



คิสรวิวิทภักษาไลน์ฟิวดเอดร์ และการประสุกดใ้งงาน
DISTRIBUTED RC LINE FILTERS AND ITS APPLICATION



ปรืตฐานัพนัสนั้ เป็นส่วนหนังของการศัภษาตามหลักสูดรุดศัภกรรรมศัภสร์บัณฑัศ
ศัภษาเทคโนโลยือืเด็ภทรอนัภสั
ภากวืศัาเทคนัคคุดศัภกรรรม
คณะวืศัภกรรรมศัภสร์
สธัาบันเทคโนโลยืพระจอมเกล้า เจ้าคุณทหาร ลาดกระบั้ง
ปัภการศัภษา 2535

1.
3
2 5

เอกสารนั้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศัภษาเท่านั้น ไม่อนุญาดัให้นัาไปใช้ประโยชนัด้านการค้า
 ไม่ว่ากรณัใดๆ ทังสััน อืกทังห้ามมิให้ดัดแปลงเน้อหาและดองอ้างอึงถึงเจ้าของเอกสารทุกคร้งที่มีกรนัไปใ้

หัวข้อปริศยานิพนธ์ คณิตรวีวิทท์อาชีไลน์ไฟเตอร์ และ การประยุกต์ใช้งาน

นักศึกษา	นาธณา พรวิศณุกุล	34131110
	นายปรีชา เฟดิมศักดิ์นิทษา	34131112
	นายมงคล ปัตนา	34131123
อาจารย์ที่ปรึกษา	ผศ.ดร.กนก เจนจิระวงศ์เวช	
ปีการศึกษา	2535	

บทคัดย่อ

ปริศยานิพนธ์ฉบับนี้ได้เสนอหลักการออกแบบวงจรรองความถี่ และวงจรอีควอไลเซอร์ที่สามารถยกระดับหรือลดระดับผลตอบสนองทางขนาดของสัญญาณที่่านความถี่สูงได้ โดยใช้วงจรแบบแอกทีฟคิสิทรีวิทท์อาชี พร้อมทั้งพิจารณาถึงเสถียรภาพของระบบและค่าความไวของอุปกรณ์ในวงจรที่ออกแบบได้

นอกจากนี้ยังเสนอหลักการในการประมาณฟังก์ชันของวงจรดีเลย์อีควอไลเซอร์ที่สามารถกำหนดคุณลักษณะผลตอบสนองของกรูฟดีเลย์ตามที่ต้องการได้ ซึ่งสามารถใช้ในการชดเชยเฟสของสัญญาณหรือแก้ไขความผิดเพี้ยนทางเวลาเพื่อให้ได้กรูฟดีเลย์คงที่ โดยการประมาณค่าเริ่มต้นของโพลและซีโรแบบขนาน และใช้คอมพิวเตอร์ช่วยในการคำนวณหาค่าตอบของทรานส์เฟอร์ฟังก์ชันที่ถูกประมาณขึ้นเพื่อให้ได้ตำแหน่งของโพลและซีโรที่ดีที่สุด ในการออกแบบวงจรอีควอไลเซอร์ดังกล่าวข้างต้นนั้นได้นำเอาสัญญาณทดสอบมีอคคูลेटชันกำลังสองพัลซ์มาช่วยในการออกแบบเพื่อแก้ความผิดเพี้ยนทางขนาดของสัญญาณสี่ในระบบการส่งสัญญาณภาพของโทรทัศน โดยไม่มีผลต่อความผิดเพี้ยนทางกรูฟดีเลย์ของสัญญาณสี่ จากการทดลองวงจรดังกล่าวนี้ให้ผลใกล้เคียงกับทฤษฎีการคำนวณด้วยคอมพิวเตอร์

Paper Title Distributed RC Line Filters and Its Application
 Name Mr.Thana Pornvisanukul 34131110
 Mr.Preecha Padermsakpittaya 34131112
 Mr.Mongkol Phatthana 34131123
 Adviser Asst.Prof. Dr.Kanok Janchitrapongvej
 Academic Year 1992

Abstract

This paper imply , the method of realization of active distributed RC filters and an equalizer using uniform distributed RC circuit are presented. The amplitude frequency response of this equalizer can be increased or decreased at higher frequencies. Here in, the stability and sensitivity due to the variation of element values of the active distributed RC filters and an equalizer are investigated.

This paper also describe an iterative method of group delay equalization over a specified amplitude frequency response of the design equalizer. Whereby an equal ripple approximation may be achieved with the aid of electronic computer in searching the optimized poles and zeros.

By the used of the modulated sine-squared test signal in designing equalizer mentioned above, it is shown that the equalizer can be used to correct the relative gain distortion of the chrominance signal in the color picture signal transmission system without relative delay distortion.

The experimental results and computer simulations of the circuit showed good agreement with the theoretical values.

ปริญญานิพนธ์ปีการศึกษา 2535

ภาควิชา เทคโนโลยีอุตสาหกรรม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

เรื่อง คณิตวิธีทำอาชีพไลน์ฟิลเตอร์ และการประยุกต์ใช้งาน

ผู้จัดทำ

1. ธนา พรวิศกุล 34131110

2. ปรีชา เฟดิมศักดิ์พิทยา 34131112

3. มงคล บัดนา 34131123

.....อาจารย์ที่ปรึกษา

(ผศ.ดร. กนก เจนจิระพงศ์เวช)

.....กรรมการ

()

.....กรรมการ

()

.....กรรมการ

()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
Abstract	II
บทที่ 1 บทนำ	1
1.1 วัตถุประสงค์ของปริศยานิพนธ์	1
1.2 เนื้อหาของปริศยานิพนธ์	2
บทที่ 2 ดิสทริบิวท์อาซี	3
บทที่ 3 การวิเคราะห์พารามิเตอร์และการออกแบบวงจร แอกทีฟดิสทริบิวท์อาซี	5
3.1 การวิเคราะห์ค่าพารามิเตอร์ของดิสทริบิวท์อาซี	5
3.2 การวิเคราะห์ทางคณิตศาสตร์เพื่อหาค่าพารามิเตอร์	7
3.3 การออกแบบวงจรแอกทีฟดิสทริบิวท์อาซี	10
3.4 ความไวของอุปกรณ์	10
3.5 การออกแบบวงจรรองความถี่แบบแอกทีฟดิสทริบิวท์อาซี	13
3.6 การทดสอบวงจรแอกทีฟดิสทริบิวท์อาซีเพื่อการใช้งาน	26
บทที่ 4 การวัดความผิดเพี้ยนของอัตราขยายและดีเลย์ โดยใช้มีมอดคูลเลทไซน์กำลังสองพัลส์	35
4.1 การคำนวณหาความถี่สเปกตรัมของสัญญาณรวมมีมอดคูลเลทพัลส์ 20T	37
4.2 ทฤษฎีและการหาสูตรสำเร็จของความผิดเพี้ยนของอัตรา การขยายและดีเลย์	38
บทที่ 5 หลักการของชุดทดสอบการตอบสนองทางขนาดและการตอบสนอง ทางเฟสของสัญญาณสี่	46
5.1 การทำงานของระบบทดสอบเพื่อแก้ความผิดเพี้ยนของ สัญญาณโครมมิตันซ์ต่อลูนิตแนนท์	49
5.2 การออกแบบวงจรอควาไลเซชันแบบแอกทีฟดิสทริบิวท์อาซี	53
5.3 การออกแบบวงจรดีเลย์อควาไลเซชัน	59
5.4 การทดลองใช้งานวงจรอควาไลเซชันแบบแอกทีฟดิสทริบิวท์อาซีเพื่อแก้ ความผิดเพี้ยนของสัญญาณโครมมิตันซ์ต่อลูนิตแนนท์	67
บทที่ 6 สรุปผลการวิจัย	76
กิตติกรรมประกาศ	77
เอกสารอ้างอิง	78

บทที่ 1

บทนำ

ในการออกแบบวงจรเพื่อให้ได้ผลของการตอบสนองทางขนาดเป็นไปตามที่ต้องการ ดังเช่น การออกแบบวงจรรองความถี่ ซึ่งการสังเคราะห์วงจรนั้นอาจจะเป็นแบบ พาสซีฟที่ใช้ อลิเมนต์ R , C และ L หรือแบบ แอกทีฟที่ใช้โอลิเมนต์ R , C ร่วมกับ ออปแอมป์ แต่สำหรับ ปรวิญญานนิพนธ์ฉบับนี้ ได้เสนอวิธีการสังเคราะห์วงจรโดยใช้ เน็ทเวิร์กแบบคิสทริบิวท์อาชี (Dis tributed RC) [11,12] แทนการออกแบบวงจรที่เป็นแบบลัมด์โอลิเมนต์ (Lumped Element) ซึ่งเน็ทเวิร์กแบบคิสทริบิวท์อาชีมีคุณลักษณะทั่วไปที่ดีกว่า มีขนาดเล็กกว่า เน็ทเวิร์กแบบลัมด์อาชี โดยจะเสนอวงจรรองความถี่ต่ำ วงจรรองความถี่สูง และวงจรอควอลไลเซอร์ที่สามารถยกหรือ ลดระดับผลตอบสนองทางขนาดของสัญญาณที่ผ่านความถี่สูง [1] โดยใช้วงจรแอกทีฟคิสทริบิวท์อาชี (Active Distributed RC) ที่มีลักษณะเป็นยูนิฟอร์มคิสทริบิวท์อาชี (Uniform Distributed RC) หรือ URC โดยวงจรถูกกล่าวว่ามีค่าความไว (Sensitivity) ของอุปกรณ์ต่ำ สามารถผลิต ขึ้นตามกรรมวิธีผลิต LSI ที่มีขนาดเล็ก โดยที่ค่า Parasitic Capacitors [5] ระหว่างตัว ยูนิฟอร์มคิสทริบิวท์อาชีกับ Substrate ของไอซีจะไม่มีผลกระทบต่อคุณลักษณะของวงจร ทำให้ สามารถนำไปใช้ได้ดีในช่วงความถี่สูง

และเพื่อต้องการให้การตอบสนองทางเฟสเป็นเชิงเส้นซึ่งจะทำให้กรุปดีเลย์คงที่ จึงจำเป็นต้องมีการชดเชยเฟสเพื่อแก้ความผิดเพี้ยนทางเวลาของสัญญาณ จึงได้อาศัยหลักการประมาณ พังก์ชันของวงจรถีเลอีอควอลไลเซอร์ [2] ซึ่งจะสามารถกำหนดคุณลักษณะของกรุปดีเลย์ตามที่ ต้องการได้ โดยการประมาณค่าเริ่มต้นให้ตำแหน่งของ โพลและซีโรขนานไปกับแกน $j\omega$ ใน S-Plane และใช้โปรแกรมคอมพิวเตอร์ในภาคผนวก ช่วยในการอิตเตอเรทหาค่าตำแหน่งของโพล และซีโรที่ดีที่สุด

1.1 วัตถุประสงค์ของปรวิญญานนิพนธ์

วัตถุประสงค์ของปรวิญญานนิพนธ์ฉบับนี้ ก็เพื่อที่จะศึกษาการใช้งานของเน็ทเวิร์กแบบคิสทริ บิวท์อาชีไลน์เฟลเตอร์เพื่อเป็นแนวทางในการออกแบบ วงจรรองความถี่ต่ำ วงจรรองความถี่สูง โดยใช่วงจรแบบแอกทีฟคิสทริบิวท์อาชี ศึกษาการวัดความผิดเพี้ยนของอัตราขยายและดีเลย์ โดยใช้อัตตูลูเลขไซน์กำลังสองพัลส์ และศึกษาการทดลองวงจรที่ใช้สำหรับแก้ไขความผิดเพี้ยนของ สัญญาณโครมิแนนซ์ต่อลุมิแนนซ์ใน ระบบการส่งสัญญาณภาพโดยใช้ วงจรอควอลไลเซอร์แบบแอกทีฟ คิสทริบิวท์อาชีเพื่อแก้ไขทางขนาด และใช่วงจรถีเลอีอควอลไลเซอร์เพื่อแก้ไขการตอบสนองทาง

1.2 เนื้อหาของปริศยานิพนธ์

สำหรับเนื้อหาของปริศยานิพนธ์นั้นจะเป็นการนำเอาตัวคิสรวิวิทอาชีไลน์ฟิลเตอร์ที่เป็นชนิดยูนิฟอร์ม ซึ่งภายในประกอบด้วยค่าความต้านทานและค่าตัวเก็บประจุ จำนวน 2 ตัวร่วมกับออปแอมป์มาใช้ในการออกแบบวงจร โดยการทดลองนั้นจะใช้การประมาณตัวยูนิฟอร์มคิสรวิวิทอาชีด้วยโครงสร้างแบบลัมด้าซีต่อเข้าด้วยกันประมาณ 10 เซกชั่น ซึ่งในบทที่ 2 จะเป็นการแสดงโครงสร้างของคิสรวิวิทอาชี การวิเคราะห์ค่าพารามิเตอร์ของคิสรวิวิทอาชี โดยการเปลี่ยนสัญลักษณ์ทางไฟฟ้าไปเป็นสัญลักษณ์ทางคณิตศาสตร์ ในบทที่ 3 จะเป็นแนวทางการออกแบบวงจรที่เสนอในปริศยานิพนธ์นี้ คือ วงจรกรองความถี่ต่ำ วงจรกรองความถี่สูง โดยใช้แอกทีฟคิสรวิวิทอาชี ซึ่งจะเปรียบเทียบกับผลที่ได้จากการวิเคราะห์โดยคอมพิวเตอร์กับการวิเคราะห์โดยการทดลอง สำหรับในบทที่ 4 จะกล่าวถึงการวัดความผิดเพี้ยนของอัตราขยายและดีเลย์โดยใช้มีอดคูลูเลทไซน์กำลังสองพัลส์เพื่อตรวจสอบข้อบกพร่องของระบบการส่งโทรทัศน และในส่วนของบทที่ 5 จะเป็นการนำเอาเนื้อหาตั้งแต่บทที่ 2 จนถึงบทที่ 4 มาประยุกต์ใช้งานเพื่อการทดสอบและเป็นแนวทางในการนำไปใช้งานจริง โดยจะกล่าวถึงการออกแบบวงจรอควาไลเซชันเพื่อแก้ไขความผิดเพี้ยนทางขนาดของสัญญาณโครมิกแนชต่อลูมิแนช และ ใช้วงจรดีเลย์อควาไลเซชันแก้ไข ความผิดเพี้ยนทางเฟส โดยอาศัยหลักการประมาณฟังก์ชันเพื่อหาค่าโพลและซีโรที่เหมาะสมซึ่งทำให้รู้ดีเลย์ โดยใช้โปรแกรมคอมพิวเตอร์ที่ภาคผนวกช่วยในการอติเตอ เรทหาโพลและซีโรตามที่ต้องการ

สำหรับในบทสุดท้ายนั้นจะเป็นการสรุปผลการวิจัยที่ได้จากการวิเคราะห์และการทดลองตลอดจนแนวทางการพัฒนาเพื่อนำไปใช้งานต่อไป

ภาคผนวก จะเป็นการแสดงการวิเคราะห์หาทรานส์เฟอร์ฟังก์ชันของวงจรกรองความถี่ต่ำ วงจรกรองความถี่สูง และวงจรอควาไลเซชัน แบบแอกทีฟคิสรวิวิทอาชี พร้อมทั้งรายละเอียดของโปรแกรมทั้งหมด

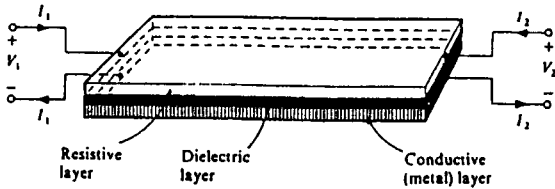
บทที่ 2

คิสรวิวิท อวชีไลน และยูนีฟอร์มคิสรวิวิทอวชี

Distributed RC line & Uniformly Distributed RC line (URC)

ปริญญานพนธ์ฉบับนี้ ได้มีการเสนอการออกแบบวงจร ที่มีส่วนประกอบของเน็ทเวิร์กแบบ คิสรวิวิท อวชี (Distributed RC) แทนการออกแบบ วงจรที่ใช้ ลัมด้อลิเม้นท์ (Lumped Element) แบบพาสซีฟ (Passive) หรือ แบบแอคทีฟ (Active) ซึ่งปัจจุบันเน็ทเวิร์กแบบคิสรวิวิทอวชีได้ถูกสร้างขึ้นโดยใช้เทคโนโลยี Thick-Film หรือ Thin-Film ในการผลิตเป็นไอซี (Integrated Circuit) โดยที่เน็ทเวิร์กแบบนี้ จะมีลักษณะต่างๆไปที่ดีกว่า และมีขนาดเล็กกว่า เน็ทเวิร์กแบบลัมด้อลิเม้นท์อวชีซึ่งทำให้การออกแบบวงจร เน็ทเวิร์กคิสรวิวิทอวชีสามารถนำไปใช้ได้ง่าย ด้วยเหตุนี้ เน็ทเวิร์กแบบคิสรวิวิทอวชีจึงถูกสร้างขึ้นมาหลายรูปแบบ ดังเช่น โครงสร้าง Thin-Film แบบ Multi-layer ที่มีชั้นของตัวนำ (conductor) , ความต้านทาน (Resistive) และฉนวน (Dielectrics) ประกอบติดเข้าด้วยกันโดยจะมีจุดต่อออกมาด้านข้างของชั้นความต้านทานและตัวนำ หรือมีโครงสร้างในรูปแบบที่เป็น Multi-Contacted P-N Junction ซึ่งตัวต้านทานจะใช้ส่วนประกอบของเซมิคอนดักเตอร์ และตัวเก็บประจุจะใช้ผลของไบแอสที่จังก์ชัน

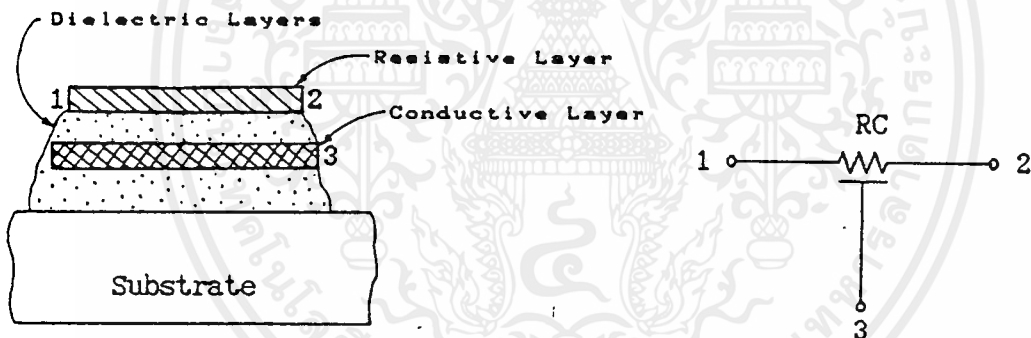
โครงสร้าง และ พารามิเตอร์ของ คิสรวิวิทอวชี - โครงสร้างแบบ 2 พอร์ตของคิสรวิวิทอวชี สามารถแสดงในรูปที่ 1.1 ซึ่งมีส่วนประกอบต่าง ๆ ดังนี้คือ ส่วนบนสุดเป็นชั้นของความต้านทาน (Resistive layer) รองลงมาเป็นชั้นของฉนวน (Dielectric layer) และ ชั้นของตัวนำ (Conductive layer) โดยที่ความหนาของแต่ละชั้นนั้นประมาณ 10^{-5} นิ้ว จึงทำให้วงจรมีขนาดเล็กมาก (Microcircuits) และสามารถทำให้อยู่ภายในไอซีได้ (Passive Integrated Circuit) [11,13] โดยมี โครงสร้างเป็นแบบ Thin-Film หรือ โครงสร้างแบบ Monolithic



เอกสารนี้เป็นเอกสารที่รูปที่ 1.1 แสดงโครงสร้างของ 2 พอร์ตของ Distributed RC โยชนด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างแบบ Thin-Film ประกอบด้วย ชั้นของวัสดุหลายอย่าง ซึ่งถูกวางอยู่บนตัวกลางที่เหมาะสม (Substrate) ตัวอย่างเช่น แผ่นบางๆ ของสาร Dielectric อาจจะถูก Titanate ด้วยแผ่นบางของ Nichrome Resistive Film ซึ่งวางไว้ด้านบน และ แผ่น Conductive Copper-Film ที่วางไว้ด้านล่าง หุ้มห่อด้วยสาร Dielectric และอยู่บน Passive Substrate ด้วยวิธี Vaporization หรือ Electrochemical Technique

โครงสร้าง Monolithic ประกอบด้วยชั้นของสารเซมิคอนดักเตอร์ถูกทำเป็นรูปสี่เหลี่ยมเล็ก ๆ เช่น Distributed Resistance ได้มาจาก Lightly Doped เซมิคอนดักเตอร์และ Distributed Capacitance ได้มาจากการป้อนไบแอสกลับข้างของรอยต่อภายใน P-N จังก์ชัน ตัวอย่างโครงสร้างของวงจร Uniform Distributed RC [5] ซึ่งอยู่ในรูปที่แสดงดังรูปที่ 1.2 (a) โดยที่มีสัญลักษณ์ของวงจรแสดงดังรูปที่ 1.2 (b)



(a) โครงสร้างของคิสทรีบิวท์อาร์ซี

(b) สัญลักษณ์

รูปที่ 1.2 แสดงโครงสร้างและสัญลักษณ์ของ Uniform

บทที่ 3

การวิเคราะห์พารามิเตอร์และการออกแบบวงจรแยกที่ฟิลส์ทรีบีวท์อาซี

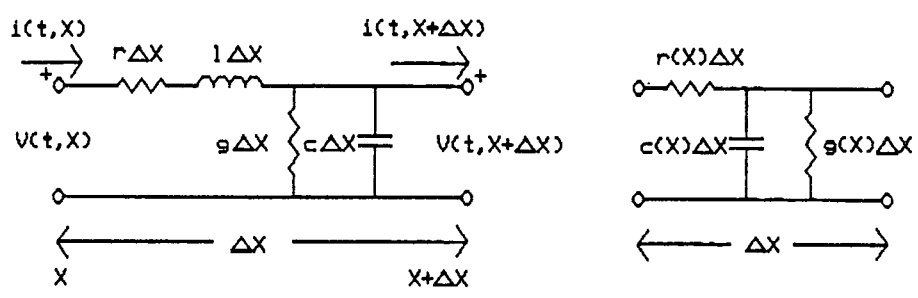
สำหรับบทนี้จะกล่าวถึงการวิเคราะห์ค่าพารามิเตอร์ทางโครงสร้าง และการออกแบบของ วงจรแยกที่แบบดิสทรีบีวท์อาซี ซึ่งการวิเคราะห์พารามิเตอร์นั้น จะช่วยในการออกแบบวงจรและ นำไปซึ่ง การทดสอบ และ การประยุกต์ใช้งานของวงจรอควอลไอเซอร์ เพื่อแก้ความผิดเพี้ยนของ สัญญาณโครมโมแนนท์ต่อลุมิแนนท์ในระบบการส่งสัญญาณภาพของโทรทัศน์ ซึ่งเป็นจุดประสงค์หลักของ ปรวิญานพนธ์ฉบับนี้ โดยที่การวิเคราะห์ค่าพารามิเตอร์ของดิสทรีบีวท์อาซีนั้นใช้ หลักการเดียวกัน กับของสายส่ง (Transmission Line)

ในส่วนของการออกแบบของวงจรมันจะ เสนอการใช้งานของ วงจรกรองความถี่ต่ำ (Low Pass Filter ,LPF) วงจรกรองความถี่สูง (High Pass Filter ,HPF) ซึ่งจะใช้ ตัวชุนiformly Distributed RC Line,URC) 2 ตัวร่วมกับออปแอมป์ โดยมีขั้นตอนการทดลองดังนี้

- ทำการวิเคราะห์โดยใช้คอมพิวเตอร์ (Computer Simulation)
- ต่อวงจรการทดลองเพื่อหาการตอบสนองของดิสทรีบีวท์อาซี
- เปรียบเทียบผลการทดลองที่ได้และสรุปผลการทดลอง

3.1 การวิเคราะห์ค่าพารามิเตอร์ของดิสทรีบีวท์อาซี

จากรูปที่ 3.1 ซึ่งเป็นโครงสร้างของสายส่ง (Transmission Line) ที่มีค่าพารามิเตอร์อินดักแตนซ์ l, คาพาซิแตนซ์ c, รีซิสเตอร์ r และคอนดัคแตนซ์ g กระจายอยู่ตลอดความยาวสาย ถ้าค่าพารามิเตอร์เหล่านี้ไม่มีความสัมพันธ์กับความยาวของสาย เรียกสายส่งชนิดนี้ว่า สายส่งแบบชุนiform (Uniform Transmission Line)

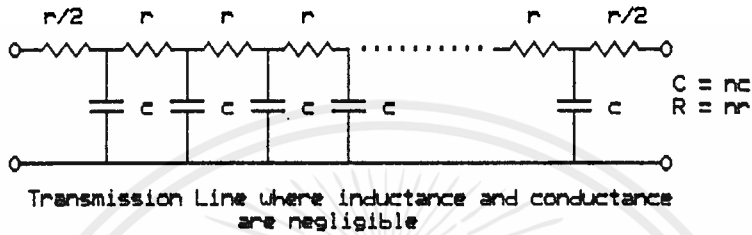


(ก) โครงสร้างของสายส่ง (ข) โครงสร้างของ Distributed RC

รูปที่ 3.1 แสดงการเปรียบเทียบโครงสร้างของสายส่ง และ ดิสทรีบีวท์อาซี

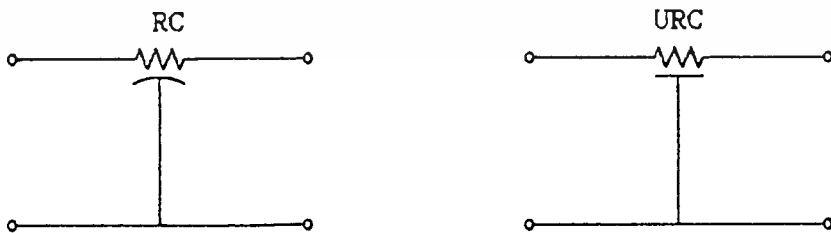
เอกสารนี้เป็นเอกสารที่งานในชั้นเรียนหรือใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในทำนองเดียวกันจากรูปที่ 3.1 (ข) ซึ่งเป็นโครงสร้างเน็ตเวิร์กแบบ 2 พอร์ตของ Distributed RC โดยมีลักษณะเหมือนกับโครงสร้างของสายส่ง รูปที่ 3.1 (ก) ขณะที่ค่าของความนำ g นั้นมีค่าน้อยมาก ๆ จนสามารถตัดทิ้งได้คือให้ค่า g เป็นศูนย์ ก็จะได้เน็ตเวิร์กแบบ ดิสทริบิวท์อาซี ดังในรูปที่ 3.2



รูปที่ 3.2 โครงสร้างของดิสทริบิวท์อาซีเมื่อค่าความนำ g มีค่าน้อยมาก ๆ

ถ้าค่าของรีซิสแตนซ์ r และคาปาซิแตนซ์ c ไม่แปรเปลี่ยนไปตามความยาวของ X ก็จะเป็นแบบ Uniform Distributed RC เช่นเดียวกับสายส่ง แต่ถ้าแปรเปลี่ยนไปตามความยาวของ X ก็จัดเป็นแบบ Nonuniform Distributed RC โดยสัญลักษณ์ทั้งสองแบบนี้แสดงในรูปที่ 3.3 ซึ่งเส้นโค้งบ่งบอกว่าเป็นเน็ตเวิร์กแบบ Nonuniform Distributed RC ในขณะที่เส้นตรงนั้นจะเป็น เน็ตเวิร์กแบบ Uniform Distributed RC (URC) ซึ่งในการวิเคราะห์และการออกแบบจะใช้เน็ตเวิร์กแบบ Uniform Distributed RC Line หรือใช้ชื่อย่อว่า URC แสดงในรูปที่ 3.3 (ข)



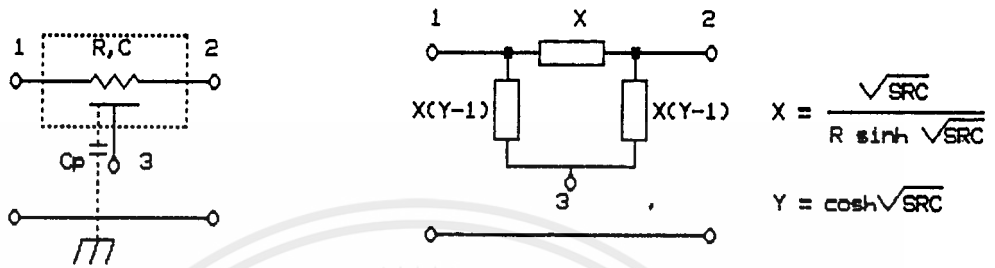
(ก) Nonuniform

(ข) Uniform

รูปที่ 3.3 แสดงสัญลักษณ์ของ Distributed RC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการวิเคราะห์เน็ตเวิร์กแบบคิสรวิบาทอาซีนั้นจะใช้ตัว ยูนิฟอร์มคิสรวิบาทอาซี (Uniformly Distributed RC;URC) แบบ 3 พรอท มาทำการวิเคราะห์ค่าพารามิเตอร์ พิจารณารูปที่ 3.4 (ก) ซึ่งแสดงสัญลักษณ์ทางไฟฟ้า

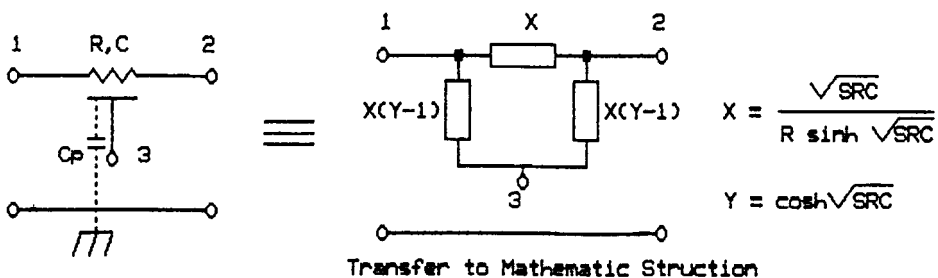


รูปที่ 3.4 แสดงสัญลักษณ์ทางไฟฟ้าแบบ 3 ports ของ URC

และเพื่อง่ายต่อการวิเคราะห์จะทำ การแปลงสัญลักษณ์ทางไฟฟ้า เป็นสัญลักษณ์ในทางคณิตศาสตร์ ดังในรูปที่ 3.4 (ข) และอาศัย Floating Matrix ทำการหาค่า Admittance Parameter $[Y_{1,1}]$ มาช่วยใช้ในการวิเคราะห์และออกแบบวงจร

3.2 การวิเคราะห์ทางคณิตศาสตร์เพื่อหาค่าพารามิเตอร์

โดยใช้หลักการที่กล่าวมาแล้วเพื่อหาค่า Admittance Parameter ของ ยูนิฟอร์มคิสรวิบาทอาซี สามารถกระทำได้ดังนี้ เปลี่ยนรูปสัญลักษณ์ทางไฟฟ้าเป็นสัญลักษณ์ในทางคณิตศาสตร์ ดังในรูปที่ 3.4 (ข) โดยใช้ Floating Matrix แก่เน็ตเวิร์กของวงจรเพื่อหาค่า Admittance Parameter $[Y_{1,1}]$ ของ URC



จากเน็ทเวิร์กของยูนิฟอร์มคิสทรีบิวท์อาชีโดยใช้ Floating Matrix ทำการหาค่า Admittance Parameter [$Y_{i,j}$]

$$\begin{bmatrix} I1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} XY & -X & -X(Y-1) \\ -X & XY & -X(Y-1) \\ -X(Y-1) & -X(Y-1) & 2X(Y-1) \end{bmatrix} \begin{bmatrix} V1 \\ V2 \\ V3 \end{bmatrix}$$

$$= X \begin{bmatrix} Y & -1 & -(Y-1) \\ -1 & Y & -(Y-1) \\ -(Y-1) & -(Y-1) & 2(Y-1) \end{bmatrix} \begin{bmatrix} V1 \\ V2 \\ V3 \end{bmatrix}$$

เมื่อแทนค่า $X = \sqrt{sRC} / R \sinh\sqrt{sRC}$ และ $Y = \cosh\sqrt{sRC}$ (หมายเหตุ ค่า X และ Y ที่ถูกกำหนดขึ้นนี้มาจากการวิเคราะห์ทางโครงสร้างแบบลัมด้อลิเมนต์ที่ เช่นเดียวกับสายส่ง) ดังนั้นจะได้พารามิเตอร์ความนำของเน็ทเวิร์กแบบยูนิฟอร์มคิสทรีบิวท์อาชี (Admittance Parameter)

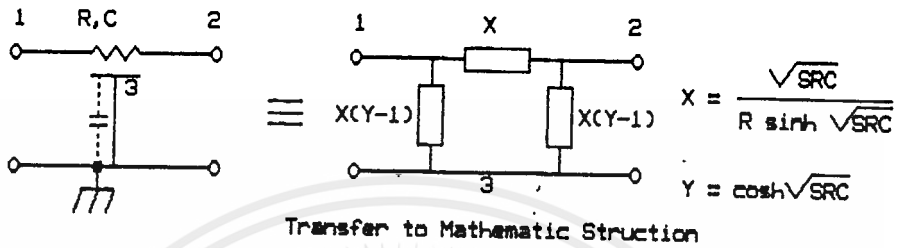
$$[Y_{i,j}] = \frac{\sqrt{sCR}}{R \sinh\sqrt{sCR}} \begin{bmatrix} \cosh\sqrt{sCR} & -1 & 1-\cosh\sqrt{sCR} \\ -1 & \cosh\sqrt{sCR} & 1-\cosh\sqrt{sCR} \\ 1-\cosh\sqrt{sCR} & 1-\cosh\sqrt{sCR} & 2(\cosh\sqrt{sCR}-1) \end{bmatrix}$$

...(I)

เมื่อ R และ C นั้นเป็นค่าผลรวมของรีซิสแตนซ์ และ คาปาซิแตนซ์ ของตัว ยูนิฟอร์มคิสทรีบิวท์อาชี ตามลำดับ และ s เป็น ตัวแปรความถี่เชิงซ้อน (Complex-Frequency Variable) เนื่องจากการแปลงสัญลักษณ์ทางไฟฟ้าในรูปที่ 3.4 (ก) ไปเป็นสัญลักษณ์ในทางคณิตศาสตร์ รูปที่ 3.4 (ข) จะทำให้ง่ายต่อการวิเคราะห์และการออกแบบเพื่อที่จะเอาตัว URC เอกสาขาไปใช้งานได้สะดวกมากขึ้น รับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ในการทำงานเดียวกันเมื่อใช้หลักการดังกล่าวข้างต้นมาทำการวิเคราะห์เพื่อหาค่าพารามิเตอร์เนกเวเจอร์ของยูนิฟอร์มดิสทริบิวท์อวี่แบบ 2 พอร์ตซึ่งจะนำไปใช้ในการออกแบบวงจรในบทความต่อไปได้ดังนี้



จากรูปเมื่อแปลงสัญลักษณ์ทางไฟฟ้าเป็นสัญลักษณ์เพื่อการวิเคราะห์ทางพีชคณิตแล้วอาศัยหลักการดังกล่าวจะได้

$$\begin{bmatrix} I_1 \\ 0 \end{bmatrix} = \begin{bmatrix} XY & -X \\ -X & XY \end{bmatrix} = X \begin{bmatrix} Y & -1 \\ -1 & Y \end{bmatrix}$$

เมื่อแทนค่าของ $X = \sqrt{sRC} / (R \sinh \sqrt{sRC})$ และ $Y = \cosh \sqrt{sRC}$ จะได้ Admittance Parameter $[Y_{1,1}]$

$$[Y] = \frac{\sqrt{sRC}}{R \sinh \sqrt{sRC}} \begin{bmatrix} \cosh \sqrt{sRC} & -1 \\ -1 & \cosh \sqrt{sRC} \end{bmatrix} \dots (II)$$

3.3 การออกแบบวงจรแอคทีฟที่ฟิสิกส์ทวิพัวท์อาซี

การสังเคราะห์วงจรแอคทีฟ (Active) จากทรานส์เฟอร์ฟังก์ชัน โดยทั่วไปจะใช้โอปแอมป์ร่วมกับพาสซีฟอิลิเมนต์ R กับ C เราเรียกวงจรดังกล่าวนี้ว่าแอคทีฟอาซี เมื่อเปรียบเทียบกับ การสังเคราะห์วงจรพาสซีฟ (Passive) ซึ่งใช้อิลิเมนต์ R, C และ L แล้ว จะเห็นได้ว่า วงจรแอคทีฟอาซีมีข้อดีที่ว่า มีน้ำหนักน้อย ไม่ต้องการพื้นที่มาก และสามารถทำเป็นรูปแบบย่อเล็ก ๆ โดยอาศัยเทคโนโลยีของ Integrated Circuit เมื่อทำเป็น Mass-Product แล้วราคาก็ไม่แพงมีขนาดเล็กกะทัดรัด ซึ่งวงจรที่มี Inductor (L) ยังต้องทำเป็น Discrete Component ที่ราคาแพง เมื่อใช้แอคทีฟอาซี เช่น วงจรแบบ Sallen and Key ในการสร้าง วงจรกรองความถี่ ถ้าหากนำไปผลิตเป็นไอซีแบบ LSI ที่มีขนาดเล็ก ผลของ Parasitic Capacitors ระหว่างตัวความต้านทาน (R) และตัวเก็บประจุ (C) กับ Substrate ของไอซีอาจมีผลทำให้คุณลักษณะของวงจรเปลี่ยนแปลงไป จากที่กล่าวมาแล้วในบทนี้จะเสนอวิธีการออกแบบวงจรกรองความถี่ต่ำและวงจรกรองความถี่สูง โดยใช้โอปแอมป์ร่วมกับฮุนิฟอร์มดิสทริบิวท์อาซี 2 ตัว ซึ่งเรียกวงจรแบบนี้ว่า วงจรแอคทีฟที่ฟิสิกส์ทวิพัวท์อาซี (Active Distributed RC) โดยวงจรที่ออกแบบได้นั้นนอกจากจะมีข้อดีเหมือนวงจรแอคทีฟอาซีแล้ว ยังมีสมรรถนะในการลดทอนสัญญาณในย่านหยุดสัญญาณ (Stopband) ที่ดี สามารถผลิตขึ้นตามกรรมวิธีผลิต LSI ที่มีขนาดเล็กโดยที่ค่า Parasitic Capacitors ระหว่างตัวฮุนิฟอร์มดิสทริบิวท์อาซีกับ Substrate ของไอซีจะไม่มีผลกระทบต่อคุณลักษณะของวงจร ฉะนั้นจึงสามารถนำไปใช้ได้ดีในช่วงความถี่สูง นอกจากนี้วงจรแอคทีฟที่ฟิสิกส์ทวิพัวท์อาซีที่ออกแบบได้นี้ยังมีค่าพารามิเตอร์ α ซึ่งสามารถกำหนดค่าของผลตอบสนองเปลี่ยนแปลงตามที่ต้องการได้ พร้อมทั้งพิจารณาหาค่าความไว (Sensitivity) ของอุปกรณ์ในวงจรที่ออกแบบได้

3.4 ความไวของอุปกรณ์ (Sensitivity)

การออกแบบวงจรใด ๆ จะต้องพิจารณาถึงค่าอิลิเมนต์แต่ละตัว เพราะค่าอิลิเมนต์ที่ใช้ งานจริง ๆ แล้วจะไม่มีค่าตรงตามอุดมคติ ค่าผิดพลาดที่เกิดขึ้นได้จากผลรวมของความคลาดเคลื่อนของตัวอุปกรณ์ที่เกิดจากกรรมวิธีในการผลิตเอง และคุณลักษณะทางโครงสร้างของอิลิเมนต์ที่เกิด การเปลี่ยนแปลงไปตามสภาพแวดล้อมที่ใช้งานอยู่ เช่น อุณหภูมิของห้อง , ความชื้น , ความเข้มของแสงสว่าง และ อายุการใช้งานของอุปกรณ์ เหตุนี้เองจึงจำเป็นต้องศึกษาผลกระทบของอิลิเมนต์ต่าง ๆ ภายในวงจรที่อาจเปลี่ยนแปลงไม่ตรงตามค่าพิกัด

เซ็นซิวิตีเป็นค่าของการเปลี่ยนแปลงที่เกิดขึ้นที่เอาท์พุทของวงจรซึ่งเป็นผลมาจากการแปรผันของพารามิเตอร์ต่าง ๆ ในวงจรเช่น การเปลี่ยนแปลงของค่าตัวต้านทาน (Resistor), ค่าตัวเหนี่ยวนำ (Inductor) , ค่าตัวเก็บประจุ (Capacitor) , ค่าอัตราขยายของทรานซิสเตอร์ และค่าอัตราขยายของออปแอมป์ (Operation Amplifier) ซึ่งแน่นอนว่าวงจรมีค่าอัตราการผลิตของการเปลี่ยนแปลงของเซ็นซิวิตีมากจะมีการแปรผันของพารามิเตอร์ภายในวงจรมีค่ามาก ซึ่งจะทำให้ผลตอบสนองทางด้านเอาท์พุทเกิดความผิดเพี้ยนได้ง่าย ฉะนั้นโครงสร้างของวงจรที่ตีความค่าเซ็นซิวิตีของอิลิเมนต์แต่ละตัวต่ำ จะทำให้ง่ายต่อการนำไปใช้งานและยังไม่ต้องเลือกใช้ค่าของอุปกรณ์ที่มีความเที่ยงตรงสูงมากเกินความจำเป็น ซึ่งจะทำให้ประหยัดค่าใช้จ่าย

การหาเซ็นซิวิตีในวงจร หาได้จากทรานส์เฟอร์ฟังก์ชัน ซึ่งเป็นตัวกำหนดค่าพารามิเตอร์ต่าง ๆ ในวงจร โดยเริ่มต้นจากการกำหนดให้

$$S_{x_i}^{T(s)} = \frac{x_i}{T(s)} \cdot \frac{\partial T(s)}{\partial x_i} \quad \dots (3.1)$$

เมื่อ x_i แทนค่าของอิลิเมนต์แต่ละตัวในวงจร ถ้าแทน s ด้วย $j\omega$ ทรานส์เฟอร์ฟังก์ชัน $T(s)$ จะได้

$$T(j\omega) = |T(j\omega)| e^{j\theta(\omega)} \quad \dots (3.2)$$

แทนค่าสมการ (3.2) ลงในสมการ (3.1) ค่าฟังก์ชันของเซ็นซิวิตี สามารถเขียนได้เป็น

$$S_{x_i}^{T(j\omega)} = \frac{x_i}{|T(j\omega)| e^{j\theta(\omega)}} \cdot \frac{\partial}{\partial x_i} \left(|T(j\omega)| e^{j\theta(\omega)} \right) \quad \dots (3.3)$$

จากสมการ (3.3) ซึ่งเป็นสมการอนุพันธ์สามารถเขียนเป็นจำนวนเชิงซ้อนได้คือ

$$S_{x_i}^{T(j\omega)} = \frac{x_i}{|T(j\omega)|} \frac{\partial}{\partial x_i} |T(j\omega)| + jx_i \frac{\partial}{\partial x_i} \theta(\omega) \quad \dots (3.4)$$

จากสมการ (3.4) จะได้ว่า

$$S_{x_i} |T(j\omega)| = \operatorname{Re} S_{x_i}^{T(j\omega)} \quad \dots (3.5)$$

$$S_{x_i} \theta(\omega) = \frac{1}{\theta(\omega)} \operatorname{Im} S_{x_i}^{T(j\omega)} \quad \dots (3.6)$$

จากสมการ (3.5) จะเป็นฟังก์ชันในการหาเซ็นซิวิตีของ Magnitude $|T(j\omega)|$ ซึ่งสามารถเขียนได้ใหม่คือ

$$\begin{aligned} S_{x_i} |T(j\omega)| &= \operatorname{Re} \left(\frac{x_i}{T(s)} \cdot \frac{\partial}{\partial x_i} T(s) \right) \\ &= \operatorname{Re} \left(x_i \left(\frac{N'(s)}{N(s)} - \frac{D'(s)}{D(s)} \right) \right) \end{aligned} \quad \dots (3.7)$$

เมื่อ $N(s)$ เป็นโพลีโนเมียลเศษของ $T(s)$

$D(s)$ เป็นโพลีโนเมียลส่วนของ $T(s)$

3.5 การออกแบบวงจรของความถี่แบบแอกทีฟดิสทริบิวต์ออาซี

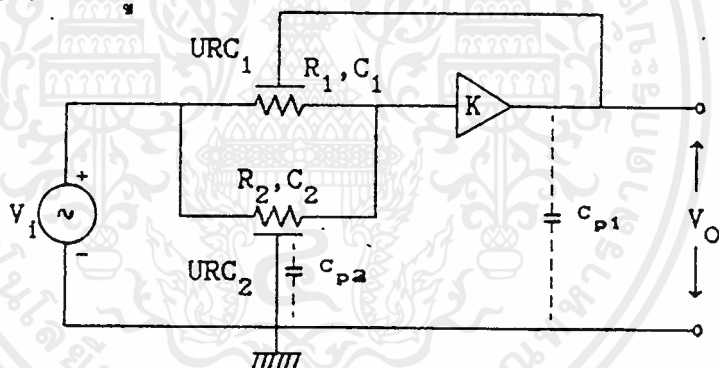
ตัว Uniform Distributed RC หรือใช้ชื่อย่อว่า URC ที่จะใช้ในการออกแบบซึ่งเขียนอยู่ในรูปพารามิเตอร์ความนำ (Admittance Parameter) จะได้

$$[Y] = \frac{\sqrt{SRC}}{R \sinh \sqrt{SRC}} \begin{pmatrix} \cosh \sqrt{SRC} & -1 \\ -1 & \cosh \sqrt{SRC} \end{pmatrix} \dots (3.8)$$

เมื่อ R และ C เป็นผลรวมของค่าความต้านทานทั้งหมดและค่าประจุทั้งหมดของ URC

S เป็นตัวแปรความถี่เชิงซ้อน (Complex Frequency Variable)

ในการออกแบบวงจรของความถี่ที่เราจะใช้ Op-Amp หรือตัว Amplifier อื่น ๆ เช่น Video Amp จำนวน 1 ตัว ร่วมกับ Uniform Distributed RC จำนวน 2 ตัว โดยในบทนี้จะเสนองานวงจรของความถี่ต่ำ และ วงจรของความถี่สูงโดยวงจรของความถี่ต่ำสามารถเขียนเป็นวงจรได้ดังแสดงในรูปที่ 3.6



รูปที่ 3.6 วงจรของความถี่ต่ำแบบแอกทีฟดิสทริบิวต์ออาซี

เมื่อ K เป็น Amplifier ที่มีอัตราขยายแรงดัน (Positive Voltage Gain) เท่ากับ K และกำหนดให้ $P_1 = \sqrt{SR_1C_1}$, $P_2 = \sqrt{SR_2C_2}$ จากวงจรในรูปที่ 3.6 เขียนเป็นทรานส์เฟอร์ฟังก์ชันได้คือ

$$T(P) = \frac{V_o}{V_i} = \frac{\frac{P_1}{R_1 \sinh P_1} + \frac{P_2}{R_2 \sinh P_2}}{\frac{1}{K} \left(\frac{P_1 \cosh P_1}{R_1 \sinh P_1} + \frac{P_2 \cosh P_2}{R_2 \sinh P_2} \right) - \frac{P_1 (\cosh P_1 - 1)}{R_1 \sinh P_1}} \dots (3.9)$$

ถ้ากำหนดให้ $P = P_1 = P_2$ หรือ $R_1 C_1 = R_2 C_2$ แล้ว แทนค่าลงในสมการ (3.9) จะได้

$$T(P) = \frac{1 + \frac{R_1}{R_2}}{\frac{1}{K} \left(1 + \frac{R_1}{R_2} - K \right) \cosh P + 1} \quad \dots (3.10)$$

จากสมการ (3.10) กำหนดให้อัตราขยายมีค่า $K = 1$ (Unity Gain Amplifier) จะได้
ทรานส์เฟอ์ฟังก์ชันของวงจรตรงความถี่ต่ำดังสมการ (3.11)

$$T(P) = \frac{\alpha + 1}{\cosh P + \alpha} \quad \dots (3.11)$$

โดยที่ $\alpha = R_2/R_1$

จากสมการ (3.10) ถ้าให้ $K = \alpha$ (Infinity Gain Amplifier) จะได้

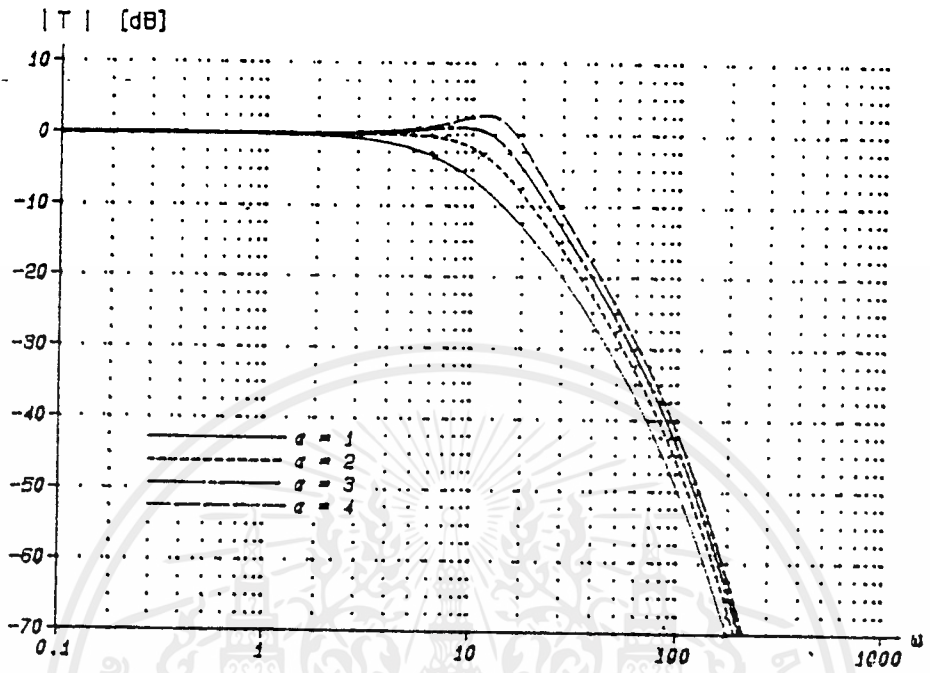
$$T(P) = \frac{\alpha + 1}{-\alpha \cosh P + \alpha} \quad \dots (3.12)$$

วงจรตรงความถี่ต่ำแบบแอกทีฟดิสรีบูท้อที่ตามในรูปที่ 3.6 ถ้าเราพิจารณา C_{p1} และ C_{p2} ซึ่งเป็น Parasitic Capacitors ระหว่างแผ่นล่าง (Conductor Plate) และ Substrate ของ URC ดังที่แสดงในรูปที่ 2.2 โดยที่ Parasitic Capacitance ของแผ่นด้านบน (Top Plate) มีค่าน้อยมากสามารถตัดทิ้งได้ ค่า Parasitic Capacitor C_{p1} จะอยู่ระหว่างเอาต์พุทของออปแอมป์กับกราวด์ (โดยที่เอาต์พุทอิมพีแดนซ์ของออปแอมป์มีค่าเป็นศูนย์) ส่วน C_{p2} จะอยู่ระหว่างกราวด์ทั้งสองข้าง และนั่นจึงไม่มีผลต่อคุณลักษณะของวงจร (Circuit Characteristics)

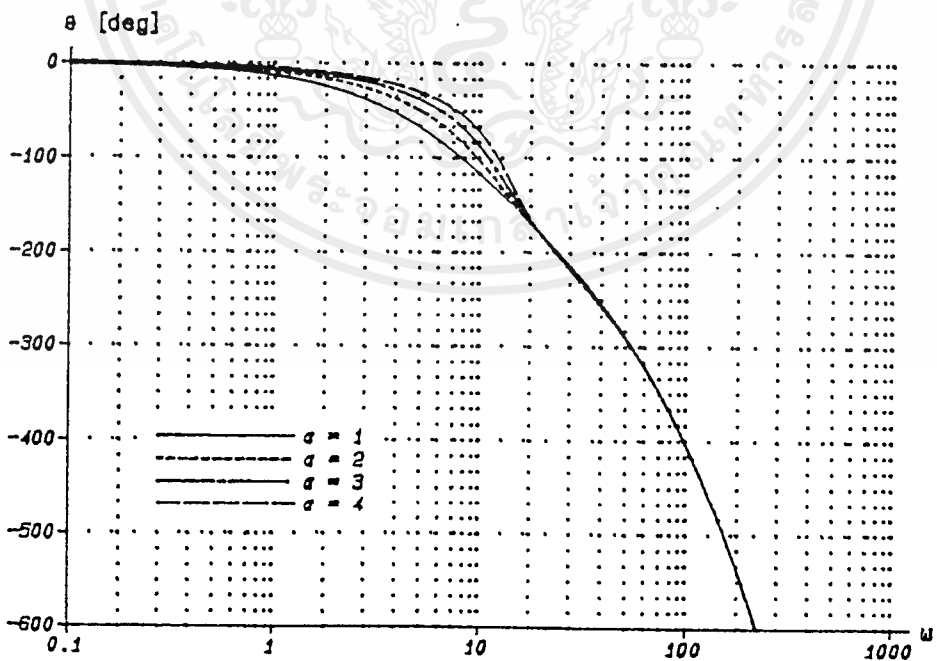
ผลตอบสนองทางความถี่ของสมการ (3.11) สามารถคำนวณได้จากความสัมพันธ์ดังนี้

$$\begin{aligned} \cosh P &= \cosh (t+jt) \\ \text{โดยที่ } P &= \sqrt{j\omega RC} \Big|_{RC=1} = \sqrt{(\omega/2)} + j\sqrt{(\omega/2)} = t + jt \quad \text{เมื่อ } t = \sqrt{(\omega/2)} \\ \text{จะได้ } \cosh P &= \cosh t \cdot \cos t + j \sinh t \cdot \sin t \quad \dots (3.13) \end{aligned}$$

จากทรานส์เฟอ์ฟังก์ชันในสมการ (3.11) ถ้ากำหนดให้ $\alpha = 1, 2, 3$ และ 4 เมื่อนำไปหาผลตอบสนองทางขนาดต่อความถี่ (Magnitude Response) จะได้ดังแสดงในรูปที่ 3.7 และได้ค่าจุดคัทออฟ -3 dB ที่ $\omega = 6.287, 11.51, 15.927$ และ 19.156 ตามลำดับ จากเอกสารนี้รูปที่ 3.7 จะเห็นได้ว่าเมื่อปรับค่าพารามิเตอร์ α จะทำให้ผลตอบสนองทางขนาดเปลี่ยนไป คือไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

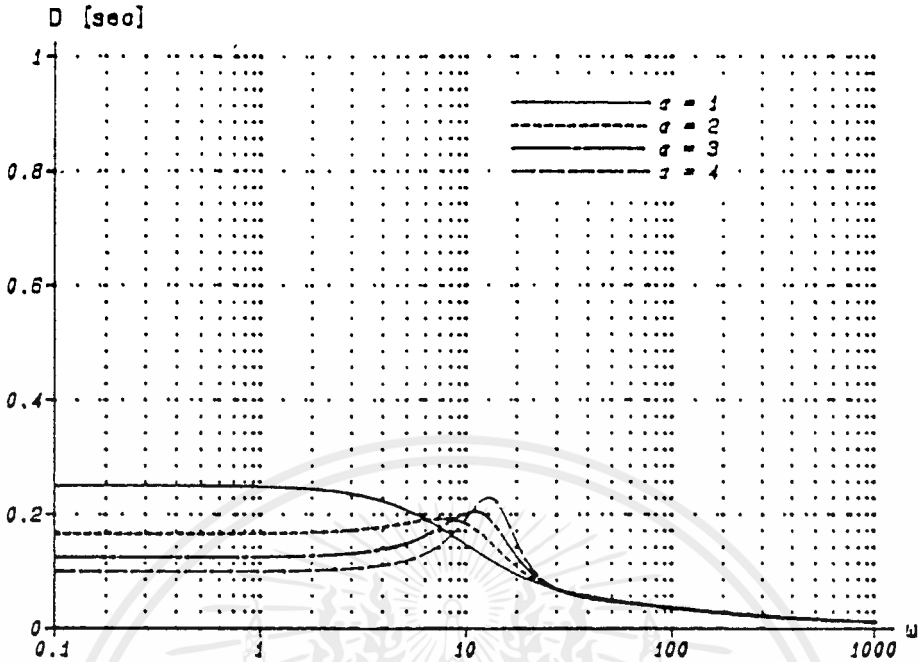


รูปที่ 3.7 แสดงผลตอบสนองทางขนาด



รูปที่ 3.8 แสดงผลตอบสนองทางเฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 แสดงผลตอบสนองของกรุปดีเลย์

เมื่อค่า α มีค่าเพิ่มขึ้น จะทำให้ผลตอบสนองทางขนาดที่ขึ้นในย่านคัทออฟ (Cut-off) และเมื่อ α มีค่ามากกว่า 2 จะทำให้ในช่วงความถี่ผ่าน (Passband) เกิด Ripple ซึ่งคล้าย Chebyshev หรือ Ultraspherical ฟิลเตอร์ จากทรานส์เฟอ์ฟังก์ชันในสมการ (3.11) เมื่อนำไปหาค่าผลตอบสนองทางเฟส (Phase Response) และผลตอบสนองของกรุปดีเลย์ (Group Delay) จะได้ดังแสดงในรูปที่ 3.8 และรูปที่ 3.9 ตามลำดับ ซึ่งจากรูปจะเห็นได้ว่าที่ α มีค่าต่ำค่าผลตอบสนองของกรุปดีเลย์จะมีลักษณะราบเรียบ

ผลตอบสนองทางความถี่ของวงจรต่าง ๆ นั้นอาจมีการเปลี่ยนแปลง ซึ่งขึ้นอยู่กับอุปกรณ์ต่าง ๆ ที่ใช้เมื่อค่าความไว (Sensitivity) ของอุปกรณ์มีค่าสูงจะต้องเลือกใช้อุปกรณ์ที่มีความคลาดเคลื่อนน้อย และนั่นวงจรที่จริงจังควรมีค่าความไวของอุปกรณ์ต่ำ จากวงจรในรูปที่ 3.6 เราสามารถหาค่าความไวของอิมพีแดนซ์แต่ละตัว R_1, C_1, R_2, C_2 โดยกำหนดให้อัตราขยาย K มีค่าเท่ากับหนึ่ง จากทรานส์เฟอ์ฟังก์ชันในสมการ (3.9) จะได้

$$T(S) = \frac{\sinh\sqrt{SR_2C_2} + \sqrt{\frac{R_1C_2}{R_2C_1}} \sinh\sqrt{SR_1C_1}}{\sinh\sqrt{SR_2C_2} + \sqrt{\frac{R_1C_2}{R_2C_1}} \sinh\sqrt{SR_1C_1} \cosh\sqrt{SR_2C_2}} \dots (3.14)$$

จากฟังก์ชันหาเช่นชีวิตี้ในสมการ (3.7) โดยการแทน x_1 ด้วย R_1 และนั่นจากสมการ (3.14) เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ใจได้
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$S_{R1}^{\tau(s)} = \frac{1}{2} \left[\frac{\frac{R_1 P_2}{R_2 P_1} (P_1 \cosh P_1 + \sinh P_1)}{\sinh P_2 + \frac{R_1 P_2}{R_2 P_1} \sinh P_1} - \frac{\frac{R_1 P_2}{R_2 P_1} \cosh P_2 (P_1 \cosh P_1 + \sinh P_1)}{\sinh P_2 + \frac{R_1 P_2}{R_2 P_1} \sinh P_1 \cosh P_2} \right] \dots (3.15)$$

จากสมการ (3.15) กำหนดให้ $R_1 C_1 = R_2 C_2$ และ $\alpha = R_2/R_1$ จะได้ค่าเช่นขั้วตีตัวของอิลิเมนต์ R_1 คือ

$$S_{R1}^{\tau(s)} = \frac{1}{2} \left(\frac{P \coth P + 1}{\alpha + 1} - \frac{\cosh P (P \coth P + 1)}{\alpha + \cosh P} \right) \dots (3.16)$$

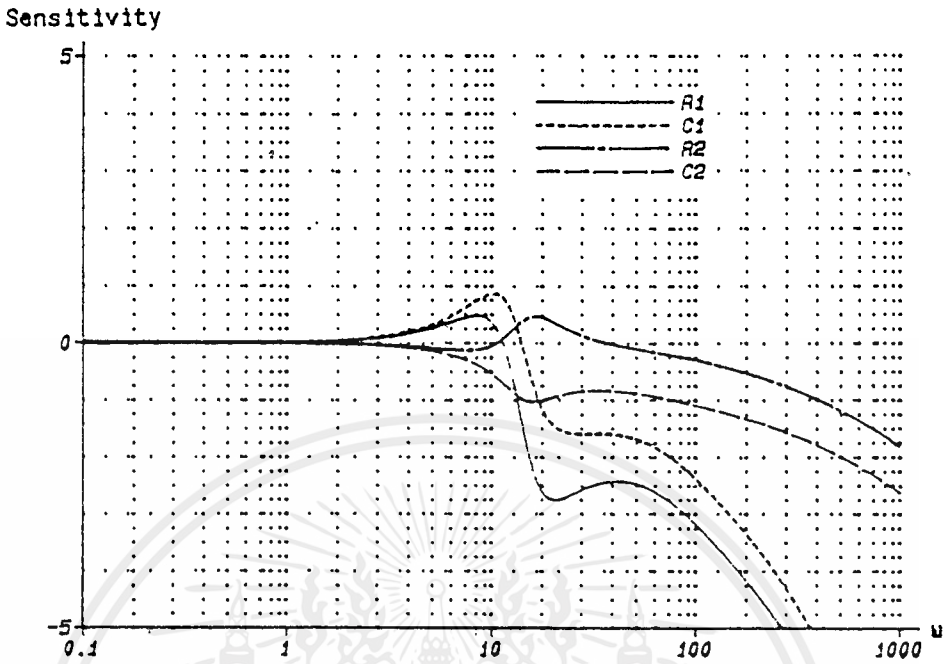
ส่วนค่าเช่นขั้วตีตัวของอิลิเมนต์ C_1, R_2, C_2 ก็สามารถหาได้ในกรณีเดียวกัน ซึ่งจากสมการ (3.15) จะได้

$$S_{C1}^{\tau(s)} = \frac{1}{2} \left(\frac{P \coth P - 1}{\alpha + 1} - \frac{\cosh P (P \coth P - 1)}{\alpha + \cosh P} \right) \dots (3.17)$$

$$S_{R2}^{\tau(s)} = \frac{1}{2} \left(\frac{\alpha P \coth P - 1}{\alpha + 1} - \frac{\alpha P \coth P + P \sinh P - \cosh P}{\alpha + \cosh P} \right) \dots (3.18)$$

$$S_{C2}^{\tau(s)} = \frac{1}{2} \left(\frac{\alpha P \coth P + 1}{\alpha + 1} - \frac{\alpha P \coth P + P \sinh P + \cosh P}{\alpha + \cosh P} \right) \dots (3.19)$$

จากสมการ (3.16) - (3.19) ถ้าสมมติให้ $\alpha = 4$ จะได้ค่าเช่นขั้วตีตัวของอิลิเมนต์ R_1, C_1, R_2, C_2 แสดงดังในรูปที่ 3.10 ซึ่งจากรูปจะเห็นได้ว่าช่วงความถี่ต่ำผ่านสัญญาณ (Passband) ค่าเช่นขั้วตีที่ได้จะมีค่าต่ำ โดยที่ $\omega = 0$ ค่าเช่นขั้วตีจะมีค่าเท่ากับศูนย์ และที่ช่วงความถี่สูงผ่านหยุดสัญญาณ (Stopband) ค่าเช่นขั้วตีของ R_2, C_2 จะมีค่าต่ำกว่าของ R_1, C_1 ค่าเช่นขั้วตีของอิลิเมนต์ K ในวงจรรูปที่ 3.6 ซึ่งเป็น Amplifier ที่มีอัตราขยายแรงดันเท่ากับ K สามารถหาได้จากทรานส์เฟอร์ฟังก์ชันในสมการ (3.10) โดยการแทนค่า X_1 ด้วย K ในสมการ (3.7) จะได้



รูปที่ 3.10 แสดง Sensitivity ของอิลิเมนต์ R_1, C_1, R_2, C_2

$$S_K^{T(s)} = \frac{1}{K} \left(\frac{\left(1 + \frac{R_1}{R_2}\right) \cosh P}{\frac{1}{K} \left(1 + \frac{R_1}{R_2} - K\right) \cosh P + 1} \right) \quad \dots (3.20)$$

จากสมการ (3.20) ถ้าให้อัตราขยาย $K = 1$ และ $\alpha = R_2/R_1$ จะได้ค่าเซ็นซิวิตีของอิลิเมนต์ K คือ

$$S_K^{T(s)} = \frac{(\alpha + 1) \cosh P}{\cosh P + \alpha} \quad \dots (3.21)$$

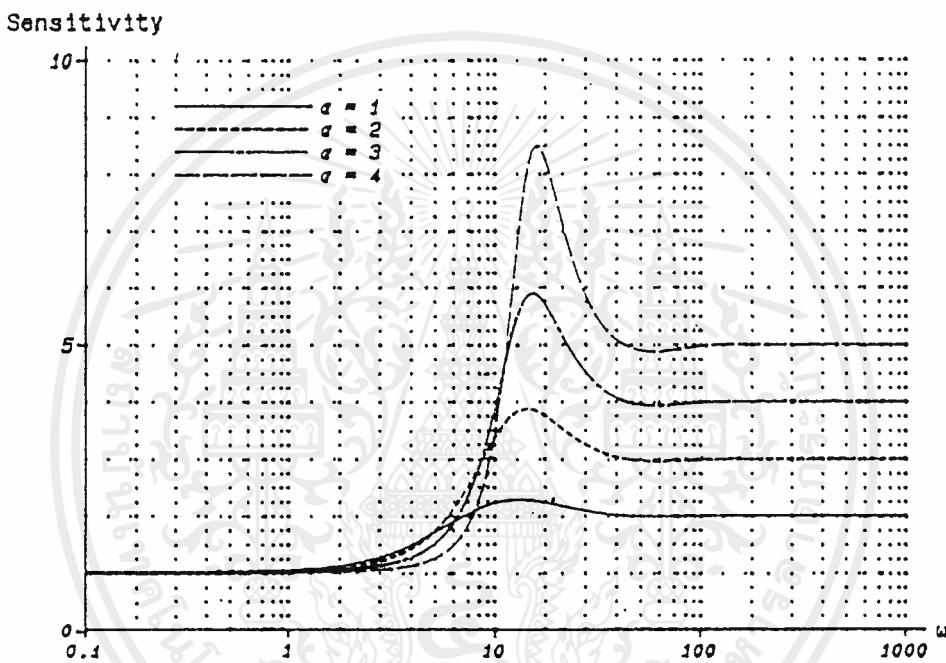
ค่าเซ็นซิวิตีของอิลิเมนต์ K ที่ $\omega = 0$ จะได้ $\cosh P = 1$ เมื่อแทนค่าลงในสมการ (3.12) จะได้

$$S_K^{T(s)} = 1$$

และที่ ω เท่ากับ infinity จะได้

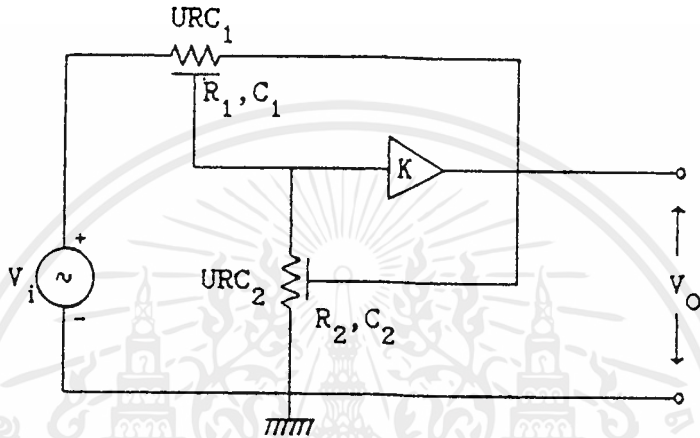
$$\frac{T(S)}{S_K} = \alpha + 1$$

จากสมการ (3.21) ถ้าสมมติให้ $\alpha = 1, 2, 3, 4$ จะได้ค่าเซ็นซิวิตีของอิลิเมนต์ K ดังแสดงในรูปที่ 3.11 ซึ่งจากรูปจะเห็นได้ว่าในช่วงความถี่ต่ำผ่านสัญญาณ (Passband) ค่าเซ็นซิวิตีของ K จะมีค่าต่ำคือมีค่าเท่ากับหนึ่ง และในช่วงความถี่สูงย่านหยุดสัญญาณ (Stopband) ค่าเซ็นซิวิตีของ K มีค่าเท่ากับ $\alpha + 1$



รูปที่ 3.11 แสดง Sensitivity ของอิลิเมนต์ K

สำหรับวงจรรองความถี่สูงแบบแอกทีฟที่ผลิตทวีปัวท์อาซีสามารถเขียนเป็นวงจรได้ดังแสดงในรูปที่ 3.12 ถ้าให้ K เป็น Amplifier ที่มีอัตราขยายแรงดัน (Positive Voltage Gain) เท่ากับ K และกำหนดให้ $P_1 = \sqrt{SR_1C_1}$, $P_2 = \sqrt{SR_2C_2}$ จากวงจรในรูปที่ 3.12 เขียนเป็นทรานส์เฟอร์ฟังก์ชันได้คือ



รูปที่ 3.12 วงจรรองความถี่สูงแบบแอกทีฟที่ผลิตทวีปัวท์อาซี

$$T(P) = \frac{P_1 (\cosh P_1 - 1)}{R_1 \sinh P_1} \dots (3.22)$$

$$\frac{1}{K} \left(\frac{2P_1 (\cosh P_1 - 1)}{R_1 \sinh P_1} + \frac{P_2 \cosh P_2}{R_2 \sinh P_2} \right) - \frac{P_1 (\cosh P_1 - 1)}{R_1 \sinh P_1} - \frac{P_2 (\cosh P_2 - 1)}{R_2 \sinh P_2}$$

ถ้ากำหนดให้ $P = P_1 = P_2$ หรือ $R_1 C_1 = R_2 C_2$ แล้วแทนค่าลงในสมการ (3.22) จะได้

$$T(P) = \frac{\cosh P - 1}{\left(\frac{2}{K} + \frac{R_1}{KR_2} - \frac{R_1}{R_2} - 1 \right) \cosh P - \frac{2}{K} + \frac{R_1}{R_2} + 1} \dots (3.23)$$

จากสมการ (3.23) กำหนดให้อัตราขยายมีค่า $K = 1$ (Unity Gain Amplifier) จะได้ทรานส์เฟอร์ฟังก์ชันของวงจรรองความถี่สูงดังสมการ (3.24)

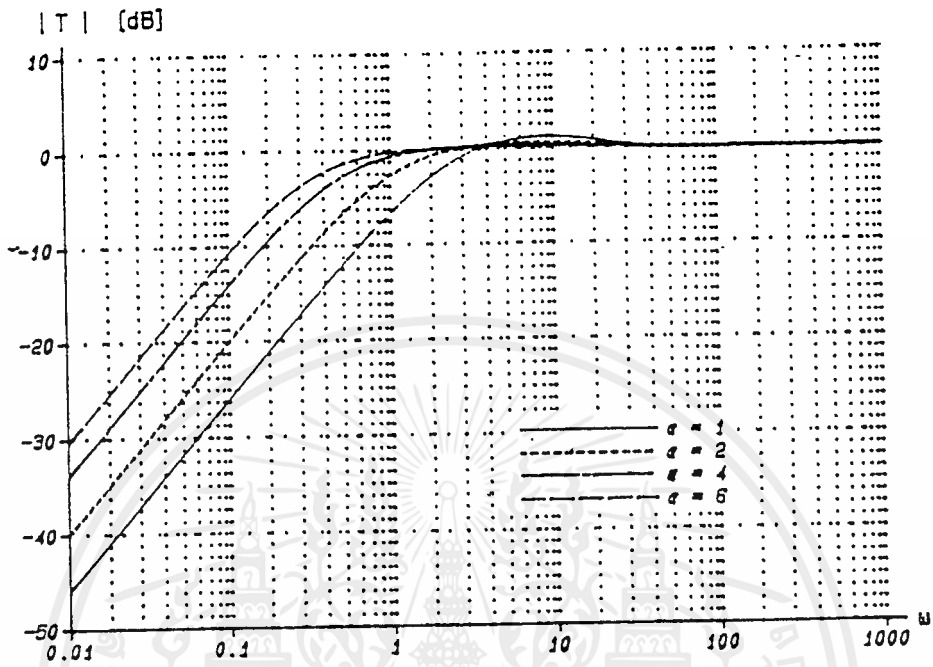
$$T(P) = \frac{\alpha \cosh P - \alpha}{\alpha \cosh P - \alpha + 1} \dots (3.24)$$

โดยที่ $\alpha = R_2/R_1$

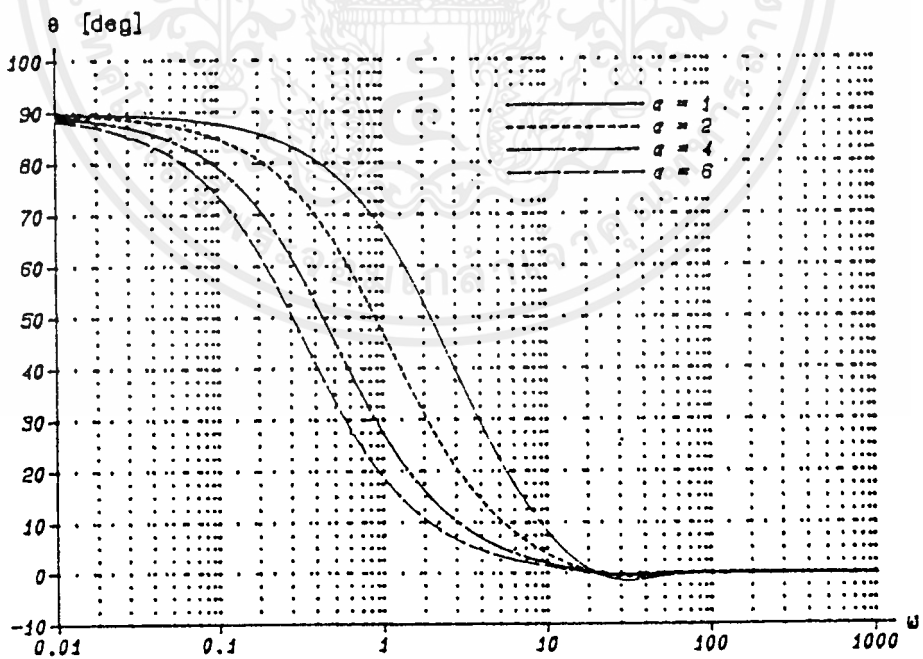
จากสมการ (3.23) ถ้าให้ $K = \alpha$ (Infinity Gain Amplifier) จะได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$T(P) \approx \frac{-\alpha \cosh P + \alpha}{(\alpha+1) \cosh P - \alpha - 1} \dots (3.25)$$

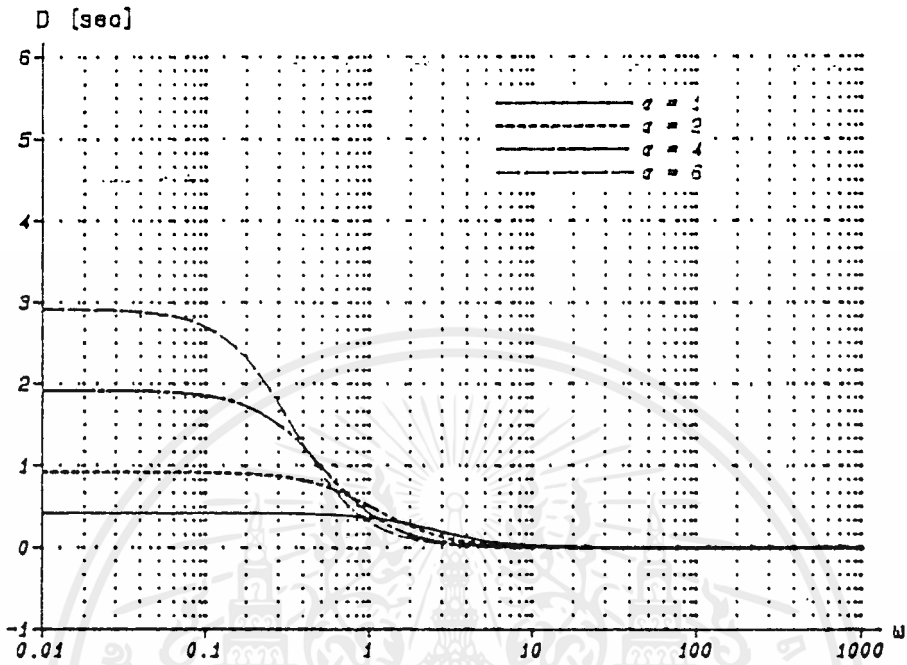


รูปที่ 3.13 แสดงผลตอบสนองทางขนาด



รูปที่ 3.14 แสดงผลตอบสนองทางเฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 แสดงผลตอบสนองของกรุปดีเลย์

จากทรานส์เฟอ์ฟังก์ชันในสมการ (3.24) ถ้ากำหนดให้ $\alpha = 1, 2, 4$ และ 6 เมื่อนำไปหาผลตอบสนองขนาดต่อความถี่ (Magnitude Response) จะได้ดังแสดงในรูปที่ 3.13 และได้ค่าจุดคutoff -3 dB ที่ $\omega = 1.73, 0.925, 0.48$ และ 0.324 ตามลำดับ ซึ่งจากรูปจะเห็นได้ว่าเมื่อปรับค่าพารามิเตอร์ α จะทำให้ผลตอบสนองทางขนาดเปลี่ยนแปลงคือ เมื่อ α มีค่าน้อยจะทำให้ได้ผลตอบสนองทางขนาดที่ขึ้นในย่านคutoff และเมื่อนำทรานส์เฟอ์ฟังก์ชันในสมการ (3.24) ไปหาค่าผลตอบสนองทางเฟส (Phase Response) และผลตอบสนองของกรุปดีเลย์ (Group Delay) จะได้ดังแสดงในรูปที่ 3.14 และรูปที่ 3.15 ตามลำดับ ซึ่งจากรูปจะเห็นได้ว่าในช่วงความถี่สูงย่านผ่านสัญญาณ (Passband) ค่าผลตอบสนองของกรุปดีเลย์ที่ได้จะมีลักษณะราบเรียบ

จากวงจรมีรูปที่ 3.12 เราจะหาค่าเช่นขั้วตีตัวของอิลิเมนต์ R_1, C_1, R_2, C_2 โดยอัตราขยายมีค่า $K = 1$ ฉะนั้นทรานส์เฟอร์ฟังก์ชันในสมการ (3.22) จะได้

$$T(S) = \frac{\sinh\sqrt{SR_2C_2} (\cosh\sqrt{SR_1C_1}-1)}{\sinh\sqrt{SR_2C_2} (\cosh\sqrt{SR_1C_1}-1) + \sqrt{\frac{R_1C_2}{R_2C_1}} \sinh\sqrt{SR_1C_1}} \quad \dots(3.26)$$

เมื่อหาเช่นขั้วตีตัวในสมการ (3.26) แล้วกำหนดให้ $R_1C_1 = R_2C_2$ และ $\alpha = R_2/R_1$ จะได้ค่าเช่นขั้วตีตัวของอิลิเมนต์แต่ละตัวดังนี้

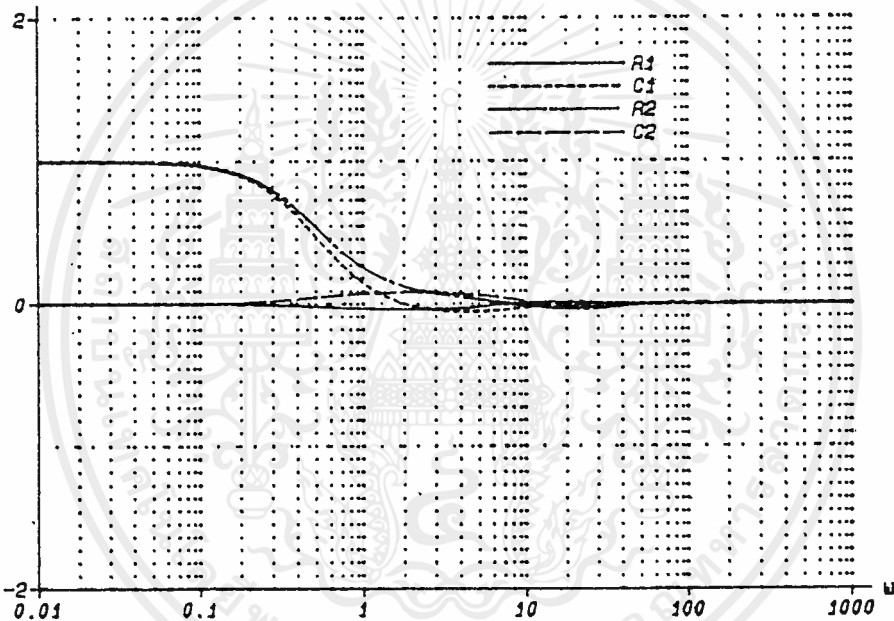
$$S_{R1} = -\frac{1}{2} \left(\frac{P \sinh P}{\cosh P - 1} - \frac{\alpha P \sinh P + P \coth P + 1}{\alpha \cosh P - \alpha + 1} \right) \quad \dots(3.27)$$

$$S_{C1} = -\frac{1}{2} \left(\frac{P \sinh P}{\cosh P - 1} - \frac{\alpha P \sinh P + P \coth P - 1}{\alpha \cosh P - \alpha + 1} \right) \quad \dots(3.28)$$

$$S_{R2} = -\frac{1}{2} \left(P \coth P - \frac{\alpha P \coth P (\cosh P - 1) - 1}{\alpha \cosh P - \alpha + 1} \right) \quad \dots(3.29)$$

$$\frac{T(s)}{S_{c2}} = \frac{1}{2} \left(P \coth P - \frac{\alpha P \coth P (\cosh P - 1) + 1}{\alpha \cosh P - \alpha + 1} \right) \quad \dots (3.30)$$

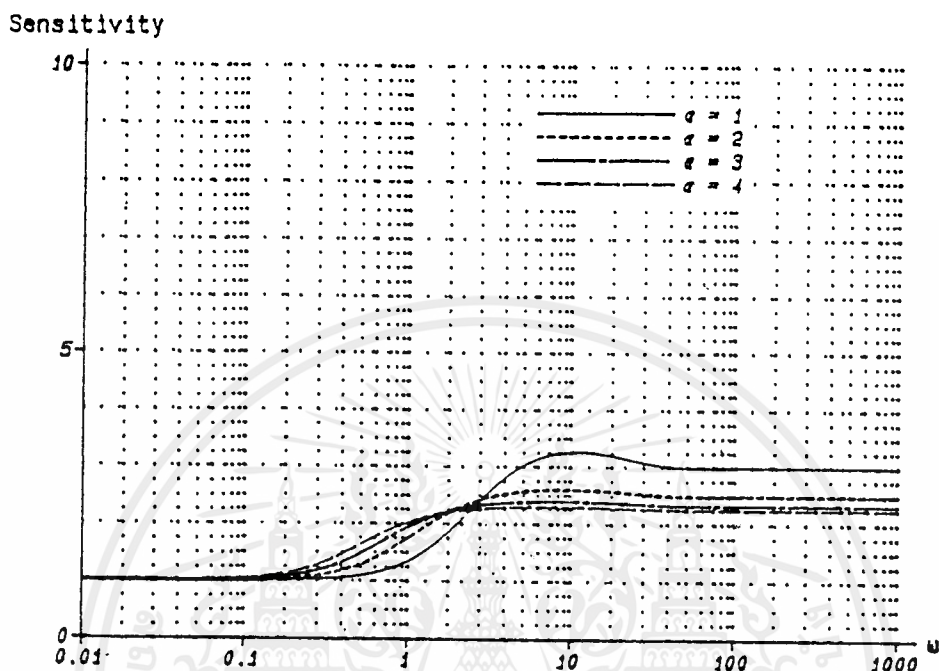
จากสมการ (3.27)-(3.30) ถ้าสมมติให้ $\alpha = 4$ จะได้เส้นชีวิตวิบัติของอิลิเมนต์ R_1, C_1, R_2, C_2 แสดงดังในรูปที่ 3.16 ซึ่งจากรูปจะเห็นได้ว่าช่วงความถี่สูงผ่านสัญญาณ (Passband) ค่าเส้นชีวิตวิบัติที่ได้จะมีค่าต่ำมากโดยที่ $\omega = \text{infinity}$ ค่าเส้นชีวิตวิบัติจะมีค่าเท่ากับศูนย์ และที่ช่วงความถี่ต่ำผ่านหยุดสัญญาณ (Stopband) ค่าเส้นชีวิตวิบัติของ R_1, C_2 จะมีค่าต่ำกว่า C_1, R_2 โดยที่ $\omega = 0$ ค่าเส้นชีวิตวิบัติของ R_1, C_2 จะมีค่าเท่ากับศูนย์และ C_1, R_2 จะมีค่าเท่ากับหนึ่ง Sensitivity



รูปที่ 3.16 แสดง Sensitivity ของอิลิเมนต์ R_1, C_1, R_2, C_2

ค่าเส้นชีวิตวิบัติของอิลิเมนต์ K ในวงจรรูปที่ 3.12 เมื่อ K เป็น Amplifier ที่มีอัตราขยายแรงดันเท่ากับ K หาได้จากทรานส์เฟอร์ฟังก์ชันในสมการ (3.23) แล้วกำหนดให้ $K = 1$ และ $\alpha = R_2/R_1$ จะได้ค่าเส้นชีวิตวิบัติของอิลิเมนต์ K คือ

$$\frac{T(s)}{S_K} = \frac{(2\alpha + 1) \cosh P - 2\alpha}{\alpha \cosh P - \alpha + 1} \quad \dots (3.31)$$



รูปที่ 3.17 แสดง Sensitivity ของอิลิเมนต์ K

จากสมการ (3.31) ถ้าสมมติให้ $\alpha = 1, 2, 3, 4$ จะได้ค่าเช่นชิตวิต^{สี่}ของอิลิเมนต์ K แสดงดังในรูปที่ 3.17 ซึ่งจากรูปจะเห็นได้ว่าในช่วงความถี่สูงย่านผ่านสัญญาณ (Passband) ค่าเช่นชิตวิต^{สี่}ของ K จะมีค่าเท่ากับ $(2\alpha+1)/\alpha$ และในช่วงความถี่ต่ำย่านหยุดสัญญาณ (Stopband) ที่ $\omega = 0$ ค่าเช่นชิตวิต^{สี่}ของ K จะมีค่าเท่ากับหนึ่ง

3.6 การทดสอบวงจรนอกที่พัลส์ทรีวิวก์อาซีเพื่อการใช้งาน

จากวงจรนอกที่พัลส์ทรีวิวก์อาซีที่เสนอในปริกฏยานพนธ์ฉบับนี้ คือ วงจรกรองความถี่ต่ำ และ วงจรกรองความถี่สูง ซึ่งได้มีการทดสอบในแต่ละแบบ เพื่อหาผลสรุปว่าวงจรนอกที่พัลส์ทรีวิวก์อาซี ดังที่เสนอไป สามารถนำไปใช้งานในเชิงปฏิบัติได้จริง โดยผลที่ได้นี้จะถูกนำไปเปรียบเทียบกับ ทางคอมพิวเตอร์ที่ทำการวิเคราะห์ในทางพีชคณิตในหัวข้อที่แล้ว และหาผลสรุปออกมาเพื่อเป็นแนวทางในการนำไปใช้งานต่อไป สำหรับการทดสอบวงจรนอกที่พัลส์ทรีวิวก์อาซีที่เสนอไปนี้ จะกำหนดให้อัตราขยายของแต่ละวงจรมีค่า $K = 1$ และกำหนดให้ $\alpha = 1, 2, 3, 4$ วงจรกรองความถี่ที่ ทดลองสมมติต้องการให้ได้จุดคัทออฟ -3 dB ที่ความถี่ 10 kHz ซึ่งทำได้โดยการสเกลทาง ความถี่ (Frequency Scaling) และเพื่อให้สามารถเลือกใช้อุปกรณ์ที่เหมาะสมในเชิงปฏิบัติ จะต้องทำการสเกลทางขนาด (Magnitude Scaling) อีกด้วย

การสเกลทางความถี่จะมีผลต่อการเปลี่ยนค่าของอุปกรณ์รีแอคทีฟ เช่น C และ L เท่านั้น เนื่องจากอิมพีแดนซ์จะมีค่าแปรตามความถี่ และเพื่อที่จะให้อิมพีแดนซ์คงที่การเปลี่ยนแปลงความถี่ ω จะต้องถูกชดเชยด้วยค่าที่สอดคล้องกันดังนี้

$$|Z_C| = \frac{1}{\omega C} = \frac{1}{K_f \omega (1/K_f) C} = \frac{1}{K_f \omega C_{new}}$$

$$|Z_L| = \omega L = K_f \omega \frac{1}{K_f} L = K_f \omega L_{new}$$

โดยที่ K_f เป็นตัวแปรของความถี่ที่สเกล

การสเกลทางขนาดคือ การเปลี่ยนแปลงค่าอิมพีแดนซ์ของอุปกรณ์ทุกตัวในวงจรด้วยตัวประกอบ K_m โดยที่อิมพีแดนซ์ของอุปกรณ์พาสซีฟมีค่าเป็นดังนี้

$$Z_R = R, \quad |Z_C| = \frac{1}{\omega C}, \quad |Z_L| = \omega L$$

ทำการสเกลทางขนาดโดยคูณด้วยค่า K_m จะได้

$$K_m Z_R = K_m R, \quad K_m |Z_C| = \frac{1}{\omega C / K_m}, \quad K_m |Z_L| = K_m \omega L$$

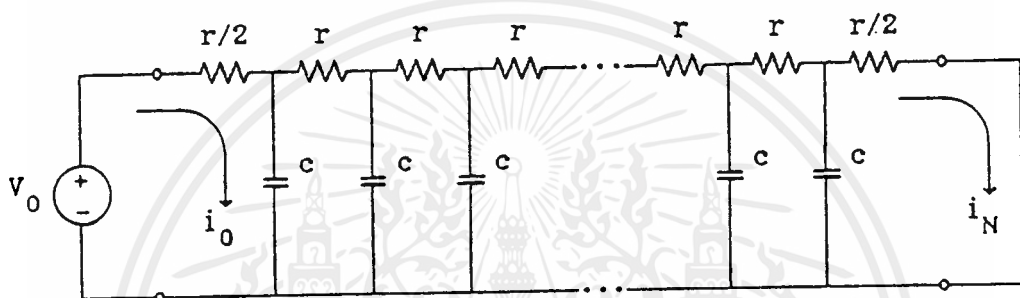
สรุปรวมผลของการสเกลทางความถี่และทางขนาดที่มีผลต่ออุปกรณ์พาสซีฟได้ดังนี้

$$C_{new} = \frac{1}{K_m K_f} C_{old} \quad \dots (3.32)$$

$$L_{\text{new}} = \frac{K_m}{K_f} L_{\text{old}} \quad \dots (3.33)$$

$$R_{\text{new}} = K_m R_{\text{old}} \quad \dots (3.34)$$

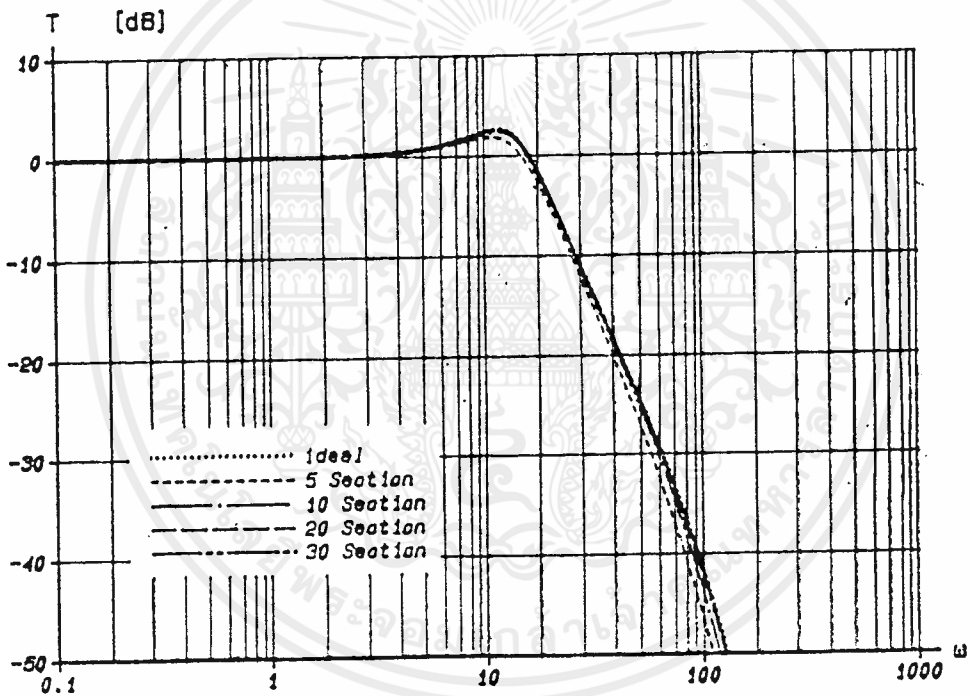
เนื่องจากตัว URC ที่ทดลองนี้ยังไม่สามารถหาได้ ฉะนั้นจึงใช้วิธีการประมาณ Uniform Distributed RC ด้วยเน็ตเวิร์กแบบ Lumped RC จากโครงสร้างของเน็ตเวิร์กแบบลัมด์ อิลิเมนต์ สามารถนำมาเขียนให้อยู่ในรูปของเน็ตเวิร์กแบบลัมด์อาซี ตามที่แสดงในรูปที่ 3.18



รูปที่ 3.18 แสดงการประมาณ Uniform Distributed RC เพื่อใช้ในการทดลอง

การประมาณดิสทริบิวท์อาซีด้วยเน็ตเวิร์กแบบลัมด์อาซี เมื่อพิจารณาถึงการวิเคราะห์ทาง โครงสร้างเน็ตเวิร์กแบบลัมด์อาซีแล้วจะได้ว่า ค่าราก (ค่า s ของทรานเฟอร์ฟังก์ชัน) ที่ ได้จะขึ้นอยู่กับการจำนวนเซคชั่น คือจะมีค่ารากทั้งหมดเท่ากับ $N+1$ โดยมีค่ารากเริ่มต้นอยู่ใกล้แกน $j\omega$ และค่ารากต่อมาจะห่างออกจากแกน $j\omega$ ไปตามลำดับ ซึ่งจะมีลักษณะเหมือนกับ ค่าราก ของดิสทริบิวท์อาซี เนื่องจากค่ารากของดิสทริบิวท์อาซีมีลักษณะกระจายออกเหมือนอนุกรมเลขคณิต ฉะนั้นค่ารากที่มีผลกับคุณลักษณะของวงจรจึงเป็นค่ารากช่วงเริ่มต้นซึ่งจะอยู่ใกล้กับแกน $j\omega$ และ การที่จะให้ได้รากช่วงเริ่มต้นจำนวนเท่าใดจะขึ้นอยู่กับการจำนวนเซคชั่นของเน็ตเวิร์กแบบลัมด์อาซีที่ นำมาต่อเข้าด้วยกัน จากวงจรร่องความถี่ต่ำแบบแอกทีฟดิสทริบิวท์อาซีที่เสนอไป ตามที่แสดงใน รูปที่ 3.6

เมื่อใช้วิธีประมาณ URC ด้วย Lumped RC โดยใช้น้ำจำนวนเซกชันเท่ากับ 5, 10, 20 และ 30 ตามลำดับ เมื่อนำไปคำนวณโดยการแทนค่าพารามิเตอร์แบบ Short-Circuit Admittance $[Y_{i,j}]$ เพื่อหาผลตอบสนองทางขนาดต่อความถี่จะได้ดังแสดงในรูปที่ 3.19 ฉะนั้นในการทดลองตัว URC จะใช้โครงสร้างของเน็ตเวิร์กแบบลิมิตค่าซีตามที่แสดงในรูปที่ 3.18 โดยต่อร่วมกัน 10 เซกชัน โดยที่ค่าความต้านทานทั้งหมด $R = rN$ และค่าตัวเก็บประจุทั้งหมด $C = cN$ เมื่อ N เป็นจำนวนเซกชัน



รูปที่ 3.19 แสดงผลตอบสนองทางขนาดที่ได้จากการประมาณ URC ด้วย Lumped RC

จากวงจรกรองความถี่ต่ำแบบนอกที่พหุสกรนิวัตที่อัตราที่เสนอ ตามรูปที่ 3.8 ที่กำหนดให้อัตราขยายเท่ากับหนึ่ง , $\alpha = R_2/R_1 = 1, 2, 3,$ และ 4 และต้องการให้เกิดความถี่คutoff -3 dB ที่ 10 KHz จะได้

กำหนดให้ $\alpha = 1 ;$

$$RC = \omega_{-3dB} / 2\pi K_p = 6.287/2\pi \times 10 \times 10^3 = 1.000 \times 10^{-4}$$

เมื่อทำการสเกลตามขนาดแล้วจะได้ค่าอุปกรณ์คือ

$$R_1 = 200 \text{ K} , C_1 = 500 \text{ pF} , R_2 = 200 \text{ K} , C_2 = 500 \text{ pF}$$

กำหนดให้ $\alpha = 2 ;$

$$RC = 11.51/2\pi \times 10 \times 10^3 = 1.831 \times 10^{-4}$$

$$R_1 = 916 \text{ K} , C_1 = 200 \text{ pF} , R_2 = 1.8 \text{ M} , C_2 = 100 \text{ pF}$$

กำหนดให้ $\alpha = 3 ;$

$$RC = 15.927/2\pi \times 10 \times 10^3 = 2.534 \times 10^{-4}$$

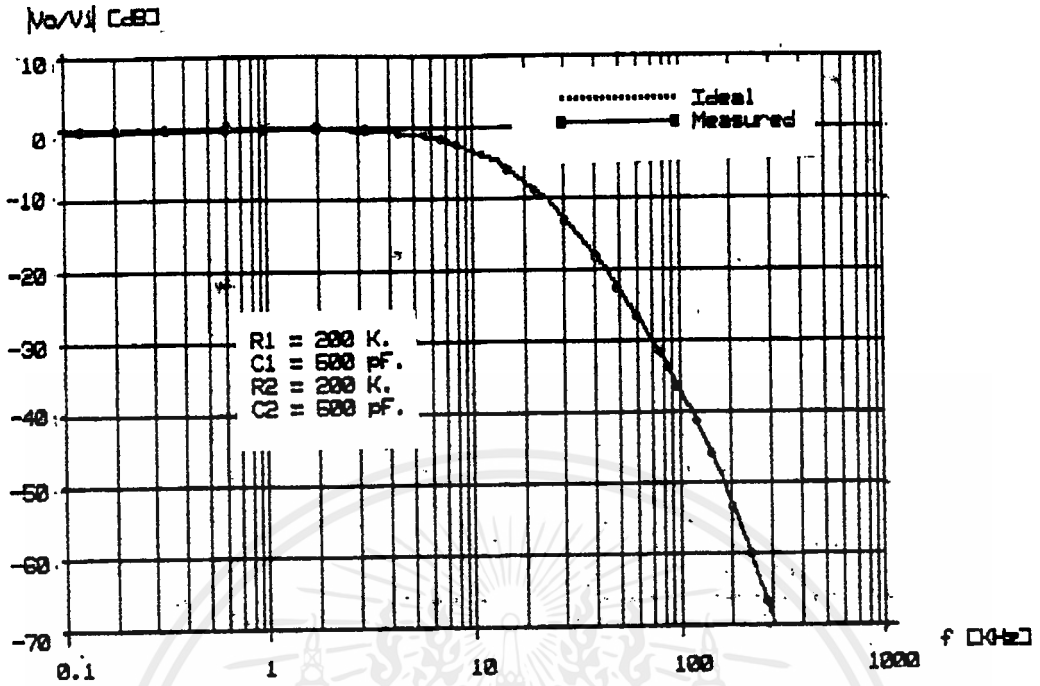
$$R_1 = 845 \text{ K} , C_1 = 300 \text{ pF} , R_2 = 2.5 \text{ M ohms} , C_2 = 100 \text{ pF}$$

กำหนดให้ $\alpha = 4 ;$

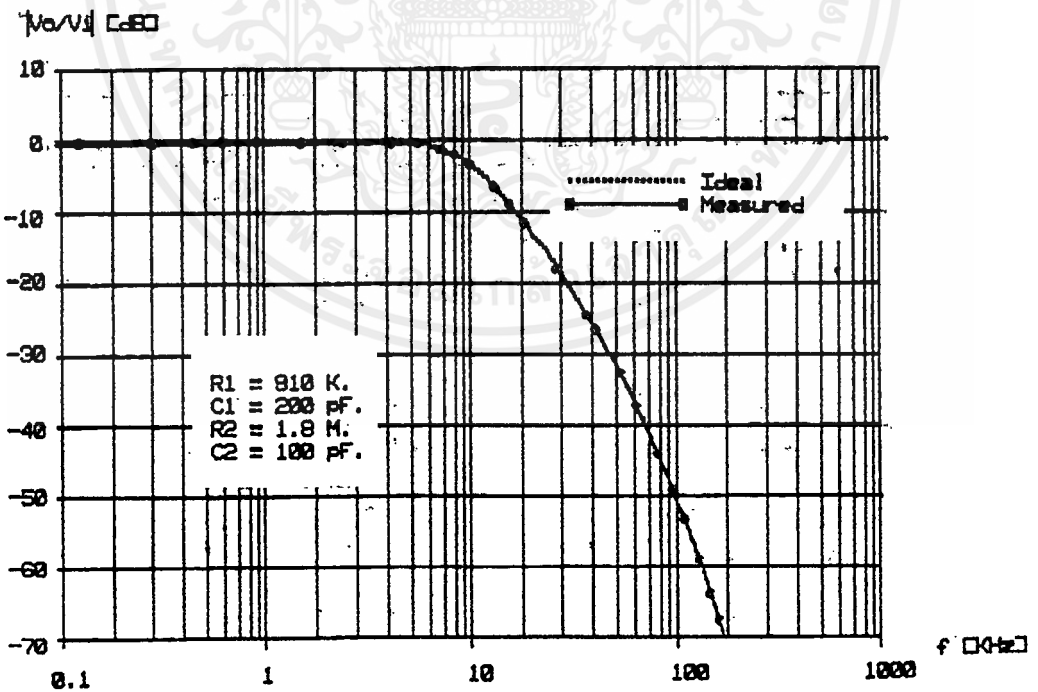
$$RC = 19.156/2\pi \times 10 \times 10^3 = 3.048 \times 10^{-4}$$

$$R_1 = 152.4 \text{ K} , C_1 = 2000 \text{ pF} , R_2 = 610 \text{ K} , C_2 = 500 \text{ pF}$$

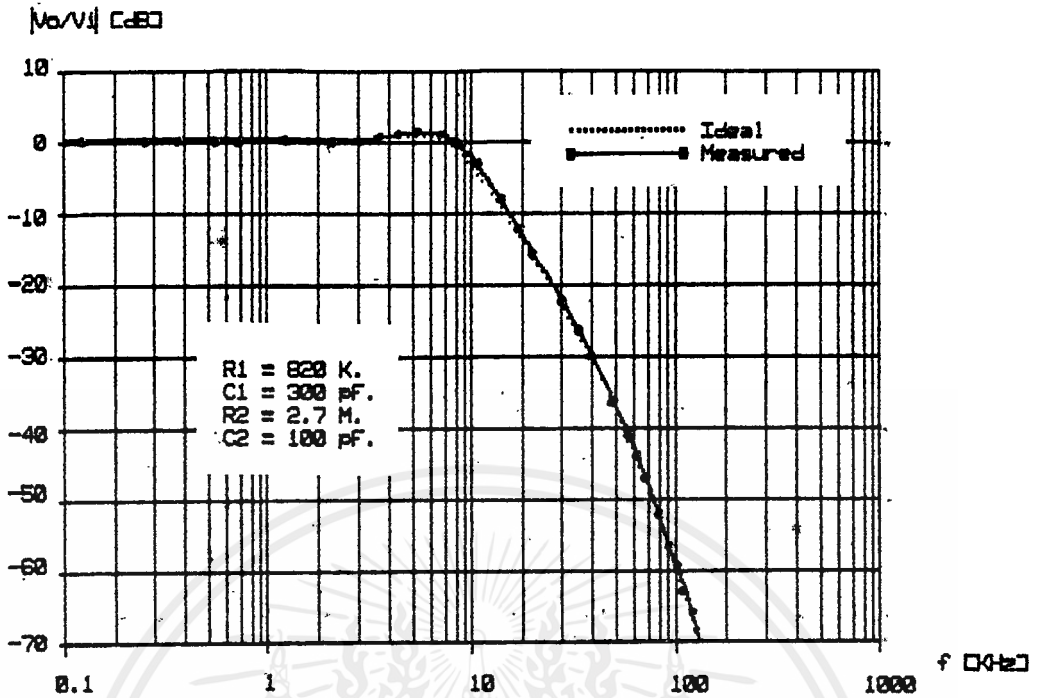
ผลตอบสนองทางขนาดต่อความถี่ที่ได้จากวงจรกรองความถี่ต่ำที่ $\alpha = 1, 2, 3,$ และ 4 แสดงดังในรูปที่ 3.20, 3.21, 3.22, และ 3.23 ตามลำดับ



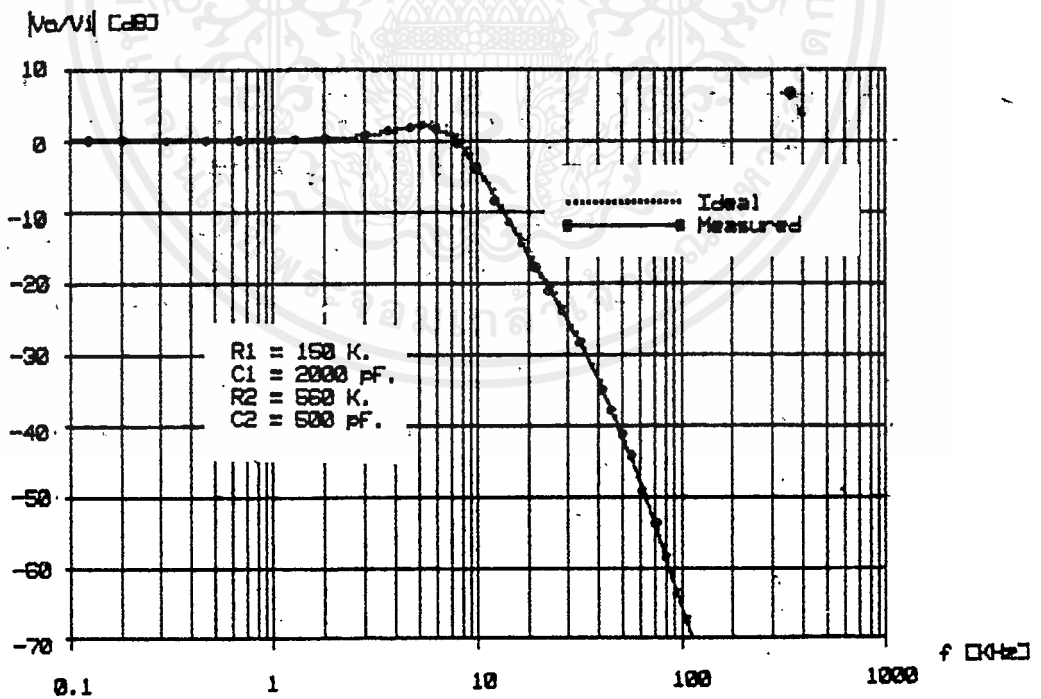
รูปที่ 3.20 แสดงผลตอบสนองทางขนาดต่อความถี่วัดได้จากวงจรของความถี่ต่าง $\alpha = 1$



รูปที่ 3.21 แสดงผลตอบสนองทางขนาดต่อความถี่วัดได้จากวงจรของความถี่ต่าง $\alpha = 2$
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับนักเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.22 แสดงผลตอบสนองทางขนาดต่อความถี่ที่ได้จากวงจรรองความถี่ต่ำที่ $\alpha = 3$



รูปที่ 3.23 แสดงการตอบสนองทางขนาดต่อความถี่ที่ได้จากวงจรรองความถี่ต่ำที่ $\alpha = 4$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับวงจรกรองความถี่สูงแบบแอกทีฟชนิดบิควิวด์อาร์ทีทีเฟสเซอร์ ตามรูปที่ 3.11 ที่กำหนดให้ อัตราขยายเท่ากับหนึ่ง , $\alpha = R_2/R_1 = 1, 2, 4$ และ 6 และต้องการให้เกิดความถี่คัทออฟ -3 dB ที่ 10 KHz จะได้

กำหนดให้ $\alpha = 1$;

$$RC = \omega_{-3dB} / 2\pi K_f = 1.73/2\pi \times 10 \times 10^3 = 2.753 \times 10^{-5}$$

เมื่อทำการสเกลทางขนาดแล้วจะได้ค่าอุปกรณ์คือ

$$R_1 = 55 \text{ K} , C_1 = 500 \text{ pF} , R_2 = 55 \text{ K} , C_2 = 500 \text{ pF}$$

กำหนดให้ $\alpha = 2$;

$$RC = 0.925/2\pi \times 10 \times 10^3 = 1.472 \times 10^{-5}$$

$$R_1 = 73.6 \text{ K} , C_1 = 200 \text{ pF} , R_2 = 147.2 \text{ K} , C_2 = 100 \text{ pF}$$

กำหนดให้ $\alpha = 4$;

$$RC = 0.48/2\pi \times 10 \times 10^3 = 7.639 \times 10^{-6}$$

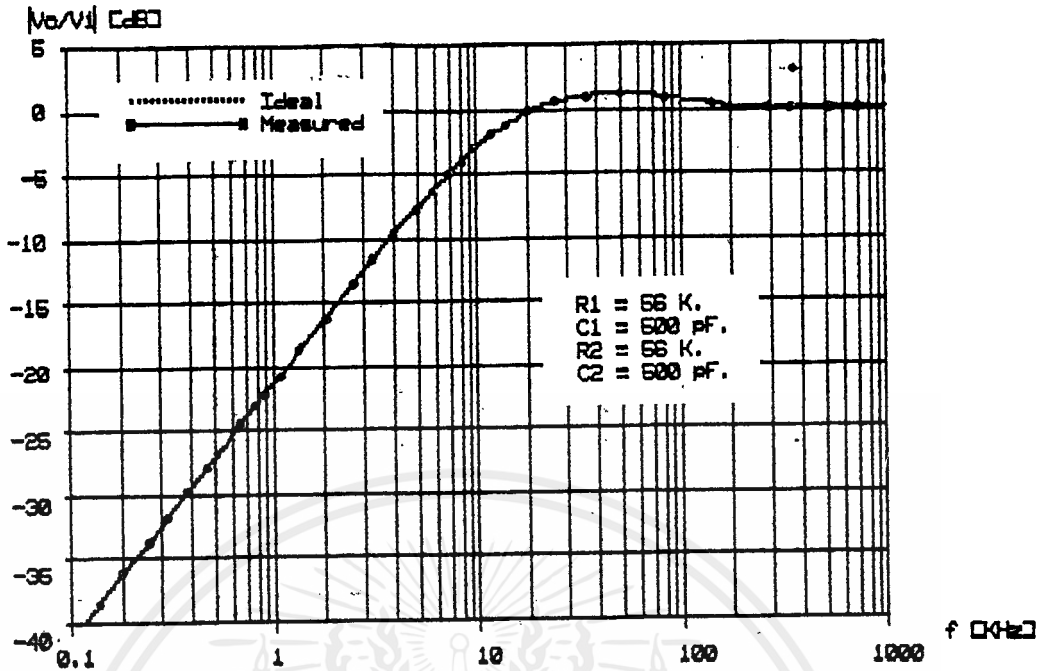
$$R_1 = 38.2 \text{ K} , C_1 = 200 \text{ pF} , R_2 = 152.78 \text{ K} , C_2 = 50 \text{ pF}$$

กำหนดให้ $\alpha = 6$;

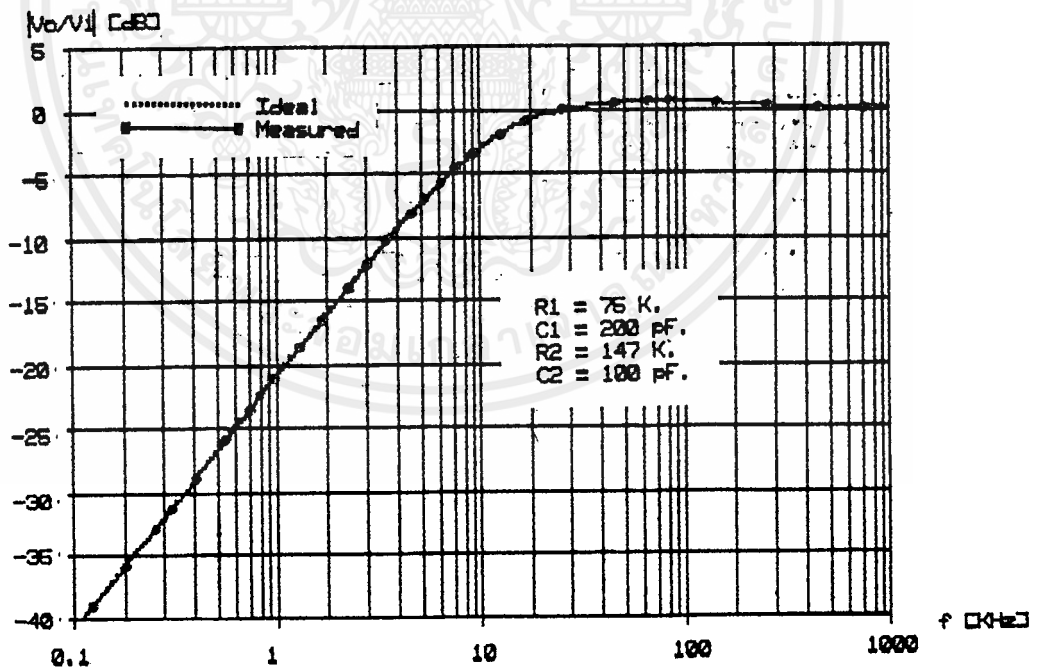
$$RC = 0.324/2\pi \times 10 \times 10^3 = 5.156 \times 10^{-6}$$

$$R_1 = 2.578 \text{ K} , C_1 = 2000 \text{ pF} , R_2 = 17188 , C_2 = 300 \text{ pF}$$

ผลตอบสนองทางขนาดต่อความถี่วัดได้จากวงจกรองความถี่ต่ำที่ $\alpha = 1, 2, 4$ และ 6 แสดงดัง ในรูปที่ 3.24, 3.25, 3.26, และ 3.27 ตามลำดับ

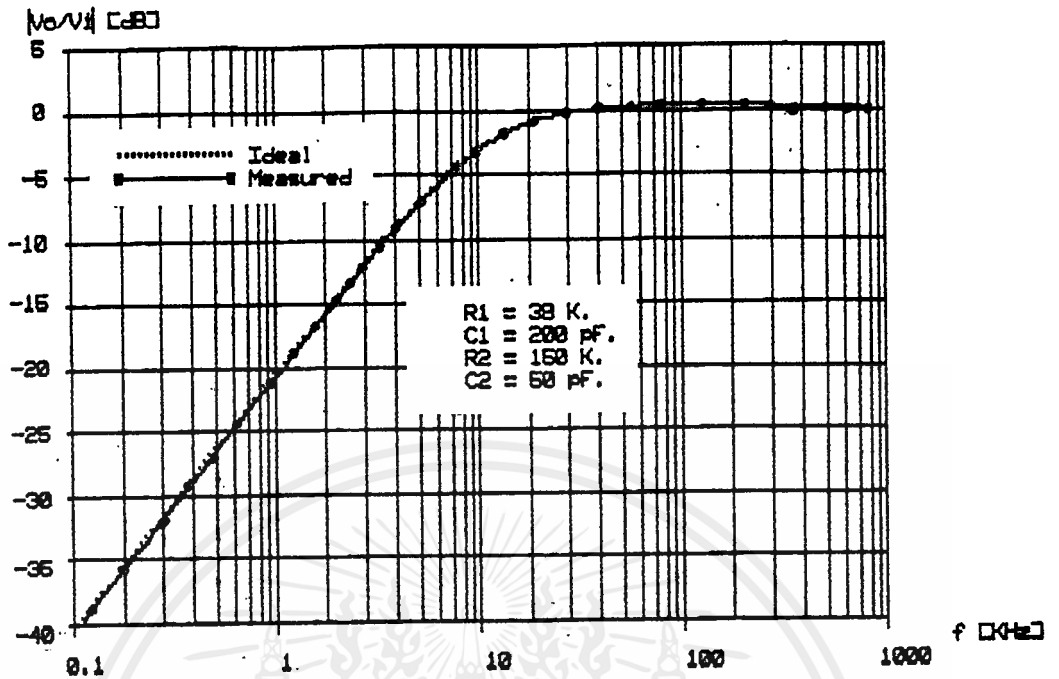


รูปที่ 3.24 แสดงผลตอบสนองทางขนาดต่อความถี่วัดได้จากวงจรความถี่สูงที่ $\alpha = 1$

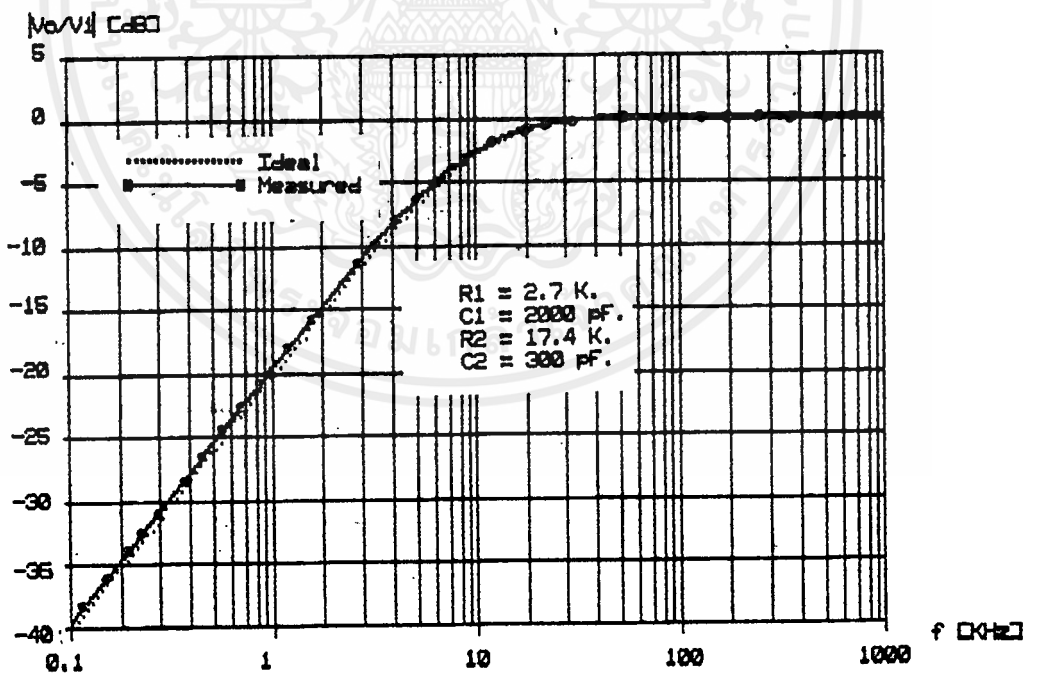


รูปที่ 3.25 แสดงผลตอบสนองทางขนาดต่อความถี่วัดได้จากวงจรความถี่สูงที่ $\alpha = 2$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.26 แสดงผลตอบสนองทางขนาดต่อความถี่วัดได้จากวงจรของความถี่สูงที่ $\alpha = 4$



รูปที่ 3.27 แสดงผลตอบสนองทางขนาดต่อความถี่วัดได้จากวงจรของความถี่สูงที่ $\alpha = 6$

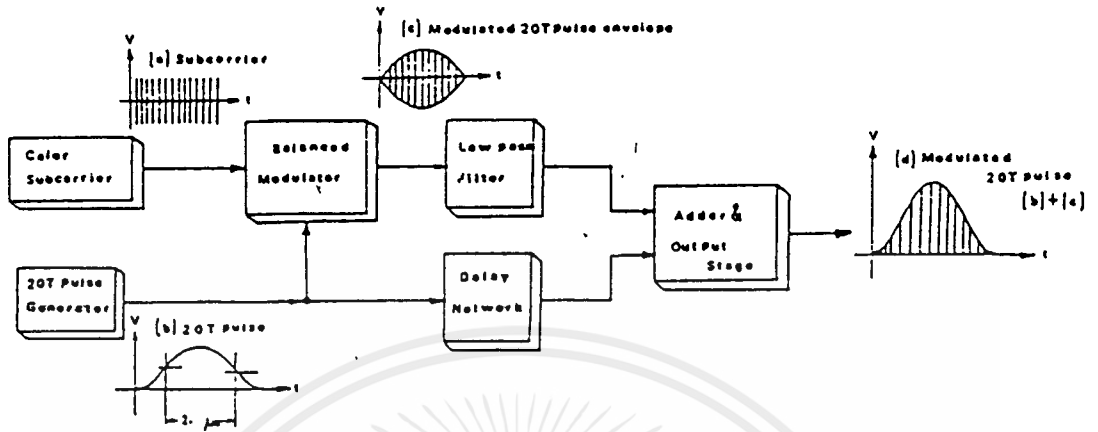
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

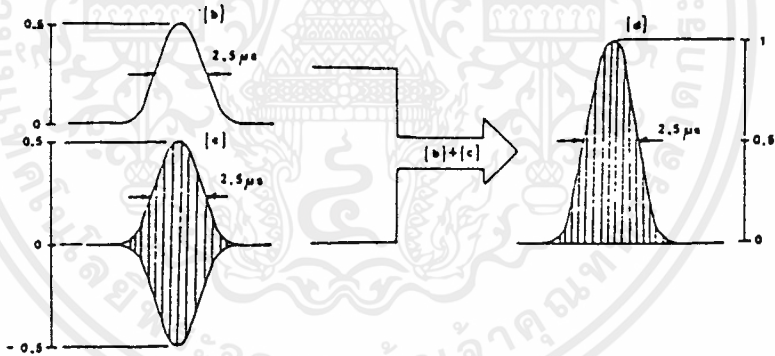
การวัดความผิดเพี้ยนของอัตราขยายและดีเลย์โดยวิธีมอดูเลตไซน์กำลังสองพัลส์

ในการวัดคุณสมบัติของสายส่งสำหรับงานด้านโทรทัศน์ การทดสอบภูมิแน้นซ์ โครมิแน้นซ์ อัตราขยายและดีเลย์ อินอีควอลิตี้ (Luminance-Chrominance Gain and Delay Inequality) โดยการนำสัญญาณรวม (Composite Signal) ซึ่งประกอบด้วยพัลส์ไซน์กำลังสอง หรือ โคลไซน์กำลังสองอย่างใดอย่างหนึ่ง ในที่นี้จะกล่าวถึง พัลส์ไซน์กำลังสอง (Sine Squared Pulse) รวมกับพัลส์มอด (Pulse Mod) ที่ความถี่คลื่นพาหะของสี (Color Subcarrier) ในกรณีที่ไม่มี ความผิดเพี้ยนเกิดขึ้นสัญญาณรวมจะมีเอ็นโวล (Envelope) โดยเฉพาะที่ฐานของพัลส์ (Baseline) เป็นเส้นตรงและในกรณีที่เกิดความผิดเพี้ยนขึ้นในระบบการส่งจากการสังเกตเอ็นโวลของฐานของพัลส์ (Envelope of Baseline) ซึ่งอาจจะเกิดเป็นลักษณะของไซน์ หรือ โคลไซน์นั้น เราสามารถพิจารณาได้จากออสซิลโลสโคป โดยการวัดค่ายอด (peak) สูงสุดและต่ำสุดของฐานพัลส์เราก็สามารถคำนวณหาค่า อัตราขยาย (Gain) และดีเลย์ (Delay) ได้

สัญญาณรวมโมดูลีตพัลส์ 20T (Composite 20T pulse) สามารถสร้างขึ้นได้ ดังแสดงในรูป 4.1 โดยการป้อนสัญญาณพัลส์ 20T จากเครื่องกำเนิดรูป (b) โดยสัญญาณดังกล่าว มีค่า H.A.D. 2 ไมโครวินาที ในระบบ PAL และ 2.5 ไมโครวินาที ในระบบ NTSC. แล้วมอด กับสัญญาณคัลเลอร์ซัพแคริเออร์ (a) จะได้สัญญาณรูป (c) โดยสัญญาณที่มอดแล้วจะผ่านวงจรรอง ความถี่ต่ำแบนด์วิดท์กว้าง เพื่อกรองเอาฮาร์โมนิกที่ไม่ต้องการออก ส่วนสัญญาณจาก (b) อีกทางหนึ่งจะผ่านวงจรหนึ่ง (Delay network) เพื่อปรับให้สัญญาณ (b) กับสัญญาณ (c) มีไทม์มิ่ง เท่ากัน แล้วจึงผ่านวงจรรวม (adder) ก็จะได้ สัญญาณรวมโมดูลีตพัลส์ 20T ในรูป (d) ซึ่งสัญญาณดังกล่าวจะถูกนำมาใช้วัดความผิดเพี้ยนของสัญญาณโทรทัศน์



รูปที่ 4.1 วงจรผลิตสัญญาณรวมโมดูลี 20T



รูปที่ 4.2 แสดงการรวมสัญญาณ (b)+(c)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1 การคำนวณหาความถี่สเปกตรัมของสัญญาณรวมโมดูล์เล็กพัลส์ 20T

พีเค้นที่สเปกตรัมของโมดูล์เล็ก 20T พัลส์และ 12.5T พัลส์ ดังจะได้กล่าวต่อไปนี้มีมอดูล์เล็กไซน์กำลังสองพัลส์ ประกอบด้วย สัญญาณในย่านความถี่ต่ำ (สัญญาณที่ยังไม่ได้มอด) และสัญญาณในย่านความถี่สูง (สัญญาณมอดแล้ว)

สัญญาณในย่านความถี่ต่ำมีสมการคือ

$$v_1(t) = \frac{A}{2} \sin^2 \frac{\pi t}{40T}$$

มีสเปกตรัมดังนี้

$$V_1(\omega) = \frac{A}{2} \cdot \frac{20T}{1 - \left(\frac{20\omega T}{\pi}\right)^2} \cdot \frac{\sin 20\omega T}{20\omega T}$$

$$V_1(f) = \frac{A}{2} \cdot \frac{20T}{1 - (40Tf)^2} \cdot \frac{\sin 40\pi f T}{40\pi f T} \quad \dots (4.1)$$

สัญญาณมอดูล์เล็กไซน์กำลังสองพัลส์แบบ 20T มีสมการดังนี้

$$v_c(t) = \begin{cases} \frac{A}{2} \sin^2 \frac{\pi t}{40T} \cos \omega_c t & |t| \leq 20T \\ 0 & |t| > 20T \end{cases}$$

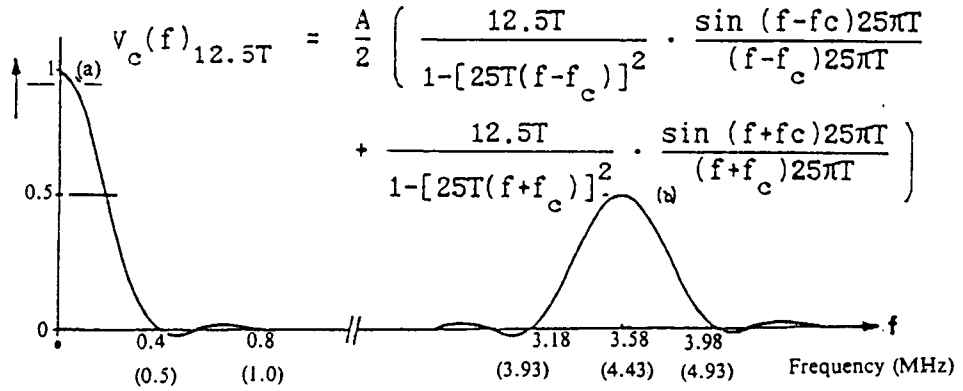
มีสเปกตรัมดังนี้

$$V_c(\omega) = \frac{A}{2} \int_0^{20T} \left(\left(1 - \cos \frac{\pi t}{20T} \right) \cos \omega_c t \right) \cos \omega t dt$$

$$V_c(\omega) = \frac{A}{2} \left(\frac{20T}{1 - \left(\frac{20T(\omega - \omega_c)}{\pi}\right)^2} \frac{\sin (\omega - \omega_c) 20T}{(\omega - \omega_c) 20T} + \frac{20T}{1 - \left(\frac{20T(\omega + \omega_c)}{\pi}\right)^2} \frac{\sin (\omega + \omega_c) 20T}{(\omega + \omega_c) 20T} \right)$$

สมการของพีเค้นที่ สเปกตรัมอาจเขียนได้ใหม่ดังนี้

$$V_c(f)_{20T} = \frac{A}{2} \left(\frac{20T}{1 - [40T(f - f_c)]^2} \cdot \frac{\sin (f - f_c) 40\pi T}{(f - f_c) 40\pi T} + \frac{20T}{1 - [40T(f + f_c)]^2} \cdot \frac{\sin (f + f_c) 40\pi T}{(f + f_c) 40\pi T} \right) \quad \dots (4.2)$$



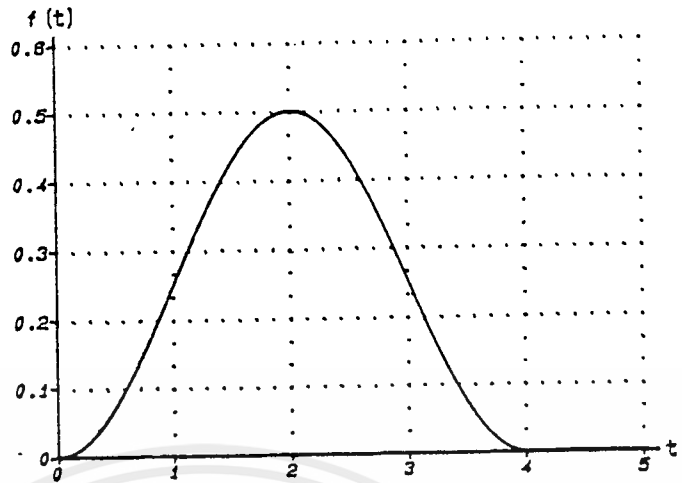
รูปที่ 4.3 แสดงพีแคว้นที่สเปคตรัมของสัญญาณมอดูเลต 20T ไซน์กำลังสองพัลส์

4.2 ทฤษฎีและการหาสูตรสำเร็จของความผิดเพี้ยนของอัตราขยายและดีเลย์

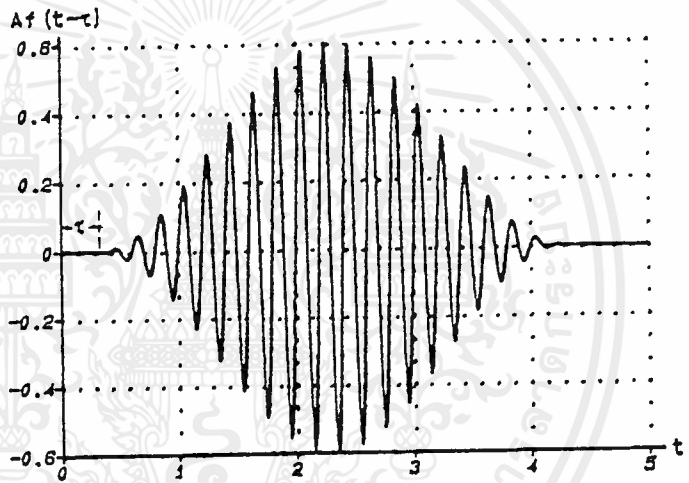
สัญญาณมอดูเลตไซน์กำลังสองพัลส์ที่จะใช้เป็นสัญญาณทดสอบประกอบด้วยส่วนที่เป็นความถี่ต่ำหรือสัญญาณลูมิแนนซ์ ร่วมกับมอดูเลตความถี่ซิกแซกหรือโครมิแนนซ์ [20] การหาความผิดเพี้ยนของสัญญาณสี (Chrominance signal) ทั้งอัตราขยาย (Gain) ซึ่งใช้สัญลักษณ์ A และ ไทม์ดีเลย์ (τ) โดยผลจากการคำนวณหาความผิดเพี้ยนทั้ง A และ จะสัมพันธ์กับค่าฮอดทั้งสองจากฐานของพัลส์ ซึ่งในทางปฏิบัติถ้าทราบค่าฮอดจากฐานของพัลส์ก็สามารถหาความผิดเพี้ยนของพัลส์ว่าในระบบทดสอบ (System Under test) จะมีผลต่อสัญญาณสีทางด้านไหน เช่น ระบบทดสอบอาจมีผลทางด้านความผิดเพี้ยนของอัตราขยาย (Gain Distortion) อย่างเดียว หรือความผิดเพี้ยนทางด้านไทม์ดีเลย์ (Time Delay Distortion) เพียงอย่างเดียว หรือมีผลของความผิดเพี้ยนทั้งอัตราขยายและไทม์ดีเลย์ ซึ่งในบทนี้จะแสดงผลการคำนวณค่าความผิดเพี้ยนทั้งอัตราขยายและไทม์ดีเลย์ และการแสดงค่าโดยประมาณด้วยค่าที่กำหนดให้ของค่าฮอดทั้งสองที่ฐานของพัลส์ และอัตราขยายที่ไม่เท่ากัน (A) และไทม์ดีเลย์ที่สัมพันธ์กัน (τ)

การคำนวณของอัตราขยายและดีเลย์ กำหนดสัญลักษณ์การคำนวณดังต่อไปนี้

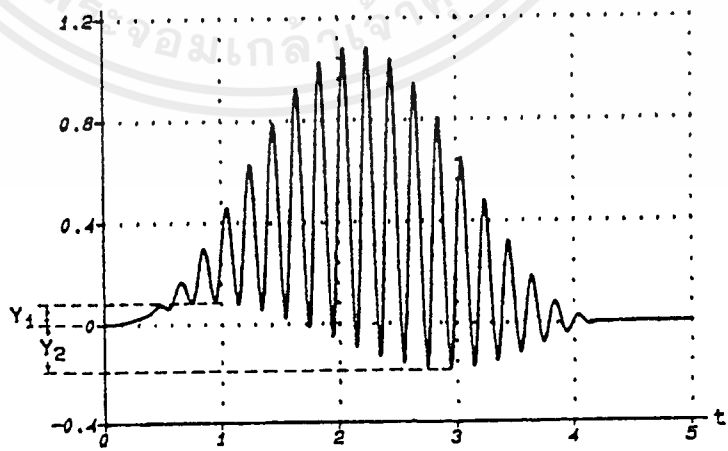
- f(t) เป็นเบสแบนด์ฟังก์ชันของไซน์กำลังสองพัลส์
- U(t) เป็นเอ็นวิโวลด้านบนของไซน์กำลังสองพัลส์
- L(t) เป็นเอ็นวิโวลด้านล่างของไซน์กำลังสองพัลส์
- Y_1 และ Y_2 เป็นขนาดของเบสไลน์ด้านล่างของไซน์กำลังสองพัลส์
- A เป็นอัตราขยายของมอดูเลตไซน์กำลังสองพัลส์ที่สัมพันธ์กัน ระหว่างระดับของสัญญาณโครมิแนนซ์ต่อระดับของสัญญาณลูมิแนนซ์
- τ เป็นความผิดเพี้ยนของไทม์ดีเลย์ คือ ส่วนของสัญญาณลูมิแนนซ์ถูกหน่วงเข้าไป หรือนำหน้าสัญญาณลูมิแนนซ์



(ก) ไซน์กำลังสองพัลส์ที่ยังไม่ถูกมอด



(ข) ไซน์กำลังสองที่ถูกมอดแล้ว



(ค) สัญญาณรวมไซน์กำลังสองพัลส์ (ก)+(ข)

รูปที่ 4.4 รูปแบบของความผิดเพี้ยนทั้งขนาดและดีเลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการคำนวณให้พิจารณาจากรูปที่ 4.4 จะได้เอ็นโวลต์ด้านบนและด้านล่างดังนี้

$$U(t) = f(t) + Af(t-\tau) \quad \dots(4.3)$$

$$L(t) = f(t) - Af(t-\tau) \quad \dots(4.4)$$

และค่าเบสแบนด์ (Baseband) ของโซ่กำลังสองพัลส์ก็คือ

$$f(t) = \begin{cases} \frac{1}{2} \sin^2 \frac{\pi t}{2T_0} & |t| \leq T_0 \\ 0 & |t| > T_0 \end{cases} \quad \dots(4.5)$$

เมื่อ T_0 เป็นขนาดความกว้างของคลื่นช่วงตรงกลาง (HAD) ของพัลส์ ในระบบ PAL กรณีของ 20 T Pulse คือ $2 \mu S$ และ 12.5 T Pulse คือ $1.25 \mu S$ ส่วนในระบบ NTSC 20 T Pulse คือ $2.5 \mu S$

การหาค่าเอ็นโวลต์ด้านบนและด้านล่างได้ด้วยการแทนค่าสมการ (4.5) ลงในสมการ (4.3) และ (4.4) ตามลำดับ จากเอ็นโวลต์ด้านล่างจะทำให้เราทราบค่า Y_1 และ Y_2 ส่วนเอ็นโวลต์ด้านบนจะทำให้เราทราบค่า Y_{max} แต่จากการวัดค่าที่แน่นอนของค่ายอดจากฐานของพัลส์ Y_1 และ Y_2 ควรจะวัดเทียบกับขนาดของพัลส์ที่สูงสุด Y_{max} ซึ่งเอ็นโวลต์ด้านล่างสามารถเขียนได้ใหม่คือ

$$L(t) = \frac{1}{4} + \frac{1}{4} \cos \frac{\pi t}{T_0} - \frac{A}{4} \cos \frac{\pi(t-\tau)}{T_0} - \frac{A}{4} \quad \dots(4.6)$$

ขนาดความสูงของเบสไลน์หาได้จาก การหาอนุพันธ์ของสมการ (4.6) เทียบกับ t และให้เท่ากับศูนย์ ซึ่งจะได้

$$\sin \frac{\pi t}{T_0} - A \sin \frac{\pi(t-\tau)}{T_0} = 0 \quad \dots(4.7)$$

จากสมการที่ (4.6) กำหนดให้

$$\cos \frac{\pi t}{T_0} - A \cos \frac{\pi(t-\tau)}{T_0} = Y_a \quad \dots(4.8)$$

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าของ Y_a เป็นฟังก์ชันของ A และ τ หาได้จากยกกำลังสมการ (4.7) บวกด้วยยกกำลังสมการ (4.8) จะได้

$$Y_a = \pm \sqrt{1 + A^2 - 2A \cos \frac{\pi\tau}{T_0}} \quad \dots (4.9)$$

ขนาดของเบสไลน์ด้านล่างของมอดคูลเลขไซน์กำลังสองพัลส์จะได้

$$Y_1 = \frac{1}{4} \left(1 - A + \sqrt{1 + A^2 - 2A \cos \frac{\pi\tau}{T_0}} \right) \quad \dots (4.10)$$

$$Y_2 = \frac{1}{4} \left(1 - A - \sqrt{1 + A^2 - 2A \cos \frac{\pi\tau}{T_0}} \right)$$

ในทำนองเดียวกันขนาดความสูงของเอ็นวิโลปด้านบนสามารถหาได้คือ

$$Y_{max} = \frac{1}{4} \left(1 + A + \sqrt{1 + A^2 + 2A \cos \frac{\pi\tau}{T_0}} \right) \quad \dots (4.11)$$

จากสมการ (4.10) , (4.11) อัตราการขยายของมอดคูลเลขไซน์กำลังสองพัลส์ คือ

$$A = \frac{1 - (y_1 + y_2 + y_1 y_2)}{1 + (y_1 + y_2 - y_1 y_2)} \quad \dots (4.12)$$

$$y_1 = \frac{Y_1}{Y_{max}} \quad , \quad y_2 = \frac{Y_2}{Y_{max}}$$

จากเอ็นวิโลปด้านล่างที่ฐานของพัลส์เราสามารถที่จะหาค่าดีเลย์ อินอิควอลิตี้ (Delay Inequality) จากค่ายอดทั้งค่าบวกและค่าลบของ Y_1 และ Y_2 ตามลำดับ โดยให้

$$[1 - (y_1 + y_2 + y_1 y_2)] [1 + (y_1 + y_2 - y_1 y_2)] = \frac{16A}{1 + AY_a}$$

โดยที่
$$\gamma_a = \sqrt{1 + A^2 + 2A \cos \frac{\pi\tau}{T_0}}$$

จะได้

$$\tau = \frac{T_0}{\pi} \cos^{-1} \left(1 + \frac{8y_1y_2}{[1-(y_1+y_2+y_1y_2)] \cdot [1+(y_1+y_2-y_1y_2)]} \right) \dots(4.13)$$

หรือ

$$\tau = \frac{2T_0}{\pi} \tan^{-1} \left(\frac{4y_1y_2}{1 - (y_1+y_2)^2} \right) \dots(4.14)$$

ค่าสอดคล้องสองที่เกิดการผิดเพี้ยนของเอ็นวีโวลต์ด้านล่างจากฐานของพัลส์ Y_1 และ Y_2 ทำการ normalized ด้วยค่าสูงสุดของเอ็นวีโวลต์ด้านบน $y_1 = Y_1/Y_{max}$ และ $y_2 = Y_2/Y_{max}$ โดยที่ค่า Y_1 และ Y_2 จะเป็นพีชคณิตที่มีเครื่องหมายตรงข้ามกัน ซึ่ง Y_1 จะเป็นค่าสอดคล้องแรก และ Y_2 จะเป็นค่าสอดคล้องต่อมาในเบสไลน์

การผิดเพี้ยนโดยความแตกต่างของอัตราขยายอย่างเดี่ยว

ในกรณีนี้แสดงว่าค่าสอดคล้อง Y_1 และ Y_2 ค่าใดค่าหนึ่งมีค่าเป็น 0 ดังนั้นจะไม่มีผิดเพี้ยนทางดีเลย์เกิดขึ้น ก็คือ $\tau = 0$ เพราะฉะนั้นความผิดเพี้ยนที่เกิดขึ้นจึงมีแต่ความผิดเพี้ยนของอัตราขยายอย่างเดี่ยวซึ่งสามารถแสดงให้เห็นจริงได้ ดังนี้

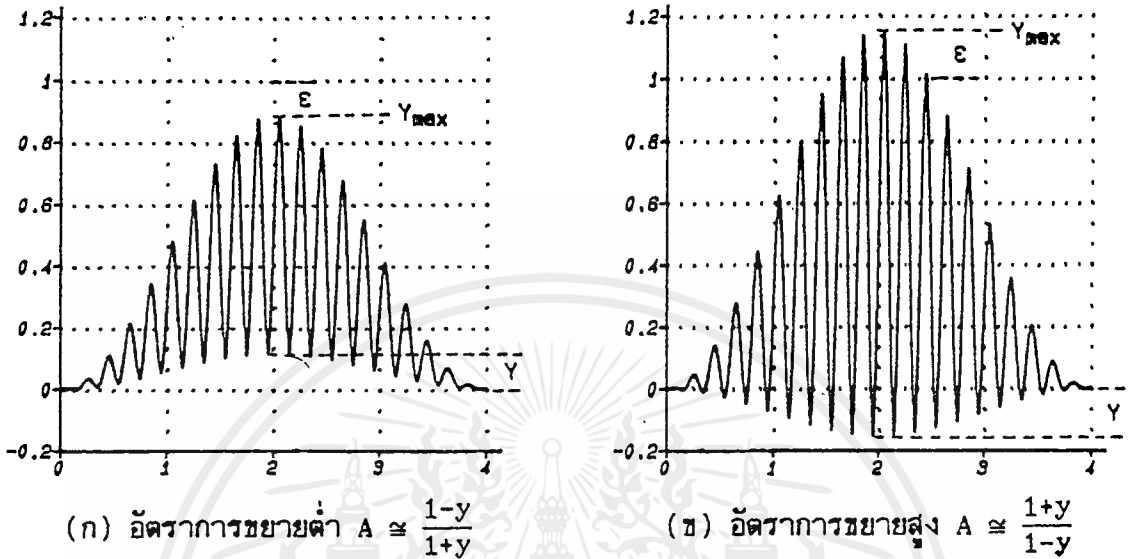
เมื่อ Y_1 หรือ Y_2 มีค่าเป็น 0 สมการ (4.13) จะกลายเป็น

$$\tau = \frac{T_0}{\pi} \cos^{-1} (1) = 0$$

และสมการ (4.12) จะกลายเป็น

$$A = \frac{1-(y_{1,2})}{1+(y_{1,2})} = \frac{1 - (Y/Y_{max})}{1 + (Y/Y_{max})} \dots(4.15)$$

ซึ่ง $(y_{1..2})$ หรือ Y อาจมีค่าเป็นบวกหรือลบ เนื่องจากขนาดของความผิดเพี้ยนที่สัมพันธ์กันดังรูปที่ 4.5



รูปที่ 4.5 แสดงความแตกต่างของความผิดเพี้ยนของอัตราขยายอย่างเดียว

ความผิดเพี้ยนของอัตราขยาย (ϵ) พิจารณาได้จากสมการ (4.15) ดังนี้

$$\epsilon = 1 - A = (2Y/Y_{max}) / (1 + Y/Y_{max}) \quad \dots (4.16)$$

ถ้า $1 \gg Y/Y_{max}$ จะได้ $\epsilon \cong 2Y/Y_{max}$ ซึ่ง $2Y/Y_{max}$ จะถูกเรียกเป็น Relative Chrominance Level (RCL) และบางครั้งถูกเรียกว่า Relative Amplitude of the Color Sub-Carrier

การผิดเพี้ยนโดยความแตกต่างของดีเลย์อย่างเดียว

ในกรณีที่ $Y_1 = -Y_2$ ซึ่งฐานของพัลส์มีรูปร่างเป็นคลื่นไซน์ที่มีค่ายอดเท่ากันแต่อยู่ตรงข้าม แสดงว่าไม่มีความผิดเพี้ยนทางอัตราขยายความผิดเพี้ยนที่เกิดขึ้นมีแต่ความผิดเพี้ยนทางด้านดีเลย์เพียงอย่างเดียวดังได้แสดงต่อไปนจากสมการ (4.12) แทนค่า $Y_1 = -Y_2$ จะได้ดังนี้

$$A = \frac{1 + (Y/Y_{max})^2}{1 + (Y/Y_{max})^2} = 1 = 0 \text{ dB}$$

จากสมการ (4.13) จะได้เป็น

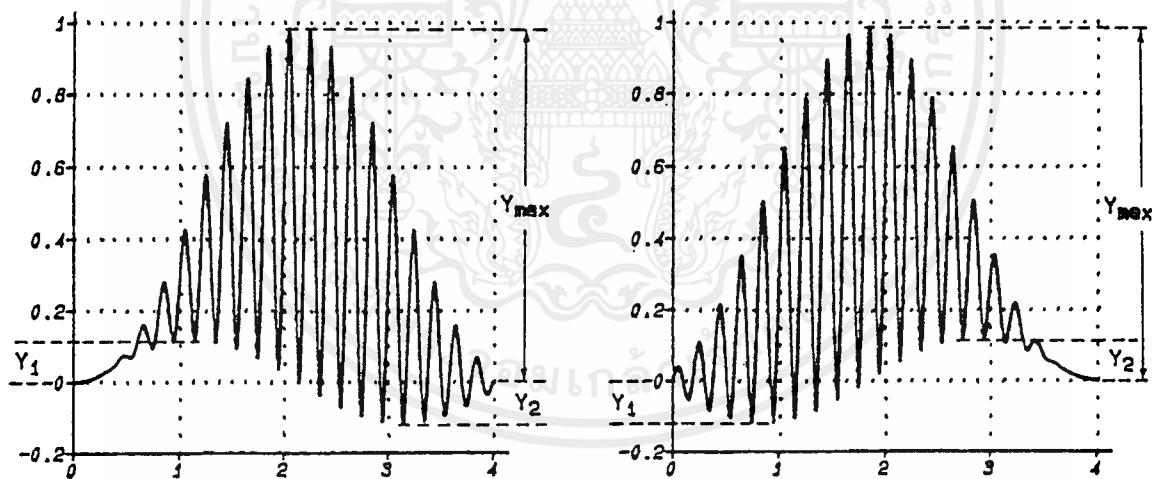
$$\tau = \frac{T_0}{\pi} \cos^{-1} \left(1 - \frac{8(Y/Y_{\max})^2}{[1 + (Y/Y_{\max})^2]^2} \right) \quad \dots(4.17)$$

โดยอาศัยสูตรทางตรีโกณมิติ สมการ (4.17) จะได้เป็น

$$\tau = \frac{2T_0}{\pi} \tan^{-1} \frac{2Y/Y_{\max}}{[1 - (Y/Y_{\max})^2]^2}$$

$$\tau = \frac{4T_0}{\pi} \tan^{-1} \frac{Y}{Y_{\max}}$$

ในกรณีที่เกิดการเพิ่มขึ้นทางด้านคลื่น สามารถสังเกตฐานของพัลส์จะเกิดเป็นลักษณะของไซน์ (Sine) หรือโคไซน์ (Cosine) ดังในรูปที่ 4.6

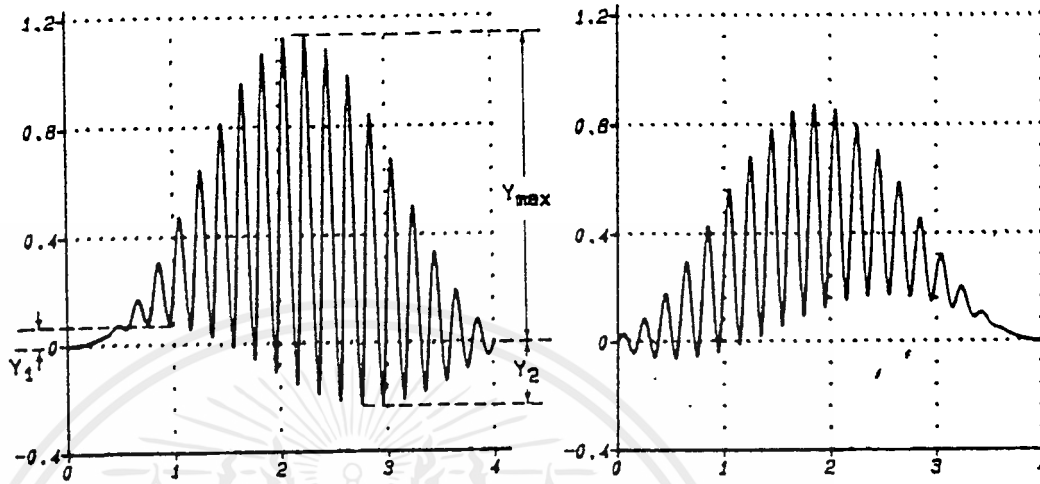


รูปที่ 4.6 แสดงการผิดเพี้ยนของคลื่นเพียงอย่างเดียว

กรณีที่เกิดการบิดเบือนทางอัตราขยายและดีเลย์ พร้อมกัน

ในกรณีที่เกิดความผิดเพี้ยนทั้งสองแบบ ค่าสอดคล้องบวกและลบมีค่าแตกต่างกันดังแสดง

ในรูปที่ 4.7



รูปที่ 4.7 แสดงความผิดเพี้ยนทางอัตราขยายและดีเลย์ พร้อมกัน

เมื่อความผิดเพี้ยนมีค่าน้อย ดังนั้นผลคูณของ $y_1 y_2$ สามารถตัดทิ้งได้สมการ (4.12)

และ (4.13) อาจเขียนได้ใหม่ดังนี้

$$A \cong \frac{1 - (y_1 + y_2)}{1 + (y_1 + y_2)} \quad \dots(4.18)$$

$$\tau \cong \frac{T_0}{\pi} \cos^{-1} \left(1 + \frac{8y_1 y_2}{1 - (y_1 + y_2)^2} \right) \quad \dots(4.19)$$

และถ้า $(y_1 + y_2)^2 \ll 1$ ความผิดเพี้ยนทางด้านดีเลย์อาจเขียนได้ใหม่ คือ

$$\tau \cong \frac{T_0}{\pi} \cos^{-1} (1 + 8y_1 y_2) \cong \frac{2T_0}{\pi} \sin^{-1} \sqrt{-4y_1 y_2}$$

$$\tau \cong \frac{4T_0}{\pi} \sqrt{-y_1 y_2} \quad \dots(4.20)$$

ความผิดเพี้ยนทางอัตราขยายคำนวณด้วยสมการ (4.12) และ (4.18) จะได้ค่าที่ใกล้เคียงกันมากและความผิดเพี้ยนทางด้านดีเลย์เมื่อคำนวณด้วยสมการ (4.14) และ (4.20) ค่าที่ได้แตกต่างกันเพียงเล็กน้อยเมื่อผลคูณ $y_1 y_2$ มีค่าน้อย ในกรณีที่ y_1 และ y_2 มีค่าไม่เกิน 10%

บทที่ 5

หลักการของชุดทดสอบการตอบสนองทางขนาดและการตอบสนองทางเฟสของสัญญาณสี่

ก่อนที่จะกล่าวถึงการทำงานของ ชุดทดสอบการตอบสนองทางขนาดและการตอบสนองทางเฟส เพื่อชุดเซชันการฝึกเขียนของสัญญาณสี่นั้น จะขอกล่าวถึงทฤษฎีเบื้องต้นของสัญญาณภาพโทรทัศน์เสียก่อน เพื่อเป็นแนวทางในการเข้าใจการทำงานของชุดทดสอบต่อไป

ดังที่ได้กล่าวมาแล้วในตอนต้นว่าสัญญาณภาพโทรทัศน์ในลักษณะที่เรียกว่าสัญญาณภาพรวม (Composite Signal) จะประกอบไปด้วยสัญญาณส่วนต่าง ๆ ที่ประกอบกันขึ้นมา โดยการรวมสัญญาณ (Modulate) หลายๆสัญญาณเข้าด้วยกัน ซึ่งส่วนประกอบที่สำคัญและเราจะนำมากล่าวถึงในที่นี้ก็ได้แก่ สัญญาณภาพขาวดำหรือสัญญาณส่องสว่าง (Luminance Signal) , สัญญาณสี (Chrominance Signal) และสัญญาณซิงค์ (Sync Signal) สัญญาณทั้งสามส่วนนี้ จะมีความสำคัญในการให้ภาพปรากฏที่หน้าจอเครื่องรับโทรทัศน์ คือ สัญญาณภาพขาวดำหรือสัญญาณส่องสว่าง จะให้ความสว่างที่หน้าจอในลักษณะของภาพขาวดำ สัญญาณสีจะเป็นส่วนที่จะทำให้ภาพที่หน้าจอปรากฏเป็นภาพที่มีสีสันต่าง ๆ เป็นการเพิ่มความสวยงามและความสมจริงให้กับภาพ ส่วนสัญญาณซิงค์จะทำหน้าที่คอยควบคุมให้เกิดภาพที่หน้าจอได้ นั่นคือ สัญญาณซิงค์จะเป็นตัวควบคุมการสแกน (Scan) ของสัญญาณภาพที่หน้าจอเพื่อทำให้เกิดภาพขึ้นได้ โดยสัญญาณซิงค์นี้จะต้องสัมพันธ์กันระหว่างเครื่องส่งและเครื่องรับ สัญญาณซิงค์ยังแบ่งออกเป็นสัญญาณซิงค์ทางแนวตั้ง (Vertical) และสัญญาณซิงค์ทางแนวนอน (Horizontal)

จะเห็นได้ว่าการส่งสัญญาณโทรทัศน์เพื่อให้ได้ภาพที่สมบูรณ์นั้น จะขาดสัญญาณใดสัญญาณหนึ่งไม่ได้เลย และนอกจากสัญญาณสามส่วนนี้แล้ว สัญญาณโทรทัศน์ยังประกอบด้วยสัญญาณส่วนอื่น ๆ อีก ซึ่งมีรายละเอียดปลีกย่อยมากมาย แต่จะไม่นำมากล่าวถึงในที่นี้

สเปกตรัมของสัญญาณ (Signal Spectrum)

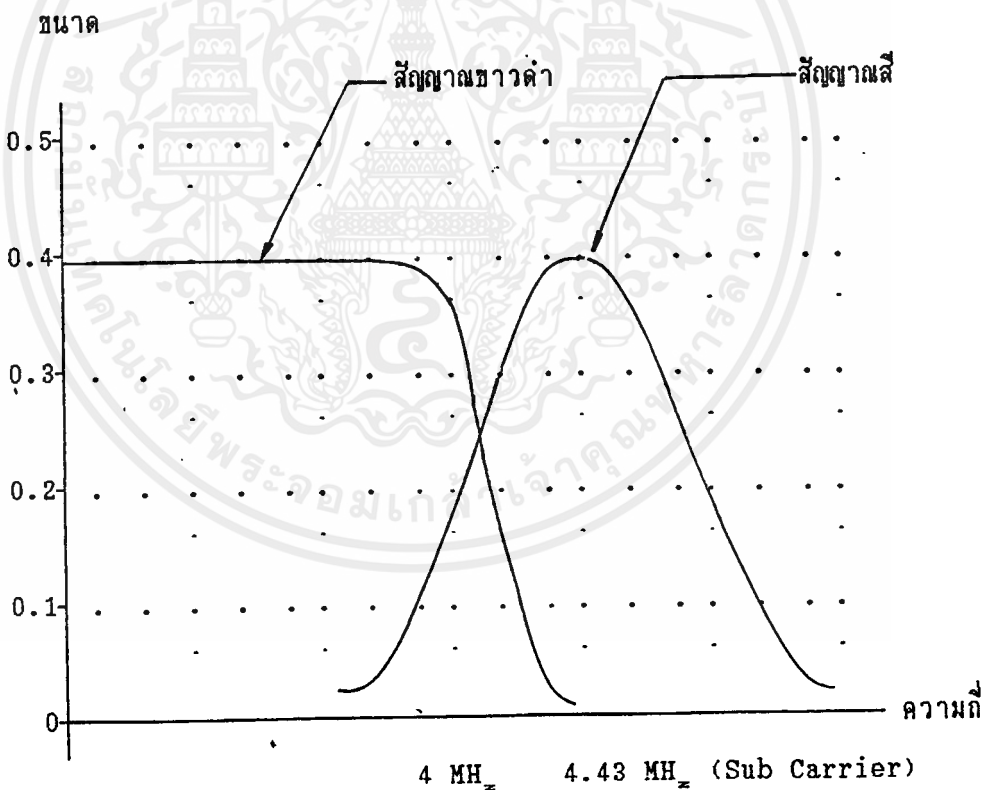
สัญญาณโทรทัศน์ดังที่กล่าวมา จะเป็นสัญญาณร่ายคาบ (Periodic) ซึ่งจะประกอบด้วยองค์ประกอบของสัญญาณย่อย ๆ จำนวนมากมายที่มีความถี่เป็นจำนวนเท่าคือ 1,2,3,... เท่าของความถี่หลักของสัญญาณ องค์ประกอบย่อย ๆ นี้ เราเรียกว่า ฮาร์โมนิกส์ (Harmonic) ของสัญญาณ เมื่อเรานำฮาร์โมนิกส์ทั้งหมดของสัญญาณ มาเขียนกราฟความสัมพันธ์ระหว่างขนาดและลำดับของฮาร์โมนิกส์ เราก็จะได้ลักษณะส่วนประกอบของสัญญาณทั้งหมด หรือที่เราเรียกว่า สเปกตรัมของสัญญาณ ดังนั้นสเปกตรัมของสัญญาณใด ๆ จึงแสดงถึงความสัมพันธ์ระหว่างองค์ประกอบของสัญญาณหลักกับความถี่ต่าง ๆ และขนาดของสัญญาณที่ความถี่นั้น ๆ ซึ่งจะทำให้เราทราบถึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความกว้างของแถบสัญญาณ (Band Width) การวัดสเปกตรัมของสัญญาณสามารถทำได้ทั้งวิธีการคำนวณและวิเคราะห์ทางคณิตศาสตร์ หรือโดยวิธีการวัดโดยใช้เครื่องมือวัด เช่น เครื่องวิเคราะห์สเปกตรัม (Spectrum Analyzer) ก็ได้ ซึ่งในหัวข้อนี้เราจะกล่าวถึงสเปกตรัมของสัญญาณภาพขาวดำ (สัญญาณส่องสว่าง) และสเปกตรัมของสัญญาณสี เพื่อให้เข้าใจหลักการเบื้องต้นในการแยกสัญญาณภาพส่วนต่าง ๆ ออกจากกัน

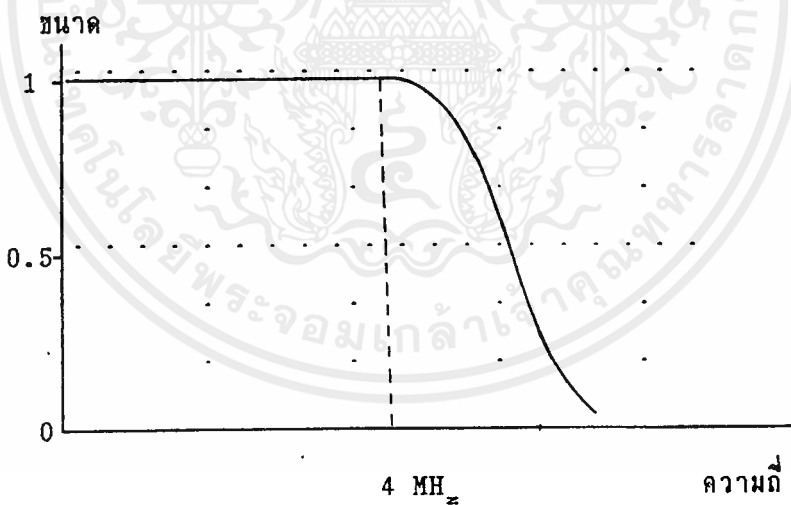
โดยปกติแล้วสเปกตรัมของสัญญาณขาวดำและสเปกตรัมของสัญญาณสีจะอยู่ในย่านความถี่ที่แตกต่างกัน ส่วนจะอยู่ในช่วงความถี่ใดนั้นก็ขึ้นอยู่กับระบบของสัญญาณโทรทัศน์ที่ใช้ เช่น PAL NTSC SECAM HDTV เป็นต้น แต่ในประเทศไทยใช้การแพร่สัญญาณโทรทัศน์ในระบบ PAL-B ดังนั้นจะขอกล่าวถึงสเปกตรัมของสัญญาณขาวดำและสัญญาณสีในระบบ PAL โดยที่สเปกตรัมของสัญญาณขาวดำและสัญญาณสีในระบบ PAL แสดงได้ดังรูป 5.1



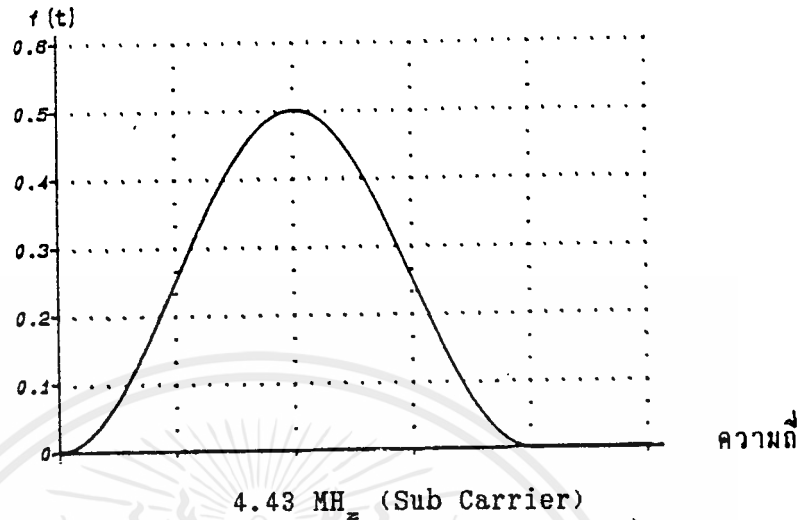
รูปที่ 5.1 แสดงสเปกตรัมของสัญญาณขาวดำและสัญญาณสี

จากรูปจะเห็นว่าสเปกตรัมของสัญญาณขาวคำจะเริ่มจากความถี่ที่เป็นสัญญาณ DC ไปจนถึงความถี่สูงเป็นเมกะเฮิรตซ์ (ประมาณ 0-5 MHz) ซึ่งครอบคลุมย่านของความถี่ค่อนข้างกว้าง ในขณะที่สเปกตรัมของสัญญาณสีจะอยู่ในช่วง 4.43 MHz บวกลบ 0.5 MHz ซึ่งจะครอบคลุมย่านของความถี่แคบกว่าสัญญาณขาวคำและจะอยู่ในย่านความถี่ที่ค่อนข้างสูง (ความถี่ 4.43 MHz เป็นความถี่พาหะย่อยที่นำมาผสมกับสัญญาณสี ก่อนที่จะนำไปรวมกับสัญญาณขาวคำให้เป็นสัญญาณภาพรวม ความถี่พาหะย่อยนี้ เรียกว่า Sub Carrier นั้นเอง)

ด้วยเหตุนี้เราจึงสามารถแยกสัญญาณขาวคำและสัญญาณสีออกจากกันได้ โดยใช่วงจรกรองสัญญาณ (Filter Circuit) กล่าวคือ เราจะสามารถใช้ วงจรกรองความถี่ต่ำ (Low Pass Filter, LPF) กรองเอาเฉพาะสัญญาณขาวคำ ออกมาได้ และในทำนองเดียวกันเราก็สามารถใช้วงจรกรองความถี่สูง (High Pass Filter, HPF) หรือจะใช้ วงจรกรองความถี่ผ่าน (Band Pass Filter, BPF) กรองเอาเฉพาะสัญญาณสีออกมาก็ได้ ดังนั้นเมื่อเราออกแบบวงจรกรองสัญญาณที่มีความเหมาะสมแล้วนำสัญญาณภาพรวมผ่านวงจรกรองสัญญาณนั้น ก็จะได้สเปกตรัมของสัญญาณดังรูปที่ 5.2



รูปที่ 5.2 สเปกตรัมของสัญญาณขาวคำที่ได้จากวงจรกรองความถี่ต่ำ



รูปที่ 5.3 สเปกตรัมของสัญญาณสี่ที่ได้จากวงจรรองความถี่สูงหรือวงจรรองความถี่ผ่าน

จากรูป จะเห็นสเปกตรัมของสัญญาณขาวดำและสัญญาณสีที่ถูกแยกออกจากกันแล้วโดย
วงจรรองสัญญาณ

5.1 การทำงานของระบบทดสอบเพื่อแก้ความผิดเพี้ยนของสัญญาณโคมิแนนซ์ต่อลุมิแนนซ์

เมื่อเราทำการแยกสัญญาณลุมิแนนซ์และโคมิแนนซ์ออกจากกันโดยวิธีการที่กล่าวมาแล้ว เรา
สามารถที่จะแก้ไขหรือชดเชยความผิดเพี้ยนของสัญญาณภาพโทรทัศน์ได้ โดยที่สาเหตุของการ
ผิดเพี้ยนนั้นได้มีการกล่าวถึงไว้แล้วในบทที่ 4 สำหรับการชดเชยสัญญาณภาพนั้นจากระบบเดิมที่
มีอยู่ใช้วงจรรองความถี่เพื่อแยกสัญญาณลุมิแนนซ์และโคมิแนนซ์ออกจากกันแล้วทำการชดเชยทาง
ขนาดและทางเฟสของสัญญาณในส่วนของสัญญาณลุมิแนนซ์ที่วงจรรองความถี่ต่ำ เพื่อแก้ความผิด
เพี้ยนที่เกิดขึ้นและให้ได้สัญญาณที่ถูกต้องดังรูปที่ 5.4 ซึ่งแสดงบล็อกไดอะแกรมของระบบแบบเดิม
แต่สำหรับปริณิงานิพนธ์ฉบับนี้ได้เสนอวิธีการใหม่ ในการออกแบบวงจรอควอไลเซอร์ที่สามารถชก
หรือลดระดับผลตอบสนองทางขนาดของสัญญาณ โดยใช้วงจรแบบแอกทีฟดิสทริบิวต์อาซี (Active
Distributed RC) และใช้วงจรดีเลย์อควอไลเซอร์ตอบสนองทางเฟสของสัญญาณ เพื่อแก้ความ
ผิดเพี้ยนที่เกิดขึ้น แทนระบบแบบเดิมที่ใช้วงจรรองความถี่ตั้งบล็อกไดอะแกรมรูปที่ 5.5

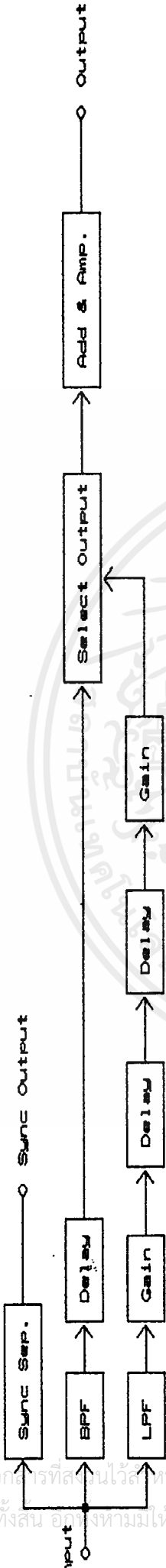


Fig. 5.4 Block Diagram of System Lumped Elements



Fig 5.5 Block Diagram of The Equalizer Distributed RC and Delay Equalizer

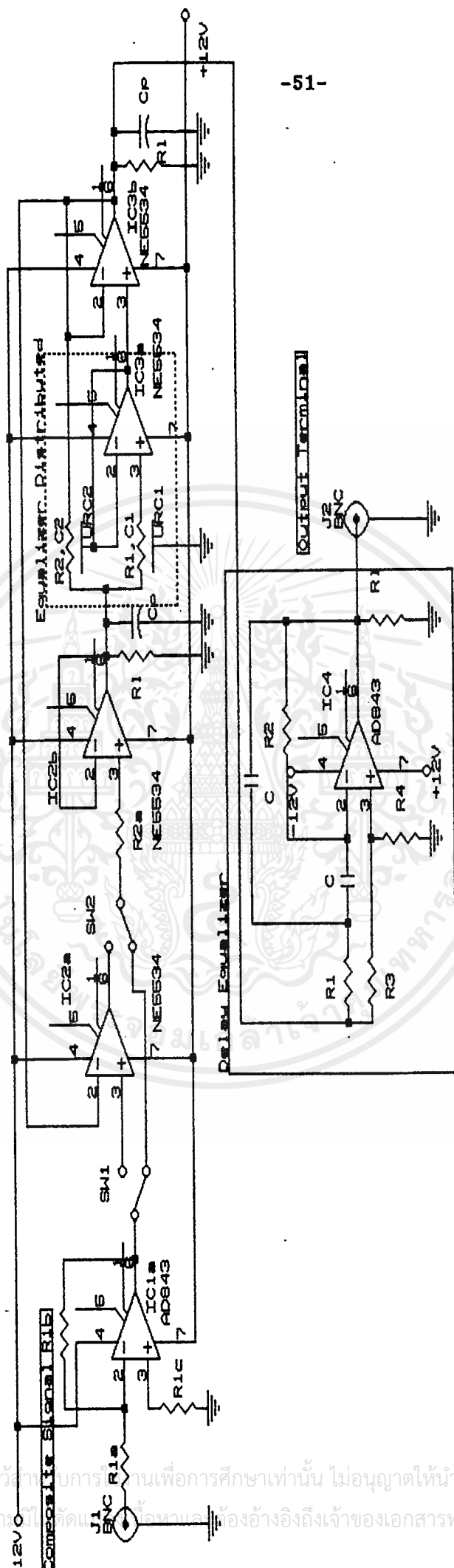


Fig 3. Equalizer Distributed RC and Delay Equalizer Circuit

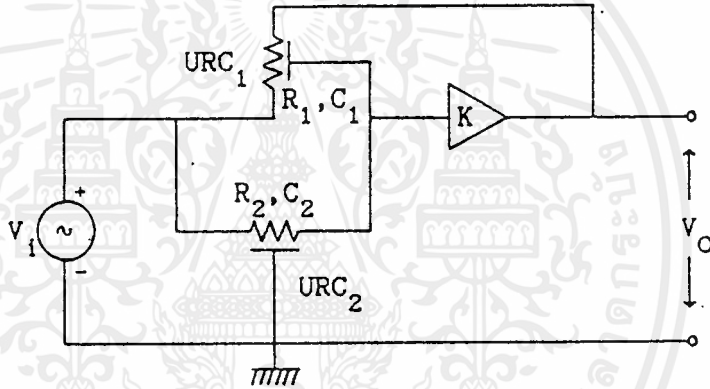
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ข้อมูลนี้ออกนอกห้องเรียนหรือห้องปฏิบัติการของอาจารย์ผู้สอน

เราจะพิจารณาการทำงานของวงจรรีจิสเตอร์แอกทีฟที่ผลิตที่บริษัทอาซี และดีเลย์รีจิสเตอร์แอกทีฟจากบล็อกไดอะแกรมรูปที่ 5.5 จากบล็อกไดอะแกรมจะเห็นว่า เราจะต้องป้อนสัญญาณอินพุตเข้าที่ขั้วอินพุตของระบบ สัญญาณอินพุตนี้เป็นสัญญาณมาตรฐาน คือจะเป็นสัญญาณที่มีลักษณะเป็นรูปพัลส์โมดูลาร์ที่โคไซน์กำลังสอง ซึ่งจะมีส่วนประกอบของสัญญาณที่เป็นสัญญาณชวาค้ารวมกับสัญญาณซี (Sub Carrier 4.43 MHz) สัญญาณนี้จะป้อนสัญญาณที่ผ่านออกมาจากระบบที่ต้องการวัดผลตอบสนองทางขนาด และผลตอบสนองทางเวลาของสัญญาณโทรทักส์น ถ้าหากระบบนั้นไม่สมบูรณ์ก็จะมีผลทำให้เกิดการผิดเพี้ยนขึ้นที่ฐานของสัญญาณมาตรฐานนั้น การทำงานของวงจรรีจิสเตอร์แอกทีฟที่ผลิตที่บริษัทอาซีและดีเลย์รีจิสเตอร์แอกทีฟจะวัดความผิดเพี้ยนทางขนาด และ กรูฟดีเลย์ของสัญญาณโดยอาศัยหลักการชดเชย (Compensate) สัญญาณ ซึ่งใช้วงจรรีจิสเตอร์แอกทีฟเพื่อชดเชยหรือลดระดับผลตอบสนองทางขนาดของสัญญาณที่ผ่านความถี่สูงโดยใช้แอกทีฟที่ผลิตที่บริษัทอาซี โดยวงจรรีจิสเตอร์แอกทีฟมีค่าความไว (Sensitivity) ของอุปกรณ์ต่ำตั้งที่กล่าวมาแล้วในเรื่องของการออกแบบวงจรและการทดสอบวงจรซึ่ง สามารถผลิตขึ้นตามกรรมวิธีผลิต LSI ที่มีขนาดเล็ก โดยที่ค่า Parasitic Capacitors ระหว่างตัวอินพุตรีจิสเตอร์แอกทีฟกับ Substrate ของไอซีจะไม่มีผลกระทบต่อคุณลักษณะของวงจรมันจึงสามารถนำไปใช้ได้ดีในช่วงความถี่สูง

แต่เนื่องจากการส่งผ่านสัญญาณเข้าวงจรต่างๆอาจทำให้ผลตอบสนองทางเฟสไม่เป็นเชิงเส้นและทำให้กรูฟดีเลย์ที่ได้ไม่คงที่ ซึ่งในระบบโทรทักส์นหรือระบบรับส่งสัญญาณพัลส์นั้นมีความจำเป็นที่ต้องการให้เฟสเป็นเชิงเส้นและให้กรูฟดีเลย์คงที่มากที่สุด ฉะนั้นจึงมีความจำเป็นที่ต้องชดเชยเฟสของสัญญาณโดยใช้วงจรรีจิสเตอร์แอกทีฟ เพื่อแก้ไขความผิดเพี้ยนทางเวลาเพื่อให้ได้กรูฟดีเลย์คงที่ (และสามารถกำหนดคุณลักษณะของกรูฟดีเลย์ตามที่ต้องการได้โดยใช้คอมพิวเตอร์ช่วยในการอิมูเลชันในการหาค่าตอบ) เมื่อสัญญาณโคมิเนนซ์และลูมิเนนซ์ผ่านวงจรชดเชยแต่ละส่วนออกมาแล้วก็จะผ่านไปส่วนเลือกสัญญาณออก (Select Output) เพื่อเลือกสัญญาณที่จะให้ปรากฏออกที่ output terminal ของเครื่อง จากการทำงานของระบบนี้จะเห็นว่าเราใช้วงจรรีจิสเตอร์แอกทีฟเพื่อทำการชดเชยหรือลดระดับของสัญญาณ และเมื่อระบบมีการตอบสนองทางเฟสไม่เป็นเชิงเส้น และทำให้กรูฟดีเลย์ที่ได้ไม่คงที่ จึงแก้ไขด้วยวงจรรีจิสเตอร์แอกทีฟ ซึ่งจะทำให้ได้กรูฟดีเลย์ที่คงที่ตามที่ต้องการได้ ดังนั้นต่อไป เราจะกล่าวถึงการออกแบบวงจรรีจิสเตอร์แอกทีฟแบบแอกทีฟที่ผลิตที่บริษัทอาซี และดีเลย์รีจิสเตอร์แอกทีฟ ซึ่งเป็นแนวทาง ในการทดลองเพื่อนำไปใช้งานต่อไป

5.2 การออกแบบวงจรอควาไลเซอร์แบบแอกทีฟคิสทริบิวท์ออาซี

ในระบบการรับส่งสัญญาณโดยทั่วไปมักจะต้องทำให้คุณลักษณะของขนาดต่อความถี่ของระบบราบเรียบที่สุด (Maximally Flat Amplitude Characteristic) แต่ในบางกรณีอาจต้องการยกระดับหรือลดระดับ(Equalize) ของสัญญาณบางช่วงความถี่ เช่น การยกหรือลดระดับความถี่สูงของ Tone control ในระบบเสียง (Audio) [18] หรือการแก้ไขความผิดเพี้ยนทางขนาดของสัญญาณสี (Chrominance) ในระบบการส่งสัญญาณภาพ (Video) [20] ในบทนี้จะเสนอวิธีการใหม่ในการออกแบบอควาไลเซอร์ (Equalizer) โดยใช้วงจรแอกทีฟคิสทริบิวท์ออาซี คือ เราจะใช้ Op-Amp หรือตัว Amplifier อื่น ๆ เช่น Video Amp จำนวน 1 ตัวร่วมกับ Uniform Distributed RC 2 ตัว โดยวงจรอควาไลเซอร์สามารถเขียนได้ดังแสดงในรูปที่ 5.6



รูปที่ 5.6 วงจรอควาไลเซอร์แบบแอกทีฟคิสทริบิวท์ออาซี

จากวงจรอควาไลเซอร์แบบแอกทีฟคิสทริบิวท์ออาซี เมื่อ K เป็น Amplifier ที่มีอัตราขยายแรงดัน (Positive Voltage Gain) เท่ากับ K และกำหนดให้ $P_1 = \sqrt{SR_1C_1}$, $P_2 = \sqrt{SR_2C_2}$ จากวงจรรูปที่ 5.6 เขียนเป็นทรานส์เฟอริงฟังก์ชันได้คือ

$$T(P) = \frac{\frac{P_1 (\cosh P_1 - 1)}{R_1 \sinh P_1} + \frac{P_2}{R_2 \sinh P_2}}{\frac{1}{K} \left(\frac{2P_1 (\cosh P_1 - 1)}{R_1 \sinh P_1} + \frac{P_2 \cosh P_2}{R_2 \sinh P_2} \right) - \frac{P_1 (\cosh P_1 - 1)}{R_1 \sinh P_1}} \dots (5.1)$$

ถ้ากำหนดให้ $P = P_1 = P_2$ หรือ $R_1C_1 = R_2C_2$ แล้วแทนค่าลงในสมการ (5.1) จะได้

$$T(P) = \frac{\cosh P - 1 + \frac{R_1}{R_2}}{\frac{1}{K} \left(2 + \frac{R_1}{R_2} - K \right) \cosh P - \frac{2}{K} + 1} \dots(5.2)$$

จากสมการ (5.2) กำหนดให้อัตราขนาดมีค่า $K = 1$ (Unity Gain Amplifier) จะได้ทรานส์เฟอ์ฟังก์ชันของวงจรอควาไลเซอร์ดังสมการ (5.3)

$$T(P) = \frac{\alpha \cosh P - \alpha + 1}{(\alpha + 1) \cosh P - \alpha} \dots(5.3)$$

โดยที่ $\alpha = R_2/R_1$

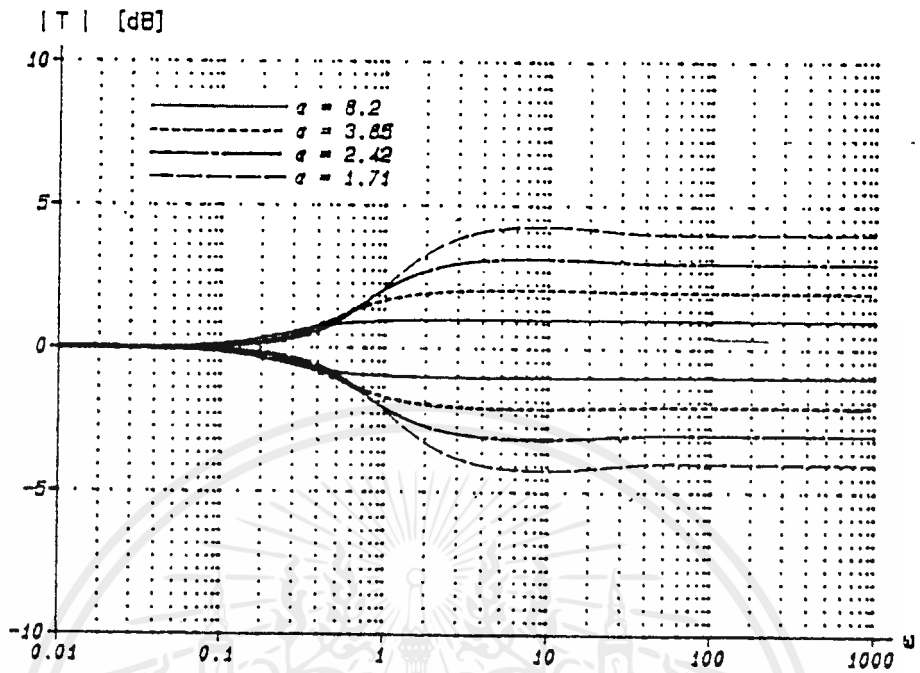
วงจรอควาไลเซอร์แบบแยกที่ผลิตทวีปักษ์ในรูปที่ 5.6. เมื่อพิจารณา Parasitic Capacitors จะเห็นได้ว่า C_{p1} จะอยู่ระหว่างแหล่งจ่ายแรงดัน (Voltage Source) กับกราวด์ ส่วน C_{p2} จะอยู่ระหว่างกราวด์ทั้งสองข้าง ฉะนั้นจึงไม่มีผลต่อคุณลักษณะของวงจร (Circuit Characteristics) จากทรานส์เฟอ์ฟังก์ชันในสมการ (5.3) สมมติต้องการให้ลดระดับที่ความถี่สูงสามารถทำได้โดยการเปลี่ยนค่า α เช่นเมื่อต้องการลดระดับที่ความถี่สูงมีค่าเท่ากับ 1, 2, 3, 4dB ซึ่งทำได้โดยกำหนดให้ α มีค่าเท่ากับ 8.2, 3.85, 2.42, และ 1.71 ตามลำดับ โดยที่ค่า α สามารถหาได้ด้วยความสัมพันธ์คือ

$$\frac{1}{\alpha} = 10^{dB/20} - 1 \dots(5.4)$$

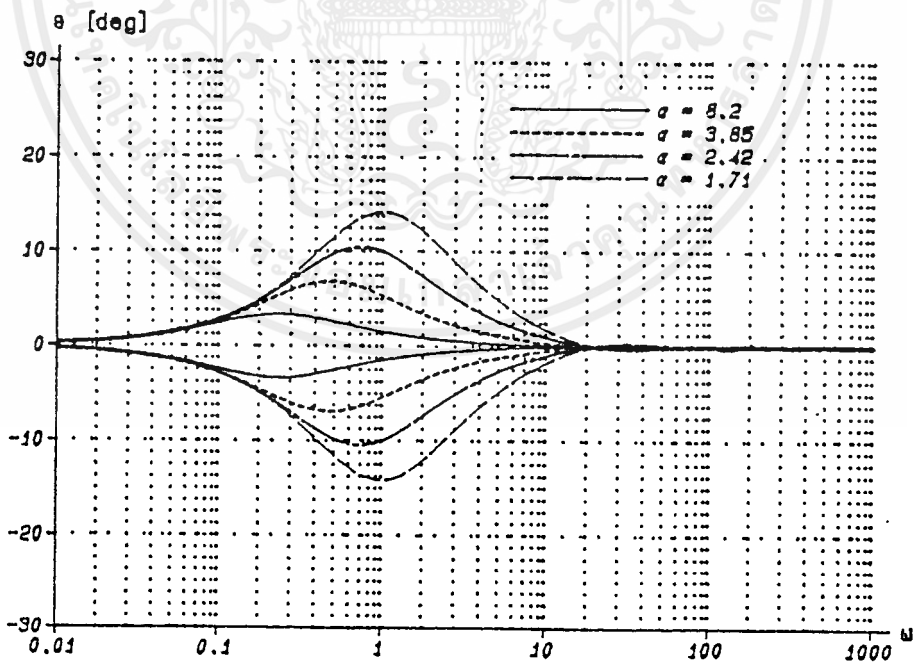
ถ้าต้องการยกระดับที่ความถี่สูง สามารถทำได้โดยเพิ่มออปแอมป์ขึ้น 1 ตัว และนำเอาสัญญาณอินพุตป้อนต่อเข้าขาบวกของออปแอมป์ และให้สัญญาณเอาต์พุตของออปแอมป์ต่อเข้าวงจรในรูปที่ 5.6 ป้อนกลับ (Feedback) เข้าออปแอมป์ทางขาลบ ซึ่งจะทำให้ทรานส์เฟอ์ฟังก์ชันกลับกัน [16,17] คือ

$$\frac{V_o}{V_i} = \frac{1}{T(P)} \dots(5.5)$$

ผลตอบสนองทางขนาดต่อความถี่ (Magnitude Response) ของวงจรอควาไลเซอร์ตามสมการ (5.3) และ (5.5) ที่สมมติให้ $\alpha = 8.2, 3.85, 2.42$ และ 1.71 แสดงดังในรูปที่ 5.7 ส่วนค่าผลตอบสนองทางเฟส (Phase Response) และผลตอบสนองของกรุปดีเลย์ (Group Delay) จะได้แสดงในรูปที่ 5.8 และ รูปที่ 5.9 ตามลำดับ

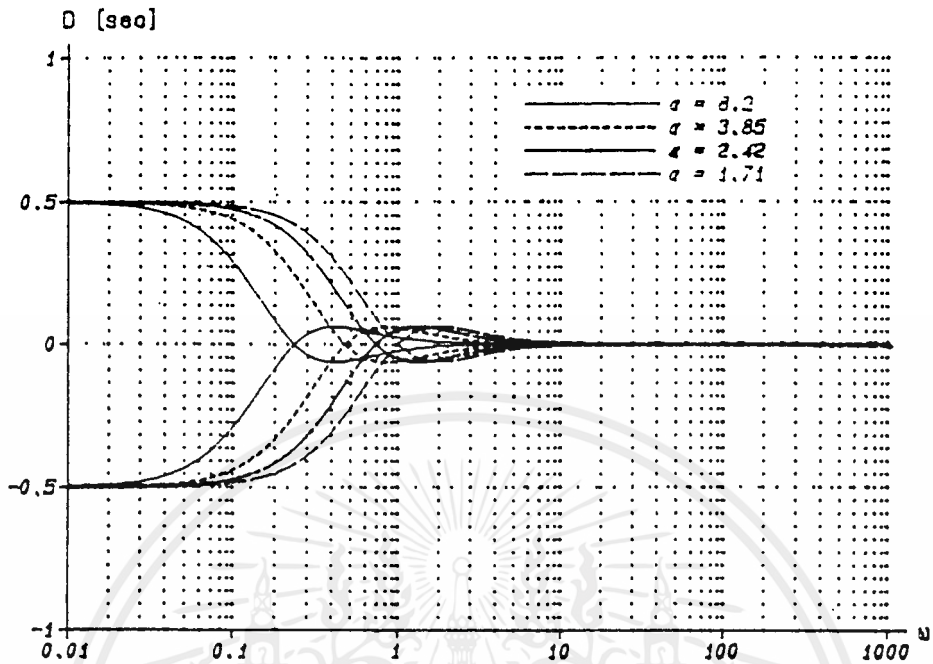


รูปที่ 5.7 แสดงผลตอบสนองทางขนาด



รูปที่ 5.8 แสดงผลตอบสนองทางเฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.9 แสดงผลตอบสนองของกริฟต์ไคเล่ย์

จากวงจรในรูปที่ 5.6 หาค่าเช่นชีวิติตของอิลิเมนต์ R_1, C_1, R_2, C_2 โดยกำหนดให้อัตราขยายมีค่า $K = 1$ ฉะนั้นจากรานส์เฟอร์ฟังก์ชันในสมการ (5.1) จะได้

$$T(s) = \frac{\sinh\sqrt{SR_2C_2} (\cosh\sqrt{SR_1C_1}-1) + \sqrt{\frac{R_1C_2}{R_2C_1}} \sinh\sqrt{SR_1C_1}}{\sinh\sqrt{SR_2C_2} (\cosh\sqrt{SR_1C_1}-1) + \sqrt{\frac{R_1C_2}{R_2C_1}} \sinh\sqrt{SR_1C_1} \cosh\sqrt{SR_2C_2}} \dots (5.6)$$

เมื่อหาเช่นชีวิติตในสมการ (5.6) แล้วกำหนดให้ $R_1C_1 = R_2C_2$ และ $\alpha = R_2/R_1$ จะได้ค่าเช่นชีวิติตของอิลิเมนต์แต่ละตัวดังนี้

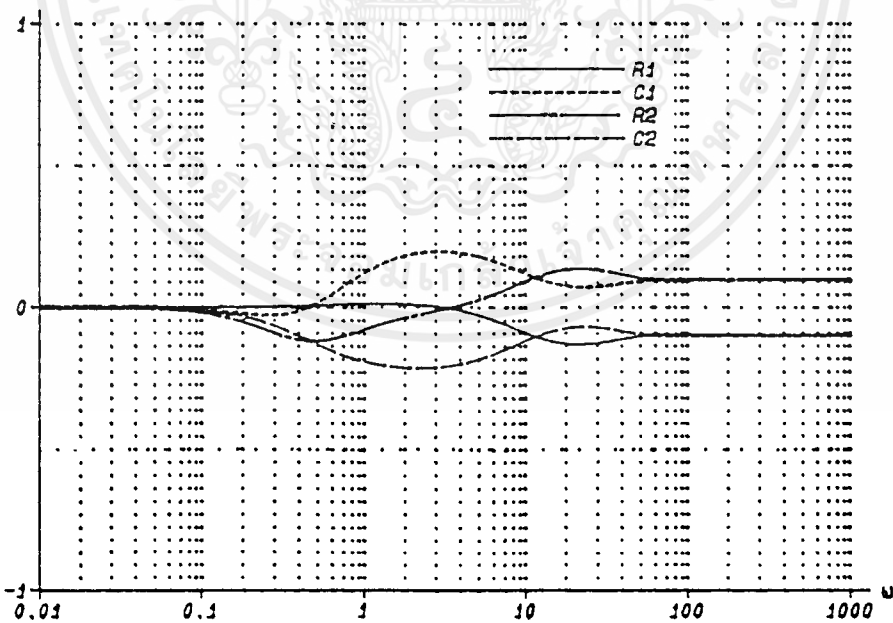
$$S_{R1} = \frac{1}{2} \left(\frac{\alpha \text{Psinh}P + \text{Pcoth}P + 1}{\alpha \cosh P - \alpha + 1} - \frac{\alpha \text{Psinh}P + (\text{Pcoth}P + 1) \cosh P}{(\alpha + 1) \cosh P - \alpha} \right) \quad \dots (5.7)$$

$$S_{C1} = \frac{1}{2} \left(\frac{\alpha \text{Psinh}P + \text{Pcoth}P - 1}{\alpha \cosh P - \alpha + 1} - \frac{\alpha \text{Psinh}P + (\text{Pcoth}P - 1) \cosh P}{(\alpha + 1) \cosh P - \alpha} \right) \quad \dots (5.8)$$

$$S_{R2} = \frac{1}{2} \left(\frac{\alpha (\cosh P - 1) \text{Pcoth}P - 1}{\alpha \cosh P - \alpha + 1} - \frac{\alpha (\cosh P - 1) \text{Pcoth}P + \text{Psinh}P - \cosh P}{(\alpha + 1) \cosh P - \alpha} \right) \quad \dots (5.9)$$

$$S_{C2} = \frac{1}{2} \left(\frac{\alpha (\cosh P - 1) \text{Pcoth}P + 1}{\alpha \cosh P - \alpha + 1} - \frac{\alpha (\cosh P - 1) \text{Pcoth}P + \text{Psinh}P + \cosh P}{(\alpha + 1) \cosh P - \alpha} \right) \quad \dots (5.10)$$

จากสมการ (5.7) - (5.10) ถ้าสมมติให้ $\alpha = 4$ จะได้ค่าเซ็นซิวิตีของอิลิเมนต์ R_1, C_1, R_2 และ C_2 แสดงดังในรูปที่ 5.10 ซึ่งจากรูปจะเห็นได้ว่าค่าเซ็นซิวิตีที่ได้มีค่าต่ำ

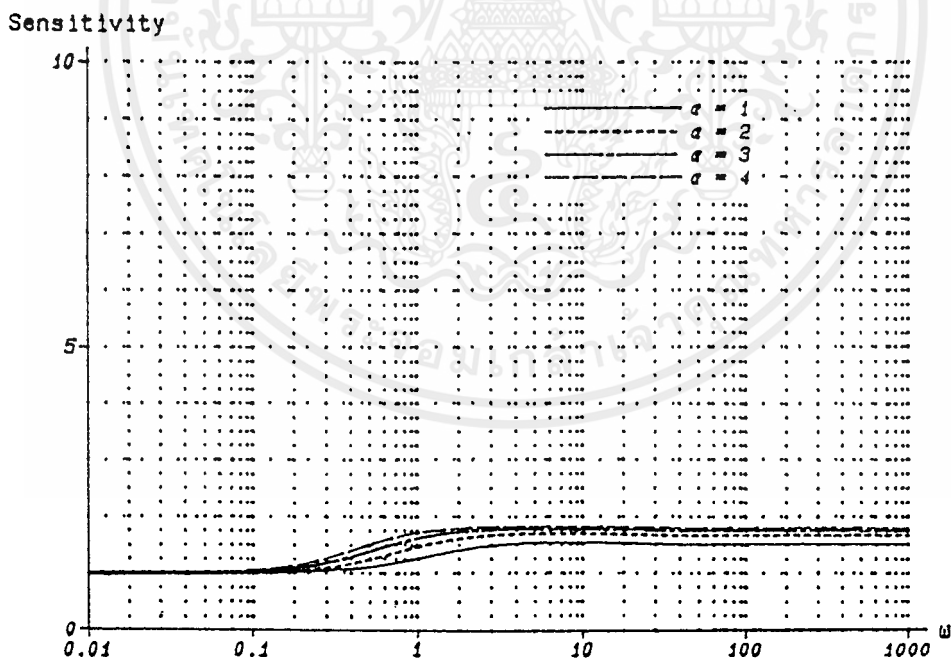


รูปที่ 5.10 แสดง Sensitivity ของอิลิเมนต์ R_1, C_1, R_2, C_2

จากวงจรในรูปที่ 5.6 เมื่อ K เป็น Amplifier ที่มีอัตราขยายแรงดันเท่ากับ K ค่าเซ็นซิวิตีของอิลิเมนต์ K หาได้จากทรานส์เฟอร์ฟังก์ชัน T(S) ในสมการ (5.2) แล้วกำหนดให้ $K = 1$ และ $\alpha = R_2/R_1$ ซึ่งจะได้ค่าเซ็นซิวิตี คือ

$$\frac{T(S)}{S} = \frac{(2\alpha + 1) \cosh P - 2\alpha}{(\alpha + 1) \cosh P - \alpha} \quad \dots(5.11)$$

จากสมการ (5.11) ถ้าสมมติให้ α มีค่าเท่ากับ 1, 2, 3, และ 4 จะได้ค่าเซ็นซิวิตีของอิลิเมนต์ K แสดงดังในรูปที่ 5.11 ซึ่งจะเห็นได้ว่าในช่วงความถี่ต่ำที่ $\omega = 0$ ค่าเซ็นซิวิตีของ K จะมีค่าเท่ากับหนึ่งและในช่วงความถี่สูงที่ ω เท่ากับ infinity ค่าเซ็นซิวิตีของ K จะมีค่าเท่ากับ $(2\alpha+1)/(\alpha+1)$ จะเห็นได้ว่าวงจรอควอไลเซออร์มีค่าเซ็นซิวิตีของอิลิเมนต์ K ต่ำ

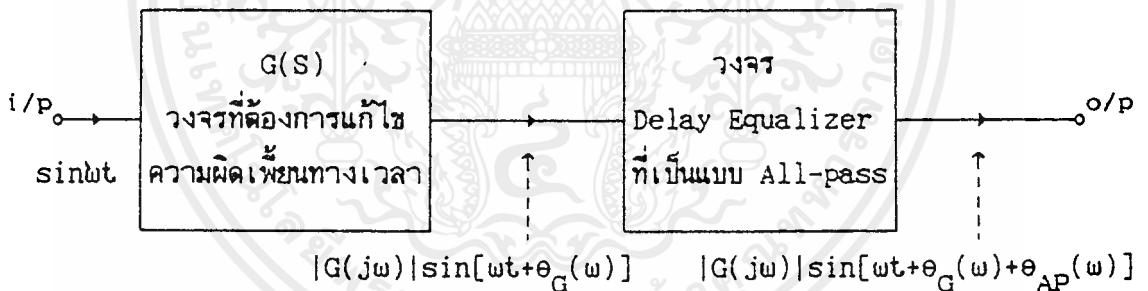


รูปที่ 5.11 แสดง Sensitivity ของอิลิเมนต์ K

5.3 การออกแบบวงจรดีเลย์อีควอลไลเซอร์ (Delay Equalizer)

ในระบบการติดต่อสื่อสารโทรคมนาคม เช่น ระบบรับส่งของสัญญาณโทรทัศนส์และพัลส์ทรานส์มิสชัน (Pulse Transmission) เรามีความจำเป็นที่จะต้องชดเชยทั้งทางเฟสและทางขนาดของสัญญาณ โดยทำให้เฟสที่ได้เป็นเชิงเส้นให้มากที่สุดตามที่ต้องการในช่วงทรานส์มิสชันแบนด์ (Transmission Band) หรืออีกนัยหนึ่งก็คือทำให้กรุปดีเลย์ (Group Delay) คงที่นั่นเอง ในบทนี้จะเสนอหลักในการประมาณฟังก์ชันของวงจรดีเลย์อีควอลไลเซอร์ เพื่อแก้ไขความผิดเพี้ยนทางเวลาเพื่อให้ได้กรุปดีเลย์คงที่ [2] โดยใช้คอมพิวเตอร์ซ้ำอีเตอเรท (Iterate) ในการหาคำตอบ ซึ่งเน็ตเวิร์กฟังก์ชันที่ได้จะต้องเป็นวงจรที่ยอมให้สัญญาณผ่านได้หมดทุกความถี่ (All-pass Network) และทรานส์เฟอร์โคเอฟฟิเชียนของมันเป็นฟังก์ชันที่มีขั้วอยู่ทางซีกขวา และมีโพลอยู่ทางซีกซ้าย ของแกนอิมเมจินนารี (Imaginary Axis) ใน S-Plane

การแก้ไขความผิดเพี้ยนทางเวลากระทำได้โดยการนำเอาวงจรดีเลย์อีควอลไลเซอร์แบบ All-pass ที่สร้างขึ้นมาต่อคาสเคดเข้ากับวงจรที่ต้องการแก้ไขความผิดเพี้ยนทางเวลาตามในรูปที่ 5.12



รูปที่ 5.12 แสดงการนำดีเลย์อีควอลไลเซอร์ไปใช้งานร่วมกับวงจรที่ต้องการแก้ไข

การออกแบบดีเลย์อิคควอลไลเซอร์กำหนดให้ทรานส์เฟอ์ฟังก์ชัน

$$H(S) = \frac{P(-S)}{P(S)} \dots (5.12)$$

โดยที่ P(S) เป็นเฮอรัวิกซ์โพลีโนเมียล

เมื่อเขียนให้อยู่ในแกนของความถี่ (Real Frequency Axis) จะได้

$$H(j\omega) = Me^{j\theta} = \frac{P(-j\omega)}{P(j\omega)} \dots (5.13)$$

เมื่อ M และ θ เป็นขนาดและเฟสตามลำดับ

$$\text{โดยที่ } |P(j\omega)| = |P(-j\omega)|$$

$$\text{และ } -\arg P(j\omega) = \arg P(-j\omega)$$

จากสมการ (5.13) จะได้ $M = 1$ และ $\theta = -2 \arg P(j\omega)$ และเมื่อเขียนเป็นทรานส์เฟอ์ฟังก์ชันอันดับ (Order) ที่ 2 จะได้

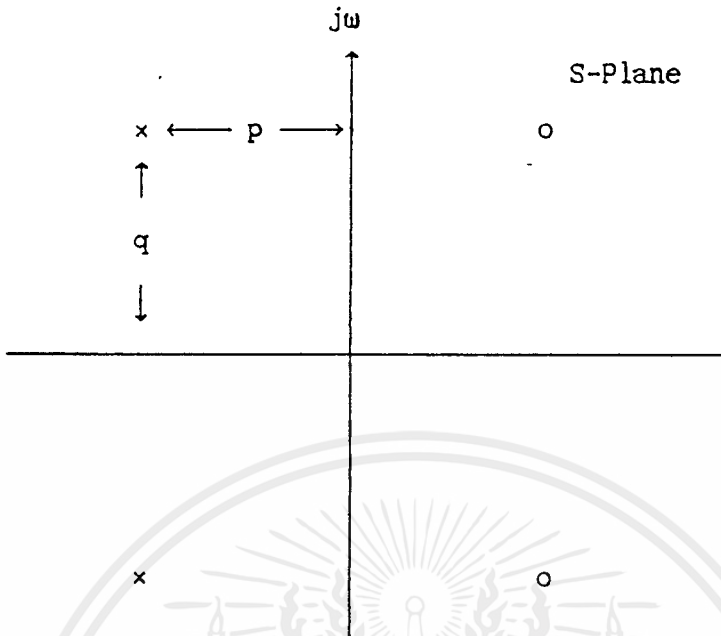
$$H(S) = \frac{S^2 - aS + b}{S^2 + aS + b} \dots (5.14)$$

ทรานส์เฟอ์ฟังก์ชัน H(S) ในสมการ (5.14) ซึ่งอยู่ในรูปความถี่เชิงซ้อน (Complex Frequency) สามารถเขียนตำแหน่งของโพลและซีโรว์ใน S-Plane แสดงได้ดังในรูปที่ 5.13 จากทรานส์เฟอ์ฟังก์ชันในสมการ (5.14) สามารถเขียนได้ใหม่เป็น

$$H(S) = \frac{S^2 - 2pS + p^2 + q^2}{S^2 + 2pS + p^2 + q^2} \dots (5.15)$$

ค่าของเฟสในสมการ (5.15) จะได้

$$\theta = -2 \left(\tan^{-1} \frac{\omega - q}{p} + \tan^{-1} \frac{\omega + q}{p} \right) \dots (5.16)$$



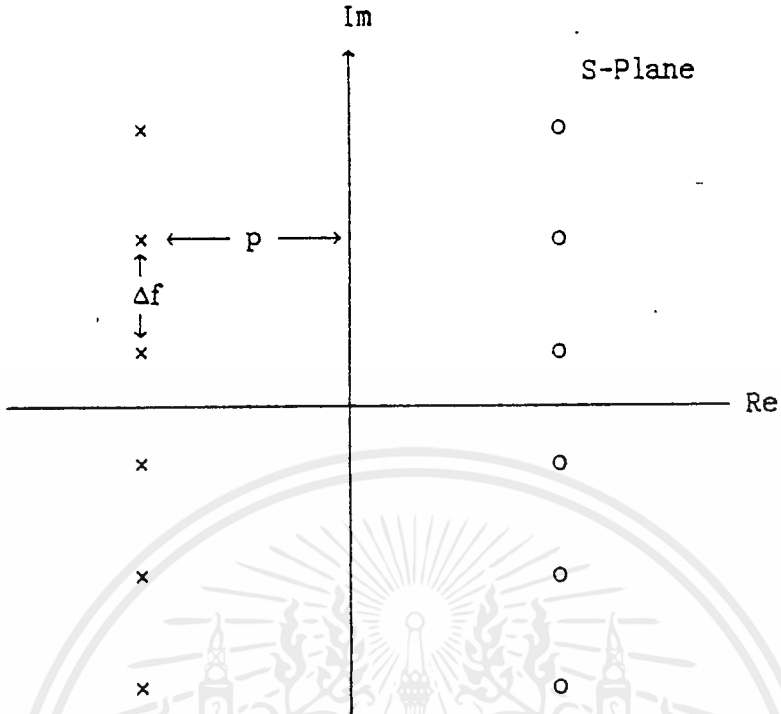
รูปที่ 5.13 แสดงตำแหน่งโพลและซีโรของ All-pass Transfer Function อันดับที่ 2

จากสมการ (5.16) ค่ากรูฟังก์ชันสามารถเขียนได้คือ

$$T_d = \frac{2p}{p^2 + (\omega - q)^2} + \frac{2p}{p^2 + (\omega + q)^2} \quad \dots (5.17)$$

จากสมการ (5.17) จะเห็นได้ค่ากรูฟังก์ชันที่ได้ขึ้นอยู่กับระยะห่างของโพลหรือซีโรใน S-Plane คือตัวแปร p และ q ฉะนั้นการประมาณเน็ทเวอร์กฟังก์ชันเพื่อแก้ไขความผิดเพี้ยนทางเวลาของวงจรที่ต้องการนั้นก็คือการบังคับหาตำแหน่งของโพลหรือซีโรที่ดีที่สุดในช่วงครึ่งโวลต์ของโวลเซอร์ เพื่อให้ได้ผลรวมของกรูฟังก์ชันที่ได้คงที่ ซึ่งจำนวนของโพลหรือซีโร (อันดับของทรานส์เฟอร์ฟังก์ชัน) ขึ้นอยู่กับช่วงแบนด์วิธของความถี่ที่ต้องการและขนาดค่าคลาดเคลื่อนสูงสุดที่ยอมให้ได้ในกรูฟังก์ชันช่วงนั้น

ในการประมาณค่าเริ่มต้นของโพลและซีโรในทรานส์เฟอร์ฟังก์ชันของ All-pass นั้น จะใช้วิธีการกำหนดค่าของโพลและซีโรกระจายขนานไปกับแกน Imaginary ใน S-Plane ดังแสดงในรูปที่ (5.14)



รูปที่ 5.14 แสดงการกำหนดค่าเริ่มต้นของโพลและซีโร

ช่วงระยะห่างระหว่างโพลหรือซีโรนั้น สามารถกำหนดได้จากช่วงแบนด์วิธของความถี่ที่ต้องการกับจำนวนเซคชันของ All-pass อันดับที่ 2 คือ

$$\Delta f = \frac{\text{แบนด์วิธของความถี่}}{\text{จำนวนเซคชัน} - 0.5} \quad \dots (5.18)$$

จากรูปที่ 5.14 จะได้ผลรวมของกรุปดีเลย์คือ

$$T_d = \frac{1}{\pi} \sum_{k=-\infty}^{\infty} \frac{p}{p^2 + \Delta f^2 (f-k)^2}$$

โดยจะปรากฏค่าจุดสูงสุดที่ $f = \pm k$ และจุดต่ำสุดที่ $f = \pm (k-1/2)$ เพราะฉะนั้นเมื่อกำหนดให้ $i = 2k$ จะได้ค่ากรุปดีเลย์ที่จุดสูงสุดและต่ำสุดคือ

$$T_{d_{\max}} = \frac{2}{\pi \Delta f} \sum_{i=1,3,\dots} \frac{(2p/\Delta f)}{(2p/\Delta f)^2 + i^2} = \frac{1}{\Delta f} \tanh \frac{\pi p}{\Delta f}$$

$$T_{d_{min}} = \frac{2}{\pi \Delta f} \sum_{i=0, -2, \dots}^{2, 4, \dots} \frac{(2p/\Delta f)}{(2p/\Delta f)^2 + i^2} = \frac{1}{\Delta f} \coth \frac{p\pi}{\Delta f}$$

ค่าเฉลี่ยของกรุปดีเลย์จะได้

$$T_{d_o} = \frac{1}{2\Delta f} \left(\coth \frac{p\pi}{\Delta f} + \tanh \frac{p\pi}{\Delta f} \right) = \frac{1}{\Delta f} \coth \frac{2p\pi}{\Delta f}$$

ซึ่งจะได้ขนาดค่าคลาดเคลื่อนสูงสุดของกรุปดีเลย์คือ

$$\epsilon = \frac{2}{\Delta f} \left(\coth \frac{p\pi}{\Delta f} - 1 \right) \cong \frac{4}{\Delta f} e^{-2p\pi/\Delta f}$$

และจะได้ระยะห่างของโพลหรือซีโรกับแกน Imaginary คือ

$$p = \frac{\Delta f}{2\pi} \ln \frac{4}{\Delta f \epsilon} \quad \dots (5.19)$$

และจะได้ระยะห่างของโพลหรือซีโรกับแกน Real คือ

$$q_i = \Delta f \left(i - \frac{1}{2} \right) \quad \dots (5.20)$$

โดยที่ i เป็นลำดับที่วงจร All-pass แต่ละชุด (Section)

ตำแหน่งของโพลหรือซีโรในเน็ทเวิร์กฟังก์ชันอันดับที่ 2 ที่ประมาณขั้นนั้น จะมีผลต่อกรุปดีเลย์ คือ ถ้า p มีค่าน้อยจะทำให้ขอดของกรุปดีเลย์มีค่าสูงและมีแบนด์วิธที่แคบ และถ้า p มีค่ามากจะทำให้ขอดของกรุปดีเลย์มีค่าต่ำและมีแบนด์วิธที่กว้าง ส่วนตัวแปร q จะมีผลต่อแกนทางความถี่คือ จะเป็นตำแหน่งของความถี่ ณ. ที่จุดขอดของกรุปดีเลย์นั้น จากค่าประมาณที่ได้จะเป็นค่าเริ่มต้นเพื่อให้คอมพิวเตอร์ช่วยหาค่าต่อไป ซึ่งถ้าค่าเริ่มต้นที่ให้มามีค่าใกล้เคียงกับค่าโพลหรือซีโรที่ต้องการก็จะทำให้จำนวนครั้งในการหาค่าต่อเนื่องสั้นลง ขั้นตอนต่อไปเริ่มจากการกำหนดให้ D เป็นผลรวมของกรุปดีเลย์ทั้งหมด คือ ของวงจรดีเลย์อิควอลไลเซอร์ และวงจรที่ต้องการแก้ไขความผิดเพี้ยนทางเวลา และมีค่าเฉลี่ยคงที่เป็น D_o ด้วยจุดสูงสุดและต่ำสุดเปลี่ยนแปลงจากค่านี้นเท่ากับ $\pm \epsilon/2$ และนั้นที่ความถี่ใด ๆ สมมติให้เป็น j จะได้

$$\begin{aligned} \text{ค่าต่ำสุด:} & \quad D_j + \epsilon/2 = D_o \\ \text{ค่าสูงสุด:} & \quad D_j - \epsilon/2 = D_o \end{aligned} \quad \dots (5.21)$$

ค่าผลรวมของกรุปดีเลย์ที่กำหนดจากค่าเริ่มต้นนั้นจะมีความคลาดเคลื่อน สมมติให้เป็น ΔD และนั้น

จากสมการ (5.21) จะเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าต่ำสุด: $\Delta D_1 + D_1 + \epsilon/2 = D_0 \dots (5.22)$

ค่าสูงสุด: $\Delta D_1 + D_1 - \epsilon/2 = D_0$

ผลรวมของกริฟต์เลเยอร์ทั้งหมด D จะขึ้นอยู่กับโคออดิเนตของ (p,q) ในแต่ละส่วนของทรานส์เฟอร์ฟังก์ชัน All-pass อันดับที่สอง การเปลี่ยนแปลงค่าใด ๆ ของ D ที่ช่วงความถี่ต่าง ๆ ขึ้นอยู่กับการเปลี่ยนค่า ($\Delta p, \Delta q$) ในโคออดิเนตที่เหมาะสม ตัวอย่างเมื่อสมมติให้ (p_0, q_0) เป็นรากเริ่มต้นของระบบสมการ

$$\Delta D(p, q) = 0$$

ถ้า ($p_0 + \Delta p, q_0 + \Delta q$) เป็นรากของระบบสมการแล้วจะได้

$$\Delta D(p_0 + \Delta p, q_0 + \Delta q) = 0$$

ฟังก์ชัน ΔD สามารถหาอนุพันธ์ได้เมื่อกระจายออกเป็นอนุกรมทอซีโซอนุกรมเทย์เลอร์หลายตัวแปรจะได้

$$\Delta D_0 + \Delta p \frac{\partial}{\partial p_0} \Delta D + \Delta q \frac{\partial}{\partial q_0} \Delta D + \dots = 0$$

เมื่อไม่คิดเทอมที่มีอนุพันธ์อันดับสองและสูงกว่าแล้วจะได้ระบบสมการเชิงเส้น

$$\Delta p \frac{\partial}{\partial p_0} \Delta D + \Delta q \frac{\partial}{\partial q_0} \Delta D = -\Delta D_0$$

ดังนั้นรากที่ประมาณได้ใหม่คือ

$$p_1 = p_0 + \Delta p$$

$$q_1 = q_0 + \Delta q$$

ถ้าต้องการหารากที่ดียิ่งขึ้น ต้องดำเนินการกระทำซ้ำหลายหนจนกระทั่งถูกต้องตามที่ต้องการ

จากหลักการดังกล่าว เราสามารถนำเอาวิธีนิวตัน-ราฟสันมาใช้แก้สมการเชิงโค้งทางพีชคณิต 2n ตัวแปรและ ๒ สมการของวงจรีอิลควอลิเซอร์ ที่ความถี่ใด ๆ จะได้ผลต่างของกริฟต์เลเยอร์ ΔD_1 และนำไปหาความสัมพันธ์เพื่อหาการเปลี่ยนแปลงของ ($\Delta p_1, \Delta q_1$) ในโคออดิเนตซึ่งจะมีคุณสมบัติของการลู่ออกที่หลักการเบื้องต้นของวิธีนิวตันคือ เร็วขึ้นก็คือใช้เทย์เลอร์กระจายแต่ละสมการออกเป็นอนุกรม เมื่อตัดเทอมที่มีอันดับสูงกว่าหนึ่งออกแล้วจะกลายเป็นการหารากของระบบสมการเชิงเส้น ซึ่งสามารถเขียนเป็นสมการได้คือ

$$\begin{pmatrix} \frac{\partial}{\partial p_1} D_1 + \frac{\partial}{\partial p_i} D_1 + \dots + \frac{\partial}{\partial p_n} D_1 + \frac{\partial}{\partial q_1} D_1 + \frac{\partial}{\partial q_i} D_1 + \dots + \frac{\partial}{\partial q_n} D_1 \\ \frac{\partial}{\partial p_1} D_j + \frac{\partial}{\partial p_i} D_j + \dots + \frac{\partial}{\partial p_n} D_j + \frac{\partial}{\partial q_1} D_j + \frac{\partial}{\partial q_i} D_j + \dots + \frac{\partial}{\partial q_n} D_j \\ \vdots \\ \frac{\partial}{\partial p_1} D_m + \frac{\partial}{\partial p_i} D_m + \dots + \frac{\partial}{\partial p_n} D_m + \frac{\partial}{\partial q_1} D_m + \frac{\partial}{\partial q_i} D_m + \dots + \frac{\partial}{\partial q_n} D_m \end{pmatrix} \begin{pmatrix} \Delta p_1 \\ \Delta p_i \\ \vdots \\ \Delta p_n \\ \Delta q_1 \\ \vdots \\ \Delta q_n \end{pmatrix} = \begin{pmatrix} \Delta D_1 \\ \Delta D_j \\ \vdots \\ \Delta D_m \end{pmatrix} \dots (5.23)$$

เมื่อ $\frac{\partial}{\partial p_i} D_j$ และ $\frac{\partial}{\partial q_i} D_j$ เป็นการหาอนุพันธ์ของ D เทียบกับ p_i และ q_i ที่ความถี่ j ซึ่งสามารถหาได้คือ

$$\frac{\partial}{\partial p_i} D_j = 2 \left(\frac{(\omega_j + q_i)^2 - p_i^2}{[p_i^2 + (\omega_j + q_i)^2]^2} + \frac{(\omega_j - q_i)^2 - p_i^2}{[p_i^2 + (\omega_j - q_i)^2]^2} \right) \dots (5.24)$$

$$\frac{\partial}{\partial q_i} D_j = 4p_i \left(\frac{\omega_j - q_i}{[p_i^2 + (\omega_j - q_i)^2]^2} - \frac{\omega_j + q_i}{[p_i^2 + (\omega_j + q_i)^2]^2} \right) \dots (5.25)$$

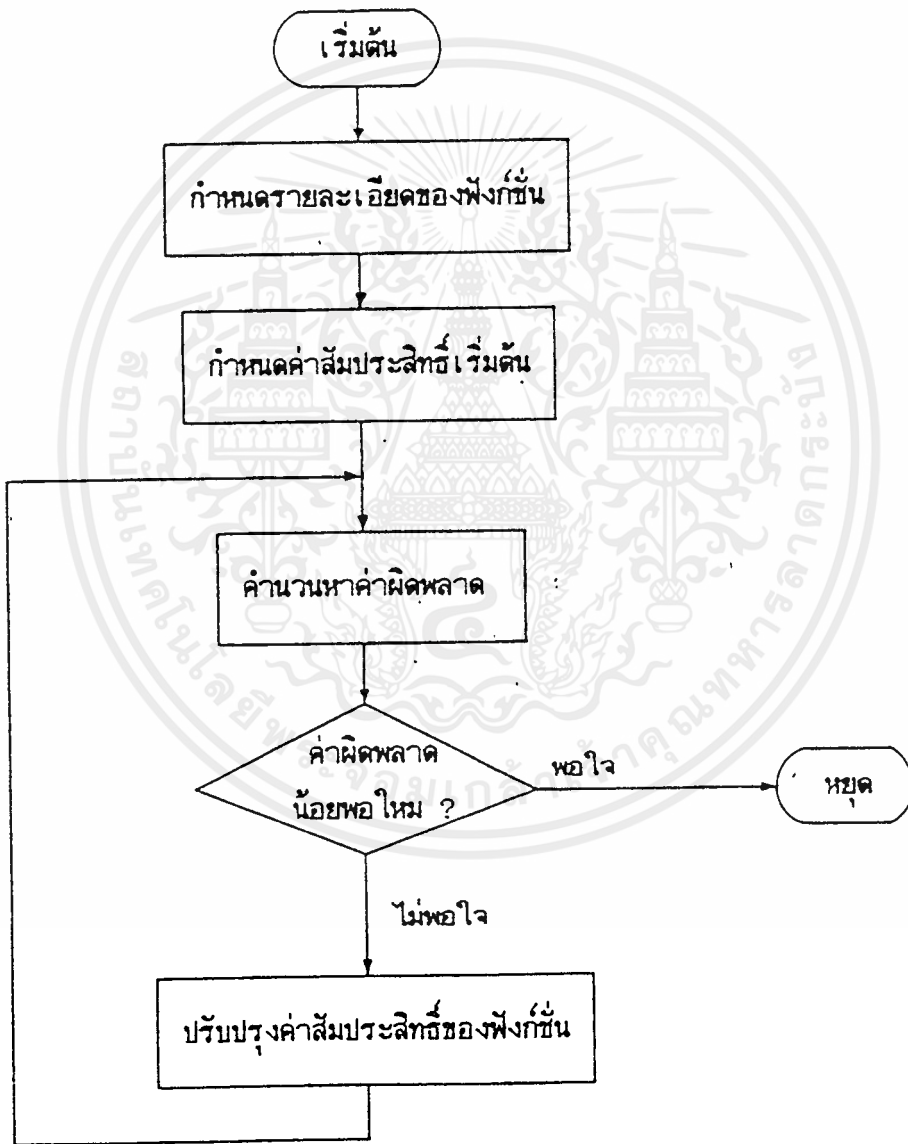
สมการ (5.23) เป็นสมการเชิงเส้นที่จะหาค่าของ Δp_i และ Δq_i เพื่อปรับปรุงค่าของ p_i และ q_i ใหม่ แล้วนำไปแทนค่าเพื่อหาค่าผลรวมของกรูฟังก์ชัน และจะได้ค่า ΔD_j น้อยลงเรื่อยๆ โดยจะกระทำซ้ำหลายหนจนกระทั่ง ΔD_j มีค่าเป็นศูนย์ ในทางปฏิบัติกระบวนการจะกระทำจนกว่าจะได้

$$\text{ค่าต่ำสุด: } \left| D_j + 1/2 - D_0 \right| \leq \text{Tol} \dots (5.26)$$

$$\text{ค่าสูงสุด: } \left| D_j - 1/2 - D_0 \right| \leq \text{Tol}$$

เอกสารนี้เป็นเอกสารที่ขอไว้สำหรับการศึกษาและเพื่อเผยแพร่เท่านั้น (Tolerance) แต่ให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากเซตของสมการที่กล่าวมาจะเห็นได้ว่ามีจำนวนของตัวแปรทั้งหมดเท่ากับ $2n+3$ เมื่อ n เป็นจำนวนเน็ทเวิร์กของวงจร All-pass อันดับที่สองที่มีตัวแปรคือ p_1 และ q_1 และตัวแปรที่เหลือคือ D_0 , ϵ และช่วงแบนด์วิธของความถี่ที่ใช้ ถ้าวงจรดีเลย์อิกวอลไลเซอร์ที่ออกแบบมีช่วงแบนด์วิธของความถี่ที่กว้างและต้องการให้มีค่าคลาดเคลื่อน (ϵ) ที่น้อย ก็จะต้องใช้จำนวนอันดับ (Order) ของเน็ทเวิร์กที่สูงส่วนค่าเฉลี่ยคงที่ของกรุปดีเลย์ (D_0) นั้นถ้ามีค่าน้อยก็ จะทำให้ช่วงแบนด์วิธของความถี่ที่ได้กว้างขึ้น และอาจทำให้มีค่าคลาดเคลื่อน (ϵ) ที่มากขึ้นด้วย



รูปที่ 5.15 โพล์ชาร์ทแสดงการหาสัมประสิทธิ์ของเน็ทเวิร์กฟังก์ชันของวงจรดีเลย์อิกวอลไลเซอร์

แต่ทั้งนี้ก็ต้องขึ้นอยู่กับวงจรที่เราต้องการแก้ไขความผิดเพี้ยนทางเวลาว่ามีความผิดเพี้ยนมากน้อยเพียงใด ซึ่งทั้งหมดนี้จะมีความสัมพันธ์ต่อกันและสามารถเขียนเป็นโปรแกรมเพื่อหาค่าตัวแปรหรือสัมประสิทธิ์ของเน็ตเวิร์กฟังก์ชันของวงจรดีเลย์อิควอไลเซอร์ได้ตั้งแสดงเป็นโพลีชาร์ทในรูปแบบที่

5.17 และมีรายละเอียดของโปรแกรมแสดงไว้ในภาคผนวก

การทำงานของโปรแกรมเพื่อหาค่าผิดพลาดจะมี 2 ช่วง คือ ในตอนแรกจะทำการคำนวณหาค่าสัมประสิทธิ์ของฟังก์ชันจากค่าสัมประสิทธิ์เริ่มต้น เพื่อให้ได้ค่าเฉลี่ยของกรุปดีเลย์ใกล้เคียงกับความเป็นจริงตามที่กำหนดไว้ หลังจากนั้นจึงจะคำนวณหาจุดสูงสุดและต่ำสุดของผลรวมกรุปดีเลย์ทั้งหมด และปรับปรุงค่าสัมประสิทธิ์ของฟังก์ชันใหม่เพื่อให้ได้ค่าคลาดเคลื่อนน้อยที่สุด

5.4 การทดลองใช้งานวงจรอิควอไลเซอร์แยกที่ฟิสิกส์ทวีธาภิเศก เพื่อแก้ความผิดเพี้ยนของสัญญาณโครมिเนนซ์ต่อลุมิเนนซ์ในระบบการส่งสัญญาณภาพของโทรทัศน์

ในระบบการส่งโทรทัศน์สีนั้นสัญญาณภาพรวมจะถูกส่งผ่านวงจรต่างๆ รวมทั้งสายส่งก่อนที่จะถูกส่งออกอากาศ ซึ่งจะมีปัญหามากกับผลตอบสนองของความถี่ (Frequency Response) ของสัญญาณภาพที่ความถี่สูงคือ ทำให้สัญญาณโทรทัศน์มีอัตราขยายทางขนาดของสัญญาณโครมิเนนซ์ไม่เท่ากับของสัญญาณลุมิเนนซ์ ขณะเดียวกันอาจจะทำให้สัญญาณโครมิเนนซ์ถูกหน่วงเวลาต่างกับสัญญาณลุมิเนนซ์ ดังนั้นที่ปลายทางด้านรับจะเกิดผลตอบสนองที่ไม่คงที่ (Non Flat Response) ขึ้น ซึ่งผลดังกล่าวจะทำให้สัญญาณสีผิดเพี้ยนไปจากเดิม ซึ่งความผิดเพี้ยนที่เกิดขึ้นเราสามารถวัดได้โดยใช้สัญญาณมอดูเลทไซน์กำลังสองพัลส์ โดยการป้อนไซน์กำลังสองพัลส์เข้าที่อินพุทของวงจรขยายสัญญาณภาพ แล้วสังเกตสัญญาณเอาต์พุทที่ได้จากออสซิลโลสโคป การใช้ไซน์กำลังสองพัลส์จะให้ผลการทดสอบที่ดีกว่าพัลส์สี่เหลี่ยม และมีความไวต่อความผิดเพี้ยนของอัตราขยายและดีเลย์ ดังนั้นไซน์กำลังสองพัลส์จึงถูกนำมาใช้ทดสอบอยู่เสมอ

การแก้ความผิดเพี้ยนของสัญญาณโครมิเนนซ์ต่อลุมิเนนซ์ทำได้โดยการออกแบบวงจรชดเชยการขยายสัญญาณภาพในย่านความถี่สูง ซึ่งมีความสำคัญมากในระบบการส่งและในระบบโทรทัศน์ การออกแบบวงจรเพื่อแก้ไขความผิดเพี้ยนทางขนาดของสัญญาณโครมิเนนซ์ต่อลุมิเนนซ์ เราสามารถใช้งานวงจรอิควอไลเซอร์แบบแยกที่ฟิสิกส์ทวีธาภิเศกตามในรูปแบบที่ 5.6

ในการทดลองเมื่อกำหนดให้อัตราขยายมีค่า $K = 1$ สมมติต้องการให้ยกหรือลดระดับที่ความถี่สูงเท่ากับ 1dB, 2dB, 3dB และ 4dB ซึ่งจะได้ค่า $\alpha = 8.2, 3.85, 2.42$ และ 1.72 โดยสเกลความถี่ไปที่ 5. KHz จะได้

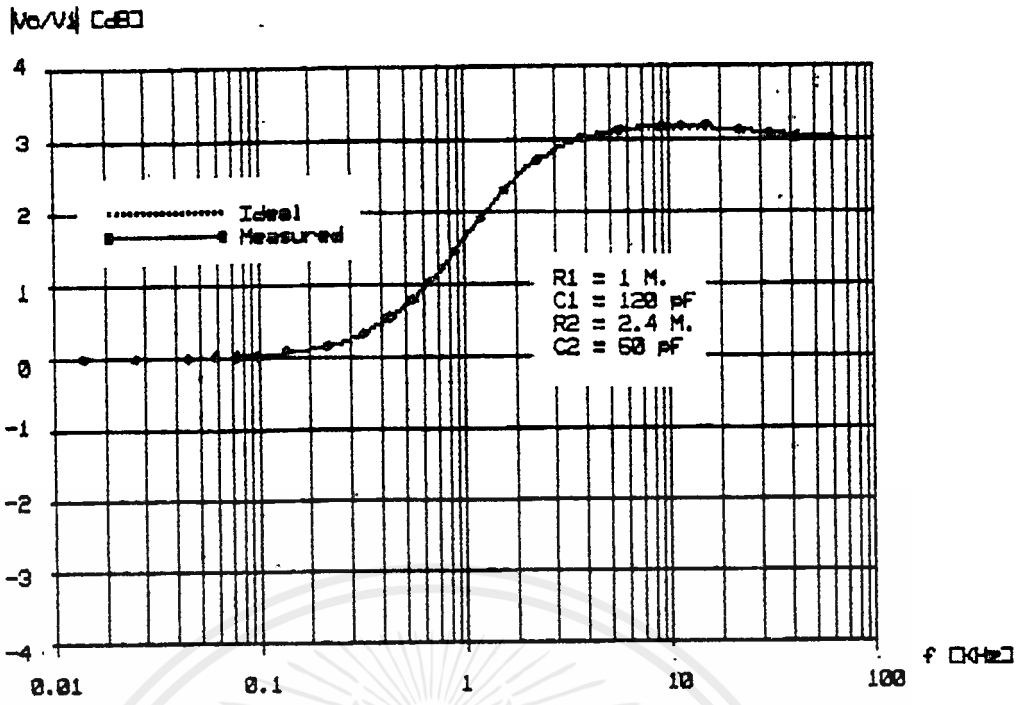
$$RC = 1/(2\pi \times 5 \times 10^3) = 3.183 \times 10^{-5}$$

เมื่อสเกลทางขนาดแล้วจะได้ค่าอุปกรณ์คือ

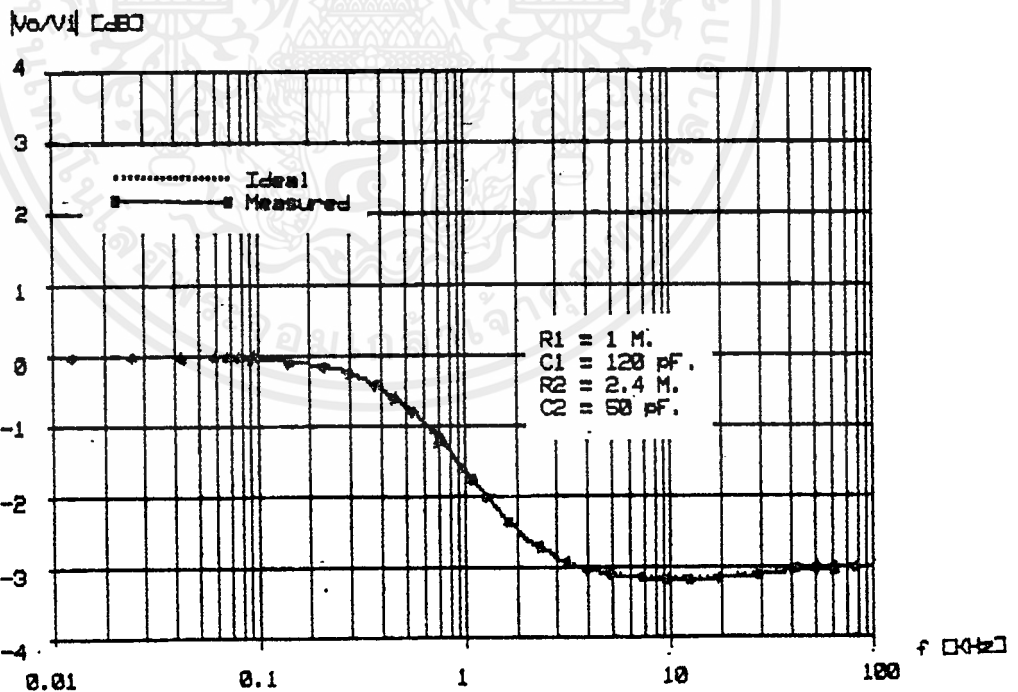
- $R_1 = 265.2 \text{ K ohms}, C_1 = 120 \text{ pF}, R_2 = 2.1 \text{ M ohms}, C_2 = 15 \text{ pF}$
- $R_1 = 63.7 \text{ K ohms}, C_1 = 500 \text{ pF}, R_2 = 245 \text{ K ohms}, C_2 = 130 \text{ pF}$
- $R_1 = 974 \text{ K ohms}, C_1 = 120 \text{ pF}, R_2 = 2.35 \text{ M ohms}, C_2 = 50 \text{ pF}$
- $R_1 = 580 \text{ K ohms}, C_1 = 80 \text{ pF}, R_2 = 1 \text{ M ohms}, C_2 = 47 \text{ pF}$

เมื่อค่า $\alpha = 8.2, 3.85, 2.42$ และ 1.72 ตามลำดับซึ่งจะได้ค่าผลตอบสนองทางขนาดต่อความถี่ที่วัดค่าได้จากวงจรอควอลไอเซอร์แบบแยกที่ฟิสิกส์วิวิทธานี เมื่อต้องการยกและลดระดับที่ความถี่ 3 dB แสดงดังในรูปที่ 5.16 และ 5.17

จากผลการทดสอบวัดค่าผลตอบสนองทางความถี่ของวงจรแยกที่ฟิสิกส์วิวิทธานี ตามที่ได้แสดงไว้แล้วนั้น จะเห็นได้ว่าผลที่ได้มีค่าใกล้เคียงกับอุดมคติมาก สำหรับการทดลองเป็นชนิดวิวิทของวงจร จากวงจรอควอลไอเซอร์แบบแยกที่ฟิสิกส์วิวิทธานีตามในรูปที่ 5.8 เมื่อสมมติต้องการให้ลดระดับที่ความถี่สูงเท่ากับ 2 dB ที่ได้ค่า $R_1 = 63.7 \text{ K}, C_1 = 500 \text{ pF}, R_2 = 245 \text{ K}, C_2 = 130 \text{ pF}$ การทดลองจะสมมติให้ค่าอิลิเมนต์ต่าง ๆ คือ R และ C แต่ละตัวมีความคลาดเคลื่อน +10 % ผลที่ได้จากการวัดผลตอบสนองทางขนาดต่อความถี่แสดงดังในรูปที่ 5.18 และจากรูปจะเห็นได้ว่า ผลตอบสนองทางความถี่ของวงจรมีค่าใกล้เคียงกับอุดมคติ ซึ่งแสดงให้เห็นว่าวงจรแยกที่ฟิสิกส์วิวิทธานีที่ได้จะมีเป็นชนิดวิวิทของอุปกรณ์ค่า

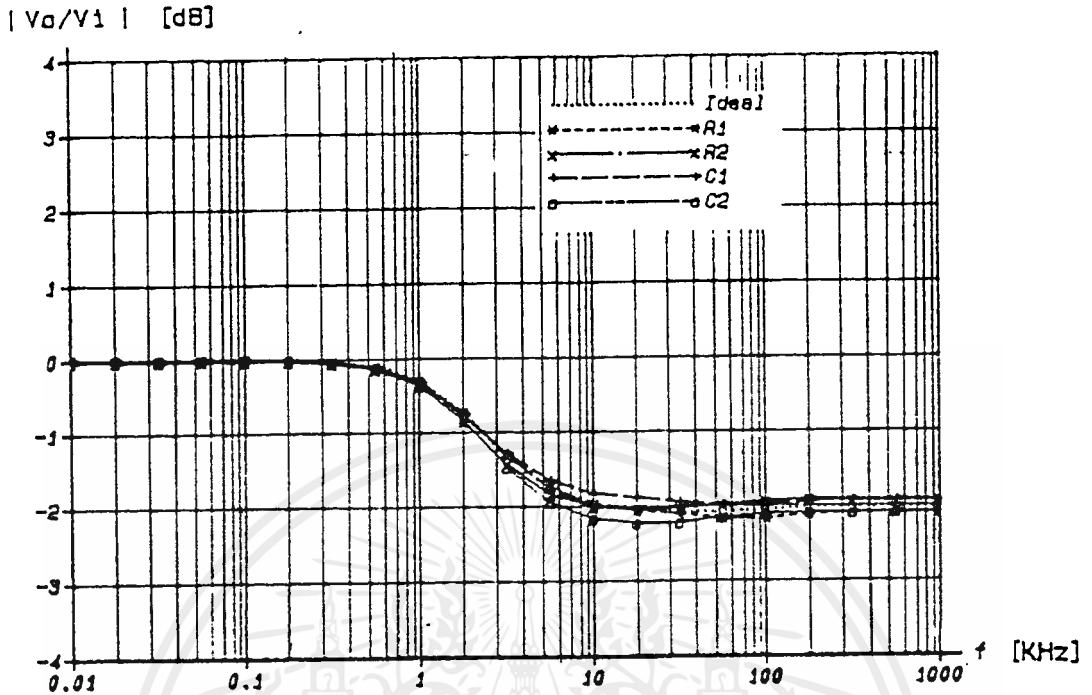


รูปที่ 5.16 แสดงผลตอบสนองทางขนาดต่อความถี่ที่วัดได้จากวงจรรีควอลไดเซอร์ เมื่อต้องการการลดระดับของสัญญาณที่ 3 dB



รูปที่ 5.17 แสดงการตอบสนองทางขนาดต่อความถี่ที่วัดได้จากวงจรรีควอลไดเซอร์ เมื่อต้องการลดระดับของสัญญาณที่ -3 dB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.18 แสดงผลตอบสนองทางขนาดต่อความถี่ที่วัดได้จากวงจรรีควอลไดเซอร์ เมื่อค่า อิมพีแดนซ์ R_1, R_2, C_1, C_2 มีความคลาดเคลื่อน +10%

สำหรับการทดลองวงจรรีควอลไดเซอร์แบบแยกที่ผลิตบริษัทอาซี เพื่อแก้ไขความผิดเพี้ยนของสัญญาณโครมียานซ์ต่อลุ่มิแนนซ์ โดยสมมติให้ต้องการลดขนาดของสัญญาณโครมียานซ์ต่อลุ่มิแนนซ์ เท่ากับ 2dB ซึ่งคำนวณหาค่า α ได้จากสมการ (5.4) โดยจะได้ค่า α เท่ากับ 3.85 เมื่อทำการสแกนทางความถี่ (4.43 MHz) แล้ว จะได้ค่า

$$C_1 = 180 \text{ pF} , R_1 = 310 \text{ ohms}$$

และ

$$C_2 = 680 \text{ pF} , R_2 = 82 \text{ ohms}$$

ส่วนการออกแบบวงจรรีควอลไดเซอร์เพื่อแก้ไขความผิดเพี้ยนทางไทม์ดีเลย์ของสัญญาณโครมียานซ์ต่อลุ่มิแนนซ์ เราสามารถใช้วิธีตามที่กล่าวมาแล้วในเรื่องการออกแบบวงจรรีควอลไดเซอร์ แต่ในที่นี้จะทดลองโดยแก้ไขความผิดเพี้ยนทางไทม์ดีเลย์ของวงจรรีควอลไดเซอร์ที่จะใช้นำมาทดสอบตามในรูปที่ 5.6 ซึ่งวงจรรีควอลไดเซอร์ยังมีค่ากรูฟดีเลย์ไม่คงที่ในช่วงแบนด์วิธทั้งหมดของสัญญาณภาพรวม คือ มีค่าคลาดเคลื่อนรวมประมาณ 31 ns โดยจะแก้ไขความผิดเพี้ยนทางไทม์ดีเลย์ให้มีกรูฟดีเลย์คงที่ในช่วงแบนด์วิธทั้งหมดของสัญญาณภาพ คือ ประมาณ 6 MHz จากสมการ (5.18)

ถ้าใช้จำนวนเซกชันของวงจร All-pass อันดับที่ 2 เท่ากับ 5 จะได้

$$\Delta f = 6/4.5 = 1.333$$

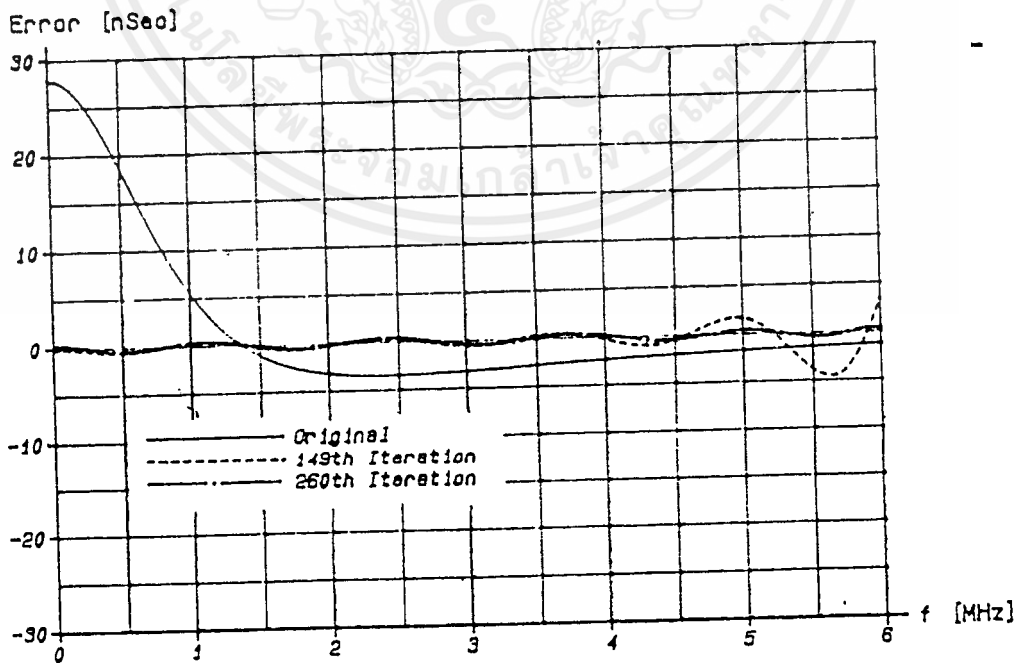
ถ้ากำหนดให้ค่าคลาดเคลื่อน ϵ มีค่าไม่เกิน 5 nS จากสมการ (5.19) จะได้ระยะห่างของตำแหน่งโพลหรือซีโร คือ

$$p = (1.333/2\pi) \ln[4/1.333 \times 0.005] = 1.358$$

และ

$$q_{1-\epsilon} = 0.667, 2, 3.333, 4.667, 6$$

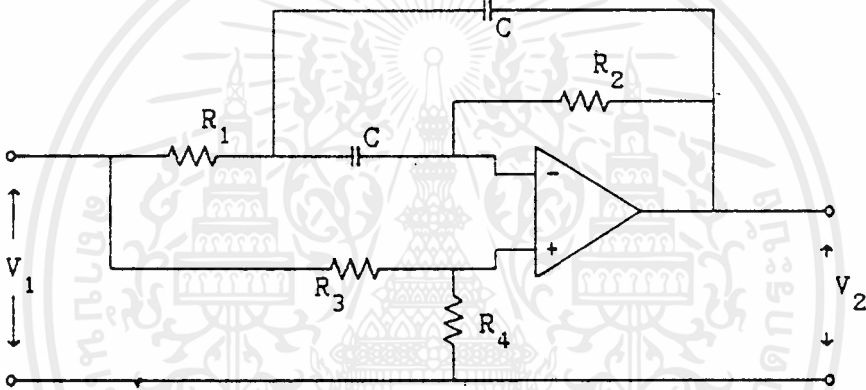
เพื่อลดจำนวนครั้งของการอิตอเรชันให้น้อยลง ซึ่งทำได้โดยจะลดค่าคู่ของโพลหรือซีโรที่เข้าใกล้ขอบของแบนด์ (band edge) ลง 20 % ฉะนั้นจะได้ผลลัพธ์ของโคออดิเนต (p,q) ดังนี้ (1.358, 0.667), (1.358, 2), (1.358, 3.333), (1.358, 4.667), (1.086, 6) หลังจากการอิตอเรทโดยใช้คอมพิวเตอร์ ซึ่งจะใช้จำนวนครั้งในการอิตอเรทช่วงแรกเท่ากับ 149 และช่วงหลังรวมเท่ากับ 260 ครั้ง ผลลัพธ์ที่ได้จะให้รัศมีเคอร์เวเจอร์ในช่วงแบนด์วิดท์ 6 MHz และมีความคลาดเคลื่อนน้อยมากคือไม่ถึง 1 nS ดังแสดงในรูปที่ 5.21 และได้โคออดิเนตของ (p,q) คือ (1.66, 0.988), (1.771, 2.415), (1.738, 3.832), (1.58, 5.179), (1.152, 6.478)



เอกสารนี้เป็นเอกสารที่รูปที่ 5.19 แสดงค่าคลาดเคลื่อนเมื่อแก้ไขความผิดเพี้ยนทางโพลซีโรเป็นการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจร All-pass อันดับที่ 2 สามารถสร้างขึ้นมาได้โดยใช้วงจรแบบพาสซีฟหรือแบบแอคทีฟ การทดลองวงจรดีเลย์อีควอลไลเซอร์จะใช้วงจรแบบแอคทีฟพาสซีฟ [16] ดังในรูปที่ 5.20 ซึ่งสามารถเขียนเป็นทรานส์เฟอร์ฟังก์ชันได้คือ

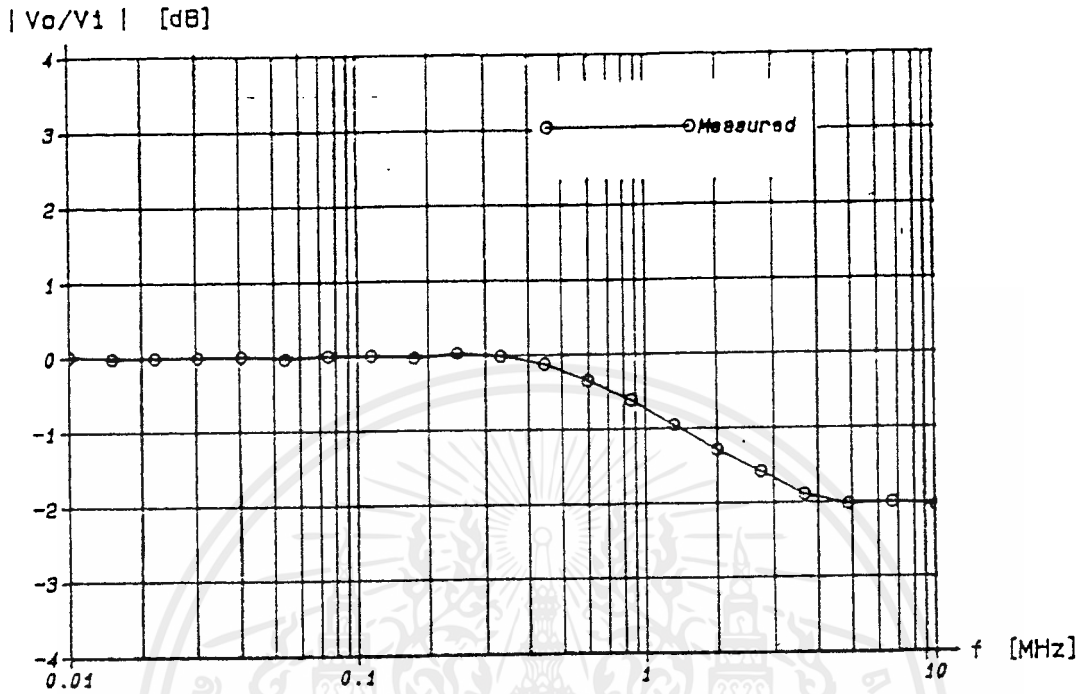
$$\frac{V_2}{V_1} = \frac{R_4}{R_3 + R_4} \frac{S^2 - \frac{2}{R_2 C} S + \frac{1}{R_1 R_2 C^2}}{S^2 + \frac{2}{R_2 C} S + \frac{1}{R_1 R_2 C^2}} \dots (5.27)$$



รูปที่ 5.20 วงจร All-pass อันดับที่ 2

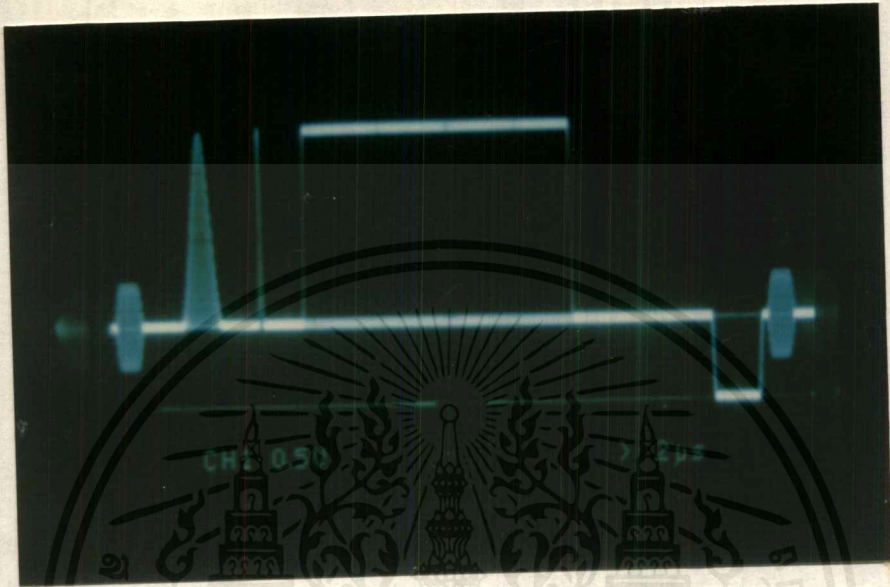
จากค่าโพลและซีโรที่ได้จากการออกแบบวงจรเพื่อแก้ความผิดเพี้ยนทางไทม์ดีเลย์เมื่อแทนค่าลงในสมการ (5.15) และเปรียบเทียบกับสมการที่ได้กับสมการ (5.27) แล้วทำการสเกลทั้งทางขนาดและความถี่ ซึ่งจะได้ค่าของอุปกรณ์แต่ละตัวของวงจร All-pass อันดับที่ 2 แต่ละชุดคือ

- Section 1 $R_1 = 708$, $R_2 = 959$, $R_3 = 2$ K, $R_4 = 677$, $C = 100$ pF
- Section 2 $R_1 = 314$, $R_2 = 899$, $R_3 = 1.6$ K, $R_4 = 1.14$ K, $C = 100$ pF
- Section 3 $R_1 = 158$, $R_2 = 916$, $R_3 = 1.5$ K, $R_4 = 2.2$ K , $C = 100$ pF
- Section 4 $R_1 = 170$, $R_2 = 2.01$ K, $R_3 = 1.6$ K, $R_4 = 4.72$ K, $C = 50$ pF
- Section 5 $R_1 = 141$, $R_2 = 4.61$ K, $R_3 = 1.5$ K, $R_4 = 12.2$ K, $C = 30$ pF



รูปที่ 5.21 แสดงผลตอบสนองทางขนาดต่อความถี่ที่วัดได้จากการทดลอง

จากวงจรที่ได้ทั้งหมดเราจะนำมาต่อคาสเคดกันตามที่แสดงไว้ในรูปที่ 5.12 ซึ่งจะได้ผลตอบสนองทางขนาดต่อความถี่ที่วัดค่าได้จากวงจรแสดงดังในรูปที่ 5.21 และเมื่อป้อนสัญญาณมีอัตรา 20 T ไซน์กำลังสองพัลส์ที่ไม่มีความผิดเพี้ยนเข้าทางอินพุท ของชุดทดสอบก็ จะได้ผลของการทดลองที่เอาท์พุทของวงจรดังรูปที่แสดง

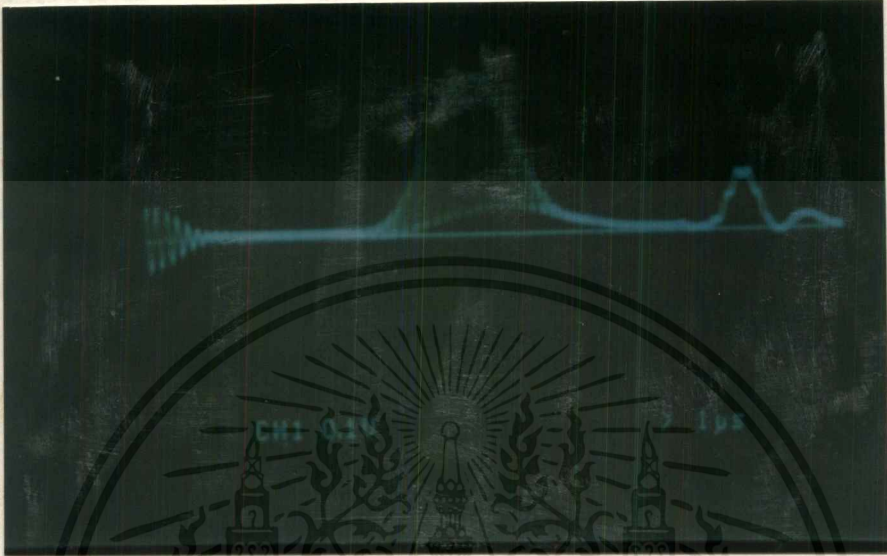


รูปที่ 5.22 สัญญาณทดสอบจาก Pattern Generator(1 Hor-Line)

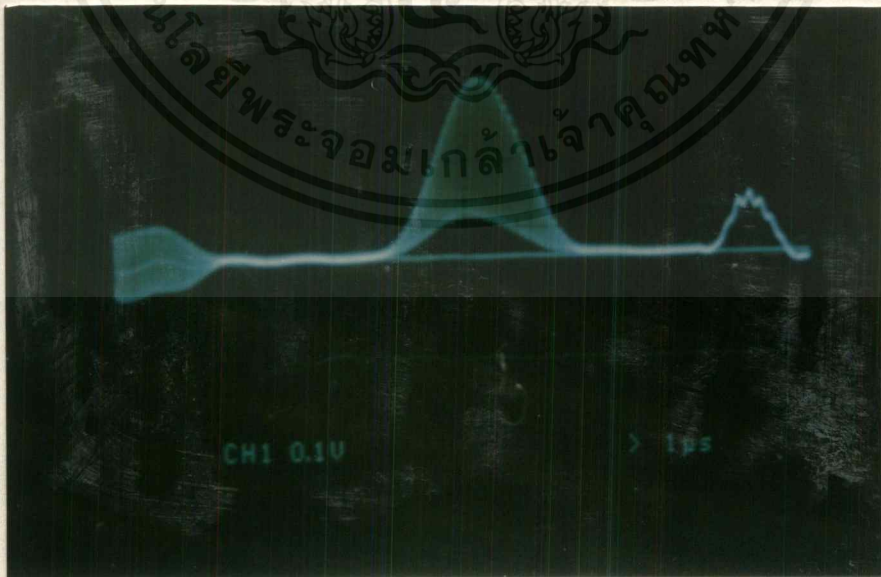


รูปที่ 5.23 สัญญาณรวมโมดูลเลขฐานกลางสองพัลส์ 20T ที่ใช้เป็นสัญญาณอ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.24 แสดงสัญญาณเมื่อคุณดูเลขชี้กำลังสองพัลส์ที่ได้จากวงจรถดลอง
เมื่อนำวงจรถดลองไอซีไอเซอร์ โดสไม่มีการแก้ไขกริฟต์เลย์



รูปที่ 5.25 แสดงสัญญาณเมื่อคุณดูเลขชี้กำลังสองพัลส์ที่ได้จากวงจรถดลอง
เมื่อนำวงจรถดลองไอซีไอเซอร์ และมีกริฟต์เลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

สรุปผลการวิจัย

วงจรแยกที่ฟิสิกส์ทริบิวท์อาชีจะมีคุณลักษณะทั่วไปที่ดีกว่าและมีขนาดเล็กกว่าวงจรที่เป็นพวก ลัมดีโอลิเมนต์ เมื่อเปรียบเทียบกับวงจรรองความถี่ในจำนวนโอลิเมนต์ที่เท่ากันวงจรแบบแยกที่ฟิสิกส์ ทริบิวท์อาชีจะมีสมรรถนะในการลดทอนสัญญาณที่ดีกว่าด้วย วงจรแยกที่ฟิสิกส์ทริบิวท์อาชีที่เสนอมี วงจรรองความถี่ต่ำ , วงจรรองความถี่สูง และ การทดลองวงจรอิควอลไลเซอ์ที่ใช้สักระดับ หรือลดระดับสัญญาณที่ผ่านความถี่สูง โดยที่ในแต่ละวงจรจะมีค่าพารามิเตอร์ที่สามารถกำหนดค่า ของคุณลักษณะผลตอบสนองเปลี่ยนแปลงตามต้องการ โดยที่วงจรดังกล่าวจะมีค่าความไว (Sensi tivity) ของอุปกรณ์ค่า ซึ่งในแต่ละแบบจะมีค่าความไวแตกต่างกัน ส่วนค่า Parasitic Capacitors ซึ่งอาจเกิดขึ้นระหว่างตัวอุปกรณ์ฟิสิกส์ทริบิวท์อาชีกับ Substrate ของไลซีนั้น ใน วงจรรองความถี่ต่ำ , วงจรรองความถี่สูง และวงจรอิควอลไลเซอ์ จะไม่มีผลกระทบต่อคุณ ลักษณะของวงจร จึงสามารถนำวงจรถูกกล่าวไปผลิตขึ้นตามกรรมวิธี LSI ที่มีขนาดเล็กมากได้

เพื่อต้องการแก้ไขการตอบสนองทางเฟส เพื่อให้ได้กริฟต์เลอ์ที่คงที่ ดังนั้นจึงอาศัย วงจรดีเลอ์อิควอลไลเซอ์ ในการหาตำแหน่งของโพลและซีโร โดยการประมาณค่าเริ่มต้นให้ค่า แห่งของโพลและซีโรขนานไปกับแกน $j\omega$ ใน S-Plane และใช้โปรแกรมคอมพิวเตอร์ในภาค ผนวกช่วยในการอิตอเรทในการหาค่าตอบ นอกจากนี้ยังได้เสนอวิธีที่จะนำเอามอดคูลเตชันก่า ลังสองพัลส์ ที่สามารถใช้ในการตรวจสอบความบกพร่องของระบบโทรทัศน โดยการใช่วงจร อิควอลไลเซอ์แบบแยกที่ฟิสิกส์ทริบิวท์อาชีแก้ไขความผิดเพี้ยนของสัญญาณโครมิแนนซ์ต่อลุ่มิแนนซ์ในระ บบการส่งสัญญาณ

กิตติกรรมประกาศ

ขอขอบพระคุณผู้ช่วยศาสตราจารย์ ดร. กนก เจนจิระพงศ์เวช เป็นอย่างสูง ที่ได้ให้การประสิทธิ์ประสาทวิชา การตลอดจนช่วยแนะนำและให้คำปรึกษา ขอขอบคุณท่านอาจารย์ ภาควิชาเทคนิคอุตสาหกรรมทุกท่านที่ให้คำชี้แนะและ ช่วยเหลือจนทำให้ปริญาพันธันจบนี้สำเร็จ ลุล่วงไปด้วยดี



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. ไพบูลย์ สิกขิโยภาสกุล, กนก เจนจิระพงษ์เวช, M. Teramoto, "การออกแบบวงจรอควอลไอเซอร์โดยใช้อ้างอิงจรรยาบรรณนิเทศวิทยาทวีตอาชี", การประชุมทางวิชาการวิศวกรรมไฟฟ้า ครั้งที่ 13 ฌ.มหาวิทยาลัยเชียงใหม่, หน้า 157-166, 8-9 พฤศจิกายน 2533
2. ไพบูลย์ สิกขิโยภาสกุล, ชาลิต เบญจางคประเสริฐ, กนก เจนจิระพงษ์เวช, "การประมาณเน็ทเว็กร์ฟังก์ชันสำหรับวงจรรองความถี่ต่ำแบบที่ใหักรู็ฟต์เล็คคั้ง", การประชุมทางวิชาการวิศวกรรมไฟฟ้า ครั้งที่ 12 ฌ.มหาวิทยาลัยเกษตรศาสตร์, หน้า 202-215, 16-17 พฤศจิกายน 2532
3. ไพบูลย์ สิกขิโยภาสกุล, กนก เจนจิระพงษ์เวช, "วงจรรองความถี่แบบบัคเตอร์เว็กร์ออลคั้งวีลวีลวีลวีลวีลวีลวีล", การประชุมทางวิชาการวิศวกรรมไฟฟ้า ครั้งที่ 10 ฌ.จุฬาลงกรณ์มหาวิทยาลัย, เล่มที่ 2 หน้า 239-247, 24-25 พฤศจิกายน 2530
4. M.Teramoto, S.Sudou and K.Janchitrapongvej, "Realization of An Active Low Pass Filter using Uniform Distributed RC Lines", IECE,CAS 89-54, October 1989.
5. B.K.Ahuja, "Implementation of Active Distributed RC Anti-Aliasing Smoothing Filters", IEEE J.Solid-State Circuit, Vol.SC-17, pp.1076-1080, December 1982.
6. M.S.Ghausi and V.G.Vello, "Active Distributed RC Realization of Low Pass Magnitude Specifications", IEEE Trans. vol CT-16, pp.346-358, August 1969.
7. M.Tera,ptp. "Active Distributed RC Line Filters", JTC-CSCC, 88-C1-2, November 1988.
8. V.G.Bello and M.S.Ghausi, "Design of Linear Phase Active Distributed RC Networks", IEEE Trans.Circuit Theory,vol. CT-16, pp.526-530, November 1969.
9. H.Mahdi , "On Synthesis of Active Distributed RC Circuit", IEEE Trans.Circuit Theory,Vol.CT-17,pp.273-275, May 1970.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. R.P.Sallen and E.L.Key, "A Practical Method of Designing RC Active Filter", IRE Trans. vol.CT-2,no1, pp.74-85, March 1955.
11. M.S.Ghausi and J.J.Kelley, "Introduction to Distributed-parameter Networks", Holt Rinehart and Winston,Inc.,1968.
12. L.P.Huelesman, "Active Filters, Lumped, Distributed, Integrated, Digital, Parametric", McGraw Book, 1970.
13. P.M.Chirlian, "Integrated and Active Network Analysis and Synthesis", Pretice-Hall, 1967.
14. G.C.Temes and S.K.Mitra, " Modern Filter Theory and Design", John Wiley & Sons,Inc., 1973.
15. A.Budak, "Passive and Active network Analysis and Synthesis", Houghton Mifflin Company, 1974.
16. V.Valkenburg, "Analog Filter Design", Holt Saunders,1982.
17. G.C.Temes and J.W.Lapatra "Introduction to Circuit Slynthesis and Design", McGraw-Hill Inc., 1977.
18. "Audio Handbook", National Semiconductor Corp.,PP.2.31-2.59,1967.
19. R.Kennedy, "Sine Squared Pulse in Television System Analysis", RCA Review,Vol.21,No.2,p.253, June 1960.
20. C.A.Siocos, "Chrominance-to-Luminance Ratio and Timing Measurements", IEEE Trans.,BC-14,1,pp.1-4, March 1968.
21. A.Jeffrey, "Mathematics for Engineers and Scientists", ELBS. Nelson, 1978.
22. J.Vlach and K.Singhal, "Computer Methodes for Circuit Analysis and Design", Van Nostrand Reinhold Company,1983.

ภาคผนวก
แสดงการวิเคราะห์ทราฟฟิกเพื่อฟังก์ชันของวงจรและรายละเอียดของโปรแกรม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

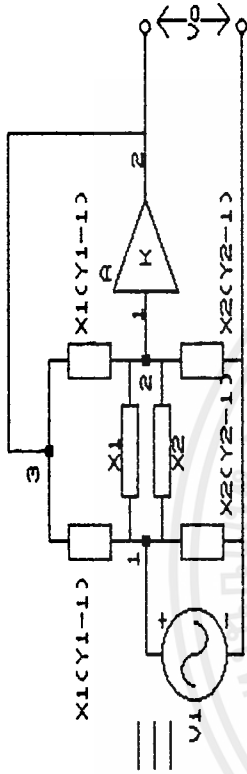
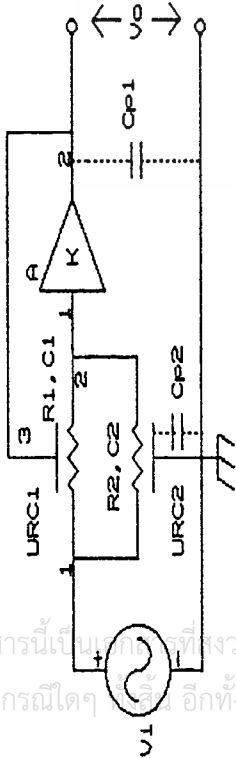


Fig. 3.6 Low-pass filter active distributed RC

From the circuit above; by Floating Matrix model

$$\begin{bmatrix} I1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} X1Y1+X2Y2 & -X1-X2 & -X1(Y1-1) \\ -X1-X2 & X1Y1+X2Y2 & -X1(Y1-1) \\ -X1(Y1-1) & -X1(Y1-1) & 2X1(Y1-1) \end{bmatrix} \begin{bmatrix} V1 \\ V2 \\ V3 \end{bmatrix}$$

Reduce of Matrix

$$\begin{bmatrix} I1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} X1Y1+X2Y2 & -X1-X2 & -X1(Y1-1) \\ -X1-X2 & X1Y1+X2Y2 & -X1(Y1-1) \\ -X1(Y1-1) & -X1(Y1-1) & 2X1(Y1-1) \end{bmatrix} \begin{bmatrix} V1 \\ V2 \\ V3 \end{bmatrix}$$

$$\Delta 1 = \begin{vmatrix} I1 & -X1-X2 & -X1(Y1-1) \\ 0 & X1Y1+X2Y2 & -X1(Y1-1) \\ 0 & X1Y1+X2Y2 & -X1(Y1-1) \end{vmatrix} = I1(X1Y1 + X2Y2 - X1(Y1-1))$$

$$\Delta 3 = \begin{vmatrix} X1Y1+X2Y2 & I1 \\ -X1-X2 & 0 \end{vmatrix} = I1(X1+X2)$$

$$TCP) = V0 = \frac{I1}{\Delta 1} = \frac{X1Y1 + X2Y2 - X1(Y1-1)}{I1(X1+X2)}$$

Substitute $X = R \sqrt{\frac{SRC}{s}}$ and $Y = \cosh \sqrt{SRC}$ and $P = \sqrt{SRC}$ obtain:

$$TCP) = \frac{R1 \sinh P1 + R2 \sinh P2}{K [R1 \cosh P1 + R2 \cosh P2] - P1 \cosh P1 P2} \dots (3.9)$$

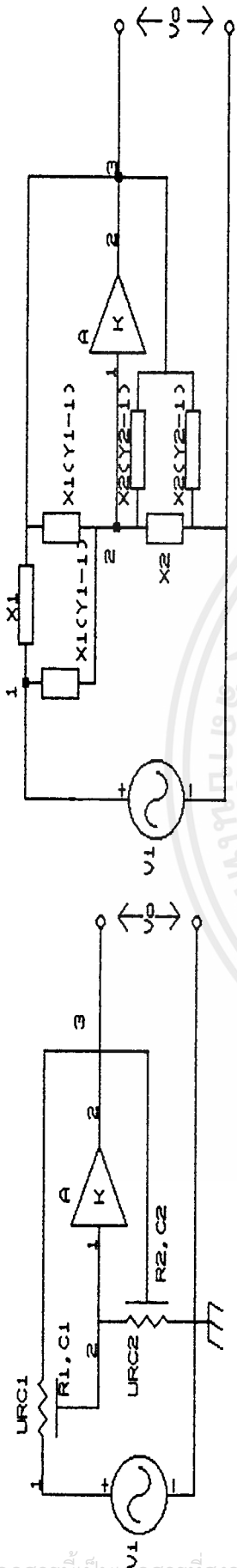


Fig. 3.11 High-pass filter active distributed RC
 From the circuit of high-pass filter active distributed RC; By the floating matrix model

$$\begin{bmatrix} I_1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} X_1 Y_1 & -X_1(Y_1-1) & -X_1 \\ -X_1(Y_1-1) & 2X_1(Y_1-1) + X_2 Y_2 & -X_1(Y_1-1) - X_2(Y_2-1) \\ -X_1 & -X_1(Y_1-1) - X_2(Y_2-1) & X_1 Y_1 + 2X_2(Y_2-1) \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix}$$

Reduce of Matrix which : $V_3 = KV_2$ or $V_2 = \frac{V_3}{K}$

$$\begin{bmatrix} I_1 \\ 0 \end{bmatrix} = \begin{bmatrix} X_1 Y_1 & -X_1(Y_1-1) - X_1 \\ -X_1(Y_1-1) & 2X_1(Y_1-1) + X_2 Y_2 - X_1(Y_1-1) - X_2(Y_2-1) \end{bmatrix} \begin{bmatrix} V_1 \\ V_3 \end{bmatrix}$$

$$\Delta_1 = \begin{vmatrix} I_1 & X_1 Y_1 \\ 0 & 2X_1(Y_1-1) + X_2 Y_2 - X_1(Y_1-1) - X_2(Y_2-1) \end{vmatrix} \text{ and } \Delta_3 = \begin{vmatrix} X_1 Y_1 & \\ & -X_1(Y_1-1) \end{vmatrix} = I_1 \cdot X_1(Y_1-1)$$

$$= I_1 (2X_1(Y_1-1) + X_2 Y_2 - X_1(Y_1-1) - X_2(Y_2-1))$$

So that: $T(p) = V_0 = \frac{\Delta_3}{\Delta_1} = \frac{K \cdot 2X_1(Y_1-1) + X_2 Y_2 - X_1(Y_1-1) - X_2(Y_2-1)}{2X_1(Y_1-1) + X_2 Y_2 - X_1(Y_1-1) - X_2(Y_2-1)}$

Substitute $X = R \sinh \sqrt{SRC}$ and $Y = \cosh \sqrt{SRC}$ and $P = \sqrt{SRC}$ obtain:

$$T(p) = \frac{R_1 \cosh p l_1 - R_2 \sinh p l_2}{R_1 \cosh p l_1 + R_2 \sinh p l_2} = \frac{R_1 \cosh p l_1 - R_2 \sinh p l_2}{R_1 \cosh p l_1 + R_2 \sinh p l_2} \dots (3.22)$$

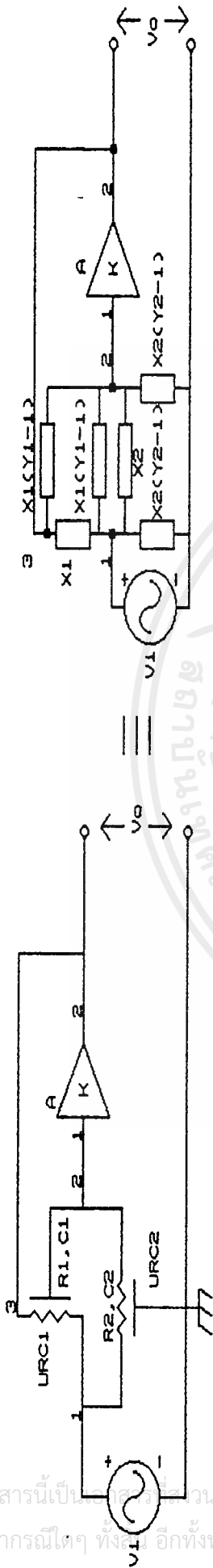


FIG 5.6 The Equalizer Distributed RC

From the circuit above:

$$\begin{bmatrix} I_1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} X_1 Y_1 + X_2 Y_2 & -X_1(Y_1 - 1) - X_2 & -X_1 \\ -X_1(Y_1 - 1) - X_2 & 2X_1(Y_1 - 1) + X_2 Y_2 & -X_1(Y_1 - 1) \\ -X_1 & -X_1(Y_1 - 1) & X_1 Y_1 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix}$$

Noise $X = R \sinh \frac{SRC}{2}$ and $Y = \cosh \frac{SRC}{2}$

$V_1 = \frac{\Delta_2}{\Delta} ; V_2 = \frac{\Delta_3}{\Delta} ; V_3 = \frac{\Delta_4}{\Delta}$

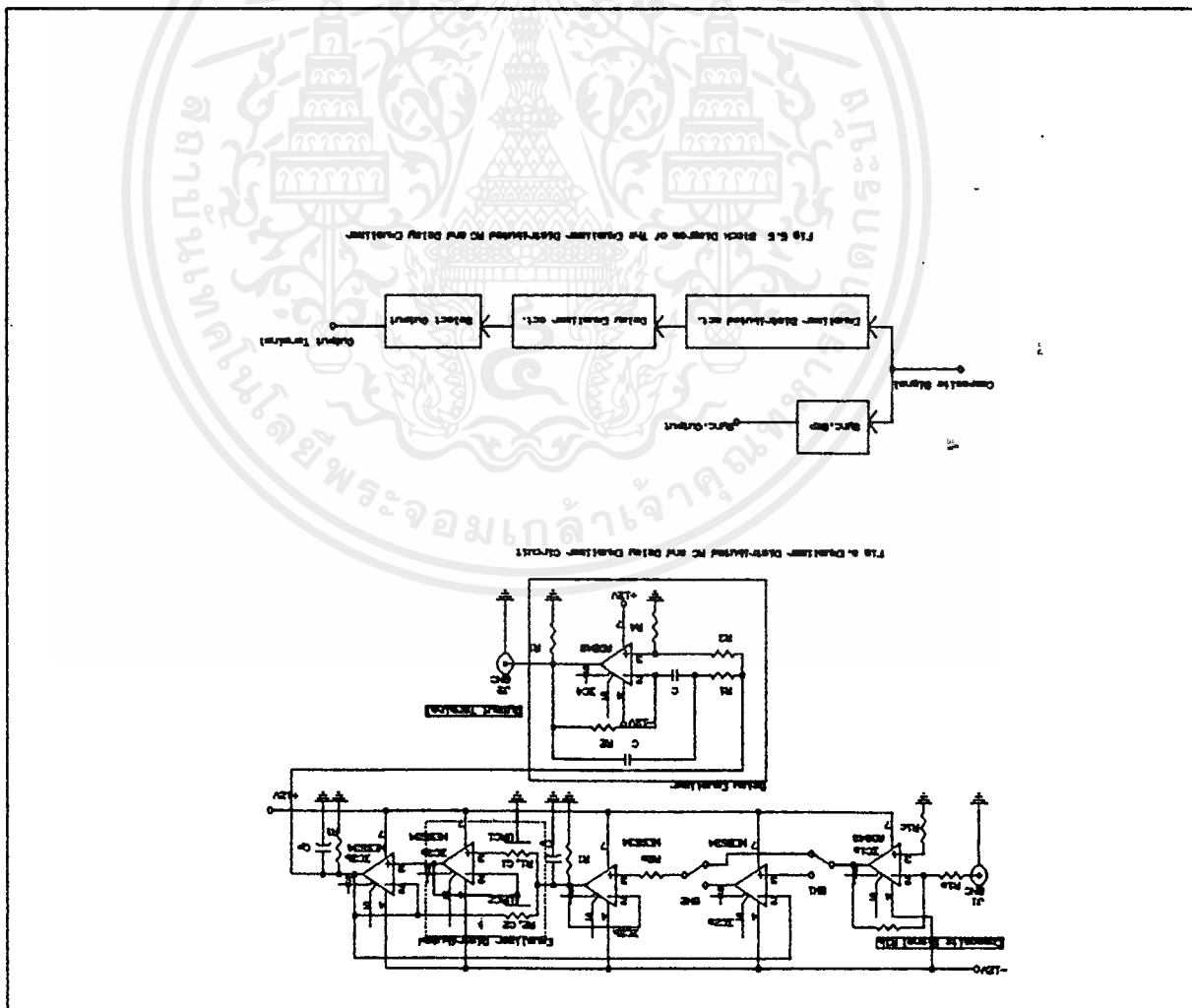
$$\Delta_3 = \begin{vmatrix} X_1 Y_1 + X_2 Y_2 & -X_1(Y_1 - 1) - X_2 \\ -X_1(Y_1 - 1) - X_2 & 2X_1(Y_1 - 1) + X_2 Y_2 \end{vmatrix} = I_{11} X_1 C(Y_1 - 1) + X_2$$

$$\Delta_1 = \begin{vmatrix} I_{11} & -X_1 \\ 2X_1(Y_1 - 1) + X_2 Y_2 & -X_1(Y_1 - 1) \end{vmatrix} = I_{11} C 2X_1(Y_1 - 1) + X_2 Y_2 - X_1 C(Y_1 - 1) X_2$$

$$\Delta_4 = \begin{vmatrix} I_{11} & -X_1 \\ 2X_1(Y_1 - 1) + X_2 Y_2 & -X_1(Y_1 - 1) \end{vmatrix} = 2X_1(Y_1 - 1) X_2 Y_2 - X_1 C(Y_1 - 1) X_2$$

Substitute $X = R \sinh \frac{SRC}{2}$ and $Y = \cosh \frac{SRC}{2}$ and $P = \sqrt{SRC}$ obtain:

$$T(P) = \frac{R \left[\frac{P^2 \cosh P}{\sinh P} + \frac{R^2 P^2 \sinh P}{\sinh P} \right] - \frac{P^2 \sinh P}{\sinh P}}{\dots (5.1)}$$



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
program Plot_2D_Function;
```

```
{ $I hcg.INC }
```

```
const PScale = 25; { 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 := 2 { _ _ _ _ _ _ _ _ }  
    else Write(OutFile, 'LT2, ', 0.20*PScale:6:4, ', ');  
3 :if ScreenDisp then LineStyle := 3 { _____ }  
    else Write(OutFile, 'LT4, ', 0.9*PScale:6:4, ', ');  
4 :if ScreenDisp then LineStyle := 4 { _____ }  
    else Write(OutFile, 'LT3, ', 0.36*PScale:6:4, ', ');  
5 :if ScreenDisp then LineStyle := 5 { _____ }  
    else Write(OutFile, 'LT6, ', 0.9*PScale:6:4, ', ');  
6 :if ScreenDisp then LineStyle := 6 { _____ }  
    else Write(OutFile, 'LT5, ', 0.9*PScale:6:4, ', ');  
7 :if ScreenDisp then LineStyle := 7 { ..... }  
    else Write(OutFile, 'LT1, ', 0.09*PScale:6:4, ', ');  
8 :if ScreenDisp then LineStyle := 8 { _____ }  
    else Write(OutFile, 'LT;SM;');
```

```
end;
```

```
end;
```

```
{ $I PLOT2D.INC }
```

```
overlay 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;
```

```
t := sqrt(abs(w)*RC/2);
```

```
case typ of
```

```
'K':begin
```

```
    m := 7;
```

```
    if fi <> 1 then
```

```
        begin
```

```
            if fi < m then
```

```
                begin
```

```
                    j := fi-1;
```

```
                    n := j;
```

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ หากมีให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end
else
begin
  j := 1;
  n := m-2;
end;
for i := j to n do
begin
  case i of
    1: begin a := 1.66018; b := 0.98794; end;
    2: begin a := 1.77084; b := 2.41510; end;
    3: begin a := 1.73841; b := 3.83236; end;
    4: begin a := 1.58011; b := 5.19676; end;
    5: begin a := 1.15152; b := 6.47771; end;
  end;
  w0ap := sqrt(sqr(a)+sqr(b));
  Qap := w0ap/(2*a);
  div_c( -sqr(w) + sqr(w0ap) , -w*(w0ap/Qap) ,
        -sqr(w) + sqr(w0ap) , w*(w0ap/Qap) , data );
  if fi=m then
  begin
    if i>1 then
      mul_c( data.re , data.im , dl.re , dl.im , data );
      dl := data;
    end;
  end;
end;
if (fi=1) or (fi=m) then
begin
  al := 3.85;
  if w>=0 then
  begin
    mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , al , 0 , dl);
    mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , al+1 , 0 , d2);
  end
  else
  begin
    mul_c( cosh(t)*cos(t) , -sinh(t)*sin(t) , al , 0 , dl);
    mul_c( cosh(t)*cos(t) , -sinh(t)*sin(t) , al+1 , 0 , d2);
  end;
  add_c( dl.re , dl.im , -al+1 , 0 , dl );
  add_c( d2.re , d2.im , -al , 0 , d2 );
  if fi=1 then
    div_c( dl.re , dl.im , d2.re , d2.im , data )
  else
  begin
    div_c( dl.re , dl.im , d2.re , d2.im , dl );
    mul_c( data.re , data.im , dl.re , dl.im , data );
  end;
end;
end;
'@':begin
  case fi of
    1: begin
      al := 4;
      add_c( cosh(t)*cos(t) , sinh(t)*sin(t) , -1+al , 0 , dl);
      div_c( al , 0 , dl.re , dl.im , data );
    end;
    2: begin
      w := w/15.92696274;
      div_c( 1 , 0 , -w*w + 1 , 1.41421356*w , data );
    end;
    3: begin
      w := w/15.92696274*1.16877089;

```

เอกสารนี้เป็นเอกสารส่วนใจสำหรับฉบับงานที่ควรศึกษาค้นคว้าให้เข้าใจ; โยชนด้านการค้า

ไม่ว่ากรณีใดๆ ล้วน อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    div_c( 0.70710678, 0 , -w*w+0.70710678, 0.64359425*w , data );
end;
end;
end;
'A':begin
  case fi of
    1: al := 1;
    2: al := 2;
    3: al := 3;
    4: al := 4;
  end;
  k := 1;
  mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , 1+1/al-k , 0 , d2);
  mul_c( d2.re , d2.im , 1/k , 0 , d2 );
  add_c( d2.re , d2.im , 1 , 0 , d2 );
  div_c( 1+1/al , 0 , d2.re , d2.im , data );
end;
'B':begin
  case fi of
    1: al := 1;
    2: al := 2;
    3: al := 4;
    4: al := 6;
  end;
  k := 1;
  mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , 1+1/al-k , 0 , d2);
  mul_c( d2.re , d2.im , 1/k , 0 , d2 );
  add_c( d2.re , d2.im , +1-1/al , 0 , d2 );
  div_c( 1 , 0 , d2.re , d2.im , data );
end;
'C':begin
  case fi of
    1: al := 1;
    2: al := 2;
    3: al := 4;
    4: al := 6;
  end;
  k := 1;
  mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , 1/k+2/k*1/al-1-1/al , 0 , d2);
  add_c( d2.re , d2.im , -2/k*1/al+1+1/al , 0 , d2 );
  div_c( 1 , 0 , d2.re , d2.im , data );
end;
'D':begin
  case fi of
    1: al := 1;
    2: al := 2;
    3: al := 4;
    4: al := 6;
  end;
  k := 1;
  mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , 1/al+1-k , 0 , d2);
  mul_c( d2.re , d2.im , 1/k , 0 , d2 );
  add_c( d2.re , d2.im , -1/al+1 , 0 , d2 );
  div_c( 1 , 0 , d2.re , d2.im , data );
end;
'E':begin
  case fi of
    3: al := 1;
    4: al := 2;
    2: al := 4;
    1: al := 3.846;
  end;
  k := 1;
  mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , -1 , 0 , d1);
  add_c( cosh(t)*cos(t) , sinh(t)*sin(t) , -1 , 0 , d1);

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆก็ตาม หากมีให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mul_c( cosh(t)*cos(t),sinh(t)*sin(t), 2/k+1/k*1/a1-1-1/a1,0, d2);
add_c( d2.re , d2.im , -2/k+1+1/a1 , 0 , d2 );
div_c( dl.re , dl.im , d2.re , d2.im , data );
end;
'F':begin
  case fi of
    1: a1 := 1;
    2: a1 := 2;
    3: a1 := 4;
    4: a1 := 6;
  end;
  k := 1;
  add_c( cosh(t)*cos(t) , sinh(t)*sin(t) , -1 , 0 , dl);
  mul_c( cosh(t)*cos(t),sinh(t)*sin(t), 1+1/a1-k*1/a1,0 , d2);
  mul_c( d2.re , d2.im , 1/k , 0 , d2 );
  add_c( d2.re , d2.im , -1+1/a1 , 0 , d2 );
  div_c( dl.re , dl.im , d2.re , d2.im , data );
end;
'G':begin
  case fi of
    1: a1 := 1;
    2: a1 := 2;
    3: a1 := 4;
    4: a1 := 6;
  end;
  k := 1;
  mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , 1+1/a1 , 0 , dl);
  add_c( dl.re , dl.im , -1-1/a1 , 0 , dl );
  mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , 2+1/a1-k , 0 , d2);
  mul_c( d2.re , d2.im , 1/k , 0 , d2 );
  add_c( d2.re , d2.im , -2/k+1 , 0 , d2 );
  div_c( dl.re , dl.im , d2.re , d2.im , data );
end;
'H':begin
  case fi of
    1: a1 := 1;
    2: a1 := 2;
    3: a1 := 4;
    4: a1 := 6;
  end;
  k := 1;
  mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , 1+1/a1 , 0 , dl);
  add_c( dl.re , dl.im , -1-1/a1 , 0 , dl );
  mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , 1+1/a1 , 0 , d2);
  mul_c( d2.re , d2.im , 1/k , 0 , d2 );
  add_c( d2.re , d2.im , -1 , 0 , d2 );
  div_c( dl.re , dl.im , d2.re , d2.im , data );
end;
'I':begin
  case fi of
    1: a1 := 1.71;
    2: a1 := 2.42;
    3: a1 := 3.85;
    4: a1 := 8.2;
  end;
  k := 1;
  add_c( cosh(t)*cos(t) , sinh(t)*sin(t) , -1+1/a1 , 0 , dl);
  mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , 1+1/a1 , 0 , d2);
  mul_c( d2.re , d2.im , 1/k , 0 , d2 );
  add_c( d2.re , d2.im , -1 , 0 , d2 );
  div_c( dl.re , dl.im , d2.re , d2.im , data );
end;
'J':begin
  case fi of
    4: a1 := 1.71;

```

เอกสารนี้เป็นทรัพย์สินของสถาบันวิจัยดาราศาสตร์แห่งชาติ ไม่อาจนำออกเผยแพร่โดยไม่ได้รับอนุญาต

โปรดแจ้งให้ทราบ หากมีข้อผิดพลาดหรือต้องการข้อมูลเพิ่มเติม กรุณาติดต่อเจ้าหน้าที่ที่เกี่ยวข้อง

'J':begin

case fi of

4: a1 := 1.71;

```

    2: a1 := 2.42;
    3: a1 := 3.85;
    1: a1 := 2.404;
end;
k := 1;
add_c( cosh(t)*cos(t) , sinh(t)*sin(t) , -1+1/a1 , 0 , d1);
mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , 2+1/a1-k , 0 , d2);
mul_c( d2.re , d2.im , 1/k , 0 , d2 );
add_c( d2.re , d2.im , -2/k+1 , 0 , d2 );
div_c( d2.re , d2.im , d1.re , d1.im , data );
end;
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;
    if (Error = 0) and (Found = false) then
    begin
        Iter := 0;
        TargetF(LeftEndpoint);
        yLeft := sqrt( sqr(data.re) + sqr(data.im) ) - 1/sqrt(2);

```

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาต
 ๒๕๖๓

```

while not(Found) and (Iter < MaxIter) do
begin
  Iter := Succ(Iter);
  MidPoint := (LeftEndpoint + RightEndpoint) / 2;
  TargetF(MidPoint);
  yMidPoint := sqrt( sqr(data.re) + sqr(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;

```

```

procedure S_Plane(al : real);
var M,A, argA : real;
begin
  case typ of
    'A': M := al;
    'B'..'D': M := al-1;
    'E','F': M := (-al+1)/al;
    'G','H': M := -al/(al+1);
    'I','J': M := -al/(al+1);
    '●': M := 0;
  end;
  if sqr(M) >= 1 then
    begin
      A := abs (-M + sqrt( sqr(M)-1 ) );
      argA := 0;
    end
  else
    begin
      A := abs( sqrt( sqr(-M) + 1-sqr(M) ) );
      if M <> 0 then argA := ArcTan( sqrt( 1-sqr(M) ) / -M )
        else argA := ArcTan( -1e30 );
    end;
  if M >= 0 then
    if fi <= 2 then argA := argA-pi
      else argA := argA+pi;
  data.re := sqr( ln(A) ) - sqr(argA + 2*ni*pi);
  data.im := 2 * ln(A) * (argA + 2*ni*pi);
end;

```

```

procedure pole(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;
  S_Plane(OldX);
  OldY := data.re;
  S_Plane(X);
  Y := data.re;
  if ABS(OldY) <= NearlyZero then
    begin
      X := OldX;
      Y := OldY;
      Found := true;
    end
  else
    if ABS(Y) <= NearlyZero then
      Found := true
    else
      if ABS(OldY - Y) <= NearlyZero then
        Error := 2; { Slope of line is zero; no intercept }
      while not(Found) and (Error = 0) and (Iter < MaxIter) do
        begin
          Iter := Succ(Iter);
          NewX := X - Y * (X - OldX) / (Y - OldY);
          S_Plane(NewX);
          NewY := data.re;
          Found := TestForRoot(NewX, OldX, NewY, Tol);
          OldX := X;
          OldY := Y;
          X := NewX;
          Y := NewY;
        end;
        Root := X;
        if not(Found) and (Error = 0) and (Iter >= MaxIter) then
          Error := 1;
        end;
      end;

procedure Nyquist(w : real);
var t      : real;
    d1,d2  : complex;
begin
  t := sqrt(abs(w)/2);
  case typ of
  'A':begin
    if w>=0 then
      begin
        add_c( cosh(t)*cos(t) , sinh(t)*sin(t) , a1 , 0 , data );
      end
    else
      begin
        add_c( cosh(t)*cos(t) , -sinh(t)*sin(t) , a1 , 0 , data );
      end;
    end;
  'B'..'D':begin
    if w>=0 then
      begin
        add_c( cosh(t)*cos(t) , sinh(t)*sin(t) , -1+a1 , 0 , data );
      end
    end
  end;
end;

```

```

else
begin
add_c( cosh(t)*cos(t) , -sinh(t)*sin(t) , -1 +al , 0 , data );
end;
end;
'E','F':begin
if w>=0 then
begin
add_c( al*cosh(t)*cos(t) , al*sinh(t)*sin(t) , -al+1, 0 , data );
end
else
begin
add_c( al*cosh(t)*cos(t) , al*(-sinh(t)*sin(t)) , -al+1, 0 , data );
end;
end;
'G','H':begin
if w>=0 then
begin
mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , al+1 , 0 , dl);
sub_c( dl.re , dl.im , al , 0 , data );
end
else
begin
mul_c( cosh(t)*cos(t) , -sinh(t)*sin(t) , al+1 , 0 , dl);
sub_c( dl.re , dl.im , al , 0 , data );
end;
end;
'I','J':begin
if w>=0 then
begin
mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , al+1 , 0 , dl);
add_c( dl.re , dl.im , -al , 0 , data );
end
else
begin
mul_c( cosh(t)*cos(t) , -sinh(t)*sin(t) , al+1 , 0 , dl);
add_c( dl.re , dl.im , -al , 0 , data );
end;
end;
'O':begin
if w>=0 then
begin
mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , al -1 , 0 , dl);
add_c( dl.re , dl.im , 1 , 0 , data );
end
else
begin
mul_c( cosh(t)*cos(t) , -sinh(t)*sin(t) , al -1 , 0 , dl);
add_c( dl.re , dl.im , 1 , 0 , data );
end;
end;
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, Iiter : 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
    begin
      X := OldX;
      Y := OldY;
      Found := true;
    end
  else
    if ABS(Y) <= NearlyZero then
      Found := true
    else
      if ABS(OldY - Y) <= NearlyZero then
        Error := 2;
      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;
          OldY := Y;
          X := NewX;
          Y := NewY;
        end;
        Root := X;
        if not(Found) and (Error = 0) and (Iter >= MaxIter) then
          Error := 1;
        end;
      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;

```

```

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]) <=
    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,wl,
      dx_max,dx_min,t,ph0,
      phl,dy_max,dy_min,Root : real;
      Error : byte;
      err1,err2 : boolean;
      dl,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;
  RC := 1;
  case num of
    '2'..'5': begin
      if freq then write(' 2πRC = ',2*pi*rc:15:8,' ---> ');
      write('RC = ',rc:15:8,' : '); readln(rc);
      if freq then rc := 2*pi*rc;
      end;
    '6': if (ord(typ)and$20)=0 then
      begin
        al := 1;
        write(' alpha = ',al:15:8,' : '); readln(al);
      end;
    '7': begin
      wl := 1;
      ww := 4;
      ni := 0;
      write(' al 1 = '); readln(wl);
      write(' al 2 = '); readln(ww);
      write(' n = '); readln(ni);
    end;
  end;

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาตให้ไปใช้ประโยชน์ด้านการค้า
 ไม่ควรนำเอกสารนี้ไปเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end;
'8': begin
  ali[0] := 1;
  wwi[0] := 22;
  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;
end;
GraphicsMode;
SelectScreen(2);
ClearScreen;
SelectScreen(1);
logs := false;
dB_log := false;
if not freq then
  begin
    xx := ^N+'x'+^O;
    xs := 3;
  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
'2':begin
  stl[1] := ^N+'a'+^O+' = 1';
  stl[2] := ^N+'a'+^O+' = 2';
  stl[3] := ^N+'a'+^O+' = 4';
  stl[4] := ^N+'a'+^O+' = 6';
  logs := true;
  dB_log := true;
  DefineWorld(0.06283,-4, 6283,4, xs,0);
  Axis(5,8, 9,1, 7,5, 'Fig.          Magnitude response',
      xx, '!T! [dB]');
end;
'3':begin
  stl[1] := ^N+'a'+^O+' = 1';
  stl[2] := ^N+'a'+^O+' = 2';
  stl[3] := ^N+'a'+^O+' = 4';
  stl[4] := ^N+'a'+^O+' = 6';
  logs := true;
  DefineWorld(0.01,-500, 500,0 ,xs,0);
  Axis(5,10, 9,1, 6,5, 'Fig.          Phase response',
      xx, ^N+'h'+^O+' [deg]');
end;
'4', '5':begin
  if num='4' then dy := 'Group Delay'
  else dy := 'Phase Delay';
  stl[1] := ^N+'a'+^O+' = 1';
  stl[2] := ^N+'a'+^O+' = 2';
  stl[3] := ^N+'a'+^O+' = 4';
  stl[4] := ^N+'a'+^O+' = 6';
  DefineWorld(0.01,-500, 500,0 ,xs,0);
  Axis(5,10, 9,1, 6,5, 'Fig.          ' + dy,
      xx, ^N+'h'+^O+' [deg]');
end;

```

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอญตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่มีการให้บริการใดๆ ให้ด้วย

```

stl[3] := ^N+'a'+^O+' = 4';
stl[4] := ^N+'a'+^O+' = 6';
logs := true;
DefineWorld(0.1,0, 1000,1, xs,-xs);
Axis(5,3, 9,2, 6,5, 'Fig.      '+dy,xx,'D [sec]');
end;
'6':begin
  Str(al:1:0,alp[0]);
  stl[1] := 'R1';
  stl[2] := 'C1';
  stl[3] := 'R2';
  stl[4] := 'C2';
  logs := true;
  DefineWorld(0.01,-5, 1000,5, 0,0);
  Axis(5,4, 9,5, 2,2, ' ',xx,'Sensitivity');
end;
'7':begin
  fn := 2;
  n := n*5;
  point := 3;
  ali[0] := 0;
  ali[1] := 0.5;
  ali[2] := 0.92058;
  ali[3] := 0.99;
  alp[0] := ^N+'a'+^O+' = 0';
  alp[1] := ^N+'a'+^O+' = 0.5';
  alp[2] := ^N+'a'+^O+' = 0.92058';
  alp[3] := ^N+'a'+^O+' = 0.99';
  DefineWorld(-40,-40, 40,40 ,0,0);
  Axis(2,2, 8,8, 0,0, 'Fig.      S-Plane Root','Re','Im');
end;
'8':begin
  Str(ali[1]:1:0,alp[1]);
  Str(ali[2]:1:0,alp[2]);
  Str(ali[3]:1:0,alp[3]);
  Str(ali[4]:1:0,alp[4]);
  stl[1] := ^N+'a'+^O+' = '+alp[1];
  stl[2] := ^N+'a'+^O+' = '+alp[2];
  stl[3] := ^N+'a'+^O+' = '+alp[3];
  stl[4] := ^N+'a'+^O+' = '+alp[4];
  DefineWorld(-20,-10, 15,10 ,0,0);
  Axis(7,4, 1,1, 1,5, ' ','X','jY');
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;
if num in ['7','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

```

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าการตีพิมพ์ในสื่ออิเล็กทรอนิกส์จะต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    ph0 := 0;
    ph1 := 0;
end;
'7':begin
    if odd(fi) then n := nn
        else n := point;
    delta := (ww-wl)/n
end;
'8':begin
    al := ali[(fi+1) shr 1];
    ww := wwi[(fi+1) shr 1];
    if odd(fi) then n := nn
        else n := 3 shl 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 ['2'..'6'] 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
        '2'..'6': Write(OutFile,'SP',fi,';');
        '7': if odd(fi) then Write(OutFile,'SP1;')
            else Write(OutFile,'SP2;');
        '8': Write(OutFile,'SP',(fi+1) shr 1,';');
    end;
for i := 0 to n do
    begin
        case num of
            '2'..'6':begin
                aa := X_min+delta*i;
                if logs then w := power(10,aa) else w := aa;
                end;
            '7': w := wl+delta*i;
            '8':begin
                if odd(fi) then w := -ww+delta*i
                    else w := Round(-ww+delta*i);
                end;
        end;
end;
case num of
    '2':begin
        TargetF(w);
        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)ต่อ* 180/pi
    else
        begin

```

เอกสารนี้เป็นเอกสารของสำนักงานคณะกรรมการการอุดมศึกษา
 ไม่ว่ากรรมใดๆ ทั้งสิ้น ของสำนักงานคณะกรรมการการอุดมศึกษา
 ไม่ว่ากรรมใดๆ ทั้งสิ้น ของสำนักงานคณะกรรมการการอุดมศึกษา

```

        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
    Sensitivity(w);
    b:= data.re;
end;
'7':begin
    if (not ScreenDisp) and (not odd(fi)) then
    begin
        w := ali[i];
        S_plane(w);
        aa := data.re;
        b := data.im;
        pset(aa,b); Write(OutFile,'CI10;PU;');
        if b>=0 then
            Write(OutFile,'PR',-AY*0.07:6:4,
                ', ',AX*0.12:6:4,';LO7;')
        else
            Write(OutFile,'PR',-AY*0.07:6:4,
                ', ',-AX*0.12:6:4,';LO9;');
        Write(OutFile,'LBK = ',alp[i],^C);
    end
    else
    begin
        S_plane(w);
        aa := data.re;
        b := data.im;
    end;
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 w<>0 then
            if i=n then
                Write(OutFile,';LO5;LB',^N,'x',^O,' = ',w:1:0,^C)
            else
                Write(OutFile,';LO5;LB',w:1:0,^C);
    end;
    end;
end;
if (b<-1e10) then
    begin
        err1 := true;
        goto exit1;
    end;
if (b>1e10) then
    begin
        err2 := true;

```

```

    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 ['2'..'6'] 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'..'6'] then
begin
    Write('function ',fi,' : YMin ');
    if err1 then write('<') else write('=');
    Writeln(dy_min,' X -->',dx_min);
    Write('YMax ':21);
    if err2 then write('>') else write('=');
    Writeln(dy_max,' X -->',dx_max);
    if num='2' then
        if typ in ['A'..'H'] then
begin
            Fre_3db(0, 1e3 , Root , Error );
            if Error=0 then Writeln('W3db = ',:23,Root:11:8);
end;
        end;
    end
else

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ไม่ว่าการแก้ไขใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if odd(fi) then
begin
if num='7' then
begin
S_plane(ww);
if data.re<0 then
begin
Write('al =':7,ww:12:5);
Writeln('S = ':29,data.re:12:5,'j':7,data.im:12:5);
end
else
begin
Pole(ww, ww-0.01 , Root , Error );
if Error=0 then
begin
S_plane(Root);
Write('al <':7,Root:12:5);
Writeln('S = ':29,data.re:12:5,'j':7,data.im:12:5);
end
else
Writeln('**** Error ****':31);
end;
end
else
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(ww, 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);
end;
Delay(1500);
if ScreenDisp and (fi<>fn) then SwapScreen;
end;
if not ScreenDisp then
begin
Write(OutFile,'SP0;IN;');
if PScale>5 then
Write(OutFile,'PA16640,11400;')
else
Write(OutFile,'PA0,0;');
Close(OutFile);
end;

```

```

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('Active distributed RC');
  gotoxy(17,6); write('-----');
  gotoxy(24,9); write('1) Display');
  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) Sensitivity');
  gotoxy(24,15); write('7) S-Plane Root');
  gotoxy(24,16); write('8) Nyquist');
  gotoxy(24,17); write('9) other Menu');
  gotoxy(24,18); write('ESC Quit');
  gotoxy(27,20); write('Select No. ');
  repeat
    gotoxy(38,20); read(kbd,num); write(num);
  until num in [#27,'1'..'9'];
  if num='9' then
    begin
      ClrScr;
      other;
    end;
  if num='1' then
    begin
      GraphicsMode;
      SwapScreen;
      Wait(1);
      SwapScreen;
      TextMode;
    end;
  if num in ['2'..'8'] then
    begin
      gotoXY(1,25);
      write('Output [A]: Screen & Printer (^P=PrintScreen)');
      write(' [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;
          begin
            ..begin

```

เอกสารนี้เป็นลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

write('                                [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;
ClrScr;
gotoxy(24,9); write('A. Lowpass filter - 1');
gotoxy(24,10); write('B. Lowpass filter - 2');
gotoxy(24,11); write('C. Lowpass filter - 3');
gotoxy(24,12); write('D. Lowpass filter - 4');
gotoxy(24,13); write('E. Highpass filter - 1');
gotoxy(24,14); write('F. Highpass filter - 2');
gotoxy(24,15); write('G. Highpass filter - 3');
gotoxy(24,16); write('H. Highpass filter - 4');
gotoxy(24,17); write('I. Equalizer - 1');
gotoxy(24,18); write('J. Equalizer - 2');
gotoxy(24,19); write('K. Delay Equalizer');
gotoxy(27,22); write('Select No. ');
repeat
  gotoxy(38,22); Read(kbd,typ);
  typ := UpCase(typ); Write(typ);
until typ in ['A'..'K',^M];
if typ=^M then typ := '@'
else
  if num='6'then
  begin
    gotoXY(1,25);
    write('Sensitivity [A]: R1,C1,R2,C2');
    write(' [B]: K ');
    repeat
      read(kbd,ch);
      ch := UpCase(ch);
    until (ch='A') or (ch='B');
    if ch='B' then typ := chr(ord(typ) or $20);
  end;
  fn := 2;
  freq := false;
  xs := 3;
  Plot_Function;
end;
until num=#27;
end.

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ;
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

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  st1      : array[1..30] of st100;
      Sx,Sy   : array[0..50] of real;
      X_min,X_max,Y_min,Y_max,
      AxGlb,AyGlb,BxGlb,ByGlb
      AX,AY,XMax,YMax,x1,y1,
      fn,AXX,expX,expY
      dot     : real;
      logs,dB_log : integer;
      OutName : byte;
              : boolean;
              : 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);
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;
  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;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

function ReadyCom:boolean;
begin
regs.ah := \$03;

```

regs.dx := 0;
intr($l4,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;SP0;');
  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;Cil;');
      Write(OutFile, 'PA',AXX+XMax+AX, ', ',YMax+AY, ' ');
    for i := 1 to 3 do
      Write(OutFile, 'PD;PU;Cil;');
    Read(kbd,ch);
  end;

procedure BufferReady;
var i : integer;
begin
  if OutName='AUX' then
    begin
      repeat
        Write(OutFile,#27, '.O');
        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;
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
  xl := Round(AxGlb+BxGlb*x);
  yl := Round(AyGlb+ByGlb*y);
  if ScreenDisp then
  begin
    if (yl>=0) and (yl<YMax-3) then DrawPoint(xl,yl);
    dot := 0;
  end
  else
    Write(OutFile,'PU;PA',AXX+xl,',',yl,',';PD;');
  end;
end;

```

```

procedure Line(x2,y2: integer);
var DeltaX,DeltaY,XStep,YStep,direction : integer;
begin
  if ScreenDisp then
  begin
    if xl>x2 then XStep := -1 else XStep := 1;
    if yl>y2 then YStep := -1 else YStep := 1;
    DeltaX := abs(x2-xl);
    DeltaY := abs(y2-yl);
    if DeltaX=0 then direction := -1
      else direction := 0;
    while not ((xl=x2) and (yl=y2)) do
    begin
      if (yl>=0) and (yl<YMax-3) then
      begin
        case LineStyle of
          1 :DrawPoint(xl,yl);
          2 :case dot of
              1 : DrawPoint(xl,yl);
              4 : Dot := -1;
            end;
          3 :case dot of
              1..13 : DrawPoint(xl,yl);
              16 : Dot := -1;
            end;
          4 :case dot of
              0..35,39,43..46 : DrawPoint(xl,yl);
              47 :begin DrawPoint(xl,yl); Dot := -1; end;
            end;
          5 :case dot of
              0..31,35..43 : DrawPoint(xl,yl);

```

เอกสารนี้เป็นเอกสารที่...
 ไม่ว่าการณ์ใดๆ...
 ...

```

    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
else
  dot := 0;
  if direction<0 then
    begin
      yl := yl+YStep;
      direction := direction+DeltaX;
    end
  else
    begin
      xl := xl+XStep;
      direction := direction-DeltaY;
    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;
      end;
    end;
  end;

```

```

for i := 3 to 5 do
begin
  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;
end;

```

```

overlay 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;
end;
end;

```

```

begin

```

```

xw1 := 30000;
xw2 := -30000;
yw1 := 30000;
yw2 := -30000;

```

เอกสารนี้เป็นลิขสิทธิ์ของมหาวิทยาลัยราชภัฏวชิรบำรุง
 ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาต
 ใช้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 SetLineStyle(8);
 Xx := X_max-X_min; ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
 Yy := Y_max-Y_min;
 if ScreenDisp then


```

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;

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;
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,';SL0.2;SM;');
      Write(OutFile,'PU;PR',(AX shr 3),',',0,'LO2;LB',st1[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;');

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

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;
          end;
        sk := false;
        pl := 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(pl,Y_min+p2*i)
          else
            if pos<=4 then
              begin
                if wxy(pl,Y_min+p2*i) then pset(pl,Y_min+p2*i);
              end
            else
              begin
                if wxy(pl,Y_min+p2*i) then
                  begin
                    if sk then
                      Write(OutFile,'PD')
                    else
                      begin
                        Write(OutFile,'PU;PA');
                        sk := true;
                      end;
                end
              end
            else
              begin
                Write(OutFile,'PU;PA');
                sk := false;
              end;
            Write(OutFile,AXX+Round(AxGlb+BxGlb*pl),',');
            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;
          pl := X_min+Xx*j/(ScaleX*ScX);

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่ควรนำออกโดยไม่ได้รับอนุญาต

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่ควรนำออกโดยไม่ได้รับอนุญาต

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่ควรนำออกโดยไม่ได้รับอนุญาต


```

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

```

```

SwapScreen;
gotoxy(1,1);
Write('ScaleX ');
if expX=0 then Write(':')
    else Write('( *le',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('( *le',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;

```

overlay procedure Q;

```

var r_rl,al,dw,w,fp,zm,z7,w1,w2,p1,p2 : real;
    ws,we : integer;

```

function Z(x:real):real;

```

var t : real;

```

```

begin

```

```

    t := sqrt(x);

```

```

    Z := r_rl*sqrt(
        sqr( cosh(t)*cos(t)-1 ) + sqr( sinh(t)*sin(t) ) )/sqrt(
        sqr( al+t*( sinh(t)*cos(t)-cosh(t)*sin(t) ) )
        +sqr( t*( sinh(t)*cos(t)+cosh(t)*sin(t) ) ) );

```

```

end;

```

```

begin

```

```

write('R/RL = '); readln(r_rl); ศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```

```

write('alpha = '); readln(al);

```

```

write('w(start) = '); readln(ws);

```

```

write('w(end) = '); readln(we);

```

```

writeln;
dw := (we-ws)/10;
w := (we-ws)/2;
fp := Z(w/2);
repeat
  w := w+dw;
  zm := Z(w/2);
  if zm<=fp then dw := -dw/2;
  fp := zm;
until abs(dw)<1e-11;
w := w+dw;
zm := Z(w/2);
writeln('Zmax = ',zm:8:8);
writeln('w0 = ',w:8:8);
z7 := zm/sqrt(2);
w1 := w;
dw := (we-ws)/1000;
repeat
  p1 := Z(w1/2);
  if p1>z7 then
    w1 := w1-dw
  else
    if p1<z7 then
      begin
        w1 := w1+dw;
        dw := dw/2;
      end;
until abs(z7-p1)<1e-10;
writeln('w1 = ',w1:8:8);
w2 := w;
dw := (we-ws)/1000;
repeat
  p2 := Z(w2/2);
  if p2>z7 then
    w2 := w2+dw
  else
    if p2<z7 then
      begin
        w2 := w2-dw;
        dw := dw/2;
      end;
until abs(z7-p2)<1e-10;
writeln('w2 = ',w2:8:8);
writeln('Q = ',w/(w2-w1):8:8);
read;
end;

```

```

overlay procedure other;
const ArraySize = 60; { Maximum number of equations }
type complex = record
  Re,Im : real;
end;
CompVector = array[0..ArraySize] of complex;
IntVector = array[0..ArraySize] of integer;
RealVector = array[1..ArraySize] of real;
matrix = array[1..ArraySize] of RealVector;
var DFile : text;

```

```

procedure Sort(var array: CompVector; n: integer);
procedure swapping(var first, second: complex);
var tem : complex;

```

```

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;
end;

procedure Laguerre(var Degree : integer;
  var Poly : CompVector;
  InitGuess : complex;
  Tol : real;
  MaxIter : integer;
  var NumRoots : integer;
  var Roots : CompVector;
  var yRoots : CompVector;
  var Iter : IntVector;
  var Error : byte);

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

var AddIter : integer;
  InitDegree : integer;
  InitPoly : CompVector;
  GuessRoot : complex;

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

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

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

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

```

```

C3.Im := C1.Im - C2.Im;
end;

procedure Mult(var C1,C2,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 Divide(var C1,C2,C3 : complex);
var Dum1,Dum2 : complex;
    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 : complex);
const NearlyZero = 1E-10;
var R,Theta : real;
begin
  R := Sqrt(Sqr(C1.Re) + Sqr(C1.Im));
  if ABS(C1.Re) < NearlyZero 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       : CompVector;
                     Tol            : real;
                     MaxIter       : integer;
                     InitGuess     : complex;
                     var NumRoots   : integer;
                     var Roots     : CompVector;
                     var yRoots    : CompVector;
                     var Iter       : IntVector;
                     var GuessRoot : complex;
                     var InitDegree : integer;
                     var InitPoly  : CompVector;
                     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;
      InitPoly := Poly;
      while (Degree > 0) and (Modulus(Poly[Degree]) < NearlyZero) 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;
end;

```

```

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

```

```

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

```

```

procedure EvaluatePoly(Degree : integer;
  Poly : CompVector;
  X : complex;
  var yValue : complex;
  var yPrime : complex;
  var yDoublePrime : complex);

```

```

var Loop : integer;
  Dummy, yDPdummy : complex;
  Deriv, Deriv2 : CompVector;
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
  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      : complex;
                             yPrime      : complex;
                             yDoublePrime : complex;
                             var Dif      : complex);
var yPrimeSQR,yTimesyDPrime,Sum,
    SRoot,Numer1,Numer2,Numer,Denom : complex;
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) < NearlyZero 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)<=NearlyZero) or (ABS(Dif)<ABS(X*Tol))
end;

begin
  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   : CompVector;
                     Root      : complex);

var Term : integer;
    NewPoly : CompVector;
    Dummy   : complex;

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

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

procedure UserInput(var Degree : integer;
                    var Poly : CompVector;
                    var Guess : complex;
                    var Tol : real;
                    var MaxIter : integer);

var Ch : char;

procedure GetCoefficientsFromKeyboard(var Degree : integer;
                                      var Poly : CompVector);
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
    GetCoefficientsFromKeyboard(Degree, Poly);
    Guess.Re := 1;
    Guess.Im := 1;
    Tol := 1E-10;
    MaxIter := 1000;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure Results(NumRoots    : integer;
                  Answer      : CompVector;
                  MaxIter     : integer;
                  Error       : byte);
var Term : integer;
begin
  Writeln;
  Sound(3000);
  Delay(2);
  Nosound;
  case Error of
  1: Writeln('This will take more then ',MaxIter,' iterations.');
```

```

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

```

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

```

  4: Writeln
      ('The maximum number of iterations must be greater then 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      : CompVector;
                 POLy2     : CompVector;
                 InitDegree1 : integer;
                 InitDegree2 : integer;
                 var Poly   : compVector;
                 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;
    end;
```

```

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

```

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;
      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 : CompVector;
    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;
  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;
    end;
end;

```

```

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;
end;

```

```

procedure Linear_Slope_Delay(var Degree : integer;
var Poly : CompVector;
var Guess : complex;
var Tol : real;
var MaxIter : integer);

```

```

var W,M,N,a,b,da,db,ki : CompVector;
i,j,Md,Nd : integer;
T0,Wd0,k : real;

```

```

begin
Initial(W);
Initial(M);
Initial(N);
Initial(a);
Initial(Poly);
Degree := 8;
T0 := 0.3;
Wd0 := 0.1;
for i := 1 to (Degree-1) do
W[i].Re := Wd0*(sqrt(1+((i*pi)/(T0*Wd0)))-1);
ki[1].Re := sqrt(power(1.3240690659E-01,1/4));
ki[2].Re := sqrt(power(1/1.4988687223E-01,1/3));
ki[3].Re := sqrt(power(1.7538461547E-01,1/4));
ki[4].Re := sqrt(power(1/1.7487009849E-01,1/3));
ki[5].Re := sqrt(power(1.9519799087E-01,1/4));
ki[6].Re := sqrt(power(1/2.3941835154E-01,1/3));
ki[7].Re := sqrt(power(3.6460358025E-01,1/4));
for i := 1 to (Degree-1) do
writeln('W[' ,i,'] = ',W[i].Re);
writeln;
M[2].Re := 1 * ki[1].Re;
M[0].Re := sqr(W[1].Re) * ki[1].Re;
Md := 2;
N[1].Re := 1;
Nd := 1;
for i := 2 to (Degree-1) do
begin
Initial(b);
a[2].Re := 1 * ki[i].Re;
a[0].Re := sqr(W[i].Re) * ki[i].Re;
if odd(i) then
begin
polmul(M,a,Md,2,b,Md);
for j := 0 to Md do
M[j].Re := b[j].Re;
end
else
begin

```

เอกสารนี้ M[j].Re := b[j].Re; งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่าใคร ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin

```

```

    polmul(N,a,Nd,2,b,Nd);
    for j := 0 to Nd do
        N[j].Re := b[j].Re;
    end;
end;
Initial(a);
Initial(b);
Initial(da);
Initial(db);
k := 1;
writeln(DFile,'k = ',k:15:8); writeln(DFile);
for i := 0 to Degree do
    M[i].Re := k * M[i].Re;
for i := 0 to (Degree div 2) do
    begin
        a[i*2].Re := power(-1,i)*M[i*2].Re;
        b[i*2+1].Re := power(-1,i)*N[i*2+1].Re;
    end;
for i := 0 to Degree do
    begin
        da[i].Re := (i+1) * a[i+1].Re;
        db[i].Re := (i+1) * b[i+1].Re;
    end;
for i := 0 to Degree do
    begin
        Poly[i].Re := M[i].Re + N[i].Re;
        writeln(' Poly[' ,i:2,'] = ',Poly[i].Re);
        writeln(DFile,'Poly[' ,i:2,'] := ',Poly[i].Re:15:8,');
    end;
writeln(DFile); write(DFile,'(');
for i := 0 to Degree do
    if a[i].Re <> 0 then
        if i <> 0 then
            begin
                if a[i].Re > 0 then
                    write(DFile,'+',a[i].Re:16,'*power(w,',i,')')
                else
                    write(DFile,' ',a[i].Re:17,'*power(w,',i,')');
            end
        else
            if a[i].Re > 0 then write(DFile,'+',a[i].Re:16)
            else write(DFile,' ',a[i].Re:17);
        writeln(DFile,')'); write(DFile,'*(');
for i := 0 to Degree do
    if db[i].Re <> 0 then
        if i <> 0 then
            begin
                if db[i].Re > 0 then
                    write(DFile,'+',db[i].Re:16,'*power(w,',i,')')
                else
                    write(DFile,' ',db[i].Re:17,'*power(w,',i,')');
            end
        else
            if b[i].Re > 0 then write(DFile,'+',b[i].Re:16)
            else write(DFile,' ',b[i].Re:17);
        writeln(DFile,')');');
Guess.Re := 1;
Guess.Im := 1;
Tol := 1E-10;
MaxIter := 1000;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure Phase_equalizer(var Degree : integer;
                          var Poly   : CompVector;
                          var Guess  : complex;
                          var Tol    : real;
                          var MaxIter : integer);
var z1,z2,z3,p1,p2,p3,g1,g3,g5,k,q,w0_1,w0_2 : real;
begin
  z1 := 0.42738;
  p1 := 0.54367;
  z2 := 49.28777;
  z3 := 176.98871;
  p2 := 48.12094;
  p3 := 174.77134;
  g1 := ( 1/z1 + 1/z2 + 1/z3 ) - ( 1/p1 + 1/p2 + 1/p3);
  g3 := -( 1/(3*power(z1,3))+1/(3*power(z2,3))+1/(3*power(z3,3)) )
        +( 1/(3*power(p1,3))+1/(3*power(p2,3))+1/(3*power(p3,3)) );
  g5 := +( 1/(5*power(z1,5))+1/(5*power(z2,5))+1/(5*power(z3,5)) )
        -( 1/(5*power(p1,5))+1/(5*power(p2,5))+1/(5*power(p3,5)) );
  K := power(g5,3) / power(g3,5);
  Q := 0.62715786;
  W0_1 := sqrt( (3*g3 * (5*power(Q,4) - 5*power(Q,2) + 1) ) /
                (5*g5*power(Q,2)*(3*power(Q,2)-1)) );
  W0_2 := sqrt( g3/g5 * (1 - 1/power(Q,2) + 1/(5*power(Q,4)) ) /
                (1 - 1/(3*power(Q,2)) ) );
  Writeln(' Z1 = ',z1:15:8,' Z2 = ',z2:15:8,' Z3 = ',z3:15:8);
  Writeln(' P1 = ',p1:15:8,' P2 = ',p2:15:8,' P3 = ',p3:15:8);
  Writeln('g1 = ',g1:15:8);
  Writeln('g3 = ',g3:15:8);
  Writeln('g5 = ',g5:15:8);
  Writeln('K = ',k:15:8);
  Writeln('Qap = ',q:15:8);
  Writeln('W0ap = ',w0_1:15:8);
  Writeln('W0ap = ',w0_2:15:8);
  Initial(Poly);
  Poly[12].Re := 1;
  Poly[10].Re := -(3 + 4*K);
  Poly[8].Re := (18/5 + 20/3*K);
  Poly[6].Re := -(11/5 + 40/9*K);
  Poly[4].Re := (18/25 + 40/27*K);
  Poly[2].Re := -(3/25 + 20/81*K);
  Poly[0].Re := (1/125 + 4/243*K);
  Degree := 12;
  Guess.Re := 1;
  Guess.Im := 1;
  Tol := 1E-10;
  MaxIter := 1000;
end;

```

```

procedure Delay_equalizer;
var Coefficients : matrix;
    Constants,Solution,a,b,w : RealVector;
    data,d1,d2 : complex;
    e,D0,D,Tol,min,max,mm : Real;
    Dimen,Row,Column,i,iter : integer;
    Error : byte;
    IterM : boolean;
label loop,exitd;

```

```

function EquF(w:real):real;
var D : real;
    i : integer;
begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 { Function of Delay Equalization }

ไม่ว่ากรณีใดๆ ทั้งสิ้น ยกเว้นผู้ไม่มีสิทธิ์เผยแพร่และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

D := 0;
for i := 1 to (Dimen shr 1) do
  D := D + ( ( 2*a[i] ) / ( sqr(a[i])+sqr(w-b[i]) ) +
             ( 2*a[i] ) / ( sqr(a[i])+sqr(w+b[i]) ) );
EquF := D/(2*pi);
end;

function DifF(x: real):real;
const Tolerance = 1e-10;
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
  LeftPoint := EquF(X - DeltaX);
  RightPoint := EquF(X + DeltaX);
  EvaluateFirstDeriv := (RightPoint - LeftPoint)/(2 * DeltaX);
end;

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

procedure MinMax(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;
  yLeft := Diff(LeftEndpoint);
  yRight := Diff(RightEndpoint);
  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
    begin
      if yLeft * yRight > 0 then
        Error := 2;
      if Tol <= 0 then
        Error := 3;
      if MaxIter < 0 then
        Error := 4;
    end;
  if (Error = 0) and (Found = false) then
    begin
      Iter := 0;
      yLeft := Diff(LeftEndpoint);
      while not(Found) and (Iter < MaxIter) do
        begin
          Iter := Succ(Iter);
          MidPoint := (LeftEndpoint + RightEndpoint) / 2;
          yMidPoint := Diff(MidPoint);
          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 Partial_Pivoting(Dimen          : integer;
                           Coefficients   : matrix;
                           Constants      : RealVector;
                           var Solution   : RealVector;
                           var Error      : byte);

const NearlyZero = 1E-10;
var Sum, Multiplier, Dummy          : real;
    Term, Row, ReferenceRow, PivotRow : integer;
    DummyRow                         : RealVector;

procedure EROmultAdd(Multiplier : real;
                    Term, Row, ReferenceRow, PivotRow : integer;
                    DummyRow : RealVector;
                    Sum, Multiplier, Dummy : real;
                    Error : byte);

```

```

        Dimen      : integer;
        var ReferenceRow : RealVector;
        var ChangingRow  : RealVector);

var Term : integer;
begin
  for Term := 1 to Dimen do
    ChangingRow[Term] := ChangingRow[Term] +
      Multiplier*ReferenceRow[Term];
  end;

begin
  Error := 0;
  if Dimen < 1 then
    Error := 1
  else
    if Dimen = 1 then
      if ABS(Coefficients[1, 1]) < NearlyZero then
        Error := 2
      else
        Solution[1] := Constants[1] / Coefficients[1, 1];
      if Dimen > 1 then
        begin
          { Make Coefficients matrix upper triangular }
          ReferenceRow := 0;
          while (Error = 0) and (ReferenceRow < Dimen - 1) do
            begin
              ReferenceRow := Succ(ReferenceRow);
              { Find row with largest element in this column }
              { and switch this row with the ReferenceRow }
              PivotRow := ReferenceRow;
              for Row := ReferenceRow + 1 to Dimen do
                if ABS(Coefficients[Row, ReferenceRow]) >
                  ABS(Coefficients[PivotRow, ReferenceRow]) then
                  PivotRow := Row;
              if PivotRow <> ReferenceRow then
                begin
                  DummyRow := Coefficients[PivotRow];
                  Coefficients[PivotRow] := Coefficients[ReferenceRow];
                  Coefficients[ReferenceRow] := DummyRow;
                  Dummy := Constants[PivotRow];
                  Constants[PivotRow] := Constants[ReferenceRow];
                  Constants[ReferenceRow] := Dummy;
                end
              else { If the diagonal element is zero, no solution exists }
                if ABS(Coefficients[ReferenceRow, ReferenceRow]) <
                  NearlyZero then Error := 2;
              if Error = 0 then
                for Row := ReferenceRow + 1 to Dimen do
                  { Make the ReferenceRow element of these rows zero }
                  if ABS(Coefficients[Row, ReferenceRow]) > NearlyZero then
                    begin
                      Multiplier := -Coefficients[Row, ReferenceRow] /
                        Coefficients[ReferenceRow, ReferenceRow];
                      EROMultAdd(Multiplier, Dimen, Coefficients[ReferenceRow],
                        Coefficients[Row]);
                      Constants[Row] := Constants[Row] +
                        Multiplier * Constants[ReferenceRow];
                    end;
                end;
              end;
            if ABS(Coefficients[Dimen, Dimen]) < NearlyZero then
              Error := 2;
            if Error = 0 then
              begin
                Term := Dimen;

```

```

while Term >= 1 do
begin
    Sum := 0;
    for Row := Term + 1 to Dimen do
        Sum := Sum + Coefficients[Term, Row] * Solution[Row];
    Solution[Term] := (Constants[Term] - Sum) /
        Coefficients[Term, Term];
    Term := Pred(Term);
end;
end;
end;
end;

begin
    Tol := 1E-10;
    iter := -1;
    IterM := false;
    FillChar(Coefficients, SizeOf(Coefficients), 0);
    FillChar(Constants, SizeOf(Constants), 0);
    Dimen := 10;
    e := 0.005;
    D0 := 0.55;
    a[1] := 1.3575; b[1] := 0.66667;
    a[2] := 1.3575; b[2] := 2;
    a[3] := 1.3575; b[3] := 3.3333;
    a[4] := 1.3575; b[4] := 4.6667;
    a[5] := 1.086; b[5] := 6;
loop:
    iter := iter+1;
    gotoxy(25,1); writeln('Iterate ',iter); writeln;
    for i := 1 to (Dimen shr 1) do
        Writeln('a',i,' = ',a[i]:13:10,'b':15,i,' = ',b[i]:13:10);
    writeln;
    min := 0;
    max := 0;
    for Row := 1 to Dimen do
    begin
        if odd(Row) then
        begin
            if Row=1 then
            begin
                if not IterM then
                    w[Row] := 0
                else
                begin
                    w[Row] := b[1]/3;
                    if Diff(w[Row])>=0 then
                        MinMax(0,w[Row],mm,Error)
                    else
                        MinMax(w[Row],b[1],mm,Error);
                    if Error=0 then w[Row] := mm;
                end;
            end
        else
            begin
                w[Row] := (b[(Row shr 1)]+b[(Row shr 1)+1])/2;
                if IterM then
                begin
                    if Diff(w[Row])>=0 then
                        MinMax(b[(Row shr 1)],w[Row],mm,Error)
                    else
                        MinMax(w[Row],b[(Row shr 1)+1],mm,Error);
                    if Error=0 then w[Row] := mm;
                end;
            end;
        end;
    end;
end;

```

```

        end;
    end;
end
else
begin
    w[Row] := b[Row shr 1];
    if IterM then
        begin
            if Diff(w[Row])<=0 then
                MinMax((b[(Row shr 1)-1]+b[(Row shr 1)])/2,w[Row],mm,Error)
            else
                MinMax(w[Row],(b[(Row shr 1)]+b[(Row shr 1)+1])/2,mm,Error);
            if Error=0 then w[Row] := mm;
        end;
    end;
D := EquF(w[Row]);
if odd(Row) then
begin
    if (Row=1) and not IterM then
        begin
            Constants[Row] := D0-D;
            if min<abs(D-D0) then min := abs(D-D0);
        end
    else
        begin
            Constants[Row] := D0-D-e/2;
            if min<abs(D-D0+e/2) then min := abs(D-D0+e/2);
        end;
    end
else
begin
    Constants[Row] := D0-D+e/2;
    if max<abs(D-D0-e/2) then max := abs(D-D0-e/2);
end;
Writeln('w',Row:2,' = ',w[Row],'delta D':15,
        ' = ',Constants[Row]);
end;
if (min<Tol) and (max<Tol) then
begin
    if IterM then
        goto exitd
    else
        begin
            gotoxy(1,1); write('First[' ,iter,']');
            IterM := true;
            read;
        end;
end;
for Row := 1 to Dimen do
begin
    for Column := 1 to Dimen do
        begin
            if Column<=(Dimen shr 1) then
                begin
                    i := Column;
                    Coefficients[Row, Column] :=
                        2*( ( sqr(w[Row]-b[i]) - sqr(a[i]) )
                            / sqr( sqr(a[i]) + sqr(w[Row]-b[i]) )
                            + ( sqr(w[Row]+b[i]) - sqr(a[i]) )
                            / sqr( sqr(a[i]) + sqr(w[Row]+b[i]) ) );
                end
            else
                begin
                    i := Dimen - Column + 1;
                    Coefficients[Row, Column] :=
                        2*( ( sqr(w[Row]-b[i]) - sqr(a[i]) )
                            / sqr( sqr(a[i]) + sqr(w[Row]-b[i]) )
                            + ( sqr(w[Row]+b[i]) - sqr(a[i]) )
                            / sqr( sqr(a[i]) + sqr(w[Row]+b[i]) ) );
                end
            end;
        end;
    end;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือมีการคุ้มครองทางกฎหมาย มิฉะนั้นให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดก็ตาม ห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        i := Column - (Dimen shr 1);
        Coefficients[Row, Column] :=
            4*a[i]*( ( w[Row]-b[i] ) /
                sqr( sqr(a[i]) + sqr(w[Row]-b[i]) ) -
                ( w[Row]+b[i] ) / sqr( sqr(a[i]) + sqr(w[Row]+b[i]) ) );
    end;
end;
end;
Partial_Pivoting(Dimen, Coefficients, Constants, Solution, Error);
if Error=0 then
begin
    for i := 1 to (Dimen shr 1) do
    begin
        a[i] := a[i]+Solution[i];
        b[i] := b[i]+Solution[(Dimen shr 1)+i];
    end;
end
else
begin
    Writeln;
    Write('***** Error *****'); read;
end;
goto loop;
exitd:
end;

var Guess : complex;
    InitDegree, Degree, MaxIter, NumRoots : integer;
    Tol : real;
    Iter : IntVector;
    InitPoly, Poly, Answer, yAnswer : CompVector;
    Error : byte;
    ch : char;

begin
    InitDegree := 0;
    Initial(InitPoly);
    repeat
        ClrScr;
        gotoxy(24,9); write('1) Multiple Polynomial');
        gotoxy(24,10); write('2) Roots & Polynomial');
        gotoxy(24,11); write('3) Linear-Slope Delay ');
        gotoxy(24,12); write('4) Phase-equalizer');
        gotoxy(24,13); write('5) Delay-equalizer');
        gotoxy(23,14); write('ESC Quit');
        gotoxy(27,17); write('Select No. ');
        repeat
            gotoxy(38,17); read(kbd,ch); write(ch);
        until ch in [#27, '1'..'5'];
        if ch<>#27 then
            begin
                ClrScr;
                Assign(DFile, 'SOLUTION');
                Rewrite(DFile);
                case ch of
                    '1': MulPoly;
                    '2': UserInput(InitDegree, InitPoly, Guess, Tol, MaxIter);
                    '3': Linear_Slope_Delay(InitDegree, InitPoly, Guess, Tol, MaxIter);
                    '4': Phase_equalizer(InitDegree, InitPoly, Guess, Tol, MaxIter);
                    '5': Delay_equalizer;
                end;
            end;
        if ch in ['2'..'4'] then
            begin
                Degree := InitDegree;

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่าการแก้ไขหรือการเปลี่ยนแปลงใดๆก็ตามจะต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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;
until ch=#27;
end;
{$I SENS.INC}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

overlay procedure Sensitivity(w : real);
var dl,d2,d3      : complex;
    t,r1,r2,c1,c2,k : real;
begin
  t := sqrt(w*RC/2);
  case typ of
'A':
  case fi of
1:begin
  mul_c( t , t , cosh(t)*cos(t) , sinh(t)*sin(t) , dl );
  div_c( dl.re , dl.im , sinh(t)*cos(t) , cosh(t)*sin(t) , dl );
  add_c( dl.re , dl.im , 1 , 0 , dl );
  mul_c( dl.re , dl.im , 1/a1 , 0 , dl );
  div_c( dl.re , dl.im , 1 + 1/a1 , 0 , d3 );
  mul_c( t , t , cosh(t)*cos(t) , sinh(t)*sin(t) , dl );
  div_c( dl.re , dl.im , sinh(t)*cos(t) , cosh(t)*sin(t) , dl );
  add_c( dl.re , dl.im , 1 , 0 , dl );
  mul_c( dl.re , dl.im , cosh(t)*cos(t) , sinh(t)*sin(t) , dl );
  mul_c( dl.re , dl.im , 1/a1 , 0 , dl );
  mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , 1/a1 , 0 ,d2 );
  add_c( d2.re , d2.im , 1 , 0 , d2 );
  div_c( dl.re , dl.im , d2.re , d2.im , dl );
  sub_c( d3.re , d3.im , dl.re , dl.im , dl );
  mul_c( dl.re , dl.im , 1/2 , 0 , data );
end;
2:begin
  mul_c( t , t , cosh(t)*cos(t) , sinh(t)*sin(t) , dl );
  div_c( dl.re , dl.im , sinh(t)*cos(t) , cosh(t)*sin(t) , dl );
  sub_c( dl.re , dl.im , 1 , 0 , dl );
  mul_c( dl.re , dl.im , 1/a1 , 0 , dl );
  div_c( dl.re , dl.im , 1 + 1/a1 , 0 , d3 );
  mul_c( t , t , cosh(t)*cos(t) , sinh(t)*sin(t) , dl );
  div_c( dl.re , dl.im , sinh(t)*cos(t) , cosh(t)*sin(t) , dl );
  sub_c( dl.re , dl.im , 1 , 0 , dl );
  mul_c( dl.re , dl.im , cosh(t)*cos(t) , sinh(t)*sin(t) , dl );
  mul_c( dl.re , dl.im , 1/a1 , 0 , dl );
  mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , 1/a1 , 0 ,d2 );
  add_c( d2.re , d2.im , 1 , 0 , d2 );
  div_c( dl.re , dl.im , d2.re , d2.im , dl );
  sub_c( d3.re , d3.im , dl.re , dl.im , dl );
  mul_c( dl.re , dl.im , 1/2 , 0 , data );
end;
3:begin
  mul_c( t , t , cosh(t)*cos(t) , sinh(t)*sin(t) , dl );
  div_c( dl.re , dl.im , sinh(t)*cos(t) , cosh(t)*sin(t) , dl );
  sub_c( dl.re , dl.im , 1/a1 , 0 , dl );
  div_c( dl.re , dl.im , 1 + 1/a1 , 0 , d3 );
  mul_c( t , t , sinh(t)*cos(t) , cosh(t)*sin(t) , dl );
  sub_c( dl.re , dl.im , cosh(t)*cos(t) , sinh(t)*sin(t) , dl );
  mul_c( dl.re , dl.im , 1/a1 , 0 , dl );
  mul_c( t , t , cosh(t)*cos(t) , sinh(t)*sin(t) , d2 );
  div_c( d2.re , d2.im , sinh(t)*cos(t) , cosh(t)*sin(t) , d2 );
  add_c( dl.re , dl.im , d2.re , d2.im , dl );
  mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , 1/a1 , 0 ,d2 );
  add_c( d2.re , d2.im , 1 , 0 , d2 );
  div_c( dl.re , dl.im , d2.re , d2.im , dl );
  sub_c( d3.re , d3.im , dl.re , dl.im , dl );
  mul_c( dl.re , dl.im , 1/2 , 0 , data );
end;
4:begin
  mul_c( t , t , cosh(t)*cos(t) , sinh(t)*sin(t) , dl );
  div_c( dl.re , dl.im , sinh(t)*cos(t) , cosh(t)*sin(t) , dl );
  add_c( dl.re , dl.im , 1/a1 , 0 , dl );

```

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่มีการอนุญาตให้คัดลอกหรือทำซ้ำโดยไม่ได้รับอนุญาต

```

div_c( dl.re , dl.im , 1 + 1/a1 , 0 , d3 );
mul_c( t , t , sinh(t)*cos(t) , cosh(t)*sin(t) , dl );
add_c( dl.re , dl.im , cosh(t)*cos(t) , sinh(t)*sin(t) , dl );
mul_c( dl.re , dl.im , 1/a1 , 0 , dl );
mul_c( t , t , cosh(t)*cos(t) , sinh(t)*sin(t) , d2 );
div_c( d2.re , d2.im , sinh(t)*cos(t) , cosh(t)*sin(t) , d2 );
add_c( dl.re , dl.im , d2.re , d2.im , dl );
mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , 1/a1 , 0 , d2 );
add_c( d2.re , d2.im , 1 , 0 , d2 );
div_c( dl.re , dl.im , d2.re , d2.im , dl );
sub_c( d3.re , d3.im , dl.re , dl.im , dl );
mul_c( dl.re , dl.im , 1/2 , 0 , data );
end;
end;
'B'..'D':
case fi of
1:begin
div_c( t*(cosh(t)*cos(t)-sinh(t)*sin(t)) ,
t*(cosh(t)*cos(t)+sinh(t)*sin(t)) ,
sinh(t)*cos(t) , cosh(t)*sin(t) , dl );
add_c( dl.re , dl.im , 1 , 0 , dl );
mul_c( dl.re , dl.im , cosh(t)*cos(t) -1 , sinh(t)*sin(t) , dl );
div_c( dl.re , dl.im , cosh(t)*cos(t) -1+a1 , sinh(t)*sin(t) , dl );
mul_c( dl.re , dl.im , -1/2 , 0 , data );
end;
2:begin
div_c( -t*(cosh(t)*cos(t)-sinh(t)*sin(t)) ,
-t*(cosh(t)*cos(t)+sinh(t)*sin(t)) ,
sinh(t)*cos(t) , cosh(t)*sin(t) , dl );
add_c( dl.re , dl.im , 1 , 0 , dl );
mul_c( dl.re , dl.im , cosh(t)*cos(t) -1 , sinh(t)*sin(t) , dl );
div_c( dl.re , dl.im , cosh(t)*cos(t) -1+a1 , sinh(t)*sin(t) , dl );
mul_c( dl.re , dl.im , 1/2 , 0 , data );
end;
3:begin
div_c( t*(cosh(t)*cos(t)-sinh(t)*sin(t)) ,
t*(cosh(t)*cos(t)+sinh(t)*sin(t)) ,
sinh(t)*cos(t) , cosh(t)*sin(t) , dl );
mul_c( dl.re , dl.im , a1 , 0 , d2 );
add_c( d2.re , d2.im , -cosh(t)*cos(t) +1 , -sinh(t)*sin(t) , d2 );
add_c( d2.re , d2.im , t*(sinh(t)*cos(t) - cosh(t)*sin(t)) ,
t*(sinh(t)*cos(t)+cosh(t)*sin(t)) , d2 );
div_c( d2.re , d2.im , cosh(t)*cos(t) -1+a1 , sinh(t)*sin(t) , d2 );
sub_c( dl.re , dl.im , d2.re , d2.im , dl );
mul_c( dl.re , dl.im , 1/2 , 0 , data );
end;
4:begin
div_c( t*(cosh(t)*cos(t)-sinh(t)*sin(t)) ,
t*(cosh(t)*cos(t)+sinh(t)*sin(t)) ,
sinh(t)*cos(t) , cosh(t)*sin(t) , dl );
mul_c( dl.re , dl.im , a1 , 0 , d2 );
add_c( d2.re , d2.im , cosh(t)*cos(t) -1 , sinh(t)*sin(t) , d2 );
add_c( d2.re , d2.im , t*(sinh(t)*cos(t) - cosh(t)*sin(t)) ,
t*(sinh(t)*cos(t)+cosh(t)*sin(t)) , d2 );
div_c( d2.re , d2.im , cosh(t)*cos(t) -1+a1 , sinh(t)*sin(t) , d2 );
sub_c( dl.re , dl.im , d2.re , d2.im , dl );
mul_c( dl.re , dl.im , 1/2 , 0 , data );
end;
end;
end;
'E'..'F':
case fi of
1:begin
mul_c( t , t , sinh(t)*cos(t) , cosh(t)*sin(t) , dl );

```

เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ผลิตขึ้นอีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

div_c( dl.re , dl.im , cosh(t)*cos(t) -1 , sinh(t)*sin(t) , d3 );
mul_c( t , t , cosh(t)*cos(t) , sinh(t)*sin(t) , dl );
div_c( dl.re , dl.im , sinh(t)*cos(t) , cosh(t)*sin(t) , dl );
add_c( dl.re , dl.im , 1 , 0 , dl );
mul_c( dl.re , dl.im , 1/al , 0 , dl );
mul_c( t , t , sinh(t)*cos(t) , cosh(t)*sin(t) , d2);
add_c( dl.re , dl.im , d2.re , d2.im , dl );
add_c( cosh(t)*cos(t) , sinh(t)*sin(t) , -1 + 1/al , 0 , d2 );
div_c( dl.re , dl.im , d2.re , d2.im , dl );
sub_c( d3.re , d3.im , dl.re , dl.im , dl );
mul_c( dl.re , dl.im , 1/2 , 0 , data );
end;
2:begin
mul_c( t , t , sinh(t)*cos(t) , cosh(t)*sin(t) , dl );
div_c( dl.re , dl.im , cosh(t)*cos(t) -1 , sinh(t)*sin(t) , d3 );
mul_c( t , t , cosh(t)*cos(t) , sinh(t)*sin(t) , dl );
div_c( dl.re , dl.im , sinh(t)*cos(t) , cosh(t)*sin(t) , dl );
sub_c( dl.re , dl.im , 1 , 0 , dl );
mul_c( dl.re , dl.im , 1/al , 0 , dl );
mul_c( t , t , sinh(t)*cos(t) , cosh(t)*sin(t) , d2);
add_c( dl.re , dl.im , d2.re , d2.im , dl );
add_c( cosh(t)*cos(t) , sinh(t)*sin(t) , -1 + 1/al , 0 , d2 );
div_c( dl.re , dl.im , d2.re , d2.im , dl );
sub_c( d3.re , d3.im , dl.re , dl.im , dl );
mul_c( dl.re , dl.im , 1/2 , 0 , data );
end;
3:begin
mul_c( t , t , cosh(t)*cos(t) , sinh(t)*sin(t) , dl );
div_c( dl.re , dl.im , sinh(t)*cos(t) , cosh(t)*sin(t) , d3 );
mul_c( t , t , cosh(t)*cos(t) , sinh(t)*sin(t) , dl );
div_c( dl.re , dl.im , sinh(t)*cos(t) , cosh(t)*sin(t) , dl );
mul_c( dl.re , dl.im , cosh(t)*cos(t) -1 , sinh(t)*sin(t) , dl );
sub_c( dl.re , dl.im , 1/al , 0 , dl );
add_c( cosh(t)*cos(t) , sinh(t)*sin(t) , -1 + 1/al , 0 , d2 );
div_c( dl.re , dl.im , d2.re , d2.im , dl );
sub_c( d3.re , d3.im , dl.re , dl.im , dl );
mul_c( dl.re , dl.im , 1/2 , 0 , data );
end;
4:begin
mul_c( t , t , cosh(t)*cos(t) , sinh(t)*sin(t) , dl );
div_c( dl.re , dl.im , sinh(t)*cos(t) , cosh(t)*sin(t) , d3 );
mul_c( t , t , cosh(t)*cos(t) , sinh(t)*sin(t) , dl );
div_c( dl.re , dl.im , sinh(t)*cos(t) , cosh(t)*sin(t) , dl );
mul_c( dl.re , dl.im , cosh(t)*cos(t) -1 , sinh(t)*sin(t) , dl );
add_c( dl.re , dl.im , 1/al , 0 , dl );
add_c( cosh(t)*cos(t) , sinh(t)*sin(t) , -1 + 1/al , 0 , d2 );
div_c( dl.re , dl.im , d2.re , d2.im , dl );
sub_c( d3.re , d3.im , dl.re , dl.im , dl );
mul_c( dl.re , dl.im , 1/2 , 0 , data );
end;
end;
end;
'G', 'H':
case fi of
1:begin
mul_c( t , t , cosh(t)*cos(t) , sinh(t)*sin(t) , dl );
div_c( dl.re , dl.im , sinh(t)*cos(t) , cosh(t)*sin(t) , dl );
add_c( dl.re , dl.im , 1 , 0 , dl );
mul_c( dl.re , dl.im , cosh(t)*cos(t) -1 , sinh(t)*sin(t) , dl );
mul_c( dl.re , dl.im , 1/al , 0 , dl );
mul_c( t , t , sinh(t)*cos(t) , cosh(t)*sin(t) , d2 );
add_c( dl.re , dl.im , d2.re , d2.im , dl );
mul_c( 1 + 1/al , 0 , cosh(t)*cos(t) , sinh(t)*sin(t) , d2 );
add_c( d2.re , d2.im , -1 -1/al , 0 , d2 );
div_c( dl.re , dl.im , d2.re , d2.im , d3 );

```

```

mul_c( t , t , cosh(t)*cos(t) , sinh(t)*sin(t) , dl );
div_c( dl.re , dl.im , sinh(t)*cos(t) , cosh(t)*sin(t) , dl );
add_c( dl.re , dl.im , 1 , 0 , dl );
mul_c( dl.re , dl.im , cosh(t)*cos(t) , sinh(t)*sin(t) , dl );
mul_c( dl.re , dl.im , 1/a1 , 0 , dl );
mul_c( t , t , sinh(t)*cos(t) , cosh(t)*sin(t) , d2 );
add_c( dl.re , dl.im , d2.re , d2.im , dl );
mul_c( 1 + 1/a1 , 0 , cosh(t)*cos(t) , sinh(t)*sin(t) , d2 );
add_c( d2.re , d2.im , -1 , 0 , d2 );
div_c( dl.re , dl.im , d2.re , d2.im , dl );
sub_c( d3.re , d3.im , dl.re , dl.im , dl );
mul_c( dl.re , dl.im , 1/2 , 0 , data );
end;

```

2:begin

```

mul_c( t , t , cosh(t)*cos(t) , sinh(t)*sin(t) , dl );
div_c( dl.re , dl.im , sinh(t)*cos(t) , cosh(t)*sin(t) , dl );
sub_c( dl.re , dl.im , 1 , 0 , dl );
mul_c( dl.re , dl.im , cosh(t)*cos(t) - 1 , sinh(t)*sin(t) , dl );
mul_c( dl.re , dl.im , 1/a1 , 0 , dl );
mul_c( t , t , sinh(t)*cos(t) , cosh(t)*sin(t) , d2 );
add_c( dl.re , dl.im , d2.re , d2.im , dl );
mul_c( 1 + 1/a1 , 0 , cosh(t)*cos(t) , sinh(t)*sin(t) , d2 );
add_c( d2.re , d2.im , -1 - 1/a1 , 0 , d2 );
div_c( dl.re , dl.im , d2.re , d2.im , d3 );
mul_c( t , t , cosh(t)*cos(t) , sinh(t)*sin(t) , dl );
div_c( dl.re , dl.im , sinh(t)*cos(t) , cosh(t)*sin(t) , dl );
sub_c( dl.re , dl.im , 1 , 0 , dl );
mul_c( dl.re , dl.im , cosh(t)*cos(t) , sinh(t)*sin(t) , dl );
mul_c( dl.re , dl.im , 1/a1 , 0 , dl );
mul_c( t , t , sinh(t)*cos(t) , cosh(t)*sin(t) , d2 );
add_c( dl.re , dl.im , d2.re , d2.im , dl );
mul_c( 1 + 1/a1 , 0 , cosh(t)*cos(t) , sinh(t)*sin(t) , d2 );
add_c( d2.re , d2.im , -1 , 0 , d2 );
div_c( dl.re , dl.im , d2.re , d2.im , dl );
sub_c( d3.re , d3.im , dl.re , dl.im , dl );
mul_c( dl.re , dl.im , 1/2 , 0 , data );
end;

```

3:begin

```

mul_c( t , t , cosh(t)*cos(t) , sinh(t)*sin(t) , dl );
div_c( dl.re , dl.im , sinh(t)*cos(t) , cosh(t)*sin(t) , dl );
mul_c( dl.re , dl.im , cosh(t)*cos(t) - 1 , sinh(t)*sin(t) , dl );
mul_c( t , t , sinh(t)*cos(t) , cosh(t)*sin(t) , d2 );
add_c( d2.re , d2.im , -cosh(t)*cos(t) + 1 , -sinh(t)*sin(t) , d2 );
mul_c( d2.re , d2.im , 1/a1 , 0 , d2 );
add_c( dl.re , dl.im , d2.re , d2.im , dl );
mul_c( 1 + 1/a1 , 0 , cosh(t)*cos(t) , sinh(t)*sin(t) , d2 );
add_c( d2.re , d2.im , -1 - 1/a1 , 0 , d2 );
div_c( dl.re , dl.im , d2.re , d2.im , d3 );
mul_c( t , t , cosh(t)*cos(t) , sinh(t)*sin(t) , dl );
div_c( dl.re , dl.im , sinh(t)*cos(t) , cosh(t)*sin(t) , dl );
mul_c( dl.re , dl.im , cosh(t)*cos(t) - 1 , sinh(t)*sin(t) , dl );
mul_c( t , t , sinh(t)*cos(t) , cosh(t)*sin(t) , d2 );
add_c( d2.re , d2.im , -cosh(t)*cos(t) , -sinh(t)*sin(t) , d2 );
mul_c( d2.re , d2.im , 1/a1 , 0 , d2 );
add_c( dl.re , dl.im , d2.re , d2.im , dl );
mul_c( 1 + 1/a1 , 0 , cosh(t)*cos(t) , sinh(t)*sin(t) , d2 );
add_c( d2.re , d2.im , -1 , 0 , d2 );
div_c( dl.re , dl.im , d2.re , d2.im , dl );
sub_c( d3.re , d3.im , dl.re , dl.im , dl );
mul_c( dl.re , dl.im , 1/2 , 0 , data );
end;

```

4:begin

```

mul_c( t , t , cosh(t)*cos(t) , sinh(t)*sin(t) , dl );

```

```

div_c( dl.re , dl.im , sinh(t)*cos(t) , cosh(t)*sin(t) , dl );
mul_c( dl.re , dl.im , cosh(t)*cos(t) -1 , sinh(t)*sin(t) , dl );
mul_c( t , t , sinh(t)*cos(t) , cosh(t)*sin(t) , d2 );
add_c( d2.re , d2.im , cosh(t)*cos(t) -1 , sinh(t)*sin(t) , d2 );
mul_c( d2.re , d2.im , 1/al , 0 , d2 );
add_c( dl.re , dl.im , d2.re , d2.im , dl );
mul_c( 1 + 1/al , 0 , cosh(t)*cos(t) , sinh(t)*sin(t) , d2 );
add_c( d2.re , d2.im , -1 -1/al , 0 , d2 );
div_c( dl.re , dl.im , d2.re , d2.im , d3 );
mul_c( t , t , cosh(t)*cos(t) , sinh(t)*sin(t) , dl );
div_c( dl.re , dl.im , sinh(t)*cos(t) , cosh(t)*sin(t) , dl );
mul_c( dl.re , dl.im , cosh(t)*cos(t) -1 , sinh(t)*sin(t) , dl );
mul_c( t , t , sinh(t)*cos(t) , cosh(t)*sin(t) , d2 );
add_c( d2.re , d2.im , cosh(t)*cos(t) , sinh(t)*sin(t) , d2 );
mul_c( d2.re , d2.im , 1/al , 0 , d2 );
add_c( dl.re , dl.im , d2.re , d2.im , dl );
mul_c( 1 + 1/al , 0 , cosh(t)*cos(t) , sinh(t)*sin(t) , d2 );
add_c( d2.re , d2.im , -1 , 0 , d2 );
div_c( dl.re , dl.im , d2.re , d2.im , dl );
sub_c( d3.re , d3.im , dl.re , dl.im , dl );
mul_c( dl.re , dl.im , 1/2 , 0 , data );
end;
end;
'I','J':
case fi of
1:begin
mul_c( t , t , cosh(t)*cos(t) , sinh(t)*sin(t) , dl);
div_c( dl.re , dl.im , sinh(t)*cos(t) , cosh(t)*sin(t) , dl );
add_c( dl.re , dl.im , 1 , 0 , dl );
mul_c( dl.re , dl.im , 1/al , 0 , dl );
mul_c( t , t , sinh(t)*cos(t) , cosh(t)*sin(t) , d2);
add_c( dl.re , dl.im , d2.re , d2.im , dl );
add_c( -1 +1/al , 0 , cosh(t)*cos(t) , sinh(t)*sin(t) , d2 );
div_c( dl.re , dl.im , d2.re , d2.im , d3 );
mul_c( t , t , cosh(t)*cos(t) , sinh(t)*sin(t) , dl);
div_c( dl.re , dl.im , sinh(t)*cos(t) , cosh(t)*sin(t) , dl );
add_c( dl.re , dl.im , 1 , 0 , dl );
mul_c( dl.re , dl.im , cosh(t)*cos(t) , sinh(t)*sin(t) , dl );
mul_c( dl.re , dl.im , 1/al , 0 , dl );
mul_c( t , t , sinh(t)*cos(t) , cosh(t)*sin(t) , d2);
add_c( dl.re , dl.im , d2.re , d2.im , dl );
mul_c( 1/al , 0 , cosh(t)*cos(t) , sinh(t)*sin(t) , d2);
add_c( d2.re , d2.im , cosh(t)*cos(t) -1 , sinh(t)*sin(t) , d2 );
div_c( dl.re , dl.im , d2.re , d2.im , dl );
sub_c( d3.re , d3.im , dl.re , dl.im , dl );
mul_c( dl.re , dl.im , 1/2 , 0 , data );
end;
2:begin
mul_c( t , t , cosh(t)*cos(t) , sinh(t)*sin(t) , dl);
div_c( dl.re , dl.im , sinh(t)*cos(t) , cosh(t)*sin(t) , dl );
add_c( dl.re , dl.im , -1 , 0 , dl );
mul_c( dl.re , dl.im , 1/al , 0 , dl );
mul_c( t , t , sinh(t)*cos(t) , cosh(t)*sin(t) , d2);
add_c( dl.re , dl.im , d2.re , d2.im , dl );
add_c( -1 +1/al , 0 , cosh(t)*cos(t) , sinh(t)*sin(t) , d2 );
div_c( dl.re , dl.im , d2.re , d2.im , d3 );
mul_c( t , t , cosh(t)*cos(t) , sinh(t)*sin(t) , dl);
div_c( dl.re , dl.im , sinh(t)*cos(t) , cosh(t)*sin(t) , dl );
add_c( dl.re , dl.im , -1 , 0 , dl );
mul_c( dl.re , dl.im , cosh(t)*cos(t) , sinh(t)*sin(t) , dl );
mul_c( dl.re , dl.im , 1/al , 0 , dl );
mul_c( t , t , sinh(t)*cos(t) , cosh(t)*sin(t) , d2);
add_c( dl.re , dl.im , d2.re , d2.im , dl );

```

```

mul_c( 1/a1 , 0 , cosh(t)*cos(t) , sinh(t)*sin(t) , d2);
add_c( d2.re , d2.im , cosh(t)*cos(t) -1 , sinh(t)*sin(t) , d2 );
div_c( d1.re , d1.im , d2.re , d2.im , d1 );
sub_c( d3.re , d3.im , d1.re , d1.im , d1 );
mul_c( d1.re , d1.im , 1/2 , 0 , data );
end;
3:begin
mul_c( t , t , cosh(t)*cos(t) , sinh(t)*sin(t) , d1);
div_c( d1.re , d1.im , sinh(t)*cos(t) , cosh(t)*sin(t) , d1 );
mul_c( d1.re , d1.im , cosh(t)*cos(t) -1 , sinh(t)*sin(t) , d1 );
sub_c( d1.re , d1.im , 1/a1 , 0 , d1 );
add_c( -1 +1/a1 , 0 , cosh(t)*cos(t) , sinh(t)*sin(t) , d2 );
div_c( d1.re , d1.im , d2.re , d2.im , d3 );
mul_c( t , t , cosh(t)*cos(t) , sinh(t)*sin(t) , d1);
div_c( d1.re , d1.im , sinh(t)*cos(t) , cosh(t)*sin(t) , d1 );
mul_c( d1.re , d1.im , cosh(t)*cos(t) -1 , sinh(t)*sin(t) , d1 );
mul_c( t , t , sinh(t)*cos(t) , cosh(t)*sin(t) , d2);
sub_c( d2.re , d2.im , cosh(t)*cos(t) , sinh(t)*sin(t) , d2 );
mul_c( d2.re , d2.im , 1/a1 , 0 , d2 );
add_c( d1.re , d1.im , d2.re , d2.im , d1 );
mul_c( 1/a1 , 0 , cosh(t)*cos(t) , sinh(t)*sin(t) , d2);
add_c( d2.re , d2.im , cosh(t)*cos(t) -1 , sinh(t)*sin(t) , d2 );
div_c( d1.re , d2.im , d2.re , d2.im , d1 );
sub_c( d3.re , d3.im , d1.re , d1.im , d1 );
mul_c( d1.re , d1.im , 1/2 , 0 , data );
end;
4:begin
mul_c( t , t , cosh(t)*cos(t) , sinh(t)*sin(t) , d1);
div_c( d1.re , d1.im , sinh(t)*cos(t) , cosh(t)*sin(t) , d1 );
mul_c( d1.re , d1.im , cosh(t)*cos(t) -1 , sinh(t)*sin(t) , d1 );
add_c( d1.re , d1.im , 1/a1 , 0 , d1 );
add_c( -1 +1/a1 , 0 , cosh(t)*cos(t) , sinh(t)*sin(t) , d2 );
div_c( d1.re , d1.im , d2.re , d2.im , d3 );
mul_c( t , t , cosh(t)*cos(t) , sinh(t)*sin(t) , d1);
div_c( d1.re , d1.im , sinh(t)*cos(t) , cosh(t)*sin(t) , d1 );
mul_c( d1.re , d1.im , cosh(t)*cos(t) -1 , sinh(t)*sin(t) , d1 );
mul_c( t , t , sinh(t)*cos(t) , cosh(t)*sin(t) , d2);
add_c( d2.re , d2.im , cosh(t)*cos(t) , sinh(t)*sin(t) , d2 );
mul_c( d2.re , d2.im , 1/a1 , 0 , d2 );
add_c( d1.re , d1.im , d2.re , d2.im , d1 );
mul_c( 1/a1 , 0 , cosh(t)*cos(t) , sinh(t)*sin(t) , d2);
add_c( d2.re , d2.im , cosh(t)*cos(t) -1 , sinh(t)*sin(t) , d2 );
div_c( d1.re , d1.im , d2.re , d2.im , d1 );
sub_c( d3.re , d3.im , d1.re , d1.im , d1 );
mul_c( d1.re , d1.im , 1/2 , 0 , data );
end;
end;
'a':begin
al := 1/fi;
mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , 1+al , 0 , d1 );
mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , al , 0 , d2 );
add_c( d2.re , d2.im , 1 , 0 , d2 );
div_c( d1.re , d1.im , d2.re , d2.im , data );
end;
'b':begin
al := 1/fi;
mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , 1+al , 0 , d1 );
mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , al , 0 , d2 );
add_c( d2.re , d2.im , -al+1 , 0 , d2 );
div_c( d1.re , d1.im , d2.re , d2.im , data );
end;
'c':begin
al := 1/fi;
mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , 1+2*al , 0 , d1 );

```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่าในรูปแบบใดก็ตาม ถือว่าห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    add_c( dl.re , dl.im , -2*al , 0 , dl );
    mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , al , 0 , d2 );
    add_c( d2.re , d2.im , -al+1 , 0 , d2 );
    div_c( dl.re , dl.im , d2.re , d2.im , data );
end;
'd':begin
    al := fi;
    mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , al+1 , 0 , dl );
    sub_c( dl.re , dl.im , 1 , 0 , dl );
    div_c( dl.re , dl.im , cosh(t)*cos(t)-1+al , sinh(t)*sin(t) , data );
end;
'e':begin
    al := 1/fi;
    mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , 2+al , 0 , dl );
    add_c( dl.re , dl.im , -2 , 0 , dl );
    add_c( cosh(t)*cos(t) , sinh(t)*sin(t) , -1+al , 0 , d2 );
    div_c( dl.re , dl.im , d2.re , d2.im , data );
end;
'f':begin
    al := 1/fi;
    mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , 1+al , 0 , dl );
    add_c( cosh(t)*cos(t) , sinh(t)*sin(t) , -1+al , 0 , d2 );
    div_c( dl.re , dl.im , d2.re , d2.im , data );
end;
'g':begin
    al := 1/fi;
    mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , 2+al , 0 , dl );
    add_c( dl.re , dl.im , -2 , 0 , dl );
    mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , 1+al , 0 , d2 );
    add_c( d2.re , d2.im , -1 , 0 , d2 );
    div_c( dl.re , dl.im , d2.re , d2.im , data );
end;
'h':begin
    al := 1/fi;
    mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , 1+al , 0 , dl );
    mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , 1+al , 0 , d2 );
    add_c( d2.re , d2.im , -1 , 0 , d2 );
    div_c( dl.re , dl.im , d2.re , d2.im , data );
end;
'i':begin
    al := 1/fi;
    mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , 1+al , 0 , dl );
    mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , 1+al , 0 , d2 );
    add_c( d2.re , d2.im , -1 , 0 , d2 );
    div_c( dl.re , dl.im , d2.re , d2.im , data );
end;
'j':begin
    al := 1/fi;
    mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , 2+al , 0 , dl );
    add_c( dl.re , dl.im , -2 , 0 , dl );
    mul_c( cosh(t)*cos(t) , sinh(t)*sin(t) , 1+al , 0 , d2 );
    add_c( d2.re , d2.im , -1 , 0 , d2 );
    div_c( dl.re , dl.im , d2.re , d2.im , data );
end;
end;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

const XScreenMax = 640;           { 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 = $A000;    { Location of the hardware screen
  SCF = 1.371428571;          { Screen Scaling adjustment Factor
  EGA : boolean = true;

```

```

type ScreenType = array[0..ScreenSizeGlb] of integer;
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;
    RowBase       : array[0..349] of integer;

```

```

procedure ClearScreen;
begin
  FillChar(Mem[GrafBase:0000], ScreenSizeGlb shl 1,0);
end;

```

```

procedure TextMode;
begin
  regs.ax := $0003;
  intr($10,regs);
end;

```

```

procedure Palette(PaletteNumber : byte);
begin
  Regs.AH := $0B;           { Set Color Palette }
  Regs.BH := PaletteNumber and $7F; { Palette Color (0..127) }
  Regs.BL := $00;           { Color Value }
  Intr ($10,Regs);
end;

```

```

procedure BackGround(Color : Byte);
begin
  Regs.AH := $0B;           { Set Color Palette }
  Regs.BH := Color;         { BackGround Color (0..15) }
  Regs.BL := $01;           { Color Value }
  Intr ($10,Regs);
end;

```

```

procedure GraphicsMode;
var i : integer;
begin
  for i := 0 to YScreenMax-1 do
    RowBase[i] := i*80;
  GrafBase := HardWareGrafBase;
  Regs.AX := $0010;         { Hi-Res 640 X 350 }
  Intr ($10,Regs);
end;

```

```

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

```

```

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

procedure SwapScreen;
const XMaxGlb = 79;           { Number of bytes -1 in one screen line }
var
    ScanLine,LineBase,RamScrSeg : integer;
    LineBuffer                   : array[0..XMaxGlb] of byte;
begin
    RamScrSeg := Seg(ScreenGlb^);
    for ScanLine := 0 to YscreenMax-1 do
        begin
            LineBase := RowBase[Scanline];
            Move(Mem[GrafBase:LineBase],LineBuffer,XMaxGlb+1);
            Move(Mem[RamScrSeg:LineBase],Mem[GrafBase:LineBase],XMaxGlb+1);
            Move(LineBuffer,Mem[RamScrSeg:LineBase],XMaxGlb+1);
        end;
    end;
end;

procedure DrawPoint(X,Y : integer);
begin
    inline(
        $8B/$46/ <Y           {           MOV     AX,   Y
        /$D1/$E0              {           SHL     AX,   1
        /$97                   {           XCHG   AX,   DI
        /$8B/$85/RowBase      {           MOV     AX,   [DI+RowBase]
                                {
                                {           AX has RowBase(Y)

        /$8B/$5E/ <X           {           MOV     BX,   X
        /$89/$DA              {           MOV     DX,BX   Save X in DX
        /$B1/$03              {           MOV     CL,03
        /$D3/$EB              {           SHR     BX,CL   BX = X div 8
        /$01/$C3              {           ADD     BX,AX

                                {           AX has RowBase(Y) + X div 8

        /$88/$D1              {           MOV     CL,DL
        /$80/$E1/$07          {           AND     CL,07
        /$B2/$80              {           MOV     DL,80
        /$D2/$EA              {           SHR     DL,CL
        /$8E/$06/GrafBase     {           MOV     ES,GrafBase
        /$80/$3E/ColorGlb/$FF {           JMP     ColorGlb,FF
        /$75/$05              {           JNZ     ZeroPix

        /$26                  {           ES:
        /$08/$17              {           OR      [BX],DL

```

```

/$EB/$05      {           JMP      3420
               { ZeroPix:
/$F6/$D2      {           NOT      DL
/$26           {           ES:
/$20/$17      {           AND      [BX],DL
);            { Done:
end;

```

```

procedure DPColor(x,y,Color : integer);
begin
  Regs.BH := $00;      { Page number }
  Regs.CX := x;        { Column number }
  Regs.DX := y;        { Row number }
  Regs.AL := Color;    { Color number }
  Regs.AH := $0C;      { Write Dot }
  Intr ($10,Regs);
end;

```

```

function GetDotColor (x,y : integer) : integer;
begin
  Regs.BH := $00;      { Page number }
  Regs.CX := x;        { Column number }
  Regs.DX := y;        { Row number }
  Regs.AH := $0D;      { Read Dot }
  Intr ($10,Regs);
  GetDotColor := Regs.AL;
end;

```

```

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

```

```

function BaseAddress(Y: integer): integer;
begin
  BaseAddress := Y * 80;
end;

```

```

function PD(x,y: integer): boolean;
begin
  PD := (ColorGlb=0) xor (mem[GrafBase:BaseAddress(y) + x shr 3]
    and (128 shr (x and 7)) <> 0);
end;

```

```

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)));
    end;
  end;

```

เอกสารนี้เป็นทรัพย์สินทางปัญญาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น ขอสงวนสิทธิ์ในเอกสารฉบับนี้ให้อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    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;

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

procedure SetDate( Year,Month,Day : integer);
begin
    Regs.CX := Year;
    Regs.DH := Month;
    Regs.DL := Day;
    Regs.AH := $2B;
    Intr ($21,Regs);
end;

procedure GetDate( Var Year,Month,Day,DayofWeek : integer);
begin
    Regs.AH := $2A;
    Intr($21,Regs);
    Year := Regs.CX;
    Month := Regs.DH;
    Day := Regs.DL;
    DayofWeek := Regs.AL;
end;

procedure SetTime(Hour,Minute,Second,Sec100 : integer);
begin
    Regs.CH := Hour;
    Regs.CL := Minute;
    Regs.DH := Second;

```

```

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;

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

procedure SetDate( Year,Month,Day : integer);
begin
Regs.CX := Year;
Regs.DH := Month;
Regs.DL := Day;
Regs.AH := $2B;
Intr ($21,Regs);
end;

procedure GetDate( Var Year,Month,Day,DayofWeek : integer);
begin
Regs.AH := $2A;
Intr($21,Regs);
Year := Regs.CX;
Month := Regs.DH;
Day := Regs.DL;
DayofWeek := Regs.AL;
end;

procedure SetTime(Hour,Minute,Second,Sec100 : integer);
begin
Regs.CH := Hour;
Regs.CL := Minute;

```

```
Regs.DH := Second;  
Regs.DL := Sec100;  
Regs.AH := $2D;  
Intr ($21,Regs);  
end;
```

```
procedure GetTime(Var Hour,Minute,Second,Sec100 : integer);  
begin  
  Regs.AH := $2C;  
  Intr($21,Regs);  
  Hour := Regs.CH;  
  Minute := Regs.CL;  
  Second := Regs.DH;  
  Sec100 := Regs.DL;  
end;
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้