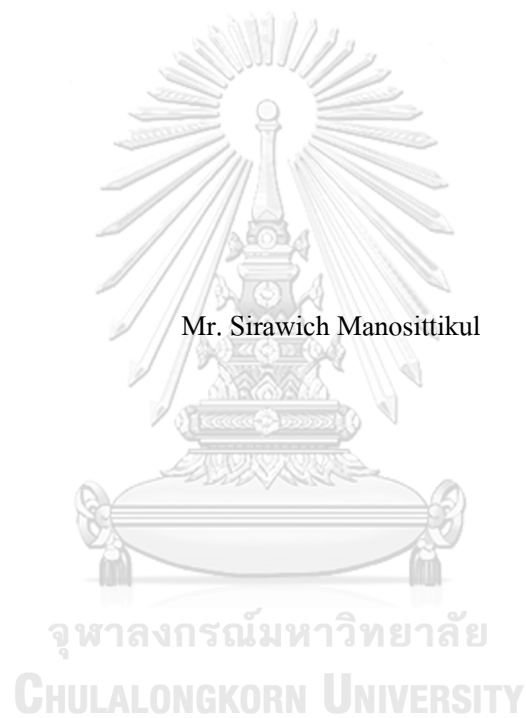


การประยุกต์ใช้คอมพิวเตอร์ซึ่งกับการวัดการสิ้นสะท้อนด้วยเซ็นเซอร์อัตราสุ่มจุดข้อมูลไม่
แน่นอน



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมเครื่องกล ภาควิชาวิศวกรรมเครื่องกล
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2565
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

Implementation of Compressed sensing technique for vibration measurement with Non-Uniform
sampling



A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Mechanical Engineering

Department of Mechanical Engineering

FACULTY OF ENGINEERING

Chulalongkorn University

Academic Year 2022

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การประยุกต์ใช้คอมพิวเตอร์เซมซิงกับการวัดการสิ้นสะท้อนด้วยเซ็นเซอร์อัตราสุ่มจุดข้อมูลไม่แน่นอน
โดย	นายศิริวิชญ์ มโนสิทธิกุล
สาขาวิชา	วิศวกรรมเครื่องกล
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร.ชนัตต์ รัตนสุมาวงศ์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้รับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

.....	คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)	
คณะกรรมการสอบวิทยานิพนธ์	ประธานกรรมการ
.....	
(รองศาสตราจารย์ ดร.นภคณีย์ อาชวาคม)	
.....	อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(รองศาสตราจารย์ ดร.ชนัตต์ รัตนสุมาวงศ์)	
.....	กรรมการ
(อาจารย์ ดร.รีนา เซย์)	
.....	กรรมการภายนอกมหาวิทยาลัย
(ผู้ช่วยศาสตราจารย์ ดร.เขมพัฒน์ ตันตวิวัฒน์กุล)	

ศิริวิชญ์ มโนสิทธิกุล : การประยุกต์ใช้คอมเพรสเซนซิงกับการวัดการสั่นสะเทือนด้วย
เซ็นเซอร์อัตราสุ่มจุดข้อมูลไม่แน่นอน. (Implementation of Compressed sensing
technique for vibration measurement with Non-Uniform sampling) อ.ที่ปรึกษาหลัก :
รศ. ดร.ชนัดต์ รัตนสุมาวงศ์

งานวิจัยนี้ศึกษาความเป็นไปได้ที่จะประยุกต์ใช้วิธีคอมเพรสเซนซิงกับการวัดการ
สั่นสะเทือนด้วยเซ็นเซอร์ที่มีอัตราการสุ่มจุดข้อมูลไม่แน่นอน เซ็นเซอร์ประเภทนี้มักจะมี
ความถี่สัญญาณต่ำด้วยและบ่อยครั้งต่ำกว่าความถี่สูงสุดที่ต้องการวัด วิทยานิพนธ์นี้ใช้
คอมเพรสเซนซิงกับสัญญาณที่วัดจากเซ็นเซอร์ดังกล่าว เพื่อประมวลผลสัญญาณให้ได้ข้อมูล
ในช่วงความถี่สูงจนถึงความถี่สูงสุดที่ต้องการ วิธีการศึกษาแบ่งออกเป็น 2 ส่วนคือ ส่วนแรก
เป็นการทดสอบเพื่อหาความเป็นไปได้ในเชิงหลักการ ซึ่งสัญญาณที่จะนำไปใช้สร้างจากการจำลอง
ในคอมพิวเตอร์ และการวัดด้วยระบบการวัดการสั่นสะเทือนมาตรฐาน ก่อนที่จะสุ่มจุดข้อมูลเพื่อ
สร้างสัญญาณที่มีอัตราสุ่มจุดไม่แน่นอน ซึ่งจะนำไปใช้กับคอมเพรสเซนซิงต่อไป ส่วนการ
ทดสอบส่วนที่สองเป็นการทดสอบเพื่อหาความเป็นไปได้ในการใช้คอมเพรสเซนซิงในทาง
ปฏิบัติ ในส่วนนี้สัญญาณการสั่นสะเทือนซึ่งเป็นสัญญาณที่มีอัตราสุ่มจุดไม่แน่นอนจะถูกวัดจาก
เครื่องจักรโดยตรง จากผลการศึกษาความเป็นไปได้เชิงหลักการพบว่า คอมเพรสเซนซิงสามารถ
สร้างสัญญาณคืนกลับได้ใกล้เคียงกับสัญญาณตั้งต้น ถึงแม้ว่าสัญญาณที่มีอัตราสุ่มจุดไม่แน่นอน
จะมีความถี่สุ่มเทียบเท่าต่ำกว่าอัตราในควิสต์ อย่างไรก็ตามจากการทดสอบความเป็นไปได้
ในทางปฏิบัติพบว่า หากสัญญาณที่มีอัตราสุ่มจุดไม่แน่นอนไม่แม่นยำแล้ว คอมเพรสเซนซิงจะ
สร้างสัญญาณคืนกลับที่ต่างจากสัญญาณตั้งต้นอย่างมาก คุณภาพของสัญญาณคืนกลับสามารถ
ปรับปรุงให้เพิ่มขึ้นได้โดยการเพิ่มระยะเวลาในการเก็บสัญญาณให้ยาวนานมากขึ้น หรือการเพิ่ม
ความถี่สุ่มเทียบเท่า ซึ่งจะส่งผลให้สัญญาณคืนกลับที่สร้างได้ใกล้เคียงกับสัญญาณตั้งต้นมาก
ยิ่งขึ้น

สาขาวิชา วิศวกรรมเครื่องกล

ลายมือชื่อนิติต

ปีการศึกษา 2565

ลายมือชื่อ อ.ที่ปรึกษาหลัก

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงลงได้ด้วยความกรุณาของ รองศาสตราจารย์ ดร. ชนิตต์ รัตน์ สุมาวงศ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้คำแนะนำ คำปรึกษา ข้อคิดเห็นอันเป็นประโยชน์ต่อ วิทยานิพนธ์ และความอนุเคราะห์ด้านอื่นๆ อีกหลายประการด้วยดีเสมอมา ผู้วิจัยรู้สึกซาบซึ้งใน พระคุณและความกรุณาเป็นอย่างยิ่งจึงขอกราบขอบพระคุณเป็นอย่างสูง

ขอกราบขอบพระคุณ รองศาสตราจารย์ ดร. นภคณัย อาชวาคม ประธานกรรมการ อ.ดร ีนา เซย์ และ ผู้ช่วยศาสตราจารย์ ดร. เจมพัฒน์ ตันติวัฒน์กุล กรรมการที่ให้คำแนะนำตลอดระยะเวลา ทำงานวิจัย ที่ทำให้วิทยานิพนธ์ฉบับนี้มีความครบถ้วนสมบูรณ์มากขึ้น

ขอขอบคุณ คณะวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมเครื่องกล จุฬาลงกรณ์มหาวิทยาลัย ที่ ให้ทุนสนับสนุนงานวิจัยนี้

สุดท้ายนี้ผู้วิจัยขอกราบขอบพระคุณ บิดามารดาและพี่สาว ที่ให้กำลังใจและสนับสนุน การศึกษาของผู้วิจัยเสมอมา และคุณค่าอันใดที่ได้รับจากวิทยานิพนธ์นี้ ขอมอบเป็นกตัญญูบูชาแด่ บิดา มารดา ครูบาอาจารย์ ตลอดจนผู้มีพระคุณทุกท่าน

ศิริวิชญ์ มโนสิทธิกุล

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ง
กิตติกรรมประกาศ.....	จ
สารบัญ.....	ฉ
สารบัญรูปภาพ	ญ
สารบัญตาราง	บ
คำอธิบายสัญลักษณ์และคำย่อ	ป
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์.....	4
1.3 ขอบเขตของวิทยานิพนธ์	5
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	5
บทที่ 2 ความรู้พื้นฐานและงานวิจัยที่เกี่ยวข้อง	6
2.1 การแก้ปัญหาระบบสมการเชิงเส้นและสัญญาณสปาร์ส.....	6
2.1.1 ระบบสมการเชิงเส้นไม่ชัดเจน (Underdetermine system)	6
2.1.2 ระบบสมการเชิงเส้นชัดเจนเกินไป (Overdetermined system)	9
2.1.2.1 การแยกค่าเอกฐาน และ มัวร์-เพนโรส อินเวอร์ส (SVD and MoorePenrose inverse)	9
2.1.3 สัญญาณสปาร์ส (Sparse signal) และผลเฉลยสปาร์ส (Sparse solution)	12
2.2 การบีบอัดสัญญาณ (Signal Compression)	12
2.3 การเปลี่ยนฐานหลักและเมทริกซ์การแปลง	15

2.3.1	เมทริกซ์การแปลงฟูรีเยร์แบบไม่ต่อเนื่อง (Discrete Fourier Transform Matrix).....	15
2.3.2	การแปลงโคไซน์แบบไม่ต่อเนื่อง (Discrete Cosine Transform Matrix).....	15
2.3.3	ความสัมพันธ์ของแถวและหลักของเมทริกซ์การแปลงกับค่าสัญญาณในโดเมนเวลา และในโดเมนความถี่.....	19
2.4	คอมเพรสเซนซิง(Compressed sensing)	20
2.4.1	กระบวนการหาจุดที่ดีที่สุดแบบคอนเวกซ์ (Convex Optimization-CVX)	23
2.4.2	การจับคู่สัญญาณบีบอัด (Compressed sampling Matching Pursuit-COSAMP)	25
2.5	วงจรกรองความถี่ต่ำผ่าน (Low pass Filter).....	25
บทที่ 3	การวัดและการประมวลผลสัญญาณ.....	28
3.1	การปรับปรุงให้คอมเพรสเซนซิงให้สามารถใช้ได้กับสัญญาณการสั่นสะเทือน.....	28
3.1.1	การได้มาซึ่งสัญญาณสุ่มวัด	28
3.1.2	การจับคู่เวลา (Time Matching) และเมทริกซ์การวัด (Measurement matrix)	30
3.1.3	การสร้างเมทริกซ์ฐานหลักเริ่มต้น (Initial basis).....	31
3.2	พารามิเตอร์ของสัญญาณแบบสุ่มคาบ.....	34
3.3	การสร้างสัญญาณสปาร์สโดยวิธีคอมเพรสเซนซิง.....	35
3.3.1	การสร้างสัญญาณสปาร์สโดยการใช้ค่าเหมาะสมที่สุดแบบคอนเวกซ์	35
3.3.2	การสร้างสัญญาณสปาร์สโดยการจับคู่สัญญาณบีบอัด (Compressed sampling Matching Pursuit COSAMP).....	37
3.3.2.1	การสร้างสัญญาณตัวแทน(Proxy Signal)	37
3.3.2.2	การหาสัญญาณคืนกลับ	40
3.3.2.3	ขั้นตอนการทำงานของอัลกอริทึม	41
3.3.3	การหาสัญญาณสปาร์สโดยให้วิธีการเฉลี่ยสัญญาณในโดเมนความถี่	45
บทที่ 4	การทดสอบการใช้คอมเพรสเซนซิงกับการวัดการสั่นสะเทือน.....	52

4.1. ภาพรวมของการทดสอบ	52
4.2. การทดสอบหาความเป็นไปได้ในเชิงหลักการ (Proof of concept).....	54
4.2.1. จำลองการวัดสัญญาณต้นสะท้อนบนคอมพิวเตอร์	54
4.2.2. การทดสอบโดยการวัดการต้นสะท้อนจากการวัดเครื่องจักรจริงด้วยระบบการวัด มาตรฐาน.....	58
4.3 การทดสอบหาความเป็นไปได้ในทางปฏิบัติจริง (Proof of work).....	64
4.4 การประเมินสัญญาณ	67
4.5 สรุปวิธีการทดสอบและจุดมุ่งหมายในการทดสอบ	73
บทที่ 5. ผลการประยุกต์ใช้คอมพิวเตอร์เซชันซิงกับการวัดการต้นสะท้อน	76
5.1 ผลการทดสอบหลักการ (Proof of concept).....	76
5.1.1 การจำลองสัญญาณการวัดบนคอมพิวเตอร์	76
5.1.2 การสร้างสัญญาณกลับคืนจากสัญญาณจริงที่ถูกวัดแบบปกติ.....	81
5.1.2.1 ผลการทดสอบจากการวัดสัญญาณจากเครื่องสร้างการต้นสะท้อน	81
5.1.2.2 ผลการวัดสัญญาณของคอมพิวเตอร์แบบ 1 ลูกสูบ	83
5.1.2.3 ผลการทดสอบจากการวัดสัญญาณจากคอมพิวเตอร์ 2 ลูกสูบ	87
5.1.2.4 การใช้ชุดข้อมูลที่กว้างขึ้นในการทดสอบ	91
5.1.2.5 สรุปการทดสอบหลักการ	96
5.2 การทดสอบการนำไปใช้จริง (Proof of practice)	97
5.2.1 การวัดการต้นสะท้อนจากเครื่องสร้างสัญญาณ	98
5.2.2 ผลการวัดการต้นสะท้อนของเครื่องจักรจริง	101
5.2.3 การใช้อัลกอริทึมอื่นๆทดแทนการใช้ การหาค่าเหมาะสมที่สุดแบบคอนเวกซ์	115
5.3 สรุป	127
บทที่ 6 บทสรุป.....	128
6.1 สารสำคัญของวิทยานิพนธ์	128

6.2 ข้อเสนอแนะ	129
บรรณานุกรม	133
ภาคผนวก ก. อุปกรณ์ และ เครื่องมือที่ใช้ทำการทดสอบ	134
ภาคผนวก ข. อัลกอริทึมคอมพิวเตอร์เชิงและ ฟังก์ชันที่ใช้งานอื่นๆ ในโปรแกรม MATLAB...146	
ภาคผนวก ค. หลักการสุ่มคาบโดยใช้บอร์ดไมโครคอนโทรลเลอร์ STM32F411RE	153
ภาคผนวก ง. ซอฟต์แวร์ที่ใช้ในการควบคุมการวัดแบบสุ่มคาบของบอร์ดไมโครคอนโทรลเลอร์ STM32F411RE	160
ภาคผนวก จ. การเก็บข้อมูลและปรับปรุงสัญญาณ ของระบบการวัดอ้างอิง (PCB และ DEWESOFT)	172
ภาคผนวก ฉ. ขั้นตอนการเก็บข้อมูลและปรับปรุงสัญญาณ ของระบบการวัดแบบสุ่มคาบ (ADXL345 และ STM32F411RE)	176
ภาคผนวก ช. ผลการทดสอบการจำลองสัญญาณและทดสอบอัลกอริทึมคอมพิวเตอร์เชิง	179
ภาคผนวก ซ. ผลการทดสอบระบบการวัดมาตรฐานทั่วไป ร่วมกับอัลกอริทึมคอมพิวเตอร์เชิง .188	
ภาคผนวก ฌ. ผลการทดสอบระบบการวัดแบบสุ่มคาบของเครื่องสร้างการสั่นสะเทือน.....	202
ประวัติผู้เขียน	259

สารบัญรูปภาพ

	หน้า
รูปที่ 1.1 ตัวอย่างภาพของเสือในความละเอียดต่ำ	3
รูปที่ 1.2 ตัวอย่างสัญญาณการสั่นสะเทือนของเครื่องจักร ก.) สัญญาณในโดเมนเวลา ข.) สัญญาณในโดเมนความถี่ ค.) สัญญาณในโดเมนความถี่ที่ทำการลดสัญญาณรบกวนออกแล้ว	3
รูปที่ 1.3 แผนผังเปรียบเทียบกระบวนการในการสร้างสัญญาณ ก.) งานวิจัยที่ผ่านมา ข.) งานวิจัยนี้	4
รูปที่ 2.1 ตำแหน่งของคำตอบของระบบสมการเชิงเส้นบนระนาบ 2 มิติ ก.) บรรทัดฐาน 1 ข.) บรรทัดฐาน 2	7
รูปที่ 2.2 การเปลี่ยนแปลงรูปร่างของคำตอบจากบรรทัดฐาน 0.25 ถึง อนันต์	8
รูปที่ 2.3 สัญญาณตัวอย่างในโดเมนความถี่และสัญญาณรบกวน	12
รูปที่ 2.4 สัญญาณตัวอย่าง ก.) ก่อนการลดสัญญาณรบกวน ข.) หลังการลดสัญญาณรบกวน	13
รูปที่ 2.5 สัญญาณภาพใน RGB layer ก.) รูปปกติ และ ข.) สัญญาณภาพหลังจากการแปลงโคไซน์	13
รูปที่ 2.6 พื้นที่ใต้กราฟสีเหลืองแสดงถึงจำนวนขนาดข้อมูลความถี่ที่ถูกใช้ไปเพื่อการสร้างสัญญาณภาพที่ถูกบีบอัด โดยขนาดข้อมูลที่ใช้คือ ก.) 99% ข.) 5% ค.) 1% ง.) 0.2%	14
รูปที่ 2.7 สัญญาณภาพที่ถูกสร้างขึ้นตามขนาดข้อมูลความถี่จากรูปที่ 2.6 ก.) 99% ข.) 5% ค.) 1% ง.) 0.2%	14
รูปที่ 2.8 เปรียบเทียบการบีบอัดสัญญาณรูปไซน์โดย ก. ใช้การแปลงฟูเรียร์ ข.) ใช้การแปลงโคไซน์	17
รูปที่ 2.9 ก.) แสดงลักษณะของอนุกรมโคไซน์ ไซน์ และฟูเรียร์ ข.) สัญญาณ $y = f(x) = x$ เมื่อทำการเพิ่มขอบเขตออกไปในทิศทางลบและบวกโดยเงื่อนไขขอบเขตแบบต่างๆ	18
รูปที่ 2.10 สมการ $y_t = C\psi x_s = \theta x_s$ ก.) ก่อนลดมิติ ข. หลังลดมิติ	21
รูปที่ 2.11 พื้นผิวแสดงความสัมพันธ์ระหว่างจำนวนของความถี่ที่ไม่เท่ากับศูนย์ของสัญญาณ K , การกระจายตัวของการสุ่มข้อมูล k_1 , จำนวนจุดที่จำเป็นต้องการวัด M	22
รูปที่ 2.12 สัญญาณต้นแบบและการสุ่มเลือกจุดในสัญญาณ	24
รูปที่ 2.13 สัญญาณทดสอบในโดเมนเวลาและโดเมนความถี่ ก.) สัญญาณต้นแบบ (สีดำ) ข.) สัญญาณสปาร์ส (สีแดง) จากการหาค่าเหมาะสมที่สุดแบบคอนเวกซ์	25
รูปที่ 2.14 แผนภาพวิธีการเลือกผลลัพธ์ของอัลกอริทึมละโมบเปรียบเทียบกับวิธีการหาค่าเหมาะสมที่สุด [18]	26
รูปที่ 2.15 แผนภาพแสดงในกรณีที่สัญญาณที่ไม่ผ่านวงจรกรองความถี่ต่ำผ่าน	26
รูปที่ 2.16 แผนภาพแสดงในกรณีที่สัญญาณผ่านวงจรกรองความถี่ต่ำผ่าน	27
รูปที่ 3.1 แสดงการเลือกจุดจากตำแหน่งของข้อมูลของสัญญาณ x_t เพื่อใช้เป็นสัญญาณ y_t	29
รูปที่ 3.2 การจับคู่เวลาของสัญญาณเพื่อสร้าง แมทริกซ์การวัด C	31
รูปที่ 3.3 ลำดับวิธีการสร้างระบบสมการไม่ชัดเจน ด้วยการจับคู่เวลาและการเพิ่ม/ลด ขนาดแมทริกซ์	34
รูปที่ 3.4 แสดงตัวอย่างของสัญญาณ ก.) x_t ข.) y_t	35

รูปที่ 3.5 อัลกอริทึม คอมเพรสชันซึ่งโดยการใช้ค่าเหมาะสมที่สุดแบบคอนเวจซ์	37
รูปที่ 3.6 ก.) สัญญาณทดสอบจากข้อมูลความถี่ 8192 เฮิรตซ์ ข.) สัญญาณตัวแทนจากสัญญาณเวลา 1030 จุด/วินาที	39
รูปที่ 3.7 ตำแหน่งใน θ ที่สัมพันธ์กับแอมพลิจูดสูงสุด K อันดับแรกของสัญญาณ v_s และการลดขนาดแมทริกซ์ θ	41
รูปที่ 3.8 แสดงวิธีการทำงานของ อัลกอริทึม Cosamp	45
รูปที่ 3.9 ก.) สัญญาณต้นแบบในโดเมนเวลา ข.) สัญญาณตัวแทนในโดเมนความถี่ ค.) สัญญาณต้นแบบในโดเมนความถี่	47
รูปที่ 3.10 เปรียบเทียบสัญญาณตัวแทนในแต่ละรอบการวัดทั้งหมด 4 ครั้งเทียบกับสัญญาณทดสอบ	48
รูปที่ 3.11 ก.) สัญญาณตัวแทนเฉลี่ยในโดเมนความถี่ ข.) สัญญาณต้นแบบในโดเมนความถี่	49
รูปที่ 3.12 แสดงวิธีการทำการของ อัลกอริทึม Mean Stack	51
รูปที่ 4.1 แผนผังแสดงภาพรวมของการทดสอบทั้งหมดในงานวิจัยนี้	54
รูปที่ 4.2 แผนผังแสดงวิธีการทำการทดสอบการจำลองสัญญาณในคอมพิวเตอร์	57
รูปที่ 4.3 แผนภาพแสดงการต่ออุปกรณ์ในงานวิจัย	58
รูปที่ 4.4 ตัวอย่างการติดตั้งเซ็นเซอร์เข้ากับเครื่องสร้างการสั่นสะเทือน	58
รูปที่ 4.5 แผนผังแสดงวิธีการทำการทดสอบการวัดแบบสุ่มหยิบสัญญาณ	60
รูปที่ 4.6 แผนผังแสดงวิธีการทำการทดสอบการวัดแบบสุ่มคาบโดยตรง	62
รูปที่ 4.7 แผนผังแสดงวิธีการจำลองสัญญาณทดสอบที่ใช้ในโปรแกรม Matlab	64
รูปที่ 4.8 ตัวอย่างผลการจำลองสัญญาณทดสอบโดยใช้ค่าจากตารางที่ 4.1	64
รูปที่ 4.9 สัญญาณมอเตอร์	65
รูปที่ 4.10 สัญญาณปั๊ม	65
รูปที่ 4.11 สัญญาณชุดเกียร์	65
รูปที่ 4.12 เครื่องคอมพิวเตอร์แบบ 1 ลูกสูบ และการติดตั้งระบบการวัด	67
รูปที่ 4.13 ระบบการวัดบนคอมพิวเตอร์แบบ 1 ลูกสูบ	68
รูปที่ 4.14 เครื่องคอมพิวเตอร์แบบ 2 ลูกสูบ และการติดตั้งระบบการวัด	68
รูปที่ 4.15 ระบบการวัดบนคอมพิวเตอร์แบบ 1 ลูกสูบ	69
รูปที่ 4.16 การตัดสัญญาณ X_S ในบริเวณที่มีแอมพลิจูดน้อยกว่าที่กำหนดก่อนทำการเปรียบเทียบ	72
รูปที่ 4.17 การตัดสัญญาณ Z_S ในบริเวณที่มีแอมพลิจูดน้อยกว่าที่กำหนดก่อนทำการเปรียบเทียบ	72
รูปที่ 4.18 การตรวจสอบตำแหน่งความถี่ระหว่างสัญญาณ X_S (รูปด้านบน) และ สัญญาณ Z_S (รูปด้านล่าง)	73
รูปที่ 4.19 ลักษณะของสัญญาณคืนกลับที่ได้รับการประเมินเป็น สีเขียว	74
รูปที่ 4.20 ลักษณะของสัญญาณคืนกลับที่ได้รับการประเมินเป็น สีส้ม	75
รูปที่ 4.21 ลักษณะของสัญญาณคืนกลับที่ได้รับการประเมินเป็น สีแดง	75

รูปที่ 5.18 การวัดสัญญาณเครื่องคอมพิวเตอร์แบบ 2 ลูกสูบ ครั้งที่ 3 ด้วยการใช้คอมพิวเตอร์ซึ่งร่วมกับการใช้ข้อมูลการวัดแบบสุ่มเทียบเท่าการใช้เซ็นเซอร์อัตราสุ่มจุด 640 เฮิร์ตซ์	91
รูปที่ 5.19 การวัดสัญญาณเครื่องคอมพิวเตอร์แบบ 2 ลูกสูบ ครั้งที่ 1 ด้วยการใช้คอมพิวเตอร์ซึ่งร่วมกับการใช้ข้อมูลการวัดแบบสุ่มเทียบเท่าการใช้เซ็นเซอร์อัตราสุ่มจุด 640 เฮิร์ตซ์ความยาวชุดข้อมูล 1 วินาที	92
รูปที่ 5.20 การวัดสัญญาณเครื่องคอมพิวเตอร์แบบ 2 ลูกสูบ ครั้งที่ 1 ด้วยการใช้คอมพิวเตอร์ซึ่งร่วมกับการใช้ข้อมูลการวัดแบบสุ่มเทียบเท่าการใช้เซ็นเซอร์อัตราสุ่มจุด 640 เฮิร์ตซ์ความยาวชุดข้อมูล 3 วินาที	92
รูปที่ 5.21 การวัดสัญญาณเครื่องคอมพิวเตอร์แบบ 2 ลูกสูบ ครั้งที่ 1 ด้วยการใช้คอมพิวเตอร์ซึ่งร่วมกับการใช้ข้อมูลการวัดแบบสุ่มเทียบเท่าการใช้เซ็นเซอร์อัตราสุ่มจุด 640 เฮิร์ตซ์ความยาวชุดข้อมูล 5 วินาที	93
รูปที่ 5.22 เปรียบเทียบการสุ่มหยิบสัญญาณเครื่องคอมพิวเตอร์แบบ 2 ลูกสูบ และผลการสร้างสัญญาณคืนกลับที่ใช้ y_t ที่มีความถี่สุ่มเทียบเท่า 640 เฮิร์ตซ์โดยความยาวข้อมูลที่ใช้ไม่เท่ากัน ก.) สัญญาณต้นแบบ ข.) y_t ความยาว 1 วินาที ค.) y_t ความยาว 3 วินาที ง.) y_t ความยาว 5 วินาที	94
รูปที่ 5.23 การวัดสัญญาณเครื่องคอมพิวเตอร์แบบ 2 ลูกสูบ ข้อมูลการวัดแบบสุ่มเทียบเท่าการใช้เซ็นเซอร์อัตราสุ่มจุด 1600 เฮิร์ตซ์ความยาวชุดข้อมูล 1 วินาที จำนวนจุดข้อมูล 1335 จุด	96
รูปที่ 5.24 การวัดสัญญาณเครื่องคอมพิวเตอร์แบบ 2 ลูกสูบ ข้อมูลการวัดแบบสุ่มเทียบเท่าการใช้เซ็นเซอร์อัตราสุ่มจุด 800 เฮิร์ตซ์ความยาวชุดข้อมูล 2 วินาที จำนวนจุดข้อมูล 1381 จุด	96
รูปที่ 5.25 เปรียบเทียบการสุ่มหยิบสัญญาณเครื่องคอมพิวเตอร์แบบ 2 ลูกสูบ และผลการสร้างสัญญาณคืนกลับที่ใช้ y_t ที่มีความยาวข้อมูลประมาณ 1300 จุด ความถี่สุ่มเทียบเท่าแบบต่างๆ ก.) สัญญาณต้นแบบ ข.) 1600 เฮิร์ตซ์ ค.) 800 เฮิร์ตซ์	97
รูปที่ 5.26 ตัวอย่างผลการทดสอบการการรสร้างสัญญาณกลับคืนจากการวัดด้วยการตั้งค่าแบบที่ 1 วัดสัญญาณ 800 เฮิร์ตซ์ แบบมอดูเลชั่น	99
รูปที่ 5.27 ตัวอย่างผลการทดสอบการการรสร้างสัญญาณกลับคืนจากการวัดด้วยการตั้งค่าแบบที่ 4 วัดสัญญาณ 800 เฮิร์ตซ์	100
รูปที่ 5.28 ตัวอย่างผลการทดสอบการการรสร้างสัญญาณกลับคืนจากการวัดด้วยการตั้งค่าแบบที่ 6 วัดสัญญาณ 500 เฮิร์ตซ์ แบบมอดูเลชั่น	100
รูปที่ 5.29 ตัวอย่างผลการทดสอบการการรสร้างสัญญาณกลับคืนจากการวัดด้วยการตั้งค่าแบบที่ 8 วัดสัญญาณ 1200 เฮิร์ตซ์ แบบมอดูเลชั่น	101
รูปที่ 5.30 ตัวอย่างผลการทดสอบการการรสร้างสัญญาณกลับคืนจากการวัดด้วยการตั้งค่าแบบที่ 1 คอมพิวเตอร์แบบลูกสูบ 1 ลูกสูบ	103
รูปที่ 5.31 ตัวอย่างผลการทดสอบการการรสร้างสัญญาณกลับคืนจากการวัดด้วยการตั้งค่าแบบที่ 2 คอมพิวเตอร์แบบลูกสูบ 1 ลูกสูบ	103

รูปที่ 5.47 เปรียบเทียบผลการสร้างสัญญาณคืนกลับของคอมเพรชเซอร์แบบ 1 ลูกสูบด้วยการปรับตั้งแบบที่ 4 ใน ความยาวของสัญญาณวัดแบบสุ่ม y_t ความยาวต่างๆ ก.) สัญญาณต้นแบบ ข.) 1 วินาที ค.) 4 วินาที	113
รูปที่ 5.48 ผลการสร้างสัญญาณคืนกลับของคอมเพรชเซอร์แบบ 2 ลูกสูบด้วยการปรับตั้งแบบที่ 8 ที่ใช้ความยาว ของสัญญาณวัดแบบสุ่ม y_t 10 วินาที	114
รูปที่ 5.49 เปรียบเทียบผลการสร้างสัญญาณคืนกลับของคอมเพรชเซอร์แบบ 2 ลูกสูบด้วยการปรับตั้งแบบที่ 8 ใน ความยาวของสัญญาณวัดแบบสุ่ม y_t ความยาวต่างๆ ก.) สัญญาณต้นแบบ ข.) 1 วินาที ค.) 10 วินาที	115
รูปที่ 5.50 ผลการสร้างสัญญาณคืนกลับของคอมเพรชเซอร์แบบ 1 ลูกสูบด้วยการปรับตั้งแบบที่ 4 โดยใช้การจับคู่ สัญญาณปีบอัด ที่ค่า K เท่ากับ 6	116
รูปที่ 5.51 เปรียบเทียบผลการสร้างสัญญาณคืนกลับของคอมเพรชเซอร์แบบ 1 ลูกสูบด้วยการปรับตั้งแบบที่ 4 โดย ใช้การจับคู่สัญญาณปีบอัดที่ค่า K เท่ากับ 6	117
รูปที่ 5.52 สัญญาณตัวแทนและระดับ เอร์เซ็นไทล์ที่เท่ากับค่า K ที่เลือก	117
รูปที่ 5.53 ผลการสร้างสัญญาณคืนกลับของคอมเพรชเซอร์แบบ 2 ลูกสูบด้วยการปรับตั้งแบบที่ 8 โดยใช้การจับคู่ สัญญาณปีบอัด ที่ค่า K เท่ากับ 15	118
รูปที่ 5.54 เปรียบเทียบผลการสร้างสัญญาณคืนกลับของคอมเพรชเซอร์แบบ 2 ลูกสูบด้วยการปรับตั้งแบบที่ 8 โดย ใช้การประมวลผลสัญญาณปีบอัด ที่ค่า K เท่ากับ 15	118
รูปที่ 5.55 สัญญาณตัวแทนและระดับ เอร์เซ็นไทล์ที่เท่ากับค่า K ที่เลือก	119
รูปที่ 5.56 ผลการสร้างสัญญาณคืนกลับของคอมเพรชเซอร์แบบ 1 ลูกสูบด้วยการปรับตั้งแบบที่ 4 โดยใช้การซ้อน สัญญาณเฉลี่ย ที่ค่า K เท่ากับ 12	120
รูปที่ 5.57 สัญญาณคืนกลับของคอมเพรชเซอร์แบบ 1 ลูกสูบ จาก ก.สัญญาณต้นแบบ ข.การจับคู่สัญญาณปีบอัด ค.การซ้อนสัญญาณเฉลี่ย	121
รูปที่ 5.58 สัญญาณตัวแทนและระดับเอร์เซ็นไทล์ที่เท่ากับค่า K ที่เลือก	121
รูปที่ 5.59 เปรียบเทียบระดับการเลือกความถี่ที่เหมาะสมระหว่าง การจับคู่สัญญาณปีบอัด และ การซ้อนสัญญาณ เฉลี่ย ของคอมเพรชเซอร์ 1 ลูกสูบ	122
รูปที่ 5.60 ผลการสร้างสัญญาณคืนกลับของคอมเพรชเซอร์แบบ 2 ลูกสูบด้วยการปรับตั้งแบบที่ 8 โดยใช้การซ้อน สัญญาณเฉลี่ย ที่ค่า K เท่ากับ 35	122
รูปที่ 5.61 สัญญาณคืนกลับของคอมเพรชเซอร์แบบ 2 ลูกสูบ จาก ก.สัญญาณต้นแบบ ข.การจับคู่สัญญาณปีบอัด ค.การซ้อนสัญญาณเฉลี่ย	123
รูปที่ 5.62 สัญญาณตัวแทนและระดับเอร์เซ็นไทล์ที่เท่ากับค่า K ที่เลือก	123
รูปที่ 5.63 เปรียบเทียบระดับการเลือกความถี่ที่เหมาะสมระหว่าง การจับคู่สัญญาณปีบอัด และ การซ้อนสัญญาณ เฉลี่ย ของคอมเพรชเซอร์ 2 ลูกสูบ	124

รูปที่ 5.64 สัญญาณคืนกลับของคอมเพรเซเซอร์แบบ 2 ลูกสูบ จาก ก.สัญญาณต้นแบบ ข.การจับคู่สัญญาณบีบอัด ค.การซ้อนสัญญาณเฉลี่ย ค.การหาค่าเหมาะสมที่สุดแบบคอนเวกซ์	125
รูปที่ 5.65 เปรียบเทียบการเลือกใช้ค่า K ต่างๆในสัญญาณตัวแทน ก.)สัญญาณต้นแบบ ข.) $K = 20$ ค.) $K = 34$ ง.) $K = 45$	127
รูปที่ ก.1 แผนผังระบบการรวบรวมและวิเคราะห์ข้อมูล (Data acquisition)สำหรับเซ็นเซอร์แบบวงจรรวม	136
รูปที่ ก.2 แผนผังระบบการรวบรวมและวิเคราะห์ข้อมูล (Data acquisition)สำหรับเซ็นเซอร์แบบเครื่องกลไฟฟ้า จุลภาค	136
รูปที่ ก.3 แสดงทิศทางของการวัดในตำแหน่งต่างเทียบกับเซ็นเซอร์	137
รูปที่ ก.4 แสดงลักษณะของ ตัววัดความเร่งแบบคาปาซิทีฟ	137
รูปที่ ก.5 แสดงแผนผังการทำงานภายใน ตัววัดความเร่ง ADXL 345	138
รูปที่ ก.6 เซ็นเซอร์ ADXL345	139
รูปที่ ก.7 การทดสอบหาลักษณะของวงจรองความถี่ต่ำ	140
รูปที่ ก.8 ค่าเฉลี่ยกำลังสองของแอมพลิจูดบนโดเมนเวลาเมื่อวัดสัญญาณการสั่นสะเทือนที่ความถี่ต่างๆ ข.) อัตราส่วนของแอมพลิจูดระหว่างเซ็นเซอร์ ADXL และ PCB ที่ความถี่ต่างๆ	141
รูปที่ ก.9 แสดงการต่อสายระหว่างบอร์ดและเซ็นเซอร์ ทั้ง 4-wire SPI และไฟเลี้ยงวงจร	142
รูปที่ ก.10 บอร์ดไมโครคอนโทรลเลอร์ STM32F411RE	143
รูปที่ ก.11 เซ็นเซอร์ PCB TLD352A56	143
รูปที่ ก.12 Data Acquisition รุ่น Dewesoft Sirius	144
รูปที่ ก.13 bruel & kjaer Type 4809	144
รูปที่ ก.14 GWinstek GFG-3015	145
รูปที่ ก.15 bruel & kjaer Type 2706	145
รูปที่ ก.16 Agilent DSO3102A	146
รูปที่ ก.17 คอมเพรเซเซอร์แบบ 1 ลูกสูบ	146
รูปที่ ก.18 คอมเพรเซเซอร์แบบ 2 ลูกสูบ	147
รูปที่ 1.ข แสดงแผนผังการทำงานของอัลกอริทึม	148
รูปที่ ค.1 คำอธิบายการควบคุมพริสเกลเลอร์รีจิสเตอร์จากหนังสือคู่มือของบอร์ดไมโครคอนโทรลเลอร์ STM32F411RE	155
รูปที่ ค.2 แผนผังการอินเตอร์รัพท์	157
รูปที่ ค.3 แผนผังการอินเตอร์รัพท์การวัดสัญญาณแบบสุ่มซึ่งทำการอัปเดตค่าอตรีโพลด์ทุกครั้ง	159
รูปที่ ง.1 แสดงหน้าต่างที่ใช้ทำงานของโปรแกรม STM32CUDEIDE	162
รูปที่ ง.2 แสดงโครงสร้างของชุดโปรแกรมในบอร์ดไมโครคอนโทรลเลอร์	163

รูปที่ จ.1 หน้าต่างโปรแกรมเบื้องต้น	173
รูปที่ จ.2 หน้าต่างโปรแกรมขณะทำการบันทึกสัญญาณ	173
รูปที่ จ.3 หน้าต่างโปรแกรมในหัวข้อ Analyse	174
รูปที่ จ.4 หน้าต่างข้อมูลที่อยู่ในไฟล์ Excel	174
รูปที่ จ.5 หน้าต่างโปรแกรม Matlab ที่ใช้บรรจุข้อมูลลงในตัวแปร	175
รูปที่ ฉ.1 หน้าต่างโปรแกรม RealTerm ตอนเริ่มเข้าโปรแกรม	177
รูปที่ ฉ.2 หน้าต่างโปรแกรม RealTerm เมื่อทำการวัด	178
รูปที่ ฉ.3 ตัวอย่างข้อมูลที่ทำการบันทึกจาก RealTerm	178
รูปที่ ฉ.4 แสดงข้อมูลจุดแรกที่ผิดพลาดซึ่งมี 5 คอลัมน์ซึ่งปกติจะมีเพียง 3 คอลัมน์	180
รูปที่ ช.1 ผลลัพธ์ของสัญญาณมอดูเลเตอร์ ครั้งที่ 1	181
รูปที่ ช.2 ผลลัพธ์ของสัญญาณมอดูเลเตอร์ ครั้งที่ 2	181
รูปที่ ช.3 ผลลัพธ์ของสัญญาณมอดูเลเตอร์ ครั้งที่ 3	182
รูปที่ ช.4 ผลลัพธ์ของสัญญาณมอดูเลเตอร์ ครั้งที่ 4	182
รูปที่ ช.5 ผลลัพธ์ของสัญญาณมอดูเลเตอร์ ครั้งที่ 5	183
รูปที่ ช.6 ผลลัพธ์ของสัญญาณบีบ ครั้งที่ 1	183
รูปที่ ช.7 ผลลัพธ์ของสัญญาณบีบ ครั้งที่ 2	184
รูปที่ ช.8 ผลลัพธ์ของสัญญาณบีบ ครั้งที่ 3	184
รูปที่ ช.9 ผลลัพธ์ของสัญญาณบีบ ครั้งที่ 4	185
รูปที่ ช.10 ผลลัพธ์ของสัญญาณบีบ ครั้งที่ 5	185
รูปที่ ช.11 ผลลัพธ์ของสัญญาณซูดเกียร์ ครั้งที่ 1	186
รูปที่ ช.12 ผลลัพธ์ของสัญญาณซูดเกียร์ ครั้งที่ 2	186
รูปที่ ช.13 ผลลัพธ์ของสัญญาณซูดเกียร์ ครั้งที่ 3	187
รูปที่ ช.14 ผลลัพธ์ของสัญญาณซูดเกียร์ ครั้งที่ 4	187
รูปที่ ช.15 ผลลัพธ์ของสัญญาณซูดเกียร์ ครั้งที่ 5	188
รูปที่ ซ.1 ผลลัพธ์ของเครื่องสร้างการสั่นสะเทือนที่สัญญาณความถี่ 300 เฮิรตซ์	189
รูปที่ ซ.2 ผลลัพธ์ของเครื่องสร้างการสั่นสะเทือนที่สัญญาณความถี่ 500 เฮิรตซ์	189
รูปที่ ซ.3 ผลลัพธ์ของเครื่องสร้างการสั่นสะเทือนที่สัญญาณความถี่ 800 เฮิรตซ์	190
รูปที่ ซ.4 ผลลัพธ์ของเครื่องสร้างการสั่นสะเทือนที่สัญญาณความถี่ 1000 เฮิรตซ์	190
รูปที่ ซ.5 ผลลัพธ์ของเครื่องสร้างการสั่นสะเทือนที่สัญญาณความถี่ 1200 เฮิรตซ์	191
รูปที่ ซ.6 ผลลัพธ์ของเครื่องสร้างการสั่นสะเทือนที่สัญญาณความถี่ 1500 เฮิรตซ์	191

รูปที่ ณ.99 การปรับตั้งแบบที่ 8 ที่สัญญาณความถี่ 300 เฮิร์ตซ์	252
รูปที่ ณ.100 การปรับตั้งแบบที่ 8 ที่สัญญาณความถี่ 500 เฮิร์ตซ์	253
รูปที่ ณ.101 การปรับตั้งแบบที่ 8 ที่สัญญาณความถี่ 800 เฮิร์ตซ์	253
รูปที่ ณ.102 การปรับตั้งแบบที่ 8 ที่สัญญาณความถี่ 1000 เฮิร์ตซ์	254
รูปที่ ณ.103 การปรับตั้งแบบที่ 8 ที่สัญญาณความถี่ 1200 เฮิร์ตซ์	254
รูปที่ ณ.104 การปรับตั้งแบบที่ 8 ที่สัญญาณความถี่ 1500 เฮิร์ตซ์	255
รูปที่ ณ.105 การปรับตั้งแบบที่ 8 ที่สัญญาณความถี่ 1800 เฮิร์ตซ์	255
รูปที่ ณ.106 การปรับตั้งแบบที่ 8 ที่สัญญาณความถี่ 300 เฮิร์ตซ์ มอดูเลชั่น	256
รูปที่ ณ.107 การปรับตั้งแบบที่ 8 ที่สัญญาณความถี่ 500 เฮิร์ตซ์ มอดูเลชั่น	256
รูปที่ ณ.108 การปรับตั้งแบบที่ 8 ที่สัญญาณความถี่ 800 เฮิร์ตซ์ มอดูเลชั่น	257
รูปที่ ณ.109 การปรับตั้งแบบที่ 8 ที่สัญญาณความถี่ 1000 เฮิร์ตซ์ มอดูเลชั่น	257
รูปที่ ณ.110 การปรับตั้งแบบที่ 8 ที่สัญญาณความถี่ 1200 เฮิร์ตซ์ มอดูเลชั่น	258
รูปที่ ณ.111 การปรับตั้งแบบที่ 8 ที่สัญญาณความถี่ 1500 เฮิร์ตซ์ มอดูเลชั่น	258
รูปที่ ณ.112 การปรับตั้งแบบที่ 8 ที่สัญญาณความถี่ 1800 เฮิร์ตซ์ มอดูเลชั่น	259

สารบัญตาราง

	หน้า
ตารางที่ 4.1 ตัวอย่างการกำหนดค่าของสัญญาณทดสอบ	64
ตารางที่ 4.2 รูปแบบของสัญญาณที่ใช้ทดสอบทั้งหมด จากเครื่องกำเนิดสัญญาณ	64
ตารางที่ 4.3 ตารางเงื่อนไขการทดสอบหาความเป็นไปได้ในการใช้คอมพิวเตอร์เซมซิงในเชิงหลักการ (Prove of concept) ของสัญญาณแต่ละแบบ	71
ตารางที่ 4.4 ตารางเงื่อนไขการทดสอบหาความเป็นไปได้ในการใช้คอมพิวเตอร์เซมซิงในเชิงปฏิบัติ (Proof of work) ของสัญญาณแต่ละแบบ	71
ตารางที่ 4.5 แสดงค่าพารามิเตอร์ของสัญญาณตัวอย่างในรูปที่ 4.16, 4.17, 4.18	73
ตารางที่ 5.1 ค่าวิเคราะห์ในโดเมนความถี่	82
ตารางที่ 5.2 ผลการทดสอบ ที่ใช้การสุ่มสัญญาณที่เทียบเท่ากับเซ็นเซอร์ 1600 เฮิร์ตซ์	83
ตารางที่ 5.3 ผลการทดสอบ ที่ใช้การสุ่มสัญญาณที่เทียบเท่ากับเซ็นเซอร์ 640 เฮิร์ตซ์	84
ตารางที่ 5.5 แสดงการปรับตั้ง ระบบการวัดแบบต่างๆที่ใช้ในการทดสอบ	90
ตารางที่ 5.6 แสดงการผลลัพธ์ของสัญญาณคืนกลับที่ถูกวัดด้วยการปรับตั้งแบบต่างๆ	102
ตารางที่ 5.7 แสดงการผลลัพธ์ของสัญญาณคืนกลับที่ถูกวัดด้วยการปรับตั้งแบบต่างๆ	112
ตารางที่ ก.1 คุณสมบัติของตัววัดความเร่ง ADXL 345	139
ตารางที่ ก.2 คุณสมบัติของไมโครคอนโทรลเลอร์ STM32F411RE	142
ตารางที่ ค.1 เปรียบเทียบพิสัยของความถี่สุ่มในกรณีความกว้างพัลส์ ระยะต่างๆ	158
ตารางที่ ค.2 เปรียบเทียบพิสัยของความถี่สุ่มในกรณีเพิ่มจำนวนพจน์ของ ฟังก์ชัน Random()	159
ตารางที่ ค.3 ตัวอย่างข้อมูลที่ถูกรับเข้ามายังคอมพิวเตอร์ก่อน และหลังทำการแปลงด้วยโปรแกรม MATLAB	161
ตารางที่ จ.1 แสดงข้อมูลเวลาที่มีความผิดพลาดเทียบกับข้อมูลที่แก้ไขแล้วสำหรับความถี่สุ่มสัญญาณ 20000 เฮิร์ตซ์	176
ตารางที่ ฉ.1 ตัวอย่างข้อมูลที่ error	179
ตารางที่ ฉ.2 ตัวอย่างข้อมูลที่ถูกแก้ไขแล้ว	180

คำอธิบายสัญลักษณ์และคำย่อ

x_s	คือ แอมพลิจูดของสัญญาณที่ถูกวัดแบบปกติในโดเมนความถี่
x_t	คือ แอมพลิจูดของสัญญาณที่ถูกวัดแบบปกติในโดเมนเวลา
t_x	คือ เวลาของข้อมูลแต่ละจุดของสัญญาณ x_t ลักษณะเป็นคาบเท่าๆ กันมีจำนวน N จุด
y_t	คือ แอมพลิจูดของสัญญาณที่ถูกวัดอัตราการสุ่มจุดไม่แน่นอนในโดเมนเวลา
t_y	คือ เวลาของข้อมูลแต่ละจุดของสัญญาณ x_t ลักษณะเป็นคาบเท่าๆ กันมีจำนวน M จุด โดยที่ $M < N$
ψ	คือ แมทริกซ์ฐานหลักทั่วไป (General Basis) ขนาด $[N \times N]$
C	คือ แมทริกซ์การวัด (Measurement Matrix) ที่ได้จากการจับคู่เวลา
θ	คือ แมทริกซ์ฐานหลักเริ่มต้น (Initial basis)
z_s	คือ แอมพลิจูดของสัญญาณคืนกลับ (Reconstruction Signal) โดเมนความถี่
z_t	คือ แอมพลิจูดของสัญญาณคืนกลับ (Reconstruction Signal) โดเมนเวลา
v_s	คือ สัญญาณตัวแทน (Proxy Signal) ในโดเมนความถี่
f_{rate}	คือ อัตราการสุ่มจุดข้อมูล (Sampling Rate)
f_{max}	คือ ความถี่สูงสุดที่ทำการวัด
f_{erate}	คือ ความถี่สุ่มเทียบเท่า
f_{range}	คือ พิสัยของความถี่สุ่ม
$N_{y,k}$	คือ จำนวนจุดข้อมูลทั้งหมดใน k วินาทีของสัญญาณ y_t
K	คือจำนวนของความถี่ที่มีค่าสูงสุดในสัญญาณตัวแทนที่ถูกนำไปคำนวณหาค่า z_s

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ในปัจจุบันแนวคิดเรื่องการบำรุงรักษาเครื่องจักรตามสภาพเป็นแนวคิดที่กำลังเริ่มได้รับความนิยมในประเทศไทยเนื่องจากเป็นวิธีที่มีประสิทธิภาพและเป็นการแก้ไขปัญหของเครื่องจักรได้ตรงจุดกว่าวิธีอื่นๆ การบำรุงรักษาเครื่องจักรตามสภาพนั้นมีความจำเป็นที่จะต้องติดตามพารามิเตอร์ต่างๆ ของเครื่องจักร เช่น อุณหภูมิ เสียง กลิ่น การสั่นสะเทือน เพื่อที่จะสามารถนำมาวิเคราะห์สภาพหรือสุขภาพของตัวเครื่องจักรนั้นได้ โดยการติดตามพารามิเตอร์ข้างต้นนั้นสามารถทำได้ตั้งแต่ง่ายที่สุดคือใช้ประสาทสัมผัสของผู้ปฏิบัติงานเอง ไปจนถึงการใช้อุปกรณ์หรือเครื่องมือพิเศษต่างๆ เข้ามาช่วยในการวัด ซึ่งแต่ละวิธีการก็จะมีคามยากง่ายและต้นทุนในการได้มาซึ่งข้อมูลสำหรับการวิเคราะห์ที่ไม่เท่ากัน ในการวิเคราะห์เครื่องจักรกลหมุนในปัจจุบันนั้น การวัดการสั่นสะเทือนของเครื่องจักรถือเป็นพารามิเตอร์ที่ถูกให้ความสำคัญมากเป็นอันดับต้นๆ เนื่องจากสามารถนำมาใช้วิเคราะห์ปัญหาของเครื่องจักรได้อย่างตรงจุด โดยเฉพาะการวิเคราะห์การสั่นสะเทือนโดยการตรวจสอบสเปกตรัมความถี่

ในการวัดการสั่นสะเทือนเพื่อให้ได้มาซึ่งสเปกตรัมความถี่ ที่มีรายละเอียดของสัญญาณครบถ้วนนั้น มีความจำเป็นที่จะต้องวัดสัญญาณโดยใช้ความถี่ในการเก็บข้อมูลสูงกว่าความถี่สูงสุดที่ต้องการวัด 2 เท่า หรือ อัตราไนควิสต์ (Nyquist rate) [1] เช่น ถ้าหากว่าต้องการวัดสัญญาณที่มีความถี่สูงสุดที่ 5000 เฮิร์ตซ์ จำเป็นจะต้องเก็บข้อมูลด้วยความถี่ 10000 เฮิร์ตซ์ เป็นอย่างน้อย ซึ่งถ้าหากว่าเก็บข้อมูลด้วยความถี่ที่ต่ำกว่านี้จะทำให้เกิด คลื่นแฝง (aliasing) ซึ่งเป็นปรากฏการณ์ที่สัญญาณความถี่สูงปรากฏให้เห็นในช่วงความถี่ต่ำทำให้เราไม่สามารถทราบได้ว่าสัญญาณที่ตรวจวัดได้นั้นเป็นสัญญาณจริงๆ หรือ สัญญาณหลอก [1] เพื่อหลีกเลี่ยงปรากฏการณ์ดังกล่าว จึงจำเป็นต้องใช้ระบบการวัดการสั่นสะเทือนที่มีความสามารถในการสุ่มตัวอย่างสูงกว่าความถี่สูงสุดของเครื่องจักรที่ต้องการจะวัดการสั่นสะเทือนอย่างน้อย 2 เท่า ปกติแล้วระบบที่มีความถี่ในการสุ่มสูงจะมีราคาแพง ส่งผลให้โรงงานอุตสาหกรรมที่ต้องการนำการวิเคราะห์การสั่นสะเทือนเข้ามาปรับใช้กับการบำรุงรักษาในโรงงานของตนเอง จำเป็นจะต้องลงทุนสูง

เนื่องจาก ระบบการวัดมีราคาสูงนี้เอง จึงมีความคิดที่จะใช้ระบบการวัดที่มีราคาไม่แพง ซึ่งมักจะมี ความถี่ในการสุ่มตัวอย่างต่ำ และเสถียรภาพของคาบเวลาแต่ละจุดที่ไม่แน่นอนมาใช้ในการวัดการสั่นสะเทือนของเครื่องจักร แต่ในการที่จะนำระบบการวัดที่มีความถี่ในการสุ่มต่ำกว่าปกติมาใช้เพื่อวัดสัญญาณนั้น หลีกเลียงไม่ได้ที่จะต้องนำกระบวนการต่างๆ เกี่ยวกับการประมวลผลสัญญาณเข้ามาช่วยเพื่อให้ได้ข้อมูลที่เพียงพอต่อการวิเคราะห์การสั่นสะเทือนซึ่งแนวคิดที่จะนำเสนอในงานวิจัยนี้ เรียกว่า คอมเพรสเซนซิง (Compressed sensing) [2]

แนวคิดการใช้คอมเพรสเซนซิง ถูกเสนอโดย David L Donoho [2] ในปี 2006 เป็นแนวคิดที่มีพื้นฐานมาจากการเก็บข้อมูลหรือสัญญาณให้เพียงพอและไม่มากเกินไปจนความจำเป็น [14] ด้วยการแก้ระบบสมการเชิงเส้นไม่ชัดเจน (Underdetermined linear system equation) เพื่อให้ได้ข้อมูลหรือสัญญาณที่ตีพอแก่การนำไปใช้งาน ซึ่งโดยปกติแล้ว ข้อมูลที่ถูกวัดจำนวนมากสามารถถูกตัดออกได้โดยที่ยังคงเนื้อหาหรือรายละเอียดหลักๆ

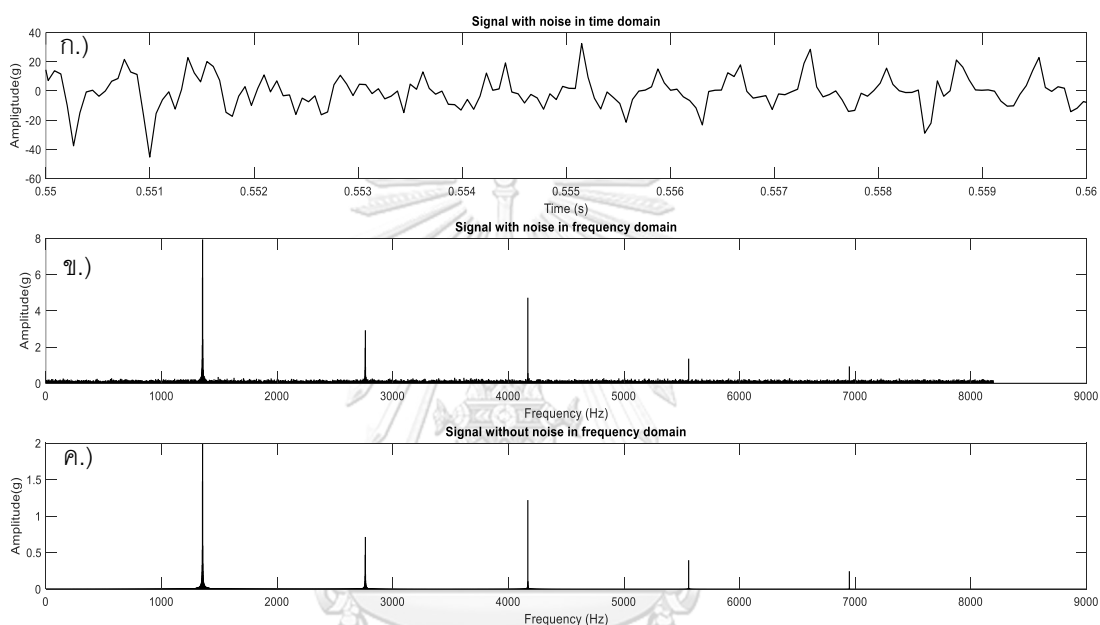
ของข้อมูลหรือสัญญาณไว้ได้ วิธีการนี้ถูกนำไปใช้กับสัญญาณและข้อมูลแบบต่างๆ เช่น การนำไปใช้สร้างสัญญาณภาพของผลการสแกนสมองด้วยเครื่องเอ็มอาร์ไอ (MRI Magnetic Resonance Imaging) [3] ระบบการจำแนกใบหน้าจากรูปภาพคุณภาพต่ำ [4] เป็นต้น การที่จะนำแนวคิดการคอมเพรสเซนซิงมาใช้ได้ดี สัญญาณที่จะนำมาประมวลผลจำเป็นจะต้องมีคุณสมบัติเป็นสเปิร์ส หรือหมายความว่า สัญญาณทั้งหมดที่ตรวจวัดมามีจำนวนข้อมูลที่มีความสำคัญอย่างแท้จริงหรือเพียงพอต่อการใช้งาน น้อยกว่าจำนวนข้อมูลที่ตรวจวัดได้ เช่น ถ้าหากต้องการจำแนกรูปภาพว่าเป็นรูปอะไร เราสามารถจำแนกจากรูปภาพของวัตถุ โดยใช้รูปภาพความละเอียดต่ำได้ ไม่จำเป็นต้องใช้ภาพความละเอียดสูงดังเช่นรูปที่ 1.1 ซึ่งสามารถทราบได้ทันทีว่า รูปภาพนั้นคือรูปของเสื้อ โดยถ้าหากเพิ่มความละเอียดของรูปภาพขึ้นไปอีกก็ยังคงทราบว่าเป็นเสื้ออยู่เช่นเดิม ฉะนั้นการถ่ายภาพที่ความละเอียดสูงที่ต้องการข้อมูลจำนวนมากๆ อาจจะไม่มีความจำเป็นสำหรับการจำแนกรูปภาพของวัตถุ แต่ถ้าหากจะนำรูปภาพดังกล่าวไประบุเกี่ยวกับรายละเอียดของตัวเสื้อเช่น ลวดลาย จำนวน กระเป๋ จำนวนกระดุมก็จำเป็นจะต้องใช้รูปภาพที่มีความละเอียดสูงกว่านี้

สำหรับสัญญาณ การสั้นสะเทือน ในโดเมนความถี่ จำนวนความถี่ที่สนใจก็มีจำนวนน้อยกว่าจำนวนของย่านความถี่ที่วัดได้ หมายความว่าในความเป็นจริงแล้วข้อมูลความถี่ทั้งหมดที่เก็บค่ามานั้นมีข้อมูลที่มีความจำเป็นจริงๆ อยู่บ้าง โดยดูได้จากตัวอย่างของสัญญาณการสั้นสะเทือนที่แสดงในรูปที่ 1.2 จากรูปจะเห็นได้ว่า ช่วงความถี่ที่วัดทั้งหมดนั้นมีตั้งแต่ 0 ถึง 4096 เฮิรตซ์ แต่ข้อมูลที่สำคัญซึ่งนำมาพิจารณาแล้วจะมีเพียงความถี่ที่มีแอมพลิจูดสูงอย่างมีนัยสำคัญเพียงไม่กี่ช่วงเท่านั้นและโดยปกติแถบสัญญาณรบกวน (Noise floor) หรือความถี่ที่ขนาดสัญญาณมีค่าน้อยมากๆ จะไม่ถูกนำมาพิจารณาหรือถูกตัดออกไปจนเหลือความถี่ที่เป็นตัวสัญญาณที่ปราศจากสัญญาณรบกวนในรูปที่ 1.2 (ค) สัญญาณที่มีคุณสมบัติชนิดนี้เรียกว่าสัญญาณสเปิร์ส (Sparse Signal) [15] และสัญญาณการสั้นสะเทือนของเครื่องจักรโดยปกติก็มีคุณสมบัติเป็นสัญญาณสเปิร์สในโดเมนความถี่

ในการประมวลผลสัญญาณจะมีกระบวนการหนึ่งๆที่เรียกว่าเทคนิคการลบทางสเปกตรัมหรือ spectral subtraction ซึ่งจะเป็นการตัดหรือลดทอนสัญญาณในความถี่ที่ต้องการซึ่งนิยมทำกันในกรณีที่ต้องการลดแถบสัญญาณรบกวน หรือบีบอัดสัญญาณให้มีขนาดเล็กลงในโดเมนความถี่ หลังจากนั้นจึงแปลงสัญญาณย้อนกลับมาเพื่อนำสัญญาณไปใช้งาน โดยสัญญาณที่แปลงกลับมาหลังจากกระบวนการบีบอัด หรือ ลบทางสเปกตรัมไปแล้วก็เป็นสัญญาณสเปิร์สเช่นกัน คอมเพรสเซนซิง (Compressed sensing) ที่จะนำเสนอในงานวิจัยนี้มีแนวคิดจากการวัดสัญญาณโดยมีจำนวนข้อมูลน้อยกว่าปกติ หรืออาจพิจารณาได้ว่าข้อมูลที่เก็บได้เป็นสัญญาณที่ถูกบีบอัดเรียบร้อยแล้ว และสัญญาณนี้จะถูกนำมาแปลงสัญญาณย้อนกลับเพื่อให้ได้สัญญาณสเปิร์สที่มีคุณภาพเพียงพอแก่การนำไปใช้งาน



รูปที่ 1.1 ตัวอย่างภาพของสื่อในความละเอียดต่ำ



รูปที่ 1.2 ตัวอย่างสัญญาณการสั่นสะเทือนของเครื่องจักร ก.) สัญญาณในโดเมนเวลา ข.) สัญญาณในโดเมนความถี่ ค.) สัญญาณในโดเมนความถี่ที่ทำการลดสัญญาณรบกวนออกแล้ว

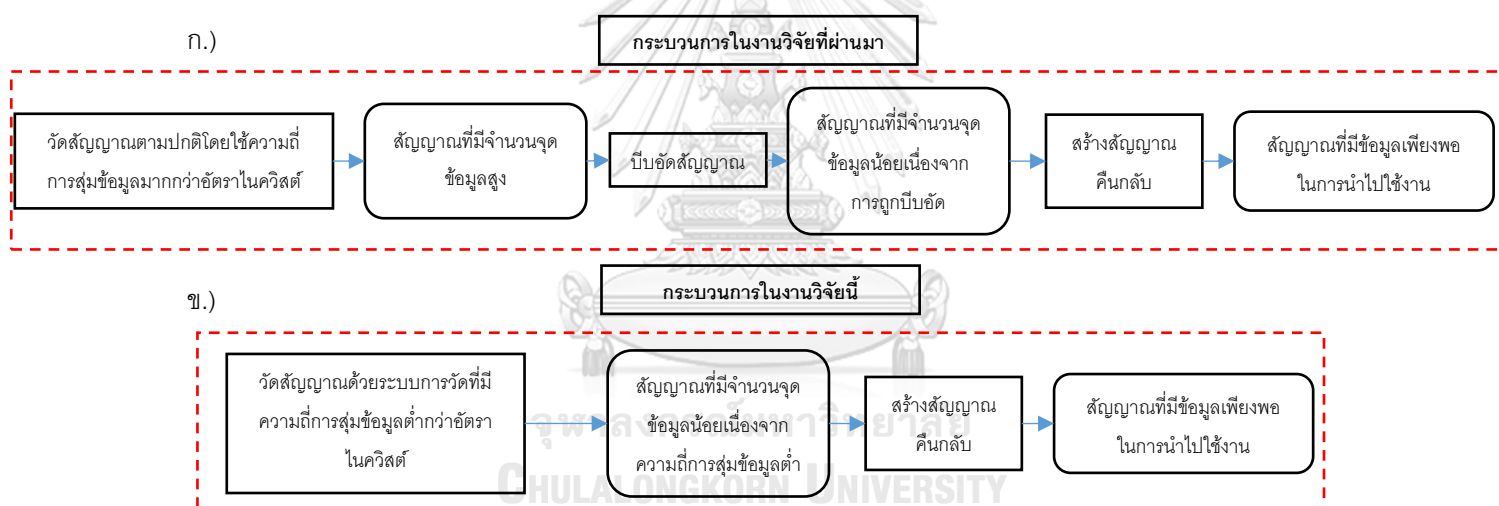
CHULALONGKORN UNIVERSITY

แนวคิด คอมเพรสชัน นั้น ได้ถูกนำไปประยุกต์ใช้ในหลายปัญหา เช่น การแพทย์ การประมวลผลภาพ วิดีโอแต่สำหรับการนำมาประยุกต์ใช้กับสัญญาณการสั่นสะเทือนของเครื่องจักรยังไม่แพร่หลายนัก โดยส่วนมากจะถูกใช้ในการจำแนกหรือระบุลักษณะความผิดปกติของเครื่องจักร เช่น การจำแนกลักษณะความเสียหายของของอุปกรณ์ต่างๆ เช่น ตลับลูกปืน [5,6,7] ลูกปืนในบีม์ลูกสูบ [12], หม้อแปลงไฟฟ้า [13], การหาความถี่ธรรมชาติของคานสั่นอิสระ [8], การลดสัญญาณรบกวน [9] และเพื่อใช้ลดพื้นที่การส่งข้อมูลและการจัดเก็บข้อมูล [10] สำหรับการลดพื้นที่การส่งและจัดเก็บข้อมูลในเอกสารอ้างอิง [10] นั้น ในงานนี้ จะวัดสัญญาณของหม้อเพลลา (Axle box) รถไฟด้วยเซ็นเซอร์ที่มีความถี่ในการสุ่มข้อมูลสูง ส่งผลให้จำนวนข้อมูลที่จำเป็นต้องถูกส่งผ่านและจัดเก็บในระบบการเก็บข้อมูลมีจำนวนมาก งานวิจัยนี้จึงได้บีบอัดสัญญาณให้มีขนาดเล็กลงเพื่อให้ประหยัดพื้นที่ในการส่งและจัดเก็บ หลังจากนั้นได้ใช้ คอมเพรสชันในการสร้างสัญญาณคืนกลับให้สมบูรณ์เมื่อต้องการนำไปใช้งาน ในงานวิจัยที่อ้างอิงถึงโดยส่วนใหญ่จะใช้คอมเพรสชันอัลกอริทึมแบบ L-

1 และ Compressed sampling matching pursuit Cosamp [11] เป็นหลัก และใช้เซ็นเซอร์ที่มีความถี่ในการสุ่มข้อมูลสูง

จากงานที่กล่าวมาทั้งหมดนั้น ส่วนมากมักมีจุดมุ่งหมายในการลดขนาดสัญญาณที่เก็บมาได้ เพื่อนำไป วิเคราะห์แยกแยะประเภท หรือลดขนาดสัญญาณเพื่อให้ประสิทธิภาพในการสื่อสารและส่งข้อมูลเพิ่มขึ้น ขั้นตอนการดำเนินการของงานเหล่านี้ถูกแสดงในรูปที่ 1.3 (ก) โดยเริ่มจากใช้ระบบการวัดสัญญาณโดยใช้ความถี่ในการสุ่มสัญญาณสูง ซึ่งทำให้ได้จุดข้อมูลจำนวนมากก่อน หลังจากนั้นจะลดจำนวนจุดข้อมูลลงโดยใช้แมทริกซ์แบบสุ่มแบบต่างๆ เช่น Gaussian random matrix ทำให้จำนวนจุดข้อมูลน้อยลง ก่อนที่จะนำไปใช้กับคอมพิวเตอร์ซึ่ง

สำหรับในงานวิจัยนี้จะใช้ระบบการวัดที่มีความถี่การสุ่มข้อมูลต่ำในการวัดสัญญาณโดยตรงจากเครื่องจักรทำให้ได้จำนวนจุดข้อมูลน้อยกว่าปกติและจำนวนจุดต่ำกว่าอัตราไนควิสต์ (Nyquist rate) หรือต่ำกว่า 2 เท่าของความถี่สูงสุดที่ต้องการวัด และนำสัญญาณที่ได้ไปใช้กับคอมพิวเตอร์ซึ่ง เพื่อให้ได้สัญญาณคืนกลับที่มีข้อมูลเพียงพอสำหรับการนำไปวิเคราะห์สภาพเครื่องจักรเพิ่มเติม ดังที่แสดงในรูปที่ 1.3 (ข)



รูปที่ 1.3 แผนผังเปรียบเทียบกระบวนการในการสร้างสัญญาณ ก.) งานวิจัยที่ผ่านมา ข.) งานวิจัยนี้

1.2 วัตถุประสงค์

ศึกษาความเป็นไปได้ที่จะนำเทคนิคคอมพิวเตอร์ซึ่ง มาใช้กับการวัดสัญญาณการสั่นสะเทือนซึ่งวัดโดยระบบการวัดที่มีความถี่สุ่มต่ำ และอัตราสุ่มจุดไม่แน่นอน เพื่อสร้างสัญญาณคืนกลับ (Reconstruction signal) ที่มีค่าความถี่สูงสุดที่วัดได้มากกว่าความถี่ไนควิสต์

1.3 ขอบเขตของวิทยานิพนธ์

1. วิธีการทดสอบ

1.1 ทดสอบโดยจำลองสัญญาณและจำลองลักษณะการสุ่มวัดด้วยโปรแกรม Matlab

1.2 ทดสอบโดยวัดสัญญาณการสั่นสะเทือนจริงซึ่งสร้างจากเครื่อง Shaker โดยใช้

1. ระบบวัดมาตรฐาน แต่วัดที่ความถี่สุ่มต่ำ
2. ระบบวัดราคาถูก ที่มีความถี่สุ่มต่ำและไม่สม่ำเสมอ

1.3 ทดสอบโดยวัดสัญญาณการสั่นสะเทือนของเครื่องจักรจริงโดยใช้

1. ระบบวัดมาตรฐาน แต่วัดที่ความถี่สุ่มต่ำ
2. ระบบวัดราคาถูก ที่มีความถี่สุ่มต่ำและไม่สม่ำเสมอ

2. วิธีการสร้างสัญญาณคืนกลับ

2.1. ทดสอบการสร้างสัญญาณคืนกลับ โดยใช้อัลกอริทึมการหาค่าเหมาะสมที่สุดแบบคอนเวกซ์ (Convex Optimixzation) เป็นหลัก และสำหรับกรณี 1.3 จะใช้อัลกอริทึมเพิ่มอีก 2 แบบรวมเป็น 3 แบบ คือ

1. อัลกอริทึมการหาค่าเหมาะสมที่สุดแบบคอนเวกซ์ (Convex Optimization)
2. อัลกอริทึมจับคู่สัญญาณบีบอัด (Compressed Sampling Matching Pursuit)
3. อัลกอริทึมซ้อนความถี่เฉลี่ย (Mean Stack Frequency)

3. ขอบเขตความถี่ที่สูงที่สุดของสัญญาณคืนกลับ หรือความถี่สูงสุดที่ต้องการทราบ

เนื่องจาก เซ็นเซอร์ ADXL345 มีวงจรรองความถี่ต่ำผ่าน ซึ่งมีความถี่คัทออฟ 1600 เฮิรตซ์ หมายความว่าจุดข้อมูลที่มีลักษณะของความถี่ที่สูงกว่า 1600 เฮิรตซ์ จะถูกลดทอนลงจนไม่สามารถทราบแอมพลิจูดที่มีความถี่มากกว่านั้นได้ ด้วยเหตุนี้ ส่งผลให้ในงานวิจัยนี้ไม่สามารถใช้คอมเพรสชันซิงในการสร้างสัญญาณคืนกลับ ที่มีความถี่มากกว่าความถี่คัทออฟได้ ทำให้ขอบเขตในการทดสอบสำหรับงานวิจัยนี้คือ 1600 เฮิรตซ์

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ทราบถึงความเป็นไปได้ที่จะนำคอมเพรสชันซิงมาใช้ร่วมกับระบบวัดการสั่นสะเทือนแบบสุ่มจุดข้อมูล

2. ทราบถึงข้อจำกัดของการใช้ระบบวัดการสั่นสะเทือนแบบสุ่มคาบด้วยคอมเพรสชันซิง

3. ได้ระบบวัดการสั่นสะเทือนแบบสุ่มคาบ ที่สามารถใช้อัตราการสุ่มจุดข้อมูลต่ำกว่าความถี่สูงสุดที่ต้องการจะวัด

บทที่ 2

ความรู้พื้นฐานและงานวิจัยที่เกี่ยวข้อง

หลักการโดยรวมของคอมเพล็กซ์เซชันคือการแก้ปัญหาาระบบสมการเชิงเส้นแบบไม่ชัดเจนด้วยวิธีต่างๆ เพื่อให้ได้คำตอบที่มีคุณภาพเพียงพอแก่การนำไปใช้งาน เช่น การแก้ระบบสมการโดยตรงด้วยการใช้การหาค่าเหมาะสมที่สุด (Optimization) หรือ จะใช้การเลือกสมการที่มีความสำคัญมากที่สุดไม่กี่ชุดในระบบสมการเชิงเส้นไม่ชัดเจนมาแก้สมการเพื่อให้ได้คำตอบแบบประมาณการ โดยโครงสร้างระบบสมการเชิงเส้นที่จะถูกนำมาแก้ปัญหานี้จะต้องมีการเปลี่ยนฐานหลัก (Change of basis) ให้อยู่ในโดเมนที่เหมาะสมก่อน สำหรับสัญญาณการสังเคราะห์จะเป็นการเปลี่ยนฐานหลักระหว่าง โดเมนความถี่ (X_S) และโดเมนเวลา (X_T) ดังนั้น ในการพัฒนาอัลกอริทึมด้วยคอมเพล็กซ์เซชัน จะต้องมีความรู้พื้นฐาน ได้แก่ การแก้ปัญหาระบบสมการเชิงเส้นและสัญญาณสปาร์ส การบีบอัดสัญญาณ การเปลี่ยนฐานหลัก การแปลงโคไซน์และการแปลงฟูเรียร์ ความรู้พื้นฐานเหล่านี้จะถูกพูดถึงในหัวข้อต่อไปในบทนี้

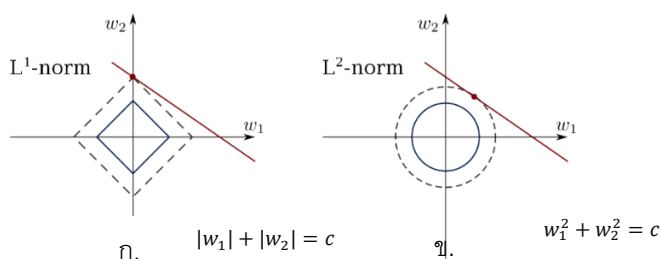
2.1 การแก้ปัญหาระบบสมการเชิงเส้นและสัญญาณสปาร์ส

2.1.1 ระบบสมการเชิงเส้นไม่ชัดเจน (Underdetermine system)

ระบบสมการเชิงเส้นไม่ชัดเจนคือระบบสมการที่มีจำนวนสมการน้อยกว่าตัวแปร ส่งผลให้คำตอบของตัวแปรที่ทำให้ระบบสมการเป็นจริงมีจำนวนเป็นอนันต์ หากกำหนดให้ระบบสมการเชิงเส้นมีทั้งหมด M สมการ และมีตัวแปร N ตัวโดยที่ $M < N$ โดยระบบสมการนี้จะสามารถเขียนให้อยู่ในรูปของแมทริกซ์ $As = y$ ได้ดังนี้

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1} & a_{M2} & \cdots & a_{MN} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_N \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix} \quad (2.1)$$

โดย y คือเวกเตอร์ค่าคงที่ ที่มีขนาด $M \times 1$, A คือแมทริกซ์สัมประสิทธิ์ของตัวแปร ที่มีขนาด $M \times N$ และ s คือเวกเตอร์ของตัวแปรไม่ทราบค่าที่มีขนาด $N \times 1$



รูปที่ 2.1 ตำแหน่งของคำตอบของระบบสมการเชิงเส้นบนระนาบ 2 มิติ ก.) บรรทัดฐาน 1 ข.) บรรทัดฐาน 2

รูปที่ 2.1 แสดงตัวอย่างของคำตอบของระบบสมการเชิงเส้นไม่ชัดเจนที่มี 1 สมการ แต่มีตัวแปร 2 ตัว จะเห็นได้ว่าชุดคำตอบที่เป็นไปได้ของระบบสมการนี้จะอยู่บนเส้นตรงสีแดงในรูป หากต้องการระบุคำตอบที่แน่นอนของระบบสมการนี้จำเป็นต้องกำหนดเป้าหมายของการแก้สมการเพิ่มเติม เช่น หากต้องการให้ผลรวมของค่าสัมบูรณ์ของคำตอบมีค่าน้อยที่สุด จะได้คำตอบของสมการเป็นจุดสีแดงในรูปที่ 2.1(ก.) โดยเส้นประซึ่งล้อมเป็นพื้นที่สี่เหลี่ยมจัตุรัสในรูปที่ 2.1(ก.) แสดงถึงเป้าหมายที่ต้องการ แต่หากต้องการให้ผลรวมกำลังสองของคำตอบมีค่าน้อยที่สุดจะได้คำตอบของสมการเป็นจุดสีแดงในรูป 2.1(ข.) โดยขอบเขตของเป้าหมายในกรณีนี้จะ มีลักษณะเป็นวงกลม ซึ่งสอดคล้องกับสมการกำลังสองน้อยที่สุด

รูปร่างเรขาคณิต (เส้นประ) ของเป้าหมาย สามารถเขียนให้อยู่ในรูปทั่วไปของฟังก์ชันบรรทัดฐาน n ดังนี้

$$\|S\|_n = \left(\sum_{i=1}^N |s_i|^n \right)^{\frac{1}{n}} \quad (2.2)$$

โดย s_i เป็นสมาชิกของตัวแปรลำดับที่ i ซึ่งมีทั้งหมด N ตัวแปร

เมื่อเทียบผลของสมการ (2.2) กับตัวอย่างก่อนหน้าในรูปที่ 2.1 แล้ว จะได้ว่า บรรทัดฐาน 2 ($n = 1$) จะหมายถึงคำตอบที่มีผลรวมของค่าสัมบูรณ์ของคำตอบน้อยที่สุด ซึ่งในกรณีนี้จะทำให้มีตัวแปรคำตอบที่เป็นศูนย์อยู่มากที่สุดด้วย จึงเรียกคำตอบในกรณีบรรทัดฐาน 1 นี้ว่าผลเฉลยแบบสปาร์ส (Sparse solution) [16] ส่วนบรรทัดฐาน 2 ($n = 2$) จะหมายถึงคำตอบที่มีผลรวมกำลังสองของคำตอบน้อยที่สุด หรือเรียกว่าผลเฉลยแบบกำลังสองน้อยที่สุด (Least square solution) สำหรับบรรทัดฐานอื่นๆ รูปร่างเรขาคณิตตามสมการ (2.2) จะเปลี่ยนไปตามค่าบรรทัดฐาน n ดังแสดงในรูปที่ 2.2



รูปที่ 2.2 การเปลี่ยนแปลงรูปร่างของคำตอบจากบรรทัดฐาน 0.25 ถึง อนันต์

สำหรับการแก้ปัญหาหระบบสมการเชิงเส้นไม่ชัดเจนโดยการเพิ่มเงื่อนไขเป้าหมายที่กล่าวไปข้างต้น เพื่อให้ได้คำตอบแบบสปาร์ส สามารถเขียนให้อยู่ในรูปสมการ ได้ดังนี้

$$\hat{s} = \underset{s}{\operatorname{argmin}} \|s\|_1 \text{ โดยที่ } \|As - y\|_2 = 0 \quad (2.3)$$

โดย \hat{s} หมายถึง คำตอบแบบสปาร์ส

$\underset{s}{\operatorname{argmin}} f(x)$ หรือ argument of the minimum หมายถึงการหาค่า s ที่ทำให้ฟังก์ชัน $f(x)$ มีค่าต่ำที่สุด

ตัวห้อย 2 ในสมการ $\|As - y\|_2 = 0$ หมายถึงบรรทัดฐาน 2 หรือเป็นการหาความผิดพลาดแบบกำลังสองน้อยที่สุด (least square error) ของ $As - y$ เมื่อกำหนดให้ $\|As - y\|_2 = 0$ จึงหมายถึง s ต้องเป็นคำตอบของระบบสมการเชิงเส้น $As = y$ และเป้าหมายจะแสดงด้วยเทอม $\underset{s}{\operatorname{argmin}} \|s\|_1$ ซึ่งหมายถึงการหาค่า $\|s\|_1$ ด้วยบรรทัดฐาน 1 เพื่อให้ได้คำตอบที่เป็นสปาร์สและมีผลรวมของค่าสัมบูรณ์ของคำตอบน้อยที่สุด สมการ (2.3) ไม่สามารถจะหาคำตอบได้โดยการแก้สมการแบบปกติ การแก้สมการ (2.3) จำเป็นต้องใช้วิธีการหาค่าเหมาะสมที่สุด (Optimization) เข้ามาช่วยในการแก้สมการ ซึ่งปัญหานี้สามารถใช้วิธีการหาค่าที่เหมาะสมที่สุดแบบคอนเวกซ์ (Convex Optimization) ได้ โดยสมการที่ใช้สำหรับการหาค่าเหมาะสมที่สุดแบบคอนเวกซ์ แสดงด้วยสมการ

$$\hat{s} = \underset{s}{\operatorname{argmin}} \|As - y\|_2 + \lambda \|s\|_1 \quad (2.4)$$

โดย λ คือ ค่าคงที่ปรับแก้ (Penalty term) เพื่อให้ \hat{s} มีการลู่อเข้าหาคำตอบที่ถูกต้อง

λ คือ ระดับของบรรทัดฐาน (Norm) เพื่อให้คำตอบ \hat{s} มีลักษณะแบบสปาร์ส

การแก้ระบบสมการเชิงเส้นไม่ชัดเจน (2.1) อาจสามารถทำได้โดยการคูณอินเวอร์สของแมทริกซ์ A ทั้งสองข้างของสมการแต่เนื่องจากขนาดของ A ไม่ใช่แมทริกซ์จัตุรัสแต่เท่ากับ $M \times N$ โดย $M < N$ ทำให้การหาอินเวอร์สของ A แบบปกติไม่สามารถทำได้ การหาอินเวอร์สในกรณีนี้ต้องใช้ มัวร์-เพนโรส อินเวอร์ส (Moore-Penrose inverse) ซึ่งก็จะให้คำตอบมีลักษณะความคลาดเคลื่อนแบบกำลังสองน้อยที่สุด (Least Square Solution) เช่นกัน แต่คำตอบที่ได้จากวิธีนี้จะไม่ใช่อินเวอร์สสปาร์ส อย่างไรก็ตามการหาคำตอบแบบนี้มี

ความสำคัญในการนำไปใช้สร้างสัญญาณตัวแทน โดยจะกล่าวอย่างละเอียดในหัวข้อ 3.2.2.1 และสำหรับวิธีการหา มัวร์-เพนโรส อินเวอร์ส (Moore-Penrose inverse) จะกล่าวถึงในหัวข้อ 2.1.2.1

2.1.2 ระบบสมการเชิงเส้นชัดเจนเกินไป (Overdetermined system)

ระบบสมการเชิงเส้นชัดเจนเกินไปคือระบบสมการที่มีจำนวนสมการมีมากกว่าจำนวนตัวแปร ส่งผลทำให้อาจจะไม่มีคำตอบของตัวแปรที่ทำให้ระบบสมการเป็นจริง หรือไม่มีคำตอบที่ชัดเจน (Exact solution) แต่สามารถหาคำตอบได้ในลักษณะของการประมาณโดยให้คำตอบที่เมื่อแทนลงไปในระบบสมการ จะทำให้ระบบสมการมีความคลาดเคลื่อนแบบผลรวมกำลังสองน้อยที่สุด (least square error) หากกำหนดให้ระบบสมการเชิงเส้นมีทั้งหมด M สมการและมีตัวแปร N โดยที่ $M > N$ ระบบสมการนี้จะสามารถเขียนให้อยู่ในรูป $As = y$ ได้ดังนี้

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1} & a_{M2} & \cdots & a_{MN} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_N \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_M \end{bmatrix} \quad (2.5)$$

โดย y คือแมทริกซ์ค่าคงที่ ที่มีขนาด $M \times 1$, A คือแมทริกซ์สัมประสิทธิ์ของตัวแปร ที่มีขนาด $M \times N$ และ s คือแมทริกซ์ของตัวแปรไม่ทราบค่า ที่มีขนาด $N \times 1$

สำหรับระบบสมการเชิงเส้นแบบปกติที่มีจำนวนสมการเท่ากับตัวแปร $As = y$ เราสามารถหาคำตอบได้โดยการใช้อินเวอร์สของแมทริกซ์ A คูณเข้าทั้งสองข้าง จะได้ $s = A^{-1}y$ แต่ในสมการที่ (2.5) เนื่องจาก A ไม่ได้เป็นแมทริกซ์จัตุรัสจึงไม่สามารถที่จะทำการอินเวอร์ส (matrix inversion) แบบปกติได้ วิธีหนึ่งในการทำอินเวอร์สแมทริกซ์ A คือการใช้ QR-factorization หรือใช้การแยกค่าเอกฐาน (Singular Value decomposition) หรือ SVD โดยแมทริกซ์ที่ได้จะถูกเรียกว่า ซูโดอินเวอร์ส (Pseudo Inverse) หรือ มัวร์-เพนโรส อินเวอร์ส (Moore-Penrose inverse) ซึ่งจะให้คำตอบมีลักษณะความคลาดเคลื่อนแบบกำลังสองน้อยที่สุด (Least Square Solution)

2.1.2.1 การแยกค่าเอกฐาน และ มัวร์-เพนโรส อินเวอร์ส (SVD and MoorePenrose inverse)

การแยกค่าเอกฐาน คือการแยกองค์ประกอบของแมทริกซ์ A ใดๆ ขนาด $[N \times M]$ ให้อยู่ในรูปแมทริกซ์ 3 ชุดคูณกันดังสมการ

$$A = U\Sigma V^T \quad (2.6)$$

โดย U ขนาด $[N \times N]$ เป็นเมทริกซ์ของเวกเตอร์เอกฐานซ้าย (Left singular vector matrix) Σ เป็นเมทริกซ์ของค่าเอกฐาน (singular value matrix) ขนาด $[N \times M]$ V^T เป็นเมทริกซ์ของเวกเตอร์เอกฐานขวา (Right singular vector matrix) ขนาด $[M \times M]$ ซึ่งสมาชิกแต่ละตัวจะมีขนาดและคุณสมบัติดังนี้

U และ V^T เป็นเมทริกซ์จัตุรัสและมีคุณสมบัติเป็นทั้งเมทริกซ์ยูนิเทรี (Unitary Matrix) และ Orthonormal matrix ซึ่งผลคูณระหว่างตัวเมทริกซ์เองกับทรานสโพสของตัวเอง (Transpose matrix) จะได้เมทริกซ์เอกลักษณ์ (Identity matrix) และทรานสโพสเมทริกซ์ในกรณีนี้ จะเป็นอินเวอร์สเมทริกซ์ด้วยดังสมการ

$$U^T U = U U^T = U U^{-1} = V^T V = V V^T = V V^{-1} = I \quad (2.7)$$

Σ เป็น Rectangular Diagonal matrix เก็บค่าเอกฐานจากมากที่สุดไปยังน้อยที่สุดโดย $\sigma_1 > \sigma_2 > \dots > \sigma_M$ ดังสมการที่ (2.8) สำหรับแถวที่ $M + 1$ เป็นต้นไปจนถึงแถวที่ N สมาชิกจะมีค่าเป็นศูนย์ทั้งหมด โดยที่ $N > M$

$$\Sigma = \begin{matrix} & & & & [N \times M] \\ \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \sigma_M \\ & & & & 0 \end{bmatrix} & & & & \end{matrix} \quad (2.8)$$

สำหรับเมทริกซ์ Σ การหาอินเวอร์สเมทริกซ์สามารถทำได้โดยการหาส่วนกลับของสมาชิกแต่ละตัวที่มีค่าไม่เท่ากับศูนย์แล้วนำไป ทรานสโพส ดังสมการ

$$\Sigma^\dagger = \begin{matrix} & & & & [M \times N] \\ \begin{bmatrix} \frac{1}{\sigma_1} & 0 & \dots & 0 \\ 0 & \frac{1}{\sigma_2} & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \frac{1}{\sigma_M} \end{bmatrix} & & & & \end{matrix} \text{ และ } \Sigma^\dagger \Sigma = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.9)$$

เมื่อนำ Σ^\dagger ไปคูณ กับ Σ ผลลัพธ์ที่ได้จะเป็นเมทริกซ์ทแยงมุมขนาด $[M \times M]$ มีค่าของสมาชิกในแนวทแยงเท่ากับ 1 จากที่กล่าวมาข้างต้น สมการ (2.6) สามารถเขียนในรูปเมทริกซ์ได้ดังนี้

$$\begin{bmatrix} N \times M \\ A \end{bmatrix} = \begin{bmatrix} N \times N \\ \vdots & \vdots & \vdots & \vdots \\ s_1 & s_2 & s_3 & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} N \times M \\ \sigma_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \sigma_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \sigma_M \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} M \times M \\ \vdots & v_1 & \cdots & \cdots \\ \vdots & v_2 & \cdots & \cdots \\ \vdots & \vdots & \cdots & \cdots \\ \vdots & v_M & \cdots & \cdots \end{bmatrix} \quad (2.10)$$

สำหรับการนำการแยกค่าเอกฐานมาใช้ทำการอินเวอร์สแมทริกซ์ที่ไม่จัตุรัส สามารถทำได้ดังนี้ จากสมการ (2.5) เขียน A ให้อยู่ในรูปการแยกเอกฐานจะได้

$$\begin{aligned} \mathbf{y} &= \mathbf{A}\mathbf{s} \\ \mathbf{y} &= \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{s} \end{aligned}$$

คูณทั้งสองข้างของสมการด้วย \mathbf{U}^T , $\mathbf{\Sigma}^\dagger$ และ \mathbf{V} ตามลำดับ จะได้

$$\mathbf{V}\mathbf{\Sigma}^\dagger\mathbf{U}^T\mathbf{y} = \mathbf{V}\mathbf{\Sigma}^\dagger\mathbf{U}^T\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{s} = \mathbf{s}$$

จากคุณสมบัติของ \mathbf{U} และ \mathbf{V} ที่เป็นแมทริกซ์ยูนิแทรี และ $\mathbf{\Sigma}$ เป็นแมทริกซ์ทแยงมุม หมายความว่าผลคูณระหว่างตัวแมทริกซ์เองกับ แมทริกซ์สลับเปลี่ยน หรือ อินเวอร์สแมทริกซ์ (Inverse Matrix) ของตัวเอง จะได้แมทริกซ์เอกลักษณ์ ส่งผลให้สมการทั้งหมดจะอยู่ในรูปสมการด้านล่าง

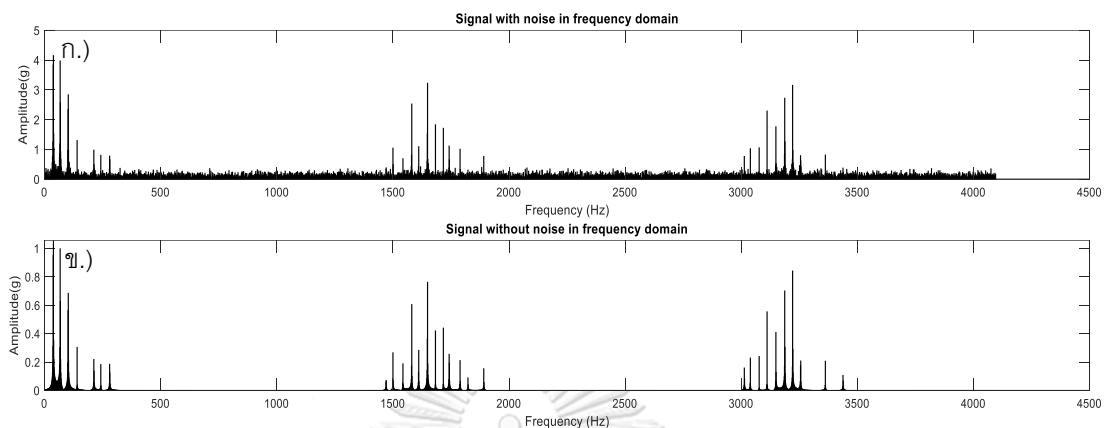
$$\mathbf{A}^\dagger\mathbf{y} = \mathbf{s} \text{ โดย } \mathbf{A}^\dagger = \mathbf{V}\mathbf{\Sigma}^\dagger\mathbf{U}^T$$

มัวร์-เพนโรส อินเวอร์ส (Moore-Penrose inverse) ในงานวิจัยนี้ถูกใช้ในการสร้างสัญญาณตัวแทน (Proxy signal) ซึ่งถือว่าการแปลงสัญญาณสู่ในโดเมนความถี่ซึ่งจะกล่าวถึงอย่างละเอียดในหัวข้อ 3.2.2.1

การลดมิติเนื่องจากเวกเตอร์ใน matrix เป็นศูนย์

จากสมการที่ (2.10) จะเห็นว่า $\mathbf{\Sigma}$ ในแถวที่มากกว่า M ขึ้นไปจะมีสมาชิกทุกตัวเป็นศูนย์ การที่สมาชิกทุกตัวในแถวหรือในหลักทุกตัวเป็นศูนย์ส่งผลให้ ผลลัพธ์จากการคูณกันของแถวหรือหลักของแมทริกซ์นี้กับแมทริกซ์อื่นที่นำมาคูณด้วยมีค่าเป็นศูนย์ด้วยเช่นกัน หมายความว่าเราสามารถที่จะลดขนาดของแมทริกซ์ในระบบสมการลงได้ให้เหลือเฉพาะ แถวหรือแนวในแมทริกซ์ที่ไม่เท่ากับ 0 ทั้งหมด ซึ่งใช้ได้กับแมทริกซ์ทุกประเภท ในที่นี้จะขอยกตัวอย่าง สมการ (2.10) เมื่อแถวที่มากกว่า M ขึ้นไปเป็นศูนย์หมายความว่าหลักที่มากกว่า M ของแมทริกซ์ \mathbf{S} ไม่ว่าจะป็นค่าอะไรก็ตามจะไม่ส่งผลต่อการคูณกับแมทริกซ์ $\mathbf{\Sigma}$ ด้วยเหตุนี้จึงสามารถตัดแถวที่เกินกว่า M ขึ้นไปของแมทริกซ์ $\mathbf{\Sigma}$ และหลักที่เกินกว่า M ขึ้นไปของแมทริกซ์ \mathbf{S} ได้ มิติของ $\mathbf{\Sigma}$ จึงลดลงเป็น $[M \times M]$ ส่วนมิติของ \mathbf{S} จะลดลงเป็น $[N \times M]$ โดยรูปสมการนี้จะถูกเรียกว่าเป็น Economy form ดังสมการที่ (2.11)

พลิจูดของสัญญาณในความถี่ที่เราไม่ต้องการ หรือตัดความถี่ที่มีขนาดไม่ถึงแอมพลิจูดที่กำหนด (Threshold Frequency) ให้เหลือศูนย์



รูปที่ 2.4 สัญญาณตัวอย่าง ก.) ก่อนการลดสัญญาณรบกวน ข.) หลังการลดสัญญาณรบกวน

รูปที่ 2.4(ก) แสดงตัวอย่างสัญญาณสั้นๆ ก่อนการบีบอัด จะเห็นว่าสัญญาณรบกวนที่มีแอมพลิจูดต่ำอยู่ตลอดย่านการวัดเมื่อสัญญาณถูกรีบอัดโดยตัดความถี่ที่มีแอมพลิจูดต่ำมากออก (Spectral subtraction) จะได้สัญญาณในรูปที่ 2.4(ข) จากรูปที่ 2.4(ข) สัญญาณที่ถูกบีบอัดแล้วจะเหลือเพียงความถี่ที่มีแอมพลิจูดสูงเท่านั้น ซึ่งเป็นส่วนที่เป็นเนื้อหาหรือข้อมูลจริงของสัญญาณที่มีประโยชน์

2.การบีบอัดสัญญาณภาพ

สำหรับการบีบอัดรูปภาพจะใช้ การแปลงโคไซน์ (Cosine Transform) เพื่อให้สัญญาณภาพอยู่ในโดเมนความถี่ที่สัญญาณมีความเป็นสปร้าสมมากขึ้น ซึ่งสามารถบีบอัดข้อมูลได้มากกว่าการแปลงฟูเรียร์ [10]

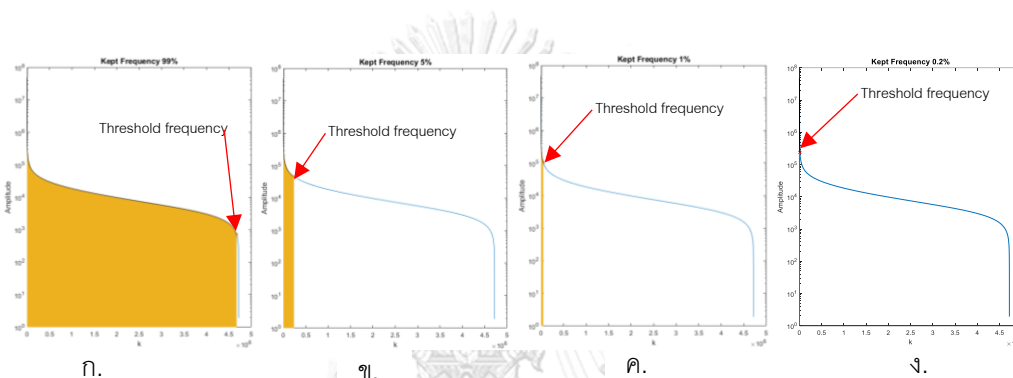


ก.

ข.

รูปที่ 2.5 สัญญาณภาพใน RGB layer ก.) รูปปกติ และ ข.) สัญญาณภาพหลังจากการแปลงโคไซน์

จากรูปที่ 2.5(ก). เป็นตัวอย่างของสัญญาณภาพปกติ เมื่อทำการแปลงโคไซน์แบบ 2 มิติแล้วจะมีลักษณะเป็นรูปที่ 2.5(ข). โดยแกนแกมมาสีทางขวาจะแสดงถึงขนาดของความถี่ ในแต่ละตำแหน่งของภาพ รูปที่ 2.6 เป็นรูปของการนำแอมพลิจูดของความถี่ต่างๆ มาจัดเรียงลำดับจากมากไปหาน้อยเพื่อเตรียมทำการบีบอัด (Spectral subtraction) กราฟความถี่ที่ถูกนำมาเรียงจากน้อยไปหามากจะถูกตัดออก โดยเหลือความถี่จากสูงสุดลดลงไปจนถึงความถี่ที่กำหนด (Threshold Frequency) เป็นพื้นที่ใต้กราฟสีเหลือง โดยในรูปจะทำการบีบและเก็บสัญญาณคงไว้ทั้งหมด 4 ระดับได้แก่ 99%,5%,1%,0.2% โดยจะแบ่งเป็นเปอร์เซ็นต์ตามจำนวนความถี่ที่ไม่เท่ากับศูนย์ที่เหลืออยู่หลังทำการบีบอัด แกนตั้งจะเป็นความถี่ของสัญญาณส่วนแกนนอนจะเป็นลำดับของ (k) ตำแหน่งที่มีความถี่มากที่สุดไปน้อยที่สุด โดยรูปที่ 2.6 และ 2.72 จะเป็นการแสดงตัวอย่างของการบีบอัดสัญญาณภาพ



รูปที่ 2.6 พื้นที่ใต้กราฟสีเหลืองแสดงถึงจำนวนขนาดข้อมูลความถี่ที่ถูกใช้ไปเพื่อการสร้างสัญญาณภาพที่ถูกบีบอัด โดยขนาดข้อมูลที่ใช้คือ ก.) 99% ข.) 5% ค.) 1% ง.) 0.2%



รูปที่ 2.7 สัญญาณภาพที่ถูกสร้างขึ้นตามขนาดข้อมูลความถี่จากรูปที่ 2.6
ก.) 99% ข.) 5% ค.) 1% ง.) 0.2%

ในรูปที่ 2.7(ง) ถึงแม้ว่าสัญญาณภาพจะถูกบีบอัดจนเหลือจำนวนสัญญาณที่ถูกนำไปใช้เพียง 0.2% ก็ยังพอทราบได้ว่าภาพนั้น เป็นภาพของเสื้อถึงแม้จะไม่เห็นรายละเอียด หากต้องการรายละเอียดมากขึ้นก็ต้องบีบอัดสัญญาณให้น้อยลง นอกจากนี้จากรูปที่ 2.6 และ รูปที่ 2.7 จะเห็นว่าตำแหน่งที่มีขนาดแอมพลิจูดสูงๆ จะเป็นตัวบ่งบอกถึงภาพรวมหรือโครงสร้างของภาพ ส่วนรายละเอียดของภาพก็จะเป็นส่วนที่มีขนาดต่ำกว่า การ

บีบอัดที่จะมากหรือน้อยนั้นขึ้นอยู่กับ การนำสัญญาณหลังจากถูกบีบอัดแล้วไปใช้ เช่น ถ้าเราต้องการทราบว่า ภาพนั้นมีรูปร่างอย่างไรเราก็สามารถใช้ภาพที่ถูกบีบอัดสูงไปใช้ได้และไม่จำเป็นจะต้องใช้ภาพที่มีความละเอียด มากนัก แต่ถ้าหากว่าเราต้องการทราบรายละเอียดมากขึ้นก็ต้องบีบอัดน้อยลงเพื่อให้คงไว้ซึ่งความละเอียดสูงขึ้น แนวคิดดังกล่าวนี้เราก็สามารถนำไปใช้กับสัญญาณใน 1 มิติ ได้เช่นกัน

2.3 การเปลี่ยนฐานหลักและแมทริกซ์การแปลง

2.3.1 แมทริกซ์การแปลงฟูเรียร์แบบไม่ต่อเนื่อง (Discrete Fourier Transform Matrix)

การแปลงฟูเรียร์เป็นเทคนิคในการวิเคราะห์สัญญาณ โดยแปลงสัญญาณที่อยู่ในโดเมนเวลา ให้อยู่ใน โดเมนความถี่ อย่างไรก็ตามโดยปกติแล้วสัญญาณในธรรมชาติที่วัดได้จะมีลักษณะเป็นจุดข้อมูล ไม่สามารถ เขียนอยู่ในรูปของฟังก์ชันได้ กระบวนการแปลงฟูเรียร์ในปัจจุบันจึงใช้การแปลงฟูเรียร์แบบไม่ต่อเนื่อง (Discrete Fourier Transform) โดยการสร้างแมทริกซ์การแปลง ที่ประกอบไปด้วยสัญญาณรูปไซน์เชิงซ้อน (Complex sinusoid) เพื่อใช้สำหรับการแปลงสัญญาณจากโดเมนเวลาเป็นสัญญาณในโดเมนความถี่ ซึ่งแมทริกซ์จะมี ลักษณะดังนี้

$$\Psi = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & w & w^2 & w^3 & w^4 & \dots & w^{(M-1)} \\ 1 & w^2 & w^4 & w^6 & w^8 & \dots & w^{2(M-1)} \\ 1 & w^3 & w^6 & w^9 & w^{12} & \dots & w^{3(M-1)} \\ 1 & w^4 & w^8 & w^{12} & w^{16} & \dots & w^{4(M-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w^{N-1} & w^{2(N-1)} & w^{3(N-1)} & w^{4(N-1)} & \dots & w^{(N-1)(M-1)} \end{bmatrix} \quad (2.12)$$

โดยให้ $w = e^{-\frac{2\pi j}{N}}$ และ N คือจำนวนจุดข้อมูลของสัญญาณในโดเมนเวลา สำหรับการแปลงฟูเรียร์แบบไม่ต่อเนื่อง (Discrete Fourier Transform) นั้นเป็นพื้นฐานสำคัญเนื่องจากเมื่อเราทราบสัญญาณในโดเมนเวลา ก็จะสามารถแปลงสัญญาณนี้ให้เป็นสัญญาณในโดเมนความถี่ได้ด้วย แมทริกซ์การแปลง ของสัญญาณรูปไซน์เชิงซ้อน (Complex sinusoid) และสัญญาณการสั่นสะเทือนในโดเมนความถี่นั้นมีลักษณะเป็นสัญญาณสปาร์สซึ่ง สอดคล้องกับการนำไปใช้ในคอมพิวเตอร์เช่นซึ่งโดยจะกล่าวเพิ่มเติมถึงในหัวข้อ 3.1.3

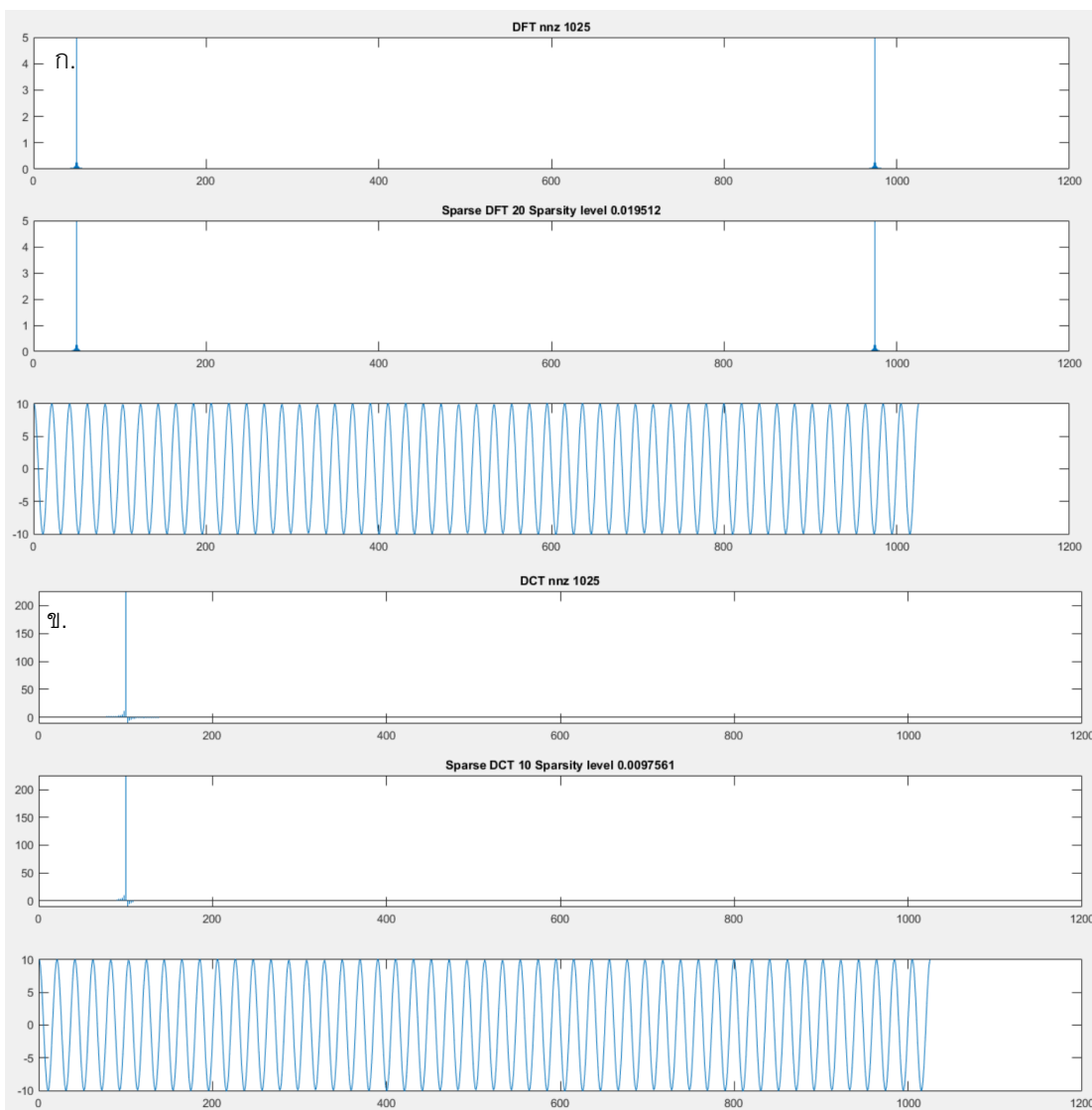
2.3.2 การแปลงโคไซน์แบบไม่ต่อเนื่อง (Discrete Cosine Transform Matrix)

จากการแปลงฟูเรียร์ในหัวข้อข้างต้นจะเห็นว่า แมทริกซ์การแปลงจะเป็นการใช้ สัญญาณรูปไซน์เชิงซ้อน (Complex sinusoid) ในการแปลงสัญญาณจากโดเมนเวลาเป็นโดเมนความถี่ สังเกตว่าสัญญาณรูปไซน์เชิงซ้อนสามารถเขียนอยู่ในรูปของผลรวมระหว่างสัญญาณรูปโคไซน์และสัญญาณรูปไซน์โดยค่า w ในสมการ (2.12) สามารถเขียนได้เป็น $w = e^{-\frac{2\pi j}{N}} = \cos\left(\frac{2\pi}{N}\right) + j\sin\left(\frac{2\pi}{N}\right)$ ทำให้คำตอบที่ได้หลัง

การแปลงจำนวนเชิงซ้อน สามารถนำไปใช้หาค่าแอมพลิจูดและเฟสของสัญญาณได้ สำหรับการแปลงโคไซน์นั้น เป็นการเลือกใช้เฉพาะพจน์ของโคไซน์เท่านั้นเพื่อใช้ในการแปลงสัญญาณจากโดเมนเวลาเป็นโดเมนความถี่ โดยการแทนค่าให้ $w = \cos\left(\frac{2\pi}{N}\right)$ ลงในสมการ (2.12) และตัด แถวที่เป็นจำนวนคู่ออกไป เนื่องด้วยการแปลงโคไซน์จะต้องเพิ่มความถี่ทีละครึ่งหนึ่งของความยาวคลื่นแทนที่จะเป็นทีละหนึ่งความยาวคลื่นเหมือนกับการแปลงฟูเรียร์ในสมการที่ (2.12) จะได้แมทริกซ์การแปลงที่มีลักษณะดังนี้

$$\psi = \sqrt{\frac{2}{N}} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \dots & \frac{1}{\sqrt{2}} \\ \cos\left(\frac{\pi}{2N}\right) & \cos\left(\frac{3\pi}{2N}\right) & \dots & \cos\left(\frac{(2m-1)\pi}{2N}\right) \\ \cos\left(\frac{\pi}{2N}\right) & \cos\left(\frac{6\pi}{2N}\right) & \dots & \cos\left(\frac{2(2m-1)\pi}{2N}\right) \\ \vdots & \vdots & \ddots & \vdots \\ \cos\left(\frac{(n-1)\pi}{2N}\right) & \cos\left(\frac{(n-1)3\pi}{2N}\right) & \dots & \cos\left(\frac{(n-1)(2m-1)\pi}{2N}\right) \end{bmatrix} \quad (2.13)$$

การแปลงโคไซน์ไม่ต่อเนื่องมีความสำคัญมากกับการใช้คอมเพรสชันซึ่งเนื่องจากคุณสมบัติที่เรียกว่า ความเข้มข้นของพลังงาน (Energy compaction) ที่สูงกว่าเมื่อเทียบกับการแปลงฟูเรียร์ เนื่องจากที่ตำแหน่งความถี่เดียวกันการกระจุกตัวของความถี่ที่มีแอมพลิจูดสูงสำหรับการแปลงโคไซน์จะมีการกระจุกตัวอยู่สูงกว่า ส่งผลให้การแปลงโคไซน์มี ความเข้มข้นของพลังงานสูงกว่า (แอมพลิจูดสูงกว่าส่งผลให้มีพลังงานสูงกว่า) และเมื่อมีการกระจุกตัวอยู่ในแต่ละความถี่สูงก็จะทำให้สัญญาณในโดเมนความถี่ที่ถูกแปลงด้วยการแปลงโคไซน์มีความสปีร์สูงกว่การแปลงฟูเรียร์เช่นกัน

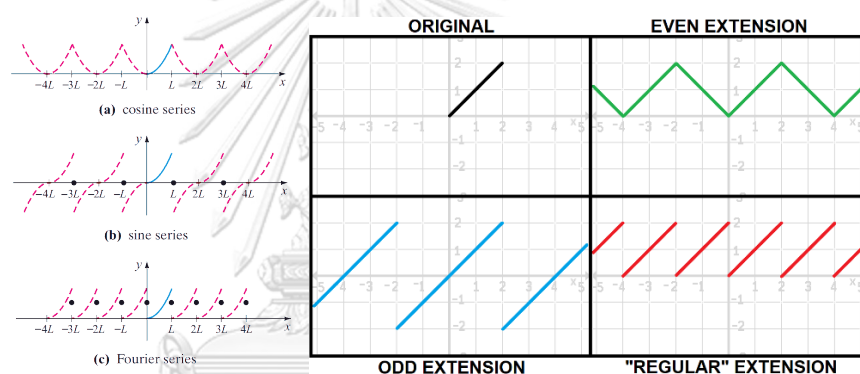


รูปที่ 2.8 เปรียบเทียบการบีบอัดสัญญาณรูปไซน์โดย ก. ใช้การแปลงฟูเรียร์ ข. ใช้การแปลงโคไซน์

รูปที่ 2.8(ก.) และ รูปที่ 2.8(ข.) แถวบนสุดคือสัญญาณในโดเมนความถี่ก่อนทำการบีบอัดสัญญาณ แถวตรงกลางคือสัญญาณในโดเมนความถี่หลังการบีบอัด (ถูกบีบอัดจนเหลือแอมพลิจูดที่มีขนาดสูงสุดเพียง 1% ของแอมพลิจูดทั้งหมด) แถวล่างสุดคือสัญญาณในโดเมนเวลาหลังการบีบอัด เมื่อกำหนดค่าความสปีร์สของสัญญาณโดยการแปลงฟูเรียร์และการแปลงโคไซน์หลังจากการบีบอัดทั้งสองแบบแล้วนำมาเปรียบเทียบกัน โดยคำนวณจากสูตร $L_S = \frac{N_{0.01}}{N}$ หรืออัตราส่วนระหว่างจำนวนความถี่ที่มีแอมพลิจูดสูงสุด 1% แรกและจำนวนของความถี่ทั้งหมด จะพบว่า ความสปีร์สของสัญญาณจากการแปลงฟูเรียร์จะเท่ากับ 0.019512 แต่สัญญาณจากการแปลงโคไซน์จะเท่ากับ 0.0097561 ซึ่งมีค่าต่ำกว่า แสดงให้เห็นว่าความสปีร์สของสัญญาณซึ่งถูกแปลงโดยการแปลงโคไซน์สูงกว่า

การแปลงฟูเรียร์ผลลัพธ์ที่ได้จะเป็นจำนวนเชิงซ้อน ซึ่งต่างจากการแปลงโคไซน์ ที่ผลลัพธ์เป็นจำนวนจริงเท่านั้น ทำให้การปรับแต่งสัญญาณบนโดเมนความถี่ของการแปลงโคไซน์มีความซับซ้อนต่ำกว่าการแปลงแบบฟูเรียร์ ซึ่งสำหรับการลดทอนหรือบีบอัดสัญญาณ (Signal Truncation and Compression) การแปลงโคไซน์ไม่ต่อเนื่องจะสามารถบีบอัดสัญญาณได้มากกว่าเนื่องจากผลความเข้มข้นของพลังงาน (Energy compaction) ที่สูงกว่า [10]

นอกจากนี้ ส่วนการแปลงโคไซน์มีเงื่อนไขขอบเขตแบบคู่ (Even extension function) จึงมีความต่อเนื่องบริเวณขอบของสัญญาณมากกว่าการแปลงฟูเรียร์ เนื่องจากการแปลงฟูเรียร์มีเงื่อนไขขอบเขตทั้งแบบคู่และแบบคี่รวมกัน (โคไซน์และไซน์รวมกัน) ทำให้มีเงื่อนไขขอบเขตแบบปกติ (Regular extension function) [10] สังเกตได้จากรูปที่ (2.9) ซึ่งถ้าหากบริเวณขอบเขตมีความไม่ต่อเนื่องสูง ค่าของสัมประสิทธิ์ที่จำเป็นต้องมีในการใช้เป็นตัวแทนสัญญาณก็ต้องเพิ่มมากขึ้นด้วยเพื่อความแม่นยำในการสร้างสัญญาณ หรืออาจพิจารณาได้ว่า ค่าสัมประสิทธิ์มีมากขึ้นความเป็นสปราร์คก็ลดลงเช่นกัน[10]



ก. ข.

รูปที่ 2.9 ก.) แสดงลักษณะของอนุกรมโคไซน์ ไซน์ และฟูเรียร์

ข.) สัญญาณ $y = f(x) = x$ เมื่อทำการเพิ่มขอบเขตออกไปในทิศทางลบและบวกโดยเงื่อนไขขอบเขตแบบต่างๆ

การที่การแปลงโคไซน์มีความสปราร์คสูงกว่า การแปลงฟูเรียร์ทำให้สัญญาณสามารถเก็บรักษารายละเอียดของสัญญาณได้มากกว่าในจำนวนความถี่ที่เท่ากัน การทำคอมเพรสชันจึงนิยมใช้แมทริกซ์การแปลงโคไซน์ ดังนั้น แมทริกซ์การแปลง ψ ที่จะกล่าวถึงต่อไปจะใช้แมทริกซ์การแปลงโคไซน์ทั้งหมด

2.3.3 ความสัมพันธ์ของแถวและหลักของเมทริกซ์การแปลงกับค่าสัญญาณในโดเมนเวลาและในโดเมนความถี่

สำหรับการแปลงสัญญาณระหว่างโดเมนความถี่และโดเมนเวลานั้นมีพื้นฐานมาจากการเปลี่ยนฐานหลัก หรือ (Change of basis) ซึ่งเป็นการแปลงเวกเตอร์ที่อยู่ในมิติหนึ่งไปอีกมิติหนึ่ง โดยจะเขียนอยู่ในรูป $x_t = \psi x_s$ กำหนดให้ ψ เป็นเมทริกซ์การแปลงรูปแบบต่างๆ ไม่ว่าจะเป็นการแปลงฟูเรียร์หรือการแปลงโคไซน์ สมการด้านล่างแสดงตัวอย่างการเปลี่ยนฐานหลักในกรณีที่สัญญาณในโดเมนเวลา x_t และสัญญาณในโดเมนความถี่ x_s มีสมาชิก 5 ตัว โดยสมการ (2.14) แสดงการแปลงจากโดเมน โดเมนความถี่ไปเป็นโดเมนเวลา ส่วนสมการ (2.15) แสดงการแปลงกลับจากโดเมนเวลาไปเป็นโดเมนความถี่

$$\begin{matrix} x_t & \psi & x_s \\ \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} & = \begin{bmatrix} \psi_{1,1} & \psi_{1,2} & \psi_{1,3} & \psi_{1,4} & \psi_{1,5} \\ \psi_{2,1} & \psi_{2,2} & \psi_{2,3} & \psi_{2,4} & \psi_{2,5} \\ \psi_{3,1} & \psi_{3,2} & \psi_{3,3} & \psi_{3,4} & \psi_{3,5} \\ \psi_{4,1} & \psi_{4,2} & \psi_{4,3} & \psi_{4,4} & \psi_{4,5} \\ \psi_{5,1} & \psi_{5,2} & \psi_{5,3} & \psi_{5,4} & \psi_{5,5} \end{bmatrix} & \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \end{bmatrix} \end{matrix} \quad (2.14)$$

$$\begin{matrix} \begin{bmatrix} \psi'_{1,1} & \psi'_{2,1} & \psi'_{3,1} & \psi'_{4,1} & \psi'_{5,1} \\ \psi'_{1,2} & \psi'_{2,2} & \psi'_{3,2} & \psi'_{4,2} & \psi'_{5,2} \\ \psi'_{1,3} & \psi'_{2,3} & \psi'_{3,3} & \psi'_{4,3} & \psi'_{5,3} \\ \psi'_{1,4} & \psi'_{2,4} & \psi'_{3,4} & \psi'_{4,4} & \psi'_{5,4} \\ \psi'_{1,5} & \psi'_{2,5} & \psi'_{3,5} & \psi'_{4,5} & \psi'_{5,5} \end{bmatrix} & \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} & = & \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \end{bmatrix} \end{matrix} \quad (2.15)$$

จะเห็นว่าการแปลงโดเมนจะสามารถทำได้โดยการคูณกันของเมทริกซ์ เช่น ถ้าหากต้องการ แปลงสัญญาณ x_s เป็น x_t จะต้องนำแถวของสมาชิกใน ψ ไปหาผลคูณแบบจุด (Dot Multiplication) กับ สมาชิกทั้งหมดใน x_s โดยที่จะสังเกตว่า สมาชิก x_t ตัวที่ i ใดๆ จะเป็นผลจากคูณของแถว ที่ i ใน ψ เท่านั้น เช่น x_1 จะสอดคล้องกับสมาชิกแถวที่ 1 ของ ψ หรืออาจกล่าวว่า $\psi_{1,1-5}$ เท่านั้น ที่จะถูกนำไปคูณกับทุกสมาชิกใน x_s เพื่อสร้าง $x_1 = \psi_{1,n} \times X_n$ โดยไม่มีความเกี่ยวข้องกับสมาชิกในแถวอื่นเลย จึงกล่าวได้ว่าแถวของสมาชิกใน ψ มีความสอดคล้องกับจุดข้อมูลในโดเมนเวลา ทำนองเดียวกันสำหรับโดเมนความถี่ ซึ่งสอดคล้องกับหลักหรือคอลัมน์ของสมาชิกใน ψ ที่ถูกแปลงเป็นแถวในเมทริกซ์ ψ'

เมื่อทราบแล้วว่า เมทริกซ์การแปลงและเวกเตอร์สัญญาณทั้งในโดเมนเวลาและในโดเมนความถี่ สอดคล้องกันอย่างไรทำให้สามารถหาค่าสัญญาณในโดเมนความถี่บางตำแหน่งหรือ สัญญาณในโดเมนเวลาบางตำแหน่ง ได้เฉพาะเจาะจงโดยไม่ต้องใช้เมทริกซ์ ψ ขนาดเต็มในการแปลง

2.4 คอมเพรสเซนซิง(Compressed sensing)

คอมเพรสเซนซิงเป็นเทคนิคการประมวลผลสัญญาณรูปแบบหนึ่งสำหรับการจัดเก็บและการสร้างสัญญาณใหม่โดยการหาผลเฉลยของระบบสมการไม่ชัดเจน ซึ่งมีหลักการว่า สัญญาณสามารถถูกสร้างกลับคืนมาได้โดยมีจำนวนตัวอย่างสุ่มที่ต่ำกว่าทฤษฎีการสุ่มตัวอย่างของ แซนนอนและไนควิสต์ (Shannon-Nyquist sampling theorem) ผ่านวิธีการหาค่าเหมาะสมที่สุด (Optimization) ภายใต้เงื่อนไข 2 ประการคือ 1. สัญญาณที่จะผ่านกระบวนการนี้ต้องมีคุณสมบัติเป็น สเปิร์ส (Sparse signal) ในบางโดเมน 2.คุณสมบัติความไม่สอดคล้อง (Incoherent) ระหว่างสัญญาณที่ถูกวัดกับเมทริกซ์การแปลง ψ ซึ่งสามารถอธิบายในรูปของคุณสมบัติการจำกัดสมมิติ (Restrict Isometry Properties) [17]

กำหนดให้ x เป็นสัญญาณใดๆ ซึ่งมีความสเปิร์สในโดเมนความถี่ สัญญาณ x ในโดเมนความถี่แสดงด้วย x_S เนื่องจากเป็นสัญญาณสเปิร์ส จำนวนข้อมูลที่มีขนาดเล็ก หรือเท่ากับศูนย์จึงมีจำนวนมาก หากให้สัญญาณ x_S มีขนาด $N \times 1$ และ K คือจำนวนของความถี่ที่มีขนาดแอมพลิจูดไม่เท่ากับศูนย์ ซึ่งจะถือว่า มีค่าน้อยกว่า N มาก โดยปกติแล้วหากจะต้องการทราบข้อมูลของสัญญาณ x_S ในอดีตสามารถทำได้โดยวัดสัญญาณ x_t ซึ่งเป็นสัญญาณ x ในโดเมนเวลาที่มิขนาด N จุด (ต่อหนึ่งหน่วยวินาที) และมีความถี่สุ่มเพียงพอโดยตรง แล้วนำไปแปลงฟูเรียร์เพื่อให้ได้ x_S แต่เนื่องจาก x_S มีความสเปิร์สในโดเมนความถี่ แสดงให้เห็นว่าข้อมูลที่จำเป็นเพื่อให้ทราบ x_S จึงมีค่าน้อยด้วย คอมเพรสเซนซิงทำให้สามารถวัดสัญญาณ y_t หรือสัญญาณสุ่มซึ่งมีความกว้างของคาบไม่แน่นอนแทนการวัดสัญญาณ x_t ซึ่ง y_t มีขนาด M จุด โดย $K \ll M \ll N$ ได้ โดยความสัมพันธ์ระหว่าง y_t และ x_t สามารถเขียนได้ดังสมการเชิงเส้น

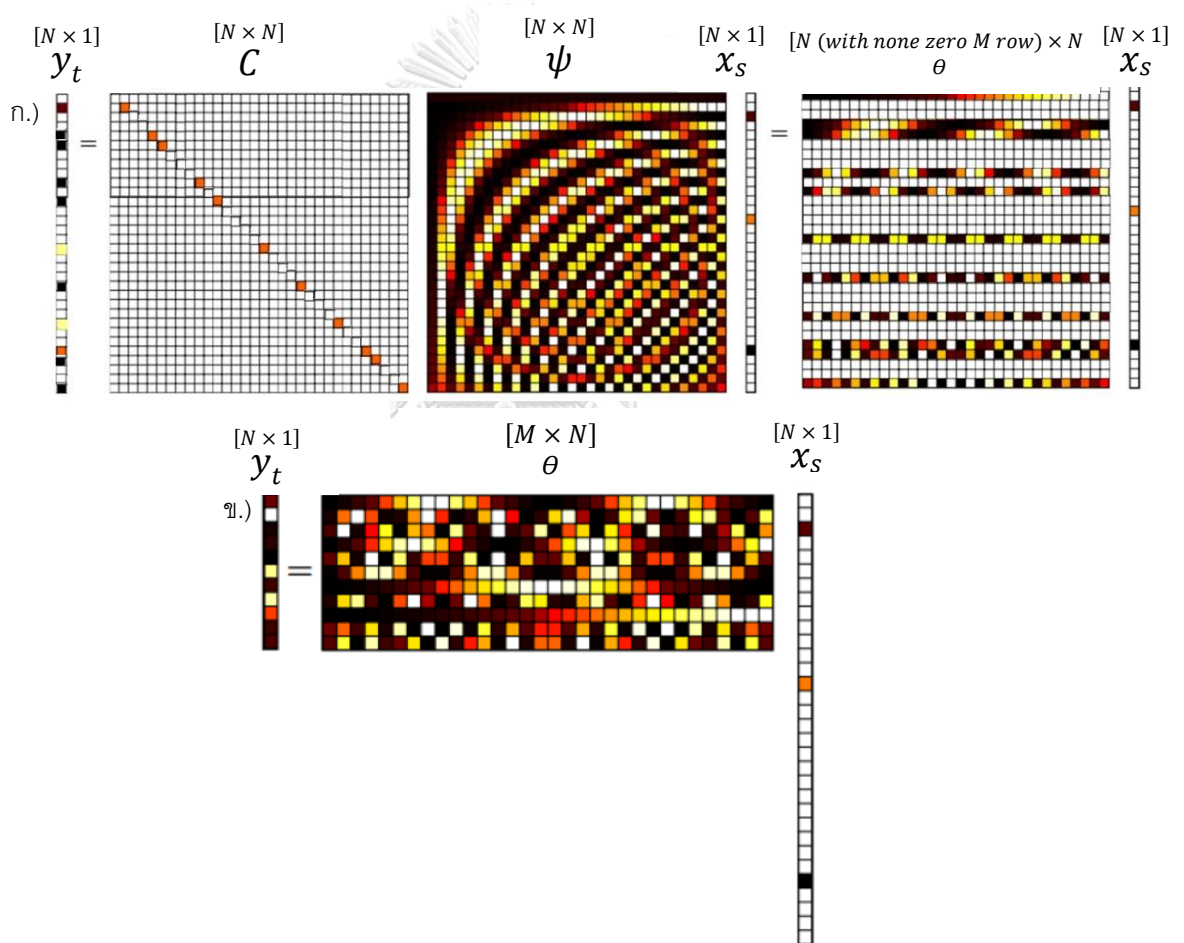
$$\begin{bmatrix} M \times 1 \\ y_t \end{bmatrix} = \begin{bmatrix} M \times N \\ C \end{bmatrix} \begin{bmatrix} N \times 1 \\ x_t \end{bmatrix} \quad (2.16)$$

โดย C คือเมทริกซ์การวัด (Measurement Matrix) จริงๆ แล้ว C มีขนาด $N \times N$ และเป็นเมทริกซ์ทแยงมุมที่มีสมาชิกเป็น 0 หรือ 1 เท่านั้น แต่มีจำนวนแถวที่มีสมาชิกทั้งหมดของแถวไม่เท่ากับศูนย์อยู่เพียง M แถว โดยตำแหน่งของสมาชิกที่มีค่าเท่ากับ 1 จะได้จากการสุ่มโดยมีการกระจายตัวแบบแบบเกาส์ (Gaussian distributed random variable) หรือแบบเบอร์นูลลี (Bernoulli distributed random variable) หรือการสุ่มแบบปกติ ก็สามารถทำได้ เพื่อให้ได้มาซึ่งข้อมูลการวัด y_t ที่มีลักษณะแบบสุ่ม และเนื่องจาก C มีแถวที่สมาชิกไม่เท่ากับศูนย์อยู่เพียง M แถว จึงสามารถลดขนาดมิติลงได้เหลือ $M \times N$ โดยหลักการเดียวกับที่เขียนในหัวข้อ 2.1.2.1 สมการที่ (2.16) แสดงการเลือกจุดของสัญญาณใน x_t ทั้งหมด M จุดเพื่อนำไปสร้างเป็นสัญญาณ y_t ด้วยการคูณเมทริกซ์ C ซึ่งถือเป็นการบีบอัดสัญญาณ กล่าวคือ บีบข้อมูลจำนวนมากให้มีจำนวนน้อยลง เพื่อจัดการสัญญาณที่ข้อมูลน้อย และนำไปประมวลกลับหาสัญญาณดั้งเดิมที่ใหญ่กว่า ซึ่งในงานวิจัยนี้ไม่ได้มีจุดมุ่งหมายที่จะทำการบีบอัดสัญญาณ โดยใช้ C และ x_t เพื่อหาค่า y_t แต่ต้องการสร้างสัญญาณ x_t ด้วยการสังเคราะห์จากสัญญาณการวัดที่ไม่สมบูรณ์ y_t และจากเมทริกซ์การวัด C ซึ่งไม่สามารถทำได้ด้วยการย้ายข้างสมการปกติเนื่องจากสมการดังกล่าวเป็นการบีบอัดแบบสูญเสียรายละเอียด

จากที่เราทราบว่าสัญญาณสเปกตรัมที่มีความหมายคือสัญญาณนั้นจะมีสมาชิกที่มีค่าไม่เท่ากับศูนย์อยู่น้อยในบางโดเมนในกรณีที่สัญญาณมีความสเปกตรัมในโดเมนความถี่ เราสามารถเขียน x ให้อยู่ในโดเมนความถี่ได้ โดยใช้ความสัมพันธ์ของแมทริกซ์การแปลง $x_t = \psi x_s$ โดยที่ ψ คือแมทริกซ์ที่ใช้สำหรับการแปลง (Transformation matrix) เมื่อแทนความสัมพันธ์นี้ลงในสมการ (2.16) จะได้

$$y_t = C\psi x_s = \theta x_s \tag{2.17}$$

สำหรับสมการที่ (2.17) จะเป็นตัวแทนของระบบสมการที่เป็นแมทริกซ์มีขนาดดังแสดงในรูปที่ 2.10



รูปที่ 2.10 สมการ $y_t = C\psi x_s = \theta x_s$ ก.) ก่อนลดมิติ ข.) หลังลดมิติ

จากรูปที่ 2.10 จุดสีเข้มบนแมทริกซ์ C แทนจุดข้อมูลที่มีค่าเท่ากับ 1 โดยจะมีจำนวนอยู่ M ตัว ส่วนจุดสีขาว แทนจุดข้อมูลที่มีค่าเท่ากับ 0 จะสังเกตว่าขนาดของแมทริกซ์ C และ θ นั้นเท่ากับ $[N \times N]$ และ y_t เท่ากับ $[N \times 1]$ แต่จะเห็นว่าแถวของสมาชิกในแมทริกซ์ θ มีจำนวน M แถวที่ไม่เท่ากับศูนย์และเช่นเดียวกับ y_t ที่มีจำนวนสมาชิกที่ไม่เท่ากับศูนย์เท่ากับ M เช่นเดียวกันใน รูปที่ 2.10(ก.) จะถือว่าเป็นรูป

สมการที่ยังไม่พร้อมนำไปใช้งานเนื่องจาก คอมเพรสเซนซึ่งต้องการระบบสมการเชิงเส้นไม่ชัดเจน จึงจำเป็นต้องลดขนาดของ y_t และจำนวนแถวของ θ ให้มีขนาดเท่ากับ M โดยการตัดแถวของ θ และสมาชิกของ y_t ที่มีค่าเท่ากับศูนย์ออกไปเพื่อให้ได้ระบบสมการเชิงเส้นไม่ชัดเจน ที่จะนำไปใช้งานได้ดัง รูปที่ 2.10(ข.)

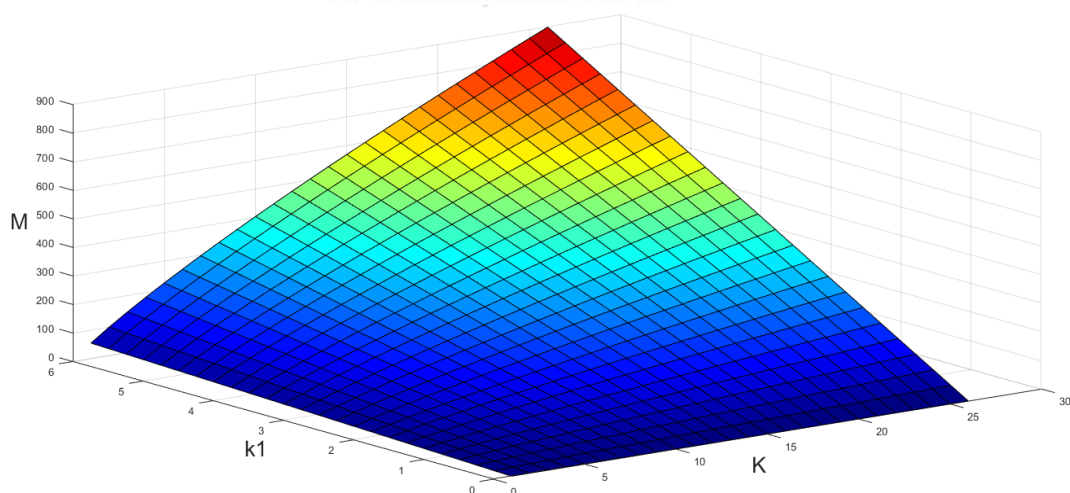
เพื่อให้การคำนวณมีโอกาสที่จะหาผลเฉลยแบบสปาร์สที่สุด (Sparsest solution) แมทริกซ์ C และจำนวนจุดข้อมูลจะต้องมีลักษณะดังนี้

1. แมทริกซ์การวัด C จำเป็นต้องไม่มีความสอดคล้อง (Incoherent) กับแมทริกซ์การแปลง ψ หรือเรียกได้ว่าแถวของ C จำเป็นจะต้องมีความตั้งฉาก (Orthogonal) กับ คอลัมน์ของ ψ ให้มากที่สุด

2. จำนวนจุดข้อมูลของสัญญาณ y_t จำนวน M จุด จำเป็นจะต้องมีจำนวนมากพอโดยสามารถประมาณได้ตามสมการที่ (2.18) [14]

$$M \approx k_1 K \log\left(\frac{N}{K}\right) \quad (2.18)$$

โดย N คือจำนวนจุดข้อมูลของสัญญาณต้นแบบ x , K คือจำนวนของตำแหน่งของสัญญาณที่ไม่เท่ากับศูนย์ของ x ในโดเมนใดๆ (ในกรณีนี้คือตำแหน่งของสัญญาณในโดเมนความถี่ที่มีค่าแอมพลิจูดไม่เท่ากับศูนย์) ค่าคงที่ k_1 ขึ้นอยู่กับความไม่สอดคล้องระหว่าง C และ ψ ถ้ายิ่งสอดคล้องกันมาก k_1 จะยิ่งมาก เรียกได้ว่าเงื่อนไขทั้งสองนี้เป็นการทำให้มั่นใจว่าแมทริกซ์ $C\psi$ ยังคงคุณสมบัติการแปลงแบบยูนิเทรี (Unitary transformation) สำหรับเวกเตอร์ x_s เพื่อให้ความสัมพันธ์ของตัวแปรแต่ละตัวในแต่ละมิติของเวกเตอร์ x_s คงเดิมหลังจากการถูกแปลงเป็นแมทริกซ์ y_t จะทำให้คำตอบที่ได้จากการใช้บรรทัดฐาน 1 ในการหาค่าเหมาะสมที่สุดมีความใกล้เคียงกับผลเฉลยที่ต้องการมากขึ้น



รูปที่ 2.11 พื้นผิวแสดงความสัมพันธ์ระหว่างจำนวนของความถี่ที่ไม่เท่ากับศูนย์ของสัญญาณ K , การกระจายตัวของการสุ่มข้อมูล k_1 , จำนวนจุดที่จำเป็นต้องทำการวัด M

รูปที่ 2.11 แสดงความสัมพันธ์ระหว่างจำนวนที่จำเป็นต่อวัด M การกระจายตัวของการสุ่มข้อมูล k_1 และจำนวนของความถี่ที่ไม่เท่ากับศูนย์ในโดเมนความถี่ของสัญญาณหรือ K จะสังเกตว่าถ้าหากการเก็บข้อมูลเป็นไปในลักษณะเป็นคาบสูง (k_1 มาก) จะส่งผลให้การใช้คอมเพรสชันซึ่งจำเป็นต้องใช้จำนวนสัญญาณที่จำเป็นต่อวัด M มากขึ้น เช่นเดียวกับจำนวนของความถี่ที่ไม่เท่ากับศูนย์ในโดเมนความถี่ของสัญญาณหรือ K ที่มากขึ้น ซึ่งแสดงถึงสัญญาณซับซ้อนมากขึ้น ก็จะต้องใช้จำนวนจุดของสัญญาณมากขึ้นเช่นกัน

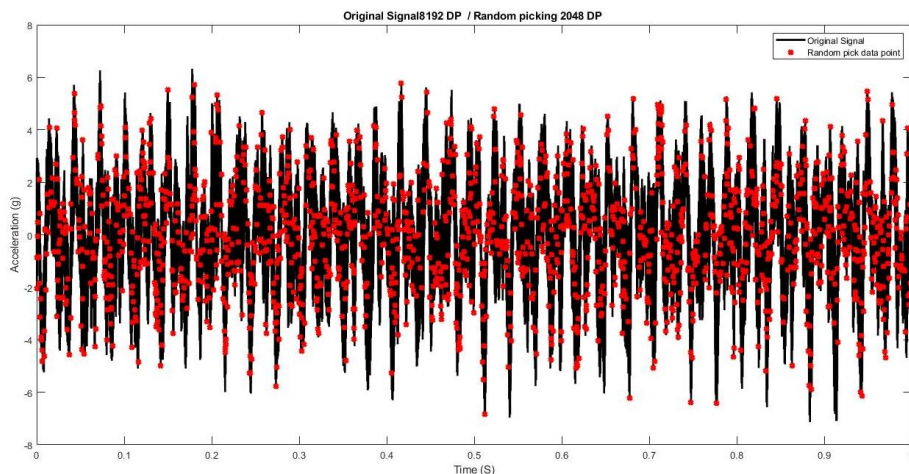
การแก้ปัญหาคอมเพรสชันซึ่งสามารถทำได้หลายวิธีด้วยกัน ในที่นี้จะขอกล่าวถึงวิธีสองวิธีที่ใช้กันในปัจจุบัน วิธีแรกคือ การใช้กระบวนการหาจุดที่ดีที่สุดแบบคอนเวกซ์ (Convex Optimization) [02] ในการแก้ระบบสมการเชิงเส้นแบบไม่ชัดเจน (Underdetermine linear system) เพื่อให้ได้มาซึ่งคำตอบของสมการเป็นสัญญาณสเปิร์ส (Sparse signal) วิธีที่สองก็คือการใช้อัลกอริทึมละโมบ (Greedy algorithm) [11] ในการค้นหาแมทริกซ์ที่เหมาะสมที่สุดสำหรับการแปลงเชิงเส้น (Transformation matrix) ซึ่งก็คือแมทริกซ์ $C\psi = \theta$ ของสัญญาณที่บีบอัด แล้วจึงนำไปใช้สร้างสัญญาณสเปิร์ส (Sparse signal) [02] ในภายหลัง

2.4.1 กระบวนการหาจุดที่ดีที่สุดแบบคอนเวกซ์ (Convex Optimization-CVX)

จากรูปที่ 2.10(ข.) ซึ่งเป็นระบบสมการไม่ชัดเจนสามารถหาได้โดยการใช้วิธีการหาค่าเหมาะสมที่สุด ที่เคยกล่าวไปแล้วใน 2.1 ได้ดังนี้

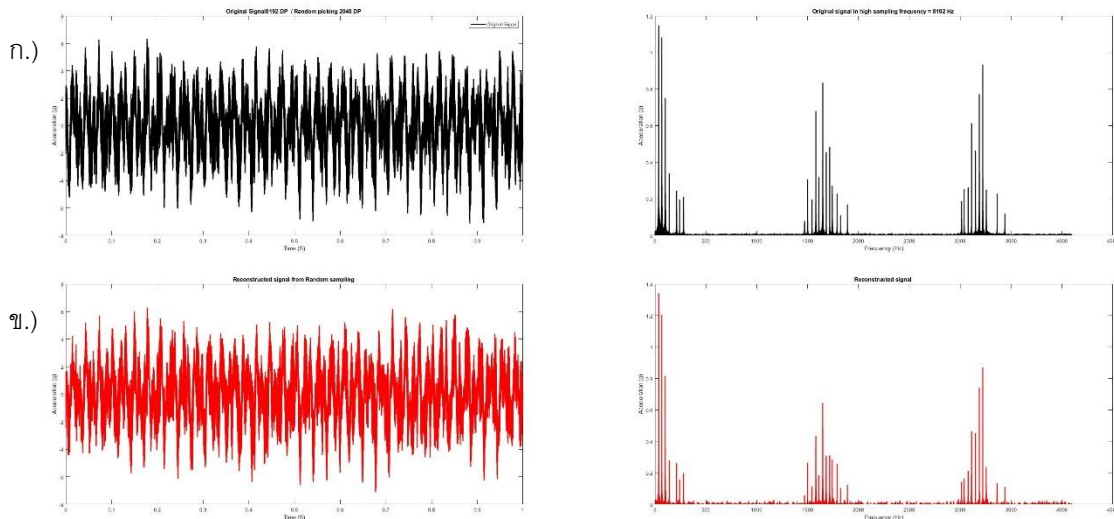
$$\hat{x}_s = \underset{x_s}{\operatorname{argmin}} \|x_s\|_1 \text{ โดยขึ้นอยู่กั } \|C\psi x_s - y_t\|_2 < \epsilon \quad (2.19)$$

โดยปกติเมื่อวัดสัญญาณ y_t มักจะมีสัญญาณรบกวนขนาดใดๆ เกิดขึ้นเสมอ ทำให้เทอม $\|C\psi x_s - y_t\|_2$ จะมีค่าไม่เท่ากับศูนย์แต่จะมีความคลาดเคลื่อน ซึ่งจะถูกกำหนดขนาดเป็น ϵ การหาผลลัพธ์โดยการแก้สมการที่ (2.19) โดยตรงเรียกว่า วิธีการหาบรรทัดฐาน 1 ที่น้อยที่สุดโดยใช้การหาค่าเหมาะสมที่สุดแบบคอนเวกซ์ (L-1 Minimization by Convex Optimization) และผลลัพธ์ที่จะได้คือ x_s ซึ่งจะเป็นสัญญาณที่ถูกสร้างขึ้นใหม่จากข้อมูลการวัดและอยู่ในโดเมนความถี่ ซึ่งเป็นโดเมนที่สัญญาณเป็นสัญญาณสเปิร์ส (Sparse signal)



รูปที่ 2.12 สัญญาณต้นแบบและการสุ่มเลือกจุดในสัญญาณ

รูปที่ 2.12 แสดงตัวอย่างในการสุ่มวัดสัญญาณเพื่อให้ได้ค่า y_t (จุดสีแดง) ซึ่งมีจำนวนข้อมูล 2048 จุด จากสัญญาณต้นแบบที่มีความถี่สุ่ม 8192 เฮิรตซ์ซึ่งวัดในเวลา 1 วินาที (เส้นสีดำ) กระบวนการนี้จะทำให้ได้เมทริกซ์การวัด C ซึ่งแสดงตำแหน่งที่มีจุดข้อมูลอยู่หลังจากนั้นก็ให้นำตัวแปร y_t และ C มาสร้างระบบสมการเชิงเส้น $y_t = C\psi x_s = \theta x_s$ โดย ψ จะเป็นเมทริกซ์การแปลง ซึ่งในที่นี้ใช้เมทริกซ์การแปลงโคไซน์ตามเหตุผลที่ได้กล่าวไปแล้วในหัวข้อ 2.4 หลังจากนั้นแก้สมการด้วย กระบวนการหาจุดที่ดีที่สุดแบบคอนเวจซ์เพื่อให้ได้ค่าประมาณของ x_s ซึ่งมีขนาดข้อมูล ขนาด 8192 จุดหรือสัญญาณสปาร์สในโดเมนความถี่ออกมา

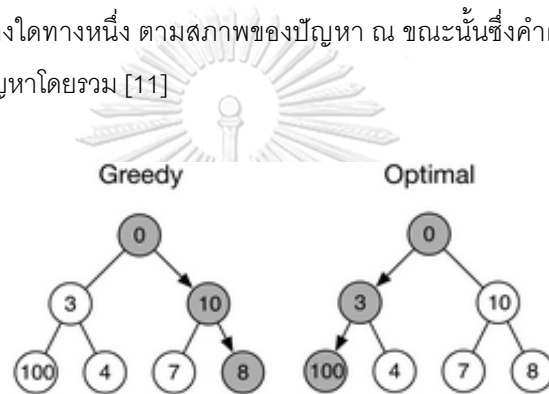


รูปที่ 2.13 สัญญาณทดสอบในโดเมนเวลาและโดเมนความถี่ ก.) สัญญาณต้นแบบ (สีดำ)
ข.) สัญญาณสปาร์ส (สีแดง) จากการหาค่าเหมาะสมที่สุดแบบคอนเวจซ์

รูปที่ 2.13(ก.) แสดงสัญญาณต้นแบบในโดเมนเวลาและโดเมนความถี่ ส่วนผลลัพธ์ของการหาค่าเหมาะสมที่สุดแบบคอนเวจซ์ x_s จะแสดงในรูป 2.13(ข.)

2.4.2 การจับคู่สัญญาณบีบอัด (Compressed sampling Matching Pursuit-COSAMP)

วิธีการจับคู่สัญญาณบีบอัดมีพื้นฐานมาจากอัลกอริทึมละโมบ ซึ่งเป็นอัลกอริทึมแบบหนึ่งที่ถูกนำมาใช้กับปัญหาการหาค่าเหมาะสมที่สุดเช่นกัน โดยคำตอบของปัญหาในแต่ละครั้งได้มาจากลำดับของการตัดสินใจแบบละโมบหรือการเลือกเงื่อนไขที่ให้ผลลัพธ์มากที่สุด หรือ น้อยที่สุด โดยใช้เกณฑ์บางอย่างในการตัดสินใจตามสภาพของปัญหา ณ ขณะนั้น ไม่ได้ตัดสินใจตามสภาพของปัญหาโดยรวมทั้งหมด ซึ่งมีจุดเด่นที่การได้มาซึ่งคำตอบนั้นมีความรวดเร็วแต่ก็มีจุดอ่อนที่มักจะได้คำตอบที่ดีที่สุดตามต้องการ แต่ในหลายปัญหาอัลกอริทึมละโมบก็สามารถที่จะได้มาซึ่งคำตอบที่ยอมรับได้และมีประสิทธิภาพ จากรูปที่ 2.14 จะอธิบายเปรียบเทียบระหว่างการใช้ อัลกอริทึมละโมบ ในการหาผลรวมที่มากที่สุดของระบบ จะเห็นว่าอัลกอริทึมละโมบ จะทำการตัดสินใจเลือกทางใดทางหนึ่ง ตามสภาพของปัญหา ณ ขณะนั้นซึ่งคำตอบที่ได้ไม่ได้เป็นคำตอบที่เหมาะสมที่สุดจริงๆ ของปัญหาโดยรวม [11]



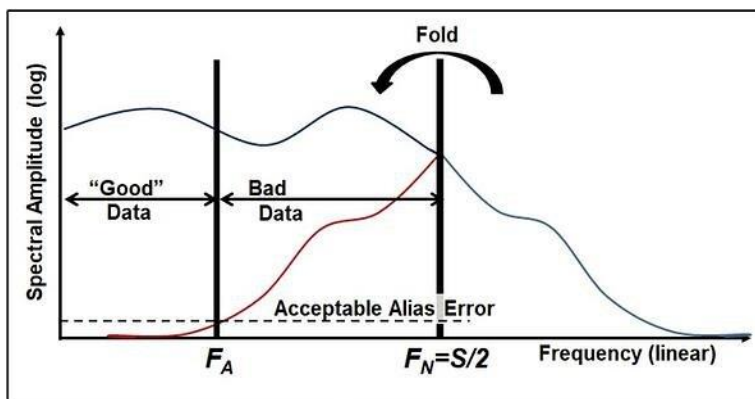
รูปที่ 2.14 แผนภาพวิธีการเลือกผลลัพธ์ของอัลกอริทึมละโมบเปรียบเทียบกับหาค่าเหมาะสมที่สุด [18]

ในภาพรวมสำหรับการนำอัลกอริทึมละโมบมาใช้ในการแก้ไข้ปัญหาของระบบสมการไม่ชัดเจนนั้น ไม่ได้เป็นการใช้เพื่อแก้สมการโดยตรงแต่นำมาใช้เพื่อลดจำนวนชุดสมการที่เหลือให้น้อยลงอย่างมาก (ด้วยการลดขนาดของเมทริกซ์ X_S ให้เล็กกว่า Y_t ด้วยวิธีการต่างๆ ที่จะถูกกล่าวถึงในหัวข้อ 3.2.2 จนระบบสมการถูกเปลี่ยนเป็นแบบระบบสมการเชิงเส้นชัดเจนเกินไป หลังจากนั้นจึงจะหาคำตอบของตัวแปรของระบบสมการด้วยการใช้ มัวร์-เพนโรส อินเวอร์ส (Moore-Penrose inverse)

2.5 วงจรกรองความถี่ต่ำผ่าน (Low pass Filter)

วงจรกรองความถี่ต่ำผ่านนั้นเป็นวงจรกรองสัญญาณที่สำคัญในการวัดการสั่นสะเทือนแบบปกติ วงจรนี้ทำหน้าที่ในการลดทอนสัญญาณที่สูงกว่าความถี่ที่กำหนดออกไปเพื่อช่วยป้องกันการเกิด คลื่นแฝง (Aliasing) เข้ามารบกวนสัญญาณในความถี่ที่มีย่านต่ำกว่า อันเนื่องมาจากที่สัญญาณมีองค์ประกอบที่ความถี่สูงกว่าอัตราไนควิสต์

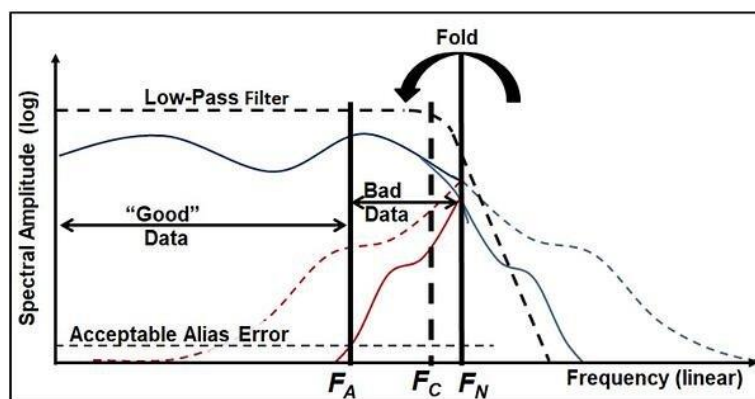
โดยปกติแล้ว วงจรกรองความถี่ต่ำผ่านถูกออกแบบมาเพื่อช่วยให้เซ็นเซอร์วัดการสั่นสะเทือนสามารถวัดค่าสัญญาณที่มีความถี่บริเวณที่ใกล้กับอัตรา ไนควิสต์ได้ดีขึ้น



รูปที่ 2.15 แผนภาพแสดงในกรณีที่สัญญาณที่ไม่ผ่านวงจรรองความถี่ต่ำผ่าน

จากรูปที่ 2.15 เป็นการแสดงตัวอย่างของสัญญาณในโดเมนความถี่ โดย เส้นสีน้ำเงิน แสดงสัญญาณในกรณีที่ไม่ม้วงจรรองความถี่ต่ำผ่าน สัญญาณส่วนที่เกินค่า $F_N = \frac{S}{2}$ จะสะท้อน (Folding) แบบสมมาตรรอบแกน F_N กลับเข้ามาครอบงำสัญญาณในบริเวณที่ต่ำกว่าอัตราในควิส์ท์ F_N ถ้าหากสัญญาณที่วัดมีความถี่ที่สูงกว่าอัตราในควิส์ท์มากๆ ก็จะทำให้การสะท้อนเกิดมากขึ้น ทำให้บริเวณที่ข้อมูลไม่สมบูรณ์ขยายขอบเขตมากขึ้น โดยสัญญาณทั้งหมดจะแบ่งช่วงออกตามความถี่ได้ทั้งหมด 3 ช่วงด้วยกันคือ บริเวณข้อมูลที่มีความสมบูรณ์ (Good Data) ซึ่งเป็นบริเวณที่ไม่ถูกการสะท้อนจากสัญญาณที่มีความถี่มากกว่าอัตราในควิส์ท์เข้ามาครอบงำ บริเวณข้อมูลไม่สมบูรณ์ (Bad Data) เป็นบริเวณที่ถูกครอบงำจากการสะท้อน ซึ่งทำให้ข้อมูลบริเวณนี้มีความผิดพลาด และ ในบริเวณที่ความถี่มากกว่าอัตราในควิส์ท์ สัญญาณในบริเวณนี้คือบริเวณที่เครื่องมือวัดปกติไม่สามารถวัดได้อย่างถูกต้องสมบูรณ์

วิธีการแก้ไขเพื่อเพิ่มบริเวณข้อมูลที่มีความสมบูรณ์ให้มากขึ้นเข้าใกล้อัตราในควิส์ท์ มักทำโดยการใช้วงจรรองสัญญาณความถี่ต่ำผ่านแบบอะนาล็อก (Analog low pass filter) เพื่อลดขนาดของสัญญาณในช่วงที่มีความถี่สูงกว่าอัตราในควิส์ท์ซึ่งจะช่วยในการลดการสะท้อนของสัญญาณที่จะเข้ามาครอบงำในย่านที่ต่ำกว่าอัตราในควิส์ท์



รูปที่ 2.16 แผนภาพแสดงในกรณีที่สัญญาณผ่านวงจรรองความถี่ต่ำผ่าน

จากรูปที่ 2.16 จะเห็นว่าในกรณีที่มึวงจรรองความถี่ต่ำผ่าน สัญญาณในบริเวณที่สูงกว่า อัตราโนควิสต์เดิม (เส้นสีน้ำเงินทึบ) ถูกลดทอนลงเนื่องวงจรรองความถี่ต่ำส่งผลให้การสะท้อนลดลง (เส้นสีแดงทึบ) บริเวณที่ข้อมูลมีความสมบูรณ์จึงมากขึ้น แต่จะเห็นได้ว่าวิธีการนี้วงจรรองความถี่ต่ำผ่านจะเริ่มทำการลดทอนสัญญาณทั้งหมดที่มีค่าสูงกว่าความถี่คัทออฟ F_C (Cutoff Frequency) ไปจนหมดเพื่อลดการสะท้อนลง

โดยปกติแล้วระบบการวัดการสั่นสะเทือนแบบปกติจำเป็นจะต้องมีวงจรรองความถี่ต่ำเพื่อช่วยในการลดการสะท้อน ส่งผลให้สัญญาณมีความถูกต้องมากขึ้น แต่สำหรับการวัดแบบสุ่มคาบเพื่อจะนำไปสร้างสัญญาณการสั่นสะเทือน X_G นั้นจะต้องคำนึงถึงความถี่คัทออฟ โดยจะต้องเลือกพิสัยของสัญญาณที่จะทำการทดสอบให้ไม่เกินไปกว่าความถี่คัทออฟ ของวงจรรองความถี่ต่ำที่มี



บทที่ 3 การวัดและการประมวลผลสัญญาณ

ในบทก่อนหน้าได้กล่าวถึงความรู้พื้นฐาน และสมการเบื้องต้นของคอมเพรสชันซึ่งไปบ้างแล้ว อย่างไรก็ตามในการนำคอมเพรสชันซึ่งไปใช้กับสัญญาณการสื่อสารอื่น จำเป็นที่จะต้องพิจารณาถึงการวัดสัญญาณ การจัดการสัญญาณที่ได้ ก่อนที่จะนำไปสร้างสัญญาณคืนกลับด้วยคอมเพรสชันซึ่งได้ ในบทนี้จะกล่าวถึงการเตรียมสัญญาณการสื่อสารอื่นให้อยู่ในรูปแบบที่เหมาะสม และจะกล่าวถึงการทำคอมเพรสชันซึ่งด้วยวิธีต่างๆ ที่จะใช้ในงานวิจัยนี้ต่อไป

3.1 การปรับปรุงให้คอมเพรสชันซึ่งไปสามารถใช้ได้กับสัญญาณการสื่อสารอื่น

คอมเพรสชันซึ่งเป็นเทคนิคที่มีพื้นฐานมาจากงานวิทยาศาสตร์ข้อมูล ซึ่งถูกนำไปประยุกต์ใช้ได้กับข้อมูลหรือสัญญาณหลากหลายรูปแบบ เช่น ข้อมูลที่เป็นตัวเลข สัญญาณภาพ สัญญาณโทรคมนาคม และสัญญาณการสื่อสารอื่น สำหรับการนำไปประยุกต์ใช้กับสัญญาณแต่ละแบบจำเป็นต้องปรับปรุงเทคนิคให้สอดคล้องกับสัญญาณที่จะนำมาใช้ โดยในที่นี้คือสัญญาณการสื่อสารอื่น

3.1.1 การได้มาซึ่งสัญญาณสุ่มวัด

การคำนวณโดยคอมเพรสชันซึ่งไปจำเป็นต้องใช้ข้อมูลในโดเมนเวลา y_t ที่มีระยะห่างของจุดข้อมูลไม่เท่ากันแบบสุ่มสำหรับการประยุกต์ใช้กับการสื่อสารอื่น สัญญาณ y_t คือสัญญาณการสื่อสารอื่นที่มีคาบการสุ่มไม่แน่นอน สำหรับในงานวิจัยนี้จะมีวิธีการได้มาซึ่งสัญญาณ y_t 3 แบบเพื่อให้เหมาะสมกับการทดสอบแบบต่างๆ

1. การสร้างข้อมูลโดยจำลองสัญญาณในโปรแกรมคอมพิวเตอร์
2. การสุ่มหยิบจุดข้อมูลจากสัญญาณการวัดด้วยความถี่สุ่มคงที่
3. การวัดสัญญาณโดยการใช้ความถี่สุ่มไม่คงที่

1. การสร้างข้อมูลโดยจำลองสัญญาณในโปรแกรมคอมพิวเตอร์

การจำลองสัญญาณในแบบแรก จะเป็นการจำลองในคอมพิวเตอร์ทั้งหมด จึงสามารถ สร้างเวกเตอร์ของเวลา t_y ที่มีระยะห่างของข้อมูลแต่ละจุดไม่เท่ากันได้โดยตรง แล้วจึงจะนำไปแทนค่าในฟังก์ชันไซน์ของสัญญาณ $y_t = f(t_y)$ สำหรับระยะห่างของแต่ละจุดข้อมูลของ t_y จะถูกกำหนดให้เป็น Δt โดยอธิบายได้จากสมการด้านล่าง

$$\Delta t \approx \Delta t_{min} + [2 \times (\Delta t_{min}) \times rand] \quad (3.1)$$

โดยกำหนดให้ Δt_{min} คือคาบของการสุ่มปกติหรือความถี่สุ่มเทียบเท่า $\frac{1}{\Delta t_{min}} = f_{erate}$ ของระบบการวัดที่ต้องการจำลองซึ่งหมายความว่า การจะวัดสัญญาณนี้ได้จำเป็นต้องใช้ระบบการวัดที่มีความถี่สุ่มเท่ากับ f_{erate} สำหรับ $rand$ คือตัวเลขทศนิยมสุ่มจาก 0 ถึง 1 ซึ่งหมายความว่าระยะห่างของจุดข้อมูลจะมีค่าต่ำสุดเท่ากับคาบการสุ่มสัญญาณของระบบการวัด และสูงสุดเท่ากับ 2 เท่าของคาบการสุ่มสัญญาณ

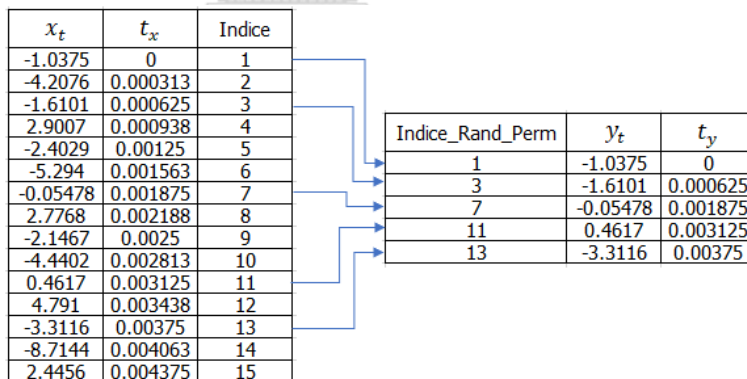
2. การสุ่มหยิบจุดข้อมูลจากสัญญาณการวัดด้วยความถี่สุ่มคงที่

การสุ่มหยิบสัญญาณในกรณีนี้ จำลองสถานการณ์การสุ่มดึงจุดข้อมูลจากสัญญาณ (t_x, x_t) ที่วัดด้วยความถี่สุ่มคงที่ f_{rate} และเป็นไปตามเกณฑ์ในควิสต์ ข้อมูลที่สุ่มดึงออกมาคือสัญญาณ (t_y, y_t) โดยกำหนด 2 เงื่อนไขด้วยกันซึ่งสามารถเขียนในรูปแบบ ข้อมูลแบบบูลีนได้ดังนี้

$$[\Delta indices > x] \cup [rand > 0.5]$$

โดยเงื่อนไขทั้ง 2 สามารถอธิบายได้ดังนี้

1. ให้จุดข้อมูลที่สุ่มหยิบต้องห่างจากจุดที่หยิบไปก่อนหน้านี้อย่างน้อย x ตำแหน่ง ส่งผลให้ระยะห่างของคาบที่น้อยที่สุดระหว่างจุดข้อมูลเท่ากับ $\frac{x}{f_{rate}}$
2. มีโอกาส 50% ที่จุดนั้นๆ จะถูกหยิบขึ้นมา



รูปที่ 3.1 แสดงการเลือกจุดจากตำแหน่งของข้อมูลของสัญญาณ x_t เพื่อใช้เป็นสัญญาณ y_t

จากตัวอย่างในรูปที่ 3.1 ตารางด้านซ้ายแสดงค่าที่วัดด้วยความถี่สุ่มคงที่ ส่วนด้านขวาแสดงการสุ่มหยิบข้อมูลเพื่อสร้างข้อมูล (t_y, y_t) เพื่อให้คำนวณคอมเพรชชันเชิง ในสัญญาณปกติ x_t ทางด้านซ้ายนั้นระยะห่างที่น้อยที่สุดคือ $\Delta indices = 1$ หรือเทียบเท่ากับคาบของการสุ่มสัญญาณด้วยความถี่ f_{rate} 3200 เฮิรตซ์ ในขณะที่การสุ่มหยิบข้อมูลทางด้านขวา ในคอลัมน์ $Index_Rand_Perm$ ระยะห่างของจุดที่น้อยที่สุดคือระหว่างจุดที่ 1 และจุดที่ 3 ซึ่งมีค่า $\Delta indices = 3 - 1 = 2$ หมายความว่า การสุ่มจุด y_t นั้น

เทียบเท่ากับสัญญาณที่ถูกวัดแบบสุ่มโดยตรงด้วยเซ็นเซอร์ที่มีอัตราสุ่มจุดสูงสุดเท่ากับ $\frac{frate}{2}$ หรือเท่ากับ 1600 เฮิร์ตซ์

การกำหนดเงื่อนไขว่า $\Delta indices$ จะต้องมากกว่าค่าเท่าไรเป็นอย่างน้อยจึงเสมือนเป็นการจำลองอัตราสุ่มจุดของเซ็นเซอร์ที่ใช้วัดว่ามีค่าเท่าไร เช่น $\Delta indices \geq 2$ หมายความว่า อัตราสุ่มจุดเพื่อให้ได้สัญญาณมีค่าสูงสุดคือ $\frac{frate}{2}$ หรือ 1600 เฮิร์ตซ์ ถ้า $\Delta indices \geq 5$ ก็จะได้เซ็นเซอร์อัตราสุ่มจุดเท่ากับ $\frac{frate}{5}$ หรือ 640 เฮิร์ตซ์

3. การวัดสัญญาณโดยการใช้ความถี่สุ่มไม่คงที่

วิธีการนี้สอดคล้องกับสถานการณ์ที่สามารถปรับให้การวัดสัญญาณวัดในเวลาที่สูงขึ้นมาได้โดยตรง โดยการควบคุมบอร์ดไมโครคอนโทรลเลอร์ให้อ่านค่าสัญญาณการสั่นสะเทือนจากเซ็นเซอร์ในเวลาสุ่มที่กำหนด ทำให้ระยะห่างของจุดข้อมูลมีค่าไม่คงที่ หรืออาจกล่าวว่าเป็นการวัดด้วยความถี่สุ่มไม่คงที่ ค่าที่วัดออกมาได้จะเป็นค่า y_t ทั้งนี้ สามารถดูรายละเอียดเพิ่มเติมได้ในภาคผนวก ค. และ ง.

3.1.2 การจับคู่เวลา (Time Matching) และเมทริกซ์การวัด (Measurement matrix)

จากที่กล่าวไปแล้วว่าสัญญาณการสั่นสะเทือนมีความสปร้าบนโดเมนความถี่ซึ่งโดยปกติจะได้มาจากการวัดแบบปกติที่มีระยะห่างของจุดข้อมูลเท่าๆ กัน ซึ่งสามารถแปลงโดเมนได้ด้วยสมการ $x_t = \psi x_s$ แต่สำหรับคอมพิวเตอร์ซึ่งที่ใช้การวัดแบบสุ่มค่าเข้ามาใช้ สัญญาณที่ได้จะไม่ใช่ x_t, t_x แต่จะเป็น y_t, t_y ซึ่งมีระยะห่างระหว่างจุดของเวลาไม่เท่ากัน ดังนั้นสมการที่ใช้ในการแปลงโดเมนเวลาให้อยู่ในโดเมนความถี่จะไม่สามารถใช้แบบข้างต้นได้ จะต้องถูกเปลี่ยนเป็น $y_t = C x_t = C \psi x_s$ ซึ่งมีเมทริกซ์การวัด C เพิ่มเข้ามาช่วยให้สามารถแปลงโดเมนได้ตามปกติ

เพื่อให้ได้ เมทริกซ์การวัด C ของสมการที่ (2.14) จะต้องนำ t_y ที่มีขนาด $[M \times 1]$ นำไปจับคู่กับ t_x ที่มีขนาด $[N \times 1]$ รูปที่ 3.2 แสดงการสร้างสัญญาณความถี่ 1 เฮิร์ตซ์ ขนาด 7 จุด จากข้อมูลสัญญาณแบบสุ่ม (t_y) 5 จุดโดยการจับคู่เวลาที่มีความใกล้เคียงกันมากที่สุดในแต่ละตำแหน่ง ให้มีค่าเท่ากับ 1 ส่วนในตำแหน่งที่เหลือก็จะให้ข้อมูลตรงจุดนั้นเท่ากับศูนย์ จะสังเกตได้ว่าการจับคูเวเวลานั้นไม่สามารถจับคู่ได้พอดีเนื่องจากในสัญญาณปกติขนาด 7 จุดนั้นมีระยะห่างเท่ากันหมดคือ 1 วินาที ทำให้มีเวลาในแต่ละจุดแน่นอน แต่สำหรับสัญญาณที่ทำการสุ่มขึ้นมา (y_t) เวลาที่ได้จะเกิดจากการสุ่ม (t_y) ซึ่งไม่พอดีกับเวลาคาบของสัญญาณที่เราจะนำไปใช้ ค่าความคลาดเคลื่อนนี้เป็นข้อจำกัดสำคัญสำหรับการใช้คอมพิวเตอร์ซึ่งกับสัญญาณที่มีลักษณะเป็นจุดของเวลาส่งผลให้สัญญาณที่ถูกสร้างใหม่ขึ้นมามีความแม่นยำลดลงเมื่อเทียบกับสัญญาณประเภทอื่นๆ ที่ตำแหน่งของข้อมูลมีความแน่นอน เช่น สัญญาณภาพ

t_y	t_x		C							
0	0	1	1	0	0	0	0	0	0	0
1.2	1	1	0	1	0	0	0	0	0	0
2.7	2	0	0	0	0	0	0	0	0	0
4.1	3	1	0	0	0	1	0	0	0	0
5.7	4	1	0	0	0	0	1	0	0	0
	5	0	0	0	0	0	0	0	0	0
	6	1	0	0	0	0	0	0	0	1

รูปที่ 3.2 การจับคู่เวลาของสัญญาณเพื่อสร้าง แมตริกซ์การวัด C

จากรูปที่ 3.2 สามารถอธิบายเพิ่มเติมในรูปแบบสมการได้ดังนี้ ในกรณีสัญญาณที่วัดออกมา y_t เป็นสัญญาณภาพหรือเป็นสัญญาณที่มีตำแหน่งที่ถูกตรวจวัดเป็นตำแหน่งเดียวกับสัญญาณ x_t ในกรณีนี้จะไม่มีความคลาดเคลื่อนของสัญญาณเหมือนกับสัญญาณการสั่นสะเทือนที่กล่าวไปในย่อหน้าก่อนหน้า สมการที่ได้จะมีรูปแบบ $y_t = Cx_t$ แต่สำหรับสัญญาณการสั่นสะเทือนที่มีความคลาดเคลื่อนของสัญญาณ x_t และ y_t สมการที่ได้จะอยู่ในรูป $y_t = Cx_t + \epsilon$ โดยที่ ϵ คือความคลาดเคลื่อนที่เกิดจากการจับคู่เวลา (Time Matching error) เมื่อเวลาคลาดเคลื่อนก็ส่งผลให้แอมพลิจูดคลาดเคลื่อนไปด้วย แต่ในกรณีนี้จะประมาณ $y_t \approx Cx_t$ เพื่อให้ง่ายแก่การแก้สมการต่อไป

จากการจับคู่ของเวลาจะทำให้ทราบตำแหน่งของจุดที่มีข้อมูลในแต่ละเวลาซึ่งก็คือ แมตริกซ์การวัด (Measurement Matrix) C ซึ่งจะเป็นแมตริกซ์ที่แยงมุมโดยตำแหน่งในแนวทแยงจะถูกแทนด้วย 1 ในตำแหน่งที่มีข้อมูล และ 0 ในตำแหน่งที่ไม่มีข้อมูล

สำหรับการทดสอบที่ใช้วิธีการสุ่มหยิบจุดจากข้อมูลที่วัดแบบปกติจะไม่จำเป็นต้องจับคู่เวลา เนื่องจากจะได้ข้อมูลตำแหน่งจุดเวลาที่จากเงื่อนไขการสุ่มหยิบเรียบร้อยแล้ว และสามารถนำตำแหน่งที่ได้มาไปใช้สร้างแมตริกซ์ฐานหลักเริ่มต้นได้เลย

3.1.3 การสร้างแมตริกซ์ฐานหลักเริ่มต้น (Initial basis)

ในหัวข้อนี้จะกล่าวถึงการได้มาซึ่ง θ หรือเมตริกซ์ฐานหลักเริ่มต้น (Initial basis) ซึ่งหาได้จาก $C\psi = \theta$ โดย ψ คือฐานหลักทั่วไป (General basis) เช่น แมตริกซ์การแปลงฟูเรียร์ แมตริกซ์การแปลงโคไซน์ หรือสามารถใช้เป็นแมตริกซ์การแปลงแบบอื่นก็ได้เช่นกัน ขึ้นอยู่กับงานที่จะนำไปประยุกต์ ในงานนี้จะใช้แมตริกซ์การแปลงโคไซน์

จากตัวอย่างในรูปที่ 3.2 และสมการ $\theta = C\psi$ เมื่อแทนค่า ของแมตริกซ์ C และแมตริกซ์การแปลงโคไซน์ ψ จะหาฐานหลักเริ่มต้น θ ได้ตามสมการด้านล่าง

$$\theta = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \sqrt{\frac{2}{N}} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \cos \frac{\pi}{2N} & \cos \frac{3\pi}{2N} & \cos \frac{5\pi}{2N} & \cos \frac{7\pi}{2N} & \cos \frac{9\pi}{2N} & \cos \frac{11\pi}{2N} & \cos \frac{13\pi}{2N} \\ \cos \frac{2\pi}{2N} & \cos \frac{6\pi}{2N} & \cos \frac{10\pi}{2N} & \cos \frac{14\pi}{2N} & \cos \frac{18\pi}{2N} & \cos \frac{22\pi}{2N} & \cos \frac{26\pi}{2N} \\ \cos \frac{3\pi}{2N} & \cos \frac{9\pi}{2N} & \cos \frac{15\pi}{2N} & \cos \frac{21\pi}{2N} & \cos \frac{27\pi}{2N} & \cos \frac{33\pi}{2N} & \cos \frac{39\pi}{2N} \\ \cos \frac{4\pi}{2N} & \cos \frac{12\pi}{2N} & \cos \frac{20\pi}{2N} & \cos \frac{28\pi}{2N} & \cos \frac{36\pi}{2N} & \cos \frac{44\pi}{2N} & \cos \frac{52\pi}{2N} \\ \cos \frac{5\pi}{2N} & \cos \frac{15\pi}{2N} & \cos \frac{25\pi}{2N} & \cos \frac{35\pi}{2N} & \cos \frac{45\pi}{2N} & \cos \frac{55\pi}{2N} & \cos \frac{65\pi}{2N} \\ \cos \frac{6\pi}{2N} & \cos \frac{18\pi}{2N} & \cos \frac{30\pi}{2N} & \cos \frac{42\pi}{2N} & \cos \frac{54\pi}{2N} & \cos \frac{66\pi}{2N} & \cos \frac{78\pi}{2N} \end{bmatrix}$$

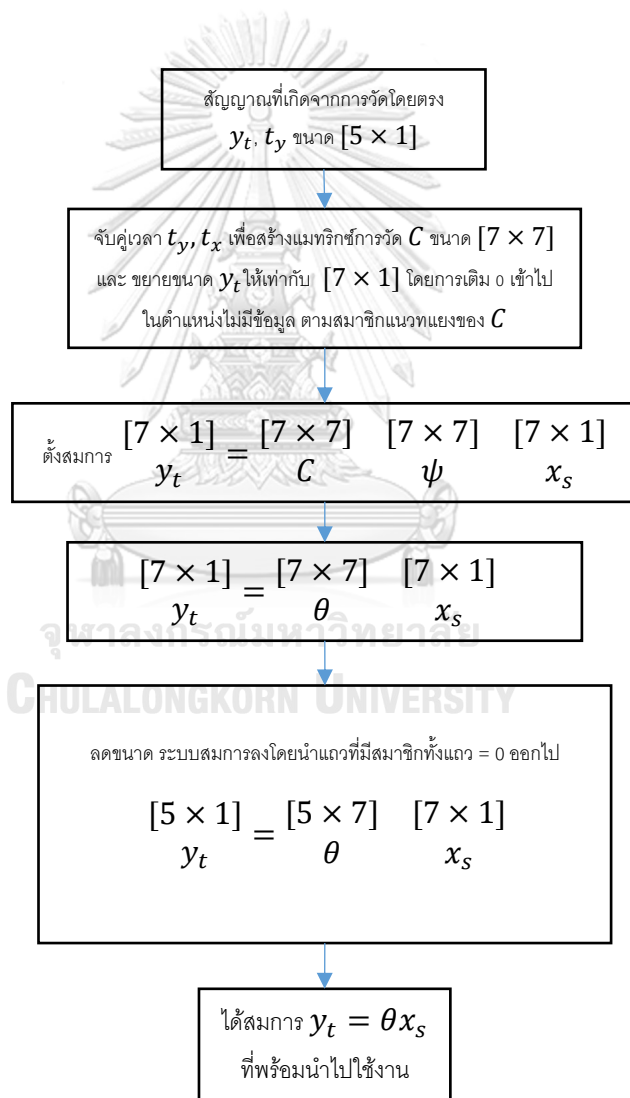
$$\theta = \sqrt{\frac{2}{N}} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \cos \frac{\pi}{2N} & \cos \frac{3\pi}{2N} & \cos \frac{5\pi}{2N} & \cos \frac{7\pi}{2N} & \cos \frac{9\pi}{2N} & \cos \frac{11\pi}{2N} & \cos \frac{13\pi}{2N} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \cos \frac{3\pi}{2N} & \cos \frac{9\pi}{2N} & \cos \frac{15\pi}{2N} & \cos \frac{21\pi}{2N} & \cos \frac{27\pi}{2N} & \cos \frac{33\pi}{2N} & \cos \frac{39\pi}{2N} \\ \cos \frac{4\pi}{2N} & \cos \frac{12\pi}{2N} & \cos \frac{20\pi}{2N} & \cos \frac{28\pi}{2N} & \cos \frac{36\pi}{2N} & \cos \frac{44\pi}{2N} & \cos \frac{52\pi}{2N} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \cos \frac{6\pi}{2N} & \cos \frac{18\pi}{2N} & \cos \frac{30\pi}{2N} & \cos \frac{42\pi}{2N} & \cos \frac{54\pi}{2N} & \cos \frac{66\pi}{2N} & \cos \frac{78\pi}{2N} \end{bmatrix}$$

ผลคูณกันระหว่าง C กับ ψ ทำให้ได้ขนาดของเมทริกซ์ θ เท่ากับ $[7 \times 7]$ แต่สังเกตว่า θ มีสมาชิกในแถวของเมทริกซ์บางแถวเท่ากับ 0 อยู่ ซึ่งหากนำไปประกอบกับสมการที่ (2.10) $y_t = \theta x_s$ จะเห็นว่าถ้ากำหนดให้ x_s มีขนาด $[7 \times 1]$ หลังจากที่ได้คูณกับ θ แล้ว สัญญาณ y_t ที่ได้ก็จะมีขนาด $[7 \times 1]$ เช่นกัน แต่จะมีสมาชิกที่ไม่เท่ากับศูนย์อยู่เพียง 5 ตัว ตามจำนวนแถวของ θ ที่ไม่เท่ากับ 0 หากตัดแถวที่มีค่าเท่ากับ 0 ทั้งแถวออก เมทริกซ์ θ จะถูกลดขนาดลงได้ดังสมการที่ 2.17 และจะทำให้กลายเป็นระบบสมการไม่ชัดเจนและสามารถนำไปหาค่า x_s ได้ตามวิธีที่กล่าวถึงไปแล้วใน บทที่ 2

$$\theta = \sqrt{\frac{2}{N}} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \cos \frac{\pi}{2N} & \cos \frac{3\pi}{2N} & \cos \frac{5\pi}{2N} & \cos \frac{7\pi}{2N} & \cos \frac{9\pi}{2N} & \cos \frac{11\pi}{2N} & \cos \frac{13\pi}{2N} \\ \cos \frac{3\pi}{2N} & \cos \frac{9\pi}{2N} & \cos \frac{15\pi}{2N} & \cos \frac{21\pi}{2N} & \cos \frac{27\pi}{2N} & \cos \frac{33\pi}{2N} & \cos \frac{39\pi}{2N} \\ \cos \frac{4\pi}{2N} & \cos \frac{12\pi}{2N} & \cos \frac{20\pi}{2N} & \cos \frac{28\pi}{2N} & \cos \frac{36\pi}{2N} & \cos \frac{44\pi}{2N} & \cos \frac{52\pi}{2N} \\ \cos \frac{6\pi}{2N} & \cos \frac{18\pi}{2N} & \cos \frac{30\pi}{2N} & \cos \frac{42\pi}{2N} & \cos \frac{54\pi}{2N} & \cos \frac{66\pi}{2N} & \cos \frac{78\pi}{2N} \end{bmatrix}$$

เมื่อผ่านขั้นตอนการจับคู่เวลาและสร้างฐานหลักเริ่มต้นเรียบร้อยแล้ว สมการที่ (2.10) $y_t = C\psi x_s = \theta x_s$ ในตอนนี้ก็จะมีเมทริกซ์ตัวแปรไม่ทราบค่าเพียงแค่เมทริกซ์เดียวคือ x_s จึงสามารถแก้ระบบสมการเชิงเส้นไม่ชัดเจน $y_t = \theta x_s$ เพื่อหาคำตอบ x_s ได้ โดยหลังจากนี้จะอธิบายการแก้ระบบสมการเชิงเส้นไม่ชัดเจนด้วยวิธีต่างๆ เพื่อหาสัญญาณสปาร์สต่อไป

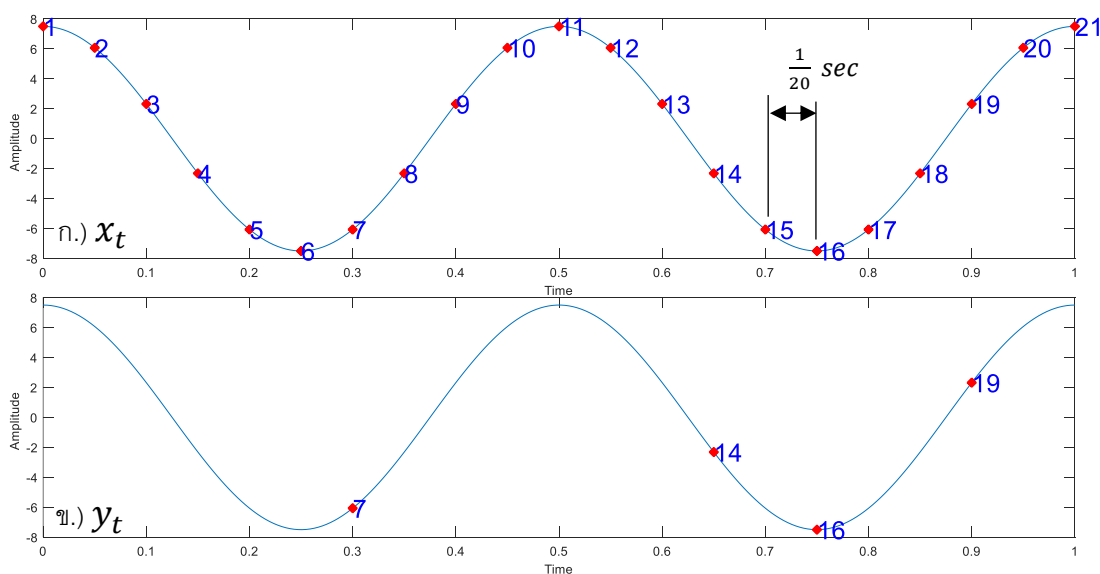
ในรูปที่ 3.3 เป็นแผนผังในการสร้างระบบสมการ $y_t = \theta x_s$ ที่จะนำไปใช้งานในคอมพิวเตอร์ซึ่งโดยนำตัวอย่างจากรูปที่ 3.3 เมื่อวัดสัญญาณ y_t, t_y โดยตรง มาอธิบายเป็นลำดับ ซึ่งจะเห็นว่าจะมีขั้นตอนการจับคู่เวลาเพื่อสร้างเมทริกซ์ C และการลดจำนวนแถวที่เท่ากับ 0 ของเมทริกซ์ θ และ y_t ลงเพื่อให้ระบบสมการกลายเป็นระบบสมการไม่ชัดเจนตามเงื่อนไขที่คอมพิวเตอร์ต้องการ



รูปที่ 3.3 ลำดับวิธีการสร้างระบบสมการไม่ชัดเจน ด้วยการจับคู่เวลาและการเพิ่ม/ลด ขนาดเมทริกซ์

3.2 พารามิเตอร์ของสัญญาณแบบสุ่มคาบ

ในหัวข้อนี้จะอธิบายถึงพารามิเตอร์ที่สำคัญของการเก็บข้อมูลแบบสุ่มคาบ y_t เนื่องจากสัญญาณ y_t มีระยะห่างระหว่างจุดข้อมูลไม่คงที่ จึงไม่สามารถหา ความถี่สุ่มหรือ f_{rate} ที่แท้จริงได้ซึ่งพารามิเตอร์ที่สามารถวัดหรือตรวจสอบได้จะถูกแบ่งออกเป็น 3 พารามิเตอร์ โดยจะอธิบายด้วยการใช้รูปที่ 3.4 ประกอบ โดยกำหนดให้สัญญาณ x_t เป็นสัญญาณจากการวัดปกติที่มีความถี่สุ่ม 20 เฮิร์ตซ์ y_t เป็นสัญญาณแบบสุ่มคาบที่เกิดจากการสุ่มหยิบสัญญาณ x_t



รูปที่ 3.4 แสดงตัวอย่างของสัญญาณ ก.) x_t ข.) y_t

- จำนวนจุดข้อมูลทั้งหมดใน x วินาทีที่ต้องการทราบ $N_{y,x}$
 จากรูปที่ 3.4(ข.) จำนวนจุดข้อมูล y_t ใน 1 วินาที คือ 4 จุดหรือเขียนได้เป็น $N_{y,1} = 4$ สำหรับงานวิจัยนี้สัญญาณ y_t ที่ใช้สำหรับคอมพิวเตอร์ซึ่งทั้งหมดจะมีความยาวเท่ากับ 1 วินาทีเท่านั้น ในกรณีที่ไม่ได้ระบุเพิ่มเติมว่า y_t มีความยาวเท่าไร
- ความถี่สุ่มเทียบเท่าหรืออัตราการสุ่มจุดเทียบเท่า f_{erate} (Equivalent Sampling Rate)
 ค่าพารามิเตอร์นี้บ่งบอกว่า การเก็บสัญญาณ y_t จะต้องใช้เซ็นเซอร์ที่มีความถี่สุ่มสูงสุดเท่าไรจึงจะสามารถวัดสัญญาณนี้ออกมาได้ ค่านี้จะได้มาจากการวัดระยะห่างของคาบที่แคบที่สุดในสัญญาณ y_t โดยในตัวอย่าง รูปที่ 3.4(ข.) จะเห็นว่าคาบที่แคบที่สุดของสัญญาณนี้คือ จุดที่ 14 และ 16 ซึ่งห่างกัน $\frac{1}{20} + \frac{1}{20} = 0.1$ วินาที หมายความว่า เซ็นเซอร์ที่สามารถสร้างหรือวัดสัญญาณนี้ออกมาต้องมีความถี่สุ่มอย่างน้อย $\frac{1}{0.1} = 10$ เฮิร์ตซ์ ฉะนั้น สัญญาณ y_t ในรูปที่ 3.4(ข.) มีความถี่สุ่มเทียบเท่า 10 เฮิร์ตซ์
- พิสัยการสุ่ม f_{range}

พิสัยการสุ่มหาได้จากผลต่างของความถี่สุ่มสูงสุดกับความถี่สุ่มต่ำสุดของ สัญญาณ y_t พิสัยการสุ่มเกิดจากการสุ่มคาบไม่คงที่ซึ่งมีระยะของคาบที่แคบที่สุด และคาบที่กว้างที่สุด พารามิเตอร์ตัวนี้จะเป็นตัวบ่งบอกว่าสัญญาณ y_t นั้นมีการสุ่มหลากหลายมากเพียงใด ซึ่งถ้าค่านี้มีค่าน้อย สัญญาณนั้นก็จะมีความเป็น Periodic มากขึ้น ในทางกลับกันถ้ามีค่ามากขึ้นสัญญาณก็จะมีความเป็น Periodic น้อยลง โดยตัวอย่างสัญญาณ y_t ในรูปที่ 3.4(ข.) จะเห็นว่า คาบที่แคบที่สุดคือระหว่างจุดที่ 14 และ 16 หรือ 0.1 วินาทีคาบที่กว้างที่สุดคือระหว่าง จุดที่ 7 และ 14 หรือ 0.7 วินาที ฉะนั้นพิสัยการสุ่มของสัญญาณ ในรูปที่ 3.4(ข.) คือ $\frac{1}{0.1} - \frac{1}{0.7} = 8.57$ เฮิร์ตซ์ สำหรับค่าพิสัยการสุ่มนั้นโดยปกติจะแปรผกผันกับจำนวนจุดใน 1 หน่วยวินาทีของสัญญาณ y_t ถ้าพิสัยการสุ่มมีค่ามาก จำนวนจุดข้อมูลใน 1 วินาทีก็จะมีจำนวนลดลงไปด้วย

พารามิเตอร์ ทั้ง 3 นี้มีความสำคัญใช้สำหรับการควบคุมการสุ่มเพื่อสร้างสัญญาณ y_t ให้อยู่ในเกณฑ์ที่ต้องการ

3.3 การสร้างสัญญาณสปาร์สโดยวิธีคอมเพรสชัน

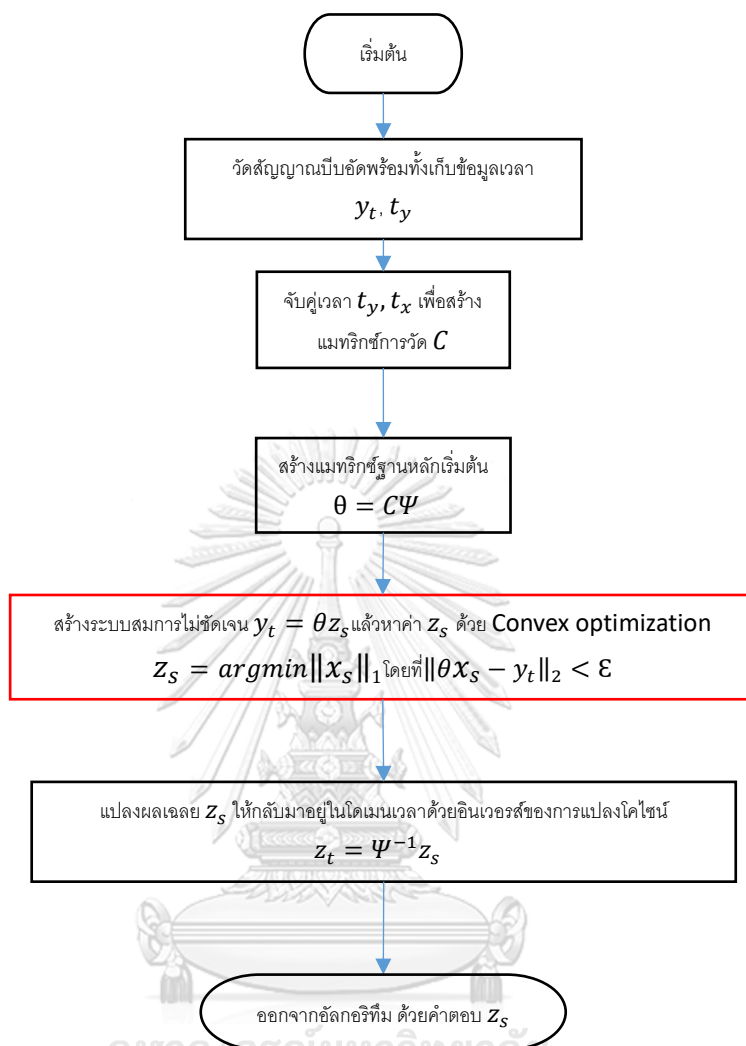
ในปัจจุบันอัลกอริทึมที่สามารถแกะระบบสมการไม่ชัดเจนในปัญหาคอมเพรสชันนั้นมีหลายหลายอัลกอริทึม แต่ในงานวิจัยนี้จะใช้ทั้งหมด 3 แบบด้วยกัน เลือกจากวิธีที่นิยมใช้กันในปัจจุบัน 2 วิธีคือ การหาค่าเหมาะสมที่สุดแบบคอนเวกซ์ (Convex Optimization) และ การจับคู่สัญญาณบีบอัด (Compressed Sampling Matching Pursuit หรือ COSAMP) และอีกวิธีที่ผู้วิจัยเสนอขึ้นโดยปรับปรุงจากวิธีการจับคู่สัญญาณบีบอัดก็คือการช้อนสัญญาณตัวแทนเฉลี่ย (Mean Stack Proxy)

3.3.1 การสร้างสัญญาณสปาร์สโดยการใช้ค่าเหมาะสมที่สุดแบบคอนเวกซ์

วิธีการนี้เป็นวิธีการแก้ระบบสมการ $y_t = C\psi x_s = \theta x_s$ โดยตรง โดยได้กล่าวไปแล้วในจากหัวข้อ 2.4.1 ซึ่งวิธีการนี้จะถูกนำมาใช้กับการหาสัญญาณคืนกลับ Z_S ด้วยการวัดสัญญาณการสั่นสะเทือน y_t ที่ได้จากการวัดสัญญาณแบบสุ่มคาบ แมทริกซ์ C ได้จากการจับคู่เวลา และแมทริกซ์การแปลงโคไซน์ ψ สำหรับคำตอบของระบบสมการ $y_t = \theta x_s$ อย่างไรก็ตามสัญญาณที่ถูกสร้างขึ้นมา จะเป็นเพียงการประมาณซึ่งใกล้เคียง x_s เท่านั้นแต่ไม่ใช่สัญญาณ x_s คำตอบของสัญญาณที่ได้จะถูกเขียนเป็น สัญญาณสปาร์ส Z_S ในโดเมนความถี่โดยที่ $Z_S \approx x_s$

$$z_s = \underset{x_s}{\operatorname{argmin}} \|x_s\|_1 \text{ โดยที่ } \|\theta x_s - y_t\|_2 < \varepsilon \quad (3.2)$$

วิธีนี้เป็นวิธีแก้ปัญหา ระบบสมการเชิงเส้นไม่ชัดเจนแบบโดยตรง โดยใช้อัลกอริทึมสำหรับการหาค่าเหมาะสมที่สุดทำให้ความยุ่งยากในการใช้งานน้อย ให้คำตอบที่มีรายละเอียดสูง แต่อัลกอริทึมชนิดนี้มีข้อจำกัดคือเวลาในการประมวลผลที่มาก ขั้นตอนการคำนวณของวิธีการนี้แสดงในรูปที่ 3.5



รูปที่ 3.5 อัลกอริทึม คอมเพรสเซนซิงโดยการใช้ค่าเหมาะสมที่สุดแบบคอนเวกซ์

จากรูปที่ 3.5 จะเห็นว่าค่าที่จำเป็นต้องวัดสำหรับคอมเพรสเซนซิงมีเพียง 2 ค่าเท่านั้นคือ y_t และ t_y ซึ่งแต่ละตัวมีสมาชิกขนาด $[M \times 1]$ โดยที่ $M < N$ แต่สามารถนำไปแก้สมการเพื่อหาค่าของ z_s ขนาด $[N \times 1]$ ได้ สำหรับวิธีการอื่นๆที่จะกล่าวถึงต่อไปก็จะมีคล้ายคลึงกันโดยจะแตกต่างการเพียงวิธีการแก้สมการที่ (2.10) หรือ $y_t = \theta z_s$ ในกรอบสีแดงเท่านั้น

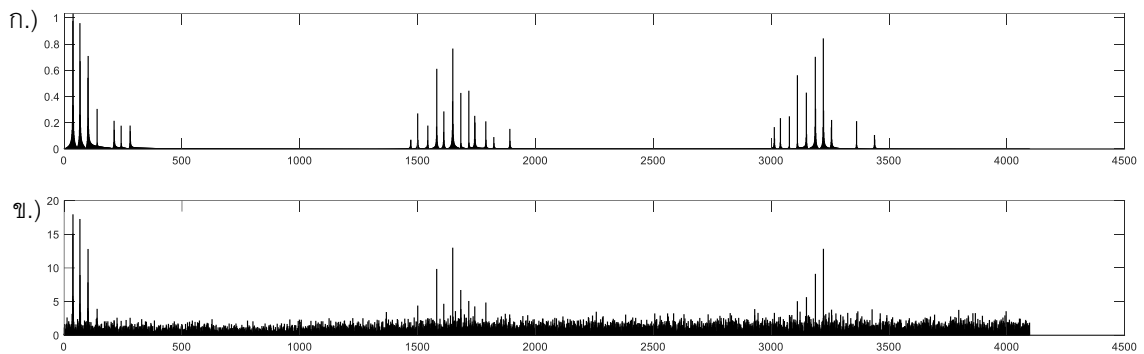
3.3.2 การสร้างสัญญาณสปาร์สโดยการจับคู่สัญญาณบีบอัด (Compressed sampling Matching Pursuit COSAMP)

สำหรับวิธีการหาค่าที่เหมาะสมที่สุดแบบคอนเวกส์ถึงแม้ว่าจะให้ความถูกต้องสูง แต่ก็มีจุดอ่อนคือใช้เวลาในการคำนวณมาก ถ้าหากว่าเวกเตอร์ x_s หรือ y_t มีขนาดใหญ่ก็จะยิ่งใช้เวลาคำนวณมากขึ้น สำหรับการจับคู่สัญญาณบีบอัดจะไม่ได้พยายามแก้สมการ $y_t = C\psi x_s = \theta x_s$ และหาค่าที่เหมาะสมที่สุดเพื่อให้ได้สัญญาณคืนกลับ เหมือนวิธีก่อนหน้านี้ แต่จะเลือกแก้เฉพาะพจน์ของสมการย่อยที่ผ่านการเลือกแล้วว่ามีค่าสำคัญที่สุด K พจน์เท่านั้น สำหรับวิธีการเลือกเบื้องต้นจะแก้สมการ $y_t = \theta x_s$ ใช้ มัวร์-เพนโรสอินเวอร์สทำให้ได้ค่า $v_s = \theta^\dagger y_t$ ซึ่งค่า v_s ที่ได้จะถูกเรียกว่าสัญญาณตัวแทน (Proxy Signal) v_s จะถูกนำไปเลือกตำแหน่งของความถี่ที่มีขนาดสูงที่สุด K อันดับแรก ซึ่งจะมีจำนวน K ความถี่ เมื่อได้ตำแหน่งความถี่ที่ต้องการแล้วก็จะสามารถปรับลดจำนวนพจน์ในระบบสมการหรือคอลัมน์บนเมทริกซ์ θ ให้สอดคล้องกับความถี่ที่เลือกมาได้ ทำให้สามารถเลือกแก้สมการเฉพาะตำแหน่งความถี่ที่ต้องการได้ และจะได้คำตอบเป็นสัญญาณ Z_s ที่มีจำนวนความถี่ที่แอมพลิจูดไม่เท่ากับ 0 อยู่ K ความถี่เท่านั้น โดยวิธีการคำนวณแบบละเอียดจะถูกอธิบายเพิ่มเติม ในหัวข้อขั้นตอนการทำงานของอัลกอริทึม

3.3.2.1 การสร้างสัญญาณตัวแทน(Proxy Signal)

สัญญาณตัวแทน (Proxy Signal) ที่ถูกสร้างขึ้นมานั้นไม่เทียบเท่ากับ x_s ซึ่งเป็นคำตอบของสมการ $y_t = \theta x_s$ เนื่องจากระบบนี้สมการนี้เป็นระบบสมการไม่ชัดเจนและมีจำนวนคำตอบที่ถูกต้องเป็นอนันต์หมายความว่าคำตอบของสมการไม่ได้มีเพียงแค่ x_s เท่านั้นที่เป็นคำตอบที่ถูกต้อง สำหรับสัญญาณตัวแทนนั้นจะถูกสร้างขึ้น โดยใช้ Moore–Penrose inverse จะให้คำตอบที่เป็นแบบกำลังสองน้อยที่สุด (Least square solution) ฉะนั้น สัญญาณที่ได้จึงเป็นสัญญาณที่มีความคล้ายคลึงกับ x_s แต่ไม่ใช่ x_s เนื่องจาก x_s จะเป็นคำตอบแบบสปาร์ส(Sparse solution) ซึ่งหาได้จากการทำ Optimization แต่เนื่องจากวิธีการนี้ไม่ได้ใช้การ Optimization จึงใช้เวลาในการคำนวณสั้นกว่ามากเมื่อเทียบกับ การหา x_s

รูปที่ 3.6ก.) เป็นตัวอย่างของสัญญาณ x_s ที่ถูกแปลงแบบปกติจากสัญญาณ x_t ที่มีจุดข้อมูลทั้งหมด 8192 จุด รูปที่ 3.6ข.เป็นตัวอย่างของสัญญาณตัวแทน v_s ที่ถูกแปลงจากสัญญาณ y_t ที่มีจุดข้อมูลทั้งหมด 1030 จุดโดยเมทริกซ์การวัดของทั้งสองสัญญาณนี้สร้างจาก เมทริกซ์ของการแปลงโคไซน์ไม่ต่อเนื่อง



รูปที่ 3.6 ก.) สัญญาณทดสอบจากข้อมูลความถี่ 8192 เฮิรตซ์

ข.) สัญญาณตัวแทนจากสัญญาณเวลา 1030 จุดวินาที

จะเห็นว่าสัญญาณทั้งสองชุดนี้มีความคล้ายคลึงกันบางส่วน โดย รูปที่ 3.6(ก.) คือสัญญาณที่รายละเอียดครบถ้วน ส่วนรูปที่ 3.6(ข.) เป็นสัญญาณตัวแทน สร้างจากการวัดแบบสุ่มซึ่งมีขนาดข้อมูลน้อยกว่าปกติ ทำให้เกิดสัญญาณรบกวน ส่วนที่สัญญาณตัวแทนมีลักษณะที่คล้ายคลึงกับสัญญาณจริงคือตำแหน่งความถี่ที่มีแอมพลิจูดมาก แต่ในส่วนแอมพลิจูดต่ำนั้นยังไม่สามารถจะแยกแยะระหว่างสัญญาณรบกวนและสัญญาณหลักได้

จากตัวอย่างนี้สัญญาณตัวแทน v_S ไม่ใช่ตัวแทนที่ดีของสัญญาณ x_S แต่ความถี่ที่มีขนาดของแอมพลิจูดมากของสัญญาณ v_S ยังคงสอดคล้องกับสัญญาณ x_S ดังนั้นวิธีการจับคู่สัญญาณบีบอัดจะใช้วิธีการสร้างสัญญาณคืนกลับซึ่งเป็นสัญญาณสปาร์สโดยพิจารณาว่า สัญญาณคืนกลับควรจะมีข้อมูลที่มีค่าไม่เท่ากับศูนย์ เฉพาะที่ตำแหน่งความถี่ที่มีแอมพลิจูดสูงที่สุด K อันดับแรกเท่านั้น ดังนั้นวิธีการนี้จึงเลือกใช้เฉพาะความถี่ที่มีแอมพลิจูดสูงสุด K อันดับแรกของสัญญาณ v_S มาใช้ในการคำนวณต่อ

โดย K จะมีค่าน้อยกว่าจำนวนจุดที่สุ่มวัดข้อมูล M และน้อยกว่า N ซึ่งเป็นจำนวนข้อมูลของ x_S ที่ต้องการมาก

พิจารณาสมการพื้นฐานของคอมเพรชชันซึ่ง

$$[M \times 1] = [M \times N][N \times 1]$$

$$y_t = \theta x_s$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} \theta_{11} & \cdots & \theta_{1N} \\ \vdots & \ddots & \vdots \\ \theta_{M1} & \cdots & \theta_{MN} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \\ \vdots \\ x_N \end{bmatrix}$$

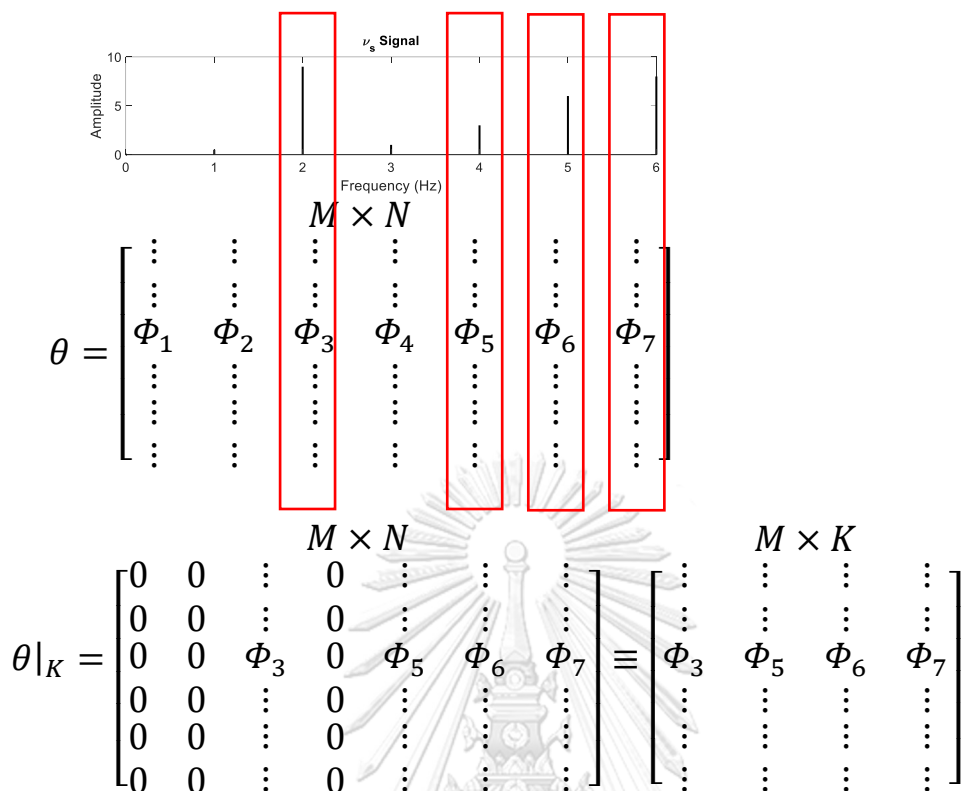
เมื่อย้ายข้าง θ ไปทางซ้ายมือจะได้

$$[N \times M][M \times 1] = [N \times 1]$$

$$\theta^\dagger y_t = v_s$$

$$\begin{bmatrix} \theta_{11} & \cdots & \theta_{1N} \\ \vdots & \ddots & \vdots \\ \theta_{M1} & \cdots & \theta_{MN} \end{bmatrix}^\dagger \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_M \\ \vdots \\ v_N \end{bmatrix}$$

เมื่อได้ สัญญาณตัวแทน v_s แล้วก็จะสามารถนำไปใช้ลดขนาดเมทริกซ์ θ จากขนาด $[M \times N]$ ให้มีขนาดเล็กลงกลายเป็น $\theta|_K$ ขนาด $[M \times K]$ ได้โดยผ่านการเลือกคอลัมน์ในเมทริกซ์ θ เฉพาะคอลัมน์ที่สัมพันธ์กับความถี่ของสัญญาณ v_s ที่มีขนาดสัญญาณสูงสุด K อันดับแรก ทั้งนี้เนื่องจากสัญญาณ v_s เป็นสัญญาณสปาร์สซึ่งมีความถี่ที่มีความสำคัญจำนวนน้อย การเลือกเช่นนี้จึงเป็นการพิจารณาเฉพาะลักษณะเด่นของสัญญาณตัวแทนเท่านั้น และทำให้ขนาดของเมทริกซ์ θ ลดลงเป็น $[M \times K]$ ซึ่ง $K \ll N$ ได้



รูปที่ 3.7 ตำแหน่งใน θ ที่สัมพันธ์กับแอมพลิจูดสูงสุด K อันดับแรกของสัญญาณ v_s และการลดขนาดเมทริกซ์ θ

รูปที่ 3.7 แสดงตัวอย่างการเลือกตำแหน่งในโดเมนความถี่ของสัญญาณ v_s ที่มีขนาด $N = 7$ โดยในที่นี้ พิจารณาว่า v_s มีความถี่ความสำคัญเพียง 4 อันดับสำหรับ θ ซึ่งเดิมมีขนาด $[M \times 7]$ เมื่อเลือกเฉพาะคอลัมน์ที่สัมพันธ์กับขนาดสูงสุด 4 อันดับแรกแล้ว θ จะถูกลดขนาดในแนวคอลัมน์จนเหลือขนาด $[M \times K]$ โดยที่ $K = 4$

3.3.2.2 การหาสัญญาณคืนกลับ

จากสมการ $y_t = \theta x_s$ เมื่อปรับให้ θ เป็น $\theta|_K$ ดังรูปที่ 3.7 จะได้ $y_t = \theta|_K x_s$ เมื่อพิจารณา $\theta|_K$ ก่อนการลดขนาดเมทริกซ์หากสมาชิกคอลัมน์ที่ i ของ $\theta|_K$ มีค่าเป็นศูนย์ ส่งผลให้ค่าในแถวที่ i ของ x_s ไม่มีความสำคัญ นั่นคือไม่ว่า x_s ในแถวที่ i จะมีค่าเท่าใดก็ตามก็ต้องคูณกับ $\theta_i = 0$ ทำให้ผลลัพธ์เท่ากับศูนย์เสมอด้วยเหตุนี้จึงสามารถลดขนาดมิติ x_s จาก $[N \times 1]$ โดยตัดแถวที่มีค่าเท่ากับศูนย์ออกจะทำให้ขนาดมิติ x_s เหลือ $[K \times 1]$ และจะถูกเขียนแทนด้วย $x_s|_K$ และสำหรับ เมทริกซ์ $\theta|_K$ ก็จะถูกลดมิติ จาก $[M \times N]$ เหลือ $[M \times K]$ และจะถูกเขียนแทนด้วย $\theta|_K$ เช่นกัน ดังนั้นสมการคอมเพรสชันจึงจะลดขนาดระบบสมการได้เป็น $y_t = \theta|_K x_s|_K$

$$\begin{aligned}
 [M \times 1] &= [M \times N][N \times 1] \equiv [M \times 1] = [M \times K][K \times 1] \\
 \mathbf{y}_t &= \boldsymbol{\theta}|_K \mathbf{x}_s|_K \equiv \mathbf{y}_t = \boldsymbol{\theta}|_K \mathbf{x}_s|_K \\
 \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix} &= \begin{bmatrix} \theta_{11} & \cdots & \theta_{1N} \\ \vdots & \ddots & \vdots \\ \theta_{M1} & \cdots & \theta_{MN} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix} \equiv \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} \theta_{11} & \cdots & \theta_{1K} \\ \vdots & \ddots & \vdots \\ \theta_{M1} & \cdots & \theta_{MK} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_K \end{bmatrix}
 \end{aligned}$$

จะเห็นว่าสมการ $\mathbf{y}_t = \boldsymbol{\theta}|_K \mathbf{x}_s|_K$ เป็นระบบสมการที่คำนวณสัญญาณที่ความถี่เฉพาะเจาะจง K ความถี่ เท่านั้น และเป็นสมการแบบชัดเจนเกินไป เนื่องจาก $M > K$ สมการนี้แก้ได้โดยใช้ มัวร์เพนโรสอินเวอร์ส ย้ายข้าง $\boldsymbol{\theta}|_K$ ทำให้ได้ $\mathbf{x}_s|_K$ ซึ่งจะเป็นสัญญาณที่มีเฉพาะความถี่ที่ถูกเลือกไป เรียบร้อยแล้ว เท่านั้นโดย $\boldsymbol{\theta}|_T^\dagger \mathbf{y}_t = \mathbf{x}_s|_K$

การสร้างสัญญาณตัวแทน และ สมการหาสัญญาณคืนกลับเป็นเพียง ขั้นตอนหนึ่งของอัลกอริทึมการ จับคู่สัญญาณบีบอัดเท่านั้นในหัวข้อถัดไปจะอธิบายถึงขั้นตอนทั้งหมดของอัลกอริทึมอย่างละเอียด

3.3.2.3. ขั้นตอนการทำงานของอัลกอริทึม

วิธีการทำงานของอัลกอริทึมนี้โดยรวมแล้วคือการพยายามแทนสัญญาณ \mathbf{x}_s ด้วยสัญญาณ $\mathbf{x}_s|_K$ ซึ่ง ประกอบด้วยความถี่ โดยที่เมื่อแปลงสัญญาณ $\mathbf{x}_s|_K$ กลับไปอยู่ในโดเมนเวลาแล้วจะมีลักษณะใกล้เคียงกับ สัญญาณ \mathbf{y}_t ที่ได้จากการวัดจริงมากที่สุด

วิธีการจับคู่สัญญาณบีบอัดมีลำดับขั้นตอนดังนี้

1. วัดสัญญาณ \mathbf{y}_t และ t_y
2. ทำการจับคู่เวลาเพื่อสร้าง แมทริกซ์การวัด C
3. สร้างแมทริกซ์ฐานหลักเริ่มต้น $\boldsymbol{\theta} = C\boldsymbol{\psi}$

ในขั้นตอนถัดไปนี้จะเป็นการเริ่มกระบวนการของอัลกอริทึมในการวนลูปครั้งแรก

4. สร้างสัญญาณตัวแทน $\mathbf{v}_s = \boldsymbol{\theta}^\dagger \mathbf{y}_t$

5. เลือกความถี่ที่มีค่าแอมพลิจูดสูงสุด $2K$ อันดับแรกบนสัญญาณ \mathbf{v}_s เก็บไว้ในตัวแปร

$$\boldsymbol{\Omega} = \text{supp}(\mathbf{v}_s)_{2K}$$

7. นำตำแหน่งของความถี่ใน $\boldsymbol{\Omega}$ ไปสร้างสัญญาณคืนกลับ $\mathbf{u}_s|_{\boldsymbol{\Omega}} = \boldsymbol{\theta}|_{\boldsymbol{\Omega}}^\dagger \mathbf{y}_t$

8. เลือกตำแหน่งของความถี่ที่มีค่าแอมพลิจูดสูงสุด K อันดับแรกบนสัญญาณ $\mathbf{u}_s|_{\boldsymbol{\Omega}}$ คือ $R = \text{supp}(\mathbf{u}_s|_T)_K$ เก็บชุดของตำแหน่งของความถี่ R เอาไว้ใช้ในขั้นตอนที่ 12

9. เมื่อถึงขั้นตอนนี้สัญญาณ \mathbf{u}_s ก็จะเหลือสมาชิกที่ไม่เท่ากับศูนย์อยู่ K ตัวที่ถูกเลือกออกมา เรียบร้อยแล้วในขั้นตอนก่อนหน้าจนได้เป็น $\mathbf{u}_s = \mathbf{u}_s|_R$ หลักจากนี้จะทำการตรวจสอบว่า สัญญาณที่ได้ผิดไปจากสัญญาณที่วัดมาอย่างน้อยขนาดไหนด้วย $\mathbf{y}_t - \boldsymbol{\theta} \mathbf{u}_s = \boldsymbol{\mu}_t$ โดยที่ค่า $\boldsymbol{\mu}_t$

คือค่าสัญญาณตกค้างที่เป็นผลต่างระหว่างสัญญาณที่วัดได้ y_t และสัญญาณคืนกลับ θu_s ซึ่ง μ_t สามารถใช้เป็นเงื่อนไขในการหยุดวนลูปของโปรแกรมได้ในกรณีเมื่อค่าลูเข้าสู่ค่าคงที่ (สำหรับงานวิจัยนี้จะได้ไม่ได้ใช้ค่าสัญญาณตกค้างเป็นเงื่อนไขการหยุดวนลูป)

ในขั้นตอนถัดไปนี้จะเป็นการวนการของอัลกอริทึมในการเริ่มต้นการวนลูปตั้งแต่ครั้งที่สองเป็นต้นไป

10. นำสัญญาณ u_s จากขั้นตอนที่ 9 หรือ 15 (สำหรับในกรณีที่วนลูปมากกว่า 1 ครั้ง) ไปใช้แทนสัญญาณตัวแทน $v_s = u_s$

11. เลือกความถี่ที่มีค่าแอมพลิจูดสูงสุด $2K$ อันดับแรกบนสัญญาณ v_s เก็บไว้ในตัวแปร

$\Omega = \text{supp}(v_s)_{2K}$ เช่นเดียวกับขั้นตอนที่ 5

12. นำสมาชิกของ Ω ขั้นตอนที่ 11 มารวมกับ R ในขั้นตอนที่ 8 หรือ 14 ของลูปก่อนหน้าซึ่งจะได้ T ซึ่งเป็นอันดับของความถี่ที่มีแอมพลิจูดสูงสุด $3K$ อันดับแรก $T = \Omega \cup R$

13. สร้างสัญญาณคืนกลับ $\theta|_T^\dagger y_t = u_s|_T$

14. เลือกตำแหน่งของความถี่ที่มีค่าแอมพลิจูดสูงสุด K อันดับแรกบนสัญญาณ $u_s|_T$ คือ $R = \text{supp}(u_s|_T)_K$ เก็บชุดของตำแหน่งของความถี่ R เอาไว้ใช้ในการวนลูปครั้งถัดไปเช่นเดิมในขั้นตอนที่ 12.

15. เมื่อถึงขั้นตอนนี้สัญญาณ u_s ก็จะเหลือสมาชิกที่ไม่เท่ากับศูนย์อยู่ K ตัวที่ถูกเลือกออกมาเรียบร้อยแล้วในขั้นตอนก่อนหน้าจนได้เป็น $u_s = u_s|_R$ แล้วในขั้นตอนก่อนหน้าซึ่งในขั้นตอนนี้ ถ้าหากตรวจสอบเงื่อนไขการหยุดลูป (เช่น ครบจำนวนครั้งการวนลูป หรือสัญญาณตกค้างน้อยกว่าค่าที่กำหนด) แล้วเป็นจริง สามารถนำค่าสัญญาณ u_s ไปตอบได้เป็น $z_s = u_s$ ถ้าหากยังไม่เข้าเงื่อนไขการหยุดก็จะกลับไปเริ่มที่ขั้นตอนที่ 10 ต่อไปจนกว่าเงื่อนไขการวนลูปจะเป็นจริง

จุฬาลงกรณ์มหาวิทยาลัย

สำหรับกระบวนการทำงานของอัลกอริทึมจับคู่สัญญาณบีบอัดจะถูกแสดงแผนผังในรูปแบบที่ 3.8

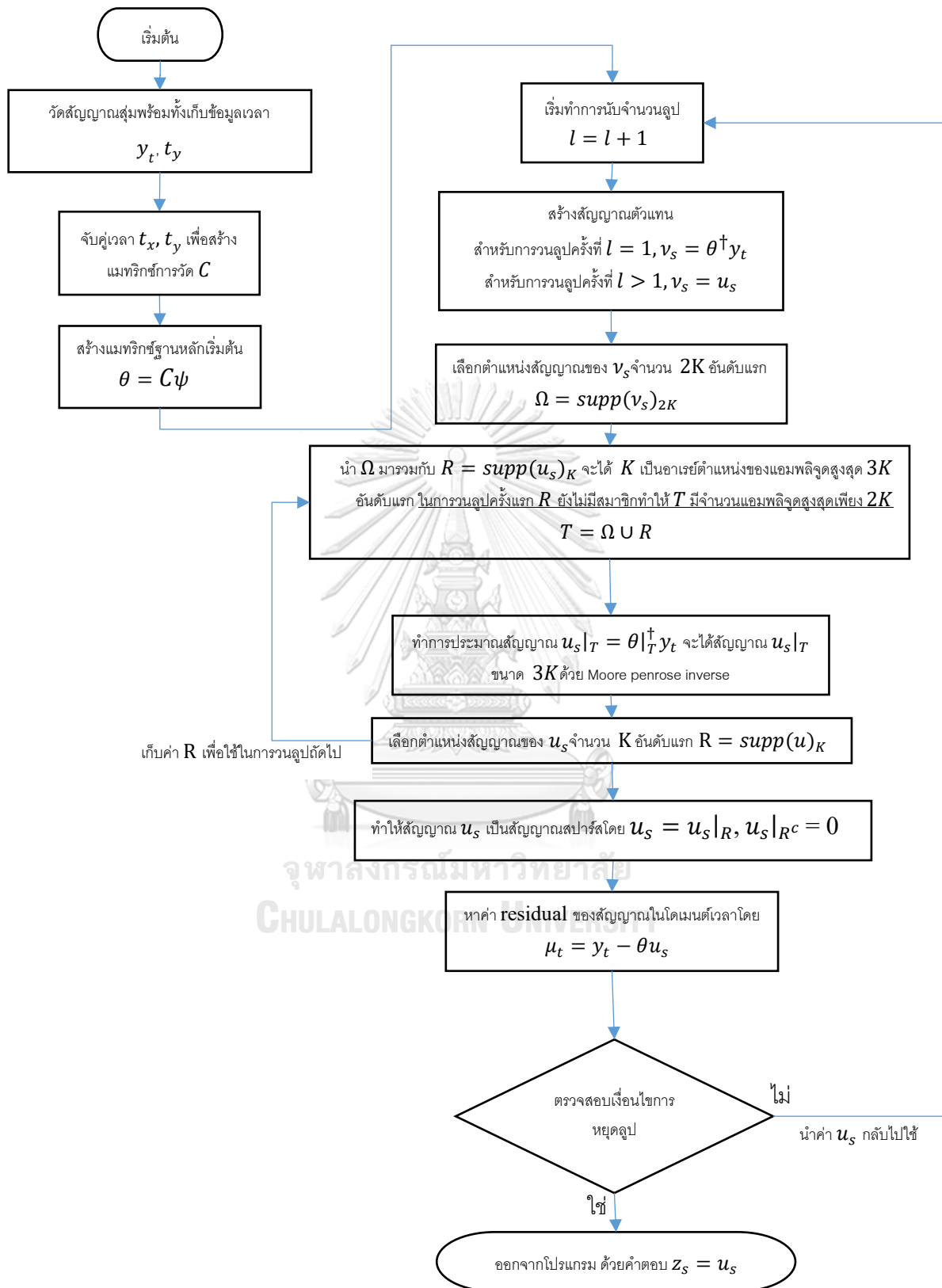
จากที่กล่าวมาข้างต้นจะเห็นว่าอัลกอริทึมประเภทนี้จะตัดสินใจเลือกเอาตำแหน่งแอมพลิจูดที่ดีที่สุดตามสัญญาณตัวแทนที่มี ณ ขณะนั้นเท่านั้น อัลกอริทึมนี้จึงถูกจัดอยู่ในอัลกอริทึมละโมบ และด้วยวิธีคำนวณเช่นนี้ จึงทำให้การคำนวณสามารถทำได้อย่างรวดเร็วกว่าการใช้ค่าเหมาะสมที่สุดแบบคอนเวกซ์ แต่ก็จะมีข้อจำกัดเรื่องการกำหนดจำนวนอันดับของความถี่ที่มีความสำคัญเข้าไปในการคำนวณ ซึ่ง ณ ตอนนี้อย่างนี้ต้องให้ผู้ใช้งานกำหนดค่าเข้าไปเอง

ตัวดำเนินการพิเศษในแผนภาพรูปที่ 3.8

1. $\text{Supp}(v_S)_{2K}$ หมายถึง ตำแหน่งของสมาชิกใน v_S ที่มีค่ามากที่สุด $2K$ อันดับแรก
2. $v_S|_K$ หมายถึง ค่าของสมาชิกในสัญญาณ v_S ที่มีแอมพลิจูดสูงสุด K อันดับแรก
3. $v_S|_{K^c}$ หมายถึง ค่าของสมาชิกในสัญญาณ v_S ในตำแหน่งอื่นๆ ที่เหลือภายในเวกเตอร์ y_S นอกเหนือไปจากสมาชิกที่มีแอมพลิจูดสูงสุด K อันดับแรก
4. $\theta|_K^\dagger$ หมายถึง มัวร์ เพนโรส อินเวอร์ส ของแมทริกซ์ $\theta|_K$

สัญกรณ์ในแผนภาพรูปที่ 3.6

1. l หมายถึง จำนวนครั้งของการวนลูปโปรแกรม ณ ปัจจุบัน
2. v_S หมายถึง สัญญาณตัวแทนในโดเมนความถี่
3. y_t หมายถึง สัญญาณสุ่มในโดเมนเวลา ได้มาจากการวัดโดยตรง
4. Ω, T, R หมายถึง เวกเตอร์ของตำแหน่งสัญญาณในโดเมนความถี่ โดยที่ $R = \emptyset$ เมื่อ $l = 1$
5. u_S หมายถึง สัญญาณสปาร์สในโดเมนความถี่
6. θ หมายถึง แมทริกซ์ฐานหลักเริ่มต้น
7. μ_t หมายถึง สัญญาณตกค้างในโดเมนเวลา เป็นสัญญาณที่เกิดการเปรียบเทียบระหว่างสัญญาณสุ่ม และสัญญาณสปาร์ส ในโดเมนเวลา
8. K หมายถึง จำนวนของสมาชิกที่ไม่เท่ากับศูนย์ในโดเมน ความถี่



รูปที่ 3.8 แสดงวิธีการทำงานของ อัลกอริทึม จับคู่สัญญาณบีบอัด

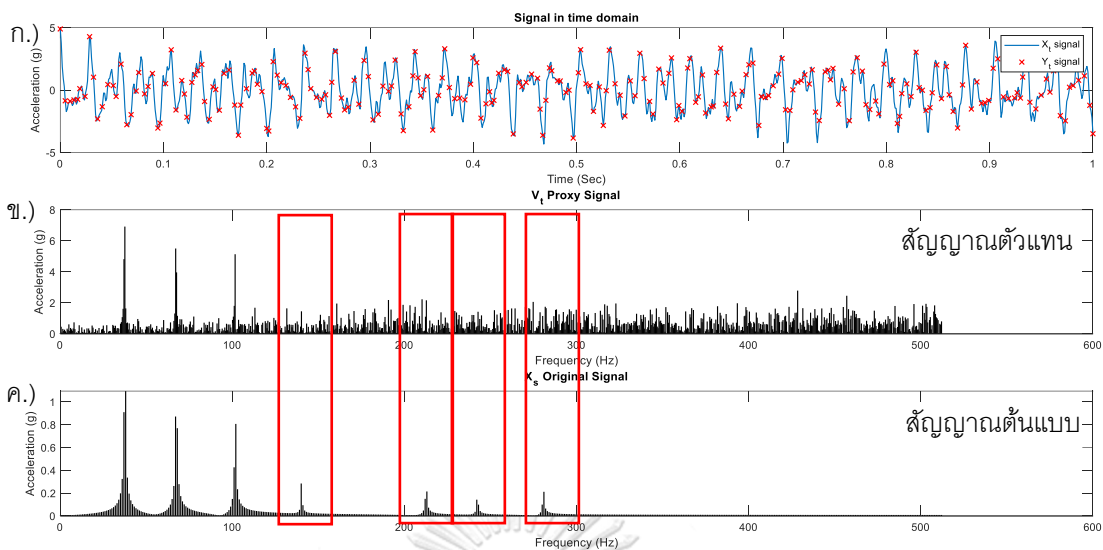
รูปที่ 3.8 ในขั้นตอนที่นำ Ω มารวมกับ $R = \text{supp}(u_s)_K$ เพื่อให้ได้ชุดของลำดับ T นั้นสำหรับการวนลูปรั้งแรกที่ $l = 1$ เนื่องจาก R ยังไม่ถูกสร้างขึ้นทำให้สมาชิกในครั้งแรกของ T มีขนาดเพียง $2K$ แทนที่จะเป็น $3K$ สำหรับเงื่อนไขในการหยุดลูปรั้งขึ้นอยู่กับผู้ใช้ว่าจะทำการหยุดเมื่อไหร่ เช่น ทำการวนลูปรั้งจำนวนครั้ง, ขนาดของสัญญาณตกค้าง $\|\mu_t\|_2$ ต่ำกว่าค่าที่กำหนด แต่ในงานวิจัยนี้จะเลือกใช้เป็นการวนลูปรั้งจำนวนครั้งโดยตั้งไว้ที่ 10 รอบ

โดยภาพรวมของอัลกอริทึมประเภทนี้นั้นมีจุดมุ่งหมายในการแก้ระบบสมการที่ (2.10) $y_t = \theta x_s$ ด้วยการลดจำนวนของตัวแปรที่ต้องการในระบบลงโดยเลือกเฉพาะตัวแปรที่คาดว่าจะมีความสำคัญมากที่สุด K อันดับ ส่งผลให้สมการคอมเพรสชันซึ่งปรับจากระบบสมการไม่ชัดเจนกลายเป็นระบบสมการชัดเจนเกินไป ทำให้ได้คำตอบของระบบสมการเป็นค่าประมาณใกล้เคียง x_s เพียงคำตอบเดียวเท่านั้น

3.3.3 การหาสัญญาณสเปิร์สโดยให้วิธีการเฉลี่ยสัญญาณในโดเมนความถี่

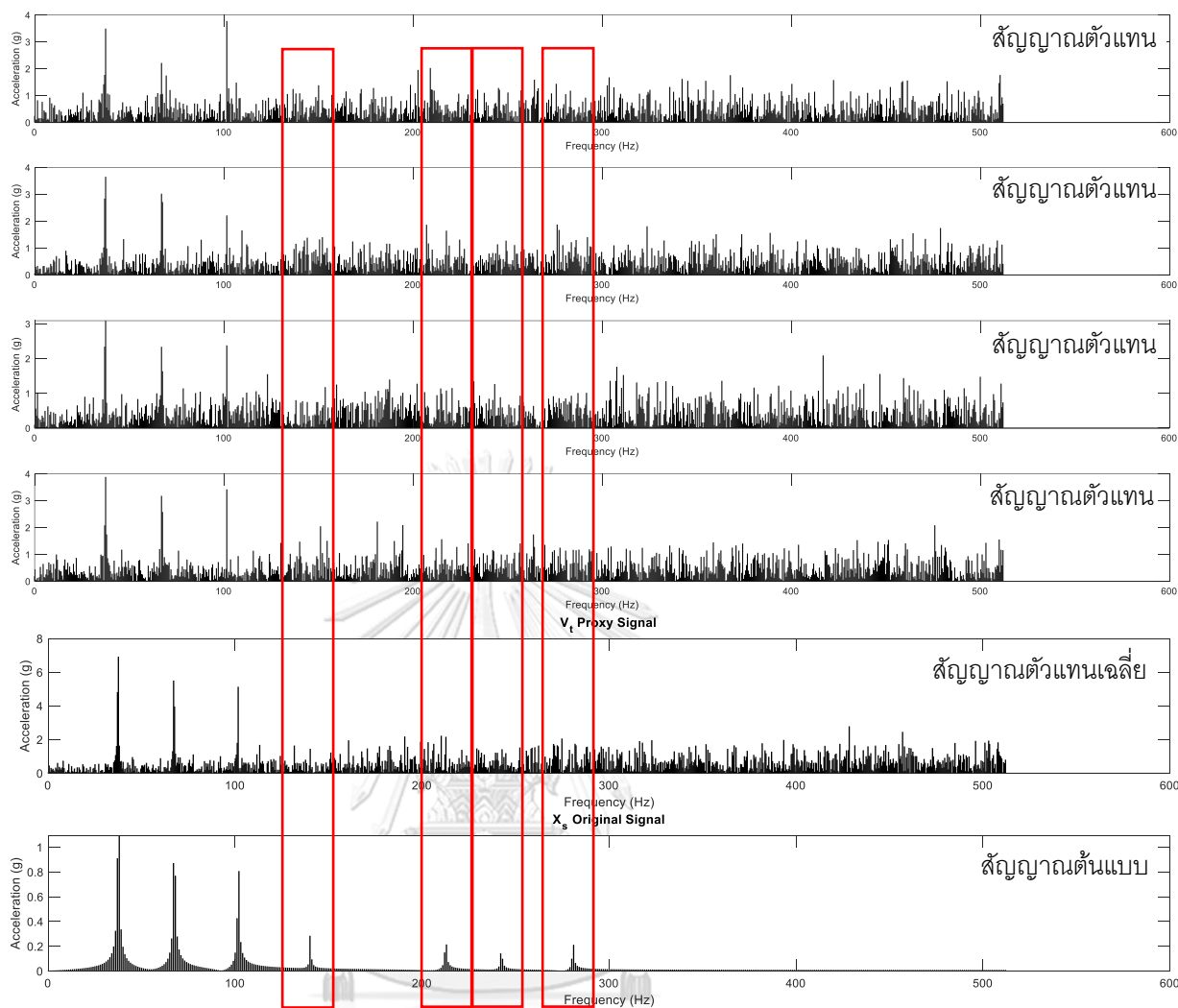
ในหัวข้อก่อนหน้า ได้กล่าวถึงคอมเพรสชันซึ่ง ในการสร้างสัญญาณสเปิร์ส (สัญญาณผลลัพท์) 2 วิธีวิธีการหาค่าเหมาะสมที่สุดแบบคอนเวกซ์ ให้ผลลัพธ์ที่ดีแต่ใช้เวลาคำนวณมาก ส่วนการจับคู่สัญญาณบีบอัดถึงแม้จะคำนวณได้รวดเร็วแต่ ความแม่นยำก็ขึ้นกับคุณภาพของสัญญาณตัวแทน

เนื่องจากการสุ่มเก็บข้อมูลเพียงชุดเดียวนำมาสร้างสัญญาณตัวแทนนั้น สัญญาณที่ได้มักมีสัญญาณรบกวนสูงเนื่องจากจำนวนจุดวัดที่ต่ำกว่าทฤษฎี แซนนอนและไนควิสต์ (Shannon-Nyquist sampling theorem) ทำให้ไม่สามารถเลือกลำดับความถี่ที่มีแอมพลิจูดสูงสุด K อันดับแรก ได้อย่างแม่นยำ โดยเฉพาะในตำแหน่งที่แอมพลิจูดของสัญญาณตัวแทน v_s ใกล้เคียงกับแถบสัญญาณรบกวนซึ่งไม่อาจจำแนกได้ว่าความถี่ใดมีความสำคัญมากกว่า ทำให้การเลือกตำแหน่งเกิดความคลาดเคลื่อนจากสัญญาณจริง วิธีการที่จะนำเสนอต่อไปนี้เป็นวิธีการที่เสนอขึ้นมาใหม่เพื่อปรับปรุงคุณภาพของสัญญาณตัวแทน โดยการเพิ่มจำนวนครั้งในการเก็บข้อมูลให้มากขึ้น เนื่องจาก การเก็บข้อมูลแต่ละครั้งมีลักษณะเป็นแบบสุ่มทำให้สัญญาณตัวแทนที่ถูกขึ้นมาแต่ละครั้งจะมีความแตกต่างกัน



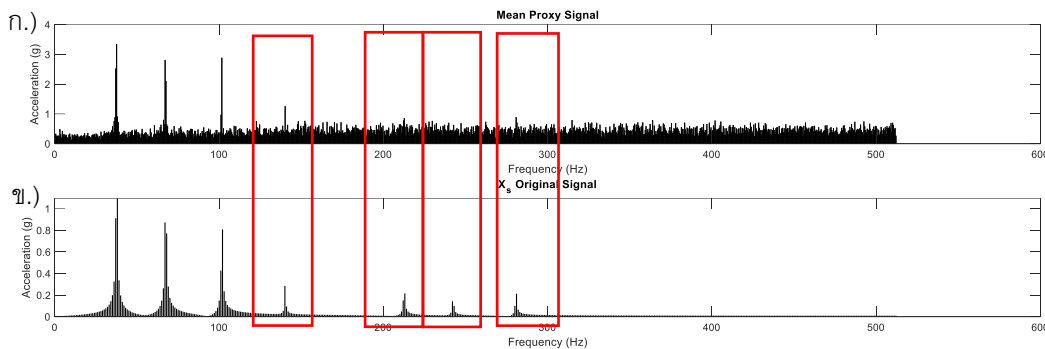
รูปที่ 3.9 ก.) สัญญาณต้นแบบในโดเมนเวลา ข.) สัญญาณตัวแทนในโดเมนความถี่ ค.) สัญญาณต้นแบบในโดเมนความถี่

รูปที่ 3.9 แสดงตำแหน่งของความถี่ในโดเมนความถี่ของสัญญาณตัวแทน เทียบกับตำแหน่งของความถี่ในสัญญาณทดสอบ จากรูป จะเห็นว่าถึงแม้สัญญาณจริงจะมีความถี่เกิดขึ้นที่ 140, 210, 240 และ 280 เฮิรตซ์ (รูปที่ 3.9 ค.) แต่เมื่อพิจารณาสัญญาณตัวแทน รูปที่ 3.9 ข.) แล้ว สัญญาณที่ความถี่เหล่านี้ขนาดสัญญาณใกล้เคียงกับสัญญาณรบกวน จึงไม่สามารถแยกออกจากสัญญาณรบกวนได้ การจะเลือกความถี่ที่มีแอมพลิจูดสูงสุดของสัญญาณเพื่อนำไปใช้ในการลดขนาดแมทริกซ์ θ จึงทำได้ไม่แม่นยำเนื่องจากมีแนวโน้มที่ความถี่ของสัญญาณรบกวนจะถูกเลือกไปแทนที่ความถี่ของสัญญาณจริง แต่หากสุ่มวัดสัญญาณและหาสัญญาณตัวแทนหลายๆ ครั้งถึงแม้ สัญญาณตัวแทนที่ได้มาใหม่จะไม่เหมือนเดิมดังแสดงในรูปที่ 3.10 แต่หากสัญญาณตั้งต้นมีความคงตัวตลอดการวัด ความถี่ที่เป็นสัญญาณจริงจะยังคงเกิดในทุกครั้งของการวัด ส่วนสัญญาณรบกวนจะเกิดแบบสุ่ม



รูปที่ 3.10 เปรียบเทียบสัญญาณตัวแทนในแต่ละรอบการวัดทั้งหมด 4 ครั้งเทียบกับสัญญาณทดสอบ

เมื่อนำสัญญาณตัวแทน ที่วัดแต่ละครั้งมาเฉลี่ยกันในโดเมนความถี่จะส่งผลให้สัญญาณที่เป็นแถบสัญญาณรบกวนถูกลดทอนลงด้วยการเฉลี่ย แต่สัญญาณจริงที่เกิดขึ้นนั้นจะยังคงอยู่ทำให้การเลือกแอมพลิจูดสูงสุด K อันดับแรก เพื่อใช้ในการลดขนาดแมทริกซ์ θ มีความแม่นยำสูงขึ้น



รูปที่ 3.11 ก.) สัญญาณตัวแทนเฉลี่ยในโดเมนความถี่ ข.) สัญญาณต้นแบบในโดเมนความถี่

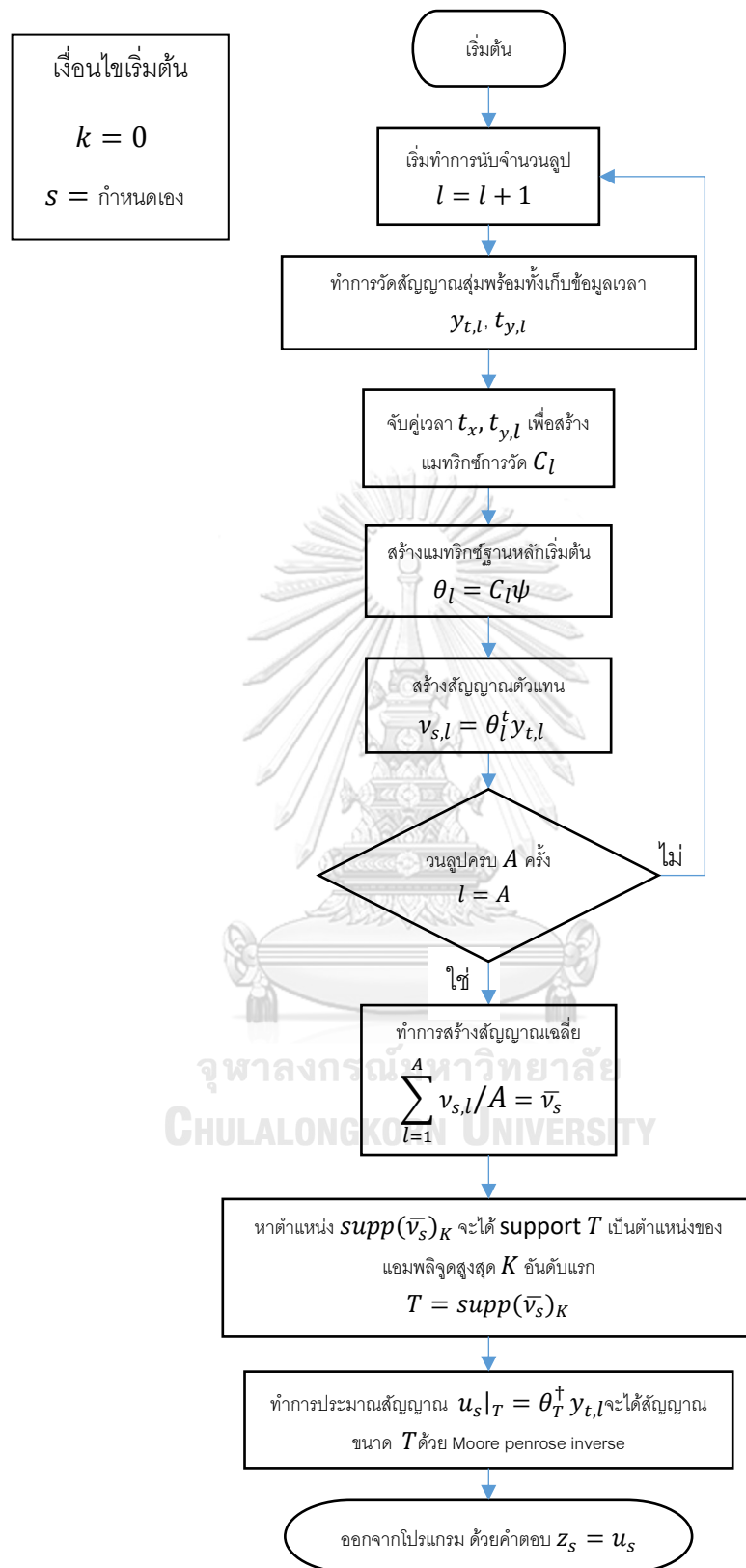
รูปที่ 3.11 แสดงการเปรียบเทียบสัญญาณตัวแทนเฉลี่ยโดยใช้การวัดทั้งหมด 10 ครั้ง เทียบกับสัญญาณต้นแบบ จากรูปสัญญาณเฉลี่ยแสดง ความถี่ที่เกิดขึ้นจริงแยกออกมาจากสัญญาณรบกวนได้ชัดเจนขึ้น ทำให้การเลือกแอมพลิจูดสูงสุด K อันดับแรก มีความแม่นยำสูงขึ้น สมการ (3.4) แสดงสัญญาณที่ได้จากการหาค่าเฉลี่ยโดยกำหนดให้ $v_{s,l}$ เป็นสัญญาณตัวแทน ของการเก็บข้อมูลครั้งที่ l A คือจำนวนครั้งของเก็บข้อมูลทั้งหมด

$$\sum_{l=1}^A \frac{v_{s,l}}{A} = \bar{v}_s \tag{3.4}$$

หลังจากที่ได้ \bar{v}_s ก็จะสามารถนำไปสร้างสัญญาณในโดเมนความถี่ โดยเลือกตำแหน่งของแอมพลิจูดสูงสุด K อันดับแรกในโดเมนความถี่ของสัญญาณ \bar{v}_s เพื่อนำไปลดขนาดเมทริกซ์ θ จาก $[M \times N]$ ให้เหลือ $[M \times K]$ ในที่สุดแล้วก็จะได้สมการ $y_t = \theta |_{K} x_s |_{K}$ แล้วทำการย้ายข้างสมการเพื่อหา $\theta |_{K} y_t = u_s |_{K}$ เช่นเดียวกับวิธีการจับคู่สัญญาณบีบอัดในหัวข้อที่ 3.2.2 โดยไม่จำเป็นต้องย้อนกลับมาทำการเปรียบเทียบกับสัญญาณ y_t เพื่อหาค่าสัญญาณตกค้าง $\mu_t = y_t - \theta u_s$ อีกต่อไป เนื่องจากในตำแหน่งความถี่ของ u_s ที่ถูกเลือกตั้งแต่นั้นมาจากการรวบรวมข้อมูลของสัญญาณ y_t หลายๆ ชุดและเป็นตำแหน่งความถี่ที่น่าจะถูกต้องแล้ว

สัญกรณ์ในรูปที่ 3.12

1. l หมายถึง จำนวนครั้งของการวนลูปโปรแกรม ณ ปัจจุบัน
2. $t_{y,l}$ หมายถึง เวลาของข้อมูลแต่ละจุดของสัญญาณสุ่ม $y_{t,l}$ รอบที่ l
3. $y_{t,l}$ หมายถึง สัญญาณสุ่มรอบที่ l
4. t_x หมายถึง เวลาของข้อมูลแต่ละจุดสำหรับสัญญาณเป้าหมาย
4. ψ หมายถึง เมทริกซ์ฐานหลักทั่วไปขนาด $N \times N$
5. C_l หมายถึง เมทริกซ์การวัดที่ได้จากการจับคู่เวลาชุดที่ l
6. θ_l หมายถึง เมทริกซ์ฐานหลักเริ่มต้นของสัญญาณบีบอัดชุดที่ ชุดที่ l
7. $v_{s,l}$ หมายถึง สัญญาณตัวแทนในโดเมนความถี่ ชุดที่ l
8. \bar{v}_s หมายถึง สัญญาณตัวแทนเฉลี่ย
9. $\text{supp}(\bar{v}_s)_K$ หมายถึง ชุดลำดับของเวกเตอร์ \bar{v}_s ที่ถูกนำมาเรียงลำดับข้อมูลจากมากไปน้อย K ลำดับ
10. θ_K หมายถึง เมทริกซ์ฐานหลักที่ถูกเลือกเฉพาะแถวตามลำดับของ K แล้วนำไปผ่านกระบวนการ มัวร์ เพนโรส อินเวอร์ส
11. K หมายถึง ค่าความสปีร์สหรือจำนวนจุดข้อมูลที่ไม่เท่ากับศูนย์ในสัญญาณต่างๆ
12. u_s หมายถึง สัญญาณสปีร์สในโดเมนความถี่



รูปที่ 3.12 แสดงวิธีการทำการของ อัลกอริทึม Mean Stack

จาก รูปที่ 3.12 เมื่อเทียบกับการจับคู่สัญญาณบีบอัด จะเห็นว่าการวัดสัญญาณ $y_{t,k}$, $t_{y,k}$ เป็นจำนวนหลายครั้งเพื่อให้ได้ \bar{V}_S จากนั้นก็จะนำไปเลือกเฉพาะความถี่ของแอมพลิจูดสูงสุดทั้งหมด K อันดับโดยไม่ต้องวนกลับมาหาค่าสัญญาณตกค้าง



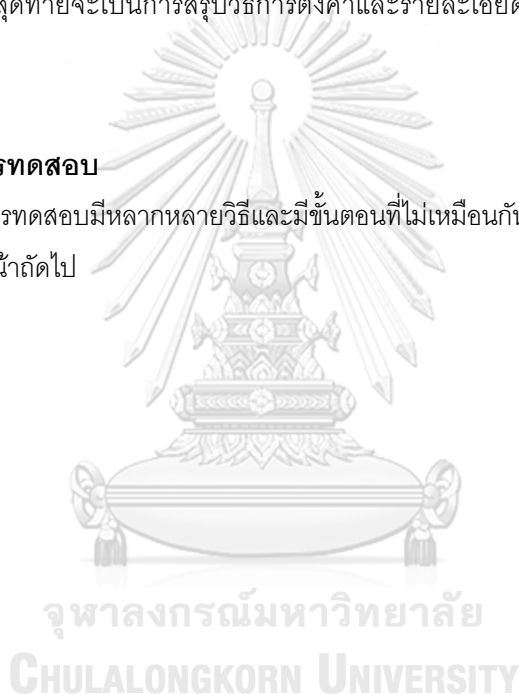
บทที่ 4

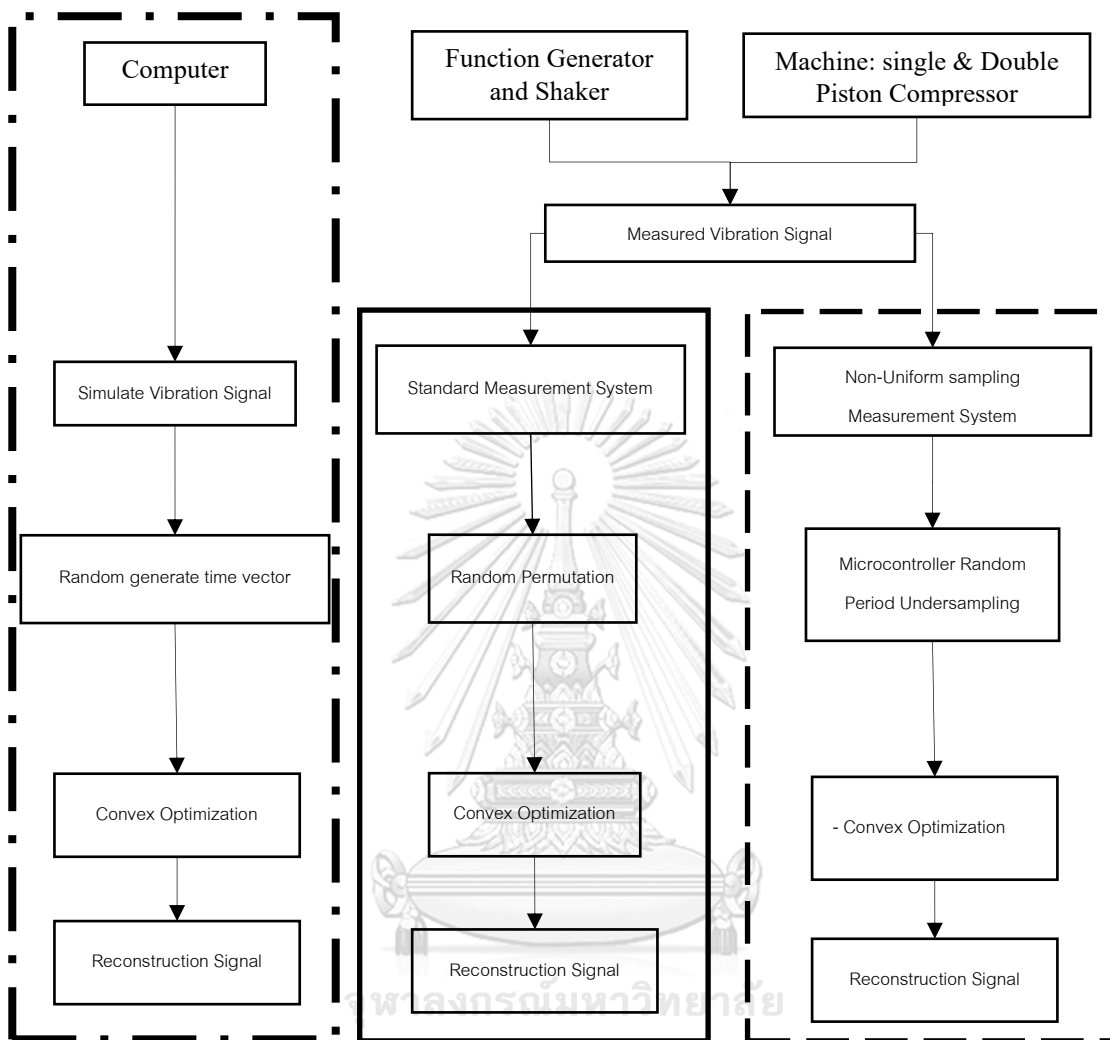
การทดสอบการใช้คอมพิวเตอร์เชิงกับการวัดการสันสะเทือน

ในบทนี้จะถูกแบ่งเป็น 5 หัวข้อหลัก คือ 1.ภาพรวมของการทดสอบ 2.การทดสอบหาความเป็นไปได้ในเชิงหลักการ 3.การทดสอบหาความเป็นไปได้ในทางปฏิบัติจริง 4.การประเมินสัญญาณ 5.สรุปวิธีการทดสอบและจุดมุ่งหมายในการทดสอบ ซึ่งใน 3 หัวข้อแรกจะอธิบายรูปแบบการทดสอบแบบต่างๆ เกี่ยวกับจุดมุ่งหมายในแต่ละการทดสอบ เครื่องมือที่ใช้ และลักษณะของสัญญาณที่ใช้ทดสอบ การตั้งค่าพารามิเตอร์และขั้นตอนการทดสอบ สำหรับหัวข้อที่ 4 จะเป็นวิธีการประเมินสัญญาณของการทดสอบแบบต่างๆที่กล่าวถึงในหัวข้อที่ 2 และ 3 เนื่องจากวิธีการในแต่ละการทดสอบแต่ละแบบมีความแตกต่างกันจึงจำเป็นต้องใช้วิธีการประเมินแตกต่างกันด้วย หัวข้อสุดท้ายจะเป็นการสรุปวิธีการตั้งค่าและรายละเอียดการทดสอบต่างๆให้อยู่ในรูปของตาราง

4.1. ภาพรวมของการทดสอบ

เนื่องจากวิธีการทดสอบมีหลากหลายวิธีและมีขั้นตอนที่ไม่เหมือนกัน จึงขอแนะนำเสนอภาพรวมของการทดสอบเป็นแผนผังในหน้าถัดไป





- Experiment One : Proof of Concept #1 Simulation
- Experiment Two : Proof of Concept #2 Standard measurement
- Experiment Three : Proof of Work Non-uniform measurement

รูปที่ 4.1 แผนผังแสดงภาพรวมของการทดสอบทั้งหมดในงานวิจัยนี้

จากรูปที่ 4.1 จะเห็นว่า เป้าหมายของการทดสอบการทดสอบจะถูกแบ่งออกเป็น 2 หัวข้อหลักก็คือการทดสอบหลักการ (Proof of Concept) และการทดสอบการนำไปใช้จริง (Proof of work) ซึ่งผลลัพธ์จะถูกแบ่งเป็น 3 แบบคือ การทดสอบในเชิงของหลักการครั้งที่ 1 (Proof of Concept) กรอบเส้นลูกโซ่ เป็นผลการทดสอบจากการจำลองสัญญาณบนคอมพิวเตอร์ การทดสอบในเชิงของหลักการครั้งที่ 2 (Proof of Concept) (กรอบเส้นเต็ม) เป็นผลการทดสอบจากการวัดสัญญาณจากเครื่องจักรจริงด้วยระบบมาตรฐานตามด้วยการสุ่มหยิบสัญญาณเพื่อให้สัญญาณที่ได้มีลักษณะเป็นสัญญาณแบบสุ่มคาบ และการทดสอบในเชิงปฏิบัติจริง (Proof of Work) (กรอบเส้นประ) การวัดจริงด้วยระบบการวัดแบบสุ่มคาบจริงๆ

สำหรับการทดสอบในเชิงหลักการและการทดสอบในทางปฏิบัติจริง ขั้นตอนอย่างละเอียดจะถูกอธิบายหัวข้อถัดไป

4.2. การทดสอบหาความเป็นไปได้ในเชิงหลักการ (Proof of concept)

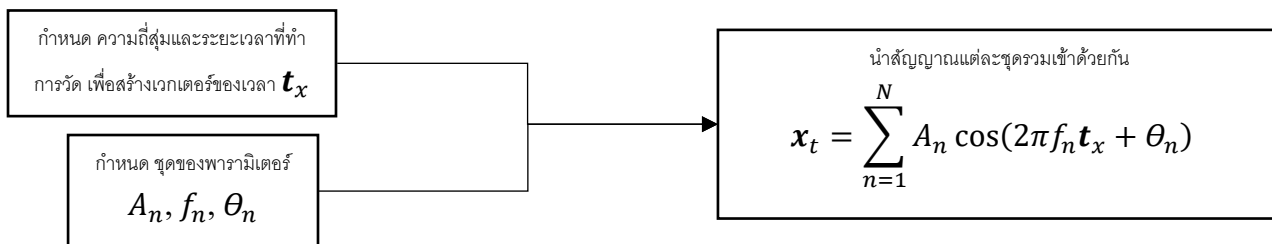
การทดสอบในเชิงหลักการจะเป็นการทดสอบว่า ในกรณีที่ทราบเวลาและขนาดของสัญญาณการสั่นสะเทือนที่แม่นยำ ไม่มีปัจจัยอื่นๆที่เข้ามารบกวนสัญญาณให้ความถูกต้องลดลงเหมือนกับการวัดสัญญาณจริง เทคนิคคอมพิวเตอร์เชิงสามารถจะนำไปใช้ได้หรือไม่ การทดสอบต้องทำโดยการจำลองบนคอมพิวเตอร์ และการใช้เซ็นเซอร์คุณภาพสูง ซึ่งจะถูกแบ่งออกเป็น 2 ครั้งคือ

1. จำลองการวัดสัญญาณสั่นสะเทือนบนคอมพิวเตอร์
 2. การทดสอบโดยการวัดการสั่นสะเทือนจากการวัดเครื่องจักรจริง
- โดยแต่ละการทดสอบจะมีรายละเอียดดังนี้

4.2.1. จำลองการวัดสัญญาณสั่นสะเทือนบนคอมพิวเตอร์

การจำลองสัญญาณทดสอบทำในโปรแกรม MATLAB ด้วยฟังก์ชัน $x_t = F(t_x)$ โดยที่ x_t คือเวกเตอร์ของแอมพลิจูดของสัญญาณในโดเมนเวลาที่มีความยาวเท่ากับ $[N \times 1]$ มีหน่วยเป็น g หรือ 1 เท่าของความเร่งเนื่องจากแรงโน้มถ่วงโลก t_x คือเวกเตอร์เวลาของสัญญาณมีความยาวเท่ากับ $[N \times 1]$ มีหน่วยเป็นวินาที (Sec) ซึ่ง t_x จะถูกสร้างขึ้นโดยกำหนดความถี่ในการสุ่มสัญญาณ (Sampling Frequency) และระยะเวลาในการวัด ส่งผลให้จำนวนจุด N มีจำนวนเท่ากับความถี่ในการสุ่มสัญญาณ คูณด้วยระยะเวลาในการทำการวัด

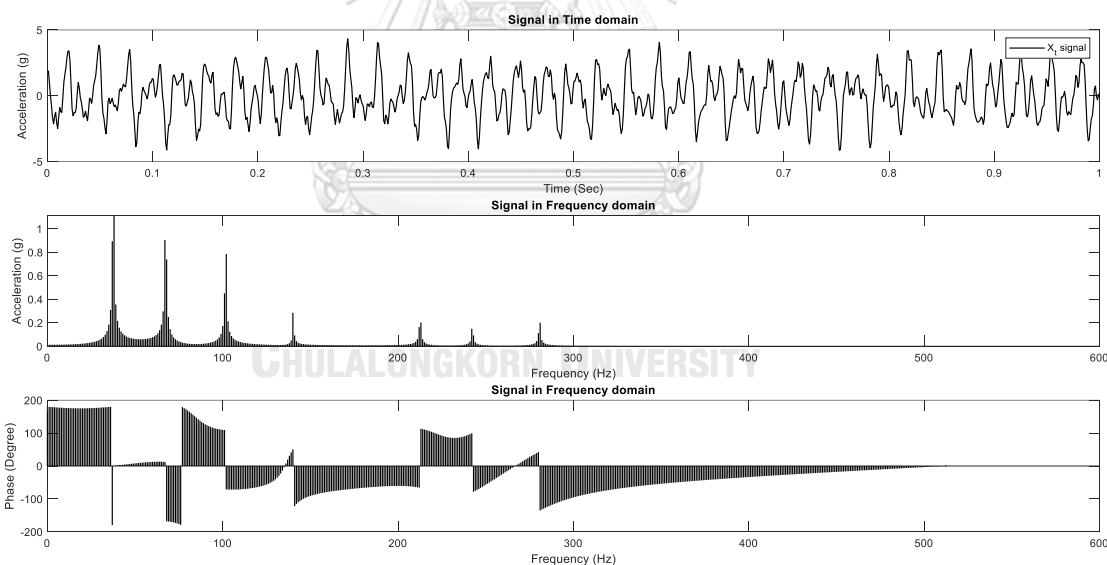
สำหรับฟังก์ชัน $x_t = F(t)$ จะกำหนดให้ x_t ซึ่งเกิดจากการประกอบกันของสัญญาณโคไซน์ $x_t = \sum_{n=1}^N A_n \cos(2\pi f_n t + \theta_n)$ โดยที่ A_n คือแอมพลิจูดของสัญญาณชุดที่ n , f_n คือความถี่ของสัญญาณชุดที่ n มีหน่วยเป็นเฮิรตซ์ (Hertz) และ θ_n คือเฟสของสัญญาณชุดที่ n มีหน่วยเป็นเรเดียน (Radian) ซึ่งทั้งหมดจะสามารถอธิบายเป็นแผนผังได้ตามรูปที่ 2 ด้านล่าง



รูปที่ 4.2 แผนผังแสดงวิธีการจำลองสัญญาณทดสอบที่ใช้ในโปรแกรม Matlab

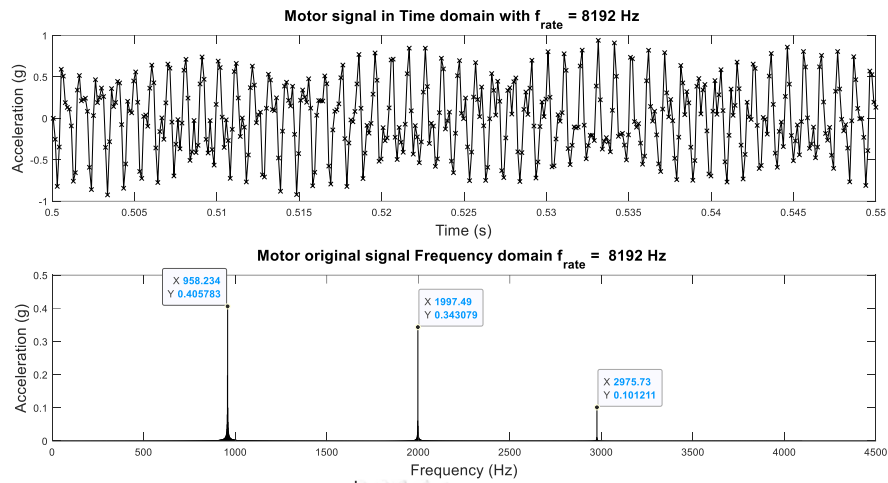
ตารางที่ 4.1 ตัวอย่างการกำหนดค่าของสัญญาณทดสอบ

Amplitude (g)	Frequency (Hertz)	Phase (Radian)
1.579	37.512	1.237
1.292	67.394	0.222
0.986	101.534	1.117
0.316	140.094	0.045
0.288	212.353	1.392
0.192	242.146	2.834
0.249	280.366	2.922

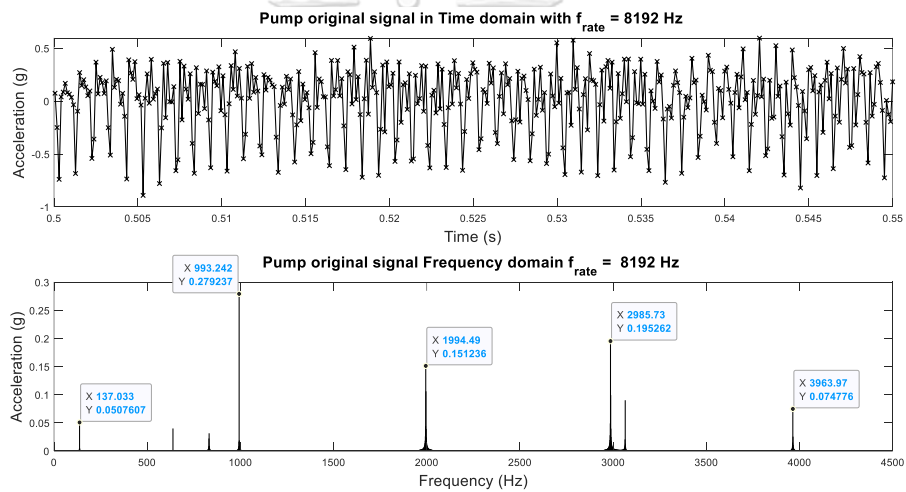


รูปที่ 4.3 ตัวอย่างผลการจำลองสัญญาณทดสอบโดยใช้ค่าจากตารางที่ 4.1

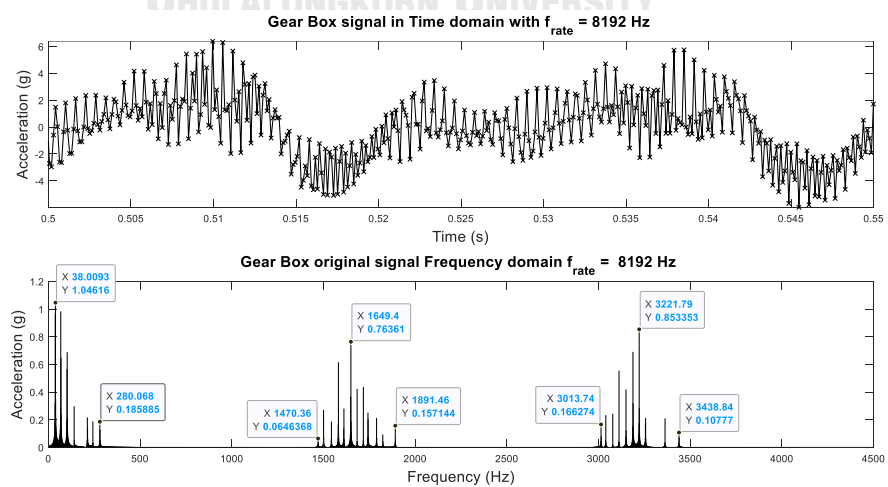
จากรูปที่ 4.3 เป็นตัวอย่างสัญญาณที่ได้จากระบบสมการ $x_t = F(t)$ ที่มีค่าสัมประสิทธิ์ A_n, f_n, θ_n ของสมการเท่ากับค่าในตารางที่ 4.1



รูปที่ 4.4 สัญญาณมอเตอร์



รูปที่ 4.5 สัญญาณปั๊ม



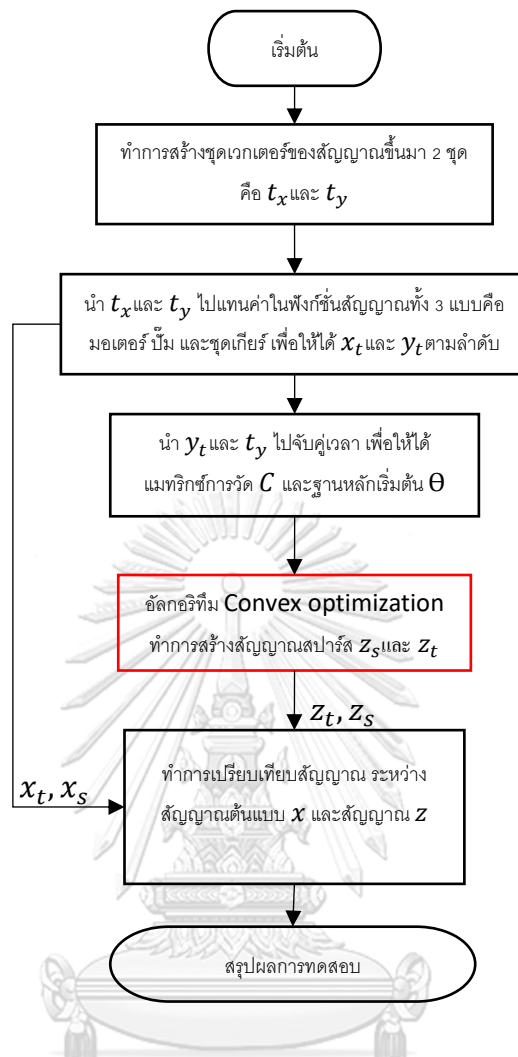
รูปที่ 4.6 สัญญาณชุดเกียร์

สำหรับการทดสอบนี้จะจำลองสัญญาณการสั่นสะเทือน 3 รูปแบบ จากรูปที่ 4.4, 4.5 และ 4.6 ซึ่งมีต้นแบบมาจากสัญญาณสเปกตรัมที่วัดจากเครื่องจักร 3 ชนิดได้แก่ มอเตอร์ บี้ม และชุดเฟืองทด โดยสัญญาณจำลองถูกสร้างจากสมการ $x_t = \sum_{n=1}^N A_n \cos(2\pi f_n t + \theta_n)$ โดยกำหนดให้ลักษณะของ A_n , f_n และ θ_n ของ เครื่องจักรทั้ง 3 ชนิดนั้นมีลักษณะในโดเมนความถี่ที่แตกต่างกันไปดังแสดงในรูปที่ 4.4, 4.5 และ 4.6 โดยในรูปที่ 4.4 และ 4.5 เป็นสัญญาณของมอเตอร์และบี้ม ซึ่งจะมีความถี่ที่เกิดขึ้นค่อนข้างจะกระจายตัว แต่ความถี่ที่เกิดขึ้นก็จะสัมพันธ์กับจำนวนเท่าของความถี่รอบการหมุน ตลอดพิสัยของสัญญาณจะมีจำนวนความถี่หลักน้อย โดยมอเตอร์จะมีเพียงความถี่หลักเท่านั้นที่มีค่าแอมพลิจูดสูง แต่บี้มจะมีความถี่บางตำแหน่งที่มีแอมพลิจูดต่ำอยู่ในบริเวณใกล้เคียงด้วย สำหรับในรูปที่ 4.6 เป็นสัญญาณของชุดเฟืองทด มีความถี่หลักจะเป็นจำนวนเท่าของความถี่รอบการหมุนเพลลา และความถี่ 1 เท่าและ 2 เท่าของความถี่การขบ (Gear Mesh Frequency, GMF) นอกจากนี้ยังพบความถี่อื่นๆ และความถี่แถบข้างรอบๆ ความถี่หลักด้วย

เมื่อพิจารณาความเป็นสเปกตรัมของสัญญาณของเครื่องจักรทั้ง 3 ชนิดจะเห็นว่าสัญญาณของมอเตอร์ มีความเป็นสเปกตรัมมากที่สุด และสัญญาณชุดเฟืองมีความสเปกตรัมน้อยที่สุด การจำลองสัญญาณเช่นนี้ทำให้สามารถทราบผลของการใช้เทคนิคคอมเพรสชันซิง กับสัญญาณต่างๆที่มีความสเปกตรัมต่างกันได้ในแต่ละรูปแบบสัญญาณ

โดยสัญญาณต้นแบบที่ใช้ จะอยู่ในรูปของฟังก์ชัน $x_t = f(t_x)$ ของมอเตอร์ บี้มและชุดเฟืองทด กำหนดให้ x_t มีอัตราสุ่มจุดข้อมูลอยู่ที่ 8192 เฮิร์ตซ์ มีความยาว 1 วินาที แต่สำหรับ $y_t = f(t_y)$ จะกำหนดให้อัตราสุ่มจุดข้อมูลเทียบเท่าอยู่ที่ 4092 เฮิร์ตซ์โดยที่ t_y จะมีระยะเวลาของจุดข้อมูลไม่เท่ากัน ดังที่ได้อธิบายไปในหัวข้อ 3.1.1 ข้อที่ 1 พิสัยของความถี่สูงสุดที่ต้องการวัดเท่ากับ 4092 เฮิร์ตซ์ หลังจากนั้นนำ t_x และ t_y ไปจับคู่เวลาเพื่อหาแมทริกซ์ฐานหลัก ก่อนใช้อัลกอริทึมแบบคอนเวกซ์ทำการแก้สมการหาสัญญาณคืนกลับ แล้วนำกลับไปเปรียบเทียบกับสัญญาณต้นแบบ x_t, x_s ที่ถูกสร้างด้วยฟังก์ชันสัญญาณเดียวกัน โดยทำการทดสอบทั้งหมด 5 ครั้งเพื่อหาค่าเฉลี่ย ขั้นตอนในการทดสอบสามารถถูกอธิบายอยู่ในรูปของแผนผังในรูปที่ 4.7 ด้านล่าง

สำหรับวิธีการเปรียบเทียบจะถูกอธิบายในหัวข้อที่ 4.5 การประเมินผลสัญญาณ



รูปที่ 4.7 แผนผังแสดงวิธีการทำการทดสอบการจำลองสัญญาณในคอมพิวเตอร์

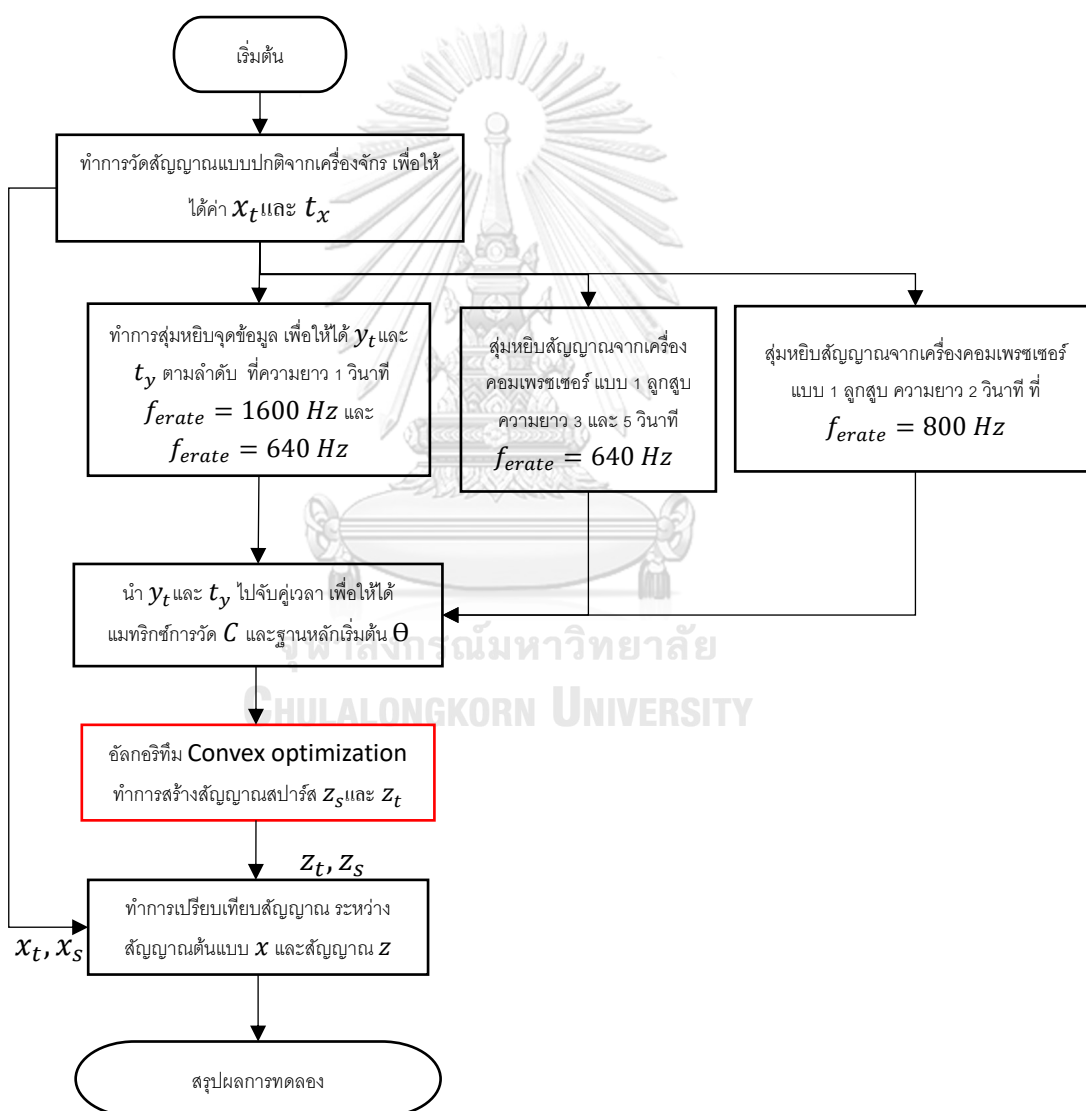
4.2.2 การทดสอบโดยการวัดการสั่นสะเทือนจากการวัดเครื่องจักรจริงด้วยระบบการวัดมาตรฐาน

วิธีการนี้จะทำการวัดสัญญาณเครื่องจักรจริง โดยระบบการวัดสัญญาณจะใช้ระบบการวัดที่มีคุณภาพสูงในการทดสอบ (ใช้เซ็นเซอร์ PCB TLD352A56 และ DAQ dewesoft sirius) ทำการวัดแบบปกติเป็นระยะเวลา 10 วินาที ด้วยความถี่ในการสุ่มจุดคงที่ที่อัตรา 20000 เฮิรตซ์ ความยาว 10 วินาที หลังจากนั้นนำไปลดความถี่สุ่มลงให้เหลือ 3200 เฮิรตซ์ ทำให้ได้ x_t และ t_x ที่พิสัยความถี่จะอยู่ที่ 0-1600 เฮิรตซ์ สำหรับสัญญาณ y_t และ t_y จะถูกสร้างขึ้นโดยใช้วิธีสุ่มหยิบจุดข้อมูลใน 3.1.1 หัวข้อที่ 2 โดยกำหนดให้สัญญาณ y_t มีอัตราการสุ่มจุดเทียบเท่ากับการวัดโดยใช้เซ็นเซอร์ 1600 เฮิรตซ์ และ 640 เฮิรตซ์ เลือกใช้ความยาวเพียง 1 วินาทีเพื่อนำมาทดสอบ วิธีการดังกล่าวจะทำการวัดเครื่องจักรทั้งหมด 2 ชนิดด้วยกัน คือ

1. วัดสัญญาณจากเครื่องสร้างการสั่นสะเทือน
2. วัดสัญญาณจากเครื่องจักรที่ถูกใช้งานจริง คือ คอมเพรสเซอร์ แบบ 1 ลูกสูบ และ แบบ 2 ลูกสูบ

หลังจากที่ทดสอบด้วยสัญญาณที่เกิดจากแหล่งกำเนิดทั้งสองแบบข้างต้นแล้ว จะเลือกเฉพาะสัญญาณคอมเพรสเซอร์แบบ 1 ลูกสูบ ที่ใช้สัญญาณแบบอัตราสุ่มจุดข้อมูลเทียบเท่า 640 เฮิร์ตซ์ในความยาวของสัญญาณที่ยาวขึ้นจากเดิม 1 วินาที เป็น 3 และ 5 วินาที เพื่อดูผลของสัญญาณคืนกลับว่ามีคุณภาพเพิ่มมากขึ้นหรือไม่เมื่อใช้ความยาวของสัญญาณสุ่มมากขึ้น

จากนั้นจะทดสอบว่าการปรับปรุงพารามิเตอร์สามารถเพิ่มคุณภาพของสัญญาณคืนกลับได้ดีกว่ากัน โดยจะกำหนดให้ สัญญาณแรก มีความถี่สุ่มเทียบเท่า 1600 เฮิร์ตซ์ ยาว 1 วินาที และ สัญญาณที่สองมีความถี่สุ่มเทียบเท่า 800 เฮิร์ตซ์ ยาว 2 วินาที ซึ่งทั้งสองสัญญาณนี้มีจำนวนจุดที่ถูกใช้คำนวณเท่ากันคือประมาณ 1300 จุด สำหรับขั้นตอนในการทดสอบสามารถถูกอธิบายอยู่ในรูปของแผนผังในรูปที่ 4.8 ด้านล่าง

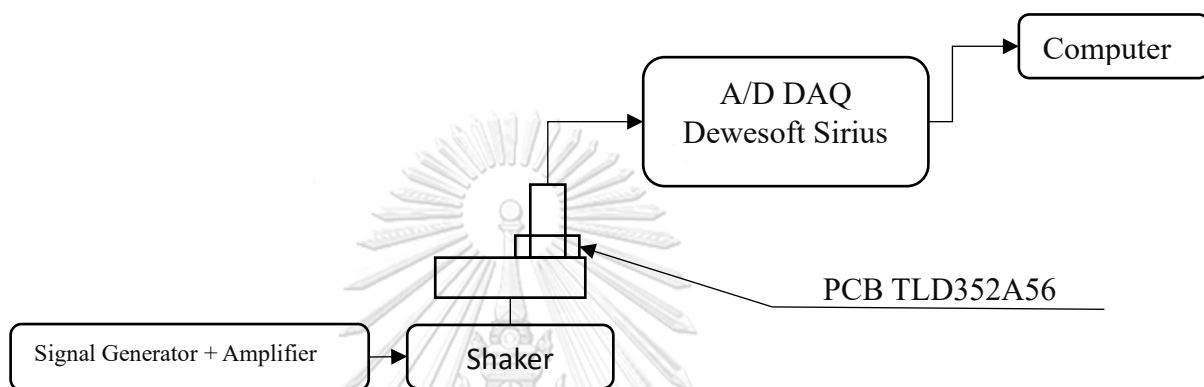


รูปที่ 4.8 แผนผังแสดงวิธีการทำการทดสอบการวัดแบบสุ่มหยิบสัญญาณ

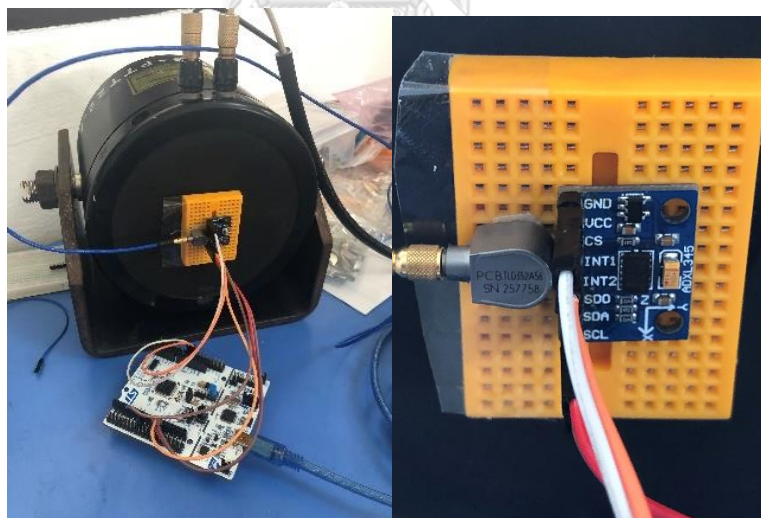
ในหัวข้อถัดๆไป จะอธิบายถึงรายละเอียดของ แหล่งกำเนิดสัญญาณการสั่นสะเทือนที่ใช้ในการทดสอบของหัวข้อ 4.2 นี้

1. สัญญาณจากเครื่องสร้างการสั่นสะเทือน

สำหรับเครื่องจักรแบบที่ 1 สัญญาณการสั่นสะเทือนที่ได้จะมาจากการวัดเครื่องสร้างการสั่นสะเทือนซึ่งมีความซับซ้อนต่ำและสามารถควบคุมลักษณะการสั่นได้ตามที่ต้องการ โดยจะถูกแสดงเป็นแผนภาพ ตามรูปที่ 4.9 และ รูปที่ 4.10 ด้านล่าง



รูปที่ 4.9 แผนภาพแสดงการต่ออุปกรณ์ในการทดสอบเชิงหลักการ



รูปที่ 4.10 ตัวอย่างการติดตั้งเซ็นเซอร์เข้ากับเครื่องสร้างการสั่นสะเทือน

จากรูปที่ 4.9 รูปที่ 4.10 แสดงให้เห็นถึงวิธีแผนผังและตำแหน่งของการติดตั้งอุปกรณ์ทั้งหมดที่ใช้ในการวัด โดยการที่จะทำให้เกิดการสั่นตามความถี่ที่ต้องการ เครื่องสร้างการสั่นสะเทือนจะได้รับสัญญาณมาจากเครื่องกำเนิดสัญญาณ ที่ถูกขยายมาแล้วผ่านเครื่องขยายสัญญาณ สำหรับระบบเซ็นเซอร์ก็จะใช้ตามที่ได้เคยกล่าวไปแล้วในย่อหน้าแรกของหัวข้อ 4.2 สำหรับรายละเอียดของอุปกรณ์แต่ละชิ้นสามารถดูเพิ่มเติมได้ในภาคผนวก ก.

สำหรับสัญญาณการสั่นสะเทือนของตัวสร้างการสั่นสะเทือน (Shaker) จะมีรูปแบบตามที่กำหนดด้วยเครื่องกำเนิดสัญญาณ (Signal Generator) โดยในการทดสอบแบ่งได้เป็น 2 แบบคือ สัญญาณที่มีความถี่เดียว และสัญญาณมอดูเลชันที่ประกอบด้วยความถี่หลายความถี่ ซึ่งจะถูกแสดงรายละเอียดตามตารางที่ 4.2 ด้านล่าง

ตารางที่ 4.2 รูปแบบของสัญญาณที่ใช้ทดสอบทั้งหมด จากเครื่องกำเนิดสัญญาณ

ทดสอบแบบที่ 1	ทดสอบแบบที่ 2
สัญญาณความถี่เดียวที่ 300 เฮิรตซ์	สัญญาณความถี่ 300 เฮิรตซ์ มอดูเลต 50 เฮิรตซ์
สัญญาณความถี่เดียวที่ 500 เฮิรตซ์	สัญญาณความถี่ 500 เฮิรตซ์ มอดูเลต 100 เฮิรตซ์
สัญญาณความถี่เดียวที่ 800 เฮิรตซ์	สัญญาณความถี่ 800 เฮิรตซ์ มอดูเลต 100 เฮิรตซ์
สัญญาณความถี่เดียวที่ 1000 เฮิรตซ์	สัญญาณความถี่ 1000 เฮิรตซ์ มอดูเลต 200 เฮิรตซ์
สัญญาณความถี่เดียวที่ 1200 เฮิรตซ์	สัญญาณความถี่ 1200 เฮิรตซ์ มอดูเลต 200 เฮิรตซ์
สัญญาณความถี่เดียวที่ 1500 เฮิรตซ์	สัญญาณความถี่ 1500 เฮิรตซ์ มอดูเลต 200 เฮิรตซ์
สัญญาณความถี่เดียวที่ 1800 เฮิรตซ์	สัญญาณความถี่ 1800 เฮิรตซ์ มอดูเลต 200 เฮิรตซ์

ตารางที่ 4.2 แสดงค่าของความถี่ที่จะถูกตั้งค่าในเครื่องกำเนิดสัญญาณเพื่อทดสอบแบบที่ 1 ที่เป็นสัญญาณความถี่เดียว และแบบที่ 2 ที่เป็นสัญญาณแบบมอดูเลต ซึ่งจะมีความถี่ตั้งแต่ 300 - 1800 เฮิรตซ์ โดยจะทำการเก็บข้อมูลเป็นระยะเวลา 10 วินาทีต่อ 1 สัญญาณ

2. สัญญาณจาก คอมเพรสเซอร์ 1 ลูกสูบ และ แบบ 2 ลูกสูบ

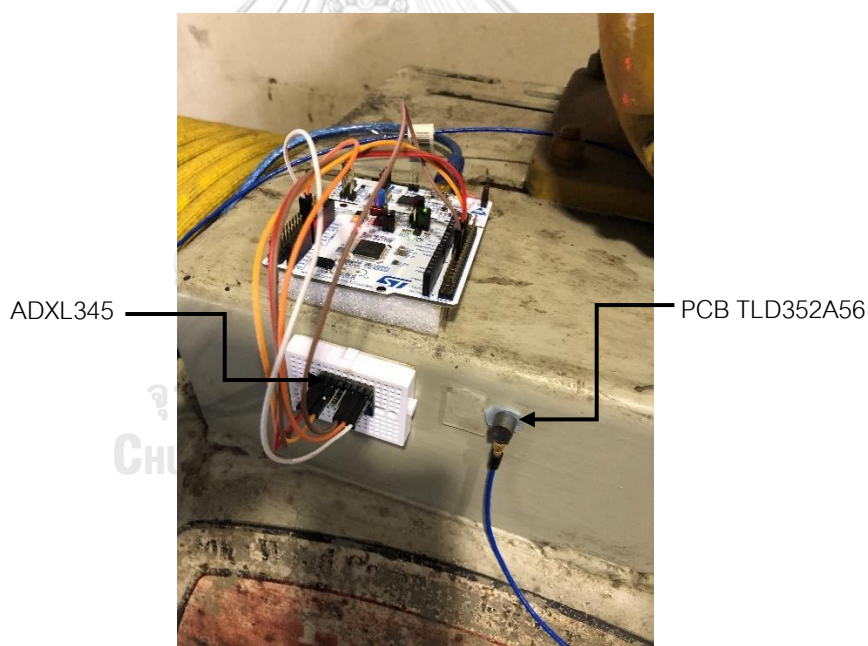
การทดสอบวัดการสั่นสะเทือนของเครื่องจักรจริง ทำโดยวัดสัญญาณจากเครื่องคอมเพรสเซอร์แบบลูกสูบ 2 เครื่อง โดยเป็นเครื่องแบบ 1 ลูกสูบและแบบ 2 ลูกสูบ ด้วยระบบการวัดแบบเดียวกันกับการวัดสัญญาณจากเครื่องสร้างการสั่นสะเทือนในหัวข้อที่ 1

การวัดสัญญาณของเครื่องคอมเพรสเซอร์แบบ 1 ลูกสูบ

เครื่องคอมเพรสเซอร์แบบ 1 ลูกสูบที่ทำการวัดนี้มีสภาพค่อนข้างดี ความเร็วรอบและการสั่นสะเทือนมีความเสถียร

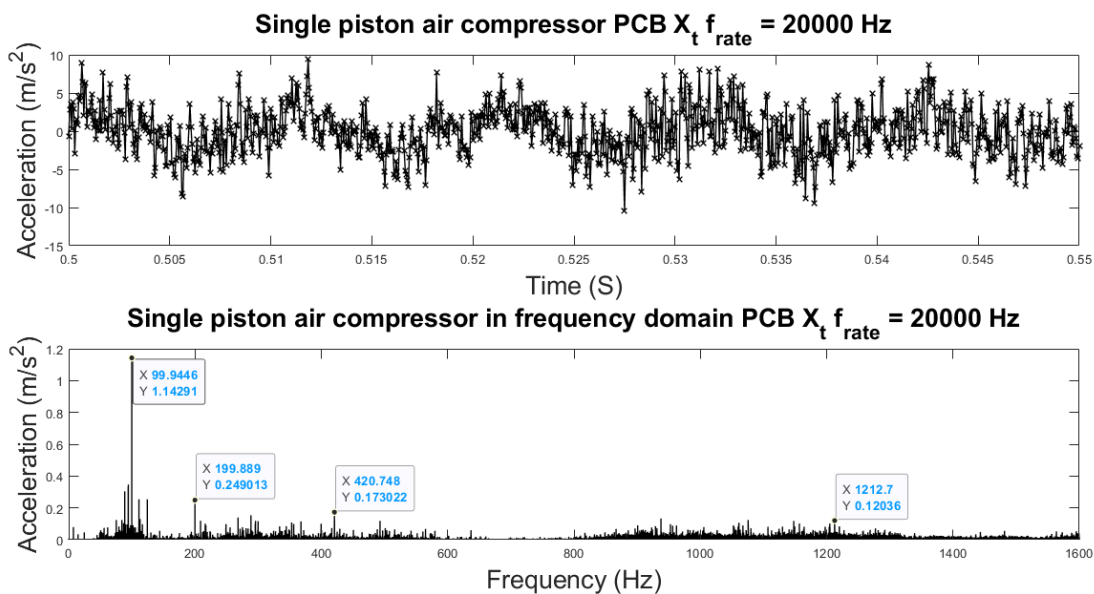


รูปที่ 4.11 เครื่องคอมเพรสเซอร์แบบ 1 ลูกสูบ และการติดตั้งระบบการวัด



รูปที่ 4.12 ระบบการวัดบนคอมเพรสเซอร์ 1 ลูกสูบ

จากรูปที่ 4.11 และ 4.12 จะเป็นรูปที่แสดงถึงเครื่องคอมเพรสเซอร์ที่ถูกวัด และตำแหน่งที่จะวัด



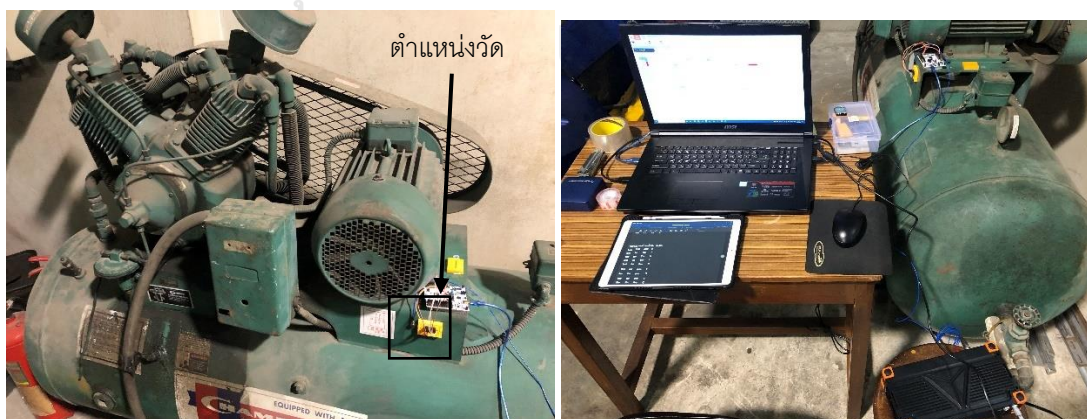
รูปที่ 4.13 สัญญาณการสั่นสะเทือนของคอมเพรสเซอร์ 1 ลูกสูบ

จากรูปที่ 4.13 เป็นลักษณะการสั่นสะเทือนของคอมเพรสเซอร์แบบ 1 ลูกสูบ ซึ่งจะมีตำแหน่งความถี่หลักอยู่ 3 ตำแหน่ง และจะมีลักษณะของแถบความถี่บริเวณ 200 - 600 เฮิรตซ์ และ 800 - 1400 เฮิรตซ์ มีความหนาแน่นของความถี่มากและมีแอมพลิจูดที่ต่ำ

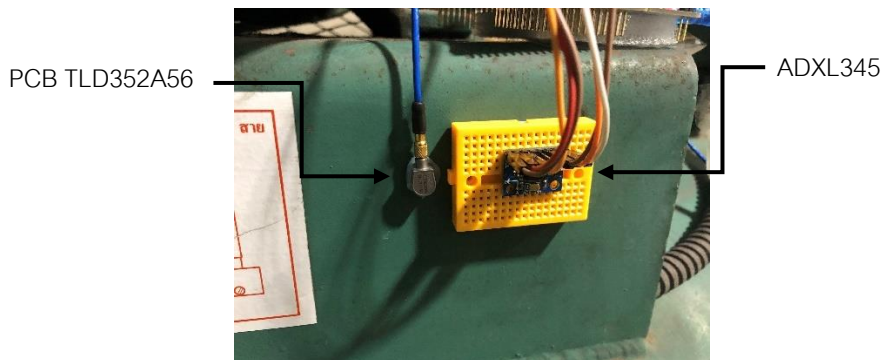
การวัดสัญญาณของเครื่องคอมเพรสเซอร์แบบ 2 ลูกสูบ

เครื่องคอมเพรสเซอร์แบบ 2 ลูกสูบที่ทำการวัดนี้ สภาพของเครื่องจักรไม่อยู่ในสภาพที่สมบูรณ์นัก และการเดินเครื่องความไม่เสถียรอยู่บ้างในระหว่างการทำงาน

จุฬาลงกรณ์มหาวิทยาลัย

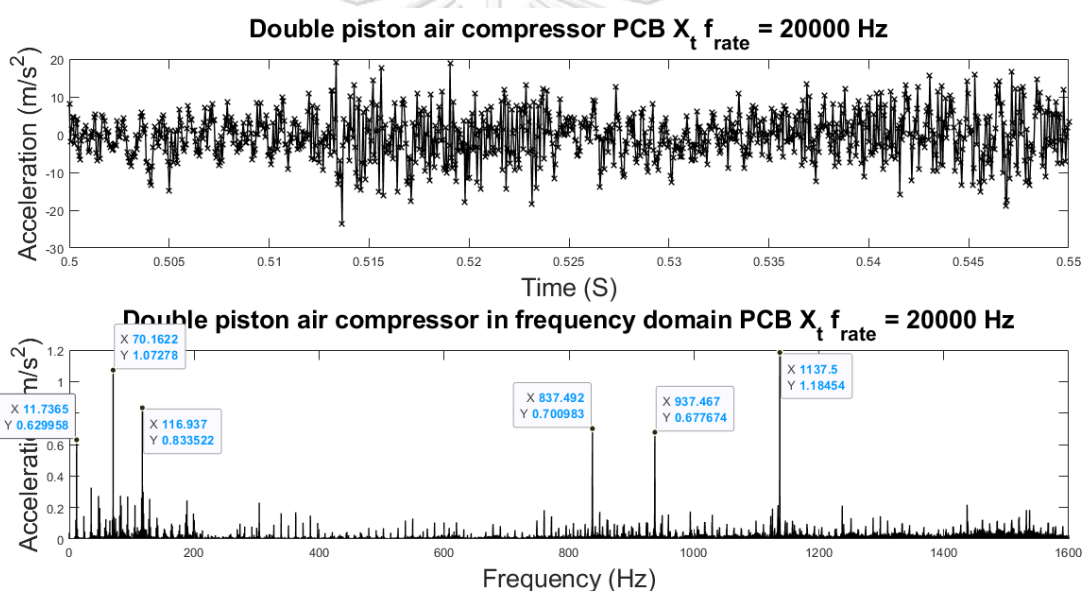


รูปที่ 4.14 เครื่องคอมเพรสเซอร์แบบ 2 ลูกสูบ และการติดตั้งระบบการวัด



รูปที่ 4.15 ระบบการวัดบนคอมเพรสเซอร์ 1 ลูกสูบ

จากรูปที่ 4.14 และ 4.15 จะเป็นรูปที่แสดงถึงเครื่องคอมเพรสเซอร์ที่ถูกวัด และตำแหน่งที่จะวัด สำหรับลักษณะทั่วไปของสัญญาณ



รูปที่ 4.16 สัญญาณการสั่นสะเทือนของคอมเพรสเซอร์ 2 ลูกสูบ

จากรูปที่ 4.16 เป็นลักษณะการสั่นสะเทือนของคอมเพรสเซอร์แบบ 2 ลูกสูบ มีความถี่หลักอยู่ประมาณ 6 ตำแหน่ง ลักษณะของแถบสัญญาณรบกวนเกิดขึ้นตลอดพิสัยของการวัด และไม่ได้มีความหนาแน่นมากเท่ากับแถบสัญญาณรบกวนของคอมเพรสเซอร์ 1 ลูกสูบ

4.3 การทดสอบหาความเป็นไปได้ในทางปฏิบัติจริง (Proof of work)

ในการทดสอบแบบนี้จะเป็นการทดสอบเพื่อให้ทราบว่าคอมเพรสเซอร์ซึ่งสามารถจะนำไปใช้งานจริงในการวัดสัญญาณแบบอัตราสุ่มจุดไม่แน่นอนโดยตรงกับเครื่องจักรจะสามารถสร้างสัญญาณคืนกลับที่ถูกต้องได้หรือไม่ การวัดสัญญาณแบบอัตราสุ่มจุดไม่แน่นอนจะต้องเขียนโปรแกรมบนไมโครคอนโทรลเลอร์เพื่อให้

สามารถควบคุมการปรับอัตราสุ่มจุดได้และต้องส่งข้อมูลการสั่นสะเทือนและเวลาได้ถูกต้อง สำหรับหลักการการปรับอัตราสุ่มจุดบนไมโครคอนโทรลเลอร์และโปรแกรมที่ใช้ควบคุมสามารถดูเพิ่มได้ในภาคผนวก ค. ง. และ ฉ.

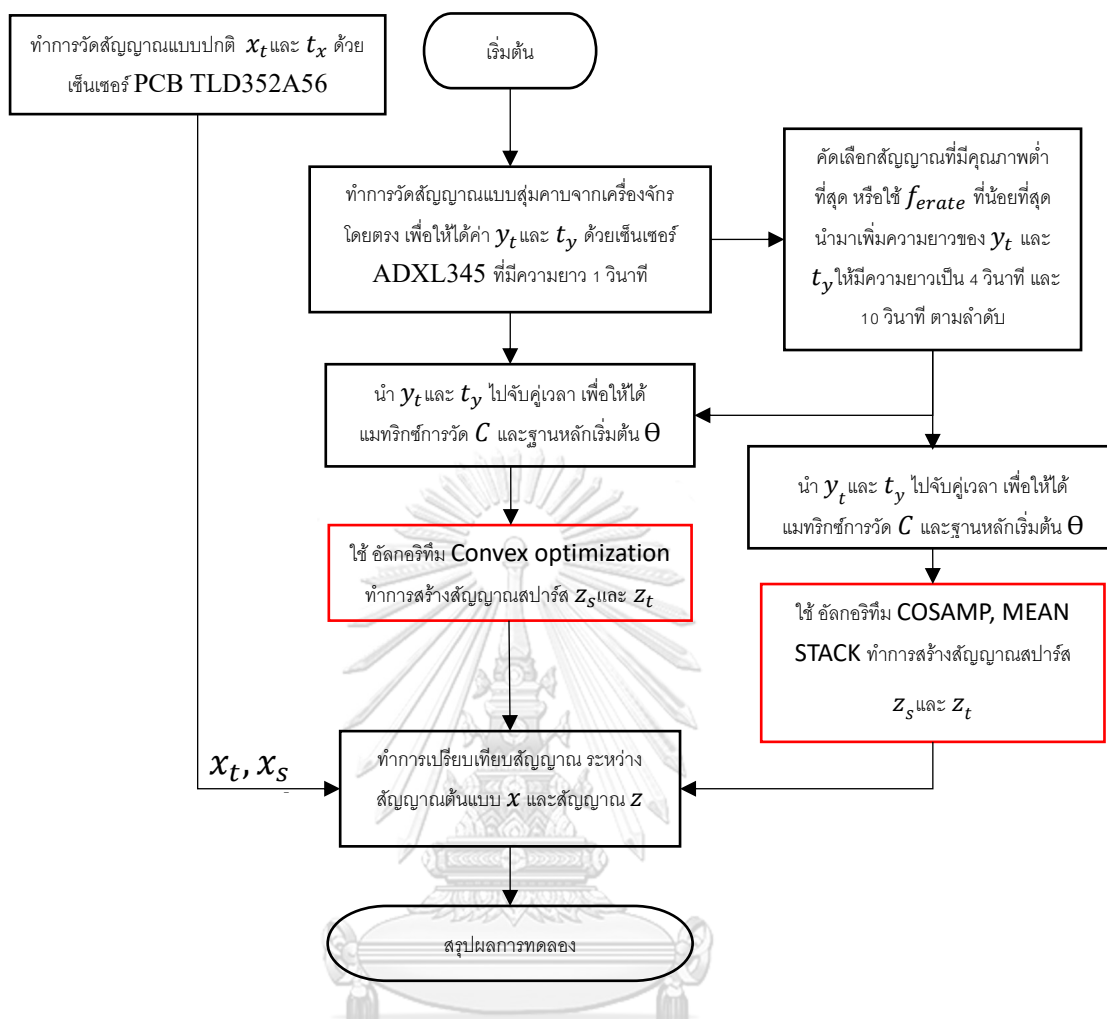
การวัดจะวัดแบบปรับอัตราสุ่มจุดให้ ค่า y_t และ t_y มีอัตราการสุ่มจุดเทียบเท่าแตกต่างกันทั้งหมด 8 แบบคือ 1600, 1000, 800 และ 500 เฮิร์ตซ์ โดยแต่ละแบบจะปรับพิสัยของการสุ่มค่า f_{range} อีก 2 ค่าซึ่งสามารถดูค่าปรับตั้งได้อย่างละเอียดในตารางที่ 4.3 โดยระยะเวลาในการวัดคือ 30 วินาที แต่ในเบื้องต้นจะใช้ความยาวข้อมูลเพียง 1 วินาทีมาทำการทดสอบเท่านั้นเพื่อนำไปสร้างสัญญาณ Z_t และ Z_s ที่มีอัตราการสุ่มจุด 3200 เฮิร์ตซ์ ด้วยคอมพิวเตอร์เซตซิง หลังจากนั้นจึงจะเลือกสัญญาณบางส่วนที่มีผลลัพธ์ที่แย่ที่สุดไปเพิ่มความยาวข้อมูลจากเดิม 1 วินาที ให้ยาวเพิ่มมากขึ้นเพื่อดูว่าจะสามารถปรับปรุงผลลัพธ์ให้มีคุณภาพมากขึ้นได้หรือไม่

ตารางที่ 4.3 แสดงการปรับตั้ง ระบบการวัดแบบต่างๆที่ใช้ในการทดสอบ

แบบที่	PSC	ARR	Random()	$N_{y,1}$			f_{range}
				min	mean	Max (f_{erate})	
1	20	500	1	1,059.60	1,273.89	1,600.00	540.40
2	20	500	4	526.32	790.51	1,600.00	1,073.68
3	20	800	1	758.29	862.07	1,000.00	241.71
4	20	800	4	439.56	609.76	1,000.00	560.44
5	20	1,000	1	637.45	709.22	800.00	162.55
6	20	1,000	4	396.04	529.10	800.00	403.96
7	80	400	1	305.34	378.79	500.00	194.66
8	80	400	4	140.85	219.30	500.00	359.15

สำหรับการทดสอบกับเครื่องจักรจริงใน นั้นย่านความถี่สูงสุดที่ต้องการวัดจะอยู่ที่ 1600 เฮิร์ตซ์ และในทุกการปรับค่าตามตารางที่ 4.3 จะใช้สัญญาณ y_t ที่มีค่า f_{erate} อย่างมากที่สุดคือ 1600 เฮิร์ตซ์ และน้อยที่สุดจะอยู่ที่ 500 เฮิร์ตซ์ แต่ในการวัดแบบทั่วไปนอกเหนืองานวิจัยนี้ จำเป็นจะต้องใช้ความถี่สุ่มของสัญญาณอย่างน้อย 3200 เฮิร์ตซ์ จึงจะสามารถตรวจวัดสัญญาณที่มีพิสัยกว้าง 1600 เฮิร์ตซ์ได้ ตามทฤษฎีการสุ่มตัวอย่าง (Nyquist's sampling theorem) แต่ในงานวิจัยนี้พยายามจะหาความเป็นไปได้ที่จะใช้คอมพิวเตอร์เซตซิงเพื่อลดข้อจำกัดดังกล่าว

การทดสอบนี้จะใช้เครื่องจักร 2 แบบเช่นเดียวกับหัวข้อที่ 4.2.2 คือเครื่องสร้างการสั่นสะเทือน (วัดสัญญาณทั้งหมด 14 แบบเช่นเดียวกัน) และคอมพิวเตอร์แบบลูกสูบ 2 เครื่อง (วัดสัญญาณในตำแหน่ง Horizontal เช่นเดียวกัน)



รูปที่ 4.17 แผนผังแสดงวิธีการทำการทดสอบการวัดแบบสุ่มคาบโดยตรง

จากรูปที่ 4.17 เป็นแผนผังอธิบายขั้นตอนในการทดสอบ จะเห็นว่าจะมีการเลือกใช้สัญญาณที่หลากหลายรูปแบบและหลากหลายวิธีการปรับตั้ง การทดสอบในขั้นแรกจะทำหลังจากการทดสอบที่ความยาว 1 วินาทีก่อนหน้าเสร็จเรียบร้อยแล้วแล้วจะหลังจากนั้นจะทำการเลือกผลการทดสอบที่สัญญาณที่มีคุณภาพต่ำที่สุด และสัญญาณที่มีความถี่สุ่มเทียบเท่าที่น้อยที่สุด มาเพิ่มความยาวของสัญญาณให้ยาวมากขึ้น เป็น 4 วินาที และ 10 วินาที ขึ้นอยู่กับสัญญาณที่ถูกเลือกขึ้นมา เพื่อดูว่าการเพิ่มความยาวของสัญญาณมากขึ้น จะช่วยเพิ่มคุณภาพของสัญญาณคืนกลับได้หรือไม่ หลังจากนั้นจะใช้สัญญาณเดียวกันนี้กับอัลกอริทึมที่เหลือทั้งหมด 2 แบบ คือ COSAMP และ MEANSTACK จากหัวข้อที่ 3.3.2 และ 3.3.3 แทนการหาค่าเหมาะสมที่สุดแบบคอนเวกซ์ เพื่อเปรียบเทียบคุณภาพและลักษณะของความแตกต่างของสัญญาณที่แต่ละอัลกอริทึมทำได้

4.4 การประเมินสัญญาณ

สำหรับการวัดคุณภาพของสัญญาณสปาร์ส จะใช้การประเมินผ่านการคำนวณพารามิเตอร์ดังนี้

1.) สหสัมพันธ์ไขว้ (Cross correlation) เป็นการวัดความเหมือนของสัญญาณ 2 สัญญาณ สำหรับงานวิจัยนี้คือระหว่างสัญญาณ x_t และ z_t ซึ่งจะสามารถทำได้โดยใช้หลักการของการหาผลคูณแบบจุดระหว่าง 2 สัญญาณนี้ พร้อมทั้งเลื่อนสัญญาณ z_{t+k} ไป k ตำแหน่งทุกครั้งตลอดการหาผลคูณ ซึ่งจะเขียนอยู่ในรูป

$$\hat{R}_{xz}(k) = C \sum_{k=-(2N-1)}^{2N-1} x_t \cdot z_{t-k} \quad (4.1)$$

กำหนดให้ k คือ จำนวนเต็มใดๆ มีค่าเท่ากับ $(2N - 1) \leq k \leq -(2N - 1)$

N คือ จำนวนจุดทั้งหมดของข้อมูล x_t หรือ z_t

C คือ ค่าสำหรับปรับบรรทัดฐาน (Normalization) ซึ่งเท่ากับ $\frac{1}{\sqrt{R_{xx}R_{zz}}}$

โดย $R_{xx} = \sum_{k=-(2N-1)}^{2N-1} x_t \cdot x_{t-k}$

จากสมการที่ (4.1) จะได้ค่าสหสัมพันธ์ไขว้ของคู่สัญญาณที่นำมาเปรียบเทียบเป็นจำนวนทั้งหมด $2(2N - 1) + 1$ ค่า แต่ข้อมูลที่สำคัญคือ ค่า $\hat{R}_{xz}(k)$ ที่มากที่สุดและตำแหน่งของ k ที่ให้ค่า $\hat{R}_{xz}(k)$ ที่มากที่สุด ซึ่งค่า $\hat{R}_{xz}(k)$ จะมีขนาดตั้งแต่ 0 ถึง 1 ซึ่งถ้าค่าเข้าใกล้ 1 แสดงว่าสัญญาณ x_t และ z_t ก็จะมีแนวโน้มใกล้เคียงกันมากขึ้น สำหรับค่า k คือตำแหน่งที่สัญญาณ z_t ถูกเลื่อนไปเพื่อให้สัญญาณ z_t มีความใกล้เคียงกับสัญญาณ x_t มากที่สุด ซึ่งสามารถบอกเป็นเวลาที่ดูจากค่าสมาชิกของ t_z ในตำแหน่งที่ k

2.) ระดับความสปาร์ส (Sparsity) เนื่องจากการคำนวณระดับความสปาร์สในปัจจุบันมีหลากหลายวิธีการแต่ละวิธีการก็จะให้ค่าที่แตกต่างกัน สำหรับงานวิจัยนี้จะเลือกใช้วิธีการคำนวณแบบเดียวกับ งานวิจัยของ S.-X. C. Yi-Qing [10] ซึ่งคำนวณระดับความสปาร์สของสัญญาณใดๆ สามารถเขียนเป็นสมการได้ดังนี้

$$L_s = N_{0.01}/N \quad (4.2)$$

กำหนดให้ $N_{0.01}$ คือ จำนวนแอมพลิจูดที่มีค่ามากกว่า 1 เปอร์เซนต์ของค่าแอมพลิจูดสูงสุดในโดเมนความถี่

ซึ่งถ้าหากค่า L_s นี้มีค่าน้อย บ่งบอกว่าระดับของความสปาร์สของสัญญาณมีมาก

3.) อัตราส่วนของผลรวมสัญญาณตกค้างกำลังสองกับสัญญาณต้นแบบ (Residual sum of square signal ratio, RSS) เป็นการวัดความถูกต้องของสัญญาณที่นำวิธีคำนวณอัตราส่วนระหว่างผลรวมของสัญญาณตกค้างกำลังสองและสัญญาณต้นแบบทำให้ทราบว่ามีความคลาดเคลื่อนของสัญญาณสปาร์สซึ่งวัดได้จาก

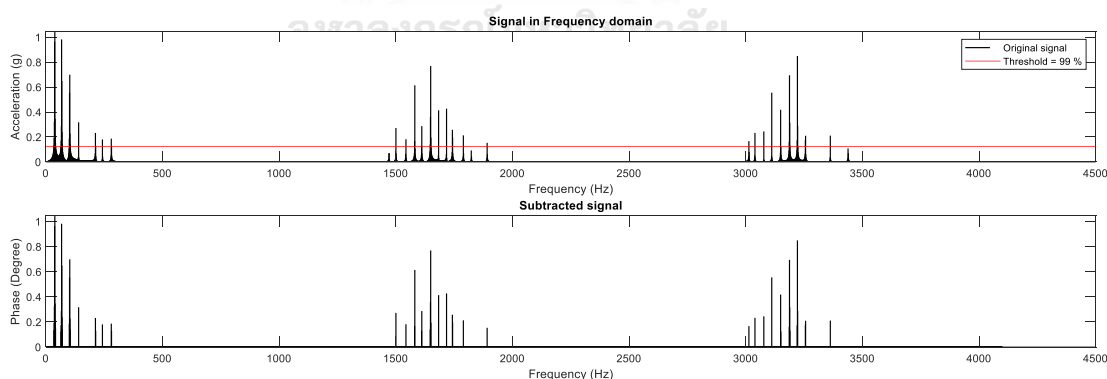
สัญญาณตกค้างคิดเป็นอัตราส่วนเท่าไรเมื่อเทียบกับสัญญาณต้นแบบเพื่อใช้ในการประเมินคุณภาพของสัญญาณสปาร์ส ที่ถูกสร้างขึ้นใหม่ [10] กำหนดให้ ตัวแปร \mathbf{x} คือสัญญาณต้นแบบ \mathbf{Z} คือสัญญาณสปาร์สที่ถูกสร้างขึ้นใหม่ โดยอัลกอริทึมมีสมการดังนี้

$$RSS = \left(\frac{\|\mathbf{x} - \mathbf{z}\|_2^2}{\|\mathbf{x}\|_2^2} \right) \quad (4.3)$$

จากสมการที่ (4.2) จะเห็นว่าสมการนี้จะทำการเปรียบเทียบสัญญาณต้นแบบ และสัญญาณที่ถูกสร้างขึ้นใหม่ถ้าหากสัญญาณสปาร์ส \mathbf{Z} มีความคล้ายคลึงกับสัญญาณต้นแบบ \mathbf{x} มาก ก็จะทำให้ค่า พารามิเตอร์น้อยลงด้วย

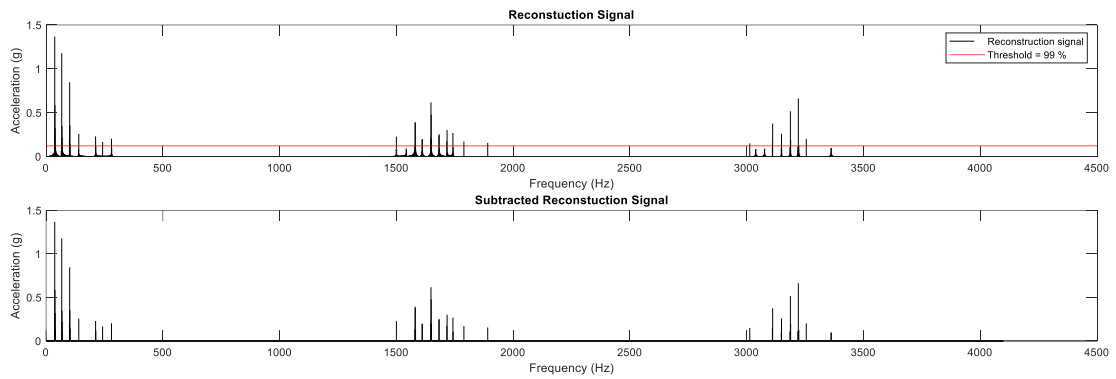
4.) การเปรียบเทียบความถูกต้องของตำแหน่งความถี่ของการสร้างสัญญาณคีนกลับ

พารามิเตอร์ที่ใช้ในการประเมินสัญญาณสปาร์สต่อไปนี้จะใช้การเปรียบเทียบตำแหน่งของความถี่ที่มีแอมพลิจูดสูงระหว่างสัญญาณต้นแบบ \mathbf{x}_S ในโดเมนความถี่ และสัญญาณสปาร์ส \mathbf{Z}_S ว่ามีความถูกต้องมากน้อยเพียงใด ด้วยความที่สัญญาณสปาร์ส \mathbf{Z}_S ถูกสร้างขึ้นมาด้วยสมมุติฐานที่ว่า \mathbf{x}_S ต้องมีความสปาร์สที่สูงทำให้ ค่าตอบของ \mathbf{Z}_S ในโดเมนความถี่นั้นมันมักมีจำนวนสมาชิกที่ไม่เท่ากับ 0 อยู่น้อย (แต่ก็ยังพอมีสัญญาณรบกวนที่แอมพลิจูดต่ำอยู่บ้าง) ต่างกับสัญญาณ \mathbf{x}_S ที่เป็นสัญญาณจริงซึ่งสมาชิกทุกตัวบนโดเมนความถี่จะไม่เท่ากับ 0 เลยการจะนำสัญญาณทั้ง 2 แบบนี้มาเปรียบเทียบกันได้นั้นจะเป็นจะต้องเปรียบเทียบกันเฉพาะตำแหน่งของความถี่ที่มีแอมพลิจูดสูงเท่านั้น ทำให้ก่อนการเปรียบเทียบ แอมพลิจูดของสัญญาณ \mathbf{x}_S ที่มีค่าน้อยๆ (ต่ำกว่าเปอร์เซ็นต์ที่ 99 ของแอมพลิจูดของสัญญาณทั้งหมด) จะถูกตัดออกให้เท่ากับศูนย์ก่อน เรียกวิธีนี้ว่า "Spectral subtraction"



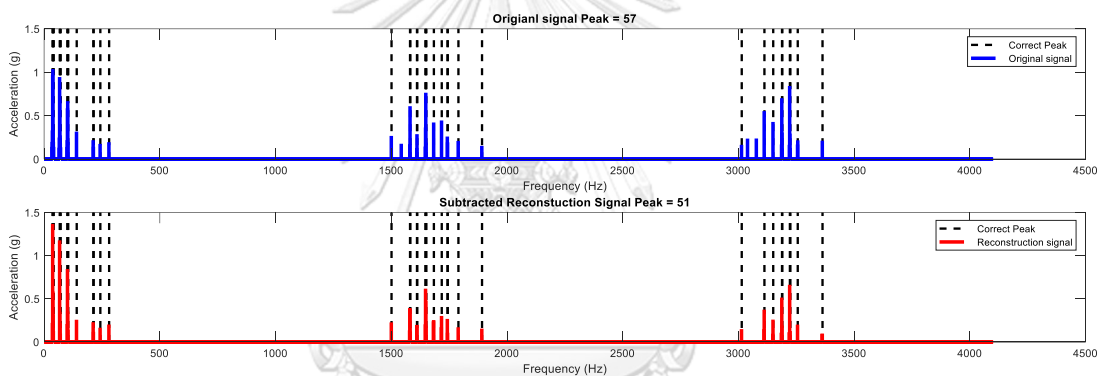
รูปที่ 4.18 การตัดสัญญาณ \mathbf{x}_S ในบริเวณที่มีแอมพลิจูดน้อยกว่าที่กำหนดก่อนทำการเปรียบเทียบ

รูปที่ 4.18 จะเห็นว่าเดิมนั้นสัญญาณ \mathbf{x}_S มีแอมพลิจูดที่ไม่เท่ากับ 0 อยู่ทุกความถี่ไม่ว่าจะมากหรือน้อยจึงจำเป็นต้องตัด สัญญาณที่มีขนาดน้อยออกก่อนจะนำไปเปรียบเทียบกับสัญญาณ \mathbf{Z}_S



รูปที่ 4.19 การตัดสัญญาณ Z_S ในบริเวณที่มีแอมพลิจูดน้อยกว่าที่กำหนดก่อนทำการเปรียบเทียบ

รูปที่ 4.19 เป็นการตัดสัญญาณบางส่วนของ Z_S ออกเช่นกันเพื่อให้การเปรียบเทียบสัญญาณเป็นสัญญาณที่มีเปอร์เซ็นต์ไพล์ของแอมพลิจูดหรือ 99%



รูปที่ 4.20 การตรวจสอบตำแหน่งความถี่ระหว่างสัญญาณ X_S (รูปด้านบน) และ สัญญาณ Z_S (รูปด้านล่าง)

รูปที่ 4.20 เป็นการเป็นการตรวจสอบตำแหน่งความถี่ระหว่างสัญญาณ X_S และสัญญาณ Z_S กำหนดให้เส้นประเป็นตำแหน่งสัญญาณที่มีตำแหน่งความถี่ตรงกัน (Matched Peak) แต่จะสังเกตว่าทั้งสัญญาณ X_S และ Z_S ก็จะมีตำแหน่งความถี่ที่ไม่ตรงกัน โดยถ้าตำแหน่งที่ไม่ตรงปรากฏบนกราฟสัญญาณ X_S จะถือว่าเป็นความถี่ที่ไม่สามารถกู้คืนมาได้ (Unrecoverable Peak) แต่ถ้าหากเป็นตำแหน่งที่ไม่ตรงปรากฏบนกราฟสัญญาณ Z_S จะถือว่าเป็นความถี่หลอก (Faults Peak) โดย ตำแหน่งความถี่ทั้งสามแบบ สามารถนำไปคำนวณเป็นเปอร์เซ็นต์ความถูกต้องของสัญญาณได้ดังนี้ กำหนดให้ MP เป็นจำนวนของความถี่ทั้งหมดที่ตรงกัน P_x เป็นจำนวนความถี่ทั้งหมดที่ไม่เท่ากับศูนย์ของสัญญาณ X_S เมื่อทำการตัดแอมพลิจูดที่มีค่าต่ำออกหมดแล้ว P_z เป็นจำนวนความถี่ทั้งหมดที่ไม่เท่ากับศูนย์ของสัญญาณ Z_S เมื่อทำการตัดแอมพลิจูดที่มีค่าต่ำออกหมดแล้ว

$$\% \text{ Matched Peak} = \frac{MP}{P_z} \times 100 \quad (4.4)$$

$$\% \text{ Faults Peak} = \frac{(P_z - MP)}{P_z} \times 100 \quad (4.5)$$

$$\% \text{ Unrecoverable Peak} = \frac{(P_x - MP)}{P_x} \times 100 \quad (4.6)$$

สำหรับสัญญาณตัวอย่างที่กล่าวถึงในรูปที่ 4.18, 4.19, 4.20 นี้ เมื่อนำไปประเมินโดย พารามิเตอร์ข้างต้นทั้งหมดแล้วจะได้ค่าดังนี้

ตารางที่ 4.4 แสดงค่าพารามิเตอร์ของสัญญาณตัวอย่างในรูปที่ 4.18, 4.19, 4.20

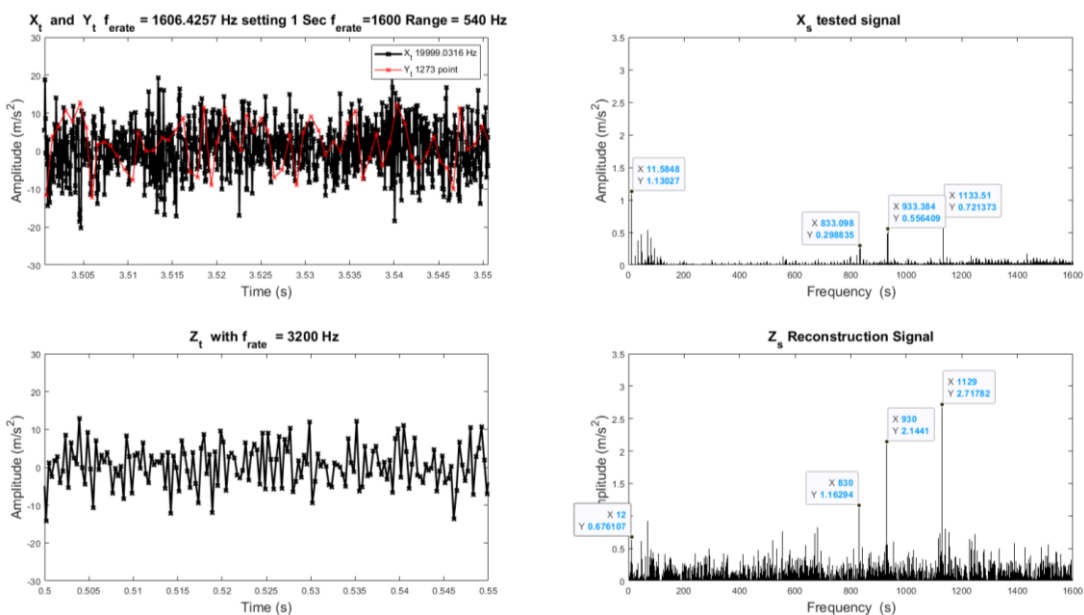
Compression Ratio	3.9632
RSNR	7.6648
Matched Peak	76.36%
Faults Peak	23.63%
Unrecoverable Peak	26.31%

ระดับของแอมพลิจูดในการเลือกเพื่อทำ Spectral subtraction ขึ้นอยู่กับความสปร้าสของสัญญาณ เช่นถ้าสัญญาณ X_S มีความสปร้าสสูงก็สามารถที่จะใช้ระดับของแอมพลิจูดที่ต่ำได้โดยไม่สูญเสียรายละเอียดของสัญญาณ แต่ถ้าหากสัญญาณมีความสปร้าสน้อย หรือมีความเป็น broadband สูงการใช้ระดับของแอมพลิจูดที่ต่ำเกินไปจะทำให้สัญญาณรบกวนยังหลงเหลืออยู่ในการประเมินสัญญาณ

4.) การเปรียบเทียบลักษณะของสัญญาณในโดเมนความถี่ด้วยการประเมินด้วยระดับความคล้ายคลึงกันของสัญญาณ

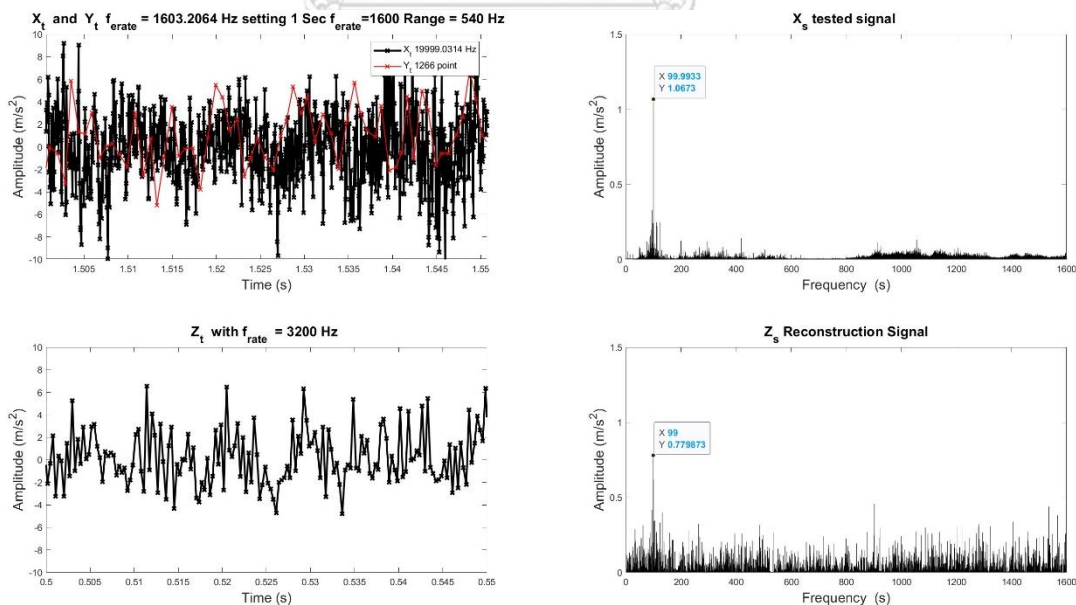
เนื่องจากการประเมินในหัวข้อที่ผ่านมา จะใช้การคำนวณเข้ามาเกี่ยวข้องทั้งหมดเช่น เปรียบเทียบตำแหน่งความถี่ อัตราส่วนสัญญาณบีบอัด อัตราส่วนของผลรวมสัญญาณตกค้างกำลังสองกับสัญญาณต้นแบบ ซึ่งเหมาะสำหรับการประเมินผลในการจำลองสัญญาณในคอมพิวเตอร์เนื่องจากมีความแน่นอนของตำแหน่งความถี่และแอมพลิจูดที่สามารถควบคุมได้ แต่ในการวัดสัญญาณจริงมีความแน่นอนของตำแหน่งความถี่และแอมพลิจูดและลักษณะของสัญญาณต่างๆ เช่น การรั่วของความถี่ (leakage) ไซด์แบนด์ของสัญญาณ แถบความถี่รบกวน มีความคลาดเคลื่อนอยู่ตลอดทุกครั้งที่ทำการวัด ทำให้การประเมินด้วยการคำนวณไม่สอดคล้องกับสภาพความเป็นจริงของสัญญาณคืนกลับที่ถูกประเมิน วิธีการนี้ผู้วิจัยจะทำการประเมินด้วยการใช้ดูลยพินิจของตัวผู้วิจัยเองว่าสัญญาณคืนกลับที่เกิดขึ้นมีความใกล้เคียงกับสัญญาณต้นแบบมากน้อยขนาดไหน โดยแบ่งเป็น 3 ระดับซึ่งจะแสดงผลเป็นสีตามลำดับคือ เขียว ส้ม แดง ดังนี้

4.1 ระดับสีเขียว สัญญาณคืนกลับที่ถูกสร้างขึ้นมีตำแหน่งแอมพลิจูดที่ใกล้เคียงกับสัญญาณต้นแบบ และสามารถแยกแยะสัญญาณรบกวนที่เป็นพื้นหลังกับสัญญาณจริงได้ชัดเจนมีการวางตัวของแพทเทิร์นสัญญาณที่ถูกต้องคล้ายคลึงกับสัญญาณต้นแบบ ซึ่งจะแสดงตัวอย่างตามรูปด้านล่าง



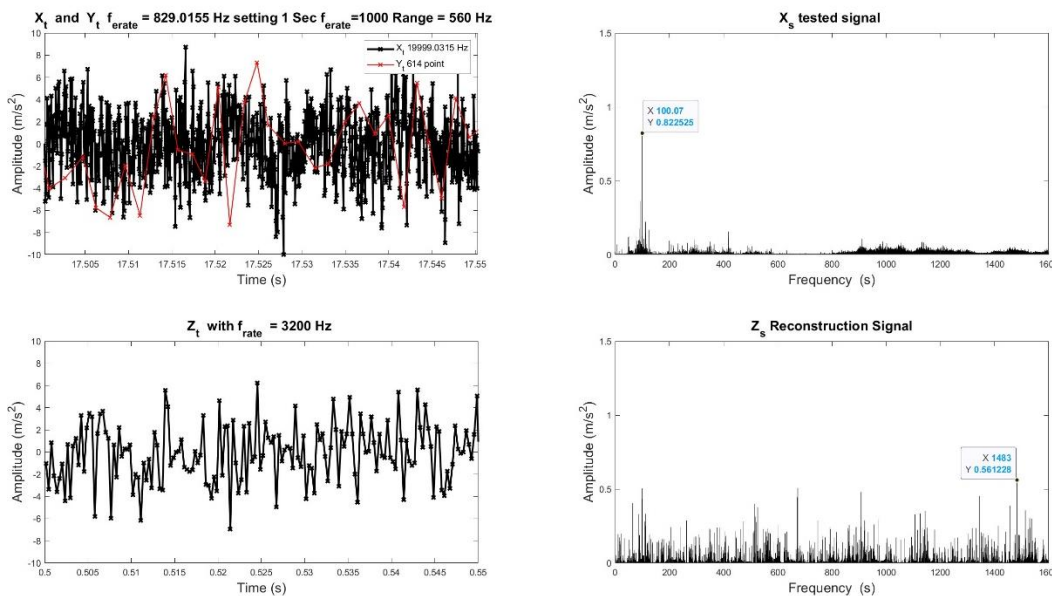
รูปที่ 4.21 ลักษณะของสัญญาณคืนกลับที่ได้รับการประเมินเป็น สีเขียว

4.2 ระดับสีส้ม สัญญาณคืนกลับมีสัญญาณรบกวนสูงไม่สามารถแยกแยะระหว่าง สัญญาณจริงเกิดขึ้นและแถบสัญญาณรบกวน ได้อย่างชัดเจนทั้งหมดแต่ยังพอจะสามารถทราบความถี่จริงที่เกิดขึ้นและลักษณะของสัญญาณที่ถูกต้องได้บ้าง ซึ่งจะแสดงตัวอย่างตามรูปด้านล่าง



รูปที่ 4.22 ลักษณะของสัญญาณคืนกลับที่ได้รับการประเมินเป็น สีส้ม

4.3) ระดับสีแดง หมายถึงสัญญาณคืนกลับมีแถบสัญญาณรบกวนสูงมากและไม่สามารถแยกแยะหรือหาความคล้ายคลึงกับสัญญาณต้นแบบได้เลย



รูปที่ 4.23 ลักษณะของสัญญาณคืนกลับที่ได้รับการประเมินเป็น สีแดง

วิธีการประเมินแบบดังกล่าวนี้จะถูกใช้ในการประเมินสัญญาณที่เกิดจากการวัดสัญญาณจริง ทั้งสัญญาณจากการวัดเครื่องสร้างการสั่นสะเทือน และสัญญาณจากการวัดเครื่องจักรจริง ระหว่างผลลัพธ์จากระบบการวัดแบบมาตรฐาน และระบบการวัดแบบสุ่มคาบ แทนวิธีการคำนวณแบบปกติ

4.5 สรุปวิธีการทดสอบและจุดมุ่งหมายในการทดสอบ

จากวิธีการทดสอบที่ถูกแบ่งเป็น 2 รูปแบบเพื่อเป้าหมายที่แตกต่างกันคือ

1. การทดสอบหาตารางความเป็นไปได้ในการใช้คอมพิวเตอร์เซเชนซิงในเชิงหลักการ (Prove of concept) ซึ่งจะมีจุดมุ่งหมายในการทดสอบคือการนำไปใช้กับ การจำลองในคอมพิวเตอร์และ การวัดด้วยระบบการวัดมาตรฐานทั่วไป ที่มีความถูกต้องสูง ซึ่งทำให้ไม่จำเป็นต้องคำนึงถึงปัจจัยภายนอกที่ นอกเหนือไปจากตัวอัลกอริทึมเอง

2. การทดสอบหาความเป็นไปได้ในการใช้คอมพิวเตอร์เซเชนซิงในทางปฏิบัติ (Prove of work) ซึ่งจะมีจุดมุ่งหมายในการทดสอบคือการนำไปใช้กับ การวัดสัญญาณด้วยระบบการวัดที่ราคาถูก มีความถี่สุ่มต่ำ มาชดเชยด้วยข้อได้เปรียบของของคอมพิวเตอร์เซเชนซิงที่ไม่จำเป็นต้องการความถี่สุ่มที่สูงก็สามารถ สร้างสัญญาณที่มีความถี่สุ่มสูงได้

จากการทดสอบทั้งหมดที่กล่าวมาจะเห็นได้ว่าจะในงานวิจัยนี้จะพยายาม ทดสอบเพื่อให้ครอบคลุมการประยุกต์ใช้คอมพิวเตอร์เซเชนซิงในระดับที่แตกต่างกันทุกระดับตั้งแต่การจำลองในคอมพิวเตอร์ การใช้กับระบบการวัดทั่วไป ไปจนถึงการใช้กับระบบการวัดแบบสุ่มคาบโดยตรง เพื่อจะแสดงให้เห็นถึงความเป็นไปได้และข้อจำกัดของการนำคอมพิวเตอร์เซเชนซิงไปใช้ในหลายๆระดับ

สำหรับภาพรวมของวิธีการทดสอบในทั้งสองหัวข้อจะถูกแสดงในตารางด้านล่างโดยตารางจะถูกแบ่งออกเป็น 8 คอลัมน์ซึ่งมีความหมายดังนี้

คอลัมน์ที่ 1. จุดมุ่งหมายในการทดสอบ แบ่งเป็นการทดสอบในเชิงหลักการ และการทดสอบในเชิงปฏิบัติ

คอลัมน์ที่ 2. ระบบการวัดที่ใช้ แบ่งเป็น การใช้การจำลองสัญญาณด้วยคอมพิวเตอร์ การวัดแบบปกติ (คาบการสุ่มจุดข้อมูลคงที่) และการวัดแบบสุ่มคาบโดยตรง

คอลัมน์ที่ 3. แหล่งที่มาของสัญญาณต้นแบบเพื่อใช้ในการอ้างอิงความถูกต้องจะถูกแบ่งเป็นลักษณะของสัญญาณที่ใช้ในการวัด และเซ็นเซอร์ที่ใช้ในการวัด ซึ่งในหัวข้อนี้การสัญญาณที่ได้จะใช้การวัดแบบปกติทั้งหมด (คาบการสุ่มจุดข้อมูลคงที่)

คอลัมน์ที่ 4. เซ็นเซอร์ที่ใช้ทำการทดสอบ จะระบุถึงเซ็นเซอร์ที่ใช้ในระบบการวัดที่เลือกใน หัวข้อ 2 โดยแบ่งออกเป็น การจำลองขึ้นมาในคอมพิวเตอร์ การวัดโดยใช้ เซ็นเซอร์ PCB และการวัดด้วยใช้เซ็นเซอร์ ADXL345 แบบคาบการสุ่มไม่คงที่

คอลัมน์ที่ 5. แหล่งที่มาของสัญญาณแบบสุ่มคาบ แบ่งออกเป็น การจำลองขึ้นมาด้วยคอมพิวเตอร์ การสุ่มหยิบจุดข้อมูล และการวัดแบบสุ่มคาบโดยตรง

คอลัมน์ที่ 6 สัญญาณที่ทำการวัด จะแบ่งเป็น 3 แหล่งกำเนิด ด้วยกันคือ จำลองขึ้นจากฟังก์ชันคณิตศาสตร์ในคอมพิวเตอร์ จากเครื่องสร้างการสั่นสะเทือน และจากเครื่องคอมพิวเตอร์เซเชนซิง ทั้ง 2 ตัว

คอลัมน์ที่ 7 อัลกอริทึมที่ใช้ในการสร้างสัญญาณคืนกลับ มีทั้งหมด 3 แบบ คือ การหาค่าเหมาะสมที่สุดแบบคอนเวกซ์ การจับคู่สัญญาณบีบอัด และการซ่อนสัญญาณเฉลี่ย

ตารางที่ 4. ตารางเงื่อนไขการทดสอบหาความเป็นไปได้ในการใช้คอมพิวเตอร์เชิงตัวเลขในการ (Prove of concept) ของสัญญาณแต่ละแบบ

Experiment Table	Experiment System	Origin of signal x_t, t_x	Sensor measurement	Origin of Undersampling Signal y_t, t_y	Signal	Algorithm
Proof of Concept	Simulation	Generate from normal (Periodic) time vector	Computer	Generate from random(non-Periodic) time vector $f_{erate} = 4096 Hz$	Simulated Signal	CVX
		Shaker Signal Generator normal Measurement	PCB Sampling Rate 3200Hz (Ref.A)	Random Permutation $f_{erate} = 1600 Hz$ $f_{erate} = 640 Hz$	Shaker Signal 300 – 1800 Hz, Both single and modulation frequency	
	Compressor01 normal Measurement	PCB Sampling Rate 20000Hz	Normal Operating Signal			
	Compressor02 normal Measurement	Downsampling to 3200Hz (Ref.B)				

ตารางที่ 4.5 ตารางเงื่อนไขการทดสอบหาความเป็นไปได้ในการใช้คอมพิวเตอร์เชิงในทางปฏิบัติ (Prove of work) ของสัญญาณแต่ละแบบ

Experiment Table	Experiment System	Origin of signal x_t, t_x	Sensor measurement	Origin of Undersampling Signal y_{tr}, t_y	Signal	Algorithm
Proof of Work	Non-Uniform sampling	Shaker Signal Generator normal Measurement PCB 3200Hz	ADXL345 with multiple Equivalent Sampling Rate and frequency range $f_{erate} = 1600 \text{ Hz}$ with $f_{range} = 540.40 \text{ Hz}$ $f_{erate} = 1600 \text{ Hz}$ with $f_{range} = 1073.68 \text{ Hz}$ $f_{erate} = 1000 \text{ Hz}$ with $f_{range} = 241.71 \text{ Hz}$ $f_{erate} = 1000 \text{ Hz}$ with $f_{range} = 560.44 \text{ Hz}$ $f_{erate} = 800 \text{ Hz}$ with $f_{range} = 162.55 \text{ Hz}$ $f_{erate} = 800 \text{ Hz}$ with $f_{range} = 403.96 \text{ Hz}$ $f_{erate} = 500 \text{ Hz}$ with $f_{range} = 194.66 \text{ Hz}$ $f_{erate} = 500 \text{ Hz}$ with $f_{range} = 359.15 \text{ Hz}$	Direct Undersampling measurement	Shaker Signal 300 – 1800 Hz, Both single and modulation frequency	CVX
		Compressor01 normal Measurement Using PCB 20000Hz			Normal Operating Signal	CVX COSAMP MEANSTACK
		Compressor02 normal Measurement Using PCB 20000Hz				

บทที่ 5.

ผลการประยุกต์ใช้คอมพิวเตอร์เซินซึ่งกับการวัดการสั้นสะเทือน

ในบทที่ 5 นี้จะกล่าวถึงผลการประยุกต์ใช้คอมพิวเตอร์เซินซึ่งกับการวัดการสั้นสะเทือน โดยผลการศึกษาจะถูกแบ่งออกเป็น 2 ส่วนหลักคือ การทดสอบหลักการ และการทดสอบการนำไปใช้จริง

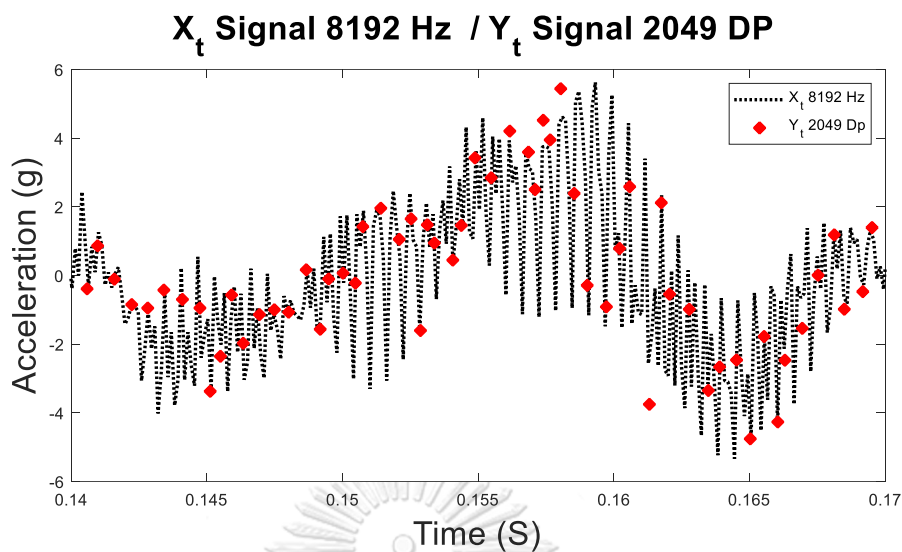
การทดสอบหลักการ จะประกอบด้วยผลการทดสอบ ทั้งหมด 3 แบบคือ 1 ผลทดสอบจากการจำลองสัญญาณในคอมพิวเตอร์ 2 ผลทดสอบจากการวัดเครื่องสร้างการสั้นสะเทือน 3 ผลทดสอบจากการวัดเครื่องจักรจริง โดย สัญญาณสุ่มในการทดสอบหลักการ จะสร้างจากอัลกอริทึมการสุ่มหยิบจุดข้อมูลทั้งหมด

การทดสอบการนำไปใช้จริง จะประกอบไปด้วยผลการทดสอบที่ได้จากการวัดเครื่องสร้างการสั้นสะเทือน และการวัดเครื่องจักรจริง โดยสัญญาณสุ่มในการทดสอบนี้จะได้จากการวัดจริงแบบสุ่มคาบโดยตรง

5.1 ผลการทดสอบหลักการ (Proof of concept)

5.1.1 การจำลองสัญญาณการวัดบนคอมพิวเตอร์

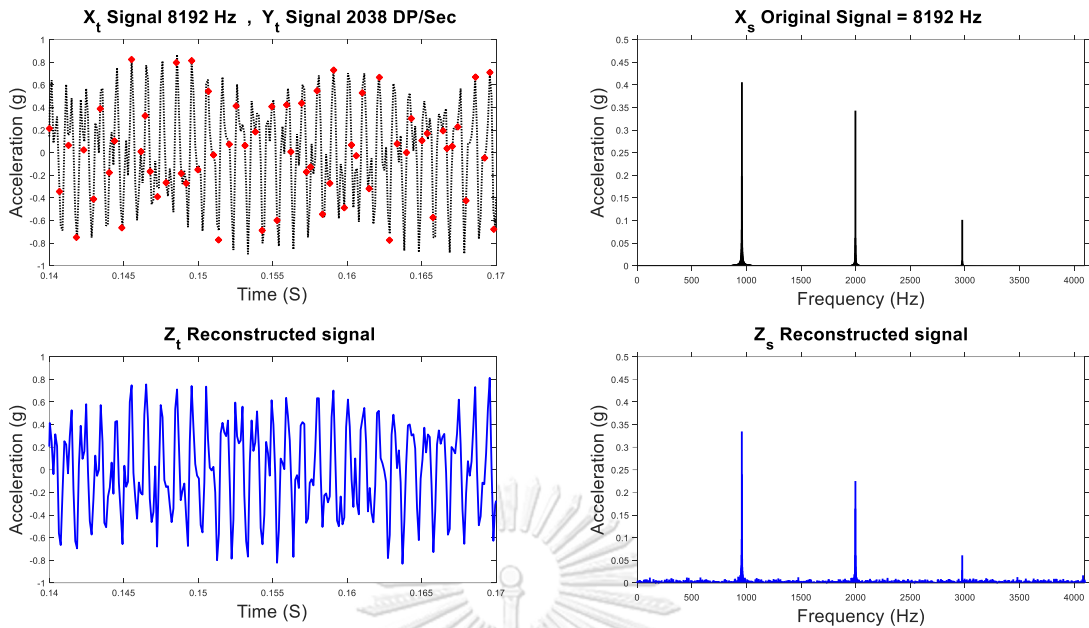
ในการศึกษานี้กำหนดให้ความถี่สูงสุดที่ต้องการวัดมีค่า 4096 เฮิรตซ์ ดังนั้นสัญญาณ x_t จึงสร้างด้วยความถี่สุ่มเท่ากับ 8192 เฮิรตซ์ หรือ 2^{16} เพื่อให้สอดคล้องกับจำนวนของเส้นสเปกตรัม ในเครื่องมือวัดการสั้นสะเทือนทั่วไป ส่วนการสร้างสัญญาณ y_t นั้น กำหนดให้ Δt_n มีค่าน้อยที่สุดสอดคล้องกับความถี่สุ่ม 4096 เฮิรตซ์ เนื่องจาก Δt_n มีค่าไม่คงที่ ความถี่สุ่มในกรณีนี้จึงเปลี่ยนแปลงตั้งแต่ประมาณ 1365 เฮิรตซ์ จนถึง 4096 เฮิรตซ์ ซึ่งคำนวณได้จาก สมการ 3.1 เนื่องจากความถี่สุ่มสูงสุดในกรณีนี้น้อยกว่า 2 เท่าของความถี่ที่ต้องการวัด ซึ่งหากไม่ใช้การประมวลผลสัญญาณแล้ว ข้อมูลในโดเมนความถี่ของ y_t ที่อาจทราบได้จะมีค่าเพียงไม่เกิน 2048 เฮิรตซ์ และเมื่อพิจารณาจำนวนจุดข้อมูลจริงของ y_t เมื่อวัดในช่วงเวลาเท่ากับ x_t แล้วจะพบว่า จำนวนจุดของข้อมูล y_t จะน้อยกว่า x_t อยู่ประมาณ 4 เท่า รูปที่ 5.1 แสดงการเปรียบเทียบสัญญาณ x_t ซึ่งมีความถี่สุ่ม 8192 เฮิรตซ์ แสดงด้วยเส้นสีดำ และ y_t ซึ่งแสดงด้วยจุดสีแดง จากรูปเห็นได้ว่าระยะห่างระหว่างจุดของสัญญาณ y_t ไม่คงที่ และจำนวนข้อมูล y_t มีเพียง 2063 จุด (ในระยะเวลา 1 วินาที) เท่านั้น ซึ่งมีจำนวนน้อยกว่าจำนวนข้อมูลของ x_t มาก



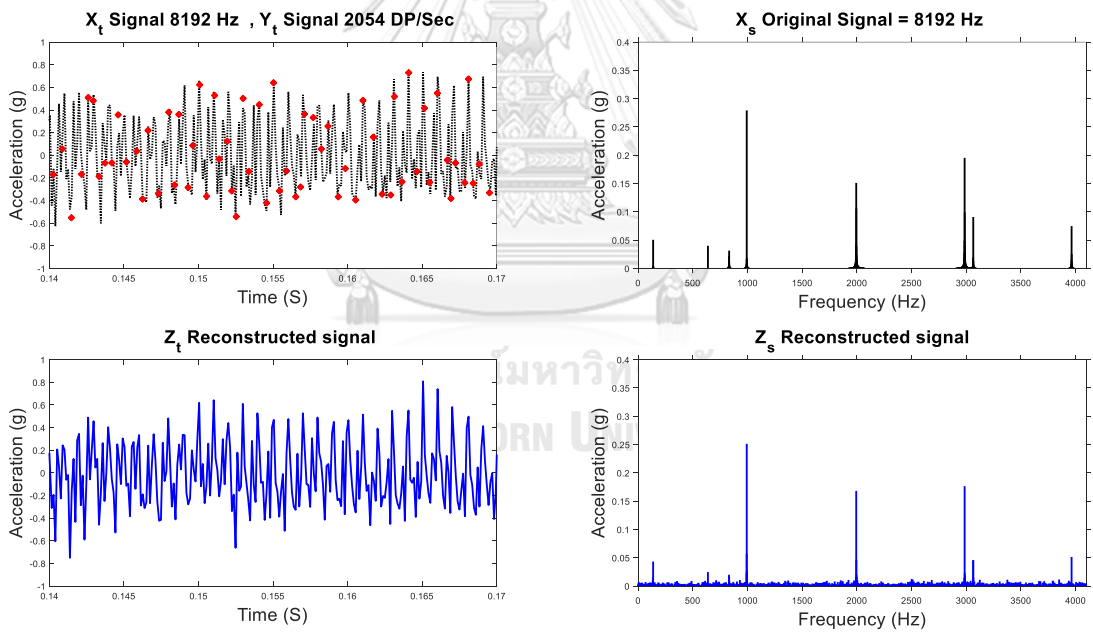
รูปที่ 5.1 ตัวอย่างสัญญาณ y_t และสัญญาณ x_t

ผลการสร้างสัญญาณคืนกลับด้วยคอมพิวเตอร์

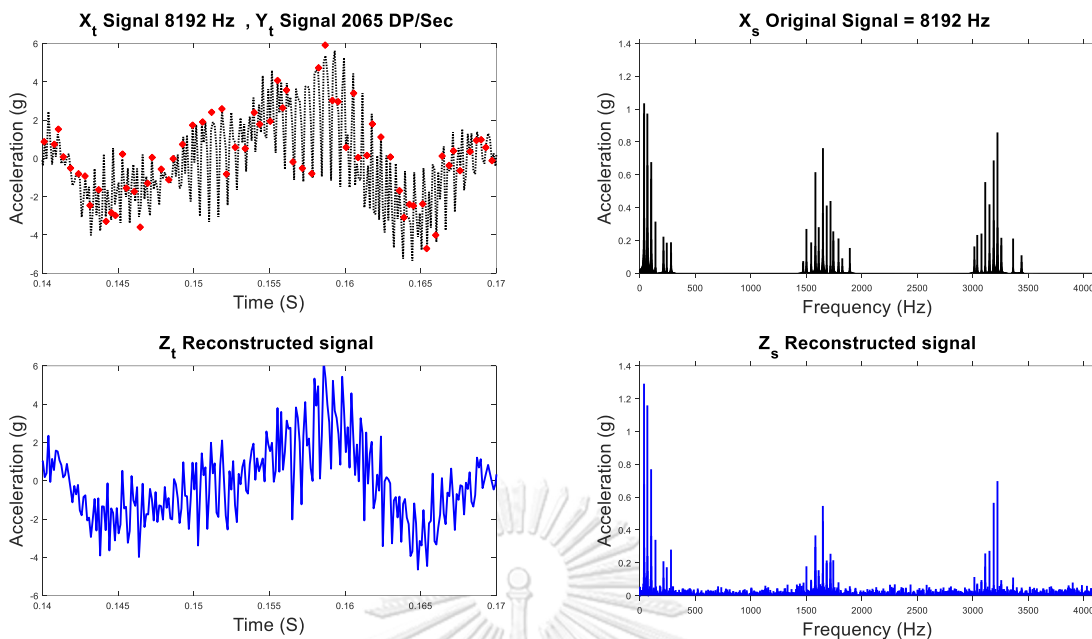
รูปที่ 5.2 - 5.4 แสดงตัวอย่างผลการสร้างสัญญาณคืนกลับ Z_S ที่พิสัยความถี่ 4096 เฮิรตซ์ แต่ละรูปแบ่งออกเป็น 4 รูปย่อย รูปด้านซ้ายบนจะเป็นสัญญาณต้นแบบในโดเมนเวลา x_t (เส้นประสีดำ) และสัญญาณที่วัดแบบสุ่มคาบ y_t (จุดสีแดง) รูปด้านขวาบนเป็นสัญญาณต้นแบบในโดเมนความถี่ X_S ส่วนรูปด้านล่างซ้ายและขวาแสดงสัญญาณคืนกลับในโดเมนเวลา Z_t และในโดเมนความถี่ Z_S ตามลำดับ ส่วนค่าวิเคราะห์ในกรณีต่างๆ แสดงในตารางที่ 5.1



รูปที่ 5.2 ผลการจำลองของการสร้างสัญญาณคั่นกลับของสัญญาณมอเตอร์



รูปที่ 5.3 ผลการจำลองของการสร้างสัญญาณคั่นกลับของสัญญาณบีม



รูปที่ 5.4 ผลการจำลองของการสร้างสัญญาณคืนกลับของสัญญาณชุดเกียร์

จากการใช้คอมพิวเตอร์ซึ่งร่วมกับการเก็บข้อมูลแบบสุ่มคาบจะเห็นว่าสำหรับกลุ่มความถี่หลักที่มีแอมพลิจูดสูง คอมพิวเตอร์ซึ่งสามารถสร้างสัญญาณที่มีความใกล้เคียงกับสัญญาณเดิมได้ดี ตลอดช่วงพิสัยความถี่ แม้ว่าความถี่สูงสุดจะน้อยกว่าเกณฑ์ของไนควิสต์ 2 เท่า และจำนวนจุดข้อมูลที่ใช้ประมวลผลจะน้อยกว่าจำนวนจุดข้อมูลที่วัดตามเกณฑ์ของไนควิสต์ประมาณ 4 เท่า เมื่อพิจารณาค่าวิเคราะห์ต่างๆ ดังตารางที่ 5.1 จะพบว่า เปอร์เซ็นต์ของความถี่ตรงกันเฉลี่ยของแต่ละสัญญาณมากกว่า 85% และเปอร์เซ็นต์ของความถี่ที่ไม่ตรงกันเฉลี่ยน้อยกว่า 3% สำหรับเปอร์เซ็นต์ของความถี่ที่ไม่สามารถสร้างกลับได้ในกรณี M/C #2 มีค่ามากในบางกรณีนั้นเนื่องจากจำนวนความถี่ที่มีแอมพลิจูดสูงทั้งหมดของสัญญาณมีน้อย เมื่อมีความผิดพลาดในการสร้างความถี่เพียง 1 หรือ 2 ความถี่ก็จะส่งผลต่อค่าเปอร์เซ็นต์มาก

ตารางที่ 5.1 ผลการทดสอบการจำลองสัญญาณ

Machine	RSS	% Match Peak	% fault Peak	% Unrecovered Peak	Cross correlation	Sparsity
M/C #1 (Motor)	0.47	100.00%	0.00%	0.00%	0.84	0.02
	0.47	100.00%	0.00%	0.00%	0.83	0.02
	0.48	100.00%	0.00%	0.00%	0.84	0.02
	0.43	100.00%	0.00%	0.00%	0.85	0.02
	0.46	100.00%	0.00%	0.00%	0.85	0.02
Avg. M/C #1	0.462	100.00%	0.00%	0.00%	0.84	0.02
M/C #2 (Pump)	0.44	75.00%	0.00%	25.00%	0.76	0.03
	0.41	87.50%	0.00%	12.50%	0.77	0.03
	0.39	87.50%	0.00%	12.50%	0.78	0.03
	0.39	100.00%	0.00%	0.00%	0.78	0.03
	0.41	87.50%	0.00%	12.50%	0.77	0.03
Avg. M/C #2	0.410	87.50%	0.00%	12.5%	0.77	0.03
M/C #3 (Gearbox)	0.47	85.71%	0.00%	14.29%	0.74	0.10
	0.49	85.71%	4.00%	14.29%	0.74	0.10
	0.47	82.14%	0.00%	17.86%	0.74	0.10
	0.46	85.71%	4.00%	14.29%	0.74	0.10
	0.47	92.86%	3.70%	7.14%	0.74	0.10
Avg. M/C #3	0.473	86.43%	2.34%	13.57%	0.74	0.10

จากในตารางที่ 5.1 จะเห็นว่าสำหรับ สัญญาณของ มอเตอร์ ที่มีความสปีร์สูงคอมเพรสชันซึ่งจะสามารถสร้างสัญญาณคืนกลับในโดเมนความถี่ได้อย่างถูกต้องโดยสามารถดูได้จาก ค่าสหสัมพันธ์ไขว้ และการเปรียบเทียบความถี่แบบต่างๆ แต่เมื่อสัญญาณต้นแบบมีจำนวนความถี่เพิ่มมากขึ้น หรือมีความสปีร์ลดลง คุณภาพของสัญญาณคืนกลับจะลดลง ซึ่งเห็นได้จากค่าเปอร์เซ็นต์ความถี่ที่ไม่ตรงกัน หรือความถี่ที่ไม่สามารถสร้างกลับมาได้เพิ่มขึ้น ค่าสหสัมพันธ์ไขว้ที่ลดลง โดยเฉพาะในกรณีของ M/C #3 นอกจากนี้ยังพบว่าความถี่ที่ไม่ตรงกันและไม่สามารถสร้างกลับมาได้มักเป็นตำแหน่งความถี่ที่มีแอมพลิจูดต่ำซึ่งใกล้เคียงกับแถบสัญญาณรบกวน สำหรับพารามิเตอร์ RSS จะเห็นว่าค่าใกล้เคียงกันทั้งหมดไม่ว่าจะเป็น สัญญาณจะมีแบบใดซึ่งพอจะสรุปได้ว่า พารามิเตอร์นี้อาจจะไม่เหมาะกับการนำมาใช้ประเมินความถูกต้องของ สัญญาณคืนกลับ

สรุปผลการจำลองการสร้างสัญญาณคืนกลับด้วยคอมเพรสชันเชิงในคอมพิวเตอร์

จากผลการจำลองในคอมพิวเตอร์จะเห็นว่าในกรณีที่ไม่ได้คำนึงถึงข้อจำกัดของระบบการวัดในการนำไปใช้งานจริงเช่น การคลาดเคลื่อนของแอมพลิจูด ความไม่เสถียรของอัตราการสุ่มจุดข้อมูล ความคลาดเคลื่อนจากระยะเวลาในการอ่านและส่งข้อมูล คอมเพรสชันเชิงสามารถที่จะถูกนำไปใช้และสามารถสร้างสัญญาณคืนกลับได้อย่างถูกต้อง ถึงแม้ว่าสัญญาณที่พิจารณาที่มีความสปีร์ต่ำ

5.1.2 การสร้างสัญญาณกลับคืนจากสัญญาณจริงที่ถูกวัดแบบปกติ

5.1.2.1 ผลการทดสอบจากการวัดสัญญาณจากเครื่องสร้างการสั่นสะเทือน

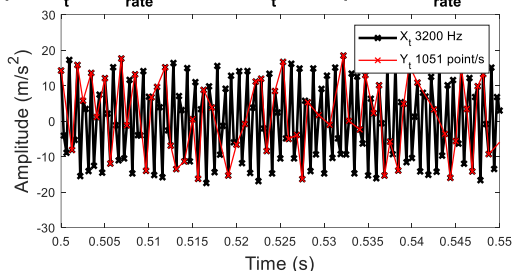
วิธีการทดสอบจะทำโดยสร้างสัญญาณการสั่นสะเทือนทั้งหมด 14 แบบตามที่กล่าวไว้แล้วในหัวข้อ 4.4 และควบคุมให้เครื่องสร้างการสั่นสะเทือนสั่นตามความถี่ที่กำหนด และวัดสัญญาณด้วย เซ็นเซอร์ PCB แบบปกติโดยใช้ความถี่ในการสุ่มจุดข้อมูล 3200 เฮิรตซ์ซึ่งมากกว่าอัตราไนควิสต์ เพื่อให้ได้สัญญาณต้นแบบ X_t หลังจากนั้นจึงทำการสุ่มหยิบข้อมูลตามเงื่อนไขที่ได้กำหนดไว้ เพื่อให้ได้สัญญาณ Y_t หลังจากนั้นจึงนำอัลกอริทึมคอมเพรสชันซึ่งแบบการหาค่าเหมาะสมที่สุดแบบคอนเวกซ์มาใช้สร้างสัญญาณกลับคืน Z_S ที่มีความถี่ในการสุ่มจุดข้อมูล 3200 เฮิรตซ์ตามเดิม

ผลการทดสอบในกรณีใช้ความถี่สุ่มสัญญาณเทียบเท่า 1600 เฮิรตซ์ ที่แสดงในตารางที่ 5.2 จะเห็นว่าการใช้ความถี่สุ่มเทียบเท่า 1600 เฮิรตซ์สามารถสร้างสัญญาณคืนกลับมาได้ทั้งหมดยกเว้น กรณีความถี่ 1800 เฮิรตซ์ เนื่องจากสัญญาณจากเซ็นเซอร์ PCB นั้นเดิมมีความถี่สุ่ม 3200 เฮิรตซ์ ซึ่งจะทำการกรองความถี่ต่ำผ่านภายใน เครื่องวิเคราะห์สัญญาณ ลดทอนแอมพลิจูดของสัญญาณในโดเมนเวลาเมื่อมีความถี่สูงเกินกว่า 1600 เฮิรตซ์ สัญญาณที่ความถี่ 1800 เฮิรตซ์จึงไม่สามารถถูกสร้างกลับมาได้อย่างถูกต้อง

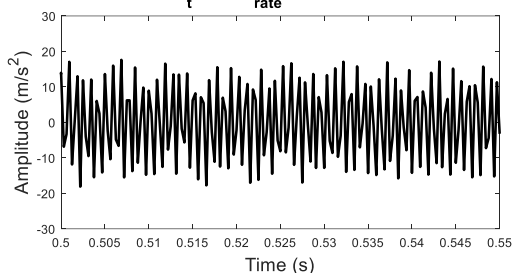
ตารางที่ 5.2 ผลการทดสอบ ที่ใช้ความถี่สุ่มเทียบเท่า 1600 เฮิรตซ์

	RSS	Y_t data point	% Match Peak	% fault Peak	% Unrecovered Peak
300Hz	0.12	1049	100%	0	0
500Hz	0.12	1049	100%	0	0
800Hz	0.42	1065	100%	0	0
1000Hz	0.44	1077	100%	0	0
1200Hz	0.45	1072	100%	0	0
1500Hz	0.44	1093	100%	0	0
1800Hz	0.51	1063	N/A	N/A	N/A
300Hz Mod 50	0.45	1063	100%	0	0
500Hz Mod 100	0.45	1069	100%	0	0
800Hz Mod 100	0.44	1073	100%	0	0
1000Hz Mod 200	0.21	1068	100%	0	0
1200Hz Mod 200	0.44	1051	100%	0	0
1500Hz Mod 200	0.44	1064	100%	0	0
1800Hz Mod 200	0.44	1070	N/A	N/A	N/A

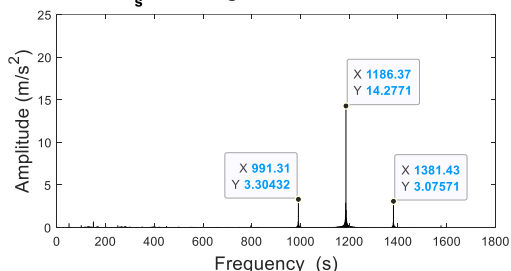
plot X_t with $f_{rate} = 3200$ Hz Y_t with equivalent $f_{rate} = 1600$ Hz



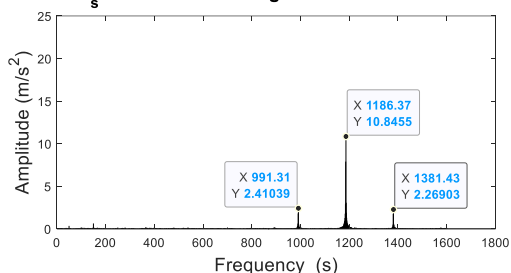
Z_t with $f_{rate} = 3200$ Hz



X_s tested signal: 1200 Hz Mod 200



Z_s Reconstruction Signal: 1200 Hz Mod 200

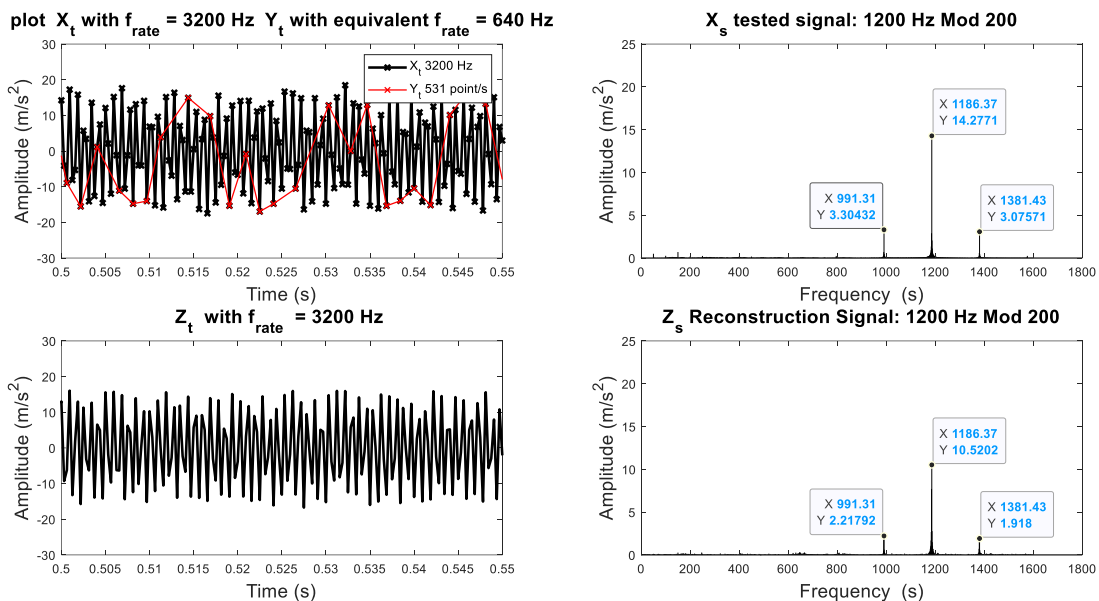


รูปที่ 5.5 ตัวอย่างผลการทดสอบการการรสร้างสัญญาณกลับคืนในกรณีเทียบเท่ากับการใช้เซ็นเซอร์ที่มีความถี่
 สุ่ม 1600 เฮิรตซ์

ตารางที่ 5.3 ผลการทดสอบ ที่ใช้ความถี่สุ่มเทียบเท่า 640 เฮิรตซ์

	RSS	Y_t data point	% Match Peak	% fault Peak	% Unrecovered Peak
300Hz	0.12	533	100%	0	0
500Hz	0.17	527	100%	0	0
800Hz	0.40	537	100%	0	0
1000Hz	0.42	537	100%	0	0
1200Hz	0.46	530	100%	0	0
1500Hz	0.45	534	100%	0	0
1800Hz	0.60	541	N/A	N/A	N/A
300Hz Mod 50	0.49	532	100%	0	0
500Hz Mod 100	0.47	538	100%	0	0
800Hz Mod 100	0.46	536	100%	0	0
1000Hz Mod 200	0.22	526	100%	0	0
1200Hz Mod 200	0.45	531	100%	0	0
1500Hz Mod 200	0.42	542	100%	0	0
1800Hz Mod 200	0.47	535	N/A	N/A	N/A

จากตารางที่ 5.3 จะเห็นว่าผลลัพธ์ที่ได้ จากความถี่สุ่มเทียบเท่า 640 เฮิรตซ์ เหมือนกับความถี่สุ่ม
 เทียบเท่า 1600 เฮิรตซ์ในตารางที่ 5.2 ที่ค่า เปอร์เซนต์ Match peak เท่ากับ 100 เปอร์เซนต์ในทุกผลการทดสอบ
 ซึ่งทำให้พอทราบได้ว่า สำหรับกรณีนี้ที่สัญญาณมีความซับซ้อนต่ำและการวัดมีความถูกต้องแม่นยำ
 คอมพิวเตอร์จึงสามารถที่จะสร้างสัญญาณคืนกลับมาได้อย่างถูกต้อง ถึงแม้ความถี่สุ่มเทียบเท่าต่ำ



รูปที่ 5.6 ตัวอย่างผลการทดสอบการการรบกวนการสั่นสะเทือนที่ความถี่ 1200 เฮิรตซ์แบบ มอดูเลชันที่มีความถี่ 640 เฮิรตซ์

รูปที่ 5.5 และ 5.6 แสดงตัวอย่างสัญญาณการสั่นสะเทือนที่ความถี่ 1200 เฮิรตซ์แบบ มอดูเลชันสัญญาณในกรณีตัวอย่างนี้เป็นสัญญาณที่มีความถี่สูงที่สุดที่ยังเห็นสัญญาณหลักและไซด์แบนด์ครบถ้วน (สำหรับความถี่ 1500 เฮิรตซ์แบบ มอดูเลชันจะเห็นไซด์แบนด์ฝั่งความถี่ต่ำเท่านั้นเนื่องจากด้านความถี่สูงอยู่ในตำแหน่งที่เกินพิสัยการวัด)

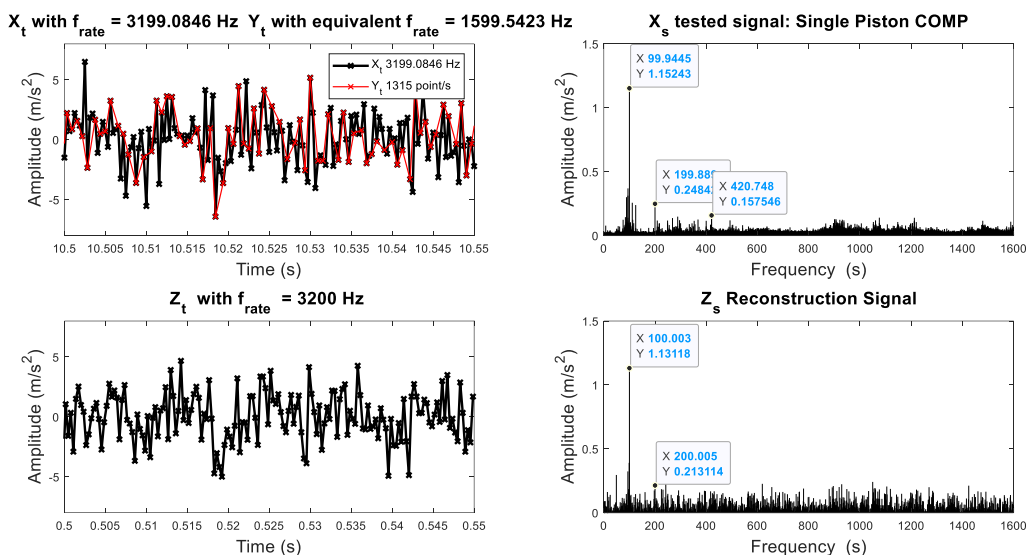
เนื่องจากสัญญาณทดสอบจริงในกรณีนี้ยังไม่ได้มีความซับซ้อนมาก หรืออาจกล่าวได้ว่าสัญญาณเหล่านี้เป็นสัญญาณที่มีความสปีร์สูง ทำให้ผลที่ได้ เทียบเท่ากับการทดสอบด้วยการจำลองในคอมพิวเตอร์ทั้งหมด ซึ่งคอมพิวเตอร์ซึ่งสามารถสร้างสัญญาณกลับคืนได้อย่างถูกต้องในทุกกรณี แต่พบว่าในกรณีที่ใช้ความถี่สุ่มที่น้อยลงก็จะส่งผลให้แอมพลิจูดของสัญญาณที่เกิดขึ้นจากคอมพิวเตอร์ซึ่งลดลงไปด้วย.

ในการทดสอบกับเครื่องจักรจริงจะเป็นการวัดการสั่นสะเทือนจาก เครื่องคอมพิวเตอร์ 1 ลูกสูบ และ 2 ลูกสูบที่ใช้งานในเวิร์คช็อป ในหัวข้อนี้จะใช้เซ็นเซอร์ PCB ที่ตั้งค่าความถี่ในการสุ่มจุดไว้ที่ 20000 เฮิรตซ์ ทำการวัดซ้ำ 3 ครั้ง หลังจากนั้นจะ Downsampling ให้เหลือ 3200 เฮิรตซ์ และสุ่มหยิบทั้งหมด 2 แบบคือแบบความถี่สุ่มเทียบเท่า 1600 เฮิรตซ์ และ 640 เฮิรตซ์

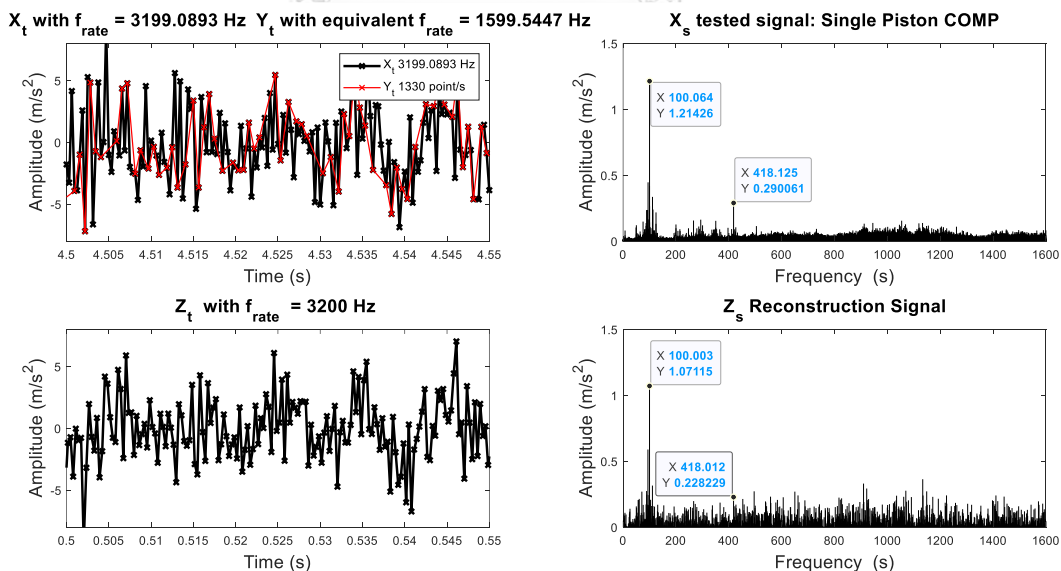
5.1.2.2 ผลการวัดสัญญาณของคอมพิวเตอร์แบบ 1 ลูกสูบ

รูปที่ 5.7 ถึง 5.12 แสดงลักษณะของสัญญาณคอมพิวเตอร์แบบ 1 ลูกสูบ จากรูปมีเพียงความถี่ 2 ความถี่ที่มีแอมพลิจูดเด่นชัด คือบริเวณ 100 เฮิรตซ์ และ 417 เฮิรตซ์ แต่สำหรับ ตำแหน่งความถี่ 417 เฮิรตซ์จะมีแอมพลิจูดไม่สูงนัก นอกเหนือจากนั้นก็จะเป็นแถบสัญญาณรบกวนซึ่งมีลักษณะที่เปลี่ยนแปลงไปทุกครั้งที่วัดจากผลการทดสอบในรูปที่ 5.7 ถึง 5.9 ซึ่งแสดงสัญญาณคืนกลับที่ถูกสร้างขึ้นโดยใช้ความถี่สุ่มเทียบเท่า 1600 เฮิรตซ์ จะเห็นว่าสัญญาณที่ตำแหน่งแอมพลิจูดหลักที่ 100 เฮิรตซ์นั้นสามารถสร้างกลับมาได้ทุกครั้ง แต่ใน

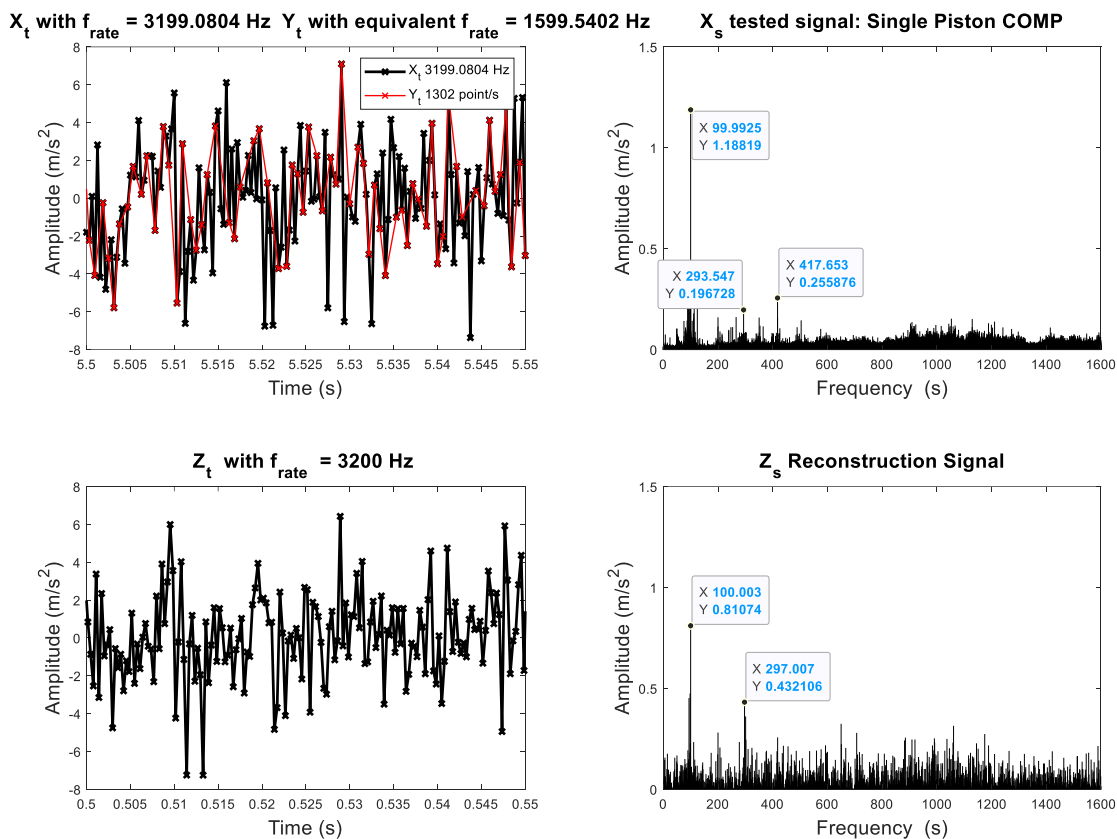
บริเวณอื่นจะมีแถบของความถี่ที่สูงกว่าสัญญาณต้นแบบ ส่งผลให้ไม่สามารถเห็นรายละเอียดของสัญญาณที่ความถี่อื่นๆได้และเมื่อทำการลดความถี่ความถี่สัมพันธ์เท่า ให้เหลือ 640 เฮิรตซ์ จากรูปที่ 5.10 ถึง 5.12 จะเห็นว่าแถบของสัญญาณรบกวนมีแอมพลิจูดเพิ่มสูงขึ้นมาก แต่ความถี่หลักนั้นมีค่าลดลง จนในบางครั้งไม่สามารถแยกแยะลักษณะและ แพทเทิร์นของสัญญาณได้



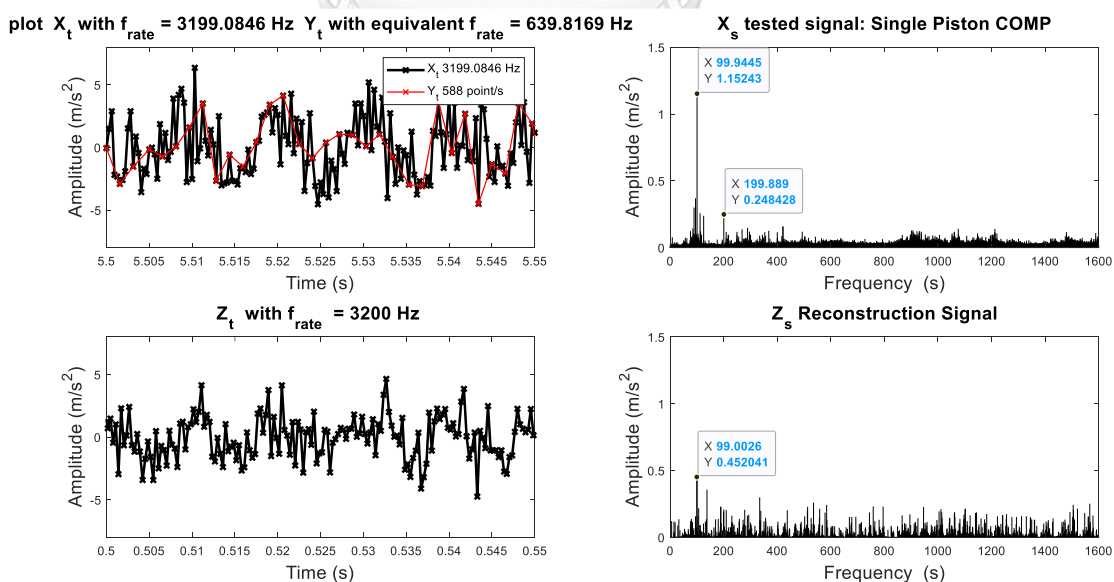
รูปที่ 5.7 การวัดสัญญาณเครื่องคอมเพรสเซอร์แบบ 1 ลูกสูบ ครั้งที่ 1 ด้วยการใช้คอมเพรสเซอร์ซึ่งร่วมกับการใช้ข้อมูลการวัดแบบสัมพันธ์เท่าการใช้เซ็นเซอร์อัตราสุ่มจุด 1600 เฮิรตซ์



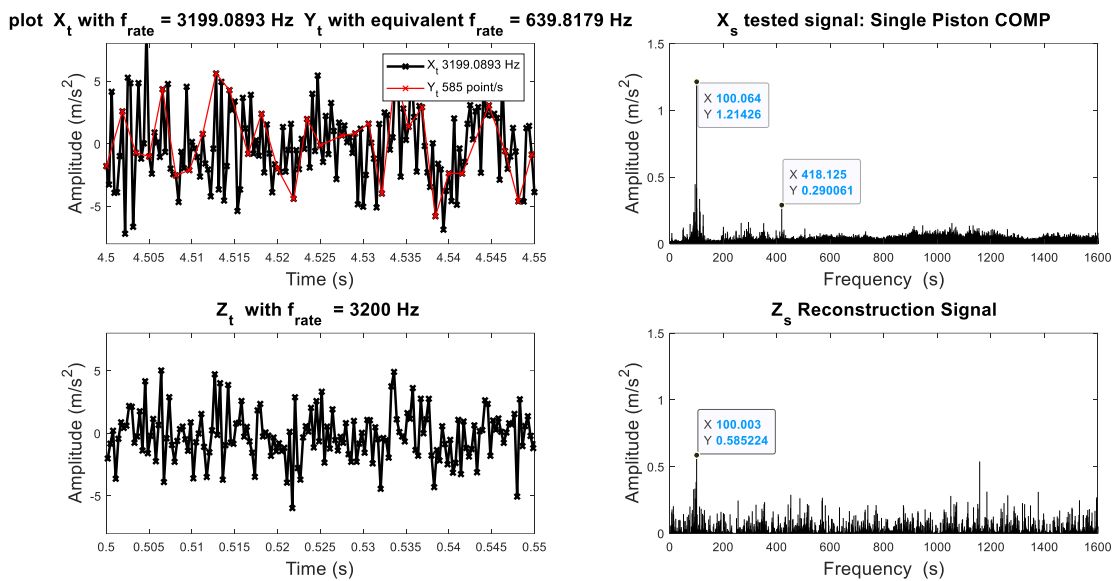
รูปที่ 5.8 การวัดสัญญาณเครื่องคอมเพรสเซอร์แบบ 1 ลูกสูบ ครั้งที่ 2 ด้วยการใช้คอมเพรสเซอร์ซึ่งร่วมกับการใช้ข้อมูลการวัดแบบสัมพันธ์เท่าการใช้เซ็นเซอร์อัตราสุ่มจุด 1600 เฮิรตซ์



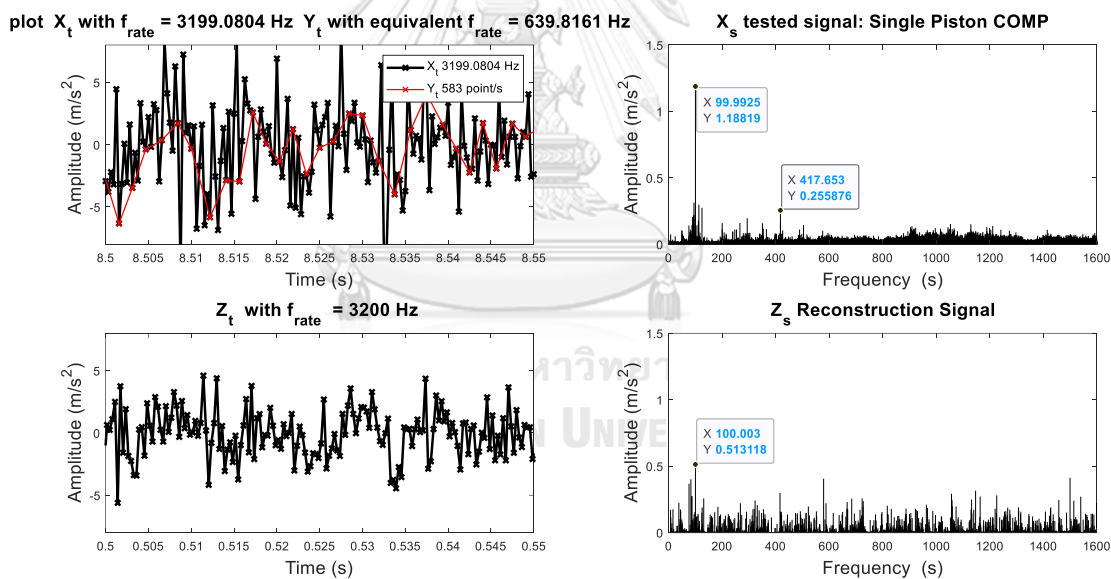
รูปที่ 5.9 การวัดสัญญาณเครื่องคอมเพรสเซอร์แบบ 1 ลูกสูบ ครั้งที่ 3 ด้วยการใช้คอมเพรสเซอร์ซึ่งร่วมกับการใช้ข้อมูลการวัดแบบสุ่มเทียบเท่าการใช้เซ็นเซอร์อัตราสุ่มจุด 1600 เฮิรตซ์



รูปที่ 5.10 การวัดสัญญาณเครื่องคอมเพรสเซอร์แบบ 1 ลูกสูบ ครั้งที่ 1 ด้วยการใช้คอมเพรสเซอร์ซึ่งร่วมกับการใช้ข้อมูลการวัดแบบสุ่มเทียบเท่าการใช้เซ็นเซอร์อัตราสุ่มจุด 640 เฮิรตซ์



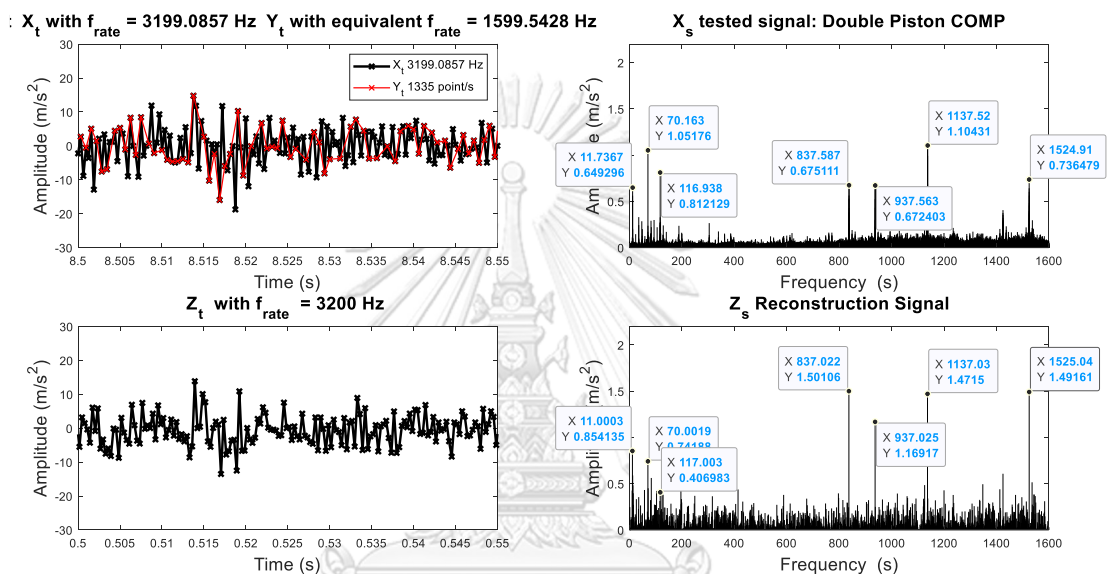
รูปที่ 5.11 การวัดสัญญาณเครื่องคอมเพรสเซอร์แบบ 1 ลูกสูบ ครั้งที่ 2 ด้วยการใช้คอมเพรสเซอร์ซึ่งร่วมกับการใช้ข้อมูลการวัดแบบสุ่มเทียบเท่าการใช้เซ็นเซอร์อัตราสุ่มจุด 640 เฮิรตซ์



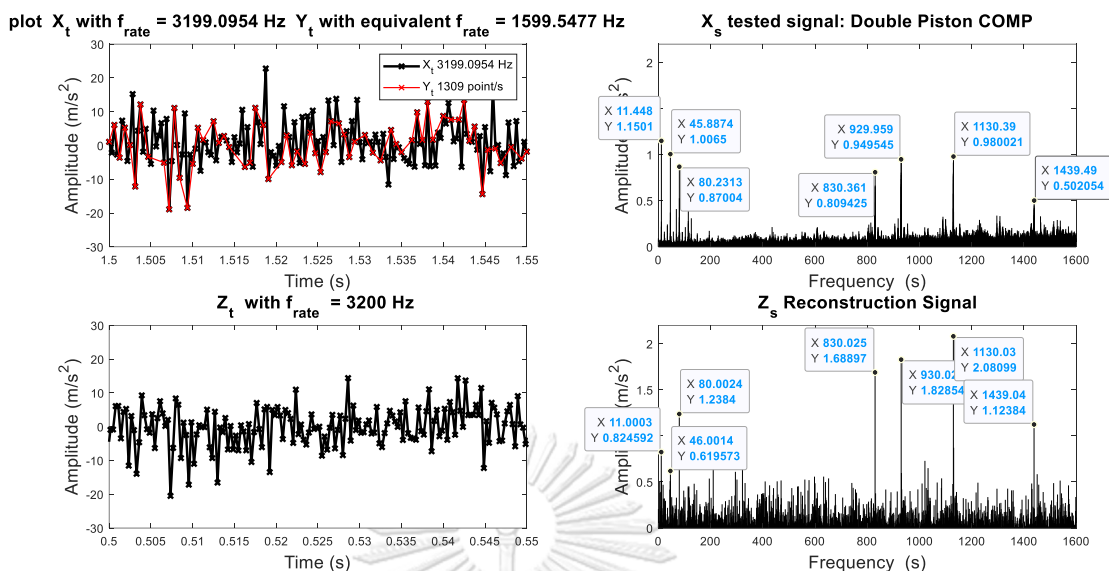
รูปที่ 5.12 การวัดสัญญาณเครื่องคอมเพรสเซอร์แบบ 1 ลูกสูบ ครั้งที่ 3 ด้วยการใช้คอมเพรสเซอร์ซึ่งร่วมกับการใช้ข้อมูลการวัดแบบสุ่มเทียบเท่าการใช้เซ็นเซอร์อัตราสุ่มจุด 640 เฮิรตซ์

5.1.2.3 ผลการทดสอบจากการวัดสัญญาณจากคอมเพรสเซอร์ 2 ลูกสูบ

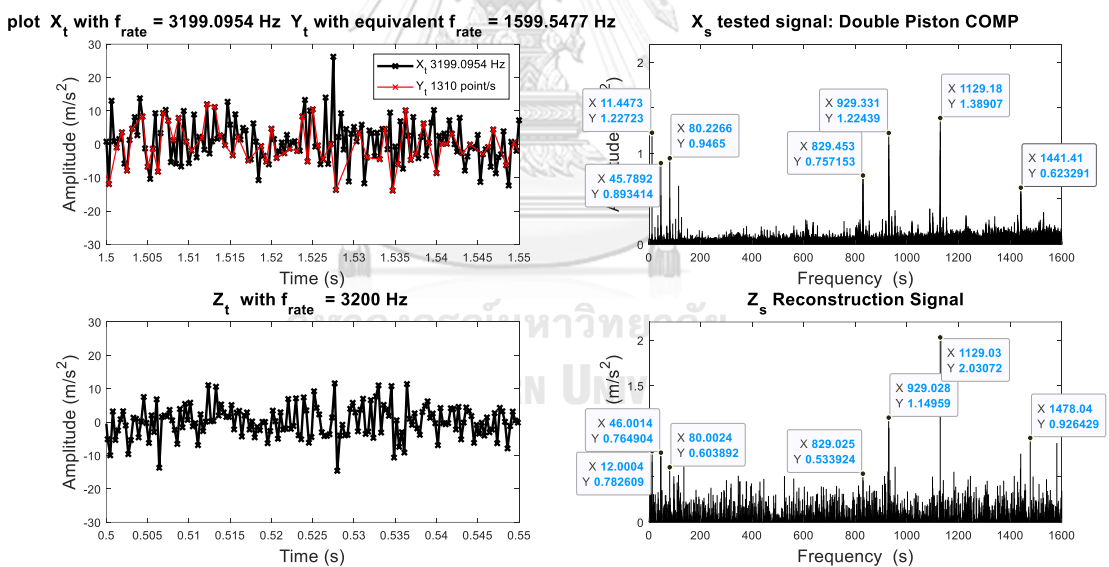
รูปที่ 5.13 ถึง 5.18 แสดงสัญญาณของคอมเพรสเซอร์แบบ 2 ลูกสูบโดยรูปที่ 5.13 - 5.15 แสดงผลในกรณีความถี่สุ่มเทียบเท่า 1600 เฮิรตซ์ ส่วนในรูปที่ 5.16 - 5.18 แสดงกรณีความถี่สุ่มเทียบเท่า 640 เฮิรตซ์ จากรูปจะมีความถี่หลักอยู่ 2 กลุ่มคือ บริเวณ 0 - 100 เฮิรตซ์และ 800 - 1200 เฮิรตซ์ จะเห็นว่าคอมเพรสเซอร์ซึ่งสร้างสัญญาณคืนกลับได้ในความถี่ที่แอมพลิจูดสูงมาก เช่น ที่ความถี่ 829,929 และ 1129 เฮิรตซ์ ทั้งการใช้ความถี่สุ่มเทียบเท่า 1600 และ 640 เฮิรตซ์ แต่ในกลุ่มความถี่ที่ 0 ถึง 100 เฮิรตซ์จะถูกแถบสัญญาณรบกวนบดบังจนไม่สามารถจะแยกแยะ ความถี่หลักกับสัญญาณรบกวนได้



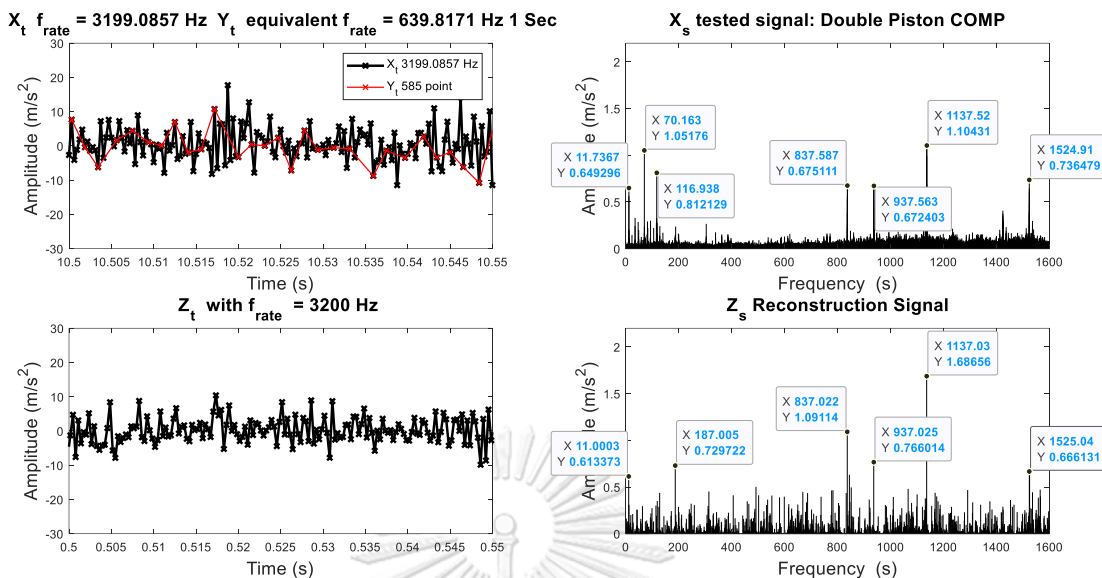
รูปที่ 5.13 การวัดสัญญาณเครื่องคอมเพรสเซอร์แบบ 2 ลูกสูบ ครั้งที่ 1 ด้วยการใช้คอมเพรสเซอร์ซึ่งร่วมกับการใช้ข้อมูลการวัดแบบสุ่มเทียบเท่าการใช้เซ็นเซอร์อัตราสุ่มจุด 1600 เฮิรตซ์



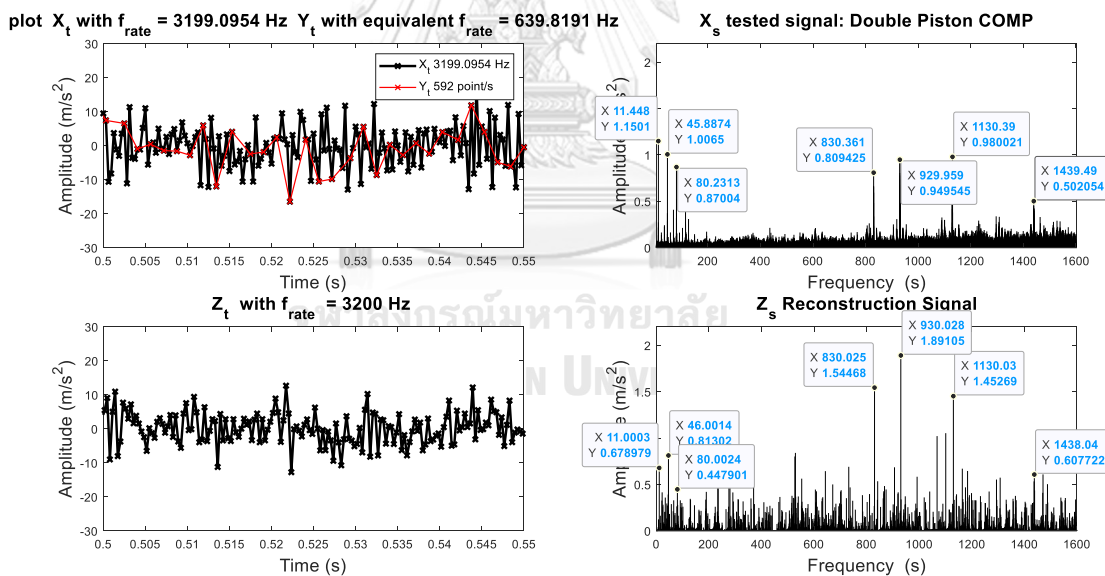
รูปที่ 5.14 การวัดสัญญาณเครื่องคอมเพรสเซอร์แบบ 2 ลูกสูบ ครั้งที่ 2 ด้วยการใช้คอมเพรสเซอร์ซึ่งร่วมกับการใช้ข้อมูลการวัดแบบสุ่มเทียบเท่าการใช้เซ็นเซอร์อัตราสุ่มจุด 1600 เฮิรตซ์



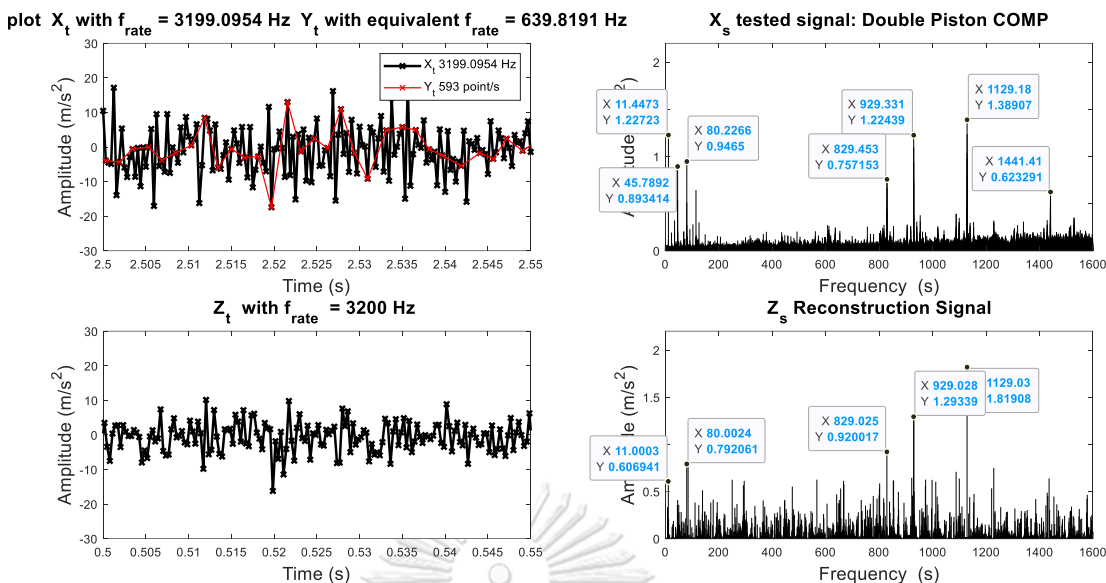
รูปที่ 5.15 การวัดสัญญาณเครื่องคอมเพรสเซอร์แบบ 2 ลูกสูบ ครั้งที่ 3 ด้วยการใช้คอมเพรสเซอร์ซึ่งร่วมกับการใช้ข้อมูลการวัดแบบสุ่มเทียบเท่าการใช้เซ็นเซอร์อัตราสุ่มจุด 1600 เฮิรตซ์



รูปที่ 5.16 การวัดสัญญาณของเครื่องคอมเพรสเซอร์แบบ 2 ลูกสูบ ครั้งที่ 1 ด้วยการใช้คอมเพรสเซอร์ซึ่งร่วมกับการใช้ข้อมูลการวัดแบบสุ่มเทียบเท่าการใช้เซ็นเซอร์อัตราสุ่มจุด 640 เฮิรตซ์



รูปที่ 5.17 การวัดสัญญาณของเครื่องคอมเพรสเซอร์แบบ 2 ลูกสูบ ครั้งที่ 2 ด้วยการใช้คอมเพรสเซอร์ซึ่งร่วมกับการใช้ข้อมูลการวัดแบบสุ่มเทียบเท่าการใช้เซ็นเซอร์อัตราสุ่มจุด 640 เฮิรตซ์



รูปที่ 5.18 การวัดสัญญาณเครื่องคอมเพรสเซอร์แบบ 2 ลูกสูบ ครั้งที่ 3 ด้วยการใช้คอมเพรสเซอร์ซึ่งร่วมกับการใช้ข้อมูลการวัดแบบสุ่มเทียบเท่าการใช้เซ็นเซอร์อัตราสุ่มจุด 640 เฮิรตซ์

ตารางที่ 5.4 ผลการทดสอบการวัดสัญญาณเครื่องคอมเพรสเซอร์ที่ความถี่สุ่มเทียบเท่าต่างๆ

Signal	f_{erate}	Cross Correlation	x_t Sparsity level
Single Piston 640Hz#1	640	0.19	0.95
Single Piston 640Hz#2	640	0.17	0.94
Single Piston 640Hz#3	640	0.18	0.94
Avg.Single Piston 640Hz	640	0.18	0.94
Single Piston 1600Hz#1	1600	0.35	0.94
Single Piston 1600Hz#2	1600	0.34	0.94
Single Piston 1600Hz#3	1600	0.30	0.94
Avg.Single Piston 1600Hz	1600	0.33	0.94
Double Piston 640Hz#1	640	0.33	0.94
Double Piston 640Hz#2	640	0.31	0.97
Double Piston 640Hz#3	640	0.30	0.98
Avg.Double Piston 640Hz	640	0.31	0.96
Double Piston 1600Hz#1	1600	0.50	0.95
Double Piston 1600Hz#2	1600	0.52	0.97
Double Piston 1600Hz#3	1600	0.50	0.98
Avg.Double Piston 1600Hz	1600	0.50	0.97

จากตารางที่ 5.4 จะเห็นว่าสัญญาณคอมเพรสเซอร์ที่ได้จากการวัดจริงนั้นมีความเป็นสปราร์ซิตีต่ำ ซึ่งและค่าสหสัมพันธ์ไขว้มีค่าน้อยมากเมื่อเทียบกับการทดสอบจำลองสัญญาณในตารางที่ 5.1 แต่อย่างไรก็ตามจากตารางนี้ก็จะเห็นแนวโน้มที่ ค่าสหสัมพันธ์ไขว้มีค่าดีขึ้นเมื่อ ใช้ความถี่สุ่มเทียบเท่าที่มากขึ้น

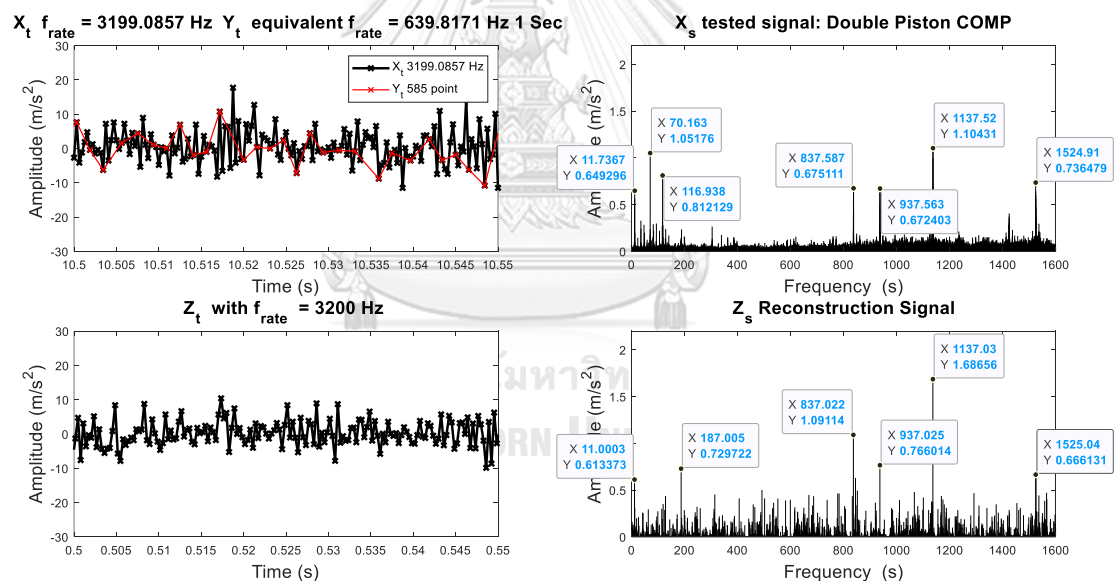
5.1.2.4 การใช้ชุดข้อมูลที่กว้างขึ้นในการทดสอบ

การใช้ชุดข้อมูลที่กว้างขึ้นจะทำการทดสอบเพื่อหาผลกระทบของพารามิเตอร์ต่างๆ ที่มีผลต่อคุณภาพสัญญาณ โดยจะทดสอบ 2 แบบด้วยกัน แบบแรกคือการใช้ชุดข้อมูลที่มากขึ้นแต่ความถี่สัมพันธ์เท่ากันที่ แบบที่สองคือการปรับความถี่สัมพันธ์เท่ากันโดยให้จำนวนจุดข้อมูลคงที่

1. การใช้ชุดข้อมูลที่มากขึ้นโดยความถี่สัมพันธ์เท่ากันที่

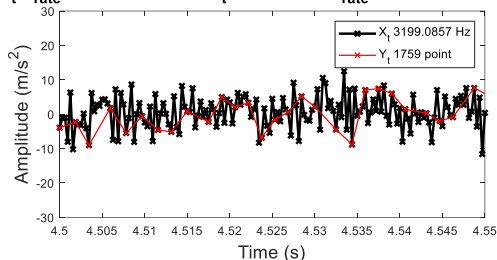
สำหรับการทดสอบนี้จะใช้สัญญาณคอมเพรสเซอร์แบบ 2 ลูกสูบ วัดครั้งที่ 1 โดยสัมพันธ์สัญญาณแบบสัมพันธ์เท่าการใช้เซ็นเซอร์อัตราสุ่มจุด 640 เฮิร์ตซ์ในรูปที่ 5.16 เป็นสัญญาณอ้างอิง แต่จะเพิ่มจำนวนจุดข้อมูล y_t, t_y จากเดิมที่มีความยาวเพียง 1 วินาที เป็น 3 วินาที และ 5 วินาทีตามลำดับ ซึ่งส่งผลให้จำนวนจุดข้อมูลที่นำมาคำนวณเพิ่มขึ้น จากเดิม 585 จุด เป็น 1759 จุดและ 2951 ตามลำดับ

จากรูปที่ 5.19 ถึง 5.22 จะเป็นรูปผลการทดสอบของสัญญาณ โดย รูปที่ 5.19 ถึง 5.21 จะเป็นภาพผลการทดสอบในกรณี 1,3 และ 5 วินาทีตามลำดับ ส่วนรูปที่ 5.22 จะเป็นรูปเปรียบเทียบของสัญญาณในโดเมนความถี่ระหว่างสัญญาณต้นแบบและกรณีที่ใช้ความยาวการวัดต่างกัน

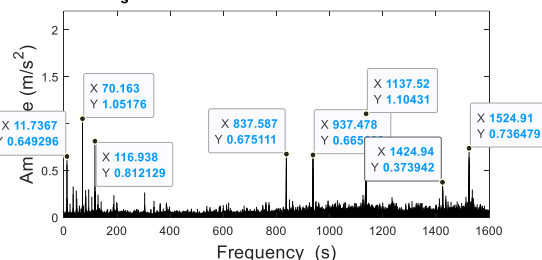


รูปที่ 5.19 การวัดสัญญาณเครื่องคอมเพรสเซอร์แบบ 2 ลูกสูบ โดยใช้ความถี่สัมพันธ์เท่า 640 เฮิร์ตซ์ และความยาวชุดข้อมูล 1 วินาที

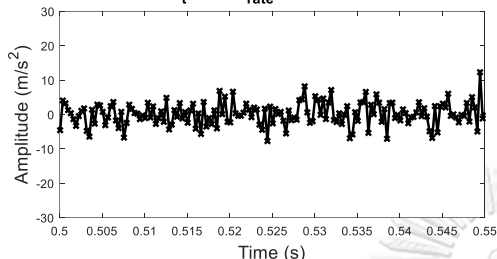
X_t $f_{rate} = 3199.0857$ Hz Y_t equivalent $f_{rate} = 639.8171$ Hz 3 Sec



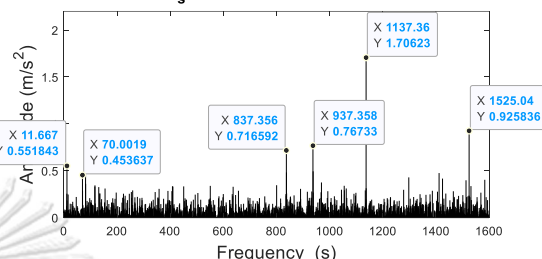
X_s tested signal: Double Piston COMP



Z_t with $f_{rate} = 3200$ Hz

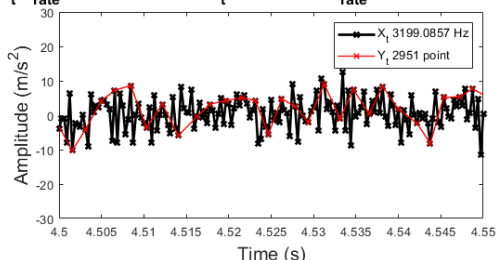


Z_s Reconstruction Signal

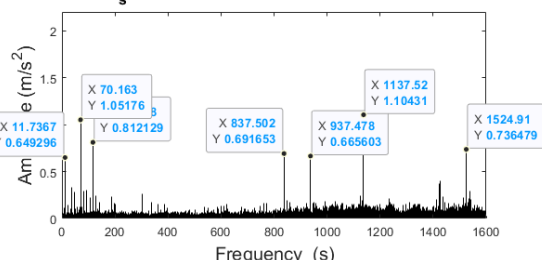


รูปที่ 5.20 การวัดสัญญาณเครื่องคอมเพรสเซอร์แบบ 2 ลูกสูบ โดยใช้ความถี่สุ่มเทียบเท่า 640 เฮิรตซ์ และความยาวชุดข้อมูล 3 วินาที

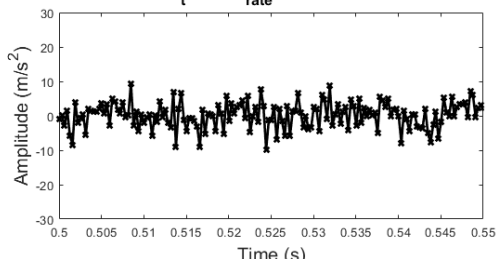
X_t $f_{rate} = 3199.0857$ Hz Y_t equivalent $f_{rate} = 639.8171$ Hz 5 Sec



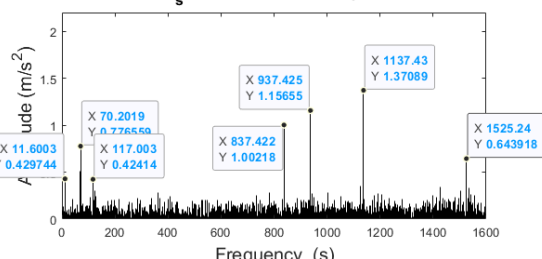
X_s tested signal: Double Piston COMP



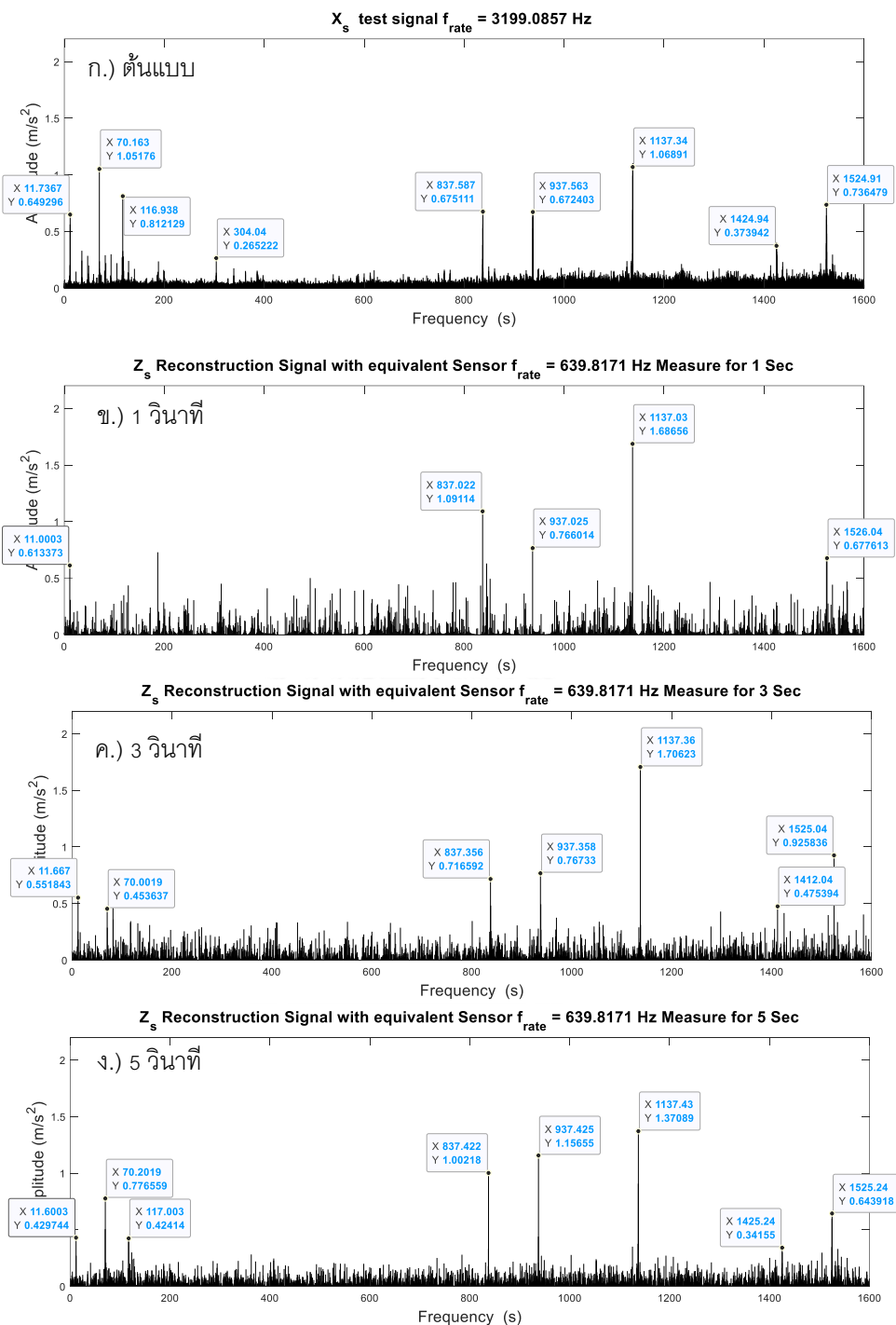
Z_t with $f_{rate} = 3200$ Hz



Z_s Reconstruction Signal



รูปที่ 5.21 การวัดสัญญาณเครื่องคอมเพรสเซอร์แบบ 2 ลูกสูบ โดยใช้ความถี่สุ่มเทียบเท่า 640 เฮิรตซ์ และความยาวชุดข้อมูล 5 วินาที



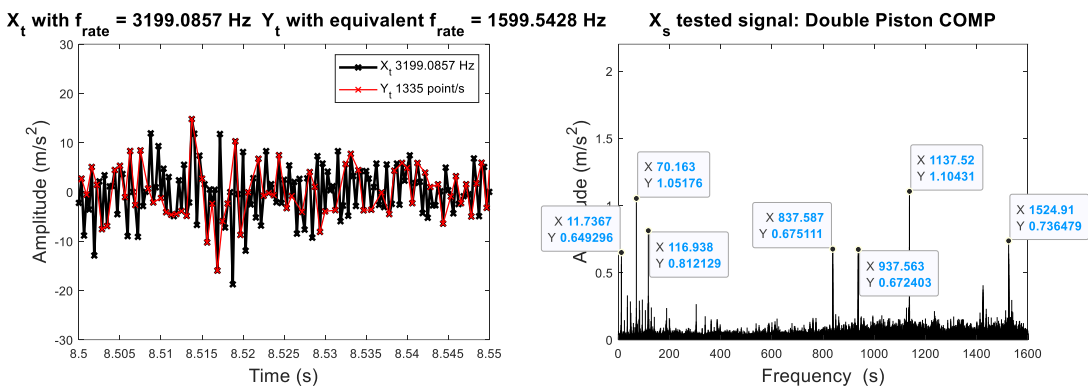
รูปที่ 5.22 ผลการสร้างสัญญาณคืนกลับของเครื่องคอมพิวเตอร์แบบ 2 ลูกสูบ ที่มีความถี่สุ่มเทียบเท่า 640 เฮิรตซ์ โดยใช้ความยาวข้อมูลไม่เท่ากัน ก.) สัญญาณต้นแบบ ข.) y_t ความยาว 1 วินาที ค.) y_t ความยาว 3 วินาที ง.) y_t ความยาว 5 วินาที

จากรูปที่ 5.22 จะเห็นว่าในกรณีที่สุ่มหยิบสัญญาณ y_t ที่มีความถี่สุ่มเทียบเท่า 640 เฮิรตซ์ การใช้ข้อมูลที่มีความยาว 1 วินาทีนั้นอาจจะไม่เพียงพอต่อการสร้างสัญญาณคืนกลับที่มีคุณภาพได้เท่ากับการใช้ข้อมูลความยาว 3 หรือ 5 วินาที และอาจสรุปได้ว่าการเพิ่มความยาวสัญญาณ y_t สามารถเพิ่มคุณภาพของสัญญาณคืนกลับได้ถึงแม้จะยังใช้ความถี่สุ่มเทียบเท่าที่มีค่าต่ำ ซึ่งในกรณีนี้ต่ำกว่าพิสัยความถี่ 1600 เฮิรตซ์ถึง 2.5 เท่า ถึงแม้การเพิ่มจำนวนจุดข้อมูลจะทำให้ได้สัญญาณที่ดีขึ้น แต่การเพิ่มความยาวของสัญญาณ y_t ส่งผลให้ขนาดของแมทริกซ์ฐานหลักเริ่มต้นมีขนาดเพิ่มขึ้นอย่างมาก จากในกรณี 1 วินาที มีขนาด $[593 \times 3200]$ 3 วินาที มีขนาด $[1759 \times 9600]$ และ 5 วินาที มีขนาด $[1951 \times 16000]$ ทำให้การคำนวณหาสัญญาณคืนกลับโดยใช้การหาค่าเหมาะสมที่สุดแบบคอนเวกซ์จะใช้เวลาเพิ่มขึ้นมาก โดย กรณีสัญญาณ y_t 1 วินาที ใช้เวลาคำนวณ 3 นาที สัญญาณ y_t 3 วินาที ใช้เวลาคำนวณ 7 นาที และสัญญาณ y_t 5 วินาที ใช้เวลาคำนวณถึง 70 นาที

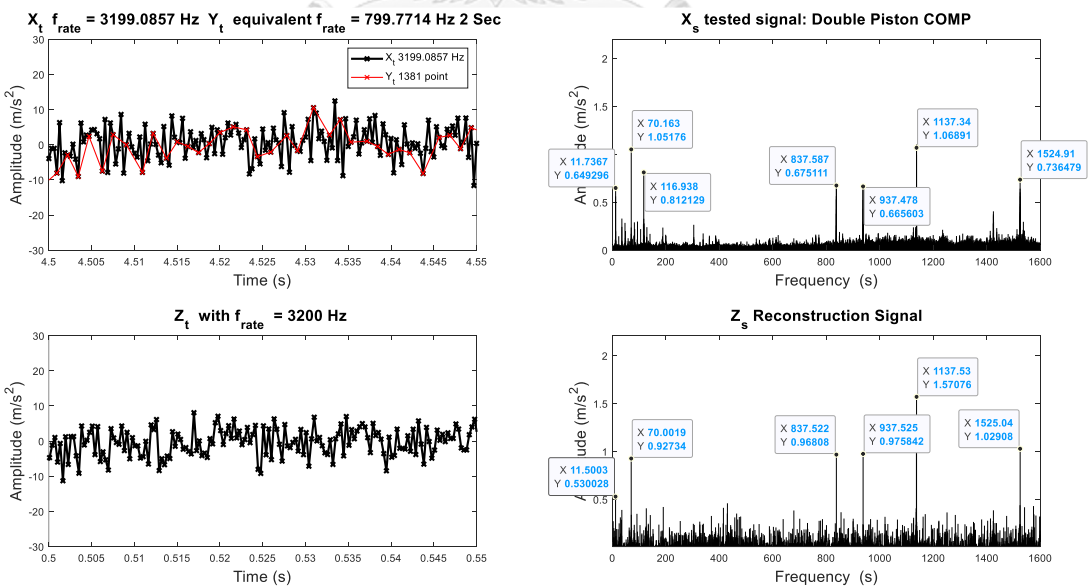
2. การปรับความถี่สุ่มเทียบเท่า โดยให้จำนวนจุดข้อมูลคงที่

การทดสอบนี้จะกำหนดให้จำนวนจุดข้อมูลที่ใช้ในคอมพิวเตอร์เซมซิงมีค่าคงที่ ประมาณ 1300 จุด และปรับความถี่สุ่มเทียบเท่าให้เท่ากับ 800 เฮิรตซ์ และ 1600 เฮิรตซ์ ซึ่งกรณีความถี่สุ่มเทียบเท่า 1600 เฮิรตซ์ จะใช้ความยาวข้อมูล 1 วินาที ส่วนกรณีความถี่สุ่มเทียบเท่า 800 เฮิรตซ์ จะใช้ความยาวข้อมูล 2 วินาทีเพื่อจะได้จำนวนจุดข้อมูลที่ใกล้เคียงกัน

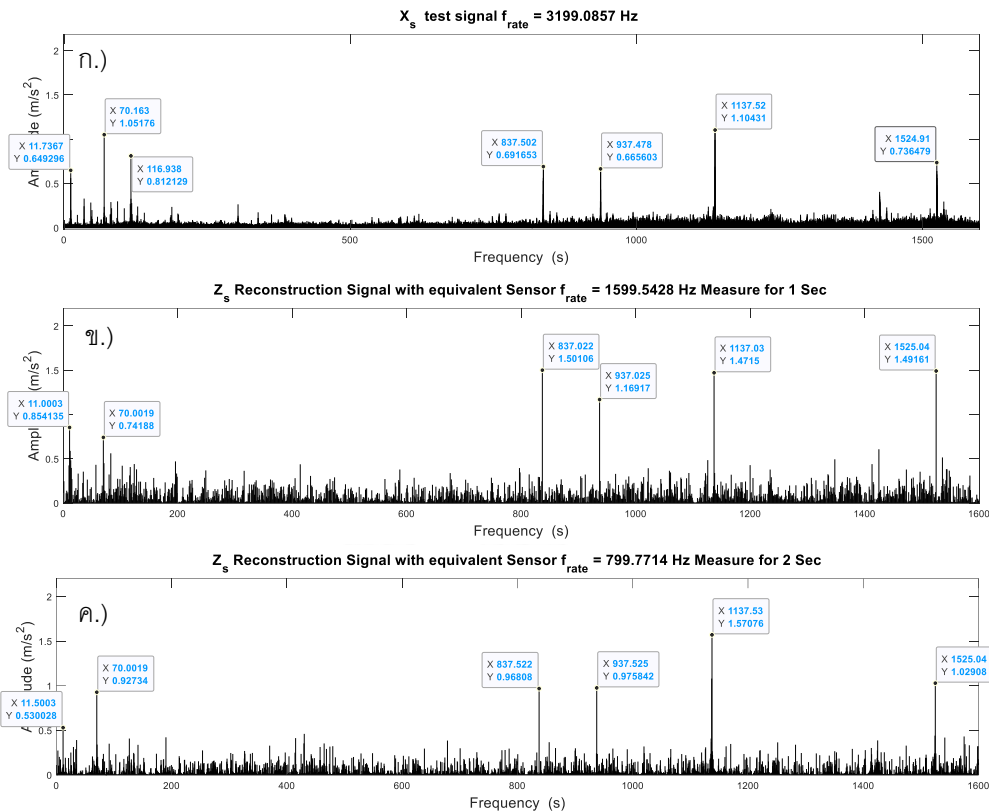
จากผลการทดสอบจะถูกแสดงในรูปที่ 5.23 และ 5.24 จะเห็นว่าการใช้ความยาวข้อมูลทั้งสองแบบให้ผลของสัญญาณคืนกลับมีแถบสัญญาณรบกวนที่ไม่สูงนักสามารถแยกแยะความถี่หลักได้อย่างชัดเจนและใกล้เคียงกับสัญญาณต้นแบบทั้งในโดเมนเวลาและโดเมนความถี่



รูปที่ 5.23 การวัดสัญญาณเครื่องคอมพิวเตอร์แบบ 2 ลูกสูบ โดยใช้อัตราสุ่มเทียบเท่า 1600 เฮิรตซ์ ความยาวชุดข้อมูล 1 วินาที (จำนวนจุดข้อมูล 1335 จุด)



รูปที่ 5.24 การวัดสัญญาณเครื่องคอมพิวเตอร์แบบ 2 ลูกสูบ โดยใช้อัตราสุ่มเทียบเท่า 800 เฮิรตซ์ ความยาวชุดข้อมูล 2 วินาที (จำนวนจุดข้อมูล 1381 จุด)



รูปที่ 5.25 ผลการสร้างสัญญาณคืนกลับของเครื่องคอมพิวเตอร์แบบ 2 ลูกสูบ โดยใช้ความถี่สุ่มเทียบเท่าแบบต่างๆ แต่คงจำนวนจุดข้อมูลประมาณ 1300 จุด ก.) สัญญาณต้นแบบ ข.) 1600 เฮิรตซ์ ค.) 800 เฮิรตซ์

จากรูปที่ 5.25 ในกรณีที่จำนวนจุดข้อมูลใกล้เคียงกันที่ประมาณ 1300 จุด จะเห็นว่าความถี่สุ่มเทียบเท่า 1600 และ 800 เฮิรตซ์ ให้ลักษณะสัญญาณที่คล้ายคลึงกัน

จุฬาลงกรณ์มหาวิทยาลัย

CHULALONGKORN UNIVERSITY

5.1.2.5 สรุปการทดสอบหลักการ

จากผลการทดสอบทั้งหมดอาจสรุปได้ว่าในเชิงหลักการแล้วมีความเป็นไปได้ที่จะใช้คอมพิวเตอร์ซึ่งกับการวัดการสั่นสะเทือน แต่ก็มีข้อจำกัดในเรื่องของความซับซ้อนของสัญญาณ สำหรับสัญญาณที่มีความซับซ้อนน้อย คอมพิวเตอร์ซึ่งสามารถที่จะใช้สัญญาณ y_t ที่มีความถี่สุ่มเทียบเท่าน้อยมากๆ ได้ แต่หากสัญญาณมีความซับซ้อนมากขึ้นหรือจำนวนจุดของ y_t ลดลง จะส่งผลให้แถบของสัญญาณรบกวนเพิ่มสูงขึ้น และสัญญาณหลักที่ต้องการทราบมีขนาดเล็กลง สัญญาณคืนกลับที่สร้างออกมาจึงมีคุณภาพลดลงจนไม่สามารถใช้งานได้ แต่อย่างไรก็ตามหากสามารถเพิ่มจำนวนจุดของ y_t ที่ใช้ในการทำคอมพิวเตอร์ซึ่ง ไม่ว่าจะเป็นการเพิ่มความถี่สุ่มเทียบเท่า หรือเพิ่มเวลาในการวัดสัญญาณให้นานขึ้น ก็สามารถจะลดแถบสัญญาณรบกวนและเพิ่มแอมพลิจูดของสัญญาณหลักกลับมามีค่าดีขึ้นได้

5.2 การทดสอบการนำไปใช้จริง (Proof of practice)

ในการทดสอบการนำไปใช้จริงจะใช้ บอร์ดไมโครคอนโทรลเลอร์ STM32F411RE ร่วมกับเซ็นเซอร์ ADXL345 ซึ่งถ้าหากใช้วัดแบบปกติจะมีความถี่สูงสุดที่ทำได้อยู่ที่ 3200 เฮิร์ตซ์ ทำให้มีพิสัยความถี่ที่วัดได้ 1600 เฮิร์ตซ์ แต่อย่างไรก็ตามในการทดสอบนี้จะทำการสอบด้วยการวัดแบบสุ่มคาบที่ความถี่สุ่มตั้งแต่ 500 เฮิร์ตซ์ไปจนถึง 1600 เฮิร์ตซ์ เพื่อจะสร้างสัญญาณที่มีพิสัยความถี่ 1600 เฮิร์ตซ์

ในการใช้คอมพิวเตอร์สร้างสัญญาณคืนกลับ Z_t จะต้องวัดสัญญาณการสั่นสะเทือนแบบสุ่มคาบ เพื่อให้ได้สัญญาณตั้งต้น y_t ออกมาก่อน อย่างไรก็ตามลักษณะการสุ่มคาบที่ต่างกัน เช่น การกำหนดให้มีความถี่สูงสุด ความถี่ต่ำสุด หรือพิสัยของความถี่สุ่มที่ต่างกันไป ก็มีผลอย่างมากต่อการสร้างสัญญาณคืนกลับ ในทางปฏิบัติแล้วการกำหนดค่าเหล่านี้ไม่อาจตั้งค่าจากไมโครคอนโทรลเลอร์ได้โดยตรง แต่อาจปรับค่าเหล่านี้ได้โดยการ ตั้งค่าพีเอสเคล (PSC) ค่าอนโตรีโพลด์ (ARR) และจำนวนพจน์ของฟังก์ชันสุ่ม (Random()) ที่แตกต่างกัน ดังรายละเอียดที่แสดงในภาคผนวก ค. ตารางที่ 5.5 แสดงการตั้งค่าพารามิเตอร์ต่างๆ ที่ใช้ในการสุ่มวัดในงานวิจัยนี้ โดย $N_{y,1}$ ในตารางคือจำนวนจุดข้อมูลใน 1 วินาที เนื่องจากการวัดแบบสุ่มคาบมีระยะห่างของจุดเวลาไม่คงที่ ค่าความถี่สุ่มจึงมีค่าไม่คงที่ด้วย และผลต่างของความถี่สุ่มสูงสุดกับต่ำสุดเรียกว่าพิสัยการสุ่ม (*frange*)

ตารางที่ 5.5 แสดงการปรับตั้ง ระบบการวัดแบบต่างๆที่ใช้ในการทดสอบ

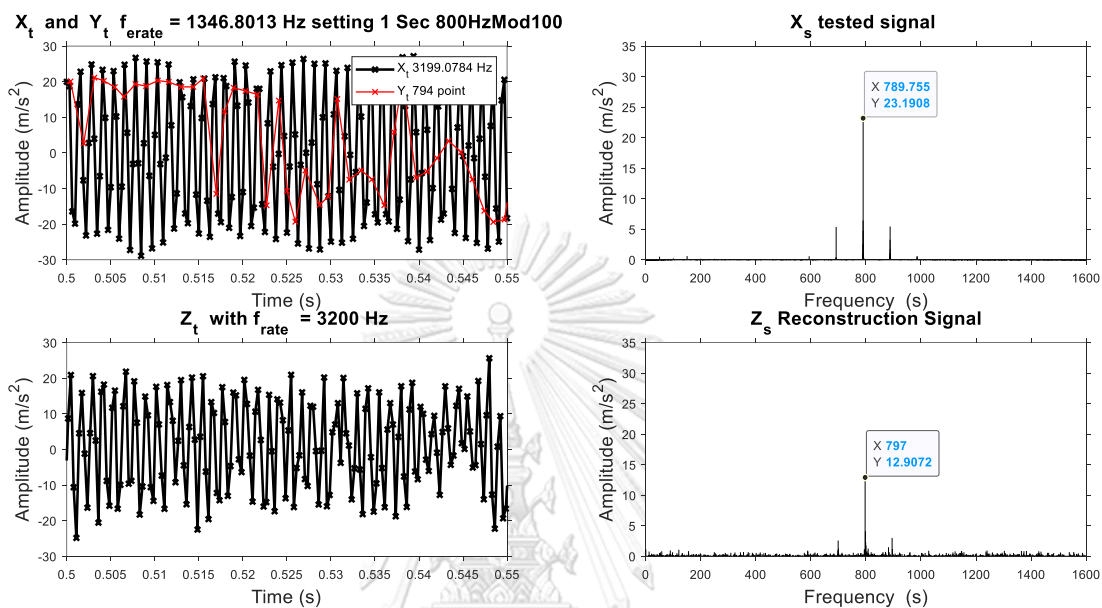
แบบที่	PSC	ARR	Random()	$N_{y,1}$			<i>frange</i>
				min	mean	max	
1	20	500	1	1,059.60	1,273.89	1,600.00	540.40
2	20	500	4	526.32	790.51	1,600.00	1,073.68
3	20	800	1	758.29	862.07	1,000.00	241.71
4	20	800	4	439.56	609.76	1,000.00	560.44
5	20	1,000	1	637.45	709.22	800.00	162.55
6	20	1,000	4	396.04	529.10	800.00	403.96
7	80	400	1	305.34	378.79	500.00	194.66
8	80	400	4	140.85	219.30	500.00	359.15

จากตารางข้างต้นหมายความว่า การวัดสัญญาณแต่ละครั้งจะต้องมีการปรับตั้งทั้งหมด 8 รูปแบบ สำหรับการทดสอบการนำไปใช้จริงจะแบ่งสัญญาณออกเป็น 2 ชนิด คือสัญญาณของเครื่องสร้างการสั่นสะเทือน 14 สัญญาณ และสัญญาณจากเครื่องจักรจริงได้แก่เครื่องคอมพิวเตอร์ 1 และ 2 ลูกสูบ 2 สัญญาณ ดังนั้นจะมีจำนวนกรณีทั้งหมดรวม 128 กรณี เนื่องจากมีผลการวัดจำนวนมาก ในหัวข้อที่ 5.2.1 และ 5.2.2 จะแสดงผลเฉพาะผลการวัดและการปรับตั้งที่มีความสำคัญบางส่วนเท่านั้น โดยผลกรณีอื่นๆ ทั้งหมดสามารถดูได้ในภาคผนวก ก.

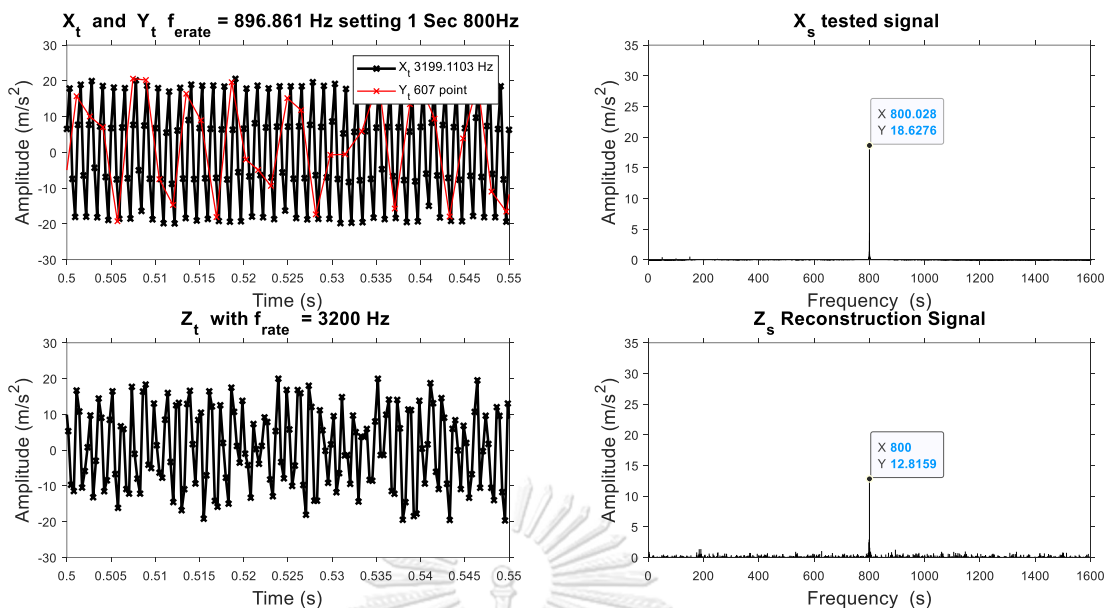
สำหรับวิธีการประเมินสัญญาณทั้งหมดในหัวข้อ นี้จะใช้ การเปรียบเทียบลักษณะของสัญญาณในโดเมนความถี่ด้วยการประเมินด้วยระดับความคล้ายคลึงกันของสัญญาณในโดเมนความถี่ดังที่ได้กล่าวไว้แล้วในหัวข้อ 4.3

5.2.1 การวัดการสั่นสะเทือนจากเครื่องสร้างสัญญาณ

ในหัวข้อนี้แสดงผลการสร้างสัญญาณคืนกลับจากสัญญาณที่สร้างจากเครื่องสร้างการสั่นสะเทือนและเครื่องกำเนิดสัญญาณด้วยการวัดแบบสุ่มคาบ ผลที่ได้ถูกแสดงในรูปที่ 5.26 - 5.29 และสรุปในตารางที่ 5.5 สำหรับผลการทดสอบทั้งหมดสามารถดูเพิ่มเติมได้ที่ ภาคผนวก ณ

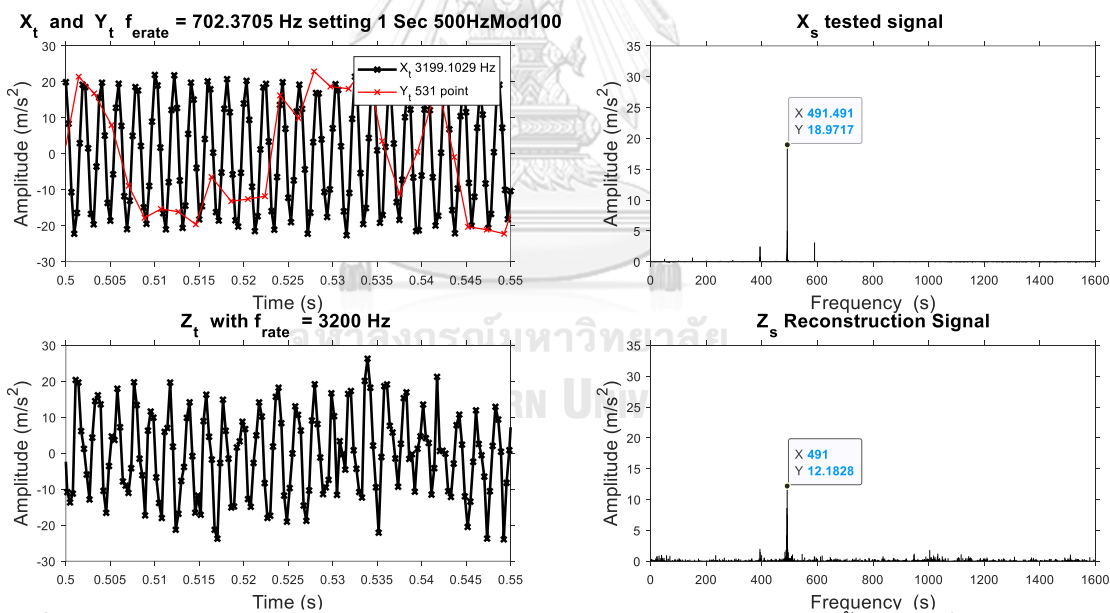


รูปที่ 5.26 ตัวอย่างผลการทดสอบการการสร้างสัญญาณกลับคืนจากการวัดด้วยการตั้งค่าแบบที่ 1 วัดสัญญาณ 800 เฮิรตซ์ แบบมอดูเลชั่น



รูปที่ 5.27 ตัวอย่างผลการทดสอบการการสร้างสัญญาณกลับคืนจากการวัดด้วยการตั้งค่าแบบที่ 4 วัดสัญญาณ

800 เฮิรตซ์



รูปที่ 5.28 ตัวอย่างผลการทดสอบการการสร้างสัญญาณกลับคืนจากการวัดด้วยการตั้งค่าแบบที่ 6 วัดสัญญาณ

500 เฮิรตซ์ แบบมอดูเลชัน

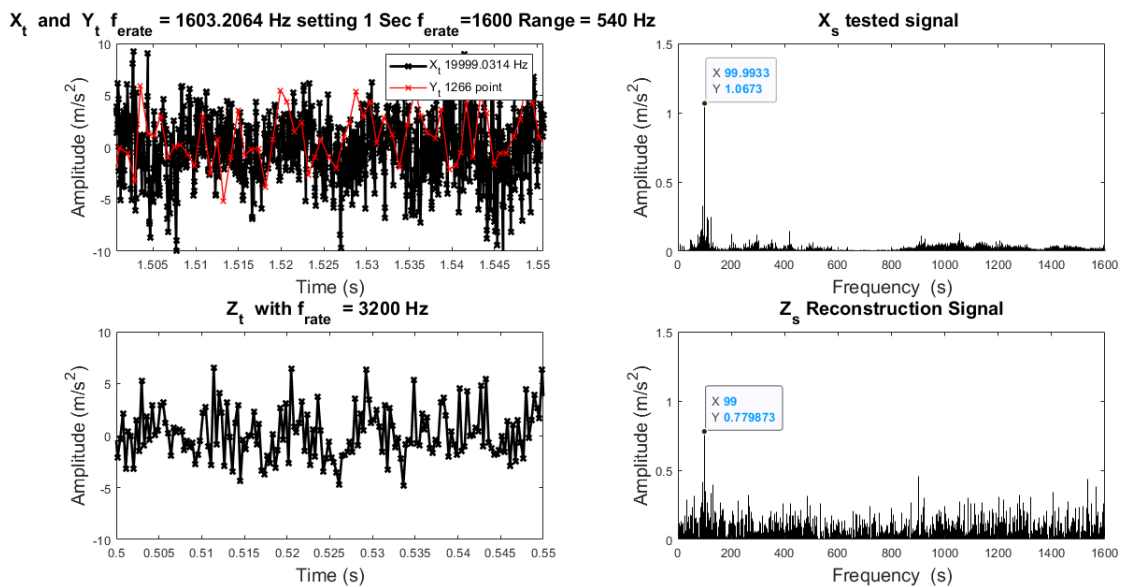
ตารางที่ 5.6 แสดงผลลัพธ์ของสัญญาณโดยเปรียบเทียบระหว่างสัญญาณคืนกลับที่ได้จากระบบการวัดแบบสุ่มคาบและสัญญาณต้นแบบที่ได้จากระบบการวัดแบบมาตรฐานด้วยเซ็นเซอร์ PCB ซึ่งจะแทนระดับคุณภาพของสัญญาณเป็นสีต่างๆ สีเขียวหมายถึงสัญญาณคืนกลับที่ถูกสร้างขึ้นมีตำแหน่งแอมพลิจูดที่ใกล้เคียงกับสัญญาณต้นแบบและสามารถแยกแยะสัญญาณรบกวนที่เป็นพื้นหลังกับสัญญาณจริงได้ชัดเจน สีส้มหมายถึงสัญญาณคืนกลับมีสัญญาณรบกวนและสัญญาณจริงปนกันและไม่สามารถแยกแยะได้อย่างชัดเจนแต่ยังพอจะสามารถทราบความถี่จริงที่เกิดขึ้นได้บ้าง สีแดงหมายถึงสัญญาณคืนกลับมีแถบสัญญาณรบกวนสูงมากและไม่สามารถแยกแยะหรือหาความคล้ายคลึงกับสัญญาณต้นแบบได้เลย โดยผลในรูป 5.26 – 5.27 แสดงตัวอย่างการประเมินในระดับสีเขียว รูป 5.28 แสดงผลระดับสีส้ม และรูป 5.29 แสดงผลในระดับสีแดง

จากตารางจะเห็นว่า สำหรับสัญญาณที่มีความถี่เดียวตั้งแต่ 300 ถึง 1000 เฮิรตซ์ ไม่ว่าจะใช้การปรับตั้งแบบใด ก็สามารถที่จะสร้างสัญญาณคืนกลับมาได้อย่างถูกต้องค่า $N_{y,1} \max$ จะต่ำกว่าความถี่สัญญาณทดสอบเกิน 1 เท่าเช่น ที่ความถี่สุ่มสูงสุด 800 และ 500 เฮิรตซ์ ก็ยังสามารถพอจะทราบความถี่ที่ถูกต้องบางส่วนของสัญญาณ 1000 เฮิรตซ์ และ 1200 เฮิรตซ์ได้ แต่สำหรับการสันสะเทือนที่ความถี่ 1500 เฮิรตซ์ ขึ้นไปที่ไม่สามารถสร้างสัญญาณคืนกลับได้ถูกต้อง สาเหตุที่อาจเป็นไปได้ อาจมาจากผลของ วงจรกรองความถี่ต่ำผ่าน ซึ่งบริเวณที่อยู่ใกล้ Cut off frequency ซึ่งจะมีความเพี้ยนของการวัด และถ้าหากการวัดสัญญาณ y_t เกิดความผิดเพี้ยนไปแล้วแน่นอนว่าการสร้างสัญญาณคืนกลับที่สร้างได้ผิดเพี้ยนไปเช่นกัน

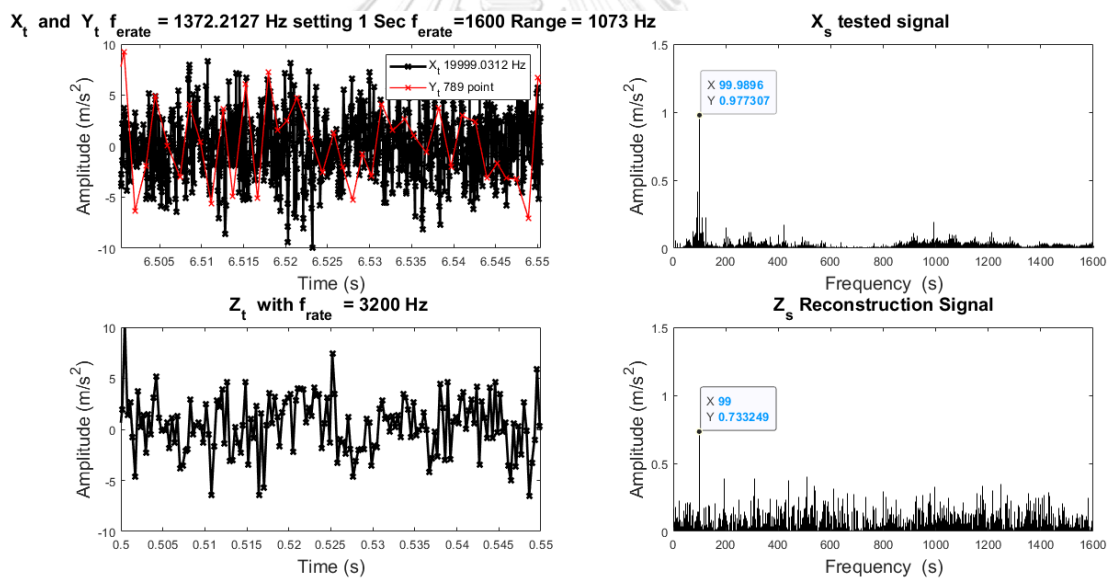
สำหรับสัญญาณมอดูเลตจะเห็นว่าการปรับตั้งส่วนใหญ่จะสามารถวัดได้สูงสุดเพียง 800 เฮิรตซ์เท่านั้น และจะลดลงเหลือเพียง 300 เฮิรตซ์เมื่อถูกปรับตั้งให้ความถี่สุ่มสูงสุด มีค่าน้อยลงซึ่งสอดคล้องกับผลจำลองในหัวข้อที่ 5.1.1 อีกข้อสังเกตหนึ่งคือ แอมพลิจูดของ Sideband ของสัญญาณสันสะเทือนที่สร้างขึ้นมีค่าต่ำมากเมื่อเทียบกับความถี่หลัก (เนื่องจากขีดจำกัดของชุดกำเนิดสัญญาณ) ให้คอมเพรชชันซึ่งไม่สามารถสร้างสัญญาณคืนกลับออกมาได้ถูกต้อง เนื่องจากสัญญาณที่มีขนาดเล็กจะถูกลิ้นเข้าไปกับแถบของสัญญาณรบกวนทำให้ไม่สามารถจะเห็นรายละเอียดที่ถูกต้องได้ครบถ้วน

5.2.2 ผลการวัดการสันสะเทือนของเครื่องจักรจริง

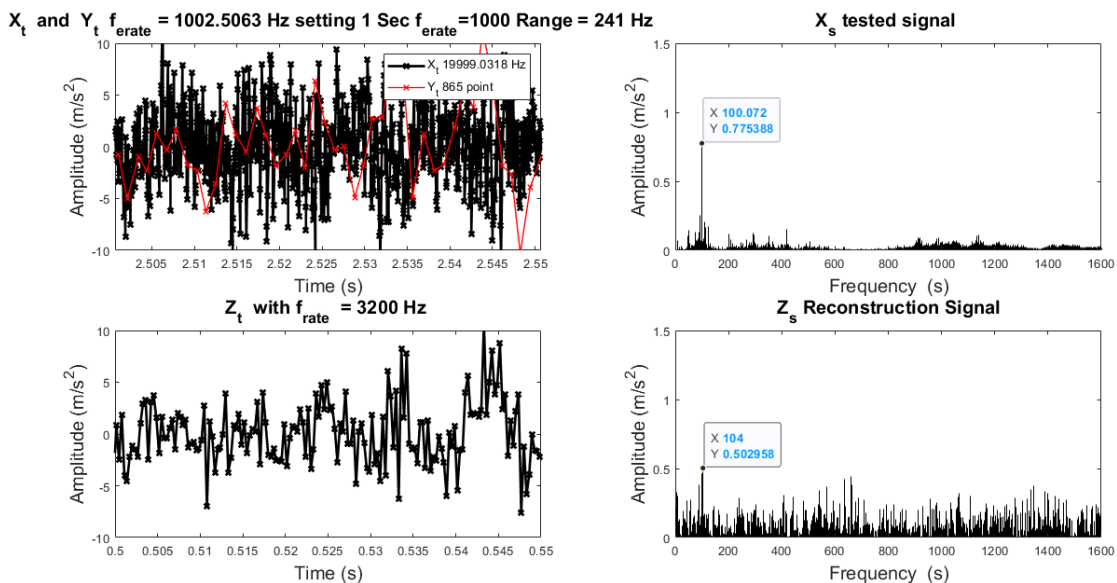
การทดสอบนี้จะวัดการสันสะเทือนของเครื่องคอมเพรชเซอร์แบบลูกสูบทั้งหมด 2 ตัว คือ แบบ 1 ลูกสูบและแบบ 2 ลูกสูบ ในทิศทางแนวนอน โดยจะวัดด้วยระบบมาตรฐานและระบบสุ่มคาบพร้อมกัน เนื่องจากในความเป็นจริงการสันสะเทือนของเครื่องคอมเพรชเซอร์ไม่คงตัว ตลอดห้วงเวลาการวัด การตรวจสอบความถูกต้องของสัญญาณคืนกลับจำเป็นจะต้องนำไปเทียบกับสัญญาณ ในเวลาเดียวกันที่ถูกวัดโดยระบบวัดและเซ็นเซอร์ มาตรฐาน โดยการปรับตั้งระบบการวัดแบบสุ่มคาบจะเลือกปรับตั้งทั้งหมด 8 แบบ เช่นเดียวกับการวัดการสันสะเทือนของเครื่องกำเนิดสัญญาณ ซึ่งจะได้สัญญาณรวมทั้งหมด 16 สัญญาณ ตามรูปและตารางด้านล่าง



รูปที่ 5.30 ตัวอย่างผลการทดสอบการการรบกวนสัญญาณกลับคืนจากการวัดด้วยการตั้งค่าแบบที่ 1 คอมเพรชเซอร์แบบลูกสูบ 1 ลูกสูบ

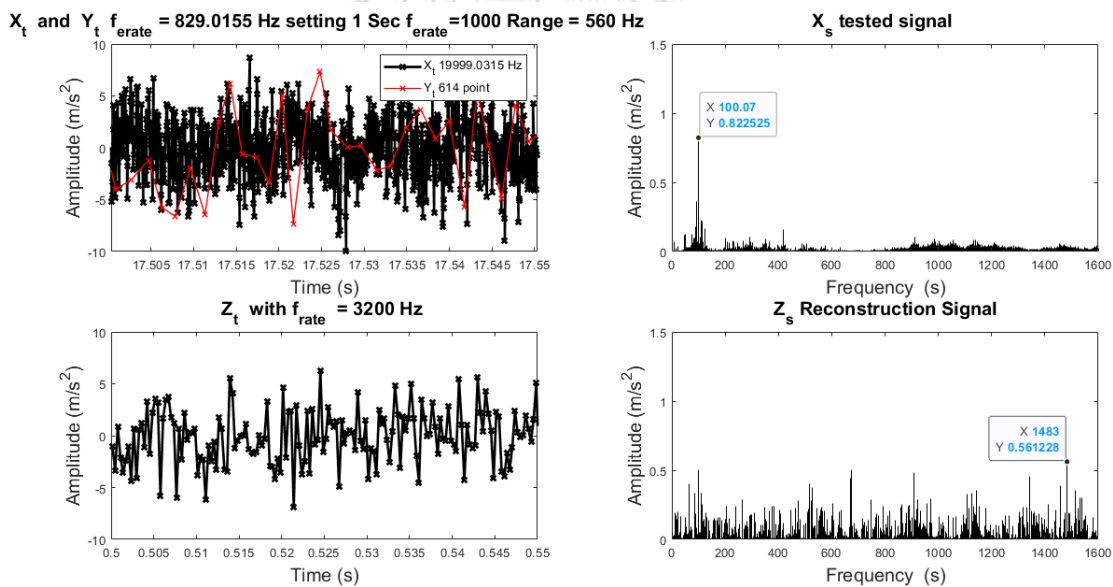


รูปที่ 5.31 ตัวอย่างผลการทดสอบการการรบกวนสัญญาณกลับคืนจากการวัดด้วยการตั้งค่าแบบที่ 2 คอมเพรชเซอร์แบบลูกสูบ 1 ลูกสูบ



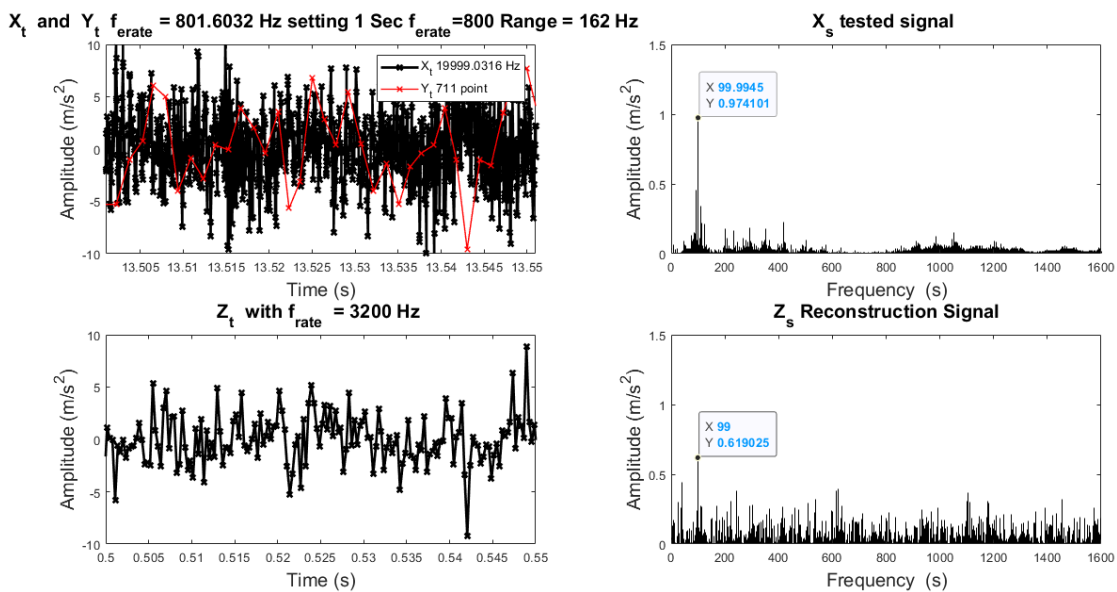
รูปที่ 5.32 ตัวอย่างผลการทดสอบการการรบกวนการกลับคืนจากการวัดด้วยการตั้งค่าแบบที่ 3 คอมเพรช

เซอร์เบบลูกสูบ 1 ลูกสูบ

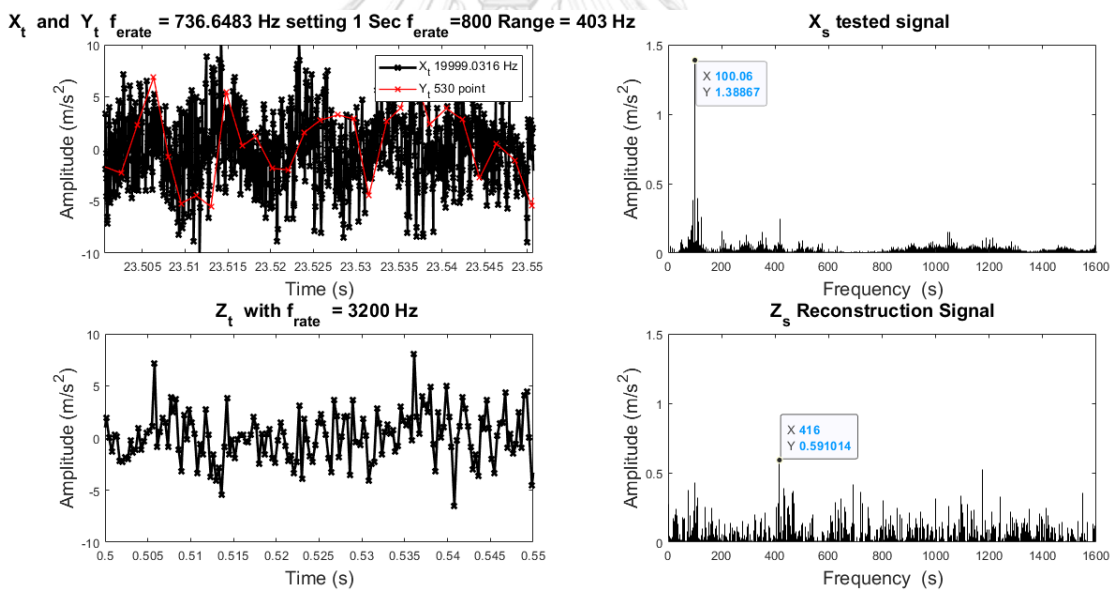


รูปที่ 5.33 ตัวอย่างผลการทดสอบการการรบกวนการกลับคืนจากการวัดด้วยการตั้งค่าแบบที่ 4 คอมเพรช

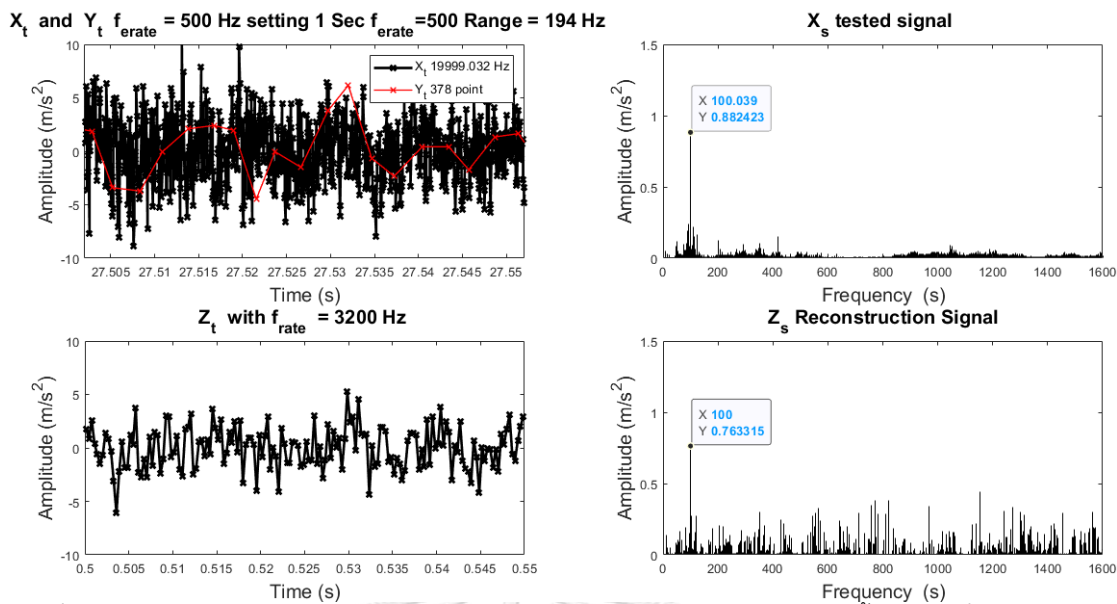
เซอร์เบบลูกสูบ 1 ลูกสูบ



รูปที่ 5.34 ตัวอย่างผลการทดสอบการการรบกวนกลับคืนจากการวัดด้วยการตั้งค่าแบบที่ 5 คอมเพรสเซอร์แบบลูกสูบ 1 ลูกสูบ

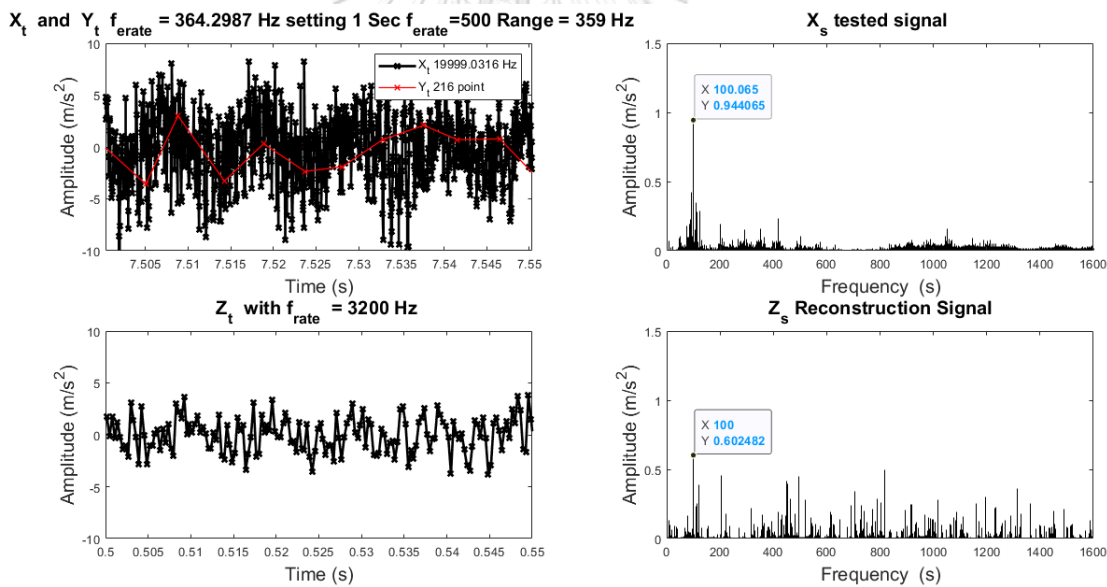


รูปที่ 5.35 ตัวอย่างผลการทดสอบการการรบกวนกลับคืนจากการวัดด้วยการตั้งค่าแบบที่ 6 คอมเพรสเซอร์แบบลูกสูบ 1 ลูกสูบ



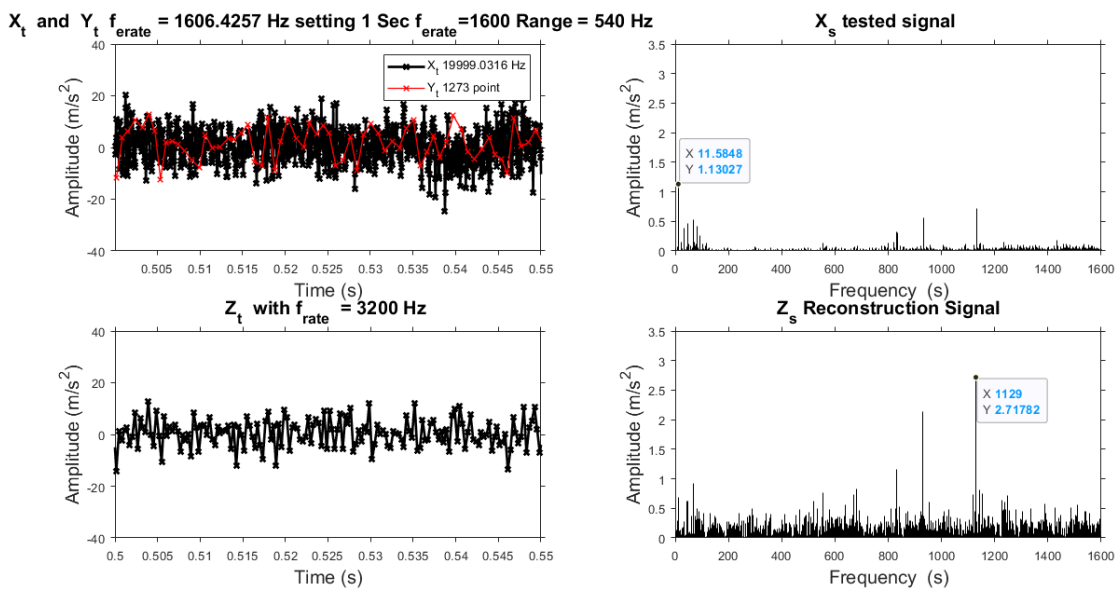
รูปที่ 5.36 ตัวอย่างผลการทดสอบการการรบกวนสัญญาณกลับคืนจากการวัดด้วยการตั้งค่าแบบที่ 7 คอมเพรช

เซอร์แบบลูกสูบ 1 ลูกสูบ

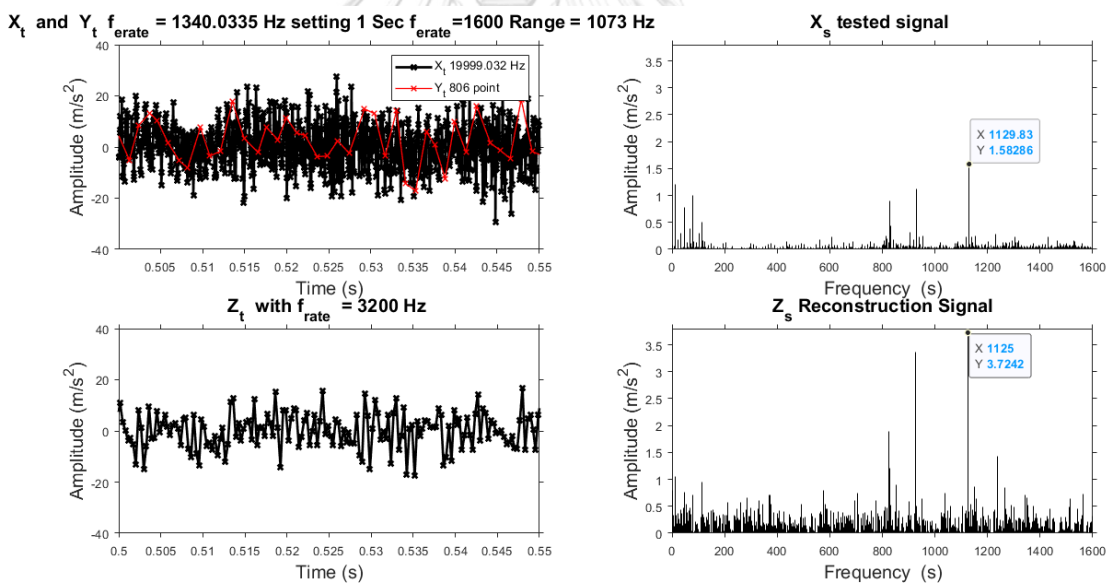


รูปที่ 5.37 ตัวอย่างผลการทดสอบการการรบกวนสัญญาณกลับคืนจากการวัดด้วยการตั้งค่าแบบที่ 8 คอมเพรช

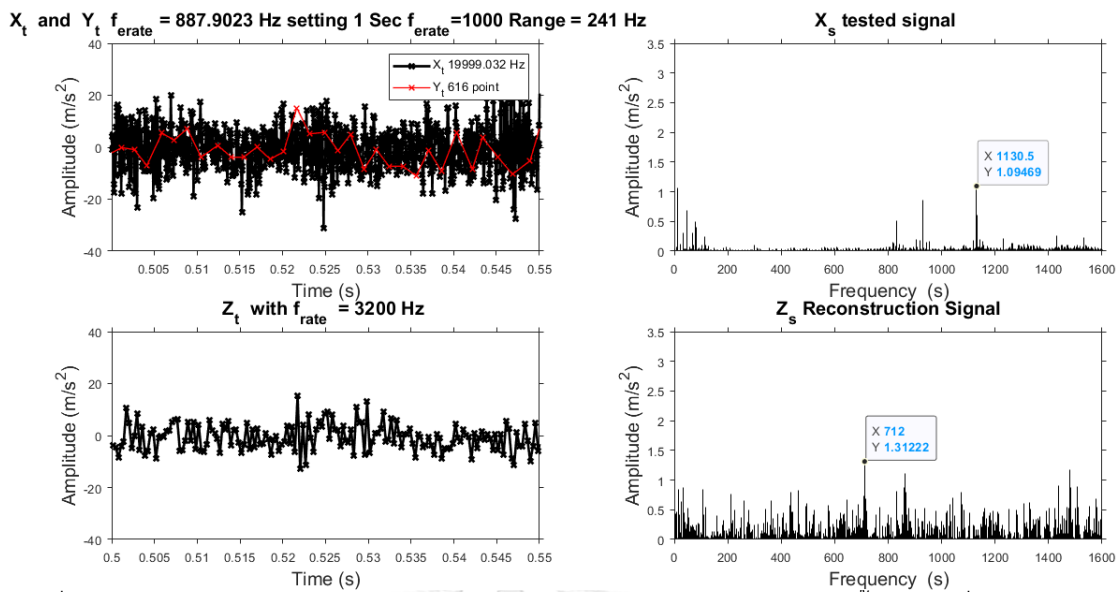
เซอร์แบบลูกสูบ 1 ลูกสูบ



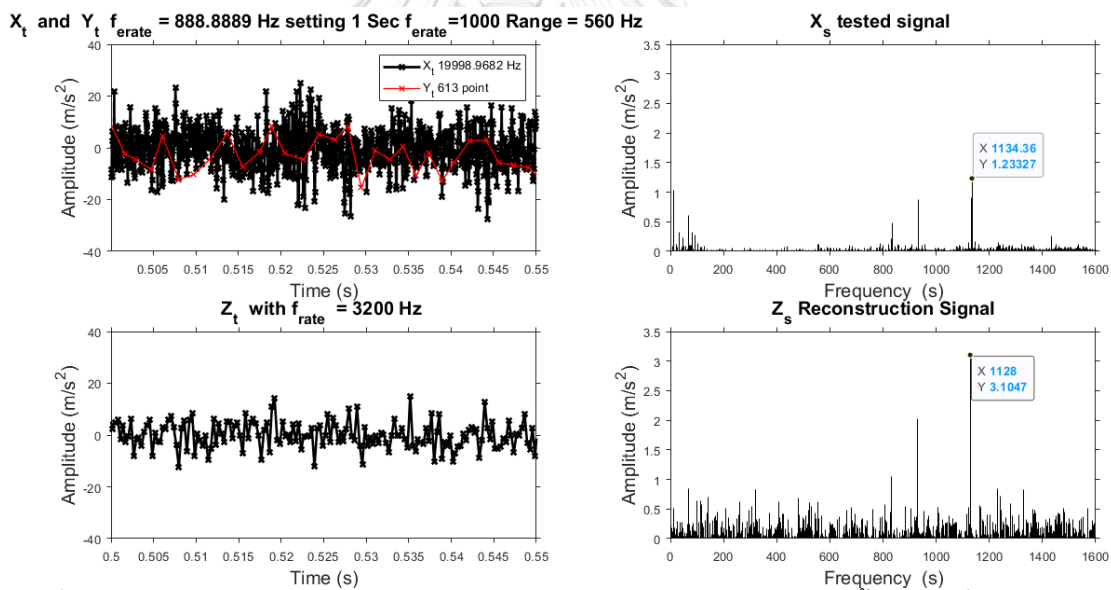
รูปที่ 5.38 ตัวอย่างผลการทดสอบการการรบกวนกลับคืนจากการวัดด้วยการตั้งค่าแบบที่ 1 คอมเพรชเซอร์แบบลูกสูบ 2 ลูกสูบ



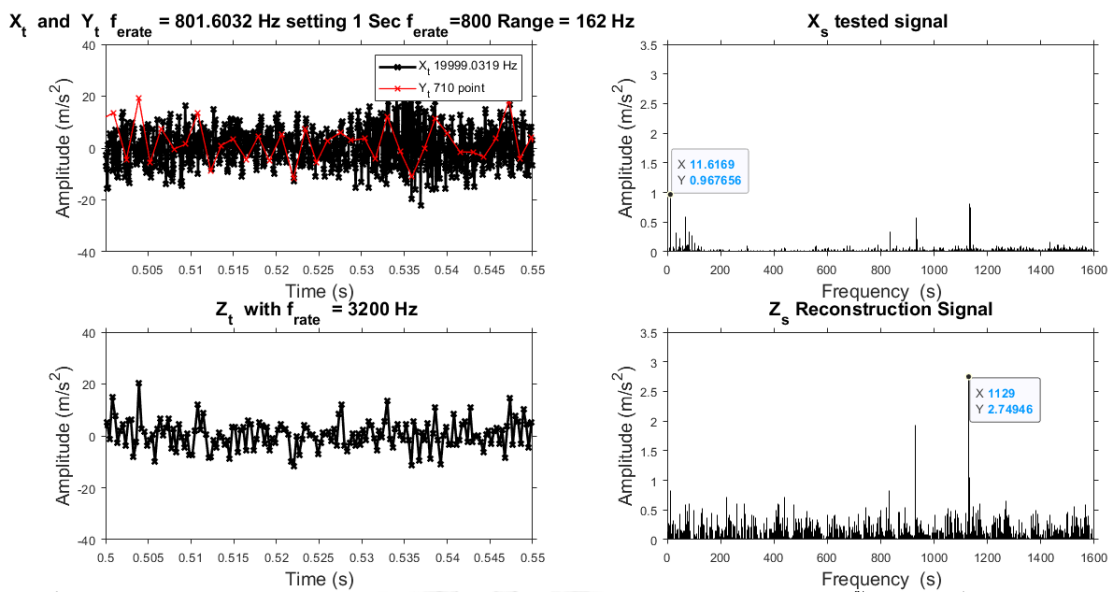
รูปที่ 5.39 ตัวอย่างผลการทดสอบการการรบกวนกลับคืนจากการวัดด้วยการตั้งค่าแบบที่ 2 คอมเพรชเซอร์แบบลูกสูบ 2 ลูกสูบ



รูปที่ 5.40 ตัวอย่างผลการทดสอบการการรบกวนกลับคืนจากการวัดด้วยการตั้งค่าแบบที่ 3 คอมเพรสเซอร์แบบลูกสูบ 2 ลูกสูบ

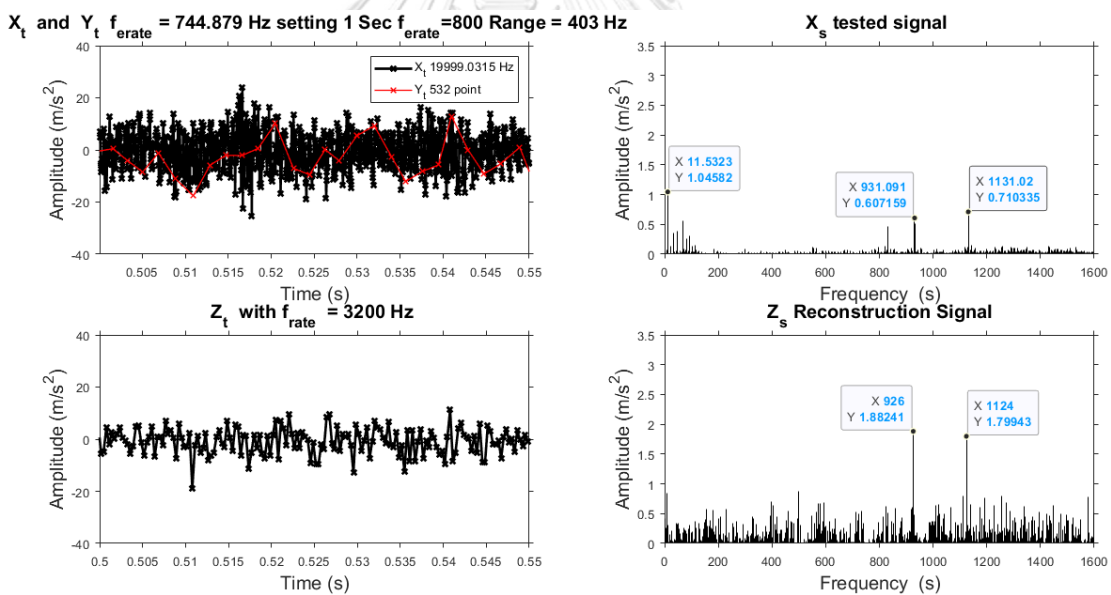


รูปที่ 5.41 ตัวอย่างผลการทดสอบการการรบกวนกลับคืนจากการวัดด้วยการตั้งค่าแบบที่ 4 คอมเพรสเซอร์แบบลูกสูบ 2 ลูกสูบ



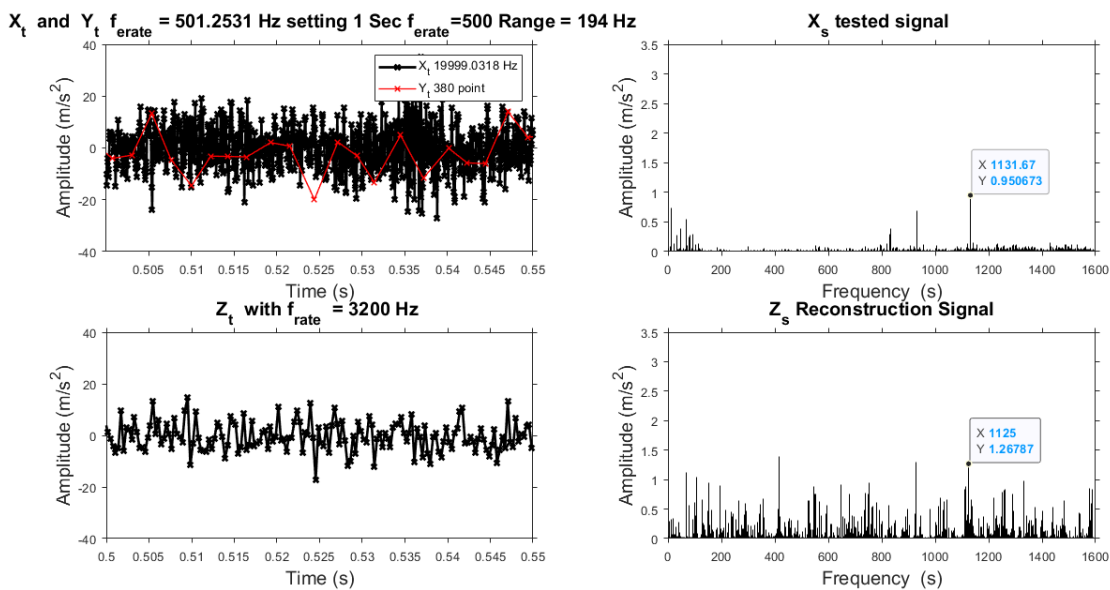
รูปที่ 5.42 ตัวอย่างผลการทดสอบการการรบกวนกลับคืนจากการวัดด้วยการตั้งค่าแบบที่ 5 คอมเพรช

เซอร์แบบลูกสูบ 2 ลูกสูบ

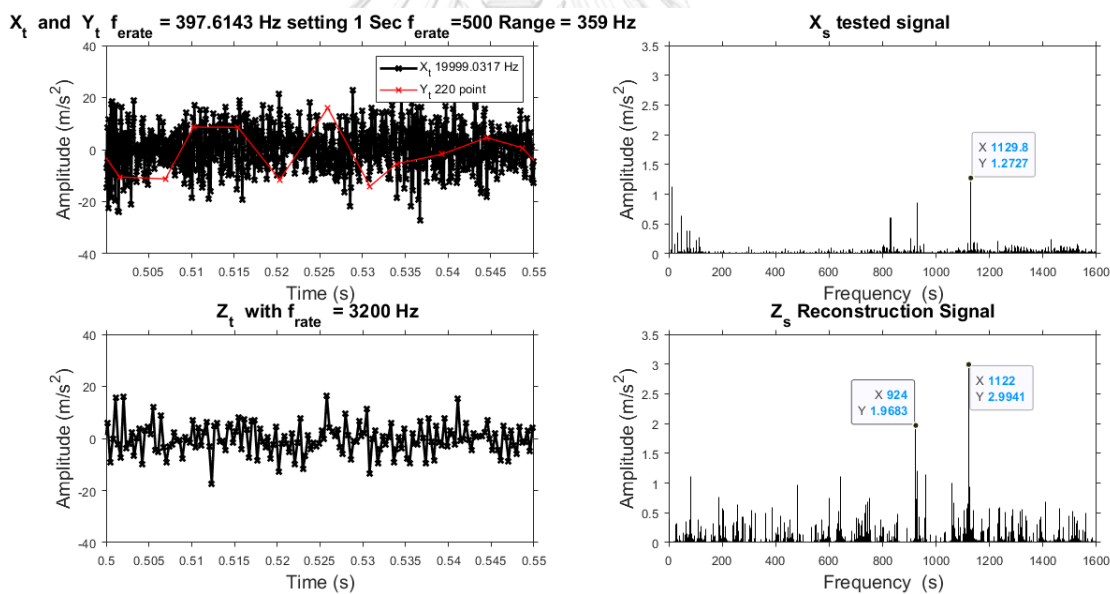


รูปที่ 5.43 ตัวอย่างผลการทดสอบการการรบกวนกลับคืนจากการวัดด้วยการตั้งค่าแบบที่ 6 คอมเพรช

เซอร์แบบลูกสูบ 2 ลูกสูบ



รูปที่ 5.44 ตัวอย่างผลการทดสอบการการรบกวนกลับคืนจากการวัดด้วยการตั้งค่าแบบที่ 7 คอมเพชเชอร์แบบลูกสูบ 2 ลูกสูบ



รูปที่ 5.45 ตัวอย่างผลการทดสอบการการรบกวนกลับคืนจากการวัดด้วยการตั้งค่าแบบที่ 8 คอมเพชเชอร์แบบลูกสูบ 2 ลูกสูบ

ตารางที่ 5.7 แสดงการผลลัพธ์ของสัญญาณคืนกลับที่ถูกวัดด้วยการปรับตั้งแบบต่างๆ

แบบที่		1	2	3	4	5	6	7	8
PSC		20	20	20	20	20	20	80	80
ARR		500	500	800	800	1000	1000	400	400
Multiply		1	4	1	4	1	4	1	4
$N_{y,1}$	min	1059.60	526.32	758.29	439.56	637.45	396.0396	305.3435	140.8451
	mean	1273.89	790.51	862.07	609.76	709.22	529.10053	378.7879	219.2982
	max	1600	1600	1000	1000	800	800	500	500
f_{range}		540.40	1073.68	241.71	560.44	162.55	403.9604	194.6565	359.1549
Compressor 1 ลูกสูบ		△	△	X	X	△	X	△	△
Compressor 2 ลูกสูบ		O	△	X	X	△	△	X	△

โดยภาพรวมแล้วจะเห็นได้ชัดเจนว่าเมื่อสัญญาณมีความซับซ้อนมากขึ้น การสร้างสัญญาณคืนกลับก็จะมีคุณภาพลดลงอย่างเห็นได้ชัด สัญญาณคืนกลับแทบทั้งหมดมีคุณภาพที่ไม่ดีนักและมีเพียง สัญญาณเดียวที่มีคุณภาพดีเพียงพอจะนำไปใช้งานหรือนำไปวิเคราะห์ได้ คือที่การปรับตั้งแบบที่ 1 ในการวัดคอมเพรสเซอร์ 2 ลูกสูบ ซึ่งสอดคล้องกับผลการทดสอบในตารางที่ 5.5 และผลการทดสอบหัวข้อที่ 5.1.2.4 ซึ่งพบว่าเมื่อความถี่สูงเทียบเท่ามากขึ้น ก็มักจะส่งผลให้คุณภาพของสัญญาณคืนกลับดีขึ้นเช่นกัน ส่วนที่ความถี่สูงเทียบเท่าต่ำกว่นั้น ก็จะทำให้ผลลัพธ์ที่มีคุณภาพไม่ดี และจะเห็นได้ว่าการสันสะเทือนของคอมเพรสเซอร์แบบ 2 ลูกสูบนั้นมีแนวโน้มที่จะสร้างสัญญาณกลับคืนออกมาได้ง่ายกว่าโดยสังเกตได้ว่า ที่สัญญาณคอมเพรสเซอร์แบบ 2 ลูกสูบจะสามารถแยกแยะ สัญญาณความถี่หลักออกจากแถบสัญญาณรบกวนได้ชัดเจนมากกว่า เช่น จากในรูปที่ 5.32 และ 5.40 เมื่อพิจารณาลักษณะของสัญญาณต้นแบบของคอมเพรสเซอร์ทั้ง 2 จะพบว่าถึงแม้คอมเพรสเซอร์แบบ 2 ลูกสูบนั้นจะมีจำนวนความถี่หลักมากกว่า คือ 6 ความถี่ เทียบกับคอมเพรสเซอร์แบบ 1 ลูกสูบที่มีเพียง 1-3 ความถี่ แต่ลักษณะของแถบความถี่รบกวนที่มีแอมพลิจูดต่ำของ คอมเพรสเซอร์แบบ 1 ลูกสูบมีลักษณะเป็นบรอดแบนด์ สูงกว่าคอมเพรสเซอร์แบบ 2 ลูกสูบมาก ประกอบกับขนาดของการสันสะเทือนของความถี่หลักของสัญญาณคอมเพรสเซอร์แบบ 1 ลูกสูบที่น้อยน้อยกว่าแล้ว ส่งผลให้คอมเพรสเซอร์ซึ่งสามารถสร้างสัญญาณคืนกลับของคือเพรสเซอร์แบบ 2 ลูกสูบได้ดีกว่า

5.2.2.1 การใช้ชุดข้อมูลที่กว้างขึ้นในการทดสอบ

จากที่ได้กล่าวมาในหัวข้อ 4.3.2 การเพิ่มความยาวการเก็บข้อมูลสามารถเพิ่มคุณภาพของสัญญาณคืนกลับได้ สำหรับในหัวข้อนี้จะเลือกกรณีสัญญาณคืนกลับที่มีคุณภาพต่ำที่สุดจากหัวข้อก่อนหน้า มาทดสอบเพิ่มเติมโดยการเพิ่มความยาวการใช้ข้อมูลของกรณีเหล่านี้สำหรับกรณีที่นำมาทำเพิ่มเติมได้แก่

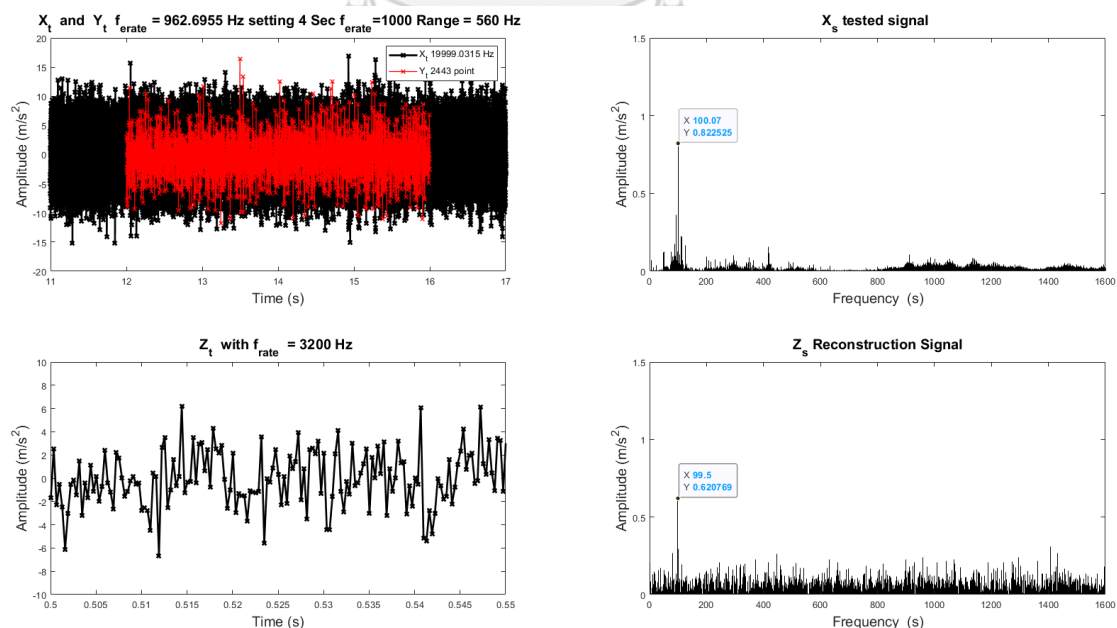
1. สัญญาณของเครื่องคอมเพรสเซอร์แบบ 1 ลูกสูบที่ใช้การปรับตั้งการวัดแบบที่ 4 ซึ่งมีความถี่สูงเทียบเท่า 1000 เฮิร์ตซ์ เพิ่มความยาวจาก 1 วินาที เป็น 4 วินาที
2. สัญญาณของเครื่องคอมเพรสเซอร์แบบ 2 ลูกสูบที่ใช้การปรับตั้งการวัดแบบที่ 8 ซึ่งมีความถี่สูงเทียบเท่า 500 เฮิร์ตซ์ เพิ่มความยาวจาก 1 วินาที เป็น 10 วินาที

จากรูปที่ 5.30 ถึง 5.45 เป็นการแสดงผลการทดสอบการใช้สัญญาณที่ยาวขึ้น ของสัญญาณ 2 แบบที่กล่าวมาข้างต้นโดยจะแสดงถึงความยาวของสัญญาณที่เพิ่มขึ้นมาเมื่อเปรียบเทียบกับสัญญาณเดิม และผลของการสร้างสัญญาณคืนกลับบนโดเมนความถี่เมื่อเปรียบเทียบกับระหว่าง สัญญาณของเครื่องจักรจริงจากเซ็นเซอร์มาตรฐาน สัญญาณที่ใช้ความยาวของสัญญาณสุ่มความถี่ 1 วินาที และ สัญญาณที่ใช้ความยาวของสัญญาณสุ่มความถี่ที่ยาวกว่านั้น (4 วินาที สัญญาณของเครื่องคอมพิวเตอร์แบบ 1 และ 10วินาที สัญญาณของเครื่องคอมพิวเตอร์แบบ 2 ลูกสูบ)

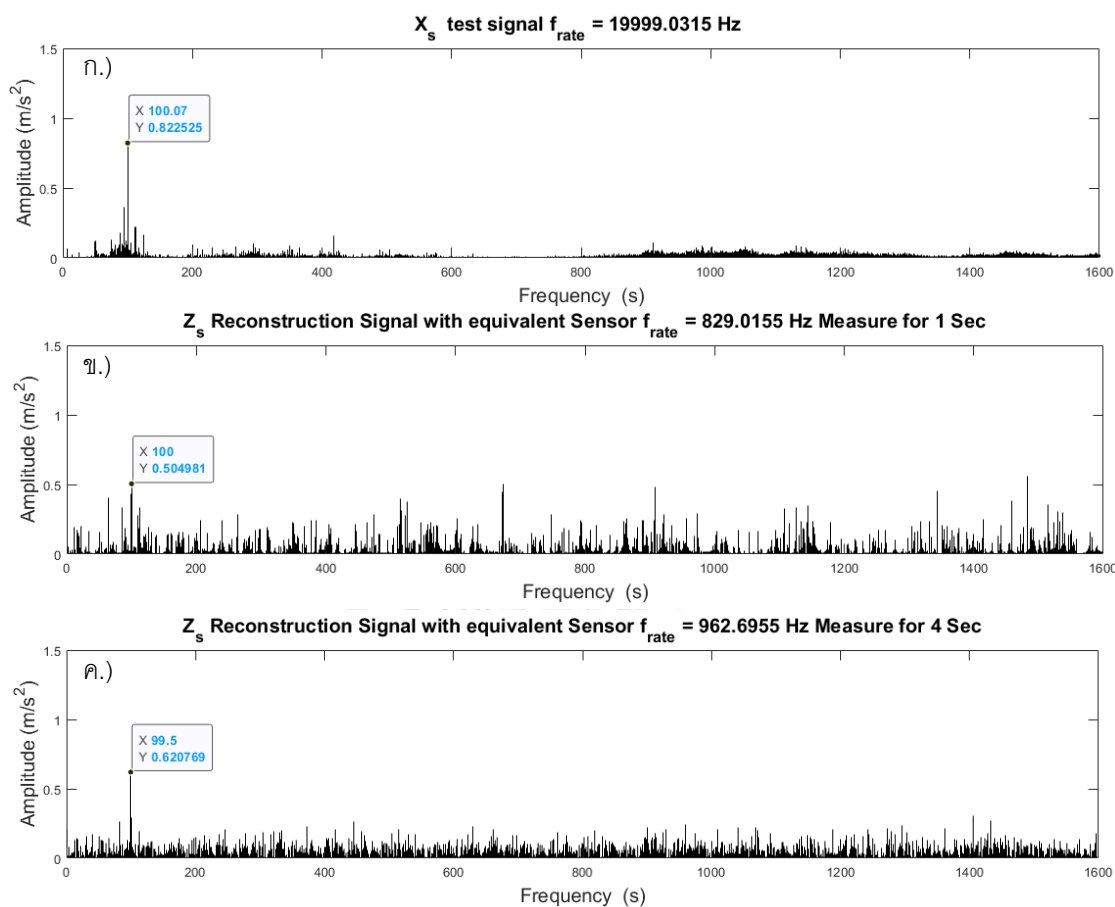
สำหรับสาเหตุที่ใช้ สัญญาณความยาวที่ไม่เท่ากันเนื่องจาก ข้อจำกัดเรื่องจำนวนจุดข้อมูลโดยเฉลี่ยของสัญญาณสุ่ม โดยสัญญาณ ที่ 1 มีความยาวเฉลี่ย 609 จุด ต่อ 1 วินาที สัญญาณ ที่ 2 มีความยาวเฉลี่ย 219 จุดต่อวินาที เมื่อสัญญาณที่ 1 เพิ่มความยาว เป็น 4 วินาที จำนวนจุดทั้งหมดจะมี ความยาวอยู่ที่ประมาณ 2400 จุด เช่นเดียวกับ สัญญาณที่ 2 ถ้าหากเพิ่มความยาวเป็น 10 วินาที จำนวนจุดทั้งหมดจะอยู่ที่ประมาณ 2200 ซึ่งใกล้เคียงกันกับสัญญาณก่อนหน้าแต่ระยะเวลาแตกต่างกัน ทั้ง 2 สัญญาณที่นำมาทดสอบนี้เนื่องจากความพยายามที่จะใช้ความยาวสัญญาณให้มากที่สุดเท่าที่ หน่วยความจำของคอมพิวเตอร์สามารถรองรับได้ จึงเป็นสาเหตุให้ สัญญาณที่เลือกมาทั้ง 2 สัญญาณ มีความยาวไม่เท่ากัน

การทดสอบสัญญาณของเครื่องคอมพิวเตอร์แบบ 1 ลูกสูบที่ใช้การปรับตั้งการวัดแบบที่ 4 ซึ่งมีความถี่สุ่มเทียบเท่า 1000 เฮิรตซ์

ในการทดสอบนี้จะเพิ่มความยาวของสัญญาณจากเดิมที่ใช้สัญญาณ y_t ที่มีความยาว 1 วินาทีเป็น 4 วินาที ตามรูปด้านล่าง



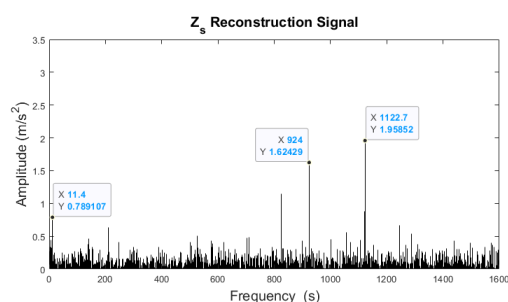
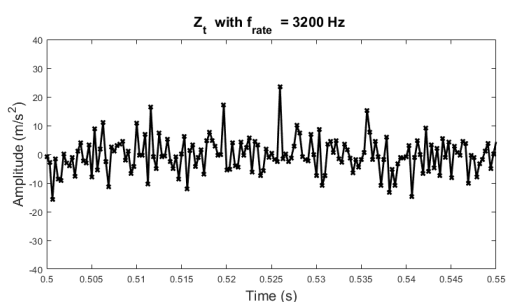
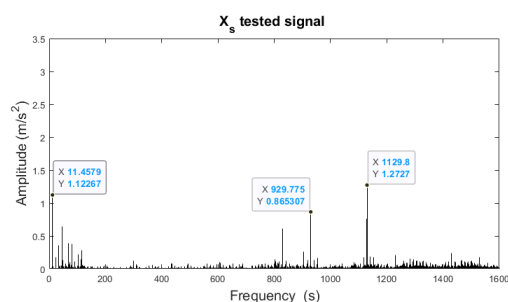
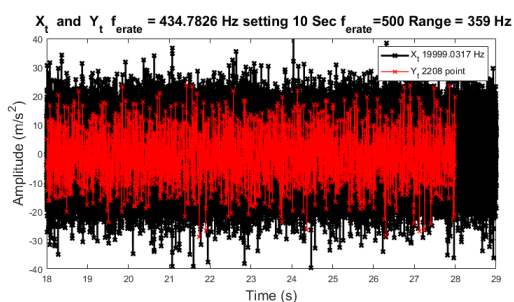
รูปที่ 5.46 ผลการสร้างสัญญาณคืนกลับของคอมพิวเตอร์แบบ 1 ลูกสูบด้วยการปรับตั้งแบบที่ 4 ที่ใช้ความยาวของสัญญาณวัดแบบสุ่ม y_t 4 วินาที



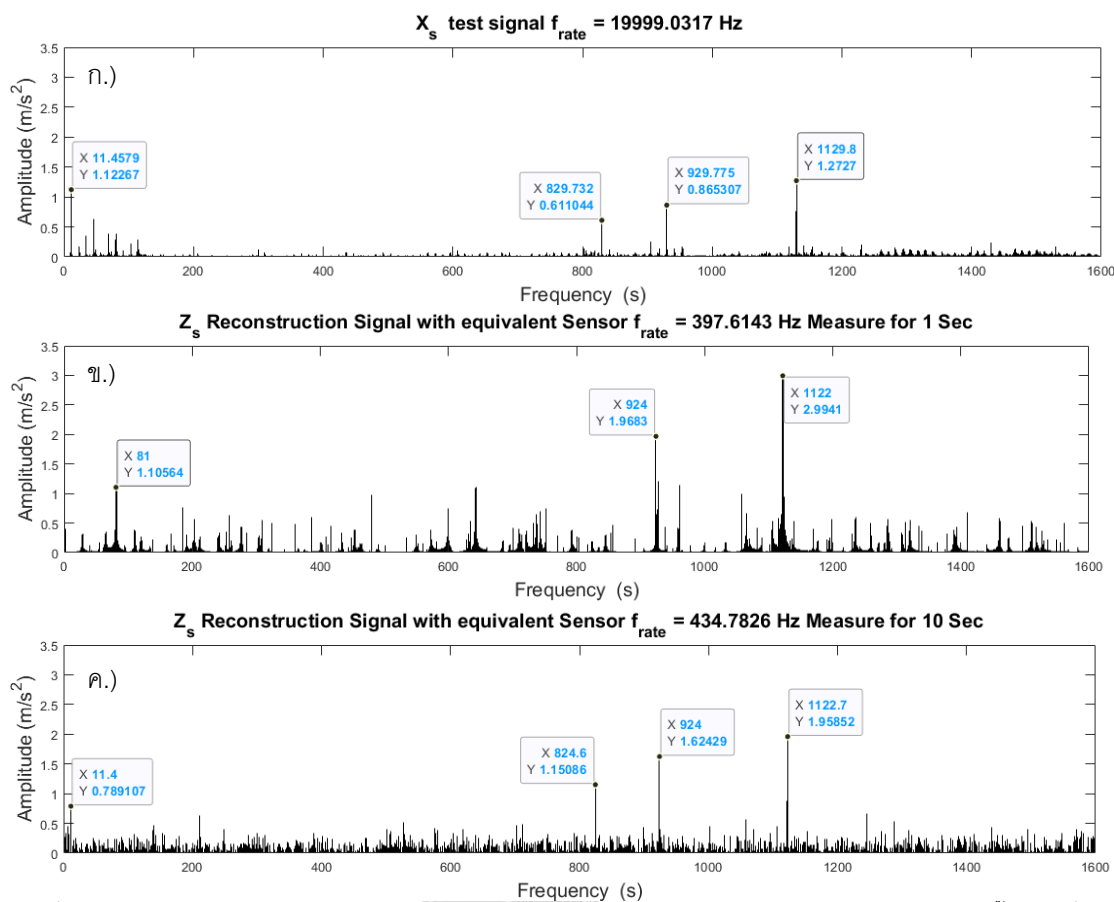
รูปที่ 5.47 เปรียบเทียบผลการสร้างสัญญาณคืนกลับของคอมพิวเตอร์แบบ 1 ลูกคู่ด้วยการปรับตั้งแบบที่ 4 ในความยาวของสัญญาณวัดแบบสุ่ม y_t ความยาวต่างๆ ก.) สัญญาณต้นแบบ ข.) 1 วินาที ค.) 4 วินาที

การทดสอบสัญญาณของเครื่องคอมพิวเตอร์แบบ 2 ลูกสูบที่ใช้การปรับตั้งการวัดแบบที่ 8 ซึ่งมีความถี่สัมพันธ์เท่ากับ 500 เฮิรตซ์

ในการทดสอบนี้การปรับตั้งแบบนี้จากเดิมที่ใช้สัญญาณ y_t ที่มีความยาว 1 วินาที จะเพิ่มเป็น 10 วินาทีตามรูปด้านล่าง



รูปที่ 5.48 ผลการสร้างสัญญาณคืนกลับของคอมพิวเตอร์แบบ 2 ลูกสูบด้วยการปรับตั้งแบบที่ 8 ที่ใช้ความยาวของสัญญาณวัดแบบสุ่ม y_t 10 วินาที



รูปที่ 5.49 เปรียบเทียบผลการสร้างสัญญาณคืนกลับของคอมพิวเตอร์แบบ 2 ลูกสูบด้วยการปรับตั้งแบบที่ 8 ในความยาวของสัญญาณวัดแบบสุ่ม y_t ความยาวต่างๆ ก.) สัญญาณต้นแบบ ข.) 1 วินาที ค.) 10 วินาที

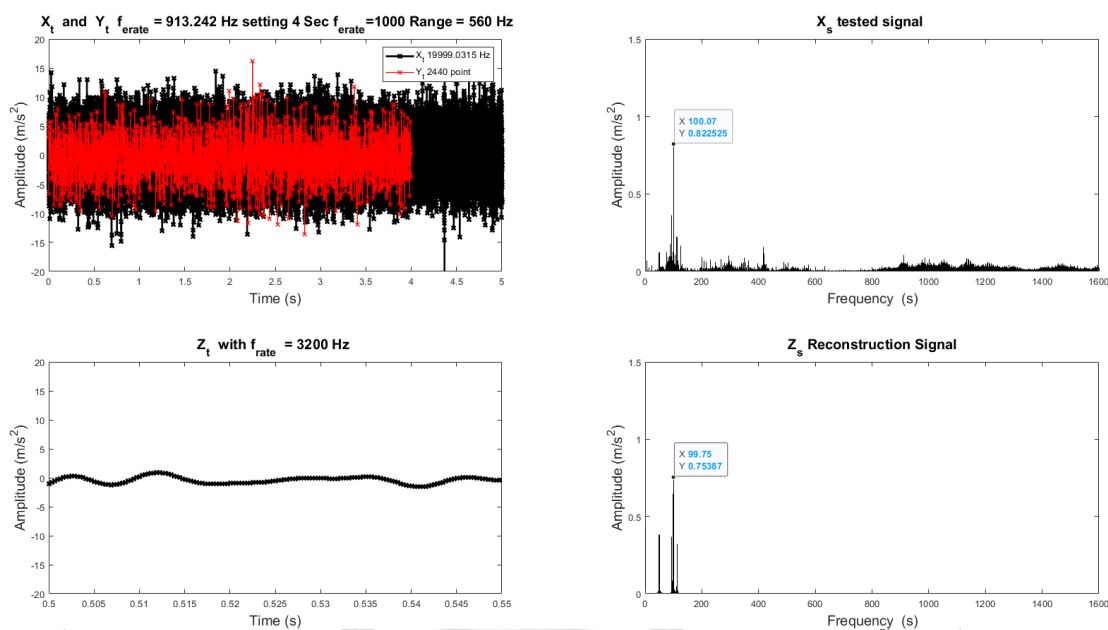
จากการทดสอบโดยเพิ่มความยาวของสัญญาณ y_t ในกรณีที่สัญญาณ จะเห็นว่าการเพิ่มความยาวสัญญาณสามารถช่วยเพิ่มคุณภาพของสัญญาณให้ดีขึ้นได้ และถึงแม้ว่าจะปรับตั้งให้ความถี่สุ่มเทียบเท่าลดลงมาเหลือเพียง 500 เฮิรตซ์สำหรับการปรับตั้งแบบที่ 8 ก็ยังสามารถที่จะสร้างสัญญาณกลับคืนที่มีย่านการวัดถึง 1600 เฮิรตซ์ได้อย่างน่าพอใจ ซึ่งโดยปกติแล้ว ระบบการวัดที่มีความถี่สุ่ม 500 เฮิรตซ์จะสามารถวัดย่านความถี่สูงสุดได้เพียงแค่ 250 เฮิรตซ์เท่านั้นตามทฤษฎีของไนควิสต์

ถึงแม้ว่าการเพิ่มความยาวของสัญญาณ จะทำให้สามารถสร้างสัญญาณคืนกลับด้วยวิธีการหาค่าเหมาะสมที่สุดแบบคอนเวกซ์ได้ดีขึ้น อย่างไรก็ตามการเพิ่มความยาวของสัญญาณ ส่งผลโดยตรงต่อขนาดของเมทริกซ์การแปลงฐานหลัก และจำนวนสมการในการหาค่าตอบแบบคอนเวกซ์ ทำให้เกิดปัญหาในด้านของหน่วยความจำและการประมวลผลเพื่อหาผลลัพธ์ของคอมพิวเตอร์ที่ใช้ เพื่อแก้ปัญหาเหล่านี้ ในการศึกษานี้ได้ทดสอบใช้อัลกอริทึมคอมพิวเตอร์เชิงอื่น ๆ ได้แก่ วิธีจับคู่สัญญาณบีบอัด และวิธีการซ่อนสัญญาณเฉลี่ย ซึ่งจะแสดงผลลัพธ์ในหัวข้อถัดไป

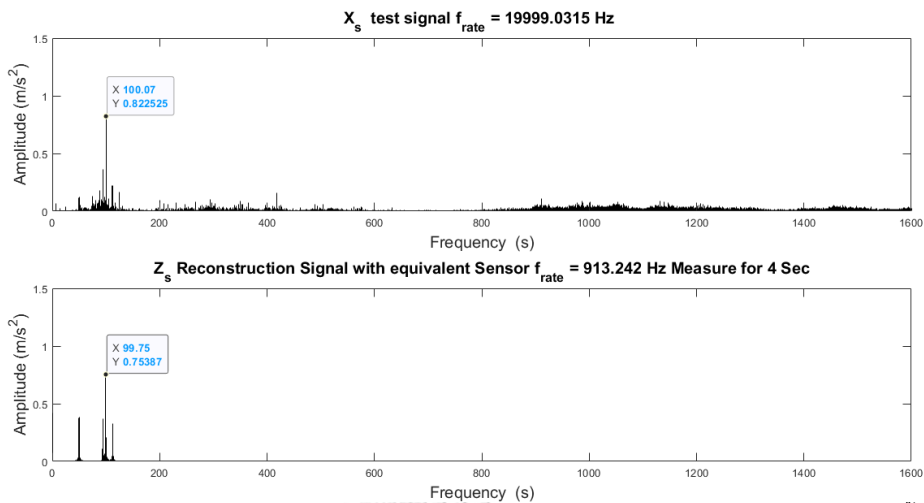
5.2.3 การใช้อัลกอริทึมอื่น ๆ แทนการใช้ การหาค่าเหมาะสมที่สุดแบบคอนเวกซ์

ทดสอบโดยใช้การใช้อัลกอริทึมจับคู่สัญญาณบีบอัด

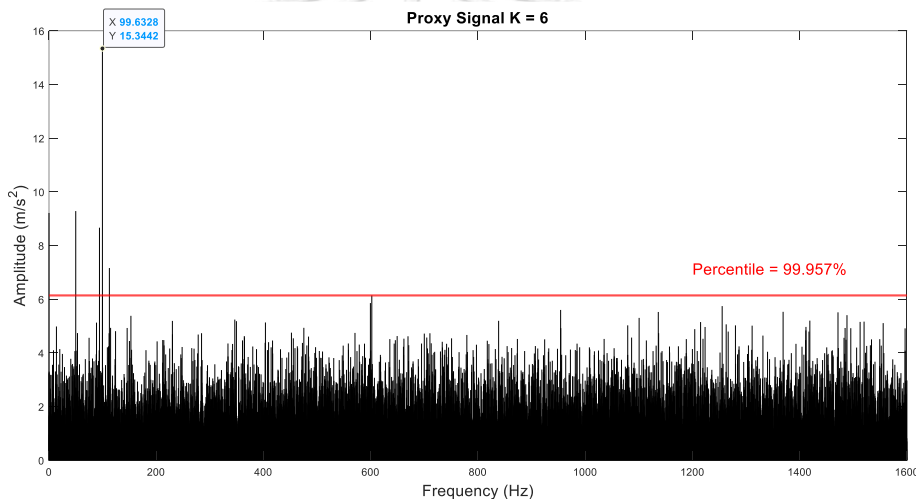
การทดสอบโดยใช้การใช้อัลกอริทึมชนิดนี้จะเลือกใช้สัญญาณที่นำมาทดสอบ 2 สัญญาณ คือ สัญญาณของคอมเพรสเซอร์ 1 ลูกสูบที่ใช้การปรับตั้งแบบที่ 4 วัดด้วยความยาวสัญญาณ 4 วินาที และ คอมเพรสเซอร์ 2 ลูกสูบที่ใช้การปรับตั้งแบบที่ 8 วัดด้วยความยาวสัญญาณ 10 วินาที สำหรับคอมเพรสเซอร์ 1 ลูกสูบจะใช้ค่า K เท่ากับ 6 สำหรับคอมเพรสเซอร์ 2 ลูกสูบจะใช้ค่า K เท่ากับ 15



รูปที่ 5.50 ผลการสร้างสัญญาณคืนกลับของคอมเพรสเซอร์แบบ 1 ลูกสูบด้วยการปรับตั้งแบบที่ 4 โดยใช้การจับคู่สัญญาณบีบอัด ที่ค่า K เท่ากับ 6

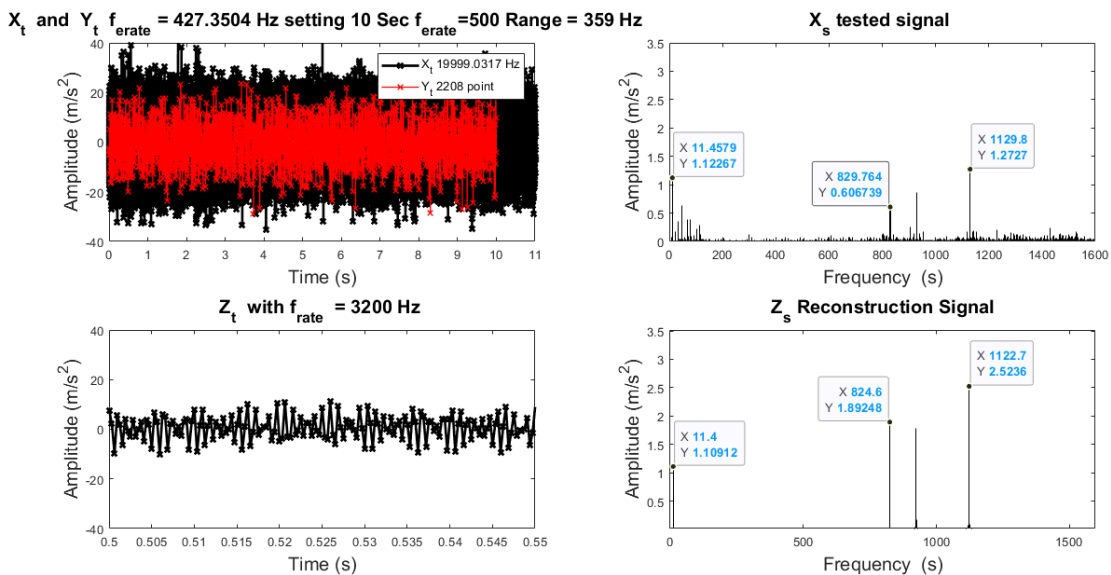


รูปที่ 5.51 เปรียบเทียบผลการสร้างสัญญาณคืนกลับของคอมพิวเตอร์แบบ 1 ลูกสูบด้วยการปรับตั้งแบบที่ 4 โดยใช้การจับคู่สัญญาณบีบอัดที่ค่า K เท่ากับ 6

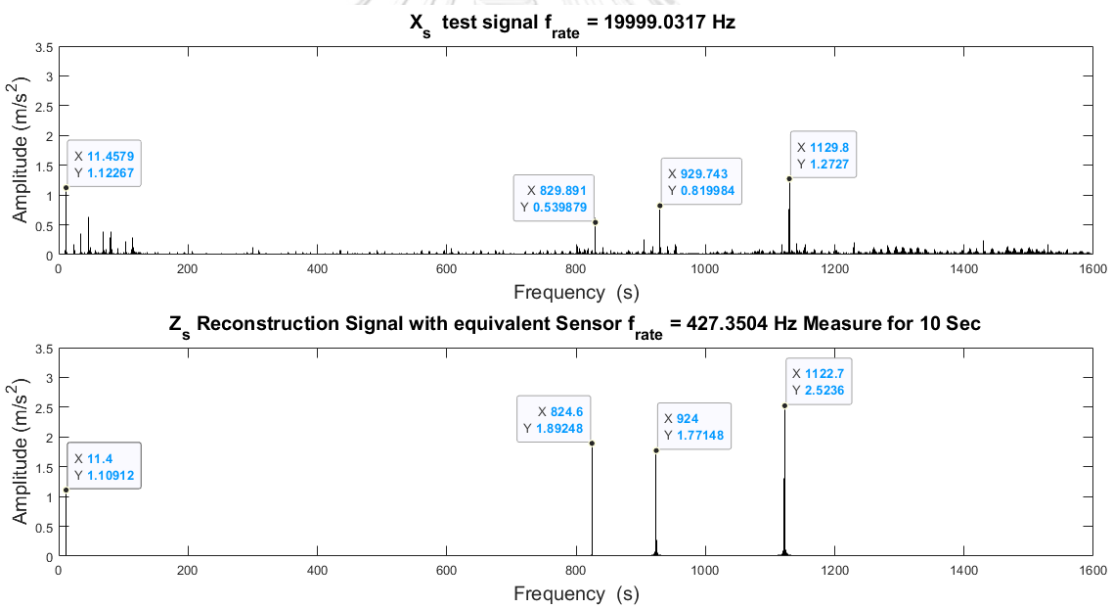


รูปที่ 5.52 สัญญาณตัวแทนและระดับ เปอร์เซ็นไทล์ที่เท่ากับค่า K ที่เลือก

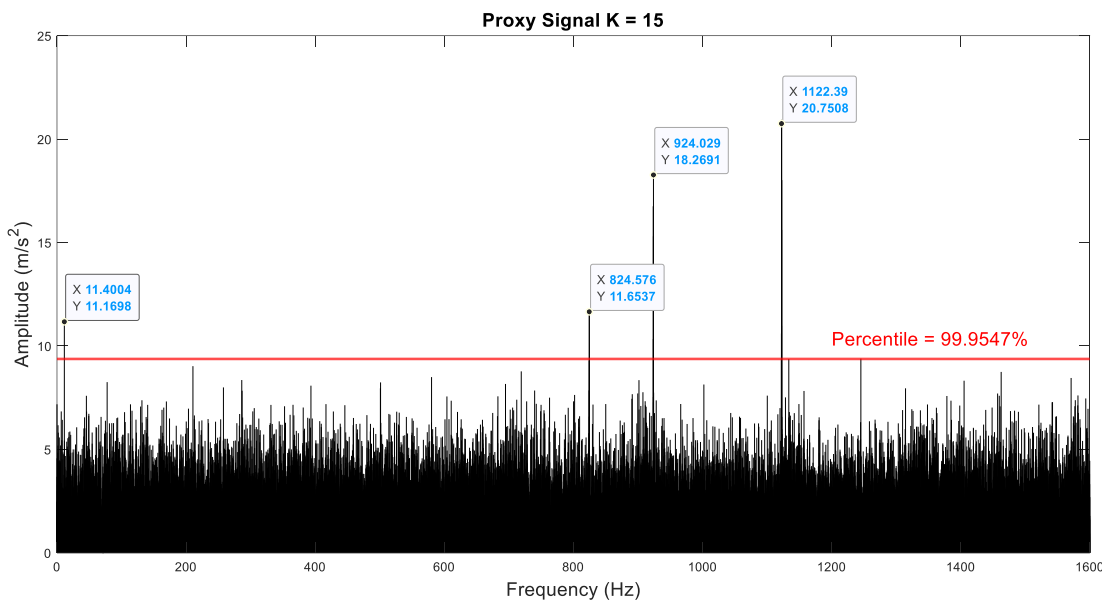
จากรูปที่ 5.43 จะเห็นว่า แถบของความถี่รบกวนในสัญญาณตัวแทนค่อนข้างจะสูง เมื่อเทียบกับ บริเวณสัญญาณหลักที่ความถี่ต่ำ ส่งผลให้ค่า K ที่เลือกมาจะไม่สามารถเห็นรายละเอียดของสัญญาณที่มากไปกว่ารายละเอียดที่ความถี่ต่ำได้ เนื่องจากลักษณะของสัญญาณต้นแบบเดิม ในบริเวณ 200 เฮิรตซ์ขึ้นไป ก็มีค่าน้อยและมีลักษณะเป็นบรอดแบนด์



รูปที่ 5.53 ผลการสร้างสัญญาณคืนกลับของคอมพิวเตอร์แบบ 2 ลูกสูบด้วยการปรับตั้งแบบที่ 8 โดยใช้การจับคู่สัญญาณบีบอัด ที่ค่า K เท่ากับ 15



รูปที่ 5.54 เปรียบเทียบผลการสร้างสัญญาณคืนกลับของคอมพิวเตอร์แบบ 2 ลูกสูบด้วยการปรับตั้งแบบที่ 8 โดยใช้การประมวลผลสัญญาณบีบอัด ที่ค่า K เท่ากับ 15



รูปที่ 5.55 สัญญาณตัวแทนและระดับ เปอร์เซ็นไทล์ที่เท่ากับค่า **K** ที่เลือก

จากรูปที่ 5.55 จะเห็นว่าสัญญาณตัวแทนเมื่อเทียบกับสัญญาณต้นแบบใน รูปที่ 5.45 จะเห็นว่ามี ความถี่หลักที่ถูกต้องเป็นส่วนใหญ่ทั้งในบริเวณความถี่สูงและความถี่ต่ำ

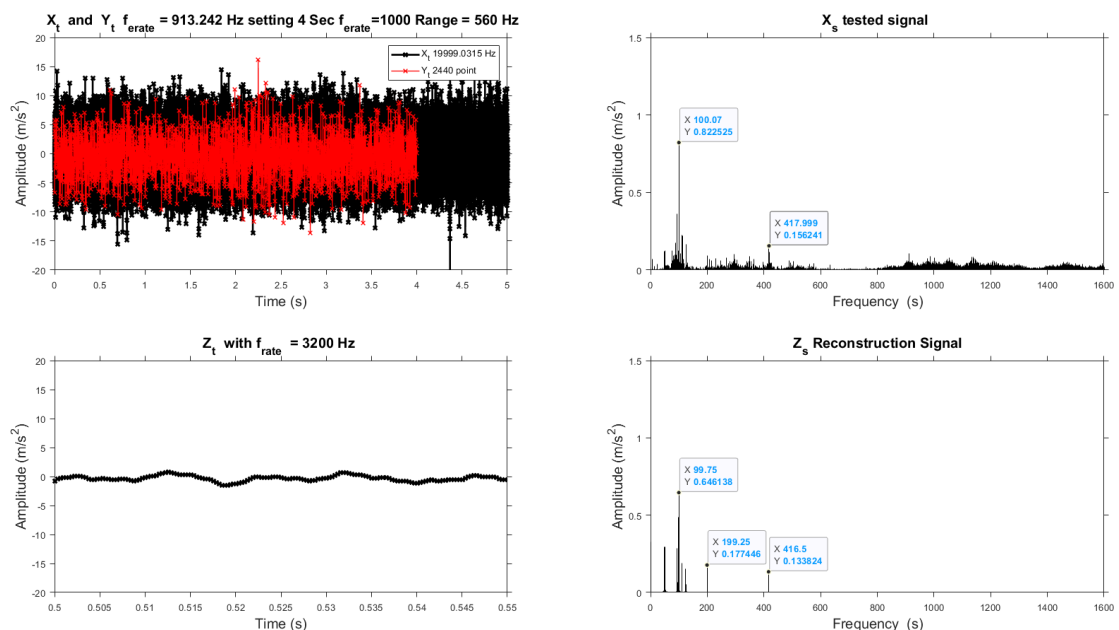
จากผลการทดสอบจะเห็นว่าสัญญาณคืนกลับ ที่ได้ นั้นจะมีค่าเฉพาะตำแหน่งความถี่หลักที่มีค่าแอมพลิจูดสูง เท่านั้น บริเวณอื่นๆของสัญญาณที่มีลักษณะเป็นแถบนั้นไม่สามารถที่จะสร้างคืนกลับได้เนื่องจาก หลักการของวิธีจับคู่สัญญาณบีบอัดและวิธีขึ้นสัญญาณเฉลี่ยนั้น ตำแหน่งของความถี่ที่สัญญาณจะถูกสร้างขึ้น จะสัมพันธ์กับค่า **K** ที่กำหนด อย่างไรก็ตามหากกำหนดค่า **K** ที่มากเกินไปกลับจะทำให้เกิดสัญญาณคืนกลับที่ ความถี่ไม่ตรงกับสัญญาณจริงเพิ่มขึ้นมาแทน ภาพรวมของสัญญาณคืนกลับจึงแตกต่างจากสัญญาณจริงไป มากโดยเฉพาะความถี่ที่มีขนาดต่ำ และเนื่องจากสัญญาณถูกสร้างขึ้นมาได้ไม่ครบถ้วนขนาดการสั่นสะเทือนใน โดเมนเวลาที่สร้างได้จึงมีขนาดเล็กและรูปร่างแตกต่างจากสัญญาณอ้างอิงมาก แต่ก็มีข้อดีคือใช้เวลาในการ คำนวณ น้อยกว่าวิธีการหาค่าที่เหมาะสมที่สุดแบบคอนเวกซ์มาก

จากผลการทดสอบโดยใช้วิธีการจับคู่สัญญาณบีบอัดพบว่าอัลกอริทึมชนิดนี้ถึงแม้จะมีความรวดเร็วใน การคำนวณหาค่าตอบซึ่งก็เป็นสาเหตุมาจากการพยายามเลือกคำนวณเฉพาะตำแหน่งความถี่ที่มีค่าแอมพลิจูด สูงสุด **K** อันดับแรกเท่านั้น ส่งผลให้สัญญาณมีความถูกต้องเฉพาะบางส่วนของสัญญาณและจะเหมาะกับ ลักษณะของสัญญาณที่มีลักษณะเป็นยอดสูงชัดเจนเมื่อเทียบกับสัญญาณโดยรวมเท่านั้น ซึ่งในความเป็นจริง สัญญาณการสั่นสะเทือนมีหลายลักษณะเช่น เป็นแถบกว้างๆที่มีแอมพลิจูดต่ำตลอดแนว ทำให้เมื่อสัญญาณ บางส่วนไม่ถูกคำนวณออกมาส่งผลให้ สัญญาณในโดเมนเวลาจะเพี้ยนไปจากความจริงค่อนข้างมาก อีก อุปสรรคหนึ่งของอัลกอริทึมชนิดนี้ก็คือจำเป็นต้องกำหนดค่า **K** อันดับขึ้นมาเองโดยสัญญาณในแต่ละแบบก็จะมี ค่า **K** ที่เหมาะสมไม่เท่ากันการใช้ค่า **K** น้อยไปก็ไม่สามารถคำนวณแอมพลิจูดที่สำคัญได้ครบทุกตำแหน่ง ถ้าใช้มากไปจะเป็นการเพิ่มสัญญาณรบกวนเข้ามาในคำตอบ โดยสรุปคืออัลกอริทึมนี้จะใช้ได้ดีสำหรับสัญญาณ การสั่นสะเทือนที่มีสัญญาณมีลักษณะเป็นแถบน้อย และแอมพลิจูดหลักมีขนาดสูงกว่าความถี่อื่นอย่างชัดเจน

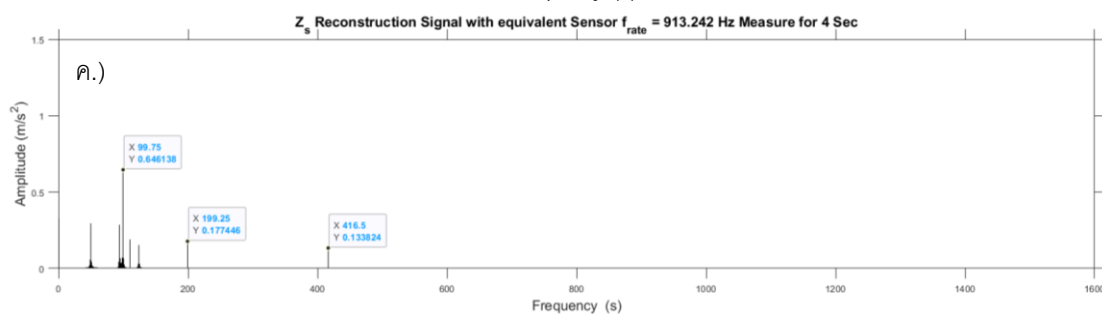
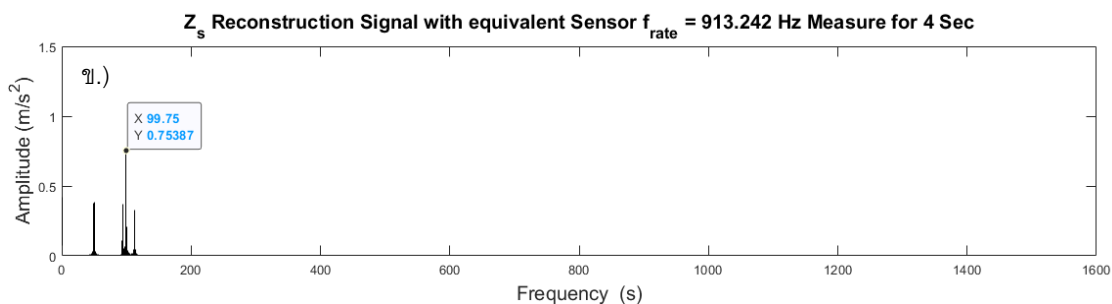
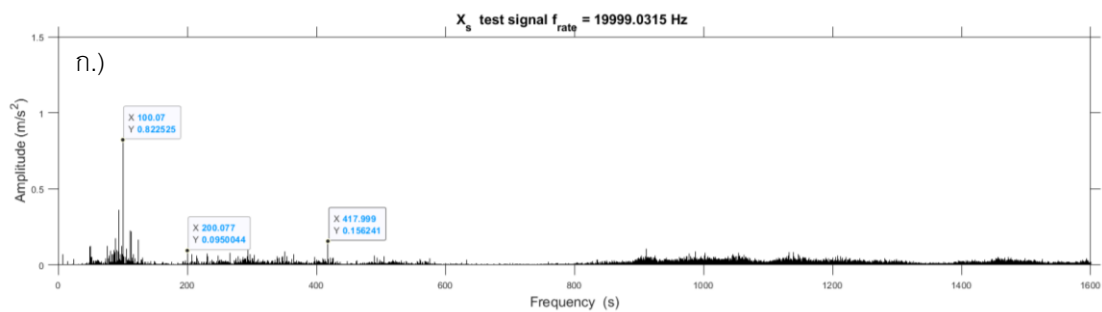
เพื่อให้การสร้างสัญญาณตัวแทนทำได้อย่างมีคุณภาพมากขึ้นโดย ความถี่หลักแยกออกจากสัญญาณรบกวนได้ชัดเจนทำให้หาตำแหน่งความถี่ที่สำคัญได้ ดังนั้นการนำวิธีการนี้ไปใช้จึงควรมีจุดประสงค์ที่เพียงพอการทราบความถี่หลักของการสั่นสะเทือนเท่านั้น ไม่ได้ต้องการทราบรายละเอียดและลักษณะของสัญญาณที่ต้องการตลอดช่วงการวัด

ทดสอบโดยการใช้การใช้อัลกอริทึมซ่อนสัญญาณเฉลี่ย

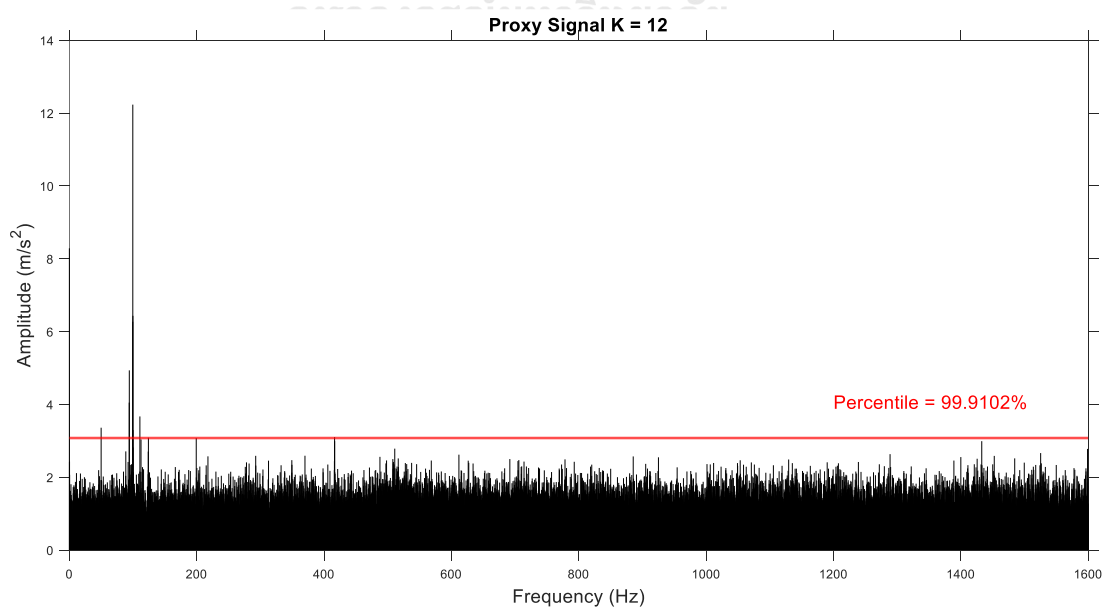
จากการทดสอบด้วยการใช้ การจับคู่สัญญาณบีบอัดจะพบว่าปัจจัยสำคัญที่จะทำให้สัญญาณคืนกลับถูกสร้างออกมาได้อย่างมีคุณภาพคือ ลักษณะของความถี่หลักในสัญญาณตัวแทน ถ้าหากความถี่หลักสามารถถูกแยกแยะออกจากแถบสัญญาณรบกวนได้อย่างชัดเจนก็ส่งผลให้สัญญาณคืนกลับถูกสร้างออกมาได้อย่างถูกต้องมากขึ้น อัลกอริทึมซ่อนสัญญาณตัวแทนเฉลี่ยจึงถูกคิดขึ้นมาเพื่อให้สัญญาณตัวแทนสามารถแยกแยะระหว่างความถี่หลักและสัญญาณรบกวนได้ดีขึ้นกว่า การจับคู่สัญญาณบีบอัด ซึ่งจะส่งผลให้สัญญาณคืนกลับมีความถูกต้องมากขึ้น



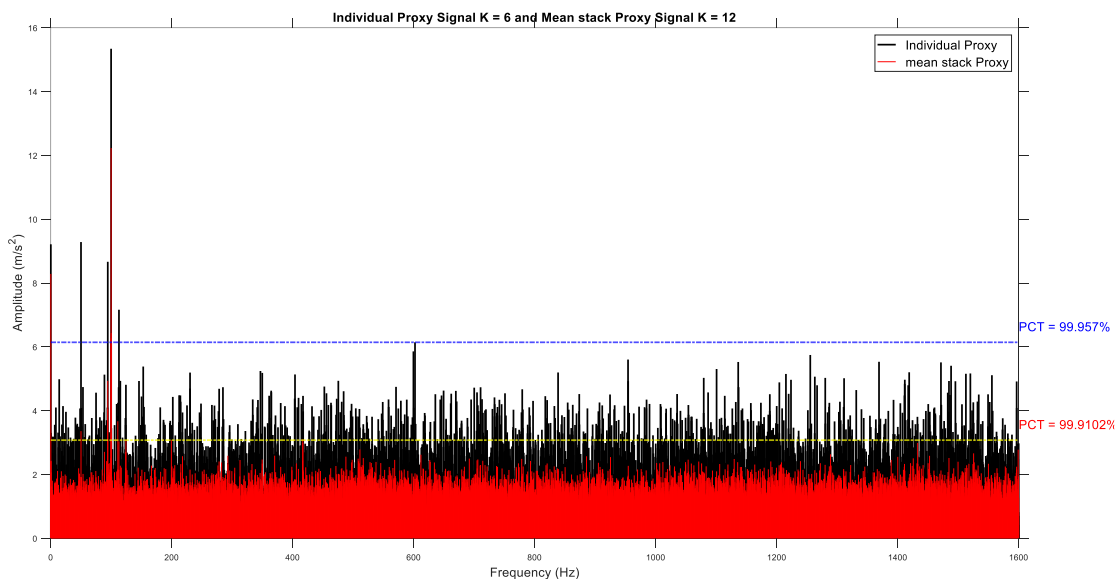
รูปที่ 5.56 ผลการสร้างสัญญาณคืนกลับของคอมพิวเตอร์แบบ 1 ถูกสุบด้วยการปรับตั้งแบบที่ 4 โดยใช้อัลกอริทึมซ่อนสัญญาณเฉลี่ย ที่ค่า K เท่ากับ 12



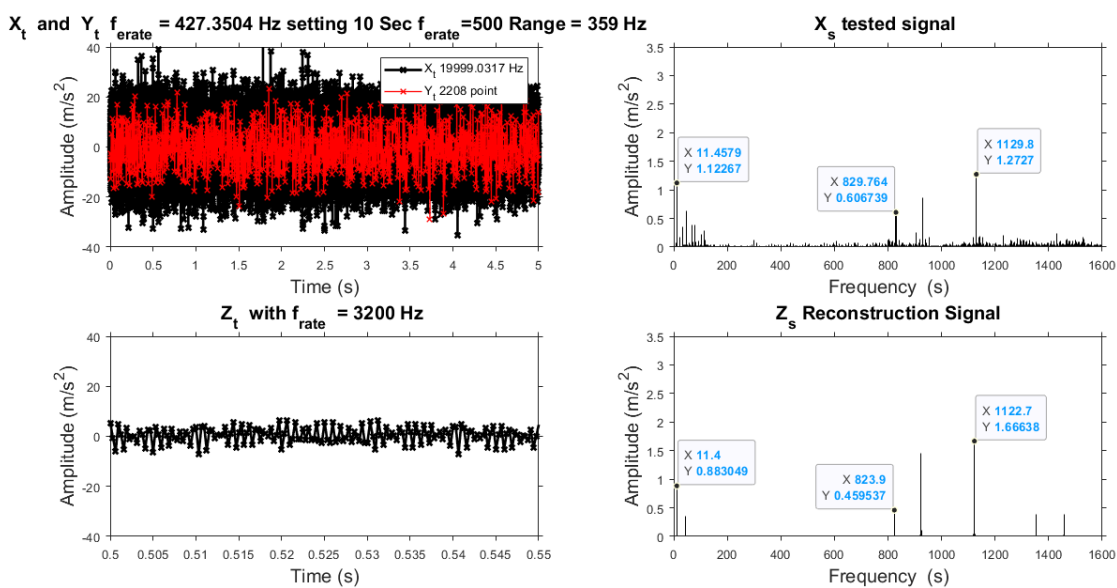
รูปที่ 5.57 สัญญาณคืนกลับของคอมเพรสเซอร์แบบ 1 ลูกสูบ จาก ก. สัญญาณต้นแบบ ข. การจับคู่สัญญาณบีบอัด ค. การซ้อนสัญญาณเฉลี่ย



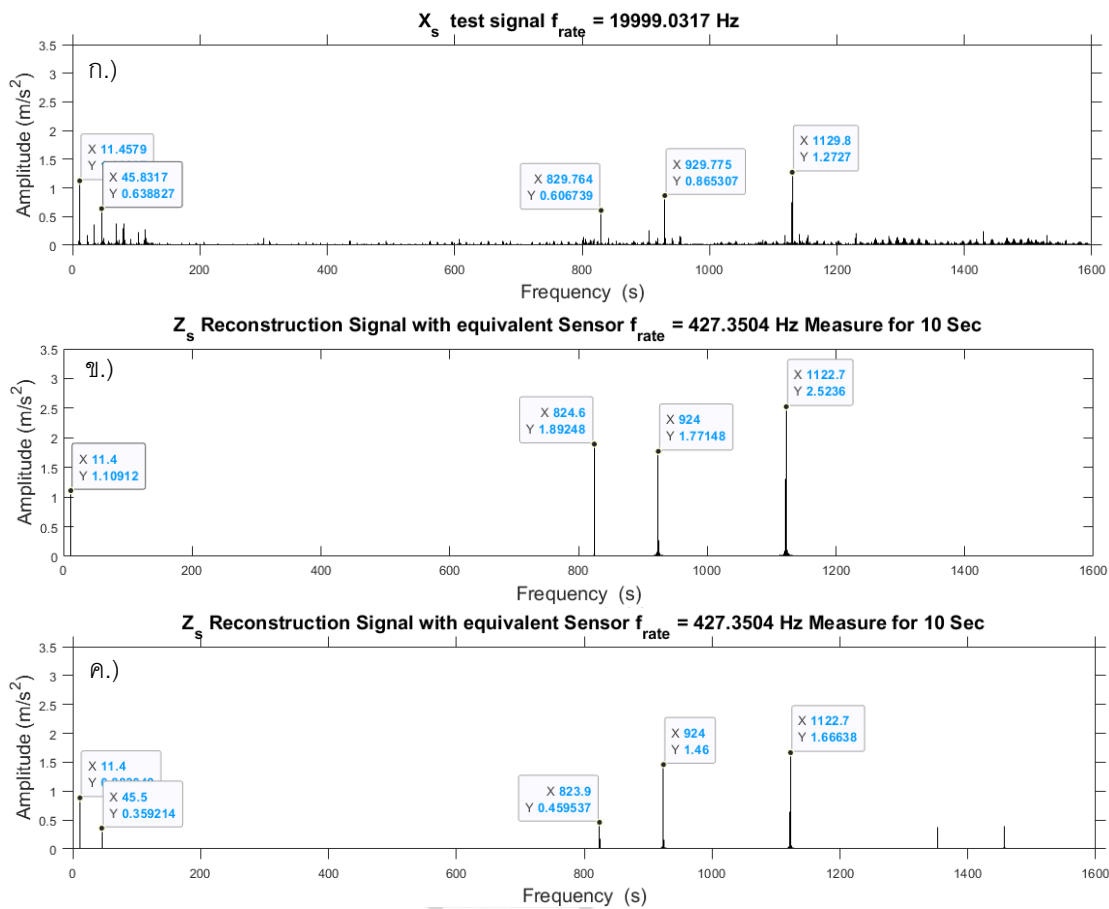
รูปที่ 5.58 สัญญาณตัวแทนและระดับเปอร์เซ็นต์ไทล์ที่เท่ากับค่า K ที่เลือก



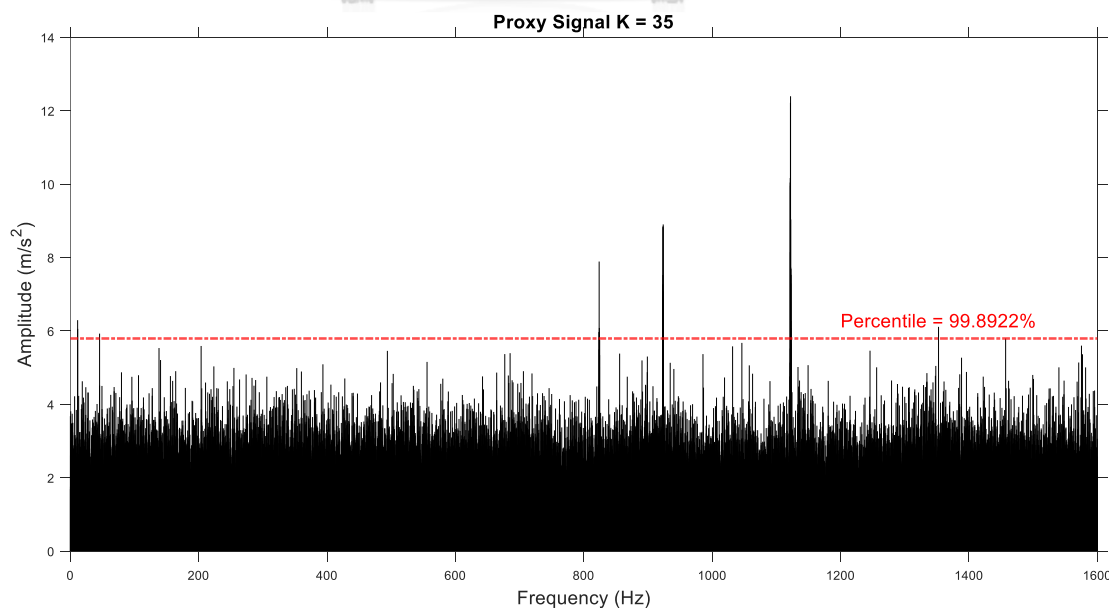
รูปที่ 5.59 เปรียบเทียบระดับการเลือกความถี่ที่เหมาะสมระหว่างการจับคู่สัญญาณบีบอัด และการซ้อนสัญญาณเฉลี่ย ของคอมเพรสเซอร์ 1 ลูกสูบ



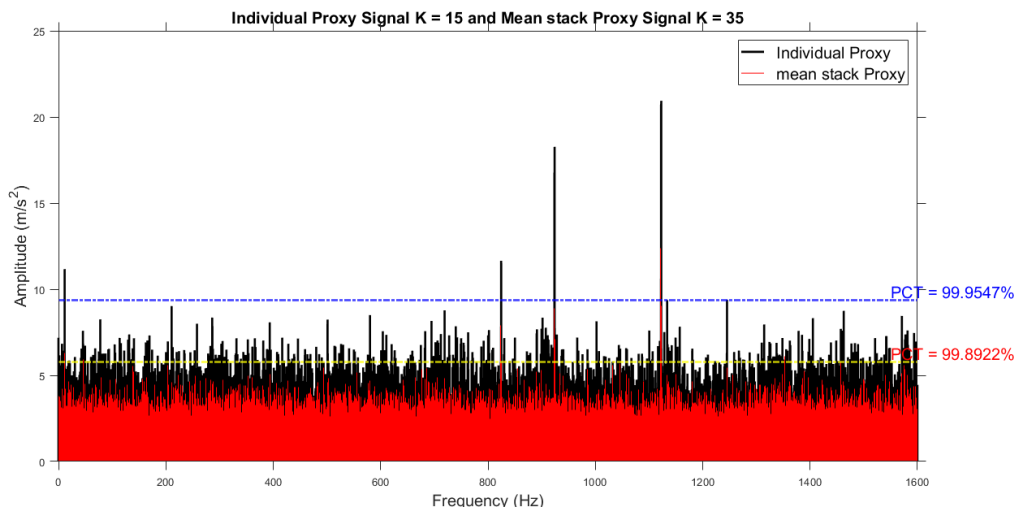
รูปที่ 5.60 ผลการสร้างสัญญาณคืนกลับของคอมเพรสเซอร์แบบ 2 ลูกสูบด้วยการปรับตั้งแบบที่ 8 โดยใช้การซ้อนสัญญาณเฉลี่ย ที่ค่า K เท่ากับ 35



รูปที่ 5.61 สัญญาณคืนกลับของคอมพิวเตอร์แบบ 2 ลูกสูบ จาก ก.สัญญาณต้นแบบ ข.การจับคู่สัญญาณบีบอัด ค.การขึ้นสัญญาณเฉลี่ย



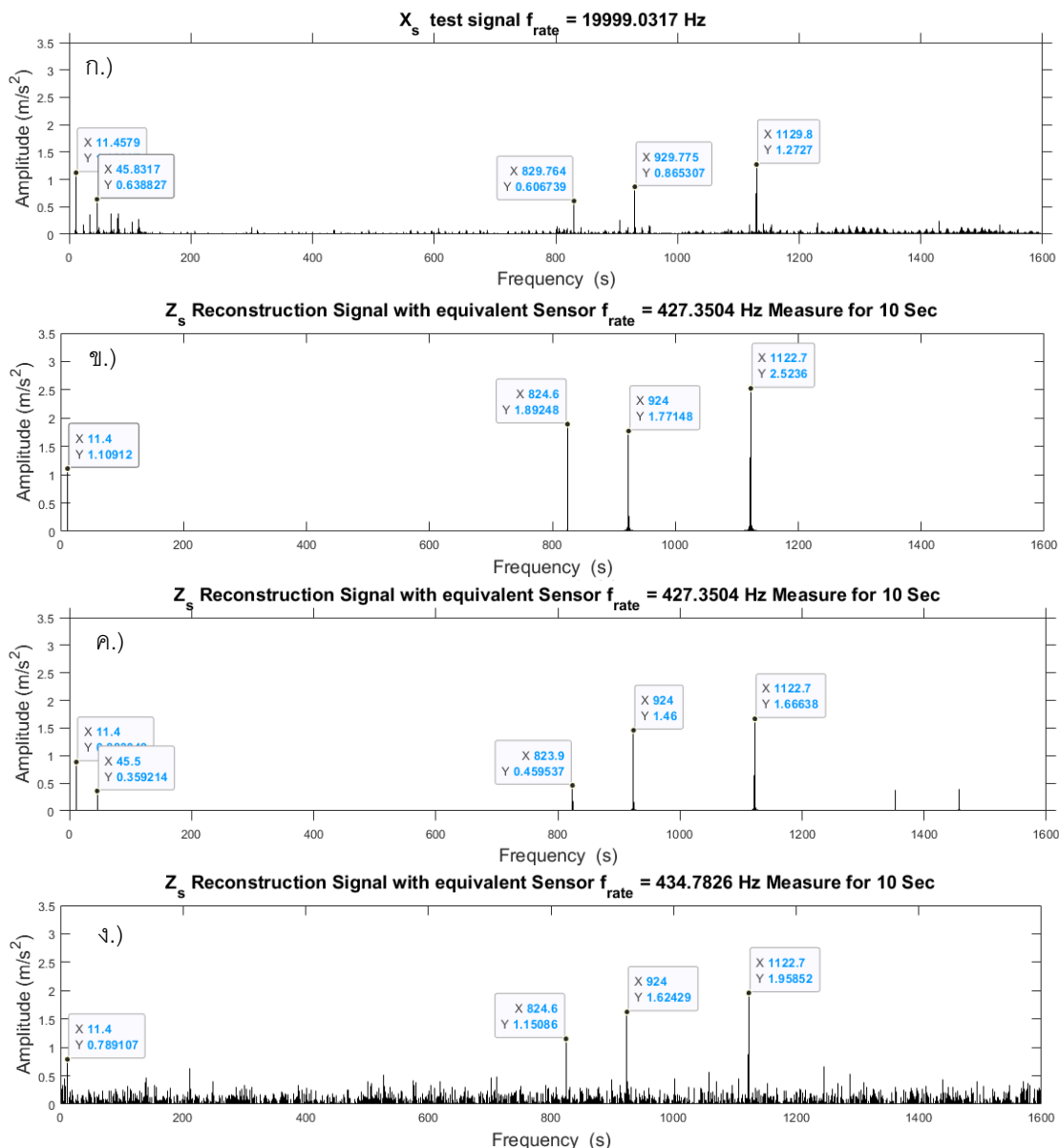
รูปที่ 5.62 สัญญาณตัวแทนและระดับเปอร์เซ็นต์ไทล์ที่เท่ากับค่า K ที่เลือก



รูปที่ 5.63 เปรียบเทียบระดับการเลือกความถี่ที่เหมาะสมระหว่างการจับคู่สัญญาณบีบอัด และการซ้อนสัญญาณเฉลี่ย ของคอมเพรสเซอร์ 2 ลูกสูบ

จากผลการทดสอบที่รูปที่ 5.48 และ 5.52 สัญญาณคืนกลับที่ถูกสร้างโดยอัลกอริทึมซ้อนสัญญาณเฉลี่ยมีลักษณะคล้ายคลึงกับ การใช้อัลกอริทึมจับคู่สัญญาณบีบอัด แต่เมื่อตรวจสอบสัญญาณตัวแทนของอัลกอริทึมทั้ง 2 แบบในรูปที่ 5.50 และ 5.54 จะพบว่าอัลกอริทึมซ้อนสัญญาณเฉลี่ยจะสามารถแยกแยะสัญญาณหลักได้ดีดีกว่าเล็กน้อย เช่นตำแหน่งที่ 200 และ 417 เฮิรตซ์ ของสัญญาณคอมเพรสเซอร์แบบ 1 ลูกสูบในรูปที่ 5.48 อัลกอริทึมซ้อนสัญญาณเฉลี่ยจะสามารถสร้างกลับคืนมาได้แต่อัลกอริทึมจับคู่สัญญาณบีบอัด ในรูปที่ 5.42 ไม่สามารถทำได้ ซึ่งสาเหตุที่การซ้อนสัญญาณเฉลี่ยสามารถทำได้ดีกว่าเนื่องจากสัญญาณที่นำมาใช้ในอัลกอริทึมมีจำนวนหลายชุดมากกว่าแบบสัญญาณบีบอัด

จากรูปที่ 5.55 เมื่อเปรียบเทียบคำตอบของสัญญาณ Z_5 ของทั้งสามอัลกอริทึม จะเห็นได้ว่า อัลกอริทึมการหาค่าเหมาะสมที่สุดแบบคอนเวกซ์จะได้คำตอบที่ดีที่สุดและใกล้เคียงกับสัญญาณต้นแบบมากที่สุดแต่ก็ใช้เวลามากที่สุดเช่นกัน



รูปที่ 5.64 สัญญาณคืนกลับของคอมเพรสเซอร์แบบ 2 ลูกสูบ จาก ก.สัญญาณต้นแบบ
 ข. การจับคู่สัญญาณบีบอัด ค. การซึ่ณสัญญาณเฉลี่ย ค. การหาค่าเหมาะสมที่สุดแบบคอนเวกซ์

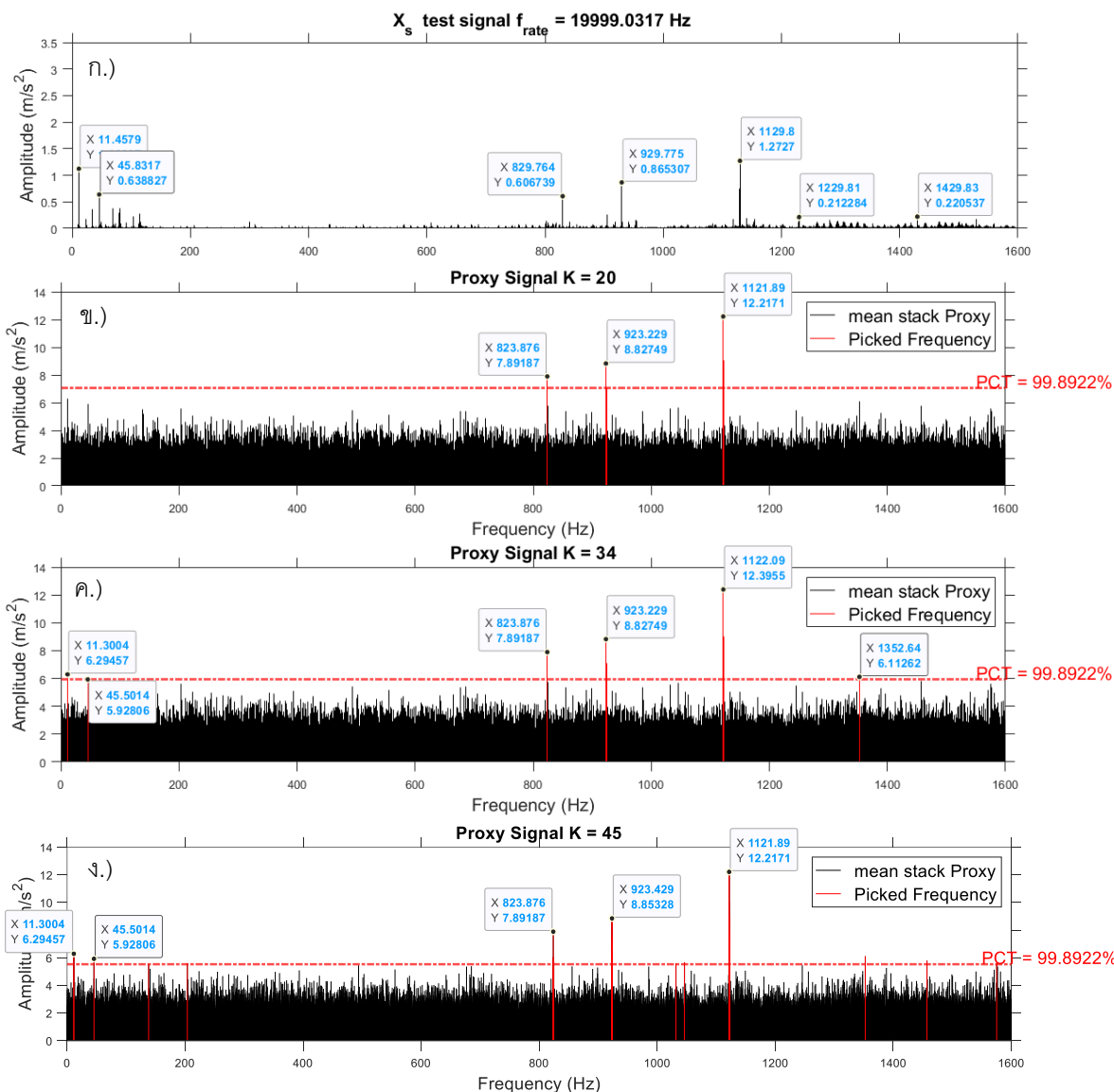
จากการทดสอบโดยการใช้อัลกอริทึมทั้งการจับคู่สัญญาณบีบอัดและการซึ่ณสัญญาณตัวแทนเฉลี่ยพบว่าทั้งสองอัลกอริทึมนี้สามารถทำการคำนวณได้รวดเร็วกว่าการใช้วิธีการหาค่าเหมาะสมที่สุดแบบคอนเวกซ์มากและจะยิ่งเห็นได้อย่างชัดเจนเมื่อโดยเฉพาะเมื่อสัญญาณ y_t ที่ใช้มีความยาวมากขึ้น แต่วิธีทั้งสองก็มีข้อเสียคือสัญญาณคืนกลับ Z_s ที่ได้จะมีค่าเฉพาะตำแหน่งความถี่ที่แอมพลิจูดสูงมากเท่านั้น เนื่องจากเงื่อนไขของอัลกอริทึมที่ถูกออกแบบมาให้เลือกคำนวณเฉพาะความถี่ที่มีแอมพลิจูดสูงสุดไม่ก็ความถี่เท่านั้นซึ่งเหมาะกับสัญญาณสั้นสะท้อนความถี่ที่ขนาดสัญญาณสูงเด่นออกมาชัดเจน และไม่มีส่วนของสัญญาณที่มีลักษณะเป็นเป็นแถบ ถึงแม้ว่าจะ เป็นแถบของความถี่ที่แอมพลิจูดไม่มาก หรือเป็นไซด์แบนขนาดเล็ก ดังเช่นบางส่วน

ของสัญญาณของคอมเพรชเซอร์ 1 ลูกสูบก็ตาม วิธีการนี้ก็ไม่สามารถหาคำตอบที่ถูกต้องออกมาได้เนื่องจากลักษณะของสัญญาณแบบนี้จะง่ายต่อการถูกรบกวนด้วยแถบสัญญาณรบกวนในสัญญาณตัวแทน และเนื่องจากสัญญาณ Z_S เป็นเพียงแค่บางส่วนของสัญญาณต้นแบบเท่านั้นทำให้สัญญาณ Z_T ในโดเมนเวลาเพี้ยนไปจากเดิมมากจนไม่สามารถจะนำไปใช้วิเคราะห์เพิ่มเติมได้ อีกข้อเสียหนึ่งคือผู้ใช้อัลกอริทึมจำเป็นต้องกำหนดค่า K ด้วยตัวเองโดยที่ไม่รู้ว่าค่าที่เหมาะสมจริงๆ เป็นเท่าไรหรือนอกจากคาดการณ์จากสัญญาณตัวแทนที่เห็น หรือจากข้อมูลที่เคยเก็บมาก่อนหน้านี้เท่านั้น ซึ่งพอจะสรุปได้ว่าการใช้อัลกอริทึมทั้ง 2 แบบ

วิเคราะห์ผลของการเลือก K สัญญาณตัวแทน

สัญญาณตัวแทนเป็นพารามิเตอร์ที่สำคัญสำหรับอัลกอริทึมจับคู่สัญญาณบีบอัดและอัลกอริทึมซ้อนสัญญาณเฉลี่ยเนื่องจากเป็นพารามิเตอร์ที่ใช้สำหรับการเลือกความถี่ที่จะถูกนำไปคำนวณเป็นคำตอบของสัญญาณ Z_S โดยที่จะเลือกจำนวนความถี่ทั้งหมด K ความถี่โดยที่ค่า K ผู้ใช้จะเป็นผู้กำหนด โดยในหัวข้อนี้จะยกตัวอย่างสัญญาณคอมเพรชเซอร์ แบบ 2 ลูกสูบเป็นตัวอย่างในกรณีอธิบาย





รูปที่ 5.65 เปรียบเทียบการเลือกใช้ค่า **K** ต่างๆ ในสัญญาณตัวแทน ก.) สัญญาณต้นแบบ ข.) **K = 20** ค.)

$$K = 34 \text{ ง.) } K = 45$$

จากรูปที่ 5.65 จะเห็นว่าสัญญาณต้นแบบในรูปที่ 5.65(ก.) มีตำแหน่งความถี่ที่เป็นตำแหน่งหลักประมาณ 5 ตำแหน่งที่ชัดเจน คือ 11, 45, 823, 923 และ 1121.89 เฮิรตซ์ ซึ่งการเลือก **K** ที่น้อยเกินไปในรูปที่ 5.65(ข.) จะส่งผลให้สัญญาณคืนกลับที่สร้างขึ้น จำนวนยอดของสัญญาณที่ถูกต้องไม่ครบถ้วน แต่ถ้าเลือก **K** ที่มากเกินไปในรูปที่ 5.65(ง.) ก็จะทำให้ยอดของสัญญาณบางส่วนที่เป็นสัญญาณรบกวนเข้ามาในคำตอบเช่นกัน จะสังเกตเห็นว่าในการวัดสัญญาณ y_t ชุดนี้การใช้ **K = 34** จึงเป็นตัวเลือกที่มีจำนวนยอดใกล้เคียงกับสัญญาณต้นแบบมากที่สุด การเลือก **K** ที่เหมาะสมเป็นค่าที่ขึ้นอยู่กับหลายปัจจัย เช่น ความคมชัดของสัญญาณตัวแทน ความซับซ้อนของสัญญาณการสั่นสะเทือน ส่งผลให้ค่า **K** เป็น tuning parameters ที่จะต้องทำเป็นรายการกรณีไปในแต่ละสัญญาณ

5.3 สรุป

สำหรับผลสรุปของการทดสอบสามารถแบ่งได้เป็นหัวข้อดังนี้

1. หากสามารถสุ่มสัญญาณโดยทราบค่าของจุดเวลา และขนาดของสัญญาณที่แน่นอน หลักการคอมเพรสชันซึ่งสามารถนำมาใช้ได้ อย่างดีกับการวัดการสั่นสะเทือนที่สัญญาณมีความสปีร์สูงซึ่งคำตอบที่ได้มีความใกล้เคียงกับสัญญาณต้นแบบมาก
2. คุณภาพของสัญญาณคืนกลับ Z_S จากคอมเพรสชันซึ่งขึ้นอยู่กับคุณภาพของสัญญาณสุ่ม y_t เนื่องจากถ้า y_t มีความคลาดเคลื่อนจาก x_t ไปมากสัญญาณที่ได้ หรือ Z_S ก็จะยิ่งคลาดเคลื่อนไปมากเช่นกันดังนั้นการวัดสัญญาณจริงจึงต้องพิจารณาทั้ง อัตราการสุ่มจุด ความไวของเซ็นเซอร์, ความละเอียด และ ความแม่นยำของเซ็นเซอร์ที่มีผลเช่นกัน
3. จากการทดสอบโดยใช้ระบบวัดแบบมาตรฐานวัดด้วยความถี่สุ่มสูงและสุ่มหยิบสัญญาณออก เพื่อให้ได้สัญญาณที่แทนการวัดแบบสุ่มที่มีก็ให้ผลสัญญาณคืนกลับ Z_S ที่ดี แต่สำหรับระบบการวัดแบบสุ่มคาบโดยใช้ระบบการวัดราคาถูก คุณภาพสัญญาณที่ได้ก็ยังไม่ดีพอที่จะนำไปใช้งานได้ ซึ่งอาจปรับปรุงคุณภาพของสัญญาณให้ดีขึ้นได้โดย เพิ่มความยาวของการวัดสัญญาณให้มากขึ้น
4. การใช้สัญญาณ y_t ที่ยาวมากขึ้น ให้คุณภาพของสัญญาณที่ดีขึ้นอย่างเห็นได้ชัด แต่มีข้อสังเกตว่าคอมเพรสชันซึ่งจะสร้างสัญญาณ Z_S ที่ต้นแบบมีลักษณะเป็นแถบความถี่ หรือเป็นบรอดแบนด์มากๆ ได้ไม่ได้นักเมื่อเทียบกับสัญญาณที่มีความสปีร์สูงหรือมีลักษณะเป็นยอดสูง
6. การใช้สัญญาณ y_t ที่ยาวส่งผลให้คุณภาพของสัญญาณ Z_S สูงขึ้น แต่ก็มีขีดจำกัดอยู่ที่การคำนวณต้องใช้หน่วยความจำมากใช้เวลาคำนวณมากขึ้นอย่างมาก เมื่อใช้อัลกอริทึมการหาค่าเหมาะสมที่สุดแบบคอนเวกซ์
7. หากไม่ได้ใช้อัลกอริทึมการหาค่าเหมาะสมที่สุดแบบคอนเวกซ์ ก็อาจใช้อัลกอริทึมทางเลือกอื่นๆ เช่น อัลกอริทึมการจับคู่สัญญาณบีบอัด อัลกอริทึมซึ้นสัญญาณเฉลี่ย ในการหาค่าของสัญญาณ Z_S ซึ่งไม่มีอุปสรรคในเรื่องของ ประสิทธิภาพของคอมพิวเตอร์ แต่อย่างไรก็ตาม การใช้อัลกอริทึมการจับคู่สัญญาณบีบอัดและอัลกอริทึมซึ้นสัญญาณเฉลี่ย จะให้ผลสัญญาณ Z_S ที่มีลักษณะเป็นยอดของแอมพลิจูดที่เป็นความถี่หลักไม่กี่ตำแหน่งเท่านั้น ไม่สามารถจะสร้างสัญญาณที่มีลักษณะใกล้เคียงกับสัญญาณต้นแบบได้ถูกต้องเท่ากับอัลกอริทึมการหาค่าเหมาะสมที่สุดแบบคอนเวกซ์ และจำเป็นจะต้อง จูนค่า K ให้เหมาะสมด้วยการตรวจสอบลักษณะของสัญญาณตัวแทนที่ได้ก่อนเสมอ
9. การใช้คอมเพรสชันซึ่งสามารถช่วยเพิ่มขีดจำกัดของเซ็นเซอร์ที่มีความถี่ในการสุ่มสัญญาณต่ำให้สามารถวัดสัญญาณที่มีพิสัยความถี่กว้างกว่าอัตราในควิสได้ดังการทดสอบที่ 5.2.2 ที่เลือกใช้ความถี่สุ่มเทียบเท่าเพียง 500 เฮิร์ตซ์แต่สามารถวัดสัญญาณที่มีความกว้างถึง 1600 เฮิร์ตซ์ได้ดีพอสมควร แต่การใช้คอมเพรสชันซึ่งกับเซ็นเซอร์ที่มีความถี่สุ่มต่ำ จะต้องแลกมาด้วยการใช้เงื่อนไขการวัดแบบสุ่มคาบที่มีความซับซ้อนในการปรับตั้งการใช้หน่วยความจำที่มากขึ้น และระยะเวลาในการประมวลผลที่นานขึ้นเช่นกัน

บทที่ 6

บทสรุป

ในบทนี้จะเป็นการสรุปเนื้อหาสำคัญในงานวิจัยนี้ ซึ่งจะเป็นเนื้อหาที่สำคัญในแต่ละบทรวมถึงข้อเสนอแนะเพื่อเป็นแนวทางในการศึกษาการประยุกต์คอมพิวเตอร์เซินซิงเข้ากับระบบการวัดการสั่นสะเทือนต่อไป

6.1 สารสำคัญของวิทยานิพนธ์

บทที่ 1 จะแนะนำถึงทฤษฎีในควิสต์และข้อจำกัดของความถี่สูงซึ่งส่งผลต่อการใช้ระบบวัดการสั่นสะเทือนในปัจจุบัน พร้อมทั้งอธิบายการนำคอมพิวเตอร์เซินซิงไปใช้ในอุตสาหกรรมต่างๆ และใช้กับการวัดการสั่นสะเทือน ปัจจุบันซึ่งนิยมใช้คอมพิวเตอร์เซินซิงกับสัญญาณที่ถูกวัดด้วยระบบการวัดคุณภาพสูงเรียบร้อยแล้ว และความพยายามของงานวิจัยนี้จะใช้ระบบการวัดคุณภาพต่ำเพื่อวัดสัญญาณให้พิสัยความถี่สูงกว่าอัตราในควิสต์

บทที่ 2 จะเป็นการแนะนำวิธีการแก้ปัญหาการเชิงเส้นทั้งแบบระบบสมการไม่ชัดเจนและแบบชัดเจนเกินไปโดยระบบสมการไม่ชัดเจนนั้นมีจำนวนของคำตอบที่ถูกต้องเป็นอนันต์ แต่ก็อาจหาคำตอบได้โดยกำหนดเงื่อนไขการแก้สมการเพิ่มเติม เช่น ต้องการให้คำตอบมีความเป็นสปีร์สมากที่สุด เป็นต้น ซึ่งแนวคิดนี้ได้นำไปประยุกต์ใช้กับสัญญาณการสั่นสะเทือนที่มีลักษณะเป็นสปีร์สในโดเมนความถี่ สำหรับการแก้วิธีการแก้ปัญหาระบบสมการให้สอดคล้องกับการวัดการสั่นสะเทือน ได้ถูกอธิบายในบทที่ 2 ซึ่งครอบคลุมถึงการบีบอัดสัญญาณ และการเปลี่ยนฐานหลักซึ่งวิธีการคำนวณก็จะสอดคล้องกับการแก้ปัญหาระบบสมการเชิงเส้นเช่นกัน โดยพื้นฐานแล้วคอมพิวเตอร์เซินซิง ก็เปรียบเสมือนกับการคำนวณย้อนกลับของการบีบอัดสัญญาณนั่นเอง

โดยการใช้คอมพิวเตอร์เซินซิงในงานวิจัยนี้จะมุ่งเน้นไปที่การแก้ปัญหาระบบสมการเชิงเส้นทั้งแบบชัดเจนและไม่ชัดเจนด้วยวิธีต่างๆ กันหลายวิธีทั้งการแก้ระบบสมการโดยตรง ได้แก่การหาค่าเหมาะสมที่สุดแบบคอนเวกซ์) การปรับแต่งระบบสมการให้มีจำนวนพจน์น้อยลงเพื่อให้ง่ายแก่การประมาณค่าคำตอบ ได้แก่การจับคู่สัญญาณบีบอัดและการซ่อนสัญญาณเฉลี่ย โดยมีเป้าหมายที่จะศึกษาการใช้คอมพิวเตอร์เซินซิงจะสามารถช่วยให้ระบบการวัดที่มีราคาไม่แพงและมีความถี่สูงต่ำ สามารถวัดสัญญาณที่มีความถี่สูงกว่า ครึ่งหนึ่งของความถี่สูงของเซ็นเซอร์ได้มากน้อยเพียงใด

บทที่ 3 จะกล่าวถึงกระบวนการที่จะนำคอมพิวเตอร์เซินซิงไปใช้กับการวัดการสั่นสะเทือนในแต่ละการทดสอบที่แตกต่างกันให้ได้อย่างเหมาะสม เช่น วิธีการจำลองสัญญาณสุ่มคาบ y_t สำหรับการจำลองในคอมพิวเตอร์ การสร้างสัญญาณสุ่มคาบ y_t จากสัญญาณซึ่งได้จาก x_t การวัดจริงแบบปกติ และการวัดสัญญาณสุ่ม y_t โดยตรงจากเครื่องจักร หลังจากนั้นก็จะเป็นการคำนวณการจับคู่เวลา และสร้างแมทริกซ์ฐานหลักเริ่มต้น

การสร้างสัญญาณสุ่มคาบแบบต่างๆ ที่ไม่เหมือนกันจะถูกใช้ในการทดสอบความเป็นไปได้ในการใช้คอมพิวเตอร์เซินซิงในกรณีต่างๆ โดยเริ่มจากวิธีการทดสอบที่ไม่พิจารณาผลกระทบจากปัจจัยภายนอก คือการจำลองในคอมพิวเตอร์ ไปจนถึงการทดสอบโดยการวัดจริงแบบสุ่มคาบด้วยระบบการวัดราคาถูก

ในบทที่ 4 จะเป็นการอธิบายรายละเอียดของวิธีการทดสอบแต่ละแบบที่มีความยากง่ายและความซับซ้อนของสัญญาณเรียงจากน้อยที่สุดไปหามากที่สุด หรือตั้งแต่การจำลองทุกอย่างในคอมพิวเตอร์ไปจนถึงการวัดจริงและวิธีการประเมินสัญญาณแบบต่างๆ

สำหรับวิธีการประเมินสัญญาณในแต่ละการทดสอบก็จะมีวิธีการประเมินที่ไม่เหมือนกันเนื่องจากข้อจำกัดของการทดสอบในแต่ละแบบที่ไม่เหมือนกันเช่น การประเมินโดยใช้การนับและจัดหมวดหมู่ของยอดของความถี่ เหมาะจะใช้กับการทดสอบที่มีความแน่นอนของตำแหน่งความถี่ ได้แก่การจำลองสัญญาณในคอมพิวเตอร์ หรือการทดสอบด้วยการใช้ระบบการวัดมาตรฐานกับสัญญาณที่ไม่ซับซ้อน แต่จะไม่เหมาะกับการประเมินสัญญาณจริงที่มีความถี่มีความคลาดเคลื่อนจากสัญญาณรบกวนเนื่องจากการวัดด้วยความถี่สุ่มต่ำ ซึ่งจะเหมาะกับการประเมินด้วยการสังเกตสัญญาณโดยตัวผู้วิจัยเองมากกว่า

บทที่ 5 แสดงถึงผลการทดสอบในการทดสอบแบบต่างๆ ตั้งแต่ การจำลองในคอมพิวเตอร์ การวัดโดยใช้ระบบการวัดมาตรฐาน การวัดโดยใช้ระบบสุ่มคาบโดยตรง ไปจนถึงการทดสอบใช้อัลกอริทึมแบบต่างๆ ในการแก้ปัญหาคอมเพรชชันซิง โดยภาพรวมแล้วพบว่า คอมเพรชชันซิงสามารถนำไปใช้ในการวัดการสั่นสะเทือนหากสามารถวัดสัญญาณที่มีคุณภาพเพียงพอถึงแม้ว่าจะเป็นระบบการวัดที่มีอัตราการสุ่มจุดต่ำกว่าความถี่สูงสุดที่ต้องการวัดจน ซึ่งเป็นไปตาม ทฤษฎีในควิสต์ แต่ต้องคำนึงถึงความรุนแรงของการสั่นสะเทือนที่จะทำการวัด, ลักษณะของสัญญาณที่คาดว่าจะเจอ, ความสามารถของคอมพิวเตอร์ที่ใช้ในการคำนวณสัญญาณคืนกลับทั้งในด้านของ ขนาดหน่วยความจำและความเร็วในการประมวลผล และสุดท้ายคือลักษณะของการนำสัญญาณคืนกลับไปใช้งาน ก็มีผลในงานที่จะเลือกใช้อัลกอริทึมในสำหรับการทำคอมเพรชชันซิงเช่นกัน

6.2 ข้อเสนอแนะ

ผู้วิจัยมีข้อเสนอแนะเพื่อให้นักวิจัยที่จะทำต่อมีผลลัพธ์ที่ดีขึ้นดังนี้

1. อัลกอริทึมสำหรับทำคอมเพรชชันซิงมีหลากหลายอัลกอริทึมมาก นอกเหนือไปจากงานวิจัยนี้ ถ้าหากมีการทดสอบโดยการใช้อัลกอริทึมแบบอื่นๆเพิ่มเติมก็อาจจะทำให้พบวิธีการที่มีประสิทธิภาพดีกว่าเดิม
2. ในงานวิจัยนี้เซ็นเซอร์ที่มีราคาไม่แพงที่มีขายทั่วไปตามท้องตลาดซึ่งปกติแล้วจะมีวงจรกรองความถี่ต่ำผ่านติดตัวมาตั้งแต่เริ่มต้นทำให้สัญญาณที่มีความถี่สูงกว่า ความถี่คัทออฟถูกตัดไป ถ้าหาสามารถหาเซ็นเซอร์ที่ไม่มีวงจรกรองความถี่ต่ำผ่านหรือสามารถปรับความถี่คัทออฟให้สูงตามที่ต้องการได้ ก็จะสามารถเข้าใจขีดจำกัดเพิ่มเติมของคอมเพรชชันซิงในการวัดการสั่นสะเทือนได้ดียิ่งขึ้น
3. ระบบการวัดและเซ็นเซอร์สำหรับการวัดการสั่นสะเทือนโดยปกติแล้วที่มีขายอยู่ทั่วไปปกติจะถูกออกแบบมาให้ใช้สำหรับการวัดแบบคาบการสุ่มคงที่ตลอด การจะทำให้ระบบการวัดแบบปกติทำการวัดแบบคาบการสุ่มไม่แน่นอนนั้นส่วนใหญ่แล้วระบบที่มีขายทั่วไปมักจะไม่มียืดหยุ่นมากพอที่จะปรับตั้งให้วัดในรูปแบบดังกล่าวได้ การจะนำคอมเพรชชันซิงไปใช้งานจริงจึงยังต้องออกแบบและสร้างระบบ

การวัดแบบสุ่มคาบขึ้นมาใหม่ ถ้าหากผู้วิจัยมีเข้าใจ ด้านระบบสมองกลฝังตัวและไม่โครคอนโทรลเลอร์เพิ่มเติม จะช่วยให้สามารถสร้างระบบการวัดแบบสุ่มคาบได้ดียิ่งขึ้น

4. สำหรับแหล่งกำเนิดสัญญาณการสั่นสะเทือน ในงานวิจัยนี้ยังมีความหลากหลายของลักษณะของสัญญาณค่อนข้างน้อยถ้าหากสามารถหาแหล่งกำเนิดที่สามารถสร้างการสั่นสะเทือนที่ให้ลักษณะของสัญญาณที่หลากหลายมากกว่านี้ ก็จะสามารถเข้าใจขีดจำกัดเพิ่มเติมของคอมพิวเตอร์ซึ่งในการวัดการสั่นสะเทือนได้ดียิ่งขึ้น
5. สำหรับการสร้างแมทริกซ์ฐานหลักเริ่มต้นที่ใช้แมทริกซ์การแปลงโคไซน์ ในงานวิจัยนี้ยังใช้วิธีการสร้างแมทริกซ์แบบง่ายโดยอาศัยฟังก์ชันในโปรแกรม MATLAB ถ้าหากสามารถสร้างแมทริกซ์ฐานหลักเริ่มต้นโดยการเริ่มจาก ฟังก์ชันพื้นฐานทางคณิตศาสตร์ตั้งแต่เริ่มต้น มีความเป็นไปได้ที่จะสามารถลดภาระหรือลดพื้นที่หน่วยความจำที่ต้องใช้ลงได้สำหรับการสร้างแมทริกซ์ฐานหลักเริ่มต้น



บรรณานุกรม

- [1] H. Nyquist, "Certain topics in telegraph transmission theory," *Proceedings of the IEEE*, vol. 90, no. 2, pp. 280-305, 2002, doi: 10.1109/5.989875.
- [2] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289-1306, 2006, doi: 10.1109/TIT.2006.871582.
- [3] C. G. Graff and E. Y. Sidky, "Compressive sensing in medical imaging," (in eng), *Appl Opt*, vol. 54, no. 8, pp. C23-44, Mar 10 2015, doi: 10.1364/ao.54.000c23.
- [4] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust Face Recognition via Sparse Representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210-227, 2009, doi: 10.1109/TPAMI.2008.79.
- [5] G. Tang, W. Hou, H. Wang, L. Ganggang, and J. Ma, "Compressive Sensing of Roller Bearing Faults via Harmonic Detection from Under-Sampled Vibration Signals," *Sensors (Basel, Switzerland)*, vol. 15, pp. 25648-25662, 10/16 2015, doi: 10.3390/s151025648.
- [6] M. L. D. Wong, M. Zhang, and A. K. Nandi, "Effects of compressed sensing on classification of bearing faults with entropic features," in *2015 23rd European Signal Processing Conference (EUSIPCO)*, 31 Aug.-4 Sept. 2015 2015, pp. 2256-2260, doi: 10.1109/EUSIPCO.2015.7362786.
- [7] H. O. A. Ahmed, M. L. D. Wong, and A. K. Nandi, "Compressive sensing strategy for classification of bearing faults," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5-9 March 2017 2017, pp. 2182-2186, doi: 10.1109/ICASSP.2017.7952543.
- [8] V. Ganesan, T. Das, N. Rahnavard, and J. Kauffman, "Vibration-based monitoring and diagnostics using compressive sensing," *Journal of Sound and Vibration*, vol. 394, 02/01 2017, doi: 10.1016/j.jsv.2017.02.002.
- [9] X. Zhang, N. Hu, C. Zhe, and Z. Hua, "A de-noising method for vibration signals based on compressed sensing," in *2014 Prognostics and System Health Management Conference (PHM-2014 Hunan)*, 24-27 Aug. 2014 2014, pp. 348-352, doi: 10.1109/PHM.2014.6988192.
- [10] S.-X. C. Yi-Qing, "Compressive sensing for vibration signals in high-speed rail monitoring," in *9th European Workshop on Structural Health Monitoring (EWSHM 2018)*, Manchester, UK, 10-13 July 2018 2018.

- [11] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301-321, 2009/05/01/ 2009, doi: <https://doi.org/10.1016/j.acha.2008.07.002>.
- [12] X. Chaoang, T. Hesheng, and R. Yan, "Compressed sensing reconstruction for axial piston pump bearing vibration signals based on adaptive sparse dictionary model," *Measurement and Control*, vol. 53, no. 3-4, pp. 649-661, 2020/03/01 2020, doi: 10.1177/0020294019898725.
- [13] X. J. Dang, F. H. Wang, and D. X. Zhou, "Compressive Sensing of Vibration Signals of Power Transformer," in *2020 IEEE International Conference on High Voltage Engineering and Application (ICHVE)*, 6-10 Sept. 2020 2020, pp. 1-4, doi: 10.1109/ICHVE49031.2020.9279709
- [14] . R. G. Baraniuk, "Compressive Sensing [Lecture Notes]," *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 118-121, 2007, doi: 10.1109/MSP.2007.4286571.
- [15] L. StankoviĆ, E. SejdiĆ, S. StankoviĆ, M. DakoviĆ, and I. OroviĆ, "A Tutorial on Sparse Signal Reconstruction and Its Applications in Signal Processing," *Circuits, Systems, and Signal Processing*, vol. 38, no. 3, pp. 1206-1263, 2019/03/01 2019, doi: 10.1007/s00034-018-0909-2.
- [16] M. Stéphane, "CHAPTER 12 - Sparsity in Redundant Dictionaries," in *A Wavelet Tour of Signal Processing (Third Edition)*, M. Stéphane Ed. Boston: Academic Press, 2009, pp. 611-698.
- [17] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489-509, 2006, doi: 10.1109/TIT.2005.862083.
- [18] B. I. Simmons, C. Hoepke, and W. J. Sutherland, "Beware greedy algorithms," *Journal of Animal Ecology*, <https://doi.org/10.1111/1365-2656.12963> vol. 88, no. 5, pp. 804-807, 2019/05/01 2019, doi: <https://doi.org/10.1111/1365-2656.12963>.

บรรณานุกรม



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ภาคผนวก ก.

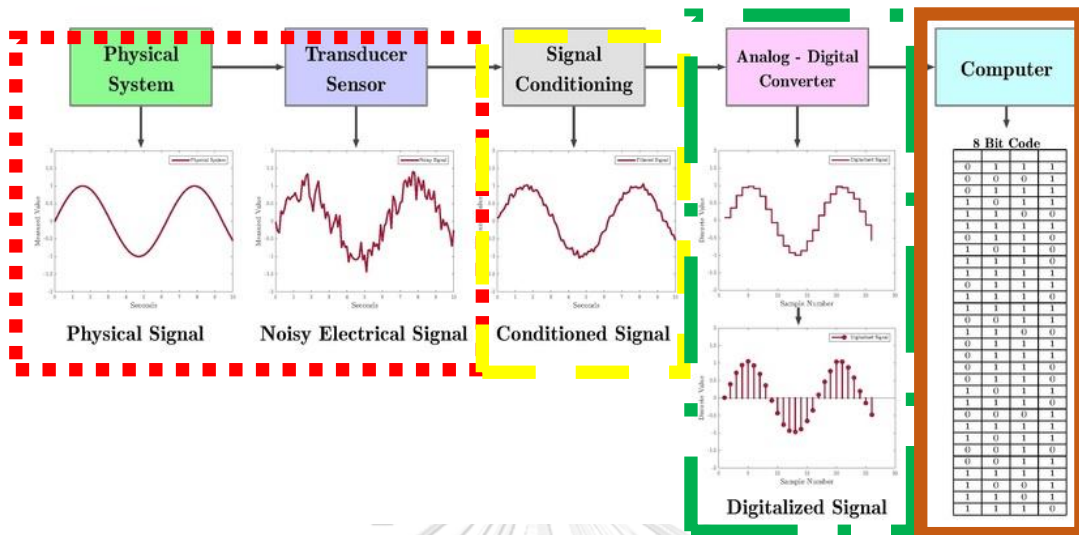
อุปกรณ์ และ เครื่องมือที่ใช้ทำการทดสอบ

ก.1 เซ็นเซอร์วัดการสั่นสะเทือนแบบตัววัดความเร่งเครื่องกลไฟฟ้าจุลภาค

MEMS หรือ Microelectromechanical system คือ ระบบเครื่องกลไฟฟ้าจุลภาค เป็นคำที่ใช้เรียกระบบเครื่องกลไฟฟ้าที่ถูกผลิตขึ้นมาด้วยเทคโนโลยีอิเล็กทรอนิกส์ขนาดเล็ก (Microelectronic fabrication technique) เทคนิคนี้สามารถสร้างโครงสร้าง หรือ กลไกรับรู้ทางกล (mechanical sensing) ที่มีขนาดเล็กมากได้ เมื่อนำไปรวมกับ วงจรไมโครอิเล็กทรอนิกส์ทำให้สามารถ ถูกใช้เป็นอุปกรณ์การวัดเชิงกล เช่น เซ็นเซอร์วัดความเร่ง (MEMS accelerometer) เซ็นเซอร์นี้ขนาดเล็ก เมื่อเทียบขนาดกับ เซ็นเซอร์วัดความเร่งแบบ Piezoelectric vibration sensor ซึ่งถูกใช้กันอย่างแพร่หลายในอุตสาหกรรม ระบบการวัดสัญญาณด้วยเซ็นเซอร์แบบเดิมกับแบบเครื่องกลไฟฟ้าจุลภาค แสดงดังรูปที่ ก.1 และ ก.2 ตามลำดับ เมื่อเปรียบเทียบกันแล้ว เซ็นเซอร์วัดความเร่งแบบเดิม มีราคาสูงและทำหน้าที่เป็นเพียงตัวรับรู้ (รูปที่ ก.1 กรอบเส้นจุด สีแดง) จำเป็นต้องมีองค์ประกอบอื่นๆ เช่น การกรองสัญญาณ การขยายสัญญาณ (รูปที่ ก.1 กรอบเส้นประ สีเหลือง) และการแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล (รูปที่ ก.1 กรอบเส้นประ-จุด สีเขียว) เพิ่มเติม เพื่อทำการปรับสภาพสัญญาณ (Signal Conditioning) ให้สามารถนำไปวิเคราะห์ได้ (รูปที่ ก.1 กรอบสีน้ำตาล) แต่สำหรับตัววัดความเร่งเครื่องกลไฟฟ้าจุลภาคนั้น มีหน่วยปรับสภาพสัญญาณ (Signal Conditioning unit) ผังตัวอยู่ภายในเรียบร้อยแล้วทำให้ลดความยุ่งยากในกระบวนการเก็บข้อมูล จึงไม่จำเป็นต้องใช้อุปกรณ์เพิ่มเติมที่ซับซ้อน (รูปที่ ก.2 กรอบสีฟ้า)

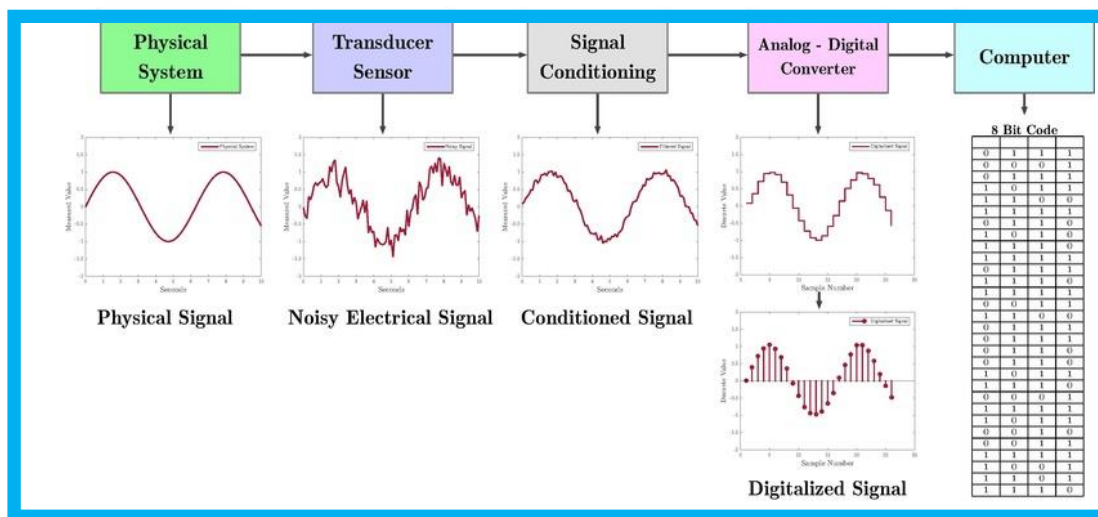
ประเภทของ ตัววัดความเร่งเครื่องกลไฟฟ้าจุลภาคสามารถแยกได้เป็น 2 ประเภทหลักๆ คือ แบบความต้านทานเพียโซ (Piezoresistive) และ แบบคาปาซิทีฟ (Capacitive) โดยงานวิจัยนี้จะเน้นไปที่เซ็นเซอร์วัดความเร่งแบบคาปาซิทีฟ เนื่องจากมีราคาถูกกว่ามากและมีวงจรปรับสภาพสัญญาณในตัว ทำให้ลดความยุ่งยากในการทำงานในงานวิจัยนี้ใช้ เซ็นเซอร์ ADXL 345 ซึ่งเป็นเซ็นเซอร์วัดความเร่งแบบคาปาซิทีฟ ในการทำการวิจัยเป็นหลัก แต่สำหรับเซ็นเซอร์วัดความเร่งความต้านทานเพียโซ (Piezoresistive) นั้นมีประสิทธิภาพที่สูงกว่าแต่ต้องแลกมาด้วยราคาและความยุ่งยากในการทำงานเนื่องจากไม่มีหน่วยปรับสภาพสัญญาณ (Signal Conditioning unit)

Digital Data Acquisition System



รูปที่ ก.1 แผนผังระบบการรวบรวมและวิเคราะห์ข้อมูล (Data acquisition) สำหรับเซ็นเซอร์แบบวงจรรวม

Digital Data Acquisition System

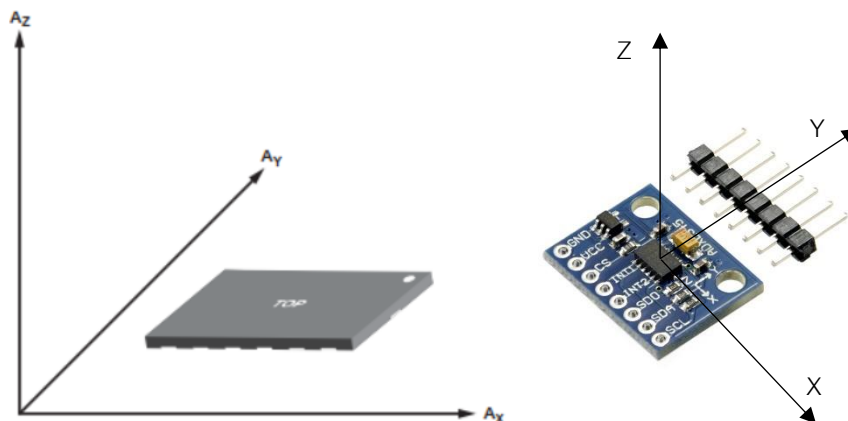


รูปที่ ก.2 แผนผังระบบการรวบรวมและวิเคราะห์ข้อมูล (Data acquisition) สำหรับเซ็นเซอร์แบบเครื่องกลไฟฟ้า
จุลภาค

ก.2 เซ็นเซอร์วัดความเร่ง ADXL 345

ตัววัดความเร่งเครื่องกลไฟฟ้าจุลภาค (Microelectromechanical System) แบบคาปาซิทีฟ (Capacitive based accelerometer) แสดงดังรูปที่ ก.3 ตัววัดชนิดนี้ อาศัยหลักการเปลี่ยนแปลงค่าความจุทาง

ไฟฟ้า (capacitance) ของตัวนำโดยการเพิ่มหรือลดช่องว่างระหว่างมวล และ อิเล็กโทรด ซึ่งสามารถวัดได้ 3 ทิศทางพร้อมกันแต่สำหรับงานวิจัยนี้จะมุ่งเน้นไปที่การวัดเพียงแคแกนเดียวเท่านั้นคือแกน Z

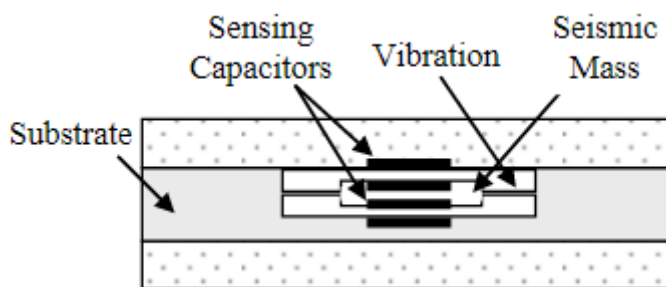


รูปที่ ก.3 แสดงทิศทางของการวัดในตำแหน่งต่างเทียบกับเซ็นเซอร์

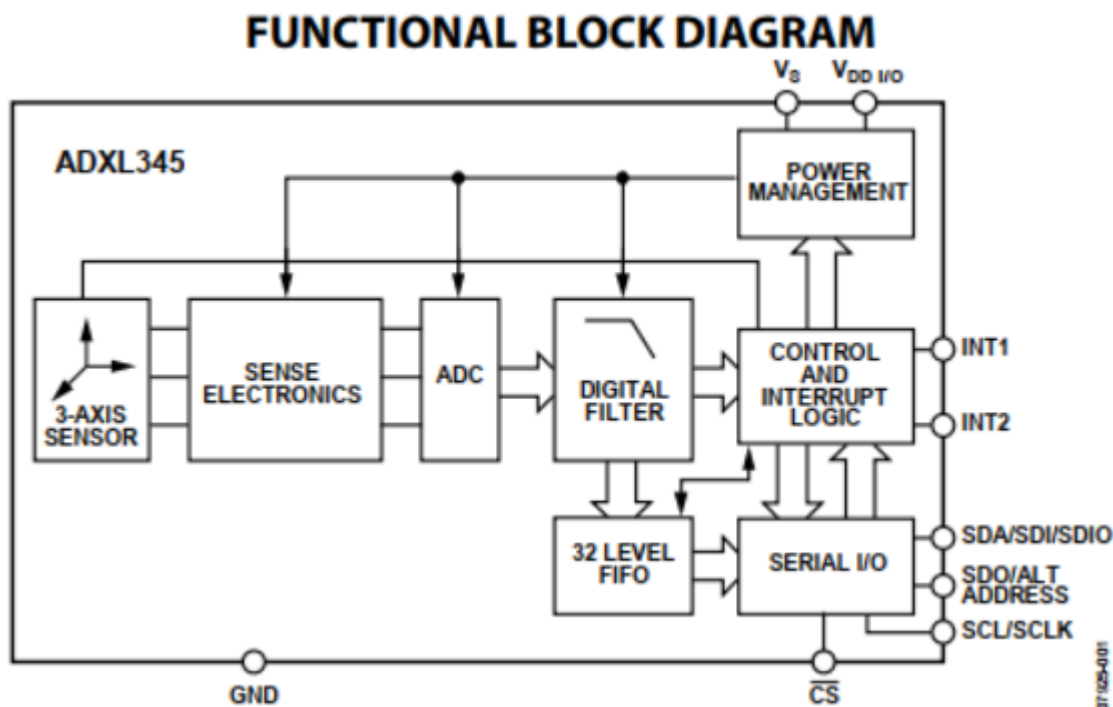
รูปที่ ก.3 เป็นการแสดงทิศทางการอ่านค่าการสั่นสะเทือนในทิศทางต่างๆ โดย แนวตั้งคือแนวแกน Z และเมื่อวางตัวเซ็นเซอร์ให้แกน Z ตั้งฉากกับพื้นโลกโดยไม่มีการสั่นสะเทือนตัวเซ็นเซอร์ก็จะส่งสัญญาณออกมาเท่ากับ 1 g และถ้ากำหนดพิสัยของเซ็นเซอร์ไว้ที่ $\pm 2g$ ค่าที่อ่านได้ของเซ็นเซอร์ จะเท่ากับ 2^{10} เนื่องจากตัวเซ็นเซอร์มีความละเอียดในการวัดอยู่ที่ $2^{10} \text{ bit} - \text{resolution}$

จากรูปที่ ก.4 แสดงถึงโครงสร้างภายใน ของ MEMS accelerometer ซึ่งจะประกอบไปด้วย Seismic Mass หรือมวลสั่นสะเทือนซึ่งซ้อนกันหลายชั้นสลับกับ Sensing Capacitor ประกอบอยู่ภายในสารกึ่งตัวนำ หรือ Substrate ซึ่งจะมีระยะห่างอยู่ค่าหนึ่ง ซึ่งเมื่อเกิดการสั่นจะทำให้ระยะห่างของ Seismic Mass และ Sensing Capacitor เปลี่ยนไปซึ่งส่งผลให้ เกิดการเปลี่ยนแปลงของค่าความจุทางไฟฟ้า ทำให้เกิดแรงดันไฟฟ้าเกิดขึ้น

จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ก.4 แสดงลักษณะของ ตัววัดความเร่งแบบคาปาซิทีฟ



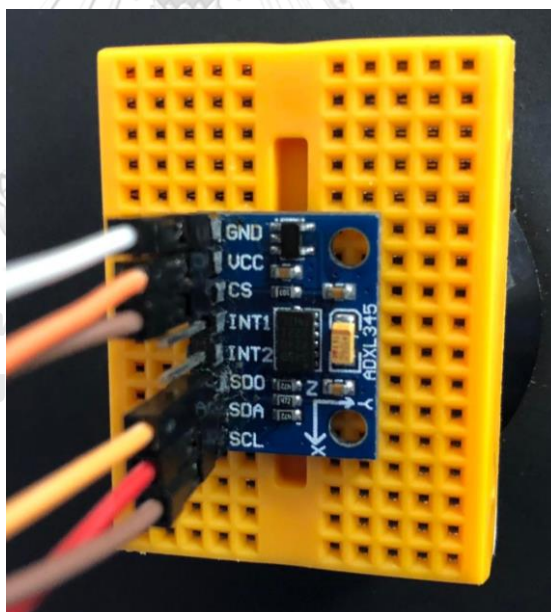
รูปที่ ก.5 แสดงแผนผังการทำงานภายใน ตัววัดความเร่ง ADXL 345

รูปที่ ก.5 แสดงแผนผังภายในตัววัดความเร่ง ADXL 345 ซึ่งมีหน่วยปรับสภาพสัญญาณ (Signal Conditioning unit) เช่น การกรองสัญญาณ การแปลงจากสัญญาณอนาล็อกเป็นดิจิทัล ผังอยู่ภายในตัววัดความเร่งเรียบร้อยแล้ว

สำหรับ ตารางที่ ก.1 แสดงคุณสมบัติของตัววัดความเร่ง ADXL345 โดย มีค่าพารามิเตอร์ที่สำคัญคือ Measurement Rate ซึ่งเป็นความถี่ในการสุ่มข้อมูลของตัวเซ็นเซอร์แต่ด้วย Bandwidth (ความเร็วในการส่งผ่านข้อมูลไปสู่อุปกรณ์อื่นๆ เช่น คอมพิวเตอร์) ที่มีขนาดครึ่งหนึ่ง ส่งผลให้ปริมาณจุดข้อมูลที่อ่านได้ต่อวินาทีเหลือ 1600 Hz หมายความว่าเซ็นเซอร์ตัวนี้สามารถวัดด้วยความถี่สูงสุดได้เพียง 1600 Hz ทำให้ความถี่สูงสุดที่วัดได้ตามเกณฑ์ในควิส น้อยกว่า 800 Hz ในการวิเคราะห์การสั่นสะเทือนของเครื่องจักรถือว่าไม่เพียงพอต่อการใช้งานเนื่องจากหากเครื่องจักร ประกอบด้วยชุดเฟือง หรือตลับลูกปืนแล้ว การสั่นสะเทือนจะเกิดที่ความถี่สูงกว่าความเร็วรอบหมุนเพลามาก

ตารางที่ ก.1 คุณสมบัติของตัววัดความเร่ง ADXL 345

คุณลักษณะ	หน่วย
Range	$\pm 2g, \pm 4g, \pm 8g, \pm 16g$
Nonlinearity	$\pm 5\%$ full scale
Sensitivity	
$\pm 2g$ Range	256 LSB/g
$\pm 4g$ Range	128 LSB/g
$\pm 8g$ Range	64 LSB/g
$\pm 16g$ Range	32 LSB/g
0 g Bias Level	$\pm 80mg$
Output Data Rate	3200Hz
Bandwidth	1600Hz



รูปที่ ก.6 เซ็นเซอร์ ADXL345

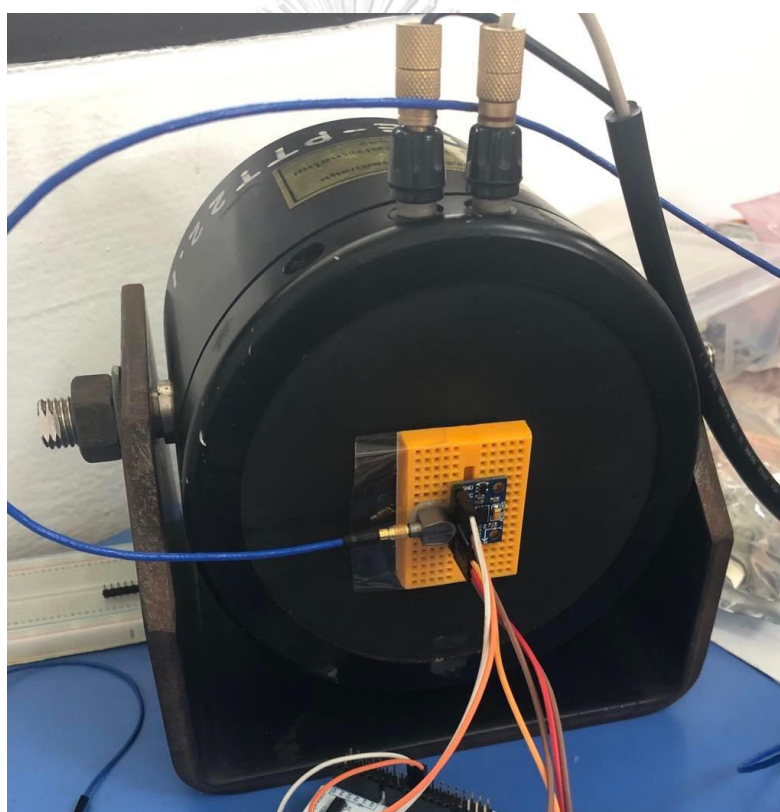
ก.2.1 วงจรกรองความถี่ต่ำผ่านของระบบเก็บข้อมูล

เซ็นเซอร์ ADXL345 มีวงจรกรองความถี่ต่ำผ่านแบบอนาล็อกรวมอยู่ภายในเรียบร้อยแล้ว ซึ่งสามารถปรับได้ ผ่านการตั้งค่า Output Data Rate ในวงจร โดยมีค่าแบนด์วิธสูงสุดที่ 1600 เฮิรตซ์ การทดสอบในงานวิจัยนี้ทั้งหมดจะตั้งค่า Output Data Rate ที่แบนด์วิธ 1600 เฮิรตซ์ เพื่อให้สัญญาณที่วัดออกมาไม่มีพิสัยของข้อมูลที่มีความสมบูรณ์กว้างที่สุดตามขีดจำกัดของตัวเซ็นเซอร์ อย่างไรก็ตามบริเวณของข้อมูลที่มีความสมบูรณ์

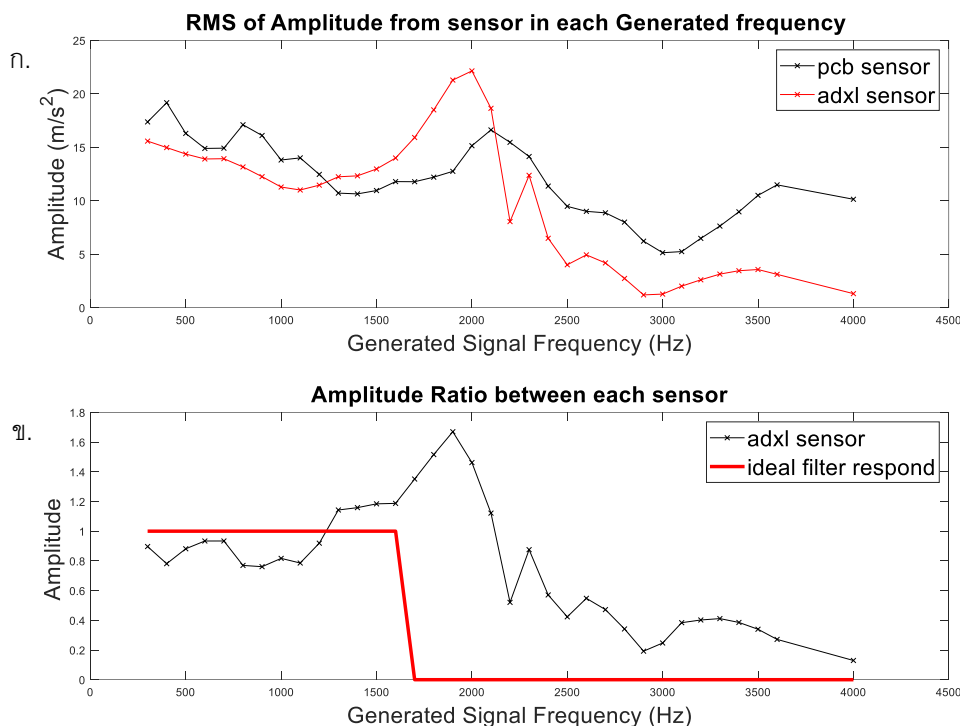
และไม่สมบูรณ์นั้นขึ้นอยู่กับลักษณะของวงจรรองความถี่ต่ำ ภายในตัวเซ็นเซอร์ ADXL345 เองซึ่งข้อมูลของวงจรรองความถี่ต่ำที่ใช้ ไม่ปรากฏในคู่มือการใช้งาน ในงานวิจัยนี้ก็จะทำการทดสอบเพื่อให้ทราบถึงคุณลักษณะและขีดจำกัดของตัวเซ็นเซอร์ ADXL345 เพื่อที่จะใช้ในการเลือกย่านความถี่ที่จะใช้ในการทดสอบการสุมวัดและการสร้างสัญญาณสปาร์ต Z_S ต่อไป

การตรวจสอบลักษณะของวงจรรองความถี่ต่ำ

การทดสอบนี้จะทำการหาบริเวณของข้อมูลที่มีความสมบูรณ์และไม่สมบูรณ์ ทำโดยการวัดสัญญาณการสั่นสะเทือนที่ความถี่ต่างๆ ด้วยเซ็นเซอร์ ADXL345 ที่มีอัตราการสุ่มจุดข้อมูล 3125 เฮิรตซ์ เปรียบเทียบกับตัววัดความเร่ง PCB TLD352A56 ที่มีอัตราการสุ่มจุดข้อมูล 20000 เฮิรตซ์ โดยวัดการสั่นสะเทือนของตัวกระตุ้น (Shaker) ดังแสดงในรูปที่ ก.7 ซึ่งเป็นเซ็นเซอร์วัดความเร่งที่ใช้ในงานวิจัยทั่วไป ความถี่ที่ ทดสอบในที่นี้จะใช้ตั้งแต่ 300 เฮิรตซ์ ถึง 4000 เฮิรตซ์



รูปที่ ก.7 การทดสอบหาลักษณะของวงจรรองความถี่ต่ำ



รูปที่ ก.8 ก. ค่าเฉลี่ยกำลังสองของแอมพลิจูดบนโดเมนเวลาเมื่อวัดสัญญาณการสั่นสะเทือนที่ความถี่ต่างๆ

ข.) อัตราส่วนของแอมพลิจูดระหว่างเซ็นเซอร์ ADXL และ PCB ที่ความถี่ต่างๆ

รูปที่ ก.8(ก.) เป็นการแสดงค่าแอมพลิจูดสูงสุดสุดของสัญญาณ ในการทดสอบแต่ละความถี่ โดยกำหนดให้แกนนอนเป็นความถี่ที่กำหนดโดย เครื่องกำเนิดสัญญาณ รูปที่ ก.8(ข.) เป็นอัตราส่วนระหว่างแอมพลิจูดสูงสุดของสัญญาณที่วัดโดยเซ็นเซอร์ ADXL และ PCB เมื่อใช้ ค่าแอมพลิจูดจากเซ็นเซอร์ PCB ซึ่งเป็นเซ็นเซอร์ที่มีคุณภาพสูงและมีความถี่ในการสุ่มสัญญาณสูงมากเป็นค่าอ้างอิง เพื่อตรวจสอบผลกระทบของวงจรกรองความถี่ต่ำผ่านของเซ็นเซอร์ ADXL345 จะเห็นว่าวงจรกรองความถี่ต่ำในเซ็นเซอร์ ADXL345 ไม่ได้ทำการลดทอนสัญญาณในย่านที่มากกว่าแบนด์วิดท์สูงสุดในทันทีแต่จะมีริบเปิลเกิดขึ้นหลังจากนั้นต่อไปอีกจนถึงความถี่ 2000 เฮิรตซ์ ก่อนที่สัญญาณเริ่มจะถูกลดทอนลงเรื่อยๆ จนเข้าใกล้ศูนย์ที่บริเวณ 2900-3000 เฮิรตซ์ เมื่อเทียบกับ การตอบสนองในอุดมคติที่ควรจะเป็น (เส้นสีแดง) ซึ่งทำให้พอทราบได้ว่าคุณภาพของสัญญาณที่ได้บริเวณความถี่ที่ใกล้กับค่าแบนด์วิดท์สูงสุดนั้นจะมีบริเวณของข้อมูลที่มีความไม่สมบูรณ์ครอบคลุมจากความถี่ที่ 1525 เฮิรตซ์ ลดลงไปจนถึง 1000 เฮิรตซ์

ลักษณะของวงจรกรองความถี่ต่ำผ่านแบบนี้จะส่งผลให้การทดสอบในกรณีนี้ที่วัดกับเครื่องจักรจริงในบริเวณที่มีความไม่สมบูรณ์เกิดความคลาดเคลื่อนสูง

ก.3 บอร์ดไมโครคอนโทรลเลอร์ STM32F411RE (Nucleo-F411RE)

บอร์ด STM32F411RE เป็นบอร์ดไมโครคอนโทรลเลอร์ ที่ใช้ชิพประมวลผล Arm Cortex M4 ใช้สำหรับการรับส่งข้อมูล และควบคุมการอ่านค่าของตัวเซ็นเซอร์ตามเวลาที่กำหนด และส่งข้อมูลที่เป็นดิจิตอล (ชุด

เวกเตอร์ของ y_t และ t_y) ให้กับคอมพิวเตอร์ เพื่อนำไปใช้กับอัลกอริทึมที่ถูกสร้างขึ้นในโปรแกรม matlab ต่อไป คุณสมบัติของไมโครคอนโทรลเลอร์ STM32F411RE แสดงในตารางที่ ก.2

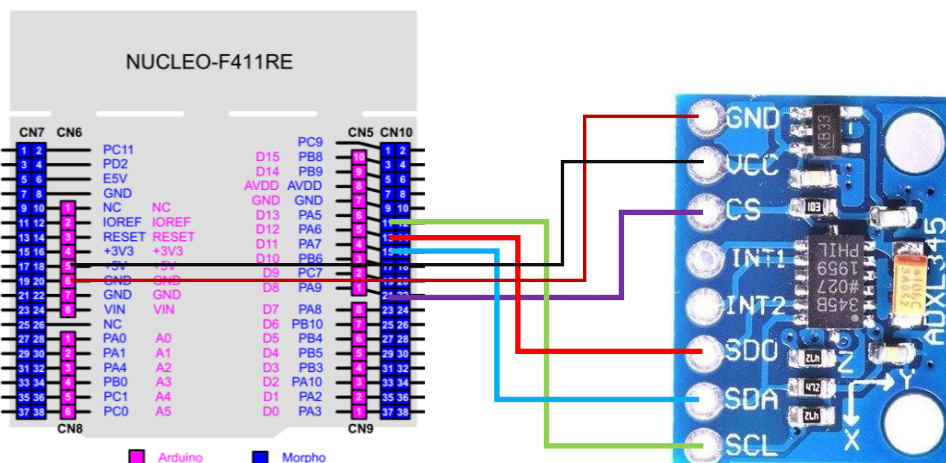
ตารางที่ ก.2 คุณสมบัติของไมโครคอนโทรลเลอร์ STM32F411RE

คุณลักษณะ	หน่วย
Operating Voltage	5V
Digital I/O Pin	14
PWM Digital I/O Pin	6
Analog Input Pin	6
DC Current Per I/O Pin	20mA
Clock speed	16MHz

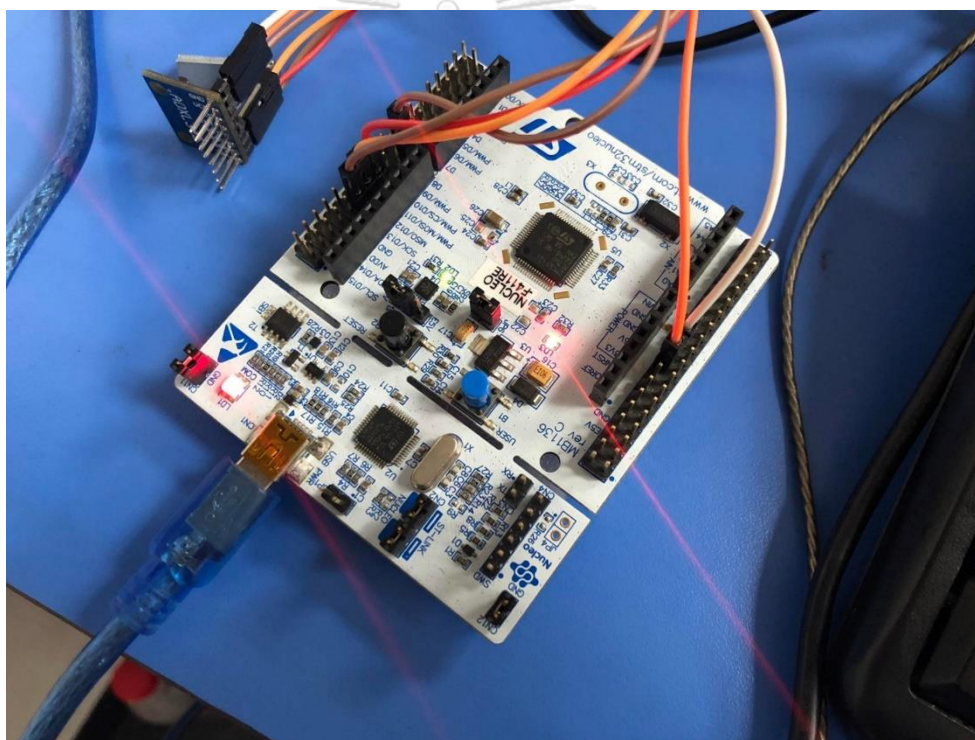
การสื่อสารระหว่างเซ็นเซอร์ STM32F411RE และบอร์ด STM32F411RE ด้วย SPI โปรโตคอล

สำหรับวิธีการที่จะควบคุมการวัดระหว่างเซ็นเซอร์และบอร์ดไมโครคอนโทรลเลอร์เองนั้นสามารถทำได้ โดยให้การสื่อสารแบบ SPI หรือเรียกว่า Serial Peripheral interface โดยกำหนดให้ บอร์ดไมโครคอนโทรลเลอร์เป็นตัวควบคุมการทำงานของเซ็นเซอร์ ตามพอร์ตสัญญาณดังนี้

- 1.SCLK (Serial Clock) เป็นสายสัญญาณนาฬิกาที่ถูกส่งออกจาก บอร์ดไมโครคอนโทรลเลอร์ เพื่อเป็นสัญญาณอ้างอิงเวลาในการทำงาน
2. SDO (Serial Data Out) เป็นสายสัญญาณที่ส่งข้อมูลออกจาก บอร์ดไมโครคอนโทรลเลอร์ไปยังเซ็นเซอร์ เช่น คำสั่ง วัดค่า
3. SDA (Serial Data In) เป็นสายสัญญาณที่รับข้อมูลจาก เซ็นเซอร์ ไปยังบอร์ดไมโครคอนโทรลเลอร์ เช่น ค่าที่วัดมาได้
4. CS (Chip Select) เป็นสายสัญญาณสำหรับเลือก เซ็นเซอร์ เพื่อให้พร้อมทำงาน



รูปที่ ก.9 แสดงการต่อสายระหว่างบอร์ดและเซ็นเซอร์ ทั้ง 4-wire SPI และไฟเลี้ยงวงจร



รูปที่ ก.10 บอร์ดไมโครคอนโทรลเลอร์ STM32F411RE

ก.4 ตัววัดความเร่ง (Piezoelectric) รุ่น PCB TLD352A56

ในการทำการทดสอบสำหรับงานวิจัยนี้จะใช้เซ็นเซอร์ PCB ที่มีคุณภาพสูงด้วยจุดประสงค์ 2 อย่าง คือ

1. เพื่อทดสอบอัลกอริทึมคอมเพรสชันซึ่งในขั้นตอนการทดสอบหลักการ (หัวข้อ 5.1)
2. เพื่อเปรียบเทียบและตรวจสอบความถูกต้องเมื่อใช้ เซ็นเซอร์ ADXL345 ที่มีคุณภาพต่ำกว่า (หัวข้อ 5.2) โดยย่านความเร่งที่ PCB ทำการวัดได้คือ $\pm 490 \frac{m}{s^2}$ และความถี่ที่วัดได้อยู่ในช่วง 0.5 – 10000Hz



รูปที่ ก.11 เซ็นเซอร์ PCB TLD352A56

ก.5 ระบบเก็บข้อมูล Data Acquisition รุ่น Dewesoft Sirius

การ์ดสำหรับเปลี่ยนสัญญาณอนาล็อกเป็นดิจิทัล, ปรับปรุงสัญญาณ และเก็บข้อมูล ใช้ควบคู่กับ เซ็นเซอร์ PCB TLD352A56 ในการนำสัญญาณการสั่นสะเทือนที่วัดได้เข้าสู่คอมพิวเตอร์ด้วยซอฟต์แวร์ Dewesoft X



รูปที่ ก.12 Data Acquisition รุ่น Dewesoft Sirius

ก.6 คอมพิวเตอร์ สำหรับวิเคราะห์และประมวลผลสัญญาณ

ใช้สำหรับรับข้อมูลจากทั้ง STM32F411RE (สัญญาณ y_t และ t_s) และ Dewesoft Sirius (สัญญาณ x_t) เข้ามาเพื่อนำไปใช้ใน Compresses sensing อัลกอริทึม ในโปรแกรม Matlab

ก.7 เครื่องสร้างการสั่นสะเทือน (Shaker Exciter) bruel & kjaer Type 4809

เป็นเครื่องสร้างการสั่นสะเทือนตามที่ถูกป้อนสัญญาณไฟฟ้าเข้าไป



รูปที่ ก.13 bruel & kjaer Type 4809

ก.8 เครื่องกำเนิดสัญญาณ (Signal Function generator) GWinstek GFG-3015

เครื่องสร้างสัญญาณไฟฟ้าให้มีความถี่และขนาดตามที่กำหนดเพื่อควบคุมการสั่นสะเทือนโดยจะส่งสัญญาณตามที่ตั้งค่าไว้และสัญญาณจะถูกขยายผ่านเครื่องขยายสัญญาณก่อนจะเข้าสู่ เครื่องสร้างการสั่นสะเทือน



รูปที่ ก.14 GWinstek GFG-3015

ก.9 เครื่องขยายสัญญาณ (Power Amplifier) bruel & kjaer Type 2706

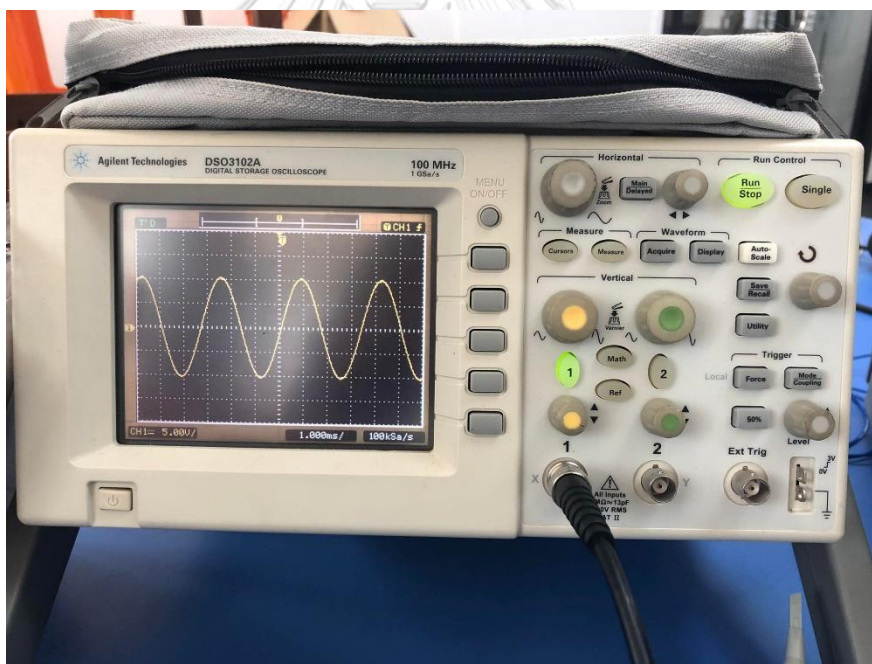
ทำหน้าที่ขยายสัญญาณที่ไฟฟ้าที่ถูกส่งมาจากเครื่องกำเนิดสัญญาณ เพื่อนำไปควบคุมเครื่องสร้างการสั่นสะเทือน



รูปที่ ก.15 bruel & kjaer Type 2706

ก.10 ออสซิลโลสโคป Agilent Technologies DSO3102A

ใช้สำหรับตรวจสอบสัญญาณไฟฟ้าที่ส่งออกมาจากเครื่องกำเนิดสัญญาณ ให้มีขนาด และความถี่ที่ถูกต้องตามที่ต้องการ

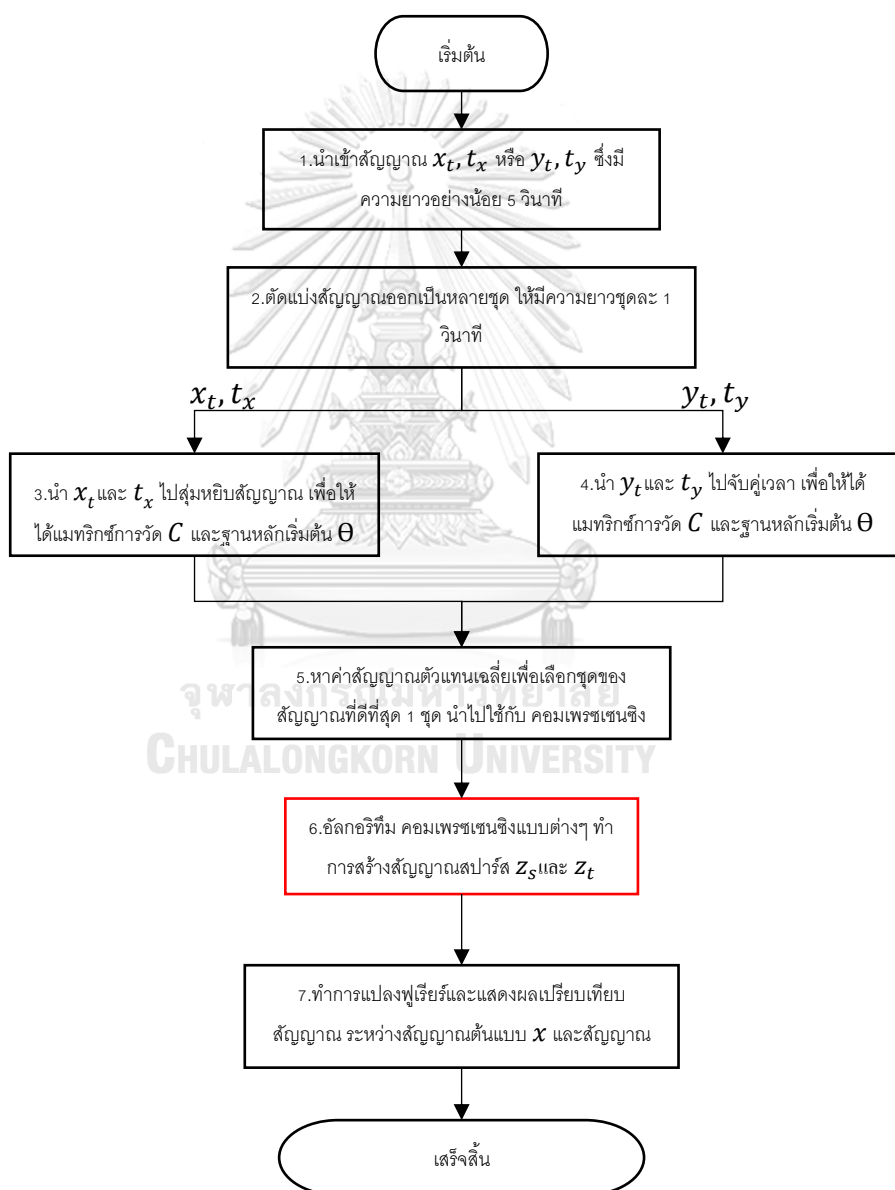


รูปที่ ก.16 Agilent DSO3102A

ภาคผนวก ข.

อัลกอริทึมคอมพิวเตอร์เซซซิงและ ฟังก์ชันที่ใช้งานอื่นๆ ในโปรแกรม MATLAB

สำหรับในงานวิจัยนี้ใช้โปรแกรม MATLAB ที่ใช้ในการเขียนอัลกอริทึมการประมวลผลสัญญาณซึ่งชุดโปรแกรมหลักสามารถจะประมวลผลสัญญาณหลายๆ ชุดได้ภายในครั้งเดียว เนื่องจากปริมาณของสัญญาณที่มาก ส่งผลให้การ manipulate หรือ การ Handling ตัวแปรมีความซับซ้อน เพื่อให้ง่ายแก่การทำงาน ชุดโปรแกรมจะถูกแบ่งออกเป็นหลายโมดูลด้วยกันดังนี้



รูปที่ 1. ข แสดงแผนผังการทำงานของอัลกอริทึม

ส่วนประกอบของโปรแกรมในแต่ละโมดูล

1. นำเข้าสัญญาณ จากไฟล์เดอร์ที่กำหนด การนำเข้าสัญญาณหลายๆ สัญญาณพร้อมกันเพื่อเตรียมนำไปเข้ากระบวนการอื่นเพิ่มเติมแล้วแต่ประเภทของสัญญาณที่นำเข้ามา โดยภายในโค้ดจะมีฟังก์ชันใช้งานอยู่ภายในคือ `Func_Signal_File_management_func_TXT_struct` ใช้สำหรับนำเข้าและปรับปรุงสัญญาณก่อนที่จะนำสัญญาณที่ปรับปรุงแล้วเข้ามาสู่โปรแกรมหลักซึ่งจะขอกว่าเพิ่มเติมในภาคผนวก ข

```
clear all,close all,clc
DIF.PCBfolder='D:\UDEMY3\THESIS\Data driven\Experiment\23-8-2022\FileforUNDERSAMP\Undersamp_Input\PCB\SHAKER\PCB3200Hz';
DIF.TargetfolderADXl='D:\UDEMY3\THESIS\Data driven\Experiment\23-8-2022\FileforUNDERSAMP\Undersamp_Input\ADXL\SHAKER\20_1000_R1';
DIF.Currentfolder = 'D:\UDEMY3\THESIS\Data driven\Experiment\23-8-2022\FileforUNDERSAMP';
%1=Hor3200 2=Hor3200 3=Hor20000 4=Hor20000
cd (DIF.PCBfolder);
DIF.datalist = dir ('*.mat');
DIF.NoFile = {DIF.datalist.name}'; % This DIF.NoFile can't use to loop just counting file used only
for i=1:length(DIF.NoFile)
    cd (DIF.PCBfolder)
    input=load (char(sprintf('%d.mat', (i))));
    % Normal
    pcb(i).x=input.x;
    pcb(i).t=input.t;
    pcb(i).srate=length(pcb(i).t)/max(pcb(i).t);
    cd (DIF.Currentfolder)
    clear input
end
[adx1]=Func_Signal_File_management_func_TXT_struct(DIF,20,1000,4);
% Convert Undersamp
for i=1:size(adx1.x,2)
input(i).x=cell2mat(adx1.x(i));
input(i).t=cell2mat(adx1.t(i));
input(i).delt=cell2mat(adx1.delt(i));
end
clear adx1
adx1=input;
clear input i
```

2. การตัดสัญญาณให้สั้นลง โดยปกติสัญญาณที่ถูกวัดจะมีความยาวที่มาก (10 ถึง 30 วินาที) จึงจำเป็นต้องตัดแบ่งสัญญาณให้มีขนาดเล็กลงเป็นหลายๆ ชุด ภายในจะมีฟังก์ชันชื่อ `Func_Signal_windowing` ใช้สำหรับตัดสัญญาณให้มีขนาดสั้นลงก่อนนำเข้าสู่โปรแกรมหลัก

```
%% Data Chop
overlap =1;
winlength=1;
for i=1:size(pcb,2)
    input.x=(pcb(i).x);
    input.t=(pcb(i).t);
    [output]=Func_Signal_windowing(input,overlap,winlength);
    pcb(i).windt=output.windt;
```

```

        pcb(i).windx=output.windx;
        pcb(i).Startimewind=output.Startimewind;
        clear input output
    end
function [output]=Func_Signal_windowing(input,overlap,winlength)
%     overlap = 0.5;
    input.winStart=0:overlap:floor(max(input.t))-winlength;
    input.winEnd=winlength:overlap:floor(max(input.t));
    for j=1:length(input.winStart)

input.indStart(j)=min(find(input.t>=input.winStart(j)));

input.indEnd(j)=min(find(input.t>=input.winEnd(j)));
        output.windt(:,j)={input.t(input.indStart(j):
input.indEnd(j))-input.t(input.indStart(j))};

output.Startimewind(:,j)=input.t(input.indStart(j));
        output.windx(:,j)={input.x(input.indStart(j):
input.indEnd(j))};
    end
        clear input
    end
end

```

3. การสุ่มหยิบข้อมูล ใช้ในกรณีที่ข้อมูลที่ได้มาเป็นสัญญาณแบบคาบคงที่ปกติ แต่ถ้าหากสัญญาณที่ได้มาเป็นสัญญาณแบบสุ่มคาบอยู่แล้ว ไม่จำเป็นต้องใช้ โมดูลตัวนี้

```

%% Downsampling Rand Perm
for i=1:size(pcb,2)
    for j=1:size(pcb(i).windt,2)
        input.t=cell2mat(pcb(i).windt(j));
        input.x=cell2mat(pcb(i).windx(j));
        input.srate=length(input.t)/max(input.t);
        perm=1;
        k=2;
        for l=2:size(input.x,2)
            if rand(1,1)>0.4 &&l-perm(k-1)>3
                perm(k)=l;
                k=k+1;
            end
        end
        perm=sort(perm,'ascend');
        % Disable time matching
        perm=perm(perm<3200*winlength);
        pcb(i).undwindt_idc(j)={perm};

        input.x = input.x(perm);
        input.t = input.t(perm);
        pcb(i).undwindx(j)={input.x};
        pcb(i).undwindt(j)={input.t};
        pcb(i).perm(j)={perm};

        for l=2:length(perm)
            timedel(l-1)={perm(l)-perm(l-1)};
        end
    end
end

```

```

                                pcb(i).timedel(j)={timedel};
                                clear input timedel perm
                                end
                                end
end

```

4. การจับคู่เวลา ทำการหาตำแหน่งของจุดเวลาของสัญญาณสุ่มคาบ บนสัญญาณปกติเพื่อนำไปตัดแปลงแมทริกซ์การวัด ภายในโมดูลจะมีฟังก์ชันชื่อ **Func_TimeMatching_struct** อยู่ใช้สำหรับจับคู่เวลาระหว่างเวลาของสัญญาณสุ่มคาบและเวลาของสัญญาณเป้าหมาย สำหรับการทดสอบ ที่ใช้การสุ่มหยิบข้อมูลอยู่แล้วไม่จำเป็นต้องทำหัวข้อนี้ เนื่องจากจะได้ตัวแปรชื่อ **pcb.perm** สามารถนำไปใช้ในขั้นตอนต่อไปได้เลย

```

%% Time array matching
target.srate=3200;
target.maxt=winlength;
target.n=target.maxt*target.srate;
target.t=linspace(0,target.maxt,target.n)';
target.srate=length(target.t)/max(target.t);
for i=1:size(adx1,2)
    for j=1:size(adx1(i).windt,2)
        input.t=cell2mat(adx1(i).windt(j));
        [I]=Func_TimeMatching_struct(target,input.t);
        adx1(i).windt_idc(j)={unique(I)};
        clear input I
    end
end
function [I]=Func_TimeMatching_struct(target,input)
    for i=1:length(input)
        utimevec=input;
        for j=1:length(target.t)
            output_tDiff(:,j)=abs(target.t(j)-utimevec(i));
        end
        [V,I(i)]=min(output_tDiff);
        clear V output_tDiff
    end
end
end

```

5. หาค่าสัญญาณตัวแทนเฉลี่ย สำหรับสัญญาณที่ถูกตัดมาหลายๆ ชุด ความยาวชุดละ 1 วินาที จะถูกแปลงให้เป็นสัญญาณตัวแทน(3.2.2.1) หลายๆ ชุดเพื่อเลือกชุดสัญญาณที่ดีที่สุดไปใช้กับคอมเพรสชันซิง โดยการเลือกสัญญาณที่มีแอมพลิจูดเฉลี่ยต่ำสุด

```

%% Mean Proxy
CS.f=target.n;
CS.Psi = dct(eye(CS.f,CS.f));
for i=1:size(adx1,2)
    for j=1:size(adx1(i).windt,2)
        input.timeind=cell2mat(adx1(i).windt_idc(j));
        input.underx=cell2mat(adx1(i).windx(j));
        input.Theta=CS.Psi(input.timeind,:);
        input.proxy=abs(input.Theta'*input.underx);
        input.noisefloor=mean(input.proxy)';
        adx1(i).proxy(:,j)=input.proxy;
        adx1(i).meanproxy(j)=input.noisefloor;
    end
end

```

```

        clear input I
    end
end
%% Mean proxy sorting
adx1(1).dcthz = linspace(0, (target.srate/2), target.n)';
for i=1:size(adx1,2)

[adx1(i).mproxyrankvalue, adx1(i).mproxyrank]=sort(adx1(i).meanproxy, '
ascend');
end
for i=1:size(adx1,2)
    bestrank=adx1(i).mproxyrank(1);
    input.bwindt=(adx1(i).windt(1,bestrank));
    input.starttime=(adx1(i).Starttimewind (bestrank));
    adx1(i).bst=input.starttime;
        for l=2:length(input.bwindt)
            timedel(l-1)=input.bwindt(l)-input.bwindt(l-1);
        end
    adx1(i).btimedel=timedel;
%     adx1(i).btimedel=input.timedel;
    clear timedel l
end
end

```

6. คอมพิวเตอร์เชิง โมโนดูมนี้สามารถใช้อัลกอริทึมได้ 2 ชนิดคือ การหาค่าเหมาะสมสุดแบบคอนเวกซ์ และการจับคู่สัญญาณบีบอัด

```

%% perform Compressed sensing
for i=1:size(adx1,2)
    bestrank=adx1(i).mproxyrank(1);
    % x t ind
    input.x=cell2mat(adx1(i).windx (bestrank));
    input.t=cell2mat(adx1(i).windt (bestrank));
    input.ind=cell2mat(adx1(i).windt_idc (bestrank));

    adx1(i).bx=input.x;
    adx1(i).bt=input.t;
    adx1(i).bind=input.ind;

    CS.Theta=CS.Psi(input.ind, :);
% COSAMP
% CS.k= 10;
% [Recon(i).xX, recon.proxy(:,i)] =
cosamp(CS.Theta, input.x', CS.k, 1e-10, 10);

% CVX
    Recon(i).proxy=abs(CS.Theta'*input.x);
    cvx_begin;
        variable s(CS.f);
        minimize(norm(s,1));
        subject to
            CS.Theta*s == input.x;
    cvx_end
    Recon(i).xX=s;
End

```

7.วิเคราะห์ผล นำสัญญาณคืนกลับที่สร้างโดยคอมพิวเตอร์ซึ่งแปลงให้อยู่ในรูปแบบต่างๆ ที่สามารถทำการวิเคราะห์ได้เช่น การแปลงฟูเรียร์

```

%% Examine Recon signal
for i=1:size(adx1,2)
    input.xX=(Recon(i).xX);
    input.x=idct(input.xX);
    input.t=(target.t);
    input.srate=(pcb(i).srate);
    [input.FFT,input.Han]=Func_Fourier_Transform
    (input.x,input.srate,length(input.x));
    Recon(i).xXamp=input.FFT.xXamp;
    Recon(i).hz=input.FFT.hz;
    Recon(i).x=input.x;
    Recon(i).t=input.t;
    Recon(i).srate=length(input.t)/max(input.t);
    clear input
end
function [signal]=Func_Fourier_Transform(x,srate,n)
% Normal Fourier Transform
    signal.hz = linspace(0,srate/2,floor(n/2)+1)';
    signal.xX=fft(x)';
    signal.xXamp=abs(signal.xX)/n;
    signal.xXamp(2:length(signal.hz)) =
    2*signal.xXamp(2:length(signal.hz));
    signal.xXPh=angle(signal.xX)*360/(2*pi);
    signal.Frest=max(signal.hz)/(length(signal.hz)-1);
end

```

8.แสดงผล

```

for i=1:length(adx1)
fig=figure();
subplot(2,2,1)
plot(pcb(i).t,pcb(i).x,'-xk','LineWidth',2,'DisplayName',['X_t ',
num2str(pcb(i).srate-1),' Hz']),hold on
plot(adx1(i).bt+adx1(i).bst,adx1(i).bx,'-
xr','LineWidth',1,'DisplayName',['Y_t
',num2str(length(adx1(i).bx)), ' point ']),hold on
xlim([adx1(i).bst+0.5 adx1(i).bst+0.55])
ylim(TDMLIM)
xlabel(['Time (s)'],'FontSize',FNTSIZE)
ylabel(['Amplitude (m/s^2)'],'FontSize',FNTSIZE)
title([' X_t f_{rate} = ',num2str(pcb(i).srate-1),' Hz Y_t
equivalent f_{rate} = ',num2str((1)/min(adx1(i).btimedel)),' Hz
',num2str(winlength),' Sec'],'FontSize',FNTSIZE)
legend('FontSize',10)

subplot(2,2,2)
sub22=stem(pcb(i).hz,
pcb(i).xXamp(1:length(pcb(i).hz)),'k','marker','none');
xlim([0 1600])
ylim(FDMLIM)
xlabel(['Frequency (s)'],'FontSize',FNTSIZE)

```



```

ylabel(['Amplitude (m/s^2)'],'FontSize',FNTSIZE)
title(['X_s tested signal: ',convertStringsToChars(FREQNAME(i))
'],'FontSize',FNTSIZE)
[V,I]=sort(pcb(i).xXamp(1:length(pcb(i).hz)),'descend');
for L=1:1
datatip(sub22,pcb(i).hz(I(L)),pcb(i).xXamp(I(L)));
end
clear L

subplot(223)
plot(Recon(i).t,Recon(i).x,'-xk','LineWidth',2,'DisplayName','Under
sampling'),hold on
xlim([0.5 0.55])
ylim(TDMLIM)
xlabel(['Time (s)'],'FontSize',FNTSIZE)
ylabel(['Amplitude (m/s^2)'],'FontSize',FNTSIZE)
title(['Z_t with f_{rate} = ',num2str(Recon(i).srate), ' Hz
'],'FontSize',FNTSIZE)

subplot(224)
sub24=stem(Recon(i).hz,
Recon(i).xXamp(1:length(Recon(i).hz)),'k','marker','none');
xlim([0 1600])
ylim(FDMLIM)
xlabel(['Frequency (s)'],'FontSize',FNTSIZE)
ylabel(['Amplitude (m/s^2)'],'FontSize',FNTSIZE)
title(['Z_s Reconstruction Signal' ],'FontSize',FNTSIZE)
[V,I]=sort(Recon(i).xXamp(1:length(Recon(i).hz)),'descend');
for L=1:1
datatip(sub24,Recon(i).hz(I(L)),Recon(i).xXamp(I(L)));
end
clear L
    fig.WindowState = 'maximized';
    saveas(fig,sprintf('%s.fig',(FREQNAME(i)))); % will create
FIG1, FIG2,...
    saveas(fig,sprintf('%s.jpeg',(FREQNAME(i)))); % will create
FIG1, FIG2,...
close

end

```

ภาคผนวก ค.

หลักการสุ่มค่าโดยใช้บอร์ดไมโครคอนโทรลเลอร์ STM32F411RE

การควบคุมบอร์ดไมโครคอนโทรลเลอร์ให้สามารถวัดและเก็บข้อมูลได้ในเวลาที่กำหนดอย่างถูกต้อง และสามารถส่งจุดของเวลาหรือ time stamp ของจุดข้อมูลจุดนั้นๆ ให้มีความคลาดเคลื่อนต่ำจำเป็นต้องทราบหลักการในการนับเวลา (Timer and Counter) ในบอร์ดไมโครคอนโทรลเลอร์

โดยปกติบอร์ดไมโครคอนโทรลเลอร์ทุกชนิดจะมีคริสตัล หรือ Oscillator ซึ่งเป็นตัวกำหนดความถี่สัญญาณนาฬิกาสำหรับบอร์ดไมโครคอนโทรลเลอร์เพื่อให้สามารถทำงานหรือ execute คำสั่งในเวลาที่กำหนด สำหรับบอร์ด STM32F411RE มีค่าความถี่อยู่ที่ 16 ล้านเฮิรตซ์ หมายความว่า โดยปกติแล้ว ใน 1 วินาทีจะมีพัลส์ของแรงดันไฟฟ้า 16 ล้านพัลส์ต่อวินาที การที่ไมโครคอนโทรลเลอร์จะทราบเวลาตอนนี้เป็นเท่าไรก็จะอ้างอิงกับการนับพัลส์ของแรงดันไฟฟ้าที่เกิดขึ้น อย่างไรก็ตามผู้ใช้งานสามารถปรับตั้งจำนวนพัลส์ต่อวินาทีได้โดยใช้พรีสเกลเลอร์ (prescaler) มาช่วยในการลดจำนวนครั้งในการนับสัญญาณพัลส์ให้มีค่าน้อยลงแบบทวีคูณตามที่ต้องการ โดยบอร์ด STM32F411RE จะมี พรีสเกลเลอร์ขนาด 16 บิตซึ่งสามารถทำพัลส์ได้ต่ำที่สุด 24.41 พัลส์ ต่อวินาที หรือ 24.41 เฮิรตซ์

12.4.11 TIM1 prescaler (TIMx_PSC)

Address offset: 0x28

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:0 PSC[15:0]: Prescaler value

The counter clock frequency (CK_CNT) is equal to $f_{CK_PSC} / (PSC[15:0] + 1)$.

PSC contains the value to be loaded in the active prescaler register at each update event (including when the counter is cleared through UG bit of TIMx_EGR register or through trigger controller when configured in "reset mode").

รูปที่ ค.1 คำอธิบายการควบคุมพรีสเกลเลอร์รีจิสเตอร์จากหนังสือคู่มือของบอร์ดไมโครคอนโทรลเลอร์

STM32F411RE

ปกติผู้ใช้งานสามารถจะตั้งค่าบอร์ดไมโครคอนโทรลเลอร์ได้ว่าจะนับพัลส์เป็นจำนวนกี่ครั้งต่อวินาที ด้วยการเขียนโปรแกรมควบคุม พรีสเกลเลอร์รีจิสเตอร์ให้เป็นไปตามค่าที่กำหนดโดยมีสูตรดังนี้

$$Pulse\ per\ sec = \frac{f_{ckpsc}}{PSC+1} \quad (ค.1)$$

โดยที่ f_{ckpsc} คือ ความถี่นาฬิกาของบอร์ดไมโครคอนโทรลเลอร์ซึ่งมีค่า 16 ล้าน เฮิรตซ์

PSC คือ ค่าพรีสเกลเลอร์สามารถเลือกได้ตั้งแต่ 0 ถึง 65536 เป็นจำนวนเต็ม

จากสูตรข้างต้นจะเห็นได้ว่าในกรณีที่ค่าพรีสเกลเลอร์เท่ากับ 0 พัลส์ที่เกิดขึ้นจะมีจำนวน 16 ล้านครั้งต่อวินาทีเท่าเดิมและในกรณีที่ค่าพรีสเกลเลอร์เท่ากับ 65536 พัลส์ที่เกิดขึ้นจะมีค่า 244 พัลส์ต่อวินาที

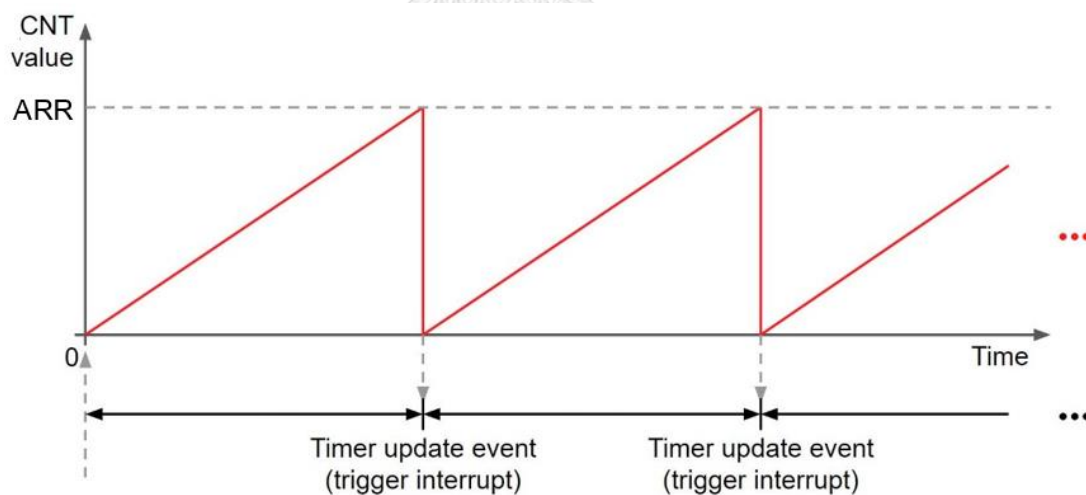
อย่างไรก็ตามการนับจำนวนพัลส์ของบอร์ด STM32F411RE จะถูกนำไปเก็บไว้ที่ เคาน์เตอร์รีจิสเตอร์ (Counter Register) ซึ่งมีขนาด 16 บิตซึ่งสามารถเก็บเลขฐานสิบได้ค่าสูงสุดเพียง 2^{16} หรือ 65536 เท่านั้น ส่งผลให้ในกรณีที่จำนวนพัลส์เกินกว่า 65536 พัลส์ต่อวินาที จะเกิดการล้น (Overflow) ทำให้ เคาน์เตอร์รีจิสเตอร์ (Counter Register) ปรับค่ากลับเป็น 0 ดังเดิม ส่งผลให้จำเป็นต้องตั้งค่าพรีสเกลเลอร์ให้เหมาะสมกับความละเอียดของการจับเวลาหรือการอินเทอร์รัพท์ที่ต้องการ ไม่มากเกินไปจนส่งผลให้ความกว้างพัลส์มากจนไม่สามารถอินเทอร์รัพท์ในเวลาที่กำหนด หรือ น้อยเกินไปจนส่งผลให้จำนวนพัลส์มากเกินไป จนเคาน์เตอร์รีจิสเตอร์ (Counter Register) นับได้ไม่หมด

การอินเทอร์รัพท์ (Interrupt)

การอินเทอร์รัพท์คือการขัดจังหวะการทำงานของโปรแกรมเพื่อให้หยุดการทำงานชั่วคราว จากนั้นให้กระโดดไปทำงานในชุดคำสั่งที่ถูกกำหนดไว้สำหรับการอินเทอร์รัพท์ ซึ่งการอินเทอร์รัพท์สามารถควบคุมได้หลายวิธี เช่น การกดปุ่มให้เกิด อินเทอร์รัพท์ การรับสัญญาณบางอย่างจากอุปกรณ์เพื่อให้เกิดอินเทอร์รัพท์ หรือ การตั้งเวลาให้เกิดการอินเทอร์รัพท์ในเวลาที่กำหนด

การวัดค่าจากเซ็นเซอร์ในกรณีที่ต้องการความแม่นยำของระยะเวลาในการเก็บข้อมูลก็สามารถทำได้ด้วยวิธีนี้เช่นกัน ซึ่งจะนำมาใช้ในงานวิจัยนี้ จะควบคุมให้เกิดการอินเทอร์รัพท์ตามเวลาที่กำหนด

การกำหนดการอินเทอร์รัพท์จะสามารถกำหนดได้จากการนับจำนวนพัลส์ใน เคาน์เตอร์รีจิสเตอร์ (Counter Register) เมื่อจำนวนครั้งถึงค่าที่กำหนดไว้ ไมโครคอนโทรลเลอร์ ก็จะทำการ อินเทอร์รัพท์ เพื่อวัดสัญญาณที่จุด เวลาที่ต้องการ พร้อมทั้งส่งค่าเวลา ขณะทำการวัดเข้าสู่ คอมพิวเตอร์ต่อไป



รูปที่ ค.2 แผนผังการอินเทอร์รัพท์

รูปที่ ค.2 แผนผังการอินเทอร์รัพท์ โดยแกนตั้งเป็นจำนวนพัลส์ที่เคาน์เตอร์นับได้ ส่วนแกนนอนเป็นเวลา ค่า ARR ในกราฟคือจำนวนพัลส์สูงสุดต่อ 1 รอบของการ overflow 1 ครั้ง จากรูป ค.2 จะเห็นว่าเมื่อเวลาผ่านไปค่าเคาน์เตอร์ (CNT) เพิ่มขึ้นเรื่อยๆ จากการนับพัลส์ที่เกิดจากความถี่นาฬิกา เมื่อถึงค่าที่กำหนดหรือค่าอัตโนมัติโหลด (Auto reload) เคาน์เตอร์รีจิสเตอร์ที่เก็บค่าจำนวนพัลส์จะทำการ รีเซ็ตกลับมาเหลือ 0 และเริ่ม

อินเตอร์รัพท์และเริ่มทำสิ่งที่ถูกโปรแกรมไว้ ในกรณีนี้คือการอ่านค่าสัญญาณการสั้นสะพานและส่งค่าที่อ่านได้พร้อมจุดเวลา ณ ขณะนั้นให้กับคอมพิวเตอร์ จนจบขั้นตอนหลังจากนั้นจะรอการอินเตอร์รัพท์รอบใหม่

จากที่กล่าวมาข้างต้นพอจะสรุปได้ว่าการควบคุมความถี่ในการสุ่มวัดสัญญาณ จะขึ้นอยู่กับการควบคุมค่า ค่าพรีสเกลเลอร์ เพื่อให้ความกว้างพัลส์และจำนวนพัลส์เหมาะสมแก่การนับและ ควบคุมค่าออโตรีโหลด (Auto reload) เพื่อให้เกิดการอินเตอร์รัพท์ในเวลาหรือความถี่ที่กำหนด

จากสมการ (ค.1) เมื่อนำค่า ออโตรีโหลด (Auto reload) เข้าไปหารในสมการจะได้จำนวนครั้งในการทำอินเตอร์รัพท์ต่อวินาทีตามสูตรดังนี้

$$\text{Interrupt Frequency} = \frac{f_{ckPSC}}{(PSC+1) \times (ARR+1)} \quad (\text{ค.2})$$

สำหรับในกรณีนี้ ต้องการให้ความถี่สำหรับการเก็บข้อมูลเท่ากับ 3200 เฮิรตซ์ สามารถตั้งค่าได้ตามนี้

$$3200 \text{ Hz} = \frac{16 \times 10^6}{(9 + 1) \times (499 + 1)}$$

จากที่กล่าวมาข้างต้นจะเห็นว่าผู้ใช้สามารถเลือกปรับได้ทั้งค่า PSC และ ARR ได้หลากหลายแบบในการอินเตอร์รัพท์ที่ความถี่เดียวกัน การเพิ่มค่าพรีสเกลเลอร์ให้มีความมาก จะทำให้จำนวนพัลส์ต่อวินาทีลดลงและความกว้างพัลส์มีความมากส่งผลให้ความถี่จะเล็กลงในการวัดลดลง อย่างไรก็ตามการลดค่าพรีสเกลเลอร์ให้น้อยมาก ๆ จะให้ความถี่ในการวัดสูงและได้ผลดีในการวัดแบบปกติ แต่สำหรับการวัดแบบสุ่มคาบจะส่งผลให้ พิสัยของความถี่ในการสุ่มสัญญาณมีค่าต่ำเกินไป (จะกล่าวถึงในย่อหน้าถัดไป)

หลักการหรือวิธีการอินเตอร์รัพท์ข้างต้นสามารถนำไปใช้ได้กับการวัดแบบปกติซึ่งมีคาบการวัดเท่าๆกัน และทราบค่าพรีสเกลเลอร์ และ ค่าออโตรีโหลดที่แน่นอนอยู่แล้ว **ดังนั้นในกรณีนี้** ข้อมูลที่ส่งออกไปสู่คอมพิวเตอร์ไม่จำเป็นต้องส่งค่าจุดของเวลาออกไป เพื่อลดภาระการคำนวณและการส่งข้อมูล โดยค่าจุดของเวลาสามารถสร้างเพิ่ม **ในภายหลัง** ด้วยการคำนวณขึ้นมาได้

การวัดแบบสุ่มคาบ

ในหัวข้อก่อนหน้ากล่าวถึงการวัดสัญญาณการสั้นสะพานด้วยคาบที่เท่ากันในทุกๆ รอบของการวัด ในหัวข้อนี้จะกล่าวถึง**การวัด**ในกรณีที่ต้องการให้ คาบการวัดสัญญาณในแต่ละครั้งไม่เท่ากัน

อย่างที่ทราบว่าการวัดแบบปกติจะต้องกำหนดค่าออโตรีโหลดเพื่อให้ เกิดการอินเตอร์รัพท์ในเวลาที่กำหนด ซึ่งค่าออโตรีโหลดจะถูกกำหนดไว้ในโปรแกรมเป็นค่าคงที่ตั้งแต่เริ่มต้น **และ** เพื่อให้การวัดแต่ละครั้งมีระยะคาบไม่เท่ากันในกรณีการวัดแบบสุ่มคาบจำเป็นจะต้อง อัปเดตค่าออโตรีโหลดทุกรอบของการอินเตอร์รัพท์เพื่อให้รอบถัดไปมีค่าไม่เหมือนเดิม การอัปเดตค่าอินเตอร์รัพท์แบบสุ่มในงานวิจัยนี้**ทำโดยเพิ่มฟังก์ชัน** สุ่มตัวเลขที่เรียกว่า Random() ซึ่งจะสุ่มตัวเลขแบบ 8 บิตหรือมีค่าตั้งแต่ 0 ถึง 255 เข้าไปในค่าออโตรีโหลดเดิมเพื่อให้ความถี่ในการสุ่มเปลี่ยนแปลงไปเรื่อยๆ ในแต่ละรอบการอินเตอร์รัพท์ดังสมการ

$$ARR = Base_{CNT} + Random() \quad (ค.3)$$

ในสมการ ค.3 นี้ จำเป็นจะต้องมีค่า $Base_{CNT}$ เพื่อเป็นค่าขอบเขตความถี่สูงสุดที่ทำการวัดได้ เนื่องจาก ARR แปรผกผันหรือยิ่งค่าออสซิลเลเตอร์ไหลดมีค่ามากก็จะทำให้ความถี่สุ่มลดลง

ข้อจำกัดของวิธีนี้คือสามารถสุ่มพิสัยของค่าออสซิลเลเตอร์ไหลด ได้อย่างจำกัด คือสุ่มค่าได้เพียง 0 ถึง 255 พัลส์ หรืออาจกล่าวได้ว่าสุ่มพิสัยได้เพียง 255 ระดับ ซึ่งในกรณีที่ความกว้างของพัลส์น้อย ก็จะทำให้พิสัยของความถี่สุ่มในการสุ่มสัญญาณมีค่าน้อย อย่างไรก็ตามยังสามารถเพิ่มพิสัยของค่าออสซิลเลเตอร์ไหลดได้โดยการเพิ่มจำนวนพจน์ของ $Random()$ เข้าไปเพิ่มในสมการ (ค.3) เช่น ถ้าใช้ฟังก์ชัน $Random()$ เพียง 1 พจน์ ค่าออสซิลเลเตอร์ไหลด ที่สามารถเป็นไปได้มีตั้งแต่ $Base_{CNT}$ ไปจนถึง $Base_{CNT} + 255$ แต่ถ้าใช้ฟังก์ชัน $Random()$ 4 พจน์ ค่า ARR ที่สามารถเป็นไปได้มีตั้งแต่ $Base_{CNT}$ ไปจนถึง $Base_{CNT} + 255 + 255 + 255 + 255 = Base_{CNT} + 1020$ ทำให้มีพิสัยการสุ่ม ($frange$) กว้างเพียงพอซึ่งจะทำให้สัญญาณที่ได้ (y_t) มีความไม่เป็น periodic หรือความคงที่ของค่าน้อย จะส่งผลให้สามารถสร้างสัญญาณคืนกลับอย่างถูกต้อง และเมื่อทำการเพิ่มพจน์ $Random()$ เข้าไปในสมการ (ค.3) ความกว้างพัลส์จะสามารถคำนวณได้ตามตัวอย่างการคำนวณด้านล่าง

$$Pulse\ Width = \frac{(PSC+1) \times (ARR+1)}{f_{ckPSC}} \quad (ค.4)$$

เมื่อกำหนดความกว้างพัลส์ด้วยการใช้ $Random()$ 1 พจน์ จะได้พิสัยการสุ่มของความถี่สุ่ม หรือ $frange$ ตามตาราง ค.1

ตารางที่ ค.1 เปรียบเทียบพิสัยของความถี่สุ่มในกรณีความกว้างพัลส์ ระยะต่างๆ

Pulse Width (μ S)	PSC	ARR	Total data point per Sec			$frange$
			min	mean	max	
0.06	1	10000	1,560.21	1,579.78	1,600.00	39.79
0.31	5	2,000	1,419.07	1,503.76	1,600.00	180.93
0.63	10	1000	1,274.90	1,418.44	1,600.00	325.10
1.25	20	500	1,059.60	1,273.89	1,600.00	540.40
5	80	125	526.32	790.51	1600.00	1073.68
10	160	62	315.46	526.32	1612.90	1297.45

ตารางที่ ค.1 จากตารางที่ แสดงผลการคำนวณค่าพารามิเตอร์ของการอินเตอร์พท์ต่างๆเมื่อตั้งค่า พรีสเกลตามที่กำหนด และค่าออสซิลเลเตอร์ตั้งต้นก่อนจะนำไปรวมกับตัวเลขสุ่มของฟังก์ชัน $Random()$ เมื่อทำการปรับค่า พรีสเกลในขนาดต่างๆ การที่จะควบคุมให้ความถี่การสุ่มสัญญาณสูงสุดที่ต้องการ เท่ากับ 1600 เฮิรตซ์คงที่ จำเป็นจะต้องปรับค่าออสซิลเลเตอร์ ให้มีค่าสวนทางกัน ซึ่งถ้าหากใช้จำนวนพจน์ของฟังก์ชัน

$Random()$ เท่าเดิมก็จะทำให้พิสัยการสุ่มลดลงตามค่าพีริสเกด ที่ลดลงด้วยถ้า พิสัยการสุ่มลดลงมาก สัญญาณ y_t จะมีความเป็น Periodic สูงขึ้น จนไม่เหมาะจะใช้งานกับคอมพิวเตอร์ซึ่งได้ แต่ถ้าหากค่าพีริสเกด สูงขึ้น ก็จำเป็นต้องปรับให้ค่าอัตราไหลลดลงเข้าใกล้ค่า 0 ส่งผลให้พิสัยการสุ่มกว้างมากจนเกินไปจนทำให้จุดข้อมูลต่อวินาทีลดลงมาก จำนวนข้อมูลสัญญาณ y_t จึงอาจน้อยเกินไปซึ่งจะทำให้คุณภาพของสัญญาณคืนกลับลดลง (สังเกตได้จากจำนวนจุดต่อวินาทีเฉลี่ย)

ตารางที่ ค.2 เปรียบเทียบพิสัยของความถี่สุ่มในกรณีเพิ่มจำนวนพจน์ของ ฟังก์ชัน $Random()$

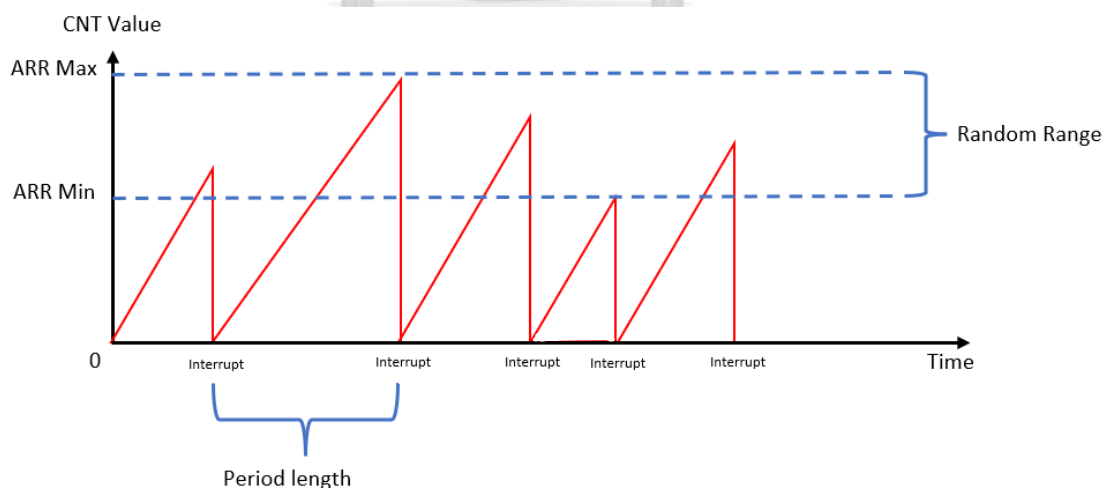
PSC	Base_CNT	Min ARR	Max ARR	Random()	$N_{y,1}$			f_{range}
					min	mean	max	
10	1,000	1,000	1255	1	1,274.90	1,418.44	1,600.00	325.10
10	1,000	1,000	1510	2	1,059.60	1,273.89	1,600.00	540.40
10	1,000	1,000	1765	3	906.52	1,156.07	1,600.00	693.48
10	1,000	1,000	2020	4	792.08	1,058.20	1,600.00	807.92

จากตาราง ค.2 จะเห็นว่าในกรณีที่กำหนดให้ความถี่การสุ่มสัญญาณสูงสุดที่ต้องการ คือ 1600 เฮิรตซ์ เท่ากันทั้ง 4 แบบ แต่การเพิ่มจำนวนพจน์ของฟังก์ชัน $Random()$ ในสมการที่ (ค.3) ส่งผลให้ค่าอัตราไหล ที่เป็นไปได้มีช่วงกว้างมากขึ้น

$$f_{range} = \frac{f_{ckPSC}}{(PSC+1) \times (ARR_{Max}+1)} - \frac{f_{ckPSC}}{(PSC+1) \times (ARR_{Min}+1)} \quad (ค.5)$$

ARR_{Max} คือ ค่าอัตราไหลสูงสุดที่สามารถจะสุ่มได้จากสมการ ค.3

ARR_{Min} คือ ค่าอัตราไหลต่ำสุดที่สามารถจะสุ่มได้จากสมการ ค.3



รูปที่ ค.3 แผนผังการอินเตอร์รัพท์การวัดสัญญาณแบบสุ่มซึ่งทำการอัปเดตค่าอัตราไหลตลอดทุกครั้งที่ทำการวัดทำให้ค่า y_t มีระยะห่างระหว่างจุดไม่เท่ากันเป็นแบบสุ่ม

จากรูปที่ ค.3 จะเห็นว่าการปรับค่าอัตราไหลส่งผลให้ระยะห่างระหว่างจุดข้อมูลเปลี่ยนแปลงไปตลอดทุกครั้งที่ทำการวัดทำให้ค่า y_t มีระยะห่างระหว่างจุดไม่เท่ากันเป็นแบบสุ่ม

เมื่อสามารถวัดค่า y_t ได้แล้วจำเป็นต้องส่งค่าจุดเวลา t_y กลับไปควบคุมกัน เพื่อที่จะลดภาวะและลดเวลาในการคำนวณจากตัวบอร์คไม่ใครคอนโทรลเลอร์ค่าของ t_y ที่ส่งออกมาจะส่งเป็นค่าจำนวนพัลส์ที่เก็บอยู่ใน เคาน์เตอร์รีจิสเตอร์ หลังจากนั้นก็จะทำการแปลงกลับมาเป็นหน่วยวินาทีด้วยโปรแกรม MATLAB ในคอมพิวเตอร์ ซึ่งจะมีสูตรคำนวณดังนี้

$$T_n = \left[(ARR) \times (n - 1) + \sum_{i=1}^{n-1} CNT_{rand,i} + CNT_n \right] \times \frac{PSC}{f_{ckPSC}} \quad (ค.6)$$

โดยที่ T_n คือ เวลา ณ จุดข้อมูลตัวที่ n โดยที่ $t_y = [T_1, T_2, T_3, \dots, T_N]$
 $CNT_{rand,i}$ คือ ค่าเคาน์เตอร์แบบสุ่มในรอบของจุดข้อมูลจุดที่ i ได้จากการวัดค่าฟังก์ชัน $Random()$ ณ จุดการวัดในรอบนั้นๆ
 CNT_n คือ ค่าเคาน์เตอร์ในรอบของจุดข้อมูลจุดที่ i ซึ่งมีค่าใกล้เคียงกับ $Base_{CNT}$

จากสมการ ค.6 จะเห็นว่า พจน์หลัก 3 ของสมการจะถูกแบ่งออกเป็น 3 พจน์ด้วยกันคือ พจน์ของ $(ARR) \times (n - 1)$ ซึ่งจะเป็นตัวบอกระยะเวลาที่ผ่านมาทั้งหมดจนถึงจุดข้อมูลปัจจุบัน $\sum_{i=1}^{n-1} CNT_{rand,i} + CNT_n$ เป็นตัว ให้ระยะเวลาในการยึดหรือหัด จุดข้อมูล ณ จุดนั้นๆ $\frac{PSC}{f_{ckPSC}}$ เป็นค่าความกว้างพัลส์

จากสมการข้างต้นจะเห็นว่ามีความคงที่ที่ถูกกำหนดไว้แล้วตั้งแต่เริ่มต้น 3 ตัวก่อนทำการเริ่มวัด คือ $f_{ck_{PSC}}$, PSC และ $Base_{CNT}$ ซึ่งอยู่ในฟังก์ชันของ ARR สำหรับค่า $CNT_{rand,i}$ คือผลของฟังก์ชัน $Random()$ และค่า CNT_n คือระยะเวลาของการเริ่มอินเตอร์รัพท์จนจบการวัดสัญญาณในแต่ละรอบอินเตอร์รัพท์ ซึ่งมันจะมีค่าใกล้เคียงกันในทุกรอบจะมากหรือน้อยขึ้นอยู่กับค่าของสัญญาณที่ทำการวัด และ เลขของฟังก์ชัน $Random()$ ที่ทำการสุ่มได้

ตารางที่ ค.3 ตัวอย่างข้อมูลที่ถูกส่งเข้ามายังคอมพิวเตอร์ก่อน และหลังทำการแปลงด้วยโปรแกรม MATLAB

CNT_n	CNT_{rand}	$y_t (bit)$	$y_t \left(\frac{m}{s^2}\right)$	$t_y (sec)$
22	39	191	-8.74322	2.75E-05
22	20	62	-18.7065	0.001146
23	144	128	-2.22877	0.002356
22	222	10	-1.99885	0.00354
22	77	157	-15.2577	0.004686
22	90	46	-9.96947	0.005969
22	177	88	0.606933	0.007174
22	69	201	-13.7249	0.008256
22	89	-89	3.136074	0.009475
22	249	86	0.913496	0.010769
23	191	146	-15.411	0.01198
23	21	-26	-2.84189	0.013195
22	155	4	-14.7978	0.014351
22	71	220	-11.4256	0.01564

จากตารางที่ 2.ค จะเห็นว่า ข้อมูลที่ถูกส่งมายังคอมพิวเตอร์ก่อนทำการคำนวณนั้นจะเป็นเลขจำนวนเต็มทั้งหมดเพื่อลดภาระของตัวบอร์ดไมโครคอนโทรลเลอร์ โดย 2 คอลัมน์แรกจะถูกนำไปคำนวณเป็น t_y ตามสมการ (ค.4) สำหรับคอลัมน์ที่ 3 y_t ในหน่วยบิต สัญญาณจะถูกส่งมาเป็นเลขฐาน 10 แบบ 10 บิต ซึ่งจะต้องเอาไปแปลงเป็น y_t ในหน่วย เมตรต่อวินาทียกกำลังสองในคอลัมน์ที่ 4 ซึ่งจะกล่าวถึงต่อไปในภาคผนวก ฉ.

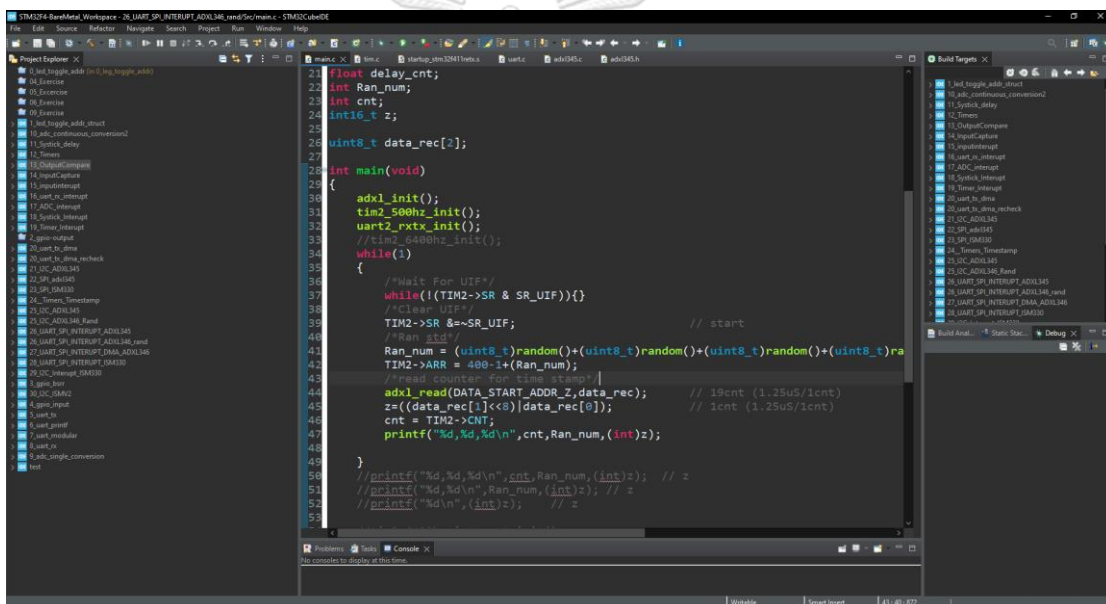
ภาคผนวก ง.

ซอฟต์แวร์ที่ใช้ในการควบคุมการวัดแบบสุ่มคาบของบอร์ดไมโครคอนโทรลเลอร์ STM32F411RE

ระบบการวัดด้วยการใช้บอร์ดไมโครคอนโทรลเลอร์โดยตรง จำเป็นต้องใช้โปรแกรม ที่สำคัญ 2 โปรแกรม คือโปรแกรมที่ใช้ในการเขียนซอฟต์แวร์ควบคุมบอร์ดไมโครคอนโทรลเลอร์ คือ STM32CUDEIDE และโปรแกรมที่ใช้อ่านค่าสัญญาณที่ไมโครคอนโทรลเลอร์ส่งมายังคอมพิวเตอร์ คือ REALTERM

โปรแกรม STM32CUDEIDE

เป็นโปรแกรมหรือแพลตฟอร์มที่ใช้เขียนภาษา C/C++ ที่ถูกสร้างขึ้นมาโดยบริษัท ST เพื่อมาใช้ในการพัฒนาซอฟต์แวร์ที่ใช้ควบคุมบอร์ดไมโครคอนโทรลเลอร์ที่ใช้ชิพประมวลผลในตระกูล ARM บนบอร์ดไมโครคอนโทรลเลอร์ของบริษัท ST โดยเฉพาะ



```

21 float delay_cnt;
22 int Ran_num;
23 int cnt;
24 int16_t z;
25
26 uint8_t data_rec[2];
27
28 int main(void)
29 {
30     adxl_init();
31     tim2_500hz_init();
32     uart2_rxtx_init();
33     //tim2_640hz_init();
34     while(1)
35     {
36         //Wait for UIF
37         while(!((TIM2->SR & SR_UIF)));
38         //Clear UIF
39         TIM2->SR &=~SR_UIF; // start
40         //Ran rnd
41         Ran_num = (uint8_t)random()+(uint8_t)random()+(uint8_t)random()+(uint8_t)random();
42         TIM2->ARR = 400-1+(Ran_num);
43         //read command from register
44         adxl_read(DATA_START_ADDR_Z, data_rec); // 10cnt (1.25uS/1cnt)
45         z=(data_rec[1]<<8)|data_rec[0]; // 1cnt (1.25uS/1cnt)
46         cnt = TIM2->CNT;
47         printf("%d,%d,%d\n", cnt, Ran_num, (int)z);
48     }
49     //printf("%d,%d\n",cnt, Ran_num, (int)z); // z
50     //printf("%d,%d\n", Ran_num, (int)z); // z
51     //printf("%d\n", (int)z); // z
52 }
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

รูปที่ ง.1 แสดงหน้าต่างที่ใช้ทำงานของโปรแกรม STM32CUDEIDE

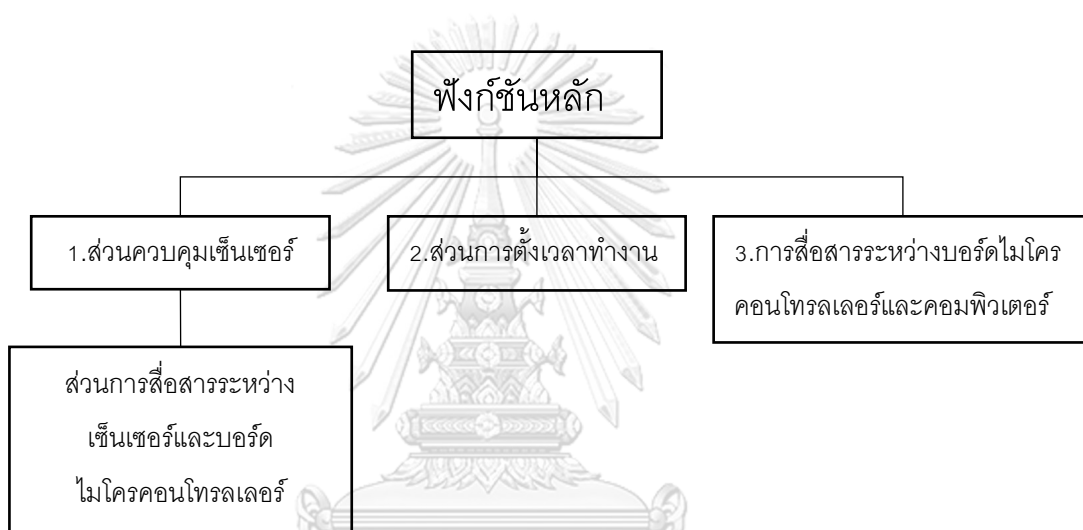
เทคนิคการเขียนซอฟต์แวร์สำหรับการควบคุมบอร์ดไมโครคอนโทรลเลอร์สามารถทำได้หลากหลายเทคนิค ในงานวิจัยนี้จะต้องเขียนซอฟต์แวร์สำหรับการวัดการสั่นสะเทือน และระยะเวลาของข้อมูล เพื่อนำไปใช้ในการแสดงผลในรูปแบบของโดเมนเวลา จุดสำคัญของงานนี้คือความเร็วแล้วความถูกต้องของตำแหน่งเวลาที่วัด ดังนั้น งานนี้จะใช้การเขียนซอฟต์แวร์การวัดสัญญาณในบางจุดจะเลือกใช้เทคนิคที่เรียกว่า Bare-metal Programming ในจุดที่ต้องการความเร็วสูง ซึ่งเป็นส่วนใหญ่ของตัวโปรแกรมทั้งหมด มีเฉพาะบางจุดที่จะเลือกใช้ฟังก์ชันสำเร็จรูป แทนการที่จะต้องเขียนขึ้นมาใหม่ทั้งหมดเองในกรณีถ้าจะใช้แบบ Bare-metal Programming เช่น ฟังก์ชัน Random() ฟังก์ชัน printf()

Bare-Metal Programming

เป็นการเขียนโปรแกรมหรือซอฟต์แวร์ให้ทำงานโดยปราศจากการใช้ไค้ทหรือชุดคำสั่งระดับสูง ที่จะต้องมีกระบวนการการทำงานหลายขั้นซึ่งจะใช้เวลามากในการทำงาน เทคนิคนี้สามารถทำได้โดยการเขียนโปรแกรมให้ควบคุมรีจิสเตอร์หรือหน่วยความจำ ณ ตำแหน่งที่ต้องการโดยตรงโดยไม่ผ่านชุดคำสั่งพื้นฐาน

ไค้ทหรือซอฟต์แวร์ที่ใช้ในการควบคุม

เนื่องจาก ในงานวิจัยนี้จะมุ่งเน้นไปที่การใช้เทคนิค Bare-metal Programming เพื่อความเร็วในการทำงาน ฉะนั้นการเขียนไค้ทจำเป็นต้องหลีกเลี่ยงการใช้ไลบรารีภายนอกในบริเวณที่ไม่จำเป็น โดยชุดของซอฟต์แวร์จะถูกแบ่งออกเป็นหลายส่วนด้วยกันดังนี้



รูปที่ ๒.๒ แสดงโครงสร้างของชุดโปรแกรมในบอร์ดไมโครคอนโทรลเลอร์

ส่วนประกอบของโปรแกรมในแต่ละโมดูล

ฟังก์ชันหลัก

เป็นส่วนหลักของซอฟต์แวร์ซึ่งจะเป็นส่วนหลักที่จะเรียกฟังก์ชันที่จำเป็นขึ้นมาทำงานเป็นลำดับเพื่อให้เกิดการวัดสัญญาณเกิดขึ้น

```

/*GPIOA (AF05) to SPI
 *
 * PA5-->CLK-->CLK
 * PA6-->MISO-->SDO
 * PA7-->MOSI-->SDA
 * PA9-->SS-->CS
 * */
#include <stdio.h>
#include <stdint.h>
#include "stm32f4xx.h"
    
```

```

#include "uart.h"
#include "adx1345.h"
#include "tim.h"
#include "stdlib.h"

#define GPIOAEN      (1U<<0)
#define GPIOA_5      (1U<<5)

#define LED_PIN      GPIOA_5

float delay_cnt;
int Ran_num;
int cnt;
int16_t z;

uint8_t data_rec 2 ;

int main void
{
    adxl_init ;
    tim2_500hz_init ;
    uart2_rxtx_init ;
    //tim2_6400hz_init();
    while 1
    {
        /*Wait For UIF*/
        while ! TIM2->SR & SR_UIF {}
        /*Clear UIF*/
        TIM2->SR &= ~SR_UIF;
        // start
        /*Ran_std*/
        Ran_num =
            uint8_t random() + uint8_t random() + uint8_t random() + uint8_t r
            andom(); //2cnt (1.25uS/1cnt)
        + (uint8_t) random() + (uint8_t) random()
            //Ran_num = (uint8_t) random(); //2cnt
            (1.25uS/1cnt) + (uint8_t) random() + (uint8_t) random()
            TIM2->ARR = 400 - 1 + Ran_num ;
            /*read counter for time stamp*/
            adxl_read DATA_START_ADDR_Z, data_rec ; //
            19cnt (1.25uS/1cnt)
            z = ( data_rec 1 << 8 | data_rec 0 ); //
            1cnt (1.25uS/1cnt)
            cnt = TIM2->CNT;
            printf "%d,%d,%d\n", cnt, Ran_num, int z ; // z
    }
}

```

นอกเหนือจากส่วนของฟังก์ชันหลัก จะสามารถแยกย่อยออกไปได้อีก 3 ส่วนดังนี้

1. ส่วนควบคุมเซ็นเซอร์

สามารถแบ่งย่อยไปอีกตาม โปรโตคอลในการสื่อสารระหว่าง เซ็นเซอร์และบอร์ด ไมโครคอนโทรลเลอร์โดยโปรโตคอลที่เลือกใช้ในงานวิจัยนี้คือ โปรโตคอลแบบ SPI

1.1 ส่วนควบคุมเซ็นเซอร์

```
#include "adxl345.h"
#include "stdint.h"

#define MULTI_BYTE_EN 0x40
#define READ_OPERATION 0x80

void adxl_read(uint8_t address, uint8_t *rxdata)
{
    /*Set read operation*/
    address |= READ_OPERATION;
    /*enable multi-byte*/
    address |= MULTI_BYTE_EN;

    /*Pull cs line low to enable slave*/
    cs_enable();

    /*Send address*/
    spil_transmit(&address, 1);

    /*Read 6 bytes */
    spil_receive(rxdata, 2);

    /*Pull cs line high to disable slave*/
    cs_disable();
}

void adxl_write(uint8_t address, char value)
{
    uint8_t data[2];
    /*Enable Multi-byte, place address into buffer*/
    data[0] = address | MULTI_BYTE_EN;
    /*place data into buffer*/
    data[1] = value;

    /*Pull cs line low to enable slave*/
    cs_enable();

    /*Transmit data and address*/
    spil_transmit(data, 2);
    /*Pull cs line high to disable slave*/
    cs_disable();
}
```

```

void adxl_init void
{
    /*Enable SPI GPIO*/
    spi_gpio_init ;
    /*Config SPI*/
    spi_config ;

    /*Set the data format range to +-4g*/
    adxl_write DATA_FORMAT_R, FOUR_G ;
    /*Reset all bits*/
    adxl_write POWER_CTL_R, RESET ;
    /*Configure power control measure bit*/
    adxl_write POWER_CTL_R, SET_MEASURE_B ;
    /*Data Rate Setting*/
    adxl_write BW_RATE, DR_3200 ;
}

```

1.2 ส่วนการสื่อสารระหว่างไมโครคอนโทรลเลอร์และเซ็นเซอร์

```

/*GPIOA (AF05) to SPI
*
* PA5-->CLK
* PA6-->MISO
* PA7-->MOSI
* PA9-->SS
* */

#include "spi.h"

#define SPI1EN (1U<<12)
#define GPIOAEN (1U<<0)

#define SR_TXE (1U<<1)
#define SR_BUSY (1U<<7)
#define SR_RXNE (1U<<0)

void spi_gpio_init void
{
    /*Enable clock access to SPI1 Module*/
    RCC->APB2ENR |= SPI1EN;
    /*Enable clock access to GPIOA*/
    RCC->AHB1ENR |= GPIOAEN;
    /*PA5 to CLK alt func*/
    GPIOA->MODER &= ~ (1U<<10) ;
    GPIOA->MODER |= (1U<<11) ;
    /*PA6 to MISO alt func*/
    GPIOA->MODER &= ~ (1U<<12) ;
    GPIOA->MODER |= (1U<<13) ;
    /*PA7 to MOSI alt func*/
    GPIOA->MODER &= ~ (1U<<14) ;
    GPIOA->MODER |= (1U<<15) ;
    /*PA9 to OUTPUT*/
    GPIOA->MODER |= (1U<<18) ;
}

```

```

GPIOA->MODER &=~ 1U<<19 ;

/*Set PA5,6,7 alternate function type to SPI1*/
/*PA5*/
GPIOA->AFR 0 |= 1U<<20 ;
GPIOA->AFR 0 &=~ 1U<<21 ;
GPIOA->AFR 0 |= 1U<<22 ;
GPIOA->AFR 0 &=~ 1U<<23 ;
/*PA6*/
GPIOA->AFR 0 |= 1U<<24 ;
GPIOA->AFR 0 &=~ 1U<<25 ;
GPIOA->AFR 0 |= 1U<<26 ;
GPIOA->AFR 0 &=~ 1U<<27 ;
/*PA7*/
GPIOA->AFR 0 |= 1U<<28 ;
GPIOA->AFR 0 &=~ 1U<<29 ;
GPIOA->AFR 0 |= 1U<<30 ;
GPIOA->AFR 0 &=~ 1U<<31 ;

void spi_config void

/*Enable clock access to SPI1 Module*/
RCC->APB2ENR |=SPI1EN;

/*fpclk = 16Mhz Set clock to fPCLK/4*/
SPI1->CR1 |= 1U<<3 ;
SPI1->CR1 &=~ 1U<<4 ;
SPI1->CR1 &=~ 1U<<5 ;

/*Set CPOL to 1 and CPHA to 1*/
SPI1->CR1 |= 1U<<0 ;
SPI1->CR1 |= 1U<<1 ;

/*Enable full duplex*/
SPI1->CR1 &=~ 1U<<10 ;

/*Set MSB first*/
SPI1->CR1 &=~ 1U<<7 ;

/*MCU master config*/
SPI1->CR1 |= 1U<<2 ;

/*Data frame format*/
SPI1->CR1 &=~ 1U<<11 ;

/*Software slave management
 * setting SSM=1 SSI=1*/
SPI1->CR1 |= 1U<<8 ;
SPI1->CR1 |= 1U<<9 ;

/*Enable SPI module*/

```

```

        SPI1->CR1 |= 1U<<6 ;
    }

    void spil_transmit(uint8_t*data,uint32_t size)
    {
        uint32_t i=0;
        uint8_t temp;

        while i<size
        {
            /*wait until TXE is set*/
            while ! SPI1->SR & SR_TXE {}

            /*Write the data to the data register*/
            SPI1->DR=data[i];
            i++;
        }
        /*Wait until TXE is set*/
        while ! SPI1->SR & SR_TXE {}

        /*Waite for busy flag to reset*/
        while SPI1->SR & SR_BUSY {}

        /*Clear overflow flag by read access to
        * DR and SR register see Page 599 */
        temp= SPI1->DR;
        temp= SPI1->SR;
    }

    void spil_receive(uint8_t*data,uint32_t size)
    {
        while size
        {
            /*Send dummy data*/
            SPI1->DR=0;
            /*Wait for RXNE flag to be set*/
            while ! SPI1->SR & SR_RXNE {}

            /*Read data from data register*/
            *data++= SPI1->DR ;
            size--;
        }
    }

    void cs_enable void
    {
        /*CS line Low to enable*/
        GPIOA->ODR &=~ 1U<<9 ;
    }

    void cs_disable void
    {
        /*CS line high to disable*/
        GPIOA->ODR |= 1U<<9 ;
    }

```

2. ส่วนควบคุมเวลาในการเก็บข้อมูล

ในส่วนนี้อธิบายหลักการไว้แล้วในภาคผนวก ค. เนื่องจากการปรับตั้ง timer พื้นฐานในแต่ละเวลา มีลักษณะเหมือนกันซึ่งจะเปลี่ยนเฉพาะ ค่า ARR และ CNT เท่านั้นจึงขอยกเฉพาะเวลาดังตัวอย่างที่ขั้บย่อเท่านั้น คือ timer ขนาด 3125 เฮิรตซ์

```
#include "stm32f4xx.h"

#define TIM2EN          (1U<<0)
#define CR1_CEN        (1U<<0)
#define DIER_UIE       (1U<<0)

void tim2_3125hz_init void
{
    /*Enable Clock access to tim2*/
    RCC->APB1ENR |=TIM2EN;
    /*Set prescalar value*/
    TIM2->PSC=20-1; //16 000 000 /1 600 = 10 000
    /*Set Auto Reload value*/
    TIM2->ARR=256-1; //10 000 / 10 000 =1
    /*Clear counter*/
    TIM2->CNT=0;
    /*Enalble timer*/
    TIM2->CR1 |= CR1_CEN;
```

3. ส่วนการสื่อสารระหว่างบอร์ดไมโครคอนโทรลเลอร์และคอมพิวเตอร์

ในงานวิจัยนี้จะใช้การสื่อสารแบบ Serial ซึ่งง่ายแก่การนำไปใช้งานและจะถูกปรับตั้งความเร็ว Baud rate ไว้ที่ 921600 บิตต่อวินาที

```
#include "uart.h"

#define GPIOAEN          (1U<<0)
#define UART2EN          (1U<<17)

#define CR1_TE           (1U<<3)
#define CR1_RE           (1U<<2)
#define CR1_RXNEIE      (1U<<5)

#define CR1_UE           (1U<<13)
#define SR_TXE           (1U<<7)
#define SR_RXNE         (1U<<5)

#define DMA1EN           (1U<<21)
#define CHSEL4           (1U<<27)
#define DMA_MEM_INC      (1U<<10)
#define DMA_DIR_MEM_TO_PERIPH (1U<<6)
#define DMA_CR_TCIE      (1U<<4)
#define DMA_CR_EN        (1U<<0)
#define UART_CR3_DMAT    (1U<<7)
```



```

#define SYS_FREQ      16000000
#define APB1_CLK      SYS_FREQ

#define UART_BAUDRATE  921600

static void uart_set_baudrate USART_TypeDef*USARTx, uint32_tPeriphCLK,
uint32_tBaudRate ;
static uint16_tcompute_uart_bd uint32_tPeriphCLK, uint32_tBaudRate ;
voiduart2_write intch ;

int __io_putchar intch
{
    uart2_write ch ;
    return ch;
}

voiddma1_stream6_init uint32_tsrc, uint32_tdst, uint32_tlen
{
    RCC->AHB1ENR |= DMA1EN;
    DMA1_Stream6->CR    &=~DMA_CR_EN;
    DMA1->HIFCR |= 1U<<16 ;
    DMA1->HIFCR |= 1U<<18 ;
    DMA1->HIFCR |= 1U<<19 ;
    DMA1->HIFCR |= 1U<<20 ;
    DMA1->HIFCR |= 1U<<21 ;
    DMA1_Stream6->PAR=dst;
    DMA1_Stream6->M0AR=src;
    DMA1_Stream6->NDTR=len;
    DMA1_Stream6->CR= CHSEL4;
    DMA1_Stream6->CR |= DMA_MEM_INC;
    DMA1_Stream6->CR |= DMA_DIR_MEM_TO_PERIPH;
    DMA1_Stream6->CR |= DMA_CR_TCIE;
    DMA1_Stream6->FCR=0;
    DMA1_Stream6->CR |= DMA_CR_EN;
    USART2->CR3 |= UART_CR3_DMAT;
    NVIC_EnableIRQ DMA1_Stream6_IRQn ;
}

voiduart2_rxtx_init void
{
    /***Configure Uart gpio pin***/
    /**Enable clock access to gpioa*/
    RCC->AHB1ENR |= GPIOAEN;

    /**Set PA2 to alternate function mode*/
    GPIOA->MODER &=~ 1U<<4 ;
    GPIOA->MODER |= 1U<<5 ;

    /**Set PA2 alteranate function type to UART TX (AF07)*/

```

```

GPIOA->AFR 0 |= 1U<<8 ;
GPIOA->AFR 0 |= 1U<<9 ;
GPIOA->AFR 0 |= 1U<<10 ;
GPIOA->AFR 0 &=~ 1U<<11 ;

/*Set PA3 to alternate function mode*/
GPIOA->MODER &=~ 1U<<6 ;
GPIOA->MODER |= 1U<<7 ;
/*Set PA3 alteranate function type to UART_RX (AF07)*/
GPIOA->AFR 0 |= 1U<<12 ;
GPIOA->AFR 0 |= 1U<<13 ;
GPIOA->AFR 0 |= 1U<<14 ;
GPIOA->AFR 0 &=~ 1U<<15 ;

/****Configure Uart Module****/
/*Enable Clock access to uart2*/
RCC->APB1ENR |= UART2EN;

/*Configure baudrate*/
uart_set_baudrate USART2,APB1_CLK,UART_BAUDRATE ;

/*Congifure Transfer direction*/
USART2->CR1 = CR1_TE|CR1_RE; // Clean everything in UART leave
only Bit 3,2 = 1

/*Enable Uart module*/
USART2->CR1 |= CR1_UE;

void uart2_tx_init void

/****Configure Uart gpio pin****/
/*Enable clock access to gpioa*/
RCC->AHB1ENR |= GPIOAEN;

/*Set PA2 to alternate function mode*/
GPIOA->MODER &=~ 1U<<4 ;
GPIOA->MODER |= 1U<<5 ;

/*Set PA2 alteranate function type to UART_TX (AF07)*/
GPIOA->AFR 1 |= 1U<<8 ;
GPIOA->AFR 1 |= 1U<<9 ;
GPIOA->AFR 1 |= 1U<<10 ;
GPIOA->AFR 1 &=~ 1U<<11 ;

/****Configure Uart Module****/
/*Enable Clock access to uart2*/
RCC->APB1ENR |= UART2EN;

/*Configure baudrate*/
uart_set_baudrate USART2,APB1_CLK,UART_BAUDRATE ;

/*Congifure Transfer direction*/

```

```

USART2->CR1= CR1_TE; // Clean everything in UART leave only Bit
3 = 1

/*Enable Uart module*/
USART2->CR1 |= CR1_UE;

void uart2_rxtx_rxinterrupt_init void

/*****Configure Uart gpio pin****/
/*Enable clock access to gpioa*/
RCC->AHB1ENR |= GPIOAEN;

/*Set PA2 to alternate function mode*/
GPIOA->MODER &=~ 1U<<4 ;
GPIOA->MODER |= 1U<<5 ;

/*Set PA2 alteranate function type to UART_TX (AF07)*/
GPIOA->AFR 0 |= 1U<<8 ;
GPIOA->AFR 0 |= 1U<<9 ;
GPIOA->AFR 0 |= 1U<<10 ;
GPIOA->AFR 0 &=~ 1U<<11 ;

/*Set PA3 to alternate function mode*/
GPIOA->MODER &=~ 1U<<6 ;
GPIOA->MODER |= 1U<<7 ;
/*Set PA3 alteranate function type to UART_RX (AF07)*/
GPIOA->AFR 0 |= 1U<<12 ;
GPIOA->AFR 0 |= 1U<<13 ;
GPIOA->AFR 0 |= 1U<<14 ;
GPIOA->AFR 0 &=~ 1U<<15 ;

/*****Configure Uart Module****/
/*Enable Clock access to uart2*/
RCC->APB1ENR |= UART2EN;

/*Configure baudrate*/
uart_set_baudrate USART2, APB1_CLK, UART_BAUDRATE ;

/*Congifure Transfer direction*/
USART2->CR1= CR1_TE|CR1_RE; // Clean everything in UART leave
only Bit 3,2 = 1
/*Enable USART Interupt*/
USART2->CR1 |= CR1_RXNEIE;
/*Enable UART2 interrupt in NVIC*/
NVIC_EnableIRQ USART2_IRQn ;

/*Enable Uart module*/
USART2->CR1 |= CR1_UE;

char uart2_read void

/*make sure the transmit data register is Not-empty*/

```

```

    while ! USART2->SR & SR_RXNE {}
    /*Read data*/
    return USART2->DR;
}

void uart2_write (int ch)
{
    /*make sure the transmit data register is empty*/
    while ! USART2->SR & SR_TXE {}

    /*write to transmit data register*/
    USART2->DR = (ch & 0xFF);
}

static void uart_set_baudrate USART_TypeDef* USARTx, uint32_t PeriphCLK,
uint32_t BaudRate
{
    USARTx->BRR = compute_uart_bd (PeriphCLK, BaudRate);
}

static uint16_t compute_uart_bd (uint32_t PeriphCLK, uint32_t BaudRate)
{
    return ( PeriphCLK + BaudRate/2U ) / BaudRate ;
}

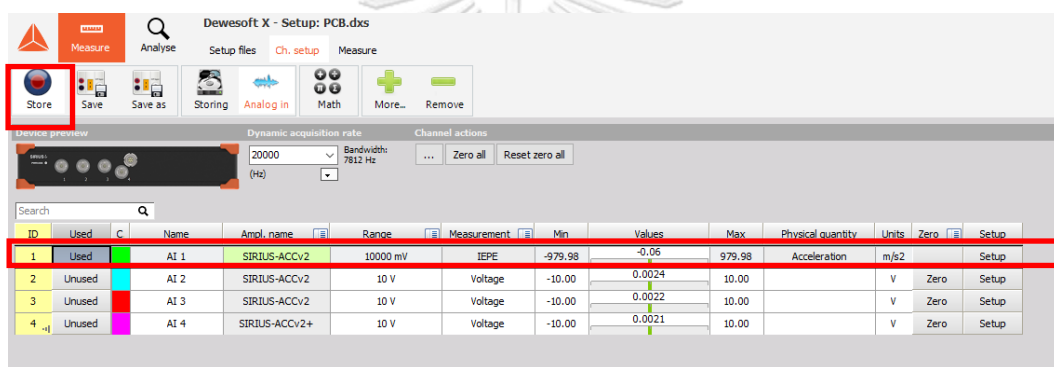
```

ภาคผนวก จ.
การเก็บข้อมูลและปรับปรุงสัญญาณ ของระบบการวัดอ้างอิง
(PCB และ DEWESOFT)

การเก็บข้อมูล

สำหรับการเก็บข้อมูลของ เซ็นเซอร์ PCB จะใช้โปรแกรมที่ชื่อว่า Dewesoft X ซึ่งจะให้มาพร้อมกับ DAQ Dewesoft Sirius โดยวิธีการเก็บข้อมูลเพื่อนำไปใช้มีดังนี้

1. เปิดโปรแกรม Dewesoft X
2. ในกรณีที่ ต่อเซ็นเซอร์ PCB เข้ากับ Dewesoft Sirius เรียบร้อยแล้ว สถานะของขาแนล จะมีสถานะ “Used” ตามรูปที่ จ.1 กรอบสีแดงด้านล่าง
3. กดวัดสัญญาณโดยการกดปุ่ม Store



รูปที่ จ.1 หน้าต่างโปรแกรมเบื้องต้น

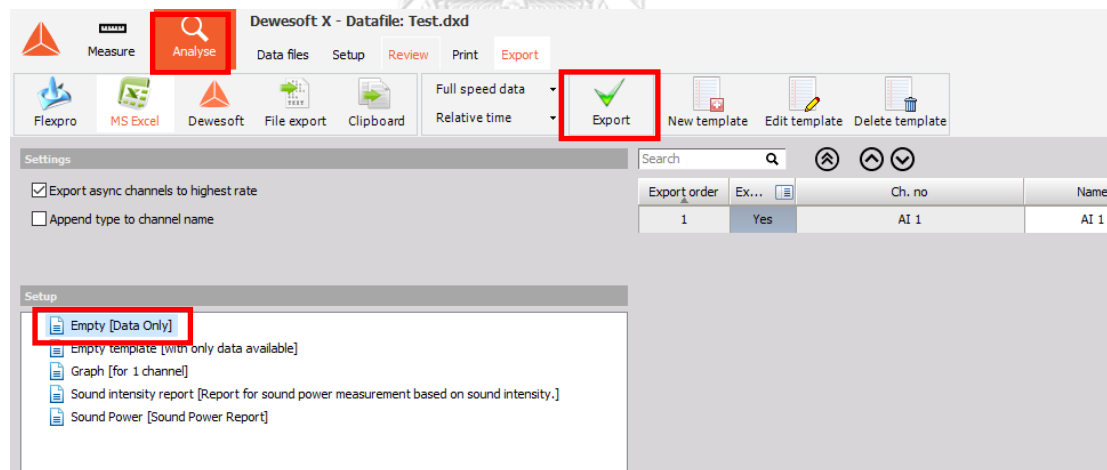
4. เมื่อวัดสัญญาณถึงเวลาที่ต้องการให้กดปุ่ม Store อีกครั้งเพื่อหยุดการบันทึกข้อมูล



รูปที่ ๑.2 หน้าต่างโปรแกรมขณะทำการบันทึกสัญญาณ

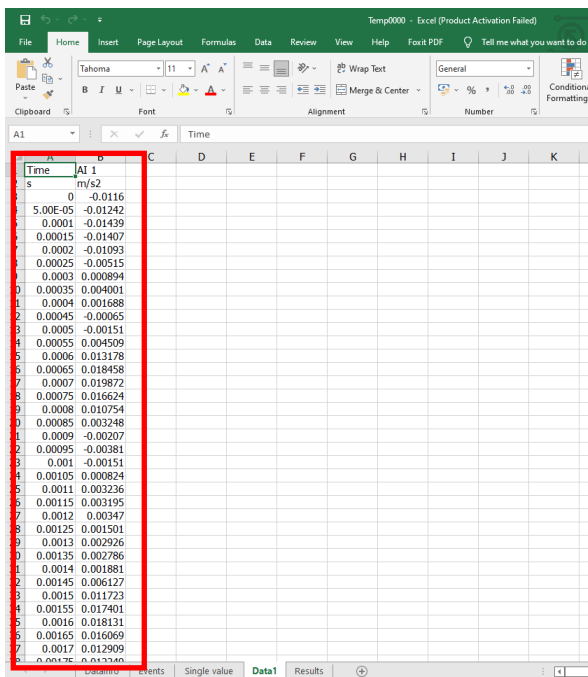
5.เลือกแท็บ Analyse และเลือก Empty [Data Only]

6.กด Export



รูปที่ ๑.3 หน้าต่างโปรแกรมในหัวข้อ Analyse

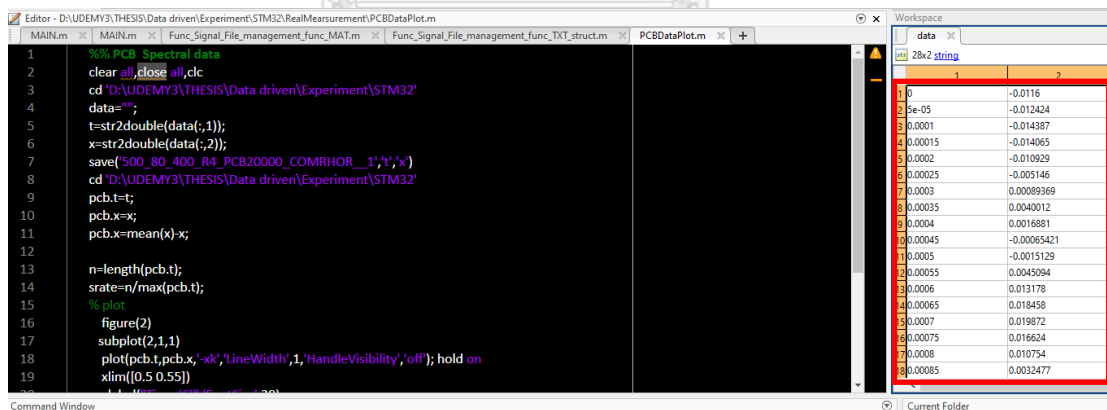
7.Dewesoft จะทำการสร้างไฟล์ Excel ขึ้นมาตามรูปด้านล่างให้ ก้อปปี คอลดัมน์เวลาและสัญญาณไว้ในคลิปบอร์ด



รูปที่ ๑.4 หน้าต่างข้อมูลที่อยู่ในไฟล์ Excel

8.นำไปวางในตัวแปรที่ได้สร้างไว้ใน Matlab

9.เซฟตัวแปรนั้นไว้เพื่อพร้อมใช้งาน



รูปที่ ๑.5 หน้าต่างโปรแกรม Matlab ที่ใช้บรรจุข้อมูลลงในตัวแปร

การปรับปรุงสัญญาณ

สำหรับการวัดสัญญาณโดยใช้ PCB และ DEWESOFT ซึ่งเป็นอุปกรณ์ที่มีคุณภาพสูงอยู่แล้วสัญญาณที่ได้จะต้อง export ออกมาเป็น excel file หลังจากนั้นทำการ Copy cell ใน excel file มาวางในตัวแปรของ matlab ที่เตรียมไว้ ข้อมูลที่มีจะแบ่งเป็น 2 อย่างคือ แอมพลิจูด และ เวลาซึ่งปกติแล้วข้อมูลสามารถนำไปใช้ได้เลยโดยที่ไม่มี error ของ แอมพลิจูดแต่อย่างใด แต่ในกรณีใช้ความถี่สูงสัญญาณ 20000 เฮิร์ตซ์ข้อมูลเวลาที่

มากกว่า 1 วินาทีขึ้นไปจะมี ปัญหาเรื่อง Floating-point error mitigation เมื่อทำการ Copy ข้อมูลเวลาจาก excel file ลงไปในตัวแปรใน matlab ซึ่งส่งผลให้ ข้อมูลเวลาที่เกิน 1 วินาทีจะมีการปัดขึ้นส่งผลให้มีเวลาซ้ำกัน เกิดขึ้นภายในสัญญาน วิธีการแก้ไขคือ ต้อง สร้างข้อมูลเวลาขึ้นมาใหม่ด้วยฟังก์ชัน linspace ใน Matlab ซึ่ง สามารถเขียนได้ดังนี้

เวลาใหม่ที่ได้รับการแก้ไขแล้ว = `linspace(0, เวลาสูงสุดของข้อมูล, จำนวนจุดข้อมูล);`

ตารางที่ จ.1 แสดงข้อมูลเวลาที่มีความผิดพลาดเทียบกับข้อมูลที่แก้ไขแล้วสำหรับความถี่สัญญาณ 20000

เฮิรตซ์

t. Raw	t. Fix
0.999750	0.999750
0.999800	0.999800
0.999850	0.999850
0.999900	0.999900
0.999950	0.999950
1.000000	1.000000
1.000100	1.000050
1.000100	1.000100
1.000200	1.000150
1.000200	1.000200
1.000300	1.000250
1.000300	1.000300
1.000400	1.000350

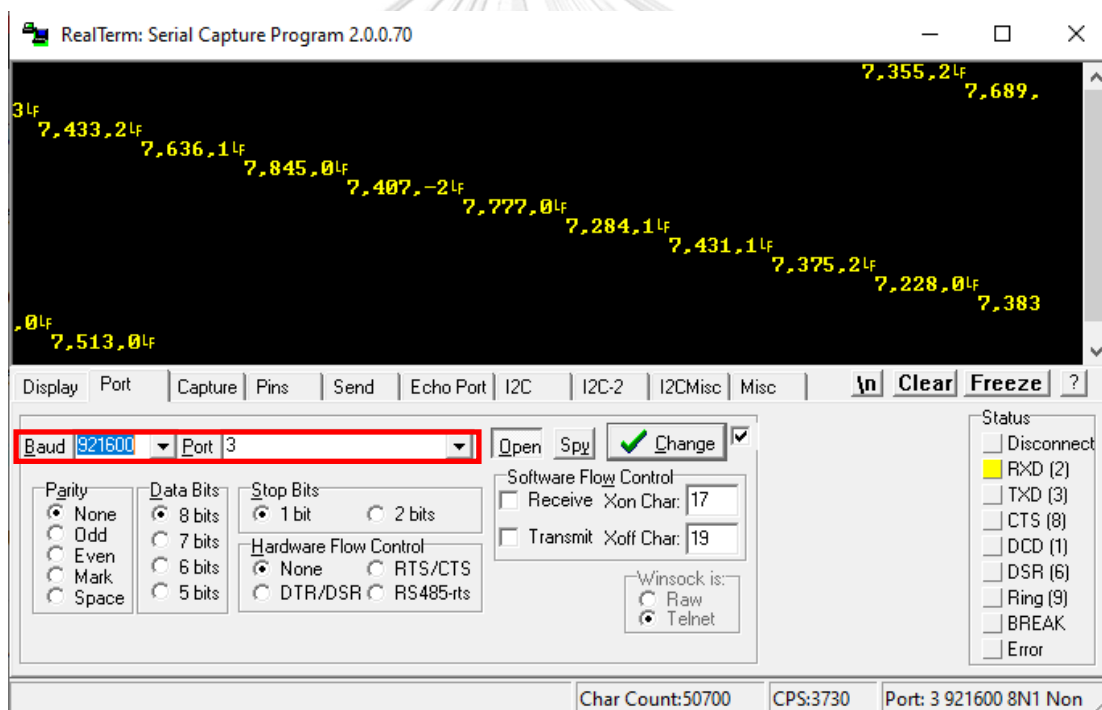
ภาคผนวก จ.

ขั้นตอนการเก็บข้อมูลและปรับปรุงสัญญาณ ของระบบการวัดแบบสุ่มคาบ (ADXL345 และ STM32F411RE)

การเก็บข้อมูล

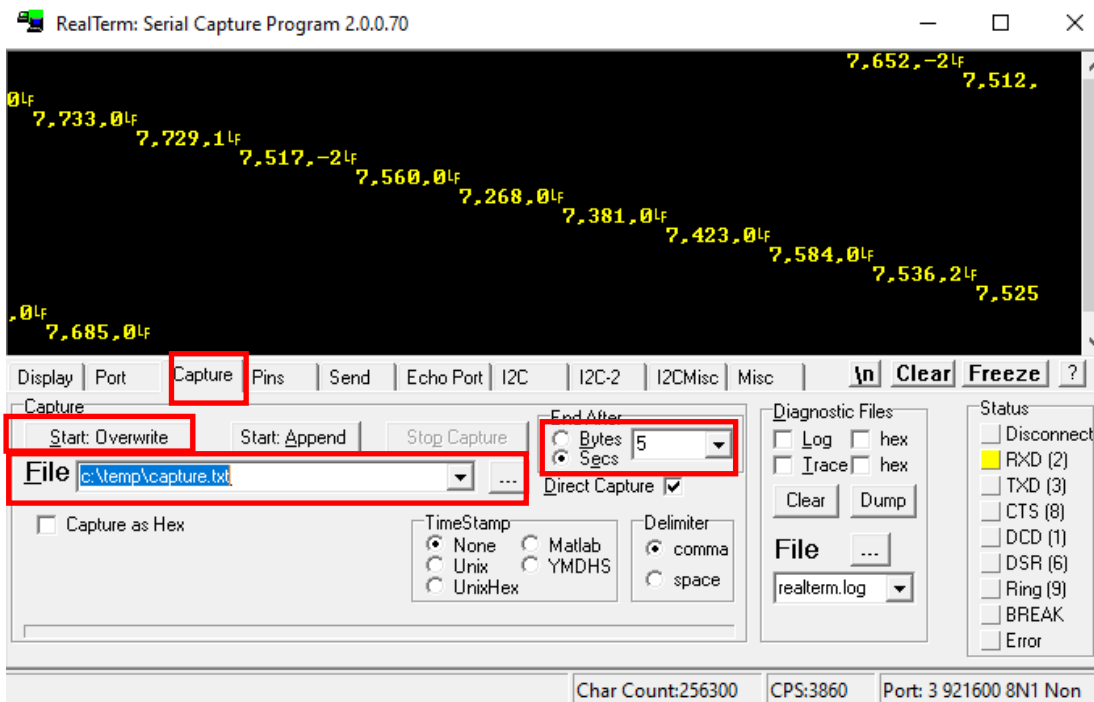
การเก็บข้อมูลโดยใช้ บอร์ดไมโครคอนโทรลเลอร์ STM32F411RE นั้นจะถูกเก็บผ่านการสื่อสารแบบ Serial communication protocols ซึ่งจะเลือกใช้ความเร็วในการส่งข้อมูลที่ 921600 บิตต่อวินาที โปรแกรมที่ใช้บันทึกข้อมูลสัญญาณที่วัดได้ จะเลือกใช้เป็น RealTerm 2.00.70 โดยเมื่ออัปเดตโค้ดในภาคผนวก ข. เพื่อให้ บอร์ดไมโครคอนโทรลเลอร์ทำงานแล้วข้อมูลจะถูกส่งมาที่ โปรแกรม RealTerm เพื่อแสดงผลทันทีดังตัวอย่างในรูปที่ 1.๑ ด้านล่าง การเก็บสัญญาณที่วัดมาได้อาศัยอยู่ในรูปของไฟล์ Notepad จะมีขั้นตอนดังนี้

1. ปรับ Baud rate ให้เท่ากับ 921600 ปรับ Port ที่ใช้ให้ตรงกับไมโครคอนโทรลเลอร์ (ซึ่งคอมพิวเตอร์แต่ละเครื่องจะไม่เหมือนกัน สามารถดูได้ที่ Device manger)



รูปที่ ๑.1 หน้าต่างโปรแกรม RealTerm ตอนเริ่มเข้าโปรแกรม

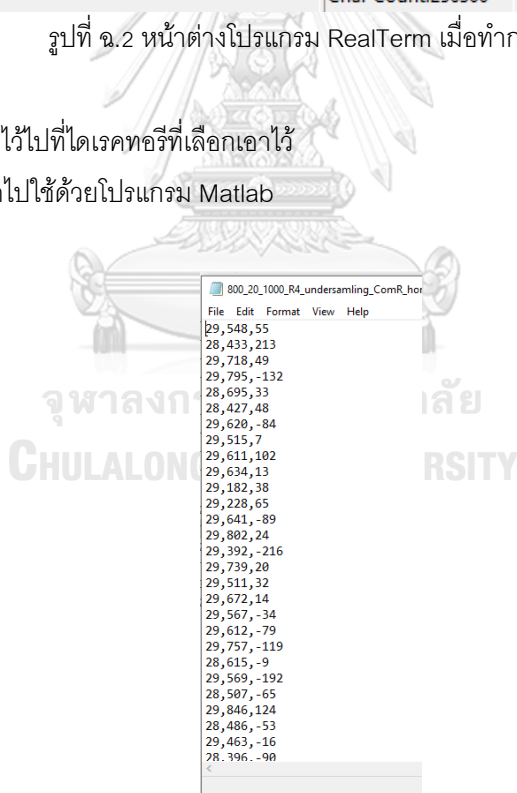
2. เลือก Tab capture ตั้งค่าไดเรกทอรีที่ต้องการจะเซฟไฟล์ และ ตั้งค่าระยะเวลาที่ต้องการจะวัด
3. กด Start Overwrite
4. รอจนโปรแกรมบันทึกค่าสัญญาณเสร็จ



รูปที่ ๓.2 หน้าต่างโปรแกรม RealTerm เมื่อทำการวัด

5.ไฟล์ข้อมูลจะถูกบันทึกไว้ไปที่ไดเรกทอรีที่เลือกเอาไว้

6.พร้อมสำหรับการโหลดไปใช้ด้วยโปรแกรม Matlab



รูปที่ ๓.3 ตัวอย่างข้อมูลที่ทำการบันทึกจาก RealTerm

จากรูปที่ 3.๓ จะเห็นว่าข้อมูลที่อยู่ใน Text file จะมีข้อมูลแต่ละบรรทัดประกอบด้วย 3 คอลัมน์ที่ถูกแบ่งด้วย จุดลูกน้ำ (.) โดยคอลัมน์แรกจะเป็นค่า CNT_n คอลัมน์ที่สองคือ CNT_{rand} คอลัมน์ที่สามคือ y_t

ในหน่วยของจำนวนบิตซึ่งจะถูกนำไปแปลงเป็น เวลาและแอมพลิจูดต่อไปตามที่ได้กล่าวไว้แล้วในภาคผนวก ค.

การปรับปรุงสัญญาณ

เนื่องจาก เซ็นเซอร์ ADXL345 และ บอร์ดไมโครคอนโทรลเลอร์ STM32F411RE เป็นระบบการวัดที่ถูกประกอบขึ้นและโปรแกรมคำสั่งถูกเขียนขึ้นสำหรับงานวิจัยนี้โดยเฉพาะ ซึ่งความเร็วในการประมวลผลมีความสำคัญ ฉะนั้นข้อมูลที่ถูกส่งออกมาทั้งหมดทั้งใน ข้อมูลเวลา และแอมพลิจูดกำหนดให้อยู่ในรูปของ จำนวนพัลส์ และ ตัวเลขแบบ 10 บิตแบบมีเครื่องหมายหรือ -512 ถึง 512 ซึ่ง สามารถดูตัวอย่างของข้อมูล,วิธีการแปลงเวลาได้ในภาคผนวก ค. สำหรับแอมพลิจูด จะต้องนำค่าจากไฟล์ Notepad มาแปลงดังนี้

$$y_t \left(\text{ในหน่วย } \frac{m}{s^2} \right) = y_t(\text{จากไฟล์ notepad}) \times \frac{g \times 9.81}{512}$$

โดยที่ g คือ พิสัยของแอมพลิจูดที่ทำการวัด ซึ่งได้จากการตั้งค่าในตัวเซ็นเซอร์ผ่านบอร์ดไมโครคอนโทรลเลอร์ มีหน่วยเป็น จำนวนเท่าของแรงโน้มถ่วงโลก โดยในที่นี้เลือกใช้พิสัย ± 4 เท่าของแรงโน้มถ่วงโลก ฉะนั้น g ในที่นี้จะเท่ากับ 4

ในภาคผนวกนี้ จะกล่าวถึงความผิดพลาดของข้อมูลที่ถูกส่งมาและวิธีแก้ไข ซึ่งโดยปกติจะมีลักษณะดังนี้

- จุดข้อมูล error หรือเสียหายซึ่งจะถูกแสดงผลใน matlab เป็น NaN

ตารางที่ ๑.1 ตัวอย่างข้อมูลที่ error

CNT_n	CNT_{rand}	y_t
22	39	191
22	90	46
22	177	88
22	69	NaN
22	89	-89
22	249	86

วิธีการแก้ไขจะทำโดยการแทนจุดที่เสียหาย ด้วยข้อมูลที่เป็นค่าเฉลี่ยระหว่างจุดก่อนหน้าและจุดถัดไปใส่เข้าไปแทน ซึ่งจะสามารถเขียนได้ในโปรแกรม Matlab ดังนี้

```
% corrupted repair
b=isnan(data); % Create Nan Matrix
for k=1:size(b,1) % Loop Through Nan Matrix
    if find(b(k,:)==1)==1; % Find NAN row
        if k==1||k==size(b,1) % Check if its an
edge of array
            data(k,:)=[]; % if yes remove edge
        else
```

```

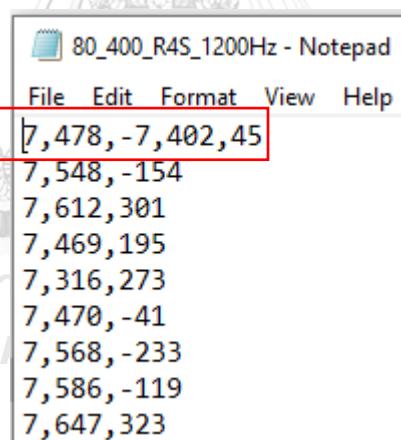
data(k,:)=floor((data(k+1,:)+data(k-
1,:))/2); % if replace with mean of upper lower
end
end
end
end

```

ตารางที่ ๑.2 ตัวอย่างข้อมูลที่ถูกแก้ไขแล้ว

CNT_n	CNT_{rand}	y_t
22	39	191
22	90	46
22	177	88
22	69	-1
22	89	-89
22	249	86

2. ข้อมูลที่ไม่สมบูรณ์ สำหรับจุดแรกและจุดสุดท้ายของไฟล์สัญญาณเนื่องจากมีข้อมูลบางส่วนจากการวัดก่อนหน้าค้างอยู่ในหน่วยความจำของไมโครคอนโทรลเลอร์



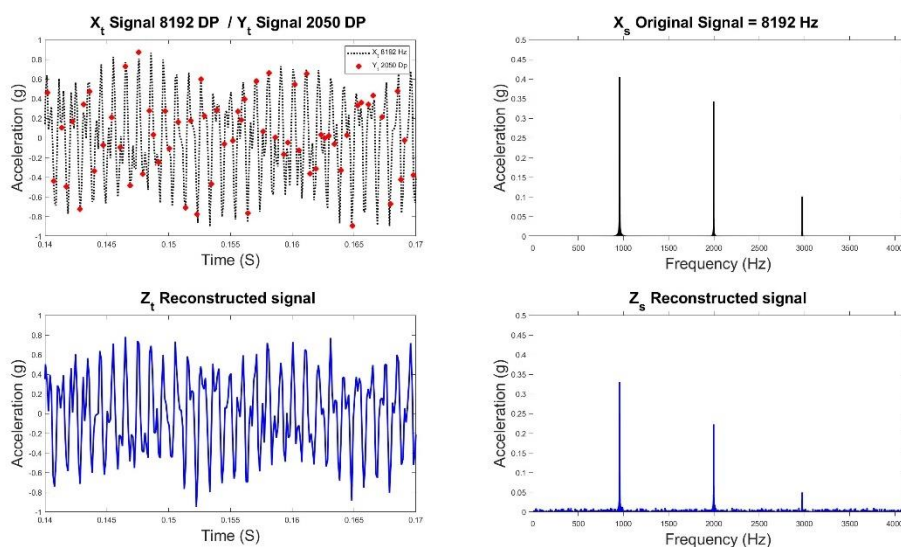
รูปที่ ๑.4 แสดงข้อมูลจุดแรกที่ผิดพลาดซึ่งมี 5 คอลัมน์ซึ่งปกติจะมีเพียง 3 คอลัมน์

ปกติแล้วข้อมูลลักษณะนี้จะเปิดขึ้นกับข้อมูลชุดแถวแรกและแถวสุดท้าย สำหรับจุดนี้สามารถแก้ไขได้โดยลบข้อมูลแถวที่ทิ้งไปทั้งหมด โดยเลือกลบด้วยตัวชี้เองหรือจะใช้โปรแกรม MATLAB สแกนหาบรรทัดของข้อมูลที่มีคอลัมน์เกินกว่า 3 แล้วลบทิ้งก็ได้

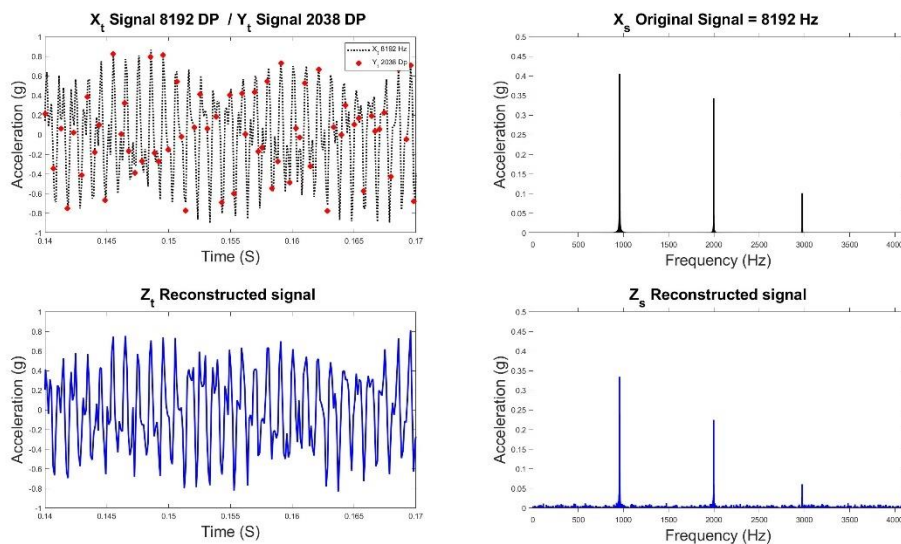
ภาคผนวก ซ.

ผลการทดสอบการจำลองสัญญาณและทดสอบอัลกอริทึมคอมเพรสชัน

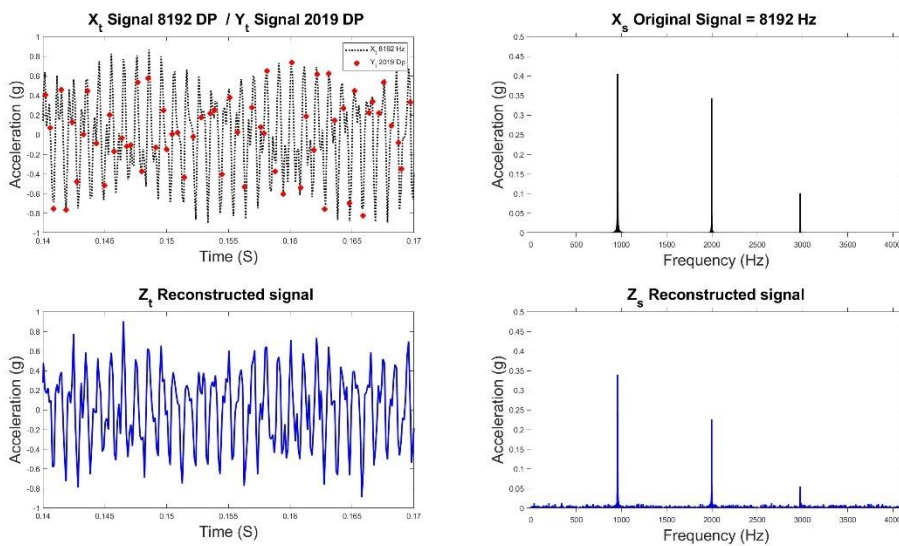
ผลการทดสอบในการจำลองสัญญาณจะมีทั้งหมด 15 ผลการทดสอบ จะแบ่งผลการทดสอบออกเป็น 3 เครื่องจักร คือสัญญาณของ มอเตอร์, ปีม และชุดเกียร์ ทำการทดสอบเครื่องจักรละ 5 ครั้ง โดยจะแสดงผลในลักษณะของรูป 4 รูป โดยรูปด้านซ้ายบนจะแสดงสัญญาณใน โดเมนเวลาทั้งของ X_t และ Y_t รูปด้านขวาบนจะแสดงสัญญาณ X_s รูปด้านซ้ายล่างแสดงสัญญาณ Z_t รูปด้านขวาล่างแสดงสัญญาณ Z_s โดยปกติวิธีการตรวจสอบก็จะเปรียบเทียบสัญญาณระหว่าง X_s และ Z_s , X_t และ Z_t ว่ามีความคล้ายคลึงกันหรือไม่



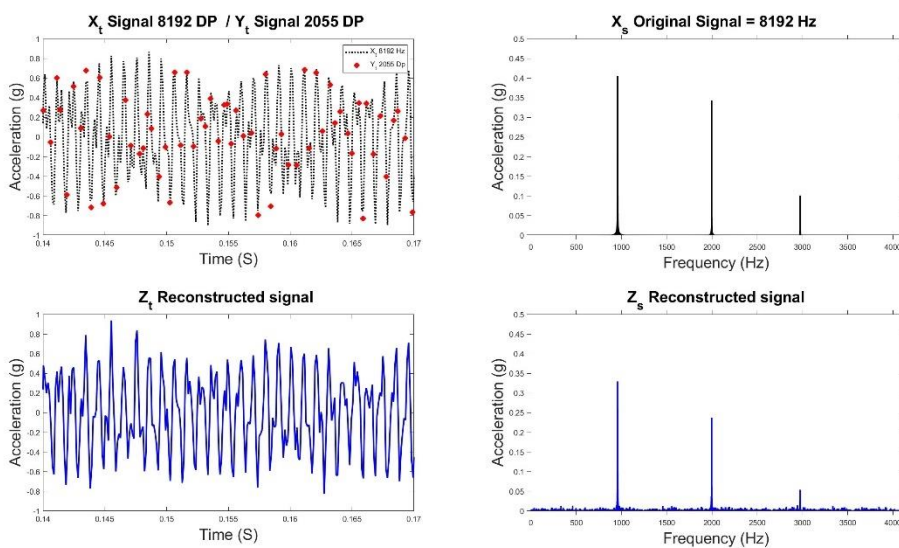
รูปที่ 1.1 ผลลัพธ์ของสัญญาณมอเตอร์ ครั้งที่ 1



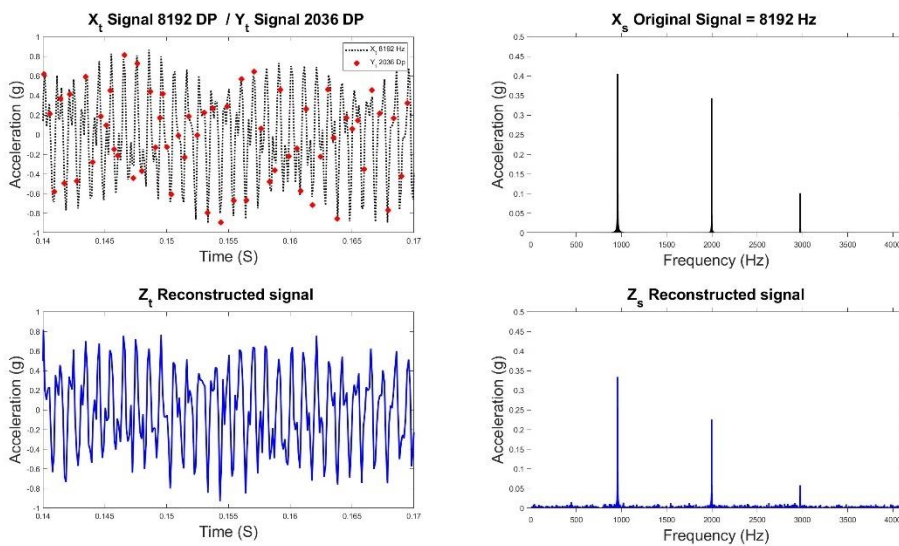
รูปที่ 1.2 ผลลัพธ์ของสัญญาณมอเตอร์ ครั้งที่ 2



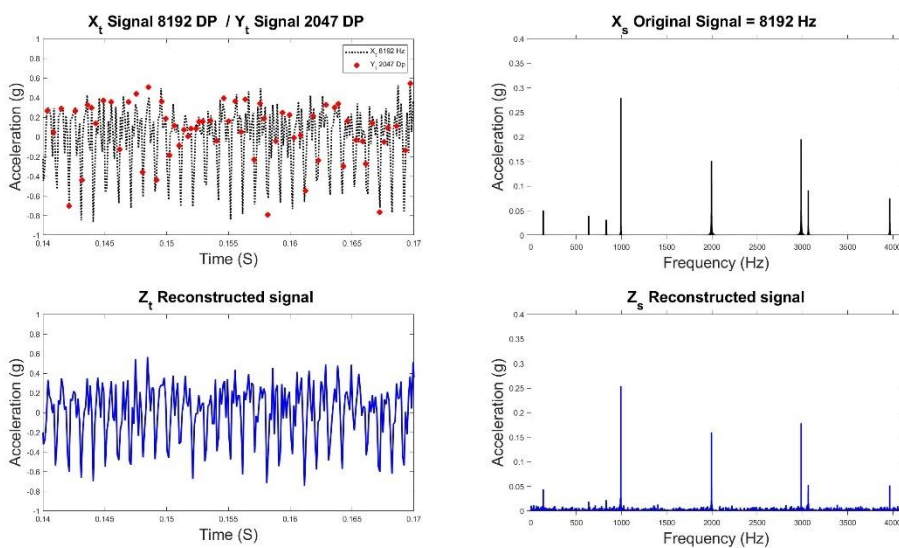
รูปที่ ๓.3 ผลลัพธ์ของสัญญาณมอเตอร์ ครั้งที่ 3



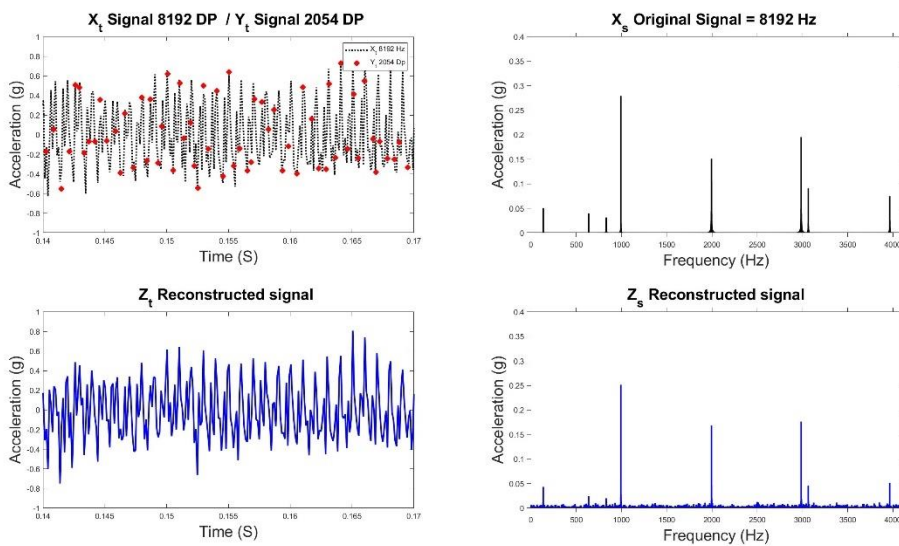
รูปที่ ๓.4 ผลลัพธ์ของสัญญาณมอเตอร์ ครั้งที่ 4



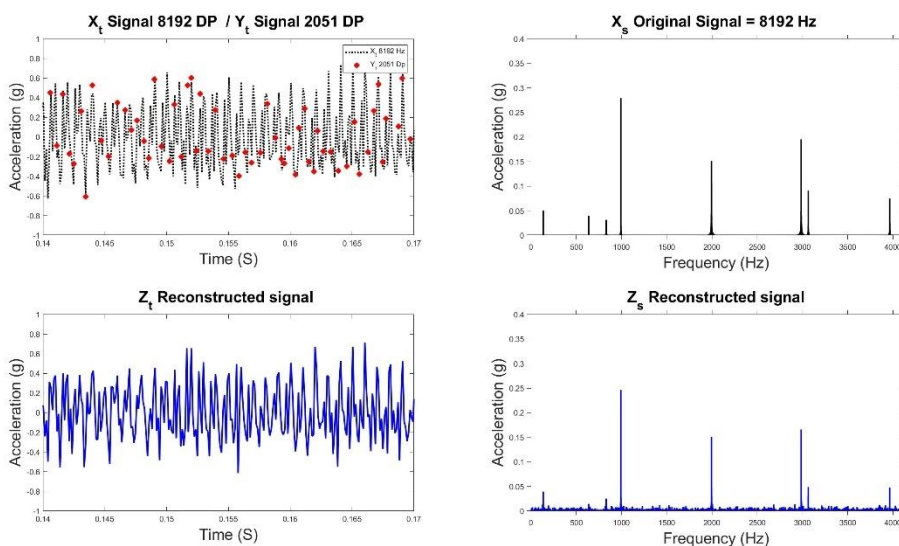
รูปที่ ๕.5 ผลลัพธ์ของสัญญาณมอเตอร์ ครั้งที่ 5



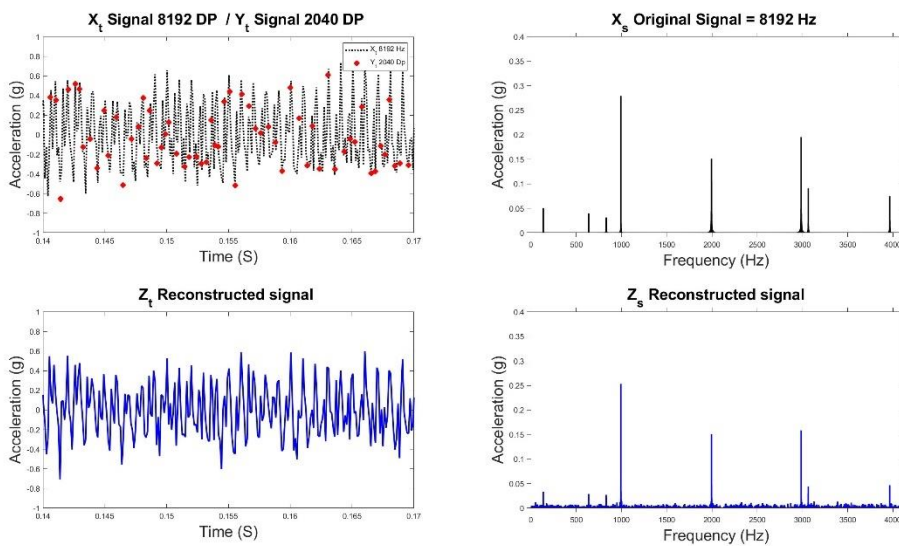
รูปที่ ๕.6 ผลลัพธ์ของสัญญาณปั๊ม ครั้งที่ 1



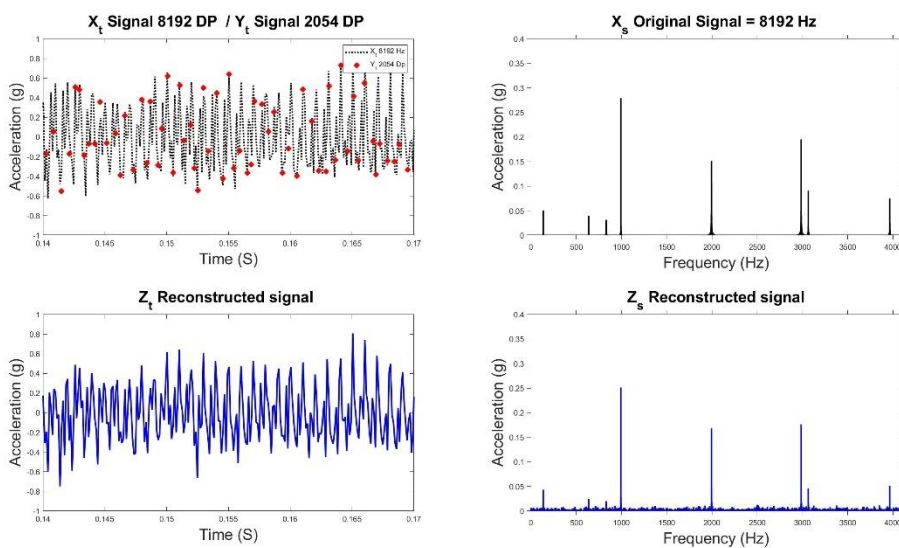
รูปที่ ๗.๗ ผลลัพธ์ของสัญญาณบีบ ครั้งที่ ๒



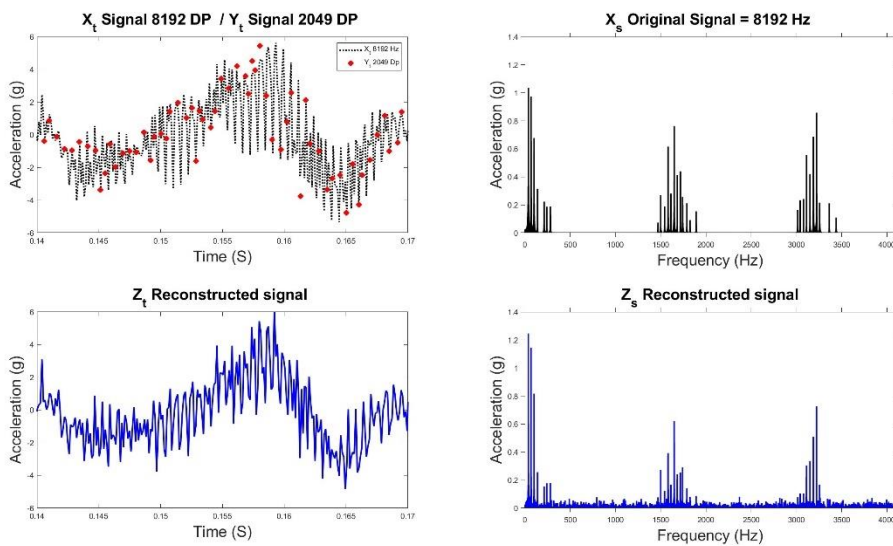
รูปที่ ๗.๘ ผลลัพธ์ของสัญญาณบีบ ครั้งที่ ๓



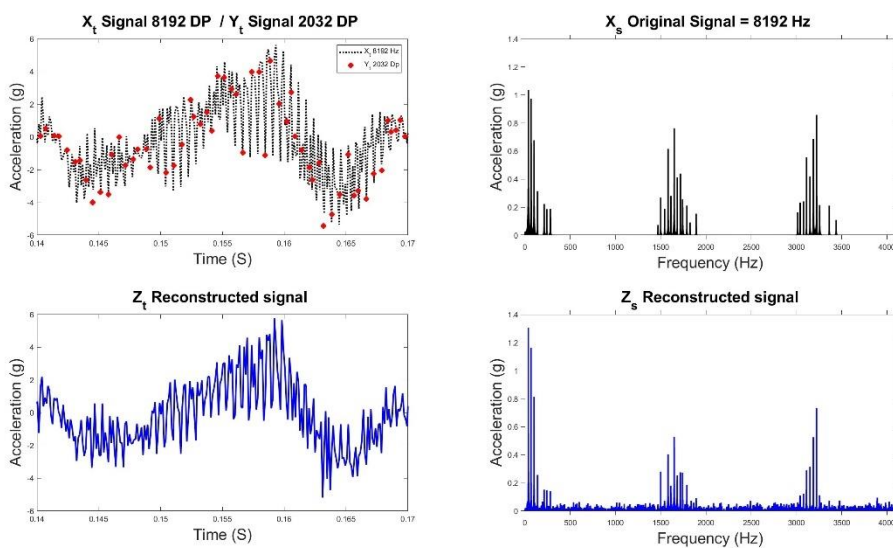
รูปที่ ๙.๙ ผลลัพธ์ของสัญญาณบีบ ครั้งที่ ๔



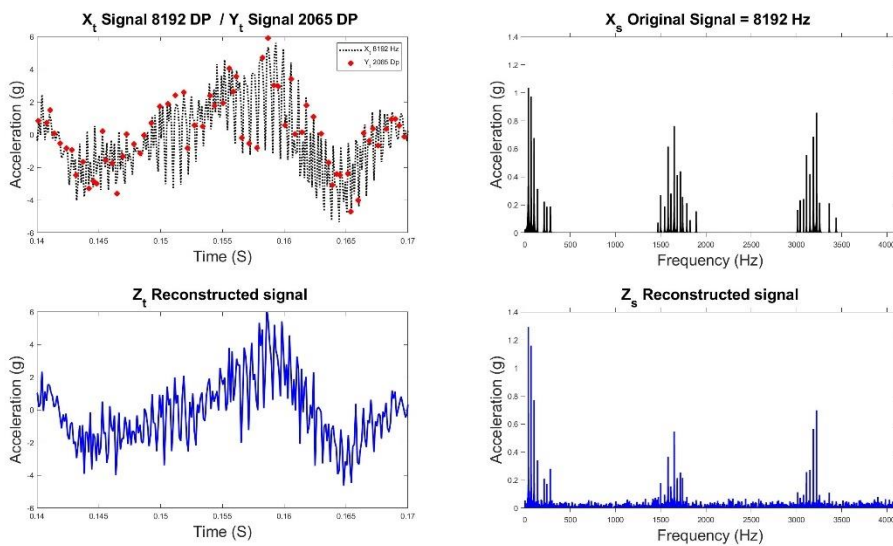
รูปที่ ๙.๑๐ ผลลัพธ์ของสัญญาณบีบ ครั้งที่ ๕



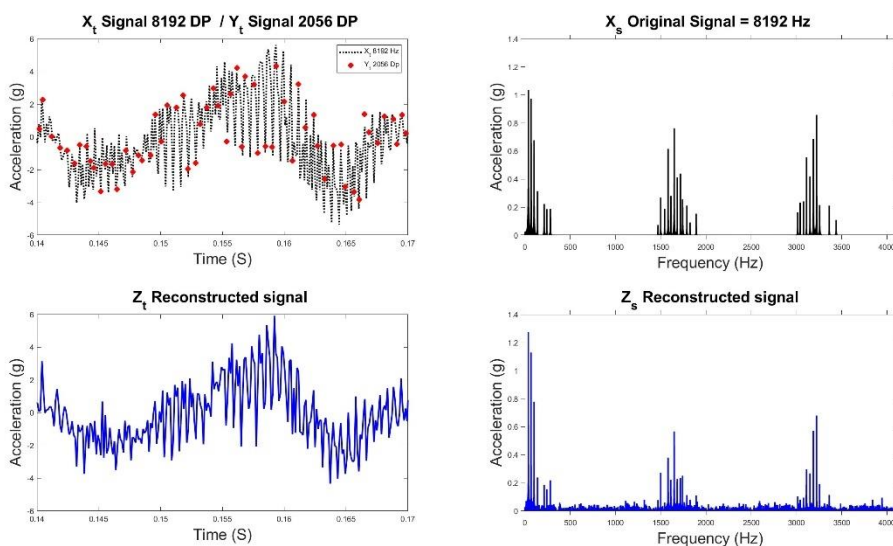
รูปที่ ๑๑.11 ผลลัพธ์ของสัญญาณชุดเกียร์ ครั้งที่ 1



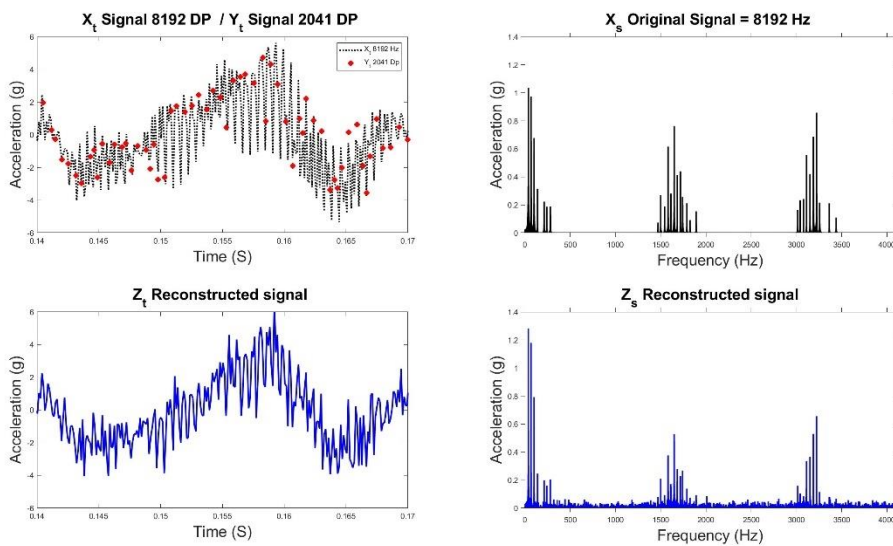
รูปที่ ๑๑.12 ผลลัพธ์ของสัญญาณชุดเกียร์ ครั้งที่ 2



รูปที่ ๑.13 ผลลัพธ์ของสัญญาณชุดเกียร์ ครั้งที่ 3



รูปที่ ๑.14 ผลลัพธ์ของสัญญาณชุดเกียร์ ครั้งที่ 4



รูปที่ ๑.15 ผลลัพธ์ของสัญญาณชุดเคียร์ ครั้งที่ 5

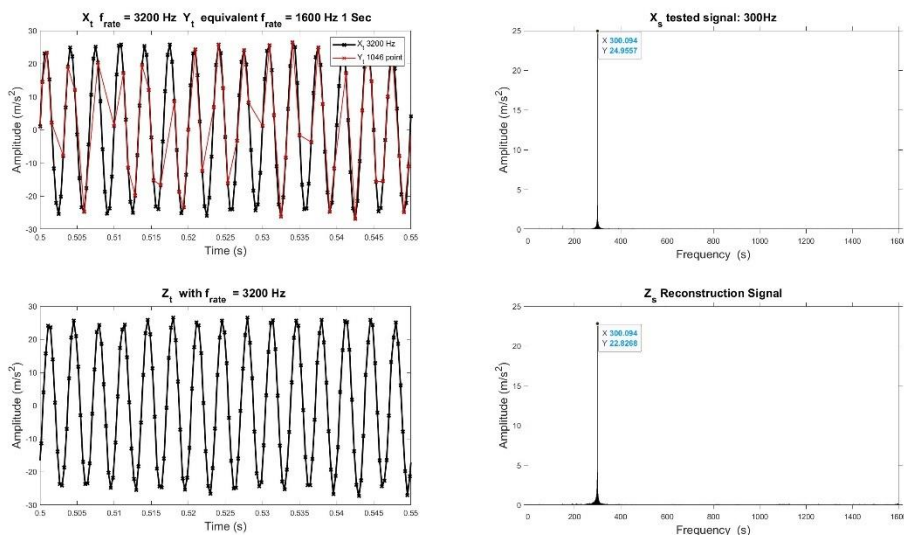


ภาคผนวก ซ.

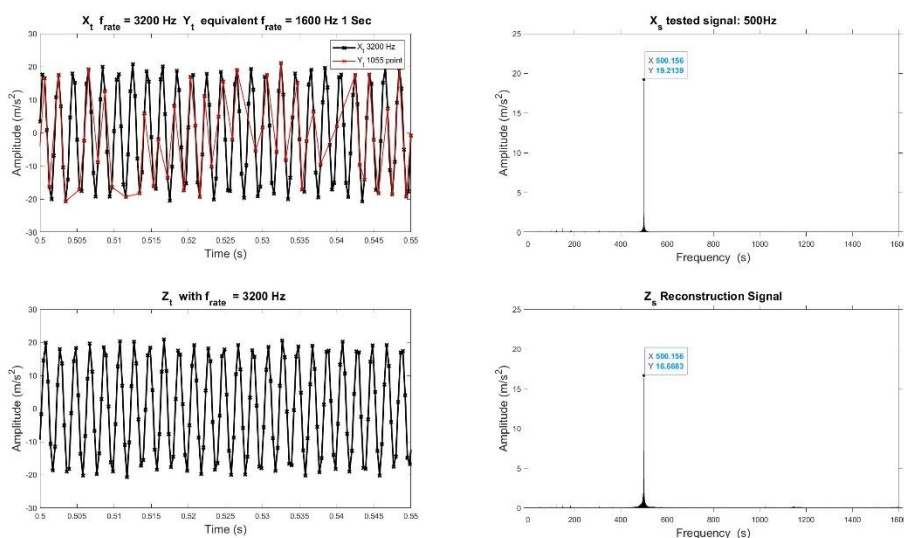
ผลการทดสอบระบบการวัดมาตรฐานทั่วไป ร่วมกับอัลกอริทึมคอมเพรสชันซึ่ง

สำหรับการใช้คอมเพรสชันซึ่งร่วมกับสัญญาณที่ถูกวัดด้วยระบบการวัดแบบมาตรฐาน ที่ความถี่สุ่ม 3200 เฮิรตซ์ จะมีผลทั้งหมด 20 ผลการทดสอบ โดยแบ่งเป็นเครื่องสร้างการสั่นสะเทือน 14 ผลการทดสอบ และ เครื่องคอมเพรสเซอร์ 1 และ 2 ลูกสูบอีก 12 ผลการทดสอบ โดยเลือกใช้ความถี่สุ่มเทียบเท่า 2 ความถี่คือ 640 เฮิรตซ์ และ 1600 เฮิรตซ์ ทดสอบความถี่ละ 3 ครั้ง เท่ากับ 6 ผลการทดสอบต่อ คอมเพรสเซอร์ 1 เครื่อง

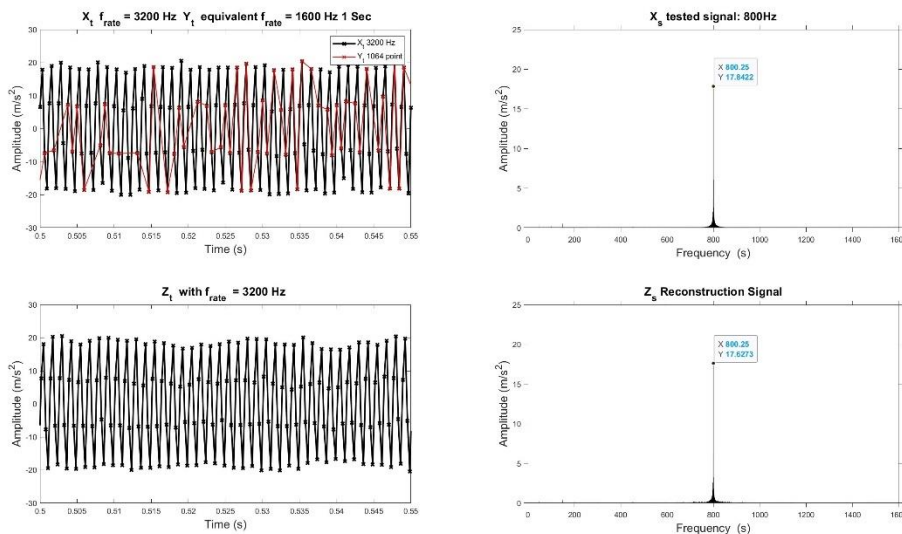
ผลการทดสอบสัญญาณจากเครื่องสร้างการสั่นสะเทือน โดยใช้ความถี่สุ่มเทียบเท่า 1600 เฮิรตซ์



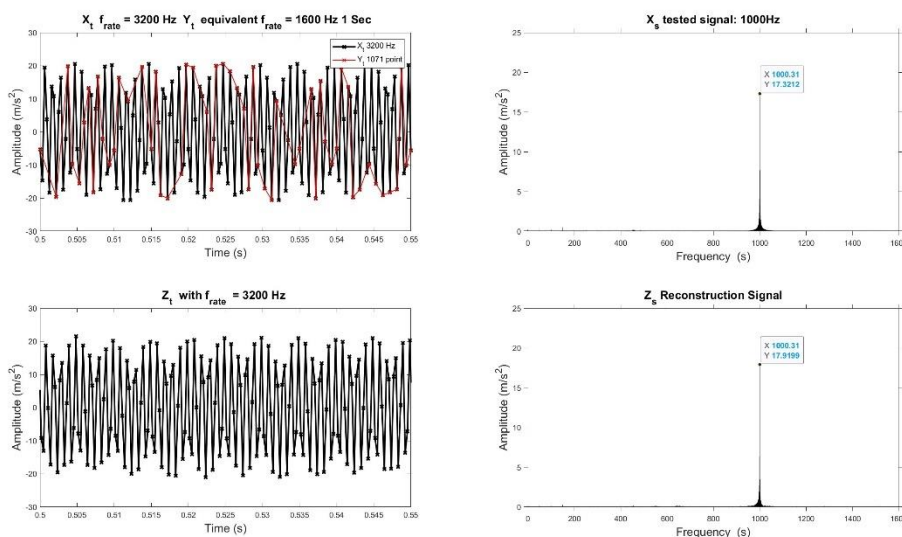
รูปที่ ซ.1 ผลลัพธ์ของเครื่องสร้างการสั่นสะเทือนที่สัญญาณความถี่ 300 เฮิรตซ์



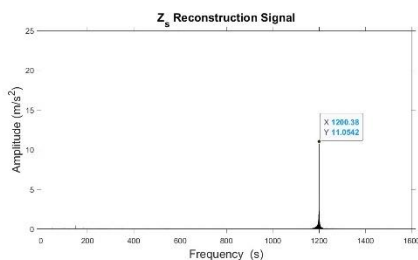
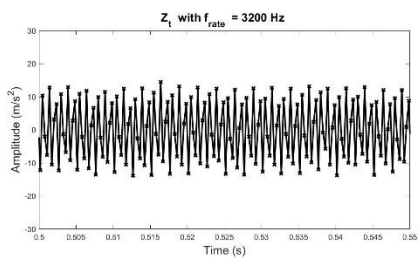
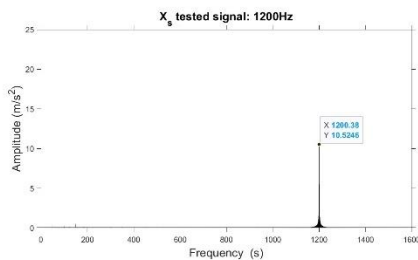
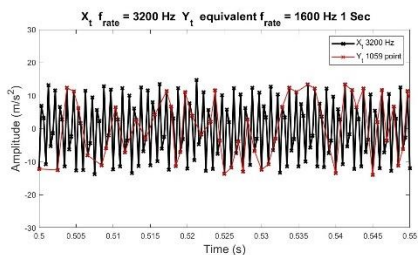
รูปที่ ซ.2 ผลลัพธ์ของเครื่องสร้างการสั่นสะเทือนที่สัญญาณความถี่ 500 เฮิรตซ์



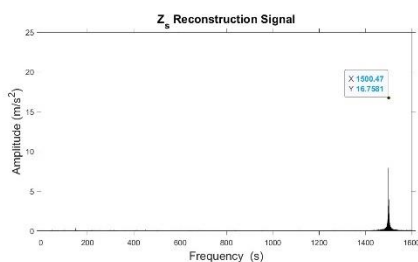
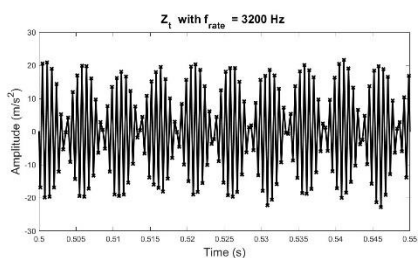
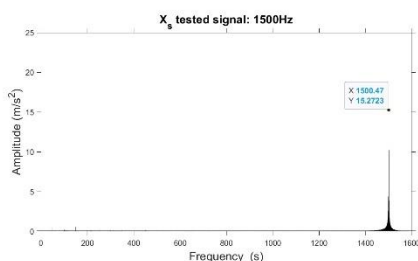
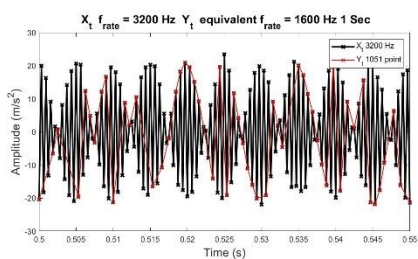
รูปที่ ๓.3 ผลลัพธ์ของเครื่องสร้างการสั่นสะเทือนที่สัญญาณความถี่ 800 เฮิรตซ์



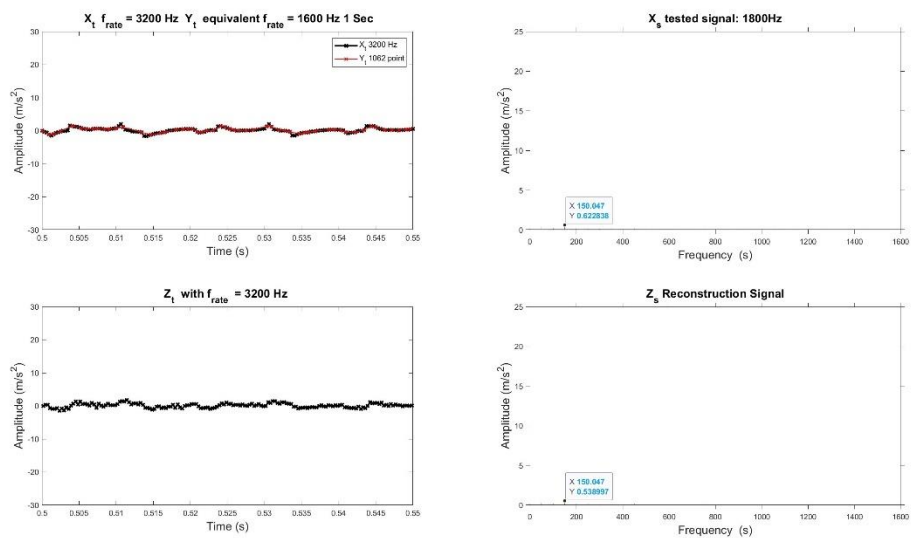
รูปที่ ๓.4 ผลลัพธ์ของเครื่องสร้างการสั่นสะเทือนที่สัญญาณความถี่ 1000 เฮิรตซ์



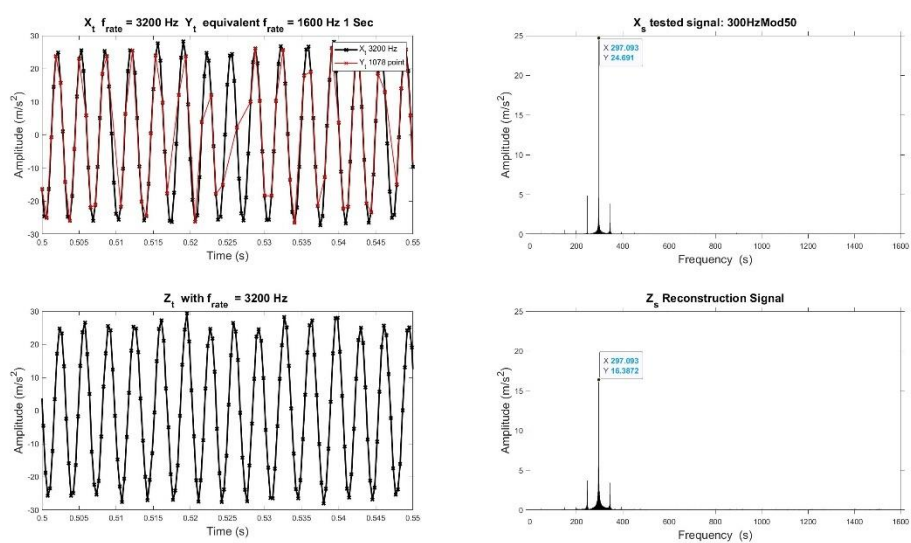
รูปที่ ๗.5 ผลลัพธ์ของเครื่องสร้างการสั่นสะเทือนที่สัญญาณความถี่ 1200 เฮิรตซ์



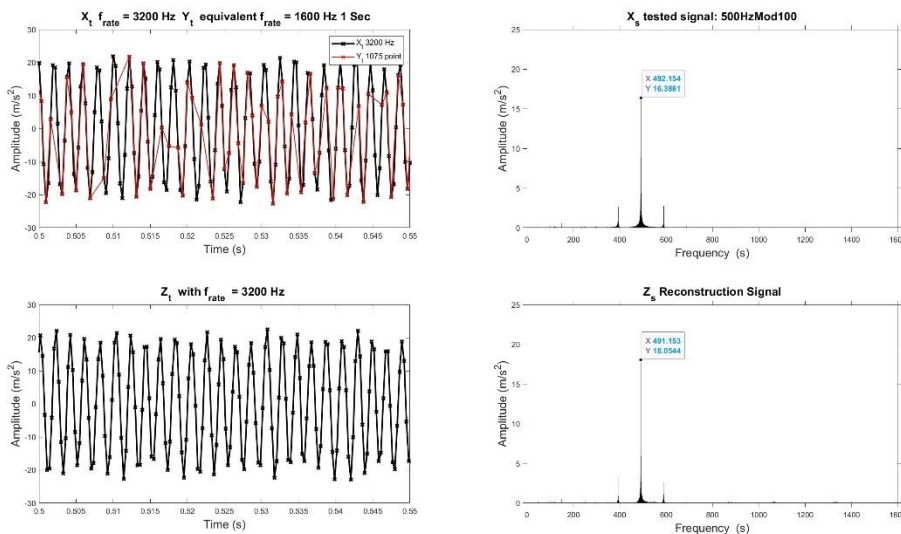
รูปที่ ๗.6 ผลลัพธ์ของเครื่องสร้างการสั่นสะเทือนที่สัญญาณความถี่ 1500 เฮิรตซ์



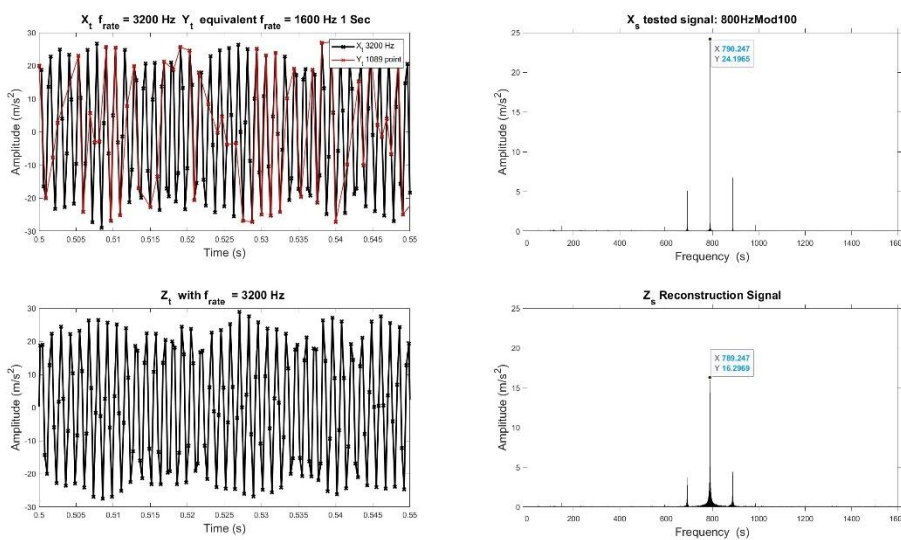
รูปที่ ๗.7 ผลลัพธ์ของเครื่องสร้างการสั่นสะเทือนที่สัญญาณความถี่ 1800 เฮิรตซ์



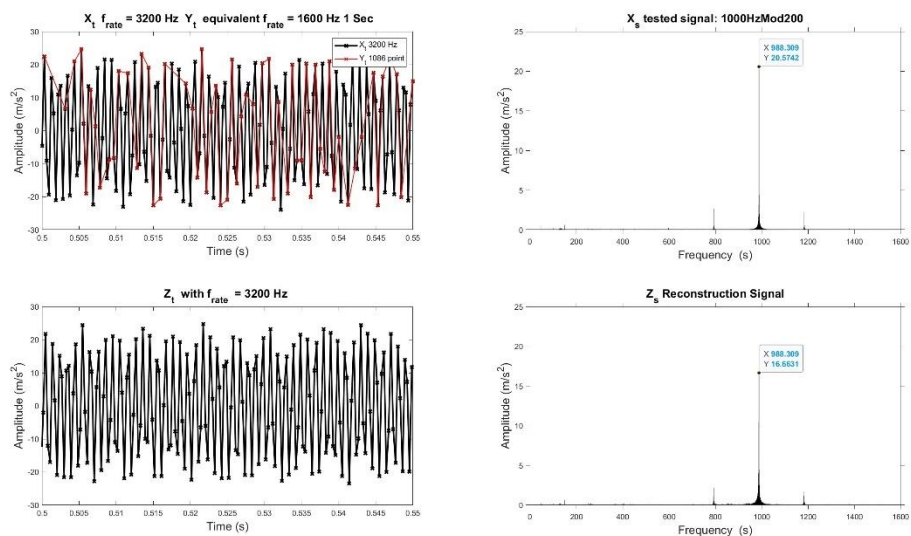
รูปที่ ๗.8 ผลลัพธ์ของเครื่องสร้างการสั่นสะเทือนที่สัญญาณความถี่ 300 เฮิรตซ์ มอดดูเลชัน 50 เฮิรตซ์



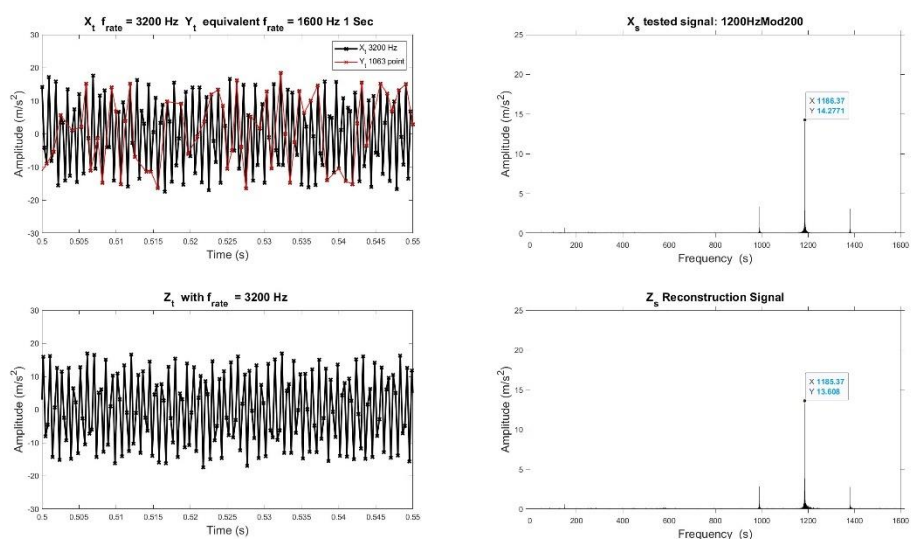
รูปที่ ๙.9 ผลลัพธ์ของเครื่องสร้างการสั่นสะเทือนที่สัญญาณความถี่ 500 เฮิรตซ์ มอดูเลชั่น 100 เฮิรตซ์



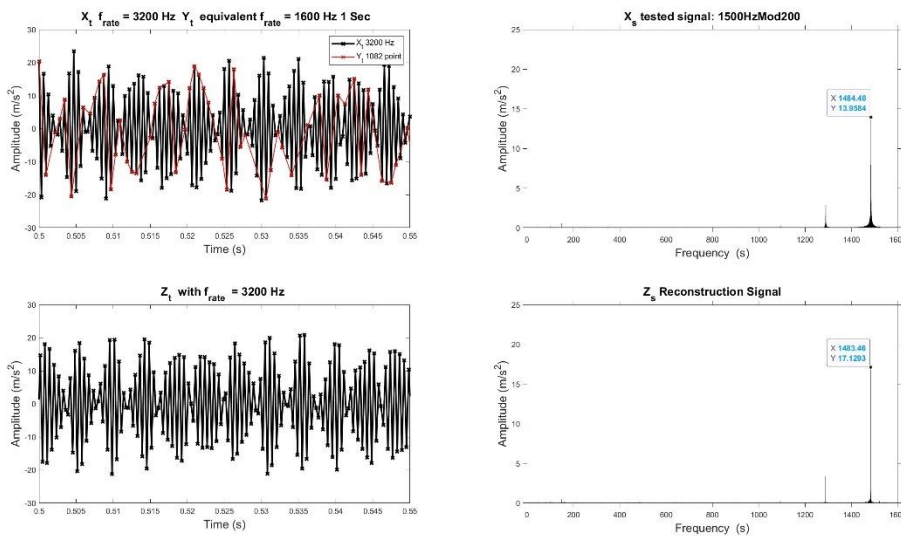
รูปที่ ๙.10 ผลลัพธ์ของเครื่องสร้างการสั่นสะเทือนที่สัญญาณความถี่ 800 เฮิรตซ์ มอดูเลชั่น 100 เฮิรตซ์



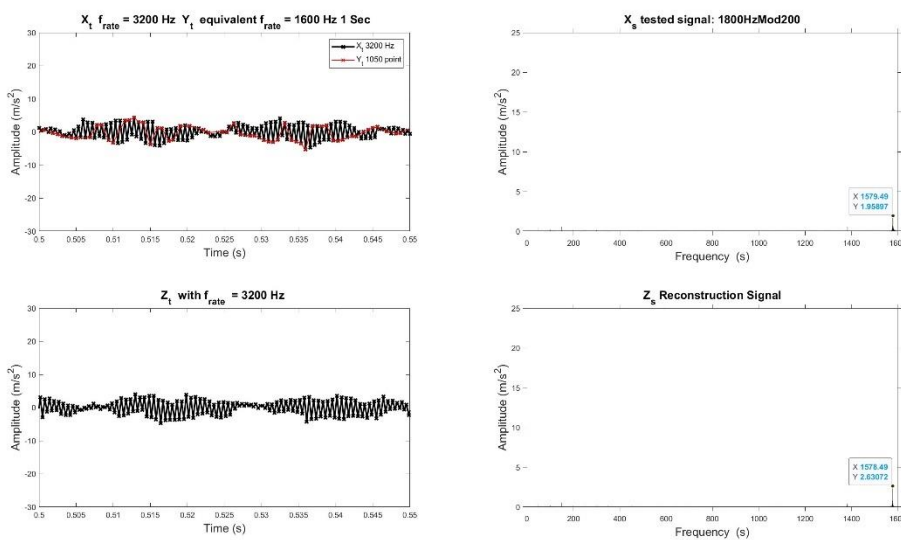
รูปที่ ๑๑.11 ผลลัพธ์ของเครื่องสร้างการสั่นสะเทือนที่สัญญาณความถี่ 1000 เฮิร์ตซ์ มอดูเลชัน 200 เฮิร์ตซ์



รูปที่ ๑๑.12 ผลลัพธ์ของเครื่องสร้างการสั่นสะเทือนที่สัญญาณความถี่ 1200 เฮิร์ตซ์ มอดูเลชัน 200 เฮิร์ตซ์

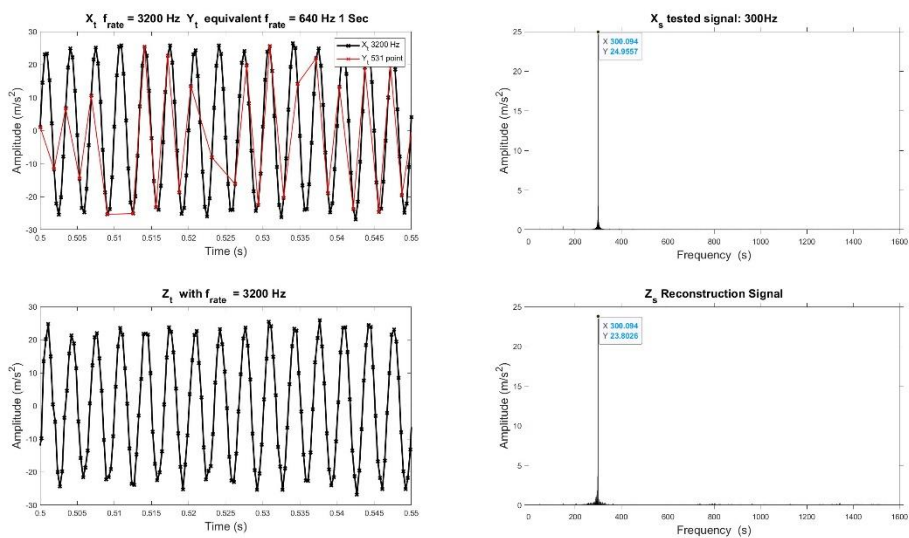


รูปที่ ๑๓. ผลลัพธ์ของเครื่องสร้างการสั่นสะเทือนที่สัญญาณความถี่ 1500 เฮิรตซ์ มอดดูเลชัน 200 เฮิรตซ์

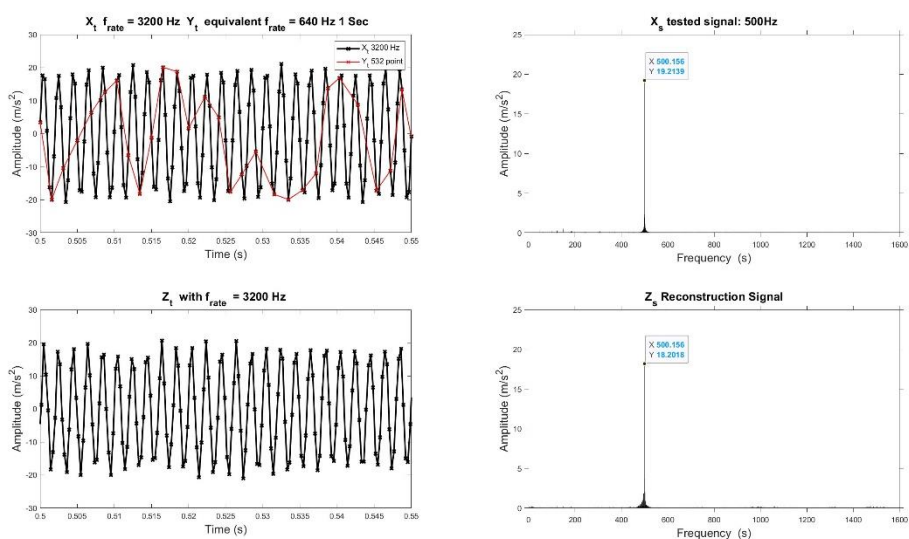


รูปที่ ๑๔. ผลลัพธ์ของเครื่องสร้างการสั่นสะเทือนที่สัญญาณความถี่ 1800 เฮิรตซ์ มอดดูเลชัน 200 เฮิรตซ์

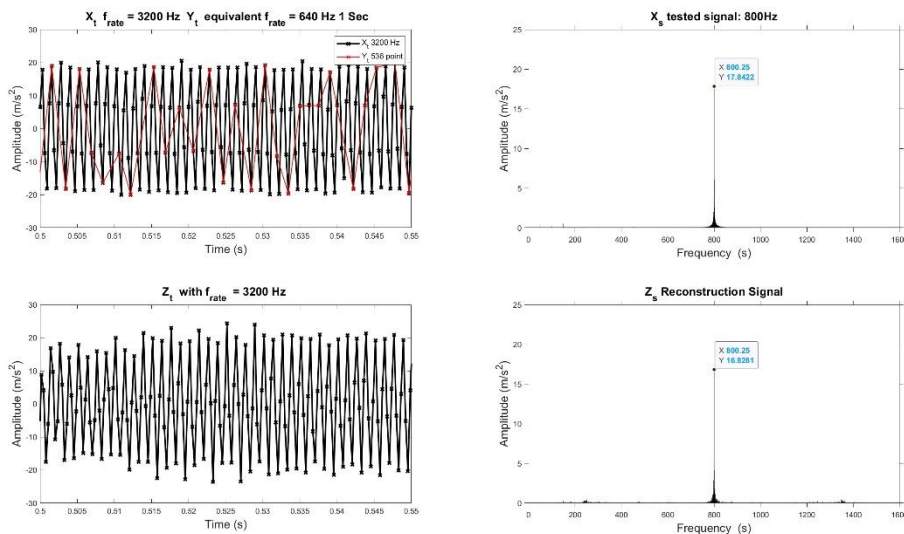
ผลการทดสอบสัญญาณจากเครื่องสร้างการสั่นสะเทือน โดยใช้ความถี่สุ่มเทียบเท่า 640 เฮิรตซ์



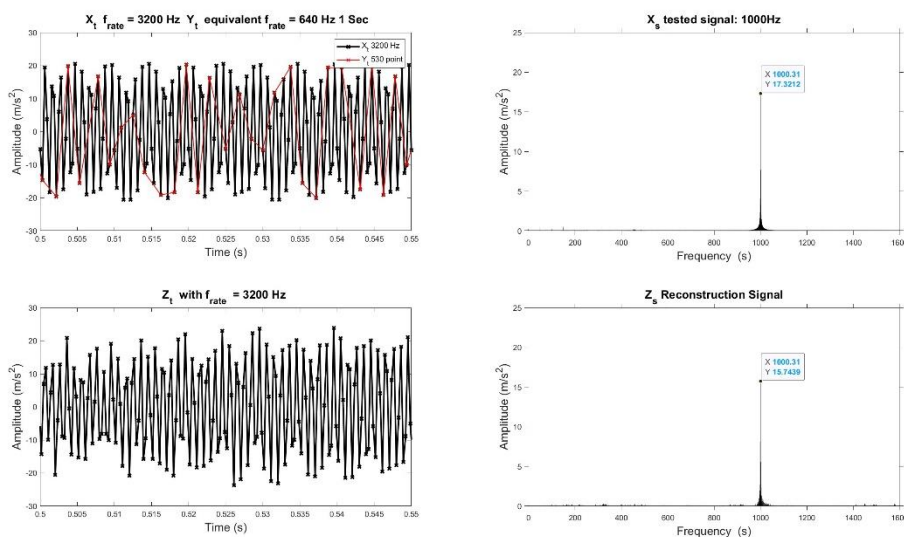
รูปที่ ๑๕.15 ผลลัพธ์ของเครื่องสร้างการสั่นสะเทือนที่สัญญาณความถี่ 300 เฮิรตซ์



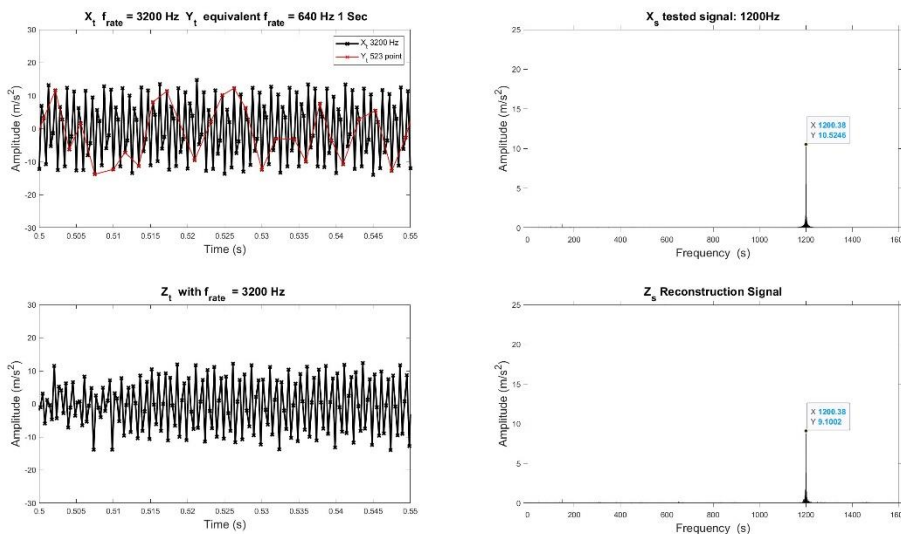
รูปที่ ๑๕.16 ผลลัพธ์ของเครื่องสร้างการสั่นสะเทือนที่สัญญาณความถี่ 500 เฮิรตซ์



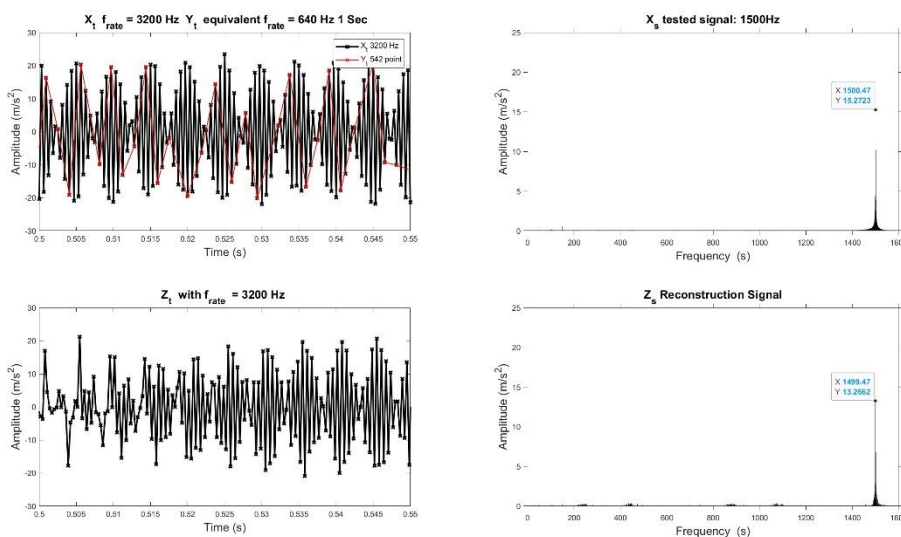
รูปที่ ๑๗.17 ผลลัพธ์ของเครื่องสร้างการสั่นสะเทือนที่สัญญาณความถี่ 800 เฮิรตซ์



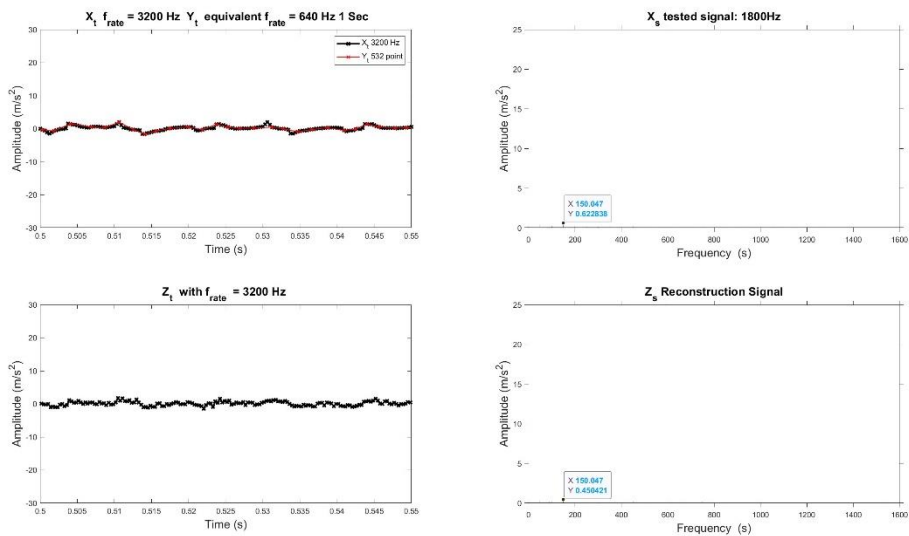
รูปที่ ๑๗.18 ผลลัพธ์ของเครื่องสร้างการสั่นสะเทือนที่สัญญาณความถี่ 1000 เฮิรตซ์



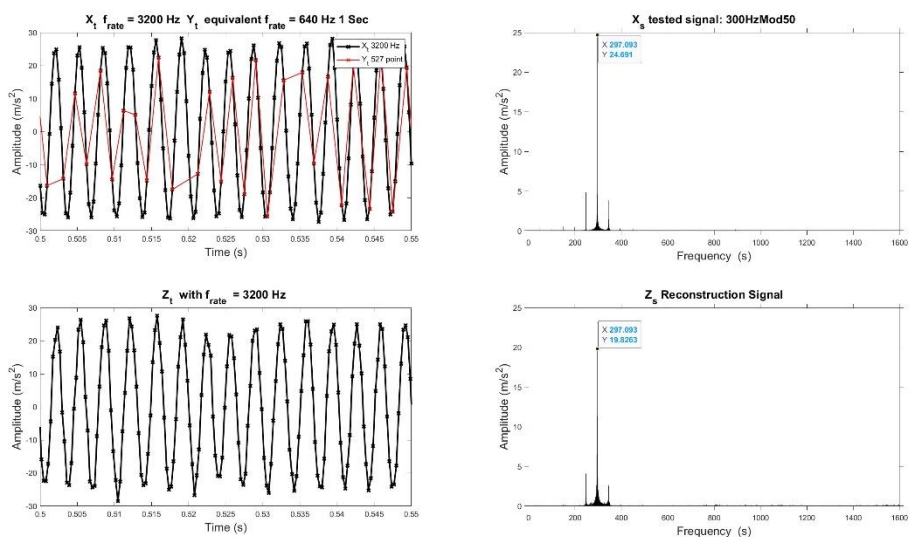
รูปที่ ๑๙.19 ผลลัพธ์ของเครื่องสร้างการสั่นสะเทือนที่สัญญาณความถี่ 1200 เฮิร์ตซ์



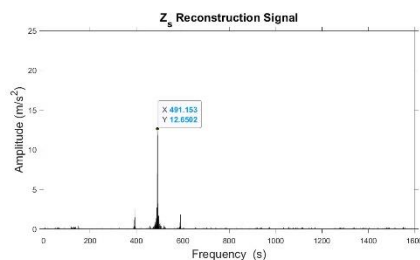
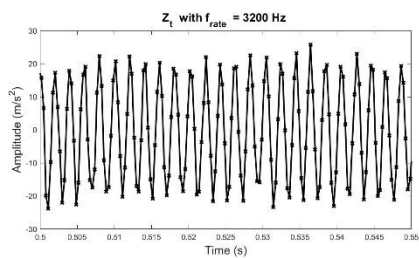
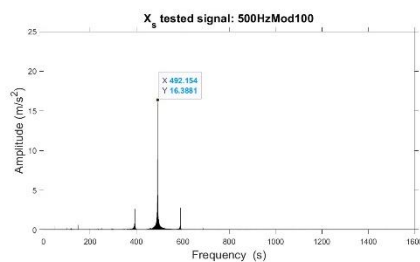
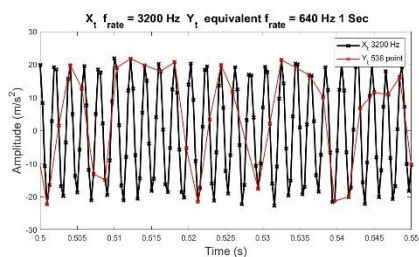
รูปที่ ๑๙.20 ผลลัพธ์ของเครื่องสร้างการสั่นสะเทือนที่สัญญาณความถี่ 1500 เฮิร์ตซ์



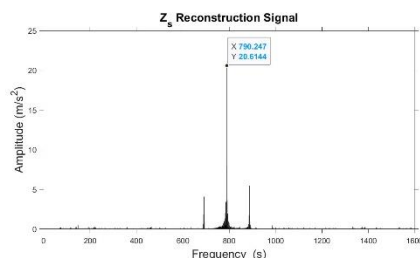
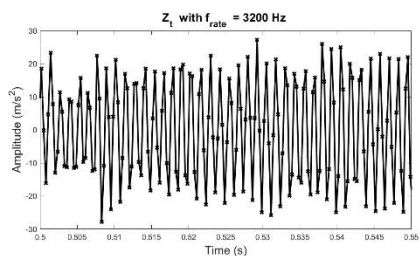
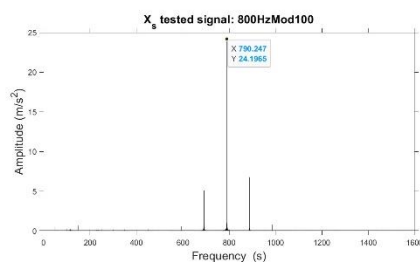
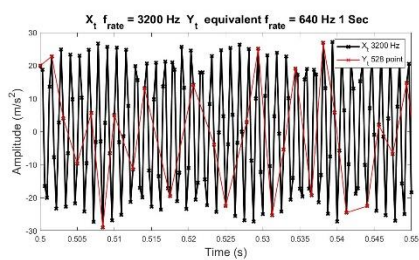
รูปที่ ช.21 ผลลัพธ์ของเครื่องสร้างการสั่นสะเทือนที่สัญญาณความถี่ 1800 เฮิรตซ์



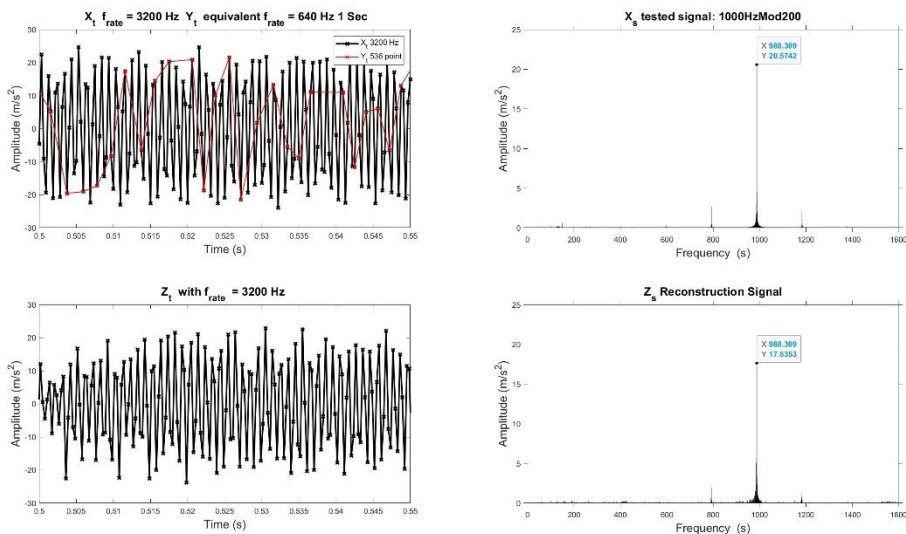
รูปที่ ช.22 ผลลัพธ์ของเครื่องสร้างการสั่นสะเทือนที่สัญญาณความถี่ 300 เฮิรตซ์ มอดุเลชั่น 50 เฮิรตซ์



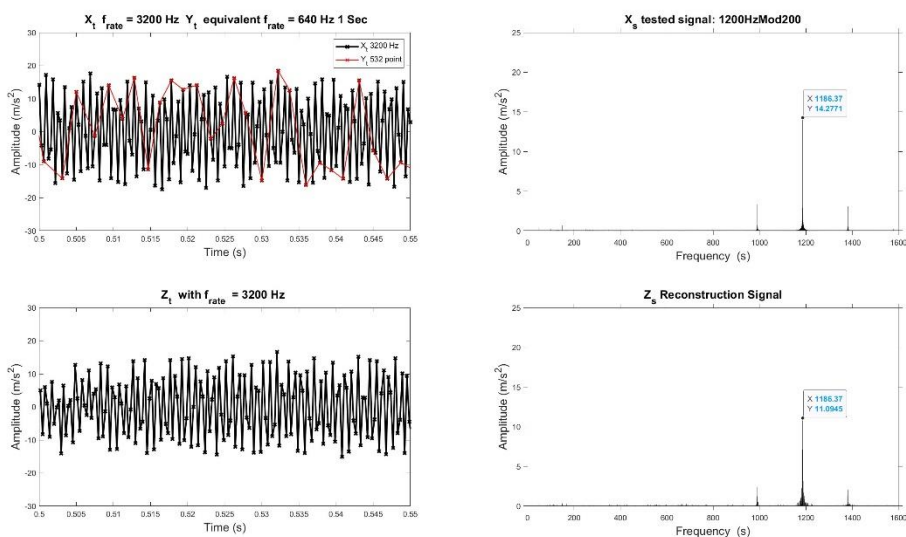
รูปที่ ๒.๒๓ ผลลัพธ์ของการสร้างการสั่นสะเทือนที่สัญญาณความถี่ 500 เฮิรตซ์ มอดูเลชั่น 100 เฮิรตซ์



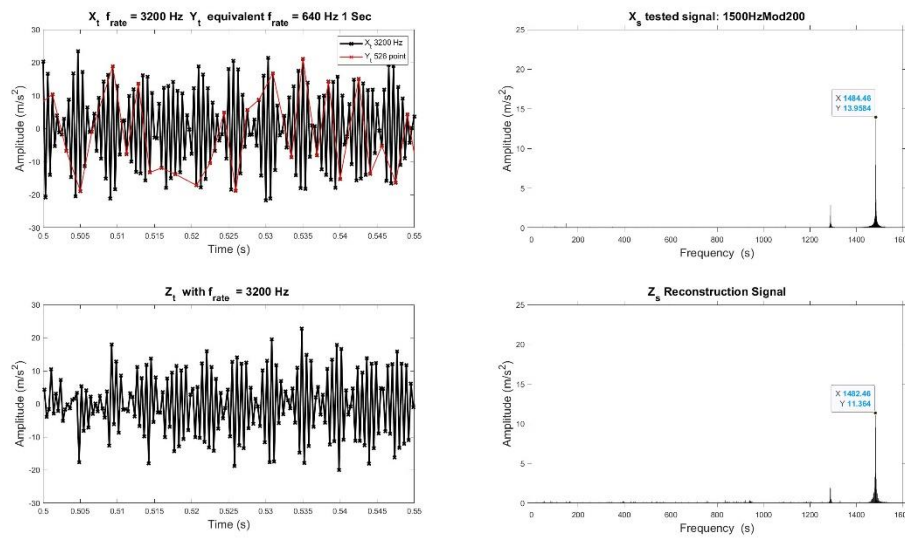
รูปที่ ๒.๒๔ ผลลัพธ์ของการสร้างการสั่นสะเทือนที่สัญญาณความถี่ 800 เฮิรตซ์ มอดูเลชั่น 100 เฮิรตซ์



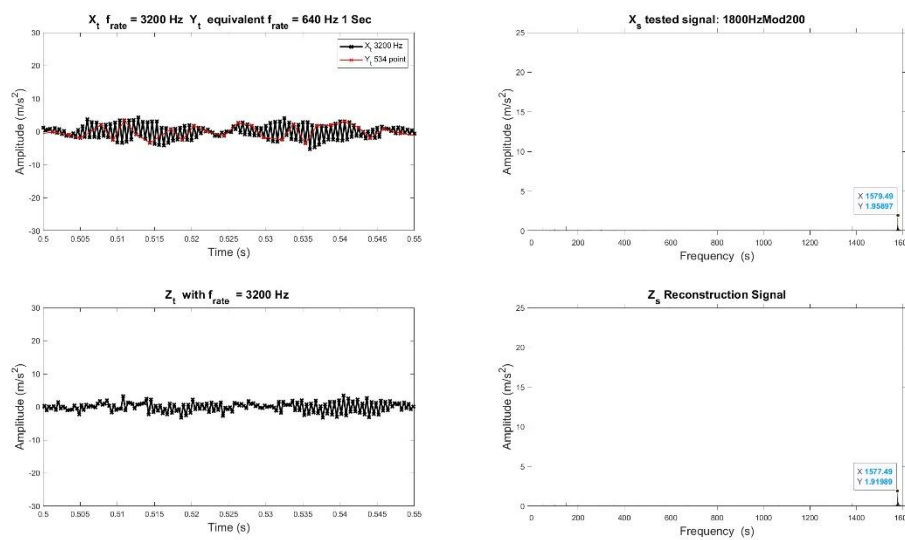
รูปที่ ๒.25 ผลลัพธ์ของเครื่องสร้างการสั่นสะเทือนที่สัญญาณความถี่ 1000 เฮิร์ตซ์ มอดูเลชัน 200 เฮิร์ตซ์



รูปที่ ๒.26 ผลลัพธ์ของเครื่องสร้างการสั่นสะเทือนที่สัญญาณความถี่ 1200 เฮิร์ตซ์ มอดูเลชัน 200 เฮิร์ตซ์



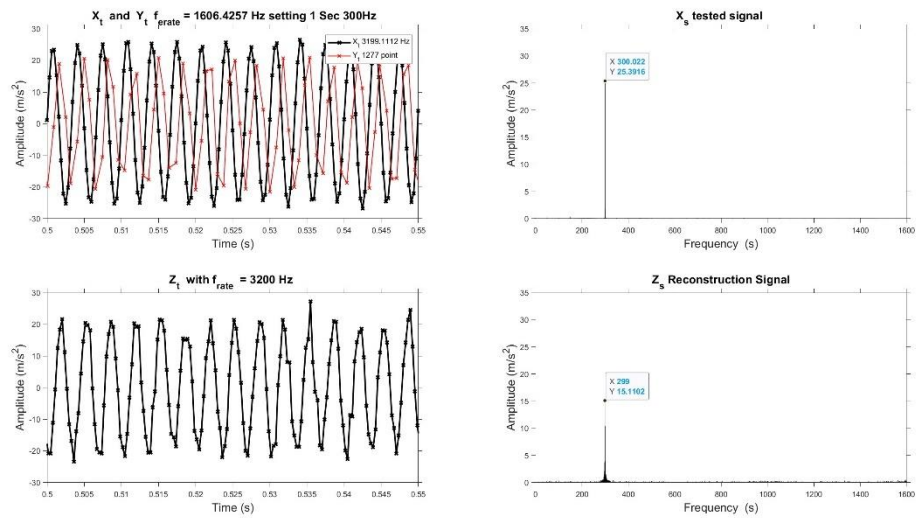
รูปที่ ๒.๒๗ ผลลัพธ์ของเครื่องสร้างการสั่นสะเทือนที่สัญญาณความถี่ 1500 เฮิร์ตซ์ มอดูเลชัน 200 เฮิร์ตซ์



รูปที่ ๒.๒๘ ผลลัพธ์ของเครื่องสร้างการสั่นสะเทือนที่สัญญาณความถี่ 1800 เฮิร์ตซ์ มอดูเลชัน 200 เฮิร์ตซ์

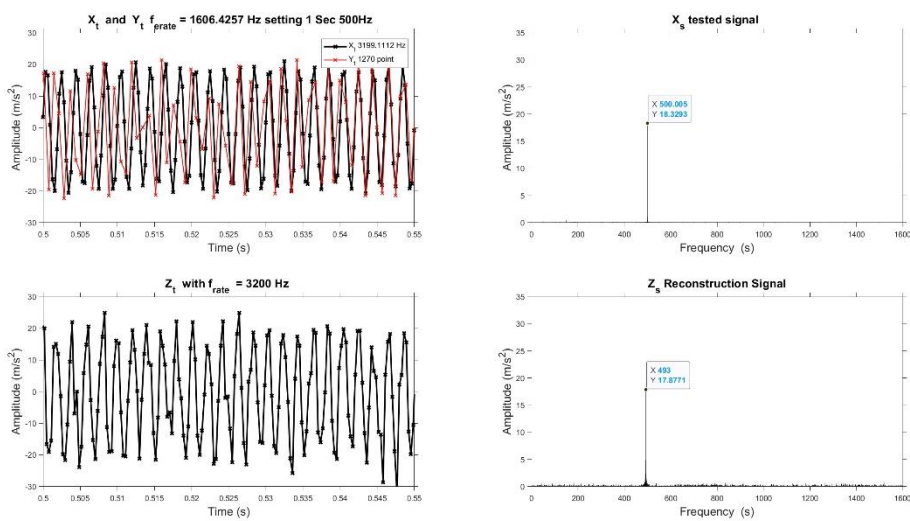
ภาคผนวก ณ.

ผลการทดสอบระบบการวัดแบบสุ่มคาบของเครื่องสร้างการสั่นสะเทือน

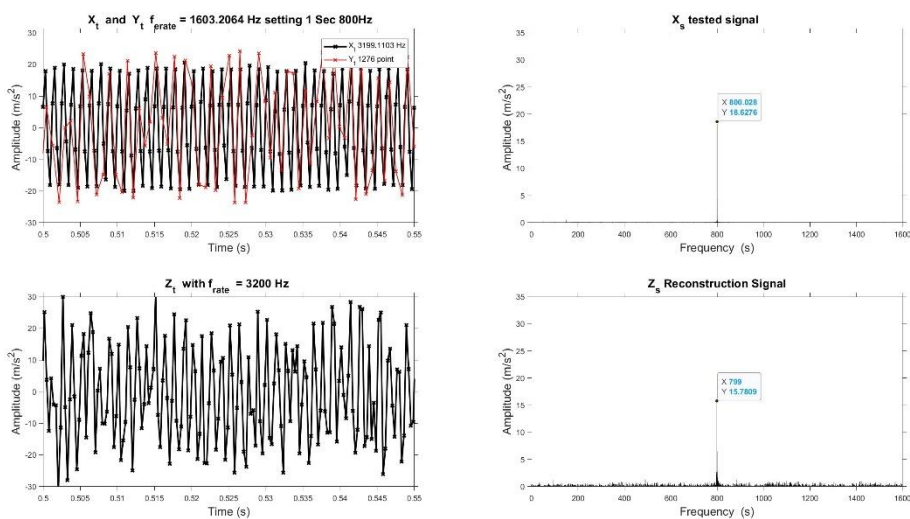


รูปที่ ณ.1 การปรับตั้งแบบที่ 1 ที่สัญญาณความถี่ 300 เฮิร์ตซ์

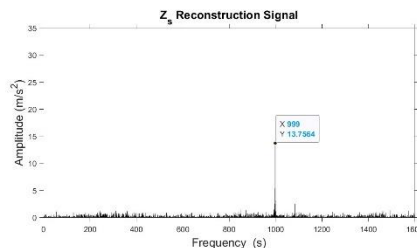
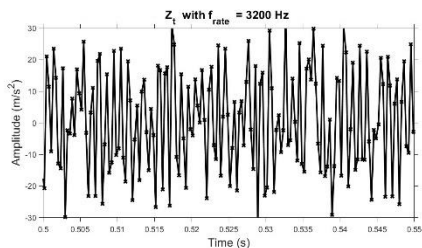
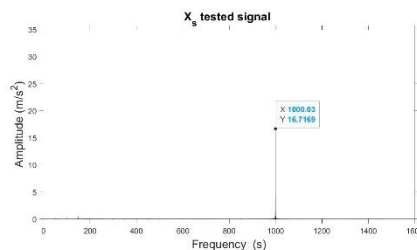
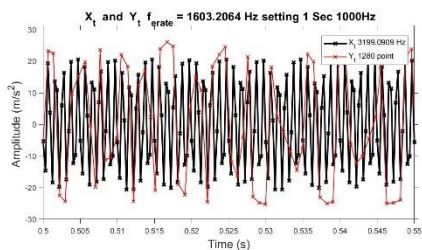




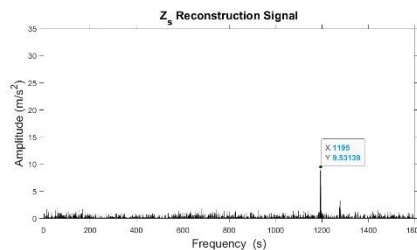
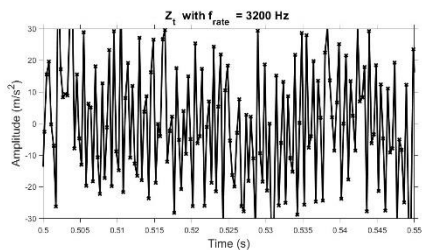
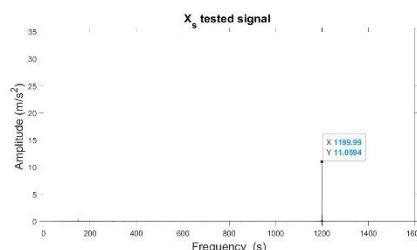
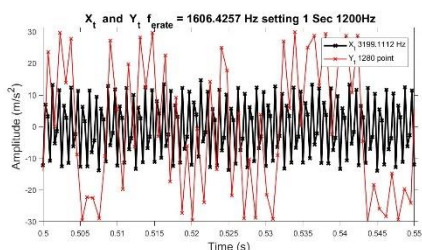
รูปที่ ๓.๒ การปรับตั้งแบบที่ 1 ที่สัญญาณความถี่ 500 เฮิรตซ์



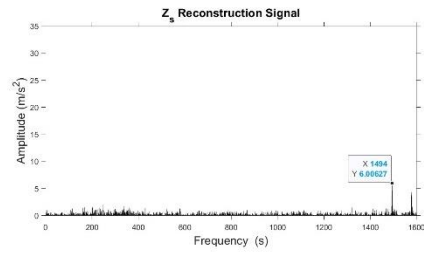
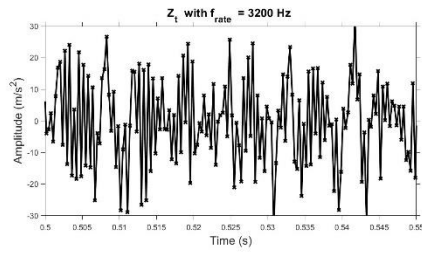
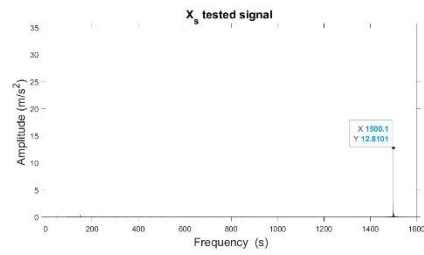
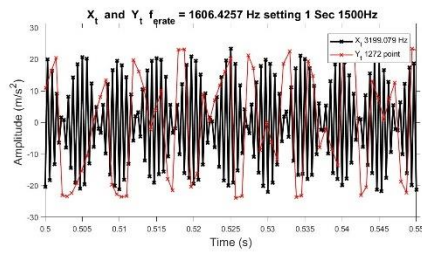
รูปที่ ๓.๓ การปรับตั้งแบบที่ 1 ที่สัญญาณความถี่ 800 เฮิรตซ์



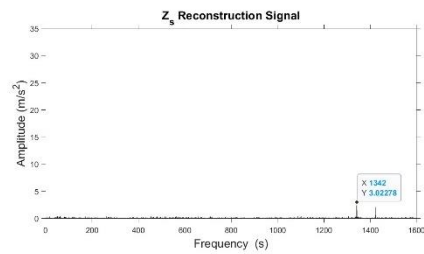
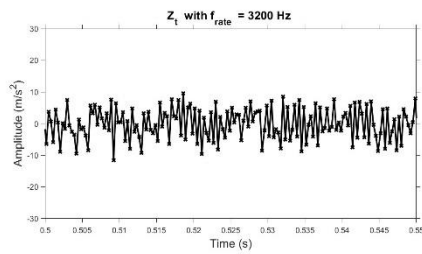
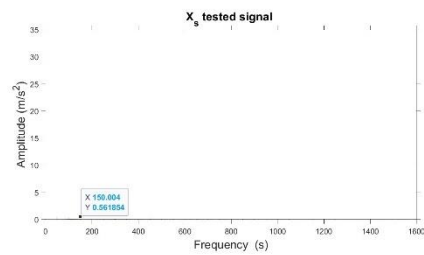
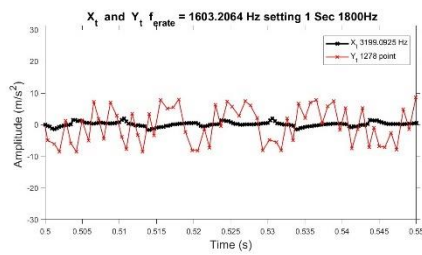
รูปที่ ๓.๔ การปรับตั้งแบบที่ 1 ที่สัญญาณความถี่ 1000 เฮิรตซ์



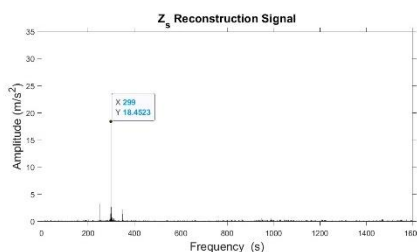
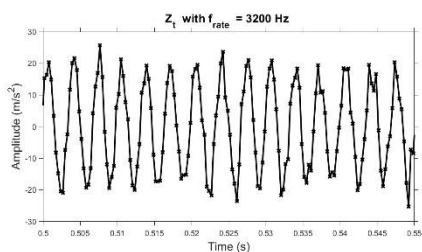
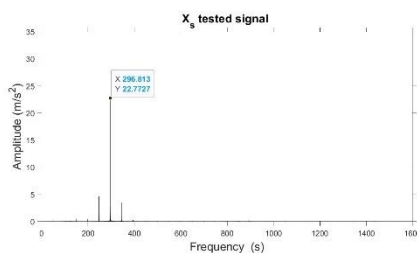
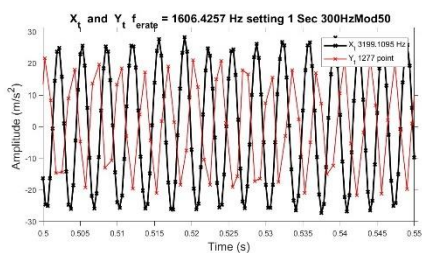
รูปที่ ๓.๕ การปรับตั้งแบบที่ 1 ที่สัญญาณความถี่ 1200 เฮิรตซ์



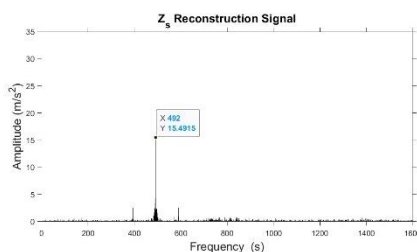
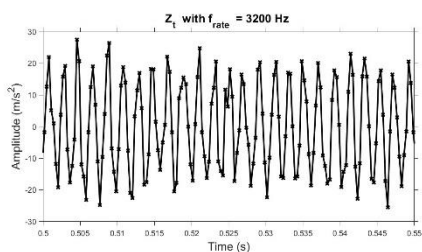
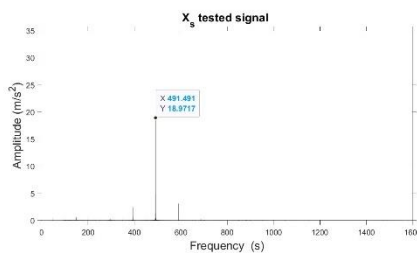
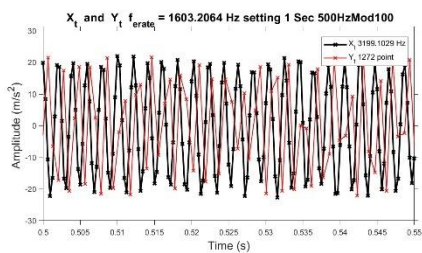
รูปที่ ๓.๖ การปรับตั้งแบบที่ 1 ที่สัญญาณความถี่ 1500 เฮิรตซ์



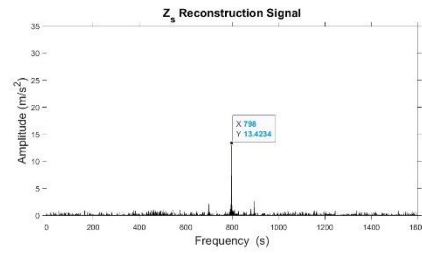
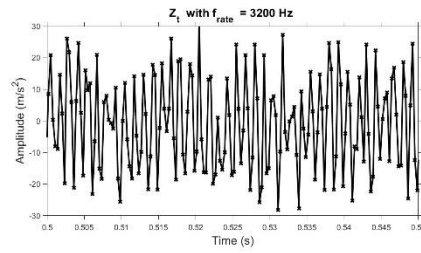
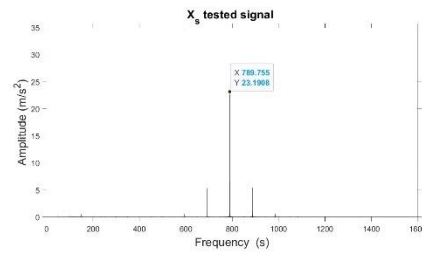
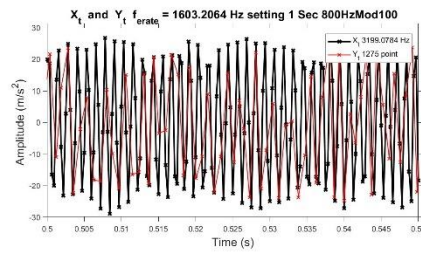
รูปที่ ๓.๗ การปรับตั้งแบบที่ 1 ที่สัญญาณความถี่ 1800 เฮิรตซ์



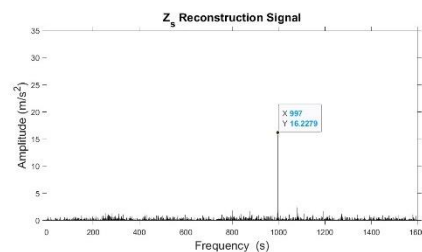
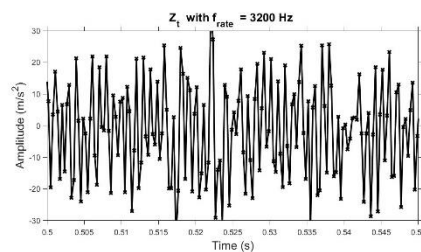
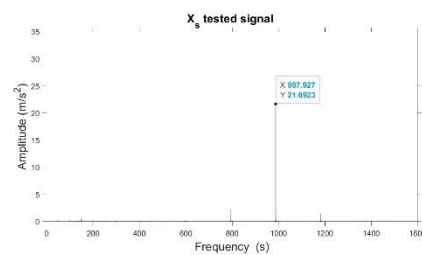
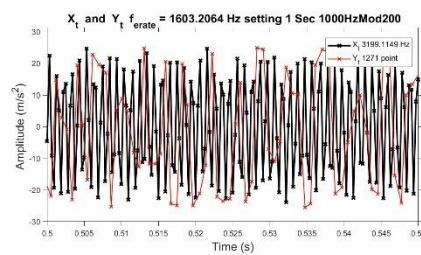
รูปที่ ๓.8 การปรับตั้งแบบที่ 1 ที่สัญญาณความถี่ 300 เฮิรตซ์ มอดูเลชั่น



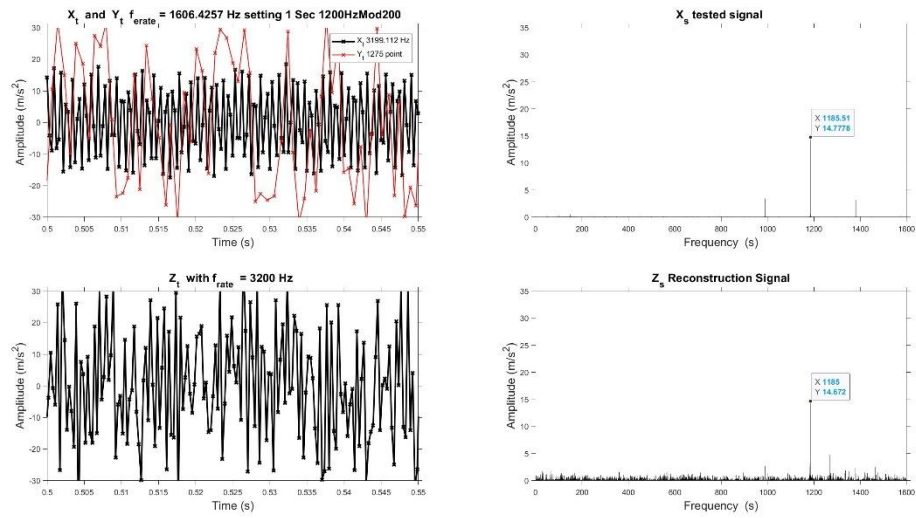
รูปที่ ๓.9 การปรับตั้งแบบที่ 1 ที่สัญญาณความถี่ 500 เฮิรตซ์ มอดูเลชั่น



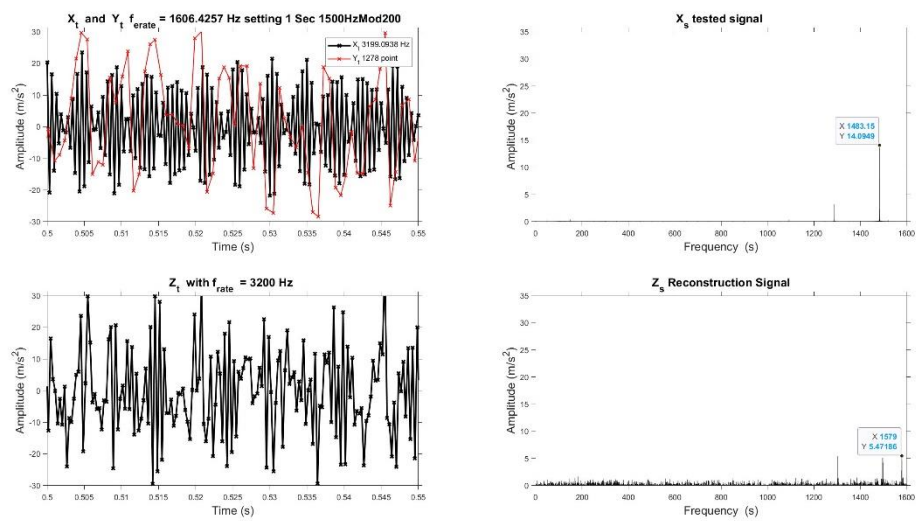
รูปที่ ๑๐.10 การปรับตั้งแบบที่ 1 ที่สัญญาณความถี่ 800 เฮิรตซ์ มอดูเลชั่น



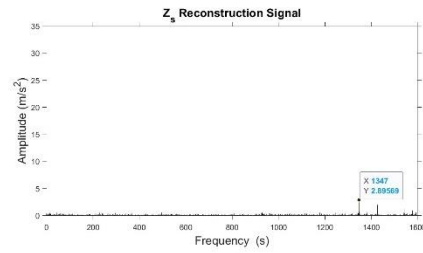
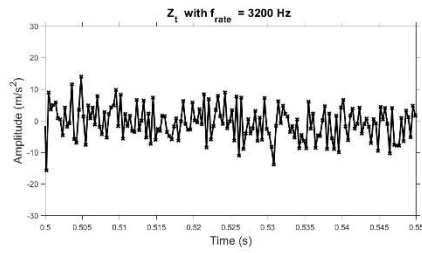
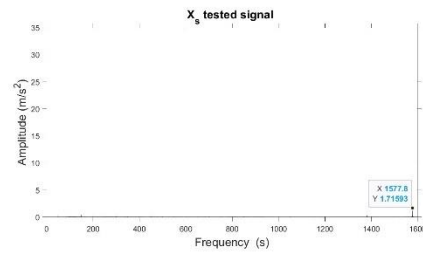
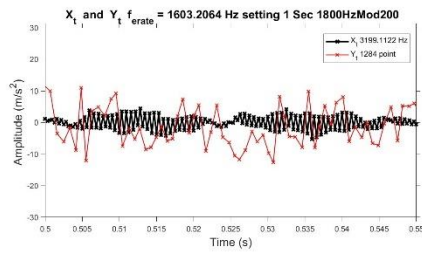
รูปที่ ๑๐.11 การปรับตั้งแบบที่ 1 ที่สัญญาณความถี่ 1000 เฮิรตซ์ มอดูเลชั่น



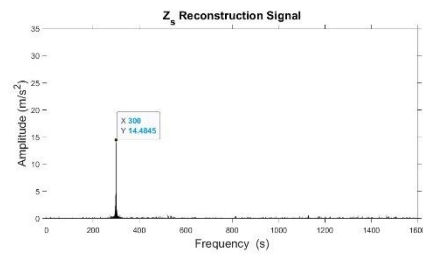
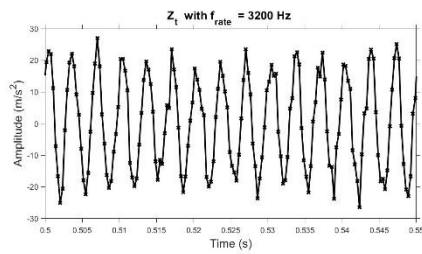
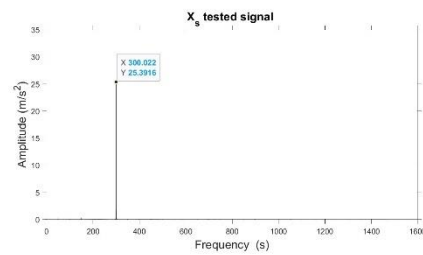
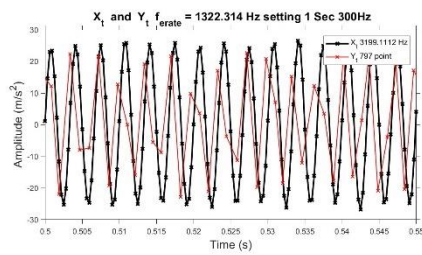
รูปที่ ๑๒.12 การปรับตั้งแบบที่ 1 ที่สัญญาณความถี่ 1200 เฮิรตซ์ มอดูเลชั่น



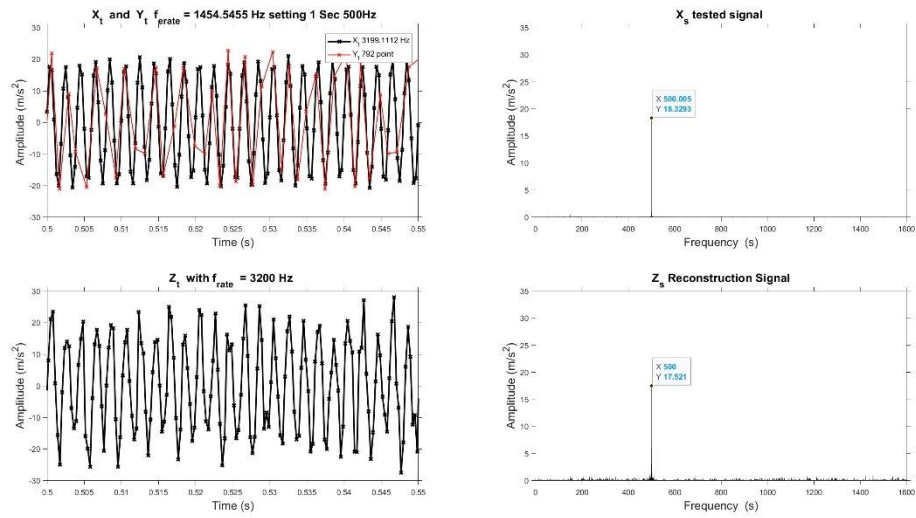
รูปที่ ๑๒.13 การปรับตั้งแบบที่ 1 ที่สัญญาณความถี่ 1500 เฮิรตซ์ มอดูเลชั่น



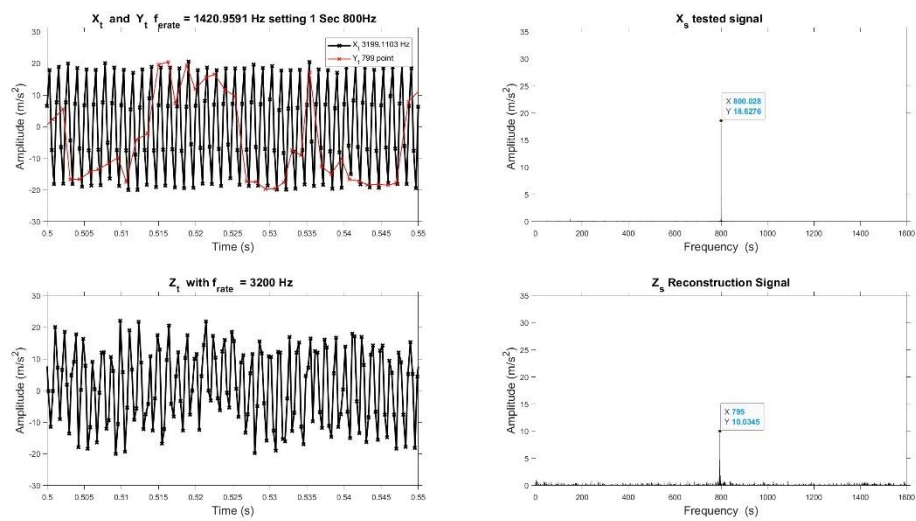
รูปที่ ๑๔.14 การปรับตั้งแบบที่ 1 ที่สัญญาณความถี่ 1800 เฮิรตซ์ มอดูเลชั่น



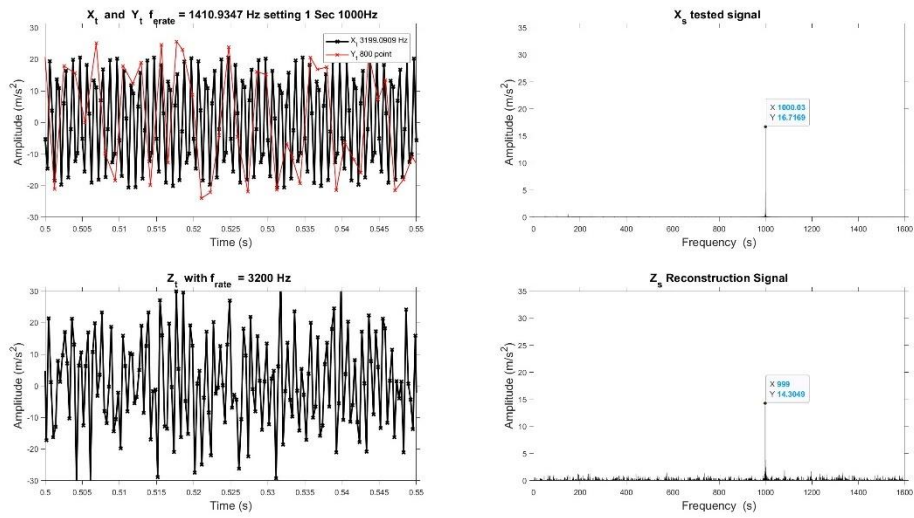
รูปที่ ๑๕.15 การปรับตั้งแบบที่ 2 ที่สัญญาณความถี่ 300 เฮิรตซ์



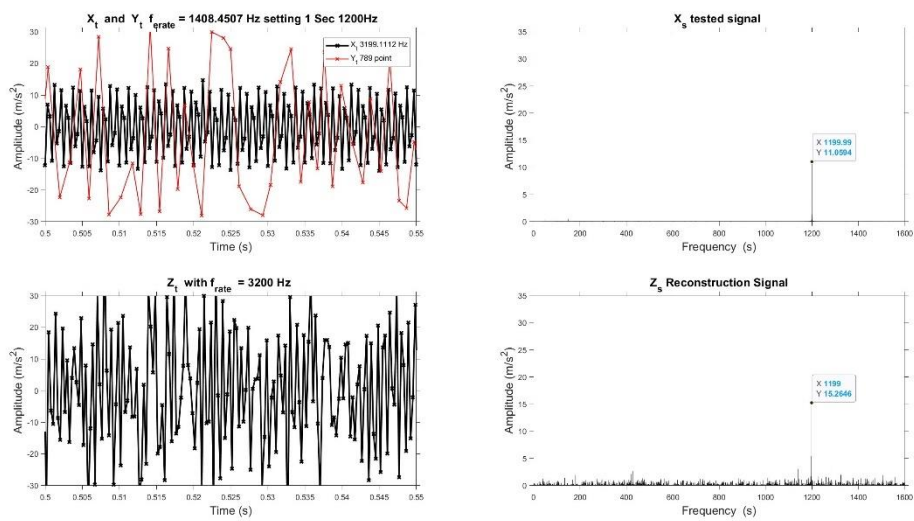
รูปที่ ๑๖. การปรับตั้งแบบที่ 2 ที่สัญญาณความถี่ 500 เฮิรตซ์



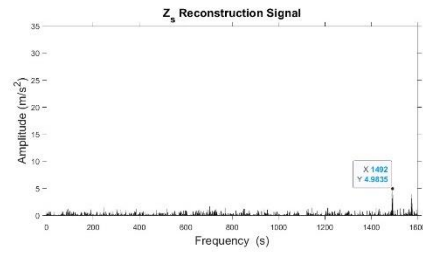
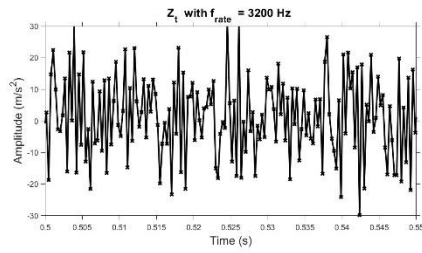
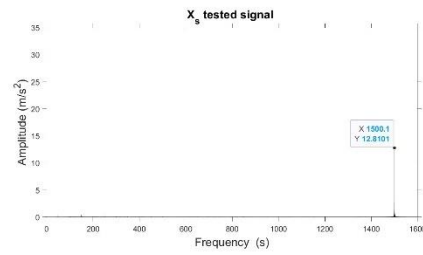
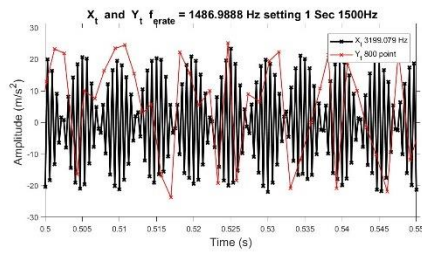
รูปที่ ๑๗. การปรับตั้งแบบที่ 2 ที่สัญญาณความถี่ 800 เฮิรตซ์



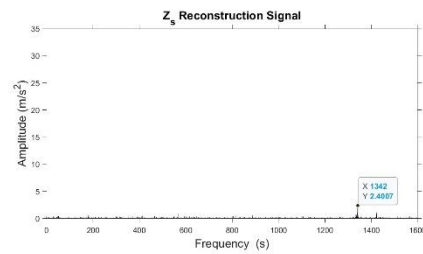
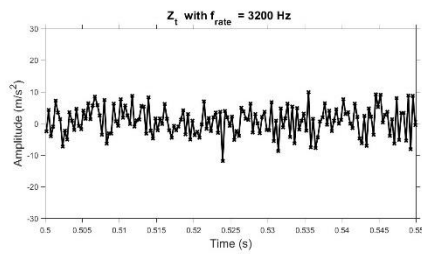
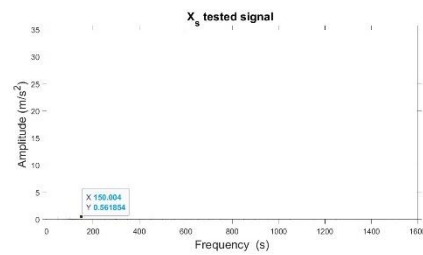
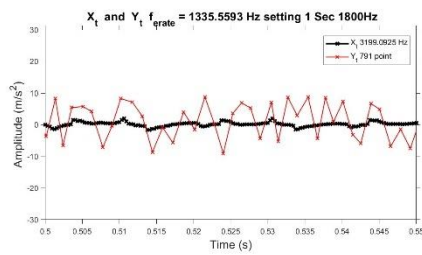
รูปที่ ๑๘ การปรับตั้งแบบที่ 2 ที่สัญญาณความถี่ 1000 เฮิรตซ์



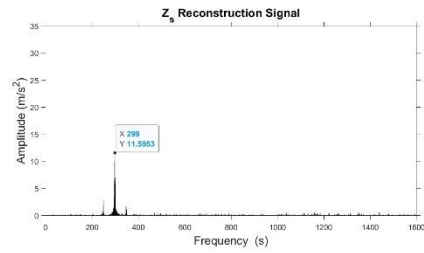
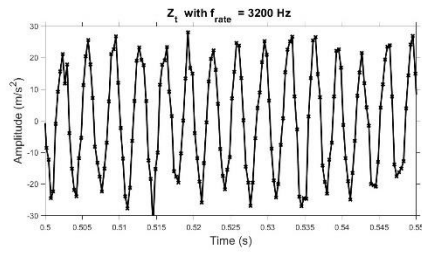
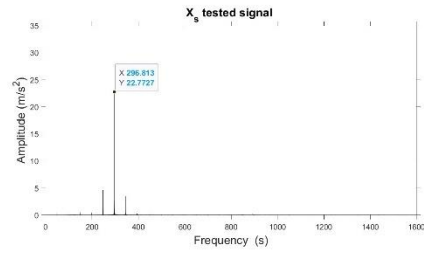
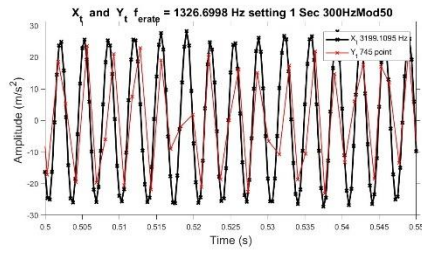
รูปที่ ๑๙ การปรับตั้งแบบที่ 2 ที่สัญญาณความถี่ 1200 เฮิรตซ์



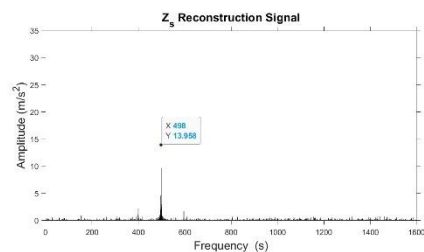
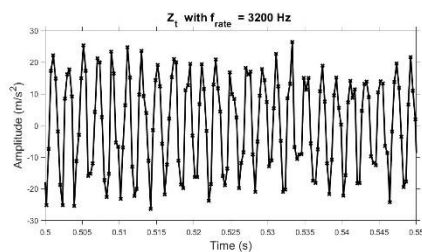
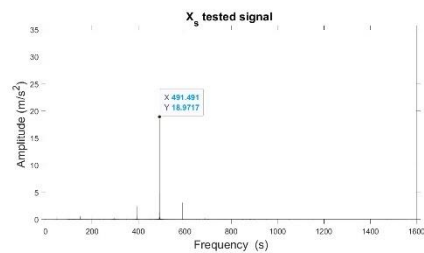
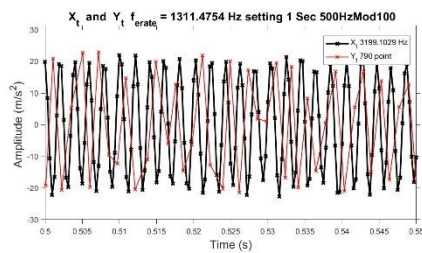
รูปที่ ๓.20 การปรับตั้งแบบที่ 2 ที่สัญญาณความถี่ 1500 เฮิรตซ์



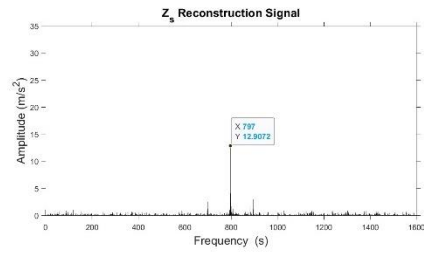
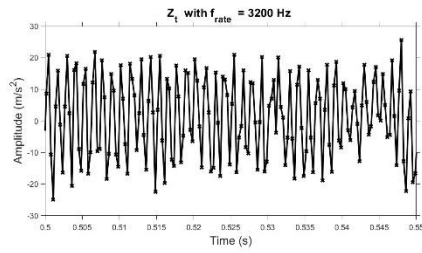
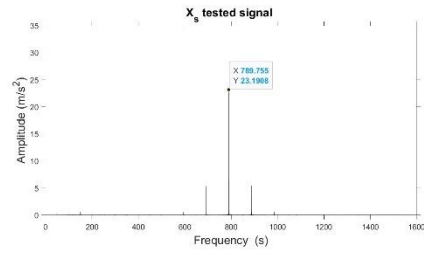
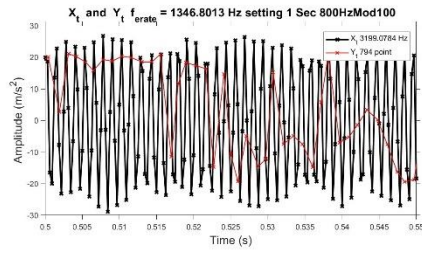
รูปที่ ๓.21 การปรับตั้งแบบที่ 2 ที่สัญญาณความถี่ 1800 เฮิรตซ์



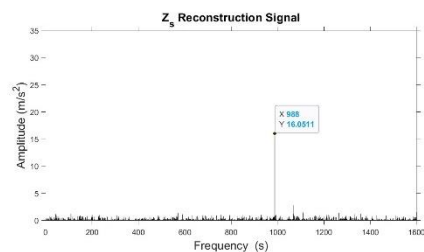
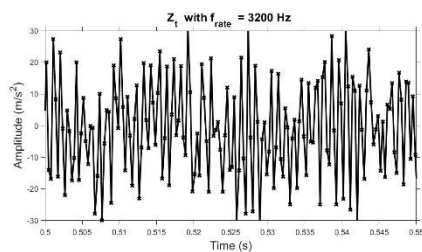
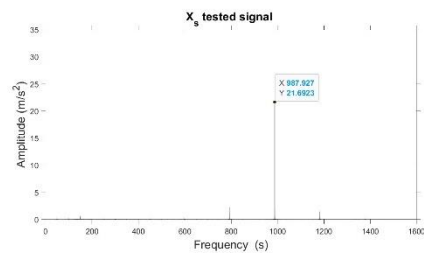
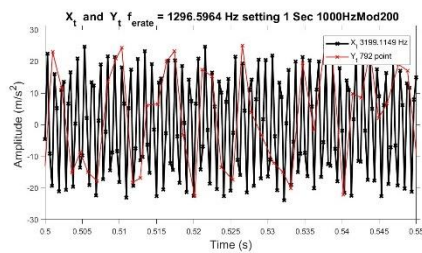
รูปที่ ๓.๒๒ การปรับตั้งแบบที่ 2 ที่สัญญาณความถี่ 300 เฮิรตซ์ มอดูเลชั่น



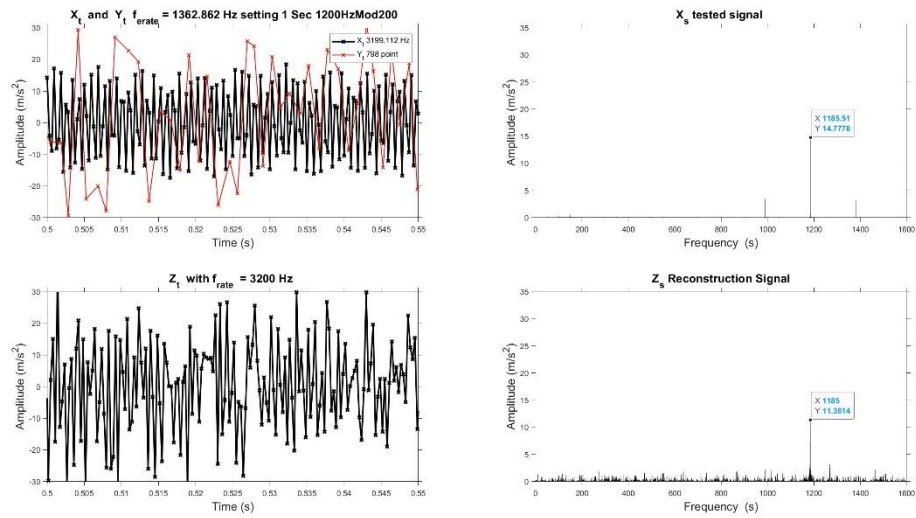
รูปที่ ๓.๒๓ การปรับตั้งแบบที่ 2 ที่สัญญาณความถี่ 500 เฮิรตซ์ มอดูเลชั่น



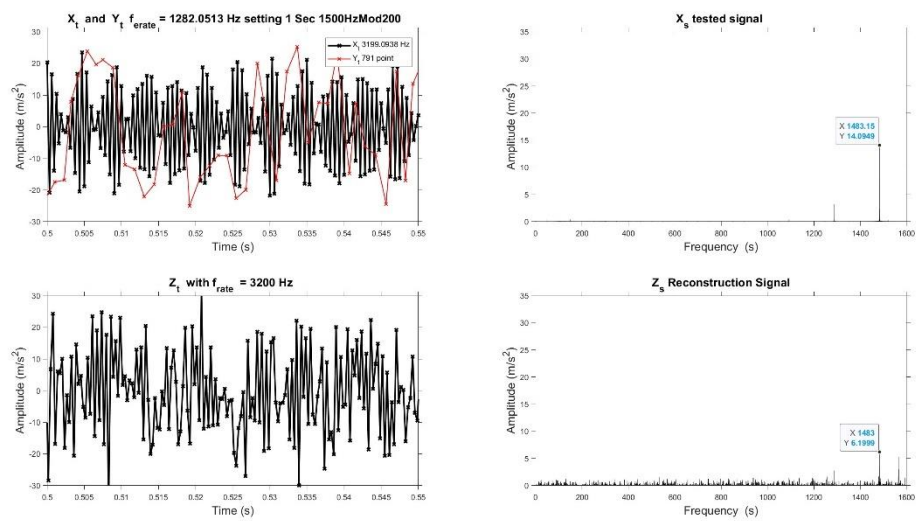
รูปที่ ๓.๒๔ การปรับตั้งแบบที่ ๒ ที่สัญญาณความถี่ 800 เฮิรตซ์ มอดูเลชั่น



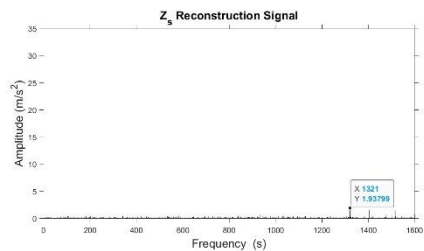
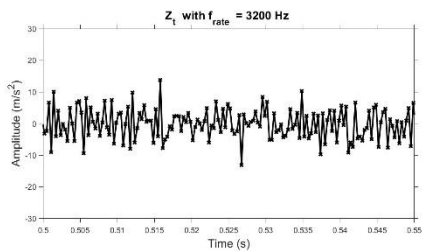
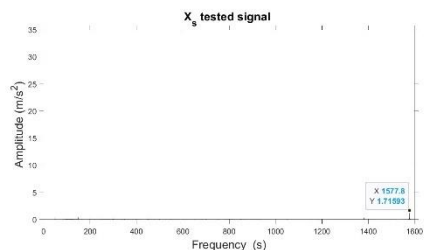
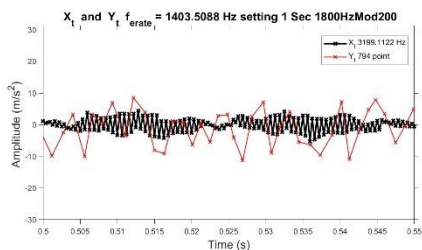
รูปที่ ๓.๒๕ การปรับตั้งแบบที่ ๒ ที่สัญญาณความถี่ 1000 เฮิรตซ์ มอดูเลชั่น



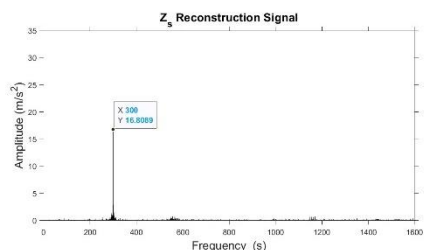
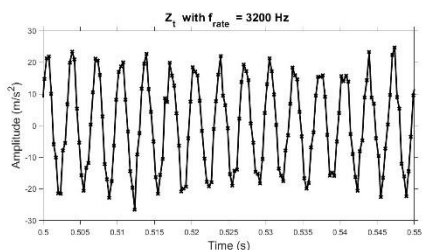
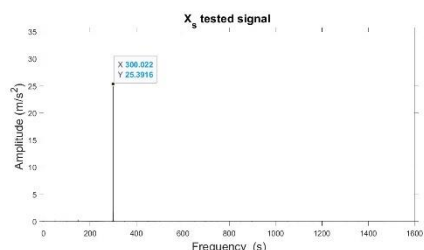
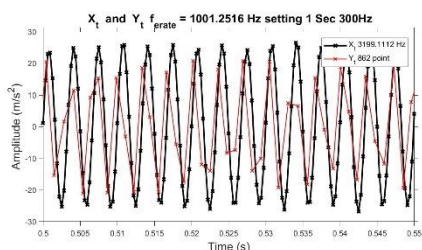
รูปที่ ๓.26 การปรับตั้งแบบที่ 2 ที่สัญญาณความถี่ 1200 เฮิรตซ์ มอดูเลชั่น



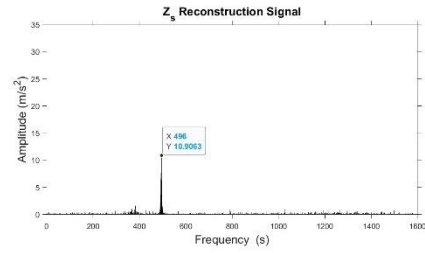
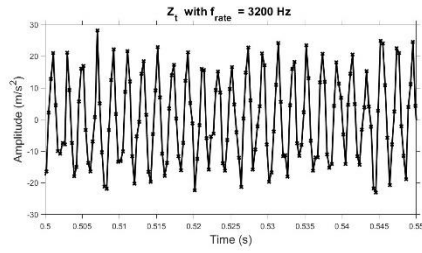
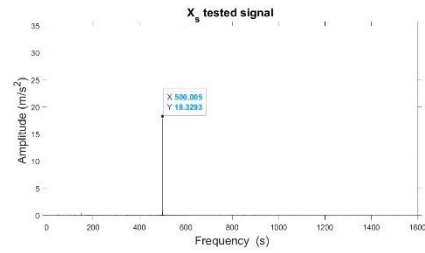
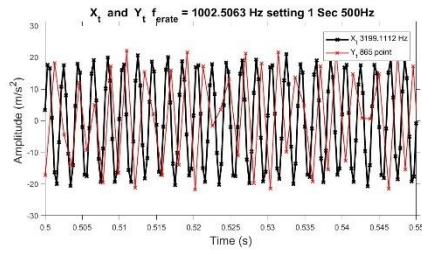
รูปที่ ๓.27 การปรับตั้งแบบที่ 2 ที่สัญญาณความถี่ 1500 เฮิรตซ์ มอดูเลชั่น



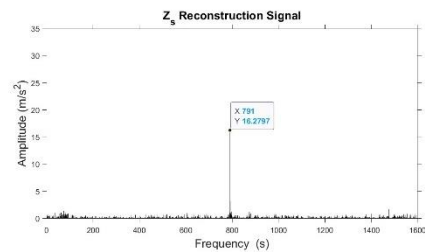
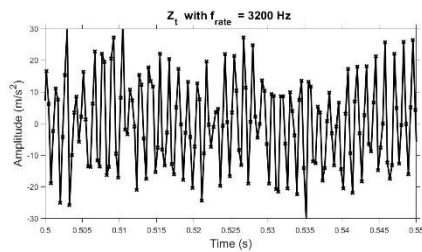
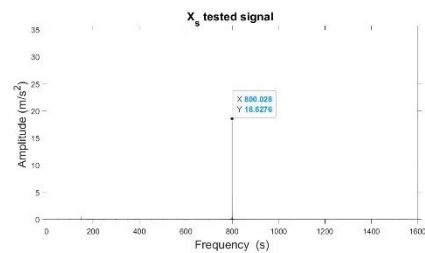
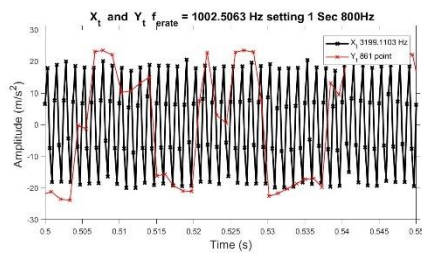
รูปที่ ๓.๒๘ การปรับตั้งแบบที่ ๒ ที่สัญญาณความถี่ 1800 เฮิรตซ์ มอดูเลชั่น



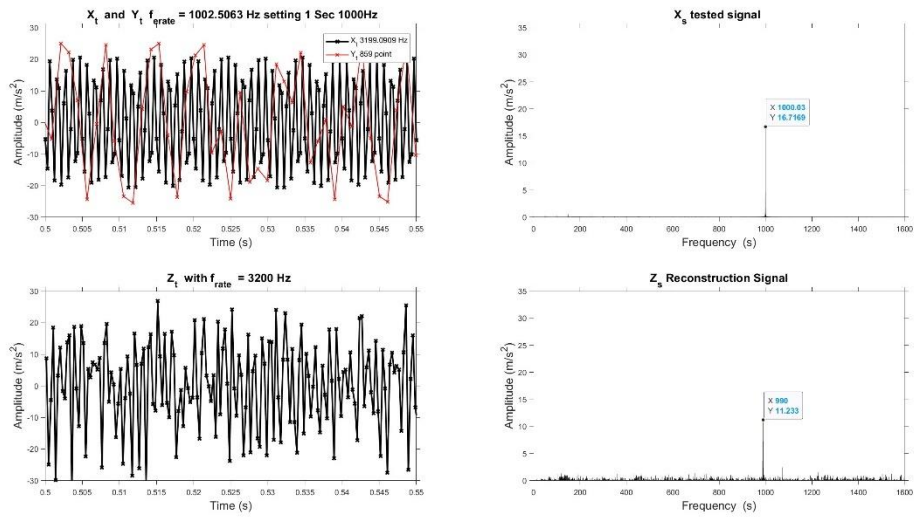
รูปที่ ๓.๒๙ การปรับตั้งแบบที่ ๓ ที่สัญญาณความถี่ 300 เฮิรตซ์



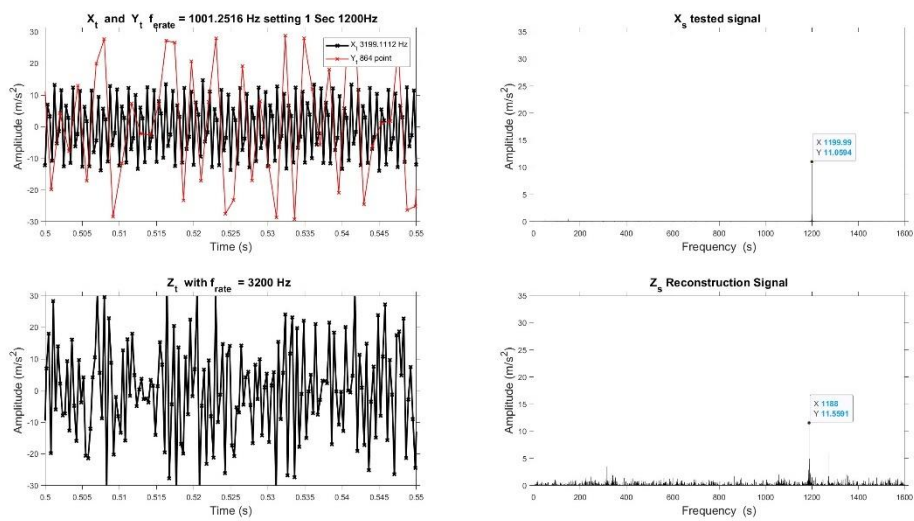
รูปที่ ๓.๓๐ การปรับตั้งแบบที่ 3 ที่สัญญาณความถี่ 500 เฮิรตซ์



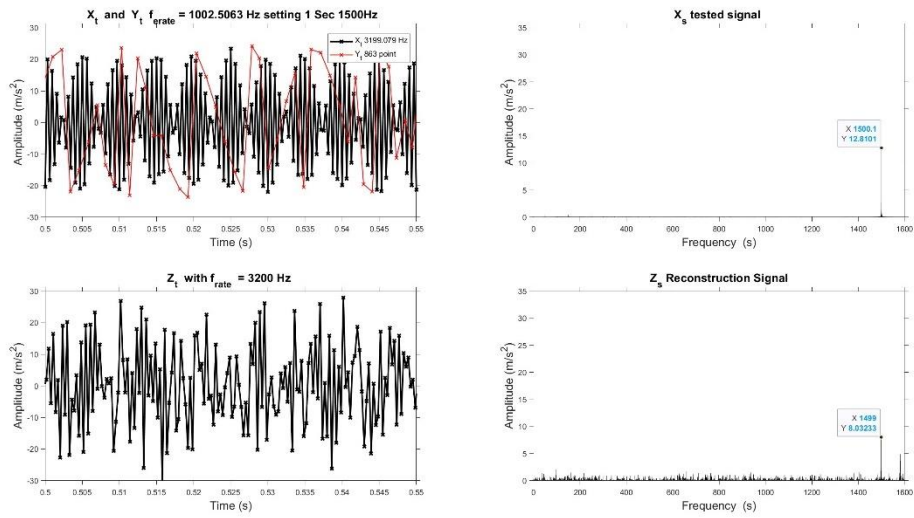
รูปที่ ๓.๓๑ การปรับตั้งแบบที่ 3 ที่สัญญาณความถี่ 800 เฮิรตซ์



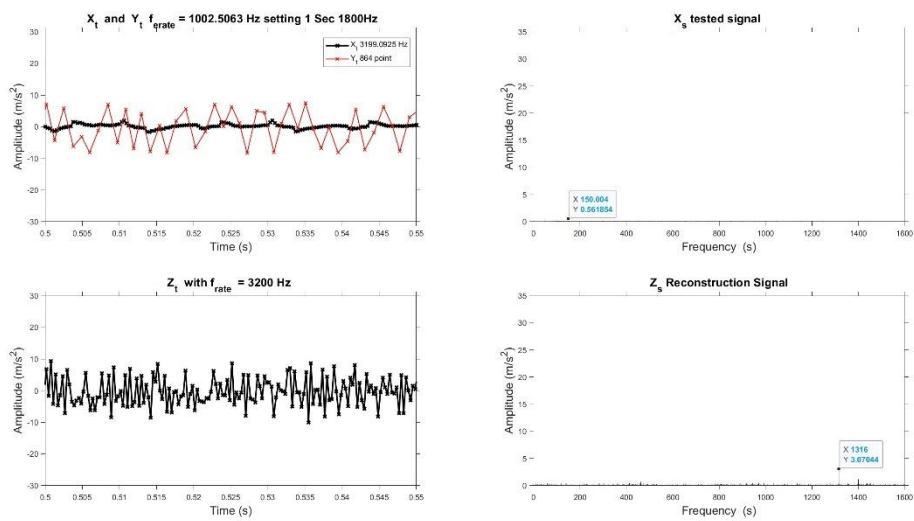
รูปที่ ๓.๓๒ การปรับตั้งแบบที่ 3 ที่สัญญาณความถี่ 1000 เฮิรตซ์



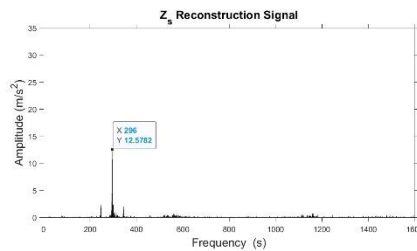
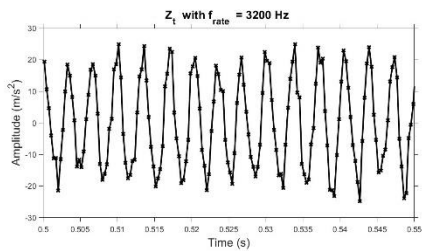
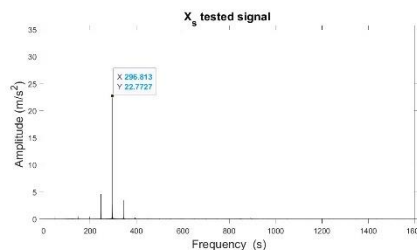
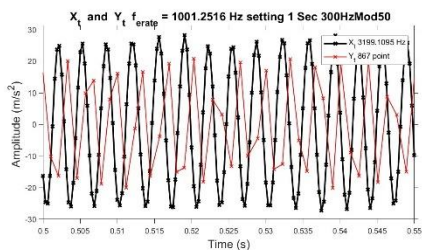
รูปที่ ๓.๓๓ การปรับตั้งแบบที่ 3 ที่สัญญาณความถี่ 1200 เฮิรตซ์



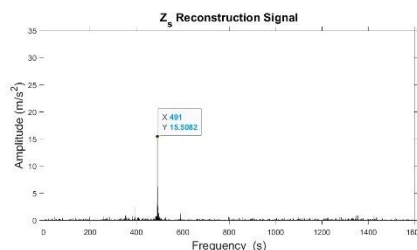
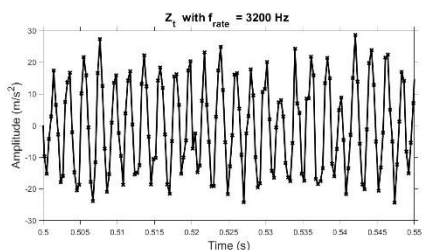
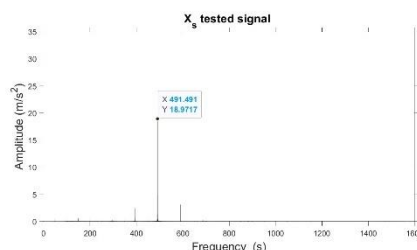
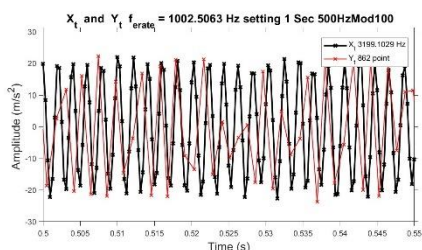
รูปที่ ๓๔ การปรับตั้งแบบที่ 3 ที่สัญญาณความถี่ 1500 เฮิรตซ์



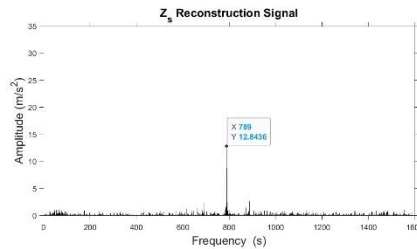
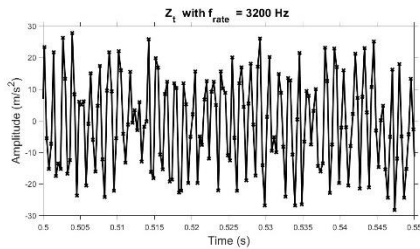
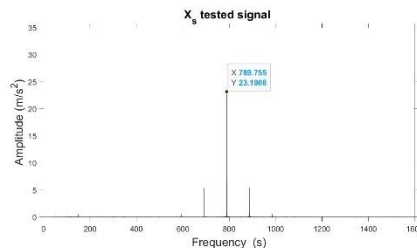
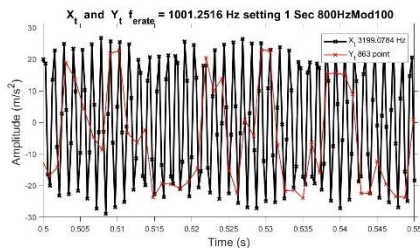
รูปที่ ๓๕ การปรับตั้งแบบที่ 3 ที่สัญญาณความถี่ 1800 เฮิรตซ์



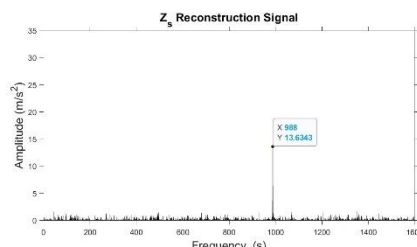
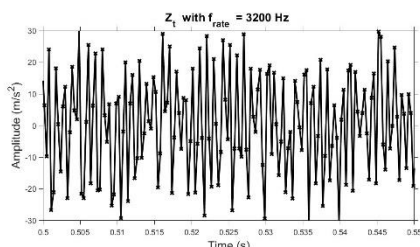
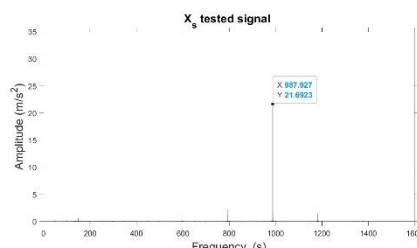
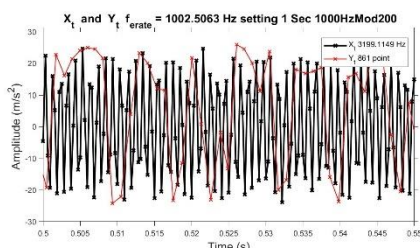
รูปที่ ๓.๓๖ การปรับตั้งแบบที่ 3 ที่สัญญาณความถี่ 300 เฮิรตซ์ มอดูเลชั่น



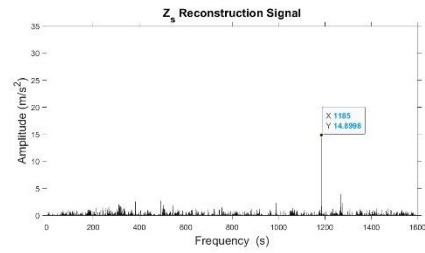
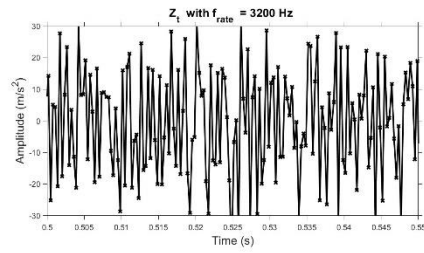
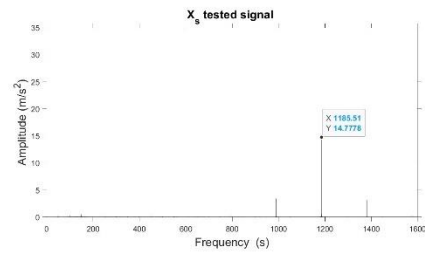
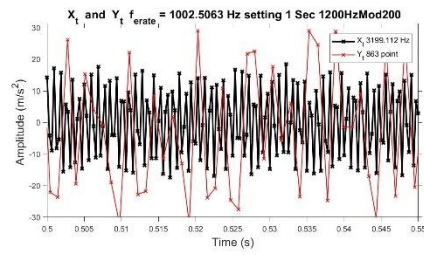
รูปที่ ๓.๓๗ การปรับตั้งแบบที่ 3 ที่สัญญาณความถี่ 500 เฮิรตซ์ มอดูเลชั่น



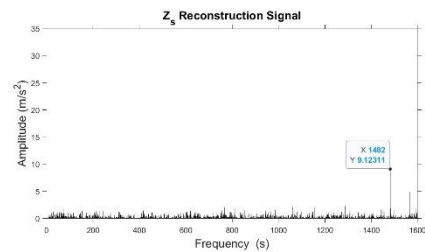
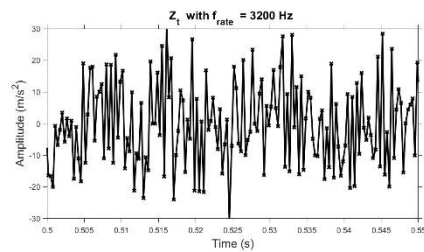
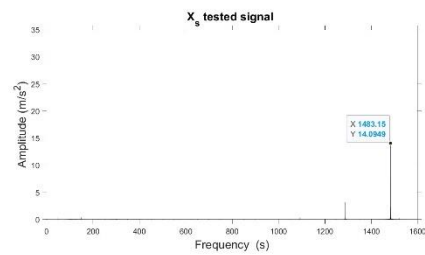
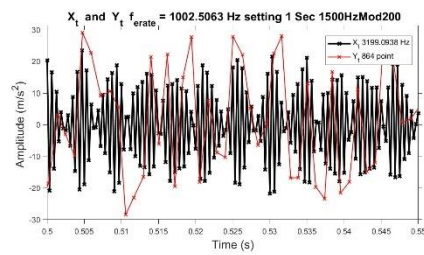
รูปที่ ๓.๓๘ การปรับตั้งแบบที่ 3 ที่สัญญาณความถี่ 800 เฮิรตซ์ มอดูเลชั่น



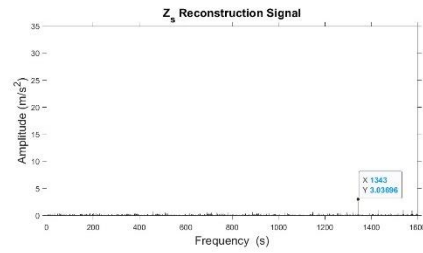
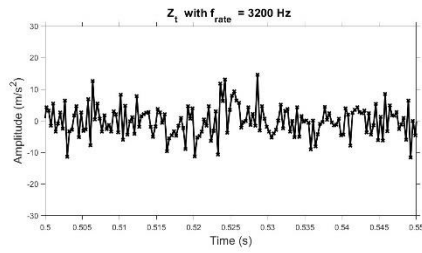
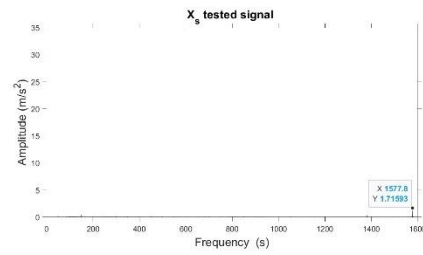
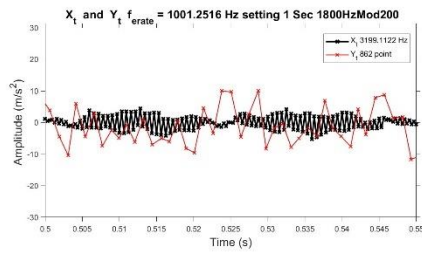
รูปที่ ๓.๓๙ การปรับตั้งแบบที่ 3 ที่สัญญาณความถี่ 1000 เฮิรตซ์ มอดูเลชั่น



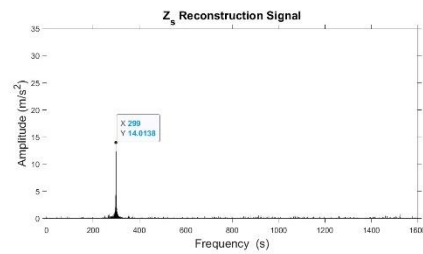
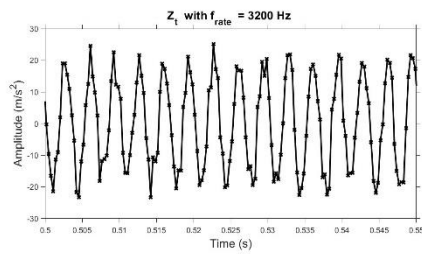
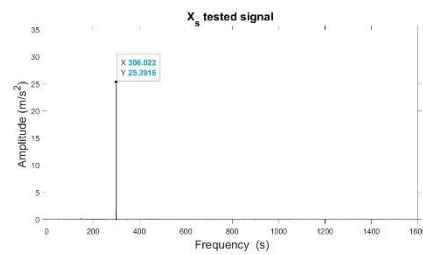
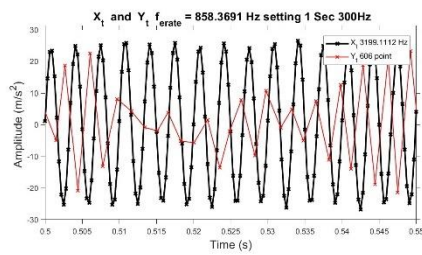
รูปที่ ๓.๔๐ การปรับตั้งแบบที่ 3 ที่สัญญาณความถี่ 1200 เฮิรตซ์ มอดูเลชั่น



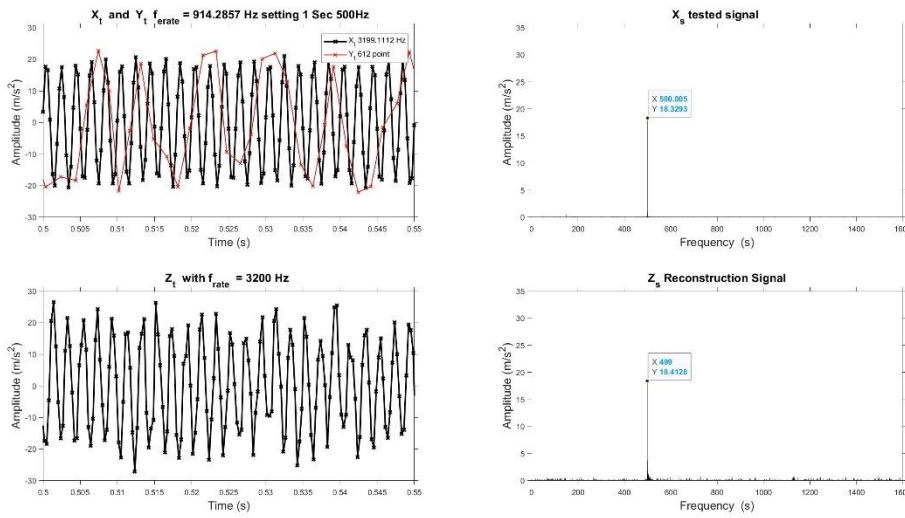
รูปที่ ๓.๔๑ การปรับตั้งแบบที่ 3 ที่สัญญาณความถี่ 1500 เฮิรตซ์ มอดูเลชั่น



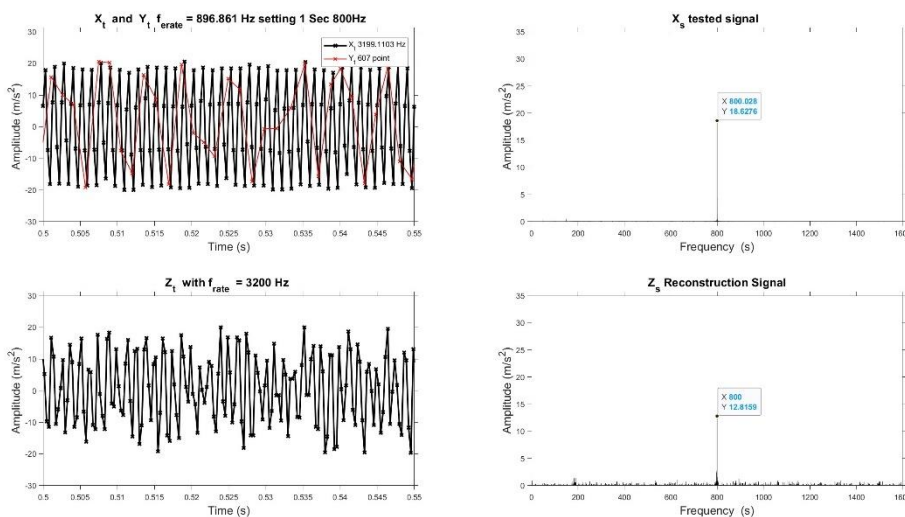
รูปที่ ๓.๔๒ การปรับตั้งแบบที่ 3 ที่สัญญาณความถี่ 1800 เฮิรตซ์ มอดูเลชั่น



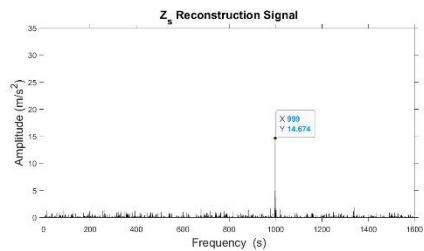
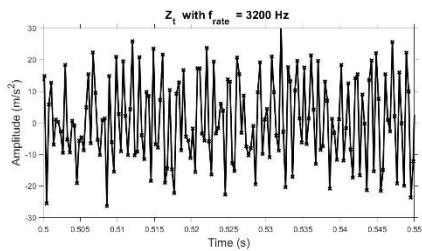
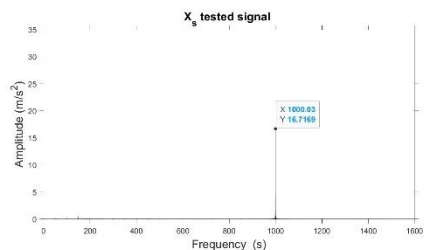
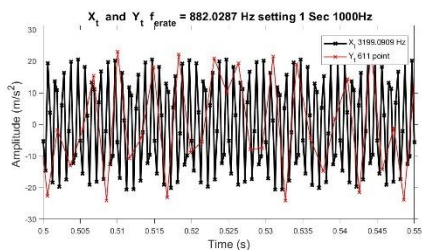
รูปที่ ๓.๔๓ การปรับตั้งแบบที่ 4 ที่สัญญาณความถี่ 300 เฮิรตซ์



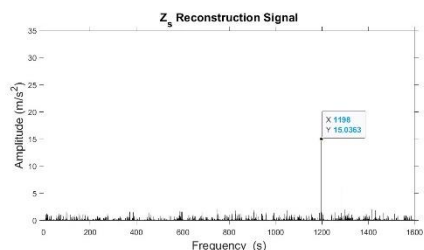
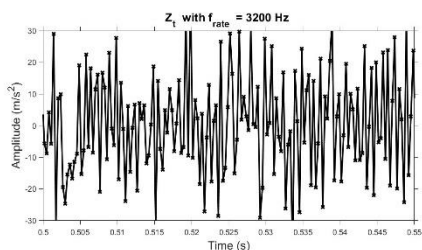
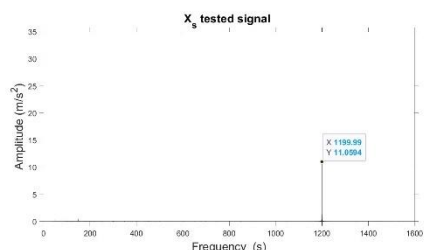
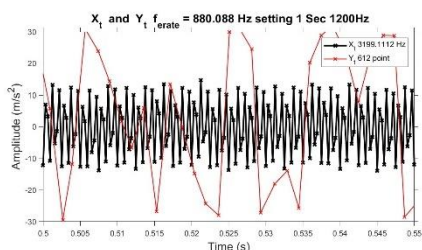
รูปที่ ๓.๔๔ การรับตั้งแบบที่ 4 ที่สัญญาณความถี่ 500 เฮิรตซ์



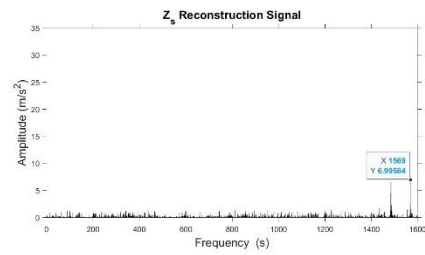
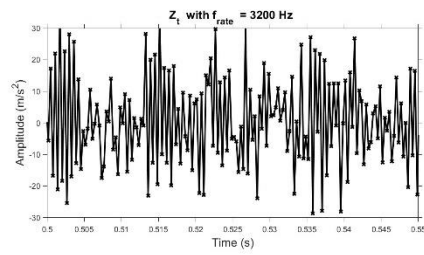
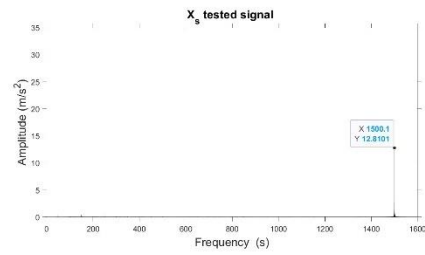
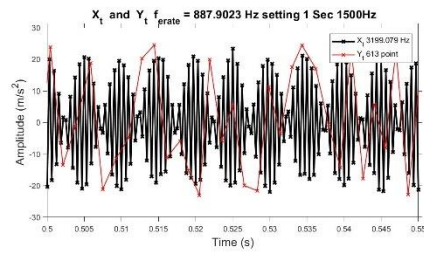
รูปที่ ๓.๔๕ การรับตั้งแบบที่ 4 ที่สัญญาณความถี่ 800 เฮิรตซ์



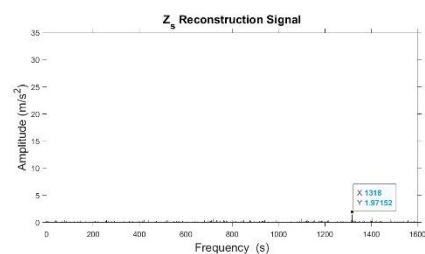
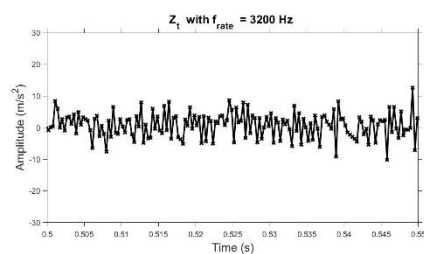
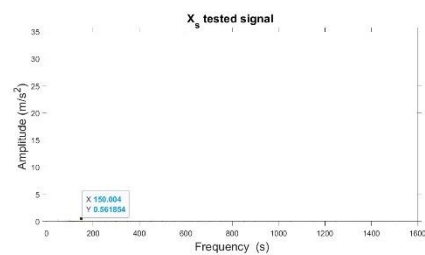
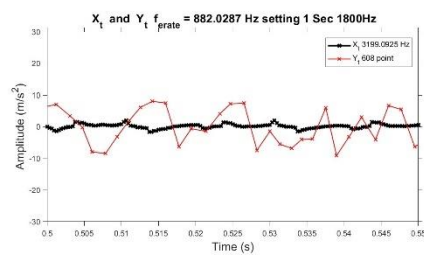
รูปที่ ๓.๔๖ การปรับตั้งแบบที่ 4 ที่สัญญาณความถี่ 1000 เฮิรตซ์



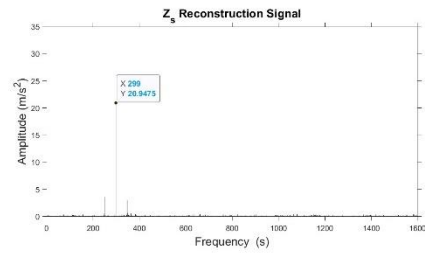
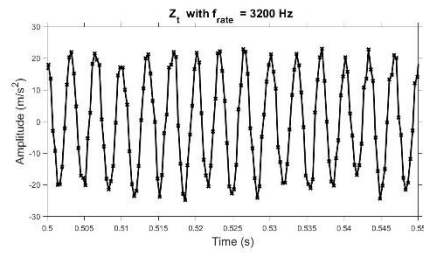
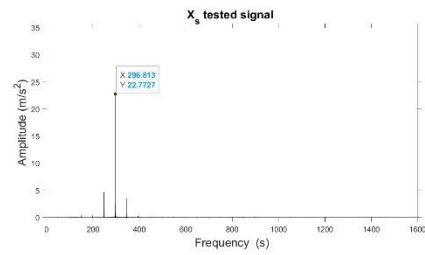
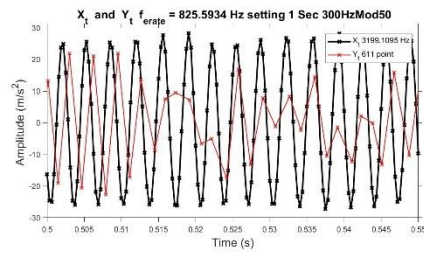
รูปที่ ๓.๔๗ การปรับตั้งแบบที่ 4 ที่สัญญาณความถี่ 1200 เฮิรตซ์



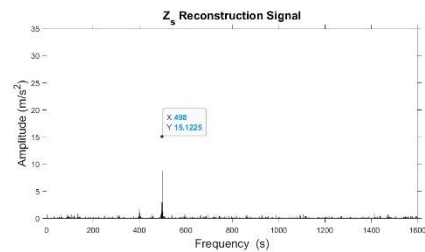
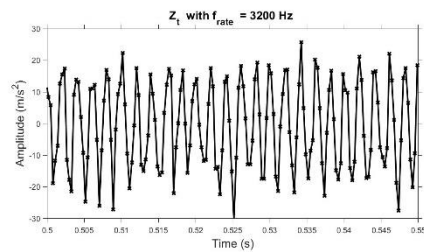
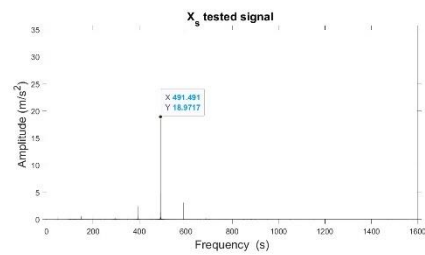
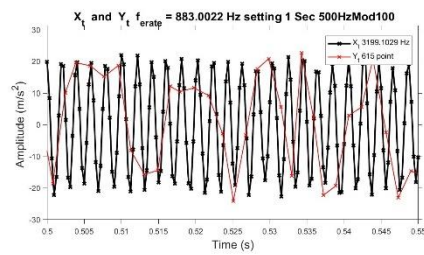
รูปที่ ๓.๔๘ การปรับตั้งแบบที่ 4 ที่สัญญาณความถี่ 1500 เฮิรตซ์



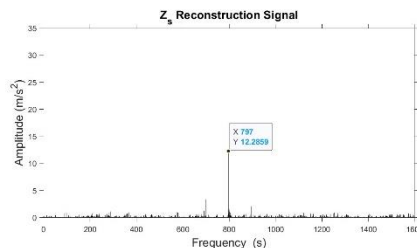
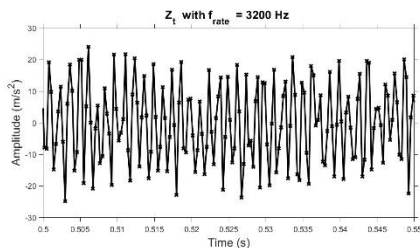
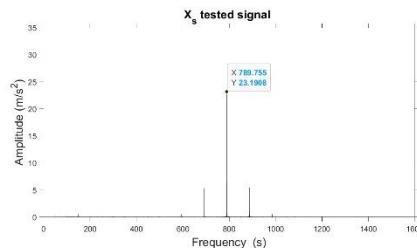
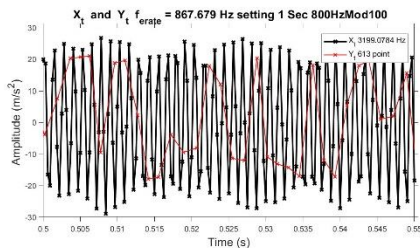
รูปที่ ๓.๔๙ การปรับตั้งแบบที่ 4 ที่สัญญาณความถี่ 1800 เฮิรตซ์



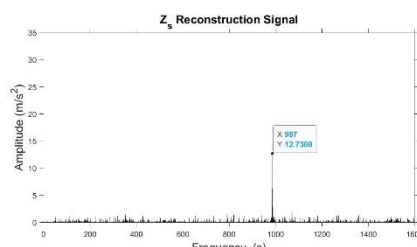
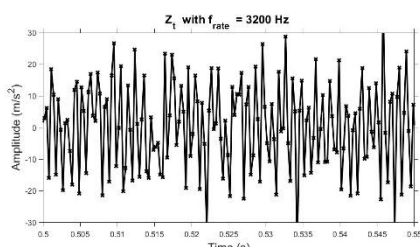
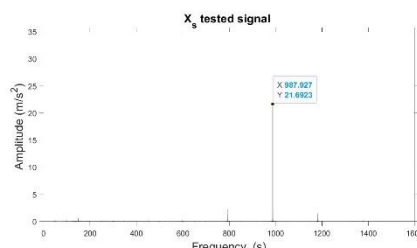
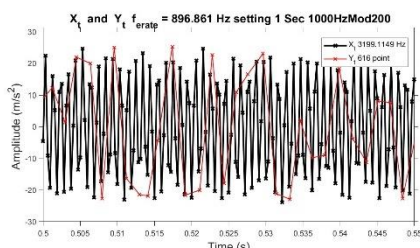
รูปที่ ๕๐ การปรับตั้งแบบที่ 4 ที่สัญญาณความถี่ 300 เฮิรตซ์ มอดูเลชั่น



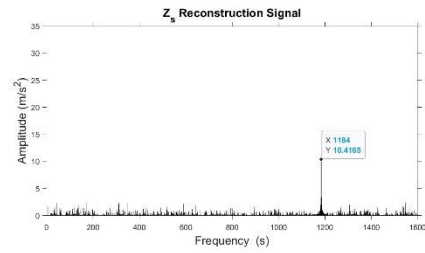
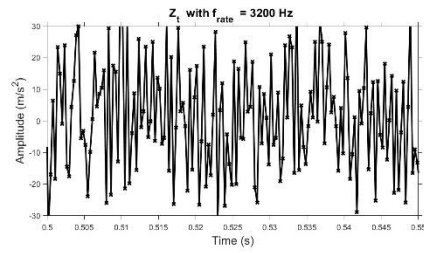
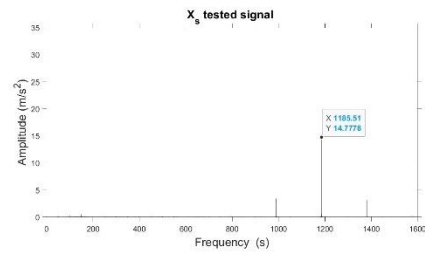
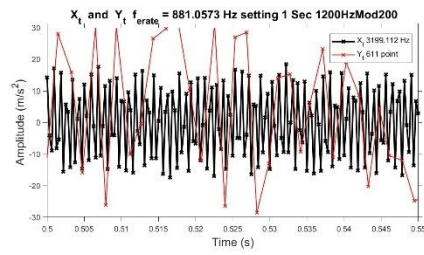
รูปที่ ๕๑ การปรับตั้งแบบที่ 4 ที่สัญญาณความถี่ 500 เฮิรตซ์ มอดูเลชั่น



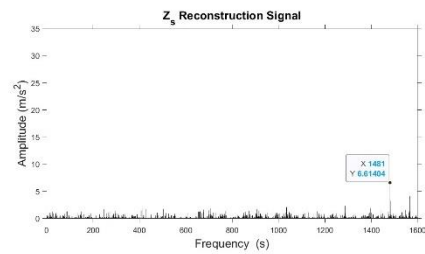
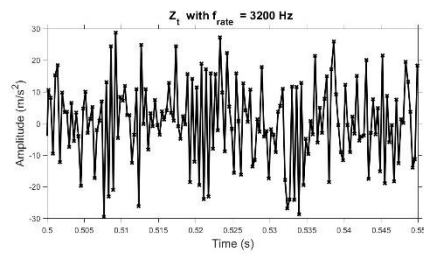
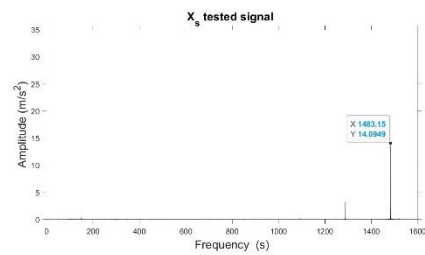
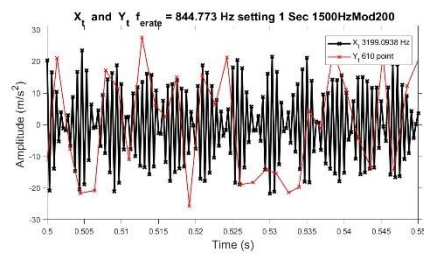
รูปที่ ๕๒ การปรับตั้งแบบที่ 4 ที่สัญญาณความถี่ 800 เฮิรตซ์ มอดูเลชัน



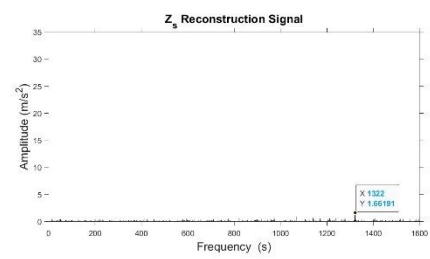
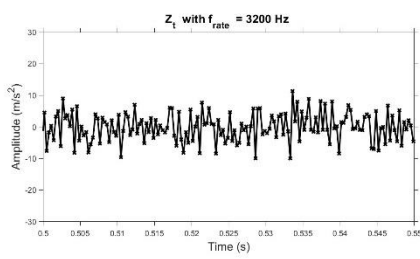
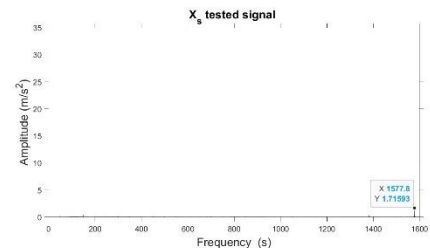
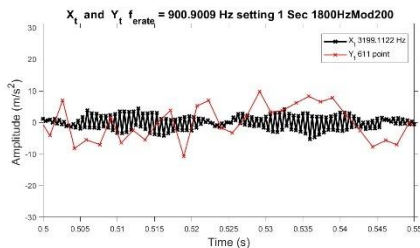
รูปที่ ๕๓ การปรับตั้งแบบที่ 4 ที่สัญญาณความถี่ 1000 เฮิรตซ์ มอดูเลชัน



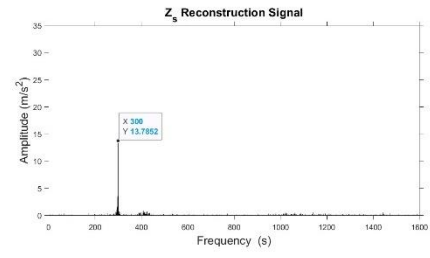
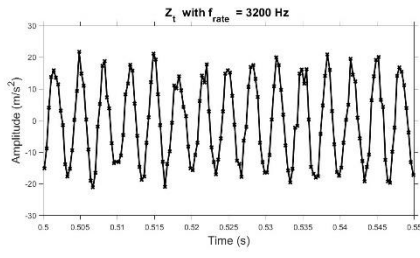
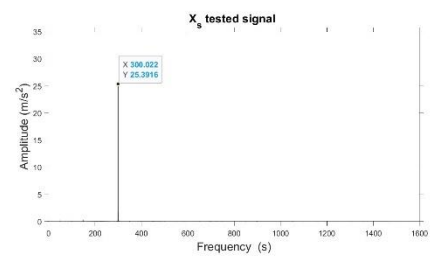
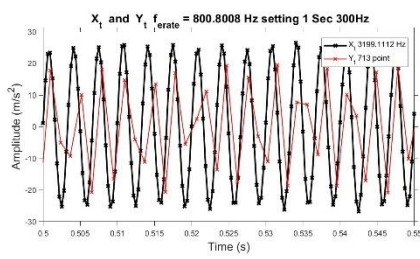
รูปที่ ๓.54 การปรับตั้งแบบที่ 4 ที่สัญญาณความถี่ 1200 เฮิรตซ์ มอดูเลชั่น



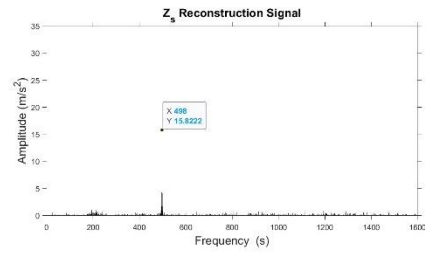
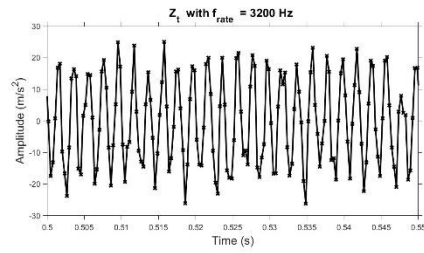
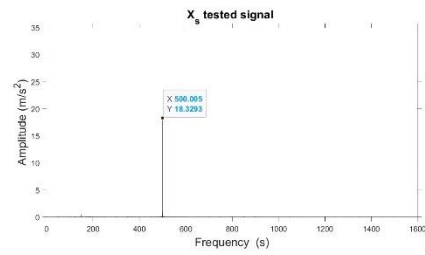
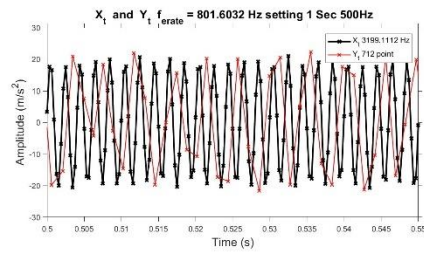
รูปที่ ๓.55 การปรับตั้งแบบที่ 4 ที่สัญญาณความถี่ 1500 เฮิรตซ์ มอดูเลชั่น



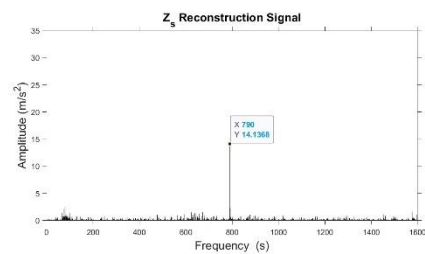
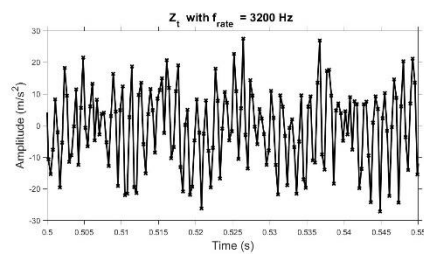
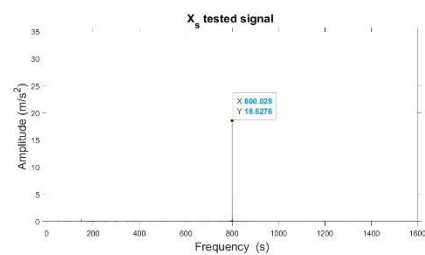
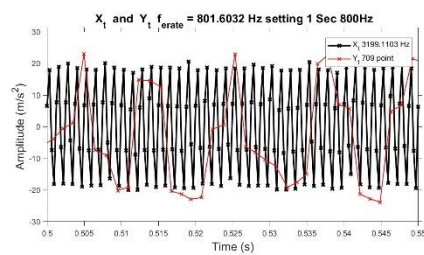
รูปที่ ๓.56 การปรับตั้งแบบที่ 4 ที่สัญญาณความถี่ 1800 เฮิรตซ์ มอดูเลชั่น



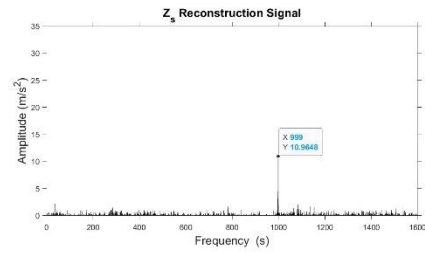
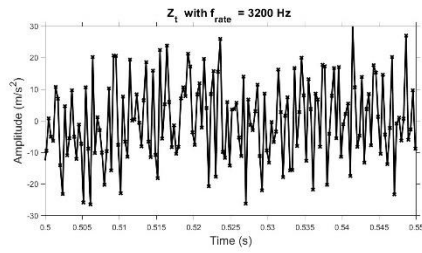
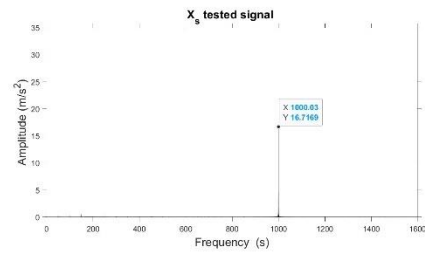
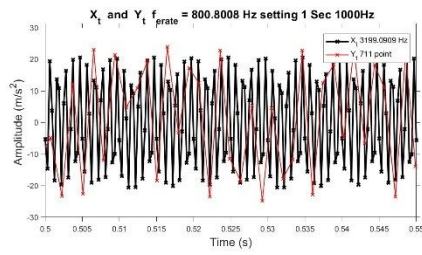
รูปที่ ๓.57 การปรับตั้งแบบที่ 5 ที่สัญญาณความถี่ 300 เฮิรตซ์



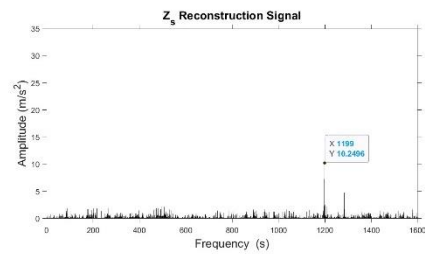
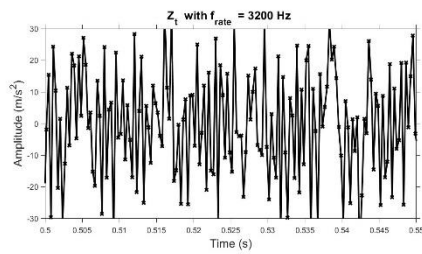
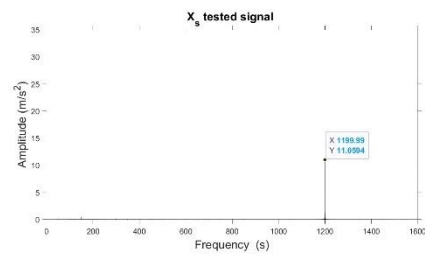
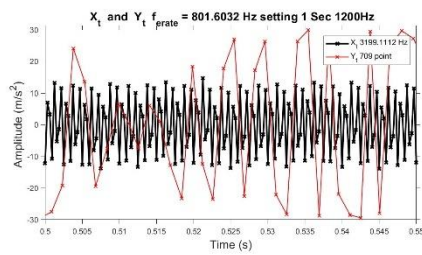
รูปที่ ๕.๕๘ การรับตั้งแบบที่ 5 ที่สัญญาณความถี่ 500 เฮิรตซ์



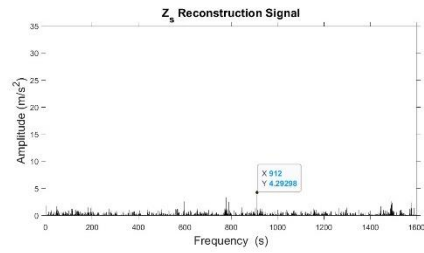
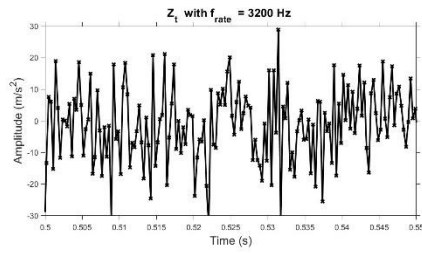
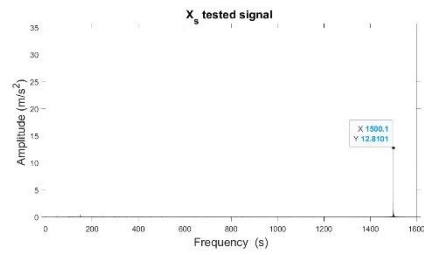
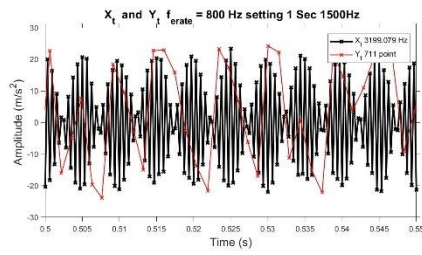
รูปที่ ๕.๕๙ การรับตั้งแบบที่ 5 ที่สัญญาณความถี่ 800 เฮิรตซ์



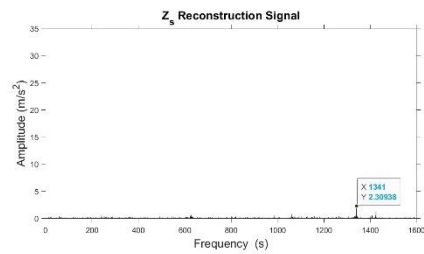
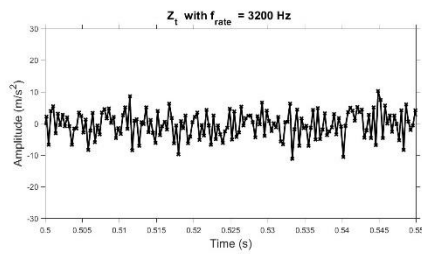
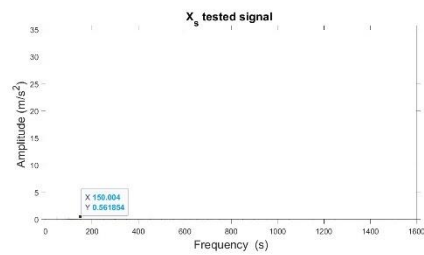
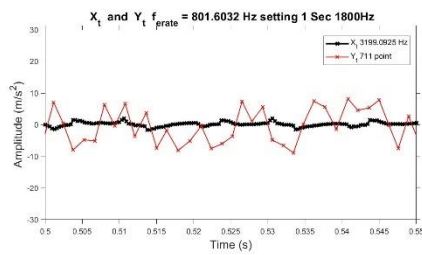
รูปที่ ๓.๖๐ การปรับตั้งแบบที่ 5 ที่สัญญาณความถี่ 1000 เฮิรตซ์



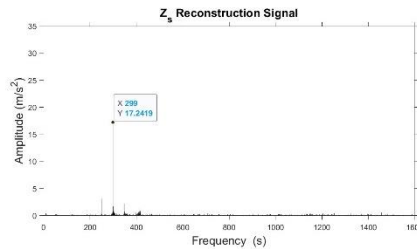
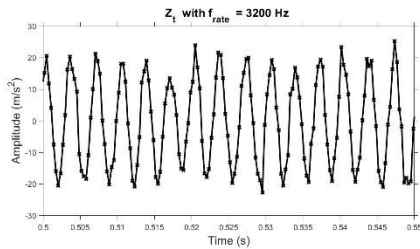
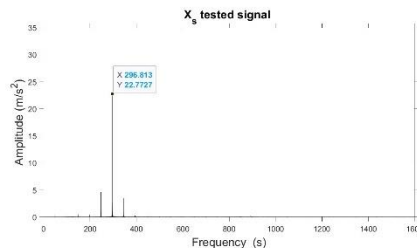
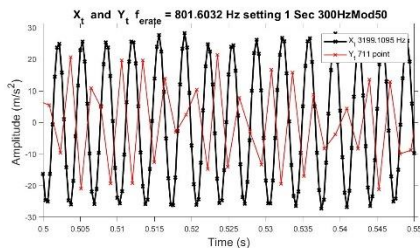
รูปที่ ๓.๖๑ การปรับตั้งแบบที่ 5 ที่สัญญาณความถี่ 1200 เฮิรตซ์



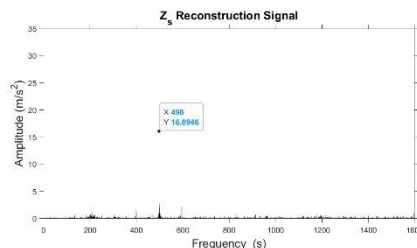
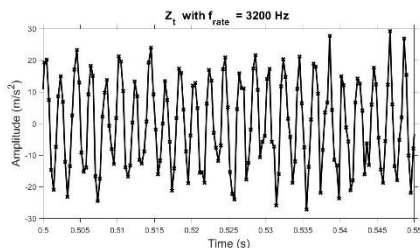
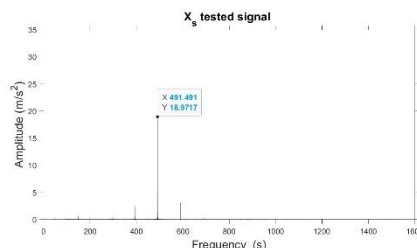
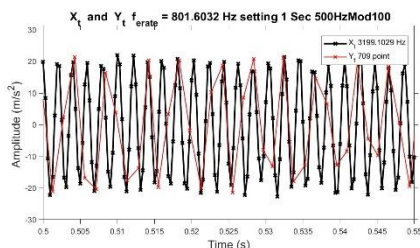
รูปที่ ๓.๖๒ การปรับตั้งแบบที่ 5 ที่สัญญาณความถี่ 1500 เฮิรตซ์



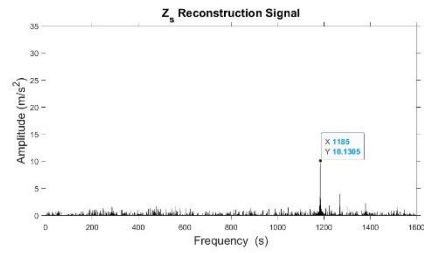
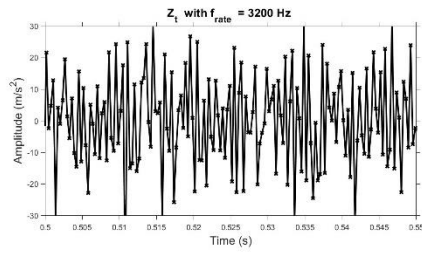
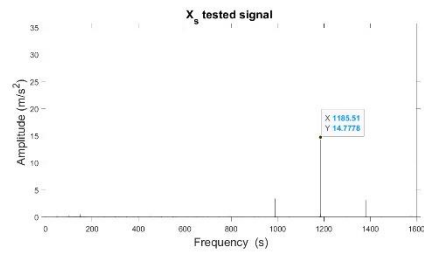
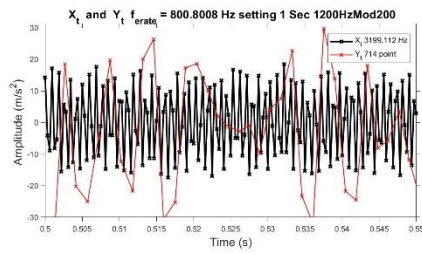
รูปที่ ๓.๖๓ การปรับตั้งแบบที่ 5 ที่สัญญาณความถี่ 1800 เฮิรตซ์



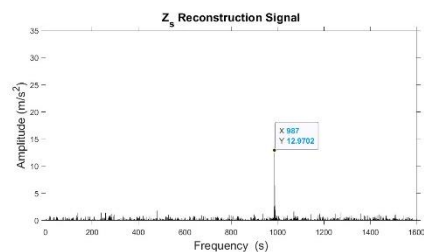
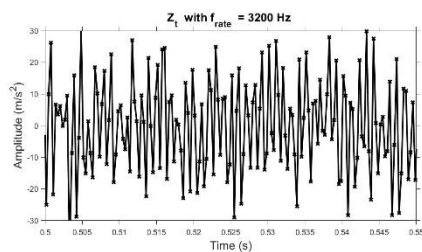
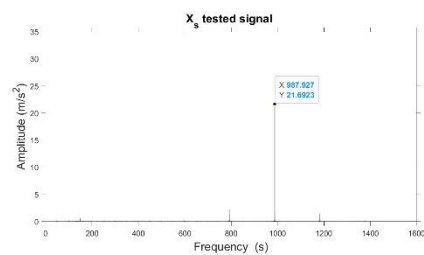
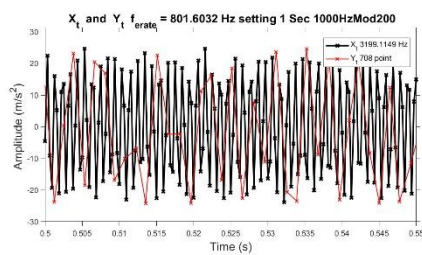
รูปที่ ๖๔ การปรับตั้งแบบที่ 5 ที่สัญญาณความถี่ 300 เฮิรตซ์ มอดูเลชั่น



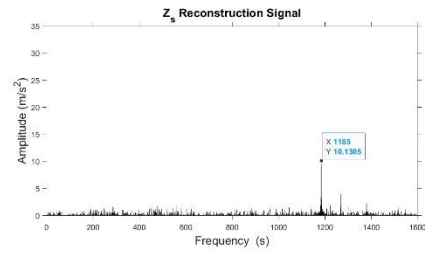
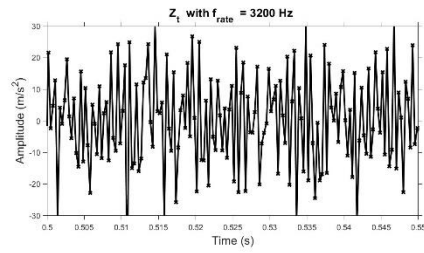
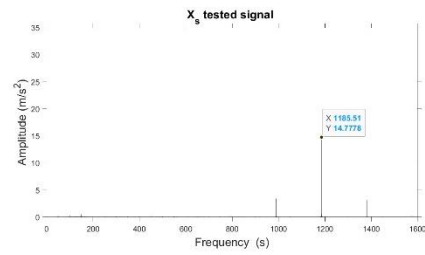
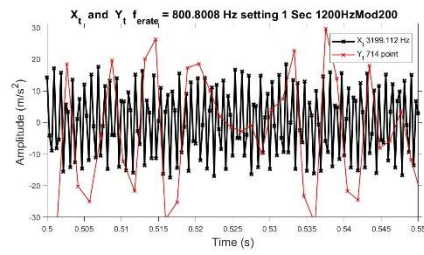
รูปที่ ๖๕ การปรับตั้งแบบที่ 5 ที่สัญญาณความถี่ 500 เฮิรตซ์ มอดูเลชั่น



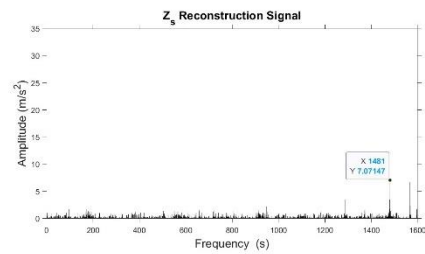
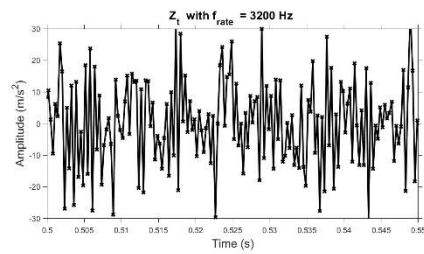
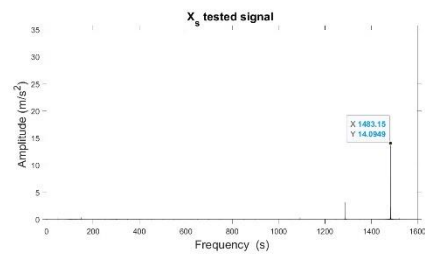
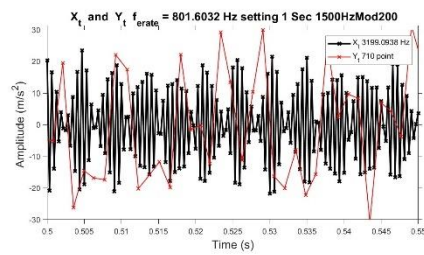
รูปที่ ๓.๖๖ การปรับตั้งแบบที่ 5 ที่สัญญาณความถี่ 800 เฮิรตซ์ มอดูเลชั่น



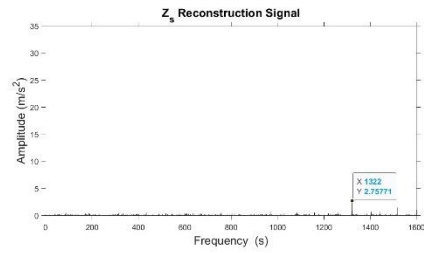
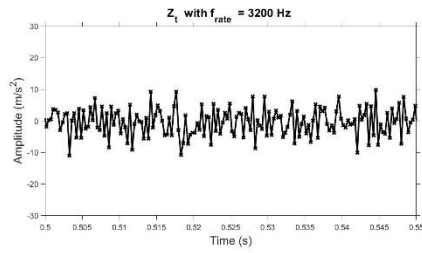
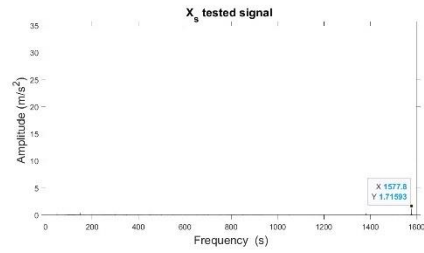
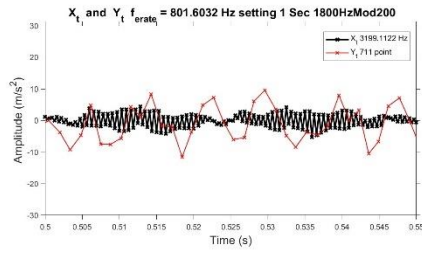
รูปที่ ๓.๖๗ การปรับตั้งแบบที่ 5 ที่สัญญาณความถี่ 1000 เฮิรตซ์ มอดูเลชั่น



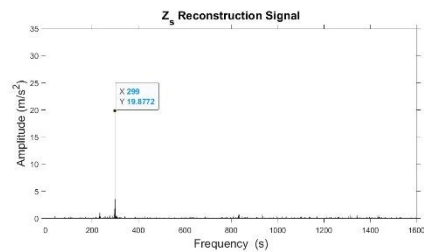
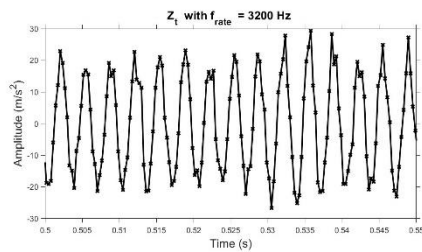
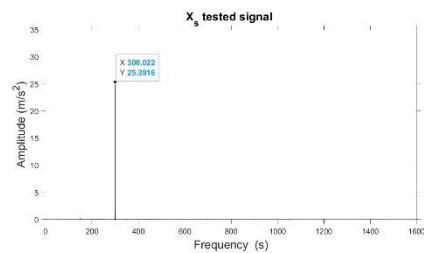
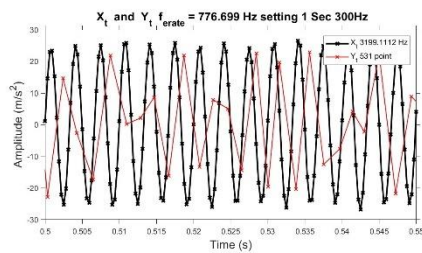
รูปที่ ๘.๖๘ การปรับตั้งแบบที่ 5 ที่สัญญาณความถี่ 1200 เฮิรตซ์ มอดูเลชั่น



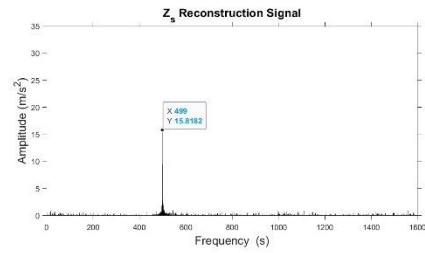
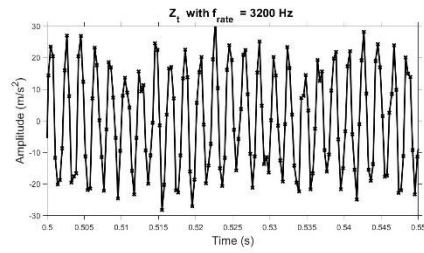
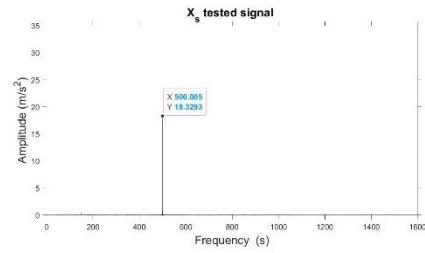
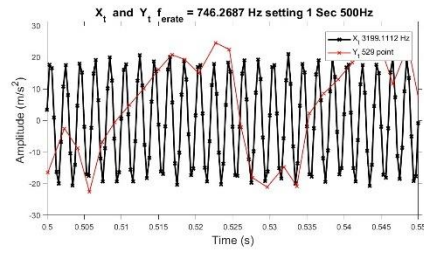
รูปที่ ๘.๖๙ การปรับตั้งแบบที่ 5 ที่สัญญาณความถี่ 1500 เฮิรตซ์ มอดูเลชั่น



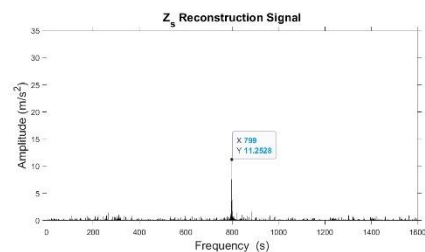
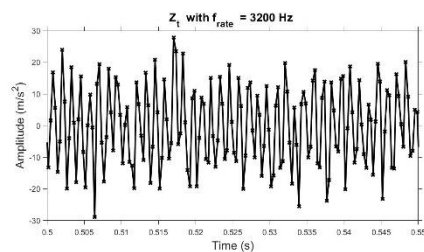
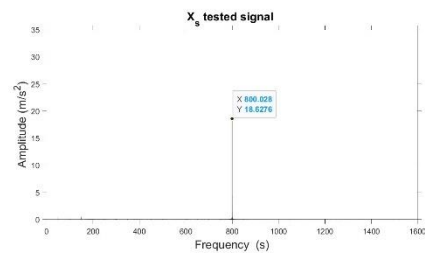
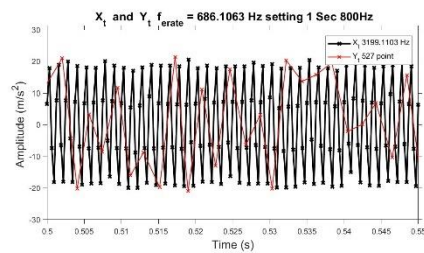
รูปที่ ๗.๗๐ การปรับตั้งแบบที่ ๕ ที่สัญญาณความถี่ 1800 เฮิรตซ์ มอดูเลชั่น



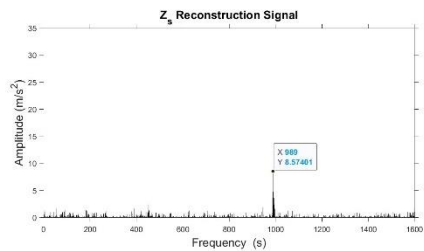
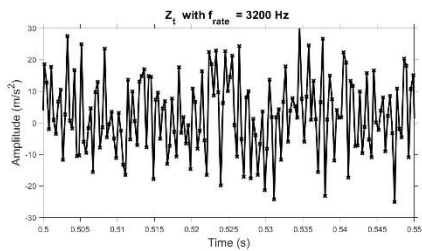
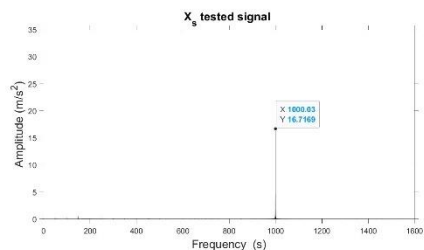
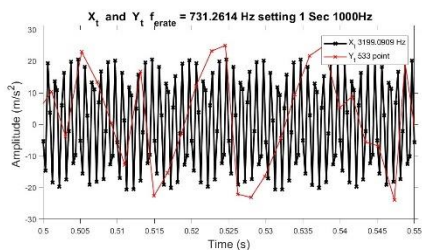
รูปที่ ๗.๗๑ การปรับตั้งแบบที่ ๖ ที่สัญญาณความถี่ 300 เฮิรตซ์



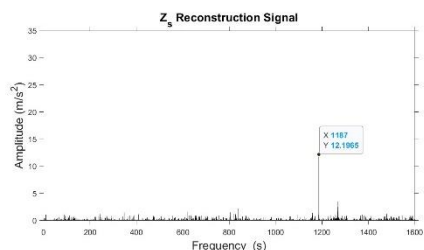
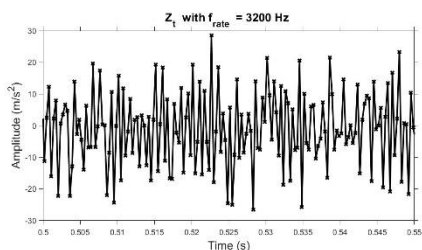
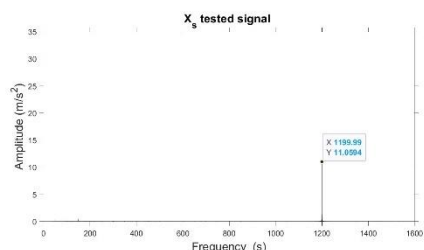
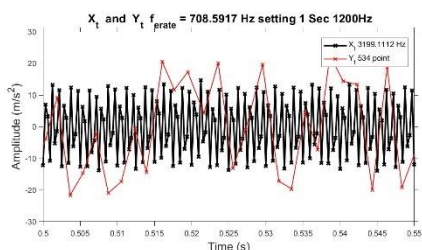
รูปที่ ๗.๗๒ การปรับตั้งแบบที่ 6 ที่สัญญาณความถี่ 500 เฮิรตซ์



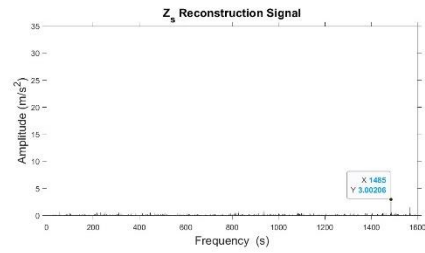
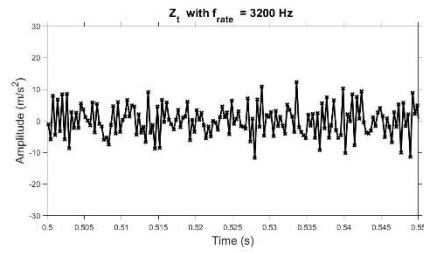
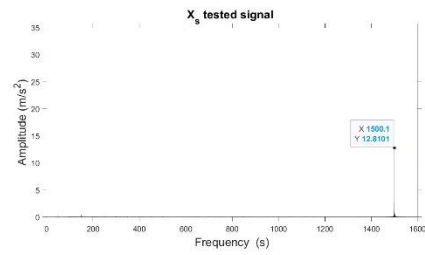
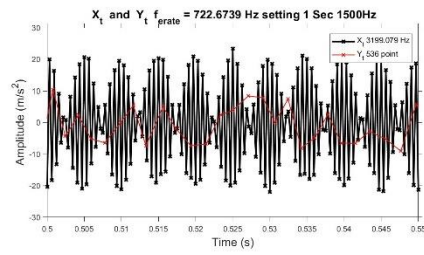
รูปที่ ๗.๗๓ การปรับตั้งแบบที่ 6 ที่สัญญาณความถี่ 800 เฮิรตซ์



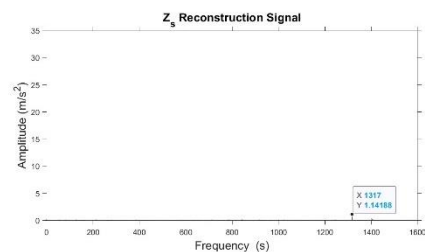
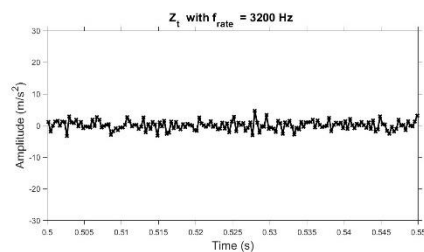
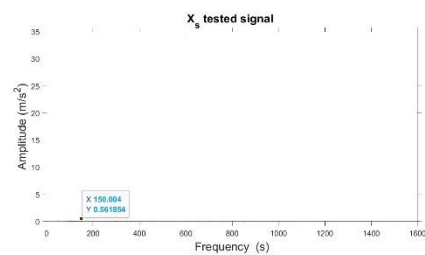
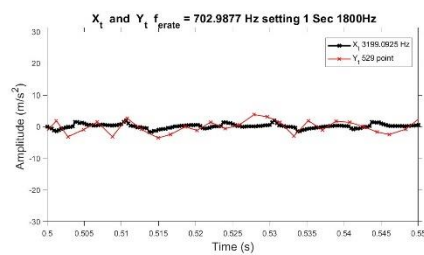
รูปที่ ๗๔ การปรับตั้งแบบที่ 6 ที่สัญญาณความถี่ 1000 เฮิรตซ์



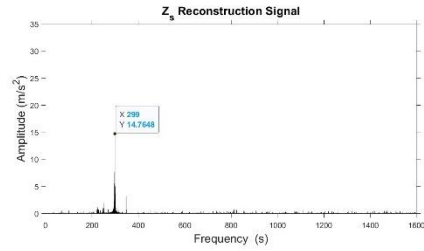
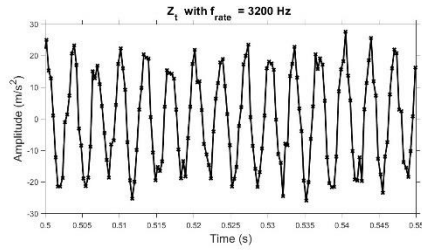
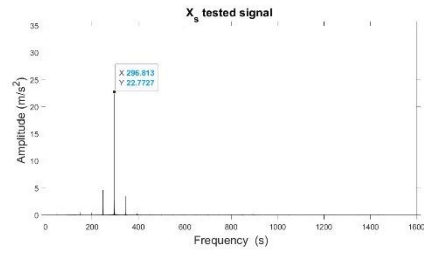
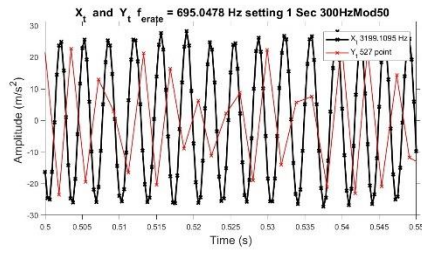
รูปที่ ๗๕ การปรับตั้งแบบที่ 6 ที่สัญญาณความถี่ 1200 เฮิรตซ์



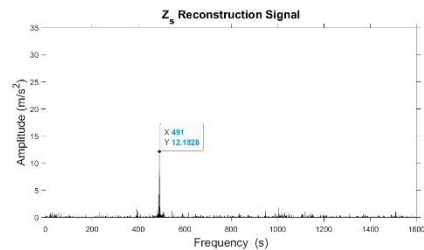
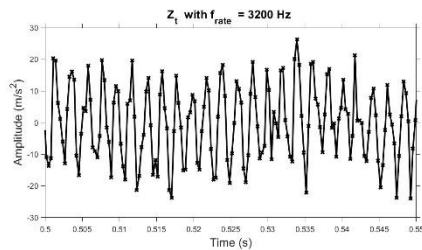
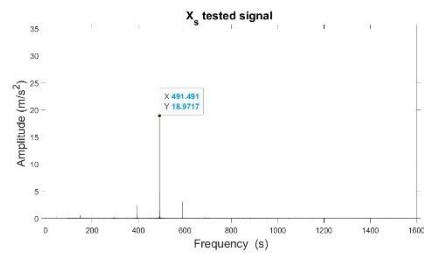
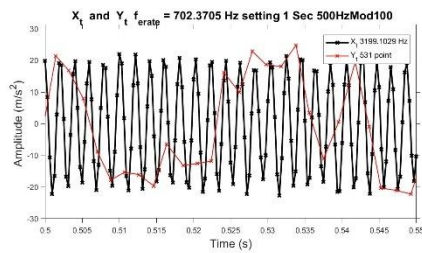
รูปที่ ๗.๗๖ การปรับตั้งแบบที่ 6 ที่สัญญาณความถี่ 1500 เฮิรตซ์



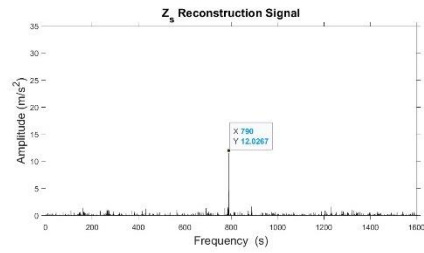
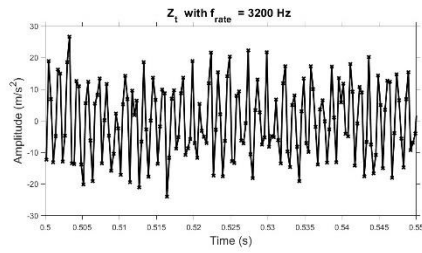
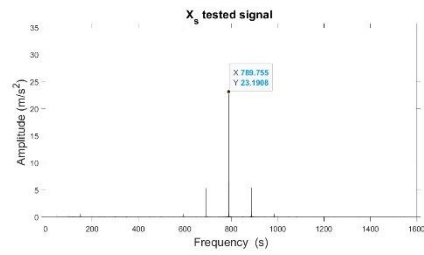
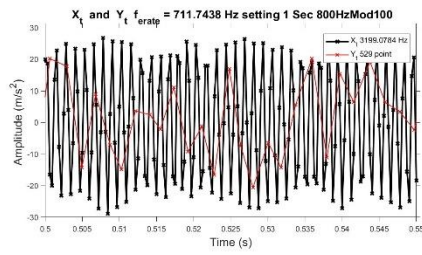
รูปที่ ๗.๗๗ การปรับตั้งแบบที่ 6 ที่สัญญาณความถี่ 1800 เฮิรตซ์



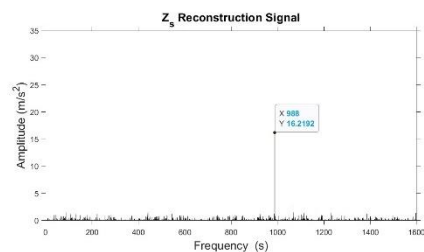
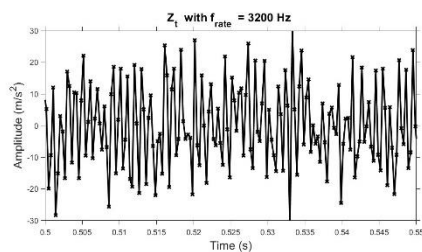
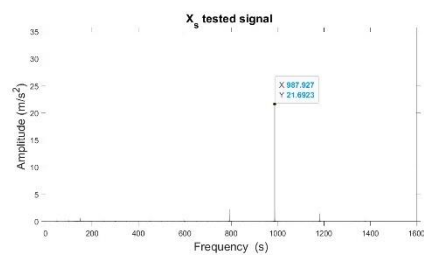
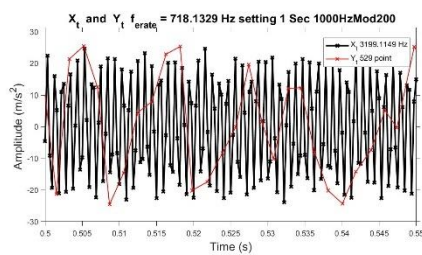
รูปที่ ๗.๗๘ การปรับตั้งแบบที่ 6 ที่สัญญาณความถี่ 300 เฮิรตซ์ มอดูเลชั่น



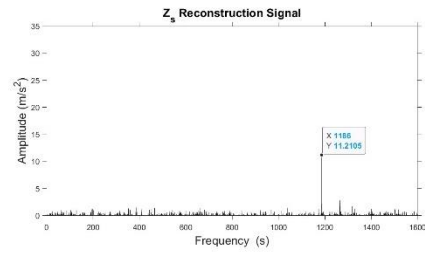
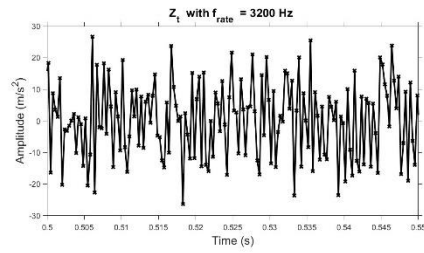
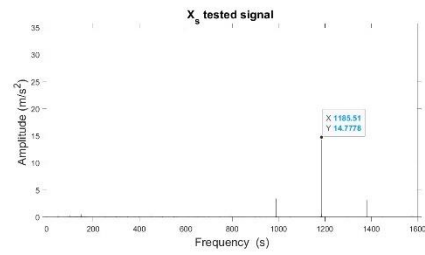
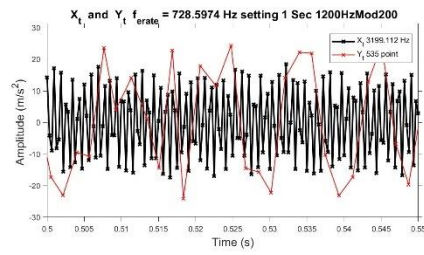
รูปที่ ๗.๗๙ การปรับตั้งแบบที่ 6 ที่สัญญาณความถี่ 500 เฮิรตซ์ มอดูเลชั่น



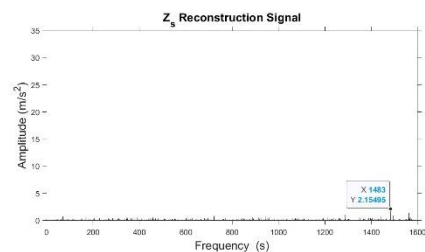
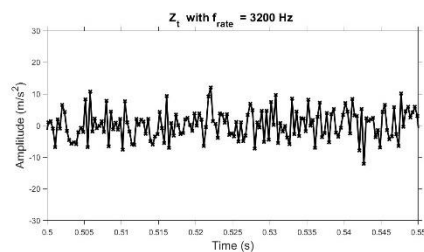
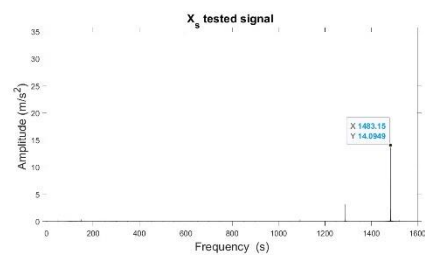
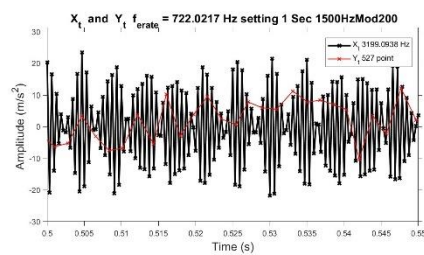
รูปที่ ๘๐ การปรับตั้งแบบที่ 6 ที่สัญญาณความถี่ 800 เฮิรตซ์ มอดูเลชั่น



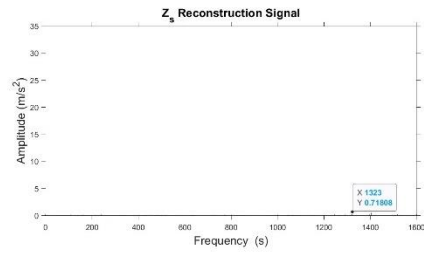
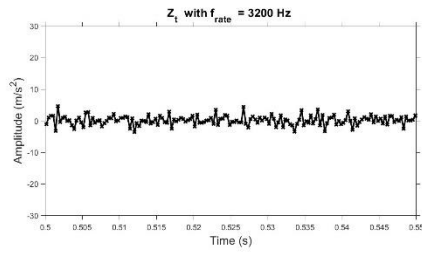
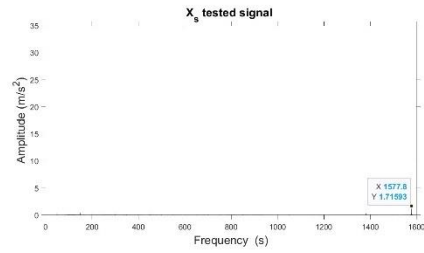
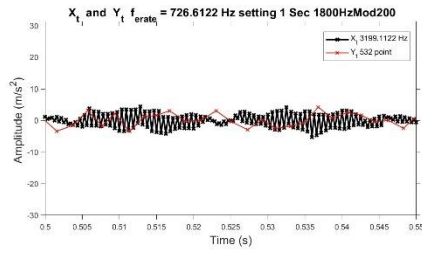
รูปที่ ๘๑ การปรับตั้งแบบที่ 6 ที่สัญญาณความถี่ 1000 เฮิรตซ์ มอดูเลชั่น



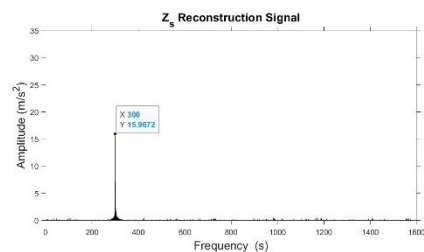
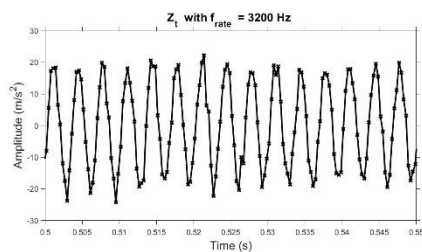
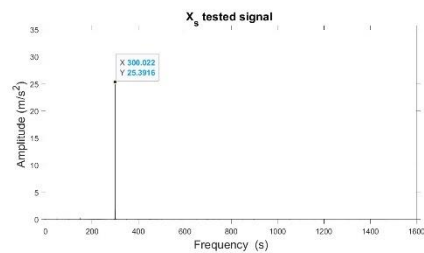
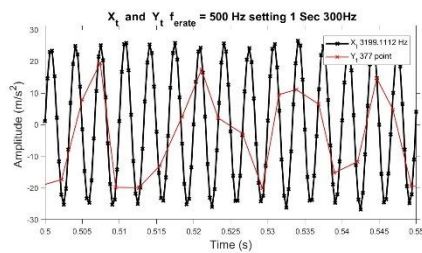
รูปที่ ๘.๘๒ การปรับตั้งแบบที่ 6 ที่สัญญาณความถี่ 1200 เฮิรตซ์ มอดูเลชั่น



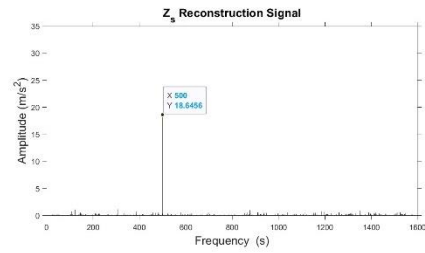
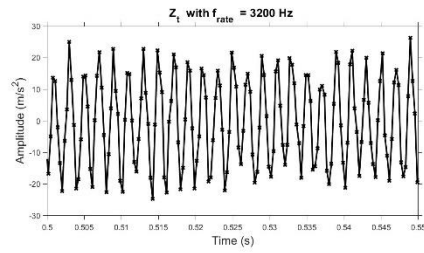
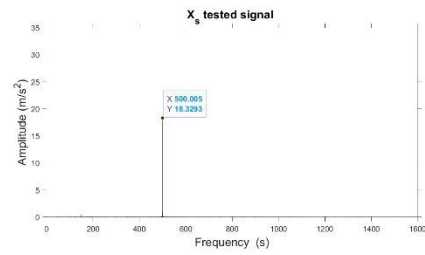
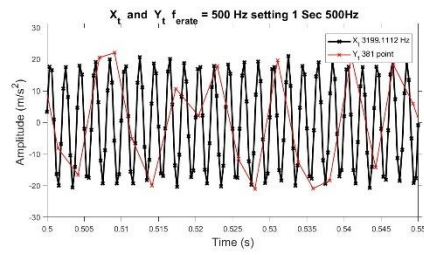
รูปที่ ๘.๘๓ การปรับตั้งแบบที่ 6 ที่สัญญาณความถี่ 1500 เฮิรตซ์ มอดูเลชั่น



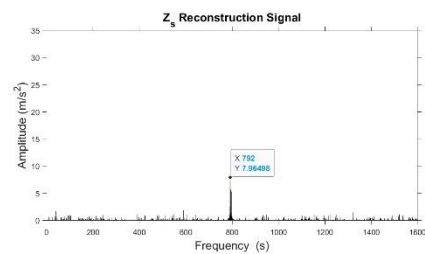
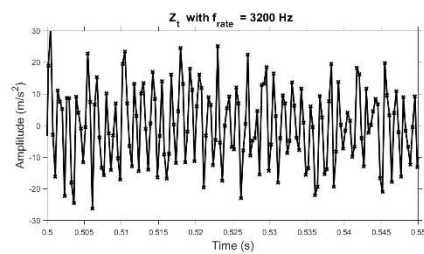
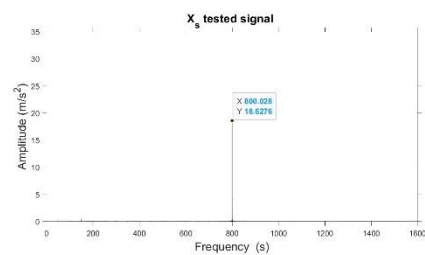
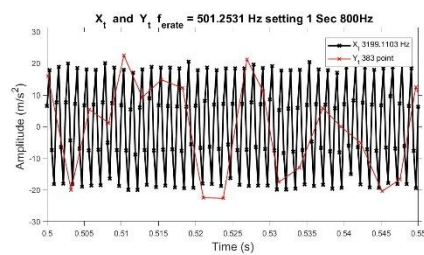
รูปที่ ๘.๘๔ การปรับตั้งแบบที่ 6 ที่สัญญาณความถี่ 1800 เฮิรตซ์ มอดูเลชั่น



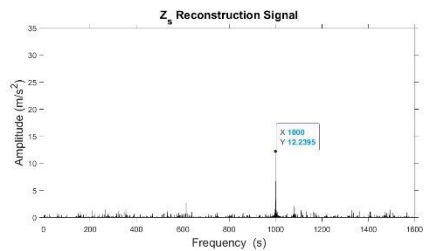
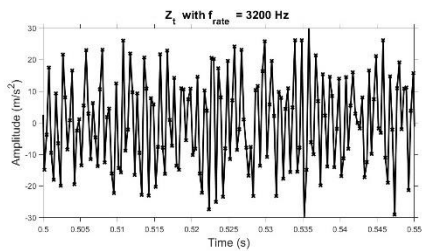
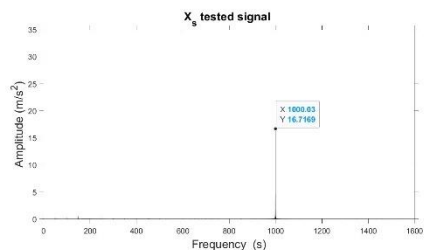
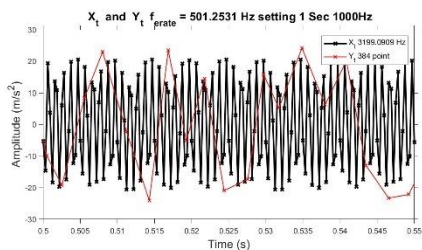
รูปที่ ๘.๘๕ การปรับตั้งแบบที่ 7 ที่สัญญาณความถี่ 300 เฮิรตซ์



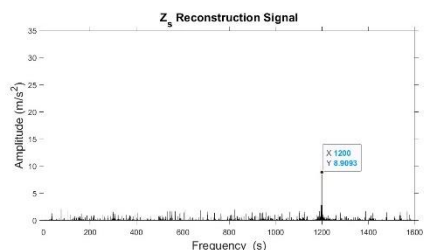
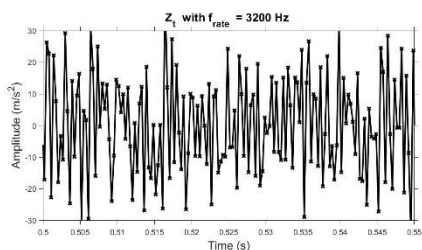
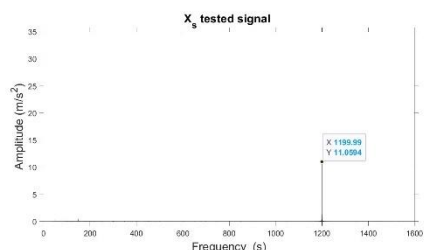
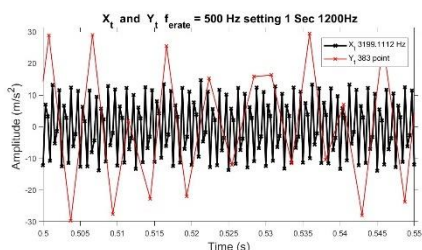
รูปที่ ๘.๘๖ การปรับตั้งแบบที่ 7 ที่สัญญาณความถี่ 500 เฮิรตซ์



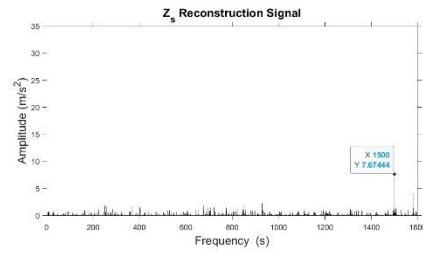
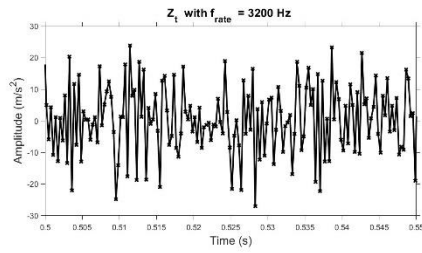
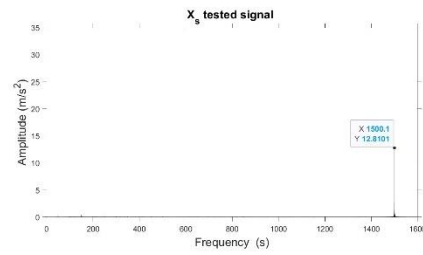
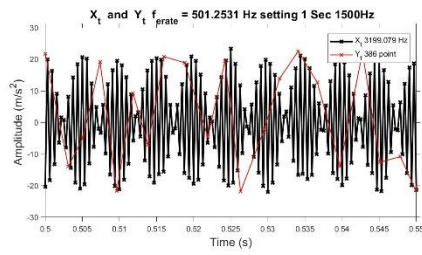
รูปที่ ๘.๘๗ การปรับตั้งแบบที่ 7 ที่สัญญาณความถี่ 800 เฮิรตซ์



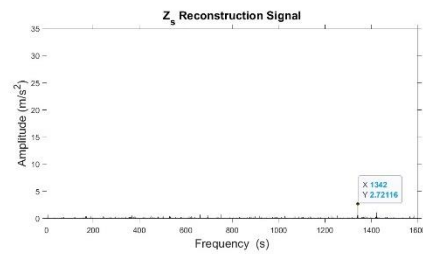
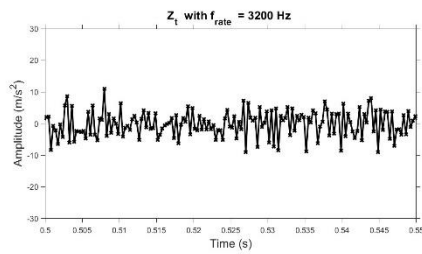
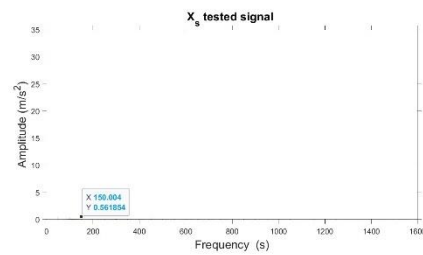
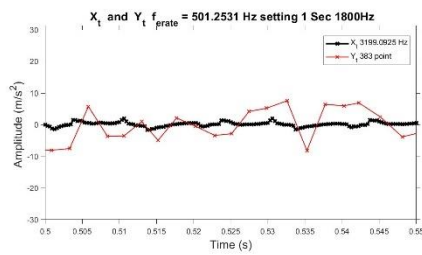
รูปที่ ๘๘ การปรับตั้งแบบที่ 7 ที่สัญญาณความถี่ 1000 เฮิรตซ์



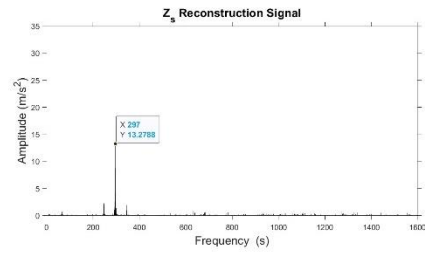
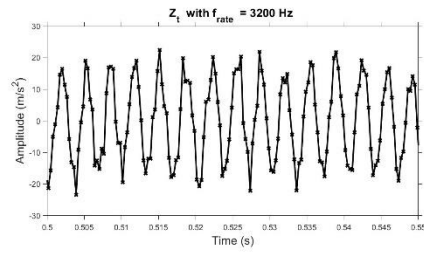
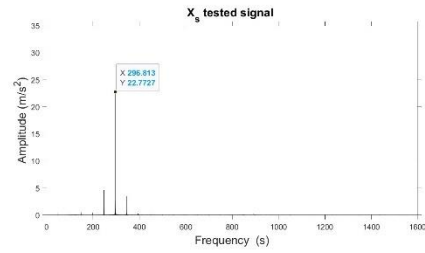
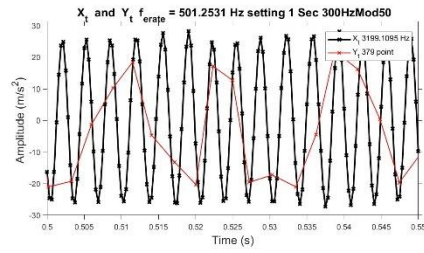
รูปที่ ๘๙ การปรับตั้งแบบที่ 7 ที่สัญญาณความถี่ 1200 เฮิรตซ์



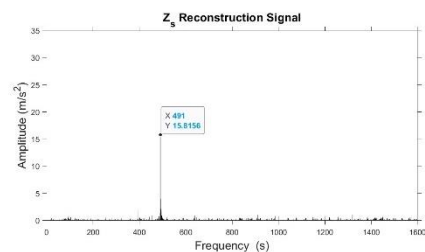
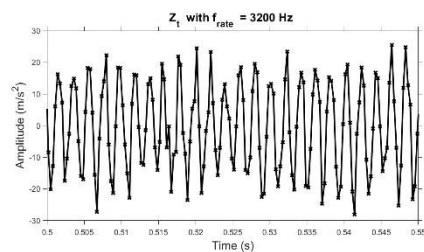
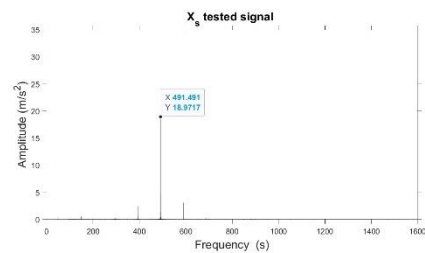
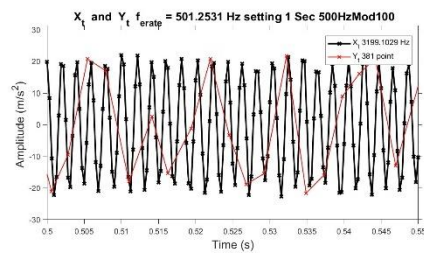
รูปที่ ๙๐ การปรับตั้งแบบที่ 7 ที่สัญญาณความถี่ 1500 เฮิรตซ์



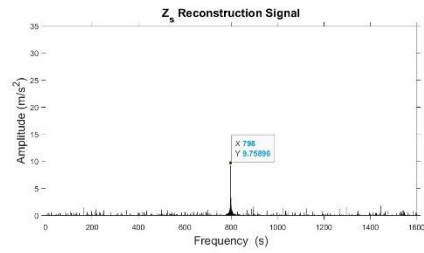
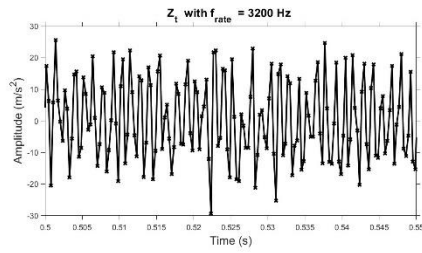
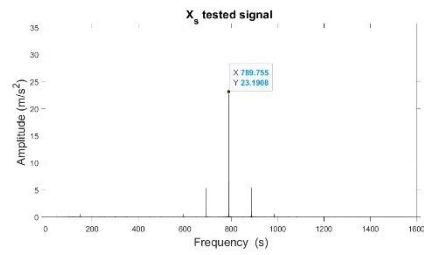
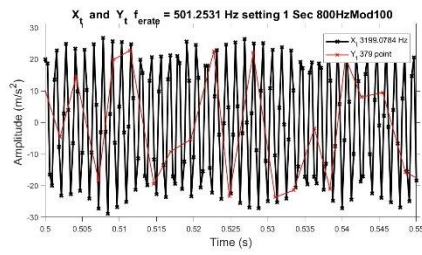
รูปที่ ๙๑ การปรับตั้งแบบที่ 7 ที่สัญญาณความถี่ 1800 เฮิรตซ์



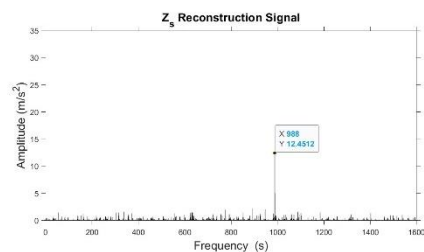
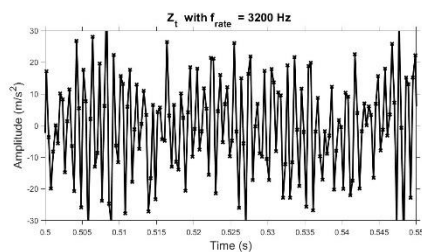
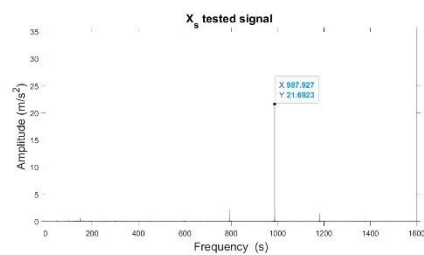
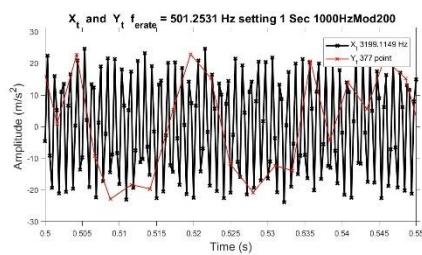
รูปที่ ๓.๙๒ การปรับตั้งแบบที่ 7 ที่สัญญาณความถี่ 300 เฮิรตซ์ มอดูเลชั่น



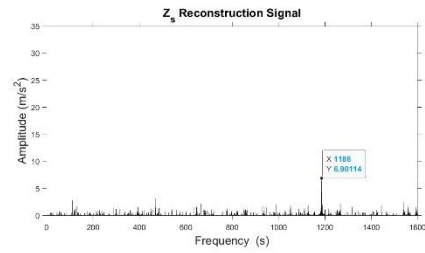
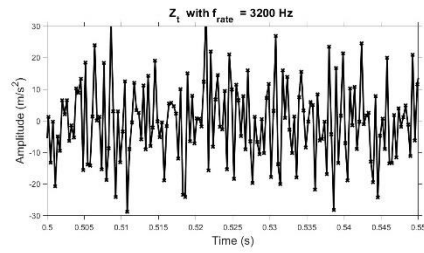
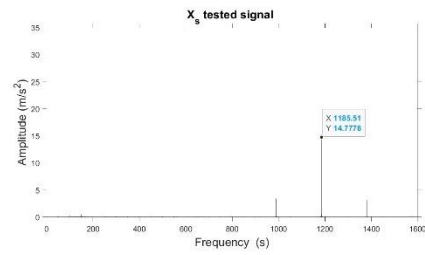
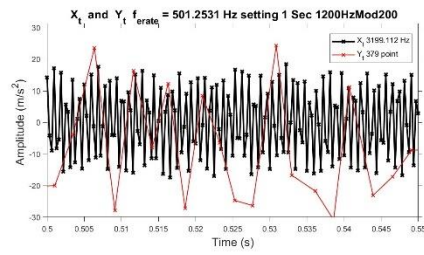
รูปที่ ๓.๙๓ การปรับตั้งแบบที่ 7 ที่สัญญาณความถี่ 500 เฮิรตซ์ มอดูเลชั่น



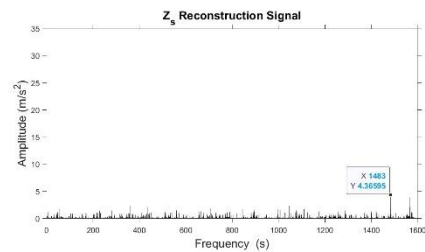
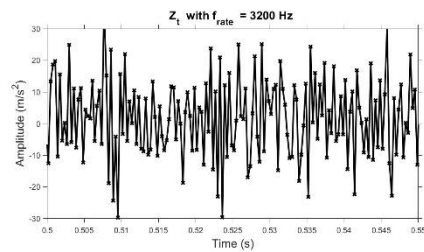
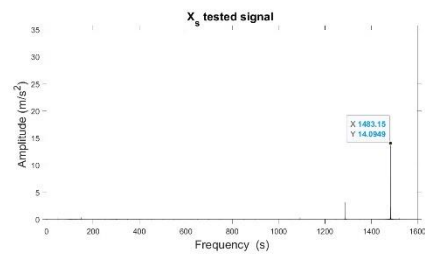
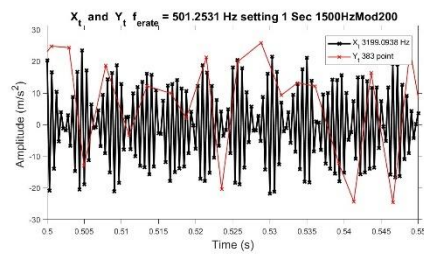
รูปที่ ๙๘ การปรับตั้งแบบที่ 7 ที่สัญญาณความถี่ 800 เฮิรตซ์ มอดูเลชั่น



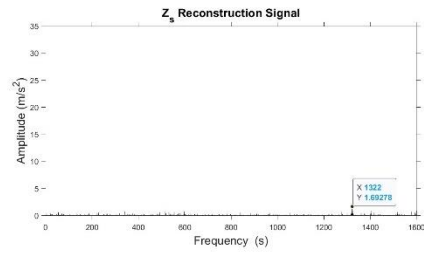
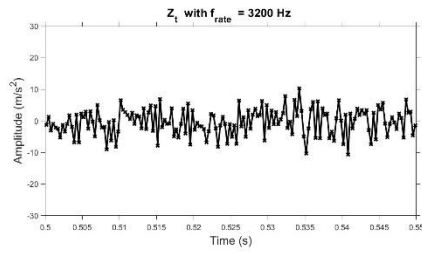
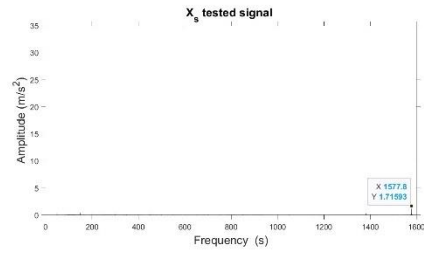
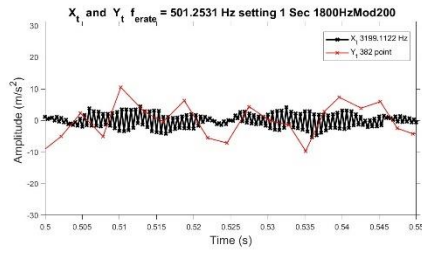
รูปที่ ๙๙ การปรับตั้งแบบที่ 7 ที่สัญญาณความถี่ 1000 เฮิรตซ์ มอดูเลชั่น



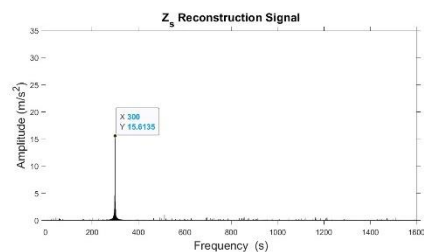
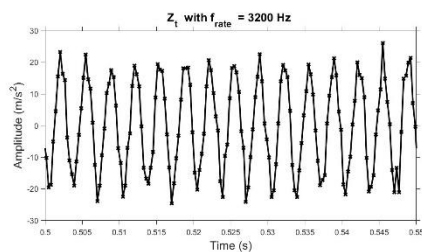
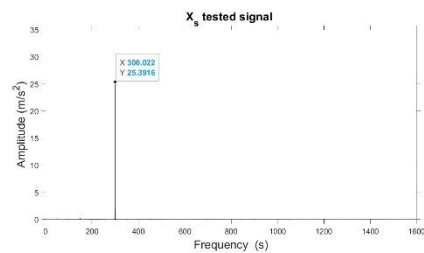
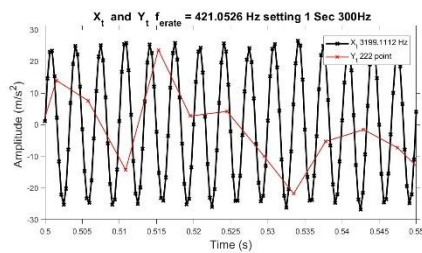
รูปที่ ๓.๙๖ การปรับตั้งแบบที่ 7 ที่สัญญาณความถี่ 1200 เฮิรตซ์ มอดูเลชั่น



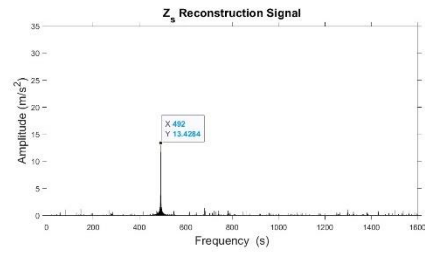
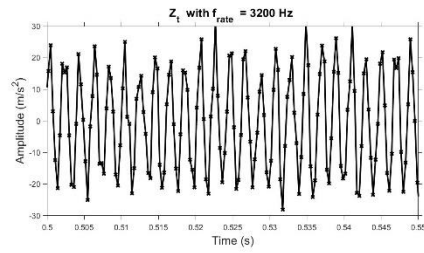
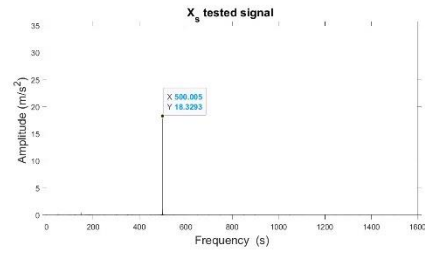
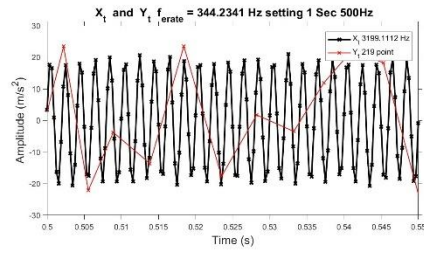
รูปที่ ๓.๙๗ การปรับตั้งแบบที่ 7 ที่สัญญาณความถี่ 1500 เฮิรตซ์ มอดูเลชั่น



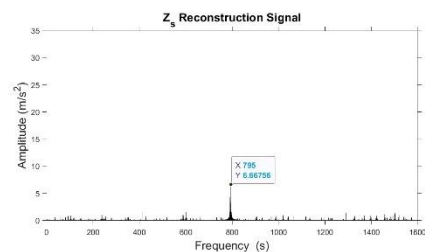
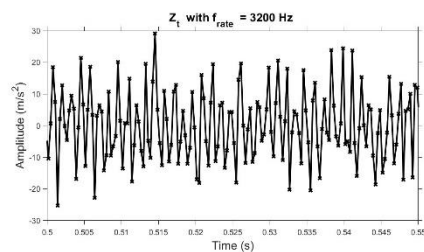
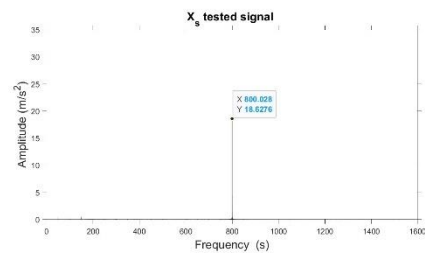
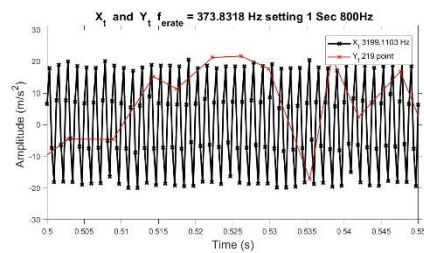
รูปที่ ๘.๙๘ การปรับตั้งแบบที่ 7 ที่สัญญาณความถี่ 1800 เฮิรตซ์ มอดูเลชั่น



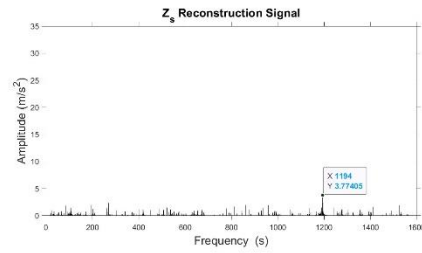
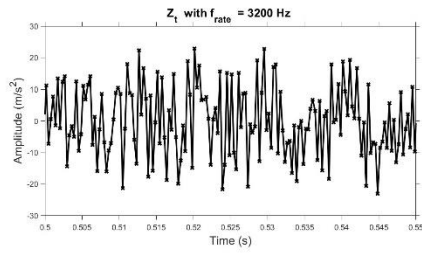
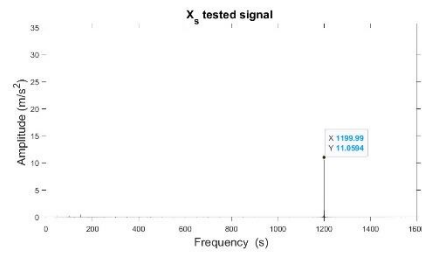
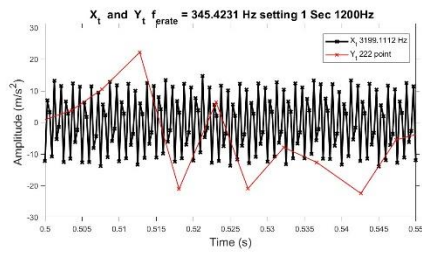
รูปที่ ๘.๙๙ การปรับตั้งแบบที่ 8 ที่สัญญาณความถี่ 300 เฮิรตซ์



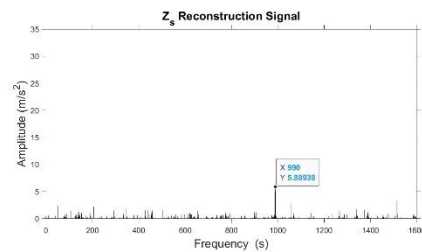
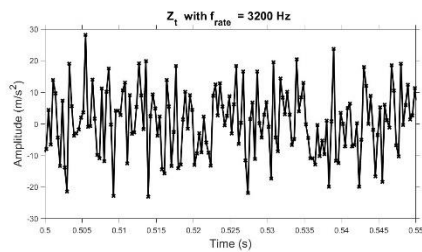
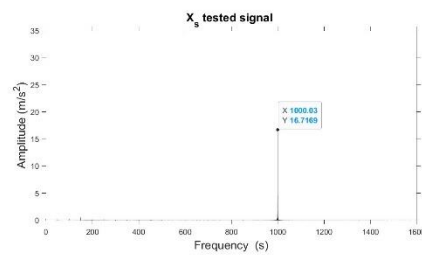
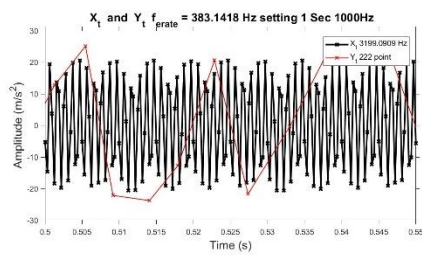
รูปที่ ๓.๑๐๐ การปรับตั้งแบบที่ ๘ ที่สัญญาณความถี่ 500 เฮิรตซ์



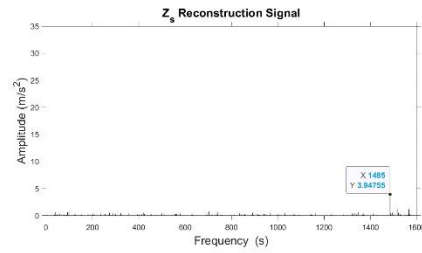
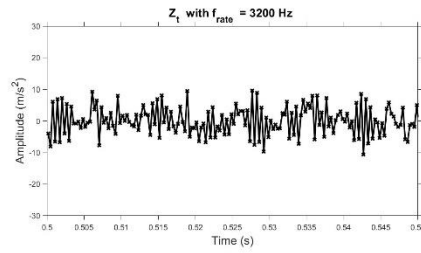
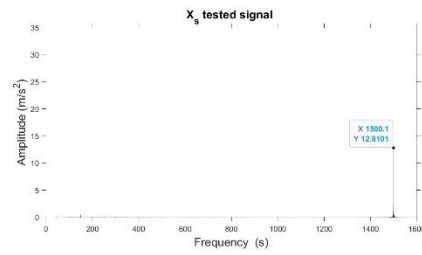
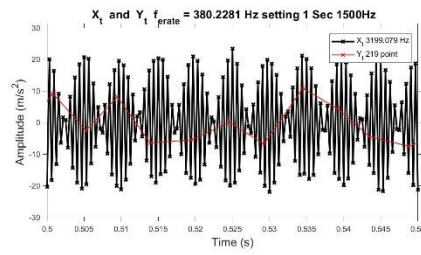
รูปที่ ๓.๑๐๑ การปรับตั้งแบบที่ ๘ ที่สัญญาณความถี่ 800 เฮิรตซ์



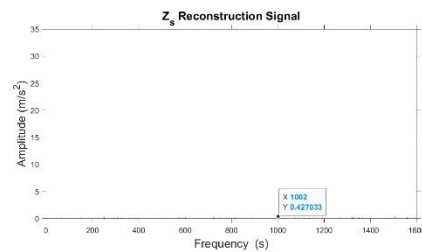
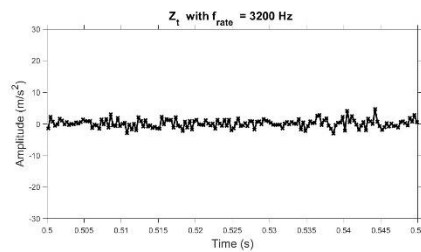
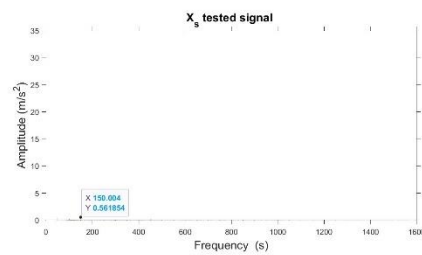
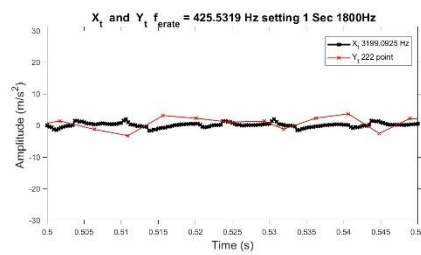
รูปที่ ๑๑.102 การปรับตั้งแบบที่ 8 ที่สัญญาณความถี่ 1000 เฮิรตซ์



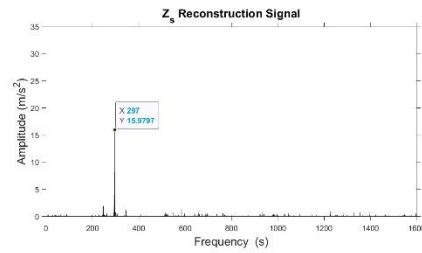
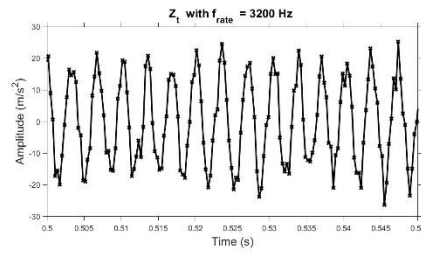
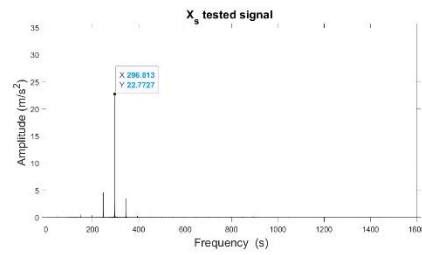
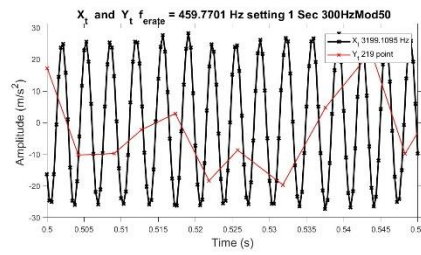
รูปที่ ๑๑.103 การปรับตั้งแบบที่ 8 ที่สัญญาณความถี่ 1200 เฮิรตซ์



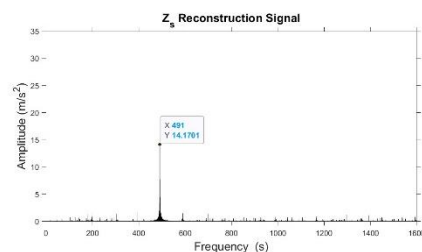
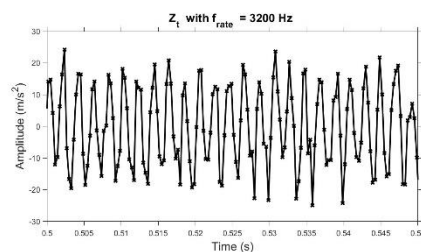
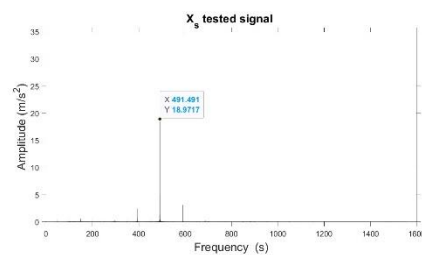
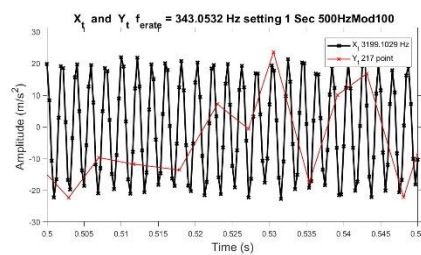
รูปที่ ๑๓.104 การปรับตั้งแบบที่ 8 ที่สัญญาณความถี่ 1500 เฮิรตซ์



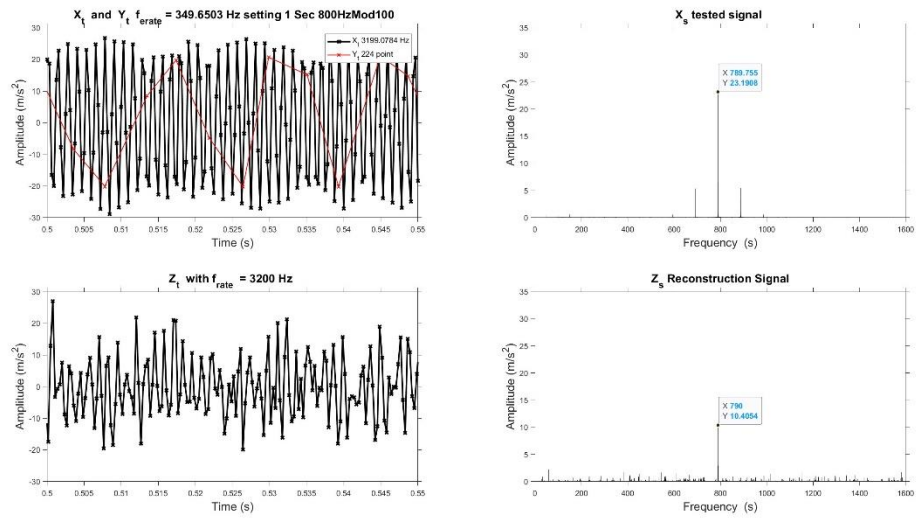
รูปที่ ๑๓.105 การปรับตั้งแบบที่ 8 ที่สัญญาณความถี่ 1800 เฮิรตซ์



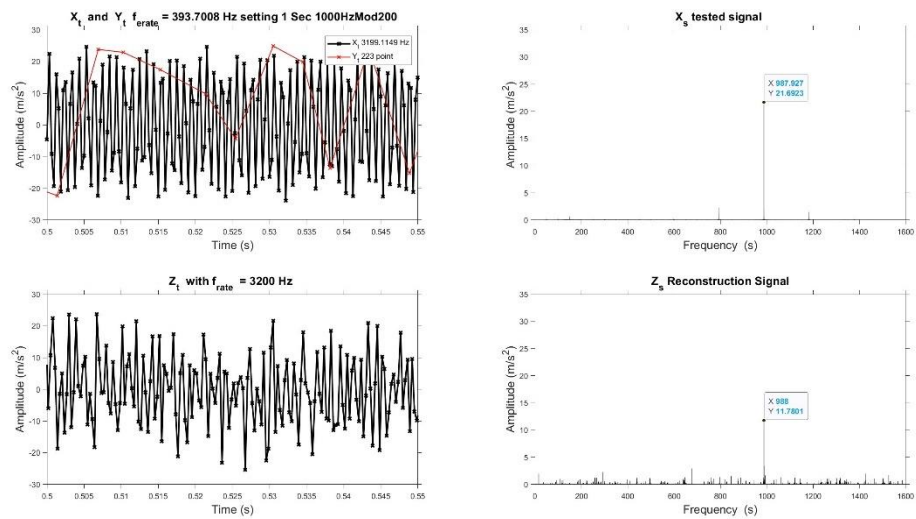
รูปที่ ๑๑.106 การปรับตั้งแบบที่ 8 ที่สัญญาณความถี่ 300 เฮิรตซ์ มอดูเลชั่น



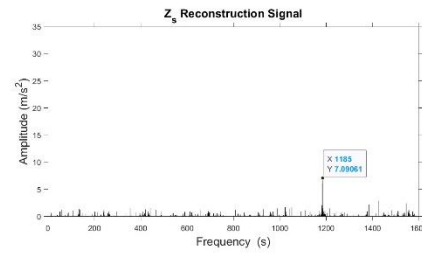
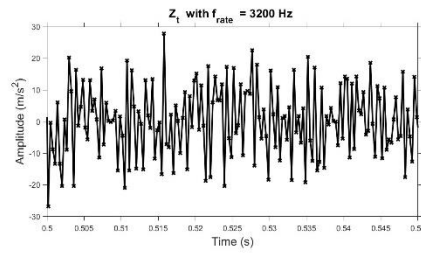
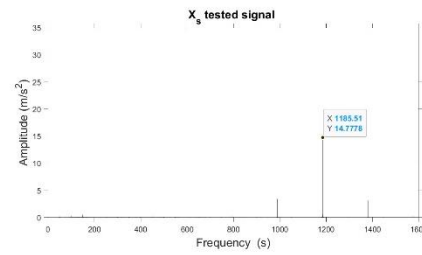
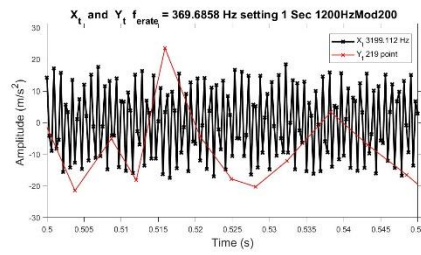
รูปที่ ๑๑.107 การปรับตั้งแบบที่ 8 ที่สัญญาณความถี่ 500 เฮิรตซ์ มอดูเลชั่น



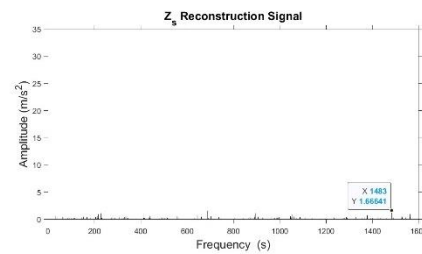
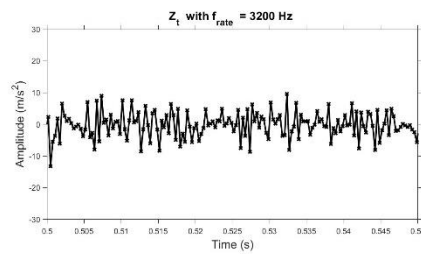
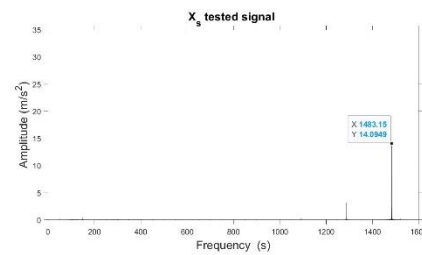
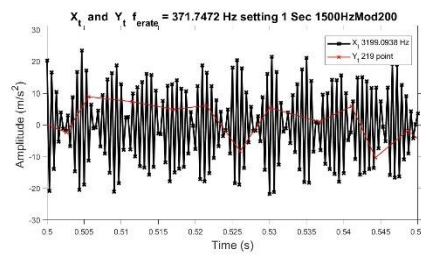
รูปที่ ๑๑.108 การปรับตั้งแบบที่ 8 ที่สัญญาณความถี่ 800 เฮิรตซ์ มอดูเลชัน



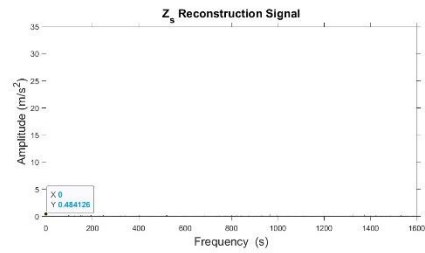
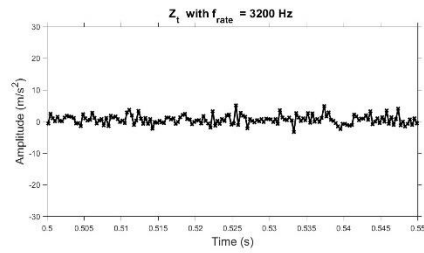
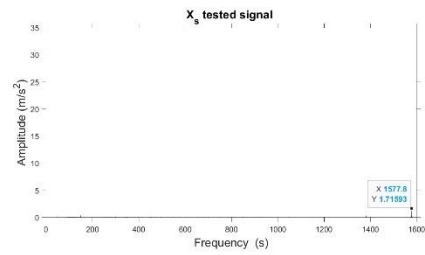
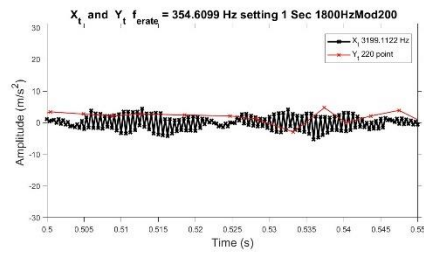
รูปที่ ๑๑.109 การปรับตั้งแบบที่ 8 ที่สัญญาณความถี่ 1000 เฮิรตซ์ มอดูเลชัน



รูปที่ ๓.๑๑๐ การปรับตั้งแบบที่ 8 ที่สัญญาณความถี่ 1200 เฮิรตซ์ มอดูเลชั่น



รูปที่ ๓.๑๑๑ การปรับตั้งแบบที่ 8 ที่สัญญาณความถี่ 1500 เฮิรตซ์ มอดูเลชั่น



รูปที่ ๓.๑๑๒ การปรับตั้งแบบที่ ๘ ที่สัญญาณความถี่ 1800 เฮิรตซ์ มอดูเลชั่น



ประวัติผู้เขียน

ชื่อ-สกุล	ศิริวิชญ์ มโนสิทธิกุล
วัน เดือน ปี เกิด	23 เม.ย. 2535
สถานที่เกิด	จังหวัดสงขลา
วุฒิการศึกษา	วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมเครื่องกล
ที่อยู่ปัจจุบัน	The Trust Residence Ratchada-Rama3 20/1182 ถ.นนทรี แขวงช่องนนทรี เขตยานนาวา กทม. 10120 0824382082



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY