image manipulation attacks for the Lena image.

6.5.4 Comparison with the Guo's Scheme

In [53], Guo et al. introduced the discrete fractional random transform to develop a novel watermarking algorithm. The mean square error (MSE) was used to measure the similarity between the original and extracted watermarks, which is defined as

$$\text{MSE} = \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{M-1} [W(i, j) - W_R(i, j)]^2}{\sum_{i=0}^{M-1} \sum_{j=0}^{M-1} [W(i, j)]^2}, \quad (6.16)$$

where $W(i, j)$ represents the $(i\text{th}, j\text{th})$ pixel value of the original watermark image W , and $W_R(i, j)$ denotes the pixel value at $(i\text{th}, j\text{th})$ location of the extracted watermark image of $(M \times M)$ pixels. It is easy to realize that smaller MSE values mean better extracted similarity and higher resistance to the attacks.

Table 6.7 shows the comparison of the MSE values between our scheme and the

Attack category	Our scheme	The Guo's scheme [53]
Cropping 1/4	0.0902	0.5970
Cropping 1/16	0.0083	0.4160
Low-pass filtering, $[4 \ 4]^1$	0.0166	0.2940
$[3 \ 3]^2$	0.0024	0.1130
Noising, $\sigma = 43.39$	0.2740	0.7530
$\sigma = 36.16$	0.2384	0.6930
$\sigma = 28.93$	0.1590	0.5820
$\sigma = 21.69$	0.1008	0.4550
$\sigma = 14.46$	0.0297	0.3230
$\sigma = 7.23$	0.0000	0.1840

¹ Filter size 4×4 .

² Filter size 3×3 .

Table 6.7: Comparison of the MSE values between the proposed scheme and the Guo's scheme under different kinds of attacks for the Lena image.

Guo's scheme under different kinds of attacks such as cropping, low-pass filtering and noising, in which Lena is used as the test image in the simulation. It is noted that the attack operations which are listed in Ref. [53] are considered in making comparison to show the robustness of the schemes. For the cropping attack, the watermarked image is cropped (1/4) and (1/16) before proceeding the extracting process. For the low-pass filtering, the watermarked image is filtered by using the filters which are the rotationally symmetric Gaussian low-pass filters of size (3×3) and (4×4) with the standard deviation σ of 0.5 before proceeding the extracting process. For the noising attack, the random noise is added to the watermarked image where the noise is randomly distributed with zero mean and its standard deviation σ is ranged from nearly (10-40) before proceeding the extracting process.

The results in Table 6.7 reveal that our proposed scheme is able to provide smaller MSE values than those of the Guo's scheme, and the differences are also significant for each attack as shown in the table. The recovered watermarks corresponding to the MSE values of our scheme for different attacks are illustrated in Figure 6.14 a-j. It can be seen from the figures that all the extracted watermarks are fairly recognized by

Figure 6.14: The extracted watermarks corresponding to the MSE values under various attacks using the introduced scheme: (a) The cropped (1/4), (b) the cropped (1/16), (c) the filtered (4×4) , (d) the filtered (3×3) , and the noised with (e) $\sigma = 43.39$, (f) $\sigma = 36.16$, (g) $\sigma = 28.93$, (h) $\sigma = 21.69$, (i) $\sigma = 14.46$, (j) $\sigma = 7.23$.

human vision and the marks can be recovered from the attacked watermarked images. Therefore, it can be concluded that our scheme is more robust to the attacks than the Guo's scheme. But it is found that the PSNR value (54.24 dB) of the embedded image, which is obtained in the Guo's scheme, is higher than that of our scheme (39.95 dB) for the Lena image.

6.6 Conclusion

A robust phase watermarking scheme for grey-scaled digital still images using the sequency-ordered complex Hadamard transform is presented in this chapter. The main analysis tool is the SCHAT which transforms the host image from the spatial domain into the sequency domain where the embedding and extracting procedures are performed to obtain the watermarked image and recover the watermark. In fact, SCHAT is a complex Hadamard transform whose transform coefficients of a real image are the complex numbers comprised of the amplitudes and the phases. Hence, the PSK modulation technique can be easily incorporated to the watermarking scheme for embedding of watermark, which aids the scheme to be more robust to the attacks due to the well-known fact that modulating the phase of carrier is able to achieve better performance and superior noise immunity than amplitude modulation in communication theory. Besides, sequency in the SCHAT is analogous to frequency in the DFT. Therefore, sequency-based image analysis can be utilized to search for the appropriate locations in the transformed image for watermark embedding process such that the visual degradation can be maintained at an acceptable level in the watermarked copy.

Section 6.2 reviews the previous works done by the others especially for various robust watermarking schemes using the transform domain approaches. Section 6.3 presents the watermark embedding algorithm in the sequency domain using the SCHAT. The transform is computed based on the 8×8 image blocks. The LABS is performed on

the whole transformed image to reduce the visual degradation of the host image caused by watermark concealing whereas the AB method is used to enhance the resistance of the watermarked image to several attacks. The visually recognizable symbol is used as the watermark in this watermarking scheme to represent the image identity. The pseudo-random permutation is used to enhance the security of the watermark and the SS method is also adopted to reinforce the robustness of the watermarking scheme as well as to ensure a large measure of security against unintentional or intentional attacks. The watermark is concealed into the host image using the PSK modulation in the sequency domain. By using the analogy of sequency with frequency, the watermark bits are inserted into the lower sequency components of the image, which are most significant to the image integrity.

Section 6.4 provides the watermark detection algorithm in detail. During the extracting process, it is important that the secret bits are decoded from the same positions where they were embedded. The same set of user's keys (Key, PN_1, PN_1) used in the embedding process is reused in the extracting process as well. The PSK demodulation, inverse SS and inverse pseudo-random permutation are performed to reverse the operations of the PSK modulation, the SS method and pseudo-random permutation employed during embedding.

Section 6.5 discusses the experimental results and comparisons with other watermarking schemes. The numerical results show that the proposed scheme is robust to various kinds of attacks such as JPEG lossy compression, scaling, rotating, cropping, painting, low-pass filtering, sharpening, noising and phase perturbation attack. Compared to the Chen's scheme, our scheme is able to provide higher PSNR value of the watermarked image with respect to the original image. In addition, it can also sustain the phase perturbation attack which is described in [63] while maintaining the visual quality of the attacked image at the acceptable level as shown in Figure 6.9-i. It is also investigated that the proposed scheme performs better than the Guo's scheme in

terms of the MSE values between the original and extracted watermarks. Comparison between the proposed methods using the SHT and the DFT shows that the SHT can be considered as a good candidate to replace the DFT for such applications. In terms of computational complexity and implementation, the SHT is much faster and simpler than the DFT with the availability of its fast algorithm and pipelined hardware structure (as pointed out in Chapter 4). Therefore, together with the supporting of numerical results, it is demonstrated that SHT can be considered as a potential and useful tool for digital watermarking applications.

Chapter 7

Conjugate Symmetric Sequency-Ordered Complex Hadamard Transform (CS-SCHT)

7.1 Introduction

As mentioned in Chapter 2, the ordering of Hadamard transforms describes the sequence in which the Walsh functions are positioned in the transform matrix. For instance, the WHT matrix simply expresses that a sampled Walsh function series is arranged in natural or Hadamard order. If the Walsh function values are arranged in ascending values of sequencies in the transform matrix, a sequency-ordered Walsh transform (SOWT) matrix is obtained. In most applications of signal processing, sequency ordering is preferred due to its analogy to frequency in the discrete Fourier transform (DFT) [64, 65]. But the abovementioned transforms are real Hadamard transforms and they are usually applied to real inputs for analysis and synthesis. A set of complex orthogonal transforms known as unified complex Hadamard transforms (UCHTs) is introduced [10, 11] for such needs especially for multiple-valued

logic design and the applications dealing with complex-valued functions. The transform matrices are confined to four complex integer values $\{\pm 1, \pm j\}$ where $j = \sqrt{-1}$. But the UCHT matrices do not exhibit certain ordering which may correspond to the WHT or the SOWT. Therefore, the SCHT which is the complex Hadamard transform and its ordering corresponds to its integer-valued real counterpart (i.e., the SOWT) is introduced in this thesis. Thereafter, the question arises whether complex Hadamard transform whose ordering may follow that of the WHT exists or not. Consequently, the natural-ordered complex Hadamard transform (NCHT) which order corresponds to the WHT is further derived and the relationship between the orderings of the SCHT and the NCHT is also investigated. We now focus on the SCHT we introduced earlier.

Sequency which is defined previously in the SCHT is actually complex sequency which describes the amount of rotation (the number of zero crossings) of each row vector of an SCHT matrix in the unit circle of a complex plane. This is analogous to frequency in the DFT if frequency is defined as the the number of times that each row vector of a DFT matrix crosses the imaginary axis in the unit circle per unit time interval. But in the DFT, the term frequency refers to periodic repetition rate of sinusoidal waves as well as the rotation of individual row vector in a DFT matrix in the unit circle over a normalized time base $0 \leq t \leq 1$. This is because the DFT transform matrix is conjugate-symmetric and we can intuitively regroup the row vectors of a DFT matrix to obtain the corresponding sinusoidal waves with respective frequencies (which is to be explained in Section 7.3). The rotation of each row vector of a DFT matrix in the unit circle and the periodic repetition rate of corresponding sinusoidal waves are observed to be the same, which define frequency in the DFT. Therefore, it is then the purpose of this chapter to introduce a new version of SCHT called conjugate symmetric sequency-ordered complex Hadamard transform (CS-SCHT for short) whose sequency is directly related to frequency in the DFT and which is expected to outperform previous introduced SCHT in spectrum

estimation. Besides, the spectrum formed by the CS-SCHT is conjugate-symmetric so only half of spectral coefficients are needed for analysis and synthesis. This in turn reduces the memory requirement in processing for the applications such as real-time image watermarking and spectrum estimation. In fact, this is advantageous over the SCHT. Because the SCHT spectral coefficients are complex numbers and they are not conjugate symmetric, hence, more memories are needed to store the coefficients for analysis and synthesis in transform implementation.

The rest of this chapter is organized as follows. Section 7.2 describes the construction of the CS-SCHT transform matrix based on conjugate symmetric natural-ordered complex Hadamard transform (CS-NCHT) whose construction is based on the WHT matrix and the direct block matrix operator. The interpretation of sequency in the CS-SCHT is presented in Section 7.3. The properties of the CS-SCHT are derived and analyzed in Section 7.4, including the exponential form of the CS-SCHT and the dyadic shift invariant CS-SCHT power spectrum and its proof. Using the sparse matrix factorization approach, a fast algorithm to compute an N -point CS-SCHT is derived in Section 7.5 and its computational complexity is examined. Some applications of the CS-SCHT are suggested in Section 7.6 together with the supporting simulation results and discussions. Finally, this chapter is concluded in Section 7.7.

7.2 Basic Definitions of the CS-SCHT

In this section, we shall present the construction of the CS-SCHT. Firstly, we define the CS-NCHT based on the WHT and the direct block matrix operator (see Subsection 3.3.1). The construction of the CS-SCHT is different from that of the SCHT as the SCHT matrices are generated based on the complex Rademacher matrices [66]. Let $\mathbf{H}_N$ be any CS-NCHT matrix of dimension $N \times N$ where $N = 2^n$. Then, it is a square

matrix defined as

$$\mathbf{H}_N = \left[\begin{array}{c|c} \mathbf{H}_{N/2} & \mathbf{H}_{N/2} \\ \hline \mathbf{H}'_{N/2} \mathbf{S}_{N/2} & -\mathbf{H}'_{N/2} \mathbf{S}_{N/2} \end{array} \right] \quad (7.1)$$

which is similar to the construction of the NCHT matrix defined in (3.65) except that $\mathbf{H}_{N/2}$ is replaced with $\mathbf{H}'_{N/2}$ which is defined recursively as

$$\mathbf{H}'_{N/2} = \left[\begin{array}{c|c} \mathbf{H}'_{N/4} & \mathbf{H}'_{N/4} \\ \hline \mathbf{H}'_{N/4} \mathbf{I}'_{N/4} & -\mathbf{H}'_{N/4} \mathbf{I}'_{N/4} \end{array} \right] \quad (7.2)$$

where $(N/4) = 2^{n-2}$,

$$\mathbf{I}'_{2^{n-2}} = \left[\begin{array}{cc} \mathbf{I}_{2^{n-3}} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I}_{2^{n-3}} \end{array} \right] \quad (7.3)$$

is the identity matrix of size $2^{n-2} \times 2^{n-2}$ where the bottom half of the elements are multiplied by (-1) , and

$$\mathbf{S}_{2^{n-1}} = \left[\begin{array}{cc} \mathbf{I}_{2^{n-2}} & \mathbf{0} \\ \mathbf{0} & j\mathbf{I}_{2^{n-2}} \end{array} \right]$$

as defined in (3.26). The following expressions illustrate the recursive decomposition of an 8×8 CS-NCHT matrix as an example. First, it is expressed as

$$\mathbf{H}_8 = \left[\begin{array}{c|c} \mathbf{H}_4 & \mathbf{H}_4 \\ \hline \mathbf{H}'_4 \mathbf{S}_4 & -\mathbf{H}'_4 \mathbf{S}_4 \end{array} \right]. \quad (7.4)$$

Substituting the expressions for $\mathbf{H}_4$ and $\mathbf{H}'_4$ into (7.4), it will become

$$\mathbf{H}_8 = \left[\begin{array}{cc|cc|c} \mathbf{H}_2 & \mathbf{H}_2 & \mathbf{H}_2 & \mathbf{H}_2 & \\ \mathbf{H}'_2 \mathbf{S}_2 & -\mathbf{H}'_2 \mathbf{S}_2 & \mathbf{H}'_2 \mathbf{S}_2 & -\mathbf{H}'_2 \mathbf{S}_2 & \\ \hline \left[\begin{array}{cc} \mathbf{H}'_2 & \mathbf{H}'_2 \\ \mathbf{H}'_2 \mathbf{I}'_2 & -\mathbf{H}'_2 \mathbf{I}'_2 \end{array} \right] \mathbf{S}_4 & - & \left[\begin{array}{cc} \mathbf{H}'_2 & \mathbf{H}'_2 \\ \mathbf{H}'_2 \mathbf{I}'_2 & -\mathbf{H}'_2 \mathbf{I}'_2 \end{array} \right] \mathbf{S}_4 & & \end{array} \right]. \quad (7.5)$$

But since

$$\mathbf{H}'_2 = \mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

and substituting the corresponding values into (7.5), $\mathbf{H}_8$ can be expressed as

$$\mathbf{H}_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j & 1 & j & -1 & -j \\ 1 & -j & -1 & j & 1 & -j & -1 & j \\ 1 & 1 & j & j & -1 & -1 & -j & -j \\ 1 & -1 & j & -j & -1 & 1 & -j & j \\ 1 & -1 & -j & j & -1 & 1 & j & -j \\ 1 & 1 & -j & -j & -1 & -1 & j & j \end{bmatrix}. \quad (7.6)$$

Likewise, a 4×4 CS-NCHT matrix, $\mathbf{H}_4$, can be expressed as

$$\mathbf{H}_4 = \left[\begin{array}{c|c} \mathbf{H}_2 & \mathbf{H}_2 \\ \hline \mathbf{H}'_2 \mathbf{S}_2 & -\mathbf{H}'_2 \mathbf{S}_2 \end{array} \right] = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \\ 1 & -j & -1 & j \end{bmatrix}. \quad (7.7)$$

It should be noted that $\mathbf{H}_4$ is the smallest dimension we can obtain as a CS-NCHT matrix. Having explained the above derivations, any CS-NCHT matrix of dimension $N \times N$ with $N = 2^n$ can be defined using the WHT matrix and the direct block matrix operator as follows:

$$\mathbf{H}_N = \mathbf{W}_N \diamond \mathbf{A}'_{n-1,n-1} \diamond \cdots \diamond \mathbf{A}'_{2,2} \diamond \mathbf{A}_{1,1} \quad (7.8)$$

where $\mathbf{W}_N$ is the $N \times N$ WHT matrix, $\diamond$ represents the direct block matrix operator as defined in Subsection 3.3.1, and

$$\begin{aligned}
\mathbf{A}_{1,1} &= [\mathbf{I}_{2^{n-1}}, \mathbf{S}_{2^{n-1}}]^T \\
\mathbf{A}'_{2,2} &= [\mathbf{I}_{2^{n-2}}, \mathbf{S}_{2^{n-2}}, \mathbf{I}_{2^{n-2}}, \mathbf{I}'_{2^{n-2}}]^T \\
&\vdots \\
\mathbf{A}'_{n-1,n-1} &= [\mathbf{I}_2, \mathbf{S}_2, \mathbf{I}_2, \mathbf{I}'_2, \dots, \mathbf{I}_2, \mathbf{I}'_2]^T
\end{aligned} \tag{7.9}$$

where $\{ \}^T$ denotes the transpose. Let us, for example, consider $N = 8$ and $n = \log_2 8 = 3$. Then, (7.8) becomes

$$\mathbf{H}_8 = \mathbf{W}_8 \diamond \mathbf{A}'_{2,2} \diamond \mathbf{A}_{1,1}. \tag{7.10}$$

Substituting the corresponding values defined in (7.9) into (7.10), we have

$$\mathbf{H}_8 = \left[\begin{array}{cc|cc} \mathbf{W}_2 & \mathbf{W}_2 & \mathbf{W}_2 & \mathbf{W}_2 \\ \mathbf{W}_2 & -\mathbf{W}_2 & \mathbf{W}_2 & -\mathbf{W}_2 \\ \hline \mathbf{W}_2 & \mathbf{W}_2 & -\mathbf{W}_2 & -\mathbf{W}_2 \\ \mathbf{W}_2 & -\mathbf{W}_2 & -\mathbf{W}_2 & \mathbf{W}_2 \end{array} \right] \diamond \left[\begin{array}{c} \mathbf{I}_2 \\ \mathbf{S}_2 \\ \mathbf{I}_2 \\ \mathbf{I}'_2 \end{array} \right] \diamond \left[\begin{array}{c} \mathbf{I}_4 \\ \mathbf{S}_4 \end{array} \right] \tag{7.11}$$

where $\mathbf{W}_2$ is the 2×2 WHT matrix. After multiplied using the direct block matrix operator, it is obtained as

$$\mathbf{H}_8 = \left[\begin{array}{cc|cc} \mathbf{W}_2 & \mathbf{W}_2 & \mathbf{W}_2 & \mathbf{W}_2 \\ \mathbf{W}_2 \mathbf{S}_2 & -\mathbf{W}_2 \mathbf{S}_2 & \mathbf{W}_2 \mathbf{S}_2 & -\mathbf{W}_2 \mathbf{S}_2 \\ \hline \left[\begin{array}{cc} \mathbf{W}_2 & \mathbf{W}_2 \\ \mathbf{W}_2 \mathbf{I}'_2 & -\mathbf{W}_2 \mathbf{I}'_2 \end{array} \right] \mathbf{S}_4 & - & \left[\begin{array}{cc} \mathbf{W}_2 & \mathbf{W}_2 \\ \mathbf{W}_2 \mathbf{I}'_2 & -\mathbf{W}_2 \mathbf{I}'_2 \end{array} \right] \mathbf{S}_4 \end{array} \right] \tag{7.12}$$

which is actually similar to the expression obtained in (7.5) since

$$\mathbf{W}_2 = \mathbf{H}'_2 = \mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

As such any CS-NCHT matrix of dimension $N \times N$ can be achieved. Having defined the CS-NCHT, a CS-SCHT matrix can be obtained from a CS-NCHT matrix through a bit reversal conversion (as mentioned in Subsection 3.3.4) and vice versa. Let $\mathbf{H}_N^S$ be any CS-SCHT matrix of size $N \times N$ and also $\mathbf{H}_N^N$ be any CS-NCHT matrix of size $N \times N$. Then it is defined as

$$\mathbf{H}_N^S(p, k) = \mathbf{H}_N^N(b(p), k) \quad (7.13)$$

where p and k are the row and column indices of a matrix such that $0 \leq p, k \leq N-1$ and $b(p)$ is the decimal number obtained by the bit-reversed operation of the decimal p . As an example, $\mathbf{H}_8^S$ is obtained as

$$\mathbf{H}_8^S = \begin{bmatrix} \mathbf{H}_8^N(0, k) \\ \mathbf{H}_8^N(4, k) \\ \mathbf{H}_8^N(2, k) \\ \mathbf{H}_8^N(6, k) \\ \mathbf{H}_8^N(1, k) \\ \mathbf{H}_8^N(5, k) \\ \mathbf{H}_8^N(3, k) \\ \mathbf{H}_8^N(7, k) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & j & j & -1 & -1 & -j & -j \\ 1 & j & -1 & -j & 1 & j & -1 & -j \\ 1 & -1 & -j & j & -1 & 1 & j & -j \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & j & -j & -1 & 1 & -j & j \\ 1 & -j & -1 & j & 1 & -j & -1 & j \\ 1 & 1 & -j & -j & -1 & -1 & j & j \end{bmatrix} \quad (7.14)$$

in which the row vectors of the matrix are arranged in increasing order of zero crossings in the unit circle of a complex plane. Besides, the p th row of the matrix is

the conjugate of the $(N - p)$ th row vector where $p = 1, 2, \dots, (N/2) - 1$ and $N = 2^n$, hence, the spectrum obtained by using this matrix is shown to be conjugate-symmetric. The 0th and $(N/2)$ th row vectors correspond to the DC and Nyquist frequency components in the DFT matrix. As such any CS-SCHT matrix of dimension $N \times N$ can be generated. This completes the construction of the CS-SCHT. Having developed the CS-SCHT matrix, the CS-SCHT of an N -point complex signal vector $\mathbf{x}_N = [x(0), x(1), \dots, x(N - 1)]^T$ is defined as

$$\mathbf{X}_N = \frac{1}{N} \overline{\mathbf{H}_N} \mathbf{x}_N \quad (7.15)$$

where $\mathbf{X}_N = [X(0), X(1), \dots, X(N - 1)]^T$ is the transformed complex column vector and $\overline{\{\}} \}$ denotes the complex conjugate and $\mathbf{H}_N$ is the CS-SCHT matrix which is defined in (7.13). The data sequence can be uniquely recovered from the inverse transform, that is,

$$\mathbf{x}_N = \mathbf{H}_N^T \mathbf{X}_N \quad (7.16)$$

where $\mathbf{H}_N^T \overline{\mathbf{H}_N} = \mathbf{H}_N^* \mathbf{H}_N = N \mathbf{I}_N$ (unitary property) where $\{\}^*$ represents the complex conjugate transpose.

7.3 Interpretation of Sequency

The complex Hadamard matrix generated in (7.13) exhibits sequency ordering, that is, the row vectors of the CS-SCHT matrix are positioned in an ascending order of zero crossings in the unit circle of a complex plane. As indicated earlier, sequency in the CS-SCHT is analogous to frequency in the DFT. In order to illustrate this analogy, let us see the DFT matrix and understand the concept of frequency. Considering $N = 8$

as an example, the DFT matrix will be

$$\mathbf{F}_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & W_8^1 & W_8^2 & W_8^3 & W_8^4 & W_8^5 & W_8^6 & W_8^7 \\ 1 & W_8^2 & W_8^4 & W_8^6 & W_8^0 & W_8^2 & W_8^4 & W_8^6 \\ 1 & W_8^3 & W_8^6 & W_8^1 & W_8^4 & W_8^7 & W_8^2 & W_8^5 \\ 1 & W_8^4 & W_8^0 & W_8^4 & W_8^0 & W_8^4 & W_8^0 & W_8^4 \\ 1 & W_8^5 & W_8^2 & W_8^7 & W_8^4 & W_8^1 & W_8^6 & W_8^3 \\ 1 & W_8^6 & W_8^4 & W_8^2 & W_8^0 & W_8^6 & W_8^4 & W_8^2 \\ 1 & W_8^7 & W_8^6 & W_8^5 & W_8^4 & W_8^3 & W_8^2 & W_8^1 \end{bmatrix} \quad (7.17)$$

where W_N is the twiddle factor in the DFT matrix and $W_N^{pk} = \exp(-j\frac{2\pi pk}{N}) = \cos(\frac{2\pi pk}{N}) - j \sin(\frac{2\pi pk}{N})$ with $p, k = 0, 1, \dots, N-1$. The row vectors of this matrix are also arranged in ascending order of zero crossings in the unit circle of a complex plane. The real parts in W_N^{pk} represents the cosine term whereas the imaginary parts are the sine terms. In order to obtain the corresponding sinusoidal waves from the above matrix, we can intuitively separate the real and imaginary terms and regroup them as follows:

$$\mathbf{F}_{R,8} = \begin{bmatrix} \mathbf{F}_8(0, k) \\ \frac{1}{2}(\Im\{\mathbf{F}_8(1, k)\} - \Im\{\mathbf{F}_8(7, k)\}) \\ \frac{1}{2}(\Re\{\mathbf{F}_8(1, k)\} + \Re\{\mathbf{F}_8(7, k)\}) \\ \frac{1}{2}(\Im\{\mathbf{F}_8(2, k)\} - \Im\{\mathbf{F}_8(6, k)\}) \\ \frac{1}{2}(\Re\{\mathbf{F}_8(2, k)\} + \Re\{\mathbf{F}_8(6, k)\}) \\ \frac{1}{2}(\Im\{\mathbf{F}_8(3, k)\} - \Im\{\mathbf{F}_8(5, k)\}) \\ \frac{1}{2}(\Re\{\mathbf{F}_8(3, k)\} + \Re\{\mathbf{F}_8(5, k)\}) \\ \mathbf{F}_8(4, k) \end{bmatrix} \Rightarrow \begin{bmatrix} \text{DC} \\ \text{sine wave (1st harmonic)} \\ \text{cosine wave (1st harmonic)} \\ \text{sine wave (2nd harmonic)} \\ \text{cosine wave (2nd harmonic)} \\ \text{sine wave (3rd harmonic)} \\ \text{cosine wave (3rd harmonic)} \\ \text{Nyquist frequency} \end{bmatrix} \quad (7.18)$$

where $\Re\{\cdot\}$ and $\Im\{\cdot\}$ represent real and imaginary parts of a complex number, respectively. The sinusoidal waveforms corresponding to each row vector of (7.18) are illustrated in Figure 7.1. Frequency is defined as periodic repetition rate of the sinusoidal waves shown in the figure.

By using this analogy, we can also obtain the respective waveforms from the CS-SCHT matrix (since the matrix is also conjugate-symmetric and sequency-ordered). As an example, this process can be demonstrated using an 8×8 CS-SCHT matrix as shown below. The new real transform matrix obtained from the 8×8 CS-SCHT

Figure 7.1: The sine and cosine waveforms in the DFT with increasing harmonic order (frequencies).

matrix is denoted as $\mathbf{H}_{R,8}$ and

$$\mathbf{H}_{R,8} = \begin{bmatrix} \mathbf{H}_8(0, k) \\ \frac{1}{2}(\Im\{\mathbf{H}_8(1, k)\} - \Im\{\mathbf{H}_8(7, k)\}) \\ \frac{1}{2}(\Re\{\mathbf{H}_8(1, k)\} + \Re\{\mathbf{H}_8(7, k)\}) \\ \frac{1}{2}(\Im\{\mathbf{H}_8(2, k)\} - \Im\{\mathbf{H}_8(6, k)\}) \\ \frac{1}{2}(\Re\{\mathbf{H}_8(2, k)\} + \Re\{\mathbf{H}_8(6, k)\}) \\ \frac{1}{2}(\Im\{\mathbf{H}_8(3, k)\} - \Im\{\mathbf{H}_8(5, k)\}) \\ \frac{1}{2}(\Re\{\mathbf{H}_8(3, k)\} + \Re\{\mathbf{H}_8(5, k)\}) \\ \mathbf{H}_8(4, k) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & -1 & -1 \\ 1 & 1 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 \\ 1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix} \quad (7.19)$$

where $\mathbf{H}_8$ is the CS-SCHT matrix of size 8×8 . The waveforms which are relevant to each row vector of (7.19) are shown in Figure 7.2 together with the corresponding sinusoidal waveforms from the DFT. It can be seen from the figure that the CS-SCHT is an approximation of sine and cosine waveforms in the DFT by using the pertinent step waveforms. Sequency can be defined as periodic repetition rate of the approximated waveforms. On the other hand, it can be defined as one half of the number of zero crossings of each row vector of a matrix in the unit circle of a complex plane per unit time interval. For example, sequency of the p th row vector of a CS-SCHT matrix is p as the number of zero crossings is $2p$. Since the p th and $(N - p)$ th row vectors of the CS-SCHT matrices where $p = 1, 2, \dots, (N/2) - 1$ are conjugate-symmetric, they represent the same sequency except that they have different directions of rotations in the unit circle, that is, clockwise and anti-clockwise directions. This is closely related to frequency in the DFT in which positive frequency and negative frequency (which correspond to the conjugate-symmetric row vectors in the DFT matrix) rotate in opposite directions as well. It is observed that the phase shifts are not the same each time the row vector of a CS-SCHT matrix rotates in the

Figure 7.2: Approximation of the sinusoidal waves in the DFT using the corresponding waveforms from the CS-SCHT.

unit circle, however, the total phase shifts of each row vector of the CS-SCHT matrix are the same as that of the DFT matrix having the same dimension. It should be noted that frequency can be regarded as a special measure of sequency applicable to the sinusoidal waveforms only.

7.4 Properties of the CS-SCHT

In this section, we shall present the properties of the CS-SCHT.

Property 1: Exponential Property of the CS-SCHT

Let $h(p, k)$ be the element at the p th row and k th column of any CS-SCHT matrix

of dimension $N \times N$ where $N = 2^n$ and $0 \leq p, k \leq N - 1$. Then,

$$h(p, k) = (-1)^{\sum_{r=0}^{n-1} g_r k_r} (-j)^{\sum_{r=0}^{n-1} f_r k_r} \quad (7.20)$$

where $n = \log_2 N$ and k_r is the r th bit of the binary representation of the decimal integers k , i.e., $(k)_{10} = (k_{n-1}, k_{n-2}, \dots, k_0)_2$. Besides, g_r is defined to be the r th bit of the binary Gray code [7] of bit reversal representation of the decimal p , and we also define f_r to be the r th bit of the binary bits of the highest power of 2 in $b(p)/2$ where $b(p)$ is the decimal number obtained through a bit-reversed conversion of the decimal p . The example for $N = 8$ is illustrated in Table 7.1 to achieve the binary values for the above notations. Subsequently, the construction of an 8×8 CS-SCHT matrix is further detailed in Tables 7.2, 7.3 and 7.4. The complex conjugate of the CS-SCHT matrix can be generated easily by just changing $-j$ to j in (7.20).

p	Binary	Bit Reversal	Gray Code, g_r	Bit-Reversed Decimal, $b(p)$	$b(p)/2$	Highest power of 2	f_r
0	000	000	000	0	0	0	000
1	001	100	110	4	2	2	010
2	010	010	011	2	1	1	001
3	011	110	101	6	3	2	010
4	100	001	001	1	0.5	0	000
5	101	101	111	5	2.5	2	010
6	110	011	010	3	1.5	1	001
7	111	111	100	7	3.5	2	010

Table 7.1: Binary representations for g_r and f_r .

		k_r							
		000	001	010	011	100	101	110	111
g_r	000	1	1	1	1	1	1	1	1
	110	1	1	-1	-1	-1	-1	1	1
	011	1	-1	-1	1	1	-1	-1	1
	101	1	-1	1	-1	-1	1	-1	1
	001	1	-1	1	-1	1	-1	1	-1
	111	1	-1	-1	1	-1	1	1	-1
	010	1	1	-1	-1	1	1	-1	-1
	100	1	1	1	1	-1	-1	-1	-1

Table 7.2: Elements generated from $(-1)^{\sum_{r=0}^{n-1} g_r k_r}$.

		k_r							
		000	001	010	011	100	101	110	111
f_r	000	1	1	1	1	1	1	1	1
	010	1	1	$-j$	$-j$	1	1	$-j$	$-j$
	001	1	$-j$	1	$-j$	1	$-j$	1	$-j$
	010	1	1	$-j$	$-j$	1	1	$-j$	$-j$
	000	1	1	1	1	1	1	1	1
	010	1	1	$-j$	$-j$	1	1	$-j$	$-j$
	001	1	$-j$	1	$-j$	1	$-j$	1	$-j$
	010	1	1	$-j$	$-j$	1	1	$-j$	$-j$

Table 7.3: Elements generated from $(-j)^{\sum_{r=0}^{n-1} f_r k_r}$.

		k_r							
		000	001	010	011	100	101	110	111
g_r	000	1	1	1	1	1	1	1	1
	110	1	1	j	j	-1	-1	$-j$	$-j$
	011	1	j	-1	$-j$	1	j	-1	$-j$
	101	1	-1	$-j$	j	-1	1	j	$-j$
	001	1	-1	1	-1	1	-1	1	-1
	111	1	-1	j	$-j$	-1	1	$-j$	j
	010	1	$-j$	-1	j	1	$-j$	-1	j
	100	1	1	$-j$	$-j$	-1	-1	j	j

Table 7.4: Elements generated from $(-1)^{\sum_{r=0}^{n-1} g_r k_r} \cdot (-j)^{\sum_{r=0}^{n-1} f_r k_r}$.

Property 2: Unitarity

Similar to the SCHT, the CS-SCHT is also an orthogonal transform whose row vectors are orthogonal in the complex domain as

$$|\det \mathbf{H}_N| = N^{N/2} \quad (7.21)$$

and

$$\mathbf{H}_N \mathbf{H}_N^* = \mathbf{H}_N^* \mathbf{H}_N = N \mathbf{I}_N \quad (7.22)$$

where $\mathbf{H}_N$ symbolizes any CS-SCHT matrix of dimension $N \times N$.

Property 3: Linearity

The CS-SCHT is also a linear transformation satisfying (3.24) as mentioned in Subsection 3.2.4.

Property 4: Parseval's Theorem

The energy of the signal in the time and sequency domains is expressed as

$$\frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2 = \sum_{k=0}^{N-1} |X(k)|^2 \quad (7.23)$$

where $X(k)$ represents the CS-SCHT coefficients in the sequency domain and $x(n)$ is the signal in the time domain.

Property 5: Conjugate Symmetry

The spectrum obtained by the CS-SCHT is conjugate-symmetric, that is, the coefficients $X(k)$ and $X(N-k)$ are conjugate-symmetric pairs where $k = 1, 2, \dots, (N/2)-1$.

Property 6: Dyadic Shift Invariant Power Spectrum

Let $h(p, k)$ be any element of a CS-SCHT transformation matrix. Then the transformation is said to be *dyadic shift invariant* (DSI) as it satisfies [67]

$$h(p, k \oplus m) = \begin{cases} h(p, k) \cdot \overline{h(p, m)}, & \text{if both elements are imaginary} \\ h(p, k) \cdot h(p, m), & \text{otherwise} \end{cases} \quad (7.24)$$

and

$$||h(p, k)||^2 = h(p, k) \cdot \overline{h(p, k)} = 1 \quad (7.25)$$

where $\oplus$ represent the modulo-2 addition and $h(p, k) \in \{\pm 1, \pm j\}$.

Proof: Let $h(p, k)$ be the (p th, k th) element of any CS-SCHT matrix where $N = 2^n$ and $p, k = 0, 1, \dots, N-1$. From *Property 1*, the exponential form of the CS-SCHT is

$$\begin{aligned} h(p, k) &= (-1)^{\sum_{r=0}^{n-1} g_r k_r} (-j)^{\sum_{r=0}^{n-1} f_r k_r} \\ h(p, k \oplus m) &= (-1)^{\sum_{r=0}^{n-1} g_r (k_r \oplus m_r)} (-j)^{\sum_{r=0}^{n-1} f_r (k_r \oplus m_r)}. \end{aligned} \quad (7.26)$$

From Table 7.5, the Boolean function $g_r(k_r \oplus m_r)$ is equivalent to the function $g_r k_r \oplus$

g_r	k_r	m_r	$k_r \oplus m_r$	$g_r k_r$	$g_r m_r$	$g_r(k_r \oplus m_r)$	$g_r k_r \oplus g_r m_r$
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	0	1	1	1
1	1	0	1	1	0	1	1
1	1	1	0	1	1	0	0

Table 7.5: The truth table for $g_r(k_r \oplus m_r)$ and $g_r k_r \oplus g_r m_r$.

$g_r m_r$. Therefore, we have

$$\begin{aligned}
h(p, k \oplus m) &= (-1)^{\sum_{r=0}^{n-1} g_r(k_r \oplus m_r)} (-j)^{\sum_{r=0}^{n-1} f_r(k_r \oplus m_r)} \\
&= (-1)^{\sum_{r=0}^{n-1} g_r k_r \oplus g_r m_r} (-j)^{\sum_{r=0}^{n-1} f_r k_r \oplus f_r m_r}.
\end{aligned} \tag{7.27}$$

Now let us consider each term contained in (7.27). Then, from Table 7.6, it can be found that

$$\begin{aligned}
(-1)^{\sum_{r=0}^{n-1} g_r k_r \oplus g_r m_r} &= (-1)^{\sum_{r=0}^{n-1} g_r k_r + g_r m_r} \\
&= (-1)^{\sum_{r=0}^{n-1} g_r k_r + \sum_{r=0}^{n-1} g_r m_r} \\
&= (-1)^{\sum_{r=0}^{n-1} g_r k_r} \cdot (-1)^{\sum_{r=0}^{n-1} g_r m_r}.
\end{aligned} \tag{7.28}$$

k_r	m_r	$k_r \oplus m_r$	$k_r + m_r$	$(-1)^{k_r \oplus m_r}$	$(-1)^{k_r + m_r}$
0	0	0	0	1	1
0	1	1	1	-1	-1
1	0	1	1	-1	-1
1	1	0	2	1	1

Table 7.6: The truth table for $(-1)^{k_r \oplus m_r}$ and $(-1)^{k_r + m_r}$.

f_r	k_r	m_r	$\begin{matrix} f_r k_r \\ \oplus \\ f_r m_r \end{matrix}$	$\begin{matrix} f_r k_r \\ + \\ f_r m_r \end{matrix}$	$(-j)^{f_r k_r \oplus f_r m_r}$	$(-j)^{f_r k_r + f_r m_r}$
0	0	0	0	0	1	1
0	0	1	0	0	1	1
0	1	0	0	0	1	1
0	1	1	0	0	1	1
1	0	0	0	0	1	1
1	0	1	1	1	$-j$	$-j$
1	1	0	1	1	$-j$	$-j$
1	1	1	0	2	1	-1

Table 7.7: The truth table for $(-j)^{f_r k_r \oplus f_r m_r}$ and $(-j)^{f_r k_r + f_r m_r}$.

On the other hand, Table 7.7 shows that the functions $(-j)^{\sum_{r=0}^{n-1} f_r k_r \oplus f_r m_r}$ and $(-j)^{\sum_{r=0}^{n-1} f_r k_r + f_r m_r}$ are not equivalent whenever $f_r = k_r = m_r = 1$. Therefore, it is expressed that

$$(-j)^{\sum_{r=0}^{n-1} f_r k_r \oplus f_r m_r} = (-j)^{f_0 k_0 \oplus f_0 m_0} (-j)^{f_1 k_1 \oplus f_1 m_1} \dots (-j)^{f_{n-1} k_{n-1} \oplus f_{n-1} m_{n-1}}. \quad (7.29)$$

In order to covert $(b_{1,r} \oplus b_{2,r})$ to $(b_{1,r} + b_{2,r})$, consideration is taken into account only

if the two binary operands (i.e., $b_{1,r}$ and $b_{2,r}$) are both equal to one. As indicated in *Property 1*, f_r represents the binary code of highest power of 2 in $b(p)/2$ where $b(p)$ is the bit-reversed converted decimal number. Therefore, it has only one bit of 1 in its binary expression and the rest are 0 (see Table 7.1 for an example). Hence, (7.29) can be rewritten as

$$\begin{aligned}
& (-j)^{\sum_{r=0}^{n-1} f_r k_r \oplus f_r m_r} \\
&= (-j)^{f_0 k_0 + f_0 m_0} (-j)^{f_1 k_1 + f_1 m_1} \dots (-j)^{f_{n-1} k_{n-1} + f_{n-1} m_{n-1}} (-1)^{f_r k_r m_r} \\
&= (-j)^{\sum_{r=0}^{n-1} f_r k_r} \cdot (-j)^{\sum_{r=0}^{n-1} f_r m_r} \cdot (-1)^{\sum_{r=0}^{n-1} f_r k_r m_r}. \tag{7.30}
\end{aligned}$$

The term $(-1)^{\sum_{r=0}^{n-1} f_r k_r m_r}$ is included in the above expression (its value can be $\{\pm 1\}$) considering for the case where $f_r = k_r = m_r = 1$ as shown in Table 7.7. It is also observed from this expression that whenever $f_r = k_r = m_r = 1$, the corresponding elements $h(p, k)$ and $h(p, m)$ are imaginary and one of them is multiplied with (-1) , which in fact is similar to the conjugation of that imaginary element. Therefore, this will prove the requirement defined in (7.24). Finally, substituting (7.28) and (7.30) into (7.27), we obtain

$$\begin{aligned}
h(p, k \oplus m) &= (-1)^{\sum_{r=0}^{n-1} g_r k_r} \cdot (-1)^{\sum_{r=0}^{n-1} g_r m_r} \cdot (-j)^{\sum_{r=0}^{n-1} f_r k_r} \cdot (-j)^{\sum_{r=0}^{n-1} f_r m_r} \\
&\quad \cdot (-1)^{\sum_{r=0}^{n-1} f_r k_r m_r} \\
&= \begin{cases} h(p, k) \cdot \overline{h(p, m)}, & \text{if both elements are imaginary} \\ h(p, k) \cdot h(p, m), & \text{otherwise} \end{cases} \tag{7.31}
\end{aligned}$$

and also

$$||h(p, k)||^2 = h(p, k) \cdot \overline{h(p, k)} = 1. \tag{7.32}$$

This completes the proof of (7.24) and (7.25). The proof is completed here. It is noted that the power spectrum of a real-valued input signal obtained through the DSI

transformation is dyadic shift invariant [67]. Hence, the CS-SCHT power spectrum is also dyadic shift invariant.

7.5 Fast Algorithm of the CS-SCHT

This section will derive the fast algorithm to compute an N -point CS-SCHT using the matrix factorization method. Since the CS-SCHT is the bit-reversed counterpart of the CS-NCHT, the method applied to the latter is also applicable to the former and vice versa. In this section, we shall present the factorization method of the CS-NCHT matrix, which leads to the fast algorithms for the CS-NCHT as well as the CS-SCHT (the output is arranged in bit-reversed order after computing for the CS-NCHT). The forward transformation matrix defined in (7.15) is the conjugate of the CS-SCHT matrix, however, we start with the factoring of the CS-NCHT matrix to derive the fast algorithm. Let us, for example, consider $N = 8$. Then, the expression in (7.1) becomes

$$\mathbf{H}_8 = \left[\begin{array}{c|c} \mathbf{H}_4 & \mathbf{H}_4 \\ \hline \mathbf{H}'_4 \mathbf{S}_4 & -\mathbf{H}'_4 \mathbf{S}_4 \end{array} \right] \quad (7.33)$$

where

$$\mathbf{S}_4 = \left[\begin{array}{cc} \mathbf{I}_2 & \mathbf{0} \\ \mathbf{0} & j\mathbf{I}_2 \end{array} \right] \text{ and } \mathbf{H}'_4 = \left[\begin{array}{c|c} \mathbf{H}_2 & \mathbf{H}_2 \\ \hline \mathbf{H}_2 \mathbf{I}'_2 & -\mathbf{H}_2 \mathbf{I}'_2 \end{array} \right].$$

Equation (7.33) can be expressed as

$$\mathbf{H}_8 = \left[\begin{array}{cc} \mathbf{H}_4 & \mathbf{0} \\ \mathbf{0} & \mathbf{H}'_4 \end{array} \right] \left[\begin{array}{cc} \mathbf{I}_4 & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_4 \end{array} \right] \left[\begin{array}{cc} \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{I}_4 & -\mathbf{I}_4 \end{array} \right]. \quad (7.34)$$

Subsequently, $\mathbf{H}_4$ and $\mathbf{H}'_4$ can be further factorized and $\mathbf{H}_8$ becomes

$$\mathbf{H}_8 = \begin{bmatrix} \mathbf{H}_2 & & & \\ & \mathbf{H}_2 & & \\ & & \mathbf{H}_2 & \\ & & & \mathbf{H}_2 \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & & & \\ & \mathbf{S}_2 & & \\ & & \mathbf{I}_2 & \\ & & & \mathbf{I}'_2 \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & \mathbf{I}_2 & & \\ \mathbf{I}_2 & -\mathbf{I}_2 & & \\ & & \mathbf{I}_2 & \mathbf{I}_2 \\ & & \mathbf{I}_2 & -\mathbf{I}_2 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I}_4 & \\ & \mathbf{S}_4 \end{bmatrix} \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{I}_4 & -\mathbf{I}_4 \end{bmatrix} \quad (7.35)$$

where

$$\mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad \mathbf{S}_2 = \begin{bmatrix} 1 & 0 \\ 0 & j \end{bmatrix} \quad \text{and} \quad \mathbf{I}'_2 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

It is easy to obtain the factorization of $\overline{\mathbf{H}}_8$ by just changing j to $-j$ in the $\mathbf{S}$ matrices shown in (7.35). This factorization will result in fast forward CS-NCHT algorithm. As mentioned earlier, rearranging the output in a bit-reversed order will provide the CS-SCHT coefficients, which gives rise to the fast forward CS-SCHT algorithm. The decomposition of fast forward CS-SCHT for $N = 8$ is illustrated by the signal flow graph in Figure 7.3. For simplicity, the multiplication by the scaling factor $(1/8)$ is omitted in the figure, which in fact can be computed by the binary shifting operations without needing any arithmetic operations. As shown in Figure 7.3, the signal flow graph contains three stages and each stage requires eight addition/subtractions. For the trivial complex multiplications by factor $-j$, it needs $2^0 + 2^1 = 3$ complex multiplications for three stages. Therefore, it can be concluded that for an N -point CS-SCHT, the fast algorithm has $n = \log_2 N$ stages and each stage needs N addition/subtractions. In general, the total computational requirements for complex input

Figure 7.3: Signal flow graph illustrating the computation of an 8-point CS-SCHT.

data are $N \log_2 N$ complex addition/subtractions and

$$2^0 + 2^1 + \dots + 2^{n-2} = \frac{1 - 2^{n-1}}{1 - 2} = (2^{n-1} - 1) \quad (7.36)$$

complex multiplications for the trivial twiddle factors $-j$. Compared to fast SCHT algorithm reported in Subsection 3.2.5, they have the same number of complex addition/subtractions but this presented fast CS-SCHT algorithm requires less trivial complex multiplications than fast SCHT algorithm, that is, $(2^{n-1} - 1)$ for the CS-SCHT and $(N/4) \log_2(N/2) = 2^{n-2}(n - 1)$ for the SCHT. The numerical numbers are presented in Table 7.8 for the comparison purpose. It can be seen from the table that when n is increased, the difference becomes larger and significant.

This fast CS-SCHT algorithm is also well suited to adopt the pipelined hardware structure, which was proposed for the SCHT in Chapter 4. The signal flow graph in Figure 7.3 is similar to that of the DIS fast SCHT algorithm (which was used to develop the pipelined hardware structure for the SCHT) except that the former needs less complex multiplications by the trivial twiddle factors. As explained in Chapter 4, the operations of swapping and subtraction can be jointly performed with the aid of

	Number of complex multiplications by the elements $-j$	
n	CS-SCHT	SCHT
3	3	4
4	7	12
5	15	32
6	31	80
7	63	192
8	127	448
9	255	1024
10	511	2304

Table 7.8: Comparison of computational complexity to compute the fast CS-SCHT and fast SCHT.

the 2 : 1 complex multiplexers to accomplish the multiplications with $-j$ (see Figure 4.6 for the modified adder/subtractor unit). Therefore, it should be noted that less operations are needed to implement the CS-SCHT as compared to the SCHT.

7.6 Applications of the CS-SCHT

In this section, we shall present some potential applications of the CS-SCHT together with supporting simulation results.

7.6.1 Spectrum Estimation

Like the SCHT, the CS-SCHT may also have certain significance in signal analysis and synthesis due to its sequency ordering. In this subsection, the CS-SCHT is applied in spectrum estimation of the sinusoidal waves. Figures 7.4 and 7.6 show the magnitude

spectra of 512-point DFT, CS-SCHT and SCHT of the sinusoidal waves of 1000 Hz and 2000 Hz using the sampling frequency (F_s) of 512000 Hz (in order to show the complete spectrum), respectively. Therefore, frequency resolution is $\Delta f = \frac{F_s}{N} = 1000$ Hz. The corresponding highlighted portions in these figures, which are zoomed in to provide the clear views, are shown in Figures 7.5 and 7.7. It can be seen from Figures 7.4 and 7.6 that the DFT can estimate frequency perfectly since there is only one peak occurred at frequency index $k = 1$ (for 1000 Hz) and $k = 2$ (for 2000 Hz) in the corresponding figures. It is also observed that both CS-SCHT and SCHT spectra are closely matched with that of the DFT. The desired main peaks occur in the same locations for all spectra as shown in Figures 7.5 and 7.7. But it is observed that the SCHT spectra has more unwanted side lobes than that of the CS-SCHT as illustrated in the figures. Therefore, it can be concluded that sequency of the CS-SCHT is more closely related to frequency of the DFT than that of the SCHT. Another obvious advantage is that the spectra obtained by the CS-SCHT are conjugate-symmetric (as that of the DFT) as shown in Figures 7.4 and 7.6, hence, only half of the spectra coefficients are required for analysis as compared with the SCHT spectrum.

Now we consider more general sinusoids with frequencies $f = 1375$ Hz and 1650 Hz so that the estimating frequencies are not an integer multiple of Δf . The magnitude spectra of the DFT, the CS-SCHT and the SCHT are shown in Figures 7.8 and 7.10. The corresponding highlighted portions are illustrated in Figures 7.9 and 7.11. The spectra shown in Figures 7.8 and 7.10 look similar to the spectra displayed in Figures 7.4 and 7.6. But let us take a look at Figures 7.9 and 7.11 for detailed analysis. Clearly, the corresponding frequencies cannot be estimated correctly since $\Delta f = 1000$ Hz. For $f = 1375$, the main lobes of all spectra are located at $k = 1$ and for $f = 1650$, they occur at $k = 2$. The widths of the DFT main lobe increase with reduced amplitudes as shown in the figures. It is expected that some energy is distributed over other frequency indices. Likewise, the widths of the CS-SCHT

and SCHAT main lobes become broader with decreased amplitudes as compared to the corresponding spectra shown in Figures 7.5 and 7.7. It is also found that their side lobes are expanded in width as shown in the figures. But no significant additional side lobes are introduced in both CS-SCHAT and SCHAT spectra.

In a nutshell, the spectrum of the CS-SCHAT is closely matched with the DFT spectrum, and also the CS-SCHAT coefficients are conjugate symmetric like the DFT coefficients. Hence, the CS-SCHAT can be considered as a good substitute to replace the DFT in spectrum estimation while providing simple implementation and less computational time.

Figure 7.4: Magnitude spectra of the DFT, CS-SCHT and SCHT for the sinusoidal wave of 1000 Hz.

Figure 7.5: The highlighted portion in Figure 7.4.

Figure 7.6: Magnitude spectra of the DFT, CS-SCHT and SCHT for the sinusoidal wave of 2000 Hz.

Figure 7.7: The highlighted portion in Figure 7.6.

Figure 7.8: Magnitude spectra of the DFT, CS-SCHT and SCHT for the sinusoidal wave of 1375 Hz.

Figure 7.9: The highlighted portion in Figure 7.8.

Figure 7.10: Magnitude spectra of the DFT, CS-SCHT and SCHT for the sinusoidal wave of 1650 Hz.

Figure 7.11: The highlighted portion in Figure 7.10.

7.6.2 Image Compression

Transform coding technique is one of the most popular methods to compress the digital images. In *transform coding*, a linear and reversible transform is used to map the image into a set of transform coefficients which are then quantized and coded. The fundamental goal of image compression is to reduce the bit rate which implies less transform coefficients to reconstruct the image while maintaining an acceptable fidelity or image quality. Therefore, the transforms having good energy compaction property are advisable to be employed in image compression. It is a well-known fact that most orthogonal transforms tend to pack a large fraction of the average energy of the images into a relatively few components of the transform coefficients (energy compaction property). Some of the well-known orthogonal transforms used in image compression are the discrete cosine transform (DCT) [1, 7, 68, 69], the Karhunen-Loeve transform (KLT) [6], the discrete Wavelet transform (DWT) [69, 70] and the Walsh Hadamard transform (WHT) [71]. The KLT is an optimal transform in an information packing sense [58]. But the KLT is data dependent and obtaining the KLT coefficients for each image is a nontrivial costly computational task. Hence, they are not preferred in practical applications. The DCT is known to be the most popular transform for its highly information packing capability as well as the availability of various efficient fast computational algorithms. The current standard for compression of still images, JPEG, uses the DCT to compress the images. The WHT is the simplest non-sinusoidal orthogonal transform and also provides comparable performance in image compression as compared with the DCT. It is perhaps a better choice for real-time implementation in some applications.

In this subsection, the CS-SCHT is proposed to be applied in image compression. Since an image can be represented as a matrix consisting of the real numbers, real-valued transforms are preferred to be used in order to obtain the real transform

coefficients. For this purpose, the 8×8 transform matrix which is derived from the CS-SCHT (shown in (7.19)) could be used for image compression with very low computational cost, which merely consists of addition/subtractions. Since the CS-SCHT is a complex orthogonal transform, its real-valued transform is also an orthogonal transform. After normalizing the elements of each row vector of (7.19) by its 2-norm, the 8×8 transform matrix is obtained as

$$\mathbf{H}_{R,8} = \mathbf{D}_{R,8} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & -1 & -1 \\ 1 & 1 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 \\ 1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix} \quad (7.37)$$

where $\mathbf{D}_{R,8} = \text{diag} \left\{ \sqrt{1/8}, \sqrt{1/4}, \sqrt{1/4}, \sqrt{1/4}, \sqrt{1/4}, \sqrt{1/4}, \sqrt{1/4}, \sqrt{1/8} \right\}$ is the diagonal matrix which consists of the normalization factors. As mentioned in Section 7.3, the row vectors of this matrix are the approximations of the sinusoidal waves in the DFT. Therefore, the input signal can be represented as a linear combination of its row vectors expecting that the signal energy could be concentrated on a few transform coefficients for good compression ratio. The corresponding transform is defined as follows:

$$\mathbf{X}_8 = \mathbf{H}_{R,8} \mathbf{x}_8 \quad (7.38)$$

where $\mathbf{X}_8 = [X(0), X(1), \dots, X(7)]^T$ is the transformed column vector and $\mathbf{x}_8 = [x(0), x(1), \dots, x(7)]^T$ represents the input data vector. The data sequence can be

recovered uniquely from the inverse transform, that is,

$$\mathbf{x}_8 = \mathbf{H}_{R,8}^T \mathbf{X}_8 \quad (7.39)$$

where $\mathbf{H}_{R,8}^T \mathbf{H}_{R,8} = \mathbf{I}_8$ (orthogonal property). The two dimensional (2-D) transform for an 8×8 real image matrix can be obtained by applying 1-D transform on the rows followed by the columns as defined in (6.3). The resultant transform coefficients are the real numbers, which are suitable for image compression.

The transformation by $\mathbf{H}_{R,8}$ can be done efficiently with minimum computational cost by using the fast algorithm. The fast algorithms to compute both forward and backward transforms are derived for the purpose of implementation for image compression. The readers may refer to Appendix A for details. The signal flow graph for fast forward 8-point transform is illustrated in Figure 7.12. For simplicity, the normalization factors are omitted in the figure, which in fact can be computed after or before the transformation. It can be seen from the figure that only the first stage needs eight addition/subtractions whereas the second and third stages merely requires

Figure 7.12: Signal flow graph of the fast forward algorithm for the real transform derived from the CS-SCHT ($N = 8$).

four and six addition/subtractions, respectively. Therefore, a total of $(8 + 4 + 6) = 18$ addition/subtractions is needed to compute an 8-point 1-D transform. On the other hand, 24 addition/subtractions are required to compute an 8-point traditional WHT using the fast algorithm as described in Subsection 2.3.2 (see Figure 2.2). The same number of multiplications (which is eight) is required for both transforms when the signal is scaled by the normalization factors which are shown in the diagonal matrix $\mathbf{D}_{R,8}$ in (7.37). Hence, an additional savings of $(24 - 18) = 6$ addition/subtractions which is $\frac{6}{24} \times 100\% = 25\%$ can be gained for one transformation by using the proposed transform.

The computer simulations are also conducted to evaluate the mean square error (MSE) performance of the proposed transform in image compression. The image block size considered in the simulation is the standard size of 8×8 and the test images used are the Lena and the Peppers of 256×256 pixels. The original images are divided into non-overlapping 8×8 blocks and the corresponding transforms are performed on these blocks. During the reconstruction process, the coefficients with maximum magnitudes on each transformed 8×8 block are selected based on the compression ratio. Figures 7.13 and 7.15 show the original images of Lena and Peppers and the corresponding reconstructed images with 16 coefficients (i.e., compression ratio = $\frac{16}{64} \times 100\% = 25\%$) using the DCT, the proposed transform and the WHT, respectively. Their corresponding PSNRs of the reconstructed images with respect to the original images are 34.63 dB, 31.17 dB and 31.85 dB for the Lena image, and 36.62 dB, 32.17 dB and 32.81 dB for the Peppers image. Obviously, the DCT outperforms the other transforms in terms of PSNR and image quality at the expense of highly computational cost. On the other hand, it is found that the PSNRs of the reconstructed images for the proposed transform and the WHT are almost similar and their image qualities are also good enough to compare to that of the DCT. No significant visual degradation occurs in both images as shown in Figures 7.13 and 7.15.

Figures 7.14 and 7.16 show the MSE comparisons of various transforms with respect to the number of coefficients chosen for reconstructing the images using the Lena and Peppers images, respectively. It can be seen from the graphs that the MSE performances of the proposed transform and the WHT are close to each other for most of the compression ratios (i.e., the number of coefficients selected). The image qualities for both transforms are also equally good enough and the difference is hardly perceived by human vision as shown in Figures 7.13 and 7.15. But the advantage of the proposed transform over the WHT is the reduced computational cost. As mentioned earlier, an extra savings of 6 addition/subtractions (25%) can be acquired for one transformation by using the proposed transform as compared to the WHT. If an image of 512×512 pixels is considered for image compression, the total number of blocks will be $(512 \times 512) \div (8 \times 8) = 4096$ and each block needs 16 transformations using (7.38). As one transformation performs 24 addition/subtractions, a total of $24 \times 16 \times 4096 = 1572864$ addition/subtractions is needed for the whole computation if the WHT is used. However, only a total of $1572864 \times 0.75 = 1179648$ addition/subtractions is required for the proposed transform, hence, we can obtain an additional saving of $1572864 \times 0.25 = 393216$ addition/subtractions, which is significant in the hardware implementation especially for real-time applications. Therefore, if the hardware savings and simplicity are the first priority to be considered rather than the image quality in the applications, the proposed transform and the WHT are shown to be better choices than the DCT, but among them, the proposed transform is better option to be considered.

Figure 7.13: (a) The original image of Lena, the decompressed images of Lena using (b) the DCT with PSNR = 34.63 dB, (c) the proposed transform derived from the CS-SCHT with PSNR = 31.17 dB, and (d) the WHT with PSNR = 31.85 dB using 16 coefficients.

Figure 7.14: MSE comparison among the DCT, the proposed transform derived from the CS-SCHT and the WHT with respect to the number of coefficients selected during reconstruction for the Lena image.

Figure 7.15: (a) The original image of Peppers, the decompressed images of Peppers using (b) the DCT with PSNR = 36.62 dB, (c) the proposed transform derived from the CS-SCHT with PSNR = 32.17 dB, and (d) the WHT with PSNR = 32.81 dB using 16 coefficients.

Figure 7.16: MSE comparison among the DCT, the proposed transform derived from the CS-SCHT and the WHT with respect to the number of coefficients selected during reconstruction for the Peppers image.

7.7 Conclusion

The conjugate symmetric sequency-ordered complex Hadamard transform (CS-SCHT) is discussed in this chapter. It has been shown that sequency defined in the CS-SCHT is more closely related to frequency in the DFT as compared to that of the SCHT. As the name follows, the CS-SCHT spectrum of a real-valued data sequence is conjugate-symmetric as well as the transform itself is sequency-ordered and suitable for complex-valued functions, which makes it unique among the transforms. Due to the conjugate-symmetric property, it is not surprising to note that only half of the spectrum are needed for analysis and synthesis, which leads to achieve memory savings in the processing of the transform in the applications (as compared to previously introduced SCHT). In addition, the CS-SCHT spectrum of a real-valued input sequence is dyadic shift invariant, hence, this transform may be useful to be employed in some applications where the WHT has been applied (as the WHT is also a dyadic shift invariant transform).

Section 7.2 mentions the construction of the CS-SCHT based on the CS-NCHT. It has been found that the CS-SCHT matrices can be obtained from the bit-reversed arrangement of the row vectors of the CS-NCHT matrices. Section 7.3 explain sequency defined in the CS-SCHT in parallel with frequency of the sinusoidal waves in the DFT. It has been observed that the CS-SCHT is a close approximation to the sinusoidal waves in Fourier analysis using the staircase waveforms.

Section 7.4 derives and lists some useful properties of the CS-SCHT. The exponential property of the CS-SCHT is mentioned in detail using an example of $N = 8$. The definition of dyadic shift invariant (DSI) complex transformation is given, and it has been proved that the CS-SCHT is a DSI transform and its power spectrum is dyadic shift invariant.

Section 7.5 derives the fast CS-SCHT algorithm using the sparse matrix factoriza-

tion approach. It has been investigated that less number of complex multiplications by the trivial twiddle factors, i.e., $-j$ are required to compute an N -point CS-SCHT as compared to that of the SCHT although they have the same number of operations for the addition/subtractions. Besides, it has been shown that the fast CS-SCHT algorithm is well fitted to be implemented using the pipelined hardware architecture presented in Chapter 4.

Section 7.6 presents the applications of the CS-SCHT in spectrum estimation and image compression. The CS-SCHT is shown to be a better choice to replace the DFT (at the cost of some loss in accuracy of estimation) in signal analysis and synthesis as compared to the SCHT with simpler implementation and less complexity. This is because of the fact that the CS-SCHT spectrum has less spurious spikes than that of the SCHT when estimating the sinusoidal waves. Moreover, the new transform which is derived from the CS-SCHT is proposed in image compression. The fast algorithm to compute the 8-point proposed transform is also derived especially for use in image compression. It has been found that the MSE performance of the proposed transform is comparable to that of the WHT at various levels of compression ratios. But it has been noted that the proposed transform requires less number of addition/subtractions with the use of the presented fast algorithm when comparing to that of the WHT. Therefore, the proposed transform have been shown to be a better alternative to substitute the DCT in image compression if simplicity and computational cost are the important factors to be considered in the applications.

Chapter 8

Conclusions and Recommendations

8.1 Conclusions

This thesis discusses the sequency-ordered complex Hadamard transforms (SCHTs) and their applications to communications and signal processing. Firstly, the reviews on the existing Hadamard and complex Hadamard transforms are conducted and different orderings of the real Hadamard transforms are investigated. Having known the lacking of sequency ordering in the current forms of complex Hadamard transforms and also the fact that sequency order is favored for communications and signal processing works, two versions of the complex Hadamard transforms are introduced in this thesis, one is the SCHAT and the other is the conjugate symmetric SCHAT (CS-SCHAT). As their names imply, both transforms show the sequency ordering in which the row vectors of the transform matrices are arranged in an ascending order of zero crossings in the unit circle over a normalized time base $0 \leq t \leq 1$. But the latter has a unique property, that is, conjugate symmetry property where the row vectors of the transform matrices are conjugate symmetric, which results in the conjugate symmetric complex spectrum as that of the DFT. This property is advantageous as only half of the spectral coefficients are needed for the analysis and synthesis of the intended signal. The construction of

the SCHAT matrices is based on the complex Rademacher functions which are the extended complex version of the conventional Rademacher functions whereas the CS-SCHAT matrices are generated based on the WHT matrix and the direct block matrix operator. Both transforms are suitable to process the problems with complex-valued functions.

In the thesis, the SCHAT is first introduced together with its applications in signal analysis and synthesis, DS-CDMA systems and digital image watermarking. In fact, the SCHAT is a discrete orthogonal transform whose elements are confined to four complex elements $\{\pm 1, \pm j\}$. It is constructed based on the complex Rademacher functions. Some important properties of the SCHAT are derived and presented. The SCHAT matrix can also be expressed using the exponential formula which is actually useful to prove some properties such as the unitary property of the SCHAT. The transform involves four complex elements $\{\pm 1, \pm j\}$, only the additions or subtractions are required to perform the transformation. The factorization of the transform matrix into a number of sparse matrices leads to the fast algorithm to compute an N -point SCHAT. Like other orthogonal transforms, the fast algorithm reduces the computational complexity to $O(N \log_2 N)$. As sequency is analogous to frequency in the DFT matrix, the relationship between the SCHAT and DFT matrices is investigated. It has been shown that they are related by a relationship matrix which can be generalized and factored as a number of sparse matrices which consist of the identity matrices and some matrices with the twiddle factors, which leads to the fast algorithm. As such the DFT can be obtained from the SCHAT. Even though this DFT computation via the SCHAT does not outperform the traditional radix-2 FFT algorithm in terms of computational load, this method is worth to consider in some applications where the sequency and frequency spectra are both desired.

The applications of the SCHAT are also discussed in this thesis. Firstly, the SCHAT is applied in signal analysis and synthesis to estimate a sinusoidal wave as its transform

matrix shows the sequency ordering which is analogous to frequency in the DFT. It has been found that the SCHAT spectrum is closely matched to that of the DFT with a few unwanted frequency components. Moreover, the SCHAT spectrum does not vary much due to the circular phase shift of the signal in the time domain.

Secondly, a new set of orthogonal complex spreading sequences called the SCHAT sequences which are derived from the row vectors of an SCHAT matrix is proposed in DS-CDMA systems. It is motivated by the fact that the orthogonal spreading sequences are greatly helpful to reduce the MAI due to zero crosscorrelation values for zero delay, and more importantly, the 4-phase complex sequences can yield a 3-dB performance enhancement in the maximum periodic correlation parameters as compared to the real binary sequences [28]. It has been found that the new SCHAT sequences exhibit very good crosscorrelation properties and they are suited to be employed in DS-CDMA systems to suppress the MAI effectively. The probability of error for DS-CDMA systems are first derived mathematically for both synchronous and asynchronous cases using the orthogonal complex spreading sequences. Then, the computer simulation has been conducted to evaluate the bit-error-rate performance of the introduced sequence in both synchronous and asynchronous DS-CDMA systems. It has been found that the SCHAT sequence offers the smallest mean-squared cross-correlation values and demonstrate better BER performance than the other existing sequences in an asynchronous DS-CDMA system over the multipath channel. The results also convey that the proposed sequence is more suitable in multiple access environment to combat the MAI but not in dense multipath environment due to its relatively poor autocorrelation properties.

Thirdly, the SCHAT is applied in image watermarking to develop a robust phase watermarking scheme. The simpler transform implementation and sequency ordering are the motivating factors to be considered in image watermarking. Since, sequency is analogous to frequency in the DFT, the image can be analyzed using the SCHAT

coefficients in the sequency domain. As such the suitable positions are located for the watermark embedding procedure. In the introduced scheme, the low amplitude block selection and the amplitude boost methods are applied to enhance the transparency of the watermarked image and to improve the robustness, respectively. Besides, the spread spectrum technique is incorporated into the watermarking scheme to ensure the security of the watermark against various malicious attacks as well as to enhance the robustness of the scheme. The experimental results illustrate that the SCHAT watermarked image provides a relatively high PSNR value as compared to that of the DFT and UCHT watermarked images. Besides, the SCHAT watermarked copy is able to sustain various kinds of image attacks such as JPEG lossy compression, scaling, rotating, cropping, painting, low-pass filtering, sharpening, noising and phase perturbation attack. Therefore, the SCHAT has been shown to be a suitable candidate and a useful tool for such applications.

As the transform is useful to many applications, dedicated hardware processor is essential to implement it for the real-time applications. For this reason, the pipelined hardware structure has been presented in this thesis to particularly deal with sequentially presented input/output data streams. At first, the decimation-in-sequency fast SCHAT algorithm is derived from the decimation-in-time fast SCHAT algorithm to efficiently map the fast algorithm into the hardware in order to accommodate the serial input for computation. It has been observed that the proposed pipelined structure is able to achieve 100% hardware utilization as well as minimum hardware requirements.

Another contribution in this thesis is the introduction of natural-ordered complex Hadamard transform (called the NCHT). The transform matrices are constructed through the WHT and the direct block matrix operation. The NCHT is related to the SCHAT through a bit reversal conversion and it should be noted that the NCHT is another ordered version of complex Hadamard transform, whose order corresponds to that of the WHT. It has been shown that the NCHT and the SCHAT are the complex

counterparts of the WHT and the SOWT. Several of its properties are derived and presented. The fast computation of the transform is also derived utilizing the matrix factorization approach and it has been found that its signal flow graph is almost similar to that of the DIS fast SCHAT algorithm. Therefore, the fast NCHAT decomposition is also suited to adopt the pipelined hardware architecture presented for the SCHAT. Since the NCHAT is a bit-reversed ordered version of the SCHAT (i.e., their row vectors are the same except that the way they are positioned in the transform matrix is different), it has the same potential to be employed in different applications as the SCHAT.

Finally, the CS-SCHAT is proposed in this thesis, whose sequency is closely related to frequency in the DFT. The CS-SCHAT matrix is conjugate symmetric as the DFT matrix except that the former is a complex Hadamard transform which consists of the elements $\{\pm 1, \pm j\}$. The applications where the SCHAT has been used may also employ the CS-SCHAT as well. The obvious advantage over the SCHAT is the reduced memory requirement for processing of the transform coefficients in the applications due to its conjugate symmetry property. With such property, only half of the spectral coefficients are required for processing. In this thesis, the CS-SCHAT has been employed in spectrum estimation and its magnitude spectrum is compared with that of the SCHAT. It has been found that the CS-SCHAT is able to estimate the sinusoidal waves better than the SCHAT. Even though both spectra are closely matched with the DFT spectrum, the CS-SCHAT spectrum has less unwanted side lobes than that of the SCHAT. The application of the CS-SCHAT in image compression has also been discussed in this thesis. The real transform is first derived from the CS-SCHAT based on the conjugate symmetry property. The resultant transform is applied to compress the image. It has been investigated that the proposed transform is comparable to the WHT in terms of the MSE performance in various compression levels. The gain of using the proposed transform is less computational cost due to the zero elements in the transform matrix with the use of presented fast algorithm.

8.2 Recommendations for Future Research

Below is the list of the possible extensions of the work presented in this thesis, which could be carried out in the future.

- The SCHAT is employed in DS-CDMA systems as the complex spreading sequences in this thesis. The applications of the SCHAT sequences can be further investigated in DS-CDMA systems. Since the SCHAT sequences show very good crosscorrelation properties, they have the potential to be employed in multi-carrier DS-CDMA system to channelize the different users. Besides, they are quadri-phase complex spreading sequences so it is compelling to investigate their bit-error-rate performance in quaternary DS-SSMA systems.
- The SCHAT-based watermarking algorithm may be used for adaptive equalization method based on the use of a watermark as a referenced signal.
- The applications of both SCHAT and CS-SCHAT are to be further explored in the areas of communications as well as signal processing. Since both transforms show the sequency ordering which is analogous to frequency in the DFT, the research should be further looked into the areas of applications where the DFT has been applied. Besides, due to the dyadic shift invariant property of the CS-SCHAT power spectrum, it can also be employed in some applications where the WHT has been applied as the WHT is also a dyadic shift invariant transform.
- Particularly, the application of the CS-SCHAT can be further extended in spectrum estimation such as analyzing the signal parameters in satellite radar altimetry. The CS-SCHAT could possibly be employed as a tool to develop audio watermarking algorithm.
- It is also interesting to find out the applications which are particularly suited to the natural-ordered complex Hadamard transform.

- To ensure the workability of the proposed pipelined hardware architecture, a proof-of-concept prototype can be constructed.

Appendix A: Fast Algorithms for a Real Transform Derived from the CS-SCHT

I. Fast Forward Algorithm

From (7.38), the forward transform is given by

$$\mathbf{X}_8 = \mathbf{H}_{R,8}\mathbf{x}_8 \quad (\text{A.1})$$

where $\mathbf{H}_{R,8}$ is the transform matrix as defined in (7.37), $\mathbf{x}_8$ and $\mathbf{X}_8$ are the input data and transformed column vectors, respectively. Firstly, the row vectors of $\mathbf{H}_{R,8}$ are rearranged through a conversion of the bit reversal and converted Gray code [7]. The example for such conversion for $N = 8$ is tabulated in Table A.1. Therefore, (A.1) becomes

$$\mathbf{P}\mathbf{X}_8 = \mathbf{P}\mathbf{H}_{R,8}\mathbf{x}_8 \quad (\text{A.2})$$

Decimal	Binary	Bit Reversal	Converted Gray Code	Decimal
0	000	000	000	0
1	001	100	111	7
2	010	010	011	3
3	011	110	100	4
4	100	001	001	1
5	101	101	110	6
6	110	011	010	2
7	111	111	101	5

Table A.1: Conversion through a bit reversal and converted gray code ($N = 8$).

where $\mathbf{P}$ is a permutation matrix,

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (\text{A.3})$$

which provides the arrangement of the row vectors as mentioned in Table A.1, and

$$\mathbf{H}_P = \mathbf{P}\mathbf{H}_{R,8} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & -1 & -1 \\ 0 & 0 & -1 & 1 & 0 & 0 & 1 & -1 \\ 1 & 1 & 0 & 0 & -1 & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 \end{bmatrix}. \quad (\text{A.4})$$

Now let us focus on the factorization of the above matrix to derive the fast algorithm.

For simplicity, the diagonal matrix $\mathbf{D}_{R,8}$ in $\mathbf{H}_{R,8}$ is ignored in the derivations. $\mathbf{H}_P$ can be rewritten and factorized as

$$\mathbf{H}_P = \begin{bmatrix} \mathbf{H}_4 & \mathbf{H}_4 \\ \mathbf{H}'_4 & -\mathbf{H}'_4 \end{bmatrix} = \begin{bmatrix} \mathbf{H}_4 & \mathbf{0} \\ \mathbf{0} & \mathbf{H}'_4 \end{bmatrix} \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{I}_4 & -\mathbf{I}_4 \end{bmatrix} \quad (\text{A.5})$$

where

$$\mathbf{H}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{H}'_4 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & -1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}.$$

It can be further factorized as

$$\mathbf{H}_P = \begin{bmatrix} \mathbf{H}_2 & \mathbf{H}_2 \\ \hat{\mathbf{I}}_2 & -\hat{\mathbf{I}}_2 \\ & \mathbf{0} & -\mathbf{H}_2 \\ & \mathbf{H}_2 & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{I}_4 & -\mathbf{I}_4 \end{bmatrix} \quad (\text{A.6})$$

where

$$\mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad \mathbf{H}'_2 = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \quad \text{and} \quad \hat{\mathbf{I}}_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Finally, we can have

$$\mathbf{H}_P = \begin{bmatrix} \mathbf{H}_2 & & & \\ & \hat{\mathbf{I}}_2 & & \\ & & \mathbf{H}'_2 & \\ & & & \mathbf{H}_2 \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & \mathbf{I}_2 \\ \mathbf{I}_2 & -\mathbf{I}_2 \\ & \mathbf{0} & \mathbf{I}_2 \\ \mathbf{I}_2 & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{I}_4 & -\mathbf{I}_4 \end{bmatrix} \quad (\text{A.7})$$

which results in fast forward algorithm after substituting it into (A.2). The output sequence is needed to be rearranged according to the permutation matrix $\mathbf{P}$. The signal flow graph for the fast forward algorithm is illustrated in Figure 7.12.

II. Fast Inverse Algorithm

Since the inverse transformation matrix is the transpose of $\mathbf{H}_{R,8}$, i.e., $\mathbf{H}_{R,8}^T$, the inverse transform has a different approach for factorization as well as different signal flow graph. (A.2) can be rewritten to find the backward transform as

$$\begin{aligned} \mathbf{P}\mathbf{X}_8 &= \mathbf{H}_P\mathbf{x}_8 \\ \mathbf{H}_P^{-1}\mathbf{P}\mathbf{X}_8 &= \mathbf{H}_P^{-1}\mathbf{H}_P\mathbf{x}_8 \\ \mathbf{x}_8 &= \mathbf{H}_P^T\mathbf{P}\mathbf{X}_8 \end{aligned} \quad (\text{A.8})$$

since $\mathbf{H}_P^{-1} = \mathbf{H}_P^T$ (orthogonal property). Therefore, fast inverse transform algorithm can be obtained by the factorization of $\mathbf{H}_P^T$, which is derived as follows. From (A.5),

$$\mathbf{H}_P^T = \begin{bmatrix} \mathbf{H}_4^T & \{\mathbf{H}'_4\}^T \\ \mathbf{H}_4^T & -\{\mathbf{H}'_4\}^T \end{bmatrix} \quad (\text{A.9})$$

which can be expressed as

$$\mathbf{H}_P^T = \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{I}_4 & -\mathbf{I}_4 \end{bmatrix} \begin{bmatrix} \mathbf{H}_4^T & \mathbf{0} \\ \mathbf{0} & \{\mathbf{H}'_4\}^T \end{bmatrix} \quad (\text{A.10})$$

where $\mathbf{H}_4$ and $\mathbf{H}'_4$ are as defined in (A.5),

$$\mathbf{H}_4^T = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & -1 & 1 & 0 \\ 1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 0 \end{bmatrix} \quad \text{and} \quad \{\mathbf{H}'_4\}^T = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \\ 1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}.$$

The expression in (A.10) can be further factorized and expressed as follows:

$$\begin{aligned} \mathbf{H}_P^T &= \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{I}_4 & -\mathbf{I}_4 \end{bmatrix} \begin{bmatrix} \mathbf{H}_2 & \hat{\mathbf{I}}_2 \\ \mathbf{H}_2 & -\hat{\mathbf{I}}_2 \\ & \mathbf{0} & \mathbf{H}_2 \\ & \{\mathbf{H}'_2\}^T & \mathbf{0} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{I}_4 & -\mathbf{I}_4 \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & \mathbf{I}_2 \\ \mathbf{I}_2 & -\mathbf{I}_2 \\ & \mathbf{0} & \mathbf{I}_2 \\ & \mathbf{I}_2 & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{H}_2 & & & \\ & \hat{\mathbf{I}}_2 & & \\ & & \{\mathbf{H}'_2\}^T & \\ & & & \mathbf{H}_2 \end{bmatrix} \quad (\text{A.11}) \end{aligned}$$

where $\mathbf{H}_2$ and $\hat{\mathbf{I}}_2$ are as defined in (A.6), and $\{\mathbf{H}'_2\}^T = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$. Substituting the above factorization result into (A.8) will lead to the fast inverse algorithm, and its signal flow graph is illustrated in Figure A.1. As shown in the figure, the fast inverse algorithm will have the same computational load as that of the fast forward algorithm.

Figure A.1: Signal flow graph of the fast inverse algorithm for the real transform derived from the CS-SCHT ($N = 8$).

Author's Publications

- [1] A. Aung, B. P. Ng, and S. Rahardja, "Performance of SCHAT sequences in asynchronous CDMA system," *IEEE Communications Letters*, vol. 11, no. 8, pp. 641-643, 2007.
- [2] G. Bi, A. Aung and B. P. Ng, "Pipelined hardware structure for sequency-ordered complex Hadamard transform," *IEEE Signal Processing Letters*, vol. 15, pp. 401-404, 2008.
- [3] A. Aung, B. P. Ng, and S. Rahardja, "Sequency-ordered complex Hadamard transform: Properties, computational complexity and applications," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3562-3571, 2008.
- [4] A. Aung, B. P. Ng, and S. Rahardja, "Conjugate symmetric sequency-ordered complex Hadamard transform," *IEEE Transactions on Signal Processing*, to be published, 2009.
- [5] A. Aung, B. P. Ng, and S. Rahardja, "A robust watermarking scheme using sequency-ordered complex Hadamard transform," *Signal Processing*, submitted for publication.
- [6] A. Aung and B. P. Ng, "Natural-ordered complex Hadamard transform," *Signal Processing*, submitted for publication.
- [7] A. Aung, B. P. Ng, and S. Rahardja, "A robust phase watermarking algorithm using conjugate symmetric sequency-ordered complex Hadamard transform," *2009 IEEE International Conference on Image Processing*, submitted for publication.
- [8] A. Aung, B. P. Ng, and C. T. Shwe, "A new transform for document image compression," *2009 IEEE International Conference on Information, Communications and Signal Processing*, submitted for publication.

Bibliography

- [1] J. G. Proakis and D. G. Manolakis, *Digital signal processing: Principles, algorithms, and applications*, 4th ed. Pearson Prentice Hall, 2007.
- [2] E. Pogossova, K. Egiazarian, and J. Astola, "Signal denoising in tree-structured Haar basis," in *Proc. of the 3rd International Symposium on Image and Signal Processing and Analysis*, vol. 2, Rome, Italy, 2003, pp. 736–739.
- [3] A. Dziech, P. Slusarczyk, and B. Tibken, "Methods of image compression by PHL transform," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 39, no. 4, pp. 447–458, 2004.
- [4] J. A. R. Macias and A. G. Exposito, "Efficient computation of the running discrete Haar transform," *IEEE Transactions on Power Delivery*, vol. 21, no. 1, pp. 504–505, 2006.
- [5] M. Effros, F. Hanying, and K. Zeger, "Suboptimality of the Karhunen-Loeve transform for transform coding," *IEEE Transactions on Information Theory*, vol. 50, no. 8, pp. 1605–1619, 2004.
- [6] B. Penna, T. Tillo, E. Magli, and G. Olmo, "Transform coding techniques for lossy hyperspectral data compression," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 5, pp. 1408–1421, 2007.
- [7] K. G. Beauchamp, *Applications of Walsh and related functions, with an introduction to sequency theory*. Academic Press, 1984.
- [8] N. Ahmed, K. R. Rao, and Abdussattar, "BIFORE or Hadamard transform," *IEEE Transactions On Audio and Electroacoustics*, vol. AU-19, no. 3, pp. 225–234, Sept. 1971.
- [9] K. R. Rao and N. Ahmed, "Complex BIFORE transform," *Int. J. Syst. Sci.*, vol. 2, no. 2, pp. 149–162, 1971.
- [10] S. Rahardja and B. J. Falkowski, "Family of unified complex Hadamard transforms," *IEEE Transactions on Circuits and Systems II-Analog and Digital Signal Processing*, vol. 46, no. 8, pp. 1094–1100, 1999.

- [11] B. J. Falkowski and S. Rahardja, "Complex Hadamard transforms: Properties, relations and architecture," *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, vol. E87A, no. 8, pp. 2077–2083, 2004.
- [12] ———, "Complex decision diagrams," *Electronics Letters*, vol. 32, no. 4, pp. 290–291, 1996.
- [13] S. Rahardja and B. J. Falkowski, "Complex composite spectra of unified complex Hadamard transform for logic functions," *IEEE Transactions on Circuits and Systems II-Analog and Digital Signal Processing*, vol. 47, no. 11, pp. 1291–1297, 2000.
- [14] S. Rahardja, W. Ser, and Z. N. Lin, "UCHT-based complex sequences for asynchronous CDMA system," *IEEE Transactions on Communications*, vol. 51, no. 4, pp. 618–626, 2003.
- [15] Z. Gu, S. Xie, and S. Rahardja, "UCHT complex sequences for downlink multi-carrier DS-CDMA systems," in *IEEE 61st Vehicular Technology Conference*, vol. 61, Stockholm, 2005, pp. 392–395.
- [16] B. J. Falkowski and L. S. Lim, "Image watermarking using Hadamard transforms," *Electronics Letters*, vol. 36, no. 3, pp. 211–213, 2000.
- [17] R. Yarlagadda and J. Hershey, *Hadamard matrix analysis and synthesis with applications to communications and signal/image processing*. Norwell, MA: Kluwer, 1997.
- [18] G. Peceli and B. Feher, "Digital filters based on recursive Walsh-Hadamard transformation," *IEEE Trans. Circuit Syst.*, vol. 37, pp. 150–152, Jan. 1990.
- [19] F. Fonda and S. Pastore, "Innovative image watermarking technique for image authentication in surveillance applications," in *Proc. IEEE International Workshop on Imaging Systems and Techniques*, Niagara Falls, Ont., Canada, 2005, pp. 32–35.
- [20] M. Faundez-Zanuy, J. Roure, V. Espinosa-Duro, and J. A. Ortega, "An efficient face verification method in a transformed domain," *Pattern Recognition Letters*, vol. 28, no. 7, pp. 854–858, 2007.
- [21] S. H. Tsai, Y. P. Lin, and C. C. J. Kuo, "MAI-free MC-CDMA systems based on Hadamard-Walsh codes," *IEEE Transactions on Signal Processing*, vol. 54, no. 8, pp. 3166–3179, 2006.
- [22] B. Gaffney and A. D. Fagan, "Walsh Hadamard transform precoded MB-OFDM: An improved high data rate ultrawideband system," in *IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications*, Helsinki, Finland, Sept. 2006, pp. 1–5.

- [23] L. Zhihua and Z. Qishan, "Ordering of Walsh functions," *IEEE Trans. Electromagnetic Compatibility*, vol. EMC-25, no. 2, pp. 115–119, 1983.
- [24] C. K. Yuen, "Remarks on the orderings of Walsh functions," *IEEE Transactions on Computers*, vol. C-21, no. 12, p. 1452, 1972.
- [25] Z. Wang, "In-place and in-order algorithms for Paley- and sequency-ordered Walsh-Hadamard transforms," in *Proc. IEEE International Symposium on Electromagnetic Compatibility Symposium*, Washington, DC, USA, 1990, pp. 90–94.
- [26] R. D. Brown, "A recursive algorithm for sequency-ordered fast Walsh transforms," *IEEE Transactions on Computers*, vol. C-26, no. 8, pp. 819–822, 1977.
- [27] B. Arambepola and K. C. Partington, "Walsh-Hadamard transform for complex-valued signals," *Electronics Letters*, vol. 28, no. 3, pp. 259–261, 1992.
- [28] S. Boztas, R. Hammons, and P. Y. Kumar, "4-phase sequences with near-optimum correlation properties," *IEEE Transactions on Information Theory*, vol. 38, no. 3, pp. 1101–1113, 1992.
- [29] K. Yang, Y.-K. Kim, and P. V. Kumar, "Quasi-orthogonal sequences for code-division multiple-access systems," *IEEE Transactions on Information Theory*, vol. 46, no. 3, pp. 982–993, 2000.
- [30] N. Ahmed and K. R. Rao, "Additional properties of complex BIFORE transform," *IEEE Transactions on Audio and Electroacoustics*, vol. AU-19, no. 3, pp. 252–253, 1971.
- [31] N. Ahmed, K. R. Rao, and R. B. Schultz, "Fast complex BIFORE transform by matrix partitioning," *IEEE Transactions on Computers*, vol. C-20, no. 6, pp. 707–710, 1971.
- [32] S. Xie and S. Rahardja, "Performance evaluation for quaternary DS-SSMA communications with complex signature sequences over Rayleigh-fading channels," *IEEE Transactions on Wireless Communications*, vol. 4, no. 1, pp. 266–277, 2005.
- [33] S. Xie, S. Rahardja, and Z. Gu, "Relationship between UCHT and FFT," in *Proc. IEEE 33rd International Symposium on Multiple-Valued Logic*, May 2003, pp. 154–159.
- [34] B. J. Falkowski and S. Rahardja, "Properties and applications of unified complex Hadamard transforms," in *Proc. IEEE 27th International Symposium on Multiple-Valued Logic*, Antigonish, NS, Canada, 1997, pp. 131–136.
- [35] S. Matsufuji and N. Suehiro, "Factorization of bent function type complex Hadamard matrices," in *Proc. IEEE 4th International Symposium on Spread Spectrum Techniques and Applications*, vol. 3, Mainz, Germany, 1996, pp. 950–954.

- [36] S. M. Kuo and W. S. Gan, *Digital signal processors: Architectures, implementations, and applications*, international ed. Pearson Prentice Hall, 2005.
- [37] A. Aung, B. P. Ng, and S. Rahardja, "Performance of SCHT sequences in asynchronous CDMA system," *IEEE Communications Letters*, vol. 11, no. 8, pp. 641–643, 2007.
- [38] —, "A robust watermarking scheme using sequency-ordered complex hadamard transform," *Signal Processing*, submitted for publication.
- [39] S. P. Mohanty, C. R. Kumara, and S. Nayak, "FPGA based implementation of an invisible-robust image watermarking encoder," in *7th International Conference on Information Technology*, Hyderabad, India, 2004, pp. 344–353.
- [40] R. Shah, A. Agarwal, and S. Ganesan, "Frequency domain real time digital image watermarking," in *IEEE International Conference on Electro Information Technology*, Lincoln, NE, US, 2005, pp. 1627–1633.
- [41] M. Shirali-Shahreza, "A new method for real-time steganography," in *IEEE 8th International Conference on Signal Processing*, vol. 4, 2006, pp. 16–20.
- [42] G. Bi and E. V. Jones, "A pipelined FFT processor for word-sequential data," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 12, pp. 1982–1985, 1989.
- [43] Y. N. Chang and K. K. Parhi, "An efficient pipelined FFT architecture," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 50, no. 6, pp. 322–325, 2003.
- [44] H. Shousheng and M. Torkelson, "Design and implementation of a 1024-point pipeline FFT processor," in *Proc. of the IEEE on Custom Integrated Circuits Conference*, 1998, pp. 131–134.
- [45] I. Oppermann and B. S. Vucetic, "Complex spreading sequences with a wide range of correlation properties," *IEEE Transactions on Communications*, vol. 45, no. 3, pp. 365–375, 1997.
- [46] P. Cotae, "Multicell spreading sequence desing algorithm for overloaded S-CDMA," *IEEE Communications Letters*, vol. 9, no. 12, pp. 1028–1030, 2005.
- [47] Z. N. Lin, S. Rahardja, and W. Ser, "Performance of UCHTs sequences in synchronous CDMA system," in *2nd International Symposium on Communication Systems, Networking, and Digital Signal Processing*, Bournemouth Univ, UK, Jul. 2000, pp. 328–331.
- [48] T. S. Rappaport, *Wireless communications: Principles and practice*, 2nd ed. Prentice Hall PTR, 2002.

- [49] J. G. Proakis, M. Salehi, and G. Bauch, *Contemporary communication systems using MATLAB and Simulink*, 2nd ed. Pacific Grove, Calif. London: Brooks/Cole, 2004.
- [50] J. J. K. Ó Ruanaidh, W. J. Dowling, and F. M. Boland, "Watermarking digital images for copyright protection," in *IEE Proceedings-Vision, Image and Signal Processing*, vol. 143, no. 4, 1996, pp. 250–256.
- [51] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1673–1687, 1997.
- [52] B. J. Falkowski, "Phase watermarking algorithm using hybrid multi-polarity Hadamard transform," *Journal of Mathematical Imaging and Vision*, vol. 30, no. 1, pp. 13–21, 2008.
- [53] J. Guo, Z. Liu, and S. Liu, "Watermarking based on discrete fractional random transform," *Optics Communications*, vol. 272, no. 2, pp. 344–348, 2007.
- [54] C. T. Hsu and J. L. Wu, "Hidden digital watermarks in images," *IEEE Transactions on Image Processing*, vol. 8, no. 1, pp. 58–68, 1999.
- [55] W. Y. Chen and C. H. Chen, "A robust watermarking scheme using phase shift keying with the combination of amplitude boost and low amplitude block selection," *Pattern Recognition*, vol. 38, no. 4, pp. 587–598, 2005.
- [56] J. J. K. Ó Ruanaidh, W. J. Dowling, and F. M. Boland, "Phase watermarking of digital images," in *Proc. IEEE International Conference on Image Processing*, vol. 3, 1996, pp. 239–242.
- [57] X. Qi and J. Qi, "A robust content-based digital image watermarking scheme," *Signal Processing*, vol. 87, no. 6, pp. 1264–1280, 2007.
- [58] R. C. Gonzalez and R. E. Woods, *Digital image processing*, International ed. Pearson Prentice Hall, 2002.
- [59] M. J. Tsai, K. Y. Yu, and Y. Z. Chen, "Joint wavelet and spatial transformation for digital watermarking," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 1, pp. 241–245, 2000.
- [60] P. Bao and M. Xiaohu, "Image adaptive watermarking using wavelet domain singular value decomposition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 1, pp. 96–102, 2005.
- [61] B. J. Falkowski and L. S. Lim, "Image watermarking using the complex Hadamard transform," in *Proc. IEEE International Symposium on Circuits and Systems*, vol. 4, 2000, pp. 573–576.

- [62] G. Bi, A. Aung, and B. P. Ng, "Pipelined hardware structure for sequency-ordered complex Hadamard transform," *IEEE Signal Processing Letters*, vol. 15, pp. 401–404, 2008.
- [63] J. S. Pan and C. Lin, "Attacking phase shift keying based watermarking," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E90A, no. 1, pp. 305–306, 2007.
- [64] H. F. Harmuth, "A generalized concept of frequency and some applications," *IEEE Trans. Information Theory*, vol. IT-14, no. 3, pp. 375–382, 1968.
- [65] B. Bhagavan and R. Polge, "Sequencing the Hadamard transform," *IEEE Trans. Audio and Electroacoustics*, vol. 21, no. 5, pp. 472–473, 1973.
- [66] A. Aung, B. P. Ng, and S. Rahardja, "Sequency-ordered complex Hadamard transform: Properties, computational complexity and applications," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3562–3571, 2008.
- [67] K. J. R. Liu, "A simple and unified proof of dyadic shift invariance and the extension to cyclic shift invariance," *IEEE Transactions on Education*, vol. 36, no. 4, pp. 369–372, 1993.
- [68] N. N. Ponomarenko, K. O. Egiazarian, V. V. Lukin, and J. T. Astola, "High-quality DCT-based image compression using partition schemes," *IEEE Signal Processing Letters*, vol. 14, no. 2, pp. 105–108, 2007.
- [69] S. Singh, V. Kumar, and H. K. Verma, "DWT-DCT hybrid scheme for medical image compression," *Journal of Medical Engineering and Technology*, vol. 31, no. 2, pp. 109–122, 2007.
- [70] L. R. Iyer and A. E. Bell, "Improving image compression performance with balanced multiwavelets," vol. 1, Pacific Grove, CA, USA, 2001, pp. 773–777.
- [71] I. Valova and Y. Kosugi, "Hadamard-based image decomposition and compression," *IEEE Transactions on Information Technology in Biomedicine*, vol. 4, no. 4, pp. 306–319, 2000.