

การสร้างลายน้ำสำหรับสัญญาณเสียงดิจิทัลโดยใช้การแปลงเวฟเล็ตและการ
ค้นหาแบบตามูขิงปรับตัว

นายพนันท์ ศรียิ่งยงค์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2548
ISBN 974-533-553-3

**WAVELET-BASED DIGITAL AUDIO
WATERMARKING USING ADAPTIVE TABU SEARCH**

Noppanan Sriyingyong

A Thesis Submitted in Partial Fulfillment of the Requirements for the

Degree of Master of Engineering in Electrical Engineering

Suranaree University of Technology

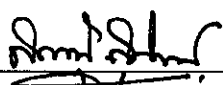
Academic Year 2005

ISBN 974-533-553-3

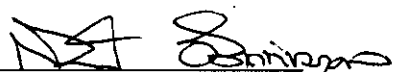
การทำลายน้ำสำหรับสัญญาอนุญาตสิทธิ์ดิจิทัลโดยใช้การแปลงเวฟเล็ตและการค้นหา
แบบตาข่ายเชิงปรับตัว

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นักวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษา
ตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์

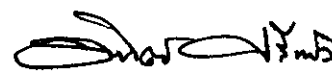

(รศ. ดร.สรารัตน์ สัจจิตร)

ประธานกรรมการ



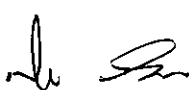
(ผศ. ดร.กิตติ อัครกิจมงคล)

กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)



(ผศ. ดร.อาทิตย์ ศรีแก้ว)

กรรมการ



(รศ. ดร.เสาวณี รัตนพานี)

รองอธิการบดีฝ่ายวิชาการ



(รศ. น.อ. ดร.วรพจน์ จำพิศ)

คณบดีสำนักวิชาวิศวกรรมศาสตร์

นพพันธ์ ศรียิ่งยงค์ : การสร้างลายน้ำสำหรับสัญญาณเสียงดิจิทัลโดยใช้การแปลงเวฟเล็ต และการค้นหาแบบตามูเชิงปรับตัว (WAVELET BASED DIGITAL AUDIO WATERMARKING USING ADAPTIVE TABU SEARCH) อาจารย์ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ ดร.กิตติ อรรถกิจมงคล, 134 หน้า
ISBN 974-533-553-3

ลายน้ำเป็นเทคโนโลยีที่นำศาสตร์ด้านต่างมารวมเข้าด้วยกัน เช่น การสื่อสาร สารสนเทศ การประมวลผลสัญญาณ เพื่อเป็นการป้องกันการละเมิดทรัพย์สินโดยไม่ได้รับอนุญาต ด้วยเหตุที่เสียงดิจิทัลสามารถถูกทำซ้ำและเผยแพร่ได้ง่ายผ่านทางอินเทอร์เน็ต การพิสูจน์ความเป็นเจ้าของจึงถูกนำมาใช้อย่างรีบด่วนเพื่อป้องกันเสียงดิจิทัลจากผู้ละเมิด

แม้ว่าการเข้ารหัสจะสามารถป้องกันการเผยแพร่สื่อดิจิทัลได้ในระดับหนึ่ง แต่เทคนิคดังกล่าวยังมีข้อเสียที่สำคัญประการหนึ่งคือ การป้องกันจะสูญเสียไปอย่างสมบูรณ์เมื่อสื่อถูกถอดรหัสแล้ว ดังนั้นการปกป้องลิขสิทธิ์ในสัญญาณเสียงจึงจำเป็นต้องมีวิธีที่มีประสิทธิภาพมากกว่าวิธีดังกล่าว

การสร้างลายน้ำดิจิทัล (digital watermarking) เป็นเทคนิคการฝังข้อมูลที่เรียกว่าลายน้ำลงบนสัญญาณเสียงต้นฉบับ โดยกระบวนการจะสร้างลายน้ำที่ไม่สามารถเอาออกได้ด้วยการประมวลผลสัญญาณ และเทคนิคการบีบอัดสัญญาณเสียงเช่น MP3 ยิ่งกว่านั้นลายน้ำต้องไม่สร้างความแตกต่างระหว่างสัญญาณเสียงต้นฉบับกับสัญญาณเสียงที่ฝังลายน้ำแล้ว ดังนั้น ถึงแม้ว่าสัญญาณเสียงที่ฝังลายน้ำแล้วจะถูกทำซ้ำหรือเผยแพร่ ลายน้ำก็ยังคงติดไปกับสัญญาณเสียง เพื่อเพิ่มความปลอดภัยข้อมูลลายน้ำจะถูกสลับลำดับโดยใช้กุญแจลับ หากปราศจากกุญแจลับนี้ถึงแม้ว่าผู้ละเมิดจะล่วงรู้อัลกอริทึมในการตรวจจับลายน้ำทั้งหมด ก็จะสามารถตรวจจับได้เพียงข้อมูลที่ไม่มีความหมายเท่านั้น

งานวิจัยวิทยานิพนธ์นี้เสนอการฝังลายน้ำแบบกึ่งปิด (semi-blind watermarking) ลงบนสัมประสิทธิ์ของเวฟเล็ตแบบดิสครีต อัลกอริทึมของการฝังลายน้ำจะค้นหาค่าความเข้มที่เหมาะสมของลายน้ำจากการแลก คุณภาพของเสียงของสัญญาณที่ฝังลายน้ำและความทนทานของลายน้ำโดยใช้การค้นหาแบบตามูเชิงปรับตัว (ATS) อัลกอริทึมใช้ SNR และการทำสหสัมพันธ์ ระหว่างลายน้ำต้นฉบับกับลายน้ำที่ตรวจจับได้เป็นตัวชี้วัดคุณภาพของเสียงและความทนทานของลายน้ำตามลำดับ

ผลการทดสอบชี้ให้เห็นว่าอัลกอริทึมที่เสนอสามารถสร้างลายน้ำที่มนุษย์ไม่สามารถได้ยินและทนทานต่อการโจมตีด้วยการประมวลผลสัญญาณแบบต่างๆและการบีบอัดแบบสูญเสีย

สาขาวิชา วิศวกรรมไฟฟ้า
ปีการศึกษา 2548

ลายมือชื่อนักศึกษา ๗๘ ๓๖
ลายมือชื่ออาจารย์ที่ปรึกษา [Signature]

NOPPANAN SRIYINGYONG : WAVELET-BASED DIGITAL AUDIO
WATERMARKING USING ADAPTIVE TABU SEARCH. THESIS
ADVISOR : ASST. PROF. KITTI ATTAKITMONGKOL, Ph.D. 134 PP.
ISBN 974-533-553-3

AUDIO WATERMARKING/OPTIMIZATION/ADAPTIVE TABU SEARCH/
DISCRETE WAVELET TRANSFORM

Watermarking is a method that combines many fields of science together such as telecommunication, information technology, signal processing to protect the ownership of digital media. Since digital audio can be easily duplicated and distributed over the Internet, the ownership verification is a critical issue to protect digital audio from piracy.

Even though coding technique is able to protect the distribution of digital media to some levels, there is a disadvantage using this technique because the protection is completely lost after the media has been decoded. Thus, a more efficient way to protect the ownership of audio signal is necessarily required.

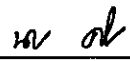
Audio watermarking is the technique that embeds some information called watermark into host audio signal. The process is performed in such a way that the watermark cannot be removed by common signal processing or audio compression technology such as MP3. Moreover, there should be no perceptual difference between the original audio signal and watermarked signal. Thus, whenever the watermarked signal is duplicated or distributed, the watermark is always hidden in the signal. To increase security, we use secret key to permute the watermark. Without this secret

key, even if the pirates know the watermark detection algorithm, they can detect only no meaningful data.

This thesis proposes a semi-blind audio watermarking algorithm in the discrete wavelet transform domain. In the watermark embedding algorithm, we search for the optimal intensity of watermark to trade off the quality of watermarked signal and robustness of watermark by using the adaptive tabu search (ATS). We use SNR and correlation between the original and watermark signals as measurements of audio quality and robustness of the watermark, respectively. The results show that this algorithm produces watermark which is inaudible and robust to many signal manipulations and lossy compression.

School of Electrical Engineering

Academic Year 2005

Student's Signature 

Advisor's Signature 

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงด้วยดี เนื่องจากพระคุณของ บิดา มารดา ขอกราบขอบพระคุณท่านทั้งสองที่มอบชีวิตอันประเสริฐ และความสำเร็จนี้คือผลงานอันน่าภาคภูมิใจของท่านทั้งสอง

ขอขอบพระคุณในความช่วยเหลืออย่างยิ่ง ทั้งด้านวิชาการและการดำเนินการวิจัย จากบุคคลและกลุ่มบุคคล อันได้แก่

ผู้ช่วยศาสตราจารย์ ดร.กิตติ อัครกิจมงคล หัวหน้าสาขาวิชาวิศวกรรมไฟฟ้า อาจารย์ที่ปรึกษาวิทยานิพนธ์ ผู้อยู่เบื้องหลังความสำเร็จทั้งหมดและคอยช่วยเหลือผู้วิจัยทั้งในด้านชีวิตและการทำงานเป็นอย่างดีจนสำเร็จการศึกษา

ผู้ช่วยศาสตราจารย์ ดร.อนันท์ อุณศิริ ไลย์ อาจารย์ประจำสาขาวิชาวิศวกรรมไฟฟ้า ที่ให้ความช่วยเหลือผู้วิจัยและคำปรึกษาที่มีคุณค่าผลักดันให้ผู้วิจัยสามารถดำรงชีวิตในมหาวิทยาลัยในช่วงเวลาหนึ่งได้เป็นปกติสุข

รองศาสตราจารย์ ดร.สรวิชัย สุจิตจร อาจารย์ประจำสาขาวิชาวิศวกรรมไฟฟ้า อาจารย์ผู้คอยชี้แนะทางที่ถูกต้องในการศึกษาให้ผู้วิจัยเสมอมา

ร.อ.ประโยชน์ คำสวัสดิ์ อาจารย์ประจำสาขาวิชาวิศวกรรมโทรคมนาคม ผู้ทุ่มเทช่วยเหลือให้กับผู้วิจัยอย่างเต็มที่โดยไม่เคยเอ่ยคำปฏิเสธ

ขอขอบคุณ คุณธวัชชัย กุลรวานิชพงษ์และผู้ช่วยศาสตราจารย์ ดร. เดชา พวงดาวเรือง ผู้เปรียบเสมือนพี่ชายเสมอ สำหรับน้องบัณฑิตศึกษา

ขอขอบคุณ คุณประพันธ์ คัทวิ วิศวกรศูนย์เครื่องมือวิทยาศาสตร์และเทคโนโลยี ตลอดจนวิศวกรประจำศูนย์เครื่องมือวิทยาศาสตร์และเทคโนโลยีทุกท่าน ที่ได้ช่วยเหลือและเป็นที่ปรึกษาที่ดีให้กับผู้วิจัย

ขอขอบคุณบัณฑิตศึกษาทุกท่าน พี่ น้องและเพื่อน ที่เคยอยู่ร่วมกันในละแวกบ้านที่อบอุ่น สำหรับการศึกษาที่ผ่านมามากมายปี

ท้ายนี้ขอขอบพระคุณ ครูบาอาจารย์ทุกท่านตั้งแต่อดีตที่ประสิทธิ์ประสาทวิชาความรู้แขนงต่างๆมาตลอด คุณสุนันทา บุญญกนก ผู้มอบสิ่งดีๆและเคียงข้างผู้วิจัยเสมอ ทุกท่านในครอบครัว ศรียังยงค์ และครอบครัว มีรักษา ที่มอบความรัก การอบรมเลี้ยงดู ส่งเสริมการศึกษาแก่ผู้วิจัยมาโดยตลอด

นพนันท์ ศรียังยงค์

สารบัญ

หน้า

บทคัดย่อ (ภาษาไทย)	ก
บทคัดย่อ (ภาษาอังกฤษ)	ค
กิตติกรรมประกาศ	ง
สารบัญ.....	จ
สารบัญตาราง.....	ช
สารบัญรูป.....	ญ
บทที่	

1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์การวิจัย	2
1.3 ขอบเขตของเบื้องต้น	2
1.4 ขอบเขตของการวิจัย.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	3
1.6 รูปเล่มวิทยานิพนธ์	4
2 ปรัชญาบรรณกรรมและงานวิจัยที่เกี่ยวข้อง.....	5
2.1 กล่าวนำ	5
2.2 ลายน้ำ.....	5
2.2.1 ประวัติความเป็นมา.....	5
2.2.2 การประยุกต์ใช้ลายน้ำ	7
2.2.3 คุณสมบัติที่สำคัญของลายน้ำ	8
2.2.4 การแบ่งประเภทของลายน้ำ.....	8
2.3 ลิขสิทธิ์ของไทย.....	9
2.4 รูปแบบสัญญาณเสียงดิจิทัล.....	10
2.4.1 ประวัติความเป็นมา	10

สารบัญ (ต่อ)

หน้า

2.4.1	รูปแบบไฟล์ WAV	11
2.4.1	รูปแบบไฟล์ MP3	12
2.5	ปริทรรศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง.....	13
3	ทฤษฎีเบื้องต้น	19
3.1	เวฟเล็ต	19
3.1.1	ประวัติความเป็นมาของเวฟเล็ต	19
3.1.2	ภาพรวมของเวฟเล็ต	19
3.1.3	ปริภูมิของฟังก์ชัน.....	23
3.1.4	ฟังก์ชันสเกลลิง.....	24
3.1.5	ฟังก์ชันสเกลลิงกับการวิเคราะห์สัญญาณแบบหลายระดับความละเอียด	25
3.1.6	ฟังก์ชันเวฟเล็ต.....	26
3.1.7	การแปลงเวฟเล็ตแบบเต็มหน่วย.....	28
3.1.8	ตระกูล Wavelet.....	30
3.1.9	การแปลงเวฟเล็ตแบบเร็ว (FWT)	34
3.1.10	อัลกอริทึมการแปลงเวฟเล็ตหนึ่งมิติ	34
3.2	การค้นหาแบบตามูเชิงปรับตัว	38
4	การฟังและการตรวจจบลายน้ำ.....	43
4.1	กล่าวนำ.....	43
4.2	การฟังลายน้ำ.....	43
4.2.1	การเตรียมสัญญาณเสียงและลายน้ำ	43
4.2.2	การฟังลายน้ำในโดเมนเวฟเล็ต	44
4.2.3	การค้นหาค่าความเข้มในฟังลายน้ำด้วย ATS.....	46
4.3	การตรวจจบลายน้ำ	49
5	ผลการทดสอบ	50
5.1	ผลการทดสอบความทนทานของลายน้ำ	51
5.1.1	ผลการทดสอบด้วยอัลกอริทึมที่เสนอในงานวิจัยวิทยานิพนธ์	51

สารบัญ (ต่อ)

หน้า

5.1.2 ผลการเปรียบเทียบความทนทานของลายน้ำระหว่างอัลกอริทึม ที่เสนอในงานวิจัยวิทยานิพนธ์ (ATS) กับอัลกอริทึมที่ใช้รหัส แก้ไขความผิดพลาด (ECC)	55
5.2 ผลการประเมินคุณภาพเสียง	57
5.3 วิเคราะห์ผลการทดสอบ	58
6 สรุปผลการวิจัยและข้อเสนอแนะ	61
6.1 สรุปผลการวิจัย	61
6.2 ข้อเสนอแนะ	62
รายการอ้างอิง	64
ภาคผนวก	
ภาคผนวก ก รายละเอียดของเสียงเพลงที่นำมาทดสอบ	66
ภาคผนวก ข รูปภาพลายน้ำ.....	70
ภาคผนวก ค โปรแกรม MATLAB ที่ใช้ในงานวิจัยวิทยานิพนธ์	73
ภาคผนวก ง โปรแกรม MATLAB ที่ใช้รหัสแก้ไขความผิดพลาด.....	99
ภาคผนวก จ ผลการทดสอบความทนทานของลายน้ำจากอัลกอริทึมที่เสนอ ในงานวิจัยวิทยานิพนธ์ และผลการทดสอบความทนทานของ ลายน้ำจากอัลกอริทึมที่ใช้รหัสแก้ไขความผิดพลาด.....	112
ภาคผนวก ฉ ค่าความเข้มในการฝังลายน้ำของเพลงต่างๆที่ได้จากการค้นหา แบบตามูเชิงปรับตัว	120
ภาคผนวก ช ผลการทดสอบคุณภาพเสียงหลังจากการฝังลายน้ำโดยการฝัง.....	123
ภาคผนวก ซ บทความที่ได้รับการตีพิมพ์เผยแพร่	132
ประวัติผู้เขียน	134

สารบัญตาราง

ตารางที่	หน้า
5.1 ผลการทดสอบกับเพลงสากล classic.....	51
5.2 ผลการทดสอบกับเพลงสากล jazz.....	52
5.3 ผลการทดสอบกับเพลงสากล pop	52
5.4 ผลการทดสอบกับเพลงไทยเดิม.....	53
5.5 ผลการทดสอบกับเพลงไทยสุนทราภรณ์.....	53
5.6 ผลการทดสอบกับเพลงไทยลูกทุ่ง	54
5.7 ผลการเปรียบเทียบในเพลง Till I get it right.....	55
5.8 ผลการเปรียบเทียบในเพลง Sunrise.....	56
5.9 ผลการเปรียบเทียบในเพลงฟลอร์เฟื่องฟ้า	56
5.10 ผลการเปรียบเทียบจำนวนบิตที่ผิดพลาดในเพลง Till I get it right	59
5.11 ผลการเปรียบเทียบจำนวนบิตที่ผิดพลาดในเพลง Sunrise	59
5.12 ผลการเปรียบเทียบจำนวนบิตที่ผิดพลาดในเพลงฟลอร์เฟื่องฟ้า.....	60
จ.1 ผลการทดสอบอัลกอริทึมที่ใช้ในงานวิจัยวิทยานิพนธ์กับเพลงสากล classic.....	114
จ.2 ผลการทดสอบอัลกอริทึมที่ใช้รหัสแก้ไขความผิดพลาดกับเพลงสากล classic.....	114
จ.3 ผลการทดสอบอัลกอริทึมที่ใช้ในงานวิจัยวิทยานิพนธ์กับเพลงสากล jazz	115
จ.4 ผลการทดสอบอัลกอริทึมที่ใช้รหัสแก้ไขความผิดพลาดกับเพลงสากล jazz	115
จ.5 ผลการทดสอบอัลกอริทึมที่ใช้ในงานวิจัยวิทยานิพนธ์กับเพลงสากล pop.....	116
จ.6 ผลการทดสอบอัลกอริทึมที่ใช้รหัสแก้ไขความผิดพลาดกับเพลงสากล pop.....	116
จ.7 ผลการทดสอบอัลกอริทึมที่ใช้ในงานวิจัยวิทยานิพนธ์กับเพลงไทยเดิม.....	117
จ.8 ผลการทดสอบอัลกอริทึมที่ใช้รหัสแก้ไขความผิดพลาดกับเพลงไทยเดิม	117
จ.9 ผลการทดสอบอัลกอริทึมที่ใช้ในงานวิจัยวิทยานิพนธ์กับเพลงไทยสุนทราภรณ์	118
จ.10 ผลการทดสอบอัลกอริทึมที่ใช้รหัสแก้ไขความผิดพลาดกับเพลงไทยสุนทราภรณ์	118
จ.11 ผลการทดสอบอัลกอริทึมที่ใช้ในงานวิจัยวิทยานิพนธ์กับเพลงไทยลูกทุ่ง	119

สารบัญตาราง (ต่อ)

ตารางที่	หน้า
จ.12 ผลการทดสอบอัลกอริทึมที่ใช้รหัสแก้ไขความผิดพลาดกับเพลงไทยลูกทุ่ง.....	119
ฉ.1 ค่าความเข้มในการฟังลายน้ำของเพลงต่างๆ	121
ฉ.1 ค่าความเข้มในการฟังลายน้ำของเพลงต่างๆ (ต่อ)	122

สารบัญรูป

รูปที่

หน้า

1.1	ตัวอย่างภาพข้อมูลลายน้ำที่ใช้ในการฝัง	2
2.1	หน้าปกหนังสือของ Prota	6
2.2	ตัวอย่างข้อความ Steganography จากหนังสือของ Prota	6
3.1	ระนาบเวลาและความถี่สำหรับการแปลงแบบต่างๆ	22
3.2	ระนาบเวลาและความถี่สำหรับการแปลงเวฟเล็ต	23
3.3	การซ้อนทับของปริภูมิย่อยที่แผ่โดยสเกลลิงฟังก์ชัน	25
3.4	ฟังก์ชันสเกลลิงแบบ Haar (ซ้าย) และฟังก์ชันสเกลลิงแบบสามเหลี่ยม(ขวา)	26
3.5	ปริภูมิของฟังก์ชันสเกลลิงและฟังก์ชันเวฟเล็ต	27
3.6	กระบวนการแปลงเวฟเล็ตแบบดิสคริต	29
3.7	กระบวนการแปลงเวฟเล็ตแบบดิสคริต 2 ระดับ	29
3.8	กระบวนการแปลงเวฟเล็ตย้อนกลับ 2 ระดับ	29
3.9	เวฟเล็ตแบบ Haar	30
3.10	เวฟเล็ตแบบ Biorthogonal	31
3.11	เวฟเล็ตแบบ Coiflets	31
3.12	เวฟเล็ตแบบ Daubechies	32
3.13	เวฟเล็ตแบบ Mexican Hat	32
3.14	เวฟเล็ตแบบ Morlet	33
3.15	เวฟเล็ตแบบ Meyer	33
3.16	เวฟเล็ตแบบ Symlets	34
3.17	การคำนวณหาฟิลเตอร์ทั้งสี่ตัวโดยที่ qmf คือ quadrature mirror filters, wrev คือ การกลับลำดับเวกเตอร์จากซ้ายเป็นขวา	35
3.18	ฟิลเตอร์สำหรับการแปลงเวฟเล็ต	36
3.19	การคำนวณ cA_j และ cD_j	36

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.20	การคำนวณ cA_{j-1} และ cD_{j-1} 37
3.21	โครงสร้างแบบต้นไม้ที่ $j=3$ 37
3.22	กระบวนการรวมกลับสัญญาณ 38
3.23	ขั้นตอนการเลือก best_neighbor1 39
3.24	การค้นหาแบบ TS ที่ best_neighbor ต่างๆ 40
3.25	ขั้นตอนการทำงานของ back tracking ใน ATS 41
3.26	ขั้นตอนการทำงานของ adaptive radius ใน ATS 42
4.1	แสดงกระบวนการทำ permutation 43
4.2	ผังแสดงขั้นตอนการฝังลายน้ำทั้งหมด 45
4.3	ผังแสดงขั้นตอนย่อยที่ 4) ในการฝังลายน้ำ 45
4.4	ผังแสดงการค้นหาคำตอบด้วย ATS 48
4.5	ผังแสดงการตรวจจับลายน้ำ 49
ข.1	รูปภาพที่ใช้เป็นลายน้ำ 71
ข.2	ตัวอย่างภาพลายน้ำที่ตรวจจับได้จากสัญญาณเสียงหลังการโจมตี 71
ข.2	ตัวอย่างภาพลายน้ำที่ตรวจจับได้จากสัญญาณเสียงหลังการโจมตี (ต่อ) 72

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ลายน้ำเป็นเทคโนโลยีใหม่ที่นักวิทยาศาสตร์ด้านต่างๆมารวบรวมเข้าด้วยกันไม่ว่าจะเป็น การสื่อสาร สารสนเทศ การประมวลผลสัญญาณ เทคโนโลยีลายน้ำถูกนำมาใช้เพื่อป้องกันการละเมิดลิขสิทธิ์ของทรัพย์สินโดยไม่ได้รับอนุญาต ซึ่งเสียงก็เป็นทรัพย์สินในรูปแบบหนึ่งที่ปัจจุบันมีการละเมิดเป็นอย่างมากจากผลของเทคโนโลยีดิจิทัล เช่น MP3 ผ่านการสื่อสารข้อมูลที่รวดเร็วทางอินเทอร์เน็ตไม่ว่าจะทางคอมพิวเตอร์ โทรศัพท์มือถือ PDA ที่นับวันจะสร้างความเสียหายออกเป็นวงกว้าง เนื่องจากสื่อข้อมูลดิจิทัลเหล่านี้สามารถเข้าถึงได้อย่างรวดเร็ว ง่ายต่อการจัดเก็บ ทำสำเนา และเผยแพร่

ถึงแม้ว่าการเข้ารหัสจะช่วยปกป้องข้อมูลเหล่านี้จากผู้ละเมิดได้ ข้อเสียสำคัญประการหนึ่งคือ ผู้ได้รับข้อมูลต้องถอดรหัสข้อมูลก่อนนำไปใช้ ทำให้การปกป้องข้อมูลสิ้นสุดลง เกิดช่องทางการสำเนาและเผยแพร่อย่างผิดกฎหมายให้กับผู้อื่นที่ไม่ได้รับอนุญาต ดังนั้นการเข้ารหัสยังไม่สามารถแก้ปัญหาการละเมิดลิขสิทธิ์ได้ทั้งหมด

การสร้างลายน้ำดิจิทัล เป็นการฝังข้อมูลของผู้เป็นเจ้าของสื่อลงบนสื่อดิจิทัลแบบต่างๆ เช่น เสียง ภาพ และวิดีโอ ข้อมูลลายน้ำเหล่านี้ส่วนมากไม่สามารถตรวจรับรู้ได้ทันทีโดยประสาทสัมผัสของมนุษย์ เราเรียกข้อมูลที่ฝังนี้ว่า “ลายน้ำดิจิทัล” เมื่อมีการสำเนาสื่อ ข้อมูลลายน้ำต้องติดไปกับสื่ออื่นๆด้วย และต้องสามารถดึงข้อมูลลายน้ำเหล่านี้ออกมาเพื่อใช้แสดงสิทธิ์ความเป็นเจ้าของสื่อได้ในภายหลัง การฝังลายน้ำแบ่งออกเป็นสองช่วงสำคัญคือ ขั้นตอนการฝังลายน้ำลงบนสื่อ และการตรวจจับลายน้ำจากสื่อที่ถูกฝัง การฝังลายน้ำจำเป็นต้องเพิ่มความปลอดภัยของการเข้าถึงข้อมูลที่เป็นลายน้ำโดยใช้ “กุญแจ” โดยในขั้นตอนการตรวจจับลายน้ำออกจากสื่อจำเป็นต้องใช้กุญแจช่วยในการได้มาซึ่งข้อมูลลายน้ำที่ถูกฝัง หากปราศจากกุญแจจะไม่สามารถได้มาซึ่งข้อมูลลายน้ำที่ถูกฝังได้เลย

1.2 วัตถุประสงค์การวิจัย

- 1) เพื่อพัฒนาอัลกอริทึมที่ใช้ฝังลายน้ำลงบนสัญญาณเสียงดิจิทัล
- 2) เพื่อสนับสนุนการปกป้องสิทธิในผลงานเพลงและระบุความเป็นเจ้าของผลงานที่ถูกต้องและโดยอ้อมเพื่อปกป้องผลกระทบทางเศรษฐกิจอันนำไปสู่การละเมิดสิทธิทางปัญญา

1.3 ข้อยกเว้นเบื้องต้น

งานวิจัยนี้ใช้การค้นหาแบบตาบอดเชิงปรับตัว และการแปลงเวฟเลตแบบเต็มหน่วย ในการสร้างลายน้ำแบบกึ่งปิด (Semi-blind watermarking) การวิจัยจะมีการศึกษาภายใต้เงื่อนไขต่อไปนี้

- 1) พิจารณาสัญญาณเสียงดิจิทัลทั้งเสียงเพลงสากลและเพลงไทย บันทึกแบบ 16 บิต 44 kHz โดยเพลงสากลแบ่งออกเป็น pop, jazz, classic แบบละ 5 ตัวอย่าง ตัวอย่างละ 30 วินาที และเพลงไทยแบ่งออกเป็น ลูกทุ่ง ไทยเดิม ไทยสุนทราภรณ์ แบบละ 5 ตัวอย่าง ตัวอย่างละ 30 วินาที
- 2) พิจารณาข้อมูลลายน้ำที่เป็นภาพขาวดำขนาด 25×25 พิกเซล ดังรูปที่ 1.1
- 3) คุณภาพเสียงเพลงที่ฝังลายน้ำต้องมีค่า SNR ไม่ต่ำกว่า 20 เดซิเบล เมื่อเทียบกับเสียงต้นฉบับ
- 4) คุณภาพลายน้ำที่ตรวจจับได้จากสัญญาณเสียงที่ผ่านการโจมตี ต้องมีค่าสหสัมพันธ์ไม่ต่ำกว่า 0.9



รูปที่ 1.1 ตัวอย่างภาพข้อมูลลายน้ำที่ใช้ในการฝัง

1.4 ขอบเขตของการวิจัย

- 1) พัฒนาอัลกอริทึมที่ใช้ฝังลายน้ำลงบนสัญญาณเสียงดิจิทัลแบบกึ่งปิด
- 2) ข้อมูลลายน้ำต้องทนทานต่อการโจมตีด้วยการประมวลผลสัญญาณแบบต่างๆดังนี้
 1. Down-sampling, 44 kHz – 22 kHz – 44 kHz
 2. Lowpass filter, cutoff frequency 6 kHz, 6th order

3. Random noise กำลังงาน 1% ของสัญญาณเสียง
4. White Gaussian noise กำลังงาน 1% ของสัญญาณเสียง
5. กำจัดสัญญาณรบกวนด้วยการแปลงเวฟเล็ต (MATLAB)
6. บีบอัดข้อมูลแบบมีความสูญเสีย, MP3 ที่ 56 kbps
7. Crop ที่อัตราต่างๆดังนี้

- Jittering, ลบ 1 แชมเปิด จากทุกๆ 100 แชมเปิด
- Crop (5×100), ลบ 500 แชมเปิด จากตำแหน่งที่ได้จากการสุ่ม 5 ตำแหน่ง
- Crop (10×100), ลบ 1000 แชมเปิด จากตำแหน่งที่ได้จากการสุ่ม 10 ตำแหน่ง
- Crop (10×500), ลบ 5000 แชมเปิด จากตำแหน่งที่ได้จากการสุ่ม 10 ตำแหน่ง
- Crop (10×1000), ลบ 10000 แชมเปิด จากตำแหน่งที่ได้จากการสุ่ม 10 ตำแหน่ง

3) ตรวจสอบคุณภาพของสัญญาณเสียงที่ฟังลายน้แล้วต้องมีค่า SNR ไม่ต่ำกว่า 20 เดซิเบล และลายน้ต้องมีค่า data payload ไม่ต่ำกว่า 20 bps (bit per-second) ตามที่ International Federation of Phonographic Industry (IFPI) กำหนด

4) คุณภาพรูปภาพลายน้ที่กู้คืนได้จากสัญญาณเสียงหลังผ่านการโจมตีด้วยวิธีข้างต้นต้องมีค่าสหสัมพันธ์ไม่ต่ำกว่า 0.9

5) ใช้การค้นหาแบบตามูเชิงปรับตัว (Adaptive tabu search) ในกระบวนการฟังลายน้ เพื่อให้ได้ผลลัพธ์ที่เหมาะสมภายใต้ข้อกำหนดข้างต้น

6) ทำการจำลองอัลกอริทึมตามขอบเขตที่กำหนดเบื้องต้นด้วยโปรแกรม MATLAB และโปรแกรม AudioConverter v2.0.367

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ใช้ระบุความเป็นเจ้าของผลงาน การมีสิทธิและได้รับผลประโยชน์จากผลงานอย่างถูกต้อง
- 2) ลดปัญหาทางเศรษฐกิจและสังคมอันเนื่องจากการละเมิดลิขสิทธิ์
- 3) สามารถนำวิธีการทางด้านปัญญาประดิษฐ์มาประยุกต์ใช้ประโยชน์ด้านการฟังลายน้บนสัญญาณเสียง

1.6 รูปเล่มวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้ประกอบไปด้วยเนื้อหาทั้งหมด 6 บท และมีภาคผนวกทั้งสิ้น 5 ส่วน โดยมีรายละเอียดของเนื้อหาในแต่ละบทดังนี้

บทที่ 1 กล่าวถึงความเป็นมาและความสำคัญของปัญหา วัตถุประสงค์การวิจัย ข้อตกลงเบื้องต้น ขอบเขตของการวิจัย และประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย

บทที่ 2 เกี่ยวกับการสร้างลายน้ำ ลิขสิทธิ์ของไทย สัญญาณเสียง การทบทวนงานวิจัยในอดีตที่เกี่ยวข้องกับการฝังลายน้ำลงบนสัญญาณเสียงดิจิทัลด้วยระเบียบวิธีต่าง ๆ และข้อจำกัดของงานวิจัยเหล่านั้นสำหรับการประยุกต์ใช้ในการสร้างลายน้ำ

บทที่ 3 นำเสนอทฤษฎีที่เกี่ยวข้องกับงานวิจัย ได้แก่ การแปลงเวฟเล็ต และการค้นหาแบบตามเชิงปรับตัว

บทที่ 4 กล่าวถึงรายละเอียดการฝังลายน้ำ ลงบนสัญญาณเสียงดิจิทัลในเวฟเล็ตโดเมนโดยการประยุกต์ใช้การค้นหาแบบตามเชิงปรับตัว เพื่อให้ลายน้ำที่ฝังมีความทนทานต่อการโจมตีด้วยการประมวลผลสัญญาณแบบต่างๆ และการบีบอัดแบบมีความสูญเสีย โดยที่คุณภาพเสียงสามารถยอมรับได้

บทที่ 5 แสดงผลการทดสอบความทนทานของลายน้ำ ที่ฝังลงบนสัญญาณเสียงดิจิทัลต่อการโจมตีแบบต่างๆ ด้วยระเบียบวิธีของงานวิจัย แสดงผลการทดสอบโปรแกรมที่ได้ค้นคว้าวิจัยเปรียบเทียบกับระเบียบวิธีที่มีอยู่ ตลอดจนผลของการรับฟังเสียงเพลงที่ผ่านการฝังลายน้ำจากผู้เข้ารับฟัง วิเคราะห์และอภิปรายผลการทดสอบ

บทที่ 6 สรุปผลการทดสอบทั้งหมดของงานวิจัยและข้อเสนอแนะสำหรับการวิจัยในอนาคต ภาคผนวกประกอบด้วยเนื้อหา 5 ส่วน ได้แก่ ภาคผนวก ก. เป็นรายละเอียดของข้อมูลเสียงเพลงทั้งหมดที่นำมาใช้ทดสอบสำหรับงานวิจัยนี้ ภาคผนวก ข. แสดงรูปภาพขาวดำที่ใช้เป็นลายน้ำในงานวิจัยและรูปภาพลายน้ำที่ตรวจจับได้หลังผ่านการโจมตี ภาคผนวก ค. โปรแกรม MATLAB ที่ใช้ในงานวิจัยวิทยานิพนธ์ ภาคผนวก ง. โปรแกรม MATLAB ที่ใช้รหัสแก้ไขความผิดพลาด (Error correction code) เพื่อเปรียบเทียบผลการทดสอบกับโปรแกรม MATLAB ที่ใช้ในงานวิจัยวิทยานิพนธ์ ภาคผนวก จ. ผลการทดสอบความทนทานของลายน้ำจากอัลกอริทึมที่เสนอในงานวิจัยวิทยานิพนธ์ และผลการทดสอบความทนทานของลายน้ำจากอัลกอริทึมที่ใช้รหัสแก้ไขความผิดพลาด ภาคผนวก ฉ. แสดงค่าความเข้มในการฝังลายน้ำของเพลงต่างๆที่ได้จากการค้นหาแบบตามเชิงปรับตัว ภาคผนวก ช. แบบทดสอบคุณภาพเสียงหลังจากการฝังลายน้ำโดยการรับฟัง ภาคผนวก ซ. บทความที่ได้รับการตีพิมพ์เผยแพร่

บทที่ 2

การสร้างลายน้ำและปริทรรศน์วรรณกรรมที่เกี่ยวข้อง

2.1 กล่าวนำ

บทนี้กล่าวถึงประวัติความเป็นมาของการสร้างลายน้ำ และความรู้พื้นฐานต่างๆที่จำเป็นเกี่ยวกับการสร้างลายน้ำ และปริทรรศน์วรรณกรรมต่างๆที่ได้ศึกษาเกี่ยวกับการฝังลายน้ำลงบนสัญญาณเสียงแบบดิจิทัลด้วยวิธีการต่างๆกัน จุดเด่นและจุดด้อยของวิธีการต่างๆที่ควรคำนึงถึง

2.2 ลายน้ำ

2.2.1 ประวัติความเป็นมา

ปัจจุบันสื่อดิจิทัลเช่น ภาพ และเสียง มีการใช้งานอย่างแพร่หลาย เพราะสามารถทำสำเนาได้ง่าย อีกทั้งยังสามารถส่งถึงกันผ่านทางอินเทอร์เน็ต ปัญหาที่ตามมาคือการละเมิดลิขสิทธิ์ ถึงแม้ว่าการเข้ารหัสจะช่วยให้ปกป้องข้อมูลเหล่านี้ได้แต่มีข้อเสียที่ก่อให้เกิดช่องทางการสำเนา และนำไปสู่การเผยแพร่ให้กับผู้อื่นที่ไม่ได้รับอนุญาตต่อไป ดังนั้นการเข้ารหัสยังไม่ใช่วิธีที่สามารถแก้ปัญหาการละเมิดลิขสิทธิ์ได้ทั้งหมด จึงได้มีแนวทางใหม่ในการแก้ปัญหาคือ วิธีการสร้างลายน้ำ

การสร้างลายน้ำดิจิทัล (Digital watermarking) เป็นการฝังข้อมูลลงในสื่อดิจิทัลแบบต่างๆเช่น เสียง ภาพ และวิดีโอ ข้อมูลลายน้ำเหล่านี้ไม่สามารถตรวจรับรู้ได้ทันทีโดยประสาทสัมผัสของมนุษย์ เราเรียกข้อมูลที่ฝังนี้ว่า “ลายน้ำดิจิทัล” เมื่อมีการสำเนาสื่อ ข้อมูลลายน้ำต้องติดไปกับสื่อเหล่านั้นด้วย และต้องสามารถดึงข้อมูลลายน้ำเหล่านี้ออกมาเพื่อใช้แสดงสิทธิ์ความเป็นเจ้าของสื่อได้ในภายหลัง การฝังลายน้ำแบ่งออกเป็นสองช่วงสำคัญคือ ขั้นตอนการฝังลายน้ำลงบนสื่อ และการตรวจจับลายน้ำออกจากสื่อที่ถูกฝัง การฝังลายน้ำจำเป็นต้องเพิ่มความปลอดภัยของการเข้าถึงข้อมูลที่เป็นลายน้ำโดยใช้ “กุญแจ” โดยขั้นตอนการดึงลายน้ำออกจากสื่อจำเป็นต้องใช้กุญแจช่วยในการดึงหากปราศจากกุญแจจะไม่สามารถได้ข้อมูลลายน้ำที่สมบูรณ์ได้

ในสมัยอียิปต์โบราณกว่า 4000 ปีที่แล้ว มีการค้นพบลายน้ำจากสัญลักษณ์ที่เป็นเครื่องหมายลับ ปรากฏจารึกอยู่ในสุสานของชนชั้นสูง แสดงให้เห็นว่าเทคนิคการฝังลายน้ำมีมาแต่สมัยโบราณในรูปแบบต่างๆ ซึ่งลายน้ำจัดเป็นศาสตร์แขนงหนึ่งของชาวเลข หรือ Steganography ซึ่งมาจากภาษากรีกที่ว่า *steyavos* แปลว่า การปกปิด และ *γραφω* ที่แปลว่า เขียน แสดงให้เห็นการสื่อสารข้อมูลลับ ลายน้ำปรากฏในบันทึกครั้งแรกที่ประเทศจีน โดยการทำลวดลายลงบนกระดาษ

จากนั้นปี ค.ศ. 1282 ถายน้ำปรากฏขึ้นอีกครั้งในโรงงานผลิตกระดาษที่อิตาลี เพื่อระบุที่มาของกระดาษว่าผลิตมาจากโรงงานใด

ในระหว่างช่วงปี ค.ศ. 1462-1526 Johannes Trithemius พระชาวเยอรมันได้เข้ารหัสตัวหนังสือเป็นคำสอนทางศาสนา และถูกสำนักกวาดล้างประณามการกระทำเช่นนี้ว่าเป็น “full of peril and superstition” ต่อมาในปี ค.ศ. 1593 GiovanniBaptista Porta ได้ตีพิมพ์หนังสือ ชื่อ “De occultis literarum notis seu artis anime occulte alijs significadi, aui asb alijs Significata expiscandi enodandique. Libri III” ภายในหนังสือเขาได้เสนอวิธีการเกี่ยวกับการสื่อสารข้อมูลลับแบบต่างๆหรือ Steganography ไว้มากมาย เช่น การอ่านข้อมูลลับที่ได้จากการปิดหน้ากากลงบนข้อความที่ต้องการ โดยอ่านเฉพาะข้อความที่ไม่ถูกปิดไว้ดังรูป 2.2



รูปที่ 2.1 หน้าปกหนังสือของ Porta (Hanjalic, 2000)

Honor Militiae tuus suit Carolus pater, nam cum infinito victus est, cum minima exercitu inuitus parte hostis fugit, ac prope ultimum diem iniurius peribit, necabit Bere illum; atque extemplo puer Arato peribit, res omnes deprehensae bonae si sunt, ante Sillam, & optimo capite non poenitentias amplius decidere sperabit. Vale.

รูปที่ 2.2 ตัวอย่างข้อความ Steganography จากหนังสือของ Porta (Hanjalic, 2000)

ต่อมาในศตวรรษที่ 17 Bishop Francis Godwin ได้เข้ารหัสชื่อของเขาลงเป็นตัวอักษรแรกของแต่ละบทความในต้นฉบับที่เขาเขียนขึ้น จึงนับเป็นตัวอย่างแรกๆของการปกป้องลิขสิทธิ์ เนื่องจากในสมัยนั้นมีการตีพิมพ์บทความโดยบิดเบือนชื่อผู้แต่ง โดยเฉพาะอย่างยิ่งบทความเกี่ยวกับประวัติศาสตร์และการเมือง ทำให้ต้องรอบคอบในบทความที่น่าเสนอเป็นอย่างมาก

ตัวอย่างของการฝังข้อมูลที่ระบุลิขสิทธิ์ความเป็นเจ้าของลงบนผลงานเพลง ถูกนำมาใช้โดย Bach โดยการฝังชื่อของเขาที่สะกดโดยอักษร B-A-C-H ลงบนโน้ตเพลง “Vor deinem Thron” ที่ใช้กับคณะร้องประสานเสียงในโบสถ์ โดย โน้ต B-flat แทนอักษร B และโน้ต B แทนอักษร H อีกทั้งการนับจำนวนการเกิดโน้ตที่สัมพันธ์กับลำดับอักษรในภาษาอังกฤษ กล่าวคือ A นับได้หนึ่งครั้ง B นับได้สองครั้ง C นับได้สามครั้ง และ H นับได้แปดครั้งแทน (Hanjalic, 2000)

2.2.2 การประยุกต์ใช้ลายน้ำ

การฝังลายน้ำสามารถใช้ประโยชน์ได้หลายทางเช่น

1) Copyright protection ใช้ปกป้องผลผลิตทางปัญญา ผู้เป็นเจ้าของข้อมูลจะฝังลายน้ำที่ระบุลิขสิทธิ์ลงบนข้อมูลเหล่านั้น เพื่อใช้เป็นหลักฐานในการพิสูจน์สิทธิ์ ในชั้นศาลกรณีเกิดการละเมิดลิขสิทธิ์

2) Fingerprinting เป็นเทคนิคที่ใช้ติดตามการทำซ้ำที่ผิดกฎหมาย โดยเจ้าของจะทำการฝังลายน้ำลงบนข้อมูลที่ทำซ้ำแต่ละชิ้นให้มีรูปแบบที่แตกต่างกัน เพื่อแจกจ่ายให้กับลูกค้าแต่ละราย ลายน้ำดังกล่าวจะคล้ายกับลายนิ้วมือ กล่าวคือข้อมูลเช่น เลขหมายประจำตัวลูกค้า หรือหลักฐานที่แสดงความสัมพันธ์ระหว่างข้อมูลกับลูกค้ารายนั้นๆ จะถูกบันทึกเป็นลายน้ำไว้ เมื่อลูกค้ารายใดทำการละเมิดลิขสิทธิ์โดยการเผยแพร่ข้อมูลให้กับบุคคลที่สามโดยผิดกฎหมาย เจ้าของสามารถใช้ข้อมูลที่ฝังลงเป็นลายน้ำนี้เพื่อพิสูจน์การละเมิดดังกล่าวได้

3) Copy protection ข้อมูลที่ใช้เป็นลายน้ำสามารถใช้ในการควบคุมเครื่องบันทึกแบบดิจิทัลเพื่อการป้องกันการทำซ้ำได้โดยเทคนิคพิเศษ ข้อมูลลายน้ำจะบอกจำนวนการทำซ้ำได้และอนุญาตให้เครื่องบันทึกทำซ้ำได้ตามที่อนุญาตเท่านั้น

4) Broadcast monitoring เป็นการฝังสัญญาณลายน้ำลงบนโฆษณาที่ออกอากาศ โดยระบบตรวจจับอัตโนมัติจะทำการตรวจเช็คว่ามีโฆษณาออกมามาตรงตามที่ตกลงไว้หรือไม่ นอกจากนี้ระบบตรวจจับยังสามารถเช็คการออกอากาศสัญญาณจากสถานีโทรทัศน์ในแต่ละช่องและทำการเรียกเก็บค่าธรรมเนียมจากจำนวนและระยะเวลาการโฆษณาได้จากข้อมูลเหล่านี้อีกด้วย

5) Data authentication การฝังลายน้ำแบบ fragile watermarking เป็นการฝังลายน้ำเพื่อใช้รับรองการปลอมแปลงข้อมูล เนื่องจากลายน้ำประเภทนี้จะเสียหายทันทีเมื่อมีการเปลี่ยนแปลงแก้ไขข้อมูลเกิดขึ้น

นอกจากนี้ลายน้ำยังสามารถใช้ประโยชน์ในทางต่างๆได้เช่น

- 1) Indexing เป็นการจัดสารบัญลงบนสื่อต่างๆเช่น วีดีโอ ภาพยนตร์ โดยการฝังคำอธิบายหรือข้อคิดเห็นลงบนข้อมูลเหล่านี้
- 2) Medical safety เป็นการฝังลายน้ำที่ระบุข้อมูลเกี่ยวกับคนไข้เช่น ชื่อ วันและเวลาลงบนรูปที่ใช้เป็นข้อมูลรักษาทางการแพทย์ เพื่อความปลอดภัย
- 3) Data hiding เป็นการซ่อนข้อมูลลับโดยเทคนิคการฝังลายน้ำลงบนสื่อต่างๆไปที่ต้องการ ซึ่งข้อมูลลายน้ำดังกล่าวสามารถล่วงรู้ได้เฉพาะผู้มีสิทธิ์เท่านั้น บุคคลที่สามไม่สามารถสังเกตหรือรับรู้ถึงการมีอยู่ของข้อมูลลับที่ซ่อนอยู่ในสื่อเหล่านี้ได้เลย

2.2.3 คุณสมบัติที่สำคัญของลายน้ำ

- 1) ไม่สามารถมองเห็นได้ (Invisibility) ลายน้ำที่ฝังจะต้องไม่สามารถตรวจรับรู้จากสื่อได้ทันทีโดยประสาทสัมผัสของมนุษย์เช่น การมองเห็น หรือการได้ยิน
- 2) ความทนทาน (Robustness) ลายน้ำต้องทนทานต่อการโจมตีจากกระบวนการประมวลผลทางสัญญาณต่างๆเพื่อความทนทานต่อการโจมตีเพื่อการละเมิดลิขสิทธิ์
- 3) ความปลอดภัย (Security) การฝังลายน้ำจำเป็นต้องใช้กุญแจร่วมในการฝังด้วย เมื่อมีการดึงลายน้ำออกมาจากสื่อ จะไม่สามารถนำลายน้ำนั้นมาเป็นประโยชน์ได้หากปราศจากกุญแจกุญแจจะถูกเก็บไว้เป็นความลับเพื่อความปลอดภัย ดังนั้นผู้ที่ไม่มีกุญแจจะไม่สามารถได้มาซึ่งลายน้ำที่สมบูรณ์ได้

2.2.4 การแบ่งประเภทของลายน้ำ

ลายน้ำถูกแบ่งออกเป็นชนิดต่างๆหลายประเภทดังยกตัวอย่าง ดังนี้

แบ่งตามความสามารถในการมองเห็น

- 1) Visible watermarking เป็นลายน้ำที่มองเห็นได้เช่นเครื่องหมายการค้า โลโก้ต่างๆสามารถสังเกตได้โดยตรงโดยการฝังทับลงบนสื่อทำให้คุณภาพของสื่อลดลง
- 2) Invisible watermarking เป็นลายน้ำที่ไม่สามารถมองเห็นได้ โดยคุณภาพของสื่อจะไม่ใกล้เคียงกับต้นฉบับ

แบ่งตามความทนทานของลายน้ำ

- 1) Robust watermarking เป็นลายน้ำที่มีความทนทานต่อการโจมตีทางกระบวนการประมวลผลสัญญาณแบบต่างๆ เป็นลายน้ำที่มีการประยุกต์ใช้อย่างกว้างขวาง
- 2) Semi-fragile watermarking คล้ายกับลายน้ำแบบแรกแต่ทนทานต่อการโจมตีของการประมวลผลสัญญาณเฉพาะแบบ ตามที่กำหนดไว้เท่านั้น

3) Fragile watermarking เป็นลายน้ำที่ไม่ทนทานต่อการโจมตีทางการประมวลผลสัญญาณเลข ส่งผลให้ลายน้ำเสียหายจากการโจมตีได้ง่าย นิยมใช้เพื่อประโยชน์ในการตรวจสอบว่าสื่อเป็นของจริงที่ไม่ถูกดัดแปลงแก้ไข

แบ่งตามวิธีที่ใช้ในการตรวจจับลายน้ำ

1) Public watermarking วิธีการนี้จำเป็นต้องใช้สื่อต้นฉบับในการดึงข้อมูลลายน้ำ เป็นวิธีการที่ไม่นิยมเนื่องจากเสี่ยงต่อการล่วงรู้ถึงสื่อต้นฉบับโดยผู้ละเมิด

2) Semi-private watermarking วิธีการนี้ไม่จำเป็นต้องใช้สื่อต้นฉบับในการดึงข้อมูลลายน้ำ แต่ต้องการเพียงข้อมูลที่มีความสัมพันธ์กับสื่อต้นฉบับและลายน้ำ

3) Private watermarking วิธีการนี้สามารถดึงข้อมูลลายน้ำออกมาจากสื่อได้โดยไม่ต้องอาศัยข้อมูลจากสื่อต้นฉบับหรือลายน้ำเลย ทำให้การฝังลายน้ำแบบนี้เป็นที่ปลอดภัยมากที่สุด

แบ่งตามกระบวนการฝังลายน้ำ

1) Spatial Domain เป็นการสร้างลายน้ำในโดเมนเวลา สามารถทำได้ง่าย

2) Frequency Domain เป็นการสร้างลายน้ำในโดเมนความถี่ ซึ่งมีความทนทานกว่าการสร้างลายน้ำในโดเมนเวลา แต่มีข้อเสียคืออัลกอริทึมมีความซับซ้อน

2.3 ลิขสิทธิ์ของไทย

งานลิขสิทธิ์ของไทยเริ่มตั้งแต่ปี พ.ศ. 2435 โดยผ่านประกาศหอพระสมุดวชิรญาณ แม้ว่าประกาศนี้จะเป็นเรื่องภายในประเทศ แต่ประกาศนี้ถือว่าเป็นกฎหมายลิขสิทธิ์ เนื่องจากมีเนื้อหาสอดคล้องกับแนวความคิดของลิขสิทธิ์ ในยุคถัดมาพระราชบัญญัติ พ.ศ. 2444 จะพบหลักลิขสิทธิ์ของประเทศสหราชอาณาจักรในระบบกฎหมายไทย และไทยยอมรับกฎเกณฑ์ของลิขสิทธิ์ระหว่างประเทศภายใต้อนุสัญญาเบิร์นในพระราชบัญญัติปี พ.ศ. 2474 ปัจจุบันตั้งแต่ประเทศไทยยอมรับมาตรฐานของความตกลง Trade-Related Aspects of Intellectual Property Rights (TRIPs) ในพระราชบัญญัติลิขสิทธิ์ พ.ศ. 2537 ซึ่งมีผลใช้บังคับตั้งแต่วันที่ 21 มีนาคม พ.ศ. 2538 และยังไม่มีการแก้ไขกฎหมายดังกล่าว เนื่องจากพระราชบัญญัตินี้สอดคล้องกับความตกลง TRIPs ตั้งแต่ที่มีการตรากฎหมายจึงไม่มีความจำเป็นที่จะต้องแก้ไขบทวนพระราชบัญญัตินี้อีกให้สอดคล้องกับหลักเกณฑ์ของความตกลง TRIPs ในปี พ.ศ. 2542 สำหรับกฎหมายลิขสิทธิ์ในประเทศไทยได้ให้ความหมายของลิขสิทธิ์ไว้ ซึ่งกล่าวพอสังเขปได้ดังนี้

ลิขสิทธิ์หมายถึง สิทธิแต่เพียงผู้เดียวที่จะกระทำการใดๆเกี่ยวกับงานที่ผู้อื่นสร้างสรรค์ได้ทำขึ้น ลิขสิทธิ์เป็นผลงานที่เกิดจากสติปัญญา ความรู้ความสามารถ ความวิริยะอุตสาหะ ในการสร้าง

สรรคงานให้เกิดขึ้นนั้น มีองค์ประกอบ 4 ข้อของการได้มาซึ่งลิขสิทธิ์ซึ่งได้รับการยอมรับกันในกฎหมาย ลิขสิทธิ์ทั้งของต่างชาติและไทย ดังนี้

- การแสดงออกซึ่งความคิด
- เป็นการแสดงออกซึ่งงานในชนิดที่กฎหมายรับรอง
- ต้องเป็นงานที่สร้างสรรค์ด้วยตนเอง
- ต้องไม่ใช่งานอันขัดต่อกฎหมาย

งานสร้างสรรค์ที่มีลิขสิทธิ์ประกอบไปด้วยงานต่างๆดังต่อไปนี้

- งานวรรณกรรม เช่น หนังสือ จุลสาร โปรแกรมคอมพิวเตอร์
- งาน นาฏกรรม เช่น งานเกี่ยวกับการรำ การเต้น การทำท่า หรือการแสดงเป็นเรื่องราวการแสดงโดยวิธีใบ้
- งานศิลปกรรม เช่น งานจิตรกรรม สถาปัตยกรรม ประติมากรรม ภาพพิมพ์ภาพประกอบแผนที่ โครงสร้าง ศิลปะประยุกต์ ภาพถ่าย และแผนผัง ของงานดังกล่าว
- งานดนตรีกรรม เช่น เนื้อร้อง ทำนอง โน้ตเพลงที่ได้มีการเรียบเรียงเสียงประสานแล้ว
- งานโสตทัศนวัสดุ เช่น วีดีโอเทป
- งานภาพยนตร์
- งานสิ่งบันทึกเสียง เช่น เทปเพลง แผ่นคอมแพคดิสก์
- งานแพร่เสียงแพร่ภาพ
- งานอื่นใดในแผนกวรรณคดี แผนกวิทยาศาสตร์ หรือ แผนกศิลปะ

ลิขสิทธิ์ในสิทธิจะเกิดขึ้นทันที ตั้งแต่ผู้สร้างสรรค์ได้สร้างสรรค์ผลงานออกมาโดยไม่ต้องมีการจดทะเบียนเหมือนกับเครื่องหมายการค้า หรือขอความคุ้มครองเหมือนสิทธิบัตร อย่างไรก็ตามผู้สร้างสรรค์สามารถยื่นคำขอแจ้งงานสร้างสรรค์ของตนต่อกรมทรัพย์สินทางปัญญาได้ (สรัญญา กาญจนรินทร์, 2547)

2.4 รูปแบบสัญญาณเสียงดิจิทัล

2.4.1 ประวัติความเป็นมา

ในสมัยก่อนที่คอมพิวเตอร์จะเข้ามามีบทบาทสำคัญอย่างในปัจจุบัน สัญญาณเสียงจะถูกบันทึกลงบนซีดีเพื่อการเล่นกับเครื่องเล่นซีดีเพียงอย่างเดียวเท่านั้น เราเรียกรูปแบบในลักษณะนี้ว่า Compact Disk - Digital Audio (CD-DA) ซึ่งจะจัดเก็บเป็นรูปแบบมาตรฐานชื่อว่า red book คิดค้นโดยบริษัท Sony และ Philips ซึ่งเป็นผู้คิดค้นต้นแบบเทคโนโลยีของซีดีขึ้น รูปแบบดังกล่าวถูกนำออกมาเผยแพร่ตั้งแต่ปี ค.ศ. 1980 ซึ่ง red book ไม่เป็นแค่เพียงการจัดเก็บสัญญาณ

เสียงเท่านั้น แต่ยังคำนึงถึงความสัมพันธ์ต่อลักษณะของแผ่นซีดี ขนาดที่เป็นไปได้ของข้อมูลที่ทำให้การจัดเก็บ ระยะห่างระหว่างแทร็ค เหล่านี้เป็นต้น ซึ่งจำเป็นอย่างยิ่งในการระบุว่าข้อมูลเหล่านี้จะถูกอ่านให้ถูกต้องได้อย่างไร สำหรับซีดีในช่วงต่อมาก็ได้อาศัยหลักการพื้นฐานของ red book นี้เอง เพื่อจัดเก็บสื่อข้อมูล

การจัดเก็บข้อมูลในรูปแบบของ CD-DA เริ่มจากการชักตัวอย่างจากแหล่งกำเนิดเสียง จากนั้นจะถูกเข้ารหัสดิจิทัลเพื่อแปลงเป็น CD-DA โดยใช้อัตราการชักตัวอย่างที่ 44.1 kHz ซึ่งสูงกว่าความถี่ที่มนุษย์สามารถได้ยินถึงสองเท่า ซึ่งแต่ละแซมเปิลจะมีความละเอียด 16 บิต แบบสเตอริโอ ซึ่งใช้พื้นที่ในการจัดเก็บประมาณ 176 kbps

ข้อมูลที่เป็นสัญญาณเสียงจะถูกจัดเก็บบนดิสก์ในลักษณะเป็นบล็อกหรือบางที่เรียกว่า เซกเตอร์ซึ่งแต่ละบล็อกจะบรรจุข้อมูลได้ 2,352 ไบต์ในจำนวนนี้ประกอบด้วยไบต์ที่ใช้เพื่อการทำรหัสแก้ไขความผิดพลาด และโครงสร้างควบคุม ดังนั้นเสียงที่นาน 1 วินาทีต้องใช้ 75 บล็อกในการจัดเก็บหรือ ซีดีมาตรฐาน 74 นาทีจะมีจำนวนข้อมูลประมาณ 747 MB หรือสัญญาณเสียงที่ไม่มี การบีบอัดนาน 1 นาที เมื่อจัดเก็บในรูปแบบ CD (44.1 kHz, 16 บิต, สเตอริโอ) จะใช้พื้นที่ในการจัดเก็บประมาณ 10 MB (Pcguide, 2005)

2.4.2 รูปแบบไฟล์ WAV

เราสามารถใช้โปรแกรมที่ออกแบบมาเป็นพิเศษให้สามารถอ่านข้อมูลจาก CD-DA แล้วจัดเก็บลงบนคอมพิวเตอร์ในรูปแบบไฟล์เสียงแบบต่างๆได้ โดยผ่านกระบวนการสกัดสัญญาณ ซึ่งข้อมูลดิจิทัลเหล่านี้จะถูกจัดเก็บเป็นลำดับตามลักษณะของรูปคลื่น แล้วถูกนำมาบีบอัดโดยวิธีต่างๆกันแล้วแต่ความต้องการในการลดขนาดข้อมูลของการจัดเก็บแต่ละชนิด ซึ่ง WAV ก็เป็นรูปแบบหนึ่งของรูปแบบไฟล์เสียงที่ใช้ในการจัดเก็บแบบดิจิทัล ไฟล์นามสกุล WAV นี้ถูกคิดขึ้นโดยบริษัท Microsoft ร่วมมือกับ IBM สำหรับระบบปฏิบัติการ Windows รูปแบบไฟล์เสียงชนิดนี้จะบรรจุข้อมูลที่เกี่ยวกับลักษณะสมบัติของรูปคลื่นสัญญาณเสียงรวมไปถึงรูปแบบของการจัดเก็บ เช่น อัตราการสุ่มของสัญญาณเสียง bit-depth ชนิดของการบีบอัดข้อมูล ความยาวของรูปคลื่น และอัตราการอ่านข้อมูลต่อวินาที เป็นต้น โดยปราศจากการบีบอัดข้อมูล ทำให้สัญญาณเสียงที่ได้จาก WAV สามารถนำไปใช้ประโยชน์อย่างกว้างขวางในการประมวลผลสัญญาณแบบต่างๆได้อย่างดี ซึ่งจากคุณสมบัติดังกล่าวทำให้ไม่ต้องสงสัยเลยว่าทำไม WAV จึงเป็นที่นิยมอย่างแพร่หลาย

2.4.3 รูปแบบไฟล์ MP3

ในปัจจุบันโลกแห่งการสื่อสารแบบดิจิทัลขยายตัวมากขึ้น ไฟล์เพลงต่างๆ ถูกลดขนาดลงเพื่อให้ส่งผ่านถึงกันได้รวดเร็วขึ้น ไฟล์แบบ MP3 จึงถือกำเนิดขึ้น โดยอาศัยการเข้ารหัสไฟล์ WAV เพื่อลดขนาดข้อมูลลง ซึ่งปกติความยาวของเพลงหนึ่งเพลงที่จัดเก็บแบบ WAV จะมีขนาดประมาณ 50 MB ทำให้มีขนาดใหญ่มากเกินไปที่จะจัดส่งบนอินเทอร์เน็ตได้ ทำให้ไฟล์แบบ MP3 เข้ามามีบทบาทสำคัญในการลดขนาดข้อมูลเหล่านี้ลงถึง 10-12 เท่า หรือขนาดแค่ 4-5 MB กระบวนการเหล่านี้เป็นการลดคุณภาพของสัญญาณเสียงลง ถึงแม้ว่าจะทำการถอดรหัสข้อมูลกลับมาสู่ WAV แล้วก็ตามแต่ไม่ทำให้คุณภาพเสียงที่ลดลงไปกลับดีขึ้นมาได้ จึงจัดว่าเป็นการบีบอัดแบบสูญเสีย และเนื่องจากไฟล์แบบ MP3 ผ่านการเข้ารหัสมาแล้วทำให้ไม่สามารถประมวลผลสัญญาณได้โดยตรงหากแต่ต้องแปลงเป็น WAV ก่อนเสมอ

ในปลายปี ค.ศ. 1980 MP3 ถูกคิดค้นและพัฒนาโดย Fraunhofer-Gesellschaft และ Thomson Multimedia และตั้งแต่ปี ค.ศ. 1997 เป็นต้นมาก็เป็นที่นิยมอย่างรวดเร็วบนอินเทอร์เน็ต MP3 เป็นชื่อย่อของ MPEG (Moving Pictures Experts Group) Layer – 1 ซึ่งเป็นกลุ่มที่สนับสนุนให้ MP3 มีมาตรฐานระดับนานาชาติ ในขั้นตอนการแปลง MP3 สามารถกำหนดขนาดของไฟล์ที่สอดคล้องกับคุณภาพของเสียงได้โดยการกำหนดอัตราบิต (Bit rate) หมายถึงจำนวนบิตที่สามารถส่งได้ต่อวินาที (Bit per second) ความหมายของ อัตราบิต คล้ายกับจำนวนเฟรมของฟิล์มหนังที่ยิ่งมีมากยิ่งให้คุณภาพที่ดี สำหรับสัญญาณเสียง อัตราบิต ยิ่งมากสัญญาณเสียงที่ได้ยิ่งมีคุณภาพดีแต่ไฟล์ย่อมมีขนาดใหญ่ตามไปด้วย สำหรับ MP3 จะเขียน อัตราบิต อยู่ในรูปของ kbps ส่วนมากมักตั้งค่านี้ไว้ที่ 128 kbps เนื่องจากให้ไฟล์ขนาดไม่ใหญ่มากนักในขณะที่มีคุณภาพเสียงที่ใช้ได้ นอกจากนี้การกำหนด bit rate สามารถกำหนดได้หลายแบบ เช่น constant bit rate (CBR), average bit rate (ABR), variable bit rate (VBR) เป็นต้น

เป็นที่วิพากษ์วิจารณ์กันเป็นอย่างมากในผลกระทบของ MP3 ต่อลิขสิทธิ์ของเพลงที่ทำการบีบอัด ความจริงแล้ว MP3 เป็นไฟล์ที่ทำการจัดเก็บสัญญาณเสียงประเภทหนึ่งเท่านั้น จึงจัดได้ว่าเป็นรูปแบบที่ถูกกฎหมาย แต่การใช้ MP3 จะเป็นการผิดกฎหมายก็ต่อเมื่อมีการทำซ้ำ เผยแพร่ หรือการกระทำอื่นใด ซึ่งขัดต่อกฎหมายลิขสิทธิ์ของแต่ละประเทศโดยไม่ได้รับการอนุญาตที่ถูกต้องจากเจ้าของลิขสิทธิ์เสียก่อน

2.5 ปริทรรศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

ในปัจจุบันมีการนำเสนอเทคนิคหลากหลายในการฝังลายน้ำบนสัญญาณเสียงดิจิทัล ซึ่งอาจจะแบ่งออกได้เป็นสองจำพวกใหญ่ๆ คือ การฝังลายน้ำลงบนโดเมนเวลา และอีกพวกคือ การฝังลายน้ำลงบนโดเมนความถี่ ซึ่งมีความทนทานต่อการโจมตีมากกว่าโดเมนเวลา (Shao Ya-fei, Wu Guo-wei and Lin Xing-gang (2001), Jianzhen Wu and Jianying Xie (2003))

Changtsheng Xu, Jiankang Wu and Qibin Sun (1999) ได้สรุปเทคนิคต่างๆ ที่นิยมในการฝังลายน้ำลงบนโดเมนเวลา เช่น Low-bit coding เป็นวิธีการฝังลายน้ำลงบนสัญญาณเสียงโดยการแทนที่บิตที่มีนัยสำคัญน้อยในแต่ละแชนเนลของข้อมูลสัญญาณเสียง ด้วยรหัสเลขฐานสองที่ได้จากข้อมูลลายน้ำ ยกตัวอย่างเช่น ในสัญญาณเสียงแบบ 16 บิต บิตที่มีนัยสำคัญน้อยสุด 4 ตัวหลังจะถูกลบออกแล้วแทนที่ด้วยรหัสของข้อมูลลายน้ำ วิธี Low-bit coding นี้จัดได้ว่าเป็นวิธีการพื้นฐานที่ง่ายในการฝังข้อมูลลายน้ำลงบนสัญญาณเสียง ข้อเสียของเทคนิคนี้คือความไม่คงทนต่อสัญญาณรบกวนและการเปลี่ยนอัตราสุ่ม (Resampling) เป็นต้น, Phase coding เป็นอีกวิธีการหนึ่งที่ดีกว่ามีประสิทธิภาพเมื่อทำการคำนึงถึงการชี้วัดคุณภาพเสียง เช่น อัตราส่วนสัญญาณต่อสัญญาณรบกวน (SNR) เนื่องจากการฝังลายน้ำด้วยวิธีนี้ผู้ฟังแทบไม่รู้ถึงความผิดเพี้ยนของสัญญาณเสียงที่เกิดจากการเลื่อนเฟสที่ราบรื่น แม้ว่าเทคนิคนี้จะนำมาซึ่งการเปลี่ยนแปลงแบบแผนของเสียงทำให้องค์ประกอบในแต่ละความถี่เปลี่ยนไป แต่ก็ไม่ทำให้ผู้ฟังรับรู้ถึงความแตกต่างจากสัญญาณเสียงเดิม ตราบใดที่กระบวนการยังอยู่ขอบเขตที่กำหนด ซึ่งวิธีการนี้ยังมีข้อเสียอยู่ที่ความทนทานต่อการบีบอัดสัญญาณ, Spread spectrum เป็นเทคนิคที่ซ่อนข้อมูลลายน้ำโดยการกระจายข้อมูลลายน้ำลงบนสเปกตรัมความถี่ของสัญญาณเสียงต้นฉบับให้มากที่สุดเท่าที่จะเป็นไปได้ โดยมอดูเลชันสัญญาณเสียงต้นฉบับกับลำดับของพัลส์สุ่มแบบไบนารี “1 หรือ -1” ซึ่งบรรจุข้อมูลลายน้ำอยู่ วิธีการดังกล่าวเป็นการเปลี่ยนแปลงเพียงเล็กน้อยเมื่อเทียบกับบิตจำนวนมากของข้อมูล ขั้นตอนการฝังลายน้ำเริ่มจากลายน้ำแบบไบนารีจะถูกเข้ารหัสแก้ไขความผิดพลาด เพื่อลดความผิดพลาดที่เกิดขึ้นในขั้นตอนการฝัง แล้วนำมาคูณเข้ากับคลื่นพาห์ที่ได้จาก pseudo-random noise sequence จากขั้นตอนนี้จะได้ข้อมูลที่มีสเปกตรัมความถี่กว้างตามต้องการ แล้วทำการลดระดับความเข้มของข้อมูลลายน้ำลง ก่อนรวมเข้ากับสัญญาณเสียงต้นฉบับคล้ายกับการเกิดสัญญาณรบกวน วิธีนี้ได้ผลเป็นที่น่าพอใจเมื่อนำมาใช้ในการฝังลายน้ำลงบนสัญญาณเสียง สัญญาณลายน้ำที่ได้มีความคงทนต่อการโจมตีทางการประมวลผลสัญญาณแบบต่างๆ เช่น การบีบอัด และ cropping, Echo hiding ใช้เทคนิคการสร้าง echo บนสัญญาณเสียง โดยสัญญาณเสียงจะถูกแบ่งออกเป็นช่วงให้เท่ากับจำนวนข้อมูลลายน้ำ สัญญาณเสียงแต่ละช่วงจะถูกทำ echo ซึ่งในการฝังลายน้ำกำหนดให้การประวิงเวลามีขนาด d สำหรับการฝังบิตที่มีค่า 1 และมีขนาด d' สำหรับการฝังบิตที่มีค่า 0 ซึ่งขนาด d และ d' ต้องมีขนาดเหมาะสมโดย

ไม่ทำให้สัญญาณเสียงฟังเพี้ยนไปจากเดิม ขั้นตอนการตรวจจับลายน้ำจะนำสัญญาณที่ต้องการมาวิเคราะห์ cepstrum เพื่อหาตำแหน่งการเกิด echo จากนั้นจะทำการวัดขนาดการประวิงเวลาเพื่อหา d และ d' ต่อไป ซึ่งเทคนิคการฝังลายน้ำแบบ echo hiding นี้มีความทนทานต่อการประมวลผลสัญญาณและการบีบอัด แต่อย่างไรก็ตามเทคนิคนี้สามารถตรวจจับลายน้ำได้ง่ายจากบุคคลที่สามเช่นกัน

Jiwu Huang, Young Wang and Yun Q. Shi (2002) เสนอเทคนิคการฝังลงบนสัมประสิทธิ์ของการแปลงโคไซน์แบบเต็มหน่วย (DCT) โดยประยุกต์ใช้ synchronization code กับข้อมูลลายน้ำก่อน โดยเลือกสัมประสิทธิ์ของ DCT ที่ระดับความถี่ต่ำเพื่อความทนทานต่อการโจมตี ข้อดีของ synchronization code เมื่อรวมกับข้อมูลลายน้ำ คือสามารถตรวจจับตำแหน่งลายน้ำจากสัญญาณเสียงได้ กระบวนการนี้เลือกใช้การแปลง DCT ทำให้ลดเวลาในการตรวจจับลายน้ำลงมากเมื่อเปรียบเทียบกับแปลงแบบอื่นๆ กระบวนการเริ่มต้นด้วยการนำสัญญาณเสียงต้นฉบับแบ่งออกเป็นช่วงๆ ให้เท่ากับจำนวนข้อมูลลายน้ำ แล้วทำการแปลง DCT ในทุกๆ ช่วงของเสียงที่แบ่งแล้ว จากนั้นนำ synchronization code มาสร้างลำดับร่วมกับข้อมูลลายน้ำก่อน แล้วนำมาคูณค่าความหนาแน่นในการฝังก่อนรวมกับสัมประสิทธิ์ของ DCT ในแต่ละช่วงดังกล่าว โดยการเลือกขนาดช่วงในการแบ่งสัญญาณเสียง ต้องสอดคล้องกับจำนวนชุดของข้อมูลลายน้ำที่ต้องการฝังและจำนวนสัมประสิทธิ์ของ DCT ในขั้นตอนการตรวจจับข้อมูลลายน้ำ สัญญาณเสียงที่ต้องการตรวจจับจะถูกแบ่งออกเป็นช่วงๆ จากนั้นจะทำการแปลง DCT เพื่อตรวจจับข้อมูลลายน้ำ ซึ่งการรู้ค่าที่ถูกต้องของขนาดความยาวช่วงที่เหมาะสมจะได้มาซึ่งข้อมูลลายน้ำที่ถูกต้องต้องรู้ความยาวของข้อมูลลายน้ำกับ synchronization code ที่ใส่ลงไปเป็นสำคัญ สัญญาณเสียงที่ได้จากกระบวนการนี้มีความคงทนต่อการโจมตีในระดับที่น่าพอใจ ในงานวิจัยนี้มีจุดเด่นที่น่าสนใจ คือการเลือกค่าความหนาแน่นที่แตกต่างกันเพื่อความเหมาะสมกับการฝังลายน้ำในแต่ละเพลง แต่ค่าที่เลือกนี้ยังขึ้นกับประสบการณ์ที่ได้จากการทดลองหลายครั้งอยู่

Xueyao Li, Min Zhang and Shiliang Sun, (2003) ใช้ลายน้ำที่เป็นภาพขาวดำ ฝังลงบนสัญญาณเสียงโดยใช้การแปลงเวฟเลตแบบดิสครีต (DWT) ข้อดีของการใช้ลายน้ำที่เป็นภาพคือสามารถระบุความเป็นเจ้าของได้อย่างเด่นชัดโดยมนุษย์ อีกทั้งยังสามารถนำเครื่องหมายการค้า ตราบริษัท มาใช้เป็นลายน้ำได้ ในอัลกอริทึมใช้ SNR เป็นตัวกำหนดค่าความเข้มในการฝังลายน้ำลงบนสัญญาณเสียง ซึ่งตั้งสมมุติฐานไว้ว่าผลที่ได้ดีต้องดีกว่าการใช้ผลการทดลองหลายๆ ครั้งมาเป็นตัวกำหนดค่าความเข้มในการฝังลายน้ำ ซึ่งมีข้อเสียคือต้องใช้ข้อมูลที่ได้จากการทดลองจำนวนมาก และค่าความเข้มดังกล่าวนี้ ต้องมีค่าเปลี่ยนไปตามคุณลักษณะที่ไม่เหมือนกันของสัญญาณเสียงแต่ละตัวอย่าง กระบวนการฝังลายน้ำเริ่มจากนำสัญญาณเสียงมาแบ่งออกเป็นช่วงทั้งหมด k ช่วง ให้เท่ากับจำนวนข้อมูลภาพลายน้ำ ซึ่งแต่ละช่วงจะถูกนำมาแยกองค์ประกอบเวฟเลต 3 ระดับ ในงานวิจัย

นี้ได้เลือกสัมประสิทธิ์เวฟเล็ตใน detail subband จากการแปลงเวฟเล็ตลำดับที่สาม แทนด้วย cD_k^3 สัมประสิทธิ์เหล่านี้จะถูกฝังลงข้อมูลที่ผ่านการเพิ่มความปลอดภัยในการเข้าถึงข้อมูล โดยการเข้ารหัส การฝังลงข้อมูลจะฝังข้อมูลเพียงหนึ่งค่า (0 หรือ 1) ลงในสัมประสิทธิ์เวฟเล็ตที่มีค่าสูงที่สุดในช่วง cD_k^3 นั้นๆ แต่การฝังข้อมูลลงบนสัมประสิทธิ์เวฟเล็ตโดยตรงจะทำให้สัญญาณเสียงที่ได้เพี้ยนไปจากเดิมจนสามารถรับฟังได้ ดังนั้นการฝังจึงต้องควบคุมค่าความเข้มในการฝังเข้ากับข้อมูลลง น้ำเสียงก่อน เพื่อทำการสังเกตขนาดของข้อมูลลงน้ำก่อนที่จะฝัง ส่งผลให้เมื่อฝังลงน้ำแล้วไม่สามารถรับรู้ความแตกต่างที่เกิดขึ้นกับสัญญาณเสียงได้ ค่าความหนาแน่นนี้คำนวณได้จากค่า SNR ระหว่างสัมประสิทธิ์ของสัญญาณเสียงเดิม กับสัมประสิทธิ์ของสัญญาณเสียงที่ฝังลงน้ำแล้วโดย SNR ต้องมีค่าไม่เกินที่กำหนดไว้ เมื่อทำงานครบทุกช่วงสัญญาณทั้งจำนวน k ช่วง แล้วนำค่าสัมประสิทธิ์ดังกล่าวมาทำการรวมกลับเวฟเล็ตจะได้สัญญาณเสียงที่ฝังลงน้ำสมบูรณ์

ผลการทดลองแสดงว่าอัลกอริทึมนี้ ทนทานการโจมตีโดยการประมวลผลสัญญาณแบบต่างๆ และการบีบอัดแบบ MP3 ที่ 56 kbps ได้ดี ค่าสหสัมพันธ์ทั้งหมดของลายน้ำที่ได้จากสัญญาณเสียงที่ผ่านการโจมตีแบบต่างๆค่ามากกว่า 90% ผลการทดลองยังได้เปรียบเทียบระหว่างอัลกอริทึมที่ใช้ SNR กับผลการทดลองยังได้เปรียบเทียบระหว่างอัลกอริทึมที่ไม่ใช้ ซึ่งผลที่ได้สอดคล้องกับสมมุติฐานที่ตั้งไว้ ซึ่งชี้ให้เห็นถึงข้อดีของอัลกอริทึมที่ใช้ SNR แต่ข้อเสียของกระบวนการนี้คือการตรวจสอบลายน้ำจำเป็นต้องใช้สัญญาณเสียงต้นฉบับร่วมด้วย (Non-blind watermarking) ซึ่งเป็นการเสียอย่างมากในการเปิดเผยสัญญาณเสียงต้นฉบับ ทำให้อาจสูญเสียข้อมูลให้กับผู้ละเมิดได้

Lili Cui, Shu-xun Wang and Tanfeng Sun (2003) เสนอวิธีการฝังลายน้ำข้อมูลภาพขาวดำลงบนสัญญาณเสียงโดยอาศัยหลักการของ Human Auditory System (HAS) ซึ่งเป็นการวิเคราะห์สเปกตรัมของสัญญาณเสียงในความถี่ย่านต่างๆ เทียบกับสเปกตรัมของสัญญาณเสียงที่หูมนุษย์สามารถรับฟังได้ในช่วงความถี่ 20Hz ถึง 20kHz ลายน้ำจะถูกฝังลงบนย่านความถี่ที่มีค่า sound pressure level (SPL) ต่ำกว่าระดับการได้ยินของมนุษย์ โดยใช้แฟกเตอร์ควบคุมค่าหนึ่งในกระบวนการฝังลายน้ำจะใช้วิธีการวิเคราะห์ cepstrum ซึ่งมีนำมาประยุกต์ใช้ในการวิเคราะห์และจดจำเสียงพูดเป็นอย่างมาก กระบวนการเริ่มต้นโดยนำสัญญาณเสียง $x(n)$ มาแปลงฟูริเยร์แล้วทำลอการิทึม จากนั้นจึงทำการแปลงกลับฟูริเยร์อีกครั้ง ซึ่งการทำลอการิทึมในขั้นตอนที่สองเป็นลอการิทึมเชิงซ้อน เราเรียก $X(n)$ ที่ได้จากสมการ (2.1) ว่า complex cepstrum

$$X(n) = IFFT(\log(FFT((xn)))) \quad (2.1)$$

เราสามารถแปลงจากโดเมน cepstrum สูโดเมนเวลาได้ดังสมการ (2.2)

$$x(n) = \text{IFFT}(\exp(\text{FFT}(X(n)))) \quad (2.2)$$

ขั้นตอนการเตรียมลายน้ำเริ่มจากการนำสัญญาณเสียงมาแบ่งเป็นช่วง แล้วแปลงฟูริเยร์แบบเร็ว (FFT) สเปกตรัมที่ได้จะถูกนำไปเทียบกับสเปกตรัมการได้ยินของมนุษย์ ช่วงสเปกตรัมที่ต่ำกว่าระดับการได้ยินของมนุษย์มากที่สุดจะถูกเลือก เพื่อป้องกันความผิดเพี้ยนของสัญญาณเสียงหลังฝังลายน้ำ แล้วจึงแปลงกลับฟูริเยร์แบบเร็ว ผลลัพธ์จะได้แฟคเตอร์ควบคุมซึ่งต้องทำการสเกลให้มีค่าเหมาะสมก่อน ด้วยค่าที่ได้จากประสบการณ์จากการทดลองหลายๆครั้ง ก่อนนำไปมอดูเลตกับสัญญาณลายน้ำก็จะได้สัญญาณลายน้ำที่พร้อมจะทำการฝัง ในขั้นตอนการฝังลายน้ำ สัญญาณเสียงต้นฉบับจะถูกแบ่งเป็นช่วงๆ ให้พอดีกับจำนวนข้อมูลของลายน้ำ แล้วนำสัญญาณแต่ละช่วงมาวิเคราะห์ cepstrum จากนั้นหาค่าเฉลี่ยของสัมประสิทธิ์ cepstrum ในแต่ละช่วง นำค่าเฉลี่ยดังกล่าวมาลบออกจากสัมประสิทธิ์ทุกตัวเพื่อปรับให้สัมประสิทธิ์ cepstrum ในช่วงนั้นมีค่าเฉลี่ยเป็นศูนย์ จากนั้นทำการรวมลายน้ำที่เตรียมไว้เข้ากับสัมประสิทธิ์เหล่านี้ แล้วทำการแปลงกลับ cepstrum ก็จะได้สัญญาณเสียงที่ฝังลายน้ำเรียบร้อยแล้ว ในขั้นตอนการตรวจจับลายน้ำจะทำการคำนวณค่าเฉลี่ยของสัมประสิทธิ์ cepstrum ในแต่ละช่วงเทียบกับระดับที่กำหนด เพื่อระบุว่าลายน้ำที่ตรวจจับได้มีค่า 0 หรือ 1

กระบวนการดังกล่าวเป็นการฝังลายน้ำแบบปิด (Blind watermarking) ยังคงอาศัยการเลือกค่าแฟคเตอร์ควบคุมที่ได้จากประสบการณ์จากการทดลองหลายๆครั้ง และไม่ได้แสดงผลความทนทานต่อการโจมตีด้วย low pass filter เนื่องจากอัลกอริทึมมีโอกาสฝังลายน้ำลงบนช่วงความถี่ต่ำได้ Wei Li, Xiangyang Xue, Xiaoqiang Li and Peizhong Lu (2003) ทำการฝังลายน้ำแบบปิดที่ไม่ต้องใช้สัญญาณเสียงต้นฉบับร่วมกับการใช้รหัสแก้ไขความผิดพลาด ซึ่งข้อดีของการทำรหัสแก้ไขความผิดพลาด คือสามารถกู้คืนข้อมูลที่เสียหายให้กลับมาสมบูรณ์ดังเดิมได้ภายใต้เงื่อนไขของรหัสที่ใช้ เมื่อนำมาประยุกต์ใช้กับลายน้ำ ทำให้ลายน้ำที่เสียหายจากการโจมตีมีโอกาสคืนสู่สภาพเดิมได้ ซึ่งในขั้นตอนการฝัง สัญญาณเสียงจะถูกแบ่งออกเป็นช่วงๆ จากนั้นทำการแปลงเวฟเลตกับสัญญาณเสียงแต่ละช่วง โดยเลือกสัมประสิทธิ์เวฟเลตใน approximation subband จากการแปลงเวฟเลตลำดับที่ห้า แทนด้วย cA^5 ที่ระดับสัญญาณนี้ทำการคำนวณค่าเฉลี่ยของสัมประสิทธิ์ทั้งหมดในช่วง แล้วทำการลบค่าเฉลี่ยดังกล่าวออกจากสัมประสิทธิ์ทั้งช่วง ในขั้นตอนนี้จะได้ช่วงสัมประสิทธิ์ที่มีค่าเฉลี่ยเป็นศูนย์ จากนั้นนำข้อมูลลายน้ำที่ผ่านการเข้ารหัสรหัสแก้ไขความผิดพลาดแล้วมาทำการฝังลายน้ำเพียงหนึ่งค่าต่อหนึ่งช่วง โดยถ้าลายน้ำมีค่า '-1' จะทำการลบ สัมประสิทธิ์ทุกตัวในช่วงนั้นๆออกด้วย Δ หรือถ้าลายน้ำมีค่า '1' จะทำการบวก Δ เข้ากับสัมประสิทธิ์ทุกตัวในช่วงนั้นๆ ค่าแฟคเตอร์ Δ ที่ใช้ในการฝังลายน้ำหาได้จากการสเกลค่าเฉลี่ยของสัมประสิทธิ์ที่คำนวณได้ในช่วงนั้นๆก่อนหน้านี้ด้วยค่าที่ได้จากประสบการณ์การทดลอง ในขั้นตอนการตรวจจับลาย

น้ำทำได้ง่ายเพียงแปลงเวฟเล็ดที่ระดับ cA' จากนั้นทำการคำนวณหาค่าเฉลี่ยสัมประสิทธิ์เวฟเล็ดในช่วงดังกล่าว ซึ่งค่าเฉลี่ยสัมประสิทธิ์ที่ได้หากมากกว่าศูนย์แสดงว่าช่วงดังกล่าวฟังलयน้ำที่มีค่า '1' หากน้อยกว่าศูนย์แสดงว่าช่วงดังกล่าวฟังलयน้ำที่มีค่า '-1'

ผลการทดลองแสดงข้อดีของการประยุกต์ใช้รหัสแก้ไขความผิดพลาด ทำให้ข้อมูลลายน้ำที่ได้รับความเสียหายจากการโจมตีสามารถกู้คืนให้ใกล้เคียงกับของเดิมได้ แต่เวลาที่ใช้ในอัลกอริทึมนี้ค่อนข้างมาก และยังคงใช้ค่าแฟคเตอร์ Δ ในการควบคุมน้ำหนักของการฟังलयน้ำจากประสบการณ์การทดลองหลายๆครั้ง

Lili Cui, Shu-xun Wang and Tanfeng Sun (2003) ฟังข้อมูลลายน้ำเป็นลำดับตัวเลข 0 และ 1 ภายใต้หลักการบีบอัดสัญญาณเสียง ที่เปลี่ยนแปลงข้อมูลเพียงบางส่วนของสัญญาณเสียงโดยไม่ทำให้คุณภาพเสียงต่ำกว่ากำหนด ในงานวิจัยนี้ได้กำหนดให้คุณภาพเสียงหลังจากการฟังलयน้ำมีความแตกต่างระหว่าง ค่าต่ำสุดของ hearing threshold กับค่าสูงสุดของ error power spectrum ไม่เกิน 1 เดซิเบล ซึ่งโดยปกติการรับรู้ของประสาทมนุษย์จะตอบสนองต่อสัญญาณเสียงที่ความถี่ต่ำได้ดีกว่าสัญญาณเสียงที่ความถี่สูงส่งผลให้ระดับของการได้ยินจะอยู่สูงในช่วงความถี่สูง การวิเคราะห์ดังกล่าวจึงจำเป็นในการเลือกความถี่จากการแปลงเวฟเล็ด ที่มีความสำคัญต่อสัญญาณเสียงน้อย การฟังलयน้ำเริ่มจากสัญญาณเสียงถูกแยกองค์ประกอบ 4 ระดับด้วยเวฟเล็ด แล้วเลือกสัมประสิทธิ์เวฟเล็ดใน detail subbands ที่ระดับ 1 ถึง 3 แทนด้วย cD^1, cD^2, cD^3 ตามลำดับ กระบวนการเลือกเฟ้นสัมประสิทธิ์เวฟเล็ดที่จะฟังलयน้ำ เริ่มจากตั้งค่าเริ่มต้นเป็นครึ่งหนึ่งของค่าสัมประสิทธิ์สูงสุดในระดับนั้นๆ จากนั้นนำค่าเริ่มต้นดังกล่าวมาเปรียบเทียบกับสัมประสิทธิ์ทุกตัวในระดับนั้นๆ หากค่าสัมประสิทธิ์ตัวใดมีค่ามากกว่าค่าเริ่มต้นนี้ จะถูกเลือกเก็บเอาไว้จนได้จำนวนเท่ากับข้อมูลลายน้ำ หากได้ไม่ครบจำนวนข้อมูลลายน้ำ จะลดค่าเริ่มต้นนี้ลงอีกครั้งหนึ่งแล้วทำการค้นหาซ้ำจนกว่าจะได้จำนวนครบตามต้องการ สัมประสิทธิ์ที่ได้เหล่านี้จะถูกฟังलयน้ำโดยคูณกับค่าแฟคเตอร์ควบคุมที่ได้จากประสบการณ์การทดลองหลายๆครั้งอีกทีหนึ่ง กระบวนการนี้หลีกเลี่ยงการฟังलयน้ำบนสัมประสิทธิ์เวฟเล็ดใน detail subband ที่ระดับความถี่สูงสุด หรือ cD^0 และที่ความถี่ต่ำสุดใน approximation subband หรือ cA^4 โดยให้ความเห็นว่าที่ความถี่ดังกล่าวสามารถถูกโจมตีโดย การประมวลผลสัญญาณและการบีบอัดได้ง่าย ผลของการฟังलयน้ำด้วยวิธีนี้สามารถทนต่อการโจมตีแบบต่างๆได้ดีแต่ต้องใช้สัญญาณเสียงต้นฉบับในการตรวจจับลายน้ำ ผลทดลองได้ทดสอบกับเวฟเล็ดตระกูล Daubechies (db) หลายแบบและได้ผลดีที่สุดกับ db8

Ronghui Tu and Jiying Zhao (2003) ทำการฟังलयน้ำแบบ semi-fragile กล่าวคือลายน้ำประเภทนี้สามารถทนทานต่อการโจมตีทางการประมวลผลสัญญาณเฉพาะแบบตามที่กำหนดไว้เท่านั้น โดยการควบคุมผ่านพารามิเตอร์ที่กำหนด กระบวนการเริ่มต้นโดยสัญญาณเสียงจะถูกแยกองค์ประกอบเวฟเล็ดออก L ระดับ ซึ่งจะทำการฟังलयน้ำบนสัมประสิทธิ์เวฟเล็ดในทุก L ระดับ ในแต่

ระดับจะทำการแบ่งสัมประสิทธิ์เวฟเล็ทออกเป็นช่วงๆ ให้พอดีกับข้อมูลลายน้ำ และจะฝังข้อมูลลายน้ำที่เหมือนกันเพียงหนึ่งค่าต่อหนึ่งช่วงสัมประสิทธิ์ โดยเลือกสัมประสิทธิ์เวฟเล็ทที่มีค่ามากที่สุดจำนวน 50 ค่าในแต่ละช่วงมาผ่านกระบวนการควอนไทเซชันเพื่อระบุว่าสัมประสิทธิ์ตัวใดจะถูกคัดเลือกให้ฝังลายน้ำ ในการฝังลายน้ำ ข้อมูลลายน้ำจะถูกคูณเข้ากับควอนไทเซชันพารามิเตอร์ก่อนเพื่อทำการสเกลขนาด ซึ่งในทางปฏิบัติค่าควอนไทเซชันพารามิเตอร์จะมีค่าต่างกันในแต่ละ L ระดับ ซึ่งจะมีค่าเท่าใดนั้นยังขึ้นอยู่กับการใช้ประสบการณ์การจากทดลองหลายๆครั้ง โดยขนาดของควอนไทเซชันพารามิเตอร์ที่เล็กเกินไปจะทำให้ลายน้ำไม่คงทนต่อการโจมตี แต่ถ้ามีขนาดใหญ่เกินไปจะทำให้สัญญาณเสียงที่ได้เกิดความผิดเพี้ยน การเลือกย่านความถี่ที่ได้จากการแปลงเวฟเล็ทก็เป็นสิ่งสำคัญ เนื่องจากการใช้เทคนิคควอนไทเซชันนี้จะให้ความทนทานของข้อมูลลายน้ำบนสัมประสิทธิ์เวฟเล็ทในระดับความถี่ระดับกลางได้ดีกว่า เมื่อเปรียบเทียบกับสัมประสิทธิ์เวฟเล็ทที่ระดับความถี่อื่น เนื่องจากระดับความถี่ต่ำเป็นข้อมูลสำคัญของสัญญาณเสียง การเปลี่ยนแปลงค่าสัมประสิทธิ์ในส่วนนี้จะส่งผลโดยตรงต่อคุณภาพของเสียง ทำให้สัญญาณเสียงเพี้ยนได้ง่ายในขณะที่เดียวกันสัมประสิทธิ์ที่ความถี่สูงจะถูกโจมตีได้ง่ายโดยการประมวลผลสัญญาณแบบต่างๆ ในการตรวจจับลายน้ำจากสัญญาณเสียงจะถูกแบ่งออกเป็นช่วงๆและในแต่ละช่วงทำการเลือกสัมประสิทธิ์เวฟเล็ทที่มีค่ามากที่สุดมาจำนวน 50 ค่า ซึ่งสัมประสิทธิ์กลุ่มดังกล่าว จะต้องให้ลายน้ำที่เหมือนกับต้นฉบับไม่เช่นนั้นจะแสดงว่ามีการละเมิดกับสัญญาณเสียงเกิดขึ้น เนื่องจากการฝังลายน้ำ 1 บิตต้องใช้สัมประสิทธิ์ที่มากถึง 50 ตัว ทำให้สัญญาณเสียงที่ใช้มีความยาวมาก ผลการทดลองชี้ให้เห็นว่าการเลือกใช้ควอนไทเซชันพารามิเตอร์เพียงค่าเดียว ให้ผลในสัญญาณเสียงแต่ละแบบไม่ดีนึก ซึ่งต้องใช้ผลการทดลองจำนวนมากเพื่อให้ได้ค่าควอนไทเซชันพารามิเตอร์ที่เหมาะสมต่อไป

บทที่ 3

ทฤษฎีเบื้องต้น

3.1 เวฟเล็ต

3.1.1 ประวัติความเป็นมาของเวฟเล็ต

นับตั้งแต่ Joseph Fourier ได้คิดค้นทฤษฎีสำหรับการวิเคราะห์สัญญาณขึ้นในช่วง ค.ศ. ที่ 19 ทฤษฎีการวิเคราะห์สัญญาณบนโดเมนความถี่ของเขาได้รับความนิยมอย่างมากมา ในและเป็นประโยชน์อย่างเห็นได้ชัดในหลายด้าน ซึ่งในยุคต่อมาได้มีการศึกษาทฤษฎีการวิเคราะห์สัญญาณเพิ่มอย่างรวดเร็ว ทำให้มีงานวิจัยจำนวนมากใช้การวิเคราะห์ความถี่ร่วมกับการวิเคราะห์สเกล(เวลา) ซึ่งได้ผลดีกับการวิเคราะห์สัญญาณที่มีการเปลี่ยนแปลงไม่คงที่ รวมถึงสัญญาณที่มีการรบกวนสูง ซึ่งพบได้มากในทางปฏิบัติ ปัจจุบันเราเรียกทฤษฎีดังกล่าวว่า “เวฟเล็ต” ทฤษฎีเวฟเล็ตได้ถูกบันทึกเป็นครั้งแรกในปี ค.ศ. 1909 ในวิทยานิพนธ์ของ Alfred Haar ในวิทยานิพนธ์นี้ได้อ้างถึงการนำเสนอนิยามและทฤษฎีการวิเคราะห์สัญญาณ ที่คิดค้นขึ้นโดย Jean Morlet และทีมงานที่ Marseille Theoretical Physics Center ประเทศฝรั่งเศส ภายใต้การควบคุมของ Alex Grossmann

การวิเคราะห์เวฟเล็ตถูกพัฒนาอย่างมากโดย Y. Meyer และทีมนักวิจัย ซึ่งเป็นการวิเคราะห์ในรูปแบบใหม่ที่ยังไม่มีการเผยแพร่มาก่อน ซึ่งอาศัยการพัฒนาจากอัลกอริทึมหลักที่ได้จากงานวิจัยของ Stephane Mallat ในปี ค.ศ. 1988 นับแต่นั้นเป็นต้นมาเวฟเล็ตก็เป็นที่แพร่หลายโดยทั่วไป และถูกใช้งานอย่างกว้างขวาง โดยเฉพาะผลงานการวิจัยในประเทศอังกฤษจากนักวิทยาศาสตร์สำคัญ เช่น Ingrid Daubechies, Ronald Coifman และ Victor Wickerhauser (MATLAB, Computer program, 2002)

3.1.2 ภาพรวมของเวฟเล็ต

เมื่อพูดถึงคลื่นในทางคณิตศาสตร์ มักนึกถึงรูปแบบฟังก์ชันที่มีการเปลี่ยนแปลงอย่างต่อเนื่องในโดเมนเวลา การแปลงฟูริเยร์ก็เป็นการวิเคราะห์คลื่นแบบหนึ่งโดยการแปลงสัญญาณให้อยู่ในรูปของไซน์หรือโคไซน์ ซึ่งสามารถนำมาใช้ประโยชน์ในสาขาต่างๆเป็นอย่างมาก ส่วนเวฟเล็ต (Wavelet) ใช้แนวคิดการวิเคราะห์สัญญาณแบบหลายระดับความละเอียด (Multiresolution Analysis) โดยเวฟเล็ตเป็นคลื่นขนาดเล็กที่มีพลังงานจำกัดมีความเหมาะสมในการวิเคราะห์สัญญาณ

ที่เป็นภาวะชั่วคราว และมีการเปลี่ยนแปลงตามเวลา แตกต่างจากสัญญาณรูปขาคณิตที่ใช้กับการแปลงฟูริเยร์ที่มีพลังงานไม่จำกัด

เมื่อพิจารณาสัญญาณ $f(t)$ ให้อยู่ในรูปการแตกกระจายเชิงเส้นดังสมการ

$$f(t) = \sum_l a_l \psi_l(t) \quad (3.1)$$

เมื่อกำหนดให้ l เป็นจำนวนเต็มใดๆ a_l เป็น สัมประสิทธิ์จำนวนจริง และ $\psi_l(t)$ เป็นฟังก์ชันมูลฐานของ $f(t)$ ถ้ามูลฐานดังกล่าวมีคุณสมบัติเชิงตั้งฉาก (Orthogonal) กล่าวคือ

$$\langle \psi_k(t), \psi_l(t) \rangle = \int \psi_k(t) \psi_l(t) dt = 0 \quad k \neq l \quad (3.2)$$

เราจะสามารถคำนวณสัมประสิทธิ์ได้ตามสมการ

$$a_k = \langle f(t), \psi_k(t) \rangle = \int f(t) \psi_k(t) dt \quad (3.3)$$

สำหรับอนุกรมฟูริเยร์ ฟังก์ชันมูลฐานที่มีคุณสมบัติเชิงตั้งฉาก คือ ฟังก์ชันของ $\sin(k\omega_0 t)$ และ $\cos(k\omega_0 t)$ นั่นเอง

ฉะนั้นเมื่อกล่าวถึงเวฟเลตสามารถสรุปได้ถึงคุณสมบัติเด่นๆดังนี้

1) เวฟเลตเป็นเซตที่ใช้อธิบายสัญญาณ หรือฟังก์ชันที่มีหนึ่งมิติหรือสูงกว่าขึ้นไป ซึ่งถ้าเซตเวฟเลตคือ $\psi_{j,k}(t)$ โดยที่ $j, k = 1, 2, 3, \dots$ เราสามารถแสดงสัญญาณได้ในรูปของ

$$f(t) = \sum_k \sum_j a_{j,k} \psi_{j,k}(t) \quad (3.4)$$

2) เวฟเลตสามารถกระจายสัญญาณให้อยู่ในรูปของ time-frequency localization ซึ่งพลังงานของสัญญาณส่วนใหญ่จะสามารถแสดงอยู่ในรูปสัมประสิทธิ์การกระจาย $a_{j,k}$ เพียงไม่กี่ค่า ซึ่งเป็นผลดีอย่างมากในการบีบอัดสัญญาณ การลดสัญญาณรบกวน และการตรวจจับสัญญาณ

3) การคำนวณสัมประสิทธิ์เวฟเลตสำหรับการแปลงเวฟเลตหลายแบบใช้การคำนวณ $O(N)$ ครั้ง หมายความว่าจำนวนของการคำนวณจะเพิ่มขึ้นเป็นเชิงเส้นเมื่อสัญญาณมีความยาวมากขึ้น โดยทั่วไปการแปลงเวฟเลตจะใช้การคำนวณเทียบเท่ากับการแปลงฟูริเยร์แบบเร็ว (FFT)

จากสมการ (3.1) สังเกตเห็นว่าข้อมูลที่ได้จากการแปลงฟูริเยร์จะปรากฏตลอดย่านความถี่ $-\infty$ ถึง ∞ ดังนั้นเมื่อเกิดการเปลี่ยนแปลงของสัญญาณอินพุตใดๆ ในช่วงเวลาสั้นๆ จะส่งผลกระทบต่อตลอดย่านของสเปกตรัม $F(\omega)$ ด้วยเป็นการชี้ให้เห็นว่าการแปลงฟูริเยร์ให้ข้อมูลมากเกินไปเกินความจำเป็น นอกจากนี้สมการของการแปลงฟูริเยร์ไม่สามารถวิเคราะห์สัญญาณเฉพาะช่วงใดๆ ได้ เมื่อสัญญาณอินพุตมีการเปลี่ยนแปลงที่ไม่คงที่ (non-stationary signal) การแปลงฟูริเยร์จำเป็นต้องคำนวณใหม่ทำให้สิ้นเปลืองเวลาเป็นอย่างมาก ดังนั้นการแปลงฟูริเยร์จึงเหมาะสมกับสัญญาณที่มีการเปลี่ยนแปลงคงที่ (stationary signal)

จากข้อจำกัดที่เกิดขึ้นในการแปลงฟูริเยร์ทำให้มีการพัฒนาการแปลงกabor (Gabor transform) ซึ่งเป็นการแปลงฟูริเยร์ ที่สามารถกำหนดช่วงเวลาและความถี่ได้โดยอาศัยฟังก์ชันหน้าต่าง (window function) ที่เป็นฟังก์ชันเกาส์เซียน (Gaussian function) จากสมการที่ (3.1) จะเป็น

$$G_{a,b}(\omega) = \int_{-\infty}^{\infty} e^{-j\omega t} f(t) g_a(t-b) dt \quad (3.5)$$

โดยที่ฟังก์ชันเกาส์เซียน คือ

$$g_a(t) = \frac{1}{2\sqrt{a\pi}} e^{-\frac{t^2}{4a}} \quad (3.6)$$

โดยแปลงแบบนี้สามารถการเลือกความถี่ จากการกำหนดพารามิเตอร์ a และสามารถเลือกตำแหน่งการวิเคราะห์สัญญาณได้ โดยกำหนดพารามิเตอร์ b แต่ฟังก์ชันหน้าต่างของการแปลงกabor เป็นฟังก์ชันเกาส์เซียนแบบเดียวทำให้ไม่เหมาะกับการประยุกต์ใช้กับสัญญาณด้านต่างๆ ดังนั้นจึงมีการพัฒนาการแปลงฟูริเยร์ช่วงเวลาสั้น (STFT) ขึ้น ดังสมการที่ (3.7) ซึ่งสามารถเปลี่ยนฟังก์ชันหน้าต่างให้เหมาะสมกับสัญญาณอินพุตที่ต้องการวิเคราะห์ได้ดังนี้

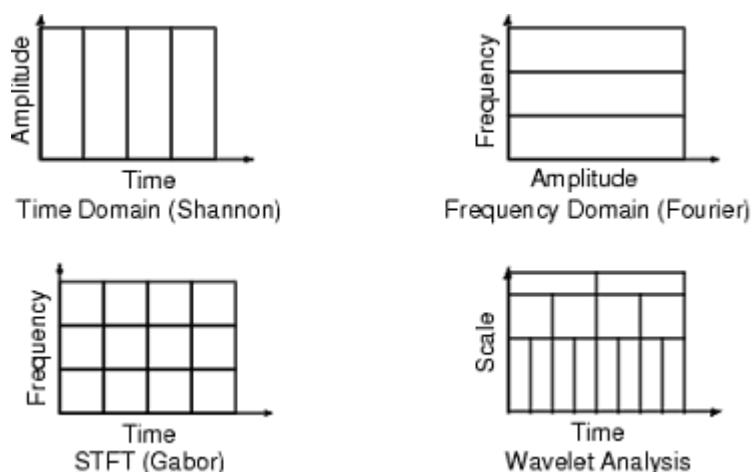
$$G_{a,b}(\omega) = \int_{-\infty}^{\infty} (e^{-j\omega t} f(t)) \overline{w(t-b)} dt \quad (3.7)$$

จะเห็นว่าการแปลงทั้งสองแบบดังกล่าวมีลักษณะช่วงของการวิเคราะห์ทั้งช่วงเวลาและช่วงความถี่ที่คงที่ (Fixed resolution transform) ทำให้ไม่เหมาะสมในทางปฏิบัติเนื่องจากที่ความถี่สูงจะมีช่วงเวลาแคบ ในขณะที่ความถี่ต่ำมีช่วงเวลาที่กว้างกว่าดังรูปที่ 3.1 ข้อมูลที่ได้จากการแปลงทั้งสองจึงมีลักษณะไม่สอดคล้องกับสัญญาณที่วิเคราะห์นั้น (MATLAB, Computer program, 2002)

จากข้อจำกัดดังกล่าวจึงมีการพัฒนาการแปลงเวฟเล็ตโดยฟังก์ชันหน้าต่างสามารถเลื่อนตำแหน่ง และปรับเปลี่ยนความกว้างของช่วงเวลาให้เหมาะสมกับความถี่ของสัญญาณที่จะนำมาวิเคราะห์ได้ โดยช่วงความถี่สูงจะใช้ช่วงเวลาที่สั้น ในขณะที่ความถี่ต่ำจะใช้ช่วงเวลาที่ทำการวิเคราะห์ยาวขึ้นซึ่งสามารถเขียนฟังก์ชันเวฟเล็ตได้เป็น

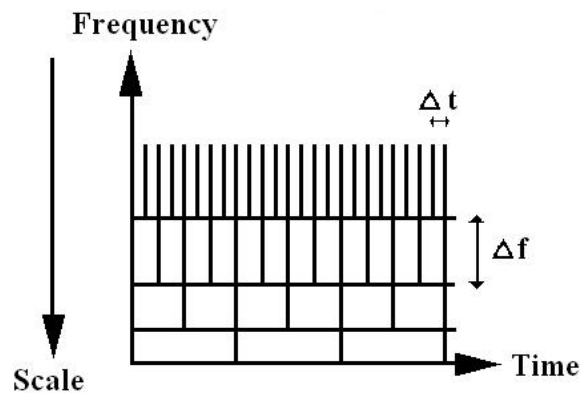
$$W(a,b) = \int_{-\infty}^{\infty} f(t) \frac{1}{\sqrt{a}} \psi^* \left(\frac{t-b}{a} \right) dt \quad (3.8)$$

เมื่อ $\psi(t)$ เป็นฟังก์ชันเวฟเล็ตแม่โดยการสเกลด้วยพารามิเตอร์ a และเลื่อนตำแหน่งโดยพารามิเตอร์ b ตามลำดับ



รูปที่ 3.1 ระยะเวลาและความถี่สำหรับการแปลงแบบต่างๆ (MATLAB, Computer program, 2002)

เมื่อพิจารณารูปที่ 3.2 ในช่วงเวลาความถี่สูงจะใช้ช่วงเวลาในการวิเคราะห์เพียงสั้นๆ ซึ่งจะให้รายละเอียดที่ดี ในขณะที่ช่วงความถี่ต่ำใช้ช่วงเวลาในการวิเคราะห์ที่ยาวกว่า อันเป็นการเหมาะสมกับลักษณะของสัญญาณในทางปฏิบัติ



รูปที่ 3.2 ระยะเวลาและความถี่สำหรับการแปลงเวฟเลต (MATLAB, Computer program, 2002)

นอกจากคุณสมบัติที่ดีของเวฟเลตในการวิเคราะห์สัญญาณดังกล่าวแล้ว เวฟเลตยังสามารถเลือกฟังก์ชันเวฟเลตหรือฟังก์ชันหน้าต่างให้เหมาะสมกับสัญญาณที่พบในทางปฏิบัติ อีกทั้งการคำนวณการแปลงเวฟเลตแบบเต็มหน่วย (DWT) ยังสอดคล้องกับการทำงานของการประมวลผลโดยคอมพิวเตอร์ เนื่องจากไม่มีแคลคูลัส การหาอนุพันธ์ หรือการอินทิเกรตเลย จะมีก็แต่เพียงการบวกและการคูณเท่านั้น

3.1.3 ปริภูมิของฟังก์ชัน

ปริภูมิในการประมวลผลสัญญาณคือ $L^2(R)$ ถ้าฟังก์ชันใดอยู่ในปริภูมินี้หมายความว่า $\int_R |f(t)|^2 dt < \infty$ เมื่อ R คือช่วงการอินทิเกรตที่เป็นจำนวนจริง สำหรับทุกๆ ฟังก์ชัน $f(t)$ ที่อยู่ในปริภูมิ S ใดๆ หรือ $f(t) \in S$ จะสามารถเขียนในรูปผลรวมเชิงเส้นตามสมการ

$$f(t) = \sum_k a_k \phi_k(t) \quad (3.9)$$

เราเรียกเซตของฟังก์ชัน $\phi(t)$ ว่าเซตการกระจาย (Expansion set) ของปริภูมิ S และจะเรียกเซตของฟังก์ชัน $\phi(t)$ ว่า ฟังก์ชันมูลฐาน (Basis function) ของปริภูมิ S เมื่อเซตของฟังก์ชัน $\phi(t)$ มีคุณสมบัติเชิงดังต่อไปนี้

$$\langle \phi_k(t), \phi_l(t) \rangle = \int_R \phi_k(t) \phi_l(t) dt = 0 \quad k \neq l \quad (3.10)$$

เราสามารถคำนวณสัมประสิทธิ์ได้ตามสมการ

$$a_k = \int_R f(t) \phi_k(t) dt \quad (3.11)$$

3.1.4 ฟังก์ชันสเกลลิง

ในการวิเคราะห์สัญญาณแบบหลายระดับความละเอียดจะเสนอฟังก์ชันมูลฐานอีกแบบหนึ่งนอกเหนือจากเวฟเลตเรียกว่า สเกลลิงฟังก์ชัน (Scaling function) หรือ $\phi(t)$ ซึ่งมีความจำเป็นอย่างมากในการอธิบายระดับความละเอียดของสัญญาณโดยนิยามดังสมการ (3.12)

$$\phi_k(t) = \phi_k(t - k) \quad (3.12)$$

เมื่อ k เป็นจำนวนเต็มใดๆ และ $\phi(t)$ เป็นเซตของฟังก์ชันมูลฐานในปริภูมิ L^2 ฉะนั้นปริภูมิใดๆ ที่มี $\phi(t)$ เป็นเซตของฟังก์ชันมูลฐานจะแทนด้วย V_0 กล่าวคือ

$$V_0 = \text{span}(\phi_k(t)) \quad (3.13)$$

เราสามารถปรับขนาดของ V_0 ได้โดยการเปลี่ยนขนาดของเวลา t ในสมการ(3.12) ออกเป็น

$$\phi_{j,k}(t) = 2^{j/2} \phi(2^j t - k) \quad (3.14)$$

ดังนั้นขนาดของ V_0 ที่ค่า j ต่างแทนโดย

$$V_j = \text{span}_k \{ \phi_k(2^j t) \} = \text{span}_k \{ \phi_{j,k}(t) \} \quad (3.15)$$

ฟังก์ชัน $f(t)$ ที่เกิดจากปริภูมิ V_j ใดๆสามารถเขียนในรูปของสมการ(3.16)

$$f(t) = \sum_k a_k \phi(2^j t + k) \quad (3.16)$$

จะเห็นว่าเมื่อ $j > 0$ สเกลลิงฟังก์ชัน $\phi_{j,k}(t)$ จะมีขนาดเล็กส่งผลให้การเลื่อน k ทำได้ในช่วงสั้นๆ ข้อมูลของสัญญาณ $f(t)$ ใดๆที่ได้จากปริภูมินี้จะมีข้อมูลที่ละเอียด ในขณะที่ $j < 0$ สเกลลิง

ฟังก์ชัน $\phi_{j,k}(t)$ จะมีขนาดใหญ่ช่วงการเลื่อนจะกว้าง ข้อมูลของสัญญาณ $f(t)$ ที่ได้จะเป็นข้อมูลที่หายาบ

3.1.5 ฟังก์ชันสเกลลิงกับการวิเคราะห์สัญญาณแบบหลายระดับความละเอียด

ก่อนอื่นเริ่มจากการนิยามปริภูมิย่อยที่มีคุณสมบัติการซ้อนทับกันดังสมการ (3.17)

$$\dots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \dots \subset L^2 \quad (3.17)$$

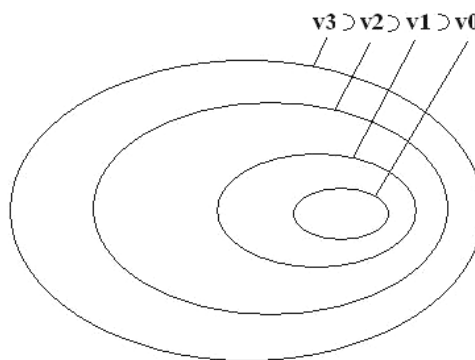
หรือสามารถกล่าวได้ว่า

$$V_j \subset V_{j+1} \quad j \in \mathbb{Z} \quad (3.18)$$

และ

$$V_{-\infty} = \{0\}, V_{\infty} = L^2 \quad (3.19)$$

จะกล่าวได้ว่าปริภูมิใดๆที่มีความละเอียดสูงกว่าจะมีปริภูมิที่มีความละเอียดต่ำกว่ารวมอยู่ด้วยดังแสดงในรูปที่ 3.3



รูปที่ 3.3 การซ้อนทับของปริภูมิย่อยที่แผ่โดยสเกลลิงฟังก์ชัน

จากการนิยามของ V_j ดังนั้น

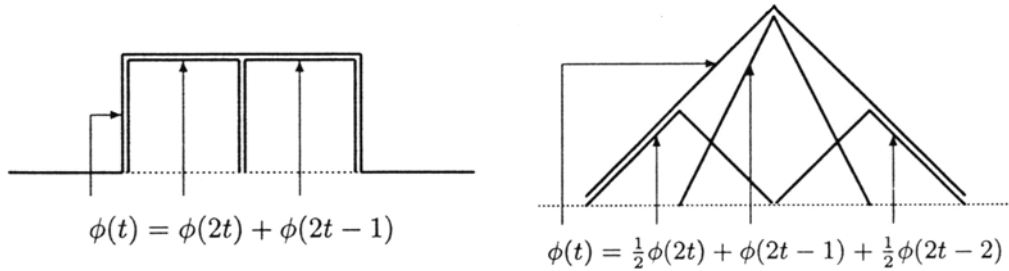
$$f(t) \in V_j \Leftrightarrow f(2t) \in V_{j+1} \quad (3.20)$$

จากคุณสมบัติตามสมการที่ (3.17) และ (3.20) หมายความว่า $\phi(t) \in V_1$ หรือ $\phi(t)$ อยู่ในปริภูมิย่อย V_0 ในขณะเดียวกันก็อยู่ในปริภูมิย่อย V_1 ด้วย จากความสัมพันธ์เขียนเป็นสมการได้คือ

$$\phi(t) = \sum_n h(n) \sqrt{2} \phi(2t - n), \quad n \in \mathbb{Z} \quad (3.21)$$

ค่าสัมประสิทธิ์ $h(n)$ คือลำดับของจำนวนจริงหรือจินตภาพเรียกว่าสัมประสิทธิ์ของฟังก์ชันสเกลลิง เรียกสมการที่ (3.21) นี้ว่าสมการการวิเคราะห์หลายระดับความละเอียดหรือสมการ Refinement

ตัวอย่างรูปที่ 3.4 แสดงสเกลลิงฟังก์ชันแบบ Haar ซึ่งแสดงให้เห็นว่า $\phi(2t)$ สามารถทำให้เกิด $\phi(t)$ ได้โดยการสเกลและการเลื่อนตำแหน่ง



รูปที่ 3.4 ฟังก์ชันสเกลลิงแบบ Haar (ซ้าย) และฟังก์ชันสเกลลิงแบบสามเหลี่ยม(ขวา)

(Burrus, Gopinath and Guo, 1998)

3.1.6 ฟังก์ชันเวฟเล็ต

นิยามฟังก์ชันเวฟเล็ต $\psi_{j,k}(t)$ เป็นเซตที่เกิดจากผลต่างระหว่างปริภูมิย่อย V_{j+1} กับ V_j จากสมการ (3.17)

$$V_0 \subset V_1 \subset V_2 \subset \dots \subset L^2 \quad (3.22)$$

นิยามให้เซตของฟังก์ชันเวฟเล็ตแผ่ทั่วปริภูมิย่อย W_0 ดังนั้น

$$V_1 = V_0 \oplus W_0 \quad (3.23)$$

$$V_2 = V_0 \oplus W_0 \oplus W_1 \quad (3.24)$$

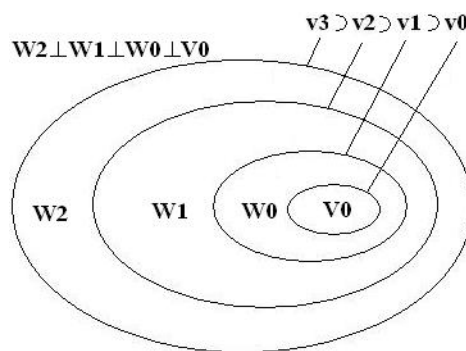
สามารถเขียนในรูปแบบทั่วไป

$$L^2 = V_0 \oplus W_0 \oplus W_1 \oplus \dots \quad (3.25)$$

การเริ่มต้นของปริภูมิสามารถเริ่มที่ปริภูมิใดๆ เช่น ที่ $j=10$

$$L^2 = V_{10} \oplus W_{10} \oplus W_{11} \oplus \dots \quad (3.26)$$

ความสัมพันธ์ดังกล่าวแสดงได้ดังรูปที่ 3.5



รูปที่ 3.5 ปริภูมิของฟังก์ชันสเกลลิ่งและฟังก์ชันเวฟเลต

ถ้า $j=-\infty$ จะทำให้ปริภูมิของฟังก์ชันสเกลลิ่งหายไป

$$L^2 = \dots \oplus W_{-2} \oplus W_{-1} \oplus W_0 \oplus W_1 \oplus W_2 \oplus \dots \quad (3.27)$$

จากสมการที่ (3.26) และ (3.27) จะได้ความสัมพันธ์ระหว่างปริภูมีย่อย V_0 และเวฟเลตคือ

$$V_0 = W_{-\infty} \oplus \dots \oplus W_{-1} \quad (3.28)$$

จากสมการ (3.23) ฟังก์ชันเวฟเลตสามารถเขียนในรูปผลรวมเชิงเส้นของฟังก์ชันสเกลลิ่งได้

$$\psi(t) = \sum_n h_1(n) \sqrt{2} \phi(2t-n) \quad n \in \mathbb{Z} \quad (3.29)$$

ฟังก์ชันที่ได้จากสมการที่ (3.29) เป็นเวฟเล็ดต้นแบบหรือเวฟเล็ดแม่ ซึ่งนิยามที่ความละเอียดใดๆ โดย

$$\psi_{j,k} = 2^{j/2} \psi(2^j t - k) \quad (3.30)$$

เมื่อ 2^j คือสเกลของ t , $2^{-j}k$ คือการเลื่อนของ t และ $2^{j/2}$ เป็นนอร์มของเวฟเล็ดที่สเกลต่างๆ ดังนั้นฟังก์ชัน $g(t)$ ใดๆที่อยู่ใน $L^2(\mathbb{R})$ สามารถเขียนสมการในรูปของฟังก์ชัน $\phi_{j,k}(t)$ และ $\psi_{j,k}(t)$ ที่แผ่ทั่วปริภูมิ $L^2(\mathbb{R})$ ได้ดังนี้

$$g(t) = \sum_{k=-\infty}^{\infty} c(k) \phi_k(t) + \sum_{j=0}^{\infty} \sum_{k=-\infty}^{\infty} d(j,k) \psi_{j,k}(t) \quad (3.31)$$

เทอมแรกของสมการที่ (3.31) เป็นส่วนประมาณของฟังก์ชันที่ได้จากฟังก์ชันสเกลลิง เทอมที่สองเป็นรายละเอียดของฟังก์ชันที่ได้จากฟังก์ชันเวฟเล็ด ในกรณีเป็นเวฟเล็ดออร์ทोगอนัลค่าสัมประสิทธิ์ $c(k)$ และ $d(j,k)$ คำนวณได้จากผลคูณภายในดังสมการ

$$c(k) = c_0(k) = \langle g(t), \phi(t) \rangle = \int g(t) \phi(t) dt \quad (3.32)$$

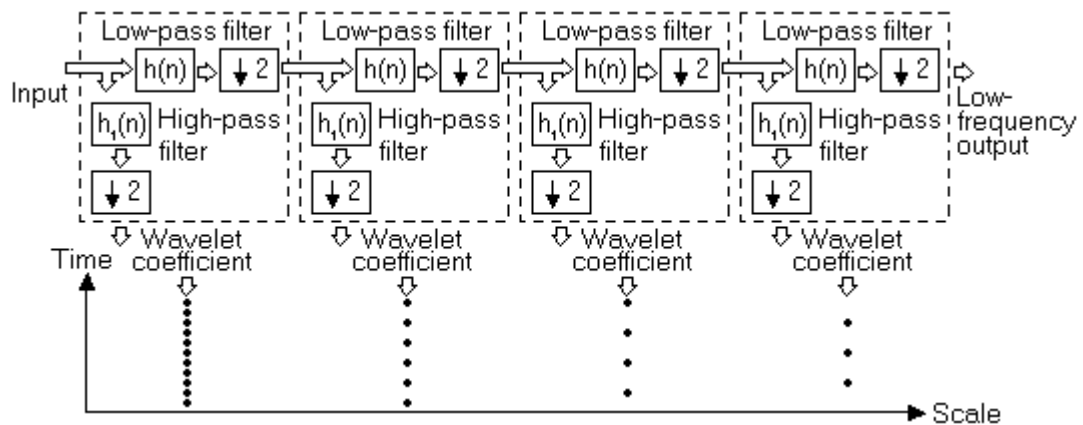
และ

$$d_j(k) = d(j,k) = \langle g(t), \psi_{j,k}(t) \rangle = \int g(t) \psi_{j,k}(t) dt \quad (3.33)$$

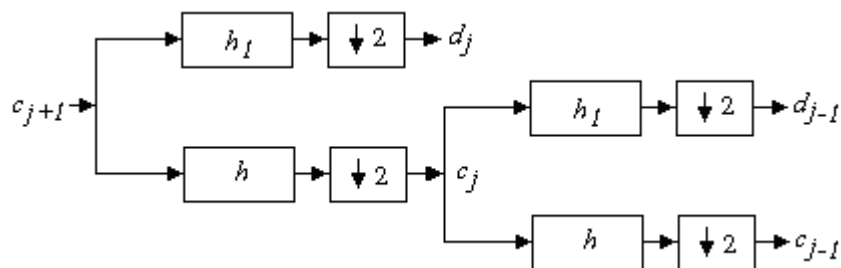
3.1.7 การแปลงเวฟเล็ดแบบเต็มหน่วย

การคำนวณหาค่าสัมประสิทธิ์ของเวฟเล็ดในทุกค่าของสเกลและความถี่ทำให้สูญเสียเวลาและปริมาณข้อมูลที่มีที่มากเกินไป หากพิจารณาการแปลงเวฟเล็ดที่มีลักษณะคล้ายกับขบวนการฟิลเตอร์แบงก์แบบสองช่องสัญญาณ (Two-channel filter banks) สัญญาณอินพุตถูกแบ่งออกเป็นสองส่วน ด้วยตัวกรองความถี่ต่ำ หรือ $h_l(k)$ และตัวกรองความถี่สูง หรือ $h_h(k)$ สัมประสิทธิ์ของตัวกรองนี้จะสัมพันธ์กับเวฟเล็ดแม่ ซึ่งค่าที่ได้จากตัวกรองความถี่สูงคือค่าสัมประสิทธิ์เวฟเล็ดนั่นเอง กระบวนการดังกล่าวแสดงดังรูปที่ 3.6 เอาท์พุทที่ได้จากตัวกรองความถี่ต่ำและผ่านการลดอัตราสุ่มลงสองเท่าแล้วจะถูกนำมาเป็นอินพุทให้กับส่วนต่อไป ดังนั้นสัญญาณหนึ่งๆสามารถแตกออกไปได้อีก ซึ่งจะได้ส่วนประกอบที่มีรายละเอียดที่ต่ำกว่า ค่าสัมประสิทธิ์เวฟเล็ดที่ได้แต่ละระดับจะมี

จำนวนลดลงเรื่อยๆ เราเรียกว่ากระบวนการนี้ว่าการแยกองค์ประกอบหลายระดับ (Multi-level decomposition) ซึ่งมีลักษณะ โครงสร้างแบบต้นไม้ (Tree structure) ดังรูปที่ 3.7 ลักษณะกระบวนการวิเคราะห์เหล่านี้เรียกว่า การแปลงเวฟเลตแบบเต็มหน่วย (DWT)



รูปที่ 3.6 กระบวนการแปลงเวฟเลตแบบเต็มหน่วย



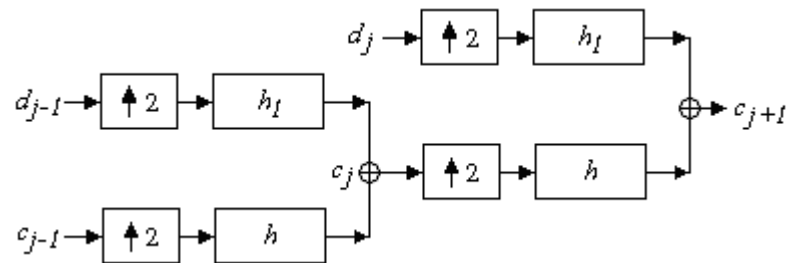
รูปที่ 3.7 กระบวนการแปลงเวฟเลตแบบเต็มหน่วย 2 ระดับ

ดังนั้นเราสามารถแตกกระจายสัญญาณออกสู่ระดับสเกลที่หยาบกว่าได้ดังสมการ

$$f(t) = \sum_k c_{j_0}(k) \phi_{j_0+k}(t) + \sum_k \sum_{j=j_0}^{j-1} d_j(k) \psi_{j,k}(t) \quad (3.34)$$

ในกระบวนการแปลงกลับเวฟเลต (IDWT) เพื่อสังเคราะห์ให้ได้สัญญาณอินพุตกลับคืนมา จะนำสัมประสิทธิ์เวฟเลตและสเกลลิงฟังก์ชันในระดับรายละเอียดที่หยาบมาสร้างสัญญาณในระดับที่รายละเอียดสูงกว่าดังรูปที่ 3.8 โดยเพิ่มอัตราสุ่มขึ้นสองเท่าดังสมการ

$$c_{j+1}(k) = \sum_m c_j(m)h(k-2m) + \sum_m d_j(m)h_1(k-2m) \quad (3.35)$$



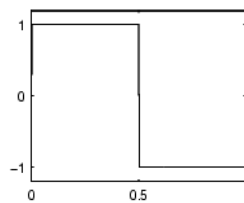
รูปที่ 3.8 กระบวนการแปลงเวฟเลตย้อนกลับ 2 ระดับ

3.1.8 ตระกูล Wavelet

เวฟเลตได้ถูกแบ่งออกเป็นตระกูลต่างๆมากมายและยังมีสมาชิกที่แตกต่างกันในแต่ละตระกูลอีกด้วย ดังจะยกตัวอย่างเวฟเลตในตระกูลต่างๆดังนี้(MATLAB, Computer program, 2002)

Haar wavelet

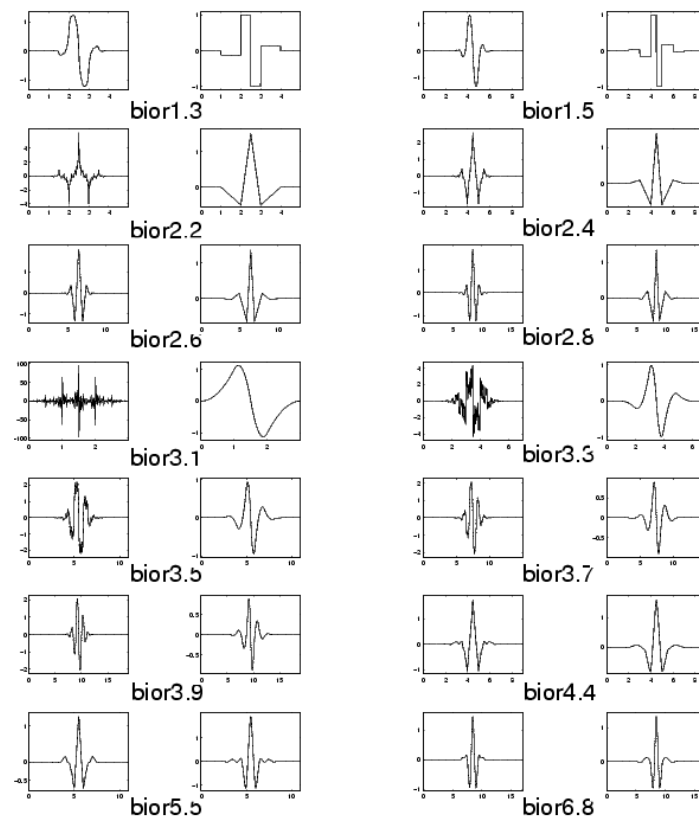
เวฟเลตหลายชนิดถูกพัฒนาขึ้นโดยอาศัยพื้นฐานของ เวฟเลตแบบ Haar ซึ่งเป็นเวฟเลตที่มีความไม่ต่อเนื่อง คล้ายกับฟังก์ชันขั้นบันได เวฟเลตแบบ Haar นี้เป็นตัวเดียวกับเวฟเลตแบบ Daubechies, db1



รูปที่ 3.9 เวฟเลตแบบ Haar

Biorthogonal wavelet

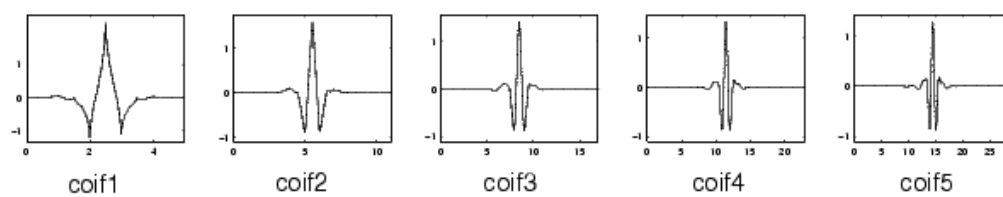
เวฟเลตตระกูลนี้มีคุณสมบัติแบบ linear phase ซึ่งเป็นคุณสมบัติที่สำคัญในการสร้างเวฟเลตตระกูลนี้แบ่งออกเป็นสองเวฟเลตย่อย สำหรับการแตกกระจายเวฟเลต (ฝั่งซ้าย) และการสร้างสัญญาณขึ้นใหม่ (ฝั่งขวา) แทนที่จะใช้เวฟเลตเพียงตัวเดียวอย่างเช่นตระกูลอื่นๆ



รูปที่ 3.10 เวฟเลตแบบ Biorthogonal

Coiflets wavelet

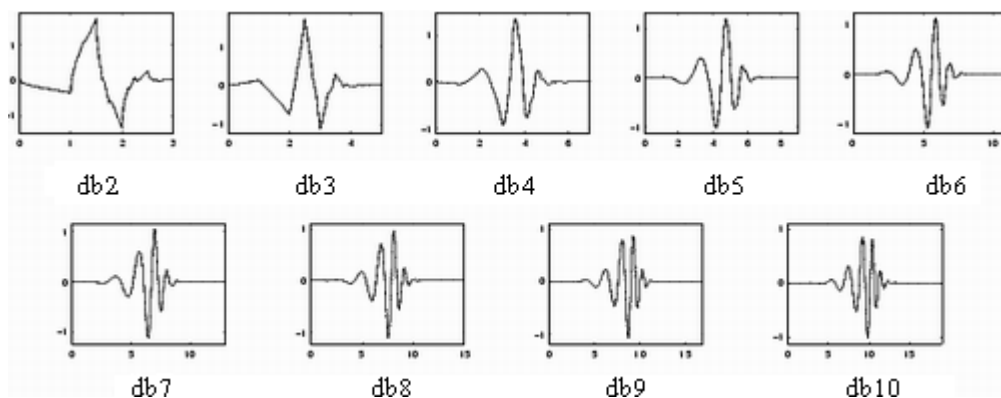
เวฟเลตตระกูลนี้ถูกพัฒนาโดย I. Daubechies โดยความต้องการของ R. Coifman



รูปที่ 3.11 เวฟเลตแบบ Coiflets

Daubechies wavelet

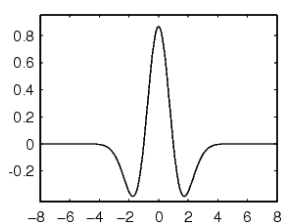
เวฟเลตตระกูลนี้กำเนิดโดย Ingrid Daubechies เป็นเวฟเลตแบบ orthonormal wavelets สมาชิกในเวฟเลตตระกูลนี้เขียนแทนด้วย dbN, โดยที่ N คือลำดับของเวฟเลต และนำหน้าด้วย db



รูปที่ 3.12 เวฟเลตแบบ Daubechies

Maxican Hat wavelet

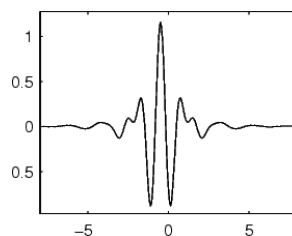
เวฟเลตชนิดนี้ได้มาจากการหาอนุพันธ์อันดับสองของ Gaussian probability density function เวฟเลตชนิดนี้มีแค่ตัวเดียวและไม่มีสเกลลิงฟังก์ชัน



รูปที่ 3.13 เวฟเลตแบบ Mexican Hat

Meyer wavelet

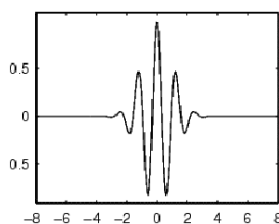
เวฟเลตชนิดนี้ถูกสร้างขึ้นในโดเมนความถี่มีแค่ตัวเดียวและไม่มีสเกลลิงฟังก์ชัน



รูปที่ 3.14 เวฟเลตแบบ Meyer

Morlet wavelet

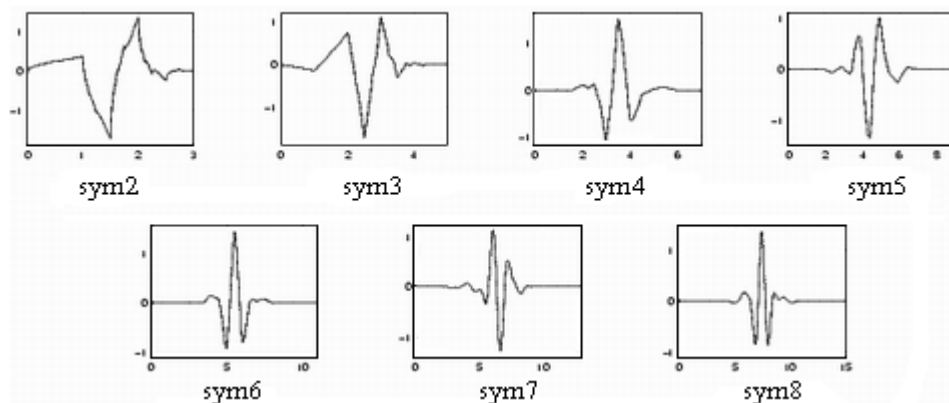
เวฟเลตชนิดนี้มีแค่ตัวเดียวและไม่มีสเกลลิงฟังก์ชัน



รูปที่ 3.15 เวฟเลตแบบ Morlet

Symlets wavelet

เวฟเลตตระกูล symlets เป็นเวฟเลตแบบเกือบสมมาตร ถูกพัฒนามาจากเวฟเลตตระกูล db โดย Ingrid Daubechies ซึ่งทำให้เวฟเลตทั้งสองตระกูลมีทฤษฎีพื้นฐานที่คล้ายกัน



รูปที่ 3.16 เวก์เลตแบบ Symlets

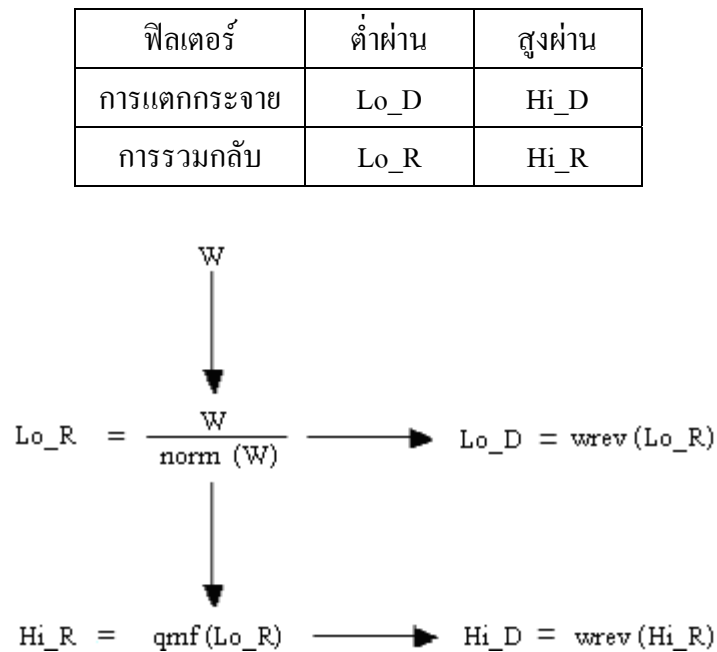
3.1.9 การแปลงเวก์เลตแบบเร็ว (FWT)

หลักการทํางานของ DWT อาศัยฟิลเตอร์แบงค์แบบสองช่องสัญญาณร่วมกับ quadrature mirror filters (QMF) ซึ่งกระบวนการดังกล่าวถูกคิดขึ้นโดย Mallat ในปี ค.ศ. 1988 ซึ่งพัฒนามาจากทฤษฎีพื้นฐานทางการประมวลผลสัญญาณสื่อสาร มีแนวคิดพื้นฐานคือ

- อัลกอริทึมสำหรับการแตกกระจายเวก์เลตออกเป็น j ระดับเริ่มจากสัญญาณ S ถูกแบ่งแบนด์ย่อย cA_j และ cD_j แล้วจากนั้น cA_j จะถูกแบ่งออกเป็น cA_{j-1} และ cD_{j-1} กระบวนการนี้จะซ้ำต่อไปอีกเรื่อยๆ ถึง cA_1 และ cD_1
- อัลกอริทึมสำหรับ DWT เริ่มจากนำ cA_1 และ cD_1 มาสร้างสัญญาณในระดับสูงขึ้น คือ cA_2 , แล้วนำมารวมกับ cD_2 จากนั้นนำมาซ้ำกระบวนการเดิมเพื่อสร้าง cA_j ต่อไป

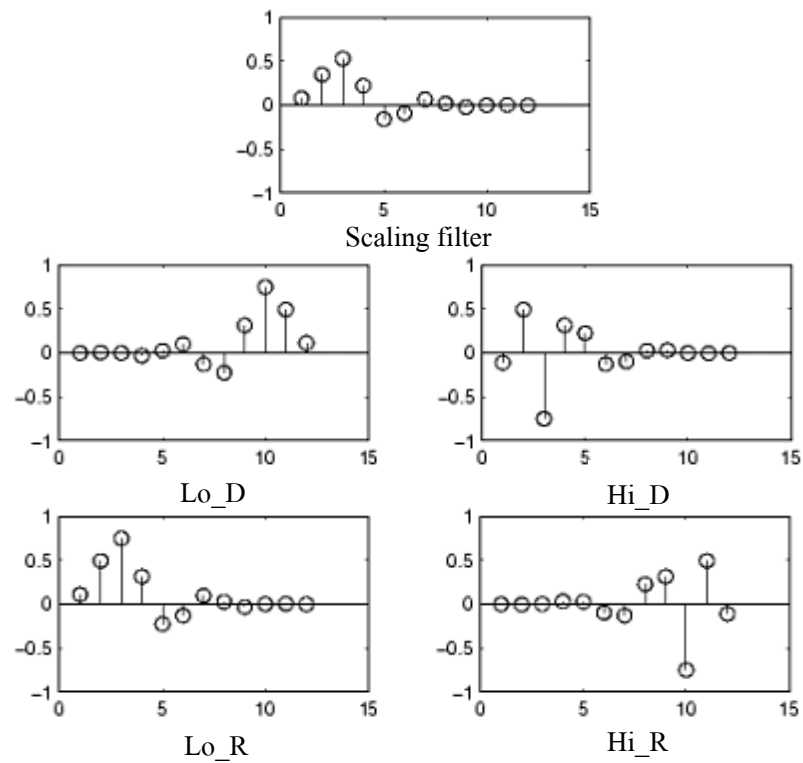
3.1.10 อัลกอริทึมการแปลงเวก์เลตแบบเต็มหน่วยหนึ่งมิติ

ในการวิเคราะห์แบบหลายระดับความละเอียดกับเวก์เลตแบบอโทโกนัล เริ่มจากการหา cA_j และ cD_j จากสัญญาณ S โดยจำเป็นต้องรู้สเกลลิงฟิลเตอร์แทนด้วย W ที่ได้จากสเกลลิงฟังก์ชันที่มีคุณสมบัติเป็น FIR, มีความยาว $2N$ โดย N คือลำดับของเวก์เลต, มีคุณสมบัติของวงจรกรองความถี่ต่ำผ่านเช่นตัวอย่างของ dbN โดย $N=3$ มีค่าสัมประสิทธิ์ $2N = 6$ ตัว ดังนี้ 0.2352 0.5706 0.3252 -0.0955 -0.0604 0.0249 เราสามารถสร้างฟิลเตอร์สำหรับการแตกกระจายและการรวมกลับได้ดังรูปที่ 3.17 และได้ผลดังรูปที่ 3.18

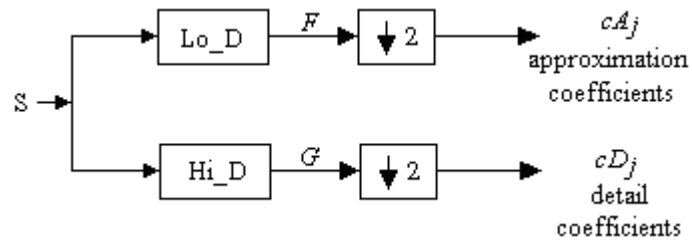


รูปที่ 3.17 การคำนวณหาฟิลเตอร์ทั้งสี่ตัวโดยที่ qmf คือ quadrature mirror filters, wrev คือ การกลับลำดับเวกเตอร์จากซ้ายเป็นขวา

สัญญาณที่มีความยาว N แซมเปิล จะถูกแบ่งออกเป็นสองส่วนย่อย คือ approximation coefficients หรือ cA_j ซึ่งได้จากการทำเซอร์คูลาคอนโวลูชัน ระหว่าง S กับวงจรกรองความถี่ต่ำผ่าน (Lo_D) และ detail coefficients หรือ cD_j ได้จากการทำเซอร์คูลาคอนโวลูชันระหว่าง S กับวงจรกรองความถี่สูงต่ำผ่าน (Hi_D) แล้วทำการลดอัตราสุ่มลงสองเท่าดังรูป 3.19

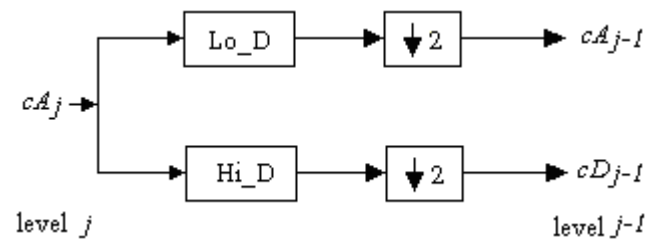


รูปที่ 3.18 ตัวอย่างฟิลเตอร์สำหรับการแปลงเวฟเลต db3



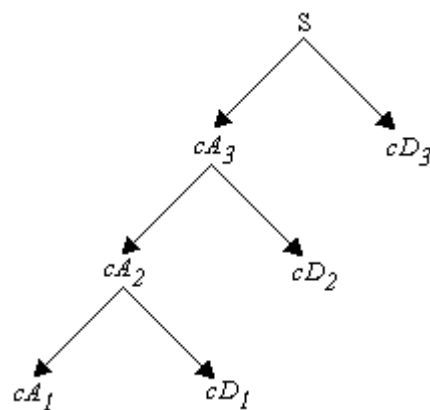
รูปที่ 3.19 การคำนวณ cA_j และ cD_j

ซึ่งสัญญาณ F และ G จะมีขนาด N ส่วน cA_j และ cD_j จะได้จากการลดอัตราสุ่ม F และ G ลงสองเท่าตามลำดับ ในการเพิ่มระดับความละเอียดของการวิเคราะห์สัญญาณทำได้โดยการซ้ำขั้นตอนเดิมที่กล่าวมาแล้วโดยแทน $S = cA_j$ ดังรูปที่ 3.20 กระบวนการนี้จะซ้ำขั้นตอนจนกว่าจะได้ระดับความละเอียดที่ต้องการ



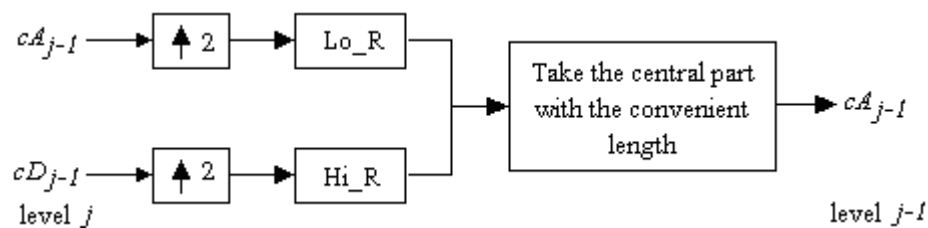
รูปที่ 3.20 การคำนวณ cA_{j-1} และ cD_{j-1}

ดังนั้นที่ $j = 3$ เราสามารถเขียนโครงสร้างแบบต้นไม้ได้ดังรูปที่ 3.21 โดยที่โครงสร้างของสัมประสิทธิ์คือ $[cA_1, cD_1, cD_2, cD_3]$



รูปที่ 3.21 โครงสร้างแบบต้นไม้ที่ $j=3$

สำหรับกระบวนการ IDWT สามารถทำได้โดยการย้อนกลับกระบวนการ DWT ตามที่กล่าวมาข้างต้น โดยเพิ่มศูนย์เข้าในขั้นตอนการเพิ่มอัตราสุมแบบสองเท่า แล้วทำเซอร์คิวลาคอนโวลูชันกับ Lo_R และ Hi_R จากนั้นทำการรวมสัญญาณได้ความยาวของสัญญาณเท่าเดิม



รูปที่ 3.22 กระบวนการรวมกลับสัญญาณ

3.2 การค้นหาแบบตามูเชิงปรับตัว

การค้นหาแบบตามูเชิงปรับตัว (ATS) (อาทิตย์ ศรีแก้ว, 2545) เป็นกระบวนการค้นหาคำตอบของสมการใดๆที่พัฒนาจากการค้นหาแบบตามู (TS) โดย TS ค้นหาจากตำแหน่งเริ่มต้นไปหาคำตอบที่ดีที่สุดที่เรียกว่าค่าเหมาะสมที่สุดในวงกว้าง (Global optimum) จุดเด่นของ TS อยู่ที่ tabu list ซึ่งจะบันทึกตำแหน่งการเดินในแต่ละครั้งไว้ สามารถนำมาใช้เป็นข้อมูลในการตัดสินใจเลือกตำแหน่งในการค้นหาใหม่ได้ในกรณีที่ไมเจอคำตอบเป็นเวลานาน หรือเจอคำตอบแบบค่าเหมาะสมที่สุดเฉพาะถิ่น (Local optimum) ลักษณะการค้นหาแบบตามูคือ สามารถตัดสินใจเลือกตำแหน่งในการค้นหาโดยอนุญาตให้เดินค้นหาในบางทิศทาง และไม่ให้เดินค้นหาในบางทิศทางหรือที่เรียกว่า Tabu ซึ่งกำหนดโดย Tabu list มีหน้าที่เก็บบันทึกข้อมูลทิศทางเดินที่ผ่านๆมาภายใต้เงื่อนไขในการตัดสินใจเลือกตำแหน่งแบบต่างๆ ซึ่งเงื่อนไขที่นิยมได้แก่ เงื่อนไขของความถี่ (Frequency) และเงื่อนไขของความคงอยู่ (Recency) ภายใต้แนวคิดที่ว่า “เมื่อเกิดการmoveขึ้นในทิศทางใดทิศทางหนึ่งแล้ว ไม่ควรจะต้องเดินทางย้อนกลับไปในทิศทางเดิมอีก เพราะดูเหมือนว่าคำตอบ ณ ตำแหน่งที่ผ่านมาจะไม่ใช่ว่าคำตอบที่ต้องการ” (อาทิตย์ ศรีแก้ว, 2545) โดยทั้งสองเงื่อนไขนี้มีความเกี่ยวข้องซึ่งกันและกัน หากใช้เพียงเงื่อนไขเดียวตามลำพัง การค้นหาคำตอบจะไม่สามารถทำงานได้อย่างเต็มที่

เงื่อนไขของความถี่ เป็นเงื่อนไขที่บอกว่าการเดินในทิศทางนั้นๆมีการใช้บ่อยเพียงใด ถ้ามีการใช้บ่อยจนถึงค่าที่ตั้งไว้ จะถูกตั้งให้เป็น Tabu คือไม่ให้เดินในทิศทางนั้นอีก

เงื่อนไขของความคงอยู่ เป็นเงื่อนไขที่สอดคล้องกับเงื่อนไขของความถี่ โดยการเดินในทิศทางใดที่ถูกตั้งไม่ให้เดินไว้เมื่อถึงภายในระยะเวลาที่กำหนด การเดินดังกล่าวจะยกเลิกความเป็น Tabu และสามารถเดินค้นหาได้อีก ซึ่งระยะเวลาดังกล่าวเป็นเงื่อนไขสำคัญในการออกแบบ TS

แนวคิดของ TS ถูกเสนอโดย Glover ในปี ค.ศ. 1977 ซึ่งเป็นแนวคิดในการแก้ไขปัญหาเกี่ยวกับ combinatorial optimization และเป็นที่ยอมรับแพร่หลายสืบมา ซึ่งข้อดีของ TS คือการที่สามารถหลีกเลี่ยงคำตอบแบบค่าเหมาะสมที่สุดเฉพาะถิ่น ได้ดี และสามารถทำการค้นหาคำตอบต่อไปจน

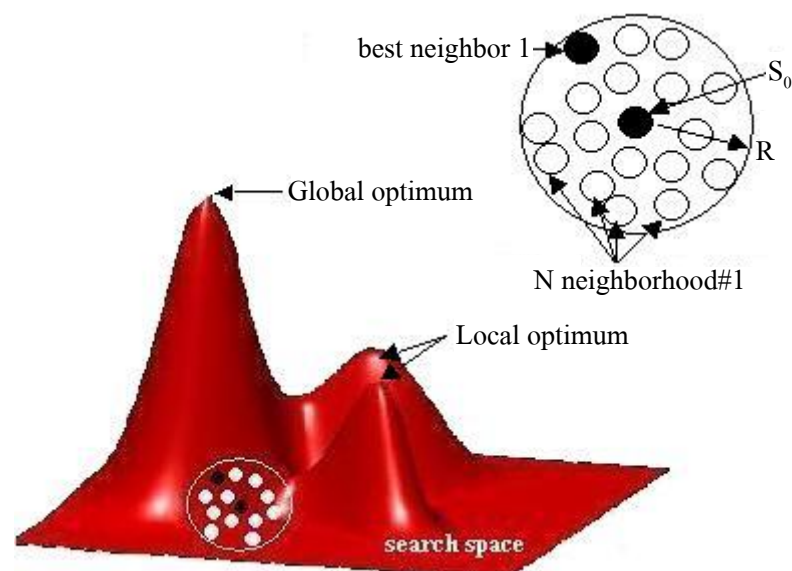
กระทั่งให้คำตอบที่ใกล้เคียงกับค่าเหมาะสมที่สุดในช่วงกว้าง ได้ อีกทั้งขั้นตอนการทำงานที่ไม่ยุ่งยากซับซ้อนเมื่อเทียบกับการค้นหาแบบอื่นๆ เช่น simulate annealing หรือ genetic algorithm (GA)

ขั้นตอนการค้นหาคำตอบแบบ TS สามารถสรุปโดยย่อได้ดังนี้

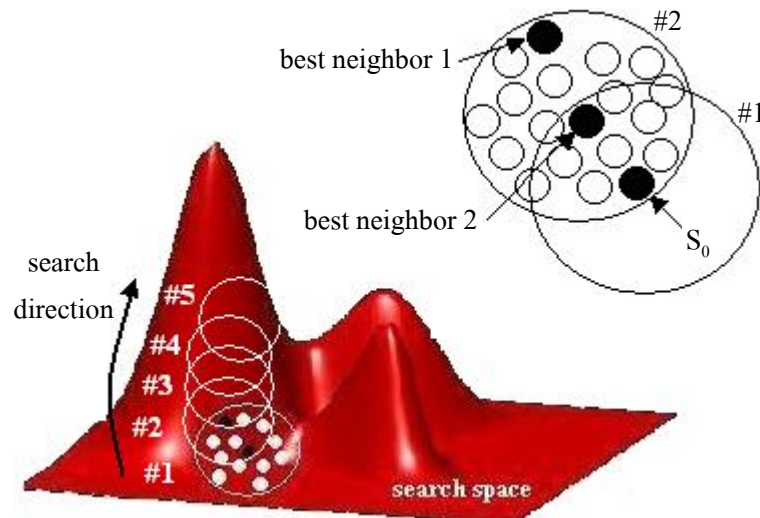
กำหนดให้รอบของการค้นหาคือ count และรอบของการค้นหาสูงสุดเท่ากับ max_count

1) ทำการเลือกสุ่มค่าเริ่มต้นของคำตอบ, S_0 ภายในปริภูมิ มีขนาดรัศมีของการค้นหาเท่ากับ R โดยกำหนดให้เสมือนว่า S_0 เป็นค่าเหมาะสมที่สุดเฉพาะถิ่น และตั้งค่า best_neighbor ให้เท่ากับ S_0

2) ในรัศมี R รอบ S_0 นี้ทำการสุ่มค่าขึ้นมา N ตำแหน่ง แล้วทำการเก็บค่าไว้ในเซต $S_1(r)$ โดยเรียกค่าทั้ง N ตำแหน่งในการสุ่มครั้งแรกนี้ว่า N neighborhood#1



รูปที่ 3.23 ขั้นตอนการเลือก best_neighbor1



รูปที่ 3.24 การค้นหาแบบ TS ที่ best_neighbor ต่างๆ

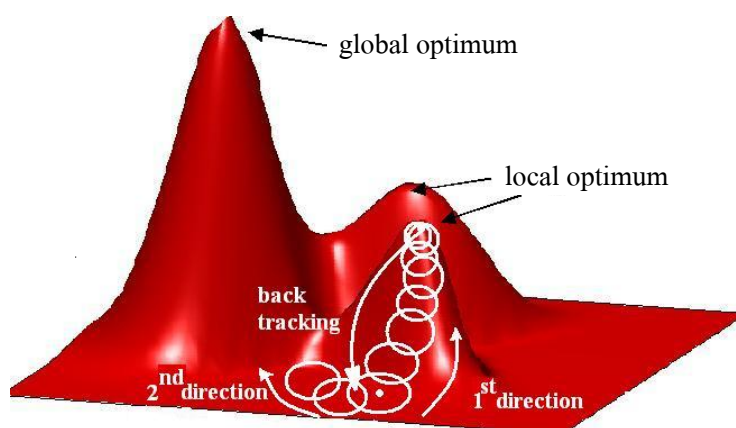
3) นำเซต $S_1(r)$ ไปทำการตรวจสอบกับฟังก์ชันวัตถุประสงค์เพื่อค้นหาคำตอบที่ดีที่สุด เก็บค่าคำตอบที่ดีที่สุดนี้ไว้เป็น best_neighbor1 ดังรูปที่ 3.24 โดยปกติฟังก์ชันวัตถุประสงค์จะเป็นสมการที่ใช้คำนวณได้ผลลัพธ์เชิงตัวเลข ซึ่งผลลัพธ์สามารถบอกได้ว่าคำตอบที่ได้จาก TS ในขณะนั้นให้ผลดีหรือไม่ดีอย่างไร ตัวอย่างของฟังก์ชันวัตถุประสงค์ที่นิยมใช้บ่อยก็คือ ฟังก์ชันที่คำนวณค่าความคลาดเคลื่อนระหว่างคำตอบที่ได้จาก TS กับคำตอบจริงๆที่ต้องการ

4) ถ้า best_neighbor1 ให้คำตอบที่ได้จากฟังก์ชันวัตถุประสงค์ดีกว่า best_neighbor ให้เก็บ best_neighbor ไว้ใน Tabu list แล้วทำการเปลี่ยน best_neighbor = best_neighbor1 ด้วย แต่ถ้า best_neighbor1 ไม่ให้คำตอบที่ดีกว่าให้ทำการเก็บ best_neighbor1 ไว้ใน Tabu list

5) ให้ $S_0 = \text{best_neighbor}$ จากขั้นตอนนี้เราจะได้ S_0 ที่ตำแหน่งใหม่เพื่อทำการค้นหาต่อไปดังรูปที่ 3.25

6) เพิ่ม count แล้วทำซ้ำขั้นตอน 2) - 6) ดังกล่าวข้างต้นจนการค้นหาคำตอบที่ต้องการ หรือ count = max_count คำตอบสุดท้ายจะเป็นคำตอบที่ดีที่สุด

สำหรับ ATS มีความพิเศษกว่า TS ตรงที่เพิ่มขั้นตอนพิเศษอีก 2 ขั้นตอน คือ การย้อนรอยรอบการค้นหา (Back tracking) และ การปรับรัศมีการค้นหา (Adaptive radius)

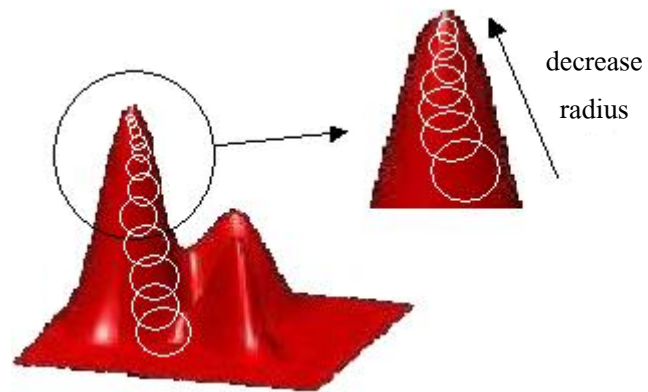


รูปที่ 3.25 ขั้นตอนการทำงานของ back tracking ใน ATS

การย้อนรอยรอบการค้นหา จะอนุญาตให้การค้นหาคำตอบสามารถเดินย้อนกลับทางเดิมที่ค้นหาแล้วได้เพื่อหลีกเลี่ยงการเจอค่าที่เหมาะสมที่สุดเฉพาะถิ่น ขั้นตอนดังกล่าวนี้จะถูกเพิ่มต่อจากขั้นตอนที่ 4) กรณีคำตอบที่ได้เกิดการวนซ้ำ โดย ATS จะเลือกค่าระหว่างการเดินที่ผ่านมาแล้วช่วงหนึ่งกับตำแหน่งปัจจุบัน แล้วทำการตั้งเป็นตำแหน่งเริ่มค้นหาใหม่หรือ S_0 ซึ่งในตอนเริ่มเดินค้นหาใหม่นี้คำตอบใหม่ที่ได้จากตำแหน่งใหม่นี้ไม่จำเป็นต้องดีกว่าคำตอบที่ผ่านมาก็ได้ ซึ่งในอัลกอริทึมที่ใช้ในวิทยานิพนธ์ได้กำหนดให้เรียกใช้การย้อนรอยรอบการค้นหา เมื่อคำตอบที่ได้เกิดการวนซ้ำหรือไม่ดีกว่าคำตอบที่ผ่านมากเกิน 3 ครั้ง โดยนำตำแหน่งการเดินที่ผ่านมาล่าสุด 5 ครั้งที่ได้จาก tabu list มาจัดลำดับเพื่อเลือกการเดินใหม่เพื่อหลีกเลี่ยงการเจอค่าเหมาะสมที่สุดเฉพาะถิ่น ที่เกิดขึ้น

การปรับรัศมีการค้นหาเป็นการลดขนาดของ R ลงในระหว่างการค้นหาซึ่งจะทำงานเมื่อการค้นหาเข้าใกล้ค่าเหมาะสมที่สุดในวงกว้างโดยจะเพิ่มขั้นตอนนี้เข้าไปในขั้นตอนที่ 2) เมื่อเริ่มต้นทำการค้นหาคำตอบรอบใหม่ ซึ่งในการค้นหาเมื่อคำตอบที่ได้เข้าใกล้ค่าเหมาะสมที่สุดในวงกว้างพื้นที่ในการค้นหาจะมีขนาดเล็กลง การใช้ R ที่มีขนาดใหญ่เกินไปอาจทำให้ข้ามคำตอบที่ดีที่สุดไปได้ ขนาดของ R ที่เล็กจะสามารถเข้าใกล้คำตอบที่ดีที่สุดได้ดีกว่า ในขณะที่ตอนเริ่มทำการค้นหาพื้นที่ในการค้นหากว้าง การใช้ R ที่เล็กเกินไปจะทำให้การค้นหาใช้เวลานานเนื่องจากมีรัศมีการค้นหาที่แคบ ฉะนั้นในการเริ่มค้นหาตอนแรกจำเป็นต้องใช้ R ที่มีขนาดใหญ่เพื่อความเร็วในการลู่เข้าและค่า R จะค่อยๆลดขนาดลงเมื่อการค้นหาเข้าใกล้ค่าเหมาะสมที่สุดในวงกว้าง ซึ่งในอัลกอริทึมที่ใช้ในวิทยานิพนธ์ได้กำหนดให้ ขนาดของ R เริ่มต้นที่ 0.25 เมื่อค่าวัตถุประสงค์สูงกว่า 10^{-1} และมีขนาดลดลงทีละ 0.1 เมื่อค่าวัตถุประสงค์ลดลง 0.1 ดังนี้ ขนาดของ R เท่ากับ 0.025 เมื่อค่าวัตถุประสงค์ต่ำกว่า 10^{-1} แต่ไม่ถึง 10^{-2} ขนาดของ R เท่ากับ 0.0025 เมื่อค่าวัตถุประสงค์ต่ำกว่า 10^{-2} แต่ไม่ถึง 10^{-3} ขนาดของ R เท่ากับ 0.25×10^{-3} เมื่อค่าวัตถุประสงค์ต่ำกว่า 10^{-3} แต่ไม่ถึง 10^{-4} ขนาดของ R

เท่ากับ 0.25×10^{-4} เมื่อค่าวัตถุประสงค์ต่ำกว่า 10^{-4} แต่ไม่ถึง 10^{-5} และขนาดของ R เท่ากับ 0.25×10^{-5} เมื่อค่าวัตถุประสงค์มีค่าต่ำกว่า 10^{-5}



รูปที่ 3.26 ขั้นตอนการทำงานของ adaptive radius ใน ATS

บทที่ 4

การฝังลายน้ำและการตรวจจับลายน้ำ

4.1 กล่าวนำ

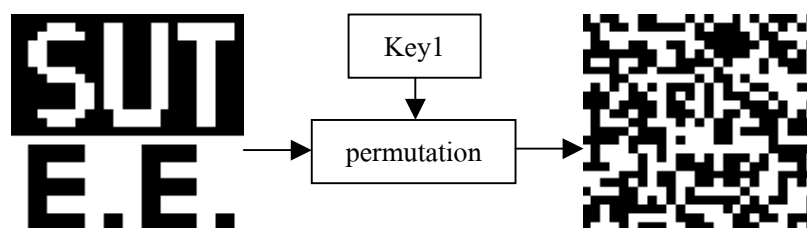
ในบทนี้กล่าวถึงระเบียบวิธีทั้งหมด ที่เกี่ยวข้องกับการฝังลายน้ำแบบกึ่งปิด และการตรวจจับลายน้ำ โดยแบ่งออกเป็นสองส่วน คือในหัวข้อข้อที่ 4.2 จะกล่าวถึงการฝังลายน้ำ ประกอบด้วย การเตรียมลายน้ำ การฝังลายน้ำลงบนโดเมนเวฟเลต และการประยุกต์ใช้การค้นหาแบบตามูเชิงปรับตัว ในการค้นหาค่าความเข้มในการฝัง และในหัวข้อที่ 4.3 จะกล่าวถึงการการตรวจจับลายน้ำ

4.2 การฝังลายน้ำ

การฝังลายน้ำใช้วิธีการฝังข้อมูลลายน้ำที่ต้องการลงบนสัมประสิทธิ์ของการแปลงเวฟเลต สัมประสิทธิ์ดังกล่าวจะถูกจัดเป็นกลุ่มๆ ให้เหมาะสมแล้วทำการเปลี่ยนค่าเฉลี่ยของแต่ละกลุ่มให้สอดคล้องกับข้อมูลลายน้ำ ซึ่งเป็นวิธีหนึ่งที่นิยมและมีประสิทธิภาพสามารถประยุกต์ใช้กับการแปลงชนิดอื่นๆ ได้

4.2.1 การเตรียมสัญญาณเสียงและลายน้ำ

สัญญาณเสียงที่ใช้ในงานวิจัยนี้เป็นสัญญาณเสียงเพลงบันทึกแบบดิจิทัลนามสกุล WAV แบบโมโน ที่อัตราสุ่ม 44100 kHz 16 บิต นาน 30 วินาที ส่วนข้อมูลลายน้ำที่ใช้เป็นรูปภาพ ขนาด 25×25 พิกเซล ซึ่งลายน้ำดังกล่าวจะนำมาผ่านการทำ permutation ดังรูปที่ 4.1 ก่อน เพื่อให้สอดคล้องกับเงื่อนไขความปลอดภัย (Security)



รูปที่ 4.1 แสดงกระบวนการทำ permutation จาก a) รูปภาพข้อความที่ใช้เป็นลายน้ำ

b) รูปที่ผ่านการทำ permutation แล้ว

4.2.2 การฝังลายน้ำในโดเมนเวฟเลต

1) สัญญาณลายน้ำ $W(x,y)$ เป็นภาพขาวดำที่ผ่านการทำ permutation แล้ว ขนาด $M_1 \times M_2$ พิกเซล โดยที่ $0 \leq x < M_1$, $0 \leq y < M_2$ แล้วทำการเปลี่ยนให้อยู่ในรูปเวกเตอร์แบบสั่มหรือ $v_k(i)$ โดยที่ $i = 1, 2, \dots, M_1, M_2$

2) สัญญาณเสียงจะถูกแยกองค์ประกอบเวฟเลตแบบ 5 ระดับ ($L = 5$) ในที่นี้เลือกใช้เวฟเลตแบบ Daubechies 4 หรือ db4 จากนั้นสั่มประสิทธิ์เวฟเลตใน approximation subband, cA^5 ดังกล่าว จะแบ่งออกเป็น k ส่วน จากขั้นตอนนี้เราจะได้สั่มประสิทธิ์เวฟเลตที่แบ่งส่วนแล้ว, $C_k(i)$

3) คำนวณค่าเฉลี่ยของ $C_k(i)$ ในแต่ละ k ส่วน แทนด้วย $m_k(i)$ แล้วนำมาลบออกจาก $C_k(i)$ ดังสมการที่ (4.1) เพื่อความสะดวกในกระบวนการฝัง

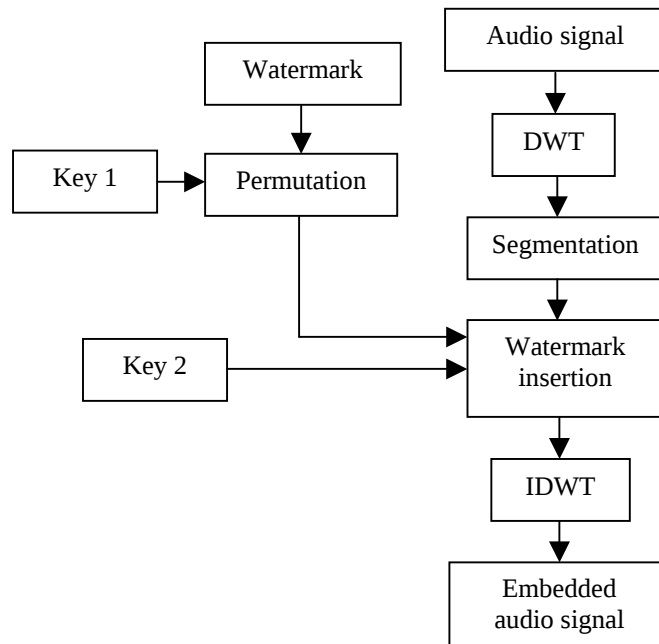
$$C'_k(i) = C_k(i) - m_k(i) \quad (4.1)$$

4) ให้ $C'_k(i)$ แทนสั่มประสิทธิ์ที่ผ่านการตัดแปลงจากขั้นตอนที่ 3 แต่ละบิตของลายน้ำจะฝังลงบน $C'_k(i)$ แต่ละส่วนโดยใช้กฎแจค็คือ Key 2 เพื่อสลับลำดับข้อมูล เงื่อนไขในการฝังแสดงตามขั้นตอนดังนี้ ถ้า $v_k(i) = 1$ สั่มประสิทธิ์ทุกตัวในส่วนที่เลือกจะถูกเพิ่มด้วย $\alpha \cdot \beta$ แต่ถ้า $v_k(i) = -1$ สั่มประสิทธิ์ทุกตัวในส่วนที่เลือกจะถูกลบด้วย $\alpha \cdot \beta$ โดยที่ β เป็นค่าที่มีขนาดเท่ากับ $m_k(i)$ ค่า α ได้จากการค้นหาโดย ATS มีฟังก์ชันวัตถุประสงค์คือ SNR สำหรับวัดคุณภาพเสียงที่ฝังลายน้ำแล้ว และการทำสหสัมพันธ์ (Sim) สำหรับวัดคุณภาพลายน้ำที่ตรวจจับได้เทียบกับลายน้ำต้นฉบับ สั่มประสิทธิ์เวฟเลตที่ฝังลายน้ำ $Y_k(i)$ แล้วสามารถหาได้จากสมการที่ (4.2)

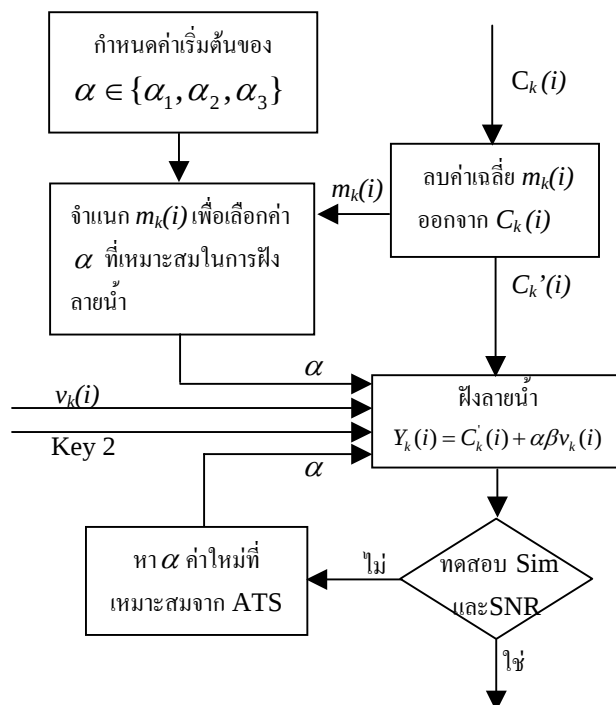
$$Y_k(i) = C'_k(i) + \alpha \beta v_k(i) \quad (4.2)$$

โดยที่ β คือ ค่าที่มีขนาดเท่ากับ $m_k(i)$ ของ $C'_k(i)$ ที่กำลังฝังลายน้ำ ส่วน $\alpha = \{ \alpha_1, \alpha_2, \alpha_3 \}$ คือ เซตของค่าความเข้มในการฝังลายน้ำ ซึ่งในตอนเริ่มแรกจะใช้เซตของ α ที่ประมาณค่าไว้ จากนั้น ATS เริ่มต้นทำงานค้นหาค่า α ที่เหมาะสม โดยการประเมินของฟังก์ชันวัตถุประสงค์ ถ้า α ที่หาได้จาก ATS ในแต่ละรอบของการค้นหาไม่ได้ผลลัพธ์ที่เหมาะสม α จะถูกค้นหาซ้ำอีกโดย ATS ซึ่งขั้นตอนนี้แสดงให้ดูดังรูปที่ 4.3

5) ซ้ำกระบวนการในขั้นตอนที่ 4) จนลายน้ำถูกฝังครบทุกบิต จากนั้นนำสั่มประสิทธิ์เวฟเลต $Y_k(i)$ ที่ฝังลายน้ำเรียบร้อยแล้วมาแปลงกลับ IDWT ให้อยู่ในรูปของสัญญาณเสียงเดิมในโดเมนเวลา



รูปที่ 4.2 แสดงขั้นตอนการฝังลายน้ำทั้งหมด



รูปที่ 4.3 แสดงขั้นตอนย่อยที่ 4) ในการฝังลายน้ำ

ในการฟังลายน้ำตามสมการที่ (4.2) นั้น สังเกตว่า $m_k(i)$ ควรเป็นค่าที่ได้จากสัญญาณเสียงในทางปฏิบัติ ซึ่งค่าดังกล่าวขึ้นอยู่กับลักษณะของสัญญาณเสียงเป็นสำคัญ เมื่อพิจารณาจะพบว่า ค่า $m_k(i)$ ที่มีค่ามากจะทนทานต่อการโจมตีได้ดีกว่า $m_k(i)$ ที่มีค่าน้อย ทำให้สามารถจำแนก $m_k(i)$ ออกเป็นกลุ่มๆ โดยกลุ่มที่มีค่า $m_k(i)$ สูงจะมีความทนทานต่อการโจมตีสูงจึงต้องการค่า α ค่าหนึ่ง ในขณะที่กลุ่มที่มีค่า $m_k(i)$ ต่ำกว่าลงมาจะมีความทนทานต่อการโจมตีลดต่ำลงไปด้วย ทำให้ต้องการค่า α อีกค่าหนึ่งที่สอดคล้องกับการลดลงของค่า $m_k(i)$ นี้

ส่วนในผลกระทบด้านคุณภาพของเสียง ปรากฏว่าในกลุ่มที่มีค่า $m_k(i)$ มากเมื่อใช้ค่า α น้อยก็ส่งผลต่อการเปลี่ยนแปลงของค่า SNR อย่างมาก ในขณะที่กลุ่มที่มีค่า $m_k(i)$ ต่ำกว่าลงมาจะสามารถใช้ค่า α ที่มากกว่าได้โดยส่งผลต่อค่า SNR น้อยลงตามลำดับ ซึ่งสอดคล้องกับข้อสังเกตด้านความทนทานข้างต้น ในงานวิจัยวิทยานิพนธ์นี้จึงได้กำหนดให้เซตของ α มีสมาชิกจำนวน 3 ค่า $\alpha = \{\alpha_1, \alpha_2, \alpha_3\}$ โดยจำแนกกลุ่ม $m_k(i)$ ออกเป็น 4 กลุ่มคือ กลุ่มที่มีค่าสูงกว่า 0.01 จะใช้ค่า $\alpha = 1$ กลุ่มที่มีค่าต่ำกว่า 0.01 ลงมาใช้ค่าความเข้มในการฟัง α_3 กลุ่มที่มีค่าต่ำกว่า 0.005 ลงมาใช้ค่าความเข้มในการฟัง α_2 และกลุ่มที่มีค่าต่ำกว่า 0.001 ใช้ค่าความเข้มในการฟัง α_1

4.2.3 การค้นหาค่าความเข้มในการฟังลายน้ำด้วย ATS

ประเด็นหลักที่ต้องคำนึงในการฟังลายน้ำ คือลายน้ำต้องมีความทนทานต่อการโจมตีทางการประมวลผลสัญญาณแบบต่างๆ กล่าวคือสามารถตรวจจับลายน้ำจากสัญญาณเสียงที่ฟังลายน้ำที่ถูกโจมตีได้ และอีกประการที่สำคัญคือสัญญาณเสียงที่ผ่านการฟังลายน้ำแล้วต้องมีคุณภาพเสียงคงเดิมโดยไม่สามารถรับฟังความแตกต่างเมื่อเปรียบเทียบกับสัญญาณเสียงต้นฉบับได้

จากความสำคัญดังกล่าว เราเลือก SNR เป็นฟังก์ชันวัตถุประสงค์หนึ่งของ ATS เพื่อค้นหาค่าความเข้มในการฟังหรือ α ที่เหมาะสม เนื่องจากลายน้ำสามารถมองได้ว่าเป็นสัญญาณรบกวนแบบมีคุณค่า SNR เป็นฟังก์ชันที่ใช้วัดพลังงานของสัญญาณรบกวนต่อพลังงานของสัญญาณเดิมใดๆ ดังสมการที่ (4.3)

$$SNR = 10 \log \frac{\sum_i S_o^2(i)}{\sum_i [S_o(i) - S_w(i)]^2} \quad (4.3)$$

กำหนดให้ S_o คือสัญญาณเสียงต้นฉบับ S_w คือสัญญาณเสียงที่ฟังลายน้ำแล้ว $i = 1, 2, \dots, N$ เมื่อ N คือความยาวของสัญญาณเสียง ถ้า α มีค่ามาก SNR จะต่ำส่งผลให้คุณภาพของเสียงต่ำลงด้วย ในทางกลับกัน α ที่น้อย SNR จะสูง ส่งผลให้คุณภาพเสียงแทบไม่มีความผิดเพี้ยนจากต้นฉบับเลย ซึ่งใน

อัลกอริทึมได้กำหนด SNR มีค่าเท่ากับ 23 เดซิเบล สอดคล้องกับมาตรฐานของ IFPI และเพื่อคุณภาพเสียงที่ดี

นอกจากนี้ในงานวิจัยวิทยานิพนธ์ได้ใช้การวัดค่าสหสัมพันธ์ (Sim) เป็นอีกฟังก์ชันวัดคุณภาพเสียงหนึ่งของ ATS ซึ่งเป็นการทำสหสัมพันธ์ระหว่างลายน้ำต้นฉบับกับลายน้ำที่ตรวจจับได้ เพื่อใช้บอกความทนทานของลายน้ำ ดังสมการที่ (4.4)

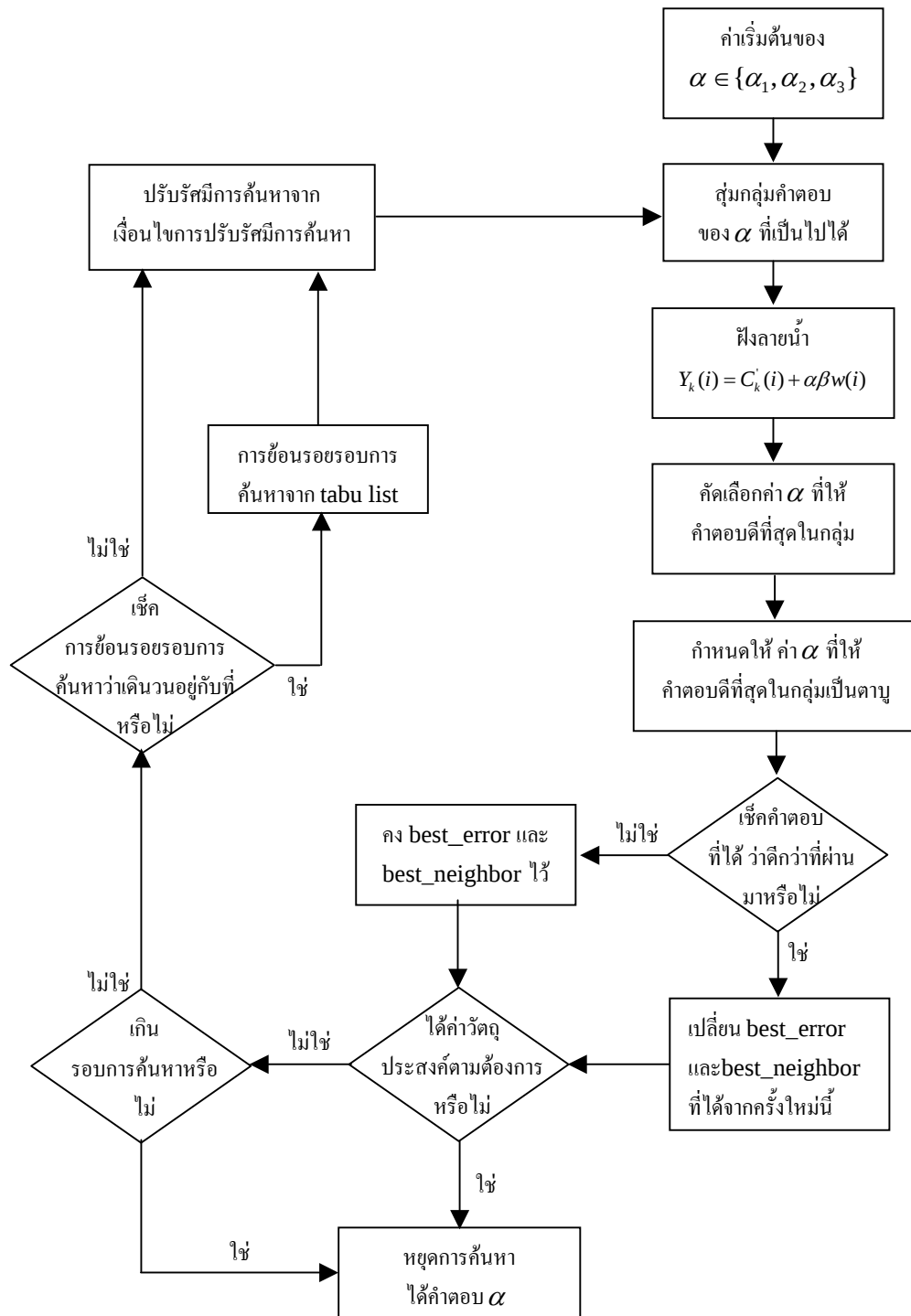
$$Sim = \frac{\sum_{i=1}^{M_1} \sum_{j=1}^{M_2} W(i, j) W^*(i, j)}{\sqrt{\sum_{i=1}^{M_1} \sum_{j=1}^{M_2} W(i, j)^2} \sqrt{\sum_{i=1}^{M_1} \sum_{j=1}^{M_2} W^*(i, j)^2}} \quad (4.4)$$

กำหนดให้ W คือลายน้ำต้นฉบับ W^* คือลายน้ำที่ตรวจจับได้ $i = 1, 2, \dots, M_1$ และ $j = 1, 2, \dots, M_2$ โดยค่าสหสัมพันธ์ที่ได้ถ้าเท่ากับหนึ่ง แสดงว่าลายน้ำที่ตรวจจับได้เหมือนกับลายน้ำต้นฉบับทุกประการ จากการที่ ATS มีสองฟังก์ชันวัดคุณภาพเสียงที่ขัดแย้งกันดังกล่าวทำให้การค้นหาคำตอบของ ATS น่าเชื่อถือมากยิ่งขึ้น ดังสมการที่ (4.5)

$$V = 0.5 \times (1 - Sim) + 0.5 \times (23 - SNR) \quad (4.5)$$

เมื่อ V เป็นค่าวัดคุณภาพเสียง ซึ่งทั้งสองฟังก์ชันวัดคุณภาพเสียงต้องมีตัวประกอบถ่วงน้ำหนักเพื่อกำหนดความสำคัญในแต่ละฟังก์ชันวัดคุณภาพเสียงขณะทำการค้นหาคำตอบด้วย ATS ตัวประกอบถ่วงน้ำหนักของทั้งสองฟังก์ชันนี้ต้องรวมกันมีค่าเท่ากับหนึ่ง ในอัลกอริทึมได้กำหนดให้มีค่าเท่ากับ 0.5 ทั้งสองฟังก์ชันวัดคุณภาพเสียง เนื่องจากทั้งสองฟังก์ชันมีความสำคัญเท่ากัน กล่าวคือในการค้นหาคำตอบของ ATS ถ้า α มีค่ามากเกินไป ค่าสหสัมพันธ์จะสูง ลายน้ำจะมีความทนทานต่อการโจมตีต่างๆ ได้ดี แต่คุณภาพของเสียงลดต่ำลง SNR จะต่ำเกินกว่ารับได้ตามเกณฑ์มาตรฐาน ฟังก์ชันวัดคุณภาพเสียงจะกระตุ้น ATS เพื่อค้นหาคำความเข้มในการฟังที่เหมาะสมค่าใหม่ ในทางกลับกัน α ที่น้อยเกินไป จะให้ SNR สูง ลายน้ำจะไม่ได้ยินแต่ก็ไม่ทนทานต่อการโจมตี กระบวนการนี้จะดำเนินซ้ำจนบรรลุฟังก์ชันวัดคุณภาพเสียงทั้งสอง ไม่เช่นนั้น ATS จะปฏิเสธคำตอบที่ได้และทำการเริ่มค้นหาคำตอบใหม่ โดยก่อนเริ่มรอบการค้นหาใหม่ ATS จะทำการเช็คคำตอบในขณะนั้นกับ tabu list ว่าดีขึ้นกว่าคำตอบที่ผ่านมาหรือไม่เพื่อเรียกใช้การย้อนรอยรอบการค้นหา แล้วทำการปรับรัศมีในการค้นหาคำตอบด้วยการปรับรัศมีการค้นหา ก่อนทำการค้นหาในรอบถัดไป ในอัลกอริทึมได้

กำหนดให้ ATS หยุดการค้นหาเมื่อค่า V ที่ได้ต่ำกว่า 10^{-5} หรือเกินจำนวนรอบการค้นหาขั้นตอนกระบวนการค้นหาคำตอบแบบ ATS ทั้งหมดแสดงไว้ในรูปที่ 4.4

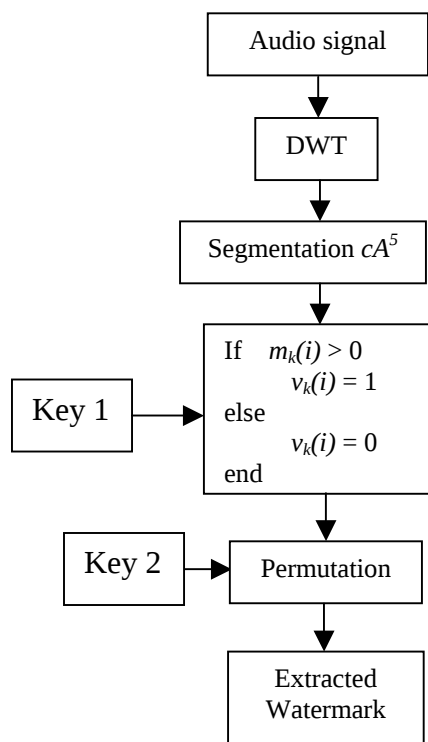


รูปที่ 4.4 ฟังก์ชันการค้นหาด้วย ATS

4.3 การตรวจจับลายน้ำ

การตรวจจับลายน้ำเป็นกระบวนการย้อนกลับ และไม่ต้องใช้แม้แต่สัญญาณเสียงต้นฉบับหรือลายน้ำต้นฉบับช่วย ดังขั้นตอนต่อไปนี้

- 1) เริ่มต้นจากการแยกองค์ประกอบสัญญาณเสียงที่ต้องการด้วยการแปลงเวฟเล็ต
- 2) นำ cA^5 มาแบ่งออกเป็นส่วนย่อย k ส่วน ให้เท่ากับจำนวนบิตของลายน้ำ
- 3) จากนั้นหาค่าเฉลี่ยของสัมประสิทธิ์เวฟเล็ตในแต่ละส่วน ถ้าค่าเฉลี่ยมีขนาดมากกว่าศูนย์ จะตรวจจับได้ลายน้ำบิต '1' ในขณะที่ถ้าค่าเฉลี่ยต่ำกว่าศูนย์แสดงว่าลายน้ำบิต '0' ถูกตรวจจับได้
- 4) ขั้นตอนดังกล่าวจะทำซ้ำจนได้ลายน้ำครบตามจำนวน k บิตที่ฝังไป จากนั้นลายน้ำที่ได้จะจัดให้อยู่ในรูปภาพขนาด $M1 \times M2$ พิกเซล แล้วย้อนกลับกระบวนการ permutation ด้วยกุญแจชุดเดียวกับในขั้นตอนการฝังลายน้ำ ท้ายสุดจะได้รูปภาพวาดเป็นลายน้ำที่ตรวจจับได้



รูปที่ 4.5 ผังแสดงการตรวจจับลายน้ำ

บทที่ 5

ผลการทดสอบ

การทดสอบความทนทานของลายน้ำต่อการโจมตีทางการประมวลผลสัญญาณแบบต่างๆทั้งหมด 12 การโจมตี แบ่งออกตามชนิดเพลงต่างๆทั้งไทยและสากล 6 ชนิด ชนิดละ 5 เพลง ผลการทดสอบในบทนี้แบ่งออกเป็น 2 ส่วนคือ ส่วนแรกเป็นผลของอัลกอริทึมที่พัฒนาขึ้นในงานวิจัยวิทยานิพนธ์ สามารถดูรายละเอียดอัลกอริทึมได้จากภาคผนวก ก ส่วนที่สองเป็นผลทดสอบเปรียบเทียบความทนทานของอัลกอริทึมในวิทยานิพนธ์กับอัลกอริทึมที่ใช้รหัสแก้ไขความผิดพลาด สามารถดูรายละเอียดอัลกอริทึมนี้ได้จากภาคผนวก ง ส่วนผลการทดสอบทั้งหมดของอัลกอริทึมทั้งสองดูรายละเอียดได้จากภาคผนวก จ แบบทดสอบคุณภาพเสียงดูรายละเอียดเพิ่มเติมได้จากภาคผนวก ฉ อัลกอริทึมที่ใช้การค้นหาแบบตามูเชิงปรับตัว ต้องสามารถหาค่าความเข้มในการฝังที่เหมาะสมในการสร้างลายน้ำกับสัญญาณเสียงอันนำมาซึ่งความทนทานของลายน้ำ และคุณภาพเสียงที่ดีได้แม้ว่าจะไม่ได้ใช้การป้องกันข้อมูลด้วยเทคนิคพิเศษอื่นใด ซึ่งให้เห็นถึงความสำคัญของค่าความเข้มในการฝังที่ขึ้นกับคุณลักษณะของสัญญาณเสียงนั้นๆเป็นสำคัญ สัญญาณเสียงที่นำมาทดสอบสามารถดูรายละเอียดในแต่ละเพลงได้จากภาคผนวก ก

อัลกอริทึมที่นำมาเปรียบเทียบ มีความทนทานของลายน้ำที่ดีโดยใช้เทคนิคการเข้ารหัสแก้ไขความผิดพลาดแบบ cyclic code เพื่อเพิ่มความทนทาน ซึ่งจัดเป็นรหัสประเภทหนึ่งในกลุ่มบล็อกเชิงเส้น เป็นรหัสที่มีโครงสร้างเป็นระบบระเบียบชัดเจน สามารถทำงานโดยวิธีการทางคณิตศาสตร์อย่างมีประสิทธิภาพ กระบวนการเข้ารหัสและถอดรหัสไม่ซับซ้อน วิธีการดังกล่าวลายน้ำจะถูกเข้ารหัสโดยข้อมูลรหัสชุดหนึ่งซึ่งข้อมูลรหัสชุดนี้จะถูกเพิ่มรวมไปกับลายน้ำด้วย เมื่อลายน้ำถูกโจมตีทำให้ข้อมูลลายน้ำผิดไป ข้อมูลรหัสชุดนี้จะสามารถกู้คืนสัญญาณลายน้ำที่ถูกต้องขึ้นมาได้ในระดับหนึ่งขึ้นอยู่กับคุณลักษณะรหัสที่ใช้ ซึ่งจากคุณสมบัติดังกล่าวสามารถชดเชยผลของความไม่ทนทานของอัลกอริทึมที่ใช้ฝังลายน้ำได้

5.1 ผลการทดสอบความทนทานของลายน้ำ

5.1.1 ผลการทดสอบด้วยอัลกอริทึมที่เสนอในงานวิจัยวิทยานิพนธ์

ผลการทดสอบความทนทานของลายน้ำต่อการโจมตีแบบต่างๆทั้งหมด 12 การโจมตี แสดงในรูปแบบตารางบันทึกค่าสหสัมพันธ์ระหว่างลายน้ำต้นฉบับกับลายน้ำที่ตรวจจับได้จาก สัญญาณเสียงเพลงสากลและเพลงไทย 6 ชนิด ชนิดละ 5 บทเพลง ดังแสดงในตารางที่ 5.1 ถึง ตารางที่ 5.6 โดยกำหนดให้ค่า SNR มีค่าเท่ากับ 23 เดซิเบล ส่วนรูปภาพลายน้ำที่ตรวจจับได้หลังผ่านการ โจมตีที่ค่าสหสัมพันธ์ต่างๆจะแสดงไว้ในภาคผนวก ข

ตารางที่ 5.1 ผลการทดสอบกับเพลงสากล classic

	Beautiful Blue Danube	Hungarian Dances	Marriage of Figaro	Moonlight	Romance for Violin & Orchestra
No attack	1	1	1	1	1
Downsampling	0.9983	1	1	1	1
LPF	0.9966	1	1	1	1
Random noise	1	1	1	0.9983	1
Gaussian noise	1	1	1	0.9983	1
Denoise-DWT	0.9983	1	1	1	1
MP3: 128 kbps	1	1	1	1	1
MP3: 56 kbps	1	1	1	0.9983	1
Jittering	0.9966	1	1	1	1
Crop (5×100)	0.9950	1	0.9966	1	1
Crop (10×100)	0.9950	0.9966	0.9950	0.9966	1
Crop (10×500)	0.9815	0.9831	0.9815	0.9815	0.9831
Crop (10×1000)	0.9640	0.9662	0.9662	0.9693	0.9693

ตารางที่ 5.2 ผลการทดสอบกับเพลงสากล jazz

	Beauty in me	Shade of pain	Till I get it right	I could wait a book	Squeeze me
No attack	1	1	1	1	1
Downsampling	0.9983	1	1	1	1
LPF	1	0.9966	1	1	1
Random noise	1	0.9966	1	0.9966	0.9983
Gaussian noise	0.9983	0.9966	1	1	0.9983
Denoise-DWT	1	1	1	1	1
MP3: 128 kbps	0.9983	1	1	0.9983	1
MP3: 56 kbps	0.9983	0.9966	0.9933	0.9983	1
Jittering	1	0.9983	1	0.9983	1
Crop (5×100)	0.9950	1	0.9966	1	1
Crop (10×100)	0.9950	0.9966	0.9950	0.9966	1
Crop (10×500)	0.9815	0.9831	0.9815	0.9815	0.9831
Crop (10×1000)	0.9640	0.9662	0.9662	0.9693	0.9693

ตารางที่ 5.3 ผลการทดสอบกับเพลงสากล pop

	Vincent	White flag	Sunrise	Sometime	Where is the love
No attack	1	1	1	1	1
Downsampling	0.9983	1	1	0.9983	1
LPF	1	0.9966	1	1	1
Random noise	0.9983	0.9950	1	1	1
Gaussian noise	0.9983	0.9950	1	1	1
Denoise-DWT	1	1	1	1	1
MP3: 128 kbps	1	1	1	1	1
MP3: 56 kbps	1	1	0.9883	1	0.9899
Jittering	0.9983	1	1	1	1
Crop (5×100)	0.9983	1	0.9966	0.9983	0.9966
Crop (10×100)	0.9966	0.9983	0.9950	0.9983	0.9933
Crop (10×500)	0.9865	0.9815	0.9730	0.9797	0.9815
Crop (10×1000)	0.9605	0.9675	0.9623	0.9744	0.9710

ตารางที่ 5.4 ผลการทดสอบกับเพลงไทยเดิม

	มหาฤกษ์	วิวาห์ พระสมุทร	เทวา ประสิทธิ์	โพธิ์สัตว์	มหาชัย
No attack	1	1	1	1	1
Downsampling	1	1	0.9983	1	1
LPF	0.9983	1	1	1	1
Random noise	0.9883	0.9966	0.9950	0.9966	0.9966
Gaussian noise	0.9933	0.9983	1	0.9966	1
Denoise-DWT	1	1	1	1	1
MP3: 128 kbps	1	1	0.9983	1	1
MP3: 56 kbps	1	1	0.9966	1	0.9983
Jittering	0.9950	0.9983	0.9983	0.9983	1
Crop (5×100)	0.9983	1	0.9966	0.9983	1
Crop (10×100)	1	1	0.9983	0.9983	0.9983
Crop (10×500)	0.9883	0.9883	0.9916	0.9916	0.9848
Crop (10×1000)	0.9710	0.9662	0.9710	0.9730	0.9662

ตารางที่ 5.5 ผลการทดสอบกับเพลงไทยสุนทราภรณ์

	เย็นลมว่าว	พรหมลิขิต	พลอร์ เฟื่องฟ้า	ตัดสวาท	ขวัญใจเจ้า ทวย
No attack	1	1	1	1	1
Downsampling	1	1	1	1	1
LPF	1	1	1	0.9966	0.9983
Random noise	1	1	1	1	1
Gaussian noise	1	0.9983	1	1	1
Denoise-DWT	1	1	1	1	1
MP3: 128 kbps	1	1	1	1	1
MP3: 56 kbps	0.9966	0.9950	0.9983	0.9916	0.9983
Jittering	0.9983	1	1	1	1
Crop (5×100)	1	0.9983	0.9950	0.9983	0.9983
Crop (10×100)	0.9966	0.9950	1	0.9950	1
Crop (10×500)	0.9782	0.9865	0.9762	0.9815	0.9831
Crop (10×1000)	0.9675	0.9640	0.9710	0.9710	0.9623

ตารางที่ 5.6 ผลการทดสอบกับเพลงไทยลูกทุ่ง

	เหนือไหม คนดี	อักษา ขอนแก่น	สร้างวิมาน	สาวฝั่งโขง	สิบหกปีแห่ง ความหลัง
No attack	1	1	1	1	1
Downsampling	1	0.9983	1	0.9983	1
LPF	1	0.9966	0.9983	0.9983	1
Random noise	1	0.9950	0.9983	0.9966	0.9882
Gaussian noise	1	0.9983	0.9983	0.9950	0.9950
Denoise-DWT	1	1	1	1	1
MP3: 128 kbps	1	0.9983	1	0.9983	0.9983
MP3: 56 kbps	1	0.9950	1	0.9950	0.9966
Jittering	1	0.9983	1	0.9983	0.9983
Crop (5×100)	0.9983	1	1	1	1
Crop (10×100)	0.9966	0.9983	0.9983	0.9933	0.9983
Crop (10×500)	0.9797	0.9933	0.9865	0.9848	0.9815
Crop (10×1000)	0.9710	0.9710	0.9693	0.9710	0.9605

ผลการทดสอบให้ค่าสหสัมพันธ์มากกว่า 0.9 ตามที่กำหนดไว้ แสดงให้เห็นว่าอัลกอริทึมที่ใช้ในวิทยานิพนธ์ให้ความทนทานต่อการโจมตีเป็นอย่างดี และลายน้ำที่ตรวจจับได้ส่วนมากมีค่าสหสัมพันธ์เท่ากับ 1 แสดงว่าลายน้ำส่วนใหญ่ไม่ได้รับความเสียหายใดๆเลยแม้จะผ่านการโจมตีแบบต่างๆ ค่าสหสัมพันธ์ที่ได้ต่ำสุดมีค่า 0.9605 จากเพลงสิบหกปีแห่งความหลังเมื่อผ่านการโจมตีแบบ Crop (10×1000) ซึ่งเป็นการโจมตีโดยลบข้อมูลสัญญาณเสียงบางส่วนออก ทำให้ข้อมูลลายน้ำสูญหายไปอย่างถาวร ในส่วนของผลการทดสอบได้ใช้ MP3 ที่อัตราบิตต่างกัน 2 ระดับซึ่งให้ผลที่แทบไม่แตกต่างกันเลย โดย MP3 ที่อัตราบิต 128 kbps ให้ค่าสหสัมพันธ์เกือบเป็น 1 ทั้งหมด ซึ่งปกติที่อัตราบิตระดับนี้จะนิยมสำหรับการบันทึกที่ละเมียดลิขสิทธิ์ และ MP3 ที่อัตราบิต 56 kbps ซึ่งเป็นอัตราบิตต่ำกว่าแบบแรก อัลกอริทึมก็สามารถให้ลายน้ำที่มีความทนทานสูงได้

5.1.2 ผลการเปรียบเทียบความทนทานของลายน้ำระหว่างอัลกอริทึมที่เสนอในงานวิจัยวิทยานิพนธ์ (ATS) กับอัลกอริทึมที่ใช้รหัสแก้ไขความผิดพลาด (ECC)

อัลกอริทึมที่ใช้เปรียบเทียบได้ใช้เทคนิคทางทฤษฎีข่าวสาร โดยการใช้รหัสแก้ไขความผิดพลาด ซึ่งเป็นวิธีที่มีประสิทธิภาพและนิยมใช้งานอย่างแพร่หลายในการสื่อสารแบบดิจิทัลและโทรคมนาคม ซึ่งการประยุกต์ใช้รหัสแก้ไขความผิดพลาดปรากฏอยู่ในงานวิจัยเกี่ยวกับการสร้างลายน้ำเป็นจำนวนมาก เช่น จากการศึกษาผลงานวิจัยของ Wei Li et al. (2003) ที่กล่าวถึงผลของการใช้รหัสแก้ไขความผิดพลาดร่วมกับการสร้างลายน้ำกับสัญญาณเสียงในเวฟเลตโดเมนซึ่งให้ผลการทดสอบที่ดี ในวิทยานิพนธ์ได้ใช้รหัส cyclic (23,12,3) ที่มีคำรหัส 11 บิต บิตข้อมูล 12 บิต แก้ไขบิตผิดพลาดได้ 3 บิตจากข้อมูลที่เข้ารหัสแล้วทั้งหมด ซึ่งการใช้รหัสดังกล่าวส่งผลดีเนื่องจากสามารถแก้ไขบิตผิดพลาดได้สูง ในขณะที่ต้องการคำรหัสไม่มากเมื่อเทียบกับรหัสแบบอื่นๆ

ตารางที่ 5.7 ผลการเปรียบเทียบในเพลง Till I get it right

	ATS	ECC
No attack	1	1
Downsampling	1	1
LPF	1	0.9693
Random noise	1	1
Gaussian noise	1	0.9693
Denoise-DWT	1	1
MP3: 128 kbps	1	1
MP3: 56 kbps	1	1
Jittering	1	0.9950
Crop (5×100)	1	1
Crop (10×100)	1	1
Crop (10×500)	0.9882	1
Crop (10×1000)	0.9730	0.9933

ตารางที่ 5.8 ผลการเปรียบเทียบในเพลง Sunrise

	ATS	ECC
No attack	1	1
Downsampling	1	1
LPF	1	0.9966
Random noise	1	1
Gaussian noise	1	0.9605
Denoise-DWT	1	1
MP3: 128 kbps	1	0.9134
MP3: 56 kbps	0.9882	0.9134
Jittering	1	0.9950
Crop (5 × 100)	1	1
Crop (10 × 100)	1	1
Crop (10 × 500)	0.9831	1
Crop (10 × 1000)	0.9797	0.9933

ตารางที่ 5.9 ผลการเปรียบเทียบในเพลงฟลอร์เฟื่องฟ้า

	ATS	ECC
No attack	1	1
Downsampling	1	1
LPF	1	0.9983
Random noise	1	1
Gaussian noise	1	1
Denoise-DWT	1	1
MP3: 128 kbps	1	0.9950
MP3: 56 kbps	1	0.9950
Jittering	1	1
Crop (5 × 100)	1	1
Crop (10 × 100)	1	1
Crop (10 × 500)	0.9950	1
Crop (10 × 1000)	0.9848	1

ตารางที่ 5.7 ถึง ตารางที่ 5.9 แสดงค่าสหสัมพันธ์ของลายน้ำจากการโจมตีแบบต่างๆ โดยเปรียบเทียบระหว่างอัลกอริทึมสองแบบคือ อัลกอริทึมวิธีที่ใช้ในวิทยานิพนธ์ (ATS) กับอัลกอริทึมที่ใช้รหัสแก้ไขความผิดพลาด (ECC) กำหนดให้อัลกอริทึมทั้งสองมี SNR เท่ากับ 20 เดซิเบล สำหรับผลการทดลองทั้งหมดสามารถดูได้จากภาคผนวก จ การทดสอบชี้ให้เห็นว่าอัลกอริทึมที่นำเสนอมีความทนทานต่อการโจมตีแบบต่างๆ และให้ผลใกล้เคียงกับอัลกอริทึมที่ใช้รหัสแก้ไขความผิดพลาด จะแตกต่างตรงการโจมตีแบบ Crop ผลปรากฏว่าเทคนิคที่นำมาเปรียบเทียบให้ผลการทดสอบดีกว่าเนื่องจากการโจมตีแบบนี้เป็นการลบข้อมูลสัญญาณเสียงบางส่วนออกอย่างถาวร รหัสแก้ไขความผิดพลาดสามารถสร้างข้อมูลขึ้นมาใหม่ได้ภายใต้เงื่อนไขของคำรหัส ทำให้การโจมตีแบบดังกล่าวไม่ส่งผลกับความทนทานของลายน้ำที่ใช้รหัสแก้ไขความผิดพลาดมากนัก

5.2 ผลการประเมินคุณภาพเสียง

การประเมินคุณภาพของเสียงหลังจากการฝังลายน้ำโดยการฟังจากผู้ทำการทดสอบจำนวน 9 คนแบ่งออกเป็นชาย 7 คน หญิง 2 คน ช่วงอายุระหว่าง 25-45 ปี ที่ห้อง preview หน่วยผลิตและพัฒนาสื่อการศึกษา อาคารปฏิบัติการและเครื่องมือ 7 มหาวิทยาลัยเทคโนโลยีสุรนารี ซึ่งรายละเอียดแบบทดสอบสามารถดูได้จากภาคผนวก ฉ

เพลงที่ใช้ฟังทดสอบแบ่งออกเป็นเพลงชนิดต่างๆทั้งไทยและสากล 6 ชนิด ชนิดละ 5 บท เพลง บทเพลงละ 3 ตัวอย่าง คือเพลงต้นฉบับที่ยังไม่ฝังลายน้ำ 1 ตัวอย่าง และเพลงทดสอบอีก 2 ตัวอย่าง รวมเพลงที่ใช้ฟัง 90 ตัวอย่าง ใช้เวลาทดสอบทั้งสิ้นประมาณ 1 ชั่วโมง

ขั้นตอนการทดสอบเริ่มจากการฟังเสียงเพลงต้นฉบับที่ยังไม่ฝังลายน้ำ 1 ตัวอย่าง 1 ครั้ง ร่วมกับฟังเพลงทดสอบ 2 ตัวอย่าง ตัวอย่างละ 1 ครั้ง โดยเพลงทดสอบจะเป็นการสุ่มระหว่างเพลงต้นฉบับ 1 เพลง กับเพลงที่มีลายน้ำฝังอยู่อีก 1 เพลง หรือจะเป็นเพลงที่มีลายน้ำฝังอยู่ทั้ง 2 เพลงก็ได้ จากนั้นทำเครื่องหมายในตัวเลือกต่างๆ ดังรูปที่ 5.10

4. เพลง Moonlight

ก. เพลงทดสอบที่ 1 มีคุณภาพลดลง

ข. เพลงทดสอบที่ 2 มีคุณภาพลดลง

ค. เพลงทดสอบที่ 1 และ 2 ไม่มีคุณภาพลดลงเลย

ง. เพลงทดสอบที่ 1 และ 2 มีคุณภาพลดลงทั้งคู่

รูปที่ 5.10 ตัวอย่างของตัวเลือกในแบบทดสอบคุณภาพเสียง

ในการฟังลายน้ำคุณภาพของเสียงจากการรับฟังระหว่างเพลงต้นฉบับ กับเพลงที่ฟังลายน้ำแล้วหรือเพลงทดสอบต้องไม่แตกต่างกัน ถ้ามีความแตกต่างเกิดขึ้นแสดงว่าอัลกอริทึมที่ใช้ยังไม่ดีพอทำให้เพลงทดสอบมีคุณภาพของเสียงลดลงไปจากเดิม ในการนับคะแนนจะให้คะแนน 1 คะแนน จากข้อที่ผู้ทำการประเมินเลือกผิด หรือเลือกตัวเลือก ค. ซึ่งแสดงว่าคุณภาพของการรับฟังเพลงต้นฉบับกับเพลงที่ฟังลายน้ำแล้วไม่แตกต่างกัน ผลจากแบบทดสอบทั้งหมด 270 ข้อ กลุ่มผู้ทำการประเมินสามารถเลือกได้ถูกต้องเพียง 14 ข้อ หรือกล่าวได้ว่าผลการทดสอบคุณภาพเสียงเพลงที่ฟังลายน้ำแล้วเทียบกับฟังเพลงต้นฉบับมีคุณภาพเหมือนกัน 94.81% แสดงให้เห็นว่าอัลกอริทึมที่ใช้ฟังลายน้ำบนเวฟเลต โดเมนให้คุณภาพเสียงที่ดี แทบจะไม่สามารถแยกความแตกต่างจากเสียงต้นฉบับได้

5.3 วิเคราะห์ผลการทดสอบ

พิจารณาผลการทดสอบ ความทนทานของลายน้ำที่ได้หลังจากผ่านการโจมตีแบบต่างๆ โดยการสร้างลายน้ำด้วยอัลกอริทึมที่นำเสนอในงานวิจัยวิทยานิพนธ์มีค่าสหสัมพันธ์มากกว่า 0.9 ตามที่กำหนดไว้ โดยสัญญาณเสียงที่ฟังลายน้ำแล้วให้คุณภาพเสียงที่ดี ค่า data pay load ที่ได้ใช้มีค่าประมาณ 21 bps และ SNR เท่ากับ 23 เดซิเบล สอดคล้องกับเงื่อนไขที่ IFPI กำหนด ซึ่งที่ค่า สหสัมพันธ์ ดังกล่าวสามารถระบุความมีลิขสิทธิ์ในสัญญาณเสียงเพลงได้อย่างถูกต้องจากรูปภาพลายน้ำที่ชัดเจนดังในภาคผนวก ข. แสดงให้เห็นว่าอัลกอริทึมที่ฟังลายน้ำลงบนสัมประสิทธิ์เวฟเลต โดยใช้ค่าความเข้มในการฟัง ($\alpha_1, \alpha_2, \alpha_3$) ที่ได้จากการค้นหาแบบตามูเชิงปรับตัวสามารถให้ความทนทานของลายน้ำที่สูงในขณะที่ยังคงรักษาคุณภาพของเสียงที่ดีไว้ ในขณะที่การตรวจจบลายน้ำสัญญาณเสียงที่ผ่านการโจมตีสามารถนำมาทำการตรวจจบลายน้ำได้เลยโดยไม่ต้องทำการปรับปรุงคุณภาพสัญญาณก่อน ยกเว้นการโจมตีบางแบบ เช่น การเพิ่มสัญญาณรบกวนแบบสุ่มที่ต้องทำการกรองสัญญาณรบกวนออกก่อนด้วยการแปลงเวฟเลต

แม้ว่าจำนวนบิตที่ผิดพลาดระหว่างอัลกอริทึมวิธีที่ใช้ในวิทยานิพนธ์ กับอัลกอริทึมที่นำมาเปรียบเทียบจะมีค่าใกล้เคียงกันก็ตาม แต่เมื่อพิจารณาผลของอัลกอริทึมที่นำมาเปรียบเทียบเมื่อไม่ได้ใช้รหัสแก้ไขความผิดพลาดร่วมด้วย (ไม่ใช่ ECC) ปรากฏว่าอัลกอริทึมที่นำมาเปรียบเทียบแท้ที่จริงแล้วมีข้อมูลลายน้ำถูกทำลายมากกว่าอัลกอริทึมที่ใช้ในวิทยานิพนธ์ แต่ความไม่ทนทานต่อการโจมตีดังกล่าวสามารถชดเชยโดยอาศัยรหัสแก้ไขความผิดพลาด ช่วยในการกู้ลายน้ำกลับคืนมา ชดเชยผลจากอัลกอริทึมฟังลายน้ำที่มีความทนทานต่ำ ในขณะที่อัลกอริทึมที่เสนอมีความทนทานต่อการโจมตีสูงโดยไม่ต้องอาศัยเทคนิคแก้ไขความผิดพลาดใดๆร่วมด้วยเลย ผลการทดสอบเปรียบเทียบดังกล่าวแสดงดังตารางที่ 5.10 ถึงตารางที่ 5.12

ตารางที่ 5.10 ผลการเปรียบเทียบจำนวนบิตที่ผิดพลาดในเพลง Till I get it right

	ATS	ไม่ใช่ ECC	ECC
No attack	0	2	0
Downsampling	0	3	0
LPF	0	93	18
Random noise	0	46	0
Gaussian noise	0	83	18
Denoise-DWT	0	0	0
MP3: 128 kbps	0	2	0
MP3: 56 kbps	0	2	0
Jittering	0	52	3
Crop (5×100)	0	7	0
Crop (10×100)	0	13	0
Crop (10×500)	7	28	0
Crop (10×1000)	16	53	4

ตารางที่ 5.11 ผลการเปรียบเทียบจำนวนบิตที่ผิดพลาดในเพลง Sunrise

	ATS	ไม่ใช่ ECC	ECC
No attack	0	1	0
Downsampling	0	22	0
LPF	0	79	2
Random noise	0	46	0
Gaussian noise	0	89	23
Denoise-DWT	0	0	0
MP3: 128 kbps	0	141	51
MP3: 56 kbps	7	141	51
Jittering	0	36	3
Crop (5×100)	0	5	0
Crop (10×100)	0	7	0
Crop (10×500)	10	28	0
Crop (10×1000)	12	50	4

ตารางที่ 5.12 ผลการเปรียบเทียบจำนวนบิตที่ผิดพลาดในเพลงฟลอร์เฟื่องฟ้า

	ATS	ไม่ใช่ ECC	ECC
No attack	0	0	0
Downsampling	0	7	0
LPF	0	41	1
Random noise	0	8	0
Gaussian noise	0	23	0
Denoise-DWT	0	0	0
MP3: 128 kbps	0	40	3
MP3: 56 kbps	0	40	3
Jittering	0	15	0
Crop (5×100)	0	3	0
Crop (10×100)	0	7	0
Crop (10×500)	3	22	0
Crop (10×1000)	9	46	0

บทที่ 6

สรุปผลการวิจัยและข้อเสนอแนะ

6.1 สรุปผลการวิจัย

งานวิจัยวิทยานิพนธ์นี้ศึกษาเกี่ยวกับการสร้างลายน้ำกับสัญญาณเสียงเพลงแบบกึ่งปิด โดยข้อสรุปของงานวิจัยได้แบ่งออกเป็น 4 ส่วนดังนี้คือ

1) การแยกองค์ประกอบของสัญญาณเสียงด้วยการแปลงเวฟเลตแบบเต็มหน่วย การเลือกชนิดของเวฟเลตที่มีสัมประสิทธิ์สเกลลิงฟิลเตอร์จำนวนต่าง ๆ กันมีผลต่อเวลาที่ใช้ในการคำนวณ ยิ่งมีมากจะใช้เวลาคำนวณมากส่งผลต่อความเร็วของการค้นหาแบบตามเชิงปรับตัว และเกินความจำเป็น อีกประการคือการเลือก subband ที่ได้จากการแยกองค์ประกอบด้วยการแปลงเวฟเลตแบบเต็มหน่วยเพื่อการฝังลายน้ำ ที่จำนวน subband สูงๆจะมีจำนวนสัมประสิทธิ์เวฟเลตลดลง ซึ่งในการฝังลายน้ำที่ subband ดังกล่าวสามารถลดเวลาในการคำนวณลงได้

2) วิธีการฝังลายน้ำ การฝังจะเลือกฝังข้อมูลลายน้ำลงบน approximate subband ของเวฟเลตเพื่อความทนทานและคุณภาพเสียงที่ดี แม้การเลือก detail subband จะส่งผลกระทบต่อคุณภาพเสียงน้อยกว่าแต่ลายน้ำจะไม่มี ความทนทาน ซึ่งในขั้นตอนการฝังข้อมูลลายน้ำจะฝังข้อมูลลายน้ำตลอดทั้ง subband เพื่อให้ข้อมูลลายน้ำกระจายทั่วทุกสัมประสิทธิ์เวฟเลต

3) การหาค่าความเข้มในการฝังด้วยการค้นหาแบบตามเชิงปรับตัว ในการฝังลายน้ำต้องคำนึงถึงความทนทานของลายน้ำร่วมกับคุณภาพเสียงควบคู่กันไป ทำให้ต้องคัดเลือกฟังก์ชันวัตถุประสงค์สองฟังก์ชันที่ขัดแย้งกันผ่านตัวประกอบถ่วงน้ำหนัก ผลลัพธ์ที่ได้จากฟังก์ชันวัตถุประสงค์ทั้งสองนี้ก่อน เพื่อหาค่าคงที่ α_1 , α_2 และ α_3 ที่เหมาะสมที่สุด ฟังก์ชันวัตถุประสงค์ของ ATS ในงานวิจัยวิทยานิพนธ์นี้ใช้ SNR, จีวัตคุณภาพเสียง และใช้ค่าสหสัมพันธ์จีวัตคุณภาพลายน้ำที่ตรวจจับได้ ซึ่งผลการทดสอบ ฟังก์ชันวัตถุประสงค์ทั้งสองให้ผลเป็นอย่างดี และสอดคล้องกับมาตรฐานของ IFPI เมื่อพิจารณาผลของฟังก์ชันวัตถุประสงค์ทั้งสอง ถ้า α มีค่ามาก ส่งผลให้มีค่าสหสัมพันธ์ ที่สูง ลายน้ำทนทานต่อการโจมตีแบบต่างๆ ได้ดี แต่คุณภาพของเสียงจะลดต่ำลง ส่งผลให้ SNR ต่ำเกินกว่ากำหนด ฟังก์ชันวัตถุประสงค์จะกระตุ้น ATS เพื่อค้นหาค่าความเข้มในการฝังลายน้ำใหม่ที่เหมาะสม ในทางกลับกัน α ที่ต่ำจะให้ค่า SNR สูง ลายน้ำส่งผลกระทบต่อคุณภาพของเสียงน้อย แต่มีความทนทานต่อการโจมตีด้อยลง ส่งผลให้มีค่าสหสัมพันธ์ที่ต่ำลงด้วย ถ้า ATS ขาด

ไปซึ่งฟังก์ชันวัตถุประสงค์ใดฟังก์ชันวัตถุประสงค์หนึ่ง ค่า α ทั้งหมดที่ค้นหาได้จะให้อัลกอริทึมการฟังลายน้ำไม่ได้ผลลัพธ์ตามที่คาดไว้

4) การค้นหาลายน้ำจากสัญญาณเสียงที่ผ่านการโจมตีแบบต่างๆ เนื่องจากการโจมตีทางการประมวลผลสัญญาณแบบต่างๆมีผลกระทบต่อสัญญาณเสียงที่แตกต่างกัน จึงจำเป็นต้องปรับปรุงสัญญาณเสียงหลังถูกโจมตีทางการประมวลผลสัญญาณบางแบบก่อนทำการตรวจจับลายน้ำ โดยศึกษาคุณลักษณะของเสียงที่เปลี่ยนไปหลังจากการโจมตีแต่ละแบบเพื่อเลือกกระบวนการปรับปรุงสัญญาณที่ถูกต้อง เนื่องจากการฟังลายน้ำแบบกึ่งปิด สัญญาณเสียงจำเป็นต้องปรับปรุงคุณภาพด้วยกระบวนการพื้นฐานทางการประมวลผลสัญญาณที่ปราศจากการใช้ข้อมูลของสัญญาณเสียงต้นฉบับ และเป็นกระบวนการที่ง่าย ไม่มีความซับซ้อน มีความเหมาะสมด้านเวลา

5) คุณภาพของสัญญาณเสียงหลังจากการฟังลายน้ำ อัลกอริทึมที่เสนอในวิทยานิพนธ์สามารถสร้างลายน้ำที่มีความทนทานลงบนสัญญาณเสียงโดยไม่ทำให้คุณภาพเสียงเมื่อทำการรับฟังเปลี่ยนไปจากต้นฉบับ สัญญาณเสียงยังคงมีคุณภาพจากการรับฟังที่ดีสามารถนำไปประยุกต์ใช้งานได้จริงเนื่องจากผู้รับฟังไม่สามารถล่วงรู้ถึงการมีอยู่ของลายน้ำได้

เสียงเพลงในแบบต่างๆทั้งไทยและสากลแบบโมโน ที่อัตราสุ่ม 44100 Hz 16 บิต จำนวน 30 ตัวอย่าง ถูกนำมาทดสอบการสร้างลายน้ำกับสัญญาณเสียงด้วยระเบียบวิธีที่นำเสนอในวิทยานิพนธ์ฉบับนี้ โดยแต่ละตัวอย่างจะผ่านการฟังลายน้ำและทำการทดสอบความทนทานจากการโจมตี 12 แบบ รวมจำนวนการทดสอบ 360 ผลลัพธ์ และนำมาเปรียบเทียบผลกับการฟังลายน้ำด้วยการศึกษาอัลกอริทึมที่มีการพัฒนาไว้ก่อนหน้านี้ รวมการทดสอบทั้งสองแบบเป็น 720 ผลลัพธ์ ผลการทดสอบทำให้สามารถสรุปได้ว่า งานวิจัยนี้สามารถให้การฟังลายน้ำกับสัญญาณเสียงที่มีความทนทานและคุณภาพเสียงที่ดี โดยมีค่าสหสัมพันธ์มากที่สุดเป็น 1 และน้อยที่สุดเป็น 0.9605 ผลจากการรับฟังสัญญาณเสียงที่ฟังลายน้ำเทียบกับเสียงต้นฉบับทั้งหมดทุกตัวอย่าง โดยผู้ทดสอบ 9 คน ให้ข้อสังเกตว่าคุณภาพเสียงเหมือนต้นฉบับถึง 94.81%

6.2 ข้อเสนอแนะ

- 1) ควรศึกษาเทคนิคเพิ่มเติมเกี่ยวกับการฟังลายน้ำกับสัญญาณเสียงพูด
- 2) ควรมีการศึกษาเพิ่มเติมในการเพิ่มความทนทานของลายน้ำ กรณีที่สัญญาณลายน้ำถูกโจมตีโดยการลบข้อมูลจำนวนมากบางส่วนไปอย่างถาวร แนวทางหนึ่งคือการฟังลายน้ำเฉพาะในส่วนที่สำคัญของสัญญาณเสียง
- 3) เพื่อความทนทานของลายน้ำ ควรเพิ่มการทำรหัสแก้ไขความผิดพลาดแบบต่างๆที่เหมาะสมเข้ากับลายน้ำ ก่อนทำการฟังลายน้ำกับสัญญาณเสียง

4) เพื่อความรวดเร็วในการค้นหาแบบตามเชิงปรับตัว ควรจะมีเงื่อนไขอื่นมาช่วยในการคัด
เลือกค่าเริ่มต้นสำหรับการค้นหาที่เหมาะสม โดยคำนึงถึงคุณลักษณะที่แตกต่างกันของสัญญาณ
เสียงในแต่ละตัวอย่าง

5) เพื่อเพิ่มคุณภาพของเสียง ควรพิจารณาผลกระทบของกฎเกณฑ์ที่เป็นไปได้ที่ให้คุณภาพ
ของเสียงดีที่สุด

รายการอ้างอิง

- ลัญจกร วุฒิสัทติกุลกิจ. (2546). เทคโนโลยีโทรคมนาคม ทฤษฎีข่าวสารและการเข้ารหัส. กรุงเทพฯ: สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย.
- ศรัญญา กาญจนรินทร์. (2547). คำถาม สิทธิบัตร และ ลิขสิทธิ์ [ออนไลน์]. ได้จาก: <http://www.marut.th.com/ans>.
- อาทิตย์ ศรีแก้ว. (2546). วิธีการค้นหาคำตอบที่ดีที่สุดแบบตาบ. เอกสารประกอบการเรียนการสอน รายวิชา 403626 Artificial intelligence for engineer. มหาวิทยาลัยเทคโนโลยีสุรนารี.
- Burrus, C. S., Gopinath, R. and Guo, H. (1998). **Introduction to wavelets and wavelet transforms a primer**. U.S.A: prentice-Hall.
- Cui, L., Wang, S. and Sun, T. (2003). The application of binary image in digital audio watermarking. **Neural Networks and Signal Processing**. 2: 1497 – 1500.
- Cui, L., Wang, S. and Sun, T. (2003). The application of wavelet analysis and audio compression technology in digital audio watermarking. **Neural Networks and Signal Processing**. 2: 1533 – 1537.
- Hanjalic, A. (2000). **Image & video database restoration, watermarking & retrieval**. U.S.A: prentice-Hall.
- Huang, J., Wang, Y. and Shi, Y.Q. (2002). A blind audio watermarking algorithm with self-synchronization. **IEEE International Symposium on Circuits and Systems 2002**. 3: 627 – 630.
- Li, W., Xue, X., Li, X. and Lu, P. (2003). A novel feature-based robust audio watermarking for copyright protection. **Information Technology: Coding and Computing [Computers and Communications]**. 554 – 558.
- Li, X., Zhang, M. and Sun, S. (2003). Adaptive audio watermarking algorithm based on SNR in wavelet domain. **Natural Language Processing and Knowledge Engineering**. 287 – 292.
- Mathworks, Inc. (2002). MATLAB (version 6.5.0.18091a Release 13) [Computer software]. Massachusetts .

Pcguide. (2005). **Compact disk digital audio (CD-DA)**. Avialble: <http://www.pcguides.com/ref/cd/formatCDDA-c.html>

Shao, Y., Wu, G. and Lin, X. (2001). A wavelet based adaptive watermarking algorithm. **Infotech and Info-net**. 3: 384 – 389.

Tu, R. and Zhao, J. (2003). A novel semi-fragile audio watermarking scheme. **Audio and Visual Environments and Their Applications**. 89 – 94.

Wu, J., and Xie, J. (2003). Blind wavelet-based watermarking scheme using fuzzy clustering theory. **Neural Networks and Signal Processing**. 2:1521 – 1524.

Xu, C., Wu, J., and Sun, Q. (1999). Digital Audio Watermarking and Its Application in Multimedia database. **Fifth International Symposium on Signal Processing and its Applications**. 91 - 94.

ภาคผนวก ก

รายละเอียดของเสียงเพลงที่นำมาทดสอบ

เสียงเพลงที่ใช้ในงานวิจัยวิทยานิพนธ์นี้ทั้งหมดมาจาก Audio CD แบบ 16 บิต 44 kHz จากนั้นจะทำการเก็บบันทึกลงคอมพิวเตอร์ที่ 16 บิต 44 kHz ในรูปแบบโมโน เพลงทั้งหมดแบ่งออกเป็นสองกลุ่มคือ กลุ่มเพลงสากลแบ่งออกเป็น classic, jazz, pop แบบละ 5 ตัวอย่าง ตัวอย่างละ 30 วินาที และกลุ่มเพลงไทยแบ่งออกเป็น ไทยเดิม ไทยสุนทราภรณ์ ลูกทุ่งแบบละ 5 ตัวอย่าง ตัวอย่างละ 30 วินาที ซึ่งรายละเอียดที่แสดงประกอบด้วย ชื่อบทเพลง ศิลปิน อัลบั้ม เจ้าของลิขสิทธิ์ และปีที่ผลิต

เพลงสากล classic

1. ชื่อเพลง *THE BEAUTIFUL BLUE DANUBE* โดย Strauss Johann Jr อัลบั้ม The Best of Semi Classics Vol.1 เจ้าของลิขสิทธิ์ APS Intermusic Co., Ltd. ปีที่ผลิต ไม่ระบุ
2. ชื่อเพลง *HUNGARIAN DANCES NO.5 IN F SHARP MAJOR* โดย Brahms อัลบั้ม The Best of Semi Classics Vol.1 เจ้าของลิขสิทธิ์ APS Intermusic Co., Ltd. ปีที่ผลิต ไม่ระบุ
3. ชื่อเพลง *THE MARRIAGE OF FIGARO* โดย Mozart อัลบั้ม The Best of Semi Classics Vol.1 เจ้าของลิขสิทธิ์ APS Intermusic Co., Ltd. ปีที่ผลิต ไม่ระบุ
4. ชื่อเพลง *MOONLIGHT* โดย Beethoven อัลบั้ม The Best of Semi Classics Vol.1 เจ้าของลิขสิทธิ์ APS Intermusic Co., Ltd. ปีที่ผลิต ไม่ระบุ
5. ชื่อเพลง *ROMANCE FOR VIOLIN & ORCHESTRA NO.1 IN G MAJOR* โดย Beethoven อัลบั้ม The Best of Semi Classics Vol.1 เจ้าของลิขสิทธิ์ APS Intermusic Co., Ltd. ปีที่ผลิต ไม่ระบุ

เพลงสากล jazz

1. ชื่อเพลง *BEAUTY IN ME* โดย Mark Hodgins อัลบั้ม Jazz in The Park 3 เจ้าของลิขสิทธิ์ Metro Records-Taps (1981) CO., LTD. ปีที่ผลิต ไม่ระบุ
2. ชื่อเพลง *SHADE OF PAIN* โดย Mark Hodgins อัลบั้ม Jazz in The Park 3 เจ้าของลิขสิทธิ์ Metro Records-Taps (1981) CO., LTD. ปีที่ผลิต ไม่ระบุ
3. ชื่อเพลง *'TILL I GET IT RIGHT* โดย Kirk Whalum อัลบั้ม Romantic Jazz เจ้าของลิขสิทธิ์ Sony Music ปีที่ผลิต ค.ศ. 1999
4. ชื่อเพลง *I COULD WRITE A BOOK* โดย Betty Carter & Ray Bryant อัลบั้ม Romantic Jazz เจ้าของลิขสิทธิ์ Sony Music ปีที่ผลิต ค.ศ. 1999

5. ชื่อเพลง *SQUEEZE ME* โดย Louis Armstrong อัลบั้ม Romantic Jazz เจ้าของลิขสิทธิ์ Sony Music ปีที่ผลิต ค.ศ. 1999

เพลงสากล pop

1. ชื่อเพลง *VINCENT* โดย Don Mclean อัลบั้ม Country Special hits เจ้าของลิขสิทธิ์ APS INTERMUSIC CO., LTD. ปีที่ผลิต ไม่ระบุ
2. ชื่อเพลง *WHITE FLAG* โดย Dido อัลบั้ม Super fresh เจ้าของลิขสิทธิ์ Sony Music ปีที่ผลิต ค.ศ. 2003
3. ชื่อเพลง *SUNRISE* โดย Simply red อัลบั้ม A Gift of Music เจ้าของลิขสิทธิ์ Universal Music (Thailand) ปีที่ผลิต ค.ศ. 2003
4. ชื่อเพลง *SOMETIME* โดย Grabielle อัลบั้ม A Gift of Music เจ้าของลิขสิทธิ์ Universal Music(Thailand) ปีที่ผลิต ค.ศ. 2003
5. ชื่อเพลง *WHERE IS THE LOVE* โดย Back Eyes Pea อัลบั้ม A Gift of Music เจ้าของลิขสิทธิ์ Universal Music (Thailand) ปีที่ผลิต ค.ศ. 2003

เพลงไทยเดิม

1. ชื่อเพลง *วิวาห์พระสมุทร* โดย คณะซวงโคศลปี่ อัลบั้ม เพลงสำหรับพิธีแต่งงาน เจ้าของลิขสิทธิ์ หจก.ว.มิวสิกชาวด์ ปีที่ผลิต พ.ศ. 2542
2. ชื่อเพลง *เทวปรีชาธิ* โดย คณะซวงโคศลปี่ อัลบั้ม เพลงสำหรับพิธีแต่งงาน เจ้าของลิขสิทธิ์ หจก.ว.มิวสิกชาวด์ ปีที่ผลิต พ.ศ. 2542
3. ชื่อเพลง *โพธิสัตว์* โดย คณะซวงโคศลปี่ อัลบั้ม เพลงสำหรับพิธีแต่งงาน เจ้าของลิขสิทธิ์ หจก.ว.มิวสิกชาวด์ ปีที่ผลิต พ.ศ. 2542
4. ชื่อเพลง *มหาฤกษ์* โดย คณะซวงโคศลปี่ อัลบั้ม เพลงสำหรับพิธีแต่งงาน เจ้าของลิขสิทธิ์ หจก.ว.มิวสิกชาวด์ ปีที่ผลิต พ.ศ. 2542
5. ชื่อเพลง *มหาชัย* โดย คณะซวงโคศลปี่ อัลบั้ม เพลงสำหรับพิธีแต่งงาน เจ้าของลิขสิทธิ์ หจก.ว.มิวสิกชาวด์ ปีที่ผลิต พ.ศ. 2542

เพลงไทยสุนทราภรณ์

1. ชื่อเพลง *เย็นลมว่าว* โดย ธรรมรัตน์ อัลบั้ม ดาวจรัสฟ้าสุนทราภรณ์ 2 เจ้าของลิขสิทธิ์ พีรพล มนต์พิชิต ปีที่ผลิต พ.ศ. 2546

2. ชื่อเพลง *พรหมลิขิต* โดย ธรรมรัตน์ อัลบั้ม ดาวจรัสฟ้าสุทธาภรณ์ 2 เจ้าของลิขสิทธิ์ พีรพล มนต์พิชิต ปีที่ผลิต พ.ศ. 2546
3. ชื่อเพลง *พลอร์เพื่องฟ้า* โดย ธรรมรัตน์ อัลบั้ม ดาวจรัสฟ้าสุทธาภรณ์ 2 เจ้าของลิขสิทธิ์ พีรพล มนต์พิชิต ปีที่ผลิต พ.ศ. 2546
4. ชื่อเพลง *ตัดสวาท* โดย ดาวใจ ไพจิตร อัลบั้ม ดาวจรัสฟ้าสุทธาภรณ์ 2 เจ้าของลิขสิทธิ์ พีรพล มนต์พิชิต ปีที่ผลิต พ.ศ. 2546
5. ชื่อเพลง *ขวัญใจเจ้าทวย* โดย โฉมฉาย อัลบั้ม ดาวจรัสฟ้าสุทธาภรณ์ 2 เจ้าของลิขสิทธิ์ พีรพล มนต์พิชิต ปีที่ผลิต พ.ศ. 2546

เพลงไทยลูกทุ่ง

1. ชื่อเพลง *เหน้อยไหมคนดี* โดย ไมค์ ภิรมย์พร อัลบั้ม เหน้อยไหมคนดี 2 เจ้าของลิขสิทธิ์ บริษัทแกรมมี่เอนเตอร์เทนเมนต์ จำกัด (มหาชน) ปีที่ผลิต พ.ศ. 2543
2. ชื่อเพลง *พรหมลิขิต* โดย ธรรมรัตน์ อัลบั้ม ในฝัน เจ้าของลิขสิทธิ์ บริษัท แกรมมี่เอนเตอร์เทนเมนต์ จำกัด (มหาชน) ปีที่ผลิต พ.ศ. 2539
3. ชื่อเพลง *สร้างวิมาน* โดย ยอดรัก สลักใจ อัลบั้ม ฮิตที่สุด เจ้าของลิขสิทธิ์ พีรพล มนต์พิชิต ปีที่ผลิต พ.ศ. 2547
4. ชื่อเพลง *สาวฝั่งโขง* โดย ปอง ปรีดา อัลบั้ม ฮิตที่สุด เจ้าของลิขสิทธิ์ พีรพล มนต์พิชิต ปีที่ผลิต พ.ศ. 2547
5. ชื่อเพลง *ลิบหกลีปแห่งความหลัง* โดย ยอดรัก สลักใจ อัลบั้ม ฮิตที่สุด เจ้าของลิขสิทธิ์ พีรพล มนต์พิชิต ปีที่ผลิต พ.ศ. 2547

ภาคผนวก ข

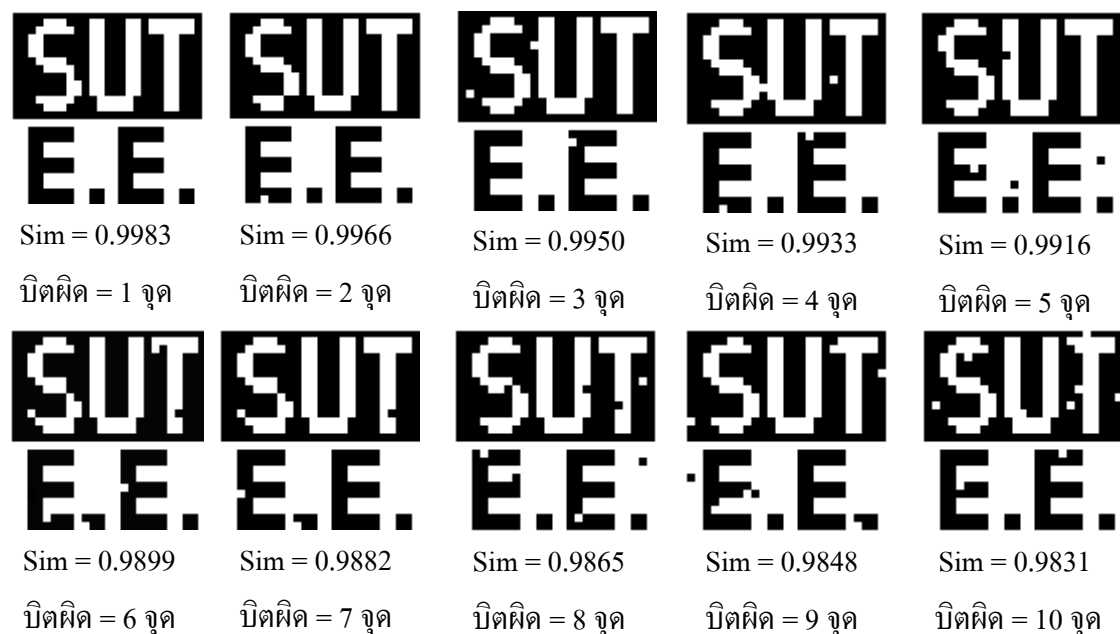
รูปภาพลายน้ำ

ข้อมูลลายน้ำเป็นรูปภาพอักษรภาษาอังกฤษแบบขาวดำขนาด 25×25 พิกเซล ส่วนบนเป็น “SUT” สีขาวบนพื้นดำ เป็นตัวย่อของ Suranaree University of Technology ส่วนล่างเป็น “E.E.” สีดำบนพื้นสีขาวภาษาอังกฤษ เป็นตัวย่อของ Electrical Engineering ดังรูป ข.1



รูปที่ ข.1 รูปภาพที่ใช้เป็นลายน้ำ

ลายน้ำที่ตรวจจับได้จากสัญญาณเสียงที่ฝังลายน้ำถูกโจมตีทางการประมวลผลสัญญาณจะเกิดบิตที่ผิดพลาดขึ้นดังรูป ข.2 แสดงจำนวนบิตที่ผิดพลาดและค่า Sim ที่สัมพันธ์กับรูปนั้นๆ



รูปที่ ข.2 ตัวอย่างภาพลายน้ำที่ตรวจจับได้จากสัญญาณเสียงหลังการโจมตี

				
Sim = 0.9815	Sim = 0.9797	Sim = 0.9782	Sim = 0.9762	Sim = 0.9744
บิตผิด = 11 จุด	บิตผิด = 12 จุด	บิตผิด = 13 จุด	บิตผิด = 14 จุด	บิตผิด = 15 จุด
				
Sim = 0.9730	Sim = 0.9710	Sim = 0.9693	Sim = 0.9675	Sim = 0.9662
บิตผิด = 16 จุด	บิตผิด = 17 จุด	บิตผิด = 18 จุด	บิตผิด = 19 จุด	บิตผิด = 20 จุด
				
Sim = 0.9640	Sim = 0.9623	Sim = 0.9605	Sim = 0.9596	Sim = 0.9580
บิตผิด = 21 จุด	บิตผิด = 22 จุด	บิตผิด = 23 จุด	บิตผิด = 24 จุด	บิตผิด = 25 จุด
				
Sim = 0.9567	Sim = 0.9548	Sim = 0.9530	Sim = 0.9514	Sim = 0.9500
บิตผิด = 26 จุด	บิตผิด = 27 จุด	บิตผิด = 28 จุด	บิตผิด = 29 จุด	บิตผิด = 30 จุด
				
Sim = 0.9815	Sim = 0.9797	Sim = 0.9782	Sim = 0.9762	Sim = 0.9744
บิตผิด = 11 จุด	บิตผิด = 12 จุด	บิตผิด = 13 จุด	บิตผิด = 14 จุด	บิตผิด = 15 จุด

รูปที่ ข.2 ตัวอย่างภาพลายน้ำที่ตรวจจับได้จากสัญญาณเสียงหลังการโจมตี(ต่อ)

ภาคผนวก ค

โปรแกรม MATLAB ที่ใช้งานวิจัยวิทยานิพนธ์

ส่วนโปรแกรมหลักที่ใช้ฝังลายน้ำด้วยการแปลงDWT

```
%.....  
%      Audio Watermarking Algorithm : Embedding  
%      Author: Noppanan Sriyingyong  
%      Issue Date: Dec.2005  
%.....
```

```
clear,close all,clc, tic
```

```
[y,Fs1,bit1] = wavread('6.Romance Violin_30s44.wav') ;
```

```
y1=rot90(y(:,1),1);
```

```
sy1=length(y1);
```

```
wavwrite(y1,Fs1,bit1,'original');
```

```
% international music
```

```
% 1.blue danube(or) 30s44.wav 2.Hungarian Dance 05(or) 30s44.wav
```

```
% 4.Marriage of figaro(or) 30s44.wav 5.Moonlight_30s44.wav 6.Romance Violin_30s44.wav
```

```
% 8.Dido_30s44.wav 1.simply red 30s.wav 2.girl2.wav
```

```
% 3.Where is the love 30s.wav 4.Vincent_30s44.wav
```

```
% Beauty in me 30s.wav Shade of pain 30s.wav Squeeze me 30s3.wav
```

```
% I could write a book 30s2.wav Till i get it right 30s.wav
```

```
% thai music
```

```
% Mike_30s44.wav Sweet paradise 30s.WAV 16 30s.wav
```

```
% GOT 30s.wav Maekhog girl 30s.wav
```

```
% MAHA RUERK 30s.wav VIVA PRASAMUT 30s.wav TAEVA PRASIT 30s.wav
```

```
% PHO DHI SAT 30s.wav MAHA CHAI 30s.wav
```

```
% Kite wind 30s.WAV Fate 30s.WAV Bougainvillaea floor02 30s.WAV
```

```
% Cast off 30s.WAV Tui 30s.WAV
```

```

alpha=[];
p=imread('sut25.jpg');
pic=double(im2bw(p));
[row,col]=size(pic);
vc=reshape(pic,1,row*col);
rand('state',16);
wp=randperm(row*col);

for i=1:1:row*col
    v(i)=vc(wp(i));
end

for i=1:1:length(v)
    if v(i)==1
        w(i)=1;
    else
        w(i)=-1;
    end
end

level=5;wv='db4';

[C,L] = wavedec(y1,level,wv);
ca5 = appcoef(C,L,wv,level);
[cd1,cd2,cd3,cd4,cd5]=detcoef(C,L,[1,2,3,4,5]);

sc=length(ca5);
seg=row*col;
seg_w =fix(sc/seg);

rand('state',28);
cp=randperm(seg);

```

```

ca_reshape=reshape(ca5(1:seg*seg_w),seg_w,seg);
ca_reshape=ca_reshape';

for i=1:1:seg
    ca(i,:)=ca_reshape(cp(i,:));
end

for i=1:1:seg
    mean_ca(i)=mean(ca(i,:));
    c_mean(i,:)=ca(i,:)-mean_ca(i);
    if abs(mean_ca(i))< .001
        caw(i,:)=c_mean(i,:)+(w(i)*alpha(1)*abs(mean_ca(i)));
    else if abs(mean_ca(i))< .005
        caw(i,:)=c_mean(i,:)+(w(i)*alpha(2)*abs(mean_ca(i)));
    else if abs(mean_ca(i))< .01
        caw(i,:)=c_mean(i,:)+(w(i)*alpha(3)*abs(mean_ca(i)));
    else
        caw(i,:)=c_mean(i,:)+(w(i)*1*abs(mean_ca(i)));
    end
end
end
end

for i=1:1:seg
    c_new(cp(i,:))=caw(i,:);
end

c_new=reshape(c_new',1,seg_w*seg);
c_new((seg*seg_w)+1:sc)=ca5((seg*seg_w)+1:sc);
Cnew = [c_new cd5 cd4 cd3 cd2 cd1];
audio = waverec(Cnew,L,wv);

```

```
wavwrite(audio,Fs1,bit1,'music_emb'); toc
```

ส่วนโปรแกรมหลักที่ใช้ตรวจจับลายน้ำจากการโจมตีแบบต่างๆ

```
%.....
%      Audio Watermarking Algorithm : Detection
%      Author: Noppanan Sriyingyong
%      Issue Date: Dec.2005
%.....
```

```
%1) Original
```

```
[y2,Fs2,bit2] = wavread('music_emb.wav');
```

```
audiox=rot90(y2(:,1),1);
```

```
[Cx,Lx] = wavedec(audiox,level,wv);
```

```
cax5 = appcoef(Cx,Lx,wv,level);
```

```
cax_reshape = reshape(cax5(1:seg*seg_w),seg_w,seg);
```

```
cax_reshape = cax_reshape';
```

```
for i=1:1:seg
```

```
    caw(i,:)=cax_reshape(cp(i,:));
```

```
    if sign(mean(caw(i,:)))>0
```

```
        noisecode(i)=1;
```

```
    else
```

```
        noisecode(i)=0;
```

```
    end
```

```
end
```

```
SIM = sum(noisecode.*v)/((sqrt(sum(noisecode.^2)))*(sqrt(sum(v.^2))));
```

```
% 2) Resampling
```

```
[y2,Fs2,bit2] = wavread('music_emb_PCM_PCM.wav');
```

```

audiox=rot90(y2(:,1),1);
audiox=audiox(6:max(size(audiox)));

[Cx,Lx] = wavedec(audiox,level,wv);
cax5 = appcoef(Cx,Lx,wv,level);
cax_reshape = reshape(cax5(1:seg*seg_w),seg_w,seg);
cax_reshape = cax_reshape';

for i=1:1:seg
    caw(i,:)=cax_reshape(cp(i,:));
    if sign(mean(caw(i,:)))>0
        noisecode(i)=1;
    else
        noisecode(i)=0;
    end
end

SIM = sum(noisecode.*v)/sqrt(sum(noisecode.^2)*sum(v.^2));

% 3) LPF
[y2,Fs2,bit2] = wavread('music_emb') ;
y2=rot90(y2(:,1),1);
[b,a] = butter(6,6000/(Fs2/2));
audiox=filter(b,a,y2);

[Cx,Lx] = wavedec(audiox,level,wv);
cax5 = appcoef(Cx,Lx,wv,level);
cax_reshape = reshape(cax5(1:seg*seg_w),seg_w,seg);
cax_reshape = cax_reshape';

for i=1:1:seg

```

```

caw(i,:)=cax_reshape(cp(i,:));
if sign(mean(caw(i,:)))>0
    noisecode(i)=1;
else
    noisecode(i)=0;
end
end

wavwrite(audio,Fs1,bit1,'music_LPF');
SIM = sum(noisecode.*v)/sqrt(sum(noisecode.^2)*sum(v.^2));

% 4) Random noise
[y2,Fs2,bit2] = wavread('music_emb') ;
y2=rot90(y2,1);
noise=rand(1,sy1);
noise2=sqrt(0.01*sum(y2.^2)/sum(noise.^2))*noise;
audiox=(y2+noise2);

[Cx,Lx] = wavedec(audiox,level,wv);
[thr,sorh,keepapp]= ddencmp('den','wv',audiox);
audiox = wdencmp('gbl',Cx,Lx,wv,level,thr,sorh,keepapp);
[Cx,Lx] = wavedec(audiox,level,wv);
cax5 = appcoef(Cx,Lx,wv,level);
cax5 = cax5-mean(cax5);
cax_reshape = reshape(cax5(1:seg*seg_w),seg_w,seg);
cax_reshape = cax_reshape';

for i=1:1:seg
    caw(i,:)=cax_reshape(cp(i,:));
    if sign(mean(caw(i,:)))>0
        noisecode(i)=1;
    end
end

```

```

else
    noisecode(i)=0;
end
end

wavwrite(audio,Fs1,bit1,'music_random');
SIM = sum(noisecode.*v)/sqrt(sum(noisecode.^2)*sum(v.^2));

% 5) Gaussian white noise
[y2,Fs2,bit2] = wavread('music_emb');
y2=rot90(y2,1);
for adj=1:1:20
    audiox=awgn(y2,adj+29);
    ratio_noise(adj)=sum((audiox-y2).^2)/sum(y2.^2);
    error_ratio(adj)=abs(.01-ratio_noise(adj));
end
[mag,pos]=min(error_ratio);
audiox=awgn(y2,pos+29);

[Cx,Lx] = wavedec(audiox,level,wv);
cax5 = appcoef(Cx,Lx,wv,level);
cax_reshape = reshape(cax5(1:seg*seg_w),seg_w,seg);
cax_reshape = cax_reshape';

for i=1:1:seg
    caw(i,:)=cax_reshape(cp(i,:));
    if sign(mean(caw(i,:)))>0
        noisecode(i)=1;
    else
        noisecode(i)=0;
    end
end

```

```

end

wavwrite(audio,Fs1,bit1,'music_awgn');

SIM = sum(noisecode.*v)/sqrt(sum(noisecode.^2)*sum(v.^2));

% 6) DWT

[y2,Fs2,bit2] = wavread('music_emb.wav');
audiox=rot90(y2(:,1),1);

[Cx,Lx] = wavedec(audiox,level,wv);
[thr,sorh,keepapp]= ddencmp('den','wv',audiox);
audiox = wdencomp('gbl',Cx,Lx,wv,level,thr,sorh,keepapp);

[Cx,Lx] = wavedec(audiox,level,wv);
cax5 = appcoef(Cx,Lx,wv,level);
cax_reshape = reshape(cax5(1:seg*seg_w),seg_w,seg);
cax_reshape = cax_reshape';

for i=1:1:seg
    caw(i,:)=cax_reshape(cp(i,:));
    if sign(mean(caw(i,:)))>0
        noisecode(i)=1;
    else
        noisecode(i)=0;
    end
end

wavwrite(audio,Fs1,bit1,'music_dwt');

SIM = sum(noisecode.*v)/sqrt(sum(noisecode.^2)*sum(v.^2));

%7) MP3

```

```

[y2,Fs2,bit2] = wavread('music_emb_56_PCM.wav') ;
audiox=rot90(y2(:,1),1);
audiox=audiox(1106:max(size(audiox)));

[Cx,Lx] = wavedec(audiox,level,wv);
cax5 = appcoef(Cx,Lx,wv,level);
cax_reshape = reshape(cax5(1:seg*seg_w),seg_w,seg);
cax_reshape = cax_reshape';

for i=1:1:seg
    caw(i,:)=cax_reshape(cp(i,:));
    if sign(mean(caw(i,:)))>0
        noisecode(i)=1;
    else
        noisecode(i)=0;
    end
end

SIM = sum(noisecode.*v)/sqrt(sum(noisecode.^2)*sum(v.^2));

% 8) Jitter
[y2,Fs2,bit2] = wavread('music_emb') ;
audiox=rot90(y2(:,1),1);
sy2=length(audiox);
for i=1:1:fix(sy2/100)
    pos = randint(1,1,[1,100]);
    audiox((100*(i-1))+pos)=0;
end

[Cx,Lx] = wavedec(audiox,level,wv);
cax5 = appcoef(Cx,Lx,wv,level);

```

```

cax_reshape = reshape(cax5(1:seg*seg_w),seg_w,seg);
cax_reshape = cax_reshape';

for i=1:1:seg
    caw(i,:)=cax_reshape(cp(i,:));
    if sign(mean(caw(i,:)))>0
        noisecode(i)=1;
    else
        noisecode(i)=0;
    end
end

wavwrite(audio,Fs1,bit1,'music_jitter');

SIM = sum(noisecode.*v)/sqrt(sum(noisecode.^2)*sum(v.^2));

% 9) Crop
[y2,Fs2,bit2] = wavread('music_emb');
y2=rot90(y2(:,1),1);
sy2=length(y2);
rand_pos=5;
rand_samp=500;
pos = randint(1,rand_pos,[1,fix(sy2/rand_samp)]);
for i=1:1:rand_pos
    y2((pos(i)*rand_samp)-(rand_samp-1):(pos(i)*rand_samp))=0;
end
audiox=y2;

[Cx,Lx] = wavedec(audiox,level,wv);
cax5 = appcoef(Cx,Lx,wv,level);
cax_reshape = reshape(cax5(1:seg*seg_w),seg_w,seg);
cax_reshape = cax_reshape';

```

```

for i=1:1:seg
    caw(i,:)=cax_reshape(cp(i,:),:);
    if sign(mean(caw(i,:)))>0
        noisecode(i)=1;
    else
        noisecode(i)=0;
    end
end

wavwrite(audio,Fs1,bit1,'music_5_500');
SIM = sum(noisecode.*v)/sqrt(sum(noisecode.^2)*sum(v.^2));

```

ส่วนโปรแกรมหลักการค้นหาแบบตามูเชิงปรับตัว

```

%.....
%      Audio Watermarking Algorithm : ATS
%      Author: Noppanan Sriyingyong
%      Issue Date: Dec.2005
%.....

```

```
clear,close all,clc,tic
```

```

[y,Fs1,bit1] = wavread('6.Romance Violin_30s44.wav ');
y1=rot90(y(:,1),1);
sy1=length(y1);

```

```

p=imread('sut25.jpg');
pic=double(im2bw(p));
[row,col]=size(pic);
vc=reshape(pic,1,row*col);
rand('state',16);
wp=randperm(row*col);

```

```

for i=1:1:row*col
    v(i)=vc(wp(i));
end

for i=1:1:length(v)
    if v(i)==1
        w(i)=1;
    else
        w(i)=-1;
    end
end
end
level=5;wv='db4';

[C,L] = wavedec(y1,level,wv);
ca5 = appcoef(C,L,wv,level);
[cd1,cd2,cd3,cd4,cd5]=detcoef(C,L,[1,2,3,4,5]);

sc=length(ca5);seg=row*col;
seg_w =fix(sc/seg);

rand('state',28);cp=randperm(seg);
ca_reshape=reshape(ca5(1:seg*seg_w),seg_w,seg);
ca_reshape=ca_reshape';

for i=1:1:seg
    ca(i,:)=ca_reshape(cp(i,:),:);
end

for i=1:1:seg
    mean_ca(i)=mean(ca(i,:));
    c_mean(i,:)=ca(i,:)-mean_ca(i);

```

```

    cax(i,:)=c_mean(i,:)+(w(i)*abs(mean_ca(i)));
end

for i=1:1:seg
    c_new(cp(i,:))=cax(i,:);
end

c_new=reshape(c_new',1,seg_w*seg);
c_new((seg*seg_w)+1:sc)=ca5((seg*seg_w)+1:sc);
Cnew = [c_new cd5 cd4 cd3 cd2 cd1];
audio = waverec(Cnew,L,wv);
Audio_snr= 10*log10(sum(y1.^2)/sum((y1-audio).^2));

% Start TABU

max_error=1e-5;
N=1;snr_thre=23;
x1limit=[1;10];
x2limit=[1;10];
x3limit=[1;10];
for r=1:300
    S(r,1)=((x1limit(2)-x1limit(1)).*rand(1,N))+x1limit(1);
    S(r,2)=((x2limit(2)-x2limit(1)).*rand(1,N))+x2limit(1);
    S(r,3)=((x3limit(2)-x3limit(1)).*rand(1,N))+x3limit(1);
    while S(r,1)<S(r,2) | S(r,2)<S(r,3) | S(r,1)<S(r,3)
        S(r,1)=((x1limit(2)-x1limit(1)).*rand(1,N))+x1limit(1);
        S(r,2)=((x2limit(2)-x2limit(1)).*rand(1,N))+x2limit(1);
        S(r,3)=((x3limit(2)-x3limit(1)).*rand(1,N))+x3limit(1);
    end
end
end

```

```

format long;
for k=1:size(S,1)
error=Obj_2D23(S(k,1),S(k,2),S(k,3),snr_thre,mean_ca,ca5,c_mean,seg,seg_w,w,v,wv,cd5,cd4,c
d3,cd2,cd1,L,y1,cp,sc);%cal 5 value from early step then keep value
errorvalue(k,1)=error;k
end

[best_error,index]=min(errorvalue);
S0=S(index,:);
max_count=100;
best_neighbor=S0;
radius=0.25;f_radius=radius;

overall_best_error=best_error;
overall_neighbor=best_neighbor;
n=0;
t=0;
tt=0;
count=0;
n_back_tracking=0;

t=t+1;
tt=tt+1;
local(t,1)=count;
local(t,2:4)=best_neighbor;
local(t,5)=best_error;

tabu_list(tt,1)=count;
tabu_list(tt,2:4)=best_neighbor;
tabu_list(tt,5)=best_error;

```

```

ttt=1;

best_error_list(ttt,1)=count;
best_error_list(ttt,2:4)=best_neighbor;
best_error_list(ttt,5)=best_error;


if 2*best_error<=max_error
    radius=f_radius*1e-5;fprintf('adap0\n')
else if 2*best_error<=1e-4
    radius=f_radius*1e-4;fprintf('adap1\n')
else if 2*best_error<=1e-3
    radius=f_radius*1e-3;fprintf('adap2\n')
else if 2*best_error<=1e-2
    radius=f_radius*1e-2;fprintf('adap3\n')
else if 2*best_error<=1e-1
    radius=f_radius*1e-1;fprintf('adap4\n')
end
end
end
end
end

disp([best_error_list])


% Loop Start


display(' Loop Start ');


for count=1:max_count
    Number_neighb=20;
    S1=random_neigh2D23(Number_neighb,radius,x1limit,x2limit,x3limit,S0);

```

```
[best_error1,best_neighbor1,best_error,best_neighbor]=Objective_2D23(S1,best_error,best_neighbor,snr_thre,mean_ca,ca5,c_mean,seg,seg_w,w,v,wv,cd5,cd4,cd3,cd2,cd1,L,y1,cp,sc);
```

```
tt=tt+1;
```

```
tabu_list(tt,1)=count;
```

```
tabu_list(tt,2:4)=best_neighbor1;
```

```
tabu_list(tt,5)=best_error1;
```

```
ttt= ttt+1;
```

```
best_error_list(ttt,1)=count;
```

```
best_error_list(ttt,2:4)=best_neighbor;
```

```
best_error_list(ttt,5)=best_error;
```

```
disp([count best_neighbor1 best_error1*2])
```

```
if (2*best_error<max_error)
```

```
fprintf('best_error<max_error\n');
```

```
t=t+1;
```

```
tt=tt+1;
```

```
local(t,1)=count;
```

```
local(t,2:4)=best_neighbor;
```

```
local(t,5)=best_error;
```

```
break;
```

```
end
```

```
if 2*best_error1<=max_error
```

```
radius=f_radius*1e-5;fprintf('adap0\n')
```

```
else if 2*best_error1<=1e-4
```

```
radius=f_radius*1e-4;fprintf('adap1\n')
```

```
else if 2*best_error1<=1e-3
```

```
radius=f_radius*1e-3;fprintf('adap2\n')
```

```

else if 2*best_error1<=1e-2
    radius=f_radius*1e-2;fprintf('adap3\n')
else if 2*best_error1<=1e-1
    radius=f_radius*1e-1;fprintf('adap4\n')
end
end
end
end
end

% Back_tracking
if (best_error1-best_error)>1e-18
    n=n+1;
else
    n=0;
end

if count>3
    if n>=3
        disp('Back tracking');
        n_back_tracking=n_back_tracking+1;
        TEMP=tabu_list(count-3:count+1,:);

        [MAX,INDEX] = max(TEMP(:,5));
        RANK(5,:) = TEMP(INDEX,:);
        TEMP(INDEX,4)= 0;

        [MAX,INDEX] = max(TEMP(:,5));
        RANK(5,:) = TEMP(INDEX,:);
        TEMP(INDEX,4)= 0;
    end
end

```

```
[MAX,INDEX] = max(TEMP(:,5));
RANK(4,:) = TEMP(INDEX,:);
TEMP(INDEX,4)= 0;
```

```
[MAX,INDEX] = max(TEMP(:,5));
RANK(3,:) = TEMP(INDEX,:);
TEMP(INDEX,4)= 0;
```

```
[MAX,INDEX] = max(TEMP(:,5));
RANK(2,:) = TEMP(INDEX,:);
TEMP(INDEX,4)= 0;
```

```
[MAX,INDEX] = max(TEMP(:,5));
RANK(1,:) = TEMP(INDEX,:);
TEMP(INDEX,4)= 0;
```

```
neighbor=RANK(5,2:4);
S0=neighbor;
```

```
if best_error<overall_best_error
    overall_best_error=best_error;
    overall_best_neighbor=best_neighbor;
```

```
t=t+1;
local(t,1)=count;
local(t,2:4)=best_neighbor;
local(t,5)=best_error;
```

```
end
```

```

        best_error=RANK(5,4);
        n=0;
    else
        S0=best_neighbor;
        best_error=best_error;
    end
end

end

end

if overall_best_error<best_error
    best_error=overall_best_error;
    best_neighbor=overall_best_neighbor;
end

alpha=[best_neighbor];

for i=1:1:seg
    if abs(mean_ca(i))< .001
        cax2(i,:)=c_mean(i,:)+(w(i)*alpha(1)*abs(mean_ca(i)));
    else if abs(mean_ca(i))< .005
        cax2(i,:)=c_mean(i,:)+(w(i)*alpha(2)*abs(mean_ca(i)));
    else if abs(mean_ca(i))< .01
        cax2(i,:)=c_mean(i,:)+(w(i)*alpha(3)*abs(mean_ca(i)));
    else
        cax2(i,:)=c_mean(i,:)+(w(i)*1*abs(mean_ca(i)));
    end
end
end
end
end

```

```

for i=1:1:seg
    c_new2(cp(i),:)=cax2(i,:);
end

c_new=reshape(c_new2',1,seg_w*seg);
c_new((seg*seg_w)+1:sc)=ca5((seg*seg_w)+1:sc);
Cnew = [c_new cd5 cd4 cd3 cd2 cd1];
sound = waverec(Cnew,L,wv);
wavwrite(sound,Fs1,bit1,'music_emb');
ERR(1,1)= 10*log10(sum(y1.^2)/sum((y1-sound).^2));

[y2,Fs2,bit2] = wavread('music_emb.wav') ;
audiox=rot90(y2(:,1),1);
[Cx,Lx] = wavedec(audiox,level,wv);
cax5 = appcoef(Cx,Lx,wv,level);
cax_reshape=reshape(cax5(1:seg*seg_w),seg_w,seg);
cax_reshape=cax_reshape';

for i=1:1:seg
    caw(i,:)=cax_reshape(cp(i),:);
    if sign(mean(caw(i,:)))>0
        noisecode(i)=1;
    else
        noisecode(i)=0;
    end
end
end

ERR(1,2)= sum(noisecode.*v)/((sqrt(sum(noisecode.^2)))*(sqrt(sum(v.^2))));
openvar('ERR');

```

ส่วนโปรแกรมย่อยต่างๆของการค้นหาแบบตามูเชิงปรับตัว

```
%.....
%      Audio Watermarking Algorithm : random_neigh2D23/ATS
%      Author: Noppanan Sriyingyong
%      Issue Date: Dec.2005
%.....
```

```
function S1=random_neigh2D23(Number_neighb,radius,x1limit,x2limit,x3limit,S0)
```

```
for n=1:Number_neighb
```

```
    S1(n,1)=S0(1,1)+(radius*abs(x1limit(2)-x1limit(1))*rand1(-1,1));
```

```
    S1(n,2)=S0(1,2)+(radius*abs(x2limit(2)-x2limit(1))*rand1(-1,1));
```

```
    S1(n,3)=S0(1,3)+(radius*abs(x3limit(2)-x3limit(1))*rand1(-1,1));
```

```
    while S1(n,1)<x1limit(1) | S1(n,1)>x1limit(2) |...
```

```
        S1(n,2)<x2limit(1) | S1(n,2)>x2limit(2) |...
```

```
        S1(n,3)<x3limit(1) | S1(n,3)>x3limit(2)
```

```
        S1(n,1)=S0(1,1)+(radius*abs(x1limit(2)-x1limit(1))*rand1(-1,1));
```

```
        S1(n,2)=S0(1,2)+(radius*abs(x2limit(2)-x2limit(1))*rand1(-1,1));
```

```
        S1(n,3)=S0(1,3)+(radius*abs(x3limit(2)-x3limit(1))*rand1(-1,1));
```

```
    end
```

```
end
```

```
%.....
%      Audio Watermarking Algorithm : Obj_2D23/ATS
%      Author: Noppanan Sriyingyong
%      Issue Date: Dec.2005
%.....
```

```
function [error] = Obj_2D23(x1,x2,x3,snr_thre,mean_ca,ca5,c_mean,seg,seg_w,w,v,wv,
cd5,cd4,cd3,cd2,cd1,L,y1,cp,sc)
```

```
for i=1:1:seg
```

```
    if abs(mean_ca(i))< .001
```

```
        cax(i,:)=c_mean(i,:)+(w(i)*x1*abs(mean_ca(i)));
```

```
    else if abs(mean_ca(i))< .005
```

```
        cax(i,:)=c_mean(i,:)+(w(i)*x2*abs(mean_ca(i)));
```

```
    else if abs(mean_ca(i))< .01
```

```
        cax(i,:)=c_mean(i,:)+(w(i)*x3*abs(mean_ca(i)));
```

```
    else
```

```
        cax(i,:)=c_mean(i,:)+(w(i)*1*abs(mean_ca(i)));
```

```
    end
```

```
end
```

```
end
```

```
end
```

```
for i=1:1:seg
```

```
    c(cp(i,:))=cax(i,:);
```

```
end
```

```
c=reshape(c',1,seg_w*seg);
```

```
c((seg*seg_w)+1:sc)=ca5((seg*seg_w)+1:sc);
```

```
Cnew = [c cd5 cd4 cd3 cd2 cd1];
```

```
sound = waverec(Cnew,L,wv);
```

```
wavwrite(sound,44100,16,'sound_emb');
```

```
[s2] = wavread('sound_emb.wav');audiox=rot90(s2(:,1),1);
```

```
[Cx,Lx] = wavedec(sound,5,wv);
```

```
cax5 = appcoef(Cx,Lx,wv,5);
```

```
cax_reshape=reshape(cax5(1:seg*seg_w),seg_w,seg);
```

```

cax_reshape=cax_reshape';

for i=1:1:seg
    caw(i,:)=cax_reshape(cp(i,:),:);
    if sign(mean(caw(i,:)))>0
        watermark(i)=1;
    else
        watermark(i)=0;
    end
end

snr= 10*log10(sum(y1.^2)/sum((y1-sound).^2));
sim= sum(watermark.*v)/((sqrt(sum(watermark.^2)))*(sqrt(sum(v.^2))));
error_snr=abs(snr_thre-snr);
error_sim=abs(1-sim);
error=(0.5*error_snr)+(0.5*error_sim);

%.....

%      Audio Watermarking Algorithm : Objective_2D23/ATS
%
%      Author: Noppanan Sriyingyong
%
%      Issue Date: Dec.2005
%.....

function [best_error1,best_neighbor1,best_error,best_neighbor]=Objective_2D23(S1,
best_error,best_neighbor,snr_thre,mean_ca,ca5,c_mean,seg,seg_w,w,v,wv,cd5,cd4,cd3,cd2,cd1,L
,y1,cp,sc)

error=[];

for k=1:size(S1,1)
    for i=1:1:seg

```

```

if abs(mean_ca(i))< .001
    cax2(i,:)=c_mean(i,:)+(w(i)*S1(k,1)*abs(mean_ca(i)));
else if abs(mean_ca(i))< .005
    cax2(i,:)=c_mean(i,:)+(w(i)*S1(k,2)*abs(mean_ca(i)));
else if abs(mean_ca(i))< .01
    cax2(i,:)=c_mean(i,:)+(w(i)*S1(k,3)*abs(mean_ca(i)));
else
    cax2(i,:)=c_mean(i,:)+(w(i)*1*abs(mean_ca(i)));
end
end
end
end

c=zeros(625,66);

for i=1:1:seg
    c(cp(i),:)=cax2(i,:);
end

c=reshape(c',1,seg_w*seg);
c((seg*seg_w)+1:sc)=ca5((seg*seg_w)+1:sc);
Cnew = [c cd5 cd4 cd3 cd2 cd1];
sound = waverec(Cnew,L,wv);
wavwrite(sound,44100,16,'sound_emb');

[s2] = wavread('sound_emb.wav');audiox=rot90(s2(:,1),1);
[Cx,Lx] = wavedec(sound,5,wv);
cax5 = appcoef(Cx,Lx,wv,5);
cax_reshape=reshape(cax5(1:seg*seg_w),seg_w,seg);
cax_reshape=cax_reshape';

```

```

for i=1:1:seg
    caw(i,:)=cax_reshape(cp(i,:));
    if sign(mean(caw(i,:)))>0
        watermark(i)=1;
    else
        watermark(i)=0;
    end
end

snr= 10*log10(sum(y1.^2)/sum((y1-sound).^2));
sim= sum(watermark.*v)/((sqrt(sum(watermark.^2)))*(sqrt(sum(v.^2))));
error_snr=abs(snr_thre-snr);
error_sim=abs(1-sim);
error2(k,1)=(0.5*error_snr)+(0.5*error_sim);

end

[best_error1,index]=min(error2);
best_neighbor1=S1(index,:);

if best_error1<best_error
    best_error=best_error1;
    best_neighbor=best_neighbor1;
else
end

```

ภาคผนวก ง

โปรแกรม MATLAB ที่ใช้รหัสแก้ไขความผิดพลาด

ส่วนโปรแกรมหลักที่ใช้ฝังลายน้ำด้วยการแปลงDWT

```
%.....  
%      Audio Watermarking Algorithm : Embedding, error correction code  
%      Author: Noppanan Sriyingyong  
%      Issue Date: Dec.2005  
%.....
```

```
clear,close all,clc
```

```
[y,Fs1,bit1] = wavread('MAHA CHAI 30s.wav') ;  
y1=rot90(y(:,1),1);  
sy1=length(y1);  
wavwrite(y1,Fs1,bit1,'original');
```

```
p=imread('sut25.jpg');  
pig=double(im2bw(p));  
[row,col]=size(pig);  
vc=reshape(pig,1,row*col);
```

```
rand('state',16);  
wp=randperm(row*col);  
for i=1:1:row*col  
    v(i)=vc(wp(i));  
end
```

```
count=5;  
wv='db4';  
n=15;  
k=5;
```

```

t=3;

msg=reshape(v,row*col/k,k);
code = encode(msg,n,k,'cyclic');
[row_c,col_c]=size(code);
wb=reshape(code,1,row_c*col_c);

for i=1:1:length(wb)
    if wb(i)==1
        w(i)=1;
    else
        w(i)=-1;
    end
end

[C,L] = wavedec(y1,count,wv);
ca5 = appcoef(C,L,wv,count);
[cd1,cd2,cd3,cd4,cd5]=detcoef(C,L,[1,2,3,4,5]);

seg=row_c*col_c;
sc=length(ca5);
seg_w =fix(sc/seg);

rand('state',28);cp=randperm(seg);
ca_reshape=reshape(ca5(1:seg*seg_w),seg_w,seg);
ca_reshape=ca_reshape';

for i=1:1:seg
    ca(i,:)=ca_reshape(cp(i,:));
end

for i=1:1:seg

```

```

mean_ca(i)=mean(ca(i,:));
caw(i,:)=ca(i,:)-mean_ca(i)+(w(i)*abs(mean_ca(i)));
end

```

```

for i=1:1:seg
    c_new(cp(i),:)=caw(i,:);
end

```

```

c_new=reshape(c_new',1,seg_w*seg);
c_new((seg*seg_w)+1:sc)=ca5((seg*seg_w)+1:sc);
Cnew = [c_new cd5 cd4 cd3 cd2 cd1];
audio = waverec(Cnew,L,wv);
wavwrite(audio,Fs1,bit1,'music_emb');

```

ส่วนโปรแกรมหลักที่ใช้ตรวจจับลายน้ำจากการโจมตีแบบต่างๆ

```

%.....
%      Audio Watermarking Algorithm : Detection, error correction code
%      Author: Noppanan Sriyingyong
%      Issue Date: Dec.2005
%.....

```

```

%1) Original
[y2,Fs2,bit2] = wavread('music_emb.wav') ;
audiox=rot90(y2(:,1),1);

[Cx,Lx] = wavedec(audiox,count,wv);
cax5 = appcoef(Cx,Lx,wv,count);
cax_reshape = reshape(cax5(1:seg*seg_w),seg_w,seg);
cax_reshape = cax_reshape';

for i=1:1:seg

```

```

caw(i,:)=cax_reshape(cp(i,:),);
if sign(mean(caw(i,:)))>0
    watermark(i)=1;
else
    watermark(i)=0;
end
end

[num,rat]=biterr(watermark,wb);
noisycode=reshape(watermark,row_c,col_c);
[newmsg,err,ccode] = decode(noisycode,n,k,'cyclic');
re_msg=reshape(newmsg,1,row*col);
SIM= sum(v.*re_msg)/((sqrt(sum(v.^2)))*(sqrt(sum(re_msg.^2))));
BIT=num;

% 2) Resampling
[y2,Fs2,bit2] = wavread('music_emb_PCM_PCM.wav');
audiox=rot90(y2(:,1),1);
audiox=audiox(6:max(size(audiox)));

[Cx,Lx] = wavedec(audiox,count,wv);
cax5 = appcoef(Cx,Lx,wv,count);
cax_reshape = reshape(cax5(1:seg*seg_w),seg_w,seg);
cax_reshape = cax_reshape';

for i=1:1:seg
    caw(i,:)=cax_reshape(cp(i,:),);
    if sign(mean(caw(i,:)))>0
        watermark(i)=1;
    else
        watermark(i)=0;
    end
end

```

```

    end
end

[num,rat]=biterr(watermark,wb);
noisycode=reshape(watermark,row_c,col_c);
[newmsg,err,ccode] = decode(noisycode,n,k,'cyclic');
re_msg=reshape(newmsg,1,row*col);
SIM= sum(v.*re_msg)/((sqrt(sum(v.^2)))*(sqrt(sum(re_msg.^2))));
BIT=num;

% 3) LPF
[y2,Fs2,bit2] = wavread('music_emb');
y2=rot90(y2(:,1),1);
[b,a] = butter(6,6000/(Fs2/2));
audiox=filter(b,a,y2);

[Cx,Lx] = wavedec(audiox,count,wv);
cax5 = appcoef(Cx,Lx,wv,count);
cax_reshape = reshape(cax5(1:seg*seg_w),seg_w,seg);
cax_reshape = cax_reshape';

for i=1:1:seg
    caw(i,:)=cax_reshape(cp(i,:));
    if sign(mean(caw(i,:)))>0
        watermark(i)=1;
    else
        watermark(i)=0;
    end
end
end

[num,rat]=biterr(watermark,wb);
noisycode=reshape(watermark,row_c,col_c);

```

```

[newmsg,err,ccode] = decode(noisycode,n,k,'cyclic');
re_msg=reshape(newmsg,1,row*col);
SIM= sum(v.*re_msg)/(((sqrt(sum(v.^2))))*(sqrt(sum(re_msg.^2))));
BIT=num;
wavwrite(audiox,Fs1,bit1,'music_emb_LPF');

```

```

% 3) Random noise
[y2,Fs2,bit2] = wavread('music_emb');
y2=rot90(y2,1);
noise=rand(1,sy1);
noise2=sqrt(0.01*sum(y2.^2)/sum(noise.^2))*noise;
audiox=(y2+noise2);

```

```

[Cx,Lx] = wavedec(audiox,count,wv);
[thr,sorh,keepapp]= ddencmp('den','wv',audiox);
audiox = wdencmp('gbl',Cx,Lx,wv,count,thr,sorh,keepapp);

```

```

[Cx,Lx] = wavedec(audiox,count,wv);
cax5 = appcoef(Cx,Lx,wv,count);
cax5 = cax5-mean(cax5);
cax_reshape = reshape(cax5(1:seg*seg_w),seg_w,seg);
cax_reshape = cax_reshape';

```

```

for i=1:1:seg
    caw(i,:)=cax_reshape(cp(i,:));
    if sign(mean(caw(i,:)))>0
        watermark(i)=1;
    else
        watermark(i)=0;
    end
end

```

```

[num, rat] = biterr(watermark, wb);
noisycode = reshape(watermark, row_c, col_c);
[newmsg, err, ccode] = decode(noisycode, n, k, 'cyclic');
re_msg = reshape(newmsg, 1, row * col);
SIM = sum(v .* re_msg) / ((sqrt(sum(v.^2))) * (sqrt(sum(re_msg.^2))));
BIT = num;
wavwrite(audiox, Fs1, bit1, 'music_emb_White noise');

```

% 4) White Gaussian noise

```

[y2, Fs2, bit2] = wavread('music_emb');
y2 = rot90(y2, 1);

```

```

for adj = 1:1:20
    audiox = awgn(y2, adj + 29);
    ratio_noise(adj) = sum((audiox - y2).^2) / sum(y2.^2);
    error_ratio(adj) = abs(.01 - ratio_noise(adj));
end

```

```

[mag, pos] = min(error_ratio);
audiox = awgn(y2, pos + 29);

```

```

[Cx, Lx] = wavedec(audiox, count, wv);
cax5 = appcoef(Cx, Lx, wv, count);
cax_reshape = reshape(cax5(1:seg * seg_w), seg_w, seg);
cax_reshape = cax_reshape';

```

```

for i = 1:1:seg
    caw(i, :) = cax_reshape(cp(i, :));
    if sign(mean(caw(i, :))) > 0
        watermark(i) = 1;
    end
end

```

```

else
    watermark(i)=0;
end
end

[num, rat] = biterr(watermark, wb);

noisycode = reshape(watermark, row_c, col_c);
[newmsg, err, ccode] = decode(noisycode, n, k, 'cyclic');
re_msg = reshape(newmsg, 1, row * col);
SIM = sum(v .* re_msg) / ((sqrt(sum(v.^2))) * (sqrt(sum(re_msg.^2))));
BIT = num; wavwrite(audiox, Fs1, bit1, 'music_emb_Gaussian noise');

% 6) DWT
[y2, Fs2, bit2] = wavread('music_emb.wav');
audiox = rot90(y2(:, 1), 1);

[thr, sorh, keepapp] = ddencmp('den', 'wv', audiox);
audiox = wdencomp('gbl', Cnew, L, wv, count, thr, sorh, keepapp);

[Cx, Lx] = wavedec(audiox, count, wv);
cax5 = appcoef(Cx, Lx, wv, count);
cax_reshape = reshape(cax5(1:seg * seg_w), seg_w, seg);
cax_reshape = cax_reshape';

for i = 1:1:seg
    caw(i, :) = cax_reshape(cp(i, :));
    if sign(mean(caw(i, :))) > 0
        watermark(i) = 1;
    else
        watermark(i) = 0;
    end
end

```

end

```
[num, rat] = biterr(watermark, wb);  
noisycode = reshape(watermark, row_c, col_c);  
[newmsg, err, ccode] = decode(noisycode, n, k, 'cyclic');  
re_msg = reshape(newmsg, 1, row * col);  
SIM = sum(v .* re_msg) / ((sqrt(sum(v.^2))) * (sqrt(sum(re_msg.^2))));  
BIT = num; wavwrite(audiox, Fs2, bit2, 'music_emb_DWT');
```

% 7) MP3

```
[y2, Fs2, bit2] = wavread('music_emb_56_PCM.wav');  
audiox = rot90(y2(:, 1), 1);  
xx = max(size(audiox));  
audiox = audiox(1:106:xx);
```

```
[Cx, Lx] = wavedec(audiox, count, wv);  
cax5 = appcoef(Cx, Lx, wv, count);  
cax_reshape = reshape(cax5(1:seg*seg_w), seg_w, seg);  
cax_reshape = cax_reshape';
```

```
for i = 1:1:seg  
    caw(i,:) = cax_reshape(cp(i,:));  
    if sign(mean(caw(i,:))) > 0  
        watermark(i) = 1;  
    else  
        watermark(i) = 0;  
    end  
end
```

```
[num, rat] = biterr(watermark, wb);  
noisycode = reshape(watermark, row_c, col_c);
```

```

[newmsg,err,ccode] = decode(noisycode,n,k,'cyclic');
re_msg=reshape(newmsg,1,row*col);
SIM= sum(v.*re_msg)/(((sqrt(sum(v.^2)))*(sqrt(sum(re_msg.^2)))));
BIT=num;

% 9) Jitter
[y2,Fs2,bit2] = wavread('music_emb');
audiox=rot90(y2(:,1),1);
sy2=length(audiox);

for i=1:fix(sy2/100)
    pos = randint(1,1,[1,100]);
    audiox((100*(i-1))+pos)=0;
end

[Cx,Lx] = wavedec(audiox,count,wv);
cax5 = appcoef(Cx,Lx,wv,count);
cax_reshape = reshape(cax5(1:seg*seg_w),seg_w,seg);
cax_reshape = cax_reshape';

for i=1:1:seg
    caw(i,:)=cax_reshape(cp(i,:));
    if sign(mean(caw(i,:)))>0
        watermark(i)=1;
    else
        watermark(i)=0;
    end
end

[num,rat]=biterr(watermark,wb);
noisycode=reshape(watermark,row_c,col_c);

```

```

[newmsg,err,ccode] = decode(noisycode,n,k,'cyclic');
re_msg=reshape(newmsg,1,row*col);
SIM= sum(v.*re_msg)/((sqrt(sum(v.^2)))*(sqrt(sum(re_msg.^2))));
BIT=num;wavwrite(audiox,Fs1,bit1,'music_emb_Jitter');
%Crop
[y2,Fs2,bit2] = wavread('music_emb');
y2=rot90(y2(:,1),1);
sy2=length(y2);

rand_pos=5;
rand_samp=500;
pos = randint(1,rand_pos,[1,fix(sy2/rand_samp)]);
for i=1:1:rand_pos
    y2((pos(i)*rand_samp)-(rand_samp-1):(pos(i)*rand_samp))=0;
end
audiox=y2;

[Cx,Lx] = wavedec(audiox,count,wv);
cax5 = appcoef(Cx,Lx,wv,count);
cax_reshape = reshape(cax5(1:seg*seg_w),seg_w,seg);
cax_reshape = cax_reshape';

for i=1:1:seg
    caw(i,:)=cax_reshape(cp(i,:));
    if sign(mean(caw(i,:)))>0
        watermark(i)=1;
    else
        watermark(i)=0;
    end
end
end

```

```
[num, rat] = biterr(watermark, wb);  
noisycode = reshape(watermark, row_c, col_c);  
[newmsg, err, ccode] = decode(noisycode, n, k, 'cyclic');  
re_msg = reshape(newmsg, 1, row * col);  
SIM = sum(v .* re_msg) / ((sqrt(sum(v.^2))) * (sqrt(sum(re_msg.^2))));  
BIT = num;  
wavwrite(audiox, Fs1, bit1, 'music_emb_crop_5_500');
```

ภาคผนวก จ

ผลการทดสอบความทนทานของลายน้ำจากอัลกอริทึมที่เสนอในงานวิจัยวิทยานิพนธ์และผลการทดสอบความทนทานของลายน้ำจากอัลกอริทึมที่ใช้รหัสแก้ไขความผิดพลาด

ตารางที่ จ.1 ถึงตารางที่ จ.12 แสดงค่าสหสัมพันธ์ของลายน้ำเมื่อสัญญาณเสียงผ่านการโจมตีด้วยการประมวลผลสัญญาณแบบต่างๆ ระหว่างอัลกอริทึมที่เสนอในวิทยานิพนธ์เปรียบเทียบกับอัลกอริทึมที่ใช้รหัสแก้ไขความผิดพลาดบนบทเพลงชนิดเดียวกันในแต่ละหน้า ซึ่งกำหนดให้ในการทดสอบมีค่า SNR เท่ากับ 20 dB เท่ากันทั้งสองอัลกอริทึม โดยตารางที่ จ.1 และตารางที่ จ.2 เป็นการเปรียบเทียบกันในเรื่อง classic ตารางที่ จ.3 และตารางที่ จ.4 เป็นการเปรียบเทียบกันในเรื่อง jazz ตารางที่ จ.5 และตารางที่ จ.6 เป็นการเปรียบเทียบกันในเรื่อง pop ตารางที่ จ.7 และตารางที่ จ.8 เป็นการเปรียบเทียบกันในเรื่อง ไทยเดิม ตารางที่ จ.9 และตารางที่ จ.10 เป็นการเปรียบเทียบกันในเรื่อง ไทยสุนทราภรณ์ ตารางที่ จ.11 และตารางที่ จ.12 เป็นการเปรียบเทียบกันในเรื่อง ไทยลูกทุ่ง

ตารางที่ จ.1 ผลการทดสอบอัลกอริทึมที่ใช้ในงานวิจัยวิทยานิพนธ์กับเพลงสากล classic

	Beautiful Blue Danube	Hungarian Dances	Marriage of Figaro	Moonlight	Romance for Violin & Orchestra
No attack	1	1	1	1	1
Downsampling	1	1	1	1	1
LPF	1	1	1	1	1
Random noise	1	1	1	1	1
Gaussian noise	1	1	1	1	1
Denoise-DWT	1	1	1	1	1
MP3: 128 kbps	1	1	1	1	1
MP3: 56 kbps	1	1	1	1	1
Jittering	1	1	1	1	1
Crop (5×100)	1	1	1	1	1
Crop (10×100)	1	1	1	1	1
Crop (10×500)	0.9831	0.9899	0.9899	0.9848	0.9899
Crop (10×1000)	0.9782	0.9744	0.9744	0.9762	0.9744

ตารางที่ จ.2 ผลการทดสอบอัลกอริทึมที่ใช้รหัสแก้ไขความผิดพลาดกับเพลงสากล classic

	Beautiful Blue Danube	Hungarian Dances	Marriage of Figaro	Moonlight	Romance for Violin & Orchestra
No attack	1	1	1	1	1
Downsampling	1	1	1	1	1
LPF	1	1	1	1	1
Random noise	1	1	1	1	1
Gaussian noise	1	1	1	1	1
Denoise-DWT	1	1	1	1	1
MP3: 128 kbps	1	1	1	1	1
MP3: 56 kbps	1	1	1	1	1
Jittering	1	1	1	1	1
Crop (5×100)	1	1	1	1	1
Crop (10×100)	1	1	1	1	1
Crop (10×500)	1	1	1	1	1
Crop (10×1000)	0.9899	0.9933	0.9933	1	1

ตารางที่ จ.1 ผลการทดสอบอัลกอริทึมที่ใช้ในงานวิจัยวิทยานิพนธ์กับเพลงสากล jazz

	Beauty in me	Shade of pain	Till I get it right	I could wait a book	Squeeze me
No attack	1	1	1	1	1
Downsampling	1	1	1	1	1
LPF	1	1	1	1	1
Random noise	1	1	1	0.9966	1
Gaussian noise	1	1	1	1	1
Denoise-DWT	1	1	1	1	1
MP3: 128 kbps	1	1	1	1	1
MP3: 56 kbps	1	1	1	1	1
Jittering	1	1	1	0.9983	1
Crop (5×100)	0.9950	1	1	1	1
Crop (10×100)	0.9950	1	1	0.9983	1
Crop (10×500)	0.9831	0.9831	0.9882	0.9899	0.9831
Crop (10×1000)	0.9710	0.9693	0.9730	0.9744	0.9693

ตารางที่ จ.2 ผลการทดสอบอัลกอริทึมที่ใช้รหัสแก้ไขความผิดพลาดกับเพลงสากล jazz

	Beauty in me	Shade of pain	Till I get it right	I could wait a book	Squeeze me
No attack	1	1	1	1	1
Downsampling	1	1	1	1	1
LPF	1	1	0.9693	0.9675	1
Random noise	1	1	1	0.9730	1
Gaussian noise	1	1	0.9693	0.9548	1
Denoise-DWT	1	1	1	1	1
MP3: 128 kbps	1	1	1	0.9548	1
MP3: 56 kbps	1	1	1	0.9548	1
Jittering	1	1	0.9950	1	1
Crop (5×100)	1	1	1	1	1
Crop (10×100)	1	1	1	1	1
Crop (10×500)	1	1	1	1	1
Crop (10×1000)	1	1	0.9933	1	1

ตารางที่ จ.1 ผลการทดสอบอัลกอริทึมที่ใช้ในงานวิจัยวิทยานิพนธ์กับเพลงสากล pop

	Vincent	White Flag	Sunrise	Sometime	Where is the love
No attack	1	1	1	1	1
Downsampling	1	1	1	1	1
LPF	1	1	1	1	1
Random noise	1	0.9966	1	1	1
Gaussian noise	1	0.9966	1	1	1
Denoise-DWT	1	1	1	1	1
MP3: 128 kbps	1	1	1	1	1
MP3: 56 kbps	1	1	0.9882	1	0.9899
Jittering	1	1	1	1	1
Crop (5×100)	1	1	1	1	1
Crop (10×100)	0.9966	1	1	1	0.9966
Crop (10×500)	0.9865	0.9899	0.9831	0.9831	0.9831
Crop (10×1000)	0.9675	0.9744	0.9797	0.9675	0.9662

ตารางที่ จ.2 ผลการทดสอบอัลกอริทึมที่ใช้รหัสแก้ไขความผิดพลาดกับเพลงสากล pop

	Vincent	White flag	Sunrise	Sometime	Where is the love
No attack	1	1	1	1	1
Downsampling	1	1	1	1	1
LPF	1	0.9899	0.9966	0.9933	0.9966
Random noise	1	1	1	1	1
Gaussian noise	1	0.9698	0.9605	1	1
Denoise-DWT	1	1	1	1	1
MP3: 128 kbps	1	0.9833	0.9134	1	0.9744
MP3: 56 kbps	1	0.9833	0.9134	1	0.9744
Jittering	1	1	0.9950	1	1
Crop (5×100)	1	1	1	1	1
Crop (10×100)	1	1	1	1	1
Crop (10×500)	1	1	1	1	1
Crop (10×1000)	1	0.9933	0.9933	1	1

ตารางที่ จ.1 ผลการทดสอบอัลกอริทึมที่ใช้ในงานวิจัยวิทยานิพนธ์กับเพลงไทยเดิม

	มหาฤกษ์	วิวาห์ พระสมุทร	เทวา ประสิทธิ์	โพธิ์สัตว์	มหาชัย
No attack	1	1	1	1	1
Downsampling	1	1	1	1	1
LPF	1	1	1	1	1
Random noise	0.9966	1	1	0.9966	1
Gaussian noise	0.9950	1	1	1	1
Denoise-DWT	1	1	1	1	1
MP3: 128 kbps	1	1	1	1	1
MP3: 56 kbps	1	1	1	1	1
Jittering	0.9966	1	1	1	1
Crop (5×100)	1	1	1	1	1
Crop (10×100)	0	0	0	0	0
Crop (10×500)	0.9916	0.9916	0.9933	0.9950	0.9882
Crop (10×1000)	0.9730	0.9762	0.9762	0.9730	0.9693

ตารางที่ จ.2 ผลการทดสอบอัลกอริทึมที่ใช้รหัสแก้ไขความผิดพลาดกับเพลงไทยเดิม

	มหาฤกษ์	วิวาห์ พระสมุทร	เทวา ประสิทธิ์	โพธิ์สัตว์	มหาชัย
No attack	1	1	1	1	1
Downsampling	1	1	1	1	1
LPF	1	1	1	1	1
Random noise	1	1	1	1	1
Gaussian noise	1	1	1	1	0.9916
Denoise-DWT	1	1	1	1	1
MP3: 128 kbps	1	1	1	1	1
MP3: 56 kbps	1	1	1	1	1
Jittering	1	1	1	1	1
Crop (5×100)	1	1	1	1	1
Crop (10×100)	1	1	1	1	1
Crop (10×500)	1	1	1	1	1
Crop (10×1000)	1	0.9899	0.9916	0.9916	1

ตารางที่ จ.1 ผลการทดสอบอัลกอริทึมที่ใช้ในงานวิจัยวิทยานิพนธ์กับเพลงไทยสุนทราภรณ์

	เขียนลว่าว	พรหมลิขิต	ฟลอร์ เฟื่องฟ้า	ตัดสวาท	ขวัญใจเจ้า ทวย
No attack	1	1	1	1	1
Downsampling	1	1	1	1	1
LPF	1	1	1	0.9966	1
Random noise	1	1	1	1	1
Gaussian noise	1	1	1	1	1
Denoise-DWT	1	1	1	1	1
MP3: 128 kbps	1	1	1	1	1
MP3: 56 kbps	1	1	1	0.9899	1
Jittering	1	1	1	1	1
Crop (5×100)	1	1	1	1	1
Crop (10×100)	1	1	1	0.9966	1
Crop (10×500)	0.9882	0.9916	0.9950	0.9916	0.9865
Crop (10×1000)	0.9730	0.9744	0.9848	0.9730	0.9744

ตารางที่ จ.2 ผลการทดสอบอัลกอริทึมที่ใช้รหัสแก้ไขความผิดพลาดกับเพลงไทยสุนทราภรณ์

	เขียนลว่าว	พรหมลิขิต	ฟลอร์ เฟื่องฟ้า	ตัดสวาท	ขวัญใจเจ้า ทวย
No attack	1	1	1	1	1
Downsampling	1	1	1	1	1
LPF	1	1	0.9983	1	0.9933
Random noise	1	1	1	1	1
Gaussian noise	1	1	1	1	1
Denoise-DWT	1	1	1	1	1
MP3: 128 kbps	1	1	0.9950	0.9933	1
MP3: 56 kbps	1	1	0.9950	0.9933	1
Jittering	1	1	1	1	1
Crop (5×100)	1	1	1	1	1
Crop (10×100)	1	1	1	1	1
Crop (10×500)	1	1	1	1	1
Crop (10×1000)	1	0.9933	1	0.9882	1

ตารางที่ จ.1 ผลการทดสอบอัลกอริทึมที่ใช้ในงานวิจัยวิทยานิพนธ์กับเพลงไทยลูกทุ่ง

	เหนือไหม คนดี	อัสสาว ขอนแก่น	สร้างวิมาน	สาวฝั่งโขง	สิบหกลีแห่ง ความหลัง
No attack	1	1	1	1	1
Downsampling	1	1	1	1	1
LPF	1	1	1	1	1
Random noise	1	0.9933	1	0.9966	0.9865
Gaussian noise	1	1	1	1	1
Denoise-DWT	1	1	1	1	1
MP3: 128 kbps	1	1	1	1	1
MP3: 56 kbps	1	1	1	1	1
Jittering	1	1	1	1	1
Crop (5×100)	1	1	1	1	1
Crop (10×100)	1	1	1	0.9933	1
Crop (10×500)	0.9882	0.9950	0.9865	0.9865	0.9848
Crop (10×1000)	0.9710	0.9730	0.9762	0.9710	0.9662

ตารางที่ จ.2 ผลการทดสอบอัลกอริทึมที่ใช้รหัสแก้ไขความผิดพลาดกับเพลงไทยลูกทุ่ง

	เหนือไหม คนดี	อัสสาว ขอนแก่น	สร้างวิมาน	สาวฝั่งโขง	สิบหกลีแห่ง ความหลัง
No attack	1	1	1	1	1
Downsampling	1	1	1	1	1
LPF	1	0.9916	1	1	1
Random noise	1	1	1	1	1
Gaussian noise	1	1	1	1	1
Denoise-DWT	1	1	1	1	1
MP3: 128 kbps	1	0.9675	1	0.9899	0.9882
MP3: 56 kbps	1	0.9675	1	0.9899	0.9882
Jittering	0.9916	1	1	1	1
Crop (5×100)	1	1	1	1	1
Crop (10×100)	1	1	1	1	1
Crop (10×500)	1	1	1	1	1
Crop (10×1000)	1	1	1	1	1

ภาคผนวก จ

ค่าความเข้มข้นในการฝังลายน้ำของเพลงต่างๆ
ที่ได้จากการค้นหาแบบตามูเชิงปรับตัว

ตารางที่ จ.1 ค่าความเข้มในการฟังลายน้ำของเพลงต่างๆ

	α_1	α_2	α_3
เพลง classic			
Beautyful Blue Danube	24.619717440000	19.808022050000	7.664960476000
Hungarian Dances	22.739892620000	19.665952080000	3.078797649000
Marriage of Figaro	29.459637120000	13.619686970000	6.403135838000
Moonlight	13.317441166886	3.861318531894	2.819040838062
Romance of Violin & Orcheatra	11.458131561197	9.365473649013	1.006297446961
เพลง jazz			
Beauty in me	45.093516136232	40.417786975114	5.884743882009
Shade of pain	39.024913246474	19.587770688347	5.750931752962
Till i get it right	19.947992056292	16.153644441978	6.940417802238
I could write a book	23.768705633140	11.808031976259	5.754973858103
Squeeze me	33.257413037617	26.263402070554	9.965192802921
เพลง pop			
Vincent	19.894133038347	10.017582881886	6.389960613282
White flag	39.177547850701	36.847995962258	9.804537555558
Red	91.346724257728	38.680107572468	12.652024968410
Sometime	85.389399966769	17.299013201269	12.181410493819
Where is the love	64.724101711118	51.120433072922	14.358381121862
เพลงไทยเดิม			
วิวาห์พระสมุทร	19.743160586587	7.807264158402	3.864867061814
เทวาประสิทธิ์	20.082033835140	14.733109123431	4.313203050708
โพธิ์สัตว์	28.957602309135	17.996125378441	2.662503348810
มหาฤกษ์	13.481151999496	10.876675470042	1.657601489127
มหาชัย	26.663405568763	10.121005887432	3.069812605782

ตารางที่ จ.1 ค่าความเข้มในการฝังลายน้ำของเพลงต่างๆ (ต่อ)

	α_1	α_2	α_3
เพลงไทยสุนทราภรณ์			
เย็นลมว่าว	25.850593620775	10.914486652183	6.305601385718
พรหมลิขิต	29.415953234452	23.050525645465	6.780528999037
ฟลอร์เฟื่องฟ้า	37.787742248516	21.972899656010	9.847270811749
ตัดสวาท	40.013760136379	32.939437639784	7.751060663319
ขวัญใจเจ้าทวย	68.220726489353	40.988409643206	6.285907824680
เพลงไทยลูกทุ่ง			
เหนื่อยไหมคนดี	64.090527870000	54.663487170000	6.214527330000
ฮักสาวขอนแก่น	19.843275200000	10.026328630000	6.346218382000
สร้างวิมาน	15.932566870000	11.328600170000	9.105824360000
สาวฝั่งโขง	20.750109460000	14.247312850000	7.470686290000
สิบหกปีแห่งความหลัง	23.701121730000	11.641100960000	4.594607500000

ภาคผนวก ข

แบบทดสอบคุณภาพเสียงหลังการฝังลายน้ำโดยการฟัง

ในแบบทดสอบ1ชุดประกอบด้วยแบบทดสอบแบ่งออกเป็น 2 ส่วน

ส่วนที่1. ข้อมูลของผู้ทำแบบทดสอบ

ส่วนที่2. แบบทดสอบเปรียบเทียบคุณภาพของเสียงเพลงทดสอบที่ฟังลายน้แล้วกับเพลงทดสอบที่ไม่ฟังลายน้

ส่วนที่1. ข้อมูลของผู้ทำแบบทดสอบ

กรอกข้อความในช่องว่างและทำเครื่องหมาย✓ใน ☐

ชื่อ (นาย/นาง/นางสาว).....นามสกุล.....

ตำแหน่ง

สถานที่ทำงาน

e-mail:.....

ความรู้ความเข้าใจในกระบวนการฟังลายน้ ☐ มาก ☐ ปานกลาง ☐ น้อย ☐ ไม่เลย

ชนิดเพลงสากลที่ฟัง

.....

ชนิดเพลงไทยที่ฟัง

.....

ส่วนที่2. แบบทดสอบเปรียบเทียบคุณภาพของเสียงเพลงทดสอบที่ยังไม่ฟังลายน้้า(เพลงต้นฉบับ)
เทียบกับเพลงทดสอบที่ฟังลายน้้าแล้ว (เพลงทดสอบที่ 1 และ เพลงทดสอบที่ 2) แล้วเลือกทำเครื่องหมาย X ในตัวเลือกเพียงตัวเลือกเดียวในแต่ละเพลง

เพลงสากล Classic

1. เพลง The Beautiful Blue Danube

- ก. เพลงทดสอบที่ 1 มีคุณภาพลดลง
- ข. เพลงทดสอบที่ 2 มีคุณภาพลดลง
- ค. เพลงทดสอบที่ 1 และ 2 ไม่มีคุณภาพลดลงเลย
- ง. เพลงทดสอบที่ 1 และ 2 มีคุณภาพลดลงทั้งคู่

2. เพลง Hungarian Dance No.05

- ก. เพลงทดสอบที่ 1 มีคุณภาพลดลง
- ข. เพลงทดสอบที่ 2 มีคุณภาพลดลง
- ค. เพลงทดสอบที่ 1 และ 2 ไม่มีคุณภาพลดลงเลย
- ง. เพลงทดสอบที่ 1 และ 2 มีคุณภาพลดลงทั้งคู่

3. เพลง Marriage of Figaro

- ก. เพลงทดสอบที่ 1 มีคุณภาพลดลง
- ข. เพลงทดสอบที่ 2 มีคุณภาพลดลง
- ค. เพลงทดสอบที่ 1 และ 2 ไม่มีคุณภาพลดลงเลย
- ง. เพลงทดสอบที่ 1 และ 2 มีคุณภาพลดลงทั้งคู่

4. เพลง Moonlight

- ก. เพลงทดสอบที่ 1 มีคุณภาพลดลง
- ข. เพลงทดสอบที่ 2 มีคุณภาพลดลง
- ค. เพลงทดสอบที่ 1 และ 2 ไม่มีคุณภาพลดลงเลย
- ง. เพลงทดสอบที่ 1 และ 2 มีคุณภาพลดลงทั้งคู่

5. เพลง Romance of Violin

- ก. เพลงทดสอบที่ 1 มีคุณภาพลดลง

- ข. เพลงทดสอบที่ 2 มีคุณภาพลดลง
- ค. เพลงทดสอบที่ 1 และ 2 ไม่มีคุณภาพลดลงเลย
- ง. เพลงทดสอบที่ 1 และ 2 มีคุณภาพลดลงทั้งคู่

เพลงสากล Jazz

1. เพลง Beauty in Me

- ก. เพลงทดสอบที่ 1 มีคุณภาพลดลง
- ข. เพลงทดสอบที่ 2 มีคุณภาพลดลง
- ค. เพลงทดสอบที่ 1 และ 2 ไม่มีคุณภาพลดลงเลย
- ง. เพลงทดสอบที่ 1 และ 2 มีคุณภาพลดลงทั้งคู่

2. เพลง Shade of Pain

- ก. เพลงทดสอบที่ 1 มีคุณภาพลดลง
- ข. เพลงทดสอบที่ 2 มีคุณภาพลดลง
- ค. เพลงทดสอบที่ 1 และ 2 ไม่มีคุณภาพลดลงเลย
- ง. เพลงทดสอบที่ 1 และ 2 มีคุณภาพลดลงทั้งคู่

3. เพลง 'Till I get it Right

- ก. เพลงทดสอบที่ 1 มีคุณภาพลดลง
- ข. เพลงทดสอบที่ 2 มีคุณภาพลดลง
- ค. เพลงทดสอบที่ 1 และ 2 ไม่มีคุณภาพลดลงเลย
- ง. เพลงทดสอบที่ 1 และ 2 มีคุณภาพลดลงทั้งคู่

4. เพลง I could write a book

- ก. เพลงทดสอบที่ 1 มีคุณภาพลดลง
- ข. เพลงทดสอบที่ 2 มีคุณภาพลดลง
- ค. เพลงทดสอบที่ 1 และ 2 ไม่มีคุณภาพลดลงเลย
- ง. เพลงทดสอบที่ 1 และ 2 มีคุณภาพลดลงทั้งคู่

5. เพลง Squeeze Me

- ก. เพลงทดสอบที่ 1 มีคุณภาพลดลง

- ข. เพลงทดสอบที่ 2 มีคุณภาพลดลง
- ค. เพลงทดสอบที่ 1 และ 2 ไม่มีคุณภาพลดลงเลย
- ง. เพลงทดสอบที่ 1 และ 2 มีคุณภาพลดลงทั้งคู่

เพลงสากล Pop

1. เพลง Vincent

- ก. เพลงทดสอบที่ 1 มีคุณภาพลดลง
- ข. เพลงทดสอบที่ 2 มีคุณภาพลดลง
- ค. เพลงทดสอบที่ 1 และ 2 ไม่มีคุณภาพลดลงเลย
- ง. เพลงทดสอบที่ 1 และ 2 มีคุณภาพลดลงทั้งคู่

2. เพลง White Flag

- ก. เพลงทดสอบที่ 1 มีคุณภาพลดลง
- ข. เพลงทดสอบที่ 2 มีคุณภาพลดลง
- ค. เพลงทดสอบที่ 1 และ 2 ไม่มีคุณภาพลดลงเลย
- ง. เพลงทดสอบที่ 1 และ 2 มีคุณภาพลดลงทั้งคู่

3. เพลง Sunrise

- ก. เพลงทดสอบที่ 1 มีคุณภาพลดลง
- ข. เพลงทดสอบที่ 2 มีคุณภาพลดลง
- ค. เพลงทดสอบที่ 1 และ 2 ไม่มีคุณภาพลดลงเลย
- ง. เพลงทดสอบที่ 1 และ 2 มีคุณภาพลดลงทั้งคู่

4. เพลง Sometime

- ก. เพลงทดสอบที่ 1 มีคุณภาพลดลง
- ข. เพลงทดสอบที่ 2 มีคุณภาพลดลง
- ค. เพลงทดสอบที่ 1 และ 2 ไม่มีคุณภาพลดลงเลย
- ง. เพลงทดสอบที่ 1 และ 2 มีคุณภาพลดลงทั้งคู่

5. เพลง Where is the Love

- ก. เพลงทดสอบที่ 1 มีคุณภาพลดลง
- ข. เพลงทดสอบที่ 2 มีคุณภาพลดลง
- ค. เพลงทดสอบที่ 1 และ 2 ไม่มีคุณภาพลดลงเลย
- ง. เพลงทดสอบที่ 1 และ 2 มีคุณภาพลดลงทั้งคู่

เพลง ไทยเดิม

1. เพลง มหาฤกษ์

- ก. เพลงทดสอบที่ 1 มีคุณภาพลดลง
- ข. เพลงทดสอบที่ 2 มีคุณภาพลดลง
- ค. เพลงทดสอบที่ 1 และ 2 ไม่มีคุณภาพลดลงเลย
- ง. เพลงทดสอบที่ 1 และ 2 มีคุณภาพลดลงทั้งคู่

2. เพลง เทวาประสิทธิ์

- ก. เพลงทดสอบที่ 1 มีคุณภาพลดลง
- ข. เพลงทดสอบที่ 2 มีคุณภาพลดลง
- ค. เพลงทดสอบที่ 1 และ 2 ไม่มีคุณภาพลดลงเลย
- ง. เพลงทดสอบที่ 1 และ 2 มีคุณภาพลดลงทั้งคู่

3. เพลง โขธสีดาว

- ก. เพลงทดสอบที่ 1 มีคุณภาพลดลง
- ข. เพลงทดสอบที่ 2 มีคุณภาพลดลง
- ค. เพลงทดสอบที่ 1 และ 2 ไม่มีคุณภาพลดลงเลย
- ง. เพลงทดสอบที่ 1 และ 2 มีคุณภาพลดลงทั้งคู่

4. เพลง วิวาท์พระสมุทร

- ก. เพลงทดสอบที่ 1 มีคุณภาพลดลง
- ข. เพลงทดสอบที่ 2 มีคุณภาพลดลง
- ค. เพลงทดสอบที่ 1 และ 2 ไม่มีคุณภาพลดลงเลย
- ง. เพลงทดสอบที่ 1 และ 2 มีคุณภาพลดลงทั้งคู่

5. เพลง มหาชัย

- ก. เพลงทดสอบที่ 1 มีคุณภาพลดลง
- ข. เพลงทดสอบที่ 2 มีคุณภาพลดลง
- ค. เพลงทดสอบที่ 1 และ 2 ไม่มีคุณภาพลดลงเลย
- ง. เพลงทดสอบที่ 1 และ 2 มีคุณภาพลดลงทั้งคู่

เพลงไทยสุนทราภรณ์

1. เพลง เย็นลมว่าว

- ก. เพลงทดสอบที่ 1 มีคุณภาพลดลง
- ข. เพลงทดสอบที่ 2 มีคุณภาพลดลง
- ค. เพลงทดสอบที่ 1 และ 2 ไม่มีคุณภาพลดลงเลย
- ง. เพลงทดสอบที่ 1 และ 2 มีคุณภาพลดลงทั้งคู่

2. เพลง พรหมลิขิต

- ก. เพลงทดสอบที่ 1 มีคุณภาพลดลง
- ข. เพลงทดสอบที่ 2 มีคุณภาพลดลง
- ค. เพลงทดสอบที่ 1 และ 2 ไม่มีคุณภาพลดลงเลย
- ง. เพลงทดสอบที่ 1 และ 2 มีคุณภาพลดลงทั้งคู่

3. เพลง ฟลอร์เฟื่องฟ้า

- ก. เพลงทดสอบที่ 1 มีคุณภาพลดลง
- ข. เพลงทดสอบที่ 2 มีคุณภาพลดลง
- ค. เพลงทดสอบที่ 1 และ 2 ไม่มีคุณภาพลดลงเลย
- ง. เพลงทดสอบที่ 1 และ 2 มีคุณภาพลดลงทั้งคู่

4. เพลง คัดสาวท

- ก. เพลงทดสอบที่ 1 มีคุณภาพลดลง
- ข. เพลงทดสอบที่ 2 มีคุณภาพลดลง
- ค. เพลงทดสอบที่ 1 และ 2 ไม่มีคุณภาพลดลงเลย
- ง. เพลงทดสอบที่ 1 และ 2 มีคุณภาพลดลงทั้งคู่

5. เพลง ขวัญใจเจ้าทวย

- ก. เพลงทดสอบที่ 1 มีคุณภาพลดลง
- ข. เพลงทดสอบที่ 2 มีคุณภาพลดลง
- ค. เพลงทดสอบที่ 1 และ 2 ไม่มีคุณภาพลดลงเลย
- ง. เพลงทดสอบที่ 1 และ 2 มีคุณภาพลดลงทั้งคู่

เพลงไทยลูกทุ่ง

1. เพลง เหนื่อยไหมคนดี

- ก. เพลงทดสอบที่ 1 มีคุณภาพลดลง
- ข. เพลงทดสอบที่ 2 มีคุณภาพลดลง
- ค. เพลงทดสอบที่ 1 และ 2 ไม่มีคุณภาพลดลงเลย
- ง. เพลงทดสอบที่ 1 และ 2 มีคุณภาพลดลงทั้งคู่

2. เพลง อักสาวขอนแก่น

- ก. เพลงทดสอบที่ 1 มีคุณภาพลดลง
- ข. เพลงทดสอบที่ 2 มีคุณภาพลดลง
- ค. เพลงทดสอบที่ 1 และ 2 ไม่มีคุณภาพลดลงเลย
- ง. เพลงทดสอบที่ 1 และ 2 มีคุณภาพลดลงทั้งคู่

3. เพลง สร้างวิมาน

- ก. เพลงทดสอบที่ 1 มีคุณภาพลดลง
- ข. เพลงทดสอบที่ 2 มีคุณภาพลดลง
- ค. เพลงทดสอบที่ 1 และ 2 ไม่มีคุณภาพลดลงเลย
- ง. เพลงทดสอบที่ 1 และ 2 มีคุณภาพลดลงทั้งคู่

4. เพลง สาวฝั่งโขง

- ก. เพลงทดสอบที่ 1 มีคุณภาพลดลง
- ข. เพลงทดสอบที่ 2 มีคุณภาพลดลง
- ค. เพลงทดสอบที่ 1 และ 2 ไม่มีคุณภาพลดลงเลย
- ง. เพลงทดสอบที่ 1 และ 2 มีคุณภาพลดลงทั้งคู่

5. เพลง สิบหกปีแห่งความหลัง

ก. เพลงทดสอบที่ 1 มีคุณภาพลดลง

ข. เพลงทดสอบที่ 2 มีคุณภาพลดลง

ค. เพลงทดสอบที่ 1 และ 2 ไม่มีคุณภาพลดลงเลย

ง. เพลงทดสอบที่ 1 และ 2 มีคุณภาพลดลงทั้งคู่

นพรัตน์ ศรียิ่งยงค์ (ผู้จัดทำแบบทดสอบ)

27/10/48

ภาคผนวก ข

บทความที่ได้รับการตีพิมพ์เผยแพร่

บทความที่ได้รับการตีพิมพ์เผยแพร่ในระดับนานาชาติ

N. Sriyingyong and K. Attakitmongcol, “Wavelet-Based Audio watermarking Using Adaptive Tabu search”, International Symposium on Wireless Pervasive Computing 2006, *IEEE 1st ISWPC*, 16-18 Jan. 2006

ประวัติผู้เขียน

นายพนันท์ ศรียิ่งยงค์ เกิดเมื่อวันที่ 3 สิงหาคม 2522 ที่ตลาดสดริมแม่น้ำท่าจีน ตำบลสามพราน อำเภอสามพราน จังหวัดนครปฐม สำเร็จการศึกษาระดับประถมศึกษาจากโรงเรียนสุพิทยานุกุล (ป้วยฮั่ว) สำเร็จการศึกษาระดับมัธยมศึกษาจากโรงเรียนนาคนาคประสิทธิ์ เริ่มเข้าศึกษาในระดับปริญญาตรี ณ มหาวิทยาลัยเทคโนโลยีสุรนารี จังหวัดนครราชสีมา เมื่อ พ.ศ. 2540 และสำเร็จการศึกษาวិชากรรมศาสตรบัณฑิต (วิศวกรรมไฟฟ้า) เมื่อ พ.ศ. 2544 ภายหลังสำเร็จการศึกษาได้เข้าทำงานในตำแหน่งวิศวกรฝ่ายขายที่ Techsource System (Thailand) Co., Ltd. ในปี พ.ศ. 2544 ได้เข้าทำงานกับสาขาวิชาวิศวกรรมไฟฟ้า สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ในตำแหน่งอาจารย์พิเศษสอนรายวิชาปฏิบัติการจนถึงปี พ.ศ. 2548 เข้าศึกษาในระดับปริญญาโท สาขาวิชาวิศวกรรมไฟฟ้า ณ มหาวิทยาลัยเทคโนโลยีสุรนารี เมื่อ พ.ศ. 2545