# การกำจัดฮาร์โมนิคที่ไม่ต้องการของดิจิตอลคอมบ์ฟิลเตอร์โดยโมเมนต์ของราก

# SPECIFIED HARMONIC REMOVAL OF DIGITAL COMB FILTER BY ROOT MOMENTS

เพ็ญฤดี ตั้งสมวรพงษ์
PENRUEDEE THANGSOMVORAPONG

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2546
ISBN 974-324-733-5

# การกำจัดฮาร์โมนิคที่ไม่ต้องการของดิจิตอลคอมบ์ฟิลเตอร์โดยโมเมนต์ของราก

# SPECIFIED HARMONIC REMOVAL OF DIGITAL COMB FILTER BY ROOT MOMENTS

T048386

เพ็ญฤดี ตั้งสมวรพงษ์

PENRUEDEE THANGSOMVORAPONG

เลขหมู่.....
เลขทะเบียน 48386
วัน, เดือน, ปี 24 ต.ค. 2546

.b.....
.i.....

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2546

ISBN 974-324-733-5

# SPECIFIED HARMONIC REMOVAL OF DIGITAL COMB FILTER BY ROOT MOMENTS

PENRUEDEE THANGSOMVORAPONG

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF ENGINEERING IN INSTRUMENTATION ENGINEERING

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTUTUTE OF TECHNOLOGY LADKRABANG

2003

ISBN 974-324-733-5

COPYRIGHT 2003

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTUTUTE OF TECHNOLOGY LADKRABANG

**บัณฑิตวิทยาลัย**

**สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**

**ใบรับรองวิทยานิพนธ์**

---

**หัวข้อวิทยานิพนธ์** การกำจัดฮาร์โมนิคที่ไม่ต้องการของดิจิตอลคอมบ์ฟิลเตอร์โดยโมเมนต์ของราก
SPECIFIED HARMONIC REMOVAL OF DIGITAL COMB FILTER BY ROOT MOMENTS

**ชื่อนักศึกษา** นางสาวเพ็ญฤดี ตั้งสมวรพงษ์

**รหัสประจำตัว** 43061405

**ปริญญา** วิศวกรรมศาสตรมหาบัณฑิต

**สาขาวิชา** วิศวกรรมการวัดคุม

**อาจารย์ผู้ควบคุมวิทยานิพนธ์** รศ.ดร.ฟูศักดิ์ ชีวสุวิทย์

| คณะกรรมการสอบวิทยานิพนธ์ | | ลายมือชื่อ |
|---|---|---|
| รศ.วิทยา | ทิพย์สุวรรณพร | |
| ผศ.ดร.ธนิตย์ | ตรีสุวรรณวัฒน์ | |
| รศ.พิพัฒน์ | เลาหสงคราม | |
| รศ.ดร.กอบชัย | เดชหาญ | |
| รศ.ดร.ฟูศักดิ์ | ชีวสุวิทย์ | |

**วัน/เดือน/ปี ที่สอบ** 29 กรกฎาคม 2546 เวลา 14.00-16.00 น.

**สถานที่สอบ** ณ อาคาร 12 ชั้น ชั้น 4 (ห้อง E12-403)

**บัณฑิตวิทยาลัยรับรองแล้ว**

**(รศ.ดร.บุญวัฒน์ อัตชู)**

**คณบดีบัณฑิตวิทยาลัย**

**วันที่**.....1.....**เดือน**.....กันยายน.....**พ.ศ**.....2546.....

| | |
|---|---|
| **หัวข้อวิทยานิพนธ์** | การกำจัดฮาร์โมนิคที่ไม่ต้องการของดิจิตอลคอมบ์ฟิลเตอร์โดยโมเมนต์ของราก |
| **นักศึกษา** | นางสาวเพ็ญฤดี ตั้งสมวรพงษ์ |
| **รหัสประจำตัว** | 43061405 |
| **ปริญญา** | วิศวกรรมศาสตรมหาบัณฑิต |
| **พ.ศ.** | 2546 |
| **สาขาวิชา** | วิศวกรรมการวัดคุม |
| **อาจารย์ผู้ควบคุมวิทยานิพนธ์** | รศ.ดร. ฟูศักดิ์ ชีวสุวิทย์ |

## บทคัดย่อ


ในปัจจุบันการประมวลผลสัญญาณดิจิตอลได้รับความนิยมมากขึ้น การออกแบบดิจิตอลฟิลเตอร์ก็ได้พัฒนาขึ้นอย่างรวดเร็วเช่นกันเนื่องจากเป็นองค์ประกอบหลักของงานดังกล่าว ดิจิตอลคอมบ์ฟิลเตอร์ก็เป็นดิจิตอลฟิลเตอร์ประเภทหนึ่งซึ่งได้ถูกนำมาใช้อย่างแพร่หลายเนื่องจากทำหน้าที่คัดกรองและแยกสัญญาณที่มีลักษณะที่เป็นฮาร์โมนิค ดังนั้นจึงมีความจำเป็นอย่างยิ่งที่จะต้องพัฒนาวิธีการและปรับปรุงการออกแบบฟิลเตอร์ประเภทนี้เพื่อทำให้สามารถใช้งานอย่างมีประสิทธิภาพมากขึ้น

งานวิจัยนี้จึงนำเสนอการออกแบบสัมประสิทธิ์ของดิจิตอลคอมบ์ฟิลเตอร์ซึ่งนำความถี่บางฮาร์โมนิคที่ไม่ต้องการออกไป โดยได้นำวิธีของการหาโมเมนต์ของรากเพื่อสร้างสัมประสิทธิ์ที่ต้องการของคอมบ์ฟิลเตอร์ วิธีที่นำเสนอนี้เหมาะสมที่จะใช้ในกรณีความสัมพันธ์ของ $2\pi$ และความถี่ฮาร์โมนิคหลักมูลเป็นเศษส่วน

| | |
|---|---|
| **Thesis Title** | Specified Harmonic Removal of Digital Comb Filter by Root Moments |
| **Student** | Miss Penruedee Thangsomvorapong |
| **Student ID.** | 43061405 |
| **Degree** | Master of Engineering |
| **Programme** | Instrumentation Engineering |
| **Year** | 2003 |
| **Thesis Advisor** | Assoc. Prof. Dr. Fusak Cheevasuvit |


## ABSTRACT

Nowadays, digital signal processing is more and more popular. So, the design algorithm of digital filter is rapidly improved, since it is the main part of the work. Digital comb filter is one kind of digital filter, which is widely used for filtering or separating the desired harmonic frequencies out of interference signals. Therefore, the desired algorithm of filter is essentially improved and developed for higher efficiency.

This thesis, we presents a design algorithm for comb filter coefficients in order to remove some desired harmonic frequencies. The root moment method is used to generate the coefficients of desired comb filter. The proposed method is quite suitable when the relation of $2\pi$ and fundamental frequency $\omega_0$ is fractional.

# กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยคำแนะนำและคำปรึกษาเกี่ยวกับงานวิจัยจาก รศ.ดร.ฟูศักดิ์ ชีวสุวิทย์ ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์ ผู้วิจัยรู้สึกซาบซึ้งในความอนุเคราะห์จากท่าน และขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบคุณ รศ. สักกรียา ชิตวงศ์ และเพื่อนๆ ผู้ให้คำแนะนำที่ดีต่อผู้วิจัยเสมอมา

ขอขอบคุณบิดามารดาของผู้วิจัย ผู้ที่เป็นผู้ให้ตลอดชีวิตของท่านและเป็นกำลังใจที่สำคัญของผู้วิจัย

ขอขอบคุณ นายชนะธิป ธำรงวิทยภาคย์ ที่คอยผลักดัน และเป็นที่ปรึกษาที่ดี

คุณค่าและคุณประโยชน์อันพึงมีจากวิทยานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบแด่ผู้มีพระคุณทุกท่าน

นางสาวเพ็ญฤดี ตั้งสมวรพงษ์

# สารบัญ


| | | หน้า |
|---|---|---|
| บทคัดย่อภาษาไทย | | I |
| บทคัดย่อภาษาอังกฤษ | | II |
| กิตติกรรมประกาศ | | III |
| สารบัญ | | IV |
| สารบัญรูป | | VII |
| บทที่ 1 | บทนำ | 1 |
| | 1.1 ความเป็นมาและความสำคัญของปัญหา | 1 |
| | 1.2 วัตถุประสงค์ของการศึกษา | 1 |
| | 1.3 ขั้นตอนของการศึกษา | 1 |
| บทที่ 2 | การออกแบบดิจิตอลคอมบ์ฟิลเตอร์ | 3 |
| | 2.1 บทนำ | 3 |
| | 2.2 การออกแบบดิจิตอลคอมบ์ฟิลเตอร์โดยวิธีการวางโพลและซีโร่ | 3 |
| | 2.3 การออกแบบดิจิตอลคอมบ์ฟิลเตอร์โดยการออกแบบตัวหน่วงเวลาเป็นเศษส่วน | 6 |
| | 2.3.1 การออกแบบโดยใช้ FIR ฟิลเตอร์(FIR:Finite Impulse Response) | 8 |
| | 2.3.2 การออกแบบโดยใช้ออลพาสฟิลเตอร์ | 11 |
| | 2.4 การประยุกต์ใช้ | 14 |
| | 2.4.1 การประยุกต์ใช้ในการบันทึกสัญญาณเสียง | 15 |
| | 2.4.2. การกำจัดสัญญาณฮาร์โมนิคในการวัดสัญญาณ อิเล็คโทรคาร์ดิโอแกรม | 15 |
| บทที่ 3 | การหาโมเมนต์ของราก | 18 |
| | 3.1 บทนำ | 18 |
| | 3.2 การประมาณค่าโมเมนต์ของรากโดยกระบวนการวนซ้ำ | 19 |
| | 3.2.1 ข้อสังเกตและผลลัพธ์ | 21 |
| | 3.3 โมเมนต์ของรากและการรวมสัญญาณ | 23 |
| | 3.4 การประยุกต์ใช้ | 24 |

# สารบัญ(ต่อ)


| | | หน้า |
|---|---|---|
| | 3.4.1 การหาค่าตัวประกอบเฟสน้อยที่สุด และมากที่สุดของสัญญาณโดยไม่ ต้องหาค่ารากของสัญญาณ [2] ..... | 24 |
| | 3.4.2 การแปลงทรานสเฟอร์ฟังก์ชั่น..... | 30 |
| | 3.4.3 การตรวจจับการเปลี่ยนแปลงในทันทีของสัญญาณ..... | 32 |
| บทที่ 4 | การกำจัดฮาร์โมนิคที่ไม่ต้องการของดิจิตอลคอมบ์ฟิลเตอร์..... | 34 |
| | 4.1 บทนำ..... | 34 |
| | 4.2 การกำจัดฮาร์โมนิคที่ไม่ต้องการของดิจิตอลคอมบ์ฟิลเตอร์..... | 34 |
| | 4.2.1 การกำจัดฮาร์โมนิคที่ไม่ต้องการของดิจิตอลคอมบ์ฟิลเตอร์เมื่อ $D$ เป็นจำนวนเต็ม..... | 35 |
| | 4.2.2 การกำจัดฮาร์โมนิคที่ไม่ต้องการของดิจิตอลคอมบ์ฟิลเตอร์เมื่อ $D$ ไม่เป็นจำนวนเต็ม..... | 35 |
| | 4.3 การกำจัดฮาร์โมนิคที่ต้องการของดิจิตอลคอมบ์ฟิลเตอร์โดยใช้โมเมนต์ของราก..... | 36 |
| | 4.4 ผลการทดลอง..... | 39 |
| | 4.4.1 ผลการจำลองการทำงานโดยใช้โปรแกรม MatLab ..... | 39 |
| | 4.4.2 ผลการทดลองโดยใช้ตัวประมวลผลสัญญาณดิจิตอล..... | 43 |
| | 4.5 สรุปผลการทดลอง | 48 |
| บทที่ 5 | การประยุกต์ใช้ในงานประเภทอื่นๆ..... | 49 |
| | 5.1 การออกแบบออลพาสฟิลเตอร์ที่มีเฟสดีเลย์ตามต้องการ..... | 49 |
| | 5.2 เอฟเฟคของสัญญาณเสียงดิจิตอล..... | 51 |
| | 5.3 การวิเคราะห์ สังเคราะห์สัญญาณเสียงจากเครื่องดนตรี..... | 53 |
| | 5.4 การปรับแต่งเอฟไอาร์คอมบ์ฟิลเตอร์..... | 57 |
| บทที่ 6 | สรุปผลการวิจัยและข้อเสนอแนะ..... | 60 |
| | 6.1 สรุปผลการวิจัย..... | 60 |
| | 6.2 ข้อเสนอแนะ..... | 60 |

# สารบัญ(ต่อ)


| | หน้า |
|---|---|
| เอกสารอ้างอิง..... | 61 |
| ภาคผนวก..... | 63 |
| ก. บทความที่ได้รับการตีพิมพ์ | 63 |
| ข. โปรแกรมการจำลองผลการทำงานโดยใช้ Matlab ..... | 68 |
| ค. โปรแกรมการคำนวณสัมประสิทธิ์และฟิลเตอร์ (TMS320C31)..... | 77 |
| ประวัติผู้เขียน..... | 91 |

# สารบัญรูป


| รูปที่ | | หน้า |
|---|---|---|
| 2.1 | แสดงการวางโพลและซีโร่ของการออกแบบคอมบ์ฟิลเตอร์โดยวิธีการวางโพล และซีโร่ | 3 |
| 2.2 | แสดงผลตอบสนองความถี่ในกรณีที่ $r > R$ | 4 |
| 2.3 | แสดงผลตอบสนองความถี่ในกรณีที่ $R > r$ | 4 |
| 2.4 | แสดงผลตอบสนองความถี่ของคอมบ์ฟิลเตอร์ที่ออกแบบโดยวิธีการวางโพลและซีโร ($\omega_0 = 0.2\pi, D = 10$ ,R =0.999) | 6 |
| 2.5 | แสดงผลตอบสนองความถี่ของดิจิตอลคอมบ์ฟิลเตอร์ $H(z) = 1 - z^{-D}$ (เส้นประ) และผลตอบสนองที่ต้องการ $H_d(\omega)$ (เส้นทึบ) โดยที่ $\omega_0 = 0.22\pi$ และ $M = 4$ | 7 |
| 2.6 | แสดงผลตอบสนองความถี่ของคอมบ์ฟิลเตอร์ของ $H_c(z)$ โดย $\omega_0 = 0.22\pi$ และ $R = 0.99$ | 8 |
| 2.7 | แสดงผลตอบสนองความถี่ของคอมบ์ฟิลเตอร์ออกแบบ $z^{-D}$ โดยใช้วิธีลากรางอินเทอร์โพเลชั่น | 9 |
| 2.8 | แสดงผลตอบสนองความถี่ของดิจิตอลคอมบ์ฟิลเตอร์ที่ออกแบบโดยวิธีการออกแบบเอฟไอาร์ฟิลเตอร์ ($\omega_0 = 0.22\pi, R = 0.999, N = 16$) | 11 |
| 2.9 | แสดงผลตอบสนองความถี่ของดิจิตอลคอมบ์ฟิลเตอร์ที่ออกแบบโดยใช้ออลพาสฟิลเตอร์ | 12 |
| 2.10 | แสดงผลตอบสนองความถี่ของดิจิตอลคอมบ์ฟิลเตอร์ออกแบบโดยใช้ออลพาสฟิลเตอร์ ($\omega_0 = 0.22\pi, \rho = 0.999, N = 9$) | 14 |
| 2.11 | แสดง สัญญาณอิเล็คโทรคาร์ดิโอแกรม | 15 |
| 2.12 | แสดงสัญญาณอิเล็คโทรคาร์ดิโอแกรมภายใน 1 คาบเวลา | 16 |
| 2.13 | แสดงสัญญาณอิเล็คโทรคาร์ดิโอแกรม ที่ถูกรบกวนโดยสัญญาณความถี่ไฟเลี้ยง | 16 |
| 2.14 | แสดงสัญญาณอิเล็คโทรคาร์ดิโอแกรม ที่ผ่านการกรองเพื่อกำจัดสัญญาณรบกวน | 17 |
| 3.1 | แสดงส่วนประกอบเฟสน้อยที่สุด | 29 |
| 3.2 | แสดงส่วนประกอบเฟสมากที่สุด | 30 |
| 3.3 | ผลลัพธ์ของสัญญาณเสียงพูด | 32 |
| 3.4 | แสดงโมเมนต์ของรากของสัญญาณ | 33 |
| 4.1 | ผลตอบสนองขนาดของดิจิตอลคอมบ์ฟิลเตอร์ซึ่ง $\omega_0 = 0.11\pi$ และ $R = 0.999$ | 39 |
| 4.2 | ผลตอบสนองเฟสของดิจิตอลคอมบ์ฟิลเตอร์ซึ่ง $\omega_0 = 0.11\pi$ และ $R = 0.999$ | 39 |

# สารบัญรูป(ต่อ)


| รูปที่ | หน้า |
|---|---|
| 4.3 ผลตอบสนองขนาดของดิจิตอลนอร์ชฟิลเตอร์ซึ่ง $\omega_0 = 0.11\pi$ และ $R = 0.999$ ..... | 40 |
| 4.4 แสดงผลตอบสนองขนาดของดิจิตอลคอมบ์ฟิลเตอร์ซึ่ง $\omega_0 = 0.11\pi$ และ $R = 0.999$ และกำจัด ฮาร์โมนิคคู่..... | 40 |
| 4.5 แสดงผลตอบสนองขนาดของดิจิตอลคอมบ์ฟิลเตอร์ซึ่ง $\omega_0 = 0.11\pi$ และ $R = 0.999$ และกำจัด ฮาร์โมนิคคี่..... | 41 |
| 4.6 แสดงผลตอบสนองขนาดของดิจิตอลคอมบ์ฟิลเตอร์ซึ่ง $\omega_0 = 0.11\pi$ และ $R = 0.999$ และกำจัดฮาร์โมนิคที่ 8..... | 42 |
| 4.7 แสดงรูปแบบในการสร้างไอไออาร์ฟิลเตอร์ (Direct Form-I) ..... | 44 |
| 4.8 แสดงโฟลชาทการคำนวณสัมประสิทธิ์โดยใช้ตัวประมวลผลสัญญาณดิจิตอล..... | 45 |
| 4.9 แสดงผลตอบสนองขนาดของดิจิตอลคอมบ์ฟิลเตอร์เมื่อกำหนดให้ $F_s$ =5863 Hz และ $\omega_0 = 0.11\pi$ (322.5 Hz) ..... | 46 |
| 4.10 แสดงผลตอบสนองเฟสของดิจิตอลคอมบ์ฟิลเตอร์เมื่อกำหนดให้ $F_s$ =5863 Hzและ $\omega_0 = 0.11\pi$ (322.5 Hz) ..... | 46 |
| 4.11 แสดงผลตอบขนาดของดิจิตอลคอมบ์ฟิลเตอร์ที่ตำแหน่งฮาร์โมนิคหลักมูลเมื่อกำหนดให้ $F_s$ =5863 Hz และ $\omega_0 = 0.11\pi$ (322.5 Hz) ..... | 46 |
| 4.12 แสดงผลตอบเฟสของดิจิตอลคอมบ์ฟิลเตอร์ที่ตำแหน่งฮาร์โมนิคหลักมูลเมื่อกำหนดให้ $F_s$ =5863 Hz และ $\omega_0 = 0.11\pi$ (322.5 Hz) ..... | 47 |
| 4.13 แสดงผลตอบสนองขนาดของดิจิตอลคอมบ์ฟิลเตอร์เมื่อกำหนดให้ $F_s$ =5863 Hz และ $\omega_0 = 0.11\pi$ (322.5 Hz) และกำจัดฮาร์โมนิคคู่..... | 47 |
| 4.14 แสดงผลตอบสนองขนาดของดิจิตอลคอมบ์ฟิลเตอร์เมื่อกำหนดให้ $F_s$ =5863 Hz และ $\omega_0 = 0.11\pi$ (322.5 Hz) และกำจัดฮาร์โมนิคคี่..... | 47 |
| 4.15 แสดงผลตอบสนองขนาดของดิจิตอลคอมบ์ฟิลเตอร์เมื่อกำหนดให้ $F_s$ =5863Hz และ $\omega_0 = 0.11\pi$ (3202.5 Hz) และกำจัดฮาร์โมนิคที่ 8..... | 48 |
| 5.1 แสดงผลตอบสนองขนาดของคอมบ์ฟิลเตอร์เมื่อ $\omega_0 = 0.055\pi, R = 0.999$ ..... | 50 |
| 5.2 แสดงผลตอบสนองขนาดของออลพาสฟิลเตอร์เมื่อ $\omega_0 = 0.055\pi, R = 0.999$ ..... | 50 |
| 5.3 แสดงผลตอบสนองเฟสดีเลย์ของออลพาสฟิลเตอร์เมื่อ $\omega_0 = 0.055\pi$, $R =$ 0.999 ..... | 51 |

# สารบัญรูป(ต่อ)


| รูปที่ | หน้า |
|---|---|
| 5.4 แสดงผลตอบสนองเฟสดีเลย์ของออลพาสฟิลเตอร์เมื่อ $R = 0.999$ และกำจัด ฮาร์-โมนิคคี่ | 51 |
| 5.5 แสดงผลตอบสนองเฟสดีเลย์ของออลพาสฟิลเตอร์เมื่อ $\omega_0 = 0.055\pi$, $R = 0.999$ ที่เหลือฮาร์โมนิคที่ 4, 8, 12, 16 | 51 |
| 5.6 แสดงระบบการประมวลผลสัญญาณเสียง | 52 |
| 5.7 แสดงการสร้าง เอฟเฟคเฟสชิพท์ติง | 52 |
| 5.8 แสดงผลตอบสนองเฟส ณ.ตำแหน่งของซี่ | 52 |
| 5.9 แสดงสัญญาณไซน์ที่มีการชิพท์เฟสไปด้วยค่าต่างๆ a. สัญญาณไซน์ b. สัญญาณไซน์ชิพท์เฟสไป $\pi/2$ c.สัญญาณไซน์ชิพท์เฟสไป $\pi$ d. สัญญาณไซน์ชิพท์เฟสไป $3\pi/2$ | 53 |
| 5.10 แสดงฟังก์ชั่นขอบเขตของสัญญาณเสียงจากเครื่องดนตรี a. เซลโล(Cello) b. กีตาร์ (Classical guitar) c. ฟลุต (Flute)d. เฟรนช์ฮอร์น (French horn) | 54 |
| 5.11 แสดงรูปคลื่นที่เกิดจากเครื่องดนตรีชนิดต่างๆ เมื่อเล่นโน้ต A เป็นเวลา 10 มิลลิวินาที a. เซลโล(Cello) b. กีตาร์(Classical guitar) c. ฟลุต (Flute)d. เฟรนช์ฮอร์น (French horn) | 54 |
| 5.12 แสดงโครงสร้างฮาร์โมนิคเมื่อเล่นโน้ต A เป็นเวลา 186 มิลลิวินาที a. เซลโล(Cello) b. กีตาร์(Classical guitar) c. ฟลุต (Flute)d. เฟรนช์ฮอร์น (French horn) | 55 |
| 5.13 แสดงสเปรคตรัมของเฟร็นช์ฮอร์น(8192 จุด) | 56 |
| 5.14 แสดงรูปคลื่นของเฟรนช์ฮอร์น โดยรูป a ไม่มีการกรองฮาร์โมนิคใดๆ รูป b กรองทุกฮาร์โมนิค รูป c ไม่กรองฮาร์โมนิคที่ 1 รูป d ไม่กรองฮาร์โมนิคที่ 2 เมื่อกำหนดให้ ( $\omega_0 = 0.02\pi, R = 0.999$ ) | 56 |
| 5.15 แสดงตำแหน่งของซีโร่เมื่อต้องการแปลงเป็นฟิลเตอร์อื่นๆ | 57 |
| 5.16 แสดงตำแหน่งของซีโร่เมื่อต้องการแปลงเป็นตัวกรองความถี่ต่ำผ่าน เมื่อ $\omega_0 = 0.22\pi$ | 58 |
| 5.17 แสดงตำแหน่งของซีโร่เมื่อต้องการแปลงเป็นตัวกรองความถี่สูงผ่าน เมื่อ $\omega_0 = 0.22\pi$ | 58 |
| 5.18 แสดงตำแหน่งของซีโร่เมื่อต้องการแปลงเป็นตัวกรองความช่วงความถี่ผ่าน เมื่อ $\omega_0 = 0.22\pi$ | 58 |

# บทที่ 1
# บทนำ

## 1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากความสำคัญของดิจิตอลคอมบ์ฟิลเตอร์ ซึ่งนำมาใช้ในการประมวลผลสัญญาณดิจิตอลในงานประเภทต่างๆ เป็นต้นว่าใช้ในการกำจัดสัญญาณรบกวนที่เป็นฮาร์โมนิค การแยกสัญญาณที่มีความถี่แตกต่างกันสองสัญญาณออกจากกัน หรืออาจเป็นส่วนหนึ่งของการจัดการสัญญาณที่มีโครงสร้างเป็นฮาร์โมนิค

ด้วยเหตุนี้จึงมีผู้นำเสนอและศึกษาวิธีการออกแบบดิจิตอลคอมบ์ฟิลเตอร์ชนิดต่างๆ อย่างเช่น เอฟไออาร์(Finite Impulse Response) คอมบ์ฟิลเตอร์ ไอไอาร์(Infinite Impulse Response)คอมบ์ฟิลเตอร์ อแดพทีพคอมบ์ฟิลเตอร์ แต่ในวิทยานิพนธ์นี้ได้มุ่งเน้นถึงปัญหาในการออกแบบไอไอาร์คอมบ์ฟิลเตอร์ โดยศึกษาและปรับปรุงการออกแบบและการกำจัดฮาร์โมนิคที่ต้องการ การคำนวณสัมประสิทธิ์ของดิจิตอลไอไอาร์คอมบ์ฟิลเตอร์ในกรณีที่ $D$ ไม่เป็นจำนวนเต็ม (เมื่อ $D = 2\pi/\omega_0$) ซึ่งจากการศึกษาดังกล่าวจะช่วยให้การคำนวณหาสัมประสิทธิ์มีประสิทธิภาพ มีความเหมาะสมในการนำไปสร้างบนตัวประมวลผลสัญญาณดิจิตอล และสะดวกในการแปลงเป็นดิจิตอลฟิลเตอร์ประเภทอื่นๆ

## 1.2 วัตถุประสงค์ของการศึกษา

1. ศึกษาการออกแบบดิจิตอลคอมบ์ฟิลเตอร์(ไอไอาร์คอมบ์ฟิลเตอร์) วิธีต่างๆ ศึกษาถึงความสัมพันธ์ของการออกแบบแต่ละวิธี
2. ปรับปรุงวิธีการออกแบบ การคำนวณสัมประสิทธิ์ในกรณีต้องการกำจัดฮาร์โมนิคที่ต้องการของดิจิตอลคอมบ์ฟิลเตอร์
3. ศึกษาการนำไปประยุกต์ใช้ในงานต่างๆ

## 1.3 ขั้นตอนของการศึกษา

มีลำดับขั้นตอนของการศึกษาและรายละเอียดในแต่ละบทดังต่อไปนี้

บทที่ 1 บทนำ

บทที่ 2 การออกแบบดิจิตอลคอมบ์ฟิลเตอร์ กล่าวถึงวิธีการออกแบบดิจิตอลคอมบ์ฟิลเตอร์(ไอไอาร์คอมบ์ฟิลเตอร์) โดยวิธีต่างๆ และความสัมพันธ์ของแต่ละวิธี

บทที่ 3 การหาโมเมนต์ของราก กล่าวถึงความหมาย การคำนวณหาโมเมนต์ของรากและการนำไปประยุกต์ใช้ในกรณีต่างๆ

บทที่ 4 การกำจัดฮาร์โมนิคที่ไม่ต้องการของดิจิตอลคอมบ์ฟิลเตอร์โดยวิธีโมเมนต์ของราก คือ การปรับปรุงวิธีคำนวณสัมประสิทธิ์ของไอไออาร์ฟิลเตอร์ในการกำจัดบางฮาร์โมนิคที่ไม่ต้องการ ผลการจำลองการทำงาน และผลการทดลองโดยตัวประมวลผลสัญญาณดิจิตอล

บทที่ 5 การประยุกต์ใช้ กล่าวถึงการนำวิธีดังกล่าวไปประยุกต์ใช้ในงานประเภทต่างๆ

บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ กล่าวถึงสรุปผลการวิจัย และข้อเสนอแนะในการนำไปประยุกต์ใช้ในงานประเภทอื่นๆต่อไป

# บทที่ 2
# การออกแบบดิจิตอลคอมบ์ฟิลเตอร์

## 2.1 บทนำ

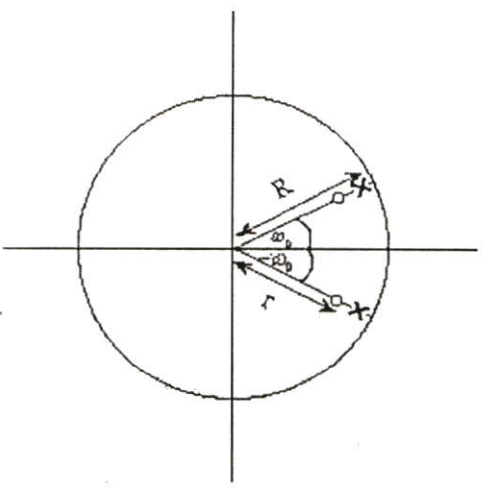
ดิจิตอลคอมบ์ฟิลเตอร์เป็นฟิลเตอร์ประเภทหนึ่งที่ถูกนำไปใช้ในการกรองสัญญาณรบกวนที่มีลักษณะเป็นฮาร์โมนิคหรือนำไปใช้ในการแยกสัญญาณ 2 สัญญาณออกจากกัน โดยที่สัญญาณมีลักษณะเป็นแบนด์แคบๆ

การออกแบบดิจิตอลคอมบ์ฟิลเตอร์สามารถกระทำได้หลายวิธี เช่น วิธีการออกแบบโดยการกำหนดตำแน่งของโพลและซีโร่ และวิธีการการออกแบบคอมบ์ฟิลเตอร์โดยการออกแบบ $z^{-D}$ แต่ไม่ว่าเป็นการออกแบบโดยวิธีใดๆ ก็ตามจุดสำคัญของการออกแบบก็คือต้องการผลตอบสนองความถี่ของฟิลเตอร์ที่มีผลตอบสนองขนาด (Magnitude Response)ที่ความถี่ฮาร์โมนิคหรือ ความถี่ที่ต้องการกำจัด และอัตราลดทอน ณ ตำแหน่งของความถี่นั้นต้องมีอัตราการลดทอนที่ต่ำมากหรือขนาดของผลตอบสนองเป็น 0

## 2.2 การออกแบบดิจิตอลคอมบ์ฟิลเตอร์โดยวิธีการวางโพลและซีโร่ [1]

เป็นวิธีการออกแบบดิจิตอลคอมบ์ฟิลเตอร์ ที่สามารถออกแบบได้ง่าย ให้ผลตอบสนองความถี่ที่ดี

พิจารณาตำแหน่งของโพลและซีโร่ของวงกลม 1 หน่วย ดังรูป




**รูปที่ 2.1** แสดงการวางโพลและซีโร่ของการออกแบบคอมบ์ฟิลเตอร์โดยวิธีการวางโพล (X)และซีโร่(O)

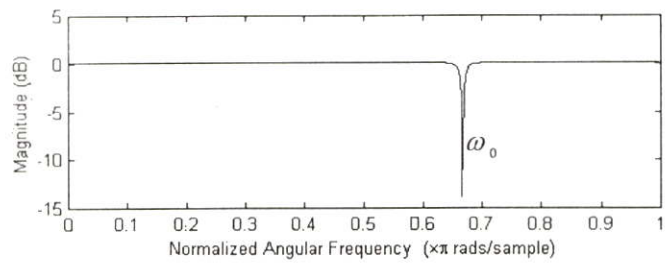
เมื่อ $$p_1 = \mathrm{Re}^{j\omega_0} \quad (2.1)$$

$$\zeta_2 = re^{j\omega_0} \quad (2.2)$$

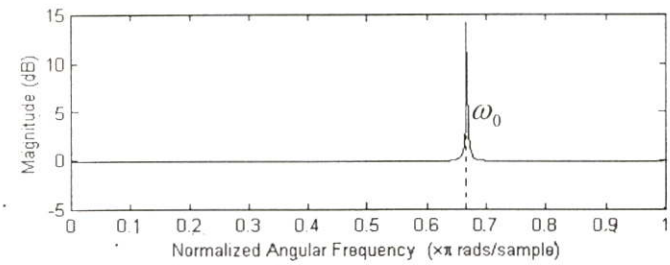
$R$ คือขนาดของโพล

$r$ คือขนาดของซีโร่

จากวิธีการดังรูปที่ 2.1 จะให้ผลตอบสนองดังรูปที่ 2.2 ถ้า $r > R$ (โดยทั่วไปให้ $r = 1$) และถ้า $R > r$ (เมื่อ $R \neq 1$) ให้ผลตอบสนองดังรูปที่ 2.3




**รูปที่ 2.2** แสดงผลตอบสนองความถี่ในกรณีที่ $r > R$




**รูปที่ 2.3** แสดงผลตอบสนองความถี่ในกรณีที่ $R > r$

จากรูปที่ 2.2 ทรานสเฟอร์ฟังก์ชั่นของวงจรกรองเมื่อ $r = 1$ คือ

$$H(z) = \frac{\left(1 - e^{j\omega_0} z^{-1}\right)\left(1 - e^{-j\omega_0} z^{-1}\right)}{\left(1 - Re^{j\omega_0} z^{-1}\right)\left(1 - Re^{-j\omega_0} z^{-1}\right)} = \frac{1 + b_1 z^{-1} + b_2 z^{-2}}{1 + Rb_1 z^{-1} + R^2 b_2 z^{-2}} \quad (2.3)$$

$$= \frac{N(z)}{N(R^{-1}z)}$$

เมื่อ $b_n$ คือสัมประสิทธิ์ของโพลิโนเมียลของทรานสเฟอร์ฟังก์ชั่น

ฉะนั้นเมื่อต้องการสร้างคอมบ์ฟิลเตอร์ที่มีผลตอบสนองขนาดทุกๆ $\omega_i$ เพื่อกำจัดสัญญาณที่เป็นฮาร์โมนิค

$$N(z) = \prod_{i=1}^{M} \left(1 - e^{j\omega_i} z^{-1}\right) \tag{2.4}$$

$$P(z) = N(R^{-1} z^{-1}) \tag{2.5}$$

เมื่อ $P(z)$ คือ ส่วนของทรานสเฟอร์ฟังก์ชั่น

$$P(z) = \prod_{i=1}^{M} \left(1 - Re^{j\omega_i} z^{-1}\right) \tag{2.6}$$

หรือทรานสเฟอร์ฟังก์ชั่นของวงจรกรองคือ

$$H(z) = \frac{N(z)}{N(R^{-1} z)} \tag{2.7}$$

$$= \frac{1 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_M z^{-M}}{1 + Rb_1 z^{-1} + R^2 b_2 z^{-2} + \ldots + R^M b_M z^{-M}}$$

$$= \frac{1 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_M z^{-M}}{1 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_M z^{-M}}$$

เมื่อ $a_i = R^i b_i, \qquad i = 1, 2, \ldots, M$

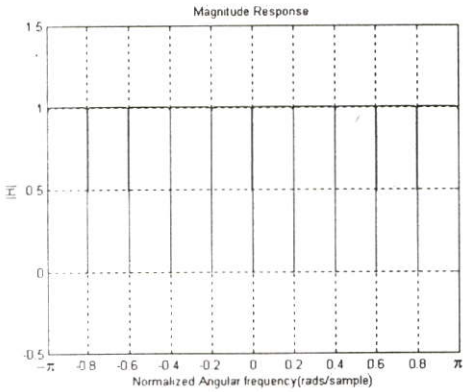
กำหนดให้ $D = 2\pi / \omega_i$ เมื่อ $\omega_i = 2\pi f_i / F_s$ ( $f_i$ คือความถี่ฮาร์โมนิค , $F_s$ คือความถี่แซมปลิ้ง) พบว่าเมื่อ D เป็นจำนวนเต็มแล้วทรานสเฟอร์ฟังก์ชั่นจะอยู่ในรูปแบบดังนี้

$$H(z) = \frac{1 - z^{-D}}{1 - R^D z^{-D}} \tag{2.8}$$

เช่น กำหนดให้ $\omega_0 = 0.2\pi, D = 10$ ทรานสเฟอร์ฟังก์ชั่นของมันคือ

$$H(z) = \frac{1 - z^{-10}}{1 - R^{10} z^{-10}} \tag{2.9}$$

กำหนดให้ R =0.999 ให้ผลตอบสนองดังรูป




**รูปที่ 2.4** แสดงผลตอบสนองความถี่ของคอมบ์ฟิลเตอร์ที่ออกแบบโดยวิธีการวางโพลและซีโร ($\omega_0 = 0.2\pi, D = 10,$ R =0.999)

## 2.3 การออกแบบดิจิตอลคอมบ์ฟิลเตอร์โดยการออกแบบตัวหน่วงเวลาเป็นเศษส่วน [2]

โดยทั่วไปแล้ว สัญญาณอินพุทของคอมบ์ฟิลเตอร์มีรูปแบบดังนี้

$$\begin{aligned} x(n) &= s(n) + \sum_{k=0}^{M} A_k \sin(k\omega_0 n + \phi_k) \\ &= s(n) + I(n) \end{aligned} \tag{2.10}$$

เมื่อ $s(n)$ คือสัญญาณที่ต้องการและ $I(n)$ คือสัญญาณรบกวนที่มีลักษณะเป็นฮาร์โมนิคที่มีความถี่ฮาร์โมนิคหลักมูล $\omega_0$ (Fundamental Harmonic Frequency) ในการที่จะแยกสัญญาณ $s(n)$ จากสัญญาณ $x(n)$ โดยไม่เกิดการลดทอนซึ่งต้องใช้คอมบ์ฟิลเตอร์ที่มีคุณสมบัติดังนี้

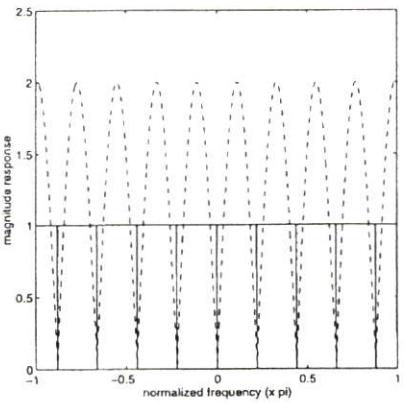
$$H_d(\omega) = \begin{cases} 0 & \omega = k\omega_0 \qquad k = 0,1,...,M \\ 1 & \text{อื่น ๆ} \end{cases} \tag{2.11}$$

เมื่อกำหนดให้ $D = 2\pi / \omega_0$ ซึ่งเป็นคาบของสัญญาณ $I(n)$ จะได้

$$\begin{aligned} I(n-D) &= \sum_{k=0}^{M} A_k \sin\left[k\omega_0 (n-D) + \phi_k\right] \\ &= \sum_{k=0}^{M} A_k \sin\left[k\omega_0 n + \phi_k + 2k\pi\right] \\ &= I(n) \end{aligned} \tag{2.12}$$

ซึ่งแสดงให้เห็นว่า $I(n)$ จะเท่ากับ $I(n)$ ที่หน่วงเวลาไป $D$ หรือ $I(n)=I(n-D)$ ฉะนั้นเมื่อกรองสัญญาณ $x(n)$ ด้วยฟิลเตอร์ $H(z)$ ซึ่งเท่ากับ $1-z^{-D}$ แล้วเอาท์พุท $y(n)$ คือ

$$\begin{aligned} y(n) &= x(n)-x(n-D) \\ &= [s(n)+I(n)]-[s(n-D)+I(n-D)] \\ &= s(n)-s(n-D) \end{aligned} \tag{2.13}$$




**รูปที่ 2.5** แสดงผลตอบสนองความถี่ของดิจิตอลคอมบ์ฟิลเตอร์ $H(z)=1-z^{-D}$ (เส้นประ) และผลตอบสนองที่ต้องการ $H_d(\omega)$ (เส้นทึบ) โดยที่ $\omega_0=0.22\pi$ และ $M=4$

ซึ่งแสดงให้เห็นอย่างชัดเจนว่า สัญญาณรบกวนที่เป็นฮาร์โมนิคถูกกำจัดออกจากเอาท์พุท $y(n)$ อย่างไรก็ตาม $y(n)$ ก็ยังไม่เท่ากับ $s(n)$ พบว่า $y(n)$ นั้นถูกลดทอนลงไป เพื่ออธิบายปรากฏการณ์นี้ พิจารณารูปที่ 2.5 ซึ่งแสดงผลตอบสนองความถี่ของ $H(z)=1-z^{-D}$ และผลตอบสนองความถี่ที่ต้องการ $H_d(\omega)$ เมื่อกำหนดให้ $\omega_0=0.22\pi$ และ $M=4$ ให้สังเกตว่า $M=\pi/\omega_0$ โดยที่ $M$ คือจำนวนเต็มที่มากที่สุดแต่จะน้อยกว่าหรือเท่ากับ $\pi/\omega_0$ จากรูปที่ 2.5 พบว่าผลตอบสนองทั้งคู่มีตำแหน่งของซี่ที่ตรงกันแต่ความแตกต่างกันในช่วงความถี่ผ่าน ดังนั้นเมื่อนำการลดทอนออกไป จึงต้องทำการชดเชย จากตำแหน่งซีโร่ของ $H(z)=1-z^{-D}$ ดังต่อไปนี้

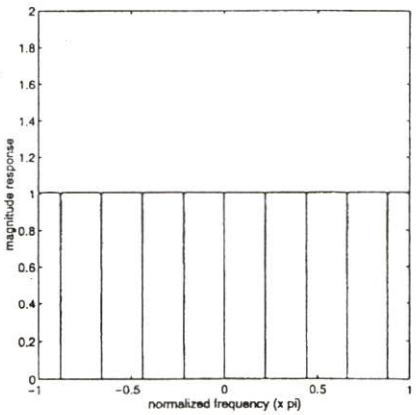
$$z_k = e^{j(2\pi/D)k} \qquad \text{เมื่อ } k \text{ คือค่าจำนวนเต็มใดๆ} \tag{2.14}$$

ทุกๆ ซีโร่ $z^k$ จะเพิ่มโพลที่สอดคล้องเข้าไป

$$p_k = Re^{j(2\pi/D)k} \qquad \text{เมื่อ } k \text{ คือค่าจำนวนเต็มใดๆ} \tag{2.15}$$

เพื่อที่จะกำจัดการลดทอนในช่วงความถี่ผ่านของ $H(z)=1-z^{-D}$ รัศมีของโพล $\rho$ จะมีค่าอยู่ระหว่าง 0 และ 1 ภายในวงกลม 1 หน่วย หลังจากทำการชดเชยแล้วทรานสเฟอร์ฟังก์ชั่นใหม่คือ

$$H_c(z)=\frac{1-z^{-D}}{1-R^D z^{-D}} \tag{2.16}$$




**รูปที่ 2.6** แสดงผลตอบสนองความถี่ของคอมบ์ฟิลเตอร์ของ $H_c(z)$ โดย $\omega_0=0.22\pi$ และ $R=0.99$

รูปที่ 2.6 แสดงผลตอบสนองความถี่ของ $H_c(z)$ เมื่อกำหนดพารามิเตอร์ $\omega_0=0.22\pi$, $R=0.99$ ซึ่ง $H_c(z)$ คือคอมบ์ฟิลเตอร์ตามจินตภาพ เมื่อ $\rho$ มีค่าเข้าใกล้ 1 และถ้า D เป็นจำนวนเต็มแล้วไม่จำเป็นต้องใช้การออกแบบใดๆ แต่ในกรณีที่ D ไม่เป็นจำนวนเต็ม จำเป็นต้องอาศัยการออกแบบตัวหน่วงเวลาเป็นเศษส่วน ซึ่งอาศัยการออกแบบโดยการประมาณซึ่งทำได้ 2 วิธีคือ

1. ออกแบบโดยใช้ เอฟไอาร์ฟิลเตอร์(FIR)ความถี่ต่ำผ่าน
2. ออกแบบโดยใช้ออลพาสฟิลเตอร์

### 2.3.1 การออกแบบโดยใช้ FIR ฟิลเตอร์(FIR:Finite Impulse Response)

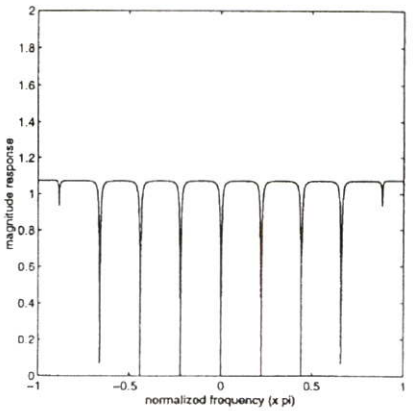
การออกแบบ $z^{-D}$ โดย FIR ฟิลเตอร์นั้นสามารถออกแบบได้หลายวิธี [3] ตัวอย่างเช่น การออกแบบโดยวิธีลากรางอินเทอร์โพเลชั่น ซึ่ง $z^{-D}$ จะถูกประมาณจาก

$$z^{-D}\approx\sum_{n=0}^{N}h(n)z^{-N} \tag{2.17}$$

และ $h(n)$ จะหาได้จาก

$$h(n) = \prod_{k=0,k\neq n}^{N} \frac{D-k}{n-k} \qquad n = 0,1,...,N \tag{2.18}$$

แต่พบว่าเมื่อใช้การออกแบบ $z^{-D}$ วิธีใดๆนั้น เมื่อนำค่าของ $z^{-D}$ มาแทนลงในทรานสเฟอร์ฟังก์ชั่นของดิจิตอลคอมบ์ฟิลเตอร์ พบว่าให้ผลตอบสนองความถี่ได้ไม่ดีเท่าที่ควร ดังเช่นรูปที่ 2.7 แสดงผลตอบสนองความถี่ของดิจิตอลคอมบ์ฟิลเตอร์ซึ่ง $\omega_0 = 0.22\pi, R = 0.99$ และ $N = 16$ เมื่อออกแบบ $z^{-D}$ โดยวิธีของลากรางมัลติไพลเออร์ ฉะนั้นจึงนำไปสู่การออกแบบ $z^{-D}$ โดยเอฟไออาร์ฟิลเตอร์บนพื้นฐานของข้อจำกัดที่ต้องการ




**รูปที่ 2.7** แสดงผลตอบสนองความถี่ของคอมบ์ฟิลเตอร์ออกแบบ $z^{-D}$ โดยใช้วิธี Lagrange Interpolation

**การออกแบบเอฟไออาร์ฟิลเตอร์แบบมีข้อจำกัด[2]**

ทรานสเฟอร์ฟังก์ชั่นของเอฟไออาร์ฟิลเตอร์ออเดอร์ N คือ

$$H(z) = \sum_{i=0}^{N} h(n) z^{-N} \tag{2.19}$$

ผลตอบสนองความถี่ของเอฟไออาร์ฟิลเตอร์ดังกล่าวคือ

$$H(\omega) = h^t e(\omega) = e^t(\omega) h \tag{2.20}$$

เมื่อเวคเตอร์ $h$ และ $e(\omega)$ คือ

$$h = \begin{bmatrix} h(0) & h(1) & \dots & h(N) \end{bmatrix}^t \tag{2.21}$$

$$e(\omega) = \begin{bmatrix} 1 & e^{-j\omega} & \dots & e^{-jN\omega} \end{bmatrix}^t \tag{2.22}$$

เมื่อ $h(n)$ เป็นจำนวนจริง ผลตอบสนองความถี่ $H(\omega)$ จะเป็นค่าสังยุคโดยที่

$$H(-\omega) = H(\omega)^* \tag{2.23}$$

เครื่องหมาย "*" เป็นการบอกว่าฟังก์ชั่นนั้นเป็นฟังก์ชั่นสังยุค

ในการออกแบบตัวหน่วงเวลาเป็นเศษส่วน $F_d(\omega)$ โดยมีผลตอบสนองความถี่ที่ต้องการก็คือ $e^{-jD\omega}$ สัมประสิทธิ์ของฟิลเตอร์ $h(n)$ จะได้จากการมินิไมซ์ตามวิธีของลีสท์สแควร์ดังนี้

$$J(h) = \int_{\omega \in (R^+ \cup R^-)} \left| H(\omega) - F_d(\omega) \right|^2 d\omega \tag{2.24}$$

เมื่อช่วงความถี่ $R^+ = [0, \alpha\pi]$ และ $R^- = [-\alpha\pi, 0]$ จากคุณสมบัติสังยุคของ $H(\omega)$ และ $F_d(\omega)$ สามารถเขียน $J(h)$ ในรูปแบบใหม่ได้ดังนี้

$$J(h) = h^t Q h - 2h^t p + C \tag{2.25}$$

เมื่อเมตริกซ์ $Q$ เวคเตอร์ $p$ และสเกลาร์ $C$ เป็นจำนวนจริงและคำนวณจากสมการต่อไปนี้

$$Q = 2 \int_{\omega \in R^+} \mathrm{Re}\left[ e(\omega) e^h(\omega) \right] d\omega \tag{2.26}$$

$$p = 2 \int_{\omega \in R^+} \mathrm{Re}\left[ F_d(\omega) e^*(\omega) \right] d\omega \tag{2.27}$$

$$C = 2 \int_{\omega \in R^+} \mathrm{Re}\left[ F_d(\omega) \times F_d(\omega) \right] d\omega = 2\alpha\pi$$

(2.28)

เมื่อ $h$ คือตัวกระทำเฮอร์มิเชียนคอนจูเกททรานสโพส และ $\mathrm{Re}(\bullet)$ คือส่วนจริงของจำนวนเชิงซ้อน และเพื่อทำให้ดิจิตอลคอมบ์ฟิลเตอร์มีค่า 0 ที่ความถี่ฮาร์โมนิค $k\omega_0$ ต้องกำหนดข้อจำกัดเหล่านี้ในการออกแบบ

$$H(k\omega_0) = e^{-jDk\omega_0}, k = 0, 1, \dots, M \tag{2.29}$$

เมื่อ Im(•) และข้อจำกัดสามารถเขียนอยู่ในรูปแบบเมตริกซ์คือ $Ch = f$ เมื่อเมตริกซ์ $C$ และเวคเตอร์ $f$ คือ·

$$C = \{\mathrm{Re}[e(0)],\ \mathrm{Re}[e(\omega_0)],\ \mathrm{Im}[e(\omega_0)],\ ...,\ \mathrm{Re}[e(M\omega_0)]\ \ \mathrm{Im}[e(M\omega_0)]\}' \quad (2.30)$$

$$f = [1,\ \cos(D\omega_0),\ -\sin(D\omega_0),\ ...,\ \cos(DM\omega_0)\ \ -\sin(DM\omega_0)]' \quad (2.31)$$

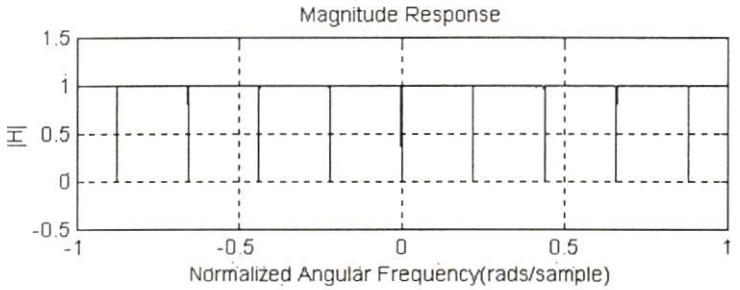
โดย Im[ ] คือส่วนจินตภาพของจำนวนเชิงซ้อนจากรายละเอียดที่กล่าวมา ปัญหาของการออกแบบกลายเป็น

$$\begin{aligned} &Minimize && h'Qh - 2h'p + C \\ &Subject && Ch = f \end{aligned} \quad (2.32)$$

เมื่อใช้วิธีของลากรานมัลติไพลเออร์ โซลูชั่นที่ดีที่สุดของระบบคือ

$$h = Q^{-1}p - Q^{-1}C'(CQ^{-1}C')^{-1}[CQ^{-1}p - f] \quad (2.33)$$

ในกรณีที่กำหนดค่าของพารามิเตอร์คือ $\omega_0 = 0.22\pi, R = 0.999, N = 16$ จะได้ผลตอบสนองความถี่ของฟิลเตอร์ $H_c(z)$ ที่ออกแบบโดยพารามิเตอร์ดังกล่าวและผลตอบสนองที่ความถี่ $k\omega_0$ มีค่าเป็น 0 ดังรูปที่ 2.8




**รูปที่ 2.8** แสดงผลตอบสนองความถี่ของดิจิตอลคอมบ์ฟิลเตอร์ที่ออกแบบโดยวิธีการออกแบบเอฟไออาร์ฟิสเตอร์ ($\omega_0 = 0.22\pi, R = 0.999, N = 16$)

### 2.3.2 การออกแบบโดยใช้ออลพาสฟิลเตอร์

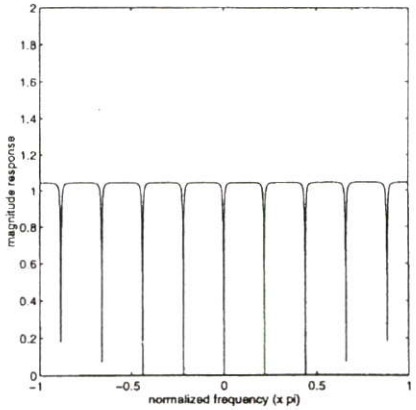
สามารถออกแบบ $z^{-D}$ โดยใช้ออลพาสฟิลเตอร์ด้วยสมการการประมาณดังต่อไปนี้

$$z^{-D} \approx \frac{a_N + a_{N-1}z^{-1} + \ldots + a_1 z^{-(N-1)} + z^{-N}}{1 + a_1 z^{-1} + \ldots + a_{N-1}z^{-(N-1)} + a_N z^{-N}} \tag{2.34}$$

เมื่อจำนวนจริงบวก D เท่ากับค่าจำนวนเต็ม N บวกกับค่าเศษ d หรือ $D = N + d$ แล้ว สัมประสิทธิ์ของฟิลเตอร์ $a_k$ คำนวณได้จาก

$$a_k = (-1)^k C_k^N \prod_{n=0}^{N} \frac{D - N + n}{D - N + k + n} \tag{2.35}$$

โดยที่ $C_k^N = N!/k!(N-k)!$ คือสัมประสิทธิ์ของไบโนเมียล รูปที่ 2.9 แสดงผลตอบสนองของดิจิตอลคอมบ์ฟิลเตอร์ที่ออกแบบโดยออลพาสฟิลเตอร์แล้วแทนลงไปในสมการที่ (2.10) พบว่าให้ผลตอบสนองที่ดีเฉพาะความถี่ต่ำๆ ซึ่งพบว่าเมื่อออกแบบออลพาสฟิลเตอร์แล้วแทนลงไปในสมการที่ (2.34) โดยตรงนั้นให้ผลตอบสนองความถี่ที่ไม่ดีเท่าที่ควร ฉะนั้นจึงต้องใช้การออกแบบออลพาสฟิลเตอร์ร่วมกับการออพติไมซ์เซชั่นดังจะกล่าวต่อไปนี้




**รูปที่ 2.9** แสดงผลตอบสนองความถี่ของดิจิตอลคอมบ์ฟิลเตอร์ที่ออกแบบโดยใช้ออลพาสฟิลเตอร์

**การออกแบบคอมบ์ฟิลเตอร์ (การออกแบบออลพาสฟิลเตอร์ (แบบมีข้อจำกัด))**

จากสมการที่ (2.34) ผลตอบสนองเฟสของออลพาสฟิลเตอร์คือ

$$\theta_A(\omega) = -N\omega + 2\,\text{arc}\,\tan\left[\frac{\sum_{k=1}^{N} a_k \sin(k\omega)}{1 + \sum_{k=1}^{N} a_k \cos(k\omega)}\right] \tag{2.36}$$

จุดประสงค์ของการออกแบบคือการประมาณค่า $\theta_A(\omega)$ ให้เท่ากับ $-D(\omega)$ ซึ่งเป็นค่าที่กำหนดไว้แล้วคือ

$$\theta_A(\omega) = -D\omega \qquad \omega \in [0, \alpha\pi] \tag{2.37}$$

นำสมการที่ (2.36) แทนลงไปในสมการที่ (2.37)

$$\sum_{k=1}^{N} a_k \sin[\beta(\omega) + k(\omega)] = -\sin[\beta(\omega)] \tag{2.38}$$

เมื่อ $\beta(\omega) = -\frac{1}{2}(-D\omega + N\omega)$ กำหนดเวคเตอร์ 2 เวคเตอร์ดังนี้

$$a = [a_1 \quad a_2 \quad \ldots \quad a_N]' \tag{2.39}$$

$$b(\omega) = \{\sin[\beta(\omega) + \omega] \quad \sin[\beta(\omega) + 2\omega] \quad \ldots \quad \sin[\beta(\omega) + N\omega]\}' \tag{2.40}$$

สามารถเขียนสมการที่ 2.38 ใหม่ได้ดังนี้

$$a'b(\omega) = -\sin(\beta(\omega)) \tag{2.41}$$

และทำการออพติไมซ์โดยวิธีลีสท์สแควร์เพื่อให้ได้ค่าที่ดีที่สุดของสัมประสิทธิ์ a ดังนี้

$$\begin{aligned} J(a) &= \int_0^{\alpha\pi} \left| a'b(\omega) + \sin[\beta(\omega)] \right|^2 d\sin[\beta(\omega) + \omega] d\omega \\ &= a'Qa - 2p'a + C \end{aligned} \tag{2.42}$$

เมื่อเมตริกซ์ $Q$ เวคเตอร์ $p$ และสเกลลาร์ $C$ คือ

$$Q = \int_0^{2\pi} b(\omega) b(\omega)' d\omega \tag{2.43}$$

$$p = -\int_0^{2\pi} b(\omega) \sin[\beta(\omega)] d\omega \tag{2.44}$$

$$C = \int_0^{2\pi} \sin[\beta(\omega)]^2 d\omega \tag{2.45}$$

เพื่อสร้างให้คอมบ์ฟิลเตอร์มีค่าเป็น 0 ที่ความถี่ฮาร์โมนิค $k\omega_0$ จึงต้องเพิ่มข้อจำกัดเหล่านี้ในการออกแบบ

$$a'b(k\omega_0) = -\sin\left[\beta(k\omega_0)\right] \qquad ,k = 1,2,...,M \tag{2.46}$$

เมื่อ $M = \lfloor \pi/\omega_0 \rfloor$ และข้อจำกัดสามารถเขียนอยู่ในรูปแบบ $Ca = f$ เมื่อเมตริกซ์ $C$ และ $f$ คือ

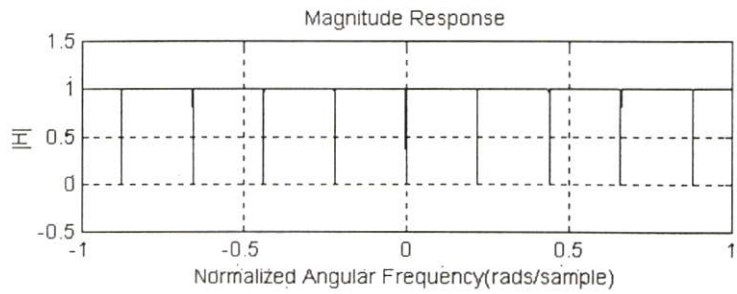
$$C = \left[b(\omega_0),\ \ b(2\omega_0),\ \ ...,\ \ b(N\omega_0)\right]' \tag{2.47}$$

$$f = \left\{-\sin[\beta(\omega_0)],\ \ -\sin[\beta(2\omega_0)],\ \ ...,\ \ -\sin[\beta(N\omega_0)]\right\}' \tag{2.48}$$

ใช้วิธีของลากรางมัลติไพลเออร์จะได้ค่าที่ดีที่สุดของสัมประสิทธิ์คือ

$$a = Q^{-1}p - Q^{-1}C'(CQ^{-1}C')^{-1}(CQ^{-1}p - f) \tag{2.49}$$

และในกรณีที่กำหนดพารามิเตอร์ในการออกแบบคือ $\omega_0 = 0.22\pi, R = 0.999$ และค่าของ $N = 2\pi/\omega_0 = 9$ ซึ่งได้ผลตอบสนองความถี่ดังรูปที่ 2.9 วิธีการออกแบบโดยวิธีนี้จะให้ผลตอบสนองเหมือนกับการออกแบบโดยใช้ FIR ฟิลเตอร์ แต่ให้ออเดอร์ของฟิลเตอร์ที่ต่ำกว่า




**รูปที่ 2.10** แสดงผลตอบสนองความถี่ของดิจิตอลคอมบ์ฟิลเตอร์ออกแบบโดยใช้ออลพาสฟิลเตอร์ ($\omega_0 = 0.22\pi, \rho = 0.999, N = 9$)

## 2.4 การประยุกต์ใช้

ดิจิตอลคอมบ์ฟิลเตอร์มีบทบาทในการนำไปใช้ในงานหลายประเภท การประยุกต์ใช้หลักๆ ของดิจิตอลคอมบ์ฟิลเตอร์จะเกี่ยวกับการกำจัดสัญญาณรบกวนที่เป็นฮาร์โมนิค การประมวลผล

สัญญาณเสียง ซึ่งพบในระบบดิจิตอลวีดีโอดิสก์ เครื่องรับสัญญาณของโทรทัศน์สี และระบบ HDTV หรือใช้ในการแยกสัญญาณบางประเภทที่มีสองความถี่ออกจากกัน ดังได้ยกตัวอย่างโดยสังเขปดังต่อไปนี้

**2.4.1 การประยุกต์ใช้ในการบันทึกสัญญาณเสียง [1]**

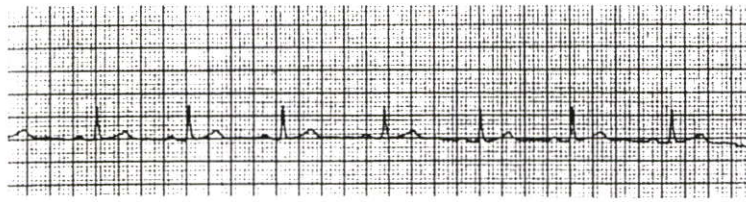
วงจรการประมวลผลสัญญาณที่จำเป็นต้องใช้ในการบันทึกสัญญาณเสียงประกอบไปด้วย

1. คอมเพลสเซอร์ และ ลิมิตเตอร์(Compressors, Limiters)
2. เอ็กซ์แพนเดอร์ และ นอยซ์เกท(Expander, Noise Gates)
3. อีควอลไลเซอร์ และ ฟิลเตอร์(Equalizer, Filters)
4. ระบบการลดสัญญาณรบกวน(Noise Reduction System)
5. ระบบหน่วงและระบบสะท้อน(Delay Systems, Reverberation Systems)
6. วงจรสำหรับเอฟเฟคพิเศษ(Special Effects)

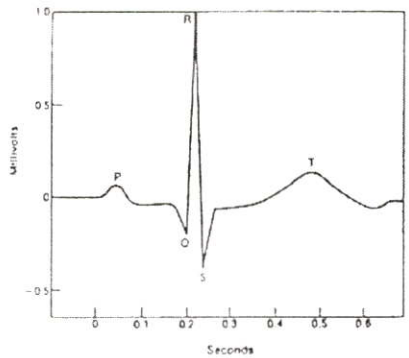
การประยุกต์ใช้คอมบ์ฟิลเตอร์จะเป็นส่วนหนึ่งของระบบฟิลเตอร์ซึ่งเป็นส่วนหนึ่งของระบบการบันทึกเสียงในกรณีที่ต้องการลดทอนความถี่บางส่วนที่มีช่วงแคบ โดยไม่กระทบกระเทือนส่วนอื่นๆ นอกจากนี้ยังเป็นส่วนหนึ่งในการสร้างเอฟเฟคพิเศษในการเปลี่ยนแปลงเฟส ณ ตำแหน่งของฮาร์โมนิคที่ต้องการดังจะกล่าวรายละเอียดในหัวข้อที่ 5.3

**2.4.2.การกำจัดสัญญาณฮาร์โมนิคในการวัดสัญญาณ อิเล็คโทรคาร์ดิโอแกรม [1],[4]**

คือการนำดิจิตอลคอมบ์ฟิลเตอร์เพื่อกำจัดสัญญาณรบกวนที่เป็นฮาร์โมนิค อันเกิดจากการรบกวนของสัญญาณไฟเลี้ยง โดยความถี่ขึ้นอยู่กับระบบไฟฟ้าของประเทศนั้นๆ ปัญหาดังกล่าวพบมากในการบันทึกสัญญาณของเครื่องมือทางการแพทย์ เช่น การบันทึกสัญญาณอิเล็คโทรคาร์ดิโอแกรม หรือเครื่องมือที่ไม่มีการป้องกันสัญญาณรบกวนที่ดีพอ



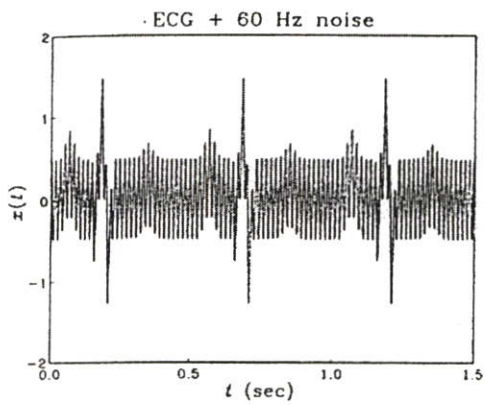
**รูปที่ 2.11** แสดงสัญญาณอิเล็คโทรคาร์ดิโอแกรม



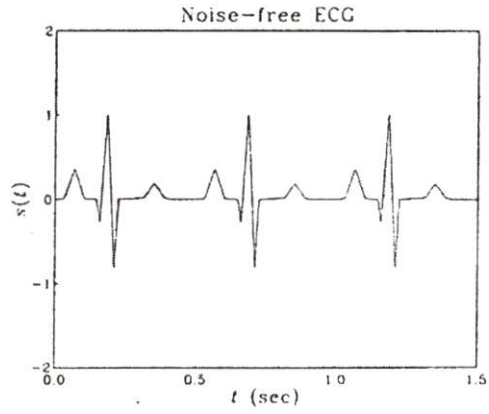

**รูปที่ 2.12** แสดงสัญญาณอิเล็คโทรคาร์ดิโอแกรมภายใน 1 คาบเวลา

จากรูปที่ 2.11 แสดงสัญญาณอิเล็คโทรคาร์ดิโอแกรม ซึ่งเป็นสัญญาณไฟฟ้าที่เกิดจากการวัดจังหวะการสูบฉีดโลหิตของหัวใจ โดยสัญญาณนี้มีลักษณะเป็นคาบ โดยสัญญาณใน 1 คาบแสดงดังรูปที่ 2.12 แต่ละส่วนของกราฟสัญญาณอิเล็คโทรคาร์ดิโอแกรม แสดงถึงขั้นตอนการทำงานของหัวใจ และจากกราฟดังกล่าวจึงแสดงสภาวะทางกายภาพของหัวใจดังเช่น ขนาดและเวลาของ p และ QRS แสดงถึงสภาวะของมวลของกล้ามเนื้อคาร์ดิแอคถ้าขนาดมีค่าน้อยหรือหายไปแสดงถึงความเสียหายของกล้ามเนื้อ หรือขนาดที่เพิ่มขึ้นแสดงถึงอัตราการเต้นของหัวใจที่ผิดปกติ เป็นต้น ซึ่งแพทย์จะรักษาอาการผิดปกติเหล่านี้โดยยาต่างๆ และตรวจสอบสัญญาณอิเล็คโทรคาร์ดิโอแกรมภายหลังคนไข้ได้รับการรักษาโดยยานั้นเพื่อตรวจสอบถึงประสิทธิภาพของยาและการรักษา

ในทางปฏิบัติแล้วมีสัญญาณรบกวนจากภายนอกที่แทรกเข้ามาในสัญญาณอิเล็คโทรคาร์ดิโอแกรม ซึ่งถ้าไม่กำจัดสัญญาณเหล่านี้ออกไปแล้ว ก็ไม่สามารถอ่านสัญญาณที่แท้จริงได้ แหล่งสัญญาณรบกวนที่พบประจำก็คือสัญญาณที่เกิดจากสัญญาณไฟเลี้ยง ซึ่งรบกวนเข้ามาโดยการคลัปปลิ้งของประจุไฟฟ้า หรือการเหนี่ยวนำของสนามแม่เหล็ก รูปที่ 2.13 แสดงสัญญาณอิเล็คโทรคาร์ดิโอแกรมที่ถูกรบกวนโดยสัญญาณไฟเลี้ยง และรูปที่ 2.14 แสดงสัญญาณอิเล็คโทรคาร์ดิโอแกรม ภายหลังการกรองโดยใช้ดิจิตอลคอมบ์ฟิลเตอร์ซึ่งมีอัตราการลดทอนที่ความถี่ไฟเลี้ยงสูงมาก




**รูปที่ 2.13** แสดงสัญญาณอิเล็คโทรคาร์ดิโอแกรม ที่ถูกรบกวนโดยสัญญาณความถี่ไฟเลี้ยง




**รูปที่ 2.14** แสดงสัญญาณอิเล็คโทรคาร์ดิโอแกรม ที่ผ่านการกรองเพื่อกำจัดสัญญาณรบกวน

48388

# บทที่ 3
# การหาโมเมนต์ของราก

## 3.1. บทนำ

โมเมนต์ของราก[5]คือทฤษฎีที่เกี่ยวกับการจัดรูปแบบของสัญญาณซึ่งขึ้นอยู่กับคุณสมบัติทางพีชคณิตของสัญญาณ โดยใช้การวิเคราะห์โพลิโนเมียลในโดเมนเชิงซ้อน สามารถนำไปแก้ไขปัญหาสำคัญหลายประการในงานประมวลผลสัญญาณ[6]

พิจารณาสัญญาณ $\{x(k)\}$ ที่อยู่ในลักษณะของลำดับสัญญาณที่ไม่ต่อเนื่อง(Signal Sequence) เมื่อ $k = 0,1,..,n$ หรือสัญญาณเรียลไทม์ที่ไม่ทราบค่าจำนวน $n+1$แซมเปิ้ล สามารถพิจารณาโดยใช้การแปลง $z$ ที่ค่าคงที่ $n$ ดังนี้

$$X(z) = x(0)z^n + x(1)z^{n-1} + x(2)z^{n-2} + ... + x(n) \tag{3.1}$$

ทำการนอร์มอลไลซ์โดยเทียบกับ $x(0)$ ได้ดังสมการต่อไปนี้

$$f(z) = z^n + p_1 z^{n-1} + p_2 z^{n-2} + ... + p_n \tag{3.2}$$

เมื่อ
$$f(z) = X(z)/x(0) \tag{3.3}$$
$$p_i = x(i)/x(0), i = 1,...,n \tag{3.4}$$

การนอร์มอลไลซ์นี้ไม่ใช่ส่วนที่สำคัญของทฤษฎีแต่ทำให้สามารถจัดการทางพีชคณิตบางส่วนได้ง่ายขึ้น และสามารถเขียน $f(z)$ ได้ดังนี้

$$f(z) = \prod_{i=1}^{n} (z - r_i) \tag{3.5}$$

เมื่อ $r_i$ คือรากของโพลิโนเมียล $f(z)$
และจะใช้สัญลักษณ์ต่อไปนี้

1. $r_j = r_{jin}$ ถ้าราก $r_j$ เป็นรากที่อยู่ภายในวงกลม 1 หน่วย
2. $r_j = r_{jout}$ ถ้าราก $r_j$ เป็นรากที่อยู่ภายนอกวงกลม 1 หน่วย
3. $r_j = r_{jo}$ ถ้าราก $r_j$ เป็นรากที่อยู่บนในวงกลม 1 หน่วย

เขียน $f(z)$ ในรูปแบบความสัมพันธ์ของ $r_{jin}, r_{jout}$ และ $r_{jo}$ ได้ดังนี้

$$f(z) = \prod_{i=1}^{n}(z - r_i)$$
$$= \left[\prod_{j}(z - r_{jin})\right]\left[\prod_{j}(z - r_{jout})\right]\left[\prod_{j}(z - r_{jo})\right] \tag{3.6}$$

หรือ $$f(z) = f_{\min}(z) f_{\max}(z) f_o(z) \tag{3.7}$$

เมื่อ $f_{\min}(z)$ คือตัวประกอบที่มีเฟสน้อยที่สุดของ $f(z)$ $f_{\max}(z)$ คือตัวประกอบที่มีเฟสมากที่สุดของ $f(z)$ และ $f_o(z)$ คือตัวประกอบที่มีรากอยู่บนวงกลม 1 หน่วย จากสมการที่ (3.2) นิวตันได้กำหนดเซตของพารามิเตอร์ดังต่อไปนี้

$$S_m = r_1^m + r_2^m + \ldots + r_n^m = \sum_{i=1}^{n} r_i^m \tag{3.8}$$

เมื่อ $r_i$ คือรากที่ $i$ ตามสมการที่ 3.2

ถึงแม้ว่าจากสมการที่ (3.8) ประกอบไปด้วยรากจากสมการที่ (3.2) แต่ก็ไม่จำเป็นต้องคำนวณ $S_m$ โดยตรง สามารถคำนวณ $S_m$ ได้โดยใช้สัมประสิทธิ์ $p_i$ ซึ่ง $S_m$ ก็คือโมเมนต์ของรากของโพลิโนเมียล $f(z)$ $S_m$ ได้ถูกนำไปใช้งานในการประมวลผลสัญญาณหลายประเภท โดยเฉพาะที่สำคัญคือการหาอนุพันธ์ของเซพสตรัม[3,4]

## 3.2 การประมาณค่าโมเมนต์ของรากโดยกระบวนการวนซ้ำ

เมื่อเขียนโพลิโนเมียลตามสมการที่ 3.2 ในรูปแบบของการคูณกันของตัวประกอบแล้ว สามารถเขียน $f'(z)$ ได้ดังนี้

$$f'(z) = \sum_{i=1}^{n} \frac{f(z)}{z - r_i} \tag{3.9}$$

และเมื่อ $f(r_i) = 0$ จะได้

$$f'(z) = nz^{n-1} + (S_1 + np_1)z^{n-2} + \left(S_2 + p_1 S_1 + np_2\right)z^{n-3}$$
$$+ (S_m + p_1 S_{m-1} + p_2 S_{m-2} + \ldots + np_m)z^{n-m-1} + \ldots \tag{3.10}$$

ทำการหาอนุพันธ์ของสมการที่ 3.2 ได้ดังนี้

$$f'(z) = nz^{n-1} + (n-1)p_1 z^{n-2} + (n-2)p_2 z^{n-3} + \ldots + (n-m)p_m z^{n-m-1} + \ldots \tag{3.11}$$

จากสมการที่ 3.10 และ 3.11 จะได้สมการความสัมพันธ์ของนิวตันไอเดนทิตี้ดังนี้

$$S_1 + np_1 = (n-1)p_1 \quad \text{หรือ} \quad S_1 + p_1 = 0 \tag{3.12}$$

$$S_2 + p_1 S_1 + np_2 = (n-2)p_2 \quad \text{หรือ} \quad S_2 + p_1 S_1 + 2p_2 = 0 \tag{3.13}$$

หรือในรูปแบบทั่วไปคือ

$$S_m + p_1 S_{m-1} + p_2 S_{m-2} + \ldots + np_m = (n-m)p_m \text{ หรือ} \tag{3.14}$$

$$S_m + p_1 S_{m-1} + p_2 S_{m-2} + \ldots + mp_m = 0 \tag{3.15}$$

ถ้าสัญญาณมีความยาวเป็นอินฟินิตี้ สมการที่ (3.15) จะถูกใช้ซ้ำๆ เพื่อคำนวณค่าของโมเมนต์ของราก หรืออีกกรณีหนึ่งถ้าสัญญาณนั้นมีความยาวจำกัด ซึ่งกรณีนี้ที่ค่าของ $m > n$ จะได้ว่า

$$S_m + p_1 S_{m-1} + p_2 S_{m-2} + \ldots + p_n S_{m-n} = 0 \tag{3.16}$$

ความสัมพันธ์ข้างต้นสามารถใช้คำนวณ $S_m$ สำหรับ $m < 0$ โดยแทนค่า $m = n-1, n-2, n-3$ ให้สังเกตว่าค่าของ $S_m$ ไม่ว่า m จะเป็นบวกหรือลบจะสามารถคำนวณได้จากสัมประสิทธิ์ในสมการที่ (3.2)

กระบวนการวนซ้ำตามสมการที่ (3.12) ถึง สมการที่ (3.15) โดย n มีค่าไม่จำกัดจะเป็นอะไรก็ได้ มันสามารถนำไปประยุกต์ใช้กับสัญญาณที่มีความยาวไม่จำกัด ถ้าสามารถสมมติให้สัมประสิทธิ์ $\{p_i\}$ คือสัญญาณที่สุ่มขึ้นมา (นอร์มอลไลซ์แล้ว) และเป็นการเพียงพอที่จะพิสูจน์ว่าสัญญาณที่มีความยาวจำกัดและไม่จำกัดเมื่อแปลงให้อยู่ในรูปของฟังก์ชั่นยกกำลังแล้ว (โดยการแปลง $z$) สามารถเขียนให้อยู่ในรูปความสัมพันธ์ของโพลิโนเมียลและนิวตันไอเดนติตี้ได้

**3.2.1 ข้อสังเกตและผลลัพธ์**

พบว่าสมการที่ (3.12) ถึง สมการที่ (3.15) ก็คือการแปลงสัมประสิทธิ์ $\{p_r\}$ ไปยังเซตของพารามิเตอร์ $\{S_m\}$ ซึ่งเป็นการแปลงแบบ 1 ต่อ 1 และให้ผลลัพธ์ดังนี้

ผลข้อที่ 1 เมื่อมีเซตของสัมประสิทธิ์ $\{p_r\}$ ของโพลิโนเมียล $n$ ตามสมการที่ (3.2) ซึ่งมีค่าราก $\{r_i\}$ $i=1,2,...,n$ ย่อมมีเซตของพารามิเตอร์ $m=1,...,n$ $S_0=n$ ดังนี้

$$S_m=\sum_{i=1}^{n} r_i^m \tag{3.17}$$

ซึ่งแสดงดังสมการที่ (3.12) ถึงสมการที่ (3.15)

ผลข้อที่ 2 ในทางตรงกันข้าม เมื่อมีเซตของโมเมนต์ของราก $\{S_m\}$ ย่อมมีเซตของสัมประสิทธิ์ $\{p_i\}r=1,...,n$ ดังโพลิโนเมียลในสมการ (3.2) ซึ่งสามารถคำนวณย้อนกลับโดยใช้สมการที่ (3.12) ถึงสมการที่ (3.15)

จะใช้โมเมนต์ของรากเพื่ออธิบายถึงคุณสมบัติของโพลิโนเมียลที่ต้องการ ซึ่งคุณสมบัติของพารามิเตอร์เหล่านี้ให้ข้อมูลที่ควรพิจารณาตามทฤษฎีต่อไปนี้

ทฤษฎีที่ 1 ถ้า $r_1,r_2,r_{3,...,}$ คือรากของโพลิโนเมียลและ $\{S_m\}$ ได้จากสมการที่ (3.12) ถึงสมการที่ (3.15)

และจะพบพฤติกรรมต่อไปนี้

1. ถ้า $r_{\max}$ เป็นรากที่เป็นจำนวนจริงและมีขนาดใหญ่ที่สุดใน $\{r_i\}$, $i=1,2,...,n$ แล้ว

$$r_{\max}=\lim_{m\to\infty}\left\{\frac{S_{m+1}}{S_m}\right\} \tag{3.18}$$

2. ถ้า $r_i,r_j$ เป็นค่ารากที่มีขนาดที่ใหญ่ที่สุดใน $\{r_i\}$ $i=1,2,...,n$ แล้ว

$$r_ir_j=\lim_{m\to\infty}\left\{\frac{S_mS_{m+2}-S_{m+1}^2}{S_{m-1}S_{m+1}-S_m^2}\right\} \tag{3.19}$$

$$r_i+r_j=\lim_{m\to\infty}\left\{\frac{S_mS_{m+3}-S_{m+1}S_{m+2}}{S_mS_{m+2}-S_{m+1}^2}\right\} \tag{3.20}$$

**พิสูจน์** ในข้อ 1 แสดงให้เห็นอย่างเด่นชัดว่ารากที่ใหญ่ที่สุดมีผลอย่างมากต่อโมเมนต์ของรากและสำหรับส่วนที่ 2 จะพิสูจน์ดังต่อไปนี้

$$S_m S_{m+2} = \left[ r_i^m + r_j^m + \sum_{\substack{p \\ p\neq i \\ p\neq j}} r_p^m r_p^2 \right] \times \left[ r_i^m r_i^2 + r_j^m r_j^2 + \sum_{\substack{p \\ p\neq i \\ p\neq j}} r_p^m r_p^2 \right] \tag{3.21}$$

$$= r_i^{2m} r_i^2 + r_j^{2m} r_j^2 + r_i^{m+2} r_j^m + r_i^m r_j^{m+2} + r_i^m \sum_{\substack{p \\ p\neq i \\ p\neq j}} r_p^m r_p^2 + r_j^m \sum_{\substack{p \\ p\neq i \\ p\neq j}} r_p^m r_p^2$$
$$+ \sum_{\substack{p \\ p\neq i \\ p\neq j}} r_p^m \sum_{\substack{p \\ p\neq i \\ p\neq j}} r_p^m r_p^2 + r_i^{m+2} \sum_{\substack{p \\ p\neq i \\ p\neq j}} r_p^m + r_j^{m+2} \sum_{\substack{p \\ p\neq i \\ p\neq j}} r_p^m$$

ยิ่งไปกว่านั้นสามารถเขียน

$$\left(S_{m+1}\right)^2 = \left[ r_i^{m+1} + r_j^{m+1} + \sum_{\substack{p \\ p\neq i \\ p\neq j}} r_p^{m+1} \right]^2 \tag{3.22}$$

$$= r_i^{2m+2} + r_j^{2m+2} + \left[ \sum_{\substack{p \\ p\neq i \\ p\neq j}} r_p^{m+1} \right]^2$$
$$+ 2 r_i^{m+1} r_j^{m+1}$$
$$+ 2 r_i^{m+1} \sum_{\substack{p \\ p\neq i \\ p\neq j}} r_p^{m+1} + 2 r_j^{m+1} \sum_{\substack{p \\ p\neq i \\ p\neq j}} r_p^{m+1}$$

ซึ่งทำให้ได้

$$\lim_{m\to+\infty} \left( \frac{\dfrac{S_m S_{m+2} - \left(S_{m+1}\right)^2}{r_i^{m+1} r_j^{m+1}}}{\dfrac{S_{m-1} S_{m+1} - \left(S_m\right)^2}{r_i^{m+1} r_j^{m+1}}} \right) = \frac{\dfrac{r_i}{r_j} + \dfrac{r_j}{r_i} - 2}{\dfrac{1}{r_i^2} + \dfrac{1}{r_j^2} - \dfrac{2}{r_i r_j}} \tag{3.23}$$

และลิมิตที่ 2 ในสมการที่ (3.20) สามารถพิสูจน์ได้โดยใช้วิธีเดียวกัน

ถ้ารากที่ใหญ่ที่สุดเป็นจำนวนเชิงซ้อนแต่โพลิโนเมียลเป็นจำนวนจริงแล้ว $r_i = \rho e^{j\phi}$ และ $r_j = \rho e^{-j\phi}$ ลิมิตจะเป็น $\rho^2$ และ $2\rho\cos\phi$ ตามลำดับ

ทฤษฎีที่ 2 ถ้า $m$ ในทฤษฎีที่ 1 เข้าสู่ $-\infty$ แล้ว ลิมิตเดียวกันจะใช้กับรากที่มีโมดูลี่น้อยที่สุด

## โมเมนต์ของรากและการรวมสัญญาณ

**3.3.1 การคูณ** สมมติให้โมเมนต์ของรากของโพลิโนเมียล $f_1(z)$ คือ $S_m^{f_1(z)}$ และโมเมนต์ของรากของ $f_2(z)$ คือ $S_m^{f_2(z)}$ ฉะนั้นโมเมนต์ของรากของ $f(z) = f_1(z)f_2(z)$ จะเป็น $S_m^{f(z)} = S_m^{f_1(z)} + S_m^{f_2(z)}$

**3.3.2 การบวก** ต้องการที่จะหาโมเมนต์ของรากของ $f(z) = f_1(z) + f_2(z)$ ในเทอมของโมเมนต์ของรากของ $f_1(z)$ และ $f_2(z)$ ฉะนั้นจึงกำหนดฟังก์ชั่น $g_1(z)$ และ $g_2(z)$ ซึ่งไม่ใช่ฟังก์ชั่นโพลิโนเมียลซึ่ง $f_1(z) = \ln g_1(z)$ และ $f_2(z) = \ln g_2(z)$ ดังนั้น

$$f(z) = f_1(z) + f_2(z) = \ln g_1(z)g_2(z) = \ln g(z) \tag{3.24}$$

เมื่อ $g(z) = g_1(z)g_2(z)$

ในกรณีของ $f(z) = \ln g(z)$ ฉะนั้น $f'(z) = g'(z)/g(z)$ หรือ $f'(z)g(z) = g'(z)$

และถ้า $f(z) = f_0 + f_1 z + f_2 z^2 + \ldots$ และ $g(z) = g_0 + g_1 z + g_2 z^2 + \ldots$ ฉะนั้น $f'(z) = f_1 + 2f_2 z + 3f_3 z^2 + \ldots$ และ $g'(z) = g_1 + 2g_2 z + 3g_3 z^2 + \ldots$

เพราะฉะนั้นเขียนความสัมพันธ์ $f'(z)g(z) = g'(z)$ ได้ดังนี้

$$\begin{aligned}&\left(f_1 + 2f_2 z + 3f_3 z^2 + \ldots\right)\left(g_0 + g_1 z + g_2 z^2 + ..\right)\\ &= \left(g_1 + 2g_2 z + 3g_3 z^2 + \ldots\right)\end{aligned} \tag{3.25}$$

เทอมทางด้านซ้ายมืออยู่ในรูปแบบของคอนโวลูชั่น จำนวน n เทอมดังนี้

$$\begin{aligned}&g_0(n+1)f_{n+1} + g_1 n f_n + g_2(n-1)f_{n-1} + \ldots + g_n f_1\\ &= (n+1)g_{n+1}\end{aligned} \tag{3.26}$$

เมื่อกระจายฟังก์ชัน $g(z)$ จะแสดงให้เห็นความสัมพันธ์ของ $f(z)$ และสัมประสิทธิ์ดังนี้

$$g_{n+1} = g_0 f_{n+1} + \frac{n}{n+1} g_1 f_n + \frac{n-1}{n+1} g_2 f_{n-1} + \ldots + \frac{1}{n+1} g_n f_1 \tag{3.27}$$

เมื่อทราบค่า $f(z)$ ก็จะสามารถหาค่า $g(z)$ โดยที่ $f(z) = \ln g(z)$ ยิ่งไปกว่านั้นในกรณีตรงกันข้ามถ้ากำหนด $g(z)$ ก็สามารถหาค่า $f(z)$ ได้ดังนี้

$$f_{n+1} = \frac{1}{g_0}\left[ g_{n+1} - \left( \frac{n}{n+1} g_1 f_n + \frac{n-1}{n+1} g_2 f_{n-1} + \ldots + \frac{1}{n+1} g_n f_1 \right) \right] \tag{3.28}$$

$g_0 \neq 0$ , $g_0 > 0$

โดยค่าเริ่มต้นคือ $f_0 = \ln g_0$

ในการบวกของสัญญาณ 2 สัญญาณสามารถใช้การแปลงดังกล่าวทั้งรูปแบบโดยตรงและย้อนกลับเพื่อคำนวณหาผลลัพธ์ได้

**3.3.3 การหาร** โมเมนต์ของรากของ $f(z) = f_1(z) / f_2(z)$ คือ $S_m^{f(z)} = S_m^{f_1(z)} - S_m^{f_2(z)}$

**3.3.4 การลบ** การลบจะใช้หลักการเดียวกับการบวก

ดังนั้นสามารถใช้วิธีของโมเมนต์ของรากในการจัดการกับสัญญาณในโดเมนใหม่ที่แปลงมาได้

## 3.4 การประยุกต์ใช้

ในส่วนนี้จะแสดงการนำโมเมนต์ของรากไปประยุกต์ใช้ดังจะกล่าวดังต่อไปนี้

**3.4.1 การหาค่าตัวประกอบเฟสน้อยที่สุด และมากที่สุดของสัญญาณโดยไม่ต้องหาค่ารากของสัญญาณ [2]**

การหาค่าตัวประกอบเป็นปัญหาที่พบในงานหลายประเภท ดังเช่น ในงานการออกแบบดิจิตอลฟิลเตอร์ [6], การประมาณค่า [7-12] , การประมวลผลสัญญาณแบบอแดพทีฟ [13] และในงานประเภทอื่นๆ แต่สำหรับปัญหาในที่นี้ก็คือความจำเป็นที่จะหาตัวประกอบเฟสที่น้อยที่สุด $f_{\min}(z)$ ตัวประกอบเฟสที่มากที่สุด $f_{\max}(z)$ จากโพลิโนเมียล $f(z)$ โดยที่

$$f(z) = f_{\min}(z) f_{\max}(z) f_0(z) \tag{3.29}$$

เมื่อ $f_0(z)$ คือตัวประกอบที่อยู่บนวงกลมหนึ่งหน่วย

สามารถเขียน

$$\frac{f'(z)}{f(z)} = \frac{\left(f_{\min}(z)\right)'}{f_{\min}(z)} + \frac{\left(f_{\max}(z)\right)'}{f_{\max}(z)} + \frac{\left(f_0(z)\right)'}{f_0(z)} \tag{3.30}$$

และเมื่อ $m \geq 0$ และ $C : |z| = 1$ แล้ว

$$\oint_C \frac{f'(z)}{f(z)} z^m dz = \oint_C \frac{\left(f_{\min}(z)\right)'}{f_{\min}(z)} z^m dz \tag{3.31}$$

$$+ \oint_C \frac{\left(f_0(z)\right)'}{f_0(z)} z^m dz$$

เนื่องจากโพลของตัวปะกอบ $f_{\max}(z)$ ไม่มีเศษฉะนั้น

$$\frac{1}{2\pi j} \oint_C \frac{f'(z)}{f(z)} z^m dz = S_m^{f_{\min}(z)} + \frac{1}{2} S_m^{f_0(z)} \tag{3.32}$$

ถ้าไม่มีรากบนวงกลม 1 หน่วยแล้ว $f_0(z) = 1$ ดังนั้น

$$S_m^{f_{\min}(z)} = \frac{1}{2\pi j} \oint_C \frac{f'(z)}{f(z)} z^m dz \tag{3.33}$$

จากผลลัพธ์แล้วสามารถหาโพลิโนเมียล $f_{\min}(z)$ ได้โดยตรงโดยใช้ความรู้ที่เกี่ยวกับโมเมนต์ของราก

ถ้า $f(z)$ มีซีโร่อยู่บนวงกลม 1 หน่วยจะใช้วิธีใดวิธีหนึ่งดังกล่าวต่อไปนี้ โดยที่ $f(z)$ เป็นโพลิโนเมียลที่เป็นจำนวนจริง ดังนั้นมันจะมีซีโร่ที่เป็นจำนวนเชิงซ้อนที่เป็นคู่คอนจุเกทซึ่งกันและกัน

**อัลกอริทึม 1**

1. จากสมการ

$$S_m^{f_{\min}(z)} + \frac{1}{2} S_m^{f_0(z)} = \frac{1}{2\pi j} \oint_C \frac{f'z)}{f(z)} z^m dz \tag{3.34}$$

2. กำหนดให้

$$\tilde{S}_m = 2\left[ S_m^{f_{\min}(z)} + \frac{1}{2} S_m^{f_0} \right] = 2S_m^{f_{\min}} + S_m^{f_0} \tag{3.35}$$

ซึ่งทำให้เกิดโพลิโนเมียล $\tilde{f}(z)$ ที่มีดีกรีดังนี้

$$p = 2S_0^{f_{\min}(z)} + S_0^{f_0(z)} \tag{3.36}$$

จากผลลัพธ์พบว่า $\tilde{f}(z)$ จะสามารถสร้างขึ้นได้จากโมเมนต์ของราก $\tilde{S}_m$ ในรูปของตัวประกอบ $Az^q$ เมื่อ $A$ คือค่าคงที่ และ $q$ คือจำนวนเต็ม โพลิโนเมียล $\tilde{f}(z)$ จะมีซีโร่ของ $f_{\min}(z)$ ภายใน $C:|z|=1$ โดยที่แต่ละตัวคูณด้วย 2 และซีโร่ของ $f(z)$ จะอยู่บนเส้นรอบวงของ $C:|z|=1$

3. สร้างโพลิโนเมียล

$$f_1(z) = z^p \tilde{f}(z^{-1}) \tag{3.37}$$

ซึ่งเป็นโพลิโนเมียลที่ไม่มีซีโร่อยู่ในวงกลม 1 หน่วย และเฉพาะซีโร่ของ $f(z)$ จะอยู่บนวงกลม 1 หน่วย ฉะนั้นพารามิเตอร์

$$S_m{}' = \frac{1}{2\pi j} \oint_C \frac{f_1'(z)}{f_1(z)} z^m dz \tag{3.38}$$

จะสามารถคำนวณได้ และเป็น $\frac{1}{2} S_m^{f_0}$ ของโพลิโนเมียลเดิม

4. นำผลลัพธ์จากข้อ 3 ไปลบจากผลลัพธ์ในข้อ 1 จะได้โมเมนต์ของรากของตัวประกอบเฟสที่น้อยที่สุด และสามารถคำนวณหา $f_{\min}(z)$
5. จากพารามิเตอร์ $\frac{1}{2} S_m^{f_0}$ สร้างโพลิโนเมียล $f_0(z)$
6. นำโมเมนต์ของรากของ $f_{\min}(z)$, $f_0(z)$ ไปลบออกจากโมเมนต์ของรากของ $f(z)$ จะได้โมเมนต์ของรากของส่วนประกอบเฟสที่มากที่สุด $f_{\max}(z)$

จากขั้นตอนที่ 6 สามารถเปลี่ยนแปลงได้โดยสามารถคำนวณได้อีกวิธีหนึ่งคือ

จากข้อ 6 สร้าง $g(z)$ จากโพลิโนเมียล $f(z)$ เดิมคือ

$$z^n f\left(z^{-1}\right)^{\cdot} = g(z) \tag{3.39}$$

ทำซ้ำขั้นตอนที่ 1 จะได้ดังนี้

$$S_m^{g_{\min}(z)} + \frac{1}{2} S_m^{g_0(z)} \tag{3.40}$$

อย่างไรก็ตาม $S_m^{g_0(z)} = S_m^{f_0(z)}$

และเมื่อพิจารณาขั้นตอนที่ 3 สามารถหา $S_m^{g_{\min}(z)}$ ฉะนั้นสามารถสร้างโพลิโนเมียล $g_{\min}(z)$ ซึ่งซีโร่ของมันจะเป็นอินเวอร์สของ $f(z)$ ซึ่งอยู่ภายนอก $C:|z|=1$ ซึ่งเป็นอีกวิธีหนึ่งในการคำนวณ $f_{\max}(z)$

**อัลกอริทึม 2**

กำหนดให้ออเดอร์ของ $f(z)$ เป็น $n$ แล้วสร้าง $g(z) = z^n f\left(z^{-1}\right)$ รากของ $g(z)$ จะเป็นค่าส่วนกลับกับรากของ $f(z)$ เพราะฉะนั้นเมื่อมีรากของ $f(z)$ อยู่บนวงกลม 1 หน่วยก็จะมีรากของ $g(z)$ ซึ่งอยู่บนวงกลม 1 หน่วยเช่นกัน และถ้า $f(z)$เป็นจำนวนจริงแล้วตำแหน่งของมันจะอยู่บนตำแหน่งเดียวกัน

ดังนั้น $f(z)$ และ $g(z)$ จึงมีตัวประกอบเหมือนกันถ้าสมมติให้ $f(z)$ ไม่มีค่ารากที่เป็นส่วนกลับซึ่งกันและกันแล้ว $f(z)$ และ $g(z)$ จะมีตัวประกอบร่วมกันคือโพลิโนเมียลที่มีรากอยู่บนวงกลม 1 หน่วย ซึ่งโพลิโนเมียลดังกล่าวสามารถคำนวณโดยใช้อัลกอริทึมของยูคลิด ขั้นตอนต่อไปก็คือการคำนวณหาโมเมนต์ของรากของตัวประกอบร่วมแล้วนำไปลบจากโมเมนต์ของรากที่คำนวณไว้แล้วในสมการที่ 3.33 จะได้ค่าของโพลิโนเมียล $f_0(z)$ และ $f_{\min}(z)$ หลังจากนั้นคำนวณโดยใช้ข้อที่ 6 ของอัลกอริทึมที่ 1

**อัลกอริทึมที่ 3**

ถ้ามีโพลิโนเมียล $f(z)$ ซึ่งมีออเดอร์ $n$ ซึ่ง $n = n_{in} + n_{out} + n_0$

$$f(z) = \left(\prod_{i=1}^{n_{in}} (z - r_{iin})\right)\left(\prod_{i=1}^{n_{out}} (z - r_{iout})\right)\left(\prod_{i=1}^{n_0} (z - r_{i0})\right) \tag{3.41}$$

เมื่อ $|r_{iin}|<1$ ,$|r_{iout}|>1$ และ $|r_{io}|=1$ รากที่อยู่ภายในวงกลม 1 หน่วยเป็นส่วนที่ไม่สนับสนุน $S_m$ ในขณะที่รากที่อยู่ภายนอกวงกลม 1 หน่วยเป็นส่วนสนับสนุน แต่อย่างไรก็ตามรากทั้ง 2 ประเภทมีความสำคัญอย่างมากในกรณีที่ $m$ เป็นบวก และในกรณีตรงกันข้ามถ้า $m$ เป็นลบจะถูกควบคุมโดยรากที่อยู่ภายในวงกลม 1 หน่วย

สำหรับกรณีที่ $m$ เป็นบวก

$$\begin{aligned} S_m &= \sum_{i=1}^{n_{in}} r_{iin}^m + \sum_{i=1}^{n_{out}} r_{iout}^m + \sum_{i=1}^{n_o} r_{io}^m \\ &= S_m^{f_{\min}(z)} + S_m^{f_{\max}(z)} + S_m^{f_o(z)} \end{aligned} \tag{3.42}$$

และเมื่อ $S_m^{f_{\min}(z)} \to 0, \left|S_m^{f_0(z)}\right| \le n_0$ เมื่อ $m \to +\infty$ และ $\left|S_m^{f_{\max}(z)}\right|$ เพิ่มขึ้นตาม $r_{\max}^m$ เมื่อ $r_{\max}^m = \max\{r_{iout}\}$ สามารถเขียน

$$S_m \cong S_m^{f_{\max(z)}} \text{ เมื่อ } m \to \infty \tag{3.43}$$

ในทางปฏิบัติ

$$\begin{aligned} S_m^{f_{\min}(z)} &\le \left|S_m^{f_{\min}(z)}\right| \\ &\le \sum_{i=1}^{n_{in}} \left|r_{iin}\right|^m \left|\exp\left[\, jm\arg\left(r_{iin}\right)\right]\right| \\ &\le \sum_{i=1}^{n_{in}} \left|r_{iin}\right|^m \le \sum_{i=1}^{n_{in}} 1 = n_{in} \\ S_m^{f_0(z)} &\le \left|S_m^{f_0(z)}\right| \\ &\le \sum_{i=1}^{n_0} \left|\exp\left[\, jm\arg\left(r_{io}\right)\right]\right| \le \sum_{i=1}^{n_o} 1 = n_o \end{aligned} \tag{3.44}$$

และ

$$S_m^{f_{\min}(z)} + S_m^{f_o(z)} \le n_{in} + n_o, \forall m \tag{3.45}$$

ถ้าเลือก $m$ ให้มีขนาดใหญ่กว่า $n_{in} + n_o$ จากเซตของ $S_m$ สามารถสร้างโพลิโนเมียล $f_{\max}(z)$ และในกรณีที่คล้ายกัน

$$S_m^{f_{\max}(z)} \to 0 \text{ และ } \left|S_m^{f_o(z)}\right| \le 0 \text{ ขณะที่ } m \to -\infty \tag{3.46}$$

และในทางปฏิบัติ

$$S_m^{f_{\max}(z)} + S_m^{f_o(z)} \le n_{out} + n_o \text{ สำหรับ } m < 0 \tag{3.47}$$

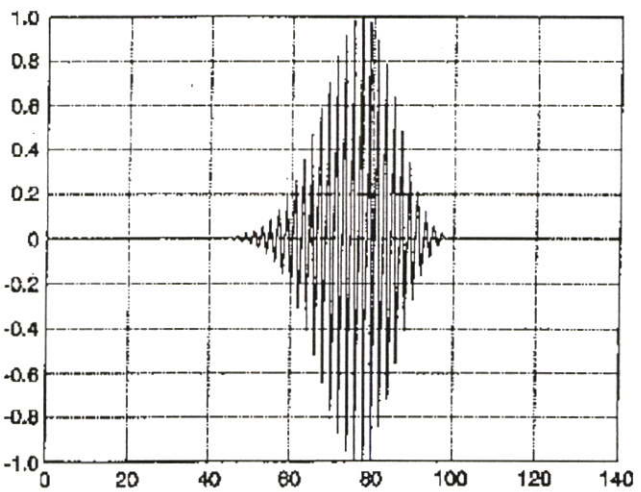
ถ้าเลือกขนาดของ $m$ ให้ใหญ่กว่า $n_{out}+n_o$ จากเซตของ $S_{-m}$ สามารถสร้าง โพลิโนเมียล $f_{\min}(z)$

เมื่อทราบค่า $f_{\min}(z)$ และ $f_{\max}(z)$ สามารถสร้าง $f_o(z)$ จากโมเมนต์ของรากของมันจากสมการต่อไปนี้

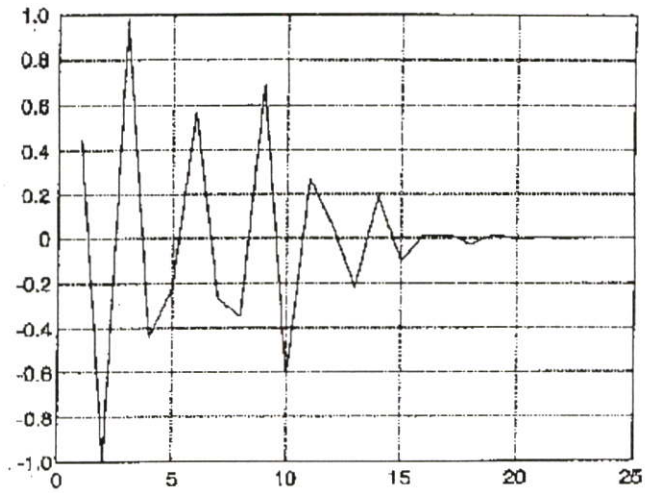
$$S_m^{f_o(z)} = S_m - S_m^{f_{\min}(z)} - S_m^{f_{\max}(z)} \tag{3.48}$$

อัลกอริทึมนี้มีความเหมาะสมจะใช้ในกรณีที่โพลิโนเมียล $f(z)$ ไม่มีซีโร่บนวงกลม 1 หน่วยเนื่องจากในกรณีนี้ในการคำนวณหาจำนวนของซีโร่ในวงกลม 1 หน่วย $n_{in}$ จะเป็นเรื่องง่ายโดยการอินทิเกรต $\oint_{|z|=1} f'(z)/f(z)dz$ แล้วหา $n_{out}=n=n_{in}$ หรืออีกนัยหนึ่งคือคำนวณตามขั้นตอนที่ 1 ถึง 3 ของอัลกอริทึม 1 นั่นเอง

รูปที่ 3.1 แสดงสัญญาณซึ่งไม่มีส่วนประกอบเฟสน้อยที่สุด และรูปที่ 3.2 แสดงส่วนประกอบเฟสมากที่สุด




**รูปที่ 3.1** แสดงส่วนประกอบเฟสน้อยที่สุด




**รูปที่ 3.2** แสดงส่วนประกอบเฟสมากที่สุด

### 3.4.2 การแปลงทรานสเฟอร์ฟังก์ชั่น

การประมาณค่าของการแปลงทรานสเฟอร์ฟังก์ชั่นเป็นส่วนสำคัญของการประยุกต์ใช้ในงานหลายประเภท ซึ่งปัญหาดังกล่าวมีรายละเอียดดังต่อไปนี้ ถ้าสัญญาณซึ่งมีเฟสที่น้อยที่สุดซึ่งผ่านการแปลง $z$ จะได้ $f(z)$ โดยฟังก์ชั่นผกผันของ $f(z)$ คือ $a(z)$ โดยที่

$$f(z)a(z) = z^M \tag{3.49}$$

และ $|f(z)a(z)| = 1$ เมื่อ $|z| = 1$

อย่างไรก็ตามจากความสัมพันธ์ข้างต้นระหว่าง $f(z)$ และ $a(z)$ แสดงถึง

$$S_m^{f(z)} + S_m^{a(z)} = 0 \tag{3.50}$$

ฉะนั้นจึงสามารถคำนวณ $a(z)$ ได้โดยตรงโดยใช้ $S_m^{f(z)}$

แน่นอนว่า เงื่อนไขดังกล่าวไม่สามารถทำได้โดยตรงในทางปฏิบัติ ฉะนั้นจึงสมมติให้ $f(z)$ เป็นโพลิโนเมียลที่มีออเดอร์ $n$ ดังนี้

$$f(z) = z^p + p_1 z^{n-1} + \ldots + p_{n-1} z + p_n \tag{3.51}$$

ถ้า $a(z)$ เป็นโพลิโนเมียลที่มีออเดอร์ $p$ ดังนี้

$$a(z) = z^p + q_1 z^{p-1} + \ldots + q_{p-1} z + q_p \tag{3.52}$$

หลังจากนั้นเมื่อใช้สมการที่ (3.49) จะได้โมเมนต์ของรากของโพลิโนเมียลที่ต้องการ

$$S_m^{f(z)} + S_m^{a(z)} = 0 \Rightarrow S_m^{a(z)} = -S_m^{f(z)}, m = 0, \ldots, p \tag{3.53}$$

และผลคูณจะได้ดังต่อไปนี้

$$\begin{aligned} f(z)a(z) &= z^{n+p} + \tilde{p}_1 z^{n+p-1} + \ldots + \tilde{p}_{n-1} z^{n+1} \\ &+ \tilde{p}_p z^n + \tilde{p}_{n+1} z^{n-1} + \ldots \\ &+ \tilde{p}_{n+p-1} z + \tilde{p}_{n+p} \end{aligned} \tag{3.54}$$

และจากคุณสมบัติการเท่ากันของโมเมนต์ของรากทำให้

$$\tilde{p}_1 = \ldots = \tilde{p}_p = 0 \tag{3.55}$$

ฉะนั้นผลคูณจากสมการที่ (3.48) จะกลายเป็น

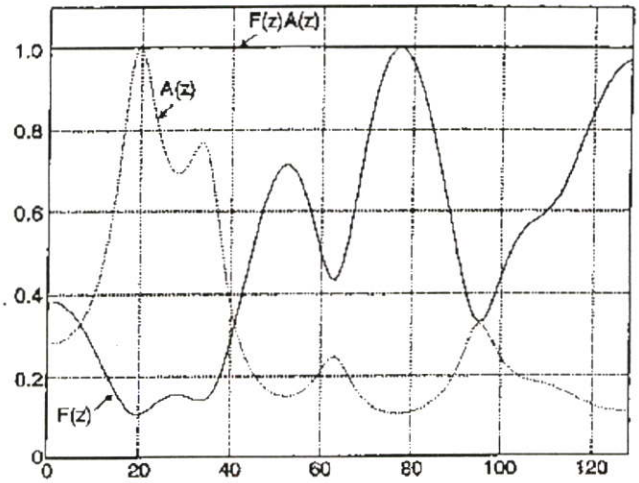
$$f(z)a(z) = z^{n+p} + \tilde{p}_{p+1} z^{n-1} + \ldots + \tilde{p}_{n+p-1} z + \tilde{p}_n + p \tag{3.56}$$

จากความสัมพันธ์ข้างต้นจะได้

$$\begin{aligned} \left| f(z)a(z) \right| &= \left| z^{n+p} + \tilde{p}_{p+1} z^{n-1} + \ldots + \tilde{p}_{n+p-1} z + \tilde{p}_{n+p} \right| \\ &\leq 1 + \left| \tilde{p}_p + 1 \right| + \ldots + \left| \tilde{p}_{n+p} \right| \end{aligned} \tag{3.57}$$

และเนื่องจากเหตุนี้จึงเห็นได้อย่างชัดเจนว่าเมื่อออเดอร์ $p$ ยิ่งมีค่ามาก เงื่อนไขตามสมการที่ (3.48) จะถูกต้องมากขึ้น ซึ่งเป็นการหาค่าของ $a(z)$ ได้โดยตรง

รูปที่ 3.3 แสดงเซตของตัวอย่างของผลลัพธ์ที่ใช้วิธีดังกล่าว เป็นส่วนของสัญญาณเสียงพูด




**รูปที่ 3.3** ผลลัพธ์ของสัญญาณเสียงพูด

### 3.4.3 การตรวจจับการเปลี่ยนแปลงในทันทีของสัญญาณ

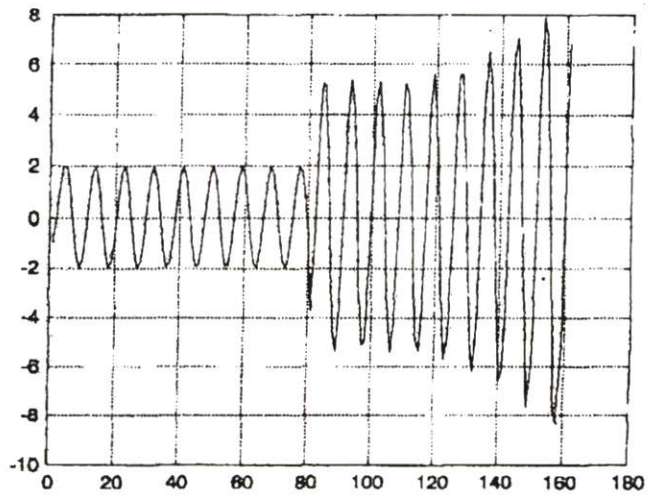
ปัญหาในการตรวจจับการเปลี่ยนแปลงของสัญญาณเป็นปัญหาที่สำคัญประการหนึ่งในการประมวลผลสัญญาณ การเปลี่ยนแปลงดังกล่าวอยู่ในรูปแบบการเปลี่ยนแปลงขนาดของสัญญาณ เช่น เกรเดียร์น หรือ โควาเรียนท์ การตรวจจับสัญญาณรบกวนซึ่งไม่สามารถกระทำได้โดยค่าเฉลี่ยมาตรฐาน ปัญหาในการตรวจจับการเปลี่ยนแปลงอย่างทันทีของสัญญาณนี้เป็นลักษณะพื้นฐานของการจัดรูปแบบสัญญาณ การเปลี่ยนแปลงดังกล่าวทำให้เกิดการจัดรูปแบบใหม่ของสัญญาณ การตรวจจับการเปลี่ยนแปลงดังกล่าวสามารถใช้เทคนิคดังเช่น ซิงกูลาร์วาลูดีคอมโพซิชั่น(Singular Value Decomposition) หรือ ออโตรีเกรสซีฟโมเดลลิง(Auto Regressive Modeling) ซึ่งต้องใช้การหาค่าของเมตริกซ์โควาเรียนท์ การหาค่าตัวประกอบหรือการแปลงค่า ซึ่งไม่สามารถใช้งานในลักษณะเรียลไทม์ ได้

ในทางตรงกันข้าม การใช้วิธีโมเมนต์ของรากดูเหมือนจะตอบสนองต่อการใช้ในลักษณะเรียลไทม์ได้ดีกว่า ถ้าสัญญาณมีการเปลี่ยนแปลงที่ตำแหน่งใดตำแหน่งหนึ่งในเวลา $n-1$ และถ้าค่านอร์มอลไลซ์ของสัญญาณที่ถูกแซมปลิ้งเป็น $\{p_i\}, i=0,...,n-1$ และถ้าให้สัญญาณที่ถูกแซมปลิ้งหลังจากตำแหน่ง $n$ เป็น $\{\tilde{p}_j\}, j=n,...$ จากความสัมพันธ์แบบวนซ้ำ (Newton Identities) ของโมเมนต์ของรากสามารถเขียน

$$\begin{aligned} S_{n-1} + p_1 S_{n-2} + p_2 S_{n-3} + ... + p_{n-2} S_1 + (n-1)p_{n-1} &= 0 \\ \tilde{S}_n + p_1 S_{n-1} + p_2 S_{n-2} + ... + p_{n-1} S_1 + n\tilde{p}_n &= 0 \\ S_n + p_1 S_{n-1} + p_2 S_{n-2} + ... + p_{n-1} S_1 + np_n &= 0 \\ \tilde{S}_n - S_n + n(\tilde{p}_n - p_n) = 0 \Rightarrow \tilde{S}_n - S_n &= -n(\tilde{p}_n - p_n) \end{aligned} \tag{3.58}$$

เมื่อ $\tilde{S}_n$ คือโมเมนต์ของรากที่เปลี่ยนแปลงหลังจากสัญญาณมีการเปลี่ยนแปลงอย่างทันที ทันใด ถ้าไม่มีค่าของสัญญาณที่ถูกแซมปลิ้งข้ามไปอย่างเห็นได้ชัด แล้วปริมาณ $(\tilde{p}_n - p_n)$ จะเป็นค่าที่เล็กมาก และฉะนั้นจะไม่มีการเปลี่ยนแปลงอย่างทันทีทันใดของโมเมนต์ของราก ณ จุดนั้น แต่กระนั้นถ้าการเปลี่ยนแปลงมีอยู่ การเปลี่ยนแปลงในค่าใหม่ของพารามิเตอร์จะเพิ่มขึ้นเรื่อยๆ ในแต่ละวงรอบของการวนซ้ำ ซึ่งโมเมนต์ของรากจะมีการเปลี่ยนแปลงที่สูงมาก จากความจริงที่ว่าการเปลี่ยนแปลง ณ จุดที่แน่นอนแสดงให้เห็นว่าสัญญาณ ณ จุดนั้นจะไม่เป็นเฟสต่ำสุด คือมันจะมีซีโร่นอกวงกลม 1 หน่วย ซึ่งซีโร่เหล่านี้ทำให้เกิดการเพิ่มขึ้นของโมเมนต์ของรากอย่างทันทีทันใด ภายหลังสัญญาณเกิดการเปลี่ยนแปลง

ผลลัพธ์ตามรูปที่ 3.5 จะแสดงให้เห็นค่าของโมเมนต์ของรากเพื่อใช้ในการตรวจจับกรณีของสัญญาณไซน์ ซึ่งมีความถี่นอร์มอลไลซ์ 0.11 และและเปลี่ยนเป็น 0.115 ณ จุดการเปลี่ยนแปลงที่แซมปลิ้งที่ 80 ซึ่งแสดงให้เห็นอย่างชัดเจน ณ ตำแหน่งของความถี่ที่มีการเปลี่ยนแปลง




**รูปที่ 3.4** แสดงโมเมนต์ของรากของสัญญาณ

# บทที่ 4
# การกำจัดฮาร์โมนิคที่ไม่ต้องการของดิจิตอลคอมบ์ฟิลเตอร์

## 4.1. บทนำ

ดิจิตอลคอมบ์ฟิลเตอร์นั้นถูกออกแบบเพื่อใช้ในการกำจัดฮาร์โมนิคที่ไม่ต้องการออกไป และให้เอาท์พุทเป็นความถี่อื่นๆ และให้เอาท์พุทเป็นความถี่อื่นๆที่คงลักษณะเดิม จากลักษณะดังกล่าวได้ถูกนำไปประยุกต์ใช้ในงานหลายประเภท ดังที่กล่าวมาแล้วในบทที่ 2 และนอกจากนี้ในการประยุกต์ใช้ในงานบางประเภทที่มีความต้องการกำจัดความถี่บางฮาร์โมนิคในขณะที่ต้องการรักษาความถี่ฮาร์โมนิคอื่นๆไว้ดังเช่น การประมวลผลสัญญาณบางประเภทที่มีลักษณะโครงสร้างเป็นฮาร์โมนิค อย่างสัญญาณเสียงจากเครื่องดนตรี เอฟเฟคของสัญญาณเสียง หรือในการกรองสัญญาณรบกวนซึ่งเกิดจากระบบบางประเภทที่มีสัญญาณรบกวนบางช่วง ซึ่งเกิดสัญญาณรบกวนเฉพาะฮาร์โมนิคคี่ ดังใน [14] เป็นต้น

ในลำดับต่อไปจะได้กล่าวถึงการกำจัดฮาร์โมนิคที่ไม่ต้องการออกในกรณีที่ $D(D = 2\pi/\omega_0)$ เป็นจำนวนเต็ม และไม่เป็นจำนวนเต็มว่ามีความแตกต่างกันอย่างไร และวิธีการกำจัดฮาร์โมนิคที่ไม่ต้องการของดิจิตอลคอมบ์ฟิลเตอร์โดยวิธีโมเมนต์ของราก

## 4.2. การกำจัดฮาร์โมนิคที่ไม่ต้องการของดิจิตอลคอมบ์ฟิลเตอร์

พิจารณาทรานสเฟอร์ฟังก์ชั่นของดิจิตอลคอมบ์ฟิลเตอร์ในกรณีที่ $D$ เป็นจำนวนเต็มคือ

$$\frac{1-z^{-D}}{1-R^{D}z^{-D}} \tag{4.1}$$

เมื่อ $D = 2\pi/\omega_0$ จะพบว่าจากสมการที่ 4.1 สัมประสิทธิ์ของเทอมเศษและส่วนจะอยู่ในรูปแบบดังนี้

$$\begin{aligned} &\text{สัมประสิทธิ์ของเทอมเศษ} && \overbrace{\begin{bmatrix}1 & 0 & 0 & 0 & \dots & -1\end{bmatrix}}^{D+1} \\ &\text{สัมประสิทธิ์ของเทอมส่วน} && \overbrace{\begin{bmatrix}1 & 0 & 0 & 0 & \dots & -R^{D}\end{bmatrix}}^{D+1} \end{aligned} \tag{4.2}$$

และลำดับต่อไปจะพิจารณาทรานสเฟอร์ฟังก์ชั่นของดิจิตอลคอมบ์ฟิลเตอร์ในกรณีที่ $D$ ไม่เป็นจำนวนเต็มคือ

$$H_c(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + ... + b_N z^{-N}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + ... + a_N z^{-N}} \tag{4.3}$$

ซึ่งได้จากการออกแบบ $z^{-D}$ เมื่อ $D$ ไม่เป็นจำนวนเต็มแล้วแทนค่าลงในสมการที่ 4.1 และสัมประสิทธิ์ของเทอมเศษและเทอมส่วนจะอยู่ในรูปแบบดังนี้

$$\begin{aligned} &\text{สัมประสิทธิ์ของเทอมเศษ} && [b_0 \quad b_1 \quad b_2 \quad ... \quad b_N] \\ &\text{สัมประสิทธิ์ของเทอมส่วน} && [a_0 \quad a_1 \quad a_2 \quad ... \quad a_N] \end{aligned} \tag{4.4}$$

### 4.2.1. การกำจัดฮาร์โมนิคที่ไม่ต้องการของดิจิตอลคอมบ์ฟิลเตอร์เมื่อ $D$ เป็นจำนวนเต็ม

จากสมการที่ (4.1) เมื่อต้องการรักษาสัญญาณที่ฮาร์โมนิคที่ $k$ จะได้ทารนสเฟอร์ฟังก์ชั่นเมื่อต้องการกำจัดฮาร์โมนิคที่ $k$ คือ

$$\begin{aligned} H_c'(z) &= \frac{1 - z^{-D}}{1 - R^D z^{-D}} \times \frac{(1 - Re^{jk\omega_0} z^{-1})(1 - Re^{-jk\omega_0} z^{-1})}{(1 - e^{jk\omega_0} z^{-1})(1 - e^{-jk\omega_0} z^{-1})} \\ &= \frac{1 - z^{-(D-2)}}{1 - R^{D-2} z^{-(D-2)}} \end{aligned} \tag{4.5}$$

หรือจะได้ทรานสเฟอร์ฟังก์ชั่นใหม่ในกรณีที่ $D$ เป็นจำนวนเต็มและต้องการกำจัดฮาร์โมนิคจำนวน $n$ ฮาร์โมนิคคือ

$$\frac{1 - z^{-(D-2n)}}{1 - R^{D-2n} z^{-(D-2n)}} \tag{4.6}$$

จะพบว่าในกรณีที่ $D$ เป็นจำนวนเต็มเมื่อต้องการกำจัดฮาร์โมนิคบางฮาร์โมนิคของดิจิตอลคอมบ์ฟิลเตอร์ออก ค่าสัมประสิทธิ์ของทรานสเฟอร์ฟังก์ชั่นใหม่จะสามารถคำนวณได้อย่างง่ายดายโดยมีการเปลี่ยนแปลงเฉพาะจำนวนพจน์และค่ายกกำลังของ $R$ เท่านั้น

### 4.2.2. การกำจัดฮาร์โมนิคที่ไม่ต้องการของดิจิตอลคอมบ์ฟิลเตอร์เมื่อ $D$ ไม่เป็นจำนวนเต็ม

จากสมการที่ (4.3) เมื่อต้องการกำจัดฮาร์โมนิคที่ $k$ จะได้ทรานสเฟอร์ฟังก์ชั่นของดิจิตอลคอมบ์ฟิลเตอร์ดังนี้

$$H_c'(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_N z^{-N}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_N z^{-N}} \times \frac{(1 - \rho e^{jk\omega_0} z^{-1})(1 - \rho e^{-jk\omega_0} z^{-1})}{(1 - e^{jk\omega_0} z^{-1})(1 - e^{-jk\omega_0} z^{-1})} \tag{4.7}$$

$$H_C'(z) = \frac{b_{0new} + b_{1new} z^{-1} + b_{2new} z^{-2} + \ldots + b_{Nnew} z^{-(N-2)}}{a_{0new} + a_{1new} z^{-1} + a_{2new} z^{-2} + \ldots + a_{Nnew} z^{-(N-2)}} \tag{4.8}$$

ถ้าจำนวนฮาร์โมนิคที่ต้องการกำจัดออกคือ $p$ จะได้ออเดอร์ของทรานเฟอร์ฟังก์ชั่นกลายเป็น $N-2p$

สัมประสิทธิ์ของทั้งเทอมเศษ และเทอมส่วน นั้นสามารถหาได้จากการแยกส่วนประกอบของโพลิโนเมียลแล้วคูณกลับไปใหม่ และในลำดับต่อไปนี้จะกล่าวถึงการใช้วิธีของการหาโมเมนต์ของรากเพื่อคำนวณค่าของสัมประสิทธิ์ดังกล่าว

## 4.3. การกำจัดฮาร์โมนิคที่ต้องการของดิจิตอลคอมบ์ฟิลเตอร์โดยใช้โมเมนต์ของราก [15],[16]

จากขั้นตอนการกำจัดฮาร์โมนิคที่ไม่ต้องการของดิจิตอลคอมบ์ฟิลเตอร์ในหัวข้อที่ 4.2 ที่กล่าวมาแล้วนั้นพบว่าการคำนวณหาค่าสัมประสิทธิ์ของดิจิตอลคอมบ์ฟิลเตอร์ใหม่ภายหลังการกำจัดฮาร์โมนิคที่ k ออกไป จะได้มาจากกระบวนการการคูณขึ้นใหม่ของตัวประกอบของรากซึ่งมีความซับซ้อน โดยเฉพาะในกรณีที่ค่า N มีค่าสูงและค่าของ k มีค่าต่ำ และในกรณีของการกำจัดฮาร์โมนิคที่ k ใหม่กระทำกับดิจิตอลคอมบ์ฟิลเตอร์เดิมก็ต้องทำการคูณตัวประกอบของรากขึ้นใหม่ทุกครั้ง ฉะนั้นจึงได้นำวิธีการของการหาโมเมนต์ของรากมาประยุกต์ใช้ในกรณีดังกล่าว เนื่องจากวิธีการหาโมเมนต์ของรากแสดงถึงความสัมพันธ์ของรากและสัมประสิทธิ์ในรูปแบบการคำนวณที่การได้มาซึ่งสัมประสิทธิ์ไม่ต้องคูณตัวประกอบของราก และเมื่อนำมาประยุกต์ใช้กับดิจิตอลคอมบ์ฟิลเตอร์แล้วพบว่าทำให้ขั้นตอนการหาสัมประสิทธิ์มีความเหมาะสมที่จะนำไปประยุกต์ใช้กับตัวประมวลผลสัญญาณดิจิตอลเนื่องจากเปลี่ยนรูปการคำนวณให้อยู่ในรูปแบบการคูณ การบวก และการลบ โดยเฉพาะอย่างยิ่งในกรณีของการกำจัดฮาร์โมนิคอื่นเพิ่มเติมจากที่เคยกำจัดไว้แล้ว เราสามารถนำโมเมนต์ของรากของฮาร์โมนิคที่ต้องการกำจัดเพิ่มไปลบออกจากโมเมนต์ของฟิลเตอร์ที่ได้จำกัดบางฮาร์โมนิคไปแล้ว หลังจากนั้นสัมประสิทธิ์ของทรานสเฟอร์ฟังก์ชั่นใหม่สามารถคำนวณได้จากกระบวนการวนซ้ำ

ในขั้นตอนการหาค่าสัมประสิทธิ์ของฟิลเตอร์ เราคำนวณเฉพาะสัมประสิทธิ์ของเทอมเศษเท่านั้นเนื่องจากสัมประสิทธิ์ของเทอมส่วน สามารถคำนวณได้จาก สัมประสิทธิ์ของเทอมเศษ และ $R$ โดยวิธีการหาสัมประสิทธิ์ที่มีขั้นตอนดังนี้

1) จากสมการที่ (3.8) และ(4.7) โมเมนต์ของรากของ $b_0 + b_1 z^{-1} + b_2 z^{-2} + ... + b_N z^{-N}$ คำนวณได้ดังนี้

$$S_m^n = 1 + \sum_{i=1}^{M} 2\cos(mi\omega_0) \tag{4.9}$$

เมื่อ m คืออันดับ(Order) ของโมเมนต์ของรากดังนั้นจะได้

$$S_1^n, S_2^n, S_3^n, ..., S_N^n \tag{4.10}$$

ในกรณีเดียวกันจากสมการที่ (4.7) โมเมนต์ของสัมประสิทธิ์ของรากของ $(1 - e^{jk\omega_0} z^{-1})(1 - e^{-jk\omega_0} z^{-1})$ คำนวณได้ดังนี้

$$S_m^d = 2\cos(mk\omega_0) \tag{4.11}$$

เมื่อ k คือฮาร์โมนิคที่ต้องการกำจัดออก

ถ้าเราต้องการกำจัดฮาร์โมนิคที่ $k_1, k_2, ..., k_m$ โมเมนต์ของรากจะกลายเป็น

$$S_m^d = 2\cos(mk_1\omega_0) + 2\cos(mk_2\omega_0) + ... + 2\cos(mk_m\omega_0) \tag{4.12}$$

และในกรณีที่คล้ายกันถ้าต้องการนำส่วนประกอบ DC ออก จะกลายเป็น

$$S_m^d = 1 + 2\cos(mk_1\omega_0) + 2\cos(mk_2\omega_0) + ... + 2\cos(mk_m\omega_0) \tag{4.13}$$

และเราจะได้โมเมนต์ของรากดังนี้

$$S_1^d, S_2^d, S_3^d, ..., S_N^d \tag{4.14}$$

2) เมื่อพิจารณาคุณสมบัติของโมเมนต์ของรากและการหารของสัญญาณ ฉะนั้นโมเมนต์ของรากดังในสมการที่ คำนวณได้ดังนี้

$$S_m^{n'} = S_m^n - S_m^d \tag{4.15}$$

จะได้โมเมนต์ของเทอมเศษที่คำนวณใหม่ดังนี้

$$S_1^{n'}, S_2^{n'}, S_3^{n'}, ..., S_N^{n'} \tag{4.16}$$

3) จากสมการ Newton Identities ดังใน (3.16) สัมประสิทธิ์ของเทอมเศษ ดัง สมการที่ (4.7) คำนวณได้ดังนี้

$$b_m = (-S_m^{n'} - b_1 S_{m-1}^{n'} - b_2 S_{m-2}^{n'} - ... - b_{m-1} S_1^{n'}) / m \tag{4.17}$$

ฉะนั้นเราคำนวณหาสัมประสิทธิ์ของเทอมเศษ ได้จาก

$$a_m = b_m \times \rho^m \tag{4.18}$$

เช่น

$$a_2 = b_2 \times \rho^2 \tag{4.19}$$

และนอกจากนี้จากความสัมพันธ์ของดิจิตอลนอทช์และคอมบ์ฟิลเตอร์ซึ่งมีลักษณะเป็นฟิลเตอร์ซึ่งเป็นคู่กัน (Complementary) ดังสมการต่อไปนี้

$$H_{notch}(\omega) + H_{comb}(\omega) = 1 \tag{4.20}$$

ฉะนั้นจากสมการที่ 4.20 เมื่อเราหาทรานสเฟอร์ฟังก์ชั่นของคอมบ์ฟิลเตอร์ซึ่งมีการกำจัดฮาร์โมนิคที่ $k$ ได้คือ

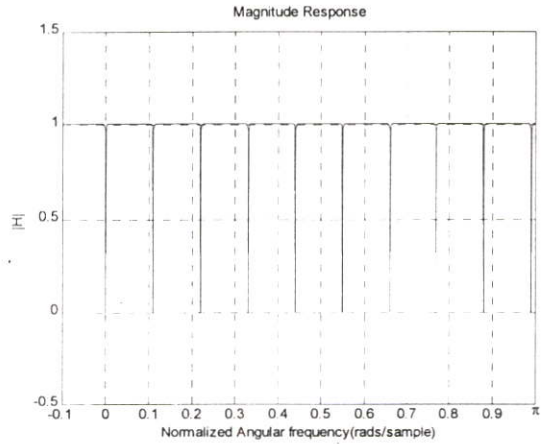
$$H_{comb}(z) = \frac{b(z)}{a(z)} \tag{4.21}$$

จะได้ทรานสเฟอร์ฟังก์ชั่นของนอทช์ฟิลเตอร์ที่เป็นคู่กันได้จาก

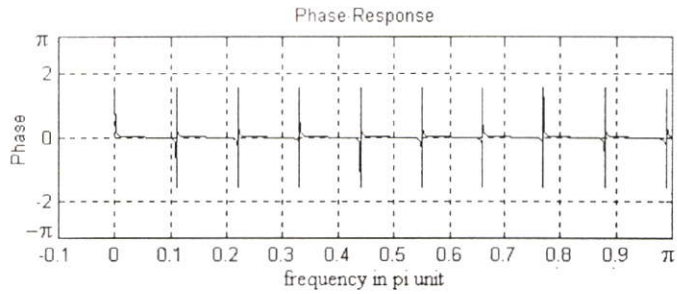
$$H_{notch}(z) = \frac{a(z) - b(z)}{a(z)} \tag{4.22}$$

## 4.4. ผลการทดลอง

### 4.4.1 ผลการจำลองการทำงานโดยใช้โปรแกรม MatLab [17],[18],[19]




**รูปที่ 4.1** ผลตอบสนองขนาดของดิจิตอลคอมบ์ฟิลเตอร์ซึ่ง $\omega_0 = 0.11\pi$ และ $R = 0.999$

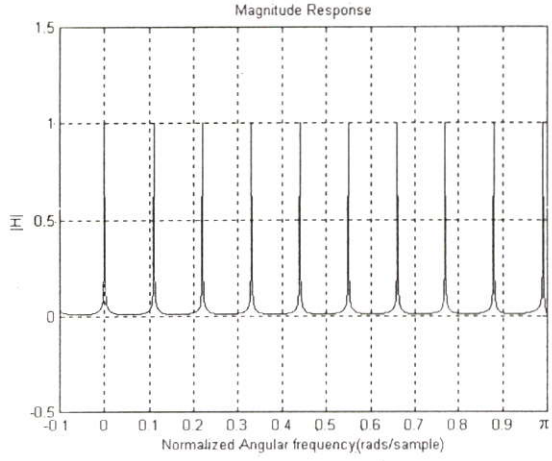



**รูปที่ 4.2** ผลตอบสนองเฟสของดิจิตอลคอมบ์ฟิลเตอร์ซึ่ง $\omega_0 = 0.11\pi$ และ $R = 0.999$

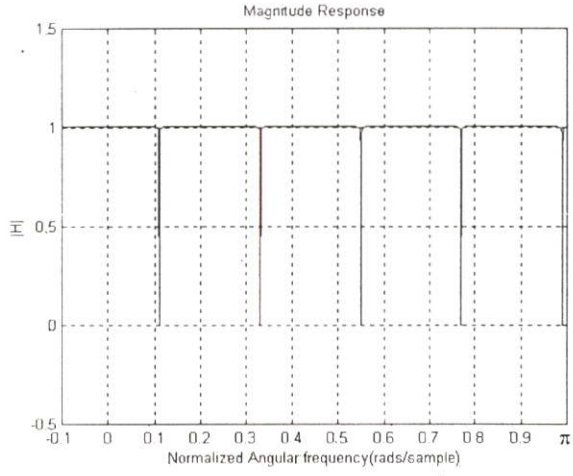
โดยมีค่าสัมประสิทธิ์ของฟิลเตอร์ในรูปที่ 4.1 คือ

$$b_0 = -b_{19} = 1.0000e+0 \quad , b_1 = -b_{18} = 8.1953e-1,$$
$$b_2 = -b_{17} = -7.5994e-2 \, , b_3 = -b_{16} = 3.1104e-2,$$
$$b_4 = -b_{15} = -1.7963e-2 \, , b_5 = -b_{14} = 1.2343e-2,$$
$$b_6 = -b_{13} = -9.4831e-3 \, , b_7 = -b_{12} = 7.9917e-3,$$
$$b_8 = -b_{11} = -7.0581e-3 \, , b_9 = -b_{10} = 6.6784e-3$$

$$a_0 = 1.0000e+0, \quad a_1 = 8.1871e-1, \quad a_2 = -7.5842e-2,$$
$$a_3 = 3.1011e-2, \quad a_4 = -1.7891e-2, \quad a_5 = 1.2281e-2,$$
$$a_6 = -9.4264e-3, \quad a_7 = 7.8565e-3, \quad a_8 = -7.0018e-3,$$
$$a_9 = 6.6185e-3, \quad a_{10} = -6.6119e-3, \quad a_{11} = 6.9808e-3,$$
$$a_{12} = -7.8173e-3, \quad a_{13} = 9.3606e-3, \quad a_{14} = -1.2171e-2,$$
$$a_{15} = 1.7695e-2, \quad a_{16} = -3.0610e-2, \quad a_{17} = 7.4713e-2,$$
$$a_{18} = -8.0490e-1, \quad a_{19} = -9.8117e-1$$




**รูปที่ 4.3** ผลตอบสนองขนาดของดิจิตอลนอร์ชฟิลเตอร์ซึ่ง $\omega_0 = 0.11\pi$ และ $R = 0.999$

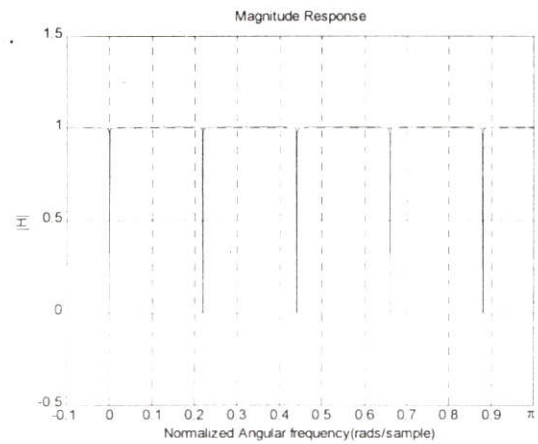



**รูปที่ 4.4** แสดงผลตอบสนองขนาดของดิจิตอลคอมบ์ฟิลเตอร์ซึ่ง $\omega_0 = 0.11\pi$ และ $R = 0.999$ และกำจัด ฮาร์โมนิคคู่

ซึ่งคำนวณค่าโมเมนต์ของรากและสัมประสิทธิ์ของเทอมเศษในรูปที่ 4.4 ได้ดังนี้

$S_1^{n'}$ = -9.1226e-001, $S_2^{n'}$ = 9.2213e-001 , $S_3^{n'}$ = -9.3991e-001 ,
$S_4^{n'}$ = 9.6821e-001, $S_5^{n'}$ = -1.0125e+000, $S_6^{n'}$ = 1.0853e+000,
$S_7^{n'}$ = -1.2234e+000, $S_8^{n'}$ = 1.5967e+000 , $S_9^{n'}$ = -9.8379e+000,
$S_{10}^{n'}$ = -1.1879e-014, $S_{11}^{n'}$ = -5.0418e-001, $S_{12}^{n'}$ = 6.9616e-001,
$S_{13}^{n'}$ = -8.2898e-001, $S_{14}^{n'}$ = 9.5862e-001, $S_{15}^{n'}$ = -1.1223e+000 ,
$S_{16}^{n'}$ = 1.3893e+000, $S_{17}^{n'}$ = -2.0371e+000, $S_{18}^{n'}$ = 9.3611e+000,
$S_{19}^{n'}$ = 1.1076e+000

$b_0 = b_{10} = 1.0000e+0$, $b_1 = b_9 = 9.1226e-1$,
$b_2 = b_8 = -4.4954e-2$, $b_3 = b_7 = 1.9226e-2$,
$b_4 = -b_6 = -1.2944e-2$, $b_5 = 1.1484e-2$

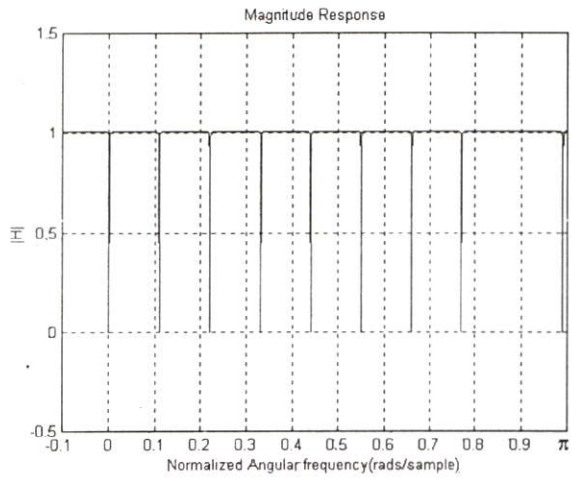



**รูปที่ 4.5** แสดงผลตอบสนองขนาดของดิจิตอลคอมบ์ฟิลเตอร์ซึ่ง $\omega_0 = 0.11\pi$ และ $R = 0.999$ และกำจัด ฮาร์โมนิคคี่

ซึ่งคำนวณค่าโมเมนต์ของรากและสัมประสิทธิ์ของเทอมเศษในรูปที่ 4.4 ได้ดังนี้

$S_1^{n'} = 9.2729e-2$, $S_2^{n'} = -9.8507e-2$, $S_3^{n'} = 1.0933e-1$,
$S_4^{n'} = -1.2759e-1$, $S_5^{n'} = 1.5838e-1$, $S_6^{n'} = -2.1383e-1$,
$S_7^{n'} = 3.2986e-1$, $S_8^{n'} = -6.7556e-1$, $S_9^{n'} = 8.8820e+0$,
$S_{10}^{n'} = 1.0000e+0$, $S_{11}^{n'} = -5.5267e-1$, $S_{12}^{n'} = 4.3600e-1$,
$S_{13}^{n'} = -4.0695e-1$, $S_{14}^{n'} = 4.2916e-1$, $S_{15}^{n'} = -5.0953e-1$,
$S_{16}^{n'} = 7.0376e-1$, $S_{17}^{n'} = -1.2817e+0$, $S_{18}^{n'} = 8.5336e+0$,
$S_{19}^{n'} = 3.3972e+0$

$b_0 = -b_9 = 1.0000e+0,\ b_1 = -b_8 = -9.2729e-2,$
$b_2 = -b_8 = 5.3553e-2,\ b_3 = -b_6 = -4.1145e-2,$
$b_4 = -b_5 = 3.6706e-2,$




**รูปที่ 4.6** แสดงผลตอบสนองขนาดของดิจิตอลคอมบ์ฟิลเตอร์ซึ่ง $\omega_0 = 0.11\pi$ และ $R = 0.999$ และกำจัดฮาร์โมนิคที่ 8

ซึ่งคำนวณค่าโมเมนต์ของรากและสัมประสิทธิ์ของเทอมเศษในรูปที่ 4.5 ได้ดังนี้

$S_1^{n'} =$ 1.0400e+000 $S_2^{n'} =$ -6.3432e-001, $S_3^{n'} = 2.0986e-002,$
$S_4^{n'} = 7.1503e-001,\ S_5^{n'} =$ -1.4721e+000 , $S_6^{n'} = 2.1463e+000,$
$S_7^{n'} =$ -2.6461e+000, $S_8^{n'} =$ 2.9054e+000, $S_9^{n'} =$ -2.8931e+000,
$S_{10}^{n'} =$ 2.6180e+000, $S_{11}^{n'} =$ -2.1285e+000, $S_{12}^{n'} = 1.5069e+000,$
$S_{13}^{n'} =$ -8.6117e-001, $S_{14}^{n'} = 3.1613e-001,\ S_{15}^{n'} =$ -1.3818e-002,
$S_{16}^{n'} = 1.5591e-001,\ S_{17}^{n'} =$ -1.3346e+000, $S_{18}^{n'} = 1.6142e+001,$
$S_{19}^{n'} = 4.3972e+000$

$b_0 = -b_{17} = 1.0000e+0,\ b_1 = -b_{16} = -1.0400e+0,$
$b_2 = -b_{15} = 8.5798e-1,\ b_3 = -b_{14} = -5.2434e-1,$
$b_4 = -b_{13} = 9.9087e-2,\ b_5 = -b_{12} = 3.5242e-1,$
$b_6 = -b_{11} = -7.6392e-1,\ b_7 = -b_{10} = 1.0760e+0,$
$b_8 = -b_9 = -1.2441e+0,$

**4.4.2 ผลการทดลองโดยใช้ตัวประมวลผลสัญญาณดิจิตอล (TMS320C31:Texas Instrument)[20], [21]**

**การกำหนดตัวแปรต่างๆ**

| | |
|---|---|
| $F_s$ | คือความถี่แซมปลิ้ง |
| Order | คือออเดอร์ของฟิลเตอร์ |
| n | คือจำนวนฮาร์โมนิคที่ต้องการกำจัด |
| k | คือฮาร์โมนิคที่ต้องการกำจัด |
| $S_m^n$ | คือโมเมนต์ของรากของคอมบ์ฟิลเตอร์ |
| $S_m^d$ | คือโมเมนต์ของรากของฮาร์โมนิคที่ต้องการกำจัด |
| $S_m^{n'}$ | คือ โมเมนต์ของรากผลลัพธ์ |
| NewOrder | คือออเดอร์ของฟิลเตอร์หลังจากกำจัดฮาร์โมนิคที่ไม่ต้องการแล้ว |
| $a_n$ | คือสัมประสิทธิ์ของเทอมส่วน |

**การคำนวณความถี่แซมปลิ้ง**

$$SCF_{new} = \frac{288(kHz) \times Bw}{3600(Hz)} \tag{4.23}$$

$$TA = \frac{6.25(MHz)}{2 \times SCF_{new}} \tag{4.24}$$

$$TB = \frac{6.25(MHz)}{2 \times TA \times TB} \tag{4.25}$$

เมื่อ $SCF_{new}$ คือค่าความถี่ของสวิทช์คาปาซิเตอร์ฟิลเตอร์

$TA$ คือค่าของรีจิสเตอร์ $TA$ ซึ่งเป็นรีจิสเตอร์ที่ใช้ในการกำหนดความถี่แซมปลิ้ง

$TB$ คือค่าของรีจิสเตอร์ $TB$ ซึ่งเป็นรีจิสเตอร์ที่ใช้ในการกำหนดความถี่แซมปลิ้ง

$Bw$ คือแบนท์วิดท์

กำหนดค่าของแบนท์วิดท์แล้วคำนวณ $TA$ และ $TB$ ตามสมการที่ (4.23)-(4.25)ซึ่งเป็นค่าทศนิยม ประมาณค่าของ $TA$ และ $TB$ ให้เป็นจำนวนเต็มแล้วจึงคำนวณหาค่าของความถี่แซมปลิ้งจริงจากสมการต่อไปนี้

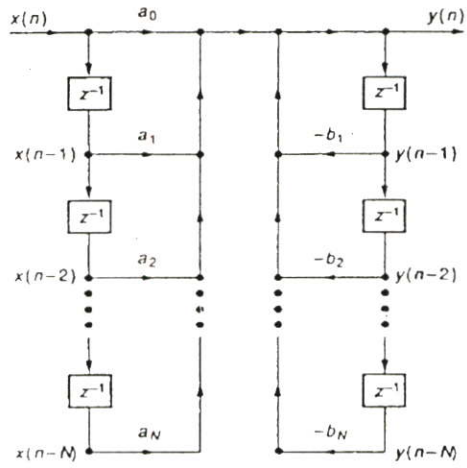
$$F_s = \frac{MCLK}{2 \times TA \times TB} = \frac{6.25(MHz)}{2 \times TA \times TB} \tag{4.26}$$

ซึ่งในที่นี้ได้กำหนดค่าตัวแปรในการคำนวณความถี่แซมปลิ้งดังนี้

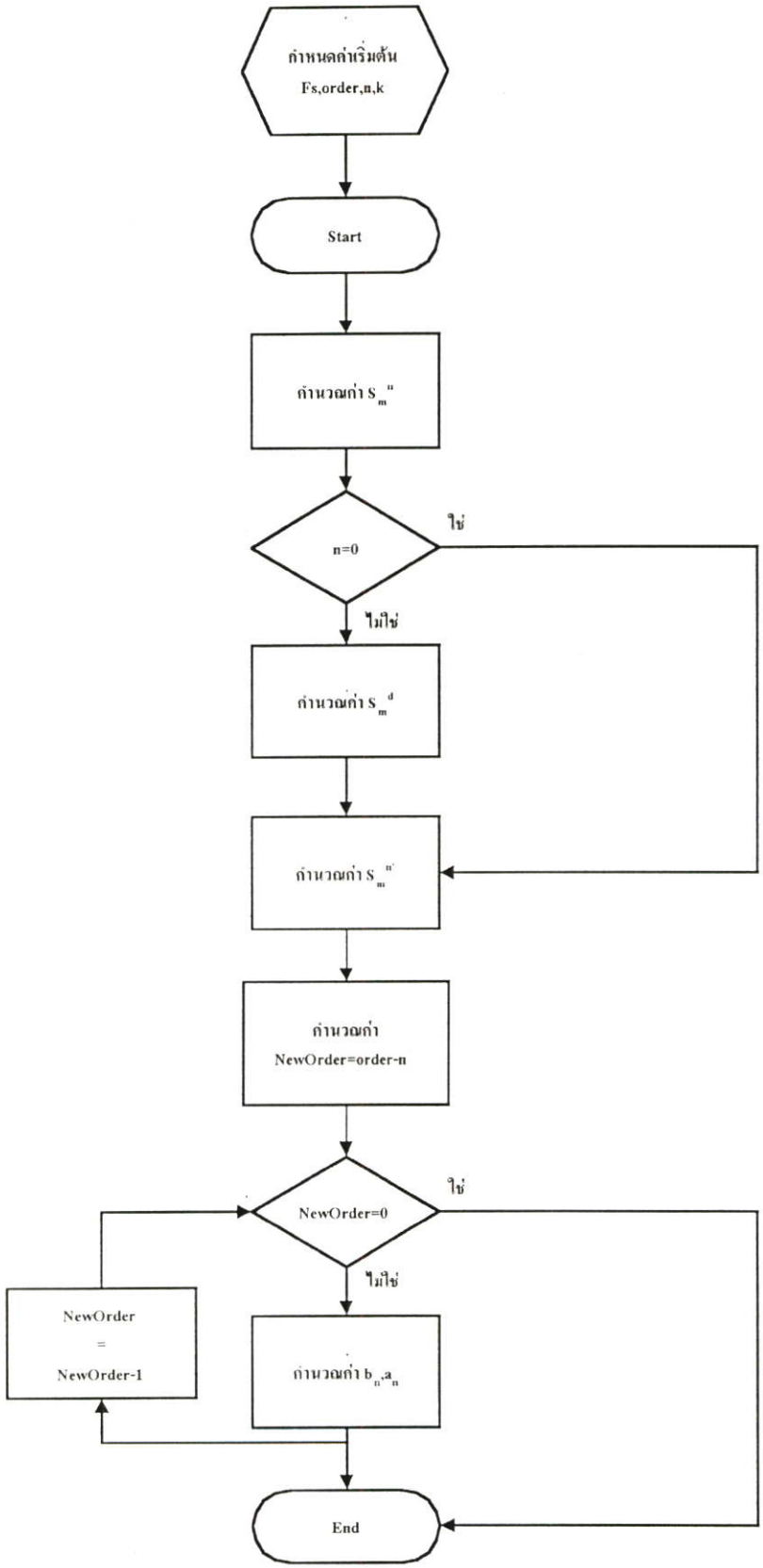
$$Bw = 2909Hz$$
$$TA = 13 = 1A34h$$
$$TB = 41 = 52A6h$$
$$F_s(\mathrm{Re}al) = 5863Hz$$

**รูปแบบของฟิลเตอร์ที่ใช้ในการสร้าง**

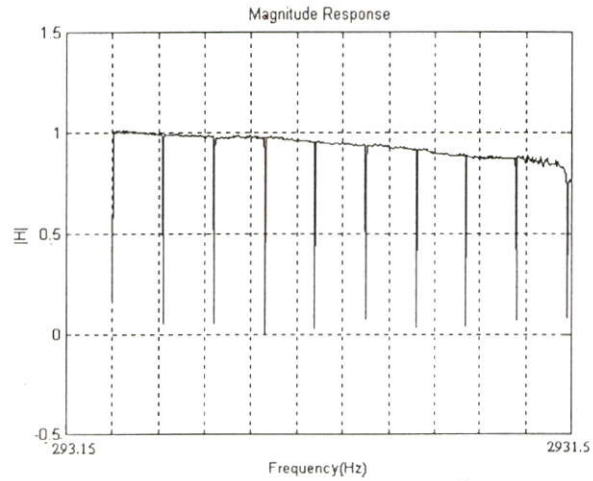
ใช้รูปแบบ Direct-Form I ดังรูป



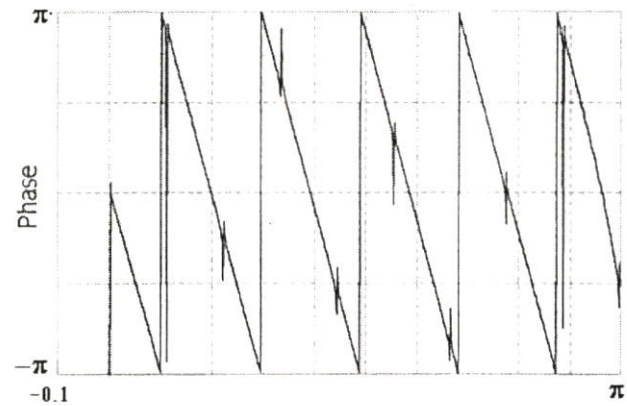

**รูปที่ 4.7** แสดงรูปแบบในการสร้างไอไออาร์ฟิลเตอร์ (Direct Form-I)



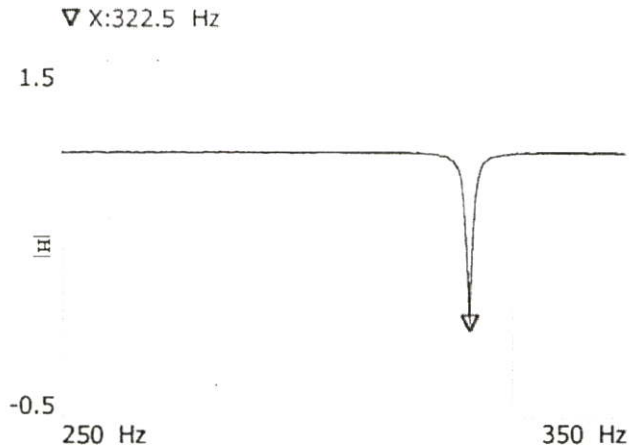

รูปที่ 4.8 แสดงโฟลชาทการคำนวณสัมประสิทธิ์โดยใช้ตัวประมวลผลสัญญาณดิจิตอล



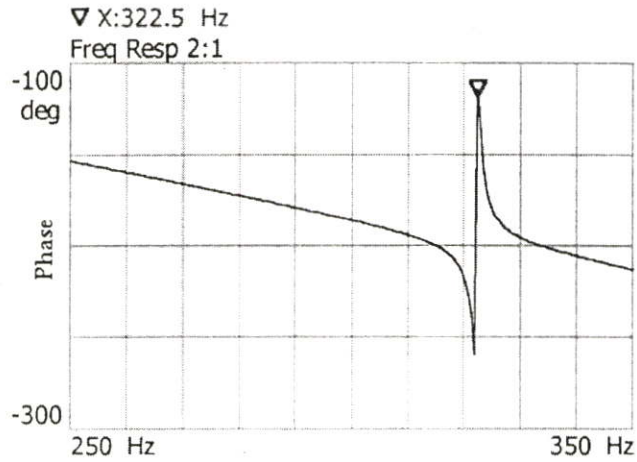

**รูปที่ 4.9** แสดงผลตอบสนองขนาดของดิจิตอลคอมบ์ฟิลเตอร์เมื่อกำหนดให้ Fs=5863 Hz และ $\omega_0 = 0.11\pi$ (322.5 Hz)



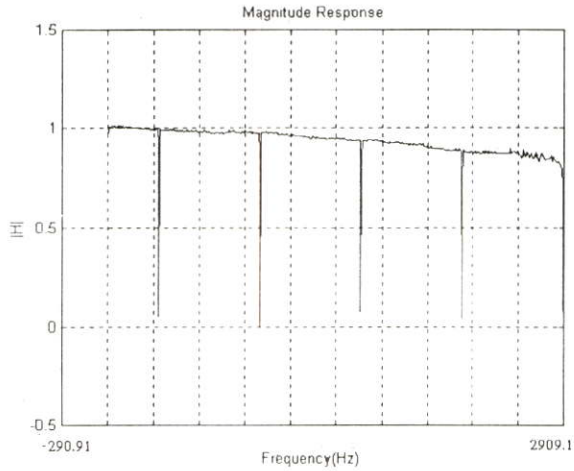

**รูปที่ 4.10** แสดงผลตอบสนองเฟสของดิจิตอลคอมบ์ฟิลเตอร์เมื่อกำหนดให้ Fs=5863 และ $\omega_0 = 0.11\pi$ (322.5 Hz)



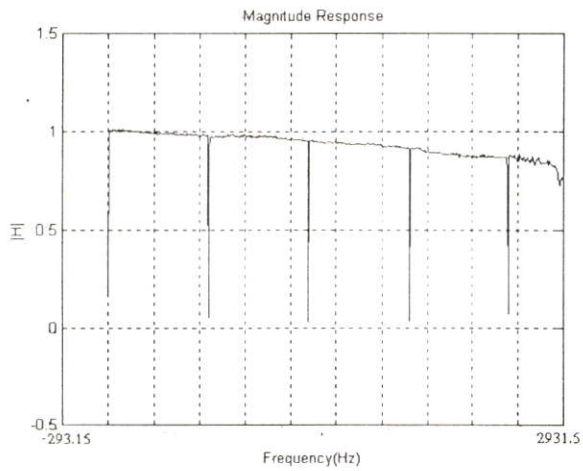

**รูปที่ 4.11** แสดงผลตอบขนาดของดิจิตอลคอมบ์ฟิลเตอร์ที่ตำแหน่งฮาร์โมนิคหลักมูลเมื่อกำหนดให้ Fs=5863 Hz และ $\omega_0 = 0.11\pi$ (322.5 Hz)



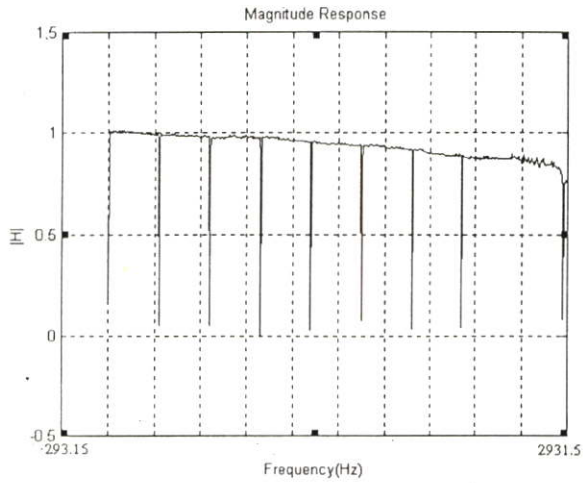

**รูปที่ 4.12** แสดงผลตอบเฟสของดิจิตอลคอมบ์ฟิลเตอร์ที่ตำแหน่งฮาร์โมนิคหลักมูลเมื่อกำหนดให้ Fs=5863 Hz และ $\omega_0 = 0.11\pi$ (322.5 Hz)




**รูปที่ 4.13** แสดงผลตอบสนองขนาดของดิจิตอลคอมบ์ฟิลเตอร์เมื่อกำหนดให้ Fs=5863 Hz และ $\omega_0 = 0.11\pi$ (322.5 Hz) และกำจัดฮาร์โมนิคคู่




**รูปที่ 4.14** แสดงผลตอบสนองขนาดของดิจิตอลคอมบ์ฟิลเตอร์เมื่อกำหนดให้ Fs=5863 Hz และ $\omega_0 = 0.11\pi$ (322.5 Hz) และกำจัดฮาร์โมนิคคี่




**รูปที่ 4.15** แสดงผลตอบสนองขนาดของดิจิตอลคอมบ์ฟิลเตอร์เมื่อกำหนดให้ Fs=5863Hz และ $\omega_0 = 0.11\pi$ (3202.5 Hz) และกำจัดฮาร์โมนิคที่ 8

## 4.5 สรุปผลการทดลอง

จากผลการจำลองการทำงานโดยใช้โปรแกรม MatLab นั้นให้ผลตอบสนองที่ดีคือฟิลเตอร์มีการลดทอนที่ดีทั้งในช่วงความถี่ผ่านและการลดทอน ณ ตำแหน่งของฮาร์โมนิค ซึ่งเมื่อพิจารณาผลการทดลองโดยการสร้างบนตัวประมวลสัญญาณดิจิตอล(TMS320C31:Texas InStrument) แล้วพบว่าผลตอบสนองขนาดให้ผลตอบสนองที่ไม่เป็นไปตามค่าทฤษฎีโดยเฉพาะช่วงความถี่ผ่านในย่านความถี่สูง ทั้งนี้เนื่องจากข้อจำกัดทางด้านฮาร์ดแวร์หลายประการดังเช่น คุณลักษณะของวงจรแปลงดิจิตอล/อนาลอก และวงจรแปลงอนาลอกเป็นดิจิตอล และนอกจากนี้ยังเกิดความผิดพลาดเนื่องจากการคำนวณค่าของตัวประมวลผลซึ่งแม้ว่าจะสามารถคำนวนในลักษณะของรูปแบบ Floating Point ก็ตาม

# บทที่ 5
# การประยุกต์ใช้ในงานประเภทอื่นๆ

ในบทนี้กล่าวถึงการประยุกต์ใช้งานของฟิลเตอร์และ การกำจัดฮาร์โมนิคของดิจิตอลคอมบ์ฟิลเตอร์ในบทที่ 4 โดยส่วนใหญ่แล้วการนำไปใช้งานมักจะอยู่ในส่วนของการประยุกต์ใช้กับการวิเคราะห์ สังเคราะห์ ปรับแต่งสัญญาณเสียง โดยเฉพาะสัญญาณเสียงจากเครื่องดนตรีซึ่งมีลักษณะเป็นฮาร์โมนิค ซึ่งในบางครั้งมีความจำเป็นในการกรองเพียงบางฮาร์โมนิคเพื่อสร้างโครงสร้างฮาร์โมนิคที่มีลักษณะตามต้องการ นอกจากนี้สามารถนำไปประยุกต์ใช้ในการออกแบบดิจิตอลฟิลเตอร์ชนิดอื่นดังจะได้กล่าวต่อไป

## 5.1 การออกแบบออลพาสฟิลเตอร์ที่มีเฟสดีเลย์ตามต้องการ

ออลพาสฟิลเตอร์ที่มีเฟสดีเลย์ตามต้องการหรือ $z^{-D}$ ได้ถูกนำไปใช้ในการประมวลผลสัญญาณดิจิตอลในงานประเภทต่างๆ ดังเช่น การซิงโครไนซ์ของดิจิตอลโมเด็ม การแปลงอัตราแซมปลิ้ง การสังเคราะห์สัญญาณเสียงเครื่องดนตรี เป็นต้น[1] จาก [2] สามารถออกแบบดิจิตอลคอมบ์ฟิลเตอร์โดยใช้ $z^{-D}$ ดังนั้นในขณะเดียวกันก็สามารถออกแบบ $z^{-D}$ จากดิจิตอลคอมบ์ฟิลเตอร์ได้เช่นเดียวกัน โดยการออกแบบดิจิตอลคอมบ์ฟิลเตอร์โดยใช้โมเมนต์ของรากในบทที่ 4

เมื่อต้องการออกแบบ $z^{-D}$ ซึ่งก็คือออลพาสฟิลเตอร์ที่มีเฟสดีเลย์ $D$ เมื่อ

$$z^{-D} = e^{-j\omega D} \tag{5.1}$$

เฟสของมันสามารถคำนวณได้จาก

$$\angle z^{-D} = -\omega D \tag{5.2}$$

และคำนวณเฟสดีเลย์ได้จาก

$$phd(z^{-D}) = -\frac{\angle z^{-D}}{\omega} = -\frac{\omega D}{\omega} = D = \frac{2\pi}{\omega_0} \tag{5.3}$$

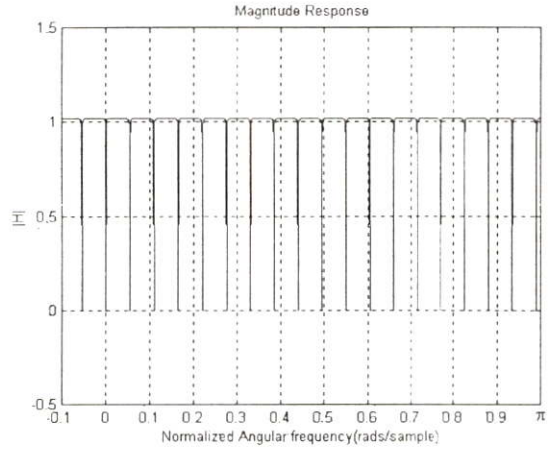
เมื่อทรานสเฟอร์ฟังก์ชั่นของคอมบ์ฟิลเตอร์ที่ออกแบบโดยวิธีโมเมนต์ของรากคือ

$$H_c(z) = \frac{b(z)}{a(z)} \tag{5.4}$$

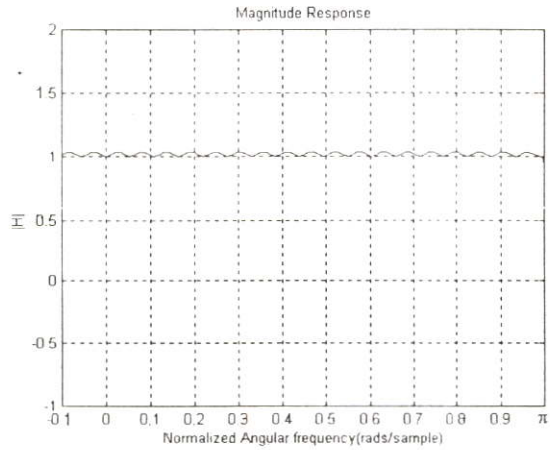
ออลพาสฟิลเตอร์ที่มีเฟสดีเลย์ $D$ คำนวณได้จาก

$$z^{-D} = \frac{b(z) - a(z)}{b(z)R^{D} - a(z)} \tag{5.5}$$

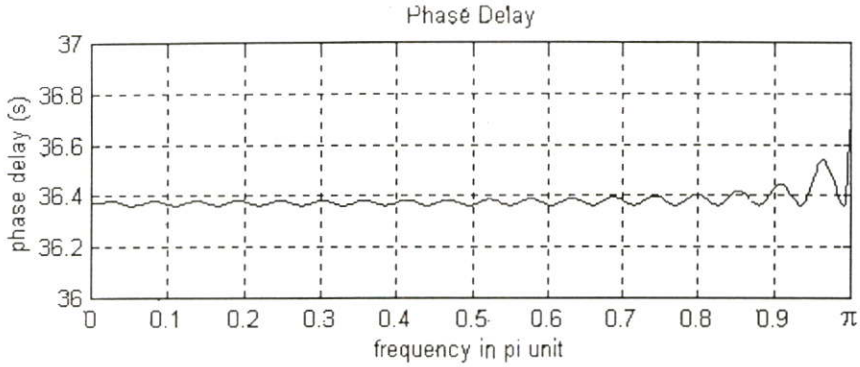
และนอกจากนี้การกำจัดบางฮาร์โมนิคของดิจิตอลคอมบ์ฟิลเตอร์ คือการเปลี่ยนแปลงค่า $D$ ของออลพาสฟิลเตอร์โดยที่ฮาร์โมนิคที่คงเหลือต้องมีช่วงความถี่ที่เท่ากัน ดังเช่นจากรูปที่ 5.1 ซึ่งก็คือคอมบ์ฟิลเตอร์ที่มีค่าของ $\omega_0$ คือ $0.055\pi$ และสามารถออกแบบออลพาสฟิลเตอร์ที่มีค่า $D$ เท่ากับ 36.364 ที่มีผลตอบสนองขนาดรูปที่ 5.2 และเฟสดีเลย์ดังรูปที่ 5.3 เมื่อกำจัดฮาร์โมนิคคี่ จะได้ออลพาสฟิลเตอร์ที่มีค่า $D$ ลดลงเป็นครึ่งหนึ่งดังรูปที่ 5.4 และเมื่อกำจัดฮาร์โมนิคให้เหลือเฉพาะฮาร์โมนิคที่ 4,8,12,16 จะได้ออลพาสฟิลเตอร์ที่มีเฟสดีเลย์ดังรูปที่ 5.5 และในกรณีดังกล่าวเมื่อต้องการกำจัดความถี่ฮาร์โมนิค $f_0$ เดิมการกำจัดบางฮาร์โมนิคในกรณีข้างต้นก็คือการเปลี่ยนแปลงอัตราการแซมปลิ้งนั่นเอง



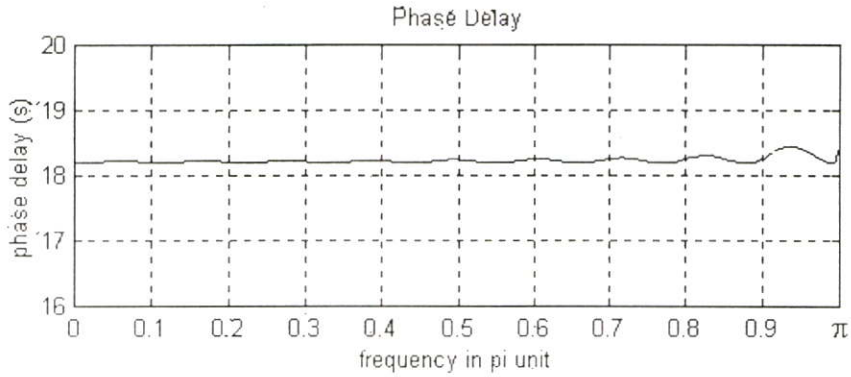

**รูปที่ 5.1** แสดงผลตอบสนองขนาดของคอมบ์ฟิลเตอร์เมื่อ $\omega_0 = 0.055\pi, R = 0.999$



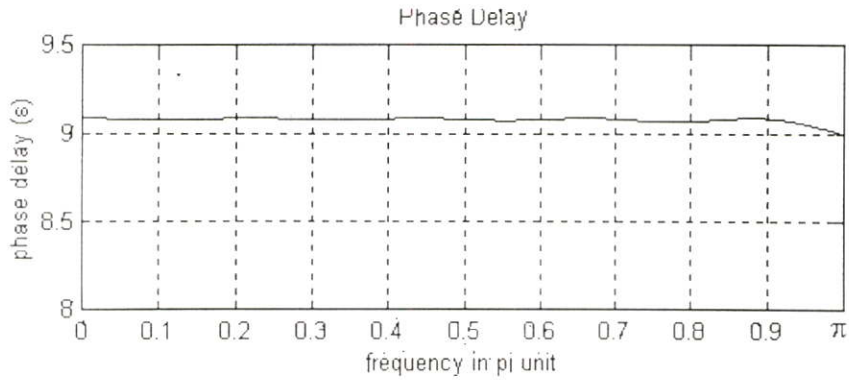

**รูปที่ 5.2** แสดงผลตอบสนองขนาดของออลพาสฟิลเตอร์เมื่อ $\omega_0 = 0.055\pi, R = 0.999$




**รูปที่ 5.3** แสดงผลตอบสนองเฟสดีเลย์ของออลพาสฟิลเตอร์เมื่อ $\omega_0 = 0.055\pi, R = 0.999$




**รูปที่ 5.4** แสดงผลตอบสนองเฟสดีเลย์ของออลพาสฟิลเตอร์เมื่อ $\omega_0 = 0.055\pi, R = 0.999$ และกำจัดฮาร์โมนิคคี่

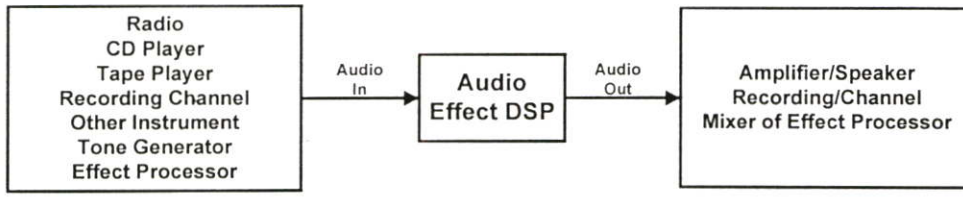



**รูปที่5.5** แสดงผลตอบสนองเฟสดีเลย์ของออลพาสฟิลเตอร์เมื่อ $\omega_0 = 0.055\pi, R = 0.999$ ที่เหลือฮาร์โมนิคที่ 4,8,12,16

## 5.2. เอฟเฟคของสัญญาณเสียงดิจิตอล

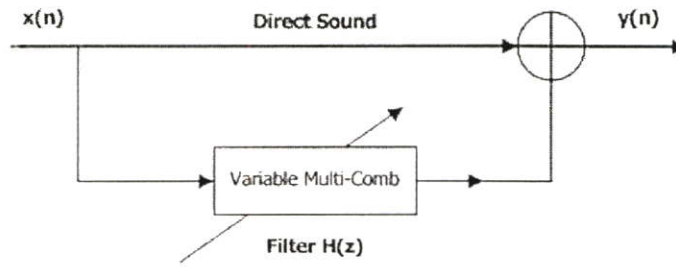
เอฟเฟคทางเสียงดังเช่น ดีเลย์ (Delay) เอคโค (Echo) รีเวอบเบชั่น (Reverberation) คอมบ์ฟิลเตอร์ และแฟรงกิ้ง (Franking) คลอรัส พิทช์ชิฟ สเตอริโออิมเมจจิ้ง เป็นกระบวนการที่อยู่ในขั้นตอนการผลิตเสียงดนตรี บางส่วนได้ถูกนำมาใช้ในระบบเสียงบ้านและรถยนต์ เอฟเฟคเหล่านี้จะถูกสร้างขึ้นโดยตัวประมวลผลสัญญาณดิจิตอล ซึ่งอาจจะแยกเป็นโมดูลต่างหากหรือสร้างรวมไว้ในคีย์บอร์ดสเตชั่น

ตัวประมวลผลเอฟเฟคทางเสียง โดยทั่วไปจะอยู่ในรูปแบบดังรูป

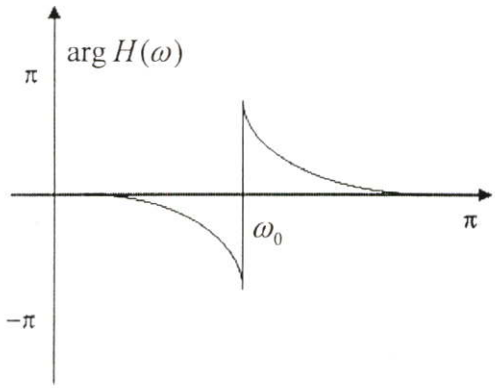



**รูปที่5.6** แสดงระบบการประมวลผลสัญญาณเสียง

เฟสซิ่ง หรือเฟสชิพท์ทิง คือเอฟเฟคทางเสียงชนิดหนึ่งเป็นเอฟเฟคของเครื่องดนตรีจำพวกกีตาร์ คีย์บอร์ด โวคอลลิส สร้างขึ้นโดยนำสัญญาณเสียงผ่านคอมบ์ฟิลเตอร์และนำเอาท์พุทรวมเข้ากับสัญญาณอินพุท ซึ่งเอฟเฟคนี้ทำให้เกิดความแตกต่างของเสียงโดยการกำจัดเสียง ณ ตำแหน่งของฮาร์โมนิคของคอมบ์ฟิลเตอร์ โดยที่สามารถกำหนดตำแหน่งของฮาร์โมนิค จำนวนฮาร์โมนิคได้โดยอิสระ ดังเช่นรูปที่ 5.4 แสดงคอมบ์ฟิลเตอร์ ณ ความถี่ฮาร์โมนิคที่ต้องการกำจัด และผลตอบสนองเฟสของมัน

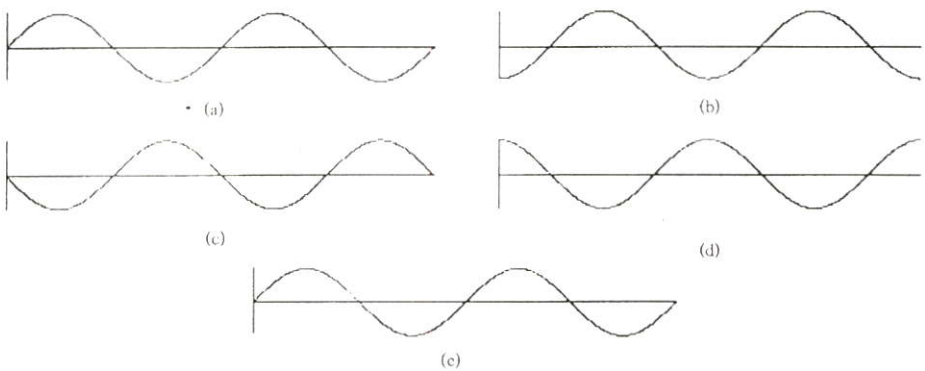



**รูปที่ 5.7** แสดงการสร้าง เอฟเฟคเฟสชิพท์ติง




**รูปที่ 5.8** แสดงผลตอบสนองเฟส ณ. ตำแหน่งของซี่

จาก [4] ได้แสดงให้เห็นว่าการเปลี่ยนแปลงของเฟส ณ ความถี่ฮาร์โมนิคของสัญญาณเสียงนั้น จะทำให้เกิดความแตกต่างในการได้ยินในบางกรณี ดังเช่นการเปลี่ยนแปลงเฟสของสัญญาณดังรูปที่ 5.9 เมื่อนำสัญญาณในรูป 5.9 (a) และ 5.9 (c) รวมเข้าด้วยกันแล้วผลตอบสนองขนาดก็จะกลายเป็น 0 เป็นต้น




**รูปที่ 5.9** แสดงสัญญาณไซน์ที่มีการชิพทีเฟสไปด้วยค่าต่างๆ a. สัญญาณไซน์ b. สัญญาณไซน์ชิพท์เฟสไป $\pi/2$ c.สัญญาณไซน์ชิพท์เฟสไป $\pi$ d. สัญญาณไซน์ชิพท์เฟสไป $3\pi/2$

## 5.3 การวิเคราะห์ สังเคราะห์สัญญาณเสียงจากเครื่องดนตรี [5]

สัญญาณเสียงที่เกิดจากเครื่องดนตรีโดยทั่วไปจะมีโครงสร้างเป็นฮาร์โมนิค ซึ่งสัญญาณดนตรีดังกล่าวแสดงโดยโมเดลทางคณิตศาสตร์ดังต่อไปนี้

$$x(t) = a(t)\sum_{m=1}^{\infty} c_m \cos(2\pi m f_0 t + \phi_m) \tag{5.6}$$

เมื่อ

$f_0$ คือความถี่หลักมูลหรือพิทช์

$c_m$ คือขนาด และ $\phi_m$ คือเฟสของฮาร์โมนิคที่ $m$

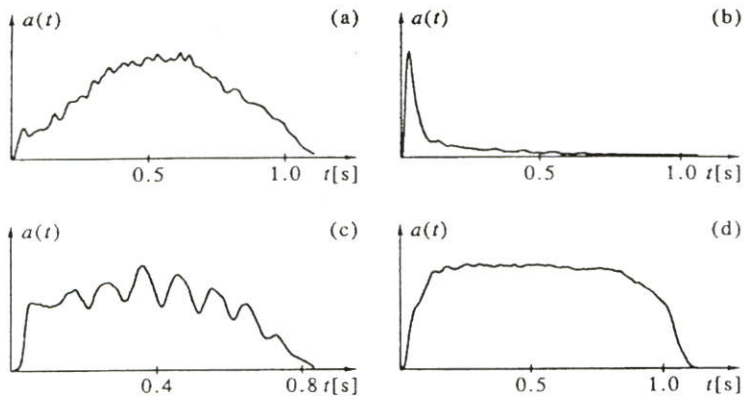
$a(t)$ คือฟังก์ชั่นขอบเขต ซึ่งเป็นสัญญาณความถี่ต่ำ ซึ่งโมดูเลทกับขนาดของสัญญาณ

ความถี่ $f_0$ จะกำหนดโดยโน้ตที่เล่น ในดนตรีตะวันตก ความถี่ของเสียงดนตรีจะเป็นดังสูตรต่อไปนี้

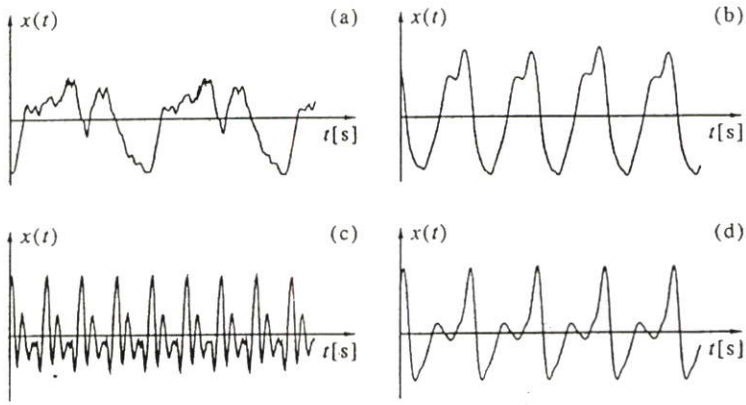
$$f_i = f_{ref} q^i, q = 2^{1/12} \tag{5.7}$$

เมื่อ $f_{ref}$ คือความถี่อ้างอิง ซึ่งความถี่อ้างอิงมาตรฐานคือ 440 Hz และโน้ตที่ความถี่นี้คือ A (หรือ ลา) อัตราส่วน $2^{1/12}$ ก็คือเซมิโทน โน้ต B,C,D,E,F,G ๖(หรือ ที โด เร มี ฟา ซอล ตามลำดับ) คือ 2 3 5 7 8 และ 10 เซมิโทนเหนือโน้ต A ( i= 2,3,5,7,8,10) ซึ่งเรียกว่าโน้ตธรรมชาติ ระหว่าง A และ B ยังมีโน้ต A-Sharp หรือ B-Flat และระหว่าง C และ D, D และ E, F และ G ,G และ A ก็มี Sharp และ Flat เช่นเดียวกัน โน้ตเซมิโทน 12 ตัวเหนือ A ก็จะเรียกว่า A อีกครั้งและเรียกว่าออคเตฟที่สูงกว่า

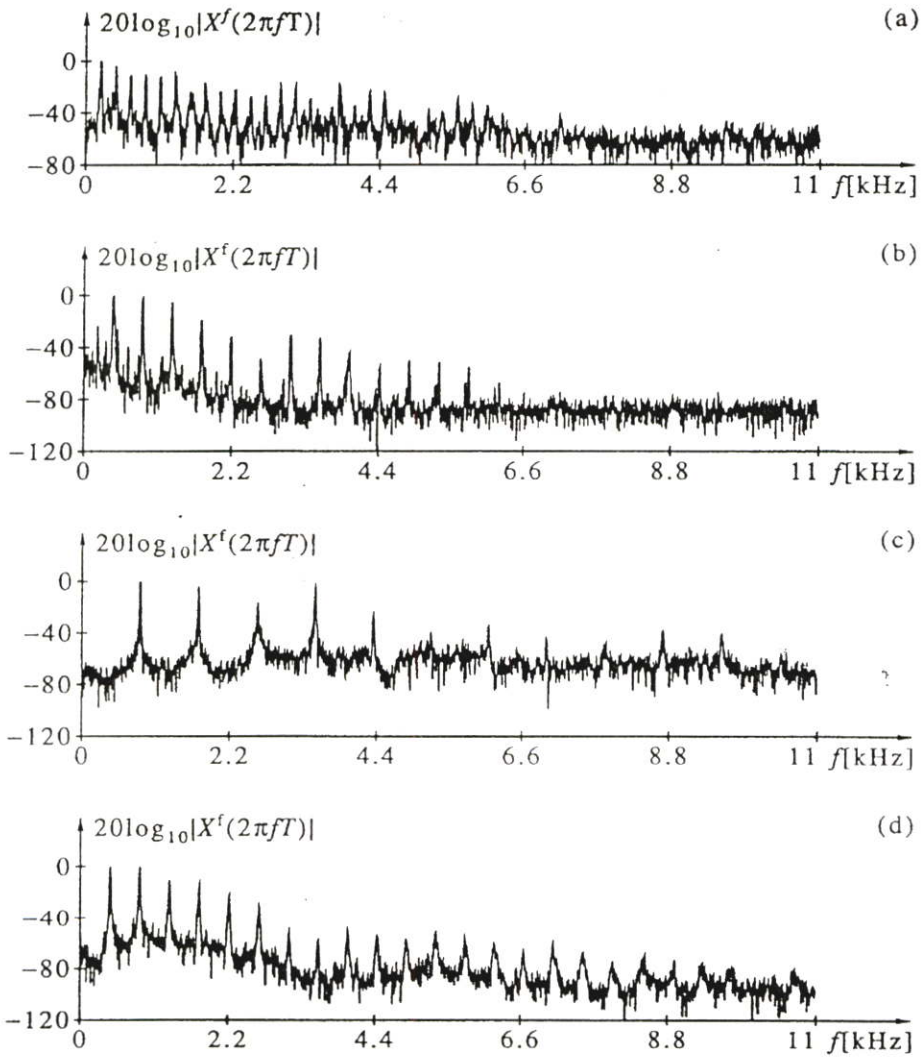
โครงสร้างฮาร์โมนิคของเครื่องดนตรีก็คือลำดับของฮาร์โมนิคที่สัมพันธ์กับความถี่หลักมูล ซึ่งมันขึ้นอยู่กับชนิดของเครื่องดนตรี ลักษณะจำเพาะของอุปกรณ์แต่ละตัว โน้ตและวิธีเล่น เครื่องดนตรีต่างชนิดกันจะมีโครงสร้างฮาร์โมนิคต่างกัน นอกจากโครงสร้างฮาร์โมนิคแล้ว ฟังก์ชั่นขอบเขตก็จะทำให้เสียงที่ได้ยินมีความแตกต่างกัน โดยในกรณีนี้เฟสจะมีความสำคัญเพียงเล็กน้อย



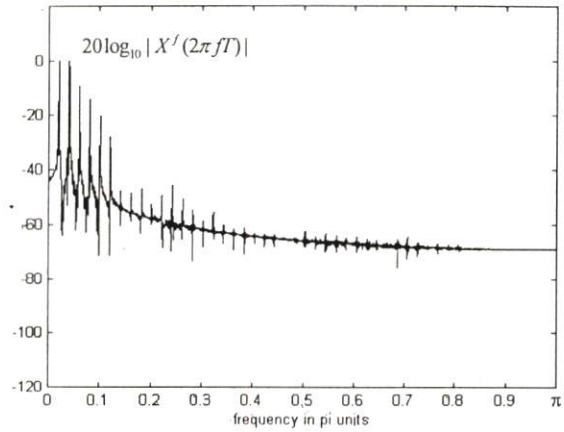

**รูปที่ 5.10** แสดงฟังก์ชั่นขอบเขตของสัญญาณเสียงจากเครื่องดนตรี a. เซลโล(Cello) b. กีตาร์ (Classical guitar) c. ฟลุต (Flute)d. เฟรนช์ฮอร์น (French horn)




**รูปที่ 5.11** แสดงรูปคลื่นที่เกิดจากเครื่องดนตรีชนิดต่างๆ เมื่อเล่นโน้ต A เป็นเวลา 10 มิลลิวินาที a. เซลโล(Cello) b. กีตาร์(Classical guitar) c. ฟลุต (Flute)d. เฟรนช์ฮอร์น (French horn)

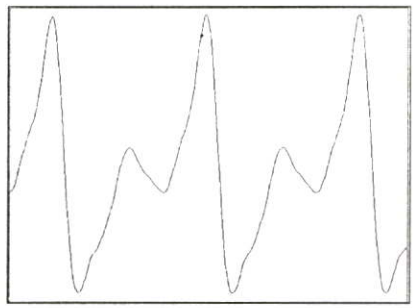



**รูปที่ 5.12** แสดงโครงสร้างฮาร์โมนิคเมื่อเล่นโน้ต A เป็นเวลา 186 มิลลิวินาที a. เซลโล(Cello) b. กีตาร์(Classical guitar) c. ฟลุต (Flute)d. เฟรนช์ฮอร์น (French horn)

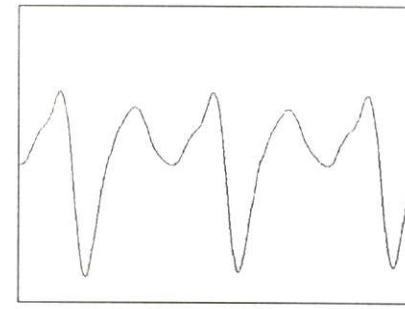



**รูปที่ 5.13** แสดงสเปรคตรัมของเฟร็นช์ฮอร์น(8192 จุด)

รูปที่ 5.10 และ 5.11 แสดงฟังก์ชั่น $a(t)$ และรูปคลื่นที่เกิดจากเครื่องดนตรีต่างชนิดกันเมื่อเล่นโน้ต A นอกจากนี้จากรูปที่ 5.12 แสดงถึงโครงสร้างฮาร์โมนิคของเครื่องดนตรีแต่ละประเภท ซึ่งจากข้อมูล $a(t)$ และโครงสร้างฮาร์โมนิค สามารถสังเคราะห์ วิเคราะห์ ปรับแต่งสัญญาณเสียงดนตรีโดยใช้การประมวลผลสัญญาณดิจิตอล และคอมบ์ฟิลเตอร์ก็มีบทบาทในการสร้างและปรับแต่งในส่วนของโครงสร้างฮาร์โมนิคของเครื่องดนตรีเพื่อให้ได้เสียงที่ต้องการดังรูปที่ 5.13a-5.13d ซึ่งแสดงรูปคลื่นของ เฟรนช์ฮอร์นเมื่อมีการกำจัดฮาร์โมนิคต่างๆ กัน โดยพิจารณาความถี่หลักมูลจากรูปที่ 5.12

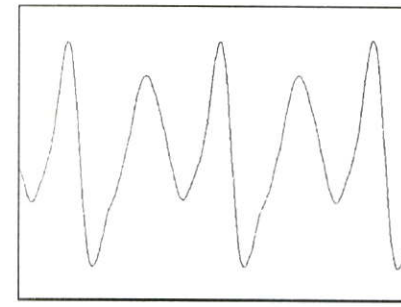


a.



b

c.



d.

**รูปที่ 5.14** แสดงรูปคลื่นของเฟรนช์ฮอร์น โดยรูป a ไม่มีการกรองฮาร์โมนิคใดๆ รูป b กรองทุกฮาร์โมนิค รูป c ไม่กรองฮาร์โมนิคที่ 1 รูป d ไม่กรองฮาร์โมนิคที่ 2 เมื่อกำหนดให้ ( $\omega_0 = 0.02\pi, R = 0.999$ )

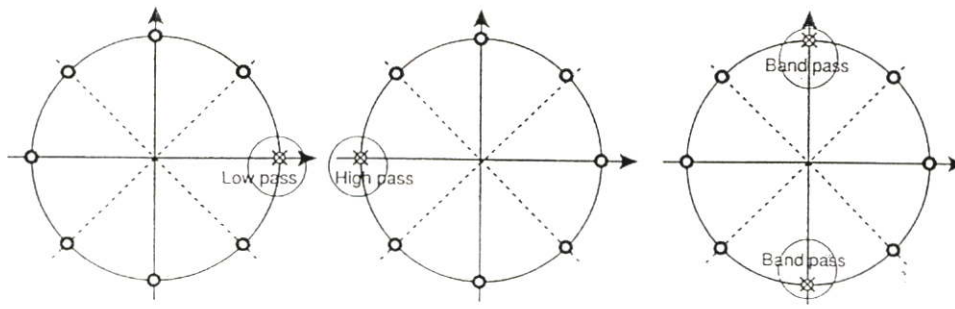
## 5.4 การปรับแต่งเอฟไออาร์คอมบ์ฟิลเตอร์[5]

จากทรานสเฟอร์ฟังก์ชั่นของเอฟไออาร์คอมบ์ฟิลเตอร์ดังต่อไปนี้

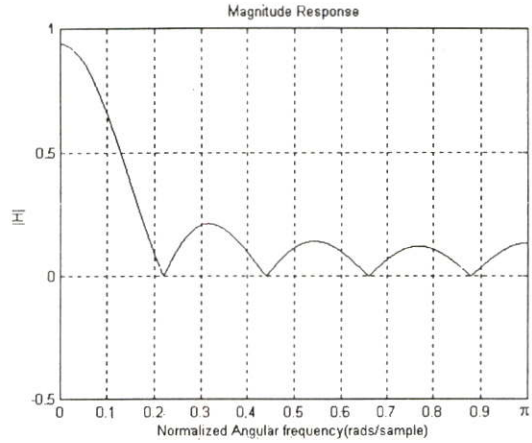
$$H(z)=\frac{Y(z)}{X(z)}=\frac{1}{D}*(1-z^{-D}) \tag{5.8}$$

ซึ่งเป็นทรานสเฟอร์ฟังก์ชั่นในกรณีที่ซีโร่สมมาตร ถ้าในกรณีที่ซีโร่ไม่สมมาตรแล้วจะพบว่าทรานสเฟอร์ฟังก์ชั่นของคอมบ์ฟิลเตอร์ชนิดนี้ก็คือเทอมเศษของคอมบ์ฟิลเตอร์ดังสมการที่ (4.3)นั่นเอง

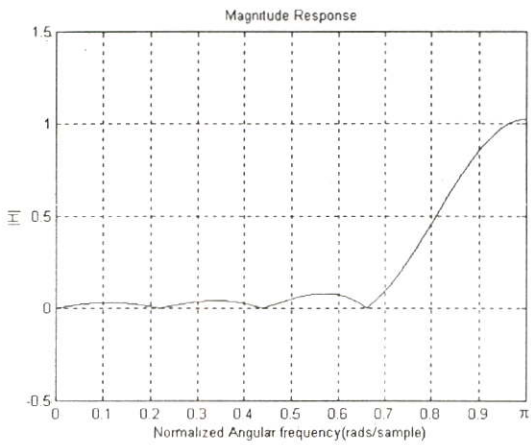
นอกจากการนำไปใช้ประโยชน์ในลักษณะคอมบ์ฟิลเตอร์โดยทั่วไปแล้ว ยังสามารถดัดแปลงทรานสเฟอร์ฟังก์ชั่นของมันให้กลายเป็นทรานสเฟอร์ฟังก์ชั่นชนิดอื่นดังเช่น ตัวกรองความถี่ต่ำผ่าน ตัวกรองความถี่สูงผ่าน ตัวกรองช่วงความถี่ผ่าน โดยการกำจัดซีโร่บางตัวออกไป หรือจากทรานสเฟอร์ฟังก์ชั่นในบทที่ 4 ก็คือคำนวณเฉพาะเทอมเศษเท่านั้น การแปลงเป็นคอมบ์ฟิลเตอร์ประเภทต่างๆดังที่กล่าวมานั้นขึ้นอยู่กับตำแหน่งของซีโร่ที่กำจัดดังในรูปที่5.12 และผลตอบสนองความถี่กำจัดซีโร่ซึ่งทำการแปลงเป็นฟิลเตอร์อื่นๆดังในรูปที่5.13 ,5.14 และ5.15 ตามลำดับ



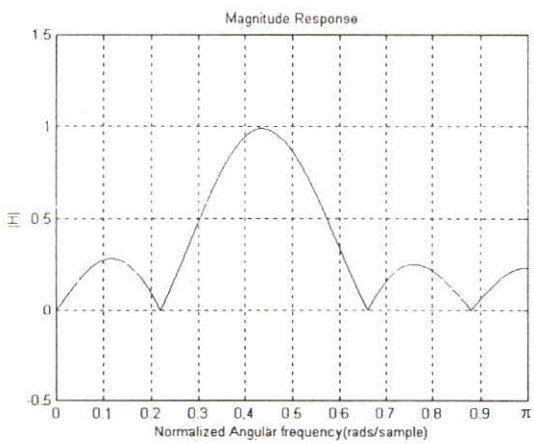

**รูปที่ 5.15** แสดงตำแหน่งของซีโร่เมื่อต้องการแปลงเป็นฟิลเตอร์อื่นๆ




**รูปที่ 5.16** แสดงตำแหน่งของซีโร่เมื่อต้องการแปลงเป็นตัวกรองความถี่ต่ำผ่าน เมื่อ $\omega_0 = 0.22\pi$




**รูปที่ 5.17** แสดงตำแหน่งของซีโร่เมื่อต้องการแปลงเป็นตัวกรองความถี่สูงผ่าน เมื่อ $\omega_0 = 0.22\pi$




**รูปที่ 5.18** แสดงตำแหน่งของซีโร่เมื่อต้องการแปลงเป็นตัวกรองความช่วงความถี่ผ่าน เมื่อ

$\omega_0 = 0.22\pi$

คอมบ์ฟิลเตอร์ชนิดนี้มีข้อดีก็คือเป็นวิธีที่ง่ายในการออกแบบและการเปลี่ยนเป็นฟิลเตอร์ประเภทอื่น โดยเฉพาะเมื่อนำไปสร้างบนตัวประมวลผลสัญญาณดิจิตอลนั้นก็สามารถทำได้โดยง่าย โดยเฉพาะในกรณีที่ซีโร่สมมาตรและ $D$ เป็นค่ายกกำลัง 2 ซึ่งเทคนิคในการเขียนโปรแกรมก็ต้องการเพียงการหน่วงเวลาของอินพุทไป $D$ และการชิพท์ขวาซึ่งก็คือการหาร (เพื่อหาค่า $D$) เท่านั้น แต่อย่างไรก็ตามถ้ากล่าวถึงในด้านเสถียรภาพแล้ววิธีการนี้ให้ผลไม่ดีนักเนื่องจากในกรณีที่เราต้องการกำจัดซีโร่ที่ 0 และ $\pi$ ซึ่งก็คือการนำโพลที่ 0 และ $\pi$ ไปวางถ้าระบบมีการคำนวณที่ไม่เที่ยงตรงพอแล้วจะทำให้เกิดโพลที่ตำแหน่งดังกล่าวซึ่งส่งผลให้ระบบเกิดการออสซิลเลทหรือไม่เสถียรนั่นเอง

# บทที่ 6
# สรุปผลการวิจัยและข้อเสนอแนะ

## 6.1 สรุปผลการวิจัย

จากผลการศึกษาการออกแบบ และวิธีคำนวณสัมประสิทธิ์ของดิจิตอลคอมบ์ฟิลเตอร์ ซึ่งออกแบบโดยวิธีต่างๆ ไม่ว่าจะเป็นการวางซีโร่และโพลแบบสมมาตรและไม่สมมาตร หรือการออกแบบโดยอาศัยออลพาสฟิลเตอร์ก็ตามล้วนแต่อาศัยความสำคัญของตำแหน่งโพลและซีโร่ และจากการวิจัยที่ช่วยให้การคำนวณสัมประสิทธิ์ของดิจิตอลคอมบ์ฟิลเตอร์อยู่ในรูปแบบความสัมพันธ์ที่ชัดเจนของตำแหน่งของซีโร่และโพลและค่าสัมประสิทธิ์ โดยไม่ต้องอาศัยการคูณตัวประกอบของราก สามารถนำไปใช้ในกรณีที่ต้องการสร้างฟิลเตอร์ขึ้นใหม่ การกำจัดฮาร์โมนิคที่ไม่ต้องการของดิจิตอลคอมบ์ฟิลเตอร์โดยเฉพาะในกรณีหลังซึ่งเมื่อการกำจัดฮาร์โมนิคนั้นกระทำกับดิจิตอลคอมบ์ฟิลเตอร์ตัวเดิม สามารถนำวิธีดังกล่าวไปสร้างบนตัวประมวลผลสัญญาณดิจิตอล โดยไม่ต้องคูณตัวประกอบของรากขึ้นใหม่ซึ่งเป็นขั้นตอนที่ไม่สะดวกในการกระทำโดยตัวประมวลผลสัญญาณดิจิตอล ดังนั้นจากผลการวิจัยจึงกล่าวได้ว่าได้ปรับปรุงวิธีการคำนวณสัมประสิทธิ์ของดิจิตอลคอมบ์ฟิลเตอร์ให้ชัดเจนและมีความสะดวกในการนำไปสร้างบนตัวประมวลผลสัญญาณในกรณีที่กล่าวมาแล้ว

## 6.2 ข้อเสนอแนะ

1. ในการวิจัยที่กล่าวมานั้น ผู้วิจัยได้ให้ความสำคัญและมุ่งเน้นไปที่ตำแหน่งของฮาร์โมนิค และได้ประยุกต์โดยนำวิธีของโมเมนต์ของรากเพื่อสร้างวิธีการคำนวณสัมประสิทธิ์ขึ้น แต่นอกจากนี้ยังมีคุณลักษณะอื่น ๆ ที่สำคัญดังเช่น อัตราการลดทอน ณ ตำแหน่งของฮาร์โมนิค หรือความถี่คัทออฟ ซึ่งสามารถพัฒนาวิธีการให้ดียิ่งขึ้นโดยนำลักษณะเหล่านี้มาพิจารณาในการออกแบบด้วย
2. เมื่อนำไปสร้างบนตัวประมวลผลสัญญาณดิจิตอล สามารถปรับปรุงขั้นตอนการคำนวณดังเช่นการคูณและการหารให้อยู่ในรูปแบบของการชิพท์เพื่อลดขั้นตอนและเวลาในการคำนวณ รวมทั้งศึกษาถึงปัจจัยต่างๆที่มีผลต่อการสร้าง ดังเช่นจำนวนบิตในการคำนวณ เพื่อลดความผิดพลาดในการคำนวณและให้ผลตอบสนองที่ใกล้เคียงกับการจำลองผลการทำงานมากขึ้น

# เอกสารอ้างอิง


[1] O. Sophocles. 1996. Introduction to Signal Processing. New Jersey: Prentice Hall.

[2] P. Soo-Chang and T. Chien-Cheng . "A Comb Filter Design Using Fractional-Sample Delay." IEEE Transaction on Circuit and System-II: Analog and Digital Signal Processing, vol. 45 ,no. 6, June 1998.pp 649-653.

[2] L. Timo . et. al. "Splitting the Unit Delay." IEEE Signal Processing magazine,Jan.,1996.pp.30-60

[3] M. Sanjit and J. F. Kaiser. Handbook for Digital Signal Processing. A Wiley-Interscience Publication. 1993.

[4] T. Stathaki. "Root moments: a Digital Signal-Processing Perspective." IEE. Proc-Vis Image Signal Process, vol. 145,no. 4,August.1998. pp. 293-302

[5] A. Antoniou . Digital Filter Analysis, Design, and Applications. Mcgraw-Hill. 1993

[6] C. L. Nikias, and F. Liu. "Bicepstrum on Computation Based on Second and Third Order Statistics with Applications." Proceeding of ICASSP 90,Albuqurque, New Mexico, April, 1990. pp. 2381

[7] B. P. Bogert. "The Frequency Analysis of Time Series for Echos : Cepstrum, Pseudo-Auto Covariances, Cross-Cepstrum and Saphe-Tracking." Proceeding of Symposium on Time Series Analysis,1963. pp. 209

[8] D. H. Brocks. and C. L. Nikias. "The Cross-Bicepstrum:Propertes and Applications for Signal Reconstruction and System Identification." Proceeding of ICASSP 91,Toronto, Canada, May 1991. pp. 3433

[9] P. Henrici . Applied and Computation Complex Analysis. New York: Mcgraw-Hill. 1986.

[10]N. E. Hurt. Phase Retrival and Zero Crossing: Mathematical method in Image Reconstruction. Kluwer Academic Publisher. 1998

[11] R. C. Snell. and F. Milinazgo. " Formant Location from LPC Analysis Data. IEEE Trans, 1993. Assp-1.

[12] C. L. Nikias. and A. Petropulu. : Higher Order Spectra Analysis. Prentice Hall.1993.

[13] DaQarta. Harmonic Distortion.[Online].Available: http://www.daqarta.com/odhhdist.htm.

[14]เพ็ญฤดี ตั้งสมวรพงษ์ และฟูศักดิ์ ชีวสุวิทย์. " การกำจัดบางฮาร์โมนิคที่ไม่ต้องการของดิจิตอล คอมบ์ฟิลเตอร์". วิศวกรรมสารลาดกระบัง, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง, ฉบับที่2, มิถุนายน 2545. หน้า 99-104.

[15] P. Thangsomvorapong and F. Cheevasuvit. "Specified Harmonic Removal of Digital Comb Filter by Root Moments." Proc. IEEE ICIT'02, vol. I, Bangkok, Thailand. 2002. pp. 323-326

[16] มนัส สังวรศิลป์ และ วรรัตน์ ภัทรอมรกุล. คู่มือการใช้งาน Matlab ฉบับสมบูรณ์. Info Press, 2543

[17] The Math Work Inc. The Student Edition of Matlab version 4., Prentice Hall., 1995.

[18] I. K. Viney, J. G. Proakis. Digital Signal processing Using matlab V.4, PWS Publishing Company, 1997.

[19] Texas Instrument, User's Guide TMS320C31, Taxas Instrument., 1994.

[20] R. Chassaing, Digital Signal Processing Laboratory Experiments Using C and TMS 320C31 DSK, John Wiley &Sons, 1999.

[21]S. Lehman. 1996, Phase Shifting, [Online]. Available: http://www.harmony-Central.com/Effect/Articles/Phase-Shifting.htm.

[22] B. Porat, A Course in Digital Signal Processing, John Wiley &Sons, 1997.

[23] J. P. Deziel, Applied Introduction to Digital Signal Processing, Prentice Hall, 2001.

# ภาคผนวก

## ก. บทความที่ได้รับการตีพิมพ์

1. "Specified Harmonic Removal of Digital Comb Filter by Root Moments", Proc. IEEE ICIT'02, vol. I, Bangkok, Thailand. 2002. pp. 323-326

# Specified Harmonic Removal of Digital Comb Filter by Root Moments

**P. Thangsomvorapong and F. Cheevasuvit**

Department of Instrumentation Engineering, Faculty of Engineering
King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand
Phone: (66-2326-4204) E-mail: s3061405@kmitl.ac.th

**Abstract**

This paper presents the designing of comb filter coefficients removing some undesired harmonic frequencies. The root moment method has been used to generate the desired comb filter coefficients. The proposed method is quite suitable when the relation of $2\pi$ and fundamental $\omega_0$ is fractional.

**Keywords:** digital comb filter, splitting unit delay, root moments,

## 1. Introduction

Digital comb filter is designed for removing some undesired harmonic frequency [1] while the other frequencies are left through the output of filter. A typical one of the application is the cancellation process of interference in the recording of electrocardiogram (ECG) at which the digital comb filter plays in this role. For some applications, we want to remove some specified harmonics while maintain the DC signal and some other harmonics. Therefore, in this paper, we present a design method of the mentioned comb filter by using root moment procedures. Here, we focus to the case of the relation of which $2\pi$ and fundamental harmonic is fractional.

## 2. Specified Harmonic Removal of Digital Comb Filter

General form of digital comb filter is given by

$$H_c(z)=\frac{1-z^{-D}}{1-\rho^D z^{-D}} \tag{1}$$

where $D=2\pi/\omega_0$, D is integer and $0<\rho<1$.

When D is not integer, the digital comb filter will be designed by approximating $z^{-D}$ from all pass filter or FIR low pass filter, then put it in (1). This technique is illustrated in [1] and the transfer function of the digital comb filter is shown by

$$H_c(z)=\frac{b_0+b_1z^{-1}+b_2z^{-2}+\ldots+b_Nz^{-N}}{a_0+a_1z^{-1}+a_2z^{-2}+\ldots+a_Nz^{-N}} \tag{2}$$

When zeros and poles of the filter in equation (2) are considered, the frequency positions of zeros are at $k\omega_0$ on the interval $[-\pi \quad \pi]$ for $-M \le k < M$ as shown in the following

$$[-M\omega_0\pi \quad \ldots \quad -2\omega_0\pi \quad -\omega_0\pi \quad 0 \quad \omega_0\pi \quad 2\omega_0\pi \quad \ldots \quad M\omega_0\pi] \tag{3}$$

where $M=\pi/\omega_0$. The frequency positions of poles are at $k\omega_0$ too, but their magnitudes are $\rho$.

When we want to remove the harmonic of digital comb filter, will cancel zeros and poles of the harmonic and its conjugate out [2]. For instance, if the $k^{th}$ harmonic is need to remove,

$$H_c'(z)=\frac{b_0+b_1z^{-1}+b_2z^{-2}+\ldots+b_Nz^{-N}}{a_0+a_1z^{-1}+a_2z^{-2}+\ldots+a_Nz^{-N}}\times \frac{(1-\rho e^{jk\omega_0}z^{-1})(1-\rho e^{-jk\omega_0}z^{-1})}{(1-e^{jk\omega_0}z^{-1})(1-e^{-jk\omega_0}z^{-1})} \tag{4}$$

Alternately, we can write (4) in the other form demonstrated in (5) and its order is decreased to N-2 as

$$H_c''(z)=\frac{b_{0new}+b_{1new}z^{-1}+b_{2new}z^{-2}+\ldots+b_{Nnew}z^{-(N-2)}}{a_{0new}+a_{1new}z^{-1}+a_{2new}z^{-2}+\ldots+a_{Nnew}z^{-(N-2)}}. \tag{5}$$

If the number of the removed harmonics is p, the order of transfer function is N-2p.

From (5), the coefficients of both numerator and denominator can be derived from division or factorization of root extraction. In this paper, the root moment method is particularly applied to obtain them.

## 3. Root Moments

Root Moment presented by Isaac Newton is the one property of polynomial. Also he presented the well-known relationship called Newton Identities presented in [4] and [5].

Considering a linear phase FIR digital filter, its function is

$$H(z)=z^n+h_1z^{n-1}+h_2z^{n-2}+\ldots+h_n=\prod_{i=1}^{n}(z-r_i) \tag{6}$$

A set of parameter $S_m$ [3] is the function of the roots of polynomial, $H(z)$ which are known as root moments of the polynomial

$$S_m = r_1^m + r_2^m + \ldots + r_n^m = \sum_{i=1}^{n} r_i^m \,. \quad (7)$$

### 3.1 Iterative Estimation of Root Moments

From (6) we can write $H'(z) = (H(z)/(z - r_i))$ and let $H(r_i) = 0$, so $H'(z)$ can be written as

$$H'(z) = nz^{n-1} + (S_1 + nh_1)z^{n-2} + (S_2 + h_1 S_1 + nh_2)z^{n-3} + \ldots + (S_m + h_1 S_{m-1} + h_2 S_{m-2} + \ldots + nh_m)z^{n-m-1} + \ldots \quad (8)$$

After the differentiation of (7), we obtain

$$H'(z) = nz^{n-1} + (n-1)h_1 z^{n-2} + (n-2)h_2 z^{n-3} + \ldots + (n-m)h_m z^{n-m-1} + \ldots \quad (9)$$

The relative equation of (8) and (9) is known by identities and shown as the following

$$S_m + h_1 S_{m-1} + h_2 S_{m-2} + \ldots + mh_{m-n} = 0. \quad (10)$$

The Newton Identities are the relationship between root moments and polynomial coefficients. Here, the relationship between root moments and signal combination, which is product and division, will be considered.

**Product:** assume that $S_m^{f_1(z)}$ is root moments of polynomial $f_1(z)$, and $S_m^{f_2(z)}$ is that of polynomial $f_2(z)$. Finally, $S_m^{f(z)} = S_m^{f_1(z)} + S_m^{f_2(z)}$ is then that of polynomial $f(z)$, that is, $f(z) = f_1(z) \times f_2(z)$.

**Division:** alternatively, when $f(z) = f_1(z)/f_2(z)$, root moments of polynomial $f(z)$ is $S_m^{f(z)} = S_m^{f_1(z)} - S_m^{f_2(z)}$.

## 4. Specified Harmonics Removal of Digital Comb Filter by Root Moments

In the procedure of finding filter coefficients, we need only to calculate the numerator coefficients because the denominator coefficients can be derived from numerator coefficients and $\rho$. The algorithm is then shown in following steps:

1). From (4), root moments of $b_0 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_N z^{-N}$ is calculated as

$$S_m^n = 1 + \sum_{i=1}^{M} 2\cos(mk\omega_0) \quad (11)$$

so we obtain

$$S_1^n, S_2^n, S_3^n, \ldots, S_N^n \,. \quad (12)$$

Similarly, root moments of $(1 - e^{jk\omega_0} z^{-1})(1 - e^{-jk\omega_0} z^{-1})$ is calculated as

$$S_m^d = 2\cos(mk\omega_0) \,. \quad (13)$$

If we want to remove the harmonics at $k_1, k_2, \ldots, k_m$, the root moments become

$$S_m^d = 2\cos(mk_1\omega_0) + 2\cos(mk_2\omega_0) + \ldots + 2\cos(mk_m\omega_0) \quad (14)$$

Similarly, to remove DC component, (14) becomes

$$S_m^d = 1 + 2\cos(mk_1\omega_0) + 2\cos(mk_2\omega_0) + \ldots + 2\cos(mk_m\omega_0) \quad (15)$$

Thus, we achieve the new root moments as

$$S_1^d, S_2^d, S_3^d, \ldots, S_N^d. \quad (16)$$

2) The properties of root moments and signal division are considered, so the numerator root moments according to (5) can be calculated as

$$S_m^{n'} = S_m^n - S_m^d \,. \quad (17)$$

The new numerator root moments can be obtained by

$$S_1^{n'}, S_2^{n'}, S_3^{n'}, \ldots, S_N^{n'} \,. \quad (18)$$

3) From Newton Identities according to (10), the numerator coefficients in (5) are calculated, we have

$$b_m = (-S_m^{n'} - b_1 S_{m-1}^{n'} - b_2 S_{m-2}^{n'} - \ldots - b_{m-1} S_1^{n'})/m \,. \quad (19)$$

4) Thus, the denominator coefficients is given by

$$a_m = b_m \times \rho^m \,. \quad (20)$$

For instance,

$$a_2 = b_2 \times \rho^2 . \tag{21}$$

## 5. Experimental Results

In this paper, digital comb filter is designed so as to maintain DC signal and the 9th harmonic. The parameter is defined as $\omega_0 = 0.11\pi$ , $\rho = 0.999$ .

The frequency response of the digital comb filter that doesn't remove any harmonics is shown in Fig. 1. Its transfer function coefficients according to (2) are given by

$$b_0 = -b_{19} = 1.0000e+0 \quad , b_1 = -b_{18} = 8.1953e-1,$$
$$b_2 = -b_{17} = -7.5994e-2 \ , b_3 = -b_{16} = 3.1104e-2,$$
$$b_4 = -b_{15} = -1.7963e-2 \ , b_5 = -b_{14} = 1.2343e-2,$$
$$b_6 = -b_{13} = -9.4831e-3 \ , b_7 = -b_{12} = 7.9917e-3,$$
$$b_8 = -b_{11} = -7.0581e-3 \ , b_9 = -b_{10} = 6.6784e-3$$

$$a_0 = 1.0000e+0, \quad a_1 = 8.1871e-1, \quad a_1 = 8.1871e-1,$$
$$a_3 = 3.1011e-2, \quad a_4 = -1.7891e-2, \quad a_4 = -1.7891e-2,$$
$$a_6 = -9.4264e-3, \quad a_7 = 7.8565e-3, \quad a_7 = 7.8565e-3,$$
$$a_9 = 6.6185e-3, \quad a_{10} = -6.6119e-3, \quad a_{10} = -6.6119e-3,$$
$$a_{12} = -7.8173e-3, \quad a_{13} = 9.3606e-3, \quad a_{13} = 9.3606e-3,$$
$$a_{15} = 1.7695e-2, \quad a_{16} = -3.0610e-2, \quad a_{16} = -3.0610e-2,$$
$$a_{18} = -8.0490e-1, \quad a_{19} = -9.8117e-1 \quad a_{19} = -9.8117e-1$$

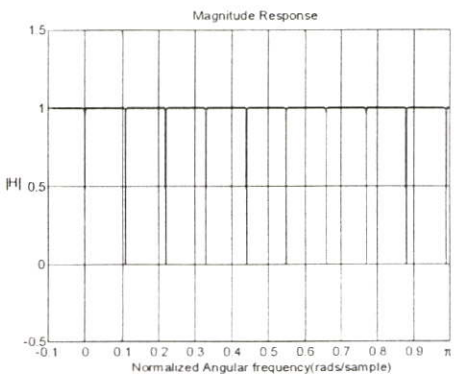



Fig. 1. The frequency response of the digital comb filter which $\omega_0 = 0.11\pi$ and $\rho$ =0.999.

To maintain DC signal and the 9th harmonic, it can be done by the following procedures:

1) According to (11), (15), and (17), the numerator root moments of the transfer function and the removal harmonic root moments are calculated, then we obtain the root moments of the new transfer function, that is

$$S_1^{n'} = 1.7948e-1, \quad , S_2^{n'} = -2.1724e+0, \ , S_3^{n'} = 1.6055e-1,$$
$$S_4^{n'} = -2.1436e+0, \ , S_5^{n'} = 1.2130e-1, \quad , S_6^{n'} = -2.0931e+0,$$
$$S_7^{n'} = 5.8346e-2, \quad , S_8^{n'} = -2.0160e+0, \ , S_9^{n'} = -3.5324e-2,$$
$$S_{10}^{n'} = -1.9021e+0, \ , S_{11}^{n'} = -1.7510e-1, \ , S_{12}^{n'} = -1.7274e+0,$$
$$S_{13}^{n'} = -4.0042e-1, \ , S_{14}^{n'} = -1.4219e+0, \ , S_{15}^{n'} = -8.4984e-1,$$
$$S_{16}^{n'} = -6.5954e-1, \ , S_{17}^{n'} = -2.5973e+0, \ , S_{18}^{n'} = 1.5206e+1,$$
$$S_{19}^{n'} = 3.7765e+0$$

2) The coefficients of transfer function maintaining DC signal and the 9th harmonic are calculated according to (19). Then

$$b_0 = b_{16} = 1.0000e+1, b_1 = b_{15} = -1.7948e-1,$$
$$b_2 = b_{14} = 1.1023e+0, b_3 = b_{13} = -2.4944e-1,$$
$$b_4 = b_{12} = 1.1530e+0, b_5 = b_{11} = -2.8637e-1,$$
$$b_6 = b_{10} = 1.1790e+0, b_7 = b_9 = -3.0304e-1,$$
$$b_8 = 1.1872e+0$$

Hence the denominator coefficients are calculated according to (20), its frequency response is shown in Fig. 2.

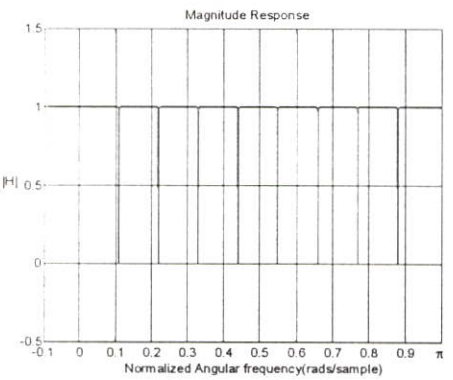



Fig. 2. The frequency response of the digital comb filter with $\omega_0 = 0.11\pi$ , $\rho = 0.999$ and the 9th harmonic and DC removed.

## 6. Conclusions

In case of which D is not an integer value, digital comb filter is applied to remove only some specified harmonics. The root moment method is one of algorithm to solve the mentioned problem, besides the traditional one used the fractional process of root extraction. The advantage of this algorithm is less complexity of computation when compared with the old one. Especially, the case of the harmonics removal performs with the old filter because we can use the old root moments of the transfer function, and calculate the root moments of the removed term only. Therefore, the transfer function coefficients can be calculated from iteration process.

**References**

[1] S.-C. Pei and C.-C. Tseng, "A Comb Filter-Design Using Fractional-Sample Delay," IEEE Trans. on Circuits and Systems-II: and Digital Signal Processing, vol. 45, no 6, pp. 649-653, Jun. 1998.

[2] S. J. Orfanidis, Introduction to Signal Processing, Prentice-Hall, pp.253-258, 1996.

[3] T.Stathaki, "Root Moments:a digital signal-processing perspective," IEE Proc.-Vis. Image Signal processing, vol.145, no. 4, pp. 293-302, Aug. 1998.

[4] T. I.lakkso, V. Valimaki, M. Karjalainen, and U.K. Laine, Splittinging the Unit Delay, IEEE Signal Processing Magazine, pp. 30-60, Jan. 1996.

[5] T.Stathaki and I. Fotinopoulos, " Equiripple Minimum Phase FIR Filter Design From Linear Phase Systems Using Root Moments," IEEE Trans. on Circuit and Systems-II, Analog and Digital Signal processing, vol. 48, no. 6., pp. 580-587, Jun. 2001.

## ข. โปรแกรมการจำลองผลการทำงานโดยใช้ Matlab

ประกอบไปด้วยโปรแกรมในการจำลองการทำงานโดยใช้ Matlab 4 โปรแกรมดังต่อไปนี้

1. โปรแกรม FPGResponse :ใช้ในการพลอทผลตอบสนองความถี่ (ขนาด) , ผลตอบสนองเฟส, เฟสดีเลย์, กรุ๊พดีเลย์
2. โปรแกรม AllpassComb : ใช้ในการออกแบบคอมบ์ฟิลเตอร์โดยใช้การออกแบบออลพาสฟิลเตอร์ (2.3.2)
3. โปรแกรม FIRComb : ใช้ในการออกแบบคอมบ์ฟิลเตอร์โดยใช้การออกแบบเอฟไออาร์ฟิลเตอร์ (2.3.1)
4. โปรแกรม RootMoments : ใช้ในการออกแบบคอมบ์ฟิลเตอร์และกำจัดฮาร์โมนิคที่ต้องการโดยโมเมนต์ของราก (4.3)

**การกำหนดตัวแปรต่างๆ**

**โปรแกรม FPGResponse**

a: สัมประสิทธิ์เทอมเศษของฟิลเตอร์

b: สัมประสิทธิ์เทอมส่วนของฟิลเตอร์

m: ผลตอบสนองความถี่ (ขนาด)

angH: ผลตอบสนองเฟสของฟิลเตอร์

phdelay: ผลตอบสนองเฟสดีเลย์ของฟิลเตอร์

**โปรแกรม AllpassComb, โปรแกรม FIRComb**

w0: ความถี่หลักมูล ( $2\pi f_0 / Fs$ )

r :ค่าคงที่เทอมเศษ (มักกำหนดให้เท่ากับ1)

R : ค่าคงที่เทอมส่วน

N : คือออเดอร์ของฟิลเตอร์

nom : คือสัมประสิทธิ์เทอมเศษ (ออลพาสฟิลเตอร์)

denom: คือสัมประสิทธิ์เทอมส่วน (ออลพาสฟิลเตอร์)

nom_new : คือสัมประสิทธิ์เทอมเศษ (คอมบ์ฟิลเตอร์)

denom_new: คือสัมประสิทธิ์เทอมส่วน (คอมบ์ฟิลเตอร์)

**โปรแกรม RootMoments**

w0: ความถี่หลักมูล ( $2\pi f_0 / Fs$ )

K: คือฮาร์โมนิคที่ต้องการกำจัดอยู่ในรูปแบบของเมตริกซ์ขนาด 1*x เมื่อ x คือจำนวนฮาร์โมนิคที่ต้องการกำจัด ดังเช่น [1 2]

S: คือโมเมนต์ของรากของคอมบ์ฟิลเตอร์

S_N: คือโมเมนต์ของรากของฟิลเตอร์ผลลัพธ์

a: สัมประสิทธิ์เทอมเศษของฟิลเตอร์

b: สัมประสิทธิ์เทอมส่วนของฟิลเตอร์

```
function[m,angH,phdelay]=Fpgresponse(a,b);
% To plot response
% figure1 : plot magnitude response
% figure2 : plot phase response, phase delay response
% figure3 : plot group delay response

m=0:length(a)-1;l=0:length(b)-1;
K=1600;
k=-1600:1:1600;
w=pi*k/K;
num=a*exp(-j*m'*w);
den=b*exp(-j*l'*w);
H=num./den;
magH=abs(H);
magHDb=20*log(magH+eps/max(magH));

figure(1);
plot(w/pi,magH);grid;axis([-1,1,-0.5,1.5]);
xlabel('Normalized Angular frequency(rads/sample)');ylabel('|H|');
title('Magnitude Response');

figure(2);
[h,w]=freqz(a,b,10000);
m=abs(h);
m=(m(2:1:10000));
angH=angle(h);
angH=unwrap(angH);
w=(w(2:1:10000));
angH=(angH(2:1:10000));
subplot(2,1,1);
plot(w/pi,angH);
grid;axis([0,1,-10*pi,pi]);
```

```
xlabel('frequency in pi unit');ylabel('Phase');
title('Phase Response');
for i=1:9999
 phdelay(i)=-1*angH(i)/w(i);
  fre(i)=w(i)/pi;
end;

subplot(2,1,2);
plot(fre,phdelay);
grid;xlabel('frequency in pi unit');
ylabel('phase delay (s)');
title('Phase Delay');
[gd,w]=grpdelay(a,b,3000);

figure(3);
plot(w/pi,gd);grid;
xlabel('frequency in pi unit');
ylabel('group delay(s)');
title('Group Delay');
%----------------------------------------------------------End----------------------------------------------------------%

 function [nom,denom,nom_new,denom_new]=allpassComb(w0,r,R,alpha,N);
%To calculate allpass filter,comb filter coefficients
% allpass filter transfer function =nom/denom (|H(w)|=1,phase delay=2*pi/w0)
% comb filter transfer function = nom_new/denom_new
% =(denom-r^D*nom)/(denom-q^D-nom)

D=(2*pi/w0);
syms w;
beta=-1/2*((-1*D*w)+(N*w));

for i=1:N
```

```
  b(i,1)=sin(beta+i*w);
end;

bT=transpose(b);

sinbeta=sin(beta);
sum1=b*bT;
Q=int(sum1,w,0,alpha*pi);
sum2=b*sinbeta;
p=-1*int(sum2,w,0,alpha*pi);
sum3=sinbeta^2;
c=int(sum3,w,0,alpha*pi);

M=floor(pi/w0);
for k=1:M
  for i=1:N
    w01=k*w0;
    beta_new=-1/2*((-1*D*w01)+(N*w01));
    b_new(i,1)=sin(beta_new+i*w01);
    C(i,k)=b_new(i,1);
    f(1,k)=-1*(sin(beta_new));
  end;
end;
f(1,2)=0;
C_new=transpose(C);
f_new=transpose(f);
Q=numeric(Q);
d=inv(Q);
p=numeric(p);
result1=d*p;
result2=d*C*inv(C_new*d*C);
result3=((C_new*d*p)-f_new);
```

```
resultall=result1-(result2*result3);


nom(N+1)=1;
for i=1:N
  nom(i)=resultall(N-i+1);
end;
denom(1)=1;
for i=2:N+1
  denom(i)=resultall(i-1);
end;
[m,angH,phdelay]=Fpgresponse(nom,denom);


nom_new=denom-nom;
denom_new=denom-(R^D*(nom));


m=0:length(nom_new)-1;l=0:length(denom_new)-1;
K=3000;
k=-3000:1:3000;
w=pi*k/K;
num=nom_new*exp(-j*m'*w);
den=denom_new*exp(-j*l'*w);
H=num./den;
magH=abs(H);



figure(5);
subplot(2,1,1);
plot(w/pi,magH);grid;axis([-1,1,-0.5,1.5]);
xlabel('Normalized Angular Frequency(rads/sample)');ylabel('|H|');
title('Magnitude Response');
%-------------------------------------------------End-------------------------------------------------%
```

```
function[b,a]=FIRComb(w0,r,R,alpha,N);
%To calculate FIR filter,comb filter coefficients
syms w;
D=2*pi/w0;
for k=0:N
  for i=0:N
    minus=i-k;
    sum1(k+1,i+1)=cos(minus*w);
  end;
end;
Q=2*int(sum1,w,0,alpha*pi);
Q=numeric(Q);

for k=0:N
  sum2(k+1,1)=cos(((-1*D)+k)*w);
end;

p=2*int(sum2,w,0,alpha*pi);
p=numeric(p);

Qinv=inv(Q);
M=(floor(pi/w0));
w01=0*w0;
for i=0:N
  e_new(i+1,1)=cos(i*w01)-j*sin(i*w01);
  C(i+1,1)=real(e_new(i+1,1));
  f(1,1)=1;
end;

for k=1:2*M
  test=rem(k,2);
  test1=ceil(k/2);
```

```
for i=0:N
  w01=test1*w0;
  e_new(i+1,k+1)=cos(i*w01)-j*sin(i*w01);
  if test==1
    C(i+1,k+1)=real(e_new(i+1,k+1));
    f(1,k+1)=cos(D*w01);
   elseif test==0
     C(i+1,k+1)=imag(e_new(i+1,k+1));
     f(1,k+1)=-1*sin(D*w01);
   end;
  end;
end;

C_new=transpose(C);
f_new=transpose(f);

result1=Qinv*p;
result2=Qinv*C*inv(C_new*Qinv*C);
result3=(C_new*Qinv*p)-f_new;
h=result1-(result2*result3);

figure(1);
[H,W]=freqz(h,1,800);
m=abs(H);
plot(W/pi,m);

b(1)=1-h(1);
a(1)=1-((R^D)*h(1));
for i=2:N+1
  b(i)=-1*h(i);
  a(i)=-1*((R^D)*h(i));
end;
```

```
  figure(2);
  [H,W]=freqz(b,a,800);
  m=abs(H);
  plot(W/pi,m,'r');
  b(1)=1-(r^D)*h(1);
  for i=2:N+1
    b(i)=-1*(r^D)*h(i);
  end;
  [H,W]=freqz(b,a,800);
  m=abs(H);
  plot(W/pi,m);axis([-1,1,-0.5,1.5]);
%----------------------------------------------------End----------------------------------------------------%

function [S,S_N,b,a]=RootMoments(w0,K);
%Calculate Comb Filter Coefficient by Root Moments
% K=Cancelled Harmonic

R=0.999;
M=floor(pi/w0);
number=length(K);
for m=1:(2*M+1)%Calculate Root Moments of Comb Filter
  S(m,1)=1;
end;
for m=1:(2*M+1)
  for k=2:M+1
    S(m,k)=2*cos(m*(k-1)*w0);
  end;
end;

for m=1:(2*M+1)%Calculate Root Moments of Cancelled harmonic
  S_N(m)=0;
```

```
  S_M(m)=0;
  for k=1:M+1;
  S_N(m)=S_N(m)+S(m,k);
        end;
  for k=1:number
    S_M(m)=S_M(m)+S(m,K(k)+1);
  end;
  S_N(m)=S_N(m)-S_M(m);%Calculate Root Moments of Result Comb Filter
end;
b(1)=1;
a(1)=1;
for m=1:(2*M+1)
  b(m+1)=0;
  for k=1:m
    b(m+1)=b(m+1)-b(k)*S_N(m-k+1);
  end;
        b(m+1)=b(m+1)/m;
  a(m+1)=b(m+1)*R^m;
end;
[m,angH,phdelay]=Fpgresponse(b,a);
%----------------------------------------------------End-----------------------------------------------------%
```

## ค. โปรแกรมการคำนวณสัมประสิทธิ์และฟิลเตอร์ (TMS320C31)

ประกอบไปด้วยโปรแกรม 4 โปรแกรมดังต่อไปนี้

1. โปรแกรม AICCOM31.ASM :ใช้ในการจัดการอินพุท เอาท์พุท อินเทอร์รัพ พอลลิ่งโดย AIC (Analog Interface Circuit)
2. โปรแกรม RMMCal.ASM: ใช้ในการคำนวณสัมประสิทธิ์โดยโมเมนต์ของรากและสร้างฟิลเตอร์
3. โปรแกรม RMMCOEF.COF : ไฟล์สัมประสิทธิ์ของ RMMCal.ASM
4. โปรแกรม INVF.ASM :ใช้ในการหารหาค่า $\frac{1}{n}$

```
*AICCOM31.ASM - AIC COMMUNICATION ROUTINES - POLLING OR INTERRUPT
        .data           ;assemble into data section
PBASE   .word   808000h     ;peripheral base address
SETSP   .word   0E970300h   ;serial port set-up data
ATABLE  .word   AICSEC      ;SP0 AIC init table address
        .text           ;assemble into text section
AICSET      PUSH    AR0         ;save AR0
            PUSH    AR1         ;save AR1
            PUSH    R0          ;save R0
            PUSH    R1          ;save R1
            LDI     @PBASE,AR0  ;AR0 -> 808000h
            LDI     1,R0        ;timer CLK=H1/2*(AIC master CLK)
            STI     R0,*+AR0(28h) ;timer period reg(TCLK0=6.25 MHZ)
            LDI     03C1h,R0    ;init timer global register
            STI     R0,*+AR0(20h) ;reset timer
            LDI     62h,IOF     ;AIC reset = 0
            LDI     @ATABLE,AR1 ;AR1 -> AIC init data
            RPTS    99          ;repeat next instr 100 times
            NOP                 ;keep IOF low for a while
            LDI     111h,R0     ;X & R port control register data
            STI     R0,*+AR0(42h) ;FSX/DX/CLKX=SP operational pins
            STI     R0,*+AR0(43h) ;FSR/DR/CLKR=SP operational pins
            LDI     @SETSP,R0   ;RESET->SP:16 bits,ext clks,std mode
            STI     R0,*+AR0(40h) ;FSX=output&INT enable SP global reg
            LDI     0,R0        ;R0 = 0
            STI     R0,*+AR0(48h) ;clear serial port XMIT register
            OR      06h,IOF     ;bring AIC out of reset
            LDI     02h,RC      ;RC=3 to transmit 4 values
            RPTB    SECEND      ;repeat 4 data transmit of sec com
            CALL    TWAIT       ;wait for data transmit
            LDI     03h,R0      ;value for secondary XMIT request
            STI     R0,*+AR0(48h) ;secondary XMIT request to AIC
```

```
                CALL   TWAIT        ;wait for data transmit
                LDI    *AR1++(1),R0  ;AR1 -> next AIC init data
SECEND  STI    R0,*+AR0(48h)  ;DTR = current AIC data
                POP    R1           ;restore R1
                POP    R0           ;restore R0
                POP    AR1          ;restore AR1
                POP    AR0          ;restore AR0
                RETS                ;return from subroutine
AICSET_I                ;---CONFIG FOR INTERRUPT ----------
                CALL   AICSET       ;call AICSET routine
                LDI    0h,IF        ;clear IF register
                OR     10h,IE       ;enable EXINT0 CPU interrupt
                OR     2000h,ST     ;global interrupt enable
                RETS                ;return from subroutine
;--------------------TRANSMIT WAIT ROUTINE-------------------------
TWAIT           PUSH   AR0          ;save AR0
                PUSH   R0           ;save R0
                LDI    @PBASE,AR0   ;AR0 -> 0808000h
TW1     LDI    *+AR0(40h),R0  ;R0=content of SP global control reg
                AND    02h,R0       ;see if transmit buffer is ready
                BZ     TW1          ;if not ready, try again
                POP    R0           ;restore R0
                POP    AR0          ;restore AR0
                RETS                ;return from subroutine
;--------------------AIC TRANSFER ROUTINE-------------------------
AICIO_I         LDI    R7,R6        ;copy output to modify for AIC
                LSH    2,R6         ;two LSB must=0 for primary AIC comm
IO              PUSH   AR0          ;save AR0
                LDI    @PBASE,AR0   ;AR0 -> 0808000h
                STI    R6,*+AR0(48h)  ;DTR = next data for AIC D/A
                LDI    *+AR0(4Ch),R6  ;R6 = DRR data from AIC A/D
                LSH    16,R6        ;left shift for sign extension
```

```
                ASH   -18,R6      ;right shift keeping sign
                POP   AR0         ;restore AR0
                RETS              ;return from subroutine
;-------------------AIC POLLING ROUTINE-------------------------
AICIO_P CALL  TWAIT       ;wait for data to be transferred
                CALL  AICIO_I     ;call AIC transfer routine
                RETS              ;return from subroutine
SW_IO           PUSH  AR0         ;save AR0
                LDI   @PBASE,AR0  ;AR0 -> 0808000h
                LDI   R7,R6       ;copy output to modify for AIC
                LSH   2,R6        ;prepare for secondary AIC com
                OR    03h,R6      ;set two LSB for secondary com
                CALL  TWAIT       ;wait for data to be transferred
                CALL  IO          ;call AIC transfer routine
                CALL  TWAIT       ;wait for data to be transferred
                STI   R1,*+AR0(48h) ;DTR = next data for AIC control
                POP   AR0         ;restore AR0
                RETS              ;return from subroutine
;SUBROUTINES FOR PRIMARY OR AUXILIARY INPUT
IOPRI           PUSH  R1          ;save R1
                LDI   063h,R1     ;load secondary com data into R1
                CALL  SW_IO       ;call IO routine to switch inputs
                POP   R1          ;restore R1
                RETS              ;return from subroutine
IOAUX           PUSH  R1          ;save R1
                LDI   073h,R1     ;load secondary com data into R1
                CALL  SW_IO       ;call IO routine to switch inputs
                POP   R1          ;restore R1
                RETS              ;return from subroutine
```

```
*RMMCAL,Calculate Comb filter Coefficient in 4.3,Implement Comb Fiter(Direct Form I)
        .start ".d ata",0x809C00  ;starting address for dat
        .start ".text",0x809900  ;starting address for text
        .include "RMMCOEF.cof"
        .include "invf.asm"
        .include "aiccom31.asm"
        .entry BEGIN
        .text
BEGIN   LDP   S_ADDR
        LDI   @S_ADDR,AR0
        LDI   @SN_ADDR,AR1    ;AR1 POINT TO SN
        LDI   0,IR0
CALSN   LDI   ORDER_C,RC      ;RC =ORDER_C(COLUMN ORDER)
        RPTB  LOOPA
        ADDF3  *AR0++,*AR1,R1
LOOPA   STF   R1,*AR1
        LDF   *++AR1,R1
        ADDI  ADD,IR0
        CMPI  ORDER_R,IR0
        BNZ   CALSN         ; LOOP TO CALCULATE S_N
        LDI   @K_ADDR,AR2     ; AR2 POINT TO K_ADDR
        LDI   *AR2,R1
        CMPI  ORDER_R+1,R1
        BZ    CALS_RE       ; IF K=ORDER_R1 CALCULATE S_RE
        LDI   @SD_ADDR,AR1
        LDI   0,IR0
        LDI   0,IR1
CALSD   LDI   NUMBER-1,RC
        RPTB  LOOPB
        LDI   IR0,R1
        MPYI  ORDER_C+1,IR0
        LDI   IR0,IR1
```

```
            LDI   R1,IR0
            LDI   @S_ADDR,AR0    ;Calculate SD
            LDF   *++AR0(IR1),R1
            LDI   *AR2++,R1
            LDI   R1,IR1
            ADDF3 *++AR0(IR1),*AR1,R1
LOOPB       STF   R1,*AR1
            LDF   *++AR1,R1
            ADDI  1,IR0
            LDI   @K_ADDR,AR2
            CMPI  ORDER_R,IR0
            BNZ   CALSD
CALS_RE     LDI   @SN_ADDR,AR0   ;Calculate Result Root Moment
            LDI   @SD_ADDR,AR1
            LDI   @S_RADDR,AR2
            LDI   ORDER_R-1,RC
            RPTB  LOOPC
            SUBF3 *AR1++,*AR0++,R1
LOOPC       STF   R1,*AR2++
            LDI   @S_RADDR,AR0   ;Calculate b
            LDI   @B_ADDR,AR1
            LDI   @A_ADDR,AR2
            LDI   @Q_MADDR,AR4
            LDI   @B_ADDR,AR5
            LDI   @RD_ADDR,AR3
            LDI   0,IR0
            LDI   1,IR1
            LDF   Q,R1
            STF   R1,*AR4
CALB        LDF   *++AR0(IR0),R1
            LDI   *AR5++,R1
            LDI   IR0,RC
```

```
            RPTB  LOOPD
            MPYF3 *AR0--,*AR1++,R1
            LDF   0,R4
            SUBF3 R1,R4,R1
            ADDF3 R1,*AR5,R1
LOOPD       STF   R1,*AR5
            CMPI  0,IR0
            BZ    CALBM
            LDF   Q,R1
            MPYF3 *AR4,R1,R1
            STF   R1,*AR4
CALBM       ADDI  1,IR0
            ADDI  1,IR1
            LDF   *AR3,R1
            ADDF  1,R1
            STF   R1,*AR3
            LDF   *AR3,R0
            CALL  INVFS
            MPYF3 *AR5,R0,R1     ;Calculate b
            STF   R1,*AR1
            MPYF3 *AR1,*AR4,R1   ;Calculate a
            STF   R1,*++AR2
            LDI   @S_RADDR,AR0
            LDI   @B_ADDR,AR1
            CMPI  ORDER_R,IR0
            BNZ   CALB           ;Calculate Y(z)
            LDI   @K_ADDR,AR2
            LDI   @OR_F,AR4
            LDI   *AR2,IR0
            CMPI  ORDER_R+1,IR0
            BNZ   CAL_O
            LDI   0,IR0
```

```
                STI   IR0,*AR4
                BR    CAL_OH
CAL_O           LDI   NUMBER-1,RC
                RPTB  LOOPO
                LDI   *AR2++,IR0
                LDI   *AR4,R2
                CMPI  0,IR0
                BNZ   NDC
                ADDI  1,R2
                BR    LOOPO
NDC             ADDI  2,R2
LOOPO           STI   R2,*AR4
CAL_OH          LDI   @RD_ADDR,AR3 ;
                LDI   ORDER_R,IR1
                LDI   *AR4,IR0
                SUBI  IR0,IR1
                SUBI  1,IR1
                STI   IR1,*AR3      ;AR3 POINT TO ORDER OF FILTER
                ;LDI  @X_ADDR,AR2
                ;LDI  @OUT_ADDR,AR4
                CALL  AICSET
IIR             LDI   @B_ADDR,AR0   ;AR0 points to coefficients address
                LDI   @DLY_ADDR,AR1 ;AR1 points to addr of delay samples
                LDF   *AR0++,R0
                CALL  AICIO_P
                FLOAT R6,R3
                ;LDF   *AR2++,R3
                LDF   R3,R2
                LDI   *AR3,R1
                LDI   R1,RC
                RPTB  LOOPF
                LDF   *AR1,R4
```

```
        STF    R2,*AR1++
        MPYF3  R4,*AR0++,R0
        ADDF3  R0,R3,R3
LOOPF   LDF    R4,R2
        LDI    @A_ADDR,AR0
        LDF    *AR0++,R0
        LDI    *AR3,R1
        LDI    R1,RC
        RPTB   LOOPG
        LDF    *AR1,R4
        STF    R2,*AR1++
        MPYF3  R4,*AR0++,R0
        SUBF3  R0,R3,R3
LOOPG   LDF    R4,R2
        LDI    *AR3,R1
        ADDI   1,R1
        LDI    R1,IR0
        STF    R3,*--AR1(IR0)
        ;STF   R3,*AR4++
        FIX    R3,R7
        BR     IIR
```

*RMMCOEF- RMMCAL COEFFICIENT

.data

S .float 1.0000e+000, 1.8818e+000,1.5410e+000 ,1.0181e+000 ,3.7476e-001

.float -3.1287e-001, -9.6351e-001 ,-1.5002e+000, -1.8596e+000,-1.9990e+000

.float 1.0000e+000 ,1.5410e+000 ,3.7476e-001,-9.6351e-001,-1.8596e+000

.float -1.9021e+000, -1.0717e+000 ,2.5067e-001,1.4579e+000 ,1.9961e+000

.float 1.0000e+000 ,1.0181e+000,-9.6351e-001,-1.9990e+000, -1.0717e+000

.float 9.0798e-001,1.9961e+000 ,1.1242e+000,-8.5156e-001,-1.9911e+000

.float 1.0000e+000 ,3.7476e-001,-1.8596e+000,-1.0717e+000 ,1.4579e+000

.float 1.6180e+000 ,-8.5156e-001,-1.9372e+000 ,1.2558e-001,1.9842e+000

.float 1.0000e+000, -3.1287e-001,-1.9021e+000 ,9.0798e-001,1.6180e+000

.float -1.4142e+000 ,-1.1756e+000 ,1.7820e+000 ,6.1803e-001,-1.9754e+000

.float 1.0000e+000,-9.6351e-001,-1.0717e+000,1.9961e+000,-8.5156e-001

.float -1.1756e+000,1.9842e+000 ,-7.3625e-001, -1.2748e+000,1.9646e+000

.float 1.0000e+000,-1.5002e+000 ,2.5067e-001,1.1242e+000,-1.9372e+000

.float 1.7820e+000 ,-7.3625e-001,-6.7748e-001,1.7526e+000,-1.9518e+000

.float 1.0000e+000,-1.8596e+000 ,1.4579e+000,-8.5156e-001,1.2558e-001

.float 6.1803e-001 ,-1.2748e+000 ,1.7526e+000,-1.9842e+000,1.9372e+000

.float 1.0000e+000,-1.9990e+000 ,1.9961e+000,-1.9911e+000,1.9842e+000

.float -1.9754e+000 ,1.9646e+000,-1.9518e+000 ,1.9372e+000,-1.9206e+000

.float 1.0000e+000, -1.9021e+000 ,1.6180e+000, -1.1756e+000 ,6.1803e-001

.float -4.8998e-015 ,-6.1803e-001,1.1756e+000,-1.6180e+000,1.9021e+000

.float 1.0000e+000, -1.5803e+000 ,4.9738e-001,7.9430e-001, -1.7526e+000

.float 1.9754e+000 , -1.3691e+000 ,1.8822e-001,1.0717e+000, -1.8818e+000

.float 1.0000e+000, -1.0717e+000 ,-8.5156e-001,1.9842e+000, -1.2748e+000

.float -6.1803e-001 ,1.9372e+000 ,-1.4579e+000,-3.7476e-001,1.8596e+000

.float 1.0000e+000, -4.3629e-001,-1.8097e+000,1.2258e+000,1.2748e+000

.float -1.7820e+000, -4.9738e-001 ,1.9990e+000,-3.7476e-001, -1.8355e+000

.float 1.0000e+000 ,2.5067e-001,-1.9372e+000,-7.3625e-001,1.7526e+000

.float 1.1756e+000 ,-1.4579e+000,-1.5410e+000,1.0717e+000,1.8097e+000

.float 1.0000e+000 ,9.0798e-001,-1.1756e+000,-1.9754e+000 ,-6.1803e-001

.float 1.4142e+000,1.9021e+000 ,3.1287e-001,-1.6180e+000,-1.7820e+000

```
        .float  1.0000e+000 ,1.4579e+000,1.2558e-001, -1.2748e+000,-1.9842e+000
        .float -1.6180e+000,-3.7476e-001,1.0717e+000 , 1.9372e+000,1.7526e+000
        .float  1.0000e+000,1.8355e+000 ,1.3691e+000, 6.7748e-001, -1.2558e-001
        .float  -9.0798e-001,-1.5410e+000,-1.9206e+000, -1.9842e+000, -1.7215e+000
        .float  1.0000e+000,1.9961e+000,1.9842e+000, 1.9646e+000,1.9372e+000
        .float  1.9021e+000,1.8596e+000,1.8097e+000 ,1.7526e+000 ,1.6887e+000
        .float  1.0000e+000 ,1.9206e+000,1.6887e+000 ,1.3226e+000 ,8.5156e-001
        .float 3.1287e-001 ,-2.5067e-001,-7.9430e-001 ,-1.2748e+000 ,-1.6542e+000
X       .float 1,0,2,1,4,2,0,1,1.5e+0,5.0e-1,2.0e-1,0,1,2.0e-1,4.0e+0
        .float 1,5,5.0e-2,1,0,2,1,0,1,1,2,3,4,6,2,0,1,2,1
AICSEC.word  162Ch,1h,4892h,63h ;AIC config data, Fs = 8 kHz
K       .int 20
QM      .float 0.0
RD      .float 0.0
ORF     .int  0
NUMBER  .set  5
ORDER_R .set  19;2M+1
ORDER_C .set  9;M
Q               .set  9.9900e-001
ADD         .set  1
RD_ADDR  .word RD
S_ADDR     .word  S
K_ADDR    .word  K
SN_ADDR  .word  SN
SD_ADDR  .word  SD
S_RADDR  .word  S_R
B_ADDR    .word  B
A_ADDR    .word  A
DLY_ADDR .word  DLY
X_ADDR      .word  X
OUT_ADDR .word  OUT
Q_MADDR  .word  QM
```

```
OR_F    .word ORF
        .brstart "SN_BUFF",20
SN      .sect   "SN_BUFF"
        .loop   20
        .float  0.0
        .endloop
        .brstart "SD_BUFF",20
SD      .sect   "SD_BUFF"
        .loop   20
        .float  0.0
        .endloop
        .brstart "S_R_BUFF",20
S_R     .sect   "S_R_BUFF"
        .loop   20
        .float  0.0
        .endloop
        .brstart "B_BUFF",22
B       .sect   "B_BUFF"
        .float  1.000e+000
        .loop   21
        .float  0.0
        .endloop
        .brstart "A_BUFF",22
A       .sect   "A_BUFF"
        .float  1.000e+000
        .loop   21
        .float  0.0
        .endloop
        .brstart "DL_BUFF",50
DLY     .sect   "DL_BUFF"
        .loop   50
        .float  0.0
```

```
        .endloop
        .brstart "OUT_BUFF",10
OUT     .sect   "OUT_BUFF"
        .loop   40
        .float  0.0
        .endloop
                                ;--------------------End--------------------------;
*INVF.ASM – Division Routine , Find 1/n
                        .text
          INVFS:  LDF R0,R3
                        ABSF R0,R0
                         PUSHF R0
                        POP R1
                        ASH -24,R1
                        NEGI R1,R1
                        SUBI 1,R1
                        ASH 24,R1
                        PUSH R1
                        POPF R1
                        MPYF3 R1,R0,R2
                        SUBRF 2.0,R2
                        MPYF R2,R1
                        MPYF3 R1,R0,R2
                        SUBRF 2.0,R2
                        MPYF R2,R1
                        MPYF3 R1,R0,R2
                         SUBRF 2.0,R2
                        MPYF R2,R1
                        MPYF3 R1,R0,R2
                        SUBRF 2.0,R2
                        MPYF R2,R1
                        MPYF3 R1,R0,R2
```

```
SUBRF 2.0,R2
MPYF R2,R1
MPYF3 R1,R0,R2
SUBRF 2.0,R2
MPYF R2,R1
RND  R1,R1
MPYF3 R1,R0,R2
SUBRF 1.0,R2
MPYF R1,R2
ADDF R2,R1
RND R1,R0
NEGF R0,R2
LDF R3,R3
LDFN R2,R0
RETS
```

# ประวัติผู้เขียน

นางสาวเพ็ญฤดี ตั้งสมวรพงษ์ เกิดเมื่อวันที่ 9 พฤศจิกายน 2518 ที่อำเภอนางรอง จ.บุรีรัมย์ จบการศึกษาวิศวกรรมศาสตร์บัณฑิต(วิศวกรรมเทคโนโลยีการวัดคุมทางอุตสาหกรรม) จากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2540

ปี พ. ศ. 2540-2543 ทำงานในตำแหน่งผู้ปฏิบัติงานสอบเทียบ(ไฟฟ้า) ห้องปฏิบัติการสอบเทียบ บริษัทแสงชัยมิเตอร์ จำกัด ปัจจุบันเป็นอาจารย์พิเศษ สถาบันเทคโนโลยีราชมงคล วิทยาเขตพาณิชยการพระนคร