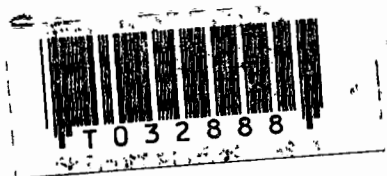


การลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจโดยใช้การแปลงเวฟเลต

ECG DATA COMPRESSION USING WAVELET TRANSFORM



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

๔, ๓๗ ก
๒๕๔๒

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่.....
เลขทะเบียน.....
วัน, เดือน, ปี.....

32888

พ.ศ. 2542

ISBN 974-622-392-5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ECG DATA COMPRESSION USING WAVELET TRANSFORM

MANIT KIATKHAMJAIKAJORN

A THESIS SUBMITTED IN PARTIAL FULFILMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

1999

ISBN 974-622-392-5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 1999

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจโดยใช้การแปลงเวฟเลต
นักศึกษา	นายมานิตย์ เกียรติกำจายขจร
รหัสประจำตัว	38621205
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2542
อาจารย์ผู้ควบคุมวิทยานิพนธ์	ผศ. ประภากร สุวรรณะ

บทคัดย่อ

ระบบบันทึกสัญญาณไฟฟ้าหัวใจของผู้ป่วยด้วยคอมพิวเตอร์ส่วนบุคคลเพื่อการวินิจฉัยในระยะยาวจะต้องใช้เนื้อที่ในการบันทึกข้อมูลที่สูงมาก เพื่อเป็นการใช้เนื้อที่ของอุปกรณ์บันทึกที่มีอยู่อย่างจำกัดให้มีประสิทธิภาพมากที่สุด จึงจำเป็นที่จะต้องลดขนาดข้อมูลก่อนการบันทึก วิทยานิพนธ์นี้เป็นการนำเสนอการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจโดยใช้การแปลงเวฟเลต การแปลงเวฟเลตจะแบ่งข้อมูลออกเป็นกลุ่มข้อมูลย่อยที่เป็นอิสระต่อกัน ทำให้ภายหลังจากขบวนการแควนไทซ์สามารถเพิ่มประสิทธิภาพการลดขนาดข้อมูลด้วยขบวนการ Run-Length Coding และขบวนการ Histogram Circular Shifting ก่อนขบวนการเข้ารหัส Huffman เพื่อการบันทึกต่อไป ผลการคำนวณด้วยโปรแกรม Matlab® แสดงให้เห็นถึงประสิทธิภาพการลดขนาดข้อมูลด้วยค่า Compression Ratio (CR), Percent Root-mean-square Difference (PRD) และการเปรียบเทียบกับรูปสัญญาณ จากความเห็นของแพทย์สามารถสรุปได้ว่า การลดขนาดข้อมูลโดยใช้เวฟเลตแบบ Coiflet-3 ที่ค่า $CR = 6$ ซึ่งให้ค่า PRD อยู่ในช่วง 3-5 % เป็นเงื่อนไขที่เหมาะสมในการนำไปใช้งาน

Thesis Title	ECG Data Compression Using Wavelet Transform
Student	Mr. Manit Kiatkhamjaikajorn
Student ID.	38621205
Degree	Master of Engineering
Programme	Electrical Engineering
Year	1999
Thesis Advisor	Assis. Prof. Prapakorn Suwana

ABSTRACT

A personal computer ECG recording system for the long time diagnosis consumes a large size of area for recording. Using the efficiency of the limited storage device, it is necessary to compress the ECG data before recording. This thesis proposes the ECG Data Compression Using Wavelet Transform. In the method, the original data is decomposed into a set of independent sub-data. Consequently, prior to the traditional Huffman Coding, after the Quantization process the efficiency of the system can be increased by the Run-Length Coding and the Histogram Circular Shifting. The compression performance of Wavelet Transform indicated by Compression Ratio (CR), Percent Root-mean-square Difference (PRD) and signal comparison is resulted by Matlab®. By doctors' suggestions it can be concluded that by Coiflet-3, CR = 6 that makes PRD in a range of 3-5 % is the practical condition.

กิตติกรรมประกาศ

การจัดทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี เพราะได้รับความเมตตาจาก คุณพ่อและคุณแม่ ผู้ที่คอยให้กำลังใจและความห่วงใยตลอดมา ขอขอบพระคุณ อาจารย์ ประภากร สุวรรณะ รองศาสตราจารย์ ดร. มนัส สังวรศิลป์ และผู้ช่วยศาสตราจารย์ ดร. สุรพันธ์ เอื้อไพบูลย์ ผู้ซึ่งสละเวลาอันมีค่าคอยให้ความรู้ ให้คำปรึกษาและชี้แนะแนวทางการแก้ปัญหาที่ต้องตลอดมา

ขอขอบพระคุณ นายแพทย์สันติ โลกเจริญลาภ จากโรงพยาบาลราชวิถี กรุงเทพมหานคร ขอขอบพระคุณ นายแพทย์สมชาย ตั้งสถิตย์ นายแพทย์วิวัฒน์ แจ่มจริยวงศ์ จาก โรงพยาบาลสรรพสิทธิประสงค์ จังหวัดอุบลราชธานี และขอขอบพระคุณนายแพทย์วิบูลย์ เดชะโกศล นายแพทย์วิรัชยุทธ คำตรี จากโรงพยาบาลขอนแก่น จังหวัดขอนแก่น ที่สละเวลาอันมีค่าให้ข้อคิดเห็นเพื่อนำไปสู่ข้อสรุปของงานวิจัยที่ถูกต้อง

ขอขอบคุณพี่ๆ เพื่อนๆ และน้องๆ ทุกคน โดยเฉพาะคุณอังกริยา เหล่าศิริ คุณพิพรรพ ทวีวัฒน์กิจ คุณอิทธิพล พจนสังข์ คุณทศพล วงศ์ชัยสุริยะ คุณทัฬหะ รุ่งสว่าง คุณธีระณัฐ เริ่มคิดการ คุณบัณฑิต พัสยา และคุณวัฒนา โพธิ์เจริญ ที่คอยให้ช่วยเหลือในการจัดเตรียมต้นฉบับในการทำวิทยานิพนธ์นี้

สุดท้าย ขอขอบพระคุณบัณฑิตวิทยาลัย ที่กรุณาให้เงินทุนสนับสนุนการจัดทำวิทยานิพนธ์ในครั้งนี้

คุณค่าและประโยชน์อันพึงมีจากวิทยานิพนธ์ฉบับนี้ ผู้วิจัยขอบอบแด่ผู้มีพระคุณทุกท่าน

มานิตย์ เกียรติคำจายขจร

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VIII
สารบัญรูป	IX
บทที่ 1 บทนำ	1
1.1 ความเป็นมาของงานวิจัย	1
1.2 วัตถุประสงค์ของงานวิจัย	1
1.3 ขอบเขตของงานวิจัยและโครงสร้างของวิทยานิพนธ์	1
บทที่ 2 ทฤษฎีการวัดสัญญาณไฟฟ้าหัวใจ	3
2.1 การกำเนิดและแบบจำลองของแหล่งกำเนิดสัญญาณไฟฟ้าหัวใจ	3
2.2 การวัดสัญญาณไฟฟ้าหัวใจ	4
2.2.1 การวัดเพื่อการวินิจฉัยคนไข้ข้างเดียวแบบมาตรฐาน	5
2.2.1.1 วิธีการวัดแบบ Standard Limb Lead	5
2.2.1.2 วิธีการวัดแบบ Augmented Limb Lead	6
2.2.1.3 วิธีการวัดแบบ Unipolar Chest Lead	10
2.2.2 การวัดเพื่อการมอนิเตอร์	11
2.3 ลักษณะและคุณสมบัติของสัญญาณไฟฟ้าหัวใจ	11
2.4 เครื่องวัดสัญญาณไฟฟ้าหัวใจ	13
2.4.1 ขั้ววัดสัญญาณไฟฟ้าหัวใจจากผิวหนัง	13
2.4.2 วงจรวัดสัญญาณไฟฟ้าหัวใจ	13
บทที่ 3 ทฤษฎีการแปลงเวฟเลต	18
3.1 ทฤษฎีการแปลงของสัญญาณ	18
3.2 การวิเคราะห์แบบมัลติเรโซลูชัน	20
3.3 อนุกรมเวฟเลต	23

สารบัญ(ต่อ)

หน้า

3.4 การวิเคราะห์สัญญาณเชิงเต็มหน่วยด้วยทฤษฎีเวฟเล็ต	23
3.5 การแปลงเวฟเล็ตเต็มหน่วย	24
3.6 การแปลงกลับเวฟเล็ตเต็มหน่วย	26
3.7 ตระกูลของออโรนอร์มัลเวฟเล็ต	28
3.8 ผลตอบสนองความถี่ของ Octave Band Wavelet Filter Bank	37
บทที่ 4 ทฤษฎีการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจ	41
4.1 วิธีการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจ	41
4.1.1 การลดขนาดข้อมูลด้วยวิธีทางตรง	41
4.1.1.1 วิธี Turning Point (TP)	41
4.1.1.2 วิธี Amplitude Zone Time Epoch Coding (AZTEC)	42
4.1.1.3 วิธี Coordinate Reduction Time Encoding System (CORTES)	42
4.1.1.4 วิธี Fan	42
4.1.1.5 วิธี Run-Length Coding (RLC)	43
4.1.1.6 วิธีการเข้ารหัสเอนโทรปี(Entropy Coding)	44
4.1.2 การลดขนาดข้อมูลด้วยวิธีการแปลง	49
4.2 มาตรฐานในการวัดประสิทธิภาพการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจ	51
บทที่ 5 การลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจโดยใช้เทคนิคการแปลงเวฟเล็ต	53
5.1 ขบวนการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจโดยใช้การแปลงเวฟเล็ต	53
5.1.1 การแควนไคซ์สัมประสิทธิ์เวฟเล็ต	54
5.1.2 การเพิ่มประสิทธิภาพการลดขนาดข้อมูลด้วยการเข้ารหัส Run-Length	62
5.1.3 การเพิ่มประสิทธิภาพการลดขนาดข้อมูลด้วยวิธี Histogram Circular Shifting	62
5.1.4 การเข้ารหัสฮัฟแมน	63
5.2 ขบวนการสร้างสัญญาณกลับคืน	63

บทที่ 6 การทดสอบประสิทธิภาพการลดขนาดข้อมูล	65
--	----

สารบัญ(ต่อ)

หน้า

6.1 การทดสอบการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจ	65
6.2 ผลการทดสอบกับตัวอย่างสัญญาณไฟฟ้าหัวใจปกติทั่วไป	66
6.2.1 ผลการเปรียบเทียบประสิทธิภาพการลดขนาดข้อมูลระหว่างวิธี DWT และวิธี Fan.....	81
6.2.2 ผลการเปรียบเทียบประสิทธิภาพการลดขนาดข้อมูลระหว่างวิธี DWT และวิธี DCT.....	84
6.3 การทดสอบเพิ่มเติมโดยใช้สัญญาณไฟฟ้าหัวใจผิดปกติ	90
6.3.1 สัญญาณไฟฟ้าหัวใจผิดปกติแบบ Premature Ventricular Contraction	90
6.3.2 สัญญาณไฟฟ้าหัวใจผิดปกติแบบ Atrial Flutter	95
บทที่ 7 การประยุกต์ใช้งานกับคอมพิวเตอร์ส่วนบุคคล	102
7.1 ระบบบันทึกสัญญาณไฟฟ้าหัวใจด้วยคอมพิวเตอร์ส่วนบุคคล	102
7.2 โปรแกรมและการควบคุมการทำงานระบบบันทึกสัญญาณไฟฟ้าหัวใจ	103
7.2.1 การควบคุมการทำงานของ A/D Card	103
7.2.2 โปรแกรมระบบบันทึกสัญญาณไฟฟ้าหัวใจ	103
7.2.2.1 รูปแบบการบันทึกเป็นแฟ้มข้อมูลในอุปกรณ์บันทึก.....	107
7.3 คุณสมบัติของโปรแกรมระบบบันทึกสัญญาณไฟฟ้าหัวใจ	109
7.4 การใช้งานโปรแกรมระบบบันทึกสัญญาณไฟฟ้าหัวใจ	109
7.5 การทดสอบระบบบันทึกสัญญาณไฟฟ้าหัวใจ	111
บทที่ 8 สรุปผลการวิจัยและข้อเสนอแนะ	114
บรรณานุกรม.....	117
ภาคผนวก ก การคำนวณผลตอบสนองความถี่ของ Octave Band Analysis Filter Bank โดยใช้โปรแกรม Matlab®	118
ภาคผนวก ข ทฤษฎีค่าเฉลี่ยเชิงสถิติ	119
ภาคผนวก ค โปรแกรมภาษาซีของระบบบันทึกสัญญาณไฟฟ้าหัวใจโดยใช้วิธีการแปลง	

สารบัญ(ต่อ)

หน้า

เว็บเด้ดในการลดขนาดข้อมูล.....	121
ภาคผนวก ง แบบสอบถามความคิดเห็นจากแพทย์.....	169
ประวัติผู้เขียน	201



สารบัญตาราง

ตารางที่	หน้า
3.1 แสดงผลตอบสนองอิมพัลส์ $h_r(n)$ ของเวฟเลตตระกูล Daubechies เมื่อ NVM มีค่า 4, 6, 8, 10 และ 12	32
3.2 แสดงผลตอบสนองอิมพัลส์ $h_r(n)$ ของเวฟเลตตระกูล Daubechies เมื่อ NVM เมื่อ NVM มีค่า 14, 16, 18, และ 20.....	33
3.3 แสดงผลตอบสนองอิมพัลส์ $h_r(n)$ คูณด้วย $\sqrt{2}$ ของเวฟเลตตระกูล Symmlet เมื่อ NVM มีค่าจาก 4 ถึง 7.....	34
3.4 แสดงผลตอบสนองอิมพัลส์ $h_r(n)$ คูณด้วย $\sqrt{2}$ ของเวฟเลตตระกูล Symmlet เมื่อ NVM มีค่าจาก 8 ถึง 10.....	35
3.5 แสดงผลตอบสนองอิมพัลส์ $h_r(n)$ คูณด้วย $\sqrt{2}$ ของเวฟเลตตระกูล Coiflet เมื่อ NVM มีค่าจาก 1 ถึง 5.....	36
6.1 แสดงผลการตรวจสอบสัญญาณไฟฟ้าหัวใจปกติที่ผ่านการลดขนาดข้อมูลโดยใช้เวฟเลตชนิดต่างๆจากแพทย์ทั้งหมด 5 ท่าน เมื่อ CR = 6.....	88
6.2 แสดงผลการตรวจสอบสัญญาณไฟฟ้าหัวใจปกติที่ผ่านการลดขนาดข้อมูลโดยใช้เวฟเลตชนิดต่างๆจากแพทย์ทั้งหมด 5 ท่าน เมื่อ CR = 9.....	88
6.3 แสดงผลการตรวจสอบสัญญาณไฟฟ้าหัวใจปกติที่ผ่านการลดขนาดข้อมูลโดยใช้เวฟเลตชนิดต่างๆจากแพทย์ทั้งหมด 5 ท่าน เมื่อ CR = 12.....	89
6.4 แสดงชนิดของเวฟเลตที่แพทย์ผู้วินิจฉัยมีความเห็นว่าสามารถรักษาสาระสำคัญของข้อมูลสัญญาณไฟฟ้าหัวใจต้นแบบได้ดีที่สุด.....	89
6.5 แสดงค่า CR สูงสุดที่แพทย์ผู้วินิจฉัยยอมรับสำหรับการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจผิดปกติแบบ Premature Ventricular Contraction โดยใช้วิธี DWT	95
6.6 แสดงค่า CR สูงสุดที่แพทย์ผู้วินิจฉัยยอมรับสำหรับการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจผิดปกติแบบ Atrial Flutter โดยใช้วิธี DWT	101

สารบัญรูป

รูปที่	หน้า
2.1 แสดงสามเหลี่ยมด้านเท่าของอิน์โทรเฟ่นและการฉายเวกเตอร์ของอิน์โทรเฟ่นลงบนแต่ละด้านของสามเหลี่ยม.....	3
2.2 แสดงวิธีพื้นฐานในการวัดสัญญาณไฟฟ้าหัวใจจากผิวหนังในตำแหน่งของ II Lead โดยประมาณ	4
2.3 แสดงแบบจำลองแหล่งกำเนิดสัญญาณไฟฟ้าหัวใจแบบ Standard Limb Lead	5
2.4 แสดงวิธีการวัดสัญญาณไฟฟ้าหัวใจแบบ Standard Limb Lead	6
2.5 แสดงแบบจำลองการวัดสัญญาณไฟฟ้าหัวใจแบบ Augmented Limb Lead aVL	6
2.6 แสดงเวกเตอร์ของ aVL Lead ที่หาได้จากเวกเตอร์ของ I Lead และ III Lead	7
2.7 แสดงทิศทางและขนาดของเวกเตอร์แบบ Standard และ Augmented Limb Lead.....	8
2.8 แสดงตัวอย่างสัญญาณไฟฟ้าหัวใจและความสัมพันธ์ระหว่างสัญญาณไฟฟ้าหัวใจแบบ Standard และ Augmented Limb Lead.....	9
2.9 แสดงวิธีการวัดสัญญาณไฟฟ้าหัวใจแบบ Augmented Limb Lead.....	10
2.10 แสดงวิธีการวัดสัญญาณไฟฟ้าหัวใจแบบ Unipolar Chest Lead.....	10
2.11 แสดงตำแหน่งการติดขั้ววัดบนหน้าอกของวิธีการวัดสัญญาณไฟฟ้าหัวใจแบบ Unipolar Chest Lead V_1 ถึง V_6	10
2.12 แสดงตัวอย่างสัญญาณไฟฟ้าหัวใจแบบ Unipolar Chest Lead V_1 ถึง V_6 และตำแหน่งการติดขั้ววัดบนหน้าอกโดยมองภาพตัดขวางของลำตัว.....	11
2.13 แสดงลักษณะสัญญาณไฟฟ้าหัวใจของคนปกติ.....	12
2.14 แสดงช่วงความถี่ของสัญญาณไฟฟ้าหัวใจสำหรับการประยุกต์ใช้งานแบบต่างๆ.....	12
2.15 แสดงบล็อกโคเดสเกรมวงจรวัดสัญญาณไฟฟ้าหัวใจ	15
2.16 แสดงวงจรวัดสัญญาณไฟฟ้าหัวใจ.....	16
3.1 แสดง Two-Channel Analysis Filter Bank ที่ใช้ในการคำนวณ $c_j(n)$ และ $d_j(n)$ จาก $c_{j+1}(n)$	26
3.2 แสดงการแปลงเวฟเลตเต็มหน่วย L ระดับในรูปของ Octave Band Analysis Filter Bank.....	26
3.3 แสดง Two-Channel Synthesis Filter Bank ที่ใช้ในการคำนวณ $\hat{c}_{j+1}(n)$ จาก $\hat{c}_j(n)$ และ $\hat{d}_j(n)$	27
3.4 แสดงการแปลงกลับเวฟเลตเต็มหน่วย L ระดับในรูปของ Octave Band Synthesis	

สารบัญรูป(ต่อ)

รูปที่	หน้า
Filter Bank.....	28
3.5 แสดงฟังก์ชันมูลฐานเวฟเล็ตตระกูล Symmlet เมื่อค่า NVM เป็น 4, 6, 8 และ 10.....	29
3.6 แสดงฟังก์ชันมูลฐานเวฟเล็ตตระกูล Daubechies เมื่อค่า NVM เป็น 4, 6, 10, 14 และ 18.....	30
3.7 แสดงฟังก์ชันมูลฐานเวฟเล็ตตระกูล Coiflet เมื่อค่า NVM จาก 1 ถึง 5.....	31
3.8 แสดงผลตอบสนองความถี่ของตัวกรองที่มีผลตอบสนองอิมพัลส์ $h_g(n)$ และ $g_g(n)$ แบบ Daubechies-20.....	37
3.9 แสดงรูปแบบสมมูลย์ของ Octave Band Analysis Filter Bank 2 ระดับ.....	38
3.10 แสดงผลตอบสนองความถี่ของ Octave Band Analysis Filter Bank 2 ระดับ โดยใช้ตัวกรองแบบ Daubechies-20.....	38
3.11 แสดงรูปแบบสมมูลย์ของ Octave Band Analysis Filter Bank 3 ระดับ.....	39
3.12 แสดงผลตอบสนองความถี่ของ Octave Band Analysis Filter Bank 3 ระดับ โดยใช้ตัวกรองแบบ Daubechies-20.....	40
4.1 แสดงการลดขนาดข้อมูลด้วยวิธี Fan (ก) การลากเส้นขอบเขตเพื่อพิจารณาการเก็บและการละข้อมูล (ข) การหาตำแหน่งของ X_{U_2} และ X_{L_2} จาก X_{U_1} , X_{L_1} และ X_0	43
4.2 แสดงต้นไม้รหัสแบบ 2 กิ่งที่แสดงการเข้ารหัสของข่าวสารที่มี 6 องค์ประกอบ.....	45
4.3 แสดงการสร้างรหัสฐานสองด้วยวิธีของฮัฟแมน	48
4.4 แสดงวิธีการสร้างรหัสฐานสองด้วยวิธีของแชนนอนฟาโน.....	49
4.5 แสดงรูปแบบทั่วไปของการลดขนาดข้อมูลด้วยวิธีการแปลง.....	51
4.6 แสดงรูปแบบการสร้างกลับคืนของข้อมูลที่ผ่านการลดขนาดข้อมูลด้วยวิธีการแปลง.....	51
5.1 แสดงขบวนการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจโดยใช้การแปลงเวฟเล็ต	53
5.2 แสดงแบบจำลองของ Two-Channel Synthesis Filter Bank สำหรับการวิเคราะห์ความเพี้ยน.....	54
5.3 แสดงแบบจำลองการวิเคราะห์ความเพี้ยนที่เกิดขึ้นเนื่องการแควน ไตซ์ของขบวนการลดขนาดข้อมูลด้วยวิธีการแปลงเวฟเล็ต.....	56
5.4 แสดงความสัมพันธ์ระหว่างค่า PRD และ ε	59
5.5 แสดงตัวอย่างข้อมูลสัญญาณไฟฟ้าหัวใจที่ได้จาก A/D ขนาดความละเอียด 12 Bits/Sample จำนวน 2048 Samples เมื่อระดับอ้างอิงของสัญญาณ(0 โวลต์)อยู่	

สารบัญรูป(ต่อ)

รูปที่	หน้า
ที่ 2048	60
5.6 แสดงกลุ่มข้อมูลที่ได้จากการแปลงเวฟเลต 2 ระดับของตัวอย่างข้อมูลสัญญาณไฟฟ้าหัวใจในรูปที่ 5.5 และนำกลุ่มข้อมูลทุกกลุ่มมารวมอยู่ในอาร์เรย์เดียวกัน	61
5.7 แสดงกลุ่มข้อมูลที่ได้จากการแปลงเวฟเลต 2 ระดับของตัวอย่างข้อมูลสัญญาณไฟฟ้าหัวใจในรูปที่ 5.5 โดยที่ $c_{11}(n)$ ถูกลบออกด้วยค่าเฉลี่ยของกลุ่มก่อนผ่านขบวนการลดขนาดข้อมูล	61
5.8 แสดงขบวนการสร้างสัญญาณกลับคืน	64
6.1 แสดงฮิสโตแกรมของ Quantized Wavelet Coefficients เมื่อ $CR = 3$	67
6.2 แสดงฮิสโตแกรมของ Quantized Wavelet Coefficients เมื่อ $CR = 6$	67
6.3 แสดงฮิสโตแกรมของ Quantized Wavelet Coefficients เมื่อ $CR = 9$	68
6.4 แสดงฮิสโตแกรมของ Quantized Wavelet Coefficients เมื่อ $CR = 12$	68
6.5 แสดงการเปรียบเทียบประสิทธิภาพการลดขนาดข้อมูลด้วยค่า PRD เมื่อใช้เวฟเลตที่มีระดับความราบเรียบต่ำ ปานกลาง และสูงของเวฟเลตตระกูล Daubechies, Symmlet และ Coiflet	69
6.6 แสดงการเปรียบเทียบประสิทธิภาพการลดขนาดข้อมูลด้วยค่า PRD ของเวฟเลตที่ให้ประสิทธิภาพสูงสุดจากแต่ละตระกูล	70
6.7 แสดงการเปรียบเทียบประสิทธิภาพการลดขนาดข้อมูลด้วยค่า PRD เมื่ออันดับการแปลง L เปลี่ยนไป	70
6.8 แสดงสัญญาณไฟฟ้าหัวใจปกติเมื่อผ่านขบวนการลดขนาดข้อมูลโดยใช้เวฟเลตตระกูล Daubechies ที่ $CR = 6$	72
6.9 แสดงสัญญาณไฟฟ้าหัวใจปกติเมื่อผ่านขบวนการลดขนาดข้อมูลโดยใช้เวฟเลตตระกูล Symmlet ที่ $CR = 6$	73
6.10 แสดงสัญญาณไฟฟ้าหัวใจปกติเมื่อผ่านขบวนการลดขนาดข้อมูลโดยใช้เวฟเลตตระกูล Coiflet ที่ $CR = 6$	74
6.11 แสดงสัญญาณไฟฟ้าหัวใจปกติเมื่อผ่านขบวนการลดขนาดข้อมูลโดยใช้เวฟเลตตระกูล Daubechies ที่ $CR = 9$	75
6.12 แสดงสัญญาณไฟฟ้าหัวใจปกติเมื่อผ่านขบวนการลดขนาดข้อมูลโดยใช้เวฟเลตตระกูล Symmlet ที่ $CR = 9$	76

สารบัญรูป(ต่อ)

รูปที่	หน้า
6.13 แสดงสัญญาณไฟฟ้าหัวใจปกติเมื่อผ่านขบวนการลดขนาดข้อมูลโดยใช้เวฟเล็ต ตระกูล Coiflet ที่ $CR = 9$	77
6.14 แสดงสัญญาณไฟฟ้าหัวใจปกติเมื่อผ่านขบวนการลดขนาดข้อมูลโดยใช้เวฟเล็ต ตระกูล Daubechies ที่ $CR = 12$	78
6.15 แสดงสัญญาณไฟฟ้าหัวใจปกติเมื่อผ่านขบวนการลดขนาดข้อมูลโดยใช้เวฟเล็ต ตระกูล Symmlet ที่ $CR = 12$	79
6.16 แสดงสัญญาณไฟฟ้าหัวใจปกติเมื่อผ่านขบวนการลดขนาดข้อมูลโดยใช้เวฟเล็ต ตระกูล Coiflet ที่ $CR = 12$	80
6.17 แสดงตัวอย่างฮิสโตแกรมของ Quantized Wavelet Coefficients ก่อนผ่านขบวนการ Histogram Circular Shifting (รูป ก) และหลังผ่านขบวนการ Histogram Circular Shifting (รูป ข)	82
6.18 แสดงสัญญาณไฟฟ้าหัวใจปกติเมื่อผ่านการขบวนการลดขนาดข้อมูลด้วยวิธี Fan และวิธี DWT เมื่อกำหนดให้ $CR = 4$	83
6.19 แสดงการเปรียบเทียบประสิทธิภาพการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจปกติ ด้วยค่า PRD ระหว่างวิธี DWT และวิธี DCT	84
6.20 แสดงสัญญาณไฟฟ้าหัวใจปกติที่ผ่านขบวนการลดขนาดข้อมูลด้วยวิธี DCT และ วิธี DWT เมื่อกำหนดให้ $CR = 6$	85
6.21 แสดงสัญญาณไฟฟ้าหัวใจปกติที่ผ่านขบวนการลดขนาดข้อมูลด้วยวิธี DCT และ วิธี DWT เมื่อกำหนดให้ $CR = 9$	86
6.22 แสดงสัญญาณไฟฟ้าหัวใจปกติที่ผ่านขบวนการลดขนาดข้อมูลด้วยวิธี DCT และ วิธี DWT เมื่อกำหนดให้ $CR = 12$	87
6.23 แสดงสัญญาณไฟฟ้าหัวใจผิดปกติในลักษณะ Premature Ventricular Contraction เมื่อผ่านขบวนการลดขนาดข้อมูลด้วยวิธี DWT เปรียบเทียบกับวิธี Fan เมื่อกำหนด ให้ $CR = 4$	90
6.24 แสดงประสิทธิภาพการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจผิดปกติแบบ Premature Ventricular Contraction ด้วยค่า PRD ของวิธี DWT เปรียบเทียบ กับวิธี DCT	91
6.25 แสดงสัญญาณไฟฟ้าหัวใจผิดปกติในลักษณะ Premature Ventricular Contraction	

สารบัญรูป(ต่อ)

รูปที่

หน้า

เมื่อผ่านขบวนการลดขนาดข้อมูลด้วยวิธี DWT เปรียบเทียบกับวิธี DCT เมื่อกำหนดให้ $CR = 6$	92
6.26 แสดงสัญญาณไฟฟ้าหัวใจผิดปกติในลักษณะ Premature Ventricular Contraction เมื่อผ่านขบวนการลดขนาดข้อมูลด้วยวิธี DWT เปรียบเทียบกับวิธี DCT เมื่อกำหนดให้ $CR = 9$	93
6.27 แสดงสัญญาณไฟฟ้าหัวใจผิดปกติในลักษณะ Premature Ventricular Contraction เมื่อผ่านขบวนการลดขนาดข้อมูลด้วยวิธี DWT เปรียบเทียบกับวิธี DCT เมื่อกำหนดให้ $CR = 12$	94
6.28 แสดงสัญญาณไฟฟ้าหัวใจผิดปกติในลักษณะ Atrial Flutter เมื่อผ่านขบวนการลดขนาดข้อมูลด้วยวิธี DWT เปรียบเทียบกับวิธี Fan เมื่อกำหนดให้ $CR = 4$	96
6.29 แสดงประสิทธิภาพการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจผิดปกติแบบ Atrial Flutter ด้วยค่า PRD ของวิธี DWT เปรียบเทียบกับวิธี DCT	97
6.30 แสดงสัญญาณไฟฟ้าหัวใจผิดปกติในลักษณะ Atrial Flutter เมื่อผ่านขบวนการลดขนาดข้อมูลด้วยวิธี DWT เปรียบเทียบกับวิธี DCT เมื่อกำหนดให้ $CR = 6$	98
6.31 แสดงสัญญาณไฟฟ้าหัวใจผิดปกติในลักษณะ Atrial Flutter เมื่อผ่านขบวนการลดขนาดข้อมูลด้วยวิธี DWT เปรียบเทียบกับวิธี DCT เมื่อกำหนดให้ $CR = 9$	99
6.32 แสดงสัญญาณไฟฟ้าหัวใจผิดปกติในลักษณะ Atrial Flutter เมื่อผ่านขบวนการลดขนาดข้อมูลด้วยวิธี DWT เปรียบเทียบกับวิธี DCT เมื่อกำหนดให้ $CR = 12$	100
7.1 แสดงระบบบันทึกสัญญาณไฟฟ้าหัวใจด้วยคอมพิวเตอร์ส่วนบุคคลที่ใช้ในวิทยานิพนธ์	102
7.2 แสดง A/D Card ที่ใช้ในระบบบันทึกสัญญาณไฟฟ้าหัวใจด้วยคอมพิวเตอร์ส่วนบุคคล	102
7.3 แสดงการทำงานของ A/D Card ในลักษณะของ Double Buffer Cyclic Mode	103
7.4 แสดงแผนผังโปรแกรมหลักของระบบบันทึกสัญญาณไฟฟ้าหัวใจ	104
7.5 แสดงแผนผังโปรแกรมน้อยที่ใช้ในการแสดงสัญญาณไฟฟ้าหัวใจเพื่อการมอนิเตอร์	105
7.6 แสดงแผนผังโปรแกรมน้อยที่ใช้ในการบันทึกสัญญาณไฟฟ้าหัวใจ	106
7.7 แสดงแผนผังโปรแกรมน้อยที่ใช้ในการอ่านสัญญาณไฟฟ้าหัวใจจากแฟ้มข้อมูล	107
7.8 แสดงรูปแบบการบันทึกเป็นแฟ้มข้อมูลในอุปกรณ์บันทึก	109

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่	หน้า
7.9 แสดงลักษณะของโปรแกรมระบบบันทึกสัญญาณไฟฟ้าหัวใจบนหน้าจอคอมพิวเตอร์.....	111
7.10 แสดงระบบบันทึกสัญญาณไฟฟ้าหัวใจค้นแบบขณะทำการทดสอบการแสดง สัญญาณเพื่อการมอิตอร์.....	112
7.11 แสดงหน้าจอโปรแกรมขณะทำการทดสอบการบันทึกสัญญาณไฟฟ้าหัวใจเป็น แฟ้มข้อมูล.....	112
7.12 แสดงหน้าจอโปรแกรมขณะทำการทดสอบการอ่านสัญญาณไฟฟ้าหัวใจจากแฟ้ม ข้อมูล.....	113



บทที่ 1

บทนำ

1.1 ความเป็นมาของงานวิจัย

ปัจจุบันคอมพิวเตอร์ส่วนบุคคลเข้ามามีบทบาทในชีวิตประจำวันของมนุษย์เป็นอย่างมาก โดยได้รับความนิยมในงานทุกสาขา ทั้งนี้ก็เนื่องมาจากความสะดวกในการใช้งาน การจัดเก็บและการค้นหาข้อมูลได้อย่างมีประสิทธิภาพนั่นเอง การประยุกต์ใช้คอมพิวเตอร์ส่วนบุคคลในการบันทึกสัญญาณไฟฟ้าหัวใจเพื่อการวินิจฉัยความผิดปกติของระบบหัวใจ นับว่าเป็นวิธีการหนึ่งที่ทำให้การจัดเก็บและค้นหาข้อมูลในทางการแพทย์มีประสิทธิภาพมากยิ่งขึ้น อย่างไรก็ตาม สำหรับการบันทึกสัญญาณไฟฟ้าหัวใจเพื่อการวินิจฉัยความผิดปกติของระบบหัวใจในระยะยาวนั้น จำเป็นจะต้องใช้เนื้อที่ในการบันทึกข้อมูลที่สูง เมื่อจำนวนผู้ป่วยเพิ่มมากขึ้น การจัดหาอุปกรณ์บันทึกให้เพียงพอกับปริมาณของข้อมูลจึงเป็นสิ่งที่จำเป็นอย่างยิ่ง ซึ่งทำให้ต้องสิ้นเปลืองค่าใช้จ่ายเพิ่มมากขึ้นไปด้วย ดังนั้นในงานวิจัยนี้จึงนำเสนอการเพิ่มประสิทธิภาพการบันทึกข้อมูลสัญญาณไฟฟ้าหัวใจโดยการใช้ความรู้ทางด้านการประมวลสัญญาณดิจิทัลและการเข้ารหัสมาช่วยในการลดขนาดข้อมูลด้วยการโปรแกรม จึงเป็นที่มาของหัวข้อวิทยานิพนธ์เรื่อง การลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจโดยใช้การแปลงเวฟเลต

1.2 วัตถุประสงค์ของงานวิจัย

วัตถุประสงค์ที่สำคัญของงานวิจัยสามารถแบ่งออกเป็นข้อๆ ได้ดังนี้

1. เพื่อศึกษาและออกแบบอัลกอริทึมการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจโดยใช้วิธีการแปลงเวฟเลต
2. เพื่อหาข้อสรุปในการใช้งานที่เหมาะสมของการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจโดยใช้วิธีการแปลงเวฟเลตที่ได้นำเสนอ
3. เพื่อเป็นแนวทางในการพัฒนาขอบข่ายการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจ ให้กับผู้ที่สนใจได้ทำการศึกษาและพัฒนาให้ดียิ่งขึ้นต่อไปในอนาคต

1.3 ขอบเขตของงานวิจัยและโครงร่างของวิทยานิพนธ์

งานวิจัยที่นำเสนอนี้เป็นการออกแบบอัลกอริทึมการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจโดยใช้วิธีการแปลงเวฟเลต โดยเลือกทำการทดสอบกับสัญญาณไฟฟ้าหัวใจเพียงลีด(Lead)เดียวเท่านั้น

นั้น วิทยานิพนธ์ฉบับนี้ประกอบด้วยเนื้อหาทั้งหมด 8 บทดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1 กล่าวถึงความเป็นมาของงานวิจัย วัตถุประสงค์ของงานวิจัย ขอบเขตของงานวิจัย และโครงสร้างของวิทยานิพนธ์

บทที่ 2 กล่าวถึงหลักการและทฤษฎีพื้นฐานในการวัดสัญญาณไฟฟ้าหัวใจในทางการแพทย์

บทที่ 3 กล่าวถึงทฤษฎีการวิเคราะห์สัญญาณด้วยทฤษฎีเวฟเล็ต การวิเคราะห์สัญญาณแบบมัลติเรโซลูชัน และการแปลงเวฟเล็ตเต็มหน่วย

บทที่ 4 กล่าวถึงหลักการและทฤษฎีพื้นฐานในการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจ

บทที่ 5 กล่าวถึงการออกแบบอัลกอริทึมการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจโดยใช้วิธีการแปลงเวฟเล็ต

บทที่ 6 กล่าวถึงการทดสอบอัลกอริทึมการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจที่ได้ออกแบบขึ้นกับสัญญาณไฟฟ้าหัวใจตัวอย่าง เพื่อแสดงประสิทธิภาพการลดขนาดข้อมูลของอัลกอริทึมที่ได้นำเสนอ

บทที่ 7 กล่าวถึงการประยุกต์ใช้งานจริงในการบันทึกสัญญาณไฟฟ้าหัวใจด้วยคอมพิวเตอร์ส่วนบุคคลโดยใช้วิธีการแปลงเวฟเล็ตในการลดขนาดข้อมูล ตลอดจนคุณสมบัติและการใช้งานของโปรแกรมที่ได้ออกแบบขึ้น

บทที่ 8 เป็นการสรุปผลการวิจัยและแนวทางในการพัฒนา

และท้ายสุดเป็นภาคผนวกเพื่อเพิ่มความเข้าใจในการศึกษาเนื้อหาบางส่วนในวิทยานิพนธ์ฉบับนี้ ตลอดจนโปรแกรมภาษาซีที่ใช้ในการควบคุมการทำงานของคอมพิวเตอร์ส่วนบุคคลในการบันทึกสัญญาณไฟฟ้าหัวใจโดยใช้วิธีการแปลงเวฟเล็ตในการลดขนาดข้อมูล

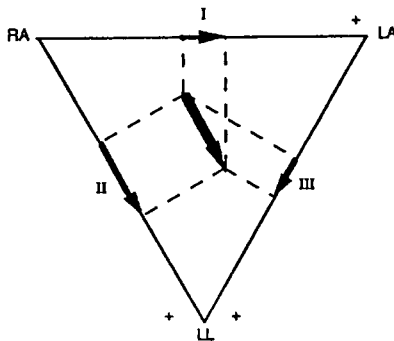
บทที่ 2

ทฤษฎีการวัดสัญญาณไฟฟ้าหัวใจ

ในทางการแพทย์ การวัดสัญญาณไฟฟ้าหัวใจนับเป็นเทคนิคที่สำคัญวิธีหนึ่งที่ใช้ในการวินิจฉัยการทำงานของระบบหัวใจที่มีประสิทธิภาพสูงและเป็นที่นิยมใช้กันมากในปัจจุบัน เนื่องจากเป็นวิธีที่สะดวกในการวัดและการวินิจฉัยของแพทย์นั่นเอง ในบทนี้จะกล่าวถึงทฤษฎีเบื้องต้นในการวัดสัญญาณไฟฟ้าหัวใจจากร่างกาย[1],[2] มาตรฐานในการวัดสัญญาณไฟฟ้าหัวใจแบบต่างๆ ตลอดจนลักษณะและคุณสมบัติของเครื่องวัดสัญญาณไฟฟ้าหัวใจ เพื่อนำไปใช้ในระบบบันทึกสัญญาณไฟฟ้าหัวใจต่อไป

2.1 การกำเนิดและแบบจำลองของแหล่งกำเนิดสัญญาณไฟฟ้าหัวใจ

การทำงานของระบบหัวใจสามารถที่จะให้กำเนิดเวกเตอร์ของความต่างศักย์ที่มีขนาดและทิศทางต่างๆตามจังหวะการทำงานของระบบหัวใจ ซึ่งสามารถทำให้เกิดความต่างศักย์ระหว่างจุด 2 จุดบนผิวหนังได้ ไอน์โรเฟ่น(Einthoven)ได้กำหนดสามเหลี่ยมด้านเท่าที่ใช้ในการจำลองแหล่งกำเนิดสัญญาณไฟฟ้าหัวใจบนร่างกาย ซึ่งต่อมาเรียกว่าสามเหลี่ยมด้านเท่าของไอน์โรเฟ่น (Einthoven Equilateral Triangle) แสดงในรูปที่ 2.1 ไอน์โรเฟ่นได้พิจารณาว่าอวัยวะหัวใจอยู่ภายในสามเหลี่ยมนี้ โดย RA หมายถึงตำแหน่งแขนขวา LA หมายถึงตำแหน่งแขนซ้าย และ LL หมายถึงตำแหน่งขาซ้าย เวกเตอร์ของความต่างศักย์ที่เกิดภายในสามเหลี่ยมด้านเท่าของไอน์โรเฟ่นคือเวกเตอร์ที่เกิดขึ้นเนื่องจากการทำงานของระบบหัวใจหรือเรียกว่าเวกเตอร์ของไอน์โรเฟ่น (Einthoven Vector) แต่ละด้านของสามเหลี่ยมได้ถูกกำหนดให้เป็นแหล่งกำเนิดสัญญาณไฟฟ้าหัวใจแบบมาตรฐาน(Standard Limb Lead)ที่ใช้วัดซึ่งประกอบไปด้วย I Lead, II Lead และ III Lead

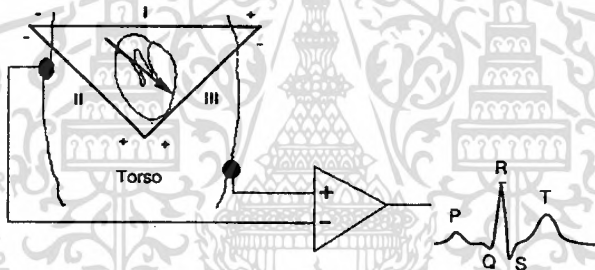


รูปที่ 2.1 แสดงสามเหลี่ยมด้านเท่าของไอน์โรเฟ่นและการฉายเวกเตอร์ของไอน์โรเฟ่นลงบนแต่ละด้านของสามเหลี่ยม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เวกเตอร์ของความต่างศักย์ที่เกิดขึ้นแต่ละด้านก็คือการฉายเวกเตอร์ของอินโฟเฟนลงบนด้านนั้นๆ การเปลี่ยนแปลงขนาดและทิศทางของเวกเตอร์ของอินโฟเฟนจะมีผลให้ขนาดและทิศทางของเวกเตอร์บนแต่ละด้านของสามเหลี่ยมเปลี่ยนแปลงไปด้วย

วิธีพื้นฐานในการวัดสัญญาณไฟฟ้าหัวใจจากผิวหนังสามารถแสดงได้ในรูปที่ 2.2 การวัดขนาดของเวกเตอร์ที่เกิดขึ้นจะใช้วงจรขยายความแตกต่าง(Differential Amplifier) ณ ช่วงเวลาใดๆ ถ้าทิศทางของเวกเตอร์ชี้ไปในทิศทางของขั้ววัดบวกของวงจรขยาย สัญญาณไฟฟ้าที่ได้ก็จะมีค่าบวก ในทางตรงกันข้ามถ้าเวกเตอร์ชี้ไปในทิศทางของขั้ววัดลบของวงจรขยาย สัญญาณไฟฟ้าที่ได้ก็จะมีค่าลบ โดยที่ขนาดความแรงของสัญญาณที่เอาต์พุตของวงจรขยาย ณ เวลาใดๆ จะขึ้นอยู่กับขนาดเวกเตอร์ที่เกิดขึ้น ณ เวลานั้นๆ เมื่อเวลาผ่านไปจึงทำให้สามารถสังเกตเห็นลักษณะการทำงานของระบบหัวใจได้ด้วยระดับของแรงดันตามเวลาที่เกิดขึ้น โดยลักษณะสัญญาณที่เอาต์พุตของวงจรขยายจะประกอบไปด้วยคลื่น P, QRS และ T ซึ่งเป็นข้อมูลที่สำคัญในการวิเคราะห์ระบบการทำงานของหัวใจ



รูปที่ 2.2 แสดงวิธีพื้นฐานในการวัดสัญญาณไฟฟ้าหัวใจจากผิวหนังในตำแหน่งของ II Lead โดยประมาณ

2.2 การวัดสัญญาณไฟฟ้าหัวใจ

โดยทั่วไปการวัดสัญญาณไฟฟ้าหัวใจสามารถทำได้ 2 ลักษณะคือ การวัดแบบเวกเตอร์คาร์ดิโอกราฟ(Vectorcardiograph) และการวัดแบบอิเล็กทรอนิกส์คาร์ดิโอกราฟ(Electrocardiograph) การวัดแบบเวกเตอร์คาร์ดิโอกราฟคือการวัดการเปลี่ยนแปลงขนาดของเวกเตอร์ของความต่างศักย์ที่เกิดขึ้นบนแกนหนึ่งเทียบกับอีกแกนหนึ่ง โดยพิจารณาจาก 3 แกนที่ตั้งฉากกัน สัญญาณที่เกิดขึ้นนี้เรียกว่าเวกเตอร์คาร์ดิโอแกรม(Vectorcardiogram : VCG) ซึ่งมีอยู่ด้วยกัน 3 ระนาบคือ ระนาบที่มองทางด้านหน้า ด้านซ้าย และด้านบน การวัดวิธีนี้จำเป็นต้องใช้ตำแหน่งในการวัดมาก การสร้างอุปกรณ์ที่ใช้ในการวัดค่อนข้างยุ่งยากซับซ้อน และสัญญาณไฟฟ้าหัวใจที่วัดได้จำเป็นต้องใช้แพทย์ผู้เชี่ยวชาญในการวินิจฉัย ส่วนการวัดแบบอิเล็กทรอนิกส์คาร์ดิโอกราฟคือการวัดการเปลี่ยนแปลงขนาดเวกเตอร์ของความต่างศักย์ที่เกิดขึ้นในแนวแกนใดๆเทียบกับเวลา สัญญาณที่เกิดขึ้นนี้เรียกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อิเล็กทรอนิกส์หัวใจ (Electrocardiogram : ECG) การวัดวิธีนี้ใช้ตำแหน่งในการวัดไม่มากนัก การสร้างอุปกรณ์ที่ใช้ในการวัดไม่ยุ่งยากซับซ้อน สามารถเลือกวัดสัญญาณเพื่อการวินิจฉัยได้หลายๆแบบ และสามารถวินิจฉัยได้ง่าย ปัจจุบันเป็นที่นิยมใช้กันอย่างกว้างขวาง

การวัดสัญญาณไฟฟ้าหัวใจแบบอิเล็กทรอนิกส์หัวใจกราฟเพื่อการวินิจฉัยโรคเกี่ยวกับระบบการทำงานของหัวใจสามารถแบ่งตามจุดประสงค์ของการวัดได้ 2 ประเภทคือ การวัดเพื่อการวินิจฉัยคนไข้ข้างเตียงแบบมาตรฐาน(Standard Clinical ECG)และการวัดเพื่อการมอนิเตอร์ (Monitoring ECG)

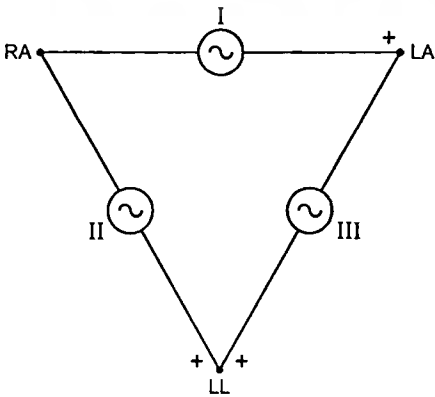
2.2.1 การวัดเพื่อการวินิจฉัยคนไข้ข้างเตียงแบบมาตรฐาน

การวัดเพื่อการวินิจฉัยคนไข้ข้างเตียงแบบมาตรฐานนั้น เป็นการวัดสัญญาณไฟฟ้าหัวใจสำหรับผู้ป่วยเพื่อการวินิจฉัยโดยละเอียด โดยตำแหน่งที่ทำการวัดสัญญาณได้ถูกกำหนดไว้เป็นมาตรฐานแล้ว แพทย์ผู้เชี่ยวชาญนิยมที่จะบันทึกสัญญาณที่วัดด้วยวิธีนี้เพื่อการวินิจฉัยโดยละเอียดต่อไป วิธีการวัดเพื่อการวินิจฉัยคนไข้ข้างเตียงแบบมาตรฐานสามารถแบ่งออกได้เป็น 3 วิธีคือ วิธีการวัดแบบ Standard Limb Lead แบบ Augmented Limb Lead และแบบ Unipolar Chest Lead

2.2.1.1 วิธีการวัดแบบ Standard Limb Lead

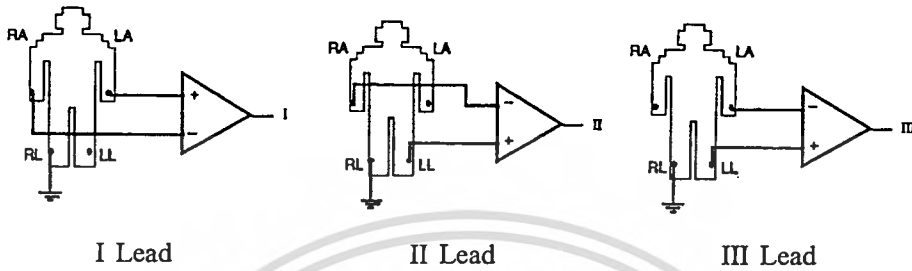
วิธีการวัดแบบ Standard Limb Lead หรือเรียกอีกชื่อหนึ่งว่าแบบ Bipolar Limb Lead เป็นมาตรฐานการวัดสัญญาณไฟฟ้าหัวใจอย่างง่าย ประกอบไปด้วย I Lead, II Lead และ III Lead ซึ่งได้จากการหาขนาดของเวกเตอร์บนสามเหลี่ยมด้านเท่าของไอน์ไซเฟินในรูปที่ 2.1 นั้นเอง ถ้าพิจารณาให้เวกเตอร์ที่เกิดขึ้นบนแต่ละด้านของสามเหลี่ยมเป็นแหล่งกำเนิดแรงดัน(Voltage Source) ของ I Lead, II Lead และ III Lead แสดงในรูปที่ 2.3 จากกฎของเคอร์ชอฟสามารถแสดงได้ว่า

$$II - I - III = 0 \tag{2.1}$$



รูปที่ 2.3 แสดงแบบจำลองแหล่งกำเนิดสัญญาณไฟฟ้าหัวใจแบบ Standard Limb Lead

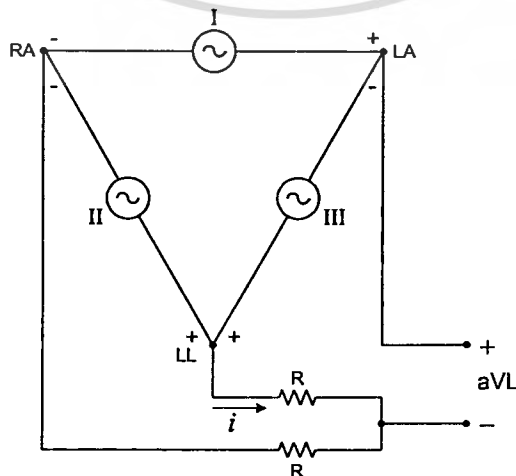
นั่นคือในการวัดสัญญาณไฟฟ้าหัวใจแบบ Standard Limb Lead นั้น ไม่จำเป็นจะต้องทำการวัดทุกสัญญาณ สามารถทำการวัดเพียง 2 สัญญาณเพื่อคำนวณสัญญาณที่เหลือได้ ในทางปฏิบัติสามารถที่จะทำการวัดสัญญาณไฟฟ้าหัวใจแบบ Standard Limb Lead ทั้ง I Lead, II Lead และ III Lead โดยการติขั้ววัดของวงจรขยายค่าความแตกต่างแสดงในรูปที่ 2.4



รูปที่ 2.4 แสดงวิธีการวัดสัญญาณไฟฟ้าหัวใจแบบ Standard Limb Lead

2.2.1.2 วิธีการวัดแบบ Augmented Limb Lead

วิธีการวัดแบบ Augmented Limb Lead หรือเรียกอีกชื่อหนึ่งว่าแบบ Unipolar Limb Lead เป็นการพิจารณาเวกเตอร์ของความต่างศักย์ในแนวดิ่งจากกับเวกเตอร์ของ I Lead, II Lead และ III Lead ซึ่งประกอบด้วย aVF Lead, aVL Lead และ aVR Lead ตามลำดับ โดยการวัด aVF Lead หมายถึงการวัดขนาดสัญญาณไฟฟ้าหัวใจระหว่างตำแหน่ง LL (ขั้วไฟฟ้าบวก) เทียบกับค่าเฉลี่ยของความต่างศักย์ระหว่างตำแหน่ง RA และ LA ในทำนองเดียวกัน การวัด aVL Lead หมายถึงการวัดขนาดสัญญาณไฟฟ้าหัวใจระหว่างตำแหน่ง LA (ขั้วไฟฟ้าบวก) เทียบกับค่าเฉลี่ยของความต่างศักย์ระหว่างตำแหน่ง RA และ LL พิจารณาแบบจำลองการวัดสัญญาณไฟฟ้าหัวใจแบบ Augmented Limb Lead aVL ในรูปที่ 2.5



รูปที่ 2.5 แสดงแบบจำลองการวัดสัญญาณไฟฟ้าหัวใจแบบ Augmented Limb Lead aVL

จากกฎของ KVL สามารถแสดงได้ว่า

$$\begin{aligned} iR + iR - II &= 0 \\ iR &= \frac{II}{2} \end{aligned} \quad (2.2)$$

และ

$$\begin{aligned} -iR + III + aVL &= 0 \\ aVL &= iR - III \end{aligned} \quad (2.3)$$

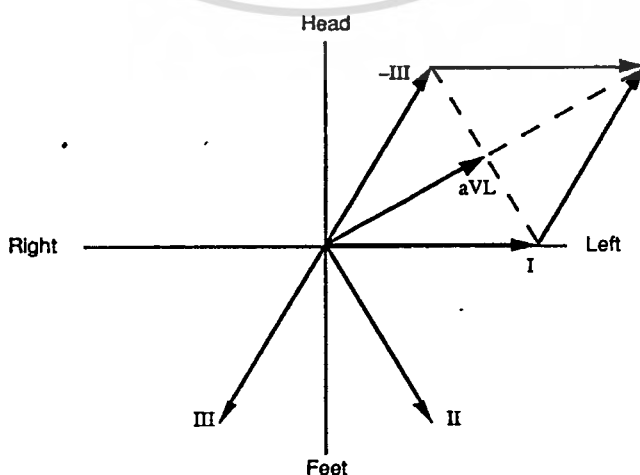
จากสมการที่ (2.2) และ (2.3) จะได้ว่า

$$aVL = \frac{II - 2III}{2} \quad (2.4)$$

จากสมการที่ (2.1) และ (2.4) จะได้ว่า

$$aVL = \frac{I - III}{2} \quad (2.5)$$

นั่นคือ aVL Lead สามารถคำนวณได้จาก I Lead และ III Lead ของ Standard Limb Lead โดยสามารถแสดงความสัมพันธ์ของเวกเตอร์ได้ในรูปที่ 2.6 จะเห็นว่าทิศทางเวกเตอร์ของ aVL Lead จะตั้งฉากกับทิศทางเวกเตอร์ของ II Lead ในทำนองเดียวกัน aVR Lead สามารถคำนวณได้จาก I Lead และ II Lead และ aVF Lead สามารถคำนวณได้จาก II Lead และ III Lead ตามสมการที่ (2.6) และ (2.7) ตามลำดับ



รูปที่ 2.6 แสดงเวกเตอร์ของ aVL Lead ที่หาได้จากเวกเตอร์ของ I Lead และ III Lead

$$aVR = -\frac{I + II}{2} \tag{2.6}$$

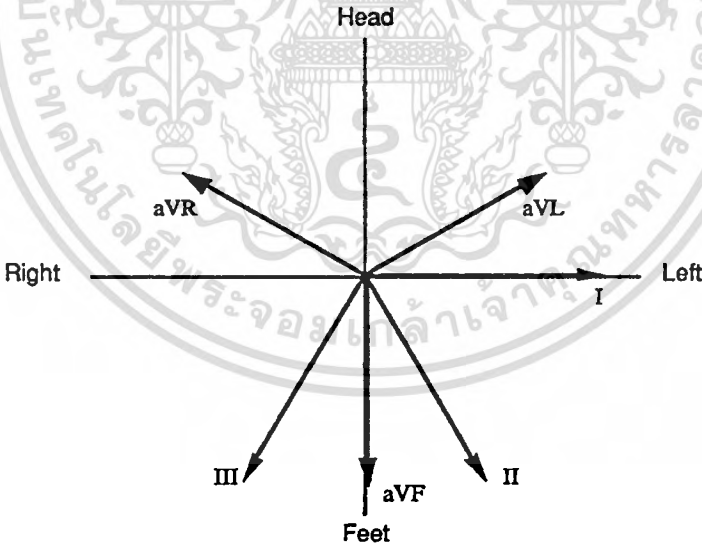
$$aVF = \frac{II + III}{2} \tag{2.7}$$

ในทางกลับกัน ผลจากสมการที่ (2.5) (2.6) และ (2.7) ทำให้สามารถที่จะคำนวณ Standard Limb Lead จาก Augmented Limb Lead ได้โดยสมการที่ (2.8) (2.9) และ (2.10) รูปที่ 2.7 แสดงทิศทางและขนาดของเวกเตอร์ของ Standard และ Augmented Limb Lead

$$I = \frac{2}{3}(aVL - aVR) \tag{2.8}$$

$$II = \frac{2}{3}(aVF - aVR) \tag{2.9}$$

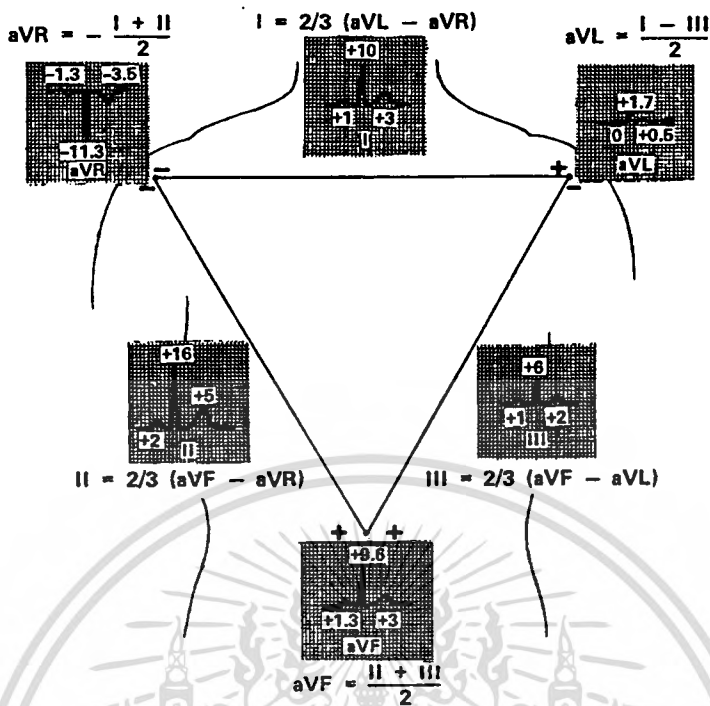
$$III = \frac{2}{3}(aVF - aVL) \tag{2.10}$$



รูปที่ 2.7 แสดงทิศทางและขนาดของเวกเตอร์แบบ Standard และ Augmented Limb Lead

การคำนวณสัญญาณไฟฟ้าหัวใจแบบ Augmented Limb Lead จาก Standard Limb Lead ในทางปฏิบัติจะต้องคูณด้วยค่าแก้ไข 0.87 [1] ในทางกลับกัน การคำนวณสัญญาณไฟฟ้าหัวใจแบบ Standard Limb Lead จาก Augmented Limb Lead จะต้องคูณด้วยค่าแก้ไข 1/0.87 หรือ 1.15 [1] จากรูปที่ 2.8 สามารถแสดงตัวอย่างการคำนวณเฉพาะส่วนของคลื่น R ได้คือ

เอกสารนี้เป็นเอกสารของกองการแพทย์ทหารกองพลที่ 11 กรมการแพทย์ทหารบก กองบัญชาการกองทัพบก กรุงเทพมหานคร ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

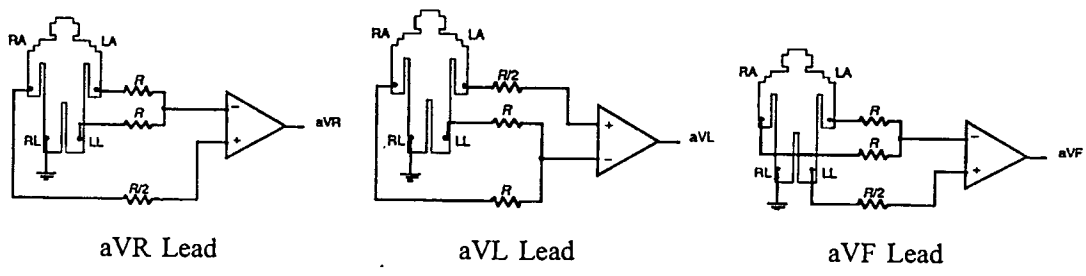


รูปที่ 2.8 แสดงตัวอย่างสัญญาณไฟฟ้าหัวใจและความสัมพันธ์ระหว่างสัญญาณไฟฟ้าหัวใจแบบ Standard และ Augmented Limb Lead

$$\begin{aligned} aVR &= -\left(\frac{I + II}{2}\right)(0.87) = -\left(\frac{10 + 16}{2}\right)(0.87) = -11.3 \\ aVL &= \left(\frac{I - III}{2}\right)(0.87) = \left(\frac{10 - 6}{2}\right)(0.87) = +1.7 \\ aVF &= \left(\frac{II + III}{2}\right)(0.87) = \left(\frac{16 + 6}{2}\right)(0.87) = +9.6 \\ I &= \frac{2}{3}(aVL - aVR)(1.15) = \frac{2}{3}(1.7 + 11.3)(1.15) = +10 \\ II &= \frac{2}{3}(aVF - aVR)(1.15) = \frac{2}{3}(9.6 + 11.3)(1.15) = +16 \\ III &= \frac{2}{3}(aVF - aVL)(1.15) = \frac{2}{3}(9.6 - 1.7)(1.15) = +6 \end{aligned}$$

สำหรับวิธีการวัดสัญญาณไฟฟ้าหัวใจแบบ Augmented Limb Lead ในทางปฏิบัติ แสดงในรูปที่ 2.9 ตัวต้านทานค่า R/2 ที่ต่ออยู่ที่ขั้ววัดบวกของวงจรขยายค่าความแตกต่างก็คือค่าความต้านทานเทวินิน(Thevenin Resistance)ที่คำนวณได้จากรูปที่ 2.5 มีไว้เพื่อสมดุลย์ค่าความต้านทานที่อินพุตของวงจรขยายความแตกต่างและให้ค่า Common Mode Rejection Ratio (CMRR) ที่ดีที่สุดนั่นเอง

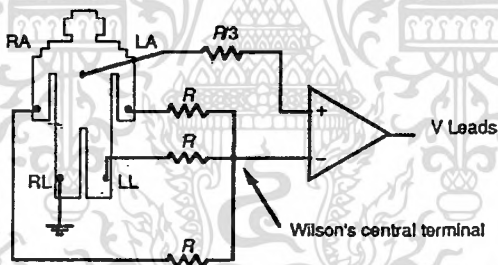
เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



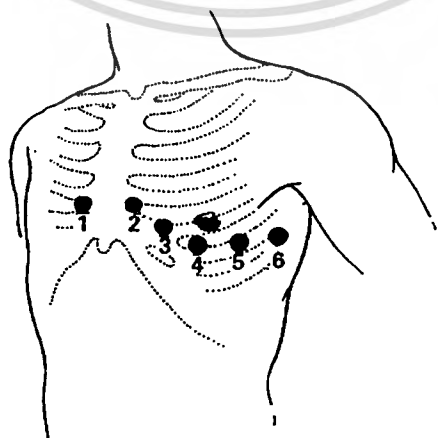
รูปที่ 2.9 แสดงวิธีการวัดสัญญาณไฟฟ้าหัวใจแบบ Augmented Limb Lead

2.2.1.3 วิธีการวัดแบบ Unipolar Chest Lead

วิธีการวัดสัญญาณไฟฟ้าหัวใจแบบ Unipolar Chest Lead เป็นการวัดขนาดของสัญญาณไฟฟ้าหัวใจระหว่างตำแหน่งใดจุดบนหน้าอก(ขั้ววัดบวก)เทียบกับค่าเฉลี่ยของความต่างศักย์ของตำแหน่ง RA, LA และ LL ทั้ง 3 ตำแหน่ง โดยสามารถแสดงวิธีการวัดในทางปฏิบัติได้ในรูปที่ 2.10 การวัดสัญญาณไฟฟ้าหัวใจแบบ Unipolar Chest Lead นี้ประกอบด้วย 6 Lead มาตรฐานคือ V₁ Lead ถึง V₆ Lead ซึ่งก็คือการกำหนดตำแหน่งของขั้ววัดบวกให้อยู่ในตำแหน่งต่างๆบริเวณหน้าอก 6 ตำแหน่งแสดงในรูปที่ 2.11 และรูปที่ 2.12

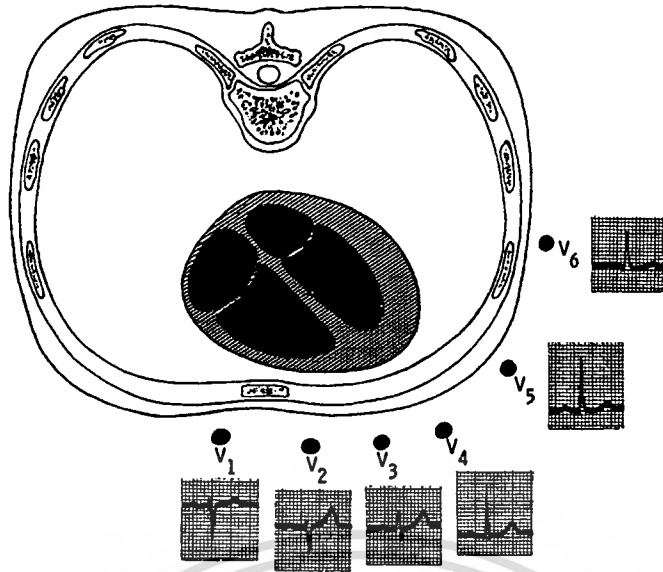


รูปที่ 2.10 แสดงวิธีการวัดสัญญาณไฟฟ้าหัวใจแบบ Unipolar Chest Lead



รูปที่ 2.11 แสดงตำแหน่งการติดขั้ววัดบวกบนหน้าอกของวิธีการวัดสัญญาณไฟฟ้าหัวใจแบบ

Unipolar Chest Lead V₁ ถึง V₆ เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 แสดงตัวอย่างสัญญาณไฟฟ้าหัวใจแบบ Unipolar Chest Lead V_1 ถึง V_6 และตำแหน่งการติดขั้ววัดบนหน้าอกโดยมองภาพตัดขวางของลำตัว

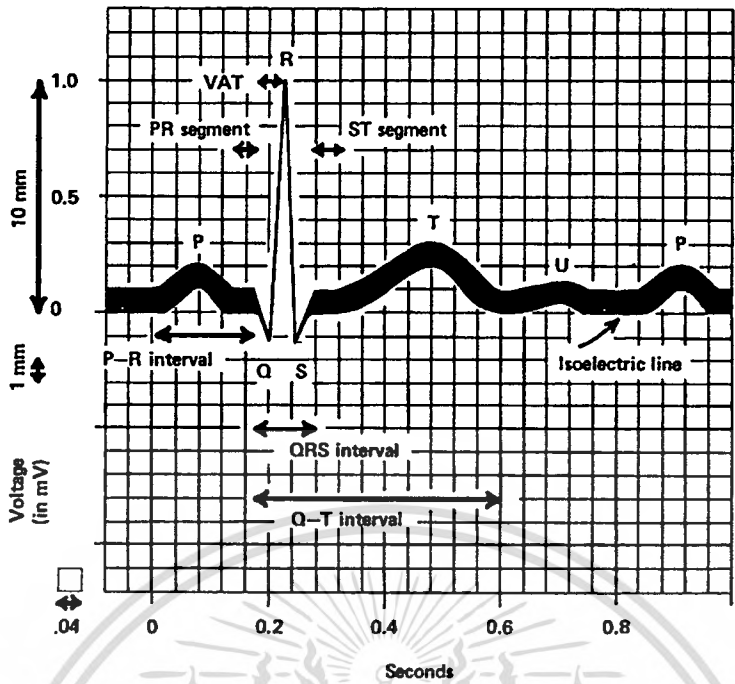
2.2.2 การวัดเพื่อการมอนิเตอร์

การวัดเพื่อการมอนิเตอร์มีจุดประสงค์เพื่อใช้ในการวัดสัญญาณไฟฟ้าหัวใจจากผู้ป่วยฉุกเฉินหรือผู้ป่วยที่จำเป็นจะต้องทำการเคลื่อนย้ายบ่อย ทั้งนี้เพื่อเป็นการพิจารณาจังหวะและอัตราการเต้นของหัวใจของผู้ป่วยเป็นหลัก ดังนั้นตำแหน่งที่ทำการวัดสัญญาณจึงควรเป็นตำแหน่งที่สามารถให้ขนาดคลื่น R ที่แรง เพื่อให้อัตราส่วนของสัญญาณไฟฟ้าหัวใจต่อสัญญาณรบกวน (Signal to Noise Ratio : S/N) มีค่าสูง ทำให้สามารถวินิจฉัยจังหวะและอัตราการเต้นหัวใจของผู้ป่วยได้อย่างถูกต้อง โดยทั่วไปการวัดสัญญาณไฟฟ้าหัวใจเพื่อการมอนิเตอร์นิยมที่จะใช้ II Lead เนื่องจากจะให้ขนาดของคลื่น R ที่สูง[2] และตำแหน่งการติดขั้ววัดจะไม่เป็นไปตามมาตรฐาน แต่จะใช้วิธีการประมาณตำแหน่งของ II Lead บริเวณหน้าอกของผู้ป่วย

2.3 ลักษณะและคุณสมบัติของสัญญาณไฟฟ้าหัวใจ

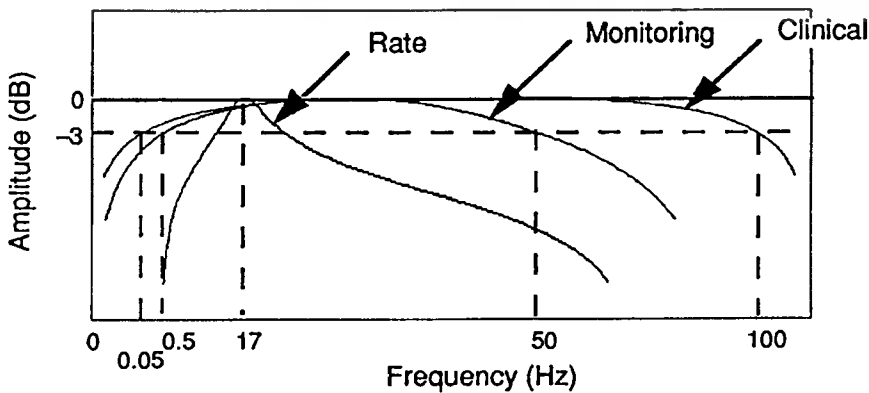
ลักษณะสัญญาณไฟฟ้าหัวใจของคนปกติประมาณหนึ่งรอบการทำงานของระบบหัวใจ แสดงได้ในรูปที่ 2.13 แต่ละช่วงของสัญญาณจะมีชื่อเรียกแทนด้วยตัวอักษร P, Q, R, S, T และ U ซึ่งแต่ละช่วงดังกล่าวจะสัมพันธ์กับการทำงานของระบบหัวใจในช่วงต่างๆ ขนาดและช่วงเวลาระหว่างตำแหน่งต่างๆสามารถบอกถึงสภาวะการทำงานของระบบหัวใจได้ สัญญาณไฟฟ้าหัวใจอาจมีขนาดประมาณ 10 μV สำหรับทารกในครรภ์[2] และอาจมีขนาดสูงถึง 5 mV สำหรับผู้ใหญ่[2] แต่อย่างไรก็ดีขนาดสัญญาณไฟฟ้าหัวใจที่วัดได้จากผู้ป่วยโดยทั่วไปจะขนาดประมาณ 1 mV [2]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 แสดงลักษณะสัญญาณไฟฟ้าหัวใจของคนปกติ

รูปที่ 2.14 แสดงย่านความถี่ของสัญญาณไฟฟ้าหัวใจที่เหมาะสมสำหรับการประยุกต์ใช้งานแบบต่างๆ โดยสำหรับการบันทึกสัญญาณไฟฟ้าหัวใจของคนไข้ข้างเตียง(Clinical) ย่านความถี่ที่อุปกรณ์รับสัญญาณจากร่างกายสามารถตอบสนองได้ควรอยู่ระหว่าง 0.05-100 Hz [2] สำหรับการวัดเพื่อการมอนิเตอร์(Monitoring)ควรอยู่ระหว่าง 0.5-50 Hz [2] และสำหรับการวัดอัตราการเต้นของหัวใจ(Heart Rate) สัญญาณที่ได้จากอุปกรณ์รับสัญญาณจากร่างกายที่มีการตอบสนองความถี่ในช่วงกว้าง จะนำมาผ่านตัวกรองความถี่แบบแบนด์พาส(Bandpass Filter)ที่มีความถี่ศูนย์กลาง(Center Frequency)อยู่ที่ 17 Hz และมีค่า Q ประมาณ 3 ถึง 4 [2] ซึ่งเป็นการกรองเฉพาะคลื่น QRS เพื่อนำไปใช้ในการคำนวณอัตราการเต้นของหัวใจนั่นเอง



รูปที่ 2.14 แสดงช่วงความถี่ของสัญญาณไฟฟ้าหัวใจสำหรับการประยุกต์ใช้งานแบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้า เมื่อผู้ผู้ใดเห็นว่าเป็นประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 เครื่องวัดสัญญาณไฟฟ้าหัวใจ

โดยทั่วไปเครื่องวัดสัญญาณไฟฟ้าหัวใจจะประกอบไปด้วยส่วนที่สำคัญ 2 ส่วนคือ ขั้ววัดสัญญาณไฟฟ้าหัวใจจากผิวหนัง และวงจรวัดสัญญาณไฟฟ้าหัวใจ ซึ่งแต่ละส่วนมีรายละเอียดดังต่อไปนี้

2.4.1 ขั้ววัดสัญญาณไฟฟ้าหัวใจจากผิวหนัง

โดยทั่วไปขั้ววัดไฟฟ้า(Electrode)ที่ใช้ในการวัดสัญญาณไฟฟ้าจากผิวหนังจะต้องประกอบไปด้วยแผ่นโลหะนำไฟฟ้าและสารอิเล็กโทรไลต์ที่ใช้ในการเชื่อมทางไฟฟ้าระหว่างแผ่นโลหะกับผิวหนัง ตัวอย่างของโลหะนำไฟฟ้าที่ใช้ทำขั้ววัดไฟฟ้าได้แก่ แผ่นทองเหลืองชุบนิกเกิล (Nickel-Plated Brass) เป็นต้น ซึ่งยังให้ประสิทธิภาพในการวัดได้ไม่ด้อย เนื่องจากขั้ววัดไฟฟ้าสองขั้วที่ใช้ในการวัด ประกอบกับสารอิเล็กโทรไลต์ที่ใช้ในการเชื่อมทางไฟฟ้าระหว่างแผ่นโลหะกับผิวหนังและสารละลายของเกลือภายในลำตัวจะปฏิบัติตัวร่วมกันเสมือนแบตเตอรี่ ซึ่งจะทำให้เกิดแรงดันไฟฟ้าตรง(DC Voltage)ป้อนมาที่สัญญาณไฟฟ้าหัวใจที่วัดได้ ลักษณะนี้เรียกว่าขั้ววัดไฟฟ้าแบบโพลาไรซ์(Polarizable Electrode) แรงดันไฟฟ้าตรงที่ป้อนมาที่สัญญาณไฟฟ้าหัวใจนี้ เมื่อผ่านการขยายสัญญาณจากวงจรขยายอาจจะทำให้เกิดการอิ่มตัว(Saturation)ขึ้นได้ ซึ่งเป็นสิ่งที่ไม่พึงปรารถนาในการวัด

เนื่องจากกระแสไฟฟ้าที่เกิดขึ้นภายในร่างกายเกิดจากการไหลของไอออน แต่กระแสไฟฟ้าที่เกิดขึ้นภายในตัวนำทางไฟฟ้าเกิดจากการไหลของอิเล็กตรอน ดังนั้นขั้ววัดไฟฟ้าที่ใช้ในการวัดสัญญาณไฟฟ้าหัวใจจากผิวหนังจะต้องทำหน้าที่ในการแปลงกระแสของไอออนไปเป็นกระแสของอิเล็กตรอนนั่นเอง ปัจจุบันวัสดุที่นิยมใช้ทำขั้ววัดไฟฟ้าก็คือซิลเวอร์-ซิลเวอร์คลอไรด์ (Ag-AgCl) ซึ่งเป็นการนำชั้นของซิลเวอร์คลอไรด์มาติดอยู่กับแผ่นโลหะซิลเวอร์หรือเงินซึ่งใช้เป็นขั้วต่อกับสายไฟ ประกอบกับสารอิเล็กโทรไลต์ที่ใช้ในการเชื่อมทางไฟฟ้าระหว่างขั้ววัดไฟฟ้ากับผิวหนังประกอบด้วยไอออนของคลอไรด์ ไอออนของคลอไรด์จะเคลื่อนที่ภายในร่างกาย สารอิเล็กโทรไลต์และชั้นของซิลเวอร์คลอไรด์ และจะถูกแปลงไปเป็นกระแสของอิเล็กตรอนเคลื่อนที่ในแผ่นโลหะซิลเวอร์ผ่านเข้าสายไฟต่อไป ลักษณะนี้สามารถประมาณได้ว่าเป็นขั้ววัดไฟฟ้าแบบนอนโพลาไรซ์(Nonpolarizable Electrode) ซึ่งจะทำให้เกิดแรงดันไฟฟ้าตรงเนื่องจากการสัมผัสระหว่างขั้ววัดไฟฟ้ากับผิวหนังที่มีขนาดเล็กกว่าขั้ววัดไฟฟ้าแบบโพลาไรซ์มาก ทำให้ประสิทธิภาพในการวัดสัญญาณไฟฟ้าหัวใจสูง

2.4.2 วงจรวัดสัญญาณไฟฟ้าหัวใจ[3]

วงจรวัดสัญญาณไฟฟ้าหัวใจที่ให้ประสิทธิภาพในการวัดสัญญาณไฟฟ้าหัวใจที่สูง ควร

จะมีลักษณะและคุณสมบัติที่สำคัญดังต่อไปนี้

เอกสารนี้เป็นเอกสารประกอบการเรียนการสอน ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **มีค่า Common Mode Rejection Ratio (CMRR) สูง**

เนื่องจากสัญญาณไฟฟ้าหัวใจที่วัดได้จากขั้ววัดไฟฟ้าที่ติดอยู่บนผิวหนังโดยทั่วไปมีค่าประมาณเพียง 1 mV [2] ดังนั้นสัญญาณอื่นๆที่ไม่พึงปรารถนาจึงอาจเข้ามารบกวนได้ง่าย โดยเฉพาะอย่างยิ่งสัญญาณไฟฟ้ากระแสสลับความถี่ 50 Hz จากไฟฟ้าตามบ้านซึ่งอยู่ในช่วงความถี่ของสัญญาณไฟฟ้าหัวใจ(0.05-100Hz สำหรับการวินิจฉัยแบบมาตรฐาน[2])อาจแทรกเข้ามาได้ทั้งทางร่างกาย ทางขั้ววัดหรือสายวัดสัญญาณ เป็นต้น ปัญหานี้สามารถแก้ไขได้โดยการเลือกใช้อุปกรณ์ขยายความแตกต่างที่มีค่า Common Mode Rejection Ratio (CMRR) ที่สูง และทำการวัดสัญญาณไฟฟ้าหัวใจในลักษณะดิฟเฟอเรนเชียลโหมด(Differential Mode) ซึ่งเป็นการขยายสัญญาณเฉพาะส่วนที่เป็นความแตกต่างเท่านั้น ส่วนสัญญาณความถี่ 50 Hz ที่เหมือนกันของแต่ละอินพุตจะหักล้างกันในลักษณะของคอมมอนโหมด(Common Mode) เนื่องจากค่า CMRR ก็คืออัตราส่วนระหว่างค่า Differential Mode Gain และค่า Common Mode Gain และการออกแบบวงจรขยายความแตกต่างในเบื้องต้นจะกำหนดให้ค่า Differential Mode Gain มีค่าตามต้องการ ดังนั้นการออกแบบวงจรขยายความแตกต่างให้มีค่า CMRR สูงที่สุดสามารถทำได้โดยการปรับแต่งวงจรให้ค่า Common Mode Gain ของวงจรมีค่าต่ำที่สุด(เท่ากับ 0 ในอุดมคติ) ซึ่งจะกล่าวถึงต่อไป

ในความเป็นจริงทางปฏิบัติเป็นการยากที่จะกำจัดสัญญาณรบกวนความถี่ 50 Hz ออกไปได้หมด ทั้งนี้อาจเนื่องมาจากความแตกต่างของขนาดและเฟสของสัญญาณความถี่ 50 Hz ที่เกิดขึ้นในแต่ละอินพุต ทำให้สัญญาณไฟฟ้าหัวใจที่วัดได้ยังคงมีสัญญาณความถี่ 50 Hz หลงเหลืออยู่ ดังนั้นภายในวงจรวัดสัญญาณไฟฟ้าหัวใจจึงควรมีวงจรกรองความถี่ 50 Hz ไว้ป้องกันเสมอ

- **มีค่าอินพุตอิมพีแดนซ์สูงมาก**

เนื่องจากการวัดสัญญาณไฟฟ้าหัวใจโดยใช้ขั้ววัดไฟฟ้าจะทำให้เกิดความต้านทานขึ้นบริเวณรอยสัมผัสระหว่างขั้ววัดไฟฟ้าและผิวหนัง ค่าความต้านทานนี้จะมีค่าที่สูงมาก ดังนั้นวงจรขยายจึงต้องมีค่าอินพุตอิมพีแดนซ์สูงกว่าค่าความต้านทานที่เกิดขึ้นระหว่างรอยสัมผัสระหว่างขั้ววัดไฟฟ้าและผิวหนังให้มากที่สุด เพื่อให้สัญญาณไฟฟ้าหัวใจที่วัดได้มีการสูญเสียที่รอยสัมผัสให้น้อยที่สุด

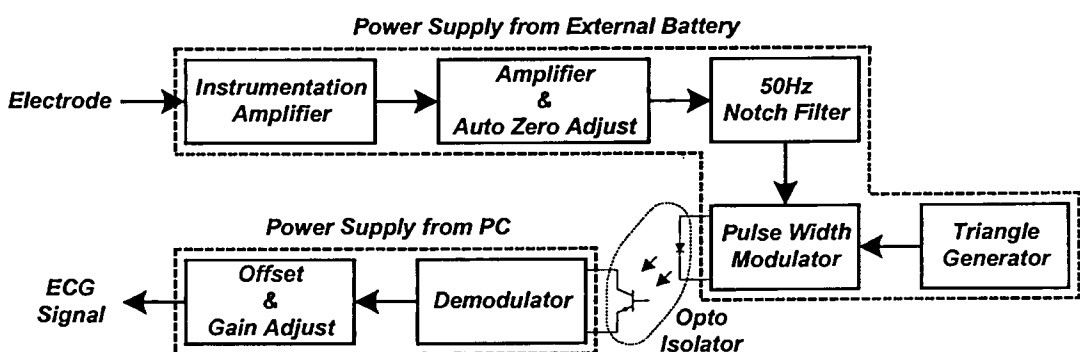
- **ป้องกันอันตรายจากกระแสไฟฟ้ารั่วไหล**

เนื่องจากในการวัดสัญญาณไฟฟ้าหัวใจต้องใช้ขั้ววัดไฟฟ้าสัมผัสโดยตรงกับผู้ป่วย ซึ่งถ้าหากเครื่องมือวัดมีกระแสไฟฟ้ารั่วไหลออกมาในปริมาณที่มากพอ ก็อาจทำให้ผู้ป่วยได้รับอันตรายขึ้นได้ นอกจากนี้ยังอาจทำให้สัญญาณไฟฟ้าหัวใจวัดได้ผิดเพี้ยนไปจากความเป็นจริง ดังนั้นวงจรขยายที่ออกแบบจึงควรมีส่วนที่ทำหน้าที่ในการแยกกราวด์ของเครื่องมือวัดออกจากส่วนของวงจรขยายสัญญาณภาคแรก

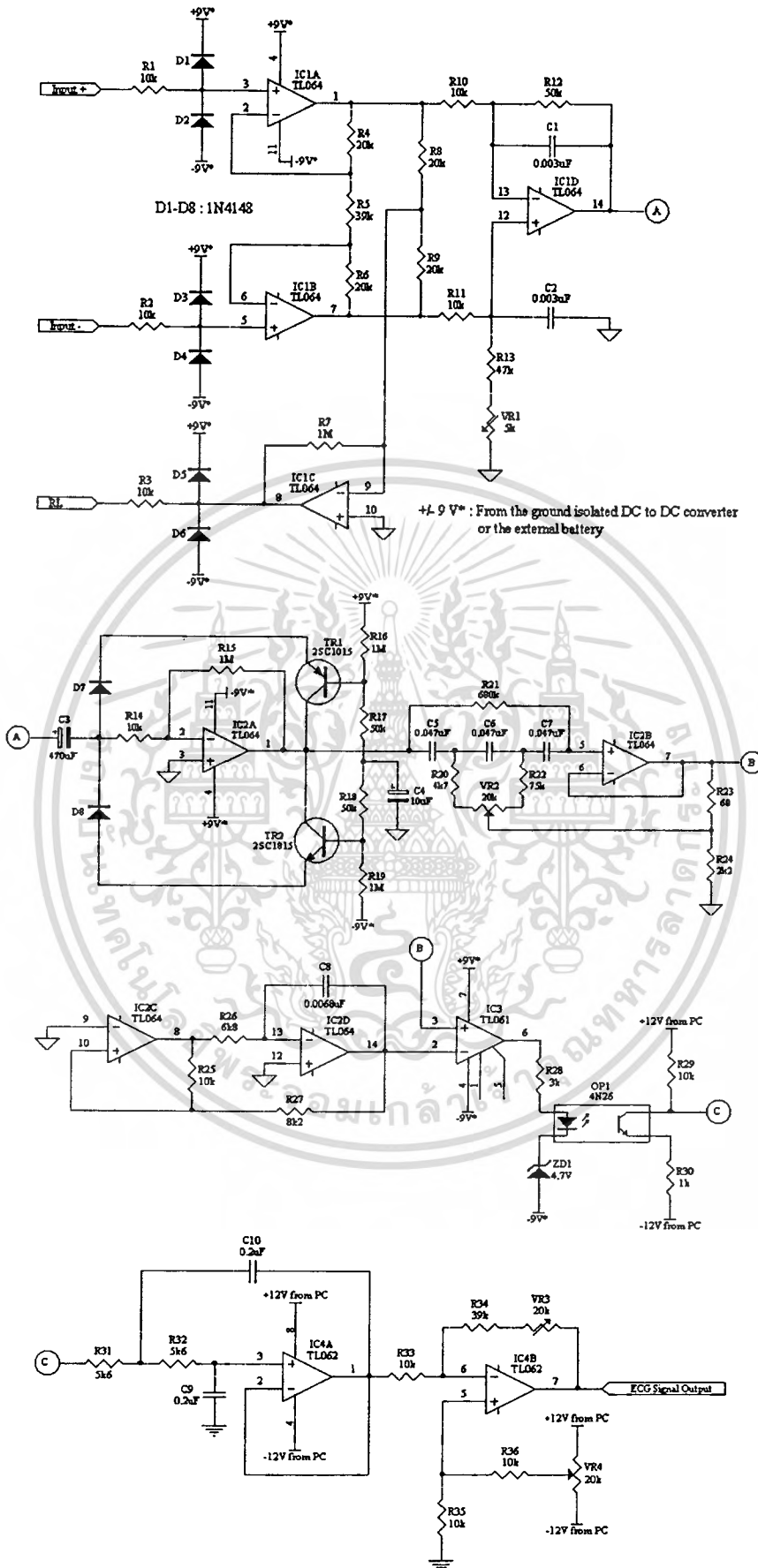
• ตอบสนองความถี่ต่ำได้ดี

เนื่องจากสัญญาณไฟฟ้าหัวใจมีความถี่ต่ำถึง 0.05 Hz ดังนั้นวงจรขยายจะต้องตอบสนองความถี่ต่ำได้ดี อาจกล่าวได้ว่าวงจรขยายที่ออกแบบจะต้องเป็นวงจรขยายสัญญาณไฟตรง(DC Amplifier)

รูปที่ 2.15 และรูปที่ 2.16 แสดงบล็อกไดอะแกรมและวงจรวัดสัญญาณไฟฟ้าหัวใจที่ใช้ในระบบบันทึกสัญญาณไฟฟ้าหัวใจ ซึ่งรายละเอียดในการพัฒนาและออกแบบวงจรแสดงใน[3] ในที่นี้จะขออธิบายโดยย่อเท่านั้น จากวงจร IC1 ประกอบกันขึ้นเป็นวงจรขยายความแตกต่างแบบอินสตรูเมนต์เชน(Instrumentation Amplifier) ที่มีค่าอินพุทอิมพีแดนซ์สูง มีค่า Differential Mode Gain ประมาณ 10 และสามารถปรับแต่งวงจรให้มีค่า CMRR ที่สูงได้ด้วย VR1 ซึ่งเป็นการปรับแต่งวงจรให้มีค่า Common Mode Gain ต่ำที่สุดดังที่กล่าวมานั่นเอง ขั้ววัดทั้ง 3 ขั้วประกอบด้วยขั้ววัดบวก(Input+) ขั้ววัดลบ(Input-) และขั้ววัดที่ตำแหน่งขาขวา(RL) การวัดแบบ Standard Limb Lead ขั้ววัดบวกและลบจะติดในตำแหน่งตามรูปที่ 2.4 ส่วนการวัดแบบ Augmented Limb Lead และ Unipolar Chest Lead ขั้ววัดบวกและลบจะต้องต่อตัวต้านทานเพิ่มและติดในตำแหน่งตามรูปที่ 2.9 และรูปที่ 2.10 ตามลำดับ อุปกรณ์ IC2A, D7, D8, TR1 และ TR2 ประกอบกันขึ้นเป็นวงจรขยายสัญญาณและปรับศูนย์อัตโนมัติ เพื่อเป็นการปรับระดับแรงดันไฟตรงที่ป้อนมา กับสัญญาณไฟฟ้าหัวใจอันเนื่องมาจากหน้าสัมผัสของขั้ววัด ไฟฟ้ากับผิวหนังให้อยู่ในระดับปกติ(ศูนย์) อย่างรวดเร็ว IC2B เป็นวงจรกรองความถี่แบบนอตช์(Notch Filter) ทำหน้าที่ในการกรองสัญญาณรบกวนความถี่ 50 Hz ที่ป้อนมา กับสัญญาณไฟฟ้าหัวใจ สัญญาณไฟฟ้าหัวใจจากตำแหน่ง B จะถูกมอดูเลตแบบพัลส์วีดท์(Pulse-Width Modulation : PWM)กับสัญญาณสามเหลี่ยม(กำเนิดโดย IC2C และ IC2D)โดย IC3 สัญญาณ PWM ที่ได้จะถูกแยกทางไฟฟ้าโดยการส่งผ่านทางแสงด้วย Opto Isolator OP1 จากนั้นจึงผ่านการดีมอดูเลต(Demodulation)ด้วยวงจรกรองความถี่ต่ำ (Low Pass Filter)ซึ่งทำหน้าที่โดย IC4A และส่วนท้ายสุด IC4B ทำหน้าที่ในการปรับศูนย์และปรับ



รูปที่ 2.15 แสดงบล็อกไดอะแกรมวงจรขยายสัญญาณไฟฟ้าหัวใจ



รูปที่ 2.16 แสดงวงจรวัดสัญญาณไฟฟ้าหัวใจ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัตราขยายของสัญญาณให้ได้ตามต้องการ วงจรที่ออกแบบได้ป้องกันอันตรายจากกระแสไฟฟ้ารั่วไหลจากแท่นเครื่องของคอมพิวเตอร์โดยให้วงจรส่วนแรก(ประกอบไปด้วยวงจรขยายความแตกต่างแบบอินสตรูเมนเตชัน วงจรขยายสัญญาณและปรับศูนย์อัตโนมัติ วงจรกรองความถี่แบบนอตร์ วงจรมอดูเลตแบบพัลส์วิดท์ และวงจรกำเนิดสัญญาณสามเหลี่ยม)ใช้ไฟเลี้ยงวงจรจากแบตเตอรี่ภายนอก และวงจรส่วนหลัง(ประกอบด้วยวงจรมอดูเลต และวงจรปรับศูนย์และอัตราขยาย)ใช้ไฟเลี้ยงจากคอมพิวเตอร์โดยตรง Opto Isolator OP1 จะเป็นตัวป้องกันกระแสไฟฟ้ารั่วไหลจากแท่นเครื่องของคอมพิวเตอร์ไปสู่วงจรในส่วนแรกซึ่งมีการสัมผัสกับผู้ป่วยโดยตรง



บทที่ 3

ทฤษฎีการแปลงเวฟเลต

สัญญาณหรือฟังก์ชันใดๆ สามารถแสดงให้อยู่ในรูปของการรวมเชิงเส้น (Linear Combination) ของฟังก์ชันมูลฐาน (Basis Function) ต่างๆ ซึ่งทำให้สามารถทำการวิเคราะห์ หรือแก้ปัญหาได้อย่างสะดวก ตัวอย่างเช่น การใช้ฟังก์ชันไซน์และโคไซน์ (Sine-Cosine Function) เป็นฟังก์ชันมูลฐานในการวิเคราะห์ฟูเรียร์ (Fourier Analysis) เป็นต้น การวิเคราะห์ เวฟเลต (Wavelet Analysis) ก็เป็นการแสดงฟังก์ชันใดๆ ให้อยู่ในรูปของการรวมเชิงเส้นของฟังก์ชัน มูลฐานเช่นเดียวกัน แต่ด้วยลักษณะพิเศษของฟังก์ชันมูลฐานเวฟเลต (Wavelet Basis Function) ซึ่งมี ลักษณะของการออสซิลเลท (Oscillation) ตามแกนแนวนอนและมีแอมพลิจูด (Amplitude) ที่ลดลงสู่ ศูนย์ทั้งทางด้านบวกและลบอย่างรวดเร็ว [4] (ต่างกับฟังก์ชันไซน์และโคไซน์ซึ่งแผ่ตามแกนแนวนอนไปสู่นอนันต์) จึงทำให้เหมาะที่จะใช้ในการวิเคราะห์สัญญาณที่มีลักษณะของ Non-Stationary ลักษณะของความไม่ต่อเนื่อง (Discontinuity) และลักษณะของปลายแหลมคม (Sharp Spike) ได้เป็นอย่างดี [5], [6], [7] เนื่องจากในโดเมนของเวฟเลต (Wavelet Domain) จะประกอบไปด้วยสัมประสิทธิ์ การกระจาย (Expansion Coefficients) หรือตัวคูณของฟังก์ชันมูลฐานเวฟเลตที่มีขนาดเล็กและลู่เข้าสู่ ศูนย์อย่างรวดเร็ว ซึ่งในโดเมนของฟูเรียร์ (Fourier Domain) ของฟังก์ชันดังกล่าวจะต้องประกอบ ด้วยสัมประสิทธิ์การกระจายหรือตัวคูณของฟังก์ชันไซน์และโคไซน์เป็นจำนวนมากมายถึงอนันต์ เมื่อมองในแง่ของการแปลงเพื่อเข้ารหัส (Transform Coding) แล้ว การแปลงเวฟเลต (Wavelet Transform) จึงเป็นการแปลงที่มีประสิทธิภาพที่สูง (เมื่อใช้กับสัญญาณที่มีลักษณะของ Non-Stationary ลักษณะของความไม่ต่อเนื่องและลักษณะของปลายแหลม) และประโยชน์ที่สำคัญ อีกประการหนึ่งของการแปลงเวฟเลตก็คือ คุณสมบัติความเป็น Localization (ต่างจากการแปลง ฟูเรียร์ซึ่งเป็นคุณสมบัติ Globalization) กล่าวคือ ความเปลี่ยนแปลงที่เกิดขึ้นในโดเมนของ เวฟเลตจะมีผลกับทำให้เกิดการเปลี่ยนแปลงในโดเมนเดิม (The Original Domain) เป็นช่วง (Interval) เท่านั้น ซึ่งเป็นผลเนื่องมาจากวิธีการวิเคราะห์แบบมัลติเรโซลูชัน (Multiresolution Analysis : MRA) [7] ซึ่งจะกล่าวในรายละเอียดต่อไป

3.1 ทฤษฎีการแปลงของสัญญาณ

ในปริภูมิ $L^2(R)$ ของ Continuous-Time Energy Function การส่ง (Mapping) สัญญาณ $f(t)$ จากโดเมนเดิมไปสู่โดเมนใหม่ (Transform Domain) นิยามด้วยผลคูณสเกลาร์ (Scalar Product)

ในสมการที่ (3.1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\langle f(t) | \sigma(t) \rangle = \int_{-\infty}^{\infty} f(t) \cdot \sigma^*(t) dt, \quad f(t) \in L_2(R) \quad (3.1)$$

ถ้ากำหนดให้ฟังก์ชันมูลฐาน $\sigma(t) = e^{j\omega t}$ หมายถึงการแปลงฟูเรียร์(Fourier Transform : FT)นั่นเอง ซึ่งนิยามโดยสมการที่ (3.2) นั่นคือสัญญาณ $f(t)$ ถูกแปลงจากโดเมนของเวลา(Time Domain)ไปสู่โดเมนของความถี่(Frequency Domain) ซึ่งในโดเมนของฟูเรียร์จะขึ้นกับพารามิเตอร์ของความถี่(ω)เท่านั้น ไม่ขึ้นกับพารามิเตอร์ของเวลา(t) อีกต่อไป ดังนั้นการเปลี่ยนแปลงที่เกิดขึ้นที่ตำแหน่งใดๆในโดเมนของฟูเรียร์จะมีผลตลอดช่วงในโดเมนของเวลา ลักษณะนี้เป็นคุณสมบัติแบบ Globalization ซึ่งสามารถแก้ไขโดยใช้การแปลงฟูเรียร์แบบแบ่งช่วงเวลา(Short Time Fourier Transform : STFT) นิยามโดยสมการที่ (3.3) โดยกำหนดให้ $\sigma(t) = w(t - \tau) \cdot e^{j\omega t}$ เมื่อ $w(t)$ คือวินโดว์ฟังก์ชัน(Window Function) และ τ เป็นพารามิเตอร์ที่ใช้บอกตำแหน่งของวินโดว์บนแกนเวลา การแปลงฟูเรียร์แบบแบ่งช่วงเวลาเป็นการแปลงฟูเรียร์ของสัญญาณ $f(t)$ เฉพาะส่วนที่วินโดว์ฟังก์ชัน $w(t - \tau)$ ครอบคลุมไว้ ลักษณะและความกว้างของวินโดว์ฟังก์ชัน $w(t)$ จะเป็นตัวกำหนดความละเอียดในการวิเคราะห์สัญญาณ วิธีนี้ทำให้การเปลี่ยนแปลงที่เกิดขึ้นในโดเมนของฟูเรียร์ที่ตำแหน่ง τ ใดๆมีผลกับในโดเมนของเวลาเฉพาะส่วนที่วินโดว์ครอบคลุมไว้เท่านั้น

$$FT_{f(t)}(\omega) = \langle f(t) | e^{j\omega t} \rangle \quad (3.2)$$

$$STFT_{f(t)}(\tau, \omega) = \langle f(t) | w(t - \tau) \cdot e^{j\omega t} \rangle \quad (3.3)$$

สำหรับการแปลงเวฟเลต(Wavelet Transform : WT)นิยามด้วยสมการที่ (3.4) พารามิเตอร์ a และ b ซึ่งเป็นพารามิเตอร์ที่กำหนดฟังก์ชันมูลฐานขึ้นใหม่โดยการสเกล(Scaling) และการเลื่อน(Translation)ไปตามแกนเวลาของฟังก์ชันมูลฐานต้นแบบ Mother Wavelet Function $\psi(t)$ พารามิเตอร์ a เปรียบเสมือนเป็นตัวกำหนดระดับความถี่ที่ใช้ในการวิเคราะห์สัญญาณ $f(t)$ ซึ่งเป็นการยืดหรือหดฟังก์ชันพื้นฐานต้นแบบตามแกนเวลาโดยที่ลักษณะของฟังก์ชันหรือรูปร่างของสัญญาณยังคงเดิม และพารามิเตอร์ b เป็นตัวกำหนดตำแหน่งบนแกนเวลาของฟังก์ชันมูลฐานเวฟเลตที่ใช้ในการวิเคราะห์สัญญาณ $f(t)$ ดังนั้นจึงกล่าวได้ว่าการแปลงเวฟเลตเป็นการแปลงไปสู่โดเมนของเวลาและความถี่(Time-Frequency Domain) นั่นคือการเปลี่ยนแปลงที่เกิดขึ้นในโดเมนของเวฟเลตที่สเกล a ใดๆสามารถที่จะบอกช่วงตำแหน่งที่เกิดการเปลี่ยนแปลงในโดเมนของเวลาได้ ซึ่งเป็นลักษณะของ Localization นั่นเอง

$$WT_{f(u)}(a,b) = \left\langle f(t) \left| \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) \right. \right\rangle \quad (3.4)$$

3.2 การวิเคราะห์แบบมัลติเรโซลูชัน

ทฤษฎีของเวฟเลตมีความคิดมาจากทฤษฎีการวิเคราะห์แบบมัลติเรโซลูชัน (Multiresolution Analysis : MRA) ซึ่งสามารถใช้ทฤษฎีของพีชคณิตเชิงเส้น (Linear Algebra) อธิบายได้เป็นอย่างดี กล่าวคือในปริภูมิของสัญญาณ $L^2(R)$ จะปรากฏปริภูมิย่อย (Subspaces) ของสัญญาณซ้อนทับกันอย่างต่อเนื่องไปดังสมการ

$$V_{-\infty} \subset \dots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \dots \subset V_{\infty} = L^2(R) \quad (3.5)$$

ซึ่ง $\bigcup_{j=-\infty}^{\infty} V_j$ อยู่กันอย่างหนาแน่น (Dense) ในปริภูมิ $L^2(R)$ โดยที่ $\bigcap_{j=-\infty}^{\infty} V_j = \{0\}$ (เมื่อ $\cup$ และ $\cap$ หมายถึงการยูเนียนและการอินเตอร์เซกชันตามลำดับ) แต่ละปริภูมิย่อย V_j จะประกอบไปด้วย ฟังก์ชันมูลฐาน (Basis Function) ในลักษณะที่ทำให้จำนวนฟังก์ชันมูลฐานในปริภูมิย่อย V_j มากขึ้นเป็นสองเท่า (Two Scale Property) เมื่อ j มีค่าเพิ่มขึ้นหนึ่งหรือ

$$f(t) \in V_j \Leftrightarrow f(2t) \in V_{j+1}, \quad j \in Z \quad (3.6)$$

เมื่อ Z คือเซตของจำนวนเต็ม ถ้ากำหนดให้ปริภูมิย่อยอ้างอิง V_0 ปรากฏฟังก์ชันมูลฐานต้นแบบ $\varphi(t)$ (เรียกว่า Scaling Function) กล่าวคือ

$$V_0 = \text{span}_k \{ \varphi(t-k) \}, \quad k \in Z \quad (3.7)$$

จากสมการที่ (3.6) สามารถกำหนดฟังก์ชันมูลฐานสำหรับปริภูมิย่อย V_j ได้โดยสมการ

$$\varphi_{j,k}(t) = 2^{j/2} \cdot \varphi(2^j t - k), \quad j, k \in Z \quad (3.8)$$

โดยพจน์ $2^{j/2}$ มีเพื่อกำหนดให้ค่าพลังงานของฟังก์ชันมูลฐานทุกฟังก์ชันในทุกรูปภูมิย่อย V_j มีค่าเท่ากับหนึ่งเสมอหรือเป็นการนอร์มัลไลซ์ (Normalized) โดยกำหนดให้ $\|\varphi_{j,k}(t)\| = 1$ เสมอ ฟังก์ชันมูลฐานที่เกิดขึ้นในปริภูมิย่อย V_j ใดๆ จะมีคุณสมบัติออร์โธนอร์มัล (Orthonormal Property) กล่าวคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$V_j = \text{span}_k \{\varphi_{j,k}(t)\}, \quad j, k \in Z \quad (3.9)$$

โดยที่ความสัมพันธ์ระหว่างฟังก์ชันมูลฐานในปริภูมิย่อย V_j ใดๆ แสดงโดย

$$\langle \varphi_{j,k}(t) | \varphi_{j,n}(t) \rangle = \delta_{k,n}, \quad k, n \in Z \quad (3.10)$$

เมื่อ δ_{xy} คือ Kronecker-Delta นิยามโดย

$$\delta_{xy} = \begin{cases} 1, & x = y \\ 0, & x \neq y \end{cases} \quad (3.11)$$

ถ้าสมมติให้ $f(t) \in V_j$ แล้ว สามารถแสดงฟังก์ชัน $f(t)$ ด้วยอนุกรมของ Scaling Function ได้ คือ

$$f(t) = \sum_n c_j(n) \cdot \varphi_{j,n}(t), \quad j, n \in Z \quad (3.12)$$

เมื่อ $c_j(n)$ คือ Coarse Expansion Coefficients จากสมการที่ (3.10) และ (3.12) สามารถหาค่าของ $c_j(n)$ ได้จาก

$$c_j(n) = \langle f(t) | \varphi_{j,n}(t) \rangle, \quad j, n \in Z \quad (3.13)$$

ดังนั้นเมื่อ $\varphi(t) \in V_0$ และ $V_0 \subset V_1$ นั่นคือสามารถแสดง Scaling Function $\varphi(t)$ ในรูปของการรวมเชิงเส้นของ $\sqrt{2}\varphi(2t-n) \in V_1$ ได้คือ

$$\varphi(t) = \sum_n h(n) \cdot \sqrt{2}\varphi(2t-n), \quad n \in Z, \quad h(n) \in l^2(Z) \quad (3.14)$$

เมื่อ $l^2(Z)$ คือปริภูมิของ Discrete-Time Energy Function ด้วยคุณสมบัติออร์โธนอร์มัลในสมการที่ (3.10) ดังนั้นค่าสัมประสิทธิ์ $h(n)$ ในสมการที่ (3.14) สามารถหาได้จากสมการ

$$h(n) = \langle \varphi(t) | \sqrt{2}\varphi(2t-n) \rangle, \quad n \in Z \quad (3.15)$$

เนื่องจาก $V_j \subset V_{j+1}$ ดังนั้น จะปรากฏปริภูมิย่อย W_j ที่เป็นออร์โธนอร์มัลคอมพลีเมนต์ (Orthonormal Complement) ในปริภูมิย่อย V_{j+1} กล่าวคือ $V_{j+1} = V_j \oplus W_j$ เช่นเดียวกันกับปริภูมิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ในการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ย่อย V_j ฟังก์ชันมูลฐานสำหรับปริภูมิย่อย W_j ใดๆสามารถแสดงด้วยฟังก์ชันมูลฐานต้นแบบ $\psi(t)$ (เรียกว่า Wavelet Function) ของปริภูมิอ้างอิง W_0 โดย

$$\psi_{j,k}(t) = 2^{j/2} \cdot \psi(2^j t - k), \quad j, k \in \mathbb{Z} \quad (3.16)$$

ฟังก์ชันมูลฐานที่เกิดขึ้นในปริภูมิย่อย W_j ใดๆจะมีคุณสมบัติออร์โธนอร์มัล กล่าวคือ

$$W_j = \text{span}_k \{\psi_{j,k}(t)\}, \quad j, k \in \mathbb{Z} \quad (3.17)$$

โดยที่ความสัมพันธ์ระหว่างฟังก์ชันมูลฐานในปริภูมิย่อย W_j ใดๆแสดงโดย

$$\langle \psi_{j,k}(t) | \psi_{m,n}(t) \rangle = \delta_{j,m} \delta_{k,n}, \quad j, k, m, n \in \mathbb{Z} \quad (3.18)$$

เนื่องจากปริภูมิ V_j และ W_j มีคุณสมบัติเป็นออร์โธนอร์มัลคอมพลีเมนต์ซึ่งกันและกัน ดังนั้นความสัมพันธ์ระหว่างฟังก์ชันมูลฐานในปริภูมิย่อย V_j และ W_j ใดๆคือ

$$\langle \varphi_{j,k}(t) | \psi_{m,n}(t) \rangle = 0, \quad j, k, m, n \in \mathbb{Z} \quad (3.19)$$

เช่นเดียวกับ Scaling Function ถ้าสมมติให้ $f(t) \in W_j$ แล้ว สามารถแสดงฟังก์ชัน $f(t)$ ด้วยอนุกรมของ Wavelet Function ได้คือ

$$f(t) = \sum_n d_j(n) \cdot \psi_{j,n}(t), \quad j, n \in \mathbb{Z} \quad (3.20)$$

เมื่อ $d_j(n)$ คือ Detail Expansion Coefficients จากสมการที่ (3.10) และ (3.20) ทำให้สามารถหาค่าของ $d_j(n)$ ได้จาก

$$d_j(n) = \langle f(t) | \psi_{j,n}(t) \rangle, \quad j, n \in \mathbb{Z} \quad (3.21)$$

เนื่องจาก $W_0 \subset V_1$ นั่นคือสามารถแสดง Wavelet Function $\psi(t)$ ในรูปของการรวมเชิงเส้นของ $\sqrt{2}\varphi(2t - n) \in V_1$ ได้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\psi(t) = \sum_n g(n) \cdot \sqrt{2} \varphi(2t - n), \quad n \in \mathbb{Z}, \quad g(n) \in l^2(\mathbb{Z}) \quad (3.22)$$

ด้วยคุณสมบัติของโธมอร์มัลในสมการที่ (3.10) ดังนั้น ค่าสัมประสิทธิ์ $g(n)$ ในสมการที่ (3.22) สามารถหาได้จากสมการ

$$g(n) = \langle \psi(t) | \sqrt{2} \varphi(2t - n) \rangle, \quad n \in \mathbb{Z} \quad (3.23)$$

3.3 อนุกรมเวฟเล็ต

ถ้าพิจารณาให้ $f(t) \in L^2(\mathbb{R})$ และกำหนดให้ปริภูมิ $L^2(\mathbb{R})$ ถูกแยก(Decompose)ออกเป็นปริภูมิย่อยถึงระดับ j แสดงโดยสมการ

$$L^2(\mathbb{R}) = V_j \oplus W_j \oplus W_{j+1} \oplus \dots \oplus W_{-1} \oplus W_0 \oplus W_1 \oplus \dots, \quad j \in \mathbb{Z} \quad (3.24)$$

นั่นคือสัญญาณ $f(t)$ จะประกอบไปด้วยฟังก์ชันมูลฐานของแต่ละปริภูมิย่อย ซึ่งสามารถแสดงด้วยสมการอนุกรมเวฟเล็ต(Wavelet Series Expansion)ได้คือ

$$f(t) = \sum_n c_j(n) \cdot \phi_{j,n}(t) + \sum_{m=j}^{\infty} \sum_n d_m(n) \cdot \psi_{m,n}(t) \quad (3.25)$$

เช่นเดียวกับสมการที่ (3.13) และ (3.21) Coarse และ Detail Expansion Coefficients สามารถหาได้จาก

$$c_j(n) = \langle f(t) | \phi_{j,n}(t) \rangle, \quad j, n \in \mathbb{Z} \quad (3.26)$$

$$d_m(n) = \langle f(t) | \psi_{m,n}(t) \rangle, \quad m, n \in \mathbb{Z} \quad (3.27)$$

3.4 การวิเคราะห์สัญญาณเชิงเวลาเต็มหน่วยด้วยทฤษฎีเวฟเล็ต

ในการคำนวณทางปฏิบัติ สัญญาณจะถูกโปรเจกชัน(Projection)ลงบนปริภูมิย่อยที่มีจำนวนจำกัด ซึ่งสามารถแสดงได้ในสมการ

$$V_0 \subset V_1 \subset V_2 \subset \dots \subset V_J \subset V_{J+1} \subset V_{J+2} \subset \dots \subset V_{J-2} \subset V_{J-1} \subset V_J \in L^2(\mathbb{R}), \quad j, J \in \mathbb{Z} \quad (3.28)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อพิจารณาให้ปริภูมิย่อย V_j ใดๆมีจำนวนฟังก์ชันมูลฐาน 2^j ฟังก์ชัน ถ้าพิจารณาให้ $f(t) \in V_j$ และกำหนดให้ปริภูมิ V_j ถูกแยกออกเป็นปริภูมิย่อยถึงระดับ j แสดงโดยสมการ

$$V_j = V_j \oplus W_j \oplus W_{j+1} \oplus W_{j+2} \oplus \dots \oplus W_{j-3} \oplus W_{j-2} \oplus W_{j-1}, \quad j < J, \quad j, J \in \mathbb{Z} \quad (3.29)$$

นั่นคือสัญญาณ $f(t)$ จะประกอบไปด้วยฟังก์ชันมูลฐานของแต่ละปริภูมิย่อย ซึ่งสามารถแสดงสมการอนุกรมเวฟเลตได้ใหม่คือ

$$f(t) = \sum_{n=0}^{2^j-1} c_j(n) \cdot \varphi_{j,n}(t) = \sum_{k=0}^{2^j-1} c_j(k) \cdot \varphi_{j,k}(t) + \sum_{m=j}^{J-1} \sum_{k=0}^{2^m-1} d_m(k) \cdot \psi_{m,k}(t) \quad (3.30)$$

การนำทฤษฎีเวฟเลตมาใช้ในการวิเคราะห์สัญญาณเชิงเวลาเต็มหน่วยสามารถทำได้โดยการกำหนดให้ Coarse Expansion Coefficients $c_j(n)$ คือสัญญาณเชิงเวลาเต็มหน่วยที่ต้องการวิเคราะห์หรือประมวลผล ถ้ากำหนดให้ $x(n)$ เป็นสัญญาณเชิงเวลาเต็มหน่วยที่ได้จากการดิจิไทซ์ (Digitized) สัญญาณเชิงเวลาต่อเนื่อง (Continuos-Time Signal) $x(t)$ การวิเคราะห์สัญญาณเชิงเต็มหน่วย $x(n)$ ด้วยทฤษฎีของเวฟเลตสามารถทำได้โดยการกำหนด

$$c_j(n) = x(n), \quad n = 0, 1, 2, \dots, 2^j - 1 \quad (3.31)$$

นั่นคือการวิเคราะห์สัญญาณเชิงเวลาเต็มหน่วยด้วยทฤษฎีเวฟเลตเป็นการคำนวณค่าของกลุ่มข้อมูลย่อย $d_{j-1}(n), d_{j-2}(n), \dots, d_j(n)$ และ $c_j(n)$ จากกลุ่มข้อมูล $c_j(n)$ เพื่อใช้เป็นข้อมูลเพื่อการประมวลผลในขั้นตอนการอื่นๆต่อไปนั่นเอง ซึ่งเรียกวิธีการคำนวณนี้ว่าการแปลงเวฟเลตเต็มหน่วย (Discrete Wavelet Transform : DWT)

3.5 การแปลงเวฟเลตเต็มหน่วย

พิจารณาสัญญาณ $f(t) \in V_{j+1}$ ที่สามารถแสดงด้วยอนุกรมของ Scaling Function ในปริภูมิย่อย V_{j+1} ได้คือ

$$f(t) = \sum_n c_{j+1}(n) \cdot \varphi_{j+1,n}(t) \in V_{j+1}, \quad j, n \in \mathbb{Z} \quad (3.32)$$

จาก $V_{j+1} = V_j \oplus W_j$ ดังนั้นสัญญาณ $f(t) \in V_{j+1}$ สามารถแสดงด้วยอนุกรมของ Scaling และ

Wavelet Function ในปริภูมิย่อย V_j และ W_j ได้โดยสมการ

ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$f(t) = \sum_k c_j(k) \cdot \varphi_{j,k}(t) + \sum_k d_j(k) \cdot \psi_{j,k}(t), \quad j, k \in \mathbb{Z} \quad (3.33)$$

พิจารณา Scaling Function จากสมการที่ (3.8) และ (3.14) สามารถแสดงได้ว่า

$$\begin{aligned} \varphi_{j,k}(t) &= 2^{j/2} \cdot \varphi(2^j t - k) \\ &= 2^{j/2} \sum_v h(v) \cdot \sqrt{2} \varphi(2^{j+1} t - 2k - v), \quad j, k, v \in \mathbb{Z} \end{aligned} \quad (3.34)$$

แทนค่า $2k + v$ ด้วย n จะได้ว่า

$$\begin{aligned} \varphi_{j,k}(t) &= \sum_n h(n - 2k) \cdot 2^{(j+1)/2} \varphi(2^{j+1} t - n) \\ &= \sum_n h(n - 2k) \cdot \varphi_{j+1,n}(t) \end{aligned} \quad (3.35)$$

เช่นเดียวกันสำหรับ Wavelet Function

$$\psi_{j,k}(t) = \sum_n g(n - 2k) \cdot \varphi_{j+1,n}(t) \quad (3.36)$$

จากสมการที่ (3.13) และ (3.35) สามารถคำนวณ Coarse Expansion Coefficients ได้โดยสมการ

$$\begin{aligned} c_j(k) &= \langle f(t) | \varphi_{j,k}(t) \rangle \\ &= \sum_n h(n - 2k) \cdot \langle f(t) | \varphi_{j+1,n}(t) \rangle \\ &= \sum_n h(n - 2k) \cdot c_{j+1}(n) \end{aligned} \quad (3.37)$$

เช่นเดียวกันสำหรับ Detail Expansion Coefficients

$$d_j(k) = \sum_n g(n - 2k) \cdot c_{j+1}(n) \quad (3.38)$$

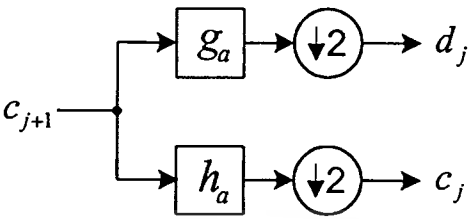
จากสมการที่ (3.37) และ (3.38) สามารถแสดงให้อยู่ในรูปของการคอนโวลูชัน(Convolution)กับผลตอบสนองอิมพัลส์ได้เป็น

$$c_j(k) = h_a(n) * c_{j+1}(n) \Big|_{n=2k} \quad (3.39)$$

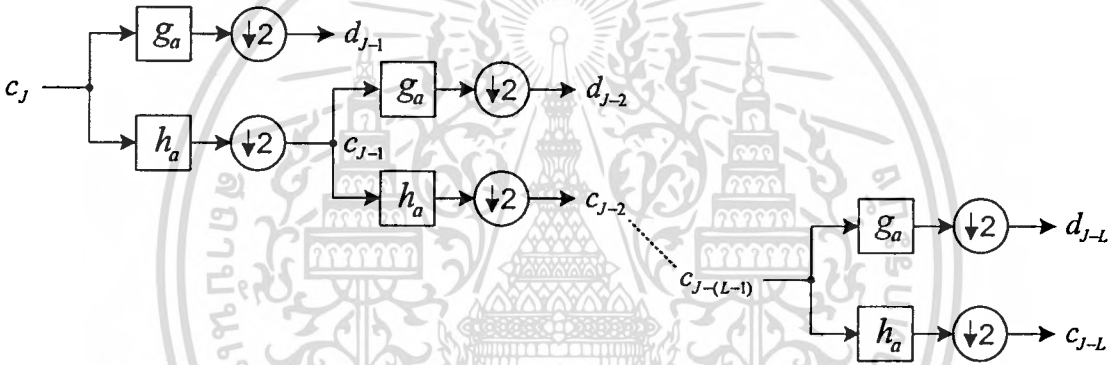
$$d_j(k) = g_a(n) * c_{j+1}(n) \Big|_{n=2k} \quad (3.40)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานภายในองค์กรโดยไม่อนุญาตให้นำไปใช้ประโยชน์ในการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยผลตอบสนองอิมพัลส์ $h_a(n)$ และ $g_a(n)$ คือ ตัวกรองวิเคราะห์(Analysis Filter) สมการที่ (3.39) และ (3.40) สามารถแสดงด้วย Two-Channel Analysis Filter Bank ในรูปที่ 3.1 และการแปลงเวฟเล็ตเต็มหน่วย L ระดับในรูปของ Octave Band Analysis Filter Bank แสดงในรูปที่ 3.2



รูปที่ 3.1 แสดง Two-Channel Analysis Filter Bank ที่ใช้ในการคำนวณ $c_j(n)$ และ $d_j(n)$ จาก $c_{j+1}(n)$



รูปที่ 3.2 แสดงการแปลงเวฟเล็ตเต็มหน่วย L ระดับในรูปของ Octave Band Analysis Filter Bank

3.6 การแปลงกลับเวฟเล็ตเต็มหน่วย

กลุ่มข้อมูลย่อยซึ่งประกอบด้วย $d_{j-1}(n), d_{j-2}(n), \dots, d_{j-L}(n)$ และ $c_{j-L}(n)$ ที่ได้จากการแปลงเวฟเล็ตเต็มหน่วยของสัญญาณ สามารถที่จะสร้างกลับคืนได้อย่างสมบูรณ์(Perfect Reconstruction : PR)โดยการแปลงกลับเวฟเล็ตเต็มหน่วย(Inverse Discrete Wavelet Transform : IDWT) จาก $V_1 = V_0 \oplus W_0$ สามารถแสดงได้ว่า

$$\sqrt{2}\varphi(2t) = \sum_v h_s(v) \cdot \varphi(t-v) + \sum_v g_s(v) \cdot \psi(t-v), \quad v \in Z \tag{3.41}$$

เมื่อผลตอบสนองอิมพัลส์ $h_s(n)$ และ $g_s(n)$ ก็คือ ตัวกรองสังเคราะห์(Synthesis Filter) จากสมการที่ (3.8) และ (3.41) สามารถแสดงได้ว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{aligned}\varphi_{j+1,n}(t) &= 2^{(j+1)/2} \cdot \varphi(2^{j+1}t - n) \\ &= \sum_v h_s(v) \cdot 2^{j/2} \varphi(2^j t - n/2 - v) + \sum_v g_s(v) \cdot 2^{j/2} \psi(2^j t - n/2 - v)\end{aligned}\quad (3.42)$$

แทนผลตอบสนองอิมพัลส์ $h_s(v)$ และ $g_s(v)$ ด้วย $h_s(-2v)$ และ $g_s(-2v)$ ตามลำดับ และแทนค่า $n+2v$ ด้วย $2k$ จะได้

$$\begin{aligned}\varphi_{j+1,n} &= \sum_k h_s(n-2k) \cdot 2^{j/2} \varphi(2^j t - k) + \sum_k g_s(n-2k) \cdot 2^{j/2} \psi(2^j t - k) \\ &= \sum_k h_s(n-2k) \cdot \varphi_{j,k}(t) + \sum_k g_s(n-2k) \cdot \psi_{j,k}(t)\end{aligned}\quad (3.43)$$

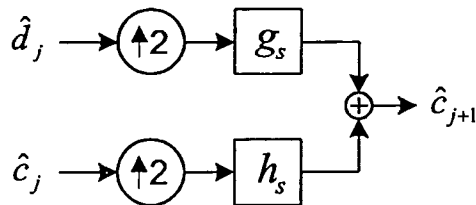
จากสมการที่ (3.13) และ (3.43) สามารถแสดงการคำนวณค่า $c_{j+1}(n)$ จาก $c_j(n)$ และ $d_j(n)$ ได้จากสมการ

$$\begin{aligned}c_{j+1}(n) &= \langle f(t) | \varphi_{j+1,n}(t) \rangle \\ &= \sum_k h_s(n-2k) \cdot \langle f(t) | \varphi_{j,k}(t) \rangle + \sum_k g_s(n-2k) \cdot \langle f(t) | \psi_{j,k}(t) \rangle \\ &= \sum_k h_s(n-2k) \cdot c_j(k) + \sum_k g_s(n-2k) \cdot d_j(k)\end{aligned}\quad (3.44)$$

ถ้าพิจารณาให้ $\hat{d}_j(n)$ และ $\hat{c}_j(n)$ เป็นข้อมูลของ $c_j(n)$ และ $d_j(n)$ ที่ผ่านการประมวลผลใน Transform Domain สมการที่ (3.44) สามารถแสดงใหม่ได้เป็น

$$\hat{c}_{j+1}(n) = \sum_k h_s(n-2k) \cdot \hat{c}_j(k) + \sum_k g_s(n-2k) \cdot \hat{d}_j(k)\quad (3.45)$$

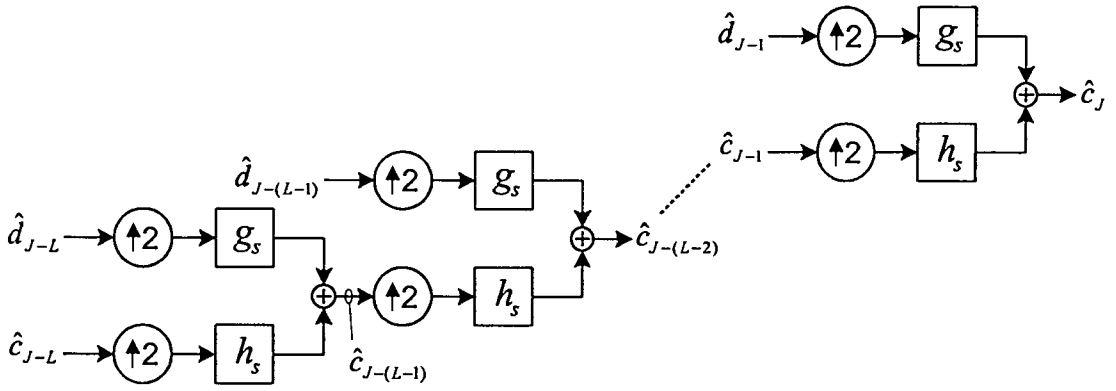
ซึ่งสามารถแสดงด้วย Two-Channel Synthesis Filter Bank ในรูปที่ 3.3 และการแปลงกลับเวฟเลตเต็มหน่วย L ระดับในรูปของ Octave Band Synthesis Filter Bank ได้ในรูปที่ 3.4



รูปที่ 3.3 แสดง Two-Channel Synthesis Filter Bank ที่ใช้ในการคำนวณ $\hat{c}_{j+1}(n)$ จาก $\hat{c}_j(n)$

และ $\hat{d}_j(n)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แสดงการแปลงกลับเวฟเลตเต็มหน่วย L ระดับในรูปของ Octave Band Synthesis

Filter Bank

3.7 ตระกูลของออโธโนมัลเวฟเลต

ตระกูลของออโธโนมัลเวฟเลต(Orthonormal Wavelet)ที่สำคัญได้แก่ Daubechies, Symmlet และ Coiflet เป็นต้น แต่ละตระกูลจะมีฟังก์ชันมูลฐาน(Scaling และ Wavelet Basis Function)ที่มีรูปร่างลักษณะที่แตกต่างกันไป เช่น ตระกูลของ Daubechies จะมีฟังก์ชันมูลฐานในลักษณะของ Asymmetric ตระกูลของ Symmlet จะมีฟังก์ชันมูลฐานในลักษณะของ Least Asymmetric และตระกูลของ Coiflet จะมีฟังก์ชันมูลฐานในลักษณะของ Nearly Symmetric เป็นต้น นอกจากนี้แต่ละตระกูลยังสามารถแบ่งได้ตามความราบเรียบ(Smooth)ของฟังก์ชันมูลฐาน ซึ่งกำหนดโดยค่าของ Number of Vanishing Moments (NVM) ท้ายชื่อตระกูล (Daubechies 4,6,8,...,20 , Symmlet 4,5,6,...,10 และ Coiflet 1,2,...,5) ลักษณะของฟังก์ชันมูลฐานที่เลือกใช้งาน จะมีความราบเรียบมากขึ้นเมื่อค่า NVM มากขึ้น ตัวอย่างฟังก์ชันมูลฐานของออโธโนมัลเวฟเลต ทั้ง Scaling Basis Function (เรียกอีกอย่างว่า Father Wavelet Basis Function) และ Wavelet Basis Function (เรียกอีกอย่างว่า Mother Wavelet Basis Function) แสดงในรูปที่ 3.5 ถึงรูปที่ 3.7

ในการเลือกชนิดของเวฟเลตสามารถทำได้โดยการกำหนดค่าผลตอบสนองอิมพัลส์ (Impulse Response) $h_s(n)$ ที่จะนำไปใช้กับสมการที่ (3.45) ค่าผลตอบสนองอิมพัลส์ $h_s(n)$ ของเวฟเลตแต่ละชนิดแสดงในตารางที่ 1 ถึงตารางที่ 5 [8] สำหรับผลตอบสนองอิมพัลส์ $g_s(n)$, $h_a(n)$ และ $g_a(n)$ จากสมการที่ (3.45) สมการที่ (3.39) และสมการที่ (3.40) สามารถหาได้จากสมการที่ (3.46) ถึงสมการที่ (3.48)

$$g_s(n) = (-1)^n h_s(1-n), \quad n \in \mathbb{Z} \quad (3.46)$$

$$h_a(n) = h_s(-n), \quad n \in \mathbb{Z} \quad (3.47)$$

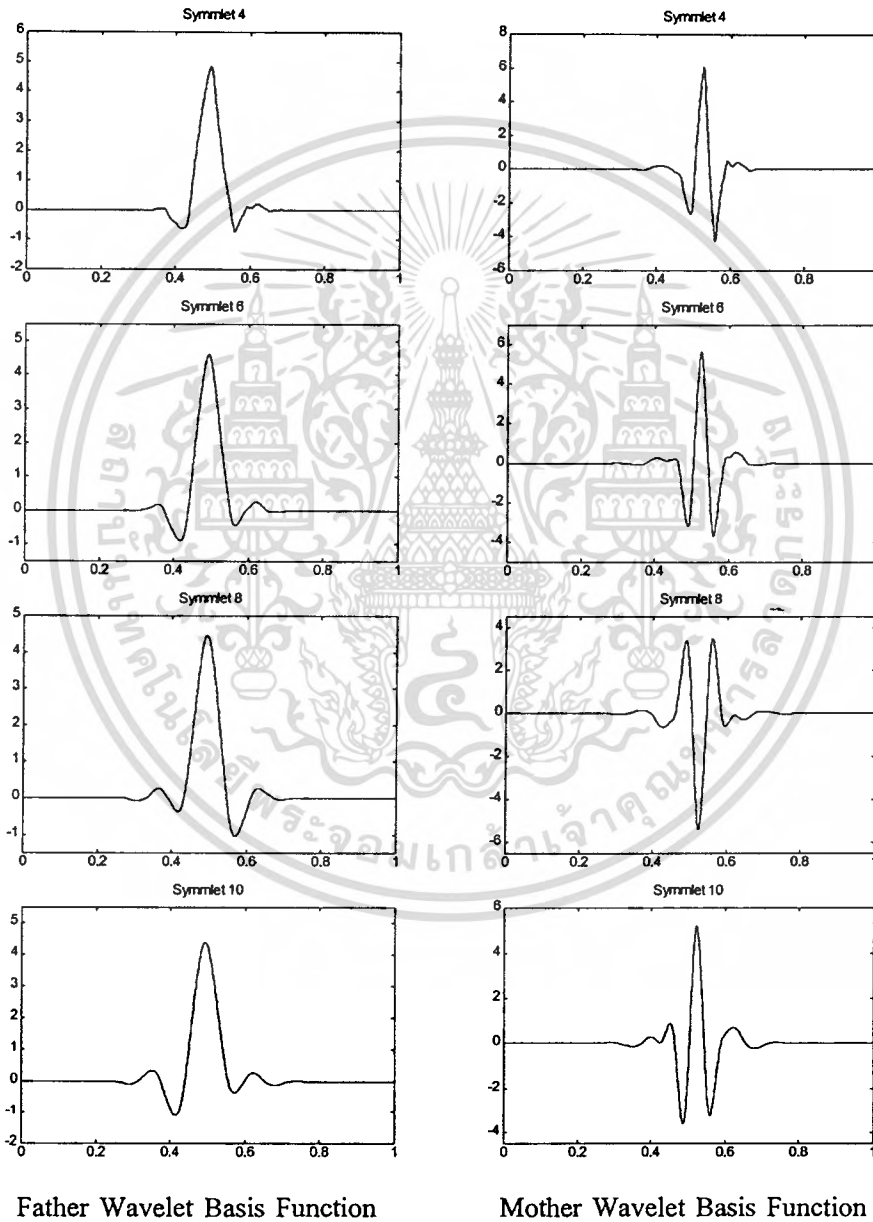
$$g_a(n) = g_s(-n), \quad n \in \mathbb{Z} \quad (3.48)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

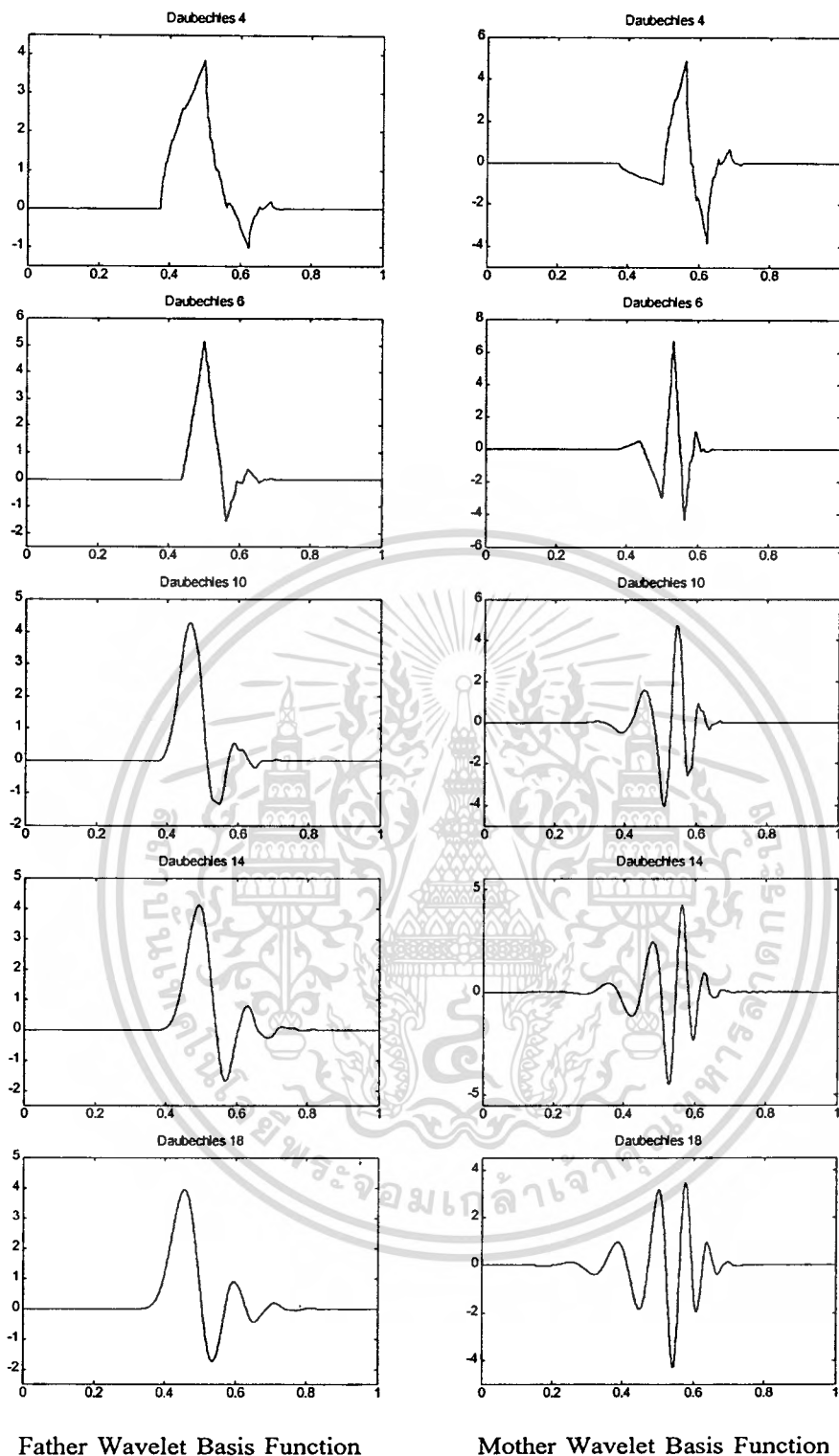
ผลตอบสนองอิมพัลส์ทั้งหมดสอดคล้องกับคุณสมบัติของออโรนอร์มัลเวฟเล็ต ซึ่งแสดงโดยสมการที่ (3.49) และ (3.50)

$$\langle h_s(n-2k) | h_s(n-2l) \rangle = \langle g_s(n-2k) | g_s(n-2l) \rangle = \delta_{kl}, \quad k, l, n \in \mathbb{Z} \quad (3.49)$$

$$\langle h_s(n-2k) | g_s(n-2l) \rangle = \langle h_a(n-2k) | g_a(n-2l) \rangle = 0, \quad k, l, n \in \mathbb{Z} \quad (3.50)$$



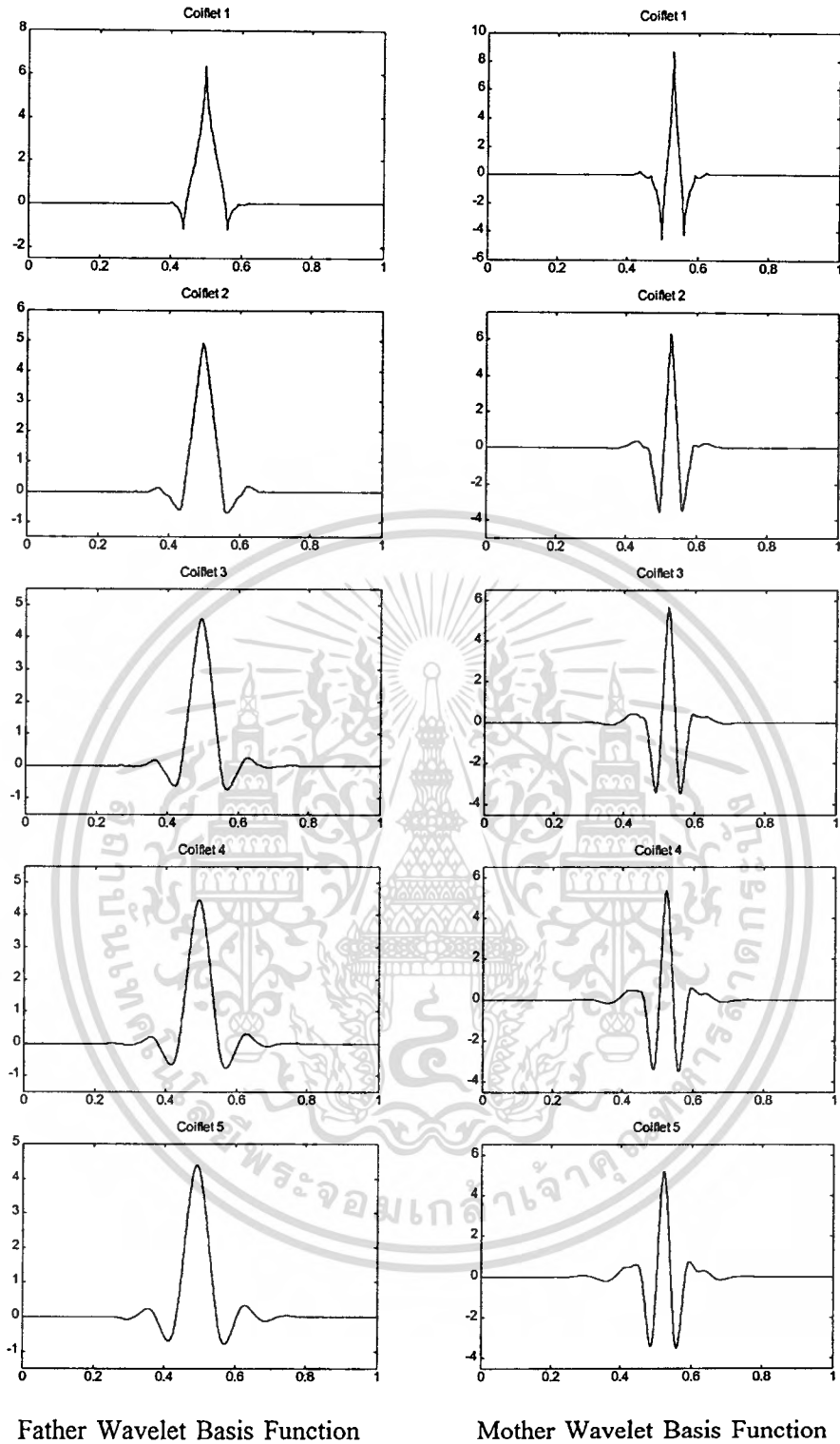
รูปที่ 3.5 แสดงฟังก์ชันมูลฐานเวฟเล็ตตระกูล Symmetlet เมื่อ NVM มีค่าเป็น 4, 6, 8 และ 10



Father Wavelet Basis Function

Mother Wavelet Basis Function

รูปที่ 3.6 แสดงฟังก์ชันมูลฐานเวฟเลตตระกูล Daubechies เมื่อ NVM มีค่าเป็น 4, 6, 10, 14 และ 18



รูปที่ 3.7 แสดงฟังก์ชันมูลฐานเวฟเลตตระกูล Coiflet เมื่อ NVM มีค่าจาก 1 ถึง 5

ตารางที่ 3.1 แสดงผลตอบสนองอิมพัลส์ $h_s(n)$ ของเวฟเลตตระกูล Daubechies เมื่อ NVM มีค่าเป็น 4, 6, 8, 10 และ 12

n	$h_s^{D^4}(n)$	$h_s^{D^6}(n)$	$h_s^{D^8}(n)$	$h_s^{D^{10}}(n)$	$h_s^{D^{12}}(n)$
0	0.482962913145	0.332670552950	0.230377813309	0.160102397974	0.111540743350
1	0.836516303738	0.806891509311	0.714846570553	0.603829269797	0.494623890398
2	0.224143868042	0.459877502118	0.630880767940	0.724308528438	0.751133908021
3	-0.129409522551	-0.135011020010	-0.027983769417	0.138428145901	0.315250351709
4		-0.085441273882	-0.187034811719	-0.242294887066	-0.226264693965
5		0.035226291886	0.030841381836	-0.032244869585	-0.129766867567
6			0.032883011667	0.077571493840	0.097501605587
7			-0.010597401785	-0.006241490213	0.027522865530
8				-0.012580751999	-0.031582039317
9				0.003335725285	0.000553842201
10					0.004777257511
11					-0.001077301085

ตารางที่ 3.2 แสดงผลตอบสนองอิมพัลส์ $h_s(n)$ ของเวฟเลตตระกูล Daubechies เมื่อ NVM มีค่าเป็น 14, 16, 18, และ 20

n	$h_s^{D14}(n)$	$h_s^{D16}(n)$	$h_s^{D18}(n)$	$h_s^{D20}(n)$
0	0.077852054085	0.054415842243	0.038077947364	0.026670057901
1	0.396539319482	0.312871590914	0.243834674613	0.188176800078
2	0.729132090846	0.675630736297	0.604823123690	0.527201188932
3	0.469782287405	0.585354683654	0.657288078051	0.688459039453
4	-0.143906003929	-0.015829105256	0.133197385825	0.281172343661
5	-0.224036184994	-0.284015542962	-0.293273783279	-0.249846424327
6	0.071309219267	0.000472484574	-0.096840783223	-0.195946274377
7	0.080612609151	0.128747426620	0.148540749338	0.127369340336
8	-0.038029936935	-0.017369301002	0.030725681479	0.093057364604
9	-0.016574541631	-0.044088253931	-0.067632829061	-0.071394147166
10	0.012550998556	0.013981027917	0.000250947115	-0.029457536822
11	0.000429577973	-0.008746094047	0.022361662124	0.033212674059
12	-0.001801640704	-0.004870352993	-0.004723204758	0.003606553567
13	0.000353713800	-0.000391740373	-0.004281503682	-0.010733175483
14		0.000675449406	0.001847646883	0.001395351747
15		-0.000117476784	0.000230385764	0.001992405295
16			-0.000251963189	-0.000685856695
17			0.000039347320	-0.000116466855
18				0.000093588670
19				-0.000013264203

ตารางที่ 3.3 แสดงผลตอบสนองของอิมพัลส์ $h_i(n)$ คูณด้วย $\sqrt{2}$ ของเวฟเลตตระกูล Symmlet เมื่อ NVM มีค่าจาก 4 ถึง 7

n	$\sqrt{2} \cdot h_i^{S4}(n)$	$\sqrt{2} \cdot h_i^{S5}(n)$	$\sqrt{2} \cdot h_i^{S6}(n)$	$\sqrt{2} \cdot h_i^{S7}(n)$
0	-0.107148901418	0.038654795955	0.021784700327	0.003792658534
1	-0.041910965125	0.041746864422	0.004936612372	-0.001481225915
2	0.703739068656	-0.055344186117	-0.166863215412	-0.017870431651
3	1.136658243408	0.281990696854	-0.068323121587	0.043155452582
4	0.421234534204	1.023052966894	0.694457972958	0.096014767936
5	-0.140317624179	0.896581648380	1.113892783926	-0.070078291222
6	-0.017824701442	0.023478923136	0.477904371333	0.024665659489
7	0.045570345896	-0.247951362613	-0.102724969862	0.758162601964
8		-0.029842499869	-0.029783751299	1.085782709814
9		0.027632152958	0.063250562660	0.408183939725
10			0.002499922093	-0.198056706807
11			-0.011031867509	-0.152463871896
12				0.005671342686
13				0.014521394762

ตารางที่ 3.4 แสดงผลตอบสนองอิมพัลส์ $h_s(n)$ คูณด้วย $\sqrt{2}$ ของเวฟเลตตระกูล Symmlet เมื่อ NVM มีค่าจาก 8 ถึง 10

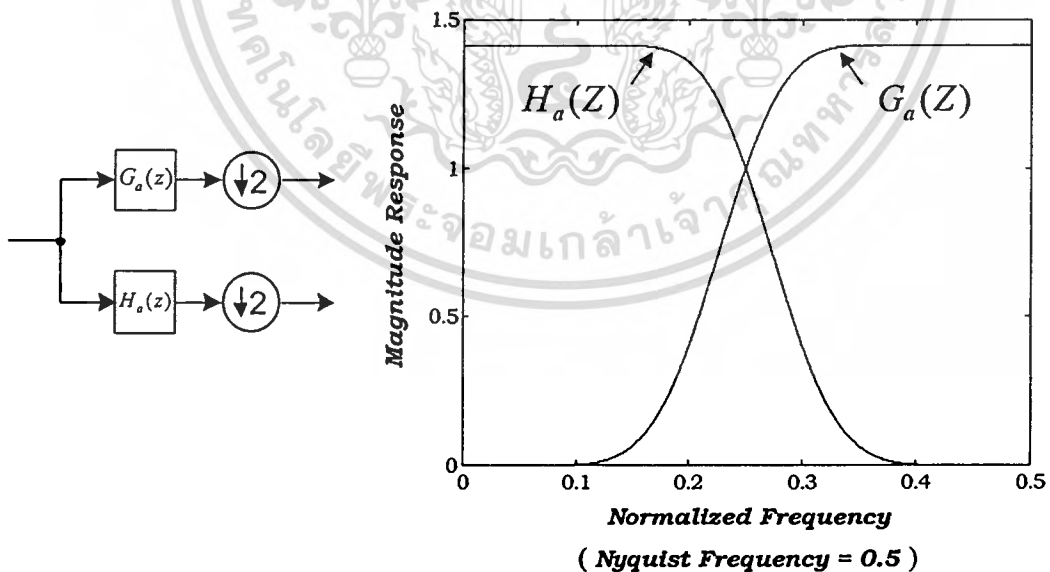
n	$\sqrt{2} \cdot h_s^{s8}(n)$	$\sqrt{2} \cdot h_s^{s9}(n)$	$\sqrt{2} \cdot h_s^{s10}(n)$
0	0.002672793393	0.001512487309	0.001089170447
1	-0.000428394300	-0.000669141509	0.000135245020
2	-0.021145686528	-0.014515578553	-0.012220642630
3	0.005386388754	0.012528896242	-0.002072363923
4	0.069490465911	0.087791251554	0.064950924579
5	-0.038493521263	-0.025786445930	0.016418869426
6	-0.073462508761	-0.270893783503	-0.225558972234
7	0.515398670374	0.049882830959	-0.100240215031
8	1.099106630537	0.873048407350	0.667071338154
9	0.680745347190	1.015259790833	1.088251530500
10	-0.086653615406	0.337658923602	0.542813011213
11	-0.202648655286	-0.077172161097	-0.050256540092
12	0.010758611751	0.000825140929	-0.045240772218
13	0.044823623042	0.042744433602	0.070703567550
14	-0.000766690896	-0.016303351226	0.008152816799
15	-0.004783458512	-0.018769396836	-0.028786231926
16		0.000876502539	-0.001137535314
17		0.001981193736	0.006495728375
18			0.000080661204
19			-0.000649589896

ตารางที่ 3.5 แสดงผลตอบสนองอิมพัลส์ $h_s(n)$ ทหารด้วย $\sqrt{2}$ ของเวฟเลตตระกูล Coiflet เมื่อ NVM มีค่าจาก 1 ถึง 5

n	$h_s^{C1}(n)/\sqrt{2}$	$h_s^{C2}(n)/\sqrt{2}$	$h_s^{C3}(n)/\sqrt{2}$	$h_s^{C4}(n)/\sqrt{2}$	$h_s^{C5}(n)/\sqrt{2}$
-10					-0.000149963816
-9					0.000253561192
-8				0.000630961046	0.001540245662
-7				-0.001152224852	-0.002941110800
-6			-0.002682418671	-0.005194524026	-0.007163781912
-5			0.005503126708	0.011362459243	0.016552066381
-4		0.011587596739	0.016583560479	0.018867235378	0.019917804362
-3		-0.029320137980	-0.046507764478	-0.057464234428	-0.064997262786
-2	-0.051429728471	-0.047639590310	-0.043220763560	-0.039652648517	-0.036800073624
-1	0.238929728471	0.273021046535	0.286503335274	0.293667390895	0.298092323496
0	0.602859456942	0.574682393857	0.561285256870	0.553126452562	0.547505429404
1	0.272140543058	0.294867193696	0.302983571773	0.307157326198	0.309706849007
2	-0.051429972847	-0.054085607091	-0.050770140755	-0.047112738865	-0.043866050755
3	-0.011070271529	-0.042026480460	-0.058196250761	-0.068038127051	-0.074652238894
4		0.016744410163	0.024434094321	0.027813640153	0.029195879490
5		0.003967883613	0.011229240962	0.017735837438	0.023110777009
6		-0.001289203356	-0.006369601011	-0.010756318516	-0.013973687948
7		-0.000509505399	-0.001820458915	-0.004001012886	-0.006480089993
8			0.000790205101	0.002652665946	0.004783001379
9			0.000329665174	0.000895594529	0.001720654697
10			-0.000050192775	-0.000416500572	-0.001175822247
11			-0.000024465734	-0.000183829769	-0.000451226967
12				0.000044080354	0.000213729756
13				0.000022082857	0.000099377599
14				-0.000002304942	-0.000029232137
15				-0.000001262175	-0.000015071991
16					0.000002640759
17					0.000001459331
18					-0.000000118375
19					-0.000000067287

3.8 ผลตอบสนองความถี่ของ Octave Band Wavelet Filter Bank

การนำทฤษฎีเวฟเล็ตมาใช้ในการวิเคราะห์สัญญาณเต็มหน่วยก็คือการพิจารณาให้สัญญาณเต็มหน่วยที่ต้องการวิเคราะห์นั้นคือกลุ่มสัมประสิทธิ์ของ Scaling Function ที่ระดับความละเอียดใดๆแล้วทำการแยกกลุ่มสัมประสิทธิ์นั้นๆออกเป็นกลุ่มสัมประสิทธิ์ของ Wavelet Function ที่ระดับความละเอียดต่างๆและกลุ่มสัมประสิทธิ์ของ Scaling Function ที่มีความละเอียดน้อยลง ขบวนการดังกล่าวสามารถแสดงให้อยู่ในรูปของตัวกรองเชิงเลข(Digital Filter)แบบ Octave Band Analysis Filter Bank ในรูปที่ 3.2 เมื่อผลตอบสนองอิมพัลส์ $h_a(n)$ และ $g_a(n)$ แสดงในรูปของ Z-Transform ได้ด้วย $H_a(z)$ และ $G_a(z)$ ตามลำดับ จากรูปที่ 3.8 แสดงตัวอย่างของ Two-Channel Analysis Filter Bank ของตัวกรองแบบ Daubechies-20 เมื่อวิเคราะห์ผลตอบสนองความถี่ $H_a(z)$ จะมีลักษณะเป็นตัวกรองผ่านความถี่ต่ำ(Low Pass Filter) และ $G_a(z)$ จะมีลักษณะเป็นตัวกรองผ่านความถี่สูง(High Pass Filter) โดยตัวกรองทั้งสองจะมีจุดตัดความถี่(Cut-Off Frequency)ที่ตำแหน่งเดียวกันคือ 0.25 (เมื่อ Nyquist Frequency คือ 0.5) สำหรับผลตอบสนองอิมพัลส์ของออโรนอร์มัลเวฟเล็ตชนิดอื่นๆก็จะมีลักษณะคล้ายคลึงกัน จะต่างกันเพียงความกว้างของช่วง Transition Band ซึ่งจะขึ้นอยู่กับอันดับหรือจำนวน Tap Delay ของตัวกรอง FIR ตัวกรองที่มีอันดับหรือจำนวน Tap Delay สูงกว่าจะมีช่วง Transition Band ที่แคบกว่าหรือให้ลักษณะการกรองความถี่ที่คมกว่า



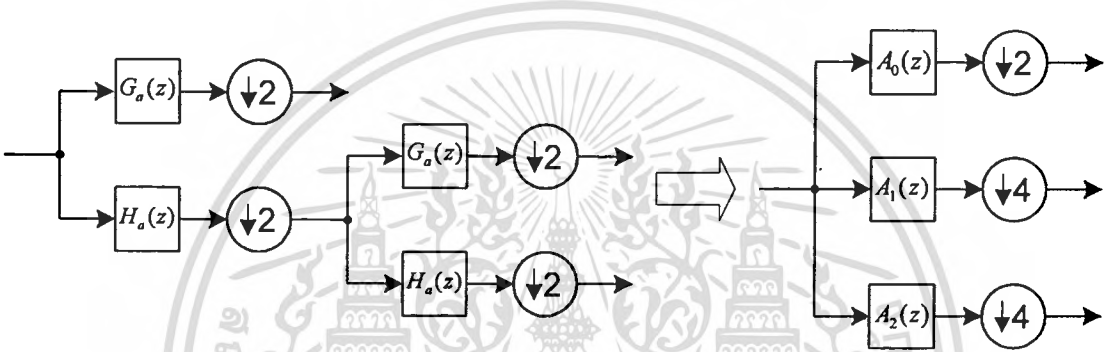
รูปที่ 3.8 แสดงผลตอบสนองความถี่ของตัวกรองที่มีผลตอบสนองอิมพัลส์ $h_a(n)$ และ $g_a(n)$

แบบ Daubechies-20

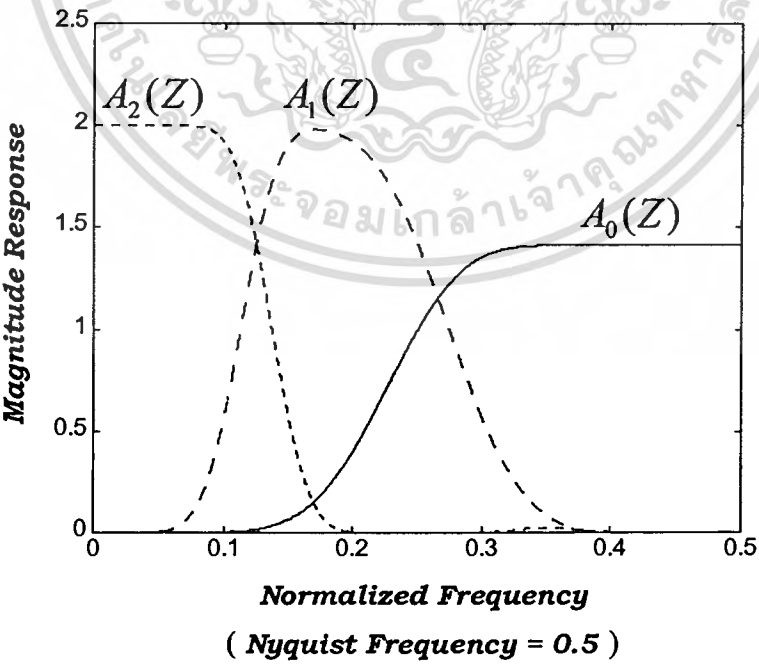
เมื่อพิจารณารูปแบบสมมูลของ Octave Band Analysis Filter Bank 2 ระดับในรูปแบบที่ 3.9 ซึ่งมีผลตอบสนองอิมพัลส์ใหม่ในสมการที่ (3.51) จากผลตอบสนองความถี่ในรูปแบบที่ 3.10 จะเห็นว่า Octave Band Analysis Filter Bank 2 ระดับเป็นการแบ่งครึ่งช่วงความถี่ 0-0.25 จากระดับที่ได้ออกเป็น 2 ช่วงคือ 0-0.125 และ 0.125-0.25 เมื่อรวมกับช่วงความถี่เดิม 0.25-0.5 จึงมีทั้งหมด 3 ช่วงความถี่

$$\begin{aligned} A_0(z) &= G_a(z) \\ A_1(z) &= H_a(z) \cdot G_a(z^2) \\ A_2(z) &= H_a(z) \cdot H_a(z^2) \end{aligned}$$

(3.51)



รูปที่ 3.9 แสดงรูปแบบสมมูลของ Octave Band Analysis Filter Bank 2 ระดับ



รูปที่ 3.10 แสดงผลตอบสนองความถี่ของ Octave Band Analysis Filter Bank 2 ระดับ โดยใช้

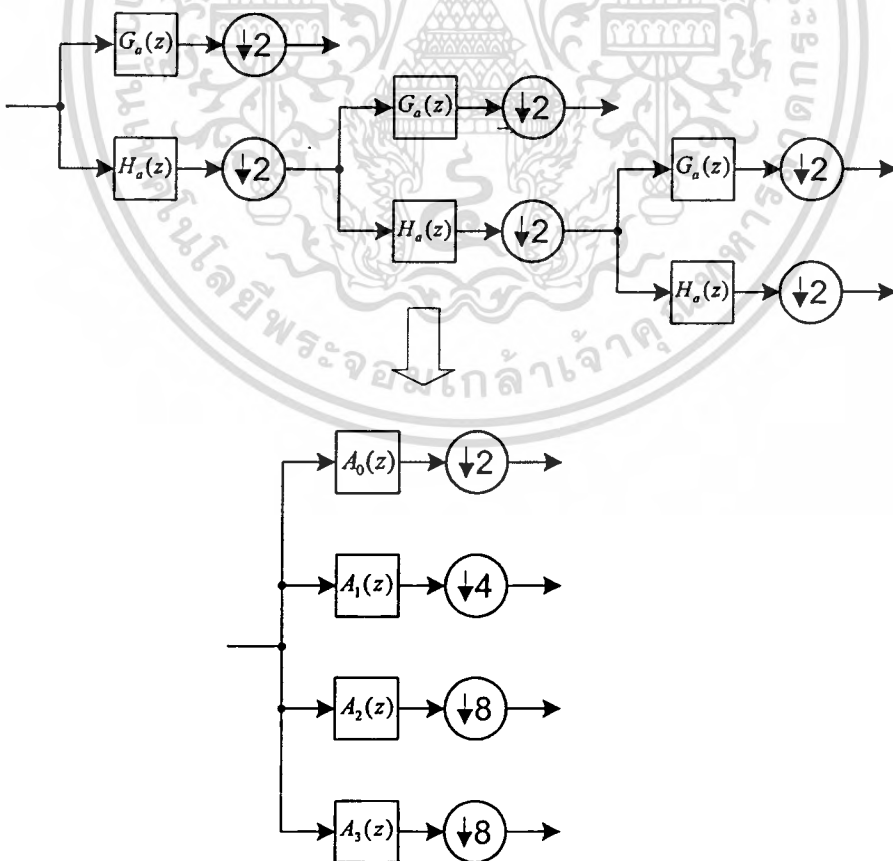
ตัวกรองแบบ Daubechies-20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่นเดียวกัน รูปแบบสมมูลของ Octave Band Analysis Filter Bank 3 ระดับแสดงในรูปที่ 3.11 ซึ่งมีผลตอบสนองอิมพัลส์ใหม่ในสมการที่ (3.52) จากผลตอบสนองความถี่ในรูปที่ 3.12 จะเห็นว่า Octave Band Analysis Filter Bank 3 ระดับ เป็นการแบ่งครึ่งช่วงความถี่ 0-0.125 จากระดับที่แล้วออกเป็น 2 ช่วงคือ 0-0.0625 และ 0.0625-0.125 เมื่อรวมกับช่วงความถี่เดิมคือ 0.125-0.25 และ 0.25-0.5 จึงมีทั้งหมด 4 ช่วงความถี่ สำหรับระดับต่อไปก็มีลักษณะเช่นเดียวกัน จะเห็นว่าการแปลงเวฟเลตเต็มหน่วยเป็นการแยกข้อมูลหรือสัญญาณที่ต้องการวิเคราะห์ออกเป็นหลายๆช่วงความถี่เพื่อทำการประมวลผลกลุ่มข้อมูลที่สัมพันธ์กับช่วงความถี่นั้นๆอย่างอิสระนั่นเอง

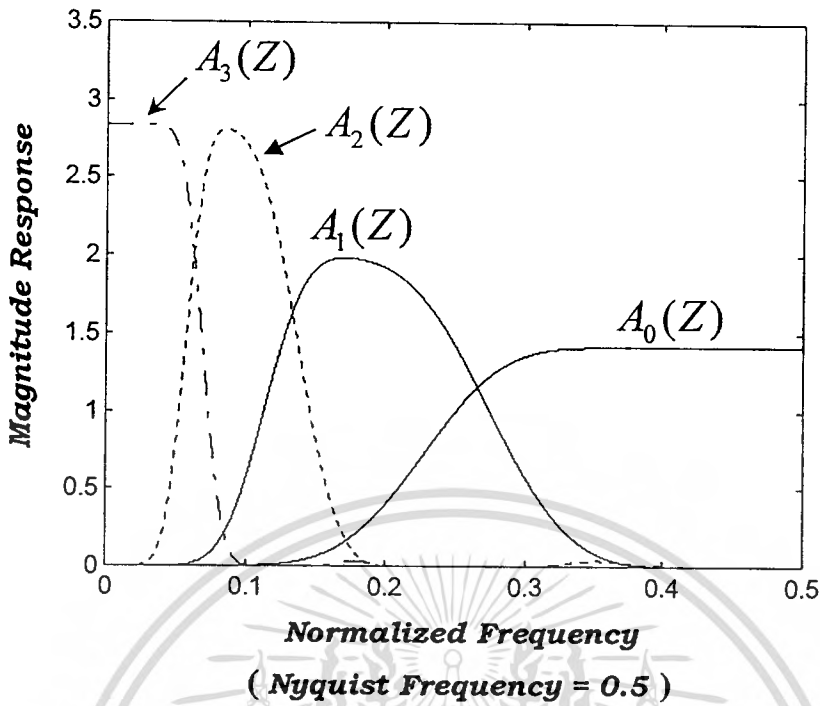
$$\begin{aligned}
 A_0(z) &= G_a(z) \\
 A_1(z) &= H_a(z) \cdot G_a(z^2) \\
 A_2(z) &= H_a(z) \cdot H_a(z^2) \cdot G_a(z^4) \\
 A_3(z) &= H_a(z) \cdot H_a(z^2) \cdot H_a(z^4)
 \end{aligned}
 \tag{3.52}$$

ตัวอย่างการหาผลตอบสนองความถี่ของ Octave Band Analysis Filter Bank สามารถแสดงด้วยการโปรแกรมในภาคผนวก ก



รูปที่ 3.11 แสดงแสดงรูปแบบสมมูลของ Octave Band Analysis Filter Bank 3 ระดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 แสดงผลตอบสนองความถี่ของ Octave Band Analysis Filter Bank 3 ระดับ โดยใช้อัตรากรองแบบ Daubechies-20

ทฤษฎีการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจ

ระบบประมวลผลสัญญาณไฟฟ้าหัวใจด้วยคอมพิวเตอร์นั้นจำเป็นต้องรับข้อมูลในปริมาณที่สูง ทำให้เกิดอุปสรรคในการบันทึกข้อมูลในอุปกรณ์บันทึกที่มีเนื้อที่จำกัด ส่งผลให้ระบบมีค่าใช้จ่ายสูงและมีประสิทธิภาพต่ำ วิธีการหนึ่งที่จะลดปัญหาดังกล่าวได้คือการลดขนาดข้อมูลก่อนการบันทึก โดยการลดขนาดข้อมูลนี้มีจุดมุ่งหมายที่จะลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจให้มากที่สุดโดยที่สาระสำคัญของข้อมูลทางการแพทย์ยังคงอยู่ในขอบเขตที่ยอมรับได้ ในบทนี้จะกล่าวถึงวิธีการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจแบบต่างๆ โดยสังเขป ตลอดจนมาตรฐานในการวัดประสิทธิภาพการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจ

4.1 วิธีการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจ

วิธีการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจที่แบ่งตามลักษณะของวิธีการมีสองกลุ่มใหญ่ๆคือ การลดขนาดข้อมูลด้วยวิธีทางตรง(Direct Method) และการลดขนาดข้อมูลด้วยวิธีการแปลง(Transformation Method)

4.1.1 การลดขนาดข้อมูลด้วยวิธีทางตรง

เป็นวิธีการลดขนาดข้อมูลที่มีการดำเนินการกับข้อมูลในโดเมนของเวลาโดยตรง ได้แก่ วิธี Turning Point (TP) วิธี Amplitude Zone Time Epoch Coding (AZTEC) วิธี Coordinate Reduction Time Encoding System (CORTES) และวิธี Fan เป็นต้น ทุกวิธีเป็นการลดขนาดข้อมูลโดยการเก็บข้อมูลที่มีสาระสำคัญมากไว้(Significant-Point-Extraction)[2] นอกจากนี้ยังมีวิธี Run-Length Coding (RLC) ซึ่งเป็นวิธีการลดจำนวนข้อมูลที่มีลักษณะของการซ้ำและต่อเนื่องลง และวิธีการเข้ารหัสเอนโทรปี(Entropy Coding)ซึ่งเป็นวิธีการลดขนาดข้อมูลโดยใช้ทฤษฎีการเข้ารหัสของชาวสาร เป็นต้น

4.1.1.1 วิธี Turning Point (TP)

เป็นวิธีการลดขนาดข้อมูลโดยใช้วิธีการลดความถี่ที่ต่ำลงทีละครั้ง โดยข้อมูลที่บันทึกจะเป็นข้อมูลที่ตำแหน่ง Peak และ Valley หรือเป็นตำแหน่งวกกลับ(Turning Point)เหมือนชื่อวิธีนั่นเอง มักจะใช้กับสัญญาณไฟฟ้าหัวใจที่เก็บมาด้วยความถี่ที่สูงกว่าแถบความถี่ของสัญญาณไฟฟ้าหัวใจมากๆ เช่น สัญญาณที่สุ่มมาด้วยความถี่ 400 Hz สามารถใช้วิธี TP ในการลดจำนวนข้อมูล

เหลือเพียง 200 Samples/Second ดังนั้นค่า Compression Ratio (CR) ของวิธี TP จึงมีค่าที่แน่นอนที่ 2, 4, 6,...

4.1.1.2 วิธี Amplitude Zone Time Epoch Coding (AZTEC)

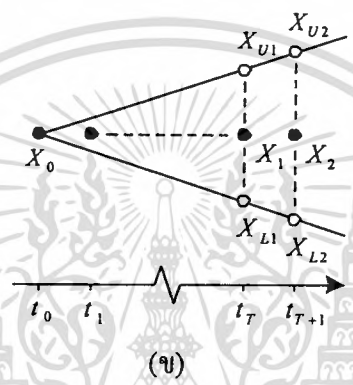
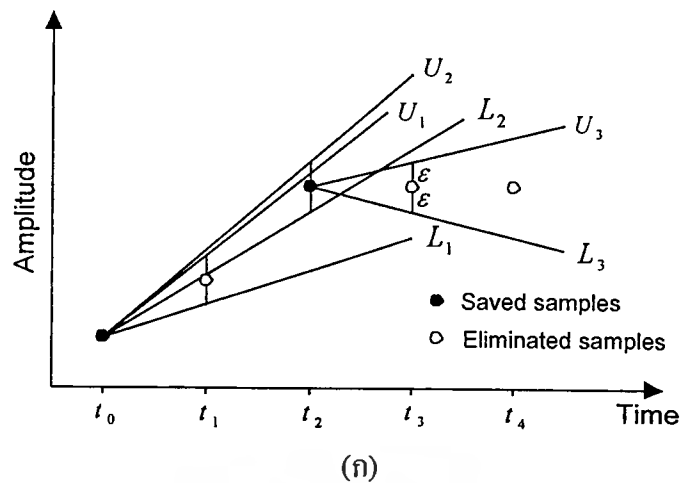
เป็นวิธีการลดขนาดข้อมูลโดยการเปลี่ยนลำดับของข้อมูลให้อยู่ในรูปของ Plateau และ Slope วิธีนี้ให้ค่า CR ที่ไม่แน่นอน ขึ้นอยู่กับลักษณะของข้อมูล สัญญาณไฟฟ้าหัวใจที่ผ่านการลดขนาดข้อมูลด้วยวิธี AZTEC จะมีลักษณะของ Steplike Quantization ซึ่งไม่สามารถยอมรับได้ในทางการแพทย์ โดยทั่วไปสัญญาณไฟฟ้าหัวใจที่ผ่านการลดขนาดข้อมูลด้วยวิธีนี้จะนำไปผ่าน Smoothing Filter เพื่อลดลักษณะของความไม่ต่อเนื่อง(Discontinuities)ลง อย่างไรก็ตามการใช้ Smoothing Filter จะทำให้เกิดความเพี้ยนของขนาด(Amplitude Distortion)ขึ้นมาก

4.1.1.3 วิธี Coordinate Reduction Time Encoding System (CORTES)

เป็นวิธีการลดขนาดข้อมูลโดยใช้วิธี TP และ AZTEC ร่วมกัน โดยจะใช้วิธี AZTEC ในการลดข้อมูลที่ไม่มีความสำคัญในส่วนของ Isoelectric Region ด้วยค่า CR ที่สูง และใช้วิธี TP ในส่วนของความถี่สูงหรือส่วนของ QRS Complex สัญญาณไฟฟ้าหัวใจที่ผ่านการลดขนาดข้อมูลด้วยวิธีนี้จะต้องนำมาผ่าน Parabolic Smoothing เพื่อลดความไม่ต่อเนื่องลง วิธี CORTES จะให้ประสิทธิภาพการลดขนาดข้อมูลที่สูงกว่าวิธี AZTEC

4.1.1.4 วิธี Fan

เป็นวิธีการลดขนาดข้อมูลที่ให้ประสิทธิภาพสูงที่สุดในกลุ่มของ Significant-Point-Extraction การลดขนาดข้อมูลด้วยวิธี Fan สามารถอธิบายได้ในรูปที่ 4.1 เริ่มจากกำหนดจุดเริ่มต้น (X_0) เป็นข้อมูลที่ต้องบันทึกเป็นจุดแรก จากนั้นจึงรับข้อมูลจุดที่สอง (X_1) และลากเส้นขอบเขต U_1 และ L_1 ในแนวของ $(X_0, X_1 + \varepsilon)$ และ $(X_0, X_1 - \varepsilon)$ ตามลำดับ โดยกำหนดให้ ε คือค่าความผิดพลาดที่ยอมรับได้ ถ้าข้อมูลจุดที่สาม (X_2) อยู่ภายในขอบเขตของเส้นตรง U_1 และ L_1 ให้ลากเส้นขอบเขตใหม่คือ U_2 และ L_2 ในแนวของ $(X_0, X_2 + \varepsilon)$ และ $(X_0, X_2 - \varepsilon)$ ตามลำดับต่อไปให้เปรียบเทียบเส้นขอบเขตทั้งสองชุด โดยให้เก็บเส้นขอบเขตบนและล่างที่ให้ระยะที่ใกล้กันที่สุดไว้เพื่อการพิจารณาข้อมูลจุดต่อไป(ในรูปที่ 4.1 เส้นขอบเขตที่ดังกล่าวคือ U_1 และ L_2) ต่อไปกำหนดให้ X_1 เท่ากับ X_2 และรับข้อมูลจุดต่อไปเข้ามาใหม่แทน X_2 จากนั้นจึงย้อนขบวนการโดยเปรียบเทียบ X_2 กับค่าที่คำนวณได้จากเส้นขอบเขตบนและล่างที่ให้ระยะที่ใกล้กันที่สุดจากที่ผ่านมา ซึ่งคำนวณได้จากสมการที่ (4.1) และ (4.2) ถ้าค่าดังกล่าวอยู่นอกขอบเขตให้ทำการบันทึกระยะระหว่างข้อมูล T และค่าสุดท้าย(X_1) ค่าสุดท้ายนี้จะถูกกำหนดให้เป็นจุดเริ่มต้น(X_0) เพื่อย้อนขบวนการเดิมต่อไป



รูปที่ 4.1 แสดงการลดขนาดข้อมูลด้วยวิธี Fan (ก) การลากเส้นขอบเขตเพื่อพิจารณาการเก็บและการละข้อมูล (ข) การหาตำแหน่งของ X_{U2} และ X_{L2} จาก X_{U1} , X_{L1} และ X_0

$$X_{U2} = \frac{X_{U1} - X_0}{T} + X_{U1} \tag{4.1}$$

$$X_{L2} = \frac{X_{L1} - X_0}{T} + X_{L1} \tag{4.2}$$

เมื่อ $T = t_T - t_0$

4.1.1.5 วิธี Run-Length Coding (RLC)

วิธี RLC เป็นวิธีการลดขนาดข้อมูลพื้นฐานอย่างง่ายที่สุด โดยเป็นการลดความซ้ำซ้อนของข้อมูลที่มีลักษณะของการซ้ำกันอย่างต่อเนื่องลงให้เหลือเพียงหนึ่งค่า และนับจำนวนของข้อมูลที่มีลักษณะดังกล่าวเป็นข้อมูลชุดใหม่ การจัดเก็บข้อมูลอาจทำได้ในลักษณะของคู่ลำดับของ(ค่าข้อมูล,จำนวน)หรือแยกเป็นข้อมูลคนละชุดก็ได้ วิธี RLC สามารถที่จะแทรกอยู่ระหว่างขบวนการใดๆก็ได้ขึ้นอยู่กับลักษณะของข้อมูล

4.1.1.6 วิธีการเข้ารหัสเอนโทรปี(Entropy Coding)

ก่อนที่จะกล่าวถึงรายละเอียด ในหัวข้อนี้จะใช้คำว่า “ข่าวสาร” แทนคำว่า “ข้อมูล” วิธีการลดขนาดข้อมูลด้วยการเข้ารหัสเอนโทรปีมีพื้นฐานมาจากทฤษฎีของข่าวสาร[9] จุดมุ่งหมายที่สำคัญของการเข้ารหัสเอนโทรปีก็คือต้องการลดความซ้ำซ้อนของข่าวสารเพื่อให้ประสิทธิภาพในการส่งหรือการบันทึกข่าวสารสูงขึ้น ตัวอย่างของการเข้ารหัสเอนโทรปีของข่าวสารซึ่งประกอบด้วยตัวอักษรที่ต่อเนื่องกันสามารถทำได้โดยการแทนตัวอักษรด้วยรหัสใหม่ซึ่งทำให้ข่าวสารใหม่ที่ได้จะมีปริมาณที่น้อยลงหรือสั้นลง รหัสใหม่ที่ถูกสร้างขึ้นมาแทนตัวอักษร(องค์ประกอบของข่าวสารเดิม)นี้เราเรียกว่า “คำรหัส” (Code Word) ซึ่งแต่ละคำรหัสที่ใช้แทนตัวอักษรแต่ละตัวอาจมีความยาวที่เท่าหรือไม่เท่ากันก็ได้ทั้งนี้ขึ้นอยู่กับค่าความน่าจะเป็นของการเกิดของตัวอักษรแต่ละตัว

ถ้ากำหนดให้ U เป็นเซตของข่าวสารที่มีองค์ประกอบของข่าวสารอยู่ M แบบคือ $u_1, u_2, \dots, u_M$ และให้ $p_1, p_2, \dots, p_M$ เป็นค่าความน่าจะเป็นของการเกิดของแต่ละองค์ประกอบ ดังนั้นสามารถหาเอนโทรปีของข่าวสารนี้ได้ในสมการ

$$H(U) = \sum_{k=1}^M p_k \log_2 \frac{1}{p_k} \quad (4.3)$$

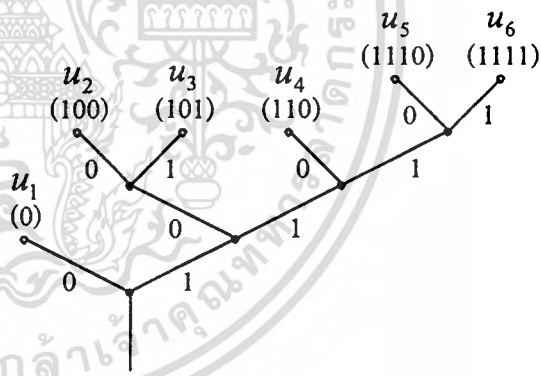
ถ้าองค์ประกอบของข่าวสารในที่นี้นั้น ถ้าข่าวสารเป็นประโยค องค์ประกอบของข่าวสารก็อาจจะเป็นตัวอักษรหรือเป็นคำก็ได้ นอกจากนั้นอาจจะหมายถึงขบวนตัวอักษรที่มีความยาวเป็น n ตัวอักษรเท่ากันหมดก็ได้ ในการเข้ารหัสข่าวสารดังกล่าวนี้ ถ้าเราใช้รหัสฐาน D ที่มีความยาว n_k ในการเข้ารหัสองค์ประกอบของข่าวสารที่เป็น u_k ความยาวเฉลี่ย $\bar{n}$ ของคำรหัสที่ได้จะคำนวณได้จากสมการ

$$\bar{n} = \sum_k n_k p_k \quad (4.4)$$

ในการเข้ารหัสที่ดีก็จะต้องพยายามทำให้ค่า $\bar{n}$ นี้ต่ำที่สุด แต่สิ่งที่จะต้องตระหนักให้ก็คือ รหัสที่สร้างขึ้นมานี้จะต้องไม่มีความกำกวมที่จะทำให้การถอดรหัสออกมาผิดได้ นั่นคือรหัสที่สร้างขึ้นนอกจากจะต้องมีความสัมพันธ์กับข่าวสารเดิมแบบ 1 ต่อ 1 แล้ว เมื่อจับไปเรียงกันเป็นขบวนจะต้องไม่เกิดคำรหัสที่ทำให้การถอดรหัสผิดพลาดได้ คุณสมบัติของรหัสที่จำเป็นจึงต้องเป็นแบบที่สามารถถอดรหัสได้เพียงแบบเดียว(Uniquely Decodable)ด้วย ความหมายของข้อความที่กล่าวมานี้จะสามารถแสดงให้เห็นได้ชัดเจนขึ้นโดยใช้การแสดงการเข้ารหัสในรูปของต้นไม้รหัส(Code Tree) ถ้าทำการสร้างรหัสโดยใช้รหัสฐาน D ถึง ซึ่งหมายถึงสามารถแสดงการสร้างรหัสที่มีเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิพนธ์ใดนำไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติที่ต้องการได้โดยใช้ต้นไม้รหัสแบบแตก D กิ่ง ซึ่งหมายถึงที่แต่ละข้อของต้นไม้จะมีการแตกกิ่งออกไปเป็น D กิ่ง อย่างไรก็ตามเพื่อความสะดวกในการอธิบายในที่นี้เราจะพิจารณากรณีที่ใช้รหัสฐาน 2 ซึ่งแสดงได้ด้วยต้นไม้รหัสแบบ 2 กิ่ง ในที่นี้จะแสดงการสร้างรหัสจากข่าวสารที่มีองค์ประกอบ 6 แบบคือ $u_1, u_2, \dots, u_6$ โดยทำการแตกกิ่งของต้นไม้ตามข้อออกไปเป็น 2 กิ่ง ดังที่แสดงไว้ในรูปที่ 4.2 เมื่อเราทำการแตกกิ่งจนปลายกิ่งทั้งหมดมีจำนวนเท่ากับหรือมากกว่าจำนวนองค์ประกอบของข่าวสาร แล้วให้ปลายกิ่งทั้งหลายกิ่งเป็นรหัสสำหรับองค์ประกอบข่าวสารที่ไม่ซ้ำกัน เมื่อทำเช่นนี้การรหัสที่ได้ออกมาจะมีความยาวไม่เท่ากัน ในขณะเดียวกันก็จะไม่เกิดความกำกวมขึ้น ตารางในรูปที่ 4.3 แสดงรหัสไบนารีที่จะใช้แทนองค์ประกอบข่าวสารแต่ละอันซึ่งจะเห็นได้ว่า ความยาวของรหัสที่สั้นที่สุดก็คือ 1 บิต ความไม่กำกวมนั้นเราอาจจะแสดงได้ง่ายๆโดยทดลองสร้างขบวนคำรหัสลำดับอย่างไรก็ได้ แล้วให้เครื่องถอดรหัสทำการถอดรหัสโดยนำขบวนคำรหัสนั้นป้อนเข้าทางโคdexของต้นไม้รหัส และเมื่อได้ตามกิ่งไม้จนพบรหัสปลายทางแล้วให้มาเริ่มต้นที่โคนต้นไม้ใหม่ เมื่อเครื่องถอดรหัสทำงานเช่นนี้ ระบบการเข้ารหัสและถอดรหัสนี้ก็จะทำงานได้สมบูรณ์ตามเงื่อนไขที่ต้องการ

องค์ประกอบของข่าวสาร	คำรหัส	n_k
u_1	0	$n_1=1$
u_2	100	$n_2=3$
u_3	101	$n_3=3$
u_4	110	$n_4=3$
u_5	1110	$n_5=4$
u_6	1111	$n_6=4$



รูปที่ 4.2 แสดงต้นไม้รหัสแบบ 2 กิ่งที่แสดงการเข้ารหัสของข่าวสารที่มี 6 องค์ประกอบ

เกี่ยวกับการเข้ารหัสข่าวสารให้เครื่องถอดรหัสปลายทางสามารถถอดรหัสได้เพียงแบบเดียว(Uniquely Decodable)นั้นจะมีทฤษฎีพื้นฐานที่สำคัญอยู่ 2 ทฤษฎีด้วยกันคือ

ทฤษฎีที่ 1 ในการเข้ารหัสข่าวสารที่มี M องค์ประกอบด้วยรหัสฐาน D ที่มีคำรหัสยาว $n_1, n_2, \dots, n_M$ ตามลำดับ เงื่อนไขที่จำเป็นและเพียงพอคือ

$$\sum_{k=1}^M D^{-n_k} \leq 1 \tag{4.5}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่เครื่องหมายเท่ากับจะเป็นจริงเมื่อ $D^{-n_k} = p_k$

ทฤษฎีที่ 2 ในการเข้ารหัสขบวนองค์ประกอบข่าวสารที่มีเอ็นโทรปีเป็น H ด้วยรหัสฐาน D ความยาวเฉลี่ยที่สั้นที่สุดของรหัสฐาน D ที่ทำได้นั้นจะเป็นดังนี้

$$\frac{H(U)}{\log_2 D} \leq \bar{n} \leq \frac{H(U)}{\log_2 D} + 1 \tag{4.6}$$

และจะไม่สามารถทำให้ $\bar{n} < H(U)/\log_2 D$ ได้ และเชนนอนได้แสดงให้เห็นว่า ในกรณีของแหล่งกำเนิดข่าวสารแบบไม่มีความจำ $\bar{n}$ เขียนได้ในรูปต่อไปนี้

$$\frac{H(U)}{\log_2 D} \leq \bar{n} \leq \frac{H(U)}{\log_2 D} + \varepsilon \tag{4.7}$$

โดยที่ ε เป็นตัวเลขบวกที่สามารถทำให้เล็กลงจนเป็น 0 ได้
เนื้อหาของทฤษฎีบททั้งสองนี้ก็เท่ากับเป็นการตีกรอบว่าการเข้ารหัสของเราจะทำได้ดีที่สุดที่
สุดแค่ไหน ในกรณีที่เข้ารหัสฐาน 2 นั้น สมการที่ (4.5) และ (4.7) ก็จะเขียนได้เป็น

$$\sum_{k=1}^M 2^{-n_k} = 1 \tag{4.8}$$

และ

$$H(U) \leq \bar{n} \leq H(U) + \varepsilon \tag{4.9}$$

วิธีเข้ารหัสเอ็นโทรปีของข่าวสารที่เสนอโดยฮัฟแมน(Huffman) จัดว่าเป็นวิธีเข้ารหัสเอ็นโทรปีของข่าวสารที่เหมาะสมที่สุด(Optimal Code)แบบหนึ่ง วิธีเข้ารหัสของฮัฟแมนนี้รหัสที่ได้จะมีคุณสมบัติแบบ 1 ต่อ 1 แล้วยังมีความยาวเฉลี่ย $\bar{n}$ ต่ำที่สุดด้วย ถ้าให้แหล่งกำเนิดข่าวสารให้กำเนิดข่าวสารที่แตกต่างกัน M องค์ประกอบ และแต่ละองค์ประกอบมีค่าความน่าจะเป็นของการเกิดเป็น $p_1, p_2, \dots, p_M$ การเข้ารหัสด้วยรหัสฐาน D ตามวิธีการของฮัฟแมนจะมีขั้นตอนดังนี้

ขั้นตอนที่ 1 เรียงลำดับขององค์ประกอบข่าวสารจากค่าความน่าจะเป็นสูงไปหาต่ำ

ขั้นตอนที่ 2 คำนวณตัวเลข m ที่เป็นไปตามเงื่อนไข 2 ประการต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$2 \leq m \leq D, \quad \frac{M-m}{D-1} = \text{เลขจำนวนเต็มที่มีค่าบวก}$$

เสร็จแล้วนำค่าความน่าจะเป็นขององค์ประกอบข่าวสาร m ตัวล่างสุดมารวมกัน แล้วให้ถือว่าเป็นองค์ประกอบข่าวสารใหม่ 1 ตัว

ขั้นตอนที่ 3 นำองค์ประกอบใหม่ที่ได้มานี้มาพิจารณาร่วมกับองค์ประกอบเดิม แล้วเรียงลำดับจากค่าความน่าจะเป็นจากสูงไปหาต่ำ

ขั้นตอนที่ 4 จากลำดับขององค์ประกอบที่ได้ตามขั้นตอนที่ 3 นำค่าความน่าจะเป็นขององค์ประกอบล่างสุด D ตัวมารวมกัน แล้วถือว่าเป็นองค์ประกอบใหม่ แล้วกลับไปทำขั้นตอนที่ 3 จนกว่าผลรวมค่าความน่าจะเป็นเท่ากับ 1

ขั้นตอนที่ 5 ทำการสร้างรหัสต้นไม้อัตโนมัติโดยให้องค์ประกอบข่าวสารเป็นจุดปลายสุดของกิ่ง แล้วโยงกิ่งต่าง ๆ เข้าหากันตามขั้นตอนที่กล่าวมาทั้งหมด

เพื่อให้เห็นภาพที่ชัดเจนขึ้นจะยกตัวอย่างการเข้ารหัสข่าวสารที่มี 5 องค์ประกอบคือ $u_1, u_2, \dots, u_5$ โดยที่มีค่าความน่าจะเป็นของการเกิดเป็น 0.35, 0.25, 0.20, 0.10, 0.10 ตามลำดับ เมื่อเข้ารหัสโดยใช้รหัสฐาน 2 ตามวิธีการของฮัฟแมนตามรายละเอียดข้างล่างนี้ ผลที่ได้จะเป็นดังที่แสดงไว้ในรูปที่ 4.3

ขั้นตอนที่ 1 เรียง u_k ตามลำดับของค่าความน่าจะเป็นจากมากไปหาน้อย สมมุติให้ผลที่ได้คือ $u_1, u_2, \dots, u_5$

ขั้นตอนที่ 2 คำนวณค่า m ได้ $m=2$ รวมค่าความน่าจะเป็นของ u_4, u_5 ได้เป็น 0.20 และให้ผลที่ได้เป็นองค์ประกอบข่าวสารใหม่ u'_4

ขั้นตอนที่ 3 เรียงลำดับของ u_1, u_2, u_3, u'_4 ตามค่าความน่าจะเป็นจากมากไปหาน้อย ซึ่งได้ผลเป็น u_1, u_2, u_3, u'_4 รวมค่าความน่าจะเป็นของ 2 องค์ประกอบล่างสุดคือ u_3, u'_4 เข้าด้วยกัน ผลที่ได้คือ 0.40 และให้องค์ประกอบใหม่ u'_3

ขั้นตอนที่ 4 เรียงลำดับ u_1, u_2, u'_3 ตามค่าความน่าจะเป็นจากสูงไปหาต่ำ ผลที่ได้คือ u'_3, u_1, u_2 จากนั้นทำการรวมค่าความน่าจะเป็นของ 2 องค์ประกอบล่างได้เป็น 0.60 และให้เป็นองค์ประกอบใหม่ u'_1

ขั้นตอนที่ 5 เรียงลำดับใหม่ได้เป็น u_1, u'_1 และเมื่อรวมค่าความน่าจะเป็นแล้วได้เป็น 1 เป็นอันสิ้นสุดการรวมกลุ่ม

ขั้นตอนที่ 6 สร้างต้นไม้รหัสโดยการโยง u_4, u_5 เข้าหากันเป็น u'_4 เสร็จแล้วโยง u'_4 กับ u_3 เข้าหากันเป็น u'_3 หลังจากนั้นโยง u_1, u_2 เข้าหากันเป็น u'_1 แล้วท้ายที่สุดโยง u'_1 กับ u'_3 เข้าหากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การลดขนาดข้อมูล โดยทั่วไปการลดขนาดข้อมูลด้วยวิธีนี้เป็นการลดจำนวนหรือลดระดับความเที่ยงตรง(Precision)ของสัมประสิทธิ์การกระจาย(Expansion Coefficients)ลงให้เหลือเพียงพอที่จะคงไว้ซึ่งสาระสำคัญของข้อมูลหลังจากการแปลงกลับสู่โดเมนเดิม

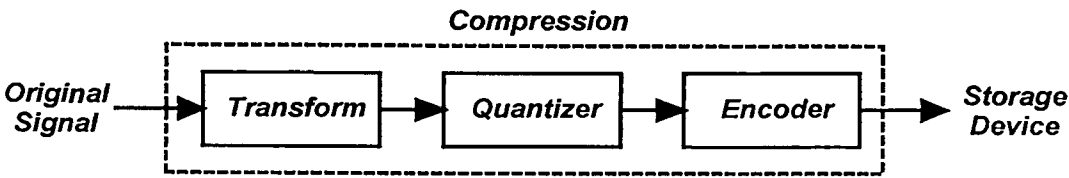
ตัวอย่างของการแปลงเชิงเต็มหน่วยที่มีให้เลือกใช้ ได้แก่ Karhunen-Loeve Transform (KLT) การแปลงฟูเรียร์เต็มหน่วย(Discrete Fourier Transform : DFT) การแปลงโคไซน์เต็มหน่วย(Discrete Cosine Transform : DCT) เป็นต้น KLT เป็นการแปลงที่ให้ผลดีที่สุด(Optimum Transform) ซึ่งทำให้สัมประสิทธิ์ที่ได้จากการแปลง(Transform or Expansion Coefficients)มีลักษณะของ Uncorrelate อย่างสมบูรณ์ ซึ่งจะทำให้ประสิทธิภาพในการลดขนาดข้อมูลสูงมาก อย่างไรก็ตาม KLT นั้นต้องการพลังคำนวณที่สูงมาก(เนื่องจาก KLT ขึ้นตรงกับข้อมูลที่กำลังพิจารณา)และไม่สามารถที่จะใช้วิธีการคำนวณที่รวดเร็ว(Fast Computation)ได้ ดังนั้น KLT จึงมักไม่ถูกเลือกใช้ในงานต่างๆมากนัก การแปลงที่ให้ผลดีใกล้เคียงกับ KLT (Suboptimum Transform) ที่สามารถเลือกใช้คือ DFT และ DCT การแปลงทั้งสองมีสูตรการคำนวณที่แน่นอนไม่ขึ้นกับข้อมูลเหมือน KLT และสามารถใช่วิธีการคำนวณที่รวดเร็วได้(Fast Fourier Transform (FFT) และ Fast Cosine Transform (FCT)) อย่างไรก็ตามก็ไม่นิยมใช้ DFT เนื่องจากจะต้องเกี่ยวข้องกับจำนวนเชิงซ้อน ต่างจากการใช้ DCT ซึ่งดำเนินการข้อมูลด้วยจำนวนจริงทั้งหมดให้ความยุ่งยากน้อยกว่า สมการที่ (4.10) แสดงการแปลงโคไซน์เต็มหน่วยแบบที่ 4 (Discrete Cosine Transform type IV : DCT-IV) ซึ่งเป็นวิธีที่นิยมใช้กันมาก

$$X(k) = \sqrt{\frac{2}{N}} \sum_{j=0}^{N-1} x(j) \cdot \cos \frac{(2j+1)(2k+1)\pi}{4N}, \quad k = 0, 1, \dots, N-1 \quad (4.10)$$

เมื่อ x คือข้อมูลอินพุตและ N คือขนาดของข้อมูลอินพุต ส่วนการแปลงกลับโคไซน์เต็มหน่วย (Inverse Discrete Cosine Transform : IDCT)ใช้สมการเดียวกัน

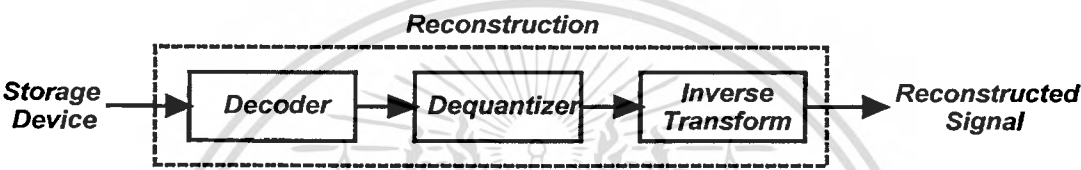
รูปที่ 4.5 แสดงรูปแบบทั่วไปของการลดขนาดข้อมูลด้วยวิธีการแปลง ข้อมูลในโดเมนเดิมหลังจากแปลงไปสู่ข้อมูลในโดเมนใหม่จะนำไปผ่านขบวนการแควนไตซ์(Quantization)หรือการลดระดับความเที่ยงตรงของสัมประสิทธิ์การกระจายลง ข้อมูลที่ได้จากการแควนไตซ์นี้เหมาะที่จะนำไปเข้ารหัสเพื่อการบันทึกต่อไป การเข้ารหัสที่ให้ประสิทธิภาพในการลดขนาดข้อมูลสูงที่นิยมใช้ได้แก่ การเข้ารหัสเอนโทรปี เป็นต้น การลดขนาดข้อมูลด้วยวิธีการการแปลงในรูปที่ 4.5 นี้มีเพียงขบวนการแควนไตซ์เพียงขบวนการเดียวเท่านั้นที่ทำให้เกิดการสูญเสียของข้อมูล ส่วนขบวนการแปลงและขบวนการเข้ารหัสเป็นขบวนการที่สามารถทำย้อนกลับได้อย่างสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 แสดงรูปแบบทั่วไปของการลดขนาดข้อมูลด้วยวิธีการแปลง

สำหรับขบวนการสร้างกลับคืนแสดงในรูปที่ 4.6 ซึ่งเป็นการทำย้อนกลับจากขบวนการลดขนาดข้อมูลในรูปที่ 4.5 นั้นเอง



รูปที่ 4.6 แสดงรูปแบบการสร้างกลับคืนของข้อมูลที่ผ่านการลดขนาดข้อมูลด้วยวิธีการแปลง

นอกจากนี้วิธีการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจยังสามารถแบ่งได้ตามลักษณะของการสูญเสียที่เกิดขึ้นได้อีกสองกลุ่มใหญ่ๆ คือ การลดขนาดข้อมูลแบบที่มีการสูญเสีย(Lossy Compression) และการลดขนาดข้อมูลแบบที่ไม่มีการสูญเสีย(Lossless Compression) การลดขนาดข้อมูลแบบที่มีการสูญเสียได้แก่ วิธี TP วิธี AZTEC วิธี CORTES และวิธี Fan ของวิธีการลดขนาดข้อมูลแบบทางตรง เป็นต้น และยังรวมถึงวิธีการลดขนาดข้อมูลด้วยวิธีการแปลงอีกด้วย ส่วนการลดขนาดข้อมูลแบบที่ไม่มีการสูญเสียได้แก่ วิธี RLC และวิธีเข้ารหัสเอนโทรปี เป็นต้น

4.2 มาตรฐานในการวัดประสิทธิภาพการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจ

ประสิทธิภาพการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจสามารถแสดงด้วยอัตราส่วนระหว่างปริมาณของข้อมูลที่ไม่ได้ผ่านการลดขนาดและปริมาณของข้อมูลที่ได้ผ่านการลดขนาดแล้ว อัตราส่วนดังกล่าวเรียกว่า Compression Ratio (CR) แสดงด้วยสมการ

$$CR = \frac{\text{ปริมาณของข้อมูลที่ไม่ได้ผ่านการลดขนาด}}{\text{ปริมาณของข้อมูลที่ได้ผ่านการลดขนาด}} \tag{4.11}$$

เมื่อปริมาณของข้อมูลที่ไม่ได้ผ่านการลดขนาดในหน่วยไบต์(Byte)สามารถหาได้จากสมการที่

(4.12)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$Bytes = \left\lceil \frac{bps \cdot f_s \cdot t_r}{8} \right\rceil \quad (4.12)$$

โดย bps หมายถึงค่าความละเอียดของ A/D ที่ใช้ในการเก็บสัญญาณ มีหน่วยเป็น Bits/Sample f_s หมายถึงความถี่สุ่มที่ใช้ในการเก็บสัญญาณ มีหน่วยเป็นเฮิร์ตซ์(Hz) t_r หมายถึงระยะเวลาในการบันทึกสัญญาณ มีหน่วยเป็นวินาที และ $\lceil \rceil$ หมายถึงการปัดเศษขึ้น

สำหรับสัญญาณไฟฟ้าหัวใจที่ผ่านการลดขนาดข้อมูลแบบที่มีการสูญเสีย(Lossy Compression) นอกจากค่า CR แล้ว ยังต้องพิจารณาถึงความเพี้ยนของสัญญาณที่เกิดขึ้นด้วย ปกติในทางการแพทย์มักจะใช้วิธีการตรวจสอบสัญญาณที่สร้างกลับคืนด้วยสายตา อย่างไรก็ตามมาตรฐานในการวัดความเพี้ยนของสัญญาณไฟฟ้าหัวใจที่เป็นที่ยอมรับกันทั่วไปสามารถวัดได้จากค่า Percent Root-mean-square Difference (PRD) นิยามด้วยสมการ

$$PRD = \sqrt{\frac{\sum_{n=0}^{N-1} (x(n) - \hat{x}(n))^2}{\sum_{n=0}^{N-1} x^2(n)}} \times 100 \quad (4.13)$$

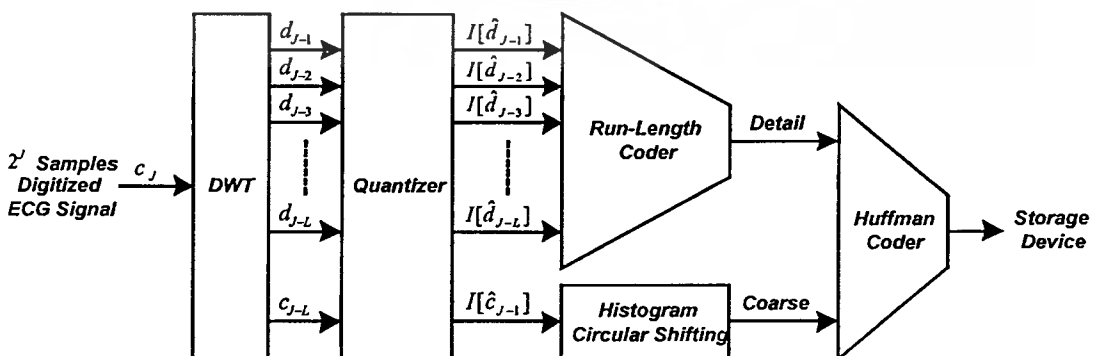
เมื่อ $x(n)$ และ $\hat{x}(n)$ คือ Original และ Reconstructed ECG Data Sequences ตามลำดับ และ N คือจำนวนของข้อมูล โดยค่า PRD ที่สูงแสดงถึงสัญญาณที่ผ่านการลดขนาดข้อมูลมีความเพี้ยนที่สูง

การลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจโดยใช้การแปลงเวฟเลต

แนวความคิดที่นำวิธีการแปลงเวฟเลตมาใช้ในการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจ ประการแรกก็เนื่องมาจากลักษณะการแปลงไปสู่โดเมนของเวลาและความถี่(Time-Frequency Domain)ของการแปลงเวฟเลต ซึ่งให้คุณสมบัติพิเศษคือ การเปลี่ยนแปลงที่เกิดขึ้น ณ ที่ใดๆในโดเมนของเวฟเลต(เนื่องจากขบวนการใดๆ เช่น การแควนไทซ์ เป็นต้น) จะมีผลกระทบในโดเมนของเวลา ณ ตำแหน่งและช่วงเวลาที่สัมพันธ์กับที่นั้นๆเท่านั้น โดยช่วงเวลาดังกล่าวอาจเป็นช่วงเวลาเพียงสั้นๆ หรือมากขึ้นจนตลอดแกนเวลา ลักษณะนี้จะทำให้สามารถรักษาสาระสำคัญของข้อมูลได้อย่างมีประสิทธิภาพ และประการที่สองเนื่องมาจากการแปลงเวฟเลตเป็นการแปลงในลักษณะของ Filter Bank ซึ่งเป็นลักษณะของการแยกข้อมูลจากกลุ่มเดียวออกเป็นหลายๆกลุ่ม ทั้งนี้เพื่อการประมวลผลข้อมูลแต่ละกลุ่มได้อย่างอิสระ ลักษณะนี้ทำให้เกิดประโยชน์ก่อนการเข้ารหัสเพื่อการบันทึก ซึ่งจะกล่าวถึงวิธีการในหัวข้อต่อไป

5.1 ขบวนการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจโดยใช้การแปลงเวฟเลต

ขบวนการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจโดยใช้การแปลงเวฟเลตแสดงในรูปที่ 5.1 สัญญาณไฟฟ้าหัวใจที่ผ่านการดิจิไทซ์(Digitization)จนเป็นข้อมูลเพื่อการประมวลผลในคอมพิวเตอร์จะถูกเตรียมเพื่อเป็นข้อมูลสำหรับการแปลงเวฟเลตเต็มหน่วย(Discrete Wavelet Transform : DWT)โดยการกำหนดให้แต่ละบล็อก(Block)มีขนาด 2^J Samples (J คือจำนวนเต็มบวก) ถ้าพิจารณาให้กลุ่มข้อมูลก่อนการแปลงเวฟเลตแทนด้วย $c_J(n)$ เมื่อ $n = 0, 1, 2, \dots, 2^J - 1$ ข้อมูลนี้เมื่อผ่านการแปลงเวฟเลต L ระดับแล้ว จะได้กลุ่มของข้อมูลย่อย $d_{J-1}(n) |_{n=0,1,2,\dots,2^{J-1}-1}$,

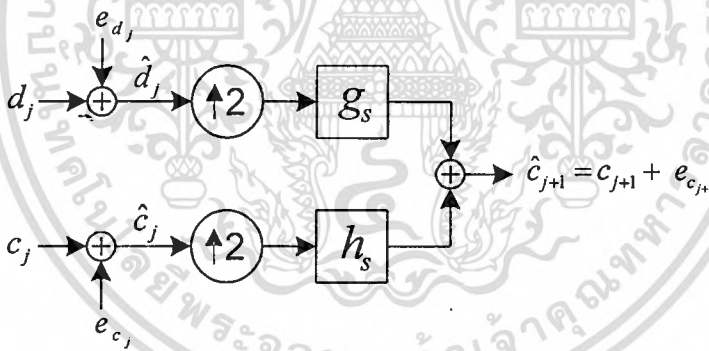


รูปที่ 5.1 แสดงขบวนการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจโดยใช้การแปลงเวฟเลต

$d_{J-2}(n)|_{n=0,1,2,\dots,2^{J-2}-1}, \dots, d_{J-L}(n)|_{n=0,1,2,\dots,2^{J-L}-1}$ และ $c_{J-L}(n)|_{n=0,1,2,\dots,2^{J-L}-1}$ ซึ่งจะผ่านขบวนการควอนไทซ์เพื่อลดช่วงของการแบ่งระดับ(Step Size)ลงต่อไป

5.1.1 การควอนไทซ์สัมประสิทธิ์เวฟเลต

ก่อนที่จะทำการควอนไทซ์(Quantization)กลุ่มข้อมูล $d_{J-1}(n), d_{J-2}(n), \dots, d_{J-L}(n)$ และ $c_{J-L}(n)$ จำเป็นจะต้องพิจารณาแบบจำลองสำหรับการวิเคราะห์ความเพี้ยนที่เกิดขึ้นเนื่องจากการควอนไทซ์สัญญาณอินพุตของ Two-Channel Synthesis Filter Bank ในรูปที่ 5.2 ทั้งนี้เพื่อศึกษาความสัมพันธ์ระหว่างความเพี้ยนที่เกิดขึ้นที่อินพุตและเอาต์พุต ถ้ากำหนดให้ $e_{d_j}(n)$ และ $e_{c_j}(n)$ คือความเพี้ยนที่เกิดขึ้นที่อินพุตเนื่องจากการควอนไทซ์กลุ่มข้อมูล $d_j(n)$ และ $c_j(n)$ ตามลำดับ และ $e_{c_{j+1}}(n)$ คือความเพี้ยนรวมที่จะเกิดขึ้นกับกลุ่มข้อมูล $c_{j+1}(n)$ หลังจากการผ่าน Two-Channel Synthesis Filter Bank สำหรับการควอนไทซ์แบบใดๆ ถ้าพิจารณาให้ค่าเฉลี่ยเชิงสถิติของ $e_{d_j}(n)$ และ $e_{c_j}(n)$ มีค่าเป็นศูนย์ หรือ $E[e_{c_j}(n)] = E[e_{d_j}(n)] = 0$ ดังนั้นสามารถแสดงค่าความแปรปรวน(Variance) σ^2 ของ $e_{d_j}(n)$ และ $e_{c_j}(n)$ ได้ในสมการที่ (5.1) และสมการที่ (5.2)



รูปที่ 5.2 แสดงแบบจำลองของ Two-Channel Synthesis Filter Bank สำหรับการวิเคราะห์ความเพี้ยน

$$\sigma_{e_{d_j}}^2 = \frac{1}{2^J} \sum_n E[e_{d_j}^2(n)] \quad (5.1)$$

$$\sigma_{e_{c_j}}^2 = \frac{1}{2^J} \sum_n E[e_{c_j}^2(n)] \quad (5.2)$$

(เกี่ยวกับค่าเฉลี่ยเชิงสถิติหรือตัวดำเนินการทางคณิตศาสตร์ $E[\cdot]$ และความแปรปรวน(Variance)ดู

ได้จากภาคผนวก ข) จากรูปที่ 5.2 และสมการที่ (3.44) สามารถแสดงได้ว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$c_{j+1}(n) + e_{c_{j+1}}(n) = \sum_k h_s(n-2k) \cdot (c_j(k) + e_{c_j}(k)) + \sum_k g_s(n-2k) \cdot (d_j(k) + e_{d_j}(k)) \quad (5.3)$$

เนื่องจาก Two-Channel Filter Bank เป็นระบบเชิงเส้น(Linear System)ดังนั้นสามารถแสดงได้ว่า

$$e_{c_{j+1}}(n) = \sum_k h_s(n-2k) \cdot e_{c_j}(k) + \sum_k g_s(n-2k) \cdot e_{d_j}(k) \quad (5.4)$$

เช่นเดียวกัน ค่าความแปรปรวนของความเพี้ยนที่เกิดขึ้นที่เอาพุทของ Two-Channel Synthesis Filter Bank สามารถหาได้จาก

$$\begin{aligned} \sigma_{e_{c_{j+1}}}^2 &= \frac{1}{2^{j+1}} \sum_n E[e_{c_{j+1}}^2(n)] \\ &= \frac{1}{2^{j+1}} \sum_n \left\{ \sum_k \sum_l h_s(n-2k) \cdot h_s(n-2l) \cdot E[e_{c_j}(k) \cdot e_{c_j}(l)] \right. \\ &\quad + \sum_k \sum_l h_s(n-2k) \cdot g_s(n-2l) \cdot E[e_{c_j}(k) \cdot e_{d_j}(l)] \\ &\quad + \sum_k \sum_l g_s(n-2k) \cdot h_s(n-2l) \cdot E[e_{d_j}(k) \cdot e_{c_j}(l)] \\ &\quad \left. + \sum_k \sum_l g_s(n-2k) \cdot g_s(n-2l) \cdot E[e_{d_j}(k) \cdot e_{d_j}(l)] \right\} \end{aligned}$$

สามารถจัดให้อยู่ในรูปใหม่ได้ในสมการ

$$\begin{aligned} \sigma_{e_{c_{j+1}}}^2 &= \frac{1}{2^{j+1}} \sum_k \sum_l \left\{ \sum_n h_s(n-2k) \cdot h_s(n-2l) \cdot E[e_{c_j}(k) \cdot e_{c_j}(l)] \right. \\ &\quad + \sum_n h_s(n-2k) \cdot g_s(n-2l) \cdot E[e_{c_j}(k) \cdot e_{d_j}(l)] \\ &\quad + \sum_n g_s(n-2k) \cdot h_s(n-2l) \cdot E[e_{d_j}(k) \cdot e_{c_j}(l)] \\ &\quad \left. + \sum_n g_s(n-2k) \cdot g_s(n-2l) \cdot E[e_{d_j}(k) \cdot e_{d_j}(l)] \right\} \quad (5.5) \end{aligned}$$

จากคุณสมบัติของออโธโนร์มัลเวฟเลต(Orthonormal Wavelet)ในสมการที่ (3.49) และ (3.50) แทนในสมการที่ (5.5) จะได้

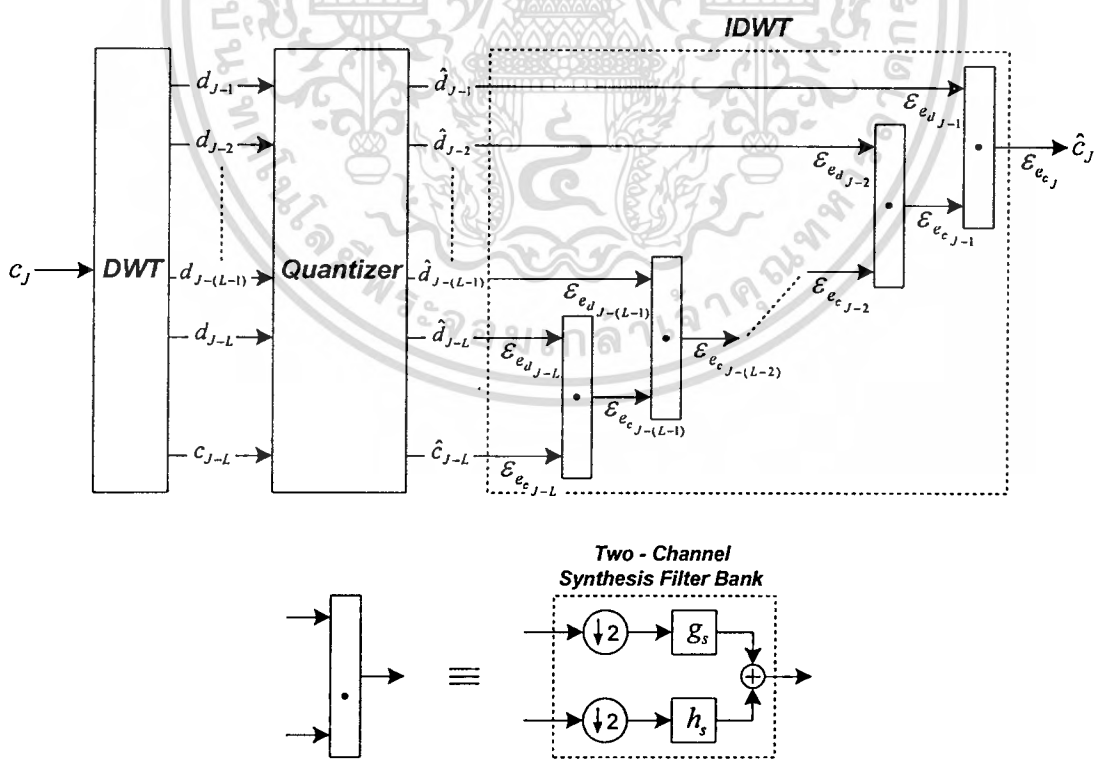
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{aligned}\sigma_{e_{c_{j+1}}}^2 &= \frac{1}{2^{j+1}} \sum_k E[e_{c_j}^2(k)] + \frac{1}{2^{j+1}} \sum_k E[e_{d_j}^2(k)] \\ &= \frac{1}{2} \sigma_{e_{c_j}}^2 + \frac{1}{2} \sigma_{e_{d_j}}^2\end{aligned}\tag{5.6}$$

สมการที่ (5.6) แสดงให้เห็นว่าสำหรับกรณีของออโคโนร์มัลเวฟเลต ค่าความเพี้ยนยกกำลังสองเฉลี่ย(Mean Square Error : MSE) ε ที่เกิดขึ้นที่เอาต์พุตของ Two-Channel Synthesis Filter Bank จะเท่ากับผลรวมของ MSE ที่เกิดขึ้นที่อินพุตของ Two-Channel Synthesis Filter Bank หรือแสดงด้วยสมการ

$$\varepsilon_{e_{c_{j+1}}} = \frac{1}{2} \varepsilon_{e_{c_j}} + \frac{1}{2} \varepsilon_{e_{d_j}}\tag{5.7}$$

ดังนั้นเมื่อพิจารณาในกรณีของการแปลงกลับเวฟเลตเต็มหน่วย L ระดับในลักษณะของ Octave Band Synthesis Filter Bank ในรูปที่ 5.3 สามารถแสดงความสัมพันธ์ของ MSE ระหว่างระดับ เช่นเดียวกับสมการที่ (5.7) ได้คือ



รูปที่ 5.3 แสดงแบบจำลองการวิเคราะห์ความเพี้ยนที่เกิดขึ้นเนื่องการแวนไดซ์ของขบวนการลดขนาดข้อมูลด้วยวิธีการแปลงเวฟเลต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{aligned}
 \varepsilon_{e_{c_j}} &= \frac{1}{2} \varepsilon_{e_{c_{j-1}}} + \frac{1}{2} \varepsilon_{e_{d_{j-1}}} \\
 \varepsilon_{e_{c_{j-1}}} &= \frac{1}{2} \varepsilon_{e_{c_{j-2}}} + \frac{1}{2} \varepsilon_{e_{d_{j-2}}} \\
 \varepsilon_{e_{c_{j-2}}} &= \frac{1}{2} \varepsilon_{e_{c_{j-3}}} + \frac{1}{2} \varepsilon_{e_{d_{j-3}}} \\
 &\vdots \\
 \varepsilon_{e_{c_{j-(L-1)}}} &= \frac{1}{2} \varepsilon_{e_{c_{j-L}}} + \frac{1}{2} \varepsilon_{e_{d_{j-L}}}
 \end{aligned} \tag{5.8}$$

สำหรับอัลกอริทึมการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจที่ได้ออกแบบนี้ ได้เลือกใช้การแควนไตซ์แบบยูนิฟอร์ม(Uniform Quantization) เนื่องจากเป็นวิธีที่ให้ความสะดวกรวดเร็วในการประมวลผล และได้พิจารณาให้ค่าความเพี้ยนที่เกิดขึ้นจากการแควนไตซ์สัมประสิทธิ์ของเวฟเลต d และ c ที่ระดับเดียวกันมีค่าเท่ากันหรือแสดงโดย

$$\begin{aligned}
 \varepsilon_{e_{d_{j-1}}} &= \varepsilon_{e_{c_{j-1}}} \\
 \varepsilon_{e_{d_{j-2}}} &= \varepsilon_{e_{c_{j-2}}} \\
 \varepsilon_{e_{d_{j-3}}} &= \varepsilon_{e_{c_{j-3}}} \\
 &\vdots \\
 \varepsilon_{e_{d_{j-L}}} &= \varepsilon_{e_{c_{j-L}}}
 \end{aligned} \tag{5.9}$$

จากสมการที่ (5.8) และ (5.9) สามารถแสดงได้ว่า

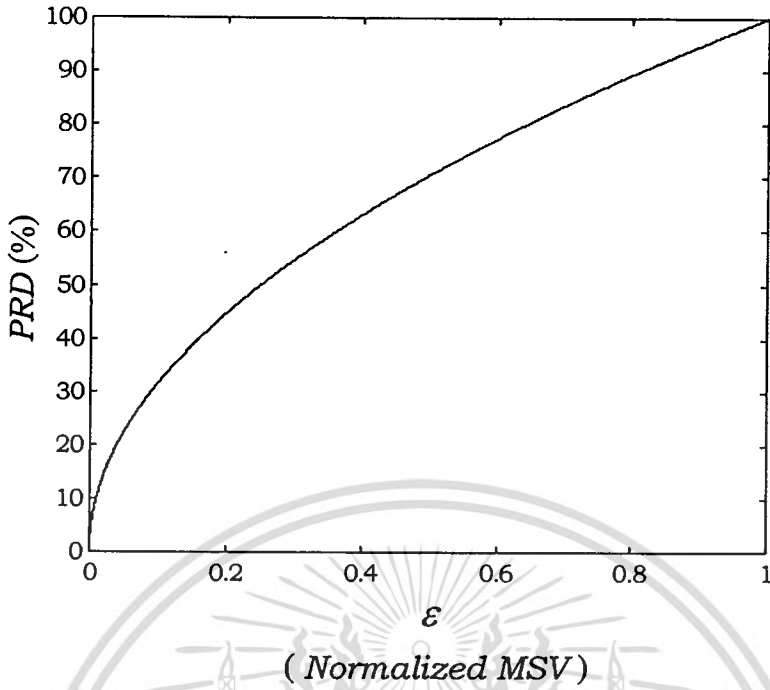
$$\varepsilon_{e_{c_j}} = \varepsilon_{e_{d_{j-1}}} = \varepsilon_{e_{d_{j-2}}} = \varepsilon_{e_{d_{j-3}}} = \dots = \varepsilon_{e_{d_{j-L}}} = \varepsilon_{e_{c_{j-L}}} \tag{5.10}$$

เป็นที่ทราบว่าการแควนไตซ์แบบยูนิฟอร์ม ค่าความแปรปรวนหรือ MSE สามารถแสดงด้วยสมการ $\sigma^2 = \varepsilon = \frac{1}{12} \Delta^2$ เมื่อ Δ คือช่วงของการแบ่งระดับในการแควนไตซ์(Quantization Step Size) แทนในสมการที่ (5.10) จะได้ช่วงของการแบ่งระดับ $\Delta_{d_{j-1}}, \Delta_{d_{j-2}}, \dots, \Delta_{d_{j-L}}$ และ $\Delta_{c_{j-L}}$ ในการแควนไตซ์กลุ่มข้อมูล $d_{j-1}(n), d_{j-2}(n), \dots, d_{j-L}(n)$ และ $c_{j-L}(n)$ ตามลำดับ มีความสัมพันธ์กันคือ

$$\Delta_{d_{j-1}} = \Delta_{d_{j-2}} = \Delta_{d_{j-3}} = \dots = \Delta_{d_{j-L}} = \Delta_{c_{j-L}} \tag{5.11}$$

นั่นคือกลุ่มข้อมูล $d_{j-1}(n), d_{j-2}(n), \dots, d_{j-L}(n)$ และ $c_{j-L}(n)$ ทุกกลุ่มจะผ่านการแควนไตซ์ด้วยช่วงของการแบ่งระดับ Δ ที่เท่ากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

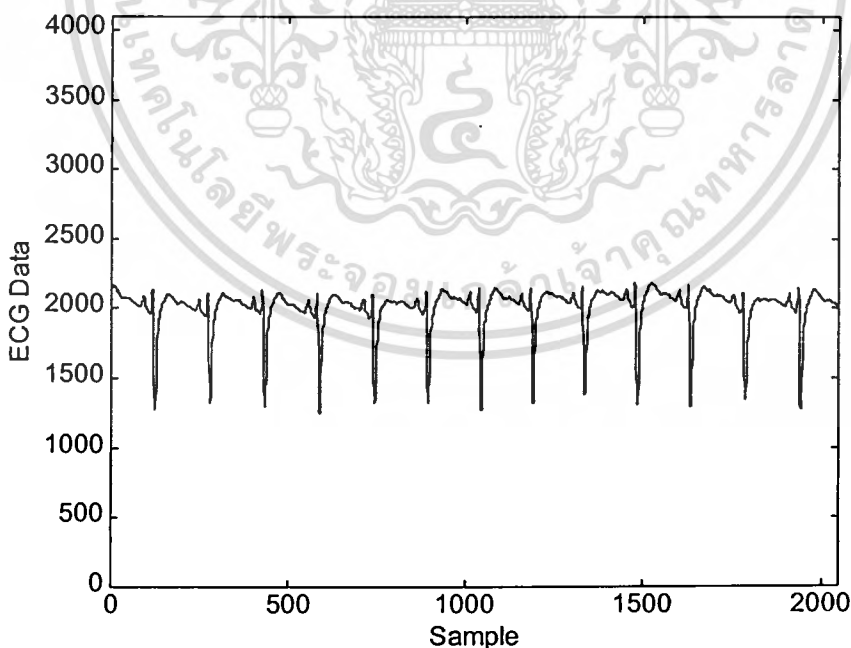


รูปที่ 5.4 แสดงความสัมพันธ์ระหว่างค่า PRD และ ε

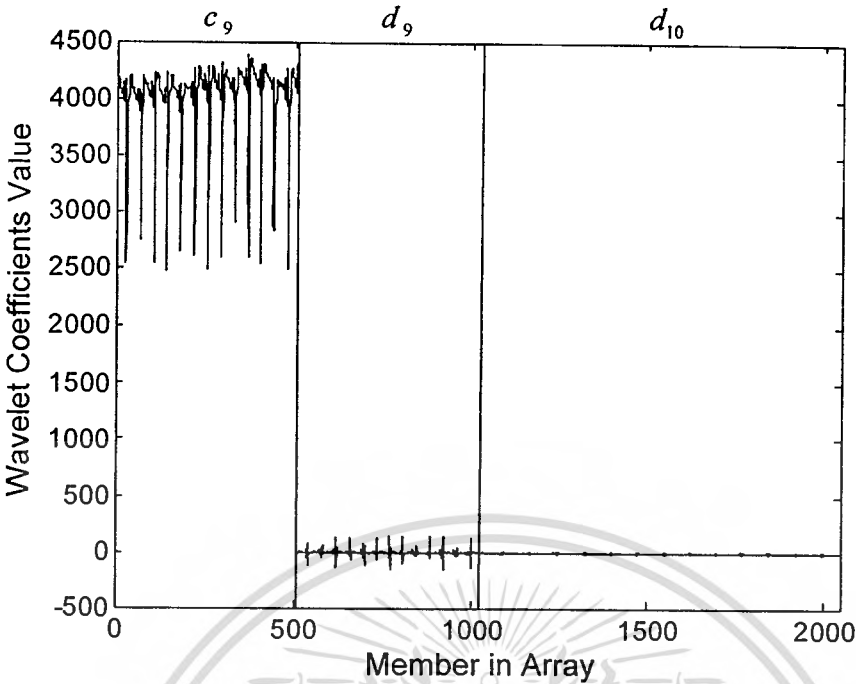
ขบวนการแควนไตซ์ในขบวนการลดขนาดข้อมูลที่ได้ออกแบบแสดงในสมการที่ (5.15) ซึ่งเป็นการแปลงให้อยู่ในรูปกึ่งสัญลักษณ์(Intermediate Symbol)เพื่อการเข้ารหัส จากสมการ *round* หมายถึงการปัดเศษให้เป็นเลขจำนวนเต็ม $\hat{x}(n)$ หมายถึงกลุ่มข้อมูลที่ได้จากการแควนไตซ์กลุ่มข้อมูล $x(n)$ ด้วยช่วงของการแบ่งระดับ Δ มีความสัมพันธ์กันคือ $x(n) = \Delta \cdot \hat{x}(n)$ และสัญลักษณ์ $I[\hat{x}(n)]$ หมายถึงการแทนกลุ่มข้อมูล $\hat{x}(n)$ ที่ได้จากการแควนไตซ์ด้วยเลขจำนวนเต็ม

$$\begin{aligned}
 I[\hat{d}_{J-1}(n)] &= \text{round}\left(\frac{d_{J-1}(n)}{\Delta}\right) \Big|_{n=0,1,2,\dots,2^{J-1}-1} \\
 I[\hat{d}_{J-2}(n)] &= \text{round}\left(\frac{d_{J-2}(n)}{\Delta}\right) \Big|_{n=0,1,2,\dots,2^{J-2}-1} \\
 I[\hat{d}_{J-3}(n)] &= \text{round}\left(\frac{d_{J-3}(n)}{\Delta}\right) \Big|_{n=0,1,2,\dots,2^{J-3}-1} \\
 &\vdots \\
 I[\hat{d}_{J-L}(n)] &= \text{round}\left(\frac{d_{J-L}(n)}{\Delta}\right) \Big|_{n=0,1,2,\dots,2^{J-L}-1} \\
 I[\hat{c}_{J-L}(n)] &= \text{round}\left(\frac{c_{J-L}(n)}{\Delta}\right) \Big|_{n=0,1,2,\dots,2^{J-L}-1}
 \end{aligned} \tag{5.15}$$

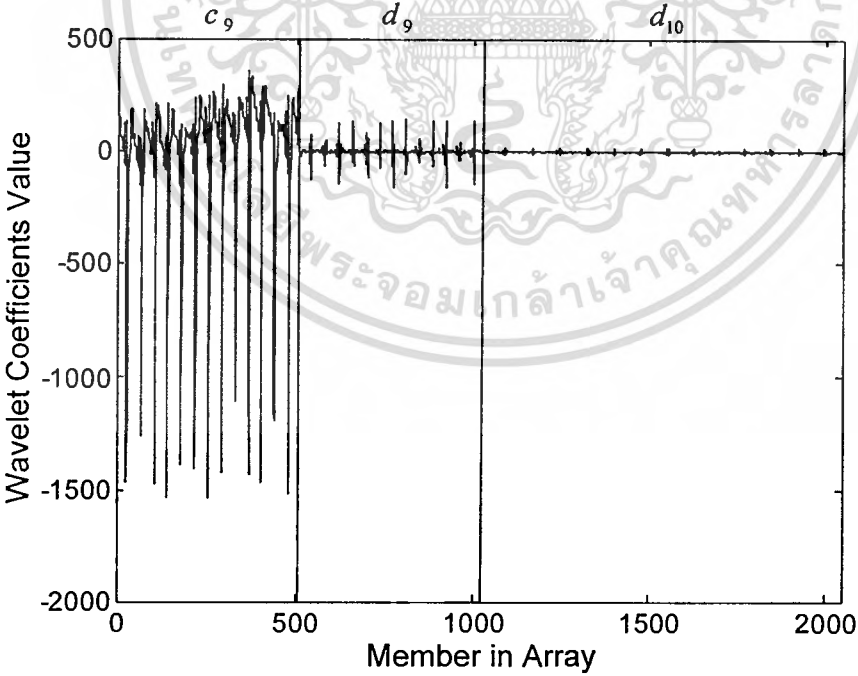
รูปที่ 5.5 แสดงตัวอย่างข้อมูลสัญญาณไฟฟ้าหัวใจที่ได้จาก A/D ขนาดความละเอียด 12 Bits/Sample ที่ความถี่สุ่ม 200 Hz จำนวน $2^{11} = 2048$ Samples เมื่อระดับอ้างอิงของสัญญาณ (0 โวลต์) อยู่ที่ 2048 ซึ่งจะกำหนดให้ข้อมูลที่ได้นี้คือ $c_J(n)$ เมื่อ $J=11$ กลุ่มข้อมูล $c_{11}(n)$ จะเป็นตัวเลขจำนวนเต็มบวกระหว่าง 0 ถึง 4095 และลอยอยู่ที่ระดับอ้างอิง 2048 ซึ่งเป็นลักษณะของข้อมูลที่มีค่าเฉลี่ยที่สูงมาก มีผลทำให้กลุ่มข้อมูลในแบนด์ความถี่ต่ำหรือ Coarse Wavelet Coefficients ในโดเมนของเวฟเลตมีค่าเฉลี่ยที่สูงตามไปด้วย รูปที่ 5.6 เป็นการแสดงตัวอย่างกลุ่มข้อมูลที่ได้จากการแปลงเวฟเลต 2 ระดับกลุ่มข้อมูล $c_{11}(n)$ และนำกลุ่มข้อมูลที่ได้ทุกกลุ่มมารวมอยู่ในอาร์เรย์เดียวกัน เมื่อพิจารณาจากรูปจะเห็นว่าข้อมูลในแบนด์ความถี่สูงหรือ Detail Wavelet Coefficients $d_9(n)$ และ $d_{10}(n)$ หลังจากการแควนไตซ์ด้วยสมการที่ (5.12) แล้ว จะได้ข้อมูลที่อยู่ในเซตของเลขจำนวนเต็ม $\{\dots, -2, -1, 0, 1, 2, \dots\}$ ส่วนกลุ่มข้อมูลในแบนด์ความถี่ต่ำหรือ Coarse Wavelet Coefficients $c_9(n)$ ค่าที่ได้จากการแควนไตซ์จะอยู่ในเซตของจำนวนเต็มบวกที่สูงมาก เนื่องจากสัญลักษณ์ของจำนวนเต็มที่ได้จากการแควนไตซ์จะผ่านการเข้ารหัสร่วมกัน ความแตกต่างระหว่างเลขจำนวนเต็มที่ได้จากการแควนไตซ์กลุ่มข้อมูล $c_9(n)$ และกลุ่มข้อมูล $d_9(n)$ และ $d_{10}(n)$ ดังที่กล่าวมาจะทำให้การเข้ารหัสมีประสิทธิภาพต่ำกว่าที่ควร ดังนั้นกลุ่มข้อมูล $c_{11}(n)$ จึงควรถูกลบออกด้วยค่าเฉลี่ยของกลุ่มก่อนขบวนการลดขนาดข้อมูล รูปที่ 5.7 เป็น



รูปที่ 5.5 แสดงตัวอย่างข้อมูลสัญญาณไฟฟ้าหัวใจที่ได้จาก A/D ขนาดความละเอียด 12 Bits/Sample จำนวน 2048 Samples เมื่อระดับอ้างอิงของสัญญาณ (0 โวลต์) อยู่ที่ 2048



รูปที่ 5.6 แสดงกลุ่มข้อมูลที่ได้จากการแปลงเวฟเล็ต 2 ระดับของตัวอย่างข้อมูลสัญญาณไฟฟ้า หัวใจในรูปที่ 5.5 และนำกลุ่มข้อมูลทุกกลุ่มมารวมอยู่ในอาร์เรย์เดียวกัน



รูปที่ 5.7 แสดงกลุ่มข้อมูลที่ได้จากการแปลงเวฟเล็ต 2 ระดับของตัวอย่างข้อมูลสัญญาณไฟฟ้า หัวใจในรูปที่ 5.5 โดยที่ $c_{11}(n)$ ถูกลบออกด้วยค่าเฉลี่ยของกลุ่มก่อนผ่านขบวนการลดขนาดข้อมูล

การแสดงกลุ่มข้อมูลที่ได้จากการแปลงเวฟเลต 2 ระดับของกลุ่มข้อมูล $c_{11}(n)$ โดยที่กลุ่มข้อมูล $c_{11}(n)$ ถูกลบออกด้วยค่าเฉลี่ยของกลุ่มก่อนผ่านขบวนการลดขนาดข้อมูล จากภาพจะเห็นว่าเมื่อสัมประสิทธิ์เวฟเลตผ่านการแควนไตซ์แล้ว ข้อมูลที่ได้จะอยู่ในเซตของเลขจำนวนเต็ม $\{\dots, -2, -1, 0, 1, 2, \dots\}$ ทุกกลุ่ม ทำให้ประสิทธิภาพการลดขนาดข้อมูลสูงขึ้น จากที่กล่าวมา ทำให้จำเป็นต้องบันทึกค่าเฉลี่ยของ $c_j(n)$ เพื่อใช้ในขบวนการสร้างสัญญาณกลับคืนต่อไป

5.1.2 การเพิ่มประสิทธิภาพการลดขนาดข้อมูลด้วยการเข้ารหัส Run-Length

เนื่องจากกลุ่มข้อมูล $d_{j-1}(n), d_{j-2}(n), \dots, d_{j-L}(n)$ เป็นข้อมูลในส่วนของแบนด์ความถี่สูงหรือเป็นส่วนที่แสดงรายละเอียด(Detail)ของสัญญาณ จึงทำให้ข้อมูลส่วนใหญ่จะมีขนาดที่เล็กมาก ดังนั้นจึงทำให้กลุ่มข้อมูล $I[\hat{d}_{j-1}(n)], I[\hat{d}_{j-2}(n)], \dots, I[\hat{d}_{j-L}(n)]$ ที่ได้จากการแควนไตซ์กลุ่มข้อมูล $d_{j-1}(n), d_{j-2}(n), \dots, d_{j-L}(n)$ จึงปรากฏค่าศูนย์อยู่เป็นจำนวนมาก ซึ่งเหมาะที่จะนำไปผ่านขบวนการเข้ารหัส Run-Length เพื่อลดจำนวนศูนย์ที่มีอยู่อย่างต่อเนื่องในแต่ละช่วงออกไปให้เหลือศูนย์เพียงค่าเดียว ทำให้ประสิทธิภาพในการลดขนาดข้อมูลสูงขึ้น และเนื่องจากขบวนการเข้ารหัส Run-Length เป็นขบวนการที่สามารถทำย้อนกลับได้อย่างสมบูรณ์ จึงไม่ทำให้เกิดการสูญเสียของข้อมูล การเข้ารหัส Run-Length จะกำหนดจำนวนกลุ่มข้อมูลที่จะทำการเข้ารหัส เช่น ถ้ากำหนดให้ทำการเข้ารหัสเพียง 1 กลุ่ม ก็จะทำการเข้ารหัสเฉพาะกลุ่มข้อมูล $I[\hat{d}_{j-1}(n)]$ เท่านั้น ถ้ากำหนดให้ทำการเข้ารหัส 2 กลุ่ม ก็จะทำการเข้ารหัสกลุ่มข้อมูล $I[\hat{d}_{j-1}(n)]$ และ $I[\hat{d}_{j-2}(n)]$ เรียงต่อกัน เป็นต้น ทั้งนี้การกำหนดจำนวนกลุ่มข้อมูลที่จะผ่านขบวนการเข้ารหัส Run-Length จะขึ้นอยู่กับช่วงของการแบ่งระดับ Δ ที่ใช้ในการแควนไตซ์ ถ้าช่วงของการแบ่งระดับ Δ มีค่าน้อย ค่าศูนย์ที่ปรากฏก็จะมีจำนวนน้อย จำนวนกลุ่มข้อมูลที่เหมาะสมจะทำการเข้ารหัส Run-Length ก็จะมีน้อย ในทางตรงกันข้าม ถ้าช่วงของการแบ่งระดับ Δ มีค่ามาก ค่าศูนย์ที่ปรากฏก็จะมีจำนวนมาก จำนวนกลุ่มข้อมูลที่เหมาะสมจะทำการเข้ารหัส Run-Length ก็จะมีมาก หลังจากขบวนการเข้ารหัส Run-Length ข้อมูลที่จำเป็นต้องบันทึกเพื่อใช้ในการสร้างสัญญาณกลับคืนคือ อาร์เรย์ของจำนวนศูนย์ที่นำออกจาก $I[\hat{d}(n)]$ (เฉพาะแบนด์ที่ทำการเข้ารหัส) ค่าความยาวหรือจำนวนสมาชิกของอาร์เรย์ดังกล่าว และค่าความยาวรวมของ $I[\hat{d}(n)]$ (เฉพาะแบนด์ที่ทำการเข้ารหัส) หลังจากผ่านขบวนการเข้ารหัส Run-Length

5.1.3 การเพิ่มประสิทธิภาพการลดขนาดข้อมูลด้วยวิธี Histogram Circular Shifting

เนื่องจากกลุ่มข้อมูล $c_{j-L}(n)$ เป็นข้อมูลในส่วนของแบนด์ความถี่ต่ำหรือเป็นส่วนที่แสดงลักษณะแบบหยาบๆ(Coarse)ของสัญญาณ จึงทำให้ข้อมูลส่วนใหญ่จึงมีขนาดที่ใหญ่ ดังนั้นเมื่อพิจารณาฮิสโตแกรม(Histogram)ของกลุ่มข้อมูล $I[\hat{c}_{j-L}(n)]$ ที่ได้จากขบวนการแควนไตซ์

กลุ่มข้อมูล $c_{j-L}(n)$ จึงมีลักษณะการกระจายตัวของข้อมูลหนาแน่นบริเวณค่าใด ๆ ที่ไม่ใช่ศูนย์(ต่างจากฮิสโตแกรมของกลุ่มข้อมูล $I[\hat{d}_{j-1}(n)], I[\hat{d}_{j-2}(n)], \dots, I[\hat{d}_{j-L}(n)]$ ซึ่งมีลักษณะการกระจายตัวของข้อมูลหนาแน่นบริเวณศูนย์) ดังนั้นจึงสามารถใช้วิธี Histogram Circular Shifting กับกลุ่มข้อมูล $I[\hat{c}_{j-L}(n)]$ เพื่อเลื่อนตำแหน่งที่มีลักษณะการกระจายตัวของข้อมูลหนาแน่นมากไปอยู่บริเวณศูนย์ ซึ่งเป็นการเพิ่มจำนวนศูนย์ที่มีอยู่ในกลุ่มข้อมูล $I[\hat{d}_{j-1}(n)], I[\hat{d}_{j-2}(n)], \dots, I[\hat{d}_{j-L}(n)]$ ให้มีค่ามากขึ้น ทำให้สามารถลดค่าความยาวบิตในการเข้ารหัสฮัฟฟแมน(Huffman Coding)ลดได้ ทำให้ประสิทธิภาพในการลดขนาดข้อมูลสูงขึ้น และเนื่องจากขบวนการ Histogram Circular Shifting เป็นขบวนการที่สามารถทำย้อนกลับได้อย่างสมบูรณ์ จึงไม่ทำให้เกิดการสูญเสียของข้อมูล หลังจากขบวนการ Histogram Circular Shifting ข้อมูลที่จำเป็นต้องบันทึกเพื่อใช้ในการสร้างสัญญาณกลับคืนคือ ขนาดสูงสุดสัมบูรณ์ของ $I[\hat{c}_{j-L}(n)]$ และจำนวนตำแหน่งที่เลื่อนไปของฮิสโตแกรม

5.1.4 การเข้ารหัสฮัฟฟแมน

หลังจากที่กลุ่มข้อมูล $I[\hat{d}_{j-1}(n)], I[\hat{d}_{j-2}(n)], \dots, I[\hat{d}_{j-L}(n)]$ และ $I[\hat{c}_{j-L}(n)]$ ผ่านการเพิ่มประสิทธิภาพการลดขนาดข้อมูลด้วยการเข้ารหัส Run-Length และวิธี Histogram Circular Shifting ตามลำดับแล้ว ข้อมูลที่ได้จะมีจำนวนที่น้อยลงและยังคงอยู่ในเซตของจำนวนเต็ม $\{\dots, -2, -1, 0, 1, 2, \dots\}$ ในอาเรย์เดียวกัน จากนั้นจึงนำข้อมูลมาสร้างตารางฮัฟฟแมนเพื่อใช้ในการเข้ารหัสข้อมูลเพื่อการบันทึกต่อไป หลังจากขบวนการฮัฟฟแมนข้อมูลที่ต้องบันทึกเพื่อใช้ในการสร้างสัญญาณกลับคืนคือ ความยาวของตารางฮัฟฟแมน อาเรย์ของข้อมูลที่ถูกเข้ารหัส อาเรย์ของค่ารหัสฮัฟฟแมน อาเรย์ของค่าความยาวบิตของรหัสฮัฟฟแมน และจำนวนบิตรวมของข้อมูลที่ผ่านการเข้ารหัสฮัฟฟแมน

5.2 ขบวนการสร้างสัญญาณกลับคืน

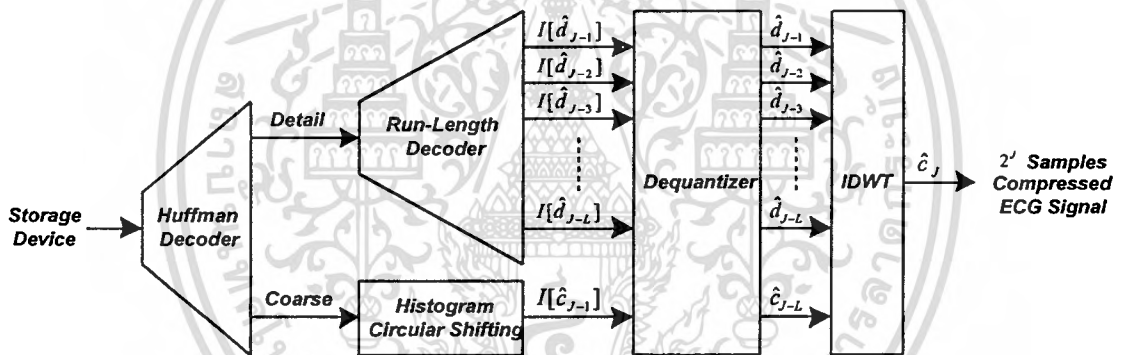
ขบวนการสร้างสัญญาณกลับคืนแสดงในรูปที่ 5.8 ขบวนการสร้างสัญญาณกลับคืนมีลักษณะที่ย้อนกลับกับขบวนการลดขนาดข้อมูลในรูปที่ 5.1 ข้อมูลที่อ่านจากอุปกรณ์บันทึกทีละ 8 บิต จะนำมาเรียงต่อกันเพื่อใช้ในการเทียบรหัสในตารางฮัฟฟแมนที่บันทึกไว้ หลังจากการถอดรหัสฮัฟฟแมน(Huffman Decoding)จะได้กลุ่มข้อมูลในเซตของเลขจำนวนเต็ม $\{\dots, -2, -1, 0, 1, 2, \dots\}$ จากนั้นจึงทำการแยกกลุ่มข้อมูลออกเป็น 2 ส่วนคือ ส่วนของ Detail และ Coarse โดยส่วนของ Detail จะนำมาผ่านขบวนการถอดรหัส Run-Length เพื่อการแทรกศูนย์คืน และส่วนของ Coarse จะนำมาผ่านขบวนการ Histogram Circular Shifting เพื่อทำการเลื่อนฮิสโตแกรมคืนสู่ตำแหน่งเดิม หลังจากขบวนการถอดรหัส Run-Length และขบวนการ Histogram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Circular Shifting จะได้กลุ่มข้อมูล $I[\hat{d}_{J-1}(n)], I[\hat{d}_{J-2}(n)], \dots, I[\hat{d}_{J-L}(n)]$ และ $I[\hat{c}_{J-L}(n)]$ กลับคืนมา จากนั้นจึงนำกลุ่มข้อมูลทุกกลุ่มไปผ่านขบวนการดีควอนไทซ์(Dequantization)ด้วยสมการ

$$\begin{aligned}\hat{d}_{J-1}(n) &= \Delta \cdot I[\hat{d}_{J-1}(n)] \Big|_{n=0,1,2,\dots,2^{J-1}-1} \\ \hat{d}_{J-2}(n) &= \Delta \cdot I[\hat{d}_{J-2}(n)] \Big|_{n=0,1,2,\dots,2^{J-2}-1} \\ \hat{d}_{J-3}(n) &= \Delta \cdot I[\hat{d}_{J-3}(n)] \Big|_{n=0,1,2,\dots,2^{J-3}-1} \\ &\vdots \\ \hat{d}_{J-L}(n) &= \Delta \cdot I[\hat{d}_{J-L}(n)] \Big|_{n=0,1,2,\dots,2^{J-L}-1} \\ \hat{c}_{J-L}(n) &= \Delta \cdot I[\hat{c}_{J-L}(n)] \Big|_{n=0,1,2,\dots,2^{J-L}-1}\end{aligned}\quad (5.16)$$

กลุ่มข้อมูล $\hat{d}_{J-1}(n), \hat{d}_{J-2}(n), \dots, \hat{d}_{J-L}(n)$ และ $\hat{c}_{J-L}(n)$ ที่ได้จะผ่านการแปลงกลับเวฟเลตเต็มหน่วยเพื่อสร้างสัญญาณคลื่นไฟฟ้าหัวใจกลับคืนมา จากนั้น $\hat{c}_J(n)$ จะถูกบวกด้วยค่าเฉลี่ยของ $c_J(n)$ ที่บันทึกไว้ก่อนการแสดงผลต่อไป



รูปที่ 5.8 แสดงขบวนการสร้างสัญญาณกลับคืน

บทที่ 6

การทดสอบประสิทธิภาพการลดขนาดข้อมูล

ในบทนี้เป็นการทดสอบการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจโดยใช้วิธีการแปลงเวฟเล็ต ผลการทดสอบได้แสดงให้เห็นถึงประสิทธิภาพการลดขนาดข้อมูลด้วยค่า PRD และการเปรียบเทียบลักษณะของสัญญาณก่อนและหลังผ่านขบวนการลดขนาดข้อมูลเมื่อกำหนดให้ค่า CR คงที่ค่าต่างๆ ตลอดจนการแสดงผลประสิทธิภาพการลดขนาดข้อมูลเปรียบเทียบกับวิธีการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจแบบอื่นๆ โดยผลการทดสอบทั้งหมดใช้โปรแกรม Matlab® ช่วยในการคำนวณและแสดงผล

6.1 การทดสอบการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจ

ในเบื้องต้นของการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจด้วยคอมพิวเตอร์จำเป็นต้องกำหนดค่าความละเอียดและความถี่สุ่มที่ใช้ในการเก็บข้อมูล ในการทดสอบการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจ ณ ที่นี้จะกำหนดให้สัญญาณไฟฟ้าหัวใจที่นำมาทดสอบถูกเก็บมาด้วยความละเอียด 12 Bits/Sample ที่ความถี่สุ่ม 200 Hz ซึ่งเป็นความถี่ที่ต่ำที่สุดที่ใช้ในการบันทึกสัญญาณไฟฟ้าหัวใจเพื่อการวินิจฉัยอย่างละเอียด และเนื่องจากขบวนการลดขนาดข้อมูลที่ได้ออกแบบขึ้นเป็นขบวนการลดขนาดข้อมูลที่ละเล็กละน้อยต่อเนื่องกันไป ดังนั้นในการทดสอบจะกำหนดให้แต่ละบล็อกของข้อมูลมีขนาดเป็น 2048 Samples (10.24 วินาที) ซึ่งเป็นการกำหนดค่า J จากสมการที่ (3.31) เป็น 11 สัญญาณไฟฟ้าหัวใจที่เลือกใช้ในการทดสอบเป็นสัญญาณไฟฟ้าหัวใจแบบ Unipolar Chest Lead V_1 เพียงลีดเดียวเท่านั้น ประกอบด้วย

- ตัวอย่างสัญญาณไฟฟ้าหัวใจปกติทั่วไป ได้จากการเก็บตัวอย่างจากบุคคลทั่วไปที่มีจังหวะการเต้นหัวใจสม่ำเสมอจำนวน 5 ตัวอย่างๆละ 10 บล็อก
- ตัวอย่างสัญญาณไฟฟ้าหัวใจผิดปกติ ได้แก่ สัญญาณไฟฟ้าหัวใจผิดปกติในลักษณะของ Premature Ventricular Contraction ที่เก็บจากผู้ป่วยจริงจำนวน 10 บล็อก และสัญญาณไฟฟ้าหัวใจผิดปกติในลักษณะของ Atrial Flutter จาก MIT-BIH Arrhythmia Database Record x_203 จำนวน 10 บล็อก

ในขั้นแรกจะทำการทดสอบกับสัญญาณไฟฟ้าหัวใจปกติทั่วไปเพื่อเป็นการพิจารณาหาชนิดของเวฟเล็ตที่เหมาะสมในการใช้งาน และจะทำการทดสอบเพิ่มเติมกับสัญญาณไฟฟ้าหัวใจผิดปกติต่อไป ผลการทดสอบได้แสดงให้เห็นถึงประสิทธิภาพการลดขนาดข้อมูลด้วยค่า PRD และ

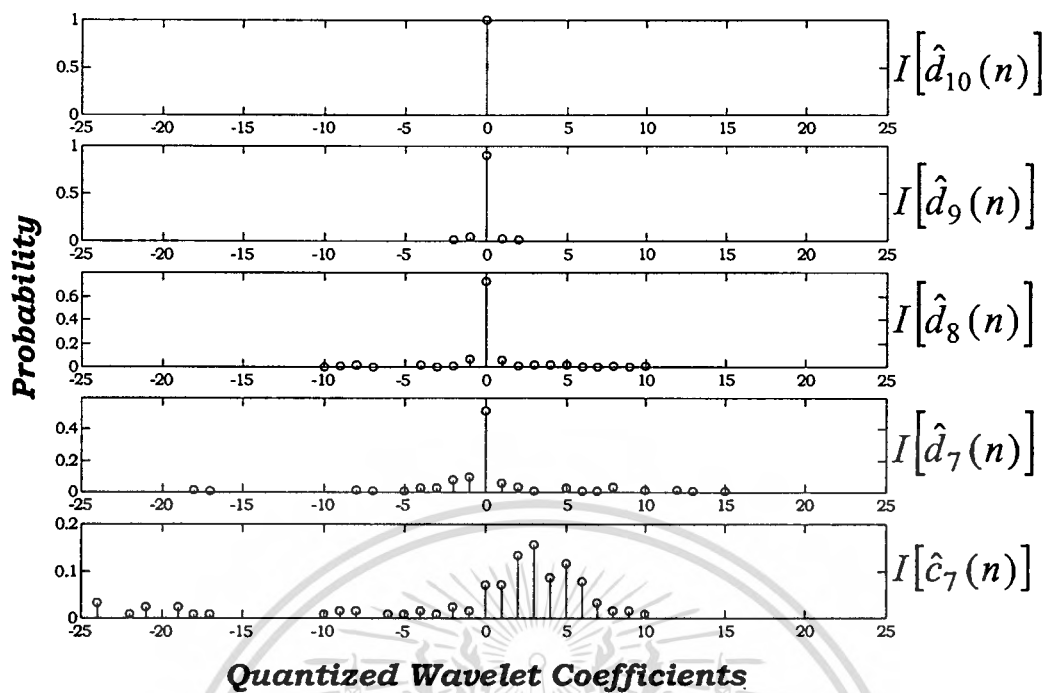
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเปรียบเทียบลักษณะของสัญญาณก่อนและหลังผ่านขบวนการลดขนาดข้อมูลเมื่อกำหนดให้ค่า CR คงที่ค่าต่างๆ ค่า CR ที่แสดงเป็นค่าที่ทำให้สัญญาณไฟฟ้าหัวใจที่ผ่านการลดขนาดข้อมูลมีคุณภาพของสัญญาณที่ตีมากไปจนสามารถสังเกตเห็นความเพี้ยนได้อย่างชัดเจน และยังได้แสดงประสิทธิภาพการลดขนาดข้อมูลเปรียบเทียบกับวิธีการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจแบบอื่นๆอีกด้วย ค่า CR และ PRD ที่แสดงในผลการทดสอบทั้งหมดจะเป็นค่าเฉลี่ยของค่า CR และ PRD ที่คำนวณจากแต่ละบล็อก

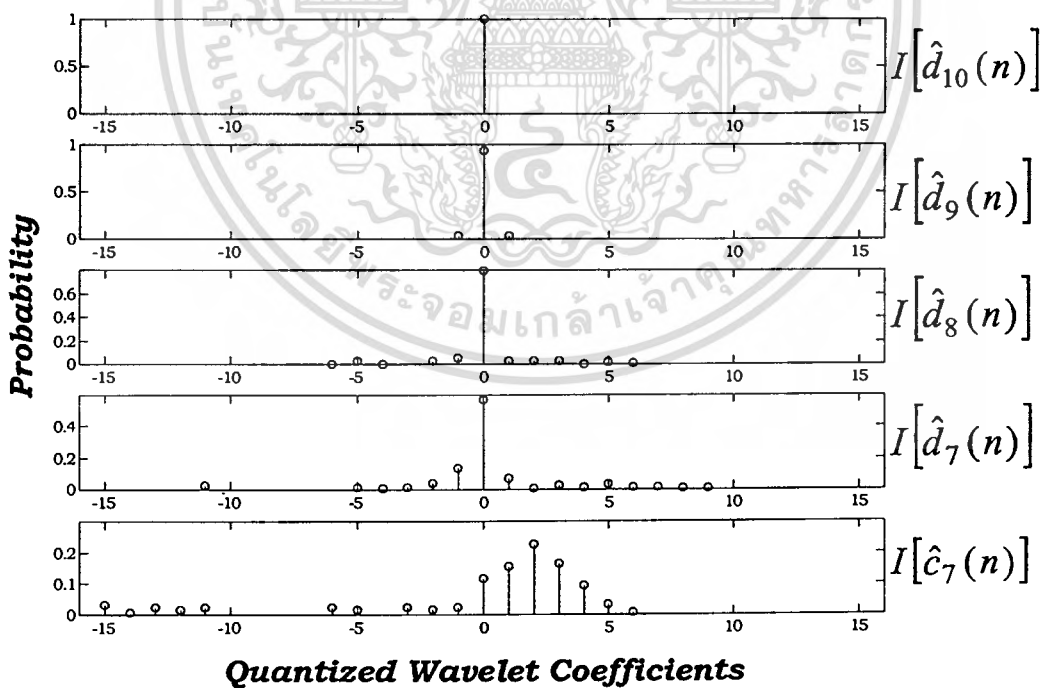
6.2 ผลการทดสอบกับตัวอย่างสัญญาณไฟฟ้าหัวใจปกติทั่วไป

รูปที่ 6.1 ถึงรูปที่ 6.4 แสดงตัวอย่างฮิสโตแกรมของ Quantized Wavelet Coefficients เมื่อค่า CR เป็น 3, 6, 9 และ 12 เท่าตามลำดับ (ใช้เวฟเลตชนิด Coiflet-3 และอันดับการแปลง L เป็น 4) แกนแนวนอนคือค่าจำนวนเต็มที่ได้จากขบวนการแควนไทซ์สัมประสิทธิ์เวฟเลต แกนแนวตั้งคือค่าความน่าจะเป็นในการเกิดข้อมูล ฮิสโตแกรมนี้แสดงให้เห็นว่าส่วนของ $I[\hat{d}_{10}(n)]$ ในรูปที่ 6.1 ส่วนของ $I[\hat{d}_{10}(n)]$ และ $I[\hat{d}_9(n)]$ ในรูปที่ 6.2 ส่วนของ $I[\hat{d}_{10}(n)]$, $I[\hat{d}_9(n)]$ และ $I[\hat{d}_8(n)]$ ในรูปที่ 6.3 และส่วนของ $I[\hat{d}_{10}(n)]$, $I[\hat{d}_9(n)]$ และ $I[\hat{d}_8(n)]$ ในรูปที่ 6.4 ปรากฏค่าศูนย์อยู่เป็นจำนวนมาก ในการทดสอบจะทำการลดจำนวนศูนย์ในกลุ่มของ Quantized Detail Wavelet Coefficients ดังกล่าวด้วยวิธี Run-Length Coding เพื่อเป็นการเพิ่มประสิทธิภาพการลดขนาดข้อมูลก่อนการเข้ารหัสฮัฟแมนต่อไป

รูปที่ 6.5 เป็นการแสดงประสิทธิภาพการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจปกติด้วยค่า PRD เมื่อใช้เวฟเลตตระกูลต่างๆ เมื่อกำหนดให้อันดับการแปลง L เป็น 4 โดยเป็นการแสดงค่า PRD เมื่อค่า CR คงที่ค่าต่างๆ ในการทดสอบจะเลือกใช้ฟังก์ชันพื้นฐานเวฟเลตที่มีค่า NVM พ่ายชื่อหรือความราบเรียบต่ำ ปานกลาง และสูงเท่านั้น โดยตระกูล Daubechies ที่มีความราบเรียบต่ำ ปานกลาง และสูงคือ Daubechies-4, Daubechies-12 และ Daubechies-20 ตามลำดับ ตระกูล Symmlet ที่มีความราบเรียบต่ำ ปานกลาง และสูงคือ Symmlet-4, Symmlet-7 และ Symmlet-10 ตามลำดับ และตระกูล Coiflet ที่มีความราบเรียบต่ำ ปานกลาง และสูงคือ Coiflet-1, Coiflet-3 และ Coiflet-5 ตามลำดับ จากผลการทดสอบในรูปที่ 6.5 จะเห็นว่า เวฟเลตที่มีความราบเรียบปานกลางมีแนวโน้มจะให้ประสิทธิภาพการลดขนาดข้อมูลสูงสุด(เมื่อเทียบกับเวฟเลตที่มีความราบเรียบต่ำและสูงในตระกูลเดียวกัน) ยกเว้นเวฟเลตตระกูล Coiflet ซึ่งจะให้ประสิทธิภาพการลดขนาดข้อมูลที่ใกล้เคียงกันเมื่อใช้เวฟเลตที่มีความราบเรียบปานกลาง(Coiflet-3)และความราบเรียบสูง(Coiflet-5) และเมื่อนำค่าประสิทธิภาพการลดขนาดข้อมูลสูงสุดจากแต่ละตระกูลในรูปที่ 6.5 ซึ่ง

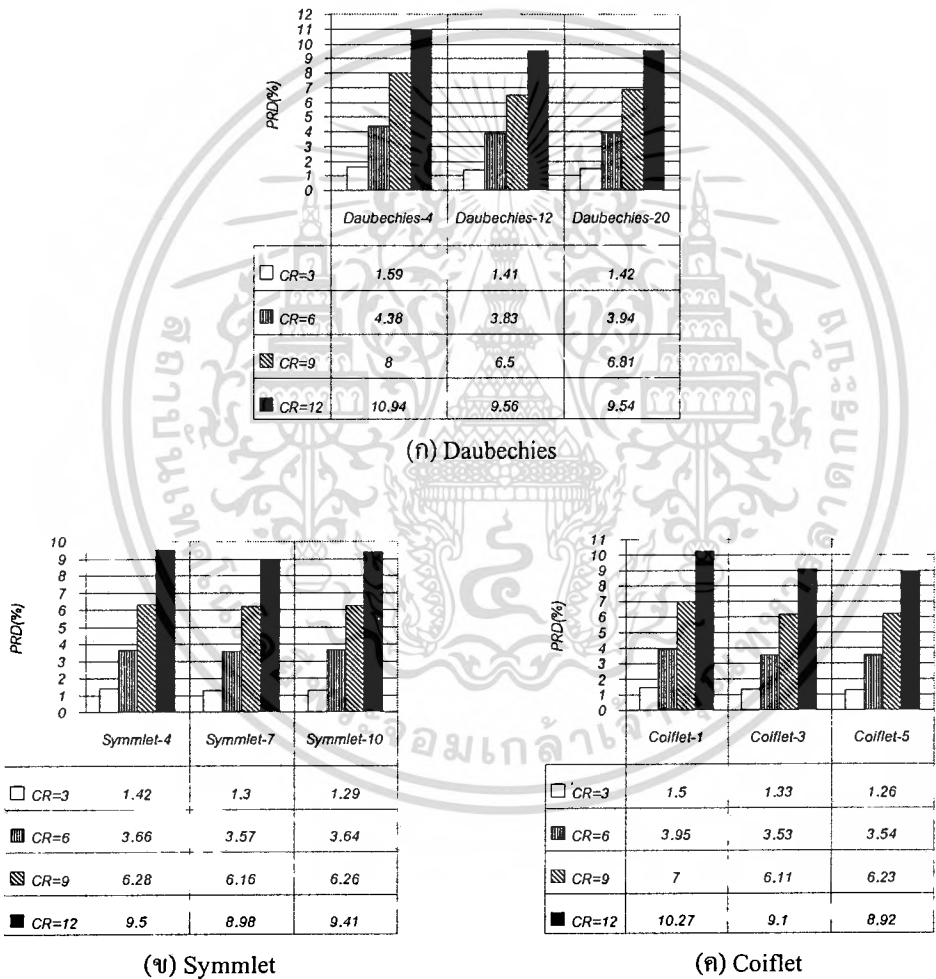


รูปที่ 6.3 แสดงฮิสโตแกรมของ Quantized Wavelet Coefficients เมื่อ $CR = 9$



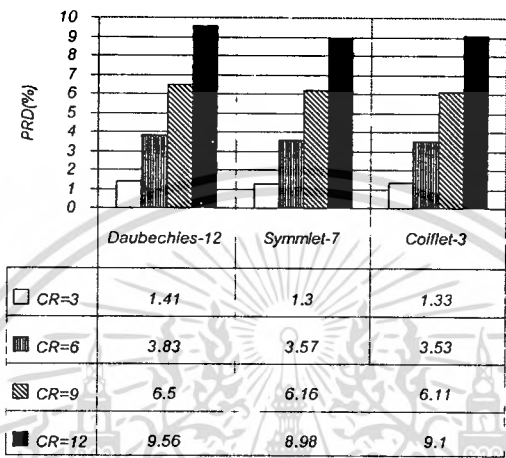
รูปที่ 6.4 แสดงฮิสโตแกรมของ Quantized Wavelet Coefficients เมื่อ $CR = 12$

ได้แก่ Daubechies-12, Symmlet-7 และ Coiflet-3 นำมาทำการเปรียบเทียบกันแสดงในรูปที่ 6.6 จะเห็นว่า เวฟเลตชนิด Daubechies-12 จะให้ประสิทธิภาพการลดขนาดข้อมูลต่ำสุด โดยจะให้ค่า PRD เป็น 1.41%, 3.83%, 6.5% และ 9.56% เมื่อ CR เท่ากับ 3, 6, 9 และ 12 ตามลำดับ และ เวฟเลตชนิด Symmlet-7 และ Coiflet-3 จะให้ประสิทธิภาพการลดขนาดข้อมูลที่ใกล้เคียงกัน โดย เวฟเลตชนิด Symmlet-7 จะให้ค่า PRD เป็น 1.3%, 3.57%, 6.16% และ 8.98% เมื่อ CR เท่ากับ 3, 6, 9 และ 12 ตามลำดับ และเวฟเลตชนิด Coiflet-3 จะให้ค่า PRD เป็น 1.33%, 3.53%, 6.11% และ 9.1% เมื่อ CR เท่ากับ 3, 6, 9 และ 12 ตามลำดับ

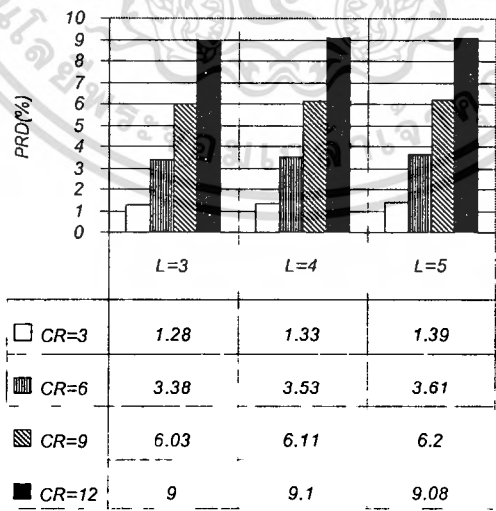


รูปที่ 6.5 แสดงการเปรียบเทียบประสิทธิภาพการลดขนาดข้อมูลด้วยค่า PRD เมื่อใช้เวฟเลตที่มีระดับความราบเรียบต่ำ ปานกลาง และสูงของเวฟเลตตระกูล Daubechies, Symmlet และ Coiflet

จากที่ผ่านมาเป็นการทดสอบประสิทธิภาพการลดขนาดข้อมูลโดยกำหนดให้อันดับการแปลง L เป็น 4 เท่านั้น รูปที่ 6.7 เป็นผลการทดสอบประสิทธิภาพการลดขนาดข้อมูลด้วยค่า PRD เมื่ออันดับการแปลง L เปลี่ยนไป ซึ่งพิจารณาให้อันดับการแปลง L คือ 3, 4 และ 5 โดยกำหนดให้ชนิดของเวฟเล็ตที่ใช้ทดสอบคือ Coiflet-3 จากภาพแสดงให้เห็นว่าเมื่ออันดับการแปลงเพิ่มขึ้นประสิทธิภาพการลดขนาดข้อมูลมีแนวโน้มที่จะลดลงเพียงเล็กน้อยเท่านั้น



รูปที่ 6.6 แสดงการเปรียบเทียบประสิทธิภาพการลดขนาดข้อมูลด้วยค่า PRD ของเวฟเล็ตที่ให้ประสิทธิภาพสูงสุดจากแต่ละตระกูล

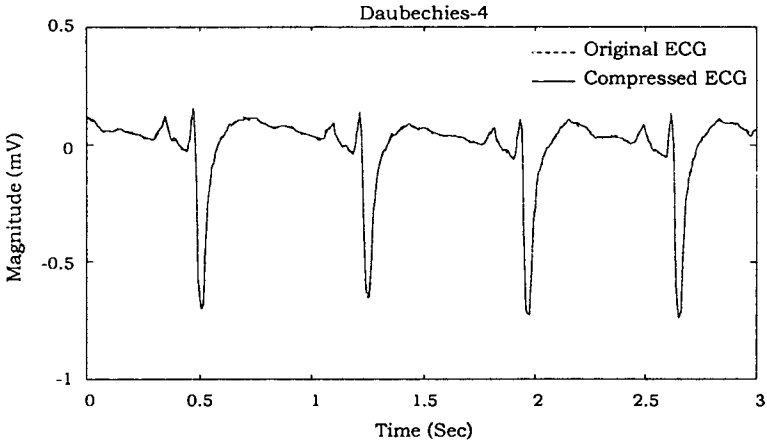


รูปที่ 6.7 แสดงการเปรียบเทียบประสิทธิภาพการลดขนาดข้อมูลด้วยค่า PRD เมื่ออันดับการแปลง L เปลี่ยนไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6.8 ถึงรูปที่ 6.16 เป็นการแสดงสัญญาณไฟฟ้าหัวใจปกติที่ผ่านขบวนการลดขนาดข้อมูลด้วยวิธีการแปลงเวฟเลต โดยรูปที่ 6.8 ถึงรูปที่ 6.10 แสดงสัญญาณที่ผ่านขบวนการลดขนาดข้อมูลที่ $CR = 6$ รูปที่ 6.11 ถึงรูปที่ 6.13 แสดงสัญญาณที่ผ่านขบวนการลดขนาดข้อมูลที่ $CR = 9$ และรูปที่ 6.14 ถึงรูปที่ 6.16 แสดงสัญญาณที่ผ่านขบวนการลดขนาดข้อมูลที่ $CR = 12$ รูปที่ 6.8 รูปที่ 6.11 และรูปที่ 6.14 แสดงสัญญาณที่ผ่านขบวนการลดขนาดข้อมูลโดยใช้เวฟเลตตระกูล Daubechies ที่ความราบเรียบต่ำ (Daubechies-4) ความราบเรียบปานกลาง (Daubechies-12) และความราบเรียบสูง (Daubechies-20) รูปที่ 6.9 รูปที่ 6.12 และรูปที่ 6.15 แสดงสัญญาณที่ผ่านขบวนการลดขนาดข้อมูลโดยใช้เวฟเลตตระกูล Symmlet ที่ความราบเรียบต่ำ (Symmlet-4) ความราบเรียบปานกลาง (Symmlet-7) และความราบเรียบสูง (Symmlet-10) และรูปที่ 6.10 รูปที่ 6.13 และรูปที่ 6.16 แสดงสัญญาณที่ผ่านขบวนการลดขนาดข้อมูลโดยใช้เวฟเลตตระกูล Coiflet ที่ความราบเรียบต่ำ (Coiflet-1) ความราบเรียบปานกลาง (Coiflet-3) และความราบเรียบสูง (Coiflet-5)

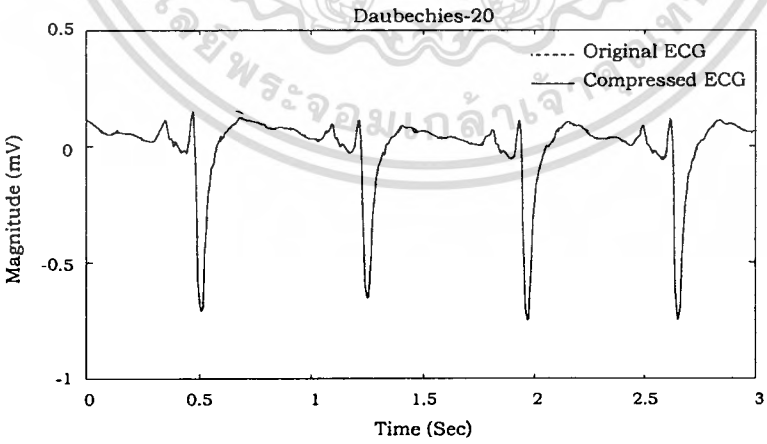
เมื่อพิจารณาสัญญาณที่ผ่านขบวนการลดขนาดข้อมูลด้วยวิธีการแปลงเวฟเลต แสดงให้เห็นว่าที่ $CR = 6$ (รูปที่ 6.8 ถึงรูปที่ 6.10) สัญญาณที่ผ่านการลดขนาดข้อมูลเกิดความเพี้ยนของสัญญาณขึ้นเพียงเล็กน้อยเท่านั้น และผลของการเลือกใช้เวฟเลตที่มีความราบเรียบต่างกันทำให้สัญญาณที่ผ่านการลดขนาดข้อมูลมีความแตกต่างกันไม่มากนัก ส่วนที่ $CR = 9$ (รูปที่ 6.11 ถึงรูปที่ 6.13) สัญญาณที่ผ่านการลดขนาดข้อมูลสามารถสังเกตเห็นความเพี้ยนได้อย่างชัดเจน การเลือกใช้เวฟเลตที่มีความราบเรียบต่ำจะทำให้สัญญาณที่ผ่านการลดขนาดข้อมูลมีลักษณะค่อนข้างคมและไม่ต่อเนื่อง (สังเกตได้อย่างชัดเจนที่สุดในรูปที่ 6.11(ก) ซึ่งใช้เวฟเลตแบบ Daubechies-4) การเลือกใช้เวฟเลตที่มีความราบเรียบสูงจะทำให้สัญญาณที่ผ่านการลดขนาดข้อมูลมีลักษณะของการออสซิลเลท (เห็นได้อย่างชัดเจนในรูปที่ 6.11(ค), รูปที่ 6.12(ค) และรูปที่ 6.13(ค)) และที่ $CR = 12$ (รูปที่ 6.14 ถึงรูปที่ 6.16) สัญญาณที่ผ่านการลดขนาดข้อมูลเกิดความเพี้ยนที่สูงมากสำหรับทุกตระกูลของเวฟเลตที่ทำการทดสอบ อย่างไรก็ตามการที่จะกำหนดค่า CR และชนิดของเวฟเลตที่เหมาะสมในใช้งานนั้น จำเป็นจะต้องให้แพทย์เป็นผู้พิจารณา ซึ่งจะกล่าวถึงต่อไป



(ก) ใช้เวฟเลตความราบเรียบต่ำ(Daubechies-4) ให้ค่า PRD = 4.38%



(ข) ใช้เวฟเลตความราบเรียบปานกลาง(Daubechies-12) ให้ค่า PRD = 3.83%

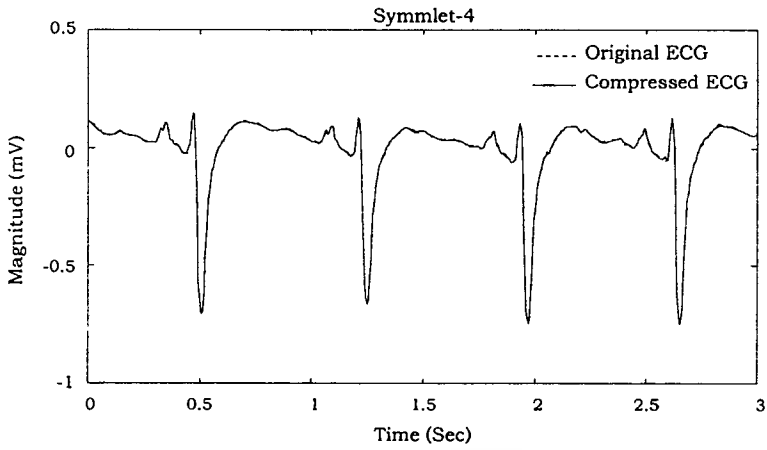


(ค) ใช้เวฟเลตความราบเรียบสูง(Daubechies-20) ให้ค่า PRD = 3.94%

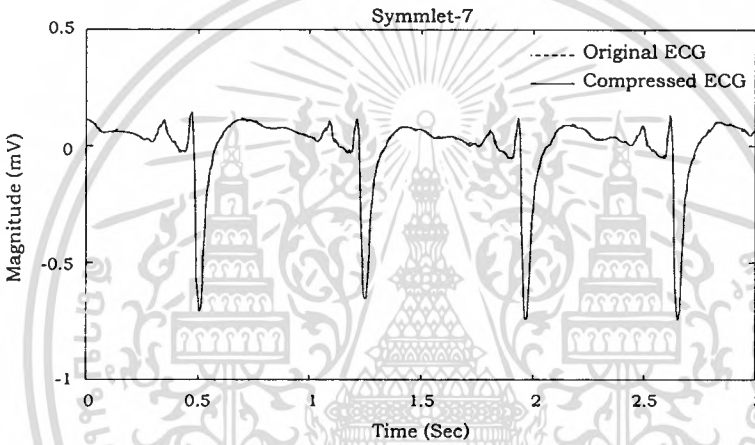
รูปที่ 6.8 แสดงสัญญาณไฟฟ้าหัวใจปกติเมื่อผ่านขบวนการลดขนาดข้อมูลโดยใช้เวฟเลตตระกูล

Daubechies ที่ CR = 6

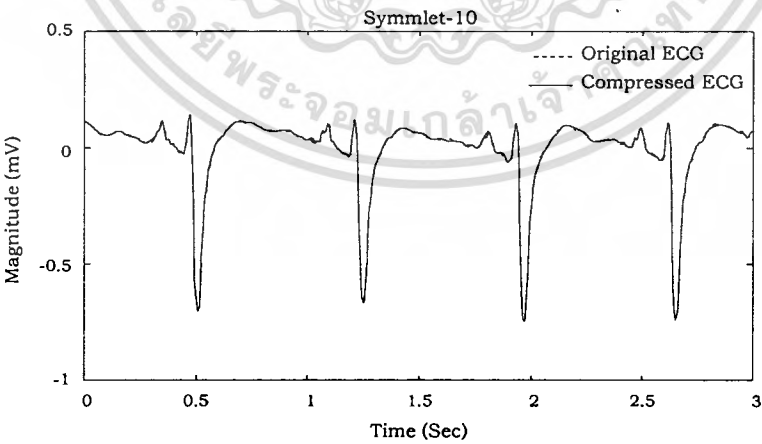
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) ใช้เวฟเล็ดความราบเรียบต่ำ(Symmlet-4) ให้ค่า PRD = 3.66%



(ข) ใช้เวฟเล็ดความราบเรียบปานกลาง(Symmlet-7) ให้ค่า PRD = 3.57%

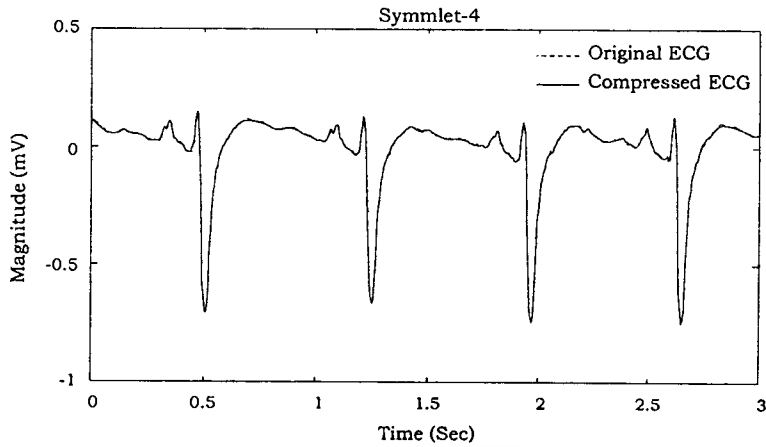


(ค) ใช้เวฟเล็ดความราบเรียบสูง(Symmlet-10) ให้ค่า PRD = 3.64%

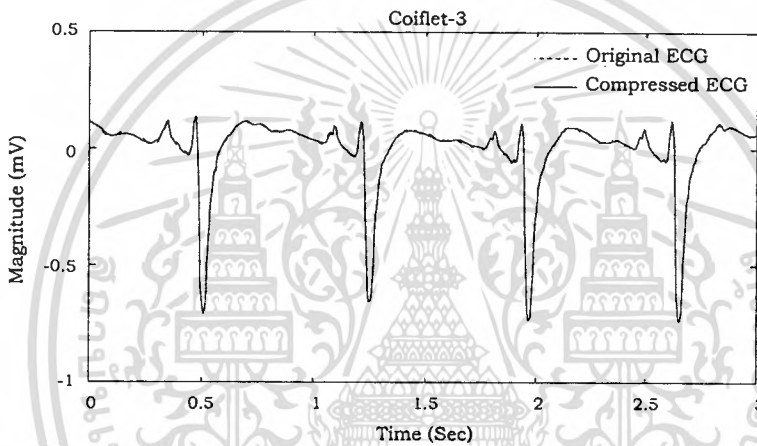
รูปที่ 6.9 แสดงสัญญาณไฟฟ้าหัวใจปกติเมื่อผ่านขบวนการลดขนาดข้อมูลโดยใช้เวฟเล็ดตระกูล

Symmlet ที่ $CR = 6$

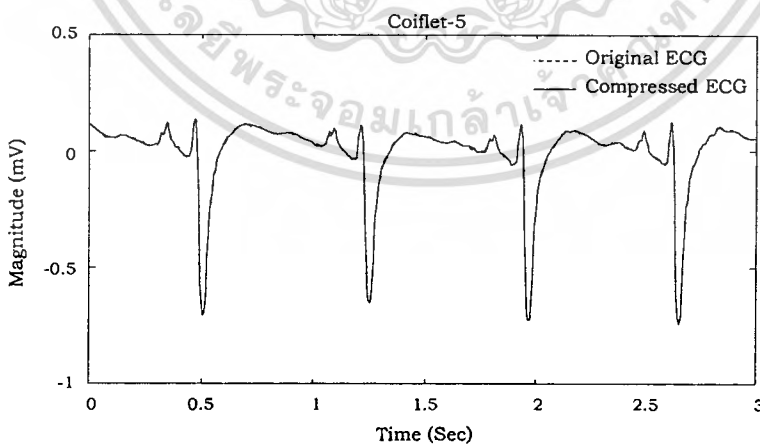
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) ใช้เวฟเล็ดความราบเรียบต่ำ(Coiflet-1) ให้ค่า PRD = 3.95%



(ข) ใช้เวฟเล็ดความราบเรียบปานกลาง(Coiflet-3) ให้ค่า PRD = 3.53%

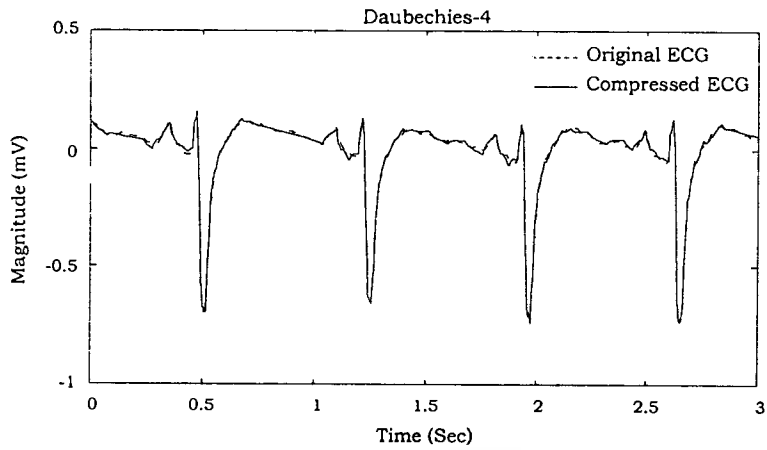


(ค) ใช้เวฟเล็ดความราบเรียบสูง(Coiflet-5) ให้ค่า PRD = 3.54%

รูปที่ 6.10 แสดงสัญญาณไฟฟ้าหัวใจปกติเมื่อผ่านขบวนการลดขนาดข้อมูลโดยใช้เวฟเล็ดตระกูล

Coiflet ที่ $CR = 6$

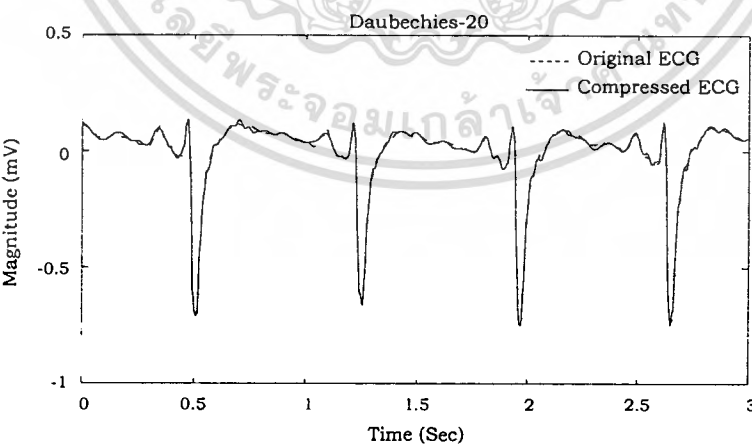
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) ใช้เวฟเล็ดความราบเรียบต่ำ(Daubechies-4) ให้ค่า PRD = 8%



(ข) ใช้เวฟเล็ดความราบเรียบปานกลาง(Daubechies-12) ให้ค่า PRD = 6.5%

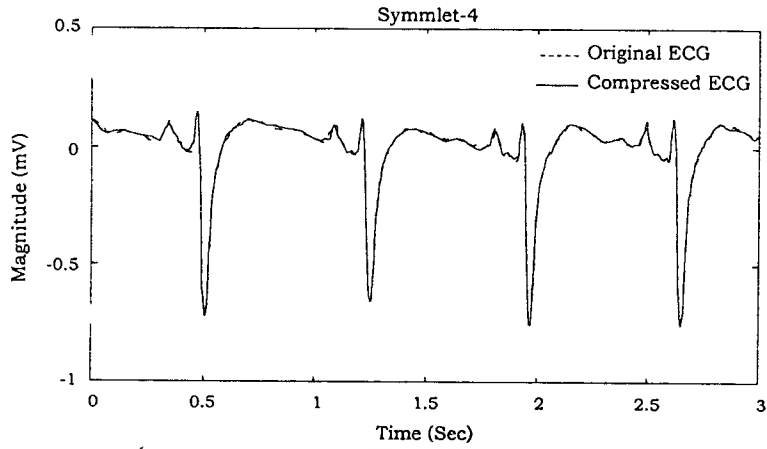


(ค) ใช้เวฟเล็ดความราบเรียบสูง(Daubechies-20) ให้ค่า PRD = 6.81%

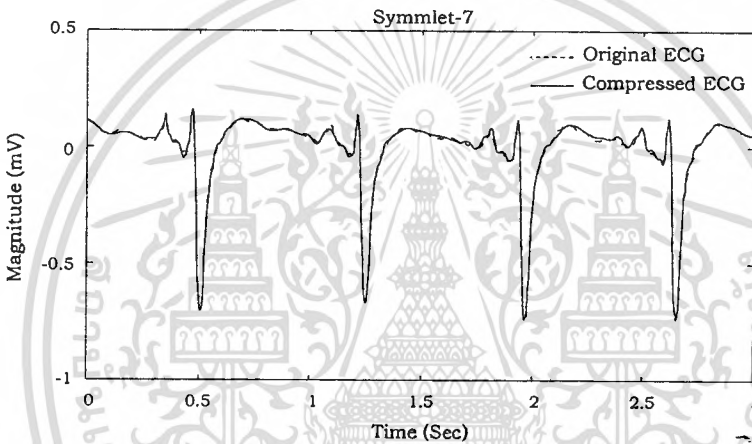
รูปที่ 6.11 แสดงสัญญาณไฟฟ้าหัวใจปกติเมื่อผ่านขบวนการลดขนาดข้อมูลโดยใช้เวฟเล็ดตระกูล

Daubechies ที่ CR = 9

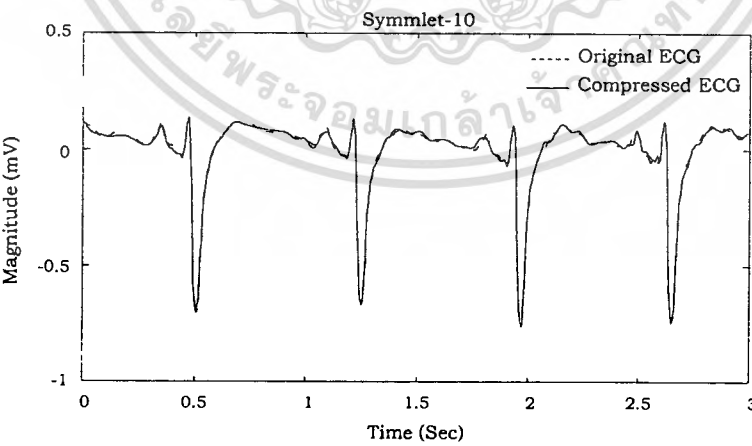
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) ใช้เวฟเล็ดความราบเรียบต่ำ(Symmlet-4) ให้ค่า PRD = 6.28%



(ข) ใช้เวฟเล็ดความราบเรียบปานกลาง(Symmlet-7) ให้ค่า PRD = 6.16%

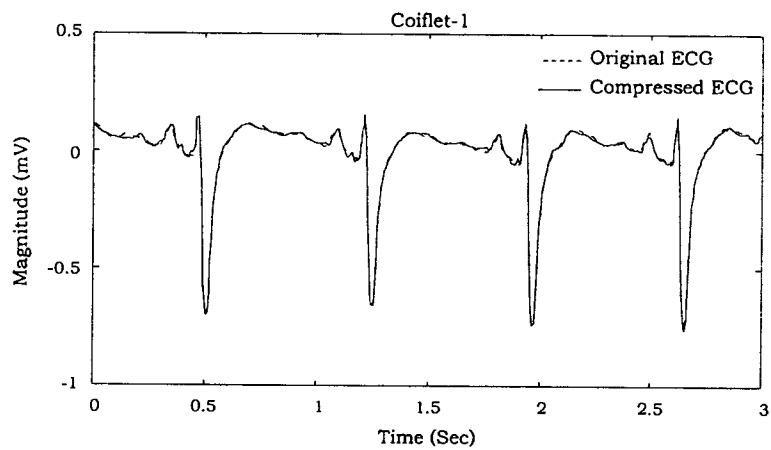


(ค) ใช้เวฟเล็ดความราบเรียบสูง(Symmlet-10) ให้ค่า PRD = 6.26%

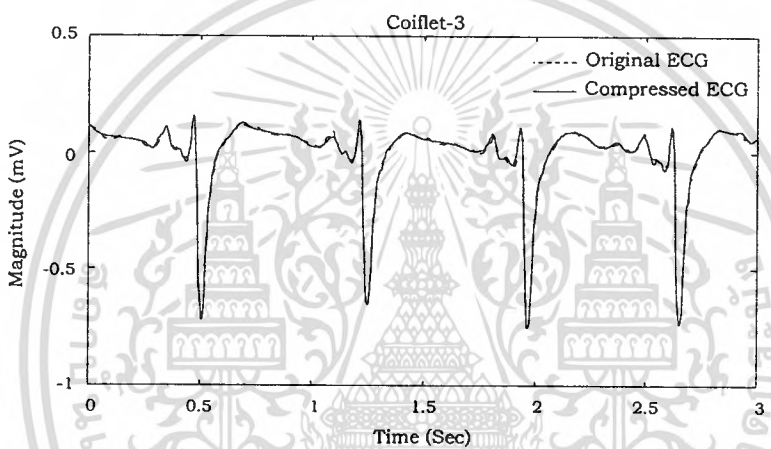
รูปที่ 6.12 แสดงสัญญาณไฟฟ้าหัวใจปกติเมื่อผ่านขบวนการลดขนาดข้อมูลโดยใช้เวฟเล็ดตระกูล

Symmlet ที่ CR = 9

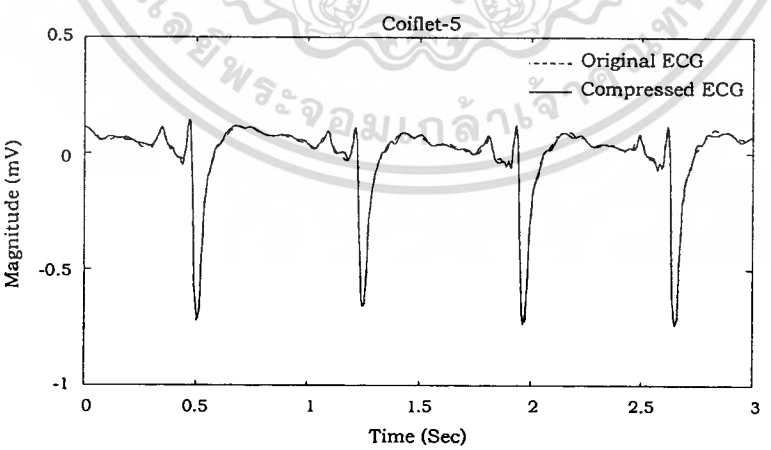
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) ใช้เวฟเล็ทความราบเรียบต่ำ(Coiflet-1) ให้ค่า PRD = 7%



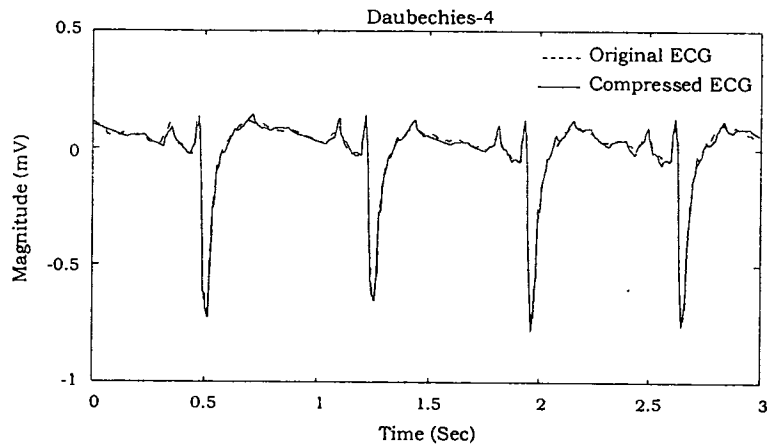
(ข) ใช้เวฟเล็ทความราบเรียบปานกลาง(Coiflet-3) ให้ค่า PRD = 6.11%



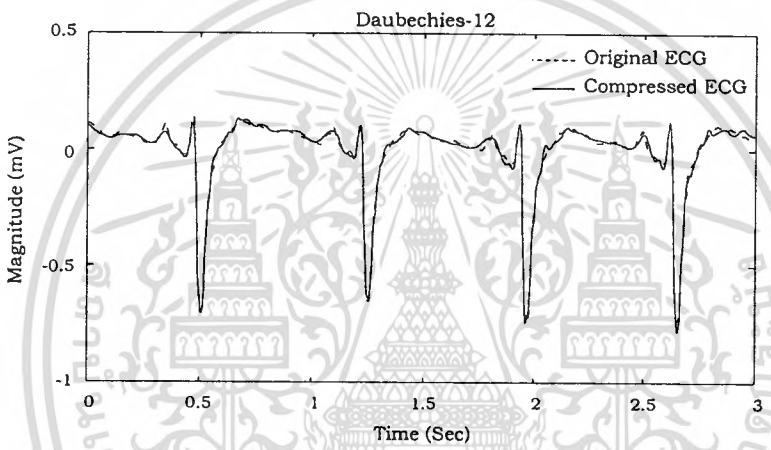
(ค) ใช้เวฟเล็ทความราบเรียบสูง(Coiflet-5) ให้ค่า PRD = 6.23%

รูปที่ 6.13 แสดงสัญญาณไฟฟ้าหัวใจปกติเมื่อผ่านขบวนการลดขนาดข้อมูลโดยใช้เวฟเล็ทตระกูล Coiflet ที่ CR = 9

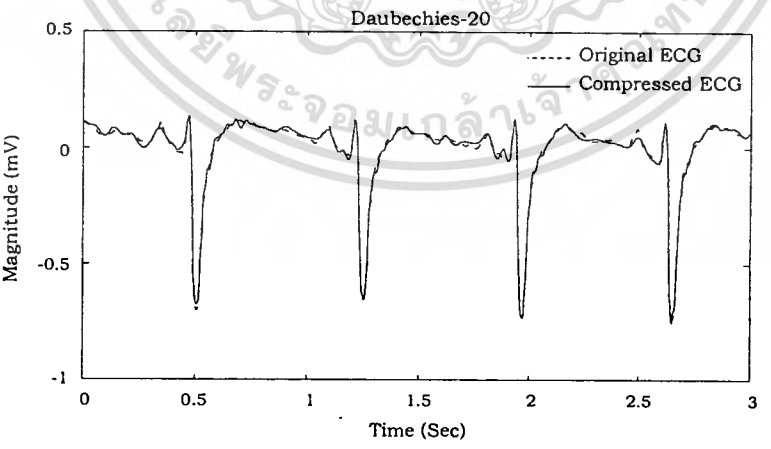
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) ใช้เวฟเลตความราบเรียบต่ำ(Daubechies-4) ให้ค่า PRD = 10.94%



(ข) ใช้เวฟเลตความราบเรียบปานกลาง(Daubechies-12) ให้ค่า PRD = 9.56%

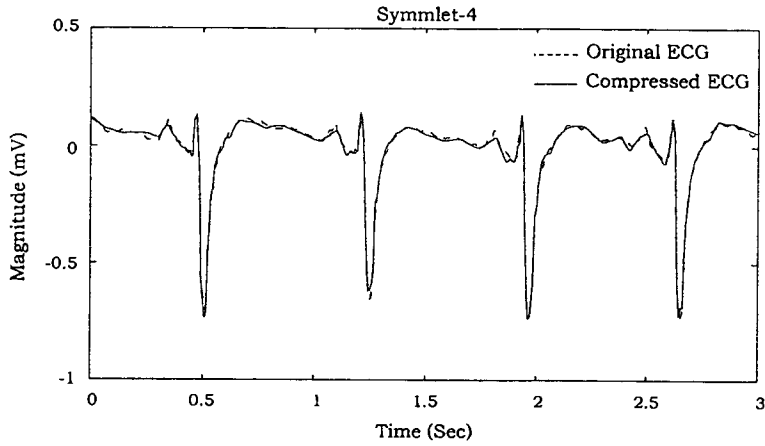


(ค) ใช้เวฟเลตความราบเรียบสูง(Daubechies-20) ให้ค่า PRD = 9.54%

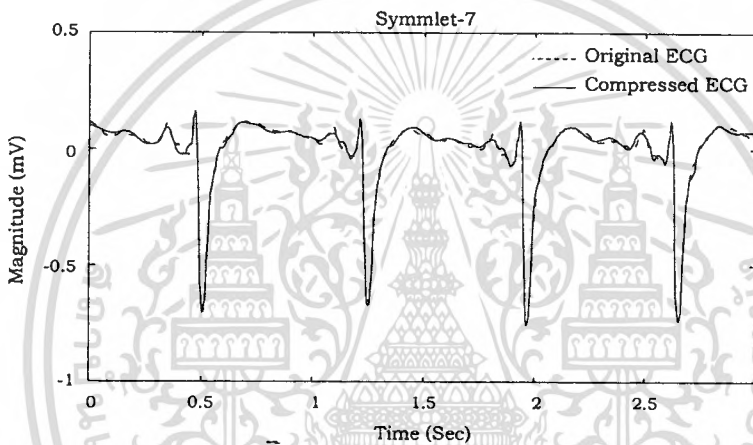
รูปที่ 6.14 แสดงสัญญาณไฟฟ้าหัวใจปกติเมื่อผ่านขบวนการลดขนาดข้อมูลโดยใช้เวฟเลตตระกูล

Daubechies ที่ CR = 12

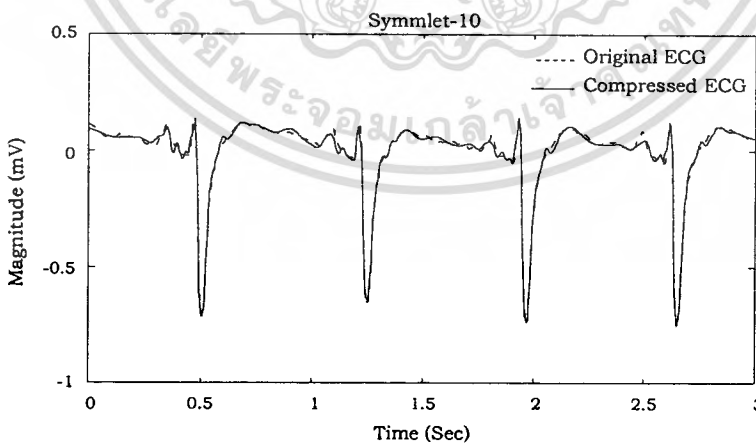
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) ใช้เวฟเล็ดความราบเรียบต่ำ(Symmlet-4) ให้ค่า PRD = 9.5%



(ข) ใช้เวฟเล็ดความราบเรียบปานกลาง(Symmlet-7) ให้ค่า PRD = 8.98%

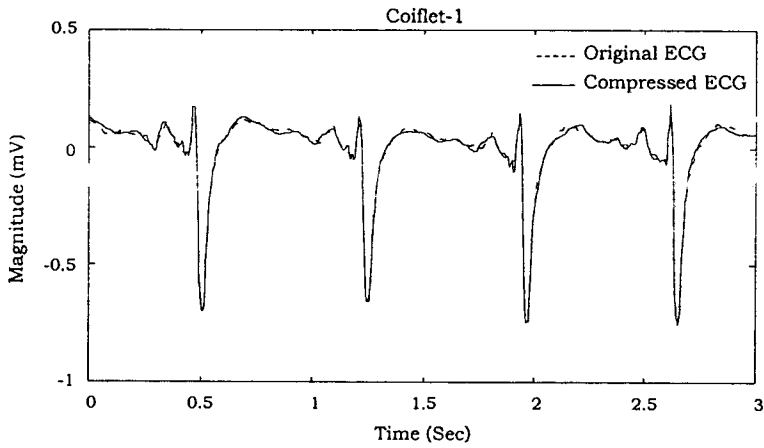


(ค) ใช้เวฟเล็ดความราบเรียบสูง(Symmlet-10) ให้ค่า PRD = 9.41%

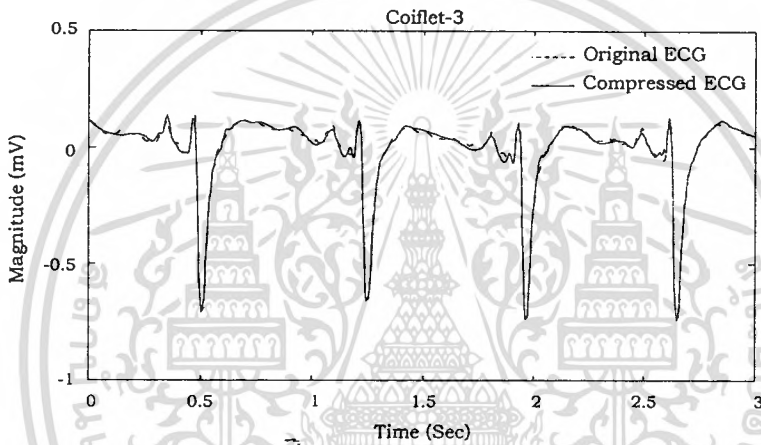
รูปที่ 6.15 แสดงสัญญาณไฟฟ้าหัวใจปกติเมื่อผ่านขบวนการลดขนาดข้อมูลโดยใช้เวฟเล็ดตระกูล

Symmlet ที่ CR = 12

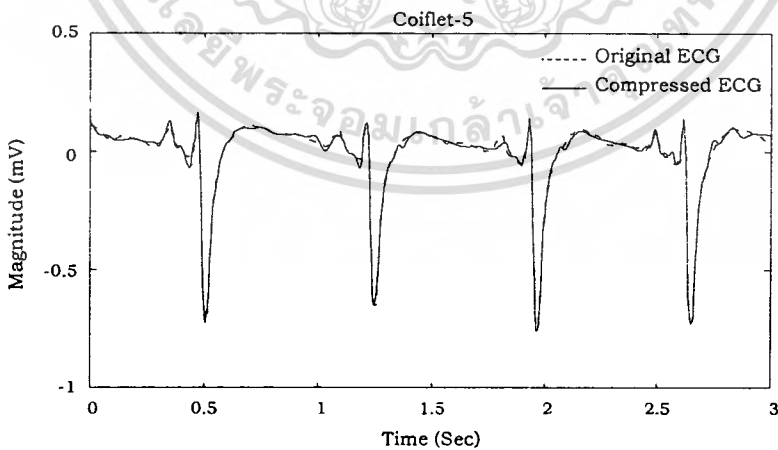
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) ใช้เวฟเล็ดความราบเรียบต่ำ(Coiflet-1) ให้ค่า PRD = 10.27%



(ข) ใช้เวฟเล็ดความราบเรียบปานกลาง(Coiflet-3) ให้ค่า PRD = 9.1%



(ค) ใช้เวฟเล็ดความราบเรียบสูง(Coiflet-5) ให้ค่า PRD = 8.92%

รูปที่ 6.16 แสดงสัญญาณไฟฟ้าหัวใจปกติเมื่อผ่านขบวนการลดขนาดข้อมูลโดยใช้เวฟเล็ดตระกูล

Coiflet ที่ CR = 12

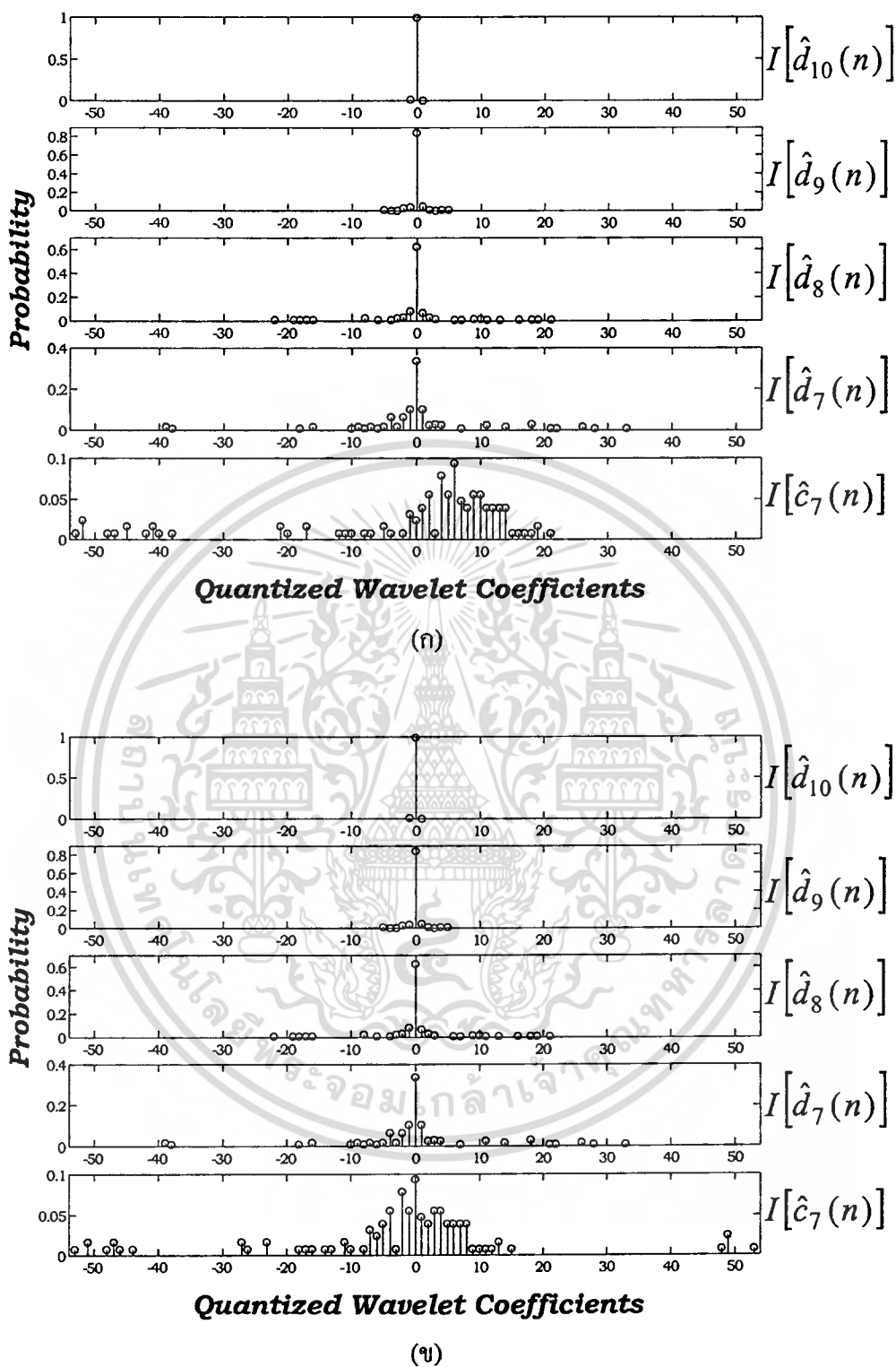
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการพิจารณาตัวอย่างฮิสโตแกรมที่ผ่านมา ซึ่งได้นำมาแสดงไว้ในรูปที่ 6.17(ก) จะสังเกตเห็นว่าส่วนของ Quantized Coarse Wavelet Coefficients $I[\hat{c}_7(n)]$ มีความน่าจะเป็นสูงสุดอยู่ทางด้านซ้ายของศูนย์ ซึ่งสามารถเพิ่มประสิทธิภาพก่อนการเข้ารหัสฮัฟแมนได้โดยวิธี Histogram Circular Shifting หรือการเลื่อนฮิสโตแกรมแบบวนไปทางซ้ายโดยให้ตำแหน่งที่มีความน่าจะเป็นสูงสุดอยู่ที่ศูนย์ แสดงในรูป 6.17(ข) ทั้งนี้เพื่อเป็นการเพิ่มจำนวนข้อมูลที่กระจายตัวอยู่รอบๆ ศูนย์ในส่วนของ Quantized Detail Wavelet Coefficients $I[\hat{d}_j(n)]_{j=10,9,8,7}$ ทำให้ความยาวของรหัสที่ได้จากวิธีของฮัฟแมนลดลง มีผลให้มีค่า CR มากขึ้นในขณะที่ค่า PRD เท่าเดิม จากการทดลองโดยใช้เวฟเลต Coiflet-3 สามารถเพิ่มค่า CR จาก 3, 6, 9 และ 12 ไปเป็น 3.08, 6.08, 9.05 และ 12.03 ตามลำดับ จากการทดสอบจะเห็นได้ว่าขบวนการ Histogram Circular Shifting ให้ผลการทดสอบไม่เป็นที่น่าพอใจ เนื่องจากสามารถเพิ่มประสิทธิภาพการลดขนาดข้อมูลได้เพียงเล็กน้อยเท่านั้น

เพื่อให้เห็นถึงประสิทธิภาพการลดขนาดข้อมูลของวิธีการแปลงเวฟเลตเต็มหน่วย (Discrete Wavelet Transform : DWT) ที่นำเสนอ ในการทดสอบจึงได้แสดงประสิทธิภาพการลดขนาดข้อมูลเปรียบเทียบกับวิธีการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจแบบอื่นๆ ได้แก่ วิธี Fan ซึ่งเป็นวิธีลดขนาดข้อมูลด้วยวิธีทางตรง (Direct Method) ที่มีประสิทธิภาพสูงสุดในกลุ่มของ Significant-Point-Extraction และวิธีการแปลงโคไซน์เต็มหน่วย (Discrete Cosine Transform : DCT) ซึ่งเป็นวิธีลดขนาดข้อมูลด้วยวิธีการแปลง (Transformation Method) ที่มีประสิทธิภาพสูงและเป็นที่นิยมใช้ ในการทดสอบจะกำหนดให้วิธี DWT ใช้เวฟเลตชนิด Coiflet-3 และอันดับการแปลง L เป็น 4

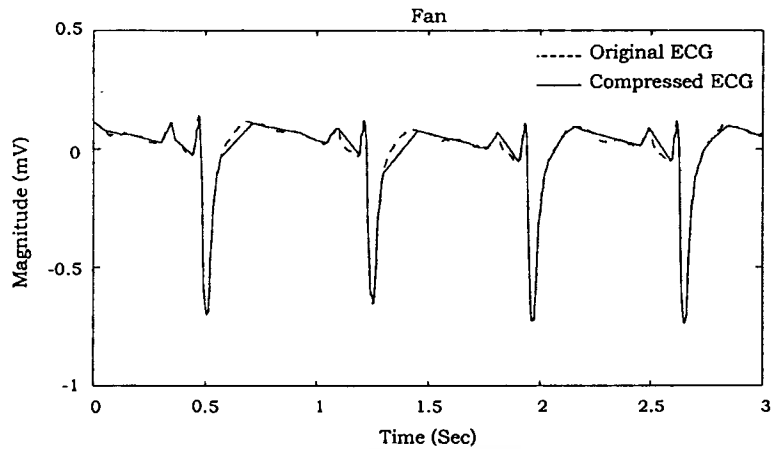
6.2.1 ผลการเปรียบเทียบประสิทธิภาพการลดขนาดข้อมูลระหว่างวิธี DWT และวิธี Fan

รูปที่ 6.18 แสดงการเปรียบเทียบประสิทธิภาพการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจปกติทั่วไปด้วยวิธี DWT และวิธี Fan เมื่อกำหนดให้ $CR = 4$ จากค่า PRD และสัญญาณที่แสดงจะเห็นได้ว่าที่ CR เท่ากัน วิธี DWT จะให้ประสิทธิภาพการลดขนาดข้อมูลที่สูงกว่าวิธี Fan อย่างเห็นได้ชัด

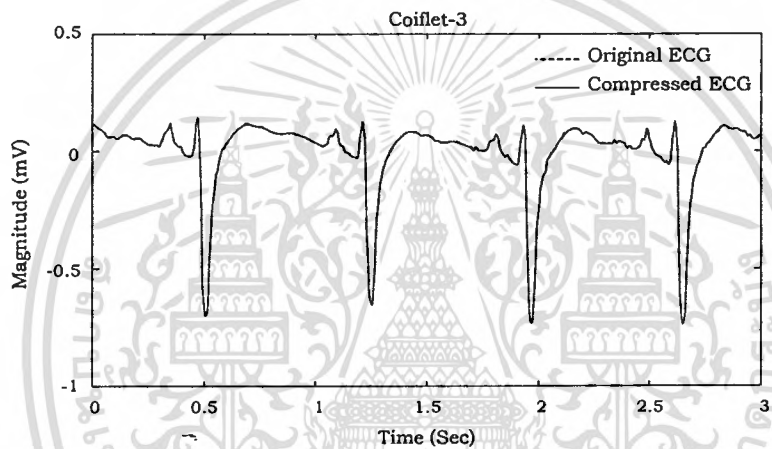


รูปที่ 6.17 แสดงตัวอย่างฮิสโตแกรมของ Quantized Wavelet Coefficients ก่อนผ่านขบวนการ Histogram Circular Shifting (รูป ก) และหลังผ่านขบวนการ Histogram Circular Shifting (รูป ข)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) วิธี Fan ให้ค่า PRD = 13.96%

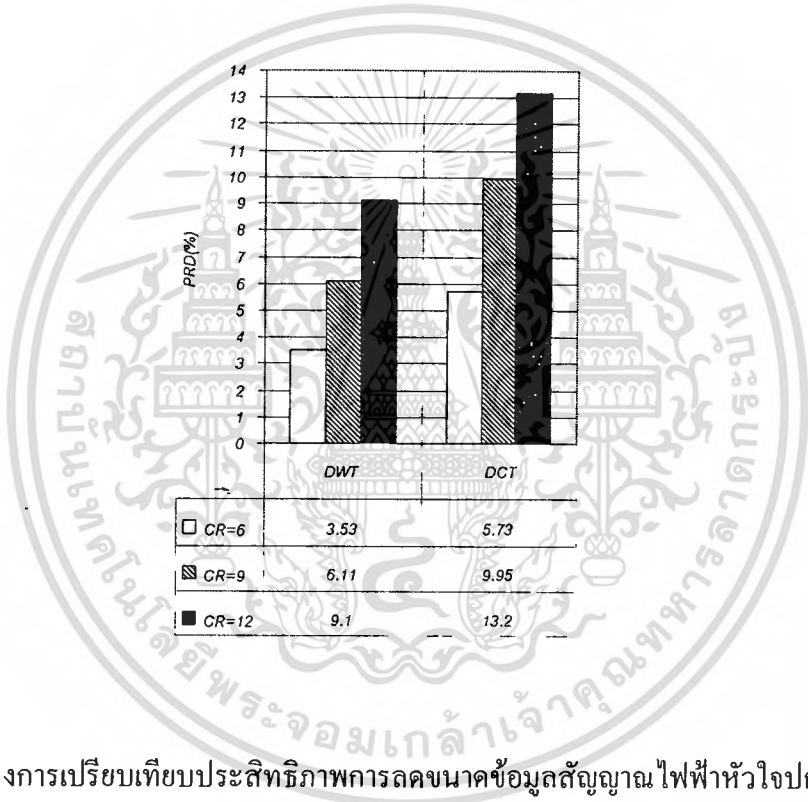


(ข) วิธี DWT ให้ค่า PRD = 2.06%

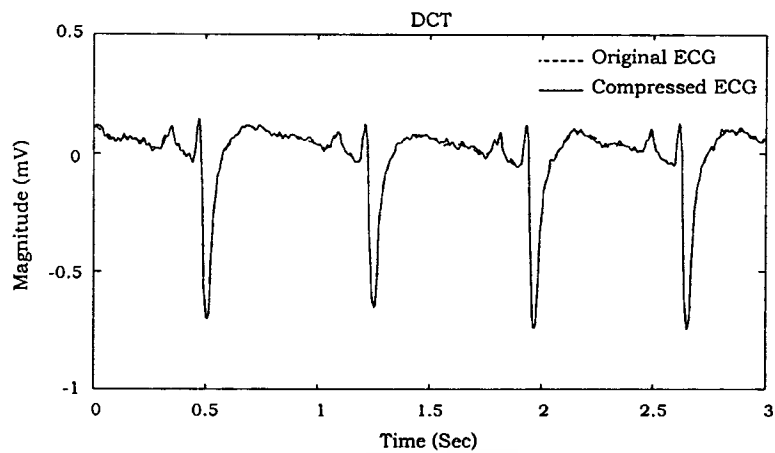
รูปที่ 6.18 แสดงสัญญาณไฟฟ้าหัวใจปกติเมื่อผ่านขบวนการลดขนาดข้อมูลด้วยวิธี Fan และวิธี DWT เมื่อกำหนดให้ CR = 4

6.2.2 ผลการเปรียบเทียบประสิทธิภาพการลดขนาดข้อมูลระหว่างวิธี DWT และวิธี DCT

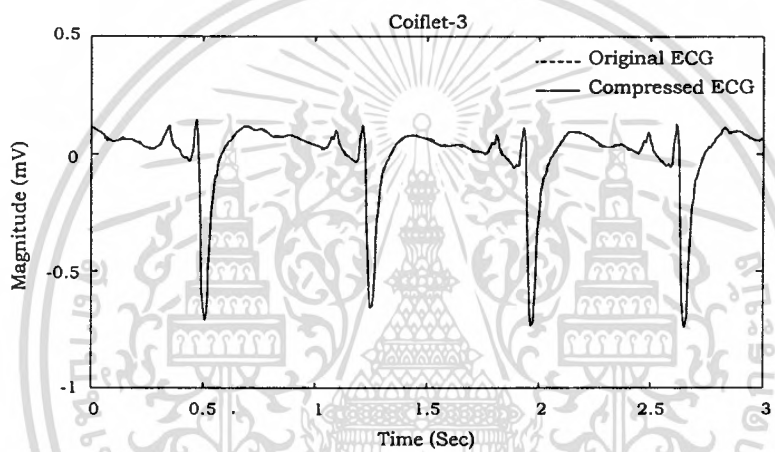
รูปที่ 6.19 ถึงรูปที่ 6.22 แสดงการเปรียบเทียบประสิทธิภาพการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจปกติด้วยวิธี DWT และวิธี DCT เมื่อกำหนดให้ค่า CR เป็น 6, 9 และ 12 จากค่า PRD และสัญญาณที่แสดงจะเห็นได้ว่าที่ CR เท่ากัน วิธี DWT จะให้ประสิทธิภาพการลดขนาดข้อมูลที่สูงกว่าวิธี DCT ทุกค่าของ CR วิธี DCT จะทำให้เกิดความเพี้ยนตลอดแกนเวลา(Globalize Error) ในขณะที่วิธี DWT จะทำให้เกิดความเพี้ยนมากเพียงบางช่วง(Localize Error)เท่านั้น ซึ่งทำให้สามารถรักษาสาระสำคัญของสัญญาณได้ดีกว่าโดยเฉพาะเมื่อค่า CR เป็น 9 และ 12



รูปที่ 6.19 แสดงการเปรียบเทียบประสิทธิภาพการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจปกติด้วยค่า PRD ระหว่างวิธี DWT และวิธี DCT

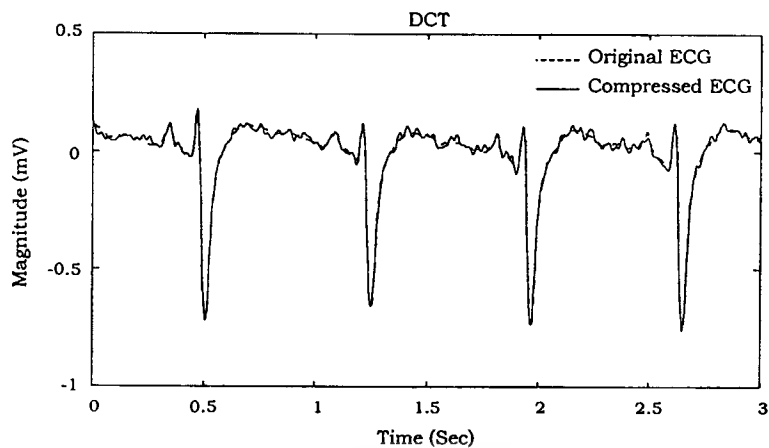


(ก) วิธี DCT ให้ค่า PRD = 5.73%

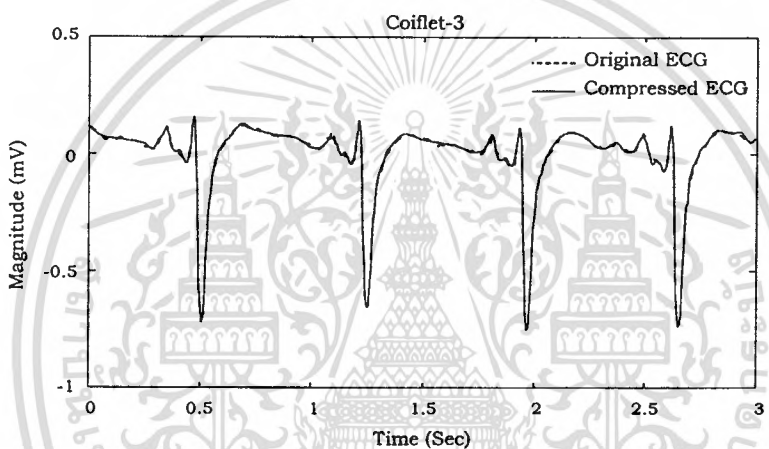


(ข) วิธี DWT ให้ค่า PRD = 3.53%

รูปที่ 6.20 แสดงสัญญาณไฟฟ้าหัวใจปกติที่ผ่านขบวนการลดขนาดข้อมูลด้วยวิธี DCT และวิธี DWT เมื่อกำหนดให้ CR = 6

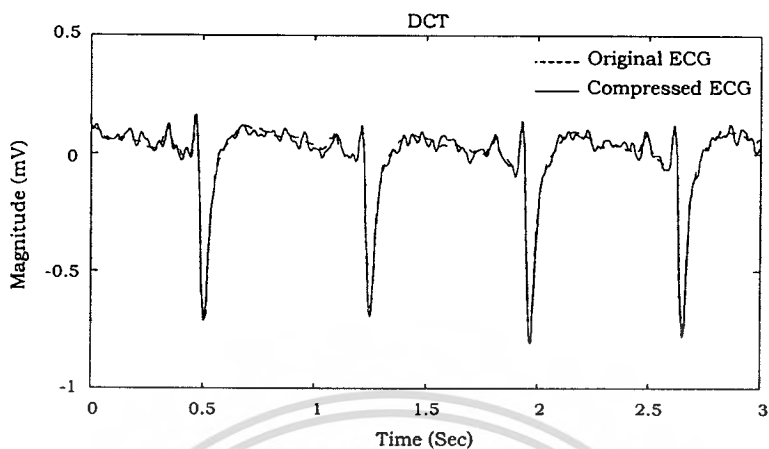


(ก) วิธี DCT ให้ค่า PRD = 9.95%

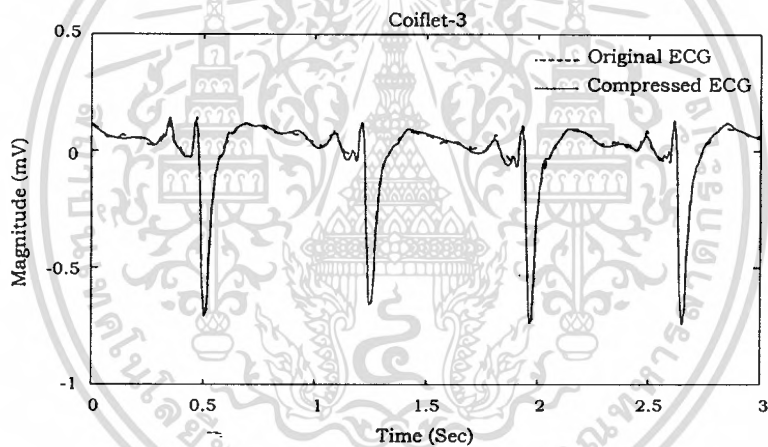


(ข) วิธี DWT ให้ค่า PRD = 6.11%

รูปที่ 6.21 แสดงสัญญาณไฟฟ้าหัวใจปกติที่ผ่านขบวนการลดขนาดข้อมูลด้วยวิธี DCT และวิธี DWT เมื่อกำหนดให้ CR = 9



(ก) วิธี DCT ให้ค่า PRD = 13.2%



(ข) วิธี DWT ให้ค่า PRD = 9.1%

รูปที่ 6.22 แสดงสัญญาณไฟฟ้าหัวใจปกติที่ผ่านขบวนการลดขนาดข้อมูลด้วยวิธี DCT และวิธี DWT เมื่อกำหนดให้ CR = 12

จากตารางที่ 6.1 ถึงตารางที่ 6.3 (เป็นข้อสรุปจากแบบสอบถามความคิดเห็นจากแพทย์
ข้อที่ 7 ในภาคผนวก ง) แสดงผลการตรวจสอบสัญญาณไฟฟ้าหัวใจปกติที่ผ่านขบวนการลดขนาด
ข้อมูลโดยใช้เวฟเลตชนิดต่างๆ(จากรูปที่ 6.8 ถึงรูปที่ 6.16)จากแพทย์ทั้งหมด 5 ท่าน เมื่อ CR เท่า
กับ 6, 9 และ 12 ตามลำดับ จากตารางแสดงให้เห็นว่า ที่ CR = 6 เป็นเงื่อนไขที่เหมาะสมในการนำ
ไปใช้งานมากที่สุดในงานวิจัย เนื่องจากผลการตรวจสอบจากแพทย์ทุกท่านมีความเห็นที่เหมือนกัน
ทุกประการ(ดูตารางที่ 6.1) กล่าวคือ ที่ CR = 6 สำหรับเวฟเลตทุกตระกูลและทุกความราบเรียบใน
การทดสอบ เป็นเงื่อนไขที่ทำให้สัญญาณไฟฟ้าหัวใจที่ผ่านขบวนการลดขนาดข้อมูลเกิดความเพี้ยน
ในเกณฑ์ที่แพทย์ยอมรับได้ทั้งหมด

ตารางที่ 6.1 แสดงผลการตรวจสอบสัญญาณไฟฟ้าหัวใจปกติที่ผ่านการลดขนาดข้อมูลโดยใช้
เวฟเลตชนิดต่างๆจากแพทย์ทั้งหมด 5 ท่าน เมื่อ CR = 6

แพทย์ ท่านที่	Daubechies			Symmlet			Coiflet			วิธีการของแพทย์ ผู้พิจารณา
	D4	D12	D20	S4	S7	S10	C1	C3	C5	
1	✓	✓	✓	✓	✓	✓	✓	✓	✓	ตรวจสอบด้วยสายตา
2	✓	✓	✓	✓	✓	✓	✓	✓	✓	ตรวจสอบด้วยสายตา
3	✓	✓	✓	✓	✓	✓	✓	✓	✓	ตรวจสอบด้วยสายตา
4	✓	✓	✓	✓	✓	✓	✓	✓	✓	ตรวจสอบด้วยสายตา
5	✓	✓	✓	✓	✓	✓	✓	✓	✓	PRD ไม่เกิน 5%

หมายเหตุ ✓ หมายถึงสัญญาณที่ผ่านการลดขนาดข้อมูลเกิดความเพี้ยนในเกณฑ์ที่แพทย์ยอมรับได้
X หมายถึงสัญญาณที่ผ่านการลดขนาดข้อมูลเกิดความเพี้ยนในเกณฑ์ที่แพทย์ไม่สามารถยอมรับได้

ตารางที่ 6.2 แสดงผลการตรวจสอบสัญญาณไฟฟ้าหัวใจปกติที่ผ่านการลดขนาดข้อมูลโดยใช้
เวฟเลตชนิดต่างๆจากแพทย์ทั้งหมด 5 ท่าน เมื่อ CR = 9

แพทย์ ท่านที่	Daubechies			Symmlet			Coiflet			วิธีการของแพทย์ ผู้พิจารณา
	D4	D12	D20	S4	S7	S10	C1	C3	C5	
1	✓	✓	✓	✓	✓	✓	X	✓	✓	ตรวจสอบด้วยสายตา
2	✓	✓	✓	✓	✓	✓	✓	✓	✓	ตรวจสอบด้วยสายตา
3	X	X	X	X	X	X	X	X	X	ตรวจสอบด้วยสายตา
4	X	X	X	X	X	X	X	X	X	ตรวจสอบด้วยสายตา
5	X	X	X	X	X	X	X	X	X	PRD ไม่เกิน 5%

หมายเหตุ ✓ หมายถึงสัญญาณที่ผ่านการลดขนาดข้อมูลเกิดความเพี้ยนในเกณฑ์ที่แพทย์ยอมรับได้
X หมายถึงสัญญาณที่ผ่านการลดขนาดข้อมูลเกิดความเพี้ยนในเกณฑ์ที่แพทย์ไม่สามารถยอมรับได้

ตารางที่ 6.3 แสดงผลการตรวจสอบสัญญาณไฟฟ้าหัวใจปกติที่ผ่านการลดขนาดข้อมูลโดยใช้
เวฟเล็ทชนิดต่างๆจากแพทย์ทั้งหมด 5 ท่าน เมื่อ CR = 12

แพทย์ ท่านที่	Daubechies			Symmlet			Coiflet			วิธีการของแพทย์ ผู้พิจารณา
	D4	D12	D20	S4	S7	S10	C1	C3	C5	
1	x	x	x	x	x	x	x	x	x	ตรวจสอบด้วยสายตา
2	✓	✓	✓	✓	✓	✓	✓	✓	✓	ตรวจสอบด้วยสายตา
3	x	x	x	x	x	x	x	x	x	ตรวจสอบด้วยสายตา
4	x	x	x	x	x	x	x	x	x	ตรวจสอบด้วยสายตา
5	x	x	x	x	x	x	x	x	x	PRD ไม่เกิน 5%

หมายเหตุ ✓ หมายถึงสัญญาณที่ผ่านการลดขนาดข้อมูลเกิดความเพี้ยนในเกณฑ์ที่แพทย์ยอมรับได้
x หมายถึงสัญญาณที่ผ่านการลดขนาดข้อมูลเกิดความเพี้ยนในเกณฑ์ที่แพทย์ไม่สามารถยอมรับได้

ตารางที่ 6.4 (เป็นข้อสรุปจากแบบสอบถามความคิดเห็นจากแพทย์ข้อที่ 1 และข้อที่ 2 ในภาคผนวก ง) เป็นการแสดงผลการพิจารณารูปสัญญาณที่ผ่านการลดขนาดข้อมูลด้วยสายตาจากแพทย์ 5 ท่าน เพื่อหาชนิดของเวฟเล็ทที่สามารถนำไปใช้งานได้ดีที่สุด โดยเป็นการพิจารณาที่ CR = 6 เพียงค่าเดียวเท่านั้น เนื่องจากเป็นเงื่อนไขที่เหมาะสมในการนำไปใช้งานมากที่สุดในงานวิจัย จากตารางสามารถสรุปได้ว่าการใช้เวฟเล็ทแบบ Coiflet-3 มีแนวโน้มที่จะทำให้สามารถรักษาสาระสำคัญของสัญญาณไฟฟ้าหัวใจต้นแบบได้ดีที่สุด ซึ่งสอดคล้องกับค่าของ PRD ที่ต่ำที่สุดในรูปที่ 6.5 หรือรูปที่ 6.6

ตารางที่ 6.4 แสดงชนิดของเวฟเล็ทที่แพทย์ผู้วินิจฉัยมีความเห็นว่าสามารถรักษาสาระสำคัญของ
ข้อมูลสัญญาณไฟฟ้าหัวใจต้นแบบได้ดีที่สุด

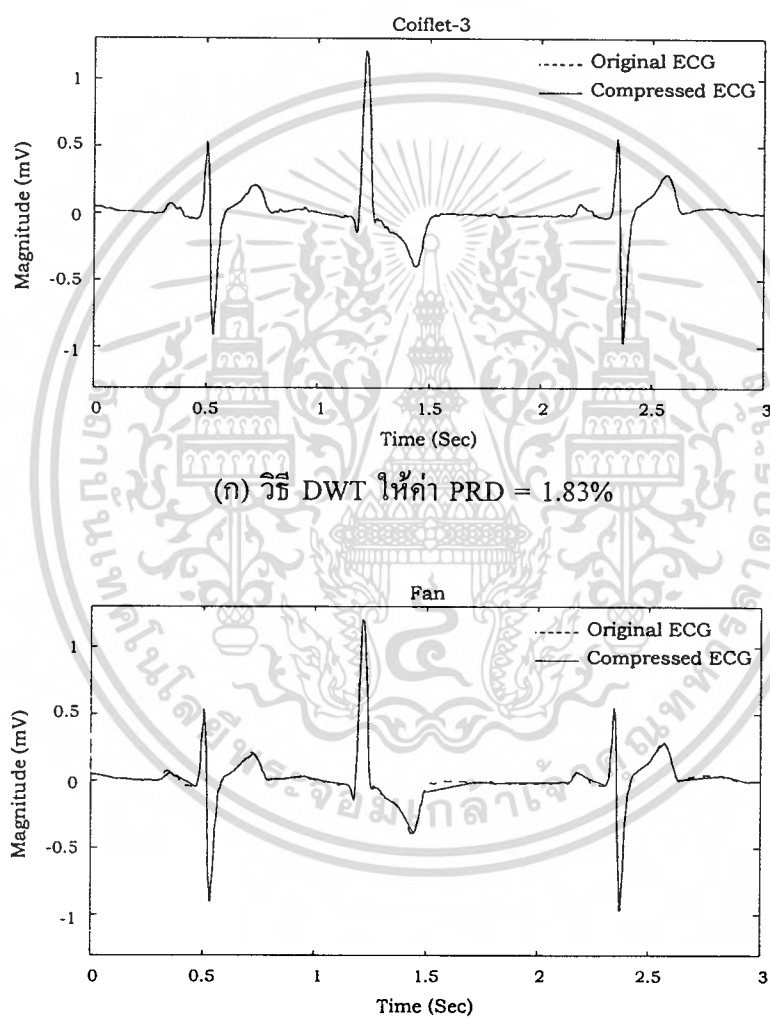
แพทย์ ท่านที่	Daubechies			Symmlet			Coiflet		
	D4	D12	D20	S4	S7	S10	C1	C3	C5
1	✓								
2				✓				✓	
3				✓				✓	
4								✓	
5								✓	

หมายเหตุ การใส่เครื่องหมาย ✓ มากกว่าหนึ่งช่องหมายถึงชนิดของเวฟเล็ทที่สามารถรักษาสาระสำคัญของ
สัญญาณไฟฟ้าหัวใจต้นแบบได้ดีที่สุดและใกล้เคียงกัน

6.3 การทดสอบเพิ่มเติมโดยใช้สัญญาณไฟฟ้าหัวใจผิดปกติ

6.3.1 สัญญาณไฟฟ้าหัวใจผิดปกติแบบ Premature Ventricular Contraction

รูปที่ 6.23 แสดงประสิทธิภาพการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจผิดปกติแบบ Premature Ventricular Contraction ด้วยวิธี DWT (Coiflet-3, $L = 4$) เปรียบเทียบกับวิธี Fan เมื่อกำหนดให้ $CR = 4$ จากค่า PRD และสัญญาณที่แสดงจะเห็นได้ว่าที่ CR เท่ากัน วิธี DWT จะให้ประสิทธิภาพการลดขนาดข้อมูลที่สูงกว่าวิธี Fan อย่างเห็นได้ชัด

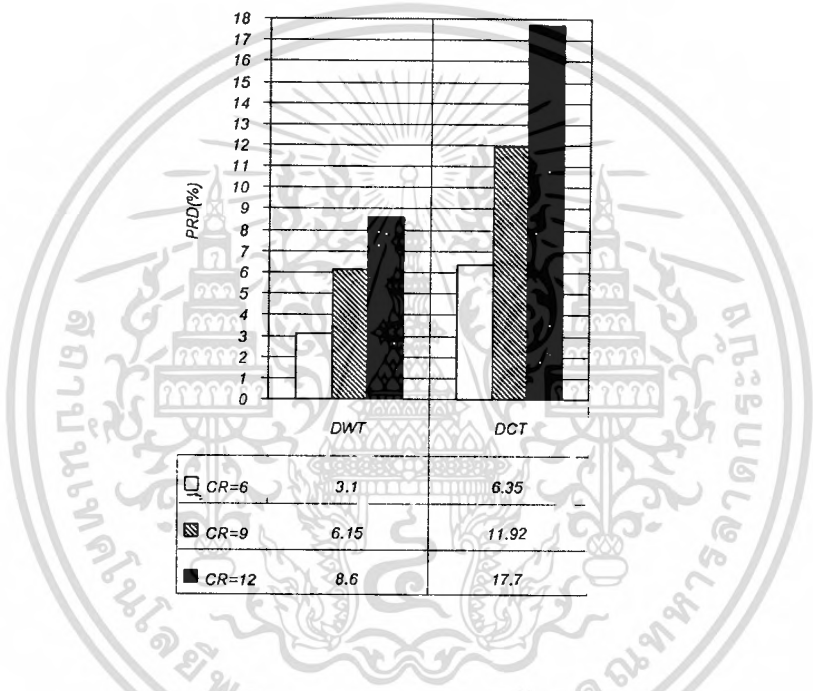


(ก) วิธี DWT ให้ค่า PRD = 1.83%

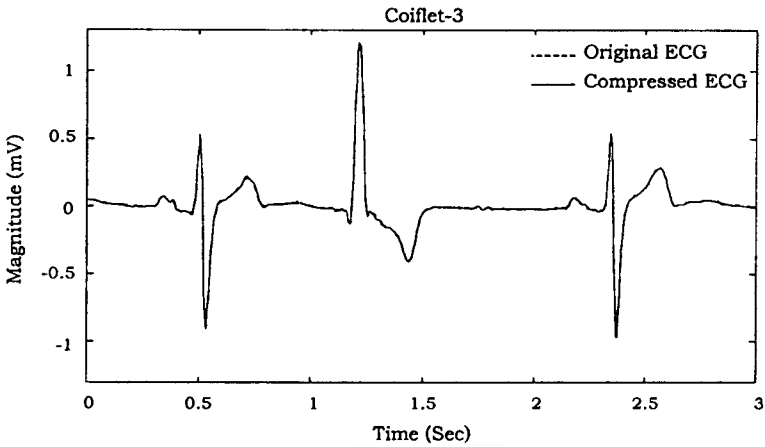
(ข) วิธี Fan ให้ค่า PRD = 6.91%

รูปที่ 6.23 แสดงสัญญาณไฟฟ้าหัวใจผิดปกติในลักษณะ Premature Ventricular Contraction เมื่อผ่านขบวนการลดขนาดข้อมูลด้วยวิธี DWT เปรียบเทียบกับวิธี Fan เมื่อกำหนดให้ $CR = 4$

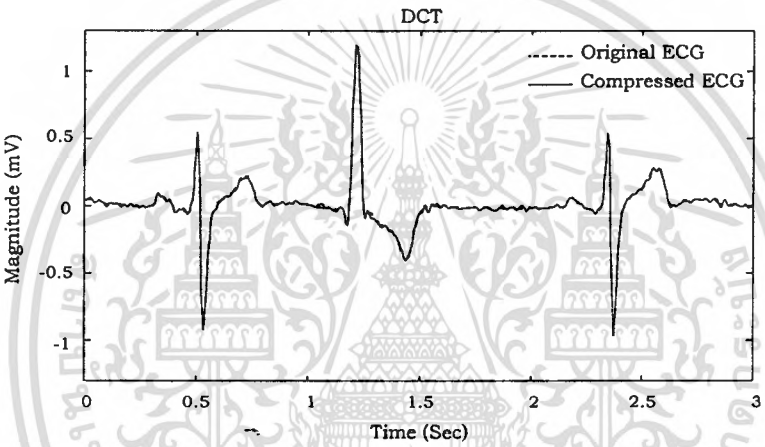
รูปที่ 6.24 ถึงรูปที่ 6.27 เป็นการแสดงประสิทธิภาพการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจผิดปกติแบบ Premature Ventricular Contraction ด้วยวิธี DWT (Coiflet-3, $L = 4$) เปรียบเทียบกับวิธี DCT เมื่อกำหนดให้ค่า CR เป็น 6, 9 และ 12 จากค่า PRD และสัญญาณที่แสดงจะเห็นว่าที่ CR เท่ากัน วิธี DWT จะให้ประสิทธิภาพการลดขนาดข้อมูลที่สูงกว่าวิธี DCT ทุกค่าของ CR วิธี DCT จะทำให้เกิดความเพี้ยนตลอดแกนเวลา(Globalize Error) ในขณะที่วิธี DWT จะทำให้เกิดความเพี้ยนมากเพียงบางช่วง(Localize Error)เท่านั้น ซึ่งทำให้สามารถรักษาสาระสำคัญของสัญญาณได้ดีกว่าโดยเฉพาะเมื่อค่า CR เป็น 9 และ 12



รูปที่ 6.24 แสดงประสิทธิภาพการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจผิดปกติแบบ Premature Ventricular Contraction ด้วยค่า PRD ของวิธี DWT เปรียบเทียบกับวิธี DCT

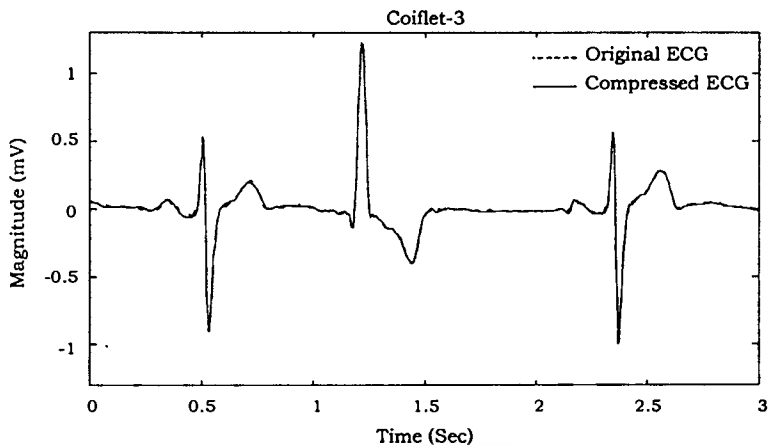


(ก) วิธี DWT ให้ค่า PRD = 3.1%

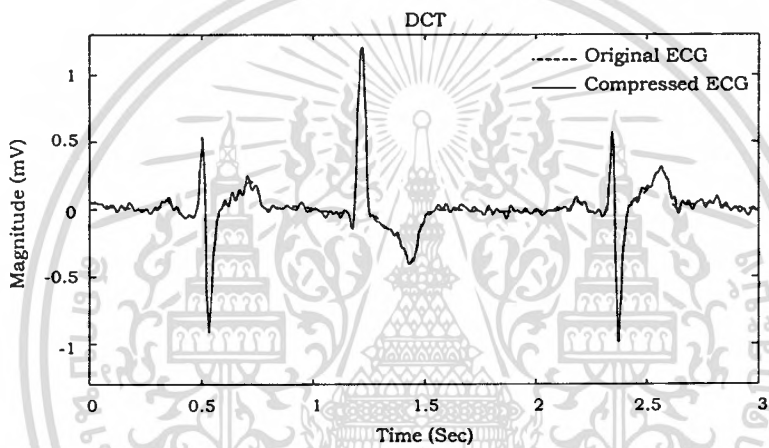


(ข) วิธี DCT ให้ค่า PRD = 6.35%

รูปที่ 6.25 แสดงสัญญาณไฟฟ้าหัวใจผิดปกติในลักษณะ Premature Ventricular Contraction ที่ผ่าน
ขบวนการลดขนาดข้อมูลด้วยวิธี DWT เปรียบเทียบกับวิธี DCT เมื่อกำหนดให้ CR = 6

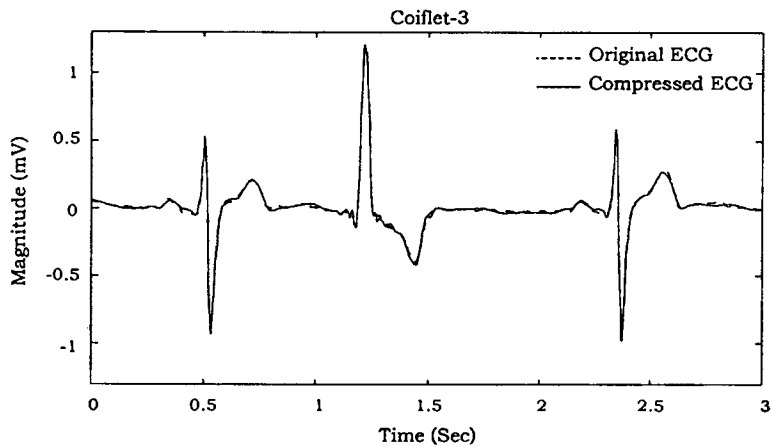


(ก) วิธี DWT ให้ค่า PRD = 6.15%

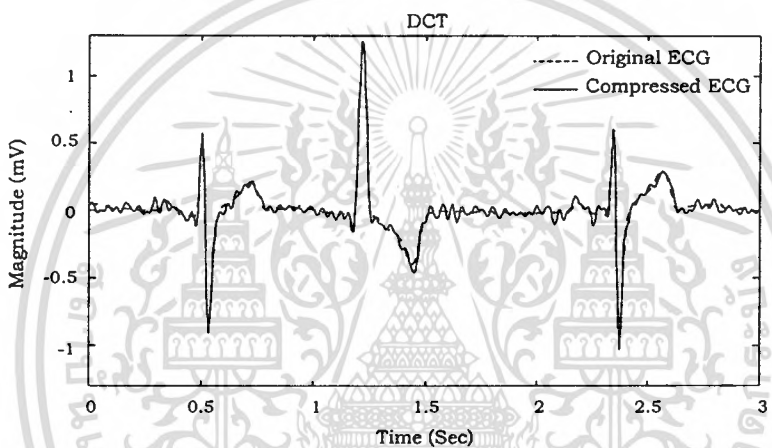


(ข) วิธี DCT ให้ค่า PRD = 11.92%

รูปที่ 6.26 แสดงสัญญาณไฟฟ้าหัวใจผิดปกติในลักษณะ Premature Ventricular Contraction ที่ผ่าน
ขบวนการลดขนาดข้อมูลด้วยวิธี DWT เปรียบเทียบกับวิธี DCT เมื่อกำหนดให้ CR = 9



(ก) วิธี DWT ให้ค่า PRD = 8.6%



(ข) วิธี DCT ให้ค่า PRD = 17.7%

รูปที่ 6.27 แสดงสัญญาณไฟฟ้าหัวใจผิดปกติในลักษณะ Premature Ventricular Contraction ที่ผ่าน
ขบวนการลดขนาดข้อมูลด้วยวิธี DWT เปรียบเทียบกับวิธี DCT เมื่อกำหนดให้ CR=12

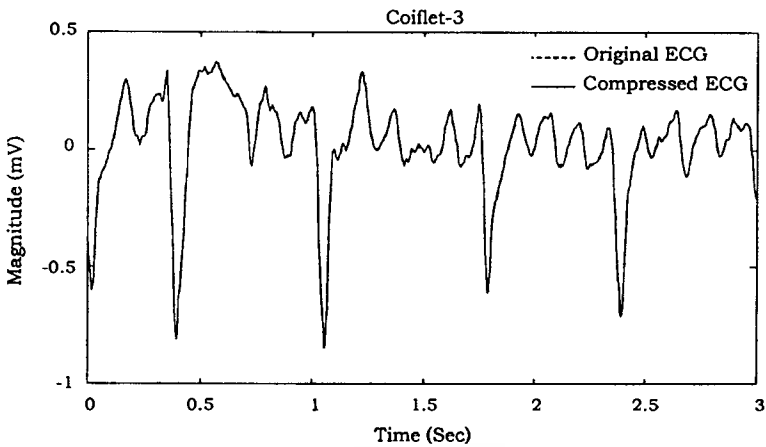
ตารางที่ 6.5 (เป็นข้อสรุปจากแบบสอบถามความคิดเห็นจากแพทย์ข้อที่ 8.1 ในภาคผนวก ง)แสดงค่า CR สูงสุดที่แพทย์ผู้วินิจฉัยยอมรับว่า สัญญาณไฟฟ้าหัวใจผิดปกติแบบ Premature Ventricular Contraction เมื่อผ่านการลดขนาดข้อมูลโดยใช้วิธี DWT จะทำให้เกิดความเพี้ยนอยู่ในระดับที่สัญญาณยังคงสามารถวินิจฉัยได้ผลเช่นเดียวกับสัญญาณต้นแบบ จากตารางจะเห็นว่าแพทย์ท่านที่ 5 เพียงท่านเดียวซึ่งพิจารณาด้วยค่า PRD ไม่เกิน 5% ยอมรับที่ CR = 6 (ให้ค่า PRD = 3.1%) ส่วน 4 ท่านแรกยอมรับได้ถึง 9 เท่า(ให้ค่า PRD = 6.15%)

ตารางที่ 6.5 แสดงค่า CR สูงสุดที่แพทย์ผู้วินิจฉัยยอมรับสำหรับการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจผิดปกติแบบ Premature Ventricular Contraction โดยใช้วิธี DWT

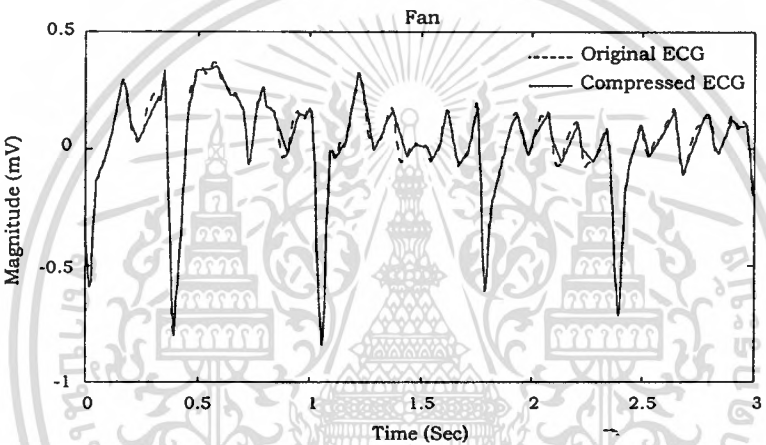
แพทย์ท่านที่	CR = 6	CR = 9	CR = 12
1		✓	
2		✓	
3		✓	
4		✓	
5	✓		

6.3.2 สัญญาณไฟฟ้าหัวใจผิดปกติแบบ Atrial Flutter

รูปที่ 6.28 แสดงประสิทธิภาพการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจผิดปกติแบบ Atrial Flutter ด้วยวิธี DWT (Coiflet-3, $L = 4$) เปรียบเทียบกับวิธี Fan เมื่อกำหนดให้ CR = 4 จากค่า PRD และสัญญาณที่แสดงจะเห็นได้ว่าที่ CR เท่ากัน วิธี DWT จะให้ประสิทธิภาพการลดขนาดข้อมูลที่สูงกว่าวิธี Fan อย่างเห็นได้ชัด



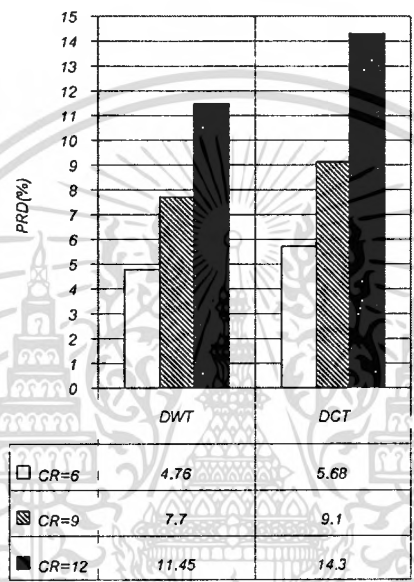
(ก) วิธี DWT ให้ค่า PRD = 3.33%



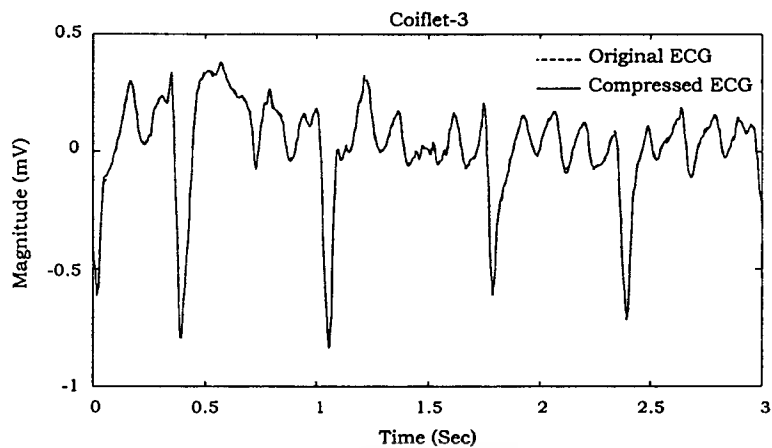
(ข) วิธี Fan ให้ค่า PRD = 13.8%

รูปที่ 6.28 แสดงสัญญาณไฟฟ้าหัวใจผิดปกติในลักษณะ Atrial Flutter เมื่อผ่านขั้นตอนการลดขนาดข้อมูลด้วยวิธี DWT เปรียบเทียบกับวิธี Fan เมื่อกำหนดให้ CR = 4

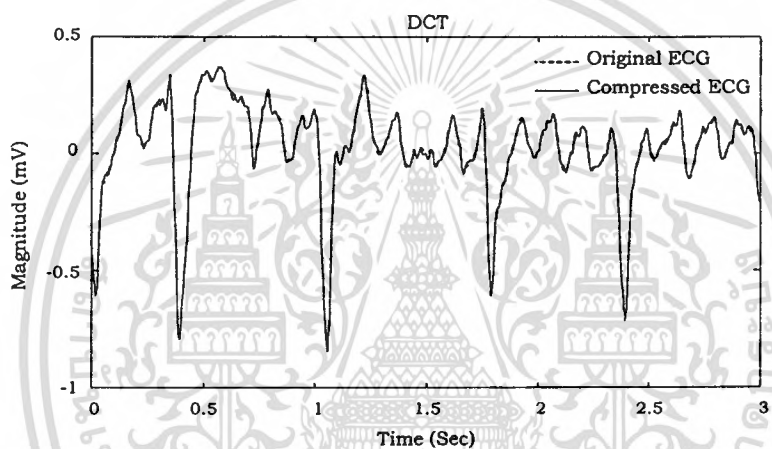
รูปที่ 6.29 ถึงรูปที่ 6.32 เป็นการแสดงประสิทธิภาพการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจผิดปกติแบบ Atrial Flutter ด้วยวิธี DWT (Coiflet-3, $L = 4$) เปรียบเทียบกับวิธี DCT เมื่อกำหนดให้ค่า CR เป็น 6, 9 และ 12 จากค่า PRD จะเห็นได้ว่าที่ CR เท่ากัน วิธี DWT จะให้ประสิทธิภาพการลดขนาดข้อมูลที่สูงกว่าวิธี DCT ทุกค่าของ CR และเมื่อพิจารณาจากสัญญาณที่แสดงจะเห็นว่าที่ค่า CR เป็น 6 และ 9 สัญญาณที่ได้จะมีคุณภาพไม่แตกต่างกันมากนัก ยกเว้นที่ CR = 12 วิธี DCT จะทำให้เกิดความเพี้ยนของขนาดที่มากกว่าวิธี DWT อย่างเห็นได้ชัด



รูปที่ 6.29 แสดงประสิทธิภาพการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจผิดปกติแบบ Atrial Flutter ด้วยค่า PRD ของวิธี DWT เปรียบเทียบกับวิธี DCT

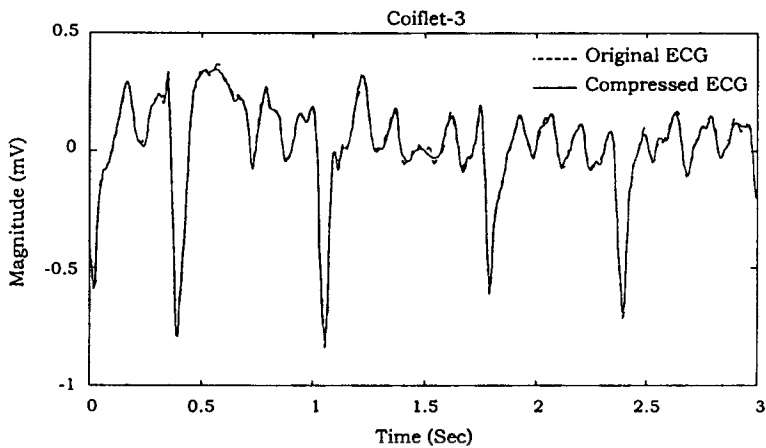


(ก) วิธี DWT ให้ค่า PRD = 4.76%

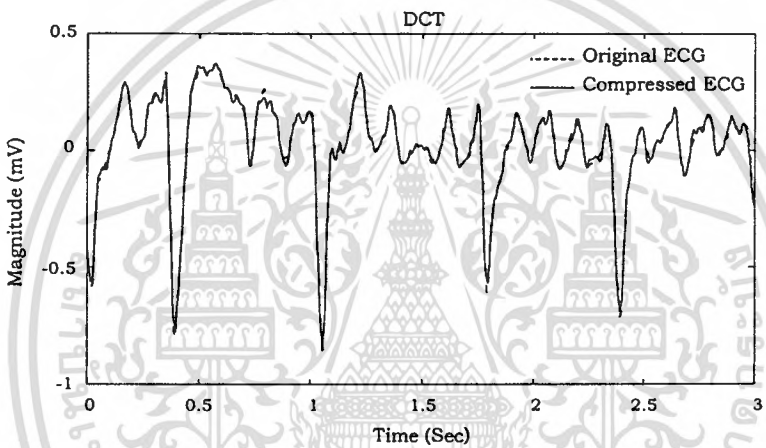


(ข) วิธี DCT ให้ค่า PRD = 5.68%

รูปที่ 6.30 แสดงสัญญาณไฟฟ้าหัวใจผิดปกติในลักษณะ Atrial Flutter ที่ผ่านขบวนการลดขนาดข้อมูลด้วยวิธี DWT เปรียบเทียบกับวิธี DCT เมื่อกำหนดให้ CR = 6

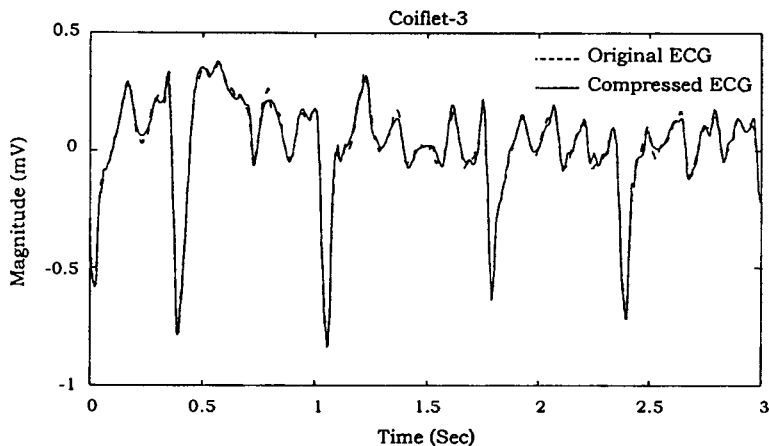


(ก) วิธี DWT ให้ค่า PRD = 7.7%

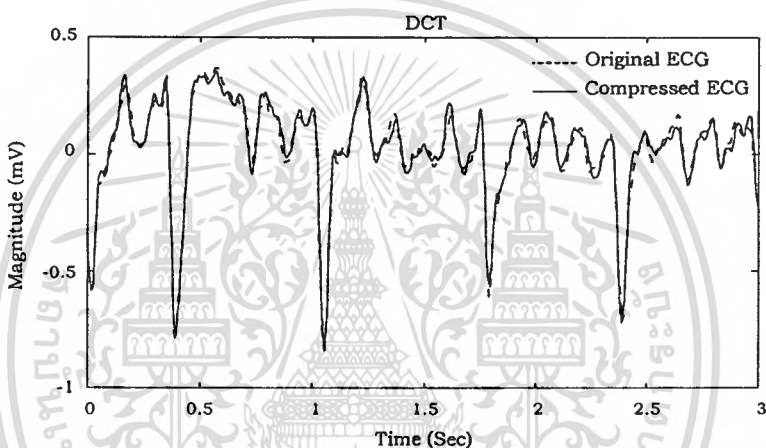


(ข) วิธี DCT ให้ค่า PRD = 9.1%

รูปที่ 6.31 แสดงสัญญาณไฟฟ้าหัวใจผิดปกติในลักษณะ Atrial Flutter ที่ผ่านขั้นตอนการลดขนาดข้อมูลด้วยวิธี DCT เปรียบเทียบกับวิธี DWT เมื่อกำหนดให้ CR = 9



(ก) วิธี DWT ให้ค่า PRD = 11.45%



(ข) วิธี DCT ให้ค่า PRD = 14.3%

รูปที่ 6.32 แสดงสัญญาณไฟฟ้าหัวใจผิดปกติในลักษณะ Atrial Flutter ที่ผ่านขบวนการลดขนาดข้อมูลด้วยวิธี DWT เปรียบเทียบกับวิธี DCT เมื่อกำหนดให้ $CR = 12$

ตารางที่ 6.6 (เป็นข้อสรุปจากแบบสอบถามความคิดเห็นจากแพทย์ข้อที่ 8.2 ในภาคผนวก ง)แสดงค่า CR สูงสุดที่แพทย์ผู้วินิจฉัยยอมรับว่า สัญญาณไฟฟ้าหัวใจผิดปกติแบบ Atrial Flutter เมื่อผ่านการลดขนาดข้อมูลโดยใช้วิธี DWT จะทำให้เกิดความเพี้ยนอยู่ในระดับที่สัญญาณยังคงสามารถวินิจฉัยได้ผลเช่นเดียวกับสัญญาณต้นแบบ จากตารางจะเห็นว่าแพทย์ท่านที่ 5 เพียงท่านเดียวซึ่งพิจารณาด้วยค่า PRD ไม่เกิน 5% ยอมรับที่ CR = 6 (ให้ค่า PRD = 4.76%) ส่วน 4 ท่านแรกยอมรับได้ถึง 12 เท่า(ให้ค่า PRD = 11.45%)

ตารางที่ 6.6 แสดงค่า CR สูงสุดที่แพทย์ผู้วินิจฉัยยอมรับสำหรับการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจผิดปกติแบบ Atrial Flutter โดยใช้วิธี DWT

แพทย์ท่านที่	CR = 6	CR = 9	CR = 12
1			✓
2			✓
3			✓
4			✓
5	✓		

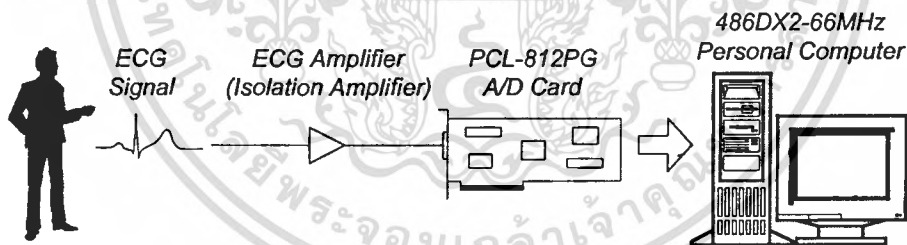
บทที่ 7

การประยุกต์ใช้งานกับคอมพิวเตอร์ส่วนบุคคล

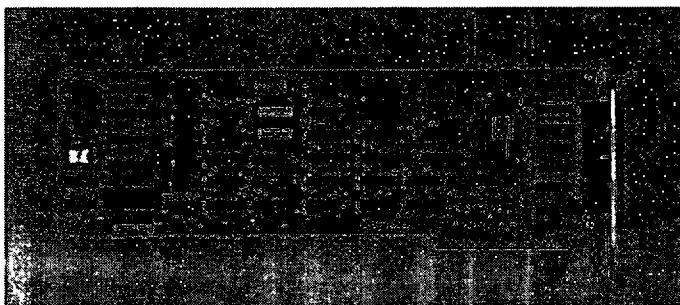
ในบทนี้เป็นการกล่าวถึงระบบบันทึกสัญญาณไฟฟ้าหัวใจด้วยคอมพิวเตอร์ส่วนบุคคล และการประยุกต์ใช้งานจริงในการบันทึกสัญญาณไฟฟ้าหัวใจโดยใช้วิธีการแปลงเวฟเล็ดในการลดขนาดข้อมูล ตลอดจนคุณสมบัติและการใช้งานของโปรแกรมที่ได้ออกแบบขึ้น

7.1 ระบบบันทึกสัญญาณไฟฟ้าหัวใจด้วยคอมพิวเตอร์ส่วนบุคคล

ระบบการบันทึกสัญญาณไฟฟ้าหัวใจด้วยคอมพิวเตอร์ส่วนบุคคลที่ใช้ในวิทยานิพนธ์นี้ แสดงในรูปที่ 7.1 สัญญาณไฟฟ้าหัวใจจากขั้ววัดไฟฟ้าที่ติดบนผิวหนังจะถูกขยายสัญญาณโดยวงจรขยายสัญญาณไฟฟ้าหัวใจ (ECG Amplifier) ที่มีส่วนป้องกันกระแสไฟฟ้ารั่วไหลไปสู่ร่างกาย (Isolation Amplifier) แสดงในรูปที่ 2.16 ในบทที่ 2 สัญญาณไฟฟ้าหัวใจแบบอนาลอกจากวงจรขยายจะถูกแปลงเป็นข้อมูลดิจิทัลเพื่อการประมวลผลในคอมพิวเตอร์ด้วย A/D Card รุ่น PCL-812PG ของบริษัท Advantech ซึ่งมีขนาดความละเอียดในการเก็บข้อมูล 12 Bits/Sample แสดงในรูปที่ 7.2 ส่วนคอมพิวเตอร์ที่ใช้ในเครื่องบันทึกสัญญาณต้นแบบใช้ CPU เบอร์ 486DX2 สัญญาณนาฬิกาความถี่ 66 MHz



รูปที่ 7.1 แสดงระบบบันทึกสัญญาณไฟฟ้าหัวใจด้วยคอมพิวเตอร์ส่วนบุคคลที่ใช้ในวิทยานิพนธ์



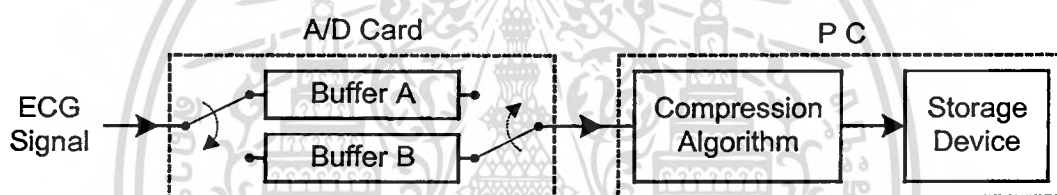
รูปที่ 7.2 แสดง A/D Card ที่ใช้ในระบบบันทึกสัญญาณไฟฟ้าหัวใจด้วยคอมพิวเตอร์ส่วนบุคคล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2 โปรแกรมและการควบคุมการทำงานของระบบบันทึกสัญญาณไฟฟ้าหัวใจ

7.2.1 การควบคุมการทำงานของ A/D Card

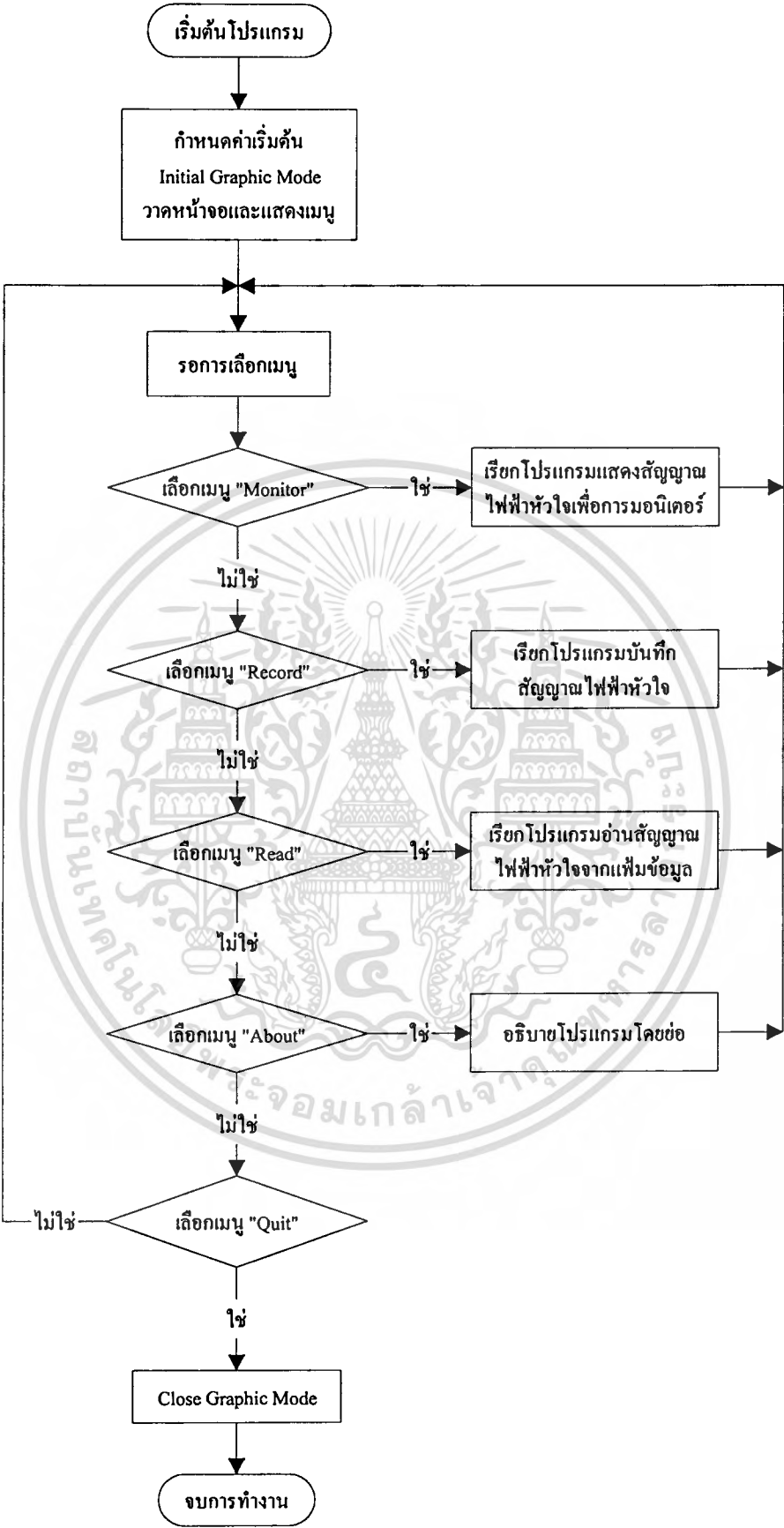
สำหรับ A/D Card รุ่น PCL-812PG ของบริษัท Advantech ที่นำมาใช้ในการออกแบบระบบบันทึกสัญญาณไฟฟ้าหัวใจนี้ ใช้วิธีการควบคุมการทำงานโดยการโปรแกรมบนคอมพิวเตอร์ ในการออกแบบได้กำหนดให้ A/D Card นี้ทำงานที่ความถี่สุ่ม 200 Hz และทำงานในลักษณะของ Double Buffer Cyclic Mode แสดงในรูปที่ 7.3 ซึ่งเป็นการทำงานสลับกันระหว่างบัฟเฟอร์(Data Buffer)ที่มีขนาดเท่ากัน 2 ชุดๆละ 2048 Samples กล่าวคือ ในขณะที่บัฟเฟอร์ A กำลังรับข้อมูล ช่วงเวลานั้นก็สามารถนำข้อมูลจากบัฟเฟอร์ B ไปผ่านขบวนการลดขนาดข้อมูลหรือแสดงผลบนจอภาพ เป็นเช่นนี้สลับกันไป ซึ่งจะทำให้สามารถประมวลผลข้อมูลสัญญาณไฟฟ้าหัวใจได้อย่างต่อเนื่องโดยไม่มีการสูญเสียของข้อมูล รายละเอียดในการควบคุมการทำงานของ A/D Card รุ่น PCL-812PG ด้วยการโปรแกรมสามารถดูได้จากเอกสารอ้างอิง[10]



รูปที่ 7.3 แสดงการทำงานของ A/D Card ในลักษณะของ Double Buffer Cyclic Mode

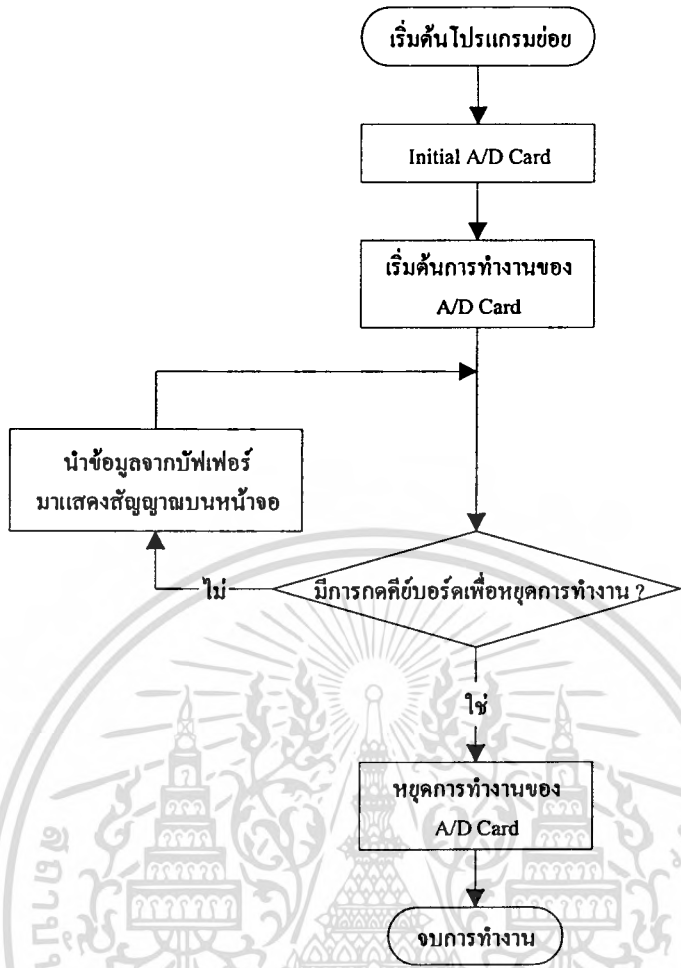
7.2.2 โปรแกรมระบบบันทึกสัญญาณไฟฟ้าหัวใจ

โปรแกรมที่ใช้ในการควบคุมการทำงานของ A/D Card และคอมพิวเตอร์ ตลอดจนอัลกอริทึมการลดขนาดข้อมูลด้วยวิธีการแปลงเวฟเลตที่ได้ออกแบบ เขียนขึ้นด้วยโปรแกรมภาษาซี เพื่อการใช้งานบนระบบปฏิบัติการ DOS โปรแกรมหลักโดยคร่าวๆของระบบบันทึกสัญญาณไฟฟ้าหัวใจที่ได้ออกแบบสามารถแสดงได้ด้วยแผนผังในรูปที่ 7.4 ซึ่งประกอบด้วยโปรแกรมย่อยที่สำคัญคือ โปรแกรมการแสดงสัญญาณไฟฟ้าหัวใจเพื่อการมอนิเตอร์แสดงในรูปที่ 7.5 โปรแกรมการบันทึกสัญญาณไฟฟ้าหัวใจแสดงในรูปที่ 7.6 และโปรแกรมการอ่านสัญญาณไฟฟ้าหัวใจจากแฟ้มข้อมูลแสดงในรูปที่ 7.7 จากแผนผังในรูปที่ 7.5 และรูปที่ 7.6 กระบวนการนำข้อมูลจากบัฟเฟอร์จะหมายถึงการสลับการนำข้อมูลจากบัฟเฟอร์ A หรือ B มาแสดงผลหรือผ่านขบวนการลดขนาดข้อมูล

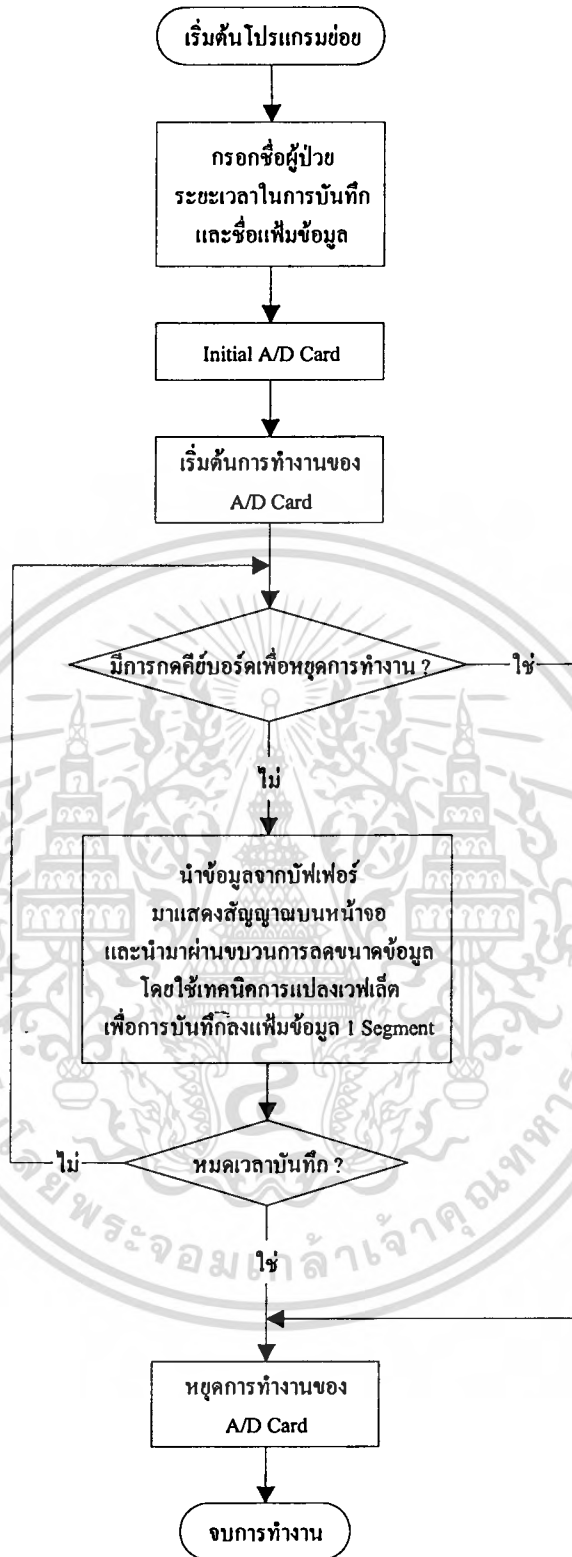


รูปที่ 7.4 แสดงแผนผังโปรแกรมหลักของระบบบันทึกสัญญาณไฟฟ้าหัวใจ

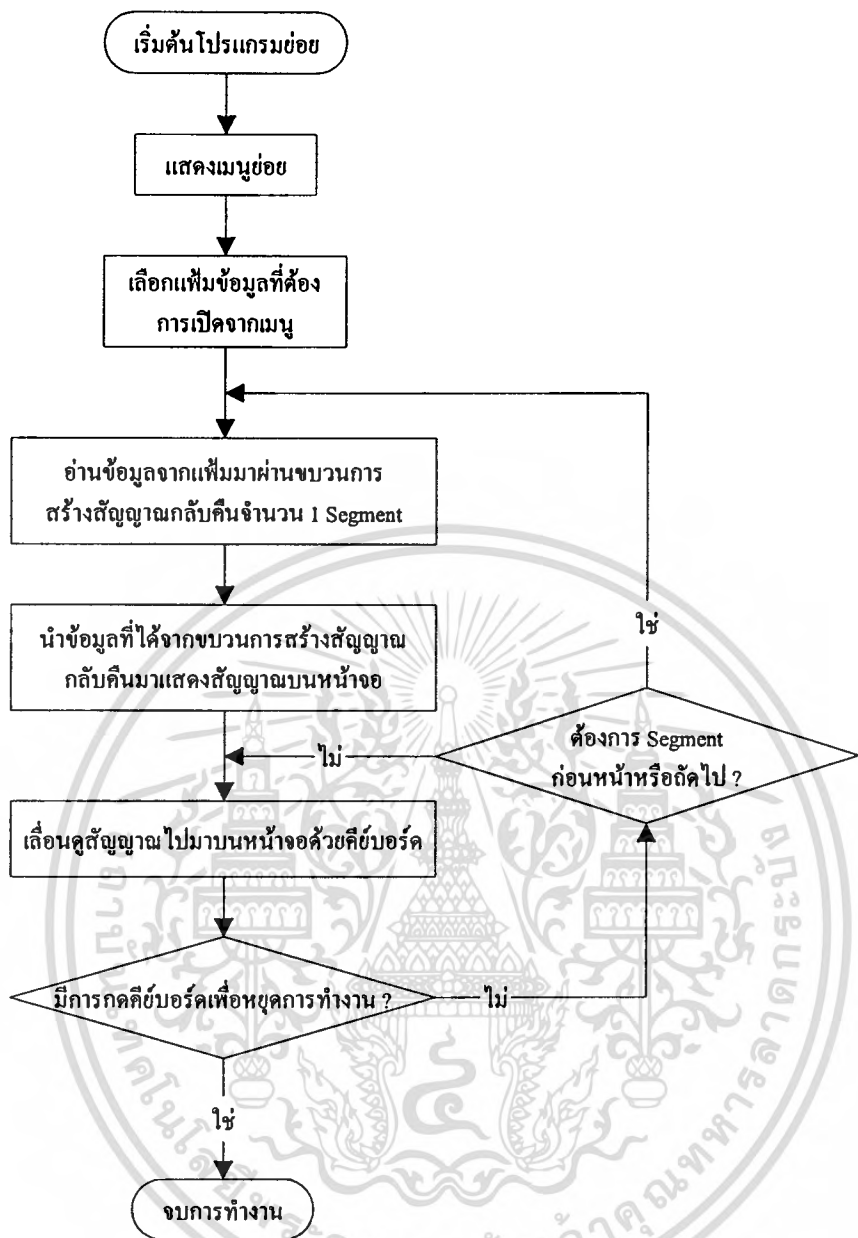
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.5 แสดงแผนผัง โปรแกรมย่อยที่ใช้ในการแสดงสัญญาณ ไฟฟ้าหัวใจเพื่อการมอนิเตอร์



รูปที่ 7.6 แสดงแผนผังโปรแกรมย่อยที่ใช้ในการบันทึกสัญญาณไฟฟ้าหัวใจ



รูปที่ 7.7 แสดงแผนผังโปรแกรมย่อยที่ใช้ในการอ่านสัญญาณไฟฟ้าหัวใจจากแฟ้มข้อมูล

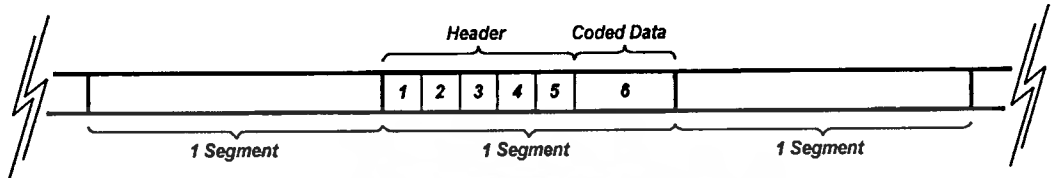
7.2.2.1 รูปแบบการบันทึกเป็นแฟ้มข้อมูลในอุปกรณ์บันทึก

จากการทดสอบการขนาดสัญญาณไฟฟ้าหัวใจด้วยวิธีการแปลงเวฟเล็ดในบทที่ 6 ทำให้สามารถแสดงรูปแบบการบันทึกเป็นแฟ้มข้อมูลในอุปกรณ์บันทึกแสดงในรูปที่ 7.8 ข้อมูลสัญญาณไฟฟ้าหัวใจแต่ละบล็อคจากบัฟเฟอร์ A หรือ B เมื่อผ่านขบวนการลดขนาดข้อมูลในรูปที่ 5.1 ในบทที่ 5 จะได้ข้อมูลเพื่อการบันทึกจำนวน 1 Segment เมื่อทำการบันทึกอย่างต่อเนื่องก็จะได้ข้อมูลแต่ละ Segment ที่ต่อเนื่องกันไป แต่ละ Segment ประกอบข้อมูล 2 ส่วนคือ ส่วนของ Header และส่วนของ Coded Data สำหรับโปรแกรมที่ออกแบบเพื่อใช้งานจริงกับคอมพิวเตอร์ส่วนบุคคล เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้เลือกใช้เวฟเลตชนิด Coiflet-3 ที่อันดับการแปลง L เป็น 4 ทำการเข้ารหัส Run-Length จำนวน 3 แบนด์ ($I[\hat{d}_j(n)]|_{j=10,9,8}$) และกำหนดช่วงของการแบ่งระดับในการแควนไทซ์ (Quantization Step Size) Δ เพื่อให้ CR อยู่ที่ประมาณ 6 เท่า รายละเอียดของข้อมูลที่จะต้องบันทึกลงเพิ่มข้อมูลในแต่ละ Segment สามารถอธิบายเรียงตามหมายเลขในรูปที่ 7.8 ได้ดังนี้

1. บันทึกค่าเฉลี่ยของ $c_{11}(n)$ ซึ่งจัดเก็บในรูปแบบของ Floating Point ขนาด 4 ไบต์
2. บันทึกข้อมูลที่ได้จากขบวนการ Run-Length ได้แก่
 - 2.1) อาร์เรย์ของจำนวนศูนย์ที่นำออกจาก $I[\hat{d}_j(n)]|_{j=10,9,8}$ แต่ละสมาชิกในอาร์เรย์จัดเก็บในรูปแบบของ Unsigned Interger ขนาด 11 บิต เรียงต่อกันไป แล้วบันทึกลงเพิ่มข้อมูลทีละ 1 ไบต์
 - 2.2) ค่าความยาวหรือจำนวนสมาชิกของอาร์เรย์ในข้อ 2.1 จัดเก็บในรูปแบบของ Unsigned Interger ขนาด 2 ไบต์
 - 2.3) ค่าความยาวรวมของ $I[\hat{d}_j(n)]|_{j=10,9,8}$ หลังจากผ่านขบวนการเข้ารหัส Run-Length จัดเก็บในรูปแบบของ Unsigned Interger ขนาด 2 ไบต์
3. บันทึกข้อมูลที่ได้จากขบวนการ Histogram Circular Shifting ได้แก่
 - 3.1) ขนาดสูงสุดสัมบูรณ์ของ $I[\hat{c}_j(n)]$ จัดเก็บในรูปแบบของ Unsigned Interger ขนาด 1 ไบต์
 - 3.2) จำนวนตำแหน่งที่เลื่อนไปของฮิสโตแกรม จัดเก็บในรูปแบบของ Unsigned Interger ขนาด 1 ไบต์
4. บันทึกตารางรหัสฮัฟแมน ได้แก่
 - 4.1) ค่าความยาวของตารางฮัฟแมน จัดเก็บในรูปแบบของ Unsigned Interger ขนาด 2 ไบต์
 - 4.2) อาร์เรย์ของ Quantized Data ที่ถูกเข้ารหัส แต่ละสมาชิกในอาร์เรย์จัดเก็บในรูปแบบของ Signed Interger ขนาด 1 ไบต์ เรียงต่อกันไป
 - 4.3) อาร์เรย์ของค่าความยาวบิตของรหัสฮัฟแมน แต่ละสมาชิกในอาร์เรย์จัดเก็บในรูปแบบของ Unsigned Interger ขนาด 4 บิต เรียงต่อกันไป แล้วบันทึกลงเพิ่มข้อมูลทีละ 1 ไบต์
 - 4.4) อาร์เรย์ของค่ารหัสฮัฟแมน แต่ละสมาชิกในอาร์เรย์จัดเก็บในรูปแบบของ Unsigned Interger ที่มีขนาดสัมพันธ์กับค่าความยาวบิตของรหัสฮัฟแมนนั้นๆ โดยจัดเก็บเรียงต่อกันไป แล้วบันทึกลงเพิ่มข้อมูลทีละ 1 ไบต์

5. จำนวนบิตรวมทั้งหมดที่ได้จากการเข้ารหัสฮัฟแมนของข้อมูล จัดเก็บในรูปแบบของ Unsigned Integer 2 ไบต์
6. บันทึกข้อมูลที่ได้จากการเข้ารหัสฮัฟแมน(มีลักษณะเป็นบิตที่เรียงต่อกันไป)โดยการบันทึกลงแฟ้มข้อมูลทีละ 1 ไบต์



รูปที่ 7.8 แสดงรูปแบบการบันทึกเป็นแฟ้มข้อมูลในอุปกรณ์บันทึก

7.3 คุณสมบัติของโปรแกรมระบบบันทึกสัญญาณไฟฟ้าหัวใจ

โปรแกรมระบบบันทึกสัญญาณไฟฟ้าหัวใจที่สร้างขึ้นมีคุณสมบัติดังต่อไปนี้

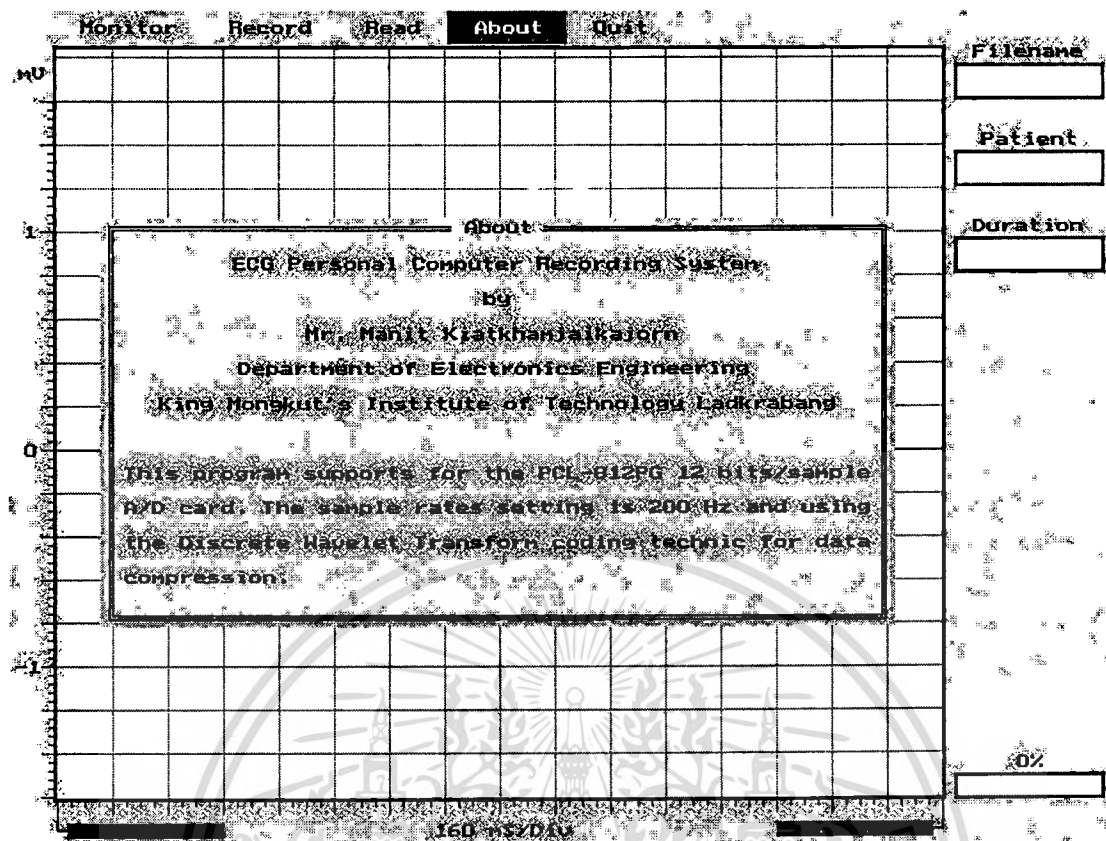
- สามารถแสดงสัญญาณไฟฟ้าหัวใจเพื่อการมอนิเตอร์
- สามารถบันทึกสัญญาณไฟฟ้าหัวใจเป็นแฟ้มข้อมูลในคอมพิวเตอร์ พร้อมกับแสดงสัญญาณไฟฟ้าหัวใจเพื่อการมอนิเตอร์ โดยสามารถกำหนดเวลาในการบันทึกได้ตั้งแต่ 1 นาที จนถึง 24 ชั่วโมง
- สามารถอ่านสัญญาณไฟฟ้าหัวใจจากแฟ้มข้อมูลที่บันทึกไว้เพื่อการวินิจฉัยบนหน้าจอคอมพิวเตอร์ สามารถค้นหาแฟ้มข้อมูลจากเมนู(Menu)
- สามารถแสดงชื่อผู้ป่วย เวลาสัมผัส ระยะเวลาที่ใช้ในการบันทึกและเปอร์เซ็นต์ของงานที่ทำขณะทำการบันทึกและการอ่าน

7.4 การใช้งานโปรแกรมระบบบันทึกสัญญาณไฟฟ้าหัวใจ

ในการใช้งานโปรแกรมระบบบันทึกสัญญาณไฟฟ้าหัวใจ ให้เริ่มจากการเรียกใช้คำสั่ง pcl-812.exe ที่ DOS Prompt ก่อนการเรียกใช้โปรแกรมที่ได้ออกแบบขึ้น คำสั่ง pcl-812.exe เป็นการโหลด Driver ของ A/D Card รุ่น PCL-812PG ลงในหน่วยความจำของคอมพิวเตอร์ หลังจากออกจากโปรแกรมบันทึกสัญญาณไฟฟ้าหัวใจให้เรียกใช้คำสั่ง free.exe เพื่อเป็นการคืนหน่วยความจำเนื่องจากการโหลด Driver ของ A/D Card เพื่อความสะดวกในการใช้งานโปรแกรม คำสั่งทั้ง 3 คำสั่งได้ถูกรวมเป็นคำสั่งแบตช์(Batch)เพื่อการเรียกใช้งานที่ DOS Prompt เพียงคำสั่งเดียว

รูปที่ 7.9 แสดงโปรแกรมระบบบันทึกสัญญาณไฟฟ้าหัวใจที่ได้ออกแบบขึ้น บริเวณมุมบนทางด้านขวาแสดงชื่อแฟ้มข้อมูล ชื่อผู้ป่วย และเวลาที่ใช้ในการบันทึกในหน่วยของชั่วโมงและนาที บริเวณมุมล่างทางด้านขวาแสดงเปอร์เซ็นต์ของงานที่ทำในขณะที่ทำการบันทึกและอ่าน ส่วนบริเวณที่ใช้ในการแสดงสัญญาณไฟฟ้าหัวใจจะมีช่วงเวลาในการแสดงสัญญาณ 2.56 วินาที(512 Samples) โดยบริเวณมุมล่างทางด้านซ้ายและขวาแสดงเวลาสัมพัทธ์หรือเวลาที่ผ่านไปเริ่มจากการบันทึก โปรแกรมระบบบันทึกสัญญาณไฟฟ้าหัวใจที่ออกแบบขึ้นประกอบด้วยเมนูทั้งหมด 5 เมนู ซึ่งมีรายละเอียดดังต่อไปนี้

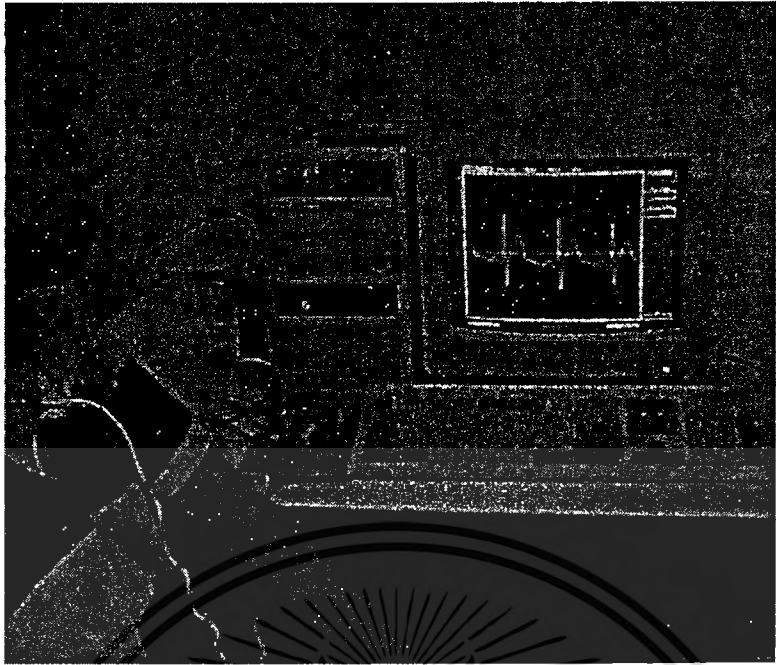
- **เมนู Monitor** ใช้ในการแสดงสัญญาณไฟฟ้าหัวใจบนจอภาพเพื่อการมอนิเตอร์เท่านั้น ไม่มีการบันทึกข้อมูลในขณะที่ทำการแสดงสัญญาณ สามารถยกเลิกการแสดงผลสัญญาณได้โดยกด Ctrl-Esc ค้างไว้
- **เมนู Record** ใช้ในการบันทึกสัญญาณไฟฟ้าหัวใจลงแฟ้มข้อมูลในคอมพิวเตอร์ พร้อมทั้งแสดงสัญญาณเพื่อการมอนิเตอร์ ก่อนการบันทึกผู้ใช้จะต้องระบุชื่อผู้ป่วย ระยะเวลาที่จะทำการบันทึกเป็นชั่วโมงและนาที และชื่อแฟ้มข้อมูลที่จะทำการบันทึก สามารถยกเลิกการทำงานระหว่างที่ทำการบันทึกได้โดยกด Ctrl-Esc ค้างไว้
- **เมนู Read** ใช้ในการอ่านสัญญาณไฟฟ้าหัวใจที่บันทึกไว้เพื่อแสดงบนจอภาพ สามารถเลื่อนดูสัญญาณไปข้างหน้าและย้อนถอยหลังทีละ 2.56 วินาทีต่อเนื่องกันโดยการกด Enter และ Backspace ตามลำดับ และสามารถเลื่อนดูสัญญาณไปข้างหน้าและย้อนถอยหลังแบบรวดเร็ว(กระโดดทีละ 10.24 วินาที)โดยการกด Ctrl-Enter และ Ctrl-Backspace ตามลำดับ สามารถยกเลิกการอ่านสัญญาณหรือปิดแฟ้มข้อมูลโดยกด Esc
- **เมนู About** ใช้ดูคำอธิบายโดยย่อของโปรแกรม
- **เมนู Quit** ใช้ในการออกจากโปรแกรมไปสู่ DOS Prompt



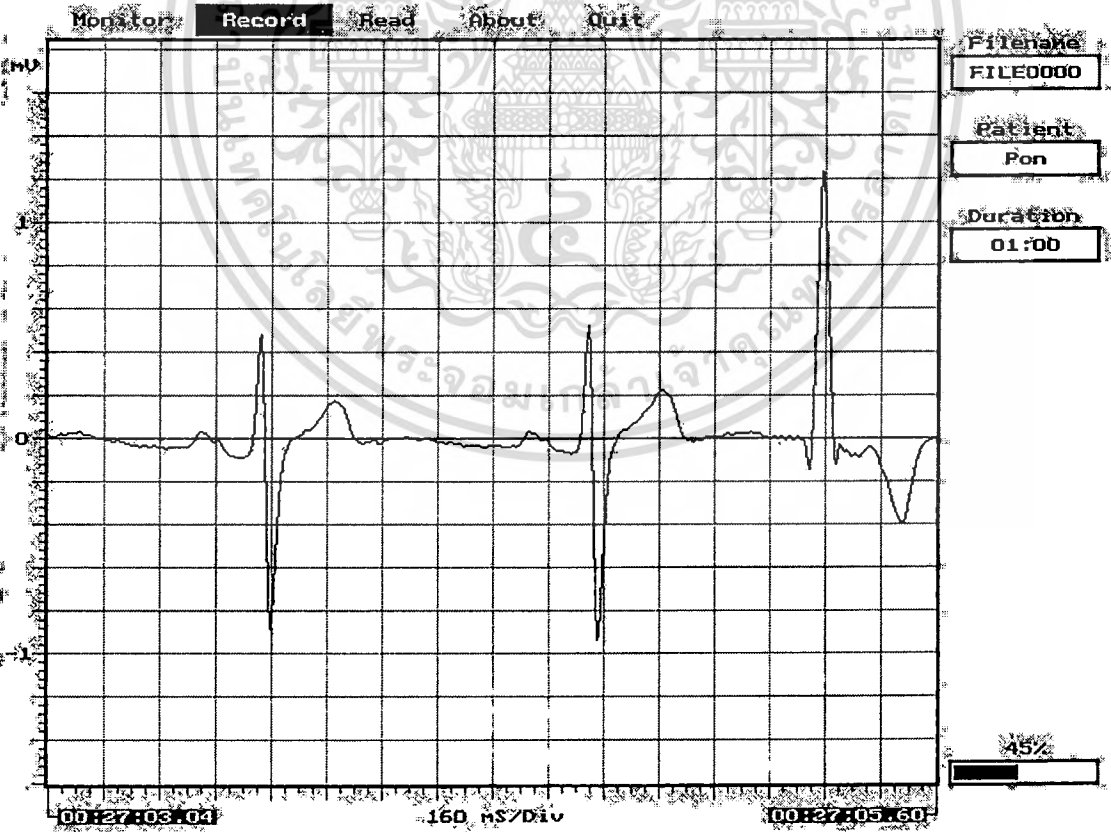
รูปที่ 7.9 แสดงลักษณะของโปรแกรมระบบบันทึกสัญญาณไฟฟ้าหัวใจบนหน้าจอคอมพิวเตอร์

7.5 การทดสอบระบบบันทึกสัญญาณไฟฟ้าหัวใจ

ในการทดสอบระบบบันทึกสัญญาณไฟฟ้าหัวใจ ได้ทำการทดสอบการแสดงผลสัญญาณเพื่อการมอนิเตอร์ การบันทึกสัญญาณเพื่อเก็บเป็นแฟ้มข้อมูล และการอ่านแฟ้มข้อมูลเพื่อแสดงสัญญาณที่บันทึกไว้ โดยในรูปที่ 7.10 แสดงระบบบันทึกสัญญาณไฟฟ้าหัวใจต้นแบบขณะทำการทดสอบการแสดงผลสัญญาณเพื่อการมอนิเตอร์ รูปที่ 7.11 แสดงหน้าจอมอนิเตอร์ขณะทำการทดสอบการบันทึกสัญญาณไฟฟ้าหัวใจจากบุคคลทั่วไปเป็นเวลาประมาณ 1 ชั่วโมง (352 บล็อก หรือ 3604.48 วินาที) ซึ่งผลที่ได้จากการบันทึกคือ สามารถลดขนาดข้อมูลจาก 1081344 ไบต์ (คำนวณจากสมการที่ (4.12) โดยที่ $bps = 12 \text{ Bits/Sample}$, $f_s = 200 \text{ Hz}$ และ $t_r = 3604.48 \text{ วินาที}$) เหลือเพียง 184845 ไบต์ หรือสามารถลดขนาดข้อมูลได้ประมาณ 5.85 เท่า ซึ่งใกล้เคียงกับค่าที่ตั้งไว้คือ $CR = 6$ และสามารถทำการอ่านแฟ้มข้อมูลที่ได้จากการทดสอบการบันทึกข้างต้นมาแสดงสัญญาณบนหน้าจอได้จริงในรูปที่ 7.12 โดยเป็นการแสดงผลสัญญาณไฟฟ้าหัวใจที่เวลาบันทึกเดียวกันกับรูปที่ 7.11 จะเห็นว่าสัญญาณที่อ่านจากแฟ้มข้อมูลที่บันทึกไว้มีความเพี้ยนเนื่องจากผลของการลดขนาดข้อมูลเพียงเล็กน้อยเท่านั้น

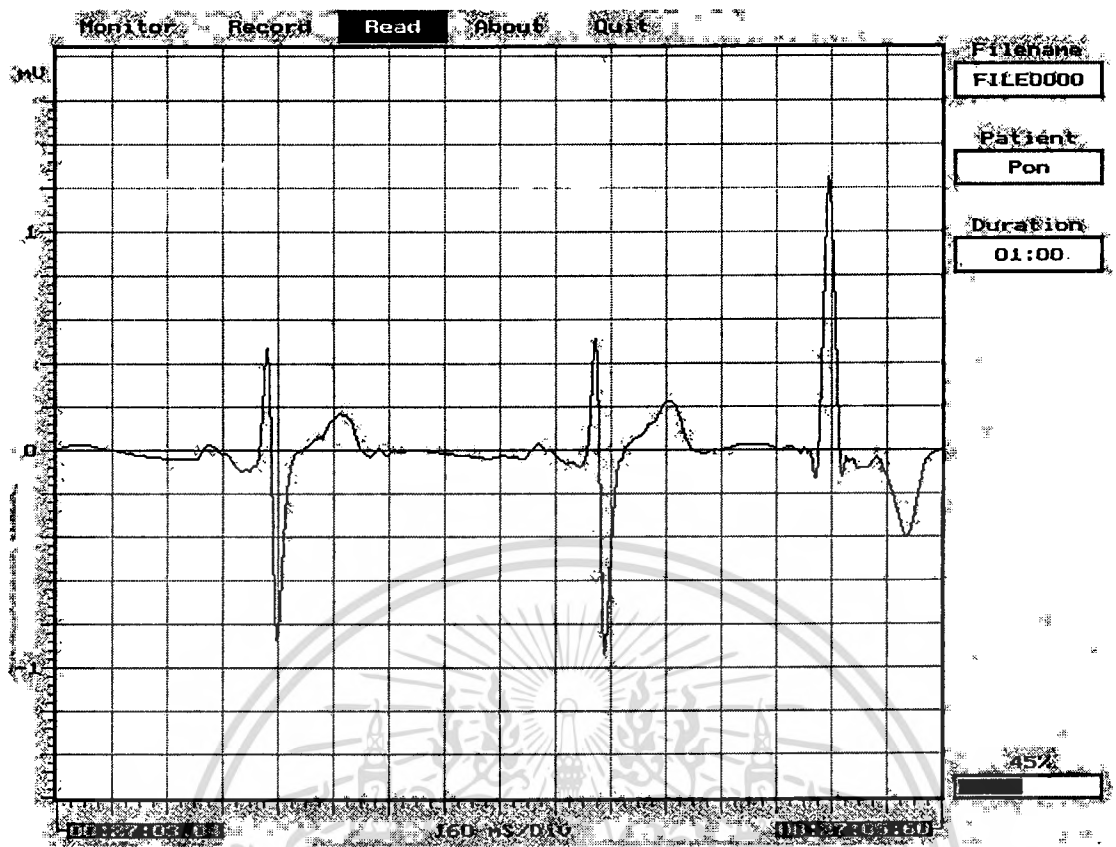


รูปที่ 7.10 แสดงระบบบันทึกสัญญาณไฟฟ้าหัวใจด้วยเครื่องบันทึกสัญญาณ
เพื่อการมอนิเตอร์



รูปที่ 7.11 แสดงหน้าจอโปรแกรมขณะทำการทดสอบการบันทึกสัญญาณไฟฟ้าหัวใจเป็นแฟ้ม

ข้อมูล
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.12 แสดงหน้าจอโปรแกรมขณะทำการทดสอบการอ่านสัญญาณไฟฟ้าหัวใจจากแฟ้มข้อมูล

สรุปผลการวิจัยและข้อเสนอแนะ

วิทยานิพนธ์นี้ ได้นำเสนอการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจโดยใช้การแปลงเวฟเล็ต ซึ่งจัดอยู่ในวิธีการลดขนาดข้อมูลด้วยวิธีการแปลง(Transformation Method) และเป็นวิธีการลดขนาดข้อมูลแบบที่มีการสูญเสีย(Lossy Compression) ทฤษฎีการแปลงเวฟเล็ตใช้พื้นฐานของการรวมเชิงเส้นของ Localize Basis Function ดังนั้นการเปลี่ยนแปลงค่าของสัมประสิทธิ์เวฟเล็ตใดๆในโดเมนของเวฟเล็ตจะทำให้เกิดการเปลี่ยนแปลงในโดเมนเดิมเพียงบางช่วงและบางตำแหน่งเท่านั้น การประยุกต์ใช้วิธีการแปลงเวฟเล็ตในการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจนั้นเมื่อสัมประสิทธิ์เวฟเล็ตผ่านขบวนการแควนไตซ์และการแปลงกลับเวฟเล็ตแล้ว ความเพี้ยนสะสมที่เกิดขึ้นกับสัญญาณไฟฟ้าหัวใจ ณ ช่วงและตำแหน่งใดๆในโดเมนเดิมจะมีสาเหตุเนื่องมาจากการเปลี่ยนแปลงค่าของสัมประสิทธิ์เวฟเล็ตเพียงบางส่วนเท่านั้น ลักษณะแบบ Localize Error นี้ จึงทำให้วิธีการแปลงเวฟเล็ตสามารถลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจได้อย่างมีประสิทธิภาพ

การแปลงเวฟเล็ตที่ใช้ในวิทยานิพนธ์นี้เป็นการแปลงในลักษณะของ Octave Band Filter Bank ซึ่งแบ่งข้อมูลออกเป็นกลุ่มข้อมูลย่อยที่ย่านความถี่ต่างๆ ทำให้สามารถประมวลผลข้อมูลแต่ละกลุ่มได้อย่างอิสระต่อกัน จากการทดสอบพบว่าภายหลังจากการแควนไตซ์(แบบยูนิฟอร์ม) สัมประสิทธิ์เวฟเล็ต กลุ่มข้อมูลย่อยในย่านความถี่สูงจะปรากฏศูนย์อยู่เป็นจำนวนมาก วิธีการเข้ารหัส Run-Length เฉพาะข้อมูลศูนย์ของกลุ่มข้อมูลนี้จึงสามารถเพิ่มประสิทธิภาพการลดขนาดข้อมูลได้มาก ส่วนกลุ่มข้อมูลในย่านความถี่ต่ำซึ่งมีฮิสโตแกรมกระจายตัวหนาแน่นอยู่ทางด้านขวาของศูนย์(รูปที่ 6.17(ก)) จะใช้วิธี Histogram Circular Shifting ทำการเลื่อนฮิสโตแกรมแบบวนไปทางซ้ายโดยให้ตำแหน่งที่มีความน่าจะเป็นสูงสุดอยู่ที่ศูนย์(รูปที่ 6.17(ข)) เพื่อเป็นการเพิ่มจำนวนข้อมูลที่กระจายตัวอยู่รอบๆศูนย์ในส่วนของกลุ่มข้อมูลย่อยในย่านความถี่สูง ซึ่งทำให้สามารถทำการเข้ารหัสฮัฟแมนได้อย่างมีประสิทธิภาพ จากผลการทดสอบสำหรับสัญญาณไฟฟ้าหัวใจปกติทั่วไป วิธี Histogram Circular Shifting สามารถเพิ่มค่า CR (ในขณะที่ยัง PRD ไม่เปลี่ยนแปลง) จาก 3, 6, 9 และ 12 ไปเป็น 3.08, 6.08, 9.05 และ 12.03 ตามลำดับ ซึ่งให้ผลไม่เป็นที่น่าพอใจ

เนื่องจากนักคณิตศาสตร์ได้สร้างฟังก์ชันมูลฐานเวฟเล็ต(Wavelet Basis Function)ไว้เป็นมาตรฐานสำหรับการนำไปใช้งานอย่างมากมาย ในงานวิจัยนี้ได้ทำการทดสอบการลดขนาดข้อมูลโดยใช้เวฟเล็ตหลายๆชนิดเพื่อหาข้อสรุปในการใช้งาน จากผลการทดสอบพบว่า ชนิดของเวฟเล็ตที่เลือกใช้ในการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจมีผลกับประสิทธิภาพการลดขนาดข้อมูล กล่าวคือ การเลือกใช้เวฟเล็ตที่มีค่าความราบเรียบ(Number of Vanishing Moments : NVM)น้อยเกินไป
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะทำให้สัญญาณที่ผ่านการลดขนาดข้อมูลมีลักษณะที่คมและไม่ต่อเนื่อง ในทางตรงกันข้าม การเลือกใช้เวฟเลตที่มีค่าความราบเรียบมากเกินไปจะทำให้สัญญาณที่ได้มีลักษณะที่ออสซิลเลท ซึ่งทั้งสองกรณีล้วนแล้วแต่ทำให้ค่า PRD สูงกว่าการเลือกใช้เวฟเลตที่ค่าความราบเรียบปานกลาง(ดูรูปที่ 6.5) อย่างไรก็ตามจากการพิจารณาสัญญาณไฟฟ้าหัวใจปกติทั่วไปที่ผ่านขบวนการลดขนาดข้อมูลด้วยวิธีการแปลงเวฟเลตจากแพทย์ทั้งหมด 5 ท่าน สามารถสรุปได้ว่า ค่า CR ที่เหมาะสมในการนำไปใช้งานคือ 6 เท่า และเวฟเลตแบบ Coiflet-3 มีแนวโน้มที่จะให้ประสิทธิภาพการลดขนาดข้อมูลสูงสุด ซึ่งสอดคล้องกับค่า PRD ที่ต่ำที่สุดในการทดสอบอีกด้วย(รูปที่ 6.5) และเมื่อใช้เวฟเลตแบบ Coiflet-3 ทำการทดสอบเพิ่มเติมกับสัญญาณไฟฟ้าหัวใจผิดปกติแบบ Premature Ventricular Contraction และ Atrial Flutter ปรากฏว่า แพทย์จำนวน 4 ท่านจากทั้งหมด 5 ท่านมีความเห็นสามารถเพิ่มค่า CR ได้ถึง 9 เท่าสำหรับสัญญาณไฟฟ้าหัวใจผิดปกติแบบ Premature Ventricular Contraction และสามารถเพิ่มค่า CR ได้ถึง 12 เท่าสำหรับสัญญาณไฟฟ้าหัวใจผิดปกติแบบ Atrial Flutter อย่างไรก็ตาม ค่า CR ที่เหมาะในการนำไปใช้งานคือค่า CR = 6 ซึ่งเป็นค่าต่ำสุดจากการทดสอบสัญญาณไฟฟ้าหัวใจปกติและผิดปกติที่แพทย์ยอมรับ และที่ค่า CR = 6 นี้จะทำให้เกิด PRD อยู่ในช่วง 3-5 %

เพื่อให้เห็นถึงประสิทธิภาพการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจโดยใช้วิธีการแปลงเวฟเลตที่นำเสนอ ในงานวิจัยจึงได้แสดงประสิทธิภาพการลดขนาดข้อมูลด้วยวิธีการแปลงเวฟเลตเปรียบเทียบกับวิธีการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจแบบอื่นๆ ได้แก่ วิธี Fan ซึ่งเป็นวิธีลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจด้วยวิธีทางตรง(Direct Method)ที่มีประสิทธิภาพสูงสุดในกลุ่มของ Significant-Point-Extraction และวิธีการแปลงโคไซน์เต็มหน่วย(Discrete Cosine Transform : DCT) ซึ่งเป็นวิธีลดขนาดข้อมูลด้วยวิธีการแปลง(Transformation Method)ที่มีประสิทธิภาพสูงและเป็นที่ยอมรับ จากการทดสอบสามารถสรุปได้ว่า การลดขนาดข้อมูลด้วยวิธีการแปลงเวฟเลต(Coiflet-3)จะให้ค่า PRD น้อยกว่าวิธี Fan และวิธี DCT ทุกๆค่า CR ที่ทำการทดสอบ และสัญญาณไฟฟ้าหัวใจที่ผ่านการลดขนาดข้อมูลด้วยวิธีการแปลงเวฟเลตจะมีคุณภาพของสัญญาณที่ดีกว่าวิธี Fan และวิธี DCT อย่างเห็นได้ชัด โดยผลการทดสอบเปรียบเทียบกับวิธี Fan ที่ CR = 4 วิธี Fan จะทำให้เกิดความเพี้ยนที่มากเป็นช่วงๆอย่างเห็นได้อย่างชัด ในขณะที่ยังไม่สามารถสังเกตเห็นความเพี้ยนได้อย่างชัดเจนเมื่อใช้วิธีการแปลงเวฟเลต ส่วนผลการทดสอบเปรียบเทียบกับวิธี DCT ที่ CR เท่ากับ 6, 9 และ 12 วิธี DCT จะทำให้สัญญาณไฟฟ้าหัวใจปกติและผิดปกติแบบ Premature Ventricular Contraction ที่ผ่านการลดขนาดข้อมูลเกิดความเพี้ยนในลักษณะ Globalize Error ส่วนวิธีการแปลงเวฟเลตจะให้เกิดความเพี้ยนในลักษณะ Localize Error ซึ่งทำให้สัญญาณไฟฟ้าหัวใจที่ผ่านการลดขนาดข้อมูลมีคุณภาพที่ดีกว่าอย่างเห็นได้อย่างชัดเจน

เมื่อ CR เท่ากับ 9 และ 12 ส่วนสัญญาณไฟฟ้าหัวใจผิดปกติแบบ Atrial Flutter เมื่อพิจารณาด้วยรูปสัญญาณแล้วพบว่า วิธีการแปลงเวฟเลตและวิธี DCT จะให้คุณภาพของสัญญาณค่อนข้างใกล้เคียงกัน ยกเว้นที่ $CR = 12$ วิธีการแปลงเวฟเลตจะให้ความเพี้ยนที่น้อยกว่าวิธี DCT อย่างเห็นได้ชัด

ส่วนข้อเสนอแนะในการพัฒนาอัลกอริทึมการลดขนาดข้อมูลสัญญาณไฟฟ้าหัวใจโดยใช้วิธีการแปลงเวฟเลต ได้แก่ การเลือกใช้เวฟเลตชนิดอื่นๆซึ่งมีลักษณะแตกต่างไปจาก Orthonormal Wavelet ที่ได้นำเสนอ เช่น Bi-Orthonormal Wavelet เป็นต้น การเลือกอัลกอริทึมการแปลงเวฟเลตแบบอื่นๆซึ่งแตกต่างไปจาก Octave Band Tree (Wavelet Series) ที่ได้นำเสนอ เช่น Arbitrary Band Tree (Wavelet Packet) หรือ Full Band Tree (Short Time Fourier Transform) เป็นต้น การเลือกใช้วิธีการแควนไตซ์แบบอื่นๆ เช่น การแควนไตซ์แบบนอนยูนิฟอร์ม เป็นต้น ตลอดจนวิธีการเข้ารหัสแบบอื่นๆ ซึ่งจะต้องทำการศึกษาและทำการวิจัยกันต่อไป

เอกสารอ้างอิง

- [1] GoldMan M. J. **Principles of Clinical Electrocariography**. 9th Ed. Tokyo : Maruzen. 1976.
- [2] Tompkins W. J. **Biomedical Digital Signal Processing : C-Language Examples and Laboratory Experiments for the IBM PC**. International Edition. New Jersey : Prentice Hall, Inc. 1993.
- [3] วัฒนา โพธิ์เจริญ. “การออกแบบระบบการวัดคลื่นไฟฟ้าหัวใจแบบรวมศูนย์.” วิทยานิพนธ์ วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2541.
- [4] Young R. K. **Wavelet Theory and Its Applications**. Massachusetts : Kluwer Academic Publishers. 1993.
- [5] Vidakovic B., Muller P. “Wavelets for Kids.” [Online]. Available : [ftp://ftp.isds.duke.edu/duke.edu/pub/brani/papers/wav4kids\[A\].ps.z](ftp://ftp.isds.duke.edu/duke.edu/pub/brani/papers/wav4kids[A].ps.z). 1994.
- [6] มานิตย์ เกียรติกำจายขจร, ประภากร สุวรรณะ และ มนัส สังวรศิลป์, “การทดสอบการประมาณข้อมูลโดยใช้การแปลงเวฟเล็ตเต็มหน่วยเทียบกับการแปลงโคซายน์เต็มหน่วย.” วารสารวิจัยและพัฒนา สจธ. ปีที่ 20, ฉบับที่ 2, ธันวาคม 2540. หน้า 79-96.
- [7] Fliege N. J. **Multirate Digital Signal Processing : Multirate Systems, Filter Banks and Wavelets**. Chichester : John Wiley & Sons. 1994.
- [8] Daubechies I. “Ten Lectures on Wavelets.” CBMS-NSF Regional Conference Series in Applied Mathematics, Vol.61. 1992. pp.195-199, pp.261.
- [9] บัณฑิต ไรจน์อารยานนท์. **หลักการไฟฟ้าสื่อสาร**. พิมพ์ครั้งที่ 3. กรุงเทพมหานคร : สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย. 2536.
- [10] Advantech. **PCL-812PG Enhanced Multi-Lab Card User's Manual**. Feb 1993.

ภาคผนวก ก

การคำนวณผลตอบสนองความถี่ของ Octave Band

Analysis Filter Bank โดยใช้โปรแกรม Matlab

ตัวอย่างโปรแกรม Matlab ที่ใช้ในการคำนวณผลตอบสนองความถี่ของ Octave Band Analysis Filter Bank 3 ระดับ โดยกำหนดให้ใช้เวฟเลตแบบ Daubechies-20 แสดงดังต่อไปนี้

```
% Assign the Phototype Daubechies-20 FIR Filter  $h(n)$  %
h = [ 0.026670057901, 0.188176800078, 0.527201188932, 0.688459039454, ...
      0.281172343661, -0.249846424327, -0.195946274377, 0.127369340336, ...
      0.093057364604, -0.071394147166, -0.029457536822, 0.033212674059, ...
      0.003606553567, -0.010733175483, 0.001395351747, 0.001992405295, ...
      -0.000685856695, -0.000116466855, 0.000093588670, -0.000013264203 ];

g = -((-1).^(1:length(h))) .* h; % Calculate the Dual Filter  $g(n)$ 
a0 = g; % Assign the FIR Filter  $a_0(n)$ 
[A0, f] = freqz(a0, 1, 1024, 1); % Calculate the Frequency Response of  $A_0(z)$ 
plot(f, abs(A0), 'w-'); hold on;
gtemp = zeros(1, 2*length(g)); gtemp(1:2:2*length(g)) = g;
htemp = zeros(1, 2*length(h)); htemp(1:2:2*length(h)) = h;
gh = conv(h, gtemp); hh = conv(h, htemp);
a1 = gh; % Assign the FIR Filter  $a_1(n)$ 
A1 = freqz(a1, 1, 1024, 1); % Calculate the Frequency Response of  $A_1(z)$ 
plot(f, abs(A1), 'w-');
gtemp = zeros(1, 4*length(g)); gtemp(1:4:4*length(g)) = g;
htemp = zeros(1, 4*length(h)); htemp(1:4:4*length(h)) = h;
ghh = conv(hh, gtemp); hhh = conv(hh, htemp);
a2 = ghh; % Assign the FIR Filter  $a_2(n)$ 
a3 = hhh; % Assign the FIR Filter  $a_3(n)$ 
A2 = freqz(a2, 1, 1024, 1); % Calculate the Frequency Response of  $A_2(z)$ 
plot(f, abs(A2), 'w-');
A3 = freqz(a3, 1, 1024, 1); % Calculate the Frequency Response of  $A_3(z)$ 
plot(f, abs(A3), 'w-'); axis([0 0.5 0 3.5]);
xlabel('Normalized Frequency(Nyquist Frequency = 0.5)');
ylabel('Magnitude Response');
hold off;
```

ภาคผนวก ข

ทฤษฎีค่าเฉลี่ยเชิงสถิติ

ในการวิเคราะห์ตัวแปรแรมดัม(Random Variable)ในเชิงสถิตินั้นมักจะใช้ค่าเฉลี่ยเชิงสถิติในการบอกคุณสมบัติของตัวแปรแรมดัมนั้น ในส่วนนี้จะกล่าวถึงรายละเอียดของค่าเฉลี่ยเชิงสถิติที่ใช้กันอยู่ทั่วไป

ค่าเฉลี่ยของตัวแปรแรมดัม

ในกรณีที่ตัวแปรแรมดัม X เป็นแบบที่มีค่าไม่ต่อเนื่อง เช่น มีค่าเป็น $x_1, x_2, x_3, \dots$ และความน่าจะเป็น(Probability)ที่จะเกิดแต่ละค่าเป็น $P(x_1), P(x_2), P(x_3), \dots$ ตามลำดับ ถ้าทำการทดลอง N ครั้งโดยที่ N มีค่าใหญ่มาก จะพบว่าโอกาสที่ $X = x_i$ นั้นจะมี $N \cdot P(x_i)$ ครั้ง ดังนั้นเมื่อนำผลที่ได้จากการทดลองทั้งหมดนี้มารวมกันก็จะได้เป็น

$$x_1 P(x_1) N + x_2 P(x_2) N + \dots = N \sum_i x_i P(x_i) \tag{ก}$$

และเมื่อนำจำนวนครั้งของการทดลองคือ N มาหารสมการ (ก) ก็จะได้ค่าเฉลี่ยของผลที่ได้จากการทดลองออกมา ค่าเฉลี่ยที่ได้นี้เราเรียกว่า ค่าคาดหวังของ X (Expectation of X) และมักจะใช้สัญลักษณ์เป็น $E[X]$ ดังนั้นถ้าให้ค่าเฉลี่ยที่ได้ออกมาเป็น m ก็จะแสดงความสัมพันธ์ได้ดังนี้

$$E[X] = m = \sum_i x_i P(x_i) \tag{ข}$$

ในกรณีที่ตัวแปรแรมดัม X เป็นแบบที่มีค่าต่อเนื่องและมี Probability Density Function เป็น $p_X(x)$ เราจะสามารถหาค่าเฉลี่ยหรือค่าคาดหวังได้โดยพิจารณาให้ $P(x_i)$ ในสมการ (ข) มีค่าเท่ากับ $p_X(x_i) \cdot \Delta x$ จากนั้นจึงเปลี่ยนรูปจากการหาผลบวกเป็นการอินทิเกรตได้ดังนี้คือ

$$E[X] = m = \int_{-\infty}^{\infty} x \cdot p_X(x) dx \tag{ค}$$

ความเบี่ยงเบนมาตรฐานและความแปรปรวน(Standard Deviation and Variance)

ความเบี่ยงเบนมาตรฐานเป็นค่าที่ใช้แสดงการกระจายของตัวแปรแรมคัมว่าจะมีกระจายs กว้างหรือเกาะกลุ่มแคบ ถ้าความเบี่ยงเบนมาตรฐานซึ่งมักใช้สัญลักษณ์ σ มีนิยามคือ

$$\sigma_x^2 = \int_{-\infty}^{\infty} (x-m)^2 p_x(x) dx \quad (ง)$$

กำลังสองของ σ ที่นิยามตามสมการ (ค) นี้เรียกว่าความแปรปรวน(Variance) จากนิยามนี้จะเห็นได้ชัดว่า ถ้า X กระจายห่างออกจากค่าเฉลี่ย m มากเท่าใด ความแปรปรวนหรือความเบี่ยงเบนมาตรฐานก็จะสูงตามไปด้วย ดังนั้นถ้ากล่าวว่ ตัวแปรแรมคัมชุดหนึ่งมีความเบี่ยงเบนมาตรฐานหรือความแปรปรวนต่ำกว่าอีกชุดหนึ่ง ก็แสดงว่าตัวแปรแรมคัมชุดแรกมีการกระจายออกจากค่าเฉลี่ยน้อยกว่าชุดหลัง และเมื่อทำการแตกเทอมทางขวามือของสมการ (ค) จะได้ผลดังนี้

$$\sigma_x^2 = E[X^2 - 2mX + m^2] = E[X^2] - m^2 \quad (จ)$$

จะเห็นได้ว่าค่าความแปรปรวนจะหาได้จากค่ากำลังสองเฉลี่ยลบด้วยค่าเฉลี่ยยกกำลังสอง และในกรณีที่ค่าเฉลี่ย m เป็นศูนย์ σ_x^2 จะเท่ากับ $E[X^2]$

ภาคผนวก ก

โปรแกรมภาษาซีของระบบบันทึกสัญญาณไฟฟ้าหัวใจ โดยใช้วิธีการแปลงเวฟเล็ดในการลดขนาดข้อมูล

โปรแกรมภาษาซีของระบบบันทึกสัญญาณไฟฟ้าหัวใจโดยใช้วิธีการแปลงเวฟเล็ดในการลดขนาดข้อมูลแสดงได้ดังต่อไปนี้

MAIN.C

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <process.h>
#include <bios.h>
#include <conio.h>
#include <dos.h>
#include <dir.h>
#include <string.h>
#include <graphics.h>

#define N_MAIN_MENU 5
#define N_FILE 200
#define X1 28
#define Y1 22
#define X_LENGTH 512
#define Y_LENGTH 430
#define X2 X1 + X_LENGTH - 1
#define Y2 Y1 + Y_LENGTH - 1
#define X1_AD_WARN_MENU (X_LENGTH-426)/2 - 1
#define Y1_AD_WARN_MENU (Y_LENGTH-236)/2 - 1
#define X1_AD_INIT_FAIL (X_LENGTH-270)/2 - 1
#define Y1_AD_INIT_FAIL (Y_LENGTH-60)/2 - 1
#define X1_AD_START_FAIL (X_LENGTH-428)/2 - 1
#define Y1_AD_START_FAIL (Y_LENGTH-60)/2 - 1
#define X1_AD_STOP_FAIL (X_LENGTH-470)/2 - 1
#define Y1_AD_STOP_FAIL (Y_LENGTH-60)/2 - 1
#define X1_RECORD_SCHEME (X_LENGTH-300)/2 - 1
#define Y1_RECORD_SCHEME (Y_LENGTH-60)/2 - 1
#define X1_FILE_MENU 143
#define Y1_FILE_MENU 10
```

```

#define X1_READ_SCHEME      (X_LENGTH-350)/2 - 1
#define Y1_READ_SCHEME      (Y_LENGTH-60)/2 - 1
#define X1_ABOUT             (X_LENGTH-460)/2 - 1
#define Y1_ABOUT             (Y_LENGTH-236)/2 - 1
#define UP_ARROW             0x4800
#define PAGE_UP              0x4900
#define LEFT_ARROW           0x4B00
#define RIGHT_ARROW          0x4D00
#define DOWN_ARROW           0x5000
#define PAGE_DOWN            0x5100
#define RETURN               0x1C0D
#define ESCAPE                0x011B
#define N_Data                2048
#define N_QMF                 18

typedef struct {
    char patient_name[11], dura[5], filename[9];
} HEAD;

void Initial_Graphics_Mode(void);
void Main_Menu_Assignment(void);
void Display_Console(void);
void Display_Scale(void);
void Window(unsigned int x1, unsigned int y1, unsigned int x2, unsigned int y2,
    unsigned int pattern, unsigned int color, unsigned int framecolor);
void Select_Main_Menu(unsigned int main_menu_no);
void Unselect_Main_Menu(unsigned int main_menu_no);
void AD_Card_Jumper_Set_Warning(unsigned int main_menu_no);
void Select_Warning_Menu(unsigned int warn_menu_no, unsigned int main_menu_no);
void Unselect_Warning_Menu(unsigned int warn_menu_no);
void Initial_AD_Card(void);
void Start_AD_Card(void);
void Stop_AD_Card(void);
void Monitor(void);
void Get_Head_File(void);
void Get_Record_Duration(void);
void Get_File_Name(void);
void Record(void);
void Open_Read(void);
void Display_File_Menu(unsigned int start);
void Select_File_Menu(unsigned int start, unsigned int file_no);
void Unselect_File_Menu(unsigned int start, unsigned int file_no);
void Read(unsigned int start, unsigned int file_no);
void Clear_Head_Array(void);
void Display_Head(void);

```

```

void Clear_Head_Display(void);
void About(void);
void Display_Signal(unsigned char sw);
void Display_Time(void);
void Display_Percent_Complete(void);
extern pcl812(int, unsigned int *);
extern void ECG_Compress(void);          /* Call to ECGCOMP.C */
extern void ECG_Reconstruct(void);       /* Call to ECGRECON.C */
struct main_menu_struct {
    unsigned int row;
    unsigned int col;
    unsigned int width;
    char *item;
} main_menu[N_MAIN_MENU];
double tmp;
HEAD *head;
struct ffbk ffbk;
unsigned char i_head, j_head;
char *ptr, key, filename[13], filename_list[N_FILE][13];
unsigned int i, j, k, ch_pos, copy, n_block, x, xx, key_main_menu, key_wam_menu, key_file_menu, n_file_exist;
unsigned int param[60], Data_A[N_Data], Data_B[N_Data], *Ptr_Data_A, *Ptr_Data_B, *fptr_pos_array;
float Data[N_Data], *Ptr_Data, *Ptr_QMF, temp, y, yy;
float QMF[N_QMF] = {
    -0.003793512864,    0.007782596426,    0.023452696142,
    -0.065771911281,   -0.061123390003,    0.405176902410,
    0.793777222626,    0.428483476378,   -0.071799821619,
    -0.082301927106,    0.034555027573,    0.015880544864,
    -0.009007976137,   -0.002574517688,    0.001117518771,
    0.000466216960,   -0.000070983303,   -0.000034599773    };
unsigned int absmax, n_IQset, n_Data_RLCout, n_code, adj, len, *histo_Data, *histo_Data_encode, *index;
int *IQset, *Iqencode, done, dec, sign;
unsigned int *Iqencode_Stream;
unsigned int *huffcval, *full_huffcval, *Data_huffcval;
unsigned int *huffcbit, *full_huffcbit, *Data_huffcbit;
FILE *fptr_wh, *fptr_rh, *fptr_w, *fptr_r;
fpos_t start, pos, end;
void main(void) {
    Initial_Graphics_Mode();
    Main_Menu_Assignment();
    Display_Console();
    About();
    Select_Main_Menu(0);
}

```

```

outtextxy(X1+218, Y2+16, "160 mS/Div");
for (i=0; i<Y_LENGTH/5-5; i++) line(X1-5, Y2-5*i, X1-3, Y2-5*i);
for (i=0; i<Y_LENGTH/25; i++) line(X1-6, Y2-25*i, X1-9, Y2-25*i);
settextstyle(DEFAULT_FONT, HORIZ_DIR, 0);
outtextxy(X1-25, Y1+Y_LENGTH/2+136, "-1");
outtextxy(X1-19, Y1+Y_LENGTH/2+11, "0");
outtextxy(X1-19, Y1+Y_LENGTH/2-114, "1");
outtextxy(X1-22, Y1+10, "mV");
Display_Scale();
setcolor(WHITE);
outtextxy(X2+19, Y1-2, "Filename");
Window(X2+10, Y1+10, getmaxx()-8, Y1+26, BLACK, WHITE, LIGHTRED);
setcolor(WHITE);
outtextxy(X2+24, Y1+48, "Patient");
Window(X2+10, Y1+60, getmaxx()-8, Y1+76, BLACK, WHITE, LIGHTGREEN);
setcolor(WHITE);
outtextxy(X2+19, Y1+98, "Duration");
Window(X2+10, Y1+110, getmaxx()-8, Y1+126, BLACK, WHITE, YELLOW);
setcolor(WHITE);
outtextxy(X2+44, Y2-22, "0%");
Window(X2+10, Y2-10, getmaxx()-8, Y2, SOLID_FILL, BLACK, LIGHTCYAN);
setcolor(LIGHTGRAY);
line(X1, Y2+10, X1, Y2+19);
line(X1+1, Y2+19, X1+5, Y2+19);
line(X2+1, Y2+10, X2+1, Y2+19);
line(X2-4, Y2+19, X2, Y2+19);
setfillstyle(SOLID_FILL, LIGHTGREEN);
bar(X1+6, Y2+15, X1+96, Y2+23);
bar(X2-95, Y2+15, X2-5, Y2+23);
}

void Display_Scale(void) {
    register unsigned int i;
    setcolor(DARKGRAY);
    for (i=1; i<X_LENGTH/32; i++) line(X1+32*i, Y1, X1+32*i, Y2);
    for (i=1; i<Y_LENGTH/25+1; i++) line(X1, Y2-25*i, X2, Y2-25*i);
    setcolor(LIGHTGRAY);
    line(X1, 251, X2, 251);
}

void Window(unsigned int x1, unsigned int y1, unsigned int x2, unsigned int y2, unsigned int pattern, unsigned int color,
            unsigned int framecolor) {
    setfillstyle(pattern, color);
    bar(x1, y1, x2, y2);
    setcolor(framecolor);

```

```

rectangle(x1-2, y1-2, x2+2, y2+2);
setcolor(framecolor);
rectangle(x1-1, y1-1, x2+1, y2+1);
}

void Select_Main_Menu(unsigned int main_menu_no) {
    setfillstyle(SOLID_FILL, YELLOW);
    bar(main_menu[main_menu_no].col, main_menu[main_menu_no].row - 6,
        main_menu[main_menu_no].col + main_menu[main_menu_no].width,
        main_menu[main_menu_no].row + 12);
    setcolor(BLACK);
    outtextxy(main_menu[main_menu_no].col, main_menu[main_menu_no].row, main_menu[main_menu_no].item);
    while (1) {
        key_main_menu = bioskey(0);
        switch (key_main_menu) {
            case LEFT_ARROW :
                if (main_menu_no == 0) {
                    Unselect_Main_Menu(0);
                    main_menu_no = N_MAIN_MENU - 1;
                }
                else Unselect_Main_Menu(main_menu_no--);
                Select_Main_Menu(main_menu_no);
            case RIGHT_ARROW :
                if (main_menu_no == N_MAIN_MENU - 1) {
                    Unselect_Main_Menu(main_menu_no);
                    main_menu_no = 0;
                }
                else Unselect_Main_Menu(main_menu_no++);
                Select_Main_Menu(main_menu_no);
            case RETURN :
                switch (main_menu_no) {
                    case 0 :
                        AD_Card_Jumper_Set_Warning(main_menu_no);
                        Initial_AD_Card();
                        Start_AD_Card();
                        Monitor();
                        Stop_AD_Card();
                        Select_Main_Menu(0);
                    case 1 :
                        AD_Card_Jumper_Set_Warning(main_menu_no);
                        Initial_AD_Card();
                        Get_Head_File();
                        Record();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Select_Main_Menu(1);

    case 2 :

        Open_Read();

        break;

    case 3 :

        About();

        Select_Main_Menu(3);

    case 4 :

        closegraph();

        exit(0);

    }

    break;

case ESCAPE :

    closegraph();

    exit(0);

}

}

}

void Unselect_Main_Menu(unsigned int main_menu_no) {
    setfillstyle(SOLID_FILL, BLUE);
    bar(main_menu[main_menu_no].col, main_menu[main_menu_no].row - 6,
        main_menu[main_menu_no].col + main_menu[main_menu_no].width,
        main_menu[main_menu_no].row + 12);
    setcolor(WHITE);
    outtextxy(main_menu[main_menu_no].col, main_menu[main_menu_no].row, main_menu[main_menu_no].item);
}

void AD_Card_Jumper_Set_Warning(unsigned int main_menu_no) {
    setviewport(X1, Y1, X2, Y2, BLACK);
    setfillstyle(SOLID_FILL, WHITE);
    bar(X1_AD_WARN_MENU, Y1_AD_WARN_MENU, X1_AD_WARN_MENU+425, Y1_AD_WARN_MENU+235);
    setcolor(LIGHTRED);
    rectangle(X1_AD_WARN_MENU+5, Y1_AD_WARN_MENU+5, X1_AD_WARN_MENU+420,
        Y1_AD_WARN_MENU+230);
    rectangle(X1_AD_WARN_MENU+7, Y1_AD_WARN_MENU+7, X1_AD_WARN_MENU+418,
        Y1_AD_WARN_MENU+228);
    setcolor(WHITE);
    line(X1_AD_WARN_MENU+177, Y1_AD_WARN_MENU+5, X1_AD_WARN_MENU+248,
        Y1_AD_WARN_MENU+5);
    line(X1_AD_WARN_MENU+177, Y1_AD_WARN_MENU+7, X1_AD_WARN_MENU+248,
        Y1_AD_WARN_MENU+7);
    setcolor(LIGHTRED);
    outtextxy(X1_AD_WARN_MENU+185, Y1_AD_WARN_MENU+2, "Warning");
    setcolor(LIGHTBLUE);

```

```

outtextxy(X1_AD_WARN_MENU+20, Y1_AD_WARN_MENU+23,
    "This program needs PCL-812PG A/D card setting as");
outtextxy(X1_AD_WARN_MENU+20, Y1_AD_WARN_MENU+38, "following.");
setcolor(GREEN);
outtextxy(X1_AD_WARN_MENU+28, Y1_AD_WARN_MENU+68, "Trigger mode (JP1)      : Internal");
outtextxy(X1_AD_WARN_MENU+28, Y1_AD_WARN_MENU+83, "IRQ level (JP5)          : 2");
outtextxy(X1_AD_WARN_MENU+28, Y1_AD_WARN_MENU+98, "DMA level (JP6 & JP7)       : 1");
outtextxy(X1_AD_WARN_MENU+28, Y1_AD_WARN_MENU+113, "A/D input range (JP9)    : +/- 10V");
outtextxy(X1_AD_WARN_MENU+28, Y1_AD_WARN_MENU+128, "I/O port base address (SW1) : 0x220");
outtextxy(X1_AD_WARN_MENU+28, Y1_AD_WARN_MENU+143, "A/D input Channel No.   : 0");
setcolor(LIGHTRED);
outtextxy(X1_AD_WARN_MENU+115, Y1_AD_WARN_MENU+173, "Is the setting correct ?");
setcolor(BLACK);
outtextxy(X1_AD_WARN_MENU+127, Y1_AD_WARN_MENU+200, "Yes");
outtextxy(X1_AD_WARN_MENU+276, Y1_AD_WARN_MENU+200, "No");
Select_Warning_Menu(0, main_menu_no);
}

void Select_Warning_Menu(unsigned int warn_menu_no, unsigned int main_menu_no) {
    if (warn_menu_no == 0) {
        setfillstyle(SOLID_FILL, YELLOW);
        bar(X1_AD_WARN_MENU+113, Y1_AD_WARN_MENU+194, X1_AD_WARN_MENU+163,
            Y1_AD_WARN_MENU+214);
        setcolor(BLACK);
        outtextxy(X1_AD_WARN_MENU+127, Y1_AD_WARN_MENU+200, "Yes");
    }
    else {
        setfillstyle(SOLID_FILL, YELLOW);
        bar(X1_AD_WARN_MENU+258, Y1_AD_WARN_MENU+194, X1_AD_WARN_MENU+308,
            Y1_AD_WARN_MENU+214);
        setcolor(BLACK);
        outtextxy(X1_AD_WARN_MENU+276, Y1_AD_WARN_MENU+200, "No");
    }
}

while (1) {
    key_warn_menu = bioskey(0);
    if ((key_warn_menu != LEFT_ARROW) && (key_warn_menu != RIGHT_ARROW) &&
        (key_warn_menu != RETURN) && (key_warn_menu != ESCAPE)) continue;
    switch (key_warn_menu) {
        case LEFT_ARROW :
            if (warn_menu_no == 0) {
                Unselect_Warning_Menu(warn_menu_no);
                warn_menu_no = 1;
            }

```

```

        Unselect_Warning_Menu(warn_menu_no);
        warn_menu_no = 0;
    }
    Select_Warning_Menu(warn_menu_no, main_menu_no);
    break;
case RIGHT_ARROW :
    if (warn_menu_no==1) {
        Unselect_Warning_Menu(warn_menu_no);
        warn_menu_no = 0;
    }
    else {
        Unselect_Warning_Menu(warn_menu_no);
        warn_menu_no = 1;
    }
    Select_Warning_Menu(warn_menu_no, main_menu_no);
    break;
case RETURN :
    switch (warn_menu_no) {
        case 0 :
            setfillstyle(SOLID_FILL, BLACK);
            clearviewport();
            setviewport(0, 0, 639, 479, BLACK);
            Display_Scale();
            break;
        case 1 :
            setfillstyle(SOLID_FILL, WHITE);
            bar(X1_AD_WARN_MENU+9, Y1_AD_WARN_MENU+158, X1_AD_WARN_MENU+416,
                Y1_AD_WARN_MENU+227);
            setcolor(LIGHTBLUE);
            outtextxy(X1_AD_WARN_MENU+20, Y1_AD_WARN_MENU+178,
                "Set to correct this configuration before running");
            outtextxy(X1_AD_WARN_MENU+20, Y1_AD_WARN_MENU+193, "this program !");
            getch();
            closegraph();
            exit(0);
    }
    break;
case ESCAPE :
    clearviewport();
    setviewport(0, 0, 639, 479, BLACK);
    Display_Scale();
    Select_Main_Menu(main_menu_no);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        break;
    }
}

void Unselect_Warning_Menu(unsigned int warn_menu_no) {
    if (warn_menu_no==0) {
        setfillstyle(SOLID_FILL, WHITE);
        bar(X1_AD_WARN_MENU+113, Y1_AD_WARN_MENU+194, X1_AD_WARN_MENU+163,
            Y1_AD_WARN_MENU+214);
        setcolor(BLACK);
        outtextxy(X1_AD_WARN_MENU+127, Y1_AD_WARN_MENU+200, "Yes");
    }
    else {
        setfillstyle(SOLID_FILL, WHITE);
        bar(X1_AD_WARN_MENU+258, Y1_AD_WARN_MENU+194, X1_AD_WARN_MENU+308,
            Y1_AD_WARN_MENU+214);
        setcolor(BLACK);
        outtextxy(X1_AD_WARN_MENU+276, Y1_AD_WARN_MENU+200, "No");
    }
}

void Initial_AD_Card(void) {
    Ptr_Data_A = Data_A;
    Ptr_Data_B = Data_B;
    param[0] = 0; /* Card Number */
    param[1] = 0x220; /* Base I/O Address */
    param[2] = 1; /* DMA Channel of Buffer A */
    param[3] = 1; /* DMA Channel of Buffer B */
    param[4] = 2; /* IRQ Level 2 */
    param[5] = 100; param[6] = 100; /* CI and C2 : Pacer Rate = 2MHz / (C1*C2) */
    param[7] = 0; /* Internal Trigger */
    param[8] = 1; /* Cyclic Conversion Mode */
    param[10] = FP_OFF(Ptr_Data_A); /* Offset of A/D Data Buffer A */
    param[11] = FP_SEG(Ptr_Data_A); /* Segment of A/D Data Buffer A */
    param[12] = FP_OFF(Ptr_Data_B); /* Offset of A/D Data Buffer B */
    param[13] = FP_SEG(Ptr_Data_B); /* Segment of A/D Data Channel */
    param[14] = N_Data; /* A/D Conversion Number */
    param[15] = 0; /* A/D Conversion Start Channel */
    param[16] = 0; /* A/D Conversion Stop Channel */
    param[17] = 0; /* Overall Gain Code : +/- 10V */
    pcl812(3, param); /* Hardware Initialization */
    if (param[45]!=0) {
        setviewport(X1, Y1, X2, Y2, BLACK);
        setfillstyle(SOLID_FILL, WHITE);
        bar(X1_AD_INIT_FAIL, Y1_AD_INIT_FAIL, X1_AD_INIT_FAIL+269, Y1_AD_INIT_FAIL+59);
    }
}

```

```

        Y1_AD_START_FAIL+53);
setcolor(WHITE);
line(X1_AD_START_FAIL+139, Y1_AD_START_FAIL+5, X1_AD_START_FAIL+288,
     Y1_AD_START_FAIL+5);
line(X1_AD_START_FAIL+139, Y1_AD_START_FAIL+7, X1_AD_START_FAIL+288,
     Y1_AD_START_FAIL+7);
setcolor(LIGHTRED);
outtextxy(X1_AD_START_FAIL+146, Y1_AD_START_FAIL+2, "Program Terminate");
setcolor(LIGHTRED);
outtextxy(X1_AD_START_FAIL+15, Y1_AD_START_FAIL+27,
          "Pacer trigger with interrupt DMA transfer failed !");
getch();
closegraph();
exit(1);
}
}

void Stop_AD_Card(void) {
    pcl812(8, param);          /* Stop Pacer Trigger A/D Conversion */
    if (param[45]!=0) {
        setviewport(X1, Y1, X2, Y2, BLACK);
        setfillstyle(SOLID_FILL, WHITE);
        bar(X1_AD_STOP_FAIL, Y1_AD_STOP_FAIL, X1_AD_STOP_FAIL+467, Y1_AD_STOP_FAIL+59);
        setcolor(LIGHTRED);
        rectangle(X1_AD_STOP_FAIL+5, Y1_AD_STOP_FAIL+5, X1_AD_STOP_FAIL+462, Y1_AD_STOP_FAIL+54);
        rectangle(X1_AD_STOP_FAIL+7, Y1_AD_STOP_FAIL+7, X1_AD_STOP_FAIL+460, Y1_AD_STOP_FAIL+52);
        setcolor(WHITE);
        line(X1_AD_STOP_FAIL+159, Y1_AD_STOP_FAIL+5, X1_AD_STOP_FAIL+308, Y1_AD_STOP_FAIL+5);
        line(X1_AD_STOP_FAIL+159, Y1_AD_STOP_FAIL+7, X1_AD_STOP_FAIL+308, Y1_AD_STOP_FAIL+7);
        setcolor(LIGHTRED);
        outtextxy(X1_AD_STOP_FAIL+166, Y1_AD_STOP_FAIL+2, "Program Terminate");
        setcolor(LIGHTRED);
        outtextxy(X1_AD_STOP_FAIL+15, Y1_AD_STOP_FAIL+27,
                  "Stop pacer trigger with interrupt DMA transfer failed !");
        getch();
        closegraph();
        exit(1);
    }
}

void Monitor(void) {
    while (1) {
        key = bioskey(2);

        if ((key!=4)&&(key!=20)&&(key!=36)&&(key!=52)&&(key!=68)&&(key!=84)&&(key!=100)&&(key!=116)) {
            do {

```

```

        pcl812(7, param);
    } while (param[46]==1);
    Display_Signal(0);
}
else break;
key = bioskey(2);
if ((key!=4)&&(key!=20)&&(key!=36)&&(key!=52)&&(key!=68)&&(key!=84)&&(key!=100)&&(key!=116)) {
    do {
        pcl812(7, param);
    } while (param[46]==3);
    Display_Signal(0);
}
else break;
}
setfillstyle(SOLID_FILL, BLACK);
bar(X1, Y1, X2, Y2);
Display_Scale();
}
void Get_Head_File(void) {
    Clear_Head_Array();
    setviewport(X1, Y1, X2, Y2, BLACK);
    setfillstyle(SOLID_FILL, WHITE);
    bar(X1_RECORD_SCHEME+30, Y1_RECORD_SCHEME, X1_RECORD_SCHEME+299-30,
        Y1_RECORD_SCHEME+59);
    setcolor(BLACK);
    rectangle(X1_RECORD_SCHEME+35, Y1_RECORD_SCHEME+5, X1_RECORD_SCHEME+264,
        Y1_RECORD_SCHEME+54);
    rectangle(X1_RECORD_SCHEME+37, Y1_RECORD_SCHEME+7, X1_RECORD_SCHEME+262,
        Y1_RECORD_SCHEME+52);
    setfillstyle(SOLID_FILL, LIGHTBLUE);
    bar(X1_RECORD_SCHEME+164, Y1_RECORD_SCHEME+23, X1_RECORD_SCHEME+250,
        Y1_RECORD_SCHEME+37);
    setcolor(BLACK);
    outtextxy(X1_RECORD_SCHEME+48, Y1_RECORD_SCHEME+27, "Patient name : ");
    moveto(X1_RECORD_SCHEME+168, Y1_RECORD_SCHEME+27);
    i_head = 0;
    ch_pos = X1_RECORD_SCHEME+168;
    while (1) {
        if (i_head!=10) {
            setcolor(WHITE);
            line(ch_pos, Y1_RECORD_SCHEME+35, ch_pos+6, Y1_RECORD_SCHEME+35);
        }
        key = getch();

```

```

if ((i_head==0)&&((key=='b')||(key==13))||(key==' ')||(key=='.')||(key=='_')) continue;
else if ((i_head!=10)&&(((key>='a')&&(key<='z'))||(key>='A')&&(key<='Z'))||(key>='0')&&(key<='9'))||
    (key==' ')||(key=='.')||(key=='_')) {
    head->patient_name[i_head++] = key;
    setcolor(LIGHTBLUE);
    line(ch_pos, Y1_RECORD_SCHEME+35, ch_pos+6, Y1_RECORD_SCHEME+35);
    ch_pos += 8;
    setcolor(YELLOW);
    outtext(&key);
}
else if (key=='\b') {
    head->patient_name[--i_head] = '\0';
    moverel(-8, 0);
    setfillstyle(SOLID_FILL, LIGHTBLUE);
    bar(getx(), gety(), getx()+7, gety()+7);
    setcolor(LIGHTBLUE);
    if (i_head!=9) line(ch_pos, Y1_RECORD_SCHEME+35, ch_pos+6, Y1_RECORD_SCHEME+35);
    ch_pos -= 8;
}
else if (key==13) {
    for (j_head = i_head; j_head<11; j_head++) head->patient_name[j_head] = '\0';
    Get_Record_Duration();
    break;
}
else if (key==27) {
    Clear_Head_Array();
    clearviewport();
    setviewport(0, 0, 639, 479, BLACK);
    Display_Scale();
    Select_Main_Menu(1);
}
}
}

void Get_Record_Duration(void) {
    setfillstyle(SOLID_FILL, WHITE);
    bar(X1_RECORD_SCHEME, Y1_RECORD_SCHEME, X1_RECORD_SCHEME+299, Y1_RECORD_SCHEME+59);
    setcolor(BLACK);
    rectangle(X1_RECORD_SCHEME+5, Y1_RECORD_SCHEME+5, X1_RECORD_SCHEME+294,
        Y1_RECORD_SCHEME+54);
    rectangle(X1_RECORD_SCHEME+7, Y1_RECORD_SCHEME+7, X1_RECORD_SCHEME+292,
        Y1_RECORD_SCHEME+52);
    setfillstyle(SOLID_FILL, LIGHTGREEN);
    bar(X1_RECORD_SCHEME+159, Y1_RECORD_SCHEME+23, X1_RECORD_SCHEME+181,

```

```

Y1_RECORD_SCHEME+37);
setfillstyle(SOLID_FILL, LIGHTGREEN);
bar(X1_RECORD_SCHEME+223, Y1_RECORD_SCHEME+23, X1_RECORD_SCHEME+245,
    Y1_RECORD_SCHEME+37);
setcolor(BLACK);
outtextxy(X1_RECORD_SCHEME+19, Y1_RECORD_SCHEME+27, "Record duration : Hrs. Min.");
moveto(X1_RECORD_SCHEME+163, Y1_RECORD_SCHEME+27);
i_head = 0;
ch_pos = X1_RECORD_SCHEME+163;
while (1) {
    if (i_head!=4) {
        setcolor(BLACK);
        line(ch_pos, Y1_RECORD_SCHEME+35, ch_pos+6, Y1_RECORD_SCHEME+35);
    }
    key = getch();
    if (i_head==0) {
        if ((key=='\b')||(key==13)) continue;
        else if (key==27) {
            Clear_Head_Array();
            clearviewport();
            setviewport(0, 0, 639, 479, BLACK);
            Display_Scale();
            Select_Main_Menu(1);
            break;
        }
        else if ((key>='0')&&(key<='2')) {
            head->dura[i_head++] = key;
            setcolor(LIGHTGREEN);
            line(ch_pos, Y1_RECORD_SCHEME+35, ch_pos+6, Y1_RECORD_SCHEME+35);
            ch_pos += 8;
            setcolor(LIGHTRED);
            outtextt(&key);
        }
    }
}

else if (i_head==1) {
    if (key==13) continue;
    else if (key==27) {
        Clear_Head_Array();
        clearviewport();
        setviewport(0, 0, 639, 479, BLACK);
        Display_Scale();
        Select_Main_Menu(1);
        break;
    }
}

```

เอกสารนี้เป็นเอกสารที่สําคัญสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

else if ((head->dura[0]=='2')&&((key>='0')&&(key<='4')) {
    head->dura[i_head++] = key;
    setcolor(LIGHTGREEN);
    line(ch_pos, Y1_RECORD_SCHEME+35, ch_pos+6, Y1_RECORD_SCHEME+35);
    ch_pos += 56;
    setcolor(LIGHTRED);
    outtext(&key);
    moverel(48, 0);
}

else if ((head->dura[0]!='2')&&((key>='0')&&(key<='9')) {
    head->dura[i_head++] = key;
    setcolor(LIGHTGREEN);
    line(ch_pos, Y1_RECORD_SCHEME+35, ch_pos+6, Y1_RECORD_SCHEME+35);
    ch_pos += 56;
    setcolor(LIGHTRED);
    outtext(&key);
    moverel(48, 0);
}

else if (key=='\b') {
    head->dura[--i_head] = '\0';
    setcolor(LIGHTGREEN);
    line(ch_pos, Y1_RECORD_SCHEME+35, ch_pos+6, Y1_RECORD_SCHEME+35);
    moverel(-8, 0);
    setfillstyle(SOLID_FILL, LIGHTGREEN);
    bar(getx(), gety(), getx()+7, gety()+7);
    ch_pos -= 8;
}

}

else if (i_head==2) {
    if (key==13) continue;
    else if (key==27) {
        Clear_Head_Array();
        clearviewport();
        setviewport(0, 0, 639, 479, BLACK);
        Display_Scale();
        Select_Main_Menu(1);
        break;
    }

    else if ((head->dura[0]=='2')&&(head->dura[1]=='4')&&((key>='0')&&(key <= '5')) {
        head->dura[i_head++] = '0';
        setcolor(LIGHTGREEN);
        line(ch_pos, Y1_RECORD_SCHEME+35, ch_pos+6, Y1_RECORD_SCHEME+35);

```

```

        ch_pos += 8;
        setcolor(LIGHTRED);
        outtext("0");
    }
    else if ((key>='0')&&(key<='5')) {
        head->dura[i_head++] = key;
        setcolor(LIGHTGREEN);
        line(ch_pos, Y1_RECORD_SCHEME+35, ch_pos+6, Y1_RECORD_SCHEME+35);
        ch_pos += 8;
        setcolor(LIGHTRED);
        outtext(&key);
    }
    else if (key=='\b') {
        head->dura[--i_head] = '\0';
        setcolor(LIGHTGREEN);
        line(ch_pos, Y1_RECORD_SCHEME+35, ch_pos+6, Y1_RECORD_SCHEME+35);
        moverel(-56, 0);
        setfillstyle(SOLID_FILL, LIGHTGREEN);
        bar(getx(), gety(), getx()+7, gety()+7);
        ch_pos -= 56;
    }
}

else if (i_head==3) {
    if (key==13) continue;
    else if (key==27) {
        Clear_Head_Array();
        clearviewport();
        setviewport(0, 0, 639, 479, BLACK);
        Display_Scale();
        Select_Main_Menu(1);
        break;
    }
    else if ((head->dura[0]=='0')&&(head->dura[1]=='0')&&(head->dura[2]=='0')&&(key=='0')) {
        head->dura[i_head++] = '1';
        setcolor(LIGHTGREEN);
        line(ch_pos, Y1_RECORD_SCHEME+35, ch_pos+6, Y1_RECORD_SCHEME+35);
        ch_pos += 8;
        setcolor(LIGHTRED);
        outtext("1");
    }
    else if ((head->dura[0]=='2')&&(head->dura[1]=='4')&&((key>='0')&&(key<='9')) {
        head->dura[i_head++] = '0';
        setcolor(LIGHTGREEN);

```

```

        line(ch_pos, Y1_RECORD_SCHEME+35, ch_pos+6, Y1_RECORD_SCHEME+35);
        ch_pos += 8;
        setcolor(LIGHTRED);
        outtext("0");
    }
    else if ((key>='0')&&(key<='9')) {
        head->dura[i_head++] = key;
        setcolor(LIGHTGREEN);
        line(ch_pos, Y1_RECORD_SCHEME+35, ch_pos+6, Y1_RECORD_SCHEME+35);
        ch_pos += 8;
        setcolor(LIGHTRED);
        outtext(&key);
    }
    else if (key=='\b') {
        head->dura[--i_head] = '\0';
        setcolor(LIGHTGREEN);
        line(ch_pos, Y1_RECORD_SCHEME+35, ch_pos+6, Y1_RECORD_SCHEME+35);
        moverel(-8, 0);
        setfillstyle(SOLID_FILL, LIGHTGREEN);
        bar(getx(), gety(), getx()+7, gety()+7);
        ch_pos -= 8;
    }
}
else if (i_head==4) {
    if (key==27) {
        Clear_Head_Array();
        clearviewport();
        setviewport(0, 0, 639, 479, BLACK);
        Display_Scale();
        Select_Main_Menu(1);
        break;
    }
    else if (key==13) {
        head->dura[4] = '\0';
        Get_File_Name();
        break;
    }
    else if (key=='\b') {
        head->dura[--i_head] = '\0';
        moverel(-8, 0);
        setfillstyle(SOLID_FILL, LIGHTGREEN);
        bar(getx(), gety(), getx()+7, gety()+7);
        ch_pos -= 8;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}
}

void Get_File_Name(void) {
    setfillstyle(SOLID_FILL, WHITE);
    bar(X1_RECORD_SCHEME+8, Y1_RECORD_SCHEME+8, X1_RECORD_SCHEME+291,
        Y1_RECORD_SCHEME+51);
    setfillstyle(SOLID_FILL, LIGHTRED);
    bar(X1_RECORD_SCHEME+174, Y1_RECORD_SCHEME+23, X1_RECORD_SCHEME+244,
        Y1_RECORD_SCHEME+37);
    setcolor(BLACK);
    outtextxy(X1_RECORD_SCHEME+18, Y1_RECORD_SCHEME+27, "Save as file name :      .ecg");
    moveto(X1_RECORD_SCHEME+178, Y1_RECORD_SCHEME+27);
    i_head = 0;
    ch_pos = X1_RECORD_SCHEME+178;
    while (1) {
        if (i_head!=8) {
            setcolor(BLACK);
            line(ch_pos, Y1_RECORD_SCHEME+35, ch_pos+6, Y1_RECORD_SCHEME+35);
        }
        key = getch();
        if ((i_head==0)&&((key=='b')||(key==13))) continue;
        else if ((i_head!=8)&&(((key>='a')&&(key<='z'))||((key>='A')&&(key<='Z'))||((key>='0')&&(key<='9'))||(key==95)))) {
            head->filename[i_head++] = key;
            setcolor(LIGHTRED);
            line(ch_pos, Y1_RECORD_SCHEME+35, ch_pos+6, Y1_RECORD_SCHEME+35);
            ch_pos += 8;
            setcolor(BLUE);
            outtext(&key);
        }
        else if (key=='b') {
            head->filename[--i_head] = '\0';
            moverel(-8, 0);
            setfillstyle(SOLID_FILL, LIGHTRED);
            bar(getx(), gety(), getx()+7, gety()+7);
            setcolor(LIGHTRED);
            if (i_head!=7) line(ch_pos, Y1_RECORD_SCHEME+35, ch_pos+6, Y1_RECORD_SCHEME+35);
            ch_pos -= 8;
        }
        else if (key==13) {
            for (j_head = i_head; j_head<9; j_head++) head->filename[j_head] = '\0';

```

```

        clearviewport();
        setviewport(0, 0, 639, 479, BLACK);
        Display_Scale();
        break;
    }
    else if (key==27) {
        Clear_Head_Array();
        clearviewport();
        setviewport(0, 0, 639, 479, BLACK);
        Display_Scale();
        Select_Main_Menu(1);
    }
}

void Record(void) {
    Display_Head();
    n_block = atol(head->dura);
    n_block = (n_block/100)*3600 + (n_block%100)*60;
    n_block = (n_block*100)/1024 + ((n_block*100)%1024!=0);
    strcpy(filename, head->filename);
    strcat(filename, ".hea");
    if ((fptr_wh=fopen(filename, "wb"))==NULL) {
        setviewport(X1, Y1, X2, Y2, BLACK);
        setfillstyle(SOLID_FILL, WHITE);
        bar(X1_RECORD_SCHEME+30, Y1_RECORD_SCHEME, X1_RECORD_SCHEME+269,
            Y1_RECORD_SCHEME+59);
        setcolor(BLACK);
        rectangle(X1_RECORD_SCHEME+35, Y1_RECORD_SCHEME+5, X1_RECORD_SCHEME+264,
            Y1_RECORD_SCHEME+54);
        rectangle(X1_RECORD_SCHEME+37, Y1_RECORD_SCHEME+7, X1_RECORD_SCHEME+262,
            Y1_RECORD_SCHEME+52);
        setcolor(BLACK);
        outtextxy(X1_RECORD_SCHEME+55, Y1_RECORD_SCHEME+27, "Can not creat .HEA file!");
        key = getch();
        while ((key!=13)&&(key!=27)) key = getch();
        clearviewport();
        setviewport(0, 0, 639, 479, BLACK);
        Clear_Head_Display();
        Display_Scale();
        Select_Main_Menu(1);
    }

    fwrite(head->patient_name, sizeof(head->patient_name), 1, fptr_wh);
    strset(filename, '^0');

```

```

strcpy(filename, head->filename);
strcat(filename, ".ecg");
if ((fptr_w=fopen(filename, "wb"))==NULL) {
    setviewport(X1, Y1, X2, Y2, BLACK);
    setfillstyle(SOLID_FILL, WHITE);
    bar(X1_RECORD_SCHEME+30, Y1_RECORD_SCHEME, X1_RECORD_SCHEME+269,
        Y1_RECORD_SCHEME+59);
    setcolor(BLACK);
    rectangle(X1_RECORD_SCHEME+35, Y1_RECORD_SCHEME+5, X1_RECORD_SCHEME+264,
        Y1_RECORD_SCHEME+54);
    rectangle(X1_RECORD_SCHEME+37, Y1_RECORD_SCHEME+7, X1_RECORD_SCHEME+262,
        Y1_RECORD_SCHEME+52);
    setcolor(BLACK);
    outtextxy(X1_RECORD_SCHEME+55, Y1_RECORD_SCHEME+27, "Can not creat .ECG file!");
    key = getch();
    while ((key!=13)&&(key!=27)) key = getch();
    clearviewport();
    setviewport(0, 0, 639, 479, BLACK);
    Clear_Head_Display();
    Display_Scale();
    Select_Main_Menu(1);
}
i = 0;
Start_AD_Card();
while (i<n_block) {
    key = bioskey(2);
    if ((key!=4)&&(key!=20)&&(key!=36)&&(key!=52)&&(key!=68)&&(key!=84)&&(key!=100)&&(key!=16)) {
        do {
            pci812(7, param);
        } while (param[46]==1);
        Display_Signal(1);
        ECG_Compress();
        i++;
    }
    else {
        n_block = i;
        break;
    }
    key = bioskey(2);
    if ((key!=4)&&(key!=20)&&(key!=36)&&(key!=52)&&(key!=68)&&(key!=84)&&(key!=100)&&(key!=116)) {
        do {
            pci812(7, param);
        } while (param[46]==3);
    }
}

```

```

        Display_Signal(1);
        ECG_Compress();
        i++;
    }
    else {
        n_block = i;
        break;
    }
}

putw((unsigned int)n_block, fptr_wh);
fclose(fptr_w); fclose(fptr_wh);
setfillstyle(SOLID_FILL, BLACK);
bar(X1, Y1, X2, Y2);
Display_Scale();
Clear_Head_Display();
setfillstyle(SOLID_FILL, BLUE);
bar(X2+36, Y2-23, getmaxx()-32, Y2-15);
setcolor(WHITE);
outtextxy(X2+44, Y2-22, "0%");
Window(X2+10, Y2-10, getmaxx()-8, Y2, SOLID_FILL, BLACK, LIGHTCYAN);
setfillstyle(SOLID_FILL, LIGHTGREEN);
bar(X1+6, Y2+15, X1+96, Y2+23);
bar(X2-95, Y2+15, X2-5, Y2+23);
Clear_Head_Array();
}

void Open_Read(void) {
    done = findfirst("*.ecg", &ffblk, 0);
    n_file_exist = 0;
    while ((!done)&&(n_file_exist<N_FILE)) {
        strcpy(filename_list[n_file_exist++], ffbk.ff_name);
        done = findnext(&ffblk);
    }
    if (n_file_exist==0) {
        setviewport(X1, Y1, X2, Y2, BLACK);
        setfillstyle(SOLID_FILL, WHITE);
        bar(X1_READ_SCHEME, Y1_READ_SCHEME, X1_READ_SCHEME+349, Y1_READ_SCHEME+59);
        setcolor(BLACK);
        rectangle(X1_READ_SCHEME+5, Y1_READ_SCHEME+5, X1_READ_SCHEME+344,
            Y1_READ_SCHEME+54);
        rectangle(X1_READ_SCHEME+7, Y1_READ_SCHEME+7, X1_READ_SCHEME+342,
            Y1_READ_SCHEME+52);
        setcolor(BLACK);
        outtextxy(X1_READ_SCHEME+17, Y1_READ_SCHEME+27, "ECG file not found in current directory.");
    }
}

```

```

key = getch();
while ((key!=13)&&(key!=27)) key = getch();
clearviewport();
setviewport(0, 0, 639, 479, BLACK);
Display_Scale();
Select_Main_Menu(2);
}
else {
    setviewport(X1, Y1, X2, Y2, BLACK);
    Window(X1_FILE_MENU, Y1_FILE_MENU, X1_FILE_MENU+103, Y1_FILE_MENU+225,
           BLACK, WHITE, WHITE);
    Display_File_Menu(0);
    Select_File_Menu(0, 0);
}
}

void Display_File_Menu(unsigned int start) {
    unsigned int final;
    setfillstyle(SOLID_FILL, BLACK);
    bar(X1_FILE_MENU, Y1_FILE_MENU, X1_FILE_MENU+103, Y1_FILE_MENU+225);
    setcolor(WHITE);
    final = start + 20;
    if (final<=n_file_exist) {
        for (i=start, j=Y1_FILE_MENU; i<final; i++, j+=11) outtextxy(X1_FILE_MENU+5, j+5, filename_list[i]);
    }
    else {
        for (i=start, j=Y1_FILE_MENU; i<n_file_exist; i++, j+=11) outtextxy(X1_FILE_MENU+5, j+5, filename_list[i]);
    }
}

void Select_File_Menu(unsigned int start, unsigned int file_no) {
    setfillstyle(SOLID_FILL, YELLOW);
    bar(X1_FILE_MENU+3, Y1_FILE_MENU+3+11*file_no, X1_FILE_MENU+100, Y1_FILE_MENU+13+11*file_no);
    setcolor(BLACK);
    outtextxy(X1_FILE_MENU+5, Y1_FILE_MENU+5+11*file_no, filename_list[start+file_no]);
    while (1) {
        key_file_menu = bioskey(0);
        if ((key_file_menu!=UP_ARROW)&&(key_file_menu!=DOWN_ARROW)&&(key_file_menu!=PAGE_UP)&&
            (key_file_menu!=PAGE_DOWN)&&(key_file_menu!=RETURN)&&(key_file_menu!=ESCAPE)) continue;
        switch (key_file_menu) {
            case UP_ARROW :
                if (file_no==0) {
                    if (start==0);
                    else {

```

```

        Unselect_File_Menu(start, file_no);

        file_no = 19;

        Display_File_Menu(start);

    }

}

else Unselect_File_Menu(start, file_no--);

Select_File_Menu(start, file_no);

break;

case DOWN_ARROW :

    if (file_no==19) {

        if (n_file_exist%20==0) {

            if (start==n_file_exist-20);

            else {

                start += 20;

                Unselect_File_Menu(start, file_no);

                file_no = 0 ;

                Display_File_Menu(start);

            }

        }

        else {

            start += 20;

            Unselect_File_Menu(start, file_no);

            file_no = 0 ;

            Display_File_Menu(start);

        }

    }

    else {

        if ((file_no==n_file_exist%20-1)&&(start==(n_file_exist/20)*20));

        else Unselect_File_Menu(start, file_no++);

    }

    Select_File_Menu(start, file_no);

    break;

case PAGE_UP :

    if (start==0);

    else {

        Unselect_File_Menu(start, file_no);

        file_no = 0;

        start -= 20;

        Display_File_Menu(start);

    }

    Select_File_Menu(start, file_no);

    break;

case PAGE_DOWN :

```

เอกสารนี้เป็นเอกสารที่สำนักงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (n_file_exist%20==0) {
    if (start==n_file_exist-20);
    else {
        Unselect_File_Menu(start, file_no);
        file_no = 0;
        start += 20;
        Display_File_Menu(start);
    }
}
else {
    if ((start==(n_file_exist/20)*20));
    else {
        Unselect_File_Menu(start, file_no);
        file_no = 0;
        start += 20;
        Display_File_Menu(start);
    }
}
Select_File_Menu(start, file_no);
break;
case RETURN :
    clearviewport();
    setviewport(0, 0, 639, 479, BLACK);
    Display_Scale();
    Read(start, file_no);
    Select_Main_Menu(2);
case ESCAPE :
    clearviewport();
    setviewport(0, 0, 639, 479, BLACK);
    Display_Scale();
    Select_Main_Menu(2);
}
break;
}
}

void Unselect_File_Menu(unsigned int start, unsigned int file_no) {
    setfillstyle(SOLID_FILL, BLACK);
    bar(X1_FILE_MENU+3, Y1_FILE_MENU+3+11*file_no, X1_FILE_MENU+100, Y1_FILE_MENU+13+11*file_no);
    setcolor(WHITE);
    outtextxy(X1_FILE_MENU+5, Y1_FILE_MENU+5+11*file_no, filename_list[start+file_no]);
}

void Read(unsigned int start, unsigned int file_no) {
    Clear_Head_Array();

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ไม่ควรนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตจากทางมหาวิทยาลัย
 ไม่ควรนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตจากทางมหาวิทยาลัย

```

ptr = strchr(filename_list[start+file_no], '.');
strncat(head->filename, filename_list[start+file_no], ptr-filename_list[start+file_no]);
strcpy(filename, head->filename);
strcat(filename, ".hea");
if ((fptr_rh=fopen(filename, "rb"))==NULL) {
    setviewport(X1, Y1, X2, Y2, BLACK);
    setfillstyle(SOLID_FILL, WHITE);
    bar(X1_READ_SCHEME+60, Y1_READ_SCHEME, X1_READ_SCHEME+289, Y1_READ_SCHEME+59);
    setcolor(BLACK);
    rectangle(X1_READ_SCHEME+65, Y1_READ_SCHEME+5, X1_READ_SCHEME+284,
        Y1_READ_SCHEME+54);
    rectangle(X1_READ_SCHEME+67, Y1_READ_SCHEME+7, X1_READ_SCHEME+282,
        Y1_READ_SCHEME+52);
    setcolor(BLACK);
    outtextxy(X1_READ_SCHEME+83, Y1_READ_SCHEME+27, "Can not open .HEA file!");
    key = getch();
    while ((key!=13)&&(key!=27)) key = getch();
    clearviewport();
    setviewport(0, 0, 639, 479, BLACK);
    Display_Scale();
    Select_Main_Menu(2);
}
fread(head->patient_name, sizeof(head->patient_name), 1, fptr_rh);
n_block = getw(fptr_rh);
fclose(fptr_rh);
fptr_pos_array = (unsigned int *)calloc(n_block, sizeof(unsigned int));
tmp = n_block * 10.24;
tmp = 100 + tmp/3600;
ptr = fcvt(tmp, 6, &dec, &sign);
strncat(head->dura, ptr+1, 2);
tmp = 100 + (tmp-(unsigned int)tmp)*60;
ptr = fcvt(tmp, 6, &dec, &sign);
strncat(head->dura, ptr+1, 2);
Display_Head();
if ((fptr_r=fopen(filename_list[start+file_no], "rb"))==NULL) {
    setviewport(X1, Y1, X2, Y2, BLACK);
    setfillstyle(SOLID_FILL, WHITE);
    bar(X1_READ_SCHEME+60, Y1_READ_SCHEME, X1_READ_SCHEME+289, Y1_READ_SCHEME+59);
    setcolor(BLACK);
    rectangle(X1_READ_SCHEME+65, Y1_READ_SCHEME+5, X1_READ_SCHEME+284,
        Y1_READ_SCHEME+54);
    rectangle(X1_READ_SCHEME+67, Y1_READ_SCHEME+7, X1_READ_SCHEME+282,
        Y1_READ_SCHEME+52);

```

```

setcolor(BLACK);
outtextxy(X1_READ_SCHEME+83, Y1_READ_SCHEME+27, "Can not open .ECG file!");
key = getch();
while ((key!=13)&&(key!=27)) key = getch();
clearviewport();
setviewport(0, 0, 639, 479, BLACK);
Clear_Head_Display();
Display_Scale();
Select_Main_Menu(2);
}
key = 13;
for (i=0; i<n_block;) {
    if (key==13) {
        fptr_pos_array[i] = (unsigned int)ftell(fptr_r);
        j = 0;
    }
    else if (key=='b') {
        fseek(fptr_r, fptr_pos_array[i], SEEK_SET);
        j = 3;
    }
    else if (key==10) {
        fptr_pos_array[i] = (unsigned int)ftell(fptr_r);
        j = copy;
    }
    else if (key==127) {
        fseek(fptr_r, fptr_pos_array[i], SEEK_SET);
        j = copy;
    }
    ECG_Reconstruct();
    while (j<4) {
        xx = X1;
        setcolor(YELLOW);
        for (k=512*j; k<512*(j+1); k++) {
            yy = (Data[k]-2048.0) / 8.192;
            if (yy>=0)
                if (yy-floor(yy)<0.5) yy = floor(yy);
                else yy = ceil(yy);
            else
                if (-yy-floor(-yy)<0.5) yy = -floor(-yy);
                else yy = -ceil(-yy);
            if (yy>229) yy = 229;
            if (yy<-200) yy = -200;
            yy = 479.0 - (yy+228.0);

```

```

        if (xx>X1) line(x, y, xx, yy);
        x = xx; y = yy;
        xx++;
    }
    Display_Time();
    Display_Percent_Complete();
    key = getch();
    if (key==13) {
        if ((i==n_block-1)&&(j==3));
        else j++;
    }
    else if (key=='b') {
        if (i==0) {
            if (j==0);
            else j--;
        }
        else {
            if (j==0) {
                setfillstyle(SOLID_FILL, BLACK);
                bar(X1, Y1, X2, Y2);
                Display_Scale();
                break;
            }
            else j--;
        }
    }
    else if (key==10) {
        if (i==n_block-1) {
            if (j!=3) j = 3;
            else ;
        }
        else {
            copy = j;
            setfillstyle(SOLID_FILL, BLACK);
            bar(X1, Y1, X2, Y2);
            Display_Scale();
            break;
        }
    }
    else if (key==127) {
        if (i==0) {
            if (j!=0) j = 0;
            else ;
        }
    }

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    else {
        copy = j;
        setfillstyle(SOLID_FILL, BLACK);
        bar(X1, Y1, X2, Y2);
        Display_Scale();
        break;
    }
}

else if (key==27) break;
setfillstyle(SOLID_FILL, BLACK);
bar(X1, Y1, X2, Y2);
Display_Scale();
}

if ((key==13)|| (key==10)) i++;
else if ((key=='b')||(key==127)) i--;
else if (key==27) break;
}

fclose(fp_r);
free(fp_pos_array);
setfillstyle(SOLID_FILL, BLACK);
bar(X1, Y1, X2, Y2);
Display_Scale();
Clear_Head_Display();
setfillstyle(SOLID_FILL, BLUE);
bar(X2+36, Y2-23, getmaxx()-32, Y2-15);
setcolor(WHITE);
outtextxy(X2+44, Y2-22, "0%");
Window(X2+10, Y2-10, getmaxx()-8, Y2, SOLID_FILL, BLACK, LIGHTCYAN);
setfillstyle(SOLID_FILL, LIGHTGREEN);
bar(X1+6, Y2+15, X1+96, Y2+23);
bar(X2-95, Y2+15, X2-5, Y2+23);
Clear_Head_Array();
}

void Clear_Head_Array(void) {
    strset(head->patient_name, '\0');
    strset(head->dura, '\0');
    strset(head->filename, '\0');
}

void Display_Head(void) {
    unsigned char pos;
    char *ptr, dura[6];
    setcolor(WHITE);

```

```

pos = ((83-8*strlen(head->filename))/2) - 1;
outtextxy(X2+12+pos, Y1+15, strdup(head->filename));
ptr = strchr(head->patient_name, '\0');
pos = ((83-8*(ptr-head->patient_name))/2) - 1;
outtextxy(X2+12+pos, Y1+65, head->patient_name);
strset(dura, '\0');
strncat(dura, head->dura, 2);
strcat(dura, ".");
strncat(dura, head->dura+2, 2);
outtextxy(X2+12+20, Y1+115, dura);
}

void Clear_Head_Display(void) {
    setfillstyle(SOLID_FILL, BLACK);
    bar(X2+10, Y1+10, getmaxx()-8, Y1+26);
    bar(X2+10, Y1+60, getmaxx()-8, Y1+76);
    bar(X2+10, Y1+110, getmaxx()-8, Y1+126);
}

void About(void) {
    setviewport(X1, Y1, X2, Y2, BLACK);
    setfillstyle(SOLID_FILL, WHITE);
    bar(X1_ABOUT, Y1_ABOUT, X1_ABOUT+459, Y1_ABOUT+235);
    setcolor(BLACK);
    rectangle(X1_ABOUT+5, Y1_ABOUT+5, X1_ABOUT+454, Y1_ABOUT+231);
    rectangle(X1_ABOUT+7, Y1_ABOUT+7, X1_ABOUT+452, Y1_ABOUT+229);
    setcolor(WHITE);
    line(X1_ABOUT+203, Y1_ABOUT+5, X1_ABOUT+256, Y1_ABOUT+5);
    line(X1_ABOUT+203, Y1_ABOUT+7, X1_ABOUT+256, Y1_ABOUT+7);
    setcolor(BLACK);
    outtextxy(X1_ABOUT+211, Y1_ABOUT+2, "About");
    setcolor(LIGHTRED);
    outtextxy(X1_ABOUT+77, Y1_ABOUT+23, "ECG Personal Computer Recording System");
    setcolor(BLACK);
    outtextxy(X1_ABOUT+222, Y1_ABOUT+43, "by");
    setcolor(BLUE);
    outtextxy(X1_ABOUT+119, Y1_ABOUT+63, "Mr. Manit Kiatkhamjaikajorn");
    setcolor(GREEN);
    outtextxy(X1_ABOUT+80, Y1_ABOUT+83, "Department of Electronics Engineering");
    outtextxy(X1_ABOUT+34, Y1_ABOUT+103, "King Mongkut's Institute of Technology Ladkrabang");
    setcolor(BLACK);
    outtextxy(X1_ABOUT+14, Y1_ABOUT+143, "This program supports for the PCL-812PG 12 bits/sample");
    outtextxy(X1_ABOUT+14, Y1_ABOUT+163, "A/D card. The sample rates setting is 200 Hz and using");
    outtextxy(X1_ABOUT+14, Y1_ABOUT+183, "the Discrete Wavelet Transform coding technic for data");
    outtextxy(X1_ABOUT+14, Y1_ABOUT+203, "compression.");
}

```

```

key = getch();
while ((key!=13)&&(key!=27)) key = getch();
clearviewport();
setviewport(0, 0, 639, 479, BLACK);
Display_Scale();
}

```

```

void Display_Signal(unsigned char sw) {
    for (j=0; j<4; j++) {
        xx = X1;
        setcolor(YELLOW);
        for (k=512*j; k<512*(j+1); k++) {
            yy = (Data[k]-2048.0) / 8.192;
            if (yy>=0)
                if (yy-floor(yy)<0.5) yy = floor(yy);
                else yy = ceil(yy);
            else
                if (-yy-floor(-yy)<0.5) yy = -floor(-yy);
                else yy = -ceil(-yy);
            if (yy>229) yy = 229;
            if (yy<-200) yy = -200;
            yy = 479.0 - (yy+228.0);
            if (xx>X1) line(x, y, xx, yy);
            x = xx; y = yy;
            xx++;
        }
        if (sw) {
            Display_Time();
            Display_Percent_Complete();
            delay(1000);
        }
        else delay(2500);
        setfillstyle(SOLID_FILL, BLACK);
        bar(X1, Y1, X2, Y2);
        Display_Scale();
    }
}

```

```

}

void Display_Time(void) {
    char time[12];
    setfillstyle(SOLID_FILL, LIGHTGREEN);
    bar(X1+6, Y2+15, X1+96, Y2+23);
    bar(X2-95, Y2+15, X2-5, Y2+23);
    setcolor(RED);
    strset(time, "0");
}

```

```

tmp = (10.24)*i + (2.56)*j;
tmp = 100 + tmp/3600;
ptr = fcvt(tmp, 6, &dec, &sign);
strncat(time, ptr+1, 2);
strcat(time, ".");
tmp = 100 + (tmp-(unsigned int)tmp)*60;
ptr = fcvt(tmp, 6, &dec, &sign);
strncat(time, ptr+1, 2);
strcat(time, ".");
tmp = 100 + (tmp-(unsigned int)tmp)*60;
ptr = fcvt(tmp, 6, &dec, &sign);
strncat(time, ptr+1, 2);
strcat(time, ".");
tmp = 100 + (tmp-(unsigned int)tmp)*100;
ptr = fcvt(tmp, 6, &dec, &sign);
strncat(time, ptr+1, 2);
outtextxy(X1+8, Y2+16, time);
strset(time, '\0');
tmp = (10.24)*i + (2.56)*(j+1);
tmp = 100 + tmp/3600;
ptr = fcvt(tmp, 6, &dec, &sign);
strncat(time, ptr+1, 2);
strcat(time, ".");
tmp = 100 + (tmp-(unsigned int)tmp)*60;
ptr = fcvt(tmp, 6, &dec, &sign);
strncat(time, ptr+1, 2);
strcat(time, ".");
tmp = 100 + (tmp-(unsigned int)tmp)*60;
ptr = fcvt(tmp, 6, &dec, &sign);
strncat(time, ptr+1, 2);
strcat(time, ".");
tmp = 100 + (tmp-(unsigned int)tmp)*100;
ptr = fcvt(tmp, 6, &dec, &sign);
strncat(time, ptr+1, 2);
outtextxy(X2-93, Y2+16, time);
}

```

```

void Display_Percent_Complete(void) {

```

```

    char string[5];

```

```

    unsigned char pos;

```

```

    float percent;

```

```

    setfillstyle(SOLID_FILL, BLUE);

```

```

    bar(X2+36, Y2-23, getmaxx()-32, Y2-15);

```

```

    percent = (float)(4*i+j+1) / (float)(4*n_block);

```

```

setfillstyle(SOLID_FILL, BLACK);
bar(X2+11, Y2-9, X2+91, Y2-1);
setfillstyle(SOLID_FILL, YELLOW);
bar(X2+11, Y2-9, X2+11+(unsigned int)(80*percent), Y2-1);
strset(string, '\0');
percent = percent * 100;
if (percent<1) strcat(string, "0");
else if (percent<10) {
    ptr = fcvt(percent, 1, &dec, &sign);
    strncat(string, ptr, 1);
}
else if (percent<100) {
    ptr = fcvt(percent, 2, &dec, &sign);
    strncat(string, ptr, 2);
}
else {
    ptr = fcvt(percent, 3, &dec, &sign);
    strncat(string, ptr, 3);
}
strcat(string, "%");
pos = (83-8*strlen(string)) / 2;
setcolor(WHITE);
outtextxy(X2+12+pos, Y2-22, string);
}

```

ECGCOMP.C

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define    N_Data                2048
#define    N_QMF                 18
#define    Level                 4
#define    Quantization_Step_Size 60
#define    N_Subband_RLC         3
#define    N_Bitpack_RLC_ZeroIndex 11
#define    N_Bitpack_Iqencode    8
#define    N_Bitpack_HuffBitCodeTable 4

void Zero_Mean_Set(void);
void DWT(void);
void Quantize(void);

void Nbit_Pack(unsigned int *input, unsigned int ni, unsigned int nbit, FILE *fptr);

```

```

void Subband_Zero_Run_Length_Code(void);
void Histogram(void);
void Histogram_Sort(void);
void Huffman_Code(void);
void Huffman_Code_Sort(void);
void Huffman_Code_Pack(void);
void Data_Stream_Bytes_Pack(void);

extern float temp;
extern float Data[N_Data], QMF[N_QMF], *Ptr_Data, *Ptr_QMF;
extern unsigned int absmax, n_IQset, n_Data_RLCout, n_code, adj, *histo_Data, *histo_Data_encode, *index;
extern int *IQset, *IQencode;
extern unsigned int *IQencode_Stream;
extern unsigned int *huffcval, *full_huffcval, *Data_huffcval;
extern unsigned int *huffcbit, *full_huffcbit, *Data_huffcbit;
extern FILE *fptr_w;
void ECG_Compress(void) {
    register unsigned int i, j;
    Zero_Mean_Set();
    DWT();
    Quantize();
    n_IQset = 2*absmax + 1;
    IQset = (int *)calloc(n_IQset, sizeof(int));
    *IQset = -absmax;
    for (i=1; i<n_IQset; i++) *(IQset+i) = *(IQset+i-1) + 1;
    Subband_Zero_Run_Length_Code();
    histo_Data = (unsigned int *)calloc(n_IQset, sizeof(unsigned int));
    Histogram();
    n_code = 0;
    for (i=0; i<n_IQset; i++) {
        if (*(histo_Data+i)) n_code++;
    }
    histo_Data_encode = (unsigned int *)calloc(n_code, sizeof(unsigned int));
    index = (unsigned int *)calloc(n_code, sizeof(unsigned int));
    j = 0;
    for (i=0; i<n_IQset; i++) {
        if (*(histo_Data+i)) {
            *(histo_Data_encode+j) = *(histo_Data+i);
            *(index+j) = i;
            j++;
        }
    }
    free(histo_Data);
    IQencode = (int *)calloc(n_code, sizeof(int));

```

```

for (i=0; i<n_code; i++) *(IQencode+i) = IQset[*(index+i)];
free(IQset);
Histogram_Sort();
huffcval = (unsigned int *)calloc(n_code, sizeof(unsigned int));
huffcbit = (unsigned int *)calloc(n_code, sizeof(unsigned int));
Huffman_Code();
free(histo_Data_encode);
Huffman_Code_Sort();
free(index);
putw((unsigned int)n_code, fptr_w);
adj = (unsigned int)pow(2,(N_Bitpack_IQencode-1)) - 1;
IQencode_Stream = (unsigned int *)calloc(n_code, sizeof(unsigned int));
for (i=0; i<n_code; i++) *(IQencode_Stream+i) = *(IQencode+i) + adj;
Nbit_Pack(IQencode_Stream, n_code, N_Bitpack_IQencode, fptr_w);
free(IQencode_Stream);
Nbit_Pack(huffcbit, n_code, N_Bitpack_HuffBitCodeTable, fptr_w);
Huffman_Code_Pack();
for (i=0; i<n_Data_RLCout; i++) Data[i] += absmax;
for (i=0; i<n_code; i++) *(IQencode+i) += absmax;
full_huffcval = (unsigned int *)calloc(n_IQset, sizeof(unsigned int));
full_huffcbit = (unsigned int *)calloc(n_IQset, sizeof(unsigned int));
for (i=0; i<n_code; i++) {
    *(full_huffcval + *(IQencode+i)) = *(huffcval+i);
    *(full_huffcbit + *(IQencode+i)) = *(huffcbit+i);
}
free(IQencode); free(huffcbit); free(huffcval);
Data_huffcval = (unsigned int *)calloc(n_Data_RLCout, sizeof(unsigned int));
Data_huffcbit = (unsigned int *)calloc(n_Data_RLCout, sizeof(unsigned int));
for (i=0; i<n_Data_RLCout; i++) {
    *(Data_huffcval+i) = *(full_huffcval + (int)Data[i]);
    *(Data_huffcbit+i) = *(full_huffcbit + (int)Data[i]);
}
free(full_huffcbit); free(full_huffcval);
Data_Stream_Bytes_Pack();
free(Data_huffcval); free(Data_huffcbit);
}

void Zero_Mean_Set(void) {
    register unsigned int i;
    float sum = 0.0;
    Ptr_Data = Data;
    for (i=0; i<N_Data; i++) sum += *Ptr_Data++;
    temp = sum / (float)N_Data;
    fwrite(&temp, sizeof(temp), 1, fptr_w);

```

```

Ptr_Data = Data;
for (i=0; i<N_Data; i++) (*Ptr_Data++) -= temp;
}

void DWT(void) {
    register unsigned int i, j, k;
    float *pad, *ptr_pad, *tmp, *ptr_tmp, sum;
    unsigned int n;
    n = N_Data;
    for (i=0; i<Level; i++) {
        pad = (float *)calloc(n+N_QMF-2, sizeof(float));
        tmp = (float *)calloc(n, sizeof(float));
        Ptr_Data = Data; ptr_pad = pad + N_QMF - 2;
        for (j=0; j<n; j++) *ptr_pad++ = *Ptr_Data++;
        Ptr_Data = Data + n - N_QMF + 2; ptr_pad = pad;
        for (j=0; j<N_QMF-2; j++) *ptr_pad++ = *Ptr_Data++;
        ptr_tmp = tmp + n/2;
        for (j=0; j<n/2; j++) {
            sum = 0.0; Ptr_QMF = QMF + N_QMF - 1; ptr_pad = pad + 2*j;
            for (k=0; k<N_QMF; k++) sum += pow(-1, k+1) * (*Ptr_QMF--) * (*ptr_pad++);
            *ptr_tmp++ = sum;
        }
        Ptr_Data = Data; ptr_pad = pad;
        for (j=0; j<n+N_QMF-2; j++) *ptr_pad++ = *Ptr_Data++;
        Ptr_Data = Data;
        ptr_pad = pad + n;
        for (j=0; j<N_QMF-2; j++) *ptr_pad++ = *(Data+j);
        ptr_tmp = tmp;
        for (j=0; j<n/2; j++) {
            sum = 0.0; Ptr_QMF = QMF; ptr_pad = pad + 2*j;
            for (k=0; k<N_QMF; k++) sum += (*Ptr_QMF++) * (*ptr_pad++);
            *ptr_tmp++ = sum;
        }
        ptr_tmp = tmp; Ptr_Data = Data;
        for (j=0; j<n; j++) *Ptr_Data++ = *ptr_tmp++;
        free(pad); free(tmp);
        n /= 2;
    }
}

```

```

void Quantize(void) {

```

```

    register unsigned int i;

```

```

    absmax = 0; Ptr_Data = Data;

```

```

    for (i=0; i<N_Data; i++) {

```

```

        *Ptr_Data /= Quantization_Step_Size;

```

```

    if (abs(*Ptr_Data)>absmax) absmax = abs(*Ptr_Data);
    if (*Ptr_Data>=0)
        if (((*Ptr_Data)-floor(*Ptr_Data)<0.5) *Ptr_Data++ = floor(*Ptr_Data);
        else *Ptr_Data++ = ceil(*Ptr_Data);
    else
        if ((-*Ptr_Data)-floor(-*Ptr_Data)<0.5) *Ptr_Data++ = -floor(-*Ptr_Data);
        else *Ptr_Data++ = -ceil(-*Ptr_Data);
}
}

void Nbit_Pack(unsigned int *input, unsigned int ni, unsigned int nbit, FILE *fptr) {
    register unsigned int i;
    unsigned char len ;
    unsigned int tot_bits;
    unsigned long int aux1, aux2, Lmask;
    len = tot_bits = aux1 = aux2 = 0;
    for (i=0; i<ni; i++) {
        len += nbit;
        tot_bits += nbit;
        aux1 = (aux1 << (unsigned int)nbit) | (unsigned long int)*(input+i);
        if (len<8) continue;
        else
            while (len>=8){
                aux2 = aux1;
                Lmask = 255;
                Lmask <=<= (len-8);
                aux2 = aux2 & Lmask;
                Lmask = ~Lmask;
                aux1 = aux1 & Lmask;
                aux2 >>= (len-8);
                putc((unsigned char)aux2, fptr);
                len -= 8;
            }
    }
    if (tot_bits%8!=0) {
        aux1 <=<= (8-len);
        putc((unsigned char)aux1, fptr);
    }
}

void Subband_Zero_Run_Length_Code(void) {
    register unsigned int i, n_o, inx, n_inx;
    unsigned int begin, n, *arinx;

    int *tmp, p;
    begin = N_Data*pow(2,N_Subband_RLC);

```

```

n = N_Data - begin;
tmp = (int *)calloc(n, sizeof(int));
for (i=0; i<n; i++) *(tmp+i) = (int)*(Data+begin+i);
arinx = (unsigned int *)calloc(n, sizeof(unsigned int));
i = n_o = n_inx = 0;
while (i<n) {
    p = *(tmp+i);
    inx = 0;
    if (i==(n-1)) {
        if (p==0) {
            inx = 0;
            *(arinx+n_inx) = inx;
            *(Data+begin+n_o) = (float)p;
            n_inx++;
        }
        else *(Data+begin+n_o) = (float)p;
        n_o++;
        break;
    }
    i++;
    while ((p==0) && (p==*(tmp+i))) {
        i++;
        inx++;
        if (i==n) break;
    }
    if (p==0) {
        if (inx==0) {
            inx = 0;
            *(arinx+n_inx) = inx;
            *(Data+begin+n_o) = (float)p;
        }
        else {
            *(arinx+n_inx) = inx;
            *(Data+begin+n_o) = (float)p;
        }
        n_inx++;
    }
    else *(Data+begin+n_o) = (float)p;
    n_o++;
}
free(tmp);
putw(n_inx, fptr_w);

```

เอกสารนี้เป็น Nbit_Pack(arinx, n_inx, N_Bitpack_RLC_ZeroIndex, fptr_w); หากท่านนั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    putw(n_o, fptr_w);

    free(arinx);

    n_Data_RLCout = begin + n_o;
}

void Histogram(void) {
    register unsigned int i;
    unsigned int *out_tmp,*in_tmp;
    int oi, ti;
    in_tmp = (unsigned int *)calloc(n_Data_RLCout, sizeof(unsigned int));
    out_tmp = (unsigned int *)calloc(n_IQset+1, sizeof(unsigned int));
    oi = IQset[0];
    for (i=0; i<n_Data_RLCout; i++) {
        ti = (int)Data[i] - oi + 1;
        in_tmp[i] = ti * (ti>0 && ti<n_IQset+1);
    }
    for (i=0; i<n_Data_RLCout; i++) out_tmp[in_tmp[i]]++;
    for (i=1; i<n_IQset+1; i++) histo_Data[i-1] = out_tmp[i];
    free(in_tmp); free(out_tmp);
}

void Histogram_Sort(void) {
    register unsigned int i, j;
    unsigned int big, loc, tmp1, tmp2;
    for (i=0; i<n_code-1; i++) {
        big = *(histo_Data_encode+i);
        loc = i;
        for (j=i+1; j<n_code; j++) {
            if (*(histo_Data_encode+j)>big) {
                big = *(histo_Data_encode+j);
                loc = j;
            }
        }
        if (loc==i) continue;
        else {
            tmp1 = *(histo_Data_encode+i);
            tmp2 = *(index+i);
            *(histo_Data_encode+i) = *(histo_Data_encode+loc);
            *(index+i) = *(index+loc);
            *(histo_Data_encode+loc) = tmp1;
            *(index+loc) = tmp2;
        }
    }
}

void Huffman_Code(void) {

```

```

register unsigned int i;
register int j;
int M, loc;
unsigned int *v, Ctmp, Ltmp;
float tmp, sum;
v = (unsigned int *)calloc(n_code-2, sizeof(unsigned int));
M = n_code;
for (i=0; i<(n_code-2); i++) {
    *(histo_Data_encode+M-2) += *(histo_Data_encode+M-1);
    loc = M-2;
    for (j=0; j<(M-1); j++) {
        if (*(histo_Data_encode+M-2)>=*(histo_Data_encode+j)) {
            loc = j;
            break;
        }
    }
    tmp = *(histo_Data_encode+M-2);
    for (j=M-2; j>=loc; j--) *(histo_Data_encode+j) = *(histo_Data_encode+j-1);
    *(histo_Data_encode+loc) = tmp;
    M--;
    *(v+(n_code-3)-i) = loc;
}
for (j=0; j<n_code; j++) {
    *(huffcval+j) = 0;
    *(huffcbit+j) = 1;
}
*huffcval = 0;
*(huffcval+1) = 1;
M = 1;
for (i=0; i<(n_code-2); i++) {
    if (*(v+i)==M) {
        *(huffcval+M+1) = (*(huffcval+M) << 1)+1;
        *(huffcval+M) = *(huffcval+M) << 1;
        (*(huffcbit+M))++;
        *(huffcbit+M+1) = *(huffcbit+M);
    }
    else {
        Ctmp = *(huffcval + (*(v+i)));
        Ltmp = *(huffcbit + (*(v+i)));
        for (j = *(v+i); j<M; j++) {
            *(huffcval+j) = *(huffcval+j+1);
            *(huffcbit+j) = *(huffcbit+j+1);

```

```

        *(huffcval+M) = Ctmp;
        *(huffcbit+M) = Ltmp;
        *(huffcval+M+1) = (*(huffcval+M) << 1)+1;
        *(huffcval+M) = *(huffcval+M) << 1;
        (*(huffcbit+M))++;
        *(huffcbit+M+1) = *(huffcbit+M);
    }
    M++;
}
free(v);
}

void Huffman_Code_Sort(void) {
    register unsigned int i, gap;
    register int j;
    unsigned int tmp;
    for (gap=n_code/2; gap>0; gap/=2) {
        for (i=gap; i<n_code; i++) {
            for (j=i-gap; (j>=0)&&(*(index+j)>*(index+j+gap)); j-=gap) {
                tmp = *(index+j);
                *(index+j) = *(index+j+gap);
                *(index+j+gap) = tmp;
                tmp = *(huffcval+j);
                *(huffcval+j) = *(huffcval+j+gap);
                *(huffcval+j+gap) = tmp;
                tmp = *(huffcbit+j);
                *(huffcbit+j) = *(huffcbit+j+gap);
                *(huffcbit+j+gap) = tmp;
            }
        }
    }
}

```

```

void Huffman_Code_Pack(void) {
    register unsigned int i;
    unsigned char len, ch;
    unsigned int tot_bits;
    unsigned long int aux1, aux2, Lmask;
    len = aux1 = tot_bits = 0;
    for (i=0; i<n_code; i++) {
        len += *(huffcbit+i);
        tot_bits += *(huffcbit+i);
        aux1 = (aux1 << *(huffcbit+i)) | (unsigned long int)*(huffcval+i);
        if (len<8) continue;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while (len>=8) {
    aux2 = aux1;
    Lmask = 255;
    Lmask <=<= (len-8);
    aux2 = aux2 & Lmask;
    aux2 >>= (len-8);
    ch = (unsigned char)aux2;
    putc(ch, fptr_w);
    Lmask = ~Lmask;
    aux1 = aux1 & Lmask;
    len -= 8;
}
}

if (tot_bits%8!=0) {
    ch = (unsigned char)(aux1 << (8-len));
    putc(ch, fptr_w);
}
}

void Data_Stream_Bytes_Pack(void){
    register unsigned int i;
    unsigned char len, ch;
    unsigned int tot_bits;
    unsigned long int aux1, aux2, Lmask;
    fpos_t pos;
    fgetpos(fptr_w, &pos);
    putw((unsigned int)0, fptr_w);
    len = tot_bits = aux1 = aux2 = 0;
    for (i=0; i<n_Data_RLCout; i++) {
        len += *(Data_huffcbit+i);
        tot_bits += *(Data_huffcbit+i);
        aux1 = (aux1<<*(Data_huffcbit+i)) | (unsigned long int)*(Data_huffcval+i);
        if (len<8) continue;
        else
            while (len>=8) {
                aux2 = aux1;
                Lmask = 255;
                Lmask <=<= (len-8);
                aux2 = aux2 & Lmask;
                Lmask = ~Lmask;
                aux1 = aux1 & Lmask;
                aux2 >>= (len-8);
                ch = (unsigned char)aux2;
                putc(ch, fptr_w);
            }
    }
}

```

```

        len -= 8;
    }
}

if (tot_bits%8!=0) {
    ch = (unsigned char)(aux1 << (8-len));
    putc(ch, fptr_w);
}

fsetpos(fptr_w, &pos);
putw(tot_bits, fptr_w);
fseek(fptr_w, 0L, SEEK_END);
}

```

ECGRECON.C

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define    N_Data    2048
#define    N_QMF    18
#define    Level    4
#define    Quantization_Step_Size    60
#define    N_Subband_RLC    3
#define    N_Bitpack_RLC_ZeroIndex    11
#define    N_Bitpack_Iqencode    8
#define    N_Bitpack_HuffBitCodeTable    4
void Nbit_Unpack(unsigned int *output, unsigned int no, unsigned int nbit, FILE *fptr);
void Huffman_Code_Unpack(void);
void Data_Stream_Byte_Unpack(void);
void Subband_Zero_Run_Length_Decode(void);
void Dequantize(void);
void IDWT(void);
extern float temp;
void Zero_Mean_Reset(void);
extern int *IQencode;
extern float Data[N_Data], QMF[N_QMF], *Ptr_Data, *Ptr_QMF;
extern unsigned int len, n_code, adj, *IQencode_Stream, *huffcval, *huffcbit;
extern FILE *fptr_r;
extern fpos_t start, pos, end;
void ECG_Reconstruct(void) {
    register unsigned int i;
    fgetpos(fptr_r, &start);
    fseek(fptr_r, sizeof(float), SEEK_CUR);

```

```

fgetpos(fp_r, &pos);
len = (unsigned int)getw(fp_r);
len *= N_Bitpack_RLC_ZeroIndex;
len = len/8 + (len%8 != 0) + sizeof(unsigned int);
fseek(fp_r, len, SEEK_CUR);
n_code = (unsigned int)getw(fp_r);
IQencode_Stream = (unsigned int *)calloc(n_code, sizeof(unsigned int));
Nbit_Unpack(IQencode_Stream, n_code, N_Bitpack_IQencode, fp_r);
adj = (unsigned int)pow(2, (N_Bitpack_IQencode-1)) - 1;
IQencode = (int *)calloc(n_code, sizeof(int));
for (i=0; i<n_code; i++) *(IQencode+i) = *(IQencode_Stream+i) - adj;
free(IQencode_Stream);
huffcbit = (unsigned int *)calloc(n_code, sizeof(unsigned int));
Nbit_Unpack(huffcbit, n_code, N_Bitpack_HuffBitCodeTable, fp_r);
huffcval = (unsigned int *)calloc(n_code, sizeof(unsigned int));
Huffman_Code_Unpack();
Data_Stream_Byte_Unpack();
fgetpos(fp_r, &end);
free(huffcval); free(huffcbit); free(IQencode);
fsetpos(fp_r, &pos);
Subband_Zero_Run_Length_Decode();
fsetpos(fp_r, &start);
fseek(fp_r, sizeof(temp), SEEK_CUR);
Dequantize();
IDWT();
fsetpos(fp_r, &start);
Zero_Mean_Reset();
fsetpos(fp_r, &end);
}

void Nbit_Unpack(unsigned int *output, unsigned int no, unsigned int nbit, FILE *fptr) {
    register unsigned int i, j;
    unsigned char len, ch, mask;
    unsigned long int aux1, aux2, inx;
    mask = 128;
    i = len = aux1 = aux2 = 0;
    inx = 1;
    while (inx) {
        ch = getc(fptr);
        for (j=0; j<8; j++) {
            aux1 = ch & mask;
            ch <<= 1;
            aux2 <<= 1;
            if (aux1!=0) aux2 |= 1;

```

```

len++;
if (len==(unsigned char)nbit) {
    *(output+i) = (unsigned int)aux2;
    len = aux2 = 0;
    i++;
    if (i==(unsigned int)no) {
        inx = 0;
        break;
    }
}
}
}
}

```

```

void Huffman_Code_Unpack(void) {
    register unsigned int i, j;
    unsigned int len;
    unsigned char ch, mask;
    unsigned long int aux1, aux2, inx;
    mask = 128;
    i = len = aux1 = aux2 = 0;
    inx = 1;
    while (inx) {
        ch =getc(fp_r);
        for (j=0; j<8; j++) {
            aux1 = ch & mask;
            ch <<= 1;
            aux2 <<= 1;
            if (aux1!=0) aux2 |= 1;
            len++;
            if (len==(huffcbit+i)) {
                *(huffcval+i) = aux2;
                len = aux2 = 0;
                i++;
                if (i==n_code) {
                    inx = 0;
                    break;
                }
            }
        }
    }
}

```

```

void Data_Stream_Byte_Unpack(void) {
    register unsigned int i, j, k, l;

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 เอกสารนี้ได้รับอนุญาตให้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unsigned char mask, ch;
unsigned long int aux1, aux2, flen;
unsigned int len, Tot_bits;
Tot_bits = (unsigned int)getw(fp_r);
i = len = flen = aux1 = aux2 = 0;
mask = 128;
for (j=0; j<(Tot_bits/8 + (Tot_bits%8 != 0)); j++) {
    ch = (unsigned char)getc(fp_r);
    for (k=0; k<8; k++) {
        if (flen>Tot_bits) break;
        aux1 = ch & mask;
        ch <<= 1;
        aux2 <<= 1;
        if (aux1!=0) aux2 |= 1;
        len++;
        for (l=0; l<n_code; l++) {
            if (*(huffcbit+l)==len) {
                if (*(huffcval+l)==aux2) {
                    aux2 = 0;
                    *(Data+i) = (float)*(IQencode+l);
                    i++;
                    len = 0;
                    break;
                }
            }
        }
        flen++;
    }
}

void Subband_Zero_Run_Length_Decode(void) {
    register unsigned int i, j, k, l;
    unsigned int n_inx, n, begin, *arinx;
    int *tmp;
    n_inx = (unsigned int)getw(fp_r);
    arinx = (unsigned int *)calloc(n_inx, sizeof(unsigned int));
    Nbit_Unpack(arinx, n_inx, N_Bitpack_RLC_ZeroIndex, fp_r);
    n = (unsigned int)getw(fp_r);
    begin = N_Data/pow(2, N_Subband_RLC);
    tmp = (int *)calloc(n, sizeof(int));
    for (i=0; i<n; i++) *(tmp+i) = (int)*(Data+begin+i);
    i = j = k = 0;

```

เอกสารนี้แจ้งให้ทราบถึงข้อจำกัดในการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

*(Data+begin+(j++)) = (float)*(tmp+i);
if (*(tmp+(i++))!=0) {
    for (l=0; l<*(arinx+k); l++) *(Data+begin+(j++)) = 0;
    k++;
}
}
free(tmp); free(arinx);
}

void Dequantize(void) {
    register unsigned int i;
    Ptr_Data = Data;
    for (i=0; i<N_Data; i++) (*Ptr_Data++) *= Quantization_Step_Size;
}

void IDWT(void) {
    register unsigned int i, j, k;
    float *pad, *ptr_pad, *tmp, *ptr_tmp, sum;
    unsigned int n;
    n = N_Data/pow(2, Level);
    for (i=0; i<Level; i++) {
        pad = (float *)calloc(2*n+N_QMF-1, sizeof(float));
        tmp = (float *)calloc(2*n, sizeof(float));
        ptr_pad = pad + N_QMF - 1; Ptr_Data = Data;
        for (j=0; j<n; j++) {
            *ptr_pad++ = *Ptr_Data++;
            ptr_pad++;
        }
        ptr_pad = pad + N_QMF - 3; Ptr_Data = Data + n - 1;
        for (j=0; j<(N_QMF/2)-1; j++) {
            *ptr_pad-- = *Ptr_Data--;
            ptr_pad--;
        }
        ptr_tmp = tmp;
        for (j=0; j<2*n; j++) {
            sum = 0.0;
            Ptr_QMF = QMF + N_QMF - 1; ptr_pad = pad + j;
            for (k=0; k<N_QMF; k++) sum += (*Ptr_QMF--) * (*ptr_pad++);
            *ptr_tmp++ = sum;
        }
        Ptr_Data = Data + n; ptr_pad = pad + 1;
        for (j=0; j<n; j++) {
            *ptr_pad++ = *Ptr_Data++;
            ptr_pad++;
        }
    }
}

```

```

Ptr_Data = Data + n; ptr_pad = pad + 2*n + 1;
for (j=0; j<(N_QMF/2)-1; j++) {
    *ptr_pad++ = *Ptr_Data++;
    ptr_pad++;
}
ptr_tmp = tmp; Ptr_Data = Data;
for (j=0; j<2*n; j++) {
    sum = 0.0; Ptr_QMF = QMF; ptr_pad = pad + j;
    for (k=0; k<N_QMF; k++) sum += pow(-1,k) * (*Ptr_QMF++) * (*ptr_pad++);
    *Ptr_Data++ = (*ptr_tmp++) + sum;
}
free(tmp); free(pad);
n *= 2;
}
}

void Zero_Mean_Reset(void) {
    register unsigned int i;
    fread(&temp, sizeof(temp), 1, fptr_r);
    Ptr_Data = Data;
    for (i=0; i<N_Data; i++) (*Ptr_Data++) += temp;
}

```

ภาคผนวก ง

แบบสอบถามความคิดเห็นจากแพทย์

เนื่องจากในงานวิจัยนี้เป็นการลดขนาดข้อมูลสัญญาณทางการแพทย์แบบที่มีการสูญเสีย (Lossy Compression) ซึ่งทำให้เกิดความเพี้ยนของสัญญาณขึ้นที่ CR คงที่ๆค่าต่างๆ ดังนั้นจึงจำเป็นต้องให้แพทย์เป็นผู้พิจารณาเพื่อหาข้อสรุปของงานวิจัยที่ว่า เวฟเลตชนิดใดและที่ค่า CR เท่าใดที่เหมาะสมในการนำไปใช้งานได้อย่างมีประสิทธิภาพที่สุด จึงได้ทำแบบสอบถามความคิดเห็นจากแพทย์จำนวนทั้งหมด 5 ท่านถึงผลการพิจารณาสัญญาณไฟฟ้าหัวใจที่ผ่านการลดขนาดข้อมูลโดยใช้เวฟเลตแบบต่างๆ ที่ CR คงที่ๆค่าต่างๆ รวมถึงค่า PRD ที่เกิดขึ้นด้วย ภายในแบบสอบถามคำว่า “วิธีที่” และ “แบบที่” มีความหมายดังต่อไปนี้

วิธีที่ 1 หมายถึง การใช้เวฟเลตตระกูล Daubechies

แบบที่ 1 หมายถึง การใช้เวฟเลตแบบ Daubechies-4

แบบที่ 2 หมายถึง การใช้เวฟเลตแบบ Daubechies-12

แบบที่ 3 หมายถึง การใช้เวฟเลตแบบ Daubechies-20

วิธีที่ 2 หมายถึง การใช้เวฟเลตตระกูล Symmlet

แบบที่ 1 หมายถึง การใช้เวฟเลตแบบ Symmlet-4

แบบที่ 2 หมายถึง การใช้เวฟเลตแบบ Symmlet-7

แบบที่ 3 หมายถึง การใช้เวฟเลตแบบ Symmlet-10

วิธีที่ 3 หมายถึง การใช้เวฟเลตตระกูล Coiflet

แบบที่ 1 หมายถึง การใช้เวฟเลตแบบ Coiflet-1

แบบที่ 2 หมายถึง การใช้เวฟเลตแบบ Coiflet-3

แบบที่ 3 หมายถึง การใช้เวฟเลตแบบ Coiflet-5

แบบสอบถามความคิดเห็นสำหรับแพทย์แต่ละท่านมีจำนวนคำถามทั้งหมด 8 ข้อ โดยข้อที่ 1 และ 2 เป็นการพิจารณาสัญญาณไฟฟ้าหัวใจปกติทั่วไปที่ผ่านการลดขนาดข้อมูลที่ CR = 6 ในรูปที่ 6.8 ถึงรูปที่ 6.10 ข้อที่ 3 และ 4 เป็นการพิจารณาสัญญาณไฟฟ้าหัวใจปกติทั่วไปที่ผ่านการลดขนาดข้อมูลที่ CR = 9 ในรูปที่ 6.11 ถึงรูปที่ 6.13 และข้อที่ 5 และ 6 เป็นการพิจารณาสัญญาณไฟฟ้าหัวใจปกติทั่วไปที่ผ่านการลดขนาดข้อมูลที่ CR = 12 ในรูปที่ 6.14 ถึงรูปที่ 6.16 ข้อที่ 7 เป็นคำถามที่ใช้ในการสรุปคำตอบในข้อที่ 1 ถึงข้อที่ 6 ส่วนข้อที่ 8 เป็นการพิจารณาสัญญาณไฟฟ้าหัวใจผิดปกติที่ผ่านการลดขนาดข้อมูลที่ CR ค่าต่างๆ โดยคำถามข้อ 8.1 หมายถึง เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพิจารณาสัญญาณไฟฟ้าหัวใจผิดปกติแบบ Premature Ventricular Contraction ที่ผ่านการลดขนาดข้อมูลที่ CR เป็น 6, 9 และ 12 ในรูปที่ 6.25(ก) รูปที่ 6.26(ก) และรูปที่ 6.27(ก) ตามลำดับ และคำถามข้อ 8.2 หมายถึง การพิจารณาสัญญาณไฟฟ้าหัวใจผิดปกติแบบ Atrial Flutter ที่ผ่านการลดขนาดข้อมูลที่ CR เป็น 6, 9 และ 12 ในรูปที่ 6.30(ก) รูปที่ 6.31(ก) และรูปที่ 6.32(ก) ตามลำดับ ผลจากการสอบถามความคิดเห็นจากแพทย์ทั้งหมด 5 ท่านด้วยแสดงหน้าถัดไป





ความคิดเห็นจากแพทย์ท่านที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ ECG ที่ผ่านการลดขนาดข้อมูลลง 6 เท่า

1. สัญญาณ ECG ที่ได้จากการลดขนาดข้อมูลแบบใด(แบบที่ 1, 2 และ 3)ในแต่ละวิธี ที่สามารถรักษาสาระสำคัญของสัญญาณ ECG ด้นแบบที่ได้ดีที่สุด ให้กรอกหมายเลขเรียงลำดับจากหมายเลข 1 (ดีที่สุด)ไปจนถึงหมายเลข 3 (แย่ที่สุด) ถ้าพิจารณาแล้วสัญญาณมีคุณภาพใกล้เคียงกันให้กรอกหมายเลขเหมือนกัน ถ้าพิจารณาแล้วไม่สามารถนำไปใช้งานได้ให้กรอกเครื่องหมาย X

1.1 วิธีที่ 1

☐

แบบที่ 1

☒

แบบที่ 2

☐

แบบที่ 3

1.2 วิธีที่ 2

☐

แบบที่ 1

☒

แบบที่ 2

☐

แบบที่ 3

1.3 วิธีที่ 3

☐

แบบที่ 1

☒

แบบที่ 2

☐

แบบที่ 3

2. สัญญาณ ECG ที่ได้จากการลดขนาดข้อมูลวิธีใด(วิธีที่ 1, 2 และ 3)ในแต่ละแบบ ที่สามารถรักษาสาระสำคัญของสัญญาณ ECG ด้นแบบที่ได้ดีที่สุด ให้กรอกหมายเลขเรียงลำดับจากหมายเลข 1 (ดีที่สุด)ไปจนถึงหมายเลข 3 (แย่ที่สุด) ถ้าพิจารณาแล้วสัญญาณมีคุณภาพใกล้เคียงกันให้กรอกหมายเลขเหมือนกัน ถ้าพิจารณาแล้วไม่สามารถนำไปใช้งานได้ให้กรอกเครื่องหมาย X

2.1 แบบที่ 1

☐

วิธีที่ 1

☒

วิธีที่ 2

☒

วิธีที่ 3

2.2 แบบที่ 2

☐

วิธีที่ 1

☐

วิธีที่ 2

☐

วิธีที่ 3

2.3 แบบที่ 3

☒

วิธีที่ 1

☐

วิธีที่ 2

☒

วิธีที่ 3

สัญญาณ ECG ที่ผ่านการลดขนาดข้อมูลลง 9 เท่า

3. สัญญาณ ECG ที่ได้จากการลดขนาดข้อมูลแบบใด(แบบที่ 1, 2 และ 3)ในแต่ละวิธี ที่สามารถรักษาสาระสำคัญของสัญญาณ ECG ด้นแบบที่ได้ดีที่สุด ให้กรอกหมายเลขเรียงลำดับจากหมายเลข 1 (ดีที่สุด)ไปจนถึงหมายเลข 3 (แย่ที่สุด) ถ้าพิจารณาแล้วสัญญาณมีคุณภาพใกล้เคียงกันให้กรอกหมายเลขเหมือนกัน ถ้าพิจารณาแล้วไม่สามารถนำไปใช้งานได้ให้กรอกเครื่องหมาย X

3.1 วิธีที่ 1 ☒ แบบที่ 1 ☒ แบบที่ 2 ☒ แบบที่ 3

3.2 วิธีที่ 2 ☒ แบบที่ 1 ☒ แบบที่ 2 ☒ แบบที่ 3

3.3 วิธีที่ 3 ☒ แบบที่ 1 ☒ แบบที่ 2 ☒ แบบที่ 3

4. สัญญาณ ECG ที่ได้จากการลดขนาดข้อมูลวิธีใด(วิธีที่ 1, 2 และ 3)ในแต่ละแบบ ที่สามารถรักษาสาระสำคัญของสัญญาณ ECG ด้นแบบที่ได้ดีที่สุด ให้กรอกหมายเลขเรียงลำดับจากหมายเลข 1 (ดีที่สุด)ไปจนถึงหมายเลข 3 (แย่ที่สุด) ถ้าพิจารณาแล้วสัญญาณมีคุณภาพใกล้เคียงกันให้กรอกหมายเลขเหมือนกัน ถ้าพิจารณาแล้วไม่สามารถนำไปใช้งานได้ให้กรอกเครื่องหมาย X

4.1 แบบที่ 1 ☒ วิธีที่ 1 ☒ วิธีที่ 2 ☒ วิธีที่ 3

4.2 แบบที่ 2 ☒ วิธีที่ 1 ☒ วิธีที่ 2 ☒ วิธีที่ 3

4.3 แบบที่ 3 ☒ วิธีที่ 1 ☒ วิธีที่ 2 ☒ วิธีที่ 3

สัญญาณ ECG ที่ผ่านการลดขนาดข้อมูลลง 12 เท่า

5. สัญญาณ ECG ที่ได้จากการลดขนาดข้อมูลแบบใด(แบบที่ 1, 2 และ 3)ในแต่ละวิธี ที่สามารถรักษาสาระสำคัญของสัญญาณ ECG ด้นแบบที่ได้ดีที่สุด ให้กรอกหมายเลขเรียงลำดับจากหมายเลข 1 (ดีที่สุด)ไปจนถึงหมายเลข 3 (แย่ที่สุด) ถ้าพิจารณาแล้วสัญญาณมีคุณภาพใกล้เคียงกันให้กรอกหมายเลขเหมือนกัน ถ้าพิจารณาแล้วไม่สามารถนำไปใช้งานได้ให้กรอกเครื่องหมาย X

5.1 วิธีที่ 1 ☒ แบบที่ 1 ☒ แบบที่ 2 ☒ แบบที่ 3

5.2 วิธีที่ 2 ☒ แบบที่ 1 ☒ แบบที่ 2 ☒ แบบที่ 3

5.3 วิธีที่ 3 ☒ แบบที่ 1 ☒ แบบที่ 2 ☒ แบบที่ 3

6. สัญญาณ ECG ที่ได้จากการลดขนาดข้อมูลวิธีใด(วิธีที่ 1, 2 และ 3)ในแต่ละแบบ ที่สามารถรักษาสาระสำคัญของสัญญาณ ECG ด้นแบบที่ได้ดีที่สุด ให้กรอกหมายเลขเรียงลำดับจากหมายเลข 1 (ดีที่สุด)ไปจนถึงหมายเลข 3 (แย่ที่สุด) ถ้าพิจารณาแล้วสัญญาณมีคุณภาพใกล้เคียงกันให้กรอกหมายเลขเหมือนกัน ถ้าพิจารณาแล้วไม่สามารถนำไปใช้งานได้ให้กรอกเครื่องหมาย X

6.1 แบบที่ 1 ☒ วิธีที่ 1 ☒ วิธีที่ 2 ☒ วิธีที่ 3

6.2 แบบที่ 2 ☒ วิธีที่ 1 ☒ วิธีที่ 2 ☒ วิธีที่ 3

6.3 แบบที่ 3 ☒ วิธีที่ 1 ☒ วิธีที่ 2 ☒ วิธีที่ 3

สรุปจากเอกสารชุดที่ 1, 2 และ 3

7. ที่อัตราส่วนการลดขนาดข้อมูลสูงสุดเท่าใด วิธีใดและแบบใด ที่แพทย์ผู้วินิจฉัยมีความเห็นว่า ทำให้สัญญาณ ECG ที่ผ่านการลดขนาดข้อมูลมีความเพี้ยนอยู่ในระดับที่ทางการแพทย์ยอมรับว่า สามารถวินิจฉัยได้ผลเช่นเดียวสัญญาณ ECG ดั้งแบบ

☒ 6 เท่า

วิธีที่ 1 แบบที่ 1-3

วิธีที่ 2 แบบที่ 1-3

วิธีที่ 3 แบบที่ 1-3

☐ 9 เท่า

วิธีที่ 1 แบบที่ 1-3

วิธีที่ 2 แบบที่ 1-3

วิธีที่ 3 แบบที่ 2,3

☐ 12 เท่า

วิธีที่ — แบบที่ —

วิธีที่ — แบบที่ —

วิธีที่ — แบบที่ —

หมายเหตุ กรอกคำตอบได้มากกว่า 1 คำตอบ ตัวอย่างเช่น

☒ 6 เท่า

วิธีที่ 1 แบบที่ 2,3

วิธีที่ 3 แบบที่ 1,2

วิธีที่ — แบบที่ —



ความคิดเห็นจากแพทย์ท่านที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ ECG ที่ผ่านการลดขนาดข้อมูลลง 6 เท่า

1. สัญญาณ ECG ที่ได้จากการลดขนาดข้อมูลแบบใด(แบบที่ 1, 2 และ 3)ในแต่ละวิธี ที่สามารถรักษาสาระสำคัญของสัญญาณ ECG ด้นแบบที่ได้ดีที่สุด ให้กรอกหมายเลขเรียงลำดับจากหมายเลข 1 (ดีที่สุด)ไปจนถึงหมายเลข 3 (แย่ที่สุด) ถ้าพิจารณาแล้วสัญญาณมีคุณภาพใกล้เคียงกันให้กรอกหมายเลขเหมือนกัน ถ้าพิจารณาแล้วไม่สามารถนำไปใช้งานได้ให้กรอกเครื่องหมาย X

1.1 วิธีที่ 1 แบบที่ 1 แบบที่ 2 แบบที่ 3

1.2 วิธีที่ 2 แบบที่ 1 แบบที่ 2 แบบที่ 3

1.3 วิธีที่ 3 แบบที่ 1 แบบที่ 2 แบบที่ 3

2. สัญญาณ ECG ที่ได้จากการลดขนาดข้อมูลวิธีใด(วิธีที่ 1, 2 และ 3)ในแต่ละแบบ ที่สามารถรักษาสาระสำคัญของสัญญาณ ECG ด้นแบบที่ได้ดีที่สุด ให้กรอกหมายเลขเรียงลำดับจากหมายเลข 1 (ดีที่สุด)ไปจนถึงหมายเลข 3 (แย่ที่สุด) ถ้าพิจารณาแล้วสัญญาณมีคุณภาพใกล้เคียงกันให้กรอกหมายเลขเหมือนกัน ถ้าพิจารณาแล้วไม่สามารถนำไปใช้งานได้ให้กรอกเครื่องหมาย X

2.1 แบบที่ 1 วิธีที่ 1 วิธีที่ 2 วิธีที่ 3

2.2 แบบที่ 2 วิธีที่ 1 วิธีที่ 2 วิธีที่ 3

2.3 แบบที่ 3 วิธีที่ 1 วิธีที่ 2 วิธีที่ 3

สัญญาณ ECG ที่ผ่านการลดขนาดข้อมูลลง 9 เท่า

3. สัญญาณ ECG ที่ได้จากการลดขนาดข้อมูลแบบใด(แบบที่ 1, 2 และ 3)ในแต่ละวิธี ที่สามารถรักษาสาระสำคัญของสัญญาณ ECG ด้นแบบที่ได้ดีที่สุด ให้กรอกหมายเลขเรียงลำดับจากหมายเลข 1 (ดีที่สุด)ไปจนถึงหมายเลข 3 (แย่ที่สุด) ถ้าพิจารณาแล้วสัญญาณมีคุณภาพใกล้เคียงกันให้กรอกหมายเลขเหมือนกัน ถ้าพิจารณาแล้วไม่สามารถนำไปใช้งานได้ให้กรอกเครื่องหมาย X

3.1 วิธีที่ 1 ☐ แบบที่ 1 ☒ แบบที่ 2 ☒ แบบที่ 3

3.2 วิธีที่ 2 ☒ แบบที่ 1 ☒ แบบที่ 2 ☒ แบบที่ 3

3.3 วิธีที่ 3 ☒ แบบที่ 1 ☒ แบบที่ 2 ☒ แบบที่ 3

4. สัญญาณ ECG ที่ได้จากการลดขนาดข้อมูลวิธีใด(วิธีที่ 1, 2 และ 3)ในแต่ละแบบ ที่สามารถรักษาสาระสำคัญของสัญญาณ ECG ด้นแบบที่ได้ดีที่สุด ให้กรอกหมายเลขเรียงลำดับจากหมายเลข 1 (ดีที่สุด)ไปจนถึงหมายเลข 3 (แย่ที่สุด) ถ้าพิจารณาแล้วสัญญาณมีคุณภาพใกล้เคียงกันให้กรอกหมายเลขเหมือนกัน ถ้าพิจารณาแล้วไม่สามารถนำไปใช้งานได้ให้กรอกเครื่องหมาย X

4.1 แบบที่ 1 ☒ วิธีที่ 1 ☒ วิธีที่ 2 ☒ วิธีที่ 3

4.2 แบบที่ 2 ☒ วิธีที่ 1 ☒ วิธีที่ 2 ☒ วิธีที่ 3

4.3 แบบที่ 3 ☒ วิธีที่ 1 ☒ วิธีที่ 2 ☒ วิธีที่ 3

สัญญาณ ECG ที่ผ่านการลดขนาดข้อมูลลง 12 เท่า

5. สัญญาณ ECG ที่ได้จากการลดขนาดข้อมูลแบบใด(แบบที่ 1, 2 และ 3)ในแต่ละวิธี ที่สามารถรักษาสาระสำคัญของสัญญาณ ECG ต้นแบบที่ได้ดีที่สุด ให้กรอกหมายเลขเรียงลำดับจากหมายเลข 1 (ดีที่สุด)ไปจนถึงหมายเลข 3 (แย่ที่สุด) ถ้าพิจารณาแล้วสัญญาณมีคุณภาพใกล้เคียงกันให้กรอกหมายเลขเหมือนกัน ถ้าพิจารณาแล้วไม่สามารถนำไปใช้งานได้ให้กรอกเครื่องหมาย X

5.1 วิธีที่ 1 ☒ แบบที่ 1 ☐ แบบที่ 2 ☒ แบบที่ 3

5.2 วิธีที่ 2 ☐ แบบที่ 1 ☒ แบบที่ 2 ☒ แบบที่ 3

5.3 วิธีที่ 3 ☒ แบบที่ 1 ☒ แบบที่ 2 ☐ แบบที่ 3

6. สัญญาณ ECG ที่ได้จากการลดขนาดข้อมูลวิธีใด(วิธีที่ 1, 2 และ 3)ในแต่ละแบบ ที่สามารถรักษาสาระสำคัญของสัญญาณ ECG ต้นแบบที่ได้ดีที่สุด ให้กรอกหมายเลขเรียงลำดับจากหมายเลข 1 (ดีที่สุด)ไปจนถึงหมายเลข 3 (แย่ที่สุด) ถ้าพิจารณาแล้วสัญญาณมีคุณภาพใกล้เคียงกันให้กรอกหมายเลขเหมือนกัน ถ้าพิจารณาแล้วไม่สามารถนำไปใช้งานได้ให้กรอกเครื่องหมาย X

6.1 แบบที่ 1 ☒ วิธีที่ 1 ☐ วิธีที่ 2 ☒ วิธีที่ 3

6.2 แบบที่ 2 ☒ วิธีที่ 1 ☐ วิธีที่ 2 ☒ วิธีที่ 3

6.3 แบบที่ 3 ☒ วิธีที่ 1 ☒ วิธีที่ 2 ☐ วิธีที่ 3

สรุปจากเอกสารชุดที่ 1, 2 และ 3

7. ที่อัตราส่วนการลดขนาดข้อมูลสูงสุดเท่าใด วิธีใดและแบบใด ที่แพทย์ผู้วินิจฉัยมีความเห็นว่า ทำให้สัญญาณ ECG ที่ผ่านการลดขนาดข้อมูลมีความเพี้ยนอยู่ในระดับที่ทางการแพทย์ยอมรับว่า สามารถวินิจฉัยได้ผลเช่นเดียวสัญญาณ ECG ดั้งเดิม

☒ 6 เท่า

วิธีที่ 1 แบบที่ 1,2,3

วิธีที่ 2 แบบที่ 1,2,3

วิธีที่ 3 แบบที่ 1,2,3

☒ 9 เท่า

วิธีที่ 1 แบบที่ 1-3

วิธีที่ 2 แบบที่ 1-3

วิธีที่ 3 แบบที่ 1-3

☒ 12 เท่า

วิธีที่ 1 แบบที่ 1-3

วิธีที่ 2 แบบที่ 1-3

วิธีที่ 3 แบบที่ 1-3

หมายเหตุ กรอกคำตอบได้มากกว่า 1 คำตอบ ตัวอย่างเช่น

☒ 6 เท่า

วิธีที่ 1 แบบที่ 2,3

วิธีที่ 3 แบบที่ 1,2

วิธีที่ แบบที่

สัญญา ECG ผิดปกติที่ผ่านการลดขนาดข้อมูลด้วยวิธีที่ 3 แบบที่ 2

8. ที่อัตราส่วนการลดขนาดข้อมูลสูงสุดเท่าใด ที่แพทย์ผู้วินิจฉัยมีความเห็นว่า ทำให้สัญญา ECG ผิดปกติที่ผ่านการลดขนาดข้อมูลมีความเพี้ยนอยู่ในระดับที่ทางการแพทย์ยอมรับว่า สามารถวินิจฉัยได้ผลเช่นเดียวสัญญา ECG ดั้งแบบ

8.1 สัญญา ECG แบบที่ 1 ☐ 6 เท่า ☒ 9 เท่า ☐ 12 เท่า

8.2 สัญญา ECG แบบที่ 2 ☐ 6 เท่า ☐ 9 เท่า ☒ 12 เท่า



ลงชื่อ.....
(..... กิจการ กิจ.....)
นายแพทย์ผู้ออกความคิดเห็น



ความคิดเห็นจากแพทย์ท่านที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ ECG ที่ผ่านการลดขนาดข้อมูลลง 6 เท่า

1. สัญญาณ ECG ที่ได้จากการลดขนาดข้อมูลแบบใด(แบบที่ 1, 2 และ 3)ในแต่ละวิธี ที่สามารถรักษาสาระสำคัญของสัญญาณ ECG ด้นแบบที่ได้ดีที่สุด ให้กรอกหมายเลขเรียงลำดับจากหมายเลข 1 (ดีที่สุด)ไปจนถึงหมายเลข 3 (แย่ที่สุด) ถ้าพิจารณาแล้วสัญญาณมีคุณภาพใกล้เคียงกันให้กรอกหมายเลขเหมือนกัน ถ้าพิจารณาแล้วไม่สามารถนำไปใช้งานได้ให้กรอกเครื่องหมาย X

1.1 วิธีที่ 1 ☐ แบบที่ 1 ☐ แบบที่ 2 ☒ แบบที่ 3

1.2 วิธีที่ 2 ☐ แบบที่ 1 ☒ แบบที่ 2 ☒ แบบที่ 3

1.3 วิธีที่ 3 ☒ แบบที่ 1 ☐ แบบที่ 2 ☒ แบบที่ 3

2. สัญญาณ ECG ที่ได้จากการลดขนาดข้อมูลวิธีใด(วิธีที่ 1, 2 และ 3)ในแต่ละแบบ ที่สามารถรักษาสาระสำคัญของสัญญาณ ECG ด้นแบบที่ได้ดีที่สุด ให้กรอกหมายเลขเรียงลำดับจากหมายเลข 1 (ดีที่สุด)ไปจนถึงหมายเลข 3 (แย่ที่สุด) ถ้าพิจารณาแล้วสัญญาณมีคุณภาพใกล้เคียงกันให้กรอกหมายเลขเหมือนกัน ถ้าพิจารณาแล้วไม่สามารถนำไปใช้งานได้ให้กรอกเครื่องหมาย X

2.1 แบบที่ 1 ☒ วิธีที่ 1 ☐ วิธีที่ 2 ☒ วิธีที่ 3

2.2 แบบที่ 2 ☒ วิธีที่ 1 ☒ วิธีที่ 2 ☐ วิธีที่ 3

2.3 แบบที่ 3 ☒ วิธีที่ 1 ☒ วิธีที่ 2 ☐ วิธีที่ 3

สัญญาณ ECG ที่ผ่านการลดขนาดข้อมูลลง 9 เท่า

3. สัญญาณ ECG ที่ได้จากการลดขนาดข้อมูลแบบใด(แบบที่ 1, 2 และ 3)ในแต่ละวิธี ที่สามารถรักษาสาระสำคัญของสัญญาณ ECG ดั้งแบบที่ได้ดีที่สุดที่สุด ให้กรอกหมายเลขเรียงลำดับจากหมายเลข 1 (ดีที่สุด)ไปจนถึงหมายเลข 3 (แย่ที่สุด) ถ้าพิจารณาแล้วสัญญาณมีคุณภาพใกล้เคียงกันให้กรอกหมายเลขเหมือนกัน ถ้าพิจารณาแล้วไม่สามารถนำไปใช้งานได้ให้กรอกเครื่องหมาย X

3.1 วิธีที่ 1 ☒ แบบที่ 1 ☒ แบบที่ 2 ☒ แบบที่ 3

3.2 วิธีที่ 2 ☒ แบบที่ 1 ☒ แบบที่ 2 ☒ แบบที่ 3

3.3 วิธีที่ 3 ☒ แบบที่ 1 ☒ แบบที่ 2 ☒ แบบที่ 3

4. สัญญาณ ECG ที่ได้จากการลดขนาดข้อมูลวิธีใด(วิธีที่ 1, 2 และ 3)ในแต่ละแบบ ที่สามารถรักษาสาระสำคัญของสัญญาณ ECG ดั้งแบบที่ได้ดีที่สุดที่สุด ให้กรอกหมายเลขเรียงลำดับจากหมายเลข 1 (ดีที่สุด)ไปจนถึงหมายเลข 3 (แย่ที่สุด) ถ้าพิจารณาแล้วสัญญาณมีคุณภาพใกล้เคียงกันให้กรอกหมายเลขเหมือนกัน ถ้าพิจารณาแล้วไม่สามารถนำไปใช้งานได้ให้กรอกเครื่องหมาย X

4.1 แบบที่ 1 ☒ วิธีที่ 1 ☒ วิธีที่ 2 ☒ วิธีที่ 3

4.2 แบบที่ 2 ☒ วิธีที่ 1 ☒ วิธีที่ 2 ☒ วิธีที่ 3

4.3 แบบที่ 3 ☒ วิธีที่ 1 ☒ วิธีที่ 2 ☒ วิธีที่ 3

สัญญาณ ECG ที่ผ่านการลดขนาดข้อมูลลง 12 เท่า

5. สัญญาณ ECG ที่ได้จากการลดขนาดข้อมูลแบบใด(แบบที่ 1, 2 และ 3)ในแต่ละวิธี ที่สามารถรักษาสาระสำคัญของสัญญาณ ECG ต้นแบบที่ได้ดีที่สุด ให้กรอกหมายเลขเรียงลำดับจากหมายเลข 1 (ดีที่สุด)ไปจนถึงหมายเลข 3 (แย่ที่สุด) ถ้าพิจารณาแล้วสัญญาณมีคุณภาพใกล้เคียงกันให้กรอกหมายเลขเหมือนกัน ถ้าพิจารณาแล้วไม่สามารถนำไปใช้งานได้ให้กรอกเครื่องหมาย X

5.1 วิธีที่ 1 ☐ แบบที่ 1 ☒ แบบที่ 2 ☒ แบบที่ 3

5.2 วิธีที่ 2 ☒ แบบที่ 1 ☒ แบบที่ 2 ☒ แบบที่ 3

5.3 วิธีที่ 3 ☒ แบบที่ 1 ☒ แบบที่ 2 ☒ แบบที่ 3

6. สัญญาณ ECG ที่ได้จากการลดขนาดข้อมูลวิธีใด(วิธีที่ 1, 2 และ 3)ในแต่ละแบบ ที่สามารถรักษาสาระสำคัญของสัญญาณ ECG ต้นแบบที่ได้ดีที่สุด ให้กรอกหมายเลขเรียงลำดับจากหมายเลข 1 (ดีที่สุด)ไปจนถึงหมายเลข 3 (แย่ที่สุด) ถ้าพิจารณาแล้วสัญญาณมีคุณภาพใกล้เคียงกันให้กรอกหมายเลขเหมือนกัน ถ้าพิจารณาแล้วไม่สามารถนำไปใช้งานได้ให้กรอกเครื่องหมาย X

6.1 แบบที่ 1 ☒ วิธีที่ 1 ☒ วิธีที่ 2 ☒ วิธีที่ 3

6.2 แบบที่ 2 ☒ วิธีที่ 1 ☒ วิธีที่ 2 ☒ วิธีที่ 3

6.3 แบบที่ 3 ☒ วิธีที่ 1 ☒ วิธีที่ 2 ☒ วิธีที่ 3

สรุปจากเอกสารชุดที่ 1, 2 และ 3

7. ที่อัตราส่วนการลดขนาดข้อมูลสูงสุดเท่าใด วิธีใดและแบบใด ที่แพทย์ผู้วินิจฉัยมีความเห็นว่า ทำให้สัญญาณ ECG ที่ผ่านการลดขนาดข้อมูลมีความเพี้ยนอยู่ในระดับที่ทางการแพทย์ยอมรับว่า สามารถวินิจฉัยได้ผลเช่นเดียวสัญญาณ ECG ต้นแบบ

☒ 6 เท่า

วิธีที่ 1 แบบที่ 1-3

วิธีที่ 2 แบบที่ 1-3

วิธีที่ 3 แบบที่ 1-3

☐ 9 เท่า

วิธีที่ แบบที่

วิธีที่ แบบที่

วิธีที่ แบบที่

☐ 12 เท่า

วิธีที่ แบบที่

วิธีที่ แบบที่

วิธีที่ แบบที่

หมายเหตุ กรอกราคำตอบได้มากกว่า 1 คำตอบ ตัวอย่างเช่น

☒ 6 เท่า

วิธีที่ 1 แบบที่ 2,3

วิธีที่ 3 แบบที่ 1,2

วิธีที่ แบบที่

สัญญาณ ECG ผิดปกติที่ผ่านการลดขนาดข้อมูลด้วยวิธีที่ 3 แบบที่ 2

8. ที่อัตราส่วนการลดขนาดข้อมูลสูงสุดเท่าใด ที่แพทย์ผู้วินิจฉัยมีความเห็นว่า ทำให้สัญญาณ ECG ผิดปกติที่ผ่านการลดขนาดข้อมูลมีความเพี้ยนอยู่ในระดับที่ทางการแพทย์ยอมรับว่า สามารถวินิจฉัยได้ผลเช่นเดียวสัญญาณ ECG ดั้งเดิม

8.1 สัญญาณ ECG แบบที่ 1 ☐ 6 เท่า ☒ 9 เท่า ☐ 12 เท่า

8.2 สัญญาณ ECG แบบที่ 2 ☐ 6 เท่า ☐ 9 เท่า ☒ 12 เท่า





ความคิดเห็นจากแพทย์ท่านที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ ECG ที่ผ่านการลดขนาดข้อมูลลง 6 เท่า

1. สัญญาณ ECG ที่ได้จากการลดขนาดข้อมูลแบบใด(แบบที่ 1, 2 และ 3)ในแต่ละวิธี ที่สามารถรักษาสาระสำคัญของสัญญาณ ECG ต้นแบบที่ได้ดีที่สุด ให้กรอกหมายเลขเรียงลำดับจากหมายเลข 1 (ดีที่สุด)ไปจนถึงหมายเลข 3 (แย่ที่สุด) ถ้าพิจารณาแล้วสัญญาณมีคุณภาพใกล้เคียงกันให้กรอกหมายเลขเหมือนกัน ถ้าพิจารณาแล้วไม่สามารถนำไปใช้งานได้ให้กรอกเครื่องหมาย X

1.1 วิธีที่ 1 ☒ แบบที่ 1 ☐ แบบที่ 2 ☒ แบบที่ 3

1.2 วิธีที่ 2 ☐ แบบที่ 1 ☐ แบบที่ 2 ☒ แบบที่ 3

1.3 วิธีที่ 3 ☒ แบบที่ 1 ☐ แบบที่ 2 ☒ แบบที่ 3

2. สัญญาณ ECG ที่ได้จากการลดขนาดข้อมูลวิธีใด(วิธีที่ 1, 2 และ 3)ในแต่ละแบบ ที่สามารถรักษาสาระสำคัญของสัญญาณ ECG ต้นแบบที่ได้ดีที่สุด ให้กรอกหมายเลขเรียงลำดับจากหมายเลข 1 (ดีที่สุด)ไปจนถึงหมายเลข 3 (แย่ที่สุด) ถ้าพิจารณาแล้วสัญญาณมีคุณภาพใกล้เคียงกันให้กรอกหมายเลขเหมือนกัน ถ้าพิจารณาแล้วไม่สามารถนำไปใช้งานได้ให้กรอกเครื่องหมาย X

2.1 แบบที่ 1 ☐ วิธีที่ 1 ☐ วิธีที่ 2 ☒ วิธีที่ 3

2.2 แบบที่ 2 ☒ วิธีที่ 1 ☒ วิธีที่ 2 ☐ วิธีที่ 3

2.3 แบบที่ 3 ☒ วิธีที่ 1 ☒ วิธีที่ 2 ☐ วิธีที่ 3

สัญญาณ ECG ที่ผ่านการลดขนาดข้อมูลลง 9 เท่า

3. สัญญาณ ECG ที่ได้จากการลดขนาดข้อมูลแบบใด(แบบที่ 1, 2 และ 3)ในแต่ละวิธี ที่สามารถรักษาสาระสำคัญของสัญญาณ ECG ต้นแบบที่ได้ดีที่สุด ให้กรอกหมายเลขเรียงลำดับจากหมายเลข 1 (ดีที่สุด)ไปจนถึงหมายเลข 3 (แย่ที่สุด) ถ้าพิจารณาแล้วสัญญาณมีคุณภาพใกล้เคียงกันให้กรอกหมายเลขเหมือนกัน ถ้าพิจารณาแล้วไม่สามารถนำไปใช้งานได้ให้กรอกเครื่องหมาย X

3.1 วิธีที่ 1 ☒ แบบที่ 1 ☒ แบบที่ 2 ☒ แบบที่ 3

3.2 วิธีที่ 2 ☒ แบบที่ 1 ☒ แบบที่ 2 ☒ แบบที่ 3

3.3 วิธีที่ 3 ☒ แบบที่ 1 ☒ แบบที่ 2 ☒ แบบที่ 3

4. สัญญาณ ECG ที่ได้จากการลดขนาดข้อมูลวิธีใด(วิธีที่ 1, 2 และ 3)ในแต่ละแบบ ที่สามารถรักษาสาระสำคัญของสัญญาณ ECG ต้นแบบที่ได้ดีที่สุด ให้กรอกหมายเลขเรียงลำดับจากหมายเลข 1 (ดีที่สุด)ไปจนถึงหมายเลข 3 (แย่ที่สุด) ถ้าพิจารณาแล้วสัญญาณมีคุณภาพใกล้เคียงกันให้กรอกหมายเลขเหมือนกัน ถ้าพิจารณาแล้วไม่สามารถนำไปใช้งานได้ให้กรอกเครื่องหมาย X

4.1 แบบที่ 1 ☒ วิธีที่ 1 ☒ วิธีที่ 2 ☒ วิธีที่ 3

4.2 แบบที่ 2 ☒ วิธีที่ 1 ☒ วิธีที่ 2 ☒ วิธีที่ 3

4.3 แบบที่ 3 ☒ วิธีที่ 1 ☒ วิธีที่ 2 ☒ วิธีที่ 3

สัญญาณ ECG ที่ผ่านการลดขนาดข้อมูลลง 12 เท่า

5. สัญญาณ ECG ที่ได้จากการลดขนาดข้อมูลแบบใด(แบบที่ 1, 2 และ 3)ในแต่ละวิธี ที่สามารถรักษาสาระสำคัญของสัญญาณ ECG ต้นแบบที่ได้ดีที่สุด ให้กรอกหมายเลขเรียงลำดับจากหมายเลข 1 (ดีที่สุด)ไปจนถึงหมายเลข 3 (แย่ที่สุด) ถ้าพิจารณาแล้วสัญญาณมีคุณภาพใกล้เคียงกันให้กรอกหมายเลขเหมือนกัน ถ้าพิจารณาแล้วไม่สามารถนำไปใช้งานได้ให้กรอกเครื่องหมาย X

5.1 วิธีที่ 1 ☒ แบบที่ 1 ☒ แบบที่ 2 ☒ แบบที่ 3

5.2 วิธีที่ 2 ☒ แบบที่ 1 ☒ แบบที่ 2 ☒ แบบที่ 3

5.3 วิธีที่ 3 ☒ แบบที่ 1 ☒ แบบที่ 2 ☒ แบบที่ 3

6. สัญญาณ ECG ที่ได้จากการลดขนาดข้อมูลวิธีใด(วิธีที่ 1, 2 และ 3)ในแต่ละแบบ ที่สามารถรักษาสาระสำคัญของสัญญาณ ECG ต้นแบบที่ได้ดีที่สุด ให้กรอกหมายเลขเรียงลำดับจากหมายเลข 1 (ดีที่สุด)ไปจนถึงหมายเลข 3 (แย่ที่สุด) ถ้าพิจารณาแล้วสัญญาณมีคุณภาพใกล้เคียงกันให้กรอกหมายเลขเหมือนกัน ถ้าพิจารณาแล้วไม่สามารถนำไปใช้งานได้ให้กรอกเครื่องหมาย X

6.1 แบบที่ 1 ☒ วิธีที่ 1 ☒ วิธีที่ 2 ☒ วิธีที่ 3

6.2 แบบที่ 2 ☒ วิธีที่ 1 ☒ วิธีที่ 2 ☒ วิธีที่ 3

6.3 แบบที่ 3 ☒ วิธีที่ 1 ☒ วิธีที่ 2 ☒ วิธีที่ 3

สรุปจากเอกสารชุดที่ 1, 2 และ 3

7. ที่อัตราส่วนการลดขนาดข้อมูลสูงสุดเท่าใด วิธีใดและแบบใด ที่แพทย์ผู้วินิจฉัยมีความเห็นว่า ทำให้สัญญาณ ECG ที่ผ่านการลดขนาดข้อมูลมีความเพิ่มขึ้นอยู่ในระดับที่ทางการแพทย์ยอมรับว่า สามารถวินิจฉัยได้ผลเช่นเดียวสัญญาณ ECG ดั้งเดิม

☒ 6 เท่า

วิธีที่ 1 แบบที่ 1-3

วิธีที่ 2 แบบที่ 1-3

วิธีที่ 3 แบบที่ 1-3

☐ 9 เท่า

วิธีที่ แบบที่

วิธีที่ แบบที่

วิธีที่ แบบที่

☐ 12 เท่า

วิธีที่ แบบที่

วิธีที่ แบบที่

วิธีที่ แบบที่

หมายเหตุ กรอกคำตอบได้มากกว่า 1 คำตอบ ตัวอย่างเช่น

☒ 6 เท่า

วิธีที่ 1 แบบที่ 2,3

วิธีที่ 3 แบบที่ 1,2

วิธีที่ แบบที่

สัญญาณ ECG ผิดปกติที่ผ่านการลดขนาดข้อมูลด้วยวิธีที่ 3 แบบที่ 2

8. ที่อัตราส่วนการลดขนาดข้อมูลสูงสุดเท่าใด ที่แพทย์ผู้วินิจฉัยมีความเห็นว่า ทำให้สัญญาณ ECG ผิดปกติที่ผ่านการลดขนาดข้อมูลมีความเพี้ยนอยู่ในระดับที่ทางการแพทย์ยอมรับว่า สามารถวินิจฉัยได้ผลเช่นเดียวสัญญาณ ECG ดั้งเดิม

8.1 สัญญาณ ECG แบบที่ 1

☐ 6 เท่า

☒ 9 เท่า

☐ 12 เท่า

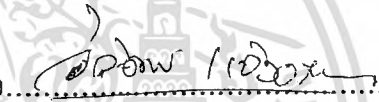
8.2 สัญญาณ ECG แบบที่ 2

☐ 6 เท่า

☐ 9 เท่า

☒ 12 เท่า

ลงชื่อ.....



(.....)

นายแพทย์ผู้ออกความคิดเห็น



สัญญาณ ECG ที่ผ่านการลดขนาดข้อมูลลง 6 เท่า

1. สัญญาณ ECG ที่ได้จากการลดขนาดข้อมูลแบบใด(แบบที่ 1, 2 และ 3)ในแต่ละวิธี ที่สามารถรักษาสาระสำคัญของสัญญาณ ECG ต้นแบบที่ได้ดีที่สุด ให้กรอกหมายเลขเรียงลำดับจากหมายเลข 1 (ดีที่สุด)ไปจนถึงหมายเลข 3 (แย่ที่สุด) ถ้าพิจารณาแล้วสัญญาณมีคุณภาพใกล้เคียงกันให้กรอกหมายเลขเหมือนกัน ถ้าพิจารณาแล้วไม่สามารถนำไปใช้งานได้ให้กรอกเครื่องหมาย X

1.1 วิธีที่ 1 ☒ แบบที่ 1 ☐ แบบที่ 2 ☐ แบบที่ 3

1.2 วิธีที่ 2 ☐ แบบที่ 1 ☐ แบบที่ 2 ☐ แบบที่ 3

1.3 วิธีที่ 3 ☒ แบบที่ 1 ☐ แบบที่ 2 ☐ แบบที่ 3

2. สัญญาณ ECG ที่ได้จากการลดขนาดข้อมูลวิธีใด(วิธีที่ 1, 2 และ 3)ในแต่ละแบบ ที่สามารถรักษาสาระสำคัญของสัญญาณ ECG ต้นแบบที่ได้ดีที่สุด ให้กรอกหมายเลขเรียงลำดับจากหมายเลข 1 (ดีที่สุด)ไปจนถึงหมายเลข 3 (แย่ที่สุด) ถ้าพิจารณาแล้วสัญญาณมีคุณภาพใกล้เคียงกันให้กรอกหมายเลขเหมือนกัน ถ้าพิจารณาแล้วไม่สามารถนำไปใช้งานได้ให้กรอกเครื่องหมาย X

2.1 แบบที่ 1 ☐ วิธีที่ 1 ☐ วิธีที่ 2 ☐ วิธีที่ 3

2.2 แบบที่ 2 ☐ วิธีที่ 1 ☐ วิธีที่ 2 ☐ วิธีที่ 3

2.3 แบบที่ 3 ☐ วิธีที่ 1 ☐ วิธีที่ 2 ☐ วิธีที่ 3

สัญญาณ ECG ที่ผ่านการลดขนาดข้อมูลลง 9 เท่า

3. สัญญาณ ECG ที่ได้จากการลดขนาดข้อมูลแบบใด(แบบที่ 1, 2 และ 3)ในแต่ละวิธี ที่สามารถรักษาสาระสำคัญของสัญญาณ ECG ต้นแบบที่ได้ดีที่สุด ให้กรอกหมายเลขเรียงลำดับจากหมายเลข 1 (ดีที่สุด)ไปจนถึงหมายเลข 3 (แย่ที่สุด) ถ้าพิจารณาแล้วสัญญาณมีคุณภาพใกล้เคียงกันให้กรอกหมายเลขเหมือนกัน ถ้าพิจารณาแล้วไม่สามารถนำไปใช้งานได้ให้กรอกเครื่องหมาย X

3.1 วิธีที่ 1 ☒ แบบที่ 1 ☒ แบบที่ 2 ☒ แบบที่ 3

3.2 วิธีที่ 2 ☒ แบบที่ 1 ☒ แบบที่ 2 ☒ แบบที่ 3

3.3 วิธีที่ 3 ☒ แบบที่ 1 ☒ แบบที่ 2 ☒ แบบที่ 3

4. สัญญาณ ECG ที่ได้จากการลดขนาดข้อมูลวิธีใด(วิธีที่ 1, 2 และ 3)ในแต่ละแบบ ที่สามารถรักษาสาระสำคัญของสัญญาณ ECG ต้นแบบที่ได้ดีที่สุด ให้กรอกหมายเลขเรียงลำดับจากหมายเลข 1 (ดีที่สุด)ไปจนถึงหมายเลข 3 (แย่ที่สุด) ถ้าพิจารณาแล้วสัญญาณมีคุณภาพใกล้เคียงกันให้กรอกหมายเลขเหมือนกัน ถ้าพิจารณาแล้วไม่สามารถนำไปใช้งานได้ให้กรอกเครื่องหมาย X

4.1 แบบที่ 1 ☒ วิธีที่ 1 ☒ วิธีที่ 2 ☒ วิธีที่ 3

4.2 แบบที่ 2 ☒ วิธีที่ 1 ☒ วิธีที่ 2 ☒ วิธีที่ 3

4.3 แบบที่ 3 ☒ วิธีที่ 1 ☒ วิธีที่ 2 ☒ วิธีที่ 3

สัญญาณ ECG ที่ผ่านการลดขนาดข้อมูลลง 12 เท่า

5. สัญญาณ ECG ที่ได้จากการลดขนาดข้อมูลแบบใด(แบบที่ 1, 2 และ 3)ในแต่ละวิธี ที่สามารถรักษาสาระสำคัญของสัญญาณ ECG ต้นแบบที่ได้ดีที่สุด ให้กรอกหมายเลขเรียงลำดับจากหมายเลข 1 (ดีที่สุด)ไปจนถึงหมายเลข 3 (แย่ที่สุด) ถ้าพิจารณาแล้วสัญญาณมีคุณภาพใกล้เคียงกันให้กรอกหมายเลขเหมือนกัน ถ้าพิจารณาแล้วไม่สามารถนำไปใช้งานได้ให้กรอกเครื่องหมาย X

5.1 วิธีที่ 1 ☒ แบบที่ 1 ☒ แบบที่ 2 ☒ แบบที่ 3

5.2 วิธีที่ 2 ☒ แบบที่ 1 ☒ แบบที่ 2 ☒ แบบที่ 3

5.3 วิธีที่ 3 ☒ แบบที่ 1 ☒ แบบที่ 2 ☒ แบบที่ 3

6. สัญญาณ ECG ที่ได้จากการลดขนาดข้อมูลวิธีใด(วิธีที่ 1, 2 และ 3)ในแต่ละแบบ ที่สามารถรักษาสาระสำคัญของสัญญาณ ECG ต้นแบบที่ได้ดีที่สุด ให้กรอกหมายเลขเรียงลำดับจากหมายเลข 1 (ดีที่สุด)ไปจนถึงหมายเลข 3 (แย่ที่สุด) ถ้าพิจารณาแล้วสัญญาณมีคุณภาพใกล้เคียงกันให้กรอกหมายเลขเหมือนกัน ถ้าพิจารณาแล้วไม่สามารถนำไปใช้งานได้ให้กรอกเครื่องหมาย X

6.1 แบบที่ 1 ☒ วิธีที่ 1 ☒ วิธีที่ 2 ☒ วิธีที่ 3

6.2 แบบที่ 2 ☒ วิธีที่ 1 ☒ วิธีที่ 2 ☒ วิธีที่ 3

6.3 แบบที่ 3 ☒ วิธีที่ 1 ☒ วิธีที่ 2 ☒ วิธีที่ 3

สรุปจากเอกสารชุดที่ 1, 2 และ 3

7. ที่อัตราส่วนการลดขนาดข้อมูลสูงสุดเท่าใด วิธีใดและแบบใด ที่แพทย์ผู้วินิจฉัยมีความเห็นว่า ทำให้สัญญาณ ECG ที่ผ่านการลดขนาดข้อมูลมีความเพี้ยนอยู่ในระดับที่ทางการแพทย์ยอมรับว่า สามารถวินิจฉัยได้ผลเช่นเดียวสัญญาณ ECG ต้นแบบ

☒ 6 เท่า

วิธีที่ 1 แบบที่ 1-3

วิธีที่ 2 แบบที่ 1-3

วิธีที่ 3 แบบที่ 1-3

☐ 9 เท่า

วิธีที่ แบบที่

วิธีที่ แบบที่

วิธีที่ แบบที่

☐ 12 เท่า

วิธีที่ แบบที่

วิธีที่ แบบที่

วิธีที่ แบบที่

หมายเหตุ กรอกคำตอบได้มากกว่า 1 คำตอบ ตัวอย่างเช่น

☒ 6 เท่า

วิธีที่ 1 แบบที่ 2,3

วิธีที่ 3 แบบที่ 1,2

วิธีที่ แบบที่

สัญญาณ ECG ผิดปกติที่ผ่านการลดขนาดข้อมูลด้วยวิธีที่ 3 แบบที่ 2

8. ที่อัตราส่วนการลดขนาดข้อมูลสูงสุดเท่าใด ที่แพทย์ผู้วินิจฉัยมีความเห็นว่า ทำให้สัญญาณ ECG ผิดปกติที่ผ่านการลดขนาดข้อมูลมีความเพี้ยนอยู่ในระดับที่ทางการแพทย์ยอมรับว่า สามารถวินิจฉัยได้ผลเช่นเดียวสัญญาณ ECG ดั้งเดิม

8.1 สัญญาณ ECG แบบที่ 1

☒ 6 เท่า

☐ 9 เท่า

☐ 12 เท่า

8.2 สัญญาณ ECG แบบที่ 2

☒ 6 เท่า

☐ 9 เท่า

☐ 12 เท่า

ลงชื่อ.....

(อินต์ ใจบุญงาม)

นายแพทย์ผู้ออกความคิดเห็น

ประวัติผู้เขียน

นายมานิตย์ เกียรติกำจายขจร เกิดเมื่อวันที่ 2 กุมภาพันธ์ พ.ศ.2513 ที่จังหวัด กรุงเทพมหานคร สำเร็จการศึกษาปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมเครื่องกล จาก สถาบันเทคโนโลยีพระจอมเกล้าธนบุรี ปีการศึกษา 2535 ผลงานทางวิชาการที่ได้รับการตีพิมพ์ “การทดสอบการประมาณข้อมูลโดยใช้การแปลงเวฟเล็ตเต็มหน่วยเทียบกับการแปลงโคไซน์เต็มหน่วย” ในวารสารวิจัยและพัฒนา สถาบันเทคโนโลยีพระจอมเกล้าธนบุรี ปีที่ 20 ฉบับที่ 2 ธันวาคม พ.ศ.2540 เข้ารับราชการในตำแหน่งอาจารย์ลูกจ้างระดับ 3 ภาควิชาอิเล็กทรอนิกส์ คณะ วิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ตั้งแต่ปี พ.ศ.2539 จนถึงปัจจุบัน

