

การประดิษฐ์โปรแกรมกำจัดเสียงสะท้อนในระบบโทรศัพท์



โดย

อาจารย์ ดร.วาทิต เบญจพลกุล
วุฒิ D. ENG. (ELECTRONICS)

โครงการวิจัยเลขที่ 122-MRD-2534
ทุนส่งเสริมการวิจัยคณะวิศวกรรมศาสตร์

สถาบันวิจัยและพัฒนาคณะวิศวกรรมศาสตร์
คณะวิศวกรรมศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย
กรุงเทพฯ

สิงหาคม 2538



บทคัดย่อ

โครงการนี้ศึกษาวิธีการกำจัดเสียงสะท้อนในระบบโทรศัพท์ ซึ่งมีอยู่ด้วยกันหลายวิธี สำหรับโครงการนี้ใช้วิธีการสังเคราะห์สัญญาณเสียงเลียนแบบสัญญาณเสียงสะท้อนโดยใช้ adaptive filter หลังจากนั้นก็นำสัญญาณเสียงเลียนแบบนั้นไปหักล้างออกจากสัญญาณรวมระหว่างสัญญาณเสียงของผู้พูดด้านใกล้กับสัญญาณเสียงสะท้อน ผลที่ได้ก็คือสัญญาณเสียงจากผู้พูดด้านใกล้เพียงอย่างเดียว ซึ่งวิธีนี้มีข้อดีคือต้นทุนสามารถพูดสวนกันได้ adaptive filter ที่ใช้ในการเลียนแบบสัญญาณเสียงสะท้อนใช้หลักการของ LMS หรือ Least Mean Square ทุกส่วนของตัวกำจัดเสียงสะท้อนรวมทั้งสัญญาณเสียงต่างๆในโครงการนี้จำลองขึ้นบนเครื่องคอมพิวเตอร์ซึ่งโปรแกรมทั้งหมดเขียนด้วยภาษา C งานหลักของโครงการคือการเขียนโปรแกรมและการหาค่าพารามิเตอร์ของตัวกำจัดเสียงสะท้อนที่ทำให้สัญญาณเสียงที่ออกจากตัวกำจัดเสียงสะท้อนมีลักษณะใกล้เคียงกับสัญญาณเสียงพูดจากด้านใกล้มากที่สุด ค่าพารามิเตอร์เหล่านี้สามารถหาได้โดยการทดลองเปลี่ยนค่าไปเรื่อยๆแล้วเปรียบเทียบสัญญาณที่ออกจากตัวกำจัดเสียงสะท้อนกับสัญญาณเสียงจากด้านใกล้ทั้งในอาณาจักรเวลา (time domain) และ อาณาจักรความถี่ (frequency domain) ทำเช่นนี้ไปจนกว่าสัญญาณเสียงที่ออกจากตัวกำจัดเสียงสะท้อนมีลักษณะใกล้เคียงกับสัญญาณเสียงพูดจากด้านใกล้มากที่สุด ซึ่งจากผลการทดลองเปลี่ยนแปลงลักษณะของสัญญาณเสียงสะท้อนในรูปแบบต่างๆ ทำให้หาค่าพารามิเตอร์ที่เหมาะสมกับสัญญาณเสียงต่างๆไปได้คือ ความยาวของเวกเตอร์นำหนักถ่วง = 10 , ค่าเริ่มต้นของเวกเตอร์นำหนักถ่วง = 0 , จำนวนรอบในการทำซ้ำ = 100 , ค่า μ หรือ loop gain = 4×10^{-7} ซึ่งค่าเหล่านี้อาจไม่เหมาะสมกับสัญญาณเสียงสะท้อนบางกรณีแต่ว่าโดยรวมแล้วให้ผลที่ดี

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



ABSTRACT

This research studies a method of cancelling echos in the telephone system. There are several methods to cancel the echo . This research uses the approach which synthesize a replica of the echo by using adaptive filter and then subtract that replica from the combined echo and the near-end speech signal to obtain only the near-end speech signal . The advantage of this method is double-talking . All parts of the echo canceler and the signals are simulated on a microcomputer using the C language. The major work of this research are programming , and searching for the optimum set of cho canceler parameters which will yield the output most similar to the near-end speech signal. These parameters can be found by varying parameters of the echo canceler and then comparing the result from the echo canceler with the near-end speech signal in both the time domain and the frequency domain and continuing this procedure until the output from the echo canceler most similar to the near-end speech signal. The optimum parameters from the experimental results are as follow : Length of weight vector = 10 . bias weight = 0, number of iterations = 100, and μ (loop gain) = 4×10^{-5} . These parameters may not be appropriate for other forms of echo but in a majority of cases they give a satisfactory result.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



กิตติกรรมประกาศ

ผู้วิจัยขอขอบคุณฝ่ายวิจัย คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัยที่ได้กรุณาสนับสนุนเงินทุนวิจัยโครงการนี้จนเสร็จสิ้นสมบูรณ์

ขอขอบคุณห้องปฏิบัติการ Digital Signal Processing แห่งภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ที่ได้ช่วยเหลือเพื่อซอฟต์แวร์ช่วยออกแบบ Digital Filter ให้กับโครงการนี้

ขอขอบคุณนายธีรเทพ สอสกุล ที่ได้มีส่วนช่วยให้โครงการนี้สำเร็จโดยเฉพาะอย่างยิ่งการทดสอบโปรแกรมและการจัดทำรายงานฉบับสมบูรณ์



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ



	หน้า
บทคัดย่อ	ก
ABSTRACT	ข
บทที่ 1 บทนำ	
1.1 วัตถุประสงค์	2
1.2 ขอบข่ายของโครงการ	2
1.3 ขั้นตอนการดำเนินงาน	2
บทที่ 2 การกำจัดเสียงสะท้อนในระบบโทรศัพท์	
2.1 การเกิดเสียงสะท้อน	3
2.2 วิธีการกำจัดเสียงสะท้อนในระบบโทรศัพท์	4
2.3 การทำงานของ LMS adaptive filter	7
บทที่ 3 แบบจำลองของเสียงพูดและเสียงสะท้อน :	
3.1 การกำเนิด Stochastic variables สำหรับการ Simulation	11
3.2 การแจกแจงความน่าจะเป็นแบบ Uniform	13
3.3 การแจกแจงความน่าจะเป็นแบบ Exponential	15
3.4 การแจกแจงความน่าจะเป็นแบบ Normal	16
3.5 การแจกแจงความน่าจะเป็นแบบ Multivariate Normal	20
3.6 การจำลองสัญญาณเสียงพูดและเสียงสะท้อน	24
บทที่ 4 การจำลอง Echo canceler บนคอมพิวเตอร์ :	
4.1 การจำลอง Adaptive filter	27
4.2 การจำลอง fixed parameters IIR filter	31
บทที่ 5 โปรแกรมอ่าน-เขียนข้อมูล และติดต่อกับผู้ใช้	
5.1 การเก็บข้อมูลลงดิสก์	34
5.2 การเปิดเพิ่มข้อมูล	35
5.3 การเข้าถึงข้อมูลในแฟ้มข้อมูล	37

5.4 การติดต่อกับผู้ใช้โปรแกรม	39
5.5 วิธีการใช้งาน	40
บทที่ 6 การทดลองและสรุปผล	
6.1 วิธีการทดลอง	44
6.2 ผลการทดลอง	47
6.3 วิเคราะห์ผลการทดลอง	80
6.4 สรุปผล	81
6.5 ประโยชน์ที่ได้รับ	82
เอกสารอ้างอิง	83



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

เลขหมู่ ๖๖๕
๖๓ ๑๕
เลขทะเบียน ๐๐๘ ๕๕๘
วัน,เดือน,ปี ๑๔ ๕.๓.๓๘



บทที่ 1

บทนำ

เนื่องจากในปัจจุบันนี้ระบบการสื่อสารข้อมูลชนิดต่างๆในประเทศไทยกำลังเจริญก้าวหน้าเป็นอย่างมาก การโทรศัพท์ทางไกลข้ามประเทศจึงใช้ดาวเทียมในการติดต่อ แต่ว่าในการโทรศัพท์ทางไกลผ่านดาวเทียมนั้นจะก่อปัญหาแก่ผู้ใช้บริการอยู่สองประการด้วยกัน ประการแรกคือ ขณะสนทนาจะรู้สึกไม่ต่อเนื่องเพราะเกิดจากสัญญาณต้องเดินทางจากสถานีภาคพื้นดินขึ้นสู่อวกาศซึ่งอยู่ห่างจากโลกเป็นระยะทางหลายหมื่นกิโลเมตรซึ่งในระยะทางขนาดนี้สัญญาณต้องใช้เวลาในการเดินทางนานพอที่จะรู้สึกว่าการสนทนาขาดตอน ประการที่สองคือ ขณะที่สนทนาจะเกิดเสียงสะท้อนของผู้พูดเองซึ่งมีสาเหตุมาจากพลังงานไม่สามารถส่งผ่านไปยังวงจรของผู้รับได้ทั้งหมดเนื่องจากเกิด impedance mismatch พลังงานส่วนที่เหลือจึงสะท้อนกลับมายังผู้พูดอีก ทำให้ผู้พูดรู้สึกรำคาญและอาจจะสนทนากันไม่รู้เรื่อง ในการที่จะแก้ปัญหาแรกนั้นก็อาจทำโดยใช้เส้นใยนำแสงแทนเพราะวางในแนวราบสัญญาณจึงใช้เวลาในการเดินทางน้อยกว่า ส่วนปัญหาที่สองนั้นได้มีผู้ทำการวิจัยการกำจัดเสียงสะท้อนของระบบโทรศัพท์ในต่างประเทศและสร้างเครื่องกำจัดเสียงสะท้อนออกมาจำหน่ายโดยใช้ชิพ VLSI ซึ่งมีราคาแพง ผู้จัดทำโครงการนี้จึงเห็นว่าถ้าเราสามารถพัฒนาเครื่องกำจัดเสียงสะท้อนนี้ได้เองในประเทศจะเป็นการลดการนำเข้าสินค้าราคาแพงจากต่างประเทศ โครงการนี้เป็นเพียงจุดเริ่มต้นของการพัฒนาเครื่องกำจัดเสียงสะท้อนในระบบโทรศัพท์ เพราะว่าโดยตัวของโครงการเองนั้นไม่ได้มีความสามารถในการกำจัดเสียงสะท้อนออกจากระบบโทรศัพท์ แต่โครงการนี้มีส่วนในการศึกษาหาค่าของพารามิเตอร์ของเครื่องกำจัดเสียงสะท้อนในระบบโทรศัพท์เพื่อใช้ในการสร้างออกมาเป็น hardware ต่อไปในอนาคต

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

1.1 วัตถุประสงค์

1. เพื่อศึกษาวิธีการกำจัดเสียงสะท้อนในระบบโทรศัพท์
2. เพื่อศึกษาวิธีการกำเนิดตัวแปรเชิงสุ่ม
3. เพื่อศึกษาวิธีการจำลองสัญญาณเสียงพูดด้วยคอมพิวเตอร์
4. เพื่อศึกษาวิธีการวิเคราะห์สัญญาณทั้งในอาณาจักรเวลาและอาณาจักรความถี่
5. เพื่อจำลองการกำจัดเสียงสะท้อนในระบบโทรศัพท์ด้วยคอมพิวเตอร์
6. หาพารามิเตอร์ที่เหมาะสมสำหรับเครื่องกำจัดเสียงสะท้อนในระบบโทรศัพท์ เพื่อเป็นประโยชน์ในการสร้าง hardware ต่อไปในอนาคต

1.2 ขอบข่ายของโครงการ

โครงการนี้เป็น software ซึ่งรวมเอาเครื่องมือที่จำเป็นต่อการจำลองการกำจัดเสียงสะท้อนในระบบโทรศัพท์อันได้แก่ โปรแกรมจำลองเสียงพูดและเสียงสะท้อน , โปรแกรมกำจัดเสียงสะท้อน , โปรแกรมวิเคราะห์สัญญาณ และ โปรแกรม filter เพื่อจำลองผลของสายโทรศัพท์ ขอบข่ายของโครงการนี้คือการหาพารามิเตอร์ของเครื่องกำจัดเสียงสะท้อนในระบบโทรศัพท์เพื่อให้สัญญาณที่ออกจากเครื่องกำจัดเสียงสะท้อนมีลักษณะเหมือนกับสัญญาณเสียงที่พูดมาจากด้านไกลี่มากที่สุด

1.3 ขั้นตอนการดำเนินงาน

1. ศึกษาหลักการของ Adaptive filter ชนิดต่างๆ เพื่อใช้ในการกำจัดเสียงสะท้อนโดยจะเน้น LMS algorithm
2. ศึกษาวิธีการกำเนิดตัวแปรสุ่มแบบต่างๆเพื่อใช้ในการกำเนิดสัญญาณเสียงพูดและเสียงสะท้อน
3. เขียนโปรแกรมเพื่อจำลองการทำงานของระบบทั้งหมด
4. ทำการทดลองกำจัดเสียงสะท้อนโดยเปลี่ยนพารามิเตอร์ของตัวกำจัดเสียงสะท้อนเพื่อให้ได้ผลดีที่สุด , ทำการตรวจสอบ แก้ไขข้อบกพร่องต่างๆ , สรุปผลและเขียนรายงาน

บทที่ 2

การกำจัดเสียงสะท้อนในระบบโทรศัพท์

2.1 การเกิดเสียงสะท้อน

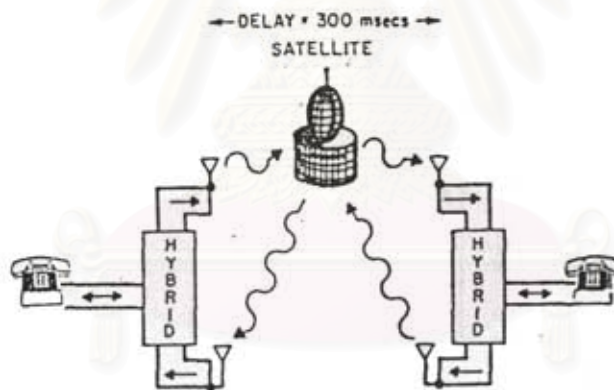
เมื่อคลื่นเสียงเดินทางไปกระทบสิ่งกีดขวาง ส่วนหนึ่งจะถูกดูดกลืนไปเป็นพลังงานความร้อนพลังงานอีกส่วนหนึ่งจะสามารถผ่านวัตถุนั้นได้ และพลังงานส่วนสุดท้ายจะสะท้อนกลับ เช่น ในห้องที่มีฝาผนังแข็งจะมีการดูดกลืนและการส่งผ่านทะลุไปน้อย แต่จะมีพลังงานที่สะท้อนมาก เราจึงสามารถเห็นผลของการสะท้อนมาก เราจึงสามารถเห็นผลของการสะท้อนชัดเจน และอาจสะท้อนไปมาหลายครั้ง ซึ่งบางครั้งอาจมีจำนวนถึง 200-300 ครั้ง ก่อนที่กำลังสะท้อนจะอ่อนกำลังลงจนไม่สามารถให้เราได้ยิน นั่นก็หมายถึงเราจะได้ยินเสียงนั้นนานกว่าปกติ และจะรู้สึกว่าเสียงนั้นยืดยาวออกไป ไม่ใช่ได้ยินเป็นคำๆ 200-300 ครั้ง ทั้งนี้เพราะแต่ละเสียงมาถึงหูในเวลาห่างกันน้อยมาก และสมองจะไม่สามารถแยกออกเป็นคนละเสียงได้ การได้ยินเสียงสะท้อนนี้ให้ความรู้สึกที่ดีในการฟังดนตรีใน CONCERT HALL แต่ ในทางตรงข้ามถ้าเสียงที่ออกไปในตัวกลางที่มีการสะท้อนสู่ผู้ฟังเลย เช่น พุดในที่โล่งแจ้งหรือในห้องที่มีตัวดูดกลืนเสียง เช่น ฝ้ายที่อ่อนนุ่ม และขรุขระ จะทำให้ผู้ฟังได้ยินเสียงห้วนๆ ไม่น่าฟัง

โดยธรรมชาติแล้วหูของมนุษย์จะสามารถแยกเสียงที่เกิดขึ้นต่างเวลากัน 2 เสียง ว่าเป็นคนละเสียงกันก็ต่อเมื่อเวลาที่เสียงทั้งสองนั้นเดินทางมาถึงหูมนุษย์เป็นเวลาที่ต่างกันอย่างน้อย 0.15 วินาที ถ้าเวลาต่างกันน้อยกว่านี้จะทำให้ได้ยินเป็นเสียงเดียวผสมกันไป

หูคนปกติสามารถได้ยินเสียงที่มีความถี่ในช่วง 20- 20000 Hz โดยการตอบสนองต่อความถี่สูงจะลดลงตามวัย ความปกติเสียงพูดของผู้ชายและผู้หญิงจะอยู่ในช่วงความถี่ประมาณ 250-3500 Hz ซึ่งผู้ชายจะมีความถี่ต่ำกว่าคืออยู่ในช่วงความถี่ 250-1000 Hz ในขณะที่ผู้หญิงจะมีความถี่ในช่วง 500-1500 Hz ซึ่งจากผลการทดลองการรับฟังของหูคนพบว่าความถี่ที่หูคนตอบสนองได้ไวที่สุดคือ 3000 Hz

ในระบบโทรศัพท์ทางไกลจะไม่เกิดปัญหาเสียงสะท้อนแม้ว่าความจริงจะมีเสียง สะท้อนก็ตาม ที่เป็นเช่นนี้ก็เพราะว่าเสียงที่สะท้อนนั้นเกิดในเวลาที่แตกต่างกันน้อยกว่า 0.15 วินาที ทำให้หูมนุษย์ไม่สามารถแยกความแตกต่างของเสียงสะท้อนกับเสียงจริงได้ แต่ในระบบโทรศัพท์ทางไกลผ่านดาวเทียมปัญหาเสียงสะท้อนเป็นปัญหาสำคัญปัญหาหนึ่ง เพราะว่าการสื่อสารผ่านดาวเทียมสิ่งที่จะหลีกเลี่ยงไม่ได้ก็คือ การเกิด Time delay ซึ่งโดยทั่วไปแล้วจะมีค่าประมาณ 600 ms (ไปและกลับ) ดังนั้นจึงทำให้เสียงสะท้อนกับเสียงจริงเกิดในเวลาที่แตกต่างกันมากกว่า 0.15 วินาที หูมนุษย์จึงตีความหมายว่าเสียงจริงกับเสียงสะท้อนเป็นคนละเสียงกัน ทำให้ผู้พูดได้ยินเสียงของตัวเองที่เพิ่งพูดไปเมื่อสักครู่นี้เอง ก่อให้เกิดความรำคาญแก่ผู้ใช้และอาจจะทำให้การสื่อ

สารไม่สัมฤทธิ์ผลได้ สาเหตุที่มีการสะท้อนของเสียงในระบบโทรศัพท์นั้นแตกต่างจากการสะท้อนของเสียงจริงๆที่ได้กล่าวไปแล้วเล็กน้อย คือเสียงสะท้อนในระบบโทรศัพท์เกิดจากปัญหาในเรื่องของ Impedance mismatch [5] เพราะในการติดต่อส่วนท้องถิ่นระหว่างเครื่องโทรศัพท์กับชุมสายจะใช้สายซึ่งติดต่อกันได้ทั้ง 2 ทาง (Bi-directional) จำนวน 2 สาย ในขณะที่การเชื่อมต่อกับระบบการสื่อสารผ่านดาวเทียมนั้นใช้สายที่ติดต่อกันได้ทางเดียว (Unidirectional) จำนวน 2 ชุด รับและส่งแยกกัน รวมเป็นสาย 4 สาย ดังนั้นจึงจำเป็นต้องมีการแปลงจากวงจร 2 สาย ของโทรศัพท์ไปเป็นวงจร 4 สาย ของระบบดาวเทียม ซึ่งอุปกรณ์ที่ใช้ในการแปลงนี้มีชื่อเรียกว่า “Hybrid” ในทางอุดมคติแล้ววงจร Hybrid ต้องสามารถส่งผ่านพลังงาน(จากผู้ส่ง)ที่มาจากวงจร 4 สายไปสู่วงจร 2 สาย(ของผู้รับ)ได้ทั้งหมดโดยไม่มีพลังงานสะท้อนกลับ แต่ในความเป็นจริงแล้ววงจร Hybrid ไม่สามารถทำงานได้อย่างในทางอุดมคติเนื่องจากปัญหา Impedance mismatch ดังนั้นจะมีพลังงานบางส่วนสะท้อนกลับสู่วงจร 4 สาย ซึ่งพลังงานนี้จะถูกส่งขึ้นดาวเทียมกลับไปสู่ผู้ส่งอีกครั้งหนึ่งทำให้ผู้ส่งได้ยินเสียงของตัวเอง ซึ่งเสียงนี้ก็คือเสียงสะท้อน(Echo) นั่นเอง รูปที่ 2-1 แสดงการเดินทางของสัญญาณในระบบโทรศัพท์ผ่านดาวเทียมซึ่งจะเห็นได้ว่าเกิดการล่าช้าทางเวลาในการเดินทางไปหรือกลับ ประมาณ 300 msec

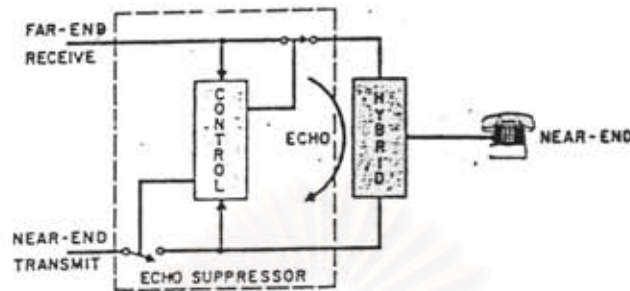


รูปที่ 2-1

2.2 วิธีการกำจัดเสียงสะท้อนในระบบโทรศัพท์

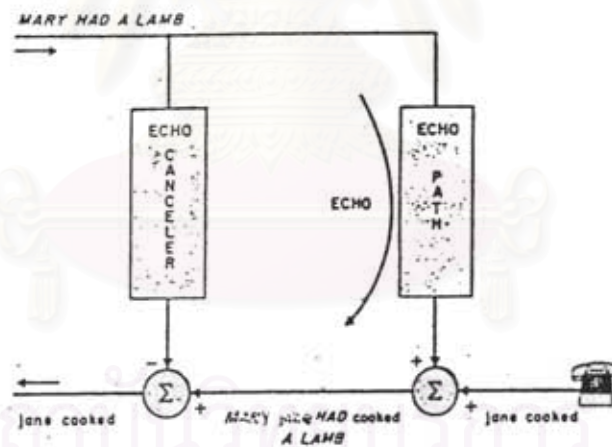
วิธีการที่จะกล่าวถึงวิธีแรกคือการใช้ Echo suppressor [5] ซึ่งอุปกรณ์นี้ใช้หลักการที่ว่าตามปกติแล้วมนุษย์มักจะไม่วางหูฟังในขณะสนทนากัน ดังนั้น Echo suppressor จะทำการต่อวงจรเฉพาะเวลาที่วางหูฟัง Echo suppressor แสดงในรูปที่ 2-2 การเปลี่ยนแปลงตำแหน่งของ switch นั้นขึ้นอยู่กับว่าขณะนั้นใครกำลังพูดอยู่ (ด้านไกล หรือ ด้านใกล้) เมื่อผู้ที่อยู่ไกลเป็นผู้พูด switch ตัวล่างจะเปิดวงจรทำให้เสียงสะท้อนไม่สามารถสะท้อนกลับไปสู่ผู้พูดที่ด้านไกลได้ ส่วน switch ตัวบนปิดวงจรเพื่อให้สัญญาณผ่านไปได้ ในทางตรงข้ามถ้าผู้ที่อยู่ด้านใกล้จะส่ง

ข้อความ switch ตัวล่างก็จะปิดวงจรของเส้นทางที่ใช้ส่งสัญญาณของด้านไกล และ switch ตัวบน จะเปิดวงจรของเส้นทางที่รับสัญญาณจากด้านไกล อย่างไรก็ตามการใช้ Echo suppressor จะทำให้ ผู้รับฟังไม่สามารถขัดจังหวะผู้พูดได้ วิธีการใหม่ซึ่งดีกว่า Echo suppressor จึงได้ถูกเสนอขึ้นมา



รูปที่ 2-2 Echo suppressor

คือใช้หลักการที่ว่าพยายามประมาณค่าเสียงสะท้อนขึ้นมาแล้วนำไปหักออกจากสัญญาณที่รวม ระหว่างสัญญาณเสียงพูดกับเสียงสะท้อน ผลที่ได้ก็คือสัญญาณเสียงพูดเพียงอย่างเดียว ทำให้ด้าน ไกลรับฟังได้ยินเฉพาะเสียงที่พูดไปจากด้านไกลจริงๆ โดยไม่มีเสียงสะท้อนผสมอยู่ หลักการนี้ เรียกว่า Echo cancellation [5] แสดงดังในรูปที่ 2-3

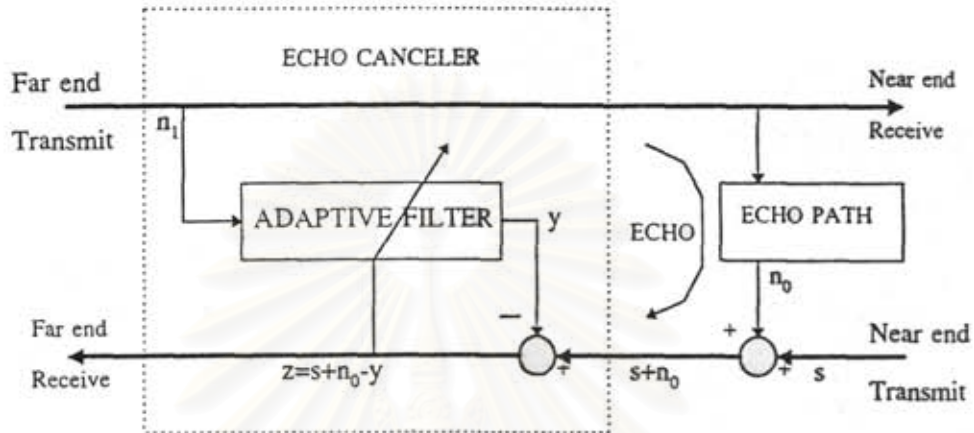


รูปที่ 2-3 หลักการ Echo cancellation

งานหลักๆที่ต้องทำเมื่อใช้วิธี Echo cancellation คือการจำลองเสียงสะท้อน แต่มีปัญหาคือว่าเราไม่รู้ function ที่แทนเส้นทางการสะท้อนของเสียง ดังนั้นเราจะต้องหา function นี้ให้ได้ วิธี ส่วนใหญ่ที่ใช้กันก็คือการใช้ Adaptive filter ในการเลียนแบบ impulse response ของเส้นทาง การสะท้อนของเสียง

Echo canceler ที่จำลองขึ้นมาในโครงงานนี้อาศัยทฤษฎีของ Adaptive noise canceler ซึ่งมีทฤษฎีโดยหลักๆดังต่อไปนี้ รูปที่2-4 แสดงบล็อกโคอะแกรมของ Echo canceler สัญญาณ

เสียงพูด s จากด้าน Near-end ถูกส่งไปในช่องสัญญาณสำหรับส่ง แล้วไปรวมกับสัญญาณเสียงสะท้อน n_0 ซึ่งไม่มีความสัมพันธ์(correlation) กับเสียงพูด s ผลรวมที่ได้คือ $s + n_0$ จะเข้าสู่ Primary input ของ Echo canceler ส่วน Reference input จะรับสัญญาณเสียงพูดจากด้าน Far-end n_1 ซึ่งมีความสัมพันธ์กับเสียงสะท้อน n_0 แต่ไม่มีความสัมพันธ์กับเสียงพูด s เสียงสะท้อน n_1 จะถูก filter เพื่อให้ได้สัญญาณ y ซึ่งจะมีลักษณะใกล้เคียงกับ n_0 มากที่สุดเท่าที่จะเป็น



รูปที่ 2-4

ไปได้ ซึ่งสัญญาณ y นี้จะไปหักล้างกับ Primary input $s + n_0$ จะได้สัญญาณออกจากระบบ $z = s + n_0 - y$

ในระบบ Echo cancelling จุดประสงค์ในทางปฏิบัติคือให้สัญญาณที่ออกจากระบบ $z = s + n_0 - y$ เป็นค่ากำลังสองที่น้อยที่สุด(least square) ของสัญญาณ s จุดประสงค์นี้สามารถบรรลุได้โดยการป้อนกลับสัญญาณออกเข้าสู่ Adaptive filter เพื่อปรับพารามิเตอร์ของตัว filter ในโครงการนี้จะใช้ LMS (Least mean squares) algorithm [4,5,8] เพื่อให้ output power มีค่าต่ำสุด

การวิเคราะห์ Echo canceler ทำได้โดย สมมติว่า s, n_0, n_1 และ y เป็น Stationary และมีค่าเฉลี่ยเป็นศูนย์ สมมติว่า s ไม่มีความสัมพันธ์กับ n_0 และ n_1 แต่ n_0 มีความสัมพันธ์กับ n_1 สัญญาณออก z คือ

$$z = s + n_0 - y \quad (2-1)$$

ยกกำลังสองสมการ (2-1) จะได้

$$z^2 = s^2 + (n_0 - y)^2 + 2s(n_0 - y) \quad (2-2)$$

หาค่าคาดหวังทั้งสองข้างของสมการ (2-2) และให้ค่านิ่งว่า s ไม่มีความสัมพันธ์กับ n_0 และ y จะได้ว่า

$$\begin{aligned} E[z^2] &= E[s^2] + E[(n_0 - y)^2] + 2E[s(n_0 - y)] \\ &= E[s^2] + E[(n_0 - y)^2] \end{aligned} \quad (2-3)$$

กำลังของสัญญาณ $E[s^2]$ จะไม่มีผลต่อการปรับตัวเองของ filter เพื่อให้ $E[z^2]$ มีค่าน้อยที่สุด นั่นคือจะได้ กำลังของสัญญาณออกต่ำสุดคือ

$$\min E[z^2] = E[s^2] + \min E[(n_0 - y)^2] \quad (2-4)$$

เมื่อ filter ปรับตัวเองจนได้ $E[z^2]$ ต่ำสุดแล้ว $E[(n_0 - y)^2]$ ก็จะมีค่าต่ำสุดด้วย สัญญาณออกจาก filter จึงเป็นการประมาณค่าของสัญญาณเสียงสะท้อน n_0 แบบกำลังสองต่ำสุด ซึ่งไปกว่านั้นเมื่อ $E[(n_0 - y)^2]$ มีค่าต่ำสุดแล้ว $E[(z - s)^2]$ ก็จะมีค่าต่ำสุดด้วยเพราะจากสมการ (2-1) จะได้

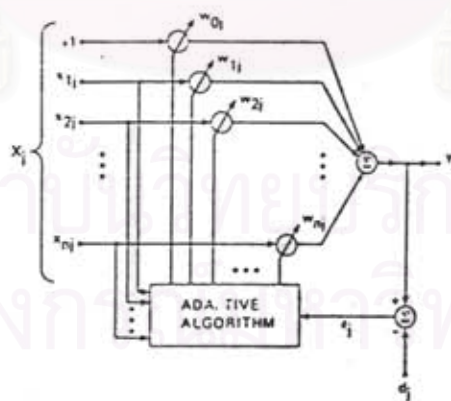
$$(z - s) = (n_0 - y) \quad (2-5)$$

ดังนั้นการปรับตัวเองของ filter เพื่อที่จะให้กำลังออกทั้งหมดมีค่าต่ำสุดก็จะเท่ากับทำให้สัญญาณออก z ประมาณค่าของสัญญาณเสียงพูด s แบบกำลังสองต่ำสุด

จากสมการที่ (2-3) จะเห็นได้ว่ากำลังออกต่ำที่สุดเท่าที่จะเป็นไปได้คือ $E[z^2] = E[s^2]$ ซึ่งจะเป็นไปได้อีกต่อเมื่อ $E[(n_0 - y)^2] = 0$ นั่นคือ $y = n_0$ และ $z = s$ ในกรณีนี้กำลังออกต่ำสุดจะทำให้สัญญาณออกปราศจากเสียงสะท้อนโดยสมบูรณ์ แต่ในทางปฏิบัติแล้วสัญญาณออกก็ยังคงมีเสียงสะท้อนอยู่บ้าง

2.3 การทำงานของ LMS Adaptive filter [4]

ส่วนประกอบหลักของระบบที่มีการปรับตัวเอง (adaptive systems) ก็คือ adaptive linear combiner ดังแสดงในรูปที่ 2-5 ตัว combiner จะทำการให้น้ำหนัก (weights) และรวมชุดของสัญญาณ input เพื่อที่จะสร้างสัญญาณ output



รูปที่ 2-5 คือ adaptive linear combiner

เวกเตอร์ของสัญญาณ input X_i มีนิยามคือ

$$X_j = \begin{bmatrix} x_{0j} \\ x_{1j} \\ \circ \\ \circ \\ \circ \\ x_{nj} \end{bmatrix} \quad (2-6)$$

ถือว่าทุกสัญญาณ input ปรากฏพร้อมกันบนสาย input ที่เวลาซึ่งไม่ต่อเนื่อง(discrete times) โดยมีดัชนีคือ j สำหรับสัมประสิทธิ์ของน้ำหนักถ่วง w_0, w_1, \dots, w_n สามารถปรับเปลี่ยนได้คงสัญลักษณ์ที่แสดงในรูป 2-5 คือเป็นวงกลมแล้วมีลูกศรผ่าน เวกเตอร์ของน้ำหนักถ่วง(weight vector) W คือ

$$W = \begin{bmatrix} w_0 \\ w_1 \\ \circ \\ \circ \\ \circ \\ w_n \end{bmatrix} \quad (2-7)$$

สัญญาณออก y_j ได้จากผลคูณระหว่าง x_j กับ W

$$y_j = X_j^T W = W^T X_j \quad (2-8)$$

สัญญาณผิดพลาด ϵ_j คือผลต่างระหว่างค่าผลตอบที่ต้องการ d_j กับผลตอบที่แท้จริง y_j

$$\epsilon_j = d_j - X_j^T W = d_j - W^T X_j \quad (2-9)$$

ซึ่ง d_j ก็คือสัญญาณที่เข้า primary input ของ Echo canceller นั้นเอง

จุดประสงค์ของ adaptive algorithm ในรูปที่ 2-5 คือปรับน้ำหนักถ่วงของ adaptive linear combiner ให้ค่าเฉลี่ยกำลังสองของสัญญาณผิดพลาดมีค่าต่ำที่สุด เราจะหารูปทั่วไปของค่าเฉลี่ยกำลังสองของสัญญาณผิดพลาดในฟังก์ชันของน้ำหนักถ่วง โดยสมมุติว่า สัญญาณ input และผลตอบที่ต้องการ เป็น statistically stationary คือมีค่าไม่ขึ้นกับเวลา และให้น้ำหนักถ่วงมีค่าค่าหนึ่ง เราจะหาฟังก์ชันได้ตามขั้นตอนต่อไปนี้

ยกกำลังสองสมการ(2-9) จะได้

$$\epsilon_j^2 = d_j^2 - 2d_j X_j^T W + W^T X_j X_j^T W \quad (2-10)$$

หาค่าคาดหวังทั้งสองข้างของสมการ(2-10)

$$E[\epsilon_j^2] = E[d_j^2] - 2E[d_j X_j^T] W + W^T E[X_j X_j^T] W \quad (2-11)$$

นิยามให้เวกเตอร์ P คือ cross correlation ระหว่างผลตอบที่ต้องการ(เป็น scalar) กับเวกเตอร์ X นั่นคือ

$$P = E[d, X_j] = E \begin{bmatrix} d_j x_{0j} \\ d_j x_{1j} \\ \circ \\ \circ \\ \circ \\ d_j x_{nj} \end{bmatrix} \quad (2-12)$$

นิยามให้เมตริกซ์ R คือ auto correlation ของสัญญาณ input นั่นคือ

$$R = E[X_j X_j^T] = E \begin{bmatrix} x_{0j} x_{0j} & x_{0j} x_{1j} & x_{0j} x_{2j} & \dots \\ x_{1j} x_{0j} & x_{1j} x_{1j} & x_{1j} x_{2j} & \dots \\ \circ & \circ & \circ & \circ \\ \circ & \circ & \circ & \circ \\ \dots & \dots & \dots & x_{nj} x_{nj} \end{bmatrix} \quad (2-13)$$

ดังนั้นค่าเฉลี่ยกำลังสองเขียนใหม่ได้ดังนี้

$$E[\varepsilon_j^2] = E[d_j^2] - 2P^T W + W^T R W \quad (2-14)$$

จะเห็นได้ว่าค่าผิดพลาดจะเป็นฟังก์ชันของน้ำหนักถ่วง ซึ่งมีรูปร่างเป็นพื้นผิวพาราโบลาหงาย (คล้ายชาม) ที่มีค่าเป็นบวกเสมอ การปรับน้ำหนักถ่วงเพื่อที่จะให้ค่าผิดพลาดต่ำที่สุดก็คือการวิ่งไปบนพื้นผิวนี้นอกจากจะถึง”ห้องชาม” วิธีการหา gradient เป็นวิธีการที่ใช้กันเพื่อให้ได้ผลนี้

เกรเดียนต์ของฟังก์ชันค่าผิดพลาดได้จากการหาอนุพันธ์ของสมการ (2-14)

$$\nabla = \begin{bmatrix} \frac{\partial E[\varepsilon_j^2]}{\partial w_0} \\ \circ \\ \circ \\ \circ \\ \frac{\partial E[\varepsilon_j^2]}{\partial w_n} \end{bmatrix} = -2P + 2RW \quad (2-15)$$

เวกเตอร์ของน้ำหนักถ่วงที่เหมาะสมคือ W^* มีชื่อเรียกว่า Wiener weight vector ซึ่งได้จากการจัดให้สมการ (2-15) เท่ากับศูนย์นั่นคือ

$$W^* = R^{-1} P \quad (2-16)$$

LMS algorithm เป็นวิธีการในทางปฏิบัติในการหาค่าประมาณใกล้เคียงของผลเฉลยในสมการ (2-16) ในเวลาจริง(real time) algorithm ไม่ต้องวัด correlation function ไม่ต้องทำ matrix inversion ตามวิธีการนี้ค่าของเวกเตอร์น้ำหนักถ่วงตัวต่อไป w_{j+1} เท่ากับเวกเตอร์น้ำหนักถ่วงตัวปัจจุบัน w_j บวกกับค่าที่เป็นสัดส่วนกับค่าลบเกรเดียนต์ นั่นคือ

$$w_{j+1} = w_j - \mu \nabla_j \quad (2-17)$$

พารามิเตอร์ μ คือ แฟคเตอร์ที่ควบคุมเสถียรภาพและอัตราการลู่เข้า ในการทำซ้ำแต่ละครั้งจะใช้เวลาคงที่ค่าหนึ่ง เกรเดียนต์ของการทำซ้ำครั้งที่ j แทนด้วย ∇_j

LMS algorithm จะประมาณค่าของเกรเดียนต์ ∇_j ด้วยการหาอนุพันธ์ของ ε_j^2 เทียบกับ \mathbf{W} ความสัมพันธ์ระหว่างค่าจริงและค่าประมาณของเกรเดียนต์เป็นไปตามนี้

$$\nabla_j = \begin{bmatrix} \frac{\partial E[\varepsilon_j^2]}{\partial w_0} \\ \vdots \\ \frac{\partial E[\varepsilon_j^2]}{\partial w_n} \end{bmatrix}_{\mathbf{w}=\mathbf{w}_j}, \quad \hat{\nabla}_j = \begin{bmatrix} \frac{\partial \varepsilon_j^2}{\partial w_0} \\ \vdots \\ \frac{\partial \varepsilon_j^2}{\partial w_n} \end{bmatrix}_{\mathbf{w}=\mathbf{w}_j} = 2\varepsilon_j \begin{bmatrix} \frac{\partial \varepsilon_j}{\partial w_0} \\ \vdots \\ \frac{\partial \varepsilon_j}{\partial w_n} \end{bmatrix}_{\mathbf{w}=\mathbf{w}_j} \quad (2-18)$$

components ของค่าประมาณเกรเดียนต์ก็คือ partial derivative ของค่าผิดพลาดในขณะนั้นเทียบกับ components ของน้ำหนักถ่วง ซึ่งหาได้จากการหาอนุพันธ์ของสมการ (2-9) เทียบกับ \mathbf{W} นั่นคือค่าประมาณเกรเดียนต์สามารถเขียนอยู่ในรูป

$$\hat{\nabla}_j = -2\varepsilon_j \mathbf{X}_j \quad (2-19)$$

ใช้ค่าประมาณเกรเดียนต์แทนค่าจริงของเกรเดียนต์ในสมการ (2-17) จะได้ซึ่ง Widrow-Hoff LMS algorithm คือ

$$\mathbf{W}_{j+1} = \mathbf{W}_j + 2\mu \varepsilon_j \mathbf{X}_j \quad (2-20)$$

algorithm นี้ง่ายและสะดวกในการติดตั้งใช้งาน สำหรับค่า μ จะขึ้นอยู่กับค่า eigenvalues ของ autocorrelation matrix ของสัญญาณที่เข้า Echo canceller ซึ่งก็คือ matrix \mathbf{R} นั้นเอง ความสัมพันธ์ระหว่างค่า eigenvalue ที่มากที่สุด (λ_{\max}) ของ \mathbf{R} กับ μ คือ

$$1/\lambda_{\max} > \mu > 0 \quad (2-21)$$

ในทางปฏิบัติแล้วเราไม่สามารถรู้ค่า λ_{\max} ได้ล่วงหน้า ดังนั้นโครงงานนี้จึงจะเป็นการทดลองหาค่า μ ที่เหมาะสมสำหรับแต่ละสัญญาณ รวมทั้งทดลองหาค่าความยาวของเวกเตอร์น้ำหนักถ่วง และจำนวนการปรับในการปรับค่าน้ำหนักถ่วงของ adaptive filter

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3

แบบจำลองของเสียงพูดและเสียงสะท้อน

3.1 การกำเนิด Stochastic variables สำหรับการ Simulation [3]

ในการพิจารณา Stochastic process ที่เกี่ยวกับตัวแปรเชิงสุ่ม (random variable) ทั้งแบบต่อเนื่องและไม่ต่อเนื่อง เราจะกำหนด ฟังก์ชัน $F(x)$ เป็น Cumulative distribution function ของตัวแปร x ซึ่งแสดงค่าความน่าจะเป็นของตัวแปรเชิงสุ่ม X จะมีค่าน้อยกว่าหรือเท่ากับ x ถ้าตัวแปร x เป็นแบบไม่ต่อเนื่อง $F(x)$ จะเป็น step function แต่ในกรณีที่ $F(x)$ ต่อเนื่องทุกค่า x เราจะสามารถหาค่า Probability density function $f(x)$ ได้จากการหาอนุพันธ์ของ $F(x)$ เทียบกับ x ซึ่งสามารถเขียนเป็นสมการคือ $f(x) = dF(x)/dx$ ส่วน $F(x)$ ก็เขียนสมการได้ดังนี้คือ

$$F(x) = P(X \leq x) = \int_{-\infty}^x f(t) dt \quad (3-1)$$

เมื่อ $F(x)$ นิยามในช่วงตั้งแต่ 0 ถึง 1 และ $f(t)$ แสดง probability density function ของตัวแปร X เมื่อ $X = t$

ในที่นี้จะศึกษาวิธีการกำเนิดค่าตัวแปรจาก probability distribution ซึ่งมีอยู่ 3 วิธีการพื้นฐาน ซึ่งสามารถใช้กับ distribution แบบต่างๆ

1. Inverse Transformation method
2. Rejection method
3. Composition method

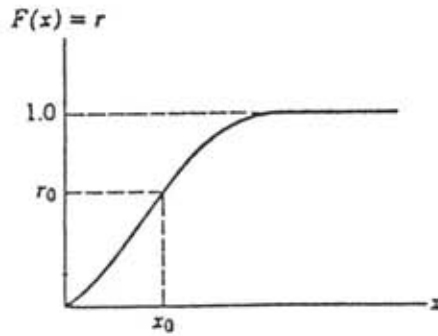
3.1.1 วิธีการ Inverse Transformation method

ถ้าต้องการกำเนิดตัวแปร x ซึ่งมี density function $f(x)$ จะต้องเริ่มที่การหา $F(x)$ เนื่องจาก $F(x)$ นิยามในช่วง 0-1 เราสามารถกำเนิดตัวเลขสุ่มที่มีการแจกแจงความน่าจะเป็นแบบ uniform r และให้ $F(x)=r$ ซึ่งเราจะเห็นได้ว่า x สามารถหาค่าได้จาก $r=F(x)$ ด้วย สำหรับค่า r ค่าหนึ่งๆ เช่น r_0 ที่เรากำเนิดขึ้นมา เราจะสามารถหาค่า x ที่สอดคล้องกันได้ ในกรณีนี้คือ x_0 ซึ่งแสดงดังรูปที่ 3-1 ซึ่งค่า x_0 สามารถหาได้จากการทำ Inverse ของ $F(x)$ นั่นคือ

$$x_0 = F^{-1}(r_0) \quad (3-2)$$

และจากสมการนี้สามารถกล่าวได้ว่า ถ้าต้องการสร้างสมาชิกของ uniform random number ที่เกิดจาก $F(x)$ ที่กำหนด

$$r = F(x) = \int_{-\infty}^x f(t) dt \quad (3-3)$$



รูปที่ 3-1 cumulative distribution function

และจาก

$$F(x) = P(X \leq x)$$

จะได้ว่า

$$F(x) = P[F^{-1}(r) \leq x] \quad (3-4)$$

ดังนั้น $F^{-1}(r)$ ก็คือ ตัวแปรที่มี $f(x)$ เป็นฟังก์ชันความน่าจะเป็นซึ่งเหมือนกับกรณีสมการที่ (3-3) สำหรับ x ในเทอมของ r

สำหรับวิธี Inverse Transformation นี้มีข้อจำกัดอยู่ที่ว่า มี cumulative distribution function อยู่หลาย function ที่จะทำให้ x ให้อยู่ในรูปของ inverse transformation $F^{-1}(r)$ ยากมากหรืออาจจะทำไม่ได้เลย ในกรณีนี้เราต้องใช้ วิธีการประมาณค่าเชิงตัวเลขของ F^{-1} หรืออาจใช้วิธีการใดวิธีการหนึ่งดังที่จะกล่าวถึงต่อไป

3.1.2 วิธีการ Rejection method

ถ้า $f(x)$ มีขอบเขตจำกัด และ $a \leq x \leq b$ เราสามารถใช้วิธีการ rejection ในการกำเนิดตัวแปรเชิงสุ่ม ซึ่งมีวิธีการดังต่อไปนี้

1. ทำการ normalize ช่วงของ f โดยใช้ค่า scale factor c

$$cf(x) \leq 1 \quad a \leq x \leq b \quad (3-5)$$

2. ให้ x มีความสัมพันธ์เชิงเส้นกับ r

$$x = a + (b-a)r \quad (3-6)$$

3. กำเนิด random number ขึ้นมาหนึ่งคู่ (r_1, r_2)

4. เมื่อใดก็ตามที่มีคู่ของ random number ที่เข้าข่ายความสัมพันธ์

$$r_2 \leq c \cdot f[a + (b-a)r_1] \quad (3-7)$$

ให้ยอมรับคู่ของ random number คู่่นั้นแล้วใช้สมการ $x = a + (b-a)r_1$ เป็นตัวแปรเชิงสุ่มที่กำเนิดขึ้นมา

ทฤษฎีของวิธีการนี้กล่าวว่า ค่าความน่าจะเป็น ของ r ที่จะมีค่าน้อยกว่า $c \cdot f(x)$

คือ

$$P[r \leq c \cdot f(x)] = c \cdot f(x) \quad (3-8)$$

เนื่องจาก x อยู่ในช่วง (a,b) ตามสมการ (3-6) และจะถูก reject ถ้า $r > c \cdot f(x)$ ดังนั้นค่า probability density function ของค่า x ที่ยอมรับจะมีค่าเท่ากับ $f(x)$ ทอดี้ แต่ว่าค่าเฉลี่ยของจำนวนของการทดลองก่อนที่จะพบคู่ที่ใช้ได้จะมีค่าเท่ากับ $1/c$ ทำให้วิธีการนี้อาจจะไม่สามารถใช้ได้ดีกับ probability density function บางอย่าง

วิธีการ rejection มี 4 ขั้นตอนคือ

1. กำหนด r_1 และคำนวณค่า $g(r_1)$
2. กำหนด r_2 แล้วเปรียบเทียบกับ $g(r_1)$
3. ถ้า $r_2 \leq g(r_1)$ ให้ยอมรับ r_1 เหมือน x จาก $f(x)$ แต่ถ้า $r_2 > g(r_1)$ ให้ตัด r_1 ทิ้งแล้วทำซ้ำข้อ 1. ใหม่
4. ซ้ำขั้นตอนทั้งหมดจนกว่าค่าของ x จะได้ครบตามต้องการ

3.1.3 วิธี Composition method

วิธีการนี้เป็นวิธีการผสมโดยถ้า $f(x)$ อยู่ในรูป probability mixture ของ $g_n(x)$ ที่แสดงโดย

$$f(x) = \sum g_n(x) p_n \quad (3-9)$$

การเลือกค่า จะทำโดยการพิจารณาค่าที่เหมาะสมที่สุดเพื่อ minimizing $\sum T_n p_n$ เมื่อ T_n คือ ค่าคาดหวังของเวลาในการคำนวณสำหรับการกำเนิดตัวแปรจาก $g_n(x)$

ส่วนที่ต่อจากนี้จะเกี่ยวกับการกำเนิด ตัวแปรเชิงสุ่ม ที่มีการแจกแจงความน่าจะเป็นต่างๆ ทั้งที่เป็นแบบต่อเนื่องและแบบไม่ต่อเนื่อง โดยจะใช้คอมพิวเตอร์ในการ Simulation

3.2 The Uniform distribution [3]

ถ้า probability density function ของการแจกแจงความน่าจะเป็นแบบ Uniform จะมีค่าคงที่ในช่วง (a,b) และมีค่าเป็นศูนย์ในที่อื่นๆ เขียนเป็นสมการได้ดังนี้

$$f(x) = \begin{cases} \frac{1}{b-a}, & a < x < b \\ 0, & \text{otherwise} \end{cases} \quad (3-10)$$

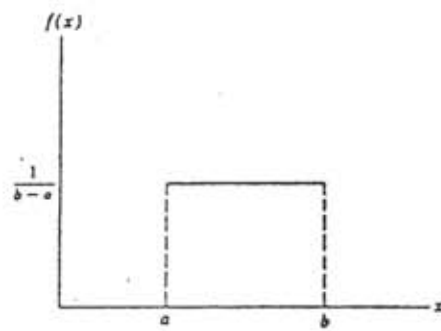
ในที่นี้ x เป็น random variable ที่นิยามในช่วง $a-b$ ดังรูปที่ 3-2 และค่า cumulative distribution function $F(x)$ สำหรับการแจกแจงความน่าจะเป็นแบบ Uniform คือ

$$F(x) = \int_a^x \frac{1}{b-a} dt = \frac{x-a}{b-a}, \quad 0 \leq F(x) \leq 1 \quad (3-1)$$

ค่าคาดหวัง(expected value) และ ค่าความแปรปรวน(variance) หาได้จากสมการ

$$Ex = \int_a^b \frac{1}{b-a} x dx = \frac{b+a}{2}$$

$$Vx = \int_a^b \frac{(x-Ex)^2}{b-a} dx = \frac{(b-a)^2}{12}$$



รูปที่ 3-2

ดังนั้นถ้าเราสามารถรู้ค่า E_x และ V_x เราจะสามารถหาค่า a และ b คือ

$$a = E_x - \sqrt{3V_x} \quad (3-14)$$

$$b = 2E_x - a \quad (3-15)$$

เพื่อที่จะ กำเนิด random number บนช่วงที่กำหนดคือ (a,b) เราต้องทำการ Inverse transformation สมการ 3-11 ตามสมการ 3-2 จะได้

$$x = a + (b-a)r \quad 0 \leq r \leq 1 \quad (3-16)$$

จากนั้นเราจะสร้างชุดของ random numbers ที่อยู่ในช่วงของความน่าจะเป็นสะสม เช่น ถ้า uniform random variates นิยามบนช่วง 0-1 random number r แต่ละตัวจะต้องจับคู่ 1 ต่อ 1 กับ x ดังรูปที่ 3-3 ถ้ามีการสร้างค่าของ r ที่เป็นแบบ 1 ต่อ 1 กับ x ค่าที่เป็นค่าเฉพาะของ $F(x)$ ที่ r_0 จะถูกจับอยู่ที่ค่า x ที่ x_0



รูปที่ 3-3

จะเห็นได้ชัดว่ากรรมวิธีนี้จะสามารถทำซ้ำจำนวนมาครั้งได้เท่าที่ต้องการ ส่วนการนำไปเขียนโปรแกรมคอมพิวเตอร์สำหรับโครงการนี้ต้องการค่าของ random variates อยู่ระหว่าง 0-1 อยู่แล้ว ดังนั้นจึงไม่จำเป็นต้องใช้สมการ (3-16) เพราะค่า x ก็คือค่า r source code ในส่วนของฟังก์ชันที่ทำหน้าที่กำเนิด uniform random variates แสดงดังนี้

```
#include <stdlib.h>
#define RAND_MAX 32767
int rand()
{
    return (int)(RAND_MAX * (double)rand() / RAND_MAX);
}
```



```

/*
 * generate uniform distribution
 */

extern double uniform()
{
    return((double)(rand() & RAND_MAX)/RAND_MAX );
}

```

เมื่อฟังก์ชัน rand() เป็นฟังก์ชันมาตรฐานในภาษา C ที่จะให้ตัวเลขสุ่มจำนวนเต็มที่มีค่าอยู่ระหว่าง 0 ถึงค่า RAND_MAX (32767) ดังนั้นฟังก์ชัน uniform() จะให้ตัวเลขสุ่มที่มีค่าตั้งแต่ 0 ถึง 1

3.3 The Exponential Distribution [3]

ในชีวิตประจำวันของเราจะเห็นได้ว่าช่วงเวลาในการเกิดของเหตุการณ์แต่ละเหตุการณ์จะเป็นแบบสุ่ม เช่นการเกิด การตาย การเกิดอุบัติเหตุ ซึ่งเหตุการณ์เหล่านี้จะเป็นอิสระต่อกัน ถ้าความน่าจะเป็นที่เหตุการณ์หนึ่งจะเกิดในช่วงเวลานั้นมีค่าน้อยมากและเหตุการณ์นั้นเป็นอิสระจากเหตุการณ์อื่น

จะได้ว่าช่วงเวลาระหว่างเหตุการณ์นั้นมีการแจกแจงความน่าจะเป็นแบบ Exponential การที่จะเป็นตัวแปรเชิงสุ่มแบบ Exponential นั้นจะต้องเป็นไปตามข้อสมมติดังนี้

1. ความน่าจะเป็นซึ่งเหตุการณ์หนึ่งเกิดขึ้นระหว่างช่วงเวลา $[t, (t+\Delta t)]$ เป็น $\alpha \Delta t$
2. α มีค่าคงที่และเป็นอิสระจากตัวแปรอื่น
3. ความน่าจะเป็นที่เกิดเหตุการณ์มากกว่าหนึ่งครั้งในช่วงเวลา $[t, (t+\Delta t)]$ มีค่าใกล้ 0

เข้า

density function ของตัวแปรเชิงสุ่ม x ที่มีการแจกแจงความน่าจะเป็นแบบ Exponential คือ

$$f(x) = \alpha e^{-\alpha x} \quad \text{เมื่อ } \alpha > 0 \text{ และ } x \geq 0 \quad (3-17)$$

ส่วน cumulative distribution function ของ x คือ

$$F(x) = \int_0^x \alpha e^{-\alpha t} dt = 1 - e^{-\alpha x} \quad (3-18)$$

ค่าคาดหวัง(Expected value) และ ความแปรปรวน (Variance)ของ x มีค่าดังต่อไปนี้

$$EX = \int_0^{\infty} x \alpha e^{-\alpha x} dx = \frac{1}{\alpha} \quad (3-19)$$

$$VX = \int_0^{\infty} (x - \frac{1}{\alpha})^2 \alpha e^{-\alpha x} dx = \frac{1}{\alpha^2} = (EX)^2 \quad (3-20)$$

เนื่องจาก Exponential distribution มีพารามิเตอร์เพียงตัวเดียวคือ α ซึ่งสามารถแสดงค่าได้เป็น

$$\alpha = \frac{1}{EX} \quad (3-21)$$

การกำเนิด Exponential random variates สามารถทำได้หลายวิธี แต่เนื่องจาก $F(x)$ อยู่ในรูปที่ไม่ซับซ้อน ดังนั้นการใช้วิธี Inverse transformation จะเป็นวิธีที่ตรง โดยสาเหตุของความเป็นสมมาตรของ Uniform distribution $F(x)$ จึงสามารถสลับเปลี่ยนกับ $1-F(X)$ ได้ นั่นคือ

$$r = e^{-\alpha x} \quad (3-22)$$

และจะได้ว่า

$$x = -\left(\frac{1}{\alpha}\right) \log(r) = -EX \log(r) \quad (3-23)$$

ดังนั้นสำหรับแต่ละค่าของ r สามารถ หาค่า x ได้โดยใช้แต่ค่าที่เป็นบวก $[0,1]$ และจะมีค่าตามสมการ (3-17) ด้วยค่าคาดหวัง EX โดยแม้ว่าวิธีการนี้จะดูง่ายแต่ต้องคิดถึงการคำนวณของ \log ที่มี power series expansion ด้วย สำหรับแต่ละ uniform variates ที่กำเนิดขึ้นมา

สำหรับการกำเนิด Exponential random variates ด้วยคอมพิวเตอร์ก็สามารถทำตาม Flow chart ในรูปที่ 3-4

Code ต่อไปนี้คือส่วนของโปรแกรมซึ่งทำหน้าที่กำเนิด Exponential variates

```
/*
 * generate Exponential distribution
 */

extern double exponent(double Ex)
{
    return(-Ex*log(uniform()));
}~
```

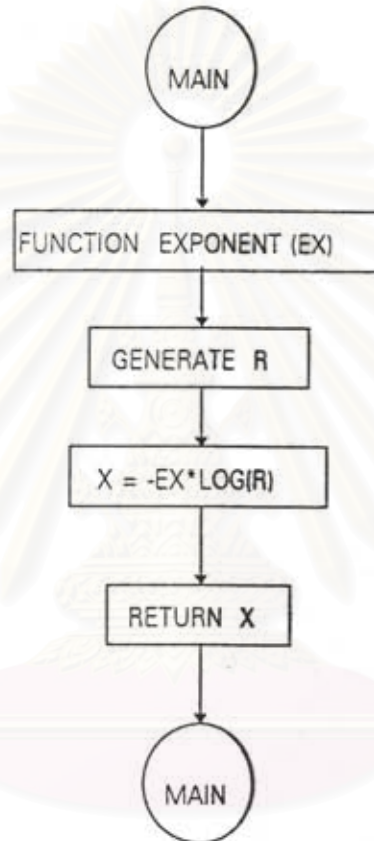
3.4 The Normal Distribution [3]

Gaussian distribution หรือ Normal distribution สามารถใช้ประโยชน์ของมันได้จาก Central Limit Theorem ซึ่งกล่าวว่า การแจกแจงความน่าจะเป็นของผลรวม N จำนวนที่เป็นอิสระต่อกันของตัวแปรเชิงสุ่ม x_i ที่มีค่าเฉลี่ยเป็น μ_i และมีค่าความแปรปรวนเป็น σ_i^2 เมื่อ N ยิ่งใหญ่

มากเท่าไร การแจกแจงความน่าจะเป็นก็จะยิ่งเข้าใกล้ Normal distribution ที่มีค่าเฉลี่ยและค่าความแปรปรวนเป็นดังนี้

$$\mu = \sum_{i=1}^N \mu_i \quad (3-24)$$

$$\sigma^2 = \sum_{i=1}^N \sigma_i^2 \quad (3-25)$$



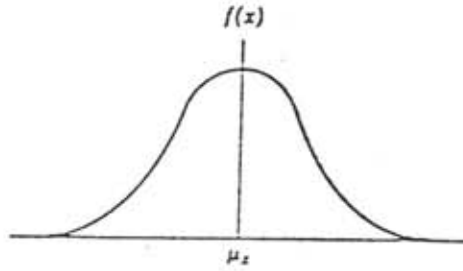
รูปที่ 3-4 Flow chart การกำเนิด Exponential variates

ดังนั้น Central Limit Theorem อนุญาตให้ใช้ Normal distribution ในการแทนผลการวัดทั้งหมดในทุกกรณีโดยไม่คำนึงว่าการแจกแจงความน่าจะเป็นของการวัดในแต่ละกรณีเป็นอย่างไร ซึ่งทำให้มักจะปรากฏข้อมูลที่มีการแจกแจงเป็นแบบ Normal ในหลายๆงานวิจัย

ถ้า ตัวแปรเชิงสุ่ม X มี density function $f(x)$ ดังสมการ

$$f(x) = \frac{1}{\sigma_x \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{x - \mu_x}{\sigma_x} \right)^2}, \quad -\infty < x < \infty \quad (3-26)$$

เมื่อ σ_x มีค่าเป็นบวก X จะเป็นการกระจายแบบ normal หรือ Gaussian ที่มีพารามิเตอร์ μ_x และ σ_x ที่มีรูปร่างของกราฟเป็นลักษณะระฆังคว่ำดังในรูปที่ 3-5



รูปที่ 3-5 กราฟการแจกแจงความน่าจะเป็นแบบ Normal

ถ้า $\mu_x = 0$ และ $\sigma_x = 1$ จะมีชื่อเรียกว่า Standard normal distribution ซึ่ง density function มีสมการดังนี้

$$f(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2} \quad -\infty < z < \infty \quad (3-27)$$

โดยที่ Normal distribution ใดๆสามารถแปลงเป็นรูปมาตรฐานโดยแทนค่า

$$z = \frac{x - \mu_x}{\sigma_x} \quad (3-28)$$

ค่าคาดหวังและค่าความแปรปรวนของ normal distribution ที่ไม่อยู่ในรูปมาตรฐานคือ

$$EX = \mu_x \quad (3-29)$$

$$VX = \sigma_x^2 \quad (3-30)$$

ในการกำเนิดตัวแปรเชิงสุ่มที่มีการแจกแจงความน่าจะเป็นแบบ normal ที่มีค่าคาดหวัง μ_x และมีส่วนเบี่ยงเบนมาตรฐาน σ_x จะทำตามหลักการทางคณิตศาสตร์ของ Central Limit Theorem คือ ถ้า r_1, r_2, \dots, r_N เป็นตัวแปรเชิงสุ่มที่เป็นอิสระแก่กันที่มีการแจกแจงความน่าจะเป็นเหมือนกัน ซึ่ง $E(r_i) = \theta$ และ $ar(r_i) = \sigma^2$ นั่นคือ

$$\lim_{N \rightarrow \infty} P \left[a < \frac{\sum_{i=1}^N r_i - N\theta}{\sqrt{N}\sigma} < b \right] = \frac{1}{\sqrt{2\pi}} \int_a^b e^{-\frac{1}{2}z^2} dz \quad (3-31)$$

เมื่อ

$$E\left(\sum_{i=1}^N r_i\right) = N\theta, \quad (3-32)$$

$$ar\left(\sum_{i=1}^N r_i\right) = N\sigma^2, \quad (3-33)$$

$$z = \frac{\sum_{i=1}^N r_i - N\theta}{\sigma\sqrt{N}}, \quad (3-34)$$

ซึ่งถ้าดูตามนิยามของ standard normal distribution และจากสมการ 3-28 แล้วจะได้ว่า z เป็นตัวแปรเชิงสุ่มแบบ standard normal

ขบวนการกำเนิดตัวแปรเชิงสุ่มแบบ normal บนคอมพิวเตอร์ จะเกี่ยวข้องกับการหาผลรวมของ ตัวแปรเชิงสุ่มที่มีการแจกแจงความน่าจะเป็นแบบ uniform จำนวน K ตัว r_1, r_2, \dots, r_K เมื่อ r_i นิยามในช่วง $0 \leq r_i \leq 1$ จากนั้นใช้ความสัมพันธ์ทางคณิตศาสตร์ ของ Central Limit Theorem และความรู้ที่ผ่านมาของการแจกแจงแบบ uniform เราจะหาได้ว่า

$$\theta = \frac{a+b}{2} = \frac{0+1}{2} = \frac{1}{2}, \quad (3-35)$$

$$\sigma = \frac{b-a}{\sqrt{12}} = \frac{1}{\sqrt{12}}, \quad (3-36)$$

$$z = \frac{\sum_{i=1}^K r_i - K/2}{\sqrt{K/12}}, \quad (3-37)$$

เมื่อเราได้ z แล้ว ตามสมการ 3-28 เราจะสามารถกำเนิดตัวแปร x ซึ่งเป็น ตัวแปรเชิงสุ่มที่มีการแจกแจงความน่าจะเป็นแบบ normal ที่มีค่าเฉลี่ยเป็น μ_x และ ส่วนเบี่ยงเบนมาตรฐานเป็น σ_x จากสมการ 3-37 เราจะได้ว่า

$$\frac{x - \mu_x}{\sigma_x} = \frac{\sum_{i=1}^K r_i - K/2}{\sqrt{K/12}}, \quad (3-38)$$

เมื่อเราแก้สมการหาค่า x เราจะได้

$$x = \sigma_x \left(\frac{12}{K} \right)^{1/2} \left(\sum_{i=1}^K r_i - \frac{K}{2} \right) + \mu_x \quad (3-39)$$

สมการ 3-39 ทำให้เราสามารถกำเนิดตัวแปรเชิงสุ่มที่มีการแจกแจงความน่าจะเป็นแบบ normal ที่มีค่าเฉลี่ยเป็น μ_x และส่วนเบี่ยงเบนมาตรฐานเป็น σ_x จากสมการ 3-39 จะเห็นได้ว่า ถ้าเราเลือกให้ $K=12$ เราจะสามารถกำจัดการคูณออกไปได้ แต่ถ้าต้องการความถูกต้องสูงเรา สามารถเลือกให้ $K=24$ แต่ก็จะทำให้ประสิทธิภาพของวิธีการ Central Limit ค่อยลงไป ในโครงการนี้ผู้จัดทำได้เลือก $K=48$ เพื่อความถูกต้องที่สูงกว่าแม้ว่าจะใช้เวลาในการคำนวณมากขึ้นก็ไม่เป็นปัญหาเพราะในปัจจุบันนี้เครื่องคอมพิวเตอร์มีความเร็วสูงมาก

Code ต่อไปนี้คือ ส่วนของโปรแกรมในการกำเนิดตัวแปรเชิงสุ่มแบบ normal

```
/*
 * generate normal distribution
 */
```

```

extern double gaussian()
{
    double sum,SD,Ex,r,x;
    double uniform();
    int i;
    sum =0.0;
    SD =1.0;
    Ex =0.0;
    for(i=1;i<=48;i++) {
        r=uniform();
        sum=sum+r;
    }
    x=SD*sqrt(12/48.)*(sum-(48./2))+Ex;
    return(x);
}

```

เมื่อตัวแปรต่างๆมีความหมายดังนี้

$$Ex = \mu_x \quad (3-40)$$

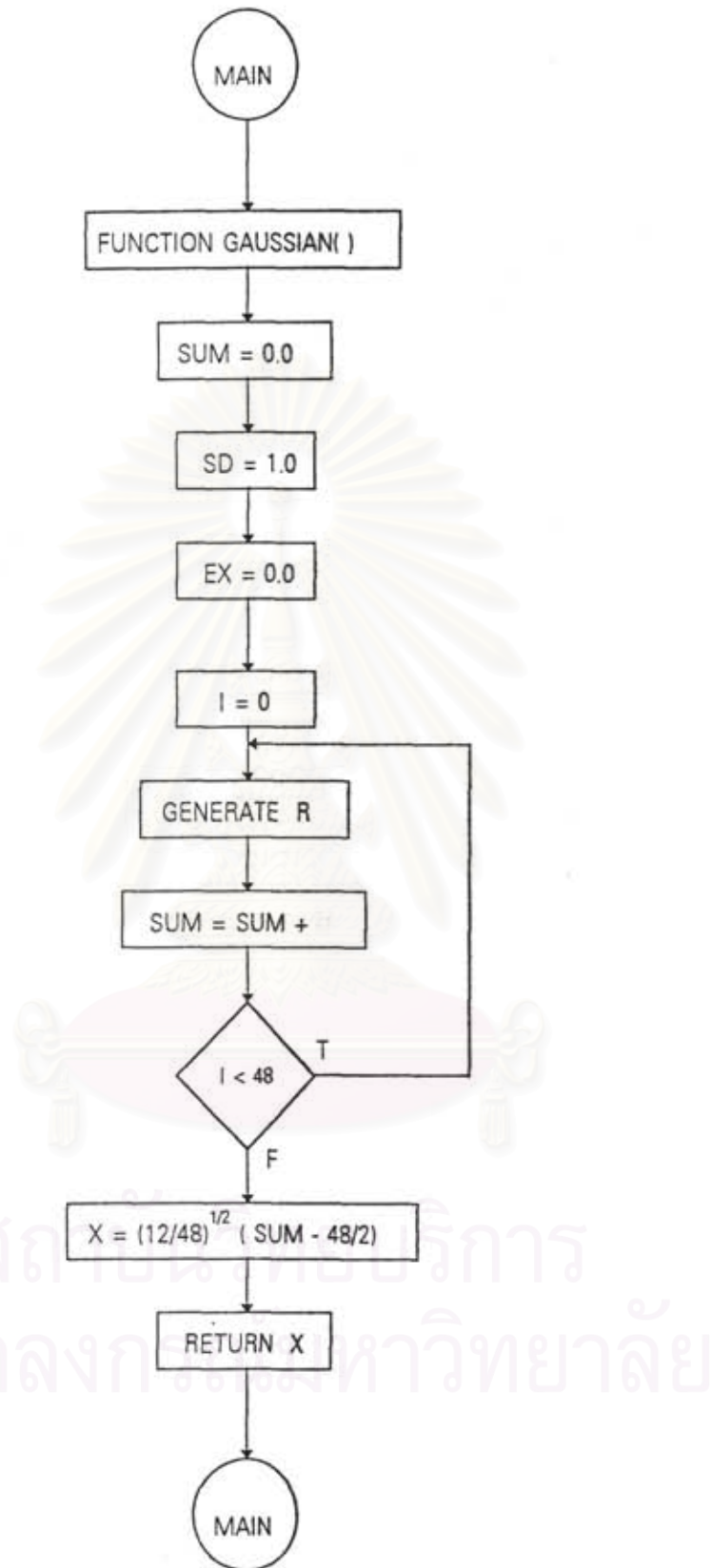
$$SD = \sigma_x \quad (3-41)$$

$$sum = \sum r \quad (3-42)$$

ในการเขียนโปรแกรมจะทำตาม Flow chart ในรูปที่ 3-6

3.5 The Multivariate Normal Distribution [3]

สำหรับเวกเตอร์ของตัวแปรเชิงสุ่มที่แต่ละส่วนประกอบของเวกเตอร์ คือตัวแปรเชิงสุ่มแบบ normal ซึ่งเรียกรวมการแจกแจงแบบนี้ว่า multivariate normal distribution ถ้าแต่ละส่วนประกอบของเวกเตอร์เชิงสุ่มนี้เป็นอิสระแก่กันเราก็สามารถใช้วิธีการที่กล่าวในหัวข้อที่แล้วได้เลย แต่ถ้าส่วนประกอบของเวกเตอร์เชิงสุ่มนี้ไม่เป็นอิสระแก่กันแล้วค่าความแปรปรวนร่วมหรือ covariance ระหว่างตัวแปรในเวกเตอร์จะมีค่าไม่เป็นศูนย์ และ variance-covariance matrix จะจำเป็นต่อการกำเนิด random normal vector



รูปที่ 3-6 แสดง Flow chart ของการกำเนิดตัวแปรเชิงสุ่มแบบ normal

เราให้ random normal vector x มีมิติเป็น m โดยมี $E(x) = \bar{\mu}$ เมื่อ $\bar{\mu}$ คือ mean vector และเราสมมติว่า x มี variance-covariance matrix V โดยที่

$$V = E[(x - \bar{\mu}) \cdot (x - \bar{\mu})^T] = \begin{bmatrix} \sigma_{11} & \dots & \sigma_{1m} \\ \dots & \dots & \dots \\ \sigma_{m1} & \dots & \sigma_{mm} \end{bmatrix}, \quad (3-43)$$

เมื่อ σ_{ii} คือค่าความแปรปรวนของส่วนประกอบตำแหน่งที่ i ของ x และ σ_{ij} คือค่าความแปรปรวนร่วมระหว่างส่วนประกอบตำแหน่งที่ i กับ ส่วนประกอบตำแหน่งที่ j ของ x ตามทฤษฎีทางสถิติแสดงได้ว่า V จะสมมาตรและมี inverse V^{-1}

probability density function ของ x คือ

$$f(x) = |2\pi V|^{-1/2} \exp \left[-\frac{1}{2} \cdot (x - \bar{\mu})^T V^{-1} (x - \bar{\mu}) \right], \quad (3-44)$$

เมื่อ $|2\pi V|$ คือ determinant ของเมตริกซ์ $2\pi V$

การกำเนิด random normal vector ที่กำหนด mean vector และ variance-covariance matrix มาให้ ถ้า z คือ standard normal vector ที่ประกอบด้วย ตัวแปรเชิงสุ่มแบบ normal ที่มีค่าเฉลี่ยเป็นศูนย์และความแปรปรวนเป็นหนึ่ง จะมีเมตริกซ์ C ซึ่งเป็น lower triangular matrix เพียงเมตริกซ์เดียวที่ทำให้

$$x = Cz + \bar{\mu} \quad (3-45)$$

ในกรณีนี้ $(x - \bar{\mu})$ มี variance-covariance matrix เป็น

$$V = C \cdot C^T \quad (3-46)$$

เพื่อที่จะได้ C จาก V สามารถทำได้โดยใช้ชุดของสมการที่มีการเรียกตัวเอง (set of recursive formulas) ในการคำนวณหาส่วนประกอบของ C

$$c_{ii} = \frac{\sigma_{ii}}{\sigma_{11}^{1/2}} \quad 1 \leq i \leq m$$

$$c_{ii} = \left(\sigma_{ii} - \sum_{k=1}^{i-1} c_{ik}^2 \right)^{1/2} \quad 1 \leq i \leq m \quad (3-47)$$

$$c_{ij} = \frac{\left(\sigma_{ij} - \sum_{k=1}^{j-1} c_{ik} c_{jk} \right)}{c_{jj}} \quad 1 < j < i \leq m$$

เนื่องจาก C เป็น lower triangular ดังนั้น $c_{ij} = 0$ สำหรับทุกค่าที่ $j > i$ ภายหลังจากเราได้ส่วนประกอบของ C แล้ว เราก็จะสามารถหาทุกส่วนประกอบของ x ได้จาก z ตามความสัมพันธ์

$$x_i = \sum_j c_{ij} z_j + \mu_i \quad (3-48)$$

การกำเนิด random vector x ที่มีค่าเฉลี่ยเป็น μ และ variance-covariance matrix เป็น V สามารถเขียนโปรแกรมที่ทำงานตามขั้นตอนต่อไปนี้

1. หาเมตริกซ์ C จาก V ตามสมการ 3-47
2. กำเนิดตัวแปรเชิงสุ่มแบบ *standard normal* ที่เป็นอิสระแก่กันจำนวน m ตัว ตามวิธีที่ได้กล่าวไปแล้ว
3. ใช้สมการ 3-45 หรือ 3-48 ก็จะได้ *normal random vector*

เนื่องจากโครงงานนี้มีการกำเนิดสัญญาณขึ้นมาสองสัญญาณเพื่อเป็นสัญญาณขาเข้า(input) ของ Echo Canceler คือ สัญญาณเสียงพูดจากด้าน Far-end ซึ่งเป็น Reference input และสัญญาณเสียงสะท้อนซึ่งเป็น Primary input ที่เกิดจากสัญญาณเสียงพูดจากด้าน Far-end ไปสะท้อนที่วงจร Hybrid ของด้าน Near-end โดยที่สัญญาณเสียงสะท้อนนี้จะมีความสัมพันธ์กับสัญญาณเสียงพูดจากด้าน Far-end ดังนั้นจึงใช้ตัวแปรเชิงสุ่มที่ไม่เป็นอิสระแก่กันสองตัวคือ x_1 และ x_2 ในการจำลองสัญญาณทั้งสองสัญญาณ ซึ่งมีวิธีการกำเนิดตามสมการต่อไปนี้

$$V = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}, \quad (3-49)$$

$$C = \begin{bmatrix} \sigma_1 & 0 \\ \rho\sigma_2 & \sigma_2\sqrt{1-\rho^2} \end{bmatrix}, \quad (3-50)$$

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = Cz + \mu = \begin{bmatrix} \sigma_1 z_1 \\ \sigma_2(\rho z_1 + \sqrt{1-\rho^2} \cdot z_2) \end{bmatrix} + \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \quad (3-51)$$

เมื่อ

$$E(x_1) = \mu_1, E(x_2) = \mu_2$$

$$Var(x_1) = \sigma_1^2, Var(x_2) = \sigma_2^2$$

$$Cov(x_1, x_2) = \rho\sigma_1\sigma_2$$

ρ คือ correlation coefficient

Code ต่อไปนี้คือส่วนของโปรแกรมที่ทำการกำเนิดตัวแปรเชิงสุ่มสองตัวที่มีค่า correlation coefficient = Cor

/*

* generate Multivariate Normal distribution

*/

extern void multivar(float Var1,float Var2,float Ex1,

```

float Ex2,float Cor,double *x1,
double *x2)
{
double z1,z2;

z1 = gaussian();
z2 = gaussian();
*x1 = sqrt(Var1)*z1+Ex1;
*x2 = sqrt(Var2)*(Cor*z1+sqrt(1-Cor*Cor)*z2)+Ex2;
}

```

3.6 การจำลองสัญญาณเสียงพูดและเสียงสะท้อน

การจำลองสัญญาณเสียงพูดต้องอาศัยการกำเนิดตัวแปรเชิงสุ่มที่มีการแจกแจงความน่าจะเป็นแบบต่างๆตามที่ได้กล่าวถึงไปแล้ว ซึ่งขั้นตอนในการจำลองสัญญาณเสียงพูดมีดังต่อไปนี้

1. สร้างพัลส์กระตุ้นเสียงตามสมการ

$$s_0(k) = \exp \left\{ \frac{\left[k + \left(\frac{M+1}{2} \right) \right]^2}{2\tau^2} \right\} \cdot \sin(2\pi \cdot F_n \cdot kT)$$

เมื่อ $k = 1, 2, \dots, M$

$= 0$ ที่ k ค่าอื่นๆ

(3-52)

โดยที่

τ คือ standard deviation ของ Gaussian envelop

F_n คือ ความถี่กลางของเสียงที่จะจำลอง

k คือ ดัชนีบอกตำแหน่งที่ของแต่ละตัวอย่างที่จำลอง

M คือ จำนวนตัวอย่างทั้งหมด

T คือ คาบเวลาในการสุ่มตัวอย่าง

ซึ่งสมการนี้ก็คือผลคูณระหว่าง สมการของ Gaussian envelop กับ สมการของสัญญาณ sine ที่มี ความถี่เป็น F_n ผลการจำลองพัลส์กระตุ้นเสียงแสดงในรูปที่ 3-7

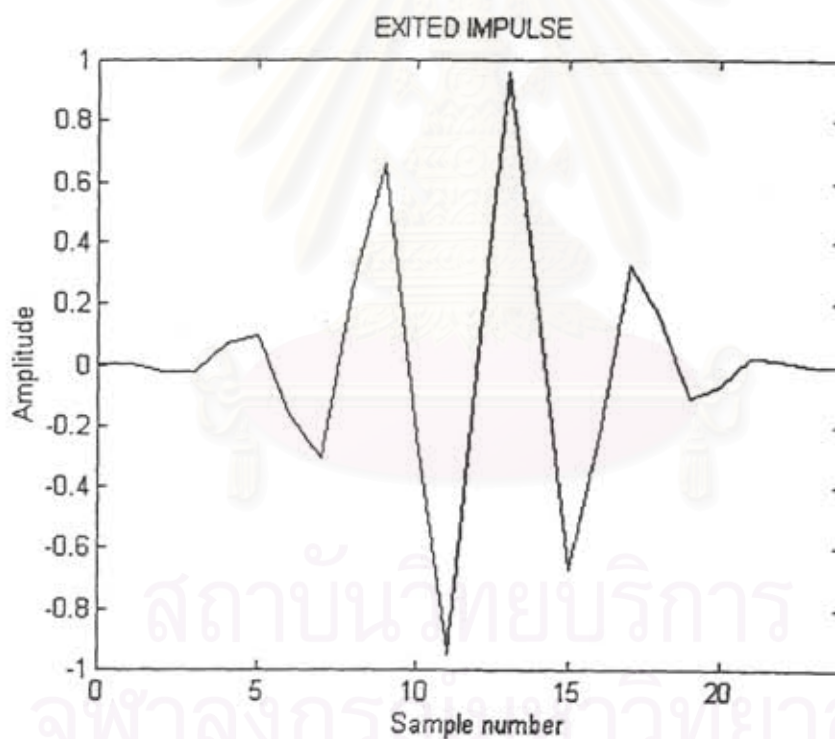
2. สร้างตัวสะท้อนพัลส์กระตุ้นเสียง โดยขนาดของตัวสะท้อนเป็นตัวแปรเชิงสุ่มที่มีการแจกแจงความน่าจะเป็นแบบ normal และช่วงเวลาในตัวสะท้อนห่างกันเป็นตัวแปรเชิงสุ่มที่มีการ

แจกแจงความน่าจะเป็นแบบ exponential ผลการจำลองตัวสะท้อนพัลส์กระตุ้นเสียงแสดงในรูปที่ 3-8

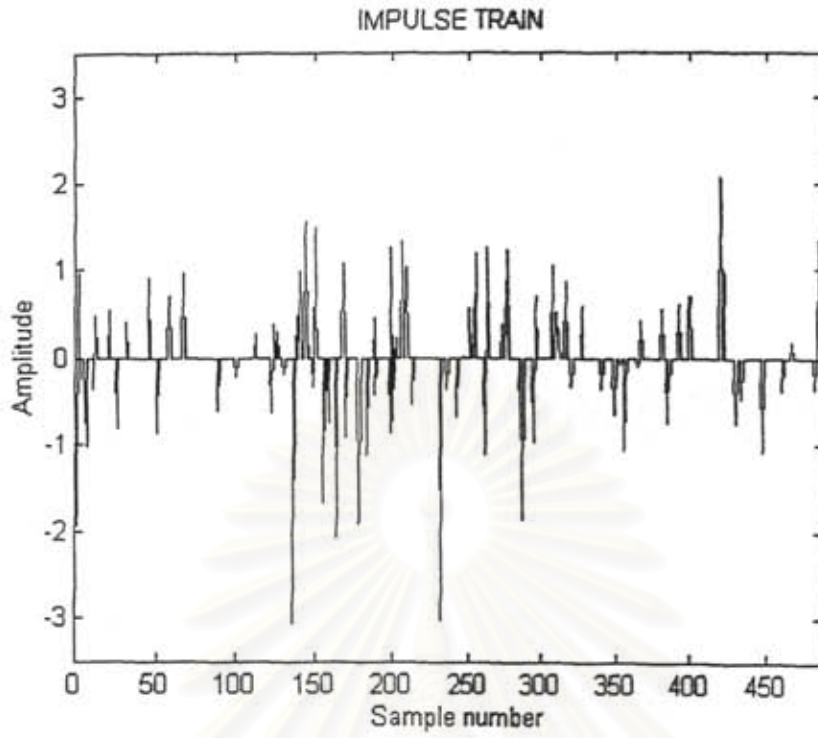


3. นำพัลส์กระตุ้นเสียงที่สร้างขึ้นในข้อ 1. มา convolution กับตัวสะท้อนที่สร้างขึ้นในข้อ 2.
2. ผลที่ได้ก็คือสัญญาณเสียงที่จำลองได้ แสดงดังรูปที่ 3-9

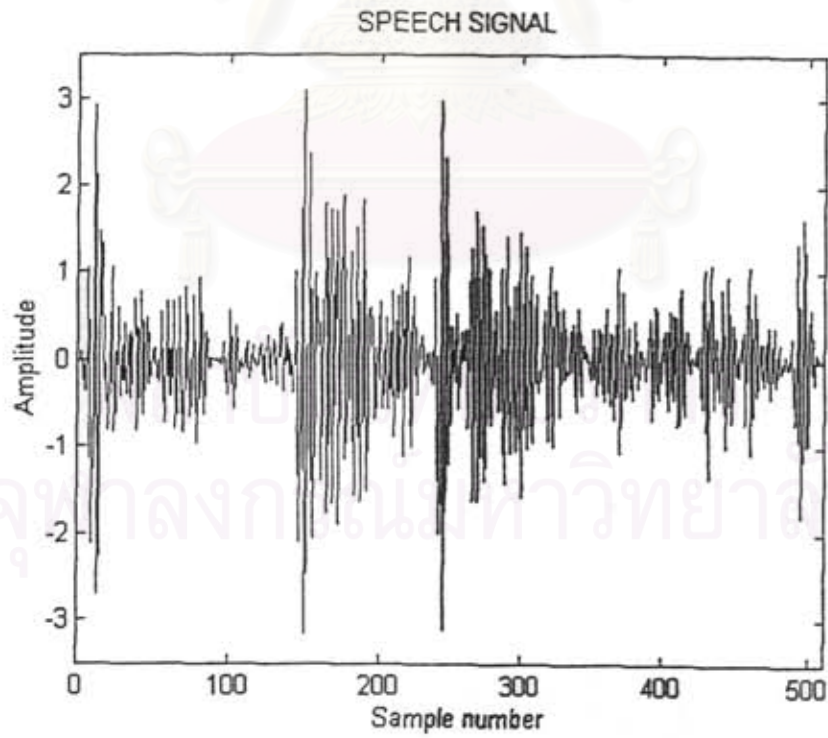
ส่วนการจำลองเสียงสะท้อนและเสียงพูดที่เป็นreference input ก็มีวิธีการเช่นเดียวกับการจำลองเสียงพูดธรรมดา แต่จะแตกต่างกันตรงที่ขั้นตอนที่สอง ก็จะมีตัวสะท้อนพัลส์กระตุ้นเสียง 2 ตัวโดยขนาดของตัวสะท้อนจะสร้างโดยใช้วิธีการของ multivariate normal distribution โดยจะมีสัมประสิทธิ์การสะท้อนคือค่า correlation coefficient เป็นตัวกำหนดความสัมพันธ์ระหว่างเสียงพูดและเสียงสะท้อนว่ามีความสัมพันธ์กันมากหรือน้อยส่วนช่วงเวลาในตัวสะท้อนห่างกันยังคงเป็นตัวแปรเชิงสุ่มที่มีการแจกแจงความน่าจะเป็นแบบ exponential เหมือนเดิม



รูปที่ 3-7 พัลส์กระตุ้นเสียงที่จำลองได้



รูปที่ 3-8 ขบวนการพัลส์ที่จำลองได้



รูปที่ 3-9 สัญญาณเสียงที่กำเนิดได้

บทที่ 4

การจำลอง Echo canceler บนคอมพิวเตอร์

จากบทที่ 2 เราได้ศึกษาถึง algorithm ของ Echo canceler ในบทนี้จะเป็นการนำ algorithm ที่ศึกษามาแล้วมาเขียนโปรแกรมเพื่อจำลองการทำงานของ Echo canceler บนคอมพิวเตอร์

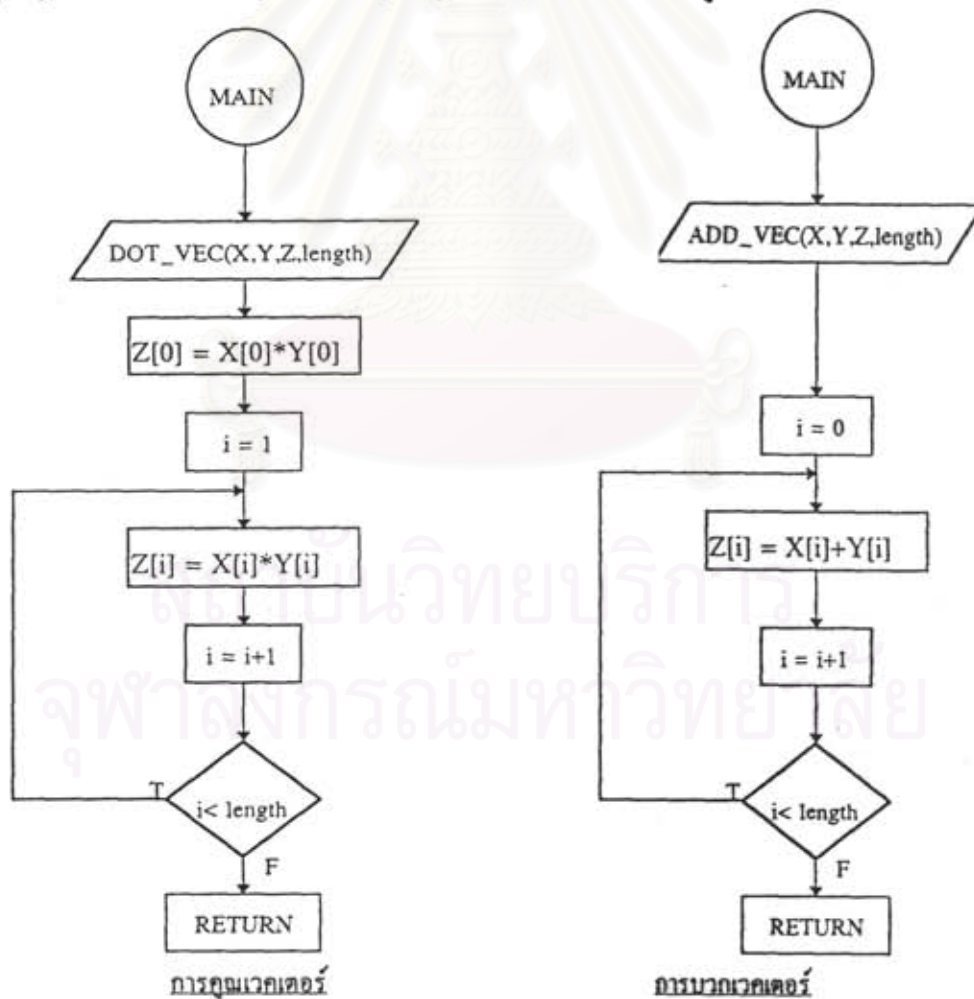
4.1 การจำลอง Adaptive filter

Adaptive filter ที่ใช้ในโครงงานนี้เขียนโปรแกรมจำลองขึ้นมาโดยใช้ LMS algorithm ตามเอกสารอ้างอิงหมายเลข[4] ซึ่ง algorithm เป็นไปตามสมการ (2-9) และ (2-20) คือ

$$\varepsilon_j = d_j - X_j^T W = d_j - W^T X_j \quad (2-9)$$

$$W_{j+1} = W_j + 2\mu \varepsilon_j X_j \quad (2-20)$$

เนื่องจาก X และ W เป็นเวกเตอร์ดังนั้นจึงจำเป็นต้องเขียนโปรแกรมเพื่อทำการคูณเวกเตอร์(ในสมการ(1-9))และบวกเวกเตอร์(ในสมการ(1-20)) ซึ่งมี flow chart ดังในรูปที่ 4-1



รูปที่4-1 flow chart ของการคูณและการบวกเวกเตอร์

ส่วน source code ของโปรแกรมนี้แสดงในโปรแกรมที่ 4-1

```

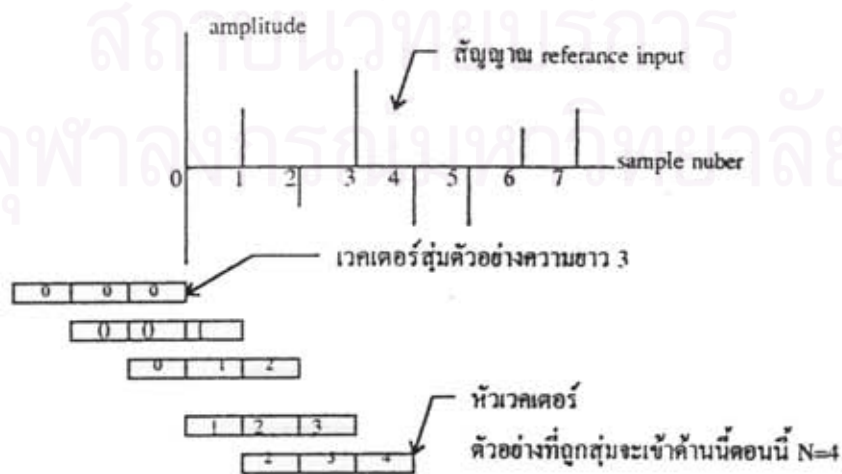
/** Function นี้จะหาผลคูณแบบคอตระหว่างเวกเตอร์ x และ y แล้วใส่ผลไว้ที่ z**/
#define DOT_VEC(x,y,z,length) { \
    double *_PTA = x ;\
    double *_PTB = y ;\
    int _IX;\
    z = (*_PTA++)*(*_PTB++);\
    for(_IX=1;_IX<(length);_IX++)\
        z += (*_PTA++)*(*_PTB++);\
}

/** Function นี้จะหาผลบวกระหว่างเวกเตอร์ x และ y แล้วใส่ผลไว้ที่ z **/
extern void ADD_VEC(double *x,double *y,double *z,int length)
{ int i;
  for(i=0;i<length;i++)
    z[i] = x[i]+y[i];
}

```

โปรแกรมที่ 4-1 source code ของโปรแกรมที่ทำการคูณและบวกเวกเตอร์

นอกจากนั้นต้องเขียนโปรแกรมเพื่อทำการสุ่มตัวอย่างสัญญาณเข้าสู่ reference input ของ Echo canceler ซึ่งจะสุ่มตัวอย่างสัญญาณจำนวน K ตัวอย่าง เมื่อ K คือความยาวของเวกเตอร์นำหนักถ่วง (ในโปรแกรมใช้ตัวแปร taps_of_weight แทน K) หลักการทำงานของโปรแกรมอธิบายโดยใช้รูปที่ 4-2



รูปที่ 4-2

จากรูปจะเห็นได้ว่าเวกเตอร์สุ่มตัวอย่างจะเลื่อนไปครั้งละหนึ่งตัวอย่างและตัวอย่างที่เข้าไปก่อนจะออกมาก่อน ซึ่งสามารถเขียนความสัมพันธ์ได้ดังนี้

$$\text{Input_filter}[0] = \text{Ref_input}[N]$$

$$\text{Input_filter}[1] = \text{Ref_input}[N-1]$$

.....

$$\text{Input_filter}[K-1] = \text{Ref_input}[N-K+1]$$

เมื่อ N คือ ตำแหน่งของสัญญาณ reference input ตำแหน่งล่าสุดที่ถูกสุ่ม

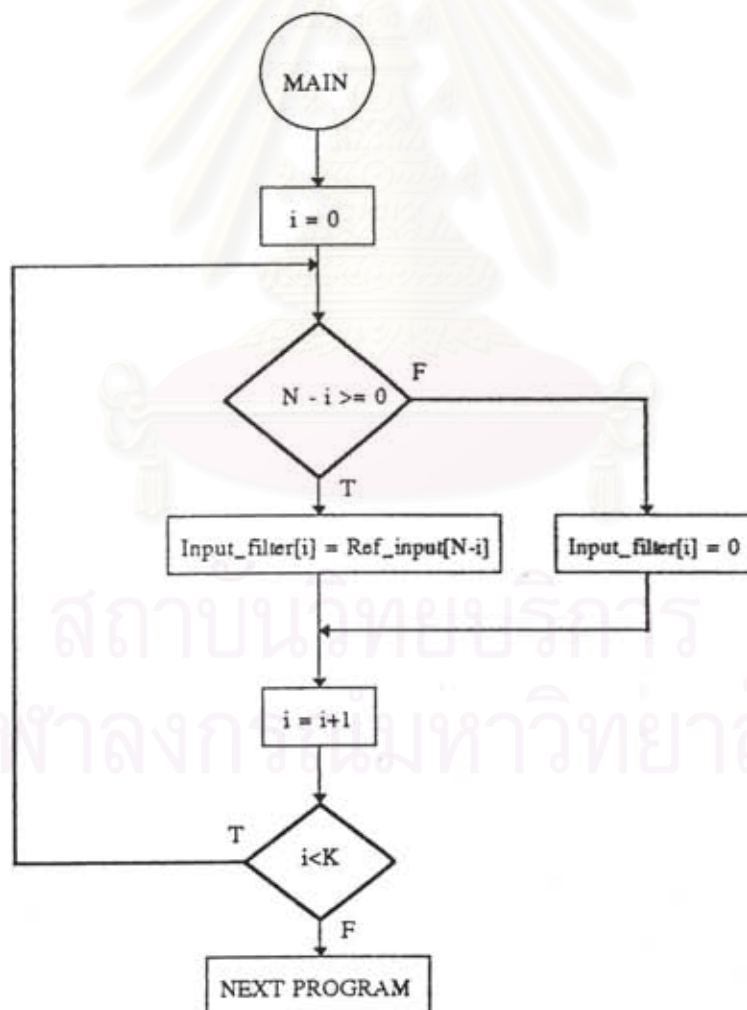
K คือ ความยาวของเวกเตอร์นำหนักถ่วง

Input_filter คือ เวกเตอร์สุ่มตัวอย่างเข้า Echo canceler

Ref_input คือ เวกเตอร์ของสัญญาณ reference input

ในกรณีที่ $N < K$ เราจะให้ค่าของ Input_filter ที่ตำแหน่ง i ซึ่ง $i > N$ มีค่าเป็นศูนย์

จากความสัมพันธ์นำมาเขียน flow chart ได้ดังรูปที่ 4-3



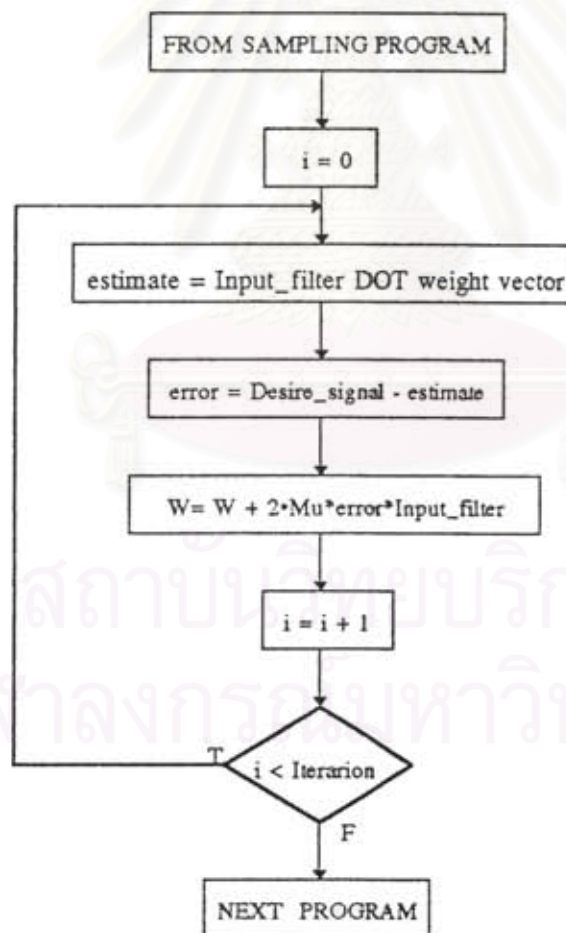
รูปที่ 4-3 แสดง flow chart ของการสุ่มตัวอย่าง

source code ของโปรแกรมส่วนนี้แสดงในโปรแกรมที่ 4-2

```
for(i=0;i<taps_of_weight;i++) {  
    if((N-i) >= 0) {  
        Input_filter[i] = Far_end_signal[N-i];  
    } else {  
        Input_filter[i] = 0;  
    }  
}
```

โปรแกรมที่ 4-2 source code ของโปรแกรมสุ่มตัวอย่างสัญญาณ reference input

เมื่อเราทำการเขียนโปรแกรมส่วนประกอบต่างๆของ Echo canceler ตามที่ได้กล่าวไปแล้ว ขั้นตอนต่อไปคือเขียนโปรแกรมหัวใจของ Echo canceler ก็คือโปรแกรม adaptive filter ซึ่งมี flow chart ดังรูปที่ 4-4 และ source code แสดงในโปรแกรมที่ 4-3



รูปที่ 4-4 flow chart ของ adaptive filter

```

for(j=0;j<Iteration;j++) ( /*LMS ALGORITHM*/
    DOT_VEC(W,Input_filter,estimate,taps_of_weight);
    error = Desire_signal[N]-estimate;
    SCAL_VEC(2*Mu*error,Input_filter,temp,taps_of_weight);
    ADD_VEC(W,temp,W,taps_of_weight);
)

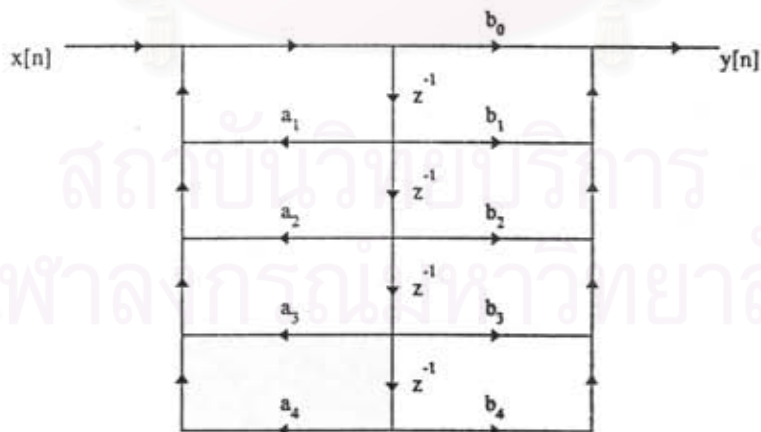
```

โปรแกรมที่ 4-3 source code ของ adaptive filter

หลังจากเขียนโปรแกรมส่วนประกอบต่างๆของ Echo canceler เรียบร้อยแล้วจากนั้นก็ทำการนำส่วนต่างๆเหล่านั้นมารวมกันเป็นโปรแกรม Echo canceler

4.2 การจำลอง fixed parameters IIR filter

เพื่อให้สัญญาณเสียงที่ออกจาก Echo canceler มีความถี่ในช่วงการตอบสนองความถี่ของสายโทรศัพท์คือ 300-3400 Hz จึงได้มีการจำลอง filter ขึ้นมาตัวหนึ่งซึ่งเป็นแบบ fixed parameters IIR [2] แล้วนำสัญญาณที่ออกจาก Echo canceler มาผ่าน filter ตัวนี้ก่อนที่สัญญาณจะออกจากระบบ Echo canceler แต่อย่างไรก็ตาม filter นี้จะใช้หรือไม่ก็ได้แล้วแต่ความประสงค์ของผู้ใช้ ซึ่งจะกล่าวถึงวิธีใช้ในภายหลัง สำหรับพารามิเตอร์ของ filter นี้ได้มาจากโปรแกรมช่วยออกแบบ digital filter [10]ของห้องปฏิบัติการ Digital signal processing ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย โครงสร้างของ filter ในแต่ละ section ซึ่งมี order 4 เป็นดังรูปที่ 4-5

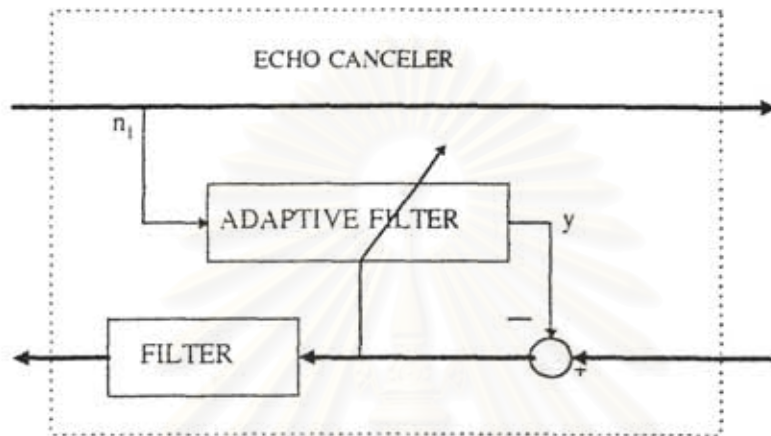


รูปที่ 4-5

difference equations ของระบบนี้คือ

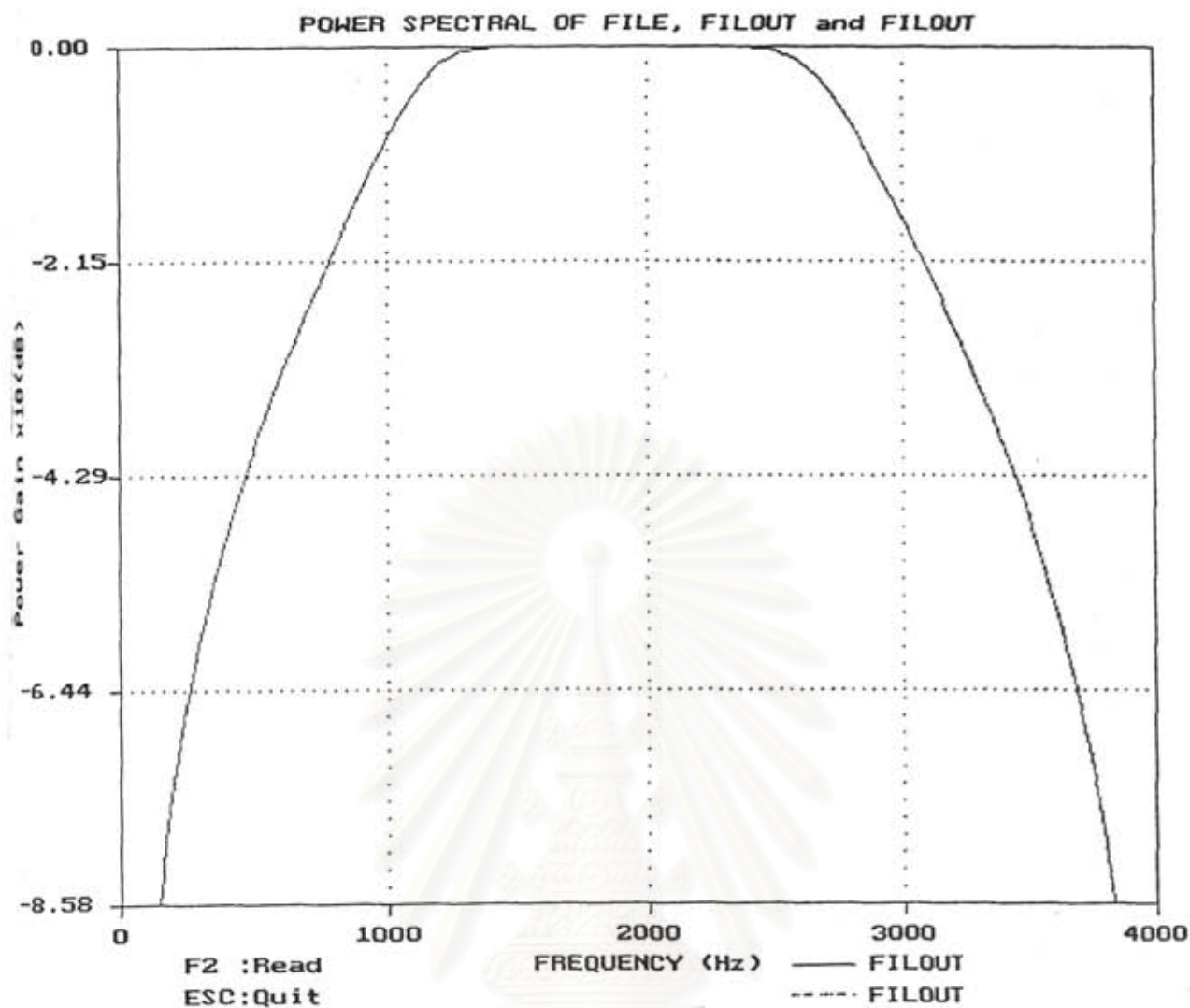
$$y[n] = a_1y[n-1] + a_2y[n-2] + a_3y[n-3] + a_4y[n-4] + b_0x[n] + b_1x[n-1] + b_2x[n-2] + b_3x[n-3] + b_4x[n-4] \quad (4-1)$$

จากสมการนี้ก็สามารถนำไปเขียนโปรแกรมได้ ซึ่งค่าที่ได้จากโปรแกรมออกแบบ filter จะต้องกลับเครื่องหมายของ a ทุกตัวเป็นเครื่องหมายตรงข้าม ส่วน b ไม่ต้องกลับเครื่องหมาย



รูปที่ 4-6 โครงสร้างของ Echo canceler ในกรณีที่มี filter ที่ขาออก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4-7 ผลตอบสนองเชิงความถี่ของ filter

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 5

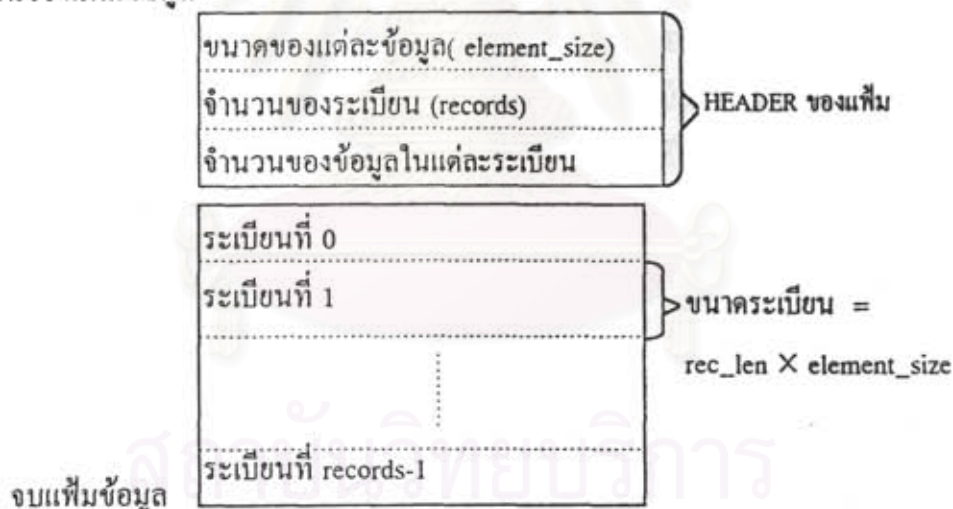
โปรแกรมอ่าน-เขียนข้อมูลและติดต่อกับผู้ใช้

เพื่อการทำงานกับ โปรแกรมนี้ได้อย่างสมบูรณ์ผู้จัดทำโครงการจึงได้ทำการเขียนโปรแกรมอำนวยความสะดวกให้กับผู้ใช้งานซึ่งแยกได้เป็นส่วนของการอ่าน-เขียนข้อมูลลงดิสก์กับส่วนการติดต่อกับผู้ใช้ [1,6,9]

5.1 การเก็บข้อมูลลงดิสก์

ภาษา C มีฟังก์ชันในการอ่าน-เขียนข้อมูลของเสียงที่สร้างขึ้นหรือเสียงที่ได้จาก Echo canceler จาก ดิสก์ โดยเราจะต้องจัดรูปแบบการเก็บข้อมูลให้เหมาะสม สำหรับโครงการนี้มีรูปแบบการเก็บข้อมูลแสดงดังรูปที่ 5-1 ซึ่งการเข้าถึงข้อมูลในดิสก์สามารถทำได้โดยการใช้ฟังก์ชันมาตรฐานที่มีอยู่ในภาษา C คือฟังก์ชัน fread() และ fwrite()

จุดเริ่มต้นของแฟ้มข้อมูล



รูปที่ 5-1

รายละเอียดของข้อมูลจะถูกเก็บอยู่ในส่วนของ HEADER ซึ่งเป็นข้อมูลแบบ structure ในภาษา C ที่นิยามไว้ดังนี้

```
typedef struct {  
    unsigned char element_size; /* ขนาดของแต่ละข้อมูล */  
    unsigned short int records; /* จำนวนระเบียน */
```




```

        unsigned short int rec_len; /* จำนวนข้อมูลในแต่ละ
ระเบียบ */

```

```

    } HEADER;

```

การใช้โครงสร้าง(Structure) HEADER ทำให้เข้าใจการทำงานและสามารถปรับปรุงแก้ไขได้ง่าย ในแต่ละฟังก์ชันที่กระทำการเกี่ยวกับข้อมูลบนดิสก์จะส่งข่าวสารของแต่ละแถมข้อมูลกลับไปยังโปรแกรมที่เรียก ฟังก์ชันนั้นๆ ข่าวสารนี้จะมีลักษณะเป็นโครงสร้างที่มีนิยามดังนี้

```

typedef struct {
        unsigned char element_size; /* ขนาดของแต่ละข้อมูล */
        unsigned short int records; /* จำนวนระเบียบ */
        unsigned short int rec_len; /* จำนวนข้อมูลในแต่ละระเบียบ */
        char *name; /* pointer ที่ชี้ไปยังชื่อแถมข้อมูล */
        FILE *fp; /* pointer ที่ชี้ไปยังโครงสร้างชนิด FILE
*/
    } ECHO_FILE;

```

ตัวแปร 3 ตัวแรกในโครงสร้าง ECHO_FILE เหมือนกับใน HEADER ส่วน 2 ตัวแปรสุดท้ายของ ECHO_FILE จะเป็น pointer ที่ชี้ไปยังชื่อแถมข้อมูล และ เป็น pointer ที่ชี้ไปยังโครงสร้าง FILE ของแถมข้อมูล ฟังก์ชัน open_write และ open_read เป็นฟังก์ชันในการเปิดแถมข้อมูลในดิสก์เพื่อทำการเขียนและอ่านข้อมูล ตามลำดับ

5.2 การเปิดแถมข้อมูล

โปรแกรมที่ 5-1 แสดง source code ของฟังก์ชัน open_write ฟังก์ชันนี้จะใช้ในการเปิดแถมข้อมูลและสร้างส่วนของ HEADER ฟังก์ชัน open_write จะไม่มีการส่งค่าใดๆกลับแต่จะทำให้ pointer ซึ่งเป็นพารามิเตอร์ตัวแรกซึ่งชี้ไปที่โครงสร้าง ECHO_FILE มีข่าวสารของข้อมูล ฟังก์ชันนี้ต้องการพารามิเตอร์ 4 ตัวคือ

1. pointer ที่ชี้ไปยัง ECHO_FILE เราต้องจองเนื้อที่ในหน่วยความจำให้ pointer ตัวนี้ก่อนที่จะเรียกใช้ ฟังก์ชันนี้
2. pointer ที่ชี้ไปยังชื่อแถมข้อมูล
3. จำนวนระเบียบ ในโครงงานนี้ใช้เพียง 1 ระเบียบในแต่ละแถมข้อมูล
4. จำนวนข้อมูลในระเบียบหนึ่งๆ ในโครงงานนี้ใช้จำนวนข้อมูล 512 ข้อมูลใน

1 ระเบียบ

```

extern void open_write(ECHO_FILE *echo_info,char *file_name,unsigned short int
    records,unsigned short int rec_len)
{
    int status;
    /* allocate the ECHO data file structure */
    if(!echo_info) {
        printf("\nError in open_write: structure allocation, file %
s\n",file_name);
        exit(1);
    }

    /* set the basics */
    echo_info->records = records;
    echo_info->rec_len = rec_len;
    echo_info->element_size = sizeof(double);
    /* open file for binary write */
    echo_info->fp = fopen(file_name,"w+b");
    if(!echo_info->fp) {
        printf("\nError opening %s in open_write\n",file_name);
        echo_info = NULL;
        return;
    }

    /* copy and allocate file name string for the ECHO_FILE structure */
    echo_info->name = malloc(strlen(file_name) + 1);
    if(!echo_info->name) {
        printf("\nUnable to allocate file_name string in open_write\n");
        exit(1);
    }
    strcpy(echo_info->name,file_name);
}

```

โปรแกรมที่ 5-1

```

/* write header to file */
status = fwrite((char *)echo_info,sizeof(HEADER),1,echo_info->fp);
if(status != 1) {
    printf("\nError writing header of file %s\n",file_name);
    exit(1);
}
}

```

โปรแกรมที่ 5-1(ต่อ)

แฟ้มข้อมูลที่เปิดขึ้นมีจะเป็นแบบ binary เพื่อให้ขนาดของแฟ้มข้อมูลไม่ใหญ่จนเกินไป (ถ้าเป็นแบบ text แฟ้มข้อมูลจะมีขนาดใหญ่)

เมื่อเราเปิดแฟ้มข้อมูลได้สำเร็จแล้วเราสามารถบันทึกข้อมูลลงแฟ้มข้อมูลได้โดยใช้ฟังก์ชัน write_record ซึ่งจะอธิบายในหัวข้อ 5.3

ส่วนฟังก์ชัน open_read ก็จะคล้ายๆกันกับ ฟังก์ชัน open_write เว้นเพียงแต่ ฟังก์ชัน open_read มี pointer ที่ชี้ไปยังโครงสร้าง ECHO_FILE และ ชื่อของแฟ้มข้อมูล เป็นทวารามิเตอร์เพียงสองตัว ทั้งนี้เพราะว่า ข่าวสารของข้อมูลได้บรรจุอยู่ในส่วนของ HEADER แล้ว ส่วนการอ่านข้อมูลจากแฟ้มข้อมูลก็สามารถทำได้โดยการใช้ ฟังก์ชัน read_record

5.3 การเข้าถึงข้อมูลในแฟ้มข้อมูล

source code ของฟังก์ชัน read_record แสดงในโปรแกรมที่ 5-2 ฟังก์ชันนี้ใช้ฟังก์ชันมาตรฐานของภาษา C เพียงฟังก์ชันเดียวคือ fread โดยทวารามิเตอร์ของฟังก์ชันนั้นได้มาจากข่าวสารในโครงสร้าง

```

extern void read_record(char *ptr,ECHO_FILE *echo_info)
{
    int status;
    if(!echo_info) {
        printf("\nError in ECHO_FILE structure passed to read_record\n");
        exit(1);
    }

```

```

status = fread(ptr,echo_info->element_size, echo_info->rec_len,echo_info->fp);
if(status != echo_info->rec_len) {

```

โปรแกรมที่ 5-2

```

printf("\nError in read_record, file %s\n",echo_info->name);
exit(1);
}
fclose(echo_info->fp);
fflush(echo_info->fp);
}

```

โปรแกรมที่ 5-2(ต่อ)

ECHO_FILE ฟังก์ชันนี้ไม่มีการส่งค่าใดๆกลับจึงประกาศฟังก์ชันเป็น void พารามิเตอร์ตัวแรกจะเป็น pointer ที่ชี้ไปยังโครงสร้าง ECHO_FILE ของแฟ้มข้อมูลที่จะอ่าน

เมื่อเกิดความผิดพลาดในการอ่านแฟ้มข้อมูล เช่น ไม่มีแฟ้มข้อมูลตามที่ใส่ชื่อไป เป็นต้น ฟังก์ชันจะแสดงข้อความที่บอกความผิดพลาดแล้วรอให้ผู้ใช้กดปุ่มใดๆเพื่อกลับสู่โปรแกรมหลักที่เรียกใช้ฟังก์ชันนี้แล้วผู้ใช้ต้องทำการแก้ไขชื่อแฟ้มข้อมูลให้ถูกต้องเพื่อให้โปรแกรมทำงานต่อไป ส่วนฟังก์ชัน write_record ก็มีลักษณะคล้ายกับฟังก์ชัน read_record แต่จะใช้ฟังก์ชันมาตรฐาน fwrite แทน

Code ต่อไปนี้แสดงตัวอย่างการใช้ฟังก์ชัน open_write และ write_record

```

int i;
int data[1000];
ECHO_FILE *out;
/* จองเนื้อที่ในหน่วยความจำสำหรับข้อมูลชนิด ECHO_FILE */
out = (ECHO_FILE *) malloc(sizeof(ECHO_FILE));
/* เป็นแฟ้มข้อมูลเพื่อการเขียนข้อมูล 1000 ข้อมูล */
open_write(out, "output.dat", 1, 1000);
/* สร้างสัญญาณรูป sine ความถี่ 50 Hz โดยใช้ความถี่สุ่ม 200 Hz */
for( i=0 ; i<1000 ; i++ )
    data[i] = sin(2*3.1416*50*i*0.05);
write_record((char *) data_out);

```

ข้อมูลจำนวน 1000 ข้อมูลจะถูกเขียนลงแฟ้มข้อมูลที่มีชื่อว่า OUTPUT.DAT และจะสังเกตได้ว่า data ถูกทำซ้ำไปที่ข้อมูลชนิดตัวอักษรโดยใช้ (char *) เพื่อที่จะให้ตรงกันกับที่นิยามไว้ในตัวฟังก์ชัน write_record

แฟ้มข้อมูล OUTPUT.DAT สามารถอ่านได้โดยใช้ open_record และ read_record ตาม code ต่อไปนี้

```
int data_in[1000];
ECHO_FILE *in;
in = (ECHO_FILE *) malloc(sizeof(ECHO_FILE));
open_read(in, "output.dat");
read_record((char *) data_in,in);
```

ตัวแปร data_in จะเก็บข้อมูลที่อ่านได้จากแฟ้ม OUTPUT.DAT



5.4 การติดต่อกับผู้ใช้โปรแกรม

ในช่วงแรกๆของการทำโครงการนี้ผู้จัดทำโครงการไม่ได้สนใจถึงเรื่อง User Interface เลย โปรแกรมที่เขียนขึ้นในช่วงแรกๆจึงเป็นการใช้งานโดยพิมพ์คำสั่งที่ DOS prompt เช่น ถ้าจะวิเคราะห์สัญญาณใน frequency domain ก็ต้องพิมพ์คำสั่ง FFTF ที่ DOS prompt เครื่องคอมพิวเตอร์ก็จะถามชื่อแฟ้มข้อมูลที่จะต้องการวิเคราะห์ เมื่อใส่ชื่อแล้วกด ENTER คอมพิวเตอร์ก็จะแสดง frequency domain ของแฟ้มข้อมูลตามที่ใส่ชื่อไป ซึ่งในตอนแรกๆการทำเช่นนี้ก็ไม่ได้ยุ่งยากอะไร แต่ต่อมาเมื่อโปรแกรมมีหลายโปรแกรมมากขึ้นและแต่ละโปรแกรมก็มีส่วนเกี่ยวข้องกัน เช่น เมื่อเรากำเนิดสัญญาณเสียงขึ้นมาแล้วเราต้องการวิเคราะห์สัญญาณว่าตรงกับที่เราต้องการหรือไม่ นอกจากนั้นเรายังอาจจะต้องการทดลองใช้ Echo canceler ในการกำจัดเสียงสะท้อนและเราอาจจะต้องการทำอย่างอื่นอีกมากมาย การที่ต้องทำสิ่งต่างๆหลายอย่างตามที่ยกตัวอย่างไปแล้วนั้นผู้ใช้จะต้องพิมพ์คำสั่งที่ซ้ำๆกันบ่อยมากทำให้เสียเวลาเยอะโดยไม่จำเป็นและยังสร้างความเบื่อหน่ายให้กับผู้ใช้โปรแกรม ผู้จัดทำโครงการนี้จึงเริ่มสนใจในเรื่อง User Interface และทำการรวมโปรแกรมที่เขียนไว้เข้าด้วยกันแล้วเขียนโปรแกรมเพิ่มในส่วนของการเข้าถึงโปรแกรมต่างๆในรูปแบบของ MENU แถบแสง ทำให้การใช้งานโปรแกรมทำได้ง่ายขึ้นมากเพราะว่าผู้ใช้เพียงแต่เลื่อนแถบสว่างบน MENU ไปที่คำสั่งที่ต้องการแล้วกด ENTER ก็สามารถใส่คำสั่งนั้นได้โดยจะมีการพิมพ์ตัวอักษรเฉพาะชื่อแฟ้มข้อมูลและค่าตัวเลขที่จำเป็นเท่านั้น การแก้ไขพารามิเตอร์ของคำสั่งต่างๆก็สามารถทำได้โดยง่าย

5.5 วิธีการใช้งาน

ความต้องการต่ำการต่ำสุดของระบบคอมพิวเตอร์ที่ใช้กับโปรแกรมนี้

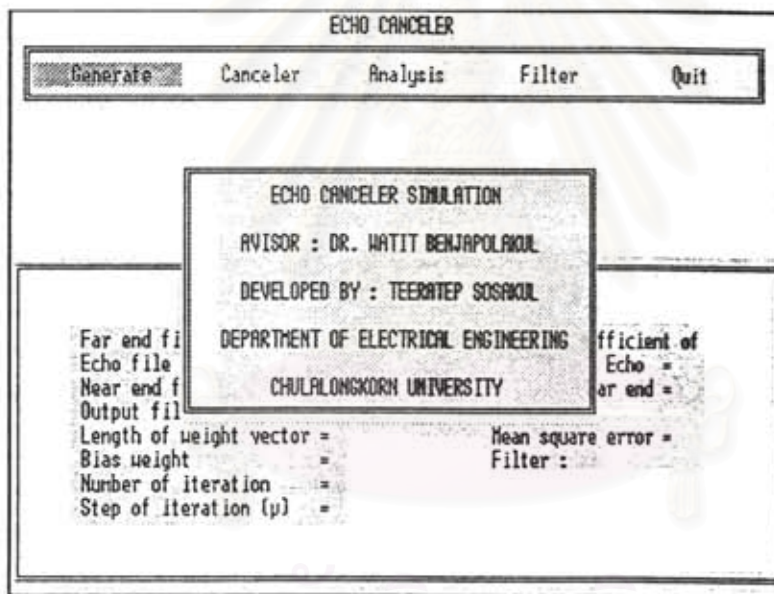
1. CPU 486DX2 66 MHz
2. ที่ว่างบน Hard disk ประมาณ 300 KB
3. RAM 4 MB
4. จอสี

5.5.1 วิธีการติดตั้ง

ทำได้ง่ายโดยการ copy ทุก file ในแผ่นดิสต์ที่มากับรายงานฉบับนี้ลง Hard disk ก็เป็นอันเสร็จการติดตั้ง

5.5.2 วิธีการใช้งาน

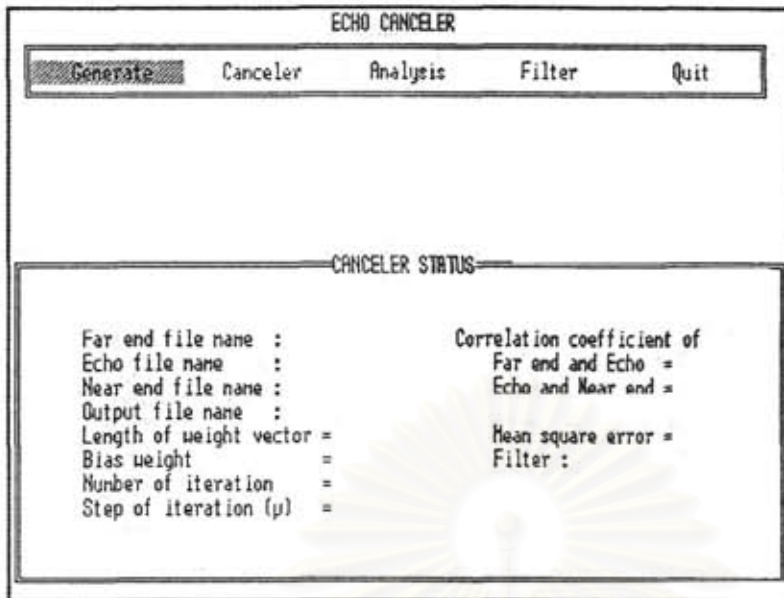
เมื่อทำการติดตั้งเรียบร้อยแล้วการใช้งานโปรแกรมก็ทำได้โดยพิมพ์ ECAN ที่ DOS prompt แล้วกด Enter หน้าจอก็จะเปลี่ยนไปเป็นดังรูปที่ 5-2



รูปที่ 5-2

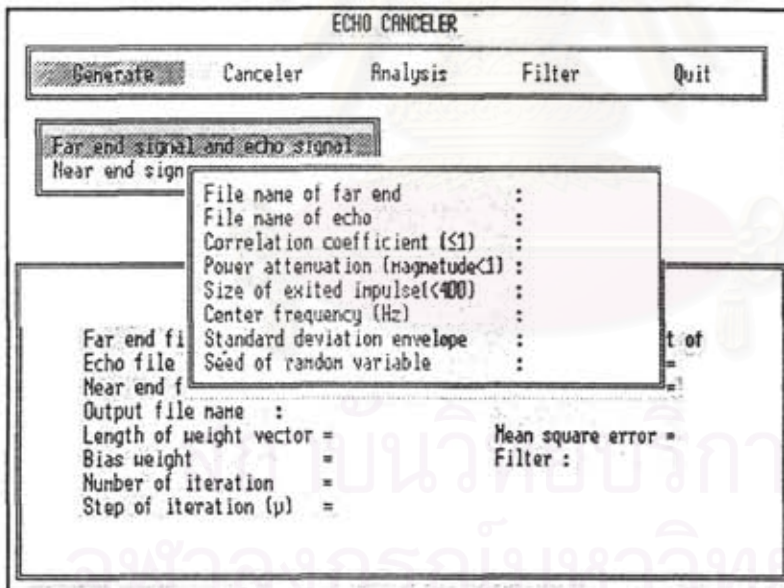
เมื่อกดแป้นใดๆเพื่อทำงานต่อ หน้าจอก็จะเปลี่ยนเป็นดังรูปที่ 5-3 ซึ่งจะแสดง MENU ของคำสั่งต่างๆการใช้คำสั่งใดก็ได้โดยเลื่อนแถบแสงไปยังคำสั่งนั้นแล้วกด ENTER ก็จะใช้คำสั่งนั้นได้ รายละเอียดของแต่ละคำสั่งมีดังต่อไปนี้

1. Generate เมื่อเลือกคำสั่งนี้จะมี MENU ย่อยให้เลือกอีกสองตัวเลือกคือ Far end signal and Echo signal และ Near end signal ซึ่งในตัวเลือกตัวแรกจะใช้ในการกำเนิดสัญญาณเสียงพูดจากด้านไกล(Far end)ซึ่งจะเป็นเสียงreference และกำเนิดเสียงสะท้อนซึ่งมีความสัมพันธ์กับเสียง reference ส่วนตัวเลือกที่สองใช้ในการกำเนิดสัญญาณเสียงที่พูดจากด้านใกล้(Near end) ซึ่งจะเข้า primary input ของ Echo canceler



รูปที่ 5-3

เมื่อเราเลือกตัวเลือกตัวแรก(โดยการใช้นิพจน์ลูกศรบน-ล่าง)ก็จะปรากฏหน้าต่างให้ใส่ค่าต่างๆดังรูปที่ 5-4



รูปที่ 5-4

ค่าต่างๆมีความหมายดังนี้

File name of far end คือ ชื่อของแฟ้มข้อมูลที่จะนำสัญญาณเสียง reference ที่กำเนิดขึ้นไปเก็บไว้มีความยาวไม่เกิน 8 ตัวอักษรและมีส่วนขยายได้ไม่เกิน 3 ตัวอักษร

File name echo คือ ชื่อของแฟ้มข้อมูลที่จะนำสัญญาณเสียง echo ที่กำเนิดขึ้นไปเก็บไว้มีความยาวไม่เกิน 8 ตัวอักษรและมีส่วนขยายได้ไม่เกิน 3 ตัวอักษร

Correlation coefficient คือ ค่าของความสัมพันธ์ระหว่างเสียง reference กับเสียง echo มีค่าน้อยกว่าหรือเท่ากับ 1 เมื่อค่ายิ่งใกล้ 1 ความสัมพันธ์ก็จะยิ่งมาก

Power attenuation คือ ค่าลดทอนของสัญญาณเสียง Echo จากสัญญาณเสียง reference มีค่าน้อยกว่า 1 คิดเป็นเท่าไม่ได้เป็น dB โดยทั่วไปค่านี้อยู่ในช่วงประมาณ 0.1-0.3

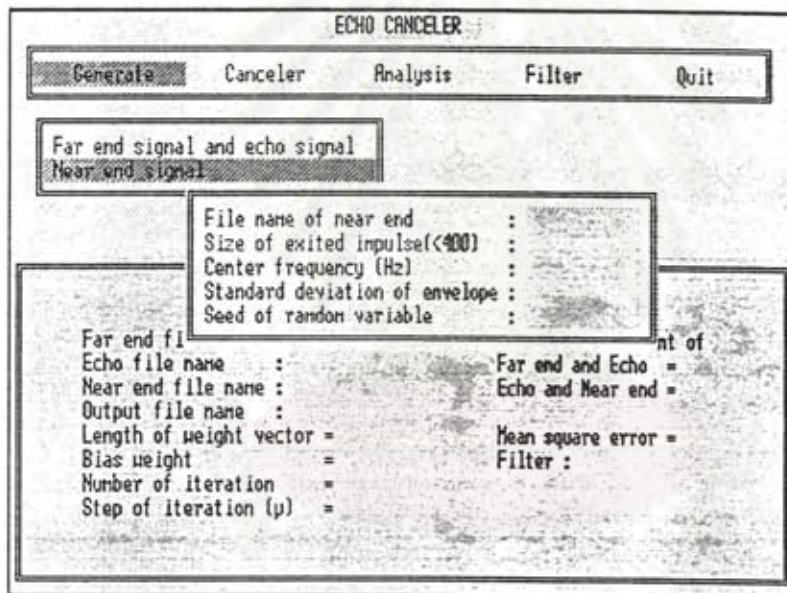
Size of exited impulse คือ ความยาวของพัลส์กระตุ้นเสียง โดยทั่วไปมีค่าอยู่ ในช่วงระหว่าง 20-60

Center frequency คือ ความถี่กลางของพัลส์กระตุ้นเสียง มีค่าได้ตั้งแต่ 0 Hz ไปจนถึง 4000 Hz

Standard deviation of envelope คือ ส่วนเบี่ยงเบนมาตรฐานของ envelope ของพัลส์กระตุ้นเสียง ถ้ามีค่ามากกว่า Size of exited impulse จะทำให้สัญญาณที่กำเนิดได้มี band width แคบในทางตรงข้าม ถ้ามีค่าน้อยกว่า Size of exited impulse ก็จะทำให้สัญญาณที่กำเนิดได้มี band width กว้าง

Seed of random variable คือ ค่าเริ่มต้นของการกำเนิดตัวแปรสุ่มมีค่าได้ตั้งแต่ 1 ถึง 32767

เมื่อเราเลือกตัวเลือกที่สองก็จะปรากฏหน้าต่างให้ใส่ค่าต่างๆดังรูปที่ 5-5



รูปที่ 5-5

ซึ่งค่าพารามิเตอร์ก็จะมี ความหมายเช่นเดียวกับของตัวเลือกแรกดังที่ได้กล่าวไปแล้ว

2. Canceler ตัวเลือกนี้จะใช้ในการจำลองระบบ ซึ่งในตัวเลือกนี้จะมี MENU ย่อยอีก 4 ตัวเลือกคือ

Input จะถามชื่อแฟ้มข้อมูลที่เป็น input ของ Echo canceler คือสัญญาณเสียงจากด้านไกลซึ่งก็คือสัญญาณ reference ,สัญญาณเสียงสะท้อน และสัญญาณเสียงจากด้านใกล้

Output จะถามชื่อของแฟ้มข้อมูลที่จะนำสัญญาณที่ออกจาก Echo canceler ไปเก็บไว้

Parameters จะถามค่าพารามิเตอร์ของ Echo canceler ได้แก่ความยาวของ weight vector ซึ่งมากก็จะยิ่งใช้เวลาในการคำนวณนาน , Bias weight คือค่าเริ่มต้นของ weight vector มักใช้ค่าเป็น 0 แต่ก็

สามารถใช้ค่าอื่นได้ , Number of iteration คือจำนวนรอบของการปรับตัวของ weight vector ในสัญญาณ input แต่ละค่า ค่ายิ่งมากก็จะยิ่งใช้เวลาในการคำนวณนาน , และ Loop gain คือค่าที่ควบคุมเสถียรภาพและอัตราการลู่เข้าของระบบ

Run เมื่อเลือกตัวเลือกนี้โปรแกรมก็จะทำการกำจัดเสียงสะท้อน ซึ่งเวลาในการคำนวณก็ขึ้นอยู่กับทางเลือกค่าพารามิเตอร์
ทุกครั้งที่เราเปลี่ยนค่าต่างๆในตัวเลือก Canceler ผลการเปลี่ยนแปลงจะดูได้จากหน้าต่าง CANCELER STATUS

3. Analysis ใช้ในการวิเคราะห์สัญญาณได้ทั้งใน Time domain และ Frequency domain ซึ่งเมื่อเลือกที่จะวิเคราะห์ใน domain ไหนแล้วในแต่ละตัวเลือกจะตามชื่อของแท็บข้อมูลที่เกี่ยวข้องกับสัญญาณที่ต้องการวิเคราะห์ เมื่อใส่ชื่อเรียบร้อยแล้วกด ENTER โปรแกรมก็จะแสดงผลการวิเคราะห์ตาม domain ที่เลือก สำหรับ Frequency domain สามารถดูได้ว่าในแต่ละความถี่นั้นมี power เท่าไรโดยการกด F2 ก็จะปรากฏเส้นตรงสีเหลืองบนกราฟ และแสดงค่า power ของสัญญาณแต่ละสัญญาณที่ความถี่ที่ตำแหน่งของเส้น สามารถเลื่อนเส้นไปมาเพื่ออ่านค่า power ได้ที่ทุกความถี่โดยการใช้นู้นลูกศร ซ้าย-ขวา เมื่อกด ESC เส้นอ่านค่าก็จะหายไป การออกจากกรวิเคราะห์สัญญาณก็ทำโดยการกด ESC เช่นกัน

4. Filter (filter ที่ใช้ในคำสั่งนี้เป็น filter แบบ fir ที่ fixed parameters) ในตัวเลือกนี้มีตัวเลือกย่อยอีก 2 ตัวเลือกคือ

Filter ON/OFF ใช้ในกรณีต้องการให้สัญญาณที่ออกจาก Echo canceler อยู่ในช่วงความถี่ 300-3400 Hz เมื่อเลือกตัวเลือกนี้ตรงตำแหน่ง Filter: ใน CANCELER STATUS จะแสดง ON หรือ OFF สลับกันไปในการเลือกแต่ละครั้ง(เมื่อเริ่มโปรแกรมใหม่ที่ตำแหน่งนี้จะไม่แสดงอะไรเลยซึ่งหมายความว่า OFF)

Filter data ใช้ในกรณีมีสัญญาณอยู่แล้วต้องการให้สัญญาณนั้นอยู่ในช่วงความถี่ 300-3400 Hz โดยการใส่ชื่อของแท็บข้อมูลที่จะเก็บสัญญาณที่ออกจาก filter แล้วกด ENTER ข้อมูลที่ใส่เข้าไปก็จะถูก filter แล้วนำผลไปใส่ในแท็บข้อมูลตามที่ใส่ชื่อไป

5. Quit เป็นตัวเลือกในกรณีที่ต้องการเลิกใช้โปรแกรมเมื่อเลือกตัวเลือกนี้จะขึ้นกรอบตัวเลือกขึ้นมาหนึ่งกรอบแล้วมีข้อความ "Press Enter to quit" ถ้าเรากด Enter ก็จะเป็นการออกจากโปรแกรมกลับสู่ DOS แต่ถ้ากด ESC กรอบข้อความก็จะหายไป

หมายเหตุ

1. เมื่อกด ESC จะเป็นการยกเลิก Menu ย่อยเฉพาะ Menu ไม่ได้รับค่าจากแป้นพิมพ์เท่านั้น เช่น Quit , Filter ON/OFF เป็นต้น ถ้าเป็น Menu ที่รับค่าจากแป้นพิมพ์เช่น Menu ย่อยในParameters จะทำการยกเลิกไม่ได้

2. การใส่ค่าต่างๆนี้ถ้าใส่ผิดแล้วต้องการแก้ไขให้ใช้เป็น Back space ในการลบตัวที่ผิดแล้วพิมพ์ค่าที่ถูกแทน แต่ถ้ากด ENTER ค่าที่ผิดไปแล้วก็ให้ใส่ค่าอื่นจนครบก่อนแล้วค่อยเข้ามาแก้ไขทีหลัง

บทที่ 6

การทดลองและสรุปผล

6.1 วิธีการทดลอง

ใช้โปรแกรมที่สร้างขึ้นกำเนิดสัญญาณเสียงพูดและเสียงสะท้อนในระบบโทรศัพท์ให้มีลักษณะต่าง ๆ กัน ในการทดลองนี้ จะใช้ค่าความถี่กลางของสัญญาณ near-end คนละค่ากันกับความถี่กลางของสัญญาณสะท้อน โดยแยกเป็น 3 กรณีคือ

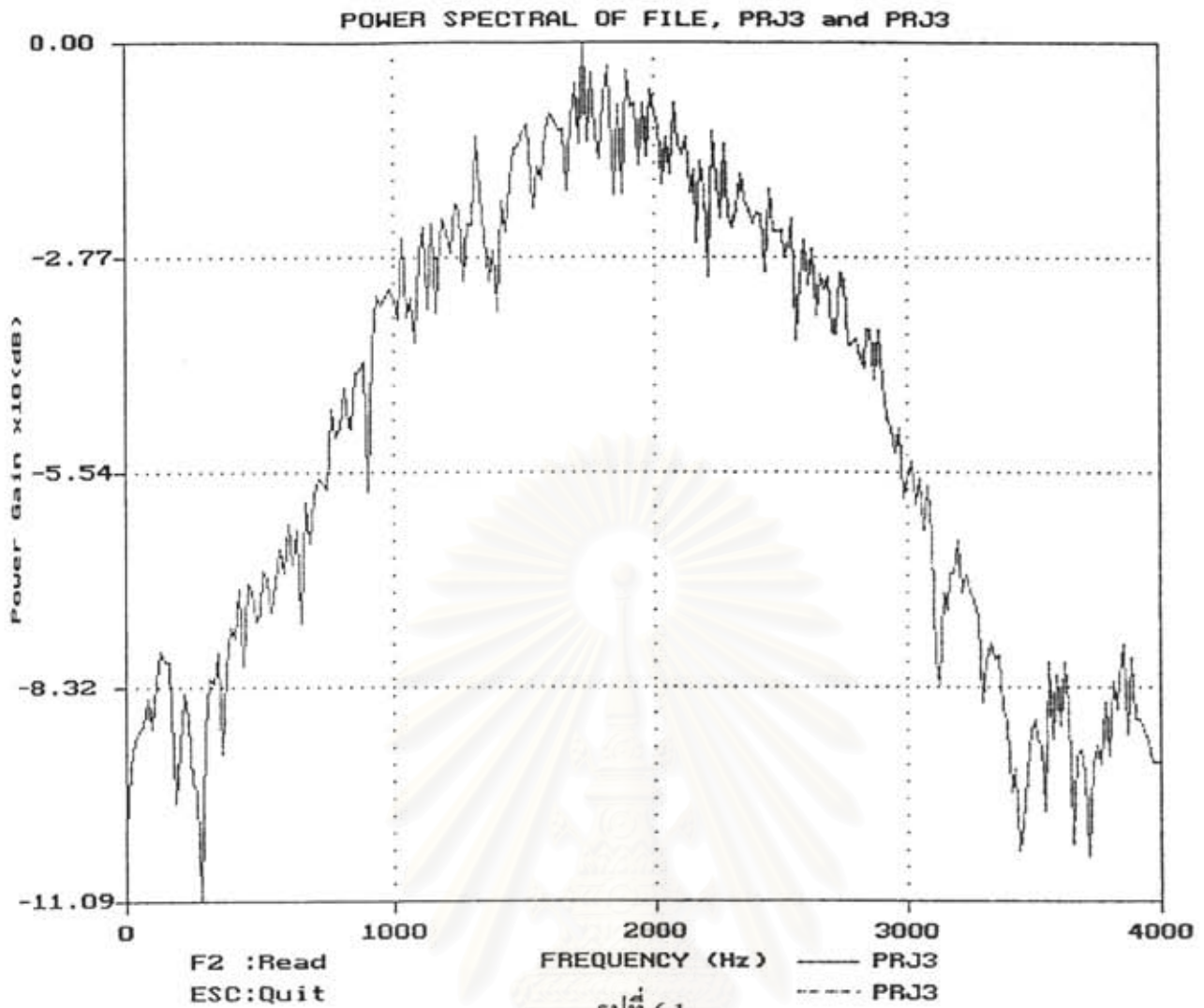
1. เสียงสะท้อนมีความถี่ต่ำ
2. เสียงสะท้อนมีความถี่สูง
3. ไม่มีเสียงสะท้อน(หมายความว่าด้าน far-end ไม่ได้พูด)

สำหรับสัญญาณด้าน near-end ใช้สัญญาณเดียวกันทุกกรณี ซึ่งมีรายละเอียดดังนี้

ชื่อเพิ่มข้อมูลของสัญญาณจาก near-end	PRJ3
ความยาวของพัลส์กระตุ้นเสียง	25 ตัวอย่าง
ความถี่กลาง	1850 Hz
ค่า standard deviation ของ envelope	13
Seed ของ random number	2000

สาเหตุที่เลือกความถี่กลางของสัญญาณเสียงจาก near-end ที่ 1850 Hz เพราะค่าความถี่กลางค่านี้ จะให้สัญญาณเสียงครอบคลุมความถี่ในช่วงการตอบสนองความถี่ของสายโทรศัพท์ (300 - 3400 Hz) สำหรับค่า standard deviation ของ envelope ถ้าเลือกค่ามากจะทำให้สัญญาณที่ได้มี band width แคบ สัญญาณ near-end ที่กำเนิดได้แสดงดังต่อไปนี้

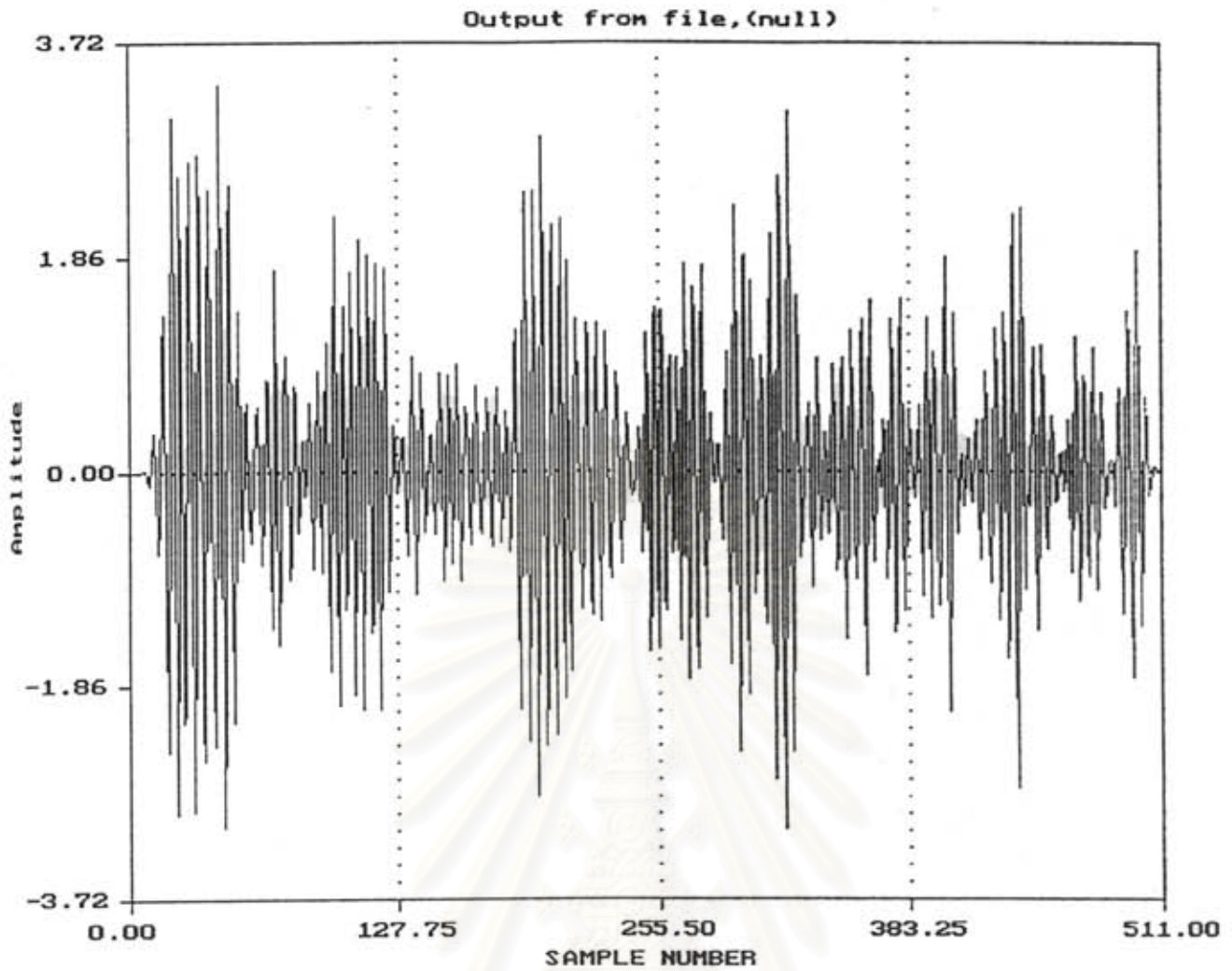
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 6-1

ผลตอบใน frequency domain ของสัญญาณเสียงพูดด้าน near-end (PRJ3)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 6-2

ผลตอบใน time domain ของสัญญาณเสียงพูดด้าน near-end (PRJ3)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

6.2 ผลการทดลอง

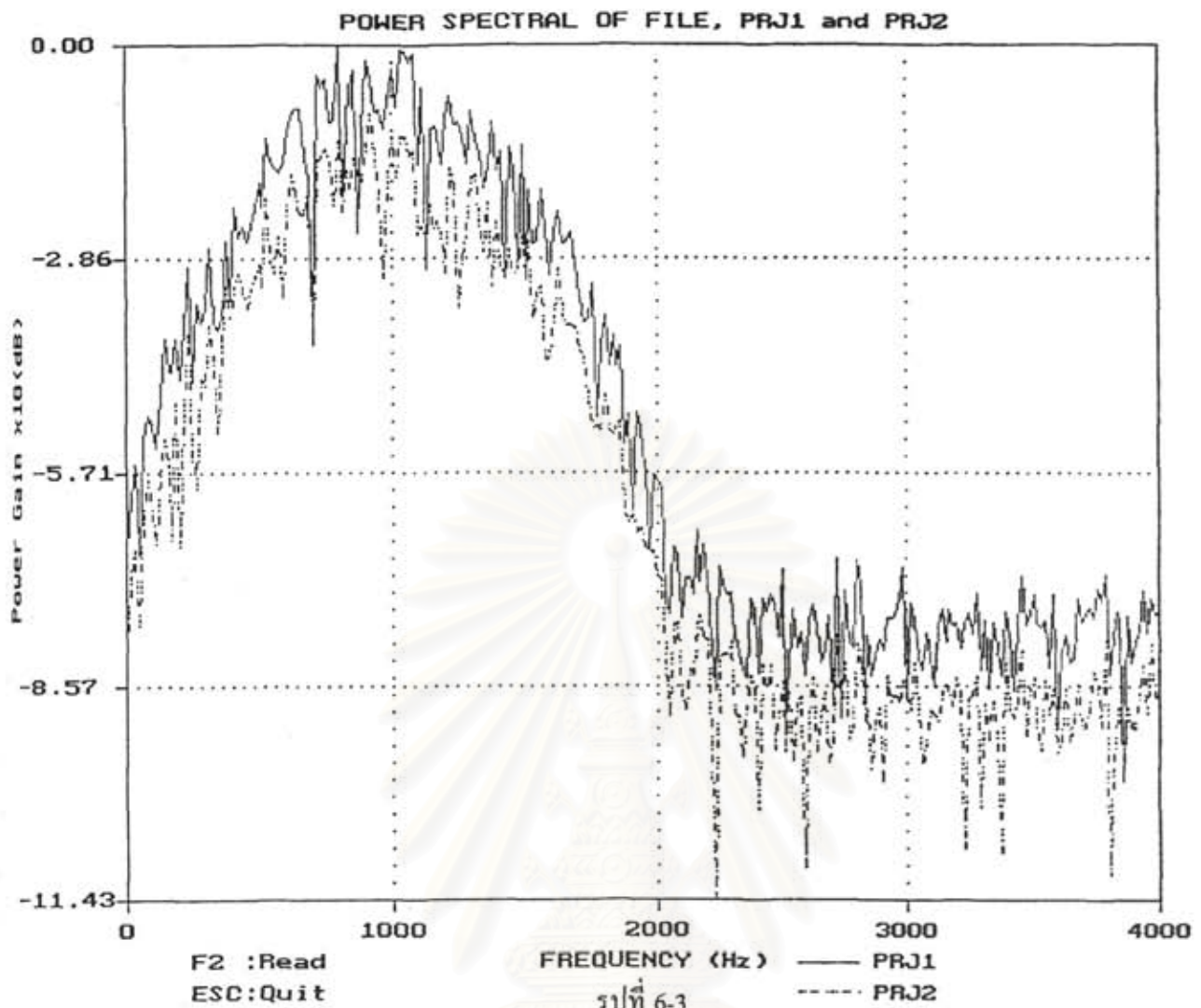
กรณีที่ 1. เสียงสะท้อนมีความดีค่า

กำเนิดสัญญาณเสียงพูดจาก far-end และ เสียงสะท้อนตามรายละเอียดดังนี้

ชื่อแฟ้มข้อมูลของสัญญาณจาก far-end	PRJ1
ชื่อแฟ้มข้อมูลของสัญญาณเสียงสะท้อน	PRJ2
Correlation coefficient ระหว่างสัญญาณ- จาก far-end กับ สัญญาณเสียงสะท้อน	0.85
การลดทอนของเสียงสะท้อนเมื่อเทียบกับสัญญาณจาก far-end	0.1 เท่า
ความยาวของพัลส์กระตุ้นเสียง	30 ตัวอย่าง
ความถี่กลาง	1000 Hz
ค่า standard deviation ของ envelope	20
Seed ของ random number	2500

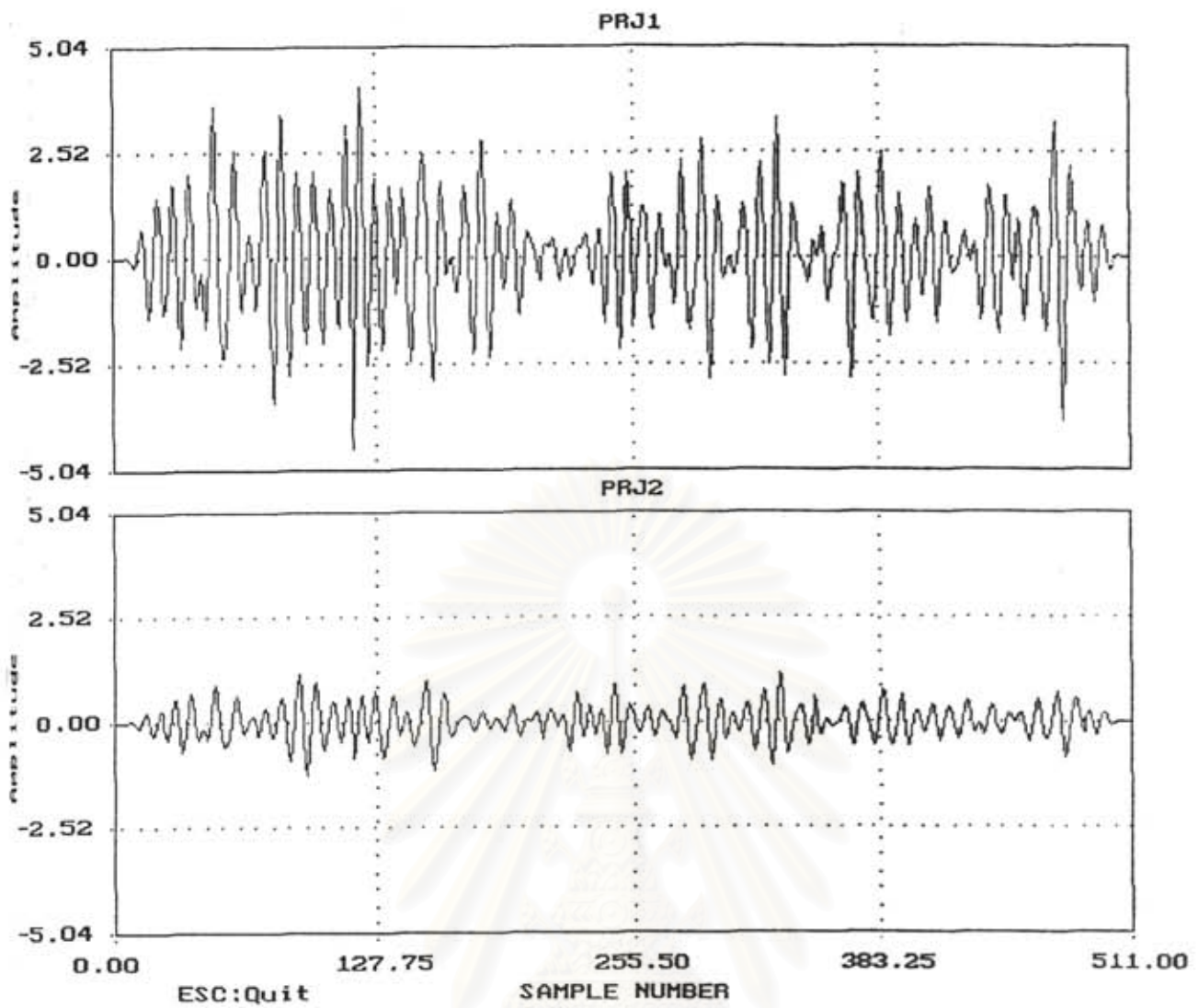
สัญญาณ far-end (PRJ1) , สัญญาณเสียงสะท้อน(PRJ2) และผลการทดลอง แสดงดังต่อไปนี้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



ผลตอบใน frequency domain ของสัญญาณเสียงพูดด้าน far-end (PRJ1) กับสัญญาณเสียงสะท้อน (PRJ2)

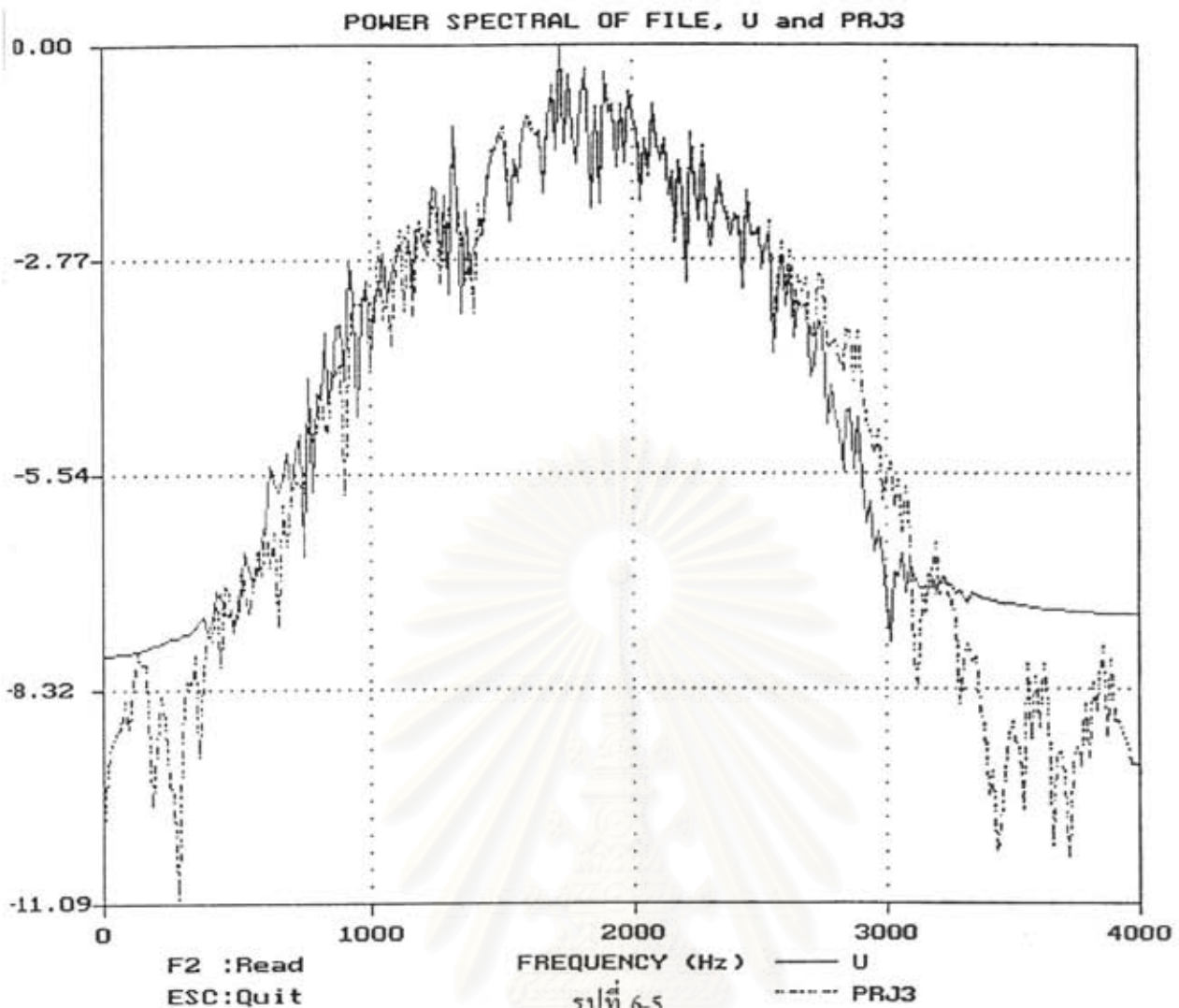
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 6-4

ผลตอบใน time domain ของสัญญาณเสียงพูดด้าน far-end (PRJ1) กับสัญญาณเสียงสะท้อน (PRJ2)

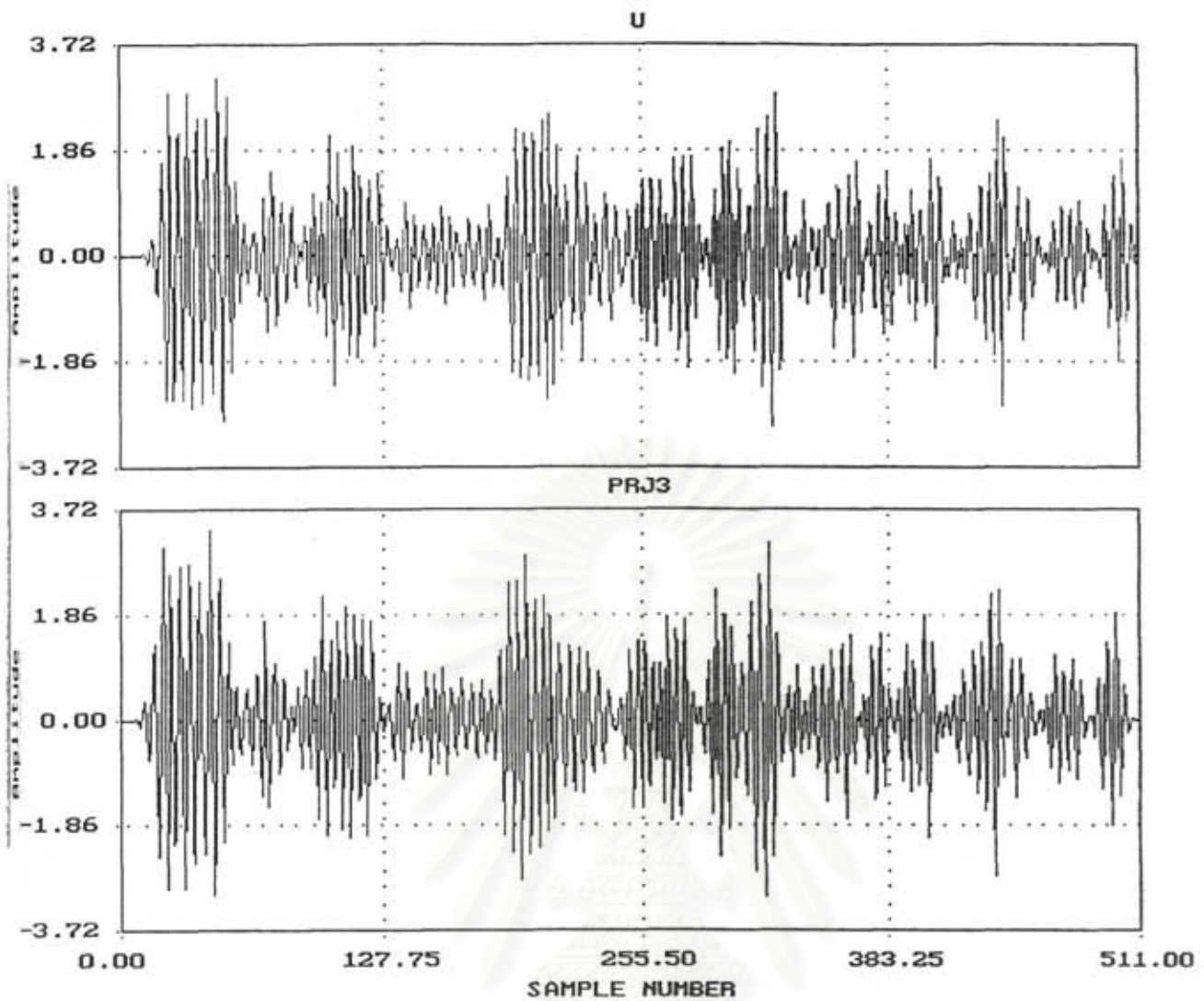
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



ผลตอบใน frequency domain เปรียบเทียบระหว่าง สัญญาณเสียงพูดด้าน near-end ชื่อไฟล์ PRJ3 กับ สัญญาณที่ออกจากเครื่องกำจัดเสียงสะท้อน ชื่อไฟล์ U ซึ่งมีพารามิเตอร์ดังนี้

- ความยาวของเวกเตอร์นำหนักร่วง = 4
- ค่าเริ่มต้นของเวกเตอร์ นำหนักร่วง = 0
- จำนวนรอบของการทำซ้ำ = 100
- Loop gain หรือ Step size หรือ μ = 4×10^{-5}

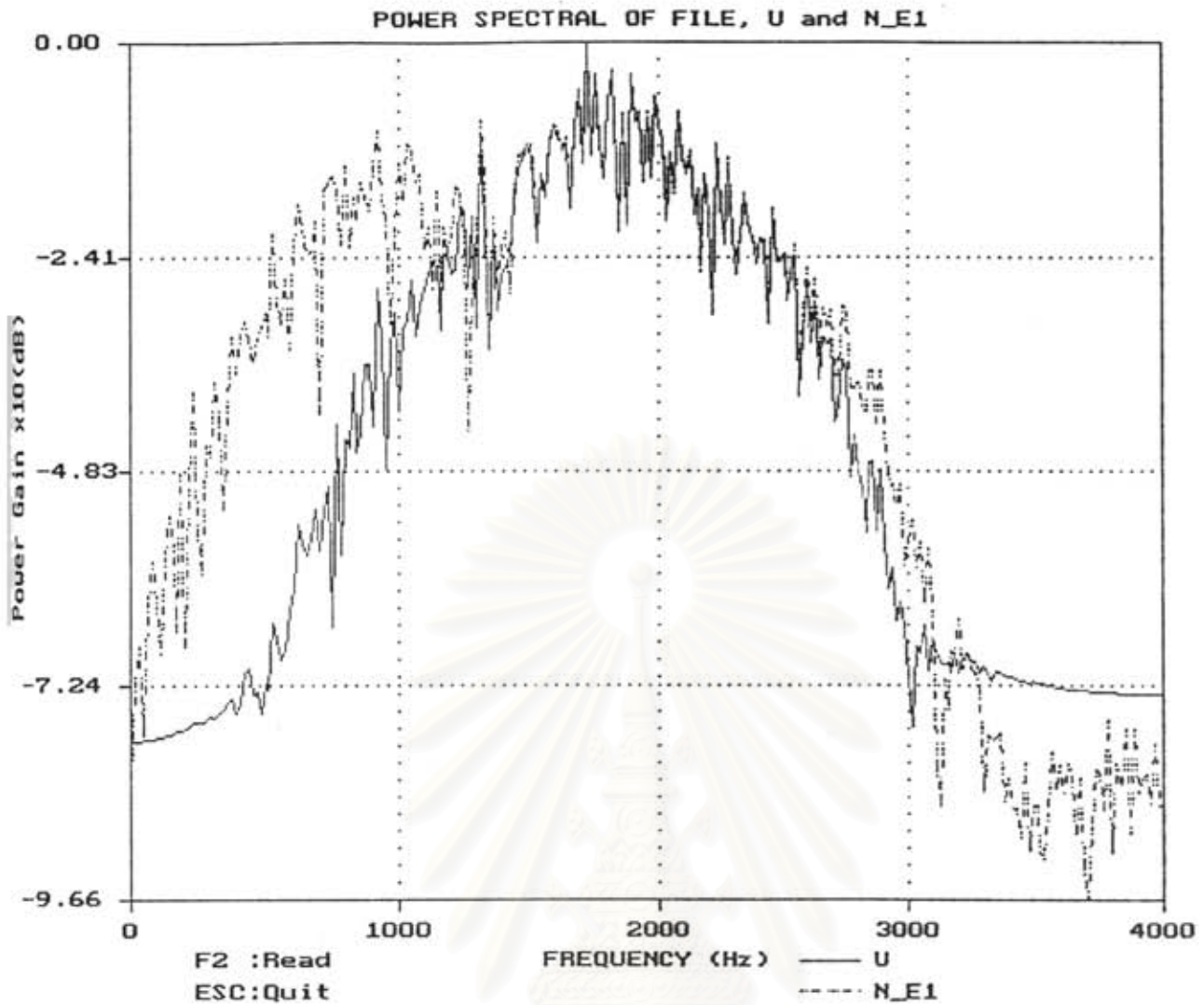
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 6-6

ผลตอบใน time domain เปรียบเทียบระหว่าง สัญญาณเสียงพูดด้าน near-end ชื่อไฟล์ PRJ3 กับ สัญญาณที่ออกจากเครื่องกำจัดเสียงสะท้อน U ซึ่งมีทารามิเตอร์ตั้งได้กล่าวไปแล้ว

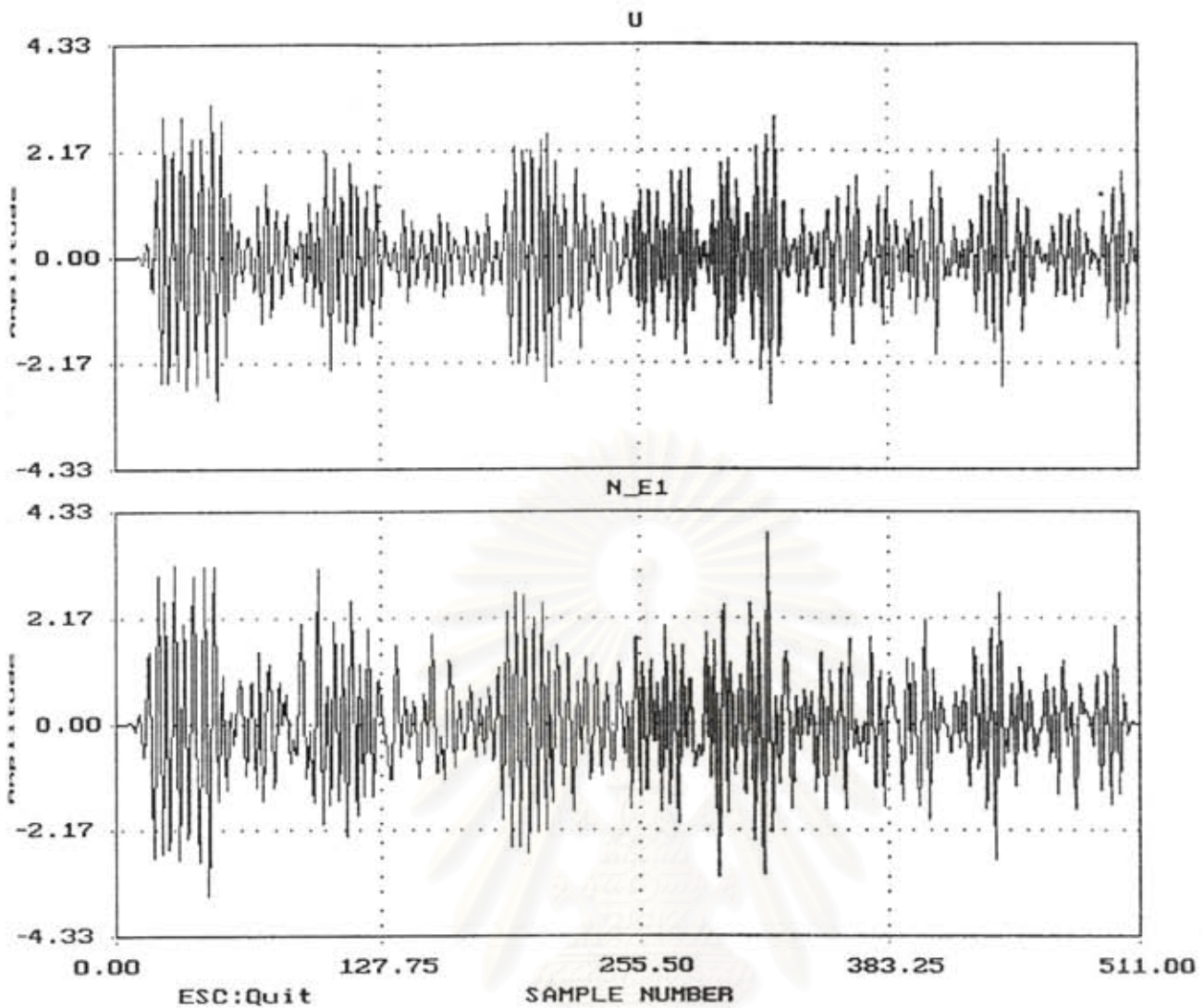
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 6-7

ผลตอบใน frequency domain เปรียบเทียบระหว่าง สัญญาณที่รวมเสียงพูดด้าน near-end (PRJ3) กับสัญญาณเสียงสะท้อน(PRJ2) ชื่อไฟล์ N_E1 กับ สัญญาณที่ออกจากเครื่องกำจัดเสียงสะท้อน ชื่อไฟล์ U ซึ่งมีพารามิเตอร์ดังกล่าวไปแล้ว

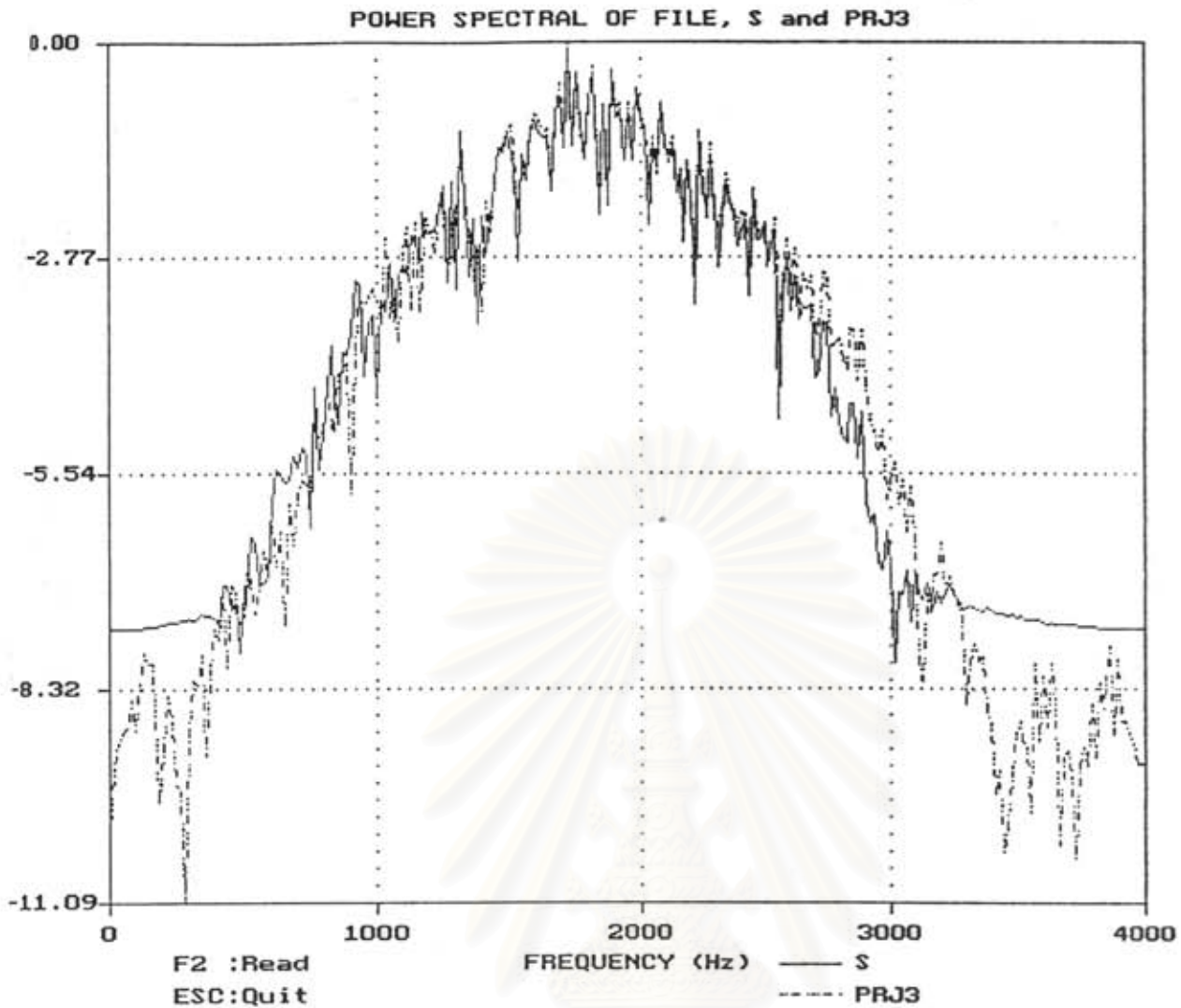
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 6-8

ผลตอบใน time domain เปรียบเทียบระหว่าง สัญญาณที่รวมเสียงพูดด้าน near-end (PRJ3) กับ สัญญาณเสียงสะท้อน(PRJ2) ชื่อไฟล์ N_E1 กับ สัญญาณที่ออกจากเครื่องกำจัดเสียงสะท้อน ชื่อไฟล์ U ซึ่งมีพารามิเตอร์ดังกล่าวไปแล้ว

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 6-9

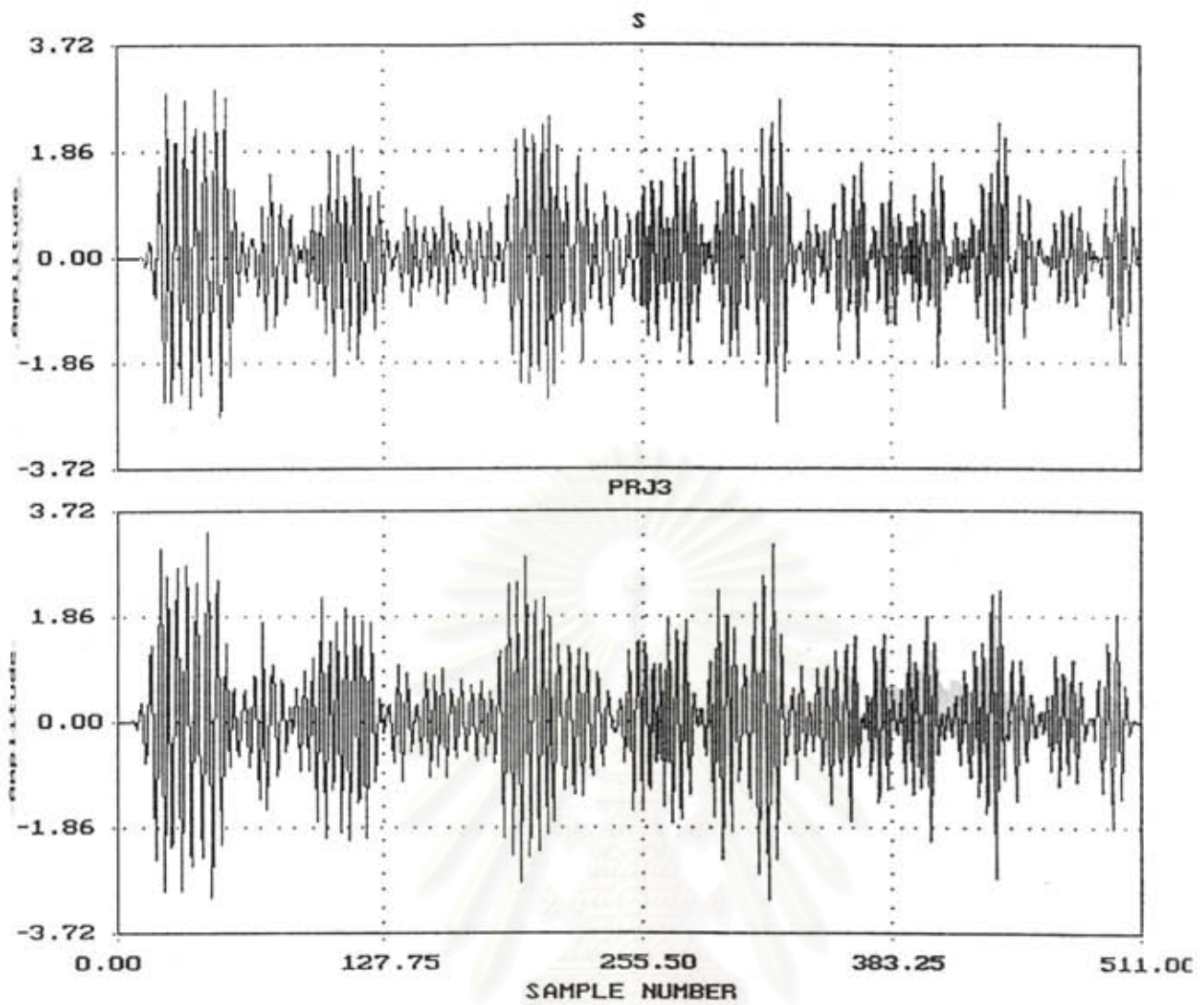
ผลตอบใน frequency domain เปรียบเทียบระหว่าง สัญญาณเสียงพูดด้าน near-end ชื่อ ไฟล์ PRJ3 กับ สัญญาณที่ออกจากเครื่องกำจัดเสียงสะท้อน S ซึ่งมีพารามิเตอร์ดังนี้

ความยาวของเวกเตอร์นำหนังก่อวง = 10

ค่าเริ่มต้นของเวกเตอร์นำหนังก่อวง = 0

จำนวนรอบของการทำซ้ำ = 100

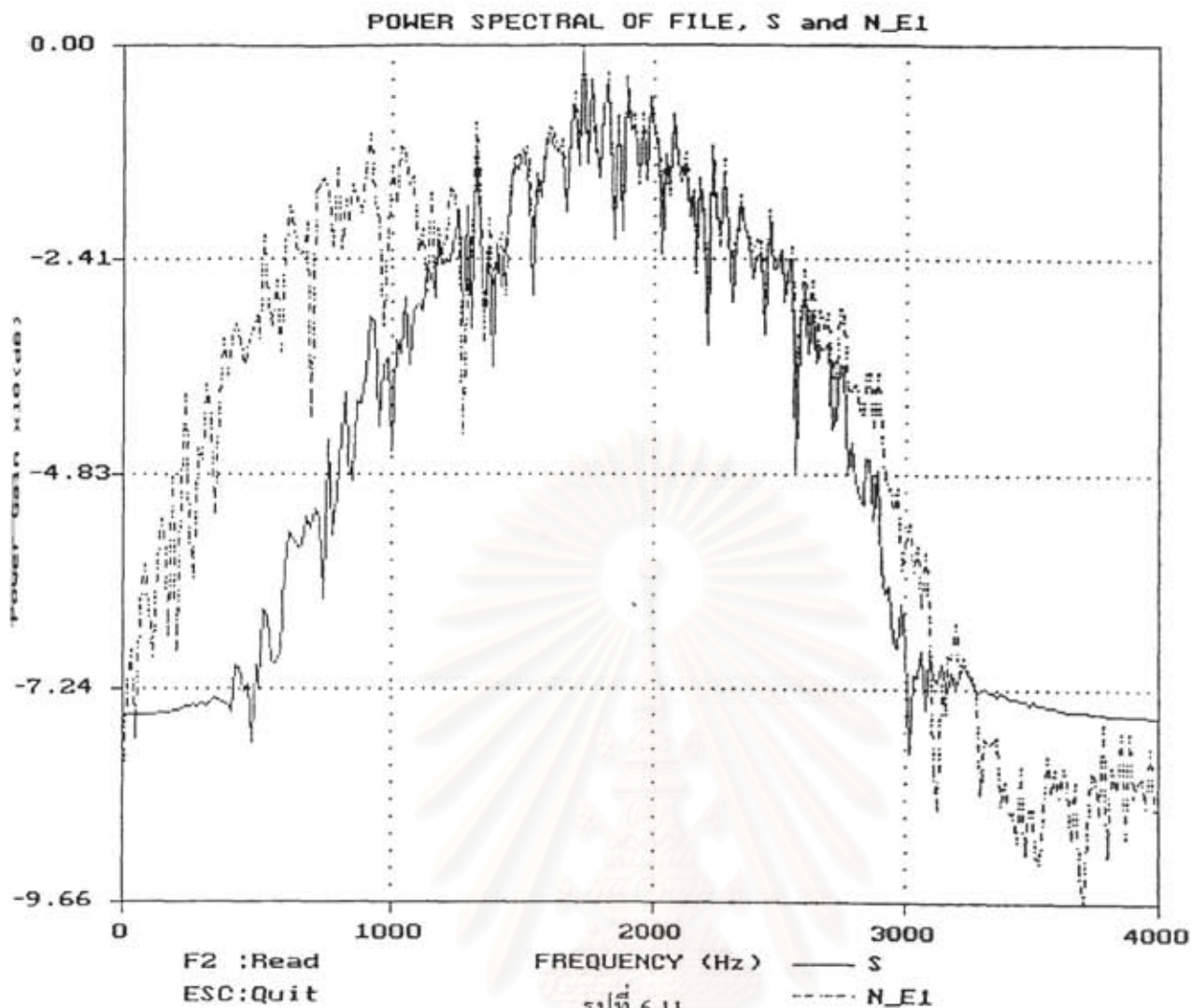
Loop gain หรือ Step size หรือ μ = 4×10^{-5}



รูปที่ 6-10

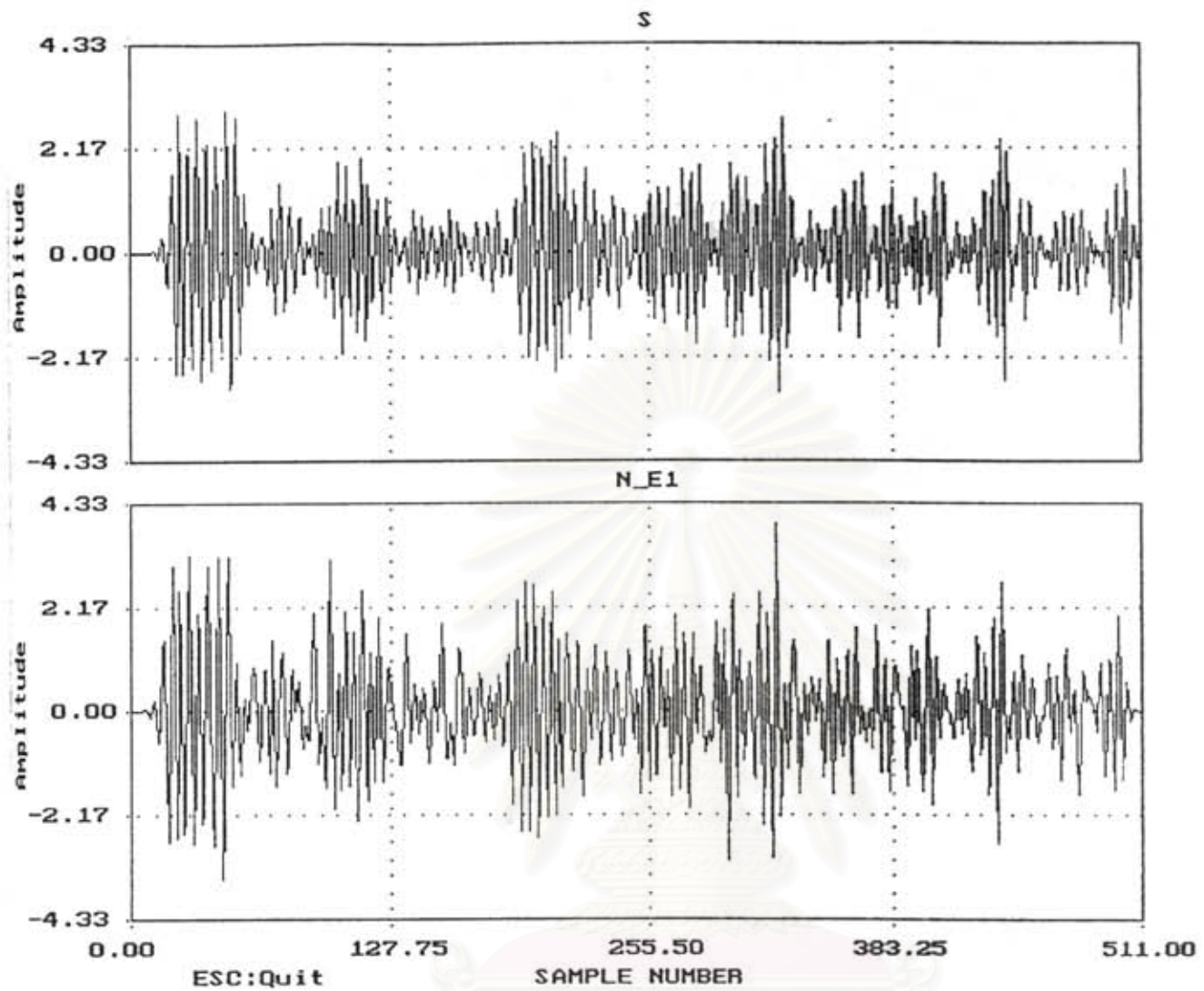
ผลตอบใน time domain เปรียบเทียบระหว่าง สัญญาณเสียงพูดด้าน near-end ชื่อไฟล์ PRJ3 กับ สัญญาณที่ออกจากเครื่องกำจัดเสียงสะท้อน S ซึ่งมีพารามิเตอร์ดังได้กล่าวไปแล้ว

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



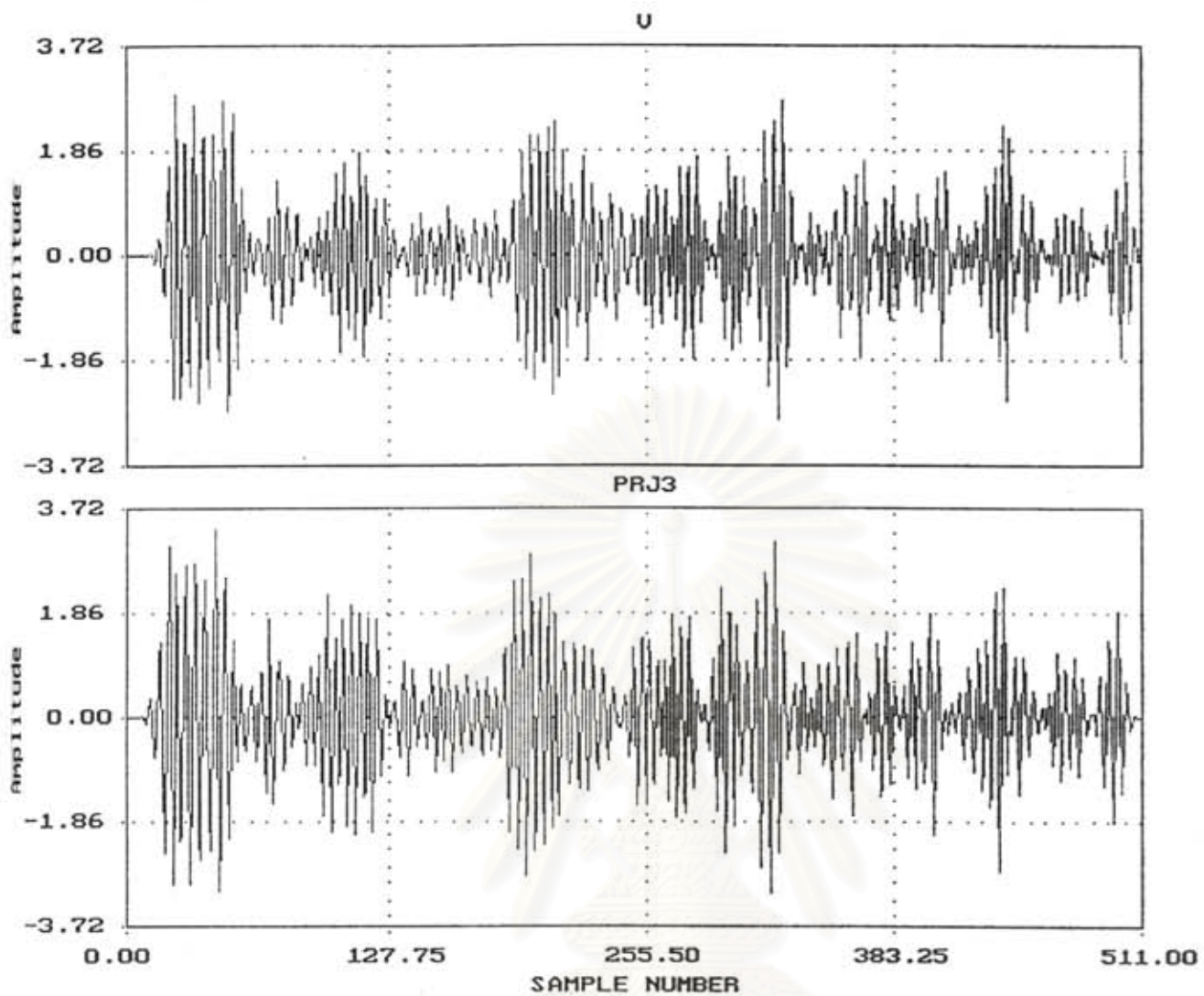
ผลตอบใน frequency domain เปรียบเทียบระหว่าง สัญญาณที่รวมเสียงพูดด้าน near-end (PRJ3) กับสัญญาณเสียงสะท้อน (PRJ2) ชื่อไฟล์ N_E1 กับ สัญญาณที่ออกจากเครื่องกำจัดเสียงสะท้อน ชื่อไฟล์ S ซึ่งมีพารามิเตอร์ดังกล่าวไปแล้ว

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



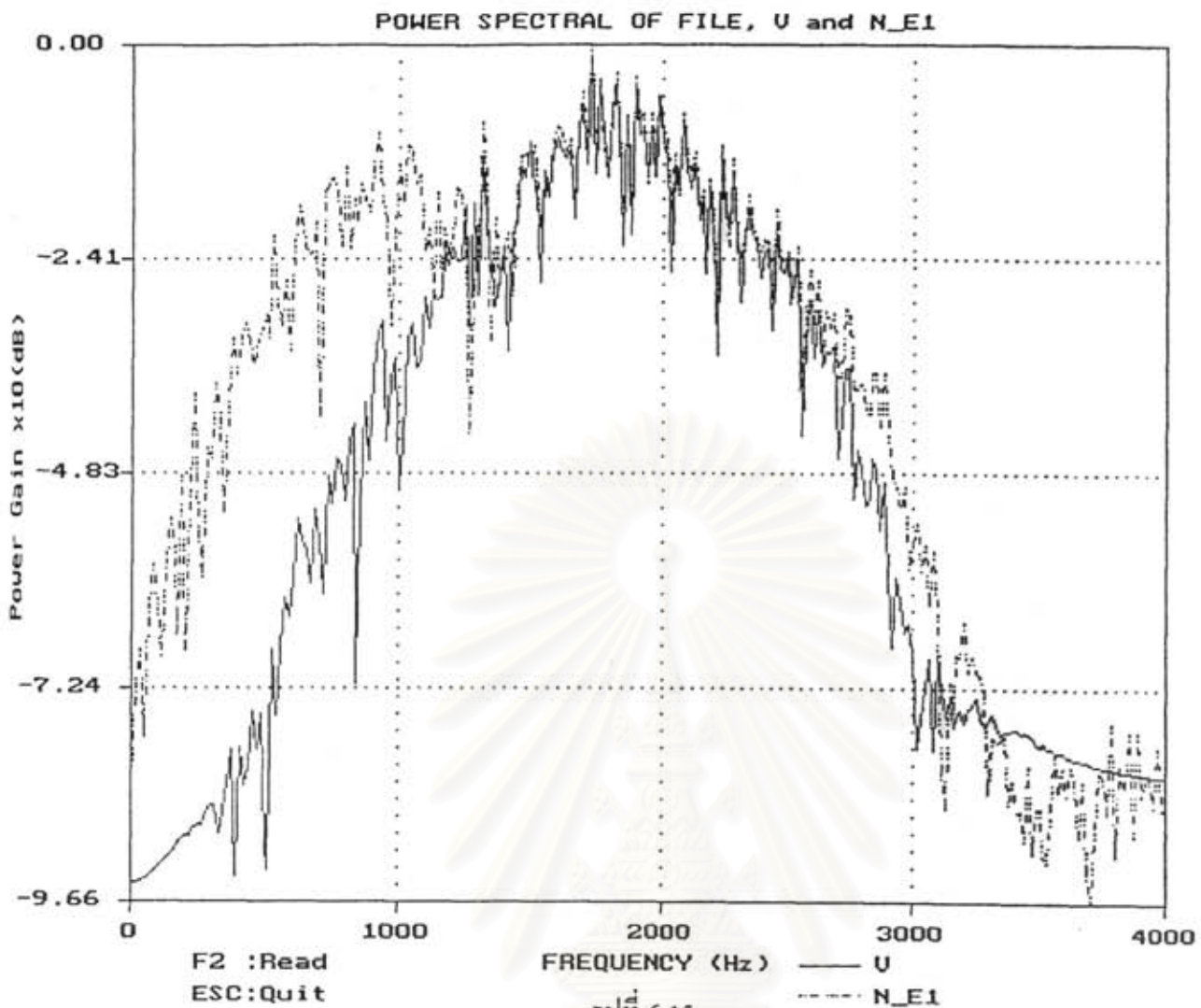
รูปที่ 6-12

ผลตอบใน time domain เปรียบเทียบระหว่าง สัญญาณที่รวมเสียงพูดด้าน near-end (PRJ3)กับ สัญญาณเสียงสะท้อน(PRJ2)ชื่อไฟล์ N_E1 กับ สัญญาณที่ออกจากเครื่องกำจัดเสียงสะท้อน ชื่อไฟล์ S ซึ่งมีพารามิเตอร์ดังกล่าวไปแล้ว



รูปที่ 6-14
 ผลตอบใน time domain เปรียบเทียบระหว่าง สัญญาณเสียงพูดด้าน near-end ชื่อไฟล์ PRJ3 กับ
 สัญญาณที่ออกจากเครื่องกำจัดเสียงสะท้อน V ซึ่งมีพารามิเตอร์ดังได้กล่าวไปแล้ว

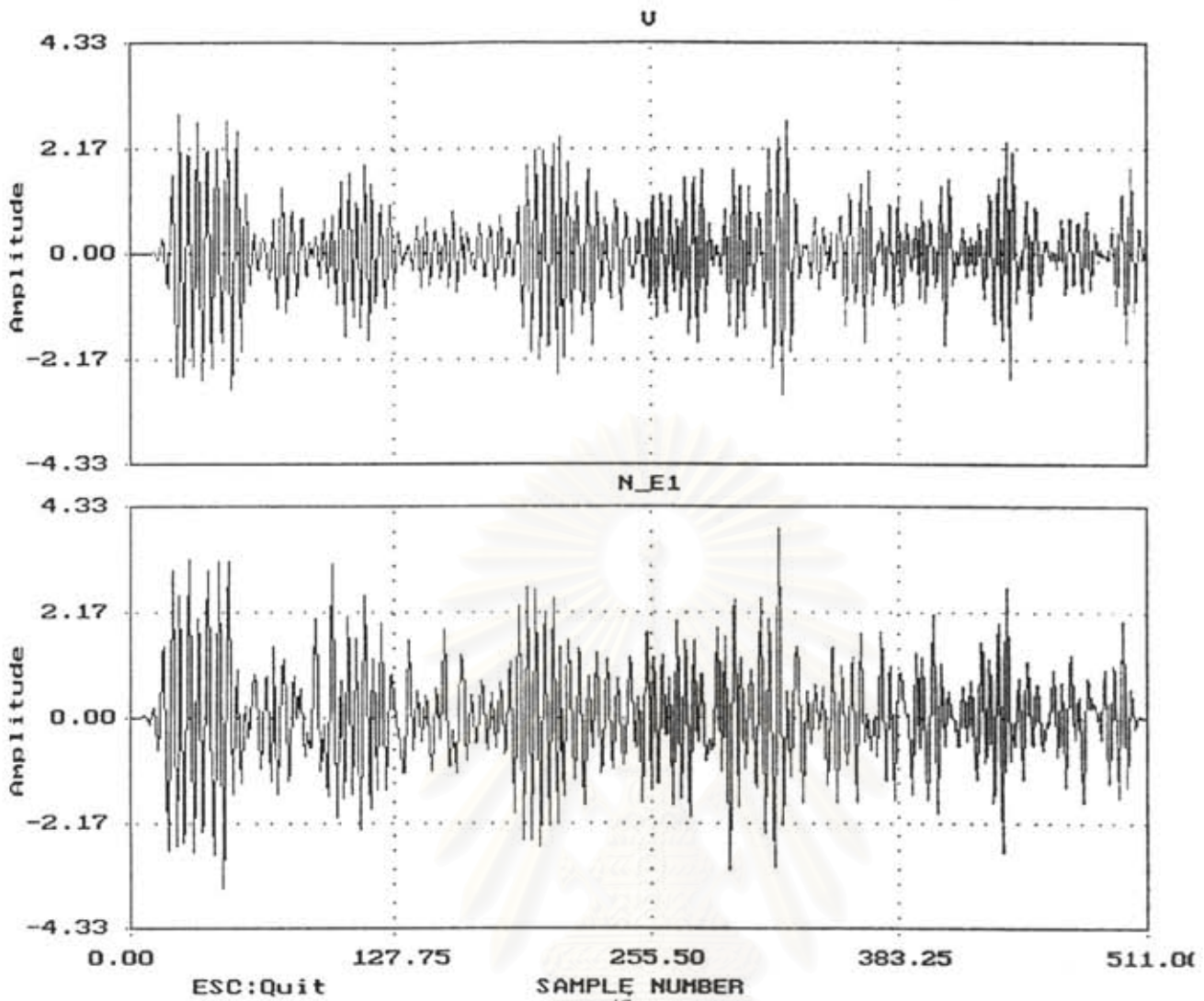
สถาบันวิทยบริการ
 จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 6-15

ผลตอบใน frequency domain เปรียบเทียบระหว่าง สัญญาณที่รวมเสียงพูดด้าน near-end (PRJ3) กับสัญญาณเสียงสะท้อน(PRJ2) ชื่อไฟล์ N_E1 กับ สัญญาณที่ออกจากเครื่องกำจัดเสียงสะท้อน ชื่อไฟล์ V ซึ่งมีพารามิเตอร์ดังกล่าวไปแล้ว

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 6-16
 ผลตอบใน time domain เปรียบเทียบระหว่าง สัญญาณที่รวมเสียงพูดด้าน near-end (PRJ3) กับ สัญญาณเสียงสะท้อน(PRJ2) ชื่อไฟล์ N_E1 กับ สัญญาณที่ออกจากเครื่องกำจัดเสียงสะท้อน ชื่อไฟล์ V ซึ่งมีพารามิเตอร์ดังกล่าวไปแล้ว

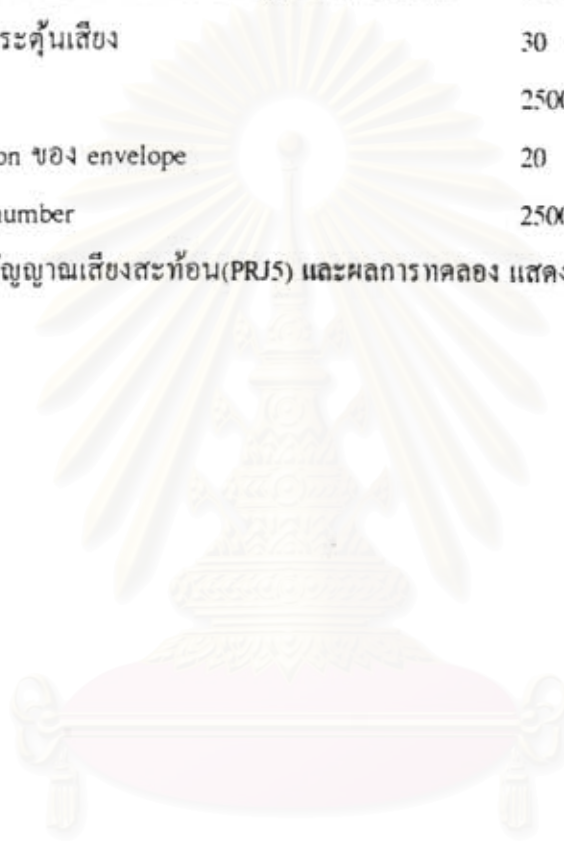
สถาบันวิทยบริการ
 จุฬาลงกรณ์มหาวิทยาลัย

กรณีที่ 2.เสียงสะท้อนมีความถี่สูง

กำเนิดสัญญาณเสียงพูดจาก far-end และ เสียงสะท้อนตามรายละเอียดดังนี้

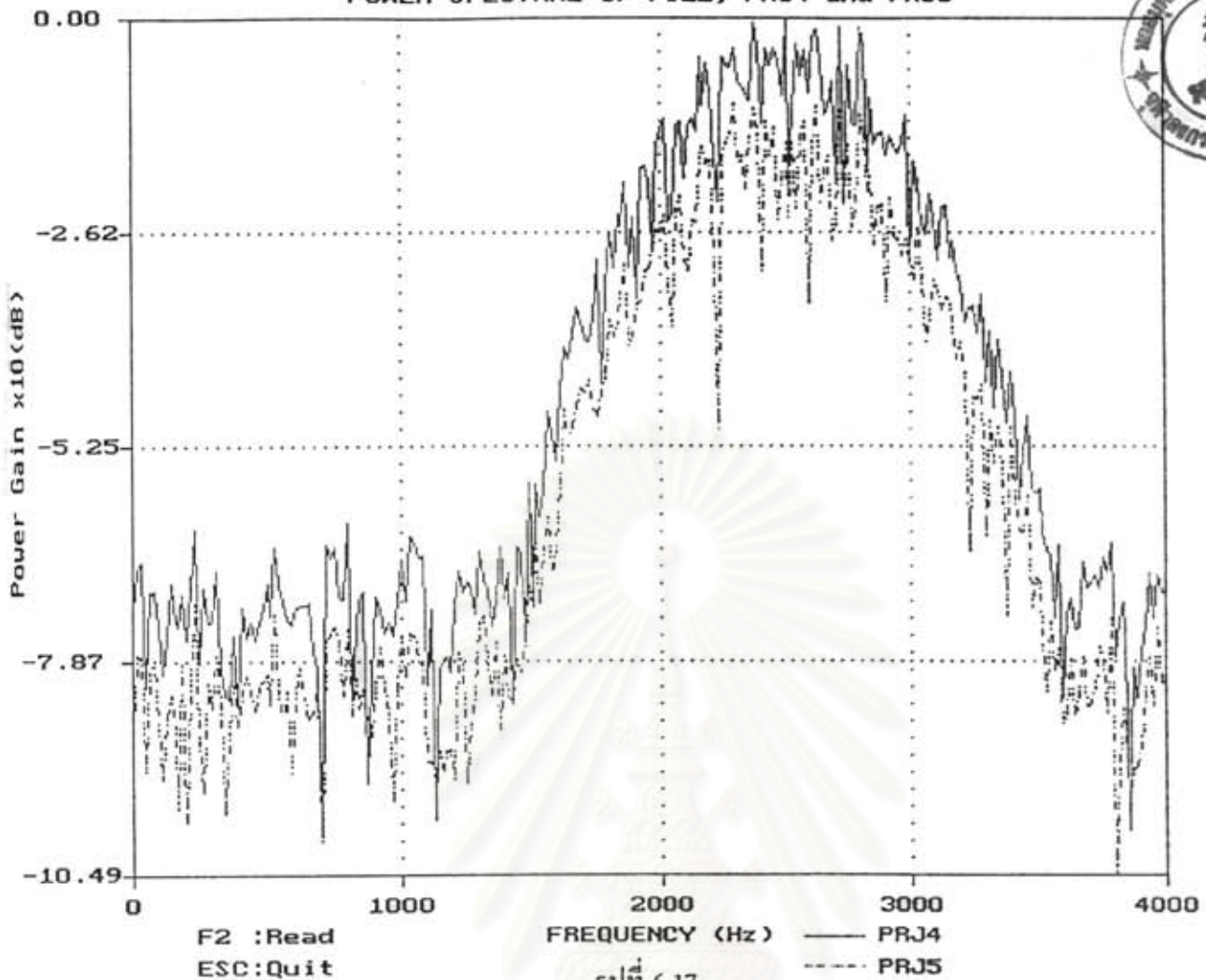
ชื่อเต็มข้อมูลของสัญญาณจาก far-end	PRJ4
ชื่อเต็มข้อมูลของสัญญาณเสียงสะท้อน	PRJ5
Correlation coefficient ระหว่างสัญญาณ- จาก far-end กับ สัญญาณเสียงสะท้อน	0.85
การลดทอนของเสียงสะท้อนเมื่อเทียบกับสัญญาณจาก far-end	0.1 เท่า
ความยาวของพัลส์กระตุ้นเสียง	30 ตัวอย่าง
ความถี่กลาง	2500 Hz
ค่า standard deviation ของ envelope	20
Seed ของ random number	2500

สัญญาณ far-end (PRJ4) , สัญญาณเสียงสะท้อน(PRJ5) และผลการทดลอง แสดงดังต่อไปนี้



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

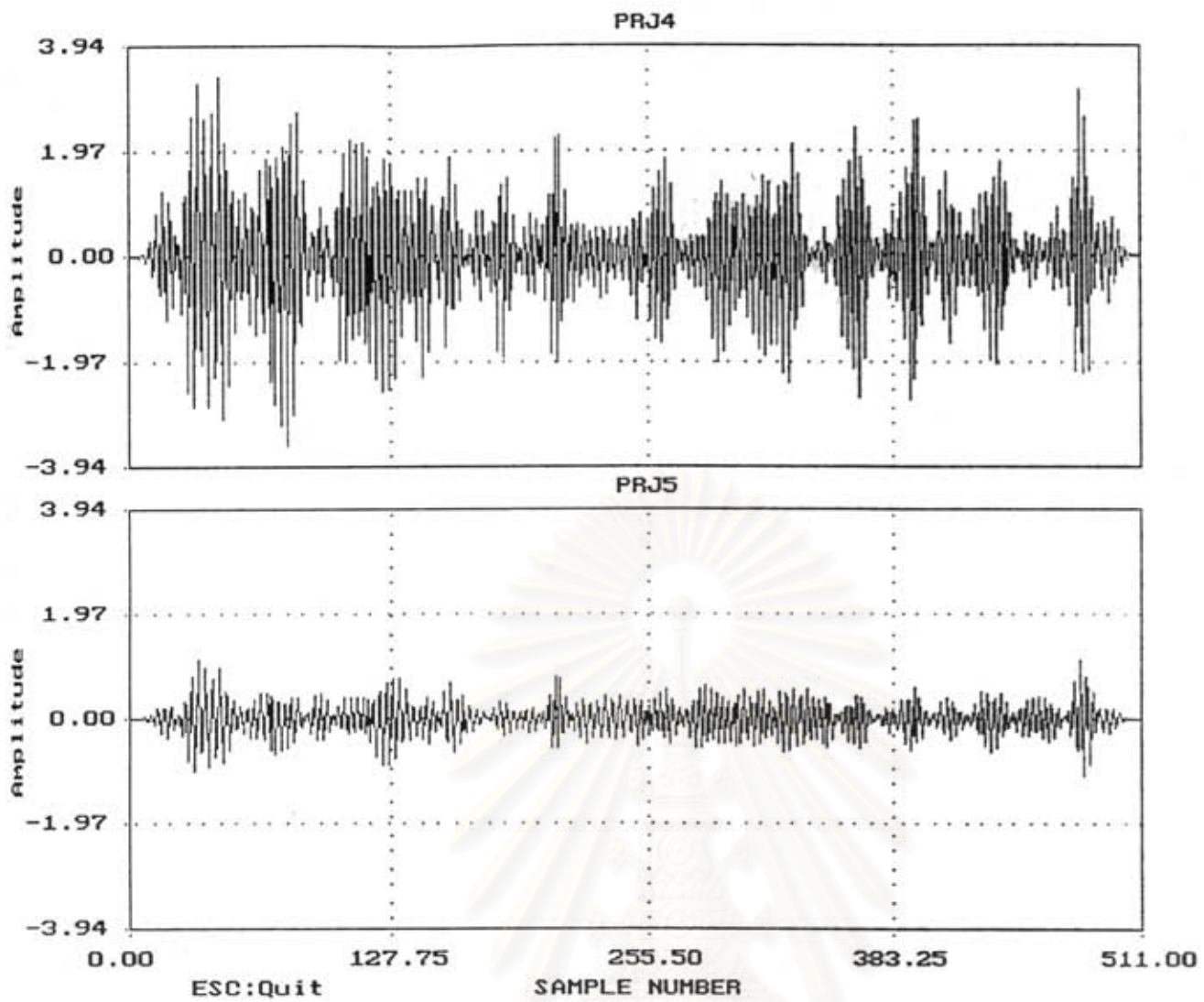
POWER SPECTRAL OF FILE, PRJ4 and PRJ5



รูปที่ 6-17

ผลตอบใน frequency domain ของสัญญาณเสียงพูดด้าน far-end (PRJ4) กับสัญญาณเสียงสะท้อน (PRJ5)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

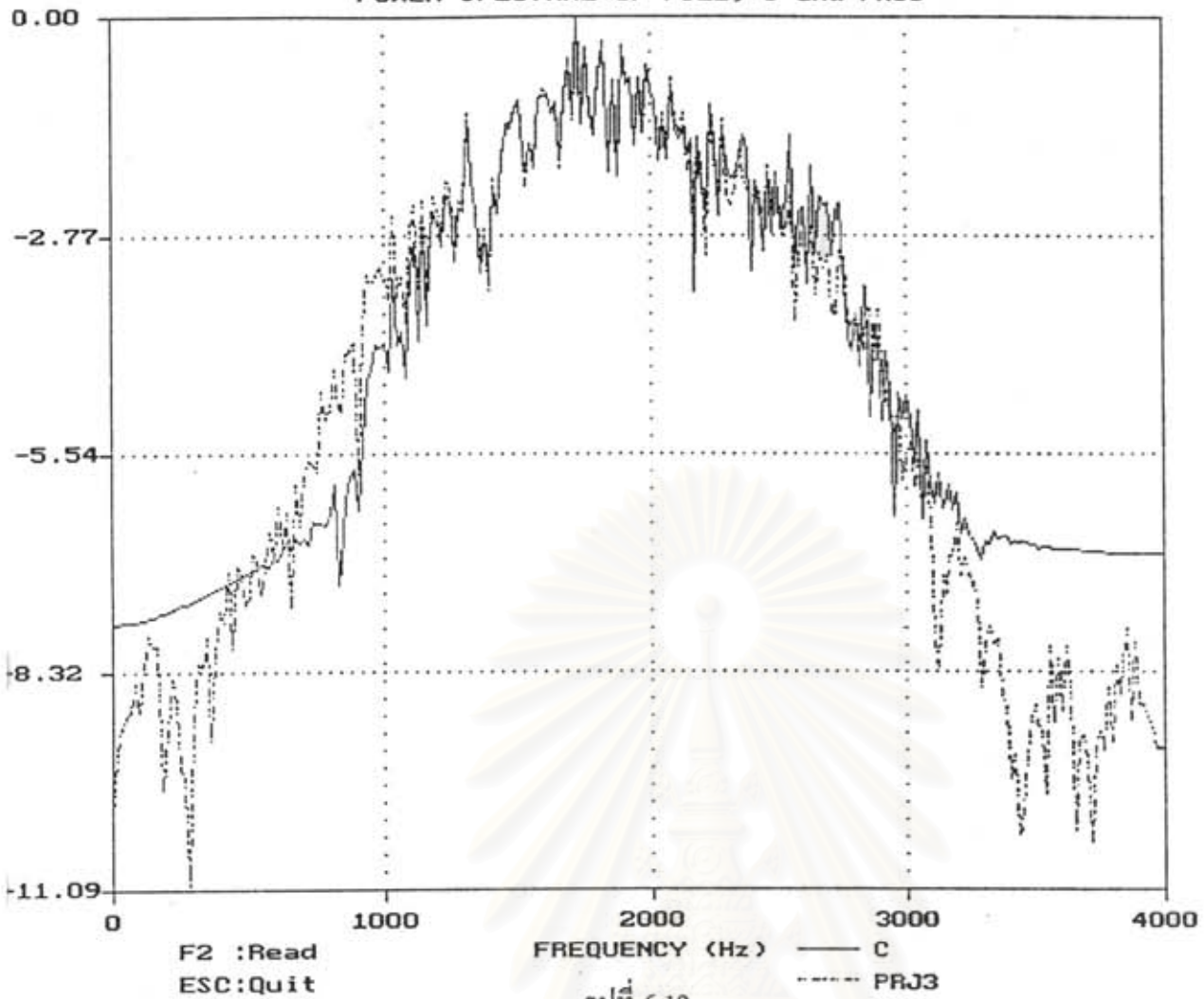


รูปที่ 6-18

ผลตอบใน time domain ของสัญญาณเสียงพูดด้าน far-end (PRJ4) กับสัญญาณเสียงสะท้อน (PRJ5)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

POWER SPECTRAL OF FILE, C and PRJ3

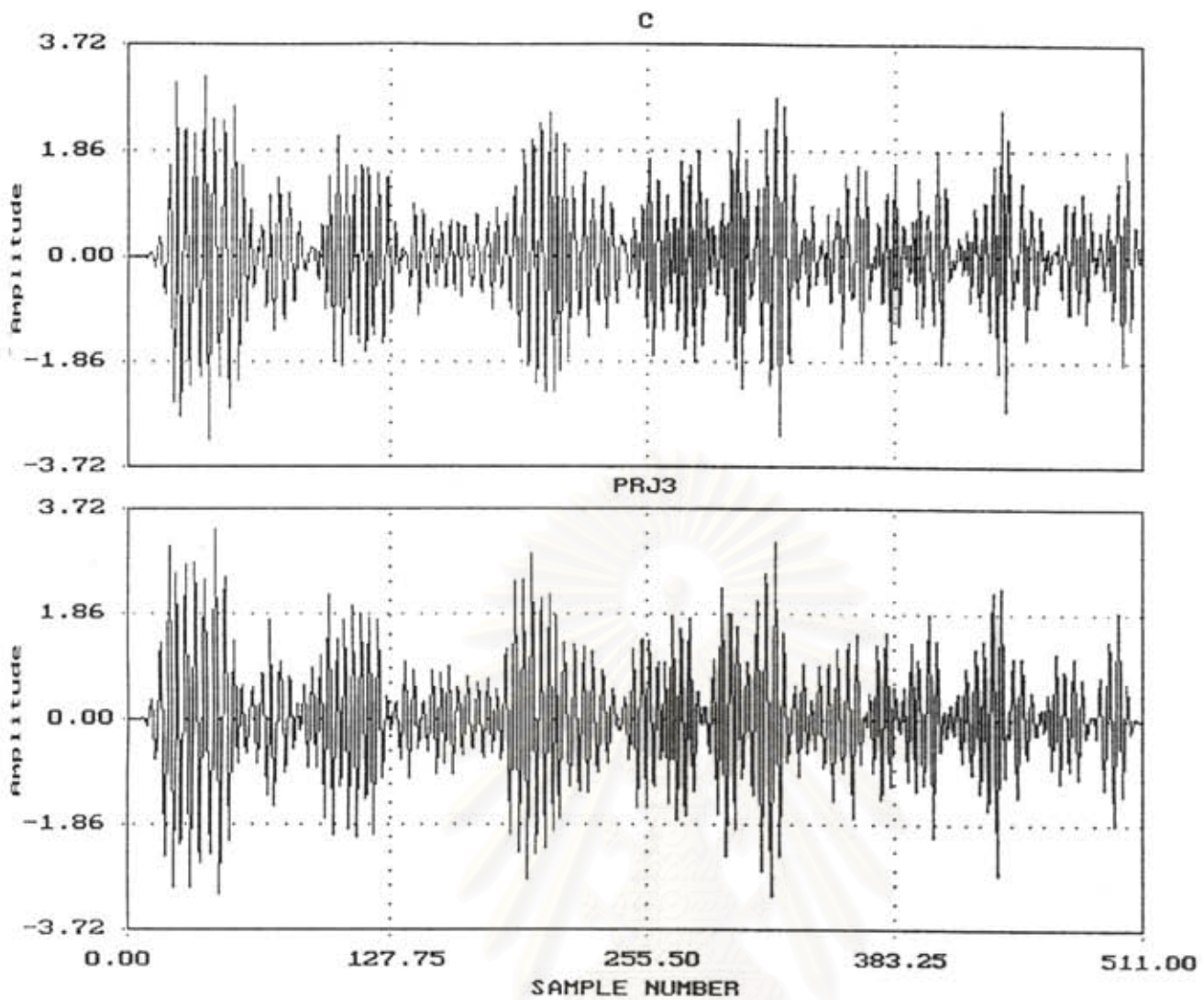


รูปที่ 6-19

ผลตอบใน frequency domain เปรียบเทียบระหว่าง สัญญาณเสียงพูดด้าน near-end ชื่อไฟล์ PRJ3 กับ สัญญาณที่ออกจากเครื่องกำจัดเสียงสะท้อน C ซึ่งมีพารามิเตอร์ดังนี้

- ความยาวของเวกเตอร์นำหนักร่วง = 4
- ค่าเริ่มต้นของเวกเตอร์นำหนักร่วง = 0
- จำนวนรอบของการทำซ้ำ = 100
- Loop gain หรือ Step size หรือ μ = 4×10^{-5}

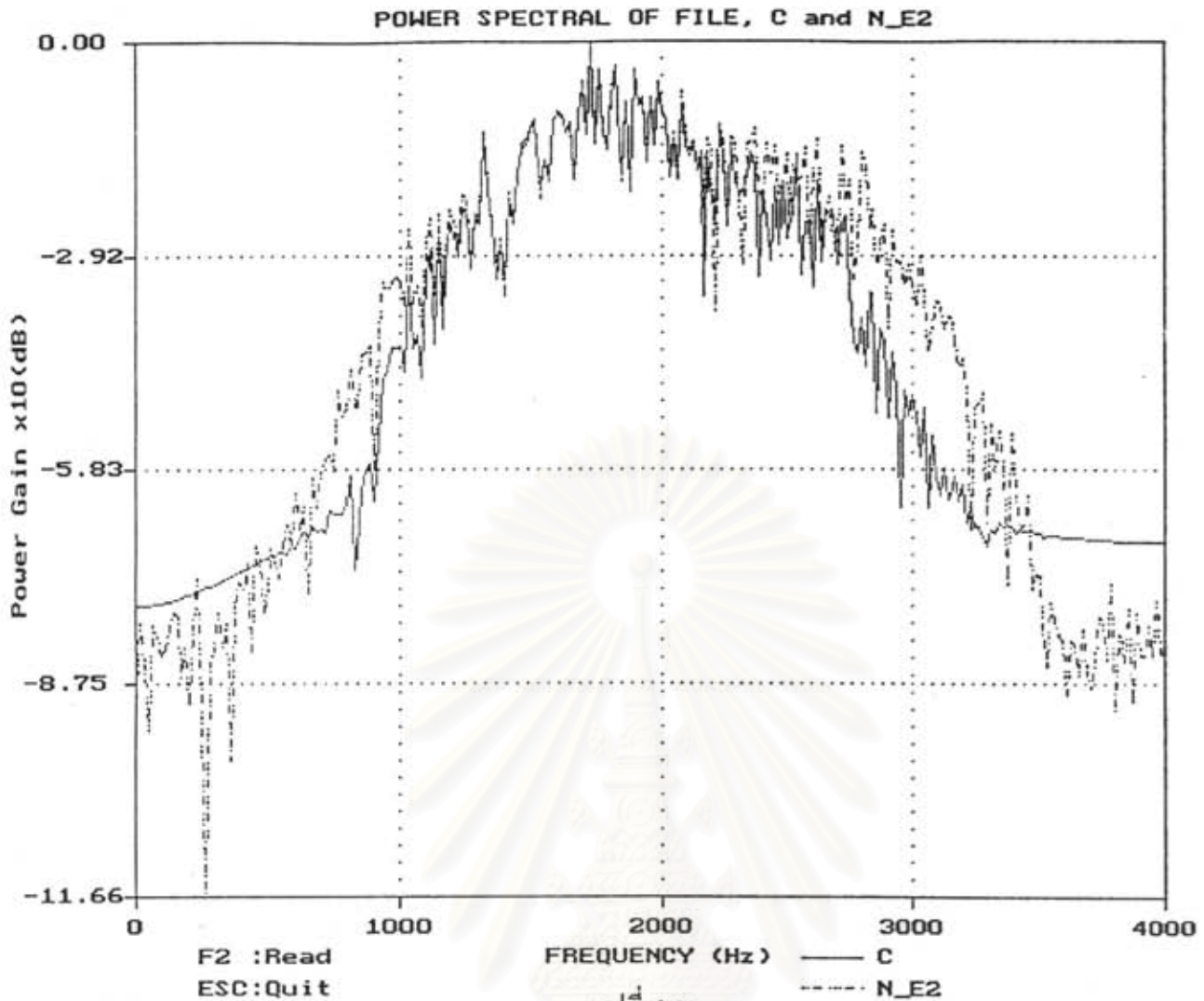
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 6-20

ผลตอบใน time domain เปรียบเทียบระหว่าง สัญญาณเสียงพูดด้าน near-end ชื่อไฟล์ PRJ3 กับ สัญญาณที่ออกจากเครื่องกำจัดเสียงสะท้อน C ซึ่งมีพารามิเตอร์ตั้งได้กล่าวไปแล้ว

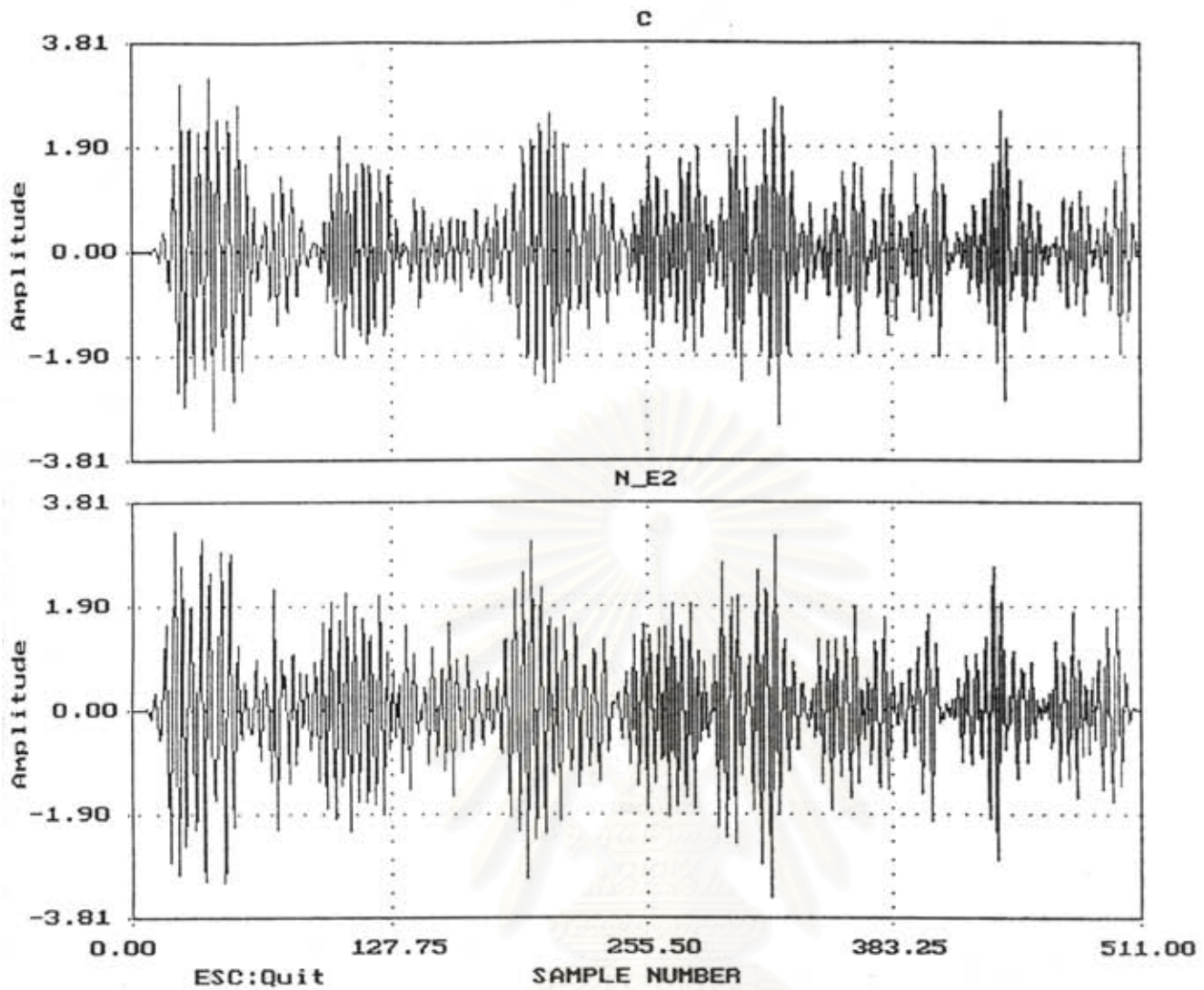
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 6-21

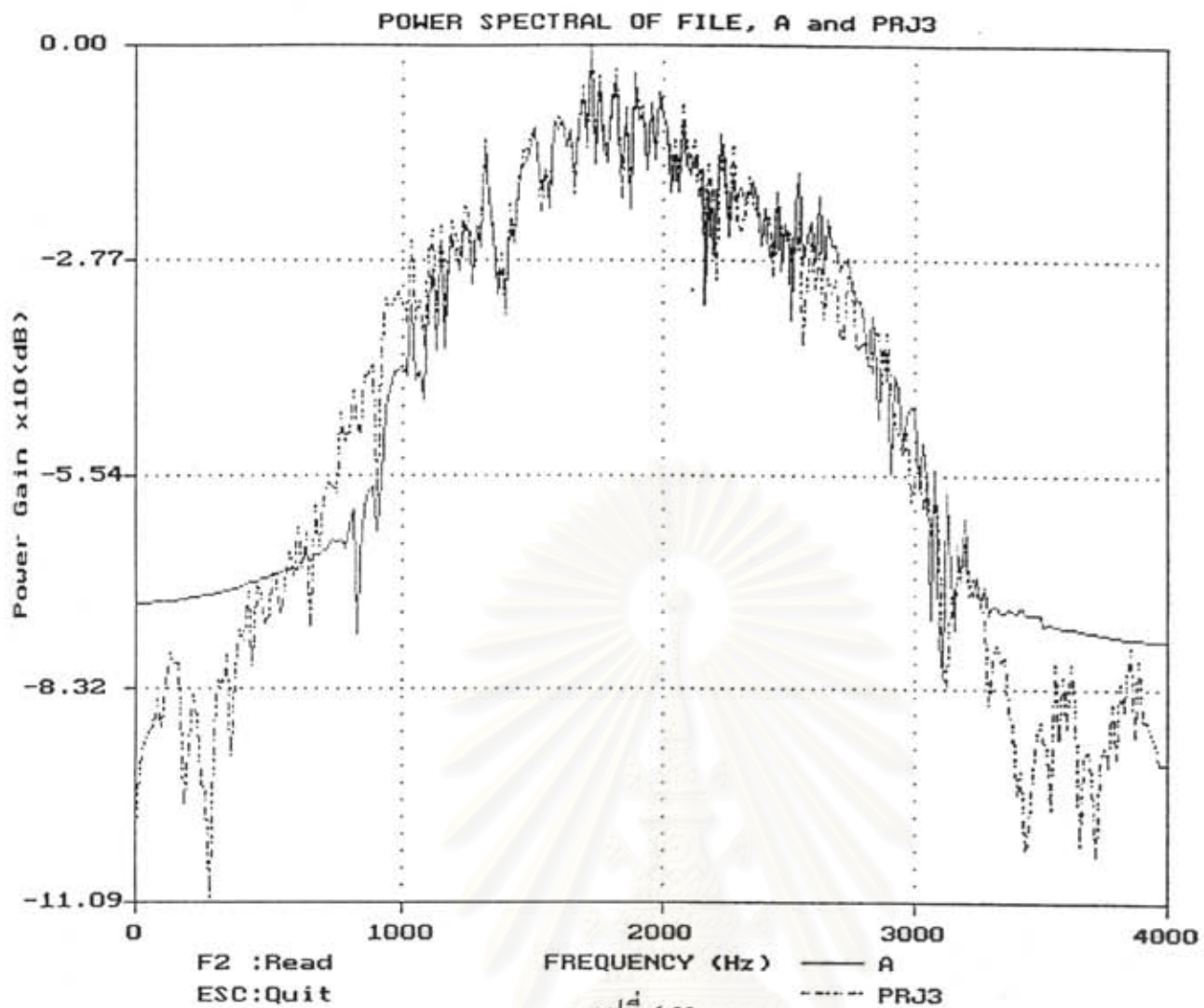
ผลตอบใน frequency domain เปรียบเทียบระหว่าง สัญญาณที่รวมเสียงพูดด้าน near-end (PRJ3) กับสัญญาณเสียงสะท้อน(PRJ5) ชื่อไฟล์ N_E2 กับ สัญญาณที่ออกจากเครื่องกำจัดเสียงสะท้อน ชื่อไฟล์ C ซึ่งมีพารามิเตอร์ดังกล่าวไปแล้ว

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 6-22
 ผลตอบใน time domain เปรียบเทียบระหว่าง สัญญาณที่รวมเสียงพูดด้าน near-end (PRJ3) กับ สัญญาณเสียงสะท้อน(PRJ5) ชื่อไฟล์ N_E2 กับ สัญญาณที่ออกจากเครื่องกำจัดเสียงสะท้อน ชื่อไฟล์ C ซึ่งมีพารามิเตอร์ดังกล่าวไปแล้ว

สถาบันวิทยบริการ
 จุฬาลงกรณ์มหาวิทยาลัย

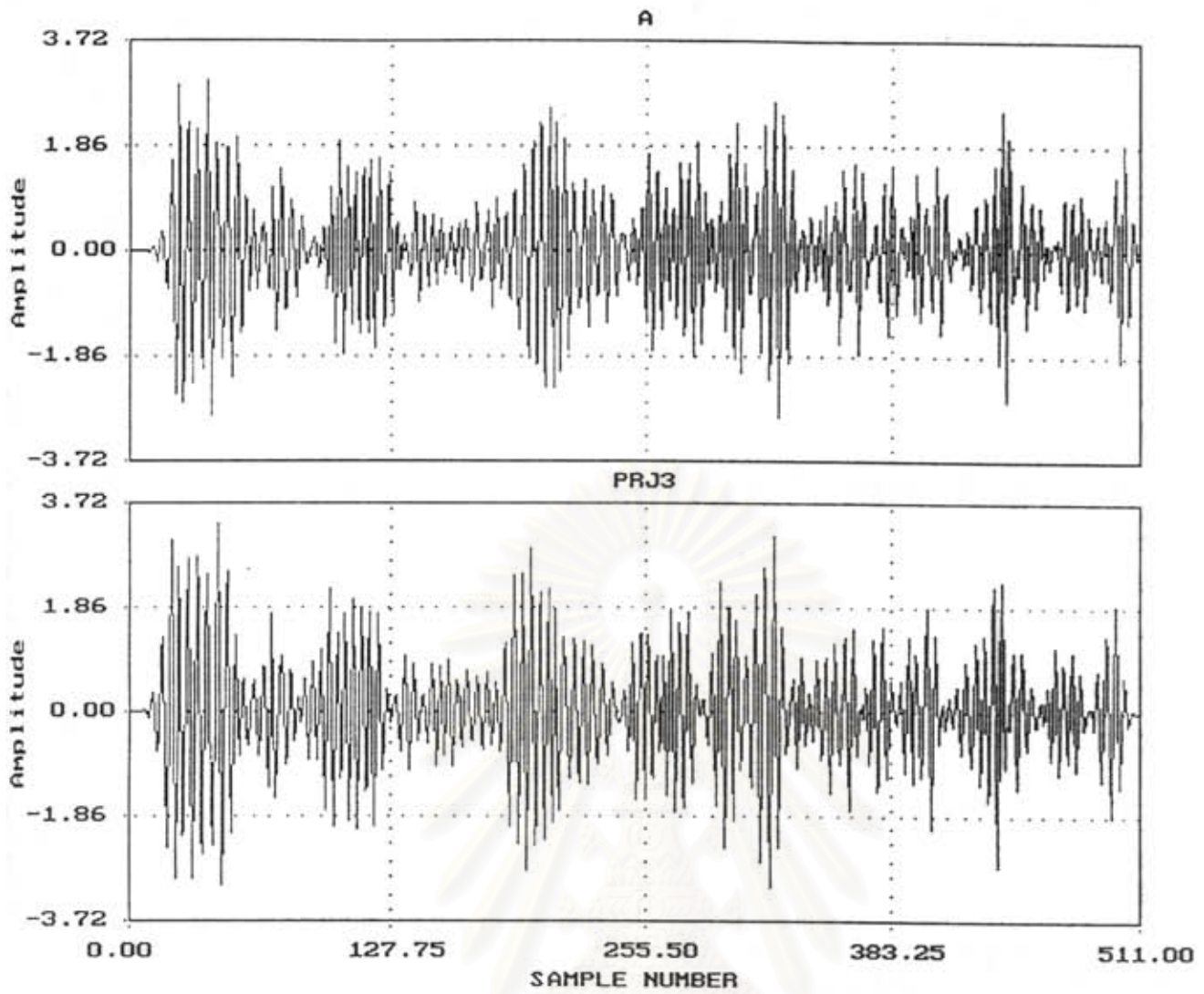


รูปที่ 6-23

ผลตอบใน frequency domain เปรียบเทียบระหว่าง สัญญาณเสียงพูดด้าน near-end ชื่อไฟล์ PRJ3 กับ สัญญาณที่ออกจากเครื่องกำจัดเสียงสะท้อน ชื่อไฟล์ A ซึ่งมีพารามิเตอร์ดังนี้

ความยาวของเวกเตอร์นำหนักร่วง	= 10
ค่าเริ่มต้นของเวกเตอร์นำหนักร่วง	= 0
จำนวนรอบของการทำซ้ำ	= 100
Loop gain หรือ Step size หรือ μ	= 4×10^{-5}

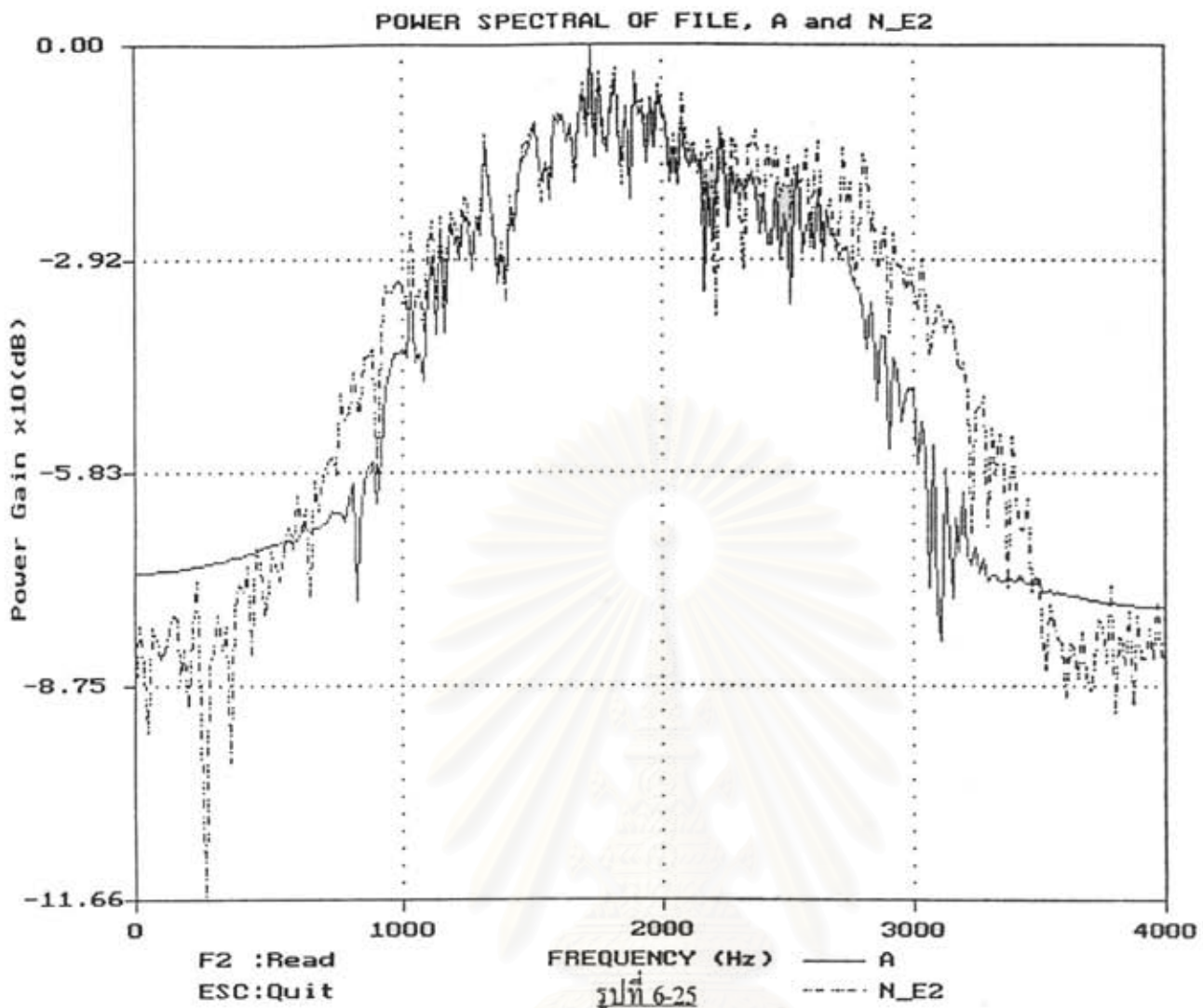
สถาบันวิจัยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 6-24

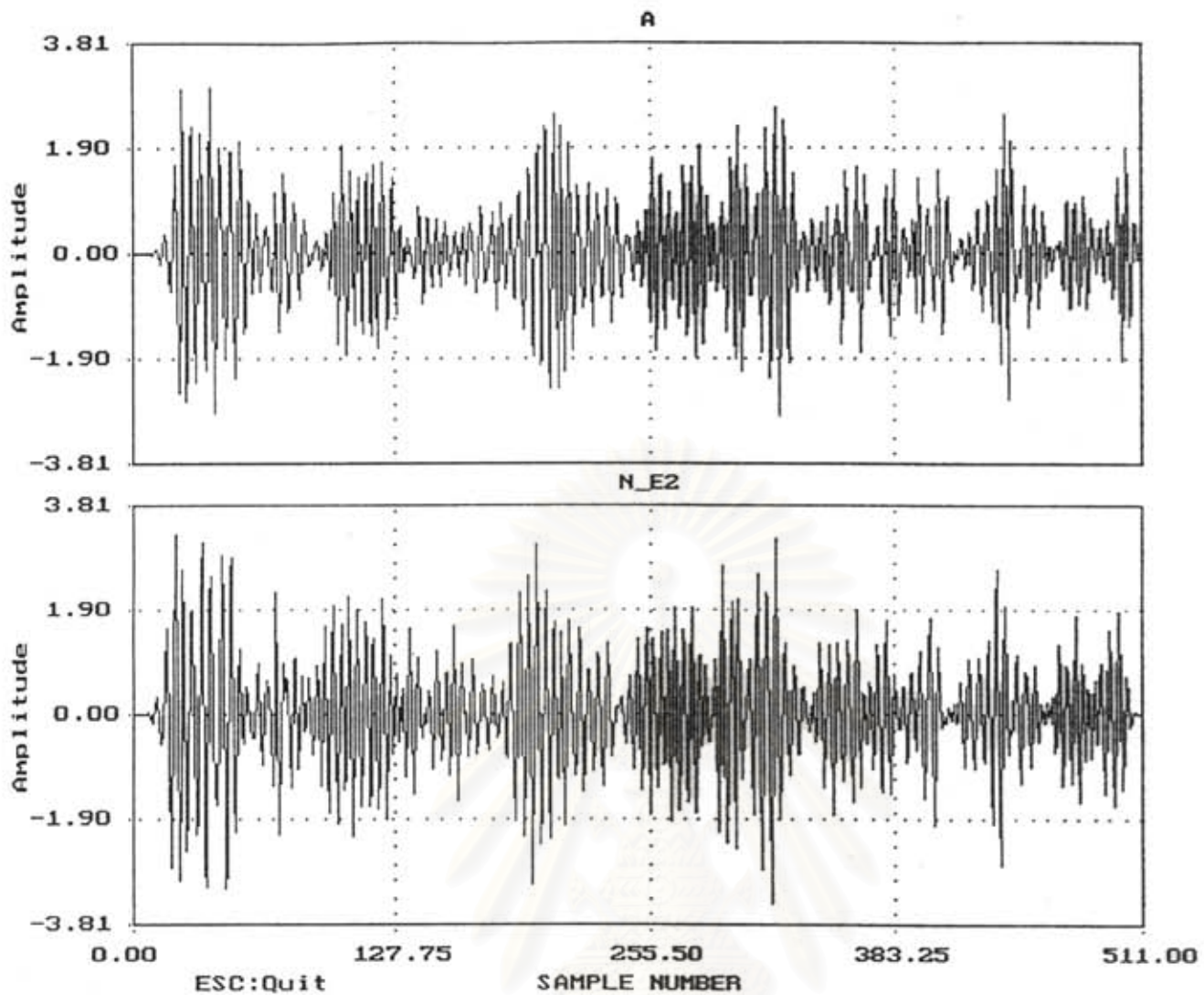
ผลตอบใน time domain เปรียบเทียบระหว่าง สัญญาณเสียงพูดด้าน near-end ชื่อไฟล์ PRJ3 กับ สัญญาณที่ออกจากเครื่องกำจัดเสียงสะท้อน A ซึ่งมีพารามิเตอร์ตั้งได้กล่าวไปแล้ว

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



ผลตอบใน frequency domain เปรียบเทียบระหว่าง สัญญาณที่รวมเสียงพูดด้าน near-end (PRJ3) กับสัญญาณเสียงสะท้อน (PRJ5) ชื่อไฟล์ N_E2 กับ สัญญาณที่ออกจากเครื่องกำจัดเสียงสะท้อน ชื่อไฟล์ A ซึ่งมีพารามิเตอร์ดังกล่าวไปแล้ว

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

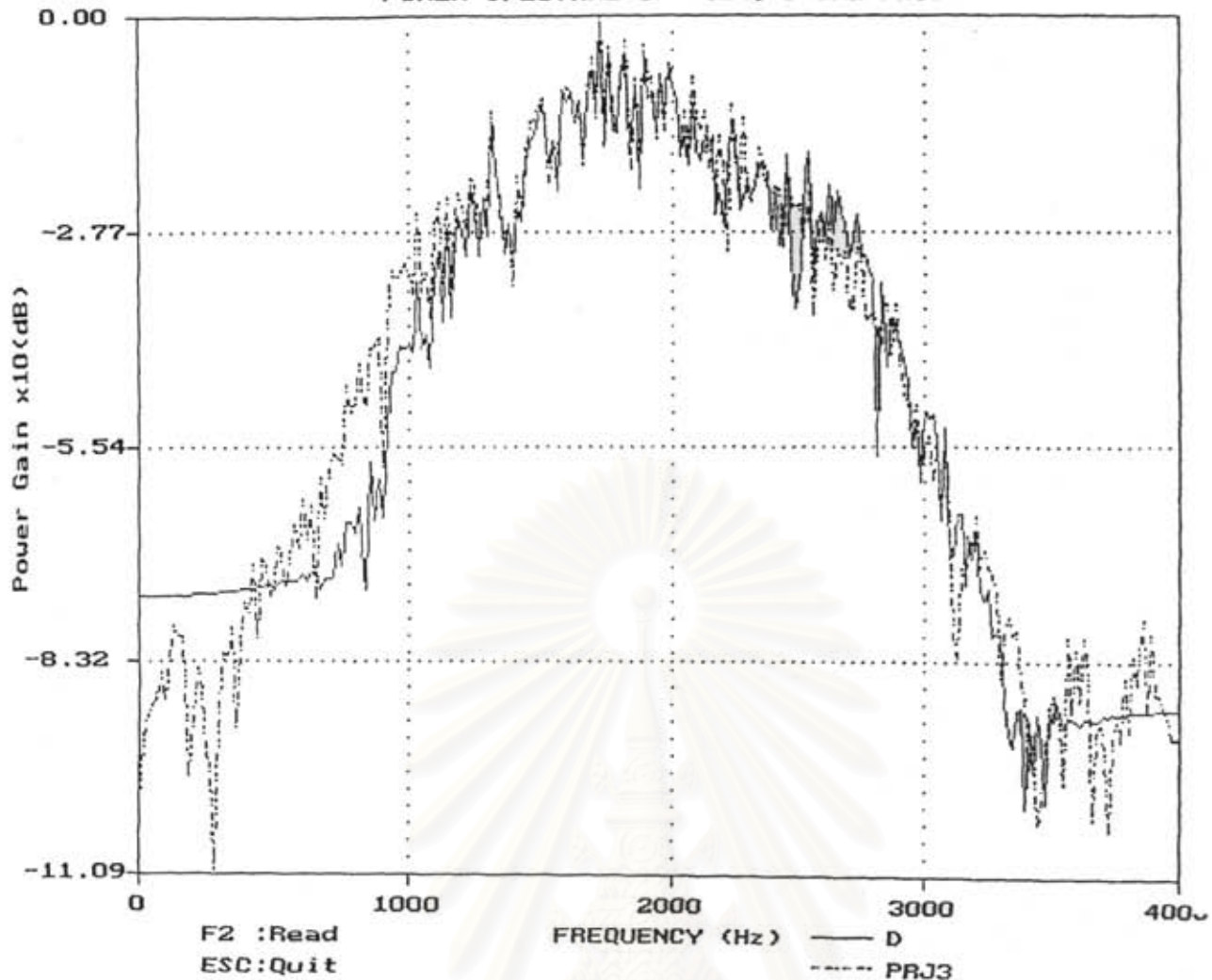


รูปที่ 6-26

ผลตอบใน time domain เปรียบเทียบระหว่าง สัญญาณที่รวมเสียงพูดด้าน near-end (PRJ3) กับ สัญญาณเสียงสะท้อน(PRJ5)ชื่อไฟล์ N_E2 กับ สัญญาณที่ออกจากเครื่องกำจัดเสียงสะท้อน ชื่อไฟล์ A ซึ่งมีพารามิเตอร์ดังกล่าวไปแล้ว

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

POWER SPECTRAL OF FILE, D and PRJ3



รูปที่ 6-27

ผลตอบใน frequency domain เปรียบเทียบระหว่าง สัญญาณเสียงพูดด้าน near-end ชื่อไฟล์ PRJ3 กับ สัญญาณที่ออกจากเครื่องกำจัดเสียงสะท้อน D ซึ่งมีพารามิเตอร์ดังนี้

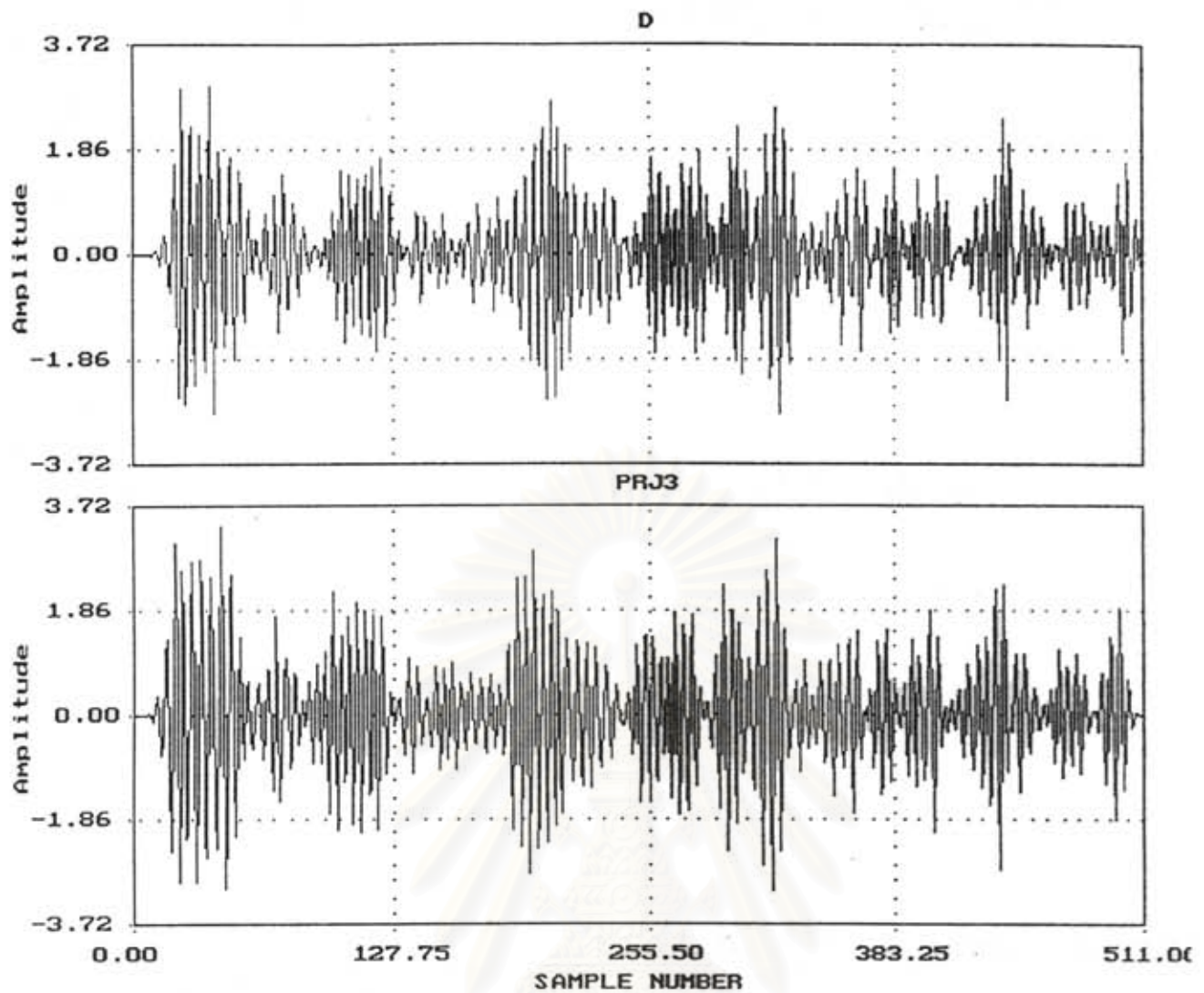
ความยาวของเวกเตอร์นำหนักร่วง = 16

ค่าเริ่มต้นของเวกเตอร์นำหนักร่วง = 0

จำนวนรอบของการทำซ้ำ = 100

Loop gain หรือ Step size หรือ μ = 4×10^{-5}

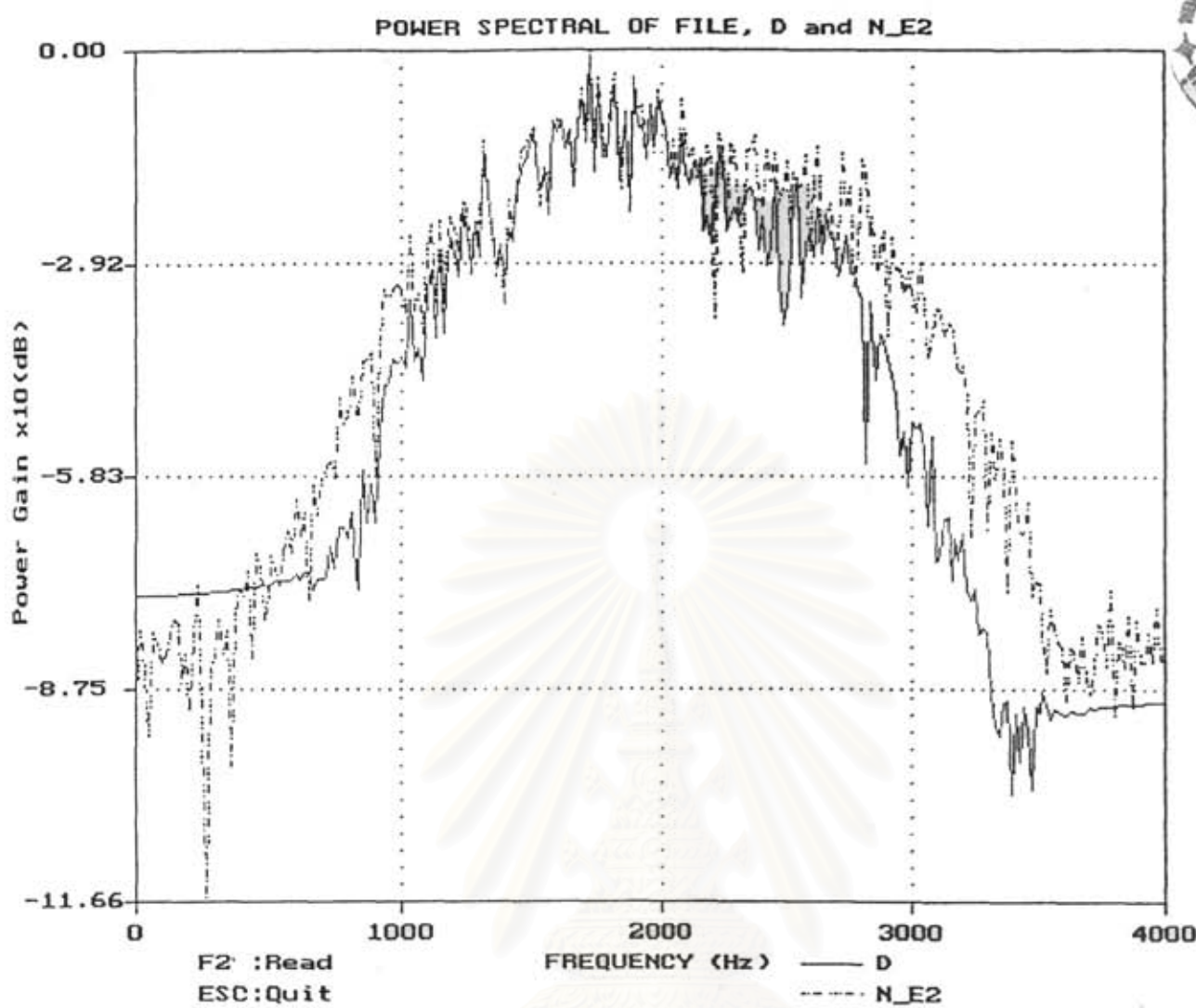
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 6-28

ผลตอบใน time domain เปรียบเทียบระหว่าง สัญญาณเสียงพูดด้าน near-end ชื่อไฟล์ PRJ3 กับ สัญญาณที่ออกจากเครื่องกำจัดเสียงสะท้อน D ซึ่งมีพารามิเตอร์คั้งได้กล่าวไปแล้ว

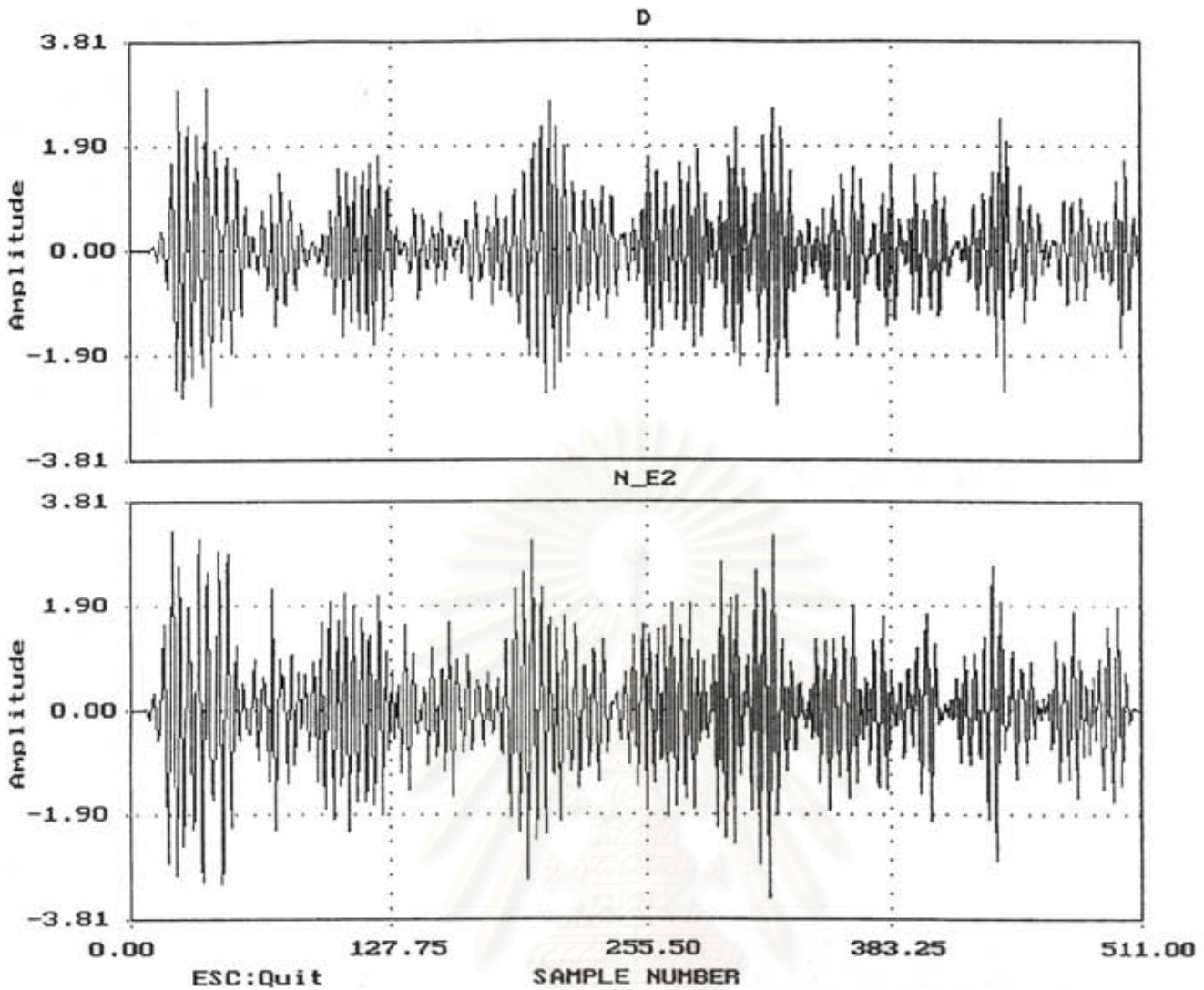
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 6-29

ผลตอบใน frequency domain เปรียบเทียบระหว่าง สัญญาณที่รวมเสียงพูดด้าน near-end (PRJ3) กับสัญญาณเสียงสะท้อน(PRJ5)ชื่อไฟล์ N_E2 กับ สัญญาณที่ออกจากเครื่องกำจัดเสียงสะท้อนชื่อไฟล์ D ซึ่งมีพารามิเตอร์ดังกล่าวไปแล้ว

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 6-30
 ผลตอบใน time domain เปรียบเทียบระหว่าง สัญญาณที่รวมเสียงพูดด้าน near-end (PRJ3) กับ สัญญาณเสียงสะท้อน (PRJ5) ชื่อไฟล์ N_E2 กับ สัญญาณที่ออกจากเครื่องกำจัดเสียงสะท้อน ชื่อไฟล์ D ซึ่งมีพารามิเตอร์ดังกล่าวไปแล้ว

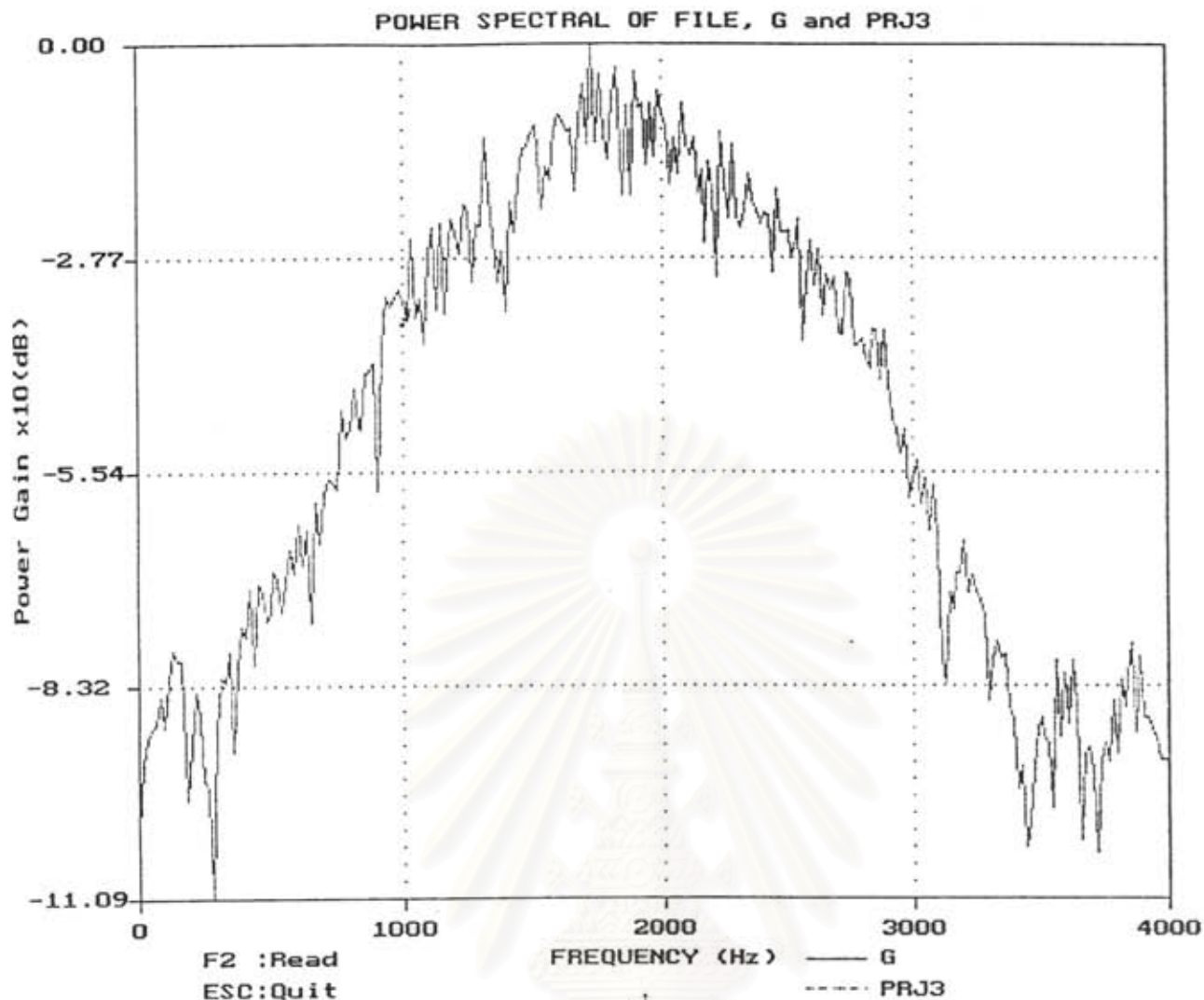
สถาบันวิทยบริการ
 จุฬาลงกรณ์มหาวิทยาลัย

กรณีที่ 3. ไม่มีสัญญาณเสียงสะท้อน

ใช้แฟ้มข้อมูลชื่อ NULLFAR ซึ่งค่าในแฟ้มมีค่าเป็นศูนย์ทุกตัว แทนสัญญาณจาก far-end และสัญญาณเสียงสะท้อน ซึ่งให้ผลเหมือนกับว่าผู้พูดด้าน far-end ไม่ได้พูด แฟ้ม NULL.FAR นี้ผู้จัดทำโครงการได้สร้างขึ้นและมีอยู่ในแผ่นดิสก์ที่แนบมาจากรายงานฉบับนี้ ผลการทดลอง แสดงเพียงชุดเดียวเพราะว่าสัญญาณออกจากเครื่องกำจัดเสียงสะท้อนในกรณีนี้ไม่แตกต่างกัน



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

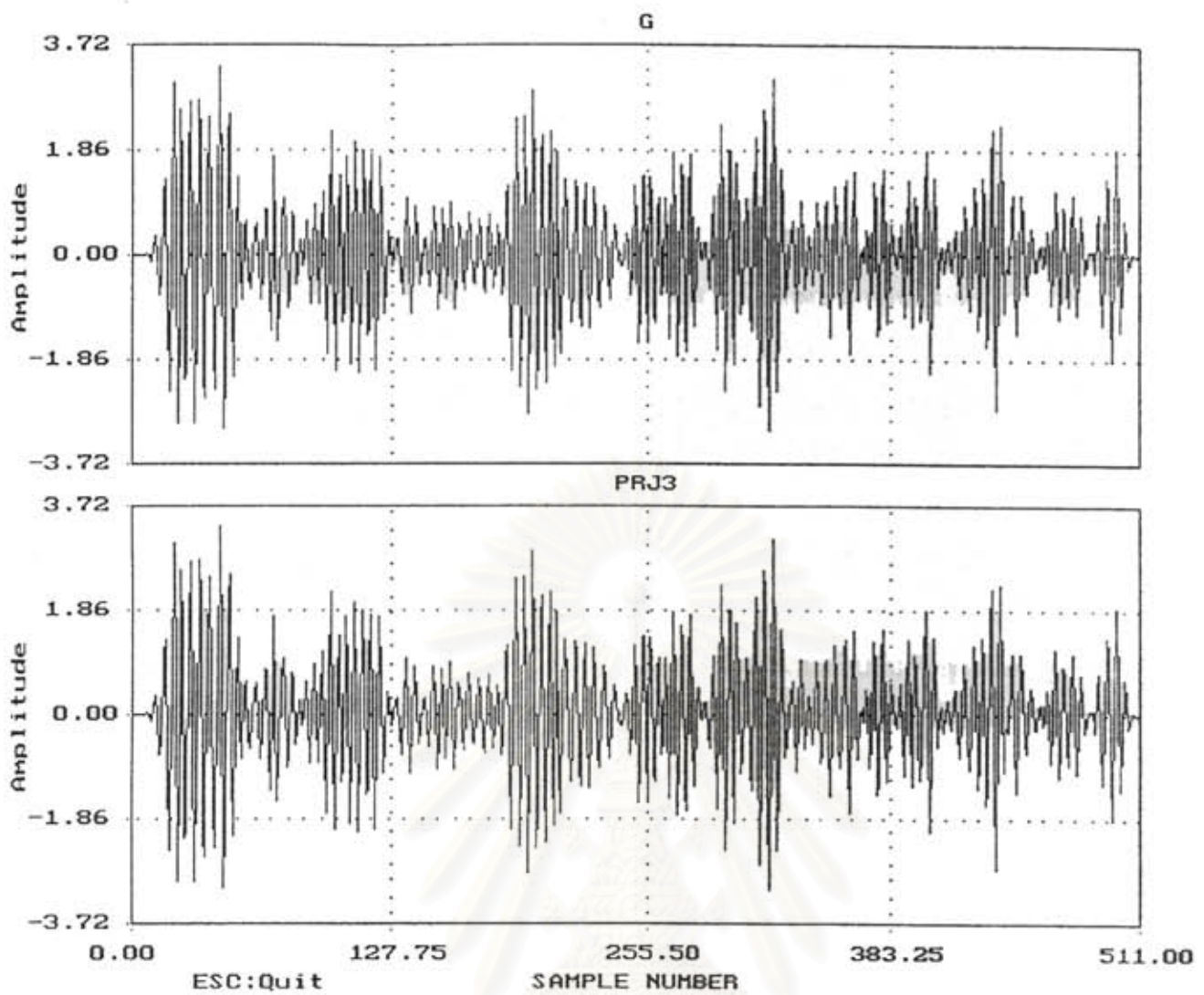


รูปที่ 6-31

ผลตอบใน frequency domain เปรียบเทียบระหว่าง สัญญาณเสียงพูดด้าน near-end ชื่อไฟล์ PRJ3 กับ สัญญาณที่ออกจากเครื่องกำจัดเสียงสะท้อน G ซึ่งมีพารามิเตอร์ดังนี้

- ความยาวของเวกเตอร์นำหนักร่วง = 10
- ค่าเริ่มต้นของเวกเตอร์นำหนักร่วง = 0
- จำนวนรอบของการทำซ้ำ = 100
- Loop gain หรือ Step size หรือ μ = 4×10^{-5}

ซึ่งเมื่อใช้พารามิเตอร์ค่าอื่นๆก็จะให้ผลเช่นเดียวกัน



รูปที่ 6-32

ผลตอบใน time domain เปรียบเทียบระหว่าง สัญญาณเสียงพูดด้าน near-end ชื่อไฟล์ PRJ3 กับ สัญญาณที่ออกจากเครื่องกำจัดเสียงสะท้อน G ซึ่งมีพารามิเตอร์ตั้งได้กล่าวไปแล้ว

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

6.3 วิเคราะห์ผลการทดลอง

ในการวิเคราะห์ผลการทดลองจะวิเคราะห์ทั้งในอาณาจักรเวลาซึ่งทำได้โดยการพล็อตข้อมูลโดยตรงจากแท่งข้อมูลลงบนแกนเวลา และการวิเคราะห์ในอาณาจักรความถี่ซึ่งทำได้โดยการแปลงข้อมูลให้อยู่ในอาณาจักรความถี่ก่อนแล้วจึงทำการพล็อตลงบนแกนความถี่ สำหรับโครงการนี้ใช้ Decimation-In-Time (DIT) FFT algorithm [2] ในการแปลงสัญญาณเป็นอาณาจักรความถี่ ในส่วนของการเขียนโปรแกรมที่ทำ DIT FFT ใช้ algorithm ของ Cooley-Tukey [9]

การทดลองทั้ง 3 กรณีจะให้ค่าพารามิเตอร์คงที่เกี่ยวกับค่าความยาวของเวกเตอร์นำหนักถ่วงที่จะเปลี่ยนค่าไปตามที่ระบุไว้ในผลการทดลอง ผลการทดลองทั้ง 3 กรณีที่ได้แสดงไปแล้วนั้น ถ้าสังเกตในอาณาจักรเวลาแล้วจะเห็นได้ว่ามีความแตกต่างกันของผลที่ได้ในแต่ละกรณีมีน้อยมากจนแทบสังเกตไม่เห็น แต่เมื่อพิจารณาใน อาณาจักรความถี่แล้วจะเห็นความแตกต่างค่อนข้างชัดเจน

ในกรณีที่ 1. เมื่อเปรียบเทียบผลที่ได้จากเครื่องกำจัดเสียงสะท้อนกับสัญญาณเสียงพูดจาก near-end จะสังเกตเห็นได้ว่าในช่วงความถี่ 300-3100 Hz ค่าความยาวของเวกเตอร์นำหนักถ่วงทุกค่าให้ผลเหมือนกัน สำหรับในช่วงความถี่นอกจากนี้จะมีความแตกต่างกันบ้าง คือ ความยาวของเวกเตอร์นำหนักถ่วง = 16 ให้สัญญาณออกจากเครื่องกำจัดเสียงสะท้อนมีลักษณะใกล้เคียงกับสัญญาณเสียงพูดจาก near-end มากที่สุด แต่อย่างไรก็ตามความยาว 4 และ 10 ก็ในผลดีพอควรทั้งนี้เนื่องมาจากความถี่ของเสียงสะท้อนมีค่าต่ำ ดังนั้นความยาวของเวกเตอร์นำหนักถ่วงจึงไม่จำเป็นต้องมากนักก็สามารถที่จะติดตามการเปลี่ยนแปลงของสัญญาณเสียงสะท้อนได้ทัน จากผลการทดลองที่เปรียบเทียบระหว่างสัญญาณรวมของสัญญาณเสียงพูดจาก near-end กับสัญญาณเสียงสะท้อน กับสัญญาณที่ได้จากเครื่องกำจัดเสียงสะท้อน ในรูปที่ 6-7 , 6-8 , 6-11 , 6-12 , 6-15 และ 6-16 เห็นได้ว่าสัญญาณที่ออกจากเครื่องกำจัดเสียงสะท้อนสามารถลดทอนกำลังของเสียงสะท้อนได้มาก

ในกรณีที่ 2. เห็นได้ชัดเจนว่า ความยาวของเวกเตอร์นำหนักถ่วง = 16 ให้ผลดีที่สุดสาเหตุก็เพราะว่า ความถี่ของเสียงสะท้อนมีค่าสูงจึงทำให้ เวกเตอร์นำหนักถ่วงที่มีความยาวน้อยไม่สามารถติดตามการเปลี่ยนแปลงของสัญญาณเสียงสะท้อนได้ทันจึงทำให้ผลที่ออกมาไม่ดี ผลการทดลองในรูปที่ 6-21 , 6-22 , 6-25 , 6-26 , 6-29 และ 6-30 เห็นได้ว่าเครื่องกำจัดเสียงสะท้อนสามารถลดทอนกำลังของเสียงสะท้อนได้มากพอควร

อย่างไรก็ตามทั้ง 2 กรณีนี้ความยาวของเวกเตอร์นำหนักถ่วง = 10 ก็อยู่ในเกณฑ์ที่พอใช้ได้

ในกรณีที่ 3. จากผลการทดลองเมื่อไม่มีเสียงพูดจากด้าน far-end หรือไม่มีเสียงสะท้อนเกิดขึ้น ไม่ว่าจะใช้ค่าความยาวของเวกเตอร์นำหนักถ่วงยาวเท่าไรก็ตาม(แต่ค่า μ ต้องทำให้ระบบมีเสถียรภาพ) จะให้ผลเหมือนกับสัญญาณด้าน near-end มาก

6.4 สรุปผล

นอกจากการทดลองที่ได้เสนอไปแล้วนั้น ผู้จัดทำโครงการยังได้ทำการ ทดลองอื่นๆอีก จนสามารถสรุปผลได้ดังนี้

1. ถ้าความถี่ของเสียงสะท้อนมีค่าสูง ความยาวของเวกเตอร์น้ำหนักถ่วงที่ยาวมากจะให้ผล ได้ดีกว่าความยาวของเวกเตอร์น้ำหนักถ่วงที่สั้น

2. ถ้าความถี่ของเสียงสะท้อนมีค่าต่ำ ความยาวของเวกเตอร์น้ำหนักถ่วงที่ไม่ยาวมากก็ เพียงพอที่จะให้ผลได้ดี

3. ถ้าความยาวของเวกเตอร์น้ำหนักถ่วงมีค่ามากขึ้นต้องลดค่า μ ลงเพื่อให้ระบบมีเสถียรภาพ

4. ถ้าสัญญาณเสียงสะท้อนมีความสัมพันธ์ (cross correlation) กับสัญญาณเสียงพูดจาก ด้าน near-end ผลที่ได้จากเครื่องกำจัดเสียงสะท้อนจะมีลักษณะเหมือนกับสัญญาณด้าน near-end ถูกลดทอนกำลังลงเพราะว่าเครื่องกำจัดเสียงสะท้อนจะกำจัดเสียงที่มีความสัมพันธ์กัน

5. ความยาวของเวกเตอร์น้ำหนักถ่วงยิ่งมากยิ่งให้ผลดีแต่เสียเวลาในการคำนวณนาน

6. ค่า μ ที่น้อยเกินไปจะไม่มีการกำจัดเสียงสะท้อนออกจากสัญญาณเสียงพูดจาก near-end แต่ถ้ามากเกินไปจะทำให้ระบบขาดเสถียรภาพ

7. ถ้าจำนวนรอบในการทำซ้ำมีค่ามากจะเสียเวลาในการคำนวณนาน

8. ค่าพารามิเตอร์ที่เหมาะสมสำหรับเครื่องกำจัดเสียงสะท้อนที่ได้จากการทดลองหลายๆกรณีคือ

ความยาวของเวกเตอร์น้ำหนักถ่วง 10

ค่าเริ่มต้นของเวกเตอร์น้ำหนักถ่วง 0

จำนวนรอบในการทำซ้ำ 100

ค่า μ หรือ loop gain 4×10^{-5}

9. โปรแกรมกำจัดเสียงสะท้อนในโครงการนี้สามารถรับสัญญาณ input จากแท็บเล็ตข้อมูล ที่มีรูปแบบการจัดเก็บข้อมูลตามที่ได้กำหนดไว้ ดังนั้นโครงการนี้สามารถพัฒนาไปสู่การรับ สัญญาณinput จากเสียงพูดจริงโดยการบันทึกเสียงพูดลงแท็บเล็ตให้มีรูปแบบ (format) ตามที่กำหนดไว้ ซึ่งการบันทึกเสียงพูดลงแท็บเล็ตเป็นเรื่องที่ต้องทำการศึกษาต่อไป

6.5 ประโยชน์ที่ได้รับ

ประโยชน์ที่ได้รับมีทั้งต่อตัวผู้ทำโครงการเองและต่อผู้ใช้บริการ โทรศัพท์ทางไกลผ่านดาวเทียมซึ่งประโยชน์ที่มีต่อผู้ทำโครงการที่สำคัญ คือ

1. ได้ความรู้เกี่ยวกับการกำเนิดตัวแปรสุ่มเพื่อใช้ในการจำลองการทำงานระบบต่างๆซึ่งเป็น ประโยชน์ในการศึกษาในระดับสูงขึ้นไป

2. ได้ความรู้เกี่ยวกับ adaptive filter

3. ได้ฝึกการทำงานอย่างมีระบบ

สำหรับประโยชน์ที่ผู้ใช้บริการโทรศัพท์ทางไกลผ่านดาวเทียมที่สำคัญ คือ ช่วยให้ผู้ให้บริการสามารถสื่อสารกันได้อย่างสะดวกมากขึ้นเมื่อมีผู้พัฒนาโครงการนี้ในส่วนของ hardware ต่อไปจนสำเร็จ ดังนั้นโครงการที่ต่อเนื่องกับโครงการนี้ควรทำการพัฒนาต่อไปในส่วนของ hardware ในราคาที่ไม่สูงนัก



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

เอกสารอ้างอิง



1. บุญเลิศ เขี่ยมทัศนาศนา , ยืน ภู่วรรณ และ สมนึก คีรีโต “โปรแกรมคอมพิวเตอร์ภาษา ซี ” , บริษัท ซีเอ็ดดูเคชั่น จำกัด ,1991
2. Alan V, Oppenheim and Renold W. Schafer, “Discrete - Time Signal Processing” , Prentice-Hall , 1989.
3. Averill M. Law and W. Daved Kelton , “Simulation Modeling & Analysis” , McGraw-Hill , 1991.
4. Bernard Widrow, et al , “Adaptive Noise Cancelling: Principles and Applications ”,Proc. IEEE , Dec 1975,PP. 1692-1716.
5. C.W.K Gritton and D.W. Lin , “Echo cancellation Algorithms.” , IEEE ASSP Magazine, April 1984.
6. Herbert Schildt , “C Power User’s Guide” , McGraw-Hill , 1988.
7. Hwei P. Hsu , “Analog and Digital Communication” , McGraw-Hill , 1993.
8. Maurice G. Bellanger , “New application of Digital signal Processing in Communications” , IEEE ASSP Magazine , July 1986.
9. Paul M. Embree and Bruce Kimble , “ C language algorithms for digital signal processing” , Prentice-Hall , 1989.
10. Software ช่วยออกแบบ Digital Filter ของห้องปฏิบัติการ Digital Signal Processing แห่งภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย