



รายงานการวิจัย

โครงการศึกษาและพัฒนาอัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียง
ที่ทำงานแบบเวลาจริง
(Real-time Audio Similarity Comparison Algorithm)

มหาวิทยาลัยเทคโนโลยีสุรนารี

ได้รับทุนอุดหนุนการวิจัยจาก
มหาวิทยาลัยเทคโนโลยีสุรนารี

ผลงานวิจัยเป็นความรับผิดชอบของหัวหน้าโครงการวิจัยแต่เพียงผู้เดียว



รายงานการวิจัย

โครงการศึกษาและพัฒนาอัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียง
ที่ทำงานแบบเวลาจริง
(Real-time Audio Similarity Comparison Algorithm)

คณะผู้วิจัย

หัวหน้าโครงการ

ผู้ช่วยศาสตราจารย์ ดร. รังสรรค์ ทองทา

สาขาวิชาวิศวกรรมโทรคมนาคม

สำนักวิชาวิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีสุรนารี

ได้รับทุนอุดหนุนการวิจัยจากมหาวิทยาลัยเทคโนโลยีสุรนารี ปีงบประมาณ พ.ศ. 2560

ผลงานวิจัยเป็นความรับผิดชอบของหัวหน้าโครงการวิจัยแต่เพียงผู้เดียว

มกราคม 2563

กิตติกรรมประกาศ

ผู้วิจัยขอขอบคุณ มหาวิทยาลัยเทคโนโลยีสุรนารีในการสนับสนุนงบประมาณวิจัย ปีงบประมาณ 2560 ที่เอื้อเพื่อสถานที่และเครื่องมือต่าง ๆ ในการทำวิจัย บัดนี้งานวิจัยได้สำเร็จลุล่วงแล้ว คณะผู้วิจัยขอขอบพระคุณมา ณ โอกาสนี้

ผู้วิจัย

มกราคม 2563



บทคัดย่อ

ในปัจจุบันมีจัดตั้งสถานีวิทยุกระจายเสียงในระบบเอฟเอ็มขึ้นเป็นจำนวนมาก ส่งผลให้เกิดการรบกวนในย่านความถี่ใกล้เคียง โดยเฉพาะอย่างการรบกวนการสื่อสารระหว่างเครื่องบินและสถานีควบคุมภาคพื้น ซึ่งอาจก่อให้เกิดความเสียหายต่อประเทศได้ ในการค้นหาแหล่งกำเนิดของสัญญาณรบกวนสามารถทำได้ด้วยการเปรียบเทียบความคล้ายกันของเสียงที่ได้จากสัญญาณรบกวนและเสียงจากสถานีวิทยุที่กำลังออกอากาศ ซึ่งใช้มนุษย์ทำการเปรียบเทียบจึงทำให้การค้นหาแหล่งกำเนิดของสัญญาณรบกวนเป็นไปอย่างล่าช้า ดังนั้นจำเป็นต้องพัฒนาอัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียงเพื่อให้การค้นหาแหล่งกำเนิดของสัญญาณรบกวนมีประสิทธิภาพมากขึ้น

งานวิจัยนี้นำเสนออัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียงที่ทำงานแบบเวลาจริงที่ใช้หลักการทางคณิตศาสตร์อย่างง่ายในการทำงาน โดยจะแบ่งเสียงที่จะเปรียบเทียบออกเป็น 3 ย่านความถี่ จากนั้นทำการคำนวณหาผลต่างของทั้งสองเสียง และนำค่าผลต่างของเสียงที่ได้มาพิจารณาเทียบกับค่าเทรสโฮลด์ของแต่ละย่านความถี่ สุดท้ายทำการรวมผลการพิจารณาจากทั้ง 3 ย่านความถี่มาทำการโหวต 2 ใน 3 เสียง เพื่อตัดสินว่าเสียงที่ทำการเปรียบเทียบนั้นมีความคล้ายกันของเสียงหรือไม่

นอกจากนี้ยังนำเสนอชุดอุปกรณ์ต้นแบบที่นำอัลกอริทึมที่พัฒนามาใช้งาน เพื่อใช้ในการค้นหาสถานีวิทยุที่เป็นต้นกำเนิดของสัญญาณรบกวนแบบอัตโนมัติ จากผลการทดลองเมื่อนำชุดอุปกรณ์ต้นแบบมาใช้งานในสภาพแวดล้อมจริง จะมีความถูกต้องในการตัดสินผลการเปรียบเทียบ 71.33% และมีความผิดพลาดในการตัดสินผลการเปรียบเทียบ 14.00% และการเปรียบเทียบความคล้ายกันของเสียงในแต่ละครั้งใช้เวลา 51 วินาทีโดยประมาณ

Abstract

Currently, there are many FM broadcasting stations which cause interference with the nearby frequency range. Especially, the interference causes communication problems between the aircraft and the ground control station. To identify sources of the interference, manually scanning a receiver to compare similarity with the sound from the interference is the only method available. So, audio similarity comparison algorithm may benefit to identify the sources of the interference.

This research proposes a real-time audio similarity comparison algorithm using simple mathematical calculations. The target audios are divided into 3 bands. Then calculate subtraction of the two sounds. The subtraction results compared with the threshold of each band. To get a final result, at least 2 similar results from 3 bands will be considered similarity.

In addition, the prototype uses the algorithm developed. It can be used in automatic searching for sources of the interference. The experimental results show 71.33% of the right comparison and 14.00% of the false comparison. The comparison each time takes 51 seconds.

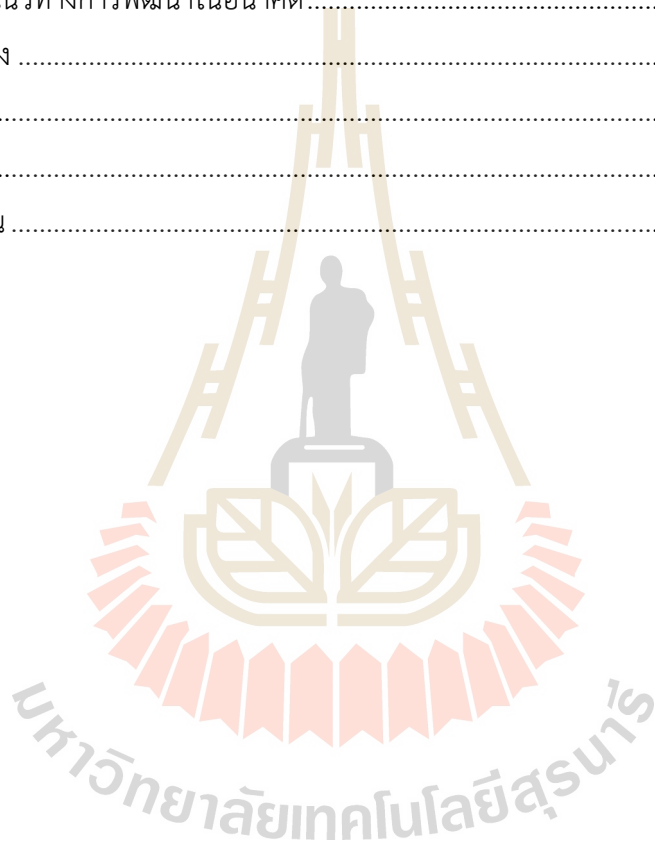
สารบัญ

หน้า

กิตติกรรมประกาศ	ก
บทคัดย่อ (ภาษาไทย)	ข
บทคัดย่อ (ภาษาอังกฤษ)	ค
สารบัญ	ง
สารบัญตาราง	ช
สารบัญรูป	ซ
บทที่	
บทที่ 1	
บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของการวิจัย	2
1.3 สมมติฐานของการวิจัย	2
1.4 ข้อตกลงเบื้องต้น	2
1.5 ขอบเขตของการวิจัย	2
1.6 วิธีดำเนินการวิจัย	3
1.7 ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่ 2	
ทฤษฎีและบทความที่เกี่ยวข้อง	4
2.1 บทนำ	4
2.2 ทฤษฎีที่เกี่ยวข้อง	4
2.2.1 อินเทอร์มอดูเลชัน (Intermodulation)	4
2.2.2 Reverse Intermodulation.....	8
2.2.3 เสียง (Audio).....	9
2.3 ปรัชญาวรรณกรรมที่เกี่ยวข้อง	11
2.4 อุปกรณ์ที่ใช้ในงานวิจัย	12
2.4.1 บอร์ด Raspberry Pi.....	12
2.4.2 USB Audio Capture	13
2.4.3 ส่วนการแสดงผล	14
2.5 สรุป	15

บทที่ 3	อัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียง ที่ทำงานแบบเวลาจริง	16
3.1	บทนำ	16
3.2	อัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียงที่ทำงานแบบเวลาจริง	16
3.3	หลักการเปรียบเทียบความคล้ายกันของเสียง	18
3.3.1	กระบวนการเตรียมข้อมูล (Data preparing)	19
3.3.2	กระบวนการคำนวณผลต่างของเสียง (Audio subtraction)	27
3.3.3	กระบวนการตัดสินใจความคล้าย (Similarly decision)	27
3.4	การใช้งานอัลกอริทึมการเปรียบเทียบความคล้ายกันของเสียงที่ทำงานแบบเวลาจริงใน การค้นหาคำที่เป็นต้นกำเนิดของสัญญาณรบกวน	28
3.4.1	หลักการทำงาน.....	29
3.4.2	ชุดอุปกรณ์ต้นแบบ	29
3.5	สรุป	31
บทที่ 4	การทดลองและวิเคราะห์ของ การทำงานบนโปรแกรมคอมพิวเตอร์.....	32
4.1	บทนำ	32
4.2	การจำลองการทำงานบนโปรแกรมคอมพิวเตอร์.....	32
4.2.1	ขั้นตอนการทำงานและพารามิเตอร์ที่ใช้งาน	32
4.2.2	เตรียมตัวอย่างการทดลอง	33
4.3	การทดลองเพื่อหาค่าเทรสโวลต์ที่เหมาะสม	33
4.3.1	ผลการทดลอง.....	34
4.4	การทดสอบประสิทธิภาพการทำงานของอัลกอริทึม	37
4.4.1	การตัดสินใจความคล้ายกันของเสียง	37
4.4.2	ความทนต่อสัญญาณรบกวน.....	38
4.4.3	ความทนต่อการคลาดเคลื่อนทางเวลา	39
4.4.4	เวลาที่ใช้ในการเปรียบเทียบ	40
4.5	สรุป	40
บทที่ 5	การทดลองและวิเคราะห์ของ การทำงานบนชุดอุปกรณ์ต้นแบบ	42
5.1	บทนำ	42
5.2	การทดสอบการทำงานของอัลกอริทึมบนชุดอุปกรณ์ต้นแบบ	42
5.2.1	ขั้นตอนการทำงาน.....	42
5.2.2	เตรียมอุปกรณ์ในการทดลอง	43

5.3	การทดสอบประสิทธิภาพการทำงานของอัลกอริทึม	46
5.3.1	การตัดสินความคล้ายกันของเสียง	46
5.3.2	เวลาที่ใช้ในการเปรียบเทียบ	47
5.4	สรุป	47
บทที่ 6	สรุปงานวิจัยและข้อเสนอแนะ	48
6.1	สรุป	48
6.2	ปัญหาและข้อเสนอแนะ	49
6.3	แนวทางการพัฒนาในอนาคต	49
	รายการอ้างอิง	50
	ภาคผนวก ก	52
	ภาคผนวก ข	66
	ประวัติผู้เขียน	72



สารบัญตาราง

ตารางที่	หน้า
2.1	ความถี่ที่เกิดจากอินเตอร์มอดูเลชัน..... 7
3.1	รายละเอียดของแต่ละย่านความถี่ในชั้นตอนกรองสัญญาณ..... 21
4.1	ตารางผลการใช้ค่าเทรสโพลต์ต่าง ๆ ในการตัดสินความคล้ายกันของเสียง..... 36
4.2	ตารางผลการโหวตเสียงข้างมากในการตัดสินความคล้ายกันของเสียง..... 37
4.3	ตารางผลการทดลองเพื่อหาความทนต่อสัญญาณรบกวน 39
5.1	ตารางผลการทดลองการทำงานของอัลกอริทึมจากการทดลองกลุ่มละ 300 ครั้ง 46

สารบัญรูป

รูปที่	หน้า
2.1	แบบจำลองของระบบที่ไม่เป็นเชิงเส้น (Nonlinear system)..... 5
2.2	แบบจำลองการเกิดอินเตอร์มอดูเลชัน (Intermodulation)..... 7
2.3	การจำลองสถานการณ์การเกิด Reverse Intermodulation..... 8
2.4	แผนภาพของสเปกตรัมของเสียงที่ได้จากการผสมกันโดยตรง (Mix-audio)..... 9
2.5	แผนภาพของสเปกตรัมของเสียงที่ได้จาก RIM..... 9
2.6	ลักษณะการส่งผ่านคลื่น 10
2.7	ขั้นตอนการทำงานของระบบลายนิ้วมือเสียง 12
2.8	บอร์ด Raspberry Pi 13
2.9	USB Audio Capture..... 14
2.10	ส่วนการแสดงผล 14
3.1	สัญญาณเสียงในทางเวลาของสัญญาณเสียงที่มีความคล้ายกัน..... 17
3.2	สัญญาณเสียงในทางเวลาของสัญญาณเสียงที่ไม่มีความคล้ายกัน 17
3.3	ขั้นตอนของการค้นหาสถานที่ที่เป็นต้นกำเนิดของสัญญาณรบกวน 18
3.4	หลักการการทำงานของอัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียง ที่ทำงานแบบเวลาจริง.. 19
3.5	สัญญาณเสียงในขั้นตอนซึ่สัญญาณ 20
3.6	ข้อมูลสัญญาณเสียงในขั้นตอนนอร์มัลไลเซชัน 21
3.7	ผลตอบสนองทางความถี่ (Frequency respond) ของช่วงความถี่ต่ำ..... 22
3.8	ผลของข้อมูลสัญญาณเสียงที่ผ่านตัวกรองสัญญาณช่วงความถี่ต่ำ 23
3.9	ผลตอบสนองทางความถี่ (Frequency respond) ของช่วงความถี่กลาง..... 23
3.10	ผลของข้อมูลสัญญาณเสียงที่ผ่านตัวกรองสัญญาณช่วงความถี่กลาง 24
3.11	ผลตอบสนองทางความถี่ (Frequency respond) ของช่วงความถี่สูง 24
3.12	ผลของข้อมูลสัญญาณเสียงที่ผ่านตัวกรองสัญญาณช่วงความถี่สูง 25
3.13	ข้อมูลสัญญาณเสียงในขั้นตอนเฉลี่ยข้อมูล 26
3.14	ขั้นตอนและอุปกรณ์ในการค้นหาสถานที่ที่เป็นต้นกำเนิดของสัญญาณรบกวน 28
3.15	แผนผังการทำงานของระบบการค้นหาคลื่นความถี่ที่เป็นต้นกำเนิดของ สัญญาณรบกวนแบบอัตโนมัติ 30
3.16	ชุดอุปกรณ์ต้นแบบ..... 30

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.1 กราฟความถี่ของค่าความคล้อยกันของเสียง (I_x) ที่เกิดขึ้นในย่านความถี่ต่ำ	34
4.2 กราฟความถี่ของค่าความคล้อยกันของเสียง (I_x) ที่เกิดขึ้นในย่านความถี่กลาง	35
4.3 กราฟความถี่ของค่าความคล้อยกันของเสียง (I_x) ที่เกิดขึ้นในย่านความถี่สูง.....	35
4.4 การทดลองความทนต่อสัญญาณรบกวน.....	38
4.5 กราฟความถูกต้องในการเปรียบเทียบของแต่ละระยะเวลาหนึ่ง	40
5.1 อุปกรณ์ภาคส่งสัญญาณวิทยุในการจำลองการออกอากาศ	43
5.2 อุปกรณ์จริงภาคส่งสัญญาณวิทยุในการจำลองการออกอากาศ	44
5.3 คลื่นความถี่ที่ได้จากการจำลองการออกอากาศและ RIM.....	44
5.4 อุปกรณ์ภาครับสัญญาณวิทยุในการจำลองการออกอากาศ.....	45
5.5 อุปกรณ์จริงภาครับสัญญาณวิทยุในการจำลองการออกอากาศ.....	45

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันสำนักงานคณะกรรมการกิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคมแห่งชาติ (กสทช.) ได้มีการอนุญาตให้ผู้ทดลองประกอบกิจการวิทยุกระจายเสียงในย่านความถี่ 88 – 108 MHz ทำให้มีการจัดตั้งสถานีวิทยุกระจายเสียงขึ้นเป็นจำนวนมาก ส่งผลให้เกิดการรบกวนในย่านความถี่อื่น ๆ โดยเฉพาะอย่างยิ่งการรบกวนการสื่อสารระหว่างเครื่องบินกับสถานีควบคุมภาคพื้นดินในย่านความถี่ 108 – 137 MHz ทำให้มีความเสี่ยงที่จะเกิดความเสียหายต่อชีวิต ทรัพย์สิน และส่งผลเสียต่อประเทศเป็นอย่างมาก

ทางสำนักงาน กสทช. และหน่วยงานอื่น ๆ ที่เกี่ยวข้องได้ทำการค้นหาแหล่งกำเนิดคลื่นสัญญาณรบกวนเพื่อแก้ไขปัญหที่เกิดขึ้นอย่างต่อเนื่อง แต่ปัญหาของกระบวนการค้นหาแหล่งกำเนิดคลื่นสัญญาณรบกวนในปัจจุบันคือ ไม่มีเครื่องมือวัดที่เหมาะสมจึงต้องอาศัยทรัพยากรบุคคลที่มีความชำนาญ เพื่อทำการเปรียบเทียบความคล้ายกันของเสียงระหว่างสัญญาณเสียงจากสถานีวิทยุที่กำลังออกอากาศกับสัญญาณเสียงจากคลื่นความถี่รบกวนที่เกิดขึ้น จึงทำให้กระบวนการค้นหาแหล่งกำเนิดคลื่นสัญญาณรบกวนในปัจจุบันใช้ระยะเวลาในการค้นหานานและสิ้นเปลืองทรัพยากรบุคคลเป็นจำนวนมาก ดังนั้นจึงมีความจำเป็นที่จะต้องพัฒนาอัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียง และสามารถนำอัลกอริทึมที่พัฒนามาใช้ในการค้นหาสถานีวิทยุที่เป็นต้นกำเนิดของสัญญาณรบกวนได้ จากการสำรวจปริทัศน์วรรณกรรมพบว่า มีแนวคิดของการนำระบบลายนิ้วมือเสียง (Audio Fingerprinting System) มาใช้ในการเปรียบเทียบความคล้ายกันของเสียง โดยการสกัดลายนิ้วมือเสียง (audio fingerprinting extract) ที่มีลักษณะเด่นทางความถี่ที่แตกต่างกันในแต่ละสัญญาณเสียง ไปจับคู่กับตัวแทนลายนิ้วมือเสียงในฐานข้อมูล เพื่อค้นหาลายนิ้วมือเสียงที่อาจมีความคล้ายกันเกิดขึ้น (ตรงกับฐานข้อมูล) ซึ่งในกระบวนการสกัดลายนิ้วมือเสียงนั้นจะใช้ฟังก์ชัน Fast Fourier Transform (FFT) เพื่อหาลักษณะเด่นทางความถี่ของสัญญาณเสียง จึงทำให้ระบบลายนิ้วมือเสียงต้องการทรัพยากรในการคำนวณทางคณิตศาสตร์เป็นจำนวนมาก และจำเป็นต้องมีการจัดเตรียมฐานข้อมูลที่ใช้ในการเปรียบเทียบความคล้ายกันของเสียง ดังนั้นระบบลายนิ้วมือเสียงนี้ไม่เหมาะสำหรับใช้ในการเปรียบเทียบความคล้ายกันของเสียงที่ทำงานแบบเวลาจริง

งานวิจัยนี้นำเสนออัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียงที่ทำงานแบบเวลาจริง โดยใช้การคำนวณทางคณิตศาสตร์อย่างง่ายจึงทำให้มีการใช้ทรัพยากรในการคำนวณทางคณิตศาสตร์น้อยสามารถทำงานได้ในอุปกรณ์ขนาดเล็กได้ และนำเสนอชุดอุปกรณ์ต้นแบบที่นำอัลกอริทึมที่พัฒนามาใช้ เพื่อใช้ในการค้นหาสถานีที่เป็นต้นกำเนิดของสัญญาณรบกวนแบบอัตโนมัติ ซึ่งจะมีส่วนช่วยในการแก้ไขปัญหา

การรบกวนการสื่อสารระหว่างเครื่องบินกับสถานีควบคุมภาคพื้นดินได้เร็วยิ่งขึ้น เพื่อช่วยลดความเสี่ยงที่จะเกิดความเสียหายต่อชีวิตและทรัพย์สินที่อาจเกิดขึ้นได้ และช่วยเพิ่มการใช้ทรัพยากรบุคคลได้อย่างมีประสิทธิภาพ

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 เพื่อศึกษาและพัฒนาอัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียงที่ทำงานแบบเวลาจริง

1.2.2 สามารถนำอัลกอริทึมที่พัฒนาประยุกต์ใช้งานในการค้นหาสถานีที่เป็นต้นกำเนิดของสัญญาณรบกวนแบบอัตโนมัติได้

1.3 สมมติฐานของการวิจัย

1.3.1 อัลกอริทึมสามารถเปรียบเทียบสัญญาณเสียงจาก 2 แหล่ง เพื่อหาความคล้ายกันของเสียงและทำงานแบบเวลาจริงได้

1.3.2 อัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียงที่ทำงานแบบเวลาจริงสามารถเพิ่มประสิทธิภาพของกระบวนการค้นหาสถานีที่เป็นต้นกำเนิดของสัญญาณรบกวนได้

1.4 ข้อตกลงเบื้องต้น

1.4.1 อัลกอริทึมเปรียบเทียบความคล้ายกันของเสียงจากสัญญาณเสียง 2 แหล่ง

1.4.2 ใช้โปรแกรม MATLAB ในการจำลองการทำงานอัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียงที่ทำงานแบบเวลาจริง

1.4.3 ทดสอบการทำงานของอัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียงที่ทำงานแบบเวลาจริงบนชุดอุปกรณ์จริง

1.5 ขอบเขตของการวิจัย

1.5.1 ศึกษากระบวนการเปรียบเทียบความคล้ายกันของเสียงจากสัญญาณเสียง 2 แหล่ง

1.5.2 พัฒนาอัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียงที่ทำงานแบบเวลาจริง

1.5.3 ทดสอบการทำงานของอัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียงที่ทำงานแบบเวลาจริง

1.6 วิธีดำเนินการวิจัย

1.6.1 แนวทางการดำเนินงาน

- 1) สํารวจปรีทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้องกับงานวิจัย
- 2) ศึกษาการเปรียบเทียบความคล้ายกันของเสียง
- 3) พัฒนาอัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียงที่ทำงานแบบเวลาจริงโดย

ใช้โปรแกรม MATLAB

4) จำลองการทำงานของอัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียงบนโปรแกรม MATLAB

5) ทดสอบการทำงานของอัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียงที่ทำงานแบบเวลาจริงบนชุดอุปกรณ์จริง

- 6) วิเคราะห์ประสิทธิภาพของอัลกอริทึมที่ได้พัฒนาขึ้นจากผลการทดลอง

1.6.2 สถานที่ทำการวิจัย

ห้องเตรียมการปฏิบัติการ อาคารเครื่องมือ 11 มหาวิทยาลัยเทคโนโลยีสุรนารี 111 ถนนมหาวิทยาลัย ต. สุรนารี อ. เมือง จ. นครราชสีมา 30000

1.6.3 เครื่องมือที่ใช้ในการวิจัย

- 1) เครื่องคอมพิวเตอร์ส่วนบุคคล
- 2) โปรแกรม MATLAB

1.6.4 การวิเคราะห์ข้อมูล

วิเคราะห์ข้อมูลที่ได้จากผลทดลองการทำงานของอัลกอริทึมที่ได้พัฒนาขึ้นในสภาพแวดล้อมที่ต่างกัน เพื่อหาประสิทธิภาพการทำงานของอัลกอริทึม และสรุปผลออกมาในรูปของรูปภาพและข้อมูลตารางตัวเลข

1.7 ประโยชน์ที่คาดว่าจะได้รับ

สามารถนำอัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียงที่ทำงานแบบเวลาจริง มาใช้งานในการค้นหาคลื่นสถานีที่เป็นต้นกำเนิดของสัญญาณรบกวนแบบอัตโนมัติ ซึ่งจะมีส่วนช่วยในการแก้ไขปัญหาการรบกวนการสื่อสารระหว่างเครื่องบินกับสถานีควบคุมภาคพื้นดินให้เป็นไปอย่างรวดเร็ว ลดการใช้ทรัพยากรบุคคลอย่างสิ้นเปลือง และลดความเสี่ยงที่จะเกิดความเสียหายต่อชีวิตและทรัพย์สินที่อาจเกิดขึ้นได้

บทที่ 2

ทฤษฎีและบทความที่เกี่ยวข้อง

2.1 บทนำ

ในบทนี้จะกล่าวถึงทฤษฎีที่เกี่ยวข้องกับการเปรียบเทียบความคล้ายกันของเสียงและสัญญาณรบกวนที่รบกวนการสื่อสารระหว่างเครื่องบินกับสถานีควบคุมภาคพื้น และประสิทธิภาพของระบบที่เกี่ยวข้องกับการเปรียบเทียบความคล้ายกันของเสียงที่มีอยู่ในปัจจุบัน เพื่อเป็นแนวทางของการพัฒนาอัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียงที่ทำงานแบบเวลาจริง รวมไปถึงรายละเอียดของอุปกรณ์ที่จะนำมาใช้เป็นชุดอุปกรณ์ต้นแบบในงานวิจัยนี้อีกด้วย

2.2 ทฤษฎีที่เกี่ยวข้อง

2.2.1 อินเตอร์มอดูเลชัน (Intermodulation)

อินเตอร์มอดูเลชัน (Intermodulation: IM หรือ Intermodulation distortion: IMD) เกิดจากความไม่เป็นเชิงเส้น (Nonlinear) ของระบบ ซึ่งความไม่เป็นเชิงเส้นนี้จะอยู่ในทุกอุปกรณ์เสมอ เพียงแค่เราจะมีการใช้งานอุปกรณ์นั้นๆ ในย่านที่มันไม่เป็นเชิงเส้นหรือไม่ ในอุปกรณ์บาง ประเภทจะมีความไม่เป็นเชิงเส้นมาก เช่น วงจรขยายสัญญาณ class C ที่มีใช้กันอย่างแพร่หลายใน วงจรขยายกำลังความถี่วิทยุระบบเอฟเอ็ม ซึ่งไม่ว่าเราจะใช้งานมันอย่างไร เราก็จะเจอความไม่เป็นเชิงเส้นของมันเสมอ ในขณะที่อุปกรณ์บางประเภทมีความเป็นเชิงเส้นสูง เช่น อุปกรณ์เครื่องมีอวด ซึ่ง เราจะเจอความไม่เป็นเชิงเส้นของมันก็ต่อเมื่อเราป้อนสัญญาณแรงกว่าที่กำหนดไว้ อย่างไรก็ตามการเกิดอินเตอร์มอดูเลชันจะเกิดจากอุปกรณ์ไม่สามารถเกิดขึ้นในอากาศว่างได้ หลักการอินเตอร์มอดูเลชันสามารถนำไปใช้ประโยชน์มากมาย เช่น ในวงจรผสมสัญญาณ (Mixer) ที่เราต้องการให้ได้ความถี่ผลต่างเพื่อป้อนให้กับภาคขยายไอเอฟ (IF amplifiers) ของเครื่องรับวิทยุแบบ Super heterodyne แต่ขณะเดียวกันมันก็สร้างปัญหามากมายในการแพร่คลื่นเช่น การกระจายเสียงในระบบเอฟเอ็ม (FM broadcast) เนื่องจากผลต่างของความถี่ที่ได้จะล่าช้าไปในช่วงสื่อสารวิทยุการบิน ซึ่งเป็นการเกี่ยวข้องกับชีวิตและทรัพย์สินของผู้คนจำนวนมาก

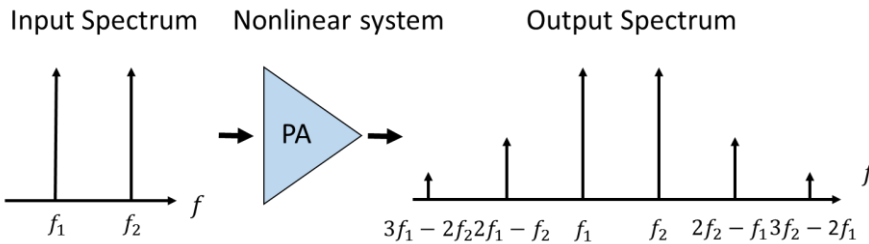
อินเตอร์มอดูเลชันเป็นกระบวนการเปลี่ยนแปลงทางความถี่ เมื่อมีสัญญาณอินพุต (input) ที่มีองค์ประกอบทางความถี่มากกว่า 1 ความถี่ ผ่านเข้าไปยังระบบที่มีความไม่เป็นเชิงเส้น โดยที่สัญญาณเอาต์พุต (output) ที่ได้นอกจากจะมีองค์ประกอบทางความถี่ดั้งเดิมแล้ว ยังมีองค์ประกอบทางความถี่ใหม่เพิ่มเติมเข้าไปด้วย

เพื่อทำความเข้าใจถึงความถี่และลักษณะของสัญญาณที่เกิดจากอินเตอร์มอดูเลชัน จึงจำลองการป้อนสัญญาณอินพุตที่มีองค์ประกอบของ 2 ความถี่เข้าไปยังวงจขยายสัญญาณความถี่วิทยุเอฟเอ็มที่มีความไม่เป็นเชิงเส้นสูง โดยที่พิจารณาระบบที่ไม่เป็นเชิงเส้นดังสมการที่ 2.1

$$\begin{aligned}
 y(t) = & \alpha_1 [A \cos(2\pi f_1 t) + B \cos(2\pi f_2 t)] + \\
 & \alpha_2 \left[\left(\frac{A^2}{2} + \frac{A^2}{2} \cos(2\pi 2f_1 t) \right) + \left(\frac{B^2}{2} + \frac{B^2}{2} \cos(2\pi 2f_2 t) \right) + \right. \\
 & \quad \left. + AB \cos(2\pi(f_1 + f_2)t) + AB \cos(2\pi(f_1 - f_2)t) \right] + \\
 & \alpha_3 \left[\left(\frac{3A^3}{4} \cos(2\pi f_1 t) + \frac{A^3}{4} \cos(2\pi 3f_1 t) \right) + \right. \\
 & \quad \left(\frac{3A^2 B}{2} \cos(2\pi f_2 t) + \frac{3A^2 B}{4} \cos(2\pi(2f_1 + f_2)t) + \frac{3A^2 B}{4} \cos(2\pi(2f_1 - f_2)t) \right) + \\
 & \quad \left(\frac{3AB^2}{2} \cos(2\pi f_1 t) + \frac{3AB^2}{4} \cos(2\pi(2f_2 + f_1)t) + \frac{3AB^2}{4} \cos(2\pi(2f_2 - f_1)t) \right) + \\
 & \quad \left. \left(\frac{3B^3}{4} \cos(2\pi f_2 t) + \frac{B^3}{4} \cos(2\pi 3f_2 t) \right) + \dots \right]
 \end{aligned}$$

ดังนั้น

$$\begin{aligned}
 y(t) = & \left[\frac{\alpha_2 A^2}{2} + \frac{\alpha_2 B^2}{2} \right] + \tag{2.2} \\
 & \left[\alpha_1 A + \frac{3\alpha_3 A^3}{4} + \frac{3\alpha_3 AB^2}{2} \right] \cos(2\pi f_1 t) + \\
 & \left[\alpha_1 B + \frac{3\alpha_3 B^3}{4} + \frac{3\alpha_3 A^2 B}{2} \right] \cos(2\pi f_2 t) + \\
 & \left[\frac{\alpha_2 A^2}{2} \right] \cos(2\pi(2f_1)t) + \left[\frac{\alpha_2 B^2}{2} \right] \cos(2\pi(2f_2)t) + \\
 & [\alpha_2 AB] \cos(2\pi(f_1 + f_2)t) + [\alpha_2 AB] \cos(2\pi(f_1 - f_2)t) + \\
 & \left[\frac{\alpha_3 A^3}{4} \right] \cos(2\pi(3f_1)t) + \left[\frac{\alpha_3 B^3}{4} \right] \cos(2\pi(3f_2)t) + \\
 & \left[\frac{3\alpha_3 A^2 B}{4} \right] \cos(2\pi(2f_1 + f_2)t) + \\
 & \left[\frac{3\alpha_3 A^2 B}{4} \right] \cos(2\pi(2f_1 - f_2)t) + \\
 & \left[\frac{3\alpha_3 AB^2}{4} \right] \cos(2\pi(2f_2 + f_1)t) + \\
 & \left[\frac{3\alpha_3 AB^2}{4} \right] \cos(2\pi(2f_2 - f_1)t) + \dots
 \end{aligned}$$



รูปที่ 2.2 แบบจำลองการเกิดอินเตอร์มอดูเลชัน (Intermodulation)

จากสมการ 2.2 ทำให้ทราบว่าในสัญญาณเอาต์พุตที่ได้จากระบบที่มีความไม่เป็นเชิงเส้นจะมีองค์ประกอบทางความถี่เพิ่มเติมนอกจาก 2 ความถี่ตั้งต้น ซึ่งองค์ประกอบที่เพิ่มเข้ามาคือ ไฟกระแสดตรง (direct current : DC) ส่วนฮาร์โมนิก (harmonic) ของทั้งสองความถี่ตั้งต้นที่ $2f_1$ $2f_2$ $3f_1$ $3f_2$ เป็นต้น และส่วนอินเตอร์มอดูเลชัน ซึ่งมีความถี่ผลลัพธ์ที่เป็นความถี่ผลบวกและผลต่างของสองความถี่ตั้งต้น รวมทั้งความถี่ผลบวกและผลต่างของความถี่ที่เป็นฮาร์โมนิกของสองความถี่ตั้งต้น โดยที่ความถี่ผลลัพธ์สามารถคำนวณได้ดังตารางที่ 2.1 นอกจากนี้ยังทราบว่าอินเตอร์มอดูเลชันที่เกิดขึ้นจะมีองค์ประกอบของสัญญาณตั้งต้น (แอมพลิจูด A และ B) รวมกันอยู่

ตารางที่ 2.1 ความถี่ที่เกิดจากอินเตอร์มอดูเลชัน

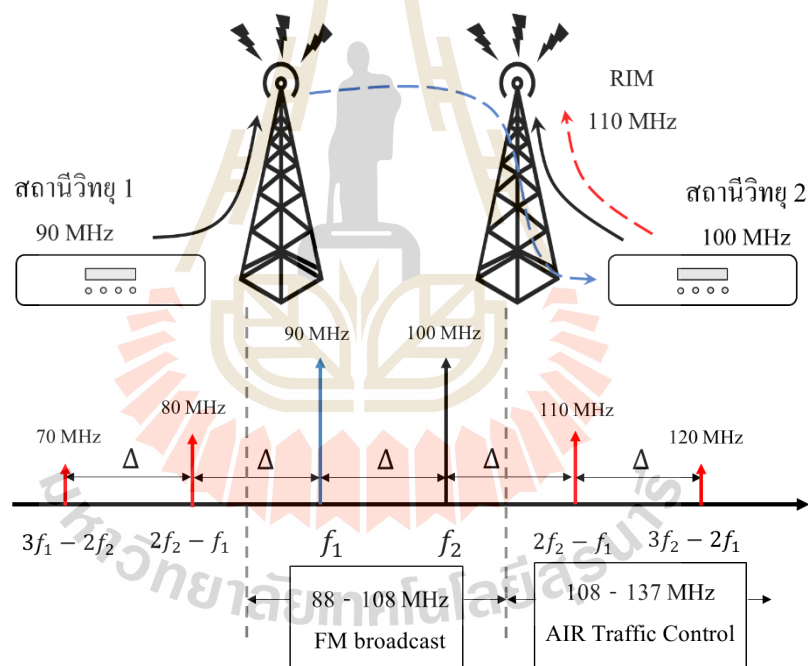
Order	สมการ			
2	$f_1 + f_2$	$f_1 - f_2$		
3	$2f_2 + f_1$	$2f_2 - f_1$	$2f_1 + f_2$	$2f_1 - f_2$
4	$2f_2 + 2f_1$	$2f_2 - 2f_1$	$3f_2 + f_1$	$3f_2 - f_1$
	$3f_1 + f_2$	$3f_1 - f_2$		
5	$3f_2 + 2f_1$	$3f_2 - 2f_1$	$3f_1 + 2f_2$	$3f_1 - 2f_2$
	$4f_2 + f_1$	$4f_2 - f_1$	$4f_1 + f_2$	$4f_1 - f_2$
6	$3f_2 + 3f_1$	$3f_2 - 3f_1$	$4f_2 + 2f_1$	$4f_2 - 2f_1$
	$4f_1 + 2f_2$	$4f_1 - 2f_2$	$5f_2 + f_1$	$5f_2 - f_1$
	$5f_1 + f_2$	$5f_1 - f_2$		
7	$4f_2 + 3f_1$	$4f_2 - 3f_1$	$4f_1 + 3f_2$	$4f_1 - 3f_2$
	$5f_2 + 2f_1$	$5f_2 - 2f_1$	$5f_1 + 2f_2$	$5f_1 - 2f_2$
	$6f_2 + f_1$	$6f_2 - f_1$	$6f_1 + f_2$	$6f_1 - f_2$

หมายเหตุ

1. Order ของอินเตอร์มอดูเลชันสามารถหาได้จากผลบวกของสัมประสิทธิ์ (ไม่นับเครื่องหมาย) ของแต่ละความถี่
2. ในที่นี้แสดงให้เห็นถึง Order 7 เท่านั้น แต่ในความจริงยังมี Order ที่สูงกว่านี้เรื่อย ๆ

2.2.2 Reverse Intermodulation

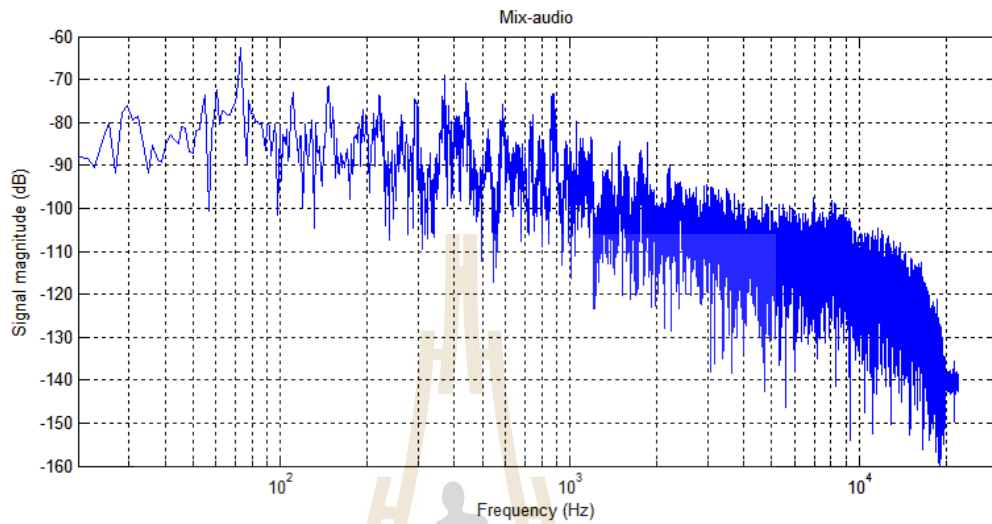
อินเทอร์มอดูเลชันที่เกิดขึ้นในกระจายเสียงระบบเอฟเอ็มนั้น เกิดขึ้นจากการที่เครื่องส่งเครื่องใดเครื่องหนึ่งหรือทั้งสองเครื่อง ไปรับเอาสัญญาณที่ส่งออกมาจากอีกเครื่อง ย้อนกลับเข้าไปในเครื่องส่ง แล้วสัญญาณดังกล่าวย้อนกลับไปจนถึงภาคขยายกำลังความถี่วิทยุ (RF power amplifier) เนื่องจากภาคขยายกำลังความถี่วิทยุมักจะไม่เป็นเชิงเส้นอยู่แล้ว ดังนั้นมันจึงขยายสัญญาณทั้งสอง (สัญญาณหลักที่ต้องการจะส่งและสัญญาณย้อนเข้ามา) พร้อมกับสร้างความถี่อินเทอร์มอดูเลชันแถมมาให้ ซึ่งเราเรียกปรากฏการณ์เช่นนี้ว่า Reverse Intermodulation : RIM ซึ่งมีความถี่อินเทอร์มอดูเลชันบางส่วนได้ไปรบกวนย่านความถี่ใกล้เคียง ยกตัวอย่างเช่น ดังรูปที่ 2.3 มีสถานีวิทยุที่ออกอากาศที่ความถี่ 90 MHz และ 100 MHz ได้ทำการออกอากาศในบริเวณใกล้เคียงกันทำให้เกิด RIM ที่ความถี่ 110 MHz และ 120 MHz ไปรบกวนการสื่อสารวิทยุการบินที่มีย่านความถี่อยู่ในช่วง 108–137 MHz ซึ่งอาจก่อให้เกิดความเสียหายขึ้นได้



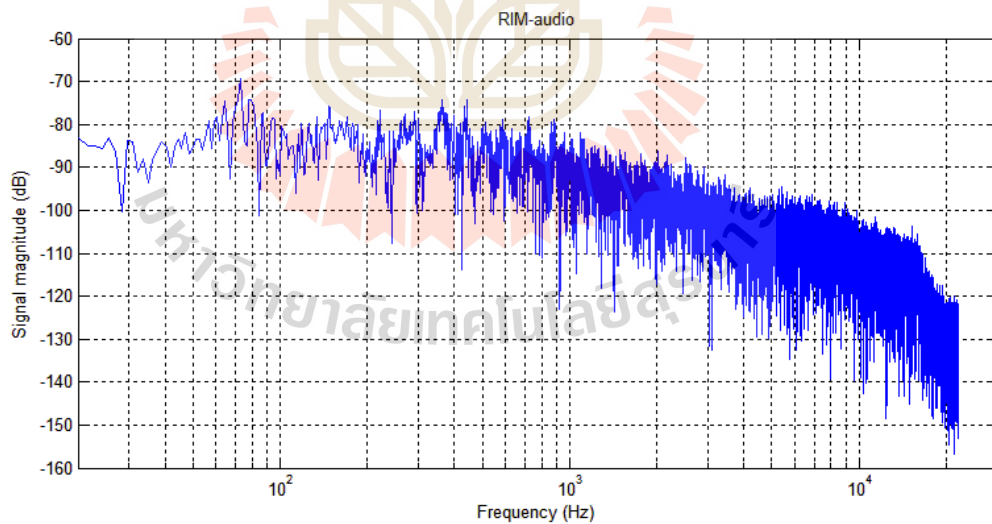
รูปที่ 2.3 การจำลองสถานการณ์การเกิด Reverse Intermodulation

เมื่อนำเครื่องรับวิทยุรับสัญญาณจาก RIM เสียงที่ได้จะมีข้อมูลของทั้งสองสถานีที่เป็นต้นกำเนิดผสมกันอยู่ แต่เนื่องจากสัญญาณ RIM ที่เกิดขึ้นนั้นมีกำลังต่ำกว่าสัญญาณจากสถานีวิทยุกระจายเสียงที่กำลังออกอากาศเป็นอย่างมาก ทำให้เสียงที่ได้นั้นมีสัญญาณรบกวน (noise) เป็นอย่างมาก แต่ยังคงมีองค์ประกอบของข้อมูลในแต่ละช่วงความถี่ ดังรูปที่ 2.4 และ 2.5 เป็นการเปรียบเทียบความแรงขององค์ประกอบทางความถี่ของเสียงที่ได้จากการผสมกันโดยตรง (Mix-audio) กับเสียงที่ได้จาก RIM ซึ่งในย่านความถี่ต่ำ (0-340 Hz) และย่านความถี่กลาง (340-3,400 Hz) จะพบว่าม้องค์ประกอบที่ใกล้เคียงกันแต่เสียง

จาก RIM มีความแรงน้อยกว่าเสียงที่ผสมกันโดยตรง และในย่านความถี่สูง (3,400-20,000 Hz) ไม่สามารถแยกความต่างของข้อมูลได้



รูปที่ 2.4 แผนภาพของสเปกตรัมของเสียงที่ได้จากการผสมกันโดยตรง (Mix-audio)

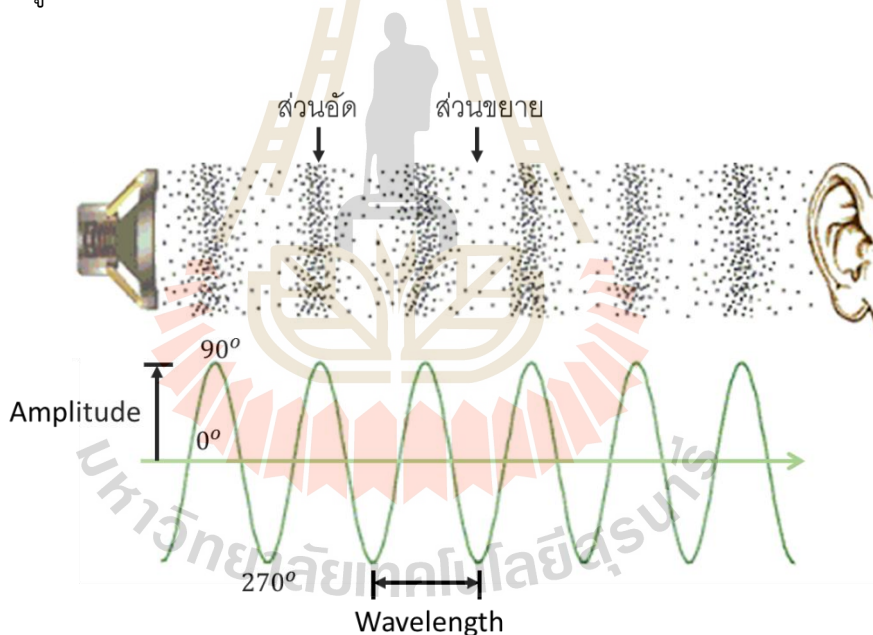


รูปที่ 2.5 แผนภาพของสเปกตรัมของเสียงที่ได้จาก RIM

2.2.3 เสียง (Audio)

เสียงเกิดจาก การสั่นของวัตถุ วัตถุสั่นด้วยวิธีการ ตี ตี และเป่า เมื่อแหล่งกำเนิดเสียงเกิดการสั่น จะทำให้โมเลกุลอากาศสั่นตามไปด้วยความถี่เท่ากับการสั่นของแหล่งกำเนิดเสียง เกิดเป็นช่วงอัด ช่วงขยายของโมเลกุลของอากาศ ซึ่งพลังงานของการสั่นจะแผ่ออกไปรอบ ๆ แหล่งกำเนิดเสียง ตรงกลางส่วนอัดและตรงกลางส่วนขยายโมเลกุลอากาศจะไม่มีเคลื่อนที่ แต่ตรงกลางส่วนอัดความดันอากาศจะมาก และตรงกลางส่วนขยายความดันอากาศจะน้อยมาก ดังนั้นคลื่นเสียงจึงเป็นคลื่นตามยาวเพราะโมเลกุลของอากาศจะสั่นในทิศเดียวกับทิศที่เสียงเคลื่อนที่ไป การส่งคลื่นผ่านตัวกลางจะเป็นลักษณะที่อัด - คลายเป็นช่วง ๆ ภาพกราฟที่สูงขึ้นของคลื่นเป็นการอัด (ตำแหน่ง 90 องศา) และกราฟต่ำลงเป็นการคลาย (ตำแหน่ง 270 องศา) ดังรูปที่ 2.6 ความดังของเสียงจะขึ้นอยู่กับช่วงกว้างของการสั่น (แอมพลิจูด) ถ้าแอมพลิจูดมาก เสียงจะดังมาก การเปลี่ยนความดันอากาศนี้สามารถเคลื่อนที่ไปข้างหน้า จนถึง หูของ ผู้ฟังทำให้ได้ยินเสียง

คุณลักษณะเฉพาะของเสียงได้แก่ ความถี่ ความยาวช่วงคลื่น แอมพลิจูด และความเร็วของเสียง ในเสียงแต่ละเสียงมีความแตกต่างกัน เสียงสูง-เสียงต่ำ เสียงดัง-เสียงเบา หรือคุณภาพของเสียงลักษณะต่าง ๆ ทั้งนี้ขึ้นอยู่กับแหล่งกำเนิดเสียง และจำนวนรอบต่อวินาทีของการสั่นสะเทือน ซึ่งมีรายละเอียดดังนี้



รูปที่ 2.6 ลักษณะการส่งผ่านคลื่น

2.2.3.1 ระดับเสียง (pitch) หมายถึง สิ่งที่ทำให้เสียงแต่ละเสียงสูงต่ำแตกต่างกันนั้นขึ้นอยู่กับความเร็วในการสั่นสะเทือนของวัตถุดังรูปที่ 2.6 วัตถุที่สั่นเร็ว เสียงจะสูงกว่าวัตถุที่สั่นช้า โดยจะมีหน่วยวัดคือ ความถี่ของการสั่นสะเทือนต่อวินาที (Hertz : Hz) เช่น 60 Hz 2,000 Hz เป็นต้น และความถี่ที่มนุษย์ได้ยินในปกติแล้วมนุษย์เราจะได้ยินเสียงที่มีช่วงความถี่ (frequency range) อยู่ที่ 20-20,000 Hz ส่วนเสียงที่มีค่าที่สูงกว่าหรือต่ำกว่าที่ 20-20,000 Hz โดยทั่วไปช่วงความถี่ของเสียงพูดจะอยู่ในช่วงความถี่ 500-3,000 Hz และมนุษย์จะไม่สามารถได้ยินเสียงนั้นได้ ซึ่งคลื่นเสียงความถี่ที่ต่ำกว่า 20 Hz จะเรียกว่าคลื่นใต้

(infrasound) หรือคลื่นอินฟราโซนิก (infrasonic wave) เป็นคลื่นกลประเภทคลื่นตามยาวซึ่งมีความถี่ต่ำกว่าคลื่นเสียงที่คนปกติได้ยิน เกิดจากการสั่นของตัวก่อกำเนิดเสียงขนาดใหญ่เช่น การสั่นของตึก แผ่นดินไหว เป็นต้น และคลื่นเสียงที่มีความถี่สูงกว่า 20,000 Hz เรียกว่า คลื่นเหนือเสียง (ultrasound) หรือคลื่นอัลตราโซนิก (ultrasonic wave) เป็นคลื่นกลประเภทคลื่นตามยาว ซึ่งเกิดจากการสั่นของตัวก่อกำเนิดเสียงขนาดเล็ก

2.2.3.2 ความยาวคลื่น (wavelength) หมายถึง ระยะทางระหว่างยอดคลื่นถึงยอดคลื่นที่ติดกัน ซึ่งเกิดขึ้นระหว่างการอัดตัวของคลื่นเสียง ถ้ามีความยาวช่วงคลื่นมากจะทำให้มีความถี่ของเสียงที่ต่ำ และถ้ามีความยาวช่วงคลื่นสั้นจะทำให้มีความถี่ของเสียงที่สูง

2.2.3.3 แอมพลิจูด (amplitude) หมายถึง ความสูงระหว่างยอดคลื่นและท้องคลื่นของคลื่นเสียงดังรูปที่ 2.6 ที่แสดงถึงความเข้มของเสียง (intensity) หรือความดังของเสียง (loudness) ยิ่งแอมพลิจูดมีค่ามาก ความเข้มหรือความดังของเสียงก็ยิ่งเพิ่มขึ้น

2.3 ปรัชญ์วรรณกรรมที่เกี่ยวข้อง

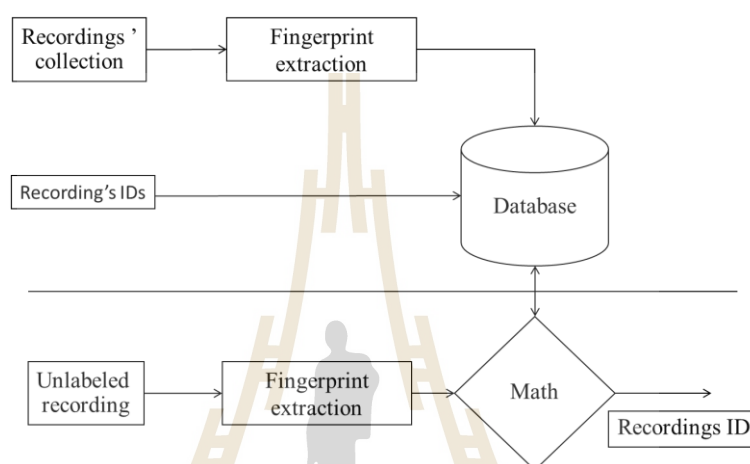
เพื่อให้เกิดความเข้าใจเกี่ยวกับการเปรียบเทียบความคล้ายกันของเสียงนั้น จำเป็นต้องศึกษาปรัชญ์วรรณกรรม และงานวิจัยที่เกี่ยวข้อง เพื่อเป็นความรู้และใช้เป็นแนวทางในการออกแบบสำหรับงานวิจัยนี้

งานวิจัยของ (P.Cano, E.Battle, T.Kalker, J.Haitsma, 2002) ได้นำเสนอกระบวนการในการเปรียบเทียบความคล้ายกันของเสียง โดยใช้หลักการของลายนิ้วมือเสียง (audio fingerprinting) ซึ่งมีหลักการทำงานเหมือนการพิสูจน์ลายนิ้วมือของตำรวจ โดยที่มนุษย์แต่ละคนจะมีลายนิ้วมือที่ต่างกัน เมื่อต้องการหาคนร้าย จะนำลายนิ้วมือมาทำการเปรียบเทียบกับลายนิ้วมือที่มีอยู่ในฐานข้อมูล ถ้าหากพบลายนิ้วมือที่ตรงกันตำรวจก็จะทราบถึงคนร้ายได้ เช่นเดียวกันกับระบบลายนิ้วมือเสียงจะมีกระบวนการสกัดลายนิ้วมือเสียง (audio fingerprinting extraction) เพื่อหาลักษณะเด่นของเสียงนั้น ๆ และจัดเก็บไว้ในฐานข้อมูลพร้อมกับระบุหมายเลขระบุตัวตน (identity number : ID) เพื่อเป็นข้อมูลอ้างอิง เมื่อต้องการหาความคล้ายกันของเสียงที่ไม่รู้จักจะทำการสกัดลายนิ้วมือเสียงและเปรียบเทียบกับฐานข้อมูลว่ามีข้อมูลที่ตรงกันหรือไม่ ถ้าตรงกันก็ให้ผลการเปรียบเทียบเป็นหมายเลขระบุตัวตนเพื่อบอกว่าเป็นเสียงเพลงอะไรหรือเพลงของใคร ดังรูปที่ 2.7

ในงานวิจัยได้ออกแบบการสกัดลายนิ้วมือเสียง มีกระบวนการเริ่มจากการแปลงข้อมูลเสียงให้เป็นสัญญาณดิจิทัล (digital) จากนั้นแบ่งข้อมูลเป็นช่วงข้อมูลเป็นส่วนย่อย ๆ เรียกว่าเฟรม (frame) และทำการแปลงให้เป็นโดเมนทางความถี่ด้วยฟังก์ชัน Fast Fourier Transform (FFT) จากนั้นทำการสร้างความเชื่อมโยงกันของลักษณะเด่นทางความถี่ และแปลงลักษณะเด่นทางความถี่ให้เป็นชุดข้อมูลเพื่อทำการบันทึกหรือทำการเปรียบเทียบต่อไป

งานวิจัยของ (Haitsma, Jaap and Kalker, Antonius, 2002) ได้เสนอการนำระบบลายนิ้วมือเสียงมาใช้ในการเปรียบเทียบความคล้ายกันของเสียง และพัฒนากระบวนการสกัดลายนิ้วมือเสียงโดยการแบ่ง

ข้อมูลให้เป็นเฟรมย่อยที่มีความยาว 11.8 ms จากนั้นใช้ฟังก์ชัน Fast Fourier Transform (FFT) และแบ่งข้อมูลที่ได้เป็น 32 ช่องสัญญาณและทำการคำนวณลักษณะเด่นทางความถี่จากพลังงานของแต่ละช่องสัญญาณที่แตกต่างกัน โดยแต่ละลายนิ้วมือเสียงจะมีเฟรมย่อยต่อกันจำนวน 256 เฟรม ซึ่งทำให้ลายนิ้วมือเสียงที่ได้จะมีขนาดที่แน่นอน และมีขนาดที่เล็กลง นอกจากนี้ได้พัฒนากระบวนการเปรียบเทียบความคล้ายกันของเสียงให้มีประสิทธิภาพมากขึ้น



รูปที่ 2.7 ขั้นตอนการทำงานของระบบลายนิ้วมือเสียง

งานวิจัยของ (Avery Li-Chun Wang, 2003) เป็นงานวิจัยของ Shazam Entertainment, Ltd. ที่มีแอปพลิเคชัน (application) ในการค้นหาเพลงผ่านโทรศัพท์สมาร์ทโฟนที่ใช้งานกันอย่างแพร่หลาย โดยในงานวิจัยได้ใช้ระบบลายนิ้วมือเสียงในการเปรียบเทียบเพื่อค้นหาเพลงที่เหมือนกันในระบบฐานข้อมูลเช่นกัน และได้พัฒนากระบวนการสกัดลายนิ้วมือเสียงเป็นของตัวเอง โดยการแบ่งข้อมูลเป็นเฟรมย่อยและใช้ฟังก์ชัน Fast Fourier Transform (FFT) และพิจารณาข้อมูลในรูปแบบกราฟ Spectrogram และหาจุดที่มีกำลังสูงสุดในแต่ละเฟรมย่อย เพื่อทำการสร้างความสัมพันธ์กันของข้อมูลผ่านจุดที่มีกำลังสูงสุดจากแต่ละเฟรมย่อย และแปลงข้อมูลความสัมพันธ์โดยฟังก์ชันแฮช (hash function) ซึ่งให้กระบวนการเปรียบเทียบความคล้ายกันของเสียงกับฐานข้อมูลมีประสิทธิภาพมากขึ้น

2.4 อุปกรณ์ที่ใช้ในงานวิจัย

2.4.1 บอร์ด Raspberry Pi

บอร์ด Raspberry Pi เป็นบอร์ดคอมพิวเตอร์ขนาดเล็กที่สามารถเชื่อมต่อกับจอมอนิเตอร์ คีย์บอร์ด และเมาส์ได้ สามารถนำมาประยุกต์ใช้ในการทำโครงงานทางด้านอิเล็กทรอนิกส์ การเขียนโปรแกรม หรือเป็นเครื่องคอมพิวเตอร์ตั้งโต๊ะขนาดเล็ก ส่งอีเมล หรือเล่นเกมส์ อีกทั้งยังสามารถเล่นไฟล์วีดีโอความละเอียดสูง (High-Definition) ได้อีกด้วย และบอร์ด Raspberry Pi รองรับระบบปฏิบัติการลินุกซ์ (Linux Operating System) ได้หลายระบบ เช่น Raspbian (Debian) Pidora (Fedora) และ Arch Linux เป็นต้น โดยติดตั้งบน SD Card นอกจากนี้บอร์ด Raspberry Pi นี้ถูกออกแบบมาให้มี CPU GPU และ RAM อยู่ภายในชิปเดียวกัน และมีจุดเชื่อมต่อ GPIO ให้ผู้ใช้สามารถนำไปใช้ร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่น ๆ

ข้อมูลจำเพาะของบอร์ด Raspberry Pi 2 Model B

- ขนาด 85 x 56 x 17 mm
- CPU: Broadcom BCM2836 ARMv7 Quad-Core 900MHz
- RAM: LPDDR2 SDRAM 1GB @ 450MHz
- GPU: VideoCore IV 3D Graphic Core
- 4 x USB 2.0 Host port
- Stereo audio out 3.5 mm jack
- 1 x Micro SD Card slot
- 1 x 10/100 Mbps Fast Ethernet via RJ-45
- Full-size HDMI port
- 40-pin GPIO header (GPIO, UART, I2C, SPI, etc.)



รูปที่ 2.8 บอร์ด Raspberry Pi

2.4.2 USB Audio Capture

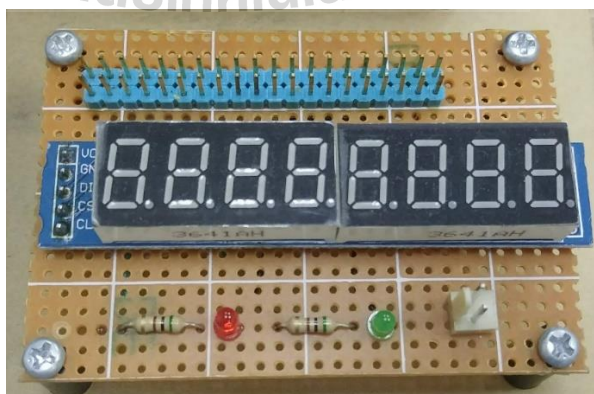
USB Audio Capture เป็นการ์ดเสียงที่ทำหน้าที่แปลงข้อมูลดิจิทัลที่เก็บรายละเอียดเกี่ยวกับเสียงต่าง ๆ เป็นอุปกรณ์เชื่อมต่อภายนอกเชื่อมต่อสื่อสารกับคอมพิวเตอร์ผ่านช่อง USB ที่ใช้งานได้ง่าย ชื้อได้ง่าย ราคาถูก ไม่ต้องใช้ไฟเลี้ยงจากภายนอก การ์ดเสียงนี้มีลักษณะทางกายภาพดังภาพที่ 2.9 ซึ่งมี ส่วนต่าง ๆ ดังนี้ ช่องต่อขนาด 3.5 มิลลิเมตรสำหรับขาเข้า 1 ช่อง (สีเขียว) และขาออก 1 ช่อง (สีเหลือง) LED แสดงสถานะการใช้งาน ปุ่มสำหรับเพิ่มลดความดังขาออก ปุ่มสำหรับปิดรับเสียงขาเข้าและขาออก และ USB สำหรับเชื่อมต่อไปยังคอมพิวเตอร์



รูปที่ 2.9 USB Audio Capture

2.4.3 ส่วนการแสดงผล

ในส่วนการแสดงผลนี้ใช้งาน LCD 8 ตำแหน่งมาใช้งาน เพื่อแสดงคะแนนของการเปรียบเทียบความคล้ายกันของเสียง แล้ใช้งานหลอด LED เพื่อแสดงสถานะการทำงาน และช่องสื่อสารกับเครื่องรับวิทยุ เพื่อส่งสัญญาณการเปลี่ยนช่องความถี่ของสถานีที่กำลังออกอากาศ และส่วนเชื่อมต่อกับบอร์ด Raspberry Pi ผ่านช่อง GPIO



รูปที่ 2.10 ส่วนการแสดงผล

2.5 สรุป

เนื้อหาในบทนี้ได้กล่าวถึงทฤษฎีการเกิดอินเตอร์มอดูเลชันที่เป็นสาเหตุของการรบกวนการสื่อสารระหว่างเครื่องบินและสถานีภาคพื้น และทำให้เราทราบว่าความถี่อินเตอร์มอดูเลชันจะมีข้อมูลของทั้ง 2 ความถี่ที่เป็นต้นกำเนิดผสมกันอยู่ ซึ่งจะยืนยันว่าการเปรียบเทียบความคล้ายกันของเสียงระหว่างเสียงที่ได้จากความถี่อินเตอร์มอดูเลชันกับเสียงจากสถานีวิทยุที่กำลังออกอากาศ จะสามารถระบุสถานีวิทยุที่เป็นต้นกำเนิดของสัญญาณรบกวนได้ และการศึกษาปริทัศน์วรรณกรรมที่เกี่ยวข้องทำให้ทราบว่าอัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียงที่พบส่วนมาก ใช้ระบบลายนิ้วมือเสียงในการเปรียบเทียบ ซึ่งกระบวนการสกัดลายนิ้วมือเสียง และการเปรียบเทียบกับฐานข้อมูลเป็นกระบวนการที่ซับซ้อน จำเป็นต้องใช้ทรัพยากรในการคำนวณมาก ทำให้ต้องการคอมพิวเตอร์สมรรถนะสูงในการทำงาน และต้องการฐานข้อมูลเพื่อใช้ในการเปรียบเทียบ ซึ่งไม่เหมาะกับการเปรียบเทียบความคล้ายกันของเสียงที่ทำงานแบบเวลาจริง และเสนอรายละเอียดของอุปกรณ์ที่จะนำมาใช้งานในงานวิจัยนี้



บทที่ 3

อัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียง ที่ทำงานแบบเวลาจริง

3.1 บทนำ

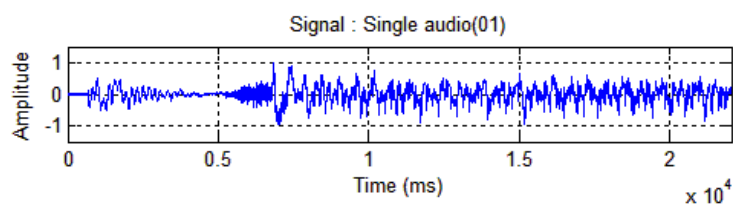
จากเนื้อหาในบทที่ 2 ทำให้ทราบว่า การรบกวนจาก Reverse Intermodulation (RIM) ไปรบกวนการสื่อสารระหว่างเครื่องบินและสถานีควบคุมภาคพื้น โดยมีสาเหตุจากมีสถานีวิทยุกระจายเสียง 2 สถานีทำการออกอากาศในพื้นที่ใกล้เคียงกัน ซึ่งเสียงของสัญญาณรบกวนนั้นเป็นเสียงที่มีการผสมจากทั้ง 2 สถานีวิทยุที่เป็นต้นกำเนิดของสัญญาณรบกวน และในปัจจุบันการค้นหาสถานีวิทยุที่เป็นต้นกำเนิดของสัญญาณรบกวน ดำเนินการโดยบุคลากรที่มีความชำนาญ ซึ่งจะการฟังและเปรียบเทียบความคล้ายกันของเสียงจากสถานีวิทยุที่กำลังออกอากาศ และเสียงจากคลื่นสัญญาณรบกวน แต่การเปรียบเทียบความคล้ายกันของเสียงเป็นไปอย่างล่าช้าจึงจัดการปัญหาได้ไม่ทันเวลา

จากการศึกษาปริทัศน์วรรณกรรมเกี่ยวข้องทำให้ทราบว่าอัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียงที่มีในปัจจุบัน มีการคำนวณที่ซับซ้อนจึงต้องการหน่วยประมวลผลที่มีสมรรถนะสูง และต้องการฐานข้อมูลในการเปรียบเทียบ ซึ่งไม่เหมาะสมที่จะใช้งานในการค้นหาสถานีวิทยุที่เป็นต้นกำเนิดของสัญญาณรบกวน จึงพัฒนาอัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียงที่ทำงานแบบเวลาจริง ที่มีการคำนวณทางคณิตศาสตร์อย่างง่าย ต้องการทรัพยากรในการคำนวณต่ำ และทำงานได้อย่างรวดเร็ว

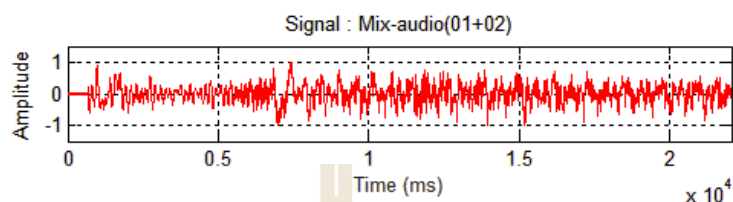
3.2 อัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียงที่ทำงานแบบเวลาจริง

ในงานวิจัยนี้เสนออัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียงจาก 2 แหล่งคือเสียงจากสถานีวิทยุที่กำลังออกอากาศ ซึ่งเป็นเสียงที่ไม่มีการผสมเสียงจากสถานีอื่น จะเรียกว่าเสียงเดี่ยว (Single audio) และเสียงจากสัญญาณรบกวน ซึ่งเป็นเสียงที่เกิดจากการผสมเสียงจากทั้งสองสถานีที่เป็นต้นกำเนิดของสัญญาณรบกวน จะเรียกว่าเสียงผสม (Mix audio)

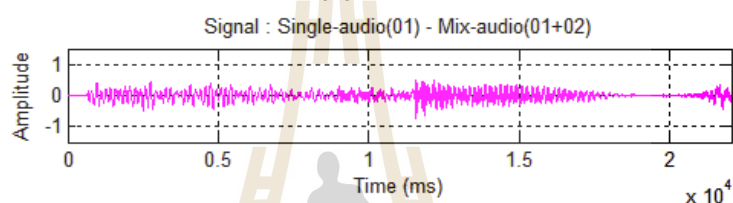
แนวคิดในการเปรียบเทียบความคล้ายกันของเสียงในงานวิจัยนี้ คือการหาผลต่างของแอมพลิจูด (amplitude) ในทางเวลา (time domain) ของเสียงเดี่ยวและเสียงผสม ถ้าหากทั้งสองเสียงมีความคล้ายกันจะทำให้ผลต่างของแอมพลิจูดมีขนาดใกล้เคียง 0 แสดงดังรูปที่ 3.1 ในทางตรงกันข้ามหากเสียงที่เปรียบเทียบไม่มีความคล้ายกัน ผลต่างของแอมพลิจูดที่ได้จะมีขนาดที่สูงขึ้น แสดงดังรูปที่ 3.2 และนำผลต่างของแอมพลิจูดที่ได้มาพิจารณาเพื่อตัดสินความคล้ายกันของเสียง



ก) สัญญาณเสียงเดี่ยว

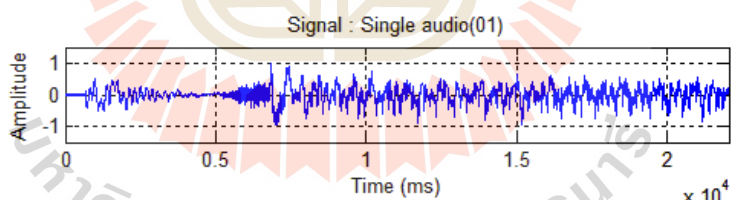


ข) สัญญาณเสียงผสม

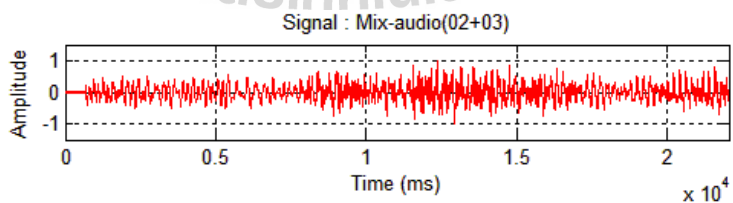


ค) ผลต่างทางแอมพลิจูดของสัญญาณเสียงที่มีความคล้ายกัน

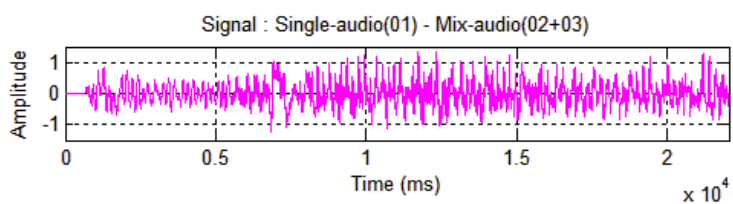
รูปที่ 3.1 สัญญาณเสียงในทางเวลาของสัญญาณเสียงที่มีความคล้ายกัน



ก) สัญญาณเสียงเดี่ยว



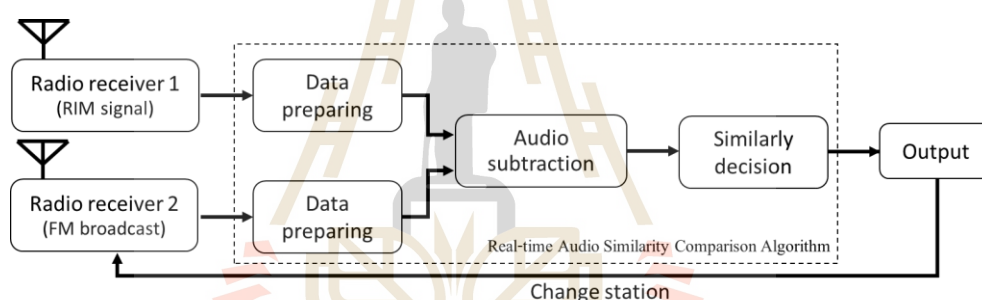
ข) สัญญาณเสียงผสม



ค) ผลต่างทางแอมพลิจูดของสัญญาณเสียงที่ต่างกัน

รูปที่ 3.2 สัญญาณเสียงในทางเวลาของสัญญาณเสียงที่ไม่มีความคล้ายกัน

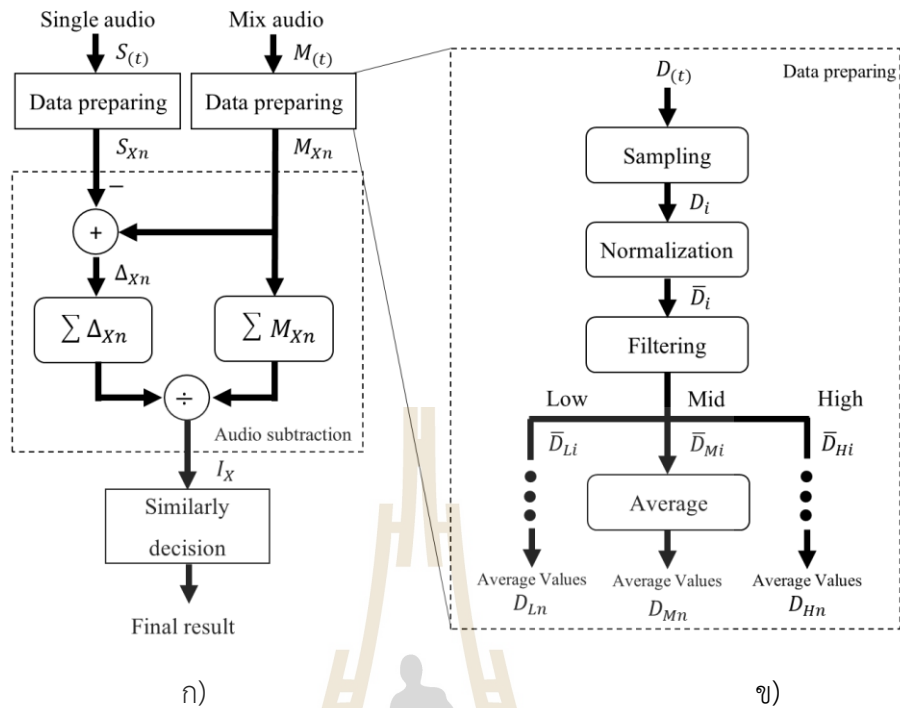
เมื่อนำอัลกอริทึมที่เสนอในงานวิจัยมาใช้งานในการค้นหาสถานีที่เป็นต้นกำเนิดของสัญญาณรบกวน จะมีขั้นตอนดังรูปที่ 3.3 โดยมีเครื่องรับวิทยุ 2 เครื่องจะรับคลื่นความถี่ที่ต่างกันคือ เครื่องที่ 1 รับคลื่นจากสัญญาณรบกวนที่เกิดขึ้น และเครื่องที่ 2 รับคลื่นจากสถานีวิทยุที่กำลังออกอากาศ และนำสัญญาณเสียงทั้งสองมาผ่านกระบวนการเตรียมข้อมูล (Data preparing) เพื่อให้สัญญาณเสียงพร้อมในการคำนวณทางคอมพิวเตอร์ จากนั้นนำทั้งสองสัญญาณเสียงมาคำนวณหาผลต่างของแอมพลิจูดในกระบวนการคำนวณผลต่างของเสียง (Audio subtraction) และนำค่าผลต่างของแอมพลิจูดที่ได้มาตัดสินความคล้ายกันของเสียงในกระบวนการตัดสินความคล้าย (Similarly decide) และถ้าหากพบว่าทั้งสองสัญญาณเสียงมีความคล้ายกันจะให้เอาต์พุต (output) เพื่อแจ้งเตือนว่าความถี่ที่เครื่องรับวิทยุเครื่องที่ 2 รับคลื่นอยู่เป็นต้นกำเนิดของสัญญาณรบกวน แต่ถ้าพบว่าทั้งสองสัญญาณเสียงต่างกัน จะสั่งการให้เครื่องรับวิทยุเครื่องที่ 2 เลื่อนไปรับคลื่นความถี่ที่กำลังออกอากาศในช่องถัดไปโดยอัตโนมัติ เพื่อให้การค้นหาสถานีที่เป็นต้นกำเนิดของสัญญาณรบกวนเป็นไปอย่างต่อเนื่อง



รูปที่ 3.3 ขั้นตอนของการค้นหาสถานีที่เป็นต้นกำเนิดของสัญญาณรบกวน

3.3 หลักการเปรียบเทียบความคล้ายกันของเสียง

หลักการทำงานของอัลกอริทึมที่นำเสนอ จะแบ่งการทำงานออกเป็น 3 กระบวนการย่อยดังนี้ 1. การเตรียมข้อมูล (Data preparing) ซึ่งจะซ้กตัวอย่าง (Sampling) ให้สัญญาณเสียงเป็นสัญญาณไม่ต่อเนื่อง (Discrete signal) เพื่อให้สามารถนำข้อมูลไปทำการคำนวณทางคอมพิวเตอร์ได้และพร้อมสำหรับการเปรียบเทียบความคล้ายกันของเสียง 2. กระบวนการคำนวณผลต่างของเสียง (Audio subtraction) ซึ่งจะนำข้อมูลของทั้งสองสัญญาณเสียงมาคำนวณหาผลต่างของแอมพลิจูด 3. กระบวนการตัดสินความคล้าย (Similarly decision) ซึ่งจะนำค่าผลต่างของแอมพลิจูดที่ได้มาตัดสินผลการเปรียบเทียบความคล้ายกันของเสียง ซึ่งรายละเอียดของแต่ละกระบวนการมีดังนี้



รูปที่ 3.4 หลักการทำงานของอัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียง ที่ทำงานแบบเวลาจริง

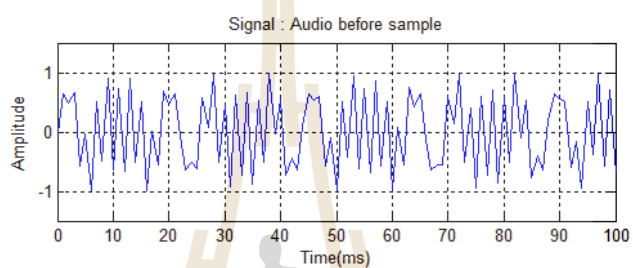
3.3.1 กระบวนการเตรียมข้อมูล (Data preparing)

ในกระบวนการนี้จะทำการเตรียมข้อมูลของทั้งสองสัญญาณเสียง เพื่อให้พร้อมสำหรับการคำนวณผลต่างของแอมพลิจูด โดยมีขั้นตอนการทำงานแสดงดังรูปที่ 3.4(ข) ซึ่งทำการซีกตัวอย่าง (sampling) สัญญาณต่อเนื่อง (continuous signal) เพื่อให้เป็นสัญญาณไม่ต่อเนื่อง (discrete signal) ซึ่งสัญญาณเสียง ($D(t)$) ที่ผ่านขั้นตอนซีกตัวอย่างจะเรียกว่าข้อมูลสัญญาณเสียง (D_i) และการเปรียบเทียบความคล้ายกันของเสียงในแต่ละครั้งจะพิจารณาสัญญาณเสียงที่มีความยาว 10 วินาที

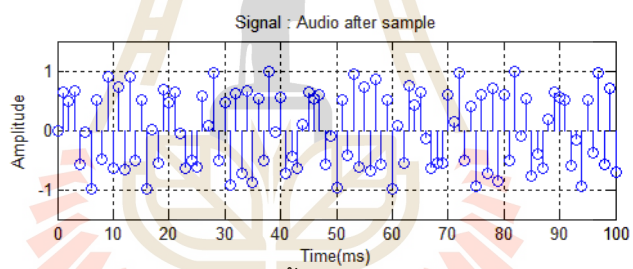
เนื่องจากทั้งสองสัญญาณเสียงที่จะทำการเปรียบเทียบกันนั้นมีแอมพลิจูดที่ต่างกัน ซึ่งอาจทำให้ผลต่างของแอมพลิจูดเกิดความคลาดเคลื่อนได้ จึงทำนอร์มัลไลเซชัน (Normalization) ข้อมูลสัญญาณเสียง (D_i) ด้วยสมการที่ 3.1 จะได้ข้อมูลสัญญาณเสียงที่ผ่านนอร์มัลไลเซชัน (\bar{D}_i) ที่มีกำลังของทั้งสองสัญญาณเสียงที่เท่ากัน และมีขนาดของแอมพลิจูดที่ใกล้เคียงกัน

$$\bar{D}_i = \frac{D_i}{\sqrt{\sum(D_i)^2}} \quad (3.1)$$

โดยที่ \bar{D}_i คือ ข้อมูลสัญญาณเสียงที่ผ่านนอร์มัลไลเซชัน
 D_i คือ ข้อมูลสัญญาณเสียง
 i คือ ตำแหน่งของข้อมูล ($i = 1,2,3,\dots$)

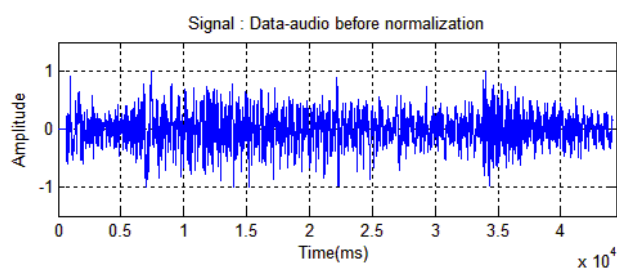


ก) ก่อนผ่านขั้นตอนซิกตัวอย่าง

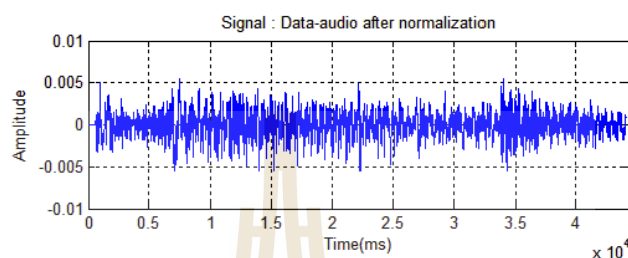


ข) หลังผ่านขั้นตอนซิกตัวอย่าง

รูปที่ 3.5 สัญญาณเสียงในขั้นตอนซิกสัญญาณ



ก) ก่อนผ่านขั้นตอนนอร์มัลไลเซชัน



ข) หลังผ่านขั้นตอนนอร์มัลไลเซชัน

รูปที่ 3.6 ข้อมูลสัญญาณเสียงในขั้นตอนนอร์มัลไลเซชัน

จากรูปที่ 3.6 จะเห็นว่าข้อมูลสัญญาณเสียงก่อนทำมัลไลเซชัน มีแอมพลิจูดสูงสุดถึงต่ำสุดเท่ากับ 1 ถึง -1 และหลังจากผ่านขั้นตอนนอร์มัลไลเซชัน แอมพลิจูดของข้อมูลสัญญาณมีขนาดที่เล็กลง ซึ่งมีความแรงสูงสุดถึงต่ำสุดเท่ากับ 0.01 ถึง -0.01 และทำให้สัญญาณเสียงที่ผ่านกระบวนการเตรียมข้อมูลจะมีขนาดของแอมพลิจูดที่ใกล้เคียงกัน

เพื่อเพิ่มความแม่นยำในการตัดสินใจคล้ายกันของเสียง จึงแยกพิจารณาข้อมูลเป็น 3 ย่านความถี่ โดยนำข้อมูลที่ได้จากขั้นตอนนอร์มัลไลเซชันมาผ่านตัวกรองสัญญาณ ซึ่งรายละเอียดของตัวกรองสัญญาณแต่ละย่านความถี่มีดังตารางต่อไปนี้

ตารางที่ 3.1 รายละเอียดของแต่ละย่านความถี่ในขั้นตอนกรองสัญญาณ

ย่านความถี่	ชนิดตัวกรอง	ความถี่ที่ทำงาน
ความถี่ต่ำ (Low)	ตัวกรองความถี่ต่ำผ่าน (low pass filter)	ต่ำกว่า 340 Hz
ความถี่กลาง (Mid)	ตัวกรองผ่านแถบ (band pass filter)	340 – 3,400 Hz
ความถี่สูง (High)	ตัวกรองความถี่สูงผ่าน (high pass filter)	สูงกว่า 3,400 Hz

ในการออกแบบตัวกรองสัญญาณในงานวิจัยนี้ได้เลือกใช้ตัวกรองแบบ Elliptic filter design ซึ่งจะมี Transfer Function ดังนี้

$$H(z) = \frac{B(z)}{A(z)} = \frac{b(1) + b(2)z^{-1} + \dots + b(n+1)z^{-n}}{a(1) + a(2)z^{-1} + \dots + a(n+1)z^{-n}} \quad (3.2)$$

โดยมีพารามิเตอร์สำหรับการออกแบบตัวกรองสัญญาณดังนี้

Order (n) = 6

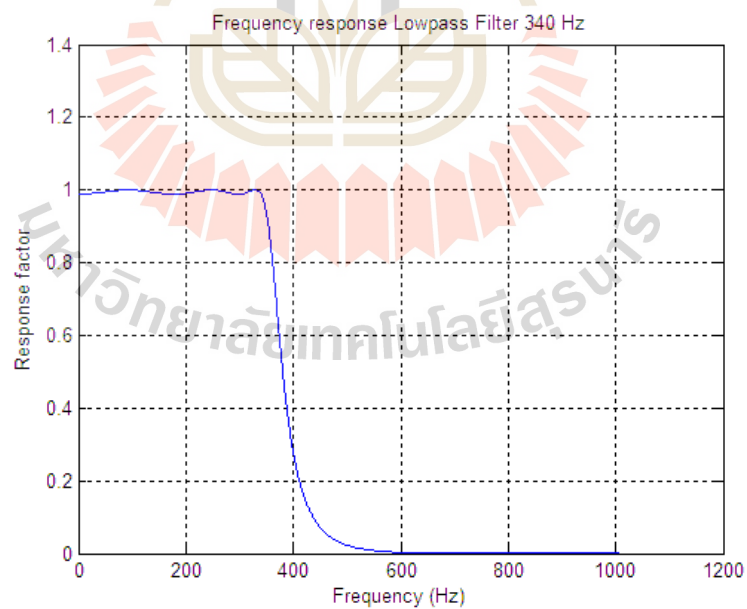
Ripple in the passband (Rp) = 0.1

Ripple in the stopband (Rs) = 80

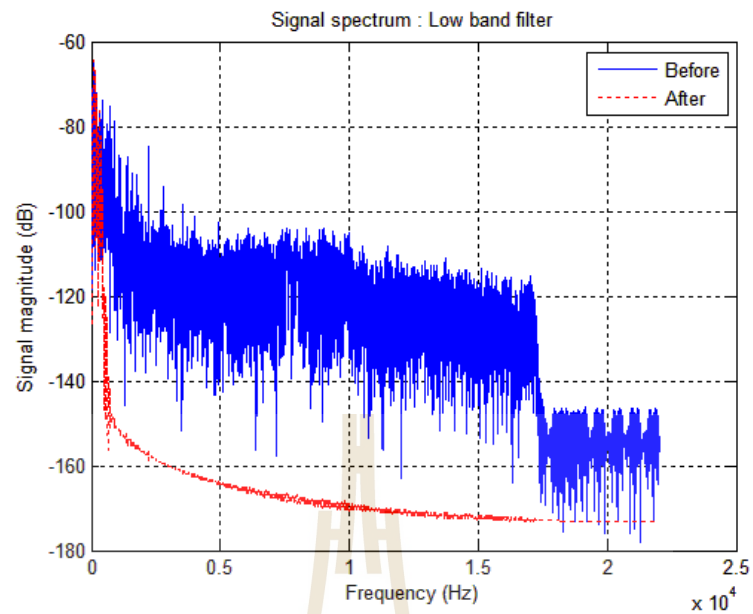
Normalized passband edge frequency (Wn) = [340 3400]*2/Fs;

Type = “lowpass”, “bandpass”, “highpass”

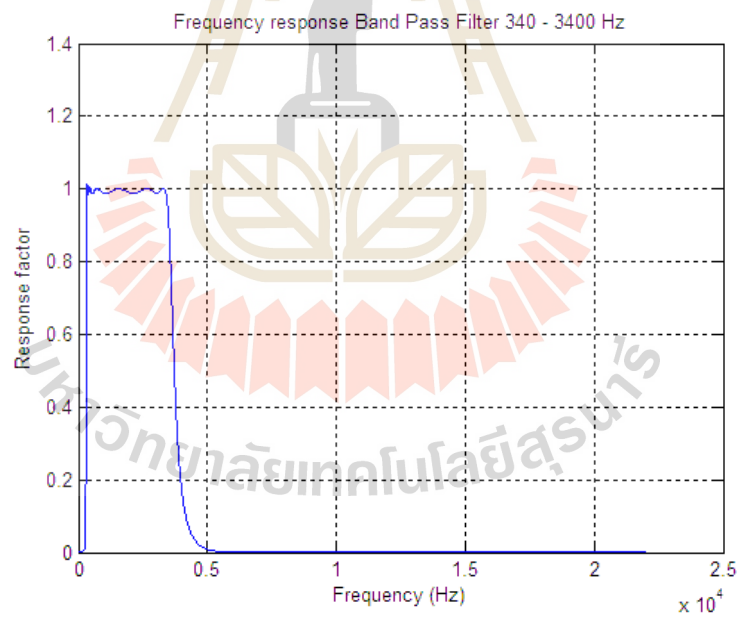
ซึ่งตัวกรองสัญญาณแต่ละย่านความถี่ได้ผลตอบสนองทางความถี่ (Frequency Respond) และผลของข้อมูลสัญญาณเสียงที่ผ่านตัวกรองสัญญาณในแต่ละย่านความถี่มีดังนี้



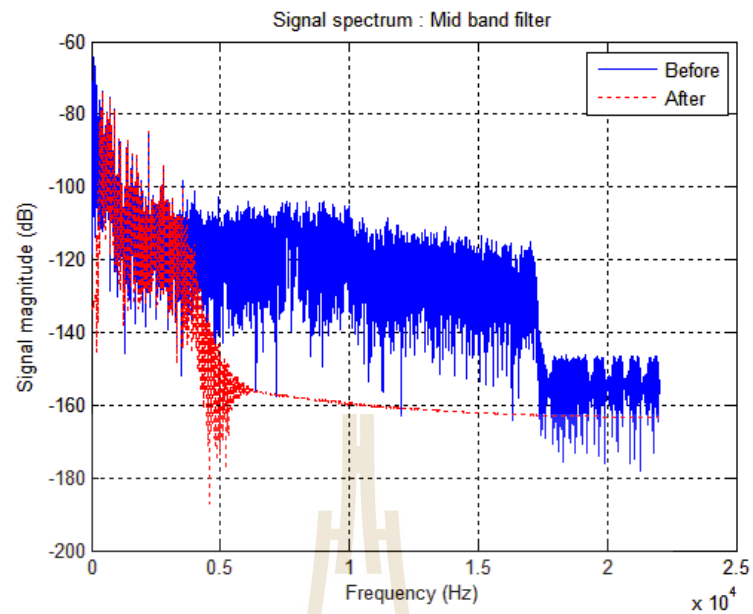
รูปที่ 3.7 ผลตอบสนองทางความถี่ (Frequency respond) ของช่วงความถี่ต่ำ



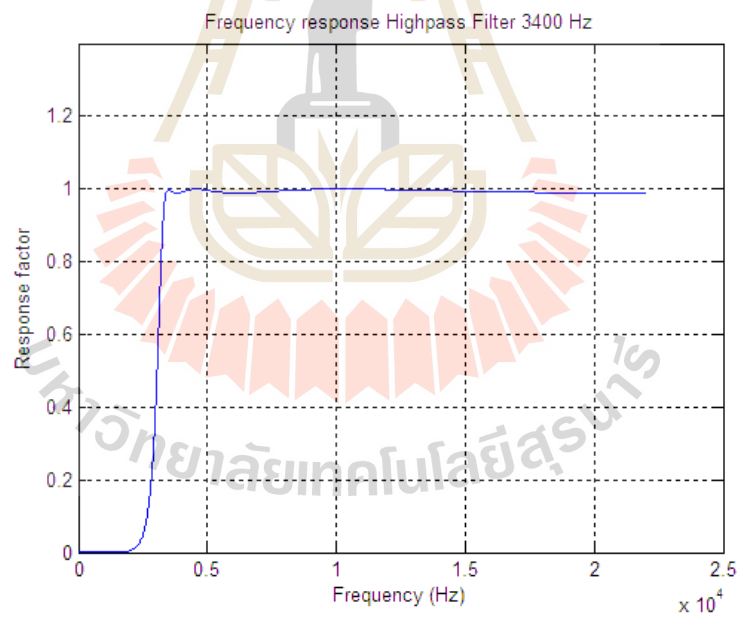
รูปที่ 3.8 ผลของข้อมูลสัญญาณเสียงที่ผ่านตัวกรองสัญญาณช่วงความถี่ต่ำ



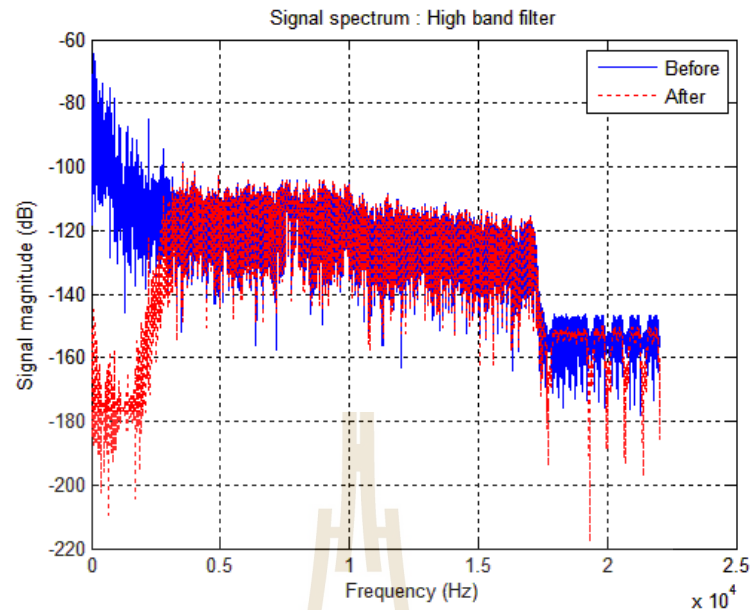
รูปที่ 3.9 ผลตอบสนองทางความถี่ (Frequency respond) ของช่วงความถี่กลาง



รูปที่ 3.10 ผลของข้อมูลสัญญาณเสียงที่ผ่านตัวกรองสัญญาณช่วงความถี่กลาง



รูปที่ 3.11 ผลตอบสนองทางความถี่ (Frequency respond) ของช่วงความถี่สูง



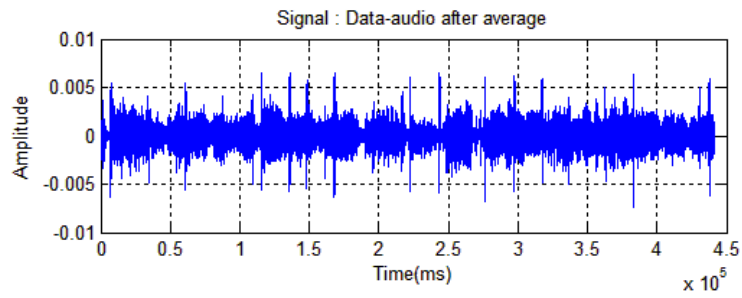
รูปที่ 3.12 ผลของข้อมูลสัญญาณเสียงที่ผ่านตัวกรองสัญญาณช่วงความถี่สูง

ถ้าหากทั้งสองสัญญาณเสียงมีการคลาดเคลื่อนทางเวลาเกิดขึ้นเล็กน้อยในขั้นตอนการบันทึกเสียง อาจทำให้ผลของการเปรียบเทียบเกิดการผิดพลาดได้ จึงทำการเฉลี่ยข้อมูล (Average) ด้วยสมการ 3.3 เพื่อให้การเปรียบเทียบสามารถทนทานต่อการคลาดเคลื่อนทางเวลาที่จะเกิดขึ้นได้

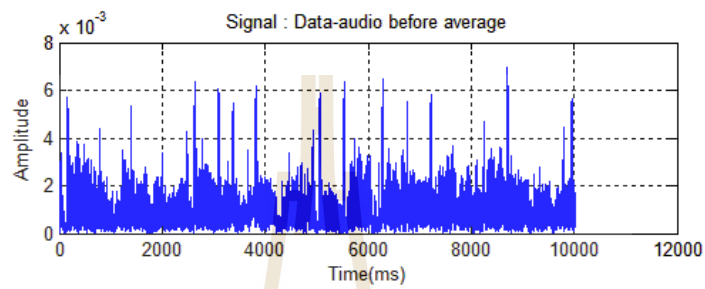
$$E_{Xn} = \langle |\bar{D}_{Xi}| \rangle = \frac{\sum_{i=1}^k |\bar{D}_{Xi}|}{k} \quad (3.3)$$

$$k = ws \times Fs \quad (3.4)$$

โดยที่	E_{Xn}	คือ ข้อมูลสัญญาณเสียงที่ผ่านขั้นตอนเฉลี่ยข้อมูล
	\bar{D}_{Xi}	คือ ข้อมูลสัญญาณเสียงที่ผ่านขั้นตอนกรอง
	x	คือ ตัวชี้ย่านความถี่ตัวกรอง (L-low, M-middle, H-high)
	n	คือ ตำแหน่งของข้อมูลที่ผ่านการเฉลี่ยข้อมูล
	k	คือ จำนวนข้อมูลของหน้าต่างรับข้อมูล
	ws	คือ ขนาดหน้าต่างรับข้อมูลขนาด 1 มิลลิวินาที
	Fs	คือ อัตราการสุ่มสัญญาณ



ก) ก่อนผ่านขั้นตอนเฉลี่ยข้อมูล



ข) หลังผ่านขั้นตอนเฉลี่ยข้อมูล

รูปที่ 3.13 ข้อมูลสัญญาณเสียงในขั้นตอนเฉลี่ยข้อมูล

จากรูปที่ 3.13 ข้อมูลสัญญาณเสียงที่ผ่านขั้นตอนเฉลี่ยข้อมูลจะมีความยาวที่สั้นลง และมีแอมพลิจูดของข้อมูลเป็นบวก ซึ่งพร้อมสำหรับกระบวนการถัดไป

3.3.2 กระบวนการคำนวณผลต่างของเสียง (Audio subtraction)

กระบวนการนี้ทำการคำนวณผลต่างของแอมพลิจูดของข้อมูลสัญญาณเสียงเดี่ยว (S_{Xn}) และข้อมูลสัญญาณเสียงรวม (M_{Xn}) ในแต่ละย่านความถี่ดังสมการที่ 3.5 และ 3.6 โดยการนำผลรวมของผลต่างของข้อมูลสัญญาณเสียงเดียวกับสัญญาณเสียงผสมมาหารด้วยผลรวมของข้อมูลสัญญาณเสียงผสม จะได้ค่าคงที่ที่เรียกว่าค่าความคล้ายกันของเสียง (I_X) ของแต่ละย่านความถี่ เพื่อใช้ในการตัดสินใจความคล้ายกันของเสียงในกระบวนการถัดไป

$$\Delta_{Xn} = |M_{Xn} - S_{Xn}| \quad (3.5)$$

$$I_X = \frac{\sum \Delta_{Xn}}{\sum M_{Xn}} \quad (3.6)$$

โดยที่	I_X	คือ ค่าความคล้ายกันของเสียง
	Δ_{Xn}	คือ ผลต่างของแอมพลิจูดของทั้งสองสัญญาณเสียง
	S_{Xn}	คือ ข้อมูลของสัญญาณเสียงเดี่ยว
	M_{Xn}	คือ ข้อมูลของสัญญาณเสียงผสม

3.3.3 กระบวนการตัดสินใจความคล้าย (Similarly decision)

ในกระบวนการตัดสินใจความคล้ายกันของเสียง จะนำค่าความคล้ายกันของเสียง (I_X) มาเทียบกับค่าเทรชโฮลด์ (threshold) ของแต่ละย่านความถี่ ถ้าหากพบว่าค่าความคล้ายกันของเสียงมีค่าน้อยกว่าค่าเทรชโฮลด์จะตัดสินใจว่าย่านความถี่นั้นมีความคล้ายกันของเสียง ตรงกันข้ามถ้าค่าความคล้ายกันของเสียงมีค่ามากกว่าค่าเทรชโฮลด์จะตัดสินใจว่าย่านความถี่นั้นไม่มีความคล้ายกันของเสียง และทำการรวมผลจากทั้ง 3 ย่านความถี่ เพื่อทำการโหวต 2 ใน 3 เสียง (2 of 3 voting) ถ้าหากพบว่ามีความคล้ายกันของเสียงมากกว่าหรือเท่ากับ 2 ย่านความถี่ จะให้ผลตัดสินใจการเปรียบเทียบว่ามีความคล้ายกันของเสียง (Similar) ตรงกันข้ามถ้าพบว่ามีความคล้ายกันของเสียงเพียง 1 ย่านความถี่หรือไม่พบว่ามีความคล้ายกันของเสียงจากทั้ง 3 ย่านความถี่ จะให้ผลตัดสินใจการเปรียบเทียบว่าไม่มีความคล้ายกันของเสียง (Dissimilar) ซึ่งค่าเทรชโฮลด์ที่ใช้ในการตัดสินใจความคล้ายกันของเสียงในกระบวนการนี้จะได้จากการทดลองที่จะกล่าวถึงในบทถัดไป

3.4 การใช้งานอัลกอริทึมการเปรียบเทียบความคล้ายกันของเสียงที่ทำงานแบบเวลาจริง ในการค้นหาสถานีที่เป็นต้นกำเนิดของสัญญาณรบกวน

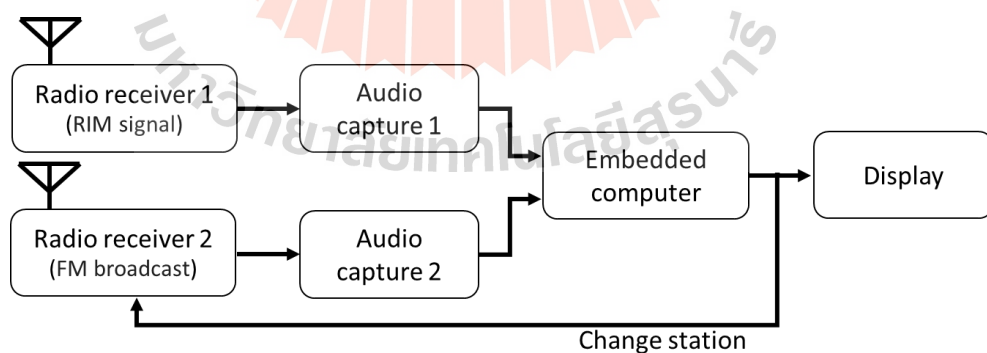
เนื้อหาในส่วนนี้นำเสนอการนำอัลกอริทึมที่พัฒนามาใช้งานในการค้นหาสถานีที่เป็นต้นกำเนิดของสัญญาณรบกวน โดยมีขั้นตอนการทำงานที่อ้างอิงจากขั้นตอนค้นหาสถานีที่เป็นต้นกำเนิดของสัญญาณรบกวนที่กล่าวในหัวข้อ 3.2 และเพิ่มขั้นตอนการให้คะแนนการเปรียบเทียบเพื่อเพิ่มความแม่นยำในการตัดสินใจมากขึ้น ซึ่งมีขั้นตอนและอุปกรณ์ที่จะใช้งานแสดงดังรูปที่ 3.14 มีรายละเอียดของแต่ละส่วนดังนี้

ส่วนที่ 1 เครื่องรับวิทยุเพื่อรับ (Radio receiver) ทำหน้าที่เลือกคลื่นวิทยุที่กำลังออกอากาศ และแยกข้อมูลข่าวสารออกมาในรูปแบบสัญญาณเสียง โดยที่ในระบบนี้ใช้งานเครื่องรับวิทยุ 2 เครื่อง เพื่อใช้ในการรับจากคลื่นรบกวนที่เกิดขึ้น และรับคลื่นของสถานีวิทยุที่กำลังออกอากาศ

ส่วนที่ 2 ตัวจับสัญญาณเสียง (Audio capture) ทำหน้าที่บันทึกสัญญาณเสียงจากทั้งสองเครื่องรับวิทยุ และแปลงสัญญาณเสียงที่เป็นอนาล็อก (Analog) ให้เป็นดิจิทัล (Digital) เพื่อให้สามารถคำนวณทางคอมพิวเตอร์ได้

ส่วนที่ 3 คอมพิวเตอร์ส่วนบุคคลขนาดเล็กหรือคอมพิวเตอร์แบบฝังตัว (Embedded computer) ทำหน้าที่ในการประมวลผลการเปรียบเทียบความคล้ายกันของเสียง การให้คะแนนผลการเปรียบเทียบ การแสดงผลคะแนนการเปรียบเทียบ และการส่งสัญญาณควบคุมการเปลี่ยนความถี่ของเครื่องรับวิทยุที่รับคลื่นวิทยุที่กำลังออกอากาศ

ส่วนที่ 4 ส่วนแสดงผล (Display) ทำหน้าที่แสดงผลการเปรียบเทียบและคะแนนการเปรียบเทียบให้แก่ผู้ใช้งาน



รูปที่ 3.14 ขั้นตอนและอุปกรณ์ในการค้นหาสถานีที่เป็นต้นกำเนิดของสัญญาณรบกวน

3.4.1 หลักการทำงาน

ในระบบการค้นหาค้นหาความถี่ที่เป็นต้นกำเนิดของสัญญาณรบกวนแบบอัตโนมัติ มีขั้นตอนการทำงานดังรูปที่ 3.15 และรายละเอียดในแต่ละขั้นตอนดังนี้

ขั้นตอนที่ 1 บันทึกสัญญาณเสียงจากเครื่องรับวิทยุทั้งสองเครื่อง โดยกำหนดให้สัญญาณเสียงที่ได้จากคลื่นรบกวนที่เกิดขึ้นให้เป็นสัญญาณเสียงผสม (Mix audio : $M(t)$) และสัญญาณเสียงที่ได้จากสถานีวิทยุที่กำลังออกอากาศเป็นสัญญาณเสียงเดี่ยว (Single audio : $S(t)$) และแปลงสัญญาณเสียงทั้งสองให้เป็นสัญญาณดิจิทัล

ขั้นตอนที่ 2 ทำการเปรียบเทียบความคล้ายกันของเสียงตามกระบวนการของอัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียงที่ทำงานแบบเวลาจริงที่ได้เสนอในเนื้อหาข้างต้น

ขั้นตอนที่ 3 ระบบจะทำการตรวจสอบผลของการเปรียบเทียบที่ได้เพื่อทำการให้คะแนนของการเปรียบเทียบ โดยที่ถ้าหากพบว่ามีค่าความคล้ายกันของเสียง ระบบจะทำการเพิ่มคะแนนของการเปรียบเทียบ แต่ถ้าหากว่าไม่มีความคล้ายกันของเสียง ระบบจะทำการลดคะแนนของการเปรียบเทียบ

ขั้นตอนที่ 4 ระบบจะทำการตรวจสอบคะแนนของการเปรียบเทียบ เพื่อตัดสินใจว่าควรจะเปลี่ยนคลื่นความถี่ของสถานีวิทยุที่กำลังออกอากาศต่อไปหรือไม่ โดยถ้าหากคะแนนมีค่ามากกว่า 0 ระบบจะทำการแสดงคะแนนที่ได้ในส่วนแสดงผล และเริ่มการทำงานในขั้นตอนที่ 1 อีกครั้งโดยอัตโนมัติ และตรงกันข้ามถ้าคะแนนน้อยกว่าหรือเท่ากับ 0 ระบบจะส่งสัญญาณเพื่อให้เครื่องรับวิทยุรับคลื่นความถี่ของสถานีวิทยุที่กำลังออกอากาศในช่องถัดไป จากนั้นทำการเริ่มนับคะแนนใหม่ และเริ่มการทำงานในขั้นตอนที่ 1 ใหม่อีกครั้ง

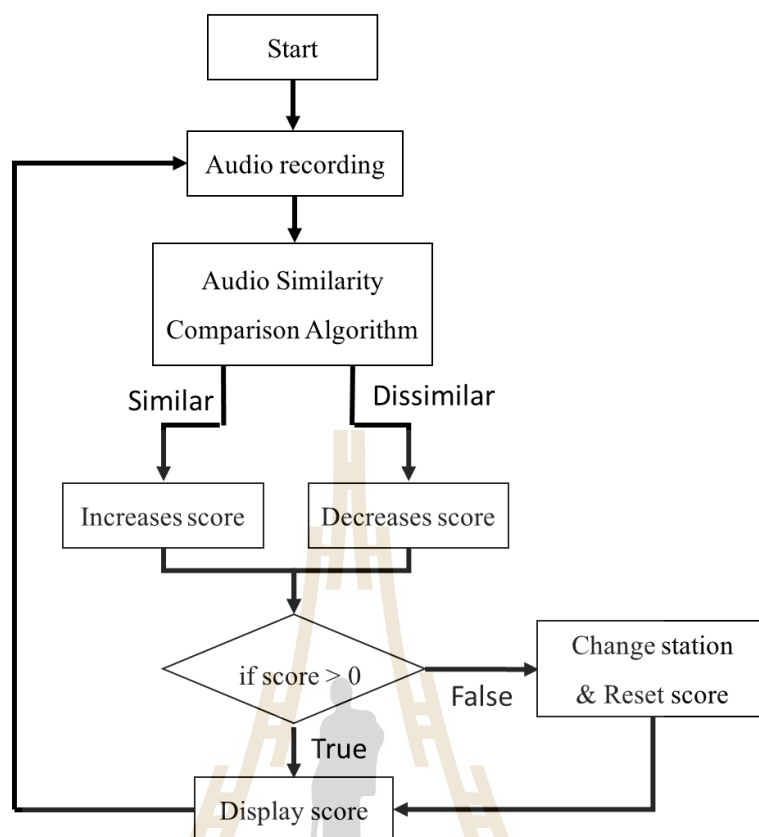
3.4.2 ชุดอุปกรณ์ต้นแบบ

ในส่วนนี้จะกล่าวถึงชุดอุปกรณ์ต้นแบบที่ใช้งานจริงในการค้นหาสถานีที่เป็นต้นกำเนิดของสัญญาณรบกวนแบบอัตโนมัติ ซึ่งประกอบไปด้วยอุปกรณ์ดังนี้

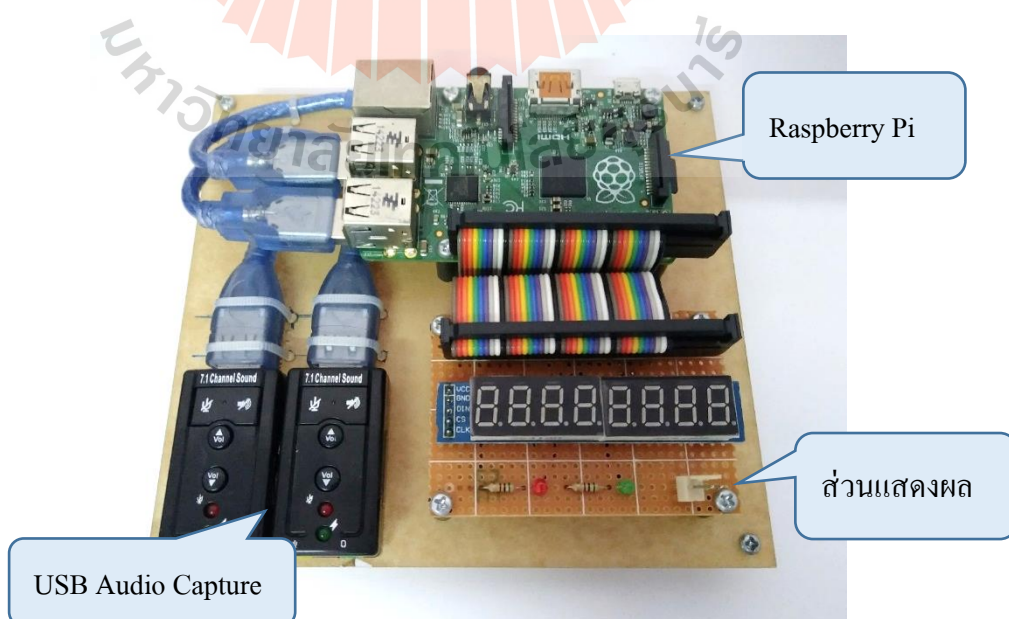
3.4.2.1 บอร์ด Raspberry Pi เป็นหน่วยประมวลผลหลักและทำหน้าที่ในการสั่งการบันทึกเสียง ประมวลผลการเปรียบเทียบความคล้ายกันของเสียง การให้คะแนนและแสดงผลของการเปรียบเทียบ และควบคุมการเปลี่ยนคลื่นความถี่ของเครื่องรับวิทยุ

3.4.2.2 USB Audio Capture ทำหน้าที่ในการบันทึกสัญญาณเสียงที่ส่งมาจากเครื่องรับวิทยุทั้งสองเครื่อง และแปลงสัญญาณเสียงอนาล็อกให้เป็นข้อมูลดิจิทัล ซึ่งถูกควบคุมการทำงานการด้วยบอร์ด Raspberry Pi ผ่านพอร์ต USB

3.4.2.3 ส่วนแสดงผล ทำหน้าที่แสดงผลให้ผู้ใช้งานสามารถเข้าถึงคะแนนและผลการตัดสินใจความคล้ายกันของเสียง ไฟสถานะการทำงานของชุดอุปกรณ์ต้นแบบ และมีพอร์ตเชื่อมต่อเพื่อควบคุมการเปลี่ยนคลื่นความถี่ของเครื่องรับวิทยุ ซึ่งถูกควบคุมการทำงานการด้วยบอร์ด Raspberry Pi ผ่านพอร์ต GPIO



รูปที่ 3.15 แผนผังการทำงานของระบบการค้นหาค้นหาคลื่นความถี่ที่เป็นต้นกำเนิดของสัญญาณรบกวนแบบอัตโนมัติ



รูปที่ 3.16 ชุดอุปกรณ์ต้นแบบ

3.5 สรุป

เนื้อหาในบทนี้ได้อธิบายแนวคิดและขั้นตอนการทำงานของอัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียงที่ทำงานแบบเวลาจริง โดยที่การทำงานของอัลกอริทึมจะแบ่งออกเป็น 3 กระบวนการย่อย ซึ่งในกระบวนการเตรียมข้อมูล ทำการเตรียมข้อมูลของทั้งสองสัญญาณเสียงที่จะเปรียบเทียบกันให้พร้อมสำหรับการคำนวณในกระบวนการถัดไป โดยเริ่มขั้นตอนการทำงานโดยการซัดตัวอย่างสัญญาณเสียงให้เป็นสัญญาณที่ไม่ต่อเนื่อง เพื่อให้สามารถนำข้อมูลของสัญญาณเสียงมาคำนวณทางคอมพิวเตอร์ได้ จากนั้นทำนอร์มัลไลเซชันข้อมูลของสัญญาณเสียง เพื่อให้สัญญาณเสียงที่จะทำการเปรียบเทียบมีแอมพลิจูดที่ใกล้เคียงกัน และแบ่งการพิจารณาข้อมูลของสัญญาณเสียงเป็น 3 ช่วงของความถี่ เพื่อเพิ่มความแม่นยำในการตัดสินใจความคล้ายกันของเสียง ในขั้นตอนสุดท้ายจะทำการเฉลี่ยข้อมูลของสัญญาณเสียงเป็นอันสิ้นสุดกระบวนการเตรียมข้อมูล ถัดจากนั้นเป็นกระบวนการคำนวณผลต่างของเสียง ซึ่งจะนำข้อมูลของทั้งสองสัญญาณเสียงมาทำการคำนวณผลต่างของเสียง และทำการหาค่าคงที่ที่เรียกว่าค่าความคล้ายกันของเสียง (I_x) ของแต่ละย่านความถี่ เพื่อใช้ในกระบวนการตัดสินใจความคล้าย ซึ่งเป็นกระบวนการสุดท้ายของอัลกอริทึม ซึ่งจะนำค่าความคล้ายกันของเสียง (I_x) มาพิจารณากับค่าเทรชโฮลด์ (threshold) ของแต่ละย่านความถี่ จากนั้นรวมผลการพิจารณาจากทั้ง 3 ย่านความถี่มาทำการโหวต 2 ใน 3 เสียง เพื่อตัดสินใจว่าทั้งสองสัญญาณเสียงมีความคล้ายกันหรือไม่ และนอกจากนั้นในบทนี้ได้เสนอชุดอุปกรณ์ต้นแบบในการค้นหาสถานีที่เป็นต้นกำเนิดของสัญญาณรบกวนที่นำอัลกอริทึมที่พัฒนามาใช้งาน โดยอุปกรณ์สามารถที่หาได้ง่ายตามท้องตลาด ซึ่งมีขั้นตอนการทำงาน โดยจะบันทึกเสียงจากเครื่องรับวิทยุทั้งสองที่รับคลื่นจากสัญญาณรบกวนและคลื่นจากสถานีที่กำลังออกอากาศ มาทำการเปรียบเทียบความคล้ายกันของเสียงตามกระบวนการทำงานของอัลกอริทึม ถ้าพบว่ามี ความคล้ายกันของเสียง ระบบจะให้คะแนนของการเปรียบเทียบและทำงานใหม่อีกครั้ง แต่ถ้าพบว่ามี ความคล้ายกันของเสียง ระบบจะหักคะแนนของการเปรียบเทียบจนกระทั่งไม่เหลือคะแนนการเปรียบเทียบให้หักแล้ว จากนั้นจะส่งสัญญาณไปยังเครื่องรับวิทยุให้รับคลื่นจากสถานีวิทยุถัดไป และทำงานใหม่อีกครั้ง ซึ่งทำให้การค้นหาสถานีที่เป็นต้นกำเนิดของสัญญาณรบกวนเป็นไปอย่างอัตโนมัติ

บทที่ 4

การทดลองและวิเคราะห์ของ การทำงานบนโปรแกรมคอมพิวเตอร์

4.1 บทนำ

เนื้อหาในบทนี้แสดงผลการทดลองและการวิเคราะห์ผลของการจำลองการทำงานของอัลกอริทึมการเปรียบเทียบความคล้ายกันของเสียงที่ทำงานแบบเวลาจริงบนโปรแกรมคอมพิวเตอร์ โดยที่ในการทดลองได้ทำการเปรียบเทียบความคล้ายกันของตัวอย่างเสียงที่ทราบว่าจะมีความคล้ายกันของเสียงและไม่มี ความคล้ายกันของเสียงอยู่ก่อนแล้ว เพื่อหาค่าเทรซโวลต์ที่เหมาะสมในการตัดสินผลการเปรียบเทียบความคล้ายกันของเสียง และทดสอบประสิทธิภาพการทำงานของอัลกอริทึมในการตัดสินความคล้ายกันของเสียง

4.2 การจำลองการทำงานบนโปรแกรมคอมพิวเตอร์

ในงานวิจัยนี้ได้ทำการพัฒนาอัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียงที่ทำงานแบบเวลาจริง บนโปรแกรม MATLAB และทดสอบการทำงานของอัลกอริทึม โดยมีหลักการทำงานและการเตรียมการดังนี้

4.2.1 ขั้นตอนการทำงานและพารามิเตอร์ที่ใช้งาน

ในการจำลองการทำงานของอัลกอริทึมที่พัฒนาขึ้นบนโปรแกรม MATLAB มีขั้นตอนการทำงานที่ยึดตามหัวข้อ 3.3 หลักการเปรียบเทียบความคล้ายกันของเสียง เพื่อทำเปรียบเทียบความคล้ายกันของเสียงเดี่ยวและเสียงผสมโดยที่จะมีกระบวนการทำงานและกำหนดพารามิเตอร์ดังนี้

4.2.1.1 กระบวนการเตรียมข้อมูล (Data preparing) จะทำการเตรียมข้อมูลให้พร้อมสำหรับกระบวนการถัดไป ซึ่งในกระบวนการนี้มีขั้นตอนการทำงานย่อยดังนี้

ขั้นตอนซีกตัวอย่าง จะทำการซีกตัวอย่าง (sampling) สัญญาณต่อเนื่อง (continuous signal) เพื่อให้เป็นสัญญาณไม่ต่อเนื่อง (discrete signal) ด้วยอัตราการซีกตัวอย่าง (sample rate : F_s) เท่ากับ 44,100 Hz อัตราบิตต่อสัญลักษณ์ (bit per symbol) เท่ากับ 16 บิต และการเปรียบเทียบในแต่ละครั้งจะพิจารณาสัญญาณเสียงที่มีความยาว 10 วินาที

ขั้นตอนนอร์มัลไลเซชัน (normalization) จะนอร์มัลไลซ์ข้อมูลของทั้งสองสัญญาณเสียง เพื่อให้ข้อมูลของสัญญาณเสียงมีความแรงที่ใกล้เคียงกัน

ขั้นตอนกรองสัญญาณ (filter) จะทำการแบ่งข้อมูลเป็น 3 ย่านความถี่ตามที่ได้ ออกแบบไว้ โดยที่ย่านความถี่ต่ำจะกรองข้อมูลในช่วงความถี่ 0–340 Hz ย่านความถี่กลางจะกรองข้อมูลในช่วงความถี่ 340–3,400 Hz และย่านความถี่ต่ำจะกรองข้อมูลในช่วงความถี่ 3,400–20,000 Hz

ขั้นตอนเฉลี่ยข้อมูล (average) จะทำการเฉลี่ยข้อมูลของสัญญาณเสียง โดยมีหน้าต่างรับข้อมูล (windows size : w_s) เท่ากับ 1 มิลลิวินาที เพื่อให้การเปรียบเทียบสามารถทนทานต่อการคลาดเคลื่อนทางเวลาที่อาจเกิดขึ้น

4.2.1.2 กระบวนการคำนวณผลต่างของเสียง (audio subtraction) จะนำข้อมูลของสัญญาณเสียงเดี่ยว และสัญญาณเสียงผสมที่ผ่านกระบวนการเตรียมข้อมูลแล้ว มาคำนวณเพื่อหาค่าความคล้ายกันของเสียง (I_x) ของแต่ละย่านความถี่

4.2.1.3 กระบวนการตัดสินใจความคล้าย (similarly decision) จะนำค่าความคล้ายกันของเสียง (I_x) มาพิจารณากับค่าเทรชโฮลด์ (threshold) ของแต่ละย่านความถี่ ถ้าหากค่าความคล้ายกันของเสียงมีค่าน้อยกว่าค่าเทรชโฮลด์จะตัดสินใจว่าย่านความถี่นั้นมีความคล้ายกันของเสียง และทำการรวมผลพิจารณาจากทั้ง 3 ย่านความถี่มาทำการโหวต 2 ใน 3 เสียง เพื่อตัดสินใจว่าทั้งสองสัญญาณเสียงมีความคล้ายกันหรือไม่ ถ้าหากพบว่ามีค่าความคล้ายกันของเสียงอยู่มากกว่าหรือเท่ากับ 2 เสียง จะให้ผลการตัดสินใจว่าทั้งสองสัญญาณเสียงมีความคล้ายกัน

4.2.2 เตรียมตัวอย่างการทดลอง

ผู้วิจัยได้ทำการเตรียมตัวอย่างเสียงเพื่อใช้ในการทดลองการเปรียบเทียบความคล้ายกันของเสียง โดยที่ตัวอย่างเสียงที่ต้องการจะแบ่งได้เป็น 2 กลุ่มคือ กลุ่มเสียงเดี่ยว เพื่อจำลองเป็นเสียงที่ได้จากเครื่องรับวิทยุที่รับคลื่นจากสถานีวิทยุที่กำลังออกอากาศ ซึ่งไม่มีการผสมเสียงจากสถานีวิทยุอื่น และกลุ่มเสียงผสม เพื่อจำลองเป็นเสียงที่ได้จากเครื่องรับวิทยุที่รับคลื่นความถี่รบกวนที่เกิดขึ้น โดยที่ไฟล์เสียงตัวอย่างจะถูกบันทึกในรูปแบบ “.mp3 ” และมีความยาวขนาด 10 วินาที ซึ่งไฟล์เสียงตัวอย่างในกลุ่มเสียงเดี่ยวมีจำนวนทั้งสิ้น 20 ไฟล์เสียง ซึ่งเนื้อหาของเสียงจะแบ่งประเภทเป็น เสียงเพลงที่ผู้ชายเป็นผู้ร้อง เสียงเพลงที่ผู้หญิงเป็นผู้ร้อง เสียงเพลงช้า และ เสียงเพลงเร็ว และกลุ่มเสียงผสม ได้จากการผสมกันของกลุ่มเสียงเดี่ยว 2 ไฟล์เสียงเข้าด้วยกัน ซึ่งจะมีจำนวนทั้งสิ้น 190 ไฟล์เสียง

4.3 การทดลองเพื่อหาค่าเทรชโฮลด์ที่เหมาะสม

จากเนื้อหาหน้าในหัวข้อ 3.3 ได้กล่าวถึงค่าเทรชโฮลด์ (threshold) ของแต่ละย่านความถี่ ซึ่งเป็นพารามิเตอร์สำคัญในการตัดสินใจความคล้ายกันของเสียง โดยที่ค่าเทรชโฮลด์นี้จะได้จากทดลองของการเปรียบเทียบความคล้ายกันของเสียง เก็บผลการทดลอง และพิจารณาค่าความคล้ายกันของเสียง (I_x) ที่ได้จากทดลอง

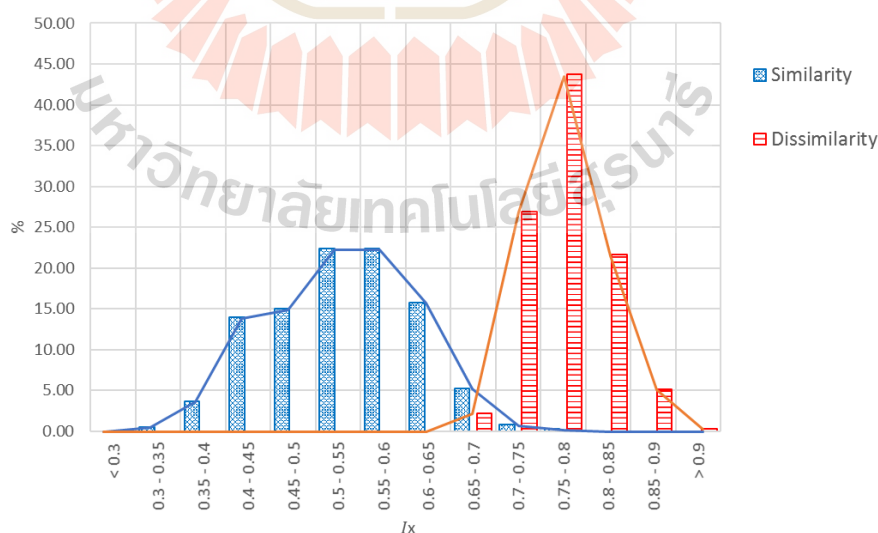
ในการทดลองนี้จะทำการเปรียบเทียบความคล้ายกันของไฟล์เสียงที่เตรียมไว้ โดยการเปรียบเทียบจะเกิดจากการจับคู่กันของกลุ่มเสียงเดี่ยว และกลุ่มเสียงผสม ซึ่งในการจับคู่นี้ 1 ไฟล์เสียงจากกลุ่มเสียงเดี่ยว จะเปรียบเทียบกับกลุ่มเสียงผสมที่มีตัวเองเป็นส่วนผสมอยู่เป็นจำนวน 19 ครั้ง ซึ่งถือว่าการเปรียบเทียบของเสียงที่มีความคล้ายกัน และจะเปรียบเทียบกับกลุ่มเสียงผสมที่ไม่มีตัวเองอยู่เป็นจำนวน 171 ครั้ง ซึ่งถือ

ว่าเป็นการเปรียบเทียบของเสียงที่ไม่มีความคล้ายกัน ดังนั้นจากกลุ่มเสียงเดี่ยวจำนวน 20 ไฟล์เสียง จะทำให้เกิดการเปรียบเทียบของเสียงที่มีความคล้ายกันทั้งสิ้น 380 ครั้ง และมีการเปรียบเทียบของเสียงที่ไม่มีความคล้ายกันทั้งสิ้น 3,420 ครั้ง

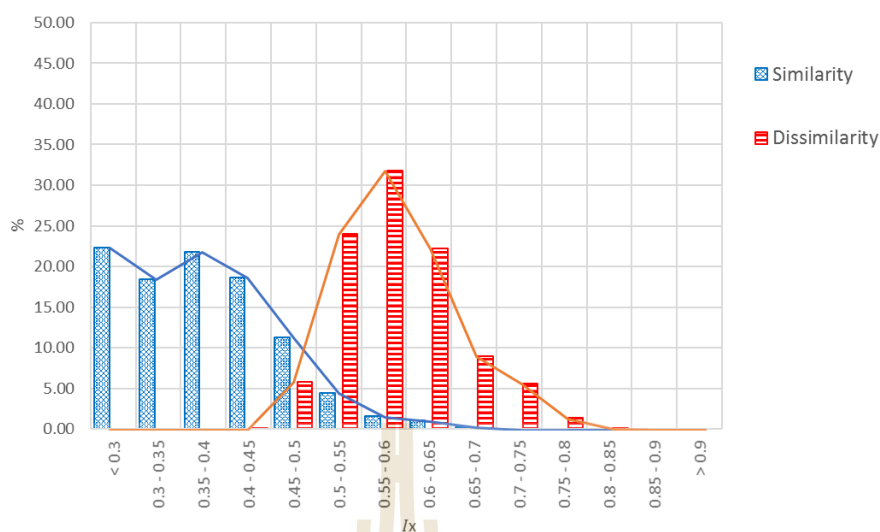
4.3.1 ผลการทดลอง

ผลการทดลองที่ได้จะแสดงในรูปแบบกราฟแสดงความถี่ของข้อมูลที่เกิดขึ้น (histogram) เพื่อแสดงความถี่ของค่าความคล้ายกันของเสียง (I_x) ที่เกิดขึ้นว่าอยู่ในช่วงใด โดยที่จะแบ่งผลการทดลองเป็น 2 กลุ่มคือ กลุ่มผลการเปรียบเทียบของเสียงที่มีความคล้ายกัน (similarity) และกลุ่มผลการเปรียบเทียบของเสียงที่ไม่มีความคล้ายกัน (dissimilarity) และแสดงผลการทดลองโดยแบ่งตามย่านความถี่ดังรูปที่ 4.1–4.3 เนื่องจากทั้ง 2 กลุ่มมีความต่างกันของจำนวนข้อมูลเป็นจำนวนมากจึงแสดงในรูปของร้อยละในการพบข้อมูล

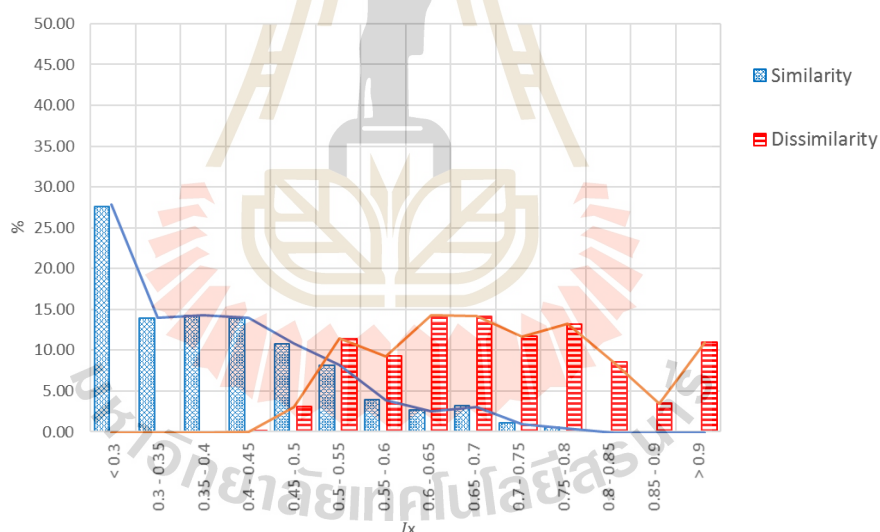
จากรูปที่ 4.1 ทำให้ทราบถึงแนวโน้มของค่า I_x โดยที่ในกลุ่มผลการเปรียบเทียบของเสียงที่คล้ายกันมีค่า I_x อยู่ในช่วง 0.4-0.7 ซึ่งต่ำกว่ากลุ่มผลการเปรียบเทียบของเสียงที่ไม่มีความคล้ายกันที่มีค่า I_x อยู่ในช่วง 0.65-0.9 เนื่องจากค่า I_x ของทั้งสองกลุ่มมีช่วงของค่า I_x ที่ความต่างกัน จึงเลือกค่าเทรสโพลด์ที่เหมาะสมจากส่วนที่มีค่า I_x ซ้อนทับกันอยู่จากทั้งสองกลุ่มผลการเปรียบเทียบ ดังนั้นจึงเลือกพิจารณาค่าเทรสโพลด์ที่เหมาะสมในการตัดสินใจผลการเปรียบเทียบในย่านความถี่ต่ำจากช่วง 0.65-0.75 และเช่นเดียวกันจากรูปที่ 4.2 และ 4.3 จึงเลือกพิจารณาค่าเทรสโพลด์ที่เหมาะสมในย่านความถี่กลางและย่านความถี่สูงจากช่วง 0.45-0.55



รูปที่ 4.1 กราฟความถี่ของค่าความคล้ายกันของเสียง (I_x) ที่เกิดขึ้นในย่านความถี่ต่ำ



รูปที่ 4.2 กราฟความถี่ของค่าความคล้ายกันของเสียง (I_x) ที่เกิดขึ้นในย่านความถี่กลาง



รูปที่ 4.3 กราฟความถี่ของค่าความคล้ายกันของเสียง (I_x) ที่เกิดขึ้นในย่านความถี่สูง

เพื่อให้ทราบอย่างแน่ชัดว่าค่าเทรสโพลด์ใดเหมาะสมที่จะไปใช้ในการตัดสินใจตัดสินความคล้ายกันของเสียง จึงนำค่าเทรสโพลด์ในช่วง 0.45-0.75 มาใช้ในการตัดสินใจตัดสินความคล้ายกันของเสียงและบันทึกผลการทดลอง ซึ่งผลของการทดลองที่ได้จะแบ่งออกเป็น 3 ส่วน ส่วนที่ 1 ความถูกต้องเชิงบวก (right positive) คือผลการตัดสินใจว่ามีคล้ายกันของเสียงจากกลุ่มเสียงที่มีความคล้ายกัน ส่วนที่ 2 ความผิดพลาดเชิงบวก (false positive) คือผลการตัดสินใจว่ามีคล้ายกันของเสียงจากกลุ่มเสียงที่ไม่มีความคล้ายกัน และเมื่อเลือกใช้

ค่าเทรสโฮลด์ที่มีค่าสูงขึ้น ค่าความถูกต้องเชิงบวกในแต่ละย่านความถี่ให้ค่าที่สูงขึ้น แต่ขณะเดียวกันค่าความผิดพลาดเชิงบวกในย่านความถี่นั้น ๆ ก็สูงขึ้นด้วยเช่นกัน เพื่อหาจุดสมดุลระหว่างค่าความถูกต้องเชิงบวกและค่าความผิดพลาดเชิงบวก จึงให้คะแนนความแม่นยำ (accuracy score) ในส่วนที่ 3 โดยหาได้จากผลต่างของค่าความถูกต้องเชิงบวกลบด้วยค่าความผิดพลาดเชิงบวกแสดงในตารางที่ 4.1

ตารางที่ 4.1 ตารางผลการใช้ค่าเทรสโฮลด์ต่าง ๆ ในการตัดสินความคล้ายกันของเสียง

Threshold values	Right positive (%)			False positive (%)			Accuracy score		
	Low band	Mid band	High band	Low band	Mid band	High band	Low band	Mid band	High band
0.45	18.16	81.32	69.74	0.00	0.03	0.06	18.16	81.29	69.68
0.5	33.16	92.63	80.53	0.00	5.88	3.13	33.16	86.75	77.40
0.55	55.53	97.11	88.68	0.00	29.88	14.53	55.53	67.22	74.15
0.6	77.89	98.68	92.63	0.00	61.64	23.77	77.89	37.05	68.86
0.65	93.68	99.74	95.26	0.00	83.86	38.04	93.68	15.88	57.22
0.7	98.95	100.00	98.42	2.22	92.87	52.19	96.73	7.13	46.23
0.75	99.74	100.00	99.47	29.18	98.51	63.86	70.56	1.49	35.61

จากตารางที่ 4.1 แสดงให้เห็นถึงค่าเทรสโฮลด์ที่เหมาะสมที่จะนำไปใช้ในการตัดสินความคล้ายกันของเสียง โดยพิจารณาจากค่าเทรสโฮลด์ที่ให้คะแนนความแม่นยำของการตัดสินที่มากที่สุด ดังนั้นค่าเทรสโฮลด์ที่เหมาะสมในย่านความถี่ต่ำคือ 0.7 ซึ่งได้คะแนนความแม่นยำ 96.73 คะแนน และค่าเทรสโฮลด์ที่เหมาะสมในย่านความถี่กลางและย่านความถี่สูงคือ 0.5 ซึ่งได้คะแนนความแม่นยำ 86.75 และ 77.40 คะแนนตามลำดับ

4.4 การทดสอบประสิทธิภาพการทำงานของอัลกอริทึม

4.4.1 การตัดสินใจความคล้ายกันของเสียง

หลังจากได้ค่าเทรชโฮลด์ที่เหมาะสมในการตัดสินใจความคล้ายกันของเสียงของแต่ละย่านความถี่แล้ว จากกระบวนการตัดสินใจความคล้ายในหัวข้อ 3.3.3 ได้กล่าวถึงการตัดสินใจความคล้ายกันของเสียง โดยการนำค่าความคล้ายกันของเสียง (I_x) มาพิจารณากับค่าเทรชโฮลด์(threshold) ของแต่ละย่านความถี่ จากนั้นรวมผลการพิจารณาจากทั้ง 3 ย่านความถี่และทำการโหวต 2 ใน 3 เสียง เพื่อตัดสินใจว่าทั้งสองสัญญาณเสียงมีความคล้ายกันหรือไม่ ถ้าหากพบว่ามีความคล้ายกันของเสียง (I_x) น้อยกว่าค่าเทรชโฮลด์ใน 2 ย่านความถี่หรือมากกว่า จะตัดสินใจว่าทั้งสองสัญญาณเสียงมีความคล้ายกัน

ในการทดลองนี้ได้ทดลองการใช้เสียงโหวตต่าง ๆ เพื่อทดสอบประสิทธิภาพในการโหวตเสียงข้างมาก และเนื่องจากขั้นตอนการโหวตเสียงข้างมากเป็นขั้นตอนสุดท้ายของอัลกอริทึม ดังนั้นผลการทดลองนี้จะบ่งชี้ถึงประสิทธิภาพการตัดสินใจความคล้ายกันของเสียงของอัลกอริทึมด้วยเช่นกัน โดยที่ผลการทดลองที่ได้จะแบ่งเป็น 3 ส่วนคือ ส่วนที่ 1 ความถูกต้องในการโหวต คือผลการเปรียบเทียบความคล้ายกันของกลุ่มเสียงที่มีเสียงคล้ายกัน และมีจำนวนที่พบว่าค่าความคล้ายกันของเสียง (I_x) น้อยกว่าค่าเทรชโฮลด์จาก 3 ย่านความถี่มีมากกว่าหรือเท่ากับเสียงโหวต ส่วนที่ 2 ความผิดพลาดในการโหวต คือผลการเปรียบเทียบความคล้ายกันของกลุ่มเสียงที่ไม่มีเสียงคล้ายกัน และมีจำนวนที่พบว่าค่าความคล้ายกันของเสียง (I_x) น้อยกว่าค่าเทรชโฮลด์จาก 3 ย่านความถี่มีน้อยกว่าเสียงโหวต และเนื่องจากเมื่อใช้เสียงโหวต 1 ใน 3 เสียง จะทำให้ได้ค่าความถูกต้องในการโหวตมีค่าสูง แต่ขณะเดียวกันค่าความผิดพลาดในการโหวตก็สูงเช่นกัน เพื่อหาจุดสมดุลระหว่างค่าความถูกต้องในการโหวตและค่าความผิดพลาดในการโหวต จึงให้คะแนนความแม่นยำในการโหวตในส่วนที่ 3 ซึ่งหาได้จากผลต่างของค่าความถูกต้องในการโหวตลบด้วยค่าความผิดพลาดในการโหวต ดังแสดงในตารางที่ 4.2

ตารางที่ 4.2 ตารางผลการโหวตเสียงข้างมากในการตัดสินใจความคล้ายกันของเสียง

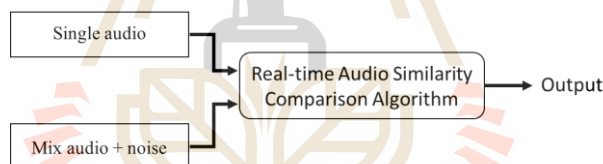
เสียงโหวต	ความถูกต้องในการโหวต (%)	ความผิดพลาดในการโหวต (%)	คะแนนความแม่นยำในการโหวต
1 ใน 3	99.21	10.85	88.36
2 ใน 3	93.42	0.38	93.04
3 ใน 3	79.47	0.00	79.47

จากตารางที่ 4.2 แสดงให้เห็นว่าการโหวต 2 ใน 3 เสียงที่ใช้งานในอัลกอริทึมได้ คะแนนความแม่นยำในการโหวตสูงที่สุดคือ 93.04 คะแนน ซึ่งถือว่าเป็นความแม่นยำในการตัดสินใจ คล้ายกันของเสียงของอัลกอริทึมเมื่อทดสอบการทำงานบนโปรแกรมคอมพิวเตอร์ด้วยเช่นกัน

4.4.2 ความทนต่อสัญญาณรบกวน

จากทฤษฎีอินเทอร์มอดูเลชัน ความถี่อินเทอร์มอดูเลชันที่เกิดขึ้นและไปรบกวนการสื่อสาร ระหว่างเครื่องบินและสถานีภาคพื้นนั้น มีกำลังที่ต่ำกว่าความถี่ตั้งต้นอย่างมาก ทำให้เมื่อเครื่องรับวิทยุรับ คลื่นความถี่อินเทอร์มอดูเลชันจะทำสัญญาณเสียงที่ได้มีคุณภาพต่ำ และมีสัญญาณรบกวน (noise) ผสมอยู่ ด้วย

ในการทดลองนี้จะเปรียบเทียบความคล้ายกันของเสียง โดยที่จะเพิ่มสัญญาณรบกวนใน กลุ่มเสียงผสมด้วยอัตราส่วนของสัญญาณต่อสัญญาณรบกวน (Signal to noise ratio: SNR) ในระดับต่าง ๆ และแสดงผลการทดลองเป็น 3 ส่วนคือส่วนที่ 1 ความถูกต้องในการตัดสินใจ คือผลการตัดสินใจที่มีความ คล้ายกันของกลุ่มเสียงที่มีความคล้ายกัน ส่วนที่ 2 ความผิดพลาดในการตัดสินใจ คือผลการตัดสินใจที่มีความ คล้ายกันของเสียงของกลุ่มที่ไม่มี ความคล้ายกันของเสียง และเพื่อหาจุดสมดุลระหว่างความถูกต้องในการ ตัดสินกับความผิดพลาดในการตัดสินใจ จึงให้คะแนนความแม่นยำในการตัดสินใจในส่วนที่ 3 โดยที่หาได้จาก ผลต่างของค่าความถูกต้องในการตัดสินใจลบด้วยค่าความผิดพลาดในการตัดสินใจแสดงดังตารางที่ 4.3



รูปที่ 4.4 การทดลองความทนต่อสัญญาณรบกวน

จากตารางที่ 4.3 แสดงให้เห็นว่ายิ่งระดับ SNR สูงขึ้นจะให้คะแนนความแม่นยำในการ ตัดสินสูงขึ้นเช่นกัน และที่ระดับ SNR มากกว่า 3 dB ซึ่งมีสัญญาณเสียงมากกว่าสัญญาณรบกวน (noise) มากกว่า 2 เท่า อัลกอริทึมมีคะแนนความแม่นยำในการตัดสินใจที่มากกว่า 70 คะแนนซึ่งถือว่าไม่ส่งผลกระทบต่อ การทำงานของอัลกอริทึมมากนัก แต่เมื่อระดับ SNR ที่ 0 dB ซึ่งมีสัญญาณเสียงเท่ากับหรือน้อยกว่า สัญญาณรบกวน ทำให้ความแม่นยำในการตัดสินใจอยู่ที่ 39.47 คะแนนซึ่งต่ำกว่าจุดที่ยอมรับได้ จึงถือว่าที่ ระดับ SNR ต่ำกว่า 0 dB ส่งผลกระทบต่อการทำงานของอัลกอริทึมเป็นอย่างมาก

ตารางที่ 4.3 ตารางผลการทดลองเพื่อหาความทนต่อสัญญาณรบกวน

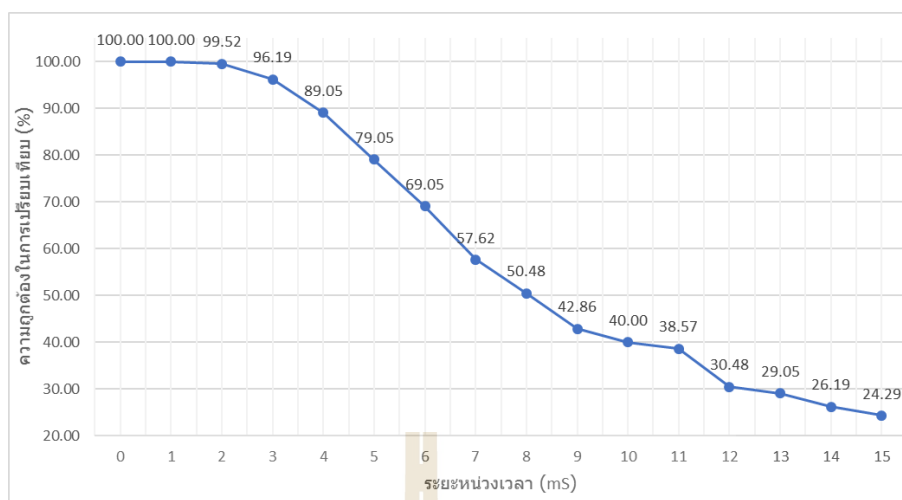
SNR (dB)	ความถูกต้องในตัดสีน (%)	ความผิดพลาดในตัดสีน (%)	คะแนนความแม่นยำในตัดสีน
-3	0.00	0.00	0.00
0	39.47	0.00	39.47
3	77.37	0.00	77.37
6	88.42	0.26	88.16
9	91.05	0.64	90.41
12	92.37	2.75	89.62

4.4.3 ความทนต่อการคลาดเคลื่อนทางเวลา

จากการออกแบบการนำอัลกอริทึมไปใช้งานบนชุดอุปกรณ์จริงในหัวข้อที่ 3.4 ซึ่งในขั้นตอนการบันทึกเสียงจากทั้ง 2 เครื่องรับวิทยุเพื่อนำมาเปรียบเทียบความคล้ายกันของเสียงนั้น อาจเกิดการคลาดเคลื่อนทางเวลาในบันทึกเสียงขึ้นได้ โดยที่การเปรียบเทียบความคล้ายกันของเสียงในงานวิจัยนี้ทำการเปรียบเทียบโดยการหาผลต่างของแอมพลิจูดในทางเวลา (time domain) ซึ่งเวลาที่คลาดเคลื่อนนั้นอาจส่งผลกระทบต่อตัดสีนผลการเปรียบเทียบของอัลกอริทึมได้ ดังนั้นจึงทดลองเพื่อหาความทนต่อการคลาดเคลื่อนทางเวลาที่เกิดขึ้น

ในการทดลองนี้จะทำการเปรียบเทียบความคล้ายกันของเสียงกับเสียงตัวเอง (ทราบอยู่แล้วว่ามีความคล้ายกันของเสียง) จำนวน 210 ไฟล์เสียง ซึ่งไฟล์เสียงที่ทำการเปรียบเทียบกันนั้นจะมีเวลาเริ่มต้นที่ต่างกันในช่วง 0–15 มิลลิวินาที และแสดงผลการทดลองดังรูปที่ 4.4 ซึ่งแสดงค่าความถูกต้องในการเปรียบเทียบ คือพบว่ามีค่าความคล้ายกันของเสียงจากการเปรียบเทียบกับเสียงตัวเอง และแสดงผลในรูปแบบร้อยละในการพบว่ามีค่าความคล้ายกันของเสียง

จากรูปที่ 4.4 แสดงให้เห็นว่ายิ่งมีความคลาดเคลื่อนทางเวลาที่มากขึ้นจะทำให้ความถูกต้องในการเปรียบเทียบต่ำลง และที่ความคลาดเคลื่อนทางเวลา 5 มิลลิวินาทีพบว่ามีค่าความถูกต้องในการตัดสีน 79.05% ซึ่งอยู่ในระดับที่ยอมรับได้ ดังนั้นถ้าในขั้นตอนการบันทึกเสียงจากทั้ง 2 เครื่องรับวิทยุเกิดความคลาดเคลื่อนทางเวลาที่ต่ำกว่า 5 มิลลิวินาที จะถือว่าไม่ส่งผลกระทบต่อการทำงานของอัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียงที่ทำงานแบบเวลาจริง



รูปที่ 4.5 กราฟความถูกต้องในการเปรียบเทียบของแต่ละระยะเวลาหน่วย

4.4.4 เวลาที่ใช้ในการเปรียบเทียบ

ในการทดลองที่ผ่านมา นอกกว่าบันทึกผลของการทดลองนั้น ๆ แล้ว ผู้วิจัยยังได้บันทึกเวลาที่ใช้ในการทดสอบการเปรียบเทียบความคล้ายกันของเสียงด้วย ซึ่งผลการทดสอบเวลาการทำงานของฟังก์ชันหลักและฟังก์ชันย่อยต่าง ๆ แสดงดังตารางที่ 4.3 โดยที่ในการเปรียบเทียบแต่ละครั้งจะใช้เวลา 3 วินาทีโดยเฉลี่ย ซึ่งในการจำลองการทำงานบนโปรแกรมคอมพิวเตอร์นี้เป็นทดลองกระบวนการเปรียบเทียบเพียงอย่างเดียว เมื่อนำอัลกอริทึมไปใช้งานจริงจำเป็นต้องเพิ่มกระบวนการอื่น ๆ เช่นการบันทึกเสียงจากภายนอก และการควบคุมการทำงานของอุปกรณ์ ซึ่งจะทำให้ใช้เวลาในการเปรียบเทียบที่มากขึ้น

4.5 สรุป

บทนี้ได้จำลองการทำงานอัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียงที่ทำงานแบบเวลาจริง บนโปรแกรม MATLAB เพื่อหาค่าเทรซโฮลด์ที่เหมาะสมที่ใช้ในการตัดสินความคล้ายกันของเสียง ที่ได้เตรียมตัวอย่างที่จะใช้ในการทดลองการเปรียบเทียบความคล้ายกันของเสียงแบ่งเป็น 2 กลุ่มคือ กลุ่มเสียงเดี่ยว เพื่อจำลองเป็นเสียงที่ได้จากเครื่องรับวิทยุที่รับคลื่นจากสถานีวิทยุที่กำลังออกอากาศ และกลุ่มเสียงผสม เพื่อจำลองเป็นเสียงที่ได้จากเครื่องรับวิทยุที่รับคลื่นความถี่รบกวนที่เกิดขึ้น และนำเอาไฟล์เสียงตัวอย่างมาทดลองการเปรียบเทียบความคล้ายกันของเสียงตามขั้นตอนการทำงานของอัลกอริทึมในกระบวนการเตรียมข้อมูลและกระบวนการคำนวณผลต่างของเสียงดังที่ได้นำเสนอในบทที่ 3 เพื่อหาค่าความคล้ายกันของเสียง (I_x) จากนั้นทำการบันทึกและวิเคราะห์ผลการทดลอง เพื่อหาค่าเทรซโฮลด์ที่เหมาะสมที่จะนำมาใช้ในกระบวนการตัดสินความคล้ายกัน จากผลการวิเคราะห์ทำให้ทราบว่าแนวโน้มของค่าความคล้ายกันของเสียง

จากกลุ่มการเปรียบเทียบของเสียงที่มีความคล้ายกันอยู่ต่ำกว่ากลุ่มการเปรียบเทียบของเสียงที่ไม่มีความคล้ายกัน ซึ่งเป็นไปแนวคิดของอัลกอริทึมที่ว่าผลต่างของเสียงที่มีความคล้ายกันจะมีผลเข้าใกล้ 0 และผลการทดลองที่ได้จะมีช่วงที่มีผลการทดลองจากทั้งสองกลุ่มซ้อนทับกัน จึงเลือกพิจารณาค่าเทรสโพลด์ที่เหมาะสมจากช่วงที่มีการซ้อนทับกัน โดยการนำค่าเทรสโพลด์ในช่วง 0.45-0.75 มาใช้ในการตัดสินความคล้ายกันของเสียง และพิจารณาค่าเทรสโพลด์ที่เหมาะสมจากคะแนนความแม่นยำที่มากที่สุด ซึ่งคะแนนความแม่นยำนี้คือจุดสมมูลระหว่างค่าความถูกต้องเชิงบวกและค่าความผิดพลาดเชิงบวก ที่หาได้จากผลต่างของค่าความถูกต้องเชิงบวกลบด้วยค่าความผิดพลาดเชิงบวก จากการคะแนนความแม่นยำที่มากที่สุดในย่านความถี่ต่ำ กลาง และสูงคือ 0.7 0.5 และ 0.5 ซึ่งมีคะแนนความแม่นยำ 96.73 86.75 และ 77.40 คะแนนตามลำดับ

นอกจากนี้ได้ทำการทดสอบประสิทธิภาพในการตัดสินความคล้ายกันของเสียงเมื่อรวมผลพิจารณาจากทั้ง 3 ย่านความถี่มาโหวต 2 ใน 3 เสียง มีคะแนนความแม่นยำในการโหวตที่ 97.04 คะแนน ซึ่งคะแนนความแม่นยำในการโหวตคือจุดสมมูลของค่าความถูกต้องในการโหวตและค่าความผิดพลาดในการโหวต หาได้จากผลต่างของความถูกต้องในโหวตลบด้วยความผิดพลาดในโหวต และเนื่องจากการโหวตเสียงข้างนี้เป็นขั้นตอนสุดท้ายในการตัดสินความคล้ายกันของเสียงจึงถือว่าคะแนนความแม่นยำในการโหวตเป็นความแม่นยำในการตัดสินความคล้ายกันของเสียงเช่นกัน และในการทดสอบความทนต่อสัญญาณรบกวนเมื่อนำอัลกอริทึมไปใช้งานจริง โดยการเพิ่มสัญญาณรบกวน (noise) ในกลุ่มเสียงผสมและทำการเปรียบเทียบความคล้ายกันของเสียง ผลที่ได้คือที่อัตราส่วนของสัญญาณต่อสัญญาณรบกวน (SNR) ต่ำกว่า 0 dB สัญญาณรบกวนจะมีผลกระทบต่ออัลกอริทึมที่พัฒนาเป็นอย่างมาก และในการทดลองเพื่อความทนต่อการคลาดเคลื่อนทางเวลาที่อาจเกิดขึ้นในขณะที่ทำงานบนชุดอุปกรณ์จริง ผลจากการทดลองทำให้ทราบว่า ที่ความคลาดเคลื่อนทางเวลาที่น้อยกว่า 5 มิลลิวินาที จะไม่ส่งผลกระทบต่อการทำงานของอัลกอริทึม นอกจากนี้ในการทดลองแต่ละครั้งได้ทำการบันทึกเวลาที่ใช้ในการเปรียบเทียบความคล้ายกันด้วย ซึ่งในการเปรียบเทียบแต่ละครั้งได้ใช้ 3 วินาทีโดยเฉลี่ย

บทที่ 5

การทดลองและวิเคราะห์ของ การทำงานบนชุดอุปกรณ์ต้นแบบ

5.1 บทนำ

เนื้อหาในบทนี้จะแสดงผลการทดลองและการวิเคราะห์ของการทำงานของชุดอุปกรณ์ที่ได้นำอัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียงที่ทำงานแบบเวลาจริงมาใช้งาน เพื่อใช้ในการค้นหาสถานีที่เป็นต้นกำเนิดของสัญญาณรบกวนที่เสนอในบทที่ 3 โดยนำพารามิเตอร์ต่าง ๆ จากการทดลองการทำงานของอัลกอริทึมบนโปรแกรมคอมพิวเตอร์ในบทที่ 4 มาใช้งาน และเพิ่มกระบวนการทำงานเพื่อให้การทำงานเป็นได้อย่างอัตโนมัติ เพื่อทดสอบประสิทธิภาพการทำงานของชุดอุปกรณ์จริงที่นำอัลกอริทึมที่พัฒนา มาใช้งาน โดยในการทดลองการเปรียบเทียบความคล้ายกันของเสียงที่ได้จากการจำลองการออกอากาศของสถานีวิทยุ ซึ่งจะสามารถกำหนดได้ว่าสัญญาณเสียงที่ทำการทดลองจะมีความคล้ายกันของเสียงหรือไม่

5.2 การทดสอบการทำงานของอัลกอริทึมบนชุดอุปกรณ์ต้นแบบ

การนำอัลกอริทึมของการเปรียบเทียบความคล้ายกันของเสียงที่ทำงานแบบเวลาจริง มาใช้งานบนชุดอุปกรณ์จริง เพื่อใช้ในการค้นหาสถานีที่เป็นต้นกำเนิดของสัญญาณรบกวน จำเป็นต้องเพิ่มกระบวนการทำงานเพื่อให้ทำงานได้อย่างอัตโนมัติโดยมีขั้นตอนการทำงานของชุดอุปกรณ์ และการเตรียมการก่อนการทดลองดังนี้

5.2.1 ขั้นตอนการทำงาน

ในการทำงานของชุดอุปกรณ์ที่นำอัลกอริทึมที่พัฒนามาใช้งานในการเปรียบเทียบความคล้ายกันของเสียง ได้มีการเขียนชุดคำสั่งการทำงานด้วยภาษาไพทอน (Python programming language) เพื่อควบคุมการทำงานของอุปกรณ์ที่เชื่อมต่อของบอร์ด Raspberry Pi และกระบวนการเปรียบเทียบความคล้ายกันของเสียง ซึ่งจะมีขั้นตอนการทำงานตามที่ได้ออกแบบไว้ในหัวข้อ 3.4 โดยมีรายละเอียดดังนี้

ขั้นตอนที่ 1 บันทึกสัญญาณเสียงด้วยตัวจับสัญญาณเสียง โดยที่กำหนดให้ตัวจับสัญญาณเสียงตัวที่ 1 จะรับสัญญาณเสียงผสม (Mix audio) จากเครื่องรับวิทยุที่รับคลื่นสัญญาณรบกวน และตัวจับสัญญาณเสียงตัวที่ 2 จะรับสัญญาณเสียงเดี่ยว (Single audio) จากเครื่องรับวิทยุที่รับคลื่นจากสถานีวิทยุ

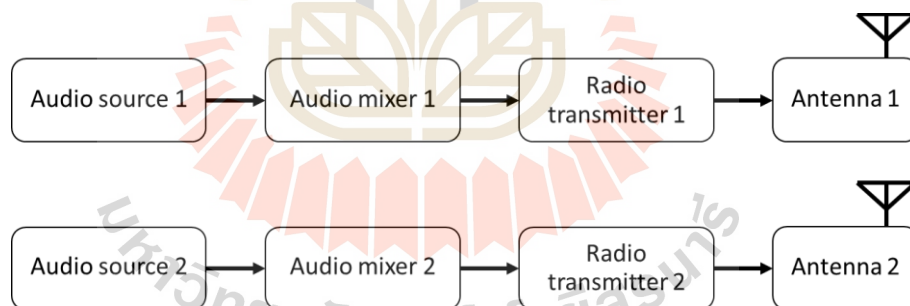
ขั้นตอนที่ 2 ทำการเปรียบเทียบความคล้ายกันของเสียงด้วยอัลกอริทึมที่ได้เสนอ โดยใช้ค่าเทรสโพลด์ที่ใช้ในการตัดสินใจความคล้ายกันของเสียงที่ได้จากการทดลองบนโปรแกรมคอมพิวเตอร์ โดยที่ในย่านความถี่ใช้ค่าเทรสโพลด์ 0.7 และในย่านความถี่กลางและย่านความถี่สูงใช้ค่าเทรสโพลด์ 0.5

ขั้นตอนที่ 3 ทำการตรวจสอบผลของการเปรียบเทียบความคล้ายกันของเสียง เพื่อทำการให้คะแนนของการเปรียบเทียบ โดยถ้าหากพบว่ามีค่าความคล้ายกันของเสียงจะทำการเพิ่มคะแนนของการเปรียบเทียบ แต่ถ้าพบว่าไม่มีความคล้ายกันของเสียงจะทำการลดคะแนนของการเปรียบเทียบ

ขั้นตอนที่ 4 ทำการตรวจสอบคะแนนของการเปรียบเทียบ เพื่อตัดสินใจว่าควรส่งสัญญาณให้เปลี่ยนคลื่นความถี่ของสถานีวิทยุที่กำลังออกอากาศหรือไม่ โดยถ้าหากคะแนนมีค่ามากกว่า 0 การทำงานใหม่อีกครั้งโดยอัตโนมัติ ตรงกันข้ามถ้าคะแนนน้อยกว่าหรือเท่ากับ 0 ส่งสัญญาณเพื่อให้เครื่องรับวิทยุรับคลื่นความถี่ของสถานีวิทยุที่กำลังออกอากาศในช่องถัดไป จากนั้นทำการเริ่มนับคะแนนใหม่ และเริ่มการทำงานใหม่อีกครั้ง

5.2.2 เตรียมอุปกรณ์ในการทดลอง

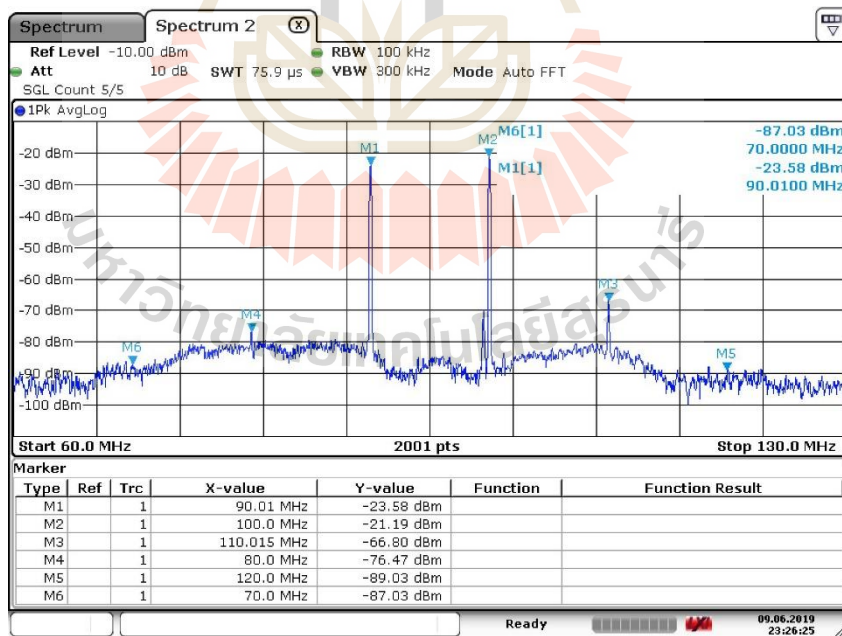
ในการทดลองได้จำลองการออกอากาศของสถานีวิทยุ โดยที่จัดเตรียมอุปกรณ์ภาคส่งสัญญาณวิทยุ เพื่อจำลองการออกอากาศของสถานีวิทยุ และการรบกวนแบบ Reverse Intermodulation (RIM) โดยที่อุปกรณ์จะประกอบไปด้วย แหล่งจ่ายสัญญาณเสียง (audio source) เครื่องผสมสัญญาณเสียง (audio mixer) เครื่องส่งสัญญาณวิทยุ (radio transmitter) และสายอากาศ (antenna) จำนวน 2 ชุดดังรูปที่ 5.1 ซึ่งได้มีการกำหนดความถี่ที่เครื่องส่งสัญญาณวิทยุจะออกอากาศในความถี่ 90 MHz และ 100 MHz และจะเกิดการสัญญาณรบกวนจาก RIM ที่ความถี่ 110 MHz ดังภาพที่ 5.3



รูปที่ 5.1 อุปกรณ์ภาคส่งสัญญาณวิทยุในการจำลองการออกอากาศ



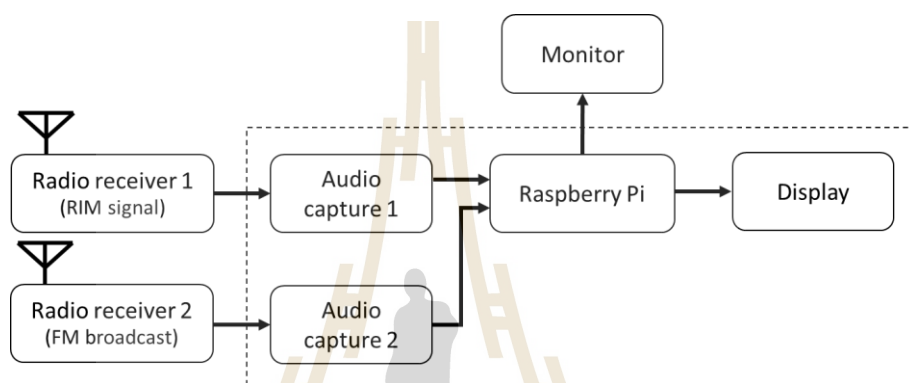
รูปที่ 5.2 อุปกรณ์จริงภาคส่งสัญญาณวิทยุในการจำลองการออกอากาศ



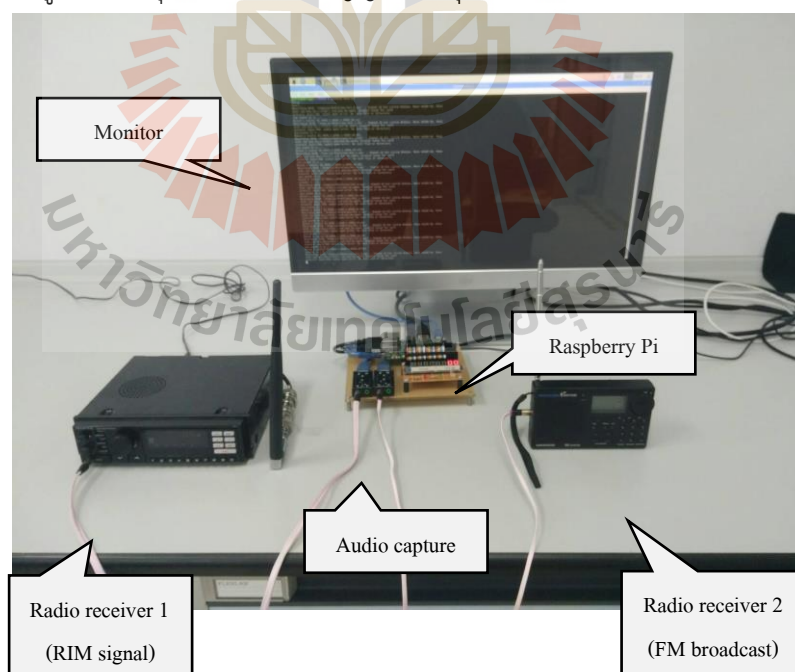
Date: 9 JUN 2019 23:26:25

รูปที่ 5.3 คลื่นความถี่ที่ได้จากการจำลองการออกอากาศและ RIM

ในชุดอุปกรณ์ภาครับสัญญาณ เป็นการทำงานของชุดอุปกรณ์ที่ออกแบบในบทที่ 3.4 และมีอุปกรณ์เพิ่มเติมคือ เครื่องรับวิทยุ (radio receiver) ทำหน้าที่ในการรับคลื่นวิทยุจากภาคส่งและแปลงเป็นสัญญาณเสียง และจอภาพทำหน้าที่แสดงการทำงานเพื่อควบคุมและบันทึกผลการทดลอง แสดงดังรูปที่ 5.3 โดยที่กำหนดให้ตัวจับสัญญาณเสียงตัวที่ 1 เชื่อมต่อเข้ากับเครื่องรับวิทยุเครื่องที่ 1 ซึ่งจะรับคลื่นจากสัญญาณรบกวน และตัวจับสัญญาณเสียงตัวที่ 2 เชื่อมต่อเข้ากับเครื่องรับวิทยุเครื่องที่ 2 ซึ่งจะรับคลื่นจากสถานีวิทยุ และเชื่อมต่อจอภาพเข้ากับบอร์ด Raspberry Pi ของชุดอุปกรณ์ที่ออกแบบ



รูปที่ 5.4 อุปกรณ์ภาครับสัญญาณวิทยุในการจำลองการออกอากาศ



รูปที่ 5.5 อุปกรณ์จริงภาครับสัญญาณวิทยุในการจำลองการออกอากาศ

5.3 การทดสอบประสิทธิภาพการทำงานของอัลกอริทึม

5.3.1 การตัดสินความคล้ายกันของเสียง

ในการทดลองการตัดสินความคล้ายกันของเสียง ได้ทำการเปรียบเทียบความคล้ายกันของเสียงที่ได้จากการจำลองการออกอากาศของสถานีวิทยุ โดยที่ในการทดลองได้แบ่งการทดลองเป็น 3 กลุ่ม กลุ่มที่ 1 ทดลองการเปรียบเทียบความคล้ายกันของเสียงที่มีความคล้ายกันของเสียง (เสียงเดี่ยว-เสียงเดี่ยว) โดยการกำหนดให้เครื่องรับวิทยุทั้ง 2 เครื่องรับคลื่นจากสถานีเดียวกัน เพื่อทดสอบความถูกต้องในการตัดสินความคล้ายกันของเสียง กลุ่มที่ 2 ทดลองการเปรียบเทียบความคล้ายกันของเสียงที่มีความคล้ายกันของเสียง (เสียงผสม-เสียงเดี่ยว) โดยการกำหนดให้เครื่องรับวิทยุ 1 รับคลื่นจากสัญญาณรบกวน และเครื่องรับวิทยุ 2 รับคลื่นจากสถานีวิทยุ เพื่อทดสอบความถูกต้องในการตัดสินความคล้ายกันของเสียง และเป็นทดสอบการทำงานที่ใกล้เคียงกับสภาพการทำงานจริงที่จำหน่ายอุปกรณ์ไปใช้งาน กลุ่มที่ 3 ทดลองการเปรียบเทียบความคล้ายกันของเสียงที่ไม่มีความคล้ายกันของเสียง โดยกำหนดให้เครื่องรับวิทยุทั้ง 2 เครื่องรับคลื่นจากสถานีวิทยุต่างกัน เพื่อทดสอบความผิดพลาดในการตัดสินความคล้ายกันของเสียง ซึ่งการทดลองในแต่ละกลุ่มจะทำการทดลองจำนวน 300 ครั้งและบันทึกผลการทดลอง ซึ่งผลการทดลองแสดงดังตารางที่ 5.1 ซึ่งจะแสดงจำนวนครั้งและอัตราร้อยละที่ตัดสินว่ามีความคล้ายกันของเสียง

ตารางที่ 5.1 ตารางผลการทดลองการทำงานของอัลกอริทึมจากการทดลองกลุ่มละ 300 ครั้ง

กลุ่ม	จำนวนครั้งที่พบว่ามี ความคล้ายกันของเสียง	ร้อยละ
การเปรียบเทียบความคล้ายกันของเสียงที่ มีความคล้ายกันของเสียง (เสียงเดี่ยว-เสียงเดี่ยว)	260	86.67
การเปรียบเทียบความคล้ายกันของเสียงที่ มีความคล้ายกันของเสียง (เสียงผสม-เสียงเดี่ยว)	214	71.33
การเปรียบเทียบความคล้ายกันของเสียงที่ ไม่มีความคล้ายกันของเสียง	42	14.00

จากตารางที่ 5.1 ผลการทดลองจากกลุ่มที่ 1 ซึ่งเป็นการทดสอบการทำงานของอัลกอริทึม มีความถูกต้องอยู่ที่ 86.67 % ซึ่งถือใกล้เคียงกับการทดสอบบนคอมพิวเตอร์ และจากผลการทดลองจากกลุ่มที่ 2 ทำให้ทราบว่าเมื่อนำอัลกอริทึมไปใช้งานสภาพแวดล้อมจริง จะมีความถูกต้องในการตัดสินที่ 71.33 % โดยที่ผลการทดลองจากกลุ่มที่ 3 ให้ความผิดพลาดในการตัดสินที่ 14.00 % นอกจากนี้ในระบบการค้นหาสถานีที่เป็นต้นกำเนิดของสัญญาณรบกวนได้เพิ่มขั้นตอนการให้คะแนนของการเปรียบเทียบ เพื่อเพิ่มอัตราความสำเร็จในการค้นหาสถานีที่เป็นต้นกำเนิดของสัญญาณรบกวนอีกส่วนหนึ่ง

5.3.2 เวลาที่ใช้ในการเปรียบเทียบ

จากการทดลองข้างต้นแล้ว ผู้ได้ทำการบันทึกเวลาที่ใช้ในการเปรียบเทียบความคล้ายกันของเสียงพร้อมกันด้วย โดยที่เวลาที่ใช้ในการเปรียบเทียบแต่ละครั้งอยู่ที่ 51 วินาทีโดยเฉลี่ย ซึ่งการทำงานบนชุดอุปกรณ์ต้นแบบได้มีการเพิ่มกระบวนการทำงานต่าง ๆ เช่น การบันทึกสัญญาณเสียง ระบบการให้คะแนนการเปรียบเทียบ และการทำงานใหม่อย่างอัตโนมัติเมื่อพบว่าผลของการเปรียบเทียบคือไม่มีความคล้ายกันของเสียง

5.4 สรุป

ในบทนี้ได้ทดสอบการทำงานของชุดอุปกรณ์ต้นแบบที่ได้นำอัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียงที่ทำงานแบบเวลาจริงไปใช้งาน เพื่อใช้ในการค้นหาสถานีที่เป็นต้นกำเนิดของสัญญาณรบกวน โดยการทดลองการเปรียบเทียบความคล้ายกันของเสียงที่ได้จากการจำลองการออกอากาศของสถานีวิทยุและการเกิดสัญญาณรบกวน ซึ่งการทดลองแบบออกเป็น 3 กลุ่มคือ กลุ่มที่ 1 ทดลองการเปรียบเทียบความคล้ายกันของเสียงที่มีความคล้ายกันของเสียง (เสียงเดี่ยว-เสียงเดี่ยว) เพื่อทดสอบการทำงานของอัลกอริทึม กลุ่มที่ 2 ทดลองการเปรียบเทียบความคล้ายกันของเสียงที่มีความคล้ายกันของเสียง (เสียงผสม-เสียงเดี่ยว) เพื่อทดสอบถูกต้องในการตัดสินใจความคล้ายกันของเสียง และเป็นทดสอบการทำงานที่ใกล้เคียงกับสภาพแวดล้อมจริงที่จะนำชุดอุปกรณ์ไปใช้งาน กลุ่มที่ 3 ทดลองการเปรียบเทียบความคล้ายกันของเสียงที่ไม่มีความคล้ายกันของเสียง เพื่อทดสอบความผิดพลาดในการตัดสินใจความคล้ายกันของเสียง และผลการทดสอบเมื่อนำชุดอุปกรณ์ที่ออกแบบไปใช้งานในสภาพแวดล้อมจริง จะมีความถูกต้องในการตัดสินใจผลการเปรียบเทียบ 71.33 % และมีความผิดพลาดในการตัดสินใจผลการเปรียบเทียบ 14.00 % และการเปรียบเทียบความคล้ายกันของเสียงในแต่ละครั้งใช้เวลา 51 วินาทีโดยประมาณ นอกจากนี้มีขั้นตอนการให้คะแนนการเปรียบเทียบ ซึ่งจะช่วยให้เพิ่มความแม่นยำของการเปรียบเทียบให้มากขึ้น

บทที่ 6

สรุปงานวิจัยและข้อเสนอแนะ

6.1 สรุป

ในปัจจุบันได้มีการจัดตั้งสถานีวิทยุกระจายเสียงในระบบเอฟเอ็มขึ้นเป็นจำนวนมาก ส่งผลให้เกิดการรบกวนในย่านความถี่อื่น ๆ โดยเฉพาะอย่างยิ่งการรบกวนการสื่อสารระหว่างเครื่องบินกับสถานีควบคุมภาคพื้นดินทำให้มีความเสี่ยงที่จะเกิดความเสียหายต่อชีวิต และทรัพย์สินขึ้นได้ และมีหน่วยงานที่เกี่ยวข้องทำการค้นหาคลื่นสัญญาณรบกวนที่เกิดขึ้นอย่างต่อเนื่อง แต่ปัญหาของกระบวนการค้นหาแหล่งกำเนิดของสัญญาณรบกวนในปัจจุบันคือ ไม่มีเครื่องมือวัดที่เหมาะสมต่ออาศัยทรัพยากรบุคคลที่มีความชำนาญ เพื่อทำการเปรียบเทียบความคล้ายกันของเสียง ระหว่างสัญญาณเสียงจากสถานีวิทยุที่กำลังออกอากาศกับสัญญาณเสียงจากคลื่นความถี่รบกวนที่เกิดขึ้น จึงทำให้กระบวนการค้นหาแหล่งกำเนิดของสัญญาณรบกวนในปัจจุบันเป็นไปอย่างล่าช้า และสิ้นเปลืองทรัพยากรบุคคลเป็นจำนวนมาก ผู้วิจัยจึงเล็งเห็นถึงความสำคัญในการพัฒนาอัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียง เพื่อใช้ในกระบวนการค้นหาแหล่งกำเนิดของสัญญาณรบกวน ซึ่งจะช่วยแก้ไขปัญหที่เกิดขึ้นได้อย่างรวดเร็ว

งานวิจัยนี้นำเสนออัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียงที่ทำงานแบบเวลาจริง ซึ่งจะใช้ในการเปรียบเทียบความคล้ายกันของเสียงจาก 2 แหล่งกำเนิดโดยใช้หลักการคณิตศาสตร์อย่างง่าย มีการทำงานที่ไม่ซับซ้อน และทำงานได้อย่างรวดเร็ว โดยในกระบวนการทำงานจะแบ่งออกเป็น 3 กระบวนการย่อยคือ ในกระบวนการเตรียมข้อมูล ซึ่งจะทำการเตรียมข้อมูลของสัญญาณเสียงทั้งสอง โดยการซีกสัญญาณเพื่อให้สัญญาณเสียงเป็นสัญญาณที่ไม่ต่อเนื่องทางเวลาทำให้สามารถคำนวณทางคอมพิวเตอร์ได้ จากนั้นนอร์มัลไลเซชันข้อมูลเพื่อให้ทั้งสองสัญญาณเสียงมีแอมพลิจูดที่ใกล้เคียงกัน และแบ่งการพิจารณาข้อมูลเป็น 3 ย่านความถี่ตัวกรองเพื่อเพิ่มความแม่นยำในการตัดสินใจความคล้ายกันของเสียง ขั้นสุดท้ายของกระบวนการเตรียมข้อมูลจะทำการเฉลี่ยข้อมูลที่ได้จากการกรองข้อมูลเพื่อให้สามารถทนต่อการคลาดเคลื่อนทางเวลาได้เล็กน้อย ถัดจากนั้นเป็นกระบวนการคำนวณผลต่างของเสียง ซึ่งจะหาผลต่างของข้อมูลของทั้งสองสัญญาณเสียง เพื่อหาค่าความคล้ายกันของเสียง (I_x) ของแต่ละย่านความถี่ และกระบวนการตัดสินใจความคล้าย ซึ่งจะนำค่าความคล้ายกันของเสียง (I_x) มาพิจารณากับค่าเทรสโฮลด์ของแต่ละย่านความถี่ และทำการรวมการพิจารณาจากทั้ง 3 ย่านความถี่มาทำการโหวต 2 ใน 3 เสียงเพื่อตัดสินใจว่าทั้งสองเสียงมีความคล้ายกันหรือไม่ และได้นำเสนอชุดอุปกรณ์ต้นแบบที่นำเอาอัลกอริทึมที่พัฒนาไปใช้งานและเพิ่มกระบวนการทำงานต่าง ๆ เช่นการบันทึกสัญญาณเสียง และการให้คะแนนการเปรียบเทียบ เพื่อให้กระบวนการค้นหาแหล่งกำเนิดของสัญญาณรบกวนมีความแม่นยำ สามารถทำงานได้อย่างอัตโนมัติ โดยจากการทดสอบการทำงานบนโปรแกรมคอมพิวเตอร์ทำให้ทราบถึงค่าเทรสโฮลด์ที่ใช้ในการตัดสินใจความคล้ายกันของเสียงของย่านความถี่ต่ำ กลาง และสูงคือ 0.7 0.5 และ 0.5 ตามลำดับ และจาก

ผลการทดสอบหลังจากนำค่าเทรสโพลต์ดังกล่าวมาทำการเปรียบเทียบความคล้ายกันของเสียงบนโปรแกรมคอมพิวเตอร์ได้คะแนนความแม่นยำ 93.04 คะแนน และผลการทดสอบการเปรียบเทียบความคล้ายกันของเสียงบนชุดอุปกรณ์ต้นแบบในสภาพแวดล้อมจริง การเปรียบเทียบความคล้ายกันของเสียงมีความถูกต้องในการตัดสินใจ 71.33 % และมีความผิดพลาดในการตัดสินใจ 14.00 % และใช้เวลาในการเปรียบเทียบในแต่ละครั้ง 51 วินาทีโดยประมาณ

6.2 ปัญหาและข้อเสนอแนะ

ในการออกแบบการทำงานของอัลกอริทึมมีความต้องการให้มีการทำงานอย่างง่าย ไม่ซับซ้อน เพื่อให้ทำงานแบบเวลาจริงและสามารถทำงานบนอุปกรณ์ประมวลผลขนาดเล็กได้ ซึ่งเมื่อนำอัลกอริทึมไปใช้งานในสภาพแวดล้อมจริงอาจมีการรบกวนจากสัญญาณรบกวน (noise) จึงจะทำให้ความแม่นยำในการตัดสินใจความคล้ายกันของเสียงลดลง เพื่อแก้ไขปัญหาดังกล่าวจำเป็นต้องพัฒนาให้มีความทนต่อสัญญาณรบกวนให้มากขึ้น และจากการทดสอบการทำงานบนชุดอุปกรณ์ต้นแบบทำให้ทราบว่า การเปรียบเทียบเสียงที่นอกเหนือจากเสียงเพลงจะทำให้ความแม่นยำต่ำลง เนื่องจากองค์ประกอบของเสียงมีน้อยเกินไป และทางผู้วิจัยได้เพิ่มฟังก์ชันขำการเปรียบเทียบเมื่อพบว่าเป็นเสียงเจี๊ยบ หรือกำลังของเสียงที่ทำการเปรียบเทียบนั้นต่ำเกินไป

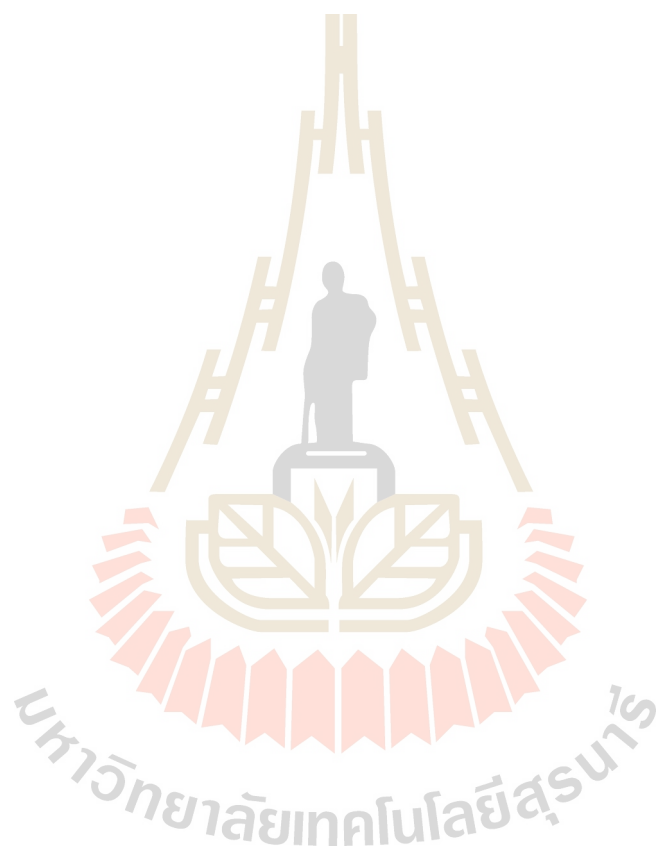
6.3 แนวทางการพัฒนาในอนาคต

งานวิจัยนี้ได้นำเสนออัลกอริทึมในการเปรียบเทียบความคล้ายกันของเสียงที่ทำงานแบบเวลาจริง โดยใช้หลักการคณิตศาสตร์อย่างง่ายมีการทำงานที่ไม่ซับซ้อน และทำงานแบบเวลาจริง ซึ่งสามารถนำอัลกอริทึมมาประยุกต์ใช้งานในการค้นหาสถานีวิทยุที่เป็นต้นกำเนิดของสัญญาณรบกวนที่ทำงานแบบอัตโนมัติได้ สำหรับแนวทางในการพัฒนาในอนาคต สามารถพัฒนาอัลกอริทึมเพื่อประยุกต์ใช้งานในรูปแบบอื่นเช่น การตรวจสอบตนเองของสถานีวิทยุว่าตนเองเป็นแหล่งกำเนิดสัญญาณรบกวนหรือไม่ เครือข่ายระบบการค้นหาแหล่งกำเนิดของสัญญาณรบกวนโดยอัตโนมัติ ซึ่งจะทำให้ติดตั้งและสั่งการทำงานชุดอุปกรณ์ได้ในหลายพื้นที่ในเวลาเดียวกัน เป็นต้น หรือพัฒนากระบวนการทำงานของชุดอุปกรณ์ให้ทำงานร่วมกับ Software-defined radio (SDR) เพื่อให้การค้นหาสถานีที่เป็นต้นกำเนิดของสัญญาณทำงานอย่างอัตโนมัติมากขึ้น

รายการอ้างอิง

- [1] ผศ. ดร.รังสรรค์ ทองทา (2546).รายงาน โครงการศึกษาวิเคราะห์พฤติกรรม Reverse Intermodulation การออกอากาศของสถานีที่อยู่ใกล้กัน และแนวทางการดูแลให้เกิดการกวนกันให้น้อยของสถานีวิทยุกระจายเสียง สำหรับการทดลองประกอบกิจการกระจายเสียง
- [2] A. Bruce Carlson and Paul B. Crilly, "Communication Systems," McGraw-Hill, 5th Edition. Boston, 2010
- [3] Lloyd Butler, "Intermodulation Performance and Measurement of Intermodulation Components", [Online]. Available: <http://users.tpg.com.au/ldbutler/Intermodulation.htm>
- [4] Raymond McArthur, "Intermodulation Fundamental", Sinclair Technologies Inc.
- [5] David Hand C.Eng MIET, "The RF challenges of ATC communications" Consultant Engineer - Park Air Systems Ltd.
- [6] Haitsma, Jaap and Kalker, Antonius, "A Highly Robust Audio Fingerprinting System." International Symposium on Music Information Retrieval (ISMIR), Eindhoven : s.n., 2002, pp. 107-115.
- [7] P.Cano, E.Battle, T.Kalker, J.Haitsma, "A Review of Algorithms for Audio Fingerprinting", IEEE International workshop on MMSP, 2002.
- [8] Wang, Avery Li-Chun., "An Industrial-Strength Audio Search Algorithm." ISMIR, London : Shazam Entertainment, Ltd., 2003.
- [9] T.Park,S.K.Beack,T.Lee, "A noise robust audio fingerprint extraction technique for mobile devices using gradient histograms",Consumer Electronics - Berlin (ICCE-Berlin), 2015 IEEE 5th International Conference on, 6-9 Sept. 2015.
- [10] Yong Fan, Shuang Feng, "A Music Identification System Based on Audio Fingerprint", Applied Computing and Information Technology/3rd Intl Conf on Computational Science/Intelligence and Applied Informatics/1st Intl Conf on Big Data, Cloud Computing, Data Science & Engineering (ACIT-CSII-BCD), 2016 4th Intl Conf on, 12-14 Dec. 2016

- [11] Reinhard Sonnleitner, Gerhard Widmer, “Robust Quad-Based Audio Fingerprinting”, IEEE/ACM Transactions on Audio, Speech, and Language Processing, March 2016.
- [12] Shanshan Yao, Baoning Niu, Jianquan Liu, “Audio Identification by Sampling Sub-fingerprints and Counting Matches, IEEE Transactions on Multimedia, Volume: 19, Issue: 9, Sept. 2017.





ภาคผนวก ก
ชุดคำสั่งการทำงาน

ชุดคำสั่งจำลองการทำงานบนโปรแกรม MATLAB

```
%% Setup
sec = 10 ;                %time (sec)
winz = .001;              %Windows Size (winz)
th_l = 0.7;               %Threshold-Low (thl)
th_m = 0.5;               %Threshold-Mid (thm)
th_h = 0.5;               %Threshold-High (thh)

%% Data preparing
[D1,SR1] = readaudio('s01.mp3'); %singel-audio
[D2,SR2] = readaudio('m0102.mp3'); %mix-audio

if SR1 ~= SR2
    D2 = resample(D2,SR1,SR2);
else
    SR = SR1;
end

% Normalize

D1_n = normal(D1,SR,sec);
D2_n = normal(D2,SR,sec);

% Filter

[D1_l,D2_l] = lowpass(D1_n,D2_n,SR);
[D1_m,D2_m] = bandpass(D1_n,D2_n,SR);
[D1_h,D2_h] = highpass(D1_n,D2_n,SR);

% Average
```



```

D1_la = avg(D1_l,SR,winz);
D1_ma = avg(D1_m,SR,winz);
D1_ha = avg(D1_h,SR,winz);

```

```

D2_la = avg(D2_l,SR,winz);
D2_ma = avg(D2_m,SR,winz);
D2_ha = avg(D2_h,SR,winz);

```

```

%% Similarly calculate

```

```

% Subtraction

```

```

sub_dl = sum (abs(D2_la - D1_la)) /sum(D2_la)
sub_dm = sum (abs(D2_ma - D1_ma)) /sum(D2_ma)
sub_dh = sum (abs(D2_ha - D1_ha)) /sum(D2_ha)

```

```

%%Similarly decision

```

```

i = 0;

```

```

if sub_dl <= th_l

```

```

    i = i +1;

```

```

end

```

```

if sub_dm <= th_m

```

```

    i = i +1;

```

```

end

```

```

if sub_dh <= th_h

```

```

    i = i +1;

```

```

end

```

```

if i >= 2

```

```

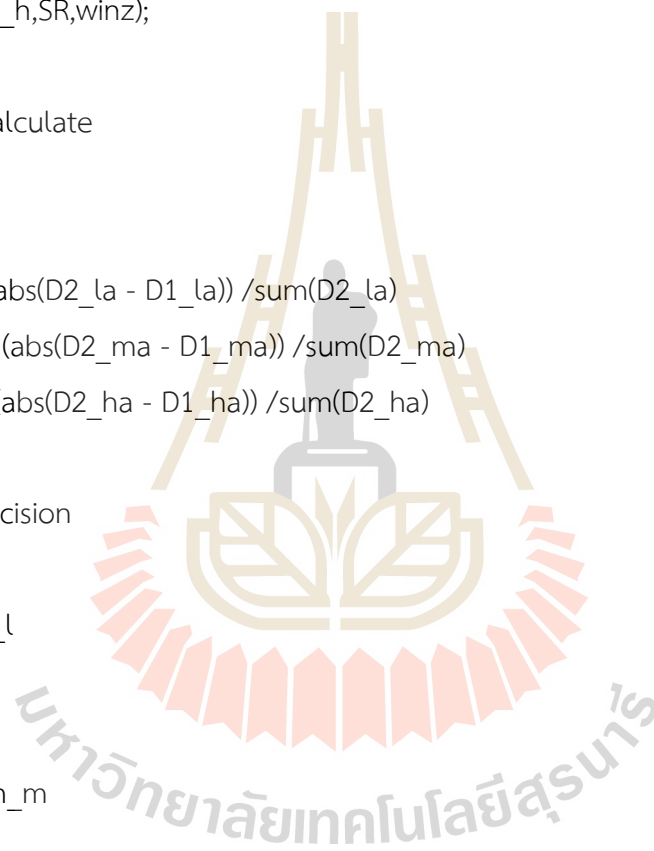
    result = 'similar'

```

```

else

```



```

    result = 'dissimilar'
end

function D_n = normal(D,SR,Sec)
% normalization

% sec = size(D,1) / SR ;
D = D(1:Sec*SR) ;

nor = sqrt( 1 / sum ( D.^2 ) );
D_n = D * nor;

end

function S = avg(D,SR,t)
%%% average
per = 0 ;
wiz = SR*t ;
ovr = floor( wiz*(per/100) );
wiz = floor(wiz);
nsf = fix( size(D,1)/(wiz-ovr) ) - floor(wiz/(wiz-ovr));

for i=0:nsf-1
    df = D( ((wiz-ovr)*i)+1 : ((wiz-ovr)*i)+wiz );
    S(i+1) = sum(abs(df))/(wiz-ovr);
end

end

function [D1_f,D2_f] = lowpass2(D1,D2,SR)
frq=340; %340 Hz

```

```

n = 6;                %Order (n)
Rp = .1;             %Ripple in the passband (Rp)
Rs = 80;            %Ripple in the stopband (Rs)
Wn = frq*2/SR;      %Normalized passband edge frequency (Wn)
Type = 'low';

```

```
% Transfer Function design
```

```
[b,a] = ellip(n,Rp,Rs,Wn,Type);
```

```
D1_f = filter(b,a,D1);
```

```
D2_f = filter(b,a,D2);
```

```
end
```

```
function [D1_f,D2_f] = bandpass2(D1,D2,SR)
```

```
frq = [340 3400];    %340-3400 Hz
```

```
n = 6;                %Order (n)
```

```
Rp = .1;             %Ripple in the passband (Rp)
```

```
Rs = 80;            %Ripple in the stopband (Rs)
```

```
Wn = frq*2/SR;      %Normalized passband edge frequency (Wn)
```

```
Type = 'bandpass';
```

```
% Transfer Function design
```

```
[b,a] = ellip(n,Rp,Rs,Wn,Type);
```

```
D1_f = filter(b,a,D1);
```

```
D2_f = filter(b,a,D2);
```

```
end
```

```
function [D1_f,D2_f] = highpass2(D1,D2,SR)
```

```
frq = 3400;          %3400 Hz
```

```
n = 6;                %Order (n)
```

```
Rp = .1;             %Ripple in the passband (Rp)
```

```
Rs = 80;            %Ripple in the stopband (Rs)
```

```

Wn = frq*2/SR;           %Normalized passband edge frequency (Wn)
Type = 'high';

% Transfer Function design
[b,a] = ellip(n,Rp,Rs,Wn,Type);
D1_f = filter(b,a,D1);
D2_f = filter(b,a,D2);
End

```

ชุดคำสั่งการทำงานบนบอร์ด Raspberry Pi

```

#setup Mix-audio in device1 & Single-audio in device2
print('Setup...')
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
lcd1=17
lcd2=18
lcd3=27
GPIO.setup(lcd1,GPIO.OUT) #LCD1
GPIO.setup(lcd2,GPIO.OUT) #LCD2
GPIO.setup(lcd3,GPIO.OUT) #Pulse
GPIO.output(lcd1,0)
GPIO.output(lcd2,0)
GPIO.output(lcd3,0)

import subprocess
import numpy as np
from scipy.io.wavfile import read
from scipy.signal import butter, lfilter
import time
from multiprocessing import Process,Array

```

```

from time import sleep
from f_show8digit import setup8digit,show8digit
#---Function-----
def record(num=0,time=10):
    cmd0 = " arecord -f cd -D hw:1 , 0 - c 1 - d {}
/home/pi/RASCA/wav/dv1_{}.wav".format(time, num)
    cmd1 = " arecord -f cd -D hw:2 , 0 - c 1 - d {}
/home/pi/RASCA/wav/dv2_{}.wav".format(time, num)
    sp1 = subprocess.Popen(cmd0, shell=True)
    sp2 = subprocess.Popen(cmd1, shell=True)

def readwav(num=0):
    nb_bit=16
    max_nb_bit=float(2 ** (nb_bit-1))
    cmd0 = "/home/pi/RASCA/wav/dv1_{}.wav".format(num)
    [rate0,x0] = read(cmd0)
    data0= x0/(max_nb_bit + 1.0)
    cmd1 = "/home/pi/RASCA/wav/dv2_{}.wav".format(num)
    [rate1,x1] = read(cmd1)
    data1= x1/(max_nb_bit + 1.0)
    return data0,data1,rate0,rate1

def norm(data_bN):
    data_N = data_bN / (( sum( data_bN ** 2 ) ) ** 0.5)
    return data_N

def avg(data_bA,sr,ws):
    data_fA = []
    winz = round(sr * ws)
    nsf = np.fix(len(data_bA)/winz)
    for ic in range(0,int(nsf)):

```

```
    data_dv = data_bA[winz*ic:(winz*(ic+1))]
    data_fA.append(sum(abs(data_dv))/winz)
return data_fA
```

```
def filter_low(data1,data2, lowcut, fs, order=5):
    nyq = 0.5 * fs
    low = lowcut / nyq
    b, a = butter(order,low, btype='low')
    y1 = lfilter(b, a, data1)
    y2 = lfilter(b, a, data2)
    return y1,y2
```

```
def filter_mid(data1,data2,lowcut, highcut, fs, order=5):
    nyq = 0.5 * fs
    low = lowcut / nyq
    high = highcut / nyq
    b, a = butter(order, [low, high], btype='band')
    y1 = lfilter(b, a, data1)
    y2 = lfilter(b, a, data2)
    return y1,y2
```

```
def filter_high(data1,data2,highcut, fs, order=5):
    nyq = 0.5 * fs
    high = highcut / nyq
    b, a = butter(order, high, btype='high')
    y1 = lfilter(b, a, data1)
    y2 = lfilter(b, a, data2)
    return y1,y2
```

```
def sub(dM,dS):
    SMS_M = np.array([])
```

```

for id in range(0,int(len(dM))):
    sub = dS[id] - dM[id]
    sum_sub =np.append(sum_sub ,sub)
result = sum( abs(sum_sub) ) / sum(dM)
return result

#-----Process2&3-----
def showlcd1(q):#---- LCD1 ----
    st = q[0]
    lcd1=17
    lcd2=18
    lcd3=27
    if st == 0: #0
        GPIO.output(lcd1,1)
        sleep(0.3)
        GPIO.output(lcd1,0)
        sleep(1.7)
    elif st == 1: #1
        sleep(1.5)
        GPIO.output(lcd1,1)
        sleep(0.5)
        GPIO.output(lcd1,0)
        sleep(1.5)
    elif st == 2: #2
        sleep(1.5)
        GPIO.output(lcd1,1)
        sleep(0.5)
        GPIO.output(lcd1,0)
        sleep(0.5)
        GPIO.output(lcd1,1)
        sleep(0.5)

```

```

        GPIO.output(lcd1,0)
        sleep(1.5)
else: #3
    sleep(1.5)
    GPIO.output(lcd1,1)
    sleep(0.5)
    GPIO.output(lcd1,0)
    sleep(0.5)
    GPIO.output(lcd1,1)
    sleep(0.5)
    GPIO.output(lcd1,0)
    sleep(0.5)
    GPIO.output(lcd1,1)
    sleep(0.5)
    GPIO.output(lcd1,0)
    sleep(1.5)
def showlcd2(q):#----- LCD2 -----
    st = q[1]
    lcd1=17
    lcd2=18
    lcd3=27
    if st == 0: #0
        GPIO.output(lcd2,0)
    elif st in range(1,6): #1-5
        for i in range(1,st+1):
            GPIO.output(lcd2,1)
            sleep(0.5)
            GPIO.output(lcd2,0)
            sleep(0.5)
        sleep(2.5)

```



```

else: #6
    GPIO.output(lcd1,0)
    GPIO.output(lcd3,1)
    sleep(0.5)
    GPIO.output(lcd3,0)
    q[1] = 0

```

```

#----Process1-----
def find_lx(num=0,time_r=10):
    lowcut = 340.0
    highcut = 3400.0
    time_ws = 0.01
    lx_l = 1.0
    lx_m = 1.0
    lx_h = 1.0
    tmf = time.time() + time_r + 2
    record(num,time_r)
    while time.time() < tmf:
        pass

    (dM,dS,rM,r1) = readwav(num)
    #check audio is low-sound ?
    print(sum(abs(dM))/(rM*time_),',',sum(abs(dS))/(rs*time_))
    if (sum(abs(d0))/(r0*time_) <= 0.09) and (sum(abs(d1))/(r1*time_) <= 0.09):
        print('low-sound in 1 + 2....')
        st = 3
        return lx_l,lx_m,lx_h,st
    elif sum(abs(d0))/(r0*time_) <= 0.09:
        print('low-sound in 1....')
        st = 1
        return lx_l,lx_m,lx_h,st

```

```

elif sum(abs(d1))/(r1*time_) <= 0.09:
    print('low-sound in 2...')
    st = 2
    return lx_l,lx_m,lx_h,st
else:
    # Algorithym doing
    dM_nor = norm(dM)
    dS_nor = norm(dS)

    (dM_low,dS_low) = filter_low(dM_nor,dS_nor, lowcut, rM)
    (dM_mid,dS_mid) = filter_mid(dM_nor,dS_nor, lowcut, highcut, rM)
    (dM_hig,dS_hig) = filter_high(dM_nor,dS_nor, highcut, rM)

    dM_low = avg(dM_low,rM,time_ws)
    dS_low = avg(dS_low,rM,time_ws)
    dM_mid = avg(dM_mid,rM,time_ws)
    dS_mid = avg(dS_mid,rM,time_ws)
    dM_hig = avg(dM_hig,rM,time_ws)
    ds_hig = avg(dS_hig,rM,time_ws)

    lx_low = sub(dM_low,dS_low)
    lx_mid = sub(dM_mid,dS_mid)
    lx_hig = sub(dM_hig,dS_hig)

    st = 0
    return lx_low,lx_mid,lx_hig,st

def process(q):
    (th_l,th_m,th_h,time_r,nc1,nc2) = getparameter()
    count = 0
    for iip in range(0,int(nc1)):

```

```

tu = time.time()
(lx_l,lx_m,lx_h,st) = find_lx(iip,time_r)
q[0] = st
if (lx_l <= th_l and lx_m <= th_m) or (lx_l <= th_l and lx_h <= th_h) or (lx_m
<= th_m and lx_h <= th_h):
    count = count + 1
q[1] = count
print('Round',iip+1,'count=',count, '(')
    print('%0.4f,%0.4f,%0.4f'%(lx_l,lx_m,lx_h))

# if 3 of 5
if count >= nc2:
    print('state1: x >= 3')
    q[2] = q[2] + 1
    show8digit(q[2])
else:
    print('state2: x < 3')
    q[2] = q[2] - 1
    if q[2] == -1:
        q[1] = 6
        q[2] = 0
        show8digit(q[2])
#-----
def getparameter():
    threshold_low = 0.7
    threshold_mid = 0.5
    threshold_high = 0.5
    time_record = 10
    num_compare1 = 5
    num_compare2 = 3
    return

```

```
#-----Main-----
if __name__ == "__main__":
    # --- run 1 time---
    print('Start...')
    setup8digit()
    st = Array('i',[0,0,0])
    Process(target=showlcd1,args=(st,)).start()
    show8digit(st[2])
    # --- Loop ---
    pro0 = Process(target=process,args=(st,))
    pro1 = Process(target=showlcd1,args=(st,))
    pro2 = Process(target=showlcd2,args=(st,))
    while (True):
        sleep(1)
        if pro0.is_alive() == False:
            pro0 = Process(target=process,args=(st,))
            pro0.start()
        if pro1.is_alive() == False:
            pro1 = Process(target=showlcd1,args=(st,))
            pro1.start()
        if pro2.is_alive() == False:
            pro2 = Process(target=showlcd2,args=(st,))
            pro2.start()
```



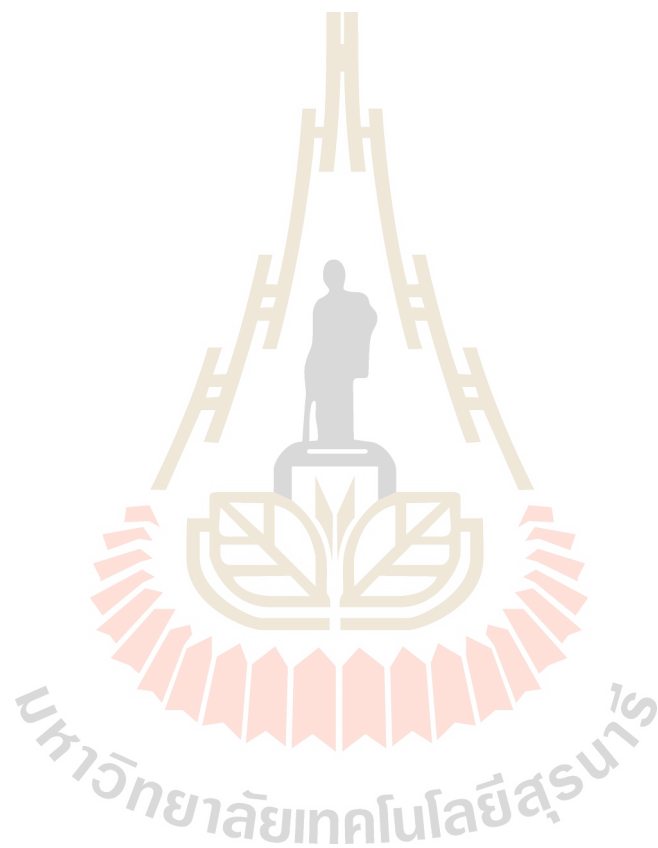
ภาคผนวก ข

บทความวิชาการที่ได้รับการตีพิมพ์เผยแพร่

มหาวิทยาลัยเทคโนโลยีสุรนารี

รายชื่อบทความวิชาการที่ได้รับการตีพิมพ์เผยแพร่

นันทวัฒน์ ใจเย็น, ปัญญา หันตุลา และรังสรรค์ ทองทา (2017) Real-time Audio Similarity Comparison Algorithm, in IEEE 15th Student Conference on Research and Development (SCoReD), 13-14 December 2017. Putrajaya, Malaysia.



Real-Time Audio Similarity Comparison Algorithm

Nantawat Jaiyen, Panya Hantula and Rangsak Tongta

School of Telecommunication Engineering, Institute of Engineering, Suranaree University of Technology.

111 University Avenue, Muang District Nakhon Ratchasima, Thailand.

E-mail: nantawat.jaiyen@gmail.com, panya_hantula@hotmail.co.th, tongta@sut.ac.th

Abstract— Interference in the Air Traffic Control (ATC) communication band usually comes from FM broadcast stations. To identify sources of the interference, manually scanning a receiver to compare similarity with the sound from the interference is the only method available. This paper presents a similarity comparison method that will make automatic interference identifying possible. The target audios are divided into 3 bands. A subtraction result between two audios of the same band will be decreased if the two audios are similar. To get a final result, at least 2 similar results from 3 bands will be considered positive. The experimental results show 93.42% of the right positive from 380 comparisons and 0.38% of the false positive from 3420 comparisons.

Keywords—audio similarity comparison; real-time audio comparison; reverse intermodulation

I. INTRODUCTION

The Air Traffic Control (ATC) systems of Thailand have interfered from FM radio broadcasting. The interference causes communication problems between controllers and pilots. A frequency range used by Air Traffic Control systems is 108-137 MHz which is close to FM broadcasting, using a frequency range between 88-108 MHz. Intermodulation signals from FM broadcast stations may cause interference fall into the frequency range of Air Traffic Control systems. FM broadcast stations use a nonlinear RF power amplifier in a final stage of a transmitter. A nonlinear RF power amplifier creates intermodulation signals when two frequencies or more present at the input stage of the amplifier. When two FM broadcast stations are closely located, the output signal from one station can radiate into the other nearby station, as shown in Figure 1. The signal travels in the reverse direction until it reaches the input stage of the amplifier. This reverse signal together with station own signal creates intermodulation signals, called Reverse Intermodulation (RIM), which can be calculated from $2f_2 - f_1$ and $3f_2 - 2f_1 \dots [1]$.

Since the RIM signal is a mixed signal between two FM broadcast stations, therefore an audio of the RIM signal consists of two sounds from both FM broadcast stations. When we want to identify the sources of the RIM signal, we use two radio receivers. One receiver listens to the RIM audio while the other receiver scans all the FM broadcast band for a similar audio, as shown in Figure 2. Scanning the receiver and comparing the two audios needs to be done manually. However, comparing one sound from an FM broadcast station with two sounds from the RIM signal is very difficult and tedious work. So, automatic comparison may benefit to identify the sources of the RIM signal.

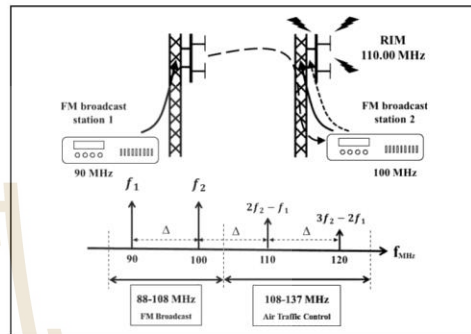


Fig. 1. Reverse Intermodulation signals from FM broadcast stations

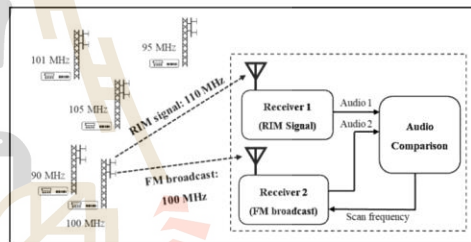


Fig. 2. Identifying the sources of the RIM signal

The Audio Fingerprint Algorithm [2-9] is automatic comparison which is high performance to identify individual audios. The algorithm uses the Fast Fourier Transform (FFT) to create unique fingerprints of audios that will be stored in database. The fingerprints of an unknown audio will be compared with the fingerprints from database to identify similarity. The algorithm requires a high-performance computer and long computational time. So, it is not suitable to be used in real-time receiver scanning.

This paper proposes a simple real-time audio similarity comparison algorithm. So, it requires a simple computer and short computational time. This algorithm can be used in real time with an embedded device.

TABLE I. SCORES OF THE ALGORITHM ACCORDING TO THRESHOLD VALUES

Threshold values	Right positive (%)			False positive (%)			Score		
	Low band	Mid band	High band	Low band	Mid band	High band	Low band	Mid band	High band
0.4	8.16	58.68	55.26	0.00	0.00	0.04	8.16	58.68	55.22
0.5	34.79	88.84	78.68	0.00	9.85	3.83	34.79	78.99	74.85
0.6	74.47	99.05	92.47	0.00	56.13	22.82	74.47	42.92	69.65
0.7	97.32	99.95	98.42	4.35	85.23	53.44	92.96	14.72	44.98
0.8	100.00	100.00	100.00	62.19	94.39	76.85	37.81	5.61	23.15

From Figure 4(a), the normalized subtraction results from similar files of mid band and high band are less than 0.5. Also, from Figure 4(b), the normalized subtraction results from dissimilar files of mid band and high band are more than 0.5. Therefore, using 0.5 for thresholding mid band and high band is suitable. The normalized subtraction results of low band are higher than the others. So, we use 0.7 for thresholding the low band. However, there are still some errors that we need to consider later.

From 380 comparisons of similar files, if the normalized subtraction results are less than the threshold, identifying similarity will be considered right positive. Percentages of right positive according to threshold values are shown in Table I. From 3420 comparisons of dissimilar files, if the normalized subtraction results are less than the threshold, identifying similarity will be considered false positive that a dissimilar song would be misinterpreted as a similar song. Percentages of false positive according to threshold values are also shown in Table I. The score of each band can be calculated from right positive minus false positive according to threshold values.

From Table I, threshold value 0.7 gives the maximum score at low band and threshold value 0.5 gives the maximum score at mid band and high band. Therefore, we will use these threshold values in the future.

To combine results from 3 bands, we use a similarity result, a normalized subtraction result that less than the threshold, from each band to vote for a final similarity result, as shown in Table II. From 380 comparisons of similar files, the final similarity results will be considered right positive. From 3420 comparisons of dissimilar files, the final similarity results will be considered false positive. Again, the score of a final similarity result can be calculated from right positive minus false positive. From Table II, The 2 of 3 voting gives the highest score, since the false positive is reduced significantly.

TABLE II. SCORES OF FINAL SIMILARITY RESULTS

Votes from 3 bands	Right positive (%)	False positive (%)	score
1 of 3	99.21	10.85	88.36
2 of 3	93.42	0.38	93.04
3 of 3	79.47	0.00	79.47

V. CONCLUSIONS

Intermodulation between FM broadcast stations creates interference which is mixed sounds from those stations. To identify the interference sources, we need to compare the RIM audio with the audio from each FM broadcast station. In this paper, without using any FFT calculation, the audios are divided into 3 frequency bands, low band, mid band and high band. Simple subtraction between the audios of the same band gives the indications of similarity, called normalized subtraction results. We use the threshold 0.5 for mid band and high band and the threshold 0.7 for low band. When the normalized subtraction result is less than the threshold, it indicates the similarity of the audios.

Score from each band can be calculated from the percentage of right positive finding from similar songs minus percentage of false positive finding from dissimilar songs. There are totally 380 similar songs and 3420 dissimilar songs comparisons. The scores from low band, mid band and high band are 92.96, 78.99 and 74.85 respectively. To get the final similarity result, we find that using 2 of 3 voting will give the maximum score of 93.04 with 93.42 right positive finding and 0.38 false positive.

ACKNOWLEDGMENT

This work was supported by Suranaree University of Technology (SUT), Thailand.

REFERENCES

- [1] David Hand C.Eng MIET, "The RF challenges of ATC communications" Consultant Engineer - Park Air Systems Ltd.
- [2] Haitsma, Jaap and Kalker, Antonius, "A Highly Robust Audio Fingerprinting System," International Symposium on Music Information Retrieval (ISMIR), Eindhoven : s.n., 2002, pp. 107-115.
- [3] P.Cano, E.Battle, T.Kalker, J.Haitsma, "A Review of Algorithms for Audio Fingerprinting", IEEE International workshop on MMSP, 2002.
- [4] Wang, Avery Li-Chun., "An Industrial-Strength Audio Search Algorithm." ISMIR, London : Shazam Entertainment, Ltd., 2003.
- [5] Heinrich A. van Nieuwenhuizen, Willie C. Venter and Leenta M.J. Grobler , "The Study and Implementation of Shazam's Audio Fingerprinting Algorithm for Advertisement Identification", School of Electrical, Electronic and Computer Engineering ,North-West University, Potchefstroom Campus, South Africa
- [6] T.Park,S.K.Beack,T.Lee, "A noise robust audio fingerprint extraction technique for mobile devices using gradient histograms",Consumer

II. AUDIO SIMILARITY COMPARISON

In this paper, we assume no time delay between two audios. Amplitude envelopes of audios are specific data which benefits to audio similarity comparison. First, we extract envelopes from both audios which are used to compare. Simple subtraction of two envelopes will reduce a subtraction result if two envelopes are similar. Otherwise, a subtraction result will be increased. Since this algorithm processes in time domain, therefore it is simple to calculate and uses short computation time.

III. REAL-TIME SIMILARITY COMPARISON ALGORITHM

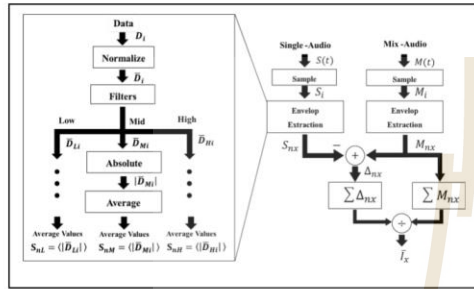


Fig. 3. Normalized subtraction results calculation

From Figure 3, two audio signals present at the inputs, called Single-Audio ($S(t)$) and Mix-Audio ($M(t)$). A continuous signal ($S(t)$) will be sampled to a discrete signal (S_i). We use sampling rate, $F_S = 44,100$ Hz. We use equation (1) to normalize the Data (D_i).

$$\bar{D}_i = \frac{D_i}{\sqrt{\sum (D_i)^2}} \quad (1)$$

Normalized data (\bar{D}_i) will be filtered into 3 frequency bands, low band (below 340 Hz), mid band (340 to 3,400 Hz) and high band (upper 3,400 Hz), called \bar{D}_{ix} . The average values (S_{nx}) are calculated from the absolute normalized data ($|\bar{D}_{ix}|$) of each band by equation (2).

$$S_{nx} = \langle |\bar{D}_{ix}| \rangle = \frac{\sum_{i=1}^k |\bar{D}_{ix}|}{k} ; \quad \begin{matrix} n = 0,1,2,\dots \\ k = w_S \times F_S \end{matrix} \quad (2)$$

Where x is the index of each filter band and w_S is an average window size, e.g., 1 msec. These processes are called Envelop Extraction.

The subtraction results (Δ_{nx}) of each band are calculated from the average values of Mix-Audio (M_{nx}) minus the average values of Single-Audio (S_{nx}) according to each band.

$$\Delta_{nx} = M_{nx} - S_{nx} \quad (3)$$

A normalized subtraction result (\bar{I}_x) of each band is calculated from total subtraction results (Δ_{nx}) divided by total average values of Mix-Audio (M_{nx}).

$$\bar{I}_x = \frac{\sum \Delta_{nx}}{\sum M_{nx}} \quad (4)$$

IV. EXPERIMENT RESULTS

In this section presents experimental results from MATLAB[®] simulation. There are totally 20 songs to test the algorithm. We combine two songs into a file to simulate a RIM audio. There are totally 190 files of the RIM audios. Then, we use 1 of 20 songs to compare similarity with those 190 files. Therefore, each song will have 19 similar files and 171 dissimilar files. For 20 songs, there are totally 380 similar files and 3420 dissimilar files.

The normalized subtraction results of each band from similar files and 3420 dissimilar files are plotted in Figure 4(a) and 4(b) respectively.

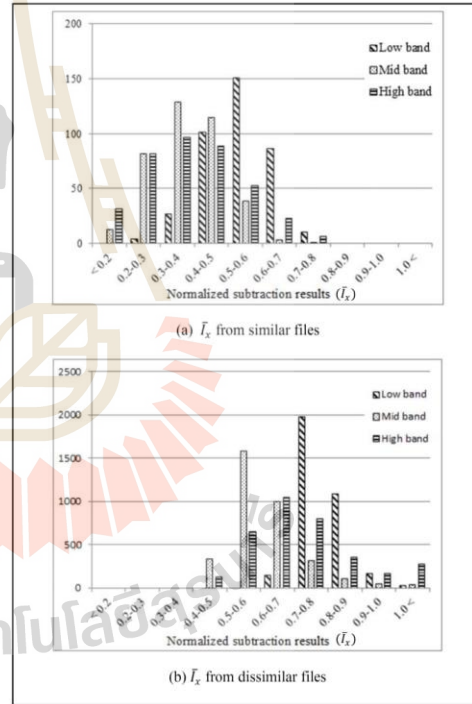
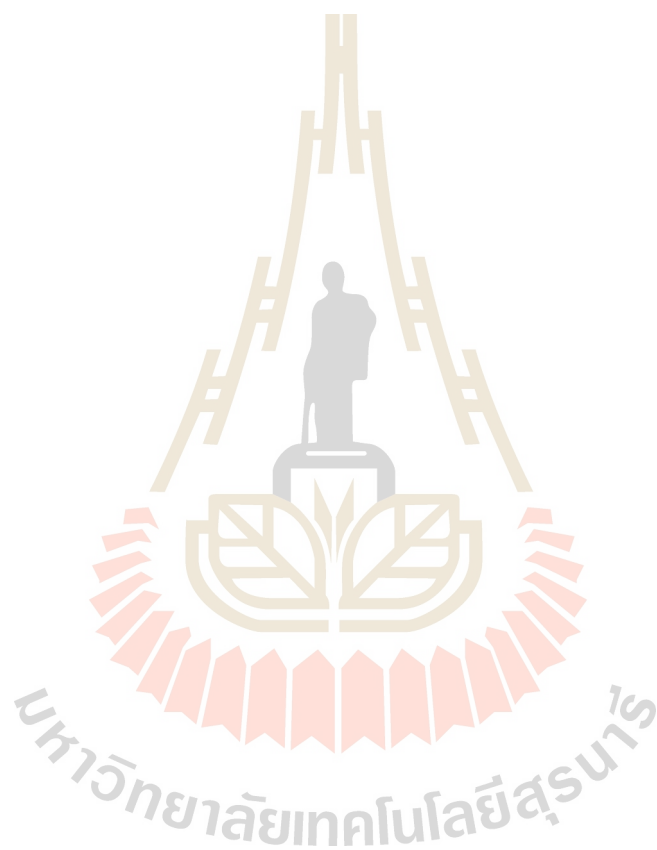


Fig. 4. Histogram of normalized subtraction results of each band

Electronics - Berlin (ICCE-Berlin), 2015 IEEE 5th International Conference on, 6-9 Sept. 2015.

- [7] Yong Fan, Shuang Feng, "A Music Identification System Based on Audio Fingerprint", Applied Computing and Information Technology/3rd Intl Conf on Computational Science/Intelligence and Applied Informatics/1st Intl Conf on Big Data, Cloud Computing, Data Science & Engineering (ACIT-CSII-BCD), 2016 4th Intl Conf on, 12-14 Dec. 2016
- [8] Reinhard Sonnleitner, Gerhard Widmer, "Robust Quad-Based Audio Fingerprinting", IEEE/ACM Transactions on Audio, Speech, and Language Processing, March 2016.
- [9] Shanshan Yao, Baoning Niu, Jianquan Liu, "Audio Identification by Sampling Sub-fingerprints and Counting Matches, IEEE Transactions on Multimedia, Volume: 19, Issue: 9, Sept. 2017.



ประวัติผู้เขียน

ผู้ช่วยศาสตราจารย์ ดร. รัชสรรค์ ทองทา สำเร็จการศึกษาวิศวกรรมไฟฟ้า จากมหาวิทยาลัยขอนแก่น เมื่อปี พ.ศ. 2531 และ MS และ PH.D. จาก Florida Institute of Technology, USA เมื่อปี 1993 และ 1998 ตามลำดับ เริ่มงานในตำแหน่งวิศวกรที่บริษัท Elcom Research เมื่อปี พ.ศ. 2531 และบริษัท การบินไทย จำกัด ในปี พ.ศ. 2532 เมื่อปี พ.ศ. 2533 ได้เริ่มในตำแหน่งอาจารย์ที่ภาควิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยขอนแก่น และมหาวิทยาลัยเทคโนโลยีสุรนารีในปี พ.ศ. 2536

