

ระบบควบคุมฟังก์ชัน Pitch Bend และ Vibrato ด้วยเซ็นเซอร์สัมผัสสำหรับ
เครื่องดนตรีคีย์บอร์ดชนิดลิ้มเปียโน

PITCH BEND AND VIBRATO CONTROL SYSTEM FOR PIANO-STYLE KEYBOARD BY
USING A TOUCH SENSOR



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2561

KMITL-2018-EN-M-040-044

ระบบควบคุมฟังก์ชัน Pitch Bend และ Vibrato ด้วยเซ็นเซอร์สัมผัสสำหรับ
เครื่องดนตรีคีย์บอร์ดชนิดลิ้มเปียโน

PITCH BEND AND VIBRATO CONTROL SYSTEM FOR PIANO-STYLE KEYBOARD BY
USING A TOUCH SENSOR



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2561

KMITL-2018-EN-M-040-044

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PITCH BEND AND VIBRATO CONTROL SYSTEM FOR PIANO-STYLE KEYBOARD BY
USING A TOUCH SENSOR



A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN ELECTRONICS ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2018
KMITL-2018-EN-M-040-044

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2018

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ ระบบควบคุมฟังก์ชัน Pitch Bend และ Vibrato ด้วยเซ็นเซอร์สัมผัสสำหรับเครื่องดนตรีคีย์บอร์ดชนิดลิ้มเปียโน

Thesis Title Pitch Bend and Vibrato Control System for Piano-Style Keyboard by using a Touch Sensor

นักศึกษา นายนันทพงศ์ วัฒนศิริ

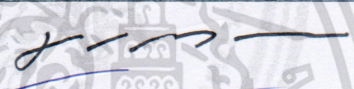
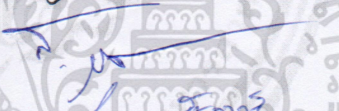
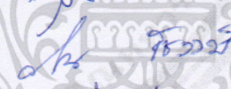
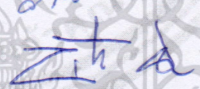

รหัสประจำตัว 58601227

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชา วิศวกรรมอิเล็กทรอนิกส์

อาจารย์ที่ปรึกษาวิทยานิพนธ์ รศ.ดร.สุรพันธุ์ เอื้อไพบูลย์

หมายเลขวิทยานิพนธ์ KMITL-2018-EN-M-040-125

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
รศ.ดร.มนัส	สังวรศิลป์	
รศ.ดร.สมศักดิ์	ชุมช่วย	
ผศ.ดร.วันวิสา	ชัชวงษ์	
ผศ.ดร.พิทักษ์	ธรรมวาริน	
รศ.ดร.สุรพันธุ์	เอื้อไพบูลย์	

วัน / เดือน / ปี ที่สอบ วันอังคารที่ 17 กรกฎาคม พ.ศ. 2561 เวลา 13.30-15.30 น.
สถานที่สอบ ณ ห้อง HM-302 อาคารเฉลิมพระเกียรติ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

คณะวิศวกรรมศาสตร์ รับรองแล้ว



(รองศาสตราจารย์ ดร. คมสัน มาลีสี)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น **ฉบับตีพิมพ์ คณะวิศวกรรมศาสตร์** ขณดำเนินการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้าง **วันที่ 17 กรกฎาคม พ.ศ. 2561** ไว้เสมอ

หัวข้อวิทยานิพนธ์	ระบบควบคุมฟังก์ชัน Pitch Bend และ Vibrato ด้วยเซ็นเซอร์สัมผัสสำหรับเครื่องดนตรีคีย์บอร์ดชนิดลิ้มเปียโน
นักศึกษา	นาย นัทพงศ์ วัฒนศิริ
รหัสประจำตัว	58601227
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมอิเล็กทรอนิกส์
ปีการศึกษา	2561
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รศ.ดร.สุรพันธ์ เอื้อโงกกุล

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้นำเสนอระบบควบคุมฟังก์ชัน Pitch bend และ Vibrato สำหรับเครื่องดนตรีอิเล็กทรอนิกส์คีย์บอร์ดชนิดลิ้มเปียโน (Piano-style Electronic Keyboard) โดยใช้งานผ่านการสัมผัสของนิ้วบนแผ่นเซ็นเซอร์ตรวจจับการสัมผัสชนิดวัดความจุไฟฟ้า (Capacitive Touch Sensor) ซึ่งจะติดตั้งบนลิ้มของคีย์บอร์ด นักดนตรีสามารถเล่นฟังก์ชันบนระบบที่นำเสนอในลักษณะเดียวกับการเล่นบนเครื่องดนตรีสาย (String Instrument) เซ็นเซอร์ตรวจจับการสัมผัสถูกใช้สำหรับวัดตำแหน่งการสัมผัสของนิ้วเพื่อนำไปใช้ระบุลักษณะการรูดนิ้วบนลิ้มคีย์ของผู้เล่น ลักษณะการรูดนิ้วจะถูกนำมาใช้ตัดสินใจการทำงานของฟังก์ชัน เพื่อให้สอดคล้องกับเงื่อนไขการรูดนิ้วของระบบโดยใช้อัลกอริทึมที่จะนำเสนอ จากนั้นจะส่งชุดคำสั่ง MIDI ไปยังเอาต์พุต MIDI ของระบบเพื่อส่งการฟังก์ชันที่ถูกเรียกใช้โดยมีการนำเสนอเทคนิค MIDI Channel Modulation เพื่อให้ฟังก์ชัน Pitch bend และ Vibrato มีผลต่อโน้ตเดี่ยวๆได้ (เนื่องจากโดยปกติแล้วฟังก์ชันจะกระทบต่อทุกๆโน้ตที่อยู่ใน MIDI channel เดียวกัน) ระบบที่นำเสนอประกอบด้วยเซ็นเซอร์สำหรับคีย์ขาว (White key) จำนวน 14 คีย์ ความยาว 2 Octave สมรรถนะของระบบที่นำเสนอถูกนำไปเปรียบเทียบกับระบบดั้งเดิมที่ใช้ล้อคันโยก (Pitch Wheel) ควบคุมฟังก์ชัน โดยเปรียบเทียบจากกราฟการเปลี่ยนแปลงระดับโน้ตระหว่างการเล่นโน้ตทดสอบฟังก์ชันด้วยระบบทั้งสอง พบว่าระบบที่นำเสนอใช้เวลาสำหรับเปลี่ยนระดับโน้ต 1 semitone โดยเฉลี่ยเท่ากับ 0.276 วินาที ซึ่งช้ากว่าระบบดั้งเดิมที่มีค่าเท่ากับ 0.138 วินาทีแต่อย่างไรก็ตาม ระบบที่นำเสนอสามารถใช้งานฟังก์ชันได้ดีสำหรับการเล่นดนตรีทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Pitch Bend and Vibrato Control System for Piano-Style Keyboard by Using a Touch Sensor
Student	Mr. Nattapong Wattanasiri
Student ID.	58601227
Graduate Level	Master Degree of Electronics Engineering
Department	Electronics Engineering
Academic Year	2018
Advisor	Assoc.Prof.Dr.Surapan Airphailboon

Abstract

This thesis presents the pitch bend and vibrato function control system for piano-style keyboard by using the finger touch on keypad. Each keypad is installed with capacitive touch sensors which are used to measure the position of the finger touch. This proposed control technique is used in the same way as in the string instruments. Touch sensors are used to measure a finger position which will be used to determine a user's finger movement. The finger movement will be used to decide when functions should be activated by using a propose algorithm, and send the MIDI control message to the MIDI output. The proposed MIDI Channel Modulation technique is used to make functions (Pitch bend and Vibrato) affect only one individual note of its MIDI channel. (Normally, they intend to affect all notes in the same MID channel) The proposed system consists of 14 touch sensors only for a white-key (2 octave length). The performance of proposed system is compared to the benchmark system which it uses pitch wheel as the controller, by matching the pitch contour of both systems from the result of the playing with test notes. Result shows that the proposed system has the transition time for pitch change within 1 semitone about 0.276 second, compared to the benchmark system has 0.138 second. Although it is slower, the proposed system can be used for the typical music applications.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้และงานวิจัยทั้งหมดสำเร็จลุล่วงได้ด้วยความกรุณาจากอาจารย์ที่ปรึกษา รศ. ดร.สุรพันธ์ เอื้อไพบูลย์ และอาจารย์ ผศ.ดร. กสิน วิเชียรชม ที่ให้โอกาสและคำแนะนำในการแก้ปัญหาข้อผิดพลาดและอุปสรรคต่างๆที่เกิดขึ้นระหว่างการดำเนินการจัดทำวิทยานิพนธ์ ตลอดจนเอื้อเพื่อสถานที่ในการทำวิทยานิพนธ์ จนวิทยานิพนธ์สามารถสำเร็จลุล่วงได้ตามเป้าหมายที่วางไว้ คณะผู้จัดทำขอขอบพระคุณเป็นอย่างสูง

ขอขอบคุณคณะวิศวกรรมศาสตร์ สาขาวิชาอิเล็กทรอนิกส์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ให้ทุนสนับสนุนและเอื้อเพื่อสถานที่ในการทำวิทยานิพนธ์ ทำให้วิทยานิพนธ์สำเร็จลุล่วงไปได้ด้วยดี

สุดท้ายนี้คณะผู้จัดทำขอขอบพระคุณบิดามารดาของคณะผู้จัดทำ ท่านทั้งสองเป็นผู้มีอุปการคุณและเป็นผู้ให้กำลังใจ วิทยานิพนธ์ฉบับนี้จึงสามารถสำเร็จลุล่วงด้วยดี

นาย นัทพงศ์ วัฒนศิริ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต่อIIIถ้าถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญรูป.....	VII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์.....	3
1.3 ขอบเขตของวิทยานิพนธ์.....	3
1.4 ขั้นตอนการดำเนินงาน.....	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
บทที่ 2 ทฤษฎีและหลักการ.....	4
2.1 ทฤษฎีด้านดนตรีสำหรับงานวิจัย.....	4
2.1.1 ระดับเสียง (Pitch).....	4
2.1.2 Pitch Bend	5
2.1.3 Vibrato.....	5
2.1.4 ความสำคัญทางด้านดนตรีของเทคนิค Pitch bend และ Vibrato.....	6
2.1.5 การรับรู้ความแตกต่างของระดับโน้ตใน Musical Scale ของมนุษย์.....	6
2.2 งานวิจัยที่เกี่ยวข้อง.....	7
2.3 หลักการทำงานของเซ็นเซอร์สัมผัสชนิดวัดความจุไฟฟ้า.....	8
2.4 การสื่อสารบนโปรโตคอล MIDI	11
2.4.1 โปรโตคอล MIDI	11
2.4.2 การเชื่อมต่อฮาร์ดแวร์ของอุปกรณ์ MIDI	12
2.5 ทฤษฎีการประมวลผลสัญญาณดิจิทัลพื้นฐานที่ใช้ในงานวิจัย.....	14
2.5.1 ทฤษฎีการสุ่มค่าตัวอย่างของ Nyquist (Nyquist Sampling Theorem)...14	
2.5.2 ตัวกรองความถี่ดิจิทัลชนิด IIR (IIR Filter).....	16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.5.3 การเข้ารหัส Unary (Unary Coding).....	17
บทที่ 3 การออกแบบระบบที่นำเสนอ.....	19
3.1 โครงสร้างและการทำงานของระบบที่นำเสนอ.....	19
3.2 การออกแบบเซ็นเซอร์สัมผัส.....	20
3.2.1 การออกแบบเซ็นเซอร์สัมผัสและวงจรที่ใช้.....	20
3.2.2 กระบวนการสกัดตำแหน่งการสัมผัส.....	22
3.3 อัลกอริธึม DSP สำหรับบอร์ดประมวลผลหลัก.....	26
3.3.1 อัลกอริธึม DSP สำหรับควบคุมฟังก์ชัน Pitch bend และ Vibrato.....	26
3.3.2 อัลกอริธึม DSP ของ Channel Modulation และ Message Generator.....	35
บทที่ 4 ผลการทดสอบของระบบที่นำเสนอ.....	41
4.1 ทดสอบการทำงานของเซ็นเซอร์สัมผัสที่ใช้.....	41
4.2 เปรียบเทียบสมรรถนะระหว่างระบบที่นำเสนอกับระบบดั้งเดิม.....	43
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....	48
5.1 สรุปผลการวิจัย.....	48
5.2 ข้อเสนอแนะ.....	48
เอกสารอ้างอิง.....	49
ภาคผนวก.....	50
ประวัติผู้เขียน.....	53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1	ตัวอย่าง MIDI message พื้นฐานที่ใช้กันบ่อยในระบบ MIDI.....12
3.1	ตารางการจอง MIDI Channel เพื่อค้นหา Channel ใหม่ที่ว่างสำหรับโน้ตที่เข้ามาใหม่.....37



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
1.1	(ก) การเล่นเทคนิค Pitch bend บนกีตาร์ และ (ข) การเล่นเทคนิค Vibrato บนไวโอลิน....	1
1.2	(ก) ล้อคันโยกที่ใช้ในระบบดั้งเดิม, (ข) การใช้งานล้อคันโยกขณะเล่นคีย์บอร์ด.....	2
1.3	ลักษณะการติดตั้งแผ่นเซ็นเซอร์วัดการสัมผัสสำหรับระบบที่นำเสนอ.....	2
2.1	ความสัมพันธ์ระหว่างความถี่ของเสียงกับตำแหน่งโน้ตบนบรรทัดห้าเส้น (Staff).....	4
2.2	(ก) สัญลักษณ์ทางดนตรีสากลสำหรับระบุการเล่น Pitch bend, (ข) ลักษณะการเปลี่ยนแปลงของระดับเสียงโน้ตระหว่างการเล่น Pitch bend.....	5
2.3	(ก) สัญลักษณ์ทางดนตรีสากลสำหรับระบุการเล่น Vibrato, (ข) ลักษณะการเปลี่ยนแปลงของระดับเสียงโน้ตระหว่างการเล่น Vibrato	5
2.4	ความสัมพันธ์ในลักษณะลอการิทึม (Logarithm) ระหว่างระดับโน้ตกับความถี่ของโน้ต.....	7
2.5	(ก) แผ่นเซ็นเซอร์ตรวจจับการสัมผัสชนิด Resistive Film สำหรับติดตั้งบนลิ้มคีย์ของระบบ MTS, (ข) การติดตั้งระบบ MST บนเครื่องดนตรี Organ-style Keyboard.....	8
2.6	แบบจำลองสำหรับอธิบายหลักการของเซ็นเซอร์สัมผัสชนิดวัดความจุไฟฟ้า.....	8
2.7	วงจรตรวจจับการเปลี่ยนแปลงค่า C_S ด้วยเทคนิค RC Charge and Discharge Method..	9
2.8	กราฟการคายประจุของกระบวนการ RC Charge and Discharge Method ซึ่งใช้เปรียบเทียบความแตกต่างของค่าสัมพัทธ์ของช่วงเวลาการคายประจุระหว่าง ขณะที่ไม่มีการสัมผัส $t_a[n]$ กับมีการสัมผัส $t_a[n + 1]$ สังเกตได้ว่า $t_a[n + 1] > t_a[n]$	10
2.9	ตัวอย่างโมดูลเซ็นเซอร์สัมผัสแบบวัดความจุไฟฟ้าที่มีขายตามตลาดอิเล็กทรอนิกส์.....	10
2.10	โครงสร้างของชุดข้อมูล MIDI สำหรับการสื่อสารข้อมูลบนโปรโตคอล MIDI.....	11
2.11	ลักษณะของพอร์ตบนอุปกรณ์ MIDI.....	13
2.12	ตัวอย่างการเชื่อมต่อระหว่างอุปกรณ์ MIDI จำนวน 2 ชิ้น.....	13
2.13	ตัวอย่างการเชื่อมต่อระหว่างอุปกรณ์ MIDI ที่มากกว่า 2 ชิ้นในลักษณะ Daisy Chain.....	13
2.14	รายละเอียดของวงจรการสื่อสารที่ใช้ภายในพอร์ต MIDI พร้อมอธิบายการส่งสัญญาณ logic HIGH จากตัวส่ง (Transmitter) ไปยังตัวรับ (Receiver).....	14
2.15	กระบวนการสุ่มค่าตัวอย่างของสัญญาณ.....	14
2.16	ผลกระทบของ Aliasing ต่อสเปกตรัมของสัญญาณที่ได้จากกระบวนการฟื้นฟูสัญญาณ.....	15
2.17	โครงสร้างของ FIR filter (ซ้าย) และ IIR Filter (ขวา).....	16
2.18	บล็อกไดอะแกรมของ 1 st -Order IIR Filter ตัวอย่างสำหรับใช้เป็น Smoothing Filter.....	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
2.19	ระดับเทอร์โมมิเตอร์ที่มีการแบ่งระดับจำนวน 7 ระดับ.....	17
2.20	ตัวอย่างการเข้ารหัส Unary ตามลักษณะของข้อมูลอินพุต 8 ระดับจากรูปที่ 2.19.....	18
3.0	Hardware Diagram โดยรวมที่ใช้ในระบบที่นำเสนอ.....	19
3.1	System Block Diagram ของระบบที่นำเสนอ.....	20
3.2	(ก) ลักษณะของแถบสไลเดอร์ (ข) หลักการระบุตำแหน่งสัมพันธ์ของแถบสไลเดอร์.....	21
3.3	วงจรวัดความจุไฟฟ้าของเซ็นเซอร์สัมผัสที่ใช้ในงานวิจัย.....	21
3.4	การตอบสนองทางความถี่ของ IIR low-pass filter ดังสมการที่ 3.1.....	22
3.5	เปรียบเทียบเอาต์พุตที่ได้จากการกรองสัญญาณ t_d ด้วย IIR Filter ในสมการที่ 3.2 (น้ำเงิน) กับสัญญาณ t_d ดั้งเดิมที่ไม่ผ่าน IIR Filter (แดง).....	23
3.6	บล็อกไดอะแกรมสำหรับระบุการสัมผัส (ต่อ 1 ขั้วสัมผัส) ด้วยค่าเวลาการคายประจุ t_d	23
3.7	การทำ Interpolation เพื่อเพิ่มตำแหน่งสัมผัสระหว่างขั้วสัมผัสที่อยู่ติดกัน.....	24
3.8	ตำแหน่งสัมผัส Y ซึ่งมีลักษณะเป็น Slider Function	25
3.9	เซ็นเซอร์สัมผัสของระบบที่นำเสนอจากการออกแบบในหัวข้อที่ 3.2.....	25
3.10	(ก) การรูดนิ้วแนวยาวสำหรับ Pitch bend (ข) การรูดนิ้วแบบกวัดแกว่งสำหรับ Vibrato	26
3.11	Full Algorithm : Pitch bend & Vibrato Control	26
3.12	อัลกอริธึมสำหรับระบุการแตะค้างและสกัดตำแหน่งแตะค้าง Y_{hold}	27
3.13	ตัวอย่างสัญญาณ <i>hold flag</i> และ Y_{hold} ที่ได้จากการทดสอบ Hold detection	28
3.14	อัลกอริธึมสำหรับควบคุมฟังก์ชัน Pitch bend	29
3.15	ฟังก์ชัน $f(Y)$ ที่ใช้แสดงความสัมพันธ์ระหว่างระดับเบนโน้ต P กับตำแหน่งสัมผัส Y	29
3.16	กราฟตำแหน่งของนิ้วที่อธิบายการทำงานของฟังก์ชันขีดเริ่มสำหรับฟังก์ชัน Pitch bend..	30
3.17	อัลกอริธึมสำหรับควบคุมฟังก์ชัน Vibrato function	30
3.18	ตำแหน่งสัมผัส Y_1, Y_2 ที่สอดคล้องต่อเงื่อนไขสำหรับเป็นตำแหน่งรอบๆจุดยอด.....	32
3.19	กราฟตำแหน่งนิ้วตัวอย่าง สำหรับอธิบายกระบวนการวัด Peak-to-peak Duration.....	32
3.20	ตัวอย่างการรูดนิ้วแบบกวัดแกว่งกรณีตำแหน่งศูนย์กลางไม่คงที่ (บน) และคงที่ (ล่าง).....	33
3.21	ตัวอย่าง Waveform ของสัญญาณ $Y_{vibrato}$, P และ V เพื่อแสดงการทำงานของอัลกอริธึมของฟังก์ชัน Pitch bend และ Vibrato ที่ได้กล่าวมา.....	34
3.22	อัลกอริธึม DSP ของ MIDI Channel Modulation และ MIDI Message Generator (ดูใช้สำหรับต่อ 1 คีย์โน้ต).....	35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.23	Flow Chart ของอัลกอริทึมของ MIDI Channel Modulation.....36
3.24	Pseudo Code ฟังก์ชัน SelectNewChannel เพื่อค้นหา MIDI Channel ใหม่ที่ว่างอยู่...38
3.25	(ก) การทำงานของ FIFO Buffer, (ข) โครงสร้างของ Circular Buffer38
3.26	รูปแบบเอาต์พุตที่ได้จาก MIDI Message Generator ตามโปรโตคอลของระบบ MIDI.....39
3.27	ฟังก์ชัน C ที่ใช้สร้าง MIDI Message สำหรับ Note ON/OFF39
3.28	ฟังก์ชัน C ที่ใช้สร้าง MIDI Message สำหรับ Vibrato39
3.29	ฟังก์ชัน C ที่ใช้สร้าง MIDI Message สำหรับ Pitch bend40
3.30	บอร์ด STM32F4Discovery พร้อมละเอียดของอุปกรณ์ต่างๆบนบอร์ด.....40
4.1	ลักษณะข้อมูลตำแหน่งสัมพันธ์ Y ที่ได้จากเซ็นเซอร์ระหว่างการรูดนิ้วทดสอบ.....41
4.2	(สีฟ้า) กราฟแรงดันวงจร RC ของขั้วสัมผัสชั้นที่ 1, (สีเหลือง) กราฟแรงดันวงจร RC ของขั้วสัมผัสชั้นที่ 2 ซึ่งอยู่ถัดต่อไปจากชั้นที่ 1 ซึ่งมีระยะห่างของเวลาการทำงานเท่ากับ 540 us42
4.3	ผลของสัญญาณรบกวน 50 Hz ที่มีต่อการวัดแรงดันวงจร RC ของขั้วสัมผัสบนเซ็นเซอร์...42
4.4	โน้ตทดสอบการเล่นฟังก์ชัน Pitch bend (บน) และ Vibrato (ล่าง).....43
4.5	ฮาร์ดแวร์สำหรับการทดสอบระบบที่นำเสนอ.....44
4.6	ไมโครโฟน Rode NT-USB พร้อมรายละเอียด Specification44
4.7	กราฟระดับโน้ตจากการทดสอบ Pitch bend ด้วยระบบดั้งเดิม (บน), ระบบที่นำเสนอ (กลาง) และ กราฟตำแหน่งนิ้ว Y บนเซ็นเซอร์ของคีย์โน้ต C4 (ล่าง).....45
4.8	กราฟระดับโน้ตจากการทดสอบ Vibrato ด้วยระบบดั้งเดิม (บน), ระบบที่นำเสนอ (กลาง) และ กราฟตำแหน่งนิ้ว Y บนเซ็นเซอร์ของคีย์โน้ต C4 (ล่าง).....46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ในบทนี้จะกล่าวถึงความเป็นมาและความสำคัญของการจัดทำวิทยานิพนธ์ รวมถึงวัตถุประสงค์ ขอบเขตของวิทยานิพนธ์ ขั้นตอนการดำเนินงานและประโยชน์ที่คาดว่าจะได้รับ

1.1 ความเป็นมาและความสำคัญของปัญหา

เครื่องดนตรีอิเล็กทรอนิกส์ประเภทคีย์บอร์ด (Keyboard) นั้น นิยมถูกนำมาใช้สำหรับการแสดงดนตรีสด (Live music) ของแนวดนตรีประเภทต่างๆ ซึ่งบ่อยครั้งที่คีย์บอร์ดถูกนำมาใช้สำหรับเล่นสังเคราะห์เสียงเครื่องดนตรีอื่นๆ อาทิเช่น เครื่องสาย (String Instrument) เครื่องเป่าลมไม้ (Woodwind Instrument) เครื่องเป่าทองเหลือง (Brass Instrument) เป็นต้น คีย์บอร์ดที่ใช้สำหรับสร้างเสียงสังเคราะห์นี้มักจะถูกเรียกในวงการดนตรีสากลว่า ซินธิไซเซอร์ (Synthesizer)

อย่างไรก็ตาม การเล่นคีย์บอร์ดเพื่อเลียนแบบเสียงเครื่องดนตรีอื่นนั้น จะไม่สามารถผลิตเสียงที่มีคุณสมบัติได้ใกล้เคียงกับการเล่นด้วยเครื่องดนตรีจริงๆ ตัวอย่างเช่น ลักษณะเสียงที่เกิดจากการเล่นเทคนิค Pitch bend และ Vibrato ในเครื่องสายโดยการขยี้หรือรูดนิ้วเพื่อเปลี่ยนความตึงของสายในลักษณะต่อเนื่อง แสดงดังรูปที่ 1.1 เนื่องจากคีย์บอร์ดนั้นไม่สามารถปรับระดับเสียงโน้ต (Pitch) ที่ถูกเล่นอยู่แบบลักษณะต่อเนื่องเพียงแค่การใช้ลิ้มคีย์บอร์ด ดังนั้น จำเป็นที่จะต้องติดตั้งอุปกรณ์เพิ่มที่สามารถควบคุมระดับโน้ตได้อย่างต่อเนื่อง สำหรับระบบดั้งเดิม (Benchmark) ของคีย์บอร์ดนั้น มีการเพิ่มล้อคันโยก (Wheel) หรือจอยสติ๊ก (Joystick) สำหรับควบคุมการเล่น Pitch bend และ Vibrato ไว้ด้านข้างของตัวเครื่องดนตรีเป็นทางเลือกหนึ่งเพื่อแก้ไขปัญหาดังกล่าว แสดงดังรูปที่ 1.2ก



(ก)



(ข)

รูปที่ 1.1 (ก) การเล่นเทคนิค Pitch bend บนกีตาร์ และ (ข) การเล่นเทคนิค Vibrato บนไวโอลิน เอกสารนี้เป็นเอกสารทงสวนไวสำหรับการใช้งานเพื่อการศึกษาทานาน ไม่นอญูญาติเนาไปไซประโยชน์ดานการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



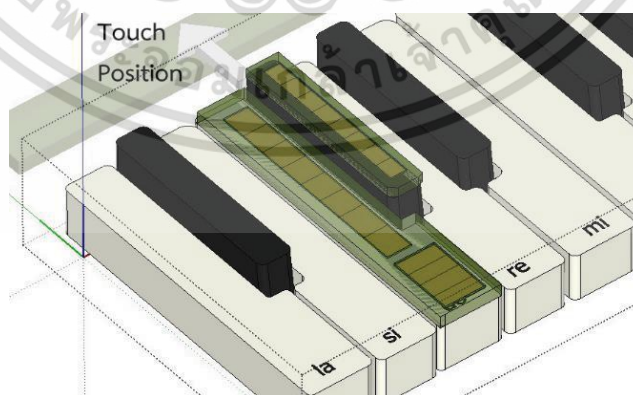
(ก)



(ข)

รูปที่ 1.2 (ก) ล้อคันโยกที่ใช้ในระบบดั้งเดิม, (ข) การใช้งานล้อคันโยกขณะเล่นคีย์บอร์ด

ถึงแม้ว่าการใช้ล้อคันโยกจะทำให้สามารถเล่นเทคนิค Pitch bend และ Vibrato ได้ แต่จะให้ความรู้สึกไม่เป็นธรรมชาติเหมือนกับขณะที่เล่นบนเครื่องสาย และยังเกิดข้อเสียที่นักดนตรีจะต้องใช้มือข้างหนึ่งสำหรับควบคุมล้อคันโยกซึ่งแสดงไว้ดังรูปที่ 1.2ข ทำให้ระบบดั้งเดิมจำกัดเทคนิคการเล่นโน้ตที่ใช้สองมือพร้อมกัน เช่น การเล่นโน้ตเบส (Baseline) บนมือหนึ่งกับเล่นโน้ตทำนองหลัก (Melody) ด้วยมืออีกข้างหนึ่งไปพร้อมๆกันขณะที่มีการเรียกใช้ฟังก์ชัน งานวิจัยนี้จึงนำเสนอระบบควบคุมการใช้งานฟังก์ชัน Pitch bend และ Vibrato ผ่านการสัมผัสปลายนิ้วบนลิ้มคีย์บอร์ดโดยตรง โดยระบบที่นำเสนอจะมีการติดตั้งเซ็นเซอร์วัดตำแหน่งการสัมผัสบนลิ้มคีย์บอร์ด แสดงลักษณะการติดตั้งด้วยแบบจำลอง 3D ในรูปที่ 1.3 ระหว่างการเล่นเทคนิค Pitch bend และ Vibrato ด้วยการควบคุมลักษณะนี้ จะให้ความรู้สึกคล้ายกับการเล่นบนเครื่องสายจริงๆ ซึ่งให้ความรู้สึกเป็นธรรมชาติและควบคุมง่ายกว่าแบบระบบดั้งเดิม เพิ่มอิสระการเล่นดนตรีแก่นักคีย์บอร์ดระหว่างแสดงดนตรีสด ทำให้เพิ่มความสร้างสรรค์ในงานดนตรีมากขึ้น



รูปที่ 1.3 ลักษณะการติดตั้งแผ่นเซ็นเซอร์วัดการสัมผัสสำหรับระบบที่นำเสนอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์

1.2.1 ศึกษาและออกแบบเซ็นเซอร์วัดตำแหน่งการสัมผัส (Touch sensor) สำหรับติดตั้งบนลิ้มคีย์บอร์ด เพื่อใช้สำหรับทดสอบอัลกอริทึมของระบบควบคุมที่นำเสนอ

1.2.2 นำเสนออัลกอริทึมการประมวลผลสัญญาณดิจิทัล (DSP Algorithm) ต่างๆที่ใช้ในระบบควบคุมฟังก์ชัน Pitch bend และ Vibrato

1.2.3 ทดสอบการทำงานและวัดสมรรถนะ (Performance) ของระบบที่นำเสนอ พร้อมเปรียบเทียบกับสมรรถนะของระบบดั้งเดิม

1.3 ขอบเขตของวิทยานิพนธ์

1.3.1 งานวิจัยนี้จะเน้นเรื่องพัฒนาอัลกอริทึมสำหรับระบบควบคุมฟังก์ชัน Pitch bend และ Vibrato โดยใช้ตำแหน่งการสัมผัสของนิ้วบนลิ้มคีย์บอร์ด

1.3.2 เซ็นเซอร์สัมผัสแบบวัดความจุไฟฟ้า (Capacitive Touch Sensor) จะถูกนำมาใช้สำหรับการทดสอบอัลกอริทึมควบคุมของระบบในงานวิจัยนี้ สามารถเปลี่ยนหรือปรับปรุงได้ในอนาคต

1.3.3 เซ็นเซอร์สัมผัสที่ใช้ทดสอบในงานวิจัยมีทั้งหมด 14 คีย์สำหรับความยาว 2 Octave

1.3.4 เซ็นเซอร์ที่ออกแบบไว้ยังไม่สามารถติดตั้งบนลิ้มคีย์บอร์ดได้จริงเนื่องจากมีขนาดเกินลิ้มคีย์ ซึ่งสามารถปรับแก้ไขได้ในอนาคต แต่ยังคงเพียงพอที่จะทดสอบอัลกอริทึมควบคุมได้

1.4 ขั้นตอนการดำเนินงาน

1.4.1 ค้นหาลักษณะการรูดนิ้วที่เหมาะสมกับการควบคุม Pitch bend และ Vibrato

1.4.2 ออกแบบเซ็นเซอร์ตรวจจับการสัมผัส และกระบวนการสกัดค่าตำแหน่งสัมผัส

1.4.3 ออกแบบอัลกอริทึมควบคุมของระบบ Pitch bend และ Vibrato

1.4.4 ทดสอบการทำงานของระบบที่นำเสนอและเปรียบเทียบสมรรถนะกับระบบดั้งเดิม

1.4.5 วิเคราะห์ผลการทดสอบ และ สรุปผล

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ระบบควบคุมการเล่น Pitch bend และ Vibrato สำหรับเครื่องดนตรีคีย์บอร์ดที่นำเสนอนั้นสามารถนำไปใช้แทนระบบดั้งเดิมที่มีอยู่ได้ เป็นการเพิ่มอิสระและเทคนิคการเล่นคีย์บอร์ดแก่นักดนตรีในระหว่างแสดงดนตรีสดมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

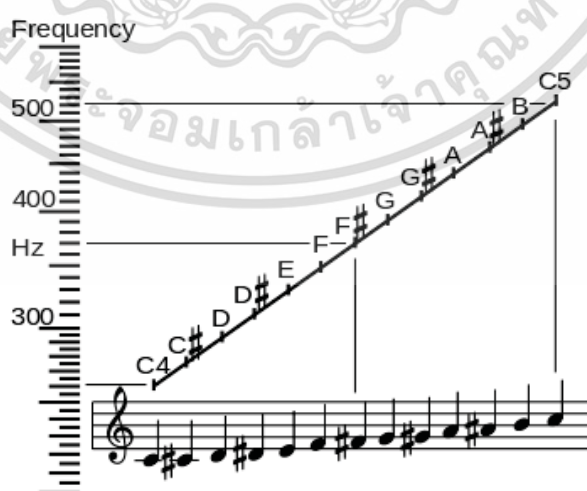
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงทฤษฎีพื้นฐานสำหรับงานวิจัยนี้ ซึ่งได้แก่ ทฤษฎีทางด้านดนตรีที่เกี่ยวข้องกับงานวิจัย พร้อมยกตัวอย่างงานวิจัยในอดีตที่เกี่ยวข้องกับการพัฒนาระบบควบคุม Pitch bend และ Vibrato บนคีย์บอร์ด, หลักการทำงานของเซ็นเซอร์ตรวจจับสัมผัสชนิดวัดความจุไฟฟ้าที่ใช้ในงานวิจัย, ทฤษฎีพื้นฐานการสื่อสารโปรโตคอล MIDI สำหรับควบคุมเครื่องดนตรีอิเล็กทรอนิกส์ต่างๆ และทฤษฎีการประมวลผลสัญญาณดิจิทัล (Digital Signal Processing) ต่างๆ ที่ใช้ในงานวิจัย

2.1 ทฤษฎีด้านดนตรีสำหรับงานวิจัย

2.1.1 ระดับเสียง (Pitch)

คือ ปริมาณการรับรู้ของเสียง (Perceptual of sound) ที่บ่งบอกถึงอัตราการสั่นของแหล่งกำเนิดเสียงซึ่งจะถูกระบุหน่วยปริมาณด้วยความถี่ (Frequency) ของเสียง ระดับโน้ตดนตรี (Pitch of a Note) นั้นจะมีความหมายเดียวกับระดับเสียงที่ได้กล่าวมา โดยระดับโน้ตนั้นจะมีความสัมพันธ์กับความถี่ของเสียงแสดงดังแผนภาพในรูปที่ 2.1

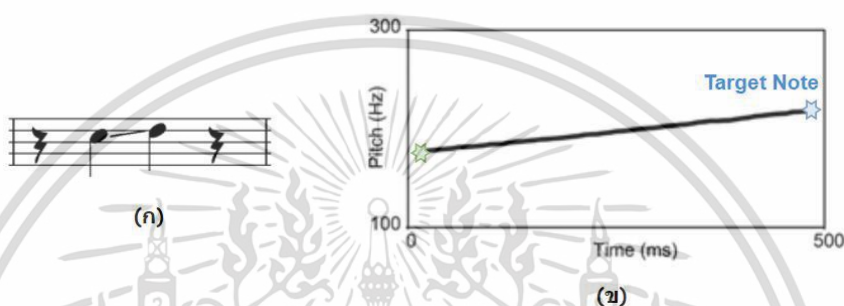


รูปที่ 2.1 ความสัมพันธ์ระหว่างความถี่ของเสียงกับตำแหน่งโน้ตบนบรรทัดห้าเส้น (Staff)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 Pitch bend

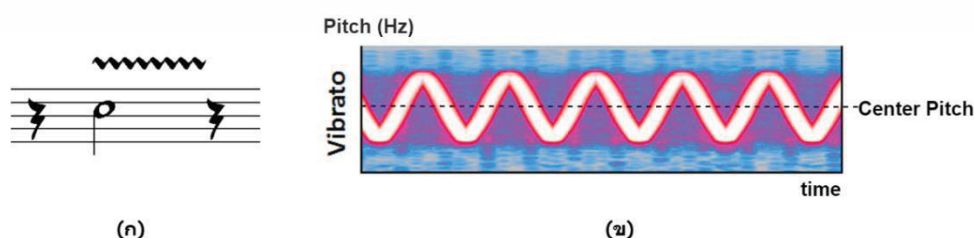
คือ เทคนิคการเล่นดนตรีที่ทำให้ระดับโน้ต (Pitch) ที่ถูกเล่นอยู่เปลี่ยนแปลงในลักษณะแบบต่อเนื่อง ซึ่งจะเพิ่มขึ้นหรือลดลงไปยังระดับโน้ตของตำแหน่งโน้ตเป้าหมายใหม่ (Target Note) ทำให้ความถี่ของโน้ตเปลี่ยนแปลงอย่างต่อเนื่องไปจากความถี่โน้ตเริ่มต้น สัญลักษณ์ทางดนตรีสากลสำหรับระบุงการเล่น Pitch bend และลักษณะการเปลี่ยนแปลงระดับโน้ตระหว่างการเล่น Pitch bend นั้นแสดงดังรูปที่ 2.2ก และ 2.2ข ตามลำดับ



รูปที่ 2.2 (ก) สัญลักษณ์ทางดนตรีสากลสำหรับระบุงการเล่น Pitch bend, (ข) ลักษณะการเปลี่ยนแปลงของระดับเสียงโน้ตระหว่างการเล่น Pitch bend

2.1.3 Vibrato

คือ เทคนิคการเล่นดนตรีที่ทำให้เกิดการเปลี่ยนแปลงระดับโน้ตที่ถูกเล่นอยู่เล็กน้อย ซึ่งโดยปกติจะไม่เกิน 1 ครึ่งเสียง (Semitone) ในลักษณะขึ้นลงเป็นจังหวะต่อเนื่องเหมือนลักษณะของการกวัดแกว่ง (Oscillation) สัญลักษณ์ทางดนตรีสากลสำหรับระบุงการเล่น Vibrato และลักษณะการเปลี่ยนแปลงระดับโน้ตระหว่างการเล่น Vibrato นั้นแสดงดังรูปที่ 2.3ก และ 2.3ข ตามลำดับ



รูปที่ 2.3 (ก) สัญลักษณ์ทางดนตรีสากลสำหรับระบุงการเล่น Vibrato, (ข) ลักษณะการเปลี่ยนแปลงของระดับเสียงโน้ตระหว่างการเล่น Vibrato

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4 ความสำคัญทางด้านดนตรีของเทคนิค Pitch bend และ Vibrato

การเพิ่ม Vibrato ให้กับโน้ตที่เล่นอยู่จะช่วยเพิ่มอารมณ์ต่างๆและความรู้สึกดูดีในการฟัง (Expressiveness, Soulfulness) ตัวอย่างที่พบบ่อยเช่น การร้องเพลง Opera การเล่นกีตาร์โซโล่ (ตัวอย่างศิลปินที่ใช้เทคนิค Vibrato บ่อย เช่น Paul Kossoff, BB King) ทำให้เสียงที่ได้ออกมา มีเสน่ห์และสามารถรับรู้ถึงความรู้สึกที่ต้องการจะสื่อออกมาได้มากขึ้น ส่วน Pitch bend ก็มีความสำคัญเช่นเดียวกับ Vibrato โดยจะพบเห็นบ่อยๆในแนวดนตรี Jazz และ Blue ผ่านการร้องหรือเล่นบนเครื่องสายและเครื่องเป่าลมต่างๆ

2.1.5 การรับรู้ความแตกต่างของระดับโน้ตใน Musical Scale ของมนุษย์

จากการศึกษาทางด้านจิตวิทยาศาสตร์ (Psychoacoustic) เกี่ยวกับผลของระดับโน้ตในสเกลดนตรี (Musical Scale) ต่อการรับรู้จากการได้ยิน (Auditory Perception) ของมนุษย์ พบว่าการรับรู้ระดับโน้ตจะมีความสัมพันธ์ในลักษณะลอการิทึม (Logarithm) กับความถี่ของโน้ต [3] จากผลดังกล่าว ทำให้มีการกำหนดมาตรฐานการตั้งค่าระดับโน้ตสำหรับดนตรีสากลดังกราฟความสัมพันธ์ในรูปที่ 2.4 โดยสามารถอธิบายความสัมพันธ์ด้วยสมการที่ 2.1 ซึ่งความสัมพันธ์นี้จะถูกนำไปใช้ในมาตรฐานของระบบ MIDI (MIDI Standard) ด้วย โดยกำหนดให้ f คือความถี่ของโน้ต และ n คือเลขตำแหน่งโน้ต (Pitch number) เช่น ตำแหน่งโน้ต A4 จะมีค่า $n = 69$ และ $f = 440$ Hz [4]

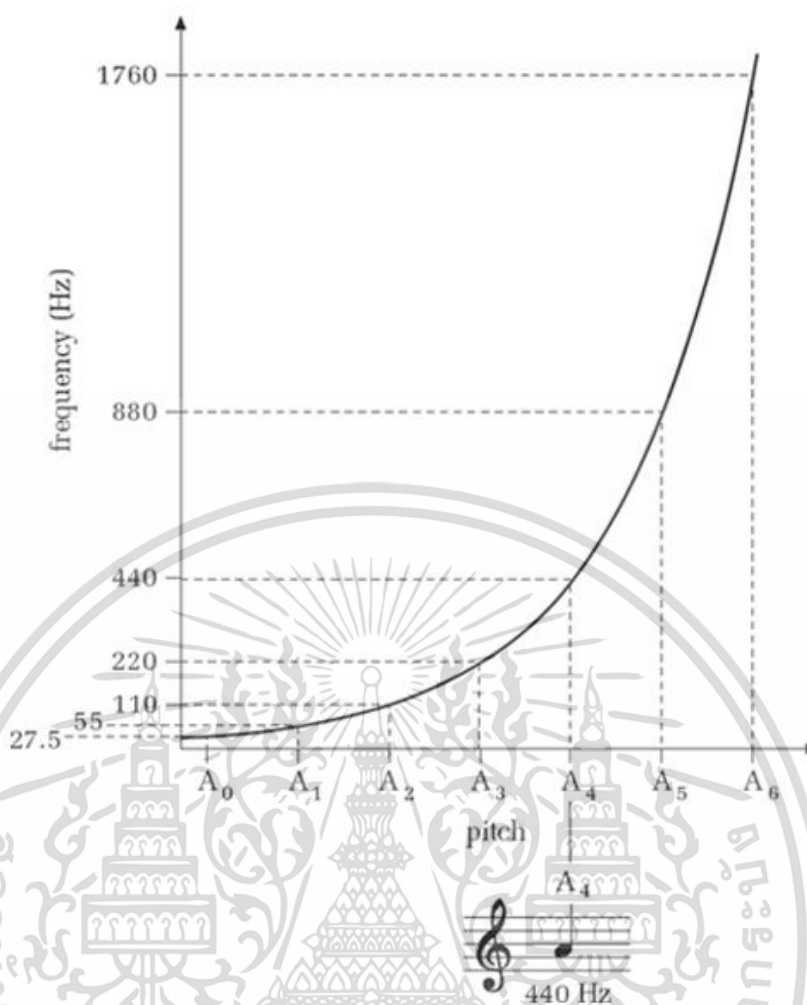
ดังนั้น การรับรู้ความแตกต่างของระดับโน้ตนั้น ควรระบุโดยใช้ค่าความแตกต่างของเลขตำแหน่งโน้ต n (Δn) แทนการวัดความแตกต่างของความถี่โน้ต f (Δf) เพื่อสร้างการวัดการรับรู้ความแตกต่างระดับโน้ตให้มีลักษณะเป็นเชิงเส้น (Linearization) ไม่ขึ้นอยู่กับตำแหน่งโน้ตโดยใช้สมการที่ 2.2 โดยที่ $\Delta n = 0.5$ จะหมายถึงความต่างระดับโน้ตขนาด 1 Semitone

$$n = 69 + 12 \log_2 \left(\frac{f}{440 \text{ Hz}} \right) \quad (2.1)$$

$$\Delta n = 12 \log_2 \left(\frac{f_1}{f_2} \right) \quad (2.2)$$

โดยที่ $\Delta n = n_1 - n_2$ และ $\Delta f = f_1 - f_2$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

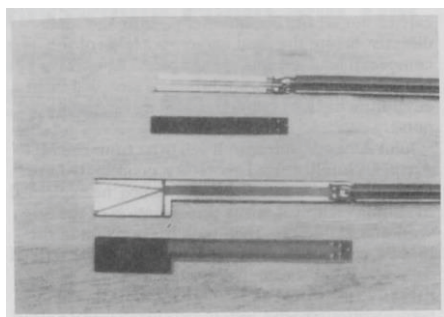


รูปที่ 2.4 ความสัมพันธ์ในลักษณะลอการิทึม (Logarithm) ระหว่างระดับโน้ตกับความถี่ของโน้ต

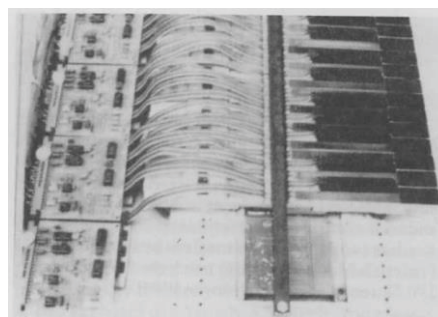
2.2 งานวิจัยที่เกี่ยวข้อง

ในปี 1982 บริษัท Moog Music (Moog Music, Inc) ได้ตั้งโปรเจกต์ที่จะพัฒนา Keyboard โดยมีระบบที่เรียกว่า Multiply-Touch-Sensitive (MTS) [5] ซึ่งระบบนี้จะใช้แผ่นเซ็นเซอร์ตรวจจับการสัมผัสสำหรับติดตั้งบนลิ้มคีย์ แต่เนื่องจากเทคโนโลยีของเซ็นเซอร์สัมผัสนั้นเป็นแบบ Resistive Film ทำให้แรงกดของนิ้วผู้เล่นบนลิ้มคีย์มีผลต่อการระบุตำแหน่งสัมผัสค่อนข้างมาก เมื่อทดสอบแล้วปรากฏว่าระบบใช้งานยาก ผู้เล่นต้องใช้เวลาฝึกฝนอย่างมากเพื่อให้สามารถเล่นคีย์บอร์ดของระบบนี้ได้อย่างชำนาญได้ โปรเจกต์นี้จึงถูกระงับและไม่นำมาผลิตเป็นสินค้าต่อไป แสดงลักษณะของแผ่นเซ็นเซอร์และการติดตั้งบนเครื่องดนตรีดังรูปที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก)



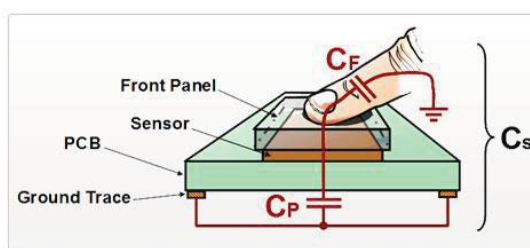
(ข)

รูปที่ 2.5 (ก) แผ่นเซ็นเซอร์ตรวจจับการสัมผัสชนิด Resistive Film สำหรับติดตั้งบนลิ้มคีย์ของระบบ MTS, (ข) การติดตั้งระบบ MST บนเครื่องดนตรี Organ-style Keyboard

2.3 หลักการทำงานของเซ็นเซอร์สัมผัสชนิดวัดความจุไฟฟ้า

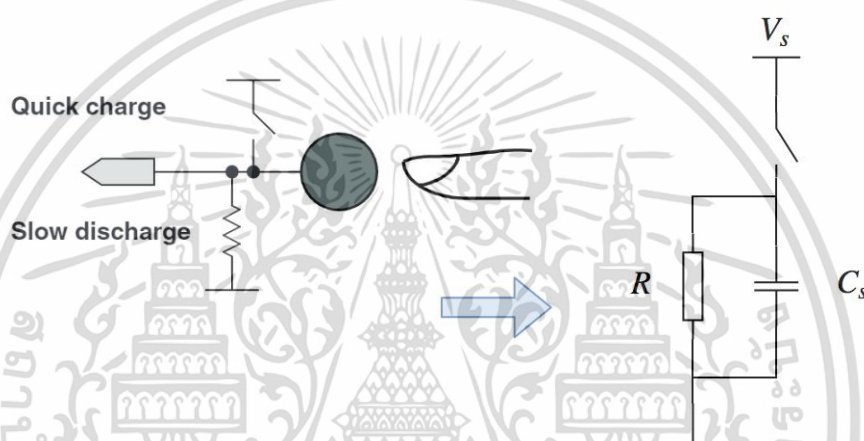
เซ็นเซอร์สัมผัสชนิดวัดความจุไฟฟ้า (Capacitive Touch Sensor) นั้นทำงานโดยอาศัยหลักการตรวจจับการเปลี่ยนแปลงความจุไฟฟ้าแผงระหว่างแผ่นขั้วสัมผัส (Sensor Pad) กับกราวด์ (Ground) ที่ถูกขั้วกลางด้วยฉนวนดังรูปที่ 2.6 ซึ่งนิยมสร้างจากแผ่นวงจรไฟฟ้า (Printed Circuit Board : PCB) หากกำหนดให้ C_S คือค่าความจุไฟฟ้าแผงของเซ็นเซอร์ขณะใดๆ และ C_P คือความจุไฟฟ้าแผงของเซ็นเซอร์ขณะที่ไม่มีการสัมผัส เมื่อมีการป้อนแรงดันไฟฟ้าแก่แผ่นสัมผัส หากนิ้วมือเคลื่อนที่เข้ามาใกล้ๆ หรือสัมผัส Front Panel ของแผ่นขั้วสัมผัสแล้ว สนามไฟฟ้าระหว่างแผ่นขั้วสัมผัสกับกราวด์จะถูกรบกวนและก่อให้เกิดการเพิ่มขึ้นของค่าความจุไฟฟ้าแผงที่เกิดจากผลของนิ้วมือ C_F ส่งผลให้ความจุไฟฟ้าแผงของเซ็นเซอร์ C_S มีค่าเพิ่มขึ้น [1], [2] โดยแสดงความสัมพันธ์ระหว่าง C_S , C_P และ C_F ดังสมการที่ 2.3

$$C_S = C_F + C_P \quad (2.3)$$



รูปที่ 2.6 แบบจำลองสำหรับอธิบายหลักการของเซ็นเซอร์สัมผัสชนิดวัดความจุไฟฟ้า เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น เมื่อนำมาใช้ในเชิงพาณิชย์หรือการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นการสัมผัสจะสามารถถูกระบุได้โดยใช้วงจรอิเล็กทรอนิกส์ที่สามารถตรวจจับการเปลี่ยนแปลงของค่าความจุไฟฟ้าแฝง C_S ส่วนเทคนิคโดยพื้นฐานของวงจรถูกกล่าวว่ามีหลากหลาย เช่น การใช้วงจร Relaxation Oscillator เป็นต้น แต่เทคนิค RC Charge and Discharge Method [1] จะถูกนำมาใช้ในงานวิจัยนี้ ซึ่งใช้หลักการของการวัด Time Constant ของวงจร RC โดยทำการตรวจจับการเปลี่ยนแปลงกราฟการชาร์ตประจุหรือคายของวงจร RC ผ่านการควบคุมการชาร์ตด้วย I/O ของวงจรดิจิทัลที่เปรียบเสมือนสวิตช์ แสดงโครงสร้างของวงจรถูกที่ 2.7 โดยความจุไฟฟ้าแฝงของเซ็นเซอร์ C_S นั้นจะเป็นส่วนหนึ่งในวงจร RC ดังกล่าว



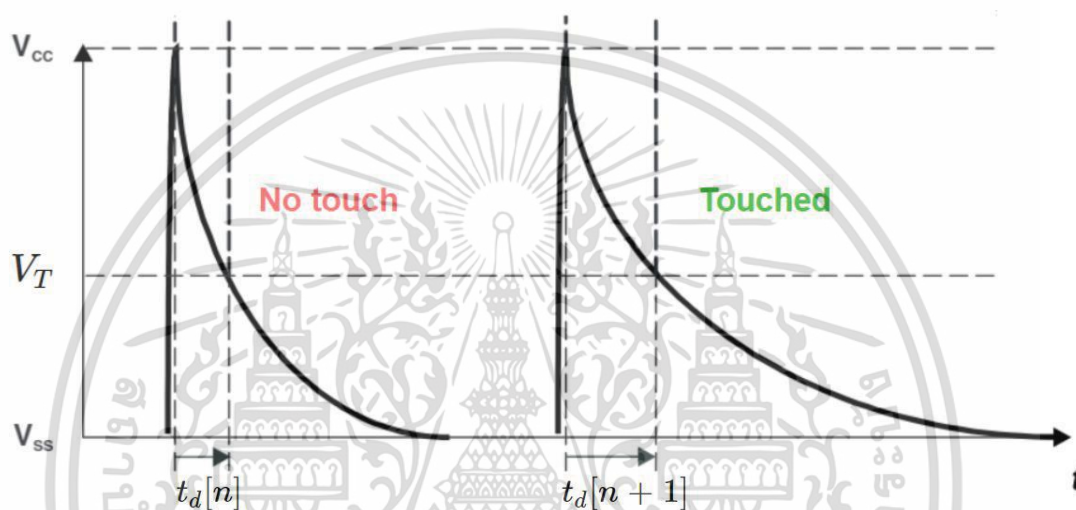
รูปที่ 2.7 วงจรตรวจจับการเปลี่ยนแปลงค่า C_S ด้วยเทคนิค RC Charge and Discharge Method

ลำดับขั้นตอนการทำงานของวงจรสำหรับ RC Charge and Discharge Method มีดังนี้

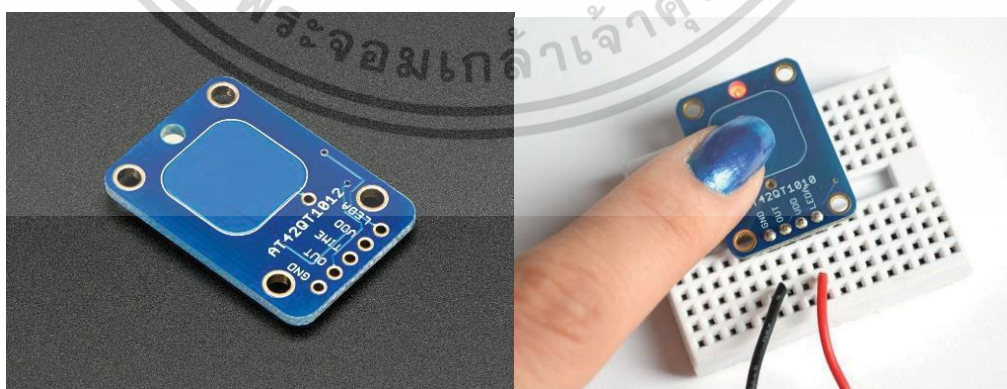
- 1) ชาร์ตประจุวงจร RC โดยทำการปิดสวิตช์เพื่อเชื่อมต่อแรงดันไฟฟ้า V_S เข้ากับวงจร RC
- 2) รอจนกว่าวงจร RC จะถูกชาร์ตจนเต็มแล้ว ($V_S = V_{RC}$)
- 3) เริ่มคายประจุวงจร RC โดยทำการเปิดสวิตช์ และเริ่มการจับเวลาและบันทึกเวลา t_0 สำหรับเป็นตัวระบุเวลาเริ่มต้นการคายประจุ
- 4) รอจนกว่าวงจร RC จะคายประจุจนแรงดันตกคร่อมวงจร RC ตกลงมาจนต่ำกว่าแรงดันขีดเริ่ม (Threshold Voltage) V_T ที่กำหนดไว้ ($V_{RC} < V_T$) เมื่อแรงดันตกลงต่ำกว่า V_T แล้ว ให้บันทึกเวลา ณ ขณะนั้นเท่ากับ t และหยุดการจับเวลา
- 5) คำนวณช่วงเวลาในการคายประจุ (Discharge Duration) $t_d = t - t_0$ และเอาต์พุตค่า t_d สำหรับเป็นผลลัพธ์ของกระบวนการ RC Charge and Discharge Method

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากกระบวนการที่กล่าวมานั้น จำเป็นต้องใช้วงจรนับเวลา (Timer) และวงจรเปรียบเทียบแรงดัน (Comparator) ด้วย โดยเอาต์พุตของกระบวนการที่นำมาใช้ระบุการเปลี่ยนแปลงของกราฟการคายประจุวงจร RC คือช่วงเวลาการคายประจุ t_d ซึ่งแปรผันเป็นสัดส่วนกับค่าความจุไฟฟ้าแฝงของเซ็นเซอร์ C_S ดังนั้น ถ้าหากมีการสัมผัสบนเซ็นเซอร์แล้ว ระยะเวลาการคายประจุจะเพิ่มขึ้นจากเดิมโดยแสดงด้วยกราฟการคายประจุดังรูปที่ 2.8 ทำให้สามารถตัดสินได้ว่าการสัมผัสเกิดขึ้นหรือไม่โดยนำค่า t_d ที่ได้มาผ่านฟังก์ชันขีดเริ่ม (Threshold Function)



รูปที่ 2.8 กราฟการคายประจุของกระบวนการ RC Charge and Discharge Method ซึ่งใช้เปรียบเทียบความแตกต่างของค่าสู่มวัดของช่วงเวลาการคายประจุระหว่างขณะที่ไม่มีการสัมผัส $t_d[n]$ กับมีการสัมผัส $t_d[n+1]$ สังเกตได้ว่า $t_d[n+1] > t_d[n]$



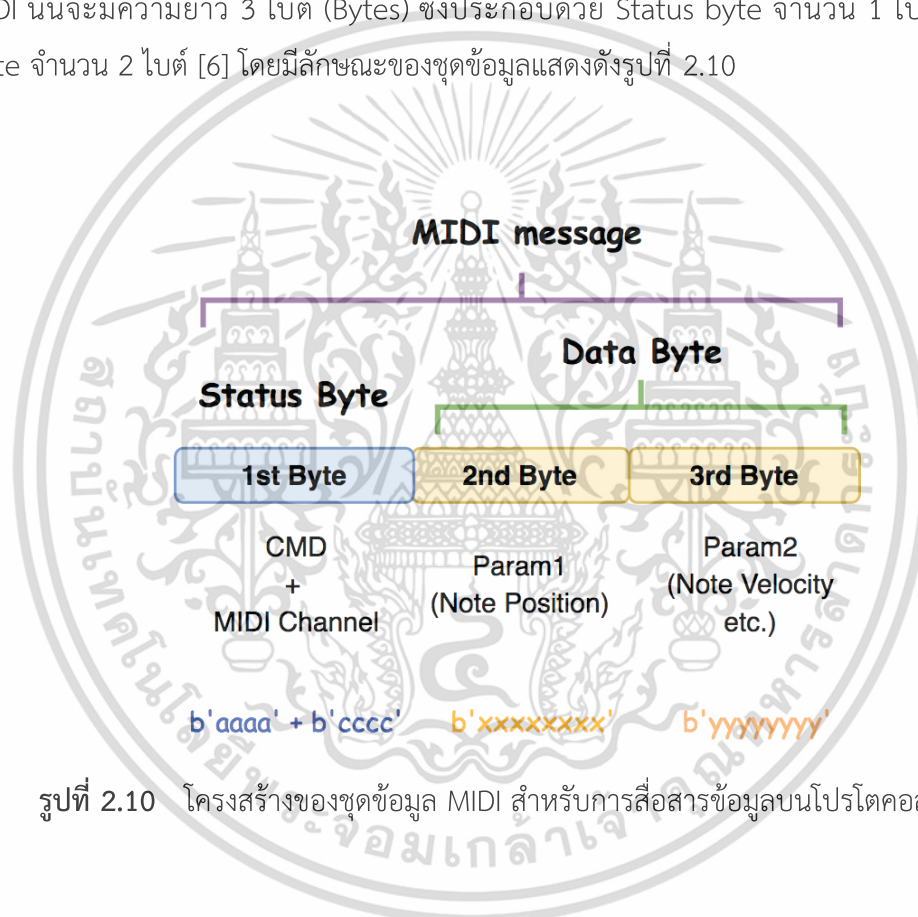
รูปที่ 2.9 ตัวอย่างโมดูลเซ็นเซอร์สัมผัสแบบวัดความจุไฟฟ้าที่มีขายตามตลาดอิเล็กทรอนิกส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 การสื่อสารบนโปรโตคอล MIDI

2.4.1 โปรโตคอล MIDI

MIDI (Musical Instrument Digital Interface) เป็นโปรโตคอล (Protocol) การสื่อสารข้อมูล ดิจิตอลบนเครื่องดนตรีอิเล็กทรอนิกส์ โดยชุดข้อมูลสำหรับการสื่อสารจะประกอบด้วย ข้อมูลต่างๆ ของโน้ตที่กำลังเล่นอยู่ เช่น สถานะของการเล่น (Event) ตำแหน่งของโน้ต (Pitch) ความดังของโน้ต (Velocity) เป็นต้น การสื่อสารอนุกรมแบบ Asynchronous (UART 8 bits, no parity bit, 1 stop bit) จะถูกใช้ในระบบ MIDI ด้วยความเร็วการส่งข้อมูลเท่ากับ 31250 bps โดยชุดข้อมูลในระบบ MIDI นั้นจะมีความยาว 3 ไบต์ (Bytes) ซึ่งประกอบด้วย Status byte จำนวน 1 ไบต์ และ Data byte จำนวน 2 ไบต์ [6] โดยมีลักษณะของชุดข้อมูลแสดงดังรูปที่ 2.10



รูปที่ 2.10 โครงสร้างของชุดข้อมูล MIDI สำหรับการสื่อสารข้อมูลบนโปรโตคอล MIDI

โครงสร้างชุดข้อมูล MIDI ดังรูปที่ 2.10 จะถูกเรียกสั้นๆว่า “MIDI Message” โดยจะอธิบายหน้าที่ของแต่ละไบต์ใน MIDI Message ดังนี้

1) Status Byte

คือ ไบต์ที่ใช้กำหนดสถานะของอุปกรณ์ MIDI ซึ่งประกอบด้วยข้อมูล 2 อย่างซึ่งจะถูกแบ่งอย่างละครึ่งไบต์ (4 บิต) ได้แก่ ชุดคำสั่ง (Command : CMD) เช่น คำสั่ง Note on/Note off, Select Instrument เป็นต้น และ หมายเลขช่องทางเชื่อมต่อของ MIDI (MIDI Channel) ที่ชุดคำสั่ง

ดังกล่าวกระทำซึ่งมีช่องทางรองรับมากถึง 16 ช่อง

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือสงวนชื่อเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) Data Byte

คือ ไบต์ที่ใช้กำหนดพารามิเตอร์ (Parameter) ต่างๆ ที่สอดคล้องกับคำสั่งจาก Status Byte ซึ่ง Data Byte จะมีขนาดเท่ากับ 2 ไบต์ ตัวอย่างพารามิเตอร์ต่างๆที่ใช้กับชุดคำสั่ง เช่น ตำแหน่งโน้ต (Note Position) ที่ถูกคำสั่งกระทำ ระดับความดังของโน้ต (Velocity) ระดับการเบนโน้ตสำหรับคำสั่ง Pitch bend เป็นต้น

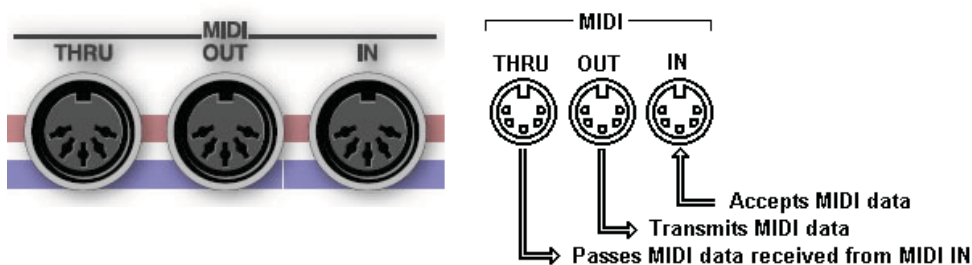
Status Byte	Data Byte 1	Data Byte 2	Message	Legend
1000nnnn	0kkkkkkk	0vvvvvvv	Note Off	n=channel, k=key (0-127), v=velocity (0-127)
1001nnnn	0kkkkkkk	0vvvvvvv	Note On	n=channel, k=key (0-127), v=velocity (0-127)
1010nnnn	0kkkkkkk	0ppppppp	Poly Key Pressure	n=channel, k=key (0-127), p=pressure (0-127)
1100nnnn	0ppppppp	[none]	Program Change	n=channel, p=preset number (0-127)
1101nnnn	0ppppppp	[none]	Channel Pressure	n=channel, p=pressure (0-127)
1110nnnn	0ffffff	0cccccc	Pitch Bend	n=channel, c=coarse, f=fine (c+f = 14-bit resolution)

ตารางที่ 2.1 ตัวอย่าง MIDI message พื้นฐานที่ใช้กันบ่อยในระบบ MIDI

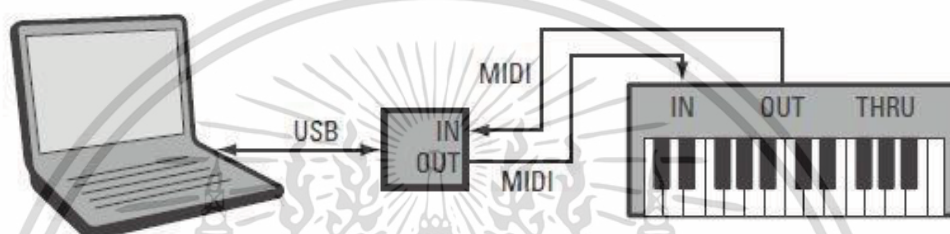
2.4.2 การเชื่อมต่อฮาร์ดแวร์ของอุปกรณ์ MIDI

การเชื่อมต่อระหว่างอุปกรณ์ MIDI ด้วยกันนั้น จะใช้พอร์ตซึ่งจะมีประกอบด้วย 3 พอร์ต ได้แก่ MIDI IN (สำหรับรับข้อมูลเข้า), MIDI OUT (สำหรับส่งข้อมูลออก), MIDI Thru (สำหรับการเชื่อมต่อแบบ Daisy Chain) ซึ่งแสดงลักษณะของพอร์ตไว้ดังรูปที่ 2.11 หากต้องการเชื่อมต่อระหว่างอุปกรณ์ MIDI จำนวน 2 ชิ้น ให้เชื่อมต่อในลักษณะเหมือนกับที่ทำบนการสื่อสารแบบ UART โดยให้ MIDI IN ของอุปกรณ์ที่ 1 เชื่อมต่อกับ MIDI OUT ของอุปกรณ์ที่ 2 ตามรูปที่ 2.12 แต่ถ้าหากต้องเชื่อมต่อระหว่างอุปกรณ์ที่มีมากกว่า 2 ชิ้นขึ้นไป จำเป็นจะต้องใช้งานพอร์ต MIDI Thru โดยให้ต่ออุปกรณ์ในลักษณะ Daisy Chain แสดงไว้ดังรูปที่ 2.13

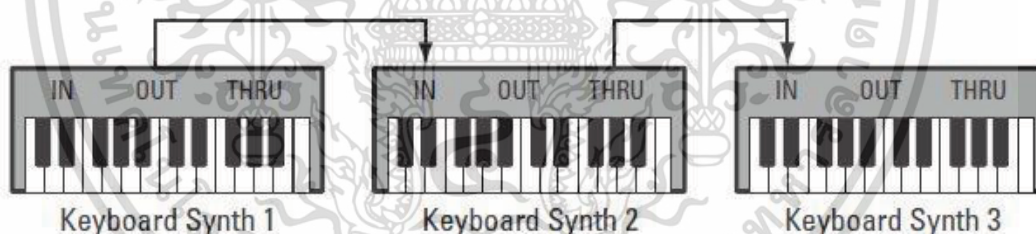
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 ลักษณะของพอร์ตบนอุปกรณ์ MIDI



รูปที่ 2.12 ตัวอย่างการเชื่อมต่อระหว่างอุปกรณ์ MIDI จำนวน 2 ชั้น

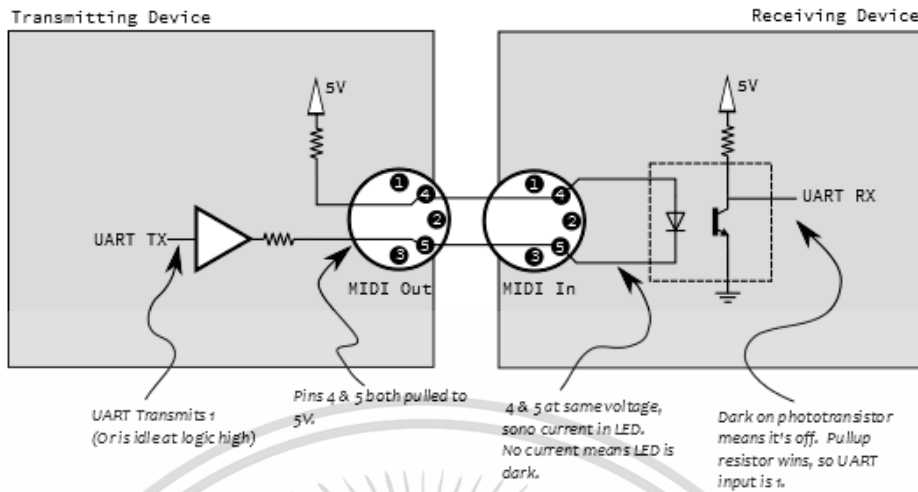


รูปที่ 2.13 ตัวอย่างการเชื่อมต่อระหว่างอุปกรณ์ MIDI ที่มากกว่า 2 ชั้นในลักษณะ Daisy Chain

วงจรรีเลย์ทรอนิกส์ที่ใช้สำหรับการส่งข้อมูลระหว่างพอร์ตนั้นจะใช้เป็นลักษณะของ Current Loop ที่มีการแยกระบบกราวด์ด้วยอุปกรณ์ Optocoupler เพื่อให้ระบบสามารถสื่อสารผ่านสายในระยะไกลๆได้แม้ว่าสัญญาณรบกวนที่เข้ามาจะมากก็ตาม โดยจะแสดงวงจรที่ใช้ตามมาตรฐานของอุปกรณ์ MIDI ดังรูปที่ 2.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SENDING LOGICAL HIGH

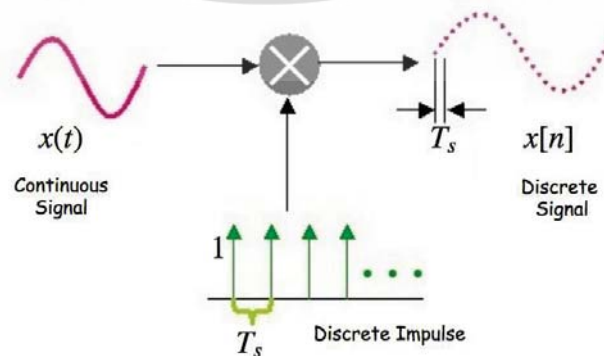


รูปที่ 2.14 รายละเอียดของวงจรการสื่อสารที่ใช้ภายในพอร์ต MIDI พร้อมอธิบายการส่งสัญญาณ logic HIGH จากตัวส่ง (Transmitter) ไปยังตัวรับ (Receiver)

2.5 ทฤษฎีการประมวลผลสัญญาณดิจิทัลพื้นฐานที่ใช้ในงานวิจัย

2.5.1 ทฤษฎีการสุ่มค่าตัวอย่างของ Nyquist (Nyquist Sampling Theorem)

การสุ่มตัวอย่างในการประมวลผลสัญญาณดิจิทัล (Digital Signal Processing) นั้น หมายถึง การสุ่มวัดค่าสัญญาณต่อเนื่องทางเวลา (Continuous Signal) $x(t)$ ทุกๆช่วงเวลา $t = nT_s$ โดยที่ T_s คือคาบเวลาการสุ่ม (Sampling Period) และ n เป็นจำนวนเต็ม และใช้ค่าที่ได้จากการสุ่มวัดซึ่งเป็นสัญญาณที่ไม่ต่อเนื่องทางเวลา (Discrete Signal) $x[n]$ เป็นตัวแทนของสัญญาณต่อเนื่องนั้น เพื่อนำไปใช้กับกระบวนการประมวลผลสัญญาณดิจิทัลต่อไปได้



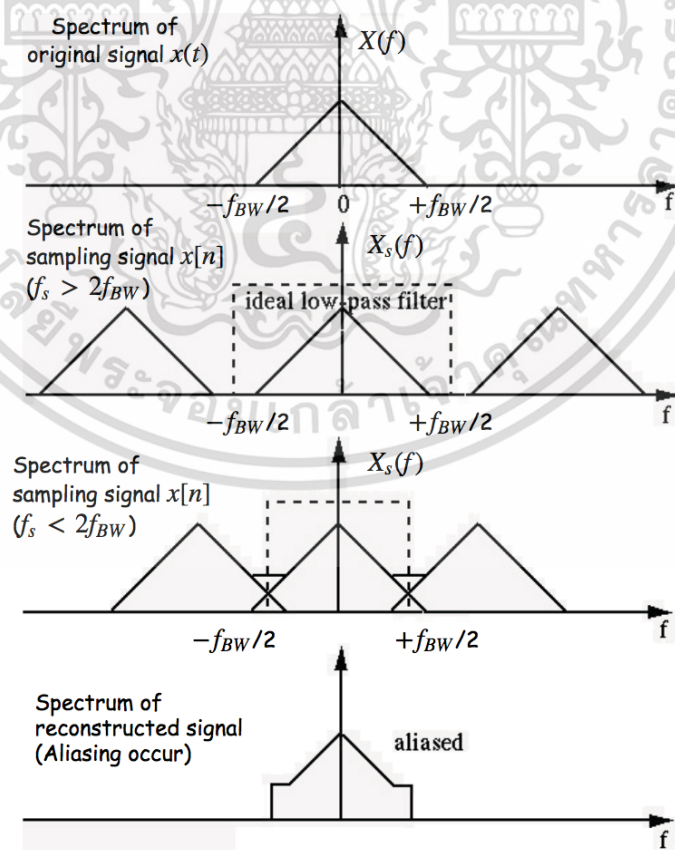
รูปที่ 2.15 กระบวนการสุ่มค่าตัวอย่างของสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติเห็นาไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทฤษฎีการสุ่มค่าตัวอย่างของ Nyquist จะกล่าวว่า “สำหรับสัญญาณต่อเนื่อง $x(t)$ ที่มีแถบความถี่จำกัดหรือแบนด์วิธ (Bandwidth) เท่ากับ f_{BW} ที่ถูกสุ่มสัญญาณไม่ต่อเนื่อง $x[n]$ แล้วจะสามารถฟื้นฟู (Reconstruct) สัญญาณ $x(t)$ กลับมาใหม่จากสัญญาณแบบสุ่มได้อย่างสมบูรณ์ก็ต่อเมื่อ ความถี่การสุ่มค่าตัวอย่าง (Sampling Frequency) f_s การสร้างจะต้องมากกว่าหรือเท่ากับสองเท่าของแบนด์วิธของสัญญาณต่อเนื่องดั้งเดิม” [7] ดังสมการที่ 2.4 โดยที่ $f_s = 1/T_s$

$$f_s \geq 2f_{BW} \quad (2.4)$$

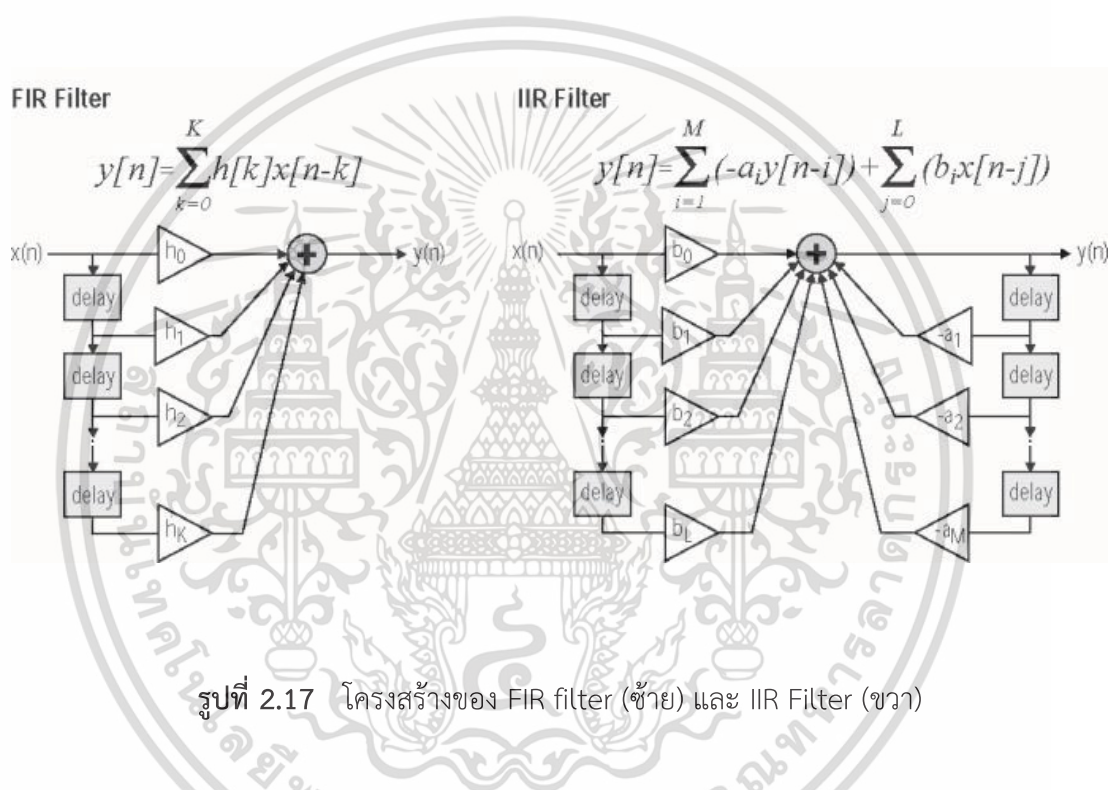
หากการสุ่มค่าตัวอย่างไม่เข้าเงื่อนไขของ Nyquist แล้ว จะเกิดการทับซ้อนแถบความถี่ (Aliasing) ซึ่งมีผลทำให้ข้อมูลสเปกตรัมของสัญญาณที่ได้จากการสุ่มผิดเพี้ยน (Distortion) ไปจากสัญญาณต่อเนื่องดั้งเดิม แสดงการการทับซ้อนแถบความถี่ดังรูปที่ 2.16 สัญญาณ $x[n]$ ขณะที่เกิด Aliasing จึงไม่สามารถเป็นตัวแทนของสัญญาณ $x(t)$ ได้อย่างแท้จริง



รูปที่ 2.16 ผลกระทบของ Aliasing ต่อสเปกตรัมของสัญญาณที่ได้จากกระบวนการฟื้นฟูสัญญาณ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 ตัวกรองความถี่ดิจิทัลชนิด IIR (IIR Filter)

หากแบ่งประเภทตัวกรองความถี่โดยใช้โครงสร้างจากการ Implement เป็นเกณฑ์ จะสามารถแบ่งออกเป็น 2 ประเภทคือ ตัวกรองแบบ Finite Impulse Response (FIR Filter) และ ตัวกรองแบบ Infinite Impulse Response (IIR Filter) โดย FIR Filter จะใช้กลไกของ Convolution ในการคำนวณเอาต์พุต ส่วน IIR Filter นั้นจะใช้กลไกของการป้อนกลับ (Feedback) เอาต์พุตของอดีตมาใช้คำนวณเอาต์พุตใหม่เหมือนการคำนวณในลักษณะ Recursive Function โดยแสดงโครงสร้างของ FIR filter และ IIR Filter ดังรูปที่ 2.17



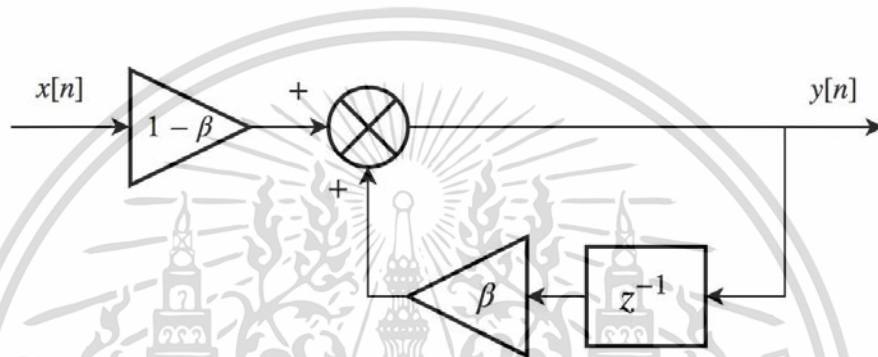
สำหรับงานวิจัยนี้ การประมวลผลสัญญาณในบางส่วนต้องการปรับแต่งสัญญาณให้เรียบขึ้น (Smoothing) หรือต้องการกำจัดสัญญาณรบกวนที่เข้ามาพร้อมกับสัญญาณอินพุต (Noise Filtering) พบว่า IIR Filter นั้นเหมาะสมและพอเพียงสำหรับงานดังกล่าวมา ซึ่ง IIR Filter จะให้ประสิทธิภาพทางการคำนวณ (Computation Efficiency) สูงกว่า FIR Filter เนื่องจากใช้จำนวน Order น้อยกว่า โดยตัวอย่างตัวกรองความถี่สำหรับปรับแต่งให้สัญญาณเรียบที่ใช้ในงานวิจัยซึ่งแสดงดัง Transfer function และ Difference Equation สำหรับฟังก์ชันนั้นดังสมการที่ 2.5 และ 2.6 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$H = \frac{1 - \beta}{1 - \beta z^{-1}} \quad (2.5)$$

$$y[n] = \beta y[n-1] + (1 - \beta)x[n] \quad (2.6)$$

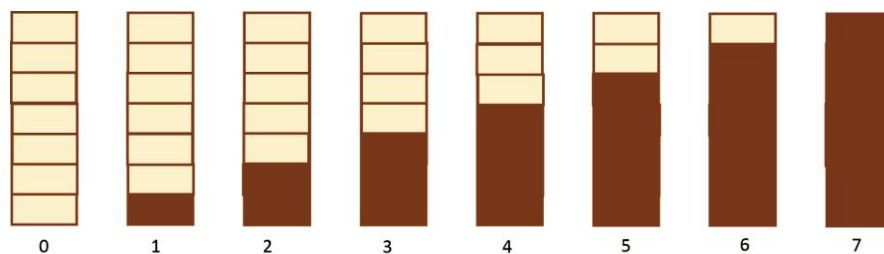
โดยที่ $0 \leq \beta \leq 1$, $x[n]$ คือ อินพุตของตัวกรอง และ $y[n]$ คือ เอาต์พุตของตัวกรอง



รูปที่ 2.18 บล็อกไดอะแกรมของ 1st-Order IIR Filter ตัวอย่างสำหรับใช้เป็น Smoothing Filter

2.5.3 การเข้ารหัส Unary (Unary Coding)

บางครั้งถูกเรียกว่า การเข้ารหัสเทอร์โมมิเตอร์ (Thermometer Code) ซึ่งเป็นการเข้ารหัสข้อมูลที่มีกลไกลักษณะเหมือนการระบุระดับเทอร์โมมิเตอร์ดังรูปที่ 2.19 หากข้อมูลอินพุตเป็นระดับของจำนวนเต็มในลักษณะของ Binary Code ที่มีความละเอียดของบิต (Resolution) เท่ากับ N บิตแล้ว จะสามารถแปลงข้อมูลให้อยู่ในรูปแบบของระดับเทอร์โมมิเตอร์ได้โดยความละเอียดใหม่ที่ได้นั้นจะเปลี่ยนไปเท่ากับ $2^N - 1$ บิต แสดงตัวอย่างการเข้ารหัส Unary ดังรูปที่ 2.20



รูปที่ 2.19 ระดับเทอร์โมมิเตอร์ที่มีการแบ่งระดับจำนวน 7 ระดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Decimal	Binary			Thermometer Code						
	b_1	b_2	b_3	d_1	d_2	d_3	d_4	d_5	d_6	d_7
0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	1
2	0	1	0	0	0	0	0	0	1	1
3	0	1	1	0	0	0	0	1	1	1
4	1	0	0	0	0	0	1	1	1	1
5	1	0	1	0	0	1	1	1	1	1
6	1	1	0	0	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1

รูปที่ 2.20 ตัวอย่างการเข้ารหัส Unary ตามลักษณะของข้อมูลอินพุต 8 ระดับจากรูปที่ 2.19

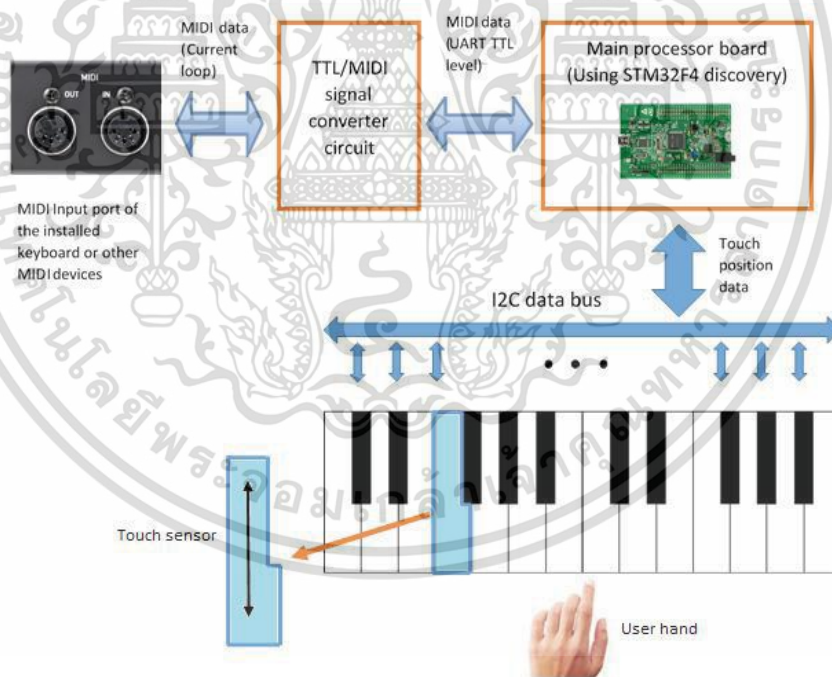
จากกระบวนการเข้ารหัสที่ได้กล่าวมา หากทำการเข้ารหัสแบบย้อนกลับโดยให้อินพุตเป็นข้อมูลลักษณะ Unary code ที่มีความละเอียด $2^N - 1$ บิตแทน เอาต์พุตที่ได้จะมีความละเอียดเท่ากับ N บิตซึ่งใช้จำนวนบิตน้อยกว่าเดิม ดังนั้นการเข้ารหัส Unary จะสามารถนำมาใช้ในการบีบอัดข้อมูล (Data Compression) ได้โดยไม่มีการสูญเสียข้อมูลระหว่างการบีบอัด ซึ่งจะจัดอยู่ในกลุ่มของ Lossless Data Compression สำหรับงานวิจัยนี้จะนำการเข้ารหัส Unary มาใช้บีบอัดข้อมูลในลักษณะดังกล่าวซึ่งจะอธิบายต่อไปในบทที่ 3 ส่วนของการออกแบบเซ็นเซอร์ตรวจจับสัมผัส

บทที่ 3

การออกแบบระบบที่นำเสนอ

ในบทนี้จะกล่าวถึงการออกแบบระบบควบคุมฟังก์ชัน Pitch bend และ Vibrato บนเครื่องดนตรีคีย์บอร์ดที่นำเสนอซึ่งประกอบด้วย การออกแบบเซ็นเซอร์สัมผัสแบบวัดความจุไฟฟ้าที่ใช้ และการออกแบบอัลกอริทึมสำหรับควบคุมฟังก์ชัน Pitch bend และ Vibrato ที่ใช้ในการประมวลผลสัญญาณดิจิทัลของระบบที่นำเสนอด้วยบอร์ดประมวลผลหลัก

3.1 โครงสร้างและการทำงานของระบบที่นำเสนอ

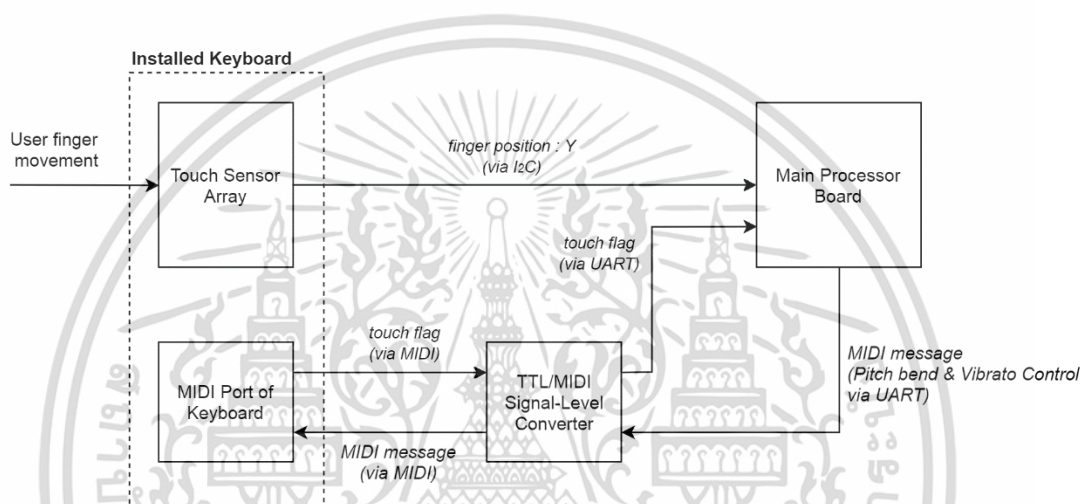


รูปที่ 3.0 Hardware Diagram โดยรวมที่ใช้ในระบบที่นำเสนอ

ระบบที่นำเสนอจะถูกควบคุมฟังก์ชัน Pitch bend และ Vibrato ผ่านการรูดนิ้วบนแผ่นเซ็นเซอร์ตรวจจับสัมผัสที่ถูกติดตั้งบนคีย์บอร์ดสำหรับวัดตำแหน่งการสัมผัสตามแนวตั้ง (Y-Axis) ของคีย์ซึ่งประกอบด้วยเซ็นเซอร์สำหรับคีย์ขาว (White key) จำนวน 14 คีย์ ดังรูปที่ 3.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซ็นเซอร์ของแต่ละคีย์จะส่งข้อมูลที่ระบุตำแหน่งการสัมผัสไปยังบอร์ดประมวลผลหลัก (Main Processor Board) ผ่านบัส I2C เพื่อประมวลผลสัญญาณดิจิทัลโดยใช้อัลกอริทึมที่นำเสนอสำหรับระบุลักษณะการรูดนิ้วที่สอดคล้องกับเงื่อนไขการเรียกใช้ฟังก์ชันที่กำหนดไว้ จากนั้นจะส่งชุดคำสั่ง MIDI ที่สอดคล้องกับเงื่อนไขของอัลกอริทึมเพื่อเปิดการใช้งานฟังก์ชัน Pitch bend และ Vibrato ผ่านการสื่อสารแบบ UART ระดับแรงดัน TTL ไปยังวงจร TTL/MIDI Signal Converter เพื่อแปลงสัญญาณเป็นรูปแบบ Current-Loop ซึ่งสามารถเชื่อมต่อกับพอร์ต MIDI ของคีย์บอร์ดที่ถูกติดตั้งได้ โดยจากโครงสร้างที่กล่าวมาสามารถนำมาเขียนเป็น System Block Diagram ดังรูปที่ 3.1



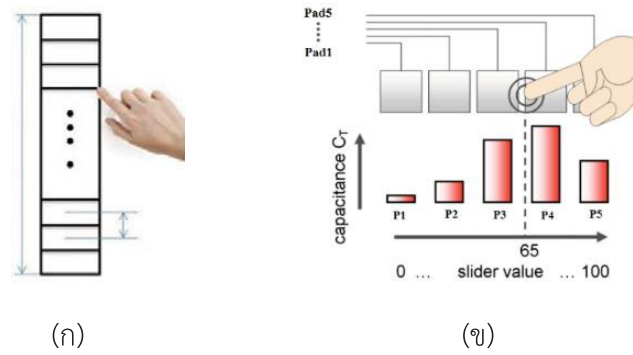
รูปที่ 3.1 System Block Diagram ของระบบที่นำเสนอ

3.2 การออกแบบเซ็นเซอร์สัมผัส

3.2.1 การออกแบบเซ็นเซอร์สัมผัสและวงจรที่ใช้

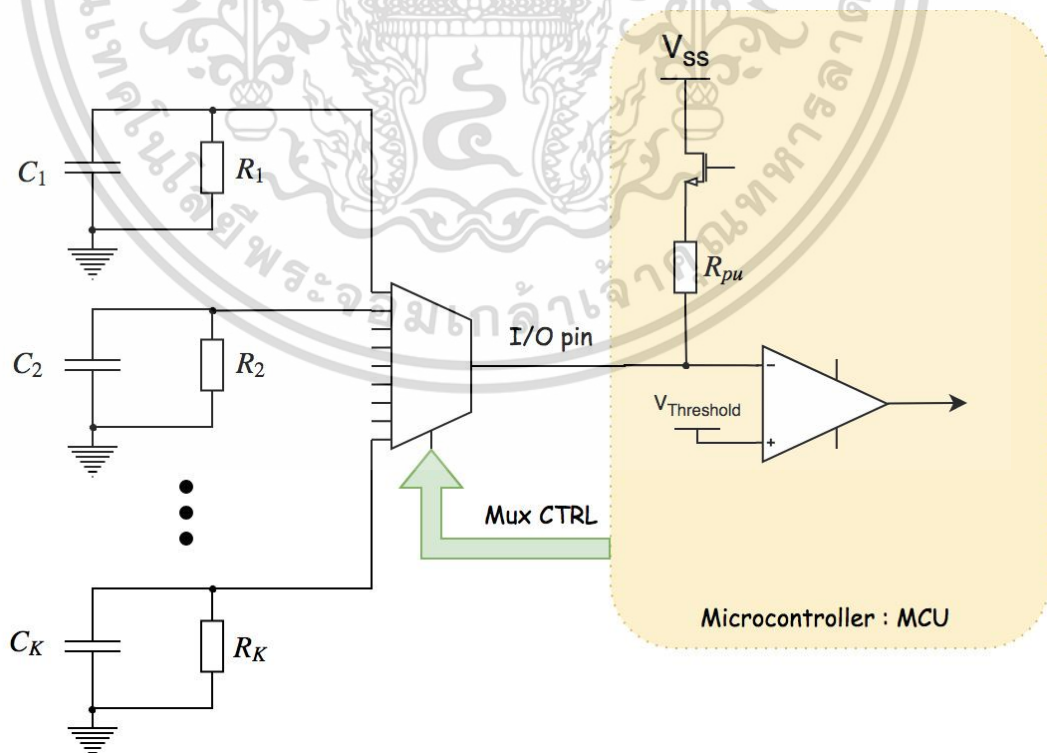
เซ็นเซอร์สัมผัสที่ใช้ถูกออกแบบบนพื้นฐานของเซ็นเซอร์ตรวจจับการสัมผัสด้วยเทคนิคการวัดความจุไฟฟ้าที่ได้กล่าวไว้ในหัวข้อที่ 2.3 สร้างจากแผ่น PCB แบบ 2 ชั้นวัสดุชนิด FR-4 หนา 1.6 มม โดยบริเวณสัมผัสนั้นประกอบด้วยแผ่นขั้วสัมผัสขนาดย่อยที่มีขนาดเท่าๆกันจำนวน $K = 16$ จุดซึ่งเรียงต่อกันแบบแถบสไลเดอร์ (Slider Strip) สำหรับระบุตำแหน่งสัมผัสตามแนวแกนยาวของลิ้นคีย์ ซึ่งแสดงดังไว้รูปที่ 3.2 หลักการระบุตำแหน่งการสัมผัสของแถบสไลเดرنั้นเริ่มจากวัดความจุไฟฟ้าของขั้วสัมผัสขนาดย่อยทุกๆจุดบนแถบสไลเดอร์ ค่าความจุไฟฟ้าของขั้วสัมผัสบริเวณใกล้จุดสัมผัสจะมีค่าสูงกว่าบริเวณอื่นๆ แล้วนำข้อมูลจากการสแกนดังกล่าวมาผ่านกระบวนการสกัดตำแหน่งสัมผัสซึ่งจะอธิบายขั้นตอนต่างๆของกระบวนการนี้ในหัวข้อที่ 3.2.2 โดยใช้หลักการที่แสดงดังรูปที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 (ก) ลักษณะของแถบสไลเดอร์ (ข) หลักการระบุตำแหน่งสัมผัสของแถบสไลเดอร์

เทคนิคของวงจรที่ใช้วัดความจุไฟฟ้าของขั้วสัมผัส คือ RC Charge and Discharge Method โดยใช้ไมโครคอนโทรลเลอร์ (Microcontroller) ควบคุมการทำงานของวงจรที่ใช้ดังรูปที่ 3.3 ซึ่งจะควบคุมการชาร์ตและคายประจุ พร้อมใช้ Timer และ Comparator ภายในไมโครคอนโทรลเลอร์ สำหรับวัดช่วงเวลาการคายประจุ t_d ดังที่อธิบายไว้ในหัวข้อที่ 2.3 ตัวมัลติเพล็กซ์เซอร์ (Multiplexer : MUX) ถูกนำมาใช้เพื่อเพิ่มช่องทางการเชื่อมต่อวงจร RC เข้ากับขา I/O ของไมโครคอนโทรลเลอร์ เมื่อวงจรสุ่มวัดค่า t_d ของทุกๆขั้วสัมผัสแล้ว จะนำมาผ่านกระบวนการสกัดตำแหน่งสัมผัสต่อไป



รูปที่ 3.3 วงจรวัดความจุไฟฟ้าของเซ็นเซอร์สัมผัสที่ใช้ในงานวิจัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

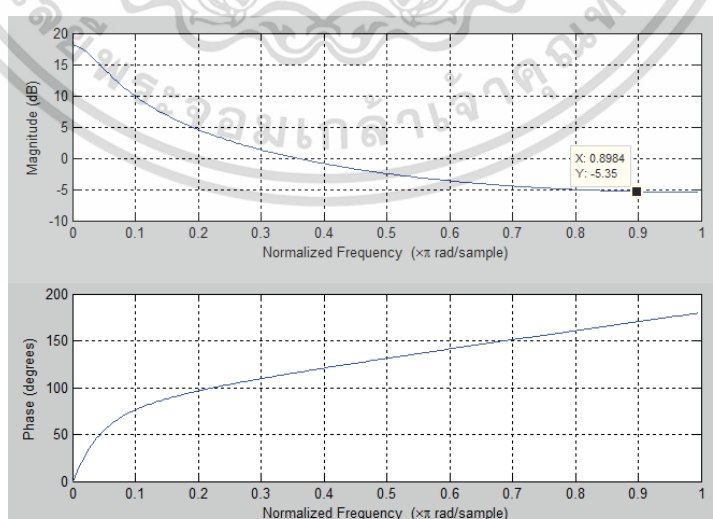
ขนาดของขั้วสัมผัสบนแถบสไลเดอร์คือ 1.2×3 ตร.ซม. ซึ่งจะให้ค่าความจุไฟฟ้าแฝงขณะไม่มีการสัมผัสประมาณ 10 pF ค่าตัวต้านทาน R ในวงจร RC นั้นเท่ากับ $2 \text{ M}\Omega$ ไมโครคอนโทรลเลอร์ที่ใช้สำหรับเซ็นเซอร์สัมผัสนี้คือ 8-Bit MCU ตระกูล AVR เบอร์ ATmega168PA-AU ทำงาน ณ Clock Speed ที่ 20 MHz ส่วน Multiplexer ที่ใช้คือ 8-Channel Analog MUX เบอร์ CD4051

3.2.2 กระบวนการสกัดตำแหน่งการสัมผัส

สัญญาณเอาต์พุตที่ต้องการจากเซ็นเซอร์สัมผัสสำหรับอัลกอริธึมควบคุมฟังก์ชัน Pitch bend และ Vibrato คือ ตำแหน่งสัมผัสบนแถบสไลเดอร์ (Y) และ สัญญาณบ่งชี้การสัมผัส (touch flag) ซึ่งเอาต์พุตทั้งสองนี้จะถูกสกัดโดยใช้ค่า t_d ที่ได้จากหัวข้อที่ 3.2.1 ของทุกๆ ขั้วสัมผัส เริ่มต้นจากนำค่า t_d มาผ่านฟังก์ชันขีดเริ่ม (Threshold function) เพื่อตัดสัญญาณสัมผัสในแต่ละขั้ว แต่เนื่องจากผลของการเหนี่ยวนำสัญญาณ 50 Hz ของร่างกายมนุษย์ขณะที่นิ้วสัมผัส [1] ทำให้ค่า t_d ที่ได้มีสัญญาณรบกวนดังกล่าวแฝงเข้าไปด้วย ตัวกรอง IIR low-pass filter ถูกนำมาใช้ในลักษณะของ Smoothing Filter เพื่อกรองสัญญาณรบกวนออกจากสัญญาณ t_d [1] ก่อนจะส่งไปยังฟังก์ชันขีดเริ่ม โดยแสดงฟังก์ชันถ่ายโอนและสมการ Difference ของตัวกรองดังสมการที่ 3.1 และ 3.2

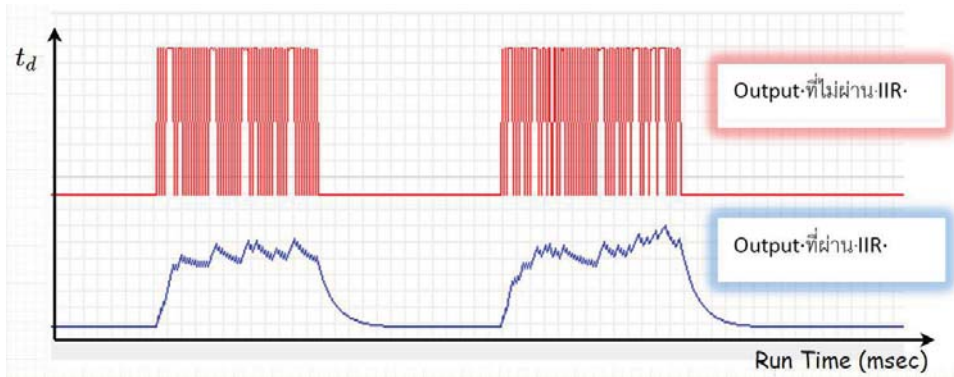
$$H(z) = \frac{1}{1 - 0.875z^{-1}} \quad (3.1)$$

$$y[n] = 0.875y[n-1] + x[n] \quad (3.2)$$



รูปที่ 3.4 การตอบสนองทางความถี่ของ IIR low-pass filter ดังสมการที่ 3.1

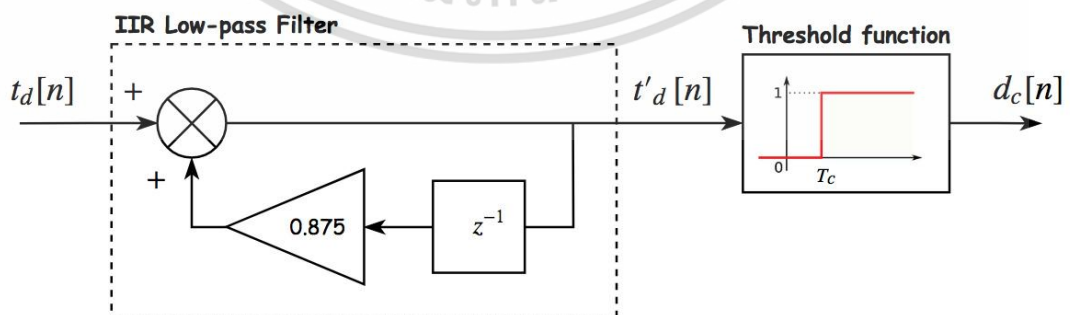
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 เปรียบเทียบเอาต์พุตที่ได้จากการกรองสัญญาณ t_d ด้วย IIR Filter ในสมการที่ 3.2 (น้ำเงิน) กับสัญญาณ t_d ดั้งเดิมที่ไม่ผ่าน IIR Filter (แดง)

หลังจากที่สัญญาณ t_d ผ่าน IIR Filter แล้ว การระบุการสัมผัสด้วยฟังก์ชันขีดเริ่มจะง่ายยิ่งขึ้นเนื่องจากลักษณะการ Oscillation ที่แผ่มาจากผลของสัญญาณรบกวน 50 Hz หายไปแสดงผลดังรูปที่ 3.5 จากนั้นนำเอาต์พุตที่ได้จาก IIR Filter คือ t'_d มาผ่านฟังก์ชันขีดเริ่มแสดงดังรูปที่ 3.6 โดยค่าขีดเริ่มที่ใช้คือ T_c กรณีที่ $t'_d \geq T_c$ จะหมายถึงค่าความจุไฟฟ้าสูงเพียงพอที่จะระบุว่าจะระบุดว่าขณะนี้มีการสัมผัสแล้ว เอาต์พุตที่ได้จากฟังก์ชันขีดเริ่ม d_c จะมีค่าเท่ากับ 1 ส่วนกรณีที่ $t'_d < T_c$ จะถือว่าไม่มีการสัมผัสและให้เอาต์พุตค่า d_c เท่ากับ 0 อธิบายได้ดังสมการที่ 3.3

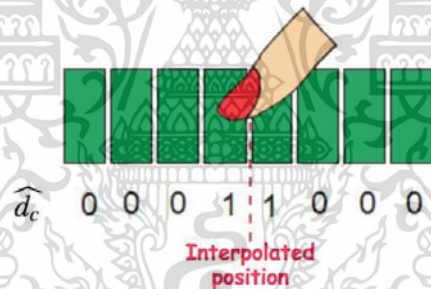
$$d_c = \begin{cases} 0 & , \text{if } t'_d < T_c \\ 1 & , \text{if } t'_d \geq T_c \end{cases} \quad (3.3)$$



รูปที่ 3.6 บล็อกไดอะแกรมสำหรับระบุการสัมผัส (ต่อ 1 ขั้วสัมผัส) ด้วยค่าเวลาการคายประจุ t_d เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการสแกนระบุการสัมผัสของนิ้วครบทุกๆ นิ้วสัมผัสในแถบสไลเดอร์โดยใช้กระบวนการที่กล่าวมาตั้งบล็อกไดอาแกรมในรูปที่ 3.6 แล้ว จะได้ Array ของเอาต์พุต d_c , \widehat{d}_c ซึ่งถูกใช้เป็นสถานะการสัมผัสของแต่ละจุด กระบวนการ Interpolation ถูกใช้เพื่อเพิ่มตำแหน่งสัมผัสระหว่างนิ้วสัมผัสที่อยู่ติดกันดังกรณีการสัมผัสในรูปที่ 3.7 ด้วยการเฉลี่ยตำแหน่งสัมผัสจุดที่ $d_c = 1$ ทั้งหมด ทำให้ความละเอียด (Resolution) ของการระบุตำแหน่งเพิ่มขึ้นเป็น 2 เท่า หากจำนวนนิ้วสัมผัสมีเท่ากับ K ความละเอียดที่ได้จากการ Interpolation คือ $2K$ ระดับ ส่วนค่า *touch flag* ที่ใช้ระบุการสัมผัสบนลิ้มคีย์นั้นคำนวณได้จากการหา norm ของ \widehat{d}_c ดังสมการที่ 3.4 หากไม่มีการสัมผัสเลยสมาชิกใน \widehat{d}_c จะต้องเท่ากับ 0 หมดซึ่งจะให้ค่า $|\widehat{d}_c| = 0$

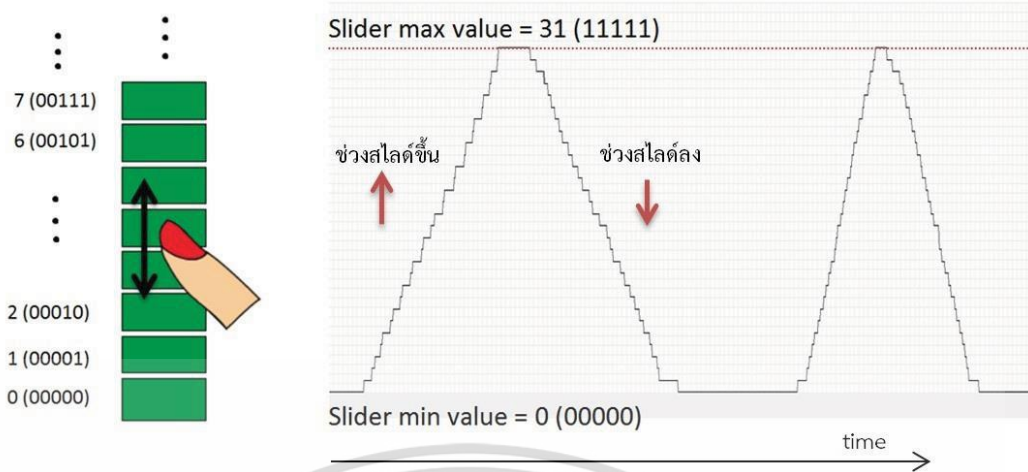
$$\text{touch flag} = \begin{cases} 0 & , \text{if } |\widehat{d}_c| = 0 \\ 1 & , \text{if } |\widehat{d}_c| \neq 0 \end{cases} \quad (3.4)$$



รูปที่ 3.7 การทำ Interpolation เพื่อเพิ่มตำแหน่งสัมผัสระหว่างนิ้วสัมผัสที่อยู่ติดกัน

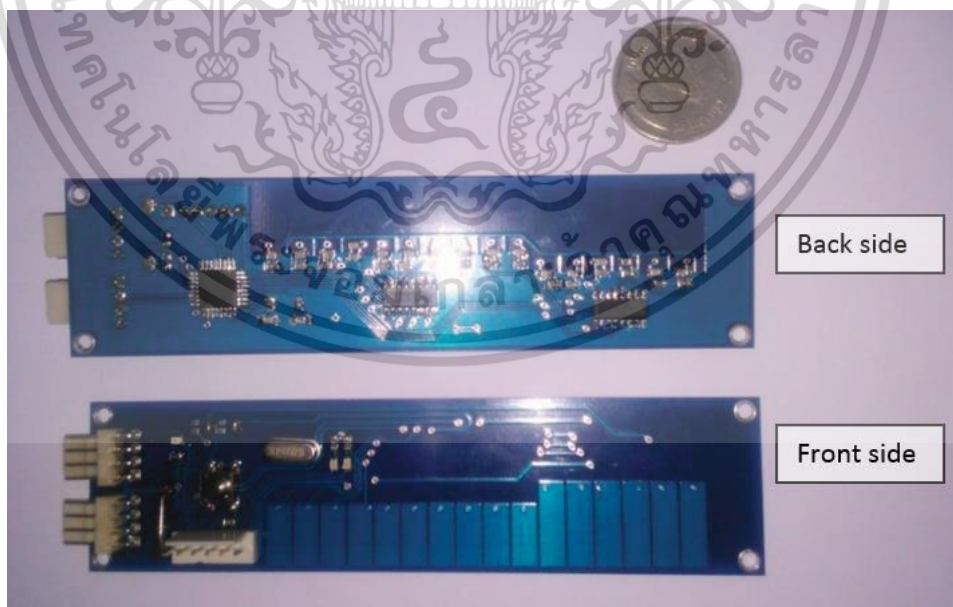
เนื่องจาก \widehat{d}_c มีลักษณะแบบ Thermometer Code [9] ซึ่งมีขนาดข้อมูล เท่ากับ $2K$ บิต (หลังจาก Interpolation แล้ว) จึงควรถูกบีบอัด (Compression) ให้ข้อมูลอยู่รูปของ Binary Code ซึ่งมีขนาดเล็กลงโดยใช้กระบวนการย้อนกลับของการเข้ารหัส Unary ที่ได้อธิบายไว้ในหัวข้อที่ 2.5.3 ก่อนจะนำไปใช้งานต่อไป โดยความละเอียดใหม่ที่ได้นั้นจะเปลี่ยนไปเท่ากับ $\log_2 2K$ บิต ทำให้ประหยัด Bandwidth การสื่อสารสำหรับการส่งตำแหน่งการสัมผัสของแต่ละลิ้มคีย์ไปยังบอร์ดประมวลผลหลักและทำให้ข้อมูลใช้งานง่ายขึ้น ข้อมูลตำแหน่งสัมผัสที่ได้หลังจากการเข้ารหัสสัญญาณ \widehat{d}_c คือสัญญาณ Y ซึ่งมีลักษณะเป็น Slider Function แสดงดังรูปที่ 3.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 ตำแหน่งสัมผัส Y ซึ่งมีลักษณะเป็น Slider Function

เซ็นเซอร์สัมผัสสำหรับการทดสอบระบบที่นำเสนอจากกระบวนการออกแบบที่ได้กล่าวมาทั้งหมดนี้แสดงดังรูปที่ 3.9 โดยเซ็นเซอร์มีความละเอียดของการระบุตำแหน่งเท่ากับ 32 ระดับ และมีอัตราการสุ่ม (Sampling Rate) ค่าตำแหน่งสัมผัสเท่ากับ 120 เฟรมต่อวินาที (fps) ซึ่งได้จากการวัดผลระหว่างการทดสอบในบทที่ 4



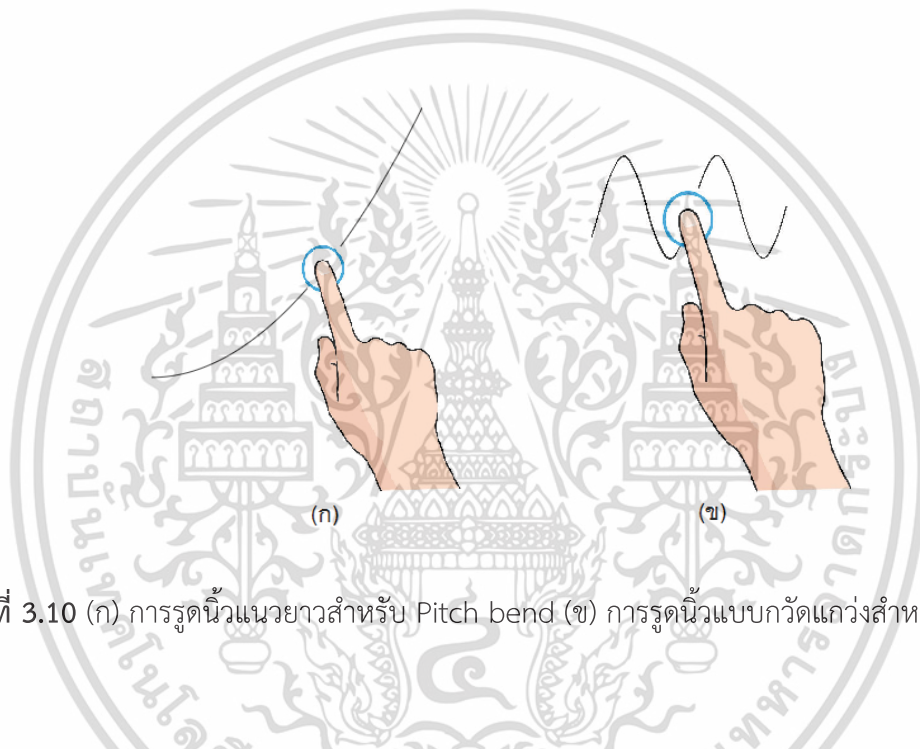
รูปที่ 3.9 เซ็นเซอร์สัมผัสของระบบที่นำเสนอจากการออกแบบในหัวข้อที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

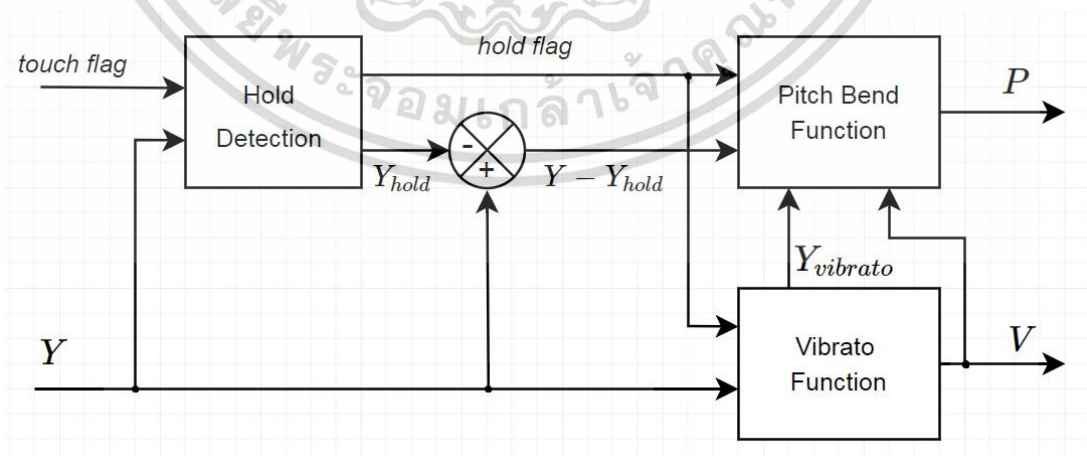
3.3 อัลกอริธึม DSP สำหรับบอร์ดประมวลผลหลัก

3.3.1 อัลกอริธึม DSP สำหรับควบคุมฟังก์ชัน Pitch bend และ Vibrato

หลังจากที่บอร์ดประมวลผลหลักทำการสุ่มวัดค่าตำแหน่งการสัมผัสจาก Touch sensor ของแต่ละตำแหน่งคีย์โน้ตแล้ว จะนำข้อมูลที่ได้มาผ่านการประมวลผลสัญญาณดิจิทัลเพื่อควบคุมการทำงานของฟังก์ชัน Pitch bend และ Vibrato ให้สอดคล้องกับลักษณะการรูดนิ้วดังรูปที่ 3.10(ก) สำหรับ Pitch bend และรูปที่ 3.10(ข) สำหรับ Vibrato โดยที่การควบคุมต่อ 1 คีย์โน้ตนั้นจะใช้ อัลกอริธึมดังบล็อกไดอะแกรมในรูปที่ 3.11 ซึ่งจะมีรายละเอียดดังนี้



รูปที่ 3.10 (ก) การรูดนิ้วแนวยาวสำหรับ Pitch bend (ข) การรูดนิ้วแบบกวัดแกว่งสำหรับ Vibrato

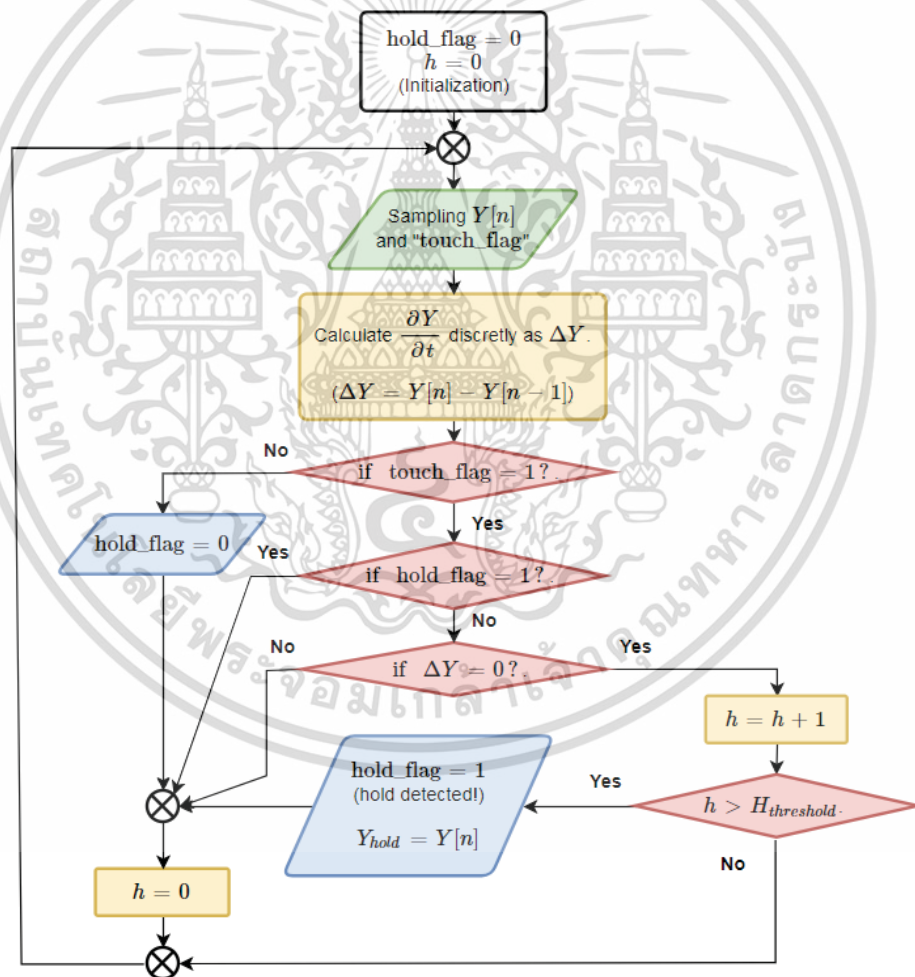


รูปที่ 3.11 Full Algorithm : Pitch bend & Vibrato Control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

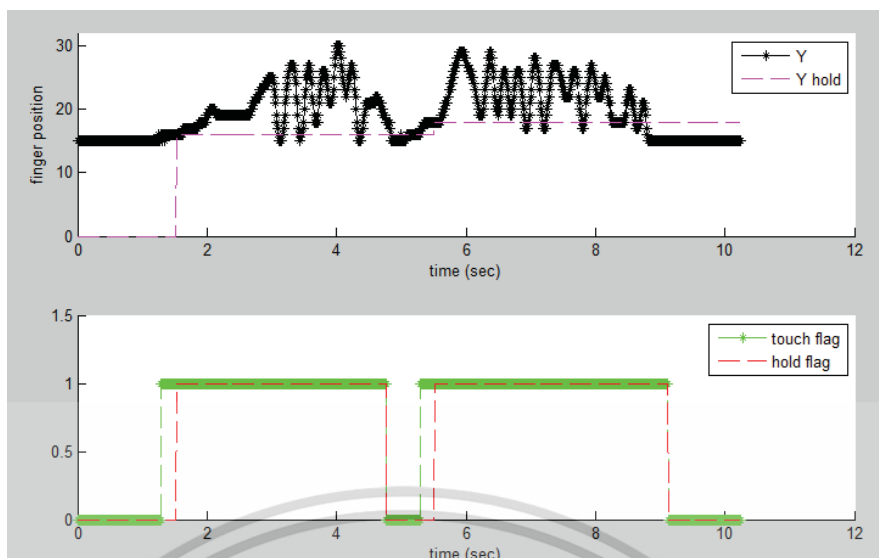
1) Hold detection

ทำหน้าที่ตรวจจับการสัมผัสค้ำและสกัดค่าตำแหน่งสัมผัสค้ำ Y_{hold} เพื่อเปิดการใช้งานฟังก์ชัน Pitch bend และฟังก์ชัน Vibrato หลังจากที่มีการสัมผัสค้ำ อัลกอริทึมส่วนนี้เริ่มต้นจากการรับสัญญาณระบุงการสัมผัส $touch\ flag$ และตำแหน่งสัมผัส Y จากเซนเซอร์ หากมีการสัมผัส ($touch\ flag = 1$) และมีการสัมผัสค้ำ ($\frac{dY}{dt} = 0$) นานเกินระยะเวลาที่กำหนดแล้ว จะส่งสัญญาณ $hold\ flag = 1$ เพื่อแสดงความพร้อมที่จะใช้งานฟังก์ชัน Pitch bend และทำการบันทึกค่า Y ณ เวลานั้นเพื่อใช้เป็นตำแหน่งสัมผัสค้ำ Y_{hold} โดยระบบจะมีการอัปเดตค่า Y_{hold} ทุกๆครั้งที่มีการสัมผัสค้ำครั้งใหม่ อัลกอริทึมดังกล่าวแสดงไว้ดังรูปที่ 3.12



รูปที่ 3.12 อัลกอริทึมสำหรับระบุงการแตะค้ำและสกัดตำแหน่งแตะค้ำ Y_{hold}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



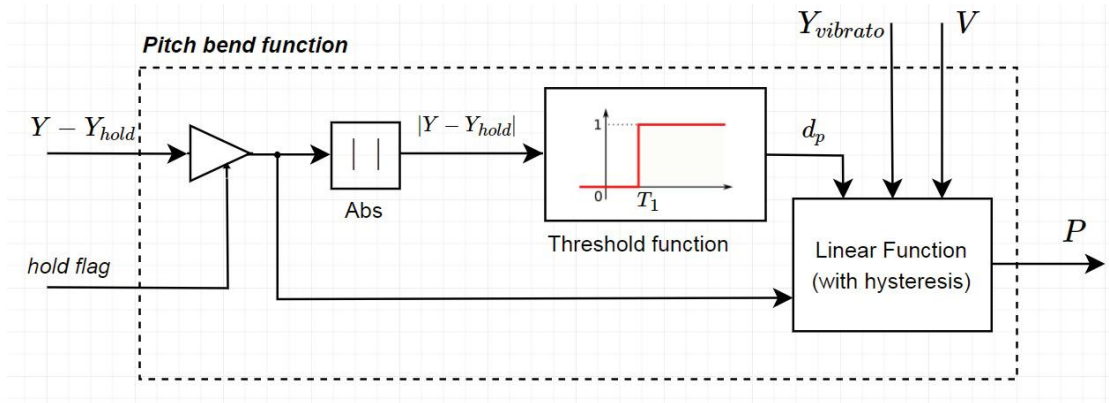
รูปที่ 3.13 ตัวอย่างสัญญาณ *hold flag* และ Y_{hold} ที่ได้จากการทดสอบ Hold detection

2) Pitch bend function

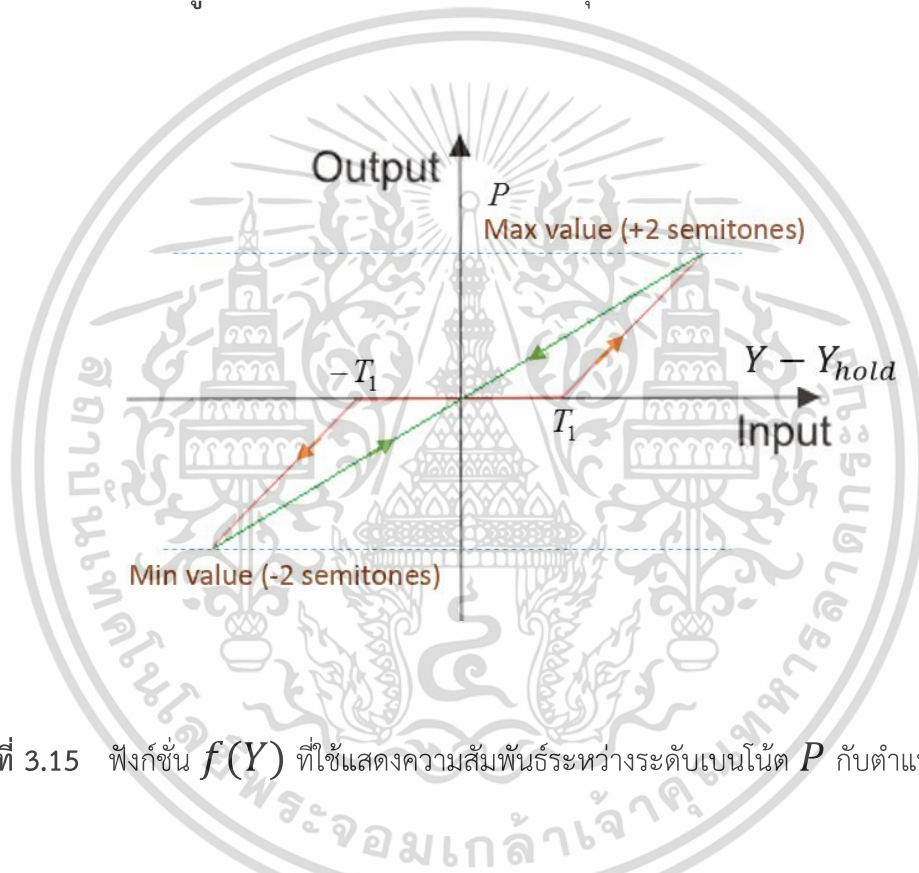
ทำหน้าที่กำหนดระดับการเบนของโน้ตที่ถูกเล่นด้วยฟังก์ชัน Pitch bend หลักการเรียกใช้งานฟังก์ชันนี้คือ เมื่อมีการตรวจจับว่ามีการสัมผัสค้ำแล้ว (*hold flag* = 1) สัญญาณ Y และ Y_{hold} จะถูกป้อนไปยังฟังก์ชันเพื่อใช้ระบุลักษณะการรูดนิ้วตามลักษณะดังรูปที่ 3.10(ก) หากมีการรูดขึ้นหรือลงต่างจากระดับสัมผัสค้ำเกินระยะที่กำหนดแล้ว ฟังก์ชัน Pitch bend จะถูกเรียกใช้งานโดยใช้อัลกอริทึมตัดสินใจการทำงานดังรูปที่ 3.14

อัลกอริทึมของฟังก์ชัน Pitch bend เริ่มต้นจากการวัดขนาดระยะห่างของตำแหน่งนิ้วปัจจุบันเทียบกับตำแหน่งสัมผัสค้ำ ได้ค่า $\bar{Y} = |Y - Y_{hold}|$ ออกมาและทำการตัดสินใจว่า \bar{Y} มีขนาดเกินกว่าค่าขีดเริ่ม T_1 หรือเปล่าด้วยฟังก์ชันขีดเริ่ม (Threshold Function) หาก $\bar{Y} > T_1$ แล้ว จะส่งสัญญาณ $d_p = 1$ เพื่อเปิดการทำงาน Linear function : $f(Y)$ ที่ใช้สำหรับกำหนดขนาด Pitch bend : P ตามความสัมพันธ์ดังกราฟในรูปที่ 3.15 ซึ่งมีคุณสมบัติ Hysteresis อยู่เพื่อกำจัดความไม่เป็นเชิงเส้น (non-linear) จากผลของการใช้ฟังก์ชัน Threshold ทำให้มีความสัมพันธ์ระหว่าง P และ Y มีความเป็นเชิงเส้นขณะที่การรูดเปลี่ยนทิศกลับมายังจุดเริ่มต้น ช่วงระดับการเบนโน้ตสูงสุดและต่ำสุดนั้นจะอยู่ที่ ± 2 semitone

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 อัลกอริธึมสำหรับควบคุมฟังก์ชัน Pitch bend



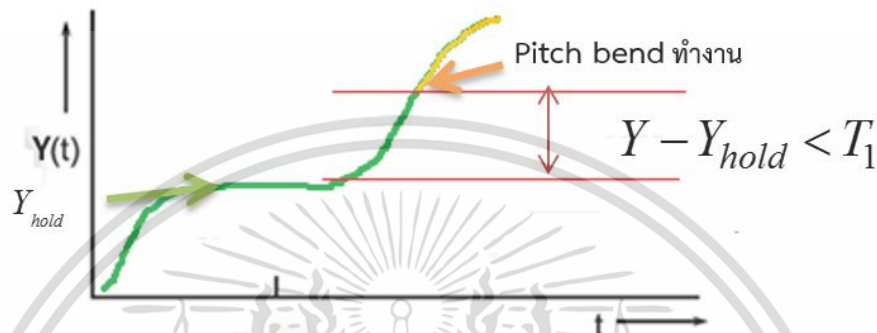
รูปที่ 3.15 ฟังก์ชัน $f(Y)$ ที่ใช้แสดงความสัมพันธ์ระหว่างระดับเบนนोट P กับตำแหน่งสัมผัส Y

ระบบจะอนุญาตให้ฟังก์ชัน Vibrato ถูกเรียกใช้งานแทรกเข้ามาขณะที่ฟังก์ชัน Pitch bend ถูกใช้งานอยู่ได้โดยผ่านสัญญาณ V เมื่อ $V = 1$ แต่ระดับ Pitch bend : P จะหยุดการอัปเดตจนกว่าจะออกจากฟังก์ชัน Vibrato ซึ่งค่า P ขณะนั้นจะเท่ากับ $g(Y_{vibrato})$ เพื่อใช้กำเนิดระดับ Pitch ศูนย์กลาง โดยที่ $Y_{vibrato}$ นั้นมาจากอัลกอริธึมของฟังก์ชัน Vibrato ซึ่งถูกใช้เป็นตัวดำเนินการสัมผัสเฉลี่ยขณะที่เกิด oscillatory motion เพื่อสั่งใช้ฟังก์ชัน Vibrato จากทั้งหมดที่กล่าวมา สามารถอธิบายค่า Pitch bend ได้ด้วยสมการที่ 3.5 และ 3.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$P[n] = \begin{cases} 0 & \text{if } d_p[n] = 0 \\ (1 - V[n]) \cdot f(Y) + V[n] \cdot g(Y_{vibrato}) & \text{if } d_p[n] = 1 \end{cases} \quad (3.5)$$

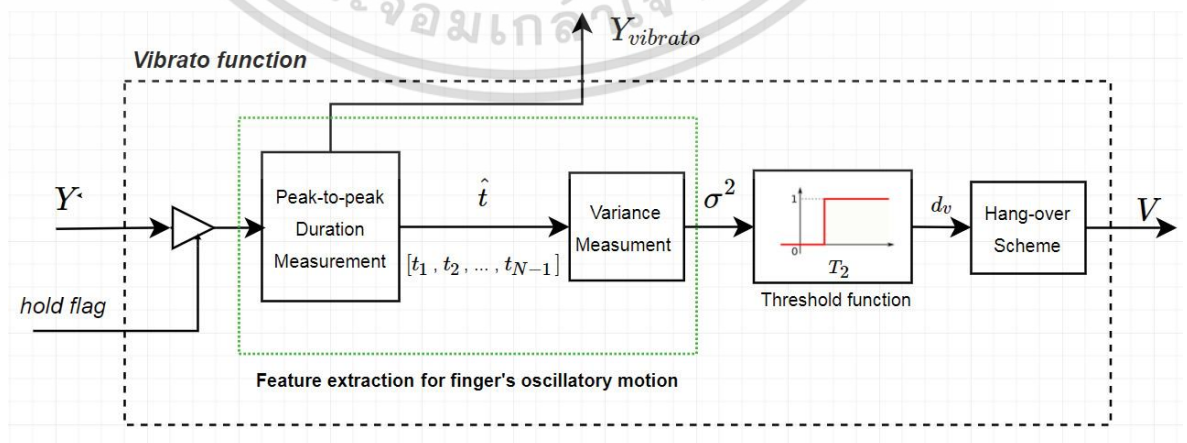
$$g(Y_{vibrato}) = (2 \text{ semitones} / Y_{\max}) \cdot (Y_{vibrato} - Y_{hold}) \quad (3.6)$$



รูปที่ 3.16 กราฟตำแหน่งของนิ้วที่อธิบายการทำงานของฟังก์ชันซิดเริ่มสำหรับฟังก์ชัน Pitch bend

3) Vibrato function

ทำหน้าที่ควบคุมการเปิด/ปิดของฟังก์ชัน Vibrato ของโน้ตที่ถูกเล่น หลักการเรียกใช้งานฟังก์ชันนี้คือ เมื่อมีการรูดนิ้วในลักษณะของการกวัดแกว่ง (Oscillatory Motion) ตามลักษณะดังรูปที่ 3.10(ข) เหมือนลักษณะของการใช้งาน Vibrato บนเครื่องดนตรีประเภทเครื่องสาย โดยใช้ อัลกอริธึมดังรูปที่ 3.17 สำหรับตรวจจับการรูดลักษณะดังกล่าวและเรียกใช้งานฟังก์ชัน Vibrato



รูปที่ 3.17 อัลกอริธึมสำหรับควบคุมฟังก์ชัน Vibrato function

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

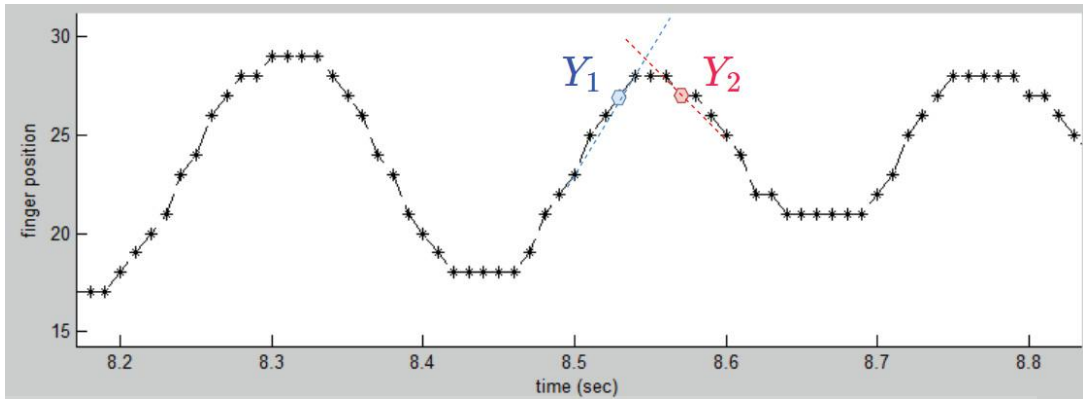
อัลกอริทึมของฟังก์ชัน Vibrato เริ่มต้นจากรับค่าตำแหน่งการสัมผัส Y เพื่อใช้สกัด Feature สำหรับใช้ตัดสินว่ามีการรูดนิ้วแบบกวัดแกว่งหรือไม่ กระบวนการสกัด Feature (Feature extraction) เริ่มจากตรวจจับจุดยอด (Peak value) ของสัญญาณ Y โดยใช้ค่า $\frac{dY}{dt}$ เพื่อตรวจจุดยอดที่ $\frac{dY}{dt} = 0$ และ ค่า $\frac{dY}{dt}$ ของตำแหน่งสัมผัสสุ่มรอบข้างจุดยอด Y_1, Y_2 (แสดงดังรูปที่ 3.18) ต้องมีเครื่องหมายเปลี่ยนแปลง ($sign\left(\frac{dY_1}{dt}\right) \cdot sign\left(\frac{dY_2}{dt}\right) = -1$) เมื่อเจอจุดยอดแล้ว ทำการวัดช่วงเวลาระหว่างจุดยอดกับจุดยอดถัดไป (peak-to-peak duration) เป็นค่า t_i แสดงดังกราฟตำแหน่งนิ้วตัวอย่างในรูปที่ 3.19 หากช่วงเวลาดังกล่าวสั้นกว่า $t_{timeout}$ ($t_i < t_{timeout}$) และผลต่างระหว่างจุดยอดใหม่กับจุดยอดเก่าเกินค่าขีดเริ่ม T_3 ($|Y_{p(i)} - Y_{p(i-1)}| > T_3$) แล้ว จะทำการบันทึกตำแหน่งจุดยอด $Y_{p(i)}$ และวัดช่วงเวลาครั้งต่อไป t_{i+1} จนกว่าจะครบช่วงเวลาทั้งหมด $N - 1$ ค่า ซึ่งเซตของตำแหน่งจุดยอดและช่วงเวลาดังกล่าวจะเรียกเป็นเวกเตอร์ \hat{Y}_p ($\hat{Y}_p = [Y_{p(1)} Y_{p(2)} \dots Y_{p(N)}]$) และ \hat{t} ($\hat{t} = [t_1 t_2 \dots t_{N-1}]$) ตามลำดับ โดยที่ N คือจำนวนจุดยอดที่ติดกันทั้งหมดที่ใช้ หากไม่ ($t_i \geq t_{timeout}$ หรือ $|Y_{p(i)} - Y_{p(i-1)}| \leq T_3$) จะไม่ถือว่าเป็นการรูดนิ้วแบบกวัดแกว่งและให้เริ่มต้นการวัดใหม่

เมื่อเวกเตอร์ \hat{t} พร้อมแล้ว จะถูกนำมาหาความกระจายตัวของสมาชิกในเวกเตอร์ \hat{t} โดยคำนวณความแปรปรวน (Variance) : σ^2 ด้วยสมการที่ 3.7 เนื่องจากว่า หากผู้ใช้งานทำการรูดนิ้วแบบกวัดแกว่งที่มีคาบเวลาแน่นอนซึ่งเหมาะกับการเรียกใช้งาน Vibrato แล้ว ช่วงเวลาที่วัดได้ t_1, t_2, \dots, t_{N-1} จะมีค่าที่ใกล้เคียงกัน ความแปรปรวนที่ได้จะมีค่าน้อย แต่หากคาบในการกวัดแกว่งไม่คงที่อันเนื่องมาจากความไม่ตั้งใจของผู้ใช้งาน ความแปรปรวนที่ได้จะมีค่าสูงชันอย่างมาก ค่า σ^2 ที่ได้จะเป็น Feature ผลลัพธ์ซึ่งจะถูกนำผ่านฟังก์ชันตัดสินใจด้วยค่าขีดเริ่ม T_2 เพื่อใช้ตัดสินการทำงานของฟังก์ชัน Vibrato ได้สัญญาณ d_v ดังสมการที่ 3.8

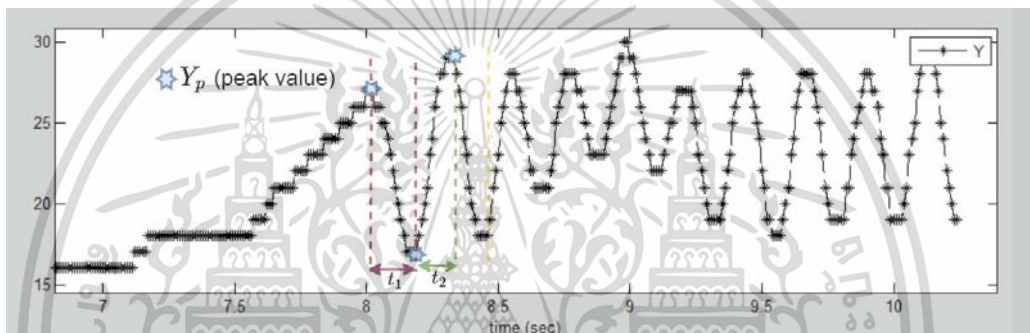
$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (t_i - \bar{t})^2 \quad (3.7)$$

$$d_v[n] = \begin{cases} 0 & , \text{if } \sigma^2[n] < T_2 \\ 1 & , \text{if } \sigma^2[n] \geq T_2 \end{cases} \quad (3.8)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.18 ตำแหน่งสัมผัส Y_1, Y_2 ที่สอดคล้องต่อเงื่อนไขสำหรับเป็นตำแหน่งรอบๆจุดยอด

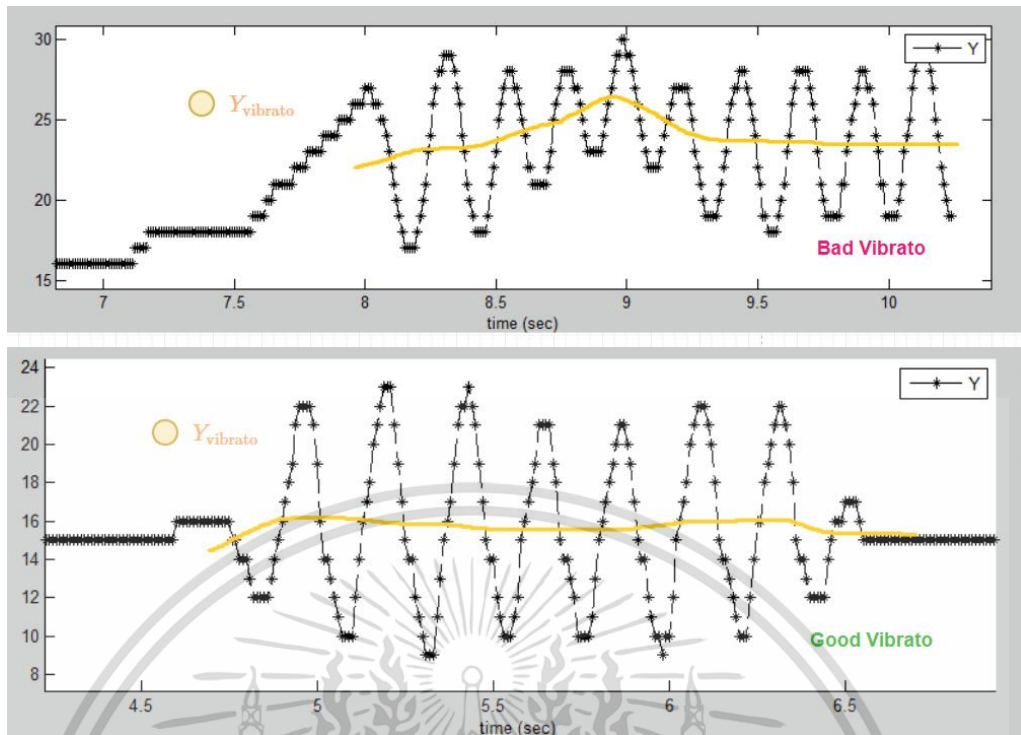


รูปที่ 3.19 กราฟตำแหน่งนิ้วตัวอย่าง สำหรับอธิบายกระบวนการวัด Peak-to-peak Duration

ตำแหน่งสัมผัส $Y_{vibrato}$ ซึ่งถูกใช้เป็นตำแหน่งการสัมผัสเฉลี่ยขณะที่เกิด Oscillatory Motion เพื่อส่งใช้ฟังก์ชัน Vibrato นั้นจะถูกคำนวณหลังจากที่เวกเตอร์ \hat{Y}_p ถูกเตรียมพร้อมแล้ว โดยจะนำค่าตำแหน่งจุดยอดต่างๆที่เก็บไว้ใน \hat{Y}_p มาหาค่าเฉลี่ยได้เป็น \bar{Y}_p และนำผลที่ได้มาผ่าน Smoothing Filter เพื่อลดระดับการเปลี่ยนแปลงของระดับ Pitch ศูนย์กลางให้น้อยลงเมื่อการรูดแบบ Oscillatory Motion มีตำแหน่งศูนย์กลางที่ไม่คงที่ แสดงตัวอย่างดังรูปที่ 3.20 จนเอาต์พุตสัญญาณ $Y_{vibrato}$ อธิบายกระบวนการที่กล่าวมาด้วยสมการที่ 3.9

$$Y_{vibrato}[n] = (1 - \alpha) \cdot Y_{vibrato}[n - 1] + \alpha \cdot \bar{Y}_p \quad (3.9)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



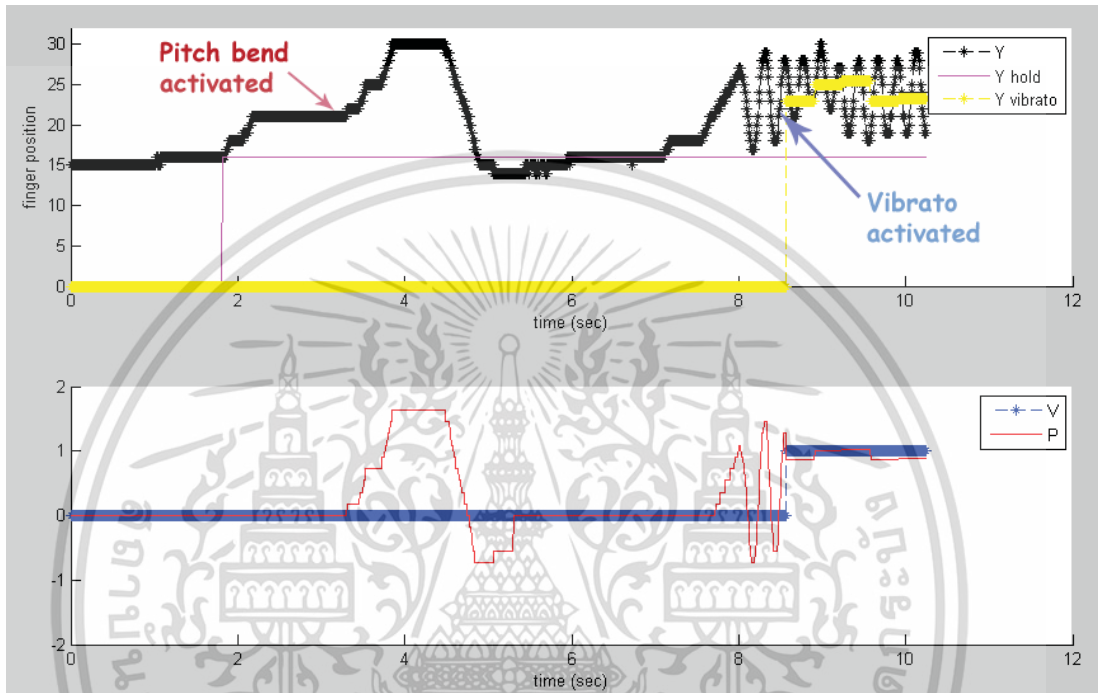
รูปที่ 3.20 ตัวอย่างการรูดนิ้วแบบกวัดแกว่งกรณีตำแหน่งศูนย์กลางไม่คงที่ (บน) และคงที่ (ล่าง)

ก่อนที่สัญญาณ d_v จะถูกส่งออกไปเป็นสัญญาณ V จะต้องผ่าน Hang-over Scheme ก่อนเพื่อให้สัญญาณ V เกิดการหน่วงค้างชั่วขณะหนึ่งเพื่อมิให้ฟังก์ชัน Vibrato ปิดเร็วจนเกินไป โดยใช้ค่า H สำหรับการนับถอยหลัง (Count-down) เพื่อการหน่วงเอาต์พุต หลังจากนั้น สัญญาณ V จะถูกเอาต์พุตออกมาโดยใช้ความสัมพันธ์ดังสมการที่ 3.10 และ 3.11 โดยที่ค่า $V = 1$ นั้น หมายถึงระบบอนุญาตให้ใช้งานฟังก์ชัน Vibrato จากกระบวนการทั้งหมดที่กล่าวมา แสดงตัวอย่างการทำงานของอัลกอริธึมควบคุมฟังก์ชัน Pitch bend และ Vibrato ดังรูปที่ 3.21

$$H[n] = \begin{cases} H_g & , \text{if } d_v[n] = 0 \\ H[n-1] - 1 & , \text{if } d_v[n] = 1 \end{cases} \quad (3.10)$$

$$V[n] = \begin{cases} 0 & , \text{if } H[n] = 0 \\ 1 & , \text{if } H[n] > 0 \end{cases} \quad (3.11)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

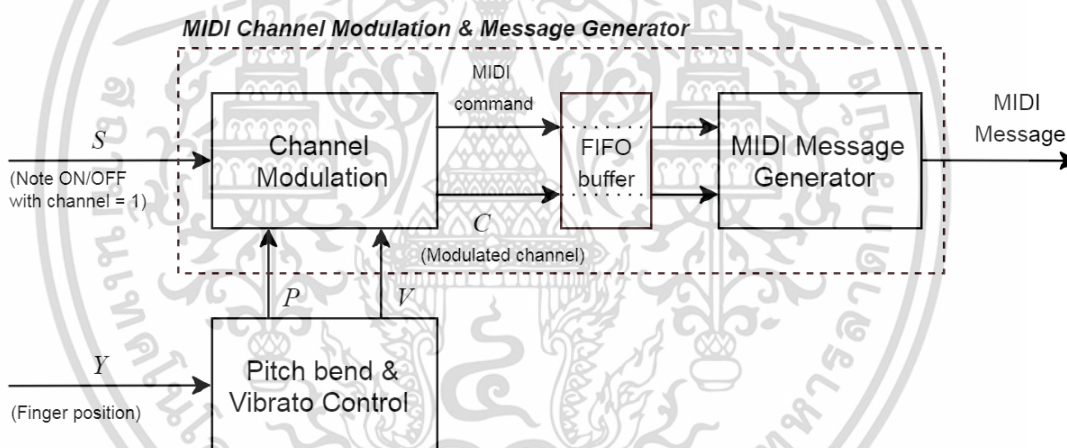


รูปที่ 3.21 ตัวอย่าง Waveform ของสัญญาณ $Y_{vibrato}$, P และ V เพื่อแสดงการทำงานของอัลกอริทึมของฟังก์ชัน Pitch bend และ Vibrato ที่ได้กล่าวมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 อัลกอริธึม DSP ของ Channel Modulation และ Message Generator

เนื่องจากว่าในระบบคีย์บอร์ดแบบดั้งเดิม ชุดคำสั่ง MIDI สำหรับฟังก์ชัน Pitch bend และ Vibrato จะส่งผลกับโน้ตทุกๆตำแหน่งต่อ MIDI Channel ซึ่งหมายความว่า หากคีย์บอร์ดถูกตั้งให้ใช้งานเพียง MIDI Channel เดียวแล้ว (ซึ่งโดยปกติจะเป็นสถานะเริ่มต้น) นักดนตรีจะไม่สามารถบังคับเล่นฟังก์ชัน Pitch bend และ Vibrato กับโน้ตเฉพาะตำแหน่งที่ต้องการ กรณีตัวอย่างเช่น หากนักดนตรีต้องการที่จะเล่น Pitch bend หรือ Vibrato กับโน้ตโซโล่ (Solo) ซึ่งในขณะนั้นจะต้องเล่นโน้ตคอร์ด (Chord) พร้อมไปด้วย ฟังก์ชันอาจเปลี่ยนคุณสมบัติของเสียงโน้ตในคอร์ดไปด้วยซึ่งนักดนตรีไม่ต้องการ เทคนิคที่จะนำเสนอสำหรับแก้ไขปัญหานี้คือ “เปลี่ยน MIDI Channel ทุกครั้งกับโน้ตที่ถูกเล่นเข้ามาใหม่ ไม่ให้ซ้ำกับโน้ตที่ถูกเล่นก่อนหน้านั้น” ซึ่งจะเรียกเทคนิคนี้ว่า MIDI Channel Modulation โดยใช้อัลกอริธึมดังบล็อกไดอะแกรมรูปที่ 3.22 จะถูกใช้ในหลายๆคีย์โน้ต ซึ่งรายละเอียดของอัลกอริธึมนี้มีดังต่อไปนี้



รูปที่ 3.22 อัลกอริธึม DSP ของ MIDI Channel Modulation และ MIDI Message Generator (ถูกใช้สำหรับต่อ 1 คีย์โน้ต)

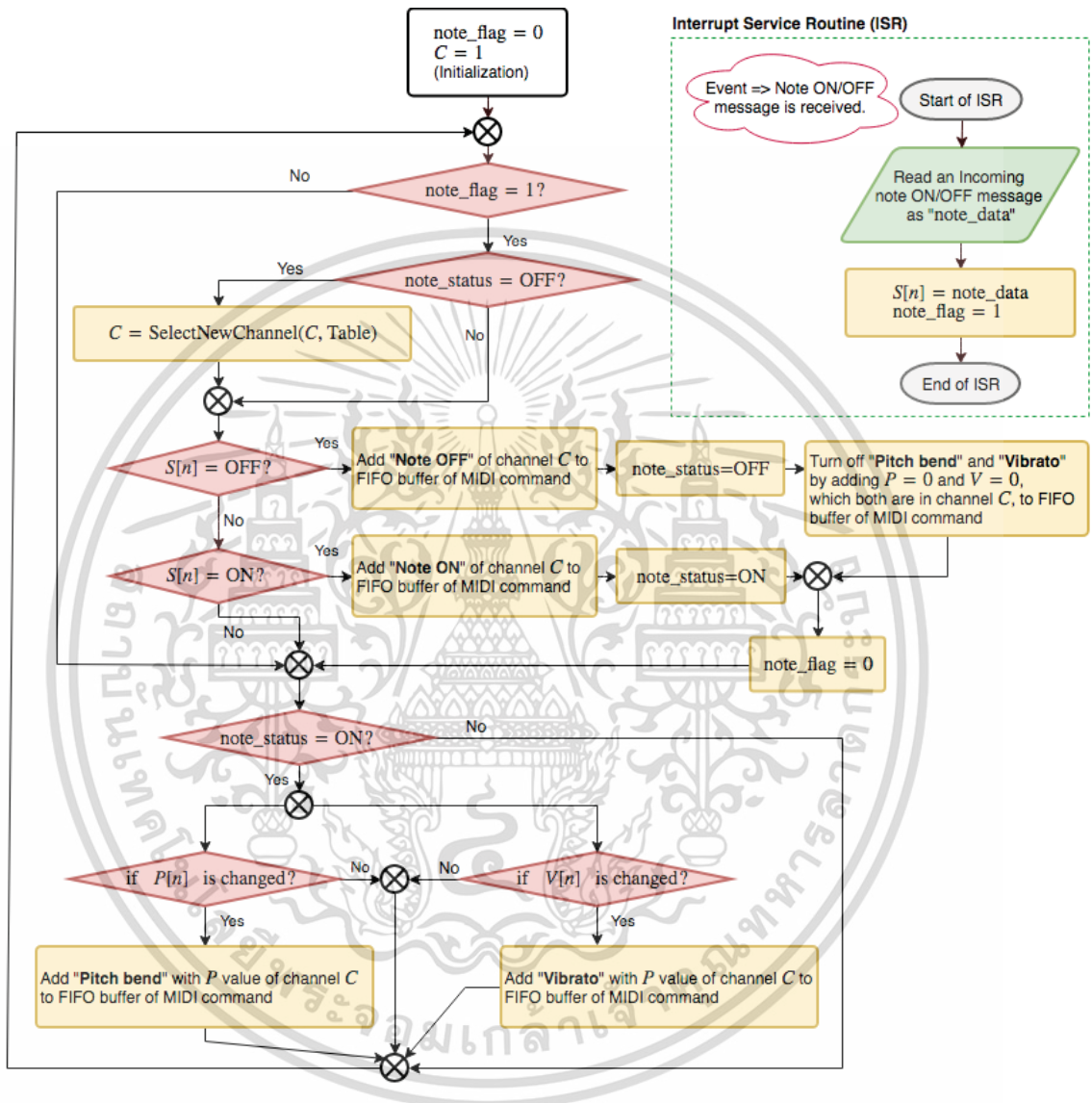
1) Channel Modulation

ทำหน้าที่เปลี่ยน MIDI Channel ของคำสั่ง MIDI ต่างๆของคีย์โน้ตตัวเองที่รับเข้ามาใหม่ให้ไม่ซ้ำกับ MIDI Channel ของคีย์โน้ตอื่นๆที่ถูกใช้งานอยู่ โดยคำสั่ง MIDI ที่รับเข้ามานั้นได้แก่

- คำสั่ง Note ON/OFF (สัญลักษณ์ S) ที่รับมาจากสัญญาณ MIDI ของคีย์บอร์ดที่ถูกระบบติดตั้ง
- คำสั่ง Pitch bend, P และ Vibrato, V ที่ได้รับมาจากอัลกอริธึมควบคุมฟังก์ชัน Pitch bend

และ Vibrato ในหัวข้อที่ 3.3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.23 Flow Chart ของอัลกอริทึมของ MIDI Channel Modulation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระบวนการของ Channel Modulation นั้นแสดงดัง Flow chart ในรูปที่ 3.20 เริ่มต้นจากกำหนดค่า MIDI Channel เริ่มต้นของระบบก่อนซึ่งในตัวอย่างนี้จะให้เป็น Channel 1 ด้วยตั้งค่า $C = 1$ โดยที่ C คือ MIDI Channel ของโน้ต หลังจากนั้นจะดักรอคำสั่งที่แสดงถึงการกดหรือปล่อยคีย์โน้ต Note ON/OFF ผ่าน Interrupt Service Routine (ISR) โดยจะบันทึกคำสั่งที่เข้ามาเก็บไว้เป็นสัญญาณ S และแสดงสถานะ (Flag) ว่ามีคำสั่งเข้ามาใหม่โดยใช้ *note flag* เมื่อมีคำสั่ง Note ON/OFF เข้ามาแล้ว อัลกอริทึมจะตรวจสอบว่าโน้ตนั้นถูกเล่นมาก่อนหรือเปล่าผ่าน หากไม่ระบบจะค้นหา Channel ใหม่ที่ว่างอยู่โดยเรียกใช้งานฟังก์ชัน “SelectNewChannel” โดยใช้ อัลกอริทึมการจอง (Booking Algorithm) ด้วยการใช้ตารางการจองดังตารางที่ 3.1 ซึ่งในตารางนั้นจะระบุสถานะของแต่ละ Channel ว่าขณะนี้ว่าง (Free) หรือไม่ว่าง (Reserved) และระบุตำแหน่งโน้ตของ Channel ที่ถูกใช้แล้ว อัลกอริทึมจะค้นหา Channel ว่างเริ่มจากบน-ล่าง ซึ่งจะหมายถึง Channel 1 เป็นช่องทางที่ควรเลือกก่อนที่สุด (1st Priority Channel) โดยอธิบายอัลกอริทึมการจองของฟังก์ชันนี้ด้วย Pseudo Code ดังรูปที่ 3.24 โดย Channel ใหม่ที่ได้จะไม่ซ้ำกับ Channel ของโน้ตตำแหน่งอื่นๆที่ถูกเล่นไปแล้ว

หลังจากที่ค้นหา MID Channel สำหรับโน้ตที่เข้ามาใหม่แล้ว จะทำการตัดสินใจว่าจะต้องเพิ่มคำสั่ง MIDI อะไรบ้าง (Note ON/OFF, Pitch bend และ Vibrato) ให้สอดคล้องกับสัญญาณ S , P และ V ที่อินพุตเข้ามา โดยจะเรียกชุดคำสั่งเอาต์พุตนี้ว่า MIDI Command สำหรับ MIDI Command ของคำสั่ง Pitch bend และ Vibrato นั้นจะถูกเอาต์พุตออกไปเฉพาะตอนที่โน้ตถูกเล่นแล้ว และค่า P หรือ V เปลี่ยนแปลงไปจากค่าอดีตเท่านั้น เอาต์พุตที่ได้จากกระบวนการทั้งหมดนี้คือ MIDI Channel, C และ MIDI Command

MIDI Channel	Status	Reserved Note
1	Reserved	D3
2	Free	-
3	Reserved	C4
⋮	⋮	⋮
⋮	⋮	⋮
⋮	⋮	⋮
14	Free	-
15	Free	-
16	Free	-

ตารางที่ 3.1 ตารางการจอง MIDI Channel เพื่อค้นหา Channel ใหม่ที่ว่างสำหรับโน้ตที่เข้ามาใหม่ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

function channel_out = SelectNewChannel(channel_in, bookingTable)

    channel_out = 0;
    bookingTable.status(channel_in) = free;

    % Search for a free MIDI channel
    for i = 1:16
        if bookingTable.status(i) == free
            channel_out = i;
            bookingTable.status(channel_out) = reserved;
            break;
        end
    end

    % If there are no free channel -> share with 16th channel
    if channel_out == 0
        channel_out = 16;
        bookingTable.status(channel_out) = reserved;
    end

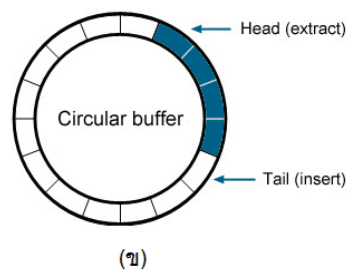
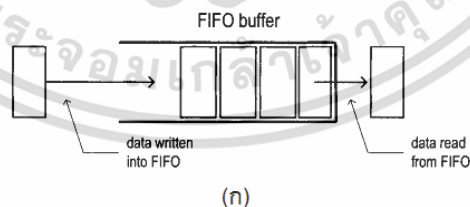
end

```

รูปที่ 3.24 Pseudo Code ฟังก์ชัน SelectNewChannel เพื่อค้นหา MIDI Channel ใหม่ที่ว่างอยู่

2) FIFO Buffer

เอาต์พุต C และ MIDI Command ที่ได้จากระบบการของ Channel Modulation จะถูกนำมาเข้าคิว (Queue) ผ่าน First-In-First-Out Buffer (FIFO Buffer) ในทฤษฎีโครงสร้างข้อมูล (Data Structure) สำหรับพักข้อมูลก่อนที่จะถูกเอาต์พุตไปยังส่วนถัดไปดังรูปที่ 3.25ก โดยชนิดของ FIFO Buffer ที่ใช้คือ Circular Buffer ดังรูปที่ 3.25ข

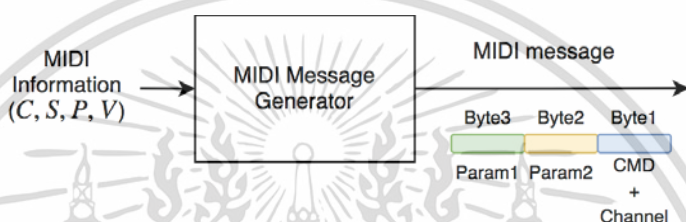


รูปที่ 3.25 (ก) การทำงานของ FIFO Buffer, (ข) โครงสร้างของ Circular Buffer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น เมื่ออนุญาตให้นำไปเผยแพร่โดยไม่เสียค่าใช้จ่าย
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) MIDI Message Generator

MIDI Command ต่างๆและ MIDI Channel, C ของโน้ตที่ได้จากกระบวนการที่ผ่านมาทั้งหมดจะถูกจัดแพ็คเกจข้อมูลให้อยู่ในรูปแบบตามโปรโตคอล (Protocol) ของระบบ MIDI ดังรูปที่ 3.26 เพื่อให้สามารถส่งข้อมูลไปยังอุปกรณ์เอาต์พุต MIDI ผ่านการสื่อสารแบบ UART ได้ดังที่ได้อธิบายไว้ในหัวข้อที่ 2.4 โดยจะเรียกเอาต์พุตที่ได้สุดท้ายนี้ว่า “MIDI Message” ซึ่งจะยกตัวอย่างโค้ดภาษา C ของฟังก์ชันที่ใช้สร้าง MIDI Message สำหรับ Note ON/OFF, Vibrato, Pitch bend ดังรูปที่ 3.27, 3.28 และ 3.39 ตามลำดับ



รูปที่ 3.26 รูปแบบเอาต์พุตที่ได้จาก MIDI Message Generator ตามโปรโตคอลของระบบ MIDI

```
void Note_On(unsigned char channel, unsigned char pitch, unsigned char velocity)
{
    if(velocity > 127)    velocity = 127;
    USART3_SendData(0x90|channel);
    USART3_SendData(pitch);
    USART3_SendData(velocity);
}
void Note_Off(unsigned char channel, unsigned char pitch)
{
    USART3_SendData(0x80|channel);
    USART3_SendData(pitch);
    USART3_SendData(0x00);
}
```

รูปที่ 3.27 ฟังก์ชัน C ที่ใช้สร้าง MIDI Message สำหรับ Note ON/OFF

```
void Vibrato_Control(unsigned channel,int vibrato_level) //coarse mode with value from 0-127
{
    USART3_SendData(0xB0|channel);
    USART3_SendData(0x01);
    if(vibrato_level > 127)    vibrato_level = 127;
    USART3_SendData(vibrato_level);
}
```

รูปที่ 3.28 ฟังก์ชัน C ที่ใช้สร้าง MIDI Message สำหรับ Vibrato

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void Pitch_Bend(unsigned char channel,int bend_level) //middle value at 0, max,min vaule at +8191,-8191
{
  int pitch_value;
  int LSB;
  int MSB;

  pitch_value = 8192 + bend_level;
  if(pitch_value > 16383) pitch_value = 16383; //overflow protection

  LSB = (pitch_value & 0b01111111);
  MSB = ((pitch_value >> 7) & 0b01111111);

  USART3_SendData(0xE0|channel);
  USART3_SendData((unsigned char)(LSB));
  USART3_SendData((unsigned char)(MSB));
}

```

รูปที่ 3.29 ฟังก์ชัน C ที่ใช้สร้าง MIDI Message สำหรับ Pitch bend

การประมวลผลสัญญาณดิจิทัลสำหรับบอร์ดประมวลผลหลักที่ได้กล่าวมาทั้งหมดนี้จะใช้บอร์ดไมโครคอนโทรลเลอร์ตระกูล ARM Cortex-M4 รุ่น STM32F4Discovery (STM32F407VGT6) จากบริษัท STmicroelectronic ซึ่งมีความเร็วสูงสุดอยู่ที่ 168MHz ต่อ 210 DMIPS แสดงหน้าตาของบอร์ดที่ใช้ดังรูปที่ 3.30



- STM32F407VGT6 microcontroller featuring 1 MB of Flash memory, 192 KB of RAM in an LQFP100 package
- On-board ST-LINK/V2 with selection mode switch to use the kit as a standalone ST-LINK/V2 (with SWD connector for programming and debugging)
- Board power supply: through USB bus or from an external 5V supply voltage
- External application power supply: 3V and 5V
- LIS302DL or LIS3DSH, ST MEMS motion sensor, 3-axis digital output accelerometer
- MP45DT02, ST MEMS audio sensor, omnidirectional digital microphone
- CS43L22, audio DAC with integrated class D speaker driver
- Eight LEDs:
 - LD1 (red/green) for USB communication
 - LD2 (red) for 3.3V power on
 - Four user LEDs, LD3 (orange), LD4 (green), LD5 (red) and LD6 (blue)
 - 2 USB OTG LEDs LD7 (green) VBus and LD8 (red) over-current
- Two pushbuttons (user and reset)
- USB OTG with micro-AB connector
- Extension header for LQFP100 I/Os for quick connection to prototyping board and easy probing

รูปที่ 3.30 บอร์ด STM32F4Discovery พร้อมละเอียดของอุปกรณ์ต่างๆบนบอร์ด
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

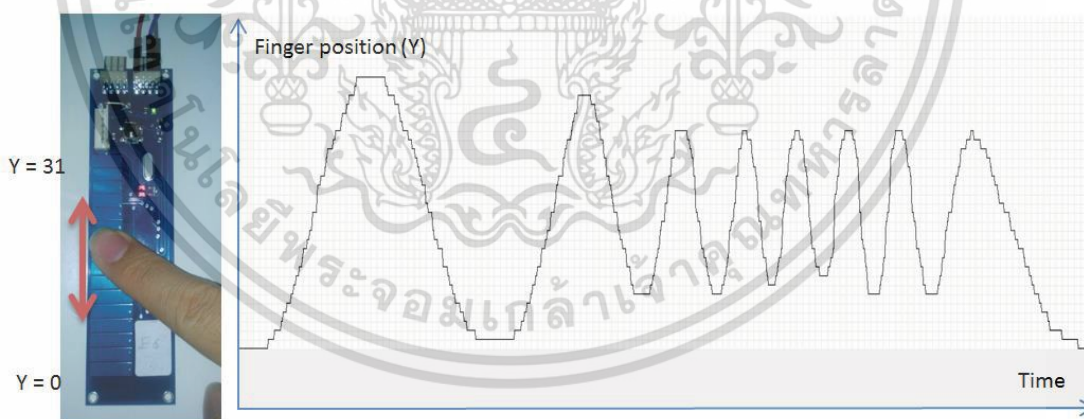
บทที่ 4

ผลการทดสอบของระบบที่นำเสนอ

ในบทนี้จะกล่าวถึงทดสอบอุปกรณ์ที่ใช้ในระบบควบคุมฟังก์ชัน Pitch bend และ Vibrato ที่นำเสนอและเปรียบเทียบสมรรถนะระหว่างระบบที่นำเสนอกับระบบดั้งเดิม

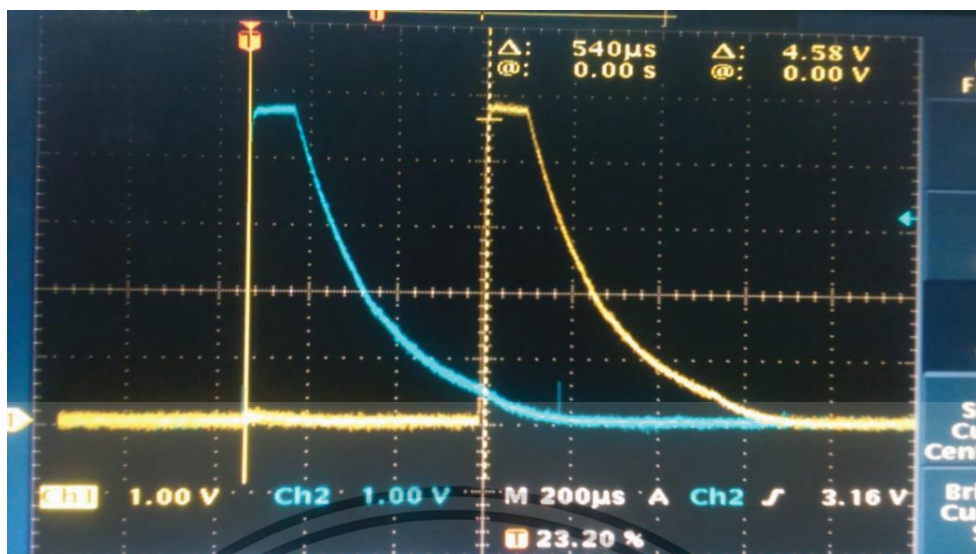
4.1 ทดสอบการทำงานของเซ็นเซอร์สัมผัสที่ใช้

ข้อมูลตำแหน่งสัมผัส Y ที่ได้จากเซ็นเซอร์ระหว่างการทดสอบรูตนิ้วขึ้นลงแบบลักษณะทั่วไป ซึ่งความเร็วการรูตนิ้วจะไม่เกินระดับที่เล่นบนเครื่องสาย แสดงดังรูปที่ 4.1 โดยที่ความละเอียดการแบ่งระดับเท่ากับ 32 ระดับ อัตราการสุ่มตำแหน่งสัมผัสอยู่ที่ 100 fps ส่วนการทำงานของวงจรวัดความจุไฟฟ้าของเซ็นเซอร์สัมผัสและผลของสัญญาณรบกวน 50 Hz ต่อการวัดแรงดันของเซ็นเซอร์ที่กล่าวไว้ในหัวข้อที่ 3.2.1 แสดงไว้ดังรูปที่ 4.2 และ 4.3 ตามลำดับ

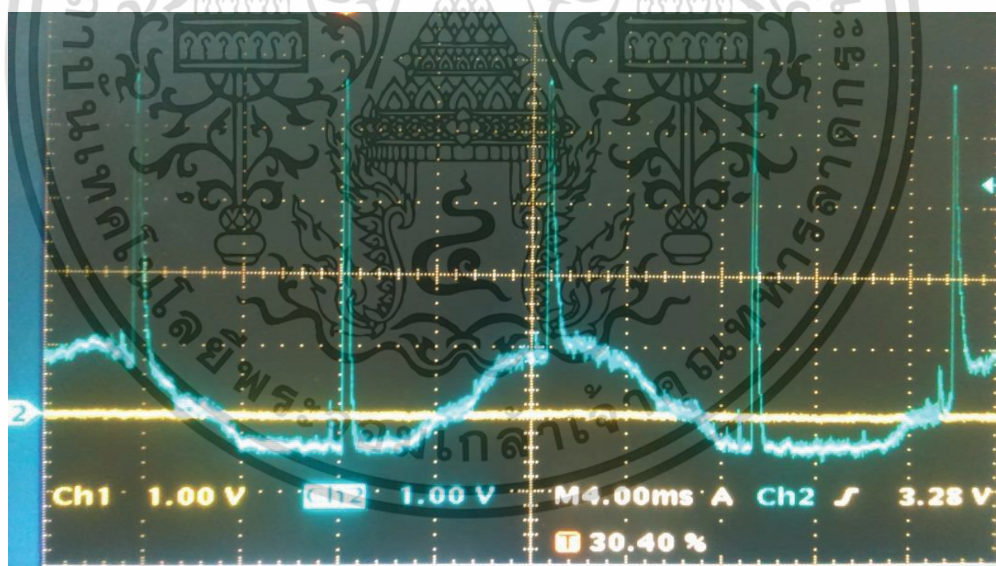


รูปที่ 4.1 ลักษณะข้อมูลตำแหน่งสัมผัส Y ที่ได้จากเซ็นเซอร์ระหว่างการรูตนิ้วทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 (สีฟ้า) กราฟแรงดันวงจร RC ของขั้วสัมผัสชั้นที่ 1, (สีเหลือง) กราฟแรงดันวงจร RC ของขั้วสัมผัสชั้นที่ 2 ซึ่งอยู่ถัดต่อไปจากชั้นที่ 1 ซึ่งมีระยะห่างของเวลาการทำงานเท่ากับ 540 us

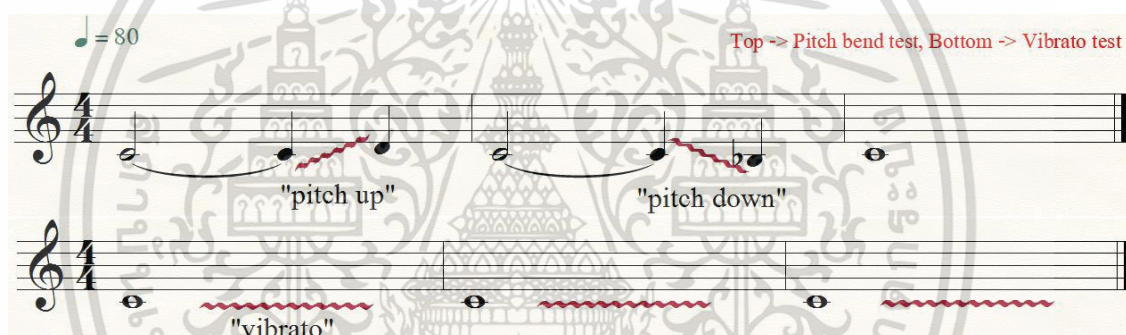


รูปที่ 4.3 ผลของสัญญาณรบกวน 50 Hz ที่มีต่อการวัดแรงดันวงจร RC ของขั้วสัมผัสบนเซ็นเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 เปรียบเทียบสมรรถนะระหว่างระบบที่นำเสนอกับระบบดั้งเดิม

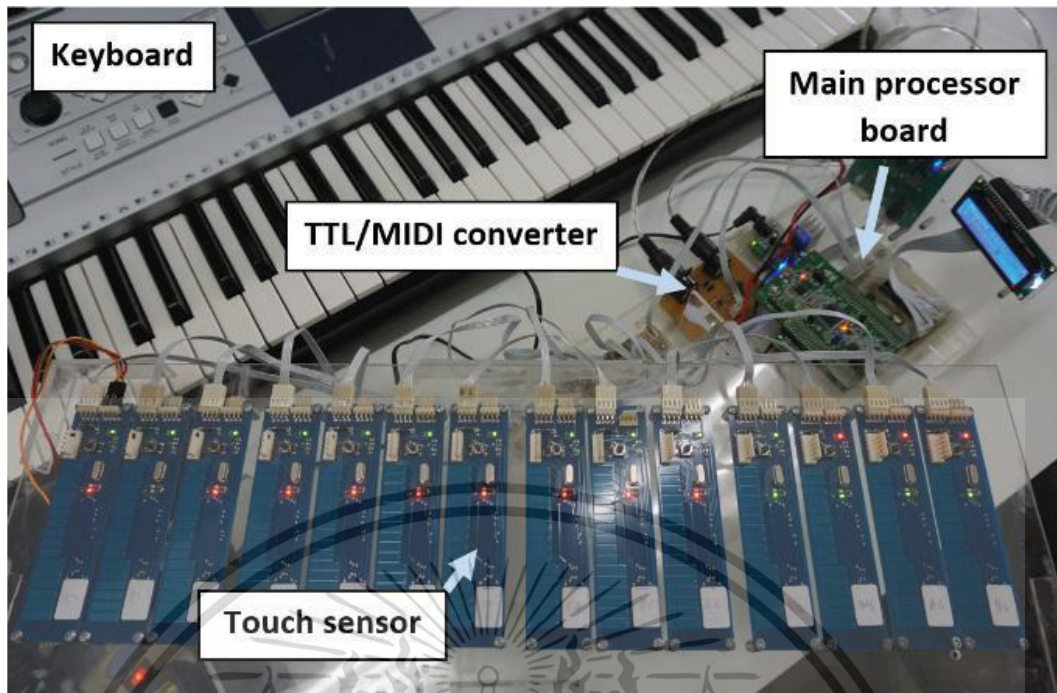
สมรรถนะของระบบควบคุมฟังก์ชัน Pitch bend และ Vibrato ที่นำเสนอ นั้น จะใช้การเปรียบเทียบความเหมือนของลักษณะการเปลี่ยนแปลงระดับโน้ต และความเร็วของการเปลี่ยนแปลงระดับโน้ตที่ตอบสนองต่ออินพุตของผู้เล่นโดยใช้การเล่นโน้ตทดสอบบนเครื่องดนตรีที่ติดตั้งระบบที่นำเสนอและเครื่องดนตรีระบบดั้งเดิม พารามิเตอร์ที่ถูกเลือกสำหรับใช้เปรียบเทียบสมรรถนะระหว่างระบบทั้งสอง คือ ช่วงเวลาที่ใช้สำหรับเปลี่ยนระดับโน้ต 1 semitone โดยเฉลี่ยจากช่วงขาขึ้นและลงของโน้ตระหว่างการเล่นโน้ตทดสอบฟังก์ชัน Pitch bend โดยจะใช้สัญลักษณ์ t_r สำหรับพารามิเตอร์นี้ โดยโน้ตทดสอบดังกล่าวจะประกอบด้วย การทดสอบสำหรับการเล่น Pitch bend และการเล่น Vibrato ซึ่งแสดงไว้ดังรูปที่ 4.4



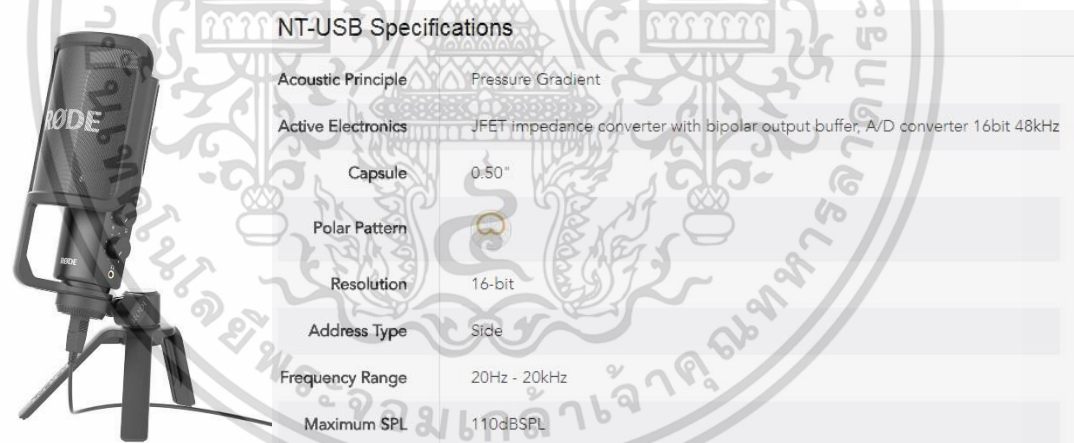
รูปที่ 4.4 โน้ตทดสอบการเล่นฟังก์ชัน Pitch bend (บน) และ Vibrato (ล่าง)

จากนั้นทำการวัดเสียงโน้ตที่ถูกเล่นด้วยไมโครโฟน (Microphone) เพื่อนำไปวิเคราะห์เพื่อเปรียบเทียบสมรรถนะ โดยฮาร์ดแวร์สำหรับการทดสอบระบบที่นำเสนอ นั้นแสดงดังรูปที่ 4.5 และไมโครโฟนที่ใช้ในการบันทึกเสียงผลการทดลองคือ ไมโครโฟนอเนกประสงค์ระดับสตูดิโอ (Studio Quality) ของ Rode รุ่น Rode NT-USB แสดงดังรูปที่ 4.6 เสียงตั้งค่าเครื่องดนตรี MIDI ของคีย์บอร์ดระหว่างการทดสอบคือ ทรัมเป็ต (Trumpet, MIDI Instrument number = 56) เนื่องจากปัญหาของขนาดเซ็นเซอร์ที่ยังใหญ่เกินไป ทำให้ยังไม่สามารถติดตั้งบนลิ้มจริงๆได้ จะแก้ปัญหาโดยติดตั้งเซ็นเซอร์สัมผัสนอกลิ้มคีย์บอร์ดที่แสดงไว้ในรูปที่ 4.5 ซึ่งเพียงพอสำหรับทดสอบอัลกอริทึมการควบคุมฟังก์ชัน Pitch bend และ Vibrato ที่นำเสนอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



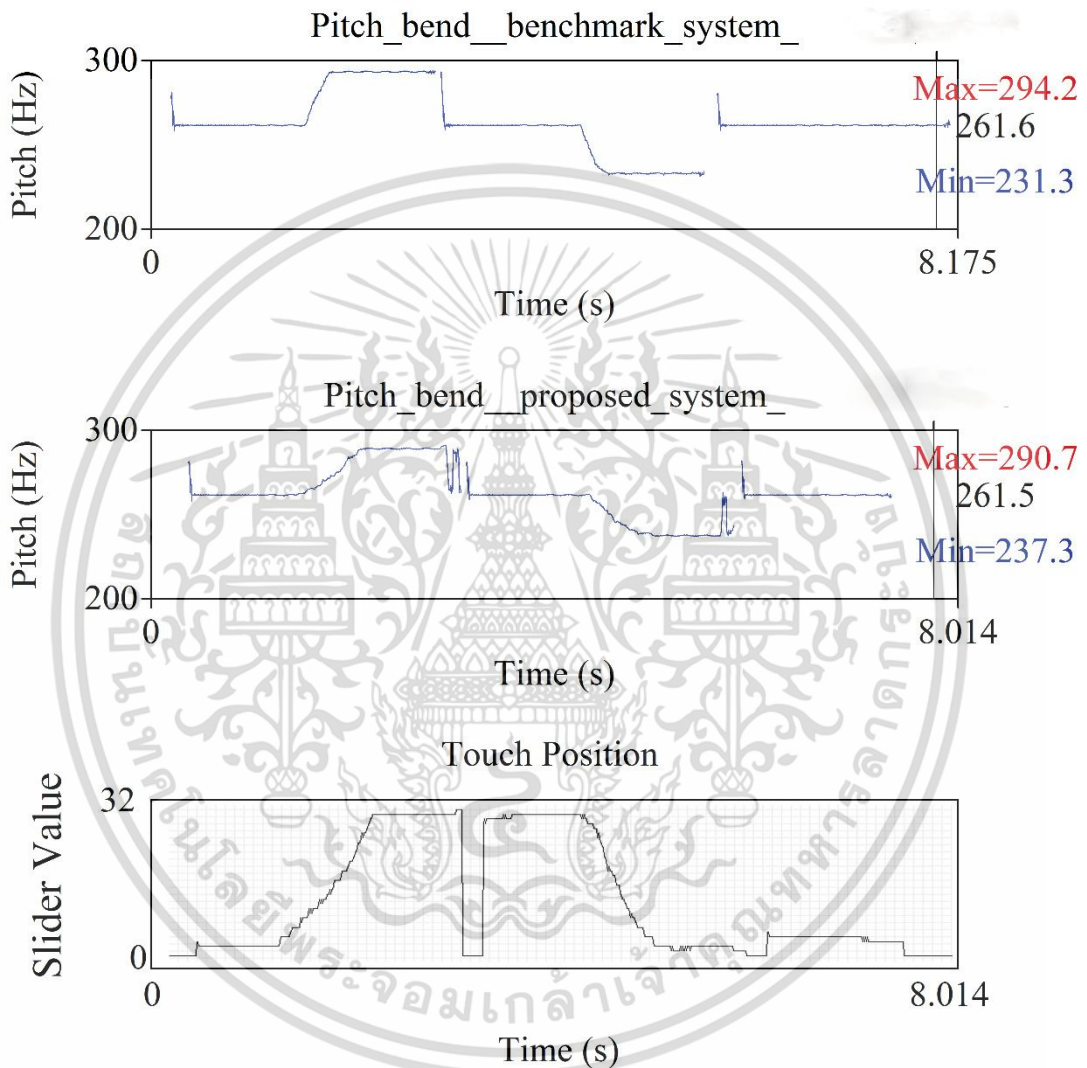
รูปที่ 4.5 ฮาร์ดแวร์สำหรับการทดสอบระบบที่นำเสนอ



รูปที่ 4.6 ไมโครโฟน Rode NT-USB พร้อมรายละเอียด Specification

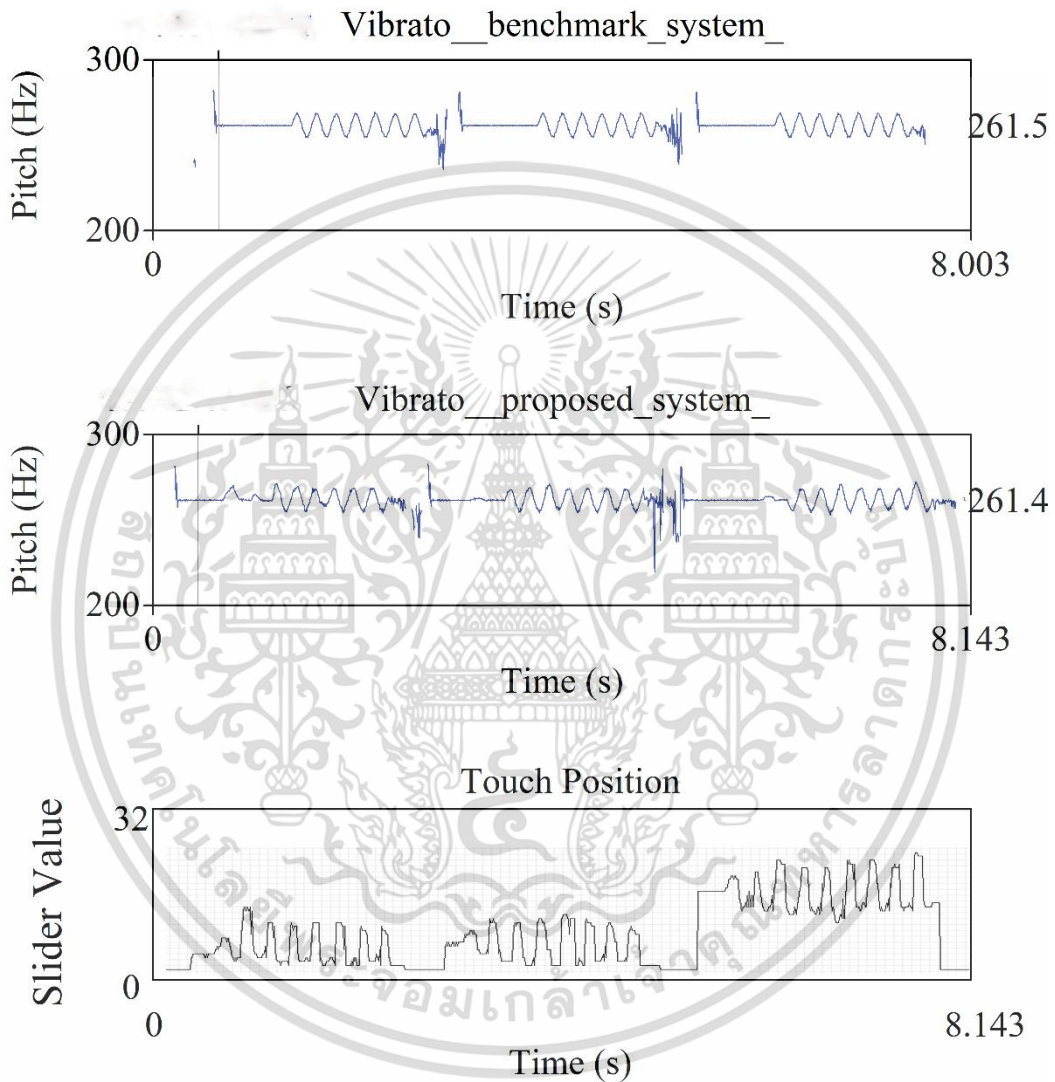
ซอฟต์แวร์ “Praat” [8] ถูกนำมาใช้สำหรับเป็นเครื่องช่วยในวิเคราะห์ทางด้านเสียงซึ่งออกแบบมาสำหรับงานด้านการประมวลผลสัญญาณเสียงสนทนา (Speech Signal Process) โดยซอฟต์แวร์สามารถสกัดระดับโน้ต (Pitch Contour) ออกจากไฟล์เสียงจากผลการทดลองและพล็อต Spectrogram พร้อมแสดงระดับโน้ตได้ ซึ่งจะแสดงผลการทดลองดังรูปที่ 4.7 และ 4.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 กราฟระดับโน้ตจากการทดสอบ Pitch bend ด้วยระบบดั้งเดิม (บน), ระบบที่นำเสนอ (กลาง) และ กราฟตำแหน่งนิ้ว Y บนเซ็นเซอร์ของคีย์โน้ต C4 (ล่าง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 กราฟระดับโน้ตจากการทดสอบ Vibrato ด้วยระบบดั้งเดิม (บน), ระบบที่นำเสนอ (กลาง) และ กราฟตำแหน่งนิ้ว Y บนเซ็นเซอร์ของคีย์โน้ต C4 (ล่าง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การคำนวณหาอัตราการเปลี่ยนแปลงระดับโน้ตที่ได้กล่าวไว้ในหัวข้อที่ 2.5.1 (การรับรู้ความแตกต่างของระดับโน้ตใน Musical Scale ของมนุษย์) จากผลการทดลองที่ได้ในรูปที่ 4.4 จำเป็นต้องแปลงหน่วยของผลต่างความถี่ของโน้ตที่มีหน่วยเป็น Hz ให้เป็นผลต่างตำแหน่งโน้ตในหน่วย semitone ด้วยสมการที่ 2.2 เพื่อให้ข้อมูลที่ได้สามารถใช้อธิบายสมรรถนะของระบบที่นำเสนอในทุกๆตำแหน่งโน้ตได้ หลังจากการคำนวณผลการทดสอบพบว่า ฟังก์ชัน Pitch bend ของระบบดั้งเดิมและระบบที่นำเสนอ ใช้เวลาสำหรับเปลี่ยนระดับโน้ต 1 semitone โดยเฉลี่ยเท่ากับ 0.138 วินาที และ 0.276 วินาที ตามลำดับ

$$t_{r(\text{benchmark})} = 0.138 \text{ sec}, \quad t_{r(\text{proposed})} = 0.276$$

ส่วนการทดสอบฟังก์ชัน Vibrato ดังรูปที่ 4.6 นั้น ระบบที่นำเสนอให้ลักษณะการเปลี่ยนความถี่โน้ตได้ใกล้เคียงกับระบบดั้งเดิม จากผลการทดลองโดยรวมแล้ว ระบบที่นำเสนอทำงานสอดคล้องกับอัลกอริทึมที่นำเสนอตามที่ได้ออกแบบไว้ในบทที่ 3 และสามารถนำมาใช้ทดแทนระบบดั้งเดิมสำหรับการเล่นดนตรีโดยทั่วๆไปได้

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

ระบบควบคุมฟังก์ชัน Pitch bend และ Vibrato สำหรับเครื่องดนตรีอิเล็กทรอนิกส์ คีย์บอร์ดที่นำเสนอ นั้น สามารถใช้แทนระบบดั้งเดิมที่ใช้ล้อคันโยกสำหรับการเล่นคีย์บอร์ดด้วยเทคนิคต่างๆไปได้และยังให้ความรู้สึกเป็นธรรมชาติมากกว่าระบบดั้งเดิม และระบบที่นำเสนอนี้รองรับการควบคุม Pitch bend และ Vibrato ของแต่ละโน้ตแบบอิสระต่อกันได้ด้วยเทคนิค MIDI Channel Modulation ที่นำเสนอและทำให้สามารถเล่น Baseline กับ Melody ด้วยมือสองข้างพร้อมกันได้ อย่างอิสระขณะที่เรียกใช้งานฟังก์ชันซึ่งทำไม่ได้ในระบบดั้งเดิม สมรรถนะด้านความเร็วการต่อการ เล่นนั้น ระบบใช้เซ็นเซอร์สัมผัสที่มีอัตราสุ่มตำแหน่งสัมผัสอยู่ที่ 120 FPS ซึ่งระบบโดยรวมสามารถ สร้างการเปลี่ยนแปลงระดับโน้ตในฟังก์ชัน Pitch bend ที่ยังคงช้ากว่าแบบระบบดั้งเดิมเนื่องจาก ข้อจำกัดของเทคนิค Capacitive Touch บวกกับการเพิ่มการประมวลผลสัญญาณดิจิทัล แต่ระบบก็ ยังเพียงพอสำหรับการเล่นทั่วไปที่โน้ตไม่เปลี่ยนรวดเร็วมาก โดยช่วงเวลาสำหรับเปลี่ยนระดับโน้ต 1 semitone โดยเฉลี่ยของระบบดั้งเดิมและระบบที่นำเสนออยู่ที่ 0.138 วินาที และ 0.276 วินาที ตามลำดับ ระบบยังไม่สามารถติดตั้งบนลิ้มคีย์บอร์ดได้จริงเนื่องจากเซ็นเซอร์ยังมีขนาดใหญ่กว่าขนาด ของลิ้มคีย์ ซึ่งสามารถแก้ไขได้ในอนาคต

5.2 ข้อเสนอแนะ

ปัญหาของอัตราสุ่มตำแหน่งนิ้วของเซ็นเซอร์สัมผัสที่น้อยเกินไปและขนาดเซ็นเซอร์สัมผัสที่มี ขนาดใหญ่เกินไป อาจจะแก้ไขได้โดยใช้ชิพ Capacitive Touch Sensor เฉพาะทางที่ใช้เทคนิคการวัด ความจุไฟฟ้าแบบอื่นที่ใช้เวลาในการวัดเร็วกว่าเทคนิคที่ใช้ช่วยในการวัดความจุไฟฟ้า หรืออาจจะ เพิ่มความเร็วการประมวลผล DSP โดยเปลี่ยนไมโครคอนโทรลเลอร์ที่มีประสิทธิภาพการคำนวณสูงกว่า ได้ ตัวอย่างเช่น MCU ตระกูล ARM Cortex-M0 และการใช้ชิพ Capacitive Touch Sensor เฉพาะ ทางอาจจะทำให้วงจรใช้อุปกรณ์น้อยลงพร้อมกับใช้แผ่น PCB แบบ 4 ชั้นเพื่อลดความหนาของแผ่น เซ็นเซอร์ เซ็นเซอร์อาจจะเล็กลงจนสามารถติดตั้งบนลิ้มคีย์ได้ และควรออกแบบใหม่ให้ระบบเพิ่ม อัตราการสุ่มตำแหน่งสัมผัส (Frame rate) ให้มากขึ้นเพื่อเพิ่มสมรรถนะของระบบให้ตอบสนองต่อการ เล่นโน้ตได้ไวเท่ากับคีย์บอร์ดระบบดั้งเดิมที่ใช้ล้อคันโยกเป็นตัวควบคุมได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] Vincent Chan and Steve Underwood, “MSP430 Capacitive Single-Touch Sensor Design Guide”, Texas Instruments, 2008.
- [2] Oscar Camacho and Eduardo Viramontes, “Designing Touch Sensing Electrodes”, Freescale Semiconductor, 2011.
- [3] Guy D. Moore, *The Physics and Psychophysics of Music: Lecture 16*, “Intervals, Scales, and Tunings”, TU Darmstadt, 2006.
- [4] Piano Key Frequencies.[Online].เข้าถึงได้จาก:
https://en.wikipedia.org/wiki/Piano_key_frequencies
- [5] Moog, R. A., and Rhea, “T. L. Evolution of the keyboard interface: The Bosendorfer. 290 SE recording piano and the Moog multiply-touch-sensitive keyboards.”, *Computer Music Journal* 14, 2 (Summer 1990), 52–60
- [6] The MIDI Manufacturers Association, Document Version 4.2, “MIDI 1.0 Detailed Specification”, Los Angeles, CA, 1995.
- [7] Tim Wescott, “Sampling: What Nyquist Didn’t Say, and What to Do About It”
- [8] Paul Boersma and David Weenink. “Praat : doing phonetics by computer”, Internet : <http://www.fon.hum.uva.nl/praat>
- [9] Unary coding.[Online].เข้าถึงได้จาก:
https://en.wikipedia.org/wiki/Unary_coding

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลงานที่ได้รับการตีพิมพ์

นัทรพงศ์ วัฒนศิริ, นพสิทธิ์ อริยพลิชฐ์, สุรพันธุ์ เอื้อไพบูลย์ และ กสิน วิเชียรชม. “ระบบควบคุม ฟังก์ชัน Pitch Bend และ Vibrato.” ด้วยเซ็นเซอร์สัมผัสสำหรับเครื่องดนตรีคีย์บอร์ดชนิดลิ้มเปียโน”, 8th ECTI-CARD 2016, Prachuap Khiri Khan, Thailand, 27 July 2016.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ-นามสกุล	นายณัฏพงค์ วัฒนศิริ
วัน เดือน ปีเกิด	29 มีนาคม 2536
ที่อยู่	9/3 หมู่ 8 ต.บางสมัคร อ.บางปะกง จ.ฉะเชิงเทรา 24180
ประวัติการศึกษา	2550 - 2555 โรงเรียนมารदानุมล 2555 - 2559 วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมอิเล็กทรอนิกส์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ความชำนาญเฉพาะด้าน	1.) Audio Signal Processing 2.) Control System Design 3.) Embedded System

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้