

ตัวกรองปรับตัวได้แบบไอไออาร์

ADAPTIVE IIR FILTER



โดย
นายพันธุ์สิน บัวทอง
นายภัทรพงศ์ จันทรัตน์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เลขหมู่.....

เลขทะเบียน 50096

วัน,เดือน,ปี 21 เม.ย. 2547

b.....

สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวกรองปรับตัวได้แบบไอไออาร์

ADAPTIVE IIR FILTER

โดย

นายพันธุ์สิน บัวทอง 42010226

นายภัทรพงศ์ จันทรัตน์ 42010252

อาจารย์ที่ปรึกษา

อ. อัครพล ตีร์รัตน์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2545

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **ตัวกรองปรับตัวได้แบบไออาร์**

ADAPTIVE IIR FILTER

ผู้จัดทำ

1. นายพันธุ์สิน บัวทอง 42010226

2. นายภัทรพงศ์ จันทรรัตน์ 42010252

อิกพล ตรีรัตน์
.....อาจารย์ที่ปรึกษา
(อาจารย์ อัครพล ตรีรัตน์)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวกรองปรับตัวได้แบบไอไออาร์

ADAPTIVE IIR FILTER

โดย นายพันธุ์สิน บัวทอง 42010226

นายภัทรพงศ์ จันทรรัตน์ 42010252

อาจารย์ที่ปรึกษา อ. อัครพล ตีร์รัตน์

บทคัดย่อ

โครงการนี้นำเสนอการศึกษาและออกแบบตัวกรองปรับตัวได้แบบไอไออาร์ (Adaptive IIR Filter) ซึ่งมีข้อดีกว่าตัวกรองธรรมดาทั่วไป คือสามารถปรับสัมประสิทธิ์ตัวเองให้เหมาะสมต่อสภาพแวดล้อมการใช้งาน ทำให้สามารถนำไปประยุกต์ใช้งานได้หลายรูปแบบ ซึ่งโครงการนี้จะทำการออกแบบโดยใช้วิธีการแบบเอาต์พุท เออเรอร์และใช้อัลกอริทึมแบบแอลเอ็มเอส (Output Error Method – LMS Algorithm) โดยใช้วิธีการทาง สโตแคสติก เกรเดียนท์ (Stochastic Gradient) เพื่อหาอัลกอริทึมและทำการซิมูเลชัน (Simulation) ด้วย โปรแกรมเมทแลบ (MATLAB) ตลอดจนนำไปทดสอบในระบบประมวลผลสัญญาณดิจิทัล

ABSTRACT

This project presents a study and design of adaptive IIR filter which has the advantage over the ordinary digital filter due to its adaptation characteristics, it is possible to use in many applications. In this paper, the output error - LMS algorithm will be introduced and derived by using Stochastic Gradient method. The derived algorithm will be simulated by using MATLAB and testing in real world digital signal processing system.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทที่ 1 บทนำ	1
1.1 ความเป็นมาของหัวข้อปริญญานิพนธ์	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์	1
1.3 ขอบเขตของปริญญานิพนธ์	1
1.4 เนื้อหาของปริญญานิพนธ์	2
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 ทฤษฎีและหลักการพื้นฐานของระบบประมวลผลสัญญาณดิจิทัล	3
2.1.1 สัญญาณเชิงเวลาไม่ต่อเนื่อง	3
2.1.2 ชนิดของสัญญาณเชิงเวลาไม่ต่อเนื่อง	4
2.1.3 การกระทำพื้นฐานของลำดับ	5
2.1.4 ลำดับพื้นฐานในการประมวลผลสัญญาณดิจิทัล	6
2.1.5 ระบบเชิงเวลาไม่ต่อเนื่อง	8
2.1.6 ผลตอบสนองอิมพัลส์และผลตอบสนองต่อสเตป	9
2.1.7 คุณลักษณะของระบบเชิงเส้นไม่เปลี่ยนแปลงตามเวลา	9
2.1.8 การต่อเชื่อมพื้นฐานเพื่อสร้างระบบเชิงเส้นไม่เปลี่ยนแปลงตามเวลา	10
2.1.9 การแบ่งชนิดของระบบเชิงเส้นไม่เปลี่ยนแปลงตามเวลา	10
2.1.10 ความสัมพันธ์ของสัญญาณ	11
2.1.11 คุณสมบัติของลำดับครอสคอร์รีเลชันและลำดับอโตคอร์รีเลชัน	12
2.1.12 รูปแบบนอร์มอลไลซ์ของของคอร์รีเลชัน	13
2.2 ทฤษฎีตัวกรองดิจิทัล	14
2.2.1 องค์ประกอบของตัวกรองดิจิทัล	14
2.2.2 โครงสร้างของตัวกรองดิจิทัลแบบไอโออาร์	14
2.3 ตัวกรองปรับตัวได้	20
2.3.1 แนวคิดของตัวกรองปรับตัวได้	20
2.3.2 กรรมวิธีการพัฒนาอัลกอริทึมของตัวกรองปรับตัวได้แบบต่างๆ	21
2.3.3 การวัดประสิทธิภาพในระบบปรับตัว	21
2.3.4 การประยุกต์ใช้งานตัวกรองปรับตัวได้	22
2.4 ตัวกรองปรับตัวได้แบบเอฟไออาร์	26
2.5 ตัวกรองปรับตัวได้แบบไอโออาร์	28
2.4.1 วิธีการแบบเอาท์พุทเออร์เรอร์	28
2.4.2 วิธีการแบบอ็ควชันเออร์เรอร์	32

บทที่ 3 การคำนวณและการสร้าง	35
3.1 การออกแบบและเลือกใช้อัลกอริทึม	35
3.1.1 การสร้างตัวกรองปรับตัวได้แบบเอฟไออาร์ลำดับที่ 2	35
3.1.2 การสร้างตัวกรองปรับตัวได้แบบไอไออาร์ลำดับที่ 2	37
3.1.3 การจัดตัวกรองเพื่อใช้ในการกำจัดสัญญาณรบกวน	43
3.1.4 การจัดตัวกรองเพื่อใช้ในการหาคุณลักษณะของระบบอื่น	44
3.2 การหาค่าเฉลี่ยด้วยโปรแกรมแมทแล็บ	46
3.2.1 การเขียนโปรแกรมหาค่าเฉลี่ยสำหรับตัวกรองปรับตัวแบบเอฟไออาร์	46
3.2.2 การเขียนโปรแกรมหาค่าเฉลี่ยสำหรับตัวกรองปรับตัวแบบไอไออาร์	47
3.2.3 การเขียนโปรแกรมหาค่าเฉลี่ยสำหรับระบบกำจัดสัญญาณรบกวน	48
3.2.4 การเขียนโปรแกรมหาค่าเฉลี่ยสำหรับการหาคุณลักษณะระบบอื่นโดยใช้ตัวกรอง	49
3.3 การสร้างตัวกรองปรับตัวในระบบประมวลผลสัญญาณดิจิทัล	50
3.3.1 การสร้างตัวกรองปรับตัวแบบเอฟไออาร์ในระบบประมวลผลสัญญาณดิจิทัล	52
3.3.2 การสร้างตัวกรองปรับตัวแบบไอไออาร์ในระบบประมวลผลสัญญาณดิจิทัล	53
3.3.3 การสร้างตัวกำจัดสัญญาณรบกวนในระบบประมวลผลสัญญาณดิจิทัล	54
บทที่ 4 การทดลองและผลการทดลอง	55
4.1 การวัดผลตอบสนอง โดยการหาค่าเฉลี่ยของระบบต่างๆ	55
4.1.1 การหาค่าเฉลี่ยของตัวกรองปรับตัวแบบเอฟไออาร์	55
4.1.1.1 หาผลตอบสนองต่อความถี่ต่างๆของตัวกรองปรับตัวแบบเอฟไออาร์	55
4.1.1.2 หาผลตอบสนองต่อขนาดสเกลต่างๆของตัวกรองปรับตัวแบบเอฟไออาร์	55
4.1.1.3 หาผลตอบสนองต่ออันดับต่างๆของตัวกรองปรับตัวแบบเอฟไออาร์	55
4.1.2 การหาค่าเฉลี่ยของตัวกรองปรับตัวแบบไอไออาร์	55
4.1.2.1 หาผลตอบสนองต่อความถี่ต่างๆของตัวกรองปรับตัวแบบไอไออาร์	55
4.1.2.2 หาผลตอบสนองต่อขนาดสเกลต่างๆของตัวกรองปรับตัวแบบไอไออาร์	55
4.1.2.3 หาผลตอบสนองต่ออันดับต่างๆของตัวกรองปรับตัวแบบไอไออาร์	55
4.1.3 การหาค่าเฉลี่ยของตัวกำจัดสัญญาณรบกวน	56
4.1.3.1 หาผลตอบสนองต่อความถี่ต่างๆของตัวกำจัดสัญญาณรบกวน	56
4.1.3.2 หาผลตอบสนองต่อขนาดสเกลต่างๆของตัวกำจัดสัญญาณรบกวน	56
4.1.3.3 ผลตอบสนองต่ออันดับต่างๆของตัวกำจัดสัญญาณรบกวน	56
4.1.4 การหาค่าเฉลี่ยของการใช้ตัวกรองปรับตัวหาคุณสมบัติของระบบที่ไม่ทราบ	56
4.2 ผลตอบสนองของต่างๆจากการหาค่าเฉลี่ย	57
4.2.1 ผลตอบสนองต่างๆของตัวกรองปรับตัวแบบเอฟไออาร์	57
4.2.1.1 ผลตอบสนองต่อความถี่ต่างๆของตัวกรองปรับตัวแบบเอฟไออาร์	57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์อื่นใด

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1.2 ผลตอบสนองต่อขนาดสเตปต่างๆของตัวกรองปรับตัวแบบเอฟไออาร์	62
4.2.1.3 ผลตอบสนองต่ออันดับต่างๆของตัวกรองปรับตัวแบบเอฟไออาร์	66
4.2.2 ผลตอบสนองต่างๆของตัวกรองปรับตัวแบบไอไออาร์	70
4.2.2.1 ผลตอบสนองต่อความถี่ต่างๆ ของตัวกรองปรับตัวแบบไอไออาร์	70
4.2.2.2 ผลตอบสนองต่อขนาดสเตปต่างๆของตัวกรองปรับตัวแบบไอไออาร์	75
4.2.2.3 ผลตอบสนองต่ออันดับต่างๆของตัวกรองปรับตัวแบบไอไออาร์	80
4.2.3 ผลตอบสนองต่างๆ ของตัวกำจัดสัญญาณรบกวน	84
4.2.3.1 ผลตอบสนองต่อความถี่ต่างๆ ของตัวกำจัดสัญญาณรบกวน	84
4.2.3.2 ผลตอบสนองต่อสัญญาณรบกวนรูปไซน์ที่ขนาดแอมพลิจูดต่างๆ ของตัวกำจัดสัญญาณรบกวน	90
4.2.3.3 ผลตอบสนองต่อสัญญาณรบกวนสุ่มที่ขนาดแอมพลิจูดต่างๆ ของตัวกำจัดสัญญาณรบกวน	93
4.2.4 ผลตอบสนองต่างๆ ของการใช้ตัวกรองปรับตัวหาคุณสมบัติของระบบ	96
4.2.4.1 ผลตอบสนองต่อขนาดสเตปต่างๆ ของการใช้ตัวกรองปรับตัวหาคุณสมบัติของระบบที่ไม่ทราบ	96
4.3 การวัดคุณสมบัติโดยการสร้างในระบบประมวลผลสัญญาณ	100
4.3.1 การวัดคุณสมบัติตัวกรองปรับตัวแบบเอฟไออาร์ที่สร้างขึ้น	100
4.3.2 การวัดคุณสมบัติตัวกรองปรับตัวแบบไอไออาร์ที่สร้างขึ้น	100
4.3.3 การวัดคุณสมบัติตัวกำจัดสัญญาณรบกวนที่สร้างขึ้น	101
4.4 ผลตอบสนองของตัวกรองปรับตัวจากการสร้างในระบบประมวลผลสัญญาณ	102
4.4.1 ผลตอบสนองของตัวกรองปรับตัวแบบเอฟไออาร์	102
4.4.2 ผลตอบสนองของตัวกรองปรับตัวแบบไอไออาร์	106
4.4.3 ผลตอบสนองของตัวกำจัดสัญญาณรบกวน	132
บทที่ 5 บทวิจารณ์และสรุป	136
ภาคผนวก	
กิตติกรรมประกาศ	
เอกสารอ้างอิง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ 2.1 แสดงระบบประมวลผลสัญญาณดิจิทัล	3
รูปที่ 2.2 แสดงสัญญาณ $x[n]$	4
รูปที่ 2.3 แสดงการกระทำพื้นฐานของลำดับ	6
รูปที่ 2.4ก แสดงยูนิตอิมพัลส์ $\delta[n]$	6
รูปที่ 2.4ข แสดงยูนิตอิมพัลส์ $\delta[n - 2]$	6
รูปที่ 2.5 แสดงสัญญาณ $y[n]$ ซึ่งเป็นผลรวมของยูนิตอิมพัลส์ที่เลื่อนเวลา	7
รูปที่ 2.6ก แสดงยูนิตสเตป $\delta[n]$	7
รูปที่ 2.6ข แสดงยูนิตสเตป $\delta[n - 2]$	7
รูปที่ 2.7 ไดอะแกรมการทำงานของตัวกรองดิจิทัลพื้นฐาน	14
รูปที่ 2.8 แสดงองค์ประกอบพื้นฐานทั้งสามที่ใช้เป็นส่วนประกอบของตัวกรองดิจิทัล	14
รูปที่ 2.9ก เขียนแยกตัวบวกสัญญาณ	15
รูปที่ 2.9ข เขียนรวมตัวบวกสัญญาณ	15
รูปที่ 2.10ก เขียนแยกอุปกรณ์หน่วงสัญญาณ	17
รูปที่ 2.10ข เขียนรวมอุปกรณ์หน่วงสัญญาณ	17
รูปที่ 2.11 แสดงโครงสร้างแบบต่ออนุกรม	18
รูปที่ 2.12 แสดงโครงสร้างตัวกรองดิจิทัลแบบ ไอ ไออาร์ ที่มีโครงสร้างแบบค้อนขวาน	19
รูปที่ 2.13 แสดงตัวลดสัญญาณรบกวน	20
รูปที่ 2.14 แสดงประยุกต์ใช้ตัวกรองปรับตัวในการจำลองคุณลักษณะของระบบที่ไม่ทราบ	23
รูปที่ 2.15 แสดงการใช้อีควอไลเซอร์แบบปรับตัวได้ในระบบส่งสัญญาณ	24
รูปที่ 2.16 แสดงโครงสร้างระบบกำจัดสัญญาณรบกวนโดยใช้ตัวกรองปรับค่าได้	24
รูปที่ 2.17 แสดงอุปกรณ์เพิ่มประสิทธิภาพสาย	25
รูปที่ 2.18 ตัวกรองปรับตัวแบบ ไอ ไออาร์ โดยใช้วิธีการแบบเอาท์พุทเออร์เรอร์	29
รูปที่ 2.19 แสดงบล็อกไดอะแกรมการคำนวณค่า $\alpha_i(n)$ และ $\beta_i(n)$	31
รูปที่ 2.20 แสดงบล็อกไดอะแกรมการคำนวณค่า $\alpha_i(n)$ และ $\beta_i(n)$ โดยการประมาณแล้ว	32
รูปที่ 2.21 แสดงบล็อกไดอะแกรมของวิธีการอิกเวนชัน เออร์เรอร์	33
รูปที่ 3.1 แสดงกระบวนการทำงานของตัวกรองปรับตัวแบบเอฟไออาร์	37
รูปที่ 3.2 แสดงกระบวนการทำงานของตัวกรองปรับตัวแบบไอไออาร์	41
รูปที่ 3.3 แสดง โครงสร้างตัวกรองปรับตัวที่ออกแบบ	42
รูปที่ 3.4 แสดง โครงสร้างตัวกรองปรับตัวในส่วนการหาค่าเวกเตอร์ปรับสัมประสิทธิ์	42
รูปที่ 3.5 แสดง โครงสร้างระบบกำจัดสัญญาณรบกวนโดยใช้ตัวกรองปรับตัว	43
รูปที่ 3.6 แสดงกระบวนการทำงานของระบบกำจัดสัญญาณรบกวน	44
รูปที่ 3.7 การจัดตัวกรองเพื่อใช้ในการหาคุณลักษณะของระบบอื่น	44
รูปที่ 3.8 แสดงกระบวนการทำงานของการหาคุณลักษณะระบบอื่นโดยใช้ตัวกรอง	45

รูปที่ 3.9	แสดงการไหลของโปรแกรมซิมูเลชันสำหรับตัวกรองปรับตัวแบบเอฟไออาร์	46
รูปที่ 3.10	แสดงการไหลของโปรแกรมซิมูเลชันสำหรับตัวกรองปรับตัวแบบไอไออาร์	47
รูปที่ 3.11	แสดงการไหลของโปรแกรมซิมูเลชันสำหรับระบบกำจัดสัญญาณรบกวน	48
รูปที่ 3.12	แสดงการไหลของโปรแกรมซิมูเลชันเพื่อทำการหาคุณลักษณะระบบอื่น	49
รูปที่ 3.13	บอร์ดประมวลผลสัญญาณดิจิทัล	50
รูปที่ 3.14	การวัดผลตอบสนองของตัวกรองปรับตัวในการสร้างจริงในระบบประมวลผลสัญญาณ	51
รูปที่ 3.15	กระบวนการพัฒนาอัลกอริทึมสำหรับตัวประมวลผลสัญญาณดิจิทัล	51
รูปที่ 3.16	แสดงการไหลของโปรแกรมซิมูเลชันสำหรับตัวกรองปรับตัวแบบเอฟไออาร์	52
รูปที่ 3.17	แสดงการไหลของโปรแกรมซิมูเลชันสำหรับตัวกรองปรับตัวแบบไอไออาร์	53
รูปที่ 3.18	แสดงการไหลของโปรแกรมสำหรับตัวกำจัดสัญญาณรบกวน	54
รูปที่ 4.1	ผลตอบสนองต่างๆ ของตัวกรองปรับตัวแบบเอฟไออาร์อันดับ 50 ต่อ สัญญาณรบกวน รูปไซน์ความถี่ 500 Hz ความถี่สัญญาณเบสแบนด์ 80 Hz ขนาดสเตปเท่ากับ 0.005	57
รูปที่ 4.2	ผลตอบสนองต่างๆ ของตัวกรองปรับตัวแบบเอฟไออาร์อันดับ 50 ต่อ สัญญาณรบกวน รูปไซน์ความถี่ 2 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz ขนาดสเตปเท่ากับ 0.005	58
รูปที่ 4.3	ผลตอบสนองต่างๆ ของตัวกรองปรับตัวแบบเอฟไออาร์อันดับ 50 ต่อ สัญญาณรบกวน รูปไซน์ความถี่ 3 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz ขนาดสเตปเท่ากับ 0.005	59
รูปที่ 4.4	ผลตอบสนองต่างๆ ของตัวกรองปรับตัวแบบเอฟไออาร์อันดับ 50 ต่อ สัญญาณรบกวน ความถี่สัญญาณเบสแบนด์ 80 Hz ขนาดสเตปเท่ากับ 0.005	60
รูปที่ 4.5	ผลตอบสนองต่างๆ ของตัวกรองปรับตัวแบบเอฟไออาร์อันดับ 50 ต่อ สัญญาณรบกวน รูปไซน์ ความถี่ 2 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz ขนาดสเตปเท่ากับ 0.003	62
รูปที่ 4.6	ผลตอบสนองต่างๆ ของตัวกรองปรับตัวแบบเอฟไออาร์อันดับ 50 ต่อ สัญญาณรบกวน รูปไซน์ความถี่ 2 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz ขนาดสเตปเท่ากับ 0.005	63
รูปที่ 4.7	ผลตอบสนองต่างๆ ของตัวกรองปรับตัวแบบเอฟไออาร์อันดับ 50 ต่อ สัญญาณรบกวน รูปไซน์ความถี่ 2 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz ขนาดสเตปเท่ากับ 0.01	64
รูปที่ 4.8	ผลตอบสนองต่างๆ ของตัวกรองปรับตัวแบบเอฟไออาร์อันดับ 10 ต่อ สัญญาณรบกวน รูปไซน์ ความถี่ 2 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz ขนาดสเตปเท่ากับ 0.005	66
รูปที่ 4.9	ผลตอบสนองต่างๆ ของตัวกรองปรับตัวแบบเอฟไออาร์อันดับ 25 ต่อ สัญญาณรบกวน รูปไซน์ความถี่ 2 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz ขนาดสเตปเท่ากับ 0.005	67
รูปที่ 4.10	ผลตอบสนองต่างๆ ของตัวกรองปรับตัวแบบเอฟไออาร์อันดับ 50 ต่อ สัญญาณรบกวน รูปไซน์ความถี่ 2 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz ขนาดสเตปเท่ากับ 0.005	68
รูปที่ 4.11	ผลตอบสนองต่างๆ ของตัวกรองปรับตัวแบบไอไออาร์อันดับ 10 ต่อ สัญญาณรบกวน รูปไซน์ความถี่ 500 Hz ความถี่สัญญาณเบสแบนด์ 80 Hz ขนาดสเตปเท่ากับ 0.005	70

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ห้ามเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.28 ผลตอบสนองต่างๆ ของตัวกำจัดสัญญาณรบกวน ต่อ สัญญาณรบกวนรูปไซน์ ความถี่ 1 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz ขนาดสเตปเท่ากับ 0.03 ขนาด แอมพลิจูดสัญญาณเบสแบนด์เท่ากับ 1 ขนาดแอมพลิจูดสัญญาณรบกวน เท่ากับ 0.3	90
รูปที่ 4.29 ผลตอบสนองต่างๆ ของตัวกำจัดสัญญาณรบกวน ต่อ สัญญาณรบกวนรูปไซน์ ความถี่ 1 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz ขนาดสเตปเท่ากับ 0.03 ขนาด แอมพลิจูดสัญญาณเบสแบนด์เท่ากับ 1 ขนาดแอมพลิจูดสัญญาณรบกวน เท่ากับ 0.5	91
รูปที่ 4.30 ผลตอบสนองต่างๆ ของตัวกำจัดสัญญาณรบกวน ต่อ สัญญาณรบกวนรูปไซน์ ความถี่ 1 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz ขนาดสเตปเท่ากับ 0.03 ขนาด แอมพลิจูดสัญญาณเบสแบนด์เท่ากับ 1 ขนาดแอมพลิจูดสัญญาณรบกวน เท่ากับ 0.7	92
รูปที่ 4.31 ผลตอบสนองต่างๆ ของตัวกำจัดสัญญาณรบกวน ต่อ สัญญาณรบกวนสุ่มความถี่ สัญญาณเบสแบนด์ 80 Hz ขนาดสเตปเท่ากับ 0.005 ขนาดแอมพลิจูด สัญญาณเบสแบนด์เท่ากับ 1 ขนาด แอมพลิจูดสัญญาณรบกวน เท่ากับ 0.3	93
รูปที่ 4.32 ผลตอบสนองต่างๆ ของตัวกำจัดสัญญาณรบกวน ต่อ สัญญาณรบกวนสุ่มความถี่ สัญญาณ เบสแบนด์ 80 Hz ขนาดสเตปเท่ากับ 0.03 ขนาดแอมพลิจูดสัญญาณ เบสแบนด์เท่ากับ 1 ขนาดแอมพลิจูดสัญญาณรบกวน เท่ากับ 0.5	94
รูปที่ 4.33 ผลตอบสนองต่างๆ ของตัวกำจัดสัญญาณรบกวน ต่อ สัญญาณรบกวนสุ่มความถี่ สัญญาณเบสแบนด์ 80 Hz ขนาดสเตปเท่ากับ 0.03 ขนาดแอมพลิจูดสัญญาณ เบสแบนด์เท่ากับ 1 ขนาดแอมพลิจูดสัญญาณรบกวน เท่ากับ 0.7	95
รูปที่ 4.34 ผลตอบสนองต่างๆ ของการใช้ตัวกรองปรับตัวหาคุณสมบัติของระบบที่ไม่ทราบ โดยใช้ขนาดสเตป เท่ากับ 0.005	96
รูปที่ 4.35 ผลตอบสนองต่างๆ ของการใช้ตัวกรองปรับตัวหาคุณสมบัติของระบบที่ไม่ทราบ โดยใช้ขนาดสเตป เท่ากับ 0.01	97
รูปที่ 4.36 ผลตอบสนองต่างๆ ของการใช้ตัวกรองปรับตัวหาคุณสมบัติของระบบที่ไม่ทราบ โดยใช้ขนาดสเตปเท่ากับ	98
รูปที่ 4.37 ผลตอบสนองต่างๆ ของการใช้ตัวกรองปรับตัวหาคุณสมบัติของระบบที่ไม่ทราบ โดยใช้ขนาดสเตปเท่ากับ 0.1	99
รูปที่ 4.38 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวน ด้วยสัญญาณรูปไซน์ความถี่ 500 Hz	102
รูปที่ 4.39 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวน ด้วยสัญญาณรูปไซน์ความถี่ 2 kHz	102
รูปที่ 4.40 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวน ด้วยสัญญาณรูปไซน์ความถี่ 6 kHz	103
รูปที่ 4.41 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวน ด้วยสัญญาณรบกวนสุ่ม	103

รูปที่ 4.42 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวน ด้วยสัญญาณรูปไซน์ความถี่ 500 Hz	104
รูปที่ 4.43 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวน ด้วยสัญญาณรูปไซน์ความถี่ 2 kHz	104
รูปที่ 4.44 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวน ด้วยสัญญาณรูปไซน์ความถี่ 6 kHz	105
รูปที่ 4.45 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวน ด้วยสัญญาณรบกวนสุ่ม	105
รูปที่ 4.46 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวน ด้วยสัญญาณรูปไซน์ความถี่ 500 Hz	106
รูปที่ 4.47 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวน ด้วยสัญญาณรูปไซน์ความถี่ 2 kHz	106
รูปที่ 4.48 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวน ด้วยสัญญาณรูปไซน์ความถี่ 6 kHz	107
รูปที่ 4.49 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวน ด้วยสัญญาณรบกวนสุ่ม	107
รูปที่ 4.50 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวน ด้วยสัญญาณรูปไซน์ความถี่ 500 Hz	108
รูปที่ 4.51 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวน ด้วยสัญญาณรูปไซน์ความถี่ 2 kHz	108
รูปที่ 4.52 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวน ด้วยสัญญาณรูปไซน์ความถี่ 6 kHz	109
รูปที่ 4.53 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวน ด้วยสัญญาณรบกวนสุ่ม	109
รูปที่ 4.54 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวน ด้วยสัญญาณรูปไซน์ความถี่ 500 Hz	111
รูปที่ 4.55 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวน ด้วยสัญญาณรูปไซน์ความถี่ 2 kHz	111
รูปที่ 4.56 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวน ด้วยสัญญาณรูปไซน์ความถี่ 6 kHz	112
รูปที่ 4.57 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวน ด้วยสัญญาณรบกวนสุ่ม	

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านอื่นใด
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ 4.1 แสดงเสถียรภาพของตัวกรองที่ขนาดสเตรปต่างๆ

101



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาของหัวข้อปริญญาโท

ในปัจจุบันการสื่อสารเข้ามามีบทบาทในชีวิตประจำวันเป็นอย่างมาก การส่งสัญญาณไปยังปลายทางนั้นจะต้องผ่านส่วนประกอบต่างๆของระบบสื่อสาร ซึ่งโดยทั่วไปสัญญาณที่ทำการส่งเมื่อไปถึงยังปลายทางแล้วนั้นพบว่ามักถูกรบกวนทำให้คุณภาพของสัญญาณลดลง การจะนำไปใช้นั้นจำเป็นต้องทำการกรองสัญญาณรบกวนทิ้งเสียก่อน แต่สัญญาณรบกวนนั้นโดยทั่วไปมักมีขนาดและความถี่ไม่แน่นอน ทำให้การใช้ตัวกรองความถี่ธรรมดาไม่สามารถกรองได้ จึงมีความจำเป็นที่จะต้องสร้างตัวกรองความถี่ที่ปรับตัวได้ซึ่งมีความสามารถที่จะเรียนรู้และปรับตัวเอง ในการที่จะกำจัดสัญญาณที่ไม่ต้องการให้ได้มากที่สุดขึ้นมาเพื่อแก้ปัญหาดังกล่าว

โครงการนี้จึงนำเสนอการออกแบบและสร้างตัวกรองความถี่ปรับตัวได้แบบไอโออาร์ซึ่งมีข้อดีกว่าตัวกรองแบบเอฟไออาร์เนื่องจากการตอบสนองด้านการลู่เข้าที่เร็วกว่า สำหรับตัวกรองที่ออกแบบสร้างนี้จะใช้วิธีการแบบเอาต์พุตเออเรอร์และใช้อัลกอริทึมแบบแอลเอ็มเอส (Output Error Method – LMS Algorithm) โดยใช้วิธีการทางสโตแคสติก เกรเดียนท์ (Stochastic Gradient) เพื่อหาอัลกอริทึมและทำการซิมูเลชัน (Simulation) ด้วยโปรแกรมแมทแล็บ (MATLAB) ก่อนนำไปทดลองใช้จริงในระบบประมวลผลสัญญาณดิจิทัลโดยการโปรแกรมลงบอร์ดระบบประมวลผลสัญญาณดิจิทัล TMS320C31 DSK และวัดคุณสมบัติของตัวกรองที่สร้างขึ้น เนื่องจากตัวกรองปรับตัวแบบเอฟไออาร์เป็นพื้นฐานในการศึกษาตัวกรองปรับตัว ดังนั้นในการซิมูเลชันนั้นจะนำเสนอผลทั้งจากตัวกรองปรับตัวแบบเอฟไออาร์และตัวกรองแบบไอโออาร์

1.2 วัตถุประสงค์ของปริญญาโท

- 1.2.1 เพื่อศึกษาหลักการกรองสัญญาณ โดยใช้ตัวกรองแบบปรับตัวได้
- 1.2.2 เพื่อศึกษาหลักการปรับตัวของอัลกอริทึมที่ใช้
- 1.2.3 เพื่อศึกษาและประยุกต์ใช้งานภาษาแอสเซมบลีในระบบประมวลผลสัญญาณดิจิทัล
- 1.2.4 เพื่อศึกษาและประยุกต์ใช้งานระบบประมวลผลสัญญาณดิจิทัล
- 1.2.5 เพื่อสร้างตัวกรองสัญญาณแบบปรับตัวได้โดยใช้ระบบประมวลผลสัญญาณสำเร็จรูป

1.3 ขอบเขตของปริญญาโท

ปริญญาโทฉบับนี้เป็นการศึกษาออกแบบและสร้างตัวกรองสัญญาณปรับตัวได้โดยเน้น การนำไปใช้งานบนบอร์ดประมวลผลสัญญาณดิจิทัลสำเร็จรูป ซึ่งใช้การประมวลผลแบบจุดทศนิยมลอยตัว (Floating Point) โดยในขั้นตอนการออกแบบจะทำการซิมูเลชันด้วยโปรแกรมแมทแล็บและการสร้างจะทำการเขียนโครงสร้างอัลกอริทึมด้วยภาษาแอสเซมบลีป้อนให้แก่ตัวประมวลผล นอกจากนี้จะทำการวัดคุณสมบัติต่างๆของกรองที่ได้

1.4 เนื้อหาของปริญญาานิพนธ์

- บทที่ 2 กล่าวถึงทฤษฎีที่นำมาใช้ในการออกแบบและสร้างตัวกรองปรับตัวได้ซึ่งประกอบด้วย พื้นฐานระบบประมวลผลสัญญาณดิจิทัล, พื้นฐานตัวกรองปรับตัวได้ ตัวกรองปรับตัวได้แบบเอฟไออาร์และตัวกรองปรับตัวได้แบบไอโออาร์ นอกจากนี้ยังได้กล่าวถึงวิธีการวัดคุณสมบัติของตัวกรองปรับตัวได้, ตัวอย่างการนำไปใช้งานตลอดจนบอร์ดประมวลผลสัญญาณที่จะนำมาใช้ในการสร้าง
- บทที่ 3 กล่าวถึงแนวคิดในการออกแบบ โดยใช้หลักการของตัวกรองปรับตัวได้และโฟลว์ชาร์ต (Flow Chart) และบล็อกไดอะแกรม (Block Diagram) ของแต่ละส่วน
- บทที่ 4 กล่าวถึงการทดลองและผลการทดลอง
- บทที่ 5 เป็นการวิจารณ์ วิเคราะห์และสรุปผล

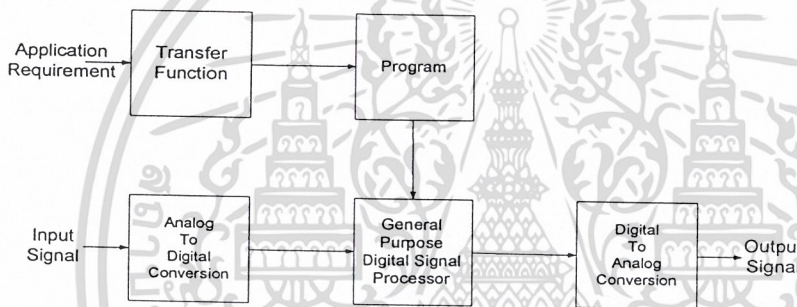


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2 ทฤษฎีและหลักการ

2.1 ทฤษฎีและหลักการพื้นฐานของระบบประมวลผลสัญญาณดิจิทัล

สัญญาณในปัจจุบันจะอยู่ในรูปของสัญญาณที่ต่อเนื่อง (Analog Signal) การประมวลผลสัญญาณโดยทั่วไปจึงมักทำในแกนเวลาที่ต่อเนื่อง แต่เนื่องจากผลเสียจากความไม่แน่นอนของระบบประมวลผลสัญญาณจึงมีการนำระบบประมวลผลสัญญาณดิจิทัลเข้ามาใช้งาน สำหรับระบบประมวลผลสัญญาณดิจิทัลนั้นจะมีส่วนประกอบเพิ่มเติมจากระบบประมวลผลสัญญาณทั่วไป 2 ส่วน คือ ส่วนแปลงสัญญาณอนาล็อกไปเป็นสัญญาณดิจิทัล (Analog to Digital Conversion) และส่วนแปลงสัญญาณดิจิทัลไปเป็นสัญญาณอนาล็อก (Digital to Analog Conversion)



รูปที่ 2.1 แสดงระบบประมวลผลสัญญาณดิจิทัล

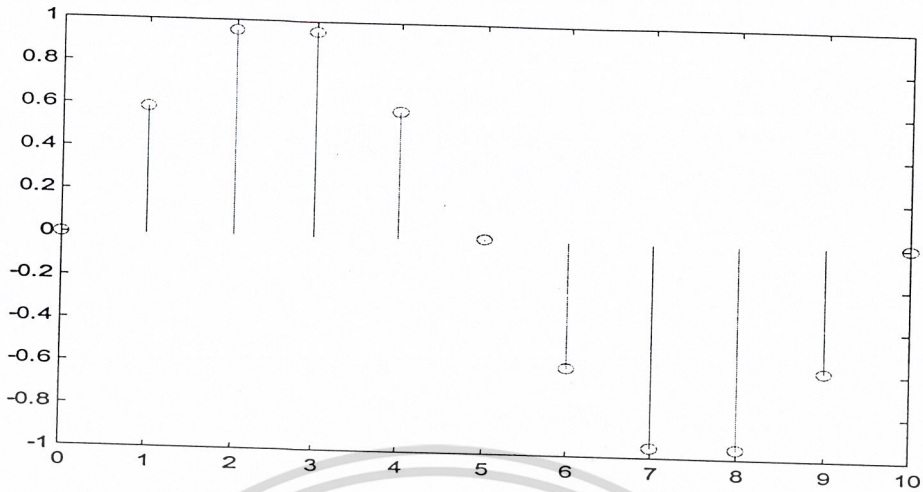
จากรูปที่ 2.1 สัญญาณอินพุตจะถูกแปลงไปอยู่ในรูปแบบสัญญาณไม่ต่อเนื่องหรือสัญญาณดิจิทัล และถูกส่งเข้าตัวประมวลผลสัญญาณดิจิทัล ซึ่งจะทำงานตามโปรแกรมที่ได้ถูกตั้งไว้และผลที่ได้จากการประมวลผลก็จะถูกแปลงเป็นสัญญาณเอาต์พุตที่ต่อเนื่องซึ่งสามารถนำไปใช้งานได้ต่อไป

2.1.1 สัญญาณเชิงเวลาไม่ต่อเนื่อง (Discrete – Time Signal)

เนื่องจากสัญญาณเชิงเวลาไม่ต่อเนื่องมักจะถูกแสดงในรูปของลำดับ (Sequence) ดังนั้นการอธิบายคุณลักษณะของสัญญาณเชิงเวลาไม่ต่อเนื่องนับจากนี้อาจใช้คำว่าลำดับแทนสัญญาณเชิงเวลาไม่ต่อเนื่องและจากที่ได้ทราบมาแล้วว่าระบบประมวลผลสัญญาณดิจิทัลจะทำการประมวลผลสัญญาณในแกนเวลาที่ discrete ดังนั้นค่าสัญญาณที่ถูกสุ่มตัวอย่าง (Sampling Signal) แต่ละค่าจะถูกแสดงโดยลำดับ $x[n]$ เมื่อ n เป็นจำนวนเต็ม เนื่องจากการซัดตัวอย่างจะกระทำที่เต็มหน่วยเวลาและค่า n จะมีค่าอยู่ระหว่าง $-\infty$ ถึง ∞ โดยทั่วไปหากค่าตัวอย่างถูกซัดตัวอย่างที่เวลา $n = 0$ แล้วก็จะมีการใช้ลูกศร \uparrow แสดงไว้ ตัวอย่างของสัญญาณที่ได้จากการซัดตัวอย่างแสดงในรูปเวกเตอร์ได้ดังนี้

$$x[n] = \{0, 0.5878, 0.9511, 0.9511, 0.5878, 0, -0.5878, -0.9511, -0.9511, -0.5878, 0\}$$

\uparrow

รูป 2.2 แสดงสัญญาณ $x[n]$

ในบางครั้งสัญญาณเชิงเวลาไม่ต่อเนื่องจะถูกสร้างโดยการซัดตัวอย่างจากสัญญาณที่ต่อเนื่อง $x_a(t)$ โดยมีคาบเวลาของการซัดตัวอย่างคงที่เท่ากับ T ค่าของ $x[n]$ จึงมีค่าเท่ากับค่าของสัญญาณที่ต่อเนื่องที่เวลา t ใดๆ และ $t = nT$ ดังนั้น

$$x[n] = x_a(t)|_{t=nT} = x_a(nT), \quad n = \dots, -2, -1, 0, 1, 2, \dots$$

จากค่าคาบเวลาในการซัดตัวอย่าง T เราสามารถหาค่าความถี่ในการซัดตัวอย่างได้เป็น $F_s = \frac{1}{T}$

2.1.2 ชนิดของสัญญาณเชิงเวลาไม่ต่อเนื่อง

สัญญาณเชิงเวลาไม่ต่อเนื่องสามารถแบ่งออกได้หลายชนิดแล้วแต่วิธีการแบ่ง

1. กรณีแบ่งโดยจำนวนค่าตัวอย่างหรือความยาวคือสัญญาณเชิงเวลาไม่ต่อเนื่องความยาวจำกัด (Finite Length) และสัญญาณเชิงเวลาไม่ต่อเนื่องความยาวไม่จำกัด (Infinite Length) นิยามโดย

- สัญญาณเชิงเวลาไม่ต่อเนื่องความยาวจำกัด เป็นสัญญาณเชิงเวลาที่มีค่า

$$N_1 \leq n \leq N_2 \quad \text{โดย } -\infty < N_1 \text{ และ } N_2 < \infty$$

และจะสามารถหาความยาวของสัญญาณนี้ได้โดย $N = N_2 - N_1 + 1$

- สัญญาณเชิงเวลาไม่ต่อเนื่องความยาวไม่จำกัด เป็นสัญญาณเชิงเวลาที่มีค่าความยาวไม่จำกัดแต่สามารถทำให้เป็นสัญญาณเชิงเวลาไม่ต่อเนื่องความยาวจำกัดได้โดยกำหนดค่าศูนย์ให้แก่ค่าตัวอย่างอื่นๆ ที่อยู่นอกระยะซึ่งวิธีการนี้เรียกว่า “Zero Padding” สัญญาณเชิงเวลาไม่ต่อเนื่องความยาวไม่จำกัดแบ่งออกเป็น 3 ลักษณะคือ

1. ลำดับด้านขวา (Right Handed Sequence) เป็นลำดับที่มีค่าตัวอย่างเป็นศูนย์สำหรับทุกค่าที่ $n < N_1$ กล่าวคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ส่วนตัวเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น ยกเว้นแต่กรณีที่ได้รับอนุญาตเป็นอย่างชัดแจ้ง

ถ้า $N_1 \geq 0$ ลำดับด้านขวานี้จะเรียกว่าลำดับคอซอล (Causal Sequence) ซึ่งมีการนำไปใช้

2. ลำดับด้านซ้าย (Left Handed Sequence) เป็นลำดับที่มีค่าตัวอย่างเป็นศูนย์สำหรับทุกค่าที่ $n > N_2$ กล่าวคือ

$$x[n] = 0 \quad ; \quad n > N_2$$

ถ้า $N_2 \leq 0$ ลำดับด้านขวานี้จะเรียกว่าลำดับแอนติคาซอล (anticausal Sequence)

3. ลำดับ 2 ข้าง (Two Sided Sequence) เป็นลำดับที่มีค่า n อยู่ในช่วง $-\infty$ ถึง ∞ กล่าวคือ

$$x[n] = 0 \quad ; \quad -\infty < n < \infty$$

2. กรณีแบ่งชนิดตามการสมมาตรแบ่งได้เป็น

- ลำดับคู่ คือลำดับที่ $x[n] = x^*[-n]$
- ลำดับคี่ คือลำดับที่ $x[n] = -x^*[-n]$

3. กรณีแบ่งชนิดตามการเป็นคาบแบ่งได้เป็น

- สัญญาณที่เป็นคาบ คือสัญญาณที่ลำดับ $\tilde{x}[n] = x[n + kN]$ เมื่อ k, N เป็นจำนวนเต็มบวก
- สัญญาณที่ไม่เป็นคาบ คือสัญญาณที่ลำดับ $\tilde{x}[n] \neq x[n + kN]$ เมื่อ k, N เป็นจำนวนเต็มบวก

โดยทั่วไปแล้วสำหรับสัญญาณเชิงเวลาไม่ต่อเนื่องความยาวจำกัดที่มีค่าเวลาเริ่มต้นที่ศูนย์นั้นค่าตัวอย่างตัวแรกจะถูกลมมติให้มีค่าเท่ากับหนึ่ง

2.1.3 การกระทำพื้นฐานของลำดับ (Basic Operation on Sequence)

ให้ $x[n]$ และ $y[n]$ เป็นลำดับที่ทราบค่าการกระทำพื้นฐานระหว่าง $x[n]$ และ $y[n]$ คือ

1. การคูณ (Production) โดยการคูณกันระหว่างค่าตัวอย่างของสองลำดับนี้จะได้ลำดับ $w_1[n]$ คือ

$$w_1[n] = x[n] \cdot y[n]$$

การประยุกต์ใช้งานของการคูณคือการสร้างลำดับที่มีความยาวจำกัดจากลำดับที่มีความยาวไม่จำกัดโดยการคูณด้วยลำดับที่มีความยาวจำกัด ซึ่งวิธีการนี้เรียกว่า “Windowing” ซึ่งมีประโยชน์มากในการประยุกต์ใช้ในตัวกรองความถี่ดิจิทัล (Digital Filter)

2. การบวก (Addition) โดยการคูณกันระหว่างค่าตัวอย่างของสองลำดับนี้จะได้ลำดับ $w_2[n]$ คือ

$$w_2[n] = x[n] + y[n]$$

3. การคูณด้วยจำนวน (Scalar Multiplication) โดยการคูณลำดับ $x[n]$ ด้วยค่า A จะได้ลำดับ $w_3[n]$ คือ

$$w_3[n] = Ax[n]$$

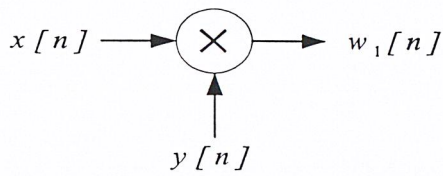
4. การเลื่อนเวลา (Time Shifting) โดยการเลื่อนเวลาไปเท่ากับ N ของลำดับ $x[n]$ จะได้ลำดับ $w_4[n]$ คือ

$$w_4[n] = x[n - N]$$

5. การกลับเวลา (Time Reversal) โดยการสร้างลำดับ $w_5[n]$ ขึ้นจากลำดับ $x[n]$ โดยการกลับเวลาจะได้

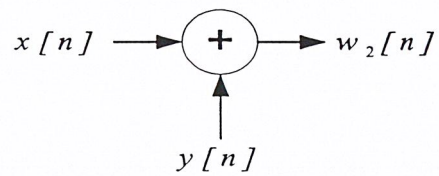
$$w_5[n] = x[-n]$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



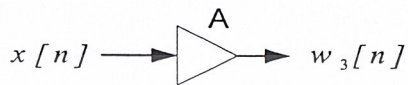
$$w_1[n] = x[n]y[n]$$

Production



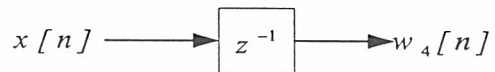
$$w_2[n] = x[n] + y[n]$$

Addition



$$w_3[n] = Ax[n]$$

Multiplication



$$w_4[n] = x[n-1]$$

Unit Delay

รูปที่ 2.3 แสดงการกระทำพื้นฐานของลำดับ

2.1.4 ลำดับพื้นฐานในการประมวลผลสัญญาณดิจิทัล

ในการประมวลผลสัญญาณดิจิทัลมีลำดับพื้นฐานที่สำคัญคือ

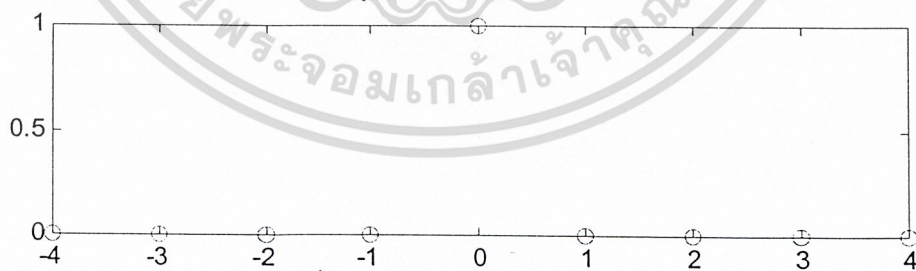
1. ลำดับยูนิตแซมเปิล (Unit Sample Sequence)

อาจเรียกว่าลำดับยูนิตอิมพัลส์ (Unit Impulse Sequence) นิยมแทนด้วยสัญลักษณ์ $\delta[n]$ โดยมีนิยามเป็น

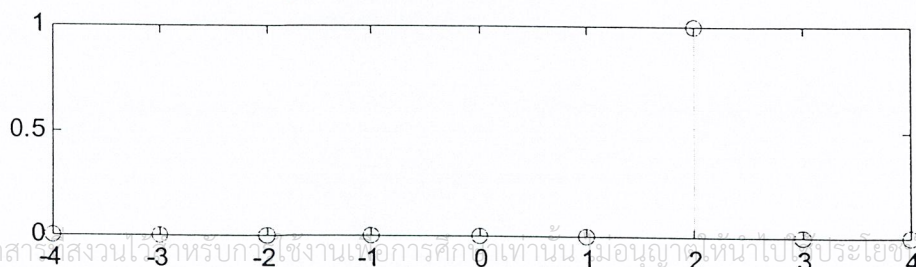
$$\delta[n] = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}$$

หากมีการเลื่อนเวลาไป จะมีนิยามเป็น

$$\delta[n-k] = \begin{cases} 1, & n = k \\ 0, & n \neq k \end{cases}$$



รูปที่ 2.4ก แสดงยูนิตอิมพัลส์ $\delta[n]$



รูปที่ 2.4ข แสดงยูนิตอิมพัลส์ $\delta[n-2]$

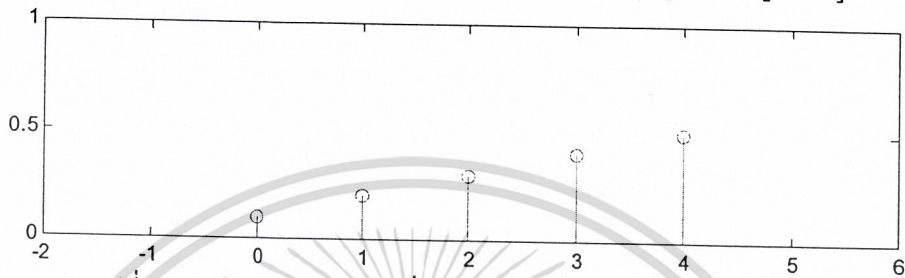
เอกสารนี้เป็นเอกสารสงวนไว้สำหรับภาควิชาวิศวกรรมไฟฟ้าใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอยู่ภายใต้เงื่อนไขเอกสารทุกครั้งที่มีการนำไปใช้

จากนิยามดังกล่าวจะพบว่าสัญญาณใดๆจะสามารถแสดงได้ในรูปของผลรวมของค่าถ่วงน้ำหนัก (Weight) ของยูนิตอิมพัลส์ที่เลื่อนเวลา สมมติให้สัญญาณ

$$y[n] = \{ 0.1, 0.2, 0.3, 0.4, 0.5 \}$$

↑

จะสามารถแสดงได้ในรูปของผลรวมของค่าถ่วงน้ำหนัก (Weight) ของยูนิตอิมพัลส์ที่เลื่อนเวลา ได้เป็น $y[n] = 0.1\delta[n] + 0.2\delta[n-1] + 0.3\delta[n-2] + 0.4\delta[n-3] + 0.5\delta[n-4]$



รูปที่ 2.5 แสดงสัญญาณ $y[n]$ ซึ่งเป็นผลรวมของยูนิตอิมพัลส์ที่เลื่อนเวลา

2. ลำดับยูนิตสเตป (Unit Step Sequence)

ลำดับยูนิตสเตปจะแทนด้วย $\mu[n]$ และมีนิยามเป็น

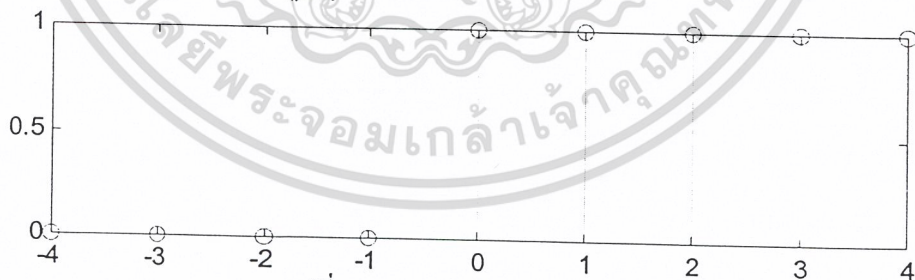
$$\mu[n] = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}$$

ส่วนยูนิตสเตปที่มีการเลื่อนเวลาไปเท่ากับ k จะมีนิยามเป็น

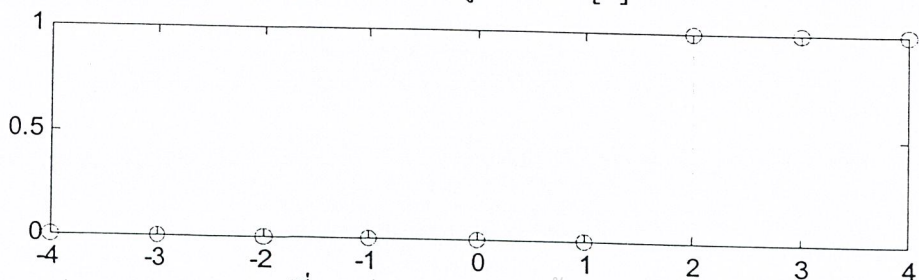
$$\mu[n-k] = \begin{cases} 1, & n \geq k \\ 0, & n < k \end{cases}$$

นอกจากนี้ความสัมพันธ์ระหว่างยูนิตสเตปและยูนิตอิมพัลส์สามารถแสดงได้เป็น

$$\mu[n] = \sum_{k=-\infty}^n \delta[k] \quad \text{และ} \quad \delta[n] = \mu[n] - \mu[n-1]$$



รูปที่ 2.6ก แสดงยูนิตสเตป $\delta[n]$



รูปที่ 2.6ข แสดงยูนิตสเตป $\delta[n-2]$ ญาติให้นำไปใช้ประโยชน์ด้านการค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับนักเรียนที่ขอทำข้อสอบโดยไม่หวังผลตอบแทนใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ลำดับรูปไซน์และลำดับเอ็กซ์โพเนนเชียล

สัญญาณ โดยทั่วไปมักจะอยู่ในรูปของลำดับสัญญาณจริงรูปไซน์ (Real Sinusoidal Sequence) ซึ่งมีค่าแอมพลิจูดคงที่ซึ่งเขียนในรูปสมการได้เป็น

$$x[n] = A \cos(\omega_0 n + \phi) \quad , \quad -\infty < n < \infty$$

เมื่อ A, ω_0 และ ϕ เป็นจำนวนเต็ม โดย A คือแอมพลิจูด, ω_0 คือค่าความเร็วเชิงมุมและ ϕ เป็นเฟสของสัญญาณ จากสมการดังกล่าวสามารถเขียนในอีกรูปแบบได้เป็น

$$x[n] = x_i[n] + x_q[n]$$

โดย $x_i[n] = A \cos \phi \cdot \cos(\omega_0 n)$

$$x_q[n] = -A \sin \phi \cdot \sin(\omega_0 n)$$

นอกจากนี้ยังมีสัญญาณเอ็กซ์โพเนนเชียลซึ่งสามารถนิยามได้โดย

$$x[n] = A \alpha^n \quad , \quad -\infty < n < \infty$$

2.1.5 ระบบเชิงเวลาไม่ต่อเนื่อง (Discrete-Time System)

ระบบเชิงเวลาไม่ต่อเนื่องโดยทั่วไปจะทำการประมวลผลลำดับอินพุตและให้ลำดับเอาต์พุตออกมา โดยในการใช้งานส่วนใหญ่จะเป็นระบบที่มีอินพุตเดียวและให้เอาต์พุตเพียงหนึ่งเอาต์พุต โดยทั่วไปแล้วระบบเชิงเวลาไม่ต่อเนื่องจะแบ่งโดยใช้ความสัมพันธ์ระหว่างอินพุตและเอาต์พุตของระบบ กล่าวคือ

- ระบบเชิงเส้น (Linear System)

เป็นระบบที่นิยมใช้กันอย่างแพร่หลายเนื่องจากระบบเชิงเส้นนี้เป็นระบบที่สามารถใช้ทฤษฎีการซ้อนทับ (Superposition Theorem) ในการอธิบายคุณลักษณะ กล่าวคือ หาก $y_1[n]$ และ $y_2[n]$ เป็นผลตอบสนองของลำดับ $x_1[n]$ และ $x_2[n]$ ดังนั้นหากป้อนอินพุต $x[n] = \alpha x_1[n] + \beta x_2[n]$ แล้วระบบจะให้ผลตอบสนอง $y[n] = \alpha y_1[n] + \beta y_2[n]$

- ระบบไม่เปลี่ยนแปลงตามเวลา (Time Invariant System)

เป็นระบบที่มีการตอบสนองต่ออินพุตไม่เปลี่ยนแปลงไปตามเวลา กล่าวคือ สำหรับอินพุต $x[n]$ ใดๆ จะมีเอาต์พุต $y[n]$ ของอินพุตตัวนั้นมีค่าเท่ากันไม่ว่าที่เวลาใดๆ กล่าวคือ หาก $y_1[n]$ เป็นผลตอบสนองของลำดับ $x_1[n]$ แล้วผลตอบสนองของอินพุต $x[n] = x_1[n - n_0]$ จะมีค่าเป็น $y[n] = y_1[n - n_0]$ เมื่อ n_0 เป็นจำนวนเต็ม

- ระบบเชิงเส้นไม่เปลี่ยนแปลงตามเวลา (Linear Time Invariant System)

เป็นระบบที่มีคุณสมบัติทั้งการไม่เปลี่ยนแปลงตามเวลาและการเป็นเชิงเส้น ระบบนี้จะมีความง่ายต่อการวิเคราะห์ทางคณิตศาสตร์และง่ายต่อการออกแบบ ดังนั้นระบบส่วนใหญ่ที่ใช้ในปัจจุบันคือระบบนี้

- ระบบกอดซอด (Causal System)

ในระบบกอดซอดนั้นผลตอบสนองตัวที่ n_0 หรือ $y[n_0]$ จะมีค่าขึ้นอยู่กับค่าตัวอย่างของ $x[n]$ ไม่กว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินพุต $x[n]$ เมื่อ $n \leq n_0$ และไม่ขึ้นกับค่าตัวอย่างอินพุตที่ $n > n_0$ หรือจะกล่าวได้ว่าผลการเปลี่ยนแปลงที่เอาท์พุทจะไม่เกิดขึ้นก่อนการเปลี่ยนแปลงที่อินพุท

2.1.6 ผลตอบสนองอิมพัลส์และผลตอบสนองต่อสเตป

ผลตอบสนองของระบบต่ออิมพัลส์เรสปอนส์เรียกว่า “ผลตอบสนองอิมพัลส์” แทนด้วย $h[n]$ ในขณะที่ผลตอบสนองต่อสเตปฟังก์ชันเรียกว่า “ผลตอบสนองสเตป” ซึ่งหลังจากนี้จะแสดงให้เห็นว่าคุณสมบัติของระบบเชิงเส้นไม่เปลี่ยนแปลงตามเวลาจะสามารถหาได้จากผลตอบสนองอิมพัลส์หรือผลตอบสนองสเตปของมัน

2.1.7 คุณลักษณะของระบบเชิงเส้นไม่เปลี่ยนแปลงตามเวลา

จากที่ได้ทราบมาแล้วว่าระบบเชิงเส้นไม่เปลี่ยนแปลงตามเวลาจะสามารถหาคุณสมบัติได้จากผลตอบสนองอิมพัลส์ของมัน ดังนั้นหากเราทราบผลตอบสนองอิมพัลส์แล้วก็จะสามารถคำนวณหาผลตอบสนองของระบบที่อินพุตใดๆ ได้ ต่อไปนี้จะแสดงให้เห็นว่าผลตอบสนองของระบบสามารถหาได้จากผลตอบสนองอิมพัลส์

สมมติให้ $h[n]$ เป็นผลตอบสนองอิมพัลส์ของระบบ เนื่องจากระบบไม่เปลี่ยนแปลงตามเวลา ทำให้ผลตอบสนองต่อ $\delta[n-1]$ มีค่าเป็น $h[n-1]$ และเช่นกันผลตอบสนองต่อ $\delta[n-2], \delta[n-3], \dots, \delta[n-k]$ มีค่าเป็น $h[n-2], h[n-3], \dots, h[n-k]$ ตามลำดับ นอกจากนี้แล้วเนื่องจากระบบเป็นระบบเชิงเส้นดังนั้นผลตอบสนองต่ออินพุต $x[n] = 0.5\delta[n+2] + 1.5\delta[n-1] - \delta[n-2] + \delta[n-4] + 0.75\delta[n-6]$ จะมีค่าเป็น $y[n] = 0.5h[n+2] + 1.5h[n-1] - h[n-2] + h[n-4] + 0.75h[n-6]$ จากตัวอย่างจะเห็นว่าสามารถเขียนลำดับอินพุต $x[n]$ ในลักษณะของผลรวมของค่าถ่วงน้ำหนักของผลตอบสนองอิมพัลส์ที่เวลาต่างๆ ได้เป็น

$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k]$$

โดย $x[k]$ คือค่าตัวอย่างที่ k ของลำดับ $x[n]$ และเนื่องจากผลตอบสนองต่อลำดับ $x[k]\delta[n-k]$ จะมีค่าเป็น $x[k]h[n-k]$ ดังนั้นผลตอบสนองของระบบจะหาได้เป็น

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

หรืออาจเขียนได้เป็น

$$y[n] = \sum_{k=-\infty}^{\infty} x[n-k]h[k]$$

ซึ่งสมการนี้มักเรียกว่า ผลบวกคอนโวลูชัน (Convolution Sum) ของลำดับ $x[n]$ และ $h[n]$ ซึ่งเขียนแทนด้วย

$$y[n] = x[n] * h[n]$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติของผลบวกคอนโวลูชัน

ให้ $x_1[n], x_2[n], x_3[n]$ เป็นลำดับใดๆ คุณสมบัติของผลบวกคอนโวลูชันแสดงได้ดังนี้

1. คุณสมบัติการจับกลุ่ม (Associative) คือ

$$(x_1[n] * x_2[n]) * x_3[n] = x_1[n] * (x_2[n] * x_3[n])$$

2. คุณสมบัติการสลับที่ (Commutative) คือ

$$x_1[n] * x_2[n] = x_2[n] * x_1[n]$$

3. คุณสมบัติการกระจาย (Distributive) คือ

$$x_1[n] * (x_2[n] + x_3[n]) = x_1[n] * x_2[n] + x_1[n] * x_3[n]$$

2.1.8 การต่อเชื่อมพื้นฐานเพื่อสร้างระบบเชิงเส้นไม่เปลี่ยนแปลงตามเวลา

วิธีการที่นิยมมี 2 รูปแบบคือ

1. การต่อแบบอนุกรม (Cascade Connection)

ในการต่อแบบอนุกรม ผลตอบสนองของระบบแรกจะถูกป้อนให้เป็นอินพุตแก่ระบบที่สองและผลตอบสนองของระบบที่สอง จะเป็นผลตอบสนองรวมของระบบ ดังนั้นผลตอบสนองอิมพัลส์ของระบบ $h_1[n]$ หากจากผลตอบสนองอิมพัลส์ของระบบแรก $h_1[n]$ และผลตอบสนองอิมพัลส์ของระบบที่สอง $h_2[n]$ คือ

$$h_1[n] = h_1[n] * h_2[n]$$

2. การต่อเชื่อมแบบขนาน (Parallel Connection)

ในการต่อเชื่อมแบบขนานนั้นผลตอบสนองของระบบทั้งสองจะถูกนำมาบวกกันโดยตรง จะได้ผลตอบสนองอิมพัลส์ของระบบ $h_1[n]$ หากจากผลตอบสนองอิมพัลส์ของระบบแรก $h_1[n]$ และผลตอบสนองอิมพัลส์ของระบบที่สอง $h_2[n]$ คือ

$$h_1[n] = h_1[n] + h_2[n]$$

2.1.9 การแบ่งชนิดของระบบเชิงเส้นไม่เปลี่ยนแปลงตามเวลา

การแบ่งชนิดของระบบเชิงเส้นไม่เปลี่ยนแปลงตามเวลานิยมแบ่งโดยลักษณะความยาวของผลตอบสนองอิมพัลส์หรือโดยลักษณะวิธีการคำนวณหาเอาต์พุตของระบบกล่าวคือ

- การแบ่งโดยลักษณะของความยาวผลตอบสนองอิมพัลส์ สามารถแบ่งออกเป็น

1. ระบบที่มีผลตอบสนองอิมพัลส์ยาวจำกัด (Finite Impulse Response : FIR)

คือระบบที่ผลบวกคอนโวลูชันสามารถเขียนในได้รูป

$$y[n] = \sum_{k=N_1}^{N_2} h[k]x[n-k]$$

เมื่อ $h[n] = 0$, $N_2 > N_1$, $n < N_1$, $n > N_2$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

โดยผลบวกคอนโวลูชันดังกล่าวจะเป็นผลบวกจำกัดและสามารถคำนวณหาค่า $y[n]$ ได้

ไม่ว่ากรณีใดๆ ทั้งสิ้น ถือว่าสงวนไว้สำหรับใช้เฉพาะในวงจำกัดของเอกสารทุกครั้งที่มีการนำไปใช้

2. ระบบที่มีผลตอบสนองอิมพัลส์ยาวไม่จำกัด (Infinite Impulse Response : IIR)

คือระบบที่ผลบวกคอนโวลูชันสามารถเขียนในได้รูป

$$y[n] = \sum_{k=0}^n x[k]h[n-k]$$

โดย $x[n]$ เป็นค่าอินพุตของระบบที่เป็นคอซอล

- การแบ่งโดยลักษณะของการคำนวณหาเอาต์พุตของระบบ สามารถแบ่งออกเป็น
 1. ระบบป้อนกลับ (Recursive) เป็นระบบที่มีการคำนวณค่าเอาต์พุตจากค่าตัวอย่างในปัจจุบัน, ค่าตัวอย่างในอดีตและค่าเอาต์พุตในอดีต
 2. ระบบไม่ป้อนกลับ (Non-Recursive) เป็นระบบที่มีการคำนวณค่าเอาต์พุตจากค่าตัวอย่างในปัจจุบันและในอดีตเท่านั้น

2.1.10 ความสัมพันธ์ของสัญญาณ (Correlation of Signal)

ในการประมวลผลสัญญาณดิจิทัลนั้น ในบางครั้งมีความจำเป็นจะต้องทำการเปรียบเทียบระหว่างสัญญาณอ้างอิงกับสัญญาณที่ต้องการประมวลผลเพื่อทำการหาความเหมือนหรือความแตกต่างตลอดจนข้อมูลอื่นๆที่รวมอยู่ในสัญญาณ จึงจำเป็นต้องหาความสัมพันธ์ของสัญญาณเพื่อใช้ในการแก้ปัญหาดังกล่าว โดยความสัมพันธ์ของสัญญาณนั้นจะทำการเปรียบเทียบในสองลักษณะคือ การหาความสัมพันธ์ของสัญญาณโดยเปรียบเทียบกับสัญญาณอื่น (Cross Correlation) และการหาความสัมพันธ์ของสัญญาณโดยเปรียบเทียบกับตัวสัญญาณเอง (Auto Correlation) กล่าวคือ

- การหาความสัมพันธ์ของสัญญาณ โดยเปรียบเทียบกับสัญญาณอื่น (Cross Correlation)
การวัดความเหมือนหรือความแตกต่างของสัญญาณพลังงาน (Energy Signal) $x[n]$ และ $y[n]$ สามารถหาได้โดยการหาลำดับครอสคอรีเลชัน $r_{xy}[\ell]$ ซึ่งนิยามโดย

$$r_{xy}[\ell] = \sum_{n=-\infty}^{\infty} x[n]y[n-\ell], \ell = 0, \pm 1, \pm 2, \dots$$

โดย ℓ คือตัวแปรล่าช้า (Lag Parameter) ซึ่งก็คือค่าความแตกต่างทางเวลาของสัญญาณทั้งสอง ด้วยย่อ xy หมายความว่า $x[n]$ เป็นลำดับอ้างอิงซึ่งมีค่าเวลาคงที่และ $y[n]$ จะมีการเลื่อนเวลาโดยเทียบกับ $x[n]$ ต่อไปจะแสดงการหาความสัมพันธ์ระหว่าง r_{xy} และ r_{yx}

จากนิยาม

$$r_{xy}[\ell] = \sum_{n=-\infty}^{\infty} x[n]y[n-\ell]$$

ทำการแทนค่า

$$n = m + \ell$$

จะได้สมการ

$$r_{xy}[\ell] = \sum_{n=-\infty}^{\infty} x[m+\ell]y[m]$$

จัดรูปใหม่ได้

$$r_{xy}[\ell] = \sum_{m=-\infty}^{\infty} y[m]x[m+\ell]$$

ซึ่งก็คือ

$$r_{xy}[\ell] = r_{yx}[-\ell]$$

จะเห็นว่า r_{xy} เป็นการกลับเวลา (Time Reversing) ของ r_{yx} นั่นเอง ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่าการหาความสัมพันธ์ของสัญญาณโดยเปรียบเทียบกับตัวสัญญาณเอง (Auto Correlation)

ลำดับอโตคอร์รีเลชันของสัญญาณพลังงาน $x[n]$ นิยามโดย

$$r_{xx}[\ell] = \sum_{n=-\infty}^{\infty} x[n]x[n-\ell]$$

จากนิยามหากให้ $\ell = 0$ จะได้สมการเป็น

$$r_{xx}[0] = \sum_{n=-\infty}^{\infty} x^2[n] = \varepsilon_x$$

ซึ่ง ε_x ก็คือพลังงานของสัญญาณ $x[n]$ นั่นเอง

2.1.11 คุณสมบัติของลำดับครอสคอร์รีเลชันและลำดับอโตคอร์รีเลชัน

การหาคุณสมบัติพื้นฐานของลำดับครอสคอร์รีเลชันและลำดับอโตคอร์รีเลชันนั้น เราจะสมมติสัญญาณพลังงานจำกัด (Finite Energy Signal) $x[n]$ และ $y[n]$ ดังนั้นพลังงานทั้งหมดของสัญญาณรวม $ax[n] + y[n-\ell]$ ก็มีค่าจำกัดด้วยและไม่เป็นลบหาก

$$\sum_{n=-\infty}^{\infty} (ax[n] + y[n-\ell])^2 = a^2 \sum_{n=-\infty}^{\infty} x^2[n] + 2a \sum_{n=-\infty}^{\infty} x[n]y[n-\ell] + \sum_{n=-\infty}^{\infty} y^2[n-\ell]$$

ซึ่งจะพบว่าเท่ากับ $a^2 r_{xx}[0] + 2ar_{xy}[\ell] + r_{yy}[0] \geq 0$

โดย $r_{xx}[0] = \varepsilon_x$, ε_x คือพลังงานของ $x[n]$

$r_{yy}[0] = \varepsilon_y$, ε_y คือพลังงานของ $y[n]$

จัดสมการในรูป

$$\begin{bmatrix} a & 1 \end{bmatrix} \begin{bmatrix} r_{xx}[0] & r_{xy}[\ell] \\ r_{xy}[\ell] & r_{yy}[0] \end{bmatrix} \begin{bmatrix} a \\ 1 \end{bmatrix} \geq 0$$

สำหรับค่า a จำกัดใดๆจะได้เมทริกซ์ $\begin{bmatrix} r_{xx}[0] & r_{xy}[\ell] \\ r_{xy}[\ell] & r_{yy}[0] \end{bmatrix}$ เป็นบวกด้วยและจะได้ว่า

$$r_{xx}[0]r_{yy}[0] - r_{xy}^2[\ell] \geq 0$$

หรือจะได้ว่า

$$|r_{xy}[\ell]| \leq \sqrt{r_{xx}[0]r_{yy}[0]} = \sqrt{\varepsilon_x \varepsilon_y}$$

จากสมการดังกล่าวแสดงให้เห็นค่าสูงของลำดับครอสคอร์รีเลชัน

หากเราให้ $x[n] = y[n]$ จะได้ว่า

$$|r_{xx}[\ell]| \leq r_{xx}[0] = \varepsilon_x$$

จากสมการดังกล่าวแสดงให้เห็นค่าสูงของลำดับอโตคอร์รีเลชัน

นอกจากนี้ในการหาคุณสมบัติของของครอสคอร์รีเลชัน สมมติให้

$$y[n] = \pm bx[n-N]$$

เมื่อ N เป็นจำนวนเต็มและ $b > 0$ ในกรณีนี้จะได้ $\varepsilon_y = b^2 \varepsilon_x$ ซึ่งทำให้

$$\sqrt{\varepsilon_x \varepsilon_y} = \sqrt{b^2 \varepsilon_x^2} = b \varepsilon_x$$

จากผลข้างต้นจะได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าการณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้เผยแพร่โดยไม่ได้รับอนุญาตอย่างอึ่งถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.12 รูปแบบนอร์มอลไลซ์ของของคอรรีเลชัน

เพื่อความสะดวกในการเปรียบเทียบสามารถเขียนลำดับอโตคอรรีเลชันและลำดับครอสคอรรีเลชันในรูปแบบนอร์มอลไลซ์ได้เป็น

$$\rho_{xx}[\ell] = \frac{r_{xx}[\ell]}{r_{xx}[0]}$$

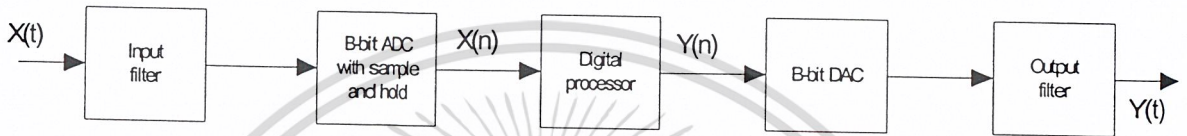
$$\rho_{xy}[\ell] = \frac{r_{xy}[\ell]}{\sqrt{r_{xx}[0]r_{yy}[0]}}$$



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ทฤษฎีตัวกรองดิจิทัล

ตัวกรองดิจิทัลเป็นตัวกรองสัญญาณแบบหนึ่ง ที่ถูกออกแบบและสร้างขึ้นโดยใช้วงจรดิจิทัล มีไคอะแกรมการทำงานดังรูปที่ 2.7 จากรูปสัญญาณอนาล็อก $x(t)$ เป็นสัญญาณอินพุทของตัวกรอง สัญญาณนี้จะถูกส่งผ่านวงจรแปลงสัญญาณอนาล็อกให้เป็นสัญญาณดิจิทัล $x(n)$ และสัญญาณดิจิทัลนี้จะถูกส่งเข้าไปคำนวณในหน่วยประมวลผล ผลที่ได้จากการคำนวณยังคงเป็นสัญญาณดิจิทัล $y(n)$ จากนั้นจะถูกส่งผ่านมายังวงจรแปลงสัญญาณดิจิทัลให้เป็นสัญญาณอนาล็อก ได้สัญญาณเอาต์พุทของตัวกรองเป็นสัญญาณอนาล็อก $y(t)$



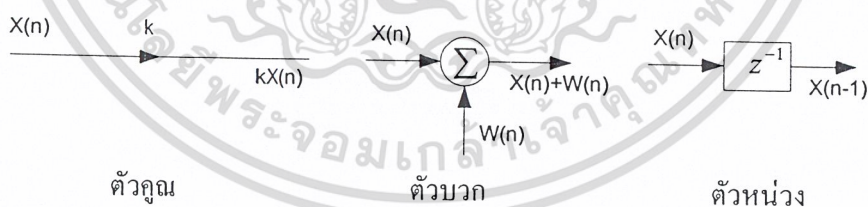
รูปที่ 2.7 ไคอะแกรมการทำงานของตัวกรองดิจิทัลพื้นฐาน

2.2.1 องค์ประกอบของตัวกรองดิจิทัล

ตัวกรองดิจิทัลจะประกอบด้วยองค์ประกอบพื้นฐานดังต่อไปนี้

- ตัวบวก (Adder)
- ตัวคูณ (Multiplier)
- ตัวหน่วง (Unit Delay)

สัญลักษณ์ขององค์ประกอบพื้นฐานทั้งสามสามารถแสดงได้ดังรูปที่ 2.8



รูปที่ 2.8 แสดงองค์ประกอบพื้นฐานทั้งสามที่ใช้เป็นส่วนประกอบของตัวกรองดิจิทัล

2.2.2 โครงสร้างของตัวกรองดิจิทัลแบบไอโออาร์ (IIR Digital Filter Structure)

ตัวกรองดิจิทัลแบบไอโออาร์สามารถเขียนอธิบายการทำงานได้ด้วยสมการผลต่างสืบเนื่องดังนี้

$$y(n) = \sum_{i=0}^M a_i x(n-i) + \sum_{i=1}^N b_i y(n-i) \quad (2.1)$$

โดยที่ $x(n)$, $y(n)$ เป็นลำดับสัญญาณเข้าและออกตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 a_i, b_i เป็นสัมประสิทธิ์ของลำดับสัญญาณเข้าและออก
 ไม่ว่าอรรถไคอะ ทั้งสิ้น อีกครั้งห้ามมิให้คัดลอกเนื้อหาและของอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
 ทำการแปลง-แซด ทั้งสองข้าง ได้ฟังก์ชันถ่ายโอนดังนี้

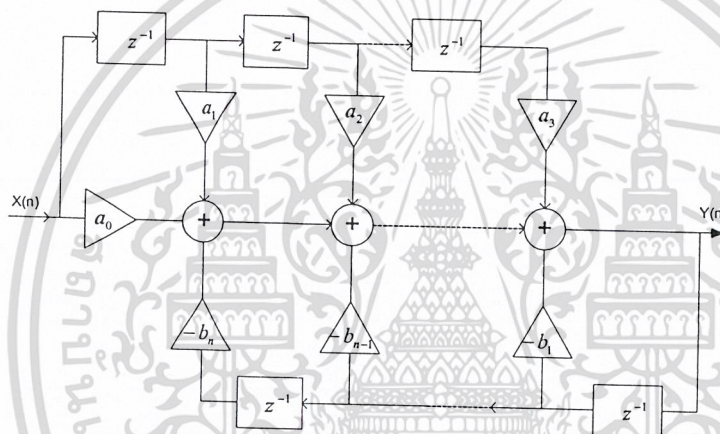
$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^M a_i z^{-i}}{\sum_{i=0}^N b_i z^{-i}} \quad (2.2)$$

จากสมการดังกล่าวสามารถจัดเป็นโครงสร้างต่างๆได้หลายรูปแบบคือ

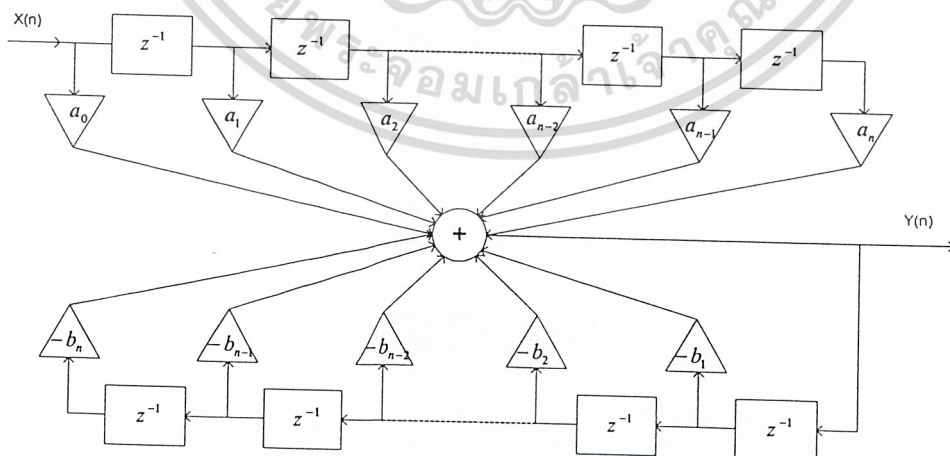
- 1 โครงสร้างแบบโดยตรง 1 (Direct Form I)
- 2 โครงสร้างแบบโดยตรง 2 (Direct Form II)
- 3 โครงสร้างแบบต่ออนุกรม (Cascade Form)
- 4 โครงสร้างแบบต่อขนาน (Parallel Form)

2.2.2.1 โครงสร้างแบบโดยตรง 1 (Direct Form I) หรือ DF1

จากสมการ (2.1) ให้ $b_0 = 1$ จากนั้นนำมาสร้างเป็นโครงสร้างดังรูปที่ 2.9



รูปที่ 2.9 เขียนแยกตัวบวกสัญญาณ



รูปที่ 2.9x เขียนรวมตัวบวกสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างแบบนี้ใช้อุปกรณห์่วงสัญญาณหรือชิฟรืจีสเตอร์ถึงสองชุด ชุดหนึ่งสำหรับบรรจุลำดับสัญญาณเข้า $x(n-i)$ อีกชุดหนึ่งจะไว้ใช้สำหรับบรรจุลำดับสัญญาณออก $y(n-i)$ โดยที่ $i = 0, 1, 2, \dots, n$ โครงสร้างของตัวกรองแบบนี้ไม่เหมาะสำหรับตัวกรองที่มีโพลวางตัวอยู่ใกล้เส้นรอบวงกลมรัศมี 1 หน่วย ทั้งนี้เนื่องจากปรากฏการณ์ซึ่งไม่เป็นเชิงเส้น เช่น สิ่งรบกวนที่เกิดจากผลการคูณ (Product Quantization Noise) และสภาพไวของสัมประสิทธิ์ (Coefficient Sensitivity) ของวงจรงสูง นิยมสร้างเฉพาะกับวงจรมีอันดับต่ำ เช่น อันดับหนึ่งหรืออันดับสอง

2.2.2.2 โครงสร้างแบบโดยตรง 2 (Direct Form II) หรือ DF2

โครงสร้างแบบนี้เริ่มด้วยการแยกฟังก์ชันถ่ายโอน หรือ $H(z)$ ออกเป็น 2 วงจรคูณแล้วต่ออนุกรมกันดังนี้

$$H(z) = \frac{Y(z)}{X(z)} = \left[\frac{1}{\sum_{i=0}^N b_i z^i} \right] \left[\sum_{i=0}^M a_i z^{-i} \right]$$

$$= H_1(z) \cdot H_2(z) \quad (2.3)$$

โดยที่ $H_1(z)$ เป็นวงจรงกรองเชิงเลขที่มีโพลเดียว

$H_2(z)$ เป็นวงจรงกรองเชิงเลขที่มีซีโรเพียงอย่างเดียว

$$\text{ให้ } H_1(z) = \frac{W(z)}{X(z)} = \frac{1}{\sum_{i=0}^N b_i z^{-i}} \quad (2.4)$$

$$\text{และ } H_2(z) = \frac{Y(z)}{W(z)} = \sum_{i=0}^M a_i z^{-i} \quad (2.5)$$

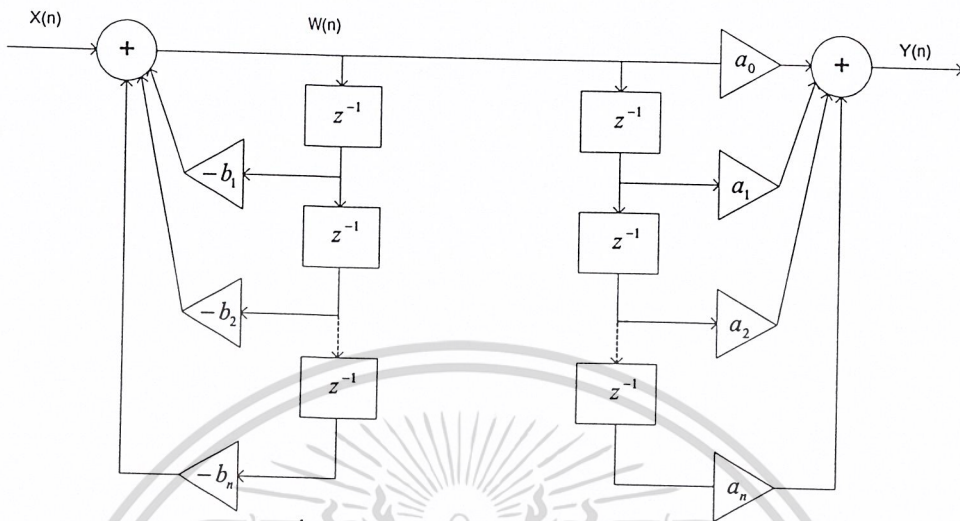
นำสมการ (2.4) และสมการ (2.5) มาเขียนในรูปสมการผลต่างสืบเนื่องได้ดังนี้

$$w(n) = x(n) - \sum_{i=1}^N b_i w(n-i) \quad (2.6)$$

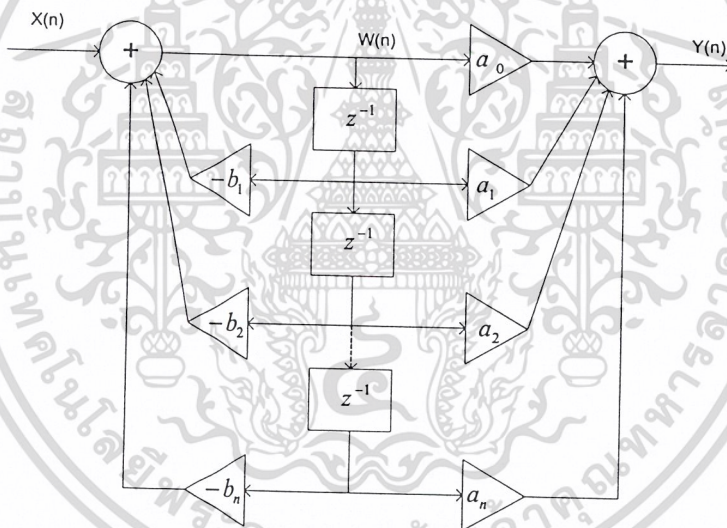
$$\text{และ } y(n) = \sum_{i=0}^M a_i w(n-i) \quad (2.7)$$

นำสมการ (2.6) และ (2.7) นำไปเขียนเป็นภาพได้ดังรูปที่ 2.10 โดยให้ $b_0 = 1$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10ก เขียนแยกอุปกรณ์หน่วยสัญญาณ



รูปที่ 2.10ข เขียนรวมอุปกรณ์หน่วยสัญญาณ

จากรูป 2.10 การคำนวณสัญญาณออก $y(n)$ ในครั้งแรกต้องคำนวณหาลำดับสัญญาณ $w(n)$ ก่อน ซึ่งเป็นสัญญาณกลาง (Intermediate Signal) ตามสมการ(2-6)แล้วจึงใช้ผลลัพธ์นี้ไปแทนในสมการ (2-7) เพื่อคำนวณหา $y(n)$ ซึ่งเป็นลำดับสัญญาณออกตามต้องการ โครงสร้างแบบโดยตรง 2 มีข้อเสียคือ ให้อุณหภูมิความไม่เป็นเชิงเส้น เช่น สิ่งรบกวนจากผลการคูณ หรือสภาพไวของสัมประสิทธิ์ เหมือนกับโครงสร้างแบบโดยตรง 1 และมีโอกาสเกิดการล้น (Overflow) ได้มากกว่าแบบโดยตรง 1 ส่วนข้อดีของโครงสร้างแบบโดยตรง 1 และโดยตรง 2 นิยมนำไปสร้างตัวกรองกันมาก ตั้งแต่จากลำดับสัญญาณเข้าของตัวคูณทุกตัวดังแสดงดังภาพที่ 2.9 และ 2.10 ลำดับสัญญาณเข้านี้ไม่ได้ขึ้นกับสัญญาณที่เพิ่งคำนวณไปเลยด้วยเหตุนี้ ทำให้เราสามารถสร้างวงจรที่สามารถทำการคูณพร้อม ๆ กันได้เลย อันเป็น

ผลให้การประมวลผลหรือการคำนวณของวงจรมีความเร็วสูง ทำให้แถบความถี่ปฏิบัติงาน (Bandwidth) ของวงจรสูงขึ้น

2.2.2.3 โครงสร้างแบบต่ออนุกรมและโครงสร้างแบบต่อขนาน

โครงสร้างตัวกรองเชิงเลขที่กล่าวมาข้างต้นคือแบบโดยตรง 1 และแบบโดยตรง 2 นั้นจะมีปัญหาในการควบคุมปรากฏการณ์ความไม่เป็นเชิงเส้นได้ยาก เมื่อค่าของ M และ N มีค่ามาก ๆ (นั่นคืออันดับสูงๆ) ดังนั้นจึงนิยมแบ่งย่อยตัวกรองเชิงเลขออกเป็นส่วนย่อยๆ อันดับต่ำมาต่ออนุกรมกัน (Cascade Form) หรือต่อขนานกัน วิธีการก็คือ นำฟังก์ชันถ่ายโอนมาเขียนในรูปผลคูณย่อย ๆ ดังแสดงในสมการที่(2.8)

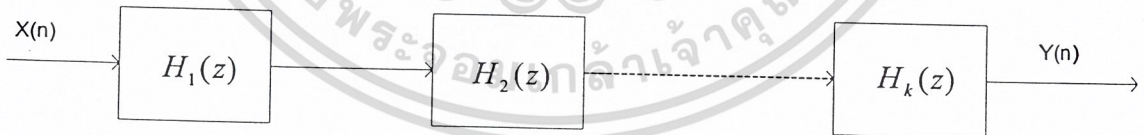
$$H(z) = \frac{Y(z)}{X(z)} = a_0 \sum_{i=0}^K H_i(z) \tag{2.8}$$

โดยที่ $H_i(z)$ คือ ฟังก์ชันถ่ายโอนของตัวกรองเชิงเลข อันดับหนึ่งหรืออันดับสอง ตามลำดับ ดังนี้

$$H_i(z) = \frac{1 + a_{1i}z^{-1}}{1 + b_{1i}z^{-1}}$$

$$H_i(z) = \frac{1 + a_{1i}z^{-1} + a_{2i}z^{-2}}{1 + b_{1i}z^{-1} + b_{2i}z^{-2}} \tag{2.9}$$

ภาพของวงจรกรองเชิงเลขแบบต่ออนุกรมเขียนได้ดังรูปที่ 2.11 โดยที่ $H_i(z)$ อาจสร้างได้ โดยใช้โครงสร้างแบบโดยตรง 1 หรือแบบโดยตรง 2 โครงสร้างแบบนี้ทำให้สภาพไวต่อสัมประสิทธิ์ต่ำกว่าแบบโดยตรง 1 และโดยตรง 2 มาก สำหรับสิ่งรบกวนนั้น ถ้าหากทำการสเกล (Scaling) สัมประสิทธิ์แต่ละภาคให้ดีแล้วก็สามารถกำจัดสิ่งรบกวนออกไปได้



รูปที่ 2.11 แสดงโครงสร้างแบบต่ออนุกรม

สำหรับโครงสร้างแบบต่อขนาน (Parallel Form) หาได้จากการนำฟังก์ชันถ่ายโอน $H(z)$ มาแยกเป็นเศษส่วนย่อย ดังแสดงในสมการ (2-10)

$$H(z) = C + \sum_{i=0}^K H_i(z) \tag{2-10}$$

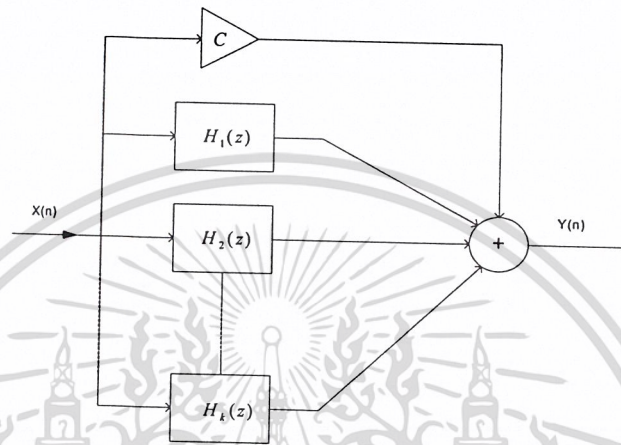
โดยที่ $H_i(z)$ อาจอยู่ในรูปตัวกรองเชิงเลข อันดับหนึ่งหรืออันดับสอง ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$H_i(z) = \frac{a_{0i}}{1 + b_{1i}z^{-1}}$$

$$H_i(z) = \frac{a_{0i} + a_{1i}z^{-1}}{1 + b_{1i}z^{-1} + b_{2i}z^{-2}} \quad (2.11)$$

โครงสร้างแบบต่อขนาน โดยทั่วไปแล้วให้ระดับของสิ่งรบกวนน้อยกว่าแบบต่ออนุกรม $H_i(z)$ สามารถนำโครงสร้างแบบโดยตรง 1 หรือแบบโดยตรง 2 มาสร้างแบบขนานแสดงดังรูปที่ 2.12



รูปที่ 2.12 แสดงโครงสร้างตัวกรองดิจิทัลแบบไอไออาร์ ที่มีโครงสร้างแบบต่อขนาน

ในการจัดโครงสร้างของตัวกรองดิจิทัลแบบไอไออาร์ ให้มีโครงสร้างเป็นแบบขนานนี้จะให้ผลดีคือ เป็นวิธีการที่เหมาะสมที่จะใช้กับระบบประมวลผลสัญญาณดิจิทัลแบบขนานซึ่งจะสามารถทำงานได้เร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

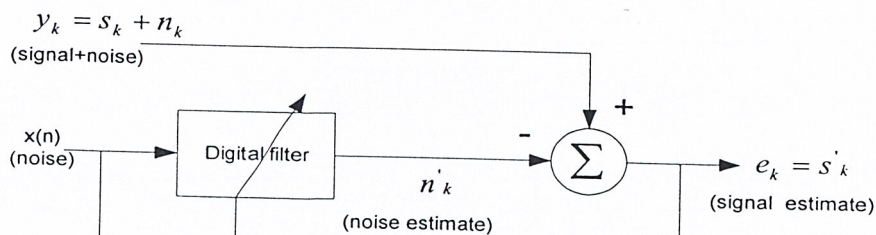
2.3 ตัวกรองปรับตัวได้ (Adaptive Filter)

ตัวกรองปรับตัวได้มีความสามารถในการกรองสัญญาณ โดยมีลักษณะการปรับค่าสัมประสิทธิ์โดยอัตโนมัติ ซึ่งออกแบบตามแนวความคิดที่ว่า จะบรรจุกลุ่มตัวแปร (Parameter) ที่ปรับค่าได้และค่านั้นจะถูกกำหนดโดยอัตโนมัติขึ้นอยู่กับการประมาณค่าคุณสมบัติทางสถิติของสัญญาณที่เกี่ยวข้อง ดังนั้น ทฤษฎีของตัวกรองปรับตัวได้จะมีความสัมพันธ์ใกล้ชิดกับทฤษฎีตัวกรองความถี่ดิจิทัลออกไป ในการออกแบบต้องการจะจัดกลุ่มของตัวแปรที่ดีที่สุดจากความรู้เกี่ยวกับคุณสมบัติของสัญญาณที่เกี่ยวข้อง ให้เหมาะสมกับกับบรรทัดฐานและความต้องการ ปัญหาหลักของตัวกรองปรับตัวได้คือการหาอัลกอริทึม (Algorithm) ที่จะปรับค่าตัวแปรได้ดีในสถานการณ์ที่ความรู้เกี่ยวกับคุณสมบัติของสัญญาณที่เกี่ยวข้องมีน้อย ซึ่งจะทำให้การการลู่เข้า (Converge) รวดเร็วยิ่งขึ้นเหมือนกับในกรณีออกแบบจากกรณีรู้คุณสมบัติของสัญญาณ โดยทั่วไปนั้นตัวกรองปรับตัวได้จะทำงานโดยใช้กระบวนการทำงานซ้ำๆ (Iterative) ในการกรองและปรับค่าตัวแปรไปในเวลาเดียวกัน ตัวกรองปรับตัวได้มีสองลักษณะคือ

1. ตัวกรองปรับตัวได้แบบลูปเปิด (Open Loop) จะทำการเรียนรู้สถิติของสัญญาณที่เกี่ยวข้องและนำผลที่ได้ไปป้อนให้แก่อัลกอริทึม โดยไม่มีการป้อนกลับ (Non-Recursive) การกระทำลักษณะนี้เพื่อให้ได้การกรองที่ดีจะต้องการกลุ่มตัวแปรจำนวนมาก ทำให้ต้องสร้างฮาร์ดแวร์ที่ซับซ้อนซึ่งมีราคาแพง ตัวกรองพวกนี้ได้แก่ ตัวกรองปรับตัวได้แบบเอฟไออาร์
2. ตัวกรองปรับตัวได้แบบลูปปิด (Close Loop) จะทำการเรียนรู้สถิติของสัญญาณที่เกี่ยวข้องและปรับปรุงกลุ่มของค่าตัวแปรปัจจุบันจากสัญญาณที่เข้ามาใหม่และสัญญาณเอาต์พุตที่ได้จากการทำงานรอบที่แล้ว การทำงานเช่นนี้จะเป็นลักษณะของการป้อนกลับ (Recursive) มีข้อดีคือต้องการกลุ่มตัวแปรจำนวนน้อย ทำให้สามารถใช้ฮาร์ดแวร์ที่มีความซับซ้อนไม่มากนักได้ ตัวกรองพวกนี้ได้แก่ ตัวกรองปรับตัวได้แบบไอไออาร์

2.3.1 แนวคิดของตัวกรองปรับตัวได้

ต่อไปจะแสดงแนวคิดของตัวกรองปรับตัวได้โดยใช้การอธิบายจากการประยุกต์ใช้เป็นตัวลดสัญญาณรบกวน (Noise Canceller) พบว่าประกอบด้วยสองส่วนคือส่วนตัวกรองดิจิทัล (Digital Filter) และส่วนที่ใช้ปรับค่าสัมประสิทธิ์ (Adaptive Algorithm) ซึ่งจะใช้ในการปรับค่าหรือปรับปรุงค่าสัมประสิทธิ์ของตัวกรองดิจิทัล ดังรูป



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ต่อผู้อื่น เนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
รูปที่ 2.13 แสดงตัวลดสัญญาณรบกวน

จากรูป 2.13 สัญญาณอินพุต 2 สัญญาณ y_k และ x_k ถูกป้อนอย่างต่อเนื่องโดยสัญญาณ y_k ประกอบด้วยสัญญาณที่ต้องการ s_k และสัญญาณรบกวน n_k โดยสมมติว่าสัญญาณ x_k เป็นสัญญาณที่จะใช้ในการตรวจสอบสัญญาณรบกวนที่ปะปนมาและมีความสัมพันธ์กับสัญญาณรบกวน n_k จะพบว่าจากการนำสัญญาณ x_k ไปผ่านตัวกรองดิจิทัลเพื่อทำการประมาณค่าเป็นสัญญาณรบกวนโดยประมาณ n'_k ดังนั้นสัญญาณที่ต้องการจะหาได้จากการหักล้างของเอาท์พุทของตัวกรองดิจิทัลจากสัญญาณ y_k ตามสมการ

$$s'_k = y_k - n'_k = s_k + n_k - n'_k$$

วัตถุประสงค์ที่สำคัญใน Noise Canceling ก็เพื่อจะสร้างค่าประมาณที่ดีที่สุดของสัญญาณที่ถูกรบกวน s'_k กลับมา

2.3.2 กรรมวิธีการพัฒนาอัลกอริทึมของตัวกรองปรับตัวได้แบบต่างๆ

ในการพัฒนาแต่ละอัลกอริทึมจะพบว่าแต่ละอัลกอริทึมมีคุณลักษณะเฉพาะตัว จึงมีความสำคัญมากในการที่จะทำความเข้าใจในความสามารถและขีดจำกัดเพื่อให้การเลือกใช้งานมีประสิทธิภาพ

1.กรรมวิธีการแบบสโตแคสติก เกรเดียนท์ (Stochastic Gradient Approach)

เป็นกรรมวิธีที่วัดค่าความผิดพลาดโดยใช้ค่าเฉลี่ยผลต่างกำลังสองและพยายามลดค่าเฉลี่ยผลต่างกำลังสองโดยการทำการหาค่าจุดต่ำสุดบนเออร์เรอร์เพอร์ฟอร์แมนซ์เซอร์เฟซ (Error Performance Surface) ซึ่งจุดต่ำสุดจะให้ค่าสัมประสิทธิ์ที่ทำให้ค่าเฉลี่ยผลต่างกำลังสองมีค่าต่ำสุด การหาอัลกอริทึมเพื่อใช้ในการปรับค่าสัมประสิทธิ์จะทำการแก้สมการของวินเนอร์และฮอปฟ์ (Wiener-Hopf Equation) อัลกอริทึมในกลุ่มนี้ที่นิยมใช้คือลีสต์มีนสแควร์ (Least Mean Square) หรือเรียกอีกอย่างว่าแอลเอ็มเอส อัลกอริทึม (LMS Algorithm)

2.กรรมวิธีการประมาณแบบลีสต์สแควร์ (Least Square Estimation)

กรรมวิธีนี้ใช้วิธีการลดคอสต์ ฟังก์ชัน (Cost Function) โดยอาจใช้วิธีการของรีเคอร์ซีฟ ลีสต์สแควร์ (Recursive Least Square: RLS) ซึ่งสามารถแบ่งออกเป็น 3 อัลกอริทึม ดังนี้

1. รีเคอร์ซีฟ ลีสต์สแควร์ อัลกอริทึมมาตรฐาน (Standard RLS Algorithm)
2. รีเคอร์ซีฟ ลีสต์สแควร์ อัลกอริทึมโดยการถอดราก (Square Root RLS Algorithm)
3. รีเคอร์ซีฟ ลีสต์สแควร์ อัลกอริทึมแบบเร็ว (Fast RLS Algorithm)

2.3.3 การวัดประสิทธิภาพในระบบปรับตัว

ในระบบปรับตัวทุกระบบมีความจำเป็นที่จะต้องทำการวัดประสิทธิภาพ โดยทำเปรียบเทียบระหว่างอัลกอริทึมต่างๆ ในตัวกรองปรับตัวก็เช่นกัน นอกจากจะต้องทำการเปรียบเทียบประสิทธิภาพของตัวกรองแล้ว ยังต้องทำการเปรียบเทียบประสิทธิภาพของอัลกอริทึมอีก เพื่อช่วยในการตัดสินใจเลือกใช้อัลกอริทึมที่ให้การตอบสนองที่ดี การวัดประสิทธิภาพของอัลกอริทึมมีหลายวิธีการดังจะได้อธิบายต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่หรือใช้เพื่อการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ความเร็วในการลู่เข้า (Convergence Rate)

อัตราการลู่เข้าของระบบปรับตัวเป็นคุณสมบัติที่สำคัญมากซึ่งจะต้องทำการวัดเพื่อให้ได้ตามความต้องการของงานที่จะนำไปใช้ซึ่งโดยทั่วไปแล้วความเร็วในการลู่เข้านั้นสามารถเปรียบเทียบประสิทธิภาพของอัลกอริทึมได้ อย่างไรก็ตามการเลือกใช้อัลกอริทึมไม่จำเป็นต้องใช้อัลกอริทึมที่มีความเร็วสูงสุดเนื่องจากการเพิ่มความเร็วราคาในการสร้างและความซับซ้อนของวงจรก็จะสูงขึ้นเช่นกัน การเลือกใช้จึงขึ้นอยู่กับความจำเป็นในการใช้งานมากกว่า

2. ค่าความผิดพลาดเฉลี่ยกำลังสอง (Mean Square Error)

จะวัดในลักษณะของค่าความผิดพลาดเฉลี่ยกำลังสองที่น้อยที่สุด (Minimum Mean Square Error :MMSE) ซึ่งใช้ในการวัดความสามารถในการทำงานของระบบในการลดสัญญาณรบกวน (Eliminating Noise), ทำนายสัญญาณ (Signal Prediction) หรือแยกแยะระบบ (System Identifying) แล้วแต่ว่าเป็นระบบอะไร โดยทั่วไปแล้วค่าความผิดพลาดเฉลี่ยกำลังสองขึ้นกับหลายปัจจัย เช่น โครงสร้างของตัวกรอง ความไวของสัมประสิทธิ์ ตลอดจนสัญญาณรบกวนต่างๆ ไป

3. ความถูกต้องในการประมาณค่าตัวแปร (Parameter Estimation Accuracy)

ความถูกต้องในการประมาณค่าตัวแปรเป็นปัจจัยที่มีความสำคัญมากเมื่อมีการนำระบบปรับตัวไปใช้ในการหาคุณลักษณะของระบบที่ไม่ทราบ (System Identifying) ซึ่งความถูกต้องในการประมาณค่าตัวแปรมีมาก จะทำให้ได้แบบจำลองของระบบที่ต้องการใกล้เคียงความจริงมากขึ้น

4. ความซับซ้อนในการประมวลผล (Computation Complexity)

ความซับซ้อนในการประมวลผลมีความสำคัญมากในการนำระบบปรับตัวไปใช้งานจริง โดยทั่วไปต้องการความซับซ้อนในการประมวลผลที่ต่ำ ทำให้การสร้างมีราคาถูก นอกจากนั้นหากนำไปโปรแกรมลงในตัวประมวลผลสัญญาณสำเร็จรูปแล้วนั้นก็ต้องคำนึงถึงความสามารถในการทำงานของตัวประมวลผลเทียบกับความซับซ้อนในการประมวลผลของอัลกอริทึมด้วย

5. เสถียรภาพ (Stability)

ความมีเสถียรภาพของระบบมีความสำคัญอย่างมากในการใช้งานตัวกรองปรับตัวได้โดยเฉพาะตัวกรองแบบไอโออาร์ เนื่องจากอาจเกิดการเลื่อนของโพลออกนอกวงกลมหนึ่งหน่วย ทำให้เกิดการออสซิลเลตและทำให้ค่าลู่ออก (Diverge) จนไม่สามารถทำงานได้ การออกแบบจึงต้องคำนึงถึงเสถียรภาพของตัวกรองด้วย

6. ความคงทนของระบบ (Robustness)

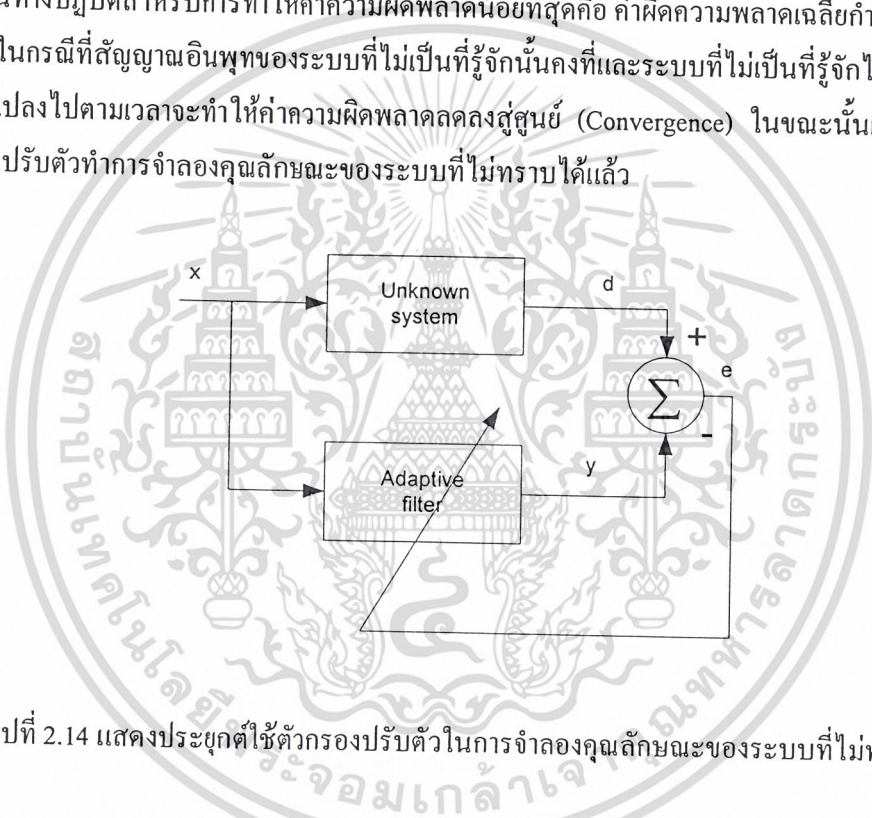
เนื่องจากค่าอินพุทของตัวกรองปรับตัวได้มักมีค่าต่างๆที่ไม่แน่นอนจึงมีความจำเป็นที่ตัวกรองจะต้องมีความทนต่อความแปรปรวนของสัญญาณ โดยทั่วไปความคงทนของระบบเป็นค่าที่ยากต่อการวัดในการทดลองโดยทั่วไปจึงไม่ทำการวัดคุณลักษณะนี้

2.3.4 การประยุกต์ใช้งานตัวกรองปรับตัวได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ตัวกรองปรับตัวได้มีคุณสมบัติในการปรับตัวได้อย่างอัตโนมัติทำให้ถูกนำมาประยุกต์
ใช้ในอุปกรณ์หลายๆอย่างต่อไปจะทำการอธิบายตัวอย่างการนำตัวกรองปรับตัวได้ไปประยุกต์ใช้งาน

1. การประยุกต์ในการจำลองคุณลักษณะของระบบที่ไม่ทราบ (System Identification)

สมมุติว่ามีระบบที่ไม่เป็นที่รู้จักและต้องการพิจารณาผลตอบสนองของระบบนั้นต่อสัญญาณที่ป้อนให้กับระบบ โดยสมมุติว่าระบบนี้ไม่เปลี่ยนแปลงไปตามเวลาและมีคุณสมบัติเชิงเส้น และต้องการที่จะพัฒนาแบบจำลอง (Model) สำหรับระบบนี้โดยใช้ตัวกรองปรับตัว เพื่อที่จะสร้างสัญญาณ เอาท์พุทให้เหมือนกับเอาท์พุทของระบบที่ไม่เป็นที่รู้จัก โดยการป้อนอินพุทที่เหมือนกันให้แก่ระบบทั้งสองและทำการเปรียบเทียบเอาท์พุททั้งสอง เพื่อสร้างสัญญาณผิดพลาด ซึ่งเป็นความแตกต่างระหว่างสัญญาณทั้งสอง อาจกล่าวได้ว่าการทำงานของตัวกรองปรับตัวเป็นการพยายามปรับผลตอบสนองให้เหมือนกับผลตอบสนองของระบบที่ไม่เป็นที่รู้จัก เพื่อที่จะทำให้สัญญาณผิดพลาดมีน้อยที่สุด ซึ่งวิธีการที่ถูกใช้บ่อยๆในทางปฏิบัติสำหรับการทำให้ค่าความผิดพลาดน้อยที่สุดคือ ค่าผิดพลาดเฉลี่ยกำลังสอง และสำหรับในกรณีที่สัญญาณอินพุทของระบบที่ไม่เป็นที่รู้จักนั้นคงที่และระบบที่ไม่เป็นที่รู้จักไม่เปลี่ยนแปลงไปตามเวลาจะทำให้ค่าความผิดพลาดลดลงสู่ศูนย์ (Convergence) ในขณะนั้นเองกล่าวได้ว่าตัวกรองปรับตัวทำการจำลองคุณลักษณะของระบบที่ไม่ทราบได้แล้ว



รูปที่ 2.14 แสดงประยุกต์ใช้ตัวกรองปรับตัวในการจำลองคุณลักษณะของระบบที่ไม่ทราบ

2. การพัฒนาอีควอลไลเซอร์แบบปรับตัวได้ (Adaptive Equalization)

ระบบการส่งข้อมูลทั่วไปจะพยายามใช้ประโยชน์จากช่องสัญญาณที่มีอยู่ให้มีประสิทธิภาพที่สุด โดยออกแบบระบบการส่งข้อมูลด้วยอัตราสูงที่สุดเท่าที่เป็นไปได้ ภายใต้ความน่าเชื่อถือที่กำหนด ซึ่งปกติถูกวัดอยู่ในรูปของอัตราการผิดพลาดหรือความน่าจะเป็นเฉลี่ยของการผิดพลาดของสัญลักษณ์ (Average probability of symbol error) การส่งข้อมูลดิจิทัลผ่านช่องทางการสื่อสารแบบเชิงเส้นถูกจำกัดด้วยองค์ประกอบสองอย่าง

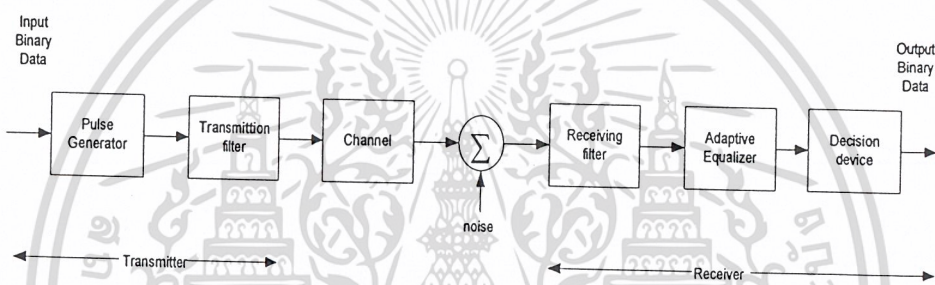
1. การแทรกซ้อนระหว่างสัญลักษณ์ (Intersymbol Interference) เกิดเป็นการแทรกซ้อนของสัญลักษณ์ที่ถูกส่ง ซึ่งเป็นผลมาจากการเบี่ยงเบนของการตอบสนองความถี่ในช่องทาง

เอกสารนี้เผยแพร่ภายใต้ลิขสิทธิ์ของสถาบันวิจัยและพัฒนาเทคโนโลยีสารสนเทศและการสื่อสารของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ไม่ว่ากรณีใด กรณีหนึ่ง ซึ่งโดยทั่วไปมักเกิดในอุปกรณ์ต่างๆของระบบสื่อสาร

สำหรับช่องทางสื่อสารที่จำกัดเช่น โทรศัพท์ปกติจะพบว่าการแทรกซ้อนระหว่างสัญลักษณ์เป็นองค์ประกอบสำคัญที่ใช้พิจารณาในการกำหนดอัตราส่งของข้อมูล

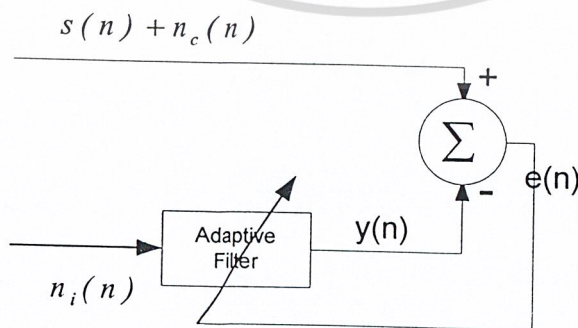
เพื่อที่จะแก้ปัญหาการแทรกซ้อนระหว่างสัญลักษณ์ต้องมีตัวกรองโดยถ้าคุณลักษณะของช่องสัญญาณเป็นที่รู้อย่างแน่นอนแล้ว จะสามารถออกแบบตัวกรองทั้งภาคส่งและภาครับได้ ซึ่งจะลดผลของการแทรกซ้อนระหว่างสัญลักษณ์ลงได้และยังช่วยลดสัญญาณรบกวนที่เพิ่มขึ้นมาแต่อย่างไรก็ตามในทางปฏิบัติ เราพบว่าคุณลักษณะของช่องสัญญาณสื่อสารนั้นเป็นแบบสุ่มฉะนั้นการใช้ตัวกรองภาคส่งและภาครับที่ออกแบบโดยคุณลักษณะของช่องสัญญาณเฉลี่ยจะไม่เพียงพอที่จะลดการแทรกซ้อนระหว่างสัญลักษณ์ จึงจำเป็นที่จะต้องใช้อีควอลไลเซอร์แบบปรับตัวได้ ซึ่งจะลดการแทรกซ้อนระหว่างสัญลักษณ์ได้ดีขึ้น โดยพื้นฐานสำหรับการอีควอลไลซ์ ของระบบการส่งข้อมูลที่มีอยู่ คือการเพิ่มขนาดสัญญาณที่บางย่าน (Pre-Equalization) ที่ภาคส่งและการลดขนาดสัญญาณที่บางย่าน (Post-Equalization) ที่ภาครับ



รูปที่ 2.15 แสดงการใช้อีควอลไลเซอร์แบบปรับตัวได้ในระบบส่งสัญญาณ

3 การกำจัดสัญญาณรบกวน (Noise Cancellation)

การนำตัวกรองปรับตัวได้มาประยุกต์ใช้ในการกำจัดสัญญาณรบกวนนั้น โดยใช้สัญญาณรบกวนที่มีความสัมพันธ์กับสัญญาณรบกวนที่ป้อนมาในสัญญาณที่ต้องการ ในการกำจัดสัญญาณรบกวน โดยการป้อนสัญญาณและ โครงสร้างของระบบกำจัดสัญญาณรบกวนเป็นไปตามรูป

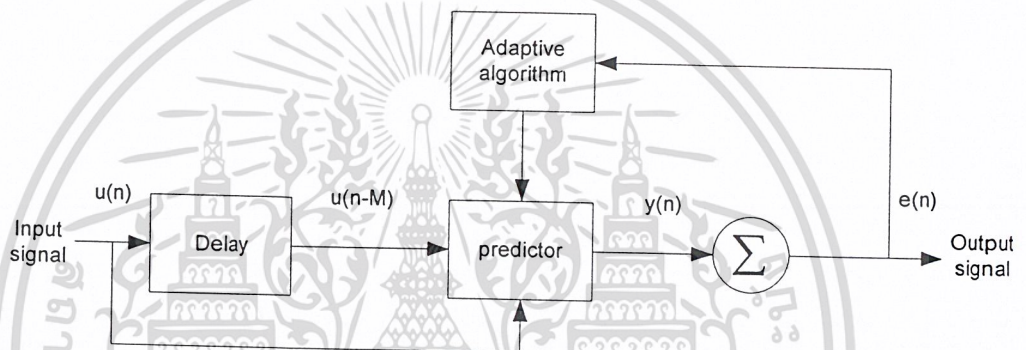


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการงานที่ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 รูปที่ 2.16 แสดง โครงสร้างระบบกำจัดสัญญาณรบกวน โดยใช้ตัวกรองปรับตัวได้
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปป้อนสัญญาณรบกวน $n_i(n)$ ซึ่งมีความสัมพันธ์กับสัญญาณรบกวน $n_c(n)$ ให้แก่ระบบจะได้เอาที่พหุของตัวกรองปรับตัว $y(n)$ แล้วจึงนำไปหักล้างกับสัญญาณรบกวน $n_c(n)$ ซึ่งปะปนอยู่กับสัญญาณที่ต้องการ $s(n)$ ทำให้ได้สัญญาณที่ต้องการ $e(n)$ ซึ่งเป็นสัญญาณที่มีสัญญาณรบกวนน้อยลงและสัญญาณนี้จะถูกป้อนให้กับตัวกรองเพื่อใช้ในการเปรียบเทียบในการปรับค่าสัมประสิทธิ์อีกทีหนึ่ง

4. การพัฒนาอุปกรณ์เพิ่มประสิทธิภาพสาย (Adaptive Line Enhancer)

อุปกรณ์เพิ่มประสิทธิภาพสาย เป็นอุปกรณ์ที่สามารถใช้แยกสัญญาณแบนด์แคบซึ่งฝังอยู่ในสัญญาณแบนด์กว้าง ซึ่งขึ้นกับการใช้งาน สัญญาณที่ต้องการจะแยกออกอาจเป็นสัญญาณที่สนใจหรือสัญญาณรบกวนที่ไม่ต้องการ ตัวอย่างของการการใช้เช่นการลดสัญญาณรบกวนความถี่ 50-60 Hz ออกจากสัญญาณของอุปกรณ์ทางการแพทย์



รูปที่ 2.17 แสดงอุปกรณ์เพิ่มประสิทธิภาพสาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ตัวกรองปรับตัวได้แบบเอฟไออาร์ (Adaptive FIR Filter)

ในการศึกษาตัวกรองปรับตัวนั้น โดยทั่วไปมักจะเริ่มจากการศึกษาตัวกรองปรับตัวได้แบบเอฟไออาร์เสียก่อน เนื่องจากตัวกรองปรับตัวแบบเอฟไออาร์มีคุณสมบัติบางประการที่ดีกว่า กล่าวคือ เสถียรภาพของระบบที่ดีกว่าเนื่องจากสัมประสิทธิ์ของตัวกรองที่มีค่าจำกัด อัลกอริทึมที่ใช้ในการปรับตัวก็ง่ายกว่า นอกจากนั้นแล้วประสิทธิภาพของตัวกรองก็สามารถมองได้ในลักษณะของการลู่เข้าและเสถียรภาพ ต่อไปจะแสดงการหาอัลกอริทึมสำหรับตัวกรองปรับตัวแบบเอฟไออาร์ โดยใช้อัลกอริทึมแบบลิสต์มินสแควร์หรืออัลกอริทึมแบบแอลเอ็มเอส

ตัวกรองแบบเอฟไออาร์ ซึ่งมีสัญญาณอินพุตเป็น $x(n)$ มีสมการทั่วไปคือ

$$y(n) = \sum_{i=0}^{N-1} w_i(n)x(n-i) \quad (2.12)$$

ให้สัญญาณที่ต้องการเป็น $d(n)$ แล้วสัญญาณความผิดพลาด $e(n)$ สามารถนิยามโดย

$$e(n) = d(n) - y(n) \quad (2.13)$$

สำหรับความต้องการโดยทั่วไปของตัวกรองปรับตัว การเลือกสัมประสิทธิ์ $w_0(n), w_1(n), \dots, w_i(n)$ นั้นจะอยู่บนพื้นฐานของการลดสัญญาณความผิดพลาดให้น้อยที่สุด โดยยึดเอาค่าเฉลี่ยกำลังสองในการเปรียบเทียบ

โดยทั่วไปนั้นแอลเอ็มเอสอัลกอริทึมจะใช้วิธีการปรับสัมประสิทธิ์แบบ สตีปเปสต์ เดสเซนท (Steepest Descent) ซึ่งก็คือ

$$W(n+1) = W(n) - \mu \nabla e^2(n) \quad (2.14)$$

เมื่อ $W(n) = [w_0(n) \ w_1(n) \ \dots \ w_{N-1}(n)]^T$ เป็นเมตริกซ์สัมประสิทธิ์ μ เป็นค่าขนาดสเกล และ ∇ เป็นค่าเกรเดียนท์เวกเตอร์ซึ่งกำหนดโดย

$$\nabla = \left[\frac{\partial}{\partial w_0} \quad \frac{\partial}{\partial w_1} \quad \dots \quad \frac{\partial}{\partial w_{N-1}} \right]^T \quad (2.15)$$

ดังนั้นเกรเดียนท์ตัวที่ i ของเกรเดียนท์เวกเตอร์ $\nabla e^2(n)$ คือ

$$\frac{\partial e^2(n)}{\partial w_i} = 2e(n) \frac{\partial e(n)}{\partial w_i} \quad (2.16)$$

แทนค่า $e(n) = d(n) - y(n)$ ลงในสมการข้างต้นจะได้เป็น

$$\frac{\partial e^2(n)}{\partial w_i} = -2e(n) \frac{\partial y(n)}{\partial w_i} \quad (2.17)$$

แทนค่า $y(n) = \sum_{i=0}^{N-1} w_i(n)x(n-i)$ ลงในสมการข้างต้นจะได้เป็น

$$\frac{\partial e^2(n)}{\partial w_i} = -2e(n)x(n-i) \quad (2.18)$$

ดังนั้นจากสมการข้างต้นจะพบว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ $\nabla e^2(n) = -2e(n)X(n)$ ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า (2.19)

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ $X(n) = [x(n) \ x(n-1) \ \dots \ x(n-N+1)]^T$ หลังจากทำการแทนค่าแล้วจะได้สมการปรับสัมประสิทธิ์เป็น

$$W(n+1) = W(n) + 2\mu e(n)X(n) \quad (2.20)$$

จากสมการทั้งหมดเราจะได้อัลกอริทึมโดยสรุปเป็น

อินพุต: เวกเตอร์สัมประสิทธิ์ $W(n) = [w_0(n) \ w_1(n) \ \dots \ w_{N-1}(n)]^T$

อินพุตเวกเตอร์ $X(n) = [x(n) \ x(n-1) \ \dots \ x(n-N+1)]^T$

สัญญาณที่ต้องการ $d(n)$

เอาต์พุต: ฟิลเตอร์เอาต์พุต $y(n)$

เวกเตอร์สัมประสิทธิ์ที่ปรับแล้ว $W(n+1)$

1. การกรองสัญญาณ

$$y(n) = W^T(n)X(n)$$

2. การประมาณค่าความผิดพลาด

$$e(n) = d(n) - y(n)$$

3. การปรับสัมประสิทธิ์

$$W(n+1) = W(n) + 2\mu e(n)X(n)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 ตัวกรองปรับตัวได้แบบไอไออาร์ (Adaptive IIR Filter)

ในบทที่ผ่านมาได้กล่าวถึงภาพรวมของตัวกรองปรับตัวได้ ซึ่งการอธิบายส่วนใหญ่จะไม่ใช้สมการทางคณิตศาสตร์เพื่อให้ง่ายต่อการทำความเข้าใจ แต่เนื่องจากการออกแบบสร้างตัวกรองปรับตัวได้มีความจำเป็นอย่างยิ่งที่จะต้องใช้คณิตศาสตร์ในการออกแบบและจัดโครงสร้างตัวกรองให้ได้ตามต้องการ ในบทนี้จะทำการอธิบายถึงรายละเอียดและโครงสร้างของตัวกรองปรับตัวแบบไอไออาร์โดยใช้แอลเอ็มเอสอัลกอริทึม

ในปัจจุบันการใช้งานตัวกรองปรับตัวได้แบบป้อนกลับเป็นที่นิยมน้อยกว่าแบบไม่ป้อนกลับ เนื่องจากตัวกรองแบบป้อนกลับมีข้อจำกัดบางประการคือ

1. ตัวกรองปรับตัวได้แบบป้อนกลับจะไม่เสถียรได้ง่าย เนื่องจากโพลอาจเลื่อนตัวออกนอกวงกลมหนึ่งหน่วยในขณะปรับตัว

2. เพอร์ฟอร์แมนซ์ฟังก์ชันของตัวกรองปรับตัวได้แบบป้อนกลับ ซึ่งเป็นฟังก์ชันที่ในการหาสัมประสิทธิ์ของตัวกรอง จะมีค่าต่ำสุดในบริเวณ (Local Minima) หลายค่า

อย่างไรก็ตามปัญหาของความไม่เสถียรสามารถแก้ไขได้ โดยการตรวจสอบค่าสัมประสิทธิ์ของฟิลเตอร์ หลังจากการเสร็จสิ้นการปรับตัวในแต่ละขั้น และจำกัดค่าสัมประสิทธิ์ของตัวกรองให้อยู่ในช่วงที่เสถียรแทนแต่จะเป็นการยากและซับซ้อนถ้าลำดับของตัวกรองมากขึ้นและเนื่องจากตัวกรองปรับตัวมีค่าต่ำสุดหลายค่าดังนั้นการลู่เข้าสู่ค่าต่ำสุดของเพอร์ฟอร์แมนซ์ฟังก์ชันจะไม่สามารถมั่นใจได้ แต่ยังมีวิธีการแก้ปัญหานี้ได้โดย

1. การแก้ปัญหาค่าต่ำสุดในบริเวณสามารถทำได้โดยใช้เพอร์ฟอร์แมนซ์ฟังก์ชันกำลังสองซึ่งคล้ายกับของตัวกรองปรับตัวแบบเอฟไออาร์ ซึ่งการแก้ปัญหานี้ทำให้เกิดการสร้างตัวกรองปรับตัวแบบอิกเวชันเออร์เรอร์ (Equation Error Method)

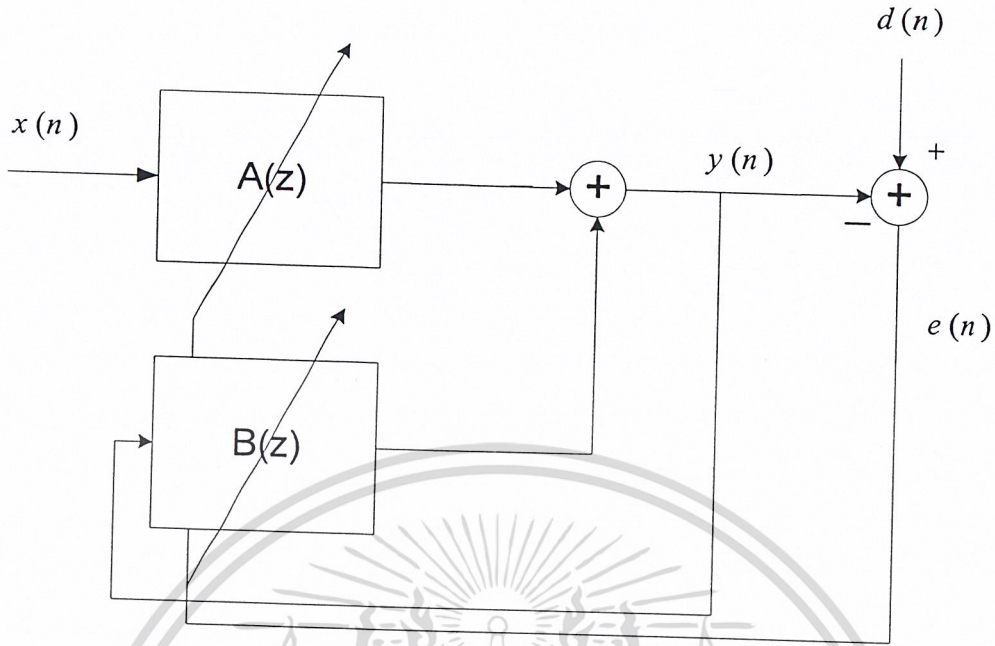
2. การแก้ปัญหาค่าต่ำสุดในบริเวณสำหรับการใช้งานที่เฉพาะ จะจำกัดฟังก์ชันของตัวกรองปรับตัวแบบไอไออาร์ที่ใช้โดยให้เพอร์ฟอร์แมนซ์ฟังก์ชันอยู่ในบริเวณที่มีค่าต่ำสุดของเพอร์ฟอร์แมนซ์ฟังก์ชันทำให้ไม่มีค่าต่ำสุดในบริเวณ วิธีนี้ทำให้สามารถวิธีเอาท์พุทเออร์เรอร์ในการสร้างได้เนื่องจากมั่นใจได้ว่าตัวกรองปรับตัวแบบเอฟไออาร์จะลู่เข้าสู่ค่าต่ำสุดของเพอร์ฟอร์แมนซ์ฟังก์ชันอย่างแน่นอน

ต่อไปจะแสดงวิธีการสร้างตัวกรองปรับตัวได้ด้วยวิธีต่างๆ

2.5.1 วิธีการแบบเอาท์พุทเออร์เรอร์ (The Output Error Method)

วิธีการแบบเอาท์พุทเออร์เรอร์เกิดจากการใช้ทฤษฎีตัวกรองเวียนเนอร์ (Wiener Filter Theory) ในการพัฒนาอัลกอริทึมสำหรับออกแบบตัวกรองปรับตัวแบบไอไออาร์โดยอธิบายได้ดังรูปที่ 2.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.18 ตัวกรองปรับตัวแบบโอไออาร์โดยใช้วิธีการแบบเอาต์พุตเออร์เรอร์

จากทฤษฎีตัวกรองเวียนเนอร์ สัมประสิทธิ์ของฟังก์ชันป้อนกลับจะเป็น

$$W(z) = \frac{A(z)}{1 - B(z)} \tag{2.21}$$

โดย $A(z)$ และ $B(z)$ เป็นโพลีโนเมียลของ Z ที่จะถูกปรับค่าโดยการลดค่าความผิดพลาดของเอาต์พุต (Output Error) $e(n)$ ให้น้อยที่สุดในเทอมของค่าเฉลี่ยกำลังสอง ดังนั้นเราต้องการหาค่าต่ำสุดของเพอร์ฟอร์มานซ์ฟังก์ชัน $\xi = E[e^2(n)]$ อย่างไรก็ตามฟังก์ชัน ξ เป็นฟังก์ชันที่มีค่าต่ำสุดหลายค่า ทำให้การเข้าสู่ของอัลกอริทึมที่ใช้วิธีเกรเดียนท์อาจเข้าสู่ค่าอื่นแต่ในขั้นตอนนี้เราจะไม่สนใจความผิดพลาดนี้ และจะพัฒนาแอลเอ็มเอสอัลกอริทึมสำหรับการปรับสัมประสิทธิ์ และเนื่องจากการหาสัมประสิทธิ์ของตัวกรองปรับตัวจะใช้วิธีการลดค่าของความผิดพลาดของเอาต์พุต จึงทำให้วิธีการนี้ได้ชื่อว่าเอาต์พุตเออร์เรอร์ (Output Error Method)

ต่อไปเราจะพัฒนาแอลเอ็มเอสอัลกอริทึมสำหรับปรับค่าสัมประสิทธิ์ของตัวกรองปรับตัวแบบโอไออาร์โดยจะกำหนดฟังก์ชันที่มีการเปลี่ยนแปลงตามเวลาดังนี้ ให้

$$A(z, n) = \sum_{i=0}^N a_i(n)z^{-i} \tag{2.22}$$

$$B(z, n) = \sum_{i=1}^M b_i(n)z^{-i} \tag{2.23}$$

ค่าเอาต์พุตของ ตัวกรองปรับตัวแบบโอไออาร์หาได้จาก

$$y(n) = \sum_{i=0}^N a_i(n)x(n-i) + \sum_{i=1}^M b_i(n)y(n-i) \tag{2.24}$$

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดใบบลึงเนื้อหาและต่อ i=1 ไปถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ เนื่องจากแอลเอ็มเอสอัลกอริทึมใช้วิธีการแบบสตอบแคสติกเกรเดียนท์ จะได้เป็น

$$\hat{\nabla}(n) = \nabla_w e^2(n) = 2e(n)\nabla_w e(n) \quad (2.25)$$

เมื่อ ∇_w เป็นการกระทำของเกรเดียนต์เทียบกับเวกเตอร์สัมประสิทธิ์

$$e(n) = d(n) - y(n) \quad (2.26)$$

ซึ่งค่าที่ได้จะเป็นค่าความผิดพลาดของเอาต์พุตและเวกเตอร์สัมประสิทธิ์นี้มาจาก

$$w(n) = [a_0(n) \ a_1(n) \ \dots \ a_N(n) \ b_1(n) \ \dots \ b_M(n)]^T \quad (2.27)$$

แทนค่า(2-17)ลงใน(2-16)และ ค่า $d(n)$ ไม่ขึ้นกับ $w(n)$ จะได้

$$\begin{aligned} \hat{\nabla}(n) &= -2e(n)\nabla_w y(n) \\ &= -2e(n) \left[\frac{\partial y(n)}{\partial a_0(n)} \quad \frac{\partial y(n)}{\partial a_1(n)} \quad \dots \quad \frac{\partial y(n)}{\partial a_N(n)} \quad \frac{\partial y(n)}{\partial b_1(n)} \quad \dots \quad \frac{\partial y(n)}{\partial b_M(n)} \right]^T \end{aligned} \quad (2.28)$$

ค่าอนุพันธ์ใน(2-19)ต้องพิจารณาเป็นพิเศษเนื่องจากค่า $y(n)$ จะขึ้นกับ ค่าของมันที่ผ่านมาแล้ว $y(n-1), y(n-2), \dots$ จาก(2-15)จะได้

$$\frac{\partial y(n)}{\partial a_i(n)} = x(n-i) + \sum_{l=1}^M b_l(n) \frac{\partial y(n-l)}{\partial a_i(n)}, \quad i = 0, 1, \dots, N \quad (2.29)$$

$$\frac{\partial y(n)}{\partial b_i(n)} = y(n-i) + \sum_{l=1}^M b_l(n) \frac{\partial y(n-l)}{\partial b_i(n)}, \quad i = 1, 2, \dots, M \quad (2.30)$$

เพื่อให้สะดวกเราจะกำหนดให้เป็น

$$\alpha_i(n) = \frac{\partial y(n)}{\partial a_i(n)}, \quad i = 0, 1, \dots, N \quad (2.31)$$

$$\beta_i(n) = \frac{\partial y(n)}{\partial b_i(n)}, \quad i = 1, 2, \dots, M \quad (2.32)$$

เราสมมุติให้สัมประสิทธิ์ $a_i(n)$ และ $b_i(n)$ เปลี่ยนแปลงตามเวลาอย่างช้าๆ จะได้

$$\frac{\partial y(n-l)}{\partial a_i(n)} \approx \frac{\partial y(n-l)}{\partial a_i(n-l)} = \alpha_i(n-l) \quad (2.33)$$

$$\frac{\partial y(n-l)}{\partial b_i(n)} \approx \frac{\partial y(n-l)}{\partial b_i(n-l)} = \beta_i(n-l) \quad (2.34)$$

สำหรับ $l = 1, 2, \dots, M$ แทนสมการ(2.33)และ(2.34)ลงใน สมการ(2.29)และ(2.30)เราจะได้สมการสำหรับการหา $\alpha_i(n)$ และ $\beta_i(n)$ ดังนี้

$$\alpha_i(n) = x(n-i) + \sum_{l=1}^M b_l(n)\alpha_i(n-l) \quad (2.35)$$

$$\beta_i(n) = y(n-i) + \sum_{l=1}^M b_l(n)\beta_i(n-l) \quad (2.36)$$

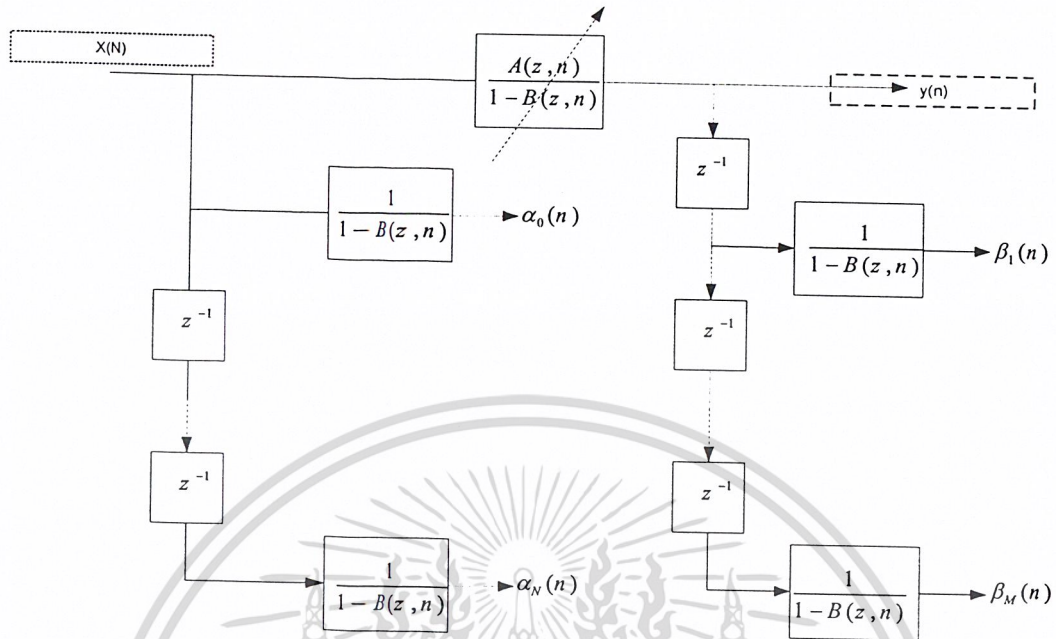
จากข้างต้น เราจะสามารถสรุปแอลเอ็มเอสอัลกอริทึมสำหรับ ตัวกรองปรับตัวแบบ ไอโออาร์ ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งนี้เอกสารนี้ได้รับการปรับปรุงแก้ไขอย่างต่อเนื่องถึงแม้ว่าเอกสารทุกครั้งที่มีการนำไปใช้

$$w(n+1) = w(n) + 2\mu e(n)\eta(n)$$

$$\eta(n) = [\alpha_0(n) \ \alpha_1(n) \ \dots \ \alpha_N(n) \ \beta_1(n) \ \dots \ \beta_M(n)]^T$$

และ $\alpha_i(n), \beta_i(n)$ หาได้จากสมการ(2.35) และ(2.36)



รูป 2.19 แสดงบล็อกโคอะแกรมการคำนวณหาค่า $\alpha_i(n)$ และ $\beta_i(n)$

จากรูปแสดงบล็อกโคอะแกรมการคำนวณหาค่า $\alpha_i(n)$ และ $\beta_i(n)$ จากสมการ(2.35) และ (2.36) จากโคอะแกรมเราจะเห็นได้ว่าการคำนวณค่าต่างๆของ $\eta(n)$ ต้องการตัวกรองป้อนกลับ ขนานกันจำนวน $M + N + 1$ ซึ่งมีทรานเฟอร์ฟังก์ชันที่เหมือนกันคือ $1/(1-B(z,n))$ แต่มีอินพุตที่แตกต่างกันของแต่ละตัว ทำให้การคำนวณแต่ละค่าจะต้องมีการคำนวณที่ซับซ้อน จึงทำการประมาณเพื่อให้ง่ายขึ้น ถ้าเราให้ทรานเฟอร์ฟังก์ชัน $1/(1-B(z,n))$ เปลี่ยนแปลงตามเวลาช้าๆ เราจะได้ค่าประมาณซึ่งในกรณีของ α_i จะประมาณได้เป็น

$$\frac{1}{1-B(z,n)} \approx \frac{1}{1-B(z,n-i)} \quad \text{สำหรับทุกค่า } i = 1, 2, \dots, N$$

ดังนั้นจะได้

$$\alpha_i(n) \approx x(n-i) + \sum_{l=1}^M b_l(n-i)\alpha_i(n-l)$$

และทำการแทนค่า $i = 0$ และ $n = n - i$ ลงในสมการ(2.35)จะได้

$$\alpha_0(n-i) = x(n-i) + \sum_{l=1}^M b_l(n-i)\alpha_0(n-i-l)$$

นำมาเปรียบเทียบกับจะพบว่า

$$\alpha_i(n) \approx \alpha_0(n-i) \quad , \quad i = 1, 2, \dots, N$$

สำหรับกรณีของ β_i จะประมาณได้เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น $\frac{1}{1-B(z,n)} \approx \frac{1}{1-B(z,n-i+1)}$ สำหรับทุกค่า $i = 1, 2, \dots, M-1$ ซึ่งจะได้จากการนำไปใช้

คังนั้นจะได้

$$\beta_i(n) \approx y(n-i) + \sum_{l=1}^M b_l(n-i+1)\beta_i(n-l)$$

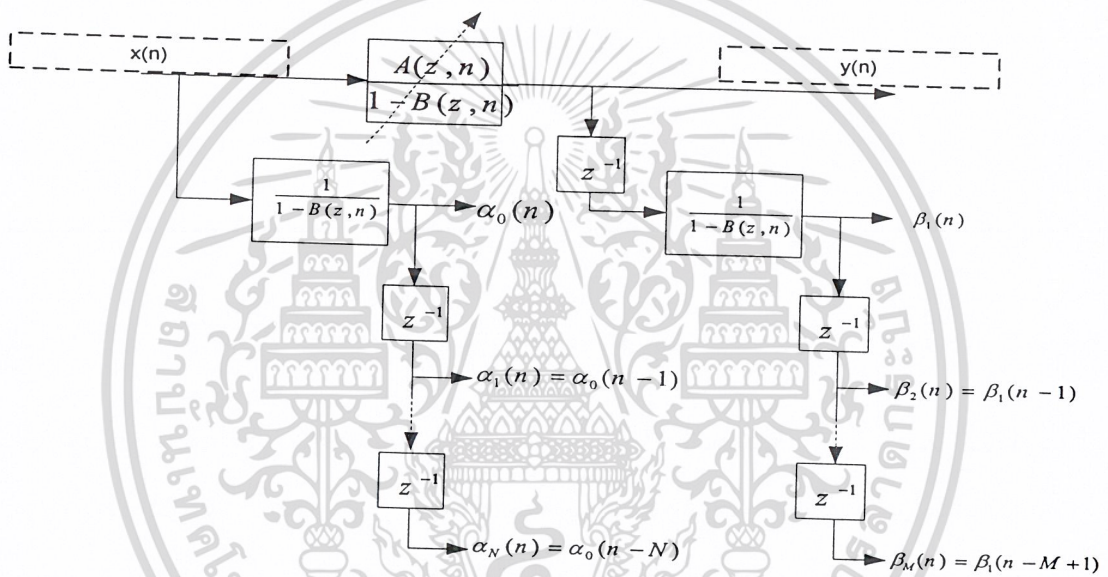
ทำการแทนค่า $i = 1$ และ $n = n - i + 1$ ในสมการ(2.27)จะได้

$$\beta_1(n) = y(n-i) + \sum_{l=1}^M b_l(n-i+1)\beta_1(n-i+1-l)$$

นำมาเปรียบเทียบกันจะพบว่า

$$\beta_i(n) \approx \beta_1(n-i+1) \quad , i = 1, 2, \dots, M-1$$

ซึ่งจะพบว่าค่า $\alpha_i(n)$ และ $\beta_i(n)$ ของตัวปัจจุบันจะหาได้จากค่าที่แล้ว

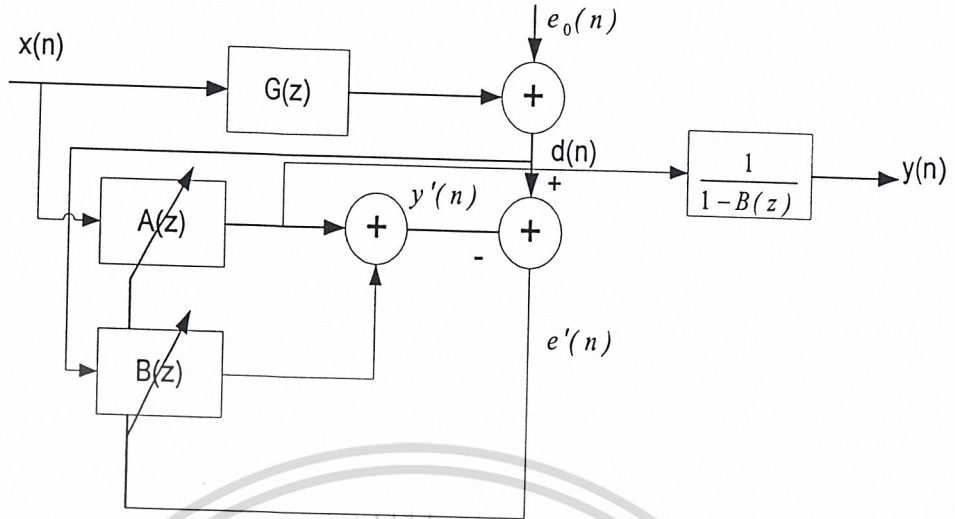


รูป 2.20 แสดงบล็อกไดอะแกรมการคำนวณหาค่า $\alpha_i(n)$ และ $\beta_i(n)$ โดยการประมาณแล้ว

2.5.2 วิธีการแบบอิกเวชัน เออร์เรอร์ (The Equation Error Method)

จากที่ได้กล่าวมาแล้วว่าวิธีการแบบเอาที่พู่เออร์เรอร์ มีปัญหาตรงที่เพอร์ฟอร์แมนซ์ฟังก์ชัน มีค่าต่ำสุดมาก ดังนั้นผลก็คือการลู่เข้าของแอลเอ็มเอสอัลกอริทึมอาจลู่เข้าสู่ค่าต่ำสุดซึ่งไม่ใช่ค่าต่ำสุดของเพอร์ฟอร์แมนซ์ฟังก์ชัน ปัญหาเหล่านี้สามารถแก้ไขได้โดยใช้วิธีการแบบอิกเวชัน เออร์เรอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.21 แสดงบล็อกไดอะแกรมของวิธีการอิคเวชัน เออร์เรอร์

จากรูปที่ 2.21 จะอธิบายบล็อกไดอะแกรมของวิธีการแบบอิคเวชัน เออร์เรอร์ ในกรณีนี้ค่าความผิดพลาดซึ่งใช้ในการปรับสัมประสิทธิ์จะหาได้จาก

$$e'(n) = d(n) - y'(n) \tag{2.37}$$

เมื่อ

$$y'(n) = \sum_{i=0}^N a_i(n)x(n-i) + \sum_{i=1}^M b_i(n)d(n-i) \tag{2.38}$$

สมการที่ได้เป็นสมการจากการแก้ไขสมการที่ (2.24) โดยแทนค่าของเอาต์พุตที่ผ่านมาแล้ว $y(n-1), y(n-2), \dots$ ในสมการ (2.24) ด้วยค่าเอาต์พุตที่ต้องการที่ผ่านมาแล้ว $d(n-1), d(n-2), \dots$ ซึ่งชื่ออิคเวชัน เออร์เรอร์ ก็ได้มาจากความแตกต่างในสมการเพื่อคำนวณค่าความแตกต่างซึ่งตรงข้ามกับวิธีการเอาต์พุต เออร์เรอร์ $e(n)$

เมื่อเลือกโครงสร้างและอันดับของฟิลเตอร์เรียบร้อยแล้ว เราจะต้องการให้ $d(n) \approx y(n)$ ในกรณีนี้ผลต่างระหว่าง $e(n)$ และ $e'(n)$ จะน้อย เมื่อตัวกรองเข้าสู่ค่าที่เหมาะสม ดังนั้นเราจะคาดว่าเพอร์ฟอร์แมนซ์ฟังก์ชันของวิธีการเอาต์พุต เออร์เรอร์ และวิธีการอิคเวชัน เออร์เรอร์จะมีค่าต่ำสุดของเพอร์ฟอร์แมนซ์ฟังก์ชันเดียวกัน ดังนั้นวิธีการอิคเวชัน เออร์เรอร์จะเป็นที่นิยมมากกว่า เพราะว่าเนื่องจากเพอร์ฟอร์แมนซ์ฟังก์ชันจะไม่มีค่าต่ำสุดในบริเวณ แต่จะมีค่าต่ำสุดเพียงค่าเดียว ผลลัพธ์ของตัวกรองดังกล่าวซึ่งไม่มีการป้อนกลับ $y'(n)$ เป็นเอาต์พุตของผลรวมเชิงเส้นของเวกเตอร์สัมประสิทธิ์ซึ่งนิยามโดย(2.27) และอินพุตเวกเตอร์

$$u'(n) = [x(n) \quad x(n-1) \quad \dots \quad x(n-N) \quad d(n-1) \quad \dots \quad d(n-N)]^T$$

ต่อไปจะแสดงให้เห็นถึงความแตกต่างของวิธีการอิคเวชัน เออร์เรอร์เทียบกับวิธีการเอาต์พุต เออร์เรอร์ โดยวิธีการของอิคเวชัน เออร์เรอร์ ในการปรับค่าสัมประสิทธิ์จะใช้

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\xi = E[e^2(n)]$$

ซึ่งฟังก์ชันดังกล่าวเป็นสมการกำลังสองของค่าสัมประสิทธิ์และเนื่องจาก $y'(n)$ ผลรวมเชิงเส้นของ $x(n)$ และ $d(n)$ ดังนั้นจะมั่นใจได้ว่าแอลเอ็มเอส อัลกอริทึมจะลู่เข้าสู่ค่าที่ดีที่สุด

เราจะสามารถเข้าใจวิธีการอิคเวชัน เออร์เรอร์ได้มากขึ้นโดยหาความสัมพันธ์ระหว่างเอาท์พุทและอิคเวชัน เออร์เรอร์ $e(n)$ และ $e'(n)$ ซึ่งสามารถหาได้โดยการแปลงแซด จากรูป 2.21 จะได้

$$D(z) = X(z)G(z) + E_0(z) \quad (2.39)$$

$$Y(z) = \frac{X(z)A(z)}{1 - B(z)} \quad (2.40)$$

เมื่อ $D(z)$, $X(z)$, $E_0(z)$ และ $Y(z)$ เป็นการแปลงแซด ของลำดับ $d(n)$, $x(n)$, $e_0(n)$ และ $y(n)$ ตามลำดับ และจาก $e(n) = d(n) - y(n)$ แทนค่าสมการ(2.39)และ(2.40)เราจะได้

$$\begin{aligned} E(z) &= D(z) - Y(z) \\ &= X(z) \left(G(z) - \frac{A(z)}{1 - B(z)} \right) + E_0(z) \end{aligned} \quad (2.41)$$

เมื่อ $E(z)$ เป็นการแปลงแซดของ $e(n)$ ในทางตรงกันข้ามจากรูปที่ 2.21 เราจะได้ $e'(n) = d(n) - y'(n)$ ด้วย $y'(n)$ ที่ได้จาก(2.38) ดังนั้นเราจะได้

$$E'(z) = D(z) - D(z)B(z) - X(z)A(z) \quad (2.42)$$

เมื่อ $E'(z)$ เป็นการแปลงแซดของ $e'(n)$ โดยการแทนค่า(2-30)ลงใน(2-33)จะได้

$$E'(z) = X(z)[(1 - B(z))G(z) - A(z)] + [1 - B(z)]E_0(z)$$

สุดท้าย เปรียบเทียบ30และ32 เราจะได้

$$E'(z) = E(z)(1 - B(z))$$

ผลที่ได้จะแสดงความสัมพันธ์ระหว่างวิธีการอิคเวชัน เออร์เรอร์กับวิธีการแบบเอาท์พุทเออร์เรอร์ ผ่านทรานเฟอร์ฟังก์ชัน $1 - B(z)$ โดยทั่วไปแล้วการลดค่าเฉลี่ยกำลังสองของเอาท์พุทและอิคเวชัน เออร์เรอร์, $e(n)$ และ $e'(n)$ ตามลำดับ จะทำให้เกิดค่าสัมประสิทธิ์ที่ต่างกัน ในกรณีที่ว่า $\xi' = E[e'^2(n)]$ ลู่เข้าสู่ค่าน้อยและ $1 - B(z)$ ไม่ได้มีค่าน้อยสำหรับทุกค่า z ทั้งหมดบนวงกลมหน่วย เป็นผลให้ค่าสัมประสิทธิ์ที่ได้มีค่าใกล้เคียงกัน แต่หาก $\xi' = E[e'^2(n)]$ และ $1 - B(z)$ มีค่าน้อย ค่าสัมประสิทธิ์ที่จะมีค่าแตกต่างกันอย่างเห็นได้ชัด

บทที่ 3

การคำนวณและการสร้าง

3.1 การออกแบบและเลือกใช้อัลกอริทึม

ในการทดลองจะทำการสร้างตัวกรองปรับตัวได้ โดยเริ่มต้นจากการออกแบบสร้างตัวกรองปรับตัวแบบเฟอไออาร์ซึ่งเป็นพื้นฐานในการศึกษาตัวกรองปรับตัว หลังจากนั้นจึงทำการออกแบบสร้างตัวกรองปรับตัวแบบไออาร์ โดยจะทำการวัดคุณสมบัติใน 2 ลักษณะ กล่าวคือ การวัดคุณสมบัติโดยการชิมูเลชั่นและการวัดคุณสมบัติของตัวกรองที่ได้จากการสร้างในระบบประมวลผลสัญญาณดิจิทัล แต่เนื่องจากข้อจำกัดทางเครื่องมือวัด ทำให้ผลตอบสนองที่ได้จากการสร้างตัวกรองในระบบประมวลผลสัญญาณบางอย่างมีอาจวัดได้ ทำให้ผลการทดลองที่เก็บได้จากการสร้างจริง อาจขาดคุณสมบัติต่างๆไปบ้าง นอกจากนั้น โครงการนี้จะทำการออกแบบการจัดการประยุกต์ใช้งานตัวกรองในสองลักษณะกล่าวคือ

1. การจัดตัวกรองเพื่อใช้ในการกำจัดสัญญาณรบกวน (Noise Cancellation)
2. การจัดตัวกรองเพื่อใช้ในการหาคุณลักษณะของระบบอื่นๆ (System Identification)

โดยในการออกแบบสร้างจะได้อธิบายดังต่อไปนี้

3.1.1 การสร้างตัวกรองปรับตัวแบบเฟอไออาร์ลำดับที่ 2

ตัวกรองปรับตัวแบบเฟอไออาร์ โดยใช้อัลกอริทึมแบบลีสต์มินสแควร์ซึ่งมีสัญญาณอินพุตเป็น $x(n)$ จะมีสมการทั่วไปเป็น

$$y(n) = \sum_{i=0}^{N-1} w_i(n)x(n-i) \quad (3.1)$$

ให้สัญญาณที่ต้องการเป็น $d(n)$ แล้วสัญญาณความผิดพลาด $e(n)$ สามารถนิยามโดย

$$e(n) = d(n) - y(n) \quad (3.2)$$

เนื่องจากแอลเอ็มเอสอัลกอริทึมจะใช้วิธีการปรับสัมประสิทธิ์แบบ สตีปเปสต์ เดสเซนท์ ซึ่งก็คือ

$$W(n+1) = W(n) - \mu \nabla e^2(n) \quad (3.3)$$

เมื่อ $W(n) = [w_0(n) \ w_1(n)]^T$ เป็นเมตริกซ์สัมประสิทธิ์ μ เป็นค่าขนาดสเกลและ ∇ เป็นเกรเดียนท์เวกเตอร์ซึ่งกำหนดโดย

$$\nabla = \left[\frac{\partial}{\partial w_0} \quad \frac{\partial}{\partial w_1} \quad \dots \quad \frac{\partial}{\partial w_{N-1}} \right]^T \quad (3.4)$$

ดังนั้นเกรเดียนท์ตัวที่ i ของเกรเดียนท์เวกเตอร์ $\nabla e^2(n)$ คือ

$$\frac{\partial e^2(n)}{\partial w_i} = 2e(n) \frac{\partial e(n)}{\partial w_i} \quad (3.5)$$

แทนสมการ (3.2) ลงในสมการ (3.5) จะได้เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้าม $\frac{\partial e^2(n)}{\partial w_i}$ ดัดแปลงเนื้อหา $\frac{\partial y(n)}{\partial w_i}$ ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการแก้ไข (3.6)

แทนค่า $y(n) = \sum_{i=0}^{N-1} w_i(n)x(n-i)$ ลงในสมการข้างต้นจะได้เป็น

$$\frac{\partial e^2(n)}{\partial w_i} = -2e(n)x(n-i) \quad (3.7)$$

ดังนั้นจากสมการข้างต้นจะพบว่า

$$\nabla e^2(n) = -2e(n)X(n) \quad (3.8)$$

เมื่อ $X(n) = [x(n) \quad x(n-1)]^T$ หลังจากทำการแทนค่าแล้วจะได้สมการปรับสัมประสิทธิ์เป็น

$$W(n+1) = W(n) + 2\mu e(n)X(n) \quad (3.9)$$

จากสมการทั้งหมดเราจะได้อัลกอริทึมโดยสรุปเป็น

เวกเตอร์สัมประสิทธิ์	$W(n) = [w_0(n) \quad w_1(n)]^T$
อินพุทเวกเตอร์	$X(n) = [x(n) \quad x(n-1)]^T$
สัญญาณที่ต้องการ	$d(n)$
ฟิลเตอร์เอาต์พุท	$y(n)$
เวกเตอร์สัมประสิทธิ์ที่ปรับแล้ว	$W(n+1)$

1. การกรองสัญญาณ

$$y(n) = W^T(n)X(n)$$

2. การประมาณค่าความผิดพลาด

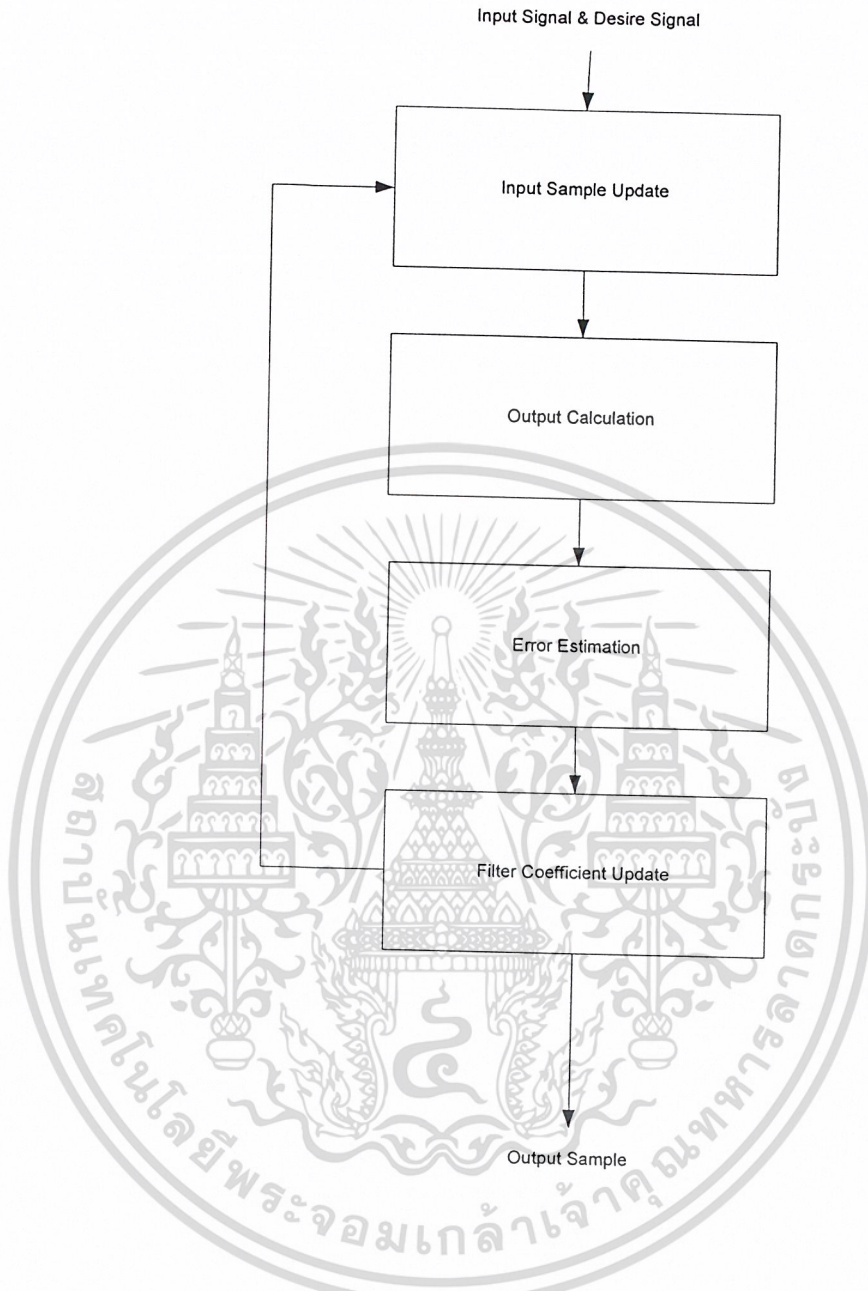
$$e(n) = d(n) - y(n)$$

3. การปรับสัมประสิทธิ์

$$W(n+1) = W(n) + 2\mu e(n)X(n)$$

อย่างไรก็ตามในทางปฏิบัติ ตัวกรองแบบเอฟไออาร์นิยมทำเป็นตัวกรองอันดับสูง (High Order) ดังนั้นในการเก็บผลการทดลองจึงเก็บผลตอบสนองของตัวกรองอันดับสูงแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 แสดงกระบวนการทำงานของตัวกรองปรับตัวแบบเอฟไออาร์

3.1.2 การสร้างตัวกรองปรับตัวได้แบบไอไออาร์ลำดับที่ 2

จากทฤษฎีตัวกรองปรับตัวได้แบบไอไออาร์โดยการใช้อัลกอริทึมแบบเอาท์พุทเออร์เรอร์-แอลเอ็มเอส ทำการหาอัลกอริทึมจากสมการ

$$y(n) = \sum_{i=0}^N a_i(n)x(n-i) + \sum_{i=1}^M b_i(n)y(n-i) \tag{3.10}$$

ในกรณีนี้เลือกใช้ค่า M และ N เท่ากับสองเนื่องจากเลือกใช้ตัวกรองอันดับสอง ทำให้ได้สมการของตัวกรองเป็น

$$y(n) = a_0(n)x(n) + a_1(n)x(n-1) + a_2(n)x(n-2) + b_1(n)y(n-1) + b_2(n)y(n-2) \tag{3.11}$$

โดยการให้ค่าสัมประสิทธิ์ตัวกรองดังกล่าวซึ่งเปลี่ยนแปลงตามเวลารวมอยู่ในรูปเวกเตอร์

$$W(n) = [a_0(n) \ a_1(n) \ a_2(n) \ b_1(n) \ b_2(n)]^T \quad (3.12)$$

และให้อินพุตของตัวกรองรวมอยู่ในรูปเวกเตอร์

$$U(n) = [x(n) \ x(n-1) \ x(n-2) \ y(n-1) \ y(n-2)]^T \quad (3.13)$$

จะจัดสมการใหม่ได้เป็น

$$y(n) = W^T(n) \cdot U(n) \quad (3.14)$$

ใช้การหาค่าความผิดพลาดโดยเทียบกับสัญญาณอ้างอิง $d(n)$

$$e(n) = d(n) - y(n) \quad (3.15)$$

ให้ค่าความผิดพลาดเฉลี่ยกำลังสองมีค่าเป็น

$$\xi = E[e^2(n)] \quad (3.16)$$

เนื่องจากแอลเอ็มเอสอัลกอริทึมใช้วิธีการปรับค่าสัมประสิทธิ์ โดยการหาเกรเดียนท์ของค่าความผิดพลาดกำลังสองเทียบกับค่าสัมประสิทธิ์ โดยให้ $\nabla(n)$ แทนค่าเกรเดียนท์ของค่าความผิดพลาดกำลังสองเทียบกับค่าสัมประสิทธิ์จะได้เป็น

$$\nabla(n) = \nabla_w \xi = 2e(n) \nabla_w e(n) = -2e(n) \nabla_w y(n) \quad (3.17)$$

ดังนั้นจะได้ว่า

$$\nabla(n) = -2e(n) \left[\frac{\partial y(n)}{\partial a_0(n)} \quad \frac{\partial y(n)}{\partial a_1(n)} \quad \frac{\partial y(n)}{\partial a_2(n)} \quad \frac{\partial y(n)}{\partial b_1(n)} \quad \frac{\partial y(n)}{\partial b_2(n)} \right]^T \quad (3.18)$$

ทำการหาค่า $\frac{\partial y(n)}{\partial a_i(n)}$ และ $\frac{\partial y(n)}{\partial b_i(n)}$ ที่ค่า i ต่าง ๆ โดยการทำอนุพันธ์สมการหลักจะได้เป็น

$$\frac{\partial y(n)}{\partial a_i(n)} = x(n-i) + \sum_{l=1}^M b_l(n) \frac{\partial y(n-l)}{\partial a_i(n)} \quad (3.19)$$

$$\frac{\partial y(n)}{\partial b_i(n)} = y(n-i) + \sum_{l=1}^M b_l(n) \frac{\partial y(n-l)}{\partial b_i(n)} \quad (3.20)$$

โดยการให้

$$\alpha_i(n) = \frac{\partial y(n)}{\partial a_i(n)} \quad , \quad i = 0, 1, 2, \dots, N \quad (3.21)$$

$$\beta_i(n) = \frac{\partial y(n)}{\partial b_i(n)} \quad , \quad i = 0, 1, 2, \dots, M \quad (3.22)$$

โดยการสมมติให้ค่าสัมประสิทธิ์เปลี่ยนแปลงช้าเมื่อเทียบกับเวลาเราสามารถประมาณใหม่ได้เป็น

$$\frac{\partial y(n-1)}{\partial a_i(n)} \approx \frac{\partial y(n-1)}{\partial a_i(n-1)} \approx \alpha_i(n-1) \quad (3.23)$$

$$\frac{\partial y(n-1)}{\partial b_i(n)} \approx \frac{\partial y(n-1)}{\partial b_i(n-1)} \approx \beta_i(n-1) \quad (3.24)$$

เมื่อ $i = 0, 1, 2, \dots, M$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นจะได้ว่า

$$\alpha_i(n) = x(n-i) + \sum_{l=1}^M b_l(n) \alpha_i(n-l) \quad (3.25)$$

$$\beta_i(n) = y(n-i) + \sum_{l=1}^M b_l(n) \beta_i(n-l) \quad (3.26)$$

ซึ่งในกรณีนี้การคำนวณหาแต่ละค่าจะต้องมีการคำนวณที่ซับซ้อนจึงทำการประมาณเพื่อให้ง่ายขึ้น (Simplified Algorithm) โดยคิดว่าทรานสเฟอร์ฟังก์ชัน $1/(1-B(z,n))$ มีการเปลี่ยนแปลงช้าเมื่อเทียบกับเวลา ดังนั้น

กรณีของ α_i จะประมาณได้เป็น

$$\frac{1}{1-B(z,n)} \approx \frac{1}{1-B(z,n-i)} \quad \text{สำหรับทุกค่า } i = 1, 2, \dots, N \quad (3.27)$$

ดังนั้นจะได้

$$\alpha_i(n) \approx x(n-i) + \sum_{l=1}^M b_l(n-i) \alpha_i(n-l) \quad (3.28)$$

และทำการแทนค่า $i = 0$ และ $n = n-i$ ลงในสมการ(3.25) จะได้

$$\alpha_0(n-i) = x(n-i) + \sum_{l=1}^M b_l(n-i) \alpha_0(n-i-l) \quad (3.29)$$

นำมาเปรียบเทียบกับจะพบว่า

$$\alpha_i(n) \approx \alpha_0(n-i) \quad , i = 1, 2, \dots, N \quad (3.30)$$

สำหรับกรณีของ β_i จะประมาณได้เป็น

$$\frac{1}{1-B(z,n)} \approx \frac{1}{1-B(z,n-i+1)} \quad \text{สำหรับทุกค่า } i = 1, 2, \dots, M-1 \quad (3.31)$$

ดังนั้นจะได้

$$\beta_i(n) \approx y(n-i) + \sum_{l=1}^M b_l(n-i+1) \beta_i(n-l) \quad (3.32)$$

ทำการแทนค่า $i = 1$ และ $n = n-i+1$ ในสมการ (3.26) จะได้

$$\beta_1(n) = y(n-i) + \sum_{l=1}^M b_l(n-i+1) \beta_1(n-i+1-l) \quad (3.33)$$

นำมาเปรียบเทียบกับจะพบว่า

$$\beta_i(n) \approx \beta_1(n-i+1) \quad , i = 1, 2, \dots, M-1 \quad (3.34)$$

สำหรับกระบวนการปรับสัมประสิทธิ์ตัวกรองนั้นเป็นไปตามสมการของ

$$W(n+1) = W(n) + 2\mu e(n)\eta(n) \quad (3.35)$$

โดย $W(n)$ คือเวกเตอร์สัมประสิทธิ์(Coefficient Vector)

μ คืออัตราการลู่เข้า(Rate of convergence)หรือขนาดความละเอียด(Stepsize)

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนลิขสิทธิ์เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น $\eta(n)$ คือเวกเตอร์ปรับสัมประสิทธิ์ซึ่งนิยามโดย จนถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\eta(n) = [\alpha_0(n) \quad \alpha_1(n) \quad \dots \quad \alpha_N(n) \quad \beta_1(n) \quad \dots \quad \beta_{M(n)}]^T$$

ดังนั้นในกรณีนี้จะได้

$$\eta(n) = [\alpha_0(n) \quad \alpha_1(n) \quad \alpha_2(n) \quad \beta_1(n) \quad \beta_2(n)]^T \quad (3.36)$$

ซึ่งเท่ากับ

$$\eta(n) = [\alpha_0(n) \quad \alpha_0(n-1) \quad \alpha_0(n-2) \quad \beta_1(n) \quad \beta_1(n-1)]^T \quad (3.37)$$

อัลกอริทึมโดยสรุปจะเป็น

อินพุทเวกเตอร์

$$U(n) = [x(n) \quad x(n-1) \quad x(n-2) \quad y(n-1) \quad y(n-2)]^T$$

เวกเตอร์สัมประสิทธิ์

$$W(n) = [a_0(n) \quad a_1(n) \quad a_2(n) \quad b_1(n) \quad b_2(n)]^T$$

เวกเตอร์ปรับสัมประสิทธิ์

$$\eta(n) = [\alpha_0(n) \quad \alpha_0(n-1) \quad \alpha_0(n-2) \quad \beta_1(n) \quad \beta_1(n-1)]^T$$

สัญญาณอ้างอิง

$$d(n)$$

สัญญาณเอาต์พุท

$$y(n)$$

1. การกรองสัญญาณ

$$y(n) = W^T(n) \cdot U(n)$$

2. การหาค่าผิดพลาด

$$e(n) = d(n) - y(n)$$

3. การหาค่าเวกเตอร์ปรับสัมประสิทธิ์

$$\alpha_0(n) = x(n) + \sum_{l=1}^M b_l(n) \alpha_0(n-l)$$

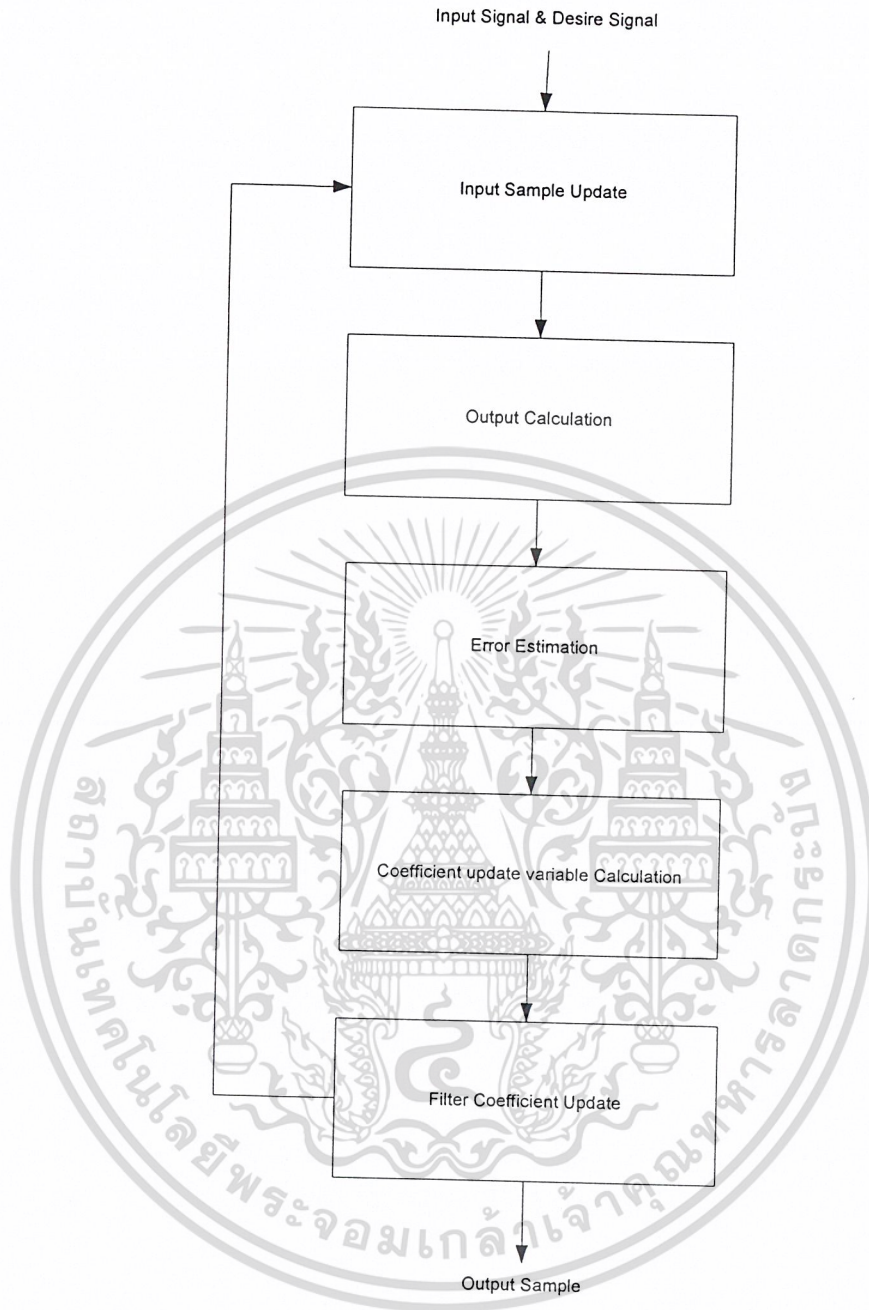
$$\beta_1(n) = y(n) + \sum_{l=1}^M b_l(n) \beta_1(n-l)$$

$$\eta(n) = [\alpha_0(n) \quad \alpha_0(n-1) \quad \alpha_0(n-2) \quad \beta_1(n) \quad \beta_1(n-1)]^T$$

4. การปรับสัมประสิทธิ์

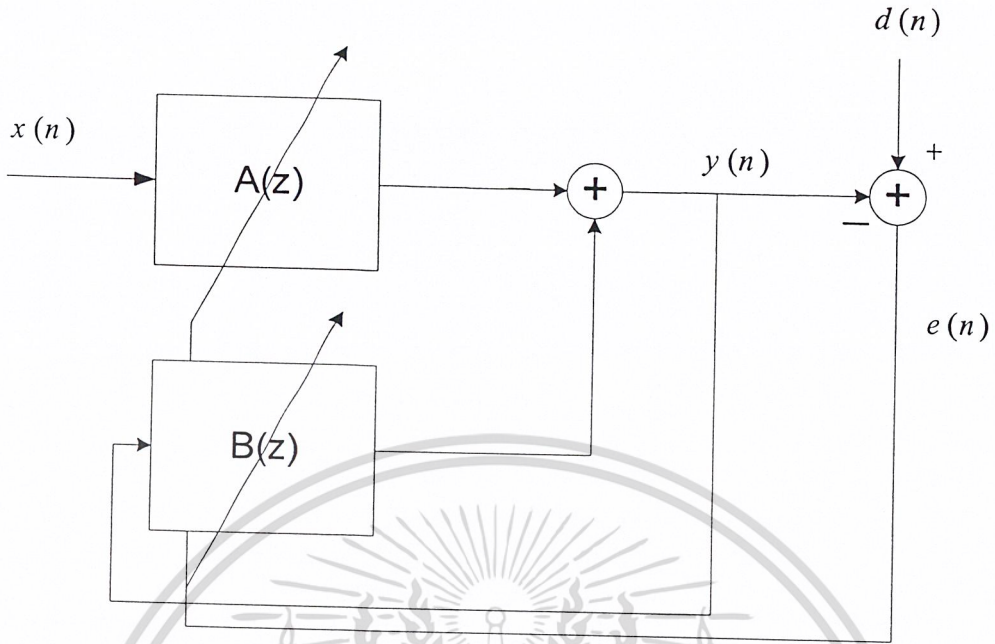
$$W(n+1) = W(n) + 2\mu e(n) \eta(n)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

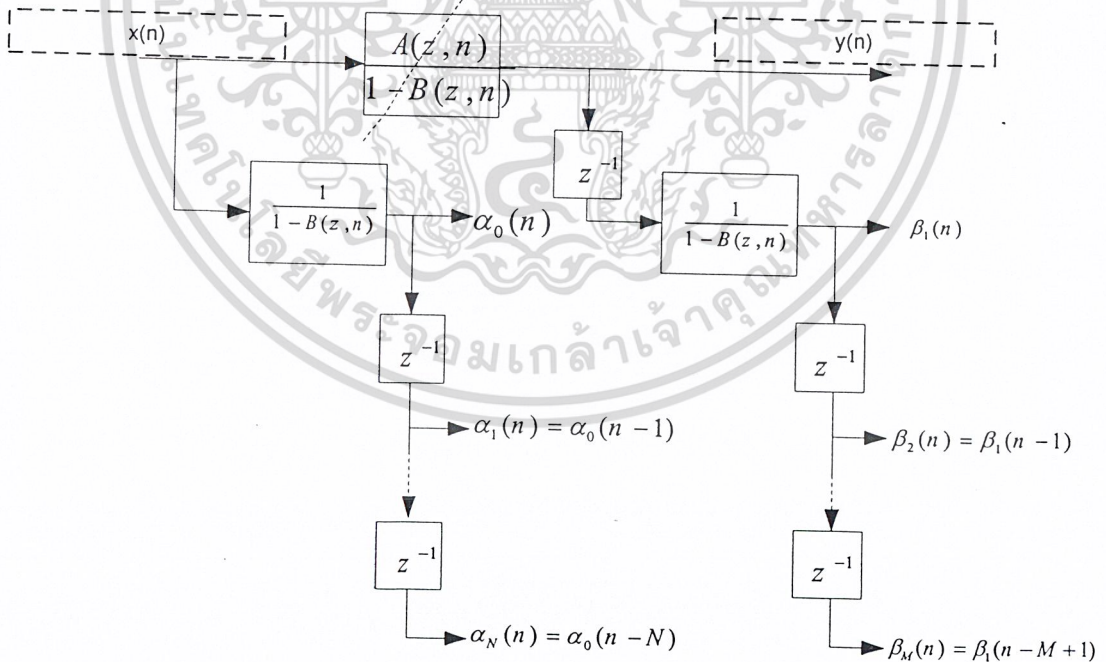


รูปที่ 3.2 แสดงกระบวนการทำงานของตัวกรองปรับตัวแบบไอไออาร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แสดง โครงสร้างตัวกรองปรับตัวได้ที่ออกแบบ

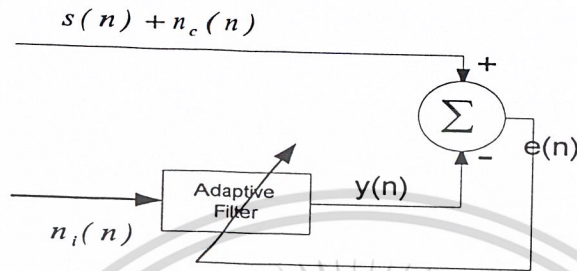


รูปที่ 3.4 แสดง โครงสร้างตัวกรองปรับตัวได้ในส่วนการหาค่าเวกเตอร์ปรับสัมประสิทธิ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 การจัดตัวกรองเพื่อใช้ในการกำจัดสัญญาณรบกวน (Noise Cancellation)

การนำตัวกรองปรับตัวได้มาประยุกต์ใช้ในการกำจัดสัญญาณรบกวนนั้น จำเป็นต้องใช้สัญญาณรบกวนที่มีความสัมพันธ์กับสัญญาณรบกวนที่ป้อนมาในสัญญาณที่ต้องการ ในการกำจัดสัญญาณรบกวน โดยการป้อนสัญญาณและ โครงสร้างของระบบกำจัดสัญญาณรบกวนเป็นไปตามรูป



รูปที่ 3.5 แสดงโครงสร้างระบบกำจัดสัญญาณรบกวนโดยใช้ตัวกรองปรับตัวได้

จากรูปป้อนสัญญาณรบกวน $n_i(n)$ ซึ่งมีความสัมพันธ์กับสัญญาณรบกวน $n_c(n)$ ให้แก่ระบบจะได้เอาที่พหุของตัวกรองปรับตัว $y(n)$ แล้วจึงนำไปหักล้างกับสัญญาณรบกวน $n_c(n)$ ซึ่งปะปนอยู่กับสัญญาณที่ต้องการ $s(n)$ ทำให้ได้สัญญาณที่ต้องการ $e(n)$ ซึ่งเป็นสัญญาณที่มีสัญญาณรบกวนน้อยลง และสัญญาณนี้จะถูกป้อนให้กับตัวกรองเพื่อใช้ในการเปรียบเทียบในการปรับค่าสัมประสิทธิ์อีกทีหนึ่ง

ในการออกแบบเลือกใช้ตัวกรองลำดับสองดังนั้นค่า $M, N = 2$ และออกแบบเช่นเดียวกับตัวกรองปรับตัว อัลกอริทึมโดยสรุปจะเป็นดังนี้

อินพุทเวกเตอร์ $U(n) = [n_i(n) \quad n_i(n-1) \quad n_i(n-2) \quad y(n-1) \quad y(n-2)]^T$

เวกเตอร์สัมประสิทธิ์ $W(n) = [a_0(n) \quad a_1(n) \quad a_2(n) \quad b_1(n) \quad b_2(n)]^T$

เวกเตอร์ปรับสัมประสิทธิ์ $\eta(n) = [\alpha_0(n) \quad \alpha_0(n-1) \quad \alpha_0(n-2) \quad \beta_1(n) \quad \beta_1(n-1)]^T$

สัญญาณที่ต้องการซึ่งมีสัญญาณรบกวนปะปน $s(n) + n_c(n)$

สัญญาณเอาที่พหุที่กำจัดสัญญาณรบกวนแล้ว $e(n)$

สัญญาณรบกวนที่ป้อนให้แก่ระบบ $n_i(n)$

1. การกรองสัญญาณ

$$y(n) = W^T(n) \cdot U(n)$$

2. การหาค่าผิดพลาดและสัญญาณเอาที่พหุที่กำจัดสัญญาณรบกวนแล้ว

$$e(n) = s(n) + n_c(n) - y(n)$$

3. การหาค่าเวกเตอร์ปรับสัมประสิทธิ์

$$\alpha_0(n) = x(n) + \sum_{l=1}^M b_l(n) \alpha_0(n-l)$$

$$\beta_1(n) = y(n) + \sum_{l=1}^M b_l(n) \beta_1(n-l)$$

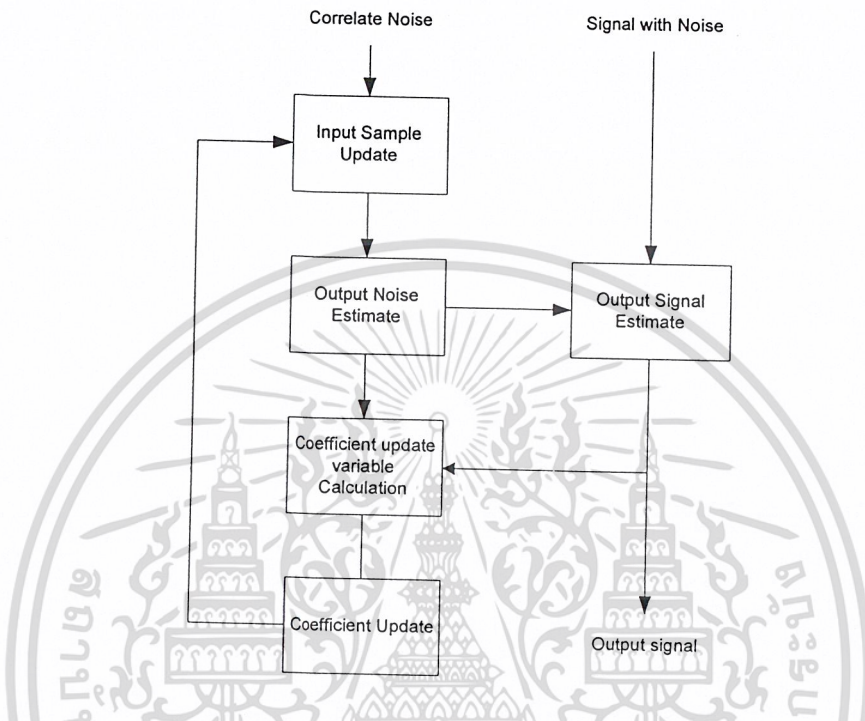
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น ยกเว้นที่มิได้ระบุเป็นอย่างอื่น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\eta(n) = [\alpha_0(n) \quad \alpha_0(n-1) \quad \alpha_0(n-2) \quad \beta_1(n) \quad \beta_1(n-1)]^T$$

4. การปรับสัมประสิทธิ์

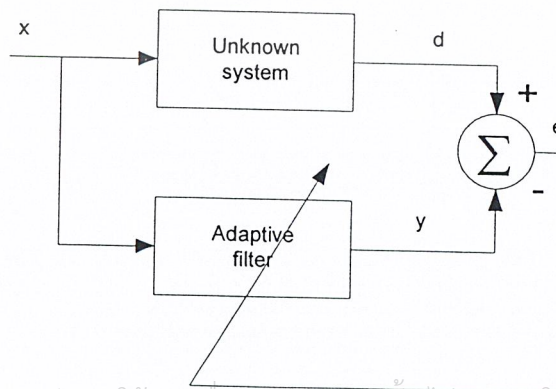
$$W(n+1) = W(n) + 2\mu e(n)\eta(n)$$



รูปที่ 3.6 แสดงกระบวนการทำงานของระบบกำจัดสัญญาณรบกวน

3.1.4 การจัดตัวกรองเพื่อใช้ในการหาคุณลักษณะของระบบอื่น (System Identification)

ในการจัดตัวกรองเพื่อใช้ในการหาคุณลักษณะของระบบอื่นนั้นจะทำการป้อนอินพุทให้แก่ระบบที่เราต้องการหาและตัวกรองปรับตัว แล้วจึงนำผลตอบสนองของระบบและตัวกรองมาทำการหาค่าความผิดพลาดและส่งค่าความผิดพลาดกลับไปยังตัวกรองเพื่อใช้ในการเปรียบเทียบในการปรับค่าสัมประสิทธิ์อีกทีหนึ่ง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.7 การจัดตัวกรองเพื่อใช้ในการหาคุณลักษณะของระบบอื่น

ทำการเลือกใช้ตัวกรองอันดับสองดังนั้นค่า $M, N = 2$ และออกแบบเช่นเดียวกับตัวกรองปรับตัว อัลกอริทึมโดยสรุปจะเป็นดังนี้

อินพุทเวกเตอร์ $U(n) = [x(n) \ x(n-1) \ x(n-2) \ y(n-1) \ y(n-2)]^T$

เวกเตอร์สัมประสิทธิ์ $W(n) = [a_0(n) \ a_1(n) \ a_2(n) \ b_1(n) \ b_2(n)]^T$

เวกเตอร์ปรับสัมประสิทธิ์ $\eta(n) = [\alpha_0(n) \ \alpha_0(n-1) \ \alpha_0(n-2) \ \beta_1(n) \ \beta_1(n-1)]^T$

ผลตอบสนองจากระบบที่ต้องการหา $d(n)$

สัญญาณเอาต์พุท $y(n)$

1. การกรองสัญญาณ

$$y(n) = W^T(n) \cdot U(n)$$

2. การหาค่าผิดพลาด

$$e(n) = d(n) - y(n)$$

3. การหาค่าเวกเตอร์ปรับสัมประสิทธิ์

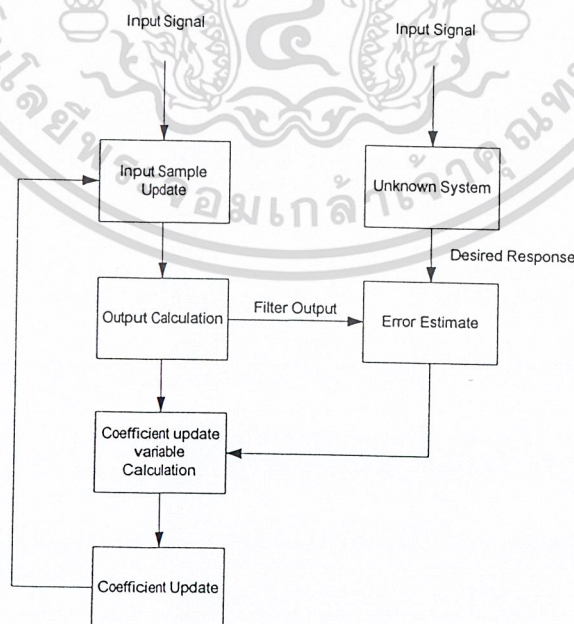
$$\alpha_0(n) = x(n) + \sum_{l=1}^M b_l(n) \alpha_0(n-l)$$

$$\beta_1(n) = y(n) + \sum_{l=1}^M b_l(n) \beta_1(n-l)$$

$$\eta(n) = [\alpha_0(n) \ \alpha_0(n-1) \ \alpha_0(n-2) \ \beta_1(n) \ \beta_1(n-1)]^T$$

4. การปรับสัมประสิทธิ์

$$W(n+1) = W(n) + 2\mu e(n)\eta(n)$$



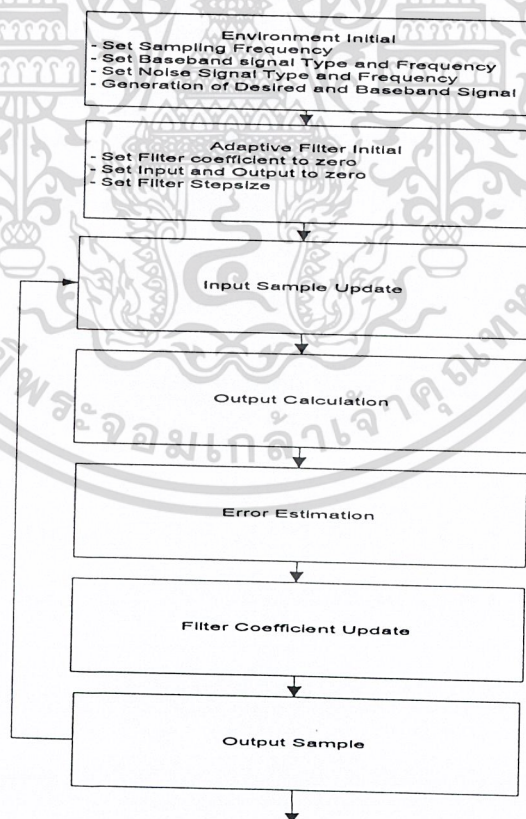
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 3.8 แสดงกระบวนการทำงานของการหาคุณลักษณะระบบอื่นโดยใช้ตัวกรอง
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การสร้างตัวกรองโดยการซิมูเลชันด้วยโปรแกรมแมทแล็บ

หลังจากการออกแบบตัวกรองและระบบต่างๆที่ต้องการแล้วนั้นจะทำการซิมูเลชันด้วยโปรแกรมแมทแล็บ ซึ่งโดยทั่วไปในการซิมูเลชันของระบบประมวลผลสัญญาณที่ไม่เปลี่ยนแปลงตามเวลา นิยมใช้ทูลบ็อกซ์และคำสั่งต่างๆซึ่งมีในตัวโปรแกรมในการหาผลตอบสนองได้โดยตรง แต่เนื่องจากตัวกรองปรับตัวเป็นระบบเปลี่ยนแปลงตามเวลา กล่าวคือ ค่าสัมประสิทธิ์ของตัวกรองจะมีการเปลี่ยนแปลงทุกครั้งที่มีค่าการผิดพลาดเกิดขึ้น ทำให้ในการซิมูเลชันต้องทำการเขียนโปรแกรมหาผลตอบสนองในแกนเวลา (Time Domain Response) ซึ่งจะช่วยให้อ่านผลตอบสนองในแกนเวลา ผลตอบสนองชั่วคราว (Transient Response) ตลอดจนค่าความผิดพลาดเทียบกับเวลาได้ ต่อไปนี้จะอธิบายแนวทางในการเขียนโปรแกรมต่างๆที่ใช้ในการซิมูเลชัน

3.2.1 การเขียนโปรแกรมซิมูเลชันสำหรับตัวกรองปรับตัวแบบเอฟไออาร์

การเขียนโปรแกรมซิมูเลชันของตัวกรองปรับตัวแบบเอฟไออาร์เพื่อทำการหาผลตอบสนองในแกนเวลา ผลตอบสนองชั่วคราว ค่าความผิดพลาดเทียบกับเวลาและเปรียบเทียบผลของอัตราการเรียนรู้ที่มีผลต่อผลตอบสนอง จะทำการเขียนโปรแกรมโดยใช้โปรแกรมแมทแล็บ เนื่องจากโครงสร้างทางภาษาไม่ซับซ้อนนักและมีฟังก์ชันต่างๆในการทำงานที่ครบครัน สำหรับการไหลของโปรแกรมจะเป็นไปดังรูป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ รูปที่ 3.9 แสดงการไหลของโปรแกรมซิมูเลชันสำหรับตัวกรองปรับตัวแบบเอฟไออาร์

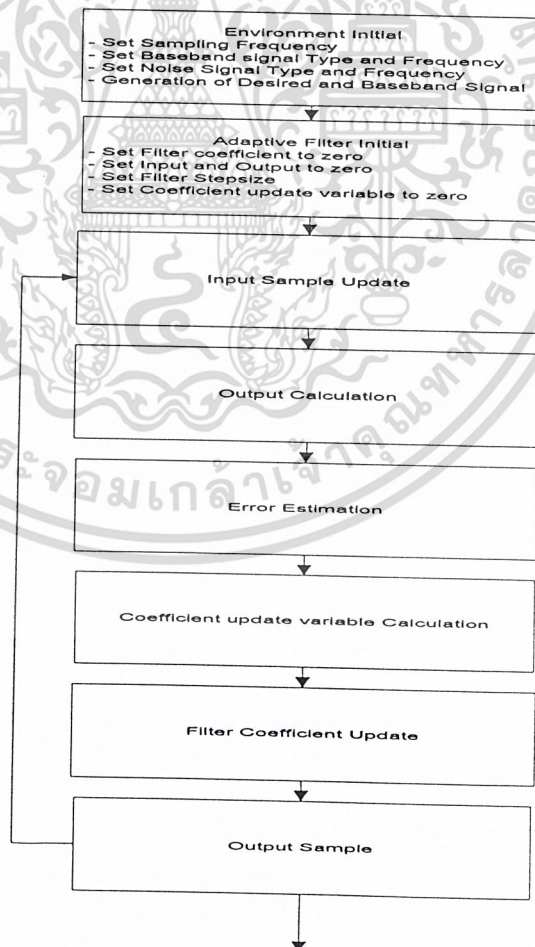
ส่วนประกอบหลักของ โปรแกรมซิมูเลชันมีดังต่อไปนี้

1. ส่วนสร้างสัญญาณอินพุต สัญญาณรบกวน
2. ส่วนทำหน้าที่กรองสัญญาณ
3. ส่วนหาค่าความผิดพลาด
4. ส่วนปรับสัมประสิทธิ์
5. ส่วนพล็อตกราฟ

สำหรับโปรแกรมที่เขียนสำเร็จแล้วแสดงไว้ในภาพผนวก ก.

3.2.2 การเขียนโปรแกรมซิมูเลชันสำหรับตัวกรองปรับตัวแบบไอโออาร์

การเขียนโปรแกรมซิมูเลชันของตัวกรองปรับตัว จะเพื่อทำการหาผลตอบสนองในแกนเวลา ผลตอบสนองชั่วคราว ค่าความผิดพลาดเทียบกับเวลาและเปรียบเทียบผลของอัตราการลู่เข้าที่มีผลต่อผลตอบสนอง โดยจะทำการเขียนโปรแกรมโดยใช้โปรแกรมแมทแล็บ เนื่องจากโครงสร้างทางภาษาไม่ซับซ้อนนักและมีฟังก์ชันต่างๆในการทำงานที่ครบครัน สำหรับการไหลของโปรแกรมจะเป็นไปดังรูป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ห้ามสืบ อีกรทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
รูปที่ 3.10 แสดงการ ไหลของโปรแกรมซิมูเลชันสำหรับตัวกรองปรับตัวแบบไอโออาร์

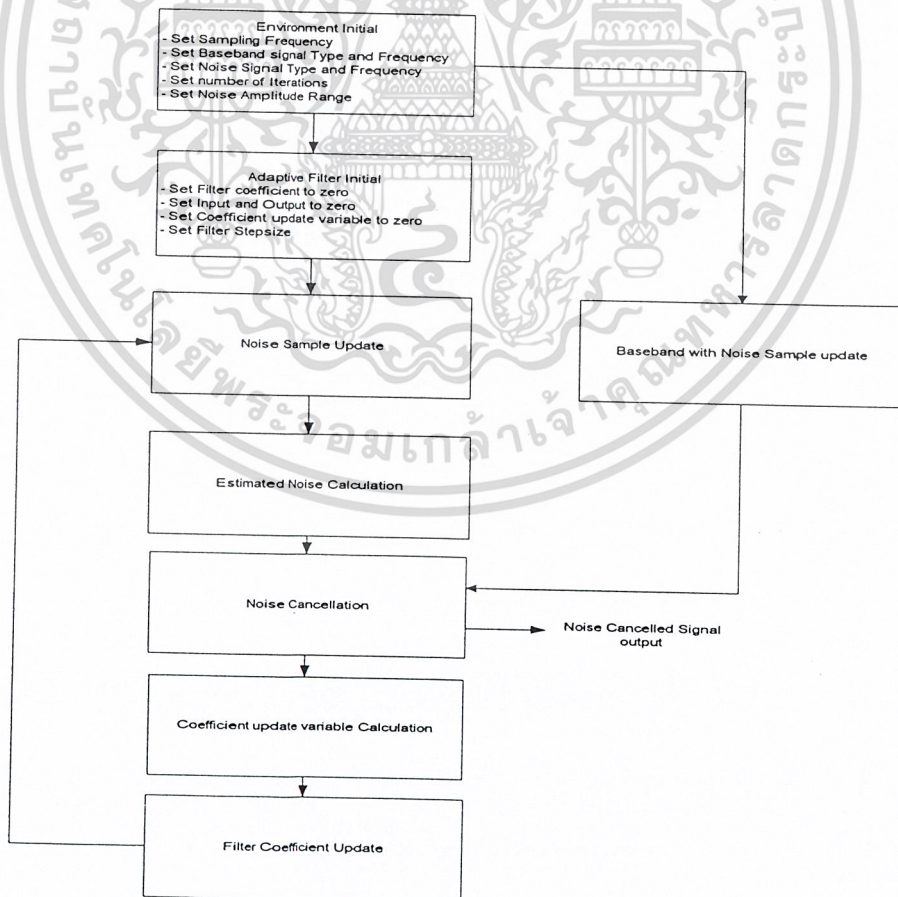
ส่วนประกอบหลักของโปรแกรมซิมูเลชันมีดังต่อไปนี้

1. ส่วนสร้างสัญญาณอินพุต สัญญาณรบกวน
2. ส่วนทำหน้าที่กรองสัญญาณ
3. ส่วนหาค่าความผิดพลาด
4. ส่วนปรับสัมประสิทธิ์
5. ส่วนพล็อตกราฟ

สำหรับโปรแกรมที่เขียนสำเร็จแล้วแสดงไว้ในภาพผนวก ก

3.2.3 การเขียนโปรแกรมซิมูเลชันสำหรับระบบกำจัดสัญญาณรบกวน

การเขียนโปรแกรมซิมูเลชันของระบบกำจัดสัญญาณรบกวนเพื่อทำการหาผลตอบสนองในแกนเวลา ผลตอบสนองชั่วคราว ค่าความผิดพลาดเทียบกับเวลาและเปรียบเทียบผลของอัตราการดูเข้าที่มีผลต่อผลตอบสนอง จะทำการเขียนโปรแกรมโดยใช้โปรแกรมแมทแล็บ เนื่องจากโครงสร้างการป้อนสัญญาณของระบบกำจัดสัญญาณรบกวนจะแตกต่างจากตัวกรองปรับตัว ทำให้การป้อนสัญญาณไม่เหมือนกัน สำหรับการไหลของโปรแกรมจะเป็นไปดังรูป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดรูปที่ 3.11 แสดงการไหลของโปรแกรมซิมูเลชันสำหรับระบบกำจัดสัญญาณรบกวน

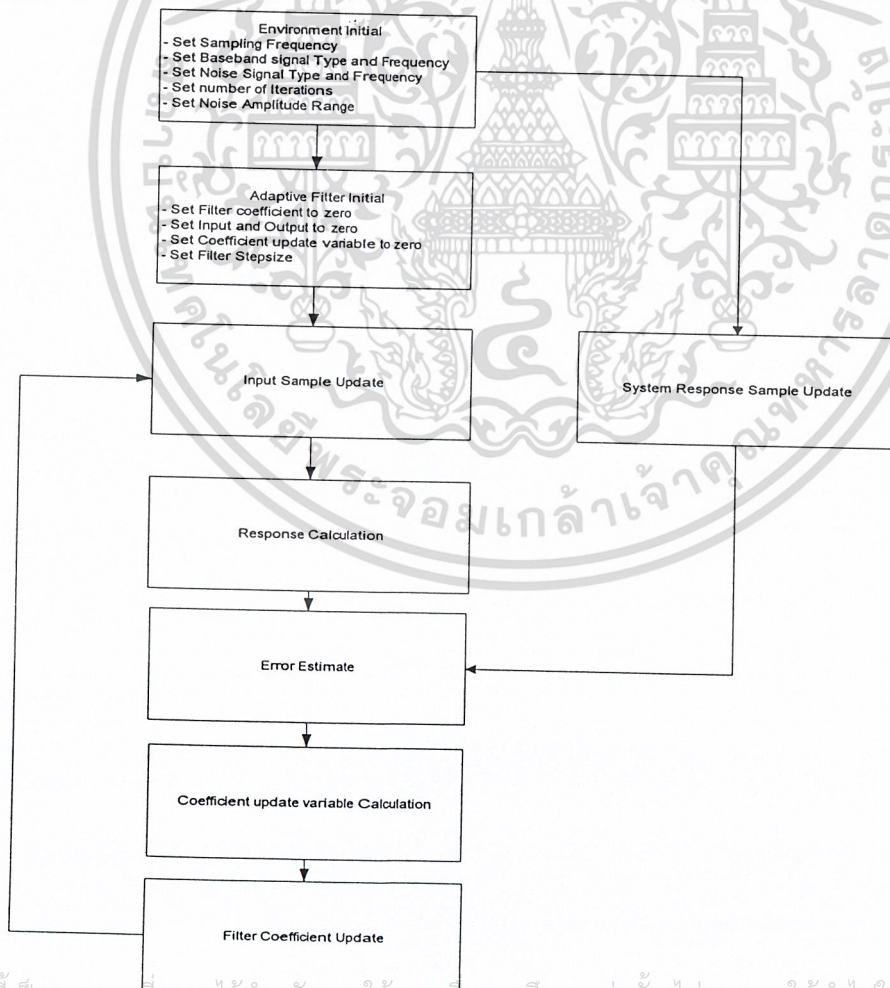
ส่วนประกอบหลักของ โปรแกรมซิมูเลชันมีดังต่อไปนี้

1. ส่วนสร้างสัญญาณอินพุต สัญญาณรบกวน
2. ส่วนทำหน้าที่กรองสัญญาณ
3. ส่วนหาค่าความผิดพลาด
4. ส่วนปรับสัมประสิทธิ์
5. ส่วนพล็อตกราฟ

สำหรับโปรแกรมที่เขียนสำเร็จแล้วแสดงไว้ในภาคผนวก ก.

3.2.4 การเขียนโปรแกรมซิมูเลชันสำหรับการหาคูณลักษณะระบบอื่นโดยใช้ตัวกรอง

การเขียนโปรแกรมซิมูเลชันเพื่อทำการหาคูณลักษณะระบบอื่นโดยใช้ตัวกรอง มีโครงสร้างการป้อนสัญญาณดังรูปที่ 3.7 สัญญาณป้อนเข้าระบบที่เราต้องการหาและตัวกรองปรับตัวพร้อมกันแล้วจึงนำผลตอบสนองของระบบและตัวกรองมาทำการหาค่าความผิดพลาดและส่งค่าความผิดพลาดกลับไปยังตัวกรองเพื่อใช้ในการเปรียบเทียบในการปรับค่าสัมประสิทธิ์ สำหรับการไหลของโปรแกรมแสดงดังรูป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหวนเลขต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.12 แสดงการ ไพล์ของ โปรแกรมซิมูเลชันเพื่อทำการหาคูณลักษณะระบบอื่นโดยใช้ตัวกรอง

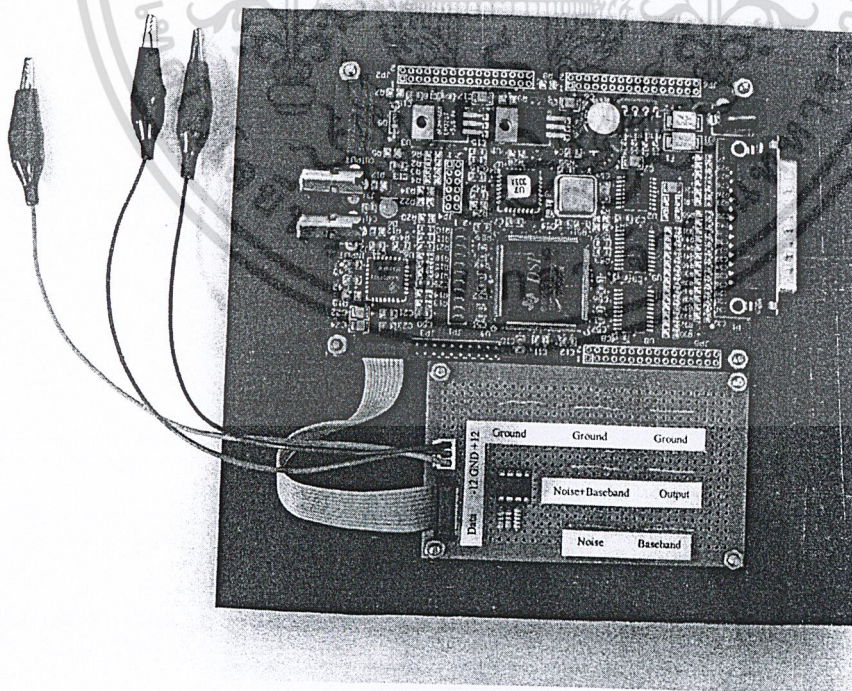
ส่วนประกอบหลักของโปรแกรมซิมูเลชันมีดังต่อไปนี้

1. ส่วนสร้างสัญญาณอินพุต สัญญาณรบกวน
2. ส่วนทำหน้าที่กรองสัญญาณ
3. ส่วนหาค่าความผิดพลาด
4. ส่วนปรับสัมประสิทธิ์
5. ส่วนพล็อตกราฟ

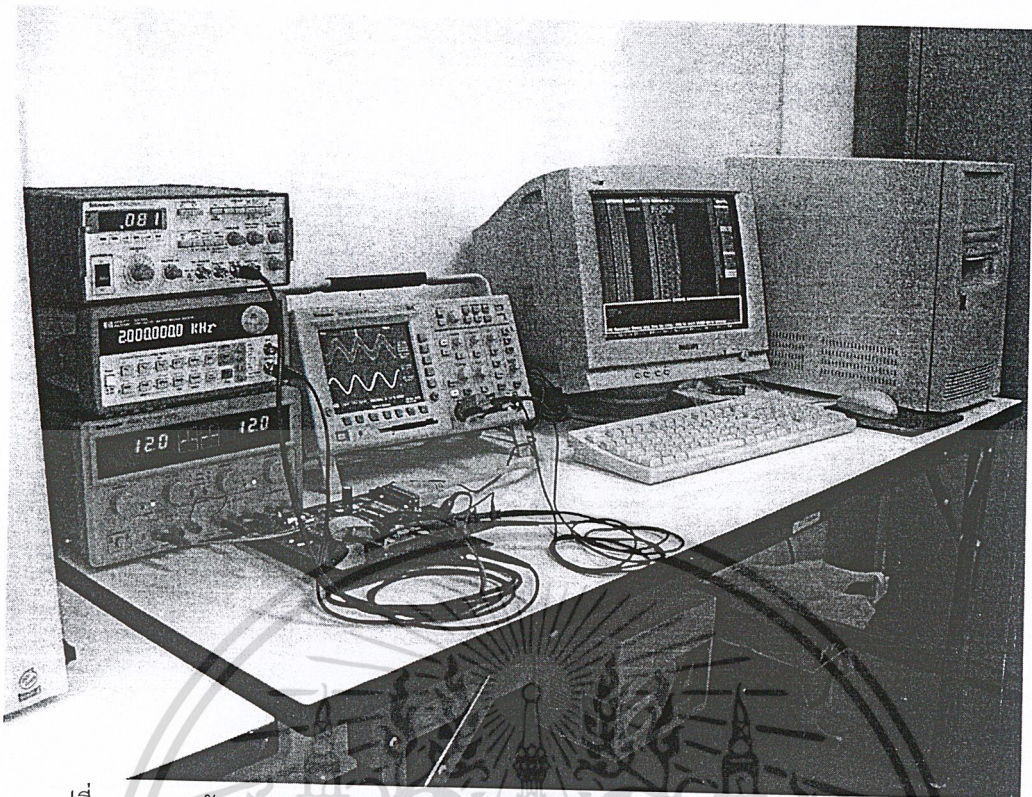
สำหรับโปรแกรมที่เขียนสำเร็จแล้วแสดงไว้ในภาคผนวก ก.

3.3 การสร้างตัวกรองปรับตัวในระบบประมวลผลสัญญาณดิจิทัล

ในการสร้างตัวกรองในระบบประมวลผลสัญญาณดิจิทัล ใช้บอร์ดประมวลผลสัญญาณดิจิทัล TMS320C31 DSK โดยจะทำการสร้างเฉพาะตัวกรองปรับตัวแบบเอฟโออาร์ ไอโออาร์และตัวกำจัดสัญญาณรบกวนเท่านั้น ซึ่งจะทำการจ่ายสัญญาณเบสแบนด์ สัญญาณที่ต้องการ ให้แก่บอร์ดทางช่องสัญญาณเข้าหลักและช่องสัญญาณเข้ารองตามลำดับ สำหรับการรวมสัญญาณเบสแบนด์กับสัญญาณรบกวนจะใช้วงจรขยายผลรวมสัญญาณ(Summing Amplifier)ในการรวมสัญญาณ ผลที่ได้วัด ณ ช่องเอาต์พุต สำหรับการโปรแกรมให้แก่บอร์ดจะเลือกใช้ภาษาแอสเซมบลี เนื่องจากใช้เวลาในการประมวลผลคำสั่งต่ำและยังเข้าถึงรีจิสเตอร์ต่างๆได้โดยง่าย



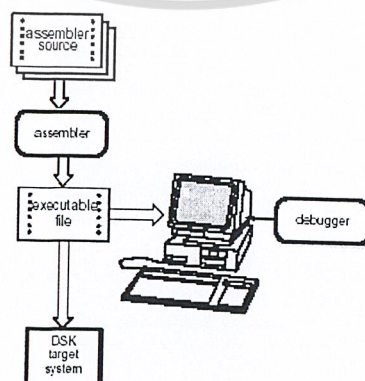
เอกสารรูปที่ 3.13 บอร์ดประมวลผลสัญญาณดิจิทัลซึ่งติดตั้งส่วนเชื่อมต่อช่องรับสัญญาณอินพุตหลัก ช่องรับสัญญาณอินพุตสำรอง ช่องสัญญาณเอาต์พุตและวงจรรวมสัญญาณ ซึ่งใช้ในการทดลอง



รูปที่ 3.14 การวัดผลตอบสนองของตัวกรองปรับตัวในการสร้างจริงในระบบประมวลผลสัญญาณ

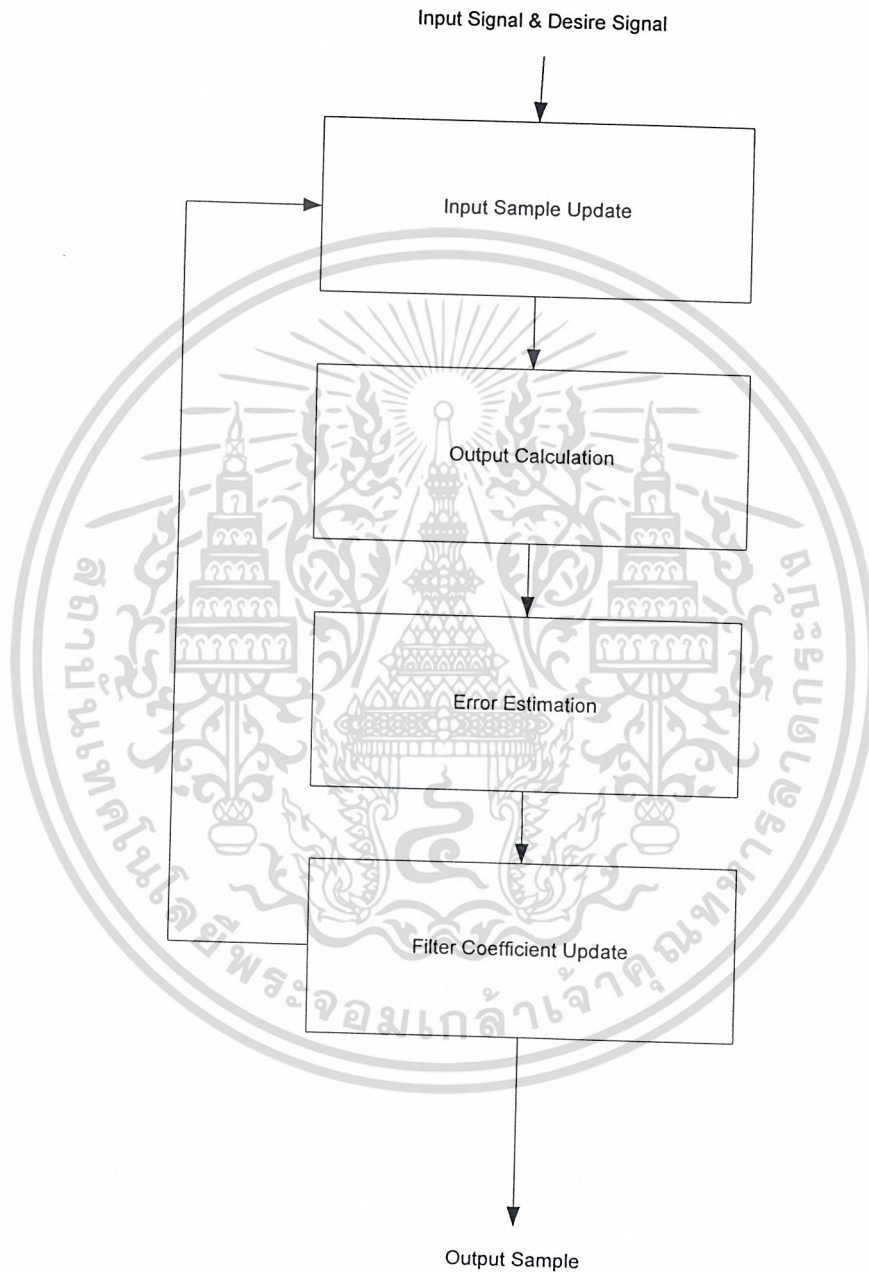
ขั้นตอนต่างๆในการพัฒนาอัลกอริทึมสำหรับตัวประมวลผลสัญญาณดิจิทัล

1. เขียนอัลกอริทึมในรูปของภาษาสำหรับคอมพิวเตอร์หรือการเขียนโค้ด ซึ่งนิยมใช้ภาษาแอสเซมบลีเนื่องจากมีคำสั่งที่สั้นและเมื่อทำการแปลงและป้อนให้ตัวประมวลผลสัญญาณแล้วสามารถทำงานได้เร็ว
2. ทำการแปลงโค้ดเป็นคำสั่งสำหรับตัวประมวลผลสัญญาณดิจิทัล โดยใช้โปรแกรมแอสเซมเบลอร์ (Assembler) ในการแปลงโค้ดและรวมโค้ดคำสั่งสำหรับตัวประมวลผลสัญญาณดิจิทัล
3. ทำการป้อนโปรแกรมให้แก่ตัวประมวลผลสัญญาณดิจิทัลแล้วตรวจสอบผลการทำงานหากไม่เป็นไปตามต้องการสามารถตรวจสอบความผิดพลาดด้วยตัวดีบักเกอร์ (Debugger)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 3.15 กระบวนการพัฒนาอัลกอริทึมสำหรับตัวประมวลผลสัญญาณดิจิทัล
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1 การสร้างตัวกรองปรับตัวแบบเอฟไออาร์ในระบบประมวลผลสัญญาณดิจิทัล
 การเขียนโปรแกรมสร้างตัวกรองปรับตัวแบบเอฟไออาร์นั้น เนื่องจากไม่มีส่วนป้อนกลับ (Recursive part) จึงสามารถเขียนโปรแกรมโดยใช้การจัดหน่วยความจำแบบวน (Circular Buffer) ได้
 สำหรับการไหลของโปรแกรมจะเป็นไปดังรูป

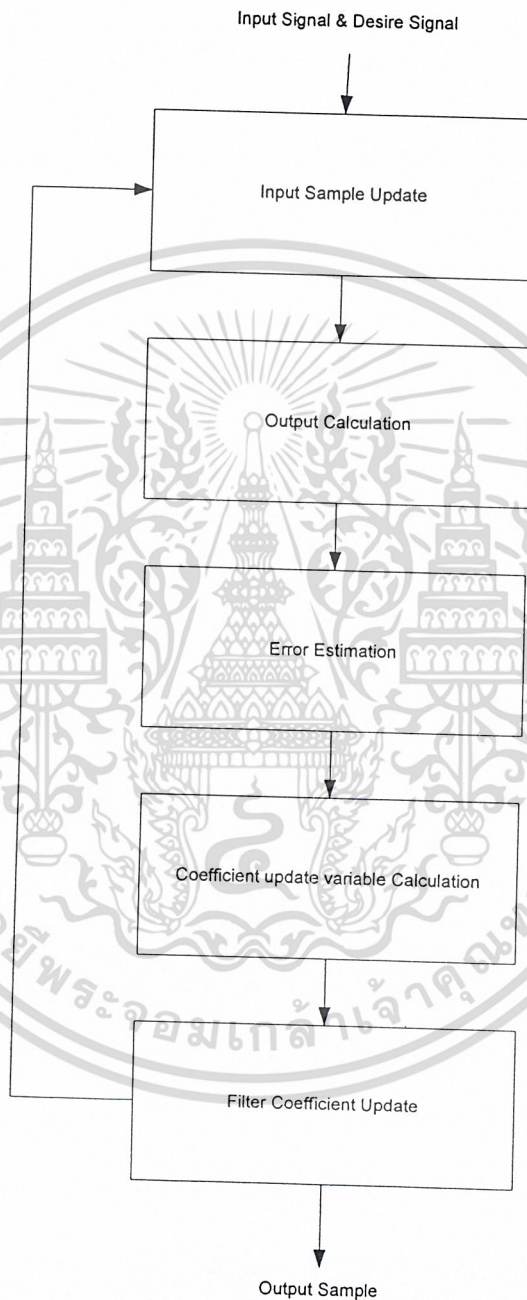


รูปที่ 3.16 แสดงการไหลของโปรแกรมซิมูเลชันสำหรับตัวกรองปรับตัวแบบเอฟไออาร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 การสร้างตัวกรองปรับตัวแบบไออาร์ในระบบประมวลผลสัญญาณดิจิทัล

การเขียนโปรแกรมสร้างตัวกรองปรับตัวแบบไออาร์นั้น เนื่องจากมีส่วนป้อนกลับจึงไม่สามารถเขียนโปรแกรมโดยใช้การจัดหน่วยความจำแบบวน (Circular Buffer) ได้ จึงใช้การจัดหน่วยความจำแบบอ้างอิงตำแหน่งโดยทั่วไปแทน สำหรับการไหลของโปรแกรมจะเป็นไปดังรูป

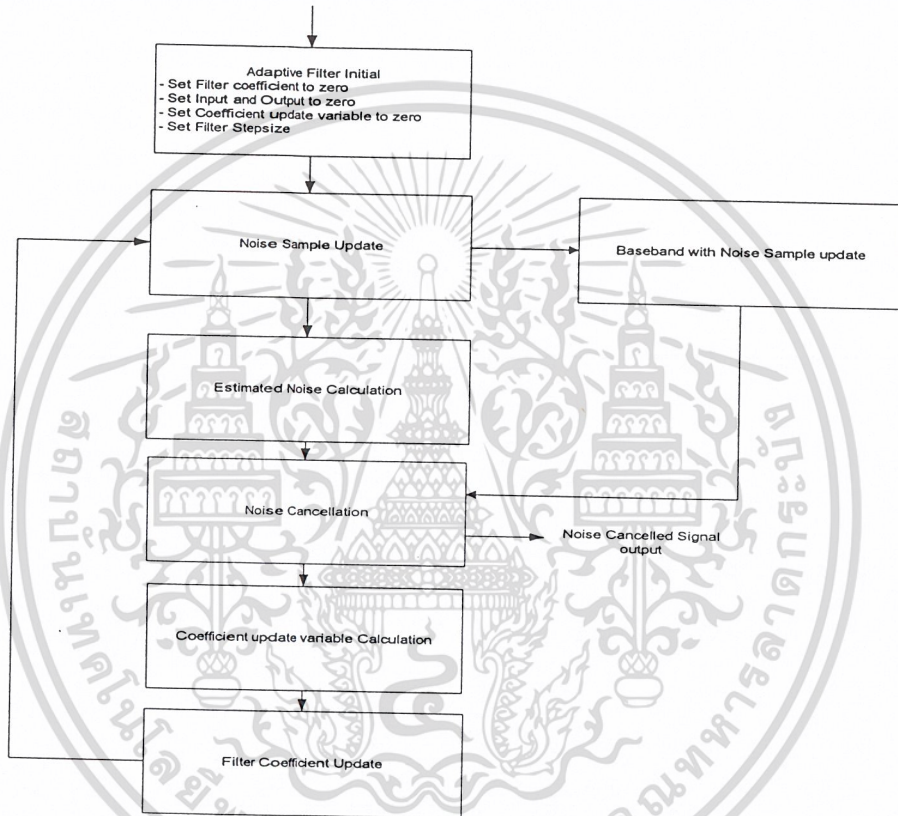


รูปที่ 3.17 แสดงการไหลของโปรแกรมซิมูเลชันสำหรับตัวกรองปรับตัวแบบเอฟไออาร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3 การสร้างตัวกำจัดสัญญาณรบกวนในระบบประมวลผลสัญญาณดิจิทัล

การเขียนโปรแกรมสร้างตัวกำจัดสัญญาณนั้น เนื่องใช้ตัวกรองปรับตัวแบบไอไออาร์เป็นหลัก จึงทำให้มีส่วนป้อนกลับ(Recursive part) ทำให้ไม่สามารถเขียนโปรแกรมโดยใช้การจัดหน่วยความจำแบบวน (Circular Buffer) ได้ จึงใช้การจัดหน่วยความจำแบบอ้างอิงตำแหน่งโดยทั่วไปแทน สำหรับการไหลของโปรแกรมจะเป็นไปดังรูป



รูปที่ 3.18 แสดงการไหลของโปรแกรมสำหรับตัวกำจัดสัญญาณรบกวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 การวัดผลตอบสนองโดยการขมิบเลขน้ของระบบต่างๆ

การวัดผลตอบสนองของตัวกรองปรับตัว โดยการขมิบเลขน้และสร้างจริงในระบบประมวลผล สัญญาณดิจิตอลนี้ ทำการวัดผลตอบสนองต่างๆที่สำคัญ โดยแสดงผลตอบสนองเป็นรูปภาพ ซึ่ง นอกจากการเก็บผลแล้วจะทำการเปรียบเทียบและอธิบายผลตอบสนองที่เก็บได้

4.1.1 การขมิบเลขน้ของตัวกรองปรับตัวแบบเอฟไออาร์

การขมิบเลขน้หาผลตอบสนองต่างๆของตัวกรองปรับตัวทำการขมิบเลขน้จากโปรแกรมซึ่งเขียนขึ้น โดยโปรแกรม แมทแลบโดยกำหนดเลือกใช้ความถี่สัญญาณเบสแบนด์ 80Hz ใช้ความถี่การซ้คตัวอย่าง เท่ากับ 8 kHz และใช้จำนวนไอเทอร์เรชั่นเท่ากับ 1000 ไอเทอร์เรชั่น และทำการเปลี่ยนค่าต่างๆคือ

4.1.1.1. หาผลตอบสนองต่อความถี่ต่างๆของตัวกรองปรับตัวแบบเอฟไออาร์

โดยเลือกใช้ตัวกรองอันดับ 50 , ขนาดสเตปในการปรับตัวค้งที่เท่ากับ 0.005 แล้วทำการหาผลตอบสนอง ต่อสัญญาณรบกวนและสัญญาณรบกวนรูปไซน์ความถี่ 500 Hz , 1 kHz และ 3 kHz

4.1.1.2. หาผลตอบสนองต่อขนาดสเตปต่างๆของตัวกรองปรับตัวแบบเอฟไออาร์

โดยเลือกใช้ตัวกรองอันดับ 50 สัญญาณรบกวนรูปไซน์ความถี่ค้งที่ 2 kHz หาผลตอบสนองที่ขนาดสเตป ต่างๆค้งนี้ 0.003, 0.005, 0.01 ใช้จำนวนไอเทอร์เรชั่น 1000 ไอเทอร์เรชั่น

4.1.1.3 หาผลตอบสนองต่ออันดับต่างๆของตัวกรองปรับตัวแบบเอฟไออาร์

โดยเลือกใช้ ขนาดสเตปในการปรับตัวค้งที่เท่ากับ 0.005 หาผลตอบสนองต่อสัญญาณรบกวนรูป ไซน์ความถี่ 2 kHz โดยเปลี่ยนอันดับของตัวกรองระหว่าง 10, 25 และ 50

4.1.2 การขมิบเลขน้ของตัวกรองปรับตัวแบบไอไออาร์

การขมิบเลขน้หาการตอบสนองต่างๆของตัวกรองปรับตัวทำโดยกำหนดเลือกใช้ความถี่สัญญาณ เบสแบนด์ 80 Hz ความถี่การซ้คตัวอย่าง 8 kHz และใช้จำนวนไอเทอร์เรชั่นเท่ากับ 1000 ไอเทอร์เรชั่นและ ทำการเปลี่ยนค่าต่างๆคือ

4.1.2.1 หาผลตอบสนองต่อความถี่ต่างๆของตัวกรองปรับตัวแบบไอไออาร์

โดยเลือกใช้ตัวกรองอันดับ 10 ขนาดสเตปในการปรับตัวค้งที่เท่ากับ 0.005 แล้วทำการหาผลตอบสนองต่อ สัญญาณรบกวนและสัญญาณรบกวนรูปไซน์ความถี่ 500 Hz , 2 kHz และ 3 kHz

4.1.2.2 หาผลตอบสนองต่อขนาดสเตปต่างๆของตัวกรองปรับตัวแบบไอไออาร์

โดยเลือกใช้ตัวกรองอันดับ 10 สัญญาณรบกวนรูปไซน์ความถี่ค้งที่ 2 kHz หาผลตอบสนองที่ขนาดสเตป ต่างๆค้งนี้ 0.001, 0.005, 0.01 และ 0.05 ซึ่งงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

4.1.2.3 หาผลตอบสนองต่ออันดับต่างๆของตัวกรองปรับตัวแบบไอไออาร์

โดยเลือกใช้ ขนาดสเตปในการปรับตัวค้งที่เท่ากับ 0.005 หาผลตอบสนองต่อสัญญาณรบกวนรูป

ไชน์ความถี่ 2 kHz โดยเปลี่ยนอันดับของตัวกรองระหว่าง 10, 25 และ 50

4.1.3 การซิมูเลชันของตัวกำจัดสัญญาณรบกวน

การซิมูเลชันหาการตอบสนองต่างๆของตัวกำจัดสัญญาณรบกวนทำได้โดยการกำหนดใช้ความถี่สัญญาณเบสแบนด์ 80Hz ความถี่การซัดตัวอย่าง 8 kHz ใช้จำนวนไอเทอร์เรชั่นเท่ากับ 1000 และทำการเปลี่ยนค่าต่างๆคือ

4.1.3.1 หาผลตอบสนองต่อความถี่ต่างๆของตัวกำจัดสัญญาณรบกวน

โดยเลือกใช้ตัวกรองอันดับ 2 ขนาดสเตปในการปรับตัวคองที่เท่ากับ 0.005 หาผลตอบสนองต่อสัญญาณรบกวนสุ่มและสัญญาณรบกวนรูปไชน์ความถี่ต่างๆดังนี้ 50 Hz, 500 Hz, 1 kHz, 2 kHz, 3 kHz ที่ขนาดแอมพลิจูดของสัญญาณรบกวนเท่ากับ 1

4.1.3.2 หาผลตอบสนองต่อขนาดสเตปต่างๆของตัวกำจัดสัญญาณรบกวน

โดยการใช้ขนาดสเตปในการปรับตัวคองที่เท่ากับ 0.03 ทำการหาผลตอบสนองต่อสัญญาณรบกวนรูปไชน์ความถี่ 1 kHz ที่ขนาดแอมพลิจูดของสัญญาณรบกวนเท่ากับ 0.3, 0.5 และ 0.7

4.1.3.3 ผลตอบสนองต่ออันดับต่างๆของตัวกำจัดสัญญาณรบกวน

โดยเลือกใช้ขนาดสเตปในการปรับตัวคองที่เท่ากับ 0.005 หาผลตอบสนองต่อสัญญาณรบกวนสุ่ม ที่ขนาดแอมพลิจูดของสัญญาณรบกวนเท่ากับ 0.3, 0.5 และ 0.7

4.1.4 การซิมูเลชันของการใช้ตัวกรองปรับตัวหาคุณสมบัติของระบบที่ไม่ทราบ

การซิมูเลชันหาการตอบสนองต่างๆของตัวกรองปรับตัวในการจำลองคุณลักษณะของระบบอื่นๆ จะใช้ความถี่สัญญาณอินพุทของระบบ 80 Hz ความถี่การซัดตัวอย่าง 8 kHz และจะมีการใช้จำนวนของ ไอเทอร์เรชั่นเท่ากับ 400 ไอเทอร์เรชั่น โดยจะมีการสมมุติให้ระบบที่ไม่ทราบ มีทรานส์เฟอร์ฟังก์ชันเป็น

$$h(z) = \frac{0.2066 + 0.4131z^{-1} + 0.2066z^{-2}}{1 - 0.3695z^{-1} + 0.1598z^{-2}}$$

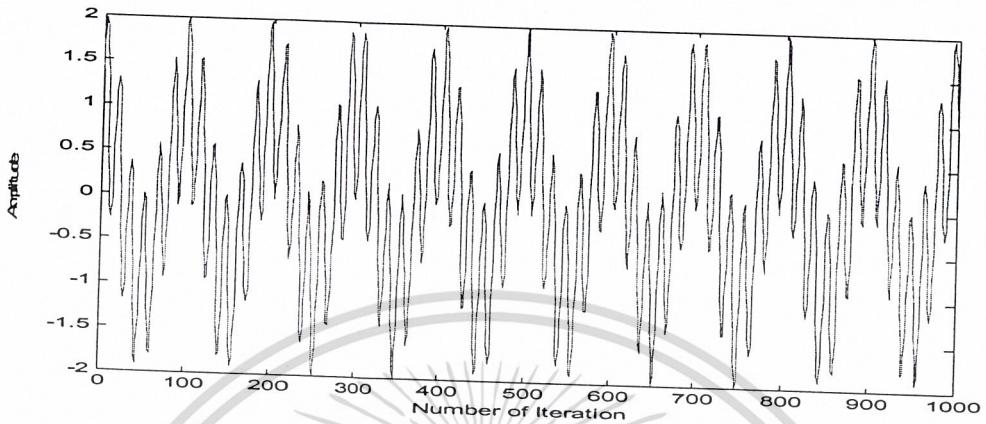
แล้วหาผลตอบสนองที่ขนาดสเตปเท่ากับ 0.005, 0.01, 0.05 และ 0.1 โดยเลือกใช้ตัวกรองอันดับ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

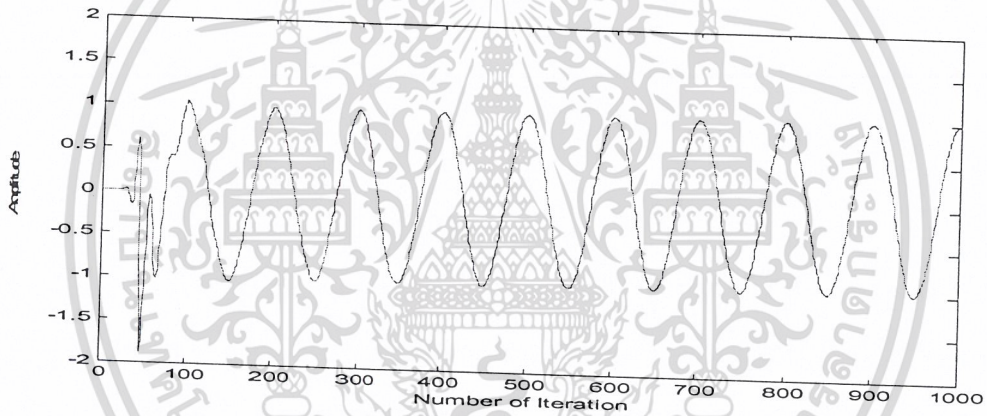
4.2 ผลตอบสนองของต่างๆจากการจิมูเลชั่น

4.2.1 ผลตอบสนองต่างๆของตัวกรองปรับตัวแบบเอฟไออาร์

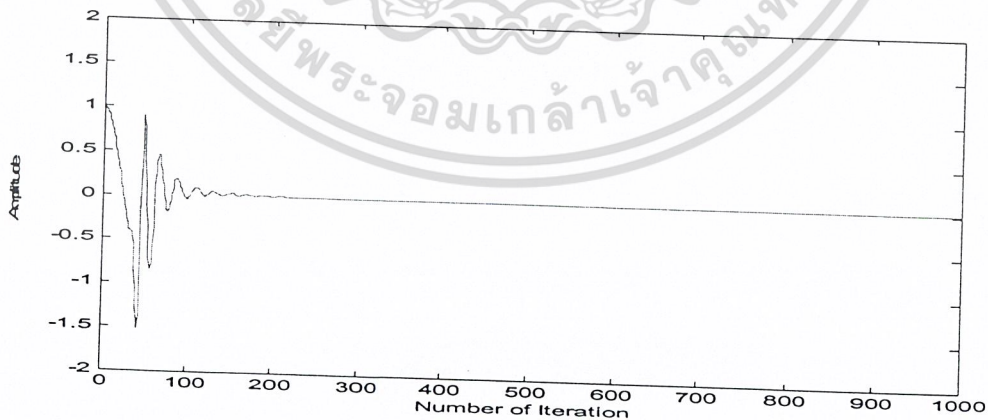
4.2.1.1 ผลตอบสนองต่อความถี่ต่างๆของตัวกรองปรับตัวแบบเอฟไออาร์



ก) สัญญาณเบสแบนด์ซึ่งถูกรบกวน

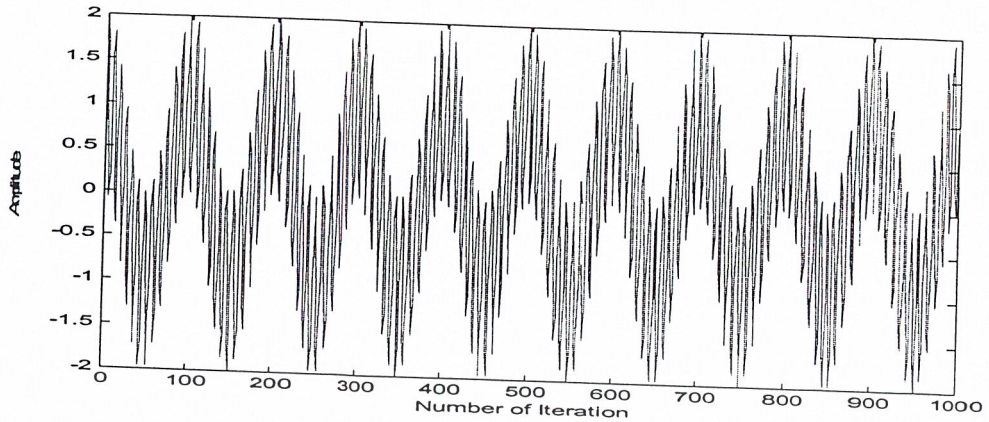


ข) สัญญาณที่กรองได้

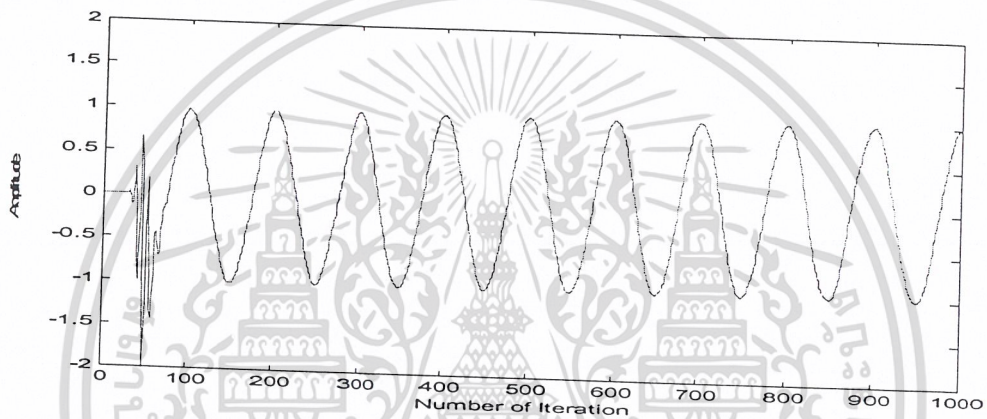


ค) สัญญาณความผิดพลาด

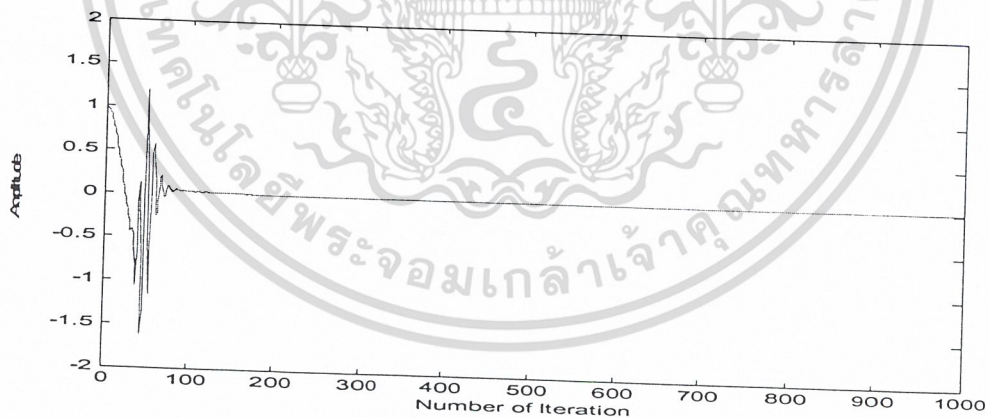
รูปที่ 4.1 ผลตอบสนองต่างๆของตัวกรองปรับตัวแบบเอฟไออาร์อันดับ 50 ต่อสัญญาณรบกวนรูปไซน์ความถี่ 500 Hz ความถี่การซึกตัวอย่าง 8 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz ไม่ว่ากรณีใดจะปรับค่าขั้นห่วยให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงใจของเอกสารทุกครั้งที่มีการนำไปใช้ ขนาดสเกลเท่ากับ 0.005



ก) สัญญาณเบสแบนด์ซึ่งถูกรบกวน



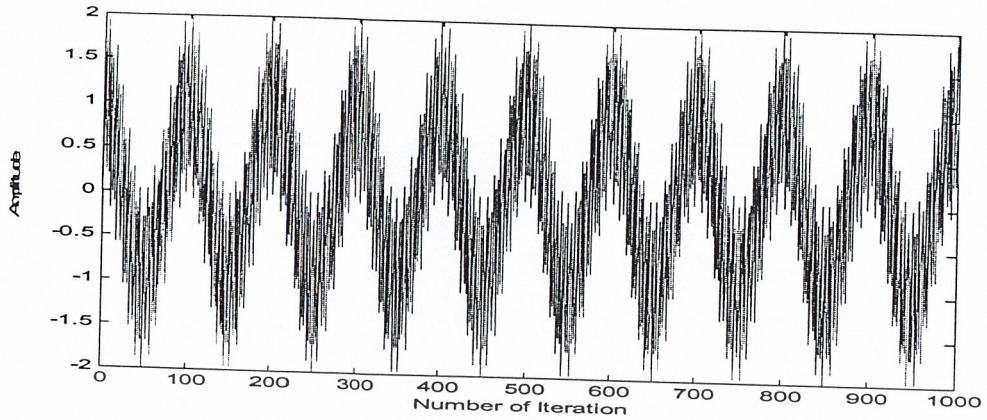
ข) สัญญาณที่กรองได้



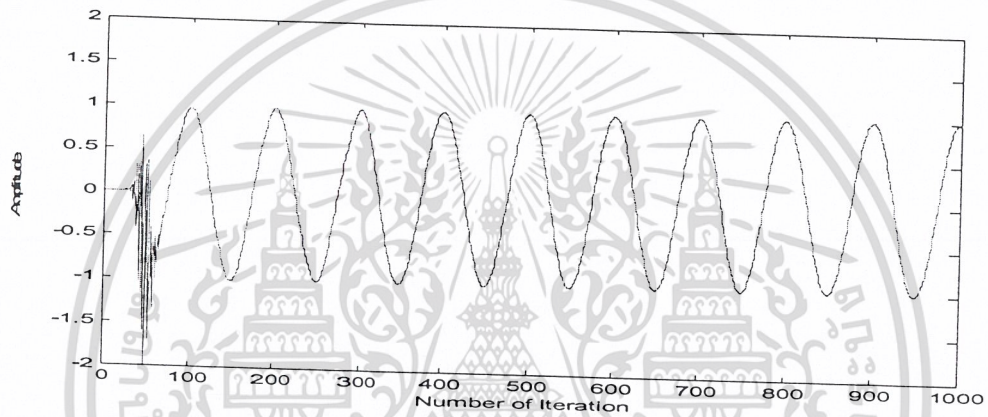
ค) สัญญาณความผิดพลาด

รูปที่ 4.2 ผลตอบสนองต่างๆ ของตัวกรองปรับควมแบนเอพไออาร์อันดับ 50 ต่อ สัญญาณรบกวนรูปไซน์ ความถี่ 2 kHz ความถี่การซ้กตัวอย่าง 8 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz ขนาดสเกลเท่ากับ 0.005

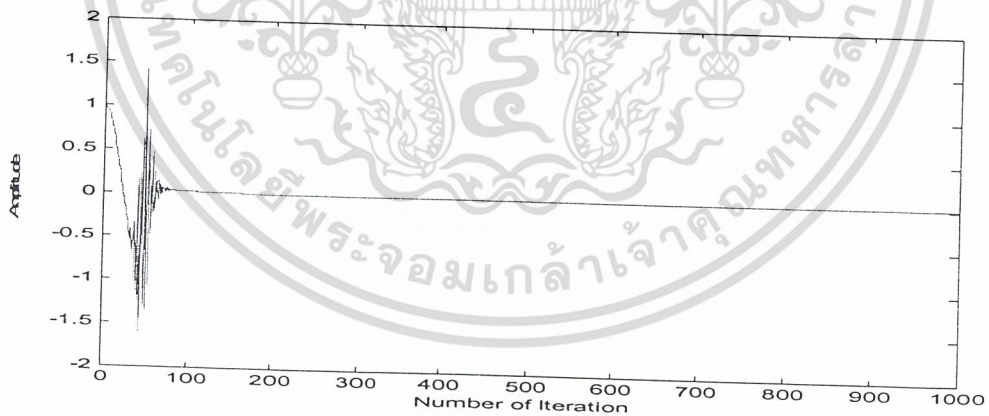
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก) สัญญาณเบสแบนด์ซึ่งถูกรบกวน



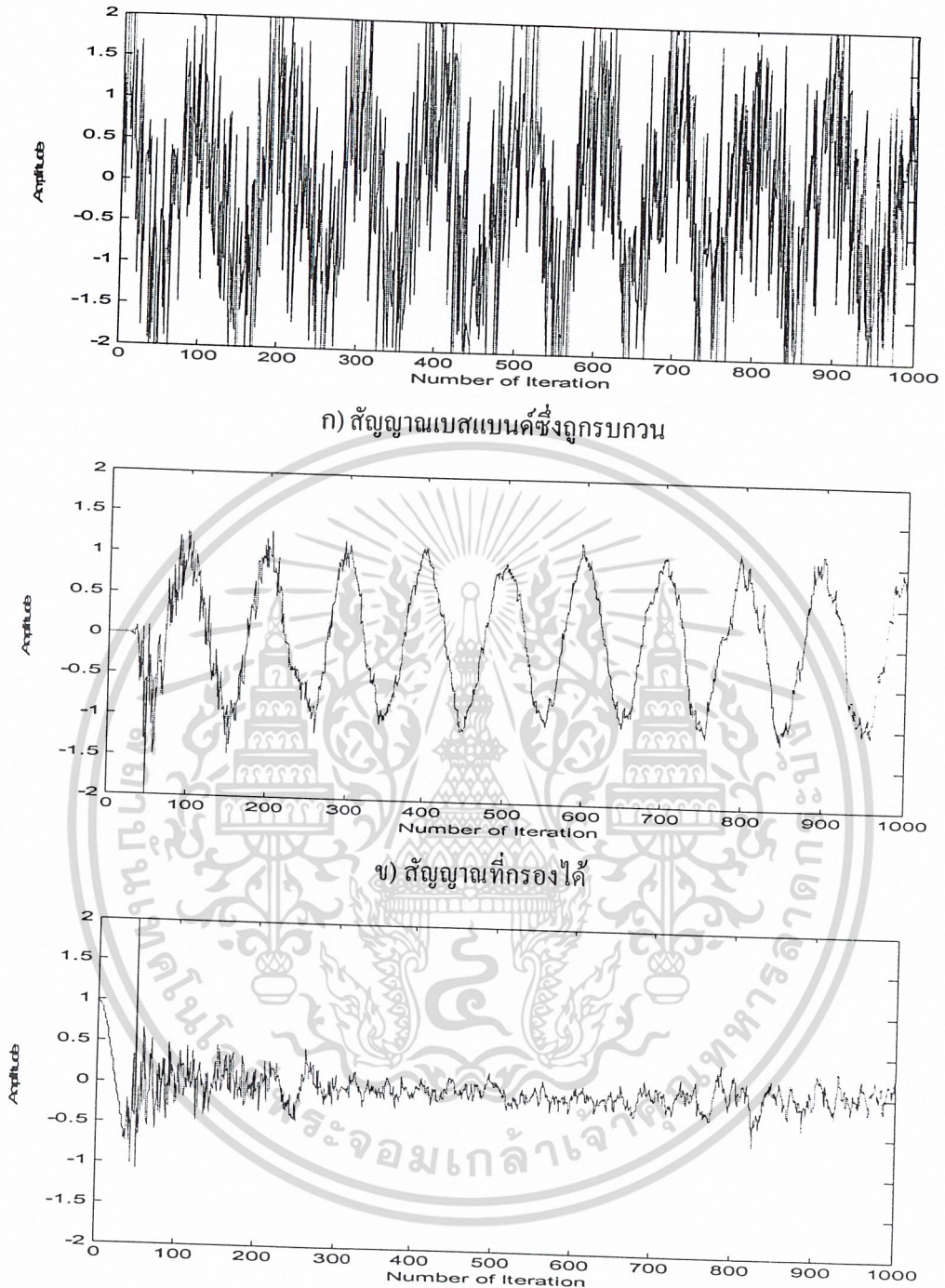
ข) สัญญาณที่กรองได้



ค) สัญญาณความผิดพลาด

รูปที่ 4.3 ผลตอบสนองต่างๆ ของตัวกรองปรับตัวแบบเอฟไออาร์อันดับ 50 ต่อ สัญญาณรบกวนรูปไซน์ ความถี่ 3 kHz ความถี่การซัดตัวอย่าง 8 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz ขนาดสเกลเท่ากับ 0.005

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 ผลตอบสนองต่างๆ ของตัวกรองปรับตัวแบบเฟอไออาร์อันดับ 50 ต่อ สัญญาณรบกวนสุ่ม ความถี่การซัดตัวอย่าง 8 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz ขนาดสเกลเท่ากับ 0.005

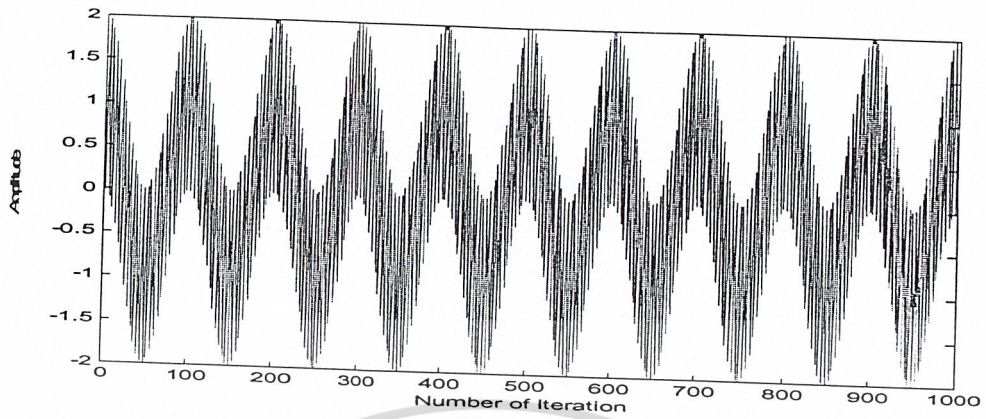
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.1 ถึง 4.4 พบว่า ผลตอบสนองต่อความถี่ต่างๆของตัวกรองปรับตัวแบบเอฟไออาร์ มีผลตอบสนองในการกำจัดสัญญาณรบกวนได้ดี และสังเกตได้ว่าจะมีการลู่อเข้าที่ดีขึ้น เมื่อสัญญาณรบกวนมีความถี่แตกต่างกับสัญญาณมากขึ้น

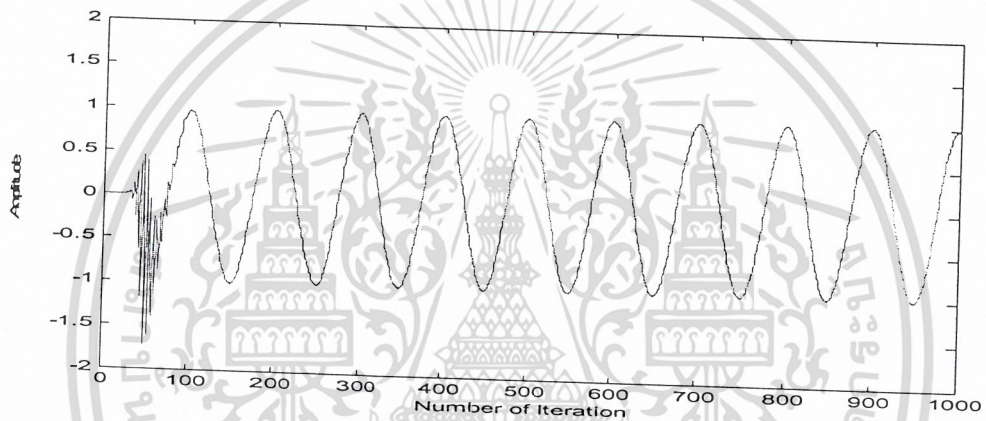


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

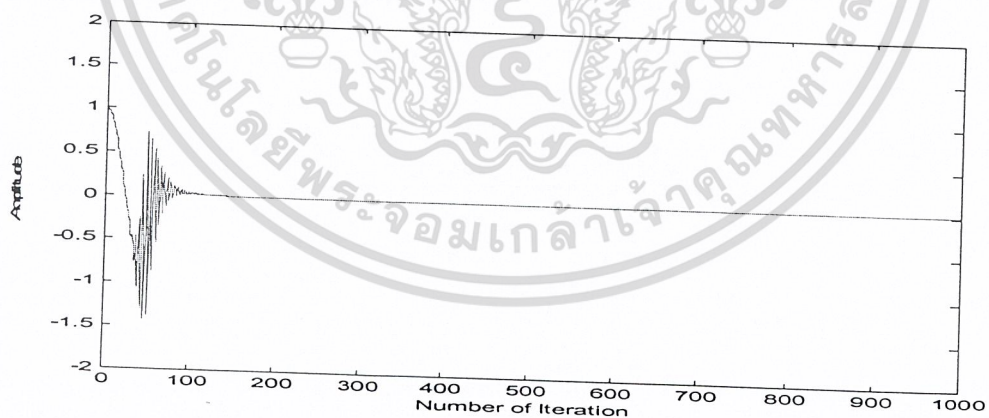
4.2.1.2 ผลตอบสนองต่อขนาดสเตปต่างๆของตัวกรองปรับตัวแบบเอฟไออาร์



ก) สัญญาณเบสแบนด์ซึ่งถูกรบกวน



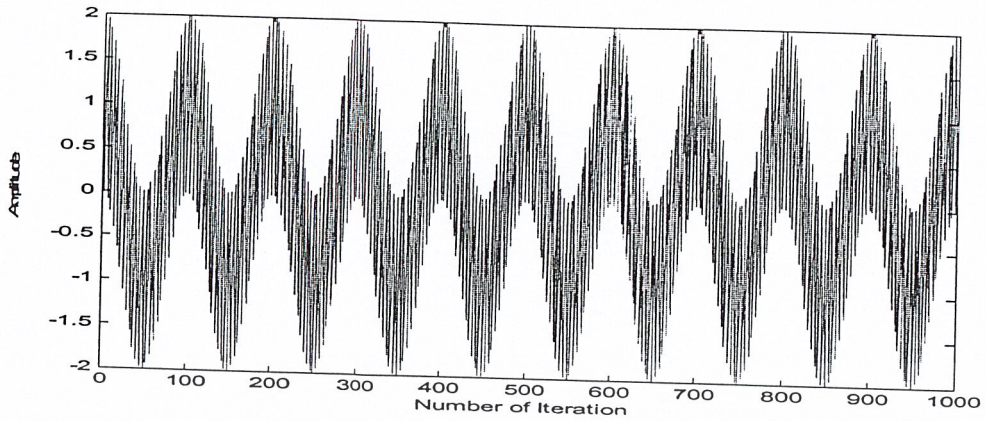
ข) สัญญาณที่กรองได้



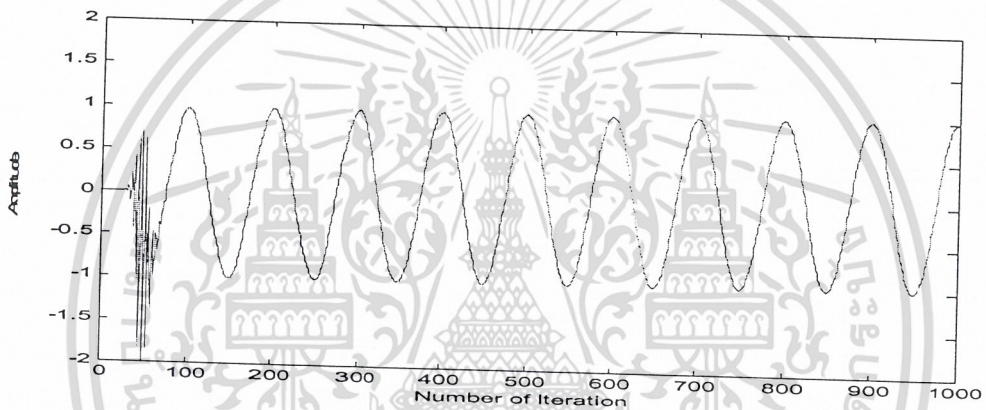
ค) สัญญาณความผิดพลาด

รูปที่ 4.5 ผลตอบสนองต่างๆ ของตัวกรองปรับตัวแบบเอฟไออาร์อันดับ 50 ต่อ สัญญาณรบกวนรูปไซน์ ความถี่ 2 kHz ความถี่การซัดตัวอย่าง 8 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz ขนาดสเตปเท่ากับ 0.003

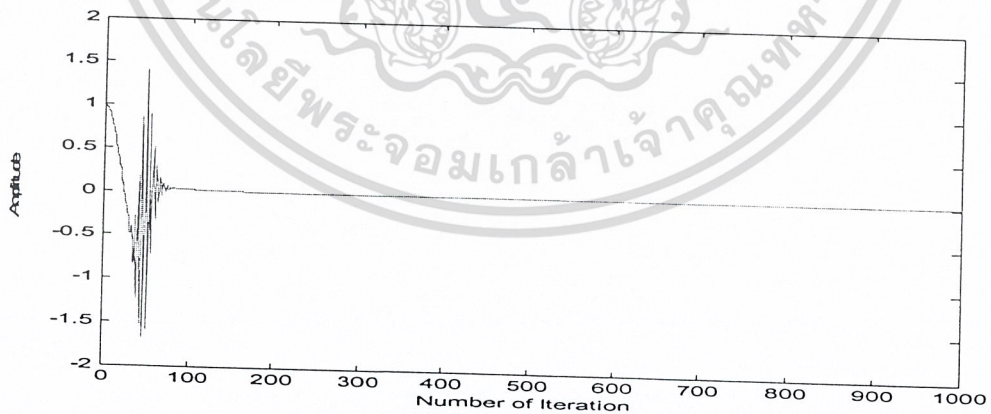
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก) สัญญาณเบสแบนด์ซึ่งถูกรบกวน



ข) สัญญาณที่กรองได้



ค) สัญญาณความผิดพลาด

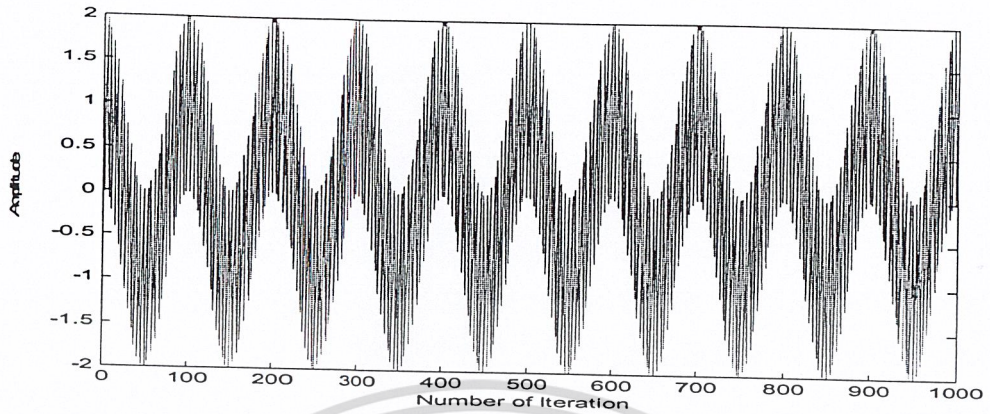
รูปที่ 4.6 ผลตอบสนองต่างๆ ของตัวกรองปรับตัวแบบเอฟไออาร์อันดับ 50 ต่อ สัญญาณรบกวนรูปไซน์

ความถี่ 2 kHz ความถี่การซัดตัวอย่าง 8 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz

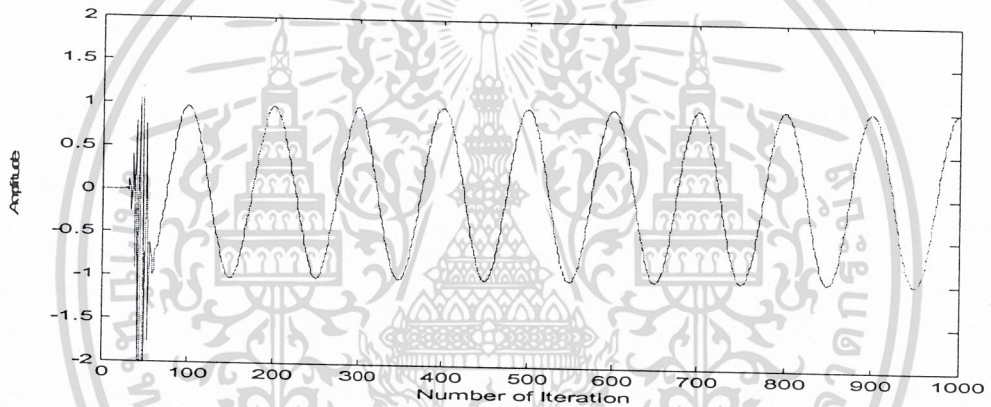
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ขนาดสเกลเท่ากับ 0.005

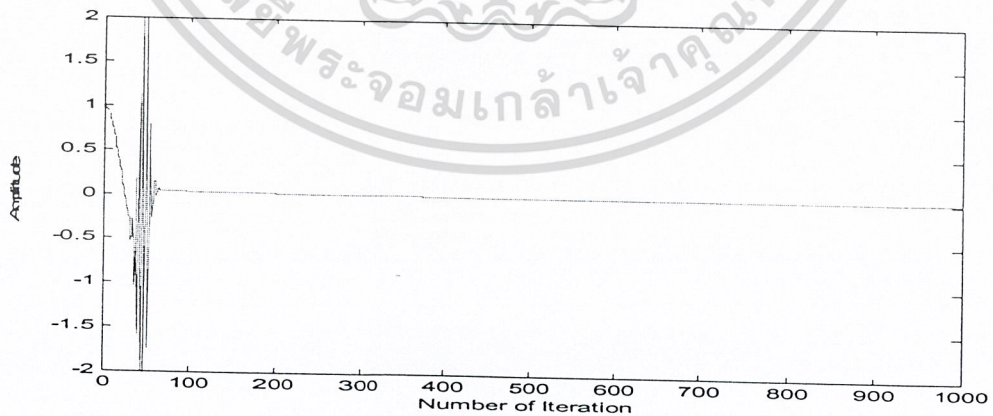
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก) สัญญาณเบสแบนด์ซึ่งถูกรบกวน



ข) สัญญาณที่กรองได้



ค) สัญญาณความผิดพลาด

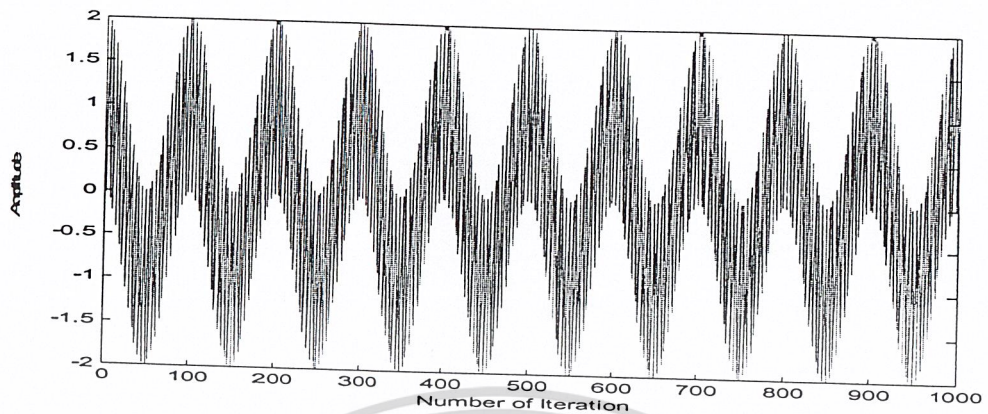
รูปที่ 4.7 ผลตอบสนองต่างๆ ของตัวกรองปรับตัวแบบเอพไออาร์อันดับ 50 ต่อ สัญญาณรบกวนรูปไซน์
 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ได้รับอนุญาต
 ความถี่ 2 kHz, ความถี่การซัดตัวอย่าง 8 kHz, ความถี่สัญญาณเบสแบนด์ 80 Hz
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงชื่อของเอกสารทุกครั้งที่มีการนำไปใช้
 ขนาดสเกลเท่ากับ 0.01

จากรูปที่ 4.5 ถึง 4.7 พบว่าแนวโน้มผลตอบสนองต่อค่าสเตปต่างๆของตัวกรองปรับตัวกรองปรับตัวแบบเอฟไออาร์ มีผลตอบสนองในการกำจัดสัญญาณรบกวนได้ดีและสังเกตได้ว่าจะมีการลู่เข้าที่ดีขึ้น เมื่อขนาดสเตปใหญ่ขึ้นมากขึ้น แต่อย่างไรก็ตาม ขนาดสเตปที่ใหญ่เกินไป อาจทำให้การลู่เข้าเลวลง เนื่องจากการปรับสัมประสิทธิ์ ในแต่ละครั้งมากเกินไป

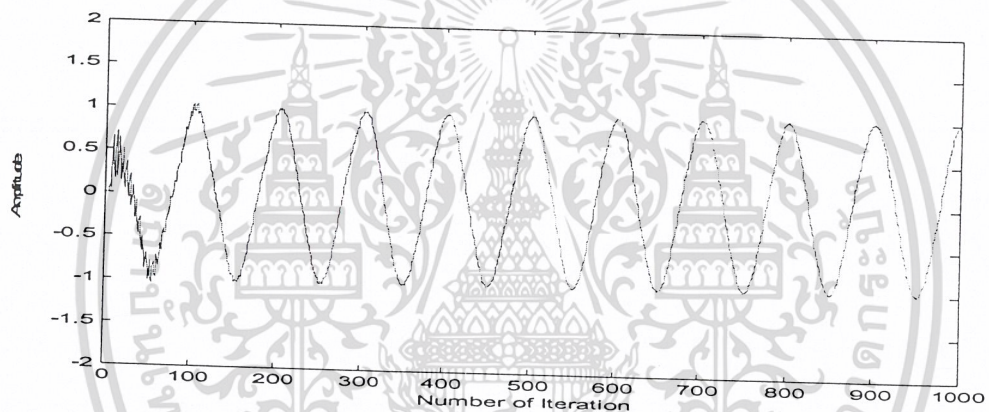


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

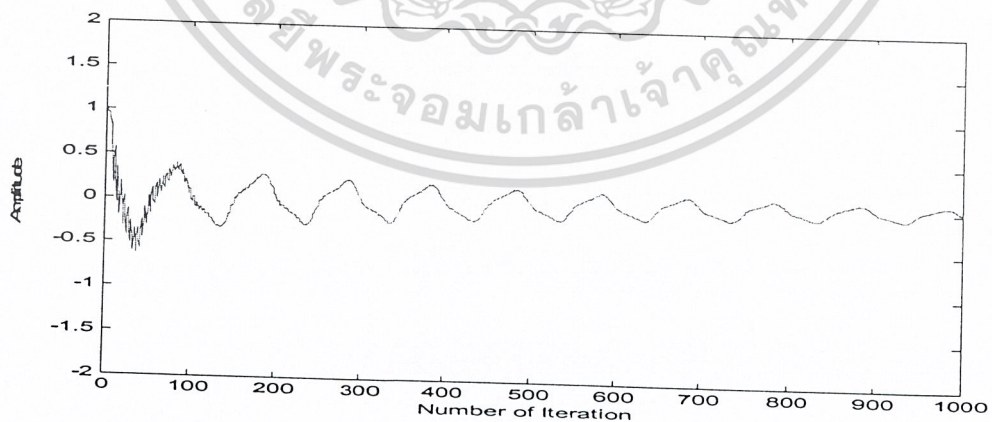
4.2.1.3 ผลตอบสนองต่ออันดับต่างๆของตัวกรองปรับตัวแบบเอฟไออาร์



ก) สัญญาณเบสแบนด์ซึ่งถูกรบกวน

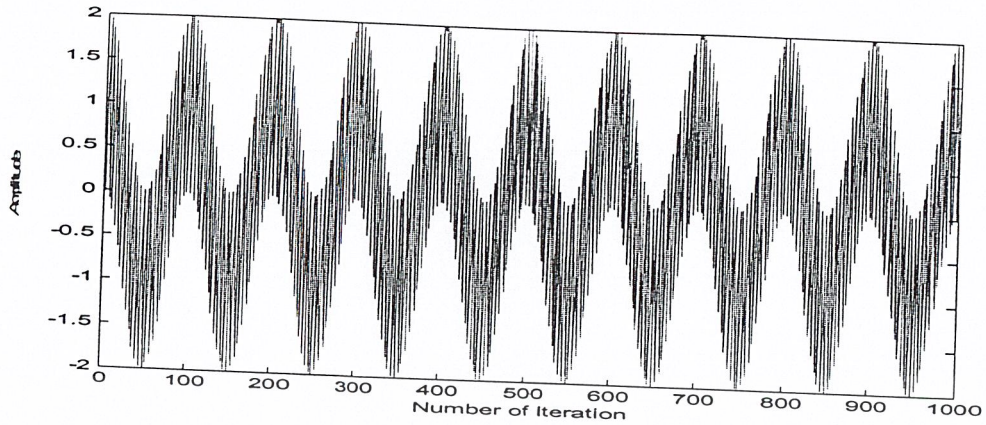


ข) สัญญาณที่กรองได้

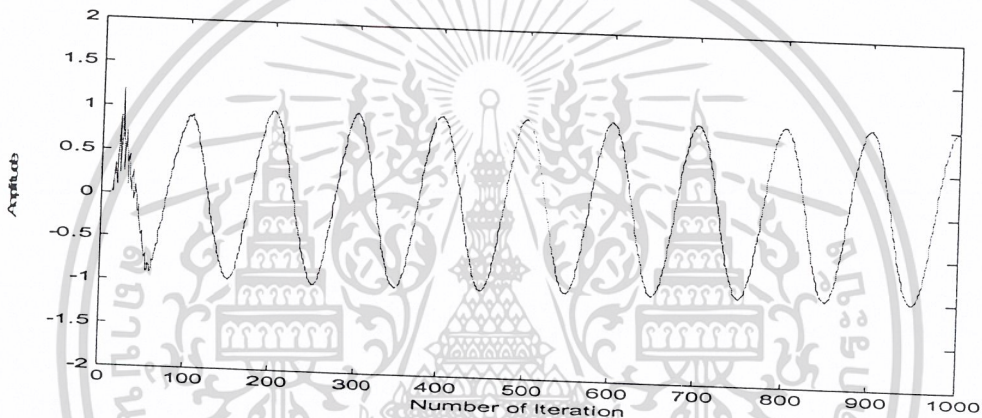


ค) สัญญาณความผิดพลาด

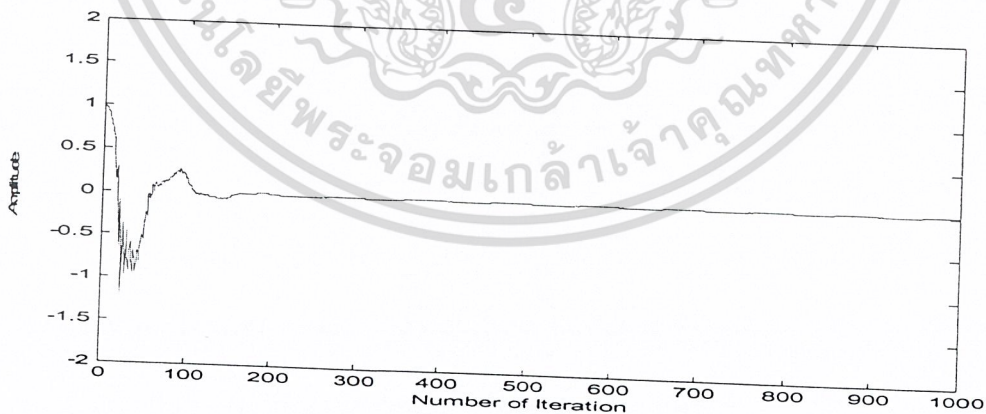
รูปที่ 4.8 ผลตอบสนองต่างๆ ของตัวกรองปรับตัวแบบเอฟไออาร์อันดับ 10 ต่อ สัญญาณรบกวนรูปไซน์ ความถี่ 2 kHz ความถี่การซัดตัวอย่าง 8 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz
ไม่ว่ากรณีใด ๆ ซึ่งเป็นการเปลี่ยนแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
ขนาดสเกลเท่ากับ 0.005



ก) สัญญาณเบสแบนด์ซึ่งถูกรบกวน

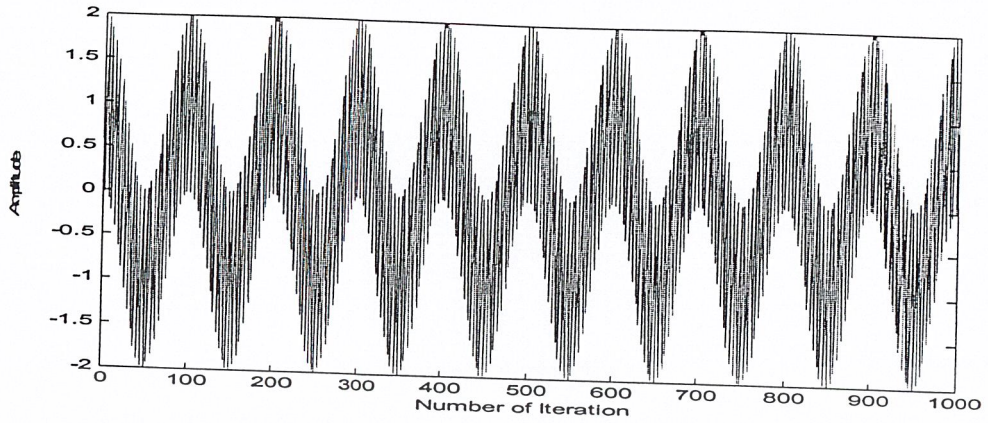


ข) สัญญาณที่กรองได้

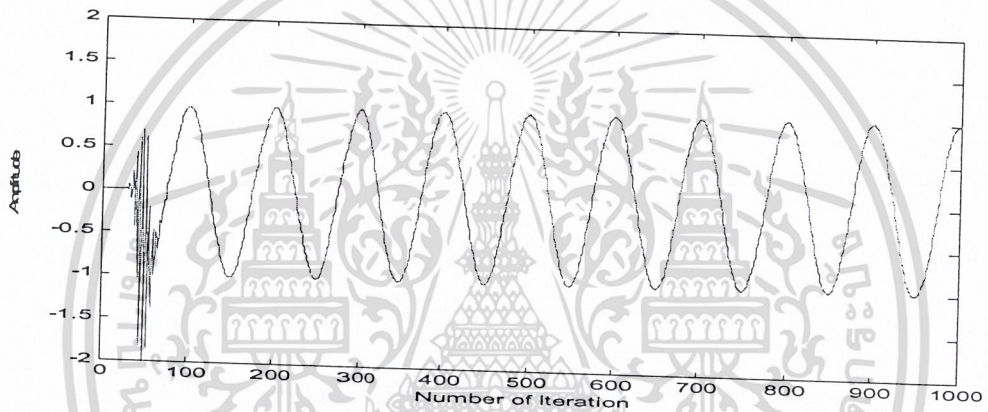


ค) สัญญาณความผิดพลาด

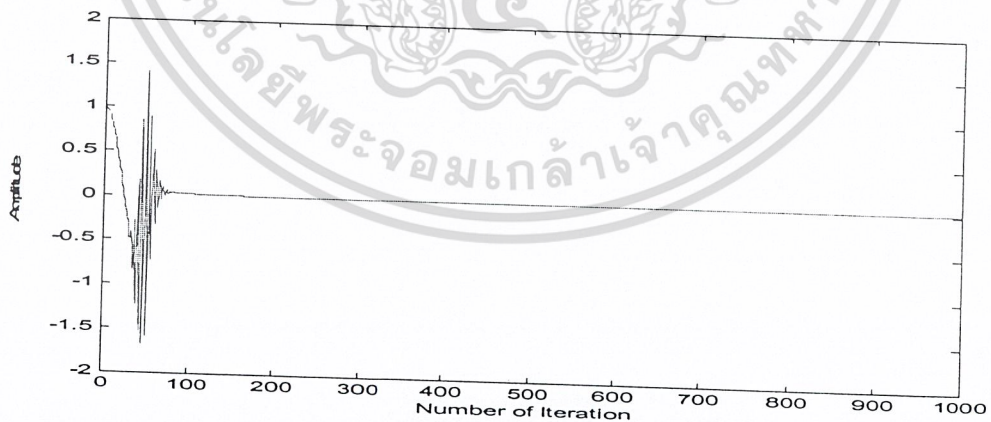
รูปที่ 4.9 ผลตอบสนองต่างๆ ของตัวกรองปรับตัวแบบเอฟไออาร์อันดับ 25 ต่อ สัญญาณรบกวนรูปไซน์ ความถี่ 2 kHz ความถี่การซัดตัวอย่าง 8 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz เอกสารนี้เป็นเอกสารลิขสิทธิ์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก) สัญญาณเบสแบนด์ซึ่งถูกรบกวน



ข) สัญญาณที่กรองได้



ค) สัญญาณความผิดพลาด

รูปที่ 4.10 ผลตอบสนองต่างๆ ของตัวกรองปรับตัวแบบเฟอไออาร์อันดับ 50 ต่อ สัญญาณรบกวนรูปไซน์

ความถี่ 2 kHz ความถี่การซัดตัวอย่าง 8 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปเผยแพร่ประโยชน์ด้านการค้า

ขนาดสเกลเท่ากับ 0.005

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

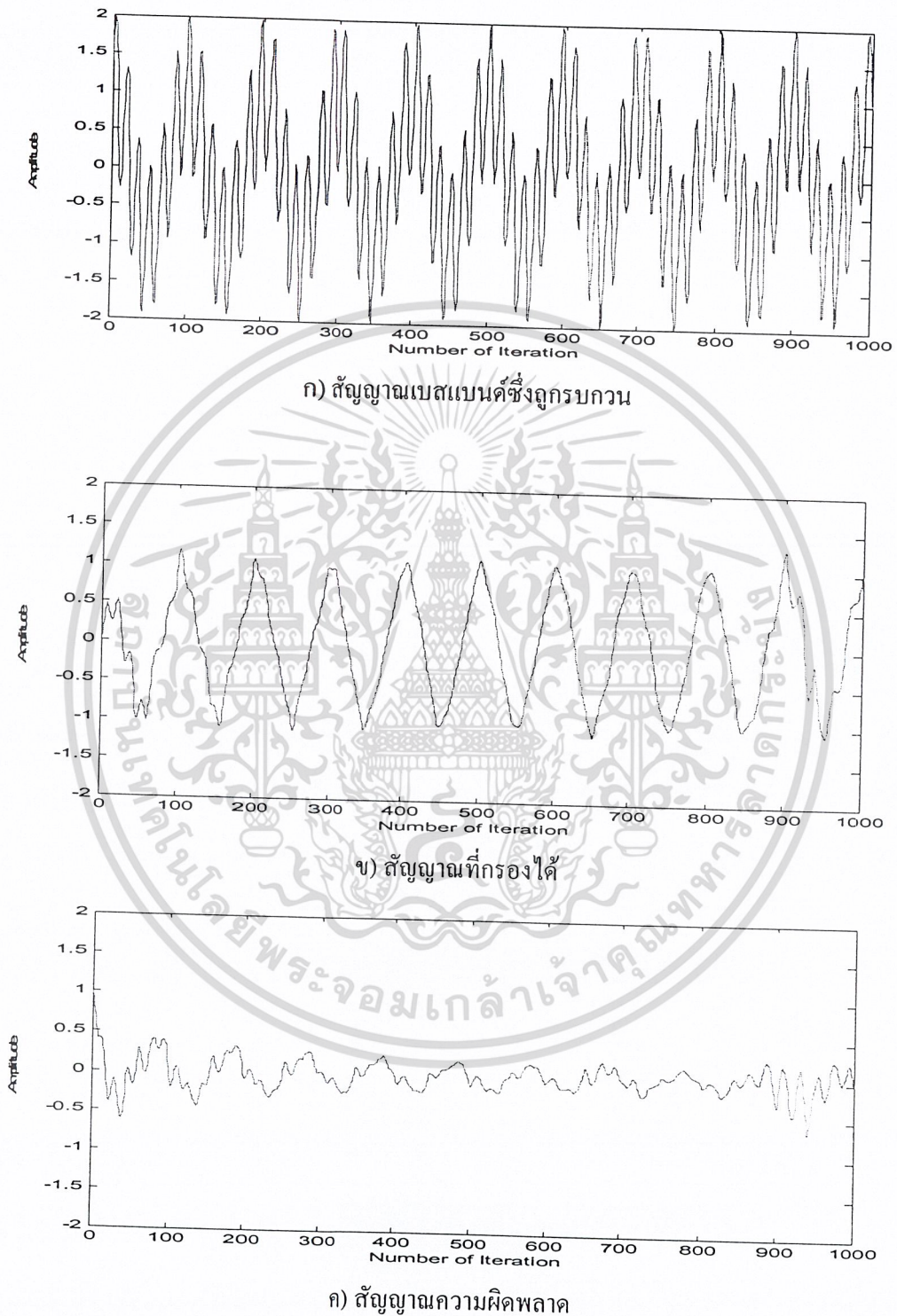
จากรูปที่ 4.8 ถึง 4.10 พบว่า ผลตอบสนองต่ออันดับต่างๆของตัวกรองปรับตัวแบบเอฟไออาร์ มีผลตอบสนองในการกำจัดสัญญาณรบกวนได้ดี และสังเกตได้ว่าจะมีการลู่อเข้าที่ดีขึ้น เมื่ออันดับของตัวกรองมากขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

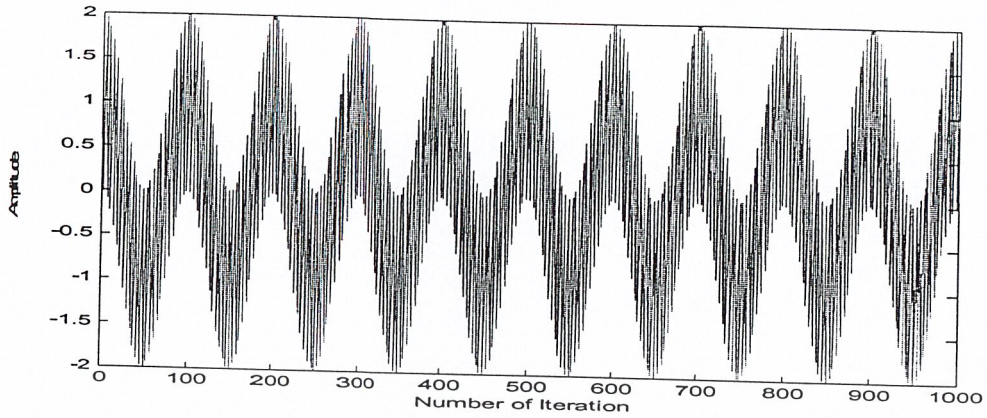
4.2.2 ผลตอบสนองต่างๆของตัวกรองปรับตัวแบบไอไออาร์

4.2.2.1 ผลตอบสนองต่อความถี่ต่างๆ ของตัวกรองปรับตัวแบบไอไออาร์

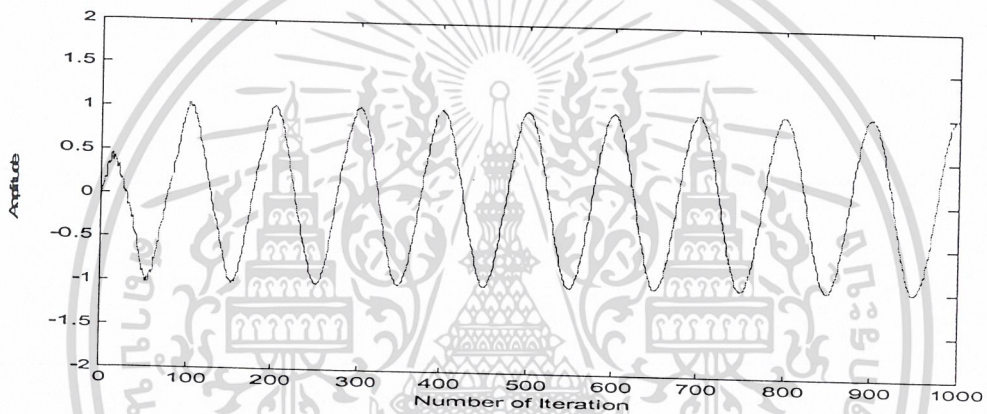


รูปที่ 4.11 ผลตอบสนองต่างๆ ของตัวกรองปรับตัวแบบไอไออาร์อันดับ 10 ต่อ สัญญาณรบกวนรูปไซน์

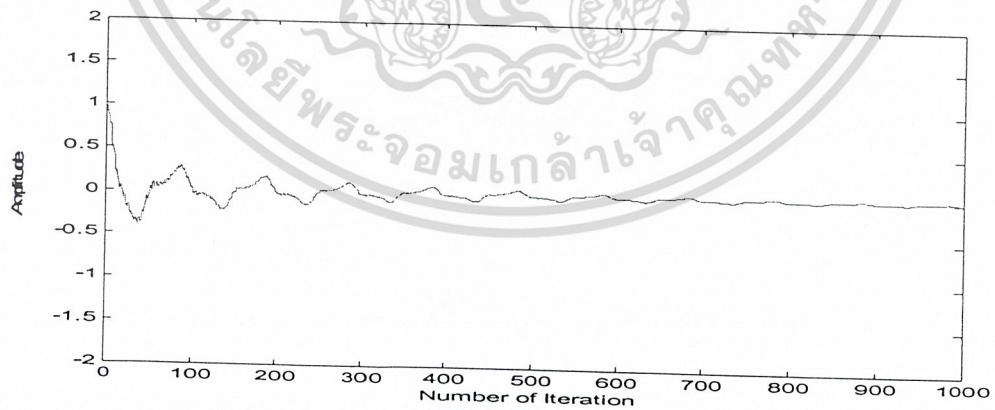
ความถี่ 500 Hz ความถี่การซัดตัวอย่าง 8 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz
 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่ในด้านการค้า
 ไม่ว่ากรณีใดๆ ขันน้ำคสเติปเท่ากับ 0.005



ก) สัญญาณเบสแบนด์ซึ่งถูกรบกวน



ข) สัญญาณที่กรองได้



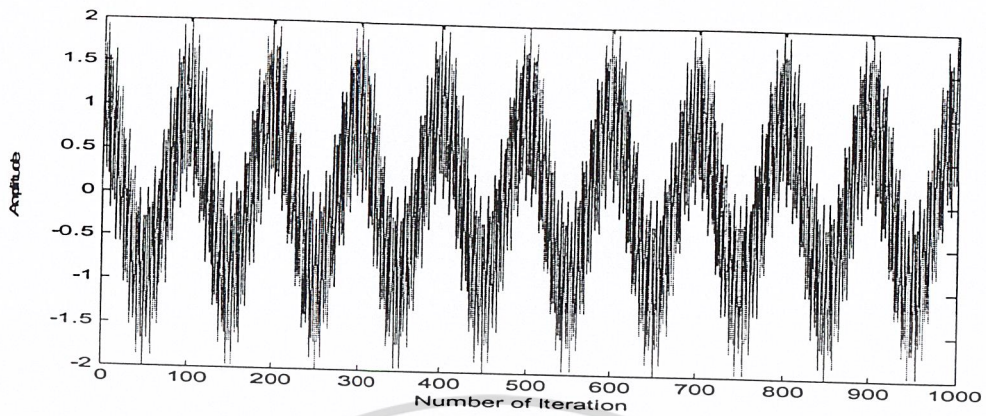
ค) สัญญาณความผิดพลาด

รูปที่ 4.12 ผลตอบสนองต่างๆ ของตัวกรองปรับตัวแบบไอไออาร์อันดับ 10 ต่อ สัญญาณรบกวนรูปไซน์

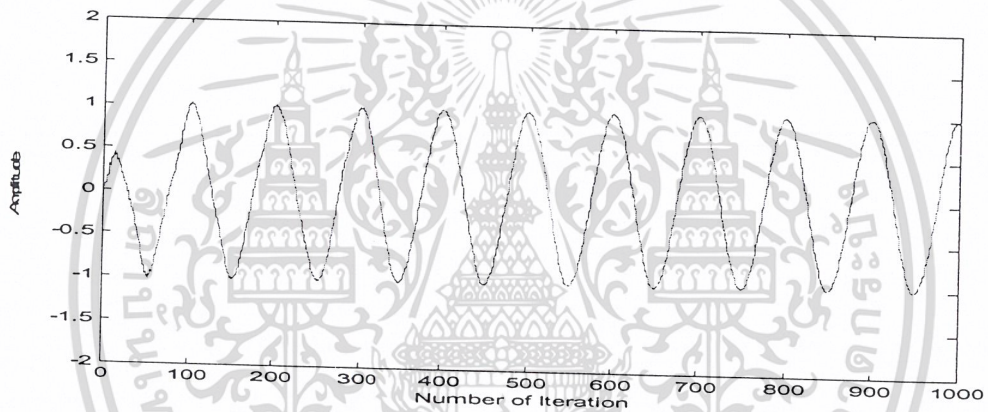
ความถี่ 2 kHz ความถี่การซิกตัดอย่าง 8 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

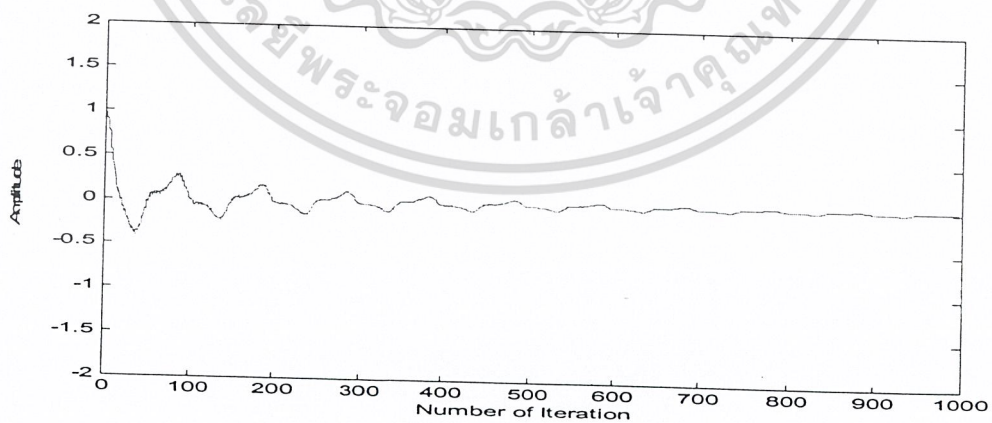
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก) สัญญาณเบสแบนด์ซึ่งถูกรบกวน

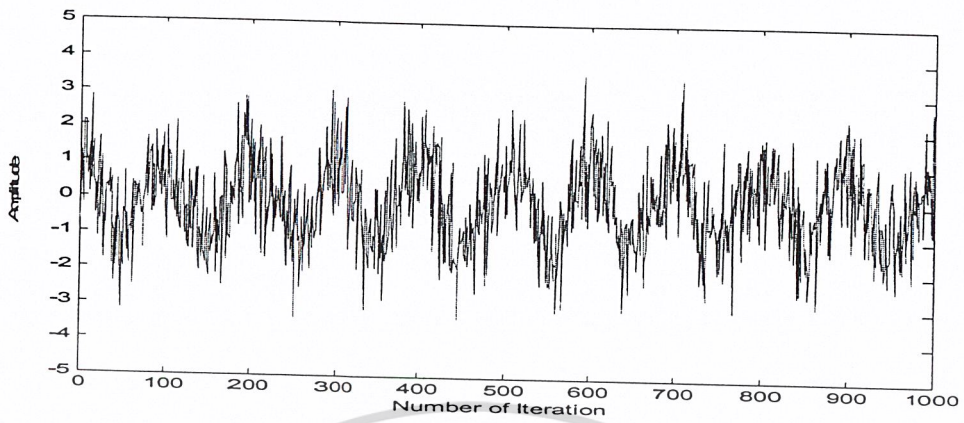


ข) สัญญาณที่กรองได้

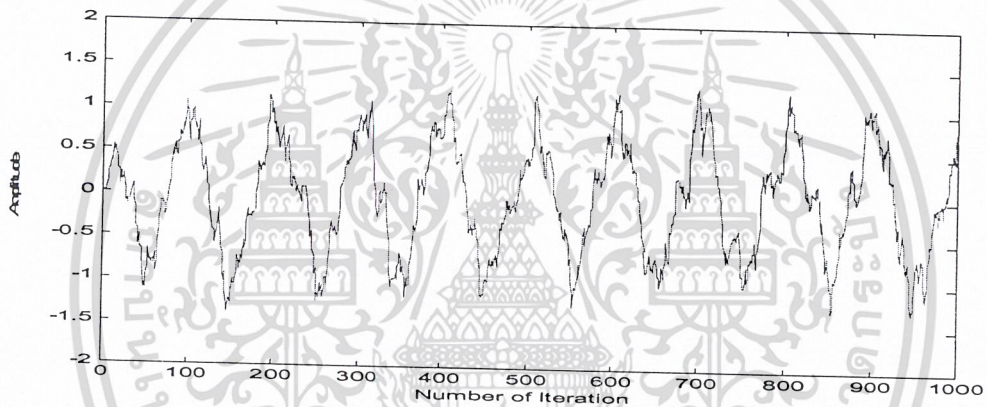


ค) สัญญาณความผิดพลาด

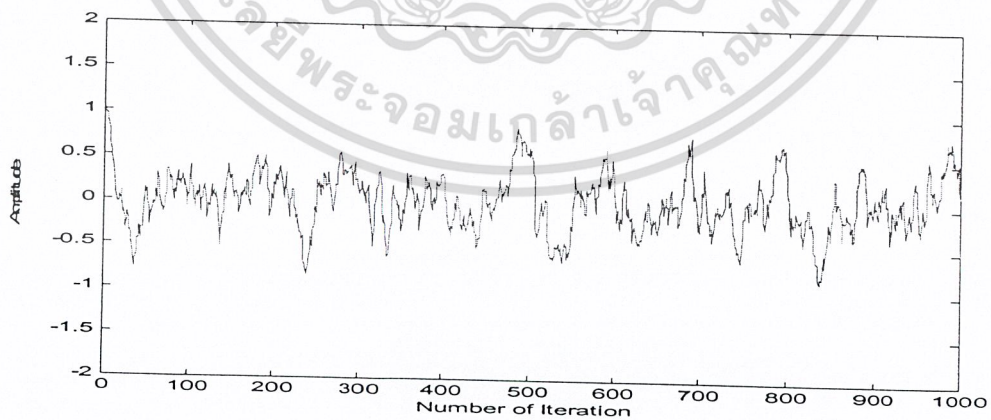
รูปที่ 4.13 ผลตอบสนองต่างๆ ของตัวกรองปรับตัวแบบไอไออาร์อันดับ 10 ต่อ สัญญาณรบกวนรูปไซน์
 เอกสารนี้เป็นเอกสารที่นำไปสำหรับการใช้เฉพาะเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ความถี่ 3 kHz ความถี่การซิกตัวอย่าง 8 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz
 ไม่ว่าจะกรณีใดๆ ห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
 ขนาดสเกลเท่ากับ 0.005



ก) สัญญาณเบสแบนด์ซึ่งถูกรบกวน



ข) สัญญาณที่กรองได้



ค) สัญญาณความผิดพลาด

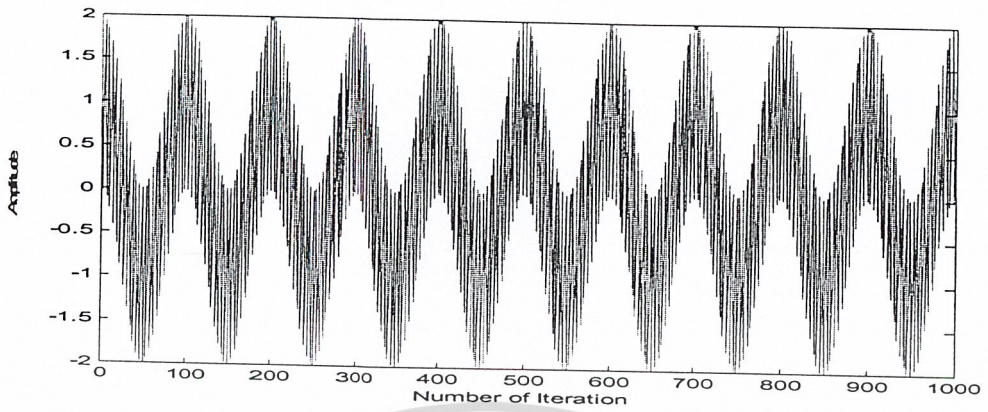
รูปที่ 4.14 ผลตอบสนองต่างๆ ของตัวกรองปรับควมแบบไอโออาร์อันดับ 10 ต่อ สัญญาณรบกวนกลุ่ม
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่บนด้านการค้า
ความถี่การซึกตัวอย่าง 8 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz ขนาดสเตปเท่ากับ 0.005
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.11 ถึง 4.14 พบว่า ผลตอบสนองต่อความถี่ต่างๆของตัวกรองปรับตัวแบบ ไอไออาร์ มีผลตอบสนองในการกำจัดสัญญาณรบกวนได้ดี และสังเกตได้ว่าจะมีการลู่เข้าที่ดีขึ้น เมื่อสัญญาณรบกวนมีความถี่แตกต่างกับสัญญาณมากขึ้น

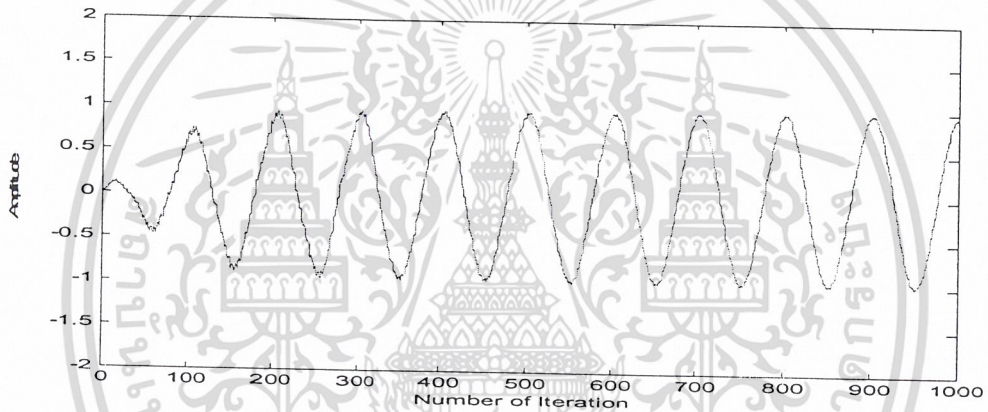


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

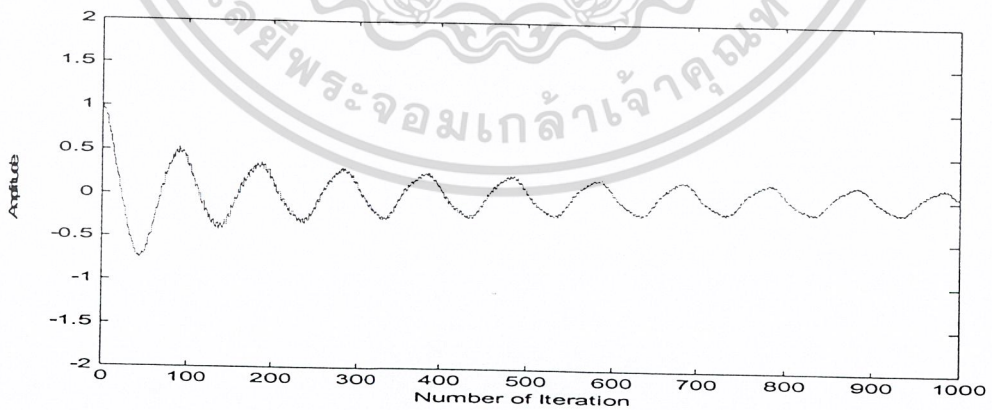
4.2.2.2 ผลตอบสนองต่อขนาดสเตปต่างๆของตัวกรองปรับตัวแบบไอไออาร์



ก) สัญญาณเบสแบนด์ซึ่งถูกรบกวน

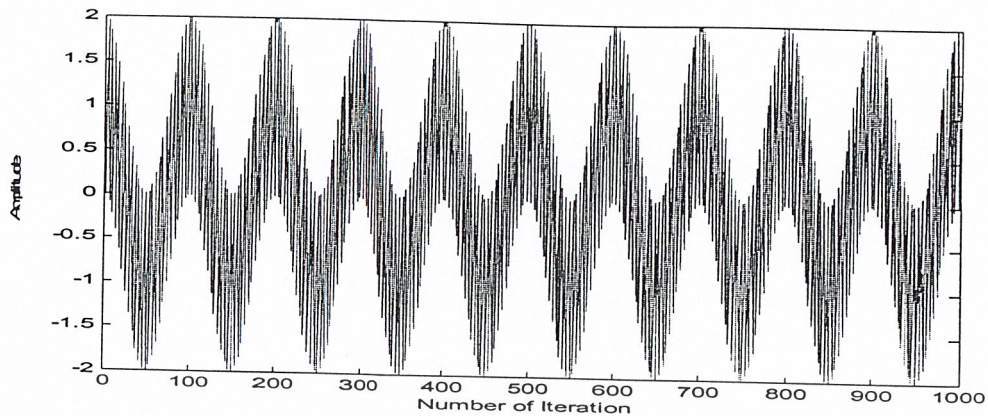


ข) สัญญาณที่กรองได้

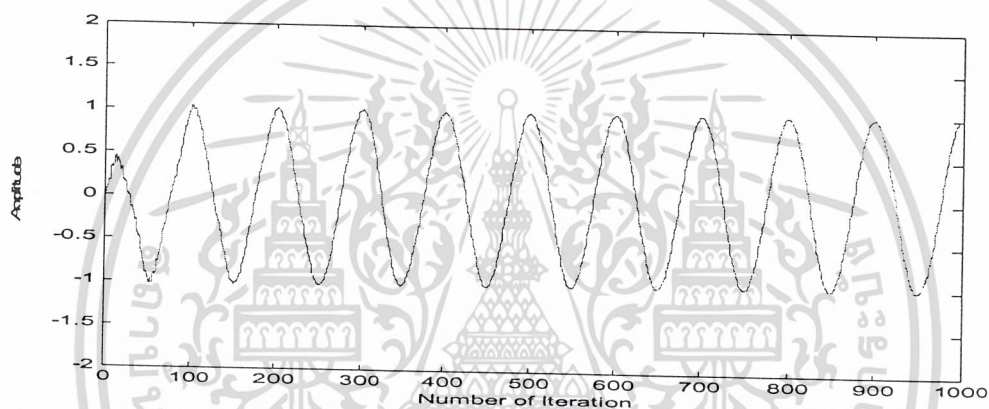


ค) สัญญาณความผิดพลาด

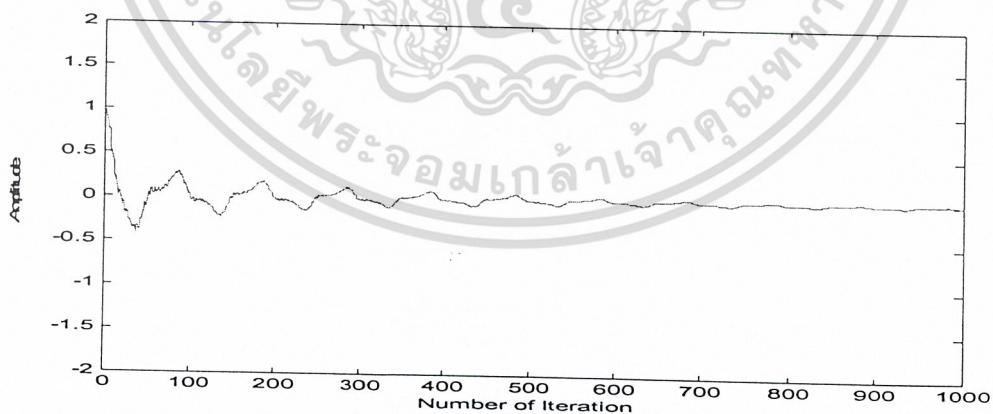
รูปที่ 4.15 ผลตอบสนองต่างๆ ของตัวกรองปรับตัวแบบไอไออาร์อันดับ 10 ต่อ สัญญาณรบกวนรูปไซน์
 เอกสารนี้เป็นเอกสารที่ 2 kHz ความถี่การซีกตัวอย่าง 8 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz
 ไม่ว่าจะกรณีใดๆ ขนาดสเตปเท่ากับ 0.001 แปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก) สัญญาณเบสแบนด์ซึ่งถูกรบกวน



ข) สัญญาณที่กรองได้



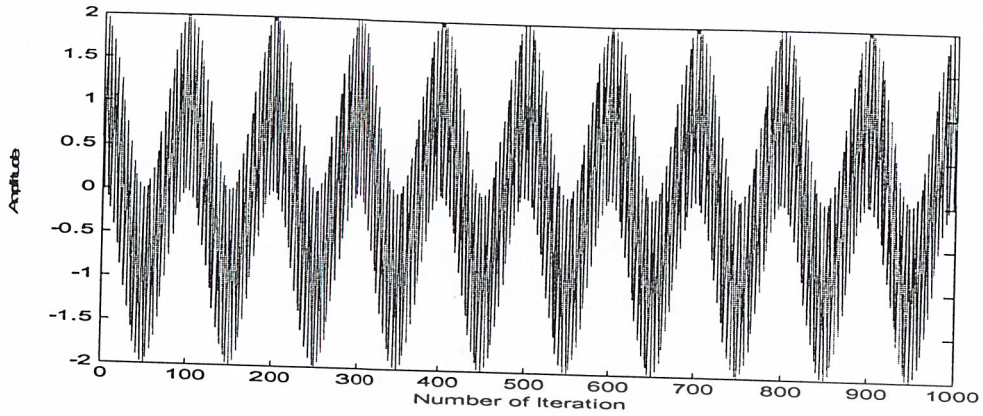
ค) สัญญาณความผิดพลาด

รูปที่ 4.16 ผลตอบสนองต่างๆ ของตัวกรองปรับตัวแบบไอไออาร์อันดับ 10 ต่อ สัญญาณรบกวนรูปไซน์

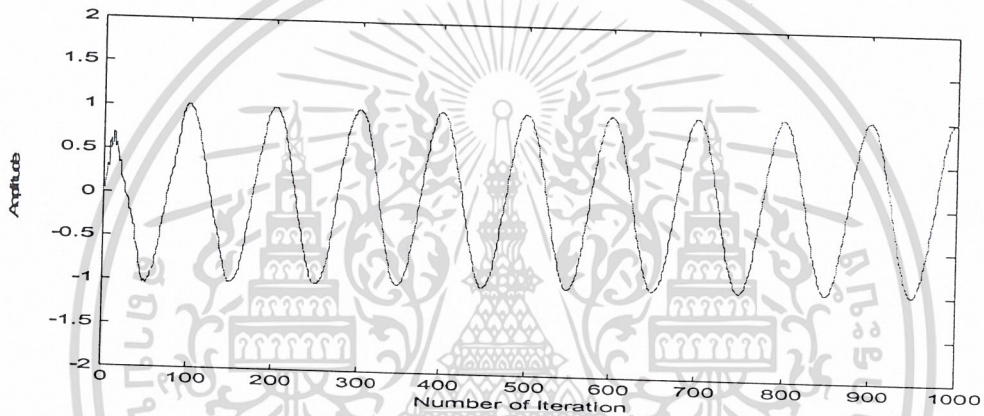
ความถี่ 2 kHz ความถี่การซัดตัวอย่าง 8 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

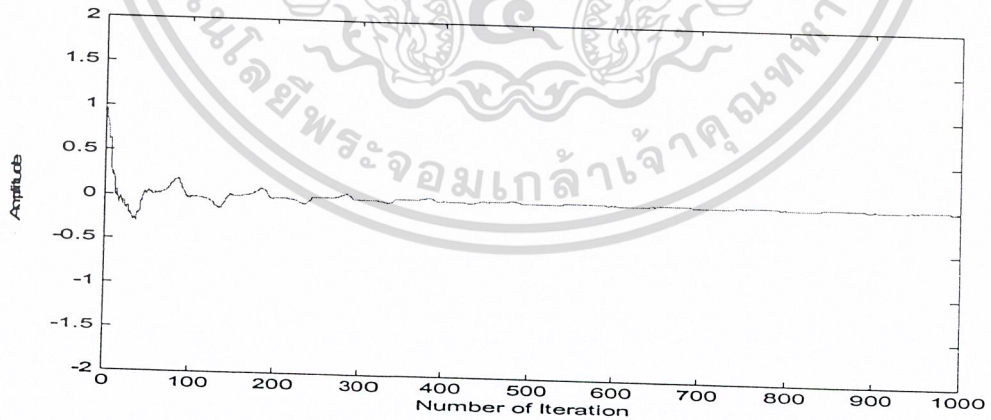
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก) สัญญาณเบสแบนด์ซึ่งถูกรบกวน

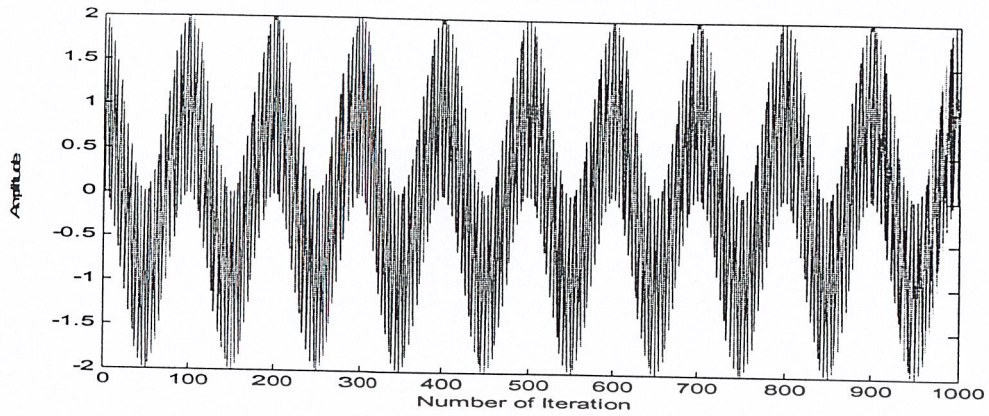


ข) สัญญาณที่กรองได้



ค) สัญญาณความผิดพลาด

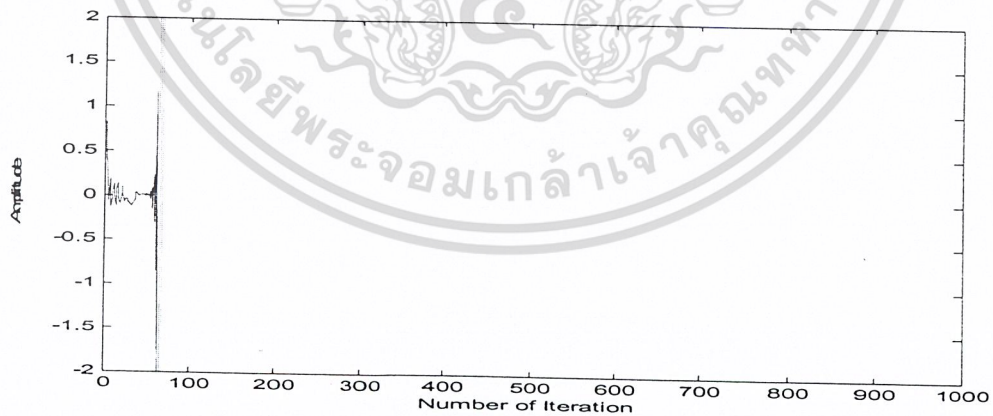
รูปที่ 4.17 ผลตอบสนองต่างๆ ของตัวกรองปรับตัวแบบไอไออาร์อันดับ 10 ต่อ สัญญาณรบกวนรูปไซน์ ความถี่ 2 kHz ความถี่การซีกตัวอย่าง 8 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz เอกสารนี้เป็นเอกสารต้นฉบับเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก) สัญญาณเบสแบนด์ซึ่งถูกรบกวน



ข) สัญญาณที่กรองได้



ค) สัญญาณความผิดพลาด

รูปที่ 4.18 ผลตอบสนองต่างๆ ของตัวกรองปรับตัวแบบไอโออาร์อันดับ 10 ต่อ สัญญาณรบกวนรูปไซน์

ความถี่ 2 kHz ความถี่การซัดตัวอย่าง 8 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz

เอกสารนี้เป็นเอกสารต้นฉบับที่จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

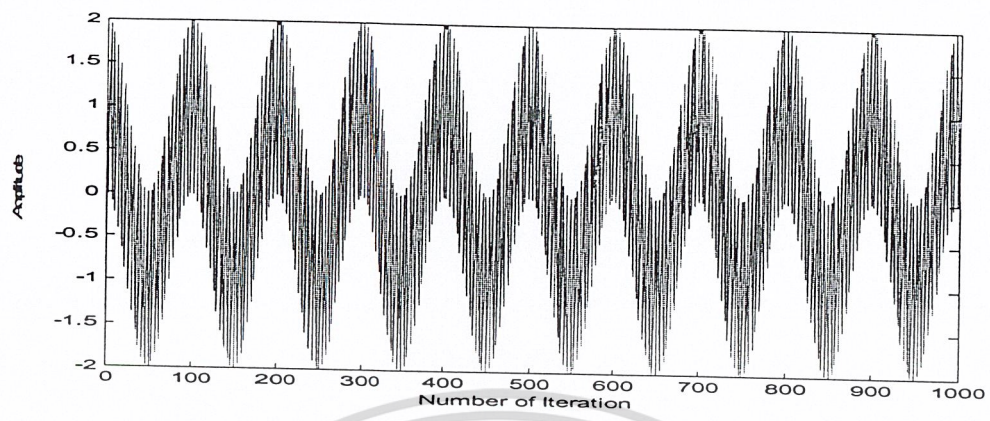
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.15 ถึง 4.18 พบว่า ผลตอบสนองต่อขนาดสเตปต่างๆของตัวกรองปรับตัวแบบไอไออาร์ มีผลตอบสนองในการกำจัดสัญญาณรบกวนได้ดี และสังเกตได้ว่าจะมีการลู่เข้าที่ดีขึ้น เมื่อขนาดสเตปใหญ่มากขึ้น อย่างไรก็ตามหากขนาดสเตปใหญ่มากเกินไปอาจทำให้การลู่เข้าเร็วลงได้เนื่องจากการปรับสัมประสิทธิ์ในแต่ละครั้งมากเกินไปและอาจเกิดการไม่เสถียรของตัวกรองได้ เนื่องจากโพลีเอินออกจากรวมหนึ่งหน่วย

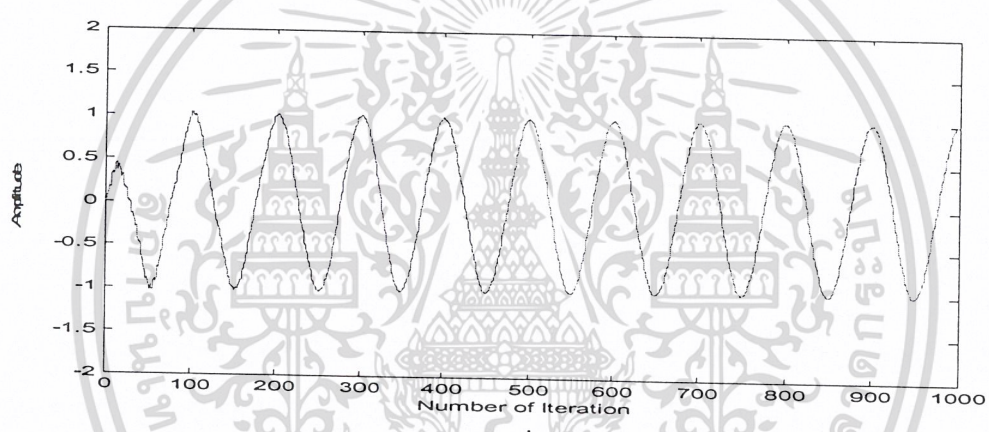


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

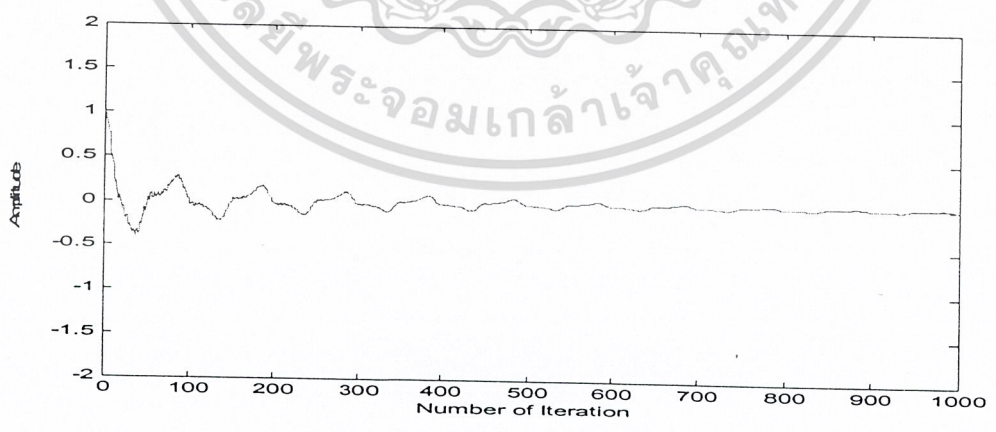
4.2.2.3 ผลตอบสนองต่ออันดับต่างๆของตัวกรองปรับตัวแบบไอไออาร์



ก) สัญญาณเบสแบนด์ซึ่งถูกรบกวน

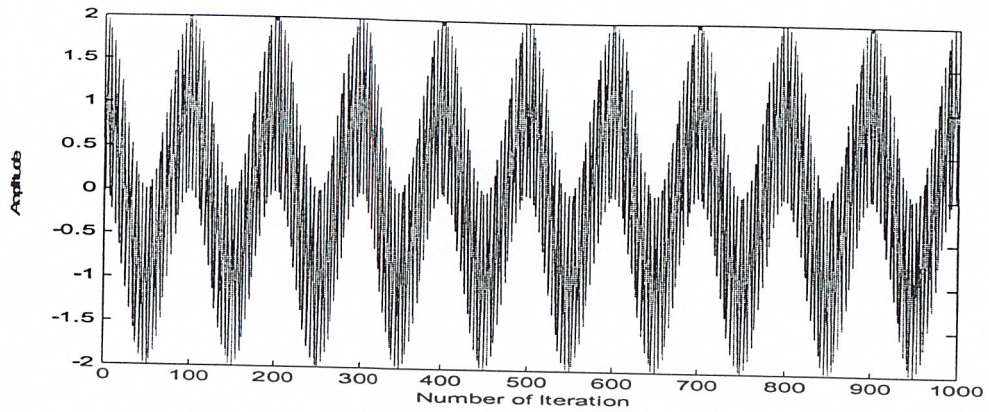


ข) สัญญาณที่กรองได้

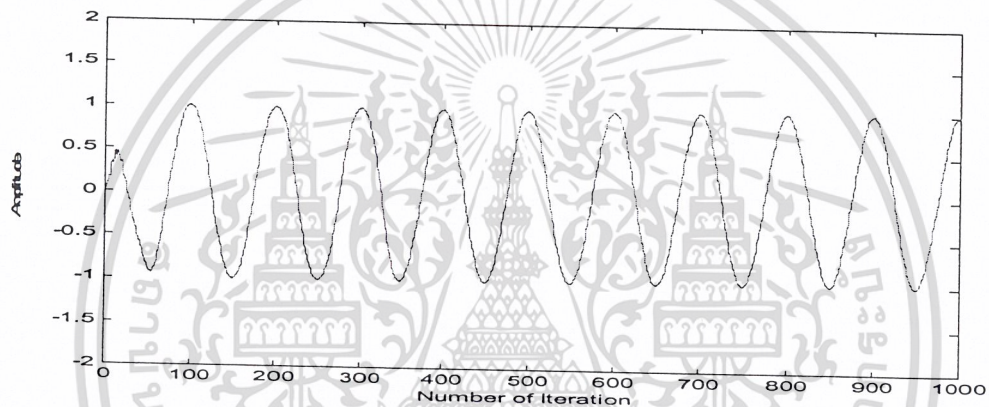


ค) สัญญาณความผิดพลาด

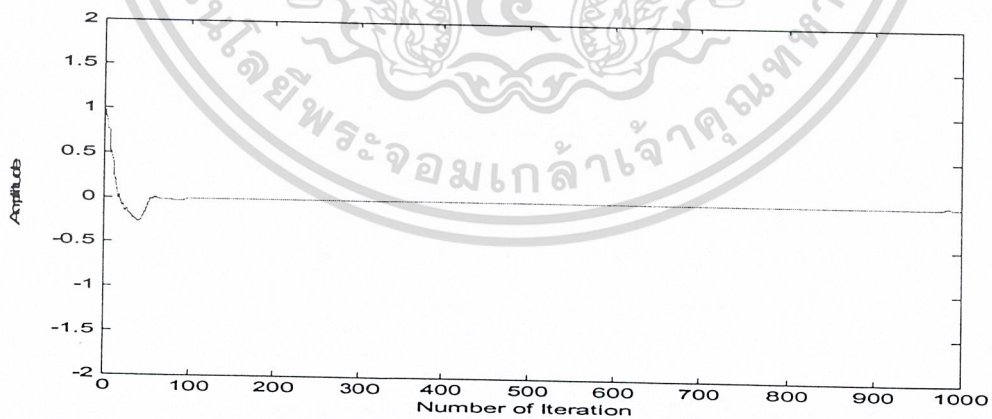
รูปที่ 4.19 ผลตอบสนองต่างๆ ของตัวกรองปรับตัวแบบไอไออาร์อันดับ 10 ต่อ สัญญาณรบกวนรูปไซน์ ความถี่ 2 kHz ความถี่การซิกคิวอย่าง 8 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz ไม่ว่าจะถี่ใดๆ ทั้งสิ้น ก็กลับให้ผลลัพธ์ที่ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ ขนาดสเกลเท่ากับ 0.005



ก) สัญญาณเบสแบนด์ซึ่งถูกรบกวน



ข) สัญญาณที่กรองได้



ค) สัญญาณความผิดพลาด

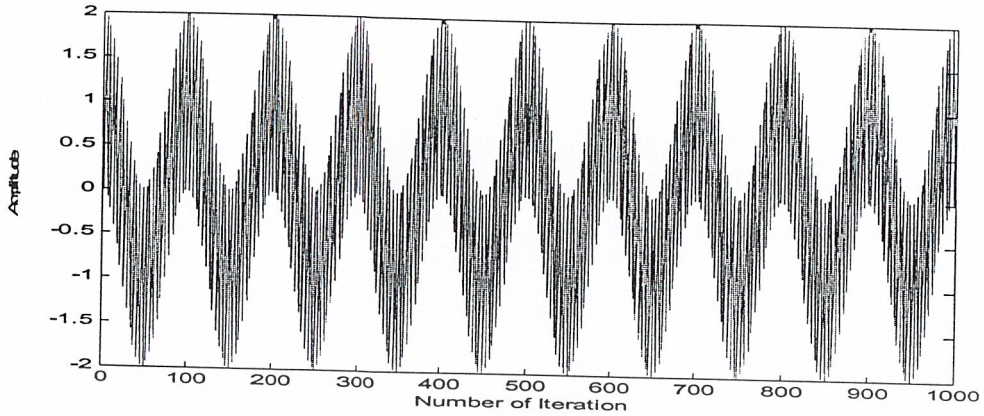
รูปที่ 4.20 ผลตอบสนองต่างๆ ของตัวกรองปรับตัวแบบไอไออาร์อันดับ 25 ต่อ สัญญาณรบกวนรูปไซน์

ความถี่ 2 kHz ความถี่การซัดตัวอย่าง 8 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz

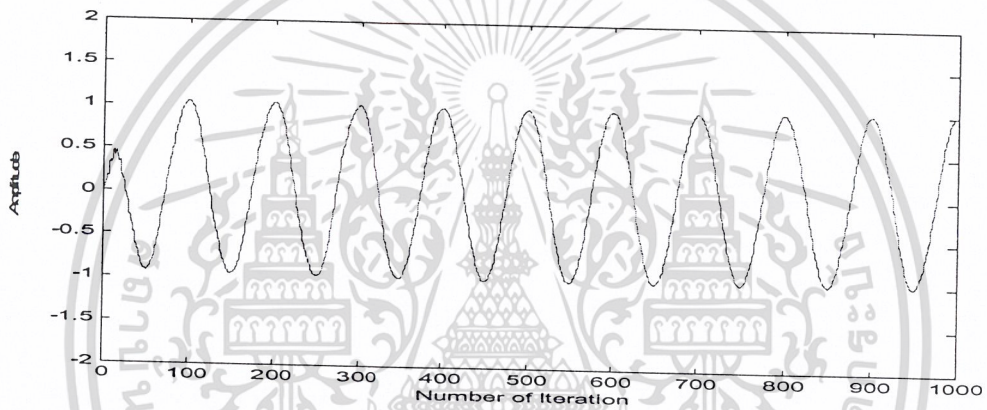
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ขนาดสเกลเท่ากับ 0.005

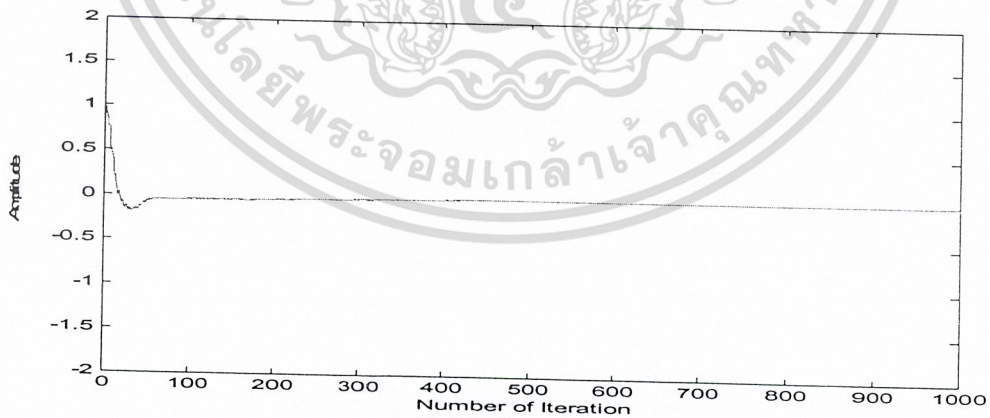
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก) สัญญาณเบสแบนด์ซึ่งถูกรบกวน



ข) สัญญาณที่กรองได้



ค) สัญญาณความผิดพลาด

รูปที่ 4.21 ผลตอบสนองต่างๆ ของตัวกรองปรับตัวแบบไอโอรันอันดับ 50 ต่อ สัญญาณรบกวนรูปไซน์

ความถี่ 2 kHz ความถี่การซีกตัวอย่าง 8 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต

ขนาดสเกลเท่ากับ 0.005

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

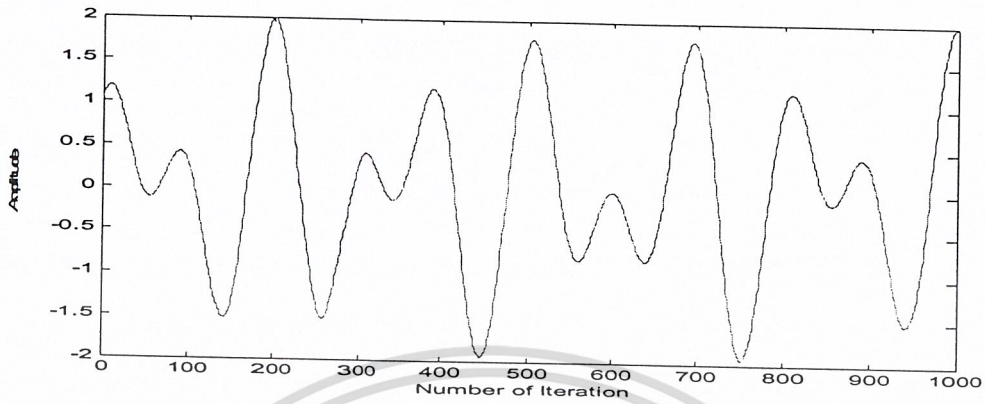
จากรูปที่ 4.19 ถึง 4.21 พบว่า ผลตอบสนองต่ออันดับต่างๆของตัวกรองปรับตัวแบบ ไอไออาร์ มีผลตอบสนองในการกำจัดสัญญาณรบกวนได้ดี และสังเกตได้ว่าจะมีการลู่อเข้าที่ดีขึ้น เมื่ออันดับของตัวกรองสูงมากขึ้น อย่างไรก็ตามอันดับของตัวกรองที่สูงเกินไปอาจทำให้การลู่อเข้าตลอดจนเสถียรภาพของระบบเลวลง เนื่องจากจำนวนสัมประสิทธิ์ที่เพิ่มขึ้นอาจทำให้ความไวของสัมประสิทธิ์ของตัวกรองเพิ่มขึ้นมากจนเกินไป



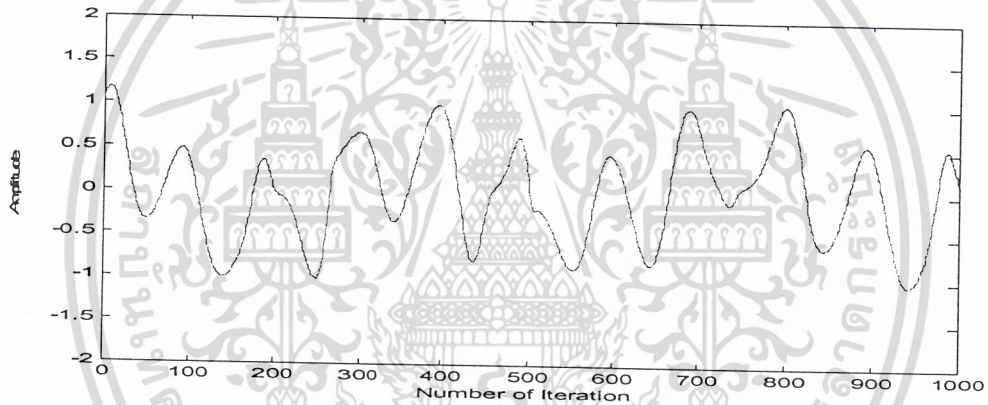
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 ผลตอบสนองต่างๆ ของตัวกำจัดสัญญาณรบกวน

4.2.3.1 ผลตอบสนองต่อความถี่ต่างๆ ของตัวกำจัดสัญญาณรบกวน



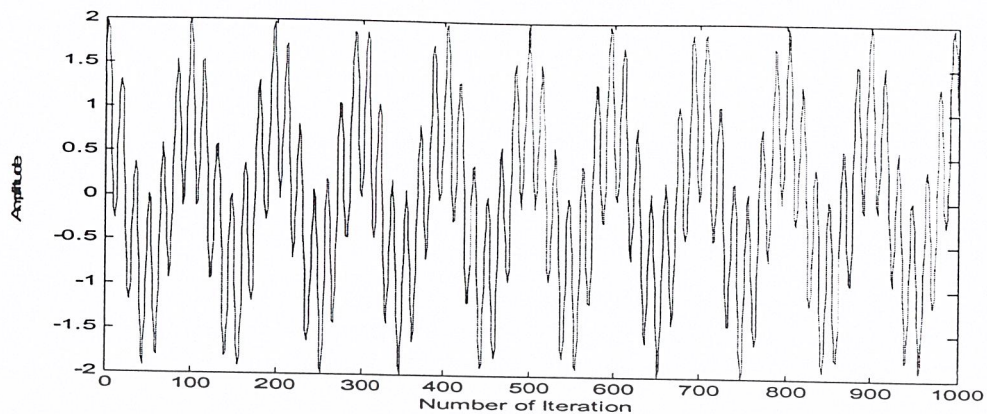
ก) สัญญาณเบสแบนด์ซึ่งถูกรบกวน



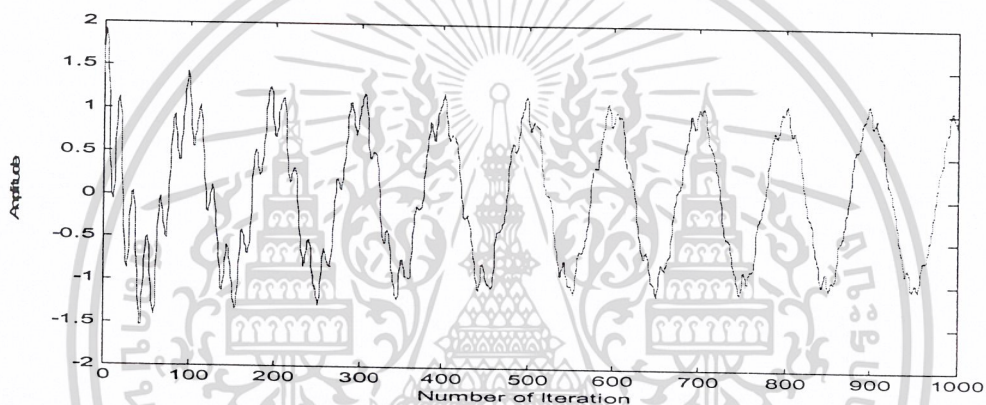
ข) สัญญาณถูกกำจัดสัญญาณรบกวนแล้ว

รูปที่ 4.22 ผลตอบสนองต่างๆ ของตัวกำจัดสัญญาณรบกวน ต่อ สัญญาณรบกวนรูปไซน์ ความถี่ 50 Hz ความถี่การซัดตัวอย่าง 8 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz ขนาดสเกลเท่ากับ 0.005

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



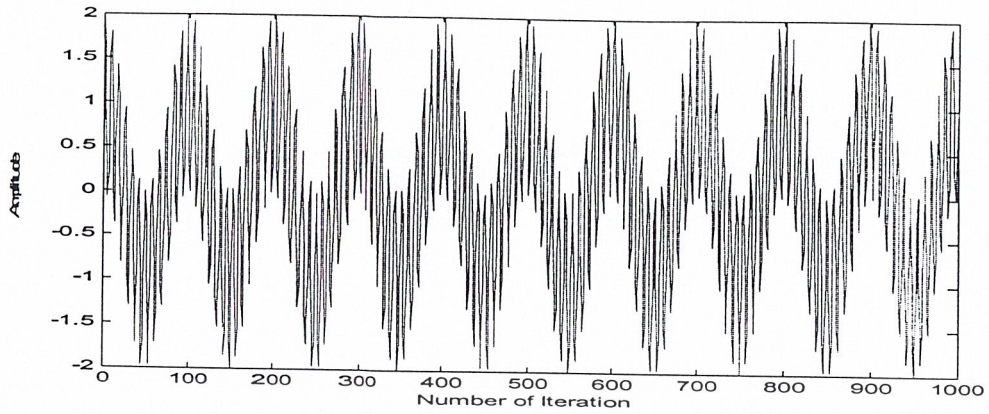
ก) สัญญาณเบสแบนด์ซึ่งถูกรบกวน



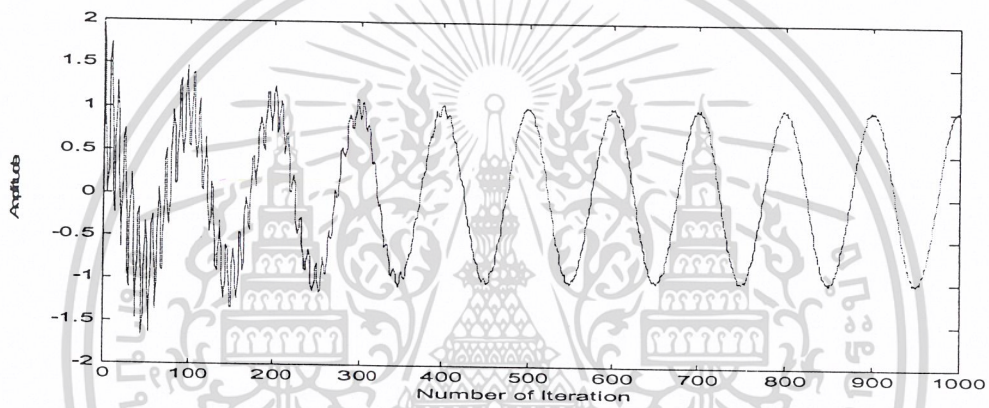
ข) สัญญาณถูกกำจัดสัญญาณรบกวนแล้ว

รูปที่ 4.23 ผลตอบสนองต่างๆ ของตัวกำจัดสัญญาณรบกวน ต่อ สัญญาณรบกวนรูปไซน์ ความถี่ 500 Hz ความถี่การซัดตัวอย่าง 8 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz ขนาดสเกลเท่ากับ 0.005

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



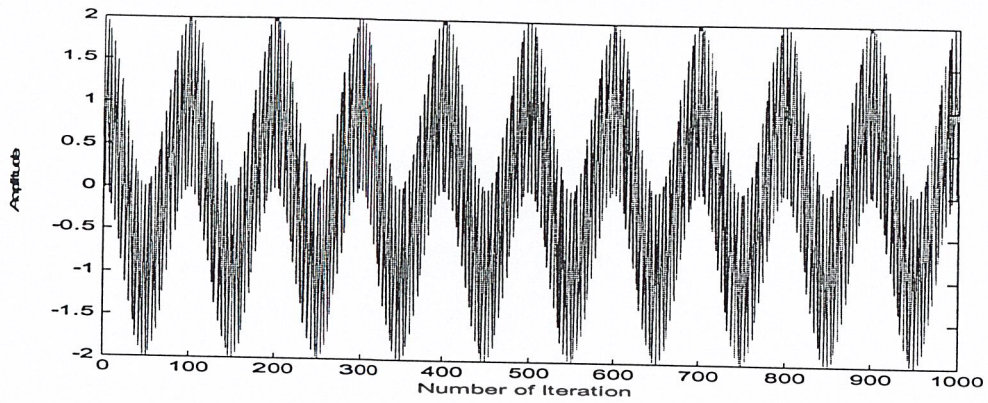
ก) สัญญาณเบสแบนด์ซึ่งถูกรบกวน



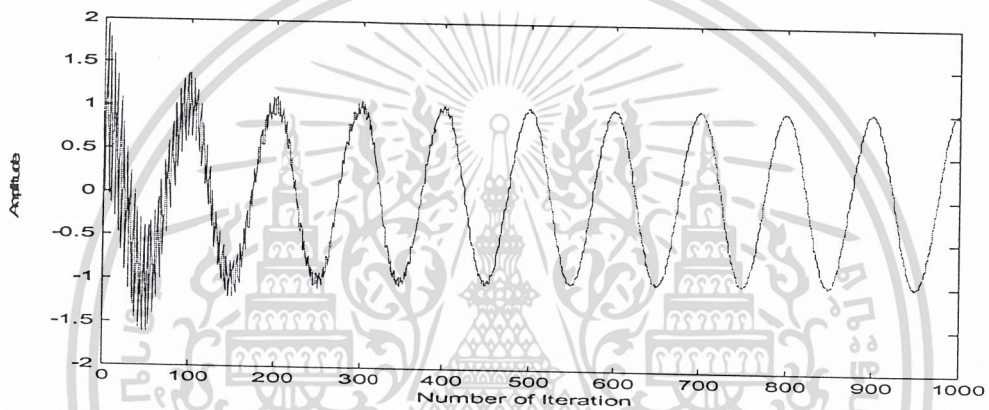
ข) สัญญาณถูกกำจัดสัญญาณรบกวนแล้ว

รูปที่ 4.24 ผลตอบสนองต่างๆ ของตัวกำจัดสัญญาณรบกวน ต่อ สัญญาณรบกวนรูปไซน์ ความถี่ 1 kHz ความถี่การซัดตัวอย่าง 8 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz ขนาดสเตปเท่ากับ 0.005

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



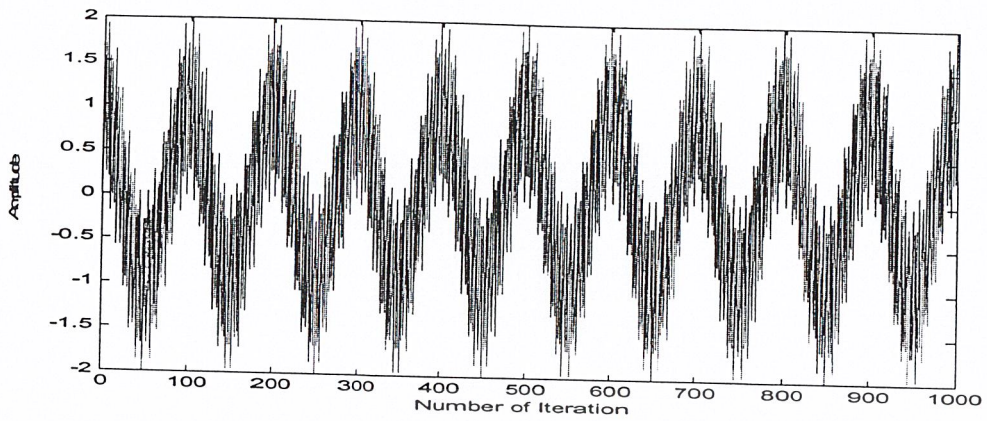
ก) สัญญาณเบสแบนด์ซึ่งถูกรบกวน



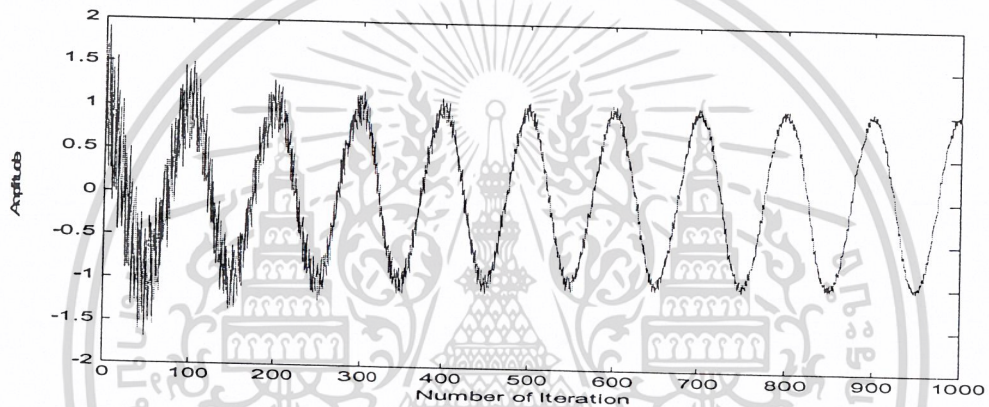
ข) สัญญาณถูกกำจัดสัญญาณรบกวนแล้ว

รูปที่ 4.25 ผลตอบสนองต่างๆ ของตัวกำจัดสัญญาณรบกวน ต่อ สัญญาณรบกวนรูปไซน์ ความถี่ 2 kHz ความถี่การซัดตัวอย่าง 8 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz ขนาดสเกลเท่ากับ 0.005

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



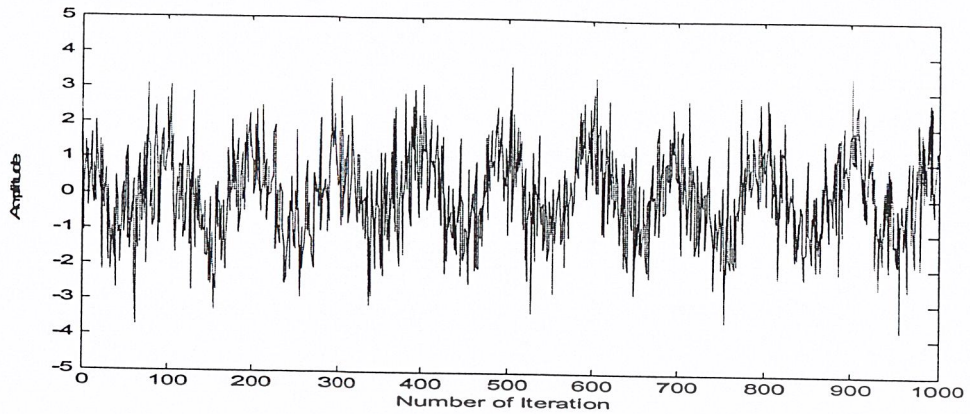
ก) สัญญาณเบสแบนด์ซึ่งถูกรบกวน



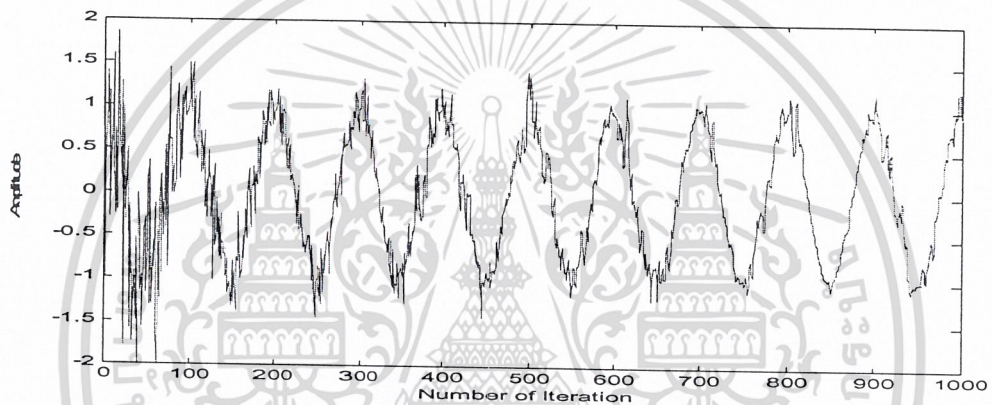
ข) สัญญาณถูกกำจัดสัญญาณรบกวนแล้ว

รูปที่ 4.26 ผลตอบสนองต่างๆ ของตัวกำจัดสัญญาณรบกวน ต่อ สัญญาณรบกวนรูปไซน์ ความถี่ 3 kHz ความถี่การซัดตัวอย่าง 8 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz ขนาดสเตปเท่ากับ 0.005

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก) สัญญาณเบสแบนด์ซึ่งถูกรบกวน



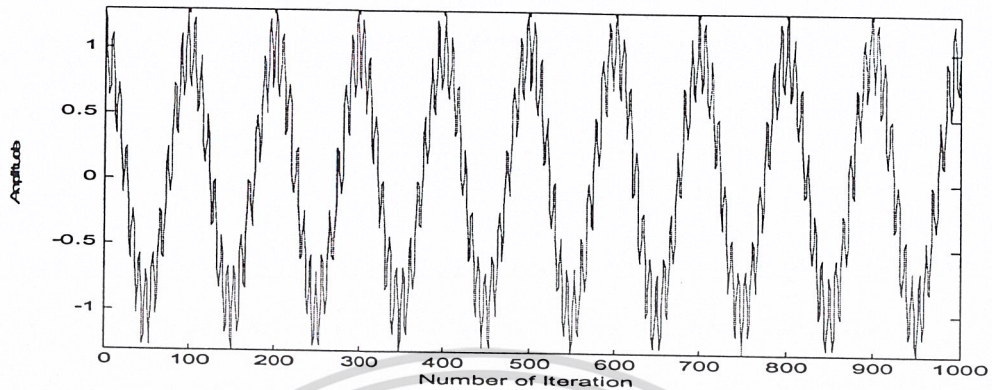
ข) สัญญาณถูกกำจัดสัญญาณรบกวนแล้ว

รูปที่ 4.27 ผลตอบสนองต่างๆ ของตัวกำจัดสัญญาณรบกวน ต่อ สัญญาณรบกวนสุ่ม
ความถี่การซีกตัวอย่าง 8 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz
ขนาดสเกลเท่ากับ 0.005

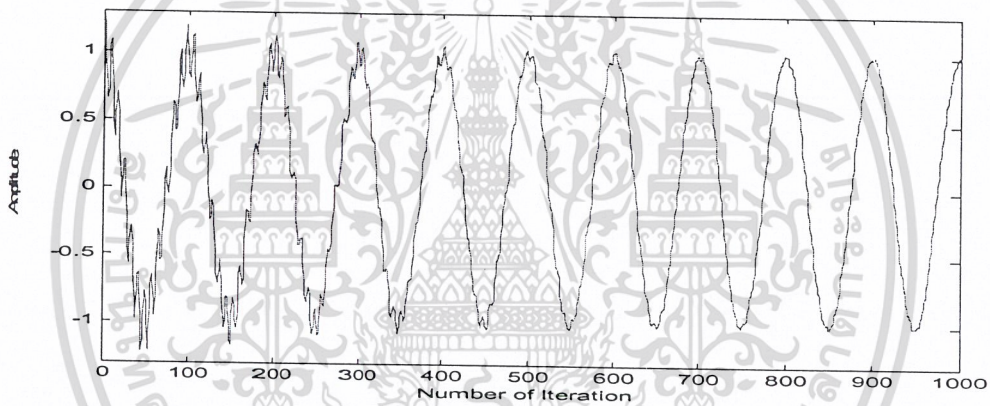
จากรูปที่ 4.22 ถึง 4.27 พบว่า ผลตอบสนองต่อความถี่ต่างๆของตัวกำจัดสัญญาณรบกวน มี
ผลตอบสนองในการกำจัดสัญญาณรบกวนได้ดี และสังเกตได้ว่าจะมีการลู่เข้าที่ดีขึ้น เมื่อสัญญาณรบกวน
มีความถี่แตกต่างกับสัญญาณมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3.2 ผลตอบสนองต่อสัญญาณรบกวนรูปไซน์ที่ขนาดแอมพลิจูดต่างๆ ของตัวกำจัดสัญญาณรบกวน



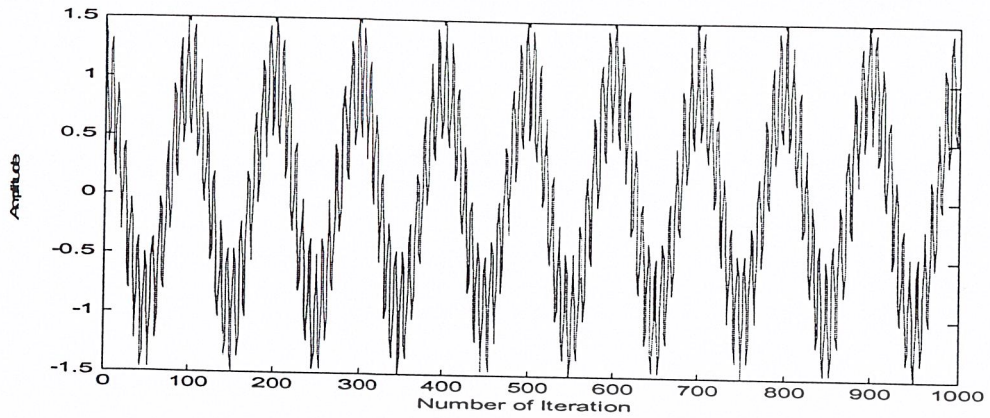
ก) สัญญาณเบสแบนด์ซึ่งถูกรบกวน



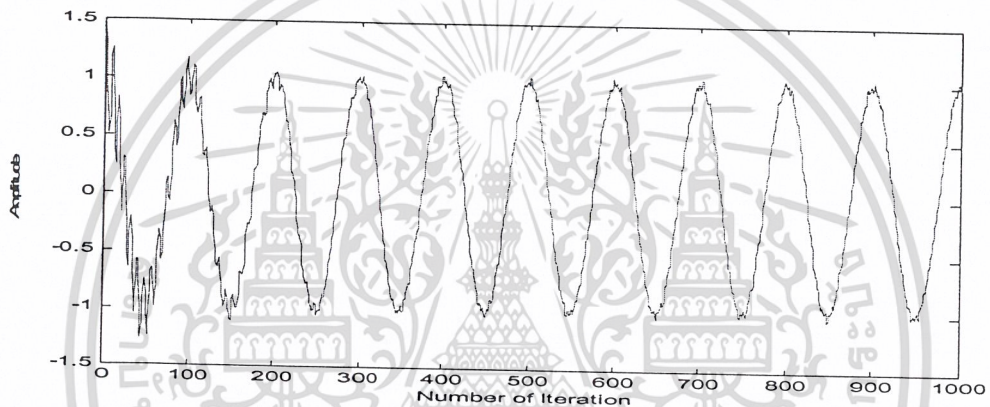
ข) สัญญาณถูกกำจัดสัญญาณรบกวนแล้ว

รูปที่ 4.28 ผลตอบสนองต่างๆ ของตัวกำจัดสัญญาณรบกวน ต่อ สัญญาณรบกวนรูปไซน์ ความถี่ 1 kHz ความถี่การซิกคิวอย่าง 8 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz ขนาดสเตปเท่ากับ 0.03 ขนาดแอมพลิจูดสัญญาณเบสแบนด์เท่ากับ 1 ขนาดแอมพลิจูดสัญญาณรบกวน เท่ากับ 0.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



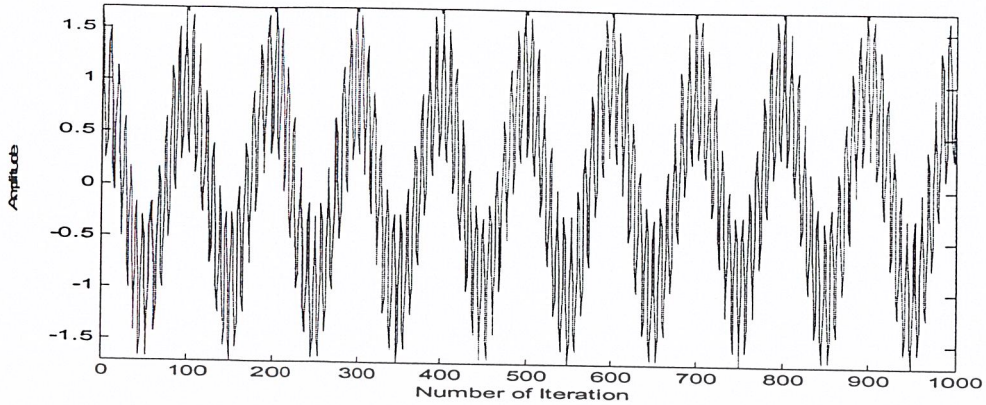
ก) สัญญาณเบสแบนด์ซึ่งถูกรบกวน



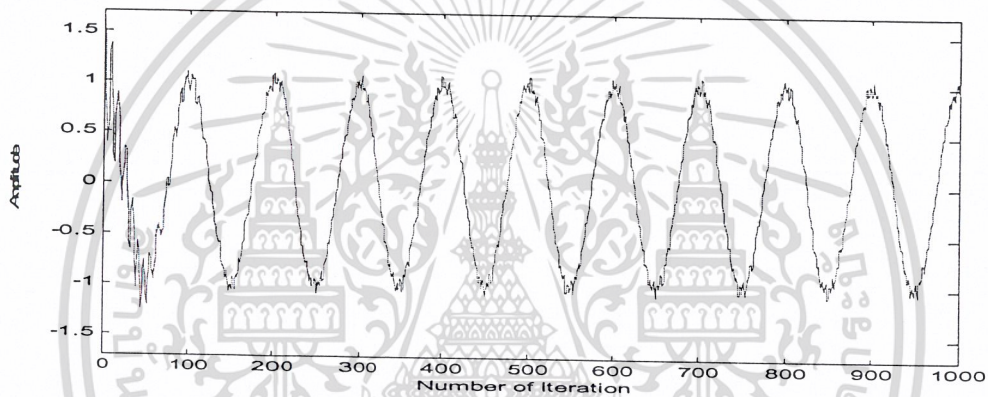
ข) สัญญาณถูกกำจัดสัญญาณรบกวนแล้ว

รูปที่ 4.29 ผลตอบสนองต่างๆ ของตัวกำจัดสัญญาณรบกวน ต่อ สัญญาณรบกวนรูปไซน์ ความถี่ 1 kHz ความถี่การซัดตัวอย่าง 8 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz ขนาดสเตปเท่ากับ 0.03 ขนาดแอมพลิจูดสัญญาณเบสแบนด์เท่ากับ 1 ขนาดแอมพลิจูดสัญญาณรบกวน เท่ากับ 0.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก) สัญญาณเบสแบนด์ซึ่งถูกรบกวน



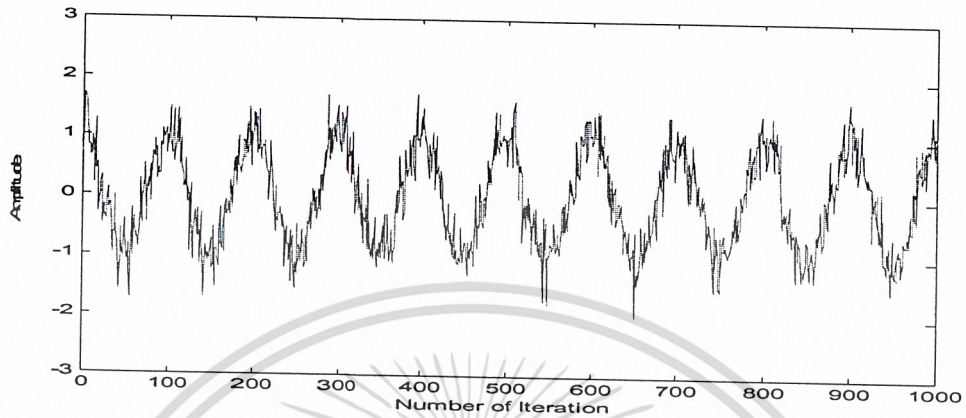
ข) สัญญาณถูกกำจัดสัญญาณรบกวนแล้ว

รูปที่ 4.30 ผลตอบสนองต่างๆ ของตัวกำจัดสัญญาณรบกวน ต่อ สัญญาณรบกวนรูปไซน์ ความถี่ 1 kHz ความถี่การซัดตัวอย่าง 8 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz ขนาดสเตปเท่ากับ 0.03 ขนาดแอมพลิจูดสัญญาณเบสแบนด์เท่ากับ 1 ขนาดแอมพลิจูดสัญญาณรบกวน เท่ากับ 0.7

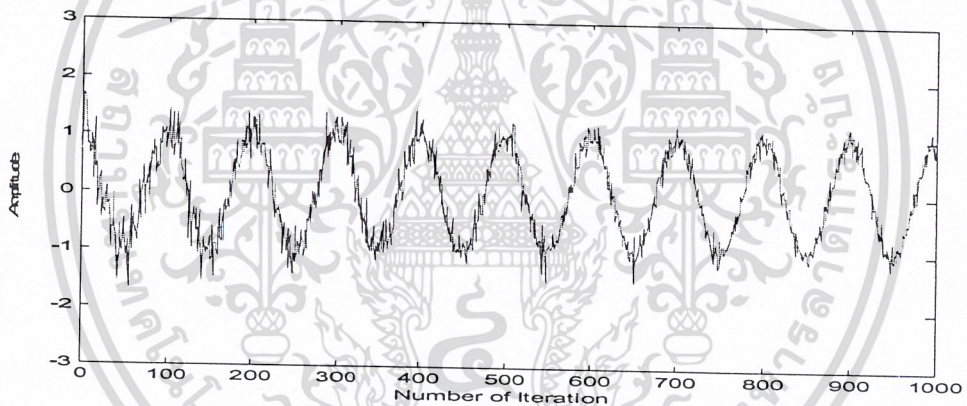
จากรูปที่ 4.28 ถึง 4.30 พบว่า ผลตอบสนองต่อสัญญาณรบกวนรูปไซน์ขนาดต่างๆ ของตัวกำจัดสัญญาณรบกวน มีผลตอบสนองในการกำจัดสัญญาณรบกวน ได้ดี และสังเกตได้ว่าจะมีการลู่เข้าที่ดีขึ้น เมื่อขนาดสัญญาณรบกวนใหญ่ขึ้น อย่างไรก็ตาม หากขนาดสัญญาณรบกวนใหญ่เกินไปอาจทำให้เสถียรภาพและการลู่เข้าเลวลงได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3.3 ผลตอบสนองต่อสัญญาณรบกวนสุ่มที่ขนาดแอมพลิจูดต่างๆ ของตัวกำจัดสัญญาณรบกวน



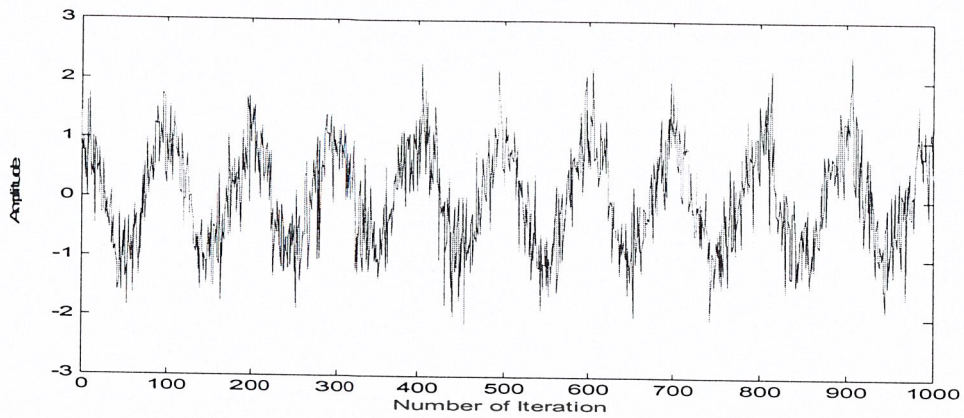
ก) สัญญาณเบสแบนด์ซึ่งถูกรบกวน



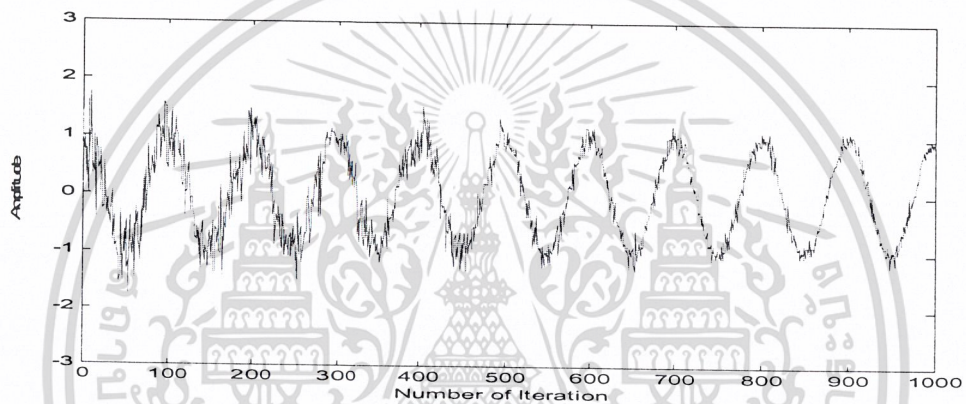
ข) สัญญาณถูกกำจัดสัญญาณรบกวนแล้ว

รูปที่ 4.31 ผลตอบสนองต่างๆ ของตัวกำจัดสัญญาณรบกวน ต่อ สัญญาณรบกวนสุ่ม ความถี่การซัดตัวอย่าง 8 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz ขนาดสเตปเท่ากับ 0.005 ขนาดแอมพลิจูดสัญญาณเบสแบนด์เท่ากับ 1 ขนาดแอมพลิจูดสัญญาณรบกวน เท่ากับ 0.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



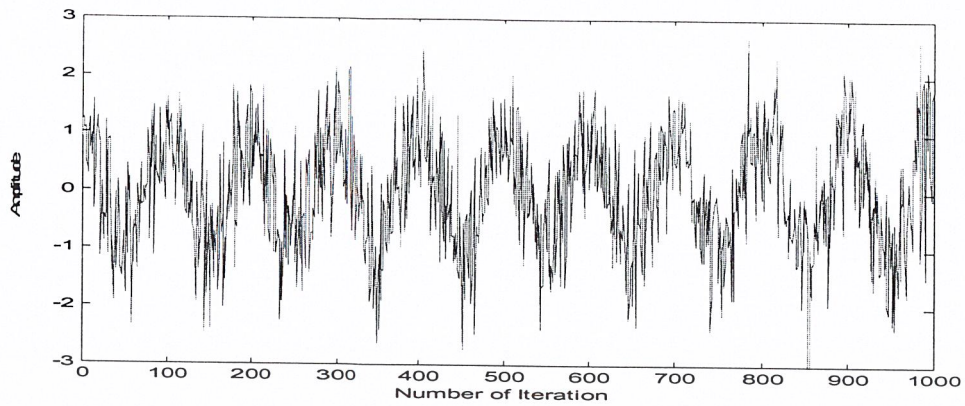
ก) สัญญาณเบสแบนด์ซึ่งถูกรบกวน



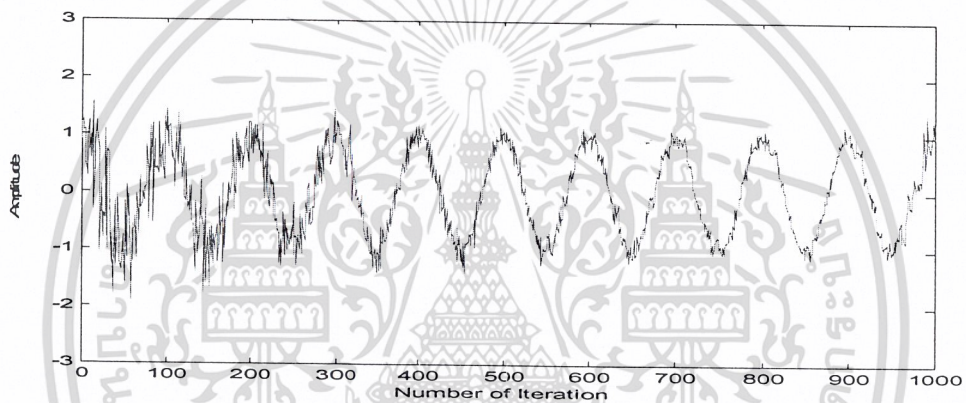
ข) สัญญาณถูกกำจัดสัญญาณรบกวนแล้ว

รูปที่ 4.32 ผลตอบสนองต่างๆ ของตัวกำจัดสัญญาณรบกวน ต่อ สัญญาณรบกวนสุ่ม ความถี่การซัดตัวอย่าง 8 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz ขนาดสเตปเท่ากับ 0.03 ขนาดแอมพลิจูดสัญญาณเบสแบนด์เท่ากับ 1 ขนาดแอมพลิจูดสัญญาณรบกวน เท่ากับ 0.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก) สัญญาณเบสแบนด์ซึ่งถูกรบกวน



ข) สัญญาณถูกกำจัดสัญญาณรบกวนแล้ว

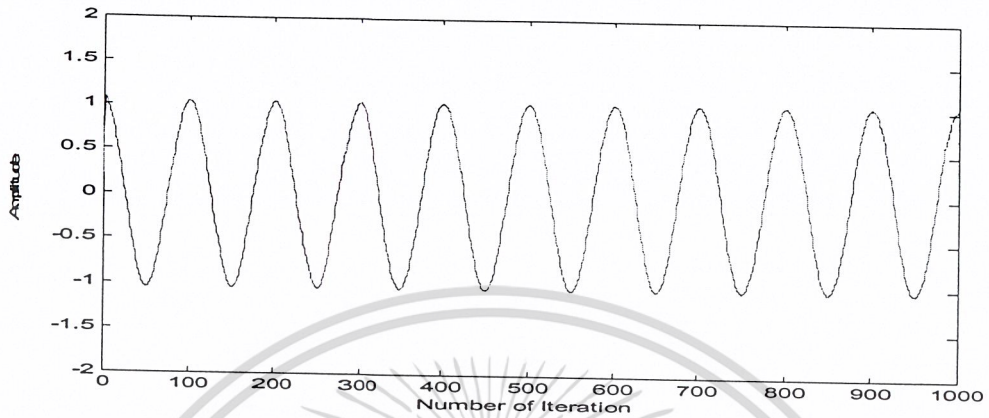
รูปที่ 4.33 ผลตอบสนองต่างๆ ของตัวกำจัดสัญญาณรบกวน ต่อ สัญญาณรบกวนสุ่ม ความถี่การซัดตัวอย่าง 8 kHz ความถี่สัญญาณเบสแบนด์ 80 Hz ขนาดสเตปเท่ากับ 0.03 ขนาดแอมพลิจูดสัญญาณเบสแบนด์เท่ากับ 1 ขนาดแอมพลิจูดสัญญาณรบกวน เท่ากับ 0.7

จากรูปที่ 4.31 ถึง 4.33 พบว่า ผลตอบสนองต่อสัญญาณรบกวนสุ่มขนาดต่างๆ ของตัวกำจัดสัญญาณรบกวน มีผลตอบสนองในการกำจัดสัญญาณรบกวนได้ดี และสังเกตได้ว่าจะมีการลู่อเข้าที่ดีขึ้นเมื่อสัญญาณรบกวนมีขนาดใหญ่ขึ้น อย่างไรก็ตามขนาดสัญญาณรบกวนที่ใหญ่เกินไปอาจทำให้เสถียรภาพและการลู่อเข้าเลวลงได้

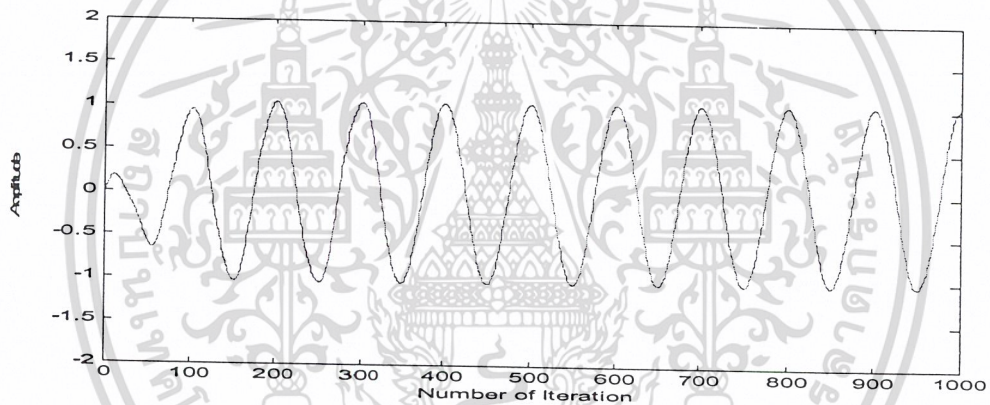
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4 ผลตอบสนองต่างๆ ของการใช้ตัวกรองปรับตัวหาคุณสมบัติของระบบที่ไม่ทราบ

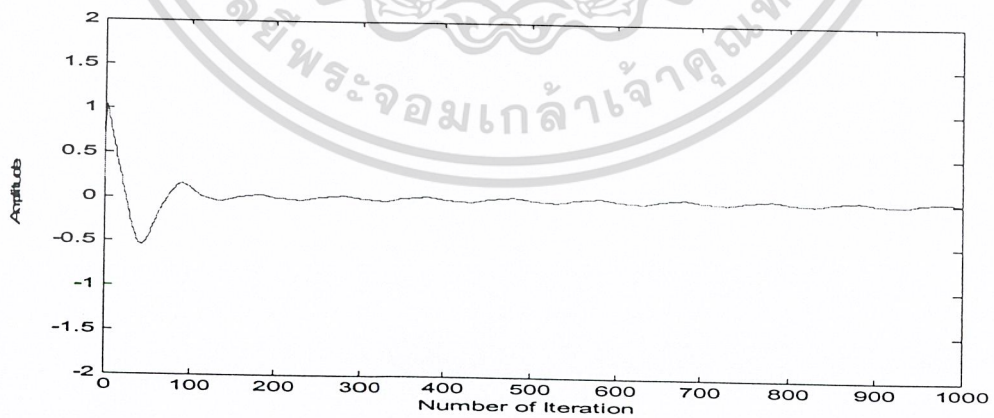
4.2.4.1 ผลตอบสนองต่อขนาดสเตปต่างๆ ของการใช้ตัวกรองปรับตัวหาคุณสมบัติของระบบที่ไม่ทราบ



ก) ผลตอบสนองจากระบบที่ไม่ทราบ

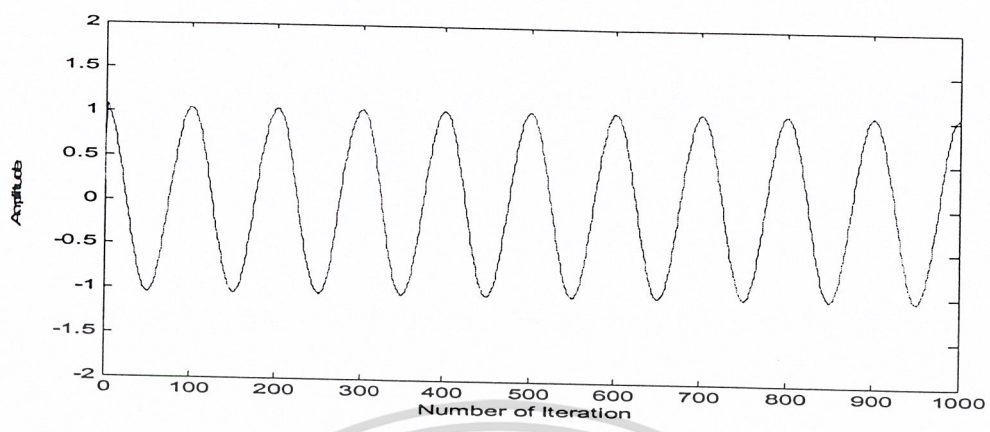


ข) ผลตอบสนองของตัวกรองปรับตัว

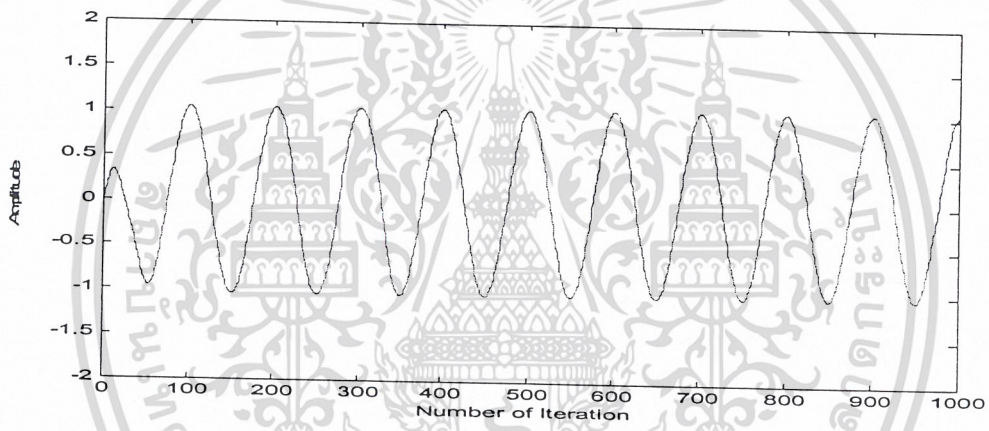


ค) ค่าความผิดพลาดของผลตอบสนอง

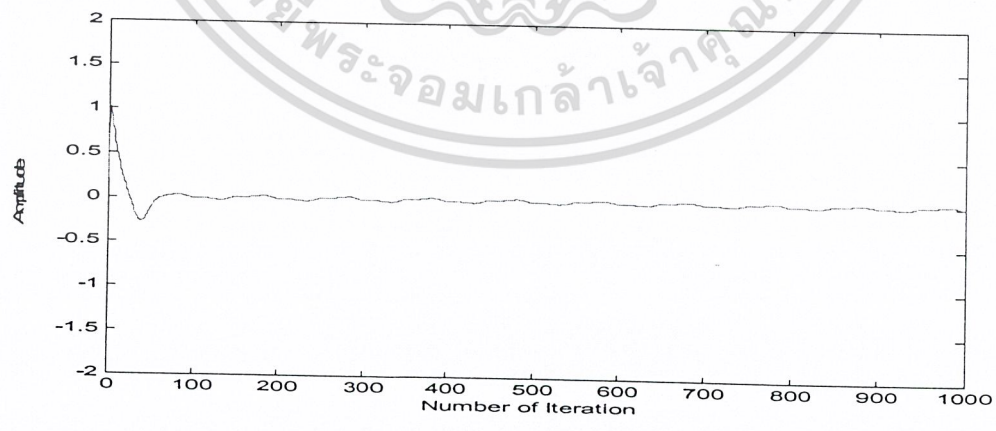
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เฉพาะเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 รูปที่ 4.34 ผลตอบสนองต่างๆ ของการใช้ตัวกรองปรับตัวหาคุณสมบัติของระบบที่ไม่ทราบ
 ไม่ว่าการณีใดๆ ใช้วิธีการนี้ใช้ค่าคงที่และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
 โดยใช้ขนาดสเตปเท่ากับ 0.005



ก) ผลตอบสนองจากระบบที่ไม่ทราบ

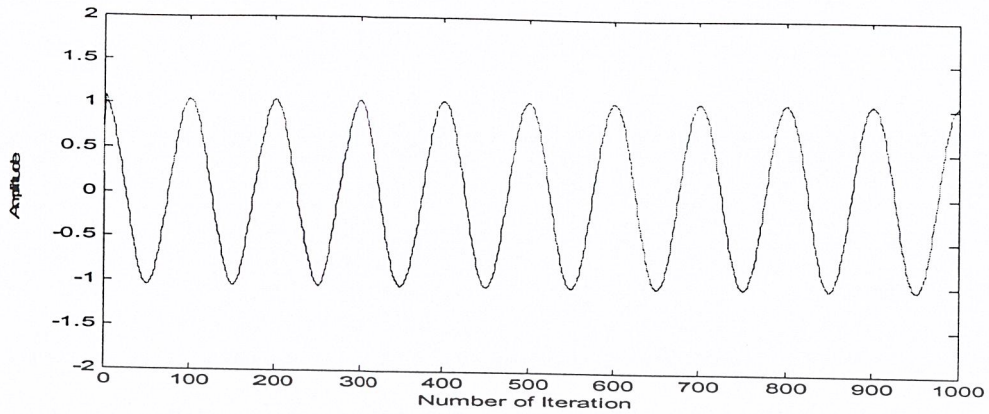


ข) ผลตอบสนองของตัวกรองปรับตัว

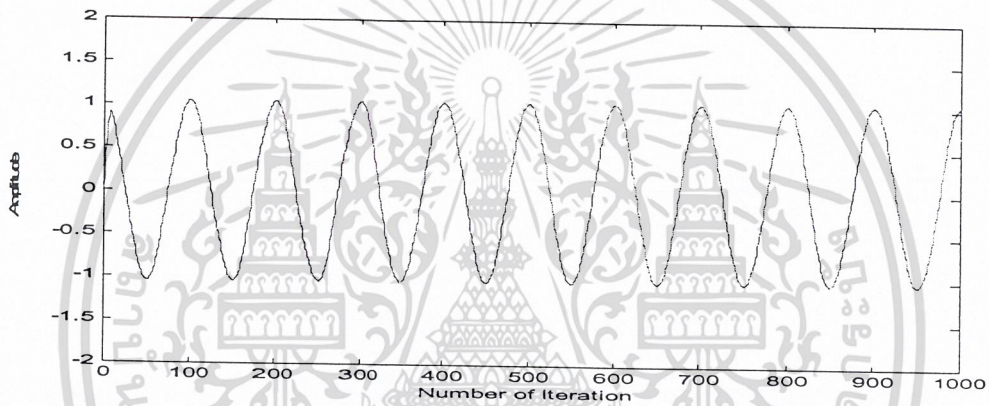


ค) ค่าความผิดพลาดของผลตอบสนอง

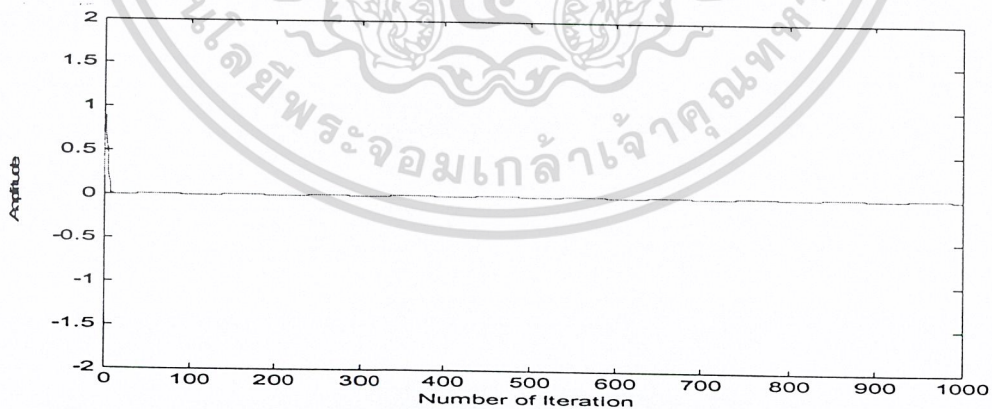
รูปที่ 4.35 ผลตอบสนองต่างๆ ของการใช้ตัวกรองปรับตัวหาคุณสมบัติของระบบที่ไม่ทราบ ประโยชน์ด้านการค้า
 เอกสารเป็นเอกสารที่สงวนลิขสิทธิ์หรือที่เรขาคณิตของเอกสารนี้ไม่ได้มีอยู่จริง
 โดยใช้นาฬิกาเทียบกับ 0.01
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้เผยแพร่และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก) ผลตอบสนองจากระบบที่ไม่ทราบ



ข) ผลตอบสนองของตัวกรองปรับตัว

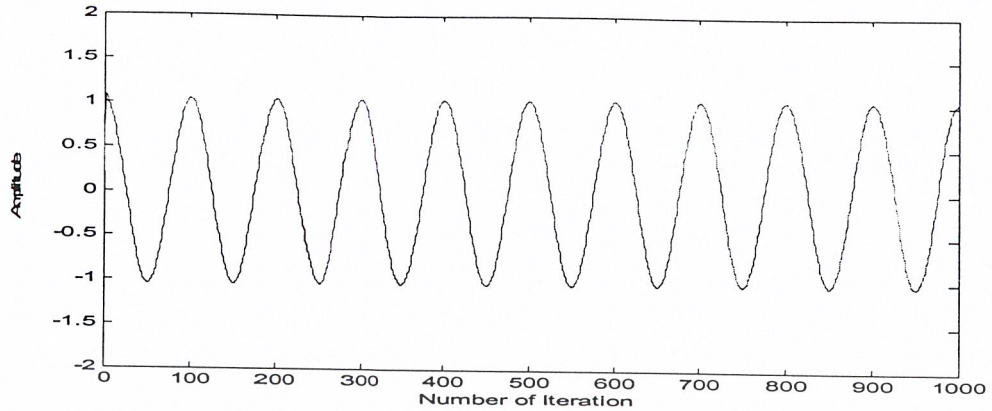


ค) ค่าความผิดพลาดของผลตอบสนอง

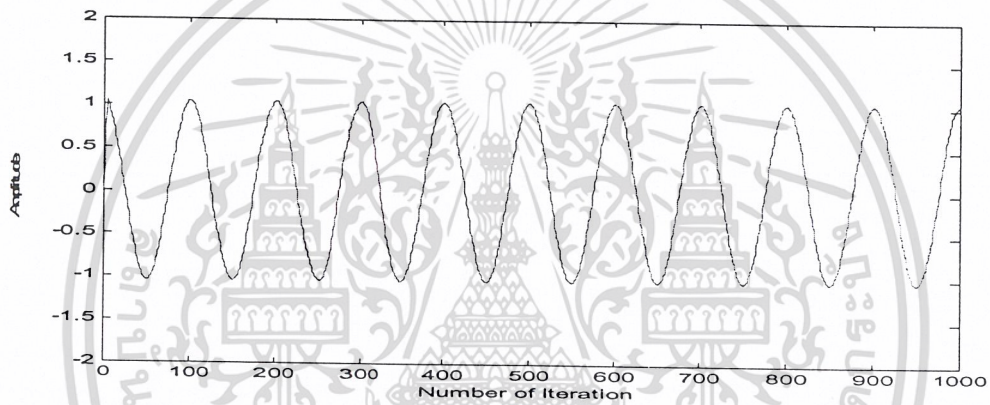
รูปที่ 4.36 ผลตอบสนองต่างๆ ของการใช้ตัวกรองปรับตัวหาคุณสมบัติของระบบที่ไม่ทราบ

โดยใช้ขนาดสเตปเท่ากับ 0.05

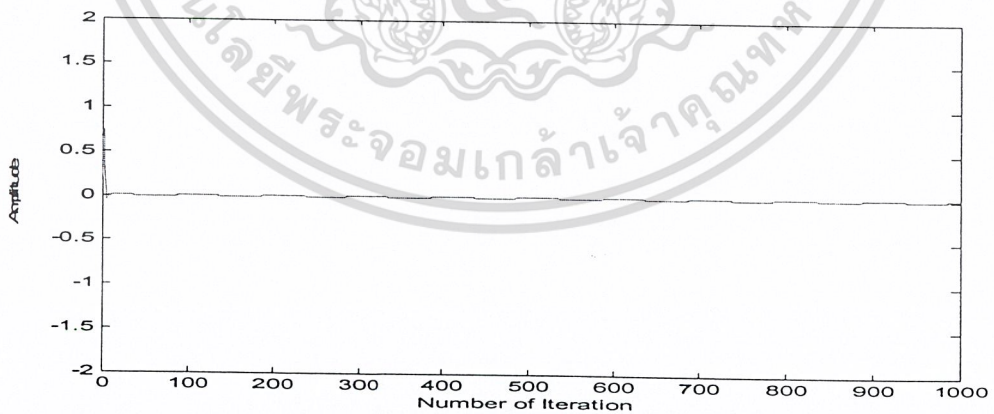
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก) ผลตอบสนองจากระบบที่ไม่ทราบ



ข) ผลตอบสนองของตัวกรองปรับตัว



รูปที่ 4.37 ผลตอบสนองต่างๆ ของการใช้ตัวกรองปรับตัวหาคุณสมบัติของระบบที่ไม่ทราบ โดยใช้ขนาดสเกลเท่ากับ 0.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การวัดคุณสมบัติโดยการสร้างในระบบประมวลผลสัญญาณ

4.3.1 การวัดคุณสมบัติตัวกรองปรับตัวแบบเอพีไออาร์ที่สร้างขึ้น

เนื่องจากโครงการนี้เน้นการสร้างตัวกรองปรับตัวแบบเอไออาร์ ดังนั้นจึงทำการสร้างตัวกรองปรับตัวแบบเอพีไออาร์ตามอัลกอริทึมที่ได้ออกแบบไว้เพียง 2 ตัวกรอง กล่าวคือ เป็นตัวกรองปรับตัวที่มีสัมประสิทธิ์ป้อนกลับจำนวน 50 และ 100 ตามลำดับ โดยจะทำการวัดสัญญาณที่กรองได้จากสัญญาณอินพุทซึ่งเป็นสัญญาณเบสแบนด์รูปไซน์ที่ถูกรบกวนด้วยสัญญาณรบกวนสุ่ม (Noise) และสัญญาณเบสแบนด์รูปไซน์ที่ถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่สูง เนื่องจากตัวกรองปรับตัวมีการทำงานเปลี่ยนไปตามเวลา ดังนั้นคุณสมบัติหนึ่งที่สำคัญคือ ผลตอบสนองชั่วคราว (Transient Response) แต่เนื่องจากข้อจำกัดของเครื่องมือ ในโครงการนี้จึงไม่ทำการวัดผลตอบสนองชั่วคราว แต่จะทำการวัดผลตอบสนองที่คงที่แล้ว (Steady State Response) ของตัวกรองที่ขนาดสเกล 8×10^{-8}

สำหรับสัญญาณที่ใช้ในการวัดคุณสมบัติมีดังนี้

- สัญญาณเบสแบนด์รูปไซน์ใช้สัญญาณ ความถี่ 80 Hz ขนาด 1.8 Volt
- สัญญาณรบกวนรูปไซน์ใช้สัญญาณ ความถี่ 500Hz ขนาด 1.06 Volt
- สัญญาณรบกวนรูปไซน์ใช้สัญญาณ ความถี่ 2 kHz ขนาด 1.06 Volt
- สัญญาณรบกวนรูปไซน์ใช้สัญญาณ ความถี่ 6 kHz ขนาด 1.06 Volt
- สัญญาณรบกวนสุ่ม

4.3.2 การวัดคุณสมบัติตัวกรองปรับตัวแบบเอไออาร์ที่สร้างขึ้น

โครงการนี้ทำการสร้างตัวกรองปรับตัวแบบเอไออาร์ตามอัลกอริทึมที่ออกแบบไว้ 5 ตัวกรอง กล่าวคือ เป็นตัวกรองปรับตัวที่มีสัมประสิทธิ์ป้อนกลับจำนวน 4, 8, 12, 16 และ 20 ตามลำดับ โดยจะทำการวัดสัญญาณที่กรองได้จากสัญญาณอินพุทซึ่งเป็นสัญญาณเบสแบนด์รูปไซน์ที่ถูกรบกวนด้วยสัญญาณรบกวนสุ่มและสัญญาณเบสแบนด์รูปไซน์ที่ถูกรบกวนด้วยสัญญาณรบกวนรูปไซน์ความถี่สูง โดยมีการปรับค่าขนาดสเกลต่างๆ เพื่อทำการเปรียบเทียบ และเนื่องจากขนาดสเกลมีผลต่อเสถียรภาพของตัวกรอง ดังนั้นในการเก็บผลจะทำการเปลี่ยนค่าสเกลต่างๆเพื่อหาค่าสเกลที่ทำให้ตัวกรองเสถียรตลอดจนทำการเก็บผลตอบสนองที่คงที่แล้วของตัวกรองที่ขนาดสเกลนั้น

สำหรับสัญญาณที่ใช้ในการวัดคุณสมบัติมีดังนี้

- สัญญาณเบสแบนด์รูปไซน์ใช้สัญญาณ ความถี่ 80 Hz ขนาด 1.8 Volt
- สัญญาณรบกวนรูปไซน์ใช้สัญญาณ ความถี่ 500Hz ขนาด 1.06 Volt
- สัญญาณรบกวนรูปไซน์ใช้สัญญาณ ความถี่ 2 kHz ขนาด 1.06 Volt
- สัญญาณรบกวนรูปไซน์ใช้สัญญาณ ความถี่ 6 kHz ขนาด 1.06 Volt
- สัญญาณรบกวนสุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากตัวกรองปรับตัวมีการทำงานเปลี่ยนไปตามเวลา ดังนั้นคุณสมบัติหนึ่งที่สำคัญคือ ผลตอบสนองชั่วคราว แต่เนื่องจากข้อจำกัดของเครื่องมือ ในโครงการนี้จึงไม่ทำการวัดผลตอบสนองชั่วคราว แต่จะทำการวัดผลตอบสนองที่คงที่แล้ว (Steady State Response) เท่านั้น

No of Tap \ Step Size	4×10^{-10}	8×10^{-10}	4×10^{-9}	8×10^{-9}	4×10^{-8}	8×10^{-8}
4	เสถียร	เสถียร	เสถียร	เสถียร	ไม่เสถียร	ไม่เสถียร
8	เสถียร	เสถียร	ไม่เสถียร	ไม่เสถียร	ไม่เสถียร	ไม่เสถียร
12	เสถียร	เสถียร	ไม่เสถียร	ไม่เสถียร	ไม่เสถียร	ไม่เสถียร
16	เสถียร	เสถียร	ไม่เสถียร	ไม่เสถียร	ไม่เสถียร	ไม่เสถียร
20	เสถียร	เสถียร	ไม่เสถียร	ไม่เสถียร	ไม่เสถียร	ไม่เสถียร

ตารางที่ 4.1 แสดงเสถียรภาพของตัวกรองที่ขนาดสเกลต่างๆ

จากผลข้างต้นพบว่าตัวกรองจะมีเสถียรภาพที่ขนาดสเกล 4×10^{-10} และ 8×10^{-10} ดังนั้นในการเก็บผลตอบสนองของตัวกรอง จะเก็บผลตอบสนองที่คงที่แล้ว ณ ความถี่สัญญาณรบกวนต่างๆที่ขนาดสเกลดังกล่าวเท่านั้น และเปรียบเทียบกันระหว่างขนาดสเกลตลอดจนจำนวนสัมประสิทธิ์ป้อนกลับ เพื่อแสดงให้เห็นถึงผลของขนาดสเกลและจำนวนสัมประสิทธิ์ป้อนกลับและความถี่ของสัญญาณรบกวน ที่มีต่อตัวกรอง

4.3.3 การวัดคุณสมบัติตัวกำจัดสัญญาณรบกวนที่สร้างขึ้น

โครงการนี้ทำการสร้างตัวกำจัดสัญญาณรบกวนเพื่อแสดงให้เห็นถึงการนำตัวกรองปรับตัวไปประยุกต์ใช้งาน ดังนั้นจึงทำการสร้างตัวกำจัดสัญญาณรบกวนโดยใช้ตัวกรองปรับตัวแบบไอไออาร์ตามอัลกอริทึมที่ได้ออกแบบไว้เพียง 1 ตัว โดยจะทำการวัดสัญญาณที่ได้กำจัดสัญญาณรบกวนแล้ว โดยสมมติให้จากสัญญาณอินพุตซึ่งเป็นสัญญาณเบสแบนด์รูปไซน์ที่ถูกรบกวนด้วยสัญญาณรบกวนรูปไซน์ ความถี่สูง และสัญญาณรบกวนซึ่งป้อนให้แก่ตัวกำจัดสัญญาณรบกวนเป็นสัญญาณรูปไซน์ความถี่สูง และเนื่องจากตัวกำจัดสัญญาณรบกวนมีการทำงานเปลี่ยนไปตามเวลา ดังนั้นคุณสมบัติตัวหนึ่งที่สำคัญคือ ผลตอบสนองชั่วคราว แต่เนื่องจากข้อจำกัดของเครื่องมือ ในโครงการนี้จึงไม่ทำการวัดผลตอบสนองชั่วคราว แต่จะทำการวัดผลตอบสนองที่คงที่แล้ว ของตัวกำจัดสัญญาณรบกวนที่ค่าสเกล 4×10^{-8}

สำหรับสัญญาณที่ใช้ในการวัดคุณสมบัติมีดังนี้

- สัญญาณเบสแบนด์รูปไซน์ใช้สัญญาณ ความถี่ 80 Hz ขนาด 1.8 Volt
- สัญญาณรบกวนรูปไซน์ใช้สัญญาณ ความถี่ 50 Hz ขนาด 1.06 Volt

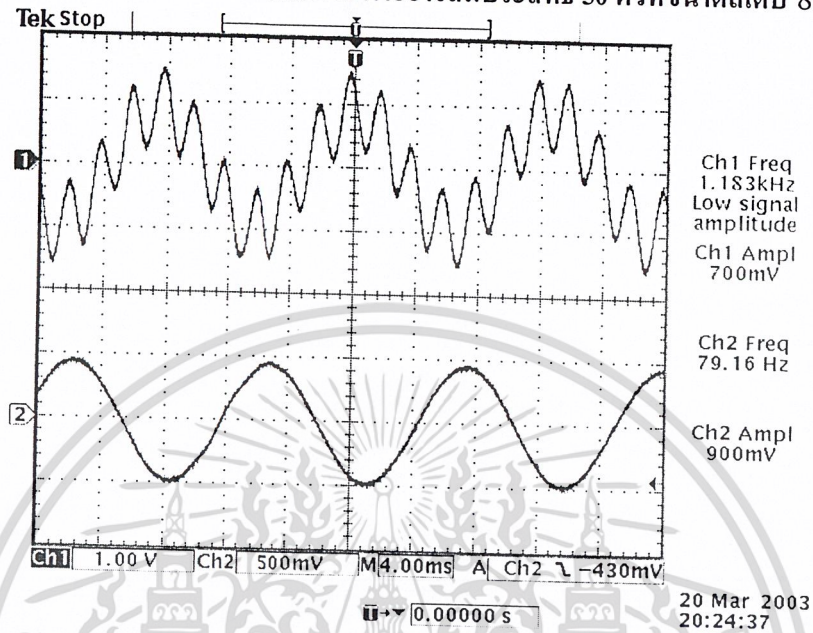
เอกสารนี้เป็นเอกสารที่เผยแพร่โดยมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี เพื่อประโยชน์ด้านการค้า

ไม่ว่าการณีใดๆ ทั้งสิ้น สัญญาณรบกวนรูปไซน์ใช้สัญญาณ ความถี่ 5 kHz ขนาด 1.06 Volt ทุกครั้งที่มีการนำไปใช้

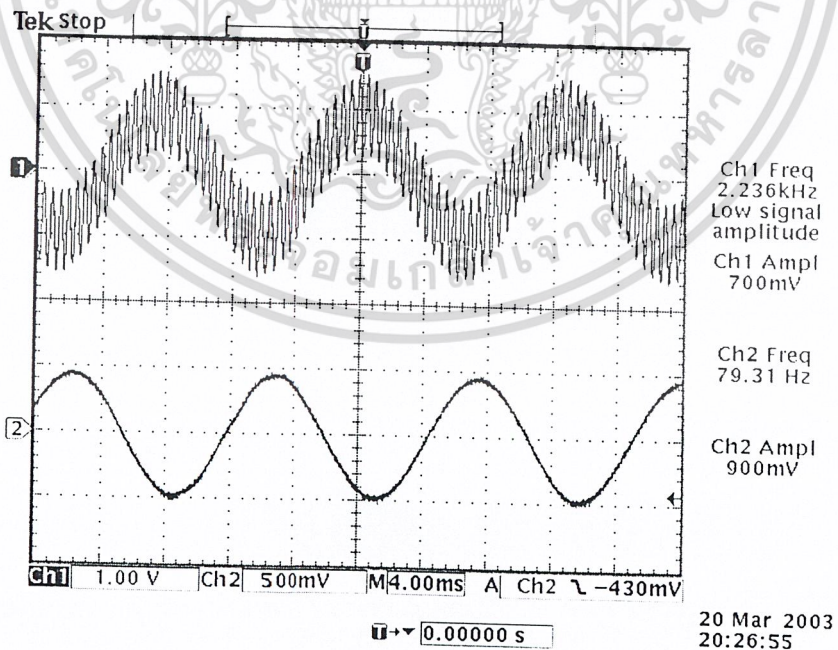
4.4 ผลตอบสนองของตัวกรองปรับตัวจากการสร้างในระบบประมวลผลสัญญาณ

4.4.1 ผลตอบสนองของตัวกรองปรับตัวแบบเอพีไออาร์

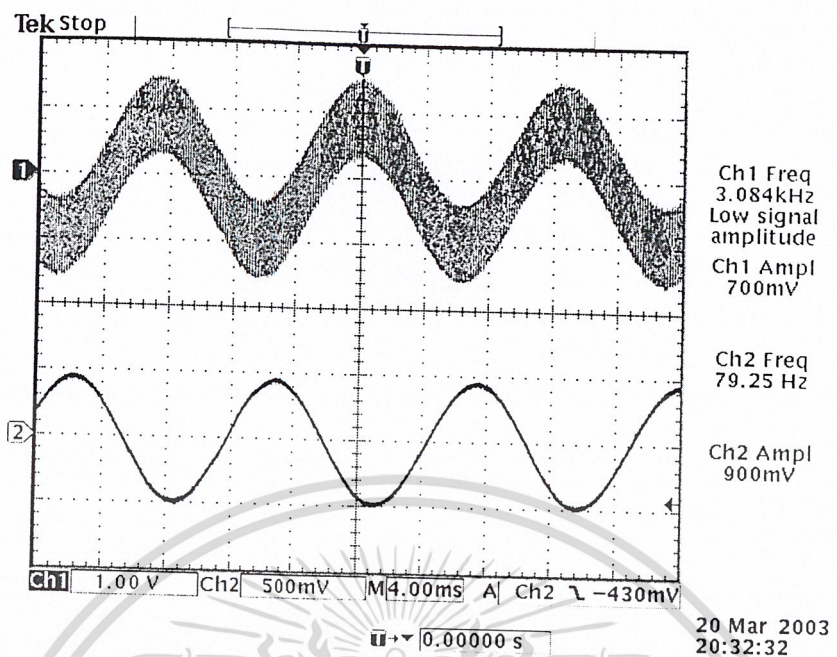
ผลตอบสนองของตัวกรองปรับตัวแบบเอพีไออาร์สัมประสิทธิ์ 50 ตัวที่ขนาดสเกล 8×10^{-10}



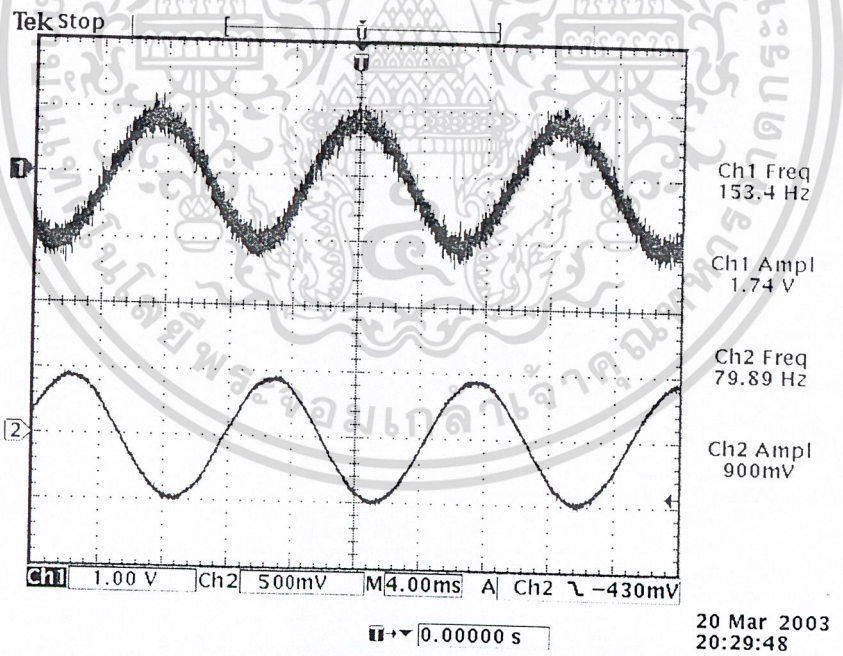
รูปที่ 4.38 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 500 Hz



รูปที่ 4.39 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับผูกมัดให้นำไปใช้ประโยชน์ด้านการค้า ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 2 kHz
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



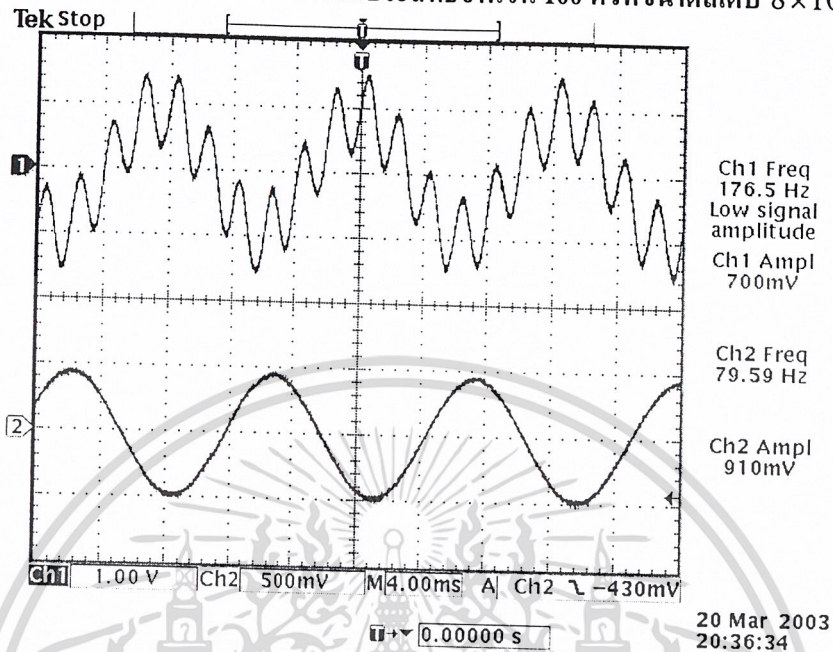
รูปที่ 4.40 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 6 kHz



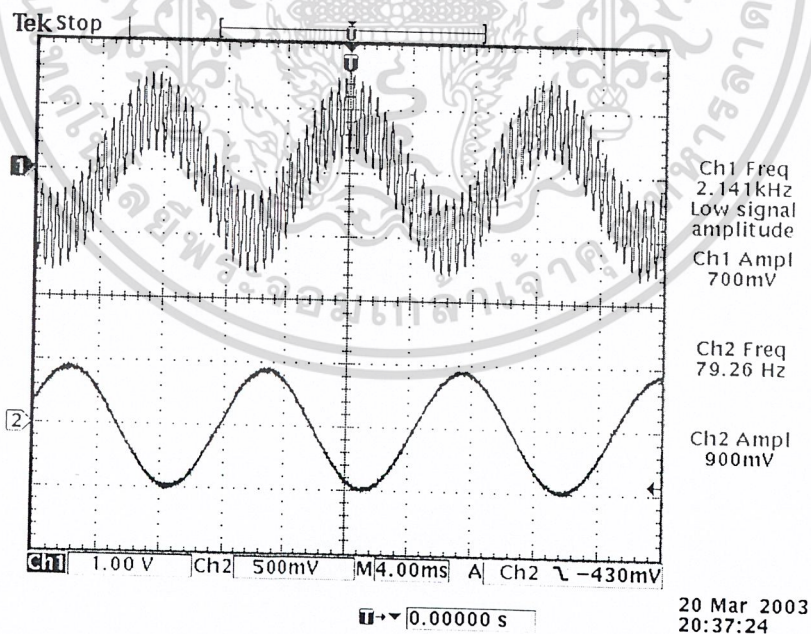
รูปที่ 4.41 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรบกวนสุ่ม

จากรูปที่ 4.38 ถึง 4.41 พบว่าตัวกรองปรับตัวแบบเอฟไออาร์มีผลตอบสนองที่ดี กล่าวคือสามารถกำจัดสัญญาณรบกวนออกได้หมด และกรองเอาเพียงแต่สัญญาณที่ต้องการได้ เอกสารทุกครั้งที่มีการนำไปใช้

ผลตอบสนองของตัวกรองปรับตัวสัมประสิทธิ์จำนวน 100 ตัวที่ขนาดสเกล 8×10^{-10}



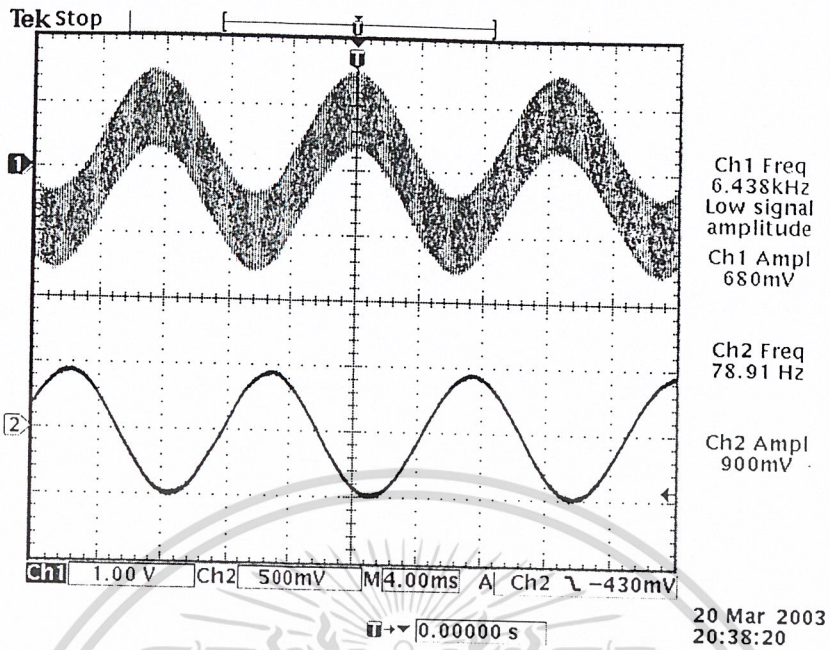
รูปที่ 4.42 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 500 Hz



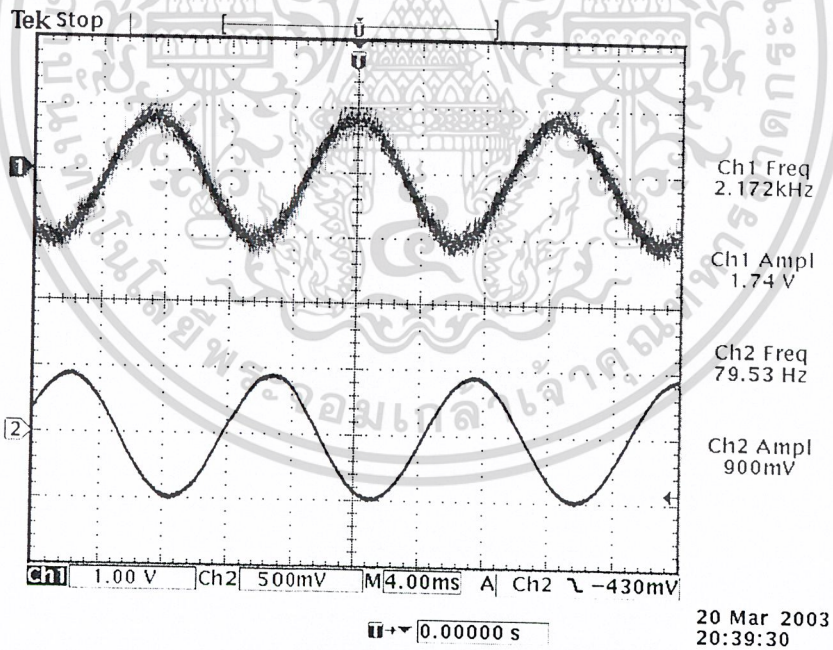
รูปที่ 4.43 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz

ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 2 Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.44 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 6 kHz



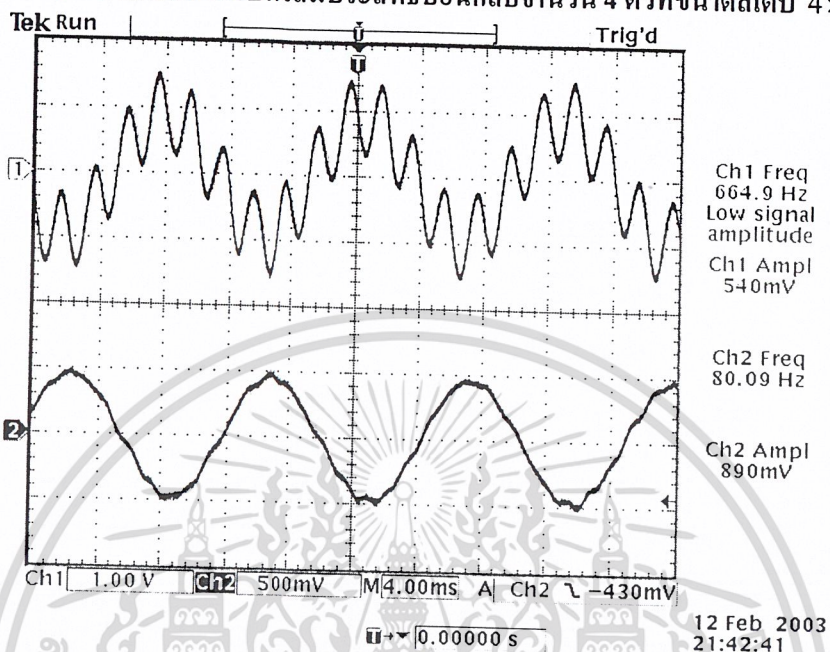
รูปที่ 4.45 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรบกวนสุ่ม

จากรูปที่ 4.42 ถึง 4.45 พบว่าตัวกรองปรับควมแบนเพาโออาร์มีผลตอบสนองที่ดี กล่าวคือสามารถกำจัดสัญญาณรบกวนออกได้หมด และกรองเอาเพียงแต่สัญญาณที่ต้องการได้

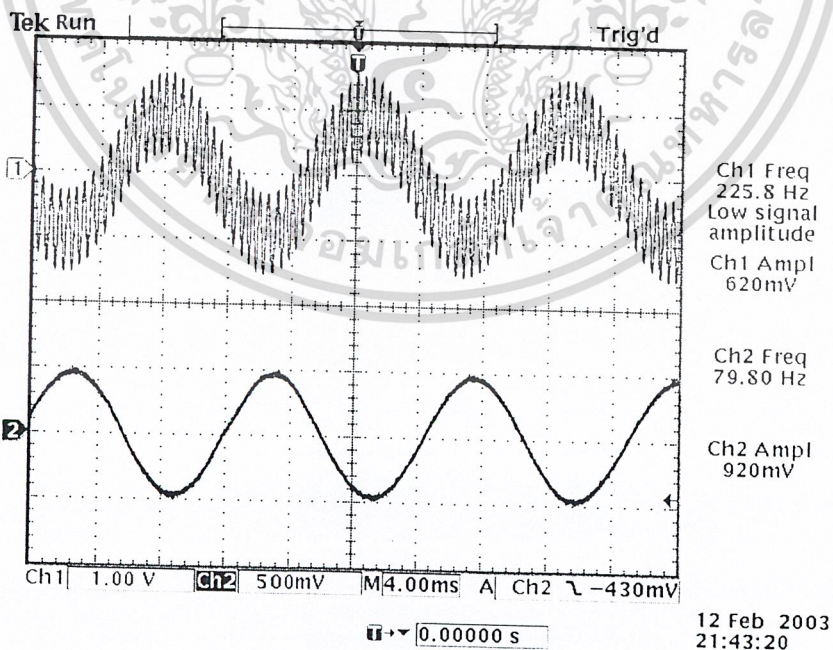
เอกสารทุกครั้งที่มีการนำไปใช้

4.4.2 ผลตอบสนองของตัวกรองปรับตัวแบบไอไออาร์

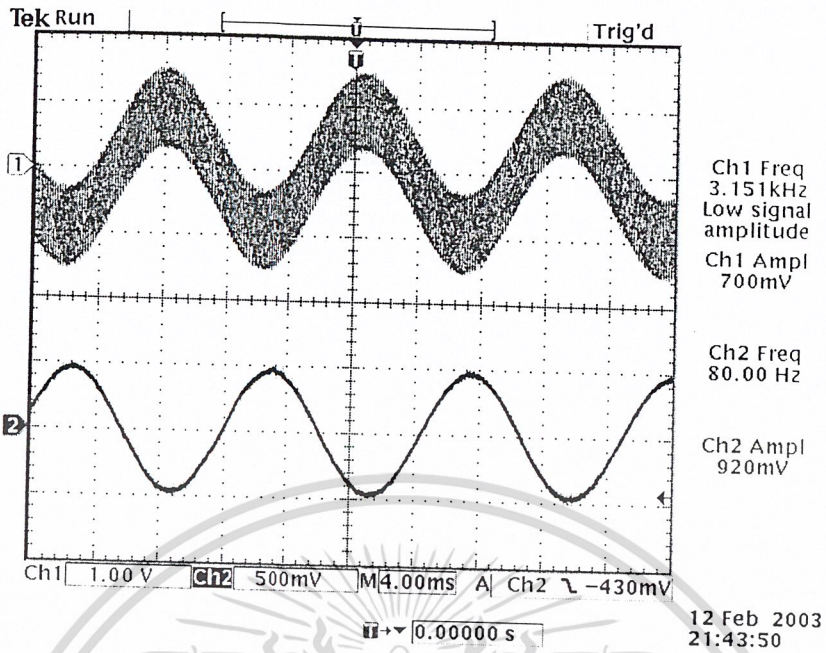
ผลตอบสนองของตัวกรองปรับตัวสัมประสิทธิ์ป้อนกลับจำนวน 4 ตัวที่ขนาดสเกล 4×10^{-10}



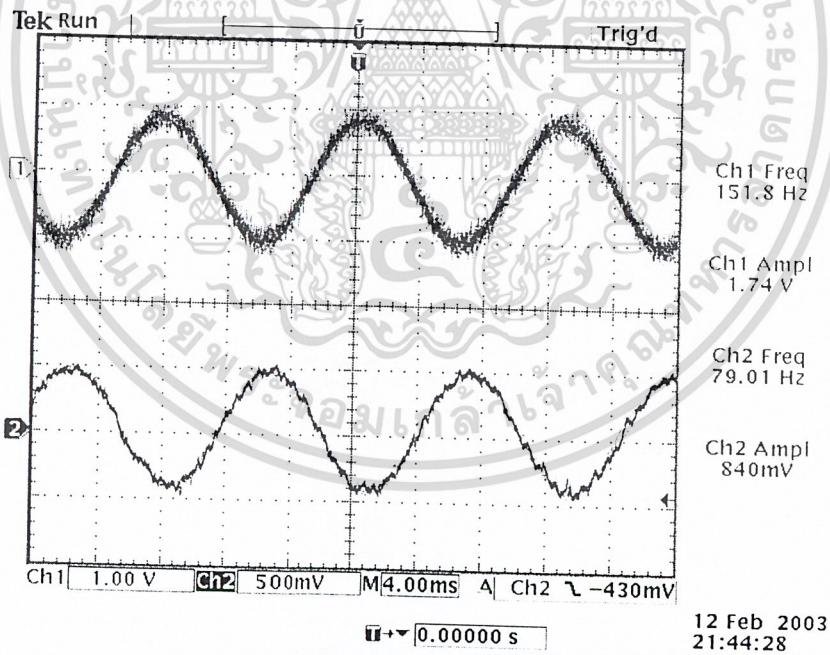
รูปที่ 4.46 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 500 Hz



รูปที่ 4.47 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดก็ตาม ยังคงมีผลผูกพันกับเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



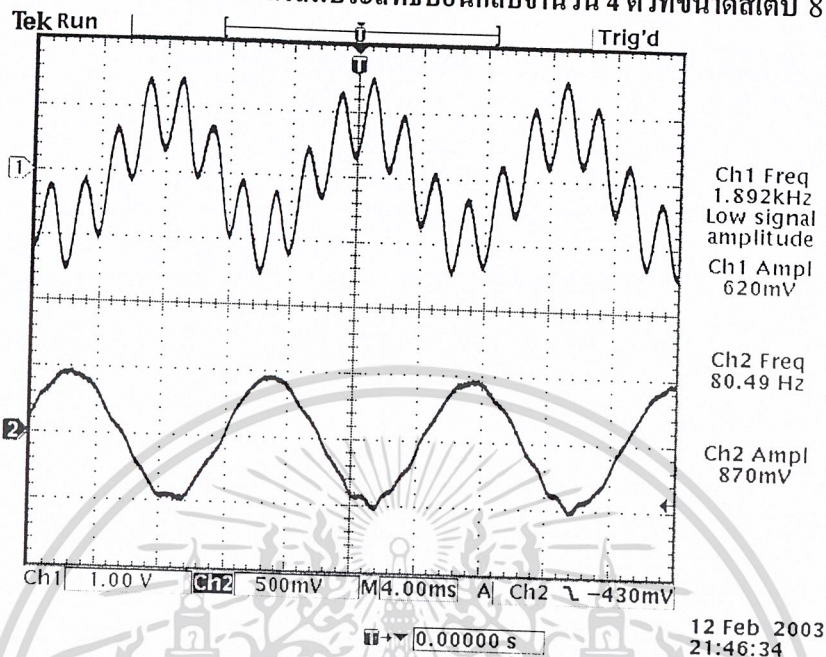
รูปที่ 4.48 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 6 kHz



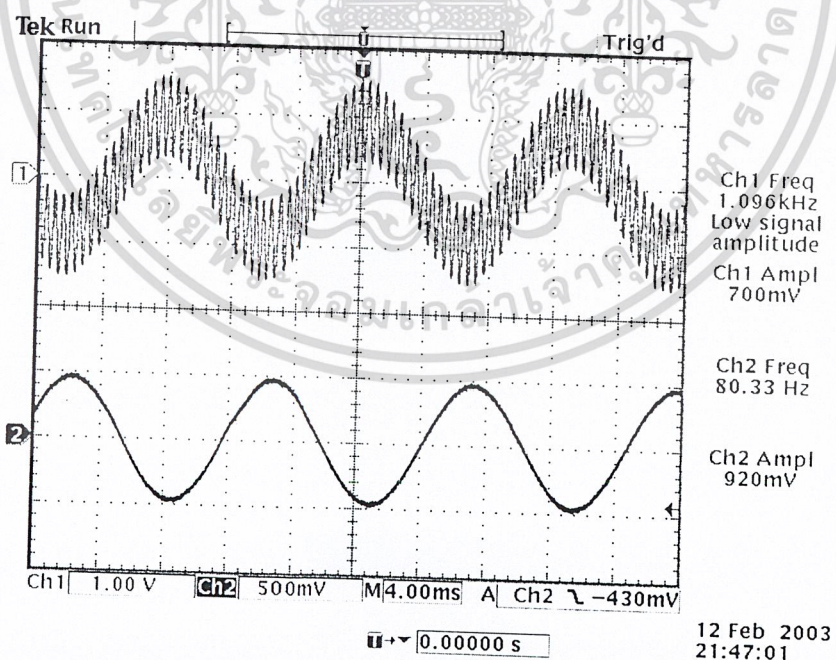
รูปที่ 4.49 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรบกวนสุ่ม

จากรูปที่ 4.46 ถึง 4.49 พบว่าตัวกรองปรับตัวแบบไอไออาร์อันดับ 4 ที่ขนาดสเกล 4×10^{10} มีผลตอบสนองค่อนข้างดี กล่าวคือสามารถกำจัดสัญญาณรบกวนรูปไซน์ออกได้หมดและสัญญาณรบกวนสุ่มออกได้เกือบหมด

ผลตอบสนองของตัวกรองปรับตัวสัมประสิทธิ์ป้อนกลับจำนวน 4 ตัวที่ขนาดสเกล 8×10^{-10}

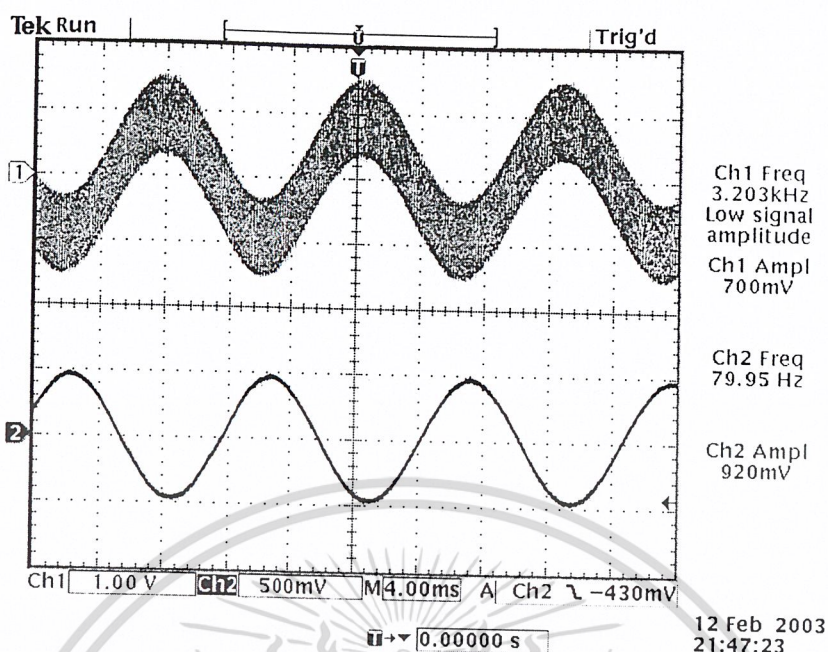


รูปที่ 4.50 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 500 Hz

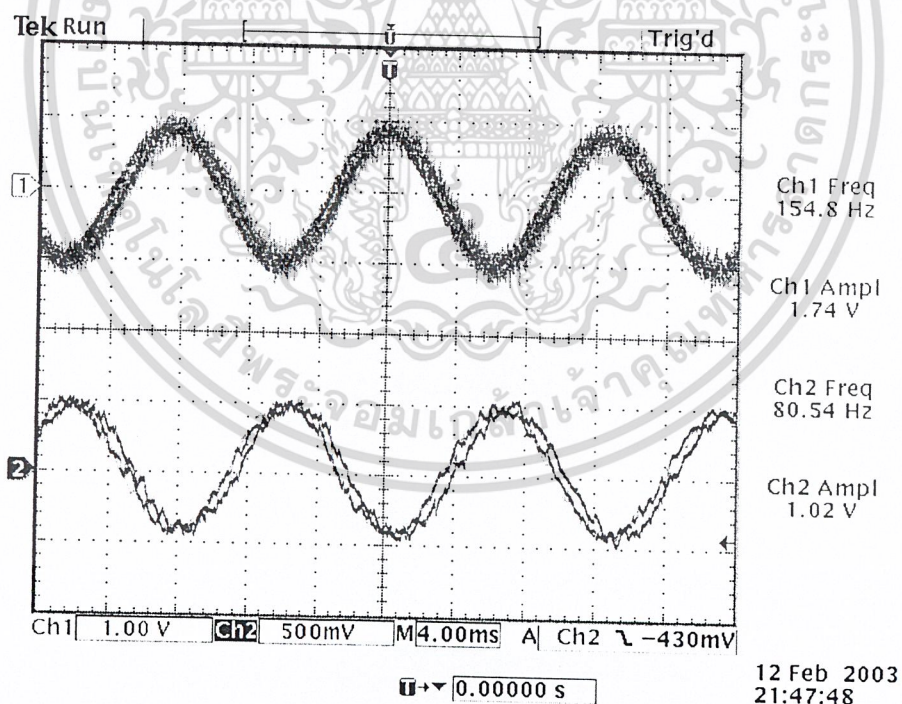


รูปที่ 4.51 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz

ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 2 kHz เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.52 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 6 kHz



รูปที่ 4.53 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรบกวนสุ่ม

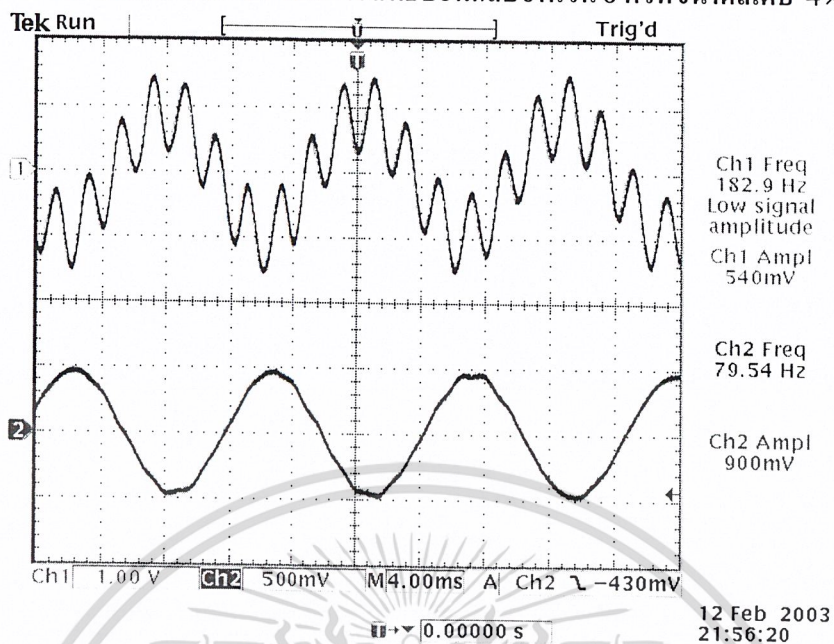
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.50 ถึง 4.53 พบว่าตัวกรองปรับตัวแบบไอโออาร์อันดับ 4 ที่ขนาดสเตป 8×10^{-10} มีผลตอบสนองค่อนข้างดี กล่าวคือสามารถกำจัดสัญญาณรบกวนรูปไซน์ออกได้หมดและสัญญาณรบกวนสุ่มออกได้เกือบหมด แต่หากสังเกตสัญญาณที่ได้จากการกรองสัญญาณรบกวนสุ่มออกจากสัญญาณเบสแบนด์ จะพบว่ายังคงมีสัญญาณรบกวนสุ่มปนอยู่มาก ซึ่งอาจเนื่องมาจากขนาดสเตปที่ใหญ่ขึ้นทำให้การปรับตัวในแต่ละครั้งมากเกินไปเป็นผลให้ตัวกรองไม่เข้าสู่ค่าสัมประสิทธิ์ที่ดีที่สุด (Optimum)

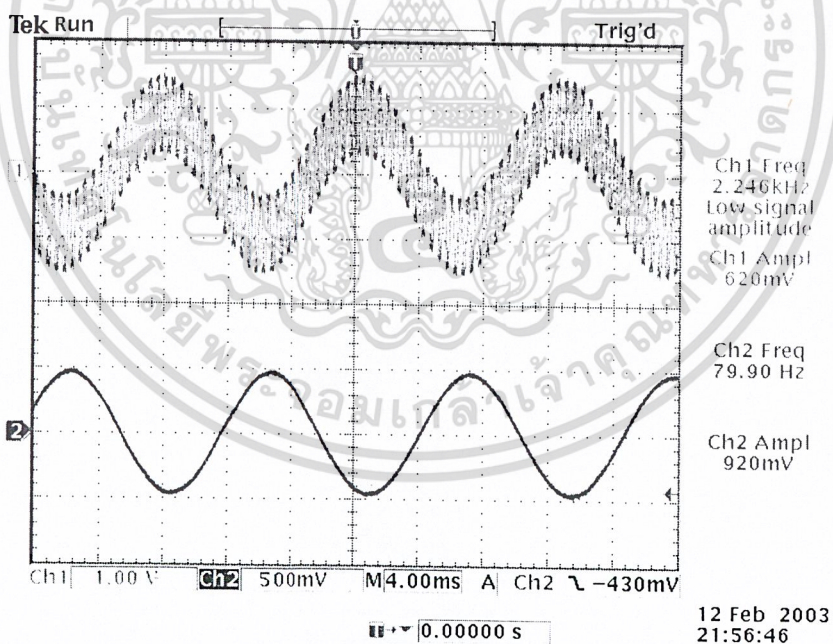


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลตอบสนองของตัวกรองปรับตัวสัมประสิทธิ์ป้อนกลับจำนวน 8 ตัวที่ขนาดสเกล 4×10^{-10}

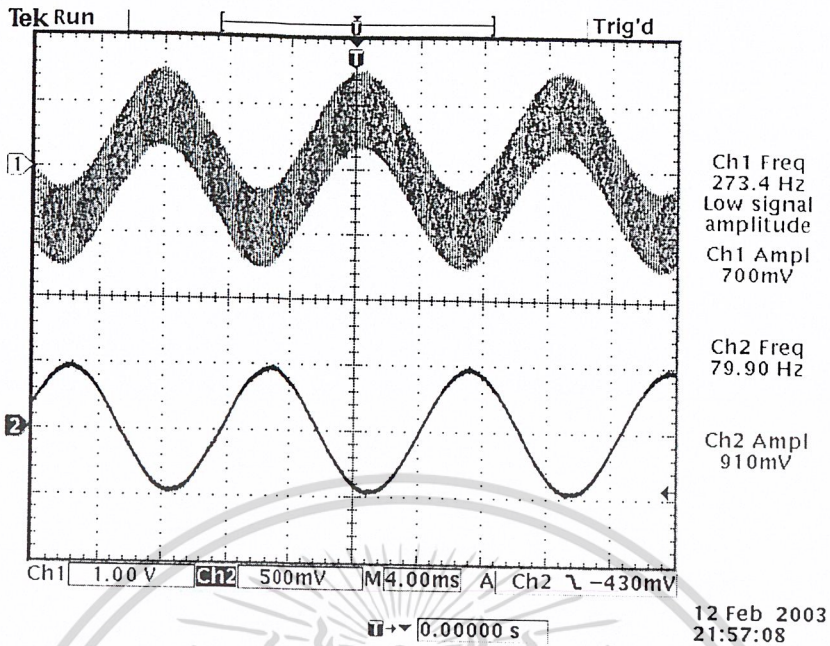


รูปที่ 4.54 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz
ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 500 Hz

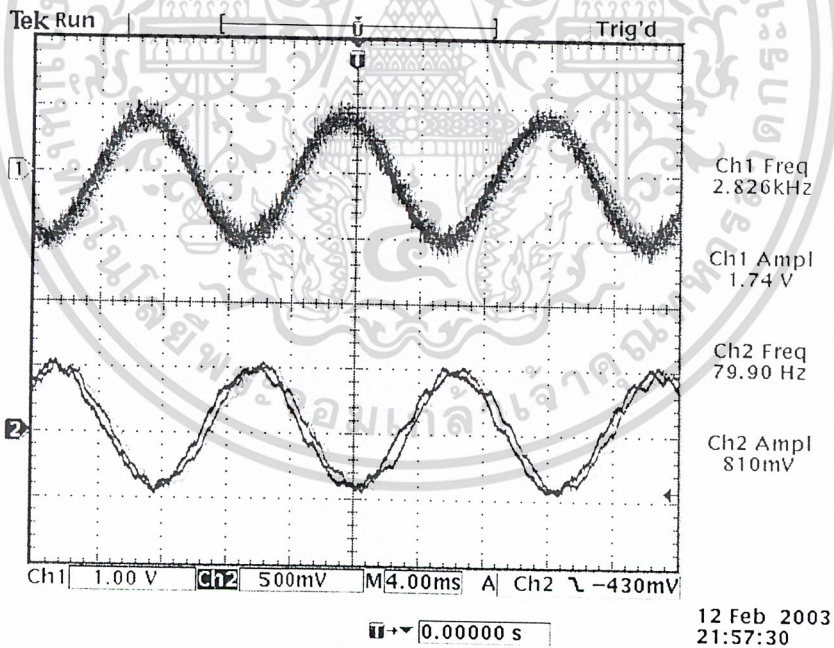


รูปที่ 4.55 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz
ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 2 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.56 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 6 kHz



รูปที่ 4.57 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรบกวนสุ่ม

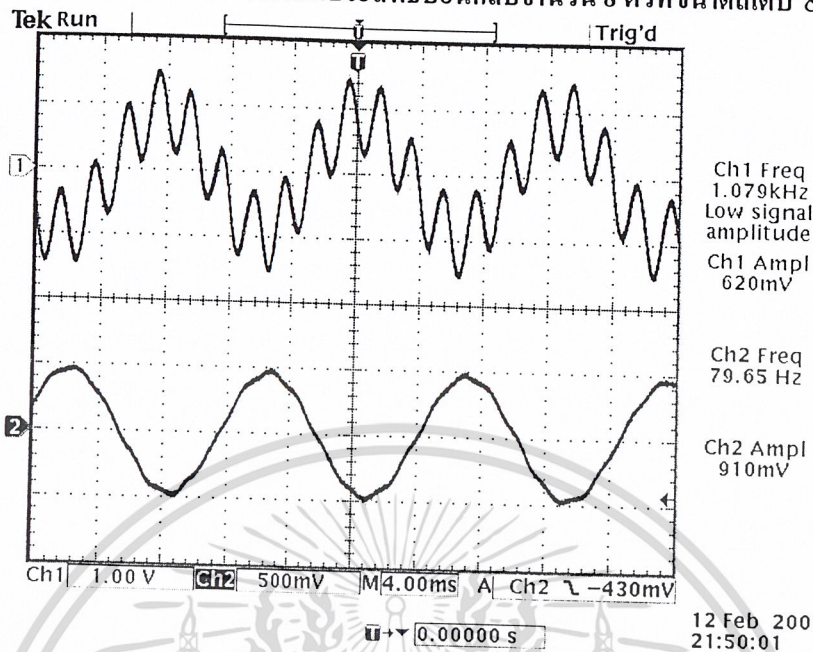
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.54 ถึง 4.57 พบว่าตัวกรองปรับตัวแบบไอไออาร์อันดับ 8 ที่ขนาดสเตป 4×10^{-10} มีผลตอบสนองค่อนข้างดี กล่าวคือสามารถกำจัดสัญญาณรบกวนรูปไซน์ออกได้หมดและสัญญาณรบกวนคู่มอดได้เกือบหมด นอกจากนั้นหากเปรียบเทียบผลการกรองสัญญาณรบกวนที่ความถี่ 500 Hz กับตัวกรองปรับตัวอันดับ 4 จะพบว่าตัวกรองปรับตัวอันดับ 8 ให้ผลที่ดีกว่า

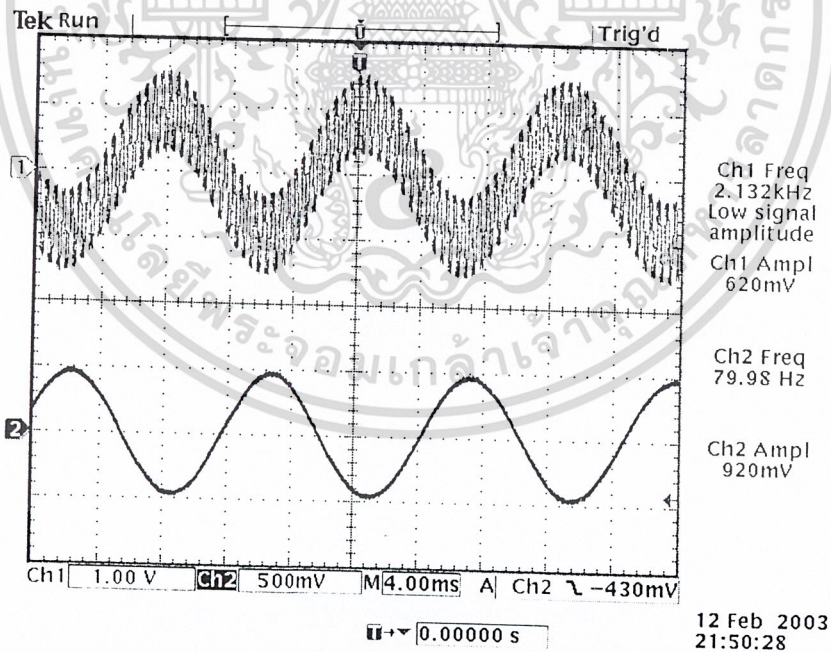


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลตอบสนองของตัวกรองปรับตัวสัมประสิทธิ์ป้อนกลับจำนวน 8 ตัวที่ขนาดสเกล 8×10^{-10}

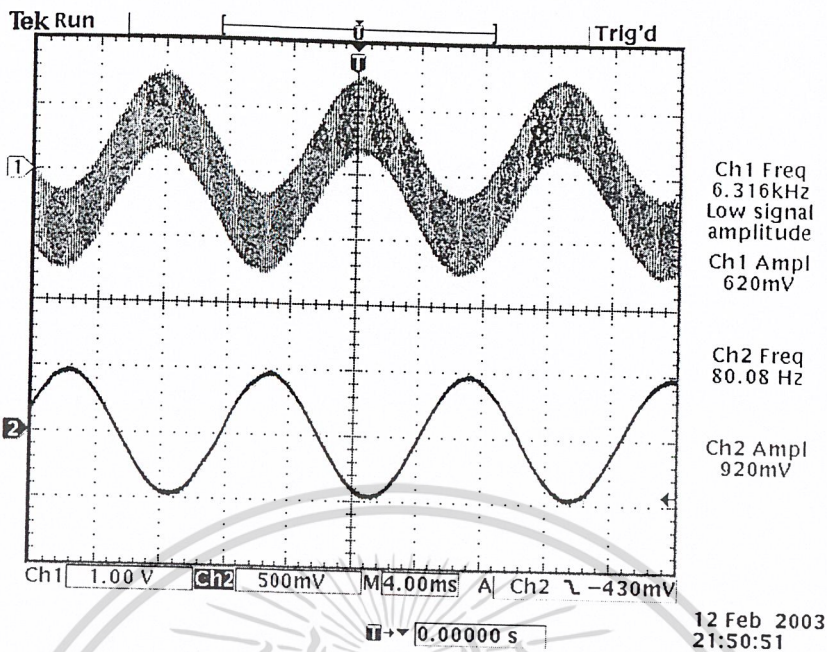


รูปที่ 4.58 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 500 Hz

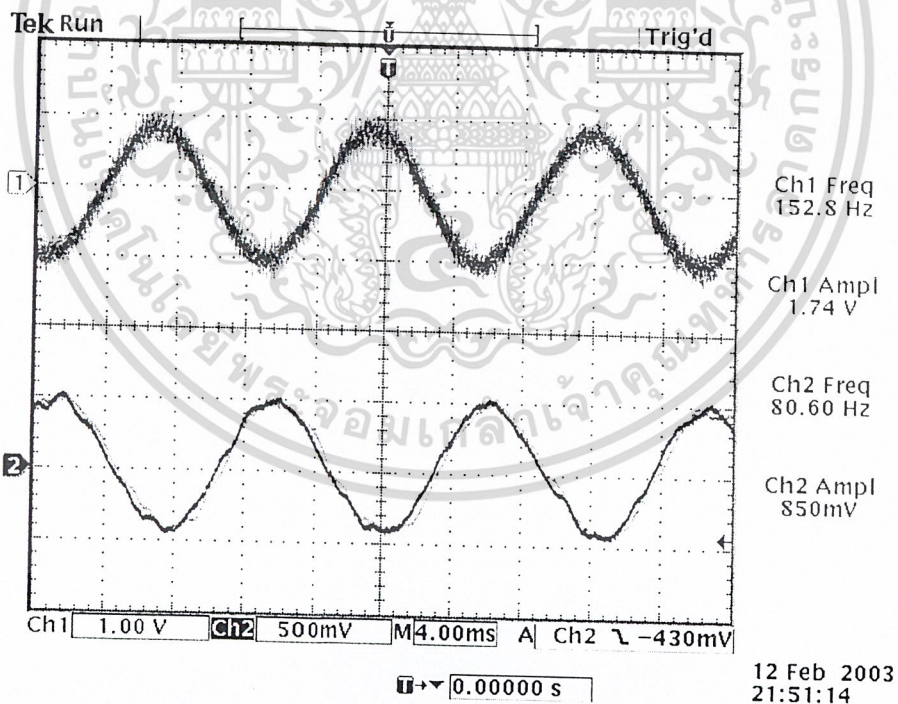


รูปที่ 4.59 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 2 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



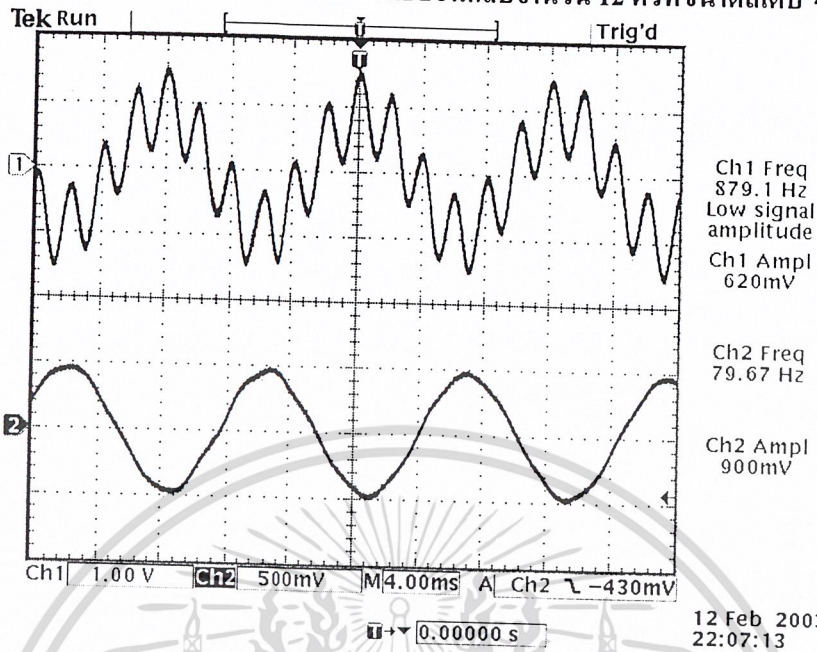
รูปที่ 4.60 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 6 kHz



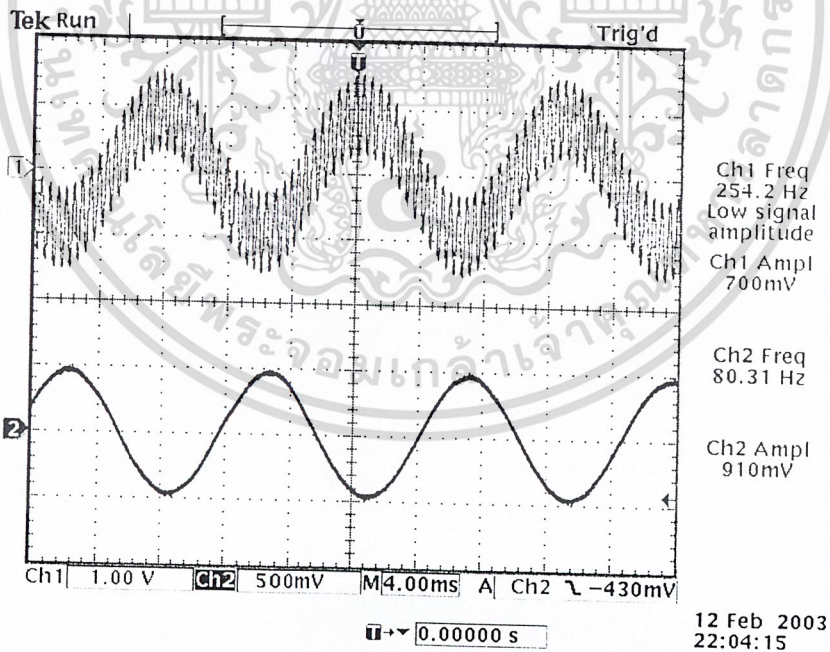
รูปที่ 4.61 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรบกวนสุ่ม

จากรูปที่ 4.58 ถึง 4.61 พบว่าตัวกรองปรับตัวแบบไอโออาร์อันดับ 8 ที่ขนาดสเกล 8×10^{-10} มีผลตอบสนองค่อนข้างดี กล่าวคือสามารถกำจัดสัญญาณรบกวนรูปไซน์ออกได้หมดและสัญญาณรบกวนสุ่มออกได้เกือบหมด

ผลตอบสนองของตัวกรองปรับตัวสัมประสิทธิ์ป้อนกลับจำนวน 12 ตัวที่ขนาดสเตป 4×10^{-10}

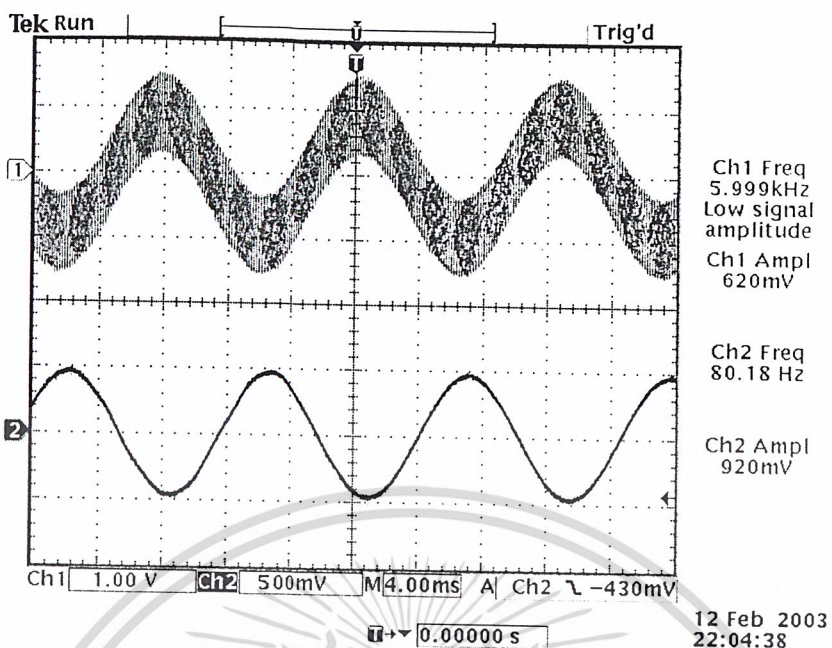


รูปที่ 4.62 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 500 Hz

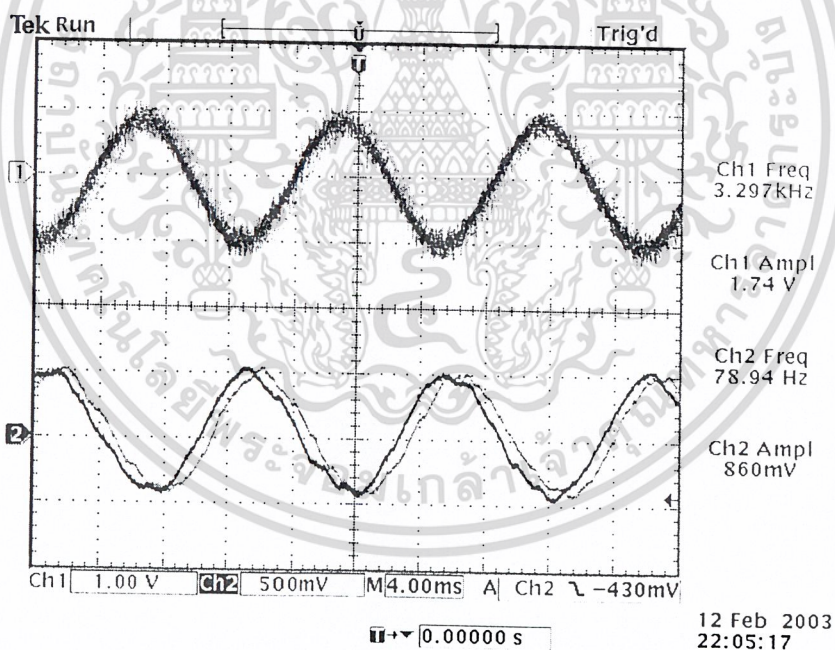


รูปที่ 4.63 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 2 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.64 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 6 kHz

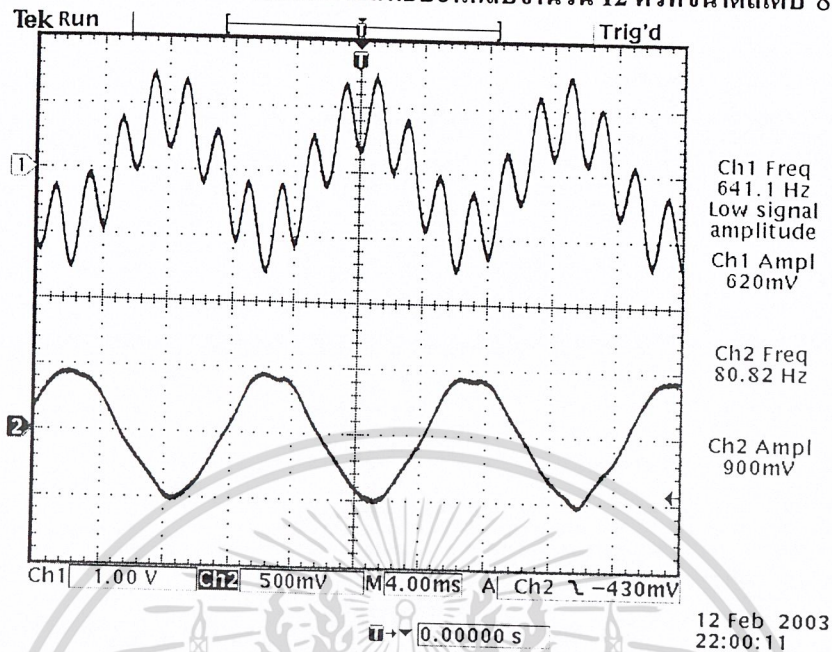


รูปที่ 4.65 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรบกวนสุ่ม

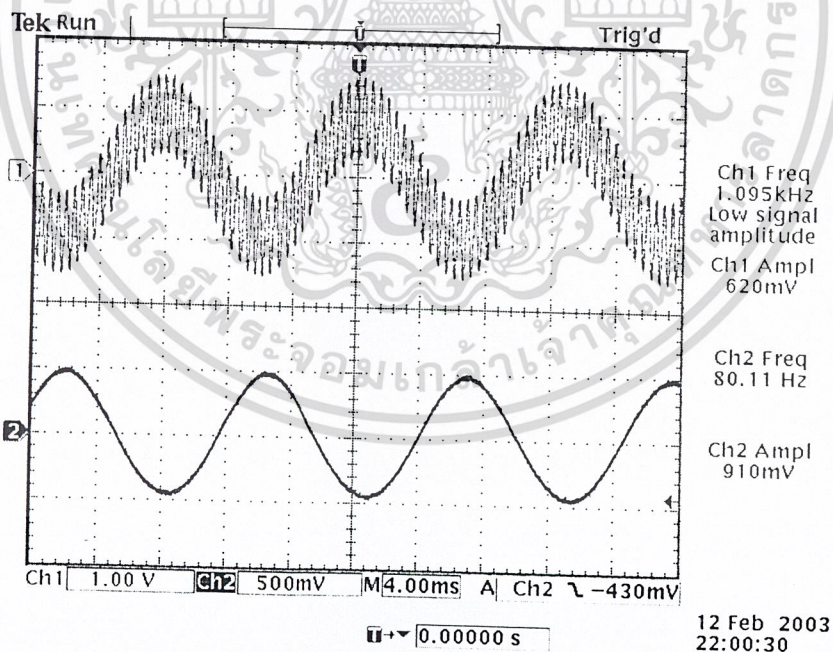
จากรูปที่ 4.62 ถึง 4.65 พบว่าตัวกรองปรับตัวแบบไอโออาร์อันดับ 12 ที่ขนาดสเกล 4×10^{-10} มี

ผลตอบสนองค่อนข้างดี กล่าวคือสามารถกำจัดสัญญาณรบกวนรูปไซน์ออกได้หมดและสัญญาณรบกวนสุ่มออกได้เกือบหมด นอกจากนี้หากเปรียบเทียบผลการกรองสัญญาณรบกวนที่มีความถี่ 500 Hz กับตัวกรองปรับตัวอันดับ 8 จะพบว่าตัวกรองปรับตัวอันดับ 12 ให้ผลที่ดีกว่า

ผลตอบสนองของตัวกรองปรับตัวสัมประสิทธิ์ป้อนกลับจำนวน 12 ตัวที่ขนาดสเตป 8×10^{-10}

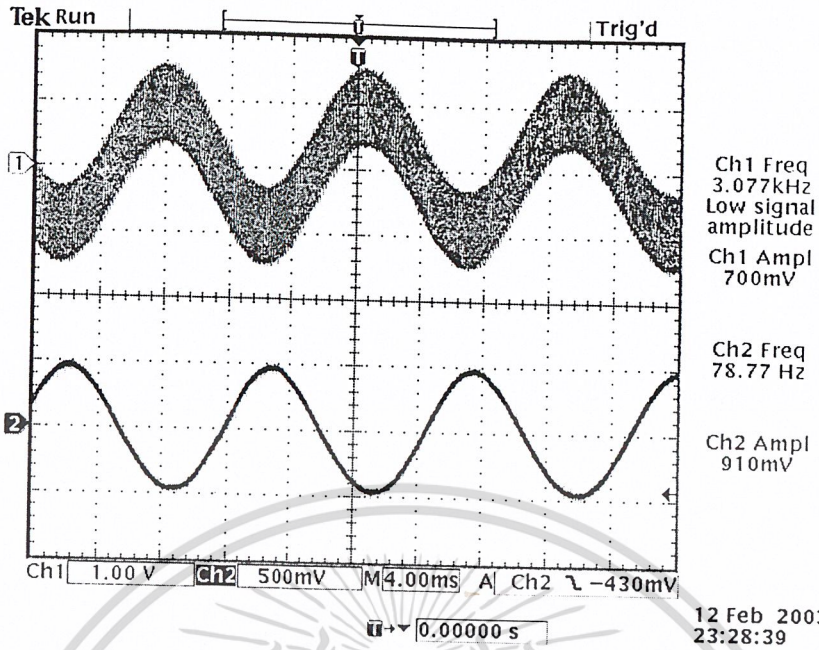


รูปที่ 4.66 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 500 Hz

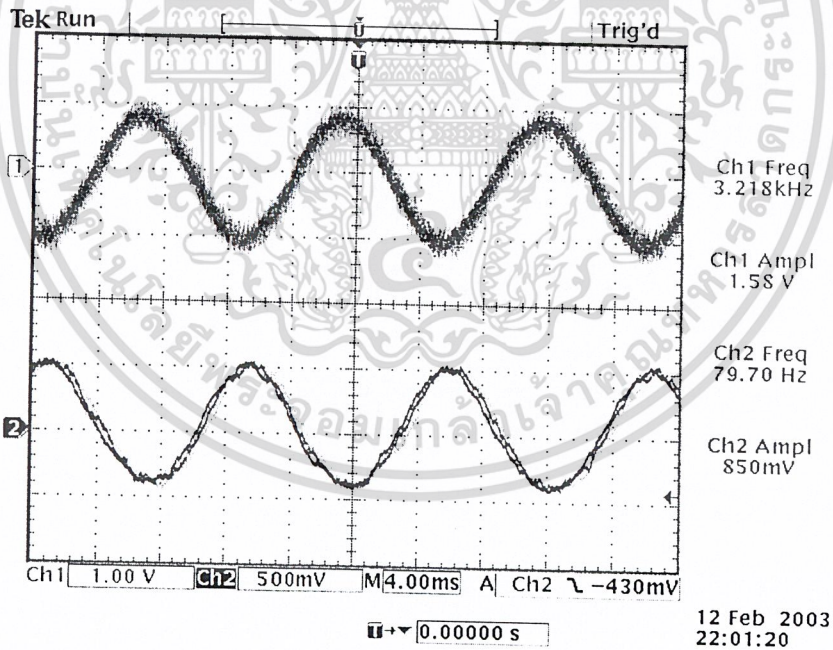


รูปที่ 4.67 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 2 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.68 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 6 kHz



รูปที่ 4.69 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรบกวนสุ่ม

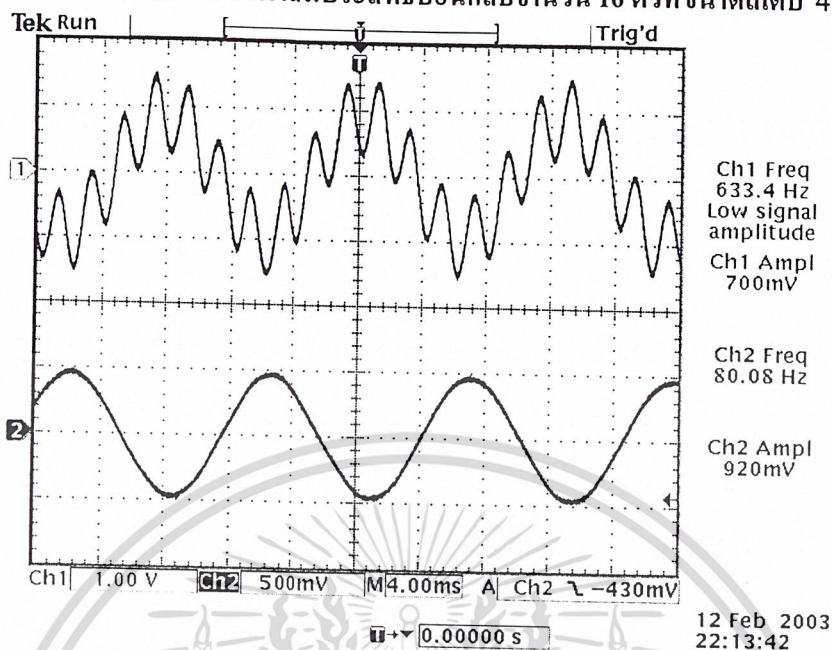
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.66 ถึง 4.69 พบว่าตัวกรองปรับตัวแบบไออาร์อันดับ 12 ที่ขนาดสเตป 8×10^{-10} มีผลตอบสนองค่อนข้างดี กล่าวคือสามารถกำจัดสัญญาณรบกวนรูปไซน์ออกได้หมดและสัญญาณรบกวนสุ่มออกได้เกือบหมด แต่หากเปรียบเทียบผลการกรองสัญญาณรบกวนที่ความถี่ 500 Hz และสัญญาณรบกวนสุ่ม กับตัวกรองปรับตัวอันดับ 8 จะพบว่าตัวกรองปรับตัวอันดับ 8 ให้ผลที่ดีกว่า ซึ่งอาจเกิดจากขนาดสเตปที่ใหญ่และจำนวนสัมประสิทธิ์ที่เพิ่มขึ้นทำให้ความไวของสัมประสิทธิ์เพิ่ม การปรับตัวในแต่ละครั้งจึงมากเกินไปเป็นผลให้ตัวกรองไม่ลู่เข้าสู่ค่าสัมประสิทธิ์ที่ดีที่สุด

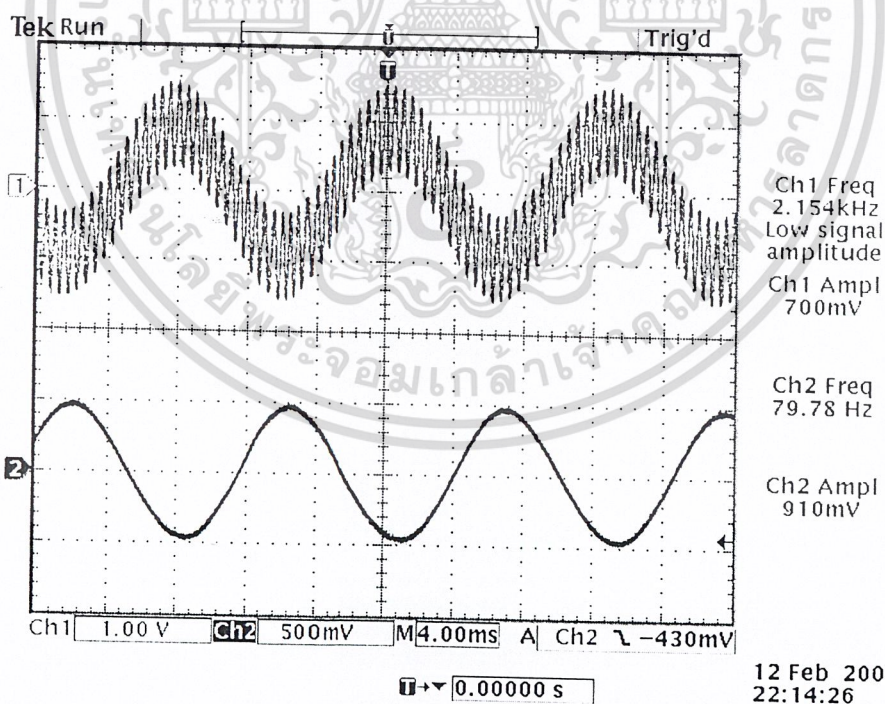


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลตอบสนองของตัวกรองปรับตัวลัมประสิทธิ์ป้อนกลับจำนวน 16 ตัวที่ขนาดสเกล 4×10^{-10}



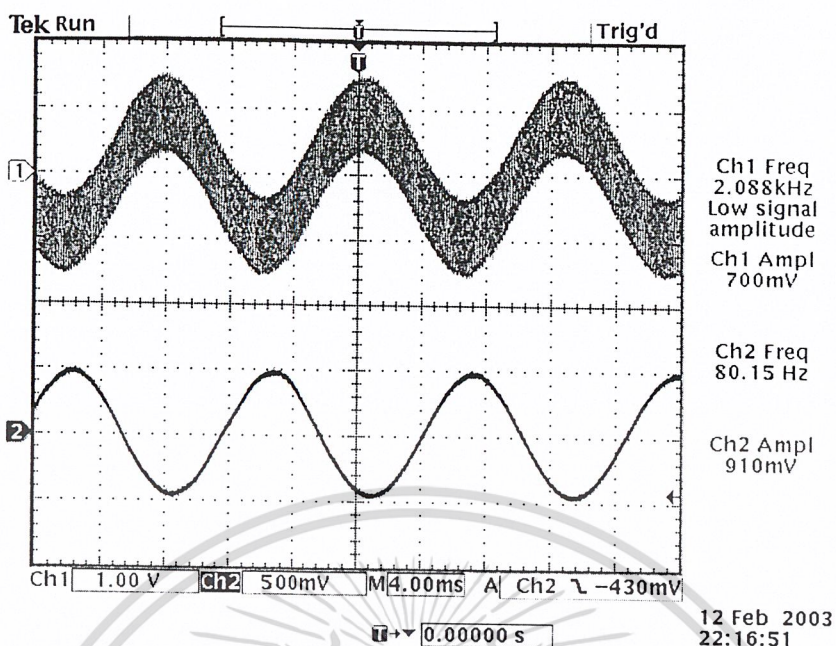
รูปที่ 4.70 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz
ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 500 Hz



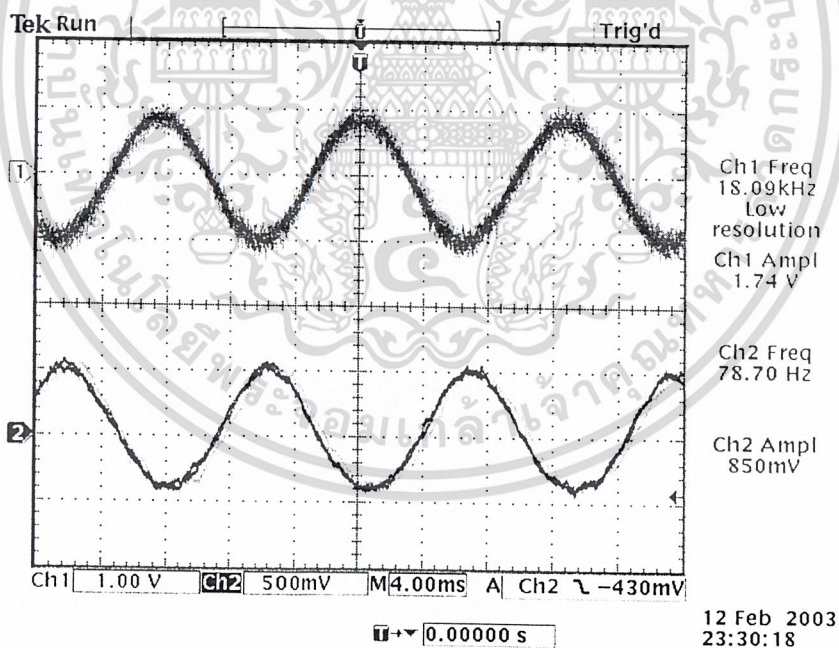
รูปที่ 4.71 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz

ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 2 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.72 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 6 kHz



รูปที่ 4.73 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรบกวนสุ่ม

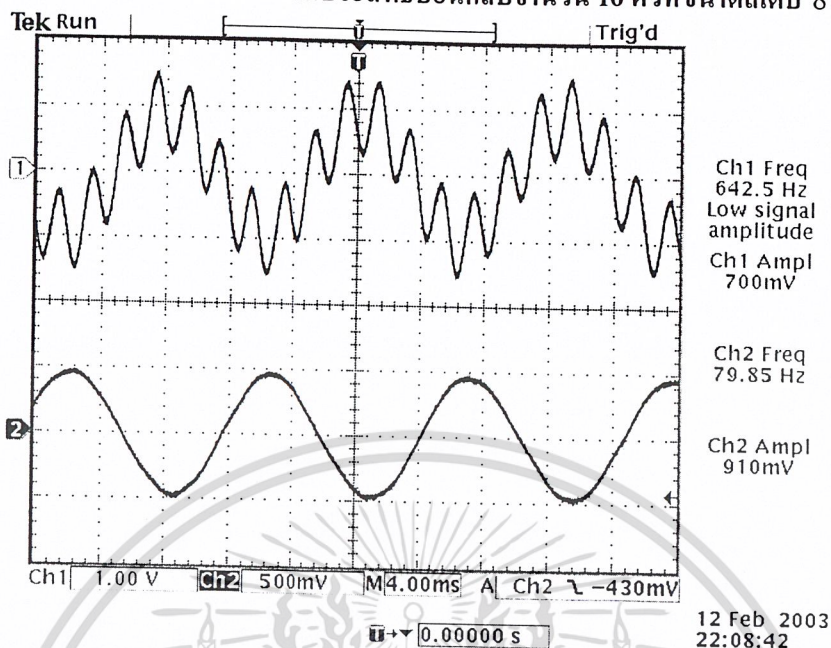
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.70 ถึง 4.73 พบว่าตัวกรองปรับตัวแบบไออาร์อันดับ 16 ที่ขนาดสเตป 4×10^{-10} มีผลตอบสนองดี กล่าวคือสามารถกำจัดสัญญาณรบกวนรูปไซน์ออกได้หมดและสัญญาณรบกวนสุ่มออกได้เกือบหมด และหากเปรียบเทียบผลการกรองสัญญาณรบกวนที่ความถี่ 500 Hz กับตัวกรองปรับตัวอันดับ 12 ที่ขนาดสเตปเท่ากันจะพบว่าตัวกรองปรับตัวอันดับ 16 ให้ผลที่ดีกว่า

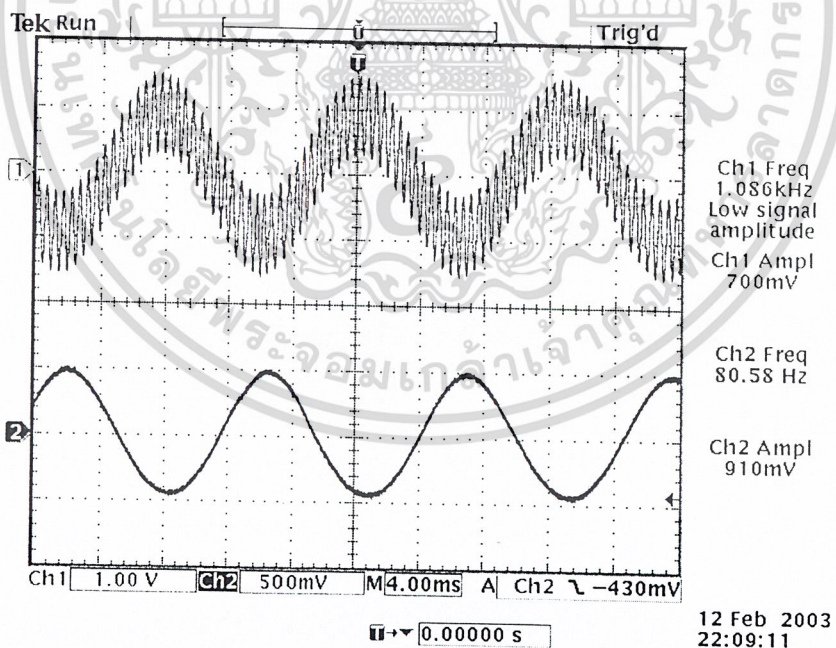


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลตอบสนองของตัวกรองปรับตัวสัมประสิทธิ์ป้อนกลับจำนวน 16 ตัวที่ขนาดสเตป 8×10^{-10}

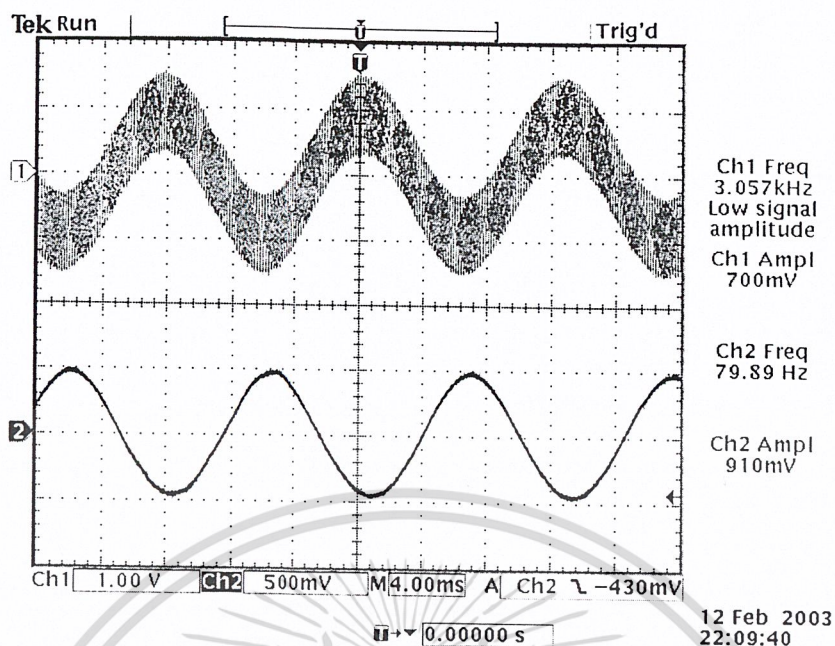


รูปที่ 4.74 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 500 Hz

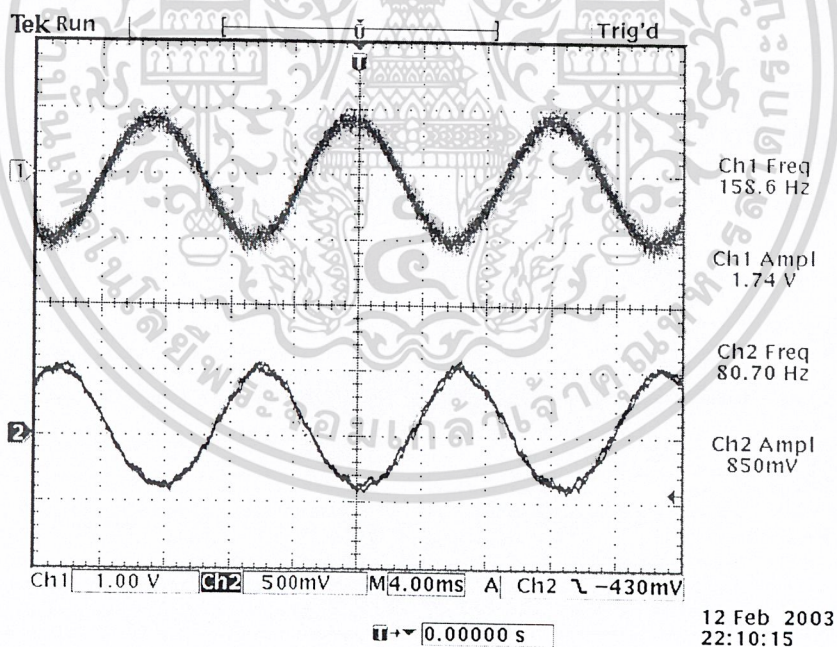


รูปที่ 4.75 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 2 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.76 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 6 kHz



รูปที่ 4.77 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรบกวนสุ่ม

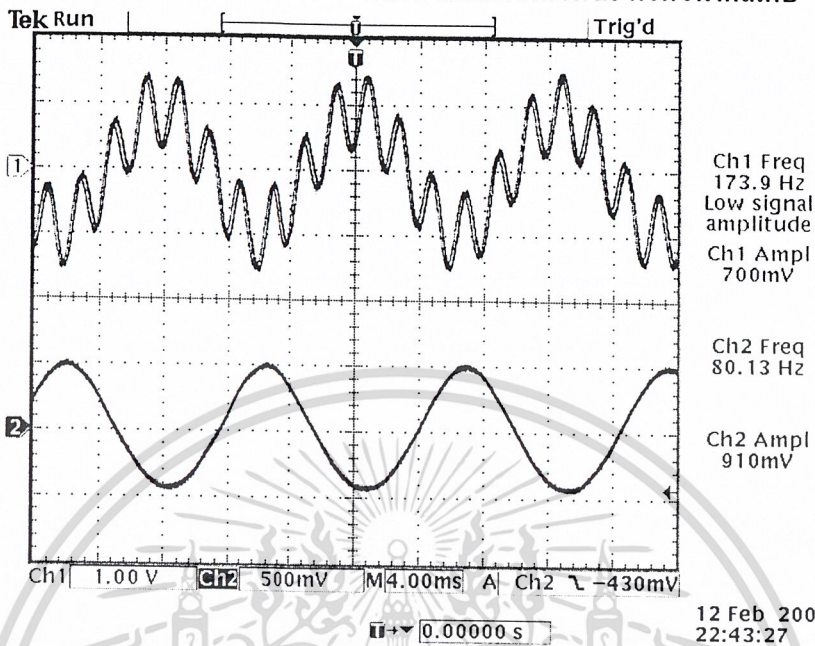
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.74 ถึง 4.77 พบว่าตัวกรองปรับตัวแบบไอไออาร์อันดับ 16 ที่ขนาดสเตป 8×10^{-10} มีผลตอบสนองดี กล่าวคือสามารถกำจัดสัญญาณรบกวนรูปไซน์ออกได้หมดและสัญญาณรบกวนสุ่มออกได้เกือบหมด และหากเปรียบเทียบผลการกรองสัญญาณรบกวนที่ความถี่ 500 Hz กับตัวกรองปรับตัวอันดับ 12 ที่ขนาดสเตปเท่ากันจะพบว่าตัวกรองปรับตัวอันดับ 16 ให้ผลที่ดีกว่า

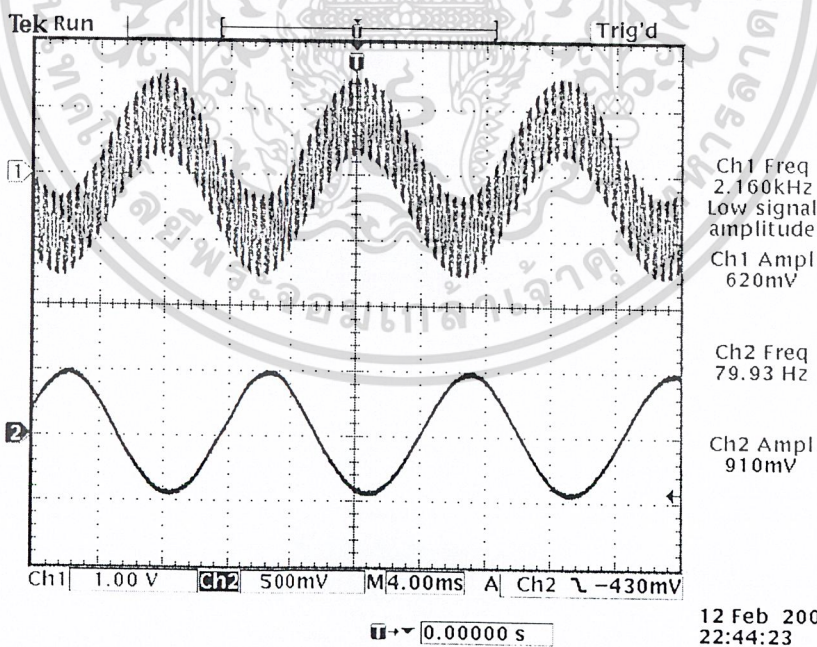


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลตอบสนองของตัวกรองปรับตัวสัมประสิทธิ์ป้อนกลับจำนวน 20 ตัวที่ขนาดสเกล 4×10^{-10}



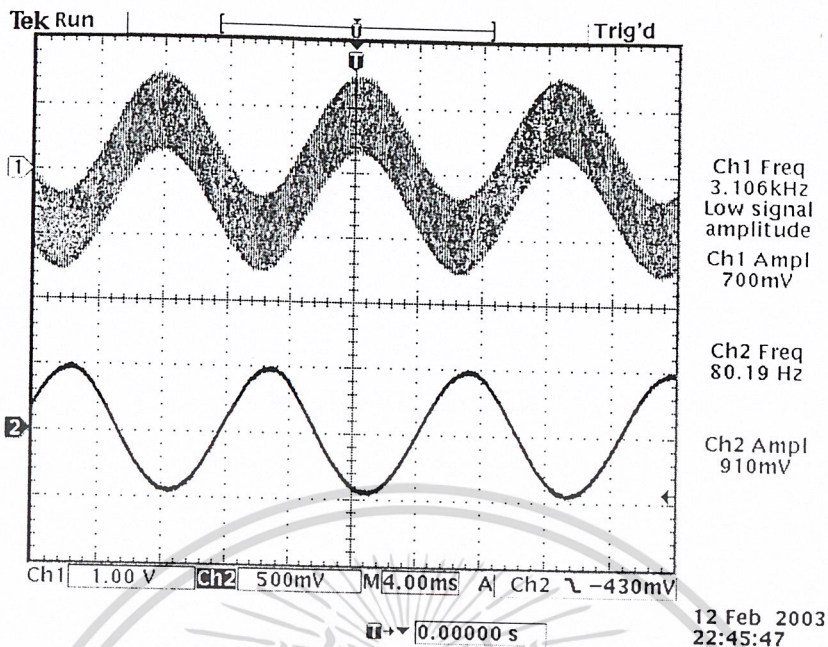
รูปที่ 4.78 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 500 Hz



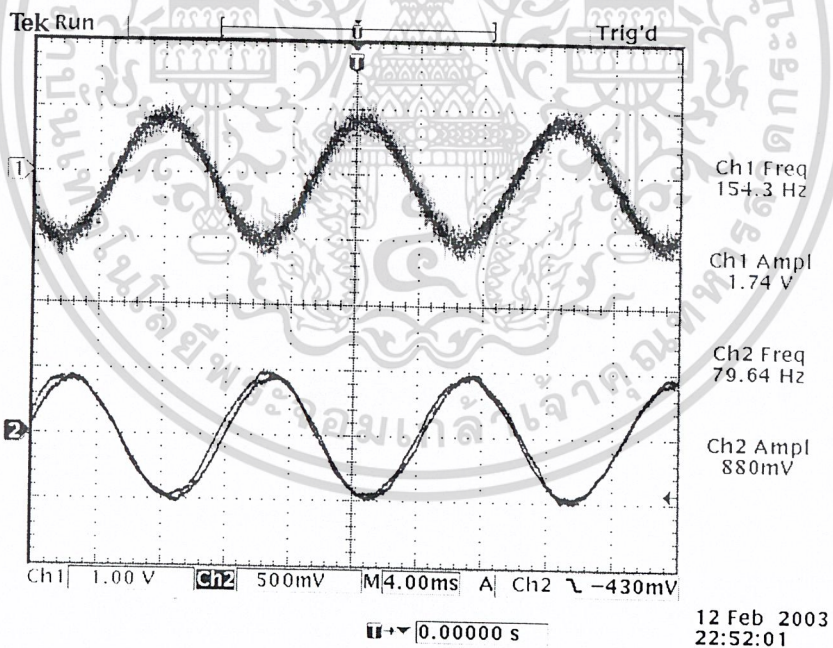
รูปที่ 4.79 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz

ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 2 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.80 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 6 kHz



รูปที่ 4.81 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรบกวนสุ่ม

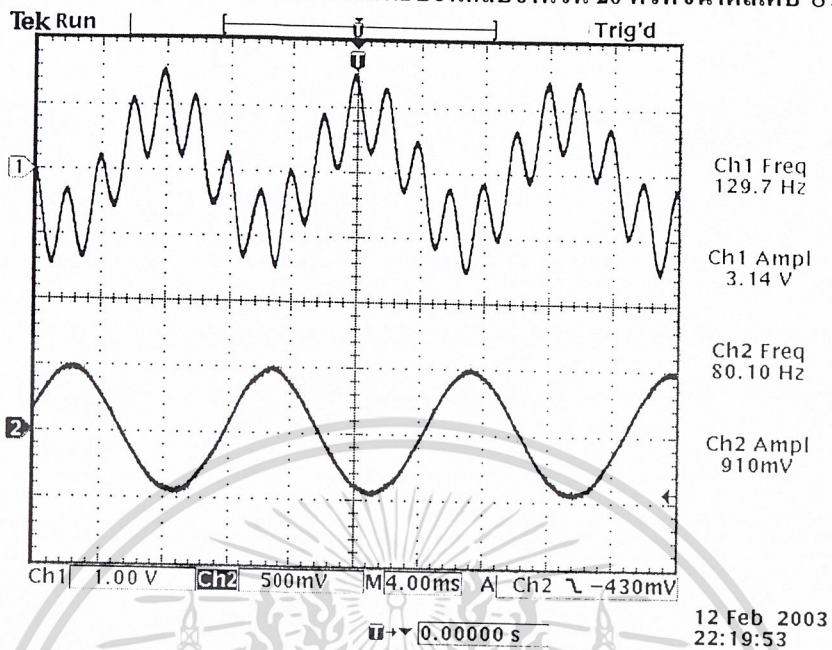
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.78 ถึง 4.81 พบว่าตัวกรองปรับตัวแบบไอไออาร์อันดับ 20 ที่ขนาดสเตป 4×10^{-10} มีผลตอบสนองดีมาก กล่าวคือสามารถกำจัดสัญญาณรบกวนรูปไซน์ออกได้หมดและสัญญาณรบกวนสุ่มออกเหลือเพียงเล็กน้อย และหากเปรียบเทียบผลการกรองสัญญาณรบกวนสุ่ม กับตัวกรองปรับตัวอันดับ 16 ที่ขนาดสเตปเท่ากันจะพบว่าตัวกรองปรับตัวอันดับ 20 ให้ผลที่ดีกว่ามาก

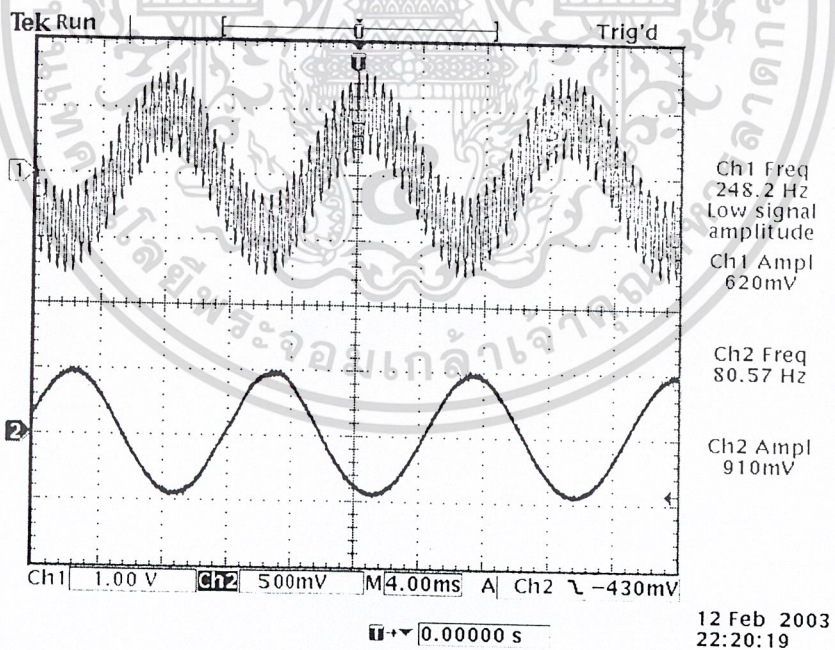


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลตอบสนองของตัวกรองปรับตัวสัมประสิทธิ์ป้อนกลับจำนวน 20 ตัวที่ขนาดสเกล 8×10^{-10}

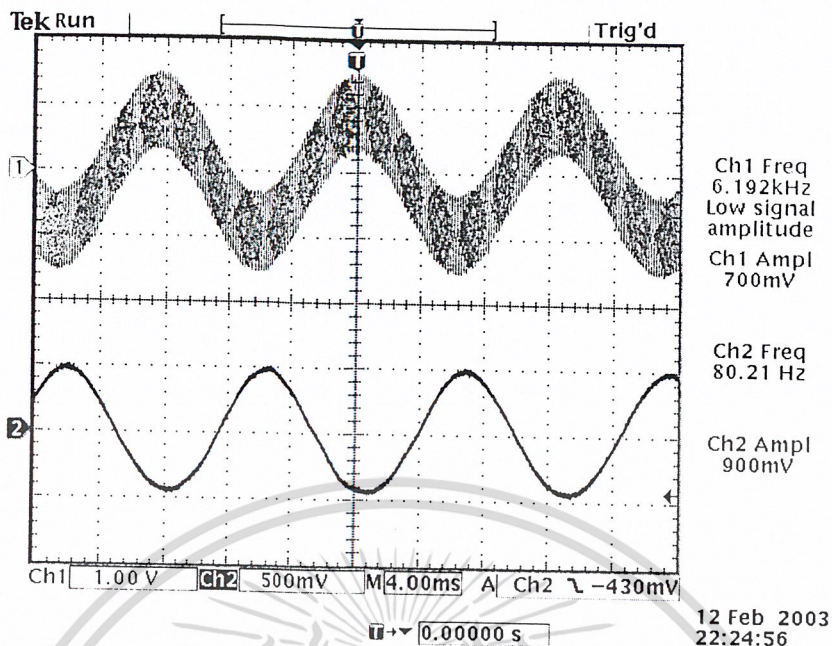


รูปที่ 4.82 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 500 Hz

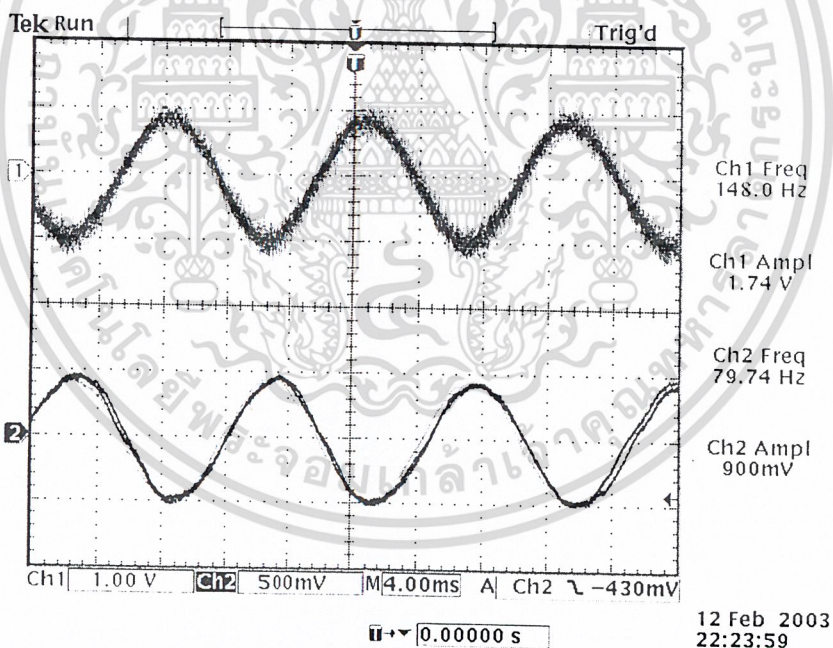


รูปที่ 4.83 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 2 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.84 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 6 kHz



รูปที่ 4.85 ผลตอบสนองของตัวกรองต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรบกวนสุ่ม

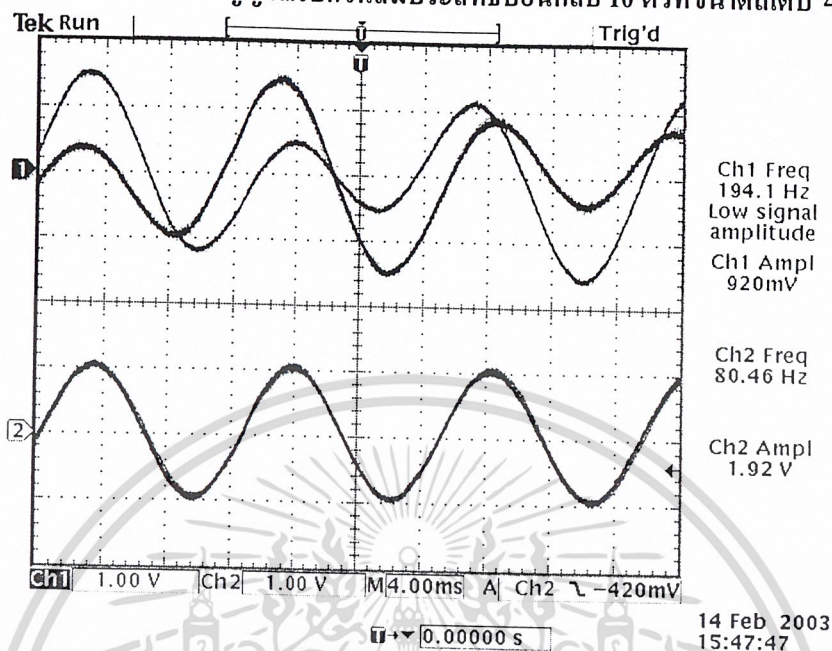
จากรูปที่ 4.82 ถึง 4.85 พบว่าตัวกรองปรับตัวแบบไอไออาร์อันดับ 20 ที่ขนาดสเกล 8×10^{-10} มีผลตอบสนองดีมาก กล่าวคือสามารถกำจัดสัญญาณรบกวนรูปไซน์ออกได้หมดและสัญญาณรบกวนสุ่ม

ออกได้เกือบหมด และหากเปรียบเทียบผลการกรองสัญญาณรบกวนทั้งหมด กับตัวกรองปรับตัวอันดับ 16 ที่ขนาดสเกลเท่ากันจะพบว่าตัวกรองปรับตัวอันดับ 20 ให้ผลที่ดีกว่า

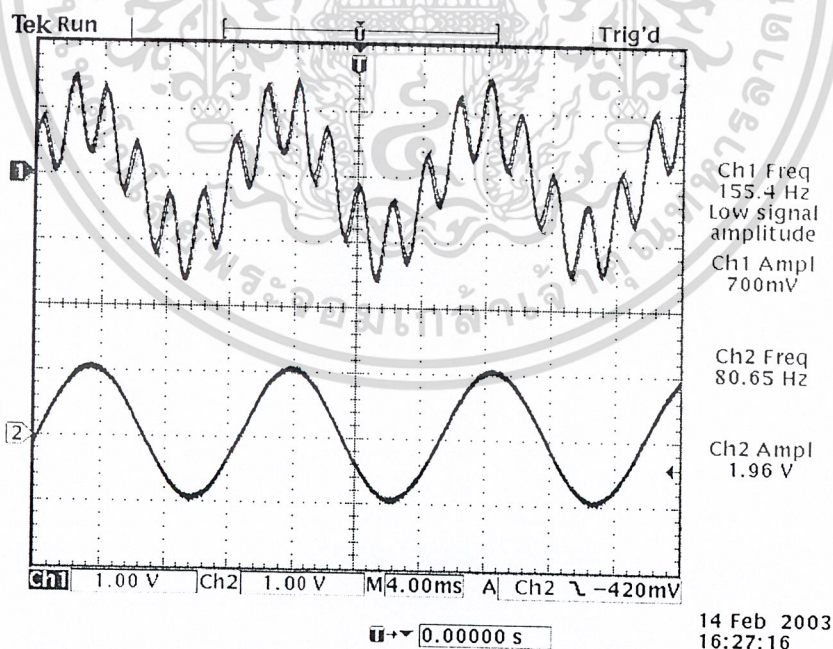
ไม่ว่ากรณีใดๆ พงษ์อน อีกทั้งห้ามมิให้ตีแบบลงเนื้อหา และต้องขออนุญาตเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.3 ผลตอบสนองของตัวกำจัดสัญญาณรบกวน

ผลตอบสนองของตัวกำจัดสัญญาณรบกวนล้มประสิทธิภาพป้อนกลับ 10 ตัวที่ขนาดสเตป 4×10^{-10}



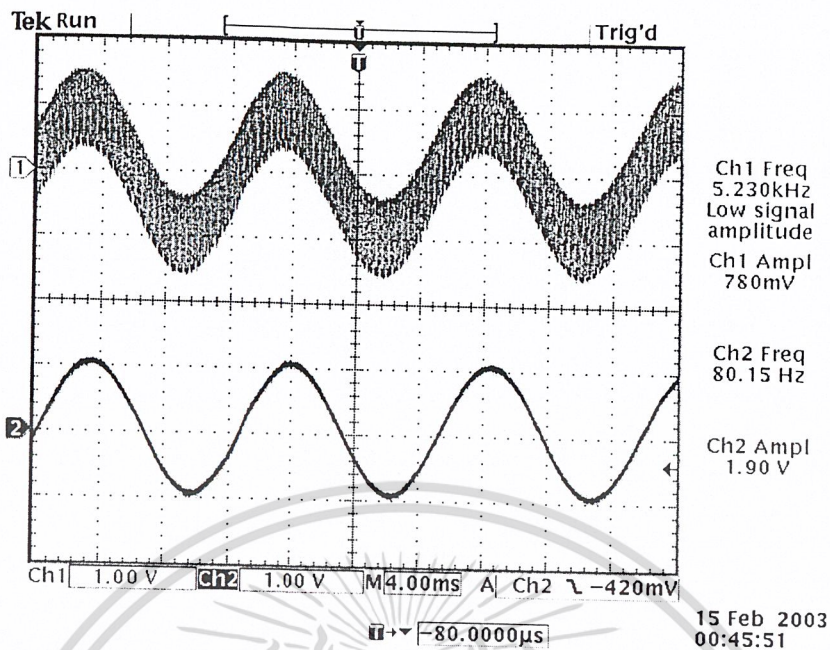
รูปที่ 4.86 ผลตอบสนองของตัวกำจัดสัญญาณรบกวนต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 50 Hz



รูปที่ 4.87 ผลตอบสนองของตัวกำจัดสัญญาณรบกวนต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz

ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 500 Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

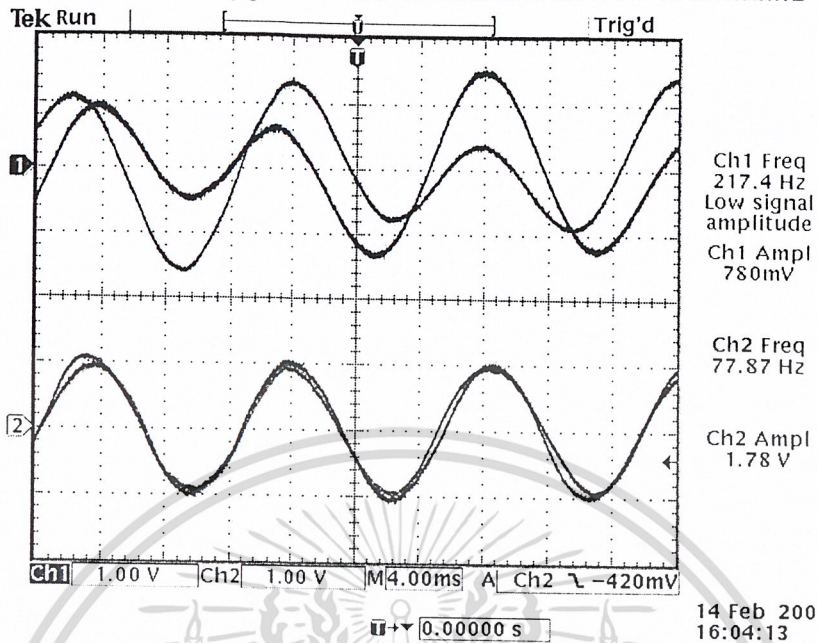


รูปที่ 4.88 ผลตอบสนองของตัวกำจัตสัญญาณรบกวนต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz
ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 5 kHz

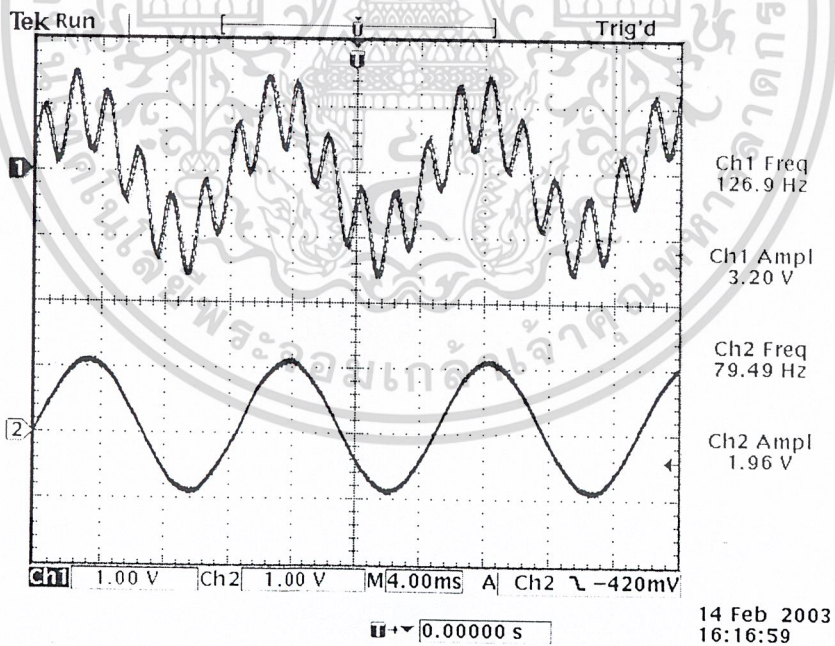
จากรูปที่ 4.85 ถึง 4.88 พบว่าตัวกำจัตสัญญาณรบกวนสัมประสิทธิ์ป้อนกลับ 10 ตัวที่ขนาดสเตป 4×10^{-10} มีผลตอบสนองที่ดี กล่าวคือสามารถกำจัตสัญญาณรบกวนรูปไซน์ทั้งที่ความถี่ต่ำกว่าและสูงกว่าสัญญาณเบสแบนด์ออกได้หมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลตอบสนองของตัวกำจัตสัญญาณรบกวนลัมประสิทธิ์ป้อนกลับ 20 ตัวที่ขนาดสเตรป 4×10^{-10}

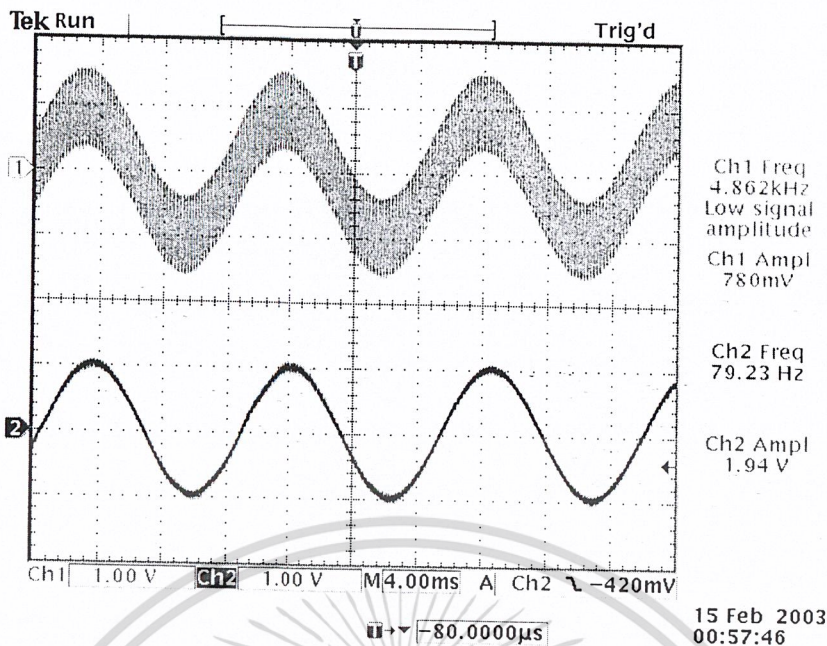


รูปที่ 4.89 ผลตอบสนองของตัวกำจัตสัญญาณรบกวนต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 50 Hz



รูปที่ 4.90 ผลตอบสนองของตัวกำจัตสัญญาณรบกวนต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 500 Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.91 ผลตอบสนองของตัวกำจัดสัญญาณรบกวนต่อสัญญาณเบสแบนด์รูปไซน์ความถี่ 80 Hz ซึ่งถูกรบกวนด้วยสัญญาณรูปไซน์ความถี่ 5 kHz

จากรูปที่ 4.89 ถึง 4.91 พบว่าตัวกำจัดสัญญาณรบกวนสัมประสิทธิ์ป้อนกลับ 20 ตัวที่ขนาดสเตป 4×10^{-10} มีผลตอบสนองค่อนข้างดี กล่าวคือสามารถกำจัดสัญญาณรบกวนรูปไซน์ทั้งที่ความถี่ต่ำกว่าและสูงกว่าสัญญาณเบสแบนด์ออกได้ อย่างไรก็ตามหากเปรียบเทียบผลตอบสนองของตัวกำจัดสัญญาณรบกวนต่อสัญญาณรูปไซน์ความถี่ 50 Hz จะพบว่าตัวกำจัดสัญญาณรบกวนสัมประสิทธิ์ป้อนกลับ 10 ตัวที่ขนาดสเตปเดียวกัน จะมีประสิทธิภาพดีกว่า อาจเนื่องมาจากอันดับของตัวกรองที่เพิ่มมากขึ้น ทำให้ความไวต่อสัมประสิทธิ์มากเกินไป การปรับตัวในแต่ละครั้งจึงมากเกินไป เป็นผลให้ตัวกรองไม่ลู่เข้าสู่ค่าสัมประสิทธิ์ที่ดีที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และบทสรุป

จากการออกแบบตัวกรองปรับตัวได้แบบไอโออาร์ โดยการใช้วิธีการแบบเอาต์พุทเออเรอร์และใช้อัลกอริทึมแบบแอลเอ็มเอส ทำการซิมูเลชันด้วยโปรแกรมเมทแลบในแกนเวลาพบว่าตัวกรองสามารถใช้งานได้ตามต้องการ โดยมีการตอบสนองซึ่งก็คือการลู่เข้าที่ดี สำหรับการนำตัวกรองไปประยุกต์ใช้งานเป็นตัวกำจัดสัญญาณรบกวนและการหาค่าคุณลักษณะของระบบที่ไม่ทราบก็ให้ผลที่ดี กล่าวคือสามารถกำจัดสัญญาณรบกวนได้พอประมาณและมีการลู่เข้าได้ดีตามลำดับ อย่างไรก็ตามพบว่า การที่ตัวกรองจะทำงานได้ดีนั้นขึ้นอยู่กับตัวแปรอื่นๆเช่น ขนาดสเตป ขนาดสัญญาณที่ต้องการกรองออก ต้องมีค่าพอเหมาะและเนื่องจากตัวกรองปรับตัวมีการป้อนกลับของสัญญาณ ทำให้มีโอกาสไม่เสถียรได้ง่าย ซึ่งจากผลการทดลองพบว่าหากขนาดสเตปใหญ่เกินไปหรือมีสัมประสิทธิ์ป้อนกลับมากเกินไป จะทำให้ตัวกรองไม่เสถียร สำหรับการทดลองสร้างตัวกรองในระบบประมวลผลสัญญาณดิจิทัลนั้นพบว่าตัวกรองสามารถทำงานได้ดี หากแต่เนื่องจากความไวของสัมประสิทธิ์ของระบบประมวลผลสัญญาณดิจิทัลนั้นมีมาก ดังนั้นในการจะทำให้ตัวกรองมีเสถียรภาพ ขนาดสเตปที่ใช้จริงๆจึงมีค่าน้อยกว่าขนาดสเตปที่ใช้ในการซิมูเลชันมาก

สำหรับปัญหาที่เกิดขึ้นในการออกแบบและทดลองคือ การซิมูเลชันสำหรับตัวกรองปรับตัวต้องทำการซิมูเลชันในแกนเวลาซึ่งยากกว่าการซิมูเลชันของตัวกรองธรรมดาทั่วไป นอกจากนั้นในการทดลองสร้างเป็นตัวกำจัดสัญญาณรบกวนจริง ยังให้ผลตอบสนองต่อสัญญาณรบกวนสู้ไม่ดีเท่าที่ควร ซึ่งอาจเป็นผลมาจากขนาดสเตปที่ใช้ไม่ใช่ขนาดสเตปที่ดีที่สุด และในการหาขนาดสเตปที่เหมาะสมในการสร้างตัวกรองก็หายาก เนื่องจากมีตัวแปรต่างๆที่ต้องควบคุมมากมาย

สำหรับพัฒนาต่อไปนี้อาจนำทฤษฎีขนาดสเตปขนาดเล็ก (Small Step Size Theory) เข้ามาช่วยในการหาขนาดสเตปที่ดีที่สุด ซึ่งช่วยให้สามารถสร้างตัวกรองที่มีการลู่เข้าที่เร็วและดีกว่าเดิมได้และนอกจากนั้นยังสามารถนำไปประยุกต์ใช้งานจริงดังเช่นตัวอย่างที่ได้กล่าวมาแล้วต่อไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.โปรแกรมซึ่งใช้ในการจำลองขั้นของตัวกรองปรับตัวแบบเอฟไออาร์

```
%---Adaptive FIR Filter showing Sinusoidal response---%
%-----Environment Initial-----%
clear;
fs=input('Type in Sampling Frequency in KHz = ');
t=1/(1000*fs);
freqbase=input('Type in Baseband Frequency in Hz = ');
baseamp=input('Type in Baseband Amplitude (VPP)= ');
noisetype=input('Select Noise Signal Type 1=Sinusoidal 2=Random noise
= ');
noiseamp=input('Type in Noise Amplitude Range (VPP)= ');
if noisetype==1
    freqnoise=input('Type in Noise Frequency in Hz = ');
end
itn=input('Type in Number of Iteration = ');
stepsize=input('Type in filter stepsize = ');
tap=input('Type in Filter Tap = ');

%-----Signal Generation-----%
baseband=baseamp*cos(2*pi*freqbase*t*[1:itn]);
    switch noisetype
case 1
    noise=noiseamp*sin(2*pi*freqnoise*t*[1:itn]);
case 2
    noise=noiseamp*randn(1,itn);
end
x=baseband+noise;

%-----Filtering Routine-----%
w=zeros(1,tap);
y=zeros(1,itn);
e=zeros(1,itn);
d=baseband;

%-----Transient-----%
for i=1:tap-1
    for j=0:tap-1
        if (i-j)>0
            y(i)=y(i)+w(1+j)*x(i-j);
        end
    end

    e(i)=d(i)-y(i);
    esq(i)=e(i)^2;
%-----Zero subbing for non define value-----%
    xt=x(i:-1:1);
    for o=1:tap-i
        xt=[0 xt];
    end

%-----%
    w=w+2*stepsize*e(i)*xt;
end
%-----Steady State-----%
for i=tap:itn
    for j=0:tap-1
        y(i)=y(i)+w(1+j)*x(i-j);
    end
    e(i)=d(i)-y(i);
    w=w+2*stepsize*e(i)*x(i:-1:i-tap+1);
end
```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าในรูปแบบใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

%Plot Out Graph
figure(1);
%Plot Baseband Signal with Noise.
subplot(3,1,1);
plot(x);
axis([0 itn -(baseamp+noiseamp) (baseamp+noiseamp)]);
title('Baseband Signal corrupt with noise');
ylabel('Amplitude')

%Plot out Filtered Signal
subplot(3,1,2);
plot(y);
axis([0 itn -(baseamp+noiseamp) (baseamp+noiseamp)]);
title('Filtered Signal');
ylabel('Amplitude')

%Plot out Error Signal
subplot(3,1,3);
plot(e);
axis([0 itn -(baseamp+noiseamp) (baseamp+noiseamp)]);
title('Error Signal');
xlabel('Number of Iteration');
ylabel('Amplitude')

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข.โปรแกรมซึ่งใช้ในการหิมุเลขันของตัวกรองปรับตัวแบบไอไออาร์

```
%Adaptive IIR Filter showing Sinusoidal response--%
%-----Environment Initial-----%
clear;
fs=input('Type in Sampling Frequency in KHz = ');
t=1/(1000*fs);
freqbase=input('Type in Baseband Frequency in Hz = ');
baseamp=input('Type in Baseband Amplitude (VPP)= ');
noisetype=input('Select Noise Signal Type 1=Sinusoidal 2=Random noise
= ');
noiseamp=input('Type in Noise Amplitude Range (VPP)= ');
if noisetype==1
    freqnoise=input('Type in Noise Frequency in Hz = ');
end
itn=input('Type in Number of Iteration = ');
stepsize=input('Type in filter stepsize = ');
tap=input('Type in Filter Tap = ');
```

```
%-----Signal Generation-----%
baseband=baseamp*cos(2*pi*freqbase*t*[1:itn]);
    switch noisetype
case 1
    noise=noiseamp*sin(2*pi*freqnoise*t*[1:itn]);
case 2
    noise=noiseamp*randn(1,itn);
end
x=baseband+noise;
```

```
%-----Filtering Routine-----%
wa=zeros(1,tap+1);
wb=zeros(1,tap);
alpha=zeros(1,itn);
beta=zeros(1,itn);
y=zeros(1,itn);
e=zeros(1,itn);
d=baseband;
nua=zeros(1,tap+1);
nub=zeros(1,tap);
```

```
%-----Transient-----%
for i=1:tap
    for j=0:tap
        if (i-j)>0
            y(i)=y(i)+wa(1+j)*x(i-j);
        end
    end
    for k=1:tap
        if (i-k)>0
            y(i)=y(i)+wb(k)*y(i-k);
        end
    end
    e(i)=d(i)-y(i);
```

```
    for l=1:tap
        if (i-l)>0
            alpha(i)=alpha(i)+wb(l)*alpha(i-l);
        end
    end
```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

alpha(i)=alpha(i)+x(i);

for m=1:tap
    if (i-m)>0
        beta(i)=beta(i)+wb(m)*beta(i-m);
    end
end
beta(i)=beta(i)+y(i);

for n=0:tap
    if (i-n)>0
        nua(1+n)=alpha(i-n);
    end
end
for o=0:tap-1
    if (i-o)>0
        nub(1+o)=beta(i-o);
    end
end

%-----%
wa=wa+2*stepsize*e(i)*nua;
wb=wb+2*stepsize*e(i)*nub;
end

%-----Steady State-----%
for i=tap+1:itn
    for j=0:tap
        y(i)=y(i)+wa(1+j)*x(i-j);
    end
    for k=1:tap
        y(i)=y(i)+wb(k)*y(i-k);
    end

    e(i)=d(i)-y(i);

    for l=1:tap
        alpha(i)=alpha(i)+wb(l)*alpha(i-l);
    end
    alpha(i)=alpha(i)+x(i);

    for m=1:tap
        beta(i)=beta(i)+wb(m)*beta(i-m);
    end
    beta(i)=beta(i)+y(i);

    for n=0:tap
        nua(1+n)=alpha(i-n);
    end
    for o=0:tap-1
        nub(1+o)=beta(i-o);
    end

%-----%
wa=wa+2*stepsize*e(i)*nua;
wb=wb+2*stepsize*e(i)*nub;
end

```

%Plot Out Graph

เอกสาร figure(1); ที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

%Plot Baseband Signal with Noise.

ไม่ว่าอย่างไรก็ตามห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
plot(x);
axis([0 itn -(baseamp+noiseamp) (baseamp+noiseamp)]);
title('Baseband Signal corrupt with noise');
ylabel('Amplitude')

%Plot out Filtered Signal
subplot(3,1,2);
plot(y);
axis([0 itn -(baseamp+noiseamp) (baseamp+noiseamp)]);
title('Filtered Signal');
ylabel('Amplitude')

%Plot out Error Signal
subplot(3,1,3);
plot(e);
axis([0 itn -(baseamp+noiseamp) (baseamp+noiseamp)]);
title('Error Signal');
xlabel('Number of Iteration')
ylabel('Amplitude')
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค.โปรแกรมซึ่งใช้ในการขมิลงนของตัวกำจัดสัญญาณรบกวน

```
%---Noise Cancellation showing Sinusoidal response---%
%-----Environment Initial-----%
clear;
fs=input('Type in Sampling Frequency in KHz = ');
t=1/(1000*fs);
freqbase=input('Type in Baseband Frequency in Hz = ');
baseamp=input('Type in Baseband Amplitude (VPP)= ');
noisetype=input('Select Noise Signal Type 1=Sinusoidal 2=Random noise
= ');
noiseamp=input('Type in Noise Amplitude Range (VPP)= ');
if noisetype==1
    freqnoise=input('Type in Noise Frequency in Hz = ');
end
itn=input('Type in Number of Iteration = ');
stepsize=input('Type in filter stepsize = ');
tap=input('Type in Filter Tap = ');
```

```
%-----Signal Generation-----%
baseband=baseamp*cos(2*pi*freqbase*t*[1:itn]);
    switch noisetype
case 1
    noise=noiseamp*sin(2*pi*freqnoise*t*[1:itn]);
case 2
    noise=noiseamp*randn(1,itn);
end
x=noise;
```

```
%-----Filtering Routine-----%
wa=zeros(1,tap+1);
wb=zeros(1,tap);
alpha=zeros(1,itn);
beta=zeros(1,itn);
y=zeros(1,itn);
e=zeros(1,itn);
d=baseband+noise;
nua=zeros(1,tap+1);
nub=zeros(1,tap);
```

```
%-----Transient-----%
for i=1:tap
    for j=0:tap
        if (i-j)>0
            y(i)=y(i)+wa(1+j)*x(i-j);
        end
    end
    for k=1:tap
        if (i-k)>0
            y(i)=y(i)+wb(k)*y(i-k);
        end
    end
    e(i)=d(i)-y(i);
```

```
    for l=1:tap
        if (i-l)>0
            alpha(i)=alpha(i)+wb(l)*alpha(i-l);
        end
    end
```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

alpha(i)=alpha(i)+x(i);

for m=1:tap
    if (i-m)>0
        beta(i)=beta(i)+wb(m)*beta(i-m);
    end
end
beta(i)=beta(i)+y(i);

for n=0:tap
    if (i-n)>0
        nua(1+n)=alpha(i-n);
    end
end
for o=0:tap-1
    if (i-o)>0
        nub(1+o)=beta(i-o);
    end
end

%-----%
wa=wa+2*stepsize*e(i)*nua;
wb=wb+2*stepsize*e(i)*nub;
end

%-----Steady State-----%
for i=tap+1:itn
    for j=0:tap
        y(i)=y(i)+wa(1+j)*x(i-j);
    end
    for k=1:tap
        y(i)=y(i)+wb(k)*y(i-k);
    end

    e(i)=d(i)-y(i);

    for l=1:tap
        alpha(i)=alpha(i)+wb(l)*alpha(i-l);
    end
    alpha(i)=alpha(i)+x(i);

    for m=1:tap
        beta(i)=beta(i)+wb(m)*beta(i-m);
    end
    beta(i)=beta(i)+y(i);

    for n=0:tap
        nua(1+n)=alpha(i-n);
    end
    for o=0:tap-1
        nub(1+o)=beta(i-o);
    end

%-----%
wa=wa+2*stepsize*e(i)*nua;
wb=wb+2*stepsize*e(i)*nub;
end

```

```

%Plot Out Graph
figure(1);

```

เอกสาร **%Plot Baseband Signal with Noise.** การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ **plot(d);** ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
axis([0 itn - (baseamp+noiseamp) (baseamp+noiseamp)]);  
title('Baseband Signal corrupt with noise');  
ylabel('Amplitude')  
  
%Plot out Filtered Signal  
subplot(2,1,2);  
plot(e);  
axis([0 itn - (baseamp+noiseamp) (baseamp+noiseamp)]);  
title('Filtered Signal');  
xlabel('Number of Iteration')  
ylabel('Amplitude')
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ง.โปรแกรมซึ่งใช้ในการหาค่าสัมประสิทธิ์ของตัวกรองปรับตัวในการหาคุณลักษณะของระบบที่ไม่ทราบ

```
%-System Identification of Sinusoidal response system-%
%-----Environment Initial-----%
clear;
fs=input('Type in Sampling Frequency in KHz = ');
t=1/(1000*fs);
freqbase=input('Type in Input Signal in Hz = ');
baseamp=input('Type in Input Signal (VPP)= ');
h1=input('Type in Unknown System Numerator Coefficient = ');
h2=input('Type in Unknown System Denominator Coefficient = ');
itn=input('Type in Number of Iteration = ');
stepsize=input('Type in adaptation stepsize = ');
tap=input('Type in Filter Tap = ');

%-----Signal Generation-----%
x=baseamp*cos(2*pi*freqbase*t*[1:itn]);
ys=filter(h1,h2,x);
%-----Filtering Routine-----%
wa=zeros(1,tap+1);
wb=zeros(1,tap);
alpha=zeros(1,itn);
beta=zeros(1,itn);
y=zeros(1,itn);
e=zeros(1,itn);
d=ys;
nua=zeros(1,tap+1);
nub=zeros(1,tap);

%-----Transient-----%
for i=1:tap
    for j=0:tap
        if (i-j)>0
            y(i)=y(i)+wa(1+j)*x(i-j);
        end
    end
    for k=1:tap
        if (i-k)>0
            y(i)=y(i)+wb(k)*y(i-k);
        end
    end
    end
    e(i)=d(i)-y(i);

    for l=1:tap
        if (i-l)>0
            alpha(i)=alpha(i)+wb(l)*alpha(i-l);
        end
    end
    alpha(i)=alpha(i)+x(i);

    for m=1:tap
        if (i-m)>0
            beta(i)=beta(i)+wb(m)*beta(i-m);
        end
    end
    beta(i)=beta(i)+y(i);

    for n=0:tap
        if (i-n)>0
            nua(1+n)=alpha(i-n);
        end
    end
end
```

เอกสารนี้เป็นเอกสารที่เผยแพร่ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ กรุณาแจ้งชื่อและที่อยู่ของเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        end
    end
    for o=0:tap-1
        if (i-o)>0
            nub(1+o)=beta(i-o);
        end
    end
end
%-----%
    wa=wa+2*stepsize*e(i)*nua;
    wb=wb+2*stepsize*e(i)*nub;
end

%-----Steady State-----%
for i=tap+1:itn
    for j=0:tap
        y(i)=y(i)+wa(1+j)*x(i-j);
    end
    for k=1:tap
        y(i)=y(i)+wb(k)*y(i-k);
    end

    e(i)=d(i)-y(i);

    for l=1:tap
        alpha(i)=alpha(i)+wb(l)*alpha(i-l);
    end
    alpha(i)=alpha(i)+x(i);

    for m=1:tap
        beta(i)=beta(i)+wb(m)*beta(i-m);
    end
    beta(i)=beta(i)+y(i);

    for n=0:tap
        nua(1+n)=alpha(i-n);
    end
    for o=0:tap-1
        nub(1+o)=beta(i-o);
    end
end
%-----%
    wa=wa+2*stepsize*e(i)*nua;
    wb=wb+2*stepsize*e(i)*nub;
end

%Plot Out Graph
figure(1);
%Plot System Resopnse.
subplot(3,1,1);
plot(ys);
axis([0 itn -(baseamp+1) (baseamp+1)]);
title('System Response');
ylabel('Amplitude')

%Plot out Adaptive Filter Response.
subplot(3,1,2);
plot(y);
axis([0 itn -(baseamp+1) (baseamp+1)]);
title('Adaptive Filter Response');
ylabel('Amplitude');
%Plot out Error Signal

```

เอกสารนี้เป็นลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ให้นำไปเผยแพร่หรือดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
subplot(3,1,3);  
plot(e);  
axis([0 itn -(baseamp+1) (baseamp+1)]);  
title('Error Signal');  
xlabel('Number of Iteration')  
ylabel('Amplitude')
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.โปรแกรมซึ่งใช้ในการทดสอบการนำสัญญาณเข้าออก

;IO.ASM - NORMAL INPUT OUTPUT TESTING ROUTINE

```
.start    ".text",0x809900 ;where text begins
.start    ".data",0x809C00 ;where data begins
.include  "AICCOM31.ASM"   ;AIC communications routines

.data                                ;assemble into data section

*-----COMMUNICATION ROUTINE VARIABLE-----*
AICSEC    .word    A14h,1h,3E7Eh,073h;For AIC,Fs = 20161.29 Hz
          .entry   BEGIN                ;start of code
          .text    ;assemble into text section
BEGIN     LDP      AICSEC
          CALL    AICSET                ;initialize AIC
RUN       CALL    IOPRI                ;get input sample from PRIMARY IN
          FLOAT   R6,R4                ;transfer input sample into R3
          FIX     R4,R7                ;R7 = R4 in integer and output
          BR      RUN                ;repeat with next sample
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข.โปรแกรมตัวกรองปรับตัวแบบเฟอไออาร์

```

;FIR50.ASM- ADAPTIVE FIR FILTER WITH 50 COEFFICIENT
        .start    ".text",0x809900 ;where text begins
        .start    ".data",0x809C00 ;where data begins
        .include  "AICCOM31.ASM"   ;AIC communications routines

        .data
                                ;assemble into data section
*-----COMMUNICATION ROUTINE VARIABLE-----*
AICSEC  .word      0A14h,1h,3E7Eh,0F3h;For AIC,Fs = 20161.29 Hz

*-----FILTERING AND ADAPTATION ROUTINE VARIABLE-----*
INPUT_ADDR .word    INPUT+LENGTH-1 ;last address of input samples
WNA_ADDR  .word    COEFFA          ;address of coefficients
ERF_ADDR  .word    ERR_FUNC        ;address of error function
ERR_FUNC  .float   0                ;initial error function
STEP      .float   2.5E-12         ;step size constant
LENGTH    .set     50              ;set filter length

COEFFA:
        .loop     LENGTH           ;memory buffer for coefficients
        .float   0                 ;loop length times
        .endloop   ;init coefficients to zero
                                ;end of loop

INPUT
        .brstart  "XN_BUFF",128    ;memory buffer for input samples
        .sect     "XN_BUFF"        ;section for input samples
        .loop     LENGTH           ;loop length times
        .float   0                 ;initialize input samples
        .endloop   ;end of loop

*----- FILTERING ROUTINE -----*
        .entry    BEGIN            ;start of code
        :text
                                ;assemble into text section
BEGIN   LDP        WNA_ADDR         ;init to data page 128
        CALL     AICSET            ;initialize AIC
        LDI     @ERF_ADDR,AR6     ;error function address to AR6
        LDI     LENGTH,BK        ;load BK = filter length
        LDI     @WNA_ADDR,AR0     ;load coefficient address to AR0
        LDI     @INPUT_ADDR,AR1  ;input samples address to AR1

FILTER  CALL     IOPRI             ;get input samples at PRIMARY IN
        FLOAT   R6,R3             ;transfer input samples into R3
        CALL     IOAUX            ;get desired samples at AUX IN
        FLOAT   R6,R4             ;transfer desired samples to R4
        STF     R3,*AR1++%        ;store input sample x(n)
        LDI     @WNA_ADDR,AR0     ;coefficients address to AR0
        LDF     0,R2              ;R2 = 0
        RPTS    LENGTH-1         ;repeat next 2 instruction
        MPYF3   *AR0++,*AR1++%,R0;w(n-i)*x(n-i)
        ||
        ADDF3   R0,R2,R2          ;accumulate
        ADDF3   R0,R2,R0          ;add last product = y(n) -> R0
        FIX     R0,R7             ;R7=R0 in integer and output

*----- ERROR ESTIMATE -----*
        SUBF3   R0,R4,R0          ; error = desired - y(n)
        MPYF   @STEP,R0           ;R0=Error Fucntion
        STF     R0,*AR6           ;store error function

        LDI     LENGTH-2,RC       ;set repeat counter register
        LDI     @WNA_ADDR,AR0     ;coefficients address to AR0
        CALL    ADAPT              ;call ADAPTATION ROUTINE
    
```

เอกสารนี้เป็นเอกสารที่... LDI LENGTH-2,RC ;set repeat counter register
 LDI @WNA_ADDR,AR0 ;coefficients address to AR0
 CALL ADAPT ;call ADAPTATION ROUTINE

```

BR          FILTER          ;repeat with next sample
;-----ADAPTATION ROUTINE-----*
ADAPT      MPYF3      *AR6, *AR1+%, R0 ;error function*x(n-i) -> R0
           LDF        *AR0, R3      ;w(N-1) -> R3
           RPTB      LOOP_END      ;repeat length-2 times
           MPYF3      *AR6, *AR1+%, R0 ;error function*x(n-i) -> R0
           ADDF3      R3, R0, R2    ;w(N-1-i)+error*x(n-1-i)
           LDF        *+AR0 (1), R3 ;load subsequent w(k) -> R3
           STF        R2, *AR0++    ;update coefficient
           ADDF3      R3, R0, R2    ;w(n+1)=w(n)+error function*x(n)
           STF        R2, *AR0     ;update coefficient
           RETS       ;return from subroutine

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค.โปรแกรมตัวกรองปรับตัวแบบไออาร์

;IIR4.ASM - ADAPTIVE IIR FILTERING WITH 4 FEEDBACK TAP

```
.start ".text",0x809900 ;where text begins
.start ".data",0x809C00 ;where data begins
.include "AICCOM31.ASM" ;AIC communications routines

.data ;assemble into data section
*-----COMMUNICATION ROUTINE VARIABLE-----*
AICSEC .word 0A14h,1h,3E7Eh,0F3h;For AIC,Fs = 20161.29 Hz

*-----FILTERING AND ADAPTATION ROUTINE VARIABLE-----*
IN_ADDR .word INPUT ;address of input samples
OUT_ADDR .word OUTPUT ;address of output samples
WNA_ADDR .word COEFFA ;address of coefficients A
WNB_ADDR .word COEFFB ;address of coefficients B
ALPHA_ADD .word ALPHA ;address of alpha parameter
BETA_ADD .word BETA ;address of beta parameter
ERF_ADDR .word ERR_FUNC ;address of error function
ERR_FUNC .float 0 ;initialize error function
STEP .float 8E-8 ;step size constant
IN_LENGTH .set 5 ;set forward filter length
OUT_LENGTH .set 4 ;set feedback filter length

ALPHA:
.loop IN_LENGTH ;loop length times
.float 0 ;init alpha parameter to zero
.endloop ;end of loop

BETA:
.loop IN_LENGTH ;loop length times
.float 0 ;init beta parameter to zero
.endloop ;end of loop

COEFFA:
.loop IN_LENGTH ;loop length times
.float 0 ;init forward coeff. to zero
.endloop ;end of loop

COEFFB:
.loop OUT_LENGTH ;loop length times
.float 0 ;init feedback coeff. to zero
.endloop ;end of loop

INPUT:
.loop IN_LENGTH ;loop length times
.float 0 ;init input samples to zero
.endloop ;end of loop

OUTPUT:
.loop OUT_LENGTH ;loop length times
.float 0 ;init output samples to zero
.endloop ;end of loop

*----- FILTERING ROUTINE -----*
.entry BEGIN ;Start of code
.text ;Assemble into text section
BEGIN LDP WNA_ADDR ;Initialize data page
CALL AICSET ;Initialize AIC
LDI @ERF_ADDR,AR6 ;error function address to AR6

FORWARD LDI @WNA_ADDR,AR0 ;coefficient address A(N) to AR0
LDI @IN_ADDR,AR1 ;input sample address to AR1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาค้นคว้าเท่านั้น ไม่ควรนำออกจำหน่าย

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL      IOPRI                      ;Get input sample at Primary In
FLOAT    R6,R4                      ;Transfer input sample into R4
CALL      IOAUX                      ;Get desired sample at Aux In
FLOAT    R6,R5                      ;Transfer desired sample into R5

LDF      *+AR1(3),R1                ;INPUT SAMPLE UPDATE
STF      R1,*+AR1(4)                ;store x(n-4)
LDF      *+AR1(2),R1                ;
STF      R1,*+AR1(3)                ;store x(n-3)
LDF      *+AR1(1),R1                ;
STF      R1,*+AR1(2)                ;store x(n-2)
LDF      *AR1,R1                    ;
STF      R1,*+AR1(1)                ;store x(n-1)
STF      R4,*AR1                    ;store x(n)

LDF      0,R0                       ;R0 = 0 Refresh register
LDF      0,R2                       ;R1 = 0 Refresh register
RPTS     IN_LENGTH-1                ;Repeat next 2 instruction
;IN_LENGTH-1 times
MPYF3   *AR0++,*AR1++,R0            ;A(N-i)*X(n-(N-i))
|| ADDF3  R2,R0,R2                    ;Accumulate
ADDF3   R2,R0,R2                    ;R2 = forward response

BACKWARD LDI @WNB_ADDR,AR2           ;coefficient address B(N) to AR2
LDI     @OUT_ADDR,AR3              ;output sample address to AR3
LDF     0,R0                       ;R2 = 0 Refresh register
LDF     0,R3                       ;R2 = 0 Refresh register
RPTS    OUT_LENGTH-1              ;Repeat next 2 instruction
;OUT_LENGTH-1 times
|| MPYF3  *AR2++,*AR3++,R0          ;B(N-i)*Y(n-(N-i))
ADDF3   R3,R0,R3                  ;Accumulate
ADDF3   R3,R0,R3                  ;R3 = feedback response

SUM      ADDF3  R2,R3,R3            ;Y(N) = forward + feedback
FIX      R3,R7                      ;R7 = R3 in integer and output
LDI     @OUT_ADDR,AR3              ;OUTPUT sample update
LDF     *+AR3(2),R0                ;
STF     R0,*+AR3(3)                ;y(n-4)
LDF     *+AR3(1),R0                ;
STF     R0,*+AR3(2)                ;y(n-3)
LDF     *AR3,R0                    ;
STF     R0,*+AR3(1)                ;y(n-2)
STF     R3,*AR3                    ;y(n-1)

*----- ERROR ESTIMATE -----*
ERROR    SUBF3  R3,R5,R0            ;error = desired - y(n)
MPYF     @STEP,R0                  ;R0 = error * stepsize
STF      R0,*AR6                  ;Store error function

*----- ADAPTATION -----*

CALL     PARA_UP                   ;Call parameter update routine
CALL     ADAPT                      ;Call adaptation subroutine
BR       FORWARD                   ;Repeat with next sample

*----- PARAMETER UPDATE SUBROUTINE -----*
PARA_UP  LDF     0,R0                ;Refresh register
LDF     0,R1                      ;Refresh register
LDI     @ALPHA_ADD,AR4             ;Alpha parameter address to AR4

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษา; Refresh register ใช้เพื่อไปใช้ประโยชน์ด้านการศึกษา
 ไม่ว่ากรณีใดๆ ทั้งสิ้น; Refresh register ใช้เพื่อไปใช้ประโยชน์ด้านการศึกษา
 LD I @ALPHA_ADD,AR4 ; Alpha parameter address to AR4

```

LDI    @BETA_ADD,AR5      ;Beta parameter address to AR5
LDF    0,R2                ;Refresh register
LDI    @WNB_ADDR,AR2      ;coefficient B address to AR2
RPTS   OUT_LENGTH-1       ;Repeat next 2 instruction
                                ;OUT_LENGTH-1 times
MPYF3  *AR2++,*AR4++,R0    ;Alpha parameter calculation
ADDF3  R2,R0,R2            ;Accumulate
ADDF3  R2,R0,R2
ADDF3  R4,R2,R4            ;Current alpha parameter

LDF    0,R2                ;Refresh register
LDI    @WNB_ADDR,AR2      ;coefficient B address to AR2
RPTS   OUT_LENGTH-1       ;Repeat next 2 instruction
                                ;OUT_LENGTH-1 times
MPYF3  *AR2++,*AR5++,R1    ;Beta parameter calculation
ADDF3  R2,R1,R2            ;Accumulate
ADDF3  R2,R1,R2
ADDF3  R3,R2,R3            ;Current beta parameter

LDI    @ALPHA_ADD,AR4     ;update alpha parameter
LDF    *+AR4(3),R0
STF    R0,*+AR4(4)
LDF    *+AR4(2),R0
STF    R0,*+AR4(3)
LDF    *+AR4(1),R0
STF    R0,*+AR4(2)
LDF    *AR4,R0
STF    R0,*+AR4(1)
STF    R4,*AR4

LDI    @BETA_ADD,AR5     ;update beta parameter
LDF    *+AR5(3),R0
STF    R0,*+AR5(4)
LDF    *+AR5(2),R0
STF    R0,*+AR5(3)
LDF    *+AR5(1),R0
STF    R0,*+AR5(2)
LDF    *AR5,R0
STF    R0,*+AR5(1)
STF    R3,*AR5

RETS

```

----- ADAPTATION SUBROUTINE -----

```

ADAPT  LDI    @ALPHA_ADD,AR4      ;Alpha parameter address to AR4
        LDI    @BETA_ADD,AR5      ;Beta parameter address to AR5
        LDI    @WNA_ADDR,AR0      ;coefficient address A(N) to AR0
        LDI    @WNB_ADDR,AR2      ;coefficient address B(N) to AR2

```

```

        LDI    IN_LENGTH-1,RC     ;repeat counter = IN_LENGTH-1
RPTB   ADAPT_A                    ;Repeat until reach ADAPT_A
MPYF3  *AR4++,*AR6,R0             ;Coefficient A adaptation
ADDF3  *AR0++,R0,R1
ADAPT_A STF   R1,*-AR0(1)

```

```

        LDI    OUT_LENGTH-1,RC    ;repeat counter = OUT_LENGTH-1
RPTB   ADAPT_B                    ;Repeat until reach ADAPT_B
MPYF3  *++AR5,*AR6,R0             ;Coefficient B adaptation
ADDF3  *AR2++,R0,R1
ADAPT_B STF   R1,*-AR2(1)
RETS

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่นับผูกพันให้ใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น ห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ง.โปรแกรมตัวกรองปรับตัวแบบเอฟไออาร์

;NOISE10.ASM - ADAPTIVE NOISE CANCELLATION WITH 10 FEEDBACK TAP

```
.start ".text",0x809900 ;where text begins
.start ".data",0x809C00 ;where data begins
.include "AICCOM31.ASM" ;AIC communications routines

.data ;assemble into data section
*-----COMMUNICATION ROUTINE VARIABLE-----*
AICSEC .word 0A14h,1h,3E7Eh,0F3h;For AIC,Fs = 20161.29 Hz

*-----FILTERING AND ADAPTATION ROUTINE VARIABLE-----*
IN_ADDR .word INPUT ;last address of input samples
OUT_ADDR .word OUTPUT ;address of output samples
WNA_ADDR .word COEFFA ;address of coefficients A
WNB_ADDR .word COEFFB ;address of coefficients B
ALPHA_ADD .word ALPHA ;address of alpha parameter
BETA_ADD .word BETA ;address of beta parameter
ERF_ADDR .word ERR_FUNC ;address of error function
ERR_FUNC .float 0 ;initialize error function
STEP .float 4E-10 ;stepsize constant
IN_LENGTH .set 11 ;set forward filter length
OUT_LENGTH .set 10 ;set feedback filter length

ALPHA:
    .loop IN_LENGTH ;loop length times
    .float 0 ;init alpha parameter to zero
    .endloop ;end of loop

BETA:
    .loop IN_LENGTH ;loop length times
    .float 0 ;init beta parameter to zero
    .endloop ;end of loop

COEFFA:
    .loop IN_LENGTH ;loop length times
    .float 0 ;init forward coeff.to zero
    .endloop ;end of loop

COEFFB:
    .loop OUT_LENGTH ;loop length times
    .float 0 ;init feedback coeff. to zero
    .endloop ;end of loop

INPUT:
    .loop IN_LENGTH ;loop length times
    .float 0 ;initialize input samples
    .endloop ;end of loop

OUTPUT:
    .loop OUT_LENGTH ;loop length times
    .float 0 ;initialize output samples
    .endloop ;end of loop

*----- FILTERING ROUTINE -----*
    .entry BEGIN ;start of code
    .text ;assemble into text section
BEGIN LDP WNA_ADDR ;init to data page 128
CALL AICSET ;initialize AIC
LDI @ERF_ADDR,AR6 ;error function address -> AR6

FORWARD LDI @WNA_ADDR,AR0 ;coeff. address A(N-1) -> AR0
LDI @IN_ADDR,AR1 ;input sample address -> AR1
```

เอกสารนี้เป็นเอกสารที่ LDI ไม่ไว้สำหรับ การศึกษาวิจัย ไม่ควรนำมาใช้เพื่อประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงชื่อของเอกสารทุกครั้งที่มีกรนำไปใช้

```

CALL    IOAUX                ;get input sample from AUX IN
FLOAT   R6,R4                ;transfer input sample into R3
CALL    IOPRI                ;get desired sample from PRI IN
FLOAT   R6,R5                ;transfer desired sample into R4

LDF     *+AR1(9),R1          ;INPUT SAMPLE UPDATE
STF     R1,*+AR1(10)         ;store x(n-10)
LDF     *+AR1(8),R1          ;
STF     R1,*+AR1(9)          ;store x(n-9)
LDF     *+AR1(7),R1          ;
STF     R1,*+AR1(8)          ;store x(n-8)
LDF     *+AR1(6),R1          ;
STF     R1,*+AR1(7)          ;store x(n-7)
LDF     *+AR1(5),R1          ;
STF     R1,*+AR1(6)          ;store x(n-6)
LDF     *+AR1(4),R1          ;
STF     R1,*+AR1(5)          ;store x(n-5)
LDF     *+AR1(3),R1          ;
STF     R1,*+AR1(4)          ;store x(n-4)
LDF     *+AR1(2),R1          ;
STF     R1,*+AR1(3)          ;store x(n-3)
LDF     *+AR1(1),R1          ;
STF     R1,*+AR1(2)          ;store x(n-2)
LDF     *AR1,R1              ;
STF     R1,*+AR1(1)          ;store x(n-1)
STF     R4,*AR1              ;store x(n)

LDF     0,R0                  ;R0 = 0 REFRESH REGISTER
LDF     0,R2                  ;R1 = 0 REFRESH REGISTER
RPTS    IN_LENGTH-1          ;next 2 instr (IN_LENGTH) times
MPYF3   *AR0++,*AR1++,R0      ;A(N-i)*X(n-(N-i))
ADDF3   R2,R0,R2              ;accumulate
ADDF3   R2,R0,R2              ;R2 = forward filtering

||
BACKWARD
LDI     @WNB_ADDR,AR2         ;coeff. address B(N-1) -> AR2
LDI     @OUT_ADDR,AR3        ;OUTPUT sample address -> AR3

LDF     0,R0                  ;R2 = 0 REFRESH REGISTER
LDF     0,R3                  ;R2 = 0 REFRESH REGISTER
RPTS    OUT_LENGTH-1         ;next 2 instr (OUT_LENGTH) times
MPYF3   *AR2++,*AR3++,R0     ;B(N-i)*Y(n-(N-i))
ADDF3   R3,R0,R3              ;accumulate
ADDF3   R3,R0,R3              ;R3 = feedback filtering

||
SUM
ADDF3   R2,R3,R3              ;Y(N) = forward + feedback

LDI     @OUT_ADDR,AR3        ;OUTPUT sample update
LDF     *+AR3(8),R0
STF     R0,*+AR3(9)          ;y(n-10)
LDF     *+AR3(7),R0
STF     R0,*+AR3(8)          ;y(n-9)
LDF     *+AR3(6),R0
STF     R0,*+AR3(7)          ;y(n-8)
LDF     *+AR3(5),R0
STF     R0,*+AR3(6)          ;y(n-7)
LDF     *+AR3(4),R0
STF     R0,*+AR3(5)          ;y(n-6)
LDF     *+AR3(3),R0
STF     R0,*+AR3(4)          ;y(n-5)
LDF     *+AR3(2),R0
STF     R0,*+AR3(3)          ;y(n-4)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น ลิขสิทธิ์ทั้งห้าประการนี้เหนือหาและต้องอ้างถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LDF      *+AR3 (1),R0
STF      R0,*+AR3 (2)      ;y(n-3)
LDF      *AR3,R0
STF      R0,*+AR3 (1)      ;y(n-2)
STF      R3,*AR3           ;y(n-1)

```

----- ERROR ESTIMATE -----

```

ERROR    SUBF3   R3,R5,R0      ;error = desire - total response
          FIX     R0,R7         ;OUTPUT FILTERED SIGNAL
          MPYF    @STEP,R0      ;R0 = error * stepsize
          STF     R0,*AR6       ;store error function

```

----- ADAPTATION -----

```

CALL     PARA_UP      ;call parameter update routine
CALL     ADAPT        ;call adaptation subroutine
BR       FORWARD     ;repeat with next sample

```

----- PARAMETER UPDATE SUBROUTINE -----

```

PARA_UP  LDF      0,R0          ;REFRESH REGISTER
          LDF      0,R1          ;REFRESH REGISTER
          LDI      @ALPHA_ADD,AR4 ;Alpha parameter address
          LDI      @BETA_ADD,AR5  ;Beta parameter address

          LDF      0,R2          ;REFRESH REGISTER
          LDI      @WNB_ADDR,AR2
          RPTS     OUT_LENGTH-1
          MPYF3    *AR2++,*AR4++,R0 ;alpha parameter calculation
          ADDF3    R2,R0,R2
          ADDF3    R2,R0,R2
          ADDF3    R4,R2,R4      ;CURRENT ALPHA PARAMETER

          LDF      0,R2          ;REFRESH REGISTER
          LDI      @WNB_ADDR,AR2
          RPTS     OUT_LENGTH-1
          MPYF3    *AR2++,*AR5++,R1 ;beta parameter calculation
          ADDF3    R2,R1,R2
          ADDF3    R2,R1,R2
          ADDF3    R3,R2,R3      ;CURRENT BETA PARAMETER

          LDI      @ALPHA_ADD,AR4 ;update alpha parameter
          LDF      *+AR4 (9),R0
          STF      R0,*+AR4 (10)
          LDF      *+AR4 (8),R0
          STF      R0,*+AR4 (9)
          LDF      *+AR4 (7),R0
          STF      R0,*+AR4 (8)
          LDF      *+AR4 (6),R0
          STF      R0,*+AR4 (7)
          LDF      *+AR4 (5),R0
          STF      R0,*+AR4 (6)
          LDF      *+AR4 (4),R0
          STF      R0,*+AR4 (5)
          LDF      *+AR4 (3),R0
          STF      R0,*+AR4 (4)
          LDF      *+AR4 (2),R0
          STF      R0,*+AR4 (3)
          LDF      *+AR4 (1),R0
          STF      R0,*+AR4 (2)
          LDF      *AR4,R0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น หากมีข้อสงสัยหรือต้องการข้อมูลเพิ่มเติม กรุณาติดต่อเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

STF      R0, *+AR4 (1)
STF      R4, *AR4

LDI      @BETA_ADD, AR5          ;update beta parameter
LDF      *+AR5 (9), R0
STF      R0, *+AR5 (10)
LDF      *+AR5 (8), R0
STF      R0, *+AR5 (9)
LDF      *+AR5 (7), R0
STF      R0, *+AR5 (8)
LDF      *+AR5 (6), R0
STF      R0, *+AR5 (7)
LDF      *+AR5 (5), R0
STF      R0, *+AR5 (6)
LDF      *+AR5 (4), R0
STF      R0, *+AR5 (5)
LDF      *+AR5 (3), R0
STF      R0, *+AR5 (4)
LDF      *+AR5 (2), R0
STF      R0, *+AR5 (3)
LDF      *+AR5 (1), R0
STF      R0, *+AR5 (2)
LDF      *AR5, R0
STF      R0, *+AR5 (1)
STF      R3, *AR5

RETS

*----- ADAPTATION SUBROUTINE -----*
ADAPT    LDI      @ALPHA_ADD, AR4      ;Alpha parameter address
         LDI      @BETA_ADD, AR5      ;Beta parameter address
         LDI      @WNA_ADDR, AR0      ;coeff. address A(N-1) ->AR0
         LDI      @WNB_ADDR, AR2      ;coeff. address B(N-1) ->AR2

         LDI      IN_LENGTH-1, RC
         RPTB     ADAPT_A
         MPYF3    *AR4++, *AR6, R0     ;coefficient A adaptation
         ADDF3    *AR0++, R0, R1
ADAPT_A  STF      R1, *-AR0 (1)

         LDI      OUT_LENGTH-1, RC
         RPTB     ADAPT_B
         MPYF3    *++AR5, *AR6, R0     ;coefficient B adaptation
         ADDF3    *AR2++, R0, R1
ADAPT_B  STF      R1, *-AR2 (1)

RETS

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จ.โปรแกรมซึ่งใช้ในการควบคุมการนำสัญญาณเข้าออก

*AICCOM31.ASM - AIC COMMUNICATION ROUTINES - POLLING OR INTERRUPT

```

.data
PBASE .word 808000h ;assemble into data section
SETSP .word 0E970300h ;peripheral base address
ATABLE .word AICSEC ;serial port set-up data
        .text ;SP0 AIC init table address
AICSET PUSH ARO ;assemble into text section
        PUSH AR0 ;save ARO
        PUSH AR1 ;save AR1
        PUSH R0 ;save R0
        PUSH R1 ;save R1
        LDI @PBASE,AR0 ;AR0 -> 808000h
        LDI 1,R0 ;timer CLK=H1/2*(AIC master CLK)
        STI R0,*+AR0(28h) ;timerperiod reg(TCLK0=6.25 MHZ)
        LDI 03C1h,R0 ;init timer global register
        STI R0,*+AR0(20h) ;reset timer
        LDI 62h,IOF ;AIC reset = 0
        LDI @ATABLE,AR1 ;AR1 -> AIC init data
        RPTS 99 ;repeat next instr 100 times
        NOP ;keep IOF low for a while
        LDI 131h,R0 ;X & R port control reg. data
        STI R0,*+AR0(42h) ;FSX/DX/CLKX=SP operational pins
        STI R0,*+AR0(43h) ;FSR/DR/CLKR=SP operational pins
        LDI @SETSP,R0 ;RESET->SP:16 bits,ext clks,std
        STI R0,*+AR0(40h) ;FSX=output&INT enable SP global
        ;reg
        LDI 0,R0 ;R0 = 0
        STI R0,*+AR0(48h) ;clear serial port XMIT register
        OR 06h,IOF ;bring AIC out of reset
        LDI 03h,RC ;RC=3 to transmit 4 values
        RPTB SECEND ;repeat 4 data transmit
        CALL TWAIT ;wait for data transmit
        LDI 03h,R0 ;value for secondary TX request
        STI R0,*+AR0(48h) ;secondary XMIT request to AIC
        CALL TWAIT ;wait for data transmit
        LDI *AR1++(1),R0 ;AR1 -> next AIC init data
SECEND STI R0,*+AR0(48h) ;DTR = current AIC data
        POP R1 ;restore R1
        POP R0 ;restore R0
        POP AR1 ;restore AR1
        POP AR0 ;restore ARO
        RETS ;return from subroutine

AICSET_I ;--- CONFIG FOR INTERRUPT ---
        CALL AICSET ;call AICSET routine
        LDI 0h,IF ;clear IF register
        OR 10h,IE ;enable EXINT0 CPU interrupt
        OR 2000h,ST ;global interrupt enable
        RETS ;return from subroutine

;-----TRANSMIT WAIT ROUTINE-----
TWAIT PUSH ARO ;save ARO
        PUSH R0 ;save R0
        LDI @PBASE,AR0 ;AR0 -> 0808000h
        TW1 LDI *+AR0(40h),R0 ;R0= SP global control reg
        AND 02h,R0 ;see if transmit buffer is ready
        BZ TW1 ;if not ready, try again
        POP R0 ;restore R0
        POP AR0 ;restore ARO
        RETS ;return from subroutine
    
```

เอกสารนี้เป็นเอกสารที่...ไว้สำหรับการใช้งานเพื่อการศึกษา...แต่ให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;-----AIC TRANSFER ROUTINE-----
AICIO_I  LDI      R7,R6          ;copy output to modify for AIC
          LSH      2,R6          ;two LSB =0 for primary AIC comm
IO        PUSH     AR0           ;save AR0
          LDI      @PBASE,AR0    ;AR0 -> 0808000h
          STI      R6,*+AR0(48h) ;DTR = next data for AIC D/A
          LDI      +AR0(4Ch),R6  ;R6 = DRR data from AIC A/D
          LSH      16,R6         ;left shift for sign extension
          ASH      -18,R6        ;right shift keeping sign
          POP      AR0           ;restore AR0
          RETS     ;return from subroutine

;-----AIC POLLING ROUTINE-----
AICIO_P  CALL     TWAIT          ;wait for data to be transferred
          CALL     AICIO_I        ;call AIC transfer routine
          RETS     ;return from subroutine
SW_IO    PUSH     AR0           ;save AR0
          LDI      @PBASE,AR0    ;AR0 -> 0808000h
          LDI      R7,R6          ;copy output to modify for AIC
          LSH      2,R6          ;prepare for secondary AIC com
          OR       03h,R6        ;set two LSB for secondary com
          CALL     TWAIT          ;wait for data to be transferred
          CALL     IO             ;call AIC transfer routine
          CALL     TWAIT          ;wait for data to be transferred
          STI      R1,*+AR0(48h) ;DTR = next data for AIC control
          POP      AR0           ;restore AR0
          RETS     ;return from subroutine

;-----SUBROUTINES FOR PRIMARY OR AUXILIARY INPUT-----
IOPRI    PUSH     R1            ;save R1
          LDI      063h,R1       ;load secondary com data into R1
          CALL     SW_IO         ;call IO routine to switch input
          POP      R1            ;restore R1
          RETS     ;return from subroutine
IOAUX    PUSH     R1            ;save R1
          LDI      073h,R1       ;load secondary com data into R1
          CALL     SW_IO         ;call IO routine to switch input
          POP      R1            ;restore R1
          RETS     ;return from subroutine

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

รายงานฉบับนี้สำเร็จลุล่วงได้ดีด้วยได้รับความช่วยเหลือและชี้แนะจากหลายท่าน ผู้จัดทำขอขอบพระคุณอาจารย์ที่ปรึกษา อาจารย์อัครพล ศรีรัตน์ที่ได้ให้คำปรึกษาตลอดจนชี้แนะแนวทางการแก้ไขปัญหา ขอขอบพระคุณ อาจารย์ศรวดี ชิวปรีชา ซึ่งให้คำปรึกษาและให้ความช่วยเหลือมาโดยตลอด ขอขอบพระคุณ อาจารย์สุรพันธ์ ยิ้มมัน ภาควิชาฟิสิกส์อุปกรณ์การแพทย์ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ ที่ให้ความช่วยเหลือด้านงานฮาร์ดแวร์ คุณไพบุลย์ ผู้ประกาย ที่ช่วยตรวจแก้ไขรายงาน ผู้เขียนพึงระลึกอยู่เสมอว่ารายงานฉบับนี้จะไม่สำเร็จลงได้โดยหากขาดความช่วยเหลือจากทุกท่านจึงขอขอบพระคุณมาอย่างสูง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] J. J. Shynk, "Adaptive IIR Filtering", IEEE ASSP Magazine, April, 1989
- [2] S. Roy, J. J. Shynk, "Analysis of the Momentum LMS Algorithm," IEEE Trans Acoustics Speech and Signal Processing, Vol 38.No.12,December 1990
- [3] M. Radenkovic, T. Bose, "Global Stability of Adaptive IIR Filters Based on the Output," Error Method," Department of Electrical Engineering University of Colorado, Denver
- [4] S. K. Mitra, "Digital Signal Processing, A Computer-Based Approach," MC Graw –Hill, 2002
- [5] J. R. Buck, M. M. Daniel, A. C. Singer, "Computer Explorations in Signals and Systems," Prentice Hall, 1997
- [6] N. Dahnoun, "Digital Signal Processing, Using the TMS320C6000™ DSP Platform," Prentice Hall, 2000
- [7] P. A. Regalia, "Adaptive IIR Filtering in Signal Processing and Control,"
- [8] S. Haykin, "Adaptive Filter Theory," Prentice Hall, 2002
- [9] R. Chassaing, "Digital Signal Processing, Laboratory Experiments Using C and the TMS320C31 DSK," John Wiley & Sons, 1996
- [10] B. Farhang-Boroujeny, "Adaptive Filters Theory and Applications," John Wiley & Sons, 1998
- [11] S. K. Mitra, "Digital Signal Processing, Laboratory Using MATLAB,"
- [12] W. K. Jenkins, A. W. Hull, J. C. Strait, B. A. Schnaufer, X. Li, "Advanced Concepts in Adaptive Signal Processing,"
- [13] M. H. Hayes, "Statistical Digital Signal Processing and Modelling," John Wiley & Sons, 1996
- [14] S. Haykin, B. V. Veen, "Signal and Systems," John Wiley & Sons, 1999
- [15] A. V. Oppenheim, R. W. Schaffer, J. R. Buck, "Discrete-Time Signal Processing," Prentice Hall, 1999

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้