

**สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง**

**การเปรียบเทียบประสิทธิภาพของตระกูล LMS สำหรับวงจรกรองแบบปรับตัวได้**  
**ON COMPARISON OF THE PERFORMANCE OF THE LMS ALGORITHM**  
**FOR ADAPTIVE FILTERING SYSTEM**



โดย  
นาย ธนภัทร ศิวขาว  
นางสาว เบญจมาภรณ์ ไทยนิยงศักดิ์

๘/๖๖-  
๕/๑๕๓๗  
๒๕๕๐

เลขที่.....  
เลขเรื่อง..... 83235  
วัน, เดือน, ปี..... 11 ส.ค. 2551

419 680 23

**ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต**  
**ภาควิชาวิศวกรรมสารสนเทศ**  
**คณะวิศวกรรมศาสตร์**  
**สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**  
**ปีการศึกษา 2550**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ON COMPARISON OF THE PERFORMANCE OF THE LMS ALGORITHM  
FOR ADAPTIVE FILTERING SYSTEM**



**By**  
**Mr.Tanaphat Piewkhao**  
**Ms.Benjamaporn Taiyeunyongsak**

**A THESIS SUBMITTED IN PARTIAL FULLFILLMENT OF  
THE REQUIREMENT FOR THE DEGREE OF  
BACHELOR IN DEPARTMENT OF INFORMATION ENGINEERING  
FACULTY OF ENGINEERING**

**KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG 2007**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**หัวข้อปฏิญานิพนธ์** การเปรียบเทียบประสิทธิภาพของตระกูล LMS สำหรับวงจรกรองแบบปรับตัว  
ได้

**ชื่อนักศึกษา** นายธนภัทร                      ผิวขาว                      รหัสนักศึกษา 47012105

นางสาวเบญจมาภรณ์                      ไทเย็นขงศักดิ์                      รหัสนักศึกษา 47012114

**ชื่ออาจารย์ที่ปรึกษา**                      ดร.จักรี ทิมภาคย์วิศิษฏ์

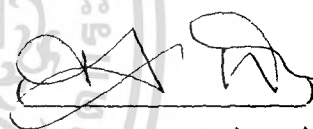
**ระดับการศึกษา**                      ปริญญาตรี วิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมสารสนเทศ

**ภาควิชา**                      วิศวกรรมสารสนเทศ

**ปีการศึกษา**                      2550

ปฏิญานิพนธ์นี้ได้รับความเห็นชอบจากอาจารย์ที่ปรึกษาเป็นที่เรียบร้อยแล้ว



(ดร.จักรี ทิมภาคย์วิศิษฏ์)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**หัวข้อปริญญานิพนธ์** การเปรียบเทียบประสิทธิภาพของตระกูล LMS สำหรับวงจรกรองแบบปรับตัว  
ได้

**ชื่อนักศึกษา** นายชนภัทร ผิวขาว รหัสนักศึกษา 47012105  
นางสาวเบญจมาภรณ์ ไทเย็นขงศักดิ์ รหัสนักศึกษา 47012114

**ชื่ออาจารย์ที่ปรึกษา** ดร.จักรี ทิมภาคย์วิศิษฐ์

**ระดับการศึกษา** ปริญญาตรี วิศวกรรมศาสตรบัณฑิต  
สาขาวิศวกรรมสารสนเทศ

**ภาควิชา** วิศวกรรมสารสนเทศ

**ปีการศึกษา** 2550

#### บทคัดย่อ

วงจรกรองแบบปรับตัวได้มีคุณสมบัติในการปรับค่าสัมประสิทธิ์ได้อย่างอัตโนมัติ ทำให้สามารถประยุกต์ใช้กับอุปกรณ์ได้หลายอย่าง ยกตัวอย่างเช่น การประยุกต์ในการจำลองคุณลักษณะของระบบ, การพัฒนาเป็นอิกวอไลเซอร์แบบปรับตัวได้, การกำจัดสัญญาณรบกวนและการพัฒนาอุปกรณ์เพิ่มประสิทธิภาพสาย สำหรับในโครงการนี้จะนำเสนอการเปรียบเทียบประสิทธิภาพของตระกูล LMS (Least Mean Square) สำหรับวงจรกรองแบบปรับตัวได้ (Adaptive Filtering) นอกจากนี้ได้นำโปรแกรมแมทแลบมาใช้ในการแสดงผลจำลองการทำงานของวงจร

**Thesis Title** ON COMPARISON OF THE PERFORMANCE OF THE LMS  
ALGORITHM FOR ADAPTIVE FILTERING SYSTEM

**Student** Mr. Tanaphat Piewkhao ID. 47012105  
Ms. Benjamaporn Taiyeunyongsak ID. 47012114

**Advisor** Dr. Chakree Teekapakvisit

**Graduate Level** Bachelor Degree of Information Engineering

**Department** Information Engineering

**Academic Year** 2007

## ABSTRACT

Adaptive Filter can change the coefficient automatically, so we can apply to use it with many applications such as System Identification, Adaptive Equalization, Noise Cancellation and Adaptive Line Enhancer. This is concerned about comparison of the performance of the LMS (Least Mean Square) algorithm. MATLAB program is used to simulate to show the simulation results.

## กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยความช่วยเหลืออย่างดียิ่งจากหลาย ๆ ฝ่ายซึ่งผู้จัดทำใคร่ขอขอบคุณทุก ๆ ท่านที่มีส่วนร่วม ช่วยเหลือ และแนะนำในทุก ๆ ด้าน

ขอขอบพระคุณ อาจารย์จักรี ทิมภาคย์วิศิษฎ์ อาจารย์ที่ปรึกษาปริญญาานิพนธ์ที่ได้เสียสละเวลาให้คำปรึกษา และข้อเสนอแนะที่เป็นประโยชน์ จึงทำให้การทำปริญญาานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี

ขอขอบคุณ เพื่อน ๆ นักศึกษาทุกคนที่ช่วยเหลือให้คำแนะนำต่าง ๆ ในการจัดทำ และตรวจสอบปริญญาานิพนธ์จนเสร็จสมบูรณ์

ขอขอบพระคุณ บิดา มารดา ผู้ให้โอกาสและคอยให้กำลังใจเสมอมา คุณค่าและประโยชน์อันพึงมีจากโครงการนี้ ทางผู้เขียนขอบอบแต่ผู้มีพระคุณทุกท่านไว้ ณ โอกาสนี้

นาย ธนภัทร

ผิขาว

นางสาว เบญจมาภรณ์

ไทยินขงศักดิ์

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญรูป	ช
สารบัญตาราง	ญ
บทที่ 1 บทนำ	1
1.1 ความเป็นมาของปริญญานิพนธ์	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์	2
1.3 ขอบเขตปริญญานิพนธ์	2
1.4 ขั้นตอนการดำเนินโครงการ	2
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง	3
2.1 การประมวลสัญญาณ	3
2.1.1 ส่วนประกอบในระบบประมวลผลสัญญาณดิจิทัล	4
2.2 วงจรกรองความถี่แบบดิจิทัล	5
2.2.1 โครงสร้างของวงจรกรองความถี่แบบดิจิทัล	6
2.3 ตัวกรองความถี่ (Filter)	7
2.4 หลักการเลือกใช้ระหว่าง FIR และ IIR	7
2.5 ตัวกรองความถี่แบบ FIR	8
บทที่ 3 การออกแบบโครงการ	10
3.1 อะแดปทีฟฟิลเตอร์ (วงจรกรองความถี่แบบปรับตัวได้)	11
3.2 ทฤษฎีพื้นฐานของอะแดปทีฟฟิลเตอร์	12
3.2.1 โครงสร้างของอะแดปทีฟฟิลเตอร์	12
3.3 กรรมวิธีการพัฒนาอัลกอริทึมของวงจรกรองความถี่แบบปรับตัวได้	14
3.4 การวัดประสิทธิภาพในระบบปรับตัว	14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ(ต่อ)

	หน้า
3.4.1 ความเร็วในการลู่เข้า (Convergence Rate)	15
3.4.2 ค่าความผิดพลาดเฉลี่ยกำลังสอง (Mean Square Error)	15
3.4.3 ความถูกต้องในการประมาณค่าตัวแปร (Parameter Estimation Accuracy)	15
3.4.4 ความซับซ้อนในการประมวลผล (Computation Complexity)	15
3.4.5 เสถียรภาพ (Stability)	15
3.4.6 ความคงทนของระบบ (Robustness)	15
3.5 การประยุกต์ใช้งานวงจรกรองปรับตัวได้	16
3.5.1 ตัวอย่างการนำตัวกรองปรับตัวได้ไปประยุกต์การใช้งาน	16
3.6 การเปรียบเทียบระหว่างฟิลเตอร์ที่เหมาะสมที่สุดกับอะแดปทีฟฟิลเตอร์	19
3.7 เสถียรภาพและภาพสมรรถนะเมื่ออยู่ตัวของอะแดปทีฟฟิลเตอร์	25
3.7.1 การวัดค่าสมรรถนะ (performance measurer)	27
3.8 Least Mean Square Algorithm	29
3.9 LMS – Based Algorithms	30
3.9.1 The LMS – Newton Algorithm	31
3.9.2 The Normalized LMS Algorithm (NLMS)	31
3.9.3 The Affine Projection Algorithm	32
3.9.4 The Partially Filter Gradient LMS Algorithm (PFGLMS)	33
3.10 Simulation Example	34
<b>บทที่ 4 ผลการทดลอง</b>	<b>38</b>
4.1 Conventional LMS Algorithm	41
4.2 The LMS – Newton Algorithm	43
4.3 The Normalized LMS Algorithm (NLMS)	46
4.4 The Partially Filter Gradient LMS Algorithm (PFGLMS)	49
4.5 การเปรียบเทียบความเร็วในการลู่เข้าของ LMS Algorithm ทั้ง 4 แบบ	51
4.6 การเปรียบเทียบความซับซ้อนของระบบ	53

## สารบัญ(ต่อ)

	หน้า
บทที่ 5	54
5.1 สรุปผลการทดลอง	54
5.2 ปัญหาที่เกิดขึ้นในด้านเทคนิค	54
5.3 แนวทางในการพัฒนาต่อและการประยุกต์ใช้งาน	54
บรรณานุกรม	56



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป

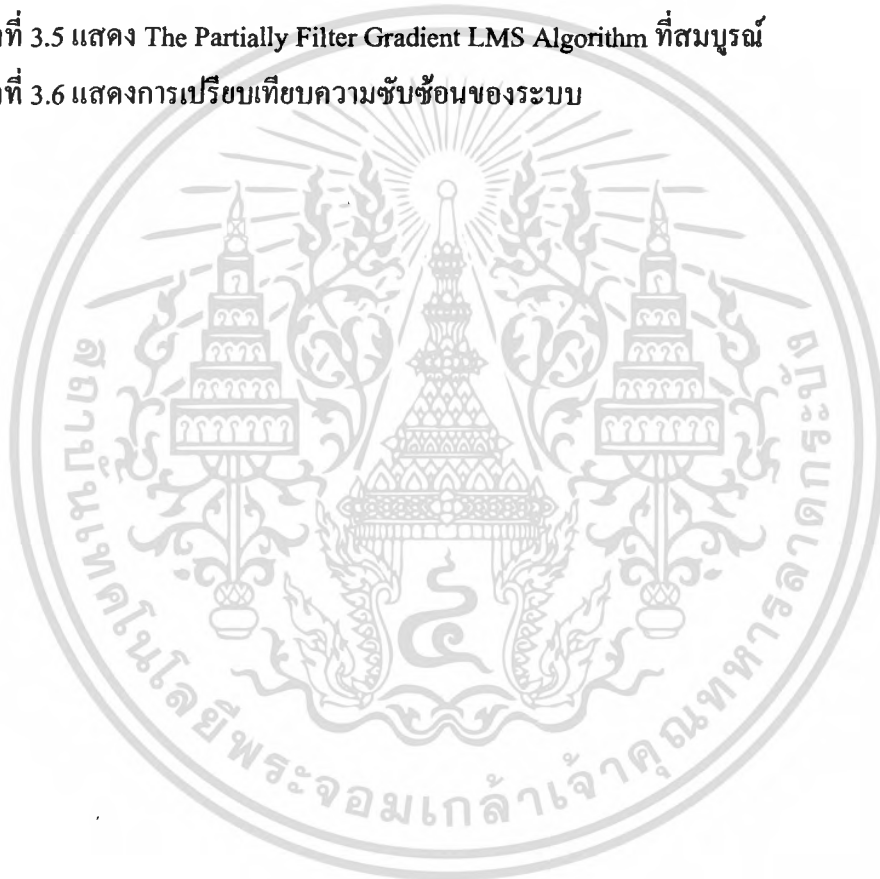
	หน้า
รูปที่ 2.1 ส่วนประกอบในระบบประมวลผลสัญญาณดิจิทัล	4
รูปที่ 2.2 บล็อกไดอะแกรมของวงจรกรองเชิงเลข	6
รูปที่ 2.3 แสดงองค์ประกอบพื้นฐานทั้งสามที่ใช้เป็นส่วนประกอบของตัวกรองดิจิทัล	6
รูปที่ 2.4 โครงสร้างของตัวกรองความถี่แบบ FIR	9
รูปที่ 3.1 ส่วนประกอบพื้นฐานของอะแดปทีฟฟิลเตอร์	12
รูปที่ 3.2 ส่วนประกอบพื้นฐานของ supervised อะแดปทีฟฟิลเตอร์	14
รูปที่ 3.3 แสดงการประยุกต์ใช้ตัวกรองปรับตัวในการจำลองคุณลักษณะของระบบที่ไม่ทราบ	16
รูปที่ 3.4 แสดงการใช้อีควอไลเซอร์แบบปรับตัวได้ในระบบส่งสัญญาณ	17
รูปที่ 3.5 แสดงโครงสร้างระบบกำจัดสัญญาณรบกวน โดยใช้ตัวกรองปรับตัวได้	18
รูปที่ 3.6 แสดงอุปกรณ์เพิ่มประสิทธิภาพสาย	19
รูปที่ 3.7 (ก) ตัวรวมสัญญาณเชิงเส้น ที่มีหลายอินพุต	20
รูปที่ 3.7 (ข) ตัวรวมสัญญาณเชิงเส้น FIR ฟิลเตอร์อินพุตเดี่ยว	20
รูปที่ 3.8 ความแตกต่างของกระบวนการระหว่าง (ก) ฟิลเตอร์ที่เหมาะสมที่สุด และ (ข) อะแดปทีฟฟิลเตอร์	22
รูปที่ 3.9 ไดอะแกรมเวลาสำหรับ priori และ posteriori อะแดปทีฟอัลกอริทึม	25
รูปที่ 3.10 การจำลองการทำงาน ของระบบปรับแต่งช่องสัญญาณ	30
รูปที่ 3.11 ก) สัญญาณเทรนนิง $d(n)$ เป็นสัญญาณดิจิทัล $\pm 1$	35
รูปที่ 3.11 ข) ส่วนขยายของสัญญาณ ที่ สัญญลักษณ์ที่ 488 ถึง 512	36
รูปที่ 3.12 ก) สัญญาณที่รับ ได้ $r(n)$ ที่เครื่องรับ ที่เกิดการผิดเพี้ยน	36
รูปที่ 3.12 ข) ส่วนขยายของสัญญาณ $r(n)$ ที่สัญญลักษณ์ที่ 488 ถึง 512	37
รูปที่ 3.13 ก) สัญญาณ $y(n)$ ที่ผ่านการปรับแต่งด้วยตัวปรับสัญญาณแล้ว	37
รูปที่ 3.13 ข) ส่วนขยายของสัญญาณ ที่ สัญญลักษณ์ที่ 488 ถึง 512	38
รูปที่ 3.14 แสดงผลตอบสนองความถี่รวมของช่องสัญญาณ โดยส่วนอินเวอร์ส	
ช่องสัญญาณจะเป็นผลตอบสนองของตัวปรับแต่งช่องสัญญาณ	38

## สารบัญรูป (ต่อ)

	หน้า
รูปที่ 3.15 ค่าผิดพลาดยกกำลังสอง $e^2(n)$ มีค่าลดลงตามการลู่เข้าของตัวปรับตัวแต่ง สัญญาณและคงตัวเมื่อ จำนวนค่าสัญญาณถึงในระดับ 300 สัญญาณ	39
รูปที่ 4.1 ความเร็วในการลู่เข้าของ Conventional LMS ที่ปรับค่า step size ( $\mu$ ) = 0.01	41
รูปที่ 4.2 ความเร็วในการลู่เข้าของ Conventional LMS ที่ปรับค่า step size ( $\mu$ ) = 0.03	41
รูปที่ 4.3 ความเร็วในการลู่เข้าของ Conventional LMS ที่ปรับค่า step size ( $\mu$ ) = 0.05	42
รูปที่ 4.4 เปรียบเทียบความเร็วในการลู่เข้าของ Conventional LMS	42
รูปที่ 4.5 ความเร็วในการลู่เข้าของ LMS - Newton ที่ปรับค่า step size ( $\mu$ ) = 0.01	43
รูปที่ 4.6 ความเร็วในการลู่เข้าของ LMS - Newton ที่ปรับค่า step size ( $\mu$ ) = 0.03	43
รูปที่ 4.7 ความเร็วในการลู่เข้าของ LMS - Newton ที่ปรับค่า step size ( $\mu$ ) = 0.05	44
รูปที่ 4.8 เปรียบเทียบความเร็วในการลู่เข้าของ LMS - Newton	44
รูปที่ 4.9 เปรียบเทียบความเร็วในการลู่เข้าของ LMS - Newton ที่ $\mu = 0.03$ และปรับค่า alpha ( $\alpha$ ) = 0.7, 0.8, 0.9	45
รูปที่ 4.10 ความเร็วในการลู่เข้าของ NLMS ที่ปรับค่า step size ( $\mu$ ) = 0.01	46
รูปที่ 4.11 ความเร็วในการลู่เข้าของ NLMS ที่ปรับค่า step size ( $\mu$ ) = 0.03	46
รูปที่ 4.12 ความเร็วในการลู่เข้าของ NLMS ที่ปรับค่า step size ( $\mu$ ) = 0.05	47
รูปที่ 4.13 เปรียบเทียบความเร็วในการลู่เข้าของ NLMS	47
รูปที่ 4.14 ความเร็วในการลู่เข้าของ NLMS ที่ปรับค่า step size ( $\mu$ ) = 0.03 และปรับค่า gamma ( $\gamma$ ) = 0.01, 0.5 และ 0.9	48
รูปที่ 4.15 ความเร็วในการลู่เข้าของ PFGLMS ที่ปรับค่า step size ( $\mu$ ) = 0.01	49
รูปที่ 4.16 ความเร็วในการลู่เข้าของ PFGLMS ที่ปรับค่า step size ( $\mu$ ) = 0.03	49
รูปที่ 4.17 ความเร็วในการลู่เข้าของ PFGLMS ที่ปรับค่า step size ( $\mu$ ) = 0.05	50
รูปที่ 4.18 เปรียบเทียบความเร็วในการลู่เข้าของ PFGLMS ที่ $\mu = 0.01, 0.03$ และ $0.05$ Lamda ( $\lambda$ ) = 0.7 และ Gamma ( $\gamma$ ) = 0.9	50
รูปที่ 4.19 เปรียบเทียบความเร็วในการลู่เข้าทั้ง 4 แบบ โดยปรับค่า step size ( $\mu$ ) = 0.01	51
รูปที่ 4.20 เปรียบเทียบความเร็วในการลู่เข้าทั้ง 4 แบบ โดยปรับค่า step size ( $\mu$ ) = 0.03	51
รูปที่ 4.21 เปรียบเทียบความเร็วในการลู่เข้าทั้ง 4 แบบ โดยปรับค่า step size ( $\mu$ ) = 0.05	52

## สารบัญตาราง

	หน้า
ตารางที่ 3.1 แสดงตาราง Conventional LMS Algorithm ที่สมบูรณ์	29
ตารางที่ 3.2 แสดงตาราง LMS – Newton Algorithm ที่สมบูรณ์	30
ตารางที่ 3.3 แสดงตาราง The Normalized LMS Algorithm ที่สมบูรณ์	31
ตารางที่ 3.4 แสดง The Affine Projection Algorithm ที่สมบูรณ์	31
ตารางที่ 3.5 แสดง The Partially Filter Gradient LMS Algorithm ที่สมบูรณ์	35
ตารางที่ 3.6 แสดงการเปรียบเทียบความซับซ้อนของระบบ	53



## บทที่ 1

### บทนำ

#### 1.1 ความเป็นมาของปัญญาประดิษฐ์

ในปัจจุบันเทคโนโลยีทางการสื่อสารและคอมพิวเตอร์ได้ก้าวหน้าไปอย่างรวดเร็วและบทบาทสำคัญของความก้าวหน้าเหล่านั้นก็มาจากการเปลี่ยนแปลงของเทคโนโลยีหลายอย่างที่เคยทำด้วยวงจรอนาล็อกมาเป็นวงจรดิจิทัล เนื่องจากในการประมวลผลสัญญาณดิจิทัลนั้น มีข้อดีเหนือกว่าการใช้วงจรในระบบอนาล็อกมาก เช่น ให้ความถูกต้องแม่นยำกว่า สามารถควบคุมได้ง่าย รวมทั้งในการออกแบบหรือเปลี่ยนแปลงแก้ไขก็สามารถทำได้ง่ายกว่าวงจรอนาล็อก และในปัจจุบันได้มีการนำเอาการประมวลผลสัญญาณดิจิทัลมาประยุกต์ใช้งานกันอย่างแพร่หลายทั้งในด้านการประมวลผลสัญญาณเสียง การประมวลผลสัญญาณแบบหลายมิติ การวิเคราะห์สัญญาณคลื่นสมอง (EEG) การกำจัดสัญญาณรบกวนในระบบการสื่อสาร เป็นต้น

ในระบบการสื่อสารสิ่งที่สำคัญที่สุดก็คือการทำให้การรับส่งข้อมูลระหว่างต้นทางและปลายทางเป็นไปได้อย่างถูกต้อง โดยสัญญาณจากด้านส่งที่จะส่งผ่านไปยังด้านรับนั้นจะต้องเป็นสัญญาณเดียวกัน ซึ่งหมายความว่าในขั้นตอนการส่งสัญญาณนั้นจะต้องไม่มีสัญญาณรบกวนเกิดขึ้นเลย แต่ในทางปฏิบัติพบว่าสัญญาณมักจะถูกรบกวนจนทำให้คุณภาพของสัญญาณที่ไปถึงปลายทางลดลง ดังนั้นการที่จะนำสัญญาณนั้นไปใช้งานจะต้องทำการกรองสัญญาณรบกวนทิ้งก่อน โดยธรรมชาติของสัญญาณรบกวนมักจะมีค่าความถี่และขนาดไม่แน่นอน ดังนั้นในการกรองสัญญาณรบกวนออกจากสัญญาณที่เราต้องการจึงต้องมีการสร้างวงจรกรองความถี่ที่สามารถปรับตัวได้ ซึ่งวงจรกรองความถี่แบบปรับตัวได้นี้ ก็ได้้นำเอาหลักการประมวลผลสัญญาณดิจิทัลมาประยุกต์ใช้งานเช่นกัน เนื่องจากถ้าเราสร้างวงจรความถี่ในระบบอนาล็อกจะต้องทราบถึงคุณสมบัติของระบบว่าต้องการวงจรกรองสัญญาณที่มีคุณสมบัติอย่างไร ตอบสนองความถี่ในย่านใด และต้องมีการคำนวณและนำมาประกอบกันเป็นวงจร หลังจากนั้นจะต้องมีการทดสอบความถูกต้องว่าเป็นไปตามที่ออกแบบไว้หรือไม่ ถ้าพบว่าวงจรมีความผิดพลาดหรือไม่เป็นไปตามที่ต้องการแล้ว ก็จะต้องมีการคำนวณและสร้างใหม่ ซึ่งมีความยุ่งยากและใช้เวลามาก

สำหรับในโครงการนี้จะนำเสนอการเปรียบเทียบประสิทธิภาพของวงจรกรองแบบปรับตัวได้ (Adaptive Filtering) แบบ LMS (Least Mean Square) ซึ่งมีการใช้งานอย่างแพร่หลายในด้านการประมวลผลสัญญาณดิจิทัล โดยพิจารณาถึงความเร็วในการลู่เข้า (Convergence Rate) และความซับซ้อน ในการประมวลผล (Computation Complexity) รวมทั้งหาค่าความผิดพลาดเฉลี่ยกำลังสอง (Mean Square Error) เมื่อใช้เทคนิคในการประมวลผลแตกต่างกัน

## 1.2 วัตถุประสงค์ของปริิณญานิพนธ์

- 1.2.1 เพื่อศึกษาหลักการและประยุกต์การใช้งานของวงจรรองความถี่แบบปรับตัวได้
- 1.2.2 เพื่อศึกษาหลักการของอัลกอริทึมค่าเฉลี่ยกำลังสองน้อยสุด (Least Mean Square, LMS)
- 1.2.3 เพื่อเปรียบเทียบประสิทธิภาพของวงจรรองแบบปรับตัวได้ LMS แบบที่มีโครงสร้างแตกต่างกัน

## 1.3 ขอบเขตปริิณญานิพนธ์

เพื่อศึกษาหลักการและประยุกต์การใช้งานของวงจรรองความถี่แบบปรับตัวได้บนพื้นฐานของอัลกอริทึมค่าเฉลี่ยกำลังสองน้อยสุด (LMS) แบบต่างๆเช่น The LMS-Newton Algorithm, The Normalized LMS Algorithm, The Transform-Domain LMS Algorithm, The Affine Projection Algorithm โดยแสดงผลการเปรียบเทียบความเร็วในการลู่เข้า (Convergence Rate) และความซับซ้อน ในการประมวลผล (Computation Complexity) รวมทั้งหาค่าความผิดพลาดเฉลี่ยกำลังสอง (Mean Square Error) ของวงจรรองแบบปรับตัวได้แบบ LMS ที่มีโครงสร้างดังที่ได้เสนอ

## 1.4 ขั้นตอนการดำเนินโครงการ

- 1.4.1 ศึกษารูปแบบการทำงานโดยรวมและประยุกต์ใช้งานของวงจรรองความถี่แบบปรับตัวได้
- 1.4.2 ศึกษารูปแบบการทำงานและหลักการของอัลกอริทึมค่าเฉลี่ยกำลังสองน้อยสุด
- 1.4.3 แก้ไขและเพิ่มเติมโปรแกรมที่ใช้ในการจำลองระบบ โดยใช้โปรแกรมเมทแลป
- 1.4.4 วิเคราะห์ผลการทดลองที่ได้จากการจำลอง
- 1.4.5 ศึกษารูปแบบการทำงานโดยรวมของแอลเอ็มเอสอัลกอริทึม

## บทที่ 2

### ทฤษฎีและหลักการที่เกี่ยวข้อง

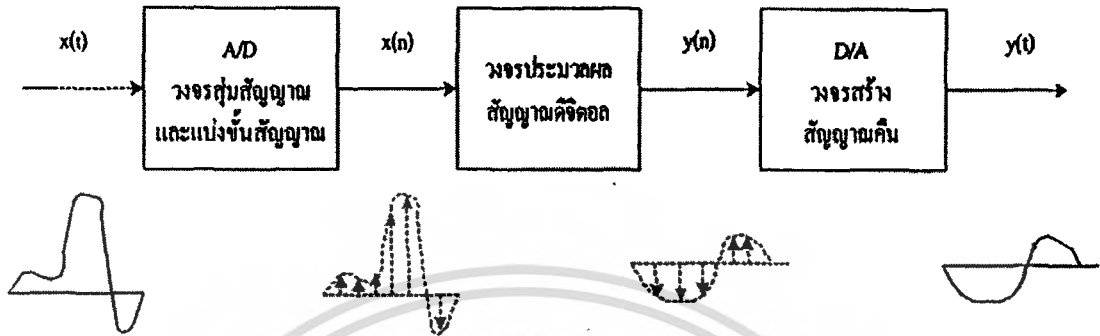
#### 2.1 การประมวลผลสัญญาณ

การประมวลผลสัญญาณ คือ การนำเอาสัญญาณมาเข้าระบบการประมวลผลที่ข้างในประกอบขึ้นด้วย ระบบอุปกรณ์ที่ทำการปฏิบัติการทางคณิตศาสตร์ เช่น การบวกสัญญาณ การคูณ การหาร การถอดรากที่สอง หรือการอินทิเกรตสัญญาณ แต่เดิมนั้นเรานิยมใช้ระบบการประมวลผลสัญญาณในรูปแบบระบบการประมวลผลแบบนี้ก็ คือ ราคาถูก การออกแบบทำได้ง่าย แต่ข้อจำกัดของระบบการประมวลผลสัญญาณเชิงอุปมานก็คือ ประสิทธิภาพและความแม่นยำในการประมวลผล คุณสมบัติของตัวอุปกรณ์ก็ยังแปรค่าตามสภาวะแวดล้อม เช่น อุณหภูมิ อายุการใช้งาน ความชื้น ทำให้ความเชื่อถือของระบบมิได้แค่ในระดับหนึ่งเท่านั้น

ผลจากการพัฒนาทางเทคโนโลยีการออกแบบและผลิตวงจรรวม ทำให้กลอุปกรณ์ดิจิทัล เช่น คอมพิวเตอร์ ไมโครคอมพิวเตอร์ อุปกรณ์ลอจิก เกต หรืออุปกรณ์สนับสนุนต่างๆ มีราคาลดลง และมีประสิทธิภาพดีขึ้นมาก ทำให้ความสนใจในการนำเอาระบบการประมวลผลสัญญาณดิจิทัลมีเพิ่มมากขึ้น ข้อดีของระบบการประมวลผลสัญญาณดิจิทัล คือ ระบบมีประสิทธิภาพและมีความแม่นยำสูง สามารถออกแบบและแก้ไขได้โดยง่าย โดยทั่วไปการศึกษา ออกแบบและพัฒนาระบบการประมวลผลสัญญาณดิจิทัล จะแบ่งออกเป็น 2 หัวข้อใหญ่ๆ คือ

1. วงจรกรองความถี่แบบดิจิทัล (Digital filtering) วงจรกรองความถี่แบบดิจิทัลมีหลายประเภทด้วยกัน แต่มีวงจรกรองอยู่ 2 ชนิด ซึ่งเป็นวงจรกรองเชิงเส้นที่เป็นพื้นฐานของวงจรกรองความถี่แบบดิจิทัลแบบอื่นๆ คือ วงจรกรองความถี่แบบไม่มีการป้อนกลับ (non-recursive filter) และวงจรกรองความถี่แบบป้อนกลับ (recursive filter)
2. การแปลงเชิงเลข (Digital transform) การแปลงเชิงเลขโดยเฉพาะการแปลงฟูริเยอร์เต็มหน่วย (Discrete Fourier Transform หรือ DFT) ที่ทำการประมวลผลโดยใช้ขั้นตอนที่เรียกว่า การแปลงฟาสต์ฟูริเยอร์ (Fast Fourier Transform หรือ FFT) มีส่วนทำให้การประมวลผลสัญญาณเชิงเลขได้รับความนิยมมากขึ้น

### 2.1.1 ส่วนประกอบในระบบประมวลผลสัญญาณดิจิทัล



รูปที่ 2.1 ส่วนประกอบในระบบประมวลผลสัญญาณดิจิทัล

ส่วนประกอบต่างๆ ของระบบประมวลผลสัญญาณดิจิทัล มีดังนี้

1. วงจรแปลงสัญญาณอนาล็อกเป็นดิจิทัล ซึ่งสามารถแบ่งออกได้เป็น 2 กระบวนการย่อย ดังนี้

- วงจรสุ่มสัญญาณ (Sampler) สัญญาณขาเข้าของวงจรนี้เป็นสัญญาณแบบอนาล็อก  $x(t)$  ส่วนสัญญาณขาออกเป็นสัญญาณไม่ต่อเนื่อง  $x(n)$  พารามิเตอร์วงจรสุ่มสัญญาณนี้ก็คือ ค่าอัตราการสุ่ม (Sampling Rate) หรือ ความถี่ในการสุ่ม ใช้สัญลักษณ์แทนว่า  $f_s$  ค่านี้เป็นตัวกำหนดว่า วงจรสุ่มจะสุ่มสัญญาณด้วยอัตราที่ครั้งต่อวินาที หรือกิโลเฮิร์ต (Hz)

- วงจรแบ่งขั้นสัญญาณ (Quantizer) สัญญาณ  $x(n)$  ที่ได้จากวงจรสุ่มสัญญาณถือว่ามีความละเอียดเต็มที่ในทางขนาด ซึ่งในทางปฏิบัติเมื่อนำไปใช้งานจะต้องลดความละเอียดของ  $x(n)$  ลง ให้สามารถแทนได้ด้วยสัญญาณดิจิทัลที่มีบิตจำกัด กระบวนการลดความละเอียดนี้เรียกว่า การแบ่งขั้นของสัญญาณ (Quantization) ความละเอียดที่ได้จากการแบ่งขั้นสัญญาณขึ้นกับจำนวนบิตที่ใช้

วงจรสุ่มสัญญาณรวมกับวงจรแบ่งขั้นสัญญาณ ในทางปฏิบัติก็คือ ตัวแปลงสัญญาณอนาล็อกเป็นดิจิทัล (A/D converter) นั่นเอง ซึ่งจะรวมสองกระบวนการนี้อยู่ในวงจรเดียวกันและโดยทั่วไปเราจะใช้ตัวแปลงสัญญาณอนาล็อกเป็นดิจิทัลในรูปของวงจรรวมสำเร็จรูป (IC)

2. วงจรประมวลผลสัญญาณ ส่วนนี้เป็นหัวใจหลัก ซึ่งจะทำหน้าที่ประมวลผลสัญญาณ  $x(n)$  เพื่อกระทำผลบางอย่างกับสัญญาณ เช่น เป็นวงจรกรองความถี่บางย่านออก และให้ผลลัพธ์ของการประมวลผลเป็นสัญญาณขาออก  $y(n)$  วงจรประมวลผลสัญญาณนี้ ถ้าจะพิจารณากันอย่างง่ายๆ ก็คือตัวคำนวณนั่นเองกล่าวได้ว่า มันกระทำการคำนวณหาสัญญาณขาออกจากสัญญาณขาเข้า โดยมองเห็นสัญญาณขาเข้าในลักษณะลำดับของค่า

3. วงจรสร้างสัญญาณคืน (Signal reconstruction) ใช้ในระบบที่สัญญาณขาออกสุดท้ายเป็นสัญญาณต่อเนื่อง (การประมวลผลสัญญาณบางอย่าง ต้องการสัญญาณขาออกเป็นไม่ต่อเนื่อง ก็ไม่จำเป็นต้องมีส่วนที่ 3) โดยทำหน้าที่แปลงสัญญาณไม่ต่อเนื่อง  $y(n)$  ให้กลับสัญญาณต่อเนื่อง  $y(t)$  ซึ่งจะเป็นสัญญาณขาออกสุดท้ายของระบบ วงจรประเภทนี้ก็คือ ตัวแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อก (D/A converter) นั่นเอง ซึ่งก็มีอยู่ในรูปแบบของวงจรสำเร็จรูปเช่นกัน

## 2.2 วงจรกรองความถี่แบบดิจิทัล

วงจรกรองความถี่แบบดิจิทัล คือ กระบวนการที่ตัดแปลงสเปกตรัมของสัญญาณ ให้มีสเปกตรัมเป็นไปตามข้อกำหนดที่ต้องการ ซึ่งอาจเป็นการเพิ่มค่าหรือลดทอนค่าขนาดของสัญญาณในแถบความถี่ที่กำหนดให้ หรือทำการตัดสัญญาณที่ไม่ต้องการในช่วงความถี่ที่กำหนดออกไป ซึ่งทำให้สามารถแยกสัญญาณในช่วงความถี่ที่ต้องการได้ ในการวิเคราะห์และสังเคราะห์วงจรนั้น ต้องใช้เครื่องมือพื้นฐานทางคณิตศาสตร์เข้าช่วย ดังนั้นเราจึงนิยมเรียกว่า วงจรกรองความถี่เชิงเลข

การที่วงจรกรองความถี่เชิงเลขถูกนำมาประยุกต์ใช้งานกันอย่างกว้างขวางนั้น อาจมาจากข้อได้เปรียบหลายประการดังนี้

1. ผลตอบสนองความถี่ของวงจรกรองความถี่ สามารถออกแบบให้มีความใกล้เคียงกับผลตอบสนองความถี่ที่กำหนดให้ หรือผลตอบสนองความถี่ที่ต้องการได้ นอกจากนี้การออกแบบวงจรกรองความถี่ให้มียกผลตอบสนองเชิงเส้นทำได้ง่าย
2. คุณสมบัติของวงจรกรองความถี่ที่ออกแบบและสร้างแล้วจะไม่ขยับเลื่อนไปตามสภาพแวดล้อม หรือตามอุณหภูมิ หรือระยะเวลาในการใช้งาน นอกจากนี้ยังสามารถใช้งานในย่านความถี่ต่ำได้เป็นอย่างดี
3. การประยุกต์ใช้งานเป็นวงจรกรองความถี่แบบปรับความถี่ได้ (Adaptive Filter) ทำได้ง่าย
4. ผู้ออกแบบสามารถออกแบบโดยคำนึงถึงความยาวของคำ (word length) ของตัวเลขฐานสองที่ต้องการใช้ และยังสามารถออกแบบให้มียกผลตอบสนองความถี่ที่ต้องการได้
5. ในปัจจุบัน ถ้าพิจารณาในแง่ของเสถียรภาพของวงจรกรองความถี่ ความเชื่อถือได้ ราคา หรือขนาดของวงจรกรองความถี่เชิงเลข สิ่งเหล่านี้กำลังได้รับการพัฒนา และปรับปรุง ซึ่งมีแนวโน้มว่าจะให้ผลลัพธ์ที่ดีกว่าของวงจรกรองความถี่แบบอนาล็อก (Analog Filter) หรือเรียกว่า วงจรกรองความถี่แบบอุปมาน

วงจรกรองความถี่เชิงเลขสามารถเขียนอธิบายในรูปของบล็อกไดอะแกรมได้ดังรูปที่ 2.2

โดยมีสัญญาณอินพุตซึ่งเป็นสัญญาณอนาล็อกจะถูกสุ่ม (Sampled) ด้วยช่วงเวลาทีคงที่ค่าหนึ่งและ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

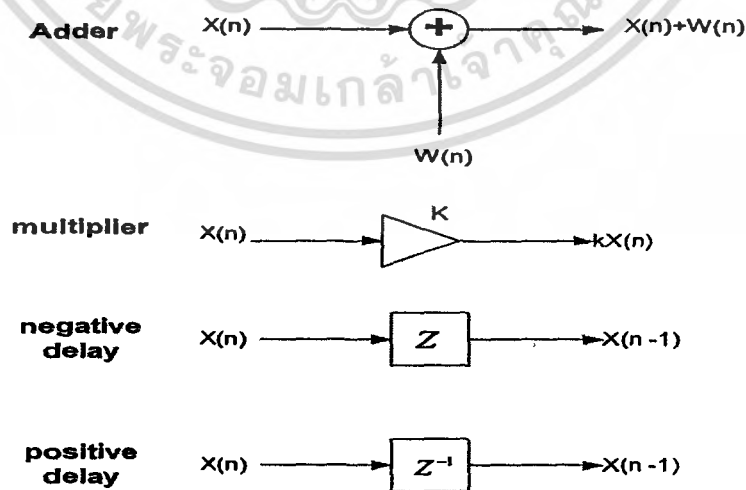
สัญญาณที่ถูกสุ่มนี้จะถูกเปลี่ยนให้อยู่ในรูปเลขฐานสอง โดยการแปลงสัญญาณอนาล็อกให้เป็นสัญญาณดิจิทัลหรือสัญญาณเชิงเลข (analog to digital converter) หลังจากนั้นเลขฐานสองที่แทนสัญญาณอนาล็อกที่เข้ามาทางอินพุตจะถูกกรองโดยวงจรกรองความถี่เชิงเลข การกรองจะเป็นการคำนวณทางตัวเลข ซึ่งจะอาศัยวงจรที่ใช้ในระบบคอมพิวเตอร์ ได้แก่ ตัวบวก ตัวคูณ รีจิสเตอร์ และ อุปกรณ์หน่วยความจำต่างๆ ต่อมาค่าเอาต์พุตที่ได้จากวงจรกรองความถี่เชิงเลขนี้จะถูกตัดแปลงกลับเป็นสัญญาณอนาล็อกอีกทีหนึ่งเป็นสัญญาณเอาต์พุตที่จะนำไปใช้งานได้



รูปที่ 2.2 บล็อกไดอะแกรมของวงจรกรองเชิงเลข

### 2.2.1 โครงสร้างของวงจรกรองความถี่แบบดิจิทัล

วงจรกรองความถี่ดิจิทัล ประกอบไปด้วยส่วนสำคัญ 3 ส่วนคือ การบวก (Adder) การคูณ (Multiplier) และการหน่วง (Unit Delay) เวลาแสดงในรูปที่ 2.3 การบวกและการคูณจะใช้แนวความคิดมาจากตัวเลขในหน่วยคอมพิวเตอร์ ส่วนการหน่วงจะทำให้การถึงข้อมูลในอนาคตมีค่าอย่างต่อเนื่อง



รูปที่ 2.3 แสดงองค์ประกอบพื้นฐานทั้งสามที่ใช้เป็นส่วนประกอบของตัวกรองดิจิทัล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหน่วงเวลานั้นแบ่งเป็น 2 ส่วน คือ บวก (positive) และลบ (negative) โดยการหน่วงแบบบวกนี้เป็นอุปกรณ์ที่ทำหน้าที่บันทึกความจำของรีจิสเตอร์ (register) จะเก็บค่าได้ตามระยะเวลาที่กำหนดสำหรับการคำนวณครั้งต่อไป การหน่วงแบบบวกจะแทนค่าด้วย  $Z^{-1}$  และสามารถอธิบายความสัมพันธ์ได้ด้วยการแปลงแซด การหน่วงแบบลบใช้แทนค่าคือไปในระดับสัญญาณ แทนค่าด้วย  $Z$  จะมีชนิดและการใช้งานที่เหมาะสม อย่างไรก็ตามการไม่สามารถใช้งานได้เสมอไป

## 2.3 ตัวกรองความถี่หรือฟิลเตอร์

การกรองความถี่จะมีวงจรกรองความถี่ที่สำคัญคือ วงจรกรองแบบผ่านต่ำ (Low-pass) วงจรกรองแบบผ่านสูง (High-pass) วงจรกรองแบบผ่านแถบ (Band-pass) วงจรกรองแบบแถบหยุด (Band-stop) และวงจรกรองแบบผ่านทุกความถี่ ซึ่งตัวกรองความถี่แบบดิจิทัลสามารถแบ่งได้สองแบบ คือ ตัวกรองความถี่แบบ Finite Impulse Respond (FIR filter or Non-recursive digital filter) และตัวกรองความถี่แบบ Infinite Impulse Respond (IIR filter or Recursive digital filter) ตัวกรองความถี่แบบ FIR นั้นจะมีฟังก์ชันถ่ายโอนหรือทรานส์เฟอร์ฟังก์ชัน (Transfer function) ที่ประกอบด้วยซีโร่ (Zero) ไม่มีโพล (Pole) ทำให้ระบบมีความเสถียรภาพอย่างแน่นอนและมีผลตอบสนองทางเฟสเป็นเชิงเส้น (Linear phase) แต่มีข้อเสียคือต้องมีจำนวนอันดับ (Order) สูงจึงทำให้ตัวกรองทำงานได้ดี ตัวกรองความถี่แบบ IIR นั้นฟังก์ชันถ่ายโอนจะประกอบไปด้วยทั้งซีโร่และโพล ซึ่งจะทำให้เกิดปัญหาทางด้านความเสถียรภาพ แต่ถ้ามีการออกแบบที่ดีจะทำให้ปัญหาดังกล่าวน้อยลงหรือไม่เกิดขึ้นเลย ข้อดีของตัวกรองความถี่แบบ IIR เมื่อเทียบกับตัวกรองความถี่แบบ FIR คือที่สมรรถนะของการทำงาน (Performance) เท่ากัน ตัวกรองความถี่แบบ IIR จะใช้จำนวนอันดับที่น้อยกว่า ซึ่งทำให้การคำนวณน้อยกว่าด้วย ซึ่งความซับซ้อนในการคำนวณนี้ มีความสำคัญมากเมื่อนำตัวกรองความถี่ไปใช้งานที่เวลาจริง (Real time) ด้วยไมโครโปรเซสเซอร์ ซึ่งมีข้อจำกัดทางด้านจำนวนบิตและความเร็วในการทำงาน ตัวกรองความถี่ที่มีจำนวนการคำนวณน้อยกว่าจะทำให้ไมโครโปรเซสเซอร์ทำงานน้อยลง และทำงานได้เร็วกว่า

## 2.4 หลักการเลือกใช้ระหว่าง FIR และ IIR

1. การประยุกต์ใช้งานทางการสื่อสารข้อมูล (data transmission) ทางกายภาพ และการประมวลผลสัญญาณภาพ จำเป็นต้องใช้ตัวกรองความถี่ที่ให้ผลตอบสนองทางเฟสเป็นเชิงเส้น ซึ่ง ตัวกรองความถี่แบบ FIR จะเหมาะสมกว่าตัวกรองความถี่แบบ IIR

2. สำหรับการใช้งานที่เวลาจริง ซึ่งต้องมีการจำกัดจำนวนบิตนั้น การคำนวณสัมประสิทธิ์ของตัวกรองจะทำให้การปัดเศษ (Round - Off) ซึ่งทำให้เกิด Round -Off Noise ขึ้น ซึ่ง Round -Off Noise นี้ จะเกิดขึ้นในตัวกรองความถี่แบบ FIR น้อยกว่าที่เกิดขึ้นในตัวกรองความถี่แบบ IIR
3. ในกรณีที่มีความต้องการความชันในช่วงแถบหยุด (Cutoff) สูงนั้น ตัวกรองความถี่แบบ FIR จะต้องใช้จำนวนสัมประสิทธิ์มากกว่าตัวกรองความถี่แบบ IIR
4. ตัวกรองความถี่แบบ IIR สามารถออกแบบได้โดยตรงจากตัวกรองต้นแบบในตัวกรองความถี่แบบอนาลอกแต่ตัวกรองความถี่แบบ FIR ทำไม่ได้
5. ตัวกรองความถี่แบบ FIR จะสังเคราะห์ได้ค่อนข้างยากถ้าหากไม่ใช่คอมพิวเตอร์ (CAD) ช่วยในการออกแบบ
6. จะใช้ตัวกรองความถี่แบบ IIR ก็ต่อเมื่อในงานที่มีความต้องการความชันในช่วงแถบหยุดสูง ซึ่งตัวกรองความถี่แบบ IIR จะใช้จำนวนสัมประสิทธิ์น้อยกว่า
7. จะใช้ตัวกรองความถี่แบบ FIR ถ้าจำนวนสัมประสิทธิ์ไม่มากจนเกินไป โดยเฉพาะสำหรับงานที่ต้องการความผิดเพี้ยนทางเฟสน้อยจะเหมาะสมอย่างยิ่ง

## 2.5 ตัวกรองความถี่แบบ FIR (Finite impulse response)

ตัวกรองความถี่แบบ FIR จะเป็นตัวกรองความถี่ที่มีการรวมเอาเฉพาะอย่างค่าของอินพุตที่เวลาปัจจุบันและอดีตมาประมวลผลเพื่อให้ได้เอาต์พุตตามที่ต้องการ ซึ่งเราจะสามารถเขียนสมการของตัวกรองความถี่แบบ FIR ได้ดังต่อไปนี้

$$y(n) = \sum_{k=0}^{N-1} h(k) \cdot x(n-k) \quad (2-1)$$

ฟังก์ชันถ่ายโอนของตัวกรองความถี่แบบ FIR คือ

$$H(z) = \sum_{k=0}^{N-1} h(k) \cdot z^{-k} \quad (2-2)$$

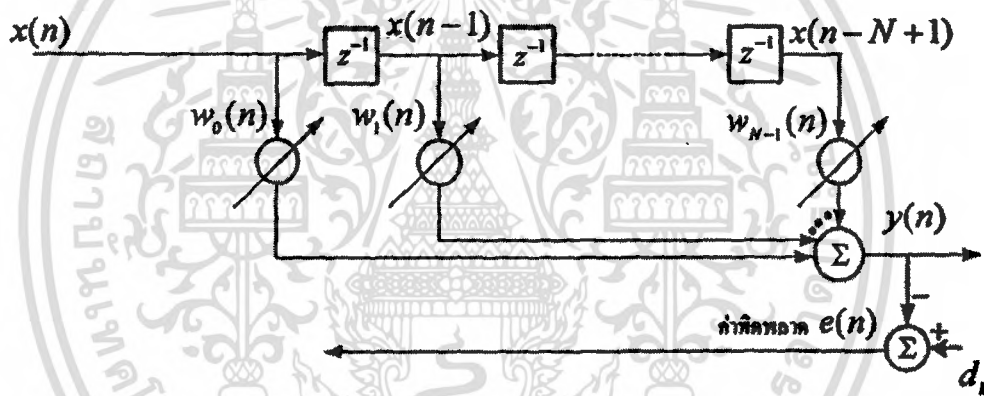
โดยที่  $h(k)$ ,  $k = 0, 1, 2, \dots, N-1$  คือ ค่าสัมประสิทธิ์ผลตอบสนองต่ออิมพัลส์ของตัวกรองความถี่  $N$  คือ จำนวนข้อมูล

ตัวกรองความถี่แบบ FIR เป็นฟิลเตอร์ที่ให้คุณสมบัติที่สำคัญคือความคงที่สัญญาณเอาต์พุต

เนื่องจากมันจะให้เฉพาะซีโรเท่านั้นและให้ความเป็นคุณลักษณะของเฟสเป็นเชิงเส้น จะป้องกันการเอกสการนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผิเคเพี้ยนของสัญญาณเอาต์พุต จำนวนของสัมประสิทธิ์ของสมการ (2-1) เป็นสมการเชิงอนุพันธ์ของ FIR ในโดเมนเวลา โดยพิจารณาเพียงค่าของอินพุต  $x(n)$  ในค่าอดีตและปัจจุบันเท่านั้นฟังก์ชันถ่ายโอนของฟิลเตอร์จะใช้ในการวิเคราะห์และหาค่าผลตอบสนองความถี่ของฟิลเตอร์

โดยทั่วไปการออกแบบของตัวกรองความถี่แบบ FIR จะพยายามลดจำนวนของค่าสัมประสิทธิ์ให้มีจำนวนน้อยที่สุดในขณะที่ยังคงรักษาคุณสมบัติที่ต้องการไว้อยู่ ในทางปฏิบัติจะมีการใช้จำนวนของค่าสัมประสิทธิ์ประมาณ 11-150 ตัว คุณสมบัติที่สำคัญคือ ความคงที่ของสัญญาณเอาต์พุตเนื่องจากตัวกรองความถี่แบบ FIR ซึ่งยากที่จะพบในตัวกรองความถี่แบบ IIR การที่ตัวกรองความถี่แบบ FIR ให้คุณลักษณะของเฟสเชิงเส้นนั้นจะมีผลให้สัญญาณที่ผ่านมาช่วงความถี่ที่ต้องการในแต่ละแบบของตัวกรองความถี่ไม่เปลี่ยนแปลงรูปร่างไปจากเดิม



รูปที่ 2.4 โครงสร้างของตัวกรองความถี่แบบ FIR

### บทที่ 3

#### การออกแบบโครงงาน

ในแควงของการออกแบบวงจรอนาล็อก คำว่าวงจรกรองอนาล็อก หรือ (Analogue filter) นั้น คือ วงจรอิเล็กทรอนิกส์ที่ให้ผลตอบสนองความถี่ (frequency response) อาจจะเป็นทางขนาดหรือเฟสที่ไม่เท่ากันในแต่ละย่านความถี่ เพื่อหวังผลในการตัดแปลง หรือ “แยก” ย่านความถี่ หรือความถี่เฉพาะ ที่ต้องการออกจากส่วนที่ไม่ต้องการ ตัวอย่างเช่นวงจรกรองต่ำผ่าน (low pass filter) นั้น เป็นวงจรกรองที่ให้สัญญาณขาเข้าเฉพาะความถี่ตั้งแต่ ศูนย์ เฮิรตซ์ (Hertz) ขึ้นไป จนถึงความถี่ที่ค่าหนึ่งซึ่งเรียกว่า ความถี่คัทออฟ (cut-off frequency) ผ่านไปได้ และ ทำหน้าที่ลดทอนสัญญาณที่มีความถี่เกินกว่านั้น ส่วนวงจรสูงผ่าน (high pass filter) จะให้ผลที่กลับกัน คือ ขอมให้สัญญาณที่มีความถี่ที่สูงกว่าความถี่คัทออฟให้ผ่านไปได้ และลดทอนสัญญาณที่มีความถี่ต่ำไม่ให้ผ่านไปได้

ตัวปรับแต่งช่องสัญญาณนั้นสร้างมาได้จากการออกแบบอัลกอริทึม (Algorithm) ทางด้านการประมวลผลสัญญาณ ซึ่งเป็นกรรมวิธีทางคณิตศาสตร์ เพื่อสร้างเป็นวงจรดิจิทัลฟิลเตอร์ (Digital filter) โดยหน้าที่หลักของวงจรปรับแต่งช่องสัญญาณ ก็คือการสร้างผลตอบสนองที่ตรงกันข้ามกับผลตอบสนองของช่องทางการสื่อสาร ซึ่งในทางด้านการประมวลผลสัญญาณจะเรียกว่าการทำ อินเวอร์สแชนเนล (inverse channel) การ Implementation ของอัลกอริทึมหนึ่งๆ ลงชีพ DSP นั้น ก็เป็นการใช้โปรแกรมทำงานเพื่อให้ผลเดียวกับการสร้างวงจรอิเล็กทรอนิกส์

การกำหนดเรียกอัลกอริทึมที่ทำหน้าที่เป็นตัวปรับแต่งช่องสัญญาณว่า filter หรือวงจรกรองนั้น มาจากไอเดียของทฤษฎีการออกแบบการประมวลผลสัญญาณ Norbert Wiener (1894-1964) ซึ่งศึกษาการสร้างอัลกอริทึมด้วยวงจรเชิงเส้น เขาพบว่า วงจรที่สร้างจาก อัลกอริทึมที่เขาคิดค้นนั้น ทำหน้าที่ในการแยก สัญญาณที่ต้องการออกจากสัญญาณรบกวน ซึ่งโดยทั่วไปจะเป็นสัญญาณรบกวนสีขาว (White noise) ดังนั้นรูปแบบของการทำงานของอัลกอริทึมนี้ จะคล้ายคลึง กับ วงจรกรองอนาล็อกที่ทำหน้าที่กรองสัญญาณที่มีความถี่ที่ต้องการออกจากสัญญาณความถี่อื่นๆ ดังนั้น เพื่อให้เกิดความรู้สึกในเชิงเทียบเคียงกันของ คำว่า วงจรกรอง และ อัลกอริทึม Wiener จึงเรียกอัลกอริทึมของเขว่าเป็น “วงจรกรอง” ซึ่งต่อมาภายหลังก็ได้ มีการกำหนดชื่ออัลกอริทึมของ Wiener ซึ่งเป็นอัลกอริทึมที่เหมาะสมที่สุด (Optimum filter) ว่าเป็น วงจรกรองวินเนอร์ (Wiener filter) เพื่อเป็นเกียรติและ การเรียกอัลกอริทึมว่าเป็นวงจรกรอง ก็เป็นที่ใช้กันเรื่อยมาตราบทุกวันนี้ ช่องสัญญาณสัญญาณสำหรับระบบสื่อสารไร้สายนั้น มีลักษณะที่แปรเปลี่ยนไปตามเวลาได้ ดังนั้นการกำหนดค่าคุณสมบัติของตัวปรับแต่งช่องสัญญาณแบบที่ดีที่สุดเหมาะสมที่สุด สำหรับช่องสัญญาณหนึ่ง ณ เวลาหนึ่ง ก็อาจจะไม่ดีไม่เหมาะสม กับช่องสัญญาณนั้น ณ เวลาอื่นๆ ก็เป็น ได้

ดังนั้นการใช้วงจรกรองที่มีการปรับตัวได้ จึงมีความเหมาะสมอย่างยิ่งที่จะนำมาประยุกต์ใช้เป็นตัวปรับแต่งช่องสัญญาณ โดยเมื่อระยะเวลาปรับตัวสิ้นสุดลง หรือที่เรียกว่า “ลู่เข้าหาคำตอบ (Convergence)” เรียบร้อยแล้ว วงจรกรองนั้นก็จะมีความสัมพันธ์ที่ดีที่สุดเหมาะสมที่สุด กับทุกๆ เวลา และ ทุกๆ ช่องสัญญาณ และนั่นหมายถึง การลด ISI ลงสู่ระดับที่ให้ความผิดพลาดที่ต่ำ

ในบทนี้จะอธิบายถึงโครงสร้างของอะแดปทีฟฟิลเตอร์ และการทำงานของอะแดปทีฟฟิลเตอร์โดยละเอียด รวมทั้งแสดงตัวอย่างการซิมูเลชัน โดยใช้โปรแกรมแมทแล็บเพื่อหาความเร็วในการลู่เข้ารวมไปถึงคำนวณค่าความผิดพลาดเฉลี่ยยกกำลังสองน้อยสุด

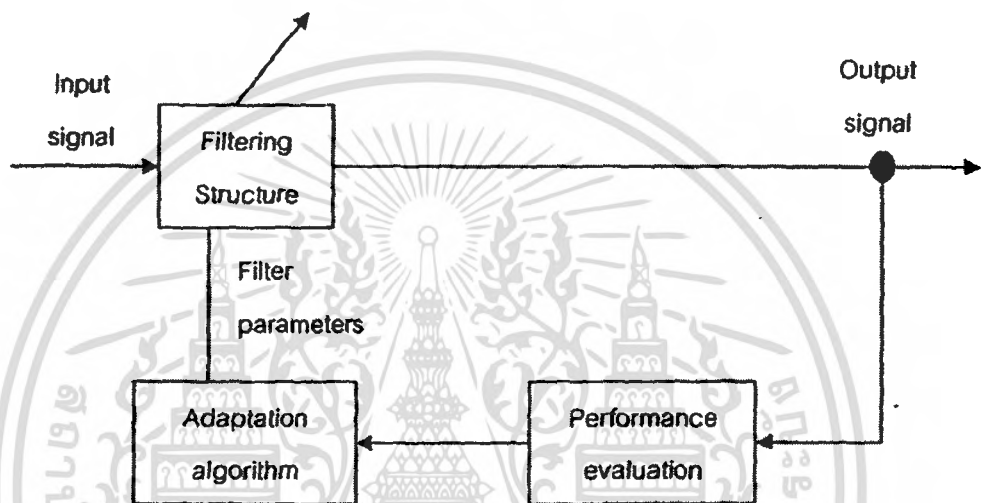
### 3.1 วงจรกรองความถี่แบบปรับตัวได้ (Adaptive Filter)

วงจรกรองความถี่ต่างๆ ไป เป็นวงจรกรองความถี่แบบสัมประสิทธิ์คงที่ หรือแบบไม่แปรตามเวลา สำหรับวงจรกรองความถี่แบบปรับตัวได้จะเป็นวงจรกรองความถี่แบบที่มีสัมประสิทธิ์ปรับเปลี่ยนตลอดเวลา โดยมีลักษณะการปรับค่าสัมประสิทธิ์โดยอัตโนมัติ โดยใช้เงื่อนไขทางสถิติของสัญญาณและของ โมเดลของสิ่งแวดลอมที่สร้างขึ้นในการหาค่าสัมประสิทธิ์ ตัวกรองแบบปรับตัวได้นี้ทำให้ขอบเขตของการประยุกต์ใช้การประมวลผลสัญญาณดิจิทัลขยายวงออกไปกว้างมาก ตัวกรองความถี่แบบปรับตัวได้นี้จะมีประโยชน์อย่างมากในสถานะที่เราไม่สามารถใช้ตัวกรองความถี่ดิจิทัลแบบธรรมดาได้หรือใช้ได้แต่ให้ผลไม่ค้ำนัก ซึ่งจะให้ผลที่ขึ้นถ้าใช้ตัวกรองความถี่แบบปรับตัวได้ ยกตัวอย่างเช่น

1. เมื่อสัญญาณรบกวนและสัญญาณที่ต้องการอยู่ในย่านความถี่เดียวกัน ซึ่งถ้าเราใช้ตัวกรองความถี่ที่มีค่าสัมประสิทธิ์คงที่ และต้องการให้สัญญาณรบกวนส่วนใหญ่หมดไป ก็จะทำให้สัญญาณที่ต้องการหายไปด้วย
2. เมื่อไม่รู้ลักษณะที่แน่นอนของสัญญาณรบกวน เช่น ไม่รู้ว่าสัญญาณรบกวนอยู่ในย่านความถี่ใดหรือเข้ามาในทิศทางใดในกรณีของสายอากาศแบบอะเรย์ เป็นต้น
3. เมื่อสัญญาณที่ต้องการมีความถี่เปลี่ยน ไม่รู้ลักษณะที่แน่นอน เช่น กรณีของโมเด็ม หรือโทรศัพท์มือถือ ที่มีการส่งสัญญาณผ่านช่องสัญญาณ สัญญาณที่ตัวรับได้รับจะถูกทำให้ผิดเพี้ยนไปด้วยฟังก์ชันถ่ายโอนของสัญญาณที่ไม่รู้ค่าแน่นอนและอาจมีค่าแปรตามเวลาด้วย เพราะฉะนั้นจึงต้องใช้ตัวกรองเพื่อทำการกรองเอาความผิดพลาดนี้ออกไป

### 3.2 ทฤษฎีพื้นฐานของอะแดปทีฟฟิลเตอร์

ในหัวข้อนี้จะกล่าวถึงโครงสร้างทางคณิตศาสตร์สำหรับใช้ในการวิเคราะห์และหาค่าสมรรถนะของอะแดปทีฟอัลกอริทึม จุดมุ่งหมายก็คือ เพื่อเป็นจุดเริ่มต้นในการพัฒนาการออกแบบ โดยประยุกต์ใช้อะแดปทีฟอัลกอริทึมเพื่อนำไปแก้ปัญหาในทางปฏิบัติ



รูปที่ 3.1 ส่วนประกอบพื้นฐานของอะแดปทีฟฟิลเตอร์

#### 3.2.1 โครงสร้างของอะแดปทีฟฟิลเตอร์

อะแดปทีฟฟิลเตอร์นั้นจะมีการเปลี่ยนแปลงค่าของพารามิเตอร์ให้เหมาะสมกับผลตอบสนองที่ต้องการในเวลาเหมาะสม ในการประยุกต์ใช้อะแดปทีฟฟิลเตอร์ โดยทั่วไปอาจจะมีสัญญาณหนึ่งอินพุตหรือมากกว่า และผลตอบสนองของสัญญาณที่ต้องการ ซึ่งอาจผ่านหรือไม่ผ่านตัวอะแดปทีฟฟิลเตอร์ก็ได้ อะแดปทีฟฟิลเตอร์โดยทั่วไปจะประกอบไปด้วย 3 ส่วนหลักๆ ดังรูปที่ 3.1 ดังนี้

- โครงสร้างของฟิลเตอร์ โดยส่วนของฟิลเตอร์นี้จะใช้วัดสัญญาณอินพุตหนึ่งสัญญาณหรือมากกว่านั้น โครงสร้างของฟิลเตอร์จะเป็นแบบเชิงเส้น ค่าที่สัญญาณเอาต์พุตมีลักษณะการรวมกันอย่างเป็นเชิงเส้นของสัญญาณอินพุตที่วัดได้ ในทางกลับกัน โครงสร้างของฟิลเตอร์จะเป็นแบบไม่เชิงเส้น ยกตัวอย่างเช่น ตัวฟิลเตอร์สามารถปรับผลตอบสนองของดิจิทัลฟิลเตอร์แบบ FIR ที่สร้าง

ด้วยโครงสร้างแบบตรง หรือแลตทิส หรือโครงสร้างแบบคาสเคด โครงสร้างต่างๆเหล่านี้จะไม่สามารถเปลี่ยนได้แต่พารามิเตอร์จะถูกรับเปลี่ยนไปโดยอะแดปทีฟอัลกอริทึม

- เกณฑ์การหาสมรรถนะ สัญญาณเอาต์พุตของอะแดปทีฟฟิลเตอร์และผลที่ตอบสนองที่ความต้องการนั้นจะผ่านกระบวนการของเกณฑ์การหาสมรรถนะ เพื่อให้ได้คุณภาพตามต้องการหรือขึ้นอยู่กับความต้องการของการประยุกต์ใช้งานเฉพาะด้าน ส่วนมากแล้วอะแดปทีฟฟิลเตอร์ จะพิจารณาที่ค่าเฉลี่ยกำลังสองเพราะเป็นคณิตศาสตร์ที่ไม่ยากนักและสามารถนำไปใช้กับระบบที่ออกแบบไว้ใช้ในทางปฏิบัติได้

- อะแดปทีฟอัลกอริทึมนั้นจะใช้ค่าของเกณฑ์การหาสมรรถนะ หรือบางฟังก์ชันของเกณฑ์การหาสมรรถนะ และทำการวัดค่าอินพุตและผลตอบสนองที่ความต้องการ เพื่อนำไปทำการตัดสินใจว่าจะทำการปรับปรุณค่าพารามิเตอร์ของฟิลเตอร์อย่างไร เพื่อนำไปสู่สมรรถนะที่ดีขึ้น ความซับซ้อนและคุณลักษณะของอะแดปทีฟอัลกอริทึมขึ้นอยู่กับฟังก์ชันของโครงสร้างของฟิลเตอร์และเกณฑ์การหาสมรรถนะ

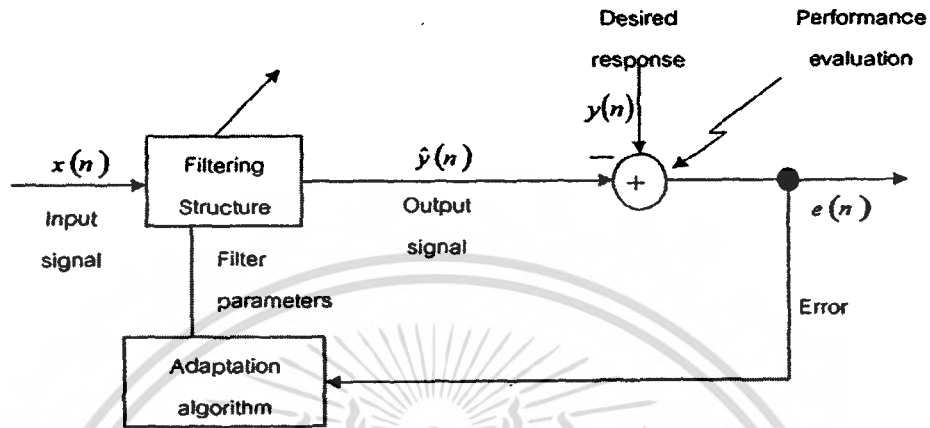
ในการออกแบบอะแดปทีฟฟิลเตอร์ บางชนิดนั้นมีความจำเป็นต้องทราบข้อมูลเบื้องต้นและลักษณะเฉพาะเจาะจงของการประยุกต์ใช้งานนั้นๆ ซึ่งข้อมูลดังกล่าวนี้จำเป็นอย่างมากสำหรับการออกแบบเพื่อใช้เป็นเกณฑ์ในการกำหนดสมรรถนะ และเลือก โครงสร้างของฟิลเตอร์ขั้นตอนการออกแบบอะแดปทีฟอัลกอริทึมเป็นขั้นตอนที่ยากที่สุดในการออกแบบและประยุกต์ใช้อะแดปทีฟฟิลเตอร์

จุดมุ่งหมายของอะแดปทีฟฟิลเตอร์คือ หากค่าพารามิเตอร์ที่ทำให้เกิดค่าสมรรถนะที่ดีที่สุดและจึงหยุดทำการปรับค่าพารามิเตอร์ ภายเวลาดังแต่เริ่มต้นกระบวนการฟิลเตอร์จนถึงกระทั่งเวลาที่ทำให้สมรรถนะเข้าใกล้ค่าที่ดีที่สุดเราเรียกว่า โหมดการลู่เข้า และถ้าหากสภาพของสัญญาณเปลี่ยนแปลงตามเวลา จะต้องกำหนดค่าเริ่มต้นให้อะแดปทีฟฟิลเตอร์ทำงานอย่างต่อเนื่องค่าพารามิเตอร์เปลี่ยนแปลงไปตามจนกระทั่งได้ค่าสมรรถนะที่ดีที่สุด จะเรียกว่าโหมดการติดตาม

ปัจจัยที่มีผลกระทบมากในการออกแบบอะแดปทีฟฟิลเตอร์คือ ผลตอบสนองของสัญญาณที่ต้องการเท่าที่จะหาได้ บางครั้งผลตอบสนองที่ความต้องการนี้ไม่สามารถหาค่าได้จากการใช้อะแดปทีฟฟิลเตอร์ ดังนั้นการปรับตัวจะต้องอาศัยวิธีการใดวิธีหนึ่ง ดังนี้

Supervised adaptation ที่ช่วงระยะเวลาใดๆ อะแดปทีฟฟิลเตอร์จะทราบถึงผลตอบสนองที่ความต้องการ จำนวนค่าผิดพลาด (ความแตกต่างระหว่างผลตอบสนองที่ความต้องการและผลตอบสนองจริง)

หาค่าเกณฑ์ของสมรรถนะและนำค่าที่ได้เหล่านี้ไปปรับค่าสัมประสิทธิ์ของฟิลเตอร์ ซึ่งโครงสร้างของ supervised อดะแคปทีฟฟิลเตอร์แสดงในรูปที่ 3.2



รูปที่ 3.2 ส่วนประกอบพื้นฐานของ supervised อดะแคปทีฟฟิลเตอร์

### 3.3 กรรณวิธีการพัฒนาอัลกอริทึมของวงจรรองความถี่แบบปรับตัวได้

ในการพัฒนาแต่ละอัลกอริทึมจะพบว่าแต่ละอัลกอริทึมมีคุณลักษณะเฉพาะตัว จึงมีความสำคัญมากในการที่จะทำความเข้าใจในความสามารถและขีดจำกัดเพื่อให้การเลือกใช้งานมีประสิทธิภาพ ยกตัวอย่างเช่น

กรรณวิธีการประมาณแบบสโตแคสติก เกรเดียนท์ (Stochastic Gradient Approach) เป็นกรรณวิธีที่วัดค่าความผิดพลาด โดยใช้ค่าเฉลี่ยผลต่างกำลังสองและพยายามลดค่าเฉลี่ยผลต่างกำลังสองโดยการทำการหาค่าจุดต่ำสุดบนเออร์เพอร์ฟอร์มันซ์เซอร์เฟส (Error Performance Surface) ซึ่งค่าต่ำสุดจะให้ค่าสัมประสิทธิ์ที่ทำให้ค่าเฉลี่ยผลต่างกำลังสองมีค่าต่ำสุด การหาอัลกอริทึมเพื่อใช้ในการปรับค่าสัมประสิทธิ์จะทำการแก้สมการของวินเนอร์และฮอปฟ์ (Wiener-Hopf Equation) อัลกอริทึมในกลุ่มนี้ที่นิยมใช้คือลีสต์มีนสแควร์ (Least Mean Square) หรือเรียกอีกอย่างว่าแอลเอ็มเอสอัลกอริทึม (LMS Algorithm)

### 3.4 การวัดประสิทธิภาพในระบบปรับตัว

ในระบบปรับตัวทุกระบบมีความจำเป็นที่จะต้องทำการวัดประสิทธิภาพ โดยทำการเปรียบเทียบระหว่างอัลกอริทึมต่างๆ ในตัวกรองปรับตัวก็เช่นกัน นอกจากจะต้องทำการเปรียบเทียบสัมประสิทธิ์ของตัวกรองแล้วยังต้องทำการเปรียบเทียบประสิทธิภาพของอัลกอริทึมอีก เพื่อช่วยในการตัดสินใจเลือกใช้อัลกอริทึมที่ให้การตอบสนองที่ดี การวัดประสิทธิภาพของอัลกอริทึมมีหลายวิธีดังจะกล่าวต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4.1 ความเร็วในการลู่เข้า (Convergence Rate)

อัตราการลู่เข้าของระบบปรับตัวเป็นคุณสมบัติที่สำคัญมากซึ่งจะต้องทำการวัดเพื่อให้ได้ตามความต้องการของงานที่จะนำไปใช้ซึ่งโดยทั่วไปแล้วความเร็วในการลู่เข้านั้นสามารถเปรียบเทียบประสิทธิภาพของอัลกอริทึมได้ อย่างไรก็ตามการเลือกใช้อัลกอริทึมไม่จำเป็นต้องใช้อัลกอริทึมที่มีความเร็วสูงสุดเนื่องจากการเพิ่มความเร็ว ราคาในการสร้างและความซับซ้อนของวงจรก็จะส่งขึ้นเช่น การเลือกใช้อัลกอริทึมขึ้นอยู่กับความจำเป็นในการใช้มากกว่า

### 3.4.2 ค่าความผิดพลาดเฉลี่ยกำลังสอง (Mean Square Error)

จะวัดในลักษณะของค่าความผิดพลาดเฉลี่ยกำลังสองที่น้อยที่สุด (Minimum Mean Square Error : MMSE) ซึ่งในการวัดความสามารถในการทำงานของระบบในการลดสัญญาณรบกวน (Eliminating Noise) ทำนายสัญญาณ (Signal prediction) หรือแยกแยะระบบ (System Identifying) ในระบบที่ไม่ทราบคุณลักษณะ (Unknown System) แล้วแต่ว่าเป็นระบบอะไร โดยทั่วไปแล้วค่าความผิดพลาดเฉลี่ยกำลังสองขึ้นกับหลายปัจจัย เช่น โครงสร้างของตัวกรอง ความไวของสัมประสิทธิ์ ตลอดจนสัญญาณรบกวนทั่วไป

### 3.4.3 ความถูกต้องในการประมาณค่าตัวแปร (Parameter Estimation Accuracy)

ความถูกต้องในการประมาณค่าตัวแปรเป็นปัจจัยที่มีความสำคัญมากเมื่อมีการนำระบบปรับตัวไปใช้ในการหาคุณลักษณะของระบบ (System Identifying) ในระบบที่ไม่ทราบ (Unknown System) ซึ่งความถูกต้องในการประมาณค่าตัวแปรมีมาก จะทำให้ได้แบบจำลองของระบบที่ต้องการใกล้เคียงความจริงมากขึ้น

### 3.4.4 ความซับซ้อนในการประมวลผล (Computation Complexity)

ความซับซ้อนในการประมวลผลมีความสำคัญมากในการนำระบบปรับตัวไปใช้งานจริง โดยทั่วไปต้องการความซับซ้อนในการประมวลผลที่ต่ำ ทำให้การสร้างมีราคาถูกลง นอกจากนั้นหากนำไปโปรแกรมลงในตัวประมวลผลสัญญาณสำเร็จรูปแล้วนั้นก็ต้องคำนึงถึงความสามารถในการทำงานของตัวประมวลผลเปรียบเทียบกับความซับซ้อนในการประมวลผลของอัลกอริทึมด้วย

### 3.4.5 เสถียรภาพ (Stability)

ความมีเสถียรภาพของระบบมีความสำคัญมากในการใช้งานตัวกรองปรับตัวได้ โดยเฉพาะตัวกรองแบบไอโออาร์ เนื่องจากอาจเกิดการเลื่อนของโพลออกนอกวงกลมหนึ่งหน่วย ทำให้เกิดการออสซิลเลตและทำให้ค่าลู่ออก (Diverge) จนไม่สามารถทำงานได้ การออกแบบจึงต้องคำนึงเสถียรภาพของตัวกรองด้วย

### 3.4.6 ความคงทนของระบบ (Robustness)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

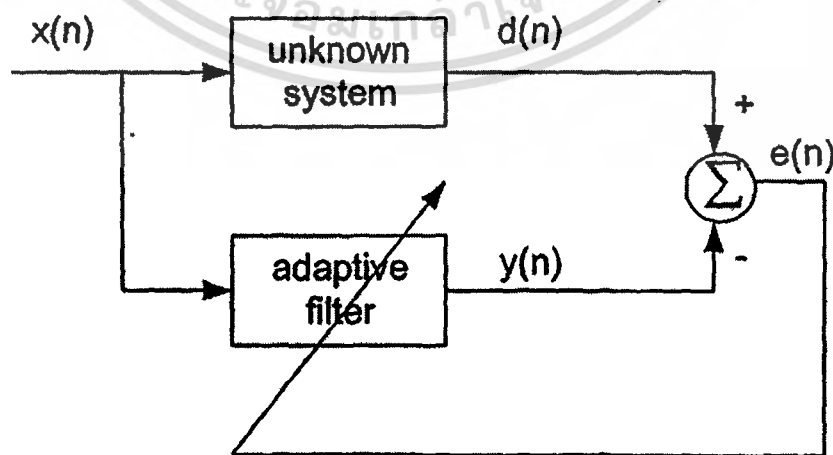
เนื่องจากค่าอินพุตของตัวกรองปรับตัวได้มักมีค่าต่างๆที่ไม่แน่นอนจึงมีความจำเป็นที่ตัวกรองจะต้องมีความทนต่อความแปรปรวนของสัญญาณ โดยทั่วไปความคงทนของระบบเป็นค่าที่ขาดต่อการวัดในการทดลอง โดยทั่วไปจึงไม่ทำการวัดคุณลักษณะนี้

### 3.5 การประยุกต์ใช้งานวงจรกรองปรับตัวได้

ตัวกรองปรับตัวได้มีคุณสมบัติในการปรับตัวได้อย่างอัตโนมัติทำให้ถูกนำมาประยุกต์ใช้ อุปกรณ์หลายๆอย่าง

#### 3.5.1 ตัวอย่างการนำตัวกรองปรับตัวได้ไปประยุกต์การใช้งาน

1. การประยุกต์ในการจำลองคุณลักษณะของระบบ (System Identification) ในระบบที่ไม่ทราบ (Unknown System) สมมติว่ามีระบบที่ไม่เป็นที่รู้จักและต้องการพิจารณาผลตอบสนองของระบบนั้นต่อสัญญาณที่ป้อนให้กับระบบ โดยสมมติว่าระบบนี้ไม่เปลี่ยนแปลงไปตามเวลาและมีคุณสมบัติเชิงเส้น และต้องการที่จะพัฒนาแบบจำลอง (Model) สำหรับระบบนี้โดยใช้ตัวกรองปรับตัวได้ เพื่อที่จะสร้างสัญญาณเอาต์พุตให้เหมือนกับเอาต์พุตของระบบที่ไม่เป็นที่รู้จัก โดยการป้อนอินพุตที่เหมือนกันให้แก่ระบบทั้งสองและทำการเปรียบเทียบเอาต์พุตทั้งสอง เพื่อสร้างสัญญาณผิดพลาด ซึ่งเป็นความแตกต่างระหว่างสัญญาณทั้งสองจากกล่าวได้ว่าการทำงานของตัวกรองปรับตัวได้เป็นการพยายามปรับผลตอบสนองให้เหมือนกับผลตอบสนองของระบบที่ไม่เป็นที่รู้จัก เพื่อที่จะทำให้สัญญาณผิดพลาดมีน้อยที่สุดคือ ค่าความผิดพลาดเฉลี่ยกำลังสอง และสำหรับในกรณีที่สัญญาณอินพุตของระบบที่ไม่เป็นที่รู้จักนั้นคงที่และระบบที่ไม่เป็นที่รู้จัก ไม่เปลี่ยนแปลงไปตามเวลาจะทำให้ค่าความผิดพลาดลดลงสู่ศูนย์ (Convergence) ในขณะนั้นเองกล่าวได้ว่าตัวกรองปรับตัวทำการจำลองคุณลักษณะของระบบที่ไม่ทราบได้แล้ว ดังรูป 3.3



รูปที่ 3.3 แสดงการประยุกต์ใช้ตัวกรองปรับตัวในการจำลองคุณลักษณะของระบบที่ไม่ทราบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

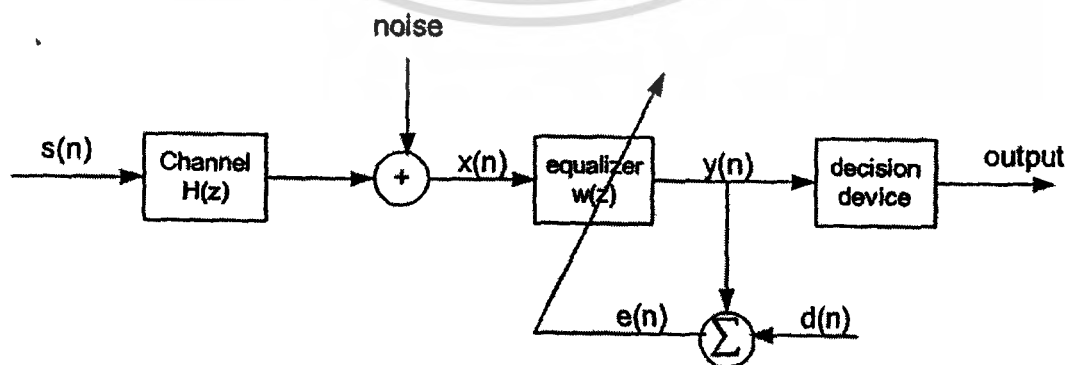
## สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2. การพัฒนาอีควอลไลเซอร์แบบปรับตัวได้ (Adaptive Equalization) ระบบการส่งข้อมูลทั่วไปจะพยายามใช้ประโยชน์จากช่องสัญญาณที่มีอยู่ให้มีประสิทธิภาพที่สุด โดยออกแบบระบบการส่งข้อมูลด้วยอัตราสูงที่สุดเท่าที่จะเป็นไปได้ ภายใต้ความน่าเชื่อถือที่กำหนด ซึ่งปกติถูกวัดอยู่ในรูปของอัตราการผิดพลาดหรือความน่าจะเป็นเฉลี่ยของการผิดพลาดหรือความน่าจะเป็นเฉลี่ยของการผิดพลาดของสัญลักษณ์ (Average probability of symbol error) การส่งข้อมูลดิจิทัลผ่านช่องทางการสื่อสารแบบเชิงเส้นถูกจำกัดด้วยองค์ประกอบสองอย่าง

- การแทรกซ้อนระหว่างสัญลักษณ์ เกิดเป็นการแทรกซ้อนของสัญญาณที่ถูกส่ง ซึ่งเป็นผลมาจากการเบี่ยงเบนของการตอบสนองความถี่ในช่องทาง

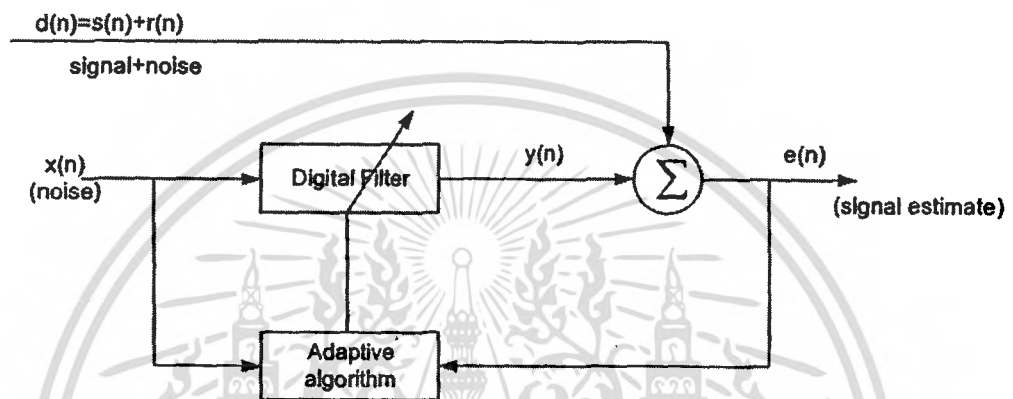
- การรบกวนจากเทอร์มอลนอยส์ เกิดจากสัญญาณรบกวนซึ่งเกิดจากอุณหภูมิ ซึ่งโดยทั่วไปมักเกิดในอุปกรณ์ต่างๆของระบบสื่อสารสำหรับช่องทางสื่อสารที่จำกัด เช่น โทรศัพท์ปกติจะพบว่าการแทรกซ้อนระหว่างสัญลักษณ์เป็นองค์ประกอบสำคัญที่ใช้พิจารณาในการกำหนดอัตราส่งของข้อมูล

เพื่อที่จะแก้ปัญหาคการแทรกซ้อนระหว่างสัญลักษณ์ต้องมีตัวกรอง โดยถ้าคุณลักษณะของช่องสัญญาณเป็นที่รู้อย่างแน่นอนแล้ว จะสามารถออกแบบตัวกรองทั้งภาคส่งและภาครับได้ ซึ่งจะลดผลของการแทรกซ้อนระหว่างสัญลักษณ์ลงได้และยังช่วยลดสัญญาณรบกวนที่เพิ่มขึ้นมา แต่อย่างไรก็ตามในทางปฏิบัติเราพบว่าคุณลักษณะของช่องสัญญาณสื่อสารนั้นเป็นแบบสุ่มฉะนั้นในการใช้ตัวกรองภาคส่งและภาครับที่ออกแบบ โดยคุณลักษณะของช่องสัญญาณเฉลี่ยจะไม่เพียงพอที่จะลดการแทรกซ้อนระหว่างสัญลักษณ์จึงจำเป็นที่จะต้องใช้อีควอลไลเซอร์แบบปรับตัวได้ ดังรูปที่ 2.7 ซึ่งจะลดการแทรกซ้อนระหว่างสัญลักษณ์ ได้ดีขึ้น โดยพื้นฐานสำหรับการอีควอลไลซ์ของระบบการส่งข้อมูลที่มีอยู่ คือการเพิ่มขนาดสัญญาณที่บางย่าน (Pre-Equalization) ที่ภาคส่งและการลดขนาดสัญญาณที่บางย่าน (Post-Equalization) ที่ภาครับ



รูปที่ 3.4 แสดงการใช้อีควอลไลเซอร์แบบปรับตัวได้ในระบบส่งสัญญาณ

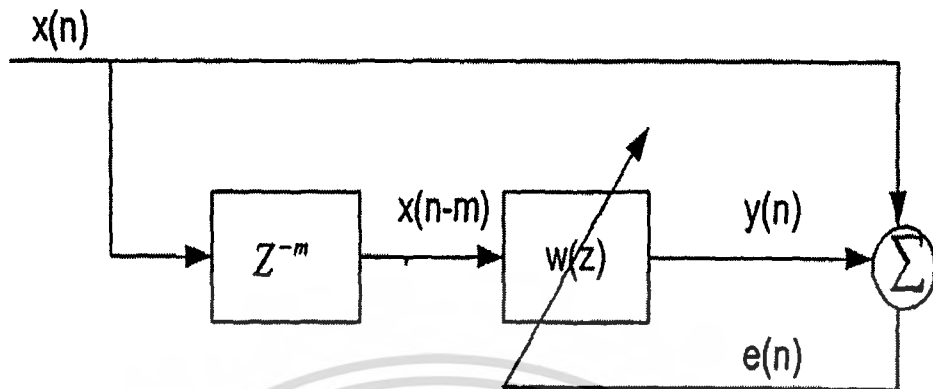
3. การกำจัดสัญญาณรบกวน (Noise Cancellation) การนำตัวกรองปรับตัวได้มาประยุกต์ใช้ในการกำจัดสัญญาณรบกวนนั้น โดยใช้สัญญาณรบกวนที่มีความสัมพันธ์กับสัญญาณรบกวนที่ป้อนมาในสัญญาณที่ต้องการ ในการกำจัดสัญญาณรบกวน โดยการป้อนสัญญาณและโครงสร้างของระบบกำจัดสัญญาณรบกวนเป็นไปตามรูปที่ 3.5



รูปที่ 3.5 แสดงโครงสร้างระบบกำจัดสัญญาณรบกวนโดยใช้ตัวกรองปรับตัวได้

จากรูปป้อนสัญญาณรบกวน  $x(n)$  ซึ่งมีความสัมพันธ์กับสัญญาณรบกวน  $r(n)$  ให้แก่ระบบ จะได้เอาท์พุทของตัวกรองปรับตัว  $y(n)$  แล้วจึงนำไปหักล้างกับสัญญาณรบกวน  $r(n)$  ซึ่งปะปนอยู่กับสัญญาณที่ต้องการ  $s(n)$  ทำให้ได้สัญญาณที่ต้องการ  $e(n)$  ซึ่งเป็นสัญญาณที่มีสัญญาณรบกวนน้อยลงและสัญญาณนี้จะถูกป้อนให้กับตัวกรองเพื่อใช้ในการเปรียบเทียบในการปรับค่าสัมประสิทธิ์อีกทีหนึ่ง

4. การพัฒนาอุปกรณ์เพิ่มประสิทธิภาพสาย (Adaptive Line Enhancer) อุปกรณ์เพิ่มประสิทธิภาพสาย ดังรูปที่ 3.6 เป็นอุปกรณ์ที่สามารถใช้แยกสัญญาณแบนด์แคบซึ่งฝังอยู่ในสัญญาณแบนด์กว้าง ซึ่งขึ้นกับการใช้งานสัญญาณที่ต้องการแยกออกอาจเป็นสัญญาณที่สนใจหรือสัญญาณรบกวนที่ไม่ต้องการ ตัวอย่างของการใช้งานเช่น การลดสัญญาณรบกวนความถี่ 50–60 Hz ออกจากสัญญาณของอุปกรณ์ทางการแพทย์



รูปที่ 3.6 แสดงอุปกรณ์เพิ่มประสิทธิภาพสาย

### 3.6 การเปรียบเทียบระหว่างฟิลเตอร์ที่เหมาะสมที่สุดกับอะแดปทีฟฟิลเตอร์

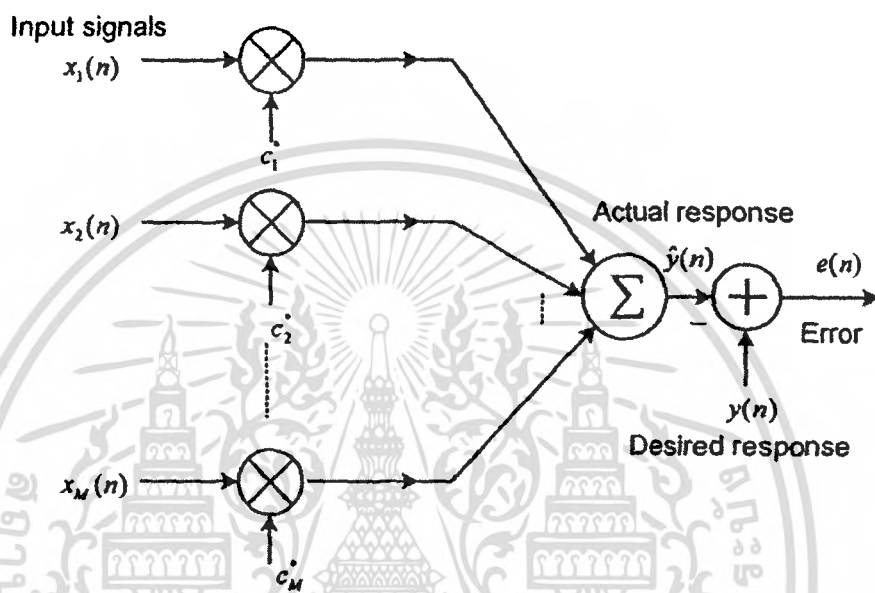
ฟิลเตอร์ที่เหมาะสมที่สุด (optimal filter) เป็นเสมือนเครื่องมือทางคณิตศาสตร์และไม่สามารถนำมาประยุกต์ใช้ได้จริงในทางปฏิบัติ เพราะเราไม่ทราบข้อมูลทางสถิติที่เกิดขึ้นจริงเช่น ค่าโมเมนต์อันดับสอง (second-order moments) ซึ่งจำเป็นต้องใช้ในการออกแบบค่าฟิลเตอร์โดย อะแดปทีฟฟิลเตอร์นี้สามารถกล่าวได้ว่าในทางปฏิบัติและมีความคล้ายคลึงกันมากกับฟิลเตอร์ที่เหมาะสมที่สุดโดยอาศัยกระบวนการในการวัดของสัญญาณต่างๆ ในเวลาจริง ซึ่งต้องสร้างส่วนของข้อมูลทางสถิติที่ขาดหายไปนั้นขึ้นมาเอง

สำหรับการวิเคราะห์แล้ว โดยทั่วไปของตัวรวมสัญญาณแบบเชิงเส้น (linear combiner) จะรวมไปถึงการฟิลเตอร์และการทำนายด้วย อย่างไรก็ตามเพื่อความสะดวกในการคำนวณจะใช้เทอมของฟิลเตอร์ ซึ่งความแตกต่างระหว่างตัวรวมสัญญาณแบบเชิงเส้นและ FIR ฟิลเตอร์หรือตัวทำนายคือ สัญญาณหรือข้อมูลทางอินพุตจะถูกเลื่อนเวลาไปดังแสดงในรูปที่ 3.7 ซึ่งมีความสำคัญในขั้นตอนของการสร้างอะแดปทีฟฟิลเตอร์ สัญญาณอินพุตมี  $M$  สัญญาณกำหนดให้เป็น  $x_k(n, \zeta)$  และผลตอบสนองของสัญญาณที่ต้องการคือ  $y(n, \zeta)$  สัญญาณเหล่านี้สามารถสร้างได้ง่ายโดยใช้ลำดับแบบสุ่ม (random sequence) ค่าประมาณของ  $y(n, \zeta)$  สามารถหาได้โดยการใช้รวมตัวสัญญาณแบบเชิงเส้น ดังสมการ

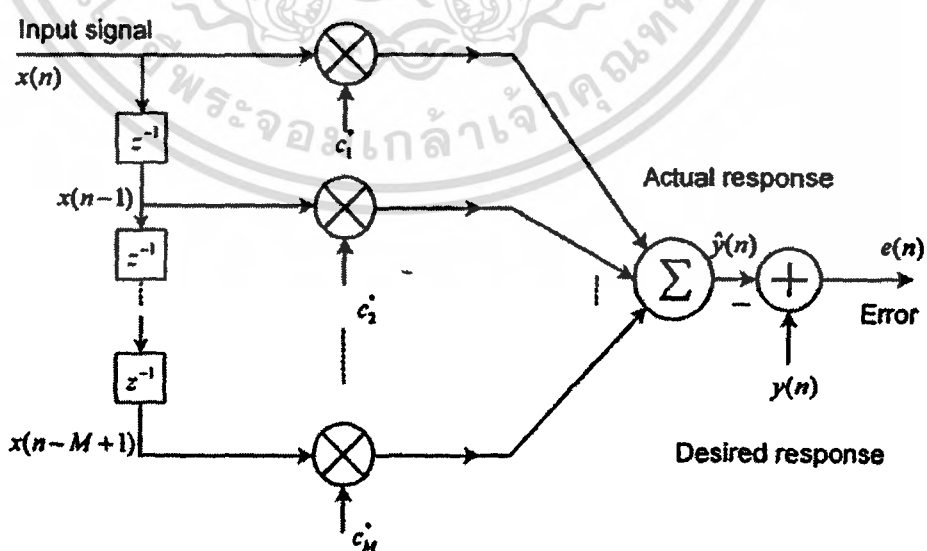
$$\hat{y}(n, \zeta) = \sum_{k=1}^M c_k^*(n) x_k(n, \zeta) \equiv c^H(n) x(n, \zeta) \quad (3-1)$$

โดยที่สัมประสิทธิ์  $c(n)$  หาได้จากสมการ

$$c(n) = [c_1(n) \ c_2(n) \dots c_m(n)]^T \quad (3-2)$$



รูปที่ 3.7 (ก) ตัวรวมสัญญาณเชิงเส้น ที่มีหลายอินพุต



รูปที่ 3.7 (ข) ตัวรวมสัญญาณเชิงเส้น FIR ฟิลเตอร์อินพุตเดี่ยว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และสัญญาณเวกเตอร์ของสัญญาณอินพุตคือ

$$x(n, \zeta) = [x_1(n, \zeta) \ x_2(n, \zeta) \dots x_M(n, \zeta)]^T \quad (3-3)$$

สำหรับการประยุกต์ใช้กับตัวตรวจจับอินพุตเดียว เวกเตอร์ของข้อมูลอินพุตจะถูกเลื่อนเวลาไปจากสมการ (3-3) สามารถเขียนใหม่ได้เป็น

$$x(n) = [x(n, \zeta) \ x(n-1, \zeta) \dots x(n-M+1, \zeta)]^T \quad (3-4)$$

และตัวรวมสัญญาณแบบเชิงเส้นเขียนในรูปแบบของ FIR ฟิลเตอร์จะได้

$$\hat{y}(n, \zeta) = \sum_{k=0}^{M-1} h(n, k)x(n-k, \zeta) \equiv c^H(n)x(n, \zeta) \quad (3-5)$$

โดยที่  $c_k(n) = h^*(n, k)$  คือค่าของผลตอบสนองของอิมพัลส์ที่เวลา  $n$

- ฟิลเตอร์ที่เหมาะสมที่สุด

ในกรณีที่เราทราบค่าโมเมนต์อันดับที่สองของสัญญาณ เราจะสามารถออกแบบฟิลเตอร์ที่เหมาะสมที่สุด โดยพิจารณา  $c_o(n)$  จากสมการต่อไปนี้

$$R(n)c_o(n) = d(n) \quad (3-6)$$

โดยที่

$$R(n) = E\{x(n, \zeta)x^H(n, \zeta)\} \quad (3-7)$$

เมื่อ  $R(n)$  คือเมตริกซ์สหสัมพันธ์ของสัญญาณอินพุตและ

$$d(n) = E\{x(n, \zeta)y^*(n, \zeta)\} \quad (3-8)$$

$d(n)$  คือค่าสหสัมพันธ์ข้ามระหว่างสัญญาณอินพุตและผลตอบสนองของสัญญาณที่ต้องการตามลำดับ ในระหว่างกระบวนการปกติฟิลเตอร์ที่เหมาะสมที่สุดจะทำการสร้างสัญญาณขึ้น นั่นคือ

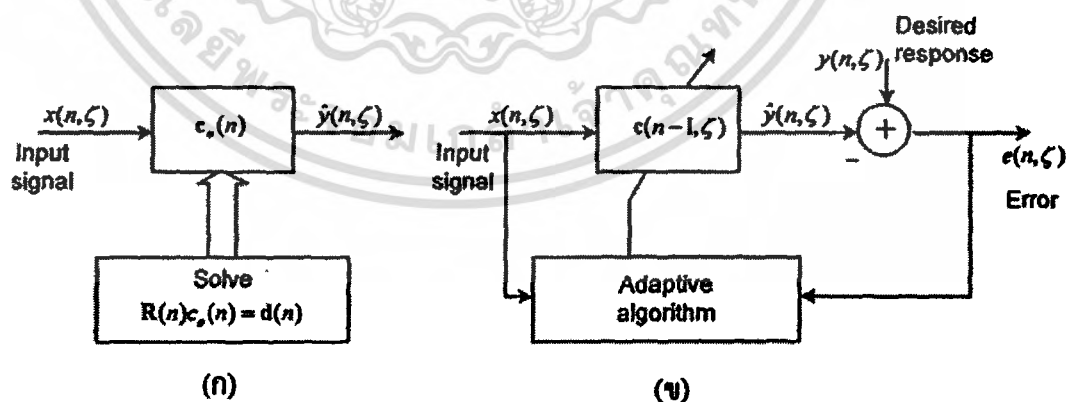
$$\hat{y}_o(n, \zeta) = c_o^H(n)x(n, \zeta) \quad (3-9)$$

$$\varepsilon_o(n, \zeta) = y(n, \zeta) - \hat{y}_o(n, \zeta) \quad (3-10)$$

เมื่อค่า  $\hat{y}_o(n, \zeta)$  คือค่าประมาณที่เหมาะสมที่สุด และค่า  $\varepsilon_o(n, \zeta)$  คือค่าผิดพลาดชั่วขณะที่เหมาะสมที่สุด ให้พิจารณารูปที่ 3.8 (ก) อย่างไรก็ตามฟิลเตอร์จะหาค่าที่เหมาะสมที่สุด โดยพิจารณาจากสัญญาณต่างๆและค่า MMSE

$$P_o(n) = E\{|\varepsilon_o(n, \zeta)|^2\} = P_y(n) - d^H(n)c_o(n) \quad (3-11)$$

จากสมการ (3-11) แสดงให้เห็นว่าฟิลเตอร์จะคำนวณหาค่าเฉลี่ย ดังนั้นจะได้ว่าสัมประสิทธิ์ที่ค่าที่สุดของฟิลเตอร์นั้นเป็นค่าที่ไม่ต้องสุ่ม และผลตอบสนองที่ต้องการนั้นไม่ได้เป็นส่วนประกอบที่จำเป็นในกระบวนการของฟิลเตอร์ที่เหมาะสมที่สุด (พิจารณาที่สมการ(3-9))



รูปที่ 3.8 ความแตกต่างของกระบวนการระหว่าง (ก) ฟิลเตอร์ที่เหมาะสมที่สุด และ (ข)อะแดปทีฟฟิลเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าหากสัญญาณไม่แปรตามเวลา ฟิลเตอร์ที่เหมาะสมที่สุดจะคำนวณสัมประสิทธิ์เพียงครั้งเดียวและใช้ในการสร้างสัญญาณ  $\{x(n, \zeta), y(n, \zeta)\}$  สำหรับสถานะที่สัญญาณแปรตามเวลา การออกแบบฟิลเตอร์ที่เหมาะสมที่สุดจะถูกกระทำซ้ำที่ทุกๆ ค่าเวลาคงที่  $n$  เพราะว่าฟิลเตอร์ที่เหมาะสมที่สุดเป็นระบบที่แปรตามเวลา

- อะแคปทีฟฟิลเตอร์

ในทางปฏิบัติโดยส่วนใหญ่ค่าโมเมนต์อันดับที่สองของ  $R(n)$  และค่า  $d(n)$  เป็นตัวแปรที่ไม่ทราบค่า การใช้อะแคปทีฟฟิลเตอร์จะเป็นคำตอบที่ดีที่สุด ถ้าสัญญาณเป็นลักษณะรายคาบจะได้

$$R = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{n=-N}^N x(n, \zeta) x^H(n, \zeta) \quad (3-12)$$

$$d = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{n=-N}^N x(n, \zeta) x^H(n, \zeta) \quad (3-13)$$

เพราะว่าค่าเฉลี่ยของจำนวนครั้งการคำนวณจะมีค่าเท่ากับค่าเฉลี่ยของเวลา ถ้าจำนวนข้อมูล  $\{x(n, \zeta), y(n, \zeta)\}_0^{N-1}$  มีค่าเพียงพอ จะได้ค่าประมาณที่ยอมรับได้ของฟิลเตอร์ที่เหมาะสมที่สุด โดยการคำนวณค่าประมาณได้ดังนี้

$$\hat{R}_N(\zeta) = \frac{1}{N} \sum_{n=0}^{N-1} x(n, \zeta) x^H(n, \zeta) \quad (3-14)$$

$$\hat{d}_N(\zeta) = \frac{1}{N} \sum_{n=0}^{N-1} x(n, \zeta) y^*(n, \zeta) \quad (3-15)$$

โดยการเฉลี่ยเวลาและทำการแก้ระบบเชิงเส้น จะได้

$$\hat{R}_{N(\zeta)} c_{N(\zeta)} = \hat{d}_{N(\zeta)} \quad (3-16)$$

ข้อมูลที่ได้อะแคปทีฟฟิลเตอร์ ณ เวลาที่  $n$  คือ เวกเตอร์ข้อมูลอินพุต  $x(n, \zeta)$  ผลตอบสนองที่ต้องการ  $y(n, \zeta)$  และค่าเวกเตอร์สัมประสิทธิ์ที่ล่าสุด  $c(n-1, \zeta)$

อะแคปเตอร์ที่แต่ละเวลาของ  $n$  จะมีขั้นตอนการคำนวณดังนี้

## 1. การฟิลเตอร์

$$\hat{y}(n, \zeta) = c^H(n-1, \zeta)x(n, \zeta) \quad (3-17)$$

## 2. ค่าผิดพลาด

$$e(n, \zeta) = y(n, \zeta) - \hat{y}(n, \zeta) \quad (3-18)$$

## 3. อะแดปทีฟฟิลเตอร์

$$c(n, \zeta) = c(n-1, \zeta) + \Delta c \{x(n, \zeta), e(n, \zeta)\} \quad (3-19)$$

การเปลี่ยนแปลงเพิ่มค่าในพจน์  $\Delta c(n, \zeta)$  ทำได้โดยการเลือกค่า  $c(n, \zeta)$  ให้มีค่าเข้าใกล้  $c_0$  เมื่อ  $c(n, \zeta) \approx c_0$  จะได้การประมาณค่าที่ดี ข้อสังเกต ค่าประมาณ  $\hat{y}(n, \zeta)$  ของผลตอบสนองที่ต้องการ  $y(n, \zeta)$  เป็นค่าที่คำนวณจากอินพุตปัจจุบัน  $x(n, \zeta)$  และสัมประสิทธิ์ของฟิลเตอร์ในอดีต  $c(n-1, \zeta)$  ค่าประมาณ  $\hat{y}(n, \zeta)$  และค่าผิดพลาด  $e(n, \zeta)$  เป็นค่าที่ประมาณได้เทียบกับค่าประมาณที่เกิดขึ้นจริง โดยใช้ค่าสัมประสิทธิ์ปัจจุบัน  $c(n, \zeta)$  วิธีการหาค่าสัมประสิทธิ์ของฟิลเตอร์ที่ใช้ค่าผิดพลาด  $e(n, \zeta)$  จะถูกเรียกว่า priori type อะแดปทีฟอัลกอริทึม

ถ้าเราใช้การประมาณค่าจริงที่ได้จากการใช้ค่าสัมประสิทธิ์ปัจจุบัน  $c(n, \zeta)$  อะแดปทีฟฟิลเตอร์จะได้

## 1. การฟิลเตอร์

$$\hat{y}_a(n, \zeta) = c^H(n, \zeta)x(n, \zeta) \quad (3-20)$$

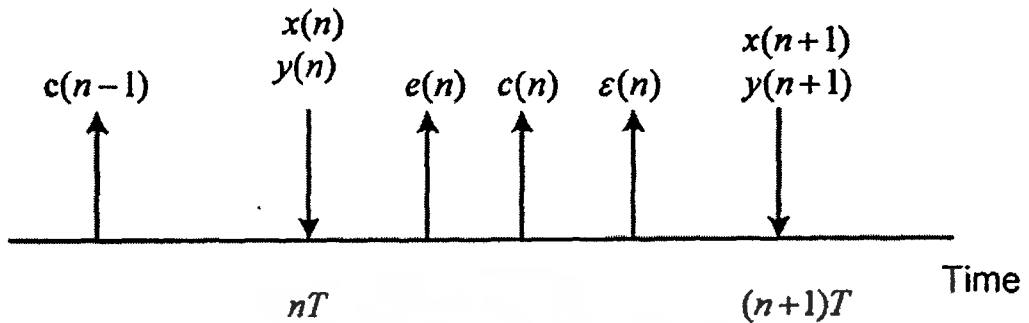
## 2. ค่าผิดพลาด

$$e(n, \zeta) = y(n, \zeta) - \hat{y}_a(n, \zeta) \quad (3-21)$$

## 3. อะแดปทีฟอัลกอริทึม

$$c(n, \zeta) = c(n-1, \zeta) + \Delta c \{x(n, \zeta), e(n, \zeta)\} \quad (3-22)$$

ซึ่งถูกเรียกว่า posteriori type อะแดปทีฟอัลกอริทึม โดยเทอม priori และ posteriori นั้นได้ถูกกล่าวไว้ใน Carayannis et al [7] โคอะแกรมเวลาสำหรับทั้งสองอัลกอริทึมดังกล่าวนี้แสดงในรูปที่ 3.9



รูปที่ 3.9 ไคอะแกรมเวลาสำหรับ priori และ posteriori อดแแปทีฟอัลกอริทึม

กล่าวโดยสรุปคือ วัตถุประสงค์ของอดแแปทีฟฟิลเตอร์ที่ใช้เพื่อหาค่าข้อมูลเป็นเวลา  $n$  เช่น  $\{x(n, \zeta), y(n, \zeta), c(n-1, \zeta)\}$  เป็นการอัปเดตค่าเวกเตอร์สัมประสิทธิ์ตัวเก่า  $c(n-1, \zeta)$  จะเป็นค่าสัมประสิทธิ์โดยประมาณตัวใหม่  $c(n, \zeta)$  ดังนั้นค่า  $c(n, \zeta)$  จะเป็นค่าที่เข้าใกล้ฟิลเตอร์ที่เหมาะสมที่สุด  $c_o(n)$  และค่าเอาต์พุต  $\hat{y}(n)$  คือ ค่าประมาณที่ดีที่สุดของผลตอบสนองที่ต้องการ  $y(n)$  ดังนั้นการทำงานของอดแแปทีฟอัลกอริทึมโดยมากจะใช้รูปแบบดังนี้

$$\begin{bmatrix} \text{New} \\ \text{coefficient} \\ \text{vector} \end{bmatrix} = \begin{bmatrix} \text{old} \\ \text{coefficient} \\ \text{vector} \end{bmatrix} + \begin{bmatrix} \text{adaptation} \\ \text{gain} \\ \text{vector} \end{bmatrix} \cdot \begin{pmatrix} \text{error} \\ \text{signal} \end{pmatrix} \quad (3-23)$$

เมื่อสัญญาณที่ผิดพลาด (error signal) คือ ความแตกต่างระหว่างผลตอบสนองที่ต้องการและค่าเอาต์พุตจริงของอดแแปทีฟฟิลเตอร์

### 3.7 เสถียรภาพและภาพสมรรถนะเมื่ออยู่ตัวของอดแแปทีฟฟิลเตอร์

เป้าหมายของอดแแปทีฟฟิลเตอร์นั้นคือ เริ่มแรกจะต้องกำหนดค่า  $c(n, \zeta)$  และหลังจากนั้น จะทำการปรับเปลี่ยนค่าเพื่อเข้าสู่ฟิลเตอร์ที่เหมาะสมที่สุด  $c_o(n)$  อย่างรวดเร็วและเที่ยงตรงเท่าที่จะเป็นไปได้ ซึ่งสามารถหาค่าสมรรถนะได้โดยการคำนวณหาค่าของการเบี่ยงเบนของ  $c_o(n)$  ดังสมการ

$$\tilde{c}(n, \zeta) \equiv c(n, \zeta) - c_o(n) \quad (3-24)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อะแดปทีฟฟิลเตอร์ที่ยอมรับได้นั้นควรที่จะมีความเสถียรในกรณี bounded input bounded output (BIBO) และค่าของมันควรจะมีค่าเข้าใกล้ฟิลเตอร์ที่เหมาะสมที่สุด การวิเคราะห์ด้านเสถียรภาพของ BIBO นั้นทำได้ค่อนข้างยากเนื่องจากอะแดปทีฟฟิลเตอร์นั้นมีความจำเป็นไม่เชิงเส้นเป็นระบบที่ทำงานเป็นตามเวลา โดยที่ค่าสมรรถนะของอะแดปทีฟฟิลเตอร์หลักๆ แล้วสามารถวัดได้จากค่า MSE ที่เป็นฟังก์ชันของเวลา ดังนั้นปัญหาดังกล่าวที่ถูกลำมาพิจารณาในข้อแรกเราควรพิจารณาว่าอะแดปทีฟฟิลเตอร์ทำงานในภาวะที่สัญญาณไม่ผันตามเวลา และข้อที่สองพิจารณาในภาวะสัญญาณแปรผันตามเวลา เสถียรภาพ (stability)

อะแดปทีฟฟิลเตอร์จะเริ่มต้นทำงานที่เวลา  $n=0$  และประมวลผลค่า  $\{x(n, \zeta), y(n, \zeta)\}_0^\infty$  ที่ได้ลำดับของเวกเตอร์  $\{c(n, \zeta)\}_0^\infty$  โดยใช้อะแดปทีฟอัลกอริทึม โครงสร้างแบบ FIR นั้นมีความเสถียรภาพเสมอ ค่าเอาต์พุตหรือค่าความผิดพลาดของอะแดปทีฟฟิลเตอร์จะถูกกำจัดอยู่ถ้าหากค่าสัมประสิทธิ์ของฟิลเตอร์ยังมีค่าไม่เข้าใกล้ค่าสัมประสิทธิ์ของฟิลเตอร์ที่เหมาะสมที่สุด อย่างไรก็ตามปัจจุบันอะแดปทีฟฟิลเตอร์ได้ใช้ระบบป้อนกลับจึงมีผลต่อเสถียรภาพได้ ในกรณีสัญญาณไม่แปรตามเวลานั้นฟิลเตอร์ที่เหมาะสมที่สุด  $c_0$  เป็นค่าคงที่และการลู่เข้าของ  $c(n, \zeta)$  ไปยัง  $c_0$  ที่ค่า  $n \rightarrow \infty$  รับรองได้ว่าอะแดปทีฟฟิลเตอร์จะมีเสถียรภาพด้าน BIBO สำหรับการสร้าง  $\zeta$  ส่วนประกอบของ  $c_k(n, \zeta)$  ที่ลำดับที่  $k$ -th หรือค่า  $norm \|c(n, \zeta)\|$  ของเวกเตอร์  $c(n, \zeta)$  สัมประสิทธิ์  $c_k(n, \zeta)$  เป็นตัวแปรสุ่ม จึงจำเป็นต้องใช้แนวคิดการลู่เข้าแบบ stochastic ซึ่งได้กล่าวไว้ใน Papoulis [8]

เราจะกล่าวได้ว่าลำดับที่มีการลู่เข้าแบบสุ่ม ถ้าลำดับของ  $c_k(n, \zeta)$  นั้นมีค่าลู่เข้าทุกๆ ที่ของ  $\zeta$  นั่นคือ

$$\lim_{n \rightarrow \infty} c_k(n, \zeta) = c_{0,k}(\zeta) \quad (3-25)$$

เมื่อค่าลิมิต  $c_{0,k}(\zeta)$  นั้นขึ้นอยู่กับค่า  $\zeta$

ถ้าต้องการให้อะแดปทีฟฟิลเตอร์ลู่เข้าสำเร็จ เราสามารถใช้แนวคิดลู่เข้าของทุกๆ ค่าของ  $\zeta$  ซึ่งเรากล่าวได้ว่าเป็นค่าลำดับแบบสุ่มของ  $c_k(n, \zeta)$  นั้นทำการลู่เข้าที่ทุกๆ ค่าของ  $\zeta$  หรือมีค่าความน่าจะเป็นเท่ากับ 1 ถ้า

$$p\left\{\lim_{x \rightarrow \infty} |c_k(n, \zeta) - c_{0,k}(\zeta)| = 0\right\} = 1 \quad (3-26)$$

นั่นหมายความว่าบางลำดับของข้อมูลอาจจะไม่ลู่เข้า ซึ่งอาจเกิดในกรณีที่มีความน่าจะเป็น

เท่ากับศูนย์ตัวอย่างการลู่เข้าแบบอื่นที่ใช้ในอะแดปทีฟฟิลเตอร์กำหนดได้ดังสมการ

$$\lim_{n \rightarrow \infty} E \left\{ |c_k(n, \zeta) - c_{o,k}(\zeta)|^2 \right\} = \lim_{n \rightarrow \infty} E \left\{ |\tilde{c}_k(n, \zeta)|^2 \right\} = 0 \quad (3-27)$$

การลู่เข้าแบบนี้ถูกเรียกว่าการลู่เข้าหาค่าเฉลี่ยกำลังสอง เหตุผลสำหรับการใช้ค่าเฉลี่ยกำลังสอง คือ การลู่เข้าดังกล่าวจะไม่เหมือนการลู่เข้าได้ในทุกๆ กรณีเนื่องจากจะใช้ลำดับของข้อมูลเพียงบางส่วน ในการนำมาคิดผลกระทบโดยเฉลี่ยของลำดับข้อมูลทั้งหมด นอกจากนั้นยังสามารถใช้ค่าโมเมนต์อันดับที่สองสำหรับการลู่เข้าหาค่าเฉลี่ยกำลังสองซึ่งไม่ได้หมายความว่า การลู่เข้านั้นจะมีความหมาย น่าจะเป็นเท่ากับ 1 ดังนั้น

$$\frac{E \left\{ |\tilde{c}_k(n, \zeta)|^2 \right\}}{\delta} = \frac{E \left\{ |\tilde{c}_k(n, \zeta)|^2 \right\}}{\delta} + \frac{\text{var} \left\{ \tilde{c}_k(n, \zeta) \right\}}{\delta^2} \quad (3-28)$$

ถ้าสามารถแสดงว่า  $E \left\{ \tilde{c}_k(n) \right\} \rightarrow 0$  ที่ค่า  $n \rightarrow \infty$  และ  $\text{var} \left\{ \tilde{c}_k(n, \zeta) \right\}$  มีค่าที่แน่นอน สำหรับทุกค่าของ  $n$  ก็จะแน่ใจได้ว่าค่าสัมประสิทธิ์จะลู่เข้าได้

### 3.7.1 การวัดค่าสมรรถนะ (performance measurer)

สมรรถนะของอะแดปทีฟฟิลเตอร์สามารถหาค่าได้โดยใช้ ค่าเบี่ยงเบนกำลังสองเฉลี่ย (mean square deviation : MSD) ดังสมการ

$$D(n) \equiv E \left\{ \|c(n, \zeta) - c_o(n)\|^2 \right\} = E \left\{ \|\tilde{c}(n, \zeta)\|^2 \right\} \quad (3-29)$$

การหาค่าระยะเฉลี่ยระหว่างสัมประสิทธิ์ของอะแดปทีฟฟิลเตอร์และสัมประสิทธิ์ของฟิลเตอร์ที่เหมาะสมที่สุด แม้ว่าค่า MSD จะไม่สามารถวัดได้ในทางปฏิบัติ และนิยมนำมาศึกษาและวิเคราะห์อะแดปทีฟอัลกอริทึมจะให้ค่า  $D(n)$  ต่ำสุดสำหรับแต่ละค่าของ  $n$  ซึ่งทราบกันดีว่าเป็นอัลกอริทึมที่สามารถเรียนรู้หาจุดที่ดีที่สุดได้

ถ้าอินพุตเป็นเมตริกซ์สหสัมพันธ์แบบบวก สำหรับค่าเบี่ยงเบนใดๆ  $\tilde{c}(n)$  ของค่าสัมประสิทธิ์ของฟิลเตอร์ที่เหมาะสมที่สุด จากการกำหนดค่าที่เหมาะสมที่สุดเป็นการเพิ่มค่ากำลังสองเฉลี่ยมีค่าเท่ากับ  $\tilde{c}^H(n)R\tilde{c}(n)$  เรียกว่า excess MSE (EMSE) ในอะแดปทีฟฟิลเตอร์ ความเบี่ยงเบนแบบสุ่ม  $\tilde{c}(n, \zeta)$  จากผลที่เหมาะสมที่สุดในรูปของ EMSE ซึ่งสามารถหาได้ด้วยค่าเฉลี่ยของ  $\tilde{c}^H(n, \zeta)R\tilde{c}(n, \zeta)$  สำหรับอะแดปทีฟฟิลเตอร์แบบ posteriori ค่า MSE จะสามารถกำหนดได้

ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$P'(n) \equiv E\{|\varepsilon(n, \zeta)|^2\} \equiv P'_o(n) + P'_{ex}(n) \quad (3-30)$$

เมื่อ  $P'_{ex}(n)$  คือค่า EMSE และค่า  $P'_o(n)$  คือค่า MMSE กำหนดโดย

$$P'(n) \equiv E\{|\varepsilon_o(n, \zeta)|^2\} \quad (3-31)$$

และค่าผิดพลาดของฟิลเตอร์ที่เหมาะสมที่สุดแบบ posteriori กำหนดโดย

$$\varepsilon_o(n, \zeta) \equiv y(n, \zeta) - c_o^H(n)x(n, \zeta) \quad (3-32)$$

ค่า posteriori EMSE  $P'_{ex}(n)$  สามารถหาค่าได้ดังนี้

$$P'_{ex}(n) \equiv P'(n) - P'_o(n) \quad (3-33)$$

สำหรับ priori อะแดปทีฟอัลกอริทึมที่ใช้สัมประสิทธิ์เก่า  $c(n-1, \zeta)$  สามารถหา priori EMSE จากสมการต่อไปนี้

$$P_{ex}(n) \equiv P(n) - P_o(n) \quad (3-34)$$

เมื่อ

$$P(n) \equiv E\{|e(n, \zeta)|^2\} \quad (3-35)$$

$$P_o(n) \equiv E\{|\varepsilon_o(n, \zeta)|^2\} \quad (3-36)$$

และ

$$e_o(n, \zeta) \equiv y(n, \zeta) - c_o^H(n-1)x(n, \zeta) \quad (3-37)$$

ที่ซึ่งค่าผิดพลาดของฟิลเตอร์ที่เหมาะสมที่สุด ถ้าค่าสัญญาณไม่แปรตามเวลา จะได้  $\varepsilon_o(n, \zeta) = e_o(n, \zeta)$  ดังนั้นค่าผิดพลาดที่เหมาะสมที่สุดของ priori ฟิลเตอร์และ posteriori ฟิลเตอร์ จะมีค่าเท่ากัน อัตราส่วนของสัญญาณที่เรียกว่า misadjustment ที่นิยมใช้เป็นตัววัดคุณภาพของ อะแดปทีฟฟิลเตอร์ สามารถเขียนได้ดังสมการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

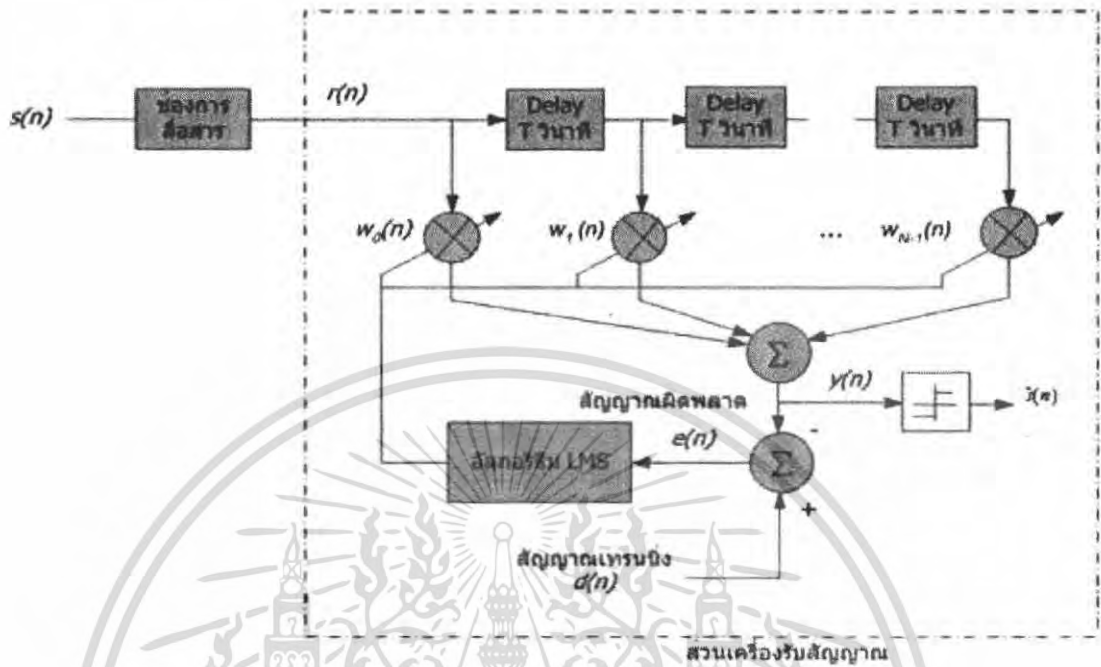
$$M(n) \equiv \frac{P_s(n)}{P_o(n)} \text{ หรือ } M'(n) \equiv \frac{P'_s(n)}{P'_o(n)} \quad (3-38)$$

### 3.8 Least Mean Square Algorithm

เป็นวิธีการหนึ่งที่พิจารณาถึงการหาค่าสัมประสิทธิ์ของ Adaptive Filter โดยอาศัยค่าเกรเดียนต์ เพื่อมาทำการปรับค่าสัมประสิทธิ์ของ Adaptive Filter เพื่อให้ได้ค่าความผิดพลาดระหว่างสัญญาณจริงกับ สัญญาณจาก Adaptive Filter น้อยที่สุด

เพื่อให้เห็นภาพของการใช้งานตัวปรับแต่งช่องสัญญาณแบบปรับตัวเองได้ เราจะทดลองอัลกอริทึมแบบง่ายๆ ด้วย MATLAB โดยอาศัยอัลกอริทึมที่เรียกว่า อัลกอริทึมค่าเฉลี่ยกำลังสองน้อยที่สุด (Least mean square) หรือ LMS อัลกอริทึมนี้คิดค้นโดย Prof. B. Widrow (ผู้ประดิษฐ์ไมโครโฟนอาร์เรย์สำหรับเครื่องช่วยฟัง (Hearing aids) ที่กล่าวไปในตอนที่แล้ว) และ นักศึกษาปริญญาเอก M. Hoff จาก Stanford University ในปี 1959 โดย ได้ทดลองใช้อุปกรณ์ hardware ในการสร้างวงจรนี้ขึ้นมา ซึ่งจุดประสงค์ ก็เพื่อหาอัลกอริทึมในการหาค่าตอบของค่าต่ำสุดของพื้นผิวของฟังก์ชันทางคณิตศาสตร์ ซึ่งอัลกอริทึม LMS นี้ นับได้ว่าเป็น อัลกอริทึมแบบปรับตัวได้ ที่แพร่หลายที่สุดสำหรับ วงจรกรองคิจิตอลแบบปรับตัวได้

หลักการที่สำคัญของ LMS ก็คือ การปรับค่าของสัมประสิทธิ์ของวงจรกรอง ซึ่งทำหน้าที่เป็นตัวปรับแต่งช่องสัญญาณ ให้ไปในทิศทางที่ทำให้ค่าเฉลี่ยกลางของค่าผิดพลาดยกกำลังสองมีค่าน้อยที่สุด ค่าผิดพลาดนี้หาได้จาก  $e^2(n) = E\{(d(n) - y(n))^2\}$  โดย ที่  $E\{\cdot\}$  นั้นเป็น ค่าทางสถิติที่มีความหมายถึง “ค่าเฉลี่ยกลาง” หรือ Mean value เราอาจจะให้คำจำกัดความอย่างง่ายๆ ของค่า  $e(n)$  นี้ว่า เป็น “ค่าผิดพลาดโดยเฉลี่ย” ของสัญญาณเทรนนิ่ง  $d(n)$  กับสัญญาณที่ได้มาจากเอาต์พุตของตัวปรับแต่งช่องสัญญาณ  $y(n)$  ในทางปฏิบัติจะทำการปรับค่า (Update) ของค่าสัมประสิทธิ์ โดยอาศัยการคำนวณ  $e(n)$  ตามเวลาจริง ที่ได้มาจากค่า  $d(n)$  และ  $y(n)$  ณ เวลา  $n$  ใดๆ ที่แสดงในรูปแบบที่ 3.10 แสดงถึงระบบปรับแต่งช่องสัญญาณที่เครื่องรับสัญญาณ โดยใช้ อัลกอริทึม LMS



รูปที่ 3.10 การจำลองการทำงาน ของระบบปรับแต่งช่องสัญญาณ

### 3.9 LMS – Based Algorithms

Adaptive Filters มีอัลกอริทึมที่อยู่หลายชนิดที่แตกแขนงออกมาจากคอนเวนชันนอลแอลเอมเอสอัลกอริทึม (Conventional LMS Algorithm) ซึ่งคอนเวนชันนอลแอลเอมเอสอัลกอริทึมนี้เป็นแอลเอมเอสแบบพื้นฐานที่มีอัลกอริทึมพื้นฐานดังตารางนี้

Conventional LMS Algorithm
<p><b>Initialization</b></p> $x(0) = w(0) = [0 \ 0 \ \dots \ 0]^T$ <p>Do for <math>k \geq 0</math></p> $e(k) = d(k) - x^T(k)w(k)$ $w(k+1) = w(k) + 2\mu e(k)x(k)$

ตารางที่ 3.1 แสดงตาราง Conventional LMS Algorithm ที่สมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยจุดประสงค์หลักของการศึกษาอัลกอริทึมชนิดอื่นนั้นเพื่อลดความซับซ้อนของการคำนวณ (Computational Complexity) และเพื่อลดเวลาความเร็วในการลู่เข้าลง (Convergence time) โดยในส่วนนี้จะนำเสนอรวมทั้งการวิเคราะห์ LMS – Based Algorithms แบบต่างๆ ซึ่งในบทที่ 4 จะเป็นการแสดงเปรียบเทียบผลการ simulation LMS – Based Algorithms ในแบบต่างๆ 4 แบบ ดังนี้

### 3.9.1 The LMS – Newton Algorithm

จุดประสงค์หลักของอัลกอริทึมชนิดนี้คือ หลีกเลี่ยงการลู่เข้าของสัญญาณที่ช้าลงของแอลเอ็มเอสอัลกอริทึม เมื่อสัญญาณอินพุตถูกทำให้มีความสัมพันธ์กันของข้อมูลที่สูงมาก โดยความเร็วในการลู่เข้าของสัญญาณจะมากขึ้นก็ต่อเมื่อมีการเพิ่มขึ้นของความซับซ้อนในการคำนวณให้มากขึ้น โดยความเร็วที่เพิ่มขึ้นนี้มาจากการใช้ค่าประมาณของส่วนกลับของสัญญาณอินพุตที่มีความสัมพันธ์ในเชิงเมตริก ที่จะนำไปสู่ความซับซ้อนของการคำนวณที่มากขึ้น

<b>LMS – Newton Algorithm</b>
<p><b>Initialization</b></p> $\mathbb{R}^{-1}(-1) = \delta I \quad (\delta \text{ a small positive constant})$ $w(0) = x(-1) = [0 \dots 0]^T$ <p>Do for <math>k \geq 0</math></p> $e(k) = d(k) - x^T(k)w(k)$ $\mathbb{R}^{-1}(k) = \frac{1}{1-\alpha} \left[ \mathbb{R}^{-1}(k-1) - \frac{\mathbb{R}^{-1}(k-1)x(k)x^T(k)\mathbb{R}^{-1}(k-1)}{\frac{1}{1-\alpha} + x^T(k)\mathbb{R}^{-1}(k-1)x(k)} \right]$ $w(k+1) = w(k) + 2\mu e(k)\mathbb{R}^{-1}(k)x(k)$

ตารางที่ 3.2 แสดงตาราง LMS – Newton Algorithm ที่สมบูรณ์

### 3.9.2 The Normalized LMS Algorithm (NLMS)

วิธีการ Normalized LMS (NLMS) เป็นวิธีการที่ดัดแปลงมาจากระเบียบวิธี LMS โดยนำเอาสัญญาณที่เปลี่ยนแปลงตามเวลาที่อินพุตของวงจรกรอง และทำการปรับค่า Normalized ของค่าพารามิเตอร์ช่วงก้าว ซึ่งจะให้ค่าเสถียรและการลู่เข้าของการปรับตัวจะให้ผลที่ดี ถ้าจะมีเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัลกอริทึมตัวใดตัวหนึ่งที่เพิ่มความเร็วในการลู่เข้าได้โดยไม่ต้องใช้ค่าประมาณของสัญญาณที่มีความสัมพันธ์ในเชิงเมตริก NLMS ก็คืออัลกอริทึมตัวนั้น โดยปกติแล้ว NLMS Algorithm จะมีความเร็วในการลู่เข้าที่มากกว่า LMS Algorithm แบบทั่วไป โดยการใช้ประโยชน์จากปัจจัยต่างๆ ของการลู่เข้าซึ่งมีจุดมุ่งหมายอยู่ที่การลดค่าความผิดพลาดที่เกิดขึ้นทางเอาท์พุทได้แบบทันที

<b>The Normalized LMS Algorithm</b>
<p><b>Initialization</b></p> <p><math>x(0) = \hat{w}(0) = [0 \dots 0]^T</math></p> <p>choose <math>\mu_n</math> in the range <math>0 &lt; \mu_n \leq 2</math></p> <p><math>\gamma = \text{small constant}</math></p> <p>Do for <math>k \geq 0</math></p> <p><math>e(k) = d(k) - x^T(k)w(k)</math></p> <p><math>w(k+1) = w(k) + \frac{\mu_n}{\gamma + x^T(k)x(k)} e(k)x(k)</math></p>

ตารางที่ 3.3 แสดงตาราง The Normalized LMS Algorithm ที่สมบูรณ์

### 3.9.3 The Affine Projection Algorithm

แนวคิดของ The Affine Projection Algorithm คือเมื่อมีสถานการณ์ที่มีความเป็นไปได้ว่าจะต้องนำข้อมูลสัญญาณเก่ากลับมาใช้เพื่อทำให้การลู่เข้าของสัญญาณดียิ่งขึ้น ข้อมูลอัลกอริทึมที่นำกลับมาใช้ใหม่เป็นทางเลือกแรกที่จะนำมาทำให้ความเร็วในการลู่เข้าของสัญญาณเพิ่มขึ้น

<b>The Affine Projection Algorithm</b>
<p><b>Initialization</b></p> $x(0) = w(0) = [0 \dots 0]^T$ <p>choose <math>\mu</math> in the range <math>0 &lt; \mu \leq 2</math></p> <p><math>\gamma =</math> small constant</p> <p>Do for <math>k \geq 0</math></p> $e_{ap}(k) = d_{ap}(k) - X_{ap}^T(k)w(k)$ $w(k+1) = w(k) + \mu X_{ap}(k) \left( X_{ap}^T(k)X_{ap}(k) + \gamma I \right)^{-1} e_{ap}(k)$

ตารางที่ 3.4 แสดง The Affine Projection Algorithm ที่สมบูรณ์

### 3.9.4 The Partially Filter Gradient LMS Algorithm (PFGLMS)

ถ้า LMS Algorithm มีการลู่เข้าที่ช้าและไม่สามารถเดินไปในทิศทางที่เสถียรแล้ว PFGLMS Algorithm จะถูกนำมาใช้เพื่อทำให้ค่าความเร็วในการลู่เข้าของสัญญาณคีย์ขึ้นของ LMS Algorithm ด้วยการเพิ่มความซับซ้อนในการคำนวณให้มากขึ้น LMS Algorithm แบบพื้นฐานหรือ Conventional LMS Algorithm ต้องใช้สมการ  $2m+1$  เป็นตัวคูณและใช้ข้อมูลเหมือนกับข้อมูลของภาครับที่ถูกเพิ่มเข้าไปในแต่ละซิมโบล แต่ PFGLMS Algorithm ต้องใช้สมการ  $4m+1$  เป็นตัวคูณและใช้ข้อมูลเหมือนกับข้อมูลทางภาครับที่ถูกเพิ่มเข้าไปในแต่ละซิมโบล ดังนั้นความซับซ้อนในการคำนวณคือค่าประมาณ ในออเดอร์  $m$  ในทั้ง LMS และ PFGLMS Algorithm

<b>The Partially Filter Gradient LMS Algorithm</b>
<p><b>Initialization</b></p> $w(t+1) = w(t) + 2\mu g(t)$ $g(t) = e\lambda \cdot xtdl + \hat{g}(t)$ $\hat{g}(t) = \lambda \hat{g}(t-1) + ye \cdot xtdl$ $\hat{g}(t-1) = [0 \ 0 \ 0 \ 0]_5$

ตารางที่ 3.5 แสดง The Partially Filter Gradient LMS Algorithm ที่สมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.10 Simulation Example

โดยเราจะสมมุติระบบการสื่อสารทางดิจิทัล โดยกำหนดให้สัญญาณที่จะส่งเป็นสัญญาณ BPSK โดยจะทำการส่งในช่องสัญญาณ AWGN เท่านั้น ดังนั้นที่ภาครับจะรับสัญญาณได้ดังนี้

$$r = x + n$$

เมื่อ  $r$  คือ สัญญาณที่ภาครับสามารถรับได้

$x$  คือ สัญญาณ BPSK ที่ส่ง

$n$  คือ สัญญาณรบกวนแบบ AWGN

นำสัญญาณ  $y$  ไปทำการ detect สัญญาณ โดยใช้วงจร Equalizer แบบ Adaptive Equalizer บนพื้นฐานของ LMS อัลกอริทึม โดยมีสมการดังนี้

$$Y(n) = W^T(n) \cdot r(n)$$

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

และ

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

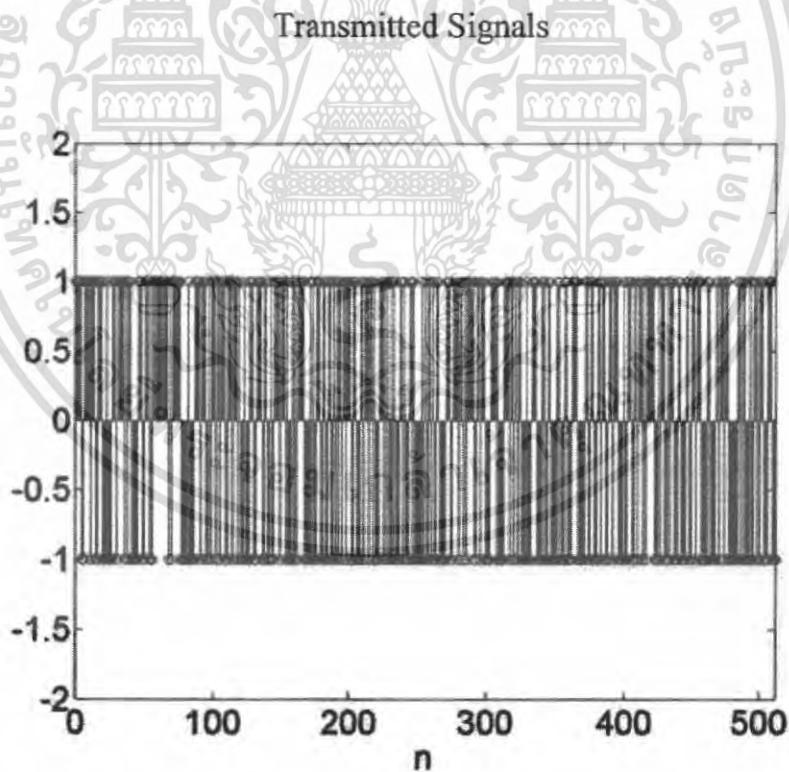
รูป 3.15 แสดงถึง convergence speed ของ LMS อัลกอริทึม ที่ใช้ในการ detect สัญญาณ และจากผลการทดลองที่ได้ แสดงรายละเอียดต่างๆดังต่อไปนี้

ในรูปที่ 3.11 ก) นั้นแสดงรูปแบบของ สัญญาณเทรนนิ่ง  $d(n)$  ที่เป็นสัญญาณดิจิทัล  $\pm 1$  และเพื่อให้เห็นลักษณะสัญญาณได้ชัดเจน ได้แสดงส่วนขยายสำหรับสัญญาณลักษณะตั้งแต่ 488 ถึง 512 ไว์ใน รูปที่ 3.11 ข) สัญญาณ  $d(n)$  ที่จริงแล้วก็คือสัญญาณ  $s(n)$  ที่ถูกส่งผ่านช่องสัญญาณ เพื่อใช้ใน ช่วงเวลาการปรับแต่งช่องสัญญาณ ข้อมูล  $d(n)$  นี้ซึ่งมีขนาด 26 บิต จะถูกส่งมาเป็นช่วงกลางในทุกๆ เฟรมข้อมูลปกติ แต่สำหรับในการจำลองแบบนี้ เราถือว่าข้อมูล  $d(n)$  เป็นแบบต่อเนื่อง เพื่อให้เห็น ลักษณะการปรับตัวของตัวปรับแต่งช่องสัญญาณ ได้อย่างชัดเจน สัญญาณที่รับได้ที่เครื่องรับจะเป็น สัญญาณ  $r(n)$  ที่แสดง ในรูปที่ 3.12 ที่สังเกตเห็นได้ชัดเจนว่ามีความผิดเพี้ยนไปจากสัญญาณ  $d(n)$  มาก โดยหากไม่มีการใช้ตัวปรับแต่งช่องสัญญาณที่เครื่องรับ ก็จะทำให้การรับสัญญาณนั้น มีอัตรา

บิตผิดพลาดสูงมากเกินระดับที่ยอมรับได้ แต่เมื่อเราใช้ตัวปรับแต่งช่องสัญญาณแล้ว จะเห็นได้ว่าช่องสัญญาณ จะค่อยๆ ถูกปรับแต่ง โดยในช่วงสั้นๆ ของการปรับแต่งช่องสัญญาณ สัญญาณ  $y(n)$  ยังมีความผิดพลาดสูงมาก ดังแสดงในรูปที่ 3.13 ก) แต่เมื่อการปรับแต่งเริ่มเข้าสู่สภาวะวะถู่เข้าแล้ว จะเห็นว่าสัญญาณ  $y(n)$  ที่มีความผิดพลาดต่ำลงมาก ซึ่งจะเปรียบเทียบได้จาก สัญญาณ  $y(n)$  ที่แสดงในรูปที่ 3.13 ข) กับ สัญญาณ  $d(n)$  ที่แสดงในรูปที่ 3.1ข)

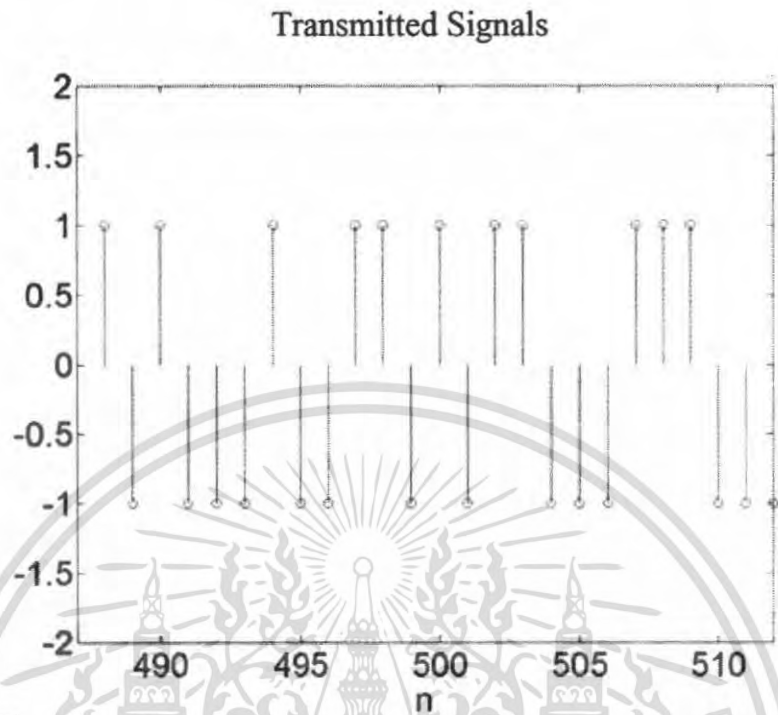
ในรูปที่ 3.14 แสดง ผลตอบสนองความถี่ของช่องสัญญาณ, ส่วนปรับแต่งช่องสัญญาณ และ ผลลัพธ์ ซึ่งเป็นผลตอบสนองรวม ของระบบซึ่งจะเห็นว่าได้ ผลตอบสนองความถี่ที่ราบเรียบ ซึ่งเป็นผลมาจากการใช้ตัวปรับแต่งช่องสัญญาณที่ให้ผลตอบสนองเป็นส่วนกลับของช่องสัญญาณ

รูปที่ 3.15 แสดงค่าความผิดพลาดยกกำลังสอง ในหน่วย dB ซึ่งจะเห็นว่าในช่วงแรกๆ ของการปรับตัว ค่าความผิดพลาดมีค่าสูงมาก จนกระทั่งเมื่อ เวลาผ่านไปค่าความผิดพลาดนี้จะลดลงเป็นอย่างมาก และเข้าสู่สภาวะคงตัวเมื่อประมาณสัญญาณลักษณะ ที่ 300

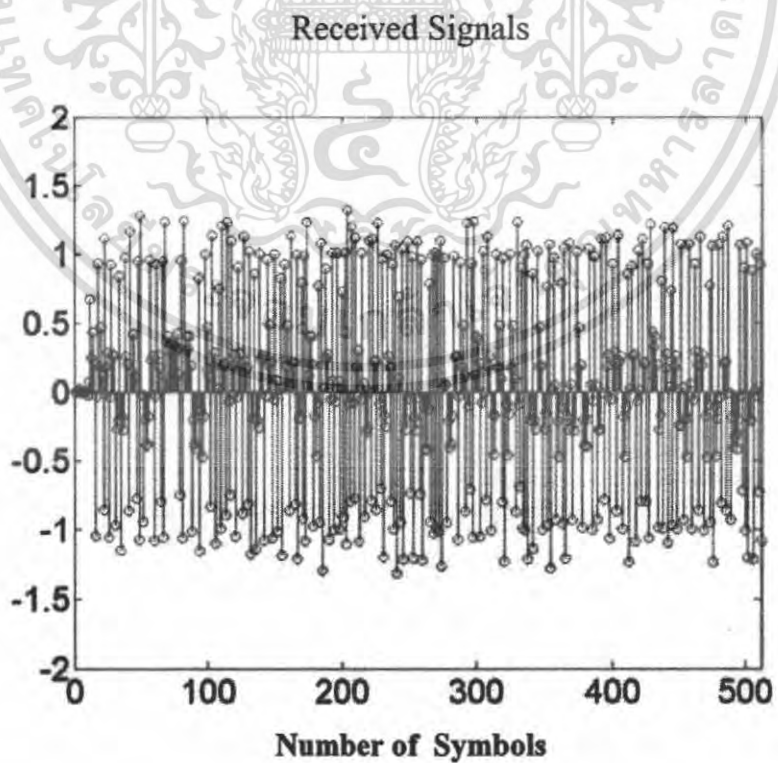


รูปที่ 3.11 ก) สัญญาณเทรนนิ่ง  $d(n)$  เป็นสัญญาณดิจิทัล  $\pm 1$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



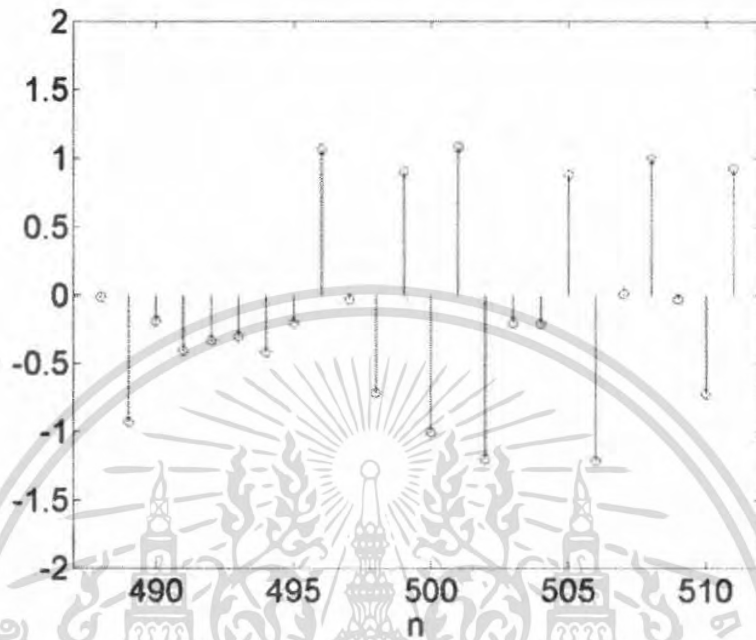
รูปที่ 3.11 ข) ส่วนขยายของสัญญาณ ที่ สัญญาณลักษณะที่ 488 ถึง 512



รูปที่ 3.12 ก) สัญญาณที่รับได้  $r(n)$  ที่เครื่องรับ ที่เกิดการผิดเพี้ยน

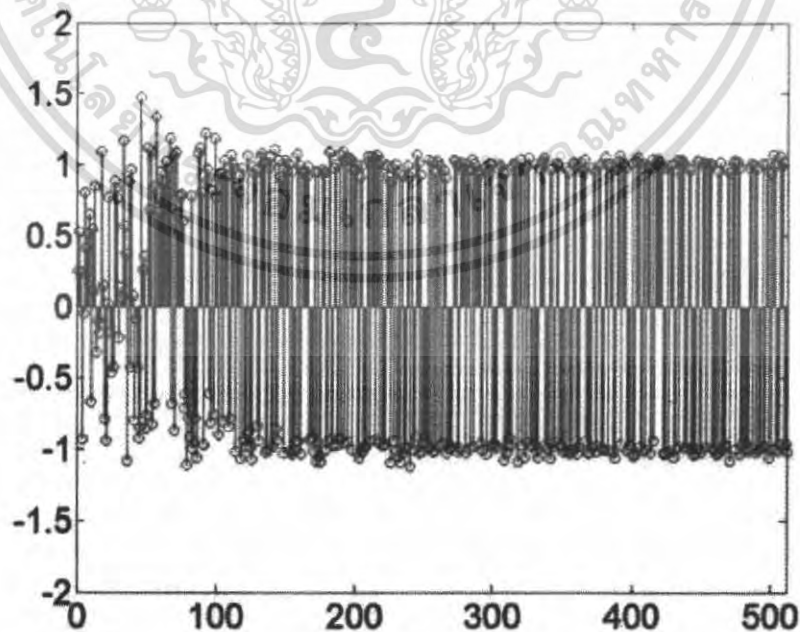
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### Received Signals



รูปที่ 3.12 ข) ส่วนขยายของสัญญาณ  $r(n)$  ที่สัญญาณลักษณะที่ 488 ถึง 512

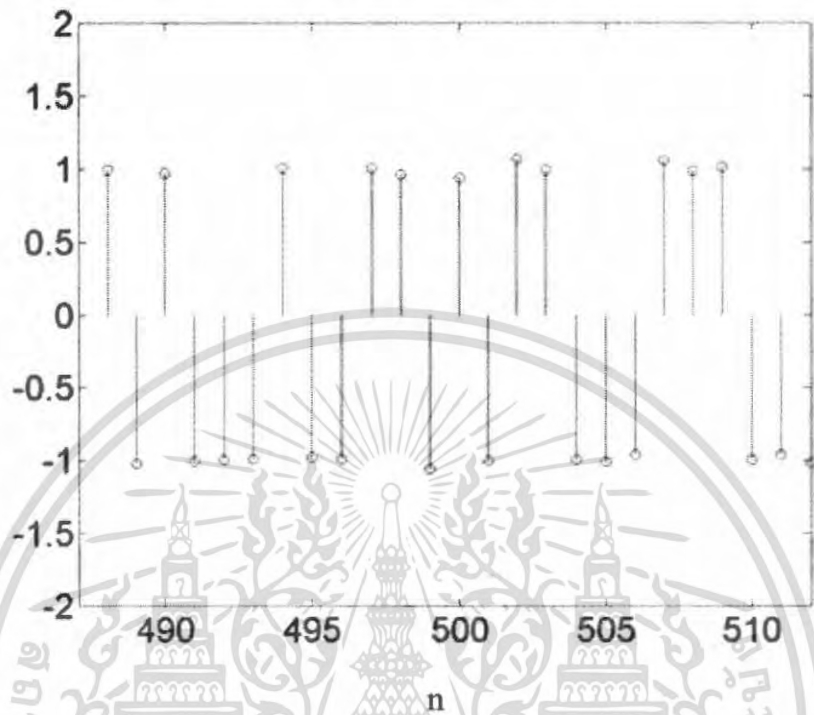
### Equalised Signals



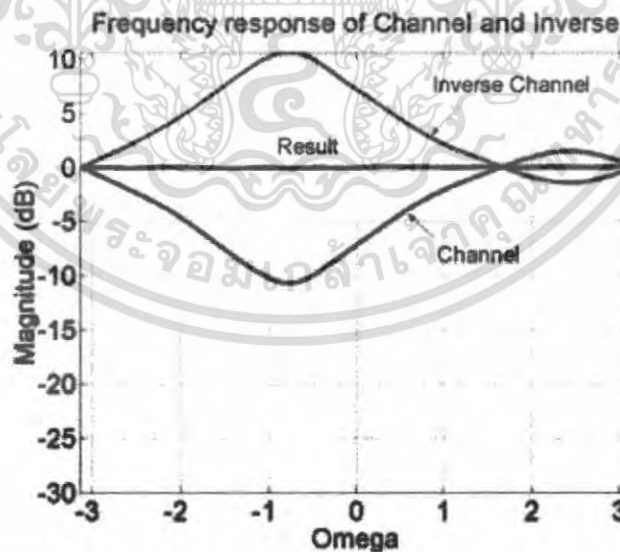
รูปที่ 3.13 ก) สัญญาณ  $x(n)$  ที่ผ่านการปรับแต่งด้วยตัวปรับสัญญาณแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Equalised Signals

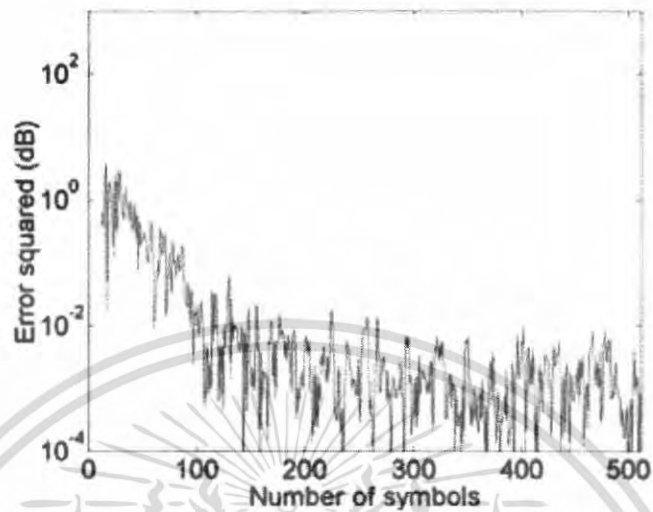


รูปที่ 3.13 ข) ส่วนขยายของสัญญาณ ที่ สัญญลักษณ์ที่ 488 ถึง 512



รูปที่ 3.14 แสดงผลตอบสนองความถี่รวมของช่องสัญญาณ โดยส่วนอินเวอร์สช่องสัญญาณจะเป็นผลตอบสนองของตัวปรับแต่งช่องสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 ค่าผิดพลาดยกกำลังสอง  $e^2(n)$  มีค่าลดลงตามการสุ่มซ้ำของตัวปรับตัวแต่งสัญญาณและคงตัวเมื่อจำนวนค่าสัญญาณถึงในระดับ 300 สัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

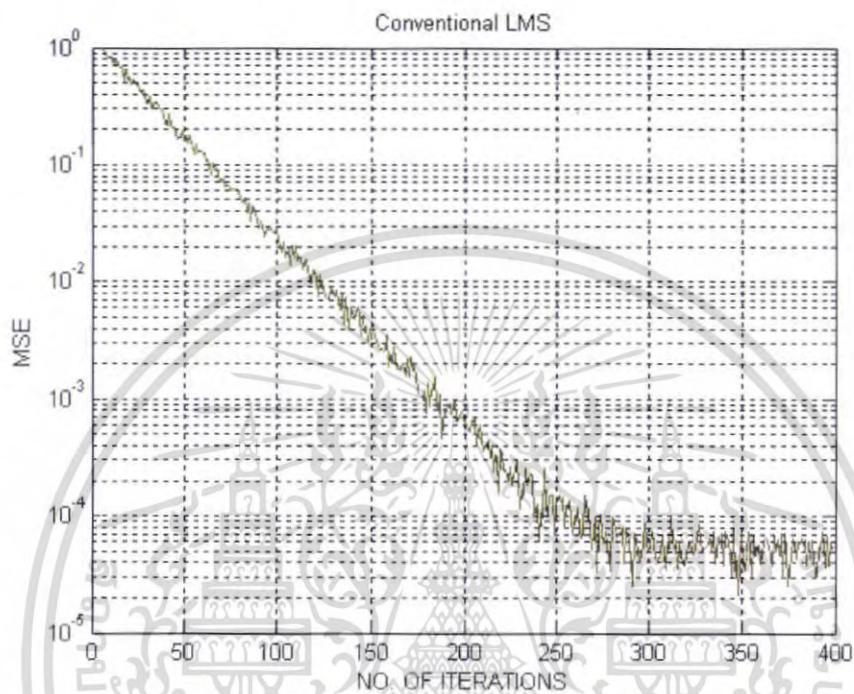
### ผลการทดลอง

ในบทนี้จะเป็นผลการทดลองที่ทำการเปรียบเทียบกันของแอลเอ็มเอสอัลกอริทึมทั้งหมด 4 อัลกอริทึมคือ Conventional LMS Algorithm, The LMS – Newton Algorithm, The Normalized LMS Algorithm และ The Partially Filter Gradient LMS Algorithm โดยใช้โปรแกรมเมทแลปในการซิมูเลชัน เพื่อศึกษาและเปรียบเทียบการนำไปใช้ในตัวกรองปรับค่าได้แบบต่างๆ ในการทดลองนี้จะทำการเปรียบเทียบความเร็วในการลู่เข้าและหาค่าผิดพลาดเฉลี่ยยกกำลังสองเพื่อจะนำผลการทดลองนี้ ไปวิเคราะห์ประสิทธิภาพโดยรวมของระบบต่อไป

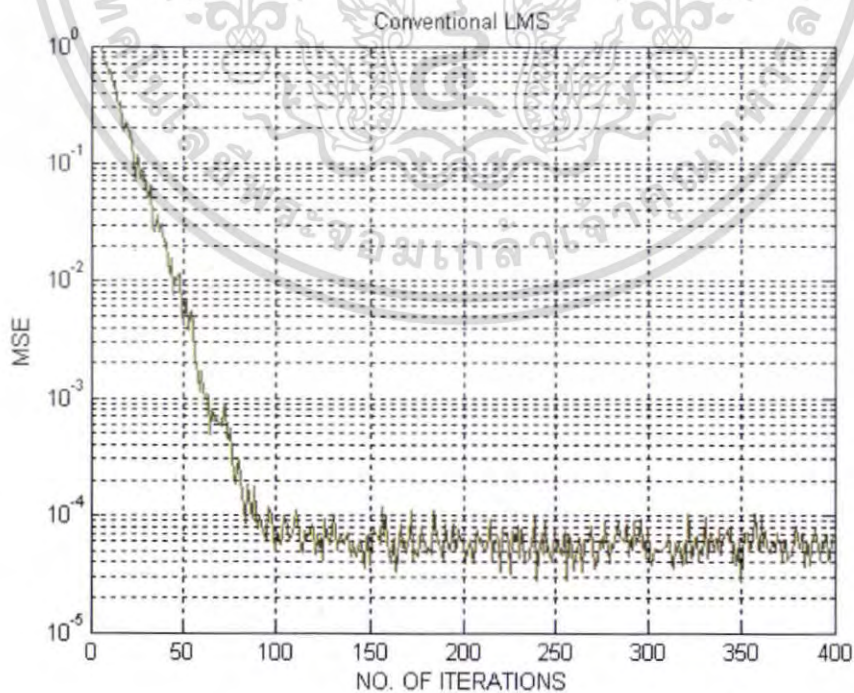
#### จุดประสงค์ของการทดลอง

1. ช่วยให้ผู้อ่านเริ่มต้นทำความเข้าใจได้อย่างรวดเร็วในการซิมูเลชันนี้
2. ตรวจสอบความแม่นยำถูกต้องของผลของทฤษฎีที่มีการพัฒนาขึ้น
3. เพิ่มความเข้าใจในผลของทฤษฎี โดยการสังเกตอย่างระมัดระวัง และอธิบายผลของการซิมูเลชัน
4. เปรียบเทียบความเร็วในการลู่เข้าของ LMS แบบดั้งเดิมกับ LMS ที่มีการพัฒนาขึ้น
5. เปรียบเทียบค่าความผิดพลาดเฉลี่ยยกกำลังสองของ LMS แบบต่างๆ
6. วิเคราะห์ประสิทธิภาพรวมของ LMS แบบต่างๆก่อนที่จะนำไปประยุกต์ใช้งานต่อไป

#### 4.1 Conventional LMS Algorithm

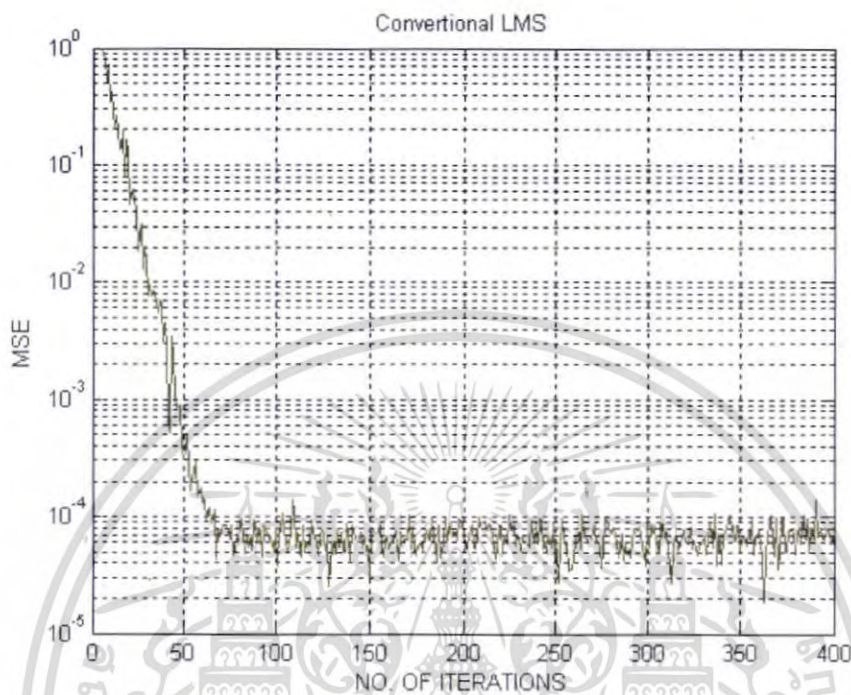


รูปที่ 4.1 ความเร็วในการลู่เข้าของ Conventional LMS ที่ปรับค่า step size ( $\mu$ ) = 0.01

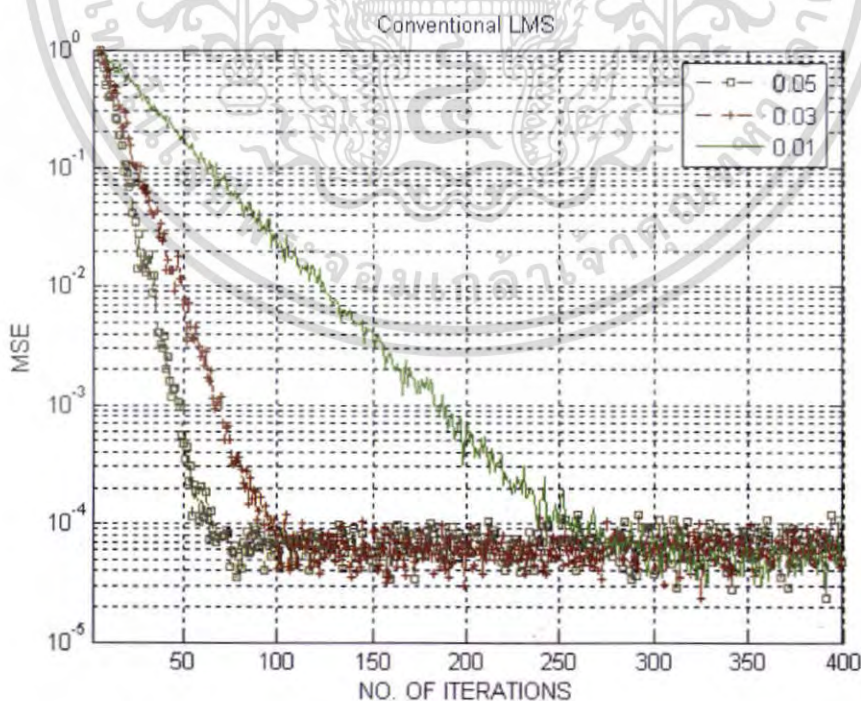


รูปที่ 4.2 ความเร็วในการลู่เข้าของ Conventional LMS ที่ปรับค่า step size ( $\mu$ ) = 0.03

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



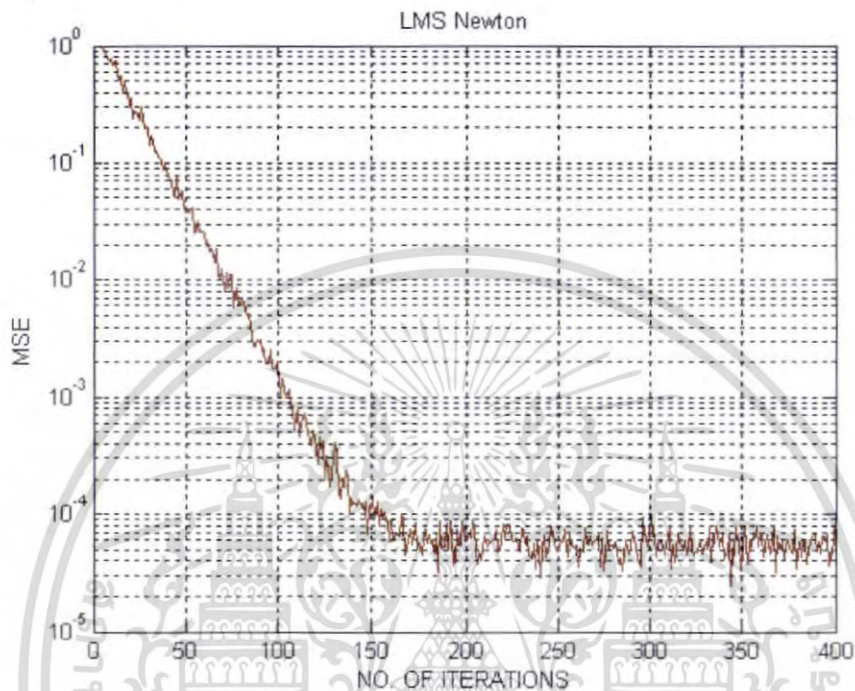
รูปที่ 4.3 ความเร็วในการถ่วงน้ำหนักของ Conventional LMS ที่ปรับค่า step size ( $\mu$ ) = 0.05



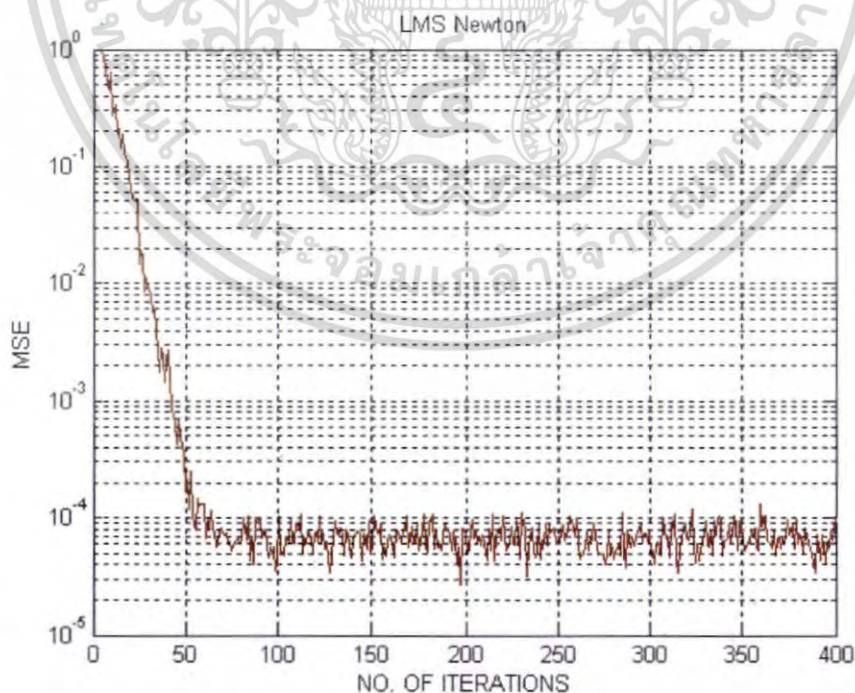
รูปที่ 4.4 เปรียบเทียบความเร็วในการถ่วงน้ำหนักของ Conventional LMS ที่  $\mu = 0.01, 0.03$  และ  $0.05$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.2 The LMS – Newton Algorithm

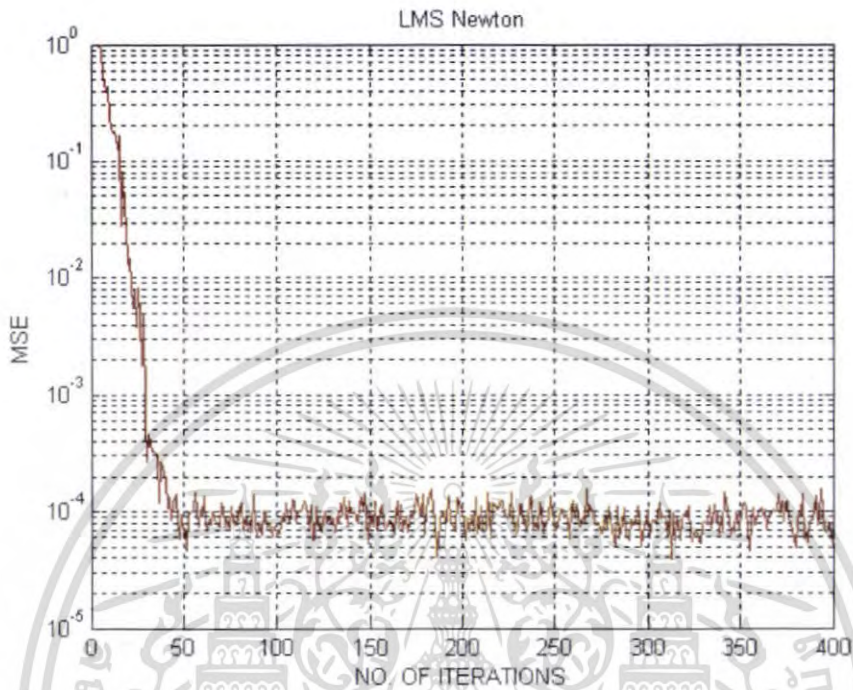


รูปที่ 4.5 ความเร็วในการลู่เข้าของ LMS - Newton ที่ปรับค่า step size ( $\mu$ ) = 0.01

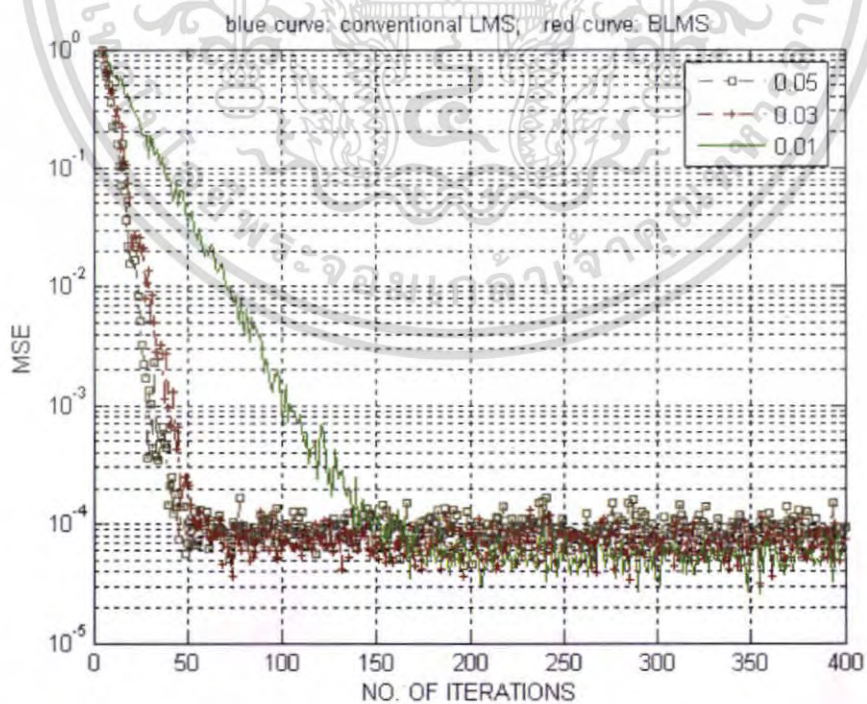


รูปที่ 4.6 ความเร็วในการลู่เข้าของ LMS - Newton ที่ปรับค่า step size ( $\mu$ ) = 0.03

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

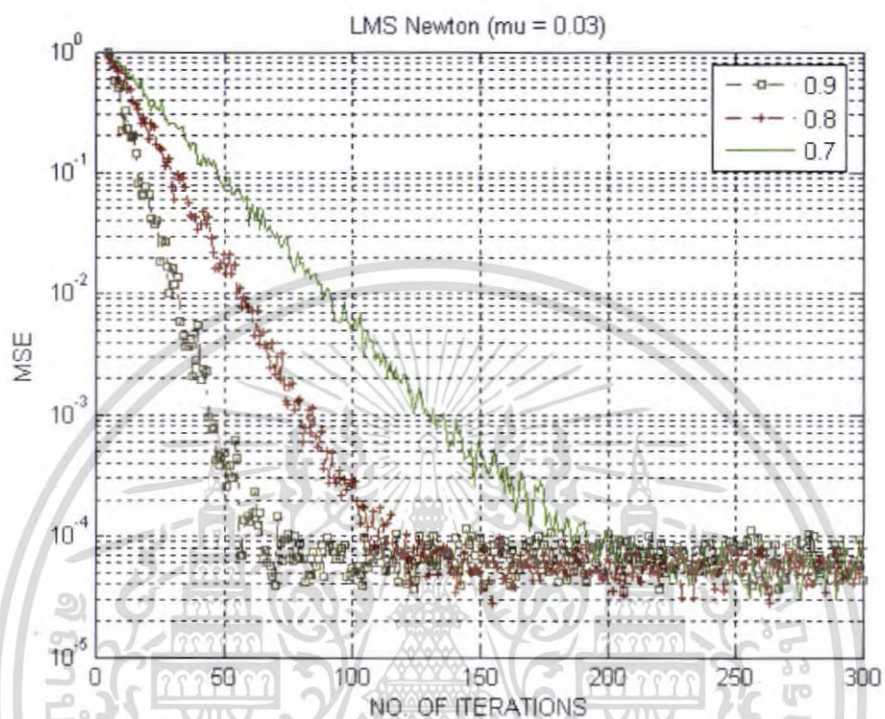


รูปที่ 4.7 ความเร็วในการลู่เข้าของ LMS - Newton ที่ปรับค่า step size ( $\mu$ ) = 0.05



รูปที่ 4.8 เปรียบเทียบความเร็วในการลู่เข้าของ LMS - Newton ที่  $\mu = 0.01, 0.03$  และ  $0.05$

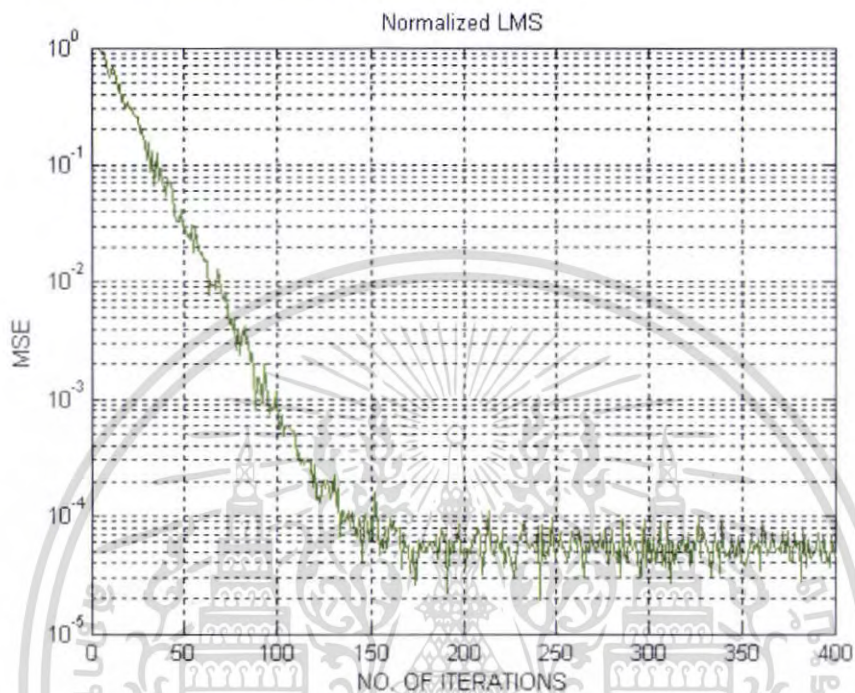
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



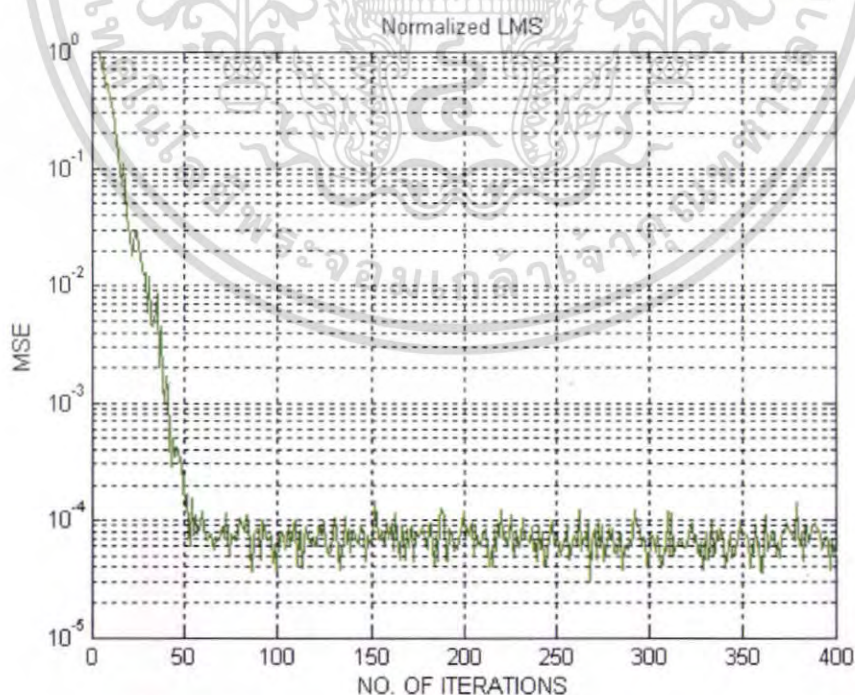
รูปที่ 4.9 เปรียบเทียบความเร็วในการลู่เข้าของ LMS - Newton ที่  $\mu = 0.03$  และปรับค่า  $\alpha (\alpha) = 0.7, 0.8, 0.9$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.3 The Normalized LMS Algorithm (NLMS)

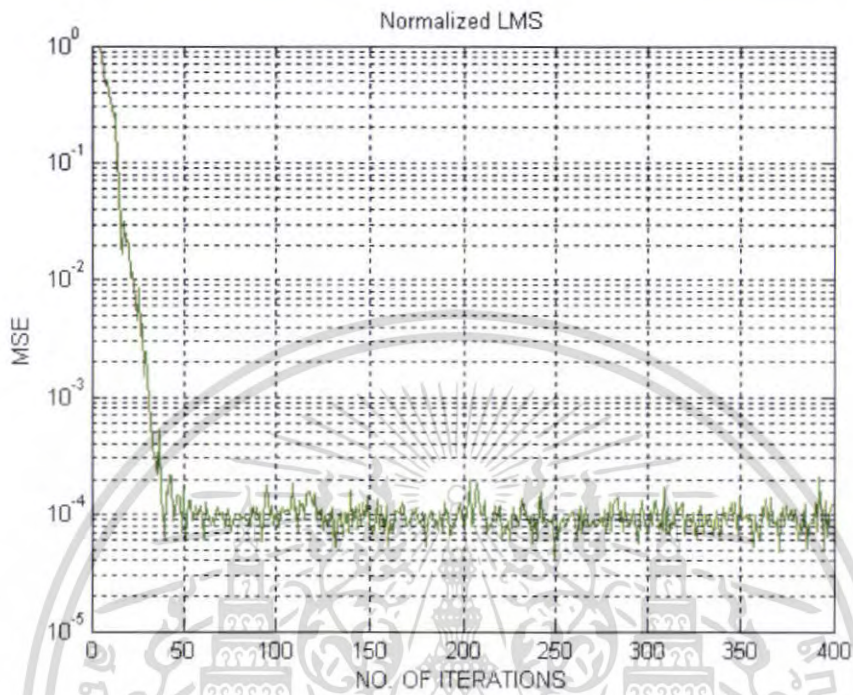


รูปที่ 4.10 ความเร็วในการลู่เข้าของ NLMS ที่ปรับค่า step size ( $\mu$ ) = 0.01

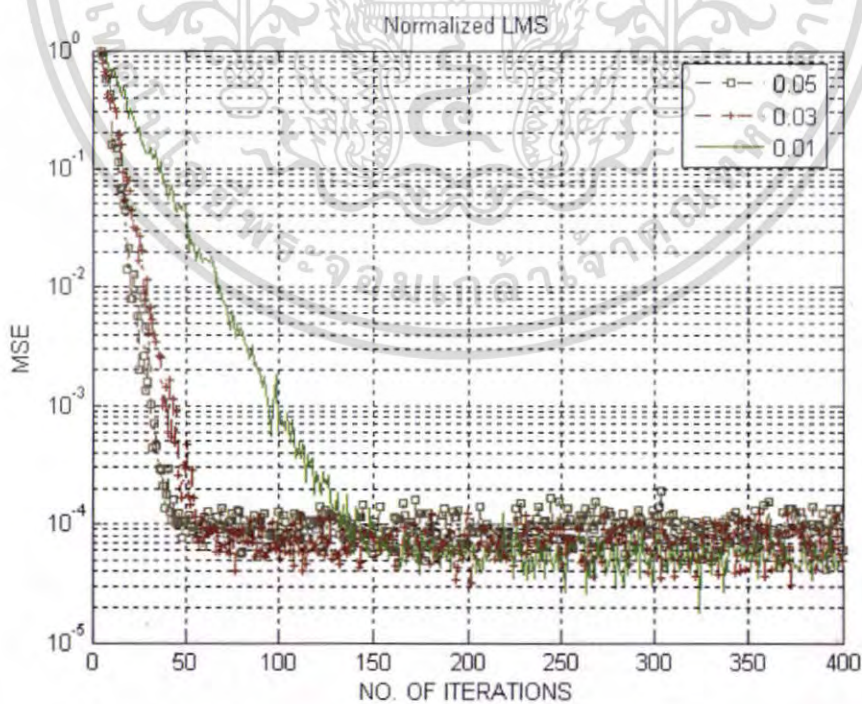


รูปที่ 4.11 ความเร็วในการลู่เข้าของ NLMS ที่ปรับค่า step size ( $\mu$ ) = 0.03

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

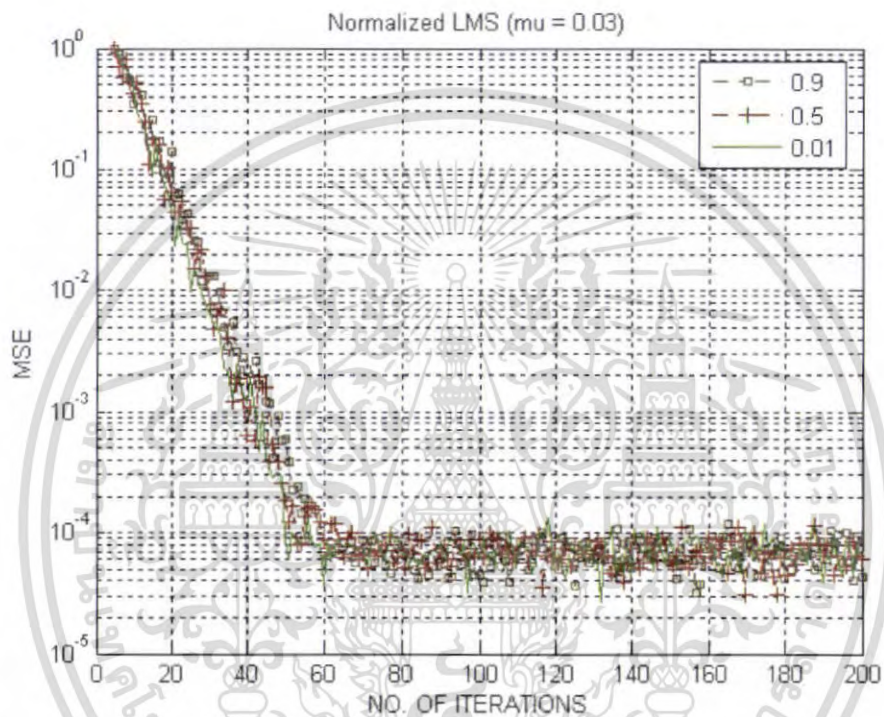


รูปที่ 4.12 ความเร็วในการลู่เข้าของ NLMS ที่ปรับค่า step size ( $\mu$ ) = 0.05



รูปที่ 4.13 เปรียบเทียบความเร็วในการลู่เข้าของ NLMS ที่  $\mu = 0.01, 0.03$  และ  $0.05$

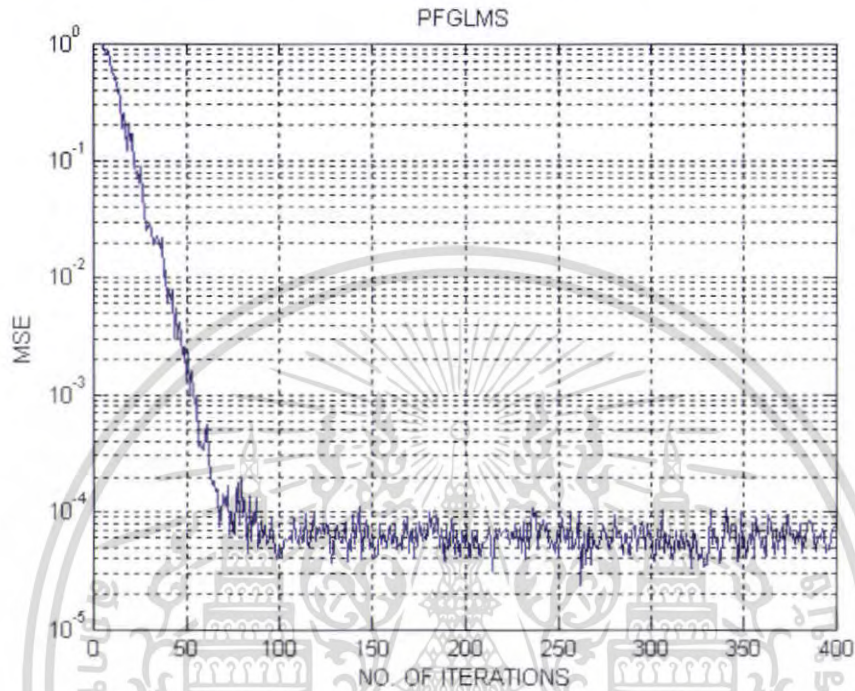
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



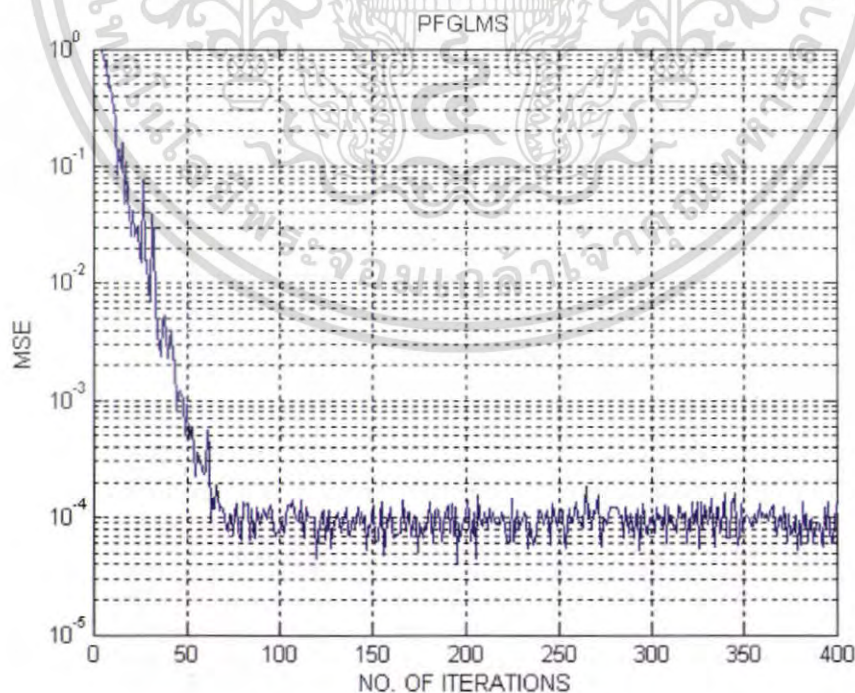
รูปที่ 4.14 ความเร็วในการลู่เข้าของ NLMS ที่ปรับค่า step size ( $\mu$ ) = 0.03 และปรับค่า  $\gamma$  ( $\gamma$ ) = 0.01, 0.5 และ 0.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.4 The Partially Filter Gradient LMS Algorithm (PFGLMS)

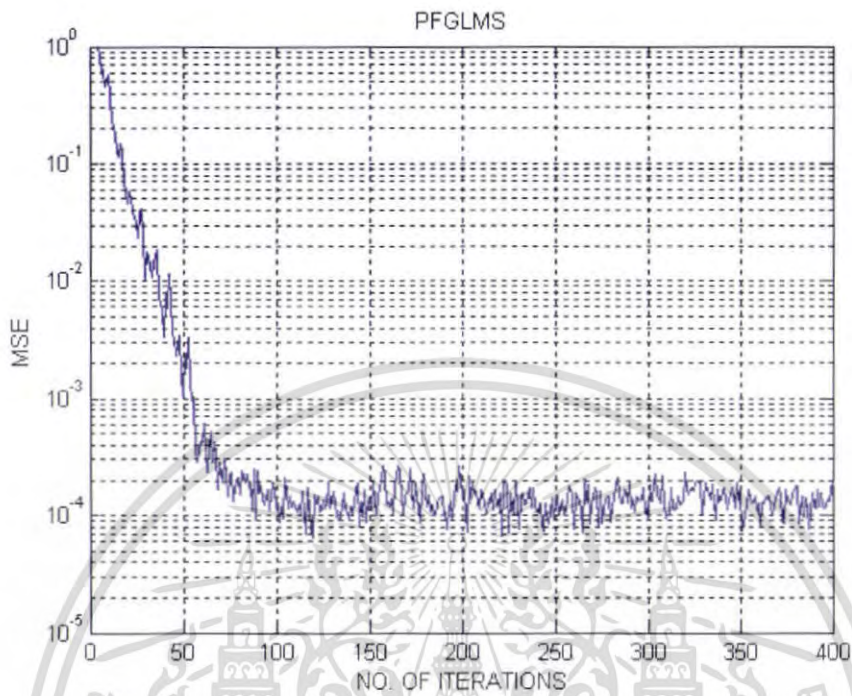


รูปที่ 4.15 ความเร็วในการลู่เข้าของ PFGLMS ที่ปรับค่า step size ( $\mu$ ) = 0.01

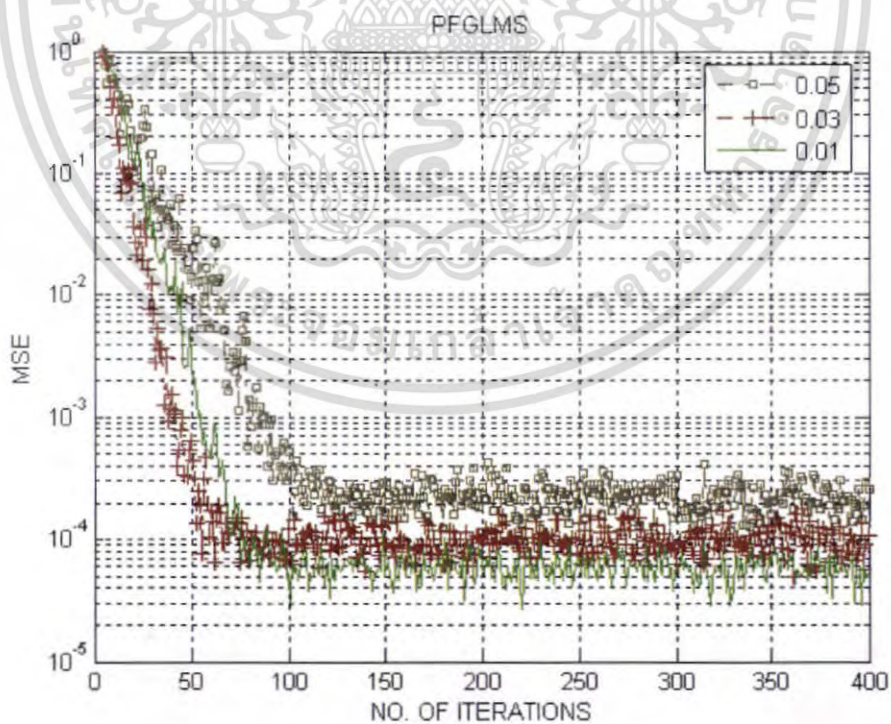


รูปที่ 4.16 ความเร็วในการลู่เข้าของ PFGLMS ที่ปรับค่า step size ( $\mu$ ) = 0.03

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.17 ความเร็วในการเข้าสู่ของ PFGLMS ที่ปรับค่า step size ( $\mu$ ) = 0.05

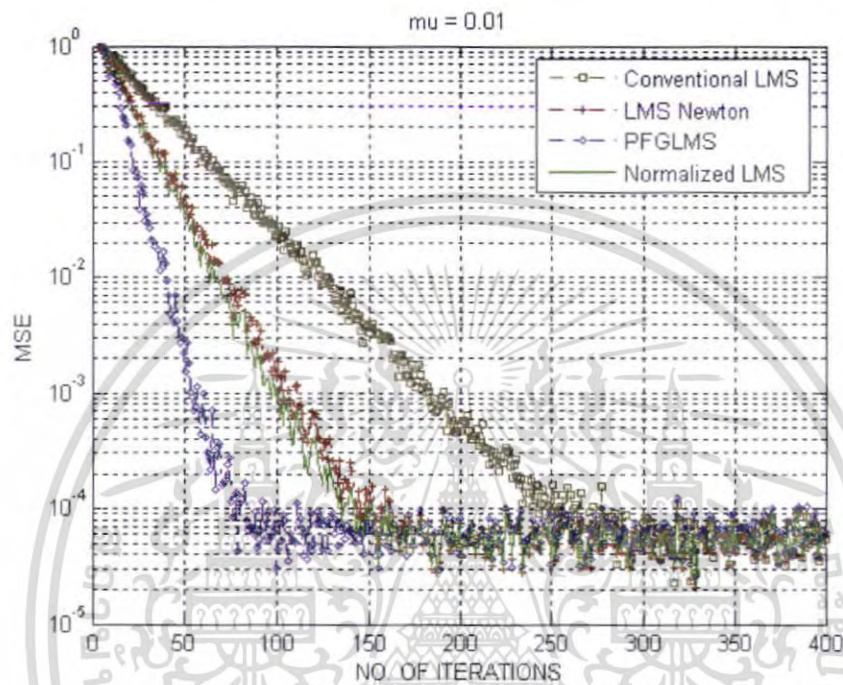


รูปที่ 4.18 เปรียบเทียบความเร็วในการเข้าสู่ของ PFGLMS ที่  $\mu = 0.01, 0.03$  และ  $0.05$

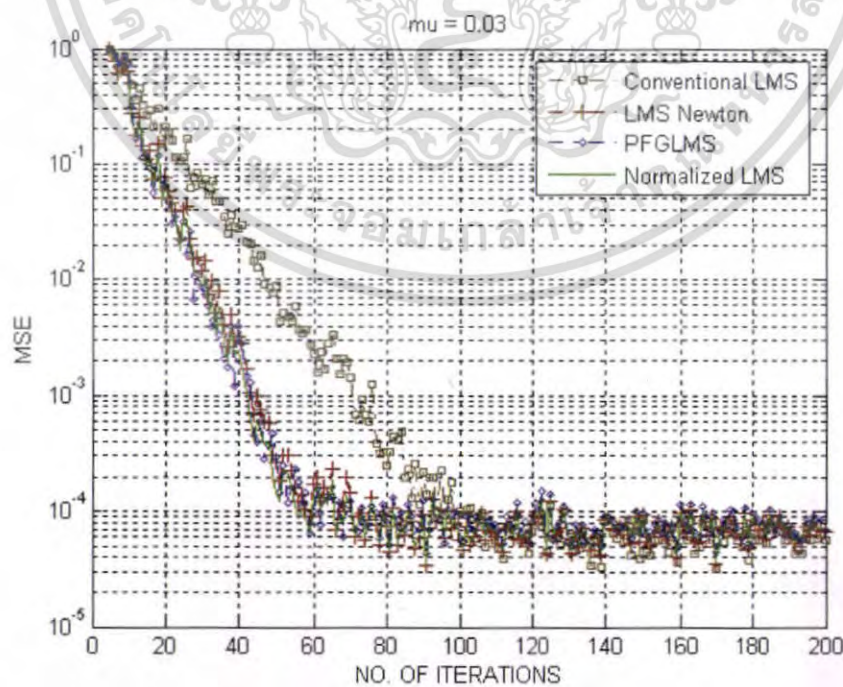
Lamda ( $\lambda$ ) = 0.7 และ Gamma ( $\gamma$ ) = 0.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.5 การเปรียบเทียบความเร็วในการเข้าสู่ของ LMS Algorithm ทั้ง 4 แบบ

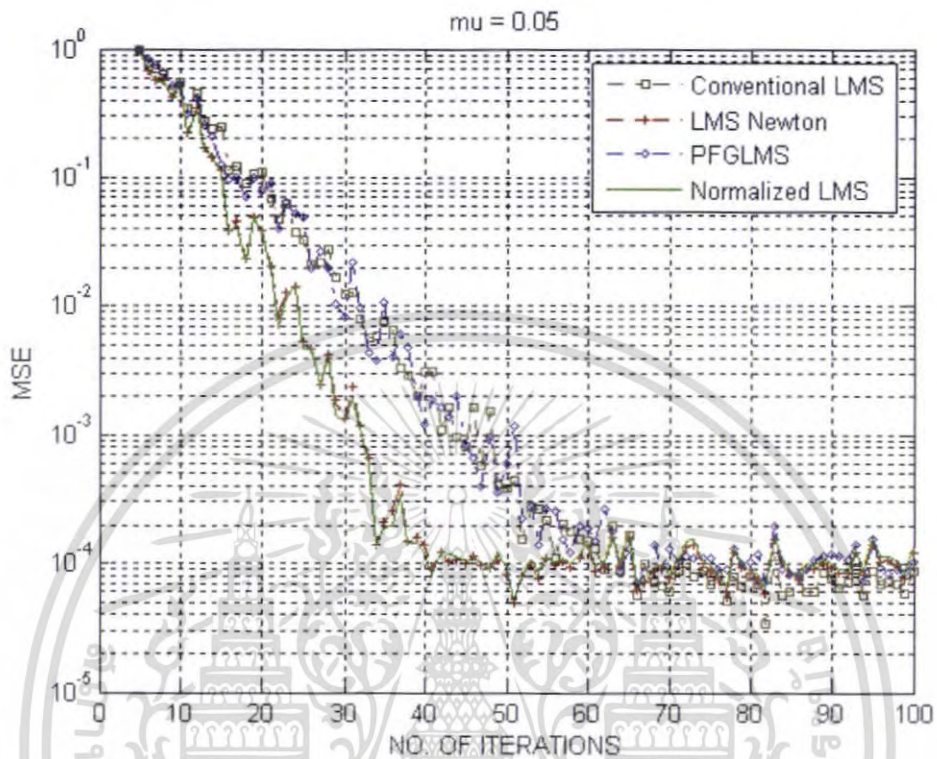


รูปที่ 4.19 เปรียบเทียบความเร็วในการเข้าสู่ทั้ง 4 แบบ โดยปรับค่า step size ( $\mu$ ) = 0.01



รูปที่ 4.20 เปรียบเทียบความเร็วในการเข้าสู่ทั้ง 4 แบบ โดยปรับค่า step size ( $\mu$ ) = 0.03

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.21 เปรียบเทียบความเร็วในการลู่เข้าทั้ง 4 แบบ โดยปรับค่า step size ( $\mu$ ) = 0.05

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.6 การเปรียบเทียบความซับซ้อนของระบบ

ในส่วนของความซับซ้อนของระบบนั้น เราสามารถหาได้จากจำนวนตัวคูณที่เพิ่มเข้าไปในแต่ละอัลกอริทึม ยิ่งอัลกอริทึมที่มีตัวคูณมากก็จะยิ่งมีความซับซ้อนมากตามไปด้วย โดยเราสามารถเปรียบเทียบความซับซ้อนของแต่ละอัลกอริทึมได้ดังนี้

<p><b>Conventional LMS</b></p> $2N + 1$ <p>โดย <math>N = 5</math> ดังนั้นความซับซ้อนของระบบคือ</p> $11$	<p><b>NLMS</b></p> $4N + 1$ <p>โดย <math>N = 5</math> ดังนั้นความซับซ้อนของระบบคือ</p> $21$
<p><b>LMS – Newton</b></p> $N^3 + 4N + 2$ <p>โดย <math>N = 5</math> ดังนั้นความซับซ้อนของระบบคือ</p> $147$	<p><b>DFG LMS</b></p> $4N + 1$ <p>โดย <math>N = 5</math> ดังนั้นความซับซ้อนของระบบคือ</p> $21$

## บทที่ 5

### บทสรุปและบทวิจารณ์

#### 5.1 สรุปผลการทดลอง

จากการทดลองในบทที่ 4 มีการนำ LMS Algorithm 4 ชนิดมาเปรียบเทียบความเร็วในการลู่เข้า และหาค่าความผิดพลาดเฉลี่ยกำลังน้อยสุด โดยการขมิเลขันด้วยโปรแกรมแมทแลป ทำให้ทราบว่า หากมีการเพิ่มค่าสัมประสิทธิ์การลู่เข้า (step size ( $\mu$ )) เข้าไปในระบบก็จะสามารถเพิ่มความเร็วในการลู่เข้าให้มากขึ้น เนื่องจากค่าสัมประสิทธิ์การลู่เข้า คือตัวที่ควบคุมว่าสัมประสิทธิ์จะมีค่าเปลี่ยนแปลงตามเวกเตอร์ทิศทางเร็วมากน้อยแค่ไหน ดังนั้นการเลือกค่าสัมประสิทธิ์การลู่เข้าที่เหมาะสมจะมีผลต่อประสิทธิภาพต่อการทำงานของระบบมาก

จากการทดลองจะเห็นได้อย่างชัดเจนว่าไม่ว่าจะทำการปรับค่า step size ( $\mu$ ) ให้มากขึ้นเท่าไร LMS Algorithm ที่ถูกพัฒนาแล้ว ในที่นี้คือ The LMS - Newton Algorithm, The Normalized LMS Algorithm และ The Partially Filter Gradient LMS Algorithm ทั้ง 3 อัลกอริทึมจะมีความเร็วในการลู่เข้าที่ดีกว่าแอลเอ็มเอสอัลกอริทึมแบบดั้งเดิมหรือ Conventional LMS Algorithm ทั้งสิ้น ทั้งนี้เพราะอัลกอริทึมที่ได้ถูกพัฒนามาแล้วทั้ง 3 แบบที่ได้กล่าวไปแล้วมีการใช้ความซับซ้อนในการคำนวณเพิ่มเข้าไปในอัลกอริทึมซึ่งวิธีดังกล่าวจะช่วยให้ประสิทธิภาพของระบบดียิ่งขึ้น แต่จะมีความยุ่งยากและความซับซ้อนในการคำนวณมากขึ้นด้วย

#### 5.2 ปัญหาที่เกิดขึ้นในด้านเทคนิค

เนื่องจากแอลเอ็มเอสอัลกอริทึมมีอยู่มากมายหลายชนิด จึงต้องใช้เวลาในการเลือกแอลเอ็มเอสอัลกอริทึมที่เหมาะสมทำการเปรียบเทียบกับแอลเอ็มเอสแบบดั้งเดิม โดยเกณฑ์ในการเลือกนั้นจะเลือกจากความเหมาะสม รวมไปถึงความนิยมในตัวอัลกอริทึม ซึ่งต้องใช้เวลาในการคัดเลือกอัลกอริทึมก่อนที่จะนำอัลกอริทึมเหล่านั้นมาเขียนเป็นโค้ดในโปรแกรมแมทแลป อีกทั้งเมื่อนำอัลกอริทึมมาขมิเลขันด้วยโปรแกรมแมทแลปแล้วยังมีข้อผิดพลาด และยังคงปรับค่าพารามิเตอร์อีกหลายตัวจนเหมาะสมซึ่งใช้เวลานานกว่าจะได้ผลการขมิเลขันที่สมบูรณ์ออกมาได้

#### 5.3 แนวทางในการพัฒนาต่อและการประยุกต์ใช้งาน

โดยภาพรวมของการเปรียบเทียบประสิทธิภาพของตระกูล LMS สำหรับวงจรกรองแบบปรับตัวได้นั้นการทดลองเบื้องต้นเกี่ยวกับความเร็วในการลู่เข้าของสัญญาณและค่าเฉลี่ยกำลังน้อยสุดซึ่งพารามิเตอร์ทั้งสองตัวนี้เป็นตัวชี้วัดประสิทธิภาพของระบบในวงจรกรองความถี่แบบปรับตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้ เมื่อมีการป้อนสัญญาณอินพุตและทำการปรับค่า step size ( $\mu$ ) ในลักษณะต่างๆกันแล้ว จะทำให้วงจรกรอมมีประสิทธิภาพโดยรวมของระบบดียิ่งขึ้นได้ ส่วนการประยุกต์ใช้งาน สามารถนำไปปรับใช้ประโยชน์ในด้านต่างๆ ได้หลายรูปแบบ เช่น การนำวงจรกรอมความถี่แบบปรับตัวได้ไปประยุกต์ใช้ในการจำลองคุณลักษณะของระบบที่ไม่ทราบ (System Identification), การพัฒนาอีควอลไลเซอร์แบบปรับตัวได้ (Adaptive Equalization), การกำจัดสัญญาณรบกวน (Noise Cancellation) เป็นต้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

- [1] Emmanuel C.Ifeachor, Barrie W.Jervis, "Digital Signal Processing A Practical Approach, "
- [2] C.F.N.Cowan, Prof. J. Mavor, "New difital Adaptive filter implementation sing distributed arithmetic techniques, " IEE Proc., vol. 128, No.4, pp.225-230, Aug.1981
- [3] Kyo Takahashi, Yoshitake Tsunekawa, " Analysis of convergence condition of LMS Adaptive Digital Filter using Distributed Arithmetic, " IECE Trans. Fundamental, vol.E85-A, pp.1249-1256, June 2002
- [4] C.F.N.Cowan, P.M.Grant, " Adaptive filter, " " Prentice Hall, 1980.
- [5] ดร.พรชัย ภวงค์ศักดิ์, " การประมวลผลสัญญาณดิจิทัลเบื้องต้น , " <http://www.ee.mut.ac.th/home/pornchai/12/1999>
- [6] ศ.ดร. วัลลภ สุระกำพลธร, " การประมวลผลสัญญาณเชิงเลข, " สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, ตุลาคม พ.ศ. 2533
- [7] " การปรับแต่งช่องสัญญาณในระบบสื่อสารไร้สายดิจิทัล " โดย พีระพล ยูวณิชตานนท์ ภาควิชาวิศวกรรมอิเล็กทรอนิกส์ มหาวิทยาลัยเทคโนโลยีมหานคร <http://www.eegrad.mut.ac.th/home/peerapol/dsp2ndse.htm>
- [8] Paulo S.R. Diniz, " Adaptive Filtering, Algorithms and Practical Implementation " Second Edition