การกำจัดสัญญาณรบกวนในเสียงพูดด้วยการใช้ตัวกรองแบบเรียบ

Noise Reduction in Speech by Using a Smoothing Filter

นายกันอัณณพ ชัยเร็ว รหัสนิสิต 46380004

นายวัฒนา มัธยมนันทน์ รหัสนิสิต 46380041

นายบุรินทร์ ฉัตรแก้ว รหัสนิสิต 46380158

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2550

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	การกำจัดสัญญาณรบกวนในเสียงพูดด้วยการใช้ตัวกรองแบบเรียบ
ผู้ดำเนินโครงงาน	นาย กันอัฌฌพ ชัยเร็ว รหัส 46380004
	นาย วัฒนา มัทธยมนันท์ รหัส 46380041
	นาย บุรินทร์ ฉัตรแก้ว รหัส 46380158
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มเม่น
สาขาวิชา	วิศวกรรมไฟฟ้า, วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2550

....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า, วิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

....ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มเม่น)

....กรรมการ
(ดร.สมพร เรืองสินชัยวานิช)

....กรรมการ
(ดร.ไพศาล มุณีสว่าง)

หัวข้อโครงงาน	การกำจัดสัญญาณรบกวนในเสียงพูดด้วยการใช้ตัวกรองแบบเรียบ
ผู้ดำเนินโครงงาน	นาย กันอัณณพ ชัยเร็ว รหัส 46380004
	นาย วัฒนา มัธยมนันทน์ รหัส 46380041
	นาย บุรินทร์ ฉัตรแก้ว รหัส 46380158
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร. สุชาติ แย้มเม่น
สาขาวิชา	วิศวกรรมไฟฟ้า, วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2550

บทคัดย่อ

ในปัจจุบันมีการใช้สัญญาณเสียงติดต่อสื่อสารอย่างแพร่หลาย แต่เมื่อเสียงถูกส่งผ่านตัวกลางอาจเกิดสัญญาณรบกวน ทำให้เสียงที่ได้รับบิดเบือนไปจากเสียงต้นแบบ ดังนั้น โครงงานนี้จึงนำเสนอวิธีการกำจัดสัญญาณรบกวนในเสียงพูด โดยการสร้างตัวกรองแบบเรียบค่าเฉลี่ยเคลื่อนที่ และ แบบเอ็กซ์โปเนนเชียล เพื่อทำการลดสัญญาณรบกวนและเปรียบเทียบประสิทธิภาพของตัวกรองจากสัญญาณที่ผ่านตัวกรองกับสัญญาณต้นแบบ

สร้างตัวกรองโดยใช้วิธีลากรานซ์หาสัมประสิทธิ์ของตัวกรองทั้งสองแบบ แล้วนำสัญญาณต้นแบบที่ถูกรบกวนจากเกาส์เซียนไวท์นอยซ์มาเข้าตัวกรอง โดยตัวกรองทั้งสองจะนำข้อมูลที่ได้มาคอนโวลูชั่นกับสัมประสิทธิ์ที่พัฒนาขึ้น โดยกำหนดค่าคงที่ของตัวกรองเอ็กซ์โปเนนเชียลมีค่าคงที่เท่ากับ 1 และ 40 จากนั้นปรับกำลังของผลลัพธ์สัญญาณที่ได้ให้เท่ากับกำลังของสัญญาณต้นแบบ พร้อมเปรียบเทียบสัญญาณทั้งสองโดยใช้วิธีหาค่าเฉลี่ยความคลาดเคลื่อนยกกำลังสอง

จากผลการทดลองพบว่าค่าเฉลี่ยความคลาดเคลื่อนยกกำลังสองของวิธีตัวกรองเอ็กซ์โปเนนเชียลที่ใช้ค่าคงที่ 40 จะมีค่าเข้าใกล้ศูนย์มากที่สุด แสดงว่ามีประสิทธิภาพในการกำจัดสัญญาณรบกวนได้ดีกว่าวิธีตัวกรองค่าเฉลี่ยเคลื่อนที่ และถ้าตัวกรองแบบเรียบเอ็กซ์โปเนนเชียลที่ใช้ค่าคงที่ 1 จะได้ค่าเฉลี่ยความคลาดเคลื่อนยกกำลังสองใกล้เคียงกับค่าที่ได้จากตัวกรองแบบเรียบค่าเฉลี่ยเคลื่อนที่ นอกจากนี้การฟังเสียงที่ผ่านการกรองสัญญาณรบกวน และ พิจารณาเปรียบเทียบจากกราฟสัญญาณเสียงที่ผ่านตัวกรองแบบต่างๆ พบว่าตัวกรองแบบเรียบเอ็กซ์โปเนนเชียลที่ใช้ค่าคงที่ 1 และตัวกรองแบบเรียบค่าเฉลี่ยเคลื่อนที่ มีประสิทธิภาพในการกำจัดสัญญาณรบกวนออกไปจากเสียงพูดต้นแบบได้ดีกว่าตัวกรองแบบเรียบเอ็กซ์โปเนนเชียลที่ใช้ค่าคงที่ 40

Project Title	Noise Reduction in Speech by Using a Smoothing Filter		
Name	Mr.Kununnop	Chairew	ID.46380004
	Mr.Watthana	Madhyamanandana	ID.46380041
	Mr.Burin	Chatkaew	ID.46380158
Project Advisor	Assistant Professor Suchart Yammen, Ph.D.		
Major	Electrical Engineering, Computer Engineering		
Department	Electrical and Computer Engineering.		
Academic Year	2007		

ABSTRACT

At present, the use of sound signal in communications is widespread. When it is sent through a medium, it may be distorted by a noise. That can cause the listeners to get an unclear sound. Thus, this project is proposed a noise reduction method in the speech by using the moving average smoothing filter and the exponential smoothing filter. Furthermore, this will also compare a efficiency of the filtered signal with the original signal.

The filter will be produced by the usage of Lagrange Multiplier Method to find out the coefficients of the two filters. Then, the original signal corrupted by Gaussian White Noise will be transmitted to the filters. Both filters will transfer the data to convolution with the developing coefficients, as set the constant exponential smoothing filter at "1" and "40". Later, adjust the power of output signal equal to the power of original signal and at the same time, compare both signals by using the Mean Square Error method.

From the experiment, it was shown that the Mean Square Error of the exponential smoothing filtered method using the constant value of "40" was closely approached to zero. This means that it can eliminate the disturbing signal more effective than the moving average smoothing filtered method. If the exponential smoothing filtered method applied the constant value of "1", we will get the Mean Square Error which is quite similar to the value from the moving average smoothing filtered method. However, when consider the sound filtered from the filters and the sound graphs, it was found that the exponential smoothing filter of constant value "1" and the moving average smoothing filter can eliminate the noise from the original speech more effective than the exponential smoothing filter using the constant value of "40".

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยความกรุณาของบุคคลหลายฝ่าย ขอกราบขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร. สุชาติ แย้มเม่น ที่ช่วยจุดประกายความคิดในด้านต่างๆ อีกทั้งได้ให้คำแนะนำ ข้อแก้ไข และดูแลการทำปริญญานิพนธ์ด้วยดีเสมอมา ดร. สมพร เรืองสินชัยวานิช ให้คำแนะนำที่ดีแก่คณะผู้เขียน และ ดร. ไพศาล มุณีสว่าง ให้คำแนะนำที่ดีมาโดยตลอด

ขอกราบขอบพระคุณ บิดา มารดา ที่ได้ให้ความอุปการะและเป็นกำลังใจให้แก่คณะผู้เขียนอย่างดียิ่ง จนกระทั่งคณะผู้เขียนสำเร็จการศึกษา ตลอดจน ญาติ พี่น้อง และเพื่อนๆที่น่ารักยิ่งทุกคนของคณะผู้เขียน ที่ได้เป็นกำลังใจช่วยผลักดันให้ปริญญานิพนธ์ฉบับนี้สำเร็จด้วยดี

ท้ายที่สุดนี้ หากมีสิ่งขาดตกบกพร่องหรือผิดพลาดประการใด คณะผู้เขียนขออภัยเป็นอย่างสูงในข้อบกพร่องและความผิดพลาดนั้น และคณะผู้เขียนหวังว่าปริญญานิพนธ์นี้จะมีประโยชน์บ้างสำหรับหน่วยงานที่เกี่ยวข้อง ตลอดจนผู้ที่สนใจศึกษารายละเอียดในด้านการกำจัดสัญญาณรบกวนออกจากเสียงพูดด้วยตัวกรองแบบเรียบต่อไป

นายกันอัฌณพ ชัยเร็ว
นายวัฒนา มัทธยมนันทน์
นายบุรินทร์ ฉัตรแก้ว

สารบัญ

สารบัญ(ต่อ)

สารบัญตาราง

ตารางที่ หน้า

สารบัญรูป

สารบัญรูป(ต่อ)

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญ

การติดต่อสื่อสารในปัจจุบันโดยใช้สัญญาณเสียงเป็นตัวกลางในการติดต่อนั้นเป็นที่นิยมอย่างแพร่หลาย อาทิเช่น การติดต่อสื่อสารทางโทรศัพท์ การพูดคุยผ่านอินเตอร์เน็ตโปรโตคอล เป็นต้น ซึ่งการติดต่อสื่อสารเหล่านี้อาจมีสัญญาณรบกวนปะปนอยู่ระหว่างการติดต่อสื่อสารซึ่งอาจส่งผลทำให้ผู้รับการติดต่อสื่อสารบริเวณปลายทางนั้นได้รับข้อมูลที่ผิดพลาดไปหรือไม่ครบถ้วนตามความเป็นจริง ดังนั้นการกำจัดสัญญาณรบกวนด้วยวงจรกรองความถี่จึงเข้ามามีบทบาทสำคัญในการติดต่อสื่อสารในปัจจุบันเพื่อให้ผู้รับการติดต่อสื่อสารนั้นได้รับข้อมูลที่ส่งมาจากต้นทางได้อย่างถูกต้องครบถ้วนตามความเป็นจริง

วงจรกรองความถี่แบ่งออกเป็น 2 รูปแบบคือ วงจรกรองความถี่แบบอนาลอก และ วงจรกรองความถี่แบบดิจิทัล โดยมีหน้าที่จำแนกความถี่ตามความต้องการของผู้ใช้โดยแบ่งตามคุณลักษณะของผลตอบสนองความถี่ 4 ชนิด ได้แก่ วงจรกรองความถี่ต่ำ (Low-pass Filter: LPF), วงจรกรองความถี่สูง (High-pass Filter: HPF), วงจรกรองแถบความถี่ผ่าน (Band-pass Filter: BPF) และวงจรกรองแถบความถี่หยุดผ่าน (Band-stop Filter: BFS)

ในโครงงานการกำจัดสัญญาณรบกวนในเสียงพูดนี้ ได้ใช้วิธีการกรองผ่านตัววงจรความถี่แบบดิจิทัลโดยใช้วงจรกรองความถี่ 2 แบบคือ วงจรกรองความถี่แบบค่าเฉลี่ยเคลื่อนที่ (Moving Average) และ วงจรกรองความถี่แบบเอ็กซ์โปเนนเชียล (Exponential) โดยผู้จัดทำต้องการที่จะศึกษาการกำจัดสัญญาณรบกวนของเสียงและเปรียบเทียบวงจรกรองความถี่ทั้ง 2 แบบว่าตัวใดมีประสิทธิภาพในการกำจัดสัญญาณเสียงได้ดีกว่ากัน

1.2 วัตถุประสงค์

1.2.1 ศึกษา วงจรกรองความถี่แบบค่าเฉลี่ยเคลื่อนที่ (Moving Average)

1.2.2 ศึกษา วงจรกรองความถี่แบบเอ็กซ์โปเนนเชียล (Exponential)

1.2.3 เปรียบเทียบประสิทธิภาพของวงจรกรองความถี่แบบค่าเฉลี่ยเคลื่อนที่ และ วงจรกรองความถี่แบบเอ็กซ์โปเนนเชียลโดยการลดสัญญาณรบกวนในเสียงพูด

1.3 ขอบข่ายของโครงงาน

1.3.1 ใช้วงจรกรองความถี่แบบดิจิทัลในการกรองสัญญาณรบกวนในเสียงพูด

1.3.2 ใช้วงจรกรองความถี่แบบค่าเฉลี่ยเคลื่อนที่ (Moving Average) ในการกรองสัญญาณรบกวนออกจากเสียงพูด

1.3.3 ใช้วงจรกรองความถี่แบบเอ็กซ์โปเนนเชียล (Exponential) ในการกรองสัญญาณรบกวนออกจากเสียงพูด

1.3.4 เปรียบเทียบประสิทธิภาพของวงจรกรองสัญญาณรบกวนทั้ง 2 แบบด้วยค่าเฉลี่ยความคลาดเคลื่อนยกกำลัง 2 (Mean Square Error)

1.4 แผนการดำเนินงานตลอดโครงการ

ตารางที่ 1.1 ตารางการปฏิบัติงาน

กิจกรรมการดำเนินงาน	ปีการศึกษา 2550				
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.
1. ศึกษาและออกแบบวงจรกรองความถี่แบบค่าเฉลี่ยเคลื่อนที่	←—	—→			
2. ศึกษาและออกแบบวงจรกรองความถี่แบบเอ็กซ์โปเนนเชียล		←—	—→		
3. ทดสอบและแก้ไขโปรแกรม			←—	—→	
4. สรุปผลและจัดทำรูปเล่มโครงงาน				←—	—→

1.5 ผลที่คาดว่าจะได้รับ

1.5.1 ได้วงจรกรองความถี่ต่ำแบบค่าเฉลี่ยเคลื่อนที่ (Moving Average)

1.5.2 ได้วงจรกรองความถี่ต่ำแบบเอ็กซ์โปเนนเชียล (Exponential)

1.5.3 ได้สัญญาณเสียงที่ถูกกำจัดสัญญาณรบกวน

1.6 รายละเอียดงบประมาณตลอดโครงงาน

1.6.1 ค่าวัสดุสำนักงาน	1,500	บาท
1.6.2 ค่าวัสดุคอมพิวเตอร์	1,000	บาท
1.6.3 ค่าเอกสาร	500	บาท
รวมทั้งสิ้น	3,000	บาท (สามพันบาทถ้วน)

บทที่ 2
หลักการและทฤษฎี

ปัจจุบันการทำธุรกิจต่างๆหรือการติดต่อสื่อสารกันนั้นต้องการความชัดเจนและแน่นอนของการรับส่งข้อมูล เนื่องจากสัญญาณเมื่อผ่านตัวกลางต่างๆแล้วทำให้มีสัญญาณรบกวนเกิดขึ้นมาดังนั้นจึงต้องมีการเรียนเทคนิคที่จะกำจัดหรือลดสัญญาณรบกวนนั้นออกจากสัญญาณ

การศึกษานี้จะใช้สัญญาณรบกวนแบบเกาส์เซียน โดยเทคนิคที่จะนำมาใช้ในการกรองสัญญาณรบกวน คือ วิธีตัวกรองแบบเรียบค่าเฉลี่ยเคลื่อนที่ (Standard Moving Average Smoothing Filters) ตัวกรองเส้นตรงไม่ป้อนกลับลำดับที่ q (Linear Non-Recursive Filters of Order q) และ ตัวกรองแบบเรียบเอ็กซ์โปเน็นเชียล (Optimum Exponential Trend Smoothing Filters) ค่าคาดคะเนความแปรปรวน (Expected Value and Variance) การหาค่าสูงสุดและต่ำสุดแบบมีข้อจำกัด รวมถึง ยูคลิเดียนนอร์ม (Euclidean norm) หรือ ทูนอร์ม (2-Norm) เพื่อรักษาขนาดเสียงให้เท่าเดิม และวัดประสิทธิภาพในการกรองสัญญาณรบกวน โดยใช้ ค่าเฉลี่ยความคลาดเคลื่อนยกกำลังสอง (Mean Square Error, MSE)

2.1 ตัวกรองเส้นตรงไม่ป้อนกลับลำดับที่ q (Linear Non-Recursive Filters of Order q)

จาก FIR filters ลำดับที่ q มีฟังก์ชันถ่ายโอนดังนี้

$$H(z) = b_0 + b_1 z^{-1} + \ldots + b_q z^{-q} \tag{2.1}$$

เทคอินเวอร์สซีทรานส์ฟอร์ม $Y(z) = H(z)X(z)$ และ ใช้ Delay property กับสมการ (2.1) จะได้วงจรกรอง Linear Non-Recursive Filters of Order q

$$y[n] = \sum_{k=0}^{q} b_k x[n-k] \tag{2.2}$$

โดยที่ $x[n]$ และ $y[n]$ แทนลำดับสัญญาณที่ด้านเข้าและด้านออกของวงจรและ $b_0, b_1, \ldots, b_q$ คือสัมประสิทธิ์ของวงจรกรอง

2.2 เกาส์เซียนไวท์นอยซ์ (Gaussian White Noise)

จำลองสัญญาณรบกวน (Noise) โดยใช้ Gaussian White Noise รบกวนสัญญาณต้นฉบับโดย Gaussian หรือ normal probability density function มีสมการดังนี้

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[\frac{-(x-\mu)^2}{\sigma^2}\right] \tag{2.3}$$

โดยที่ μ คือค่าเฉลี่ย

σ คือส่วนเบี่ยงเบนมาตรฐาน

ในที่นี้จะใช้ฟังก์ชันของแมตแล็บ (Mat lab) ในการสร้างสัญญาณรบกวนคือ rand โดยที่ $\mu = 0$ และ σ =1 ซึ่งมีลักษณะของสัญญาณรบกวนเป็นกราฟรูประฆังคว่ำ (Bell-Shape)

2.3 ค่าคาดคะเนและความแปรปรวน (Expected Value and Variance)

2.3.1 ค่าคาดคะเน (Expected Value)

เมื่อ $g(x) = x$ จะได้ว่า $g(X) = X$ ดังนั้นในกรณีที่ $E[X] < \infty$ จะได้ว่า

$$E[X] = \begin{cases} \displaystyle\sum_{x \in lmX} xP(X = x) & \text{เมื่อ เป็นตัวแปรสุ่มไม่ต่อเนื่อง} \\ \displaystyle\int_{-\infty}^{\infty} xf(x)dx & \text{เมื่อเป็นตัวแปรสุ่มต่อเนื่อง} \end{cases}$$

คุณสมบัติของค่าคาดคะเน

1. กำหนดให้ X เป็นตัวแปรสุ่มซึ่งสามารถหาค่าคาดคะเนได้ จะได้ว่า สำหรับค่าคงตัว a และ b ใดๆ

$$E[aX + b] = aE[X] + b \tag{2.4}$$

2. ให้ X และ Y เป็นตัวแปรสุ่มซึ่งสามารถหาค่าคาดคะเนได้ จะได้ว่า

$$E[X \pm Y] = E[X] \pm E[Y] \tag{2.5}$$

3. ให้ X และ Y เป็นตัวแปรสุ่มซึ่งสามารถหาค่าคาดคะเนได้ ถ้า X และ Y เป็นอิสระต่อกัน แล้วจะได้ว่า XY สามารถหาค่าคาดคะเนได้โดยที่

$$E[XY] = E[X]E[Y] \tag{2.6}$$

4. กำหนดให้ X และ Y เป็นตัวแปรสุ่มซึ่งสามารถหาค่าคาดคะเนได้ จะได้ว่า X และ Y เป็นอิสระต่อกันก็ต่อเมื่อ

$$E[g(X)h(Y)] = E\left[g(x)\right]\left[E(h(Y)\right] \tag{2.7}$$

เมื่อทุกฟังก์ชัน $f,h : R \to R$ ที่ทำให้ $g(X)$ และ $h(X)$ เป็นตัวแปรสุ่มที่หาค่าคาดคะเนได้

2.3.2 ความแปรปรวน (Variance)

ความแปรปรวน (Variance) ของ X ใช้สัญลักษณ์แทนด้วย σ_x^2 หรือ $Var(X)$ กล่าวคือ

$$\begin{aligned}
\sigma_X^2 &= E\left[\left(X - E(X)\right)^2\right] \\
&= E\left[X^2 - 2XE[X] + \left(E[X]\right)^2\right] \\
&= E\left[X^2\right] - 2\left(E[X]\right)^2 + \left(E[X]\right)^2 \\
&= E\left[X^2\right] - \left(E[X]\right)^2
\end{aligned} \tag{2.8}$$

เนื่องจาก $Var(X) \geq 0$

ดังนั้นจากสมการ (2.8) สรุปได้ว่า $E[X^2] \geq (E[X])^2$

ความแปรปรวนร่วม (Covariance) ของตัวแปรสุ่ม X และ Y คือ

$E[(X - E(X))(Y - E(Y))]$ เขียนแทนด้วย σ_{xy} หรือ $Cov(X,Y)$

ข้อสังเกต

1) $\sigma_{xy} = E[XY] - E[X]E[Y]$

2) ถ้า X และ Y เป็นอิสระต่อกันจะได้ว่า $\sigma_{xy} = 0$ เนื่องจาก

$E[XY] = E[X]E[Y]$

คุณสมบัติความแปรปรวน

1. สำหรับค่าคงตัว b ใดๆ

$$Var(b) = 0 \tag{2.9}$$

2. ถ้า X เป็นตัวแปรสุ่มที่สามารถหาค่าความแปรปรวนได้จะได้ว่า

$$Var(aX + b) = a^2 Var(X) \tag{2.10}$$

เมื่อ a และ b เป็นค่าคงตัวใดๆ

3. สำหรับตัวแปรสุ่ม X และ Y ใดๆที่สามารถหาค่าความแปรปรวนได้จะได้ว่า

$$Var(aX + bY) = a^2 Var(X) + b^2 Var(Y) + 2ab.Cov(X,Y) \tag{2.11}$$

เมื่อ a และ b เป็นค่าคงตัวใดๆ

4. ถ้า X และ Y เป็นตัวแปรสุ่มที่เป็นอิสระต่อกันแล้ว สามารถหาค่าความแปรปรวนได้ว่า

$$Var(aX + bY) = a^2 Var(X) + b^2 Var(Y) \tag{2.12}$$

เมื่อ a และ b เป็นค่าคงตัวใดๆ

5. กำหนดให้ $X_1, X_2, ..., X_q (q \geq 2)$ เป็นอิสระต่อกันจะได้ว่า

$$Var(X_1, X_2, ..., X_q) = Var(X_1) + Var(X_2) + ... + Var(X_q) \tag{2.13}$$

6. ให้ $y[n] = \sum_{k=0}^{q} b_k x[n-k]$ เป็นตัวแปรสุ่มที่อิสระต่อกัน

$$\sigma_y^2 = \rho \sigma_w^2 \tag{2.14}$$

$$\rho = \sum_{k=0}^{q} b_k^2 \tag{2.15}$$

เมื่อ b เป็นค่าคงที่ใดๆ

2.4 การหาค่าสูงสุดและต่ำสุด (Maximize and Minimize)

การหาค่าสูงสุดและต่ำสุด แบ่งออกเป็น 2 กรณีใหญ่ๆ คือ การหาค่าสูงสุดและต่ำสุด กรณีไม่มีข้อจำกัด กับการหาค่าสูงสุดและต่ำสุด กรณีมีข้อจำกัด

2.4.1 กรณีมีข้อจำกัด

2.4.1.1 กรณีตัวแปรอิสระตัวเดียว จะต้องมีเงื่อนไข 2 ข้อ คือ

เงื่อนไขที่ 1: เรียกว่าเงื่อนไขจำเป็น (Necessary Condition) โดยการหา First Derivative ของฟังก์ชันและกำหนดให้เท่ากับศูนย์ เพื่อแก้สมการหาค่าวิกฤต (Critical Value) ของตัวแปรอิสระที่ทำให้ฟังก์ชันมีค่าสูงสุดหรือต่ำสุด

เงื่อนไขที่ 2: เรียกว่าเงื่อนไขพอเพียง (Sufficient Condition) โดยการหา Second Derivative ของฟังก์ชันเพื่อตรวจสอบเครื่องหมาย

1. ถ้าได้เครื่องหมายบวก แสดงว่า เป็นค่าต่ำสุดของฟังก์ชัน
2. ถ้าได้เครื่องหมายลบ แสดงว่า เป็นค่าสูงสุดของฟังก์ชัน

ฟังก์ชันที่มีตัวแปรอิสระตัวเดียวเช่น $y = f(x)$ จะได้

เงื่อนไขที่จำเป็น คือ $\frac{dy}{dx} = 0$

เงื่อนไขเพียงพอ คือ $\frac{d^2y}{d^2x} > 0$ ได้ เป็นค่าต่ำสุดของฟังก์ชัน

หรือ $\frac{d^2y}{d^2x} < 0$ ได้ เป็นค่าสูงสุดของฟังก์ชัน

2.4.1.2 กรณีตัวแปรอิสระหลายตัว เช่น $y = f(x_1, x_2, ..., x_n)$

จะได้ First Order Total Differential คือ $dy = 0$

$$\frac{\partial . dx_1}{\partial x_1} + \frac{\partial . dx_2}{\partial x_2} + ... + \frac{\partial . dx_n}{\partial x_n} = 0$$

แต่ $dx_1, dx_2, ..., dx_n$ ไม่จำเป็นต้องเท่ากับศูนย์ ดังนั้นเงื่อนไขจำเป็นของตัวแปรอิสระหลายตัวคือ

$$\frac{\partial f}{\partial x_1} = \frac{\partial f}{\partial x_2} + ... + \frac{\partial f}{\partial x_n} = 0$$

เงื่อนไขที่เพียงพอ คือ $\frac{\partial^2 f}{\partial^2 x} > 0$ ได้เป็นค่าต่ำสุดของฟังก์ชัน

หรือ $\frac{\partial^2 f}{\partial^2 x} < 0$ ได้เป็นค่าสูงสุดของฟังก์ชัน

2.4.2 กรณีมีข้อจำกัด

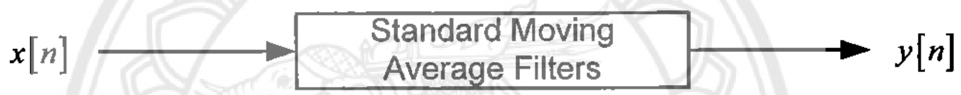
เป็นวิธีการคำนวณที่นิยมใช้เป็นอย่างมาก การหาค่าสูงสุดและต่ำสุดกรณีนี้จะใช้วิธีการของLagrange เรียกว่า Lagrange Multiplier Method มีขั้นตอนการหาดังนี้

ขั้นที่ 1. สร้างสมการเป้าหมายใหม่ เรียกว่า Lagrange Function (L) โดยการใช้ Lagrange Multiplier (λ) คูณกับสมการข้อจำกัดและนำไปบวกกับสมการเดิม

ขั้นที่ 2. หาค่า Partial Derivative ของ L มุ่งตรงต่อการเปลี่ยนแปลงของ $x_1, x_2, \ldots, x_n$ ที่ละตัวแล้วกำหนดให้มีค่าเท่ากับศูนย์

ขั้นที่ 3. แก้สมการหาค่า $x_1, x_2, \ldots, x_n$ และ λ แล้วแทนค่าที่ได้ลงในสมการเป้าหมายก็จะได้ค่าสูงสุดหรือต่ำสุด

2.5 ตัวกรองแบบเรียบค่าเฉลี่ยเคลื่อนที่ (Standard Moving Average Smoothing Filters)



จากสมการ (2.2) คือ

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

สมมุติให้ $x[n]$ เป็นค่าคงที่ คือ

$$x[n] = a_0$$

ดังนั้น

$$x[n-k] = a_0$$

นำ $x[n]$ แทนในสมการ (2.2) จะได้

$$y[n] = \sum_{k=0}^{q} b_k a_0$$

$$y[n] = a_0 \sum_{k=0}^{q} b_k = x[n] = a_0$$

ดังนั้นเงื่อนไขที่จะทำให้ $y[n] = x[n]$ คือ

$$\sum_{k=0}^{q} b_k = 1$$

หรือ

$$b_0 + b_1 + b_2 + \ldots + b_q = 1 \tag{2.16}$$

ต่อมาทำการหาค่า ρ จากสมการ (2.15) คือ $\rho = \sum_{k=0}^{q} b_k^2$ ให้มีค่าต่ำสุด (เข้าใกล้ค่าศูนย์) เพื่อทำการคาดคะเน $y[n]$ ให้มีค่าใกล้เคียงกับ $x[n]$ โดยใช้วิธี Lagrange Multiplier กรณีมีข้อจำกัดจากสมการ (2.16) จะได้

$$\min_{\sum_{k=0}^{q} b_k = 1} \sum_{k=0}^{q} {b_k}^2 \tag{2.17}$$

จากสมการ (2.17) จะได้เงื่อนไขของ b_k คือ

$$1 - \sum_{k=0}^{q} b_k = 0 \tag{2.18}$$

สร้างฟังก์ชันใหม่คือ

$$f(b_k, \lambda) = \sum_{k=0}^{q} b_k^2 + \lambda \left\{ 1 - \sum_{k=0}^{q} b_k \right\} \tag{2.19}$$

ทำการ Differential ฟังก์ชัน f ด้วย $b_0, b_1, \ldots, b_k$ ตามลำดับ

$$\frac{\partial f}{\partial b_0} = 2b_0 + \lambda(0-1)$$
$$\frac{\partial f}{\partial b_1} = 2b_1 + \lambda(0-1)$$
$$\vdots$$
$$\frac{\partial f}{\partial b_k} = 2b_k + \lambda(0-1) \tag{2.20}$$

สำหรับ k=0, 1, 2,..., q

ทำการ Differential ฟังก์ชัน f ด้วย λ จะได้

$$\frac{\partial f}{\partial \lambda} = 1 - \sum_{k=0}^{q} b_k \tag{2.21}$$

กำหนดสมการ (2.20) = 0 จะได้

$$2b_k^{\circ} - \lambda^{\circ} = 0$$
$$2b_k^{\circ} = \lambda^{\circ} \tag{2.22}$$

กำหนดสมการ (2.21) = 0 จะได้

$$1 - \sum_{k=0}^{q} b_k^{\circ} = 0$$
$$\sum_{k=0}^{q} b_k^{\circ} = 1 \tag{2.23}$$
$$\sum_{k=0}^{q} 2b_k^{\circ} = 2 \tag{2.24}$$

นำสมการ (2.22) แทนในสมการ (2.24) จะได้

$$\sum_{k=0}^{q} \lambda^{\circ} = 2$$
$$\lambda^{\circ}(q+1) = 2$$
$$\lambda^{\circ} = \frac{2}{q+1} \tag{2.25}$$

นำสมการ (2.25) แทนในสมการ (2.22) จะได้

$$2b_k^{\circ} = \frac{2}{q+1}$$

$$b_k^{\circ} = \frac{1}{q+1} \tag{2.26}$$

จากสมการ (2.15) คือ

$$\rho = \sum_{k=0}^{q} {b_k^{\circ}}^2$$

นำสมการ (2.26) แทนในสมการ (2.15) จะได้

$$\rho = \sum_{k=0}^{q} \left(\frac{1}{q+1}\right)^2$$

$$\rho = \frac{1}{q+1} \sum_{k=0}^{q} \frac{1}{q+1}$$

ดังนั้นจะได้

$$\rho = \frac{1}{q+1} \tag{2.27}$$

นำสมการ (2.26) แทนในสมการ (2.2) จะได้

$$y[n] = \frac{1}{q+1}(x[n] + x[n-1] + x[n-2] + \ldots + x[n-q])$$

เมื่อ $x[n]$ คือข้อมูลลำดับที่ n และ $y[n]$ คือค่าของข้อมูลที่กรองแล้วลำดับที่ n สมการของ Standard Moving Average Filters มีดังนี้

$$y[n] = \frac{1}{q+1} \sum_{k=0}^{q} x[n-k] \tag{2.28}$$

สำหรับ $n \in \{0,1,2,...,N-1\}$ จากสมการ (2.27) เห็นได้ว่าผลลัพธ์ $y[n]$ นั้นเป็นผลรวมข้อมูลปัจจุบันย้อนหลังไปจนถึงข้อมูลที่ q และหารด้วยจำนวนข้อมูลทั้งหมด ซึ่งก็คือค่ากลางจะเห็นได้ว่าข้อมูลถูก Delay ไป $\frac{q}{2}$

2.6 ตัวกรองแบบเรียบเอ็กซ์โปเน็นเชียล (Optimum Exponential Trend Smoothing Filters)

กำหนดให้

$$x[n] = a_0\left(a_1\right)^n \tag{2.29}$$

จากสมการ (2.2) คือ

$$y[n] = b_0x[n] + b_1x[n-1] + b_2x[n-2] + ... + b_qx[n-q] \tag{2.30}$$

นำสมการ (2.29) แทนในสมการ (2.30) จะได้

$$y[n] = b_0a_0\left(a_1\right)^n + b_1a_0\left(a_1\right)^{n-1} + b_2a_0\left(a_1\right)^{n-2} + ... + b_qa_0\left(a_1\right)^{n-q} \tag{2.31}$$

ต้องการให้ $y[n] \cong X[n]$ จะได้

$$a_0\left(a_1\right)^n = b_0a_0\left(a_1\right)^n + b_1a_0\left(a_1\right)^{n-1} + b_2a_0\left(a_1\right)^{n-2} + ... + b_qa_0\left(a_1\right)^{n-q} \tag{2.32}$$

นำ a_0 หารทั้ง 2 ข้างของสมการ (2.32) จะได้

$$\left(a_1\right)^n = b_0\left(a_1\right)^n + b_1\left(a_1\right)^{n-1} + b_2\left(a_1\right)^{n-2} + ... + b_q\left(a_1\right)^{n-q} \tag{2.33}$$

นำ $\left(a_1\right)^n$ หารทั้ง 2 ข้างของสมการ (2.33) จะได้

$$1 = b_0 + b_1\left(a_1\right)^{-1} + b_2\left(a_1\right)^{-2} + ... + b_q\left(a_1\right)^{-q}$$

$$1=\sum_{k=0}^{q}\left[b_k\left(a_1\right)^{-k}\right]$$

หรือ

$$\sum_{k=0}^{q}\left[b_K\left(a_1\right)^{-k}\right]=1 \tag{2.34}$$

สำหรับ k=0, 1, 2,…,q

ต่อมาทำการหาค่า ρ จากสมการ (2.15) คือ $\rho=\sum_{k=0}^{q}b_k^2$ ให้มีค่าต่ำที่สุด (เข้าใกล้ค่าศูนย์) เพื่อทำการคาดคะเน $y[n]$ ให้มีค่าใกล้เคียงกับข้อมูลจริง $x[n]$ ด้วยวิธี Lagrange Multiplier

$$\min_{\sum_{k=0}^{q}b_k(a_1)^{-k}=1}\sum_{k=0}^{q}{b_k}^2 \tag{2.35}$$

จากสมการ (2.35) จะได้เงื่อนไข b_K คือ

$$1-\sum_{k=0}^{q}\left[b_K\left(a_1\right)^{-k}\right]=0 \tag{2.36}$$

สร้างฟังก์ชันใหม่คือ

$$f\left(b_0,b_1,b_{2,\ldots,}b_q,\lambda\right)=\sum_{k=0}^{q}\left({b_K}^2\right)+\lambda\left\{1-\sum_{k=0}^{q}\left[b_K\left(a_1\right)^{-k}\right]\right\} \tag{2.37}$$

ทำการ Differential ฟังก์ชัน f ด้วย $b_0,b_1,\ldots,b_k$ ตามลำดับ

$$\begin{aligned}\frac{\partial f}{\partial b_0}&=2b_0+\lambda\{0-1\}\\ \frac{\partial f}{\partial b_1}&=2b_1+\lambda\left\{0-\left(a_1\right)^{-1}\right\}\\ &\vdots\\ \frac{\partial f}{\partial b_k}&=2b_k+\lambda\left\{0-\left(a_1\right)^{-k}\right\}\end{aligned} \tag{2.38}$$

สำหรับ k=0, 1, 2,..., q

ทำการ Differential ฟังก์ชัน f ด้วย λ จะได้

$$\frac{\partial f}{\partial \lambda} = 1 - \sum_{k=0}^{q}\left[b_k \left(a_1\right)^{-k} \right] \tag{2.39}$$

กำหนดให้สมการ (2.38) = 0 จะได้

$$2b_k^{\circ} - \lambda^{\circ}\left(a_1\right)^{-k} = 0$$

$$2b_k^{\circ} = \lambda^{\circ}\left(a_1\right)^{-k} \tag{2.40}$$

กำหนดให้สมการ (2.39) = 0 จะได้

$$1 - \sum_{k=0}^{q} b_k^{\circ}\left(a_1\right)^{-k} = 0$$

$$\sum_{k=0}^{q} b_k^{\circ}\left(a_1\right)^{-k} = 1 \tag{2.41}$$

นำ 2 คูณสมการ (2.41) จะได้

$$\sum_{k=0}^{q} 2b_k^{\circ}\left(a_1\right)^{-k} = 2 \tag{2.42}$$

นำสมการ (2.40) แทนในสมการ (2.42) จะได้

$$\sum_{k=0}^{q} \lambda^{\circ}\left(a_1\right)^{-2k} = 2$$

$$\lambda^{\circ} = \frac{2}{\sum_{k=0}^{q}\left(a_1\right)^{-2k}} \tag{2.43}$$

นำสมการ (2.43) แทนในสมการ (2.40)

$$2b_k^{\circ} = \frac{2}{\sum_{k=0}^{q}(a_1)^{-2k}}(a_1)^{-k}$$

$$b_k^{\circ} = \frac{1}{\sum_{k=0}^{q}(a_1)^{-2k}}(a_1)^{-k} \qquad (2.44)$$

จากสมการ (2.15) คือ

$$\rho = \sum_{k=0}^{q} b_k^{\circ 2}$$

นำสมการ (2.44) แทนในสมการ (2.15) จะได้

$$\rho = \sum_{k=0}^{q}\left[\frac{(a_1)^{-k}}{\sum_{m=0}^{q}(a_1)^{-2m}}\right]^2$$

$$= \sum_{k=0}^{q}\frac{(a_1)^{-2k}}{\left[\sum_{m=0}^{q}(a_1)^{-2m}\right]^2}$$

$$= \frac{1}{\left[\sum_{m=0}^{q}\left(a_1^{-2m}\right)\right]^2}\left[\sum_{k=0}^{q}(a_1)^{-2k}\right]$$

ดังนั้นจะได้

$$\rho = \frac{1}{\sum_{m=0}^{q}\left(a_1^{-2m}\right)} \qquad \text{เมื่อ } a_1 \neq 0 \qquad (2.45)$$

นำสมการ (2.44) แทนค่าในสมการ (2.2) จะได้ Optimum Exponential Trend Smoothing Filters

$$y[n] = \frac{1}{\sum_{k=0}^{q}(a_1)^{-2k}} \sum_{k=0}^{q}(a_1)^{-k} x[n-k] \tag{2.46}$$

สำหรับ $n \in \{0,1,2,...,N-1\}$ เมื่อข้อมูล $y[n]$ เป็นผลลัพธ์ของข้อมูลหลังจากผ่านตัวกรองลำดับที่ n และ $x[n]$ เป็นข้อมูลที่ถูกสัญญาณรบกวนลำดับที่ n

2.7 ยูคลิเดียนนอร์ม (Euclidean norm) หรือ ทูนอร์ม (2-Norm)

เมื่อลดสัญญาณรบกวนแล้วขนาดของเสียงลดลงทำให้เสียงที่ผ่านตัวกรองมีเสียงเบาลงจึงต้องใช้ 2-Norm เพื่อหาขนาดของสัญญาณก่อนเข้าตัวกรองและปรับให้สัญญาณที่กรองออกมามีขนาดเสียงเท่าเดิมโดยบนวงกลมหนึ่งหน่วยวิธีที่จะหาขนาดเวกเตอร์ x มีสูตรดังนี้

$$\|\underline{x}\|_2 = (\sum_{i=0}^{n}|x_i|^2)^{\frac{1}{2}} \tag{2.47}$$

โดยที่ $\|\underline{x}\|_2$ คือขนาดของเวกเตอร์ x

2.8 การวัดประสิทธิภาพ

ในการวัดประสิทธิภาพของการลดสัญญาณรบกวนของแต่ละวิธีนั้นจะใช้ค่าเฉลี่ยความคลาดเคลื่อนยกกำลังสอง (Mean Square Error, MSE) เพื่อวัดประสิทธิภาพความคลาดเคลื่อนของแต่ละวิธี มีสมการดังนี้

$$MSE = \frac{1}{N}\sum_{n=0}^{N-1}(y[n]-s[n])^2 \tag{2.48}$$

สำหรับ $n \in \{0,1,2,...,N-1\}$ เมื่อ N คือจำนวนข้อมูลทั้งหมด โดยที่ $y[n]$ เป็นค่าของสัญญาณเสียงที่ถูกกำจัดสัญญาณรบกวนลำดับที่ n และ $s[n]$ เป็นค่าของสัญญาณเสียงที่ปราศจากสัญญาณรบกวนของข้อมูลลำดับที่ n

บทที่ 3
ขั้นตอนและวิธีการกำจัดสัญญาณรบกวน

การกำจัดสัญญาณรบกวนคือการกำจัดสัญญาณที่ไม่ต้องการหรือสัญญาณที่ทำให้สัญญาณต้นแบบเปลี่ยนแปลงไปจากเดิม โดยวิธีการในการกำจัดสัญญาณรบกวนนั้นจะจำแนกตามความถี่ที่ผู้ใช้ต้องการ โดยแบ่งตามคุณลักษณะของผลตอบสนองความถี่ 4 ชนิด ได้แก่ วงจรกรองความถี่ต่ำ (Low-pass Filters: LPF), วงจรกรองความถี่สูง (High-pass Filters: HPF), วงจรกรองแถบความถี่ผ่าน (Band-pass Filters: BPF) และวงจรกรองแถบความถี่หยุดผ่าน (Band-stop Filters: BFS) ซึ่งแล้วแต่ว่าผู้ใช้นั้นต้องการความถี่ของสัญญาณแบบไหน

ในบทนี้ ขอนำเสนอขั้นตอนและวิธีการกำจัดสัญญาณรบกวน โดยผ่านตัวกรองแบบเรียบค่าเฉลี่ยเคลื่อนที่ (Moving Average: MA) และ เอ็กซ์โปเนนเชียล (Exponential: Expo) ซึ่งในบทนี้อธิบายถึงขั้นตอนและวิธีการในการออกแบบตัวกรองแบบเรียบ (Smoothing Filters) และลำดับขั้นตอนในการกำจัดสัญญาณรบกวนในแต่ละวิธี รวมถึงการตรวจสอบหาความแตกต่างว่าวิธีการกำจัดสัญญาณแบบใดจะสามารถกำจัดสัญญาณ ได้ดีที่สุด

3.1 การกรองสัญญาณรบกวนโดยใช้ตัวกรองแบบเรียบค่าเฉลี่ยเคลื่อนที่

กำหนดให้ $x[n]$ คือ ค่าเวกเตอร์ของสัญญาณข้อมูลเสียงที่ถูกรบกวนโดย Gaussian White Noise

$$x[n] = s[n] + w[n] \tag{3.1}$$

โดยที่ $s[n]$ คือ ค่าเวกเตอร์ของสัญญาณข้อมูลเสียง

$w[n]$ คือ ค่าของสัญญาณรบกวนแบบ Gaussian White Noise ในสมการที่ (2.3) โดยทำการจำลองสัญญาณรบกวนลงในข้อมูลเสียงต้นแบบ

ขั้นตอนที่ 1 นำ $x[n]$ จากสมการ (3.1) ไปแทนในสมการ (2.28) ซึ่งเป็นสมการตัวกรองแบบเรียบค่าเฉลี่ยเคลื่อนที่ (Moving Average Smoothing Filters)

$$y[n] = \frac{1}{q+1}\sum_{k=0}^{q} x[n-k] \tag{3.2}$$

โดยที่ $y[n]$ ที่ได้นี้เป็นสัญญาณข้อมูลเสียงที่ถูกกำจัดสัญญาณรบกวนแล้ว

ขั้นตอนที่ 2 เนื่องจากหลังจากผ่านตัวกรองไปแล้วเสียงเบาลงจึงต้องทำการปรับให้มีขนาดเท่าเดิม จากสมการ (2.47) นำ $x[n]$ จากสมการ (3.1) ไปแทนค่าเพื่อหาขนาดของเวกเตอร์ได้ดังนี้

$$\|\underline{x}\|_2 = ((x[0])^2 + (x[1])^2 + \ldots + (x[q])^2)^{\frac{1}{2}}$$
$$\|\underline{x}\|_2 = (\sum_{i=0}^{q} |x[i]|^2)^{\frac{1}{2}} \tag{3.3}$$

โดยที่ $\|\underline{x}\|_2$ คือขนาดของเวกเตอร์ $x[n]$ ที่มีจำนวนข้อมูล $q+1$ ตัว
จากนั้นนำค่า $y[n]$ จากสมการ (3.2) มาหาขนาดของเวกเตอร์

$$\|\underline{y}\|_2 = ((y[0])^2 + (y[1])^2 + \ldots + (y[q])^2)^{\frac{1}{2}}$$
$$\|\underline{y}\|_2 = (\sum_{i=0}^{q} |y[i]|^2)^{\frac{1}{2}} \tag{3.4}$$

โดยที่ $\|\underline{y}\|_2$ คือขนาดของเวกเตอร์ $x[n]$ ที่มีจำนวนข้อมูล $q+1$ ตัว
นำ $y[n]$ จากสมการ (3.2) หารด้วย $\|\underline{y}\|_2$ ที่ได้จากสมการ (3.4) ซึ่งจะได้เวกเตอร์หนึ่งหน่วยมาคูณกับ $\|\underline{x}\|_2$ จากสมการ (3.3) ทำให้ขนาดเสียงดังเท่ากับเสียงต้นแบบก่อนผ่านตัวกรอง

$$yma[n] = \frac{y[n]}{\|\underline{y}\|_2} \times \|\underline{x}\|_2 \tag{3.5}$$

โดยที่ $yma[n]$ คือค่าที่ได้รับการกรองสัญญาณรบกวนและเพิ่มขนาดเสียงให้เท่ากับเสียงต้นแบบ

ขั้นตอนที่ 3 การวัดประสิทธิภาพในการวัดประสิทธิภาพของการลดสัญญาณรบกวนนั้นจะใช้ค่าเฉลี่ยความคลาดเคลื่อนยกกำลังสอง (Mean Square Error, MSE) เพื่อวัดประสิทธิภาพความคลาดเคลื่อนโดยใช้สมการ (2.48) คือ

$$MSE = \frac{1}{N}\sum_{n=0}^{N-1}(y[n] - s[n])^2$$

3.2 การกรองสัญญาณรบกวนโดยใช้ตัวกรองแบบเรียบเอ็กซ์โปเนนเชียล

ขั้นตอนที่ 1 นำ $x[n]$ จากสมการ (3.1) ไปแทนในสมการ (2.45) ซึ่งเป็นสมการตัวกรองแบบเรียบเอกซ์โปเนนเซียล

$$y[n]=\frac{1}{\sum_{k=0}^{q}(a_1)^{-2k}}\sum_{k=0}^{q}(a_1)^{-k}x[n-k] \tag{3.6}$$

โดยที่ $y[n]$ ที่ได้นี้เป็นสัญญาณข้อมูลเสียงที่ถูกกำจัดสัญญาณรบกวนแล้ว

ขั้นตอนที่ 2 เนื่องจากหลังจากผ่านตัวกรองไปแล้วเสียงเบาลงจึงต้องทำการปรับให้**มีขนาดเท่าเดิม** จากสมการ (2.47) นำ $x[n]$ จากสมการ (3.1) ไปแทนค่าเพื่อหาขนาดของเวกเตอร์ได้ดัง**นี้**

$$\begin{aligned}\|\underline{x}\|_2 &= ((x[0])^2+(x[1])^2+\ldots+(x[q])^2)^{\frac{1}{2}}\\ \|\underline{x}\|_2 &= (\sum_{i=0}^{q}|x[i]|^2)^{\frac{1}{2}}\end{aligned} \tag{3.7}$$

โดยที่ $\|\underline{x}\|_2$ คือขนาดของเวกเตอร์ $x[n]$ ที่มีจำนวนข้อมูล $q+1$ ตัว
จากนั้นนำค่า $y[n]$ จากสมการ (3.6) มาหาขนาดของเวกเตอร์

$$\begin{aligned}\|\underline{y}\|_2 &= ((y[0])^2+(y[1])^2+\ldots+(y[q])^2)^{\frac{1}{2}}\\ \|\underline{y}\|_2 &= (\sum_{i=0}^{q}|y[i]|^2)^{\frac{1}{2}}\end{aligned} \tag{3.8}$$

โดยที่ $\|\underline{y}\|_2$ คือขนาดของเวกเตอร์ $x[n]$ ที่มีจำนวนข้อมูล $q+1$ ตัว
นำ $y[n]$ จากสมการ (3.6) หารด้วย $\|\underline{y}\|_2$ ที่ได้จากสมการ (3.8) ซึ่งจะได้เวกเตอร์หนึ่งหน่วยมาคูณกับ $\|\underline{x}\|_2$ จากสมการ (3.7) ทำให้ขนาดเสียงดังเท่ากับเสียงต้นแบบก่อนผ่านตัวกรอง

$$yex[n]=\frac{y[n]}{\|\underline{y}\|_2}\times\|\underline{x}\|_2 \tag{3.9}$$

โดยที่ $yex[n]$ คือค่าที่ได้รับการกรองสัญญาณรบกวนและเพิ่มขนาดเสียงให้เท่ากับเสียงต้นแบบ

ขั้นตอนที่ 3 การวัดประสิทธิภาพ ในการวัดประสิทธิภาพของการลดสัญญาณรบกวนนั้นจะใช้ค่าเฉลี่ยความคลาดเคลื่อนยกกำลังสอง (Mean Square Error, MSE) เพื่อวัดประสิทธิภาพความคลาดเคลื่อนโดยใช้สมการ (2.48) คือ

$$MSE = \frac{1}{N}\sum_{n=0}^{N-1}(y[n]-s[n])^2$$

3.3 การเท่ากันของค่า b_k ของ Moving Average Smoothing Filters และ Exponential Smoothing Filters กรณีที่ $a_1 = 1$ และ $q = 80$

b_k ของ Moving Average Smoothing Filters จากสมการ (2.26) จะได้

$$\begin{aligned} b_k &= \frac{1}{q+1} \\ &= \frac{1}{80+1} \\ &= \frac{1}{81} \end{aligned}$$

b_k ของ Exponential Smoothing Filters จากสมการ (2.44) จะได้

$$\begin{aligned} b_k^{\circ} &= \frac{1}{\sum_{k=0}^{q}(a_1)^{-2k}}(a_1)^{-k} \\ &= \frac{(1)^{-80}}{(1)^{-2(0)}+(1)^{-2(1)}+(1)^{-2(2)}+\ldots+(1)^{-2(80)}} \\ &= \frac{1}{1+1+1+\ldots+1} \\ &= \frac{1}{81} \end{aligned}$$

ดังนั้น ค่า b_k ของทั้ง Moving Average Smoothing Filters และ Exponential Smoothing Filters ในกรณีที่ $a_1 = 1$ และ $q = 80$ มีค่าเท่ากัน

บทที่ 4

ผลการทดลอง

ในบทนี้จะแสดงผลการเปรียบเทียบค่าของ $a_1 = 1$ และ $a_1 = 40$ ระหว่าง Exponential Filters และ Moving Average Filters ด้วยการวัดประสิทธิภาพของสัญญาณรบกวนด้วยวิธี ค่าเฉลี่ยความคลาดเคลื่อนยกกำลังสอง (Mean Square Error, MSE) เมื่อกำหนดให้ q ซึ่งคือความยาวของฟิลเตอร์มีค่าคงที่เท่ากับ 80 และ แอมปลิจูดของสัญญาณรบกวนเท่ากับ 0.1, 0.5, 0.05 ดังแสดงในตารางที่ 4.1-4.3 ตามลำดับ และ แสดงรูปลักษณะของสัญญาณเสียงที่ผ่านตัวกรองสัญญาณรบกวนแบบต่างๆเมื่อแอมปลิจูดของสัญญาณรบกวนเท่ากับ 0.1, 0.5, 0.05 ดังแสดงในรูปที่ 4.1-4.15 ตามลำดับ

ตารางที่ 4.1 เปรียบเทียบค่า Mean Square Error ของ $a_1 = 1$ และ $a_1 = 40$ ระหว่าง Exponential Filters และ Moving Average Filters โดยกำหนดค่าความยาวของฟิลเตอร์เท่ากับ 80 และ แอมปลิจูดของสัญญาณรบกวนเท่ากับ 0.1

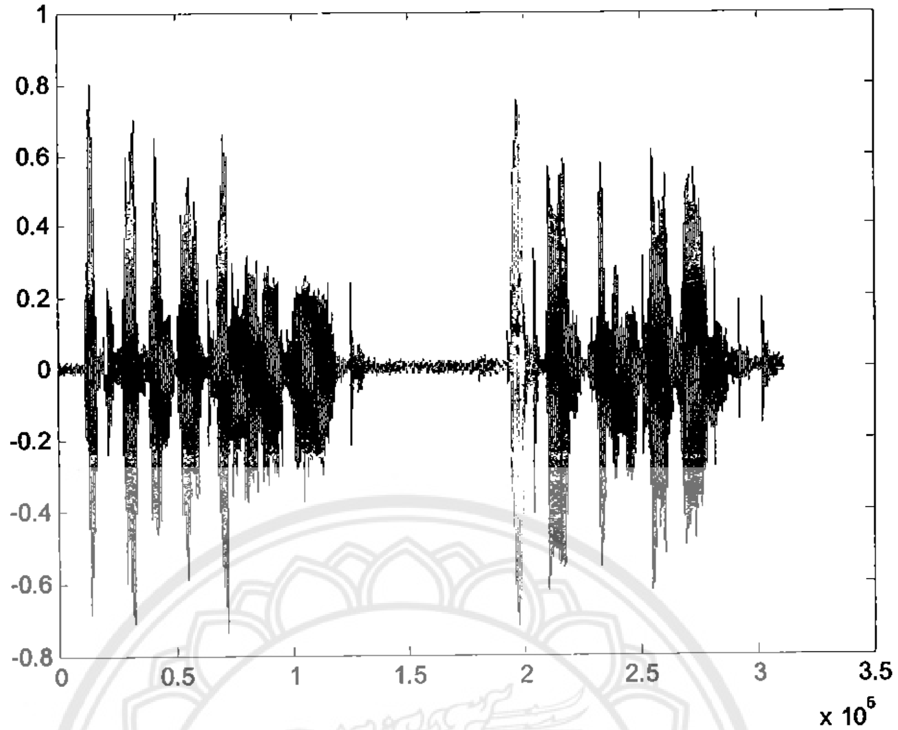
Exponential Filters เมื่อ q=80		Moving Average Filters เมื่อ q=80
$\frac{1}{N}\sum_{n=0}^{N-1}(y[n]-s[n])^2$ ที่ $a_1 = 1$	$\frac{1}{N}\sum_{n=0}^{N-1}(y[n]-s[n])^2$ ที่ $a_1 = 40$	$\frac{1}{N}\sum_{n=0}^{N-1}(y[n]-s[n])^2$
0.0333	0.0097	0.0334
0.0332	0.0097	0.0335
0.0333	0.0098	0.0334
0.0334	0.0097	0.0334
0.0333	0.0097	0.0334
0.0332	0.0098	0.0333
0.0333	0.0097	0.0334
0.0333	0.0097	0.0333
0.0333	0.0097	0.0335
0.0334	0.0098	0.0334
0.0332	0.0097	0.0333
0.0334	0.0098	0.0334

ตารางที่ 4.1(ต่อ) เปรียบเทียบค่า Mean Square Error ของ $a_1 = 1$ และ $a_1 = 40$ ระหว่าง Exponential Filters และ Moving Average Filters โดยกำหนดค่าความยาวของฟิลเตอร์เท่ากับ 80 และ แอมปลิจูดของสัญญาณรบกวนเท่ากับ 0.1

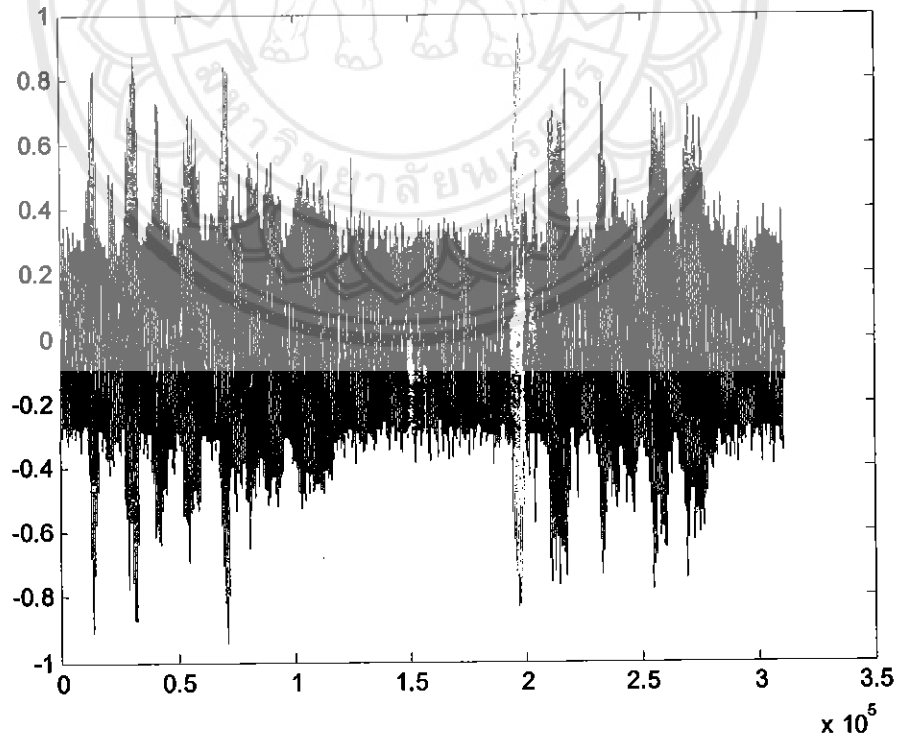
0.0333	0.0098	0.0335
0.0332	0.0097	0.0334
0.0333	0.0097	0.0335
0.0332	0.0097	0.0335
0.0333	0.0097	0.0335
0.0333	0.0097	0.0334
0.0333	0.0097	0.0335
0.0333	0.0097	0.0334

ค่าเฉลี่ย

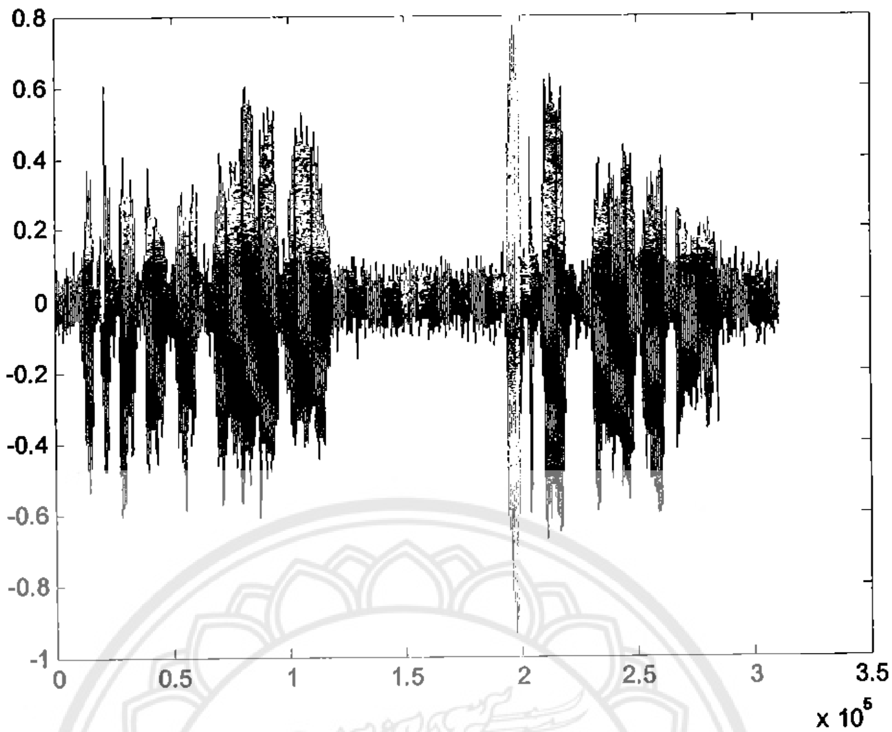
0.0333	0.0097	0.0334



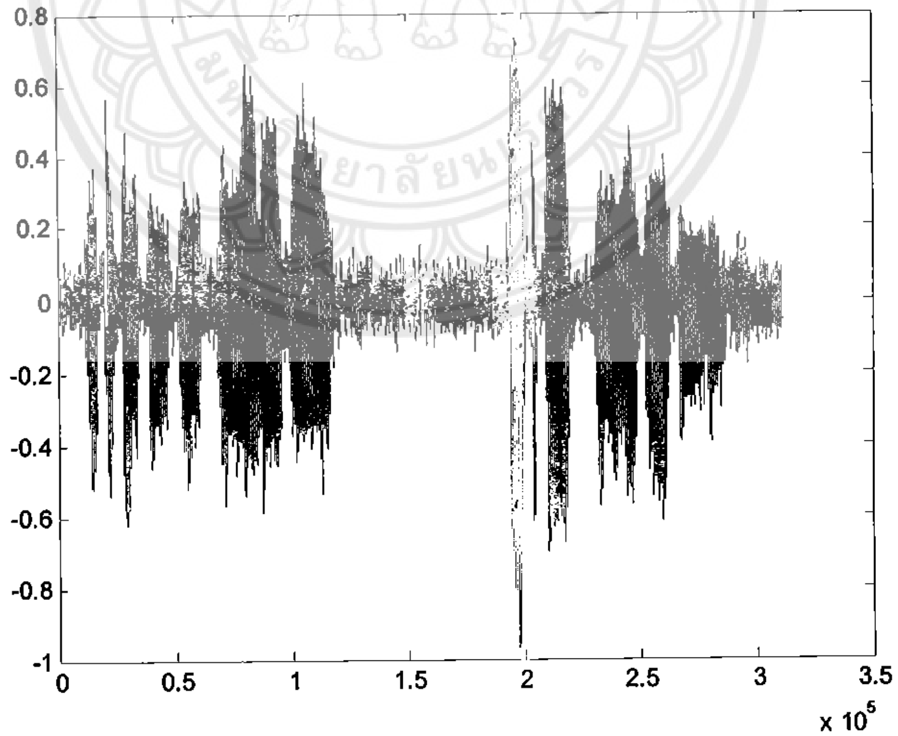
รูปที่ 4.1 สัญญาณเสียงที่ปราศจากสัญญาณรบกวนเมื่อแอมปลิจูดของสัญญาณรบกวนเท่ากับ 0.1



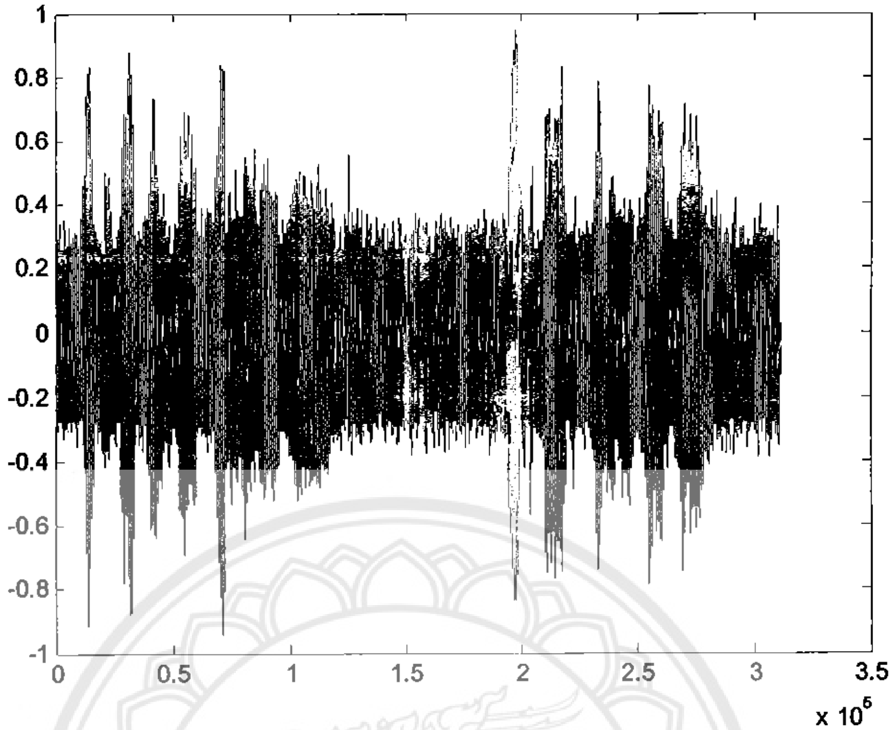
รูปที่ 4.2 สัญญาณเสียงที่รวมกับสัญญาณรบกวนเมื่อแอมปลิจูดของสัญญาณรบกวนเท่ากับ 0.1



รูปที่ 4.3 สัญญาณเสียงที่ถูกกำจัดสัญญาณรบกวน กรณีที่ $a_1 = 1$ ของ Moving Average Filters เมื่อแอมปลิจูดของสัญญาณรบกวนเท่ากับ 0.1



รูปที่ 4.4 สัญญาณเสียงที่ถูกกำจัดสัญญาณรบกวน กรณีที่ $a_1 = 1$ ของ Exponential Filters เมื่อแอมปลิจูดของสัญญาณรบกวนเท่ากับ 0.1



รูปที่ 4.5 สัญญาณเสียงที่ถูกกำจัดสัญญาณรบกวน กรณีที่ $a_1 = 40$ ของ Exponential Filters เมื่อแอมปลิจูดของสัญญาณรบกวนเท่ากับ 0.1

4.1 วิเคราะห์และวิจารณ์ผลการทดลองกรณีที่ค่าความยาวของฟิลเตอร์เท่ากับ 80 และ แอมปลิจูดของสัญญาณรบกวนเท่ากับ 0.1

จากตารางที่ 4.1 กรณีของ Exponential Filters เมื่อทำการเปรียบเทียบ $a_1 = 1$ และ $a_1 = 40$ โดยการวัดประสิทธิภาพของการลดสัญญาณรบกวนด้วยวิธีค่าเฉลี่ยความคลาดเคลื่อนยกกำลังสอง (Mean Square Error, MSE) พบว่า เมื่อ $a_1 = 40$ ค่า MSE เข้าใกล้ค่าศูนย์มากกว่า กรณีที่ $a_1 = 1$ แสดงว่าเมื่อ $a_1 = 40$ มีประสิทธิภาพในการกำจัดสัญญาณรบกวนได้ดีกว่า $a_1 = 1$ กรณี Moving Average Filters เมื่อทำการวัดประสิทธิภาพของการลดสัญญาณรบกวนด้วยวิธีค่าเฉลี่ยความคลาดเคลื่อนยกกำลังสอง (Mean Square Error, MSE) พบว่าค่า MSE มีค่าใกล้เคียงกับ $a_1 = 1$ กรณี Exponential Filters และเมื่อนำมาเปรียบเทียบกับค่า MSE ของ $a_1 = 40$ กรณี Exponential Filters จะเห็นว่าค่า MSE ของ $a_1 = 40$ กรณี Exponential Filters มีค่าเข้าใกล้ค่าศูนย์มากกว่า แสดงว่า เมื่อ พิจารณาเปรียบเทียบในเชิงตัวเลขแล้ว $a_1 = 40$ กรณี Exponential Filters มีประสิทธิภาพในการกำจัดสัญญาณรบกวนได้ดีกว่าวิธีใดๆดังที่ได้กล่าวมา

จากรูปที่ 4.3 เป็นรูปสัญญาณเสียงที่ถูกกำจัดสัญญาณรบกวน กรณีที่ $a_1 = 1$ ของ Moving Average Filters เมื่อนำมาเปรียบเทียบกับรูปที่ 4.2 คือสัญญาณเสียงรวมกับสัญญาณรบกวน พบว่าสัญญาณรบกวนถูกลดทอนลงไปมากซึ่งมีลักษณะคล้ายคลึงกับรูปที่ 4.1 ซึ่งเป็นรูปสัญญาณเสียงที่

ปราศจากสัญญาณรบกวน ดังนั้นแสดงว่า เมื่อพิจารณาเปรียบเทียบเชิงรูปภาพแล้ว กรณีที่ $a_1 = 1$ ของ Moving Average Filters สามารถกำจัดสัญญาณรบกวนออกจากเสียงพูดได้ดี

จากรูปที่ 4.4 เป็นรูปสัญญาณเสียงที่ถูกกำจัดสัญญาณรบกวน กรณีที่ $a_1 = 1$ ของ Exponential Filters เมื่อนำมาเปรียบเทียบกับรูปที่ 4.2 คือสัญญาณเสียงรวมกับสัญญาณรบกวน พบว่า สัญญาณรบกวนถูกลดทอนลงไปมากซึ่งมีลักษณะคล้ายคลึงกับรูปที่ 4.1 ซึ่งเป็นรูปสัญญาณเสียงที่ปราศจากสัญญาณรบกวน ดังนั้นแสดงว่า เมื่อพิจารณาเปรียบเทียบเชิงรูปภาพแล้ว กรณีที่ $a_1 = 1$ ของ Exponential Filters สามารถกำจัดสัญญาณรบกวนออกจากเสียงพูดได้ดี

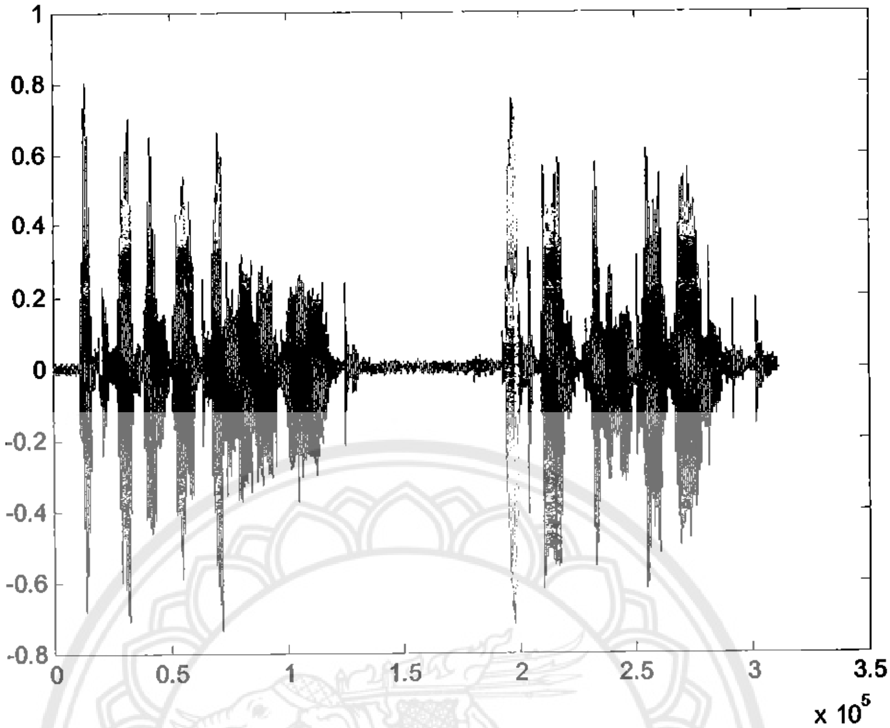
จากรูปที่ 4.5 เป็นรูปสัญญาณเสียงที่ถูกกำจัดสัญญาณรบกวน กรณีที่ $a_1 = 40$ ของ Exponential Filters เมื่อนำมาเปรียบเทียบกับรูปที่ 4.2 คือ สัญญาณเสียงรวมกับสัญญาณรบกวน พบว่ามีลักษณะใกล้เคียงกันกับรูปที่ 4.2 แสดงว่า สัญญาณรบกวนถูกกำจัดหรือลดทอนออกไปจากเสียงพูดได้น้อยมาก ดังนั้น เมื่อพิจารณาเปรียบเทียบเชิงรูปภาพกับรูปที่ 4.1 คือ สัญญาณเสียงที่ปราศจากสัญญาณรบกวน พบว่า กรณีที่ $a_1 = 40$ ของ Exponential Filters สัญญาณรบกวนถูกกำจัดออกไปจากเสียงพูดได้น้อยมาก

ตารางที่ 4.2 เปรียบเทียบค่า Mean Square Error ของ $a_1 = 1$ และ $a_1 = 40$ ระหว่าง Exponential Filters และ Moving Average Filters โดยกำหนดค่าความยาวของฟิลเตอร์เท่ากับ 80 และ แอมปลิจูดของสัญญาณรบกวนเท่ากับ 0.5

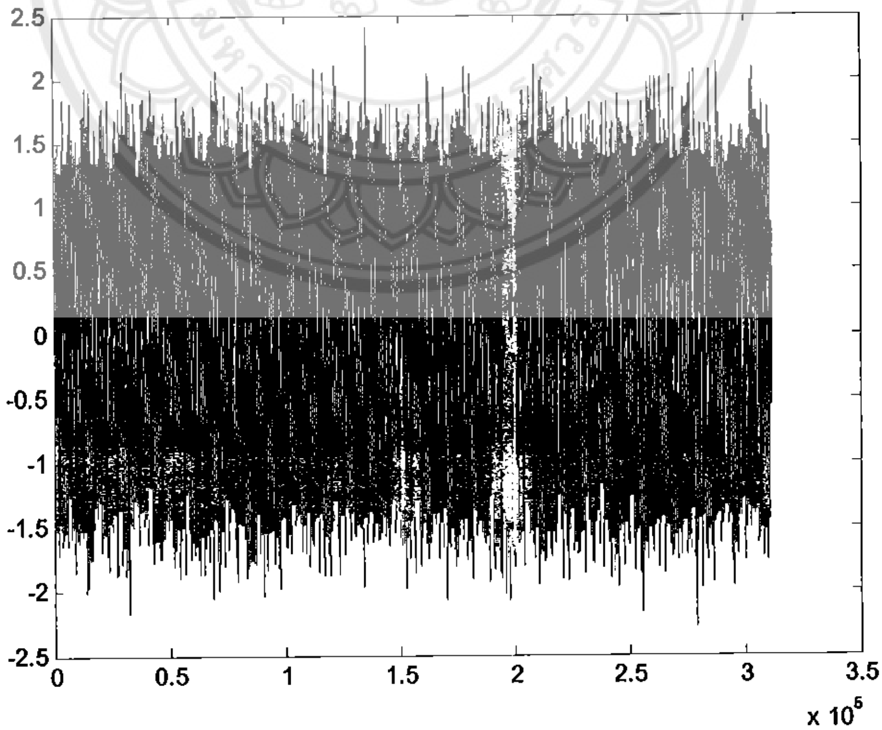
Exponential Filters เมื่อ q=80		Moving Average Filters เมื่อ q=80
$\frac{1}{N}\sum_{n=0}^{N-1}(y[n]-s[n])^2$ ที่ $a_1 = 1$	$\frac{1}{N}\sum_{n=0}^{N-1}(y[n]-s[n])^2$ ที่ $a_1 = 40$	$\frac{1}{N}\sum_{n=0}^{N-1}(y[n]-s[n])^2$
0.2710	0.2502	0.2712
0.2697	0.2485	0.2700
0.2706	0.2495	0.2709
0.2713	0.2490	0.2716
0.2706	0.2489	0.2708
0.2690	0.2484	0.2692
0.2703	0.2490	0.2706
0.2708	0.2488	0.2710
0.2697	0.2487	0.2699
0.2729	0.2503	0.2731
0.2694	0.2490	0.2796
0.2722	0.2505	0.2725
0.2700	0.2491	0.2703
0.2719	0.2498	0.2722
0.2707	0.2485	0.2709
0.2711	0.2498	0.2714
0.2719	0.2500	0.2722
0.2702	0.2496	0.2705
0.2707	0.2495	0.2710
0.2702	0.2492	0.2707

ค่าเฉลี่ย

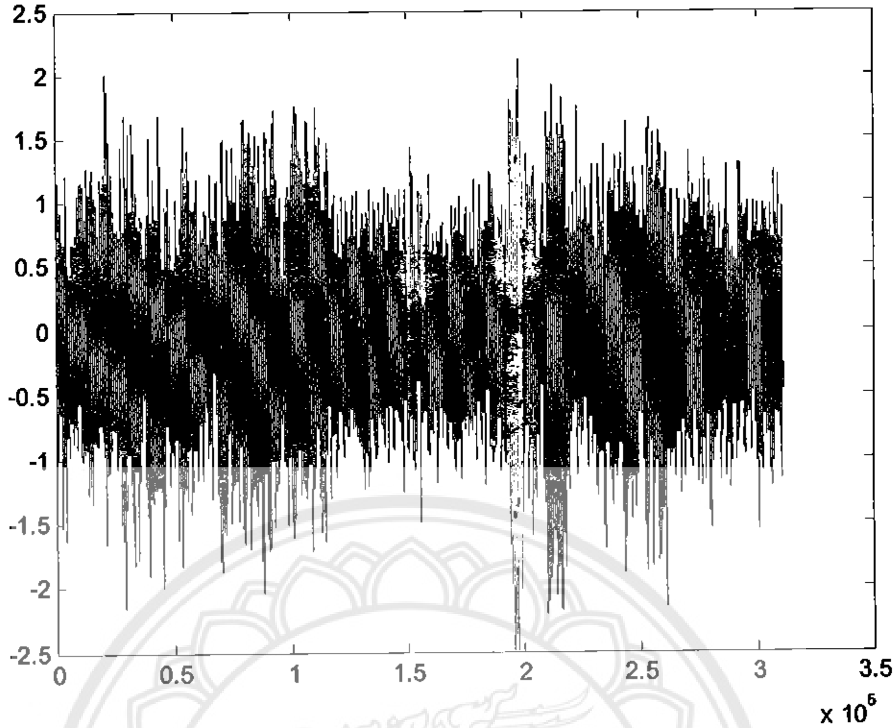
0.2707	0.2493	0.2715



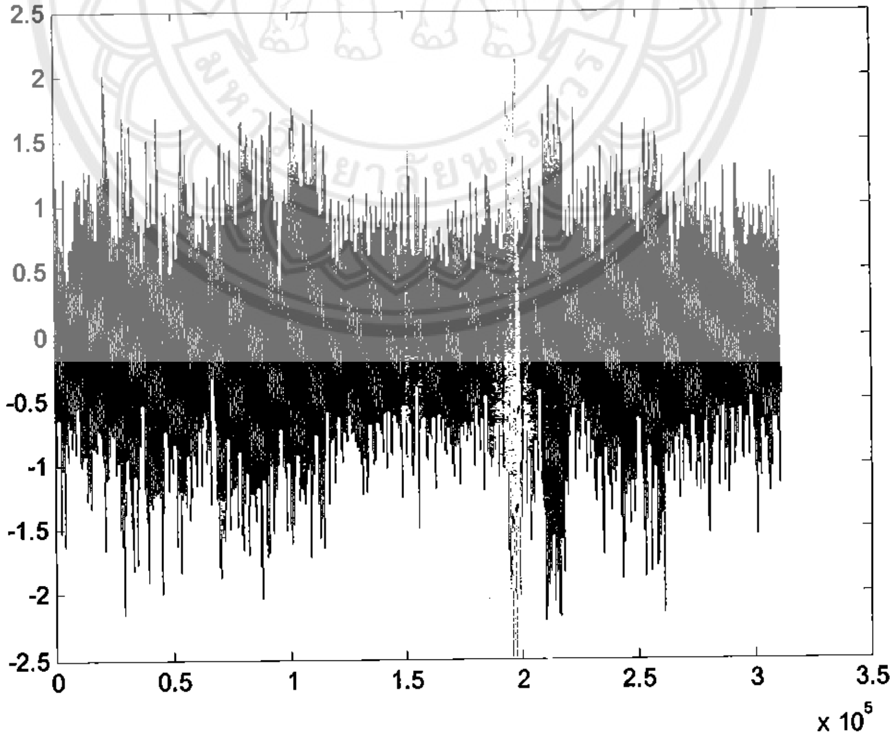
รูปที่ 4.6 สัญญาณเสียงที่ปราศจากสัญญาณรบกวนเมื่อแอมปลิจูดของสัญญาณรบกวนเท่ากับ 0.5



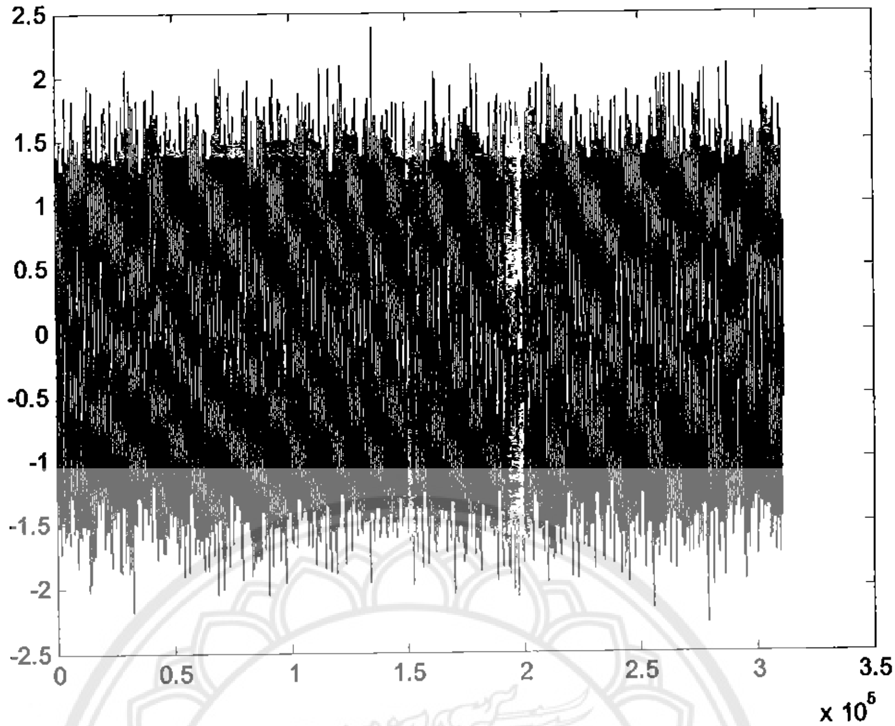
รูปที่ 4.7 สัญญาณเสียงที่รวมกับสัญญาณรบกวนเมื่อแอมปลิจูดของสัญญาณรบกวนเท่ากับ 0.5



รูปที่ 4.8 สัญญาณเสียงที่ถูกกำจัดสัญญาณรบกวน กรณีที่ $a_1 = 1$ ของ Moving Average Filters เมื่อแอมปลิจูดของสัญญาณรบกวนเท่ากับ 0.5



รูปที่ 4.9 สัญญาณเสียงที่ถูกกำจัดสัญญาณรบกวน กรณีที่ $a_1 = 1$ ของ Exponential Filters เมื่อแอมปลิจูดของสัญญาณรบกวนเท่ากับ 0.5



รูปที่ 4.10 สัญญาณเสียงที่ถูกกำจัดสัญญาณรบกวน กรณีที่ $a_1 = 40$ ของ Exponential Filters เมื่อแอมปลิจูดของสัญญาณรบกวนเท่ากับ 0.5

4.2 วิเคราะห์และวิจารณ์ผลการทดลองเมื่อค่าความยาวของฟิลเตอร์เท่ากับ 80 และ แอมปลิจูดของสัญญาณรบกวนเท่ากับ 0.5

จากตารางที่ 4.2 กรณีของ Exponential Filters เมื่อทำการเปรียบเทียบ $a_1 = 1$ และ $a_1 = 40$ โดยการวัดประสิทธิภาพของการลดสัญญาณรบกวนด้วยวิธีค่าเฉลี่ยความคลาดเคลื่อนยกกำลังสอง (Mean Square Error, MSE) พบว่า เมื่อ $a_1 = 40$ ค่า MSE เข้าใกล้ค่าศูนย์มากกว่า กรณีที่ $a_1 = 1$ แสดงว่าเมื่อ $a_1 = 40$ มีประสิทธิภาพในการกำจัดสัญญาณรบกวนได้ดีกว่า $a_1 = 1$ กรณี Moving Average Filters เมื่อทำการวัดประสิทธิภาพของการลดสัญญาณรบกวนด้วยวิธีค่าเฉลี่ยความคลาดเคลื่อนยกกำลังสอง (Mean Square Error, MSE) พบว่าค่า MSE มีค่าใกล้เคียงกับ $a_1 = 1$ กรณี Exponential Filters และเมื่อนำมาเปรียบเทียบกับค่า MSE ของ $a_1 = 40$ กรณี Exponential Filters จะเห็นว่าค่า MSE ของ $a_1 = 40$ กรณี Exponential Filters มีค่าเข้าใกล้ค่าศูนย์มากกว่า แสดงว่า เมื่อ พิจารณาเปรียบเทียบในเชิงตัวเลขแล้ว $a_1 = 40$ กรณี Exponential Filters มีประสิทธิภาพในการกำจัดสัญญาณรบกวนได้ดีกว่าวิธีใดๆดังที่ได้กล่าวมา

จากรูปที่ 4.8 เป็นรูปสัญญาณเสียงที่ถูกกำจัดสัญญาณรบกวน กรณีที่ $a_1 = 1$ ของ Moving Average Filters เมื่อนำมาเปรียบเทียบกับรูปที่ 4.7 คือ สัญญาณเสียงรวมกับรบกวน พบว่า สัญญาณ

รบกวนถูกลดทอนออกไปจากเสียงพูดได้น้อยมาก เนื่องจากค่าแอมปลิจูดของสัญญาณรบกวนมีค่ามากกว่าสัญญาณเสียงค่อนข้างมากเมื่อพิจารณาเปรียบเทียบเชิงรูปภาพ

จากรูปที่ 4.9 เป็นรูปสัญญาณเสียงที่ถูกกำจัดสัญญาณรบกวน กรณีที่ $a_1 = 1$ ของ Exponential Filters เมื่อนำมาเปรียบเทียบกับรูปที่ 4.7 คือ สัญญาณเสียงรวมกับสัญญาณรบกวน พบว่า สัญญาณรบกวนถูกลดทอนออกไปจากเสียงพูดได้น้อยมาก ซึ่งมีลักษณะคล้ายคลึงกับรูปที่ 4.8 เนื่องจากค่าแอมปลิจูดของสัญญาณรบกวนมีค่ามากกว่าสัญญาณเสียงค่อนข้างมาก เมื่อพิจารณาเปรียบเทียบเชิงรูปภาพ

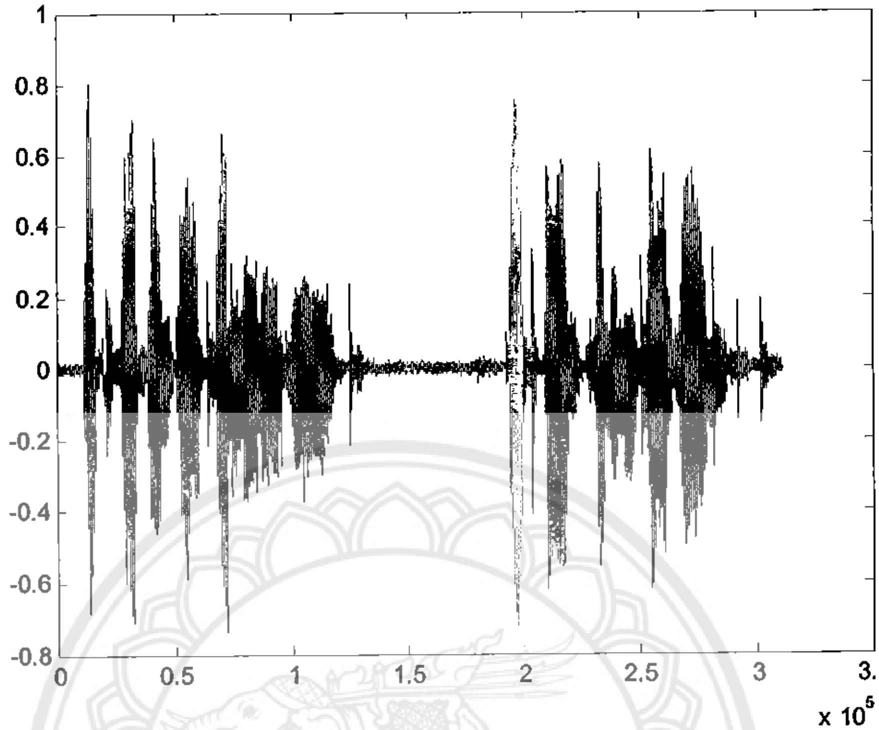
จากรูปที่ 4.10 เป็นรูปสัญญาณเสียงที่ถูกกำจัดสัญญาณรบกวน กรณีที่ $a_1 = 40$ ของ Exponential Filters เมื่อนำมาเปรียบเทียบกับรูปที่ 4.7 คือสัญญาณเสียงรวมกับสัญญาณรบกวนพบว่ามีลักษณะคล้ายคลึงกับรูปที่ 4.7 แสดงว่า สัญญาณรบกวนแทบจะไม่ถูกกำจัดสัญญาณรบกวนออกจากเสียงพูดเลย เมื่อพิจารณาเปรียบเทียบเชิงรูปภาพ

ตารางที่ 4.3 เปรียบเทียบค่า Mean Square Error ของ $a_1 = 1$ และ $a_1 = 40$ ระหว่าง Exponential Filters และ Moving Average Filters โดยกำหนดค่าความยาวของฟิลเตอร์เท่ากับ 80 และ แอมปลิจูดของสัญญาณรบกวนเท่ากับ 0.05

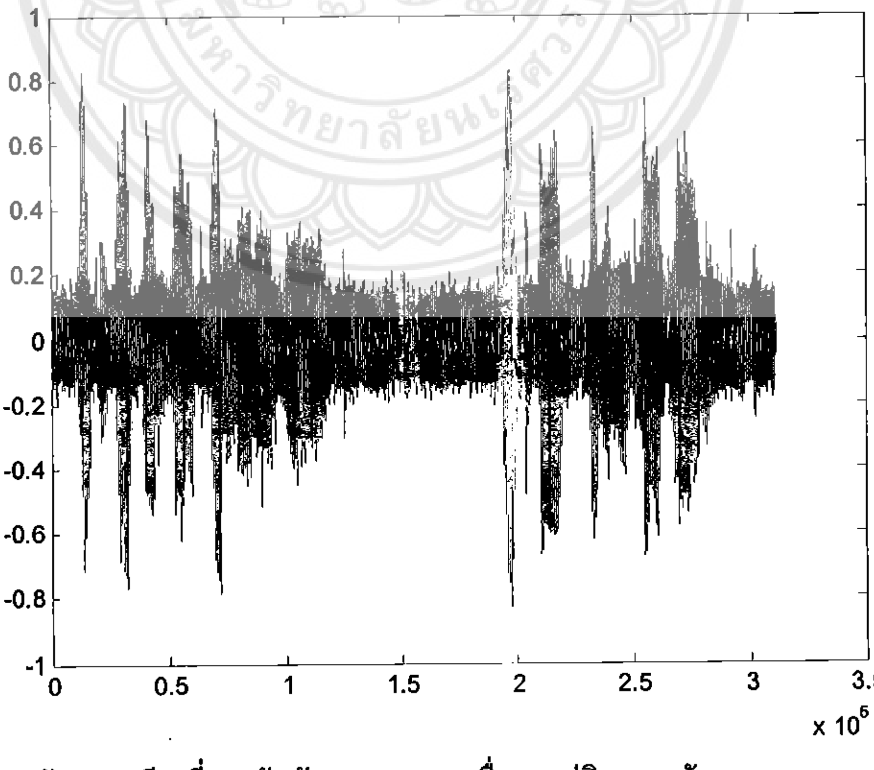
Exponential Filters เมื่อ q=80		Moving Average Filters เมื่อ q=80
$\frac{1}{N}\sum_{n=0}^{N-1}(y[n]-s[n])^2$ ที่ $a_1 = 1$	$\frac{1}{N}\sum_{n=0}^{N-1}(y[n]-s[n])^2$ ที่ $a_1 = 40$	$\frac{1}{N}\sum_{n=0}^{N-1}(y[n]-s[n])^2$
0.0261	0.0024	0.0262
0.0261	0.0024	0.0262
0.0261	0.0024	0.0262
0.0261	0.0024	0.0262
0.0261	0.0024	0.0262
0.0261	0.0024	0.0262
0.0261	0.0024	0.0262
0.0261	0.0024	0.0262
0.0261	0.0024	0.0262
0.0261	0.0024	0.0262
0.0261	0.0024	0.0262
0.0261	0.0024	0.0262
0.0261	0.0024	0.0262
0.0261	0.0024	0.0262
0.0261	0.0024	0.0262
0.0261	0.0024	0.0262
0.0261	0.0024	0.0262
0.0261	0.0024	0.0262
0.0261	0.0024	0.0262
0.0261	0.0024	0.0262

ค่าเฉลี่ย

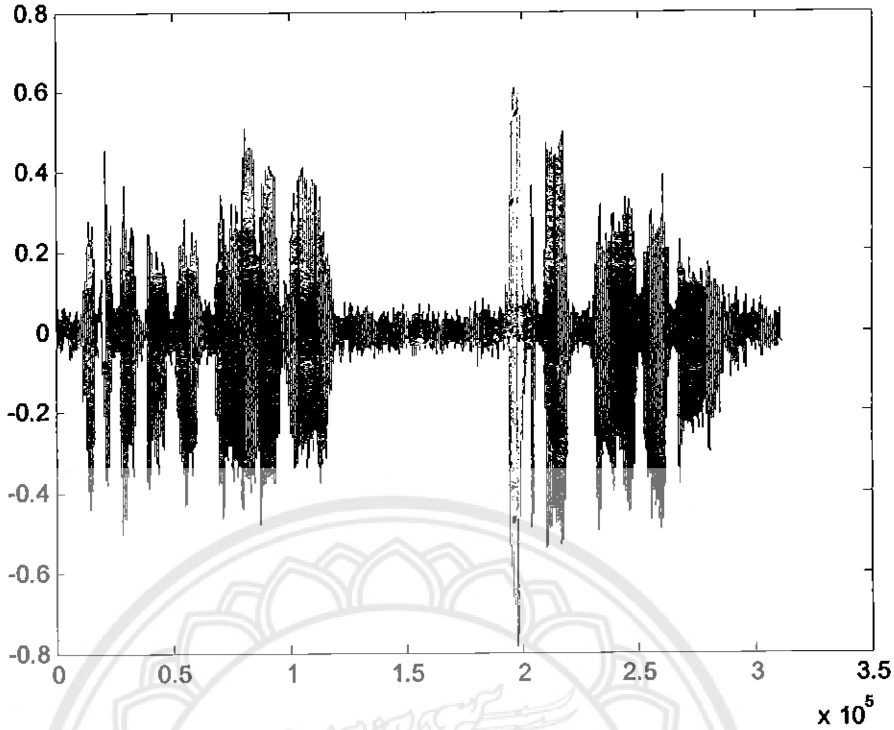
0.0261	0.0024	0.0262



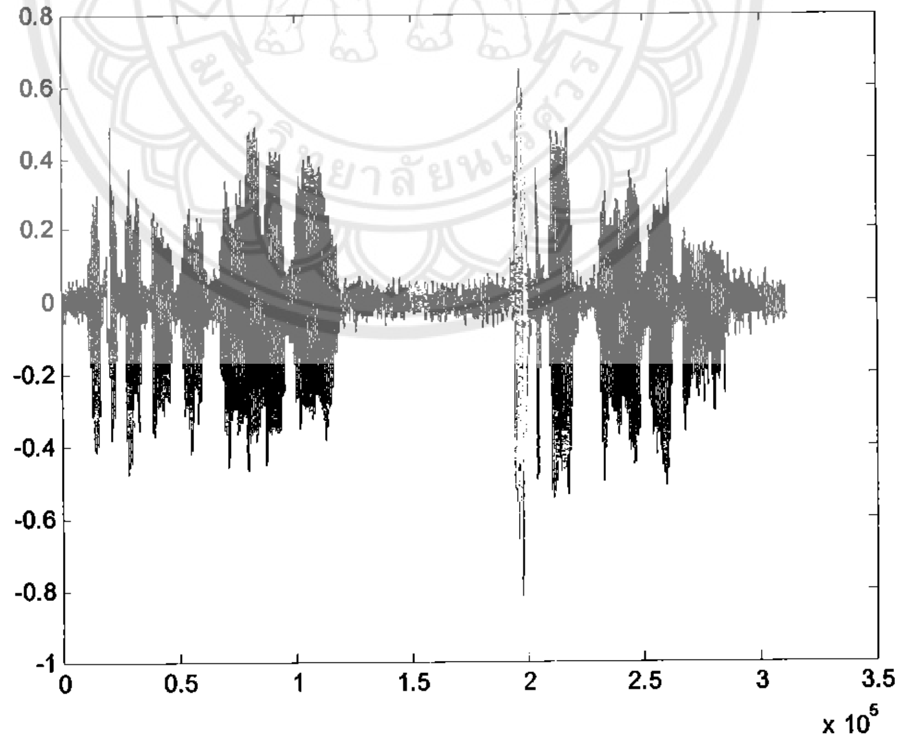
รูปที่ 4.11 สัญญาณเสียงที่ปราศจากสัญญาณรบกวนเมื่อแอมปลิจูดของสัญญาณรบกวนเท่ากับ 0.05



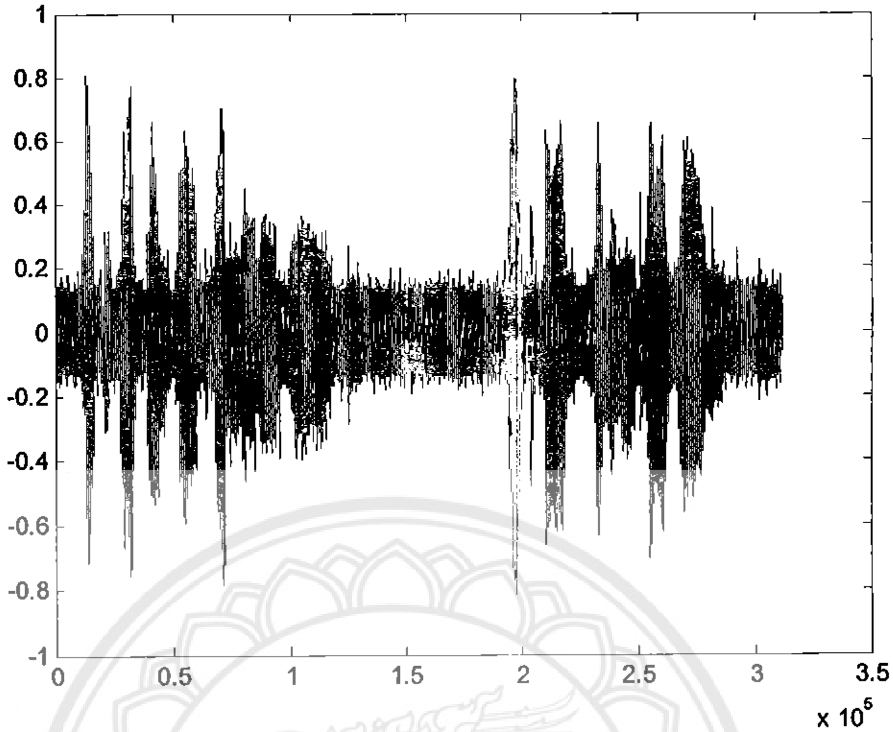
รูปที่ 4.12 สัญญาณเสียงที่รวมกับสัญญาณรบกวนเมื่อแอมปลิจูดของสัญญาณรบกวนเท่ากับ 0.05



รูปที่ 4.13 สัญญาณเสียงที่ถูกกำจัดสัญญาณรบกวน กรณีที่ $a_1 = 1$ ของ Moving Average Filters เมื่อแอมปลิจูดของสัญญาณรบกวนเท่ากับ 0.05



รูปที่ 4.14 สัญญาณเสียงที่ถูกกำจัดสัญญาณรบกวน กรณีที่ $a_1 = 1$ ของ Exponential Filters เมื่อแอมปลิจูดของสัญญาณรบกวนเท่ากับ 0.05



รูปที่ 4.15 สัญญาณเสียงที่ถูกกำจัดสัญญาณรบกวน กรณีที่ $a_1 = 40$ ของ Exponential Filters เมื่อแอมปลิจูดของสัญญาณรบกวนเท่ากับ 0.05

4.3 วิเคราะห์และวิจารณ์ผลการทดลองเมื่อค่าความยาวของฟิลเตอร์เท่ากับ 80 และ แอมปลิจูดของสัญญาณรบกวนเท่ากับ 0.05

จากตารางที่ 4.3 กรณีของ Exponential Filters เมื่อทำการเปรียบเทียบ $a_1 = 1$ และ $a_1 = 40$ โดยการวัดประสิทธิภาพของการลดสัญญาณรบกวนด้วยวิธีค่าเฉลี่ยความคลาดเคลื่อนยกกำลังสอง (Mean Square Error, MSE) พบว่า เมื่อ $a_1 = 40$ ค่า MSE เข้าใกล้ค่าศูนย์มากกว่า กรณีที่ $a_1 = 1$ แสดงว่าเมื่อ $a_1 = 40$ มีประสิทธิภาพในการกำจัดสัญญาณรบกวนได้ดีกว่า $a_1 = 1$ กรณี Moving Average Filters เมื่อทำการวัดประสิทธิภาพของการลดสัญญาณรบกวนด้วยวิธีค่าเฉลี่ยความคลาดเคลื่อนยกกำลังสอง (Mean Square Error, MSE) พบว่าค่า MSE มีค่าใกล้เคียงกับ $a_1 = 1$ กรณี Exponential Filters และเมื่อนำมาเปรียบเทียบกับค่า MSE ของ $a_1 = 40$ กรณี Exponential Filters จะเห็นว่าค่า MSE ของ $a_1 = 40$ กรณี Exponential Filters มีค่าเข้าใกล้ค่าศูนย์มากกว่า แสดงว่า เมื่อ พิจารณาเปรียบเทียบในเชิงตัวเลขแล้ว $a_1 = 40$ กรณี Exponential Filters มีประสิทธิภาพในการกำจัดสัญญาณรบกวนได้ดีกว่าวิธีใดๆดังที่ได้กล่าวมา

จากรูปที่ 4.13 เป็นรูปสัญญาณเสียงที่ถูกกำจัดสัญญาณรบกวน กรณีที่ $a_1 = 1$ ของ Moving Average Filters เมื่อนำมาเปรียบเทียบกับรูปที่ 4.12 คือสัญญาณเสียงรวมกับสัญญาณรบกวน พบว่าสัญญาณรบกวนถูกลดทอนลงไปมากซึ่งมีลักษณะคล้ายคลึงกับรูปที่ 4.11 ซึ่งเป็นรูปสัญญาณเสียงที่

ปราศจากสัญญาณรบกวน ดังนั้นแสดงว่า เมื่อพิจารณาเปรียบเทียบเชิงรูปภาพแล้ว กรณีที่ $a_1 = 1$ ของ Moving Average Filters สามารถกำจัดสัญญาณรบกวนออกจากเสียงพูดได้ดี

จากรูปที่ 4.14 เป็นรูปสัญญาณเสียงที่ถูกกำจัดสัญญาณรบกวน กรณีที่ $a_1 = 1$ ของ Exponential Filters เมื่อนำมาเปรียบเทียบกับรูปที่ 4.12 คือสัญญาณเสียงรวมกับสัญญาณรบกวน พบว่า สัญญาณรบกวนถูกลดทอนลงไปมากซึ่งมีลักษณะคล้ายคลึงกับรูปที่ 4.11 ซึ่งเป็นรูปสัญญาณเสียงที่ปราศจากสัญญาณรบกวน ดังนั้นแสดงว่า เมื่อพิจารณาเปรียบเทียบเชิงรูปภาพแล้ว กรณีที่ $a_1 = 1$ ของ Exponential Filters สามารถกำจัดสัญญาณรบกวนออกจากเสียงพูดได้ดี

จากรูปที่ 4.15 เป็นรูปสัญญาณเสียงที่ถูกกำจัดสัญญาณรบกวน กรณีที่ $a_1 = 40$ ของ Exponential Filters เมื่อนำมาเปรียบเทียบกับรูปที่ 4.12 คือ สัญญาณเสียงรวมกับสัญญาณรบกวน พบว่า มีลักษณะคล้ายคลึงกับรูปที่ 4.12 แสดงว่า สัญญาณรบกวนถูกกำจัดหรือลดทอนออกไปจากเสียงพูดได้น้อยมาก ดังนั้น เมื่อพิจารณาเปรียบเทียบเชิงรูปภาพกับรูปที่ 4.11 คือ สัญญาณเสียงที่ปราศจากสัญญาณรบกวน พบว่า กรณีที่ $a_1 = 40$ ของ Exponential Filters สัญญาณรบกวนถูกกำจัดออกไปจากเสียงพูดได้น้อยมาก

บทที่ 5
สรุปผลการทดลอง

ทำการศึกษาทฤษฎีและหลักการเกี่ยวกับการกำจัดสัญญาณรบกวนในเสียงพูดด้วยวิธี ตัวกรองแบบเรียบค่าเฉลี่ยเคลื่อนที่ (Moving Average Shooting Filters) และ ตัวกรองแบบเรียบ เอ็กซ์โปเนนเชียล (Optimum Exponential Trend Smoothing Filters) โดยการนำข้อมูลเสียงมารวมกับสัญญาณรบกวนแบบเกาส์เซียนไวท์นอยซ์ (Gaussian White Noise) จากนั้นนำมาผ่านตัวกรองแบบเรียบทั้ง 2 วิธี และนำผลที่ที่ได้มาเปรียบเทียบว่า วิธีใดมีประสิทธิภาพในการกำจัดสัญญาณรบกวนได้ดีกว่ากัน โดยพิจารณา จากค่าเฉลี่ยความคลาดเคลื่อนยกกำลังสอง (Mean Square Error, MSE) จากรูปสัญญาณเสียงที่ผ่านตัวกรอง และ จากการฟังเสียงที่ผ่านการกรองสัญญาณรบกวนออกไปแล้ว

ในการประเมินค่าความคลาดเคลื่อนยกกำลังสอง (MSE) ที่จำนวน q เท่ากัน พบว่า $a_1 = 40$ กรณีของ Exponential Filters มีค่า MSE เข้าใกล้ค่าศูนย์มากกว่า $a_1 = 1$ กรณีของ Exponential Filters แสดงให้เห็นถึง ประสิทธิภาพในการกรองสัญญาณรบกวนออกจากเสียงพูด เมื่อ $a_1 = 40$ มีความคลาดเคลื่อนเฉลี่ยในการกรองน้อยกว่า $a_1 = 1$ และเมื่อนำมาเปรียบเทียบกับกรณี Moving Average Filters ค่า MSE ของ $a_1 = 40$ กรณีของ Exponential Filters ยังคงมีค่าเข้าใกล้ค่าศูนย์มากกว่า ดังนั้นเมื่อพิจารณาเปรียบเทียบในเชิงตัวเลข เมื่อ $a_1 = 40$ กรณีของ Exponential Filters มีประสิทธิภาพในการกำจัดสัญญาณรบกวนออกจากเสียงพูดได้ดีกว่าตัวกรองแบบเรียบแบบอื่นๆดังที่ได้กล่าวมา

แต่ในความเป็นจริง จากการฟังเสียงที่ผ่านการกรองสัญญาณรบกวน และ พิจารณาเปรียบเทียบจากรูปสัญญาณเสียงที่ผ่านตัวกรองแบบต่างๆ พบว่า $a_1 = 1$ กรณีของ Exponential Filters และ กรณีของ Moving Average Filters มีประสิทธิภาพในการกำจัดสัญญาณรบกวนออกไปจากเสียงพูดได้ดีกว่า $a_1 = 40$ กรณีของ Exponential Filters

เอกสารอ้างอิง

[1] Robert J.Schilling.Sandra L.Harris. **Digital Signal Processing.** United States of America: Nelson. 2005.

[2] S. Yammen and J.A Cadzow. **"Optimal Linear Trend Smoothing Filters".** "Proceeding of the 29th Electrical Engineering Conference ," Pattaya,Chonburi, Thailand. Volume 2, November 9-10,2006, PP.917-920.

[3] มนัส สังวรศิลป์, วรรัตน์ ภัทรอมรกุล. **คู่มือการใช้งาน Matlab** . กรุงเทพ:อินโฟเพรส. 2543

[4] ไม่ปรากฏชื่อผู้แต่ง. "Norm". [Online]. Available : http://en.wikipedia.org/wiki/Norm_%28mathematics%29

ภาคผนวก ก

โปรแกรมการกรองสัญญาณรบกวนออกจากเสียงพูดโดยใช้ MATLAB

1. กรณีที่แอมปลิจูดของสัญญาณรบกวนเป็น 0.1

1.1 Moving Average Filters กรณีที่ q=80

```
clear all
clc
% Data sequence x[n] = s[n] + w[n]
[s,fs,NBITS]=wavread('data.wav');
x = s + 0.1*randn(size(s));
EnergyX = norm(x,2)^2;
%------------1--------------------
sound(x, fs);
disp('OK Enter to Continue');
pause;
% Moving Average Model
q = 80;      %length of Filter
% Filter selection
Nx = length(x);
hn_ma = ones(1,q)/q;
for kk=1: 1: 1;
   % response of the filter
   tp = conv(x, hn_ma);
   yn_ma = tp(1: Nx);
   yn_ma = sqrt(EnergyX)*yn_ma/norm(yn_ma,2);
   sound(yn_ma, fs);
   disp(['OK Enter to Round : ' num2str(kk)]);
   pause;
end
MSE=sum((yn_ma-s).^2)/Nx
```

อธิบายการทำงานของโปรแกรม Moving Average Filters กรณีที่ q=80

[s,fs,NBITS]=wavread('data.wav');

คือคำสั่งในการจำลองค่าเวกเตอร์ของสัญญาณเสียงที่ชื่อ data.wav มาไว้ในตัวแปร s

x = s + 0.1*randn(size(s));

ให้ตัวแปร x มีค่าเท่ากับ เวกเตอร์ของสัญญาณเสียงเริ่มต้นรวมกับ Gaussian White Noise ดังที่กล่าวไว้ในบทที่ 2 สมการที่ (2.3) ดังนั้น x คือสัญญาณเสียงที่รวมกับสัญญาณรบกวน โดยที่ขนาดของแอมปลิจูดของสัญญาณรบกวนนั้นขึ้นอยู่กับค่าที่คูณเข้าไปใน randn (ในที่นี้ใช้ 0.1)

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[\frac{-(x-\mu)^2}{\sigma^2}\right]$$

EnergyX = norm(x,2)^2;

คือการหาค่าขนาดของเวกเตอร์ x โดยใช้คำสั่ง norm มาเก็บค่าไว้ในตัวแปรชื่อ EnergyX เพื่อนำไปใช้ในการรักษาขนาดของเสียง โดยกล่าวไว้ในสมการที่ (3.3) ดังนี้

$$\|\underline{x}\|_2 = ((x[0])^2 + (x[1])^2 + \ldots + (x[q])^2)^{\frac{1}{2}}$$

$$\|\underline{x}\|_2 = (\sum_{i=0}^{q} |x[i]|^2)^{\frac{1}{2}}$$

sound(x, fs);

คำสั่งในการนำเวกเตอร์ x มาแสดงในรูปแบบของเสียง

Nx = length(x);

นำขนาดข้อมูลของเวกเตอร์ x มาเก็บไว้ในตัวแปร Nx

hn_ma = ones(1,q)/q;

คือการนำสมการของ b_k° มาไว้ในตัวแปร hn_ma ซึ่งทำหน้าที่เป็นตัวกรองดังสมการที่ (2.26)

$$b_k^\circ = \frac{1}{q+1}$$

tp = conv(x, hn_ma);

คือการนำเวกเตอร์ x มาคอลโวลูชั่นกับ hn_ma ที่มีหน้าที่เป็นตัวกรองแล้วเก็บค่าที่ได้ไว้ในตัวแปร tp โดยคุณสมบัติของการทำคอลโวลูชั่นคือ

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

ทำหน้าที่เดียวกับ

$$Y[z] = X[z]H[z]$$

และ

$$y[n] = Z^{-1}\{Y(z)\}$$

นั่นคือ สมบัติของการทำคอลโวลูชั่นจะเทียบได้กับการคูณของซีทรานฟอร์มนั่นเอง

yn_ma = tp(1: Nx);

คือการปรับขนาดของ tp ให้มีขนาดเท่ากับข้อมูลเริ่มต้น ที่เก็บไว้ใน Nx จากนั้นนำค่าที่ได้มาเก็บไว้ในตัวแปร yn_ma

yn_ma = sqrt(EnergyX)*yn_ma/norm(yn_ma,2);

ปรับขนาดเสียงของ yn_ma ให้มีขนาดเสียงเท่ากับขนาดเสียงก่อนที่จะนำไปเข้าตัวกรองโดยนำเวกเตอร์ที่จะปรับขนาดเสียงหารด้วยขนาดของเวกเตอร์นั้นซึ่งจะได้เวกเตอร์หนึ่งหน่วยมาคูณกับขนาดของเวกเตอร์ต้นแบบ (ในที่นี้คือ EnergyX) ทำให้ขนาดเสียงดังเท่ากับเสียงต้นแบบก่อนผ่านตัวกรองดังที่กล่าวไว้ในสมการที่ (3.5) ดังนี้

$$yma[n] = \frac{y[n]}{\|\underline{y}\|_2} \times \|\underline{x}\|_2$$

MSE=sum((yn_ma-s).^2)/Nx

คือการหาค่าเฉลี่ยความคลาดเคลื่อนยกกำลังสอง (Mean Square Error, MSE) เพื่อวัดประสิทธิภาพความคลาดเคลื่อน โดยกล่าวไว้ในสมการที่ (2.48) คือ

$$MSE = \frac{1}{N}\sum_{n=0}^{N-1}(y[n] - s[n])^2$$

1.2 Exponential Filters กรณีที่ $a1 = 1$ และ q=80

```
clear all
clc
% Data sequence x[n] = s[n] + w[n]
[s,fs,NBITS]=wavread('data.wav');
x = s + 0.1*randn(size(s));
EnergyX = norm(x,2)^2;
%------------1-------------------
sound(x, fs);
disp('OK Enter to Continue');
pause;
% Exponentail Model
q = 80;       %length of Filter
a1 =1  ;    % Exponential constant
% Filter selection
k = 0: 1: q;
Nx = length(x);
hn_et = (a1.^(-k))/sum(a1.^(-2*k));
for kk=1: 1: 1;
   % response of the filter
   tp = conv(x, hn_et);
   yn_et = tp(1: Nx);
   yn_et = sqrt(EnergyX)*yn_et/norm(yn_et,2);
   sound(yn_et, fs);
   disp(['OK Enter to Round : ' num2str(kk)]);
   pause;
end
MSE1=sum((yn_et-s).^2)/Nx
```

อธิบายการทำงานของโปรแกรม Exponential Filters กรณีที่ $a1 = 1$ และ q=80

[s,fs,NBITS]=wavread('data.wav');

คือคำสั่งในการจำลองค่าเวกเตอร์ของสัญญาณเสียงที่ชื่อ data.wav มาไว้ในตัวแปร s

x = s + 0.1*randn(size(s));

ให้ตัวแปร x มีค่าเท่ากับ เวกเตอร์ของสัญญาณเสียงเริ่มต้นรวมกับ Gaussian White Noise ดังที่กล่าวไว้ในบทที่ 2 สมการที่ (2.3) ดังนั้น x คือสัญญาณเสียงที่รวมกับสัญญาณรบกวน โดยที่ขนาดของแอมปลิจูดของสัญญาณรบกวนนั้นขึ้นอยู่กับค่าที่คูณเข้าไปใน randn (ในที่นี้ใช้ 0.1)

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[\frac{-(x-\mu)^2}{\sigma^2}\right]$$

EnergyX = norm(x,2)^2;

คือการหาค่าขนาดของเวกเตอร์ x โดยใช้คำสั่ง norm มาเก็บค่าไว้ในตัวแปรชื่อ EnergyX เพื่อนำไปใช้ในการรักษาขนาดของเสียง โดยกล่าวไว้ในสมการที่ (3.3)ดังนี้

$$\|\underline{x}\|_2 = ((x[0])^2 + (x[1])^2 + \ldots + (x[q])^2)^{\frac{1}{2}}$$

$$\|x\|_2 = (\sum_{i=0}^{q} |x[i]|^2)^{\frac{1}{2}}$$

sound(x, fs);

คำสั่งในการนำเวกเตอร์ x มาแสดงในรูปแบบของเสียง

k = 0: 1: q;

นำค่า k ไปใช้ในสมการที่ (2.44) โดยให้ k มีค่าเท่ากับ 0 และเพิ่มขึ้นทีละ 1 จนถึง q

$$b_k^{\circ} = \frac{1}{\sum_{k=0}^{q} (a_1)^{-2k}} (a_1)^{-k}$$

Nx = length(x);

นำขนาดข้อมูลของเวกเตอร์ x มาเก็บไว้ในตัวแปร Nx

hn_et = (a1.^(-k))/sum(a1.^(-2*k));

คือการนำสมการของ b_k° มาไว้ในตัวแปร hn_et ซึ่งทำหน้าที่เป็นตัวกรองดังสมการที่ 2.44 ดังนี้

$$b_k^{\circ} = \frac{1}{\sum_{k=0}^{q}(a_1)^{-2k}}(a_1)^{-k}$$

tp = conv(x, hn_ma);

คือการนำเวกเตอร์ x มาคอลโวลูชั่นกับ hn_ma ที่มีหน้าที่เป็นตัวกรองแล้วเก็บค่าที่ได้ไว้ในตัวแปร tp โดยคุณสมบัติของการทำคอลโวลูชั่นคือ

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

ทำหน้าที่เดียวกับ

$$Y[z] = X[z]H[z]$$

และ

$$y[n] = Z^{-1}\{Y(z)\}$$

นั่นคือ สมบัติของการทำคอลโวลูชั่นจะเทียบได้กับการคูณของซีทรานฟอร์มนั่นเอง

yn_ma = tp(1: Nx);

คือการปรับขนาดของ tp ให้มีขนาดเท่ากับข้อมูลเริ่มต้น ที่เก็บไว้ใน Nx จากนั้นนำค่าที่ได้มาเก็บไว้ในตัวแปร yn_ma

yn_ma = sqrt(EnergyX)*yn_ma/norm(yn_ma,2);

ปรับขนาดเสียงของ yn_ma ให้มีขนาดเสียงเท่ากับขนาดเสียงก่อนที่จะนำไปเข้าตัวกรองโดยนำเวกเตอร์ที่จะปรับขนาดเสียงหารด้วยขนาดของเวกเตอร์นั้นซึ่งจะได้เวกเตอร์หนึ่งหน่วยมาคูณกับขนาดของเวกเตอร์ต้นแบบ (ในที่นี้คือ EnergyX) ทำให้ขนาดเสียงดังเท่ากับเสียงต้นแบบก่อนผ่านตัวกรองดังที่กล่าวไว้ในสมการที่ (3.5) ดังนี้

$$yma[n] = \frac{y[n]}{\|\underline{y}\|_2} \times \|\underline{x}\|_2$$

MSE=sum((yn_ma-s).^2)/Nx

คือการหาค่าเฉลี่ยความคลาดเคลื่อนยกกำลังสอง (Mean Square Error, MSE) เพื่อวัดประสิทธิภาพความคลาดเคลื่อน โดยกล่าวไว้ในสมการที่ (2.48) คือ

$$MSE = \frac{1}{N}\sum_{n=0}^{N-1}(y[n] - s[n])^2$$

1.3 Exponential Filters กรณีที่ $a1 = 40$ และ q=80

```
clear all
clc
% Data sequence x[n] = s[n] + w[n]
[s,fs,NBITS]=wavread('data.wav');
x = s + 0.1*randn(size(s));
EnergyX = norm(x,2)^2;
%------------1-------------------
sound(x, fs);
disp('OK Enter to Continue');
pause;
% Exponentail Model
q = 80;         %length of Filter
a1 =40  ;     % Exponential constant
% Filter selection
k = 0: 1: q;
Nx = length(x);
hn_et = (a1.^(-k))/sum(a1.^(-2*k));
for kk=1: 1: 1;
   % response of the filter
   tp = conv(x, hn_et);
   yn_et = tp(1: Nx);
   yn_et = sqrt(EnergyX)*yn_et/norm(yn_et,2);
   sound(yn_et, fs);
   disp(['OK Enter to Round : ' num2str(kk)]);
   pause;
end
MSE1=sum((yn_et-s).^2)/Nx
```

อธิบายการทำงานของโปรแกรม Exponential Filters กรณีที่ $a1 = 40$ และ q=80

[s,fs,NBITS]=wavread('data.wav');

คือคำสั่งในการจำลองค่าเวกเตอร์ของสัญญาณเสียงที่ชื่อ data.wav มาไว้ในตัวแปร s

x = s + 0.1*randn(size(s));

ให้ตัวแปร x มีค่าเท่ากับ เวกเตอร์ของสัญญาณเสียงเริ่มต้นรวมกับ Gaussian White Noise ดังที่กล่าวไว้ในบทที่ 2 สมการที่ (2.3) ดังนั้น x คือสัญญาณเสียงที่รวมกับสัญญาณรบกวน โดยที่ขนาดของแอมปลิจูดของสัญญาณรบกวนนั้นขึ้นอยู่กับค่าที่คูณเข้าไปใน randn (ในที่นี้ใช้ 0.1)

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[\frac{-(x-\mu)^2}{\sigma^2}\right]$$

EnergyX = norm(x,2)^2;

คือการหาค่าขนาดของเวกเตอร์ x โดยใช้คำสั่ง norm มาเก็บค่าไว้ในตัวแปรชื่อ EnergyX เพื่อนำไปใช้ในการรักษาขนาดของเสียง โดยกล่าวไว้ในสมการที่ (3.3) ดังนี้

$$\|\underline{x}\|_2 = ((x[0])^2 + (x[1])^2 + \ldots + (x[q])^2)^{\frac{1}{2}}$$

$$\|\underline{x}\|_2 = (\sum_{i=0}^{q} |x[i]|^2)^{\frac{1}{2}}$$

sound(x, fs);

คำสั่งในการนำเวกเตอร์ x มาแสดงในรูปแบบของเสียง

k = 0: 1: q;

นำค่า k ไปใช้ในสมการที่ (2.44) โดยให้ k มีค่าเท่ากับ 0 และเพิ่มขึ้นทีละ 1 จนถึง q

$$b_k^{\circ} = \frac{1}{\sum_{k=0}^{q} (a_1)^{-2k}} (a_1)^{-k}$$

Nx = length(x);

นำขนาดข้อมูลของเวกเตอร์ x มาเก็บไว้ในตัวแปร Nx

hn_et = (a1.^(-k))/sum(a1.^(-2*k));

คือการนำสมการของ b_k° มาไว้ในตัวแปร hn_et ซึ่งทำหน้าที่เป็นตัวกรองดังสมการที่ (2.44)

$$b_k^{\circ} = \frac{1}{\sum_{k=0}^{q}(a_1)^{-2k}}(a_1)^{-k}$$

tp = conv(x, hn_ma);

คือการนำเวกเตอร์ x มาคอลโวลูชั่นกับ hn_ma ที่มีหน้าที่เป็นตัวกรองแล้วเก็บค่าที่ได้ไว้ในตัวแปร tp โดยคุณสมบัติของการทำคอลโวลูชั่นคือ

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

ทำหน้าที่เดียวกับ

$$Y[z] = X[z]H[z]$$

และ

$$y[n] = Z^{-1}\{Y(z)\}$$

นั่นคือ สมบัติของการทำคอลโวลูชั่นจะเทียบได้กับการคูณของซีทรานฟอร์มนั่นเอง

yn_ma = tp(1: Nx);

คือการปรับขนาดของ tp ให้มีขนาดเท่ากับข้อมูลเริ่มต้น ที่เก็บไว้ใน Nx จากนั้นนำค่าที่ได้มาเก็บไว้ในตัวแปร yn_ma

yn_ma = sqrt(EnergyX)*yn_ma/norm(yn_ma,2);

ปรับขนาดเสียงของ yn_ma ให้มีขนาดเสียงเท่ากับขนาดเสียงก่อนที่จะนำไปเข้าตัวกรองโดยนำเวกเตอร์ที่จะปรับขนาดเสียงหารด้วยขนาดของเวกเตอร์นั้นซึ่งจะได้เวกเตอร์หนึ่งหน่วยมาคูณกับขนาดของเวกเตอร์ต้นแบบ(ในที่นี้คือ EnergyX)ทำให้ขนาดเสียงดังเท่ากับเสียงต้นแบบก่อนผ่านตัวกรอง ดังที่กล่าวไว้ในสมการที่ (3.5) ดังนี้

$$yma[n] = \frac{y[n]}{\|\underline{y}\|_2} \times \|\underline{x}\|_2$$

MSE=sum((yn_ma-s).^2)/Nx

คือการหาค่าเฉลี่ยความคลาดเคลื่อนยกกำลังสอง (Mean Square Error, MSE) เพื่อวัดประสิทธิภาพความคลาดเคลื่อน โดยกล่าวไว้ในสมการที่ (2.48) คือ

$$MSE = \frac{1}{N}\sum_{n=0}^{N-1}(y[n]-s[n])^2$$

2. กรณีที่แอมปลิจูดของสัญญาณรบกวนเป็น 0.5

2.1 Moving Average Filters กรณีที่ q=80

```
clear all
clc
% Data sequence x[n] = s[n] + w[n]
[s,fs,NBITS]=wavread('data.wav');
x = s + 0.5*randn(size(s));
EnergyX = norm(x,2)^2;
%------------1--------------------
sound(x, fs);
disp('OK Enter to Continue');
pause;
% Moving Average Model
q = 80;     %length of Filter
% Filter selection
Nx = length(x);
hn_ma = ones(1,q)/q;
for kk=1: 1: 1;
   % response of the filter
   tp = conv(x, hn_ma);
   yn_ma = tp(1: Nx);
   yn_ma = sqrt(EnergyX)*yn_ma/norm(yn_ma,2);
   sound(yn_ma, fs);
   disp(['OK Enter to Round : ' num2str(kk)]);
   pause;
end
MSE=sum((yn_ma-s).^2)/Nx
```

อธิบายการทำงานของโปรแกรม Moving Average Filters กรณีที่ q=80

[s,fs,NBITS]=wavread('data.wav');

คือคำสั่งในการจำลองค่าเวกเตอร์ของสัญญาณเสียงที่ชื่อ data.wav มาไว้ในตัวแปร s

x = s + 0.5*randn(size(s));

ให้ตัวแปร x มีค่าเท่ากับ เวกเตอร์ของสัญญาณเสียงเริ่มต้นรวมกับ Gaussian White Noise ดังที่กล่าวไว้ในบทที่ 2 สมการที่ (2.3) ดังนั้น x คือสัญญาณเสียงที่รวมกับสัญญาณรบกวน โดยที่ขนาดของแอมปลิจูดของสัญญาณรบกวนนั้นขึ้นอยู่กับค่าที่คูณเข้าไปใน randn (ในที่นี้ใช้ 0.5)

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[\frac{-(x-\mu)^2}{\sigma^2}\right]$$

EnergyX = norm(x,2)^2;

คือการหาค่าขนาดของเวกเตอร์ x โดยใช้คำสั่ง norm มาเก็บค่าไว้ในตัวแปรชื่อ EnergyX เพื่อนำไปใช้ในการรักษาขนาดของเสียง โดยกล่าวไว้ในสมการที่ (3.3) ดังนี้

$$\|\underline{x}\|_2 = ((x[0])^2 + (x[1])^2 + \ldots + (x[q])^2)^{\frac{1}{2}}$$

$$\|\underline{x}\|_2 = (\sum_{i=0}^{q} |x[i]|^2)^{\frac{1}{2}}$$

sound(x, fs);

คำสั่งในการนำเวกเตอร์ x มาแสดงในรูปแบบของเสียง

Nx = length(x);

นำขนาดข้อมูลของเวกเตอร์ x มาเก็บไว้ในตัวแปร Nx

hn_ma = ones(1,q)/q;

คือการนำสมการของ b_k° มาไว้ในตัวแปร hn_ma ซึ่งทำหน้าที่เป็นตัวกรองดังสมการที่ (2.26)

$$b_k^\circ = \frac{1}{q+1}$$

tp = conv(x, hn_ma);

คือการนำเวกเตอร์ x มาคอลโวลูชั่นกับ hn_ma ที่มีหน้าที่เป็นตัวกรองแล้วเก็บค่าที่ได้ไว้ในตัวแปร tp โดยคุณสมบัติของการทำคอลโวลูชั่นคือ

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

ทำหน้าที่เดียวกับ

$$Y[z] = X[z]H[z]$$

และ

$$y[n] = Z^{-1}\{Y(z)\}$$

นั่นคือ สมบัติของการทำคอลโวลูชั่นจะเทียบได้กับการคูณของซีทรานฟอร์มนั่นเอง

yn_ma = tp(1: Nx);

คือการปรับขนาดของ tp ให้มีขนาดเท่ากับข้อมูลเริ่มต้น ที่เก็บไว้ใน Nx จากนั้นนำค่าที่ได้มาเก็บไว้ในตัวแปร yn_ma

yn_ma = sqrt(EnergyX)*yn_ma/norm(yn_ma,2);

ปรับขนาดเสียงของ yn_ma ให้มีขนาดเสียงเท่ากับขนาดเสียงก่อนที่จะนำไปเข้าตัวกรองโดยนำเวกเตอร์ที่จะปรับขนาดเสียงหารด้วยขนาดของเวกเตอร์นั้นซึ่งจะได้เวกเตอร์หนึ่งหน่วยมาคูณกับขนาดของเวกเตอร์ต้นแบบ(ในที่นี้คือ EnergyX)ทำให้ขนาดเสียงดังเท่ากับเสียงต้นแบบก่อนผ่านตัวกรอง ดังที่กล่าวไว้ในสมการที่ (3.5) ดังนี้

$$yma[n] = \frac{y[n]}{\|\underline{y}\|_2} \times \|\underline{x}\|_2$$

MSE=sum((yn_ma-s).^2)/Nx

คือการหาค่าเฉลี่ยความคลาดเคลื่อนยกกำลังสอง (Mean Square Error, MSE) เพื่อวัดประสิทธิภาพความคลาดเคลื่อน โดยกล่าวไว้ในสมการที่ (2.48) คือ

$$MSE = \frac{1}{N}\sum_{n=0}^{N-1}(y[n]-s[n])^2$$

2.1 Exponential Filters กรณีที่ $a1 = 1$ และ q=80

```
clear all
clc
% Data sequence x[n] = s[n] + w[n]
[s,fs,NBITS]=wavread('data.wav');
x = s + 0.5*randn(size(s));
EnergyX = norm(x,2)^2;
%------------1--------------------
sound(x, fs);
disp('OK Enter to Continue');
pause;
% Exponentail Model
q = 80;        %length of Filter
a1 =1  ;     % Exponential constant
% Filter selection
k = 0: 1: q;
Nx = length(x);
hn_et = (a1.^(-k))/sum(a1.^(-2*k));
for kk=1: 1: 1;
   % response of the filter
   tp = conv(x, hn_et);
   yn_et = tp(1: Nx);
   yn_et = sqrt(EnergyX)*yn_et/norm(yn_et,2);
   sound(yn_et, fs);
   disp(['OK Enter to Round : ' num2str(kk)]);
   pause;
end
MSE1=sum((yn_et-s).^2)/Nx
```

อธิบายการทำงานของโปรแกรม Exponential Filters กรณีที่ $a1 = 1$ และ q=80

[s,fs,NBITS]=wavread('data.wav');

คือคำสั่งในการจำลองค่าเวกเตอร์ของสัญญาณเสียงที่ชื่อ data.wav มาไว้ในตัวแปร s

x = s + 0.5*randn(size(s));

ให้ตัวแปร x มีค่าเท่ากับ เวกเตอร์ของสัญญาณเสียงเริ่มต้นรวมกับ Gaussian White Noise ดังที่กล่าวไว้ในบทที่ 2 สมการที่ (2.3) ดังนั้น x คือสัญญาณเสียงที่รวมกับสัญญาณรบกวน โดยที่ขนาดของแอมปลิจูดของสัญญาณรบกวนนั้นขึ้นอยู่กับค่าที่คูณเข้าไปใน randn (ในที่นี้ใช้ 0.5)

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[\frac{-(x-\mu)^2}{\sigma^2}\right]$$

EnergyX = norm(x,2)^2;

คือการหาค่าขนาดของเวกเตอร์ x โดยใช้คำสั่ง norm มาเก็บค่าไว้ในตัวแปรชื่อ EnergyX เพื่อนำไปใช้ในการรักษาขนาดของเสียง โดยกล่าวไว้ในสมการที่ (3.3) ดังนี้

$$\|\underline{x}\|_2 = ((x[0])^2 + (x[1])^2 + \ldots + (x[q])^2)^{\frac{1}{2}}$$

$$\|\underline{x}\|_2 = (\sum_{i=0}^{q} |x[i]|^2)^{\frac{1}{2}}$$

sound(x, fs);

คำสั่งในการนำเวกเตอร์ x มาแสดงในรูปแบบของเสียง

k = 0: 1: q;

นำค่า k ไปใช้ในสมการที่ (2.44) โดยให้ k มีค่าเท่ากับ 0 และเพิ่มขึ้นทีละ 1 จนถึง q

$$b_k^{\circ} = \frac{1}{\sum_{k=0}^{q} (a_1)^{-2k}} (a_1)^{-k}$$

Nx = length(x);

นำขนาดข้อมูลของเวกเตอร์ x มาเก็บไว้ในตัวแปร Nx

hn_et = (a1.^(-k))/sum(a1.^(-2*k));

คือการนำสมการของ b_k° มาไว้ในตัวแปร hn_et ซึ่งทำหน้าที่เป็นตัวกรองดังสมการ (2.44)

$$b_k^\circ = \frac{1}{\sum_{k=0}^{q}(a_1)^{-2k}}(a_1)^{-k}$$

tp = conv(x, hn_ma);

คือการนำเวกเตอร์ x มาคอลโวลูชั่นกับ hn_ma ที่มีหน้าที่เป็นตัวกรองแล้วเก็บค่าที่ได้ไว้ในตัวแปร tp โดยคุณสมบัติของการทำคอลโวลูชั่นคือ

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

ทำหน้าที่เดียวกับ

$$Y[z] = X[z]H[z]$$

และ

$$y[n] = Z^{-1}\{Y(z)\}$$

นั่นคือ สมบัติของการทำคอลโวลูชั่นจะเทียบได้กับการคูณของซีทรานฟอร์มนั่นเอง

yn_ma = tp(1: Nx);

คือการปรับขนาดของ tp ให้มีขนาดเท่ากับข้อมูลเริ่มต้น ที่เก็บไว้ใน Nx จากนั้นนำค่าที่ได้มาเก็บไว้ในตัวแปร yn_ma

yn_ma = sqrt(EnergyX)*yn_ma/norm(yn_ma,2);

ปรับขนาดเสียงของ yn_ma ให้มีขนาดเสียงเท่ากับขนาดเสียงก่อนที่จะนำไปเข้าตัวกรองโดยนำเวกเตอร์ที่จะปรับขนาดเสียงหารด้วยขนาดของเวกเตอร์นั้นซึ่งจะได้เวกเตอร์หนึ่งหน่วยมาคูณกับขนาดของเวกเตอร์ต้นแบบ (ในที่นี้คือ EnergyX) ทำให้ขนาดเสียงดังเท่ากับเสียงต้นแบบก่อนผ่านตัวกรองดังที่กล่าวไว้ในสมการที่ (3.5) ดังนี้

$$yma[n] = \frac{y[n]}{\|\underline{y}\|_2} \times \|\underline{x}\|_2$$

MSE=sum((yn_ma-s).^2)/Nx

คือการหาค่าเฉลี่ยความคลาดเคลื่อนยกกำลังสอง (Mean Square Error, MSE) เพื่อวัดประสิทธิภาพความคลาดเคลื่อน โดยกล่าวไว้ในสมการที่ (2.48) คือ

$$MSE = \frac{1}{N}\sum_{n=0}^{N-1}(y[n]-s[n])^2$$

2.2 Exponential Filters กรณีที่ $a1 = 40$ และ q=80

```
clear all
clc
% Data sequence x[n] = s[n] + w[n]
[s,fs,NBITS]=wavread('data.wav');
x = s + 0.5*randn(size(s));
EnergyX = norm(x,2)^2;
%------------1--------------------
sound(x, fs);
disp('OK Enter to Continue');
pause;
% Exponentail Model
q = 80;         %length of Filter
a1 =40  ;      % Exponential constant
% Filter selection
k = 0: 1: q;
Nx = length(x);
hn_et = (a1.^(-k))/sum(a1.^(-2*k));
for kk=1: 1: 1;
   % response of the filter
   tp = conv(x, hn_et);
   yn_et = tp(1: Nx);
   yn_et = sqrt(EnergyX)*yn_et/norm(yn_et,2);
   sound(yn_et, fs);
   disp(['OK Enter to Round : ' num2str(kk)]);
   pause;
end
MSE1=sum((yn_et-s).^2)/Nx
```

อธิบายการทำงานของโปรแกรม Exponential Filters กรณีที่ $a1 = 40$ และ q=80

[s,fs,NBITS]=wavread('data.wav');

คือคำสั่งในการจำลองค่าเวกเตอร์ของสัญญาณเสียงที่ชื่อ data.wav มาไว้ในตัวแปร s

x = s + 0.5*randn(size(s));

ให้ตัวแปร x มีค่าเท่ากับ เวกเตอร์ของสัญญาณเสียงเริ่มต้นรวมกับ Gaussian White Noise ดังที่กล่าวไว้ในบทที่ 2 สมการที่ (2.3) ดังนั้น x คือสัญญาณเสียงที่รวมกับสัญญาณรบกวน โดยที่ขนาดของแอมปลิจูดของสัญญาณรบกวนนั้นขึ้นอยู่กับค่าที่คูณเข้าไปใน randn (ในที่นี้ใช้ 0.5)

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[\frac{-(x-\mu)^2}{\sigma^2}\right]$$

EnergyX = norm(x,2)^2;

คือการหาค่าขนาดของเวกเตอร์ x โดยใช้คำสั่ง norm มาเก็บค่าไว้ในตัวแปรชื่อ EnergyX เพื่อนำไปใช้ในการรักษาขนาดของเสียง โดยกล่าวไว้ในสมการที่ (3.3) ดังนี้

$$\|\underline{x}\|_2 = ((x[0])^2 + (x[1])^2 + \ldots + (x[q])^2)^{\frac{1}{2}}$$
$$\|\underline{x}\|_2 = (\sum_{i=0}^{q} |x[i]|^2)^{\frac{1}{2}}$$

sound(x, fs);

คำสั่งในการนำเวกเตอร์ x มาแสดงในรูปแบบของเสียง

k = 0: 1: q;

นำค่า k ไปใช้ในสมการที่ (2.44) โดยให้ k มีค่าเท่ากับ 0 และเพิ่มขึ้นทีละ 1 จนถึง q

$$b_k^{\circ} = \frac{1}{\sum_{k=0}^{q} (a_1)^{-2k}} (a_1)^{-k}$$

Nx = length(x);

นำขนาดข้อมูลของเวกเตอร์ x มาเก็บไว้ในตัวแปร Nx

hn_et = (a1.^(-k))/sum(a1.^(-2*k));

คือการนำสมการของ b_k° มาไว้ในตัวแปร hn_et ซึ่งทำหน้าที่เป็นตัวกรองดังสมการที่ (2.44)

$$b_k^{\circ} = \frac{1}{\sum_{k=0}^{q}(a_1)^{-2k}}(a_1)^{-k}$$

tp = conv(x, hn_ma);

คือการนำเวกเตอร์ x มาคอลโวลูชั่นกับ hn_ma ที่มีหน้าที่เป็นตัวกรองแล้วเก็บค่าที่ได้ไว้ในตัวแปร tp โดยคุณสมบัติของการทำคอลโวลูชั่นคือ

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

ทำหน้าที่เดียวกับ

$$Y[z] = X[z]H[z]$$

และ

$$y[n] = Z^{-1}\{Y(z)\}$$

นั่นคือ สมบัติของการทำคอลโวลูชั่นจะเทียบได้กับการคูณของซีทรานฟอร์มนั่นเอง

yn_ma = tp(1: Nx);

คือการปรับขนาดของ tp ให้มีขนาดเท่ากับข้อมูลเริ่มต้น ที่เก็บไว้ใน Nx จากนั้นนำค่าที่ได้มาเก็บไว้ในตัวแปร yn_ma

yn_ma = sqrt(EnergyX)*yn_ma/norm(yn_ma,2);

ปรับขนาดเสียงของ yn_ma ให้มีขนาดเสียงเท่ากับขนาดเสียงก่อนที่จะนำไปเข้าตัวกรองโดยนำเวกเตอร์ที่จะปรับขนาดเสียงหารด้วยขนาดของเวกเตอร์นั้นซึ่งจะได้เวกเตอร์หนึ่งหน่วยมาคูณกับขนาดของเวกเตอร์ต้นแบบ (ในที่นี้คือ EnergyX) ทำให้ขนาดเสียงดังเท่ากับเสียงต้นแบบก่อนผ่านตัวกรองดังที่กล่าวไว้ในสมการที่ (3.5) ดังนี้

$$yma[n] = \frac{y[n]}{\|\underline{y}\|_2} \times \|\underline{x}\|_2$$

MSE=sum((yn_ma-s).^2)/Nx

คือการหาค่าเฉลี่ยความคลาดเคลื่อนยกกำลังสอง (Mean Square Error, MSE) เพื่อวัดประสิทธิภาพความคลาดเคลื่อน โดยกล่าวไว้ในสมการ (2.48) คือ

$$MSE = \frac{1}{N}\sum_{n=0}^{N-1}(y[n]-s[n])^2$$

3. กรณีที่แอมปลิจูดของสัญญาณรบกวนเป็น 0.05

3.1 Moving Average Filters กรณีที่ q=80

```
clear all
clc
% Data sequence x[n] = s[n] + w[n]
[s,fs,NBITS]=wavread('data.wav');
x = s + 0.05*randn(size(s));
EnergyX = norm(x,2)^2;
%------------1--------------------
sound(x, fs);
disp('OK Enter to Continue');
pause;
% Moving Average Model
q = 80;     %length of Filter
% Filter selection
Nx = length(x);
hn_ma = ones(1,q)/q;
for kk=1: 1: 1;
   % response of the filter
   tp = conv(x, hn_ma);
   yn_ma = tp(1: Nx);
   yn_ma = sqrt(EnergyX)*yn_ma/norm(yn_ma,2);
   sound(yn_ma, fs);
   disp(['OK Enter to Round : ' num2str(kk)]);
   pause;
end
MSE=sum((yn_ma-s).^2)/Nx
```

อธิบายการทำงานของโปรแกรม Moving Average Filters กรณีที่ q=80

[s,fs,NBITS]=wavread('data.wav');

คือคำสั่งในการจำลองค่าเวกเตอร์ของสัญญาณเสียงที่ชื่อ data.wav มาไว้ในตัวแปร s

x = s + 0.05*randn(size(s));

ให้ตัวแปร x มีค่าเท่ากับ เวกเตอร์ของสัญญาณเสียงเริ่มต้นรวมกับ Gaussian White Noise ดังที่กล่าวไว้ในบทที่ 2 สมการที่ (2.3) ดังนั้น x คือสัญญาณเสียงที่รวมกับสัญญาณรบกวน โดยที่ขนาดของแอมปลิจูดของสัญญาณรบกวนนั้นขึ้นอยู่กับค่าที่คูณเข้าไปใน randn (ในที่นี้ใช้ 0.05)

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left[\frac{-(x-\mu)^2}{\sigma^2}\right]$$

EnergyX = norm(x,2)^2;

คือการหาค่าขนาดของเวกเตอร์ x โดยใช้คำสั่ง norm มาเก็บค่าไว้ในตัวแปรชื่อ EnergyX เพื่อนำไปใช้ในการรักษาขนาดของเสียง โดยกล่าวไว้ในสมการที่ (3.3) ดังนี้

$$\|\underline{x}\|_2 = ((x[0])^2 + (x[1])^2 + \ldots + (x[q])^2)^{\frac{1}{2}}$$
$$\|\underline{x}\|_2 = (\sum_{i=0}^{q}|x[i]|^2)^{\frac{1}{2}}$$

sound(x, fs);

คำสั่งในการนำเวกเตอร์ x มาแสดงในรูปแบบของเสียง

Nx = length(x);

นำขนาดข้อมูลของเวกเตอร์ x มาเก็บไว้ในตัวแปร Nx

hn_ma = ones(1,q)/q;

คือการนำสมการของ b_k° มาไว้ในตัวแปร hn_ma ซึ่งทำหน้าที่เป็นตัวกรองดังสมการที่ (2.26)

$$b_k^\circ = \frac{1}{q+1}$$

tp = conv(x, hn_ma);

คือการนำเวกเตอร์ x มาคอลโวลูชั่นกับ hn_ma ที่มีหน้าที่เป็นตัวกรองแล้วเก็บค่าที่ได้ไว้ในตัวแปร tp โดยคุณสมบัติของการทำคอลโวลูชั่นคือ

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

ทำหน้าที่เดียวกับ

$$Y[z] = X[z]H[z]$$

และ

$$y[n] = Z^{-1}\{Y(z)\}$$

นั่นคือ สมบัติของการทำคอลโวลูชั่นจะเทียบได้กับการคูณของซีทรานฟอร์มนั่นเอง

yn_ma = tp(1: Nx);

คือการปรับขนาดของ tp ให้มีขนาดเท่ากับข้อมูลเริ่มต้น ที่เก็บไว้ใน Nx จากนั้นนำค่าที่ได้มาเก็บไว้ในตัวแปร yn_ma

yn_ma = sqrt(EnergyX)*yn_ma/norm(yn_ma,2);

ปรับขนาดเสียงของ yn_ma ให้มีขนาดเสียงเท่ากับขนาดเสียงก่อนที่จะนำไปเข้าตัวกรองโดยนำเวกเตอร์ที่จะปรับขนาดเสียงหารด้วยขนาดของเวกเตอร์นั้นซึ่งจะได้เวกเตอร์หนึ่งหน่วยมาคูณกับขนาดของเวกเตอร์ต้นแบบ (ในที่นี้คือ EnergyX) ทำให้ขนาดเสียงดังเท่ากับเสียงต้นแบบก่อนผ่านตัวกรอง ดังที่กล่าวไว้ในสมการที่ (3.5) ดังนี้

$$yma[n] = \frac{y[n]}{\|\underline{y}\|_2} \times \|\underline{x}\|_2$$

MSE=sum((yn_ma-s).^2)/Nx

คือการหาค่าเฉลี่ยความคลาดเคลื่อนยกกำลังสอง (Mean Square Error, MSE) เพื่อวัดประสิทธิภาพความคลาดเคลื่อน โดยกล่าวไว้ในสมการที่ (2.48) คือ

$$MSE = \frac{1}{N}\sum_{n=0}^{N-1}(y[n] - s[n])^2$$

3.2 Exponential Filters กรณีที่ $a1 = 1$ และ q=80

```
clear all
clc
% Data sequence x[n] = s[n] + w[n]
[s,fs,NBITS]=wavread('data.wav');
x = s + 0.05*randn(size(s));
EnergyX = norm(x,2)^2;
%------------1--------------------
sound(x, fs);
disp('OK Enter to Continue');
pause;
% Exponentail Model
q = 80;        %length of Filter
a1 =1  ;    % Exponential constant
% Filter selection
k = 0: 1: q;
Nx = length(x);
hn_et = (a1.^(-k))/sum(a1.^(-2*k));
for kk=1: 1: 1;
   % response of the filter
   tp = conv(x, hn_et);
   yn_et = tp(1: Nx);
   yn_et = sqrt(EnergyX)*yn_et/norm(yn_et,2);
   sound(yn_et, fs);
   disp(['OK Enter to Round : ' num2str(kk)]);
   pause;
end
MSE1=sum((yn_et-s).^2)/Nx
```

อธิบายการทำงานของโปรแกรม Exponential Filters กรณีที่ $a1 = 1$ และ q=80

[s,fs,NBITS]=wavread('data.wav');

คือคำสั่งในการจำลองค่าเวกเตอร์ของสัญญาณเสียงที่ชื่อ data.wav มาไว้ในตัวแปร s

x = s + 0.05*randn(size(s));

ให้ตัวแปร x มีค่าเท่ากับ เวกเตอร์ของสัญญาณเสียงเริ่มต้นรวมกับ Gaussian White Noise ดังที่กล่าวไว้ในบทที่ 2 สมการที่ (2.3) ดังนั้น x คือสัญญาณเสียงที่รวมกับสัญญาณรบกวน โดยที่ขนาดของแอมปลิจูดของสัญญาณรบกวนนั้นขึ้นอยู่กับค่าที่คูณเข้าไปใน randn (ในที่นี้ใช้ 0.05)

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[\frac{-(x-\mu)^2}{\sigma^2}\right]$$

EnergyX = norm(x,2)^2;

คือการหาค่าขนาดของเวกเตอร์ x โดยใช้คำสั่ง norm มาเก็บค่าไว้ในตัวแปรชื่อ EnergyX เพื่อนำไปใช้ในการรักษาขนาดของเสียง โดยกล่าวไว้ในสมการที่ (3.3) ดังนี้

$$\|\underline{x}\|_2 = ((x[0])^2 + (x[1])^2 + \ldots + (x[q])^2)^{\frac{1}{2}}$$
$$\|\underline{x}\|_2 = (\sum_{i=0}^{q} |x[i]|^2)^{\frac{1}{2}}$$

sound(x, fs);

คำสั่งในการนำเวกเตอร์ x มาแสดงในรูปแบบของเสียง

k = 0: 1: q;

นำค่า k ไปใช้ในสมการที่ (2.44) โดยให้ k มีค่าเท่ากับ 0 และเพิ่มขึ้นทีละ 1 จนถึง q

$$b_k^{\circ} = \frac{1}{\sum_{k=0}^{q} (a_1)^{-2k}} (a_1)^{-k}$$

Nx = length(x);

นำขนาดข้อมูลของเวกเตอร์ x มาเก็บไว้ในตัวแปร Nx

hn_et = (a1.^(-k))/sum(a1.^(-2*k));

คือการนำสมการของ b_k° มาไว้ในตัวแปร hn_et ซึ่งทำหน้าที่เป็นตัวกรองดังสมการที่ (2.44)

$$b_k^{\circ} = \frac{1}{\sum_{k=0}^{q}(a_1)^{-2k}}(a_1)^{-k}$$

tp = conv(x, hn_ma);

คือการนำเวกเตอร์ x มาคอลโวลูชั่นกับ hn_ma ที่มีหน้าที่เป็นตัวกรองแล้วเก็บค่าที่ได้ไว้ในตัวแปร tp โดยคุณสมบัติของการทำคอลโวลูชั่นคือ

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

ทำหน้าที่เดียวกับ

$$Y[z] = X[z]H[z]$$

และ

$$y[n] = Z^{-1}\{Y(z)\}$$

นั่นคือ สมบัติของการทำคอลโวลูชั่นจะเทียบได้กับการคูณของซีทรานฟอร์มนั่นเอง

yn_ma = tp(1: Nx);

คือการปรับขนาดของ tp ให้มีขนาดเท่ากับข้อมูลเริ่มต้น ที่เก็บไว้ใน Nx จากนั้นนำค่าที่ได้มาเก็บไว้ในตัวแปร yn_ma

yn_ma = sqrt(EnergyX)*yn_ma/norm(yn_ma,2);

ปรับขนาดเสียงของ yn_ma ให้มีขนาดเสียงเท่ากับขนาดเสียงก่อนที่จะนำไปเข้าตัวกรองโดยนำเวกเตอร์ที่จะปรับขนาดเสียงหารด้วยขนาดของเวกเตอร์นั้นซึ่งจะได้เวกเตอร์หนึ่งหน่วยมาคูณกับขนาดของเวกเตอร์ต้นแบบ (ในที่นี้คือ EnergyX) ทำให้ขนาดเสียงดังเท่ากับเสียงต้นแบบก่อนผ่านตัวกรองดังที่กล่าวไว้ในสมการที่ (3.5) ดังนี้

$$yma[n] = \frac{y[n]}{\|\underline{y}\|_2} \times \|\underline{x}\|_2$$

MSE=sum((yn_ma-s).^2)/Nx

คือการหาค่าเฉลี่ยความคลาดเคลื่อนยกกำลังสอง (Mean Square Error, MSE) เพื่อวัดประสิทธิภาพความคลาดเคลื่อน โดยกล่าวไว้ในสมการที่ (2.48) คือ

$$MSE = \frac{1}{N}\sum_{n=0}^{N-1}(y[n]-s[n])^2$$

3.3 Exponential Filters กรณีที่ $a1 = 40$ และ q=80

```
clear all
clc
% Data sequence x[n] = s[n] + w[n]
[s,fs,NBITS]=wavread('data.wav');
x = s + 0.05*randn(size(s));
EnergyX = norm(x,2)^2;
%-----------1-------------------
sound(x, fs);
disp('OK Enter to Continue');
pause;
% Exponentail Model
q = 80;        %length of Filter
a1 =40  ;    % Exponential constant
% Filter selection
k = 0: 1: q;
Nx = length(x);
hn_et = (a1.^(-k))/sum(a1.^(-2*k));
for kk=1: 1: 1;
   % response of the filter
   tp = conv(x, hn_et);
   yn_et = tp(1: Nx);
   yn_et = sqrt(EnergyX)*yn_et/norm(yn_et,2);
   sound(yn_et, fs);
   disp(['OK Enter to Round : ' num2str(kk)]);
   pause;
end
MSE1=sum((yn_et-s).^2)/Nx
```

อธิบายการทำงานของโปรแกรม Exponential Filters กรณีที่ $a1 = 40$ และ q=80

[s,fs,NBITS]=wavread('data.wav');

คือคำสั่งในการจำลองค่าเวกเตอร์ของสัญญาณเสียงที่ชื่อ data.wav มาไว้ในตัวแปร s

x = s + 0.05*randn(size(s));

ให้ตัวแปร x มีค่าเท่ากับ เวกเตอร์ของสัญญาณเสียงเริ่มต้นรวมกับ Gaussian White Noise ดังที่กล่าวไว้ในบทที่ 2 สมการที่ (2.3) ดังนั้น x คือสัญญาณเสียงที่รวมกับสัญญาณรบกวน โดยที่ขนาดของแอมปลิจูดของสัญญาณรบกวนนั้นขึ้นอยู่กับค่าที่คูณเข้าไปใน randn (ในที่นี้ใช้ 0.05)

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[\frac{-(x-\mu)^2}{\sigma^2}\right]$$

EnergyX = norm(x,2)^2;

คือการหาค่าขนาดของเวกเตอร์ x โดยใช้คำสั่ง norm มาเก็บค่าไว้ในตัวแปรชื่อ EnergyX เพื่อนำไปใช้ในการรักษาขนาดของเสียง โดยกล่าวไว้ในสมการที่ (3.3) ดังนี้

$$\|\underline{x}\|_2 = ((x[0])^2 + (x[1])^2 + \ldots + (x[q])^2)^{\frac{1}{2}}$$
$$\|\underline{x}\|_2 = (\sum_{i=0}^{q} |x[i]|^2)^{\frac{1}{2}}$$

sound(x, fs);

คำสั่งในการนำเวกเตอร์ x มาแสดงในรูปแบบของเสียง

k = 0: 1: q;

นำค่า k ไปใช้ในสมการที่ (2.44) โดยให้ k มีค่าเท่ากับ 0 และเพิ่มขึ้นทีละ 1 จนถึง q

$$b_k^{\circ} = \frac{1}{\sum_{k=0}^{q} (a_1)^{-2k}} (a_1)^{-k}$$

Nx = length(x);

นำขนาดข้อมูลของเวกเตอร์ x มาเก็บไว้ในตัวแปร Nx

hn_et = (a1.^(-k))/sum(a1.^(-2*k));

คือการนำสมการของ b_k° มาไว้ในตัวแปร hn_et ซึ่งทำหน้าที่เป็นตัวกรองดังสมการที่ (2.44)

$$b_k^\circ = \frac{1}{\sum_{k=0}^{q}(a_1)^{-2k}}(a_1)^{-k}$$

tp = conv(x, hn_ma);

คือการนำเวกเตอร์ x มาคอลโวลูชั่นกับ hn_ma ที่มีหน้าที่เป็นตัวกรองแล้วเก็บค่าที่ได้ไว้ในตัวแปร tp โดยคุณสมบัติของการทำคอลโวลูชั่นคือ

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

ทำหน้าที่เดียวกับ

$$Y[z] = X[z]H[z]$$

และ

$$y[n] = Z^{-1}\{Y(z)\}$$

นั่นคือ สมบัติของการทำคอลโวลูชั่นจะเทียบได้กับการคูณของซีทรานฟอร์มนั่นเอง

yn_ma = tp(1: Nx);

คือการปรับขนาดของ tp ให้มีขนาดเท่ากับข้อมูลเริ่มต้น ที่เก็บไว้ใน Nx จากนั้นนำค่าที่ได้มาเก็บไว้ในตัวแปร yn_ma

yn_ma = sqrt(EnergyX)*yn_ma/norm(yn_ma,2);

ปรับขนาดเสียงของ yn_ma ให้มีขนาดเสียงเท่ากับขนาดเสียงก่อนที่จะนำไปเข้าตัวกรองโดยนำเวกเตอร์ที่จะปรับขนาดเสียงหารด้วยขนาดของเวกเตอร์นั้นซึ่งจะได้เวกเตอร์หนึ่งหน่วยมาคูณกับขนาดของเวกเตอร์ต้นแบบ (ในที่นี้คือ EnergyX) ทำให้ขนาดเสียงดังเท่ากับเสียงต้นแบบก่อนผ่านตัวกรองดังที่กล่าวไว้ในมีสมการที่ (3.5) ดังนี้

$$yma[n] = \frac{y[n]}{\|\underline{y}\|_2} \times \|\underline{x}\|_2$$

MSE=sum((yn_ma-s).^2)/Nx

คือการหาค่าเฉลี่ยความคลาดเคลื่อนยกกำลังสอง (Mean Square Error, MSE) เพื่อวัดประสิทธิภาพความคลาดเคลื่อน โดยกล่าวไว้ในสมการที่ (2.48) คือ

$$MSE = \frac{1}{N}\sum_{n=0}^{N-1}(y[n]-s[n])^2$$

ประวัติผู้เขียนโครงงาน

ชื่อ นายกันอัณณพ ชัยเร็ว

ภูมิลำเนา 51 ซ.2 ถ.พระเจ้าทันใจ ต.เวียงเหนือ อ.เมือง จ.ลำปาง 52000

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนบุญวาทย์วิทยาลัย
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail: o_penicillin_o@hotmail.com

ชื่อ นายวัฒนา มัธยมนันทน์

ภูมิลำเนา 190/1-3 ถ.บุญวาทย์ ต.สวนดอก อ.เมือง จ.ลำปาง 52000

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนบุญวาทย์วิทยาลัย
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail: mumrah13@hotmail.com

ชื่อ นายบุรินทร์ ฉัตรแก้ว

ภูมิลำเนา 256 ม.5 ต.เวียงตาล อ.ห้างฉัตร จ.ลำปาง 52190

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนลำปางกัลยาณี
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail: Shadow_bcgroup@hotmail.com