

การออกแบบวงจรกรองความถี่ต่ำด้วยเวฟดิสครีทฟิลเตอร์
Design of The Low Pass Type Wave Discrete Filters

ชวลิต เบญจางคประเสริฐ
Chawalit Benjangkprasert

วิทยานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2533

ISBN 974-8155-44-7

การออกแบบวงจรกรองความถี่ต่ำด้วยเวฟดิสครีทฟิลเตอร์
Design of The Low Pass Type Wave Discrete Filters

ชวลิต เบนจางคประเสริฐ
Chawalit Benjangkprasert

อาจารย์ที่ปรึกษา
ผู้ช่วยศาสตราจารย์ ดร.กนก เชนจิระพงษ์เวช
Advisor
Asst.Prof. Dr.Kanok Janchitrapongvej

วิทยาลัยวิศวกรรมศาสตร์มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

สาขาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2533

ISBN 974-8155-44-7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
แบบฟอร์มการให้คะแนนการสอบวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การออกแบบวงจรกรองความถี่ต่ำด้วยเวฟดิสครีทฟิลเตอร์
Design of The Low Pass Type Wave Discrete Filters
ชื่อนักศึกษา นายชวลิต เบญจางคประเสริฐ รหัสประจำตัว 31126-003
หลักสูตร วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชา วิศวกรรมไฟฟ้า
ภาควิชา วิศวกรรมไฟฟ้า
อาจารย์ผู้ควบคุมวิทยานิพนธ์ ผศ.ดร.กนก เจนจิระพงศ์เวช

ชื่ออาจารย์ผู้ควบคุมการสอบ	ลายมือชื่อ
ผศ.ดร.กนก เจนจิระพงศ์เวช	
ผศ.วิชัย สุรพัฒน์	
รศ.ดร.พุฒศักดิ์ ชิวสุวิทย์	
รศ.ดร.วัลลภ สุระกาศพลธร	
รศ.วิพันธุ์ ปรีชาพานิช	

ค่าระดับคะแนนรวมทั้ง เป็น เอกฉันท์จากคณะกรรมการสอบ G (GOOD)
วัน/เดือน/ปี ที่สอบ 9 พฤษภาคม 2534 เวลา 10.00 น.
สถานที่สอบ ห้องประชุมภาควิชาอิเล็กทรอนิกส์ (B-312) ตึกภาควิชาอิเล็กทรอนิกส์

บัณฑิตวิทยาลัยรับรองแล้ว



รักษาราชการแทนคณบดีบัณฑิตวิทยาลัย

วันที่...17...เดือน...พ.ค.....พ.ศ. ๒๕๓๔..

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์ การออกแบบวงจรกรองความถี่ต่ำด้วยเวฟดิสครีทฟิลเตอร์
 นักศึกษา นายชวลิต เหมจุอาจคประเสริฐ
 อาจารย์ที่ปรึกษา ผศ.ดร.กนก เจริญพงศ์เวช
 ระดับการศึกษา วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า
 ปีการศึกษา 2533

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้ได้เสนอการออกแบบวงจรเวฟดิสครีทฟิลเตอร์แบบคอม โพลิต โพลีโนเมียลชนิดวงจรกรองความถี่ต่ำบัตเตอร์วีท-อูลตราสเฟียริคัล โพลีโนเมียล โดยทรานส์เฟอร์ฟังก์ชันของวงจรให้คุณสมบัติของผลตอบสนองต่อความถี่เปลี่ยนแปลงอยู่ระหว่างบัตเตอร์วีทกับอูลตราสเฟียริคัลฟังก์ชัน ทำให้สามารถกำหนดผลตอบสนองทางขนาดตามที่ต้องการได้ โดยโครงสร้างของวงจรเวฟดิสครีทฟิลเตอร์สามารถสร้างขึ้นจากวงจรต้นแบบที่เป็นวงจรพาสซีฟ LC โดยเฉพาะอย่างยิ่งวงจรดังกล่าวจะให้ค่าความไวต่ำ และจากโครงสร้างดังกล่าวสามารถทำการสร้างวงจรจริงขึ้นโดยใช้วงจรอะนาล็อกแบบขนานกับอะนาล็อกแบบอนุกรมเป็นหลัก ซึ่งทำหน้าที่เป็นวงจรรวมและวงจรคุมต่อร่วมกับวงจรแชนเนลและไฮลด์ โดยทำหน้าที่เป็นดีเลย์ยูนิต จากผลการทดสอบวงจร โดยการป้อนสัญญาณนาฬิกา และทำการวัดค่าผลตอบสนองทางขนาดนั้น ให้ผลใกล้เคียงกับทฤษฎีการคำนวณ

Thesis Title Design of The Low Pass Type Wave Discrete Filters
Name Mr.Chawalit Benjangkprasert
Thesis Advisor Asst.Prof. Dr.Kanok Janchitrapongvej
Level of Study Master of Engineering in Electrical Engineering
Academic Year 1990

Abstract

The thesis imply, the realization of wave discrete filter circuit from a composite polynomial, Here in, the Butterworth-Ultraspherical low pass type was designed and it can be realized by mean of wave discrete filter. The amplitude characteristic of this filter is transitional between Butterworth and Ultraspherical function. This response possesses both Butterworth-like and Ultraspherical-like characteristics. With the passive structure, a class of wave discrete filter can be realized to posses the properties of the original LC passive filter. From the filter structure, it can be realized by means of series adaptor and parallel adaptor circuits using an adder and multiplier in conjunction with the sample-and-hold circuits as delay units.

The measured amplitude frequency characteristic of the filter can be operate with the input clock signal.

Based on the proposed circuit, the measured amplitude frequency characteristic of the filter is in good agreement with the theoretical characteristic.

สารบัญ

	หน้า
บทคัดย่อ	I
Abstract	II
บทที่ 1 บทนำ	1
1.1 วัตถุประสงค์ของวิทยานิพนธ์	1
1.2 เนื้อหาของวิทยานิพนธ์	1
1.3 ประโยชน์ที่ได้รับจากวิทยานิพนธ์	2
บทที่ 2 ทฤษฎีทั่วไป	3
2.1 เน็ตเวิร์กฟังก์ชัน	3
2.2 Z-Transform	6
2.3 วงจรแชนเนลและไฮลด์	12
บทที่ 3 เวกต์ริคัลฟิลเตอร์	16
3.1 ความไวของอุปกรณ์	16
3.2 คุณสมบัติของเวกต์ริคัล	18
3.3 การแปลงอิมพีแดนซ์ของวงจร	21
บทที่ 4 การออกแบบวงจรกรองความถี่ต่ำแบบคอมโพสิต โพลีโนเมียล	32
4.1 วงจรกรองความถี่ต่ำแบบบัตเตอร์เวิร์ท	32
4.2 วงจรกรองความถี่ต่ำแบบอูลตราสเฟียริคัล โพลีโนเมียล	35
4.3 การออกแบบวงจรกรองความถี่ต่ำแบบบัตเตอร์เวิร์ท-อูลตราสเฟียริคัล โพลีโนเมียล	40
4.4 คุณสมบัติของวงจรกรองความถี่ต่ำแบบบัตเตอร์เวิร์ท-อูลตราสเฟียริคัล โพลีโนเมียล	45
บทที่ 5 การออกแบบวงจรกรองความถี่ต่ำเวกต์ริคัลฟิลเตอร์	50
5.1 การหาค่าอิมพีแดนซ์ของวงจรจากทรานส์เฟอร์ฟังก์ชัน	50
5.2 การออกแบบวงจรกรองความถี่ต่ำเวกต์ริคัลฟิลเตอร์แบบบัตเตอร์เวิร์ทและแบบบัตเตอร์เวิร์ท-อูลตราสเฟียริคัล โพลีโนเมียล	55
5.3 ผลการทดสอบวงจร	69
บทที่ 6 สรุปผลการวิจัย	76
กิตติกรรมประกาศ	77
เอกสารอ้างอิง	78
ภาคผนวก รายละเอียดของโปรแกรม	80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1 บทนำ

ในการออกแบบวงจรของความถี่ต่ำ เพื่อให้ได้คุณลักษณะผลตอบสนองทางขนาดที่ต่ำที่สุด สามารถทำได้หลายวิธีทั้งแบบแอกทีฟและพาสซีฟ ซึ่งแต่ละวิธีมีข้อดีและข้อเสียแตกต่างกันไป เช่น การออกแบบวงจรพาสซีฟ [17] ที่ใช้อิเหน้ที่ R, L และ C ให้ค่าความไวแปรเปลี่ยนในตัวอิเหน้ที่ต่ำ แต่วงจรจะมีขนาดใหญ่ เพราะอินดักเตอร์ (L) ยังต้องทำเป็น Discrete Component ส่วนวงจรแอกทีฟ [14,17] นั้น สามารถใช้ออปแอมป์ร่วมกับอิเหน้ที่ R,C ทำให้วงจรจะมีขนาดเล็กและราคาถูก แต่ค่าความไวแปรเปลี่ยนในตัวอิเหน้ที่จะสูง ทำให้วงจรไม่เสถียรได้ จากโครงสร้างของวงจรพาสซีฟ LC ที่มีความไวต่ออิเหน้ที่ต่ำดังกล่าวแล้วนั้น สามารถแปลงวงจรพาสซีฟ LC ให้เป็นวงจรกรองความถี่แบบดิครีตที่เรียกว่า เวกดิครีตฟิลเตอร์ [4,5,6,7] ได้ โดยวิทยานิพนธ์นี้ได้เสนอการออกแบบวงจรกรองความถี่ต่ำด้วยวิธีใหม่ โดยใช้วงจรเวกดิครีตฟิลเตอร์ (Wave Discrete Filter Circuit)[1] โดยกำหนดทรานส์เฟอร์ฟังก์ชันของวงจรกรองความถี่ต่ำเป็นแบบบัตเตอร์ เวทและแบบบัตเตอร์ เวท-อูลตราสเฟียริคัล โพลีโนเมียล ซึ่งแบบบัตเตอร์ เวท-อูลตราสเฟียริคัล โพลีโนเมียลมีพารามิเตอร์สามารถกำหนดผลตอบสนองทางขนาดตามต้องการได้ วงจรดังกล่าวมีค่าความไว (Sensitivity) ของอิเหน้ที่ต่ำ และสามารถผลิตเป็นวงจร LSI โดยใช้เทคโนโลยีสมัยใหม่ที่มีขนาดเล็กได้

1.1 วัตถุประสงค์ของวิทยานิพนธ์

1. เพื่อศึกษาวิธีการทางคณิตศาสตร์ และนำมาประยุกต์ใช้งานทางวิศวกรรม
2. เพื่อศึกษาเทคนิคในการออกแบบ และผลตอบสนองของวงจรกรองความถี่ต่ำ
3. เพื่อศึกษาวิธีการออกแบบวงจรกรองความถี่ต่ำแบบคอม โพลิต โพลีโนเมียลฟังก์ชัน
4. เพื่อศึกษาแนวทางการออกแบบวงจรกรองความถี่ต่ำอย่างมีประสิทธิภาพ โดยใช้

วงจรเวกดิครีตฟิลเตอร์

1.2 เนื้อหาของวิทยานิพนธ์

สำหรับวิทยานิพนธ์นี้ จะกล่าวถึงวิธีการนำเอาทรานส์เฟอร์ฟังก์ชันที่ได้จากการออกแบบ นำมาสังเคราะห์เป็นวงจรกรองความถี่ต่ำแบบเวกดิครีตฟิลเตอร์ โดยอาศัยทฤษฎีการแปลงโครงสร้างจากวงจรพาสซีฟ LC ออกแบบเป็นวงจรอะนาล็อกอนุกรมและอะนาล็อกอนุกรม โดยใช้ออปแอมป์ร่วมกับความต้านทาน ต่อร่วมกับวงจรดีเลย์ยูนิต ซึ่งจะเป็นการหาผลตอบสนองทางขนาดเป็นหลัก

ในบทที่ 2 จะกล่าวถึงผลตอบสนองของเน็ตเวิร์กฟังก์ชัน Z-Transform แบบไพลีเนียร์ทรานส์ฟอร์ม และวงจรแรมเปลและโฮลด์

ในบทที่ 3 กล่าวถึงทฤษฎีทั่วไปของเวฟดิซครีทฟิลเตอร์ ซึ่งจะพิจารณาถึงความไว (Sensitivity) ของวงจร คุณลักษณะของเวฟเน็ตเวิร์ก และการแปลงอิมพีแดนซ์จากอิมพีแดนซ์ทางอนาล็อก ไปเป็นอิมพีแดนซ์ทางดิซครีทแบบเวฟเน็ตเวิร์ก

ในบทที่ 4 กล่าวถึงวงจรกรองความถี่ต่ำแบบบัตเตอร์เวท วงจรกรองความถี่ต่ำแบบอูลตราสเฟียวิคัล โพลีโนเมียล และการออกแบบวงจรกรองความถี่ต่ำแบบคอมโพสิตโพลีโนเมียลชนิดบัตเตอร์เวท-อูลตราสเฟียวิคัล โพลีโนเมียล ซึ่งสามารถกำหนดผลตอบสนองทางขนาดตามต้องการได้ พร้อมทั้งพิจารณาถึงผลตอบสนองทางเฟส, กรูฟตีเลเยอร์, อิมพัลส์ และ Unit step

ในบทที่ 5 เป็นการออกแบบวงจรกรองความถี่ต่ำ แบบบัตเตอร์เวทและแบบบัตเตอร์เวท-อูลตราสเฟียวิคัล โพลีโนเมียล โดยใช้วงจรเวฟดิซครีทฟิลเตอร์ (Wave Discrete Filter) และแสดงผลการทดสอบจากวงจรที่ออกแบบได้ ซึ่งจะพิจารณาถึงผลตอบสนองทางขนาดเท่านั้น

ในบทที่ 6 เป็นบทสรุปผลการวิจัย

ภาคผนวก เป็นรายละเอียดของโปรแกรม สำหรับทดสอบวงจรที่ออกแบบไว้ในบทที่ 4 ซึ่งผลลัพธ์ที่ได้จะแสดงเป็นกราฟ พร้อมทั้งรายละเอียดทั้งหมด โดยสามารถเลือกแสดงผลได้ 3 ทาง คือ 1. จอแสดงผล (Monitor) 2. เครื่องพิมพ์ (Printer) 3. เอกซ์-วายพล็อตเตอร์ (X-Y Plotter)

1.3 ประโยชน์ที่ได้รับจากวิทยานิพนธ์

1. สามารถนำคอมโพสิตโพลีโนเมียลไปใช้ในการออกแบบวงจรกรองความถี่ต่ำ
2. สามารถนำวงจรเวฟดิซครีทฟิลเตอร์ไปใช้ในการออกแบบวงจรกรองความถี่ต่ำ
3. สามารถนำไปใช้ในการประมาณหาเน็ตเวิร์กฟังก์ชันในขอบข่ายของความถี่ เพื่อให้ได้คุณลักษณะทางขนาดหรือกรูฟตีเลเยอร์ ใกล้เคียงกับคุณลักษณะทางขนาดหรือกรูฟตีเลเยอร์ตามที่ต้องการ
4. หลักการและการคำนวณหาค่าผลตอบสนองต่าง ๆ ในวิทยานิพนธ์นี้ บางส่วนได้เขียนเป็นโปรแกรมไว้เพื่อเป็นการจำลองผลตอบสนอง แล้วจึงทำการสร้างวงจรเพื่อใช้งานจริง ซึ่งสามารถนำไปเป็นแนวทางการพัฒนาหรือประยุกต์ใช้งานอย่างกว้างขวาง และมีประสิทธิภาพ

บทที่ 2 ทฤษฎีทั่วไป

2.1 เน็ตเวิร์กฟังก์ชัน

การออกแบบวงจรใด ๆ เพื่อให้ผลตอบสนองตามที่ต้องการ จะทำได้โดยหาเน็ตเวิร์กฟังก์ชันของวงจรเทียบเคียงกับฟังก์ชันที่ต้องการ ในอุดมคติ ซึ่งค่าที่ได้จากเน็ตเวิร์กฟังก์ชันจะต้องใกล้เคียงกับฟังก์ชัน ในอุดมคติมากที่สุด เช่น การออกแบบวงจรกรองความถี่จะต้องให้เน็ตเวิร์กฟังก์ชันที่ได้ยอมให้สัญญาณความถี่ในย่านผ่านสัญญาณ (Passband) ผ่านออกไปได้ทั้งหมดโดยไม่มีการสูญเสียหรือผิดเพี้ยน และสามารถกำจัดสัญญาณความถี่ในย่านหยุดสัญญาณ (Stopband) ได้ทั้งหมดโดยสิ้นเชิง แต่เนื่องจากวงจรใด ๆ สามารถแทนได้แต่เฉพาะฟังก์ชันที่อยู่ในรูปโพลีโนเมียลเศษส่วนเท่านั้น จึงจำเป็นต้องทำความเข้าใจเกี่ยวกับผลตอบสนองชนิดต่าง ๆ ของฟังก์ชันแบบนี้ ซึ่งผลตอบสนองของเน็ตเวิร์กฟังก์ชันต่อความถี่ [16, 17, 18] ที่สำคัญ ได้แก่

- ผลตอบสนองทางขนาด (Magnitude Response)
- ผลตอบสนองทางเฟส (Phase Response)

ผลตอบสนองทั้งสองชนิดนี้จะถูกพิจารณาเปรียบเทียบกับความถี่ของสัญญาณอินพุตที่มีการเปลี่ยนแปลง เน็ตเวิร์กโดยทั่วไปสามารถเขียนเป็นทราנסเฟอร์ฟังก์ชันที่อยู่ในรูปโพลีโนเมียลคือ

$$H(S) = \frac{N(S)}{D(S)} = \frac{a_m S^m + a_{m-1} S^{m-1} + \dots + a_1 S + a_0}{b_n S^n + b_{n-1} S^{n-1} + \dots + b_1 S + b_0} \quad \dots(2.1)$$

โดยที่ $N(S)$ และ $D(S)$ เป็นโพลีโนเมียลของตัวเศษและโพลีโนเมียลของตัวส่วนตามลำดับ m เป็นอันดับของโพลีโนเมียลเศษ และ n เป็นอันดับของโพลีโนเมียลส่วน โดยที่ $n \geq m$ a และ b เป็นสัมประสิทธิ์ที่เป็นตัวเลขจำนวนจริง S เป็นตัวแปรความถี่เชิงซ้อน (Complex-Frequency Variable) สมการ (2.1) ที่อยู่ในรูปโพลีโนเมียลสามารถแยกตัวประกอบซึ่งจะได้

$$H(S) = K \frac{(S-z_1)(S-z_2)\dots(S-z_m)}{(S-p_1)(S-p_2)\dots(S-p_n)} \quad \dots(2.2)$$

$$H(S) = K \frac{\prod_{i=1}^m (S-z_i)}{\prod_{j=1}^n (S-p_j)}$$

โดยที่ K เป็นอัตราขยายซึ่งมีค่าคงที่
 z_i เป็นตำแหน่งซีโรว์อันดับที่ i
 p_j เป็นตำแหน่งโพลอันดับที่ j

เมื่อพิจารณาผลตอบสนองทางความถี่ของเน็ตเวิร์ก โดยสมมติป้อนสัญญาณอินพุตเป็นรูปไซน์และเน็ตเวิร์กเป็นแบบลิเนียร์ ผลตอบสนองของวงจรในสถานะเสถียรของสัญญาณรูปไซน์ ซึ่ง S มีค่าเท่ากับ $j\omega$ ดังนั้นสมการ (2.2) จะได้

$$H(j\omega) = K \frac{(j\omega-z_1)(j\omega-z_2)\dots(j\omega-z_m)}{(j\omega-p_1)(j\omega-p_2)\dots(j\omega-p_n)} \dots(2.3)$$

เมื่อพิจารณาพหุคูณตัว ($j\omega-S_i$) โดยให้ S_i เป็นซีโรว์ คือ $S_i = z_i$ หรือเป็นโพล คือ $S_i = p_j$ ซึ่งค่า S_i จะเป็นตัวเลขเชิงซ้อนสมมติให้

$$S_i = \alpha_i + j\beta_i$$

ค่าพหุคูณตัว ($j\omega - S_i$) จะได้เป็น

$$-\alpha_i + j(\omega - \beta_i) = M_i e^{j\theta_i} \dots(2.4)$$

เมื่อ $M_i = \sqrt{\alpha_i^2 + (\omega - \beta_i)^2}$

$$\theta_i = \tan^{-1} \frac{\omega - \beta_i}{-\alpha_i}$$

ฉะนั้นสมการ (2.3) สามารถเขียนใหม่เป็น

$$\begin{aligned} H(j\omega) &= K \frac{M_{z1} e^{j\theta_{z1}} M_{z2} e^{j\theta_{z2}} \dots M_{zm} e^{j\theta_{zm}}}{M_{p1} e^{j\theta_{p1}} M_{pj} e^{j\theta_{pj}} \dots M_{pn} e^{j\theta_{pn}}} \\ &= K \frac{M_{z1} M_{z2} \dots M_{zm}}{M_{p1} M_{pj} \dots M_{pn}} e^{j(\theta_{z1} + \theta_{z2} + \dots + \theta_{zm} - \theta_{p1} - \theta_{pj} - \dots - \theta_{pn})} \dots(2.5) \end{aligned}$$

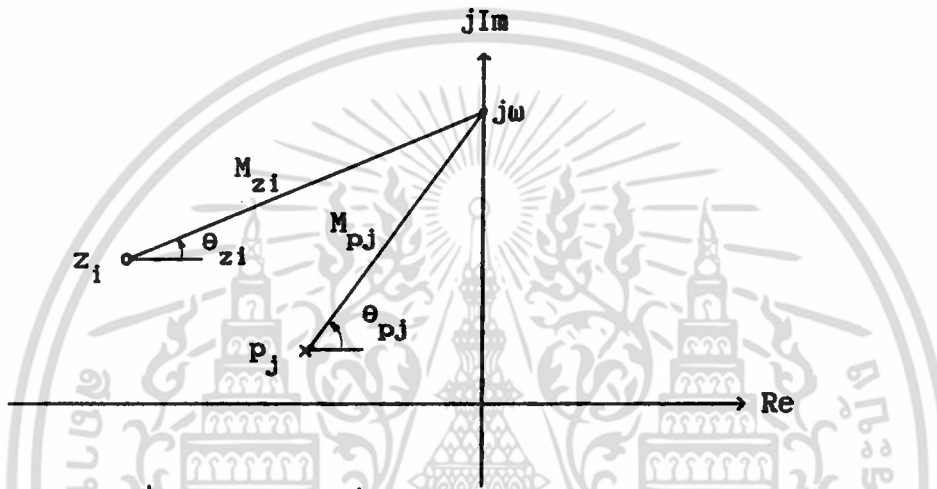
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$H(j\omega) = M(\omega) e^{j\theta(\omega)}$$

เมื่อ $M(\omega)$ เป็นขนาดของ $H(j\omega)$

$\theta(\omega)$ เป็นเฟสของ $H(j\omega)$

ใน S-Plane ค่าของแฟคเตอร์ $(j\omega - z_1)$ สามารถแสดงด้วยเวกเตอร์จาก z_1 ไปยัง $j\omega$ และแฟคเตอร์ $(j\omega - p_j)$ สามารถแสดงด้วยเวกเตอร์จาก p_j ไปยัง $j\omega$ ดังนั้นสามารถเขียนเวกเตอร์ซึ่งแสดงขนาด และเฟส ได้ดังในรูปที่ 2.1



รูปที่ 2.1 เวกเตอร์ของซีโรและโพลใน S-Plane

จากสมการ (2.5) และรูปที่ 2.1 จะเห็นได้ว่าผลตอบสนองทางขนาด $M(\omega)$ คือค่าของ K คูณกับผลคูณของขนาดของเวกเตอร์ที่ลากจากตำแหน่งซีโรไปยังความถี่ ω บนแกนจินตภาพ M_{z1} หารด้วยผลคูณของเวกเตอร์ที่ลากจากตำแหน่งโพลไปยังความถี่ ω บนแกนจินตภาพ M_{pj} สำหรับผลตอบสนองเฟส $\theta(\omega)$ คือผลรวมของมุมเวกเตอร์ของซีโร θ_{z1} ลบด้วยผลรวมของมุมเวกเตอร์ของโพล θ_{pj}

ผลตอบสนองของกรุปดีเลย์ (Group Delay) ได้จากการพิจารณาสัญญาณในโดเมนของเวลา โดยแสดงให้อยู่ในรูปของโดเมนของความถี่ เช่นเดียวกับกับผลตอบสนองทางขนาดและเฟส โดยสมมติสัญญาณอินพุต V_1 บ้อนไปยังเน็ตเวิร์กที่มีค่ากรุปดีเลย์เท่ากับ D วินาที ดังนั้นสัญญาณทางเอาท์พุท V_2 จะได้

$$V_2(t) = V_1(t - D) \quad \dots(2.6)$$

เนื่องจากสัญญาณใด ๆ เกิดจากองค์ประกอบของสัญญาณรูปไซน์เสมอ ฉะนั้นสัญญาณอินพุทจะได้

$$V_1 = A \sin(\omega t + \phi) \quad \dots(2.7)$$

จากสมการ (2.6) จะได้สัญญาณเอาท์พุทคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$V_2 = A \sin [\omega(t - D) + \phi]$$

หรือ $V_2 = A \sin [\omega t - \omega D + \phi] \dots(2.8)$

จากสมการ (2.7) และ (2.8) จะเห็นได้ว่าสัญญาณอินพุต และเอาต์พุตมีเฟสต่างกันคือ

$$\theta = -\omega D \dots(2.9)$$

ฟังก์ชันกรุปดีเลย์ (Group Delay) จะได้จากการหาอนุพันธ์ของสมการ (2.9) เทียบกับ ω ซึ่งจะได้

$$D = -\frac{d}{d\omega} \theta \dots(2.10)$$

ดังนั้นผลตอบสนองของกรุปดีเลย์ก็คือค่าลบการหาอนุพันธ์ผลตอบสนองทางเฟสนั่นเอง

2.2 Z-Transform

เทคนิคที่ใช้ในการออกแบบวงจรกรองความถี่ดิจิทัลมีหลายวิธี แต่ที่นิยมใช้กันมาก คือ การทราנסฟอร์ม หรือการแมปปิง (Mapping) จากวงจรถองความถี่อนาล็อกไปเป็นวงจรถองความถี่ดิจิทัล หรือกล่าวอีกอย่างหนึ่งได้ว่า ชั้นแรกเราหาทราנסเฟอร์ฟังก์ชัน $H(s)$ ของวงจรถองความถี่อนาล็อกตามที่ต้องการออกแบบ หลังจากนั้นก็ใช้การทราנסฟอร์มที่เหมาะสมจาก S-Plane ไปเป็น Z-Plane ก็จะได้รับทราנסเฟอร์ฟังก์ชัน $H(z)$ ของวงจรถองความถี่ดิจิทัล จากทฤษฎีของ Z-Transform มีหลายวิธี ตัวอย่าง เช่น วิธี Impulse-Invariant Transformation วิธี Matched Z-Transform แต่ในวิทยานิพนธ์นี้จะใช้วิธีของไบลิเนียร์ทราנסฟอร์มเมชัน

วิธีไบลิเนียร์ทราנסฟอร์มเมชัน (Bilinear-Transformation Method)

การแปลงด้วยวิธีของ Impulse-Invariant และ Match Z-Transform [7] นั้นยังมีข้อผิดพลาดเกิดขึ้น สาเหตุก็เนื่องมาจากขั้นตอนการ Map จาก S-Plane ไปเป็น Z-Plane เป็นการ Map จากจุดหนึ่งใน S-Plane ไม่ได้หลาย ๆ จุดใน Z-Plane แต่วิธีไบลิเนียร์ทราנסฟอร์ม จะเป็นการ Map จากจุดหนึ่งไปได้เพียงจุดเดียว ซึ่งจะทำให้ผลที่ได้มีคุณลักษณะดีกว่าสองแบบแรก

Derivation

พิจารณาคุณสมบัติของอนาล็อกอินทีเกรเตอร์ โดยกำหนดทราנסเฟอร์ฟังก์ชันเป็น

$$H_I(s) = \frac{1}{s} \dots(2.11)$$

จะได้ผลตอบสนองของอินทีเกรเตอร์ต่ออิมพัลส์

$$\mathcal{L}^{-1}H_I(s) = h_I(t) = \begin{cases} 1 & \text{สำหรับ } t \geq 0^+ \\ 0 & \text{สำหรับ } t \leq 0_- \end{cases} \dots(2.12)$$

และถ้ากำหนดอินพุตเป็น $x(t)$ เอาท์พุทเป็น $y(t)$ และ $h(t)$ เป็นผลตอบสนองอิมพัลส์ของวงจร โดยใช้ทฤษฎีของ Convolution Integral จะได้

$$y(t) = \int_0^t x(\tau)h_I(t-\tau)d\tau \quad \dots(2.13)$$

ถ้า $0^+ < t_1 < t_2$, เราสามารถที่จะเขียนสมการ (2.13) ใหม่ได้ว่า

$$y(t_2) - y(t_1) = \int_0^{t_2} x(\tau)h_I(t_2-\tau)d\tau - \int_0^{t_1} x(\tau)h_I(t_1-\tau)d\tau \quad \dots(2.14)$$

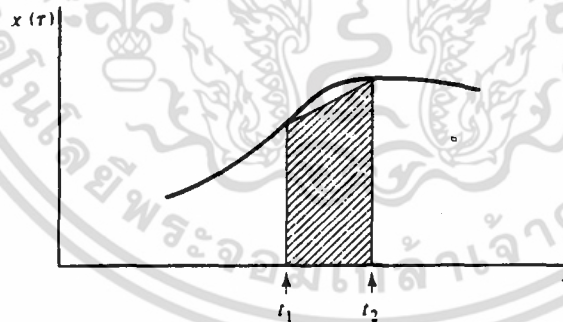
สำหรับ $0^+ < \tau \leq t_1, t_2$

$$h_I(t_2-\tau) = h_I(t_1-\tau) = 1 \quad \dots(2.15)$$

และจากสมการ (2.14) สามารถเปลี่ยนใหม่ได้เป็น

$$y(t_2) - y(t_1) = \int_{t_1}^{t_2} x(\tau)d\tau \quad \dots(2.16)$$

ที่ $t_1 \rightarrow t_2$ แสดงดังรูปที่ 2.2



รูปที่ 2.2 ผลตอบสนองของอนาล็อกอินทิเกรเตอร์

จากสมการ (2.16) จะได้

$$y(t_2) - y(t_1) \cong \frac{t_2-t_1}{2} [x(t_1) + x(t_2)] \quad \dots(2.17)$$

และถ้าแทนค่า $t_1 = nT-T$ และ $t_2 = nT$ ลงในสมการ (2.17) จะได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$y(nT) - y(nT-T) = \frac{T}{2} [x(nT-T)+x(nT)] \quad \dots(2.18)$$

จากสมการ (2.18) ทำการแปลงให้อยู่ในรูปดีจิตอลด้วย Z-Transform จะได้

$$Y(Z) - Z^{-1} Y(Z) = \frac{T}{2} [Z^{-1} X(Z) + X(Z)] \quad \dots(2.19)$$

และสามารถเขียนเป็นทราנסเฟอร์ฟังก์ชัน ของอินทีเกรเตอร์ทางดีจิตอลได้

$$H_I(Z) = \frac{Y(Z)}{X(Z)} = \frac{T}{2} \frac{Z+1}{Z-1} \quad \dots(2.20)$$

จากคุณลักษณะของวงจรวงความถี่อนาล็อก

$$H_A(S) = \frac{\sum_{i=0}^N a_i S^{N-i}}{S^N + \sum_{i=1}^N b_i S^{N-i}} \quad \dots(2.21)$$

จากทราנסเฟอร์ฟังก์ชันในสมการ (2.21) สามารถเขียนเป็นวงจรถัดแสดงในรูปที่ 2.3 ซึ่งมีส่วนประกอบที่สำคัญ คือ วงจรวก วงจรคูณ และวงจรมอดูเลเตอร์ ถ้าแต่ละอิลิเมนต์ทางอนาล็อกในรูปที่ 2.3 แทนด้วย อิลิเมนต์ทางดีจิตอลตามลำดับแล้ว ผลลัพธ์ที่ได้ก็จะกลายเป็นวงจรวงความถี่ดีจิตอลถ้าอินทีเกรเตอร์ทางดีจิตอลหาตามตัวอย่างข้างต้น ผลของทราנסเฟอร์ฟังก์ชันแบบวงจรวงความถี่ดีจิตอลที่ใช้การแปลงแบบ ไบเลเนียน์ทราנסเฟอร์ม คือ

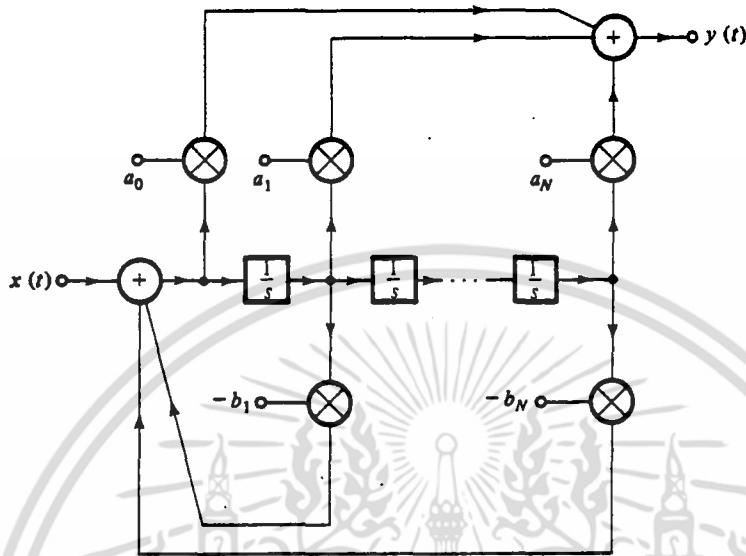
$$S = \frac{2}{T} \frac{Z - 1}{Z + 1} \quad \dots(2.22)$$

หรือ

$$Z = \frac{2/T + S}{2/T - S} \quad \dots(2.23)$$

จะได้

$$H_D(Z) = H_A(S) \Big|_S = \frac{2(Z - 1)}{T(Z + 1)} \quad \dots(2.24)$$



รูปที่ 2.3 วงจรที่ได้จากทราสเฟอร์ฟังก์ชัน $H_A(S)$

ขั้นตอนการแปลงจากวงจรของความถี่อนาล็อก ด้วยวิธีไบลิเนียร์ทรานส์ฟอร์มเพื่อให้ได้ผลลัพธ์ อยู่ในรูปของวงจรของความถี่ดิจิทัล สามารถแปลงได้ดังที่กล่าวมาข้างต้น

วิธี ไบลิเนียร์ทรานส์ฟอร์มนี้นิยมใช้กันมากในการทรานส์ฟอร์มจากวงจรของความถี่อนาล็อกไปเป็นวงจรของความถี่ดิจิทัล ทรานส์เฟอร์ฟังก์ชันของวงจรของความถี่ดิจิทัล เมื่อนำมาประมาณ Magnitude จะให้คุณลักษณะใกล้เคียงกันกับ Magnitude ที่ได้จากวงจรของความถี่อนาล็อก

จากสมการ (2.22) และ (2.23) เรามาศึกษาคุณสมบัติของไบลิเนียร์ทรานส์ฟอร์ม โดยพิจารณาการ Map จากจุด $Z = re^{j\theta}$ ใน Z-Plane ไปยังจุด $S = \sigma + j\omega$ ใน S-Plane โดยใช้สมการ (2.23) เราสามารถเขียนใหม่ได้ว่า

$$re^{j\theta} = \frac{2/T + \sigma + j\omega}{2/T - \sigma - j\omega} \quad \dots(2.25)$$

โดยที่
$$r = \left[\frac{(2/T + \sigma)^2 + \omega^2}{(2/T - \sigma)^2 + \omega^2} \right]^{1/2} \quad \dots(2.26)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ
$$\theta = \tan^{-1}\left(\frac{\omega}{2/T+\sigma}\right) + \tan^{-1}\left(\frac{\omega}{2/T-\sigma}\right) \dots(2.27)$$

จากสมการ (2.26) ถ้าให้

$$\sigma > 0 \text{ จะได้ } r > 1$$

$$\sigma = 0 \text{ จะได้ } r = 1$$

$$\sigma < 0 \text{ จะได้ } r < 1$$

เมื่อ $\sigma > 0$ อยู่ทางด้านครึ่งขวาของ S-Plane เมื่อแปลงไปอยู่ใน Z-Plane จะอยู่ภายนอกของวงกลม Unit circle $|Z| = 1$

เมื่อ $\sigma < 0$ อยู่ทางด้านครึ่งซ้ายของ S-Plane เมื่อแปลงไปอยู่ใน Z-plane จะอยู่ภายในของวงกลม Unit circle $|Z| = 1$

และเมื่อ $\sigma = 0$ บนแกน $j\omega$ -axis ของ S-Plane เมื่อทำการแปลงแล้วจะอยู่บน Unit circle ใน Z-Plane $|z| = 1$

สำหรับ $\sigma = 0, r = 1$ และจากสมการ (2.27) จะได้

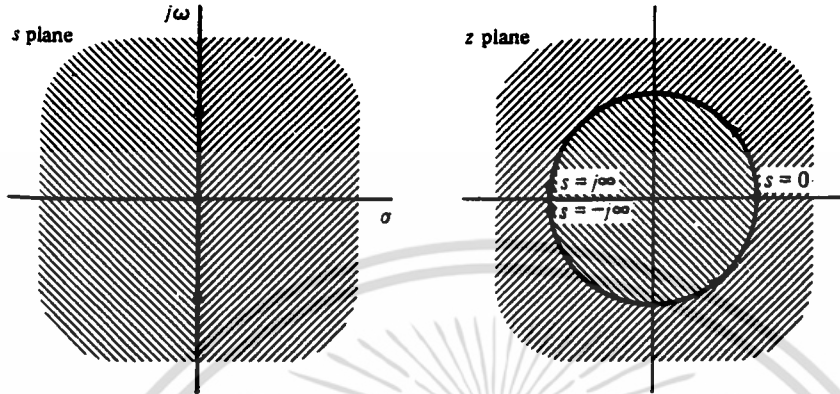
$$\theta = 2 \tan^{-1} \frac{(\omega T)}{2}$$

จะได้

$$\begin{aligned} \theta &= 0 && \text{ที่ } \omega = 0 \\ \theta &\rightarrow \pi && \text{ที่ } \omega \rightarrow +\infty \\ \theta &\rightarrow -\pi && \text{ที่ } \omega \rightarrow -\infty \end{aligned}$$

จุดศูนย์กลางของ S-Plane จะถูก Map ไปอยู่ที่จุด (1,0) ของ Z-Plane และค่าบวกกับค่าลบบนแกน $j\omega$ -axis ของ S-Plane จะถูก Map ไปอยู่ส่วนบนและส่วนล่างของวงกลมรัศมีเท่ากับหนึ่งหน่วย $|Z|=1$ ตามลำดับ ผลที่ได้จากการทรานส์ฟอร์ม แสดงไว้ในรูปที่ 2.4

ไบเลนิเยร์ทรานส์ฟอร์มนั้น เป็นการแปลงจาก $j\omega$ -axis ของ S-Plane ไปเป็น Unit circle ของ Z-Plane ซึ่งจะทำให้เกิดความแตกต่างกันทางสเกลความถี่ระหว่างวงจรรองความถี่อนาล็อกกับวงจรรองความถี่ดิจิทัล เนื่องมาจากสาเหตุของปรากฏการณ์ที่เบี่ยงเบนออกจากลิเนียร์ เพื่อให้เข้าใจง่ายขึ้น โดยให้พิจารณาความสัมพันธ์ระหว่างความถี่ของวงจรรองความถี่อนาล็อกและของวงจรรองความถี่ดิจิทัล ด้วยการกำหนดความถี่ทางวงจรรองความถี่อนาล็อกแปรค่าตาม Ω และความถี่ทางดิจิทัลแปรค่าตาม ω แล้วมาศึกษาผลของการเบี่ยงเบนจากลิเนียร์ โดยการแทน $S=j\Omega$ และ $Z = e^{j\omega T}$ ในสมการ (2.22) จะได้



รูปที่ 2.4 โพลีเนียร์ทรานส์ฟอร์มจาก S-Plane ไปเป็น Z-Plane

$$j\Omega = \frac{2}{T} \left(\frac{e^{j\omega T} - 1}{e^{j\omega T} + 1} \right) = \frac{2}{T} \left(\frac{(e^{j\omega T/2} - e^{-j\omega T/2})/j2}{(e^{j\omega T/2} + e^{-j\omega T/2})/j2} \right)$$

$$j\Omega = j \frac{2}{T} \tan \frac{\omega T}{2} \dots (2.28)$$

จากสมการ (2.28) จะได้ความสัมพันธ์ทางความถี่ระหว่างวงจรวงความถี่นาล็อกกับวงจรวงความถี่ดิจิทัล ดังนี้

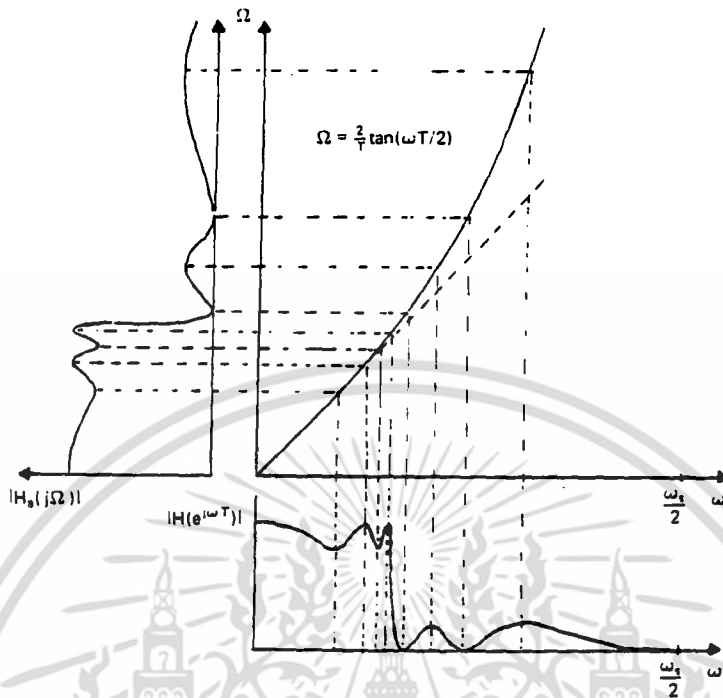
$$\Omega = \frac{2}{T} \tan \frac{\omega T}{2} \dots (2.29)$$

จากสมการ(2.29) เมื่อนำมาพล็อตจะได้ดังรูปที่ 2.5 จากรูปจะเห็นได้ว่าเป็นกราฟนอนลิเนียร์ (Nonlinear) ก็เนื่องมาจากผลของการเบี่ยงเบนทางความถี่ ซึ่งคล้ายกับการเบี่ยงเบนที่เกิดจากการแปลงผลตอบสนองทางขนาดจากวงจรวงความถี่นาล็อกไปเป็นวงจรวงความถี่ดิจิทัล เมื่อ $\omega T/2 \ll 1$ ความสัมพันธ์จะประมาณเป็นลิเนียร์ไปเพราะว่า

$$\Omega \cong \frac{2}{T} \omega \frac{T}{2} = \omega \dots (2.30)$$

ถ้ากำหนดความถี่ช่วงขอบของ Passband เป็น ω_p จากสมการ (2.29) ความถี่ช่วงขอบของ Passband ของวงจรวงความถี่นาล็อกจะเป็น

$$\Omega_p = \frac{2}{T} \tan \frac{\omega_p T}{2} \dots (2.31)$$



รูปที่ 2.5 กราฟแอนลิเนียร์ในการแปลงจากอนาล็อก $|H_a(j\Omega)|$ ไปเป็นดิจิทัล $|H_d(e^{j\omega T})|$ โดยใช้ไบลิเนียร์ทรานส์ฟอร์ม ในทำนองเดียวกันจะให้ความถี่ขอบของ Stopband ดังนี้

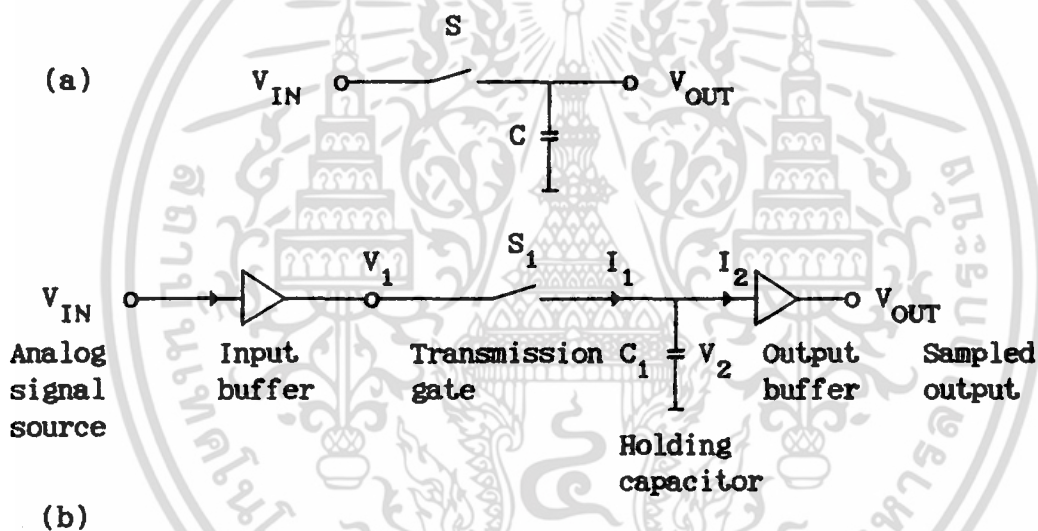
$$\Omega_r = \frac{2}{T} \tan \frac{\omega_r T}{2} \dots(2.32)$$

โดยที่ ω_r เป็นความถี่ช่วงขอบของ Stopband

2.3 วงจรแซมเปิลและโฮลด์ (Sample-and-Hold Circuit)

ข้อมูลที่เราได้รับมานั้น จะมาจากแหล่งกำเนิดสัญญาณหลาย ๆ ที่ต่างชนิดกัน ในการที่จะส่งหรือถ่ายโอนสัญญาณ จะต้องเปลี่ยนแปลงสัญญาณเหล่านี้ให้อยู่ในรูปแบบที่เหมาะสมเพื่อนำส่งสัญญาณไปยัง ช่องติดต่อสื่อสารข้อมูล หรือคอมพิวเตอร์ จะต้องใช้มัลติเพล็กซ์เซอร์ (Multiplexer) เป็นตัวเลือกเพื่อลำดับสัญญาณ ให้เรียงตามลำดับ ถ้าสัญญาณที่อยู่ในรูปของอนาล็อกถูกเปลี่ยนไปเป็นค่าแรงดันคงที่ ภายในช่วงเวลาหนึ่งเราจะเรียกการทำงานของขั้นตอนนี้ว่าวงจรแซมเปิลและโฮลด์ (Sample-and-Hold Circuit) แล้วถ้าหากเปลี่ยนค่าแรงดันคงที่ที่ทางเอาต์พุตของวงจรแซมเปิลและโฮลด์ ไปเป็นสัญญาณดิจิทัลจะเรียกวิธีการนี้ว่าการเปลี่ยนข้อมูลจากสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล (Analog to Digital Converter) สำหรับกรณีที่ต้องการติดต่อสื่อสารข้อมูลแบบดิจิทัล

สัญลักษณ์ของวงจรแซมเปิลและโฮลด์ แสดงไว้ในรูปที่ 2.6(a) ส่วนการทำงานของวงจรแซมเปิลและโฮลด์ สามารถอธิบายได้ดังแสดงในรูปที่ 2.6(b) ซึ่งมีส่วนประกอบที่สำคัญคือ อินพุตบัฟเฟอร์ (Input Buffer) ทรานส์มิสชันเกท (Transmission Gate) โฮลด์คิงคาปาซิเตอร์ (Holding Capacitor) และเอาต์พุตบัฟเฟอร์ (Output Buffer) ในแต่ละส่วนจะมีหน้าที่แตกต่างกัน อินพุตบัฟเฟอร์ทำหน้าที่รักษาแรงดัน V_{IN} ให้ปรากฏที่ V_1 และป้องกันไม่ให้เกิดการดึงกระแสจากแหล่งกำเนิดของสัญญาณอนาล็อกมากเกินไป และยังเป็นตัวจ่ายกระแสเพื่อไปประจุให้แก่ Holding Capacitor C_1 ตามความต้องการด้วย ส่วนเอาต์พุตบัฟเฟอร์จะเป็นตัวรักษาแรงดันของ Holding Capacitor V_2 ให้ไม่ปรากฏที่ V_{OUT} ในช่วงเวลาระหว่างที่สวิตช์ S_1 เปิดออก (OFF) และยังป้องกันไม่ให้ Holding Capacitor คายประจุออก หรือคายให้น้อยที่สุด



รูปที่ 2.6 (a) สัญลักษณ์ (b) วงจรแซมเปิลและโฮลด์

สถานะของสวิตช์ S_1 จะถูกควบคุมด้วยสัญญาณดิจิทัล เมื่อสวิตช์ S_1 ปิด (ON) ตัวเก็บประจุจะเริ่มประจุ (Charged) จนกระทั่งมีแรงดันเท่ากับแรงดันอนาล็อก V_1 ซึ่งมีค่าเท่ากับ V_{IN} เราเรียกการทำงานของขั้นตอนนี้ว่า โหมดแซมเปิล (Sample Mode) เมื่อสวิตช์ S_1 เปิดออก (OFF) ตัวเก็บประจุ C_1 จะต้องรักษาแรงดัน V_2 ให้มีค่าเท่ากับ V_1 เรียกการทำงานในหน้าที่นี้ว่า โหมดโฮลด์ (Hold Mode) ถ้าหากเอาต์พุตบัฟเฟอร์ดึงกระแสจาก Holding Capacitor น้อย ตัวเก็บประจุก็จะสามารถรักษาระดับแรงดัน V_2 ให้คงอยู่ได้เป็นเวลานาน ซึ่งในทางอุดมคติวงจรแซมเปิลและโฮลด์ กระแสรั่วไหล I_1 และ I_2 ต้องเท่ากับศูนย์ และแรงดัน V_2 ต้องคงอยู่ตลอดไป [20, 21]

ในโหมดโฮลด์ แรงดันของ Holding Capacitor จะขึ้นอยู่กับกระแสรั่วไหลผ่าน สวิตช์ S_1 (I_1) กับกระแสที่ใช้ไบอัส (Bias) ให้แก่ เอาก์พุทบัฟเฟอร์ (I_2) มีหน่วยเป็น Volts/Sec และสามารถเขียนเป็นสมการได้ดังนี้

$$\frac{\Delta V}{\Delta t} = \frac{I}{C_1} = \frac{I_1 + I_2}{C_1} \text{ V/Sec} \quad \dots(2.33)$$

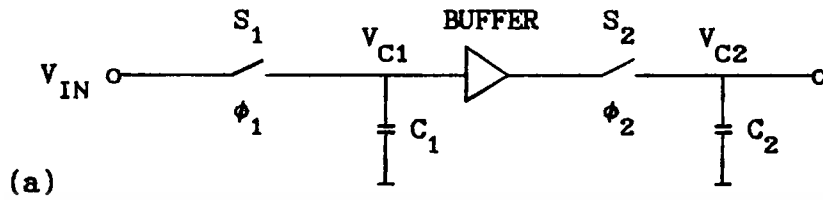
จากสมการ (2.33) ต้องกำหนดให้ตัวเก็บประจุ C_1 เป็นชนิดที่มีกระแสรั่วไหลต่ำมาก หรือมีค่าเท่ากับศูนย์

ถ้าหากต้องการใช้วงจรแซมเปิลและโฮลด์ ที่ให้ผลตอบสนองต่อการเปลี่ยนแปลงของ สัญญาณอินพุตที่ดี ในโหมดแซมเปิลความเร็วในการประจุพลังงานที่ Holding Capacitor จะ ถูกจำกัดโดยความต้านทานบนตัวสวิตช์ S_1 (R_S ON) และพลังงานที่สะสมไว้ในตัวเก็บประจุ C_1 จะลดลง 3 dB จากแรงดันทางอินพุทบัฟเฟอร์ซึ่งสามารถหาได้จาก

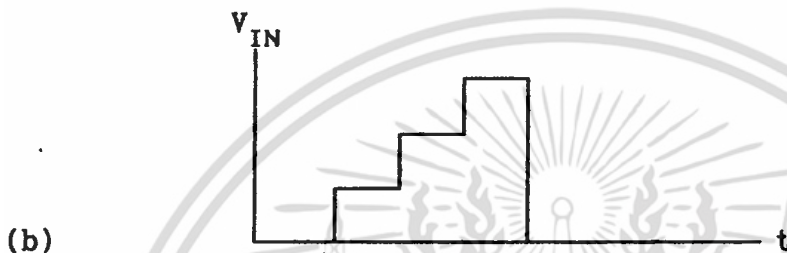
$$f_{3dB} = \frac{1}{2\pi R_S C_1} \quad \dots(2.34)$$

จากสาเหตุนี้ สำหรับวงจรที่ต้องการความเที่ยงตรงมาก ๆ ความถี่ใช้งานทางอินพุทจะต้องมีค่าต่ำกว่าความถี่หาได้จาก สมการ (2.34) ในทางปฏิบัติจากสมการ (2.33) และสมการ (2.34) ถ้าหากใช้ฮอปแอมป์ที่ต้องการกระแสไบอัสต่ำ ๆ ความต้านทานของสวิตช์ (ON) ต่ำ และตัวเก็บประจุมีกระแสรั่วไหลต่ำด้วย ก็จะได้วงจรแซมเปิลและโฮลด์ที่ให้ผลตอบสนองเร็ว และเที่ยงตรงตามต้องการ

วงจรแซมเปิลและโฮลด์ ยังสามารถทำหน้าที่เป็นวงจรดีเลย์ยูนิต (Delay Unit) ได้ พิจารณาจากรูปที่ 2.7 ซึ่งเป็นวงจรแซมเปิลและโฮลด์ ต่อ Cascade กัน 2 Stage โดยใช้ ฮอปแอมป์ต่อกันกลางระหว่าง Stage เพื่อเป็นบัฟเฟอร์ การทำงานของวงจรเพื่อทำหน้าที่เป็นดีเลย์ยูนิต อธิบายได้ดังนี้ จากวงจรต้องใช้สัญญาณนาฬิกา 2 ชุด คือ ϕ_1 และ ϕ_2 เป็นสัญญาณ Sampling เมื่อป้อนสัญญาณนาฬิกา ϕ_1 ให้แก่วงจร สวิตช์ S_1 ปิด (ON) ตัวเก็บประจุ(C_1) จะ เริ่มเก็บประจุจนกระทั่งมีแรงดันเท่ากับ V_{IN} แล้วเมื่อสวิตช์(S_1)เปิดออก (OFF) ที่เวลา t_1 แรงดันที่ตกคร่อมตัวเก็บประจุ (C_1) จะยังไม่เปลี่ยนแปลง และมีค่าเท่ากับค่าแรงดันอินพุทที่เวลา $t=t_1$ ช่วงเวลาที่ดีเลย์ ไปนั้นจะขึ้นอยู่กับสัญญาณนาฬิกาที่ป้อนให้แก่สวิตช์ S และจำนวนของตัวเก็บ ประจุที่ใช้ในวงจรที่คาสเคดกัน เหมือนกับลักษณะของสัญญาณแรงดัน ที่แสดงในรูป 2.7 จากรูปจะ เห็นได้ว่าที่จุด V_{C1} สัญญาณแรงดันจะดีเลย์ไปช่วงเวลาหนึ่ง ส่วนที่จุด V_{C2} สัญญาณแรงดันจะดีเลย์ไปเพิ่มอีกช่วงเวลาหนึ่ง แล้วแต่จุดประสงค์ของผู้นำไปใช้งานว่าต้องการดีเลย์มากน้อยเพียง ใด จากรูปสังเกตเห็นได้ว่า เราสามารถใช้วงจรแซมเปิลและโฮลด์ทำหน้าที่เป็นวงจรดีเลย์ยูนิตได้



(a)



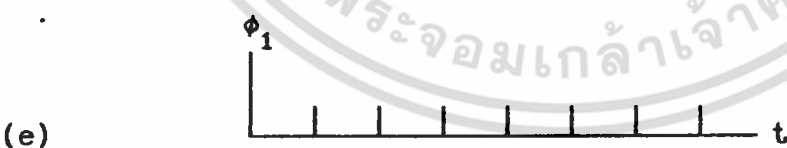
(b)



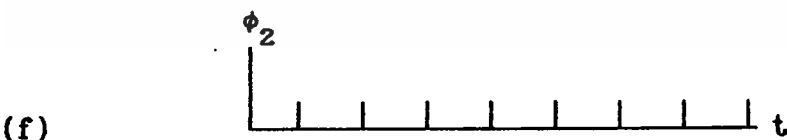
(c)



(d)



(e)



(f)

รูปที่ 2.7 วงจรตัวเลข 2 Stage และสัญญาณรูปคลื่นที่จุดต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 เวทิจิตตอลฟิลเตอร์

การออกแบบวงจรของความถี่แบบจิตตอล โดยทั่วไป ถ้าหากใช้โครงสร้างของวงจรที่มีความไวแปรเปลี่ยนในตัวอิลิเมนต์ต่ำ (Low Sensitivity Structures) สัมประสิทธิ์ในวงจรจิตตอลนั้น ก็จะมีผลต่อการเปลี่ยนแปลงของผลตอบสนองน้อย จากคุณสมบัตินี้ ในการสังเคราะห์วงจรที่มีโครงสร้างความไวแปรเปลี่ยนในตัวอิลิเมนต์ต่ำ Fettweis [8] เป็นผู้ริเริ่มขึ้น และได้ปรับปรุงพัฒนาโดย Sedlmeyer กับ Fettweis [9] ในบทนี้จึงได้เสนอโครงสร้างที่ใช้วงจรถาสาซีฟ LC เป็นวงจรต้นแบบ นำมาออกแบบเป็นวงจรจิตตอล วิธีการออกแบบนั้นทำได้โดยการแทนอิลิเมนต์ก่อนนำล๊อคด้วยอิลิเมนต์จิตตอลที่เหมาะสม วงจรของความถี่ก่อนนำล๊อคพาสซีฟ LC ก็จะถูกเปลี่ยนไปเป็นวงจรของความถี่จิตตอล ซึ่งการสังเคราะห์นั้นจะใช้คุณลักษณะของเวทิจิตตอลเป็นหลักพื้นฐาน จากหลักการดังกล่าว เราจะเรียกวงจรของความถี่นี้ว่า เวทิจิตตอลฟิลเตอร์ [4,5,6,7] วงจรเวทิจิตตอลฟิลเตอร์นี้ จะให้ค่าความไวแปรเปลี่ยนในตัวอิลิเมนต์ต่ำ เพราะโครงสร้างของวงจรถาสาซีฟ LC ให้ค่าความไวแปรเปลี่ยนในตัวอิลิเมนต์ต่ำ

3.1 ความไวของอุปกรณ์ (Sensitivity)

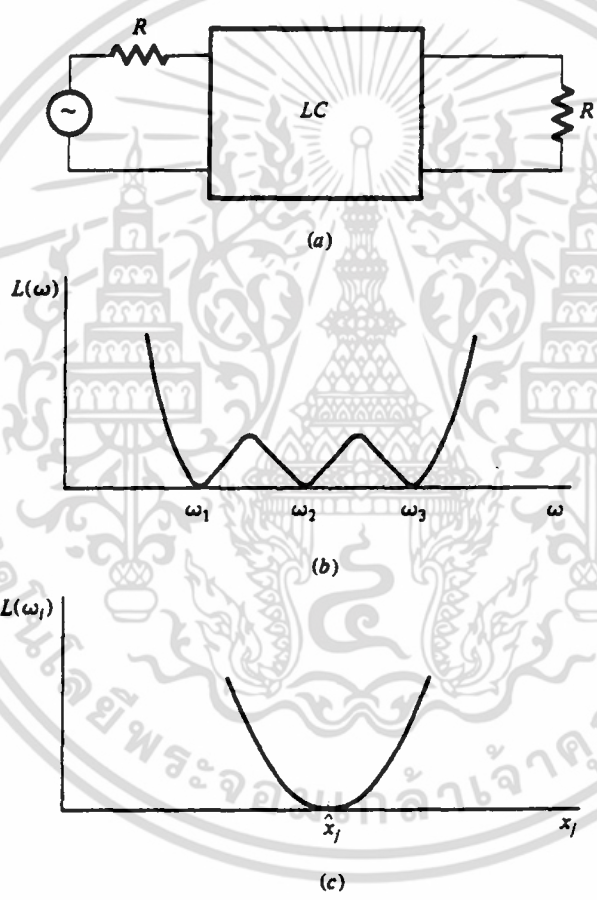
การออกแบบวงจรใด ๆ จะต้องพิจารณาถึงค่าอิลิเมนต์แต่ละตัว เพราะค่าอิลิเมนต์ที่ใช้ในงานจริง ๆ แล้วจะไม่มีค่าตรงตามอุดมคติ ค่าผิดพลาดที่เกิดขึ้นได้จากผลรวมของความคลาดเคลื่อนของตัวอุปกรณ์ที่เกิดจากการรวมวิธีในการผลิตเอง และคุณลักษณะทางโครงสร้างของอิลิเมนต์ที่เกิดการเปลี่ยนแปลงไปตามสภาพแวดล้อมที่ใช้งานอยู่ เช่น อุณหภูมิของห้อง , ความชื้น , ความเข้มของแสงสว่าง และ อายุการใช้งานของอุปกรณ์ เหตุนี้เองจึงจำเป็นต้องศึกษาผลกระทบของอิลิเมนต์ต่าง ๆ ภายในวงจรที่อาจเปลี่ยนแปลงไม่ตรงตามที่คำนวณ

เช่นซิวิตีตีเป็นค่าของการเปลี่ยนแปลงที่เกิดขึ้นที่เอาท์พุทของวงจร ซึ่งเป็นผลมาจากการแปรผันของพารามิเตอร์ต่าง ๆ ในวงจร เช่น การเปลี่ยนแปลงของค่าตัวต้านทาน (Resistor), ค่าตัวเหนี่ยวนำ (Inductor), ค่าตัวเก็บประจุ (Capacitor), ค่าอัตราขยายของทรานซิสเตอร์ และค่าอัตราขยายของออปแอมป์ (Operation Amplifier) ซึ่งแน่นอนว่าวงจรที่มีค่าอัตราการเปลี่ยนแปลงของเช่นซิวิตีตีมาก จะมีการแปรผันของพารามิเตอร์ภายในวงจรมีค่ามาก ซึ่งจะทำให้ผลตอบสนองทางด้านเอาท์พุทเกิดความผิดเพี้ยนได้ง่าย ฉะนั้นโครงสร้างของวงจรที่ดีจึงควรมีค่าเช่นซิวิตีตีของอิลิเมนต์แต่ละตัวต่ำ จะทำให้ง่ายต่อการนำไปใช้งานและยังไม่ต้องเลือกใช้ค่าของอุปกรณ์ที่มีความเที่ยงตรงสูงมากเกินไปจนความจำเป็น ซึ่งจะช่วยให้ประหยัดค่าใช้จ่าย

พิจารณาความไวของวงจรความถี่แบบ LC ดังแสดงในรูปที่ 3.1(a) สามารถกำหนดคุณลักษณะของการสูญเสีย ได้ดังนี้

$$L(\omega) = 10 \log \frac{P_m(\omega)}{P(\omega)}$$

- โดยที่
- $L(\omega)$ เป็นกำลังสูญเสีย
 - $P(\omega)$ เป็นกำลังทางเอาต์พุตที่แท้จริง
 - $P_m(\omega)$ เป็นกำลังทางเอาต์พุตสูงสุด ภายใต้สภาพที่ไม่มี การสูญเสีย



รูปที่ 3.1 (a) วงจรกรองความถี่พาสซีฟ LC (b) Loss Characteristic (c) $L(\omega_1)$ เทียบกับ X_j

ในเมื่อวงจรเน็ตเวิร์ก 2 พอร์ท แบบ LC นั้น เป็นเน็ตเวิร์กแบบ Passive ที่มีการสูญเสียน้อย จะทำให้ได้กำลังทางเอาต์พุตที่แท้จริงมีค่าน้อยกว่ากำลังทางเอาต์พุตสูงสุดที่อยู่ภายใต้สภาพที่ไม่มี การสูญเสีย หรือกำลังเอาต์พุตที่แท้จริงมีค่าเท่ากับกำลังเอาต์พุตสูงสุดในกรณีวงจรเน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เวกเตอร์แบบ LC ไม่มีการสูญเสีย และเขียนได้ว่า

$$P(\omega) \leq P_m(\omega)$$

$$\text{และ } L(\omega) \geq 0$$

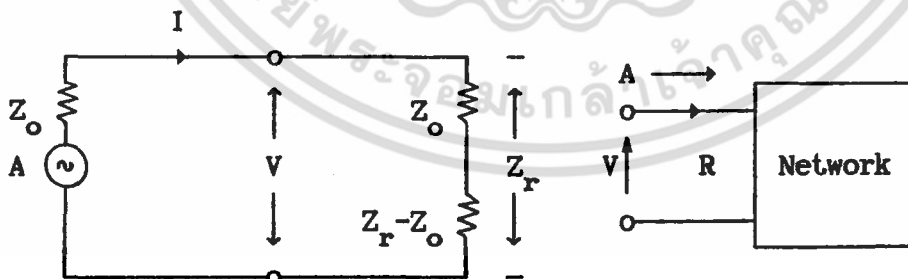
ถ้าหากสมมติให้เน็ทเวกเตอร์ ไม่มีการสูญเสีย คือ $L(\omega_i) = 0$ ที่ $i = 1, 2, \dots$, ดังแสดงในรูปที่ 3.1(b) ที่เหมือนกับคุณลักษณะของ Elliptic [6] ที่ความถี่เท่ากับ ω_i วงจรกรองความถี่จะส่งผ่านกำลังงานให้ทางเอาต์พุตสูงสุด และถ้าการสูญเสียในอิลิเมนต์ต่าง ๆ กำหนดเป็น X_j มีการสูญเสียเพิ่มขึ้น หรือลดลงจากค่า X_j เพียงเล็กน้อย จะทำให้ $L(\omega_i)$ ต้องเพิ่มขึ้นมากกว่าศูนย์ดังแสดงในรูปที่ 3.1(c) เพื่อความกระจำจางสามารถเขียนเป็นสมการได้ดังนี้

$$\lim_{X_j \rightarrow \hat{X}_j} \frac{\Delta L(\omega_i)}{\Delta X_j} = \frac{d L(\omega_i)}{d X_j} = 0$$

ที่ $i = 1, 2, \dots$ และ $j = 1, 2, \dots$ ขึ้นอยู่กับอันดับของวงจรกรองความถี่ [10] เพราะฉะนั้นการสูญเสียในช่วงผ่านความถี่ (Passband) ที่เกิดจากการแปรเปลี่ยนของอิลิเมนต์ในวงจรกรองความถี่แบบ LC นั้น โดยตามธรรมชาติของวงจรกรองความถี่ LC จะมีค่าความไวแปรเปลี่ยนในตัวอิลิเมนต์ต่ำ เพราะฉะนั้นการออกแบบวงจรกรองความถี่ดิจิทัล ที่ได้มาจากวงจรพื้นฐานที่เป็นวงจรกรองความถี่แบบ LC จะทำให้ได้รัวงจรกรองความถี่แบบดิจิทัลที่มีโครงสร้างของค่าความไวแปรเปลี่ยนในตัวอิลิเมนต์ต่ำ เช่นเดียวกับกับวงจรกรองความถี่แบบ LC

3.2 คุณลักษณะของเวฟเน็ทเวกเตอร์ (Wave Network Characterization)

คุณลักษณะของเวฟเน็ทเวกเตอร์ สามารถพิจารณาได้จากรูปที่ 3.2



รูปที่ 3.2 คุณลักษณะของเวฟเน็ทเวกเตอร์

โดยที่ $Z_0 =$ Characteristic Impedance

จากรูปที่ 3.2 จะได้

$$A = IZ_0 + IZ_R \quad \dots(3.1)$$

โดยที่ A เป็น Incident Wave

$$\text{และ } V = IZ_r \quad \dots(3.2)$$

แทนค่าสมการ (3.2) ลงในสมการ (3.1) จะได้

$$A = V + IZ_0 \quad \dots(3.3)$$

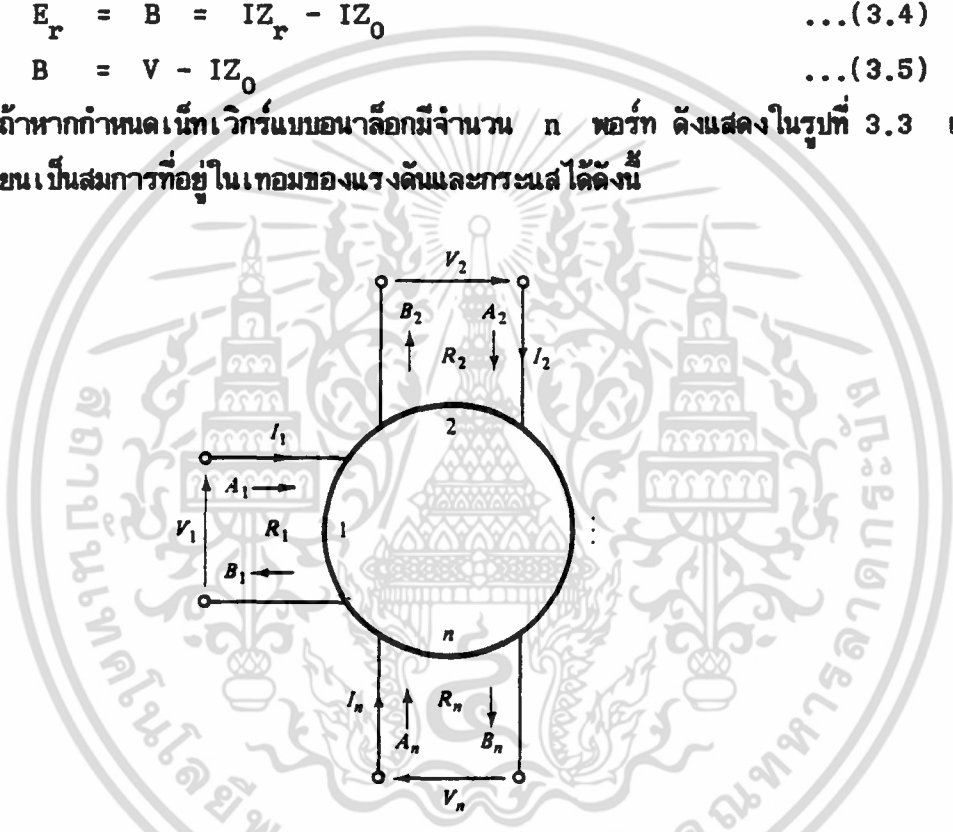
โดยที่ $Z_0 = R$ เรียก Port Resistance

และหา Reflected Wave(B) ได้ โดยการแทน Voltage drop บน $(Z_r - Z_0)$ ด้วย E_r จะได้

$$E_r = B = IZ_r - IZ_0 \quad \dots(3.4)$$

$$\text{หรือ } B = V - IZ_0 \quad \dots(3.5)$$

ถ้าหากกำหนดเน็ทเวอ์กรแบบอนุาล็อกมีจำนวน n พอร์ท ดังแสดงในรูปที่ 3.3 เราสามารถเขียนเป็นสมการที่อยู่ในเทอมของแรงดันและกระแสได้ดังนี้



รูปที่ 3.3 เน็ทเวอ์กร n พอร์ท

$$A_k = V_k + I_k R_k \quad \dots(3.6)$$

$$B_k = V_k - I_k R_k \quad \dots(3.7)$$

โดยที่

$$k = 1, 2, \dots, n$$

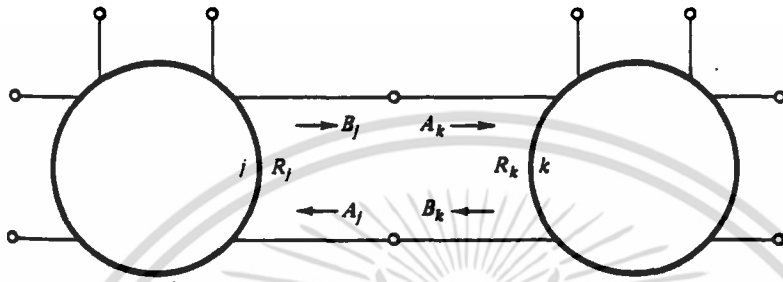
A_k เป็นขนาดของ Incident Wave

B_k เป็นขนาดของ Reflected Wave

R_k เป็น Port Resistance

และถ้าเป็นเน็ทเวอ์กร n พอร์ท 2 ชุด เชื่อมต่อกันดังแสดงในรูปที่ 3.4 จะต้องกำหนดความต้านทานพอร์ทดังนี้

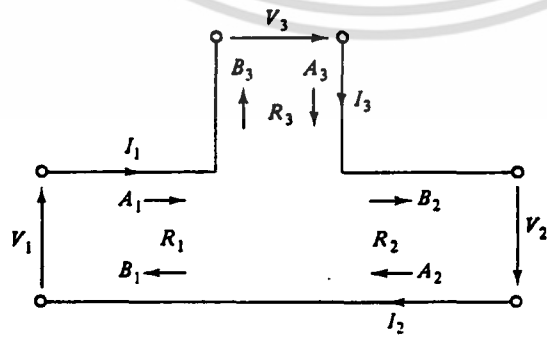
จะได้ $R_j = R_k \dots(3.8)$
 $A_k = B_j$
 $A_j = B_k \dots(3.9)$



รูปที่ 3.4 เน็ตเวิร์ก n พอร์ต เชื่อมต่อกัน

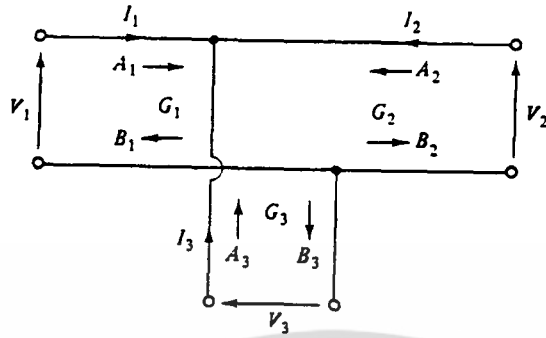
จากกรณีนี้ เราจำเป็นต้องกำหนดความต้านทานร่วมที่จุดเชื่อมต่อของพอร์ทั้ง 2 ที่ต่อ
 ร่วมกัน ดังเช่นลักษณะการไหลของคลื่น ให้เป็นไปอย่างต่อเนื่อง โดย R_k กำหนดได้ดังสมการ
 (3.8) ซึ่งจะเห็นได้ว่าความต้านทานพอร์ที่เชื่อมต่อกันระหว่างพอร์ทั้งสองนั้นเท่ากัน เป็นผลทำ
 ให้ Incident Wave ของพอร์หนึ่งกลายเป็น Reflected Wave ของอีกพอร์หนึ่ง ดังสมการ
 (3.9) และสามารถเชื่อมต่อกันทั้งสองนั้นได้โดยตรง ดังรูปที่ 3.4

โดยทั่วไปแล้วในวงจรของความถี่แบบ LC จะพิจารณาถึงอิมพีแดนซ์ (R, SL หรือ
 1/SC) แหล่งจ่าย (แรงดัน หรือกระแส) และการต่อ 3 พอร์ที่แบบอนุกรม หรือ การต่อ 3
 พอร์ที่แบบขนาน ดังแสดงในรูป 3.5 จากอิลิเมนต์ทางอนุกรมเหล่านี้ ทำการแปลงให้เป็นอิลิ
 เม้นท์ทางดัดจิตอล แล้วแทนด้วยคุณลักษณะของเวฟเน็ตเวิร์กตามชนิดอิลิเมนต์ ก็จะเป็นการสัง
 เคราะห์แบบเวฟดัดจิตอลฟิเตอร์



(a)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



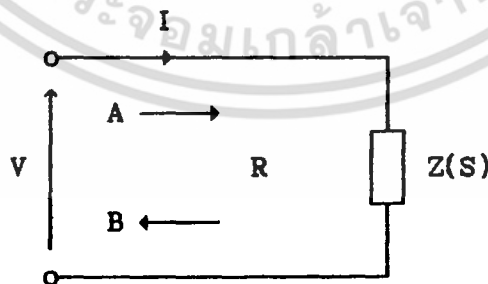
(b)

รูปที่ 3.5 (a) การต่อ 3 พอร์ตแบบอนุกรม
(b) การต่อ 3 พอร์ตแบบขนาน

3.3 การแปลงอิมพีแดนซ์ (Element Realizations)

ในการออกแบบวงจรเวฟดิคตอลฟิลเตอร์ จะต้องแปลงอิมพีแดนซ์ทางอนาล็อกให้อยู่ในรูปของอิมพีแดนซ์ทางดิจิทัล ขั้นแรกให้พิจารณาอิมพีแดนซ์ที่อยู่ในเทอมของคัมพลิกซ์ของเวฟเน็ทเวกเตอร์ ขั้นที่สองจัดตัวแปร V_k, I_k และ S โดยใช้สมการลูปและสมการโหนด (Loop and Node Equations) และใช้การแปลงแบบไบเลเนียน์ทรานส์ฟอร์ม $S=(2/T).(Z-1)/(Z+1)$ ขั้นที่สามหาขนาดของ Reflected Wave ให้อยู่ในฟังก์ชันของขนาด Incident Wave ขั้นที่สี่สร้างวงจรจริงขึ้นจากสมการที่ได้ โดยใช้ยูนิตดีเลย์ วงจรรวม อินเวอร์เตอร์ และวงจรคูณ ตามแต่ละชนิดของอิมพีแดนซ์ดังต่อไปนี้

อิมพีแดนซ์ (Impedance)



รูปที่ 3.6 Impedance

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิจารณาอิมพีแดนซ์

$$Z(S) = S^\lambda R_x \quad \dots(3.10)$$

โดยที่ R_x เป็นค่าคงที่บวก
 $\lambda = -1$ ในกรณีที่ เป็น คาปาซิแตนซ์
 $\lambda = 0$ ในกรณีที่ เป็น ความต้านทาน
 $\lambda = 1$ ในกรณีที่ เป็น อินдукแตนซ์

จากสมการ (3.6), (3.7) และ รูปที่ 3.6 จะได้

$$A = V + IR \quad \dots(3.11)$$

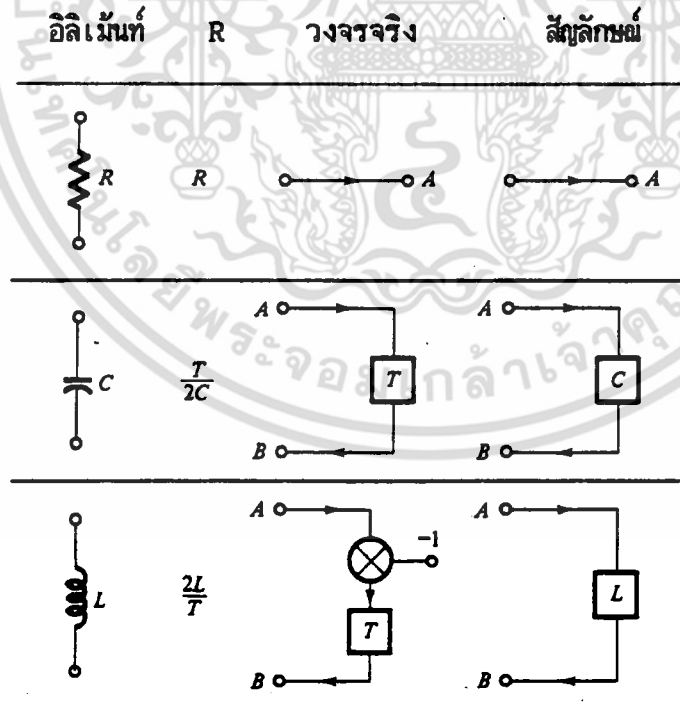
$$B = V - IR \quad \dots(3.12)$$

และ $V = IZ(S) \quad \dots(3.13)$

แทนค่าสมการ (3.13) ลงในสมการ (3.11) และ (3.12) จะได้

$$B = f(Z)A \quad \dots(3.14)$$

โดยที่ $f(Z) = \frac{Z(S)-R}{Z(S)+R} \Big|_S = \frac{2(Z-1)}{T(Z+1)} \quad \dots(3.15)$



รูปที่ 3.7 การแปลงอิมพีแดนซ์จากอิมพีแดนซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดให้ $R = \left(\frac{2}{T}\right)^\lambda R_x \dots(3.16)$

แทนค่าสมการ (3.10) และ (3.16) ในสมการ (3.15) จะได้

$$f(Z) = \begin{cases} Z^{-1} & \text{ในกรณีที่ } \lambda = -1 \\ 0 & \text{ในกรณีที่ } \lambda = 0 \\ -Z^{-1} & \text{ในกรณีที่ } \lambda = 1 \end{cases}$$

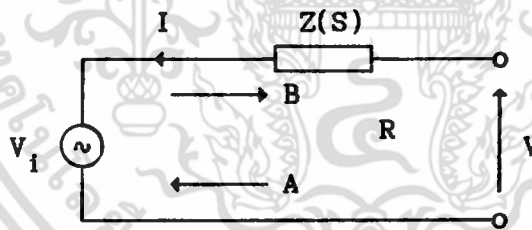
จากสมการที่ (3.14) และค่า $f(Z)$ ที่หาได้ แสดงให้เห็นถึงผลที่ได้จากการแปลงอิลิเมนต์ ซึ่งสามารถแสดงได้ดังรูปที่ 3.7 จากรูปจะเห็นได้ว่าความต้านทานนั้นจะเปลี่ยนไปเป็น Digital Sink คาปาซิแตนซ์จะเปลี่ยนเป็นดีเลย์ยูนิต และอินดักแตนซ์เปลี่ยนเป็นดีเลย์ยูนิตต่อคาสเคดกับอินเวอร์เตอร์

แหล่งจ่ายแรงดัน (Voltage Sources)

พิจารณาแหล่งจ่ายแรงดัน ในรูปที่ 3.8 จะได้

$$V = IZ(S) + V_1 \dots(3.17)$$

แทนค่าสมการ (3.17) ในสมการ (3.11) และ (3.12) จะได้



รูปที่ 3.8 แหล่งจ่ายแรงดัน

$$B = f_1(Z)V_1 + f_2(Z)A \dots(3.18)$$

โดยที่ $f_1(Z) = \frac{2R}{R+Z(S)} \Big|_{S = \frac{2(Z-1)}{T(Z+1)}}$

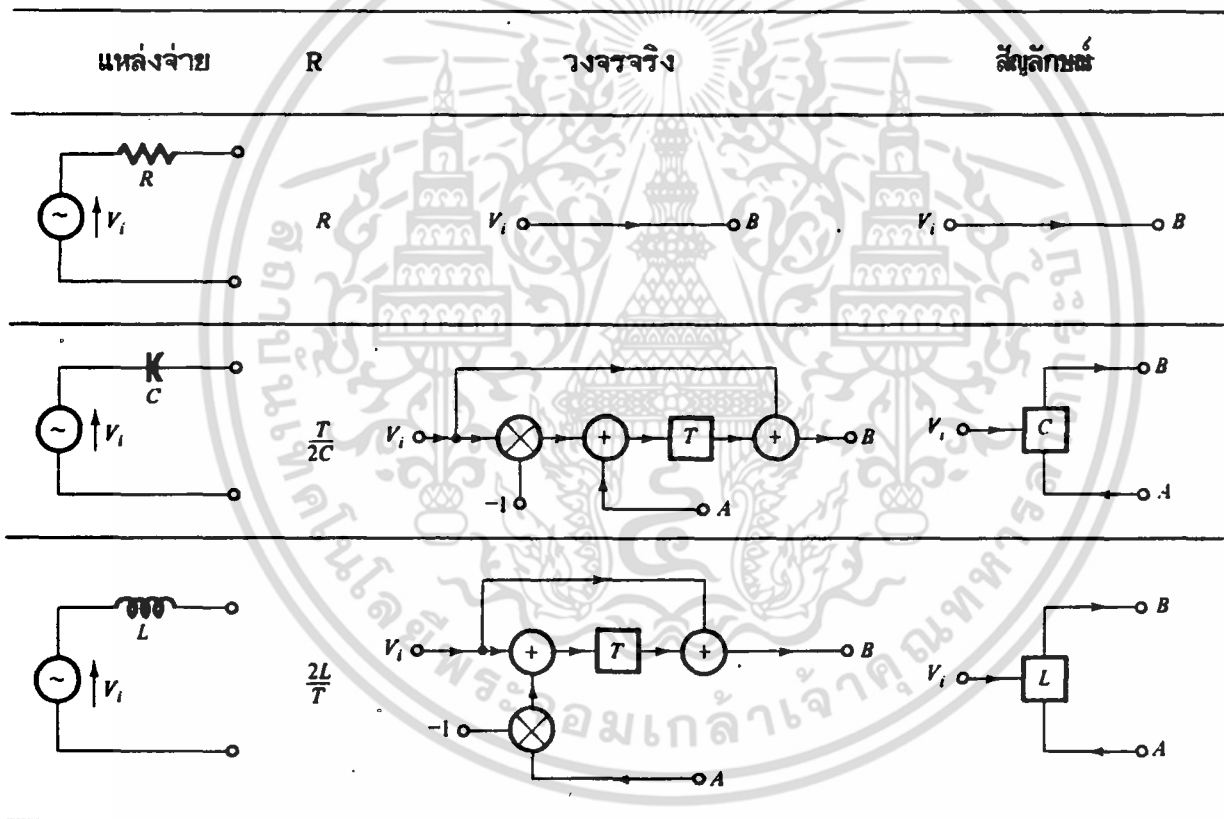
$$f_2(Z) = \frac{Z(S)-R}{Z(S)+R} \Big|_{S = \frac{2(Z-1)}{T(Z+1)}} \dots(3.19)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แทนค่าสมการ (3.10) และ (3.16) ในสมการ (3.19) จะได้

$$f_1(Z) = \begin{cases} 1-Z^{-1} & \text{ในกรณีที่ } \lambda = -1 \\ 1 & \text{ในกรณีที่ } \lambda = 0 \\ 1+Z^{-1} & \text{ในกรณีที่ } \lambda = 1 \end{cases}$$

$$f_2(Z) = \begin{cases} Z^{-1} & \text{ในกรณีที่ } \lambda = -1 \\ 0 & \text{ในกรณีที่ } \lambda = 0 \\ -Z^{-1} & \text{ในกรณีที่ } \lambda = 1 \end{cases}$$



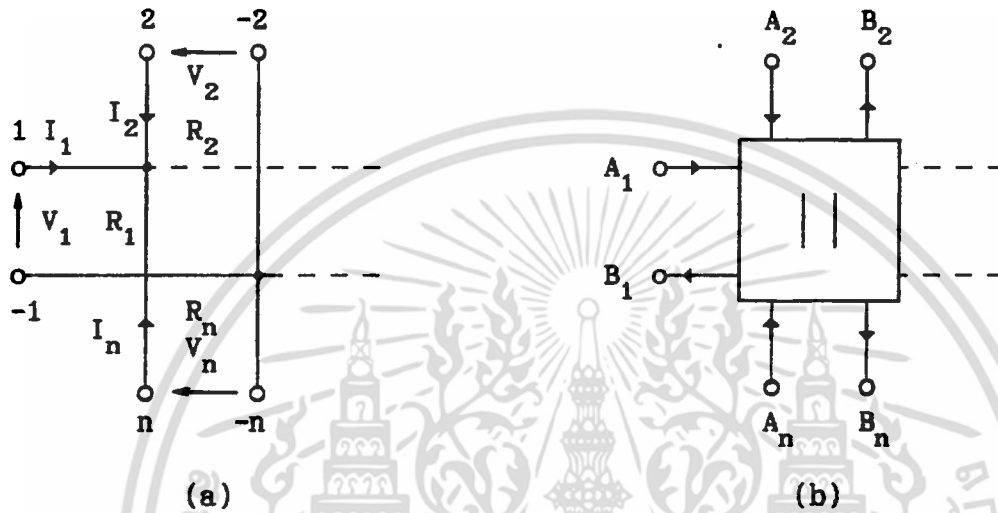
รูปที่ 3.9 การแปลงอิลิเมนต์จากแหล่งจ่ายแรงดัน

จากสมการ (3.18) และค่า $f(Z)$ ที่ได้ แสดงให้เห็นถึงผลที่ได้จากการแปลงแหล่งจ่ายที่ต่อร่วมกับความต้านทาน คาปาซิแตนซ์ และอินดักแตนซ์ ดังแสดงในรูปที่ 3.9 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อะแดปเตอร์แบบขนาน

พิจารณาอะแดปเตอร์ n พอร์ต(1,2,...,n) ที่มีความต้านทานพอร์ตเป็น R_1, R_2, \dots, R_n ตามลำดับ ดังแสดงในรูปที่ 3.10 จากคุณสมบัติของการต่อแบบขนานจะเป็นดังนี้



รูปที่ 3.10 (a) อะแดปเตอร์แบบขนาน n พอร์ต (b) สัญญาณ

$$V_1 = V_2 = V_3 = \dots = V_n \quad \dots(3.20)$$

$$I_1 + I_2 + I_3 + \dots + I_n = 0 \quad \dots(3.21)$$

เมื่อแทนด้วยคุณลักษณะของเวฟเน็ทเวียร์ จากสมการ (3.6),(3.7),(3.20) และ (3.21) สามารถหาสมการของอะแดปเตอร์แบบขนานที่อยู่ในเทอมของ Incident Wave และ Reflected Wave ได้ดังนี้ [8]

$$B_k = (\alpha_1 A_1 + \alpha_2 A_2 + \dots + \alpha_n A_n) - A_k \quad \dots(3.22)$$

โดยที่
$$\alpha_k = \frac{2 G_k}{G_1 + G_2 + \dots + G_n} \quad \dots(3.23)$$

$$G_k = \frac{1}{R_k} \quad \dots(3.24)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\alpha_1 + \alpha_2 + \dots + \alpha_n = 2 \quad \dots(3.25)$$

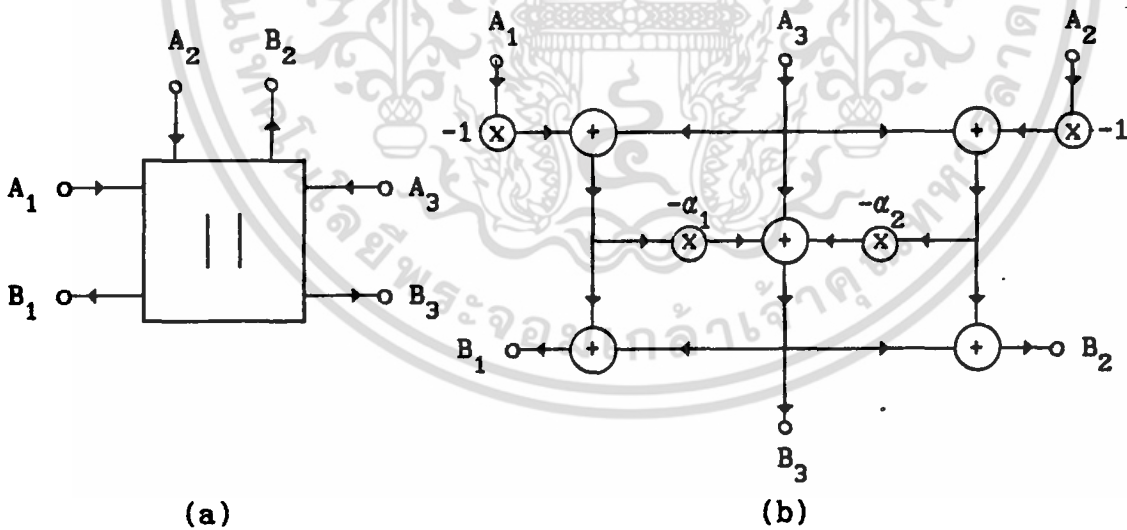
โดยที่ $k = 1, 2, 3, \dots, n$

จากสมการ (3.25) สัมประสิทธิ์ α_k ของพอร์ทนั้นจะมีพอร์ทหนึ่งๆที่เรียกว่า Dependent Port ซึ่งสามารถตัดสัมประสิทธิ์ของพอร์ทนั้นได้ แล้วจะทำให้ต้องใช้ตัวรวมสัญญาณในสมการน้อยที่สุด โดยการเลือกพอร์ท n เป็น Dependent Port จากสมการ (3.22) สามารถเขียนใหม่ได้เป็น

$$B_n = A_n - \sum_{k=1}^{n-1} \alpha_k (A_n - A_k) \quad \dots(3.26)$$

$$B_k = B_n + (A_n - A_k), \quad (k = 1, 2, \dots, n-1) \quad \dots(3.27)$$

ผลที่ได้จากสมการ (3.26), (3.27) จะเป็นสมการอะแดปเตอร์แบบขนานทั่วไปที่มีจำนวนพอร์ท n พอร์ท ต้องใช้ตัวคูณสัญญาณ $n-1$ ตัว และต้องมีตัวรวมสัญญาณ $3n-3$ ตัว ถ้าหากกำหนดให้ $n = 3$ จะเขียนเป็นไดอะแกรมทางเดินของสัญญาณได้ดังแสดงในรูปที่ 3.11(b) ส่วนสัญลักษณ์แสดงไว้ในรูปที่ 3.11(a) จากรูปจะเห็นได้ว่าต้องใช้ตัวรวมสัญญาณ 5 ตัว มี 1 ตัวใน 5 ตัวนั้นมีอินพุตถึง 3 อินพุต ซึ่งที่จริงแล้วต้องใช้ตัวรวมสัญญาณ 6 ตัว



รูปที่ 3.11 3 พอร์ทอะแดปเตอร์ขนาน กำหนดพอร์ท 3 เป็น Dependent Port
(a) สัญลักษณ์ (b) ไดอะแกรมทางเดินของสัญญาณ

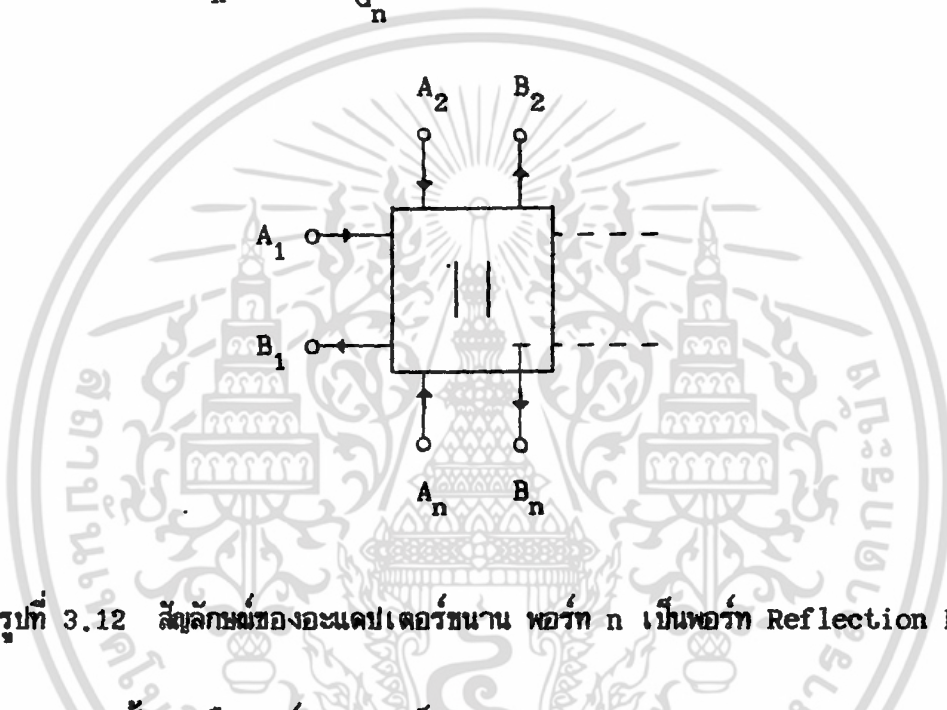
ในกรณีที่กำหนดพอร์ท n ของอะแดปเตอร์แบบขนานเป็นพอร์ท Reflection Free ดังสัญลักษณ์ในรูปที่ 3.12 แล้วจะได้ [9]

$$\alpha_n = 1$$

$$G_n = G_1 + G_2 + \dots + G_{n-1} \quad \dots(3.28)$$

$$\alpha_1 + \alpha_2 + \dots + \alpha_{n-1} = 1 \quad \dots(3.29)$$

$$\alpha_k = \frac{G_k}{G_n} \quad \dots(3.30)$$



รูปที่ 3.12 สัญลักษณ์ของอะแดปเตอร์ขนาน พอร์ท n เป็นพอร์ท Reflection Free

จากนั้น เลือกพอร์ท $n-1$ เป็น Dependent Port จากสมการ (3.26) และ (3.27) สามารถเขียนใหม่ได้เป็น

$$B_0 = - \sum_{k=1}^{n-2} \alpha_k (A_{n-1} - A_k) \quad \dots(3.31)$$

$$B_{n-1} = B_0 + A_n \quad \dots(3.32)$$

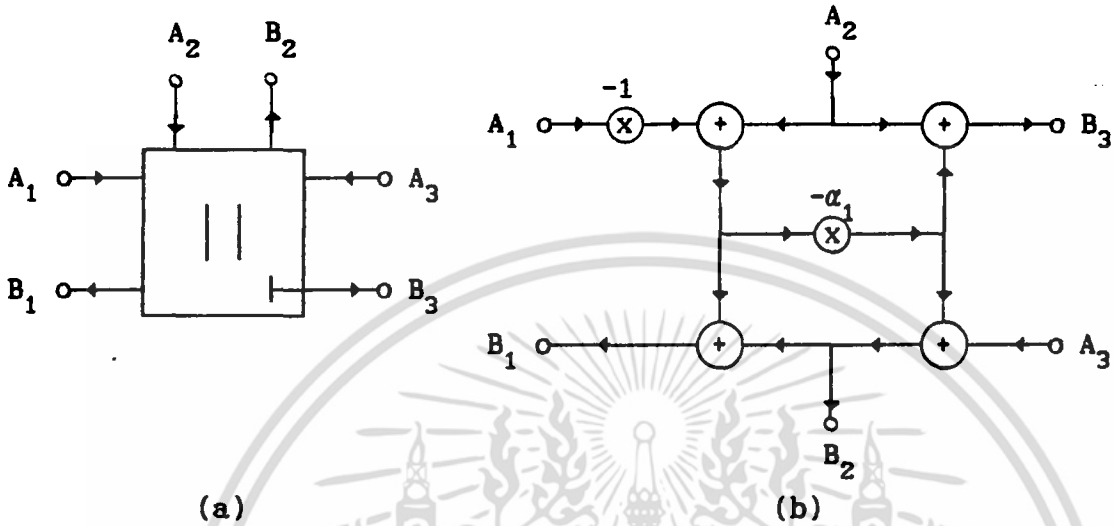
$$B_n = B_0 + A_{n-1} \quad \dots(3.33)$$

$$B_k = B_{n-1} + (A_{n-1} - A_k) \quad \dots(3.34)$$

โดยที่ $k = 1, 2, \dots, n-2$

ซึ่งต้องการตัวคูณสัญญาณเพียง $n-2$ ตัว และตัวรวมสัญญาณเพียง $3n-5$ ตัว 3 พอร์ท อะแดปเตอร์แบบขนาน ที่กำหนดพอร์ท 3 เป็น Reflection Free และพอร์ท 2 เป็น

Dependent Port แสดงไว้ในรูปที่ 3.13

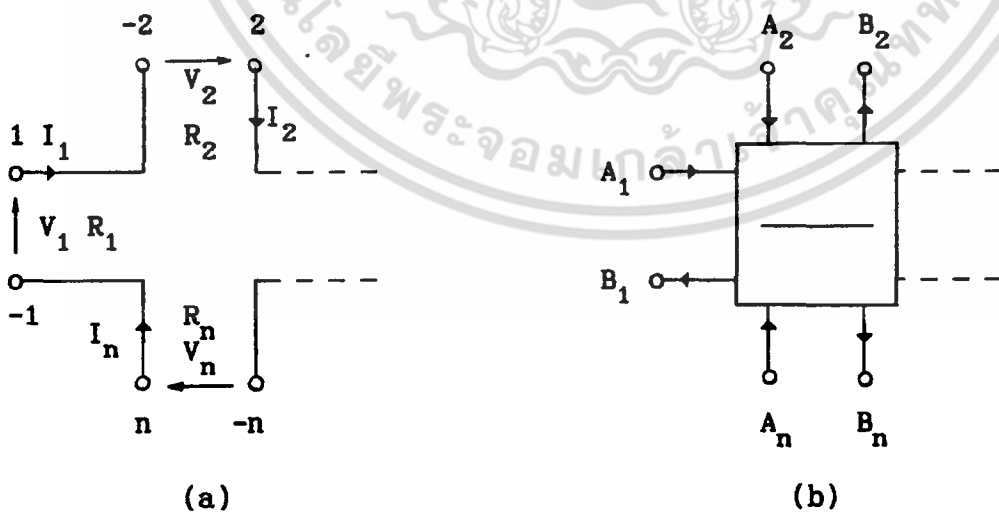


รูปที่ 3.13 3 พอร์ตอะเนลปเตอร์ขนาน โดยกำหนดพอร์ท 3 เป็นพอร์ท Reflection Free และพอร์ท 2 เป็น Dependent Port

(a) สัญลักษณ์ (b) ไดอะแกรมทางเดินของสัญญาณ

อะเนลปเตอร์แบบอนุกรม

พิจารณาอะเนลปเตอร์ n พอร์ต $(1, 2, \dots, n)$ ซึ่งมีความต้านทานพอร์ทเป็น R_1, R_2, \dots, R_n ตามลำดับ ดังแสดงในรูป 3.14 จากคุณสมบัติของการต่อแบบอนุกรมจะเป็นดังนี้



รูปที่ 3.14 (a) อะเนลปเตอร์แบบอนุกรม n พอร์ต (b) สัญลักษณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$I_1 = I_2 = I_3 = \dots = I_n \quad \dots(3.35)$$

$$V_1 + V_2 + V_3 + \dots + V_n = 0 \quad \dots(3.36)$$

จากสมการ (3.6), (3.7), (3.35) และ (3.36) สามารถหาสมการของอะนัลเตอร์แบบอนุกรมที่อยู่ในเทอมของ Incident Wave และ Reflected Wave ได้ดังนี้

$$B_k = A_k - \beta_k (A_1 + A_2 + \dots + A_n) \quad \dots(3.37)$$

โดยที่

$$\beta_k = \frac{2 R_k}{R_1 + R_2 + \dots + R_n} \quad \dots(3.38)$$

$$\beta_1 + \beta_2 + \beta_3 + \dots + \beta_n = 2 \quad \dots(3.39)$$

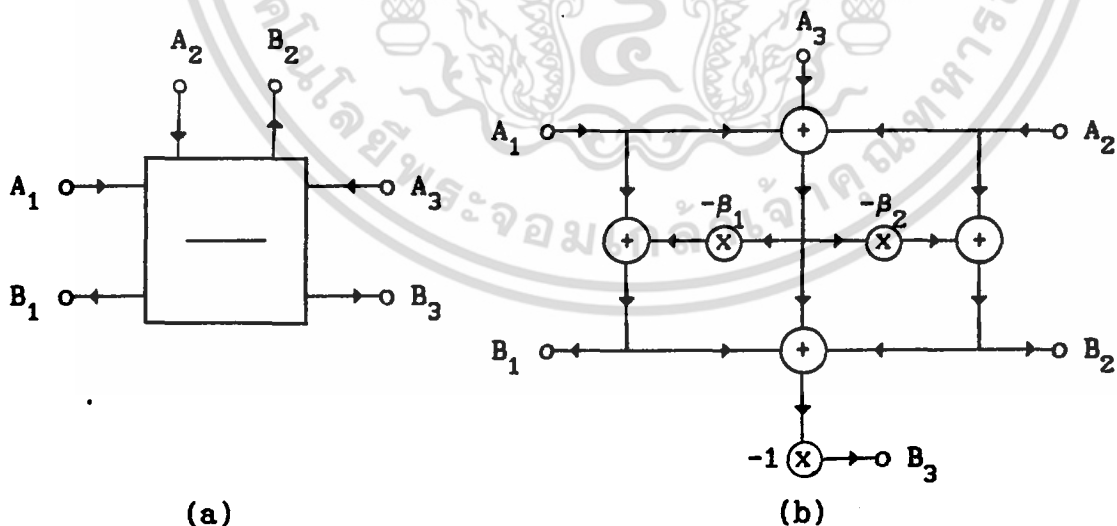
ในทำนองเช่นเดียวกันกับอะนัลเตอร์แบบขนาน ถ้าหากเลือกพอร์ต n เป็น Dependent Port จากสมการ (3.37) จะเขียนใหม่ได้เป็น

$$A_0 = A_1 + A_2 + \dots + A_n \quad \dots(3.40)$$

$$B_k = A_k - \beta_k A_0, \quad (k = 1, 2, \dots, n-1) \quad \dots(3.41)$$

$$B_n = -(B_1 + B_2 + \dots + B_{n-1} + A_0) \quad \dots(3.42)$$

จากสมการ (3.40), (3.41) และ (3.42) เป็นสมการอะนัลเตอร์แบบอนุกรมทั่วไปที่ต้องการตัวคุณสัญญาณ $n-1$ ตัว และตัวรวมสัญญาณ $3n-3$ ตัว เช่นเดียวกันกับอะนัลเตอร์แบบขนาน ถ้ากำหนด $n = 3$ จะได้ไดอะแกรมทางเดินของสัญญาณดังแสดงในรูปที่ 3.15(b) จะต้องใช้ตัวรวมสัญญาณ 6 ตัว จากรูปใช้เพียง 4 ตัว แต่มี 2 ตัวที่มีอินพุต 3 อินพุต ส่วนรูปที่ 3.15(a) แสดงสัญลักษณ์



รูปที่ 3.15 3 พอร์ตอะนัลเตอร์อนุกรม กำหนดพอร์ต 3 เป็น Dependent Port
(a) สัญลักษณ์ (b) ไดอะแกรมทางเดินของสัญญาณ

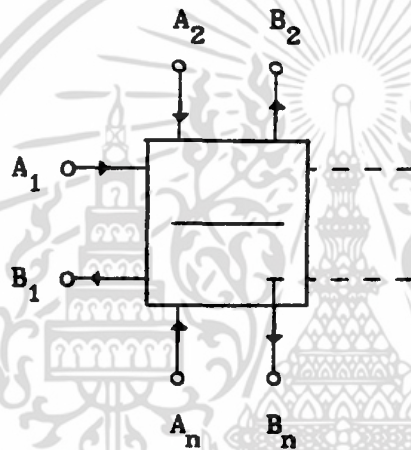
ในกรณีที่กำหนดพอร์ต n เป็นพอร์ต Reflection Free ดังสัญลักษณ์ในรูป 3.16 จะ
ได้ [9]

$$\beta_n = 1$$

$$R_n = R_1 + R_2 + \dots + R_{n-1} \quad \dots(3.43)$$

$$\beta_1 + \beta_2 + \dots + \beta_{n-1} = 1 \quad \dots(3.44)$$

$$\beta_k = \frac{R_k}{R_n} \quad \dots(3.45)$$



รูปที่ 3.16 สัญลักษณ์อะแดปเตอร์แบบอนุกรม กำหนดพอร์ต n เป็นพอร์ต Reflection Free

ถ้าหากเราเลือกพอร์ต $n-1$ เป็น Dependent Port จากสมการ (3.40), (3.41) และ (3.42) จะเขียนใหม่ได้ว่า

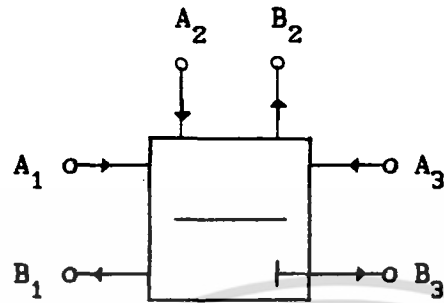
$$B_n = -(A_1 + A_2 + \dots + A_{n-1}) \quad \dots(3.46)$$

$$A_0 = A_n - B_n \quad \dots(3.47)$$

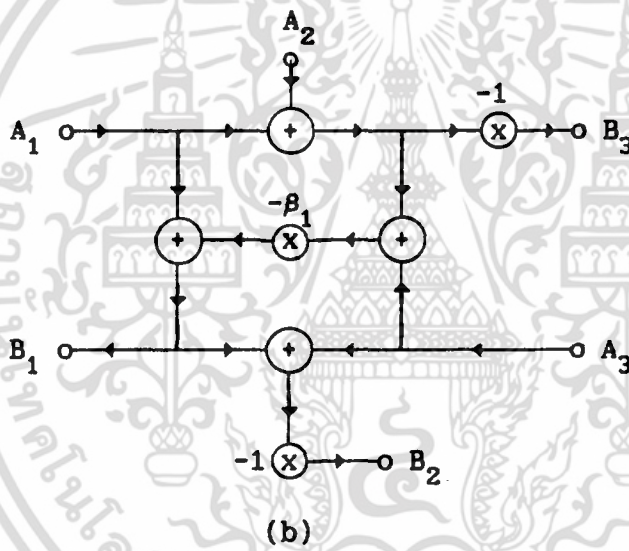
$$B_k = A_k - \beta_k A_0, (k=1, 2, \dots, n-2) \quad \dots(3.48)$$

$$B_{n-1} = -(B_1 + B_2 + \dots + B_{n-2} + A_n) \quad \dots(3.49)$$

จะเห็นได้ว่าต้องการตัวคูณสัญญาณเพียง $n-2$ ตัว และตัวรวมสัญญาณเพียง $3n-5$ ตัว เช่นเดียวกับกับอะแดปเตอร์แบบขนาน 3 พอร์ต อะแดปเตอร์แบบอนุกรม ที่กำหนดพอร์ต 3 เป็น Reflection Free และพอร์ต 2 เป็น Dependent Port แสดงได้ดังรูปที่ 3.17



(a)



(b)

รูปที่ 3.17 3 พอร์ตอะเนคไดอะแกรม กำหนดพอร์ต 3 เป็นพอร์ต Reflection Free และพอร์ต 2 เป็น Dependent Port

(a) สัญลักษณ์ (b) ไดอะแกรมทางเดินของสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การออกแบบวงจรกรองความถี่ต่ำแบบคอม โพลี โพลี โนเมียล

วงจรกรองความถี่ต่ำ ในอุดมคติ นั้น จะเป็นวงจรที่ยอมให้ช่วงความถี่ต่ำที่ต้องการผ่านสัญญาณออกไปได้ทั้งหมด โดย ไม่มีการสูญเสีย หรือผิดเพี้ยน และจะกำจัดสัญญาณในช่วงความถี่สูง (ย่านหยุดสัญญาณ) ได้ทั้งหมด ถ้าให้สัญญาณที่ป้อนเข้าทางอินพุตเป็น V_1 และให้สัญญาณที่ออกทางเอาต์พุตเป็น V_2 อัตราส่วนระหว่าง V_2 ต่อ V_1 จะเป็นทรานส์เฟอร์ฟังก์ชันของวงจร ซึ่งสามารถเขียนได้เป็น

$$\frac{V_2(S)}{V_1(S)} = H(S) = \frac{N(S)}{D(S)} \quad \dots(4.1)$$

เมื่อ $H(S)$ เป็นทรานส์เฟอร์ฟังก์ชันของวงจร

$N(S)$ และ $D(S)$ เป็นโพลีโนเมียลของตัวเศษและโพลีโนเมียลของตัวส่วนตามลำดับ ซึ่งเขียนอยู่ในรูปของ Laplace Transforms

เนื่องจากวงจรใด ๆ สามารถแทนได้แต่เฉพาะฟังก์ชันที่อยู่ในรูปของ โพลีโนเมียลเศษส่วนเท่านั้น จากสมการ (4.1) จะได้ $D(S)$ เป็นเฮอวิทซ์โพลีโนเมียล (Hurwitz Polynomials) แต่ในทางปฏิบัติฟังก์ชันที่อยู่ในรูปโพลีโนเมียลเศษส่วนนั้น ไม่สามารถที่จะเป็นไปได้ตามอุดมคติที่กล่าวมา ฉะนั้นวงจรที่มีคุณสมบัติที่คั่น ค่าโพลีโนเมียลฟังก์ชันของวงจร เมื่อนำไปเทียบเคียงกับฟังก์ชันของวงจรกรองความถี่ต่ำ ในอุดมคติ จะต้องให้ค่าที่ใกล้เคียงกับฟังก์ชันในอุดมคติมากที่สุด ซึ่งโพลีโนเมียลที่นิยมใช้ในการออกแบบวงจรกรองความถี่ต่ำ โดยทั่วไปจะมีดังนี้ คือ บัตเตอร์เวิร์ทโพลีโนเมียล (Butterworth Polynomials) [12,18] จะให้ผลตอบสนองทางขนาดในช่วงความถี่ต่ำ(ย่านผ่านสัญญาณ)ราบเรียบที่สุด (Maximally Flat Amplitude) ส่วน เชบิเชฟโพลีโนเมียล (Chebyshev Polynomials) [16,18] จะให้ผลตอบสนองทางขนาดในย่านคัทออฟที่ชัน และเบสเซลโพลีโนเมียล (Bessel Polynomials)[12] ซึ่งจะให้ผลตอบสนองของกรุปดีเลย์ราบเรียบที่สุด (Maximally Flat Group Delay) [2] นอกจากนี้ยังมีโพลีโนเมียลอีกชนิดหนึ่งที่ไม่ได้ใช้ในวงจรกรองความถี่โดยทั่วไป แต่มีคุณสมบัติของผลตอบสนองทางขนาดในย่านคัทออฟที่ชันกว่าเชบิเชฟโพลีโนเมียล ซึ่งก็คืออัลตราสเฟียลโพลีโนเมียล (Ultraspherical Polynomials) [11,13] นอกจากนั้นแล้วยังมีพหุนามิเตอร์ที่สามารถกำหนดผลตอบสนองทางขนาดเปลี่ยนแปลงตามที่ต้องการได้ รายละเอียดของโพลีโนเมียลแต่ละแบบจะได้กล่าวถึงต่อไป

4.1 วงจรกรองความถี่ต่ำแบบบัตเตอร์โวลต์

วงจรกรองความถี่ต่ำแบบบัตเตอร์โวลต์เฟอ์ฟังก์ชัน ซึ่งเขียนในรูปของขนาดกำลังสอง (Magnitude Square) คือ

$$H(S).H(-S)|_{s=j\omega} = |H_n(j\omega)|^2 = \frac{1}{1 + \omega^{2n}} \quad \dots(4.2)$$

โดยที่ n เป็นอันดับ (order) ของทรานส์เฟอ์ฟังก์ชัน

ถ้ากำหนดให้ loss = $|B_n(j\omega)|^2$ จะได้

$$|B_n(j\omega)|^2 = \frac{1}{|H_n(j\omega)|^2} = 1 + \omega^{2n} \quad \dots(4.3)$$

จากสมการ (4.3) แทน $S = j\omega$; $\omega = -Sj$ จะได้

$$|B_n(j\omega)|^2 = 1 + (-S^2)^n \quad \dots(4.4)$$

จากสมการ (4.4) หารากของสมการได้โดย

$$\begin{aligned} 1 + (-1)^n(S^{2n}) &= 0 \\ (-1)^n S^{2n} &= -1 = e^{j(2k-1)\pi} \quad \dots(4.5) \end{aligned}$$

โดยที่ $(-1)^n = e^{jn\pi}$ จะได้

$$S^{2n} = e^{j(2k+n-1)\pi} \quad \dots(4.6)$$

จะหารากของสมการหาได้จาก

$$S_k = e^{j[\pi/2 \cdot (2k+n-1)/n]} \quad \dots(4.7)$$

โดยที่ $k = 1, 2, \dots, n$

ตัวอย่าง ถ้า $n = 2$ เราสามารถหารากจากสมการ (4.7) ได้ดังนี้

$$\begin{aligned} H_2(S) &= \frac{1}{(S+S_1)(S+S_2)} \\ S_1 &= e^{j3\pi/4} = -0.707 + j0.707 \\ S_2 &= e^{j5\pi/4} = -0.707 - j0.707 \end{aligned}$$

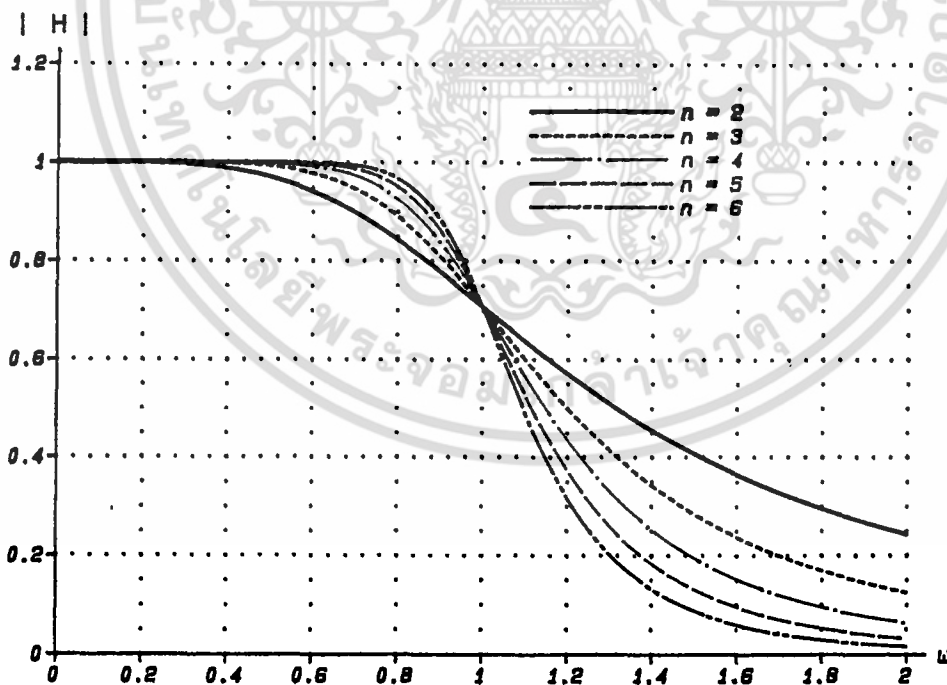
$$\begin{aligned} H_2(S) &= \frac{1}{(S+0.707-j0.707)(S+0.707+j0.707)} \\ &= \frac{1}{S^2 + \sqrt{2}S + 1} \quad \dots(4.8) \end{aligned}$$

จากสมการ (4.8) เป็นทรานส์เฟอ์ฟังก์ชันของวงจรกรองความถี่ต่ำแบบบัตเตอร์โวลต์ เมื่อกำหนดให้ $n = 2$ และจากสมการ (4.7) ถ้าแทน $n = 2$ ถึง 6 จะได้สัมประสิทธิ์ของ

โพลีโนเมียลดังแสดงในตารางที่ 4.1 จากรูปที่ 4.1 แสดงผลตอบสนองทางขนาดของวงจรถองความถี่ต่ำแบบบัตเตอร์เวิร์ท

ตารางที่ 4.1 แสดงสัมประสิทธิ์ของบัตเตอร์เวิร์ทโพลีโนเมียล $B_n(s) = s^n + \sum_{i=0}^{n-1} a_i s^i$

n	a ₀	a ₁	a ₂	a ₃	a ₄	a ₅
2	1	1.4142				
3	1	2	2			
4	1	2.6131	3.4142	2.6131		
5	1	3.2361	5.2361	5.2361	3.2361	
6	1	3.8637	7.4641	9.1416	7.4641	3.8637



รูปที่ 4.1 ผลตอบสนองทางขนาดของวงจรถองความถี่ต่ำแบบบัตเตอร์เวิร์ท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 วงจรรองความถี่ต่ำแบบอูลตราสเฟียวิคัล โพลีโนเมียล

วงจรรองความถี่ต่ำแบบอูลตราสเฟียวิคัล โพลีโนเมียล มีทรานส์เฟอร์ฟังก์ชันซึ่งเขียนอยู่ในรูปของขนาดกำลังสอง (Magnitude Square) คือ

$$H(S).H(-S)|_{s=j\omega} = |H_n(j\omega)|^2 = \frac{1}{1 + U_n^\alpha(\omega^2)} \quad \dots(4.9)$$

โดยที่ $U_n^\alpha(\omega^2)$ เป็นขนาดกำลังสอง (Magnitude Square) ของอูลตราสเฟียวิคัล โพลีโนเมียล และ n เป็นอันดับของทรานส์เฟอร์ฟังก์ชัน

สูตรทั่วไปของอูลตราสเฟียวิคัล โพลีโนเมียล [13] $U_n^\alpha(\omega)$ ที่ใช้คือ

$$U_n^\alpha(\omega) = \frac{n!}{(1+\alpha)_n} P_n^{(\alpha, \alpha)}(\omega) \quad \dots(4.10)$$

$$\text{โดยที่ } (1+\alpha)_n = (1+\alpha)(2+\alpha)\dots(n+\alpha) \quad \dots(4.11)$$

$$n = 1, 2, 3, \dots; \alpha > -1$$

α เป็นค่าพารามิเตอร์ของอูลตราสเฟียวิคัล โพลีโนเมียล และ โพลีโนเมียล $P_n^{(\alpha, \alpha)}(\omega)$ ได้มาจาก Jacobi โพลีโนเมียล โดยที่ Jacobi โพลีโนเมียล $P_n^{(\alpha, \beta)}(\omega)$ กำหนดได้ดังนี้ [13]

$$P_n^{(\alpha, \beta)}(\omega) = \frac{(1+\alpha)_n}{n!} {}_2F_1 \left(\begin{matrix} -n, \alpha+\beta+n+1 \\ 1+\alpha \end{matrix}; \frac{1-\omega}{2} \right) \quad \dots(4.12)$$

โดยที่ ${}_2F_1$ เป็น Hypergeometric ฟังก์ชัน [13] ซึ่งมีสูตรทั่วไปดังนี้

$${}_2F_1 \left(\begin{matrix} a, b \\ c \end{matrix}; x \right) = \sum_{k=0}^n \frac{(a)_k (b)_k x^k}{k! (c)_k} \quad \dots(4.13)$$

เมื่อ $\beta = \alpha = 0$ ค่าโพลีโนเมียลในสมการ (4.12) กลายเป็น Legendre โพลีโนเมียล จากสมการ (4.12) จะเห็นได้ว่า ค่า $P_n^{(\alpha, \beta)}(\omega)$ เป็นโพลีโนเมียลของอันดับที่ n และเมื่อ $\omega = 1$ จะได้

$$P_n^{(\alpha, \beta)}(\omega) = P_n^{(\alpha, \beta)}(1) = \frac{(1+\alpha)_n}{n!} \quad \dots(4.14)$$

ในกรณีของ Jacobi โพลิโนเมียล ค่าของ ${}_2F_1$ สามารถประยุกต์ได้ดังสมการต่อไปนี้

$$P_n^{(\alpha, \beta)}(\omega) = \frac{(1+\alpha)_n}{n!} \left(\frac{\omega+1}{2}\right)^n {}_2F_1 \left(\begin{matrix} -n, -\beta+n \\ 1+\alpha \end{matrix}; \frac{\omega-1}{\omega+1} \right) \dots(4.15)$$

$$P_n^{(\alpha, \beta)}(\omega) = \frac{(-1)^n (1+\beta)_n}{n!} {}_2F_1 \left(\begin{matrix} -n, \beta+\alpha+n+1 \\ 1+\beta \end{matrix}; \frac{1+\omega}{2} \right) \dots(4.16)$$

จากสมการ (4.12) , (4.15) และ (4.16) ถ้าให้ผลลัพธ์อยู่ในรูปของ Finite Series สำหรับค่า $P_n^{(\alpha, \beta)}(\omega)$ จะเขียนได้ใหม่ดังสมการ (4.17) , (4.18) และ (4.19)

$$P_n^{(\alpha, \beta)}(\omega) = \sum_{k=0}^n \frac{(1+\alpha)_n (1+\alpha+\beta)_{n+k}}{k!(n-k)!(1+\alpha)_k (1+\alpha+\beta)_n} \left(\frac{\omega-1}{2}\right)^k \dots(4.17)$$

$$P_n^{(\alpha, \beta)}(\omega) = \sum_{k=0}^n \frac{(1+\alpha)_n (1+\beta)_n}{k!(n-k)!(1+\alpha)_k (1+\beta)_{n-k}} \left(\frac{\omega-1}{2}\right)^k \left(\frac{\omega+1}{2}\right)^{n-k} \dots(4.18)$$

$$P_n^{(\alpha, \beta)}(\omega) = \sum_{k=0}^n \frac{(-1)^{n-k} (1+\beta)_n (1+\alpha+\beta)_{n+k}}{k!(n-k)!(1+\beta)_k (1+\alpha+\beta)_n} \left(\frac{\omega+1}{2}\right)^k \dots(4.19)$$

ในกรณีที่ n กับสมการที่มี order มาก จากสมการ (4.17), (4.18) และ (4.19) สามารถเขียนใหม่ได้ดังสมการ (4.20), (4.21) และ (4.22) ตามลำดับ

$$P_n^{(\alpha, \beta)}(\omega) = \frac{(1+\alpha+\beta)_{2n}}{n!(1+\alpha+\beta)_n} \left(\frac{\omega-1}{2}\right)^n {}_2F_1 \left(\begin{matrix} -n, -\alpha-n \\ -\alpha-\beta-2n \end{matrix}; \frac{2}{1-\omega} \right) \dots(4.20)$$

$$P_n^{(\alpha, \beta)}(\omega) = \frac{(1+\beta)_n}{n!} \left(\frac{\omega-1}{2}\right)^n {}_2F_1 \left(\begin{matrix} -n, -\alpha-n \\ 1+\beta \end{matrix}; \frac{\omega+1}{\omega-1} \right) \dots(4.21)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$P_n^{(\alpha, \beta)}(\omega) = \frac{(1+\alpha+\beta)_{2n}}{n!(1+\alpha+\beta)_n} \left(\frac{\omega+1}{2}\right)^n {}_2F_1 \left(\begin{matrix} -n, -\beta-n \\ -\alpha-\beta-2n \end{matrix}; \frac{2}{\omega+1} \right) \dots(4.22)$$

จากสมการข้างต้นเป็นความสัมพันธ์ของฟังก์ชัน Jacobi โพลิโนเมียล และ ถ้ากำหนดให้ $\beta = \alpha$ แล้วจากสมการ (4.12) จะได้โพลิโนเมียล $P_n^{(\alpha, \alpha)}(\omega)$ ดังนี้

$$P_n^{(\alpha, \alpha)}(\omega) = \frac{(1+\alpha)_n}{n!} {}_2F_1 \left(\begin{matrix} -n, 2\alpha+n+1 \\ 1+\alpha \end{matrix}; \frac{1-\omega}{2} \right) \dots(4.23)$$

จะแทนที่ค่า Hypergeometric ฟังก์ชัน ${}_2F_1$ ในสมการ (4.13) ลงในสมการ (4.23) จะได้

$$P_n^{(\alpha, \alpha)}(\omega) = \frac{(1+\alpha)_n}{n!} \sum_{k=0}^n \frac{(-n)_k (2\alpha+n+1)_k}{k! (1+\alpha)_k} \left(\frac{1-\omega}{2}\right)^k \dots(4.24)$$

จะนำค่าแทนค่าสมการ (4.24) ลงในสมการ (4.10) จะได้

$$U_n^\alpha(\omega) = \sum_{k=0}^n \frac{(-n)_k (2\alpha+n+1)_k}{k! (1+\alpha)_k} \left(\frac{1-\omega}{2}\right)^k \dots(4.25)$$

โดยที่ $(-n)_k = (-1)^k \frac{n!}{(n-k)!} \dots(4.26)$

โดยการแทนค่าสมการ (4.26) ลงในสมการ (4.25) จะได้

$$U_n^\alpha(\omega) = \sum_{k=0}^n (-1)^k C_k^n \frac{(2\alpha+n+1)_k}{(1+\alpha)_k} \left(\frac{1-\omega}{2}\right)^k \dots(4.27)$$

โดยที่ C_k^n เป็นสัมประสิทธิ์ของ Binomial ซึ่งเท่ากับ $\frac{n!}{(n-k)!k!}$

จากสมการ (4.27) ถ้า $n = 1, 2, 3, 4$ และ 5 ค่า $U_n^\alpha(\omega)$ เขียนเป็นตารางได้ดังแสดงในตารางที่ 4.2

ตารางที่ 4.2 แสดงค่าอนุตรวีาสเฟียริคัล โพลีโนเมียล

n	$U_n^\alpha(\omega)$
0	1
1	ω
2	$\frac{(2\alpha+3)\omega^2 - 1}{2\alpha + 2}$
3	$\frac{(2\alpha+5)\omega^3 - 3\omega}{4\alpha^2 + 12\alpha + 8}$
4	$\frac{(4\alpha^2+24\alpha+35)\omega^4 - (12\alpha+30)\omega^2 + 3}{8\alpha^3 + 44\alpha^2 + 76\alpha + 40}$
5	$\frac{(8\alpha^3+84\alpha^2+286\alpha+315)\omega^5 - (40\alpha^2+240\alpha+350)\omega^3 + (30\alpha+75)\omega}{8\alpha^3 + 44\alpha^2 + 76\alpha + 40}$

ถ้ากำหนดให้ $n = 3$ เปลี่ยนค่า α ตั้งแต่ -0.2 ถึง 1 จะได้ค่า $U_3^\alpha(\omega)$ ดังแสดงในตารางที่ 4.3

ตารางที่ 4.3 แสดงค่า $U_3^\alpha(\omega)$ โดยการกำหนดค่า α

α	$U_3^\alpha(\omega)$
-0.2	$2.875\omega^3 - 1.875\omega$
0	$2.5\omega^3 - 1.5\omega$
0.2	$2.25\omega^3 - 1.25\omega$
0.4	$2.0714\omega^3 - 1.0714\omega$
0.6	$1.9375\omega^3 - 0.9375\omega$
0.8	$1.8333\omega^3 - 0.8333\omega$
1	$1.75\omega^3 - 0.75\omega$

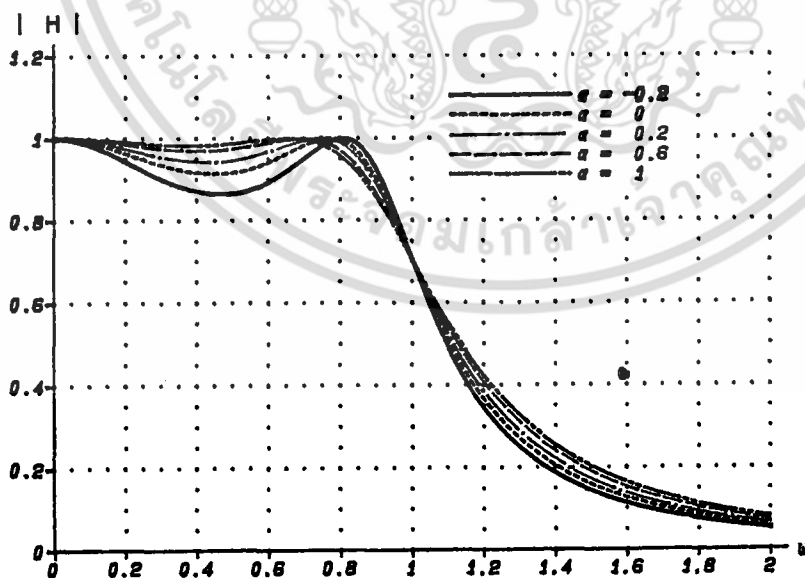
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากค่า $U_3^\alpha(\omega)$ ในตารางที่ 4.3 แทนลงในสมการ (4.9) ได้ทราบสเฟอริกฟังก์ชันของอูลตราสเฟียริคัลโพลีโนเมียล ที่อยู่ทางด้านซ้ายมือของ S-Plane ดังแสดงในตารางที่ 4.4

ตารางที่ 4.4 แสดงค่าอูลตราสเฟียริคัลโพลีโนเมียล

α	โพลีโนเมียล
-0.2	$S^3 + 0.8397S^2 + 1.0047S + 0.3478$
0	$S^3 + 0.9622S^2 + 1.0629S + 0.4$
0.2	$S^3 + 1.0616S^2 + 1.1191S + 0.4444$
0.4	$S^3 + 1.1438S^2 + 1.1712S + 0.4828$
0.6	$S^3 + 1.2125S^2 + 1.2189S + 0.5161$
0.8	$S^3 + 1.2709S^2 + 1.2622S + 0.5455$
1	$S^3 + 1.3212S^2 + 1.3014S + 0.5714$

จากรูปที่ 4.2 แสดงผลตอบสนองทางขนาดของวงจรรองความถี่ต่ำแบบอูลตราสเฟียริคัลโพลีโนเมียล โดยกำหนดให้ $n = 3$ กำหนดค่า α ที่ค่าต่างๆ



รูปที่ 4.2 ผลตอบสนองทางขนาดของวงจรรองความถี่ต่ำแบบอูลตราสเฟียริคัลโพลีโนเมียล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การออกแบบวงจรกรองความถี่ต่ำแบบบัตเตอร์เวท-อูลตราสเฟียรีคัล โพลีโนเมียล

เป็นที่ทราบกันดีอยู่แล้วว่า ในการประมาณผลตอบสนองของวงจรกรองความถี่ต่ำแบบบัตเตอร์เวท โพลีโนเมียลนั้น ให้คุณลักษณะของผลตอบสนองทางขนาดในช่วงความถี่ผ่าน (Pass band) ราบเรียบที่สุด (Maximally Flat Amplitude) ส่วนวงจรกรองความถี่ต่ำแบบอูลตราสเฟียรีคัล โพลีโนเมียล (Ultraspherical Polynomials Filter) ให้ผลตอบสนองทางขนาดในย่านคัทออฟที่ชันกว่า และมีสมรรถนะการลดทอน (Attenuation) สูงกว่าในย่านความถี่สูงมากกว่าแบบบัตเตอร์เวท นอกจากนี้ อูลตราสเฟียรีคัล โพลีโนเมียลยังมีพารามิเตอร์ (α) ซึ่งสามารถกำหนดค่าผลตอบสนองทางขนาดได้ จากข้อดีดังกล่าวข้างต้น ในบทนี้จึงได้เสนอวิธีใหม่ โดยนำเอาบัตเตอร์เวทร่วมกับอูลตราสเฟียรีคัล โพลีโนเมียล หรือเรียกอีกชื่อหนึ่งว่าคอมโพสิทโพลีโนเมียล (Composite Polynomials) ระหว่างบัตเตอร์เวทกับอูลตราสเฟียรีคัล [3] ทำให้สามารถกำหนดค่าของผลตอบสนองทางขนาดตามที่ต้องการได้ พร้อมทั้งพิจารณาเสถียรภาพของวงจร

ในการออกแบบวงจรกรองความถี่ต่ำแบบบัตเตอร์เวท-อูลตราสเฟียรีคัล โพลีโนเมียลนั้น กำหนดทรานส์เฟอร์ฟังก์ชัน เป็น

$$H(s) \cdot H(-s) \Big|_{s=j\omega} = |H_n(j\omega)|^2 = \frac{1}{1 + F_n(\omega^2)} \dots(4.28)$$

โดยที่ $F_n(\omega^2)$ เป็นขนาดกำลังสอง (Magnitude Square) ของคอมโพสิทโพลีโนเมียล (Composite Polynomials) ซึ่งกำหนดได้โดย

$$F_n(\omega^2) = B_k^2(\omega) \cdot U_{n-k}^2(\omega) \dots(4.29)$$

ค่า k ในสมการ (4.29) เป็นเลขจำนวนเต็ม n เป็นอันดับของทรานส์เฟอร์ฟังก์ชัน โดย $0 \leq k \leq n$

สูตรทั่วไปของบัตเตอร์เวท โพลีโนเมียล คือ

$$B_k(\omega) = \omega^k \dots(4.30)$$

และจากสูตรทั่วไปของอูลตราสเฟียรีคัล โพลีโนเมียล ในสมการ (4.27) คือ

$$U_n(\omega) = \sum_{k=0}^n (-1)^k C_k^n \frac{(2\alpha+n+1)_k}{(1+\alpha)_k} \left(\frac{1-\omega}{2}\right)^k \dots(4.31)$$

โดยที่ C_k^n เป็นสัมประสิทธิ์ของ Binomial ซึ่งเท่ากับ $\frac{n!}{(n-k)!k!}$

สมมติให้อันดับของคอมโพลีโนเมียล $n = 3$ และให้ $k = 0, 1$ และ 2 ตามลำดับ แทนลงในสมการ (4.30) และ (4.31) จะได้โพลีโนเมียลของบัตเตอร์เวิร์ท และของอูลตราสเฟียริคัล ตามลำดับ แล้วนำโพลีโนเมียลที่ได้แทนลงในสมการ (4.29) จะได้ค่า $F_3(\omega)$ เขียนเป็นตารางได้ดังแสดงในตารางที่ 4.5

ตารางที่ 4.5 แสดงค่า $F_3(\omega)$ โดยกำหนดค่า α

$k \backslash \alpha$	-0.2	1
0	$2.875\omega^3 - 1.875\omega$	$1.75\omega^3 - 0.75\omega$
1	$1.625\omega^3 - 0.625\omega$	$1.25\omega^3 - 0.25\omega$
2	ω^3	ω^3

โดยการแทนค่า $F_3(\omega)$ ลงในสมการ (4.28) สำหรับราก (Roots) และคอมโพลีโนเมียลของทรานส์เฟอร์ฟังก์ชัน $H_3(s)$ ที่อยู่ทางด้านซ้ายมือของ S-Plane ถูกแสดงไว้ในตารางที่ 4.6 ดังนี้

เพื่อให้ง่ายต่อการนำไปใช้งาน เราจะทำการ Normalize ผลตอบสนองทางขนาด ให้มีค่าเท่ากับหนึ่งที่ $\omega = 0$ และให้ได้จุดคัทออฟ (Cut off) -3 dB ที่ $\omega = 1$ จากโพลีโนเมียลในตารางที่ 4.6 จะทำได้โดยการ Normalize ผลตอบสนองทางขนาด และทางความถี่ ซึ่งมีขั้นตอนดังต่อไปนี้

ตารางที่ 4.6 แสดงค่ารากและโพลีโนเมียลของบัตเตอร์เวิร์ท-อูลตราสเฟียริคัล โพลีโนเมียลที่ยังไม่ได้ Normalize

k	α	ราก	คอมโพลิตโพลีโนเมียล
0	1	-0.660619	$S^3 + 1.3212S^2 + 1.3014S + 0.5714$
$F_3(\omega) = U_3^\alpha$	-0.2	-0.330309 ± j0.869417	$S^3 + 0.8397S^2 + 1.0047S + 0.3478$
		-0.419851	
1	-0.2	-0.209925 ± j0.885653	$S^3 + 1.7133S^2 + 1.6677S + 0.7999$
		-0.856657	
		-0.428328 ± j0.866255	
$F_3(\omega) = B_1 U_2^\alpha$	-0.2	-0.701697	$S^3 + 1.4034S^2 + 1.3694S + 0.6154$
		-0.350848 ± j0.868274	
		-1	
2	1, -0.2	-0.5 ± j0.866025	$S^3 + 2S^2 + 2S + 1$
$F_3(\omega) = B_2 U_1 = B_3$			

การ Normalize ทางขนาด [15]

กำหนดทราเนอส์เฟอร์ฟังก์ชันเป็น

$$H_n(p) = \frac{d_0}{p^n + d_{n-1}p^{n-1} + \dots + d_{n-i}p^{n-i} + \dots + d_1p + d_0} \quad \dots(4.32)$$

โดยที่ d_{n-i} เป็นสัมประสิทธิ์ของ บัตเตอร์เวิร์ท-อูลตราสเฟียริคัล โพลีโนเมียล จากสมการ (4.32) จะได้

$$H_n(p) = \frac{1}{\frac{p^n}{d_0} + \frac{d_{n-1}}{d_0} p^{n-1} + \dots + \frac{d_{n-i}}{d_0} p^{n-i} + \dots + \frac{d_1}{d_0} p + 1} \quad \dots(4.33)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยการแทน $S^n = \frac{p^n}{d_0^n}$

หรือ $S = \frac{p}{d_0^{1/n}} = (d_0)^{-1/n} \cdot p \quad \dots(4.34)$

นำสมการ (4.34) แทนลงในสมการ (4.33) จะได้

$$H_n(S) = \frac{1}{S^n + C_{n-1}S^{n-1} + \dots + C_{n-1}S^{n-1} + \dots + C_1S + 1} \quad \dots(4.35)$$

โดยที่ $C_{n-1} = d_{n-1} \cdot d_0^{-1/n}$
 $C_{n-i} = d_{n-i} \cdot d_0^{-i/n}$
 $C_1 = d_1 \cdot d_0^{-(1-1/n)}$

จากสมการ (4.35) จะได้ผลตอบสนองทางขนาดมีค่าเท่ากับหนึ่งที่ $\omega = 0$

การ Normalize ทางความถี่

กำหนดทรานส์เฟอ์ฟังก์ชัน ที่จะ Normalize ทางความถี่ ดังนี้

เมื่อ n เป็นจำนวนคี่

$$H_n(S) = \frac{G}{\prod_{i=1}^{n/2} [S^2 + (\omega_{oi}/Q_i)S + \omega_{oi}^2]} \quad \dots(4.36)$$

และเมื่อ n เป็นจำนวนคู่

$$H_n(S) = \frac{G}{(S+\omega_{on}) \prod_{i=1}^{(n-1)/2} [S^2 + (\omega_{oi}/Q_i)S + \omega_{oi}^2]} \quad \dots(4.37)$$

โดยที่ n เป็นอันดับ (order) ของทรานส์เฟอ์ฟังก์ชัน

G เป็นค่าคงที่ที่พอดีทำให้ $H(j0) = 1$

ผลตอบสนองทางขนาดของทรานส์เฟอ์ฟังก์ชัน $|H_n(j\omega)|$ เท่ากับ $1/\sqrt{2}$ ที่ความถี่

$\omega = \omega_d$ ถ้าพิจารณาอันดับของทรานส์เฟอ์ฟังก์ชันที่เป็น จำนวนคี่ สมการ (4.36) จะเขียนใหม่ได้เป็น

$$\sqrt{2} G = \prod_{i=1}^{n/2} \left((\omega_{oi}^2 - \omega_d^2)^2 + \left(\frac{\omega_{oi} \omega_d}{Q_i} \right)^2 \right)^{1/2} \quad \dots(4.38)$$

และอันดับของทราจโพล์ฟังก์ชันเป็นจำนวนคี่ จากสมการ (4.37) จะได้

$$\sqrt{2} G = \sqrt{\omega_d^2 + \omega_{on}^2} \prod_{i=1}^{(n-1)/2} \left((\omega_{oi}^2 - \omega_d^2)^2 + \left(\frac{\omega_{oi} \omega_d}{Q_i} \right)^2 \right)^{1/2} \quad \dots(4.39)$$

จากสมการ (4.38) และ (4.39) พารามิเตอร์ G และ Q_i และ ω_{oi} เป็นตัวทราบค่าสิ่งที่ต้องคำนวณหาคือ ω_d การหา ω_d จะใช้คอมพิวเตอร์ช่วย และพิจารณาค่ารากของ ω_d ที่เป็นจริงมีค่าเป็นบวกเท่านั้น

จากโพลีโนเมียลในตารางที่ 4.6 เมื่อทำการ Normalize ผลตอบสนองทางขนาดและทางความถี่แล้วจะได้ค่าโพลีโนเมียลดังแสดงในตารางที่ 4.7

ตารางที่ 4.7 แสดงโพลีโนเมียลของมอดเตอร์เวท-อูลตราสเฟียคัล ที่ Normalize แล้ว

k	α	คอม โพลิต โพลีโนเมียล
0	1	$1.7499S^3 + 2.3122S^2 + 2.2775S + 1$
	-0.2	$2.8746S^3 + 2.4141S^2 + 2.8885S + 1$
1	1	$1.2497S^3 + 2.1414S^2 + 2.0847S + 1$
	-0.2	$1.6251S^3 + 2.2806S^2 + 2.2252S + 1$
2	1, -0.2	$S^3 + 2S^2 + 2S + 1$

โดยทั่วไปเสถียรภาพของวงจรจะขึ้นอยู่กับตำแหน่งค่าราก (Dominant Poles) ของสมการคุณลักษณะ (Characteristic Equation) ถ้าหากกำหนดทราจโพล์ฟังก์ชันของระบบเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$H(S) = \frac{N(S)}{D(S)} = \frac{N_0 + N_1S + N_2S^2 + \dots + N_mS^m}{D_0 + D_1S + D_2S^2 + \dots + D_nS^n} \dots(4.40)$$

เมื่อ $N(S)$ และ $D(S)$ เป็น โพลิโนเมียลของตัวเศษและโพลิโนเมียลของตัวส่วนตามลำดับ ซึ่งเขียนอยู่ในรูปของ S -Plane อันดับของโพลิโนเมียลของตัวเศษ (Numerator) จะต้องน้อยกว่าอันดับของโพลิโนเมียลของตัวส่วน (Denominator)

คุณลักษณะของวงจรที่เสถียรภาพนั้น ค่ารากของตัวส่วนของทรานส์เฟอว์ฟังก์ชัน จะต้องอยู่ในครึ่งซ้ายของ S -Plane เราสามารถหาเสถียรภาพของวงจรได้โดยใช้วิธีของไนควิสต์ไดอะแกรม (Nyquist Diagram) [22] โดยการกำหนด

$$D(S) = Re + j Im \dots(4.41)$$

เมื่อ $D(S)$ เป็น โพลิโนเมียลของตัวส่วน โดยที่ระบบจะเสถียรภาพ เมื่อจุดศูนย์กลาง (Origin) ใน $D(S)$ -Plane ถูกปิดล้อมด้วยเส้นทางของไนควิสต์พล็อต

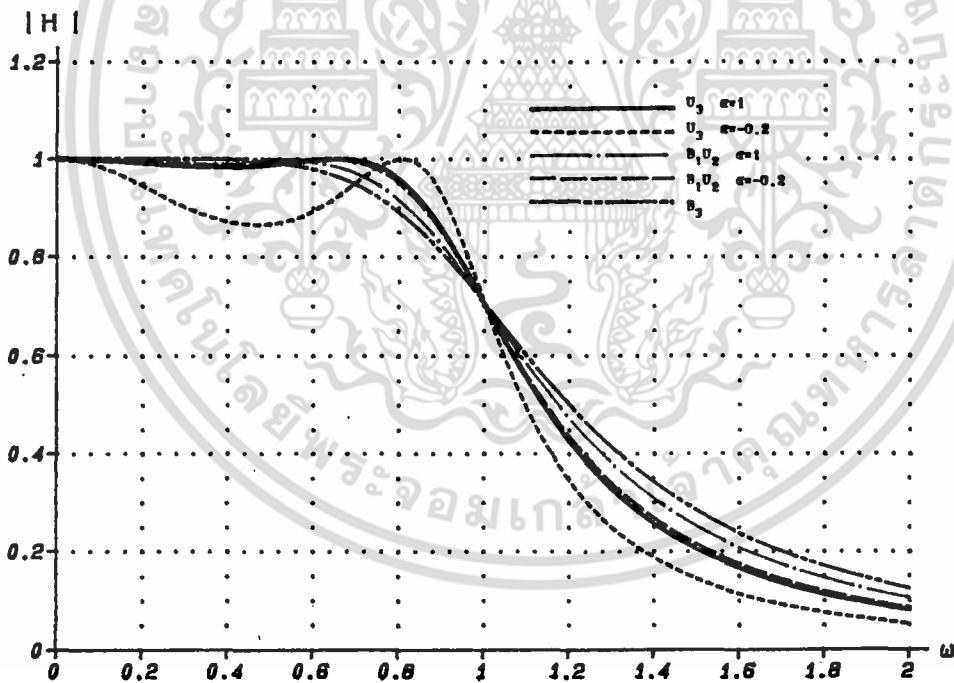
4.4 คุณสมบัติของวงจรกรองความถี่ต่ำแบบบัตเตอร์เวิร์ท-อูลตราสเฟียริคัล โพลิโนเมียล

จากตารางที่ 4.5 $F_3(\omega)$ ที่ $k = 0$ นั้นวงจรกรองความถี่ต่ำที่มีคุณลักษณะทางขนาดเป็นแบบอูลตราสเฟียริคัล โพลิโนเมียล โดยการกำหนดพารามิเตอร์ (α) ของวงจรกรองความถี่ต่ำแบบอูลตราสเฟียริคัล โพลิโนเมียล เมื่อสมมติ $\alpha = -0.2$ และ $\alpha = 1$ จะเห็นได้ว่า α ที่เป็นลบจะให้คุณลักษณะทางขนาดที่ชันในย่านคัทออฟ (Cut off) ในช่วงความถี่ผ่าน (Passband) เกิด Ripple ซึ่งคล้ายวงจรกรองความถี่ต่ำแบบเชบีเชฟ เมื่อค่า α ของวงจรกรองความถี่ต่ำแบบอูลตราสเฟียริคัล โพลิโนเมียลเป็นลบมากขึ้นแต่ไม่น้อยกว่า $\alpha > -1$ ระดับของ Ripple จะมากกว่าวงจรกรองความถี่ต่ำแบบเชบีเชฟ [11] ซึ่งคุณลักษณะทางขนาดของวงจรกรองความถี่ต่ำแบบบัตเตอร์เวิร์ท-อูลตราสเฟียริคัล โพลิโนเมียล ดังแสดงในรูปที่ 4.3 จากรูป 4.3 จะเห็นได้ว่า B_1U_2 วงจรกรองความถี่ต่ำแบบบัตเตอร์เวิร์ท-อูลตราสเฟียริคัล โพลิโนเมียล โดยพารามิเตอร์ $\alpha = 1, -0.2$ ให้คุณลักษณะทางขนาดดีกว่าวงจรกรองความถี่ต่ำแบบบัตเตอร์เวิร์ท B_3

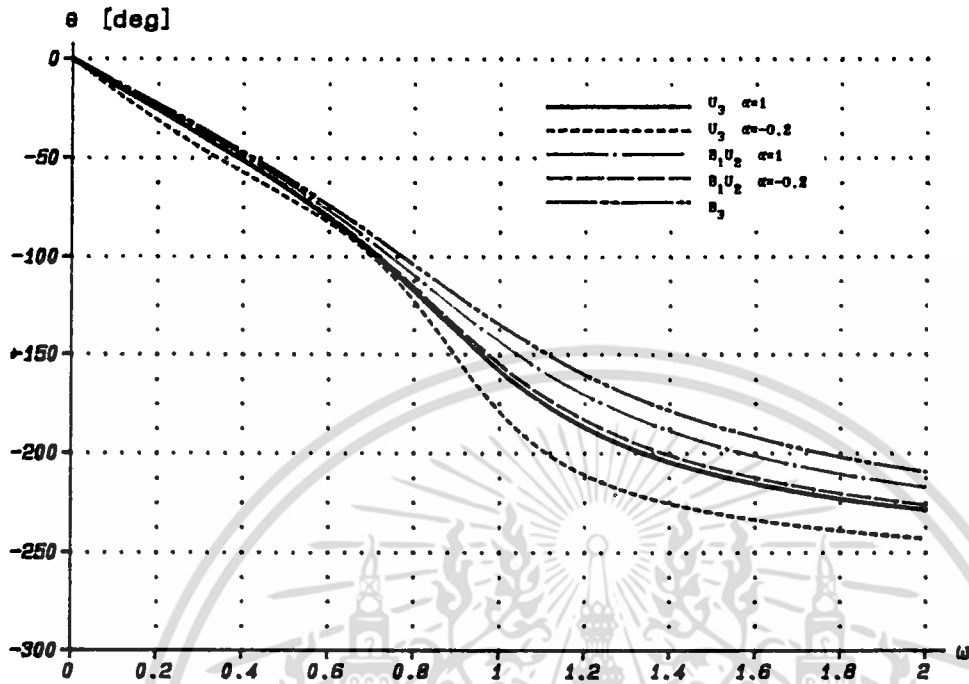
ในรูปที่ 4.4 และรูป 4.5 แสดงคุณลักษณะ ของเฟส (Phase Characteristics) และกรุปดีเลย์ (Delay Characteristics) ซึ่ง B_1U_2 ให้ค่าคุณลักษณะของเฟสและกรุปดีเลย์ดีกว่าวงจรกรองความถี่ต่ำแบบอูลตราสเฟียริคัล โพลิโนเมียล U_3 และในรูปที่ 4.6 แสดงผลตอบสนองอิมพัลส์ (Impulse Response) ซึ่ง B_1U_2 ให้คุณลักษณะผลตอบสนองดีกว่า U_3 ส่วนในรูป 4.7 แสดงผลตอบสนอง Unit-Step ซึ่ง B_1U_2 ให้ Delay และ Rise time น้อยกว่า

U_3 และรูปที่ 4.8 แสดงในควิสต์ไดอะแกรม จากรูปจะเห็นว่าในควิสต์ไดอะแกรมของวงจรรองความถี่ต่ำแบบบัตเตอร์เวท-อูลตราสเฟียริคัลโพลีโนเมียลที่ออกแบบได้จะล้อมรอบจุด Origin ใน $D(S)$ -Plane ทั้งหมด แสดงให้เห็นว่าระบบมีความเสถียรภาพ

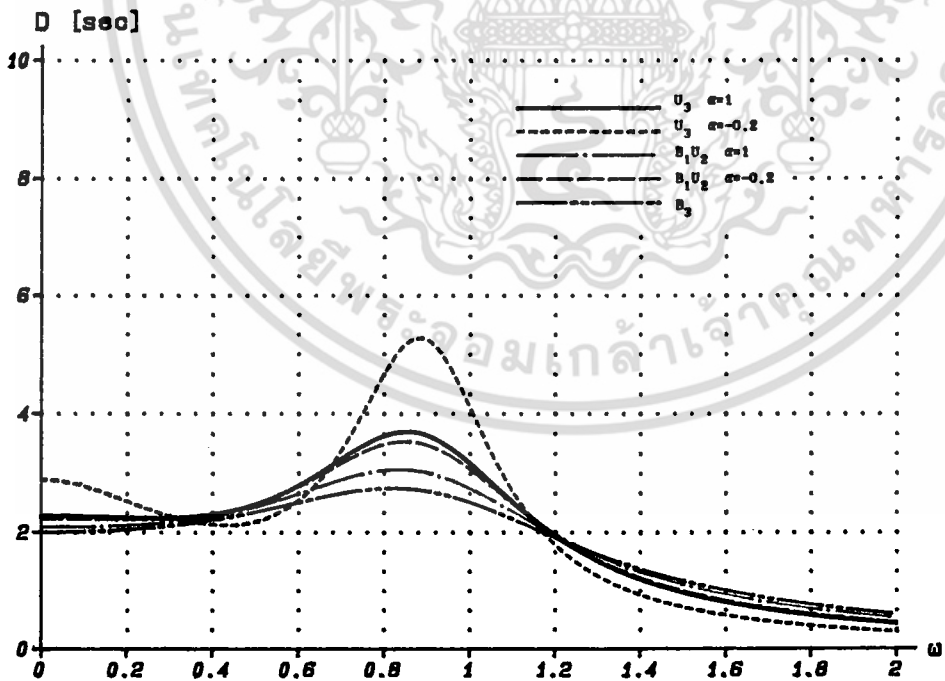
จากคุณลักษณะพิเศษดังที่ได้กล่าวมานี้ แสดงให้เห็นว่าวงจรรองความถี่ต่ำแบบบัตเตอร์เวท-อูลตราสเฟียริคัลโพลีโนเมียล ให้คุณลักษณะผลตอบแทนทางขนาดที่ค้ำทอพอทอมกว่าแบบวงจรรองความถี่ต่ำแบบบัตเตอร์เวท ซึ่งคุณลักษณะทางขนาดกำหนดได้โดยการกำหนดค่าพารามิเตอร์ α ส่วนข้อเสียของวงจรรองความถี่ต่ำดังกล่าวคือเฟสกับกรุปดีเลย์ สิวจรรองความถี่ต่ำแบบบัตเตอร์เวทไม่ได้ แต่ข้อเสียดังกล่าวสามารถแก้ไขได้โดยการใส่เฟสอีควอไลเซอร์ซึ่งเป็นวงจรรองความถี่ All Pass ต่อร่วมกับวงจรรองความถี่ต่ำแบบบัตเตอร์เวท-อูลตราสเฟียริคัลโพลีโนเมียล



รูปที่ 4.3 คุณลักษณะทางขนาดของวงจรรองความถี่ต่ำแบบบัตเตอร์เวท-อูลตราสเฟียริคัล

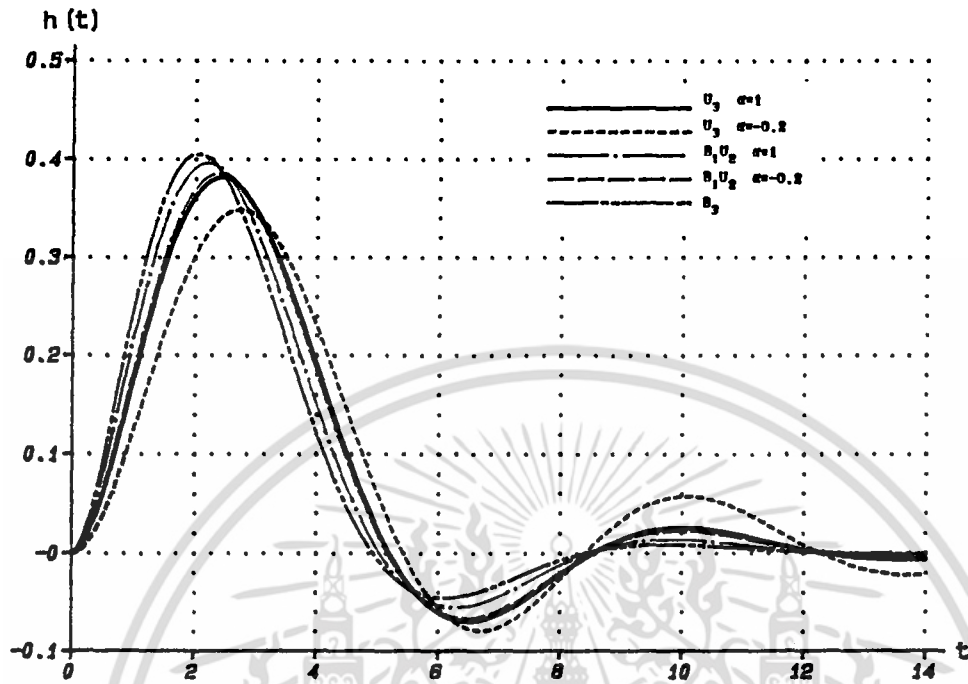


รูปที่ 4.4 คุณลักษณะทางเฟสของวงจรงอบความถี่ต่ำแบบบัตเตอร์เวิร์ท-อูลตราสเฟียรัคัล

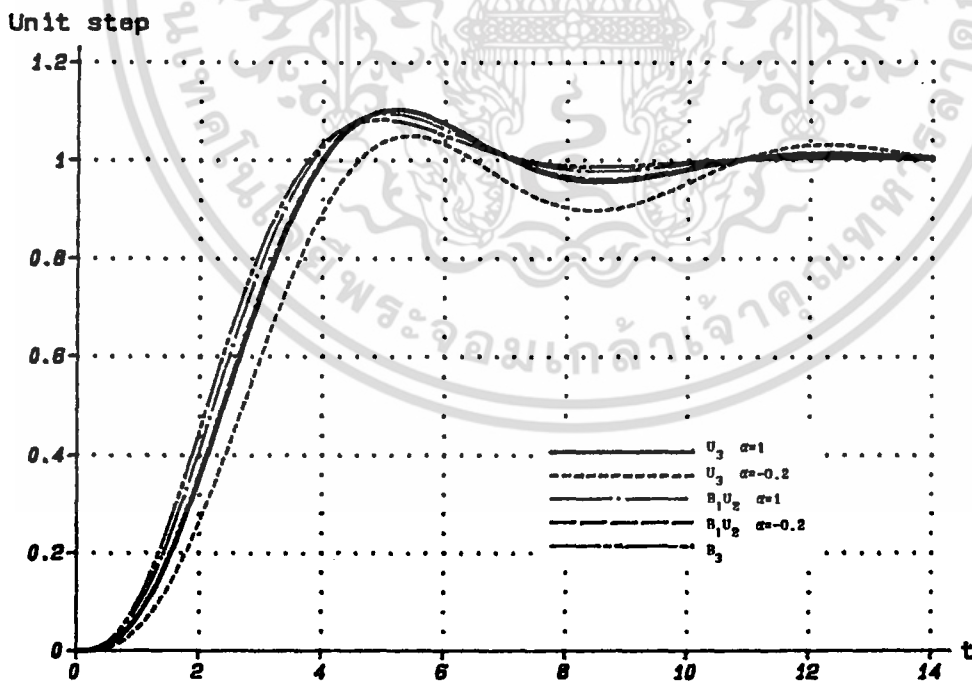


รูปที่ 4.5 คุณลักษณะการหน่วงของวงจรงอบความถี่ต่ำแบบบัตเตอร์เวิร์ท-อูลตราสเฟียรัคัล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

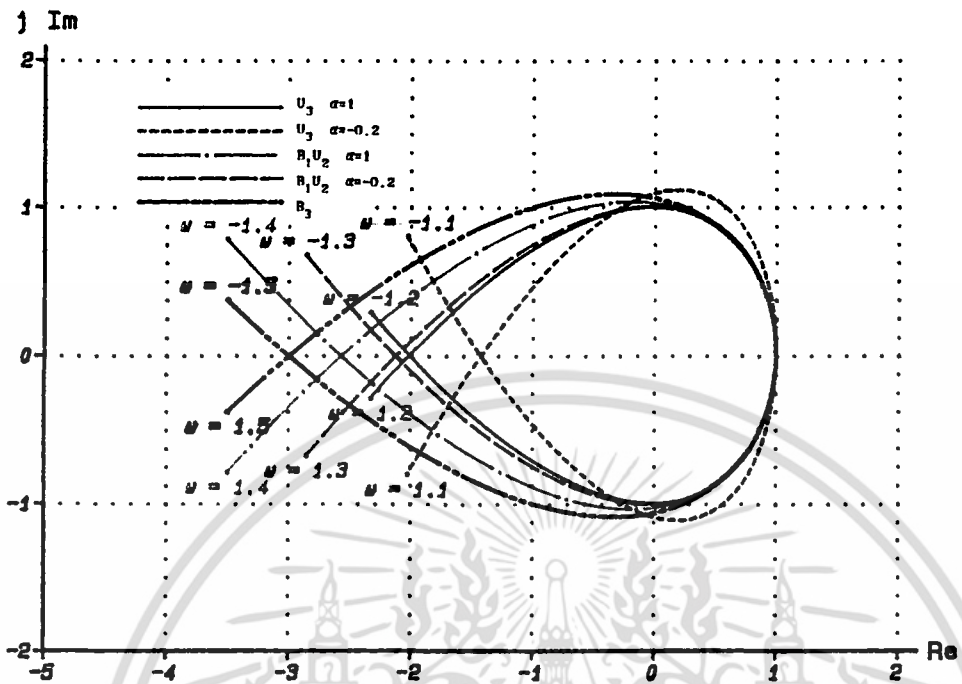


รูปที่ 4.6 ผลตอบสนองอิมพัลส์ของวงจรองความถี่ต่ำแบบบัตรเตอร์เวท-อูลตราสเฟียรัคัล



รูปที่ 4.7 ผลตอบสนอง Unit-Step ของวงจรองความถี่ต่ำแบบบัตรเตอร์เวท-อูลตราสเฟียรัคัล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 ไนควิสต์โคจรแอมพลิจูดของวงจรถองความถี่ต่ำแบบบัตเตอร์เวิร์ท-ฮอลต์วาล์สเฟียรัคัล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การออกแบบวงจรกรองความถี่ต่ำเวฟดิสครีทฟิลเตอร์

ในการออกแบบวงจรกรองความถี่จากทรานส์เฟอร์ฟังก์ชัน โดยทั่วไปทำได้หลายวิธี แต่วิธีนั้นจะมีข้อดีและข้อเสียแตกต่างกัน เช่น การออกแบบวงจรพาสซีฟ (Passive) [17] ต้องใช้อุปกรณ์ (Element) R, L และ C ซึ่งจะทำให้วงจรมีขนาดใหญ่ ส่วนการออกแบบวงจรแอคทีฟ RC (Active RC) [14,17] นั้น สามารถใช้ออปแอมป์ (Op Amp) ร่วมกับพาสซีฟอีลิเมนต์ R กับ C ทำให้วงจรมีขนาดเล็กลง มีน้ำหนักน้อยและสามารถทำเป็นรูปแบบย่อเล็ก ๆ โดยอาศัยเทคโนโลยีของ Integrated Circuit เมื่อทำเป็น Mass-Product แล้วราคาก็ไม่แพง แต่ให้ค่าความไว (Sensitivity) แปรเปลี่ยนในตัวอีลิเมนต์สูง ซึ่งอาจจะทำให้วงจรไม่เสถียรได้ เป็นที่ทราบดีกันอยู่แล้วว่าวงจรพาสซีฟ LC มีค่าความไวแปรเปลี่ยนในตัวอีลิเมนต์ต่ำ ในบทนี้ได้เสนอวิธีการใหม่ในการออกแบบวงจรกรองความถี่ต่ำ โดยอาศัยโครงสร้างของวงจรพาสซีฟ LC เราสามารถแปลงวงจรพาสซีฟ LC ให้เป็นวงจรกรองความถี่แบบดิจิทัล ซึ่งเรียกวางจรแบบนี้ว่าวงจรเวฟดิจิตอลฟิลเตอร์ (Wave Digital Filter) [6] จากวงจรเวฟดิจิตอลฟิลเตอร์สามารถนำมาออกแบบเป็นวงจรเวฟดิสครีทฟิลเตอร์ได้ โดยวงจรดังกล่าวให้ค่าความไวของอีลิเมนต์ต่ำ และสามารถผลิตขึ้นตามกรรมวิธีการผลิตวงจรถวาย LSI ที่มีขนาดเล็กได้ นอกจากนี้วงจรเวฟดิสครีทฟิลเตอร์ที่ออกแบบได้ ยังสามารถให้สมรรถนะในการลดทอนสัญญาณในย่านหยุดสัญญาณที่ดีกว่าการออกแบบวงจรทางอนาล็อกอีกด้วย [1]

5.1 การหาค่าอีลิเมนต์ของวงจรจากทรานส์เฟอร์ฟังก์ชัน

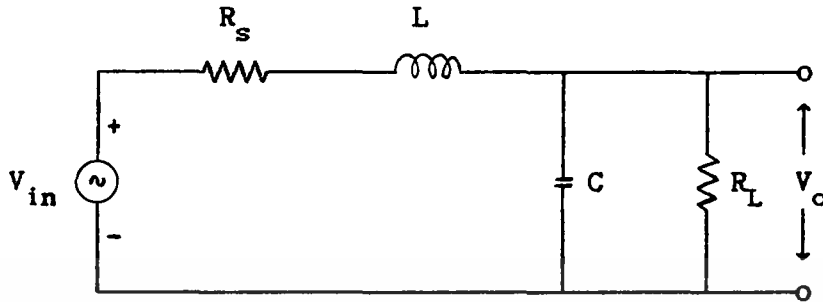
ในบทที่ผ่านมา ได้ทำการออกแบบวงจรกรองความถี่ต่ำแบบบัตเตอร์เวิร์ท-อูลตราสเฟียรัคัลโพลีโนเมียล ซึ่งมีพารามิเตอร์ (α) สามารถกำหนดผลตอบสนองทางขนาดตามที่ต้องการได้จากคุณสมบัตินี้ ในบทนี้ได้เสนอวิธีการนำทรานส์เฟอร์ฟังก์ชันที่ได้ นำมาออกแบบเป็นวงจรกรองความถี่ต่ำด้วยเวฟดิสครีทฟิลเตอร์ 2 แบบคือแบบบัตเตอร์เวิร์ทและแบบบัตเตอร์เวิร์ท-อูลตราสเฟียรัคัลโพลีโนเมียล โดยเริ่มจากการคำนวณหาค่าอีลิเมนต์ของวงจรจากวงจรพาสซีฟ LC หลังจากนั้นจึงแปลงอีลิเมนต์เหล่านี้ให้อยู่ในรูปของอีลิเมนต์ทางดิจิทัลแบบเวฟดิจิตอลฟิลเตอร์

จากรูปที่ 5.1 แสดงวงจรกรองความถี่ต่ำพาสซีฟ LC อันดับ 2 ซึ่งสามารถเขียนเป็นทรานส์เฟอร์ฟังก์ชันได้

$$H(S) = \frac{R_L}{R_S + R_L} \dots (5.1)$$

$$\frac{R_L LC}{R_S + R_L} S^2 + \frac{R_S R_L C + L}{R_S + R_L} S + 1$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

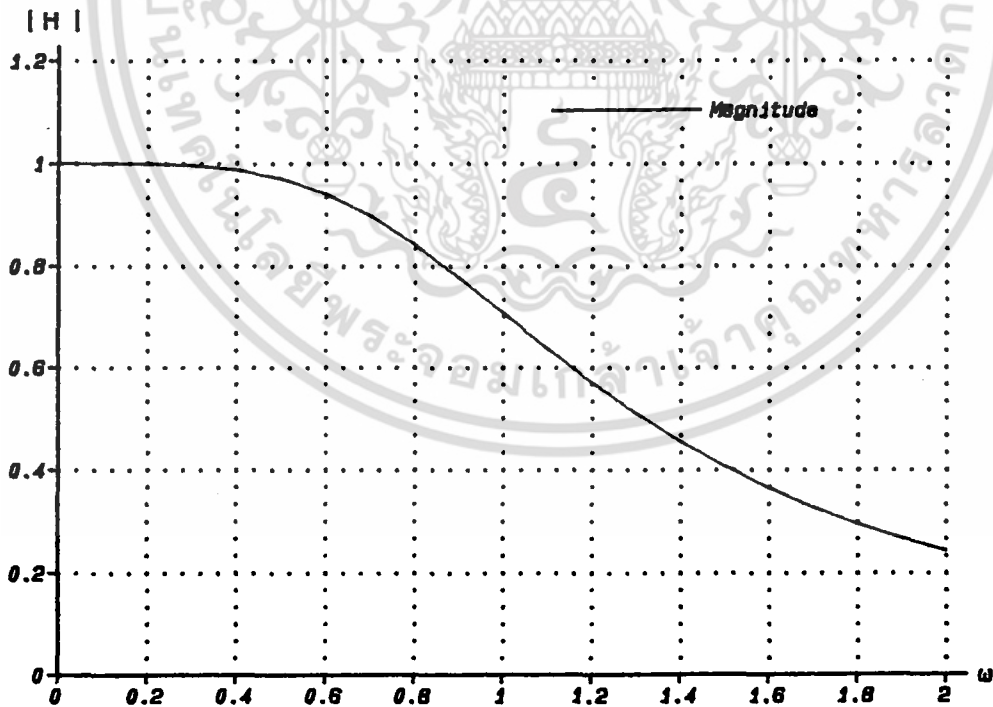


รูปที่ 5.1 วงจรกรองความถี่ต่ำพาสซีฟ LC อันดับ 2

จากทรานส์เฟอร์ฟังก์ชันของบัตเตอร์เวิร์ทในสมการที่ (4.8) ที่ให้คุณลักษณะที่ราบเรียบที่สุดคือ

$$H_2(s) = \frac{1}{s^2 + \sqrt{2}s + 1} \dots(5.2)$$

จากทรานส์เฟอร์ฟังก์ชันในสมการ (5.2) ถ้า Normalize ให้ความถี่คัตออฟที่ $\omega_c = 1$ ผลตอบสนองทางขนาดพล็อตได้ดังแสดงในรูปที่ 5.2



รูปที่ 5.2 ผลตอบสนองทางขนาดจากสมการ (5.2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากทรานส์เฟอร์ฟังก์ชันในสมการ (5.2) สามารถออกแบบทำเป็นวงจรกรองความถี่ต่ำพาสซีฟ LC ดังแสดงในรูปที่ 5.1 ได้ โดยการเทียบสัมประสิทธิ์กับทรานส์เฟอร์ฟังก์ชันในสมการ (5.1) ดังนี้

$$\frac{R_L LC}{R_S + R_L} = 1 \quad \dots(5.3)$$

$$\frac{R_S R_L C + L}{R_S + R_L} = \sqrt{2} \quad \dots(5.4)$$

สำหรับค่าเซ็นซิวิตีตีที่สุด [16] จากสมการ (5.3) และ (5.4) เมื่อกำหนดให้ $R_S = R_L = 1 \Omega$ จะได้ค่า L และ C ตามลำดับ ดังนี้

$$L = \sqrt{2} \text{ H}$$
$$C = \sqrt{2} \text{ F} \quad \dots(5.5)$$

และ

โดยการแทนค่าอุปกรณ์ที่คำนวณได้ในสมการ (5.5) ลงในวงจรในรูปที่ 5.1 จะเป็นวงจรกรองความถี่ต่ำบัตเตอร์เวส มีความถี่คัทออฟ (cut off) ที่ -3 dB เป็น $\omega_{-3dB} = 1$ ซึ่งพล็อตได้ดังแสดงในรูปที่ 5.2

จากวงจรกรองความถี่ต่ำที่ให้คุณลักษณะของผลตอบสนองทางขนาดดังแสดงในรูปที่ 5.2 และจากรูปที่ 5.1 สามารถทำเป็นวงจรกรองความถี่ต่ำแบบดิจิทัลได้ โดยการ Mapping Pole จาก S-Plane ไปเป็น Z-Plane ด้วยวิธีการแปลงไบลิเนียร์ทรานส์ฟอร์ม (Bilinear Transformation) ที่กล่าวไว้ในบทที่ 2 ผลของทรานส์เฟอร์ฟังก์ชันแบบดิจิทัลเพื่อให้ได้ความถี่คัทออฟที่ -3 dB คือ

$$\frac{\omega_{-3dB}}{\omega_{ck}} = \frac{1}{2\pi} = \frac{1}{6.28} \quad \dots(5.6)$$

โดยที่ ω_{ck} เป็นความถี่ของสัญญาณ Sampling ที่ Normalize ไปที่ 2π เพราะฉะนั้นคาบเวลา(Period)ของสัญญาณ Sampling ที่ต้องใช้คือ $T=1$ จากทฤษฎีของเวฟดิจิทัลสโพลเดอร์ในบทที่ 3 ใช้ไบลิเนียร์ทรานส์ฟอร์ม เพื่อแปลงจาก S-Plane ไปเป็น Z-Plane ดังนี้

$$S = \frac{2(Z-1)}{T(Z+1)} \quad \dots(5.7)$$

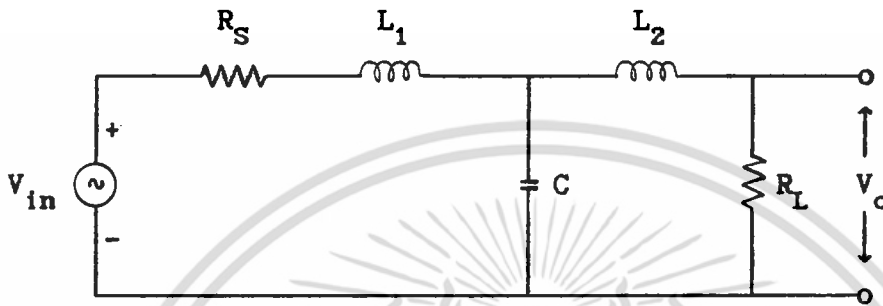
ถ้าหากใช้ไบลิเนียร์ทรานส์ฟอร์มในสมการ (5.7) แล้ว หมายความว่าคาบเวลาของสัญญาณ Sampling ถูก Normalize ให้ $T = 2$ จะทำให้อัตราส่วนทางความถี่ในสมการ(5.6) ของวงจรกรองความถี่ LC ที่ความถี่คัทออฟถูกสเกลไป 0.5 เพราะฉะนั้นจะต้องสเกลทางความถี่ของอุปกรณ์ในสมการ (5.5) ไป 2 เท่า จะได้

$$L = 2\sqrt{2} \text{ H}$$

$$C = 2\sqrt{2} \text{ F}$$

และ

$$R_S = R_L = 1 \ \Omega \quad \dots(5.8)$$



รูปที่ 5.3 วงจรของความถี่ต่ำพาสซีฟ LC อันดับ 3

จากรูปที่ 5.3 แสดงวงจรของความถี่ต่ำพาสซีฟ LC อันดับ 3 ซึ่งสามารถเขียนเป็นทรานส์เฟอร์ฟังก์ชันได้

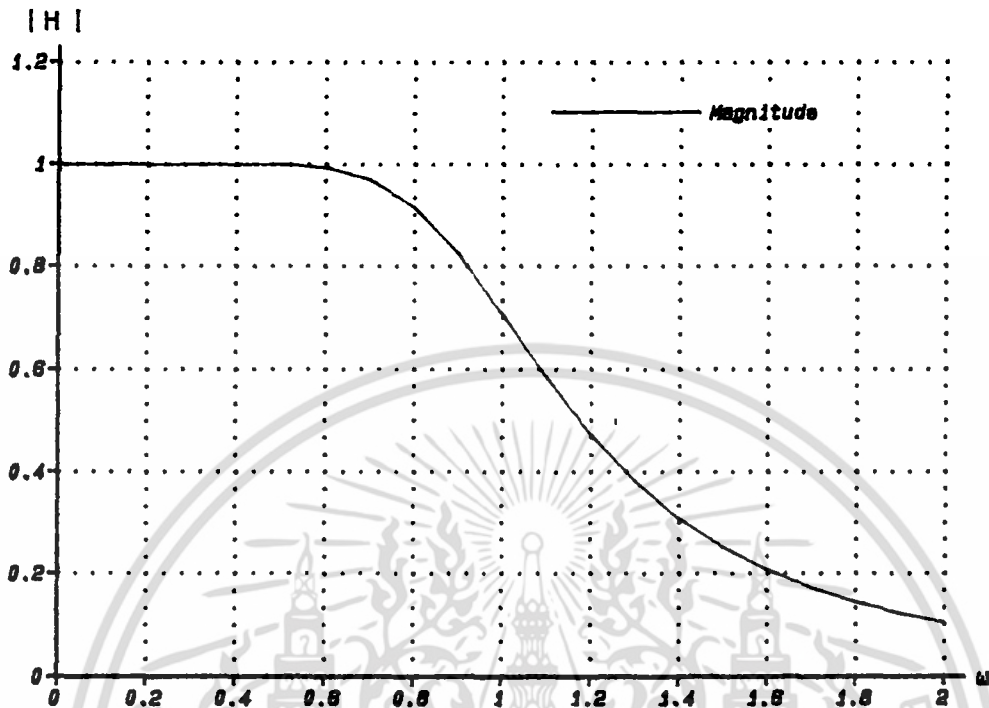
$$H(S) = \frac{R_L}{R_S + R_L} \dots(5.9)$$

$$\frac{L_1 L_2 C}{R_S + R_L} S^3 + \frac{(R_S L_2 C + R_L L_1 C)}{R_S + R_L} S^2 + \frac{(R_S R_L C + L_1 + L_2)}{R_S + R_L} S + 1$$

จากทรานส์เฟอร์ฟังก์ชันของบัตเตอร์เวิร์ท-ออลด์รอสเฟียริคัลโพลีโนเมียล ตารางที่ 4.7 ในบทที่ 4 ได้เลือกค่า $k = 1$ และ $\alpha = 1$ แทนค่าลงในสมการ (4.26) จะได้ทรานส์เฟอร์ฟังก์ชันของบัตเตอร์เวิร์ท-ออลด์รอสเฟียริคัลโพลีโนเมียล ดังนี้

$$H_3(S) = \frac{1}{1.2497S^3 + 2.1414S^2 + 2.0847S + 1} \dots(5.10)$$

จากทรานส์เฟอร์ฟังก์ชันในสมการ (5.10) ถ้า Normalize ให้ความถี่คัทออฟที่ $\omega_c = 1$ ผลตอบสนองทางขนาด พล็อตได้ดังแสดงในรูปที่ 5.4



รูปที่ 5.4 ผลตอบสนองทางขนาดจากสมการ (5.10)

จากทราณส์เฟอ์ฟังก์ชันในสมการ (5.10) สามารถออกแบบทำเป็นวงจรกรองความถี่ต่ำพาสซีฟ LC ดังแสดงในรูปที่ 5.3 ได้ โดยการเทียบสัมประสิทธิ์กับทราณส์เฟอ์ฟังก์ชันในสมการ (5.9) ดังนี้

$$\frac{L_1 L_2 C}{R_S + R_L} = 1.2497 \quad \dots (5.11)$$

$$\frac{R_S L_2 C + R_L L_1 C}{R_S + R_L} = 2.1414 \quad \dots (5.12)$$

$$\frac{R_S R_L C + L_1 + L_2}{R_S + R_L} = 2.0847 \quad \dots (5.13)$$

สำหรับค่าเซ็นซิวิตีตี้สูงสุด [16] จากสมการ (5.11), (5.12) และ (5.13) เมื่อกำหนดให้ $R_S = R_L = 1 \Omega$ จะได้ค่า L_1 , L_2 และ C ตามลำดับ ดังนี้

$$L_1 = 1.1929 \text{ H}$$

$$L_2 = 1.1425 \text{ H}$$

และ $C = 1.8338 \text{ F} \quad \dots (5.14)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยการแทนค่าอุปกรณ์ที่คำนวณได้ในสมการ (5.14) ลงในวงจรในรูปที่ 5.3 จะเป็นวงจรกรองความถี่ต่ำตัดเดือวี-ออลทราสเฟียริคัล โพลีโนเมียล มีความถี่คัทออฟ(cut off) ที่ -3 dB เป็น $\omega_{-3dB} = 1$ ซึ่งพล็อตได้ดังแสดงในรูปที่ 5.4

ในการทำงานเช่นเดียวกันกับการหาค่าอิมพีแดนซ์ทางคิวิตอลของวงจรกรองความถี่ต่ำแบบตัดเดือวี-ออลทราสเฟียริคัล โพลีโนเมียล จะต้องสเกลทงความถี่ของอุปกรณ์ในสมการ (5.14) ไป 2 เท่า จะได้

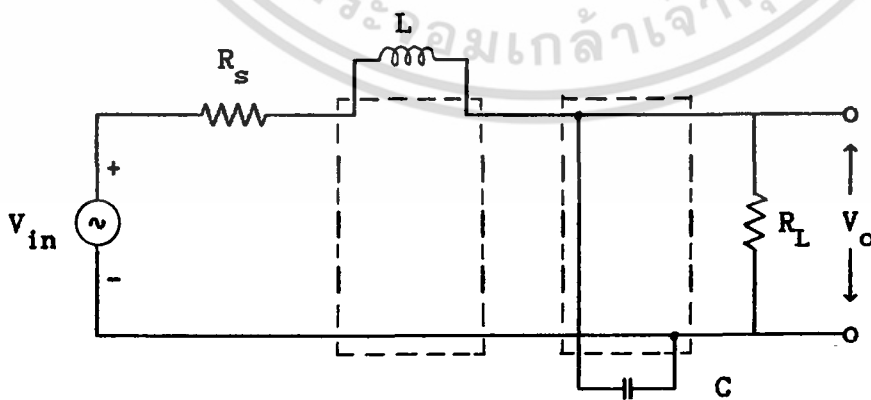
$$L_1 = 2.3858 \text{ H}, \quad L_2 = 2.2850 \text{ H}$$

$$C = 3.6676 \text{ F}$$

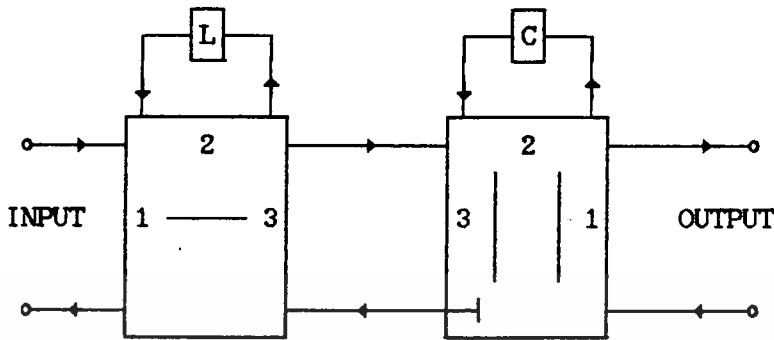
และ $R_S = R_L = 1 \Omega \quad \dots(5.15)$

5.2 การออกแบบวงจรกรองความถี่ต่ำ เพลดสคริปทิลเตอร์แบบตัดเดือวีและแบบตัดเดือวี-ออลทราสเฟียริคัล โพลีโนเมียล

วงจรเพลดสคริปทิลเตอร์ สร้างขึ้นจากวงจรต้นแบบที่เป็นวงจรพาสซีฟ LC และจากโครงสร้างดังกล่าว โดยการแทนอิมพีแดนซ์ต่าง ๆ ทางอนาล็อกด้วยคุณลักษณะของเวฟเน็ตเวิร์ก ผลที่ได้ของวงจรเพลดสคริปทิลเตอร์ จะมีส่วนประกอบหลัก คือ วงจรอะแดปเตอร์ขนาน (Parallel Adaptor) และวงจรอะแดปเตอร์อนุกรม (Series Adaptor) โดยที่วงจรอะแดปเตอร์นั้นจะใช้ขั้วต่อเชื่อมต่อกับ R ซึ่งทำหน้าที่เป็นวงจรรวมและวงจรคูล ส่วนอิมพีแดนซ์แต่ละตัวคือ R, L และ C จะแปลงได้เป็น Digital Sink ดีเลย์ยูนิตคาสเคดกับอินเวอร์เตอร์ และเป็นดีเลย์ยูนิต ตามลำดับดังที่กล่าวมาแล้ว โดยใช้วงจรแซมเปิลและโฮลด์ (Sample-and-Hold) ให้ทำหน้าที่เป็นดีเลย์ยูนิต ซึ่งในหัวข้อนี้ได้เสนอการออกแบบวงจรกรองความถี่ต่ำ เพลดสคริปทิลเตอร์แบบตัดเดือวีอันดับ 2 และวงจรกรองความถี่ต่ำ เพลดสคริปทิลเตอร์แบบตัดเดือวี-ออลทราสเฟียริคัล โพลีโนเมียลอันดับ 3



รูปที่ 5.5 การเชื่อมต่อภายในจากวงจรรูปที่ 5.1

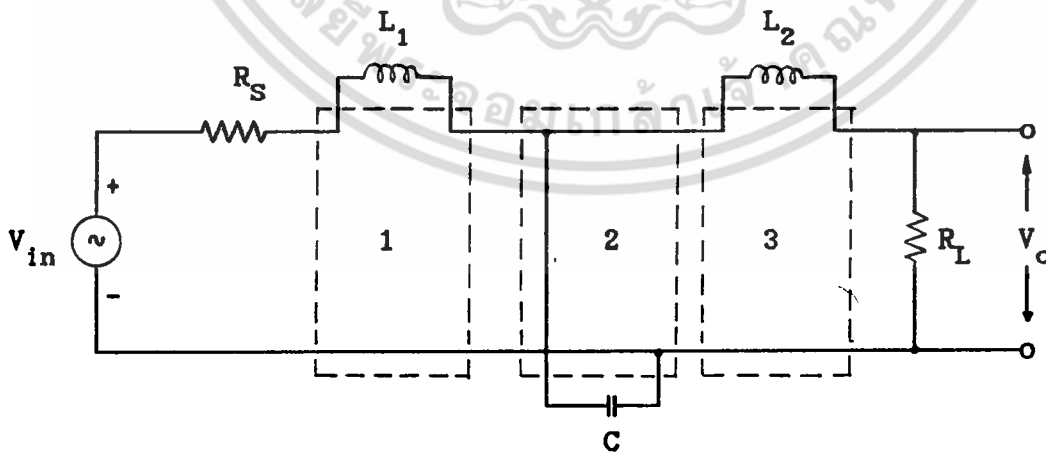


รูปที่ 5.6 วงจรเวทิจิตตอลฟิลเตอร์จากรูปที่ 5.5

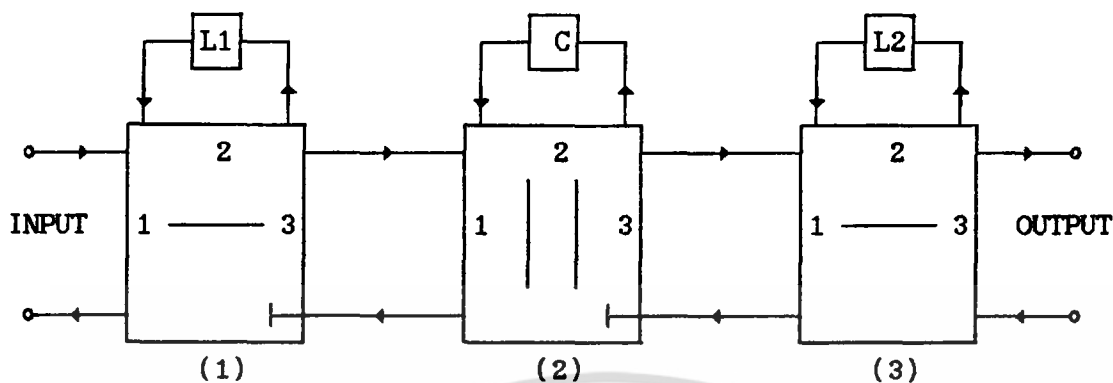
จากวงจรของความถี่ต่ำพาสซีฟ LC อันดับ 2 ในรูปที่ 5.1 และค่าอุปกรณ์ที่คำนวณได้ดังในสมการ (5.8) สามารถนำมาออกแบบเป็นวงจรเวทิจิตตอลฟิลเตอร์ได้ [1,6] ดังแสดงในรูปที่ 5.5 และ รูปที่ 5.6 ตามลำดับ

จากรูปที่ 5.5 แสดงถึงการเชื่อมต่อภายในของวงจรรูปที่ 5.1 จะเห็นได้ว่าตรงส่วนของอินดักเตอร์(L) จะถูกขยายออกไป บนกรอบเส้นประนั้นจะกลายเป็น 3 พอร์ทอะแคปเตเตอร์แบบอนุกรม ในทำนองเดียวกันตรงส่วนของคาปาซิเตอร์ (C) จะเป็น 3 พอร์ทอะแคปเตเตอร์แบบขนาน จากกรณีนี้เราสามารถเขียนเป็นวงจรเวทิจิตตอลฟิลเตอร์ที่สมบูรณ์ได้ ดังแสดงในรูปที่ 5.6

ในทำนองเดียวกันจากวงจรของความถี่ต่ำพาสซีฟ LC อันดับ 3 ในรูปที่ 5.3 และค่าอุปกรณ์ที่คำนวณได้ ดังในสมการ (5.15) สามารถนำมาออกแบบเป็นวงจรเวทิจิตตอลฟิลเตอร์ได้ [1,6] ดังแสดงในรูปที่ 5.7 และ รูปที่ 5.8 ตามลำดับ



รูปที่ 5.7 การเชื่อมต่อภายในจากวงจรรูปที่ 5.3



รูปที่ 5.8 วงจรเวฟดิคตอลฟิลเตอร์จากรูปที่ 5.7

จากรูปที่ 5.7 แสดงถึงการเชื่อมต่อภายในของวงจรรูปที่ 5.3 จะเห็นได้ว่าตรงส่วนของอินดักเตอร์ (L_1) จะถูกขยายออกไป บนกรอบเส้นประนั้นจะกลายเป็น 3 พอร์ตอะแดปเตอร์แบบอนุกรม ในทำนองเดียวกันตรงส่วนของคาปาซิเตอร์ (C) จะเป็น 3 พอร์ตอะแดปเตอร์แบบขนาน และอินดักเตอร์ (L_2) เป็น 3 พอร์ตอะแดปเตอร์แบบอนุกรมเช่นเดียวกับอินดักเตอร์ (L_1) จากกรณีนี้เราสามารถเขียนเป็นวงจรเวฟดิคตอลฟิลเตอร์ที่สมบูรณ์ได้ ดังแสดงในรูปที่ 5.8

จากรูปที่ 5.6 อะแดปเตอร์แบบขนาน ถ้ากำหนดพอร์ท 3 เป็นพอร์ท Reflection Free และเลือกพอร์ท 2 เป็น Dependent Port ดังแสดงในรูปที่ 5.9 จากสมการ (3.28) จะได้ $\alpha_3 = 1$ และ $G_3 = G_1 + G_2$ และจากสมการ (3.29) - (3.34) จะได้

$$B_0 = -\alpha_1 (A_2 - A_1) \quad \dots(5.16)$$

$$B_1 = B_2 + (A_2 - A_1) \quad \dots(5.17)$$

$$B_2 = B_0 + A_3 \quad \dots(5.18)$$

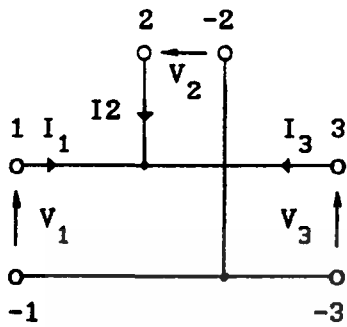
$$B_3 = B_0 + A_2 \quad \dots(5.19)$$

$$\alpha_1 = \frac{G_1}{G_3} \quad \dots(5.20)$$

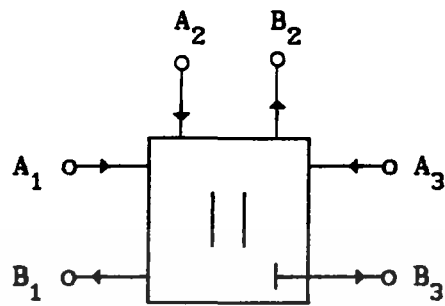
$$\alpha_2 = \frac{G_2}{G_3} \quad \dots(5.21)$$

และ $\alpha_1 + \alpha_2 = 1 \quad \dots(5.22)$

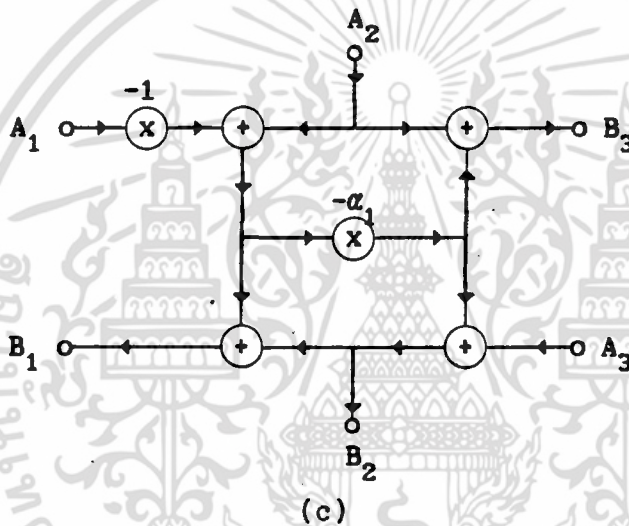
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a)



(b)



(c)

รูปที่ 5.9 3 พอร์ตทอเนตปเตอร์ขนาน กำหนดพอร์ต 3 เป็นพอร์ต Reflection Free และพอร์ต 2 เป็น Dependent Port
 (a) การต่อ 3 พอร์ตแบบขนาน (b) สัญลักษณ์
 (c) ไดอะแกรมทางเดินของสัญญาณ

จากรูปที่ 5.9(a) แสดงถึงการต่อ 3 พอร์ตแบบขนาน รูปที่ 5.9(b) แสดงสัญลักษณ์ของ 3 พอร์ตทอเนตปเตอร์ขนาน โดยกำหนดพอร์ต 3 เป็นพอร์ต Reflection Free และเลือกพอร์ต 2 เป็น Dependent Port ส่วนรูปที่ 5.9(c) แสดงไดอะแกรมทางเดินของสัญญาณ จะเห็นได้ว่าต้องการตัวคูณสัญญาณเพียง 1 ตัว และตัวรวมสัญญาณ 4 ตัว

จากสมการ (5.16) - (5.20) สามารถเขียนใหม่ได้เป็น

$$B_1 = \alpha_1(A_1 - A_2) - A_1 + A_2 + A_3 \quad \dots(5.23)$$

$$B_2 = \alpha_1(A_1 - A_2) + A_3 \quad \dots(5.24)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$B_3 = \alpha_1(A_1 - A_2) + A_2 \quad \dots(5.25)$$

โดยที่

$$\alpha_1 = \frac{G_1}{G_1 + G_2} \quad \dots(5.26)$$

G_1 และ G_2 เป็นค่าความนำ (Admittances) ของพอร์ท 1 และพอร์ท 2 ของอะแดปเตอร์ขนาน ตามลำดับ สามารถคำนวณหาค่าได้ดังนี้

$$G_1 = 1/R_L = 1$$

$$G_2 = 1/(1/C) = 2\sqrt{2}$$

เมื่อแทนค่า G_1 และ G_2 ลงในสมการ (5.26) จะได้

$$\alpha_1 = \frac{1}{1+2\sqrt{2}} = 0.2612 \quad \dots(5.27)$$

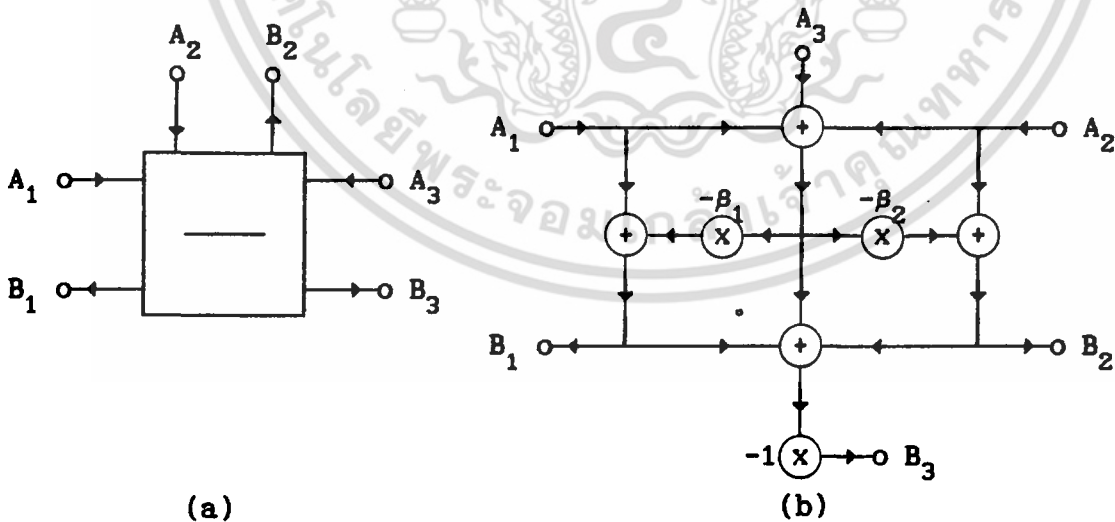
นำค่า α_1 ในสมการ (5.27) แทนลงในสมการ (5.23), (5.24) และ (5.25) จะได้

$$B_1 = -0.7388A_1 + 0.7388A_2 + A_3 \quad \dots(5.28)$$

$$B_2 = 0.2612A_1 - 0.2612A_2 + A_3 \quad \dots(5.29)$$

$$B_3 = 0.2612A_1 + 0.7388A_2 \quad \dots(5.30)$$

จากรูปที่ 5.6 อะแดปเตอร์แบบอนุกรม ถ้าเลือกพอร์ท 3 เป็น Dependent Port ดังแสดงในรูป 5.10 จากสมการ (3.38) - (3.42) จะได้



รูปที่ 5.10 3 พอร์ทอะแดปเตอร์อนุกรม กำหนดพอร์ท 3 เป็น Dependent Port
(a) สัญลักษณ์ (b) ไดอะแกรมทางเดินของสัญญาณ

$$B_1 = A_1 - \beta_1(A_1 + A_2 + A_3) \quad \dots(5.31)$$

$$B_2 = A_2 - \beta_2(A_1 + A_2 + A_3) \quad \dots(5.32)$$

$$B_3 = A_3 + (\beta_1 + \beta_2 - 2)(A_1 + A_2 + A_3) \quad \dots(5.33)$$

โดยที่

$$\beta_1 = \frac{2R_1}{R_1 + R_2 + R_3} \quad \dots(5.34)$$

$$\beta_2 = \frac{2R_2}{R_1 + R_2 + R_3} \quad \dots(5.35)$$

R_1 , R_2 และ R_3 เป็นความต้านทานของพอร์ท 1, 2 และ 3 ของอะแดปเตอร์
อนุกรม ตามลำดับ สามารถคำนวณหาค่าได้ดังนี้

$$R_1 = R_S = 1$$

$$R_2 = L = 2\sqrt{2}$$

$$R_3 = 1/G_3 = 1/(1+2\sqrt{2}) = 0.2612$$

เมื่อแทนค่า R_1 , R_2 และ R_3 ลงในสมการ (5.34) และ (5.35) สามารถหา
 β_1 และ β_2 ได้

$$\beta_1 = \frac{2}{1+2\sqrt{2}+0.2612} = 0.4890 \quad \dots(5.36)$$

$$\beta_2 = \frac{2(2\sqrt{2})}{1+2\sqrt{2}+0.2612} = 1.3832 \quad \dots(5.37)$$

นำแทน β_1 และ β_2 ในสมการ (5.36) และ (5.37) แทนลงในสมการ (5.31) - (5.33)
จะได้

$$B_1 = 0.5110A_1 - 0.4890A_2 - 0.4890A_3 \quad \dots(5.38)$$

$$B_2 = -1.3832A_1 - 0.3832A_2 - 1.3832A_3 \quad \dots(5.39)$$

$$B_3 = -0.1278A_1 - 0.1278A_2 + 0.8722A_3 \quad \dots(5.40)$$

จากรูปที่ 5.8 อะแดปเตอร์แบบอนุกรม (1) ถ้าหากกำหนดพอร์ท 3 เป็นพอร์ท
Reflection Free และเลือกพอร์ท 2 เป็น Dependent Port ดังแสดงในรูปที่ 5.11 จากสม
การ (3.43) จะได้ $\beta_3 = 1$ และ $R_{13} = R_{11} + R_{12}$ และจากสมการ (3.44) - (3.49)
จะได้

$$A_0 = A_3 - B_3 \quad \dots(5.41)$$

$$B_1 = A_1 - \beta_1 A_0 \quad \dots(5.42)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

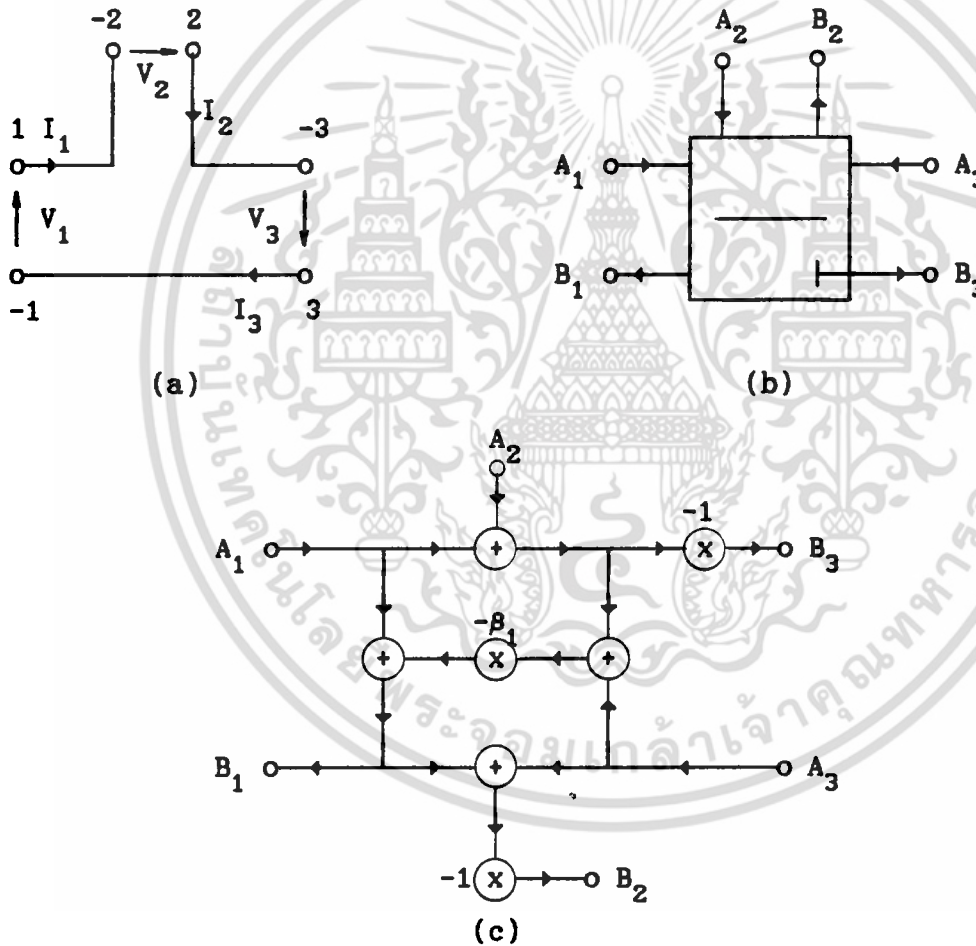
$$B_2 = - (B_1 + A_3) \quad \dots(5.43)$$

$$B_3 = - (A_1 + A_2) \quad \dots(5.44)$$

$$\beta_1 = \frac{R_1}{R_3} \quad \dots(5.45)$$

$$\beta_2 = \frac{R_2}{R_3} \quad \dots(5.46)$$

และ $\beta_1 + \beta_2 = 1 \quad \dots(5.47)$



รูปที่ 5.11 3 พอร์ตทอแคบเตอร์อนุกรม กำหนดพอร์ต 3 เป็นพอร์ต Reflection Free และพอร์ต 2 เป็น Dependent Port

(a) การต่อ 3 พอร์ตแบบอนุกรม (b) สัญลักษณ์

(c) ไดอะแกรมทางเดินของสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 5.11(a) แสดงถึงการต่อ 3 พอร์ตแบบอนุกรม รูปที่ 5.11(b) แสดงสัญลักษณ์ของ 3 พอร์ตอะเนปเดอ์อนุกรม โดยกำหนดพอร์ท 3 เป็นพอร์ท Reflection Free และเลือกพอร์ท 2 เป็น Dependent Port ส่วนรูปที่ 5.11(c) แสดงไดอะแกรมทางเดินของสัญญาณ จะเห็นได้ว่าต้องการตัวคูณสัญญาณเพียง 1 ตัว และตัวรวมสัญญาณ 4 ตัว

จากสมการ (5.41) - (5.45) สามารถเขียนใหม่ได้เป็น

$$B_1 = A_1 - \beta_1(A_1 + A_2 + A_3) \quad \dots(5.48)$$

$$B_2 = \beta_1(A_1 + A_2 + A_3) - A_1 - A_3 \quad \dots(5.49)$$

$$B_3 = -A_1 - A_2 \quad \dots(5.50)$$

โดยที่

$$\beta_1 = \frac{R_{11}}{R_{11} + R_{12}} \quad \dots(5.51)$$

R_{11} และ R_{12} เป็นค่าความต้านทานของพอร์ท 1 และพอร์ท 2 ของอะเนปเดอ์อนุกรม(1) ตามลำดับ สามารถคำนวณหาค่าได้ดังนี้

$$R_{11} = R_S = 1$$

$$R_{12} = L_1 = 2.3858$$

$$R_{13} = R_{11} + R_{12} = 3.3858$$

เมื่อแทนค่า R_{11} และ R_{12} ลงในสมการ (5.51) จะได้

$$\beta_1 = \frac{1}{3.3858} = 0.2953 \quad \dots(5.52)$$

นำค่า β_1 ในสมการ (5.52) แทนลงในสมการ (5.48), (5.49) และ (5.50) จะได้

$$B_1 = 0.7047A_1 - 0.2953A_2 - 0.2953A_3 \quad \dots(5.53)$$

$$B_2 = -0.7047A_1 + 0.2953A_2 - 0.7047A_3 \quad \dots(5.54)$$

$$B_3 = -A_1 - A_2 \quad \dots(5.55)$$

จากรูปที่ 5.8 อะเนปเดอ์แบบขนาน(2) ถ้ากำหนดพอร์ท 3 เป็นพอร์ท Reflection Free และเลือกพอร์ท 2 เป็น Dependent Port ดังแสดงในรูปที่ 5.9 จากสมการ (3.28) จะได้ $\alpha_3 = 1$ และ $G_{23} = G_{21} + G_{22}$ จากสมการ (5.26) สามารถเขียนใหม่ได้เป็น

$$\alpha_1 = \frac{G_{21}}{G_{21} + G_{22}} \quad \dots(5.56)$$

โดยที่ G_{21} และ G_{22} เป็นค่าความนำ (Admittances) ของพอร์ท 1 และพอร์ท 2 ของอะแดปเตอร์ขนาน(2) ตามลำดับ สามารถคำนวณหาค่าได้ดังนี้

$$G_{21} = 1/R_{13} = 1/3.3858 = 0.2953$$

$$G_{22} = 1/(1/C) = C = 3.6676$$

$$G_{23} = G_{21} + G_{22} = 0.2953 + 3.6676 = 3.9629$$

เมื่อแทนค่า G_{21} และ G_{22} ลงในสมการ (5.56) จะได้

$$\alpha_1 = \frac{0.2953}{3.9629} = 0.0745 \quad \dots(5.57)$$

นำค่า α_1 ในสมการ (5.57) แทนลงในสมการ (5.23), (5.24) และ (5.25) จะได้

$$B_1 = -0.9255A_1 + 0.9255A_2 + A_3 \quad \dots(5.58)$$

$$B_2 = 0.0745A_1 - 0.0745A_2 + A_3 \quad \dots(5.59)$$

$$B_3 = 0.0745A_1 + 0.9255A_2 \quad \dots(5.60)$$

จากรูปที่ 5.8 อะแดปเตอร์แบบอนุกรม (3) ถ้าเลือกพอร์ท 3 เป็น Dependent Port ดังแสดงในรูป 5.10 จากสมการ (5.34) และ (5.35) สามารถเขียนใหม่ได้เป็น

$$\beta_1 = \frac{2R_{31}}{R_{31} + R_{32} + R_{33}} \quad \dots(5.61)$$

$$\beta_2 = \frac{2R_{32}}{R_{31} + R_{32} + R_{33}} \quad \dots(5.62)$$

โดยที่ R_{31} , R_{32} และ R_{33} เป็นความต้านทานของพอร์ท 1, 2 และ 3 ของอะแดปเตอร์อนุกรม(3) ตามลำดับ สามารถคำนวณหาค่าได้ดังนี้

$$R_{31} = \frac{1}{G_{23}} = \frac{1}{3.9629} = 0.2523$$

$$R_{32} = L_2 = 2.285$$

$$R_{33} = R_L = 1$$

เมื่อแทนค่า R_{31} , R_{32} และ R_{33} ลงในสมการ (5.61) และ (5.62) สามารถหา β_1 และ β_2 ได้

$$\beta_1 = \frac{2 \times 0.2523}{0.2523 + 2.285 + 1} = 0.1426 \quad \dots(5.63)$$

$$\beta_2 = \frac{2 \times 2.285}{0.2523 + 2.285 + 1} = 1.2919 \quad \dots(5.64)$$

นำแทน β_1 และ β_2 ในสมการ (5.63) และ (5.64) แทนลงในสมการ (5.31) - (5.33) จะได้

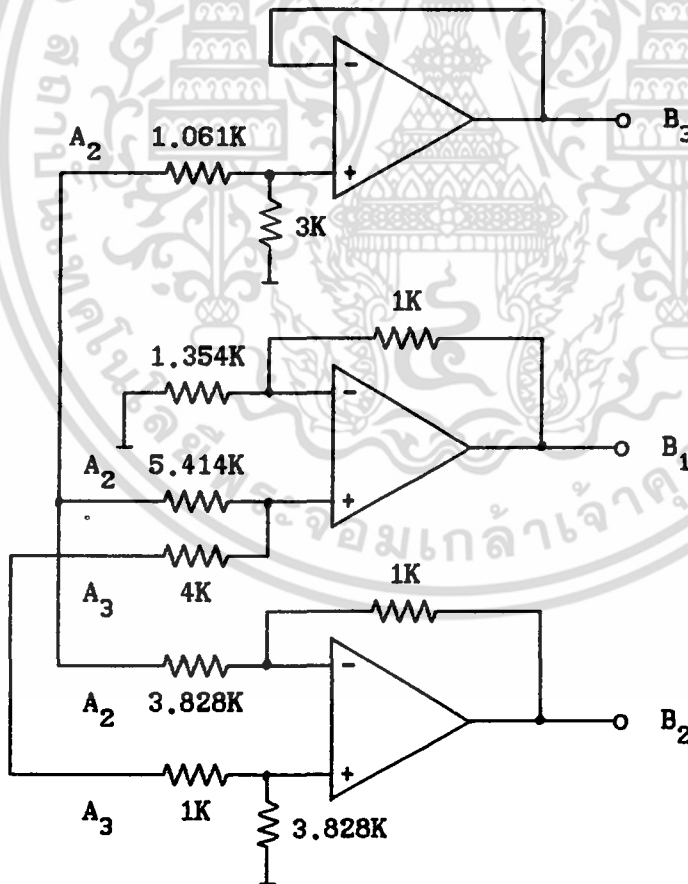
$$B_1 = 0.8574A_1 - 0.1426A_2 - 0.1426A_3 \quad \dots(5.65)$$

$$B_2 = -1.2919A_1 - 0.2919A_2 - 1.2919A_3 \quad \dots(5.66)$$

$$B_3 = -0.5655A_1 - 0.5655A_2 + 0.4345A_3 \quad \dots(5.67)$$

การออกแบบวงจรอะนาล็อก

วงจรอะนาล็อกจะประกอบด้วยวงจรรวมและวงจรคูณ ในการออกแบบเราจะใช้ข้อต่อร่วมกับความต้านทาน(R) จากสมการ (5.28) - (5.30) ซึ่งเป็นสมการคุณลักษณะของเฟรเ็นท์เวิร์ทของ 3 พอร์ต อะนาล็อกแบบขนาน โดยที่พอร์ท 3 เป็นพอร์ท Reflection Free และพอร์ท 2 เป็น Dependent Port สามารถนำมาออกแบบเป็นวงจรได้ดังแสดงในรูปที่ 5.12

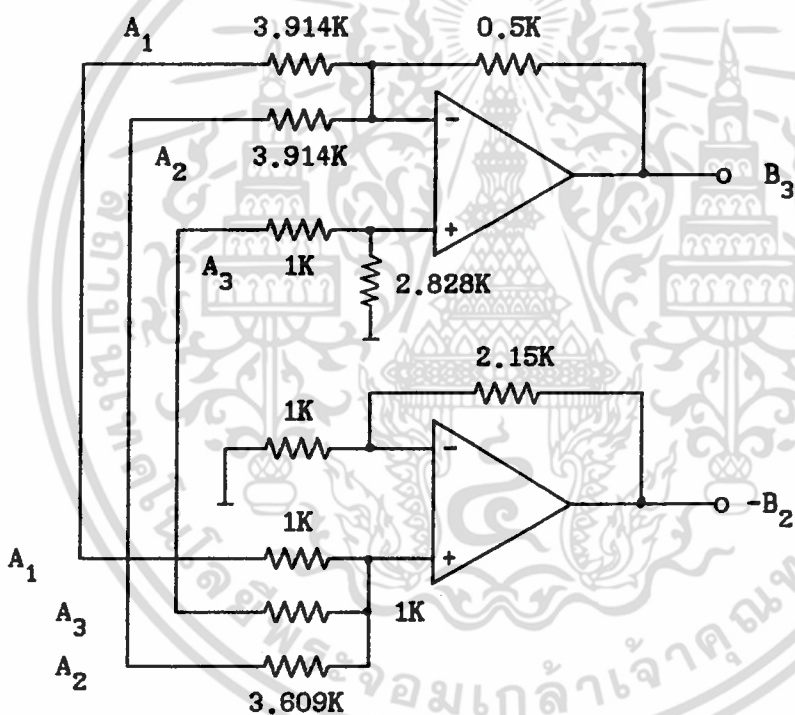


รูปที่ 5.12 วงจรอะนาล็อกแบบขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.12 แสดงถึงวงจรอะแดปเตอร์ขนาน จากวงจรในรูปที่ 5.6 จะเห็นได้ว่า พอร์ต 1 ของอะแดปเตอร์ขนานเป็นเอาต์พุตพอร์ทัลและต่อเข้ากับ Digital Sink เพราะฉะนั้น อินพุต A_1 สามารถละไว้ได้ เอาต์พุต B_2 ของพอร์ต 2 จะต่อเข้ากับดีเลย์ยูนิต ส่วน B_1 เป็น เอาต์พุต

จากสมการ (5.38) - (5.40) ซึ่งเป็นสมการคุณลักษณะของเฟรเอนท์เวิร์กของ 3 พอร์ต อะแดปเตอร์อนุกรม โดยที่พอร์ต 3 เป็น Dependent Port สามารถนำมาออกแบบ เป็นวงจรได้ดังแสดงในรูปที่ 5.13



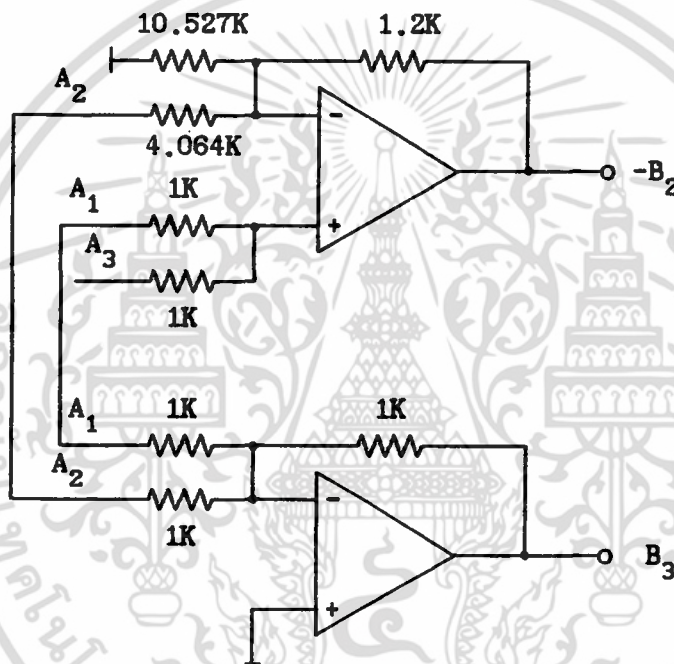
รูปที่ 5.13 วงจรอะแดปเตอร์แบบอนุกรม

รูปที่ 5.13 แสดงถึงวงจรอะแดปเตอร์อนุกรม จากวงจรในรูปที่ 5.6 จะเห็นได้ ว่าพอร์ต 1 ของอะแดปเตอร์อนุกรมเป็นอินพุตพอร์ทัลและต่อเข้ากับ Digital Sink เพราะฉะนั้นเอาต์พุต B_1 สามารถละไว้ได้ เอาต์พุต B_2 ของพอร์ต 2 จะต่อเข้ากับดีเลย์ยูนิตและคาสเดดกับอินเวอร์เตอร์ แต่ตัวอินเวอร์เตอร์นั้นได้ออกแบบรวมเข้ากับในวงจรแล้ว โดยที่เอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์ท 2 จะตรงข้ามกับ B_2 คือเป็น $-B_2$ ส่วนเอาต์พุต B_3 ต่อเข้ากับอินพุต A_3 ของอะแดปเตอร์ขนานดังรูปที่ 5.6

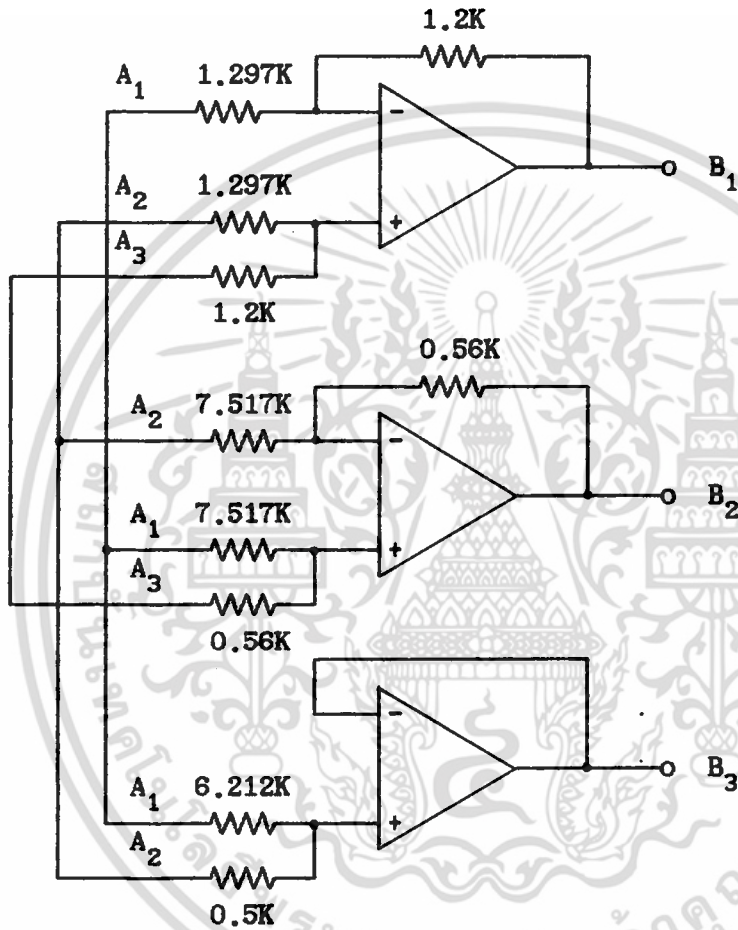
จากสมการ (5.53) - (5.55) ซึ่งเป็นสมการคุณลักษณะของเวฟเน็ตเวิร์กของ 3 พอร์ท อะแดปเตอร์แบบอนุกรม (1) โดยที่พอร์ท 3 เป็นพอร์ท Reflection Free และพอร์ท 2 เป็น Dependent Port สามารถนำมาออกแบบเป็นวงจรได้ดังแสดงในรูปที่ 5.14



รูปที่ 5.14 วงจรอะแดปเตอร์แบบอนุกรม (1)

รูปที่ 5.14 แสดงถึงวงจรอะแดปเตอร์อนุกรม (1) จากวงจรในรูปที่ 5.8 จะเห็นได้ว่าพอร์ท 1 ของอะแดปเตอร์เป็นอินพุตพอร์ทและต่อเข้ากับ Digital Sink เพราะฉะนั้นเอาต์พุต B_1 สามารถลະไว้ได้ เอาต์พุต B_2 ของพอร์ท 2 จะต่อเข้ากับดีเลย์ยูนิตและคาสเดคกับอินเวอร์เตอร์ แต่ตัวอินเวอร์เตอร์นั้นได้ออกแบบรวมเข้ากับในวงจรแล้ว โดยที่เอาต์พุตพอร์ท 2 จะตรงข้ามกับ B_2 คือเป็น $-B_2$

จากสมการ (5.58) - (5.60) ซึ่งเป็นสมการคุณลักษณะของเวฟเน็ทเวอ์กร์ของ 3 พอร์ท อะแดปเตอ์ชานาน (2) โดยที่พอร์ท 3 เป็นพอร์ท Reflection Free และพอร์ท 2 เป็น Dependent Port สามารถนำมาออกแบบเป็นวงจรได้ดังแสดงในรูปที่ 5.15

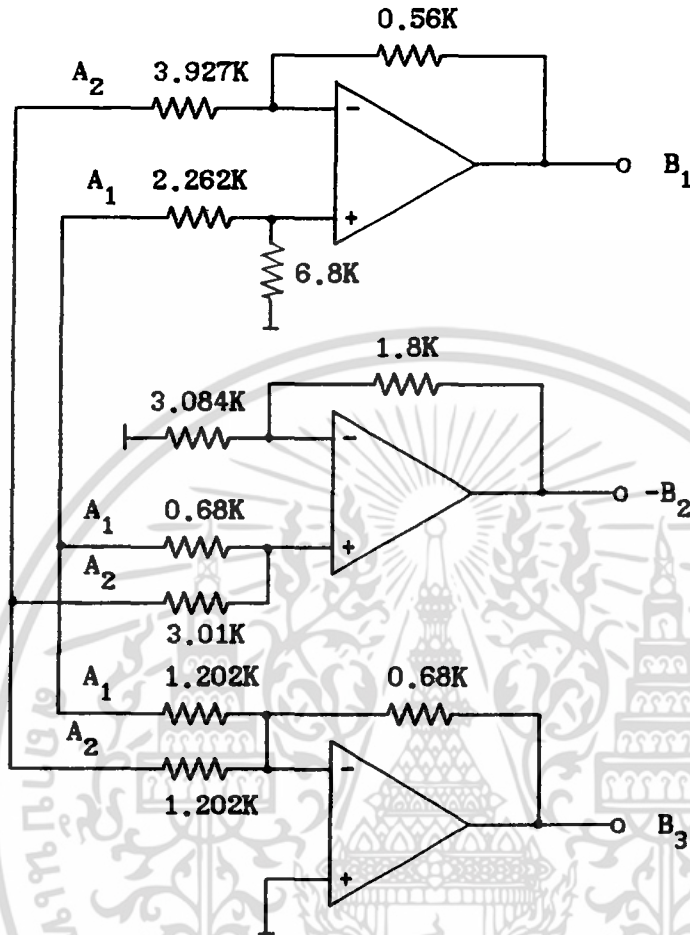


รูปที่ 5.15 วงจรอะแดปเตอ์ชานาน (2)

รูปที่ 5.15 แสดงถึงวงจรอะแดปเตอ์ชานาน (2) โดยที่ B_1 , B_2 และ B_3 เป็นเอาต์พุท

และในทำนองเดียวกัน จากสมการ (5.65) - (5.67) ซึ่งเป็นสมการคุณลักษณะของเวฟเน็ทเวอ์กร์ของ 3 พอร์ทอะแดปเตอ์ชานาน (3) โดยที่พอร์ท 3 เป็น Dependent Port สามารถนำมาออกแบบเป็นวงจรได้ ดังแสดงในรูปที่ 5.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

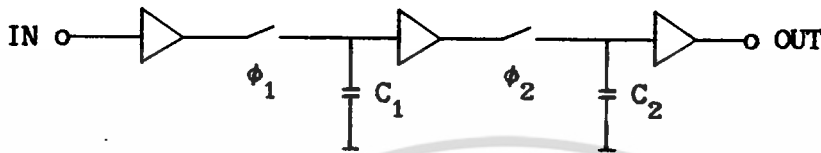


รูปที่ 5.16 วงจรอะแดปเตอร์แบบอนุกรม (3)

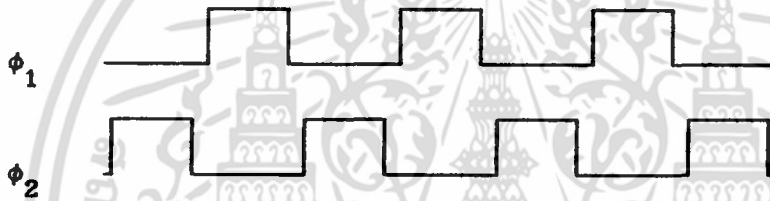
รูปที่ 5.16 แสดงถึงวงจรอะแดปเตอร์อนุกรม (3) จากวงจรในรูปที่ 5.8 พอร์ต 3 ของอะแดปเตอร์อนุกรม (3) จะเป็นเอาต์พุตพอร์ท และต่อเข้ากับ Digital Sink เพราะฉะนั้นอินพุต A_3 สามารถละไว้ได้

สำหรับดีเลย์ยูนิต จะใช้วงจรแซมเปิลและโฮลด์ (Sample-and-Hold) ต่อкасาดกัน 2 Stage ดังแสดงในรูปที่ 5.17 จากวงจรต้องใช้สัญญาณนาฬิกา 2 ชุด คือ ϕ_1 และ ϕ_2 เป็นสัญญาณ Sampling C_1 และ C_2 เป็น Holding Capacitor การทำงานช่วงเริ่มแรก เฟสของสัญญาณนาฬิกา ϕ_2 จะแซมเปิลสัญญาณอินพุตก่อน สัญญาณอินพุตก็จะไปปรากฏที่เอาต์พุตของวงจรแซมเปิลและโฮลด์ ในสัญญาณนาฬิกาต่อมาเอาต์พุตของวงจรอยู่ในโหมดโฮลด์ ขณะเดียวกัน

สัญญาณนาฬิกา ϕ_1 จะควบคุมเปิดสัญญาณทางอินพุต ทำงานเช่นนี้ต่อเนื่องกันไป ซึ่งลักษณะดีเลย์ของ วงจรแชนเปลและโฮลด์ที่ต่อคาสเดคกัน 2 Stage แสดงไว้ในรูปที่ 2.17 ส่วนในรูปที่ 5.18 แสดงถึงสัญญาณนาฬิกา (Clock) ที่ใช้กับดีเลย์ยูนิต



รูปที่ 5.17 วงจรแชนเปลและโฮลด์

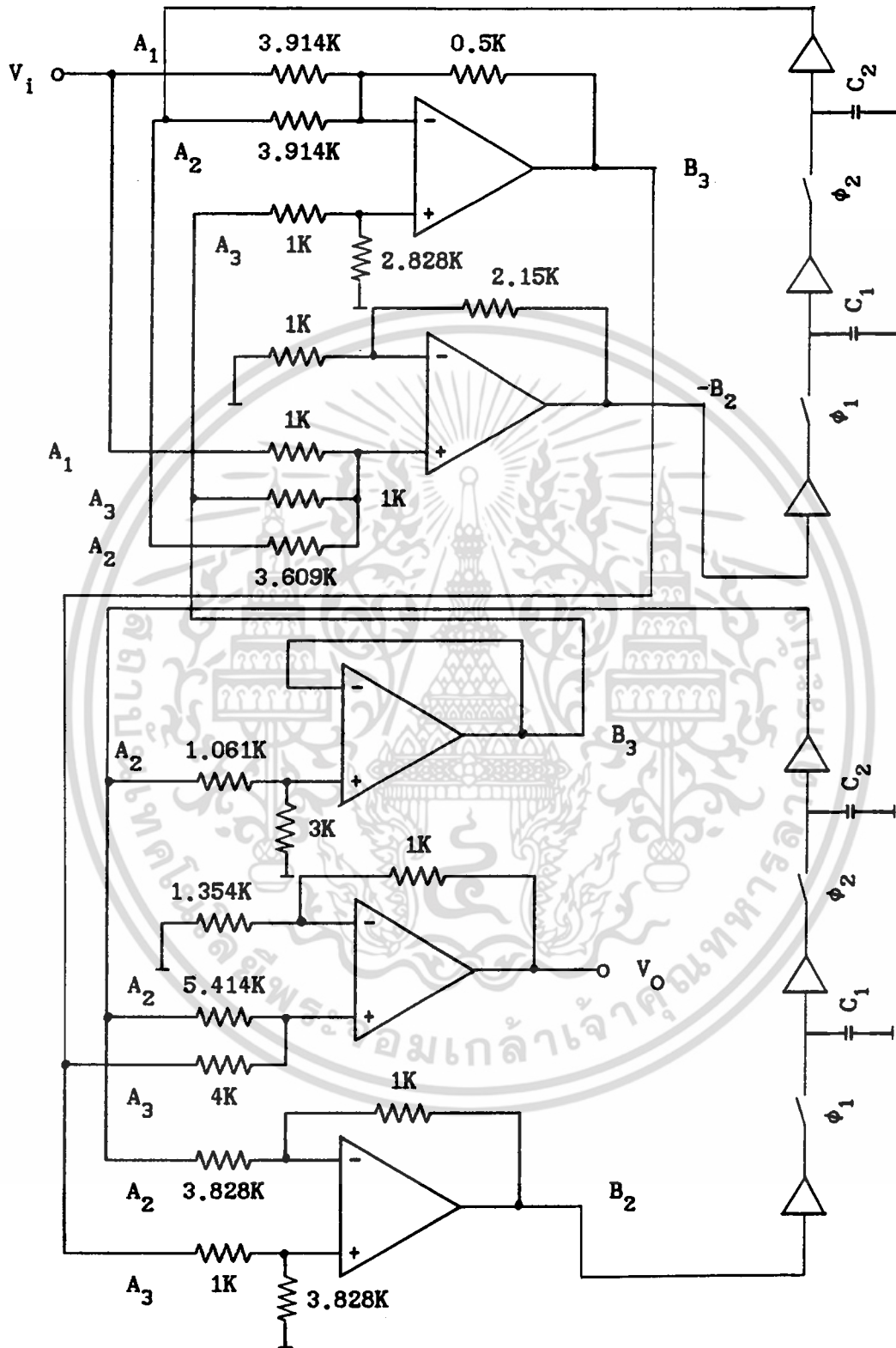


รูปที่ 5.18 สัญญาณนาฬิกา ϕ_1 และ ϕ_2

5.3 ผลการทดสอบวงจร

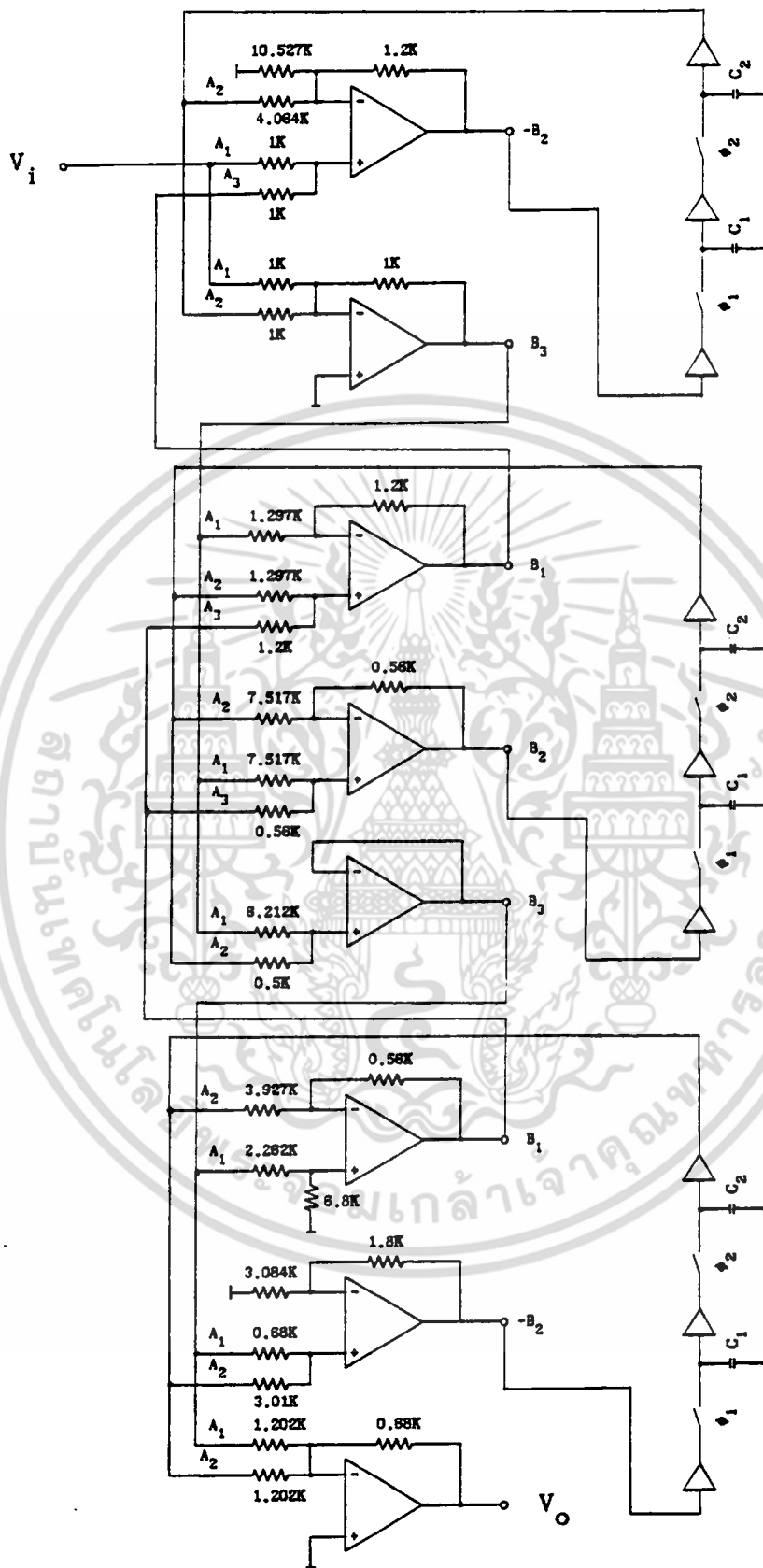
เพื่อเป็นการยืนยันให้เห็นว่า วงจรกรองความถี่ต่ำเวฟดิสครีทฟิลเตอร์แบบบัตเตอร์เวท และวงจรกรองความถี่ต่ำเวฟดิสครีทฟิลเตอร์แบบบัตเตอร์เวท-อูลตราสเฟียริคัล โพลีโนเมียลที่ออกแบบไว้ ดังแสดงในรูปที่ 5.19 และ 5.20 สามารถนำไปใช้งานในเชิงปฏิบัติได้จริง จึงได้ทำการทดสอบวงจร โดยใช้สัญญาณนาฬิกา (Clock) ϕ_1 และ ϕ_2 บ้อนเข้าวงจรแชนเปลและโฮลด์ ซึ่งจะทำหน้าที่เป็นดีเลย์ยูนิต โดยที่ ϕ_1 และ ϕ_2 เป็นสัญญาณที่ใช้สำหรับ Sampling และมีเฟสตรงข้ามกัน ดังแสดงในรูปที่ 5.18 ในการทดลองนี้ ได้ใช้ความถี่ Sampling ที่ 125 KHz, 250 KHz และ 500 KHz ผลตอบสนองทางขนาดต่อความถี่ที่วัดค่าได้จากวงจรกรองความถี่ต่ำเวฟดิสครีทฟิลเตอร์แบบบัตเตอร์เวทแสดงดังในรูปที่ 5.21, 5.22 และ 5.23 ตามลำดับ และผลตอบสนองทางขนาดต่อความถี่ที่วัดค่าได้จากวงจรกรองความถี่ต่ำเวฟดิสครีทฟิลเตอร์แบบบัตเตอร์เวท-อูลตราสเฟียริคัล โพลีโนเมียลแสดงดังในรูปที่ 5.24, 5.25 และ 5.26 ตามลำดับ

จากผลการทดสอบวัดค่าผลตอบสนองทางความถี่ของวงจรของความถี่ต่ำเวฟดิสครีทฟิลเตอร์ตามที่ได้แสดงแล้วนั้น จะเห็นได้ว่าถ้าใช้ความถี่ Sampling 125 KHz จะได้ความถี่คัทออฟประมาณ 20 KHz และใช้ความถี่ Sampling 250 KHz, 500 KHz จะได้ความถี่คัทออฟประมาณ 40 KHz, 80 KHz ตามลำดับ ผลที่ได้แสดงให้เห็นว่า ความถี่คัทออฟของวงจรสามารถกำหนดได้จากสัญญาณความถี่ Sampling ที่ใช้กับวงจรแชนเนลและโฮลด์ ถ้าเพิ่มความถี่ Sampling เป็น 2 เท่า จะได้จุดคัทออฟเพิ่มขึ้น 2 เท่าด้วย ซึ่งความถี่สูงสุดของสัญญาณ Sampling จะถูกจำกัดโดยความเร็วในการทำงานของวงจรแชนเนลและโฮลด์ และเมื่อเปรียบเทียบผลตอบสนองทางขนาดของอนุโลกกับผลตอบสนองทางขนาดที่วัดค่าได้จากการทดลอง จะได้ว่าคุณลักษณะผลตอบสนองทางขนาดที่วัดค่าได้จากการทดลองให้ผลใกล้เคียงกับผลทางทฤษฎี ซึ่งสามารถลดทอนสัญญาณในย่านหยุดสัญญาณได้ดีกว่าอนุโลกฟิลเตอร์ เหตุที่เป็นเช่นนี้เนื่องจากผลของไบเลเนียร์ทรานส์ฟอร์ม และสามารถทดสอบ Sensitivity ของวงจร โดยกำหนดให้อิลิเมนต์ R มีค่าผิดพลาดประมาณ 5% จากรูปสังเกตเห็นได้ว่า ผลตอบสนองทางขนาดที่วัดค่าได้จากการทดลองมีค่าใกล้เคียงกับค่าที่วัดได้จริง แสดงให้เห็นว่าวงจรที่ออกแบบได้มีค่าความไวแปรเปลี่ยนในตัวอิลิเมนต์ต่ำ

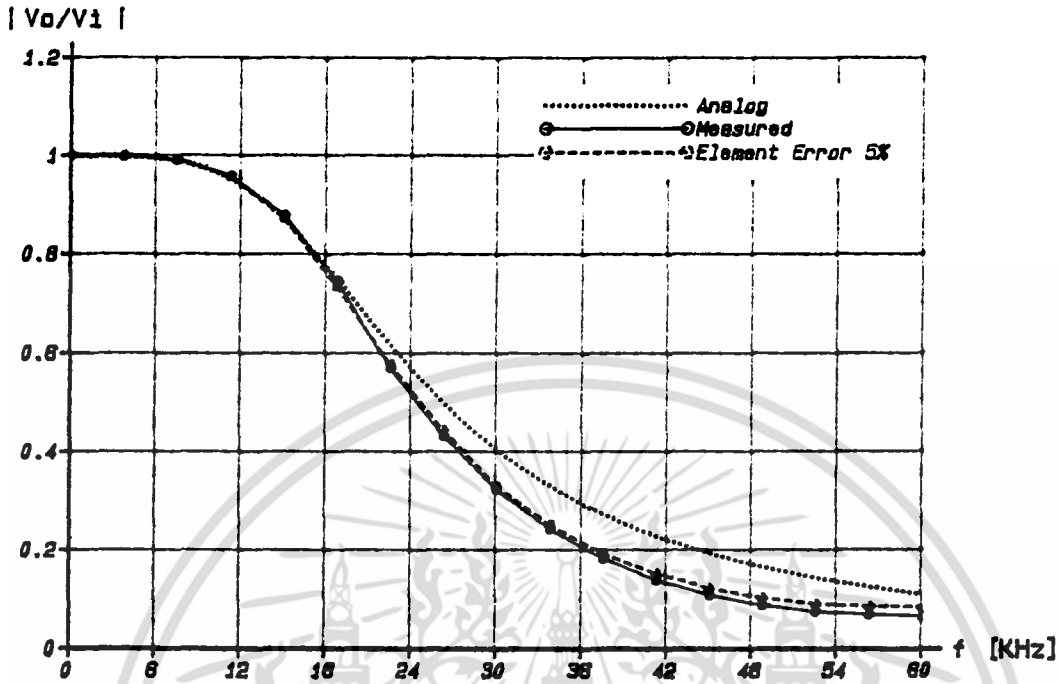


รูปที่ 5.19 วงจรกรองความถี่ต่ำแบบฟิลเตอร์แบบบัตเตอร์โวลต์เวจอันดับ 2

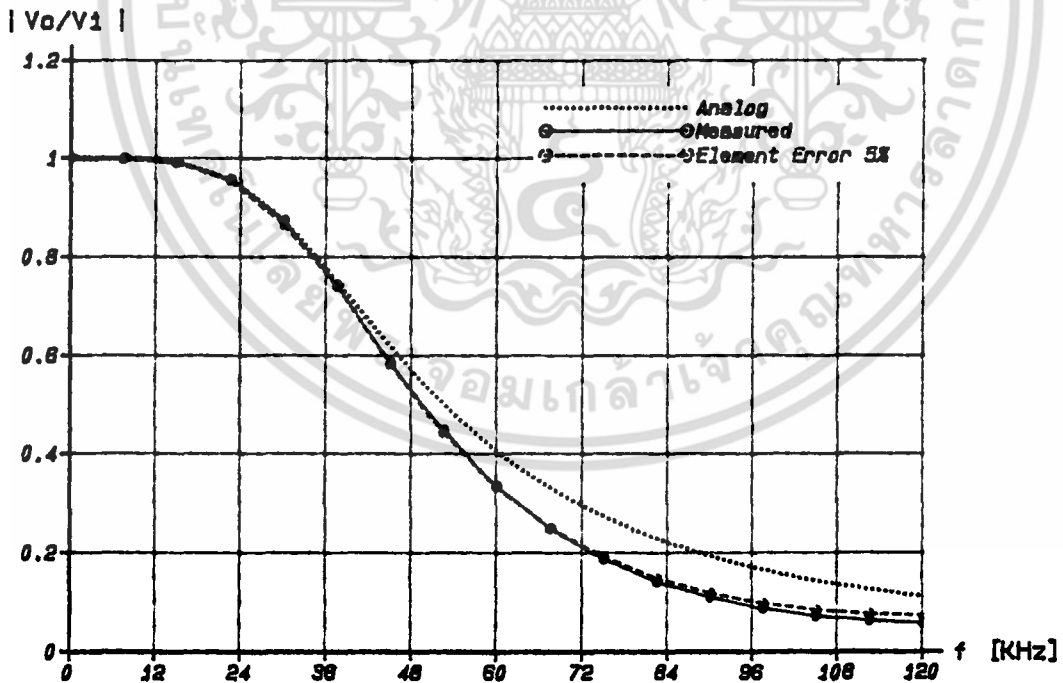
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.20 วงจรกรองความถี่ต่ำเวฟสี่เหลี่ยมแบบบัตเตอร์เวิร์ท-อลด์ริสเพื่อวีคัล เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า โพลีโนเมียลอันดับ 3 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

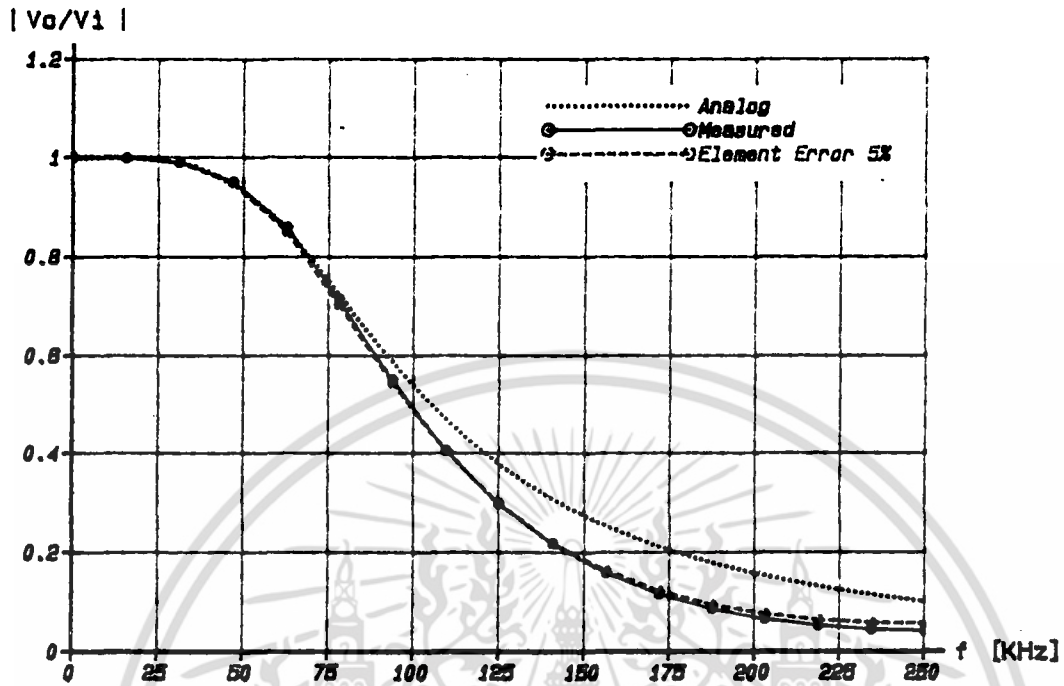


รูปที่ 5.21 ผลตอบสนองทางขนาดต่อความถี่วัดได้จากวงจรในรูปที่ 5.19 โดยใช้ความถี่ Sampling 125 KHz

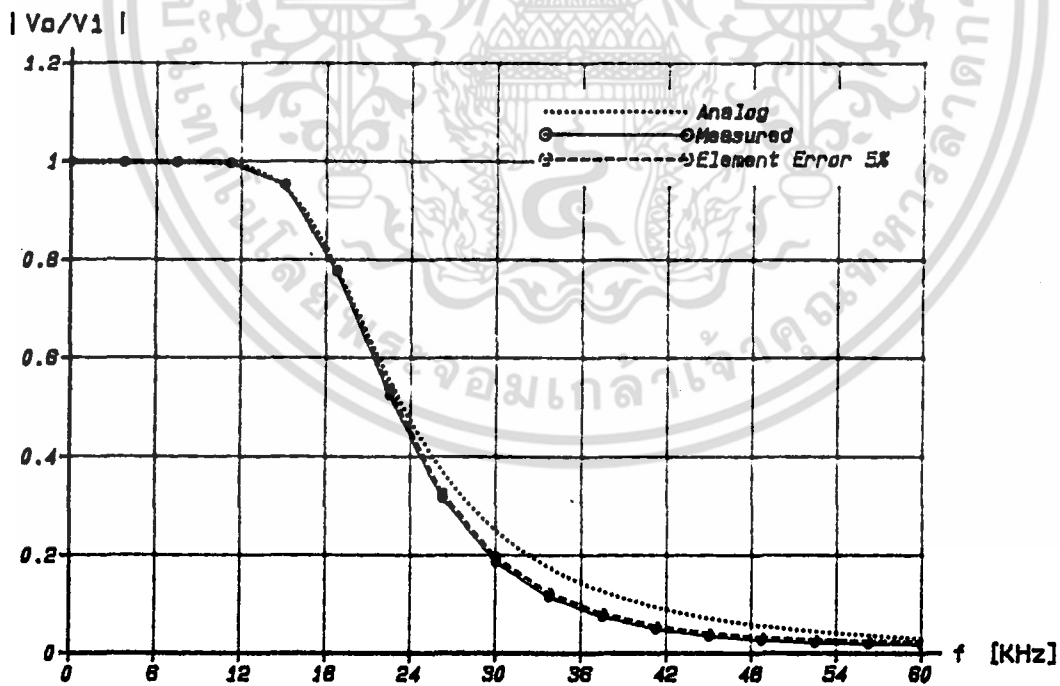


รูปที่ 5.22 ผลตอบสนองทางขนาดต่อความถี่วัดได้จากวงจรในรูปที่ 5.19 โดยใช้ความถี่ Sampling 250 KHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

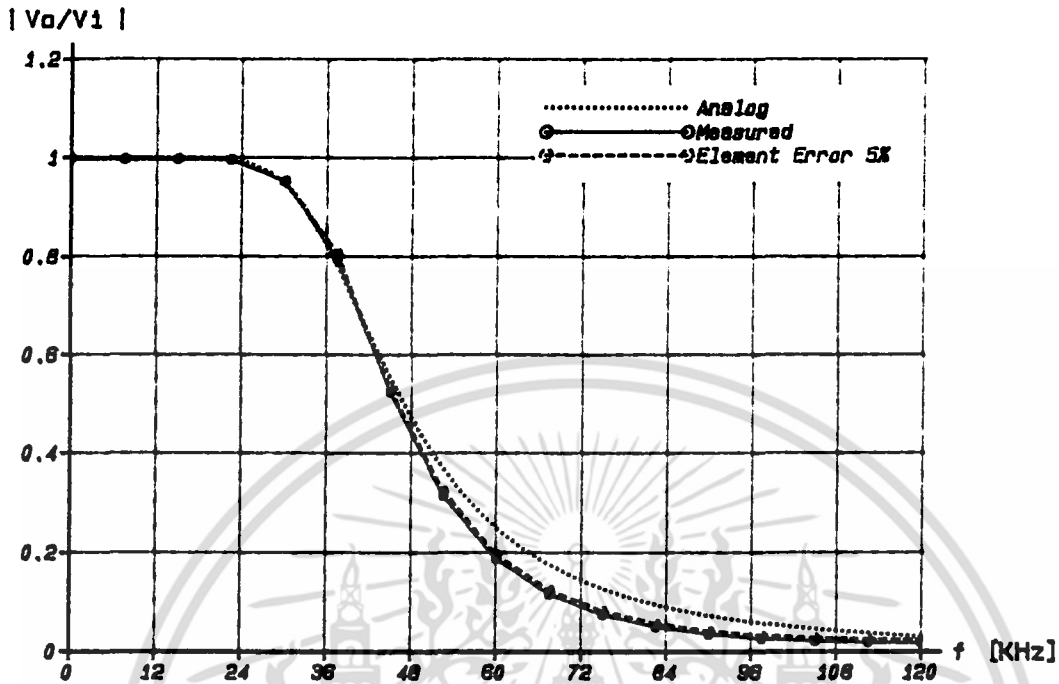


รูปที่ 5.23 ผลตอบสนองทางขนาดต่อความถี่ที่วัดได้จากวงจรในรูปที่ 5.19 โดยใช้ความถี่ Sampling 500 KHz

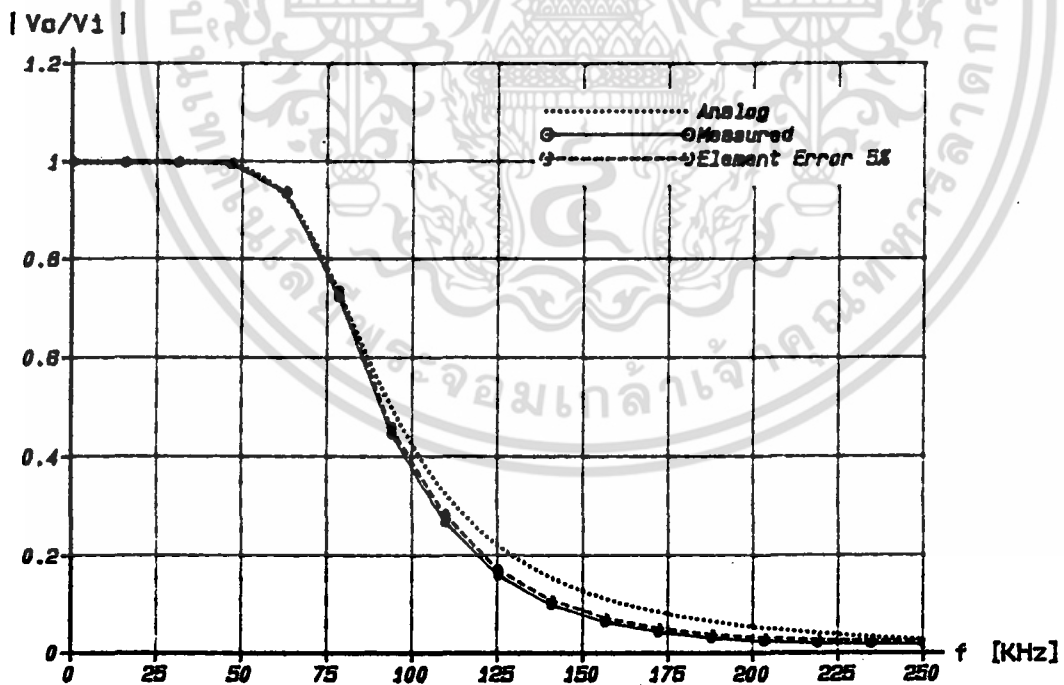


รูปที่ 5.24 ผลตอบสนองทางขนาดต่อความถี่ที่วัดได้จากวงจรในรูปที่ 5.20 โดยใช้ความถี่ Sampling 125 KHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.25 ผลตอบสนองทางขนาดต่อความถี่ที่วัดได้จากวงจรในรูปที่ 5.20 โดยใช้ความถี่ Sampling 250 KHz



รูปที่ 5.26 ผลตอบสนองทางขนาดต่อความถี่ที่วัดได้จากวงจรในรูปที่ 5.20 โดยใช้ความถี่ Sampling 500 KHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6 สรุปผลการวิจัย

จากการออกแบบวงจรกรองความถี่ต่ำแบบคอมโพสิตโพลีโนเมียลบัตเตอร์เวิร์ท-อัลตราสเฟียริคัล อันเป็นวิธีการใหม่ ซึ่งสามารถกำหนดคุณลักษณะของผลตอบสนองทางขนาดอยู่ระหว่างบัตเตอร์เวิร์ทกับอัลตราสเฟียริคัล โพลีโนเมียล จึงทำให้สามารถกำหนดค่าผลตอบสนองจากผลตอบสนองทางขนาดในช่วงความถี่ผ่าน (Passband) ราบเรียบที่สุด (Maximally Flat Amplitude) ไปจนถึงช่วงคutoffที่ขึ้นซึ่งเป็นแบบ Equal Ripple โดยมีพหามิเตอร์ที่สามารถกำหนดผลตอบสนองทางขนาดได้ จากคุณลักษณะพิเศษดังกล่าว จึงได้นำวิธีการนี้มาสร้างวงจรกรองความถี่ต่ำ โดยอาศัยทฤษฎีการแปลงโครงสร้างของวงจรจากวงจรพาสซีฟ LC มาเป็นวงจรเวฟดีสครีทฟิลเตอร์ โดยออกแบบวงจรอะแคปเตออร์อนุกรมและอะแคปเตออร์ขนาน ต่อร่วมกับวงจรแชนเนลและไฮลด์ ซึ่งทำหน้าที่เป็นดีเลย์ยูนิต โดยช่วงความถี่คutoffของวงจรสามารถกำหนดได้จากสัญญาณนาฬิกาที่ใช้กับวงจรแชนเนลและไฮลด์ ในการทดลองนี้ ได้ใช้ความถี่ของสัญญาณนาฬิกา 125 KHz, 250 KHz และ 500 KHz มาทำการทดสอบวงจรกรองความถี่ต่ำเวฟดีสครีทฟิลเตอร์ ทั้งสองแบบคือแบบบัตเตอร์เวิร์ทและแบบบัตเตอร์เวิร์ท-อัลตราสเฟียริคัล โพลีโนเมียล ซึ่งจากผลการทดสอบพบว่าวัดค่าของผลตอบสนองทางขนาดต่อความถี่ได้ใกล้เคียงกับผลทางทฤษฎี ซึ่งความถี่สูงสุดของสัญญาณนาฬิกา จะถูกจำกัดโดยความเร็วในการทำงานของวงจรแชนเนลและไฮลด์ โดยวงจรดังกล่าวให้ค่าความไวของอิลิเมนต์ต่ำและสามารถผลิตเป็นวงจร LSI ที่มีขนาดเล็กได้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดี ด้วยความกรุณาของผู้ช่วยศาสตราจารย์ ดร.กนก เจนจิระพงศ์เวช ที่ได้ให้การประสิทธิ์ประสาทวิชาการแก่ผู้เขียน ตลอดจนช่วยแนะนำ และให้คำปรึกษาในการทำวิทยานิพนธ์ ผู้เขียนขอกราบขอบพระคุณเป็นอย่างสูง ขอขอบคุณท่าน อาจารย์และเจ้าหน้าที่ภาควิชาเทคนิคอุตสาหกรรมทุกท่าน ที่ได้กรุณาให้ความช่วยเหลือและสนับสนุนการวิจัยมาโดยตลอด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. ชาลิต บุญจางคประเสริฐ, ธนิตพงษ์ วิบูลยานนท์, กนก เจนจิระพงศ์เวช, "การออกแบบวงจรเวฟดิจิทัลฟิลเตอร์แบบบัตเตอร์เวิร์กอันดับสอง" การประชุมทางวิชาการทางวิศวกรรมไฟฟ้า 9 สถาบัน ครั้งที่ 13 มหาวิทยาลัยเชียงใหม่, หน้า 81-91, พฤศจิกายน 2533
2. ชาลิต บุญจางคประเสริฐ, ไพศาล สิทธิโยภาสกุล, กนก เจนจิระพงศ์เวช, "การประมาณเท็เว็กร์ฟังก์ชันสำหรับวงจรของความถี่ต่ำแบบที่ใหกรู้อดีเลย์คงที่" การประชุมทางวิชาการทางวิศวกรรมไฟฟ้า 9 สถาบัน ครั้งที่ 12 มหาวิทยาลัยเกษตรศาสตร์, หน้า 202-215, พฤศจิกายน 2532
3. ไพศาล สิทธิโยภาสกุล, กนก เจนจิระพงศ์เวช, "วงจรของความถี่แบบบัตเตอร์เวิร์ก-อูลตราสเฟียริคัล โพลีโนเมียล" การประชุมทางวิชาการทางวิศวกรรมไฟฟ้า 9 สถาบัน ครั้งที่ 10 จุฬาลงกรณ์มหาวิทยาลัย, เล่มที่ 2 หน้า 239-247, พฤศจิกายน 2530
4. A. Fettweis and K. Meerkotter, "On Adaptor for Wave Digital Filters", IEEE trans. Acoustics and Signal Processing, Vol. ASSP-23, No. 6, December 1975.
5. A. Fettweis, "Wave digital Filters : Theory and Practice", Proceedings of the IEEE, Vol.74. No.2, pp.270-327, February 1986.
6. A. Antoniou, "Digital Filters Analysis and Design", McGraw-Hill, Inc. CH-12, New York, 1979.
7. D.F. Elliott, "Handbook of Digital Signal Processing Engineering Applications", Academic Press, Inc., 1987.
8. A. Fettweis, "Some Principles of Designing Digital Filters Imitating Classical Filter Structures", IEEE Trans.Circuit Theory, Vol.CT-18, pp.314-316, March 1971.
9. A. Sedlmeyer and A. Fettweis, "Digital Filters with True Ladder Configuration", Int.J. Circuit Theory Appl., Vol.1, pp.5-10, March 1973.
10. H.J. Orchard, "Inductorless Filters", Electron. Lett., Vol.2, pp.224-225, June 1966.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. D.E. Johnson, and J.R. Johnson, "Low pass Filters using Ultraspherical polynomials", IEEE Trans. on circuit Theory, Vol. CT-13, No.4, pp. 364-368, December 1966.
12. D.E. Johnson, "Introduction to Filter Theory", Prentice-Hall, pp.44-47, 1976.
13. E.D. Rainville, "Special Functions", The Macmillan Company, pp. 254-283, 1967.
14. G.S. Moschytz and P. Horn, "Active Filter Design Handbook", John Wiley & Sons, New York, 1981.
15. Y. Pelless, and T. Murakami, "Analysis and Synthesis of Transitional Butterworth-Thomson Filters and Bandpass Amplifiers", RCA Review, Vol.18, pp.66-94, March 1957.
16. G.C. Temes and S.K. Mitra, "Modern Filter Theory and Design", John Wiley & Sons, Inc., 1973.
17. A. Budak, "Passive and Active Network Analysis and Synthesis", Houghton Mifflin Company, 1974.
18. V. Valkenburg, "Analog Filter Design", Holt Saunders, New York, 1982.
19. J. Vlach and K. Singhal, "Computer Methodes for Circuit Analysis and Design", Van Nostrand Reinhold Company, 1983.
20. M.N. Horenstein, "Microelectronic circuits & Devices", Prentice-Hall International Inc. Edition, 1990.
21. J. Millman, "Microelectronic : Digital and Analog circuit and Systems", McGraw-Hill, Inc., 1979.
22. M.S. Ghausi and J.J. Kelley, "Introduction to Distributed-parameter Network", Holt Rinehart and Winston, Inc., 1968.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
program Composite_Polynomials;

const XScreenMax = 720;      { Number of pixels in one screen line }
      YScreenMax = 350;      { Number of lines on the screen      }
      ScreenSizeGlb = 16383; { Total size in integers of the screen}
      HardwareGrafBase = $B000; { Location of the hardware screen  }
      SCF = 1.542857;        { Screen Scaling adjustment Factor    }
      VRowsGlb = $58; {Change to $57 if monitor loses horizontal hold }
      RamScreenInCard : boolean = false;
      FontLoaded : boolean = false; { has the font been loaded yet? }
      GrafModeGlb : boolean = false;
      EGA : boolean = false;

type ScreenType = array [0..ScreenSizeGlb] of integer;
   FontChar = array [0..13] of byte;
   GrfFont = array [0..255] of FontChar;
   RegType = record case integer of
       1 : (AX,BX,CX,DX,BP,SI,DI,DS,ES,Flags : integer);
       2 : (AL,AH,BL,BH,CL,CH,DL,DH : byte);
   end;

var Regs : RegType;
    ScreenGlb : ^ScreenType;
    GrafBase : integer;
    ColorGlb : byte;
    Font : GrfFont;
    ConOutPtrSave,
    XTextGlb,YTextGlb : integer;

procedure HerculesMode(Mode: integer); { 0=Text, 1=Graphics }
type ModeDescriptor = record
    CRTMode : byte;
    R6845 : array[0..11] of byte;
end;

const VRo = VRowsGlb;
      HercModes : array[0..1] of ModeDescriptor =
      ((CRTMode:32; R6845:($61,$50,$52,$0F,$19,$06,
        $19,$19,$02,$0D,$0B,$0C)),
      (CRTMode:2; R6845:($35,$2D,$2E,$07,$5B,$02,
        VRo,VRo,$02,$03,$00,$00)));

begin
  Delay(100);
  inline($8B/$B6/Mode/$81/$E6/$01/$00/$74/$03/$BE/$0D/$00/$81/$C6/
    HercModes/$BA/$BF/$03/$B0/$01/$EE/$2E/$8A/$04/$B2/$B8/$EE/
    $B9/$00/$00/$B2/$B4/$88/$C8/$EE/$46/$2E/$8A/$04/$42/$EE/
    $4A/$41/$80/$F9/$0C/$75/$F0/$83/$EE/$0C/$2E/$8A/$04/$0C/
    $08/$B2/$B8/$EE);
  if RAMScreenInCard then
    Port[$03BF] := 3;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure ClearScreen;
begin
  fillchar(mem[GrafBase:0000],ScreenSizeGlb Shl 1,0);
end;

procedure TextMode;
begin
  HerculesMode(0);
  ClrScr;
  if GrafModeGlb then ConOutPtr := ConOutPtrSave;
  GrafModeGlb := false;
end;

procedure DC(C : byte);
  { Draw the the character with ASCII code C
    at position (XTextGlb, YTextGlb). }
var XShr3      : integer;
    RightBit   : boolean;
begin
  Inline($8A/$AE/ C /$8A/$C5/$24/$E0/$3C/$C0/$75/$02/$B0/$01/$24/
    $01/$88/$86/ RightBit /$8A/$1E/ XTextGlb /$8A/$3E/ YTextGlb/
    $8A/$C5/$32/$E4/$8B/$D0/$B1/$04/$D3/$E0/$D1/$E2/$2B/$C2/$8B/
    $F0/$A1/ GrafBase /$8E/$C0/$81/$EB/$01/$01/$32/$F6/$8A/$D3/
    $B1/$03/$D3/$E2/$02/$D3/$80/$D6/$00/$8B/$C2/$D3/$E8/$89/$86/
    XShr3 /$8A/$CA/$80/$E1/$07/$8A/$D7/$BB/$80/$FF/$D3/$EB/$F7/
    $D3/$80/$F1/$07/$32/$F6/$8B/$C2/$D1/$E2/$D1/$E2/$D1/$E2/$2B/
    $D0/$D1/$E2/$B5/$0D/$53/$51/$87/$D9/$32/$E4/$8A/$C7/$03/$C2/
    $8B/$D8/$80/$E3/$03/$B1/$0D/$D3/$E3/$D1/$E8/$D1/$E8/$B4/$5A/
    $F6/$E4/$03/$C3/$03/$86/ XShr3 /$8B/$F8/$59/$5B/$26/$8B/$05/
    $86/$C4/$23/$C3/$53/$8A/$DD/$32/$FF/$8A/$98/ Font /$50/$8A/
    $86/ RightBit /$22/$C3/$D1/$E3/$0A/$D8/$58/$D3/$E3/$0B/$C3/
    $86/$C4/$26/$89/$05/$5B/$FE/$CD/$79/$B1);
end;

procedure DisplayChar(C : byte);
{ Same as DC. Intended for internal use by the graphics system }
begin
  if C=8 then
    begin
      if XTextGlb>1 then XTextGlb := XTextGlb - 1;
    end
  else
    if C=10 then
      begin
        if YTextGlb<25 then YTextGlb := YTextGlb + 1;
      end
    else
      if C=13 then XTextGlb := 1
      else
        begin
          DC(C);
          if XTextGlb<80 then XTextGlb := XTextGlb + 1;
        end
      end
    end
  end
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    end;
end;

procedure GraphicsMode;
var FontFile : file of GrfFont;
begin
  GrafBase := HardwareGrafBase;
  if not FontLoaded then
    begin
      Assign(FontFile, '14x9.FON');
      {$I-} Reset(FontFile); {$I+}
      if IOresult=0 then
        begin
          Read(FontFile, Font);
          Close(FontFile);
        end
      else
        FillChar(Font, SizeOf(Font), 0);
        FontLoaded := true;
      end;
    HerculesMode(1);
    if not GrafModeGlb then ConOutPtrSave := ConOutPtr;
    ConOutPtr := Ofs(DisplayChar);
    ClearScreen;
    GrafModeGlb := true;
  end;

procedure AllocateRAMScreen;
var test : ^integer;
begin
  new(ScreenGlb);
  while ofs(ScreenGlb^) <> 0 do
    begin
      dispose(ScreenGlb);
      new(test);
      new(ScreenGlb);
    end;
  end;

procedure SelectScreen(i:integer);
begin
  case I of
    1 : GrafBase := HardwareGrafBase;
    2 : GrafBase := Seg(ScreenGlb^);
  end;
end;

procedure CopyScreen;
var ToBase : integer;
begin
  GrafBase := Seg(ScreenGlb^);
  ToBase := HardwareGrafBase;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    move(mem[GrafBase:0000],mem[ToBase:0000],ScreenSizeGlb shl 1);
end;

procedure SwapScreen;
const SS = $4000; { ScreenSizeGlb+1 }
var G      : integer;
begin
    G := Seg(ScreenGlb^);
    Inline($8B/$86/ G /$8E/$C0/$1E/$B8/ HardwareGrafBase /$8E/$D8/$B9/
           SS /$31/$DB/$8B/$07/$26/$87/$07/$89/$07/$43/$43/$E2/$F5/$1F);
end;

procedure DrawPoint(X,Y: integer);
begin
    Inline($8B/$96/ Y /$B8/$03/$00/$21/$D0/$B1/$0D/$D3/$E0/$D1/$EA/$81/
           $E2/$FE/$FF/$89/$D3/$D1/$E2/$01/$DA/$89/$D3/$B1/$04/$D3/$E2/
           $29/$DA/$01/$D0/$8B/$96/ X /$89/$D3/$FE/$C9/$D3/$EB/$01/$C3/
           $88/$D1/$80/$E1/$07/$A1/ GrafBase /$8E/$C0/$26/$8A/$07/$8A/
           $2E/ ColorGlb /$08/$ED/$74/$09/$BA/$80/$00/$D3/$EA/$08/$D0/
           $EB/$07/$BA/$7F/$FF/$D3/$EA/$20/$D0/$26/$88/$07);
end;

procedure GotoXYTurbo(X,Y:integer);
begin
    GotoXY(X,Y);
end;

procedure GotoXY(X,Y:integer);
begin
    if not GrafModeGlb then GotoXYTurbo(X,Y);
    XTextGlb:=X;
    YTextGlb:=Y;
end;

procedure hardcopy(inverse: boolean; mode: byte);
label exit;
var i,j,top      : integer;
    ColorLoc,PrintByte : byte;

function BaseAddress(Y: integer): integer;
{ Calculate the address of scanline Y }
begin
    BaseAddress := (Y and 3) shl 13 + 90 * (Y shr 2);
end;

function PD(X,Y: integer): boolean;
{ Return true if the color of the pixel at (X, Y) matches ColorGlb }
begin
    PD := (ColorGlb = 0) xor (Mem[GrafBase:BaseAddress(Y) + X shr 3]
        and (128 shr (X and 7)) <> 0);
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure doline(top: integer);
function ConstructByte(j,i: integer): byte;
  const Bits : array [0..7] of byte=(128,64,32,16,8,4,2,1);
  var CByte,k : byte;
  begin
    i :=i shl 3;
    CByte := 0;
    for k := 0 to top do
      if PD(j,i+k) then CByte := CByte or Bits[k];
    ConstructByte := CByte;
  end;
var a: integer;
begin
  for a := 1 to 2 do
    begin
      if a=1 then write(lst,^[ '3'#1)
        else write(lst,^[ '3'#23);
      if mode=1 then write(lst,^[ 'L')
        else write(lst,^[ '*',chr(mode));
      write(lst,chr(lo(XScreenMax)),chr(Hi(XScreenMax)));
      for j := 0 to XScreenMax-1 do
        begin
          PrintByte := ConstructByte(j,i);
          if inverse then PrintByte := not PrintByte;
          write(lst,chr(PrintByte));
        end;
      if mode<>4 then writeln(lst);
    end;
  end;
begin
  top := 7;
  ColorLoc := ColorGlb;
  ColorGlb := 255;
  mode := mode and 7;
  if (mode=5) or (mode=0) then mode := 4;
  for i := 0 to (YScreenMax shr 3)-1 do
    begin
      doline(7);
      regs.ah := 6;
      regs.dl := $ff;
      MsDos(Regs);
      if (regs.al=27) then goto exit;
    end;
    i := (YScreenMax shr 3);
    if YScreenMax and 7<>0 then doline(YScreenMax and 7);
  exit:
    writeln(lst,^[ '2');
    ColorGlb := ColorLoc;
  end;
end;
```

```
procedure Wait(mode: byte);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
var ch : char;
begin
  repeat
    Read(kbd,ch);
    if ch=^p then Hardcopy(false,mode);
  until ch=^m;
end;

(*-----*)
const PScale = 2.5; { Plotter Scale (1-7) }
type complex = record
  Re,Im : real;
end;
var LineStyle : byte;
  ScreenDisp,freq : boolean;
  OutFile : text;
  RC,al : real;
  fi,ni,xs : integer;
  data : complex;
  num,ch,typ : char;

procedure SetLineStyle(m: byte);
begin
  case m of
    1.:if ScreenDisp then LineStyle := 1 {_____}
      else Write(OutFile,'LT;');
    2 :if ScreenDisp then LineStyle := 3 {_ _ _ _ _}
      else Write(OutFile,'LT2,',0.15*PScale:6:4,';');
    3 :if ScreenDisp then LineStyle := 4 {_____}
      else Write(OutFile,'LT4,',0.9*PScale:6:4,';');
    4 :if ScreenDisp then LineStyle := 7 {_____}
      else Write(OutFile,'LT3,',0.36*PScale:6:4,';');
    5 :if ScreenDisp then LineStyle := 6 {_____}
      else Write(OutFile,'LT6,',0.9*PScale:6:4,';');
    6 :if ScreenDisp then LineStyle := 5 {_____}
      else Write(OutFile,'LT5,',0.9*PScale:6:4,';');
    0 :if ScreenDisp then LineStyle := 2 {.....}
      else Write(OutFile,'LT1,',0.09*PScale:6:4,';');
    7,8 :if ScreenDisp then LineStyle := 8 {_____}
      else Write(OutFile,'LT;SM;');
  end;
end;

const XGlbMax = 2008;{ Max. Number of pixels in Horizontal direction}
  YGlbMax = 1394;{ Max. Number of pixels in Vertical direction }
  AxisX = 200; { Min. Number of pixels in Horizontal direction}
  AxisY = 150; { Min. Number of pixels in Vertical direction }
type st100 = string[100];
var stl : array[1..30] of st100;
  Sx,Sy : array[0..50] of real;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
X_min,X_max,Y_min,Y_max,
AxGlb,AyGlb,BxGlb,ByGlb      : real;
AX,AY,XMax,YMax,x1,y1,
fn,AXX,expX,expY            : integer;
dot                          : byte;
logs,dB_log                 : boolean;
OutName                      : String[5];

function log(x:real):real;
begin
  log := ln(x) / ln(10);
end;

function fact(n:integer):real;
var x : real;
begin
  x := 1;
  for n := n downto 1 do
    x := x*n;
  fact := x;
end;

function power(x,n:real):real; { function power(base,order) }
begin
  if x=0 then power := 0
  else
    if x>0 then power := exp(ln(x)*n)
    else
      if frac(n)=0 then
        begin
          if odd(trunc(n)) then power := -exp(ln(abs(x))*n)
          else power := exp(ln(abs(x))*n);
        end
      else
        begin
          writeln('*** error ***');
          read;
        end;
      end;
end;

function sinh(x:real):real;
begin
  sinh := (exp(x)-exp(-x))/2;
end;

function cosh(x:real):real;
begin
  cosh := (exp(x)+exp(-x))/2;
end;

procedure Add_C( C1_Re,C1_Im,C2_Re,C2_Im : real;
                 var C3 : complex);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
  C3.Re := C1_Re + C2_Re;
  C3.Im := C1_Im + C2_Im;
end;

procedure Sub_C( C1_Re,C1_Im,C2_Re,C2_Im : real;
                 var C3 : complex);
begin
  C3.Re := C1_Re - C2_Re;
  C3.Im := C1_Im - C2_Im;
end;

procedure Mul_C( C1_Re,C1_Im,C2_Re,C2_Im : real;
                 var C3 : complex);
begin
  C3.Re := C1_Re * C2_Re - C1_Im * C2_Im;
  C3.Im := C1_Im * C2_Re + C1_Re * C2_Im;
end;

procedure Div_C( C1_Re,C1_Im,C2_Re,C2_Im : real;
                 var C3 : complex);
var Dum : complex;
    E : real;
begin
  Dum.Re := C1_Re * C2_Re + C1_Im * C2_Im;
  Dum.Im := C1_Im * C2_Re - C1_Re * C2_Im;
  E := Sqr(C2_Re) + Sqr(C2_Im);
  C3.Re := Dum.Re / E;
  C3.Im := Dum.Im / E;
end;

procedure InitCom;
(* 7 6 5 4 | 3 2 1 0 *)
(* -BAUD RATE- -PARITY- STOPBIT -WORD LENGTH- *)
(* 000 - 110 X0 - NONE 0 - 1 10 - 7 BITS *)
(* 001 - 150 01 - ODD 1 - 2 11 - 8 BIT *)
(* 010 - 300 11 - EVEN *)
(* 011 - 600 *)
(* 100 - 1200 *)
(* 101 - 2400 *)
(* 110 - 4800 *)
(* 111 - 9600 *)
begin
  regs.ax := $00FF;
  regs.dx := 0;
  intr($14,regs);
end;

procedure SendCom(c: byte);
begin
  regs.ah := $01;
  regs.al := c;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
regs.dx := 0;
intr($14,regs);
end;

procedure ReceiveCom(var c: byte);
begin
regs.ah := $02;
regs.dx := 0;
intr($14,regs);
c := regs.al;
end;

function ReadyCom:boolean;
begin
regs.ah := $03;
regs.dx := 0;
intr($14,regs);
if (regs.ah and $01)=$01 then
ReadyCom := true
else
ReadyCom := false;
end;

procedure InitPlotter;
var i : integer;
ch : char;
begin
if OutName='AUX' then InitCom;
Write(OutFile,#27,'.K');
Write(OutFile,'IN;PS3;SPO;');
gotoxy(1,1);
if PScale<=5 then
begin
Write('Paper size A4 ');
Write(OutFile,'R090;');
end
else
Write('Paper size A3 ');
Write(OutFile,'PA',AXX,',0;');
for i := 1 to 3 do
Write(OutFile,'PD;PU;CI1;');
Write(OutFile,'PA',AXX+XMax+AX,',',YMax+AY,',');
for i := 1 to 3 do
Write(OutFile,'PD;PU;CI1;');
Read(kbd,ch);
end;

procedure BufferReady;
var i : integer;
begin
if OutName='AUX' then
begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
repeat
  Write(OutFile,#27,'.0');
  Read(AUX,i);
  Delay(1000);
until (i and $8) = 8;
end;
end;

procedure DefineWorld(X1W,Y2W,X2W,Y1W : real;
                     eX,eY           : integer);
var X_1,Y_1,X_2,Y_2 : integer;
    d                : real;
begin
  if ScreenDisp then
    begin
      X_1 := 3;
      Y_1 := 0;
      X_2 := XMax-1;
      Y_2 := YMax-4;
    end
  else
    begin
      X_1 := AX;
      Y_2 := AY;
      X_2 := XMax;
      Y_1 := YMax;
    end;
  if logs then
    begin
      X_min := Log(X1W*power(10,eX));
      X_max := Log(X2W*power(10,eX));
    end
  else
    begin
      X_min := X1W*power(10,eX);
      X_max := X2W*power(10,eX);
    end;
  Y_min := Y2W*power(10,eY);
  Y_max := Y1W*power(10,eY);
  BxGlb := (X_2-X_1)/(X_max-X_min);
  ByGlb := (Y_2-Y_1)/(Y_min-Y_max);
  AxGlb := X_1-X_min*BxGlb;
  AyGlb := Y_1-Y_max*ByGlb;
  expX := eX;
  expY := eY;
end;

procedure pset(x,y: real);
begin
  x1 := Round(AxGlb+BxGlb*x);
  y1 := Round(AyGlb+ByGlb*y);
  if ScreenDisp then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
  if (y1>=0) and (y1<YMax-3) then DrawPoint(x1,y1);
  dot := 0;
end
else
  Write(OutFile,'PU;PA',AXX+x1,',',y1,';PD;');
end;
```

```
procedure Line(x2,y2: integer);
var DeltaX,DeltaY,XStep,YStep,direction : integer;
begin
  if ScreenDisp then
  begin
    if x1>x2 then XStep := -1 else XStep := 1;
    if y1>y2 then YStep := -1 else YStep := 1;
    DeltaX := abs(x2-x1);
    DeltaY := abs(y2-y1);
    if DeltaX=0 then direction := -1
      else direction := 0;
    while not ((x1=x2) and (y1=y2)) do
    begin
      if (y1>=0) and (y1<YMax-3) then
      begin
        case LineStyle of
          1 :DrawPoint(x1,y1); {_____}
          2 :case dot of {.....}
            1 : DrawPoint(x1,y1);
            4 : Dot := -1;
            end;
          3 :case dot of {-----}
            1..13 : DrawPoint(x1,y1);
            16 : Dot := -1;
            end;
          4 :case dot of {____.____}
            0..35,39,43..46 : DrawPoint(x1,y1);
            47 : begin DrawPoint(x1,y1); Dot := -1; end;
            end;
          5 :case dot of {____-____}
            0..31,35..43 : DrawPoint(x1,y1);
            47 : begin DrawPoint(x1,y1); Dot := -1; end;
            end;
          6 :case dot of {____..____}
            0..33,37,41,45,46 : DrawPoint(x1,y1);
            47 : begin DrawPoint(x1,y1); Dot := -1; end;
            end;
          7 :case dot of {____-____}
            0..37,41..46 : DrawPoint(x1,y1);
            47 : begin DrawPoint(x1,y1); Dot := -1; end;
            end;
          8 :DrawPoint(x1,y1); {_____}
        end;
        dot := dot+1;
      end;
    end;
  end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end
else
  dot := 0;
  if direction<0 then
    begin
      y1 := y1+YStep;
      direction := direction+DeltaX;
    end
  else
    begin
      x1 := x1+XStep;
      direction := direction-DeltaY;
    end;
  end;
end
end
else
  Write(OutFile, 'PD',AXX+x2,',',y2,','');
end;

procedure PlotLine(x,y: real);
begin
  Line(Round(AxGlb+BxGlb*x),Round(AyGlb+ByGlb*y));
end;

procedure Alarm;
var i,j : integer;
label exitA;

procedure Play(Octave,Note,Duration: integer);
var Frequency : real;
    i : integer;
begin
  Frequency := 32.625;
  for i := 1 to Octave do
    Frequency := Frequency * 2;
  for i := 1 to Note - 1 do
    Frequency := Frequency * 1.059463094;
  Sound(Round(Frequency));
  Delay(Duration);
  NoSound;
end;

begin
  if ScreenDisp then
    begin
      for i := 1 to 2 do
        begin
          Play(4,9,80);
          Play(4,2,80);
        end;
      for i := 3 to 5 do
        begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Play(4,i,80);
    Play(i,4,80);
end;
end
else
begin
  for j := 1 to 5 do
  begin
    for i := 1 to 8 do
    begin
      Play(4,9,80);
      Play(4,2,80);
    end;
    Delay(500);
    regs.ah := 6;
    regs.dl := $ff;
    MsDos(Regs);
    if (regs.al=13) or (regs.al=27) then goto exitA;
  end;
exitA:
  for i := 1 to 8 do
  begin
    Play(4,i,90);
    Play(i,4,90);
  end;
end;
end;
```

```
procedure Axis(ScaleX,ScaleY:integer; ScX,ScY:integer;
  pos,ws:byte; st,Xst,Yst:st100);
```

```
label exitS;
var xw1,xw2,yw1,yw2,
  i,j,k,xi,yi,x2 : integer;
  Xx,Yy,p1,p2    : real;
  sk              : boolean;
```

```
function wxy(x,y:real):boolean;
```

```
begin
  wxy := true;
  if pos<>0 then
  begin
    x := Round(AxGlb+BxGlb*x);
    y := Round(AyGlb+ByGlb*y);
    if x>xw1 then
      if x<xw2 then
        if y>yw1 then
          if y<yw2 then
            wxy := false;
  end;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
  xw1 := 30000;
  xw2 := -30000;
  yw1 := 30000;
  yw2 := -30000;
  SetLineStyle(8);
  Xx := X_max-X_min;
  Yy := Y_max-Y_min;
  if ScreenDisp then
    begin
      pset(X_min,Y_min);
      PlotLine(X_max,Y_min);
      pset(X_min,Y_min);
      PlotLine(X_min,Y_max);
    end
  else
    begin
      Write(OutFile,'SP8;VS5;CA8;SI',0.1+0.016*PScale:6:4,',',
        0.14+0.026*PScale:6:4,','');
      pset(X_min,Y_min);
      Write(OutFile,'PD',AXX+X1+Xmax-AX*0.8:6:4,',',Y1,','PU;');
      Write(OutFile,'PA',AXX+X1+Xmax-AX*0.65:6:4,',',Y1,','L02;');
      Write(OutFile,'LB',Xst,^C);
      pset(X_min,Y_min);
      Write(OutFile,'PD',AXX+X1,',',Y1+Ymax-AY*0.8:6:4,','PU;');
      Write(OutFile,'PA',AXX+X1,',',Y1+Ymax-AY*0.55:6:4,','L04;');
      Write(OutFile,'LB',Yst,^C);
      Write(OutFile,'SI',0.08+0.014*PScale:6:4,',',
        0.12+0.023*PScale:6:4,','SL0.2;');
      Write(OutFile,'PU;PA',AXX+X1+(Xmax-AX) shr 1,',',
        Y1-AY*0.7:6:4,','L06;');
      Write(OutFile,'LB',st,^C);
    end;
  for j := 0 to ScaleX do
    begin
      Sx[j] := X_min+Xx*j/ScaleX;
      pset(Sx[j],Y_min);
      if ScreenDisp then
        for i := 1 to 3 do DrawPoint(x1,y1+i)
      else
        begin
          Write(OutFile,'PD',AXX+x1,',',y1-AX*0.08:6:4,','PU;');
          Write(OutFile,'PA',AXX+x1,',',y1-AX*0.15:6:4,','L06;');
          if logs then
            begin
              if frac(abs(power(10,Sx[j])*power(10,-expX))+
                0.001)<0.01 then
                Write(OutFile,'LB',power(10,Sx[j])*power(10,-expX):1:0,^C)
              else
                if frac(abs(power(10,Sx[j])*power(10,-expX))+0.01)<0.1 then
                  Write(OutFile,'LB',power(10,Sx[j])*power(10,-expX):4:2,^C)
                end
            end
          end
        end
    end
  end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    else
      Write(OutFile,'LB',power(10,Sx[j])*power(10,-expX):3:1,^C);
    end
  else
    if frac(abs(Sx[j]*power(10,-expX))+0.001)<0.01 then
      Write(OutFile,'LB',Sx[j]*power(10,-expX):1:0,^C)
    else
      if frac(abs(Sx[j]*power(10,-expX))+0.01)<0.1 then
        Write(OutFile,'LB',Sx[j]*power(10,-expX):4:2,^C)
      else
        Write(OutFile,'LB',Sx[j]*power(10,-expX):3:1,^C);
      end;
    end;
  end;
end;

for j := 0 to ScaleY do
  begin
    Sy[j] := Y_min+Yy*j/ScaleY;
    pset(X_min,Sy[j]);
    if ScreenDisp then
      for i := 1 to 3 do DrawPoint(x1-i,y1)
    else
      begin
        Write(OutFile,'PD',AXX+x1-AX*0.08:6:4,',',y1,';PU;');
        Write(OutFile,'PA',AXX+x1-AX*0.15:6:4,',',y1,';LO8;');
        if frac(abs(Sy[j]*power(10,-expY))+0.001)<0.01 then
          Write(OutFile,'LB',Sy[j]*power(10,-expY):1:0,^C)
        else
          if frac(abs(Sy[j]*power(10,-expY))+0.01)<0.1 then
            Write(OutFile,'LB',Sy[j]*power(10,-expY):4:2,^C)
          else
            Write(OutFile,'LB',Sy[j]*power(10,-expY):3:1,^C);
          end;
        end;
      end;
    end;
  end;

if not ScreenDisp then
  begin
    Write(OutFile,'PU;');
    if pos<>0 then
      for i:=1 to fn do
        begin
          SetLineStyle(i);
          Write(OutFile,'SP',i,',';');
          case pos of
            1,5: begin
              x1 := AX*2;
              y1 := Trunc(YMax-AY/3-AY/3*i);
              x2 := Trunc(AX*3.5);
            end;
            2,6: begin
              x1 := (XMax shr 1)+AX;
              y1 := Trunc(YMax-AY/3-AY/3*i);
              x2 := (XMax shr 1)+Trunc(AX*2.5);
            end;
          end;
        end;
      end;
    end;
  end;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;
3,7: begin
  x1 := AX*2;
  y1 := Trunc((YMax shr 1)-AY/2-AY/3*i);
  x2 := Trunc(AX*3.5);
end;
4,8: begin
  x1 := (XMax shr 1)+AX;
  y1 := Trunc((YMax shr 1)-AY/2-AY/3*i);
  x2 := (XMax shr 1)+Trunc(AX*2.5);
end;
end;
if xw1>x1 then xw1 := x1;
if xw2<x2 then xw2 := x2;
if yw1>y1 then yw1 := y1;
if yw2<y1 then yw2 := y1;
Write(OutFile,'PA',AXX+x1,',',y1,',';PD;');
Line(x2,y1);
Write(OutFile,'SI',0.08+0.014*PScale:6:4,',',
0.12+0.023*PScale:6:4,',';SLO.2;SM;');
Write(OutFile,'PU;PR',(AX shr 3),',',0,'LO2;LB',stl[i],^C);
end;
xw1 := xw1 - (AX shr 3);
xw2 := Round( xw2 + (AX shr 2) +
(0.08+0.014*PScale)*ws*400*1.3 );
yw1 := yw1 - (AY shr 2);
yw2 := yw2 + (AY shr 2);
Write(OutFile,'VS40;');
SetLineStyle(8);
end;

if (ScX<>0) and (ScY<>0) then
begin
  if not ScreenDisp then Write(OutFile,'SP7;');
  xi := ScaleX*ScX;
  yi := ScaleY*ScY;
  if xi>yi then yi := yi*Round(xi/yi)
  else xi := xi*Round(yi/xi);
  xi := xi*round(PScale * 20/xi);
  yi := yi*round(PScale * 12/yi);
  p2 := Yy/yi;
  if logs then
  for j := 1 to ScaleX do
  for k := 1 to ScX do
  begin
    regs.ah := 6;
    regs.dl := $ff;
    MsDos(Regs);
    if (regs.al=27) then
    begin
      if not ScreenDisp then Write(OutFile,#27,'.K');
      goto exitS;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;
sk := false;
p1 := X_min+Xx*j/ScaleX-Xx/ScaleX*(power(10,-k/ScX)-0.1)/0.9;
for i := 0 to yi do
  if ScreenDisp then
    pset(p1,Y_min+p2*i)
  else
    if pos<=4 then
      begin
        if wxy(p1,Y_min+p2*i) then pset(p1,Y_min+p2*i);
      end
    else
      begin
        if wxy(p1,Y_min+p2*i) then
          begin
            if sk then
              Write(OutFile,'PD')
            else
              begin
                Write(OutFile,'PU;PA');
                sk := true;
              end;
            end
          else
            begin
              Write(OutFile,'PU;PA');
              sk := false;
            end;
          Write(OutFile,AXX+Round(AxGlb+BxGlb*p1),',,');
          Write(OutFile,Round(AyGlb+ByGlb*(Y_min+p2*i)),',,');
        end;
      end
    end
  else
    for j := 1 to ScaleX*ScX do
      begin
        regs.ah := 6;
        regs.dl := $ff;
        MsDos(Regs);
        if (regs.al=27) then
          begin
            if not ScreenDisp then Write(OutFile,#27,'.K');
            goto exitS;
          end;
        sk := false;
        p1 := X_min+Xx*j/(ScaleX*ScX);
        for i := 0 to yi do
          if ScreenDisp then
            pset(p1,Y_min+p2*i)
          else
            if pos<=4 then
              begin
                if wxy(p1,Y_min+p2*i) then pset(p1,Y_min+p2*i);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end
else
begin
  if wxy(p1,Y_min+p2*i) then
  begin
    if sk then
      Write(OutFile,'PD')
    else
      begin
        Write(OutFile,'PU;PA');
        sk := true;
      end;
    end
  else
  begin
    Write(OutFile,'PU;PA');
    sk := false;
  end;
  Write(OutFile,AXX+Round(AxGlb+BxGlb*p1),',');
  Write(OutFile,Round(AyGlb+ByGlb*(Y_min+p2*i),',');
end;
end;
p1 := Xx/xi;
for j := 1 to ScaleY*ScY do
begin
  regs.ah := 6;
  regs.dl := $ff;
  MsDos(Regs);
  if (regs.al=27) then
  begin
    if not ScreenDisp then Write(OutFile,#27,'.K');
    goto exitS;
  end;
  sk := false;
  p2 := Y_min+Yy*j/(ScaleY*ScY);
  for i := 0 to xi do
  if ScreenDisp then
    pset(X_min+p1*i,p2)
  else
  if pos<=4 then
  begin
    if wxy(X_min+p1*i,p2) then pset(X_min+p1*i,p2);
  end
  else
  begin
    if wxy(X_min+p1*i,p2) then
    begin
      if sk then
        Write(OutFile,'PD')
      else
        begin
          Write(OutFile,'PU;PA');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        sk := true;
      end;
    end
  else
    begin
      Write(OutFile,'PU;PA');
      sk := false;
    end;
    Write(OutFile,AXX+Round(AxGlb+BxGlb*(X_min+p1*i)),',');
    Write(OutFile,Round(AyGlb+ByGlb*p2),',');
  end;
end;
end;
end;
exitS:

SwapScreen;
gotoxy(1,1);
Write('ScaleX  ');
if expX=0 then Write(':')
  else Write('( *1e',expX,' )  ');
for i := 0 to ScaleX do
  begin
    if (i mod 7)=0 then writeln;
    if logs then Write(power(10,Sx[i])*power(10,-expX):9:2,',')
      else Write(Sx[i]*power(10,-expX):9:2,',');
    end;
  writeln; writeln;

  Write('ScaleY  ');
  if expY=0 then Write(':')
    else Write('( *1e',expY,' )  ');
  for i := 0 to ScaleY do
    begin
      if (i mod 7)=0 then writeln;
      Write(Sy[i]*power(10,-expY):9:2,',');
    end;
  writeln; writeln; writeln;
  Sound(1000);
  Delay(200);
  NoSound;

  if ScreenDisp then
    begin
      Delay(1000);
      SwapScreen;
    end
  else
    Write(OutFile,'SP0;');
end;
```

program other;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
const TNArraySize = 30;
type TNcomplex    = record
                    Re,Im : real;
                    end;
TNIntVector      = array[0..TNArraySize] of integer;
TNCompVector     = array[0..TNArraySize] of TNcomplex;
var DFile : text;

function power(x,n:real):real;    { function power(base,order) }
begin
  if x=0 then power := 0
  else
    if x>0 then power := exp(ln(x)*n)
    else
      if frac(n)=0 then
        begin
          if odd(trunc(n)) then power := -exp(ln(abs(x))*n)
          else power := exp(ln(abs(x))*n);
        end
      else
        begin
          writeln('*** error ***');
          read;
        end;
      end;
end;

procedure Sort(var ary: TNCompVector; n: integer);

procedure swapping(var first,second: TNcomplex);
var tem : TNcomplex;
begin
  tem := first;
  first := second;
  second := tem;
end;

var d,i : integer;
    Done : boolean;
begin
  d := n;
  while d>1 do
    begin
      d := d div 2;
      repeat
        done := true;
        for i := 1 to (n-d) do
          if ary[i].Re>ary[i+d].Re then
            begin
              swapping(ary[i],ary[i+d]);
              done := false;
            end;
        until done;
    end;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;
end;

procedure Laguerre(var Degree      : integer;
                  var Poly        : TNCompVector;
                  InitGuess      : TNcomplex;
                  Tol             : real;
                  MaxIter        : integer;
                  var NumRoots    : integer;
                  var Roots       : TNCompVector;
                  var yRoots      : TNCompVector;
                  var Iter        : TNIntVector;
                  var Error       : byte);

const TNNearlyZero = 1E-10;
type TNquadratic = record
    A,B,C : real;
end;

var AddIter      : integer;
    InitDegree   : integer;
    InitPoly     : TNCompVector;
    GuessRoot    : TNcomplex;

procedure Conjugate(var C1,C2 : TNcomplex);
begin
    C2.Re := C1.Re;
    C2.Im := -C1.Im;
end;

function Modulus(var C1 : TNcomplex) : real;
begin
    Modulus := Sqrt(Sqr(C1.Re) + Sqr(C1.Im));
end;

procedure Add(var C1,C2,C3 : TNcomplex);
begin
    C3.Re := C1.Re + C2.Re;
    C3.Im := C1.Im + C2.Im;
end;

procedure Sub(var C1,C2,C3 : TNcomplex);
begin
    C3.Re := C1.Re - C2.Re;
    C3.Im := C1.Im - C2.Im;
end;

procedure Mult(var C1,C2,C3 : TNcomplex);
begin
    C3.Re := C1.Re * C2.Re - C1.Im * C2.Im;
    C3.Im := C1.Im * C2.Re + C1.Re * C2.Im;
end;

procedure Divide(var C1,C2,C3 : TNcomplex);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
var Dum1,Dum2 : TNcomplex;
    E : real;
begin
    Conjugate(C2,Dum1);
    Mult(C1,Dum1,Dum2);
    E := Sqr(Modulus(C2));
    C3.Re := Dum2.Re / E;
    C3.Im := Dum2.Im / E;
end;

procedure SquareRoot(var C1,C2 : TNcomplex);
const TNNearlyZero = 1E-15;
var R,Theta : real;
begin
    R := Sqrt(Sqr(C1.Re) + Sqr(C1.Im));
    if ABS(C1.Re) < TNNearlyZero then
        begin
            if C1.Im < 0 then Theta := -Pi / 2
            else Theta := -Pi / 2;
        end
    else
        if C1.Re < 0 then Theta := ArcTan(C1.Im / C1.Re) + Pi
        else Theta := ArcTan(C1.Im / C1.Re);
    C2.Re := Sqrt(R) * Cos(Theta / 2);
    C2.Im := Sqrt(R) * Sin(Theta / 2);
end;

procedure InitAndTest(var Degree : integer;
    var Poly : TNCompVector;
    Tol : real;
    MaxIter : integer;
    InitGuess : TNcomplex;
    var NumRoots : integer;
    var Roots : TNCompVector;
    var yRoots : TNCompVector;
    var Iter : TNIntVector;
    var GuessRoot : TNcomplex;
    var InitDegree : integer;
    var InitPoly : TNCompVector;
    var Error : byte);

var Term : integer;
begin
    Error := 0;
    if Degree <= 0 then Error := 2;
    if Tol <= 0 then Error := 3;
    if MaxIter < 0 then Error := 4;

    if Error = 0 then
        begin
            NumRoots := 0;
            GuessRoot := InitGuess;
            InitDegree := Degree;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
InitPoly := Poly;
while (Degree > 0) and (Modulus(Poly[Degree]) < TNNearlyZero) do
  Degree := Pred(Degree);
while (Modulus(Poly[0]) = 0) and (Degree > 0) do
  begin
    NumRoots := Succ(NumRoots);
    Roots[NumRoots].Re := 0;
    Roots[NumRoots].Im := 0;
    yRoots[NumRoots].Re := 0;
    yRoots[NumRoots].Im := 0;
    Iter[NumRoots] := 0;
    Degree := Pred(Degree);
    for Term := 0 to Degree do
      Poly[Term] := Poly[Term + 1];
    end;
  end;
end;

procedure FindOneRoot(Degree : integer;
  Poly : TNCompVector;
  GuessRoot : TNcomplex;
  Tol : real;
  MaxIter : integer;
  var Root : TNcomplex;
  var yValue : TNcomplex;
  var Iter : integer;
  var Error : byte);

var Found : boolean;
  Dif : TNcomplex;
  yPrime,yDoublePrime : TNcomplex;

procedure EvaluatePoly(Degree : integer;
  Poly : TNCompVector;
  X : TNcomplex;
  var yValue : TNcomplex;
  var yPrime : TNcomplex;
  var yDoublePrime : TNcomplex);

var Loop : integer;
  Dummy,yDPdummy : TNcomplex;
  Deriv,Deriv2 : TNCompVector;
begin
  Deriv[Degree] := Poly[Degree];
  for Loop := Degree - 1 downto 0 do
    begin
      Mult(Deriv[Loop + 1],X,Dummy);
      Add(Dummy,Poly[Loop],Deriv[Loop]);
    end;
  yValue := Deriv[0];

  Deriv2[Degree] := Deriv[Degree];
  for Loop := Degree - 1 downto 1 do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
  Mult(Deriv2[Loop + 1],X,Dummy);
  Add(Dummy,Deriv[Loop],Deriv2[Loop]);
end;
yPrime := Deriv2[1];

yDPdummy := Deriv2[Degree];
for Loop := Degree - 1 downto 2 do
begin
  Mult(yDPdummy,X,Dummy);
  Add(Dummy,Deriv2[Loop],yDPdummy);
end;
yDoublePrime.Re := 2 * yDPdummy.Re;
yDoublePrime.Im := 2 * yDPdummy.Im;
end;

procedure ConstructDifference(Degree : integer;
  yValue : TNcomplex;
  yPrime : TNcomplex;
  yDoublePrime : TNcomplex;
  var Dif : TNcomplex);
var yPrimeSQR,yTimesyDPrime,Sum,SRoot,
  Numer1,Numer2,Numer,Denom : TNcomplex;
begin
  Mult(yPrime,yPrime,yPrimeSQR);
  yPrimeSQR.Re := Sqr(Degree - 1) * yPrimeSQR.Re;
  yPrimeSQR.Im := Sqr(Degree - 1) * yPrimeSQR.Im;
  Mult(yValue,yDoublePrime,yTimesyDPrime);
  yTimesyDPrime.Re := (Degree - 1) * Degree * yTimesyDPrime.Re;
  yTimesyDPrime.Im := (Degree - 1) * Degree * yTimesyDPrime.Im;
  Sub(yPrimeSQR,yTimesyDPrime,Sum);
  SquareRoot(Sum,SRoot);
  Add(yPrime,SRoot,Numer1);
  Sub(yPrime,SRoot,Numer2);
  if Modulus(Numer1) > Modulus(Numer2) then Numer := Numer1
  else Numer := Numer2;
  Denom.Re := Degree * yValue.Re;
  Denom.Im := Degree * yValue.Im;
  if Modulus(Numer) < TNNearlyZero then
  begin
    Dif.Re := 0;
    Dif.Im := 0;
  end
  else
    Divide(Denom,Numer,Dif);
end;

function TestForRoot(X,Dif,Y,Tol : real) : boolean;
begin
  TestForRoot := (ABS(Y) <= TNNearlyZero) or
    (ABS(Dif) < ABS(X * Tol))
end; { procedure TestForRoot }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin { procedure FindOneRoot }
  Root := GuessRoot;
  Found := false;
  Iter := 0;
  EvaluatePoly(Degree, Poly, Root, yValue, yPrime, yDoublePrime);
  while (Iter < MaxIter) and not(Found) do
  begin
    Iter := Succ(Iter);
    ConstructDifference(Degree, yValue, yPrime, yDoublePrime, Dif);
    Sub(Root, Dif, Root);
    EvaluatePoly(Degree, Poly, Root, yValue, yPrime, yDoublePrime);
    Found := TestForRoot(Modulus(Root), Modulus(Dif),
      Modulus(yValue), Tol);
  end;
  if not(Found) then Error := 1;
end;

procedure ReducePoly(var Degree : integer;
  var Poly : TNCompVector;
  var Root : TNcomplex);
var Term : integer;
  NewPoly : TNCompVector;
  Dummy : TNcomplex;
begin
  NewPoly[Degree - 1] := Poly[Degree];
  for Term := Degree - 1 downto 1 do
  begin
    Mult(NewPoly[Term], Root, Dummy);
    Add(Dummy, Poly[Term], NewPoly[Term - 1]);
  end;
  Degree := Pred(Degree);
  Poly := NewPoly;
end;

begin
  InitAndTest(Degree, Poly, Tol, MaxIter, InitGuess, NumRoots, Roots,
    yRoots, Iter, GuessRoot, InitDegree, InitPoly, Error);
  while (Degree > 0) and (Error = 0) do
  begin
    FindOneRoot(Degree, Poly, GuessRoot, Tol, MaxIter,
      Roots[NumRoots + 1], yRoots[NumRoots + 1],
      Iter[NumRoots + 1], Error);
    if Error = 0 then
    begin
      FindOneRoot(InitDegree, InitPoly, Roots[NumRoots + 1],
        Tol, MaxIter, Roots[NumRoots + 1],
        yRoots[NumRoots + 1], AddIter, Error);
      Iter[NumRoots + 1] := Iter[NumRoots + 1] + AddIter;
      NumRoots := Succ(NumRoots);
      ReducePoly(Degree, Poly, Roots[NumRoots]);
    end;
  end;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    GuessRoot := Roots[NumRoots];
end;
end;

procedure Initial(var Poly : TNCompVector);
begin
    FillChar(Poly,SizeOf(Poly),0);
end;

procedure UserInput(var Degree : integer;
                    var Poly : TNCompVector;
                    var Guess : TNcomplex;
                    var Tol : real;
                    var MaxIter : integer);

var Ch : char;

procedure GetCoefficientsFromKeyboard(var Degree : integer;
                                       var Poly : TNCompVector);

var Term : integer;
begin
    Write('Degree of the polynomial = ');
    Readln(Degree);
    Writeln;
    for Term := 0 to Degree do
    begin
        Write(' Poly[' ,Term:2,'] Re : ',Poly[Term].Re,' = ');
        Readln(Poly[Term].Re);
        Write(' Im : ',Poly[Term].Im,' = ');
        Readln(Poly[Term].Im);
    end;
end;

begin { procedure UserInput }
    GetCoefficientsFromKeyboard(Degree,Poly);
    Guess.Re := 1;
    Guess.Im := 1;
    Tol := 1E-10;
    MaxIter := 1000;
end;

procedure Results(NumRoots : integer;
                  Answer : TNCompVector;
                  MaxIter : integer;
                  Error : byte);

var Term : integer;
begin
    Writeln;
    Sound(3000);
    Delay(2);
    NoSound;
    case Error of
        1 : Writeln('This will take more than ',MaxIter,' iterations.');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
2 : Writeln('The degree of the polynomial must be greater
      than zero.');
```

```
3 : Writeln('The tolerance must be greater than zero.');
```

```
4 : Writeln('The maximum number of iterations must be greater
      than zero.');
```

```
end;
if Error <= 1 then
begin
  for Term := 1 to NumRoots do
  begin
    Write(' Root ',Term:2,'      Re = ',Answer[Term].Re:15:8);
    Writeln('      Im = ',Answer[Term].Im:15:8);
  end;
  read;
  end;
end;

procedure polmul(Poly1      : TNCompVector;
                 Poly2      : TNCompVector;
                 InitDegree1 : integer;
                 InitDegree2 : integer;
                 var Poly    : TNCompVector;
                 var Degree  : integer);
var i,k,j : integer;
begin
  Degree := InitDegree1+InitDegree2;
  for i := 0 to Degree do
  begin
    poly[i].Re := 0; poly[i].Im := 0;
    for k := 0 to InitDegree2 do
    begin
      j := i-k;
      if (j>=0) and(j<=InitDegree1) then
      begin
        poly[i].Re := poly[i].Re+
          poly1[j].Re*poly2[k].Re-poly1[j].Im*poly2[k].Im;
        poly[i].Im := poly[i].Im
          +poly1[j].Re*poly2[k].Im+poly1[j].Im*poly2[k].Re;
      end;
    end;
  end;
end;

procedure ResultPoly(NumRoots  : integer;
                     Answer    : TNCompVector);
var a,b,c      : TNCompVector;
    i,j,k,n    : integer;
begin
  writeln;
  Sound(3000);
  Delay(2);
  NoSound;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
writeln('Result:');
writeln;
j := 0;
n := -1;
for i := 1 to NumRoots do
  if Answer[i].Re < 0 then
    begin
      j := j+1;
      if j=1 then
        begin
          n := 1;
          a[1].Re := 1;
          a[1].Im := 0;
          a[0].Re := -Answer[i].Re;
          a[0].Im := -Answer[i].Im;
        end
      else
        begin
          b[1].Re := 1;
          b[1].Im := 0;
          b[0].Re := -Answer[i].Re;
          b[0].Im := -Answer[i].Im;
          polmul(a,b,n,1,c,n);
          for k := 0 to n do
            begin
              a[k].Re := c[k].Re; a[k].Im := c[k].Im;
            end;
          end;
        end;
      for i := 0 to n do
        begin
          write(' Poly[';i:2;'] Re = ',a[i].Re:15:8);
          writeln(' Im = ',a[i].Im:15:8);
        end;
      end;
end;

procedure MulPoly;
var a,b,c : TNCompVector;
    n,n1,n2,i,j : integer;
begin
  n := 0;
  n1 := 0;
  Initial(a);
  Write('Number of polynomial = '); Readln(n);
  Writeln;
  Write('Degree of the polynomial 1 = '); readln(n1);
  for i := 0 to n1 do
    begin
      Write(' Poly[';i:2;'] Re = '); Readln(a[i].Re);
      Write(' Im = '); Readln(a[i].Im);
    end;
  writeln;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
for j := 2 to n do
begin
n2 := 0;
Initial(b);
Write('Degree of the polynomial ',j,' = '); readln(n2);
for i := 0 to n2 do
begin
Write(' Poly[' ,i:2,' ] Re = '); Readln(b[i].Re);
Write(' Im = '); Readln(b[i].Im);
end;
polmul(a,b,n1,n2,c,n1);
writeln;
for i := 0 to n1 do
begin
a[i].Re := c[i].Re; a[i].Im := c[i].Im;
end;
end;
Sound(3000);
Delay(2);
NoSound;
writeln('Result:');
writeln;
for i := 0 to n1 do
begin
write(' Poly[' ,i:2,' ] Re = ',a[i].Re:15:8);
writeln(' Im = ',a[i].Im:15:8);
end;
end;

var Guess : TNcomplex;
InitDegree, Degree : integer;
InitPoly, Poly,
ChevPoly, UltraPoly : TNCompVector;
Tol : real;
Iter : TNIntVector;
MaxIter : integer;
NumRoots : integer;
Answer : TNCompVector;
yAnswer : TNCompVector;
Error : byte;
ch : char;
```

```
procedure Select;
begin
ClrScr;
Assign(DFile, 'SOLUTION');
Rewrite(DFile);
case ch of
'1' : MulPoly;
'2' : UserInput(InitDegree, InitPoly, Guess, Tol, MaxIter);
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if ch in ['1'..'2'] then
begin
Degree := InitDegree;
Poly := InitPoly;
Laguerre(Degree,Poly,Guess,Tol,MaxIter,
          NumRoots,Answer,yAnswer,Iter,Error);
Sort(Answer,Numroots);
Results(NumRoots,Answer,MaxIter,Error);
if Error <= 1 then ResultPoly(NumRoots,Answer);
end;
read;
Close(DFile);
end;

begin
InitDegree := 0;
Initial(InitPoly);
Initial(ChevPoly);
Initial(UltraPoly);
repeat
ClrScr;
gotoxy(24,9); write('1) Multiple Polynomial');
gotoxy(24,10); write('2) Roots & Polynomial');
gotoxy(24,11); write('3) ...');
gotoxy(24,12); write('4) ...');
gotoxy(24,13); write('5) ...');
gotoxy(23,18); write('ESC Quit');
gotoxy(27,20); write('Select No. ');
repeat
gotoxy(38,20); read(kbd,ch); write(ch);
until ch in [#27,'1'..'5'];
if ch<>#27 then Select;
until ch=#27;
end.

procedure TargetF(w : real);
var k,al,t,t1,t2,r1,r2,
    c1,c2,Qap,w0ap,a,b : real;
    i,j,m,n : integer;
    d1,d2,d3 : complex;
begin
if freq then
case xs of
3 : w := w/1e3;
6 : w := w/1e6;
9 : w := w/1e9;
end;
case fi of
1: div_c( 1, 0 , -2.3122*w*w+1 , -1.7499*w*w*w+2.2775*w , data );
2: div_c( 1, 0 , -2.4141*w*w+1 , -2.8746*w*w*w+2.8885*w , data );
3: div_c( 1, 0 , -2.1414*w*w+1 , -1.2497*w*w*w+2.0847*w , data );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
4: div_c( 1, 0 , -2.2806*w*w+1 , -1.6251*w*w*w+2.2252*w , data );
5: div_c( 1, 0 , -2*w*w+1 , -w*w*w+2*w , data );
end;
end;
```

```
function TargetT(w : real):real;
var k,al,t,t1,t2,r1,r2,
    c1,c2,Qap,w0ap,a,b : real;
    i,j,m,n : integer;
    d1,d2,d3 : complex;
begin
if freq then
case xs of
3 : w := w/1e3;
6 : w := w/1e6;
9 : w := w/1e9;
end;
case fi of
1: TargetT := 0.661*exp(-0.661*w)
- 0.66*exp(-0.33*w)*cos(0.869*w)
+ 0.25*exp(-0.33*w)*sin(0.869*w);
2: TargetT := 0.420*exp(-0.420*w)
- 0.42*exp(-0.21*w)*cos(0.886*w)
+ 0.1 *exp(-0.21*w)*sin(0.886*w);
3: TargetT := 0.857*exp(-0.857*w)
- 0.856*exp(-0.428*w)*cos(0.866*w)
+ 0.424*exp(-0.428*w)*sin(0.866*w);
4: TargetT := 0.702*exp(-0.702*w)
- 0.702*exp(-0.351*w)*cos(0.868*w)
+ 0.284*exp(-0.351*w)*sin(0.868*w);
5: TargetT := exp(-w) - exp(-0.5*w)*cos(0.866*w)
+ 0.578*exp(-0.5*w)*sin(0.866*w);
end;
end;
```

```
function TargetU(w : real):real;
var k,al,t,t1,t2,r1,r2,
    c1,c2,Qap,w0ap,a,b : real;
    i,j,m,n : integer;
    d1,d2,d3 : complex;
begin
if freq then
case xs of
3 : w := w/1e3;
6 : w := w/1e6;
9 : w := w/1e9;
end;
case fi of
1: TargetU := 1-exp(-0.661*w)
- 0.760*exp(-0.33*w)*sin(0.869*w);
2: TargetU := 1-exp(-0.420*w)
- 0.474*exp(-0.21*w)*sin(0.886*w);
end;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
3: TargetU := 1-exp(-0.857*w)
   - 0.988*exp(-0.428*w)*sin(0.866*w);
4: TargetU := 1-exp(-0.702*w)
   - 0.808*exp(-0.351*w)*sin(0.868*w);
5: TargetU := 1-exp(-w) - 1.154*exp(-0.5*w)*sin(0.866*w);
end;
end;

procedure Fre_3db(LeftEndPoint : real;
                  RightEndPoint : real;
                  var Answer      : real;
                  var Error       : byte);

const NearlyZero = 1E-10;
var Tol,yLeft,yRight,MidPoint,yMidPoint : real;
    Iter, MaxIter : integer;
    Found : boolean;

function TestForRoot(X, OldX, Y, Tol : real) : boolean;
begin
    TestForRoot := (ABS(Y) <= NearlyZero) or
                   (ABS(X - OldX) < ABS(OldX * Tol));
end;

begin
    Maxiter := 500;
    Tol := 1E-10;
    Error := 0;
    Found := false;
    TargetF(LeftEndpoint);
    yLeft := sqrt( sqr(data.re) + sqr(data.im) ) - 1/sqrt(2);
    TargetF(RightEndpoint);
    yRight := sqrt( sqr(data.re) + sqr(data.im) ) - 1/sqrt(2);
    if ABS(yLeft) <= NearlyZero then
    begin
        Answer := LeftEndpoint;
        Found := true;
    end;
    if ABS(yRight) <= NearlyZero then
    begin
        Answer := RightEndpoint;
        Found := true;
    end;
    if not Found then { Test for errors }
    begin
        if yLeft * yRight > 0 then
            Error := 2;
        if Tol <= 0 then
            Error := 3;
        if MaxIter < 0 then
            Error := 4;
    end;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (Error = 0) and (Found = false) then
begin
  Iter := 0;
  TargetF(LeftEndPoint);
  yLeft := sqrt( sqrt(data.re) + sqrt(data.im) ) - 1/sqrt(2);
  while not(Found) and (Iter < MaxIter) do
  begin
    Iter := Succ(Iter);
    MidPoint := (LeftEndpoint + RightEndpoint) / 2;
    TargetF(MidPoint);
    yMidPoint := sqrt( sqrt(data.re) + sqrt(data.im) ) - 1/sqrt(2);
    Found := TestForRoot(MidPoint, LeftEndpoint, yMidPoint, Tol);
    if (yLeft * yMidPoint) < 0 then
      RightEndpoint := MidPoint
    else
      begin
        LeftEndpoint := MidPoint;
        yLeft := yMidPoint;
      end;
    end;
    Answer := MidPoint;
    if Iter >= MaxIter then
      Error := 1;
    end;
  end;
end;

procedure Nyquist(w : real);
var t
  : real;
  d1,d2
  : complex;
begin
  t := w;
  case fi of
  1..2: begin (*---- U3 1 ----*)
    Data.Re := -2.3122*w*w+1;
    Data.Im := -1.7499*w*w*w+2.2775*w;
  end;
  3..4: begin (*--- U3 -0.2 ---*)
    Data.Re := -2.4141*w*w+1;
    Data.Im := -2.8746*w*w*w+2.8885*w;
  end;
  5..6: begin (*---- U1B2 1 ----*)
    Data.Re := -2.1414*w*w+1;
    Data.Im := -1.2497*w*w*w+2.0847*w;
  end;
  7..8: begin (*-- U1B2 -0.2 --*)
    Data.Re := -2.2806*w*w+1;
    Data.Im := -1.6251*w*w*w+2.2252*w;
  end;
  9..10: begin (*----- B3 -----*)
    Data.Re := -2*w*w+1;
    Data.Im := -w*w*w+2*w;
  end;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;
end;

procedure Stable(Guess1 : real;
                 Guess2 : real;
                 var Root : real;
                 var Error : byte);

const NearlyZero = 1E-10;
var Found : boolean;
    OldX,OldY,X,Y,NewX,NewY,Tol : real;
    MaxIter,Iter : integer;

function TestForRoot(X, OldX, y, Tol : real) : boolean;
begin
    TestForRoot := (ABS(y) <= NearlyZero) or
                   (ABS(X - OldX) < ABS(OldX*Tol))
end;

begin
    MaxIter := 500;
    Tol := 1E-10;
    Found := false;
    Iter := 0;
    Error := 0;
    OldX := Guess1;
    X := Guess2;
    Nyquist(OldX);
    OldY := data.im;
    Nyquist(X);
    Y := data.im;
    if ABS(OldY) <= NearlyZero then { OldX is the root }
    begin
        X := OldX;
        Y := OldY;
        Found := true;
    end
    else
        if ABS(Y) <= NearlyZero then
            Found := true { X is the root }
        else
            if ABS(OldY - Y) <= NearlyZero then
                Error := 2; { Slope of line is zero; no intercept }
            else
                while not(Found) and (Error = 0) and (Iter<MaxIter) do
                begin
                    Iter := Succ(Iter);
                    NewX := X - Y * (X - OldX) / (Y - OldY);
                    Nyquist(NewX);
                    NewY := data.im;
                    Found := TestForRoot(NewX, OldX, NewY, Tol);
                    OldX := X;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
OldY := Y;
X := NewX;
Y := NewY;
end;
Root := X;
if not(Found) and (Error = 0) and (Iter >= MaxIter) then
  Error := 1;
end;

function Differentiate(x: real):real;
const Tolerance = 1e-7;
type vector = array[1..100] of real;
var Term,Iter,TwoToTheIterMinus2,Extrap : integer;
    DeltaX,FourToTheExtrapMinus1 : real;
    OldEstimate,NewEstimate : vector;

function EvaluateFirstDeriv(X : real;
    DeltaX : real):real;
var LeftPoint,RightPoint : real;
begin
  TargetF(X - DeltaX);
  LeftPoint := ArcTan(data.im / data.re);
  TargetF(X + DeltaX);
  RightPoint := ArcTan(data.im / data.re);
  EvaluateFirstDeriv := (RightPoint - LeftPoint)/(2 * DeltaX);
end; { function EvaluateFirstDeriv }

begin
  if ABS(X) < Tolerance then
    DeltaX := Sqrt(Tolerance)
  else
    DeltaX := ABS(X * Sqrt(Tolerance));
  OldEstimate[1] := EvaluateFirstDeriv(X,DeltaX);
  Iter := 1;
  TwoToTheIterMinus2 := 1;
  repeat
    Iter := Succ(Iter);
    DeltaX := DeltaX / 2;
    NewEstimate[1] := EvaluateFirstDeriv(X,DeltaX);
    TwoToTheIterMinus2 := TwoToTheIterMinus2 * 2;
    FourToTheExtrapMinus1 := 1;
    for Extrap := 2 to Iter do
      begin
        FourToTheExtrapMinus1 := FourToTheExtrapMinus1 * 4;
        NewEstimate[Extrap] := (FourToTheExtrapMinus1 *
          NewEstimate[Extrap - 1] -
          OldEstimate[Extrap - 1]) /
          (FourToTheExtrapMinus1 - 1);
      end;
    OldEstimate := NewEstimate;
  until (ABS(NewEstimate[Iter - 1] - NewEstimate[Iter]) <=
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ABS(Tolerance * NewEstimate[Iter])) or
        (ABS(DeltaX) < Tolerance);
    Differentiate := NewEstimate[Iter];
end;
```

```
procedure plot_function;
label exit1;
var q,i,n,nn,point          : integer;
    delta,w,aa,b,ww,w1,
    dx_max,dx_min,t,ph0,
    ph1,dy_max,dy_min,Root  : real;
    Error                    : byte;
    err1,err2                : boolean;
    d1,d2                    : complex;
    dy,xx                    : string[11];
    alp                      : array[0..20] of string[20];
    wwi                      : array[0..20] of integer;
    ali                      : array[0..20] of real;
begin
    ClrScr;
    n := 100;
    case num of
    '8': begin
        ali[0] := 1;
        wwi[0] := 2;
        for i := 1 to fn do
            begin
                ali[i] := ali[i-1];
                write(' alpha[' ,i,'] = '); readln(ali[i]);
                wwi[i] := wwi[i-1];
                write(' w[' ,i,'] = '); readln(wwi[i]);
            end;
        end;
    end;
    GraphicsMode;
    SelectScreen(2);
    ClearScreen;
    SelectScreen(1);
    logs := false;
    dB_log := false;
    if not freq then
        begin
            if num in ['1'..'5'] then
                begin
                    xx := ^N+'x'+^0;
                    xs := 0;
                end
            else
                begin
                    xx := 't';
                    xs := 0;
                end
            end;
        end;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end
end
else
case xs of
0 : xx := 'F [Hz]';
3 : xx := 'F [KHz]';
6 : xx := 'F [MHz]';
9 : xx := 'F [GHz]';
end;
if not ScreenDisp then
begin
Assign(OutFile,OutName);
Rewrite(OutFile);
InitPlotter;
gotoxy(1,1); Write(' Plot ... ');
end;
case num of
'1':begin
stl[1] := 'U3 =1 ';
stl[2] := 'U3 =-0.2';
stl[3] := 'B1U2 =1 ';
stl[4] := 'B1U2 =-0.2 ';
stl[5] := 'B3 ';
logs := true;
dB_log := true;
DefineWorld(0.01,-6, 1000,6, xs,0);
Axis(5,4, 9,1, 7,5, 'Fig. Magnitude response',xx,' T [dB]');
end;
'2':begin
stl[1] := 'U3 =1 ';
stl[2] := 'U3 =-0.2';
stl[3] := 'B1U2 =1 ';
stl[4] := 'B1U2 =-0.2 ';
stl[5] := 'B3 ';
DefineWorld(0,0, 2,1.2, xs,0);
Axis(10,6, 1,1, 2,5, 'Fig. Magnitude response',xx,' |H |');
end;
'3':begin
stl[1] := 'U3 =1 ';
stl[2] := 'U3 =-0.2';
stl[3] := 'B1U2 =1 ';
stl[4] := 'B1U2 =-0.2 ';
stl[5] := 'B3 ';
DefineWorld(0,-300, 2,0 ,xs,0);
Axis(10,6, 1,1, 2,5, 'Fig.Phase response',xx,
'+^N+h'+^0+ [deg]');
end;
'4','5':begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if num='4' then dy := 'Group Delay'
                else dy := 'Phase Delay';
stl[1] := 'U3    =1  ';
stl[2] := 'U3    =-0.2';
stl[3] := 'B1U2  =1  ';
stl[4] := 'B1U2  =-0.2';
stl[5] := 'B3    ';
DefineWorld(0,0, 2,10, xs,0);
Axis(10,5, 1,1, 2,5, 'Fig. group delay',xx,' '+'D [sec]');
end;

'6':begin
  stl[1] := 'U3    =1  ';
  stl[2] := 'U3    =-0.2';
  stl[3] := 'B1U2  =1  ';
  stl[4] := 'B1U2  =-0.2';
  stl[5] := 'B3    ';
  DefineWorld(0,-0.1, 14,0.5, xs,0);
  Axis(7,6, 1,1, 2,5, 'Fig. Impulse response',xx,'h(t)');
end;

'7':begin
  stl[1] := 'U3    =1  ';
  stl[2] := 'U3    =-0.2';
  stl[3] := 'B1U2  =1  ';
  stl[4] := 'B1U2  =-0.2';
  stl[5] := 'B3    ';
  DefineWorld(0,0, 14,1.2, xs,0);
  Axis(7,6, 1,1, 4,5, 'Fig. Unit step response',
        xx,'Unit step');
end;

'8':begin
  stl[1] := 'U3    =1  ';
  stl[2] := 'U3    =-0.2';
  stl[3] := 'B1U2  =1  ';
  stl[4] := 'B1U2  =-0.2';
  stl[5] := 'B3    ';
  DefineWorld(-5,-2, 2,2 ,0,0);
  Axis(7,4, 1,1, 1,5, ' ','Re','j Im');
end;
end;

if not ScreenDisp then
begin
  BufferReady;
  Write(OutFile,'IW',AXX+AX,' ','AY',' ',AXX+XMax,' ','YMax,');
  n := n*5;
  Write(OutFile,'CA8;');
  Write(OutFile,'SI',0.08+0.014*PScale:6:4,' ',
        0.12+0.022*PScale:6:4,' ');
end;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if num in ['8'] then
begin
  fn := fn shl 1;
  nn := n;
end
else
delta := (X_max-X_min)/n;
for fi := 1 to fn do
begin
  case num of
'3','5':begin
  ph0 := 0;
  ph1 := 0;
  end;
'8':begin
  al := ali[(fi+1) shr 1];
  ww := wwi[(fi+1) shr 1];
  if odd(fi) then n := nn
  else n := 1;
  delta := 2*ww/n;
  end;
end;
dx_min := 1e20;
dx_max := -1e20;
dy_min := 1e20;
dy_max := -1e20;
err1 := false;
err2 := false;
b := Y_min;
if num in ['1'..'7'] then
  SetLineStyle(fi);
else
  if num='8' then
    if odd(fi) then SetLineStyle((fi+1) shr 1)
    else SetLineStyle(8);
  if not ScreenDisp then
  case num of
'1'..'7': Write(OutFile,'SP',fi,',');
end;
'8': Write(OutFile,'SP',(fi+1) shr 1,',');
end;
for i := 0 to n do
begin
  case num of
'1'..'7':begin
  aa := X_min+delta*i;
  if logs then w := power(10,aa) else w := aa;
  end;
end;
  case num of
'1','2':begin
  TargetF(w);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    b := sqrt( sqr(data.re) + sqr(data.im) );
end;
'3','5':begin
    TargetF(w);
    ph0 := ph0-arctan(data.im / data.re);
    if ph0<(-pi/2) then ph1 := ph1-1
        else if ph0>(pi/2) then ph1 := ph1+1;
    ph0 := arctan(data.im / data.re);
    if num='3' then
        b := (ph1*pi + ph0) * 180/pi
    else
        begin
            if w<>0 then b := -(ph1*pi + ph0) / w
                else b := -(ph1*pi + ph0) / 1e-10;
            if freq then b := b/(2*pi);
        end;
    end;
'4':begin
    b := -Differentiate(w);
    if freq then b := b/(2*pi);
    end;
'6':begin
    b := TargetT(w);
    end;
'7':begin
    b := TargetU(w);
    end;
'8':begin
    Nyquist(w);
    aa := data.re;
    b := data.im;

    if (not ScreenDisp) and (not odd(fi)) then
        begin
            pset(aa,b); Write(OutFile,'CI5;PU;');
            if b>=0 then
                Write(OutFile,'PR',0,',',',',AX*0.15:6:4)
            else
                Write(OutFile,'PR',0,',',',',-AX*0.15:6:4);
            if (i=n) or (i=0) then
                Write(OutFile,';L05;LB',^N,'x',^O,' = ',w:3:1,^C);
        end;
    end;
end;
if (b<-1e10) then
    begin
        err1 := true;
        goto exit1;
    end;
if (b>1e10) then
    begin
        err2 := true;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    goto exit1;
end;
if dB_log then
begin
    b := 20*log(b);
    if (b<-300) then
    begin
        err1 := true;
        goto exit1;
    end;
end;
if num in ['1'..'7'] then
begin
    if dy_min>b then
    begin
        dy_min := b;
        dx_min := w;
    end;
    if dy_max<b then
    begin
        dy_max := b;
        dx_max := w;
    end;
    if i=0 then pset(aa,b) else PlotLine(aa,b);
end
else
if odd(fi) then
begin
    if dy_min>b then dy_min := b;
    if dy_max<b then dy_max := b;
    if dx_min>aa then dx_min := aa;
    if dx_max<aa then dx_max := aa;
    if i=0 then pset(aa,b) else PlotLine(aa,b);
end;
regs.ah := 6;
regs.dl := $ff;
MsDos(Regs);
if (regs.al=27) then
begin
    if not ScreenDisp then Write(OutFile,#27,'.k');
    goto exit1;
end;
end;
exit1:
Sound(2000);
Delay(150);
NoSound;
if ScreenDisp then SwapScreen;
if num in ['1'..'7'] then
begin
    Write('function ',fi,' : YMin ');
    if err1 then write('<') else write('=');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Writeln(dy_min,'      X -->',dx_min);
Write('YMax ':21);
if err2 then write('>') else write('=');
Writeln(dy_max,'      X -->',dx_max);

if num in ['1','2'] then
begin
  Fre_3db(0, 1e3 , Root , Error );
  if Error=0 then Writeln('W3db = ':23,Root:11:8);
end;
end;
else
if odd(fi) then
begin
  Writeln('function ',(fi+1) shr 1,':');
  Write(' w 1 =');
  Stable(0, 0.1 , Root , Error );
  if Error=0 then
  begin
    Nyquist(Root);
    Write(Root:12:5,' Re 1 =',data.re:12:5);
  end;
else
  Write('**** Error ****':31);
  Write(' w 2 =');
  Stable(w, ww+0.1 , Root , Error );
  if Error=0 then
  begin
    Nyquist(Root);
    Writeln(Root:12:5,' Re 2 =',data.re:12:5);
  end;
else
  Writeln('**** Error ****':31);
end;
Write(' Xmin =',dx_min:10:3);
Write(' Xmax =',dx_max:10:3,' Ymin ');
if err1 then write('<') else write('=');
Write(dy_min:10:3,' Ymax ');
if err2 then write('>') else write('=');
Writeln(dy_max:10:3);
Delay(1500);
if ScreenDisp and (fi<>fn) then SwapScreen;
end;
if not ScreenDisp then
begin
  Write(OutFile,'IN;SP0;');
  if PScale>5 then
    Write(OutFile,'PA16640,11400;')
  else
    Write(OutFile,'PA0,0;');
  Close(OutFile);
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Alarm;
Wait(1);
if ScreenDisp then
begin
  SwapScreen;
  Wait(1);
  SwapScreen;
  Wait(1);
end;
TextMode;
end;

begin
TextMode;
AllocateRAMScreen;
ColorGlb := 255;
repeat
  ClrScr;
  gotoxy(17,4); write('-----');
  gotoxy(25,5); write('Composite Polynomials Filters ');
  gotoxy(17,6); write('-----');
  gotoxy(24,9); write('1) Magnitude Response [db]');
  gotoxy(24,10); write('2) Magnitude Response');
  gotoxy(24,11); write('3) Phase Response');
  gotoxy(24,12); write('4) Group Delay');
  gotoxy(24,13); write('5) Phase Delay');
  gotoxy(24,14); write('6) Impulse Response');
  gotoxy(24,15); write('7) Unit Step Response');
  gotoxy(24,16); write('8) Nyquist');
  gotoxy(24,17); write('9) Other Menu');
  gotoxy(24,18); write('0) Display');
  gotoxy(24,19); write('ESC Quit');
  gotoxy(27,21); write('Select No. ');
  repeat
    gotoxy(38,21); read(kbd,num); write(num);
  until num in [#27,'0'..'9'];
  if num='9' then
    begin
      ClsScr;
      other;
    end;
  if num='0' then
    begin
      GraphicsMode;
      SwapScreen;
      Wait(1);
      SwapScreen;
      TextMode;
    end;
  if num in ['1'..'8'] then
    begin
      gotoXY(1,25);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
write
('Output [A]: Screen & Printer (^P=PrintScreen) [B]: Plotter ');
repeat
  read(kbd,ch);
  ch := UpCase(ch);
until (ch='A') or (ch='B');
if ch='A' then
  begin
    ScreenDisp := true;
    XMax := XScreenMax;
    YMax := YScreenMax;
  end
else
  begin
    gotoXY(1,25);
    write('Output Port [A]: Serial [B]: Parallel ');
    repeat
      read(kbd,ch);
      ch := UpCase(ch);
    until (ch='A') or (ch='B');
    if ch='A' then OutName := 'AUX'
    else OutName := 'PRN';
    ScreenDisp := false;
    XMax := Round(XGlbMax * PScale);
    YMax := Round(YGlbMax * PScale);
    AX := Round(AxisX * PScale);
    AY := Round(AxisY * PScale);
    AXX := Round(AX*1.2);
  end;
  fn := 5;
  freq := false;
  xs := 0;
  Plot_Function;
end;
until num=#27;
end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้