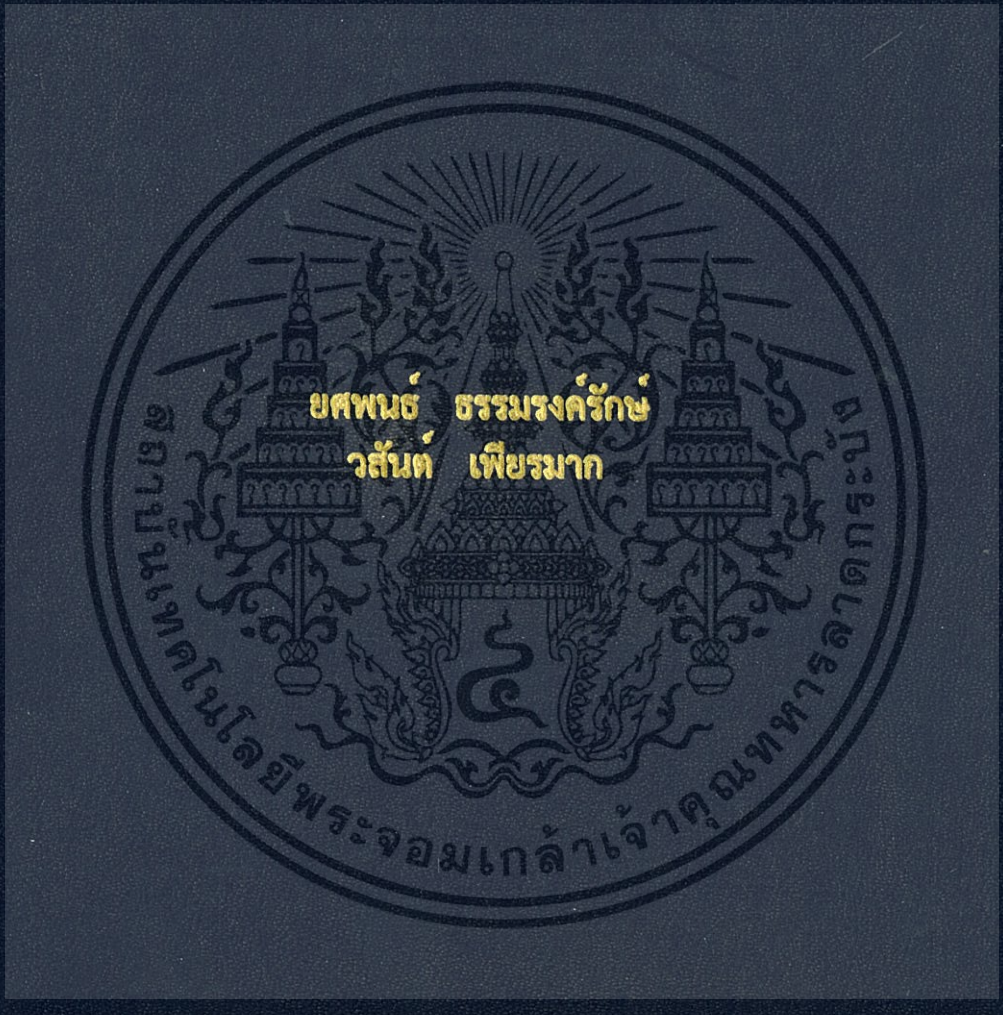
ต้นแบบการสั่งงานด้วยเสียงบนคอมพิวเตอร์ขนาดเล็ก
Prototype of Voice Command on Small Computer



ยศพนธ์ ธรรมรงค์รักษ์
วสันต์ เพียรมาก

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2558

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ต้นแบบการสั่งงานด้วยเสียงบนคอมพิวเตอร์ขนาดเล็ก

Prototype of Voice Command on Small Computer

T149256

ยศพนธ์ ธรรมรงค์รักษ์

วสันต์ เพียรมาก

ว.พ.
ย154 ๐๒
2558
ตขหมู่.....
เลขทะเบียน 149256
วัน,เดือน,ปี 30 ส.ค. 2561
b 12879252
i.....

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2558

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROTOTYPE OF VOICE COMMAND ON SMALL COMPUTER

YOSAPON THAMMARONGRAK
WASAN PIANMAK

A SPECIAL PROBLEM SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR
THE DEGREE OF BACHELOR OF SCIENCE (COMPUTER SCIENCE)
DEPARTMENT OF COMPUTER SCIENCE, FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2015

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	ต้นแบบการสั่งงานด้วยเสียงบนคอมพิวเตอร์ขนาดเล็ก Prototype of Voice Command on Small Computer
ชื่อนักศึกษา	นายยศพนธ์ ธรรมรงค์รักษ์ รหัสนักศึกษา 55050424 นายวสันต์ เพียรมาก รหัสนักศึกษา 55050452
ปริญญา	วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา	วิทยาการคอมพิวเตอร์
ปีการศึกษา	2558
อาจารย์ที่ปรึกษา	ผศ.ธีระ ศิริธีรากุล

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์) ประจำปีการศึกษา 2558

คณะกรรมการสอบ	ลายมือชื่อ
อ.สันธนะ อู่อุดมยิ่ง ประธานกรรมการ	
ดร.สายชล ใจเย็น กรรมการ	
ผศ.ธีระ ศิริธีรากุล กรรมการและอาจารย์ที่ปรึกษา	

ลิขสิทธิ์ของคณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	ต้นแบบการสั่งงานด้วยเสียงบนคอมพิวเตอร์ขนาดเล็ก
ชื่อนักศึกษา	นายยศพนธ์ ธรรมรงค์รักษ์ รหัสนักศึกษา 55050424
	นายวสันต์ เพียรมาก รหัสนักศึกษา 55050452
ปริญญา	วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา	วิทยาการคอมพิวเตอร์
คณะ	วิทยาศาสตร์
มหาวิทยาลัย	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)
ปีการศึกษา	2558
อาจารย์ที่ปรึกษา	ผศ.ธีระ ศิริธีรากุล

บทคัดย่อ

ปัญหาพิเศษนี้ ได้ศึกษาและจัดทำต้นแบบการสั่งงานด้วยเสียงบนคอมพิวเตอร์ขนาดเล็กขึ้น เพื่อแก้ปัญหาของการที่ต้องใช้หน่วยประมวลผลที่มีประสิทธิภาพสูงในการประมวลผลเสียง โดยในโครงการนี้ใช้บอร์ดราสเบอร์รี่พาย และใช้วิธีการสกัดคุณลักษณะของเสียงแบบสัมประสิทธิ์เมลฟรีเควันซีเซปสตรอล จากนั้นผู้จัดทำได้เลือกวิธีการไดนาร์มิคไทม์วาร์ปิง ในการเปรียบเทียบกับคลังเสียงที่ได้ทำการบันทึกไว้ เพราะเป็นวิธีการที่มีความแม่นยำสูง แล้วนำผลลัพธ์ที่ได้ไปใช้ในการตัดสินใจ โดยคำสั่งเสียงที่ใช้จะเป็นแบบเจาะผู้พูด โดยใช้ภาษาไทยในการสั่งงาน

คำสำคัญ : การจำแนกรูปแบบ การสั่งงานด้วยเสียง ไดนาร์มิคไทม์วาร์ปิง ไดนาร์มิคไทม์วาร์ปิงแบบ-เร็ว สัมประสิทธิ์เมลฟรีเควันซีเซปสตรอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	Prototype of Voice Command on Small Computer	
Students	Mr. Yosapon Thammarongrak	Student ID 55050424
	Mr. Wasan Pianmak	Student ID 55050452
Degree	Bachelor of Science (Computer Science)	
Department	Computer Science	
Faculty	Science	
University	King Mongkut's Institute of Technology Ladkrabang (KMITL)	
Academic Year	2015	
Advisor	Asst.Prof.Teera Siriteerakul	

Abstract

This special problem studied and produced Prototype of Voice Command on Small Computer for solve the problem that require high performance processor in voice analysis by this project using Raspberry Pi and the feature extraction by MFCC (Mel-frequency Cepstral Coefficients), and then we choose DTW (Dynamic Time Warping) to compare the feature, bring the result compared to decide and then select the optimal method for do the right command. Using speaker dependent and thai language for command.

Keywords : Pattern Classification , Voice Command , Dynamic time warping (DTW) , FastDTW , Mel-frequency Cepstral Coefficients (MFCC)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปัญหาพิเศษนี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องมาจากความอนุเคราะห์และความร่วมมือจากทุกๆ ท่าน ขอขอบพระคุณ ผศ.ธีระ ศิริธีรากุล ซึ่งเป็นอาจารย์ที่ปรึกษาที่คอยให้ความช่วยเหลือแนะนำที่ดีในการปรับปรุงข้อบกพร่องในการทำปัญหาพิเศษนี้ให้สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบพระคุณอาจารย์ผู้ควบคุมการสอบปัญหาพิเศษ ที่ให้คำแนะนำทำให้ปัญหาพิเศษนี้มีความสมบูรณ์ยิ่งขึ้น

ขอขอบพระคุณ บิดา-มารดา ที่ให้การสนับสนุน ให้รับการศึกษา คอยเลี้ยงดูอบรมสั่งสอนตลอดจนเป็นกำลังใจมาโดยตลอด

ขอขอบพระคุณสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่มอบโอกาสให้ได้เข้าศึกษาในสถาบัน ได้พบคณาจารย์และบุคลากรที่มีศักยภาพในการช่วยพัฒนาทักษะและมอบความรู้ให้แก่นักศึกษา

นอกจากนี้ขอขอบพระคุณทุกท่านที่ไม่ได้กล่าวถึงมา ณ ที่นี้ที่คอยให้คำปรึกษา แนะนำช่วยเหลือและเป็นกำลังใจให้ปัญหาพิเศษนี้สำเร็จลุล่วงไปได้ด้วยดี

ยศพนธ์ ธรรมรงค์รักษ์
วสันต์ เพียรมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	ฉ
สารบัญรูป	ช
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญ	1
1.2 วัตถุประสงค์ของการทำ	1
1.3 ขอบเขตของงานวิจัย	1
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
1.5 ขั้นตอนการดำเนินงาน	2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	3
2.1 การเตรียมข้อมูลเบื้องต้นก่อนการวิเคราะห์เสียง (Preprocessing)	4
2.1.1 การพรีเอมฟาซิส (Pre-emphasis)	4
2.1.2 การแบ่งช่วงสัญญาณ (Frame blocking)	4
2.1.3 การวินโดว์ (Windowing)	4
2.2 การวิเคราะห์เสียงพูด (Speech Analysis)	6
2.2.1 สัมประสิทธิ์การทำนายพันธะเชิงเส้น (LPC)	6
2.2.2 สัมประสิทธิ์เมลฟรีเคว็นซีเซปสตรอล (MFCC)	7
2.3 การจำแนกรูปแบบ (Pattern Classification)	8
2.3.1 แบบไดนาร์มิคไทม์วาร์ปิง (DTW)	8
2.3.2 แบบไดนาร์มิคไทม์วาร์ปิงแบบเร็ว (FastDTW)	9
2.3.3 แบบจำลองฮิดเดน มาร์คอฟ (HMM)	10
บทที่ 3 ขั้นตอนวิธีในการดำเนินงานวิจัย	11
3.1 การเลือกใช้อุปกรณ์	11
3.2 ขั้นตอนการดำเนินการ	13
3.3 การเตรียมข้อมูลเบื้องต้น (Preprocessing)	14
3.4 การวิเคราะห์เสียงพูด (Speech Analysis)	15
3.5 การจำแนกรูปแบบ (Pattern Classification)	15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
บทที่ 4 การทดลองและผลการทดลอง	16
4.1 ข้อกำหนดในการทดลอง	16
4.1.1 ลักษณะของข้อมูลเสียงพูด	16
4.1.2 การประมวลผลเบื้องต้น	16
4.2 วิธีการทดลอง	16
4.2.1 การทดลองหาค่าความต่างของวิธีการ DTW และ FastDTW	16
4.2.2 ทดลองหาค่าความแม่นยำในการสั่งงานจากค่าความต่างที่กำหนด	17
4.3 ผลการทดลอง	17
4.3.1 ผลทดลองหาค่าความต่างของวิธีการ DTW และ FastDTW	17
4.3.1.1 การทดลองหาค่าความต่างของวิธีการ DTW	17
4.3.1.2 การทดลองหาค่าความต่างของวิธีการ FastDTW	19
4.3.1.3 การทดลองหาค่าความแม่นยำในการสั่งงาน	21
บทที่ 5 สรุปผลการทดลอง	22
เอกสารอ้างอิง	23
ภาคผนวก	25
Source Code ของโปรแกรม	26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่		หน้า
4.1	แสดงค่าความต่างของเสียงของวิธี DTW	18
4.2	แสดงค่าเฉลี่ยของค่าความต่างของเสียงของวิธี DTW	18
4.3	แสดงค่าเฉลี่ยและส่วนเบี่ยงเบนมาตรฐานของค่าความต่างของเสียงของวิธี DTW	19
4.4	แสดงค่าความต่างของเสียงของวิธี FastDTW	19
4.5	แสดงค่าเฉลี่ยของค่าความต่างของเสียงของวิธี FastDTW	20
4.6	แสดงค่าเฉลี่ยและส่วนเบี่ยงเบนมาตรฐานของค่าความต่างของเสียงของวิธี FastDTW	20
4.7	แสดงผลการทดลองสั่งงานด้วยเสียง	21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
2.1	ขั้นตอนหลักในการดำเนินการรู้จำเสียง	3
2.2	แสดงไดอะแกรมขั้นตอนการทำงานของ LPC	7
2.3	แสดงไดอะแกรมขั้นตอนการทำงานของ MFCC	7
2.4	แสดงแนวคิดของการหาค่าความห่างของ DTW	9
2.5	แสดงอัลกอริทึม Sakoe-Chiba Band(ซ้าย) และ Itakura Parallelogram(ขวา)	9
2.6	แสดงการลดจำนวนตัวแทนของ FastDTW ทำให้เส้นทางมีการเปลี่ยนแปลงเล็กน้อย	10
2.7	แสดงถึงการทำงานของ FastDTW (radius = 1)	10
3.1	Raspberry Pi 2 Model B	11
3.2	Channel USB External Sound Card Audio Adapter	12
3.3	ไมโครโฟน NUBWO (NUB111)	12
3.4	แสดงขั้นตอนการทำงานทั้งหมดของระบบ	13
3.5	ก่อนการทำ Hamming window และหลังจากทำ Hamming window	14
4.1	แสดงภาพหน้าจอใช้งานโปรแกรม	21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญ

เนื่องจากปัจจุบันการสั่งงานด้วยเสียงนั้นมีการใช้งานที่หลายหลากเช่น Siri ที่ใช้ในการสั่งงานอุปกรณ์คอมพิวเตอร์ต่างๆที่ใช้ระบบปฏิบัติการ iOS หรือแม้กระทั่ง GPS ในรถยนต์ เป็นต้น แต่การจะประมวลผลได้นั้นต้องอาศัยหน่วยประมวลที่มีประสิทธิภาพสูง และมีการใช้งานโดยเรียก API ของค่ายต่างๆที่เปิดให้ใช้งานเช่น Google Voice API เป็นต้น จึงเป็นการยากที่จะใช้การสั่งงานด้วยเสียงบนคอมพิวเตอร์ขนาดเล็ก

ดังนั้นปัญหาพิเศษนี้จึงได้ทำต้นแบบการสั่งงานด้วยเสียงบนคอมพิวเตอร์ขนาดเล็กขึ้นเพื่อแก้ปัญหาของการที่ต้องใช้หน่วยประมวลผลที่มีประสิทธิภาพสูง โดยในปัญหาพิเศษนี้ใช้บอร์ด Raspberry Pi และใช้วิธีการสกัดคุณลักษณะของเสียงแบบ MFCC (Mel-frequency cepstral coefficients) แล้วนำข้อมูลที่ได้ไปเปรียบเทียบกับคลังเสียงที่ได้ทำการบันทึกไว้แล้วด้วยวิธีการ DTW (Dynamic time warping) จากนั้นนำผลลัพธ์ที่ได้ไปตัดสินใจเพื่อทำตามคำสั่ง ซึ่งคำสั่งเสียงที่ใช้ในการสั่งงานคือภาษาไทย

การทำต้นแบบการสั่งงานด้วยเสียงบนคอมพิวเตอร์ขนาดเล็กนั้น จะช่วยให้คอมพิวเตอร์ขนาดเล็กต่างๆสามารถที่จะใช้การสั่งงานด้วยเสียงได้โดยที่ไม่ต้องใช้ API หรือการประมวลผลที่สูง

1.2 วัตถุประสงค์ของการทำ

เพื่อสร้างต้นแบบการสั่งงานด้วยเสียงบนคอมพิวเตอร์ขนาดเล็ก

1.3 ขอบเขต

สามารถมีผู้ใช้งานได้เพียง 1 คน

สั่งงานด้วยเสียงภาษาไทย

มีคำสั่งต้นแบบ 4 คำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 สามารถสั่งงานคอมพิวเตอร์ขนาดเล็กด้วยเสียงได้โดยไม่ต้องใช้ API

1.4.2 สามารถนำต้นแบบนี้ไปประยุกต์ใช้ในอุปกรณ์คอมพิวเตอร์ต่างๆ

1.5 ขั้นตอนการดำเนินงาน

1.5.1 ศึกษาหลักการทำงานของ Voice Recognition

1.5.2 ศึกษาภาษา Python

1.5.3 ศึกษาการใช้งานบอร์ด Raspberry Pi

1.5.4 ออกแบบต้นแบบการสั่งงานด้วยเสียงบน Raspberry Pi

1.5.5 พัฒนาต้นแบบการสั่งงานด้วยเสียงบน Raspberry Pi โดยใช้ภาษา Python

1.5.6 ทดสอบต้นแบบการสั่งงานด้วยเสียงบน Raspberry Pi

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ระบบการสั่งงานด้วยเสียงที่ใช้ในปัญหาพิเศษนี้มีวิธีการใกล้เคียงกับระบบการรู้จำเสียงเป็นอย่างมาก ทางผู้จัดทำจึงได้ประยุกต์ระบบการรู้จำเสียงเพื่อนำมาใช้ในปัญหาพิเศษนี้

ในปัจจุบัน สามารถแบ่งวิธีวิเคราะห์เสียงพูดได้เป็น 4 ประเภทใหญ่ๆ ตามลักษณะออกเสียง คือ

1. วิเคราะห์เฉพาะผู้พูดที่เจาะจงและเป็นคำโดด (Speaker Dependent Isolated Word Recognition)

2. วิเคราะห์เฉพาะผู้พูดที่เจาะจงและเป็นคำต่อเนื่อง (Speaker Dependent Continuous Word Recognition)

3. วิเคราะห์แบบไม่เจาะจงผู้พูดและเป็นคำโดด (Speaker Independent Isolated Word Recognition)

4. วิเคราะห์แบบไม่เจาะจงผู้พูดและเป็นคำต่อเนื่อง (Speaker Independent Continuous Word Recognition)

จากการแบ่งวิธีวิเคราะห์เสียงพูดข้างต้นคือ วิเคราะห์เฉพาะผู้พูดที่เจาะจงและเป็นคำต่อเนื่อง นั้นมีความเหมาะสมกับระบบการสั่งงานด้วยเสียงที่ทางผู้จัดทำต้องการ ซึ่งในการดำเนินงานนั้นเป็นไปตามโครงสร้างดังรูปที่ 2.1 จะประกอบไปด้วยขั้นตอนการดำเนินงาน 3 ขั้นตอนหลัก

1.การเตรียมข้อมูลเบื้องต้นก่อนการวิเคราะห์เสียง (Preprocessing)

2.การวิเคราะห์เสียงพูด (Speech Analysis)

3.การจำแนกรูปแบบ (Pattern Classification)

รูปที่ 2.1 ขั้นตอนหลักในการดำเนินการรู้จำเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1 การเตรียมข้อมูลเบื้องต้นก่อนการวิเคราะห์เสียง (Preprocessing)

การเตรียมข้อมูลเบื้องต้นเป็นการจัดเตรียมข้อมูลดิบของเสียงพูดที่ได้จากการบันทึกเสียง ซึ่งโดยทั่วไปจะมีสัญญาณรบกวน โดยขั้นตอนนี้จะเป็นการลดสัญญาณรบกวนและเตรียมข้อมูลเบื้องต้น เพื่อใช้ในการประมวลผลในขั้นตอนต่อไป เนื่องจากสัญญาณเสียงพูดโดยรวมจะแปรเปลี่ยนตามเวลา (Time Variant) และไม่เสถียร (Unstable) ดังนั้นในการประมวลผลสัญญาณเชิงเลขกับเสียงพูด จะแบ่งสัญญาณเสียงพูดออกเป็นช่วงย่อยๆ โดยแต่ละช่วงของสัญญาณเสียงจะมีความยาวประมาณ 10-40 มิลลิวินาที ซึ่งถือได้ว่าสัญญาณเสียงในแต่ละช่วงมีความเสถียร (Stable) และไม่แปรเปลี่ยนตามเวลา (Time Invariant) จากนั้นจึงสามารถทำการประมวลผลสัญญาณเชิงเลขกับสัญญาณเสียงในแต่ละช่วงได้

2.1.1 การพรีเอมฟาซิส (Pre-emphasis)

องค์ประกอบส่วนใหญ่ของสัญญาณเสียงพูดจะอยู่ที่บริเวณช่วงความถี่ต่ำ เมื่อเทียบกับแถบความถี่ (bandwidth) ไม่เกิน 4 kHz จึงทำให้ช่วงบริเวณความถี่สูงจะมีอัตราส่วนสัญญาณเสียงต่อสัญญาณรบกวน (Signal to Noise Ratio : SNR) สูง เพื่อให้อัตราส่วนสัญญาณเสียงต่อสัญญาณรบกวนมีค่าค่อนข้างคงที่ตลอดช่วงแถบความถี่ทั้งหมด จึงต้องมีการพรีเอมฟาซิส (Pre-emphasis) ก็คือการกรองสัญญาณด้วยวงจรกรองแบบผ่านความถี่สูงซึ่งมักนิยมใช้วงจรกรองอันดับหนึ่ง (First Order Digital Filters) ตามสมการที่ 2.1 ค่า a อยู่ในช่วงระหว่าง 0.9 ถึง 1 ค่าที่นิยมใช้คือ 0.95 เป็นการเน้นให้สัญญาณช่วงความถี่สูงมีขนาดสูงขึ้น

$$s'(n) = s(n) - a * s(n-1) \tag{2.1}$$

เมื่อ a เป็นสัมประสิทธิ์ตัวกรอง

s'(n) เป็นค่าของสัญญาณเสียงพูดขาออกที่ผ่านวิธีพรีเอมฟาซิส

s(n) เป็นค่าของสัญญาณเสียงพูดขาเข้า

2.1.2 การแบ่งช่วงสัญญาณ (Frame blocking)

เป็นขั้นตอนในการแบ่งสัญญาณเสียงเป็นส่วนย่อยขนาดความยาวประมาณ 10 – 40 มิลลิวินาทีซึ่งทำให้สัญญาณเสียงมีคุณสมบัติเปลี่ยนแปลงตามเวลาน้อยมาก หรือ ไม่มีเลย เพื่อให้สามารถสร้างแบบจำลองการกระจายของหน่วยสัญญาณเสียงย่อยทางสถิติได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 การวินโดว์ (Windowing)

โดยปกติแล้วการประยุกต์ส่วนมากจะใช้กรอบของสัญญาณที่มีช่วงกว้างกว่าช่องของข้อมูลที่คงที่และเน้นเฉพาะช่องตรงกลางของกรอบให้เป็นส่วนของข้อมูลที่ใช้ในการวิเคราะห์เช่น ข้อมูลสัญญาณเสียงที่มีลักษณะคงที่ในช่วงเวลา 10 มิลลิวินาที ก็อาจจะใช้ช่องแคบขนาด 20 มิลลิวินาที โดยช่วงกึ่งกลางช่องแคบ ซึ่งรูปแบบของกรอบมีอยู่ด้วยกันหลายลักษณะ เช่น Blackman Windows : BMW, Barlett Windows : BLW, Hamming Windows : HMW เป็นต้น โดยฟังก์ชั่นกรอบที่นิยมใช้กันมากในการวิเคราะห์สัญญาณเสียงคือ Hamming Windows : w(n) ซึ่งจะมีรูปร่างตามลักษณะของ Cosine Pulse โดยจะมีนิยามดังสมการที่ 2.2

$$w(n) = (1-\alpha) - \alpha\cos\left(\frac{2\pi n}{N-1}\right) \tag{2.2}$$

w(n) เป็นฟังก์ชั่นหน้าต่าง

n เป็นลำดับข้อมูลในกรอบสัญญาณเสียงพูด โดยที่ n = 0, 1, ..., N-1

N เป็นจำนวนข้อมูลทั้งหมดในกรอบสัญญาณเสียงพูด

ในปัญหาพิเศษนี้ได้เลือกใช้ฟังก์ชั่นกรอบชนิด Hamming Window ซึ่งจะมีผลทำให้เกิดการลดทอนแอมปลิจูดอย่างช้าๆ ที่บริเวณปลายแต่ละข้างของกรอบสัญญาณเสียงพูด เพื่อป้องกันการเปลี่ยนแปลงที่ไม่ต่อเนื่องตรงบริวณจุดปลาย โดยสมการที่ 2.2 จะแสดงค่าฟังก์ชั่นกรอบ w(n) และสมการที่ 2.3 จะแสดงค่าสัญญาณเสียงพูดที่ผ่านกรรมวิธีการวางกรอบ

$$\tilde{x}_l(n) = x_l(n)w(n) \tag{2.3}$$

w(n) เป็นฟังก์ชั่นหน้าต่างซึ่งจะทำการลดความไม่ต่อเนื่องของเฟรมที่ l ตัวอย่าง

l เป็นลำดับของกรอบสัญญาณเสียงพูด โดยที่ l = 0, 1, ... , L-1

L เป็นจำนวนของกรอบสัญญาณเสียงพูด

$x_l(n)$ เป็นค่าสัญญาณเสียงพูดของข้อมูลที่ n

$\tilde{x}_l(n)$ เป็นค่าสัญญาณเสียงพูดที่ผ่านกรรมวิธีการวางกรอบ โดยสัญญาณเสียงที่ได้จะมีความยาวเท่ากับจำนวนของกรอบสัญญาณเสียงชุดและมีหน่วยเป็น L เฟรม

n เป็นจำนวนตัวอย่างที่อยู่ในแต่ละเฟรม โดยที่ n = 0, 1, ... , N-1

N เป็นจำนวนตัวอย่างทั้งหมดใน 1 เฟรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การวิเคราะห์เสียงพูด (Speech Analysis)

เป็นเทคนิคการลดจำนวนข้อมูล โดยที่จากข้อมูลจำนวนมากจะถูกแปลงเป็นชุดข้อมูลที่มีจำนวนน้อยลง และยังคงแสดงคุณสมบัติสำคัญของรูปคลื่นสัญญาณเสียงได้อย่างถูกต้อง ซึ่งการวิเคราะห์หาค่าที่ใช้แทนสัญญาณเสียง แบ่งได้เป็น 3 กลุ่มหลัก

กลุ่มแรกเป็นค่าลักษณะสำคัญระดับสูง (High Level Feature) ได้แก่ สำเนียงการพูด รูปแบบในการพูด และความเร็วในการพูด เป็นต้น

กลุ่มที่สอง จะใช้ค่าลักษณะสำคัญทางฉันทลักษณ์ (Prosodic Feature) เช่น ค่าความถี่มูลฐาน (Fundamental Frequency) ความถี่ฟอร์แมนท์ (Formant Frequency) และระดับพลังงาน (Energy Profile) เป็นต้น ถึงแม้ว่าค่าลักษณะสำคัญแบบนี้จะมีประสิทธิภาพสูงในการรู้จำ แต่ยากในการสกัดจากสัญญาณ

กลุ่มสุดท้ายเรียกว่าค่าลักษณะสำคัญแบบเอนเวโลปของสเปกตรัม (Spectral Envelop Feature) เป็นกลุ่มที่ใช้ในปัญหาพิเศษนี้ เนื่องจากค่าลักษณะสำคัญส่วนใหญ่จะรวมอยู่ในข้อมูลเชิงสเปกตรัมนี้ อีกทั้งยังง่ายและสะดวกในการคำนวณหาค่า ซึ่งในกลุ่มนี้มีวิธีการที่นิยมใช้บ่อยคือ LPC และ MFCC

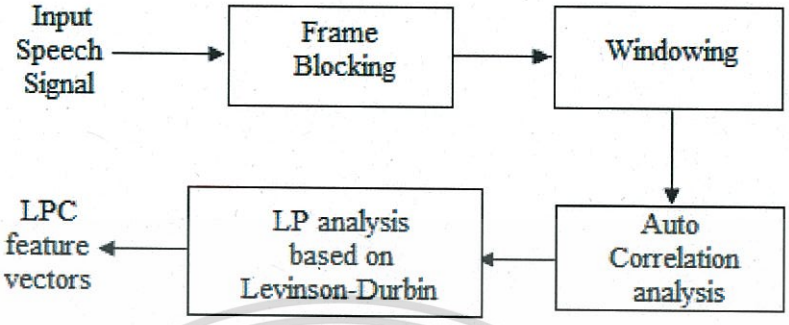
โดยในปัญหาพิเศษนี้ได้เลือกใช้ MFCC เนื่องจากเสียงของมนุษย์นั้นไม่เชิงเส้น แต่พารามิเตอร์ของ LPC นั้นเป็นเชิงเส้น ดังนั้น LPC จึงไม่ใช่ทางเลือกที่ดีนัก ส่วน MFCC นั้นใกล้เคียงระบบการได้ยินของมนุษย์จึงเหมาะสมมากกว่า

2.2.1 สัมประสิทธิ์การทำนายพันธะเชิงเส้น (Linear Predictive Coefficient : LPC)

การประมาณพันธะเชิงเส้น (Linear Predictive : LP) เป็นเทคนิคที่นิยมนำมาใช้ในการวิเคราะห์คุณสมบัติทางกายภาพสัญญาณ เช่น Spectrum Magnitude ของสัญญาณ เทคนิคนี้พิจารณาได้อย่างรวดเร็วและมีประสิทธิภาพ ซึ่งก็ใช้ได้ดีสำหรับสัญญาณเสียงพูดและนำมาประยุกต์ใช้กับระบบรู้จำเสียงพูดได้เป็นอย่างดี การประมาณพันธะเชิงเส้นเป็นวิธีการที่ถูกพัฒนาขึ้นเพื่อลดอัตราการส่งสัญญาณดิจิตอลลงอย่างมาก กล่าวคือ การวิเคราะห์สัญญาณเสียงที่ผ่านเข้ามาเพื่อหาสัมประสิทธิ์ตัวทำนายไปเปรียบเทียบกับสัญญาณจริง เพื่อจะนำค่าความผิดพลาดนั้นไปเข้ากระบวนการเข้ารหัส พร้อมๆกับชุดสัมประสิทธิ์ที่ได้วิเคราะห์ แล้วจึงทำการเข้ารหัสเพื่อส่งออกไป โดยค่าสัมประสิทธิ์ของการประมาณพันธะเชิงเส้นหาได้จากค่าสัมประสิทธิ์อัตสหสัมพันธ์โดยการใช้วิธีของ Levinson Durbin Theory : LDT และแปลงค่าที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นสัมประสิทธิ์เซปตรัม ซึ่งเป็นค่าพารามิเตอร์ที่มีลักษณะน่าเชื่อถือได้ดีกว่าและมีความสัมพันธ์ใกล้ชิดกับการรู้จำเสียงตามความรู้สึกของมนุษย์โดยแท้จริง

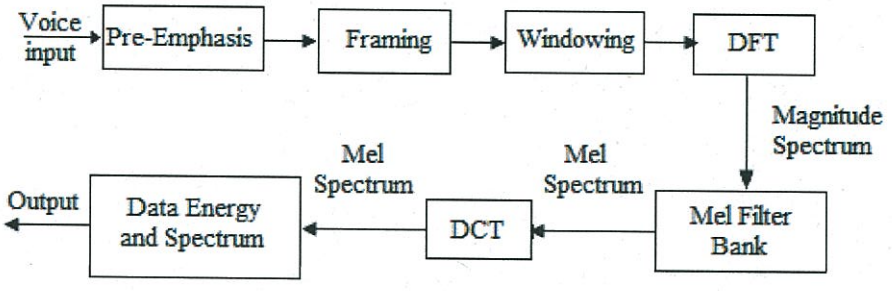


รูปที่ 2.2 แสดงไดอะแกรมขั้นตอนการทำงานของ LPC

2.2.2 สัมประสิทธิ์เมลฟรีเคว็นซีเซปสตรอล (Mel Frequency Cepstral Coefficient : MFCC)

สัมประสิทธิ์เมลฟรีเคว็นซีเซปสตรอล เป็นลักษณะเด่นที่ปรับปรุงมาจากสัมประสิทธิ์เซปสตรัมปกติ ด้วยการปรับสเกลของสเปกตรัมให้อยู่บนสเกลที่เหมาะสมสำหรับการรับฟังของมนุษย์ เนื่องจากสัญญาณเสียงพูดในช่วงความถี่ต่ำ จะมีนัยสำคัญมากกว่าสัญญาณเสียงพูดในช่วงความถี่สูง จึงมีการออกแบบสเกลของสเปกตรัมให้สามารถเก็บรายละเอียดของสัญญาณเสียงช่วงความถี่ต่ำได้มากกว่า เรียกว่า สเกลเมล (Mel Scale)

สัมประสิทธิ์เมลฟรีเคว็นซีเซปสตรอล คำนวณได้จากการวิเคราะห์โดยไม่ใช้พารามิเตอร์ (Non-Parametric Analysis) คือ ไม่มีการสร้างแบบจำลองสัญญาณเพื่อปรับค่าพารามิเตอร์ในแบบจำลองให้ได้สัญญาณที่เราต้องการ แต่เป็นการนำสัญญาณผ่านวงจรกรองแบบผ่านแถบความถี่ (Band Pass Filter) หลายวงจร โดยแต่ละวงจรกรองมีช่วงความกี่ที่ผ่านได้แตกต่างกัน โดยวงจรกรองที่ใช้เป็นชุดวงจรกรองแบบดิจิตอล (Digital Filter Bank) ที่ใช้วิเคราะห์สเปกตรัมของสัญญาณเสียงพูด โดยเลียนแบบตามการได้ยินของมนุษย์



รูปที่ 2.3 แสดงไดอะแกรมขั้นตอนการทำงานของ MFCC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 การจำแนกรูปแบบ (Pattern Classification)

การจำแนกรูปแบบประกอบด้วย 2 หน้าที่หลัก คือการนำเวกเตอร์ของค่าลักษณะสำคัญของสัญญาณเสียงที่อยู่ในชุดอ้างอิงหรือชุดฝึกฝน มาทำการเรียนรู้ เมื่อเรียนรู้แล้วเวกเตอร์ของสัญญาณเสียงที่ต้องการทดสอบจะถูกนำมาเทียบเคียง ขั้นตอนในการเรียนรู้นั้นขึ้นอยู่กับวิธีในการรู้จำของระบบนั้นๆ บางวิธีก็เพียงแค่เก็บข้อมูลชุดเรียนรู้ไว้เปรียบเทียบกับข้อมูลชุดทดสอบ

2.3.1 แบบไดนาร์มิคไทม์วาร์ปิง (Dynamic Time Warping : DTW)

การทำงานของ DTW จะเป็นการเปรียบเทียบสัญญาณอินพุต 2 สัญญาณ เพื่อหาระยะห่างระหว่างสัญญาณอ้างอิงกับสัญญาณทดสอบที่นำมาเปรียบเทียบ โดยกำหนดให้ลำดับของเวกเตอร์ 2 เวกเตอร์คือ เวกเตอร์ A และ เวกเตอร์ B ดังสมการ 2.4 การเปรียบเทียบระยะห่างระหว่างเวกเตอร์ A และเวกเตอร์ B แสดงในสมการที่ 2.5 เมื่อ d(i,j) เป็นระยะห่างระหว่างคู่จุดที่จุด i ของเฟรม A และ j ของเฟรม B โดยที่ a_{in} เป็นค่าลำดับที่ i ของเวกเตอร์ A และ b_{jn} เป็นค่าลำดับที่ j ของเวกเตอร์ B และ K เป็นจำนวนของฟีเจอร์ของเวกเตอร์ ซึ่งค่าที่จะได้จะเป็นค่าระยะทางที่มีค่าน้อยที่สุด

$$A = a_1, a_2, \dots, a_i \qquad (2.4)$$

$$B = b_1, b_2, \dots, b_j$$

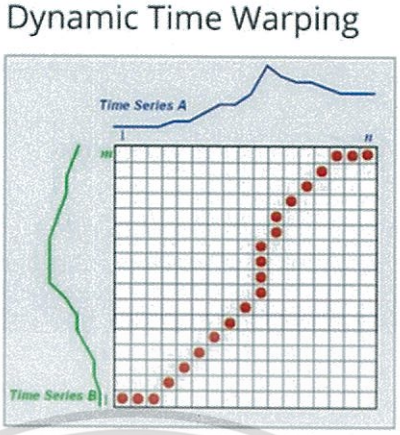
$$d(i,j) = \sum_{n=1}^{k} (a_{in} - b_{jn})^2 \qquad (2.5)$$

อัลกอริทึมการเปรียบเทียบรูปแบบด้วยวิธี DTW ซึ่งจะประกอบด้วยการคำนวณ 2 ส่วนคือ

1. Local Distance เป็นส่วนที่คำนวณหาระยะห่างของข้อมูลจากค่าผลรวมของผลต่างยกกำลังสองของสัมประสิทธิ์ LPC ของข้อมูลทั้งสอง จำนวน p ค่า ซึ่งหาทีละคู่อันดับ

2. Accumulated Distance เป็นส่วนที่ทำหน้าที่บวกสะสมเพื่อหาระยะห่างรวมที่น้อยที่สุดระหว่างข้อมูลสองชุดดังกล่าว แล้วทำการระบุให้เป็นเสียงที่มีรูปแบบใกล้เคียงกับชุดทดสอบนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

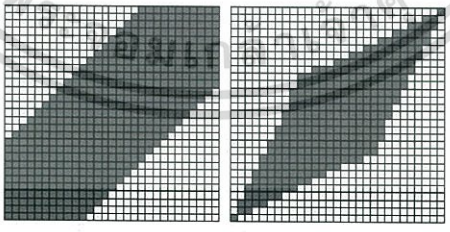


รูปที่ 2.4 แสดงแนวคิดของการหาค่าความห่างของ DTW

2.3.2 แบบไดนาร์มิคไทม์วาร์ปิงแบบเร็ว (Fast Dynamic Time Warping : FastDTW)

เนื่องจากวิธีการ DTW ($O(n^2)$) ใช้เวลาในการประมวลผลที่สูง จึงได้เกิดวิธีการ FastDTW ($O(n)$) ซึ่งมีความเร็วในการประมวลผลที่มากกว่า DTW แบบปกติ โดย FastDTW จะลดจำนวนตัวแทนของข้อมูลลง เพื่อให้เวลาที่ใช้ประมวลผลลดลง แต่ค่าความแม่นยำก็ลดลงด้วย หลักการทำงานของ FastDTW มี 3 วิธีหลักๆดังนี้

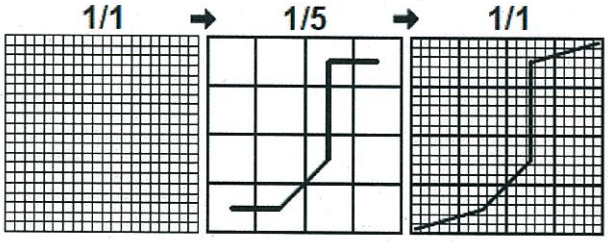
1. Constraints ปกติใน cost matrix ต้องมีการเติมเต็มค่าให้กับทุกๆช่องในเมทริกซ์ แต่ใน FastDTW นั้นได้ใช้อัลกอริทึมของ Sakoe-Chiba Band และ Itakura Parallelogram ในการบังคับให้มีการเติมค่าแคในช่องที่กำหนด แต่แค่วิธีการนี้ก็ยังไม่เพียงพอที่จะทำให้เวลาในการประมวลผลลดลง



รูปที่ 2.5 แสดงอัลกอริทึม Sakoe-Chiba Band(ซ้าย) และ Itakura Parallelogram(ขวา)

2. Data Abstraction ทำการลดจำนวนของตัวแทนข้อมูลที่จะนำมาทำการคำนวณหาเส้นทาง โดยการลดความละเอียดของข้อมูลลง เช่น จากรูปที่ 2.6 จะเห็นได้ว่ามีการลดความละเอียดจาก 1:1 เป็น 1:5 เพื่อทำการหาเส้นทาง จากนั้นทำการแมพกับไปยังความละเอียดปกติ แต่จากอัลกอริทึมนี้จะทำให้ความแม่นยำลดน้อยลงเนื่องจากในตอนที่ทำการแมพกลับจาก 1:5 ไป 1:1 นั้น จะไม่ละเอียดพอที่จะหาเส้นทางที่น้อยที่สุดได้

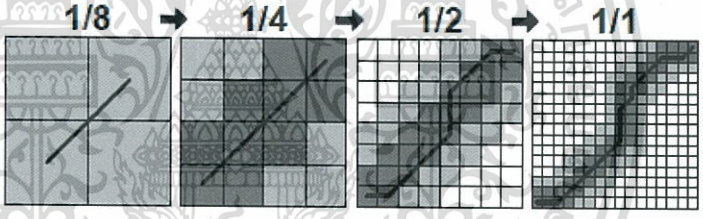
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 แสดงการลดจำนวนตัวแทนของ FastDTW ทำให้เส้นทางมีการเปลี่ยนแปลงเล็กน้อย

3. Indexing เป็นการกำหนดขอบต่ำสุดเพื่อลดจำนวนรอบของ DTW ที่จะต้องทำการประมวลผล

FastDTW นั้นได้ใช้วิธีการในข้อที่ 1 และ 2 ร่วมกันเพื่อทำให้เวลาในการประมวลผลลดลงจาก n^2 เหลือแค่ n และใน FastDTW นั้น จะมีพารามิเตอร์ที่เรียกว่า radius ซึ่งสามารถกำหนดได้ โดยที่ค่าของ radius นั้นส่งผลทั้งความเร็วในการประมวลผลและความแม่นยำ



รูปที่ 2.7 แสดงถึงการทำงานของ FastDTW (radius = 1)

2.3.3 แบบจำลองฮิดเดน มาร์คอฟ (Hidden Markov Model : HMM)

แบบจำลองฮิดเดน มาร์คอฟ ถือเป็นขั้นตอนวิธีการจำแนกรูปแบบที่ดีวิธีการหนึ่งที่มีอยู่ในขณะนี้ โดยอาศัยวิธีการทางสถิติ ขั้นตอนวิธีการนี้มีข้อได้เปรียบที่สำคัญเหนือวิธีการเข้าคู่ต้นแบบก็คือ สามารถเก็บข้อมูลรายละเอียดในทางสถิติเกี่ยวกับเสียงพูดไว้ได้มากกว่าวิธีการเข้าคู่ต้นแบบ โดยเก็บข้อมูลการกระจายที่สมบูรณ์ของลักษณะสำคัญที่มีอยู่ในข้อมูลฝึกฝน จึงสามารถจำแนกความแตกต่างระหว่างเสียงพูดได้ดีมากยิ่งขึ้น อีกทั้งขั้นตอนวิธีการนี้ยังอาศัยการโปรแกรมแบบพลวัต (Dynamic Programming) ทำให้มีความรวดเร็วในการประมวลผลมากยิ่งขึ้น

ในปัญหาพิเศษนี้ได้เลือกใช้ อัลกอริทึมแบบไดนาร์มิคไทม์วาร์ปิง (Dynamic Time Warping : DTW) และ อัลกอรึทึมแบบไดนาร์มิคไทม์วาร์ปิงแบบเร็ว (Fast Dynamic Time Warping : FastDTW) ในการทดสอบเพื่อเลือกวิธีที่เหมาะสม

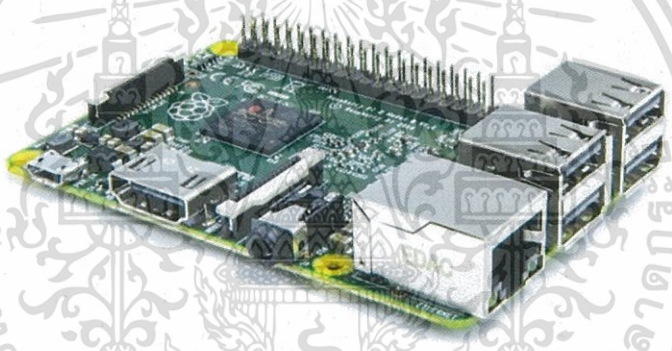
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ขั้นตอนวิธีในการดำเนินงานวิจัย

3.1 การเลือกใช้อุปกรณ์

3.1.1 Raspberry Pi 2 Model B

ทางผู้จัดทำได้เลือกใช้ Raspberry Pi 2 Model B เพราะเนื่องจากเป็นบอร์ดที่เหมาะกับการนำมาวิเคราะห์เสียง รองรับการทำระบบรู้จำด้วยเสียง และมีไลบรารี่หลากหลายให้เลือกใช้



รูปที่ 3.1 Raspberry Pi 2 Model B

คุณสมบัติของ Raspberry Pi 2 Model B

- Broadcom BCM2836 Arm7 Quad Core Processor powered Single Board Computer ความเร็วการทำงานอยู่ที่ 900MHz
- หน่วยความจำ (RAM) 1 GB
- 40 พิน โดยที่มีการเพิ่ม GPIO แล้ว
- USB เวอร์ชั่น 2.0 4 พอร์ท
- 4 pole Stereo output and Composite video port
- HDMI แบบขนาดเต็ม
- มีพอร์ทรองรับการต่อกล้องของ Raspberry Pi
- มีพอร์ทรองรับการต่อหน้าจอแบบสัมผัสของ Raspberry Pi
- มีช่องสำหรับใส่การ์ด Micro SD สำหรับใส่ระบบปฏิบัติการและเก็บข้อมูล
- ช่องสำหรับเสียบเพื่อชาร์จไฟด้วย Micro USB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 7.1 Channel USB External Sound Card Audio Adapter

ทางผู้จัดทำได้เลือกนำมาใช้เนื่องจากจะต้องมีการติดต่อกับไมโครโฟนเพื่อที่จะได้ทำการบันทึกเสียงพูดและนำไปใช้ จึงเลือกใช้เป็นตัวกลางระหว่างไมโครโฟนกับบอร์ดเพื่อที่จะสามารถที่ติดต่อสื่อสารกันได้

รูปที่ 3.2 Channel USB External Sound Card Audio Adapter

3.1.3 ไมโครโฟน NUBWO (NUB111)

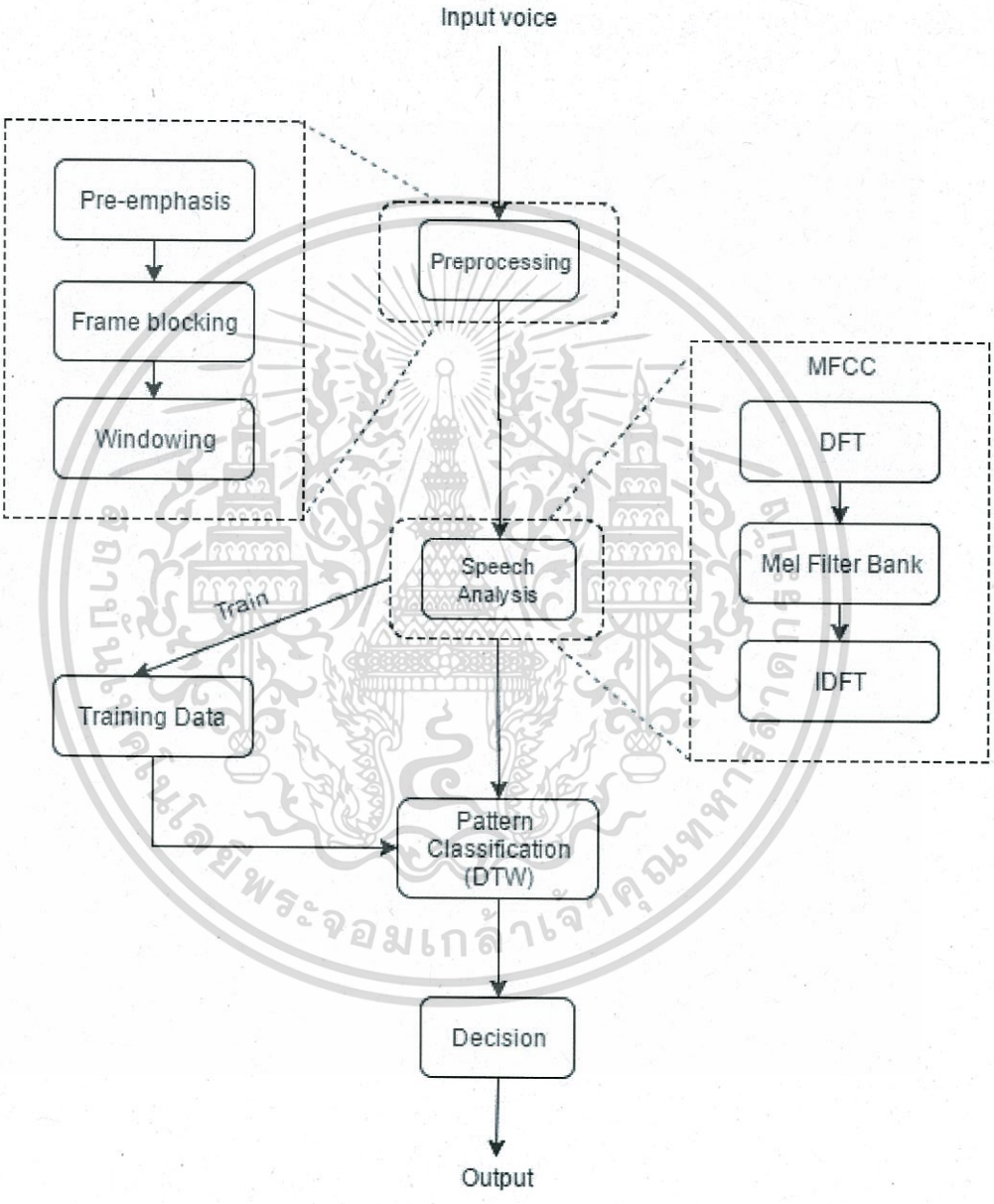
ผู้จัดทำได้เลือกใช้ไมโครโฟนดังกล่าวนี้เพราะมีราคาถูกและมีประสิทธิภาพเหมาะสมสำหรับการใช้งานในปัญหาพิเศษนี้

รูปที่ 3.3 ไมโครโฟน NUBWO (NUB111)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ขั้นตอนการดำเนินการ

ในส่วนนี้จะเป็นการนำเสนอวิธีการที่ใช้ในการพัฒนาระบบสั่งงานด้วยเสียง ซึ่งขั้นตอนการดำเนินการเป็นไปดังรูปที่ 3.4



รูปที่ 3.4 แสดงขั้นตอนการทำงานทั้งหมดของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การเตรียมข้อมูลเบื้องต้น (Preprocessing)

หลังจากรับเสียงเข้ามาจากไมโครโฟนแล้ว จะต้องผ่านขั้นตอนการประมวลผลเบื้องต้น (Preprocessing) ซึ่งจะต้องผ่านขั้นตอนนี้ก่อนทุกครั้ง เนื่องจากเราต้องทำการวิเคราะห์เสียงเป็นช่วงเวลาสั้นๆ เพื่อให้ข้อมูลเสียงพูดที่จะทำการวิเคราะห์ช่วงนั้นมีความเสถียรและไม่แปรเปลี่ยนตามเวลา

3.3.1 การพรีเอมฟาซิส (Pre-emphasis)

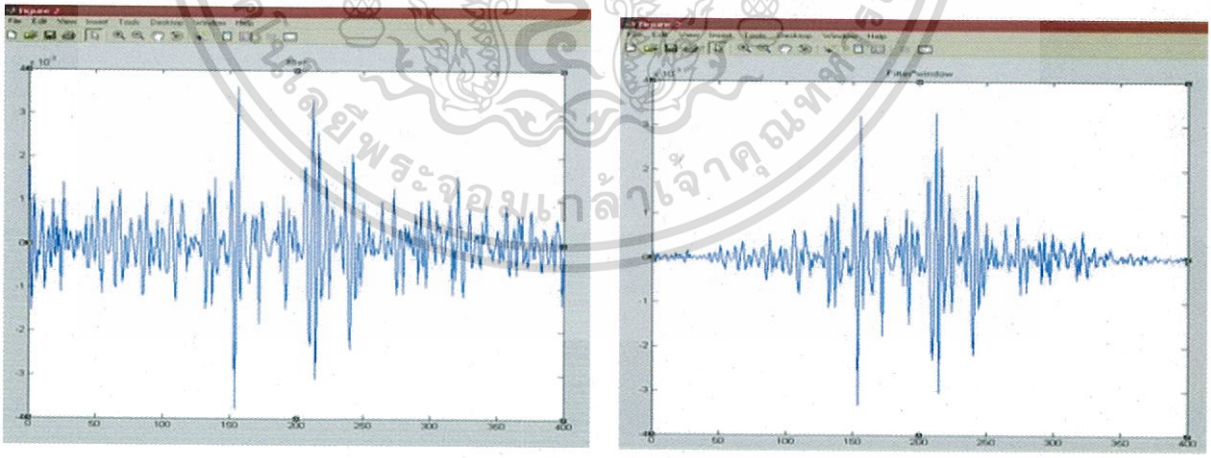
เป็นขั้นตอนทำให้อัตราส่วนของสัญญาณต่อสัญญาณรบกวนมีค่าคงที่ ตลอดทุกช่วงความถี่ จากสมการที่ 2.1 เราแทนค่า a=0.95

3.3.2 การแบ่งช่วงสัญญาณ (Frame blocking)

สัญญาณที่ผ่านการพรีเอมฟาซีสแล้ว s'(n) จะถูกตัดออกเป็นเฟรมๆ เฟรมละ N ข้อมูล (เราใช้ N เป็น 256) และแต่ละเฟรมจะมีส่วนที่เหลื่อมกัน 128 ข้อมูล เพื่อให้ข้อมูลในการวิเคราะห์มีความต่อเนื่อง

3.3.3 การวินโดว์ (Windowing)

เป็นการนำสัญญาณเสียงพูดแต่ละเฟรมมาผ่านฟังก์ชั่น Hamming window ซึ่งจะค่อยๆลดทอนแอมพลิจูดที่ปลายทั้งสองด้านของเฟรม ดังที่แสดงในรูปที่ 3.5



รูปที่ 3.5 ก่อนการทำ Hamming window (ภาพด้านซ้าย) และ

หลังจากทำ Hamming window (ภาพด้านขวา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การวิเคราะห์เสียงพูด (Speech Analysis)

วิเคราะห์โดยใช้วิธีการ Mel-Frequency Cepstral Coefficient (MFCC) ซึ่งมีขั้นตอนย่อยดังนี้

3.4.1 หลังจากที่ทำ Hamming window เรียบร้อยแล้ว ต้องทำการแปลงจากโดเมนเวลาเป็นโดเมนความถี่ โดยใช้ Fast Fourier Transform (FFT) ในการคำนวณ Discrete Fourier Transform (DFT) ซึ่งใน อัลกอริทึมที่ดีที่สุดใน FFT นั้นก็คือ Fast Hartley Transform (FHT) ซึ่งเป็นวิธีที่โครงงานพิเศษนี้เลือกใช้โดยมีสมการดัง 3.1

$$\mathrm{x}(k) = \sum_{n=1}^{N-1} x(n) \cdot [\cos\left(\frac{2\pi nk}{N}\right) + \sin\left(\frac{2\pi nk}{N}\right)] \quad (3.1)$$

3.4.2 จากนั้นทำการปรับให้อยู่ในสเกลความถี่ของเมลโดยการใช้วิธีการ Mel Filter Bank ดังสมการที่ 3.2

$$M = 2595 log_{10}\left(1 + \frac{f}{700}\right) \quad (3.2)$$

3.4.3 ทำการแปลงสเกลเมลให้กลับไปเป็นโดเมนเวลาอีกครั้ง โดยใช้ Discrete Cosine Transform (DCT) ซึ่งมีสมการดังสมการที่ 3.3

$$X_k = \alpha \sum_{i=0}^{N-1} x_i \cos\left\{\frac{(2i+1)\pi k}{2N}\right\} \quad (3.3)$$

เมื่อทำวิธีการ MFCC เสร็จเรียบร้อยแล้ว ค่าที่ได้จะถูกจัดเก็บเข้าไปยังฐานข้อมูล (เฉพาะเสียงต้นแบบ)

3.5 การจำแนกรูปแบบ (Pattern Classification)

ทำการเปรียบเทียบสัญญาณอินพุต 2 สัญญาณ โดยใช้วิธี DTW และ FastDTW และหาวิธีที่เหมาะสมที่สุด เพื่อหาระยะห่าง (หรือค่าความคล้าย) ระหว่างสัญญาณอ้างอิงกับสัญญาณทดสอบที่นำมาเปรียบเทียบ และนำคำตอบที่เทียบได้ไปให้ระบบตัดสินใจแล้วแสดงผล เช่น เมื่อพูดคำสั่งว่า “เปิดบราวเซอร์” ระบบก็จะนำเสียงที่ได้จากคำสั่งเสียงไปประมวลผลแล้วทำการเปรียบเทียบโดยใช้ DTW หรือ FastDTW ซึ่งถ้าเมื่อตรงกับคลังเสียงที่มีแล้วก็จะดำเนินการเปิดบราวเซอร์ตามที่ผู้ใช้สั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

ในบทนี้จะกล่าวถึงการทดลองและแสดงผลการทดลองโดยใช้วิธี MFCC ในการวิเคราะห์ลักษณะเด่นของเสียง แล้วเลือกใช้ DTW และ FastDTW ในการหาค่าความต่างของเสียง เพื่อกำหนด ค่าความต่างที่ยอมรับได้ในโปรแกรมต่อไป แล้วทดสอบประสิทธิภาพความแม่นยำของโปรแกรม

4.1 ข้อกำหนดในการทดลอง

4.1.1 ลักษณะของข้อมูลเสียงพูด

สัญญาณเสียงที่ใช้ในการทดลองนี้ จะใช้ผู้พูดคนเดียว หมายความว่า คำสั่งเสียง และต้นแบบเสียง พูดจากคนเดียวกัน โดยใช้ภาษาไทย ในการพูด มีคำสั่งพื้นฐาน 4 คำสั่ง (เนื่องจาก Raspberry Pi มีระบบประมวลผลที่ไม่สูงมากนัก ทำให้เวลาในการประมวลผลสูง) คือ “เปิดบราวเซอร์”, “เปิดเครื่องคิดเลข”, “เปิดเทอมินอล” และ “เปิดตัวจัดการไฟล์” ไฟล์เสียงที่ได้จะอยู่ในรูป ไฟล์ “.wav”

4.1.2 การประมวลผลเบื้องต้น

การประมวลผลเบื้องต้นจัดเป็นขั้นตอนในการลดจำนวนข้อมูลโดยการแสดงลักษณะของสัญญาณเสียงพูดด้วยพารามิเตอร์เพียงไม่กี่ค่าได้อย่างมีประสิทธิภาพ ในการทดลองนี้ใช้ การวิเคราะห์เสียงพูดโดยใช้วิธี สัมประสิทธิ์เมลฟรีเคว้นซีเซปสตรอล (Mel Frequency Cepstral Coefficient : MFCC)

4.2 วิธีการทดลอง

4.2.1 การทดลองหาค่าความต่าง (distance) ของวิธีการ Dynamic Time Warping และ Fast Dynamic Time Warping

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ทำการทดลองโดย ให้ผู้พูดบันทึกเสียง(ไม่มีเสียงรบกวน) แล้วนำเข้าการประมวลผลเบื้องต้น เพื่อลดจำนวนข้อมูลแล้วนำเปรียบเทียบคำสั่งเสียงต่างๆโดยวิธี DTW และ FastDTW เพื่อหาค่าความต่าง (distance) โดยให้ผู้ทดลอง พูดคำสั่งทั้ง 4 คำสั่งเบื้องต้น คำสั่งละ 5 ครั้ง

หลังจากทดลองแล้วเปรียบเทียบประสิทธิภาพและเวลาแล้วเลือกวิธีการที่ยอมรับได้ กำหนดค่าความต่างที่เหมาะสม เพื่อใช้ในโปรแกรม

4.2.2 ทดลองหาค่าความแม่นยำในการสั่งงานจากค่าความต่าง(distance) ที่กำหนด

ให้ผู้ทดลองสั่งคำสั่งต่างๆทั้ง 4 คำสั่ง คำสั่งละ 10 ครั้ง เพื่อวัดความแม่นยำโดยเลือกใช้วิธี DTW และกำหนดค่าความต่าง(distance) = 49

โดยให้อัดเสียงต้นแบบทั้ง 4 คำสั่งในโปรแกรมดังรูป แล้วทดสอบว่าสั่งงานได้ตามที่สั่งหรือไม่

4.3 ผลการทดลอง

4.3.1 การทดลองหาค่าความต่าง (distance) ของวิธีการ Dynamic Time Warping และ Fast Dynamic Time Warping

4.3.1.1 การทดลองหาค่าความต่าง (distance) ของวิธีการ Dynamic Time Warping

ในการทดลอง ใช้ผู้พูดคนเดียว มีคำสั่งพื้นฐาน 4 คำสั่ง คือ “เปิดบราวเซอร์”, “เปิดเครื่องคิดเลข”, “เปิดเทอมินอล” และ “เปิดตัวจัดการไฟล์” คำสั่งละ 5 ครั้ง ได้ผลดังตารางที่ 4.1

149256

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

voice	browser1	browser2	browser3	browser4	browser5	calculator1	calculator2	calculator3	calculator4	calculator5	terminal1	termina2	terminal3	terminal4	terminal5	file1	file2	file3	file4	file5
browser1	0	41.9	43.5	45.8	44.25	58.41	58.24	57.5	57.56	55.01	52.25	53.73	53.25	52.66	52.63	58.13	58.47	58.74	61.9	58.35
browser2		0	41.57	45.57	46.71	54.93	56.97	55.69	55.31	54.5	57.48	55.35	57.59	55.17	53.73	58.34	59.58	58.87	61.07	58.95
browser3			0	44.41	44.13	56.13	56.16	56.01	55.95	53.93	54.78	54.11	54.57	52.72	52.28	57.92	57.22	57.07	60.97	58.16
browser4				0	48.17	54.1	53.87	54.16	53.44	52.23	59.57	57.38	56.87	56.7	56.94	54.79	54.92	54.5	59.29	57.21
browser5					0	58.21	56.61	57.92	57.07	54.04	54.77	53.6	57.85	54.47	53.9	56.99	56.67	56.47	60.43	57.3
calculator1						0	45.56	47.26	45.25	46.81	52.25	57.48	54.78	59.57	54.77	62.04	63.5	60.11	64.35	64.2
calculator2							0	47.57	44.36	43.54	53.73	55.35	54.11	57.38	53.6	58.49	59.89	57.94	62.51	60.42
calculator3								0	45.93	46.15	53.25	57.59	54.57	56.87	57.85	61.06	62.22	59.61	64.17	61.93
calculator4									0	43.73	52.66	55.17	52.72	56.7	54.47	60.39	62.28	59.77	63.06	62.04
calculator5										0	52.63	53.73	52.28	56.94	53.9	59.73	60.41	57.03	61.65	60.67
terminal1											0	44.57	42.28	45.32	41.45	59	58.99	57.6	60.48	60.88
terminal2												0	43.92	47.87	43.59	57.64	57	57.68	60.15	57.5
terminal3													0	46.69	43.56	58.54	58.41	56.09	61.92	58.83
terminal4														0	46.36	60.38	59.84	61.33	63.84	61.26
terminal5															0	58.07	58.57	58.47	60.11	59.96
file1																0	42.35	44.06	43.5	45.74
file2																	0	43.88	43.99	42.59
file3																		0	45.47	46.69
file4																			0	48.26
file5																				0

ตารางที่ 4.1 แสดงค่าความต่างของเสียงของวิธี DTW

ในแต่ละการเปรียบเทียบจะใช้เวลา ประมาณ 18 ถึง 28 วินาที ซึ่งโดยเฉลี่ยจะอยู่ที่ 22 วินาที ในสีที่เหมือนกันคือ คำสั่งเดียวกัน และที่ไม่มีสีคือค่าความต่าง ของคำสั่งที่ต่างกัน

ในการทดลองหาค่าความต่าง (distance) ของวิธีการ Dynamic Time Warping ดังตารางที่ 4.1 จะเห็นว่าคำสั่งเดียวกันจะให้ค่าความต่าง(distance) ที่ประมาณ 40-49 และคำสั่งที่ต่างกันจะให้ค่าความต่าง(distance) ที่ 52-70

	บราวเซอร์	เครื่องคิดเลข	เทอมินอล	ตัวจัดการไฟล์
บราวเซอร์	44.60	55.76	54.97	58.09
เครื่องคิดเลข		45.62	54.97	61.18
เทอมินอล			44.56	59.30
ตัวจัดการไฟล์				44.65

ตารางที่ 4.2 แสดงค่าเฉลี่ยของค่าความต่างของเสียงของวิธี DTW

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	เสียงเหมือนกัน	เสียงต่างกัน
ค่าเฉลี่ย (Average)	44.86	57.38
ส่วนเบี่ยงเบนมาตรฐาน (Standard Deviation)	1.8333	3.0000

ตารางที่ 4.3 แสดงค่าเฉลี่ยและส่วนเบี่ยงเบนมาตรฐานของค่าความต่างของเสียงของวิธี DTW

จากตารางที่ 4.3 จะสรุปได้ว่าเสียงที่เหมือนกันมีค่าเฉลี่ย 44.86 มีส่วนเบี่ยงเบนมาตรฐาน 1.833 และเสียงที่ต่างกันจะมีค่าเฉลี่ย 57.38 มีส่วนเบี่ยงเบนมาตรฐาน 3.0 ซึ่งจะเห็นผลต่างของเสียงที่เหมือนกัน และเสียงที่ต่างกันได้ชัดเจน ทำให้สามารถกำหนดค่าความต่าง ในโปรแกรมได้ง่าย

4.3.1.2 การทดลองหาค่าความต่าง (distance) ของวิธีการ Fast Dynamic Time Warping

ในการทดลอง ใช้ผู้พูดคนเดียว มีคำสั่งพื้นฐาน 4 คำสั่ง คือ “เปิดบราวเซอร์”, “เปิดเครื่องคิดเลข”, “เปิดเทอมินอล” และ “เปิดตัวจัดการไฟล์” คำสั่งละ 5 ครั้ง โดยใช้ radius = 5 ได้ผลดังนี้

voice	browser1	browser2	browser3	browser4	browser5	calculator1	calculator2	calculator3	calculator4	calculator5	terminal1	terminal2	terminal3	terminal4	terminal5	file1	file2	file3	file4	file5
browser1	0	19845.98	14784.96	19034.84	14738.47	22111.47	19642.78	25057.88	21881.23	20954.42	20437.69	17354.12	18922.43	19954.02	18075.49	21640.64	20383.27	22165.81	22528.77	20591.15
browser2		0	15922.35	17614.41	16941.61	21149.1	18629.23	21802.7	18393.05	17801.95	17557.29	16751.08	17312.58	23586.95	16823.08	19957.36	21053.5	20872.73	20757.92	22640.65
browser3			0	16071.81	16154.28	19735.6	18294.51	21480.12	20674.4	18010.89	18176.49	16921.99	17510.02	20422.35	15361.81	19615	19216.13	21956.31	21413.66	20489.27
browser4				0	15563.08	19367.31	18668.2	22500.94	20233.53	18494.38	19285.32	17390.28	18023.27	20102.87	17464.12	20252.57	20369.37	20729.4	20540.67	21386.33
browser5					0	21701.36	18247.96	20906.2	16911.66	17288.51	16787.85	15080.52	16408.07	18364.52	14654.72	19275.48	18660.83	19418.45	20849.77	19001.33
calculator1						0	14930.93	17884.49	16138.04	15655.04	16254.41	19416.56	20228.16	25465.78	19489.07	22369.16	23321.26	20385	21056.83	23316.65
calculator2							0	19897.7	17373.51	13748.18	17069.46	15801.64	16796.66	21236.19	15985.45	20149.19	20432.77	19022.24	21862.23	21519.65
calculator3								0	17084.54	18088.99	18629.48	19801.81	20195.63	23783.69	21028.86	22962.33	23371.43	22296.82	23996.59	24052.85
calculator4									0	13773.81	15973.47	16090.14	17574.91	18612.53	17833.92	20201.25	20198.85	19923.34	19627.99	22799.92
calculator5										0	16995.15	15425.93	16388.86	18596.77	16337.05	20193.47	21816.45	19722.3	19602.41	21169.23
terminal1											0	14311.34	18198.13	22601.34	13836.01	18726.9	19754.26	18190.6	18398.24	21018.84
terminal2												0	12743.75	16158.38	12065.59	17871.42	17560.06	18078.22	19022.45	19933.14
terminal3													0	17335.65	12991.27	20280.72	19890.52	18921.91	20371.9	21820.49
terminal4														0	15369.81	21513.93	20714.21	25493.5	22956.9	23379
terminal5															0	20888.46	18615.26	17916.37	18510.11	21252.41
file1																0	15187.75	18627.76	19290.5	18071.13
file2																	0	15101.86	20169.23	16260.45
file3																		0	18847.52	18127.63
file4																			0	21327.69
file5																				0

ตารางที่ 4.4 แสดงค่าความต่างของเสียงของวิธี FastDTW

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในแต่ละการเปรียบเทียบจะใช้เวลา ประมาณ 7 ถึง 13 วินาที ซึ่งโดยเฉลี่ยจะอยู่ที่ประมาณ 10 วินาที ในสีที่เหมือนกันคือ คำสั่งเดียวกัน และที่ไม่มีสีคือค่าความต่างของคำสั่งที่ต่างกัน

ในการทดลองหาค่าความต่าง (distance) ของวิธีการ Fast Dynamic Time Warping ดังตารางที่ 4.4 จะเห็นว่าคำสั่งเดียวกันจะให้ค่าความต่าง(distance) ที่ประมาณ 12000-21000 และคำสั่งที่ต่างกัน จะให้ค่าความต่าง(distance) ที่ประมาณ 15000-25000

	บราวเซอร์	เครื่องคิดเลข	เทอมินอล	ตัวจัดการไฟล์
บราวเซอร์	16667.18	19997.58	17949.16	20630.65
เครื่องคิดเลข		16457.52	18440.46	21414.81
เทอมินอล			15561.13	20043.19
ตัวจัดการไฟล์				18101.15

ตารางที่ 4.5 แสดงค่าเฉลี่ยของค่าความต่างของเสียงของวิธี FastDTW

	เสียงเหมือนกัน	เสียงต่างกัน
ค่าเฉลี่ย (Average)	16696.75	19745.98
ส่วนเบี่ยงเบนมาตรฐาน (Standard Deviation)	2397.07	2219.32

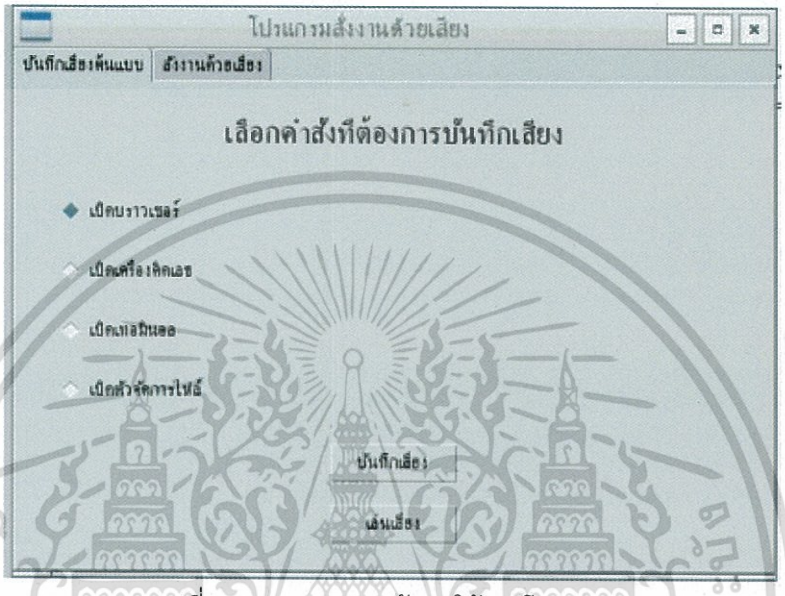
ตารางที่ 4.6 แสดงค่าเฉลี่ยและส่วนเบี่ยงเบนมาตรฐานของค่าความต่างของเสียงของวิธี FastDTW

จากตารางที่ 4.6 จะสรุปได้ว่าเสียงที่เหมือนกันมีค่าเฉลี่ย 16696.75 มีส่วนเบี่ยงเบนมาตรฐาน 2397.07 และเสียงที่ต่างกันมีค่าเฉลี่ย 19745.98 มีส่วนเบี่ยงเบนมาตรฐาน 2219.32 แต่จะเห็นได้ว่าค่าความต่าง (distance) มีการแกว่งและมีช่วงที่ซ้อนทับกันมาก(จากในตารางที่ 4.4) ซึ่งทำให้กำหนดค่าในโปรแกรมได้ยาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1.3 ทดลองหาค่าความแม่นยำในการสั่งงานจากค่าความต่าง(distance) ที่กำหนด

ให้ผู้ทดลองสั่งคำสั่งต่างๆทั้ง 4 คำสั่ง คำสั่งละ 10 ครั้ง ผ่านโปรแกรม เพื่อวัดความแม่นยำโดยเลือกใช้วิธี DTW และกำหนดค่าความต่าง(distance) = 49



รูปที่ 4.1 แสดงภาพหน้าจอใช้งานโปรแกรม

ครั้งที่ทดลอง	1	2	3	4	5	6	7	8	9	10
เปิดบราวเซอร์										
เปิดเครื่องคิดเลข										
เปิดเทอมินอล										
เปิดตัวจัดการไฟล์										
หมายเหตุ สีเขียว=เปิดถูก, สีแดง=เปิดผิด, สีดำ=ไม่เปิด(ให้อัดเสียงใหม่)										

ตารางที่ 4.7 แสดงผลการทดลองสั่งงานด้วยเสียง

ในการทดลองหาค่าความแม่นยำในการสั่งงานจากค่าความต่าง(distance) ที่กำหนด ผู้จัดทำจึงได้เลือกวิธีการ DTW ถึงแม้ความเร็วในการประมวลผลจะต่ำแต่มีความแม่นยำที่สูง ผลที่ได้คือทุกคำสั่ง ไม่มีการเปิดผิดคำสั่ง ผลทั้งหมดคือ คำสั่ง “เปิดบราวเซอร์” เปิดถูก 9 ใน 10 ครั้ง, คำสั่ง “เปิดเครื่องคิดเลข” เปิดถูก 10 ใน 10 ครั้ง, คำสั่ง “เปิดเทอมินอล” เปิดถูก 9 ใน 10 ครั้ง, คำสั่ง “เปิดตัวจัดการไฟล์” เปิดถูก 9 ใน 10 ครั้ง

จะเห็นได้ว่าตัวโปรแกรมมีความแม่นยำสูง ประมาณ 92.5 เปอร์เซ็นต์ และไม่มีการเปิดคำสั่งที่ไม่ได้สั่ง แต่ก็ยังมีผลเสียที่เวลาที่ใช้ในการประมวลผลที่สูงนั้นเนื่องมาจาก Raspberry Pi มีประสิทธิภาพในการประมวลผลที่ต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลอง

ปัญหาพิเศษนี้เป็นปัญหาพิเศษที่ศึกษาในเรื่อง ระบบสั่งงานด้วยเสียง ที่ผู้จัดทำได้ออกแบบให้มีคำสั่งพื้นฐาน 4 คำสั่ง ซึ่งมีคำสั่งดังนี้ คือ “เปิดบราวเซอร์”, “เปิดเครื่องคิดเลข”, “เปิดเทอมินอล” และ “เปิดตัวจัดการไฟล์” โดยทางผู้จัดทำได้ออกแบบการทดลองทั้งหมด 3 การทดลอง เพื่อหาวิธีที่เหมาะสมและทดสอบประสิทธิภาพของ ระบบสั่งงานด้วยเสียง ได้แก่ การทดลองหาค่าความต่าง (distance) ของวิธีการ Dynamic Time Warping, การทดลองหาค่าความต่าง (distance) ของวิธีการ Fast Dynamic Time Warping, การทดลองหาค่าความแม่นยำในการสั่งงานจากค่าความต่าง(distance) ที่กำหนด

ในการทดลองหาค่าความต่าง (distance) ของวิธีการ Dynamic Time Warping ดังตารางที่ 4.1 ในบทที่ 4 จะเห็นว่าคำสั่งเดียวกันจะให้ค่าความต่าง(distance) ที่ประมาณ 40-49 และคำสั่งที่ต่างกันจะให้ค่าความต่าง(distance) ที่ 52-70 ซึ่งเห็นผลต่างที่ชัดเจน สามารถกำหนดค่าความต่าง ในโปรแกรมได้ง่าย ที่ค่าความต่าง(distance) 48-51 แล้วแต่จะกำหนด เพื่อเป็นตัวตัดสินใจในโปรแกรม แต่วิธีการ DTW นี้จะใช้เวลาในการประมวลผลต่อคำสั่งสูง(ประมาณ 20 วินาทีต่อคำสั่งพื้นฐานในโปรแกรม) ผู้จัดทำจึงได้ทดลองในวิธี FastDTW เพื่อหาทางลดเวลาในการประมวลผล

ในการทดลองหาค่าความต่าง (distance) ของวิธีการ Fast Dynamic Time Warping ดังตารางที่ 4.4 ในบทที่ 4 จะเห็นว่าคำสั่งเดียวกันจะให้ค่าความต่าง(distance) ที่ประมาณ 12000-21000 และคำสั่งที่ต่างกัน จะให้ค่าความต่าง(distance) ที่ประมาณ 15000-25000 จะเห็นได้ว่าค่าความต่าง (distance) มีการแกว่งและมีช่วงที่ซ้อนทับกันมาก ทำให้กำหนดค่าความต่าง(distance) ในโปรแกรมได้ยาก เพราะจะทำให้โปรแกรมมีความแม่นยำที่น้อย แต่มีความเร็วในการประมวลผลที่สูง(ประมาณ 10 วินาทีต่อคำสั่งฐานในโปรแกรม) (ค่า radius ยิ่งมากความเร็วยิ่งลดลง ความแม่นยำยิ่งสูงขึ้น ในการทดลองกำหนด radius=5) ผู้จัดทำจึงได้กลับมาเลือกวิธีการ DTW เพราะวิธีการ FastDTW มีความแม่นยำที่น้อยเกินไป

จากการทดลองทั้งหมดทำให้เราได้ทราบว่า ระบบสั่งงานด้วยเสียง ที่ออกแบบ(MFCC & DTW) สามารถทำตามคำสั่งได้ดี มีความแม่นยำที่สูง แต่มีระยะเวลาในการประมวลผลที่สูงเช่นกัน โดยระยะเวลาจะแปรผันตรงกับจำนวนคำสั่งพื้นฐานที่ใช้ในการสั่งงาน

ข้อที่น่าสังเกตคือ ในการประมวลผล MFCC จะใช้เวลาประมวลผลที่น้อยมากแม้จะใช้ Raspberry Pi ที่มีประสิทธิภาพในการประมวลผลต่ำ ทำให้ระบบการรู้จำ หรือการใช้ API โดยการต่อระบบเน็ตเวิร์คในการจับคู่ที่ใกล้เคียงกันมีการใช้อย่างแพร่หลาย เพราะกำจัดข้อจำกัดทางการประมวลผลของ Raspberry Pi

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

เกรียงไกร เหลืองอำพล. **การพัฒนาเทคนิคการรู้จำเสียงพูดด้วย DTW กับ LPC และ LSP.** วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี, 2553.

เกรียงศักดิ์ เตมีย์. **การเปรียบเทียบวิธีการหาลักษณะเด่นของเสียงสำหรับการรู้จำคำไทย.** วิทยานิพนธ์ปริญญามหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2544.

เจษฎา กานต์ประชา. **การรู้จำเสียงพูดภาษาไทยอย่างคงทนโดยใช้สัมประสิทธิ์เมลฟรีเควนซีเซปสตรอลของค่าอัตสหสัมพันธ์ของเสียงพูดที่มีสัญญาณรบกวน.** วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมไฟฟ้า จุฬาลงกรณ์มหาวิทยาลัย, 2545.

ชัย วุฒิวิวัฒน์ชัย, สุทัศน์ แซ่ตั้ง และวารินทร์ อัจฉริยะกุลพร. ความก้าวหน้าของการพัฒนาระบบระบุผู้พูดภาษาไทย. **NECTEC Technical Journal Vol 2,** 7(2542) :24-35.

ฐนียา สัตยพานิช, อัศนีย์ ก่อตระกูล. **ระบบรู้จำเสียงพูดภาษาไทยต่อเนื่องแบบเฉพาะบุคคลสำหรับการเข้าถึงอีเมล์.** คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์, 2546.

ธีราพร เรือนจำรูญ. แนวทางการพัฒนา Speech Recognition กับภาษาไทย. **วารสารร่มพฤษ์,** 1(2552-2553) : 24-52.

สมชาย จิตะพันธ์กุล. **การรู้จำเสียงพูดภาษาไทยระยะที่หนึ่ง : การรู้จำเสียงพูดคำไทยโดดๆโดยไม่ขึ้นกับผู้พูด.** สถาบันวิจัยและพัฒนาของคณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2540.

สุวรรณ รื่นสุคนธ์. **ประสิทธิภาพของ LPC และ LSP ในการรู้จำแนวทางเดินเสียงพูดภาษาไทยร่วมกับโครงข่ายประสาทเทียม.** วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี, 2552.

Anjali Bala, Abhijeet Kumar, Nidhika Birla. VOICE COMMAND RECOGNITION SYSTEM BASED ON MFCC AND DTW. **International Journal of Engineering Science and Technology.** 2(2555):7335-7342.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

James Lyons. 2013. **library provides common speech features for ASR including MFCCs and filterbank energies**. [Online]. Available: https://github.com /jameslyons/python_speech_features. เข้าถึงเมื่อวันที่ 9 ธ.ค. 58

Jeet Kumar, Om Prakash Prabhakar, Navneet Kumar Sahu. Comparative Analysis of Different Feature Extraction and Classifier Techniques for Speaker Identification System: A Review. **International Journal of Innovative Research in Computer and Communication Engineering**. 2(2557):2760-2769.

Kazuaki Tanida. 2015. **A Python implementation of FastDTW.** [Online]. Available: https://github.com/slaypni/fastdtw. เข้าถึงเมื่อวันที่ 6 ธ.ค. 58.

Manish Soni, Padma Kunthe. A General Comparison Of FFT Algorithms. **9th National Conference**. 2554.

Namrata Dave. Feature Extraction Methods LPC, PLP and MFCC In Speech Recognition. **INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY**. 1(2556):1-5.

Pierre Rouanet. 2014. **DTW (Dynamic Time Warping) python module**. [Online]. Available: https://pypi.python.org/pypi/dtw/1.0. เข้าถึงเมื่อวันที่ 8 ธ.ค. 58.

Pratik K. Kurzekar, Ratnadeep R. Deshmukh, Vishal B. Waghmare, Pukhraj P. Shrishrimal. A Comparative Study of Feature Extraction Techniques for Speech Recognition System. **International Journal of Innovative Research in Science, Engineering and Technology**. 3(2557):18006-18016.

Stan Salvador, Philip Chan. **FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space**. Dept. of Computer Sciences Florida Institute of Technology,2004

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Source Code ของโปรแกรม

```
# -*- coding: utf-8 -*-
from Tkinter import *
from ttk import *
from features import mfcc
from features import logfbank
from numpy.linalg import norm
from dtw import dtw
import pyaudio, wave, tkMessageBox, time, tkFont
import scipy.io.wavfile as wav
import os, sys

#input microphone
class Recorder(object):
    '''A recorder class for recording audio to a WAV file.'''

    def __init__(self, channels=1, rate=44100, frames_per_buffer=8192):
        self.channels = channels
        self.rate = rate
        self.frames_per_buffer = frames_per_buffer
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    def open(self, fname, mode='wb'):
        return RecordingFile(fname, mode, self.channels, self.rate,
                             self.frames_per_buffer)

class RecordingFile(object):
    def __init__(self, fname, mode, channels,
                 rate, frames_per_buffer):
        self.fname = fname
        self.mode = mode
        self.channels = channels
        self.rate = rate
        self.frames_per_buffer = frames_per_buffer
        self._pa = pyaudio.PyAudio()
        self.wavefile = self._prepare_file(self.fname, self.mode)
        self._stream = None

    def __enter__(self):
        return self

    def __exit__(self, exception, value, traceback):
        self.close()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
def record(self, duration):
    # Use a stream with no callback function in blocking mode
    self._stream = self._pa.open(format=pyaudio.paInt16,
                                 channels=self.channels,
                                 rate=self.rate,
                                 input=True,
                                 frames_per_buffer=self.frames_per_buffer)
    for _ in range(int(self.rate / self.frames_per_buffer * duration)):
        audio = self._stream.read(self.frames_per_buffer)
        self.wavefile.writeframes(audio)
    return None

def start_recording(self):
    # Use a stream with a callback in non-blocking mode
    self._stream = self._pa.open(format=pyaudio.paInt16,
                                 channels=self.channels,
                                 rate=self.rate,
                                 input=True,
                                 frames_per_buffer=self.frames_per_buffer,
                                 stream_callback=self.get_callback())
    self._stream.start_stream()
    return self
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
def stop_recording(self):
    self._stream.stop_stream()
    return self

def get_callback(self):
    def callback(in_data, frame_count, time_info, status):
        self.wavefile.writeframes(in_data)
        return in_data, pyaudio.paContinue
    return callback


def close(self):
    self._stream.close()
    self._pa.terminate()
    self.wavefile.close()

def _prepare_file(self, fname, mode='wb'):
    wavefile = wave.open(fname, mode)
    wavefile.setnchannels(self.channels)
    wavefile.setsampwidth(self._pa.get_sample_size(pyaudio.paInt16))
    wavefile.setframerate(self.rate)
    return wavefile
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
# start record(Tab1) by triger the button
def startRec():
    fileName = var.get()
    rec = Recorder(channels=2)
    with rec.open(fileName, 'wb') as recfile2:
        recfile2.start_recording()
        result = tkMessageBox.showinfo("กำลังบันทึกเสียง", "ระบบกำลังบันทึกเสียง หากต้องการ
หยุดบันทึก กดปุ่ม \"OK\"")
        if result:
            recfile2.stop_recording()
            finish = tkMessageBox.showinfo("บันทึกเสียงเสร็จสิ้น", "ระบบได้ทำการบันทึกคำสั่งเสียง
เสร็จสิ้นแล้ว")

#start record(Tab2) to compare wav with voice in the database
def startCommand():
    rec = Recorder(channels=2)
    with rec.open("voiceCommand.wav", 'wb') as recfile2:
        recfile2.start_recording()
        result = tkMessageBox.showinfo("กำลังบันทึกเสียง", "ระบบกำลังบันทึกเสียง หากต้องการ
หยุดบันทึก กดปุ่ม \"OK\"")
        if result:
            recfile2.stop_recording()
            voiceCommand()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#play sound that recorded
def playVoice():
    fileName = var.get()
    CHUNK = 1024
    wf = wave.open(fileName, 'rb')
    p = pyaudio.PyAudio()

    stream = p.open(format=p.get_format_from_width(wf.getsampwidth()),
                    channels=wf.getnchannels(),
                    rate=wf.getframerate(),
                    output=True)

    data = wf.readframes(CHUNK)

    while data != '':
        stream.write(data)
        data = wf.readframes(CHUNK)

    stream.stop_stream()
    stream.close()
    p.terminate()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#Extract feature from wave file and find the distance between voiceCommand and the
others
def mfcc_dtw(fileName):
    (rate1,sig1) = wav.read("/home/pi/voiceCommand.wav")
    (rate2,sig2) = wav.read("/home/pi/"+fileName)
    mfcc1 = mfcc(sig1,rate1)
    mfcc2 = mfcc(sig2,rate2)
    dist, cost, path = dtw(mfcc1, mfcc2, dist=lambda x, y: norm(x - y, ord=1))
    return dist

#set the dist for tell the program to do what
def voiceCommand():
    start = time.time()
    while True:
        dist1 = mfcc_dtw("OpenBrowser.wav")
        print dist1
        if dist1<=49:
            elapsed = time.time()-start
            #print elapsed
            os.system('epiphany-browser')
            break

        dist2 = mfcc_dtw("OpenCalculator.wav")
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
print dist2
if dist2<=49:
    elapsed = time.time()-start
    #print elapsed
    os.system('galculator')
    break

dist3 = mfcc_dtw("OpenTerminal.wav")
print dist3
if dist3<=49:
    elapsed = time.time()-start
   #print elapsed
    os.system('lxterminal')
    break

dist4 = mfcc_dtw("OpenFileManager.wav")
print dist4
if dist4<=49:
    elapsed = time.time()-start
    #print elapsed
    os.system('pcmanfm')
    break
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        tkMessageBox.showinfo("ไม่พบคำสั่ง", "กรุณาบันทึกเสียงใหม่อีกครั้ง", icon='warning')
        elapsed = time.time()-start
        #print elapsed
        break


#create Window GUI
root = Tk()
customFont = tkFont.Font(size=15)
root.title("โปรแกรมสั่งงานด้วยเสียง")

#create Notebook(Tab) widget
n = Notebook(root, width=500, height=320)
f1 = Frame(n)   #create frame in Notebook
f2 = Frame(n)
n.add(f1, text="บันทึกเสียงต้นแบบ") #add Tab1
n.add(f2, text="สั่งงานด้วยเสียง")   #add Tab2
n.pack()

Label(f1, text="เลือกคำสั่งที่ต้องการบันทึกเสียง", font=customFont).pack(pady=20)

global var
var = StringVar()   #set Variable to contain the value of RadioButton
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
var.set("OpenBrowser.wav") #set default value
R1 = Radiobutton(f1, text="เปิดบราวเซอร์", variable=var, value="OpenBrowser.wav")
R1.pack(padx=30, pady=10, fill=BOTH)

R2 = Radiobutton(f1, text="เปิดเครื่องคิดเลข", variable=var, value="OpenCalculator.wav")
R2.pack(padx=30, pady=10, fill=BOTH)

R3 = Radiobutton(f1, text="เปิดเทอมินอล", variable=var, value="OpenTerminal.wav")
R3.pack(padx=30, pady=10, fill=BOTH)

R4 = Radiobutton(f1, text="เปิดตัวจัดการไฟล์", variable=var, value="OpenFileManager.wav")
R4.pack(padx=30, pady=10, fill=BOTH)

#record button Tab1
record_button1 = Button(f1, text = "บันทึกเสียง", command=startRec).pack(padx=15,
pady=15)
record_button3 = Button(f1, text = "เล่นเสียง", command=playVoice).pack(padx=50)

#Tab2
Label(f2, text="กดปุ่ม \"บันทึกเสียง\" เพื่อเริ่มการสั่งงานด้วยเสียง",
font=customFont).pack(padx= 10, pady=25)
record_button2 = Button(f2, text = "บันทึกเสียง",
command=startCommand).pack(padx=15, pady=25)
root.mainloop()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้