สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การใช้คอมพิวเตอร์ช่วยในการออกแบบวงจรกรองความถี่ต่ำและโนโมกราฟ

Nomograph and Low - Pass Filter by Computer - Aid Design

T 086860

นายฉัตรชัย คำปล้อง
นายรณกร มโนเสงี่ยม
นายสมพงษ์ ทานอก

เลขหมู่.....
เลขทะเบียน 86860
วัน,เดือน,ปี 16 ส.ค. 2552

.b 10908818
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตรบัณฑิต
ภาควิชาเทคนิคอุตสาหกรรม คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2541

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

การใช้คอมพิวเตอร์ช่วยในการออกแบบวงจรกรองความถี่ต่ำ และโนโมกราฟ

NOMOGRAPH AND LOW - PASS FILTER BY COMPUTER - AID DESIGN

นายฉัตรชัย คำปล้อง เลขประจำตัว 39013307

นายรณกร มโนเสงี่ยม เลขประจำตัว 39013321

นายสมพงษ์ ทานอก เลขประจำตัว 39013330

อาจารย์ที่ปรึกษา

อ. นภพินทุ์ อนันตรศิริชัย

รศ. ดร. กนก เจนจิระพงศ์เวช

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตรบัณฑิต

ภาควิชาเทคนิคอุตสาหกรรม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NOMOGRAPH AND LOW-PASS FILTER BY COMPUTER - AID DESIGN

CHATCHAI	KUMPLONG	39.013307
RONNAKRON	MANOSNGIAM	39.013321
SOMPONG	TANOK	39.013330

ADVISOR

NOPPIN ANANTRASIRICHAI

KANOK JANJIRAPONGVEJ

PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIRMENTS

FOR THE BACHELOR'S DEGREE

DEPARTMENT OF INDUSTRIAL TECHNOLOGY

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การใช้คอมพิวเตอร์ช่วยในการออกแบบวงจรกรองความถี่ต่ำและโนโมกราฟ NOMOGRAPH AND LOW - PASS FILTER BY COMPUTER - AID DESIGN
ชื่อนักศึกษา	นายฉัตรชัย คำปล้อง นายรณกร มโนเสงี่ยม นายสมพงษ์ ทานอก
อาจารย์ที่ปรึกษา	อ. นภพินทุ์ อนันตรศิริชัย รศ.ดร. กนก เจนจิระพงศ์เวช
ภาควิชา	เทคนิคอุตสาหกรรม
ปีการศึกษา	๒๕๔๑

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
อนุมัติให้นับปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตรบัณฑิต

คณะกรรมการสอบปริญญานิพนธ์

.....ประธานกรรมการ
( )

.....กรรมการ
( )

.....กรรมการ
( )

.....กรรมการ
( )

.....กรรมการ
( )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

หัวข้อปริญญานิพนธ์	การใช้คอมพิวเตอร์ช่วยในการออกแบบวงจรกรองความถี่ต่ำและโนโมกราฟ
ชื่อนักศึกษา	นายฉัตรชัย คำปล้อง
	นายรณกร มโนเสงี่ยม
	นายสมพงษ์ ทานอก
อาจารย์ที่ปรึกษา	อ. นภพินทุ์ อนันตรศิริชัย
	รศ. ดร. กนก เจนจิระพงศ์เวช
ภาควิชา	เทคนิคอุตสาหกรรม
คณะ	วิศวกรรมศาสตร์
ปีการศึกษา	๒๕๔๑

บทคัดย่อ

เนื่องจากในปัจจุบัน ถ้าเราต้องการทราบค่า order ของ filter ชนิดหนึ่ง ๆ ก็สามารถทำได้โดยเปิดดูจากหนังสือคู่มือของ filter ชนิดนั้น ๆ ซึ่งบางครั้งอาจจะทำให้เสียเวลาและไม่สะดวก ดังนั้น กลุ่มของข้าพเจ้าจึงได้ทำการคิดค้น program ขึ้นมา program หนึ่ง เพื่อใช้สำหรับหาค่า order ของ filter แทนการเปิดดูจากหนังสือคู่มือ ซึ่งจะทำให้เกิดความสะดวกรวดเร็วในการหาค่าของ order มากกว่า และยังมีความถูกต้องแม่นยำสูงอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROJECT REPORT TITLE NOMOGRAPH AND LOW - PASS FILTER BY COMPUTER - AID DESIGN

NAME CHATCHAI KUMPLONG
RONNAKORN MANOSNGIAM
SOMPONG TANOK

ADVISOR NOPPIN ANANTRASIRICHAI
KANOK JANJIRAPONGVEJ

DEPARTMENT OF INDUSTRIAL TECHNOLOGY
FACULTY OF ENGINEERING
ACADEMIC YEAR 1998

ABSTRACT

Nowadays, the order n^{th} of any low-pass filter can be determined using the corresponding manual to that specific filter. This method is however time-consuming and, in most cases, very inconvenient. A user-friendly computer program has been developed in this study and is presented as an alternative mean for fast determination of the order n^{th} of any filter, providing high accuracy of predicted results.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การที่ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ ทางคณะผู้จัดทำต้องขอกราบขอบพระคุณท่านอาจารย์ที่ปรึกษาโครงงานทั้ง 2 ท่าน คือ อาจารย์ นภพินทุ์ อนันตรศิริชัย และ อาจารย์ ดร. กนก เจนจิระพงศ์เวช เป็นอย่างสูง ที่ได้กรุณาให้คำปรึกษา คำแนะนำ และข้อเสนอแนะด้วยดีมาตลอด รวมไปถึงการให้ยืมตำราต่าง ๆ ที่จำเป็นตลอดระยะเวลาในการทำโครงงาน และขอขอบพระคุณ อาจารย์ อรลาภ แสงอรุณ ที่ได้ให้ความอนุเคราะห์ในการให้ยืมหนังสือ “ELECTRONIC FILTER DESIGN HANDBOOK” มาประกอบในการทำโครงงาน

ขอขอบคุณห้องสมุดทั้งที่สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง และที่ภาควิชาเทคนิคอุตสาหกรรม ศูนย์นนทบุรี ที่เป็นแหล่งข้อมูลสำคัญในการค้นคว้าโครงงาน

สุดท้ายนี้ที่จะลืมไม่ได้ ขอกราบขอบพระคุณ บิดา - มารดา ผู้ให้กำเนิด และเพื่อนๆ ที่คอยให้กำลังใจด้วยดีเสมอมา

ตุลาคม ๒๕๔๑

ฉัตรชัย คำปล้อง

รณกร มโนเสงี่ยม

สมพงษ์ ทานอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทคัดย่อ	ก
ABSTRACT	ข
กิตติกรรมประกาศ	ค
บทนำ	1
บทที่ 1 ประวัติความเป็นมาของ NOMOGRAPH	4
บทที่ 2 การประยุกต์ NOMOGRAPH มาใช้กับ FILTER	12
บทที่ 3 ULTRASPHERICAL FILTER	19
บทที่ 4 โครงงาน	22
4.1 แนวทางการออกแบบโปรแกรม	22
4.2 การทำงานของโปรแกรม	23
NOMOGRAPH ที่ได้จากหน้าจอ MONITOR	63
ปัญหาและอุปสรรค	77
สรุปและวิจารณ์	78
ภาคผนวก	79
ก. คู่มือการใช้งานของ PROGRAM	80
ข. LISTING PROGRAM	85
ค. NOMOGRAPH จริงที่ใช้ในการหา ORDER	110
บรรณานุกรม	124

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำนำ

เนื่องจากในปัจจุบันนี้ computer ได้เข้ามามีบทบาทกับชีวิตของเรามากขึ้นทุกที โดยเฉพาะอย่างยิ่งในทางวิศวกรรมและอุตสาหกรรม เช่นการใช้ computer ควบคุมเครื่องจักรกลในโรงงาน หรือการใช้ computer ในการควบคุมดาวเทียม ซึ่งเป็นที่ทราบกันดีอยู่แล้วว่า computer นั้นมีประสิทธิภาพในการปฏิบัติงานสูงมาก ดังนั้นจึงจะเป็นการดีกว่าถ้าจะให้ computer ทำงานแทนมนุษย์ซึ่งอาจจะทำงานผิดผลาด ได้ทุกเวลา

จากแนวความคิดอันนี้ จึงก่อให้เกิดความคิดในการเขียน program เพื่อที่จะใช้ computer นั้นทำงานแทนมนุษย์ โดยการหาข้อมูลที่อยู่ภายในหนังสือคู่มือ (Handbook) ของอุปกรณ์หรือเครื่องมือเครื่องใช้ต่างๆ (ซึ่งจะมีความละเอียดมาก) ขึ้นมาเพื่อที่จะช่วยลดความผิดพลาดจากการหาข้อมูลในหนังสือคู่มือของมนุษย์เอง

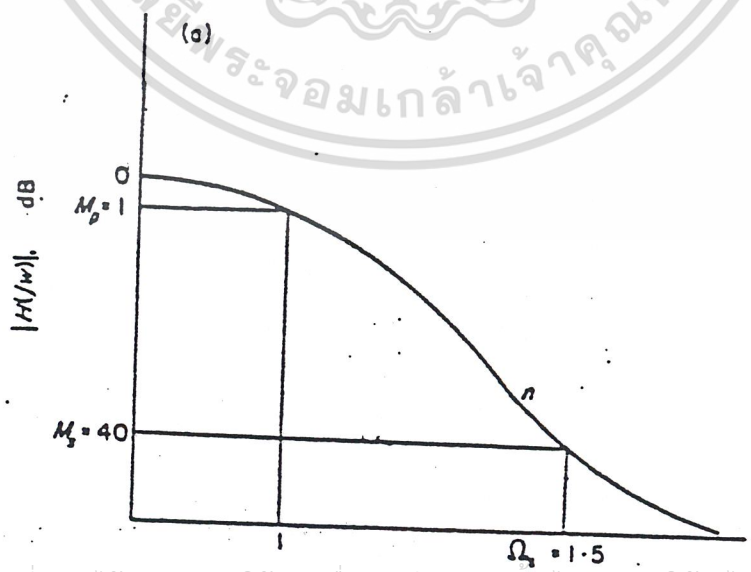
ดังเช่น การอ่านค่า order ของ filter นั้นก็จำเป็นจะต้องเปิดอ่านจากหนังสือคู่มือ ซึ่งถ้าเรามีความต้องการอ่านค่า order ของ filter หลายๆ ตัวพร้อมๆ กันก็อาจจะทำให้เกิดความผิดพลาดในการอ่านค่าได้ แต่ถ้าเรามี program ที่สามารถหาค่า order ของ filter ออกมาได้ก็จะช่วยให้ความผิดพลาดในการหาค่า order ลดลง ซึ่งมีความสะดวกรวดเร็วกว่าการเปิดจากหนังสือคู่มือ นอกจากนั้นยังมีความถูกต้องแม่นยำสูงอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทนำ

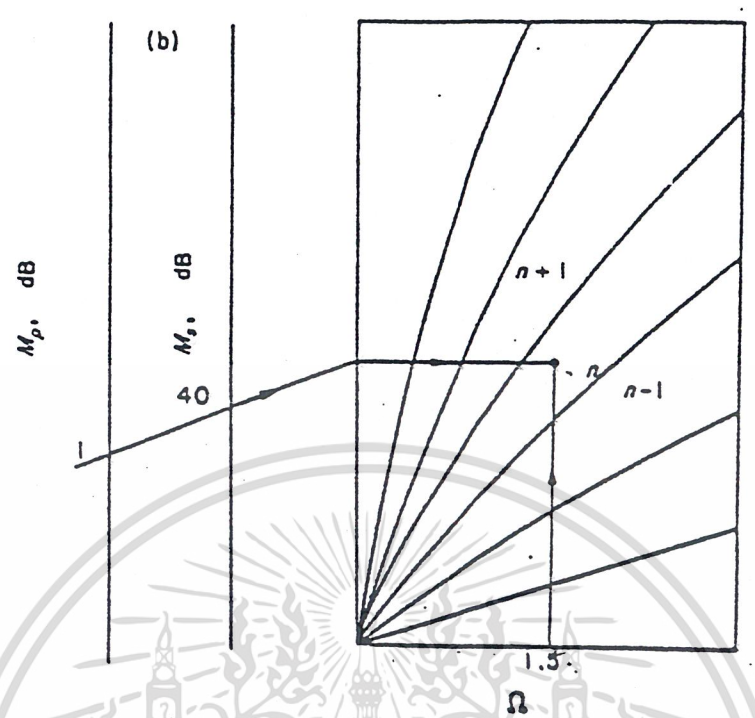
ความเป็นมาของโครงงาน

ในระบบการสื่อสารแบบ digital นั้นอุปกรณ์ที่เราจะพบบ่อยๆทั้งในภาคส่งและภาครับก็คือ Low - Pass Filter (LPF) และเพื่อให้การสื่อสารเป็นไปอย่างราบรื่น LPF ก็จำเป็นจะต้องมีความคมของ responses สูง ซึ่งก็จะขึ้นอยู่กับ order ของ LPF นั้นเอง (ดังรูปที่ 1a) ดังนั้นในการออกแบบ LPF จึงจำเป็นจะต้องทราบค่าของ order ที่ถูกต้อง ซึ่งจะมีเครื่องมืออยู่ชนิดหนึ่งที่จะช่วยในการหาค่า order ของ LPF โดยเฉพาะ นั้นก็คือ " Nomograph " ซึ่งภายใน nomograph นั้นจะแสดงถึงค่าความสัมพันธ์ของ parameter 4 ตัว อันประกอบด้วย Maximum Passband Attenuation (Mp) , Minimum Stopband Rejection (Ms) , Normalized Frequency (Ω) และ order (n) โดยความสัมพันธ์จะอยู่ในลักษณะของการที่ถ้าทราบค่า Mp กับ Ms ก็จะสามารถหา γ (gamma) ได้และเมื่อทราบค่า γ กับ Ω (ซึ่งก็คือค่า Frequency Stopband (fs) ต่อ Frequency Passband (fp)) ก็จะหาค่า n ของ LPF ออกมาได้ (ดังรูปที่ 1b) ซึ่ง nomograph นั้นเป็นเครื่องมือที่มีความสะดวกรวดเร็วในการหา n มาก แต่เพื่อเป็นการปรับปรุงประสิทธิภาพของ nomograph ให้ดียิ่งขึ้นรวมไปถึงเพื่อให้ก้าวทันโลกในยุคปัจจุบันที่ได้มีการนำเอา computer มาประยุกต์ใช้กับงานด้านวิศวกรรมอย่างแพร่หลาย จึงได้เกิดแนวความคิดในการเขียน program ของnomograph ขึ้นซึ่งเป็น program ที่จะช่วยให้สามารถหาค่าของ n ได้อย่างรวดเร็วมากยิ่งขึ้นมีความถูกต้องแม่นยำสูงขึ้น อีกทั้งยังพกพาไปไหนมาไหนได้สะดวกเพราะเป็น diskettes เพียงแผ่นเดียว



รูปที่ 1 a) magnitude responses ของ Low - Pass Filter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



b) nomograph ของ magnitude responses

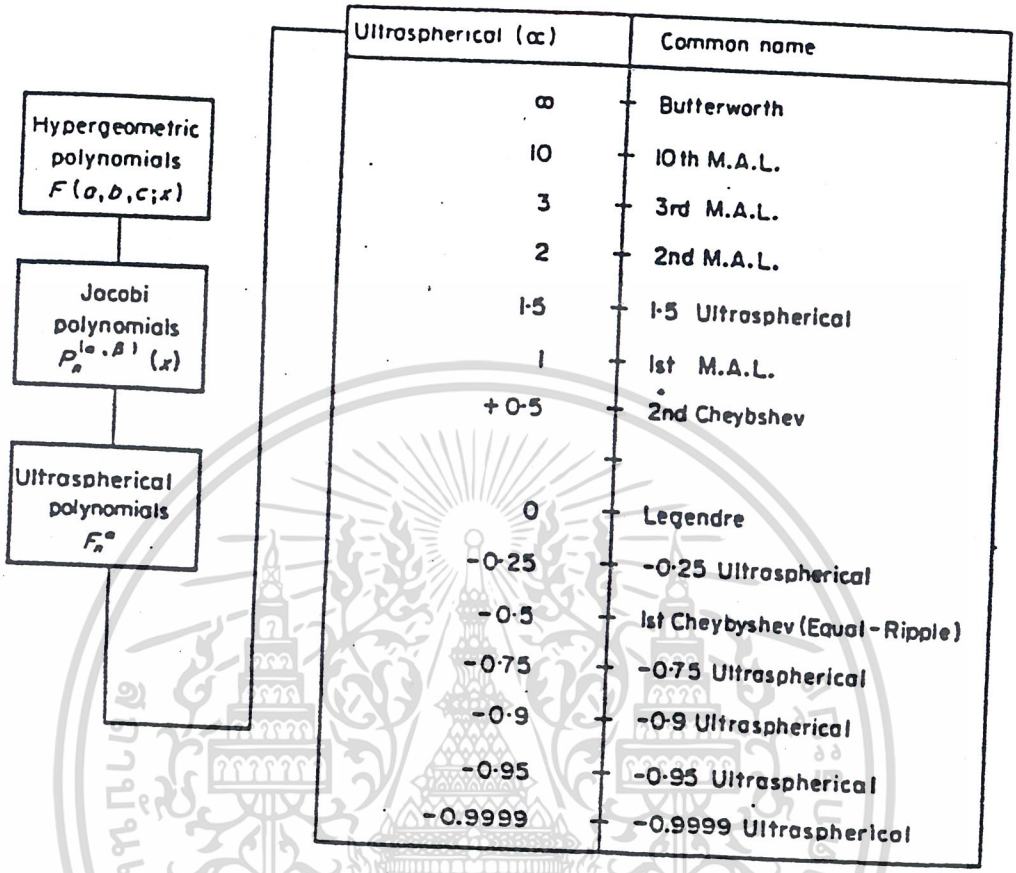
วัตถุประสงค์ของโครงงาน

1. เพื่อรู้จักการใช้ computer ในการสร้าง program ของ nomograph สำหรับแก้ปัญหาด้าน Low - Pass Filter
2. เพื่อให้เกิดความสะดวกรวดเร็วในการหาค่า parameter ต่าง ๆ ด้วยการใช้ computer
3. เพื่อเป็นการนำเทคโนโลยีของ computer มาใช้ให้เกิดประโยชน์สูงสุด

ขอบเขตของโครงงาน

program ของ nomograph ที่ได้ทำการเขียนขึ้นมานี้จะสามารถใช้ได้กับ Low - Pass Filter 14 ชนิด โดยที่แต่ละชนิดนั้นจะมีสมการ polynomails มาจาก recursion formula ที่ค่า α (alpha) ต่าง ๆ กัน (ภายในช่วง $-1 < \alpha < \infty$) ของ Ultraspherical polynomails ซึ่ง Ultraspherical polynomails จะได้มาจาก Jacobi polynomails ที่ค่า $\alpha=\beta$ ซึ่งมันมีต้นกำเนิดมาจาก Hypergeometric polynomails ดังตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



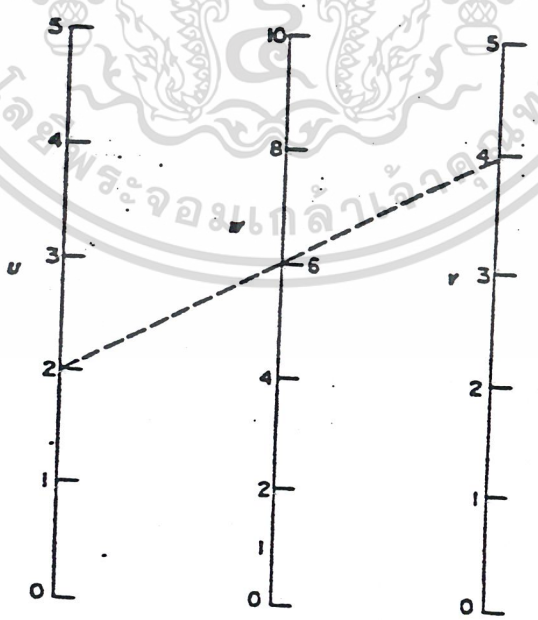
ตาราง แสดงค่า α และชื่อสามัญของ Classical Filter ชนิดต่าง ๆ ที่มีต้นกำเนิดมาจาก Hypergeometric polynomails

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
ประวัติและความเป็นมาของ NOMOGRAPH

nomograph เป็นคำในภาษากรีกโบราณ ได้มาจากการรวมกันของคำว่า nomos (แปลว่า กฎ) และคำว่า graphein (แปลว่า การเขียน) ซึ่งตามตัวหนังสือแล้วเป็นการแสดงด้วยกราฟของ กฎเกณฑ์ หรือความสัมพันธ์ ทางคณิตศาสตร์ nomography เป็นการศึกษาและการนำ nomograph มาใช้ ซึ่งถูกนำมาใช้ครั้งแรกโดย นักคณิตศาสตร์ชาวฝรั่งเศส ชื่อ M. d'Ocagne ในปี ค.ศ 1891 ในหนังสือ " Traite De Nomographie " ของเขา nomography ได้มีการแพร่หลายอย่างรวดเร็วไปยังส่วนต่าง ๆ ของโลก ในบางครั้งอาจเรียกได้อีกชื่อหนึ่งว่า " Alignment charts "

nomograph เป็นการแสดงผลแทนสมการที่มี 3 ตัวแปรขึ้นไปด้วยกราฟ โดยในอันดับแรกเราจะเริ่มอธิบายถึง nomograph ที่ใช้กับ 3 ตัวแปรก่อน โดยแต่ละตัวแปรจะถูก plot ลงบนเส้น ซึ่งถ้าเรารู้ 2 จุดเราก็จะสามารถหาจุดที่ 3 ได้ เมื่อลากเส้นตรงระหว่างจุด 2 จุดที่ทราบค่า จุดตัดที่เกิดขึ้นจากเส้นที่ 3 ก็จะเป็นค่าของตัวแปรที่ 3 ตัวอย่างของ nomograph แสดงในรูปที่ 2 ตอนนี้เราจะพิจารณาจากพื้นฐานทางทฤษฎีสำหรับ nomograph การพัฒนา nomograph ของ d'Ocagne อยู่ บน projective ทางเรขาคณิตและหลักการของการ duality โดยการใช้พิกัดของเส้นขนาน เราจะใช้แนวทางของ Allcock และ Epstein ซึ่งเป็นการใช้ determinant และระบบพิกัดฉาก



รูปที่ 2 แสดง nomograph ของสมการ $u + v = w$ มื่อให้ $u = 2$ และ $v = 4$ จะได้ $w = 6$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการพื้นฐานของ nomograph ทุกชนิดคือ จุด 3 จุดใด ๆ เราให้เป็น (x_1, y_1), (x_2, y_2) และ (x_3, y_3) ที่อยู่บนเส้นตรงเดียวกัน เมื่อค่าของ determinant เป็นศูนย์

$$\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} = 0 \qquad (1)$$

เมื่อค่า det ใน (1) เป็นศูนย์แสดงให้เป็นจริงอย่างง่าย ๆ ในรูปที่ 3 ดังนี้ ให้ u , v และ w เป็น 3 ฟังก์ชัน ซึ่งมีจุด (x_1, y_1), (x_2, y_2) และ (x_3, y_3) อยู่ดังแสดงในรูปที่ 3 จากนั้นให้ A , B และ C เป็นตำแหน่งของจุดตัดที่เกิดขึ้นโดยเส้นตรงใดๆ ที่ลากไปตัดกับ curve ทั้ง 3 ตามลำดับ โดยกำหนดให้จุด (x_2, y_1) เป็น D และจุด (x_3, y_2) เป็น E ดังนั้น $AD = x_2 - x_1$ และ $DB = y_2 - y_1$ ในทำนองเดียวกัน $EC = y_3 - y_2$ และ $BE = x_3 - x_2$ เมื่อทำการเข้าสมการตามสามเหลี่ยมคล้ายของ ΔABD และ ΔBCE เราจะได้ว่า

$$\frac{y_2 - y_1}{x_2 - x_1} = \frac{y_3 - y_2}{x_3 - x_2} \qquad (2)$$

จัดรูปใหม่ได้

$$x_1 y_2 + x_2 y_3 + x_3 y_1 - x_1 y_3 - x_2 y_1 - x_3 y_2 = 0 \qquad (3)$$

ซึ่งสามารถเขียนเป็น det ในรูป

$$\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} = 0 \qquad (4)$$

และถ้าให้ x_1 กับ y_1 เป็นฟังก์ชันของ u , x_2 กับ y_2 เป็นฟังก์ชันของ v และ x_3 กับ y_3 เป็นฟังก์ชันของ w จะทำให้สามารถเขียน det ได้ใหม่เป็น

$$\begin{vmatrix} g_1(u) & f_1(u) & 1 \\ g_2(v) & f_2(v) & 1 \\ g_3(w) & f_3(w) & 1 \end{vmatrix} = 0 \qquad (5)$$

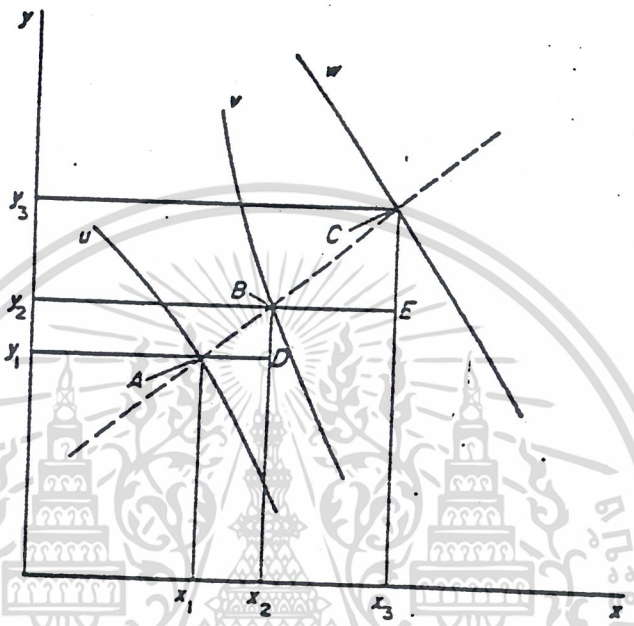
ซึ่ง det นี้ส่วนมากเรียกว่า "basic determinant" ถ้าสมการใด ๆ ก็ตามสามารถเขียนให้อยู่ในรูป basic det ได้มันก็จะสามารถนำไปสร้าง nomograph ที่ประกอบด้วย 3 curve บนระนาบ (x , y) ซึ่งแยกจากกันและแทนตัวแปร u , v และ w

basic determinant ของ nomograph ใด ๆ ก็ตาม ต้องเป็นไปตามเงื่อนไขดังต่อไปนี้

1. ค่าของ det ต้องเท่ากับศูนย์
2. แต่ละแถวจะต้องมีตัวแปรเพียงแค่ตัวเดียว
3. สมาชิกตัวสุดท้ายในแต่ละแถวจะต้องมีค่าเท่ากับ 1 เสมอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

nomograph สร้างมาจาก basic det โดยการ plot จุด $[\, g_1(u) = x_1 \,,\, f_1(u) = y_1 \,]$, $[\, g_2(v) = x_2 \,,\, f_2(v) = y_2 \,]$ และ $[\, g_3(w) = x_3 \,,\, f_3(w) = y_3 \,]$ เหล่านี้ สำหรับค่าต่าง ๆ ของ u , v และ w ทั้งหมดนี้เราจะได้เส้นของ nomograph ในรูปที่ 2



รูปที่ 3 แสดงฟังก์ชันทั้ง 3 ของ u , v และ w

ขั้นตอนแรกในการออกแบบ nomograph คือการจัดรูปของ basic det วิธีทั่วๆ ไปในการสร้าง basic det จากสูตรใด ๆ ที่มี 3 ตัวแปร ประกอบด้วยการสร้างสมการ 3 สมการที่อยู่ในรูปของ $Ax + By + C = 0$ จากสมการเหล่านี้ det

$$\begin{vmatrix} A_1 & B_1 & C_1 \\ A_2 & B_2 & C_2 \\ A_3 & B_3 & C_3 \end{vmatrix} = 0 \qquad (6)$$

อาจได้มาจากการเขียนสัมประสิทธิ์โดยตรงตามลำดับที่แสดงข้างบน เมื่อนำเอาข้อได้เปรียบของหลักการจัดการกับ det มาใช้ det อันนี้จะถูกเปลี่ยนให้อยู่ในรูปของ basic nomographic เราจะแสดงการจัดรูปของ basic det ด้วยตัวอย่างดังนี้

เราออกแบบ nomograph ที่แสดงในรูปที่ 2 ได้จากสมการ

$$u + v = w \qquad (7)$$

เพราะสมการนี้จะใช้ออกแบบ filter ภายหลัง ขั้นแรกเราจะเขียนสมการใหม่เป็น

$$u + v - w = 0 \qquad (8)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากนั้นเรากำหนดสมการตัวแปรให้ x = u และ y = v เราจะได้ 3 สมการที่ต้องการคือ

$$\begin{aligned} x-u&=0\\ y-v&=0\\ x+y-w&=0 \end{aligned} \tag{9}$$

แทนค่าสมการเหล่านี้ในรูปของ det จาก (6) จะได้

$$\begin{vmatrix} 1 & 0 & -u\\ 0 & 1 & -v\\ 1 & 1 & -w \end{vmatrix} = 0 \tag{10}$$

ขั้นตอนที่เหลือคือการเปลี่ยน (10) ให้อยู่ในรูปของ basic det ตาม (5) ซึ่งทำได้โดยใช้หลักการจัดการกับ det ดังที่จะกล่าวดังต่อไปนี้

จากสมการ (10)

- แทน C_1 ด้วยผลบวกของ C_1 กับ C_2 และคูณ C_3 ด้วย (- 1) จะได้

$$\begin{vmatrix} 1 & 0 & u\\ 1 & 1 & v\\ 2 & 1 & w \end{vmatrix} = 0 \tag{11}$$

- หาร R_3 ด้วย 2 จะได้

$$\begin{vmatrix} 1 & 0 & u\\ 1 & 1 & v\\ 1 & 1/2 & w/2 \end{vmatrix} = 0 \tag{12}$$

- สุดท้ายให้ทำการสลับ Column โดยให้ C_1 แทน C_3 , C_2 แทน C_1 และ C_3 แทน C_2 จะได้

$$\begin{vmatrix} 0 & u & 1\\ 1 & v & 1\\ 1/2 & w/2 & 1 \end{vmatrix} = 0 \tag{13}$$

ซึ่งจะได้เป็น basic det ออกมา เราสามารถทำขั้นตอนทั้งหมดที่กล่าวมาได้เพราะค่าของ det = 0 nomograph ที่แสดงดังรูปที่ 2 ถูกสร้างขึ้นจาก basic det นี้ และเมื่อเปรียบเทียบ (13) กับ (4) จะได้ว่า

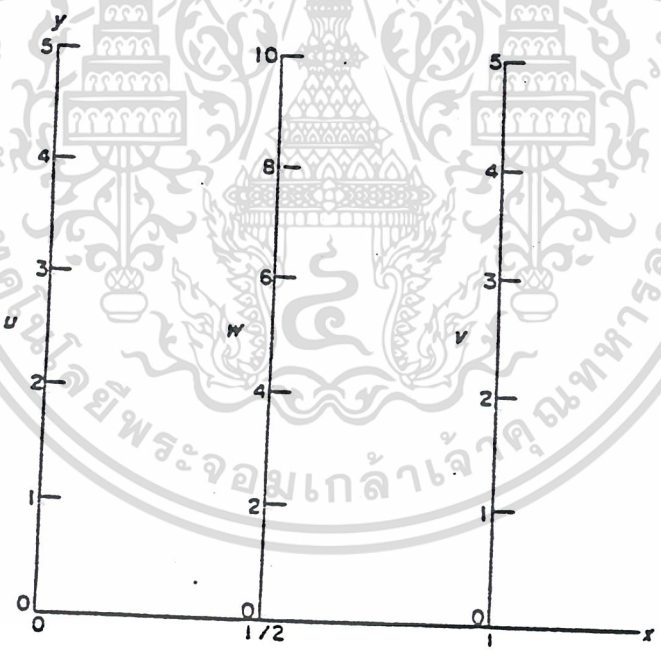
$$\begin{aligned} x_1 &= 0 \quad ; \quad y_1 = u\\ x_2 &= 1(unit) \quad ; \quad y_2 = v\\ x_3 &= 1/2(unit) \quad ; \quad y_3 = w/2 \end{aligned} \tag{14}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราสร้าง nomograph โดยการเริ่มเขียน "scale support" 3 ตัวสำหรับค่า u , v และ w ที่

$$x_1 = 0$$
$$x_3 = 1/2 \qquad (15)$$
$$x_2 = 1$$

หลังจากนั้นเขียน "functional scale" สำหรับ u , v และ w functional scale คือ scale ที่มีเครื่องหมายบอกระดับแสดงไว้พร้อมกับค่าตัวแปรของ u , v หรือ w และระดับเหล่านี้ถูกแสดงในอัตราส่วนซึ่งตรงกันกับค่าของ "ฟังก์ชันของตัวแปร" f(u) , f(w) หรือ f(v) ระยะทางถูกกำหนดออกจากจุดเริ่มต้นของ scale ซึ่งไม่จำเป็นต้องเท่ากับศูนย์ ส่วนใหญ่เราจะใช้ตารางในการกำหนดค่าของ functional scale แม้จะไม่จำเป็นในตัวอย่างนี้ก็ตาม เมื่อกำหนดให้ u และ v แปรค่าจาก 0 ถึง 5 (cm) และ w แปรค่าจาก 0 ถึง 10 (cm) จากนั้นนำค่า f(u) และ f(v) ไปใส่ในตารางที่ 1(a) และ f(w) ไปใส่ในตารางที่ 1(b) functional scale จะนำไปเขียนได้ดังรูปที่ 4 ซึ่งเป็นรูปที่สมบูรณ์ของ nomograph สำหรับ u + v = w



รูปที่ 4 รูป nomograph สำหรับสมการ u + v = w

เมื่อค่า functional scale ของ f(u) , f(v) หรือ f(w) ไม่ได้ตั้งอยู่ในช่วงที่เหมาะสมสำหรับการ plot ค่า เราสามารถแก้ปัญหาได้โดยการนำ "scale modulus" มาใช้ โดยที่ scale modulus (m) คือตัวคูณที่สามารถทำให้เราจัดค่าของฟังก์ชันนั้นให้อยู่ในช่วงที่กำหนดได้ ในทางคณิตศาสตร์แล้ว m จะมีค่าเท่ากับ

$$m_u = \frac{x}{f(u)_H - f(u)_L} \qquad (16)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ x คือ ความยาว , f_H และ f_L คือ ค่าของที่จุดสูงสุดและต่ำสุดของ u และ (f_H - f_L) คือ ช่วงของฟังก์ชันนั้นๆสมการนี้เป็นสมการหนึ่งที่มีประโยชน์ในการสร้าง nomograph ยกตัวอย่าง เช่น พิจารณาจากสมการ u + v = w อีกครั้ง แต่ในครั้งนี้เราสมมติให้ u แปรค่าจาก 2 ถึง 10 และ v แปรค่าจาก 5 ถึง 15 ถ้าความยาว scale ของ u และ v มีค่าเป็น 6 cm ดังนั้นเราจะได้ ค่า scale modulus เป็น

$$m_u = \frac{6}{10-2} = \frac{3}{4} \tag{17}$$

และ

$$m_v = \frac{6}{15-5} = \frac{3}{5}$$

(a)

u	0	1	2	3	4	5
$f(u) = u$	0	1	2	3	4	5
$y = u$ (cm)	0	1	2	3	4	5

(b)

w	0	2	4	6	8	10
$f(w) = w$	0	2	4	6	8	10
$y = w/2$ (cm)	0	1	2	3	4	5

ตารางที่ 1 แสดง functional scale สำหรับรูปที่ 4

จากนั้นมาดูผลที่เกิดขึ้นของ scale modulus เหล่านี้บน nomograph ของรูปที่ 4

- ทำการจัด scale u และ v ด้วย scale modulus m_u และ m_v ใน (9) จะได้

$$\begin{aligned} x - m_u u &= 0 \\ y - m_v v &= 0 \\ x/m_u + y/m_v - w &= 0 \end{aligned} \tag{18}$$

- เขียนให้อยู่ในรูป det ได้เป็น

$$\begin{vmatrix} 1 & 0 & -m_u u \\ 0 & 1 & -m_v v \\ 1/m_u & 1/m_v & -w \end{vmatrix} = 0 \tag{19}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- det นี้สามารถจัดให้อยู่ในรูปของ basic det ได้ดังนี้

$$\begin{vmatrix} 0 & m_u u & 1 \\ 1 & m_v v & 1 \\ m_u/(m_u+m_v) & m_u m_v w/(m_u+m_v) & 1 \end{vmatrix} = 0 \qquad (20)$$

- เมื่อแทนค่า $m_u = 3/4$ และ $m_v = 3/5$ สมการที่ (20) จะเท่ากับ (21)

$$\begin{vmatrix} 0 & 3u/4 & 1 \\ 1 & 3v/5 & 1 \\ 5/9 & w/3 & 1 \end{vmatrix} = 0 \qquad (21)$$

เพื่อจะเขียน nomograph เราจะปรับค่าฟังก์ชันของ u , v และ w จนกระทั่งค่าแรกของฟังก์ชันมีค่าเท่ากับศูนย์ตรงกัน โดยปกติจะใช้การแปรรูปอย่างง่าย ๆ ของ u , v และ w เมื่อแปรรูป u , v และ w ใน (21) แล้ว จะได้

$$\begin{vmatrix} 0 & [3(u-2)/4] & 1 \\ 1 & [3(v-5/5] & 1 \\ 5/9 & [(w-7)/3] & 1 \end{vmatrix} = 0 \qquad (22)$$

ตอนนี้เราก็สามารถสร้างตาราง functional scale ของ nomograph ดังแสดงในตารางที่ 2 จากตารางที่ 2 และค่าพิกัด x ใน (22) จะทำให้สามารถสร้าง nomograph ได้ดังรูปที่ 5 โดยค่า basic det ใน (20) นั้นจะเป็น general det ของสมการ $u + v = w$ ซึ่งสามารถจะลดรูปให้อยู่ใน (13) ได้ โดยให้ $m_u = m_v = 1$

u	2	4	6	8	10
$f(u) = u-2$	0	2	4	6	8
$y_1 = 3(u-2)/4$	0	3/2	3	9/2	6

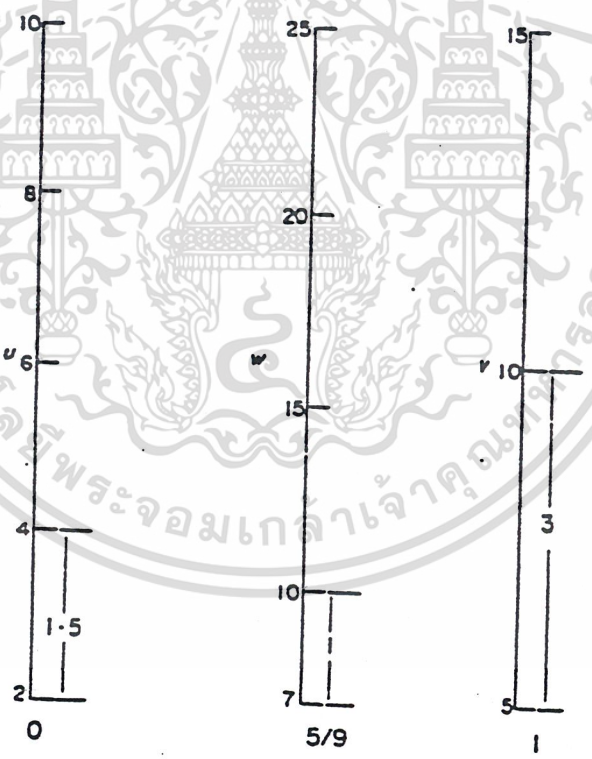
v	5	10	15
$f(v) = v-5$	0	5	10
$y_2 = 3(v-5)/5$	0	3	6

ตารางที่ 2 แสดง functional scale สำหรับ รูปที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

w	7	10	15	20	25
$f(w) = w-7$	0	3	8	13	18
$y_3 = (w-7)/3$	0	1	8/3	13/3	6

ตารางที่ 2 แสดง functional scale สำหรับรูปที่ 5 (ต่อ)



รูปที่ 5 แสดง nomograph ของสมการ $u + v = w$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

การประยุกต์ Nomograph มาใช้กับ Filter

Transfer function H(s) ของ LPF ที่ order n มีค่าเท่ากับ

$$H_n(s) = \frac{1}{M_n(s)} = \frac{N_m(s)}{D_n(s)} \qquad ;m < n \qquad (23)$$

เมื่อกำหนด M เป็นส่วนกลับของ Transfer function และ N กับ D เป็น polynomails ในเทอมของ S Classical filter มีค่า magnitude responses ยกกำลัง 2 เป็น

$$|H(j\omega)|^2 = \frac{1}{|M(j\omega)|^2} = \frac{1}{1+\varepsilon^2 S_n^2(\omega)} \qquad (24)$$

เมื่อ $S_n(\omega)$ สัมพันธ์กับ classical polynomail ของ order ที่ n และ ε คือค่า passband ripple factor ที่ transfer function ถูก normalized ทางความถี่ ซึ่งเราจะกล่าวถึงในลำดับต่อไป ค่า maximum inband gain นั้นจะถูก normalized ไปเป็นที่ " 1 " หรือมีค่าเท่ากับ 0 dB magnitude responses ที่เห็นกันทั่ว ๆ ไปแสดงในรูปที่ 1(a) nomograph ของรูป 1(b) เชื่อมโยง order ที่ n ของ filter , ค่า passband M_p , ค่า stopband M_s และค่า normalized frequency Ω

เราจัดรูป (24) ใหม่ได้เป็น

$$|M(j\omega)|^2 = 1+\varepsilon^2 S_n^2(\omega) \qquad (25)$$

โดยความถี่ ω ที่ถูก normalized เป็น

$$\Omega \equiv \frac{\omega}{\omega_p} \qquad (26)$$

ดังรูปที่ 1(a) จะมี

$$M_p \equiv |M(j1)| \qquad (27a)$$

$$M_s \equiv |M(j\Omega)| \qquad (27b)$$

แทน (27a) ลงใน (25) จะได้

$$M_p^2 = 1+\varepsilon^2 S_n^2(1) \equiv 1+\varepsilon^2 \qquad (28)$$

ดังนั้นเงื่อนไขของการ normalized ทางความถี่ ก็คือ

$$S_n^2(1) = 1 \qquad (29)$$

เพื่อให้ ripple parameter เท่ากับ

$$\varepsilon^2 = (M_p^2 - 1) \qquad (30)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่ง (30) นี้ แสดงให้เห็นว่าเมื่อกำหนดค่า M_p ก็สามารถจะหาค่า ε ได้ และโดยการแทน (30) ลงใน (25) จะได้

$$|M(j\omega)|^2 = 1+(M_p^2-1)S_n^2(\omega) \tag{31}$$

เพื่อแทนค่าในเงื่อนไข M_s จาก (27b) ลงใน (31) เราก็จะได้สมการที่เชื่อมโยง M_p , M_s , Ω และ n เข้าด้วยกันดังนั้น

$$M_s^2 = 1+(M_p^2-1)S_n^2(\Omega) \tag{32}$$

ซึ่ง (32) นี้ คือสมการที่เราจะนำมาสร้าง nomograph ของเรา โดยจะต้องให้ M_p และ M_s เป็นค่าตัวเลขและไม่มีหน่วยเป็น dB

เราได้นำเสนอความคิดพื้นฐานในการสร้าง nomograph จากสมการที่มี 3 ตัวแปรไปแล้ว อย่างไรก็ตาม สมการนี้ประกอบไปด้วยตัวแปร 4 ตัว (M_p , M_s , Ω , n) นับว่าโชคดีที่หลังจากการศึกษาที่ผ่านมาปรากฏว่า สมการนี้นำเราไปสู่กรณีพิเศษที่มี nomograph ที่ประกอบด้วย 4 ตัวแปร nomograph เหล่านี้ถูกอ้างถึงว่า เป็น "collineation nomographs","combination nomgraphs" และ "grid nomographs" เราชอบที่จะจัดพวก มันว่าเป็น combination nomographs เพื่อให้ได้ nomograph นี้มา เราจะอธิบายขั้นตอนการปฏิบัติการของ Otto ดังต่อไปนี้

$$f_1(u)+f_2(v)=f_4(\omega,t) \tag{33}$$

ดังนั้นถ้าฟังก์ชัน $f_4(\omega, t)$ ไม่เป็นทั้ง ผลบวก และ ผลต่าง ของ 2 ฟังก์ชัน ซึ่งแต่ละฟังก์ชันจะขึ้นอยู่กับตัวแปรเพียงตัวเดียวเราสามารถแทนสมการนี้ด้วยสมการ 2 สมการดังนี้

$$f_1(u)+f_2(v)=\gamma \tag{34a}$$

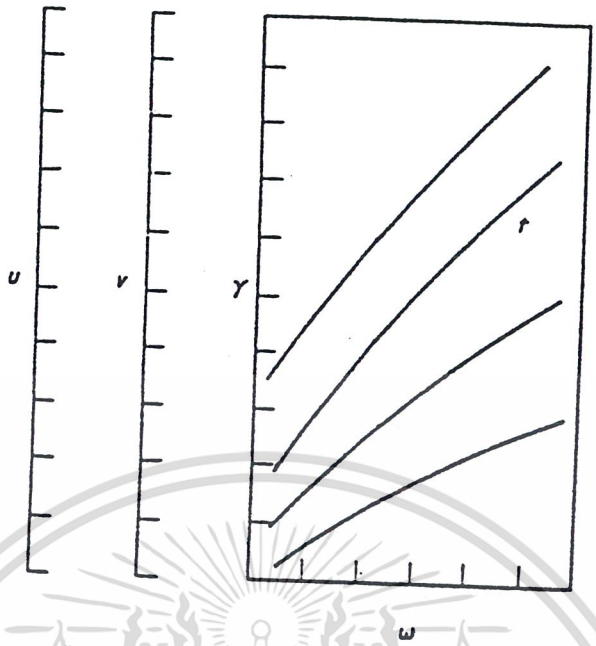
$$\gamma = f_4(\omega,t) \tag{34b}$$

สำหรับ (34b) นี้เราจะต้องเขียนกลุ่มของ curve ภายในระบบ Cartesian ซึ่งมีพิกัดของแกนเป็น (γ , t) หรือ(γ , ω) ในกรณีนี้เราจะเลือกใช้แกน γ และ ω ดังนั้นกลุ่มของ curve ในพิกัด (γ ,ω) จึงเป็นฟังก์ชันของ t เราจะรวมรูปนี้เข้ากับ nomograph ของสมการ (34a) ซึ่งประกอบไปด้วย 3 scale เมื่อ scale γ เหมือนกับ scale γ ของ (34b) ทุกประการ ผลลัพธ์ที่ได้นี้เป็นรูปแบบสุดท้ายของ nomograph สำหรับ (33) ดังแสดงในรูปที่ 6

ในตอนนี้ก็เหลือเพียงการแสดงสมการ (32) ให้อยู่ในรูปของ (33) อย่างไรก็ตามสมการที่ (33)จะเกี่ยวเนื่องกับผลที่ได้ เทคนิคง่ายๆซึ่งใช้กันอย่างแพร่หลายในการแยกตัวแปรใน nomograph คือการใส่ log (Logarithm) ลงใน (32) เราจัดรูป (32) ใหม่ได้เป็น

$$M_s^2-1=(M_p^2-1)S_n^2(\Omega) \tag{35}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6 แสดง Combination nomograph

โดยการ take Log จะได้

$$\log(M_s^2 - 1) = \log(M_p^2 - 1) + \log S_n^2(\Omega) \tag{36}$$

หรือเขียนได้เป็น

$$\log(M_s^2 - 1) - \log(M_p^2 - 1) = \log S_n^2(\Omega) \tag{37}$$

จะเห็นว่า ถ้าไม่คำนึงถึงเครื่องหมาย (37) นั้นจะอยู่ในรูปเดียวกับ (32) ดังนั้น nomograph จะเป็นไปตามรูปที่ 6 เมื่อทำตามขั้นตอนของ Otto เราจะสร้างสมการ 2 สมการ ดัง(34) และค่า γ แสดง (37) ใหม่ได้เป็น

$$\log(M_s^2 - 1) - \log(M_p^2 - 1) = \gamma \tag{38}$$

และ

$$\boxed{\gamma = \log S_n^2(\Omega)} \tag{39}$$

พิจารณา (38) และ (39) สังเกตุว่าทั้งสองสมการนี้จะเปลี่ยนแปลงก็ต่อเมื่อ Transfer function มีการเปลี่ยนแปลง ดังนั้นแสดงให้เห็นว่า nomograph จาก(38)สามารถใช้สำหรับ filter ทุกชนิดได้ ซึ่ง gain สามารถ เขียนในรูปของ (24) สมการที่ (39) จะถูกใช้เพื่อสร้างส่วนของกราฟ ของ combination nomograph ในรูปที่ 6 เมื่อ (38) ถูกสร้างขึ้นมา M_p และ M_s จะมีค่าเป็นตัวเลขและไม่มีหน่วยเป็น dB ดังที่เราต้องการ เราจะใช้ประโยชน์จากการแทนค่าดังต่อไปนี้ เพื่อเปลี่ยนพวกมันให้อยู่ในรูปของ dB เมื่อ

$$M_s(dB) = 20 \log M_s \tag{40}$$

หรือ

$$M_s = 10^{M_s(dB)/20} \tag{41}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการยกกำลัง 2 ทั้งสองข้าง จะได้

$$M_s^2 = 10^{M_s(dB)/10} \tag{42}$$

โดยวิธีเดียวกันนี้ จะได้ค่า M_p เป็น

$$M_p^2 = 10^{M_p(dB)/10} \tag{43}$$

นำ (42) และ (43) ไปแทนลงใน (38) จะได้

$$\boxed{\log(10^{M_s(dB)/10} - 1) - \log(10^{M_p(dB)/10} - 1) = \gamma} \tag{44}$$

ซึ่งตอนนี้เราสามารถที่จะใช้ (39) และ (44) เพื่อสร้าง combination nomograph สำหรับ filter ต่างๆ จากจุดนี้เป็นต้นไปเราจะใช้ M_p(dB) และ M_s(dB) และเพื่อความง่าย เราจะเขียนเพียงแค่ M_p และ M_s

พิจารณา (44) เราจะเห็นว่า เมื่อ M_s และ M_p มีค่ามากกว่า 10dB ก็สามารถที่จะตัดค่า "1" ทิ้งออก ไปจากสมการได้ ซึ่งจะทำให้สามารถเขียน (44) ใหม่ได้เป็น

$$\frac{M_s}{10} - \frac{M_p}{10} = \gamma \qquad ; M_p, M_s >> 10dB \tag{45}$$

สมการนี้มีค่าเป็นจริง ซึ่งแสดงได้โดยการคำนวนอย่างง่าย ๆ ดังข้างล่างนี้

$$\begin{aligned} \log(10^1) &= 1 & \log(10^1 - 1) &= 0.9542 \\ \log(10^2) &= 2 & \log(10^2 - 1) &= 1.9956 \\ \log(10^3) &= 3 & \log(10^3 - 1) &= 2.99956 \end{aligned} \tag{46}$$

จะเห็นว่าค่าผิดพลาดจะมีค่าน้อยกว่า 5 เปอร์เซนต์ สำหรับค่า $M \geq 10dB$ และจะน้อยกว่า 0.5 เปอร์เซนต์ สำหรับค่า $M \geq 20dB$ เราจะเห็นว่า (45) นั้นอยู่ในรูปเดียวกับ u + v = w และเป็นสมการเดียวกับตัวอย่าง เนื่องจากเรายังไม่ได้มีการใช้ scale modulus จาก (8) และ (20)(เมื่อ $m_u = m_v = 1$) ได้เป็น

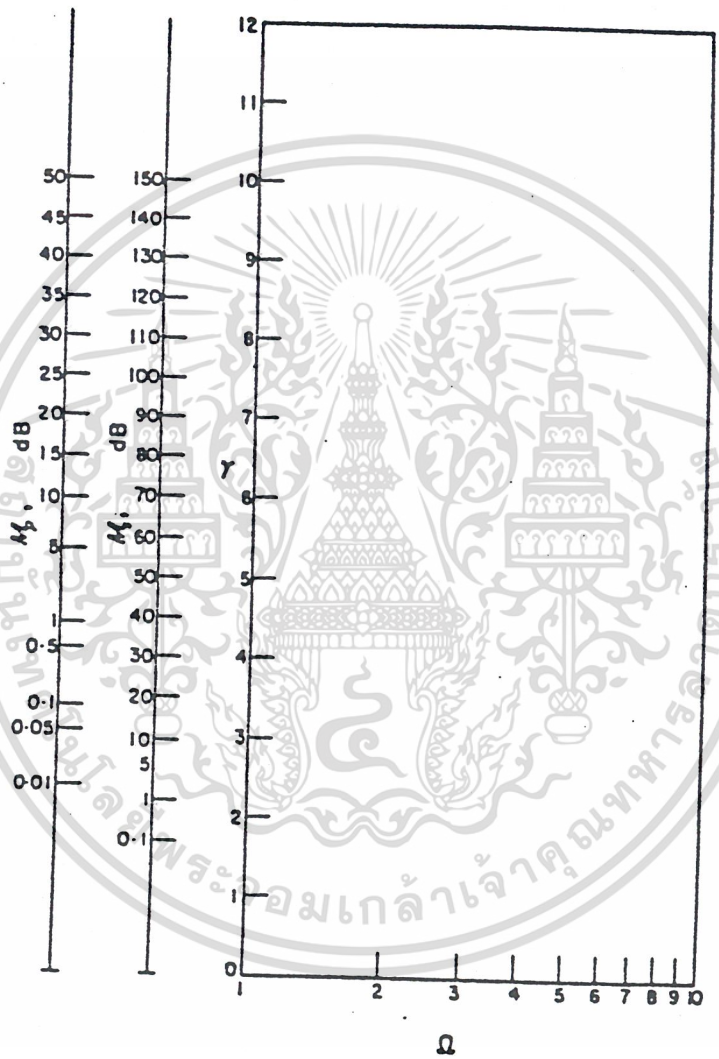
$$\begin{vmatrix} 0 & M_p/10 & 1 \\ 1 & \gamma & 1 \\ 1/2 & M_s/20 & 1 \end{vmatrix} = 0 \tag{47}$$

จาก basic det เราจะได้ว่า

$$\begin{aligned} x_1 &= 0 \qquad ; \quad y_1 = Mp/10 \\ x_2 &= 1(unit) \quad ; \quad y_2 = \gamma \\ x_3 &= 1/2(unit) \; ; \quad y_3 = M_s/20 \end{aligned} \tag{48}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้วยเหตุนี้ค่า scale support (คือค่าแกน M_p , M_s และ γ) สามารถเขียนได้ดังรูปที่ 7 และถ้าพิจารณาให้ดีจะพบว่าที่จริงแล้ว x_3 จะมีค่าเป็นครึ่งหนึ่งระหว่าง x_1 กับ x_2 เมื่อ x_2 คือค่าที่กำหนดขึ้นเอง ณ ที่นี้ค่า x_2 ถูกเลือกเพื่อให้เหมาะสมกับค่าแกน Ω ซึ่งจะกล่าวถึงในส่วนต่อไป เมื่อ scale support ถูกวางลงไปเราก็จะทำ scale มันได้



รูปที่ 7 รูป nomograph ที่มี scale เสร็จสมบูรณ์แล้ว

มันเป็นความจำเป็นที่จะต้องระบุช่วงสำหรับ M_p และ M_s หลังจากการพิจารณาอย่างรอบคอบจึงตกลงเลือกใช้ค่า M_p จาก 0.01 ถึง 50 dB และ M_s จาก 0.1 ถึง 150 dB จาก (45) ค่า γ จะตรงกับค่าสูงสุดของ M_p และ M_s คือ 10 โดยการใช้กระดาษกราฟ เราเลือก scale แกน γ จาก 0 ถึง 12 ซึ่งแต่ละจุดห่างกัน 2 cm เรายังได้กำหนดตำแหน่งของค่าสูงสุดของ M_p

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

= 50 dB และ M_s = 150 dB ให้อยู่บนเส้นแนวนอนเดียวกันกับค่า γ ที่ซึ่งค่า γ = 10 ส่วน scale ที่เหลือจะถูกวางตำแหน่งไปตามสัดส่วน และเพื่อให้เกิดการจัดวางให้อยู่ในแนวเดียวกัน (48) จึงกลายเป็น

$$\begin{aligned} x_1 &= 0 \quad ; \quad y_1 = M_p/10 + 5 \\ x_2 &= 1(unit) \quad ; \quad y_2 = \gamma \\ x_3 &= 1/2(unit) \quad ; \quad y_3 = M_s/20 + 2.5 \end{aligned} \tag{49}$$

เมื่อ y_1 และ y_3 มีค่าเปลี่ยนไป ค่า scale modulus ของ $M_p/10$, $M_s/20$ และ γ จะมีค่าเป็น 1 เท่ากันหมด ตราบใดก็ตามที่ M_p และ M_s มีค่ามากกว่า 20 dB ทั้ง 2 scale ก็จะเขียนได้ง่าย เพราะ (45) มีค่าเป็นจริง (ภายใต้ค่าผิดพลาด 0.5 เปอร์เซนต์) เนื่องจากค่า scale modulus ของ $M_s/20$ คือ 1 ดังนั้นการ scale จาก 20 ถึง 150 dB จะให้ห่างกัน 1 cm ระหว่างจุดหลัก (จำนวนนับคูณกับ 10dB) และด้วยวิธีเดียวกันนี้ เราจะเขียน scale ของ $M_p/10$ จาก 20 ถึง 50 dB โดยให้จุดหลักอยู่ห่างกัน 2 cm (จำนวนนับคูณกับ 10dB)

เพื่อทำการตรวจหลาย ๆ จุด สังเกตุว่าข้อมูลเหล่านั้นบน nomograph ของเราอยู่บนเส้นตรงแนวนอน (คือ $y_1 = y_2 = y_3$ ใน (49)) เมื่อใดก็ตามที่ค่า gain เป็นไปตาม

$$M_s = 2M_p + 50 \quad ; M_p > 20dB \tag{50}$$

ซึ่งในกรณีนี้

$$\gamma = M_p/10 + 5 = M_s/20 + 2.5 \quad ; M_p > 20dB, M_s > 90dB \tag{51}$$

สังเกตุด้วยว่า เมื่อใดก็ตามค่า gain มีค่าเท่ากัน เมื่อนั้น (44) ก็จะได้

$$\gamma = 0 \quad \text{ที่} \quad M_p = M_s \tag{52}$$

nomograph ใดๆก็ตามต้องเป็นไปตามเงื่อนไขนี้(ที่ M_p , M_s)ใดๆซึ่งเป็นจุด check point ใดๆ ก็ได้

เพื่อที่จะได้ scale ของ M_p , M_s ตั้งแต่ 20 dB ลงมา เราจะย้อนกลับไปดู (44) ซึ่ง nomograph กำลังถูกสร้างขึ้น basic det ของ (44) คือ

$$\begin{vmatrix} 0 & \log(10^{M_p/10} - 1) & 1 \\ 1 & \gamma & 1 \\ 1/2 & \log(10^{M_s/10} - 1)/2 & 1 \end{vmatrix} = 0 \tag{53}$$

ซึ่งลดรูปลงมาเป็น (47) สำหรับ M_p , M_s > 20 dB ดังนี้ สมการ nomograph จะกลายเป็น

$$\begin{aligned} x_1 &= 0 \quad ; \quad y_1 = \log(10^{M_p/10} - 1) + 5 \\ x_2 &= 1(unit) \quad ; \quad y_2 = \gamma \\ x_3 &= 1/2(unit) \quad ; \quad y_3 = [\log(10^{M_s/10} - 1)]/2 + 2.5 \end{aligned} \tag{54}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

86860

โดยการเทียบกับ (49) ค่า (y_1 , y_2 , y_3) จาก (54) สามารถหาค่าได้โดยการใช้ โปรแกรมภาษา Fortran ค่าเหล่านี้ทำให้เราต่อเติม scale M_p และ M_s ได้อย่างสมบูรณ์ โดยพวกมันจะถูกใช้ที่ M_p และ M_s น้อยกว่า 40 dB สิ่งนี้ทำให้เราสามารถพิสูจน์สมมติฐานได้ว่าเป็นจริงที่นำไปสู่ (45)

มันเป็นความน่าสนใจในการสังเกตุพฤติกรรมของ y_1 และ y_2 ที่ M_p , M_s มีค่าน้อยกว่า 10 dB มาก ๆ โดยการใช้อนุกรม Maclaurin สำหรับ a^x เป็น

$$a^x = 1 + x \ln a + \frac{(x \ln a)^2}{2!} + \cdots\cdots \tag{55}$$

เมื่อ a = 10 และ x = M/10 ดังนั้นที่ x << 1

$$\log(a^x - 1) \cong \log(x \ln a) = \log x + \log(\ln a) \tag{56}$$

เนื่องจาก a = 10 และ log(ln10) = 0.362 ดังนั้นจาก (54)

$$\begin{aligned} y_1 &= \log(M_p/10) + 5.362 \\ y_2 &= \gamma \qquad\qquad ; M_p, M_s << 10dB \\ y_3 &= [\log(M_s/10)]/2 + 2.681 \end{aligned} \tag{57}$$

ดังนั้น y_1 และ y_2 จะแปรค่าเป็นแบบ Logarithm สำหรับ M_p และ M_s ที่น้อยกว่า 1 dB เนื่องจากค่า scale modulus สำหรับ log(M_p / 10) คือ "1" ใน scale ของ M_p ที่ต่ำกว่า 1dB จะเขียนเป็น 2 cm ระหว่าง 2 จุดหลัก [จำนวนนับคูณกับ log(M_p/10)]ในทำนองเดียวกันของ M_s ที่ต่ำกว่า 1dB ก็จะเขียนได้เป็น 1 cm ระหว่าง 2 จุดหลัก[จำนวนนับคูณกับ log(M_s / 10)]

เราได้สร้าง nomograph ขึ้นมาแล้วเท่าที่ scale M_p , M_s และ γ เกี่ยวข้อง เหลือแต่เพียงการเขียน scale ลงบนแกน γ support และการ plot curve ที่เป็นเอกภาพลงบน grid แสดงผล กลับไปดู (39) อีกครั้ง จาก

$$\gamma = \log S_n^2(\Omega) \tag{58}$$

จะสังเกตุเห็นว่า

$$S_n^2(\Omega) = \Omega^{2(n-m)} \qquad ; \Omega \to \infty \tag{59}$$

สำหรับทุก ๆ filter ที่เป็นไปได้ ที่มี gain ของ (23) ดังนั้น

$$\gamma = 2(n-m)\log\Omega \qquad ; \Omega \to \infty \tag{60}$$

ด้วยเหตุนี้เพื่อสร้าง linear curve เส้นตรงที่เป็นแบบ asymtotic ในระนาบ (γ , Ω) เราจึงเลือก Ω ในเป็นแกนที่มี scale เป็นแบบ Logarithm จากประสบการณ์ ค่า Ω สามารถเขียนให้อยู่ในช่วง 1 cycle ~log ก็เพียงพอแล้ว การสร้าง nomograph ดังรูปที่ 7 จึงเสร็จสมบูรณ์ลง จากนี้ต่อไปเราก็พร้อมที่จะพิจารณา filter ต่าง ๆ และ พร้อมที่จะ plot ค่า $S_n(\Omega)$ เหล่านั้นจาก (39)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

Ultraspherical filter

Ultraspherical filter จะมีค่า magnitude function ที่กำหนดโดย (24) เมื่อ

$$S_n^2(\omega) = [F_n^{\alpha}(\omega)]^2 \tag{61}$$

นำไปแทนใน (39) nomograph ก็จะถูกจัดขึ้นโดยการ plot

$$\gamma = log[F_n^{\alpha}(\omega)]^2 \tag{62}$$

จาก (58) จะหาค่าของ F_n^{α} เราจะใช้ recursion formula โดย

$$(2\alpha + n)F_n^{\alpha}(x) = x(2\alpha + 2n - 1)F_{n-1}^{\alpha}(x) - (n-1)F_{n-2}^{\alpha}(x) \quad ;n > 2 \tag{63}$$

และ

$$F_0^{\alpha}(x) = 1 \tag{64a}$$

$$F_1^{\alpha}(x) = x \tag{64b}$$

เนื่องจาก $F_n^{\alpha}(1) = 1$ สมการ(61) จึงเป็นไปตามเงื่อนไขของการ normalized ทางความถี่ของ (29) เราสามารถเขียนโปรแกรมคอมพิวเตอร์ภาษา Fortran เพื่อจะสร้าง Ultraspherical polynomail (ถึง order ที่ 20) ได้โดยการใช้ (63) กับ (64) และใช้ค่า γ ใน (62) nomograph จะถูกสร้างขึ้นสำหรับค่า α ที่แสดงไว้ในตาราางที่ 4 และแสดงให้เห็นในรูปที่ A1 ถึง A14 ในภาคผนวกโดยพวกมันถูกแบ่งแยกหมวดหมู่ตามชื่อและตามค่าของ α ซึ่งจะตามหลังอยู่ในวงเล็บ

Butterworth filter เป็น filter ที่มีลักษณะแบบ all - pole maximally -flat magnitude (MFM) ดังนั้นจึงไม่มีค่า passband ripple และจะมีค่า magnitude function เป็น

$$|H(j\omega)|^2 = \frac{1}{1 + \varepsilon^2 \omega^{2n}} \tag{65}$$

เมื่อ

$$S_n^2(\omega) = \omega^{2n} \tag{66}$$

เนื่องจาก $|S_n(1)| = 1$ ดังนั้นจาก (58)

$$\gamma = log\, \omega^{2n} = 2n \log \omega \tag{67}$$

สมการเหล่านี้ คือ เส้นตรงที่มีความชันเท่ากับ 2n ดังนั้น nomograph ของ Butterworth filter จะสามารถ plot ได้อย่างง่ายดาย ดังรูปที่ A1 เมื่อใช้ $\alpha = 10^9$ แทนลงใน (62) นั้นจะเห็นว่าผลลัพธ์ที่ออกมามีค่าเหมือนกัน ซึ่งเคยถูกใช้พิสูจน์โปรแกรมของ Ultraspherical filter

modified associated Legendre (MAL) filter จะเป็นผลลัพธ์สำหรับค่าของ α ที่เป็นจำนวนเต็มบวก โดยพวกมันจะใช้ modified associated Legendre polynomails $P_n^{(m)}$ ซึ่งได้มาจากการอนุพันธ์ของ Legendre polynomail ที่มีลำดับที่ (n + m) เป็นจำนวน m ครั้ง เมื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

$$P_n^{(m)}(x) = \frac{d^m}{dx^m} P_{(n+m)}(x) \qquad (68)$$

ถ้าไม่รวมถึง elliptic filter แล้ว มันก็จะเป็น filter แรกที่มี 3 ตัวแปร ที่ซึ่ง (n , ε , α) ถูกเลือกออกไป M.A.L. filter ที่ 10^{th} , 3^{rd} และ 2^{nd} มี nomograph แสดงในรูป A2 , A3 และ A4 ตามลำดับ

Ultraspherical filter จะเป็นผลลัพธ์สำหรับค่าที่ไม่เป็นจำนวนเต็มของค่า $\alpha > -1$ ซึ่งสำหรับ 1.5 Ultraspherical filter จะมี nomograph ดังรูปที่ A5 และ 1^{st} M.A.L. filter จะมี nomograph ดังรูปที่ A6

ที่ $\alpha = 0.5$ จะมีผลลัพธ์เป็น Chebyshev filter ชนิดที่ 2 ซึ่งชนิดนี้ไม่ใช่แบบ equal - riple ทั่ว ๆ ไปหรือ Chebyshev (ชนิดที่ 1) เมื่อ $\alpha = -0.5$ จะเห็นว่าพวกมันมีความเหมือนกันน้อยมาก ยกเว้นค่า α ที่มีเครื่องหมายต่างกันเท่านั้น พวกมันจะใช้ Chebyshev polynomails ชนิดที่ 2 $U_n(\omega)$ ซึ่ง nomograph จะแสดงได้ดังรูปที่ A7

Legendre filter มีค่า $\alpha = 0$ filter เหล่านี้ใช้ Legendre polynomails $P_n(\omega)$ โชคไม่ดีที่คนมักจะสับสน filter นี้กับ Papoulis หรือ monotonic-L filter ซึ่งจะพิจารณาต่อไป โดยรูปที่ A8 ได้แสดง nomograph ของ Legendre filter

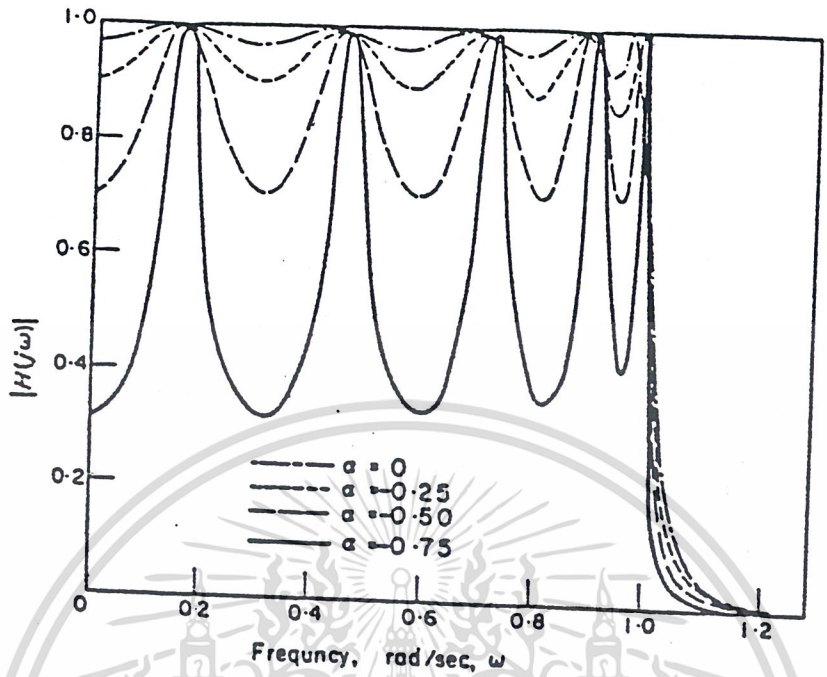
รูป A8 แสดง nomograph ของ -0.25 Ultraspherical filter ซึ่ง filter นี้อธิบายไว้โดย Petrela และ Budak

Chebyshev filter (ชนิดที่ 1) หรือ equal - ripple จะมีค่า $\alpha = -0.5$ มันจะใช้ Chebyshev polynomails ของชนิดที่ 1 $T_n(\omega)$ โดยจะมี nomograph ดังรูป A10

สำหรับ - 0.75 ,- 0.90 ,- 0.95 และ - 0.9999 Ultraspherical filter จะแสดง nomograph ได้ดังรูป A11 ถึง A14 ซึ่ง filter เหล่านี้จะไม่สามารถใช้การได้จริง เพราะว่ามันจะมีค่า ripple ที่สูงมาก ซึ่งจะพิจารณาได้ดังต่อไปนี้

Ultraspherical filter ในส่วนนี้ ($\alpha = \infty$ ถึง -1) ถูกลำดับไว้ในรปแบบของ bandedge selectivity ที่เพิ่มขึ้นคือ $-dH / d\omega|_{\omega=1}$ สำหรับค่า magnitude responses ต่าง ๆ นั้นถูกแสดงไว้ในรูปที่ 8 เมื่อ $n = 10$ และ $\varepsilon = 1$ โดย filter จะมีค่าความถี่ที่ 3 dB เป็นหนึ่ง เนื่องจาก $\varepsilon = 1$ เมื่อ $\alpha > -0.5$ ค่า inband ripple จะเพิ่มขึ้นเมื่อเข้าใกล้ค่า passband edge สำหรับค่า $-1 < \alpha < -0.5$ ค่า ripple จะลดลง การลด α ลง จะเป็นการเพิ่มค่า ripple ยกเว้นแต่ $\varepsilon < 1$ ค่า ripple นี้จะมากเกินไปใน passband เราจะพบว่าตัวแปรที่ 3 ซึ่งก็คือ α สามารถจะใช้เพื่อให้ได้รับความคล่องตัวในการออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8 แสดง magnitude resposes ของ Ultraspherical filter ที่ order 10^{th} โดยมีค่า $\varepsilon = 1$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
โครงงาน

4.1 แนวทางในการออกแบบโปรแกรม

หลังจากการศึกษา JOURNAL ของโนโมกราฟอย่างละเอียดทางคณะผู้จัดทำโครงงานจึงได้ทำการศึกษาโปรแกรมคอมพิวเตอร์ของแต่ละภาษาว่ามีข้อดีและข้อเสียต่างกันอย่างไร จากนั้นจึงเลือกที่จะนำเอาภาษาซี(TURBO C) มาใช้ในการเขียนโปรแกรม เนื่องจากการนำเอาภาษาซีมาใช้ในการออกแบบนั้นจะสามารถเข้าถึงข้อมูลได้ดีกว่าภาษาอื่นเพราะเป็นภาษาระดับกลาง อีกทั้งหนังสือที่ใช้ประกอบในการค้นคว้าก็สามารถที่จะหาอ่านได้ง่าย เพราะเป็นภาษาที่มีการนำไปใช้งานกันอย่างแพร่หลาย โดยเฉพาะอย่างยิ่งสามารถที่จะทำการคำนวนชั้นสูงได้ดีพอสมควรเพียงแต่เราต้องมีการเข้าใจฟังกชันทางคณิตศาสตร์ที่มีในภาษาซีนั้น ซึ่งสามารถนำไปพัฒนาโปรแกรมเพื่อใช้งานในด้านต่าง ๆ อย่างมีประสิทธิภาพ ในอนาคต ภาษาซี ก็คงเป็นภาษาหลักของโปรแกรมคอมพิวเตอร์ทางวิทยาศาสตร์และวิศวกรรม

ในงานด้านวิศวกรรมศาสตร์ นอกจากการคำนวณแล้ว ผู้ใช้อาจจะต้องสร้างฮาร์ดแวร์เพื่อเชื่อมต่อเข้ากับระบบคอมพิวเตอร์ อาจจะต้องเขียนโปรแกรมเพื่อสั่งให้ระบบทำงานได้ตามที่ต้องการ ภาษาที่เลือกใช้ในการทำงานในลักษณะนี้จึงต้องสามารถเข้าถึงระบบได้อย่างมีประสิทธิภาพมีความมาตรฐานเพียงพอ

ภาษาซี นับเป็นภาษาหนึ่งที่อาจเรียกได้ว่าเป็นภาษามาตรฐานสำหรับการเขียนโปรแกรมบนระบบคอมพิวเตอร์ เราจะเห็นคอมไพเลอร์สำหรับภาษาซีตั้งแต่บนเครื่องไมโครคอมพิวเตอร์ไปจนถึงเมนเฟรม แม้ในการเขียนโปรแกรมที่เข้าถึงตัวระบบภาษาซีก็มีความอ่อนตัวถึงระดับที่ใกล้เคียงกับภาษาแอสเซมบลี อันเป็นภาษาเฉพาะบนระบบคอมพิวเตอร์แต่ละระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทำงานของโปรแกรม

การทำงานของโปรแกรม Filter จะทำงานร่วมกับ File อื่นๆ คือ

FILTER.C	จะทำหน้าที่หลักในการที่จะควบคุมการทำงานของฟังก์ชันต่างๆ
IN_PARAMS.C	จะทำหน้าที่หน้าที่ในการรับค่าพารามิเตอร์
F_SELECT.C	จะทำหน้าที่ในการแสดงชนิดของฟิลเตอร์
F_SUM.C	จะทำหน้าที่ในการแสดงผลรวมของ Filter ที่ได้จากการออกแบบ
F_SCREEN.C	จะทำหน้าแสดงผลของ Filter ที่ได้โดยใช้ Nomograph
UTILITY.C	จะทำหน้าที่ในการรับผลของแต่ละ Function และส่งไปให้ Function อื่นในการคำนวณต่อไป

โดยโมดูล(module)(นามสกุล .C)จะบรรจุการทำงานของฟังก์ชันโดยฟังก์ชันในภาษาซีก็คือโปรแกรมย่อย(Subprograms)โดยฟังก์ชัน หรือ โปรแกรมย่อยในการทำงานในโปรแกรมที่ออกแบบนี้เราจะให้แต่ละฟังก์ชันทำงานในจุดประสงค์ใดจุดประสงค์หนึ่ง โดยเมื่อในแต่ละฟังก์ชันทำงานเสร็จก็จะส่งข้อมูลให้อีกฟังก์ชันหนึ่งทำงานต่อไป ดังนั้นการเรียกฟังก์ชันในโปรแกรมก็คือการเรียกใช้โปรแกรมย่อยมาทำงาน

ฟังก์ชันที่ใช้ทำงานทั้งหมดในโปรแกรม

```
1) void  Title1(void);
2) void  Title2(void);
3) void  Text_Windows(int x1,int y1,int x2,int y2);
4)void  Display_Header(int Page);
5)void Summary_Specify(int number_F,float Mag_Pass,float Mag_Stop,
        float Fp,float Fs,float Alpha,int Order);
6)void Show_Type_Filter(int number_F,float Mag_Pass,float Mag_Stop,float
7)Fp,float Fs,float Alpha);
8)float Design_Alpha();
9)float Select_Filter(float *Alpha);
10)void Nomograph(int *point_x,int *point_y,float Max_Pass,float
        Min_Stop,float Freq_Pass,float Freq_Stop,float Alpha);
11)int convert_y(int y);
12)float Abs(float x);
13)int Find_Order(int point_x,int point_y,int *Order_Ture);
14)void Compare(float *alpha,int *Order_Ture);
15)float Input_Params_Filter(float *Max_Pass,float *Min_Stop,float
*Freq_Pass,float *Freq_Stop);
```

Listing4.1 ฟังก์ชันย่อยที่ใช้ในโปรแกรมทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชันทั้งหมดจะแยกกันอยู่ในโมดูล ที่มีนามสกุล .C โดยลักษณะการทำงานเดียวกันจะถูกจัดให้อยู่ในโมดูลเดียวกัน โดยการทำงานของโปรแกรมทั้งหมดจะได้อธิบายรายละเอียดต่อไป

main() Function

จะเป็นส่วนที่สำคัญมากในภาษาซีเพราะในภาษาซีจะประกอบไปด้วยหลายฟังก์ชันโดยในแต่ละฟังก์ชันสามารถที่จะเรียกใช้กันได้ และจะมีฟังก์ชันที่เป็นทางเข้าออกของโปรแกรม และในฟังก์ชัน main ของ Filter จะมีการเรียกใช้ฟังก์ชันอื่นเข้ามาทำงานดังนั้นจะต้องมีการเรียกใช้โมดูลอื่นเข้ามาทำงานด้วย โดยใช้ไดเรกทีฟ(#include Directive)เพื่อแจ้งให้คอมไพเลอร์ว่าให้อ่านไฟล์หรือโมดูลอื่นเข้ามาแปลร่วมด้วย

```
#include <conio.h>
#include <stdio.h>
#include <math.h>
#include <in_param.c>
#include <utility.c>
```

โดยเราจะมีการเรียกโมดูลมาตรฐานในภาษาซีเข้ามาแปลร่วมด้วย(โมดูลมาตราฐานคือโมดูลที่มีมาอยู่แล้วในภาษาซี)

```
/*************************************************
*          ----------------------------          *
* ------- Copyright (c) 1997 by Som&Oat&Sing------- *
*          ----------------------------          *
*************************************************/
#include <conio.h>
#include <stdio.h>
#include <math.h>
#include <in_param.c>
#include <utility.c>

void  Text_Windows(int x1,int y1,int x2,int y2);
void  Display_Header(int Page);
void  Title1(void);
void  Title2(void);

/*************************************************
  Function MAIN()
*************************************************/
void main(void)
{
    int Order,n,Array_Order[15];
    float Mp,Ms,Fp,Fs,Alpha,Array_Param[4];
    float Array_Alpha[15]={1000000000,10,3,2,1.5,1,0.5,0,-0.25,-0.5,
-0.75,-0.9,-0.95,-0.9999};
    char ans_cont,ans_compare;
    char Name_Filter[15][40]={"Butterworth",
                              "10th M.A.L.",
                              "3rd  M.A.L.",
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                "2nd  M.A.L.",
                                "1.5  Ultrasphérical",
                                "1st  M.A.L.",
                                "2nd  Cheybyshev",
                                "Legendre",
                                "-0.25 Ultrasphérical",
                                "1st  Cheybyshev(Equal-Riple)",
                                "-0.75 Ultrasphérical",
                                "-0.9  Ultrasphérical",
                                "-0.95 Ultrasphérical",
                                "-0.9999 Ultrasphérical"};
      FILE *fin;
      clrscr();
      Title1();
      getch();
      Display_Header(1);
      Title2();
      getch();
      do{
      Input_Params_Filter(&Mp,&Ms,&Fp,&Fs);
      Utility(Mp,Ms,Fp,Fs);
      textcolor(LIGHTGRAY);
      printf("You want to compare another filter(yes/no) :");
      ans_compare=getch();
    if(ans_compare=='y')
    {
      printf("yes\n");
      fin=fopen("params.dat","r");
      for(n=0;n<4;n++)
      fscanf(fin,"%f",&Array_Param[n]);
      fclose(fin);
      for(n=0;n<14;n++)
          {
          Alpha=Array_Alpha[n];
          Compare(&Alpha,&Order);
          Array_Order[n]=Order;
          }
          clrscr();
          Display_Header(5);
          printf("\n\n");
          printf("     Maximum pass band attenuation(dB) :
%.4f\n",Array_Param[0]);
          printf("     Minimum pass band Attenuation(dB) :
%.4f\n",Array_Param[1]);
          printf("     Frequency pass band          (Hz) : %.4f\n",Array_Param[2]);
          printf("     Frequency stop band          (Hz) : %.4f\n",Array_Param[3]);
          fclose(fin);
    for(n=0;n<14;n++)
          printf("%30s  Alpha = %-15.4f
Order=%d\n",Name_Filter[n],Array_Alpha[n],Array_Order[n]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
    printf("\nPress Y or y to continue if not press any key to EXIT : ");
     ans_cont=getch();
     if(ans_cont=='y'||ans_cont=='Y')printf("yes");
     switch(ans_cont)
      {
       case 'y':break;
       case 'Y':break;
       default :{
                  printf("no");
                  printf("\nThank you choose NOMOGRAPH to design filter\n");
                  exit(0);
                 }
      }
        }while(1);
}
```

Listing4.2 FILTER

ฟังก์ชัน void Title1(void); ทำหน้าที่ในการแสดงไตเติลขณะเริ่มโปรแกรม และจะมีฟังก์ชัน void Title2(void); ทำหน้าที่เป็น ไตเติลอีกครั้งเพื่อเป็นการแนะนำประโยชน์ของโปรแกรม และ วิธีการใช้โปรแกรม

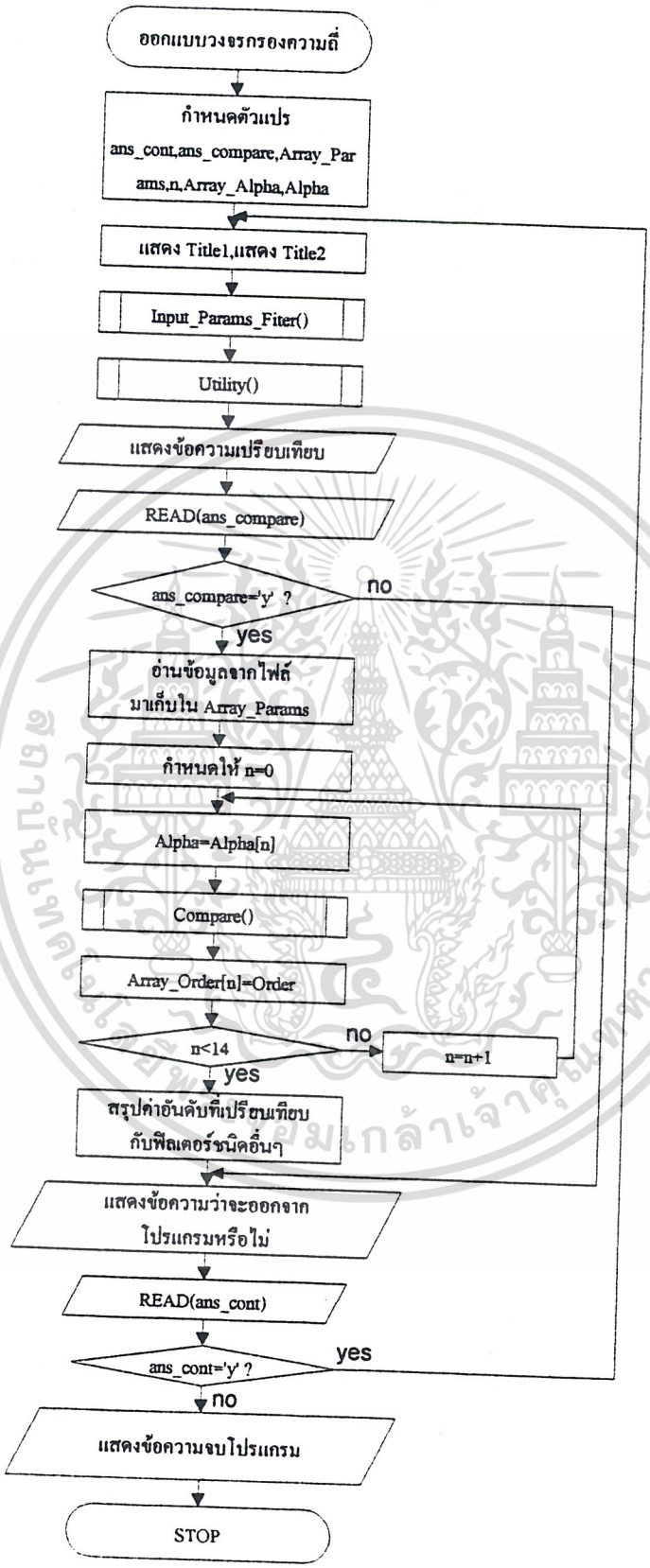
ในส่วนการทำงานของสเตทเมนท์ do-while Statement จะเป็นลักษณะการทำงานของการวนลูปโดยจะทำงานตามคำสั่งนั้นหรือกลุ่มคำสั่งก่อนตรวจสอบค่าในการวนลูปแล้วจนกว่าค่าที่ได้จากการตรวจสอบจะเป็นจริง ซึ่งเราจะกำหนดให้มันเป็นจริงเสมอโดยกำหนดให้ค่าที่ใช้ในการตรวจสอบเป็น “จริง” (ในภาษาซีจะให้ความหมายจริงคือ 1)

```
do{  statement } while(1);
```

ดังนั้นจะทำให้โปรแกรมทำการวนลูปไม่หยุด แต่เราสามารถที่จะออกจากโปรแกรมได้โดยใช้คำสั่ง exit(); โดยจะมีผลให้ออกจากโปรแกรมอย่างถาวร โดยในโปรแกรมนี้เราจะใช้ในการให้โปรแกรมทำงานในลักษณะที่เมื่อผู้ใช้ยังไม่ต้องการออกจากโปรแกรม และถ้าผู้ใช้ต้องการออกจากโปรแกรม ก็ให้เลือกใช้ฟังก์ชัน exit();

```
do{
    Input_Params_Filter(&Mp,&Ms,&Fp,&Fs);
    Utility(Mp,Ms,Fp,Fs);
    textcolor(LIGHTGRAY);
    printf("You want to compare another filter(yes/no) :");
   ans_compare=getch();   /* reciev value keyboard */
  if(ans_compare=='y')      /* check value keyboard */
  {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 9 ไฟลวชาร์ตการทำงานของฟังก์ชัน main()

```
        printf("yes\n");
        fin=fopen("params.dat","r");    /* read data file form params.dat */
        for(n=0;n<4;n++)
        fscanf(fin,"%f",&Array_Param[n]);
        fclose(fin);                    /* close file params.dat */
        for(n=0;n<14;n++)
            {
            Alpha=Array_Alpha[n];
            Compare(&Alpha,&Order);     /* call function Compare() */
            Array_Order[n]=Order;
            }
            clrscr();
            Display_Header(5);
            printf("\n\n");
            printf("    Maximum pass band attenuation(dB) :
%.4f\n",Array_Param[0]);
            printf("    Minimum pass band Attenuation(dB) :
%.4f\n",Array_Param[1]);
            printf("    Frequency pass band        (Hz) : %.4f\n",Array_Param[2]);
            printf("    Frequency stop band        (Hz) : %.4f\n",Array_Param[3]);
            fclose(fin);
        for(n=0;n<14;n++)
            printf("%30s   Alpha = %-15.4f
Order=%d\n",Name_Filter[n],Array_Alpha[n],Array_Order[n]);
      }
      printf("\nPress Y or y to continue if not press any key to EXIT : ");
        ans_cont=getch();
        if(ans_cont=='y'||ans_cont=='Y')printf("yes");
        switch(ans_cont)   /* check keyboard */
        {
         case 'y':break;  /* press 'y' to continue */
         case 'Y':break;  /* press 'Y' to continue */
         default :{
                    printf("no");
                    printf("\nThank you choose NOMOGRAPH to design filter\n");
                    exit(0);        /* close program */
                  }
        }
          }while(1);  /*infinity Loop */
```

Listing 4.3 ลูป do-whileในฟังก์ชัน MAIN()

Title1() Function

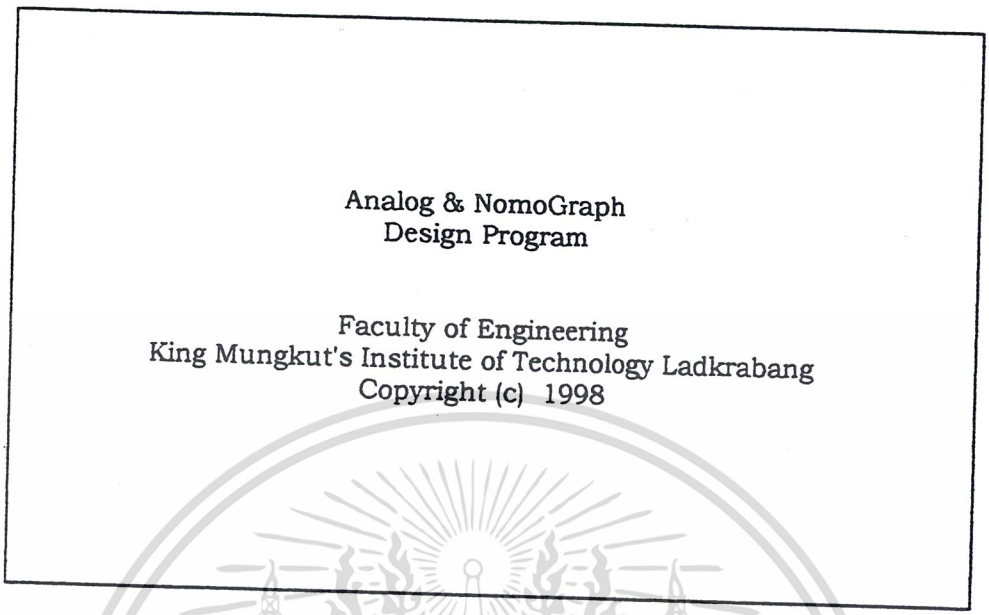
ฟังก์ชัน Title1() จะทำหน้าที่แสดงไตเติลขณะที่เริ่มโปรแกรม โดยจะมีการเรียกฟังก์ชัน Text_Windows () เพื่อวาดกรอบโดยจะมีการนำรหัสแอสกีทำเป็นเส้นกรอบ และใช้ฟังก์ชัน cprintf() เพื่อแสดงข้อความออกทางหน้าจอ โดยทำให้ตัวรหัสแอสกีที่ปรากฏที่หน้าจอไม่เลื่อน

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void Title1(void)
{
    Text_Windows(1,1,79,24);
    textcolor(GREEN);
    gotoxy(29,7);
    cprintf("Analog & NomoGraph");
    gotoxy(31,8);
    cprintf("Design Program");
    gotoxy(27,13);
    cprintf("Faculty of Engineering");
    gotoxy(16,14);
    cprintf("King Mungkut's Institute of Technology Ladkrabang");
    gotoxy(28,15);
    cprintf("Copyright (c)  1998");
    gotoxy(79,24);
    textcolor(LIGHTGRAY);
}
/*=====================================================
  Text_Windows() - draw windows for title
  Prototype:  void Text_Windows();
  Return:     none
=====================================================*/
void Text_Windows(int x1,int y1,int x2,int y2)
{
 int x,y;
 clrscr();
for(y=y1+1;y<y2;y++)
{
  for(x=x1+1;x<x2;x++)
  {
    gotoxy(x,y);
   putch(0x20);
  }
  gotoxy(x1,y);putch(0xba);
  gotoxy(x2,y);putch(0xba);
}
gotoxy(x=x1+1,y1);
for(x=x1+1;x<x2;x++)putch(0xcd); /* Horizontal line */
gotoxy(x1+1,y2);
for(x=x1+1;x<x2;x++)putch(0xcd); /* Horizontal line */
gotoxy(x1,y1);putch(0xc9);       /* Corner line */
gotoxy(x2,y1);putch(0xbb);
gotoxy(x1,y2);putch(0xc8);
gotoxy(x2,y2);putch(0xbc);
}
```

Listing 4.4 ของฟังก์ชัน Title1()และ Text_Windows()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 10 ที่ได้จากฟังก์ชัน *Title()*

ส่วนของการกำหนดชื่อในฟังก์ชัน main ในลักษณะการกำหนดด้านล่างจะเป็นการกำหนดให้ ตัวแปรชื่อ Array_Alpha และ Name_Filter เป็นตัวแปรชุด(Array)และให้ Array_Alpha เป็นตัวแปรที่ใช้เก็บจำนวนจริง และกำหนดให้ Name_Filter เป็นตัวแปรที่ใช้เก็บตัวแปรชนิดตัวอักษร โดยเป็นลักษณะตัวแปรภายในกลุ่มคำสั่ง(Local Variables) โดยจะกำหนด ให้ Array_Alpha เป็นตัวแปรที่เก็บได้ 15 ชุดแต่กำหนดให้มีข้อมูล 14 ข้อมูล และ กำหนดให้ Name_Filter เป็นตัวแปรชุดที่เก็บข้อมูลได้ 15 ชุด โดยแต่ละชุดเก็บได้ ไม่เก็บ 40 ตัวอักษรโดยเราจะกำหนดเพื่อใช้ในการวนลูปในตอนสรุปชนิดของ Filter และ Order ที่ได้

```
float Array_Alpha[15]={1000000000,10,3,2,1.5,1,0.5,0,-0.25,-0.5,
                       -0.75,-0.9,-0.95,-0.9999};
char Name_Filter[15][40]={"Butterworth",
                          "10th M.A.L.",
                          "3rd  M.A.L.",
                          "2nd  M.A.L.",
                          "1.5  Ultrasphерical",
                          "1st  M.A.L.",
                          "2nd  Cheybyshev",
                          "Legendre",
                          "-0.25 Ultrasphерical",
                          "1st  Cheybyshev(Equal-Riple)",
                          "-0.75 Ultrasphерical",
                          "-0.9  Ultrasphерical",
                          "-0.95 Ultrasphерical",
                          "-0.9999 Ultrasphерical"};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Input_Params_Filter() Function

ในฟังก์ชัน main จะมีการใช้ฟังก์ชัน Input_Params_Filter โดยหน้าที่การทำงานของฟังก์ชันนี้จะทำหน้ารับค่า พารามิเตอร์ ที่ผู้ใช้ต้องการออกแบบ โดยฟังก์ชันนี้จะอยู่ใน โมดูล IN_PARAMS.C ดังนั้นจะต้องมีการสั่ง Include Directive เพื่อเป็นการบอกให้โปรแกรมทราบว่าฟังก์ชันนี้อยู่ใน โมดูลใดเพื่อ จะได้นำมาแปลร่วมด้วย

```
#include <in_param.c>
```

โดยการที่ฟังก์ชัน main เรียกใช้ฟังก์ชัน Input_Params_Filter เพื่อให้ฟังก์ชัน Input_Params_Filter เป็นส่วนรับค่านั้นเมื่อรับค่าแล้ว ฟังก์ชัน Input_Params_Filter จะต้องส่งค่าพารามิเตอร์ 4 ค่า คือ Mp,Ms,Fp,Fs

โดย ค่า Mp คือค่า Maximun Attenuation Pass Band

Ms คือค่า Minimun Attenuation Stop Band

Fp คือค่า Frequency Pass Band

Fs คือค่า Frequency Stop Band

แต่เนื่องจากข้อจำกัดของภาษาซีที่สามารถส่งค่าออกจากฟังก์ชันได้เพียงค่าเดียว โดยการส่งค่านั้นแบบเดิมจะใช้คำสั่ง return; ในการส่งค่ากลับมาที่จุดที่เรียกใช้ฟังก์ชัน

```
return[ค่าที่ส่งกลับ]
```

แต่เนื่องจากฟังก์ชัน main ต้องการให้ฟังก์ชัน Input_Params_Filter() ส่งค่ากลับมามากกว่า 1 ค่า ดังนั้นการเขียนโปรแกรมจึงต้องใช้วิธีการอื่นคือใช้ตัวชี้ (Pointers) โดยตัวชี้คือตัวแปรที่ใช้เก็บข้อมูลชนิดตำแหน่งหน่วยความจำที่ใช้ในการเข้าหาข้อมูลชนิดตัวชี้หรือ พอยน์เตอร์

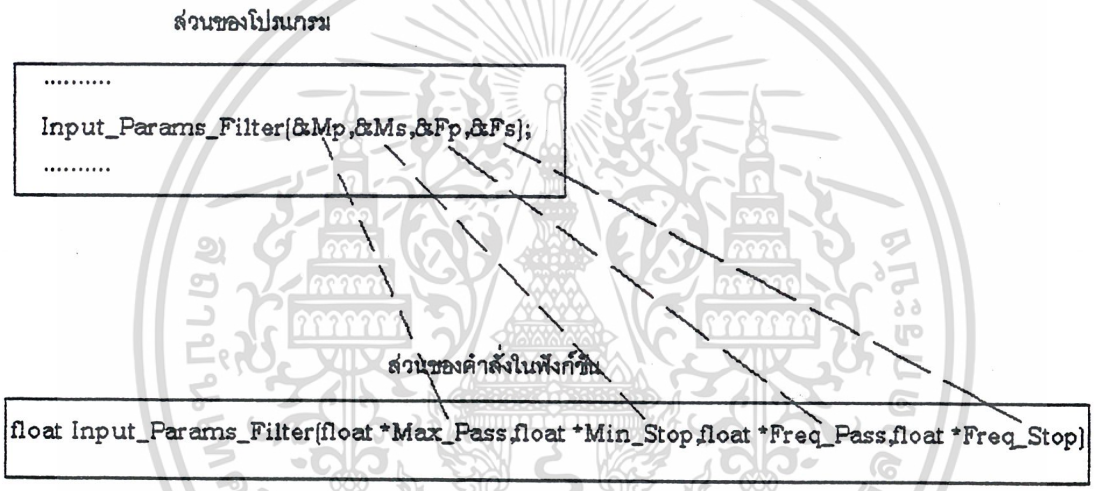
```
float Mp,Ms,Fp,Fs;
ส่วนของโปรแกรม
Input_Params_Filter(&Mp,&Ms,&Fp,&Fs);
```

โดยในฟังก์ชัน main มีการเรียกใช้ฟังก์ชัน Input_Params_Filter() โดยฟังก์ชันนี้จะส่งค่าเข้าไปคือ Mp,&Ms,&Fp และ &Fs ซึ่งมีความหมายคือเป็นการส่งค่าตำแหน่งหน่วยความจำของตัวแปรชนิด float (จำนวนจริง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

float Input_Params_Filter(float *Max_Pass,float *Min_Stop,float Freq_Pass,float *Freq_Stop)

เราจะเห็นว่าค่าของตัวแปรทั้ง 4 คือค่าข้อมูลชนิด float เราจะเห็นว่าค่าที่ &Mp,&Ms,&Fp และ &Fsส่งมาให้ฟังก์ชัน Input_Params_Filter()ในฟังก์ชันInput_Params_Filter(); ก็จะมีการรับค่าเข้ามาในฟังก์ชัน เราจะเข้าถึงข้อมูลโดยใช้กรรมวิธี โดยใช้ตัวชี้ทั่วไปคือใช้ตัวกระทำกับตัวแปรที่หมายถึงข่อมูลที่ตัวชี้นั้นๆ อยู่ ซึ่งสามารถแก้ข้อจำกัดข้องภาษาซี โดยเมื่อจบการทำงานของฟังก์ชันและกลับมายังจุดที่เรียกใช้ก็จะได้ค่าใหม่ในการทำงานต่อไป



จากรูปจะเห็นได้ว่าค่าตำแห่งหน่วยความจำที่ตัวชี้ Max_Pass ได้รับคือค่าตำแหน่งของหน่วยความจำของตัวแปรMpเป็นต้นโดยการกระทำใดที่กระทำกับตัวชี้ Max_Pass , Min_Stop ,Freq_Pass และFreq_Stop ก็เท่ากับว่ากระอยู่กับตัวแปร Mp,Ms,Fp,Fs นั่นเอง

ในส่วนของการทำงานในฟังก์ชัน Input_Params_Filter(); จะทำหน้าบอกให้ผู้ใช้ป้อนค่าพารามิเตอร์ เข้ามาโดยจะแสดงข้อความบอกให้ผู้ใช้ป้อนค่าพารามิเตอร์เข้ามา ลักษณะดังรูปด้านล่าง

ในส่วนของการป้อนพารามิเตอร์เข้าไปในฟังก์ชัน Input_Params_Filter(); ค่าที่ได้จะส่งกลับไปที่ฟังก์ชัน main และส่งไปในส่วนของฟังก์ชันอื่นด้วยเพื่อใช้ในการทำงาน เพื่อเป็นการง่ายในการส่งค่าไปให้ฟังก์ชันอื่นทำงานในการออกแบบโปรแกรมจึงได้ทำการจัดเก็บลงไปในรูปไฟล์ โดยได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
**********Filter - The Design Program **********
**********Nomograph 1998 Som&Oat&Sing**********Page=2

You must be in put parameter of Low-Pass Filter
By...
Value Maximum Attenuation Pass Band must be(Positive) between 0-50dB
Value Minimum Attenuation Pass Band must be(Positive) between 0-150dB
And...
Value Frequency Stop Band must'n be 10 times Frequency Pass Band

  Input Maximum Attenuation Pass Band(dB).....: 0.1
  Input Minimum Attenuation Pass Band(dB).....: 60
  Input Frequency Pass Band          (Hz).....: 250
  input Frequency Stop Band          (Hz).....: 2000
 ---------------------------------------------------------------
```

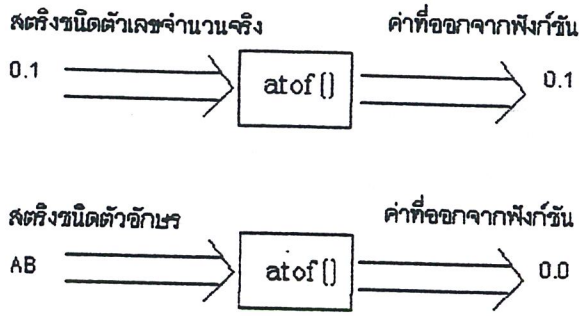
รูปที่ 11 แสดงการป้อนพารามิเตอร์

ทำการจัดเก็บไว้ในไฟล์ชื่อ params.dat โดยจะทำการกำหนดชื่อที่ใช้ในโปรแกรมคือ

```
FILE *fout;      /* define output keep to file */
   ..........................
fout=fopen("params.dat","w");  /* file name params */
   ..........................
fclose(fout);    /* close file params.dat */
```

โดย FILE คือแบบข้อมูลชนิดไฟล์ โดยเราจะต้องทำการเปิดไฟล์โดยฟังก์ชันที่ใช้ในการเปิดไฟล์ คือ fopen() จะทำหน้าที่จองไฟล์บัฟเฟอร์ ตามชื่อของ filename ที่กำหนด เมื่อเปิดไฟล์ได้ตามต้องการก็จะมีการส่งค่าการชี้ข้อมูลไฟล์ ไปเชื่อมกับไฟล์ที่เปิด และเมื่อเราทำการเปิดไฟล์เพื่ออ่านหรือเขียนได้ตามต้องการ แล้วเราก็ต้องทำการคืนไฟล์บัฟเฟอร์ให้แก่ระบบ โดยทำการเปิดไฟล์โดยใช้ฟังก์ชัน fclose() การรับค่าพารามิเตอร์เข้ามาในฟังก์ชันนี้ถ้าเราป้อนค่าที่ไม่เป็นค่าจำนวนจริงจะทำให้โปรแกรมทำงานผิดผลาดและไม่สามารถที่จะทำงานต่อไปได้ จึงได้ทำการแก้ไขโดยใช้ฟังก์ชัน gets() แทนฟังก์ชัน scanf() โดย ฟังก์ชันนี้จะทำหน้าที่รอรับการพิมพ์จากคีย์บอรด์จนกว่าเราจะกดปุ่ม ENTER และส่งค่าที่อ่านได้จากคีย์บอรด์ให้กับฟังก์ชัน atof();

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 12 แสดงการทำงานของฟังก์ชัน atof()

โดยการทำงานของฟังก์ชัน atof(); นี้จะทำหน้าที่ในการแปลงตัวเลขที่อยู่ในรูปสตริงให้เป็นค่าชนิด double ดังนั้นขณะที่เราป้อนตัวอักษรที่เป็นเลขที่เป็นจำนวนจริงเข้ามา ฟังก์ชันนี้จะทำหน้าที่แปลงตัวอักษรที่เป็นตัวเลขให้เป็นเลขจำนวนจริง และในกรณีที่ผู้ใช้โปรแกรมป้อนค่าที่เป็นตัวอักษรเข้ามาฟังก์ชัน atof(); จะแปลงค่าสตริงที่เป็นตัวอักษรให้เป็นตัวเลขซึ่งจะทำให้ฟังก์ชันนี้ส่งค่า ศูนย์กลับมา การเขียนโปรแกรมก็จะมีการตรวจสอบค่าที่ออกมาจากฟังก์ชันว่ามีค่าเป็นศูนย์หรือไม่ถ้ามีค่าเป็นศูนย์ก็จะทำการวนลูปเพื่อให้ผู้ใช้ทำการป้อนค่าให้ถูกต้อง

```
char min[10],max[10],fs[10],fp[10];
    ..................
do{
    printf("      Input Maximum Attenuation Pass Band(dB)..... : ");gets(min);
    Filter_PARA[0].PARA=atof(min);
    if(Filter_PARA[0].PARA==0)
    printf("\a\nYou input value not correct!!....Again please\n\n");
     }while(Filter_PARA[0].PARA==0);
    do{
    printf("      Input Minimum Attenuation Stop Band(dB)..... : ");gets(max);
    Filter_PARA[1].PARA=atof(max);
    if(Filter_PARA[1].PARA==0)
    printf("\a\nYou input value not correct!!....Again please\n\n");
     }while(Filter_PARA[1].PARA==0);
    do{
    printf("      Input Frequency Pass Band(Hz)..... : ");gets(fp);
    Filter_PARA[2].PARA=atof(fp);
    if(Filter_PARA[2].PARA==0)
    printf("\a\nYou input value not correct!!....Again please\n\n");
     }while(Filter_PARA[2].PARA==0);
    do{
    printf("      Input Frequency Stop Band(Hz)..... : ");gets(fs);
    Filter_PARA[3].PARA=atof(fs);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if(Filter_PARA[3].PARA==0)
printf("\a\nYou input value not correct!!....Again please\n\n");
 }while(Filter_PARA[3].PARA==0);
```

Listing 4.5 ส่วนของการตรวจสอบการรับค่า

และในส่วนของการรับค่าพารามิเตอร์เข้ามาก็จะมีการตรวจสอบค่าให้อยู่ในช่วงที่กำหนด โดย ค่า Mp คือค่า Maximum Attenuation Pass Band

Ms คือค่า Minimum Attenuation Stop Band

Fp คือค่า Frequency Pass Band

Fs คือค่า Frequency Stop Band

โดยค่า Mp จะต้องมีค่าอยู่ระหว่าง $0 < Mp < 50$ และค่า $0 < Ms < 150$ และค่า $1 < Fs/Fp < 10$ โดยจะใช้ลูป do-while และจแสดงข้อความบอกผู้ใช้ว่าเกิดข้อผิดผลาดในการป้อนพารามิเตอร์เมื่อผู้ใช้โปรแกรมป้อนพารามิเตอร์ผิดผลาด 3 ครั้ง

```
do{
    ..........
if((Min<=1)||(Max>=50)||(Min<=1)||(Min>=150)||(Fs/Fp)>=10||(Fs/Fp)<=1
)
    printf("\a");
    if(n>=3)
          {
    printf("/***You must input Maximum Attenuation Pass Band between 0-
50 (dB)***/\n");
    printf("/***Input Minimum Attenuation Stop Band between 0-150
(dB)**********/\n");
    printf("/***and Input (Frequency stop band/Frequency pass
Band)<10**********/\n");
    printf("\n/------------------- Press any key to continue ----------------------
/");
    getch();  /* stop */
    n=0;
          }
    }while((Min<=1)||(Max>=50)||(Min<=1)||(Min>=150)||
(Fs/Fp)>=10||(Fs/Fp)<=1);
```

Listing 4.6 ส่วนของการตรวจสอบช่วงของค่าที่ป้อนพร้อมแสดงข้อความผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Utility() Function

การทำงานของฟังก์ชัน Utility(); จะทำหน้ารับค่าพารามิเตอร์ที่ส่งมาจากฟังก์ชัน main() เพื่อที่จะส่งให้ฟังก์ชันอื่นทำงานหรือรับพารามิเตอร์จากฟังก์ชันหนึ่งเพื่อส่งให้กับอีกฟังก์ชันหนึ่งทำงานซึ่งการทำงานของฟังก์ชันนี้จะทำหน้าจะทำหน้าที่คล้ายฟังก์ชัน main() โดยจะเป็นส่วนที่ทำหน้าที่แบ่งการทำงานของฟังก์ชัน main() โดยจะให้ฟังก์ชัน main()ควบคุมการทำงานของโมดูล Utility และอีกประการหนึ่งก็คือทำง่ายในการพัฒนาโปรแกรม

โมดูล Utility จะมีการเรียกใช้ฟังก์ชันอื่นในการเข้ามาทำงานร่วมด้วยคือฟังก์ชัน

Select_Filter();	อยู่ในโมดูล F_SELECT.C
Show_Type_Filter();	อยู่ในโมดูล F_SUM.C
Nomograph();	อยู่ในโมดูล F_SCREEN.C
Find_Order();	อยู่ในโมดูล F_SCREEN.C
Summary_Specify();	อยู่ในโมดูล F_SUM.C

จึงต้องมีการสั่ง include directive เพื่อให้โปรแกรมทราบว่าอยู่โมดูลใดเพื่อที่จะได้นำมาแปลร่วมด้วย

การทำงานของฟังก์ชัน Select_Filter(); โดยฟังก์ชัน Utility(); จะส่งค่า ตำแหน่งหน่วยความจำของตัวแปร Alpha ให้กับฟังก์ชัน Select_Filter(); โดยเมื่อทำฟังก์ชันนี้เสร็จฟังก์ชัน Utility() จะได้ค่าของตัวแปรสองตัวคือค่าของ Alpha (ค่า Alpha จะเป็นค่าเฉพาะของแต่ละฟิลเตอร์เมื่อเราได้ทำการเลือกค่า No_F แล้วค่า Alpha จะถูกกำหนดไว้ล่วงหน้าแล้วแต่เราสามารถที่จะกำหนดค่า Alpha เอง ได้เมื่อเราเลือกชนิดที่ 15) และค่า No_F (ค่า No_F คือค่าหมายเลขที่กดเพื่อเลือกชนิดของฟิลเตอร์คือหมายเลข 0 -15 ซึ่งจะเป็นจำนวนเต็ม)

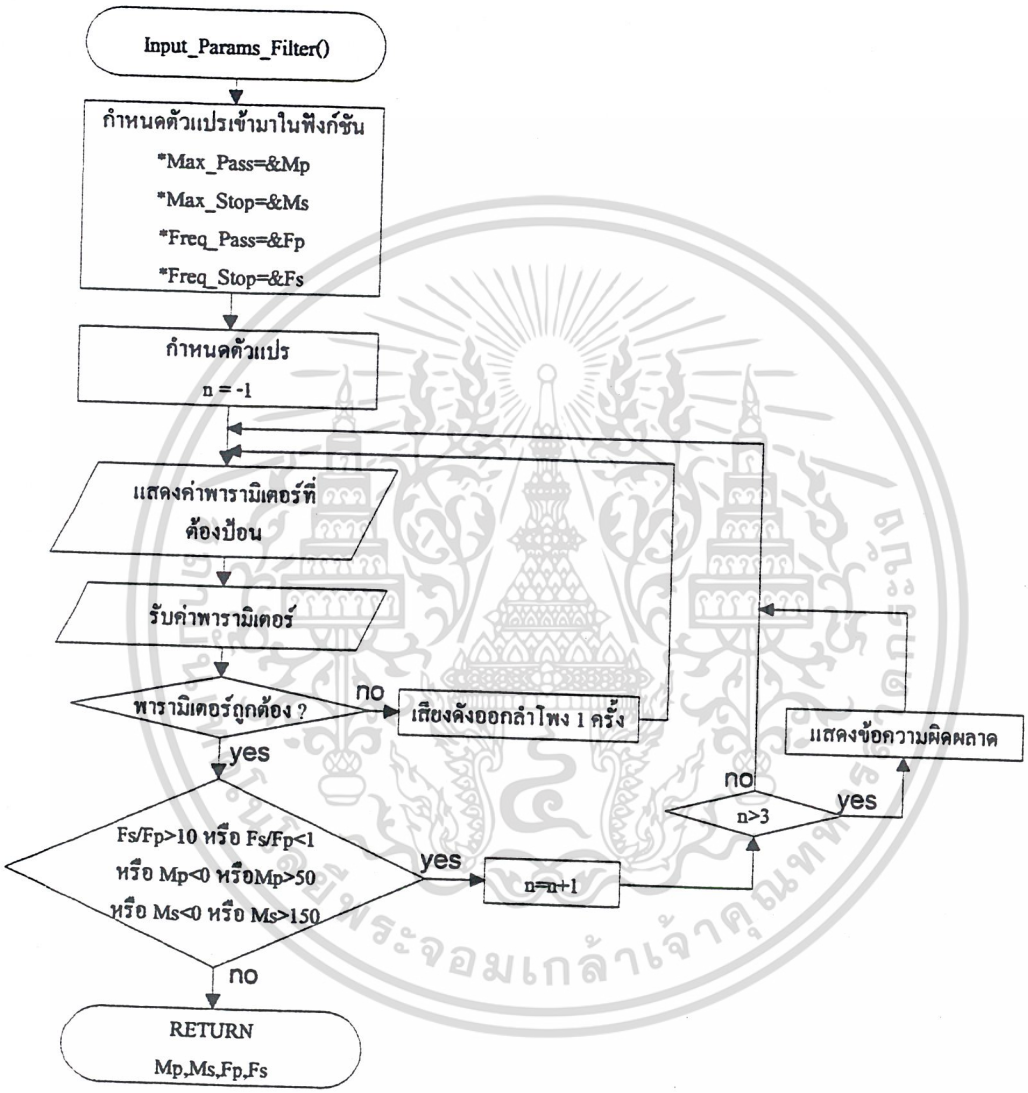
การทำงานของฟังก์ชัน Show_Type_Filter(); โดยฟังก์ชัน Utility(); จะส่งค่าหมายเลขที่ได้จากการเลือกคือ No_F มาให้และค่าพารามิเตอร์ Mp,Ms,Fp,Fs มาให้เพื่อส่งไปให้ฟังก์ชันนี้ทำหน้าที่สรุปผลว่าผู้ใช้ป้อนค่าค่าใดมาบ้างและเลือกฟิลเตอร์ชนิดใดในการออกแบบ

การทำงานของฟังก์ชัน Nomograph(); จะทำหน้าที่ในการแสดงผลของโนโมกราฟ อยู่ในโหมดของกราฟฟิก และมีการส่งค่าของตัวแปรกลับออกมา 2 ค่าคือค่า Point_x,Point_y โดยเป็นค่าตำแหน่งของจุด(pixel) เพื่อใช้ในการคำนวณหาค่า order ของ filter ชนิดนั้น

การทำงานของฟังก์ชัน Find_Order(); จะทำหน้าที่คำนวณหาค่า order ของฟิลเตอร์และส่งค่า order ที่ได้จากการคำนวนกลับมาที่ฟังก์ชัน Utility();

```
#include <conio.h>
#include <stdio.h>
#include <f_select.c>
#include <f_sum.c>
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

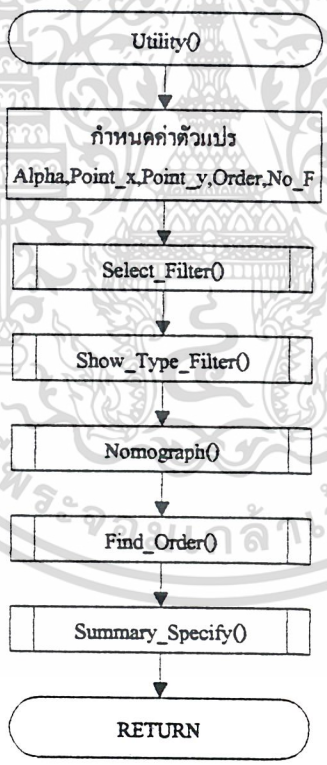


รูป 13 ไฟลวชาร์ตแสดงการทำงานของการรับค่าพารามิเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <f_screen.c>

void Utility(float Mp,float Ms,float Fp,float Fs)
{
    float Alpha;
    int Point_x,Point_y,Order,No_F;

    No_F=Select_Filter(&Alpha);
    clrscr();
    Show_Type_Filter(No_F,Mp,Ms,Fp,Fs,Alpha);       /*display summary */
    Nomograph(&Point_x,&Point_y,Mp,Ms,Fp,Fs,Alpha);
    Find_Order(Point_x,Point_y,&Order);
    Summary_Specify(No_F,Mp,Ms,Fp,Fs,Alpha,Order);  /*display summary
*/
}
```

Listing 4.7 Utility Function



รูป 14 โฟลวชาร์ตการทำงานของฟังก์ชัน Utility

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Select_Filter() Function

ฟังก์ชันจะทำหน้าที่ในการแสดงชนิดของฟิลเตอร์และเลือกฟิลเตอร์ที่จะออกแบบและส่งค่าการเลือกกลับไปที่ฟังก์ชัน Utility();

```
************************ FILTER - The Design Program ************************
************************ Nomograph  1998  Som&Oat&Sing ******************Page=3

   Your select number of Alpha by you must be choose munber
   between 1-15 !
-----------------------------------------------------------------
      1) Alpha=infinity   (Butterworth)
      2) Alpha=10         (10-th M.A.L.)
      3) Alpha=3          (3rd  M.A.L.)
      4) Alpha=2          (2nd  M.A.L.)
      5) Alpha=1.5        (1.5  Ultrasphericall)
      6) Alpha=1          (1st  M.A.L.)
      7) Alpha=0.5        (2nd  Chebyshev)
      8) Alpha=0          (Legendre)
      9) Alpha=-0.25      (-0.25 Ultrasphericall)
     10) Alpha=-0.5       (1st Chevbyshev(Equal-Riple))
     11) Alpha=-0.75      (-0.75 Ultrasphericall)
     12) Alpha=-0.9       (-0.9  Ultrasphericall)
     13) Alpha=-0.95      (-0.95 Ultrasphericall)
     14) Alpha=-0.9999    (-0.9999 Ultrasphericall)
     15) Design Alpha by your!

     Input number of Alpha..... :
```

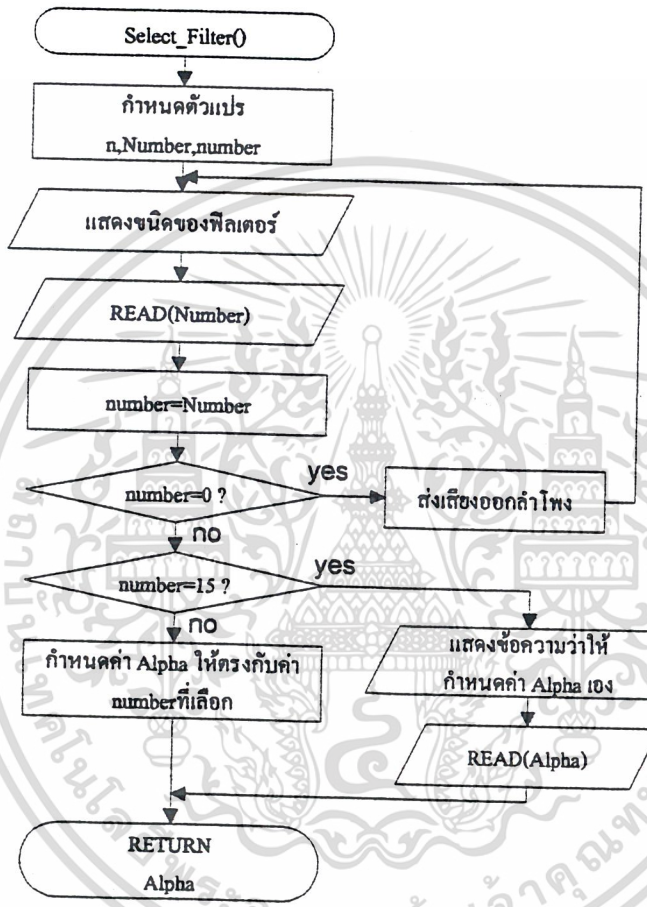
รูปที่ 15 แสดงการทำงานของฟังก์ชัน Select_Filter

```
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>

void  Display_Header(int Page);
float Design_Alpha();
float Select_Filter(float *Alpha);

/*=======================================================
  Select_Filter() - Select type Filter
  Prototype:  float Select_Filter(void);
  Return   :  float
=======================================================*/
float Select_Filter(float *Alpha)
{
    int n=0,number;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    char Number[5];
  do{
    n++;
    Display_Header(3);
    printf("\n");
    printf("\n");
    printf("       Your select number of Alpha by you must be choose
munber\n");
    printf("       between 1-15 !\n");
    printf("-----------------------------------------------------------------\n");
    printf("          1) Alpha=infinity (Butterworth)\n");
    printf("          2) Alpha=10       (10-th M.A.L.)\n");
    printf("          3) Alpha=3        (3rd  M.A.L.)\n");
    printf("          4) Alpha=2        (2nd  M.A.L.)\n");
    printf("          5) Alpha=1.5      (1.5  Ultrasphericaal)\n");
    printf("          6) Alpha=1        (1st  M.A.L.)\n");
    printf("          7) Alpha=0.5      (2nd  Chebyshev)\n");
    printf("          8) Alpha=0        (Legendre)\n");
    printf("          9) Alpha=-0.25    (-0.25 Ultrasphericaal)\n");
    printf("         10) Alpha=-0.5     (1st Chevbyshev(Equal-Riple))\n");
    printf("         11) Alpha=-0.75    (-0.75 Ultrasphericaal)\n");
    printf("         12) Alpha=-0.9     (-0.9  Ultrasphericaal)\n");
    printf("         13) Alpha=-0.95    (-0.95 Ultrasphericaal)\n");
    printf("         14) Alpha=-0.9999  (-0.9999 Ultrasphericaal)\n");
    printf("         15) Design Alpha by your!\n");
    printf("\n");
    printf("           Input number of Alpha..... : ");
    gets(Number);
    number=atoi(Number);
    if(number!=0)
    switch(number)
    {
     case 1 :*Alpha=100000000;break;
     case 2 :*Alpha=10;break;
     case 3 :*Alpha=3;break;
     case 4 :*Alpha=2;break;
     case 5 :*Alpha=1.5;break;
     case 6 :*Alpha=1;break;
     case 7 :*Alpha=0.5;break;
     case 8 :*Alpha=0;break;
     case 9 :*Alpha=-0.25;break;
     case 10:*Alpha=-0.5;break;
     case 11:*Alpha=-0.75;break;
     case 12:*Alpha=-0.9;break;
     case 13:*Alpha=-0.95;break;
     case 14:*Alpha=-0.9999;break;
     case 15:*Alpha=Design_Alpha();
    }
    if((number<1)||(number>15))printf("\a");
  }while((number<1)||(number>15));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 16 โฟลวชาร์ตการทำงานของฟังก์ชัน Select_Filter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 return number;
}
/*============================================================
  Design_Alpha(void) - Select Alpha by yourself
  Prototype:  float Desing_Alpha(void);
  Return   :  float
============================================================*/
float Design_Alpha()
{
 float Al_ph;
 printf("       Please specify Alpha..... : ");
 scanf("%f",&Al_ph);
 return Al_ph;
}
```

Listing 4.8 Select_Filter Function

Show_Type_Filter() Function

ฟังก์ชันนี้จะอยู่ในโมดูล F_SUM.C จะทำหน้าที่ในการสรุปพารามิเตอร์ที่ป้อนเข้ามาและแสดงออกมาที่หน้าจอ โดยจะรับค่าชนิดของฟิลเตอร์ที่เลือก และค่า Mp,Ms,Fs,Fp เข้ามา

โดยค่า ชนิดของฟิลเตอร์ที่ส่งมาจากฟังก์ชัน Utility(); จะใช้ตัวแปร No_F ซึ่งเป็นตัวแปรชนิดจำนวนเต็ม แต่เมื่อส่งผ่านค่าเข้ามาในฟังก์ชัน Show_Type_Filter(); เราจะกำหนดค่าชนิดของฟิลเตอร์ให้กับตัวแปร number_F ซึ่งเป็นตัวแปรชนิดจำนวนเต็มและฟังก์ชัน Show_Type_Filter(); จะมีการใช้ Switch-case Statement เพื่อตรวจสอบค่าการเลือกว่าจะแสดงข้อความของชนิดฟิลเตอร์ว่ามีการเลือกใช้ฟิลเตอร์ใด

```
#include <stdio.h>
#include <conio.h>

void Display_Header(int Page);
void Summary_Specify(int number_F,float Mag_Pass,float Mag_Stop,
                     float Fp,float Fs,float Alpha,int Order);

/*============================================================
  Show_Type_Filter() - displays of type of Filter you choose
  Prototype:  void Show_Type_Filter();
  Return   :  void
============================================================*/
void Show_Type_Filter(int number_F,float Mag_Pass,float Mag_Stop,float
Fp,float Fs,float Alpha)
{
 char ans;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 Display_Header(4);
 printf("\n\n");
 printf("                    SUMARY SPECIFY RESPONSE\n");
 printf("\n\n");
 switch(number_F)
{
 case 1:
 {
 printf("        Approximation n-th order Low-Pass..Butterworth filter\n");
 break;
 }
 case 2:
 {
 printf("        Approximation n-th order Low-Pass..10th M.A.L. filter\n");
 break;
 }
 case 3:
 {
 printf("        Approximation n-th order Low-Pass..3rd M.A.L. filter\n");
 break;
 }
 case 4:
 {
 printf("        Approximation n-th order Low-Pass..2nd M.A.L. filter\n");
 break;
 }
 case 5:
 {
 printf("        Approximation n th order Low-Pass..1.5 Ultraspherical
filter\n");
 break;
 }
 case 6:
 {
 printf("        Approximation n-th order Low-Pass..1st M.A.L. filter\n");
 break;
 }
 case 7:
 {
 printf("        Approximation n th order Low-Pass..2nd Shevbyshev
filter\n");
 break;
 }
 case 8:
 {
 printf("        Approximation n th order Low-Pass..Legendre filter\n");
 break;
 }
 case 9:
 {
```

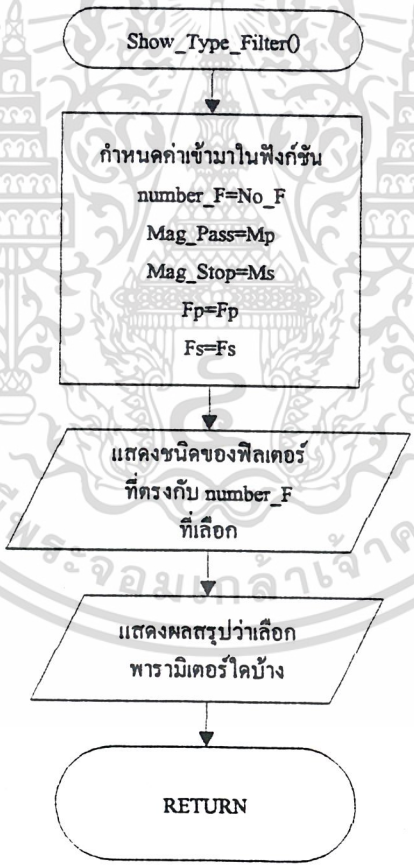
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 printf("          Approximation n th order Low-Pass..-0.25 Ultraspherical
filter\n");
 break;
 }
 case 10:
 {
 printf("       Approximation n th order Low-Pass..1st Chevbyshev(Equal-
Ripple) filter\n");
 break;
 }
 case 11:
 {
 printf("          Approximation n th order Low-Pass..-0.75 Ultraspherical
filter\n");
 break;
 }
 case 12:
 {
 printf("          Approximation n th order Low-Pass..-0.9 Ultraspherical
filter\n");
 break;
 }
 case 13:
 {
 printf("          Approximation n th order Low-Pass..-0.95 Ultraspherical
filter\n");
 break;
 }
 case 14:
 {
 printf("          Approximation n th order Low-Pass..-0.9999 Ultraspherical
filter\n");
 break;
 }
 case 15:
 {
 printf("          Approximation n th order Low-Pass..????? Ultraspherical
filter\n");
 break;
 }
}
if(number_F==1)
 {
 printf("\n");
 printf("                Filter impelmentation is  :   Analog\n");
 printf("                Filter Alpha is           :   Infinity\n");
 printf("                Passband gain        (dB):   %0.2f\n",Mag_Pass);
 printf("                Stopband gain        (dB):   %0.2f\n",Mag_Stop);
 printf("                Passband frequency   (Hz):   %0.2f\n",Fp);
 printf("                Stopband frequency   (Hz):   %0.2f\n",Fs);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 }
else
 {
 printf("\n");
 printf("                Filter impelmentation is :   Analog\n");
 printf("                Filter Alpha is          :   %0.4f\n",Alpha);
 printf("                Passband gain            (dB):   %0.2f\n",Mag_Pass);
 printf("                Stopband gain            (dB):   %0.2f\n",Mag_Stop);
 printf("                Passband frequency       (Hz):   %0.2f\n",Fp);
 printf("                Stopband frequency       (Hz):   %0.2f\n",Fs);
 }
 getch();
}
```

Listing 4.9 Show_Type_Filter Function



รูป 17 ไฟลวชาร์ตการทำงานของฟังก์ชัน Show_Type_Filter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Nomograph() Function

เป็นฟังก์ชันที่ทำหน้าที่ในการแสดงผลลักษณะเป็นโนโมกราฟโดยจะแสดงเป็นกราฟฟิก ที่จะอยู่ในโมดูล F_SCREEN.C ที่ฟังก์ชันนี้จะมีการรับค่า พารามิเตอร์ที่ใช้ในการคำนวณคือ Mp,Ms ,Fp,Fs จากฟังก์ชัน Utility(); แต่ฟังก์ชัน Nomograph(); จะทำการรับค่า มาใส่ให้กับตัวแปร Max_Pass,Min_Stop,Freq_Pass,Freq_Stop และมีการรับค่าตำแหน่งหน่วยความจำ จากฟังก์ชัน Utility(); มากำหนดให้กับตัวแปร *point_x,*point_y ที่ทำหน้าที่เป็นตัวชี้ข้อมูล ชนิดจำนวนเต็ม(int)

```
void Nomograph(int *point_x,int *point_y,float Max_Pass,float Min_Stop
                ,float Freq_Pass,float Freq_Stop,float Alpha)
```

เพื่อที่จะส่งค่า point_x,point_y กลับไปให้กับฟังก์ชัน Utility(); เพื่อป้อนให้กับฟังก์ชัน Find_Order(); ในการหาอันดับของฟิลเตอร์ โดยจะมีการกำหนดตัวแปรที่เก็บค่าที่ใช้การพล็อตกราฟ โดยกำหนดให้เป็นตัวแปร โกบอล(Golbal Variables) ภายในโมดูล F_SCREEN.C โดยที่จะให้ฟังก์ชัน Find_Order(); สามารถ ใช้ค่าในตัวแปรนี้ได้

```
int Value[20][450];
```

ตัวแปรนี้จะมีชื่อเป็น Value เป็นตัวแปรชนิด Array เก็บค่าจำนวนเป็น Array หลายมิติ(Multi-dimenstional Array) โดยจะใช้เก็บค่า Pixel ที่ใช้ในการพล็อตกราฟ

การทำงานของฟังก์ชัน Nomograph(); จะแบ่งการทำงานออกเป็นสองส่วนคือ

1)การแสดงหน้าจอของโนโมกราฟ

2)การพล็อต Curve Order

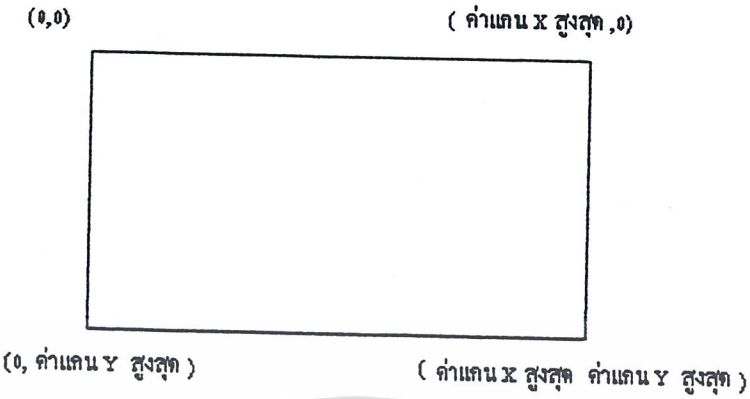
การแสดงหน้าจอโนโมกราฟ

การแสดงหน้าจอโนโมกราฟจะแสดงกราฟในโหมดของฟังก์ชันกราฟฟิก(Graphics) โดยปรกติจะแสดงอยู่ในโหมดของตัวอักษร เมื่อต้องการเปลี่ยนให้อยู่ในโหมดภาพจะใช้ฟังก์ชัน initgraph();เพื่อเปลี่ยนการทำงานให้เป็นโหมดภาพและจะใช้ร่วมกับฟังก์ชัน closegraph(); เพื่อปิดโหมดภาพ

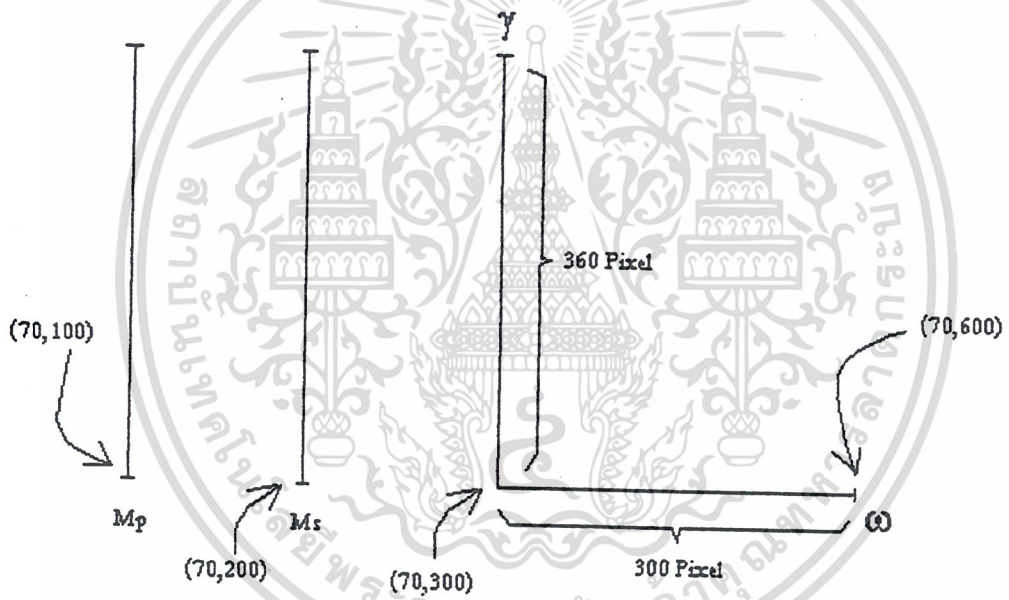
พารามิเตอร์ใน initgraph(); มีรายละเอียดดังนี้

graphdriver เป็นตัวชี้สำหรับชี้ไปยังหมายเลขโหมดที่ต้องการ โดยทั่วไปจะกำหนดให้เท่ากับ DETECT ซึ่งเป็นค่าคงที่ หมายถึงกำหนดให้แสดงภาพให้มีความละเอียดสูงสุดที่การ์ดสามารถทำงานได้ ซึ่งในโหมดภาพที่แสดงในฟิลเตอร์จะแสดงตามมาตรฐานวีจีเอ ซึ่งได้จาก initgraph(); จะมีความละเอียด 640 จุด คูณ 480 จุด 16 สี โคออร์ดิเนตของมุมซ้ายของมุมบนซ้ายของจอมีค่า (0,0) และมุมล่างขวามีค่า (639,479)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ค่าสูงสุดทางแกม X และแกน Y สามารถที่จะหาได้โดยใช้ฟังก์ชัน getmaxy() , getmaxy()



รูปที่ 18 ลักษณะการวางตำแหน่งของการแสดงโนโมกราฟ

การเริ่มในการพล็อตกราฟนั้นเราจะทำการจัดแบ่งหน้าจอในการแสดงผลออกเป็นส่วนๆ โดย ส่วนที่ทำหน้าที่แสดงแกน Mp , Ms และ γ นั้นจะให้เริ่มแสดงที่ตำแหน่ง Pixel ที่ โคออร์ดิเนต (70,100) , (70,200) และ (70,300) ความยาวของแกน Mp,Ms และ γ มีค่าเท่ากันคือ 360 pixel ค่าความยาวของแกน ω ให้มีความยาวของแกน 300 pixel ดังรูป

การสร้างขีดสเกลบนแกน Mp , Ms และ Ω เราจะต้องใช้สมการคณิตศาสตร์เพื่อช่วยในการเขียนซึ่งจะให้ผลที่ได้นั้นมีความถูกต้องที่สุด โดยใช้ความสัมพันธ์ของวงจรกรองความถี่ต่ำคือ

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$|M(j\omega)|^2 = 1 + \varepsilon^2 S^2{}_n(\omega)$$

และจะได้ว่า $$M_S{}^2 = 1 + \left(M_p{}^2 - 1\right) S_n{}^2(w)$$

เมื่อทำสมการให้อยู่ในรูปของ Logalithm

$$log(M_S{}^2 - 1) = log(M_P{}^2 - 1) + Log\left(S_n{}^2(\omega)\right)$$

ทำให้อยู่ในรูปของความสัมพันธ์

$$f(u) + f(v) = \gamma$$

$$\gamma = f(\omega, t)$$

เนื่องจากค่าที่แทนใน Ms และ Mp ด้านบนนั้นจะไม่ได้อยู่ในหน่วย dB ดังนั้นเพื่อทำให้ค่าที่นำมาแทนนั้นอยู่ในรูป dB จึงได้เป็น

$$Log(10^{M_S(dB)/10} - 1) - Log(10^{M_P(dB)/10} - 1) = Log\, S_n{}^2(\omega)$$

ดังนั้นเราจะให้เป็น γ เป็นตัวเชื่อมความสัมพันธ์ทั้งสอง

จะได้ $$Log(10^{M_S(dB)/10} - 1) - Log(10^{M_P(dB)/10} - 1) = \gamma$$

และ $$\gamma = Log\, S_n{}^2(\omega)$$

เมื่อนำ $$Log(10^{M_S(dB)/10} - 1) - Log(10^{M_P(dB)/10} - 1) = \gamma$$

มาพิจารณา ในความสัมพันธ์ที่เราเคยศึกษามาแล้วในตอนต้นคือ

$$u + v = w$$

ดังนั้นถ้า Ms และ $Mp >> 10$ จะได้ว่า

$$M_S/10 - M_P/10 = \gamma \quad \text{เมื่อ } Ms \text{ และ } Mp >> 10\ dB$$

เมื่อเราสร้างให้ระยะห่างของแกนมีระยะดังรูป 18

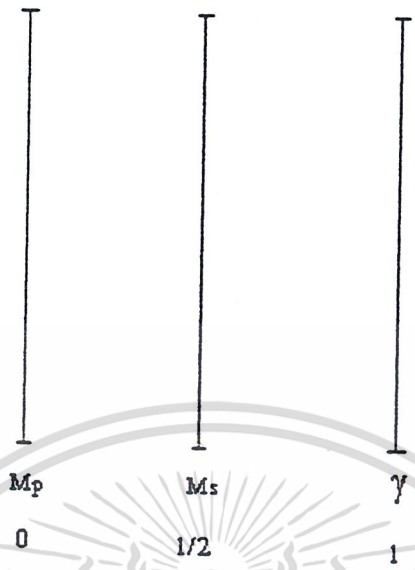
การวางระยะห่างดังรูปด้านบนจะทำให้สามารถที่จะจัดให้อยู่ในรูป Basis determinant คือ

$$\begin{vmatrix} 0 & u & 1 \\ 1 & v & 1 \\ 1/2 & w/2 & 1 \end{vmatrix} = 0$$

จะได้

$$\begin{vmatrix} 0 & M_P/10 & 1 \\ 1 & \gamma & 1 \\ 1/2 & M_S/20 & 1 \end{vmatrix} = 0$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 19

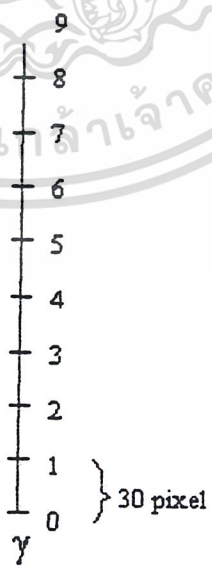
คือ

$$x1 = 0 \quad y1 = Mp/10$$

$$x2 = 1/2 \quad y2 = \gamma$$

$$x3 = 1 \quad y3 = Ms/20$$

ดังนั้นเราจะเริ่มสร้างแกน γ ให้มีความยาว 12 หน่วย โดยกำหนดให้แต่ละหน่วยมีระยะห่างบนแกน y เท่ากันคือ 30 จุดต่อหน่วย(ค่านี้จะกำหนดอย่างไรก็ได้ขึ้นอยู่กับความเหมาะสมในการแสดงผลของภาพบนหน้าจอ



รูปที่ 20 แสดงการกำหนดจุดบนแกน γ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่เนื่องจากให้ *Mp* มีค่าไม่เกิน 50 dB และ *Ms* มีค่าไม่เกิน 150 dB เราจะได้ γ ไม่เกิน 10 ซึ่งเราจะกำหนดให้อยู่บนแนวเดียวกัน

ดังนั้นเราจะได้ความสัมพันธ์ต่ออีกว่าความยาวของแกน *Mp* , *Ms* และ γ จะมีความยาวเท่ากันคือ 10 หน่วยและจะมีจุดบนแกน y เท่ากันคือ 300 จุด และเพื่อทำให้ค่า Mp = 50 และ Ms =150 อยู่

รูปที่ 21

บนแนวเดียวกันกับ $\gamma = 10$ จึงได้เป็น

$$x1 = 0 \quad y1 = Mp/10 + 5$$

$$x2 = 1/2 \quad y2 = \gamma$$

$$x3 = 1 \quad y3 = Ms/20 + 2.5$$

เราจะใช้ความสัมพันธ์นี้เมื่อ Mp>20 และ Ms>90 dB และจะได้ระยะห่างระหว่างจุดหลัก(จุดหลักคือจุดที่เพิ่มขึ้นครั้งละ 10 dB) ของแกน Mp คือ 1 หน่วย(20 , 30 , 40 , 50 dB) ส่วนแกน Ms คือ 0.5 หน่วยระหว่างจุดหลัก(90 , 100 , 110 , 120 , 130 , 140 , 150 dB) ถ้า Mp< 20 dB จะไม่ใช้สมการนี้เพราะจะทำให้เกิดการผิดผลาดมาก

เช่น Mp= 50 dB ค่าความยาวแกน $\gamma = 10$ จำนวนจุดในแนวแกนตั้ง 300 จุด

Mp= 40 dB ค่าความยาวแกน $\gamma = 9$ จำนวนจุดในแนวแกนตั้ง 270 จุด

Mp= 50 dB ค่าความยาวแกน $\gamma = 8$ จำนวนจุดในแนวแกนตั้ง 240 จุด

Mp= 50 dB ค่าความยาวแกน $\gamma = 7$ จำนวนจุดในแนวแกนตั้ง 210 จุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และถ้าอยู่ในกรณี Mp<20 จะใช้ความสัมพันธ์

$$Log(10^{M_P(dB)/10} - 1) + 5 = \text{ความยาวแกน } \gamma$$

เช่นเมื่อ Mp = 10 dB จะได้ความยาวของแกน γ = 5.95 หน่วย = 178 จุด เมื่อเรากำหนดจุดหลักที่ต้องการหาเช่น 5 , 1 , 0.5 , 0.1 , 0.05 , 0.01 เราก็จะได้จำนวนจุดที่เปลี่ยนไปบนแกน y ในทำนองเดียวกับการทำสเกลบนแกน Ms เราก็จะได้ความสัมพันธ์

$$M_S/20 + 2.5 = \text{ความยาวแกน } \gamma$$

และ Ms<10 dB ก็จะใช้ความสัมพันธ์

$$Log(10^{Ms(dB)} - 1) + 2.5 = \text{ความยาวแกน } \gamma$$

และนำค่าความยาวนี้ไปเปลี่ยนเป็นจุบนแกน γ เพื่อเตรียมไว้พล็อตโดยใช้ความสัมพันธ์

จำนวนจุดที่ได้บนแกน y = ระยะบนแกน y (หน่วย) x 30

การสร้างแกน Ω (normalized stop band frequency) โดยความถี่ที่ป้อนเข้ามาจะต้องทำการนอมอไรส์ก่อนเพื่อที่จะได้เปรียบเทียบกับฟิลเตอร์ชนิดอื่นๆ ได้
โดย

$$\Omega = \frac{f_S}{f_P}$$

เมื่อ fs คือความถี่ Frequency Stop และ fp คือ Freq Pass
และทำการใส่ Logarithm เพื่อทำการบีบสเกลให้มีขนาดเล็กเพื่อที่จะได้พล็อตลงในหน้าจอได้พอดี

จำนวนหน่วยบนแกน $X = Log(\Omega)$

และเนื่องจากว่าเรากำหนดให้ความยาวของแกน Ω มีความยาวทั้งหมดเท่ากับ 300 จุดดังนั้นเราจำเปลี่ยนหน่วยที่ได้บนแกน X ให้เป็นจำนวนจุดได้ดังนี้

จำนวนจุดที่ได้บนแกน $y = 300 \times Log(\Omega)$

การเข้าสู่โหมดรูปภาพ

```
#include <graphics.h>
.....
void Nomograph(int *point_x,int *point_y,float Max_Pass,float Min_Stop,float
Freq_Pass,float Freq_Stop,float Alpha)
{
.....
int   drive=DETECT,mode=0,Omega,Gamma;
.....
/* set graphic Mode */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    initgraph(&drive,&mode,"");
.....
setcolor(RED);
    outtextxy((getmaxx()/2-50),470,"NomoGraph");
    setcolor(7);  /* color LIGHTGRAY */
    line(1,1,1,getmaxy());
    line(1,1,getmaxx(),1);
    line(getmaxx(),getmaxy(),1,getmaxy());
    line(getmaxx(),getmaxy(),getmaxx(),1);
    line(100,convert_y(70),100,convert_y(430));      /*draw axil Mp*/
    line(200,convert_y(70),200,convert_y(430));      /*draw axil Ms*/
    rectangle(300,convert_y(430),600,convert_y(70)); /*draw Rectan */
    line(298,convert_y(100),302,convert_y(100));
    line(298,convert_y(130),302,convert_y(130));
    line(298,convert_y(160),302,convert_y(160));
    line(298,convert_y(190),302,convert_y(190));
    line(298,convert_y(220),302,convert_y(220));
    line(298,convert_y(250),302,convert_y(250));
    line(298,convert_y(280),302,convert_y(280));
    line(298,convert_y(310),302,convert_y(310));
    line(298,convert_y(340),302,convert_y(340));
    line(298,convert_y(370),302,convert_y(370));
    line(298,convert_y(400),302,convert_y(400));
/* Ms */
    line(198,convert_y(310),202,convert_y(310));
    line(198,convert_y(355),202,convert_y(355));
    line(198,convert_y(340),202,convert_y(340));
    line(198,convert_y(325),202,convert_y(325));
    line(198,convert_y(370),202,convert_y(370));
    line(198,convert_y(295),202,convert_y(295));
    line(198,convert_y(280),202,convert_y(280));
    line(198,convert_y(265),202,convert_y(265));
    line(198,convert_y(235),202,convert_y(235));
    line(198,convert_y(175),202,convert_y(175));
    line(198,convert_y(190),202,convert_y(190));
    line(198,convert_y(205),202,convert_y(205));
    line(198,convert_y(220),202,convert_y(220));
    line(198,convert_y(250),202,convert_y(250));
    line(198,convert_y(265),202,convert_y(265));
    line(198,convert_y(235),202,convert_y(235));
    line(198,convert_y(220),202,convert_y(220));
    line(198,convert_y(205),202,convert_y(205));
    line(198,convert_y(190),202,convert_y(190));
    line(198,convert_y(175),202,convert_y(175));
    line(198,convert_y(159),202,convert_y(159));
    line(198,convert_y(136),202,convert_y(136));
    line(198,convert_y(120),202,convert_y(120));
    line(198,convert_y(70),202,convert_y(70));
/* Mp */
    line(98,convert_y(310),102,convert_y(310));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

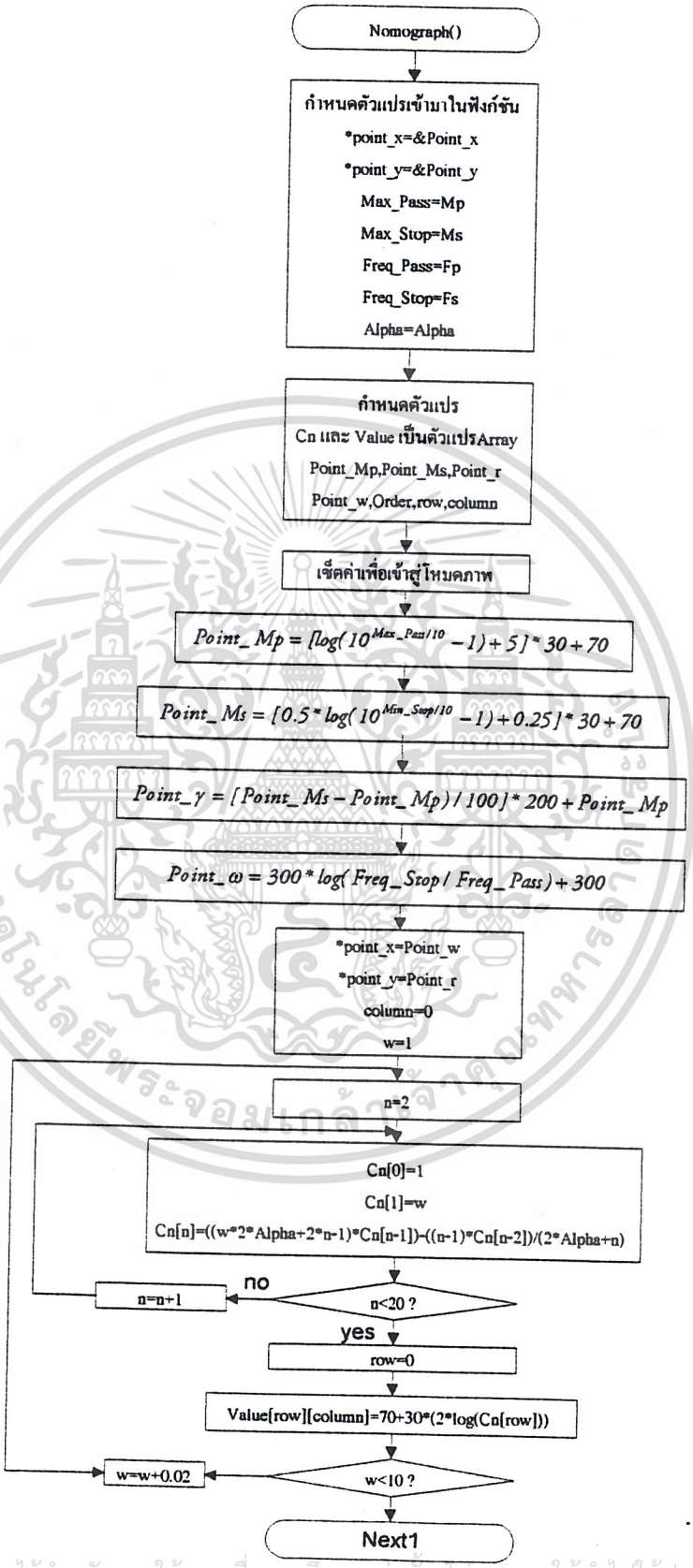
```
        line(98,convert_y(355),102,convert_y(355));
        line(98,convert_y(325),102,convert_y(325));
        line(98,convert_y(295),102,convert_y(295));
        line(98,convert_y(355),102,convert_y(355));
        line(98,convert_y(340),102,convert_y(340));
        line(98,convert_y(370),102,convert_y(370));
        line(98,convert_y(280),102,convert_y(280));
        line(98,convert_y(70),102,convert_y(70));
        line(98,convert_y(264),102,convert_y(264));
        line(98,convert_y(248),102,convert_y(248));
        line(98,convert_y(230),102,convert_y(230));
        line(98,convert_y(202),102,convert_y(202));
        line(98,convert_y(192),102,convert_y(192));
        line(98,convert_y(171),102,convert_y(171));
        line(98,convert_y(162),102,convert_y(162));
        line(98,convert_y(141),102,convert_y(141));
    /* w */
        setcolor(7);  /* color LIGHTGRAY */
        line(390,convert_y(70),390,convert_y(430));
        line(443,convert_y(70),443,convert_y(430));
        line(481,convert_y(70),481,convert_y(430));
        line(510,convert_y(70),510,convert_y(430));
        line(534,convert_y(70),534,convert_y(430));
        line(554,convert_y(70),554,convert_y(430));
        line(571,convert_y(70),571,convert_y(430));
        line(586,convert_y(70),586,convert_y(430));
    /* text axil w  */
        setcolor(11);
        outtextxy(300-2,convert_y(60),"1");
        outtextxy(390-2,convert_y(60),"2");
        outtextxy(443-2,convert_y(60),"3");
        outtextxy(481-2,convert_y(60),"4");
        outtextxy(510-2,convert_y(60),"5");
        outtextxy(534-2,convert_y(60),"6");
        outtextxy(554-2,convert_y(60),"7");
        outtextxy(571-2,convert_y(60),"8");
        outtextxy(586-2,convert_y(60),"9");
        outtextxy(600-4,convert_y(60),"10");
    /* text axil r  */
        outtextxy(290,convert_y(70)-2,"0");
        outtextxy(290,convert_y(100)-2,"1");
        outtextxy(290,convert_y(130)-2,"2");
        outtextxy(290,convert_y(160)-2,"3");
        outtextxy(290,convert_y(190)-2,"4");
        outtextxy(290,convert_y(220)-2,"5");
        outtextxy(290,convert_y(250)-2,"6");
        outtextxy(290,convert_y(280)-2,"7");
        outtextxy(290,convert_y(310)-2,"8");
        outtextxy(290,convert_y(340)-2,"9");
        outtextxy(290-8,convert_y(370)-2,"10");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        outtextxy(290-8,convert_y(400)-2,"11");
        outtextxy(290-8,convert_y(430)-2,"12");
        outtextxy(450-3,convert_y(40),"ē");
    /* text axil Ms  */
        outtextxy(190-17,convert_y(370)-2,"150");
        outtextxy(190-17,convert_y(340)-2,"130");
        outtextxy(190-17,convert_y(310)-2,"110");
        outtextxy(190-8,convert_y(280)-2,"90");
        outtextxy(190-8,convert_y(250)-2,"70");
        outtextxy(190-8,convert_y(220)-2,"50");
        outtextxy(190-8,convert_y(190)-2,"30");
        outtextxy(190-8,convert_y(175)-2,"20");
        outtextxy(190-8,convert_y(159)-2,"10");
        outtextxy(190,convert_y(136)-2,"1");
        outtextxy(190-17,convert_y(120)-2,"0.1");
        outtextxy(100-6,convert_y(60),"Mp");
        outtextxy(100-6,convert_y(440),"dB");
    /* text axil Mp  */
        outtextxy(90-8,convert_y(370)-2,"50");
        outtextxy(90-8,convert_y(340)-2,"40");
        outtextxy(90-8,convert_y(310)-2,"30");
        outtextxy(90-8,convert_y(280)-2,"20");
        outtextxy(90-8,convert_y(248)-2,"10");
        outtextxy(90,convert_y(230)-2,"5");
        outtextxy(90,convert_y(202)-2,"1");
        outtextxy(90-17,convert_y(192)-2,"0.5");
        outtextxy(90-17,convert_y(171)-2,"0.1");
        outtextxy(90-24,convert_y(162)-2,"0.05");
        outtextxy(90-24,convert_y(141)-2,"0.01");
        outtextxy(200-6,convert_y(60),"Ms");
        outtextxy(200-6,convert_y(440),"dB");

        getch();                /* stop and show Nomograph */
        closegraph();

}                               /*  end of function nomograph  */
```

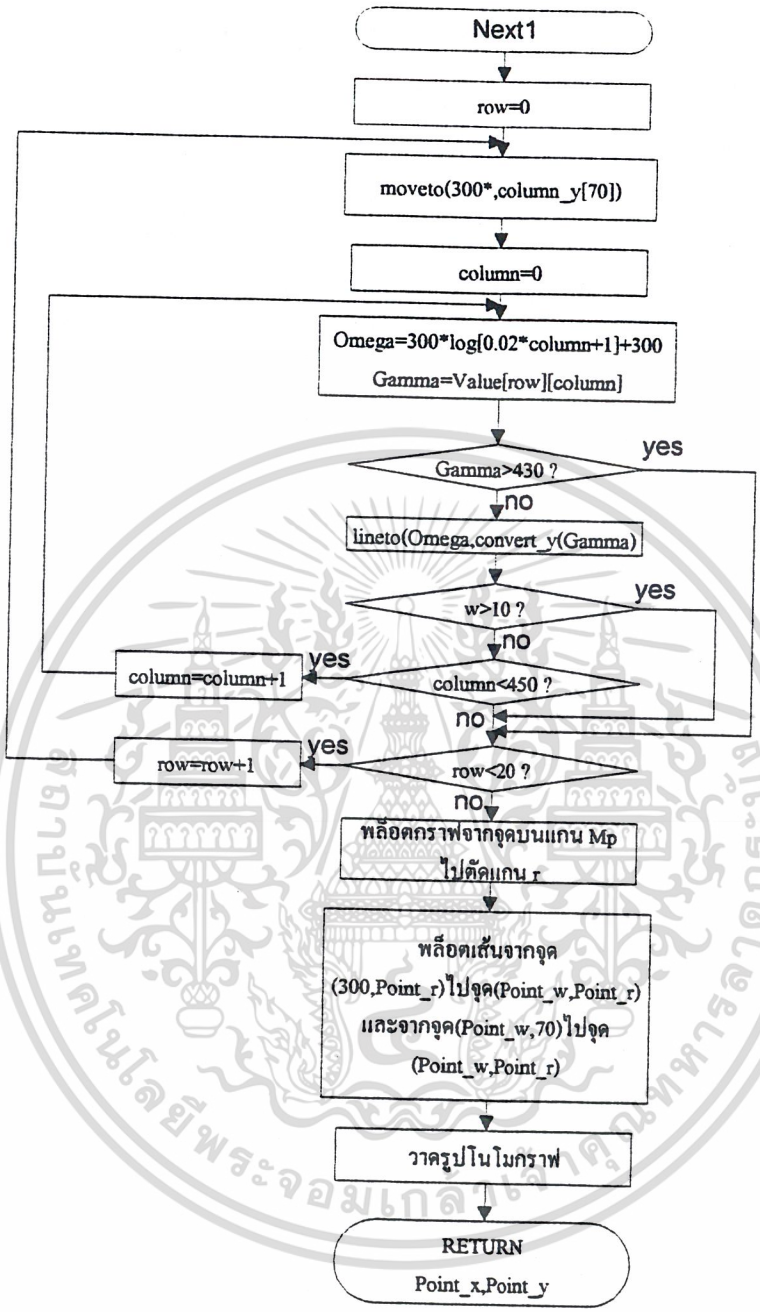
Listing 4.10 การแสดงหน้าจอโนโมกราฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



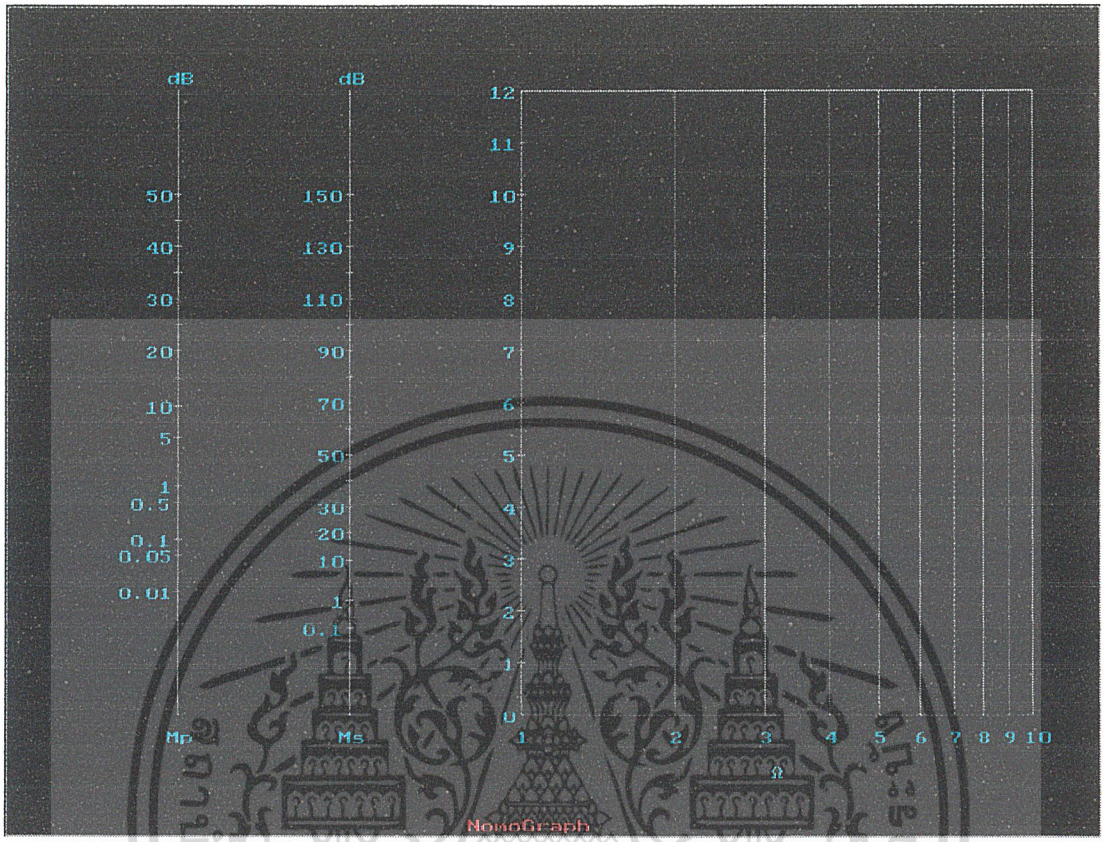
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 22 โฟลวชาร์ตการทำงานของฟังก์ชัน Nomograph



รูป 22b โฟลวชาร์ตการทำงานของฟังก์ชัน Nomograph

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 23 แสดงโหมดภาพของโนโมกราฟ

การพล็อตเคอฟ์แสดงอันดับ

การพล็อตเคือฟ์เราจะนำเอาสมการรีเคอร์ชั่นของ Ultraspheirical filter มาทำการพล็อต โดยมีสมการคือ

$$(2\alpha+n)F_n^{\alpha}(x)=x(2\alpha+2n-1)F_{n-1}^{\alpha}(x)-(n-1)F_{n-2}^{\alpha}(x)$$

เมื่อ

$$F_0^{\alpha}(x)=1$$

$$F_1^{\alpha}(x)=x$$

การทำงานของในส่วนของการพล็อตเคอร์ฟออร์เดอร์นั้นใช้การทำงานของลูป for statement โดยจะกำหนดให้มีการเพิ่มขึ้นของตัวแปรตั้งแต่ 1 เพิ่มขึ้นที่ละ 0.02 จนกระทั่งถึงค่า 9.98 ซึ่งจะมีทั้งหมด 450 ค่า โดยเราจะพล็อตเคอร์ฟออร์เดอร์ ทั้งหมด 20 ออร์เดอร์

การทำงานของโปรแกรมในส่วนนี้จะทำหน้าที่เพิ่มค่าของ w ขึ้นที่ละ 0.02 แล้วนำไปแทนค่าลงในสมาการรีเคอร์ชั่น ซึ่งการทำงานของโปรแกรมจะทำหน้าที่หาค่าโดยใช้ค่า w ค่าเดียวหาค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไปจนถึง20 ออร์เดอร์ แล้วนำไปเก็บในตัวแปร float Cn[20]; ที่ทำหน้าที่ในการเก็บตัวแปรชนิดค่าจำนวนจริง

```
/*Plot curve Ultraspheirical filter*/
column=0;
for(w=1;w<10;w=w+0.02)
 {
 for(n=2;n<20;n++)  /*calculation and keep to 20 cell*/
        {                                   /**/
        Cn[0]=1;                            /**/
        Cn[1]=w;                            /**/
        Cn[n]=((w*(2*Alpha+2*n-1)*Cn[n-1])-((n-1)*Cn[n-2]))/(2*Alpha+n);
        }
```

และในส่วนของการทำงานต่อไปของลูปต่อไปจะทำการนำค่าที่อยู่ในตัวแปรชนิด Array คือ float Cn[20]; แปลงให้อยู่ในรูปของ pixel ที่เป็นจำนวนเต็มแล้วในไปเก็บในตัวแปร

```
int Value[20][450];
```

ซึ่งตัวแปร int Value[20][450]; จะเหมือนกับมีค่าในการเก็บทั้งหมด 9000 ค่า

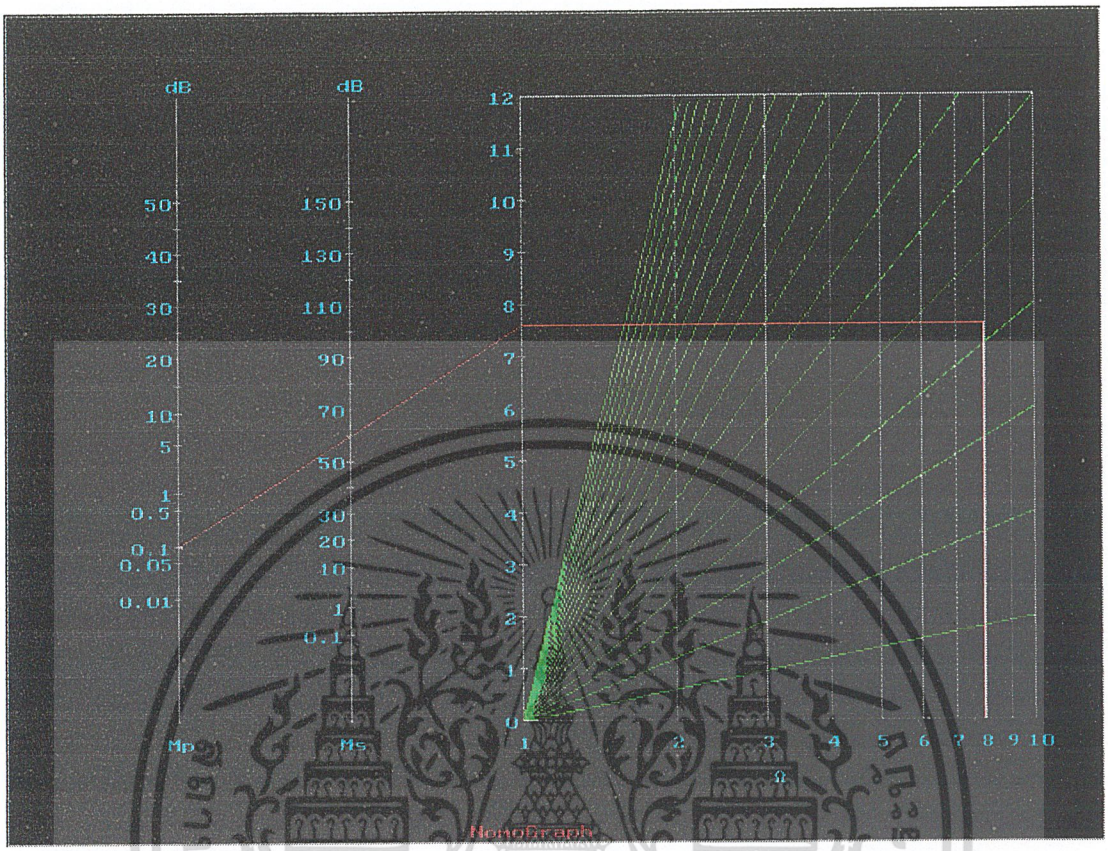
และในส่วนการทำงานต่อไปก็จะนำค่าที่อยู่ในตัวแปร Value ที่เป็นตัวแปร Array จะมีการนำค่าที่อยู่ใน Array ออกมาพล็อต

```
        for(row=0;row<20;row++)         /* keep 20 cell to main matrix */
        Value[row][column]=(int)(70+30*(2*log10(Cn[row])));        /**/
 column++;                              /**/
 }
for(row=0;row<20;row++)
 {
 moveto(300,convert_y(70));
 for(column=0;column<450;column++)
        {
        w=(0.02*column)+1;
        p=300*log10(w)+300;
        Omega=(int)p;
        Gamma=Value[row][column];
        if(Gamma>430)break;
        lineto(Omega,convert_y(Gamma));
        if(w>10)break;
        }
}
```

Listing 4.11 ของการแสดงการพล็อตอันดับ

และในส่วนของการทำงานของส่วนโปรแกมในฟังก์ชันต่อไปก็คือการแสดงผลว่า อันดับของฟิลเตอร์ที่เหมาะสมควรจะมีค่าเท่าใด โดยเกิดจากการลากเส้นจากแกน Mp ไปตัดแกน Ms และไปตัดแกน γ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 24 แสดงโหมดภาพของโนโมกราฟในการแสดงค่าอันดับ

ค่า Ω ที่ได้จะถูกพล็อตบนแกน Ω การที่จะทราบค่าออร์เดอร์ได้นั้นจะเกิดจากการลากเส้นตั้งฉากแกน γ และแกน Ω การที่ลากเส้นที่ตั้งฉากแกน γ นั้นเราจะใช้วิธีการของการนำค่า X ของตำแหน่ง Ω ไปเป็นจุดปลายของเส้นที่ตั้งฉากกับแกน γ

และการลากเส้นที่ตั้งฉากกับแกน Ω จะใช้วิธีการนำค่า Y ของตำแหน่ง γ ไปเป็นจุดปลายของเส้นที่ตั้งฉากกับแกน Ω

การทำงานของโปรแกรมส่วนล่างนี้จะเป็นลักษณะที่เป็นการหาค่าทางแกน Y ของแกน Mp โดยสมการเดิมคือ

$$y_1 = Log(10^{M_P(dB)} - 1) + 0.25$$

ค่าของ y1 ด้านบนจะต้องมีการทำให้อยู่ในตำแหน่ง pixel ที่เป็นจำนวนเต็ม โดยใช้สมการคือ

$$Point_M_P = (int)(y_1 * 30) + 70$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่า Point_Mpที่ได้นั้น จะเป็นค่าตำแหน่งของตำแหน่งบนแกน Y ที่ใช้ในการพล็อตกราฟจริงและมีการบวกเพิ่มอีก 30 เพื่อให้เกิดการ Shift ทางแกน Y ขึ้นไปอีก 30 จุด

ในส่วนของโปรแกรมจะเขียนได้เป็น

```
/* Plot Maximum Passbamd Gain */
    a=log10(pow(10,Max_Pass/10)-1)+5;  /* formula plot axial Mp */
    b=a*30+70;                         /* convert to coordinate (x,y) */
    Point_Mp=(int)b;
```

Listing 4.12 ส่วนของการเปลี่ยนค่า Mp ที่ป้อนให้อยู่ในรูปของ pixel

การทำงานของโปรแกรมส่วนล่างนี้จะเป็นลักษณะที่เป็นการหาค่าทางแกน Y ของแกน Ms โดยสมการเดิมคือ

$$y_3 = 0.5 * Log(10^{M_s/10} - 1) + 0.25$$

ค่าของ y3 ด้านบนจะต้องมีการทำให้อยู่ในตำแหน่ง pixel ที่เป็นจำนวนเต็ม โดยใช้สมการคือ

$$Point_M_s = (int)(y_3 * 30) + 70$$

ค่า Point_Msที่ได้นั้น จะเป็นค่าตำแหน่งของตำแหน่งบนแกน Y ที่ใช้ในการพล็อตกราฟจริงและมีการบวกเพิ่มอีก 30 เพื่อให้เกิดการ Shift ทางแกน Y ขึ้นไปอีก 30 จุด

ในส่วนของโปรแกรมจะเขียนได้เป็น

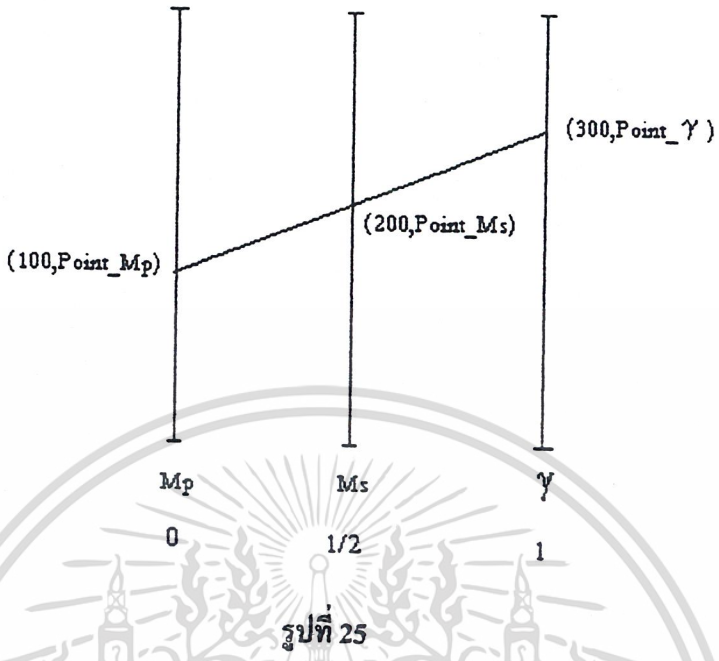
```
/* Plot Minimum Stopband Gain */
    aa=0.5*log10(pow(10,Min_Stop/10)-1)+2.5;
    bb=aa*30+70;
    Point_Ms=(int)bb;
```

Listing 4.13 ส่วนของการเปลี่ยนค่า Ms ที่ป้อนให้อยู่ในรูปของ pixel

การทำงานของโปรแกรมส่วนล่างนี้จะเป็นลักษณะที่เป็นการหาค่าทางแกน Y ของแกน γ โดยจากสมการเส้นตรงคือเมื่อเราทราบจุด 2 จุดเราสามารถที่จะหาค่าของจุดที่ 3 ได้

```
/* Plot Gamma  */
    AA=(bb-b)/100;
    BB=(200*AA)+b;
    Point_r=(int)BB;
/* Plot Omega */
    w=300*log10(Freq_Stop/Freq_Pass)+300;
    Point_w=(int)w;
/* Plot Point Mp-r  */
    line(100,convert_y(Point_Mp),300,convert_y(Point_r));
```

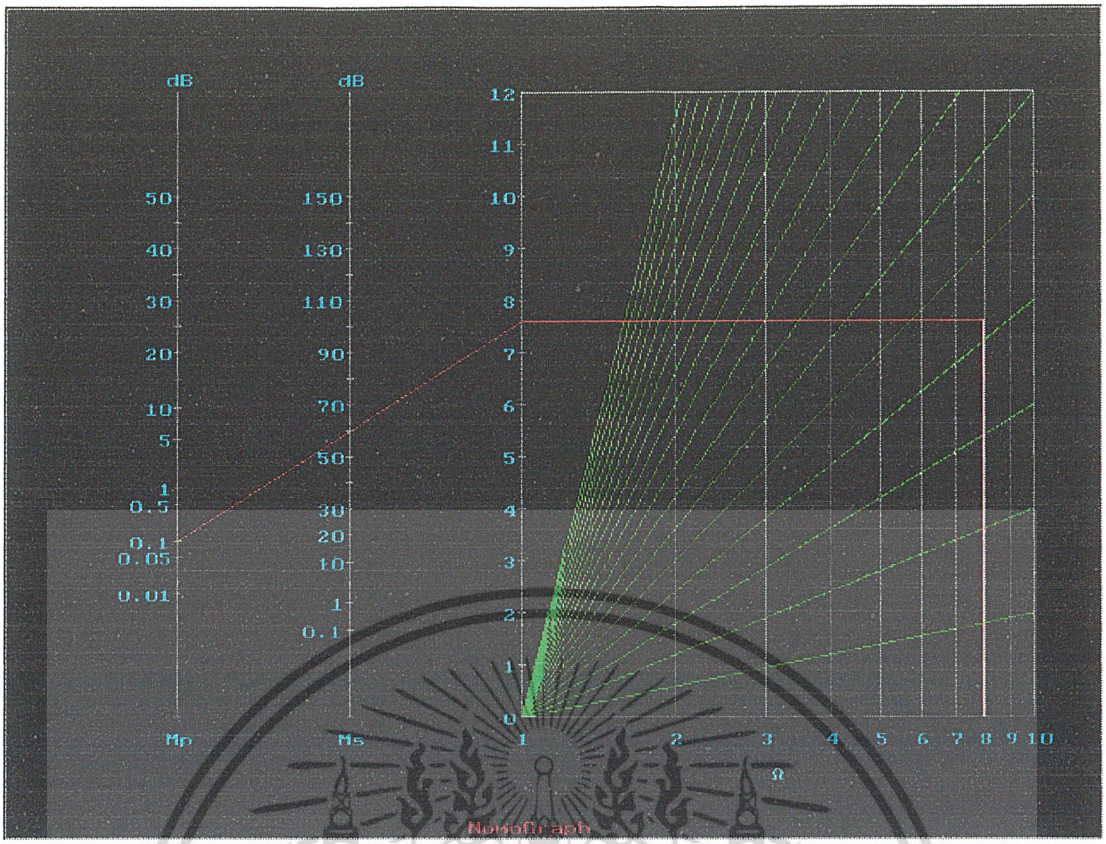
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 25

```
/* Plot to find Order  */
    line(300,convert_y(Point_r),Point_w,convert_y(Point_r));
    line(Point_w,convert_y(70),Point_w,convert_y(Point_r));
/* Return value out to function Main */
   *point_x=Point_w;
   *point_y=Point_r;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

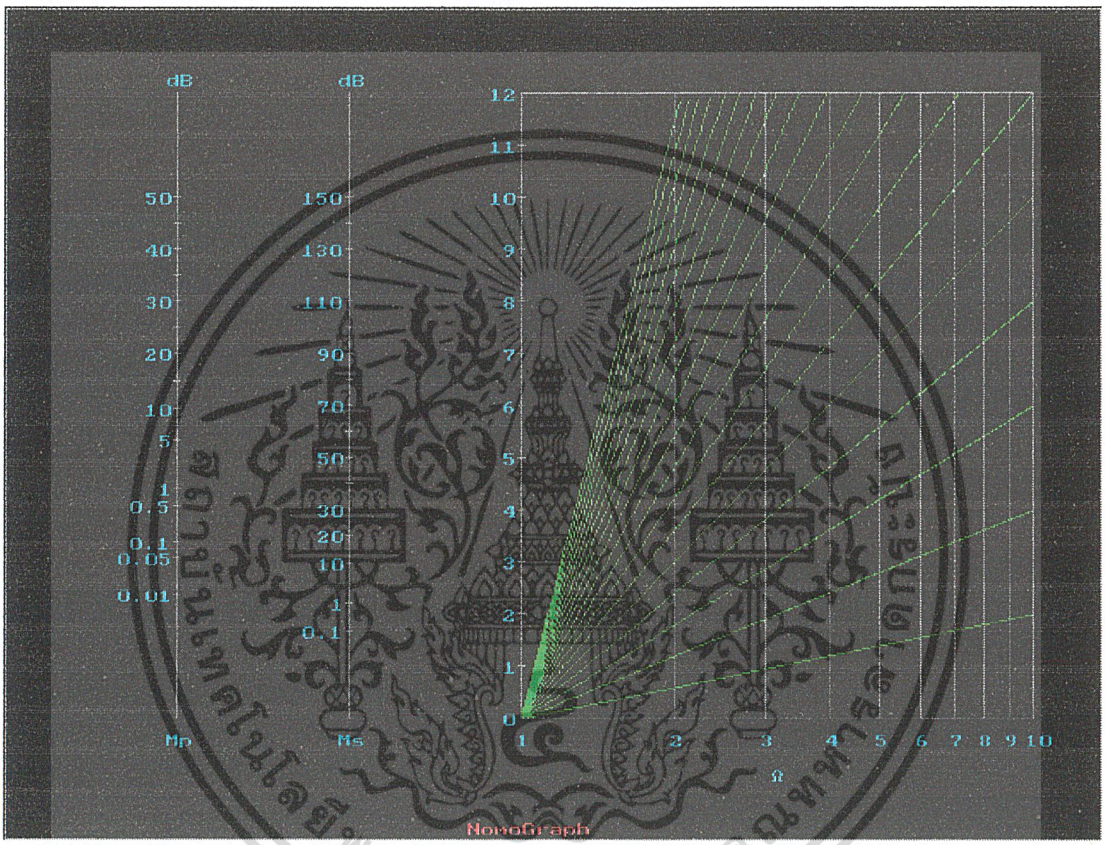


รูปที่ 26 แสดงรูปโนโมกราฟในการแสดงค่าอันดับที่สมบูรณ์

จากรูปด้านบนเมื่อป้อนค่าพารามิเตอร์ Mp=0.1 dB, Ms=60 dB, Fs=250 Hz, Fp=2000 Hz และเมื่อทำการเลือกชนิดของฟิลเตอร์คือ Butterworth มีค่า Alpha = infinity(Alpha ประมาน 10^8)

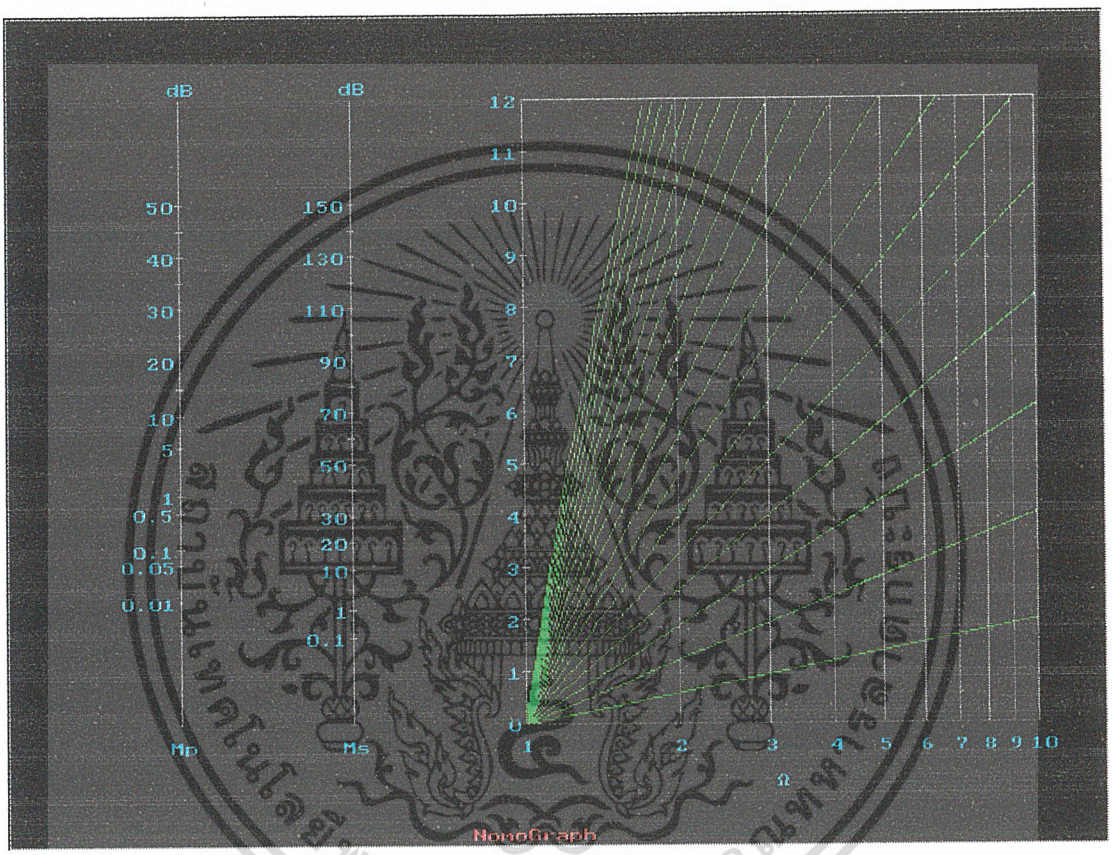
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NOMOGRAPH ที่ได้จากหน้าจอ MONITOR



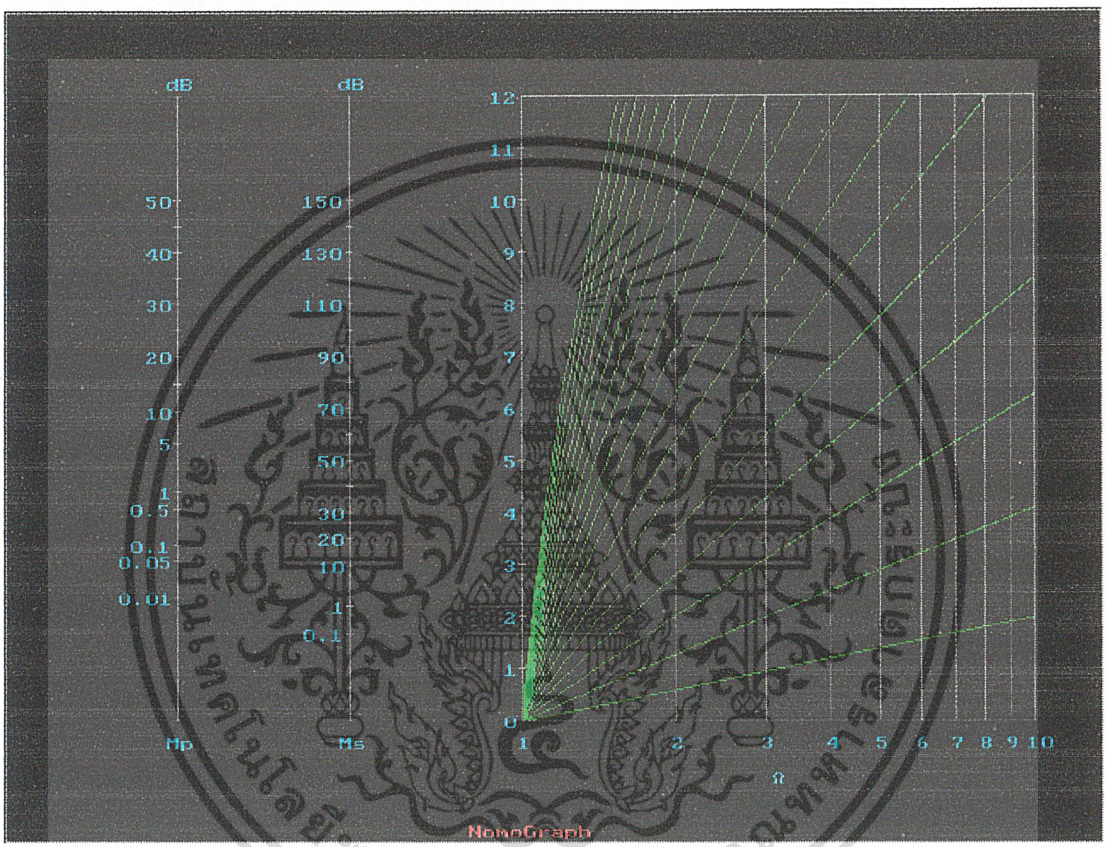
รูปที่ 1. Nomograph ของ Butterworth (∞) filters.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



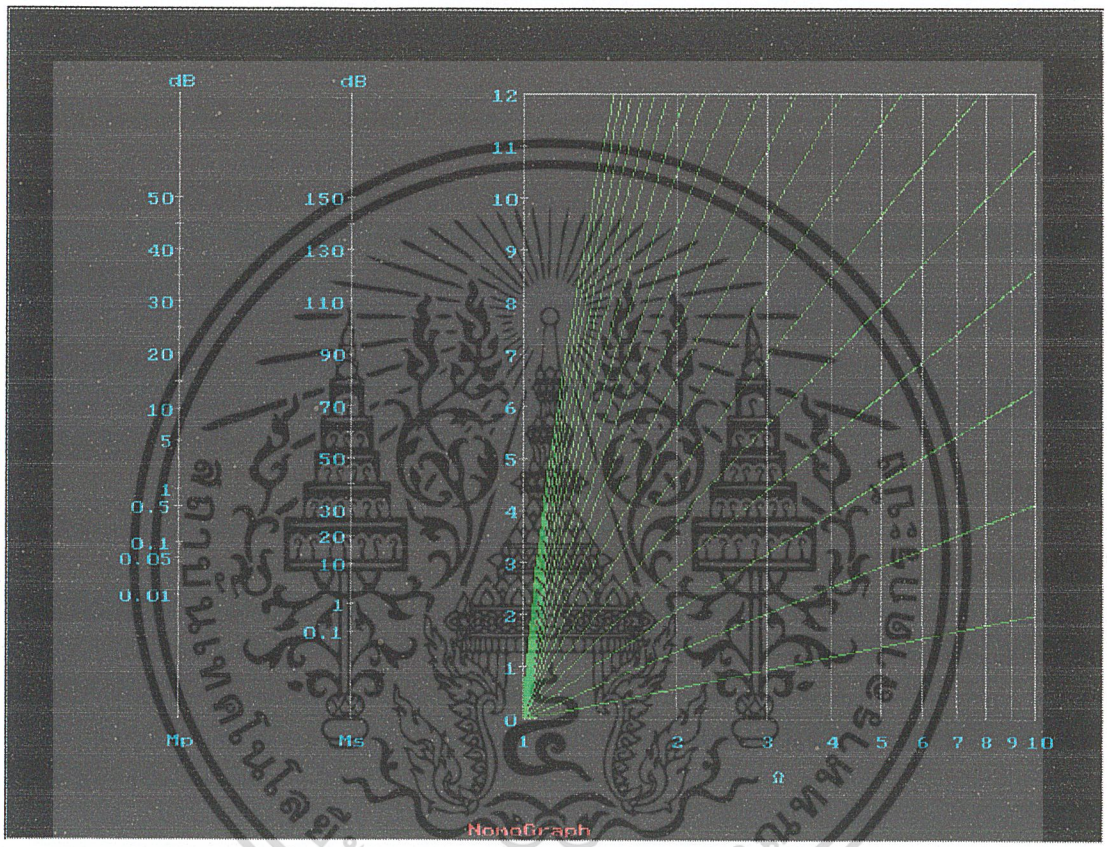
รูปที่ 2. Nomograph ของ 10th MAL (10) filters.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



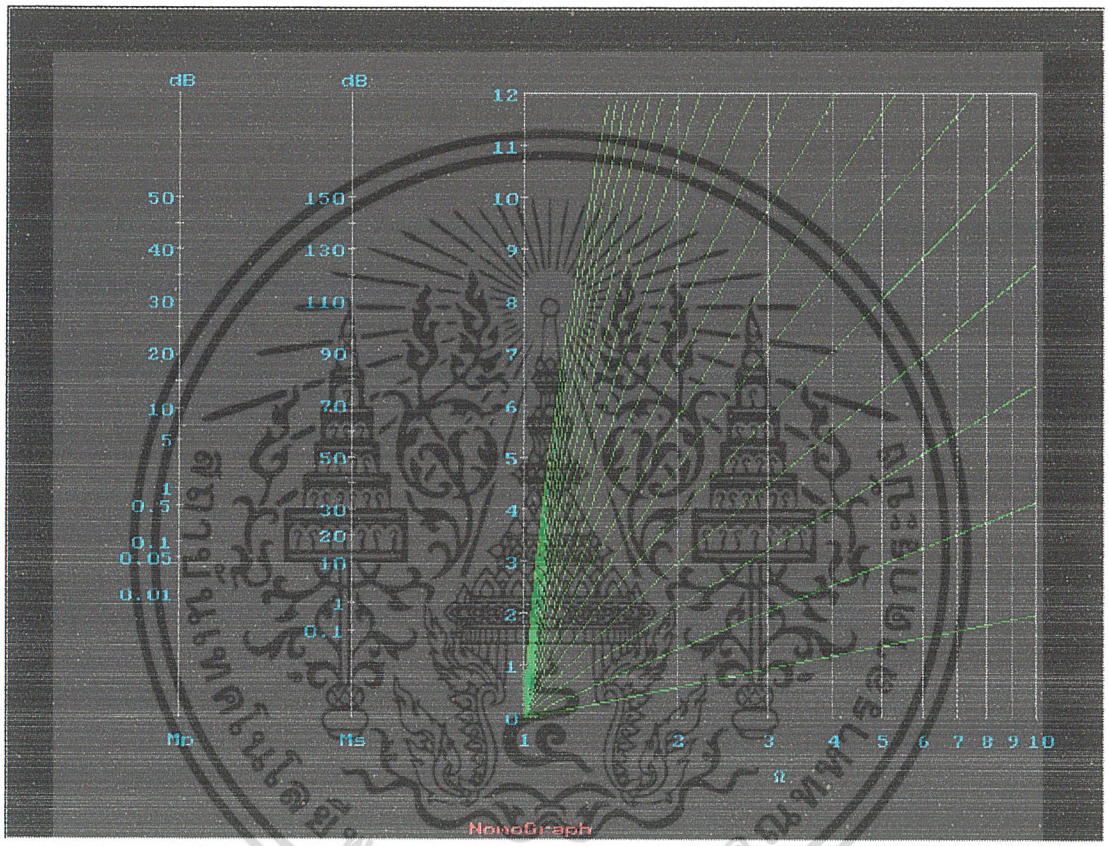
รูปที่ 3. Nomograph ของ 3rd MAL (3) filters.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



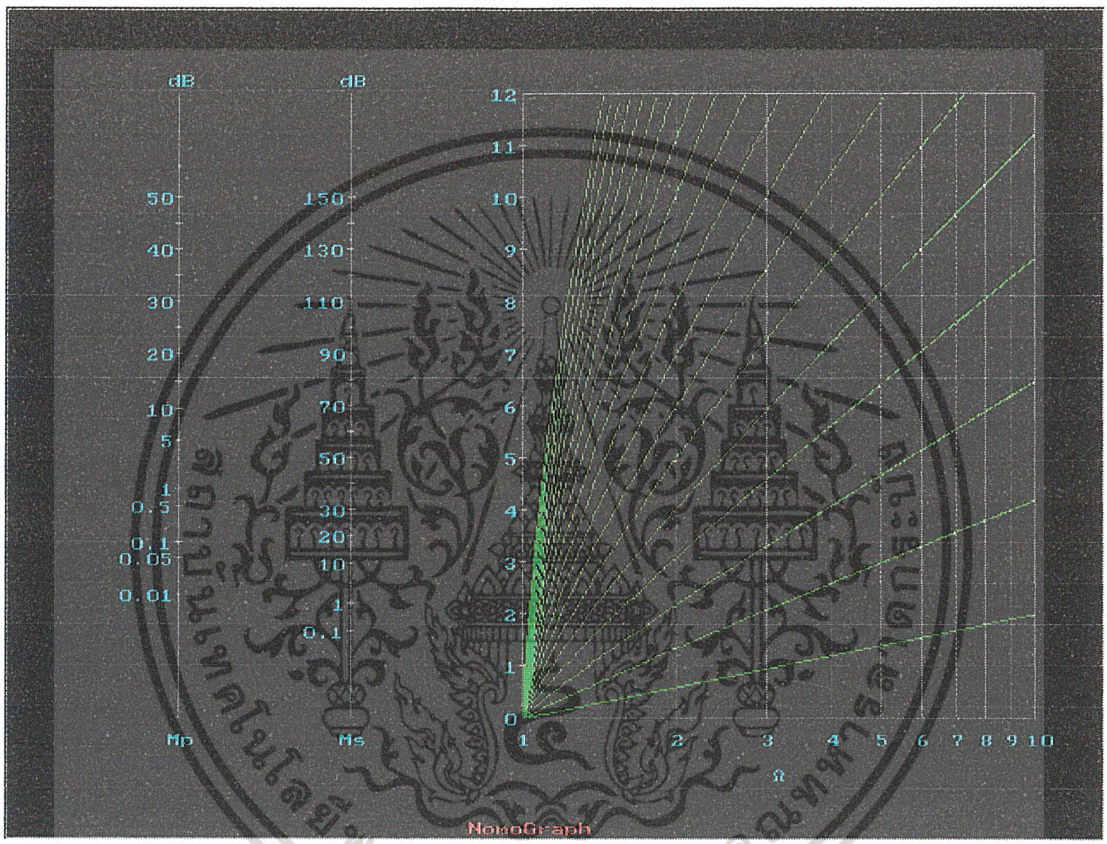
รูปที่ 4. Nomograph ของ 2nd MAL (2) filters.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



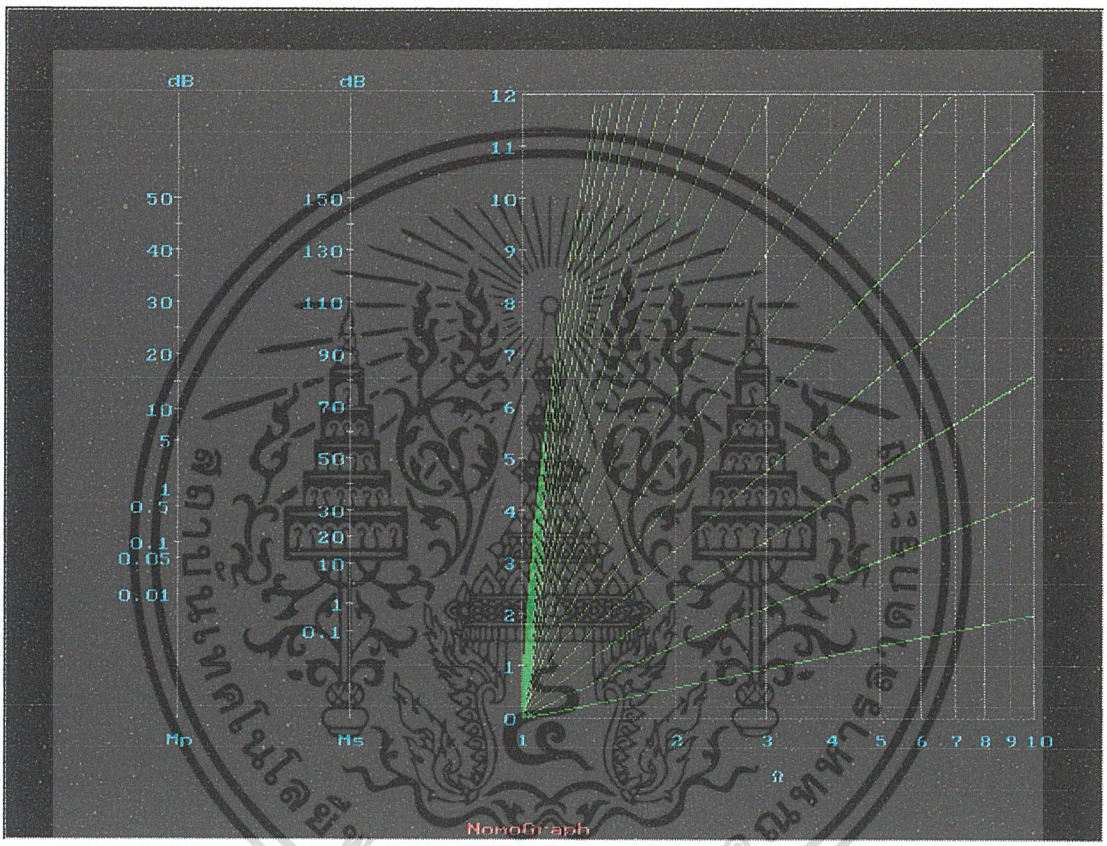
รูปที่ 5. Nomograph ของ 1.5 ultraspherical (1.5) filters.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



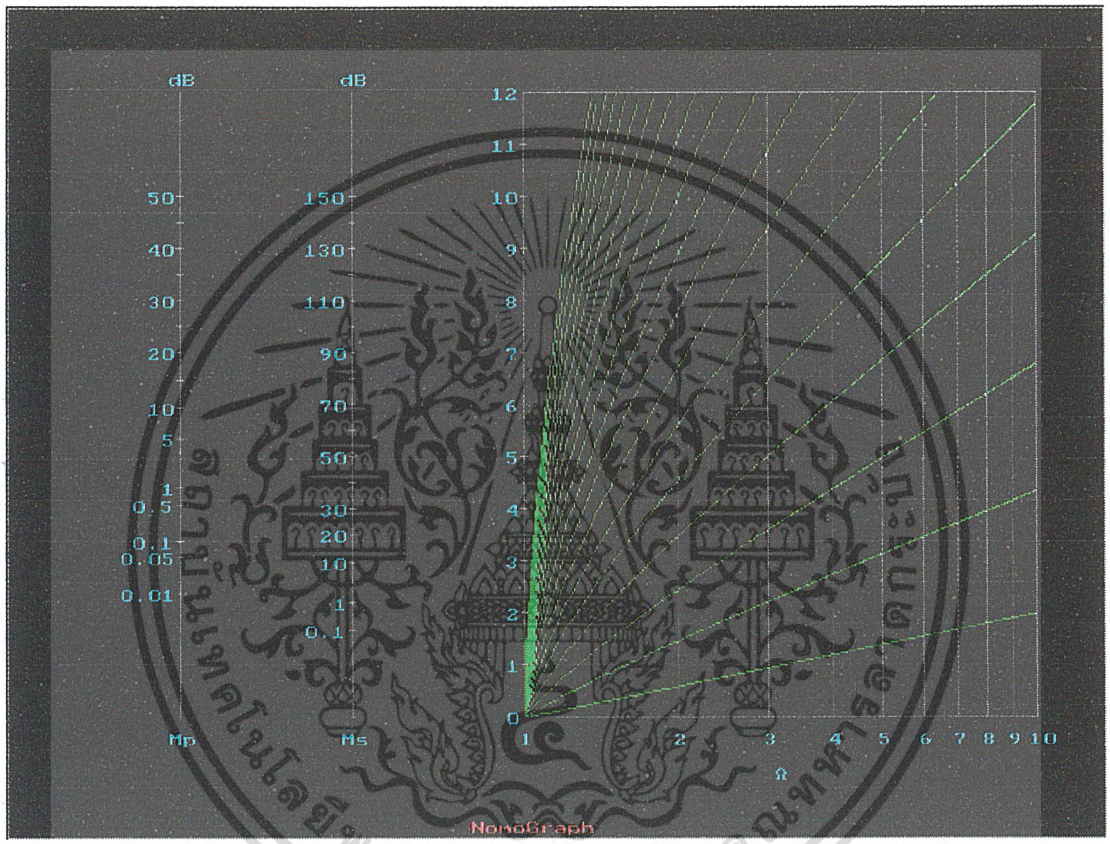
รูปที่ 6. Nomograph ของ 1st MAL (1) filters.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



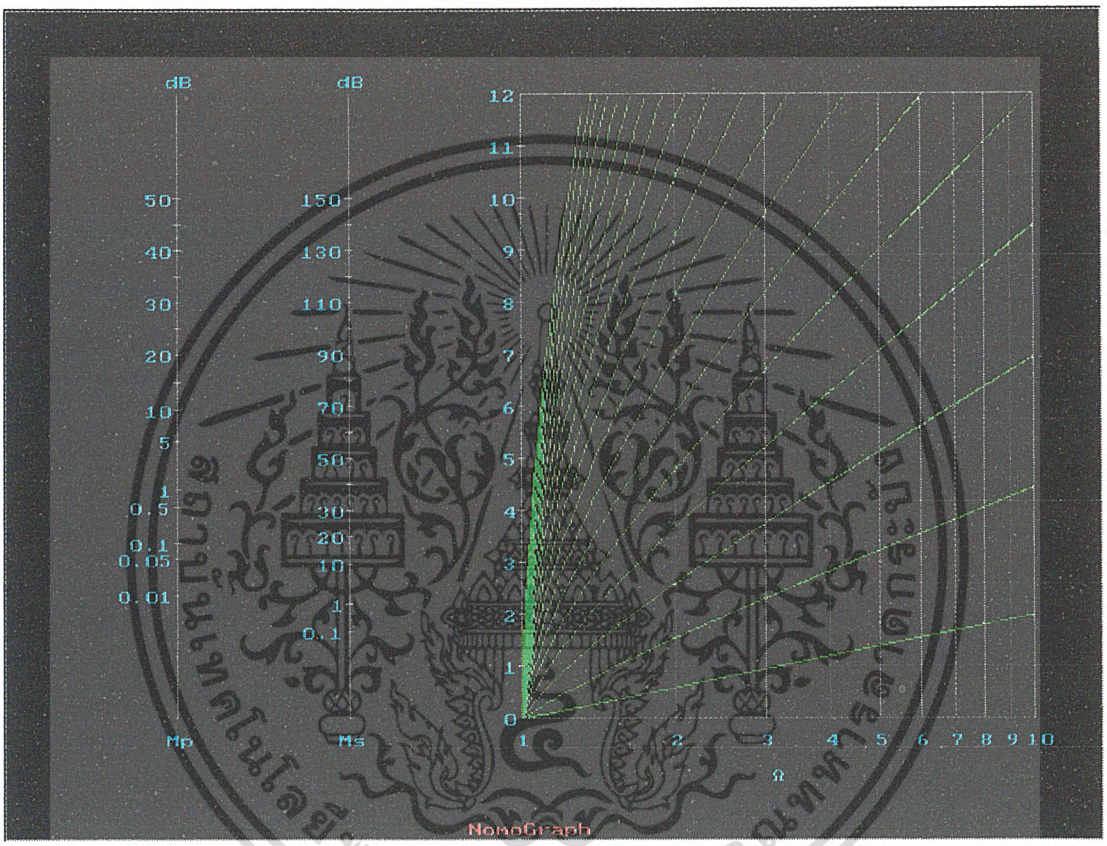
รูปที่ 7. Nomograph ของ Chebyshev (0.5) filters ชนิดที่ 2.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



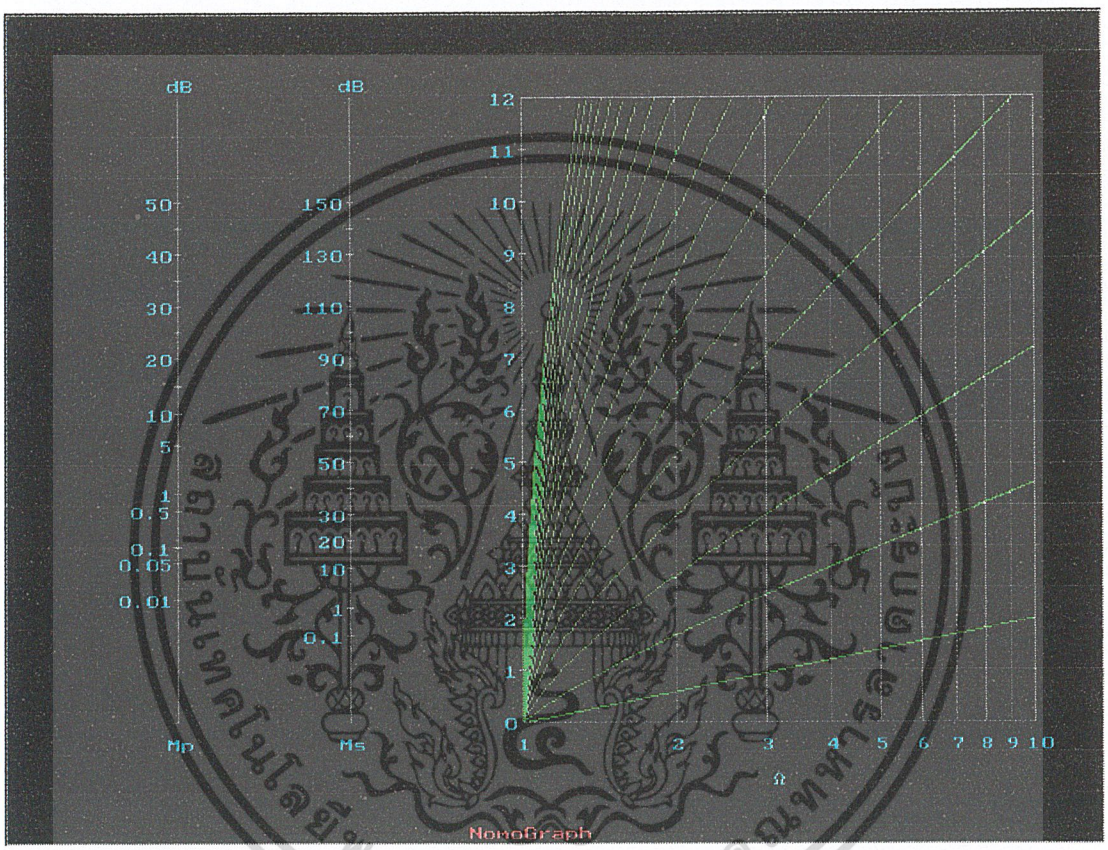
รูปที่ 8. Nomograph ของ Legendre (0) filters.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



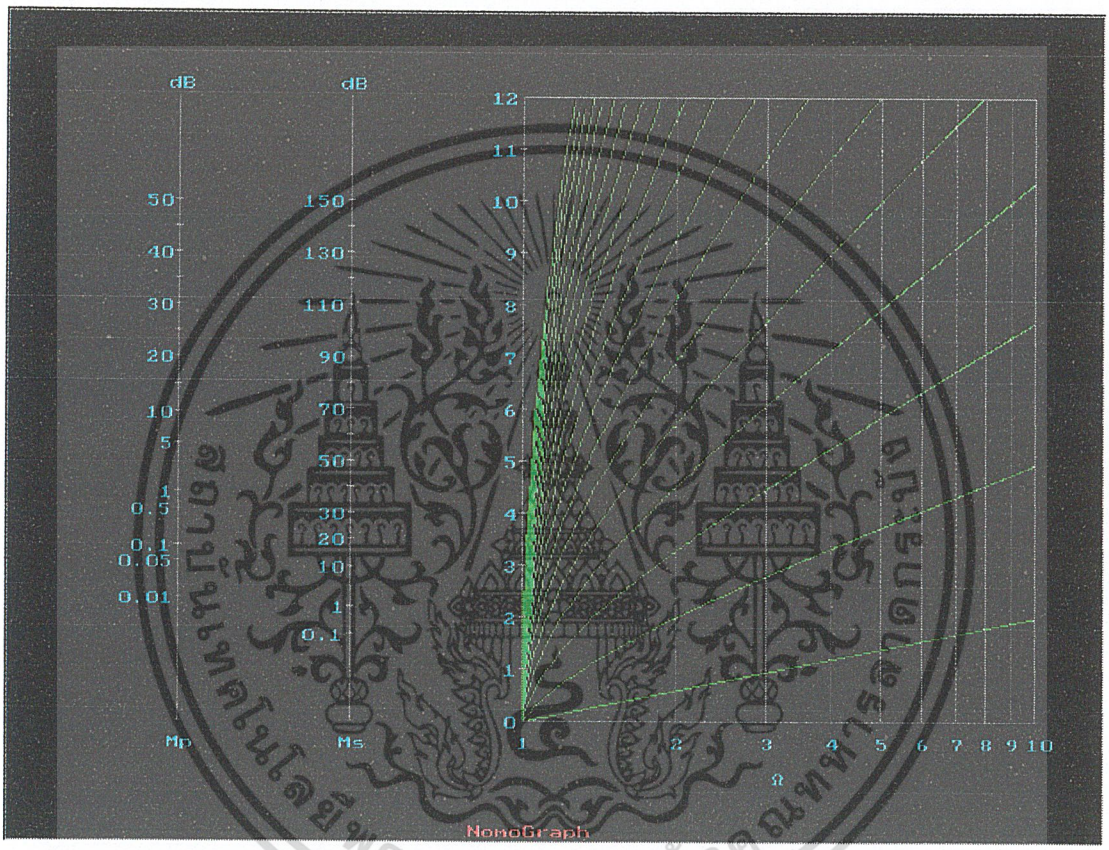
รูปที่ 9. Nomograph ของ -0.25 ultraspherical (-0.25) filters.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



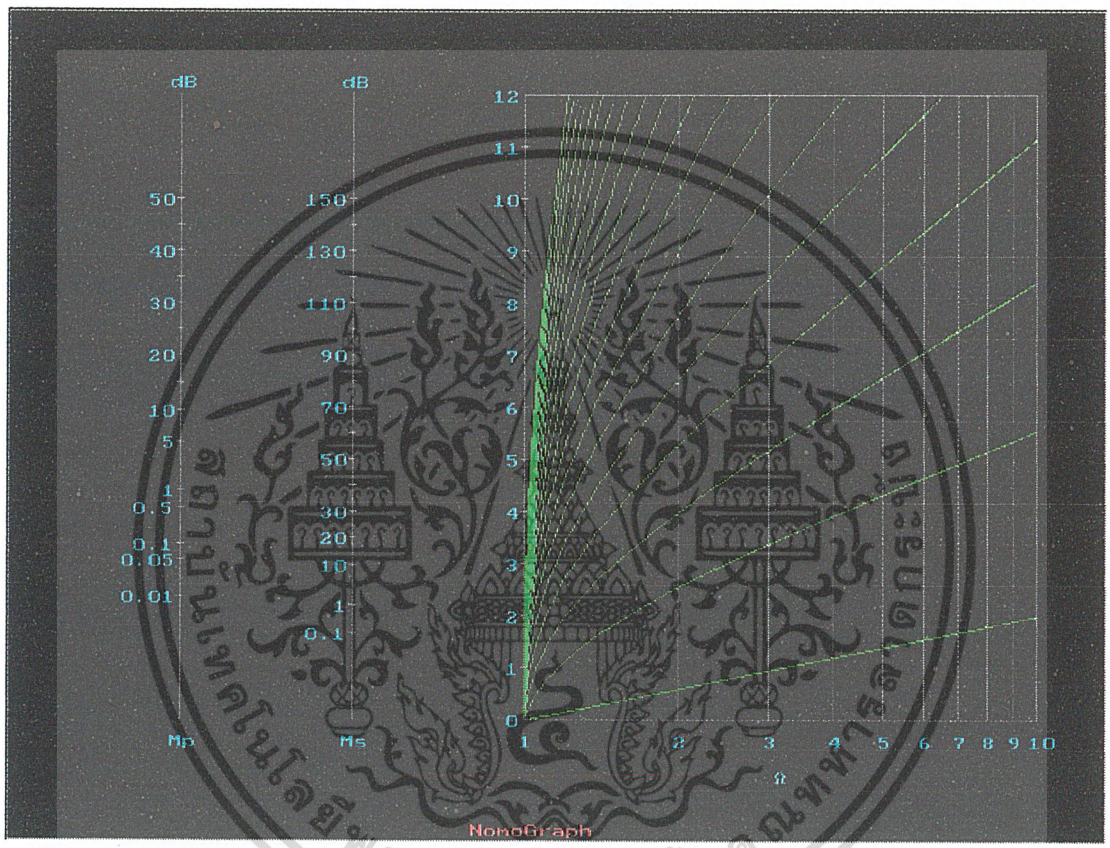
รูปที่ 10. Nomograph ของ Chebyshev (-0.5) filters ชนิดที่ 1 (equal ripple).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



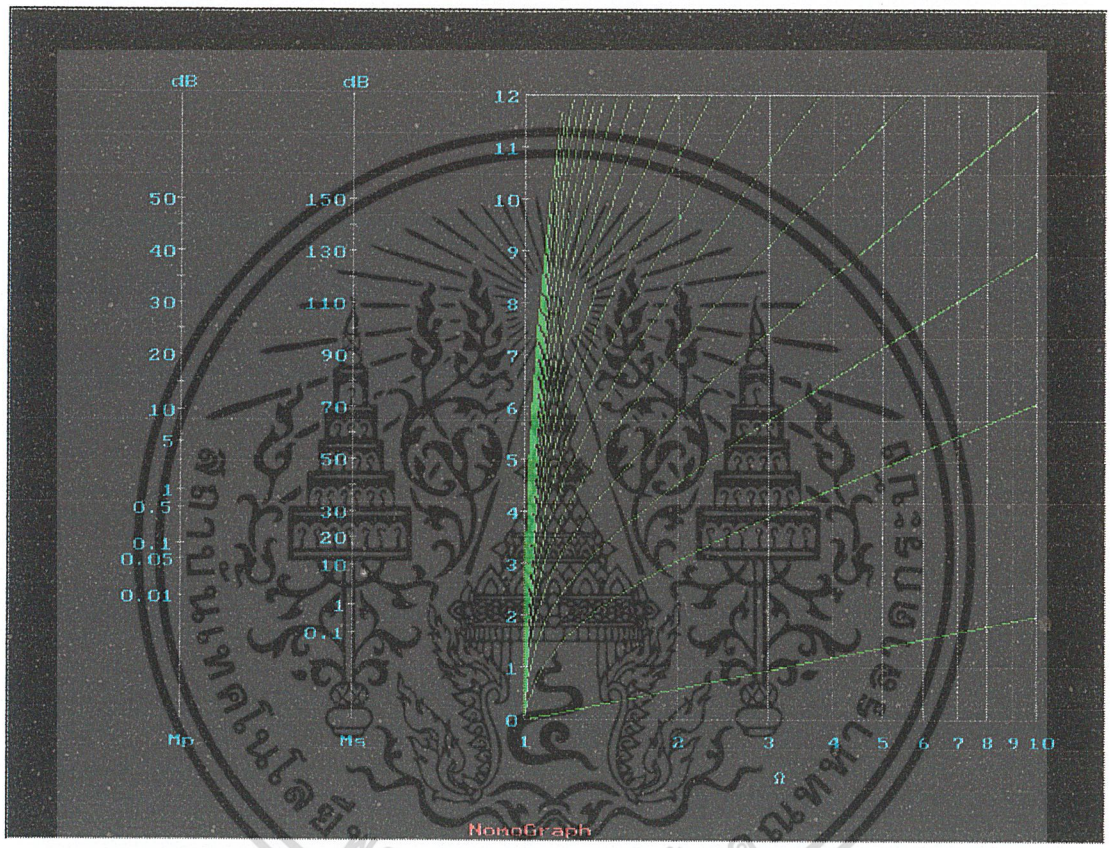
รูปที่ 11. Nomograph ของ -0.75 ultraspherical (-0.75) filters.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



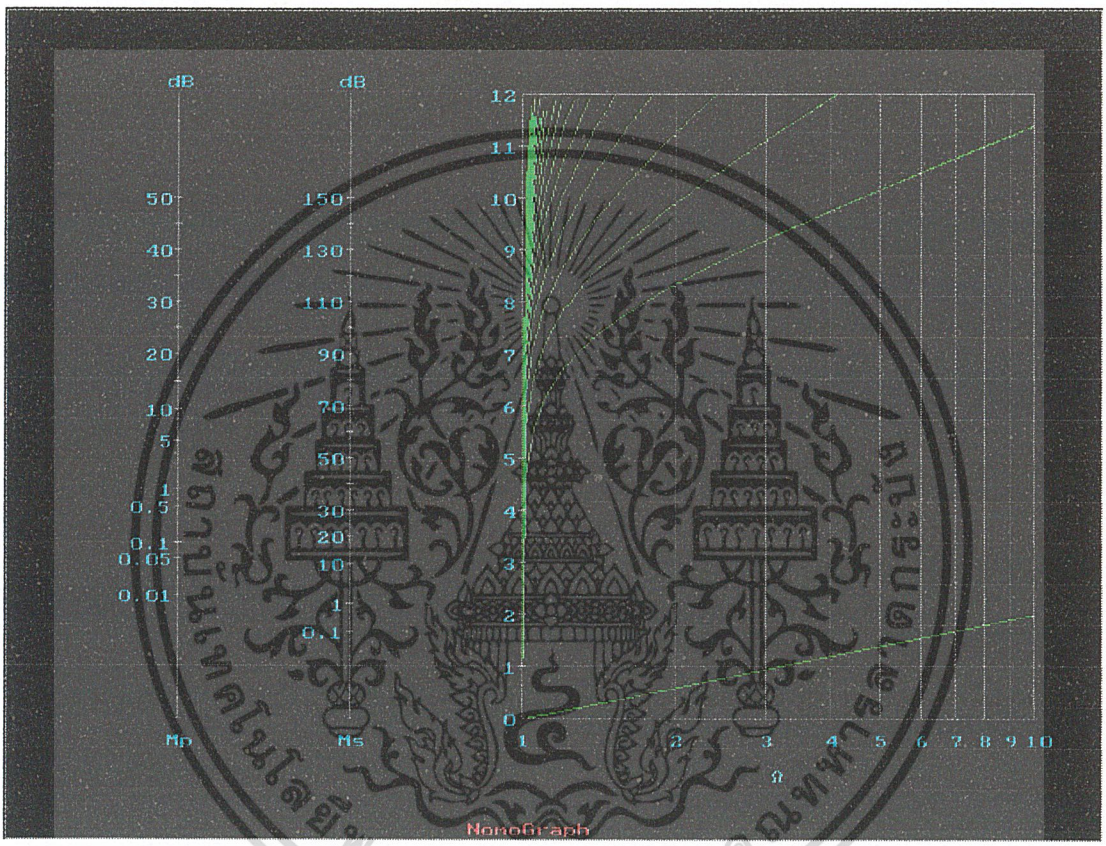
รูปที่ 12. Nomograph ของ -0.90 ultraspherical (-0.90) filters.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 13. Nomograph ของ -0.95 ultraspherical (-0.95) filters.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 14. Nomograph ของ -0.9999 ultraspherical (-0.9999) filters

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาและอุปสรรค

ปัญหาที่พบขณะทำโครงงานคือสมการที่ใช้ในการศึกษาในการทำปริญญานิพนธ์นี้ เป็นคณิตศาสตร์ชั้นสูงที่ศึกษาในระดับปริญญาโท จึงมีความสับซ้อนในสมการมาก ทำให้ไม่สามารถศึกษาให้เข้าใจอย่างท่อแท้ และอุปสรรคที่เกิดขึ้นอีกประการหนึ่งก็คือ คณะผู้จัดทำโครงงานไม่มีพื้นฐานในการเขียนโปรแกรมคอมพิวเตอร์ด้วยภาษาซีมาก่อน จึงทำให้โครงงานไม่มีความสมบูรณ์เท่าที่ควร เพราะว่าคณะผู้จัดทำจะต้องใช้เวลาในการศึกษาโปรแกรมภาษาซีเป็นอันมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปและวิจารณ์

เราสามารถกล่าวโดยสรุปได้ว่าโครงงานนี้ นั้นก็คือการสร้างโปรแกรมคอมพิวเตอร์ด้วยภาษาซี ที่นำมาใช้ในการหาค่า order ของ Low-Pass filter ด้วยการป้อนค่าพารามิเตอร์ต่าง ๆ [คือ Mp (Maximum passband attenuation) ,Ms (Minimum stopband rejection) ,Fp (Frequncy passband) และ Fs (Frequency stopband)] ลงในโปรแกรมตามขอบเขต และ ข้อกำหนดที่ได้กล่าวไว้แล้วในตอนต้น และได้แสดงผลลัพธ์ในการหาค่า order ของโปรแกรมที่สร้างขึ้นมานี้ด้วย " Nomgraph " (ซึ่ง Nomograph นี้ก็คือเครื่องมืออีกชนิดหนึ่งที่สามารถใช้ในการหาค่า order ของ LPF ได้) ออกมาทางจอ monitor พร้อมทั้งได้สั่งให้โปรแกรมทำการคำนวณค่าของ order ออกมาและแสดงผลออกมาทางจอ monitor ด้วยเพื่อให้สามารถอ่านค่าของ order ได้สะดวก โดยค่า order ที่ได้ออกมานี้จะมีความถูกต้องแม่นยำสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

คู่มือการใช้งานของ PROGRAM

โปรแกรมที่คุณกำลังจะใช้นี้เขียนขึ้นมาจากภาษาซี โดยจะเป็นการออกแบบวงจรกรองความถี่ต่ำ(Low-pass filter) เพื่อหาค่าอันดับของวงจรกรองความถี่ต่ำ การใช้งานจะใช้งานได้ง่ายเพียง เราจะต้องป้อนค่าพารามิเตอร์ 4 ค่าคือ Mp,Ms,Fp,Fs และสามารถที่จะทราบอันดับที่ได้โดยใช้โนโมกราฟ(ศึกษาได้จากเนื้อหาในปริญญานิพธ์)

วิธีการติดตั้งโปรแกรม

- การใช้งานบนแผ่น

โปรแกรมสามารถที่จะใช้งานบนแผ่นได้เลยทันที โดยจะทำการ UN-WriteProtect ของแผ่นก่อน และสามารถที่จะใช้งานบนแผ่นได้โดยเลือกไดรฟ์ที่มีแผ่นโปรแกรมอยู่ แล้วทำการพิมพ์ ชื่อเพื่อเข้าโปรแกรมคือ filter

เช่น

เมื่อโปรแกรมอยู่ไดรฟ์ A: และเราได้ทำการย้ายมาอยู่ที่ไดรฟ์ A: เรียบร้อยแล้วจึงพิมพ์

A:\>filter

เพื่อเข้าโปรแกรมจะปรากฏหน้าจอแสดงชื่อโปรแกรม และกดปุ่มใดๆ เพื่อทำงานต่อไป การทำงานขั้นต่อไปก็จะสามารถที่จะสังเกตได้จากข้อความบนหน้าจอ

วิธีการป้อนค่าพารามิเตอร์

โดยจุดประสงค์ของโปรแกรมจะต้องการค่าพารามิเตอร์ 4 ค่า ของการออกแบบวงจรกรองความถี่ต่ำคือโดยค่าโดยมีพารามิเตอร์ทั้งหมด 4 ค่าคือ

1. ค่าพารามิเตอร์ **Mp** (Maximum Attenuation Pass band) หรือค่าการบั่นทอนสัญญาณมากที่สุดซึ่งโดยส่วนมากแล้วขณะที่ออกแบบค่า ที่ป้อนควรจะอยู่ในช่วง 0 dB ถึง 50 dB เพราะต้องการให้ค่า การบั่นทอนสัญญาณในแถบความถี่ที่ต้องการให้ผ่านนี้มีค่าต่ำที่สุดและควรจะมีค่าเป็นบวก(ในบางโปรแกรมก็จะมีค่าเป็นลบ)
2. ค่าพารามิเตอร์ **Ms** (Minimum Attenuation Stop band)หรือเกิดการบั่นทอนสัญญาณมากที่สุดในช่วงแถบความถี่ที่ไม่ต้องการค่าที่จะป้อนควรจะอยู่ในช่วง มากกว่า 0 dB แต่ไม่เกิน 150 dB และควรจะมีค่าเป็นบวก(ในบางโปรแกรมก็จะมีค่าเป็นลบ)
3. ค่าพารามิเตอร์ **Fp** (Frequency Pass band) คือค่าความถี่คัทออฟ โดยจะมีความสัมพันธ์กับค่า Mpคิดที่ 6 dB หรือเป็นช่วงค่าความถี่ที่ต้องการให้ผ่านมีหน่วยเป็น Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ค่าพารามิเตอร์ Fs (Frequency Stop band) คือค่าความถี่ที่ต้องการให้เกิดค่าการบั่นทอนมากที่สุด โดยจะมีความสัมพันธ์กับค่า Ms หรือเป็นช่วงค่าความถี่ที่ต้องการให้เกิดค่าการบั่นทอนมากที่สุด เช่น

เมื่อเราต้องการออกแบบวงจรโดยใช้ Butterworth filter โดยมีค่าพารามิเตอร์

Mp = 0.1 dB Ms = 60 dB Fp = 250 Hz Fs = 2000 Hz

การป้อนค่าพารามิเตอร์แสดงได้ดังรูปที่ 27

```
**********Filter - The Design Program **********
**********Nomograph 1998 Som&Oat&Sing**********Page=2

You must be in put parameter of Low-Pass Filter
By...
Value Maximum Attenuation Pass Band must be(Positive) between 0-50dB
Value Minimum Attenuation Pass Band must be(Positive) between 0-150dB
And...
Value Frequency Stop Band must'n be 10 times Frequency Pass Band

  Input Maximum Attenuation Pass Band(dB).....: 0.1
  Input Minimum Attenuation Pass Band(dB).....: 60
  Input Frequency Pass Band          (Hz).....: 250
  input Frequency Stop Band          (Hz).....: 2000
 ---------------------------------------------------------------
```

รูปที่ 27 แสดงค่าพารามิเตอร์ที่ป้อนเข้าไป

การเลือกชนิดของฟิลเตอร์

การเลือกชนิดของวงจรกรองความถี่จะมีการเลือกได้ทั้งหมด 14 ชนิด

```
Your select number of Alpha by you must be choose munber
between 1-15 !
-------------------------------------------------------------
1) Alpha=infinity     (Butterworth)
2) Alpha=10           (10-th M.A.L.)
3) Alpha=3            (3rd  M.A.L.)
4) Alpha=2            (2nd  M.A.L.)
5) Alpha=1.5          (1.5  Ultraspherical)
6) Alpha=1            (1st  M.A.L.)
7) Alpha=0.5          (2nd  Chebyshev)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
8) Alpha=0          (Legendre)
9) Alpha=-0.25      (-0.25 Ultraspherical)
10) Alpha=-0.5      (1st Chevbyshev(Equal-Riple))
11) Alpha=-0.75     (-0.75 Ultraspherical)
12) Alpha=-0.9      (-0.9  Ultraspherical)
13) Alpha=-0.95     (-0.95 Ultraspherical)
14) Alpha=-0.9999   (-0.9999 Ultraspherical)
15) Design Alpha by your!
Input number of Alpha..... :
```

Listing 1 แสดงชนิดของวงจรกรองความถี่ที่มีในโปรแกรม

จากรูปด้านบนเมื่อป้อนค่าพารามิเตอร์ Mp=0.1 dB, Ms=60 dB, Fs=250 Hz, Fp=2000 Hz และเมื่อทำการเลือกชนิดของฟิลเตอร์คือ Butterworth มีค่า Alpha = infinity(Alpha ประมาน 10^9)

```
********** FILTER - The Design Program **********
********** Nomograph 1998 Som&Oat&Sing **********Page=3

  Your select number of Alpha by you must be choose munber
  between 1-15 !
-------------------------------------------------------------
    1) Alpha=infinity  (Butterworth)
    2) Alpha=10        (10-th M.A.L.)
    3) Alpha=3         (3rd  M.A.L.)
    4) Alpha=2         (2nd  M.A.L.)
    5) Alpha=1.5       (1.5  Ultraspherical)
    6) Alpha=1         (1st  M.A.L.)
    7) Alpha=0.5       (2nd  Chebyshev)
    8) Alpha=0         (Legendre)
    9) Alpha=-0.25     (-0.25 Ultraspherical)
   10) Alpha=-0.5      (1st Chevbyshev(Equal-Riple))
   11) Alpha=-0.75     (-0.75 Ultraspherical)
   12) Alpha=-0.9      (-0.9  Ultraspherical)
   13) Alpha=-0.95     (-0.95 Ultraspherical)
   14) Alpha=-0.9999   (-0.9999 Ultraspherical)
   15) Design Alpha by your!

    Input number of Alpha..... : 1
```

รูปที่ 28 เมื่อทำการเลือกชนิดของ ฟิลเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยตัวอย่างเมื่อทำการป้อนค่าพารามิเตอร์เข้าไปก็จะมีการสรุปค่าพารามิเตอร์ที่ป้อนเข้าไป และทำการแสดงโนโมกราฟซึ่งมีความสัมพันธ์กับพารามิเตอร์ที่ป้อนเข้าไป และทราบค่าอันดับของวงจรกรองความถี่ที่เราต้องการออกแบบ

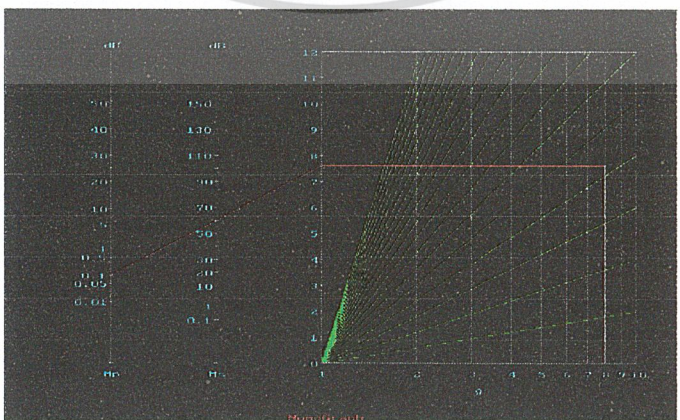
โดยวิธีการออกแบบด้วยคอมพิวเตอร์วิธีนี้เราสามารถที่จะเปรียบเทียบค่าพารามิเตอร์เดียวกันกับวงจรกรองความถี่ต่ำชนิดอื่นๆ ได้ รูปที่ 32

```
********** FILTER - The Design Program **********
********** Nomograph  1998  Som&Oat&Sing **********Page=4

                SUMARY SPECIFY RESPONSE

     Approximation n-th order Low-Pass..Butterworth filter

          Filter impelmentation is      :   Analog
          Filter Alpha is               :   Infinity
          Passband gain            (dB):    0.10
          Stopband gain            (dB):   60.00
          Passband frequency       (Hz):  250.00
          Stopband frequency       (Hz): 2000.00
```

รูปที่ 29 แสดงการสรุปค่าพารามิเตอร์ของ ฟิลเตอร์

การแสดงค่าอันดับที่ได้

การแสดงค่าอันดับที่ได้จะแสดงโดยใช้การแสดงผลในโหมดกราฟฟิกในภาษาซี ภาพการแสดงอันดับที่ได้โดยการป้อนค่าพารามิเตอร์ดังรูปที่ 30



รูปที่ 30 แสดงการPLOT เพื่อหาค่า Order ของฟิลเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
********** FILTER - The Design Program **********
********** Nomograph 1998 Som&Oat&Sing **********Page=4

                SUMARY SPECIFY RESPONSE


        Approximation n-th order Low-Pass..Butterworth filter

          Filter impelmentation is    :  Analog
          Filter Alpha is             :  Infinity
          Passband gain          (dB):  0.10
          Stopband gain          (dB):  60.00
          Passband frequency     (Hz):  250.00
          Stopband frequency     (Hz):  2000.00
          Order filter is             :  5
```

รูปที่ 31 แสดงการสรุปค่าพารามิเตอร์ของ ฟิลเตอร์ และค่า *Order* ที่ได้รับ

```
********** FILTER - The Design Program **********
********** Nomograph 1998 Som&Oat&Sing **********Page=5
 Maximum pass band attenuation(dB) : 0.1000
 Minimum pass band Attenuation(dB) : 60.0000
 Frequency pass band          (Hz) : 250.0000
 Frequency stop band          (Hz) : 2000.0000
          Butterworth   Alpha = 1000000000.0000   Order=5
          10th M.A.L.   Alpha = 10.0000           Order=5
          3rd  M.A.L.   Alpha = 3.0000            Order=4
          2nd  M.A.L.   Alpha = 2.0000            Order=4
    1.5  Ultraspherical Alpha = 1.5000            Order=4
          1st  M.A.L.   Alpha = 1.0000            Order=4
       2nd  Cheybyshev  Alpha = 0.5000            Order=4
            Legendre    Alpha = 0.0000            Order=4
   -0.25 Ultraspherical Alpha = -0.2500           Order=4
1st  Cheybyshev(Equal-Riple)  Alpha = -0.5000     Order=4
   -0.75 Ultraspherical Alpha = -0.7500           Order=4
   -0.9  Ultraspherical Alpha = -0.9000           Order=3
   -0.95 Ultraspherical Alpha = -0.9500           Order=3
  -0.9999 Ultraspherical Alpha = -0.9999          Order=2

    Press Y or y to continue if not press any key to EXIT :
```

รูปที่ 32 แสดงการสรุปค่าพารามิเตอร์ของ ฟิลเตอร์ และการเปรียบเทียบกับฟิลเตอร์ชนิดอื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคภนวก ข.

LISTING PROGRAM

โมดูล FILTER . C

```
/*****************************************************
*          ------------------------------           *
* ------- Copyright (c) 1997 by Som&Oat&Sing------- *
*          ------------------------------           *
*****************************************************/
#include <conio.h>
#include <stdio.h>
#include <math.h>
#include <in_param.c>
#include <utility.c>
void  Text_Windows(int x1,int y1,int x2,int y2);
void  Display_Header(int Page);
void  Title1(void);
void  Title2(void);

/*****************************************************
  Function MAIN()
*****************************************************/
void main(void)
{
    int Order,n,Array_Order[15];
    float Mp,Ms,Fp,Fs,Alpha,Array_Param[4];
    float Array_Alpha[15]={1000000000,10,3,2,1.5,1,0.5,0,-0.25,-0.5,-0.75
    ,-0.9,-0.95,-0.9999};
    char ans_cont,ans_compare;
    char Name_Filter[15][40]={"Butterworth",
                              "10th M.A.L.",
                              "3rd  M.A.L.",
                              "2nd  M.A.L.",
                              "1.5  Ultrasphperical",
                              "1st  M.A.L.",
                              "2nd  Cheybyshev",
                              "Legendre",
                              "-0.25 Ultrasphperical",
                              "1st  Cheybyshev(Equal-Riple)",
                              "-0.75 Ultrasphperical",
                              "-0.9  Ultrasphperical",
                              "-0.95 Ultrasphperical",
                              "-0.9999 Ultrasphperical"};
    FILE *fin;
clrscr();
    Title1();
    getch();
    Display_Header(1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Title2();
    getch();
    do{
    Input_Params_Filter(&Mp,&Ms,&Fp,&Fs);
    Utility(Mp,Ms,Fp,Fs);
    textcolor(LIGHTGRAY);
    printf("You want to compare another filter(yes/no) :");
    ans_compare=getch();
   if(ans_compare=='y')
   {
    printf("yes\n");
    fin=fopen("params.dat","r");
    for(n=0;n<4;n++)
    fscanf(fin,"%f",&Array_Param[n]);
    fclose(fin);
    for(n=0;n<14;n++)
        {
        Alpha=Array_Alpha[n];
        Compare(&Alpha,&Order);
        Array_Order[n]=Order;
        }
        clrscr();
        Display_Header(5);
        printf("\n\n");
        printf("    Maximum pass band attenuation(dB) :
%.4f\n",Array_Param[0]);
        printf("    Minimum pass band Attenuation(dB) :
%.4f\n",Array_Param[1]);
        printf("    Frequency pass band      (Hz) : %.4f\n",Array_Param[2]);
        printf("    Frequency stop band      (Hz) : %.4f\n",Array_Param[3]);
        fclose(fin);
    for(n=0;n<14;n++)
        printf("%30s   Alpha = %-15.4f
Order=%d\n",Name_Filter[n],Array_Alpha[n],Array_Order[n]);
  }
  printf("\nPress Y or y to continue if not press any key to EXIT : ");
    ans_cont=getch();
    if(ans_cont=='y'||ans_cont=='Y')printf("yes");
    switch(ans_cont)
    {
     case 'y':break;
     case 'Y':break;
     default :{
              printf("no");
              printf("\nThank you choose NOMOGRAPH to design filter\n");
              exit(0);
             }
    }
     }while(1);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*=====================================================
  Title():  draw title program first
  Prototype:  void Title();
  Return:     none
=====================================================*/
void Title1(void)
{
    Text_Windows(1,1,79,24);
    textcolor(GREEN);
    gotoxy(29,7);
    cprintf("Analog & NomoGraph");
    gotoxy(31,8);
    cprintf("Design Program");
    gotoxy(27,13);
    cprintf("Faculty of Engineering");
    gotoxy(16,14);
    cprintf("King Mungkut's Institute of Technology Ladkrabang");
    gotoxy(28,15);
    cprintf("Copyright (c)  1998");
    gotoxy(79,24);
    textcolor(LIGHTGRAY);
}

/*=====================================================
  Text_Windows() - draw windows for title
  Prototype:  void Text_Windows();
  Return:     none
=====================================================*/
void Text_Windows(int x1,int y1,int x2,int y2)
{
 int x,y;
 clrscr();
for(y=y1+1;y<y2;y++)
{
   for(x=x1+1;x<x2;x++)
   {
     gotoxy(x,y);
    putch(0x20);
   }
   gotoxy(x1,y);putch(0xba);
   gotoxy(x2,y);putch(0xba);
}
gotoxy(x=x1+1,y1);
for(x=x1+1;x<x2;x++)putch(0xcd); /* Horizontal line */
gotoxy(x1+1,y2);
for(x=x1+1;x<x2;x++)putch(0xcd); /* Horizontal line */
gotoxy(x1,y1);putch(0xc9);       /* Corner line */
gotoxy(x2,y1);putch(0xbb);
gotoxy(x1,y2);putch(0xc8);
gotoxy(x2,y2);putch(0xbc);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
/*=====================================================
  Select_type_Filter() - displays and select type Filter
  Prototype:  int Select_type_Filter(void);
  Return   : none
=====================================================*/
void Title2(void)
{
 printf("\n");
 printf("   This Program use to designing Low-Pass Filter will determine the
number of\n");
 printf("   the n-th order transferfunction of T(s) for any Filter is usually
required\n");
 printf("     - The maximum Pass band attenuation A(max)\n");
 printf("     - The minimum Stop band attenuation A(min)\n");
 printf("     - The frequence Pass band attenuation Fp\n");
 printf("     - The frequency Stop band attenuation Fs\n");
 printf("\n");
 printf("              Approximation  n-th  order lowpass filter\n");
 printf("          by use Ultraspherical polynomials  \n");
 printf("              The Ultraspherical use for design sever filter\n");
 printf("          by change value Alpha\n");
 printf("\n      EXAMPLE\n");
 printf("          Alpha =  0   you are design Legendre  Low-Pass Filter\n");
 printf("          Alpha = -0.5 you are design Chebyshev Low-Pass Filter\n");
 printf("          ------------------------------------------------\n");

 printf("\n\n");
}

/*=====================================================
  Display_Header() - displays a title and copyright
  Prototype:  void Display_Header(void);
  Return:     none
=====================================================*/
void Display_Header(int Page)
{
 clrscr();
 printf("*************************");
 printf(" FILTER - The Design Program ");
 printf("*************************\n");
 printf("*************************");
 printf(" Nomograph  1998  Som&Oat&Sing ");
 printf("*****************Page=%d \n",Page);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โมดูล F_SCREEN . C

```
/*************************************************
*          ------------------------------        *
* ------- Copyright (c) 1997 by Som&Oat&Sing------- *
*          ------------------------------        *
*************************************************/

#include <graphics.h>
#include <stdio.h>
#include <math.h>

int Value[20][450];

void Nomograph(int *point_x,int *point_y,float Max_Pass,float Min_Stop,float
Freq_Pass,float Freq_Stop,float Alpha);
int convert_y(int y);
float Abs(float x);
int Find_Order(int point_x,int point_y,int *Order_Ture);
void Compare(float *alpha,int *Order_Ture);

/*=======================================================
  Nomograph() - Plot response Nomograph and curve
  Prototype:  void Nomograph(void);
  Return:     none
=======================================================*/
void Nomograph(int *point_x,int *point_y,float Max_Pass,float Min_Stop,float
Freq_Pass,float Freq_Stop,float Alpha)
{
    int   Point_Mp,Point_Ms,Point_r,Point_w,Order;
    float a,b,aa,bb,AA,BB;
    float p,q,r,n,x,w;   /*variable of plot curve*/
    float Cn[20];
    int   row,column;
    int   drive=DETECT,mode=0,Omega,Gamma;

    clrscr();
/* set graphic Mode */
    initgraph(&drive,&mode,"");
/* Plot Maximum Passbamd Gain */
    a=log10(pow(10,Max_Pass/10)-1)+5;  /* formular plot axial Mp */
    b=a*30+70;                         /* convert to coordinate (x,y) */
    Point_Mp=(int)b;
/* Plot Minimum Stopband Gain */
    aa=0.5*log10(pow(10,Min_Stop/10)-1)+2.5;
    bb=aa*30+70;
    Point_Ms=(int)bb;
/* Plot Gamma  */
    AA=(bb-b)/100;
    BB=(200*AA)+b;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      Point_r=(int)BB;
 /* Plot Omega */
      w=300*log10(Freq_Stop/Freq_Pass)+300;
      Point_w=(int)w;
 /* Return value out to function Main */
     *point_x=Point_w;
     *point_y=Point_r;
     setcolor(2);
 /*Plot curve Ultraspheirical filter*/
 column=0;
 for(w=1;w<10;w=w+0.02)
   {
   for(n=2;n<20;n++)  /*caiculation and keep to 20 cell*/
         {                              /**/
         Cn[0]=1;                       /**/
         Cn[1]=w;                       /**/
         Cn[n]=((w*(2*Alpha+2*n-1)*Cn[n-1])-((n-1)*Cn[n-2]))/(2*Alpha+n);
         }
         for(row=0;row<20;row++)        /* keep 20 cell to main matrix */
         Value[row][column]=(int)(70+30*(2*log10(Cn[row])));        /**/
   column++;                            /**/
   }
 for(row=0;row<20;row++)
   {
   moveto(300,convert_y(70));
   for(column=0;column<450;column++)
         {
         w=(0.02*column)+1;
         p=300*log10(w)+300;
         Omega=(int)p;
         Gamma=Value[row][column];
         if(Gamma>430)break;
         lineto(Omega,convert_y(Gamma));
         if(w>10)break;
         }
 }

 /*end plot curve Ultraspheirical filter*/
     setcolor(12);
 /* Plot Point Mp-r  */
     line(100,convert_y(Point_Mp),300,convert_y(Point_r));
 /* Plot to find Order  */
     line(300,convert_y(Point_r),Point_w,convert_y(Point_r));
     line(Point_w,convert_y(70),Point_w,convert_y(Point_r));
     setcolor(RED);
     outtextxy((getmaxx()/2-50),470,"NomoGraph");
     setcolor(7);  /* color LIGHTGRAY */
     line(1,1,1,getmaxy());
     line(1,1,getmaxx(),1);
     line(getmaxx(),getmaxy(),1,getmaxy());
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        line(getmaxx(),getmaxy(),getmaxx(),1);
        line(100,convert_y(70),100,convert_y(430));      /*draw axil Mp*/
        line(200,convert_y(70),200,convert_y(430));      /*draw axil Ms*/
        rectangle(300,convert_y(430),600,convert_y(70)); /*draw Rectan */
        line(298,convert_y(100),302,convert_y(100));
        line(298,convert_y(130),302,convert_y(130));
        line(298,convert_y(160),302,convert_y(160));
        line(298,convert_y(190),302,convert_y(190));
        line(298,convert_y(220),302,convert_y(220));
        line(298,convert_y(250),302,convert_y(250));
        line(298,convert_y(280),302,convert_y(280));
        line(298,convert_y(310),302,convert_y(310));
        line(298,convert_y(340),302,convert_y(340));
        line(298,convert_y(370),302,convert_y(370));
        line(298,convert_y(400),302,convert_y(400));
    /* Ms */
        line(198,convert_y(310),202,convert_y(310));
        line(198,convert_y(355),202,convert_y(355));
        line(198,convert_y(340),202,convert_y(340));
        line(198,convert_y(325),202,convert_y(325));
        line(198,convert_y(370),202,convert_y(370));
        line(198,convert_y(295),202,convert_y(295));
        line(198,convert_y(280),202,convert_y(280));
        line(198,convert_y(265),202,convert_y(265));
        line(198,convert_y(235),202,convert_y(235));
        line(198,convert_y(175),202,convert_y(175));
        line(198,convert_y(190),202,convert_y(190));
        line(198,convert_y(205),202,convert_y(205));
        line(198,convert_y(220),202,convert_y(220));
        line(198,convert_y(250),202,convert_y(250));
        line(198,convert_y(265),202,convert_y(265));
        line(198,convert_y(235),202,convert_y(235));
        line(198,convert_y(220),202,convert_y(220));
        line(198,convert_y(205),202,convert_y(205));
        line(198,convert_y(190),202,convert_y(190));
        line(198,convert_y(175),202,convert_y(175));
        line(198,convert_y(159),202,convert_y(159));
        line(198,convert_y(136),202,convert_y(136));
        line(198,convert_y(120),202,convert_y(120));
        line(198,convert_y(70),202,convert_y(70));
    /* Mp */
        line(98,convert_y(310),102,convert_y(310));
        line(98,convert_y(355),102,convert_y(355));
        line(98,convert_y(325),102,convert_y(325));
        line(98,convert_y(295),102,convert_y(295));
        line(98,convert_y(355),102,convert_y(355));
        line(98,convert_y(340),102,convert_y(340));
        line(98,convert_y(370),102,convert_y(370));
        line(98,convert_y(280),102,convert_y(280));
        line(98,convert_y(70),102,convert_y(70));
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        line(98,convert_y(264),102,convert_y(264));
        line(98,convert_y(248),102,convert_y(248));
        line(98,convert_y(230),102,convert_y(230));
        line(98,convert_y(202),102,convert_y(202));
        line(98,convert_y(192),102,convert_y(192));
        line(98,convert_y(171),102,convert_y(171));
        line(98,convert_y(162),102,convert_y(162));
        line(98,convert_y(141),102,convert_y(141));
    /* w */
        setcolor(7);  /* color LIGHTGRAY */
        line(390,convert_y(70),390,convert_y(430));
        line(443,convert_y(70),443,convert_y(430));
        line(481,convert_y(70),481,convert_y(430));
        line(510,convert_y(70),510,convert_y(430));
        line(534,convert_y(70),534,convert_y(430));
        line(554,convert_y(70),554,convert_y(430));
        line(571,convert_y(70),571,convert_y(430));
        line(586,convert_y(70),586,convert_y(430));
    /* text axil w  */
        setcolor(11);
        outtextxy(300-2,convert_y(60),"1");
        outtextxy(390-2,convert_y(60),"2");
        outtextxy(443-2,convert_y(60),"3");
        outtextxy(481-2,convert_y(60),"4");
        outtextxy(510-2,convert_y(60),"5");
        outtextxy(534-2,convert_y(60),"6");
        outtextxy(554-2,convert_y(60),"7");
        outtextxy(571-2,convert_y(60),"8");
        outtextxy(586-2,convert_y(60),"9");
        outtextxy(600-4,convert_y(60),"10");
    /* text axil r  */
        outtextxy(290,convert_y(70)-2,"0");
        outtextxy(290,convert_y(100)-2,"1");
        outtextxy(290,convert_y(130)-2,"2");
        outtextxy(290,convert_y(160)-2,"3");
        outtextxy(290,convert_y(190)-2,"4");
        outtextxy(290,convert_y(220)-2,"5");
        outtextxy(290,convert_y(250)-2,"6");
        outtextxy(290,convert_y(280)-2,"7");
        outtextxy(290,convert_y(310)-2,"8");
        outtextxy(290,convert_y(340)-2,"9");
        outtextxy(290-8,convert_y(370)-2,"10");
        outtextxy(290-8,convert_y(400)-2,"11");
        outtextxy(290-8,convert_y(430)-2,"12");
        outtextxy(450-3,convert_y(40),"ê");
    /* text axil Ms  */
        outtextxy(190-17,convert_y(370)-2,"150");
        outtextxy(190-17,convert_y(340)-2,"130");
        outtextxy(190-17,convert_y(310)-2,"110");
        outtextxy(190-8,convert_y(280)-2,"90");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    outtextxy(190-8,convert_y(250)-2,"70");
    outtextxy(190-8,convert_y(220)-2,"50");
    outtextxy(190-8,convert_y(190)-2,"30");
    outtextxy(190-8,convert_y(175)-2,"20");
    outtextxy(190-8,convert_y(159)-2,"10");
    outtextxy(190,convert_y(136)-2,"1");
    outtextxy(190-17,convert_y(120)-2,"0.1");
    outtextxy(100-6,convert_y(60),"Mp");
    outtextxy(100-6,convert_y(440),"dB");
 /* text axil Mp  */
    outtextxy(90-8,convert_y(370)-2,"50");
    outtextxy(90-8,convert_y(340)-2,"40");
    outtextxy(90-8,convert_y(310)-2,"30");
    outtextxy(90-8,convert_y(280)-2,"20");
    outtextxy(90-8,convert_y(248)-2,"10");
    outtextxy(90,convert_y(230)-2,"5");
    outtextxy(90,convert_y(202)-2,"1");
    outtextxy(90-17,convert_y(192)-2,"0.5");
    outtextxy(90-17,convert_y(171)-2,"0.1");
    outtextxy(90-24,convert_y(162)-2,"0.05");
    outtextxy(90-24,convert_y(141)-2,"0.01");
    outtextxy(200-6,convert_y(60),"Ms");
    outtextxy(200-6,convert_y(440),"dB");

    getch();              /* stop and show Nomograph */
    closegraph();

}                         /*  end of function nomograph  */
/*=====================================================
  Abs():     convert value to Absolute
  Prototype:  float Abs(float x);
  Return:     float
=====================================================*/
float Abs(float x)
{
  float n,y;
  n=x*x;
  y=sqrt(n);
  return y;
}

/*=====================================================
  convert_y() - convert point y(real) to y(plot)
  Prototype:  int Display_Opening_Screen(int y);
  Return:     int
=====================================================*/
int convert_y(int y)
{
 float ypos;
 ypos=getmaxy()-y;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  return ypos;
}
/*=====================================================
  Find_Order() - find near value in Array
  Prototype:  int Find_Order(int point_x,int point_y);
  Return:     int
=====================================================*/
int Find_Order(int point_x,int point_y,int *Order_Ture)
{ int row,column,Diff,Diff_x[20];
  int n,Order,Omega,Axial_x[20];
  float w;

 for(row=0;row<20;row++)
  {
   for(column=0;column<450;column++)
   {
    Value[row][column]=point_y-Value[row][column];
   }
  }
n=0;
 for(row=0;row<20;row++)
 {
  for(column=0;column<450;column++)
  {

   if(Abs(Value[row][n])>=Abs(Value[row][column]))
    {
     n=column;
    }
  }
  w=0.02*n+1;
  Omega=(int)(300*log10(w)+300);
  n=0;
  Axial_x[row]=Omega;
}
  for(row=0;row<20;row++)
  {
   Diff_x[row]=point_x-Axial_x[row]; /* put value diff in to Diff_x */
  }
 /* Loop check value negative and clear value to zero */
  for(row=0;row<20;row++)
  {
   if(Diff_x[row]<0)
    {
     Diff_x[row]=0;
    }
  }
n=0;
  for(row=0;row<20;row++)
  {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if(Abs(Diff_x[n])>=Abs(Diff_x[row]))
     {
      n=row;
     }
   }
   Order=n+1;
   *Order_Ture=Order;
}

/*=====================================================
 Nomograph() - Plot response Nomograph and curve
 Prototype:  void Nomograph(void);
 Return:     none
=====================================================*/
void Compare(float *alpha,int *Order_Ture)
{
 int point_x,point_y;
 float Max_Pass,Min_Stop,Freq_Pass,Freq_Stop,Alpha;
 int c;
 float Array[4];

 int Point_Mp,Point_Ms,Point_r,Point_w,Order;
 float a,b,aa,bb,AA,BB;
 float p,q,r,n,x,w;   /*variable of plot curve*/
 float Cn[20];
 int   row,column;
 int   Omega,Gamma;

 int   Diff,Diff_x[20];
 int   Axial_x[20];

 FILE *fin;


 fin=fopen("params.dat","r");
 Alpha=*alpha;
 for(c=0;c<4;c++)
 fscanf(fin,"%f",&Array[c]);
 fclose(fin);

 Max_Pass=Array[0];
 Min_Stop=Array[1];
 Freq_Pass=Array[2];
 Freq_Stop=Array[3];

/* Plot Maximum Passbamd Gain */
    a=log10(pow(10,Max_Pass/10)-1)+5;  /* formular plot axial Mp */
    b=a*30+70;                         /* convert to coordinate (x,y) */
    Point_Mp=(int)b;
/* Plot Minimum Stopband Gain */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        aa=0.5*log10(pow(10,Min_Stop/10)-1)+2.5;
        bb=aa*30+70;
        Point_Ms=(int)bb;
    /* Plot Gamma  */
        AA=(bb-b)/100;
        BB=(200*AA)+b;
        Point_r=(int)BB;
    /* Plot Omega */
        w=300*log10(Freq_Stop/Freq_Pass)+300;
        Point_w=(int)w;
    /* Return value out to function Main */
       point_x=Point_w;
       point_y=Point_r;
    /*Plot curve Ultraspheirical filter*/
    column=0;
    for(w=1;w<10;w=w+0.02)
     {
     for(n=2;n<20;n++)  /*calculation and keep to 20 cell*/
           {                              /**/
           Cn[0]=1;                       /**/
           Cn[1]=w;                       /**/
           Cn[n]=((w*(2*Alpha+2*n-1)*Cn[n-1])-((n-1)*Cn[n-2]))/(2*Alpha+n);
           }
           for(row=0;row<20;row++)          /* keep 20 cell to main matrix */
           Value[row][column]=(int)(70+30*(2*log10(Cn[row])));          /**/
     column++;                              /**/
     }
    for(row=0;row<20;row++)
    {
     for(column=0;column<450;column++)
     {
      Value[row][column]=point_y-Value[row][column];
     }
    }
   c=0;
    for(row=0;row<20;row++)
   {
     for(column=0;column<450;column++)
     {

      if(Abs(Value[row][c])>=Abs(Value[row][column]))
       {
        c=column;
       }
     }
     w=0.02*c+1;
     Omega=(int)(300*log10(w)+300);
     c=0;
     Axial_x[row]=Omega;
   }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    for(row=0;row<20;row++)
    {
     Diff_x[row]=point_x-Axial_x[row]; /* put value diff in to Diff_x */
    }
    /* Loop check value negative and clear value to zero */
    for(row=0;row<20;row++)
    {
     if(Diff_x[row]<0)
      {
       Diff_x[row]=0;
      }
    }
  c=0;
    for(row=0;row<20;row++)
    {
     if(Abs(Diff_x[c])>=Abs(Diff_x[row]))
      {
       c=row;
      }
    }
    Order=c+1;
    *Order_Ture=Order;
}
```

```
/************************************************
*          ----------------------------        *
* ------- Copyright (c) 1997 by Som&Oat&Sing------- *
*          ----------------------------        *
************************************************/
#include <conio.h>
#include <stdio.h>
#include <f_select.c>
#include <f_sum.c>
#include <f_screen.c>

void Utility(float Mp,float Ms,float Fp,float Fs)
{
    float Alpha;
    int Point_x,Point_y,Order,No_F;

    No_F=Select_Filter(&Alpha);
    clrscr();
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Show_Type_Filter(No_F,Mp,Ms,Fp,Fs,Alpha);       /*display summary */
    Nomograph(&Point_x,&Point_y,Mp,Ms,Fp,Fs,Alpha);
    Find_Order(Point_x,Point_y,&Order);
    Summary_Specify(No_F,Mp,Ms,Fp,Fs,Alpha,Order);  /*display summary
*/
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โมดูล IN_PARAM . C

```
/*************************************************
*          ----------------------------          *
* ------- Copyright (c) 1997 by Som&Oat&Sing------- *
*          ----------------------------          *
*************************************************/
#include <stdio.h>
#include <stdlib.h> /* for atof() */
#define  MAX 3

void Display_Header(int Page);
float Input_Params_Filter(float *Max_Pass,float *Min_Stop,float
*Freq_Pass,float *Freq_Stop);

struct{
      double PARA;
        }Filter_PARA[MAX];  /* define 4 array */

/*=============================================================
==
 Input_Params_Filter() - displays and input Parameter of Filter
 Prototype: float Input_Params_Filter(float *Max_Pass,float
                  *Min_Stop,float *Freq_Pass,float *Freq_Stop)
  Return   :  float
=============================================================
==*/
float Input_Params_Filter(float *Max_Pass,float *Min_Stop,float
*Freq_Pass,float *Freq_Stop)
{
    float Min,Max,Fp,Fs;
    char min[10],max[10],fs[10],fp[10];
    FILE *fout;      /* define output keep to file */
    int n=-1,count;
    fout=fopen("params.dat","w");  /* file name params */

    /* start Loop new again when occur error in recive value */
 do{
    clrscr();
    Display_Header(2);
    n++;
    printf("\n");
    printf("\n");
    printf("      Your must be in put paramter of Low-Pass Filter \n");
    printf("By...\n");
    printf(" Value Maximum Attenuation Pass Band must be(positive)
between 0-50 dB\n");
    printf(" Value Minimum Attenuation Pass Band must be(positive) between
0-150 dB\n");
    printf("\nAnd...\n");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ดัดแปลง เผยแพร่ หรือแสวงหาผลประโยชน์ทางการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งผู้ใช้งานต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    printf(" Value Frequency Stop Band must'n be 10 times Frequency Pass
Band\n");
    printf("\n");
    do{
    printf("     Input Maximum Attenuation Pass Band(dB)..... : ");gets(min);
    Filter_PARA[0].PARA=atof(min);
    if(Filter_PARA[0].PARA==0)
    printf("\a\nYou input value not correct!!....Again please\n\n");
     }while(Filter_PARA[0].PARA==0);
    do{
    printf("     Input Minimum Attenuation Stop Band(dB)..... : ");gets(max);
    Filter_PARA[1].PARA=atof(max);
    if(Filter_PARA[1].PARA==0)
    printf("\a\nYou input value not correct!!....Again please\n\n");
     }while(Filter_PARA[1].PARA==0);
    do{
    printf("     Input Frequency Pass Band(Hz)..... : ");gets(fp);
    Filter_PARA[2].PARA=atof(fp);
    if(Filter_PARA[2].PARA==0)
    printf("\a\nYou input value not correct!!....Again please\n\n");
     }while(Filter_PARA[2].PARA==0);
    do{
    printf("     Input Frequency Stop Band(Hz)..... : ");gets(fs);
    Filter_PARA[3].PARA=atof(fs);
    if(Filter_PARA[3].PARA==0)
    printf("\a\nYou input value not correct!!....Again please\n\n");
     }while(Filter_PARA[3].PARA==0);

    printf("\n");
    printf("-----------------------------------------------------------------\n");
for(count=0;count<4;count++)
    fprintf(fout,"%f\n",Filter_PARA[count].PARA);
    fclose(fout);

    Max=Filter_PARA[0].PARA;  /* define value to MAXIMUM PASS BAND */
    Min=Filter_PARA[1].PARA;
    Fp=Filter_PARA[2].PARA;
    Fs=Filter_PARA[3].PARA;

    *Max_Pass=Filter_PARA[0].PARA;  /* define value to Pointer */
    *Min_Stop=Filter_PARA[1].PARA;
    *Freq_Pass=Filter_PARA[2].PARA;
    *Freq_Stop=Filter_PARA[3].PARA;


if((Min<=1)||(Max>=50)||(Min<=1)||(Min>=150)||(Fs/Fp)>=10||(Fs/Fp)<=1
)
    printf("\a");
    if(n>=3)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            {
        printf("/***You must input Maximum Attenuation Pass Band between 0-
 50 (dB)***/\n");
        printf("/***Input Minimum Attenuation Stop Band between 0-150
 (dB)**********/\n");
        printf("/***and Input (Frequency stop band/Frequency pass
 Band)<10**********/\n");
        printf("\n/---------- Press any key to continue ----------
 /");
        getch();  /* stop */
        n=0;
              }

}while((Min<=1)||(Max>=50)||(Min<=1)||(Min>=150)||(Fs/Fp)>=10||(Fs/Fp
)<=1);
     getch();  /* stop */
 return 1;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โมดูล F_SELECT.C

```
/*************************************************
*          ----------------------------          *
* ------- Copyright (c) 1997 by Som&Oat&Sing------- *
*          ----------------------------          *
*************************************************/
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
void  Display_Header(int Page);
float Design_Alpha();
float Select_Filter(float *Alpha);

/*=======================================================
  Select_Filter() - Select type Filter
  Prototype:  float Select_Filter(void);
  Return   :  float
=======================================================*/
float Select_Filter(float *Alpha)
{
    int n=0,number;
    char Number[5];
  do{
    n++;
    Display_Header(3);
    printf("\n");
    printf("\n");
    printf("       Your select number of Alpha by you must be choose
munber\n");
    printf("       between 1-15 !\n");
    printf("-------------------------------------------------------------\n");
    printf("          1) Alpha=infinity (Butterworth)\n");
    printf("          2) Alpha=10       (10-th M.A.L.)\n");
    printf("          3) Alpha=3        (3rd   M.A.L.)\n");
    printf("          4) Alpha=2        (2nd   M.A.L.)\n");
    printf("          5) Alpha=1.5      (1.5   Ultraspherical)\n");
    printf("          6) Alpha=1        (1st   M.A.L.)\n");
    printf("          7) Alpha=0.5      (2nd   Chebyshev)\n");
    printf("          8) Alpha=0        (Legendre)\n");
    printf("          9) Alpha=-0.25    (-0.25 Ultraspherical)\n");
    printf("         10) Alpha=-0.5     (1st Chevbyshev(Equal-Riple))\n");
    printf("         11) Alpha=-0.75    (-0.75 Ultraspherical)\n");
    printf("         12) Alpha=-0.9     (-0.9  Ultraspherical)\n");
    printf("         13) Alpha=-0.95    (-0.95 Ultraspherical)\n");
    printf("         14) Alpha=-0.9999  (-0.9999 Ultraspherical)\n");
    printf("         15) Design Alpha by your!\n");
    printf("\n");
    printf("                Input number of Alpha..... : ");
    gets(Number);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        number=atoi(Number);
        if(number!=0)
        switch(number)
        {
         case 1 :*Alpha=100000000;break;
         case 2 :*Alpha=10;break;
         case 3 :*Alpha=3;break;
         case 4 :*Alpha=2;break;
         case 5 :*Alpha=1.5;break;
         case 6 :*Alpha=1;break;
         case 7 :*Alpha=0.5;break;
         case 8 :*Alpha=0;break;
         case 9 :*Alpha=-0.25;break;
         case 10:*Alpha=-0.5;break;
         case 11:*Alpha=-0.75;break;
         case 12:*Alpha=-0.9;break;
         case 13:*Alpha=-0.95;break;
         case 14:*Alpha=-0.9999;break;
         case 15:*Alpha=Design_Alpha();
        }
        if((number<1)||(number>15))printf("\a");
      }while((number<1)||(number>15));
    return number;
   }

   /*=======================================================
    Design_Alpha(void) - Select Alpha by yourself
    Prototype:  float Desing_Alpha(void);
    Return   :  float
   =======================================================*/
   float Design_Alpha()
   {
    float Al_ph;
    printf("       Please specify Alpha..... : ");
    scanf("%f",&Al_ph);
    return Al_ph;
   }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โมดูล F_SUM.C

```
/*************************************************
*          ------------------------------        *
*  ------- Copyright (c) 1997 by Som&Oat&Sing------- *
*          ------------------------------        *
*************************************************/

#include <stdio.h>
#include <conio.h>

void Display_Header(int Page);
void Summary_Specify(int number_F,float Mag_Pass,float Mag_Stop,
                     float Fp,float Fs,float Alpha,int Order);

/*=======================================================
 Show_Type_Filter() - displays of type of Filter you choose
 Prototype:  void Show_Type_Filter();
 Return   :  void
=======================================================*/
void Show_Type_Filter(int number_F,float Mag_Pass,float Mag_Stop,float
Fp,float Fs,float Alpha)
{
 char ans;

 Display_Header(4);
 printf("\n\n");
 printf("                    SUMARY SPECIFY RESPONSE\n");
 printf("\n\n");
 switch(number_F)
{
 case 1:
 {
 printf("        Approximation n-th order Low-Pass..Butterworth filter\n");
 break;
 }
 case 2:
 {
 printf("        Approximation n-th order Low-Pass..10th M.A.L. filter\n");
 break;
 }
 case 3:
 {
 printf("        Approximation n-th order Low-Pass..3rd M.A.L. filter\n");
 break;
 }
 case 4:
 {
 printf("        Approximation n-th order Low-Pass..2nd M.A.L. filter\n");
 break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
    case 5:
    {
    printf("          Approximation n th order Low-Pass..1.5 Ultraspherical
   filter\n");
    break;
    }
    case 6:
    {
    printf("          Approximation n-th order Low-Pass..1st M.A.L. filter\n");
    break;
    }
    case 7:
    {
    printf("          Approximation n th order Low-Pass..2nd Shevbyshev
   filter\n");
    break;
    }
    case 8:
    {
    printf("          Approximation n th order Low-Pass..Legendre filter\n");
    break;
    }
    case 9:
    {
    printf("          Approximation n th order Low-Pass..-0.25 Ultraspherical
   filter\n");
    break;
    }
    case 10:
    {
    printf("       Approximation n th order Low-Pass..1st Chevbyshev(Equal-
   Ripple) filter\n");
    break;
    }
    case 11:
    {
    printf("          Approximation n th order Low-Pass..-0.75 Ultraspherical
   filter\n");
    break;
    }
    case 12:
    {
    printf("          Approximation n th order Low-Pass..-0.9 Ultraspherical
   filter\n");
    break;
    }
    case 13:
    {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    printf("          Approximation n th order Low-Pass..-0.95 Ultraspherical
   filter\n");
    break;
    }
    case 14:
    {
    printf("          Approximation n th order Low-Pass..-0.9999 Ultraspherical
   filter\n");
    break;
    }
    case 15:
    {
    printf("          Approximation n th order Low-Pass..????? Ultraspherical
   filter\n");
    break;
    }
   }
   if(number_F==1)
    {
    printf("\n");
    printf("                    Filter impelmentation is :   Analog\n");
    printf("                    Filter Alpha is          :   Infinity\n");
    printf("                    Passband gain        (dB):   %0.2f\n",Mag_Pass);
    printf("                    Stopband gain        (dB):   %0.2f\n",Mag_Stop);
    printf("                    Passband frequency   (Hz):   %0.2f\n",Fp);
    printf("                    Stopband frequency   (Hz):   %0.2f\n",Fs);
    }
   else
    {
    printf("\n");
    printf("                    Filter impelmentation is :   Analog\n");
    printf("                    Filter Alpha is          :   %0.4f\n",Alpha);
    printf("                    Passband gain        (dB):   %0.2f\n",Mag_Pass);
    printf("                    Stopband gain        (dB):   %0.2f\n",Mag_Stop);
    printf("                    Passband frequency   (Hz):   %0.2f\n",Fp);
    printf("                    Stopband frequency   (Hz):   %0.2f\n",Fs);
    }
   getch();
}
/*=====================================================
  Show_Type_Filter() - displays of type of Filter you choose
  Prototype:  void Show_Type_Filter();
  Return   : int number_F,float Mag_Pass,float Mag_Stop,float Fp,float
Fs,float Alpha
=====================================================*/
void Summary_Specify(int number_F,float Mag_Pass,float Mag_Stop,
                     float Fp,float Fs,float Alpha,int Order)

{
 char ans;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 Display_Header(4);
 printf("\n\n");
 printf("                    SUMARY SPECIFY RESPONSE\n");
 printf("\n\n");
 switch(number_F)
{
 case 1:
 {
 printf("        Approximation n-th order Low-Pass..Butterworth filter\n");
 break;
 }
 case 2:
 {
 printf("        Approximation n-th order Low-Pass..10th M.A.L. filter\n");
 break;
 }
 case 3:
 {
 printf("        Approximation n-th order Low-Pass..3rd M.A.L. filter\n");
 break;
 }
 case 4:
 {
 printf("        Approximation n-th order Low-Pass..2nd M.A.L. filter\n");
 break;
 }
 case 5:
 {
 printf("        Approximation n th order Low-Pass..1.5 Ultraspherical
filter\n");
 break;
 }
 case 6:
 {
 printf("        Approximation n-th order Low-Pass..1st M.A.L. filter\n");
 break;
 }
 case 7:
 {
 printf("        Approximation n th order Low-Pass..2nd Shevbyshev
filter\n");
 break;
 }
 case 8:
 {
 printf("        Approximation n th order Low-Pass..Legendre filter\n");
 break;
 }
 case 9:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 {
 printf("          Approximation n th order Low-Pass..-0.25 Ultraspherical
filter\n");
 break;
 }
 case 10:
 {
 printf("       Approximation n th order Low-Pass..1st Chevbyshev(Equal-
Ripple) filter\n");
 break;
 }
 case 11:
 {
 printf("          Approximation n th order Low-Pass..-0.75 Ultraspherical
filter\n");
 break;
 }
 case 12:
 {
 printf("          Approximation n th order Low-Pass..-0.9 Ultraspherical
filter\n");
 break;
 }
 case 13:
 {
 printf("          Approximation n th order Low-Pass..-0.95 Ultraspherical
filter\n");
 break;
 }
 case 14:
 {
 printf("          Approximation n th order Low-Pass..-0.9999 Ultraspherical
filter\n");
 break;
 }
 case 15:
 {
 printf("          Approximation n th order Low-Pass..????? Ultraspherical
filter\n");
 break;
 }
}
if(number_F==1)
 {
 printf("\n");
 printf("                Filter impelmentation is :   Analog\n");
 printf("                Filter Alpha is          :   Infinity\n");
 printf("                Passband gain       (dB):   %0.2f\n",Mag_Pass);
 printf("                Stopband gain       (dB):   %0.2f\n",Mag_Stop);
 printf("                Passband frequency  (Hz):   %0.2f\n",Fp);
```

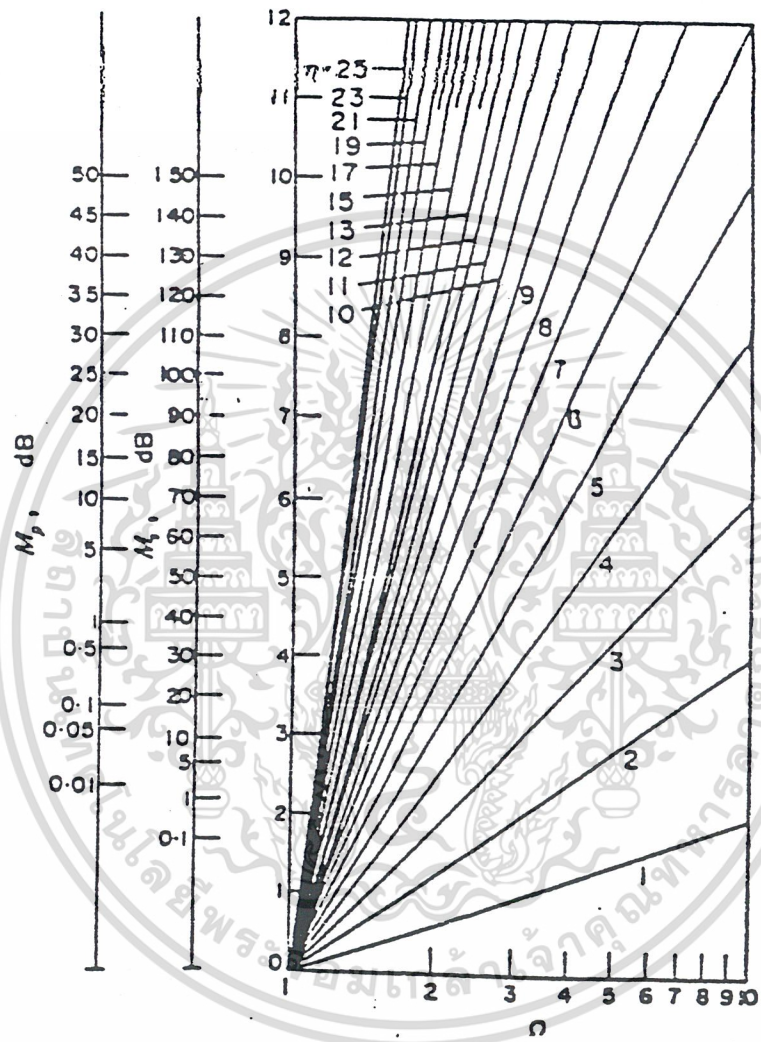
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 printf("                Stopband frequency   (Hz):    %0.2f\n",Fs);
 printf("                Order filter is          :    ");
 textcolor(BLINK|GREEN);
 cprintf("%d",Order);
 printf("\n");
 }
else
 {
 printf("\n");
 printf("                Filter impelmentation is  :    Analog\n");
 printf("                Filter Alpha is           :    %0.4f\n",Alpha);
 printf("                Passband gain        (dB):    %0.2f\n",Mag_Pass);
 printf("                Stopband gain        (dB):    %0.2f\n",Mag_Stop);
 printf("                Passband frequency   (Hz):    %0.2f\n",Fp);
 printf("                Stopband frequency   (Hz):    %0.2f\n",Fs);
 printf("                Order filter is          :    ");
 textcolor(BLINK|GREEN);
 cprintf("%d",Order);
 printf("\n");
 }
 getch();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

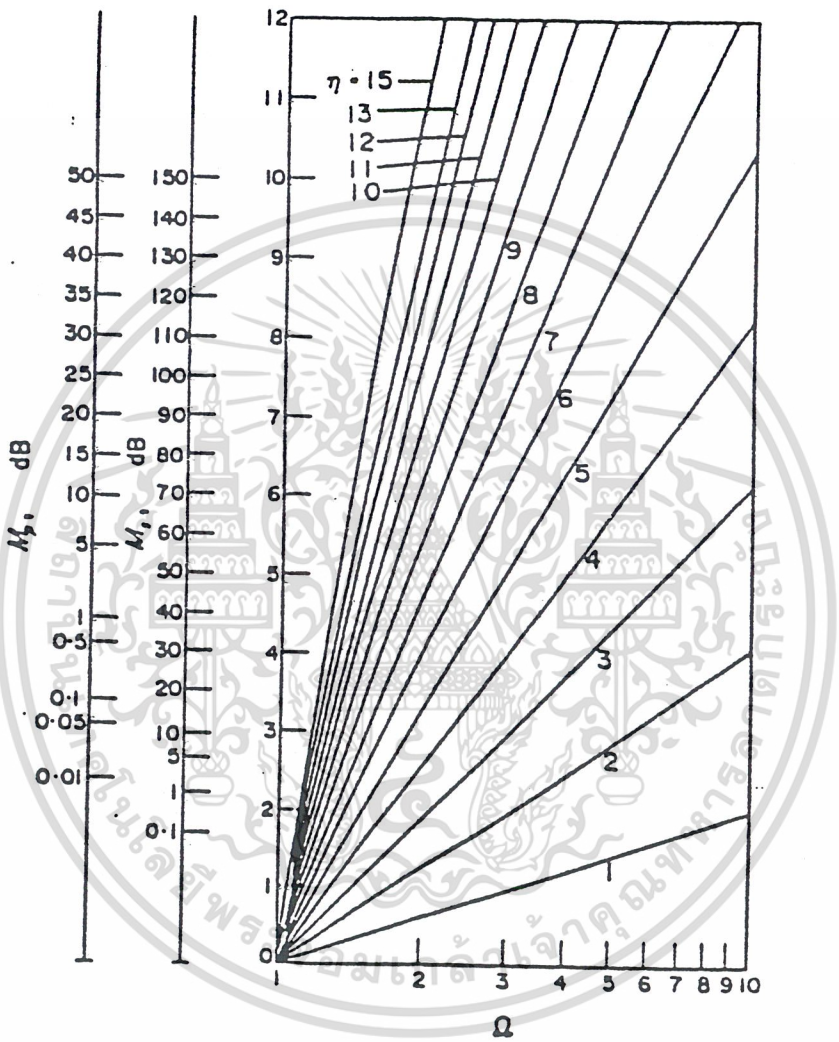
ภาคผนวก ค

NOMOGRAPH จริงที่ใช้ในการหา ORDER



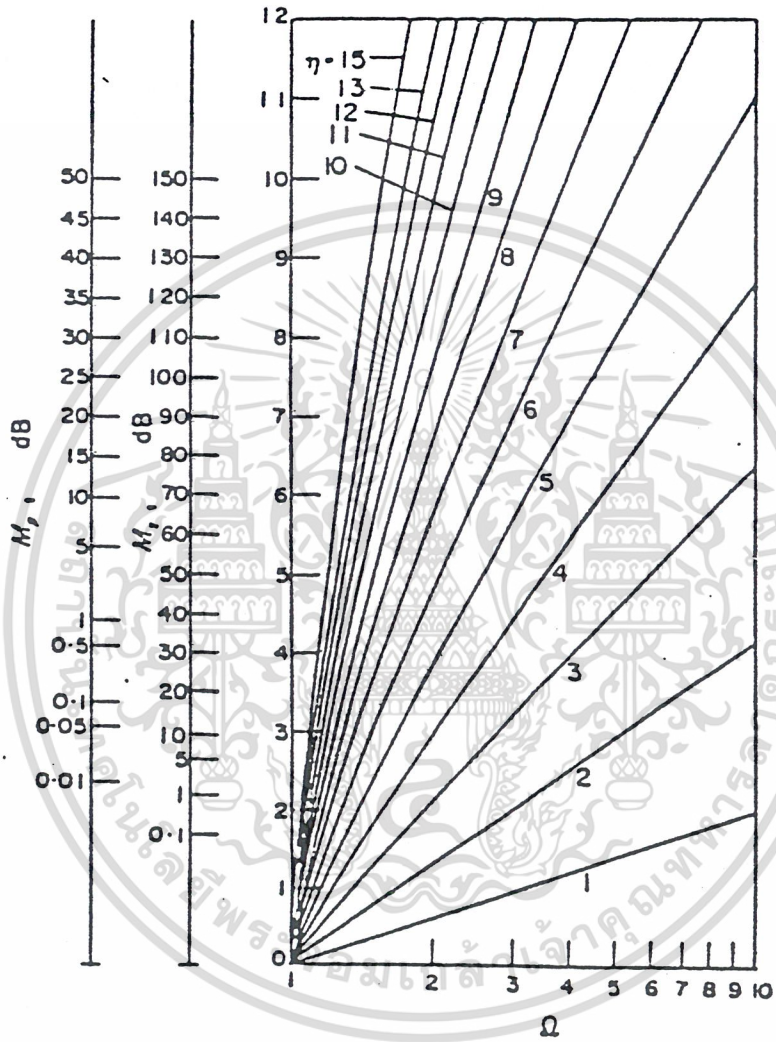
รูป A1. Nomograph ของ Butterworth (∞) filters.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



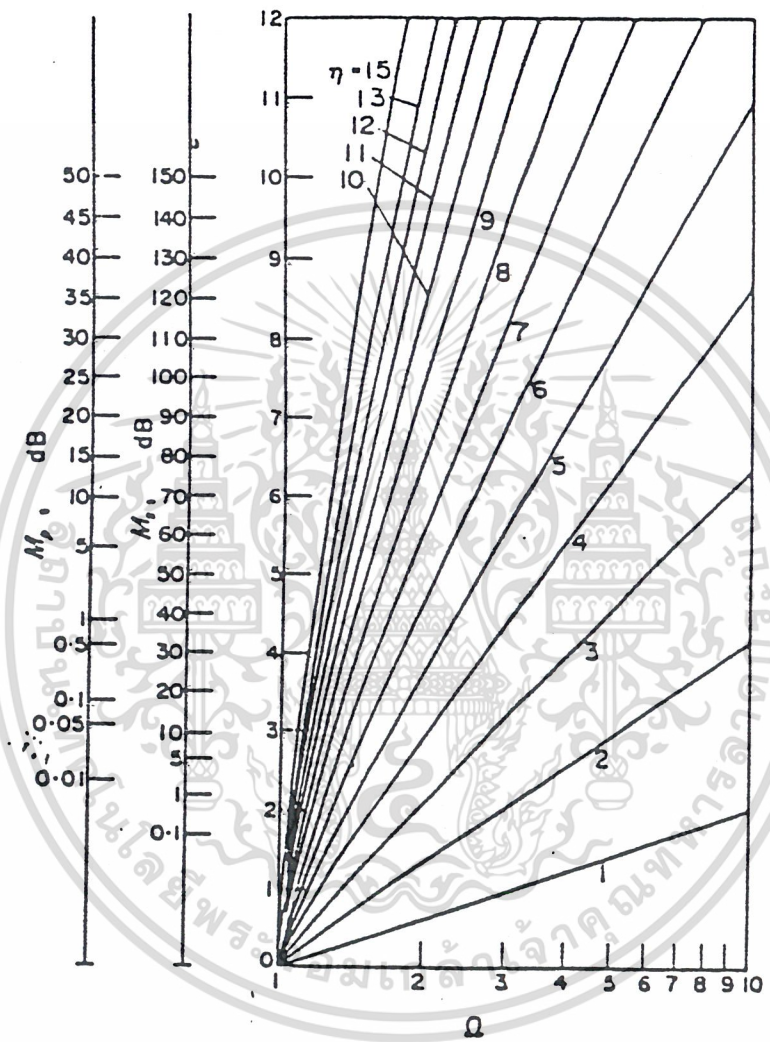
รูป A2. Nomograph ของ 10^{th} MAL (10) filters.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



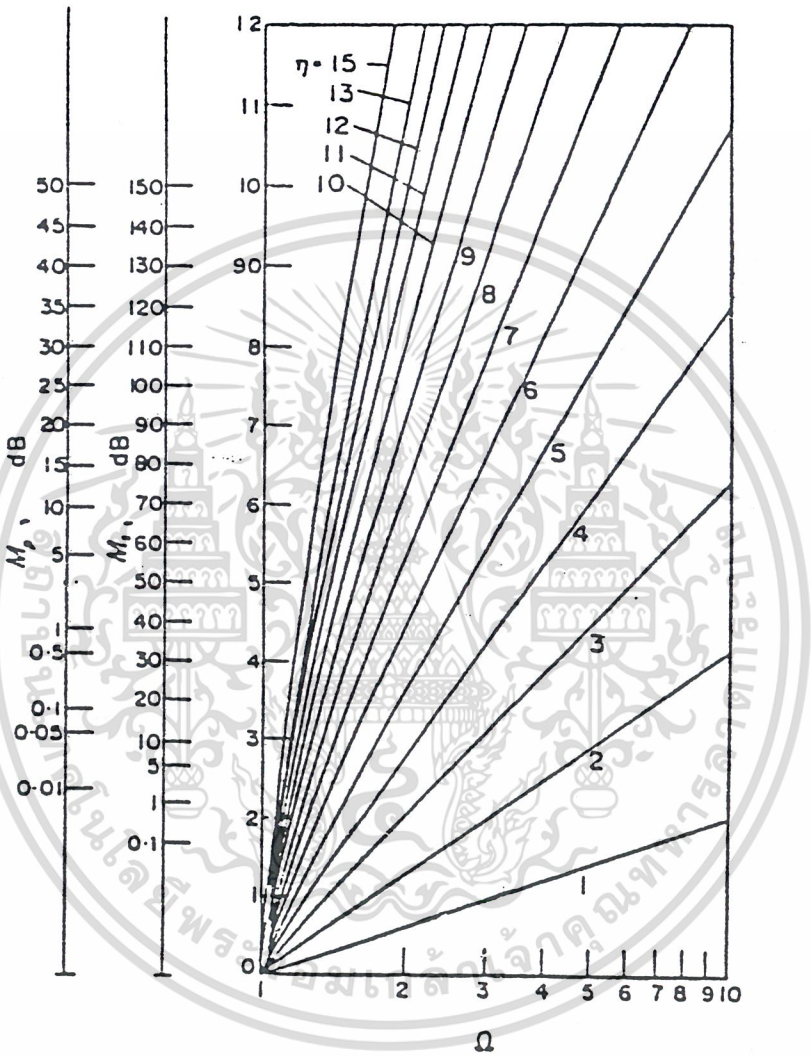
รูป A3. Nomograph ของ 3rd MAL (3) filters.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



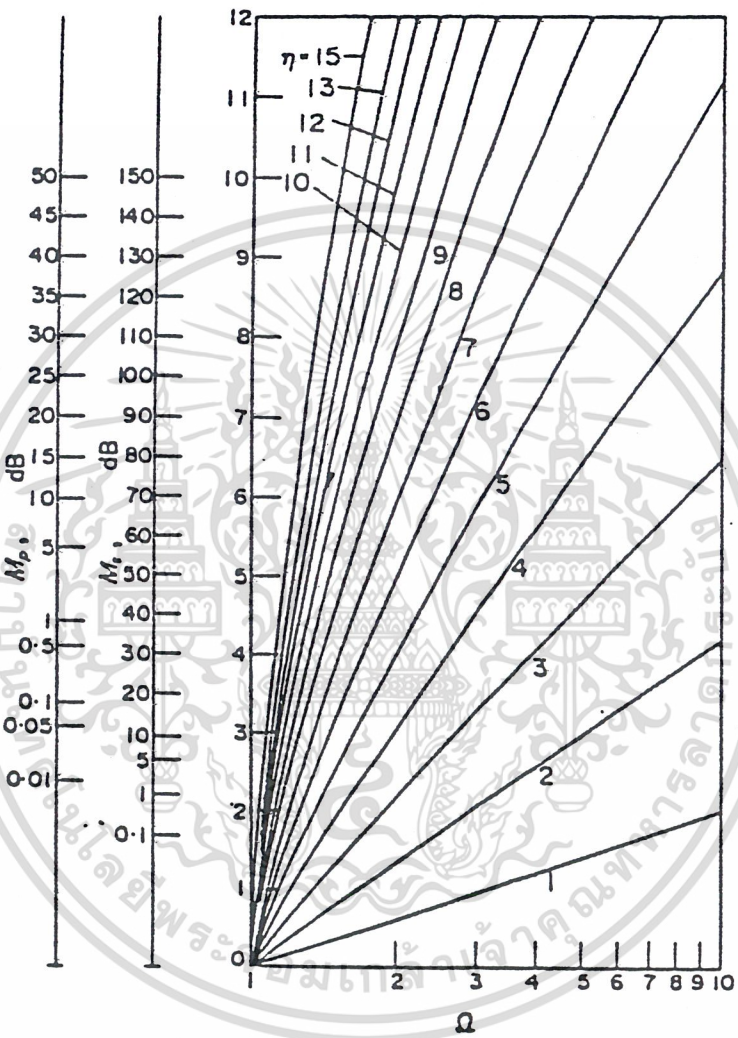
รูป A4. Nomograph ของ 2nd MAL (2) filters.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



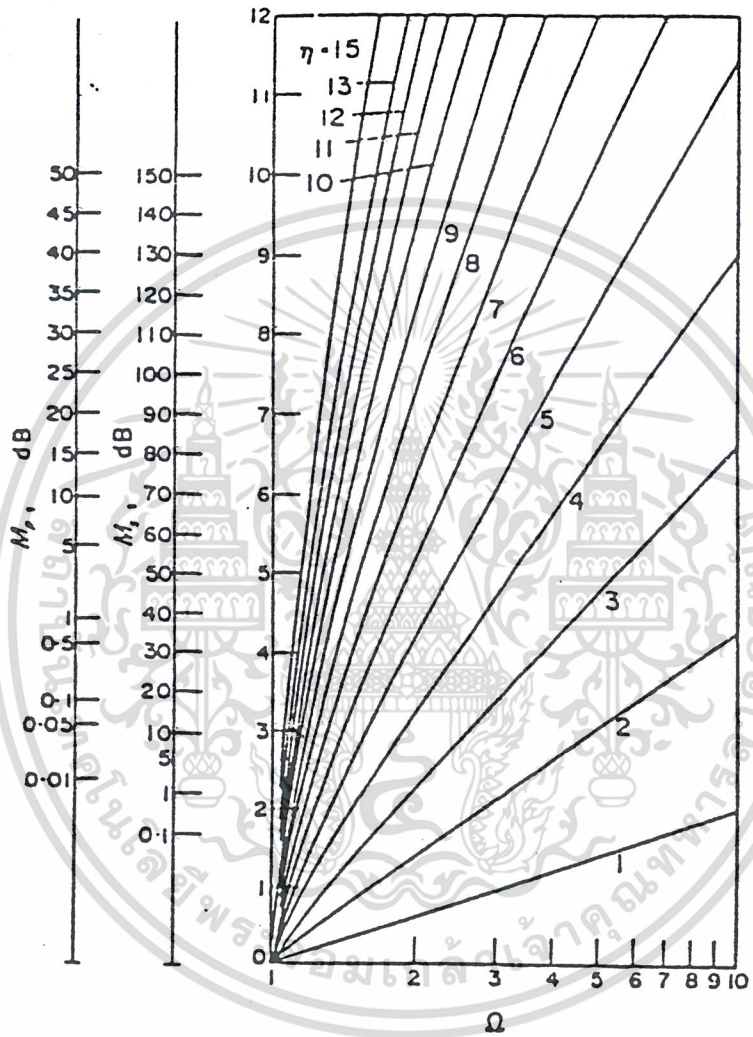
รูป A5. Nomograph ของ 1.5 ultraspherical (1.5) filters.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



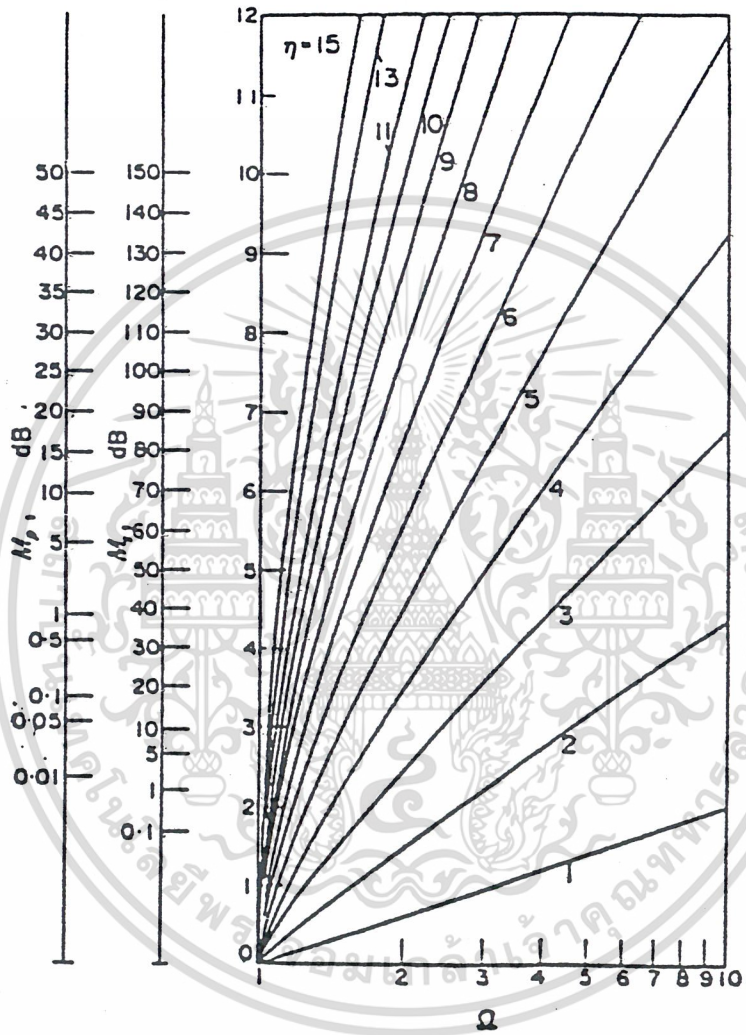
รูป A6. Nomograph ของ 1st MAL (1) filters.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



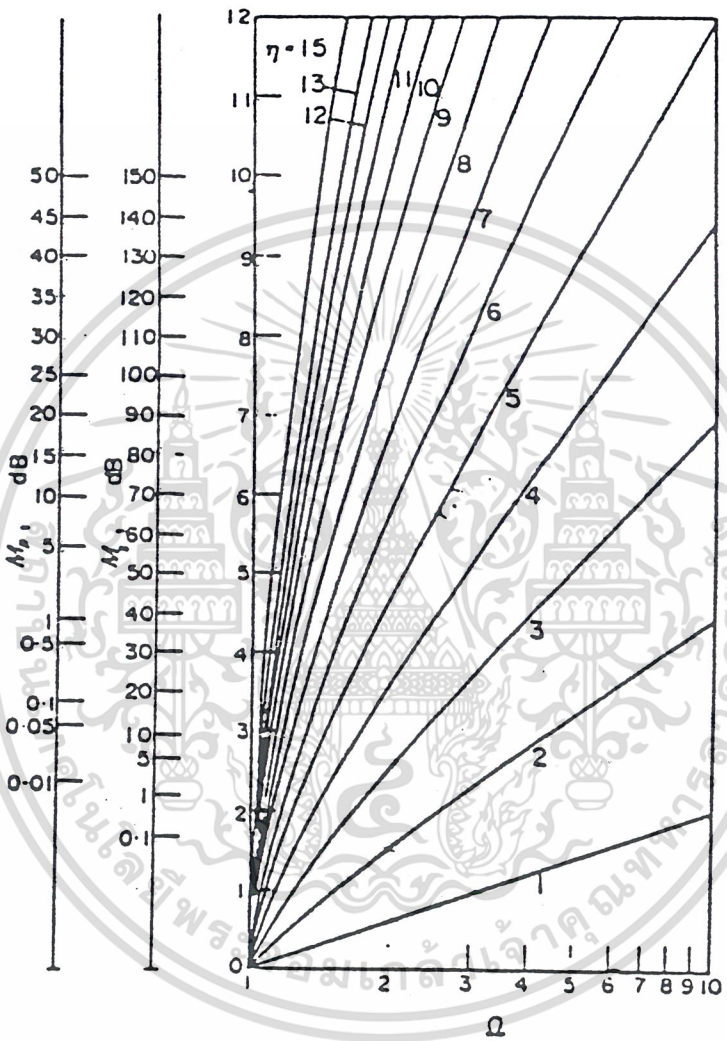
รูป A7. Nomograph ของ Chebyshev (0.5) filters ชนิดที่ 2.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



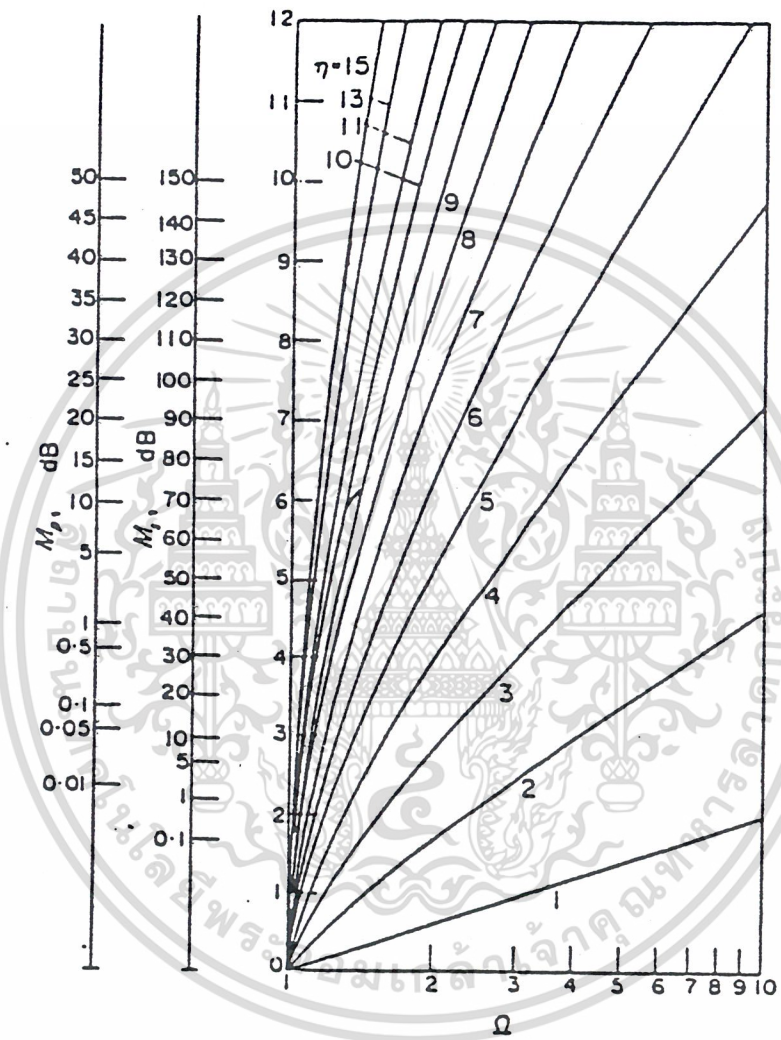
รูป A8. Nomograph ของ Legendre (0) filters.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



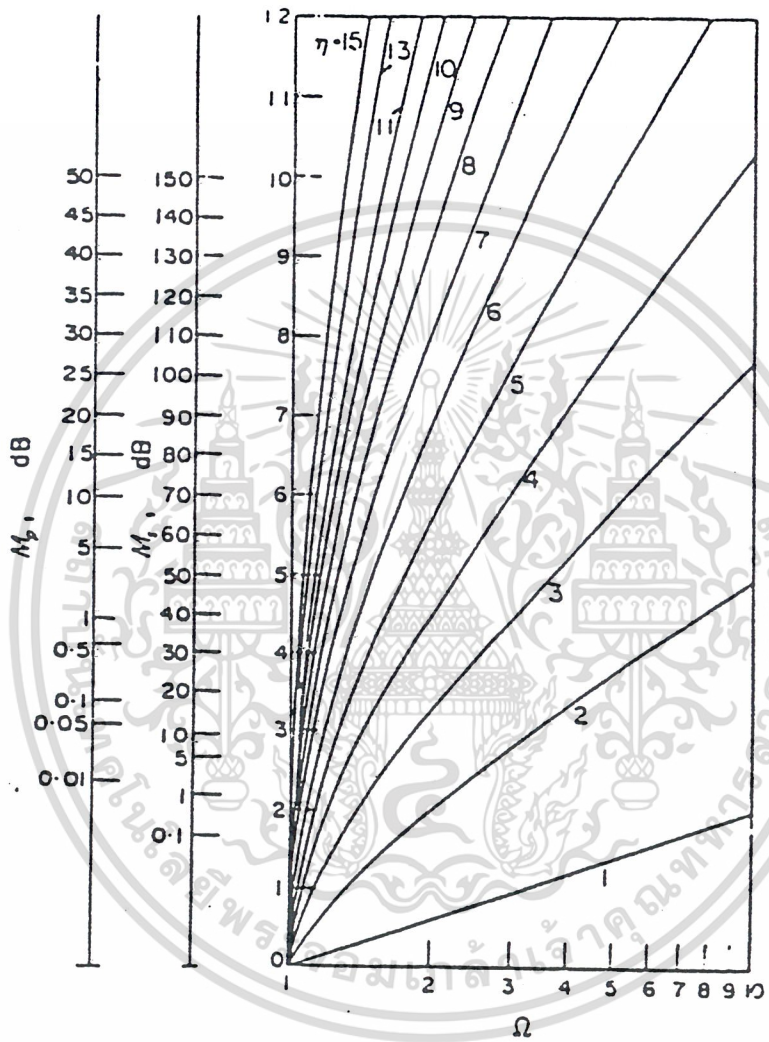
รูป A9. Nomograph ของ -0.25 ultraspherical (-0.25) filters.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



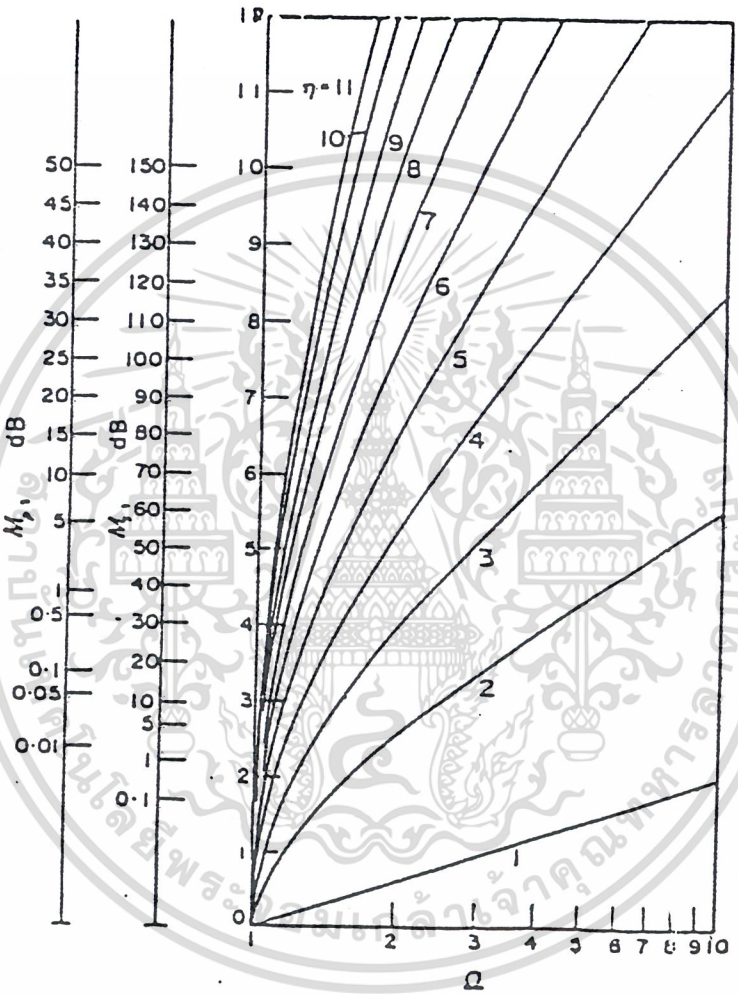
รูป A10. Nomograph ของ Chebyshev (-0.5) filters ชนิดที่ 1 (equal ripple).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



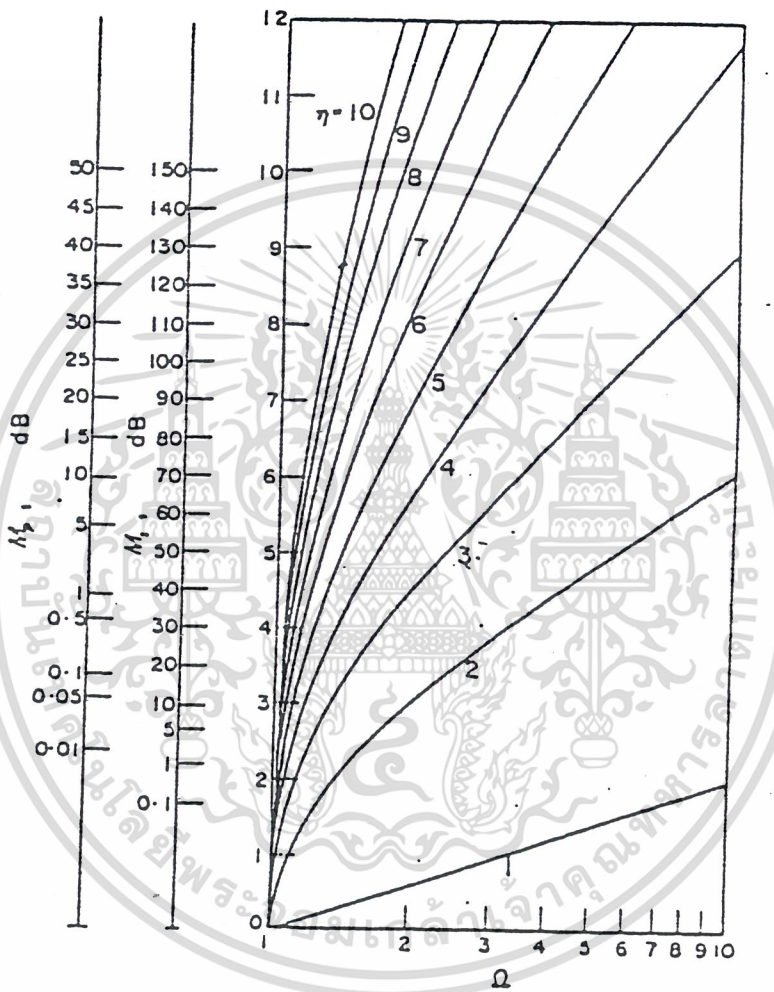
รูป A11. Nomograph ของ -0.75 ultraspherical (-0.75) filters.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



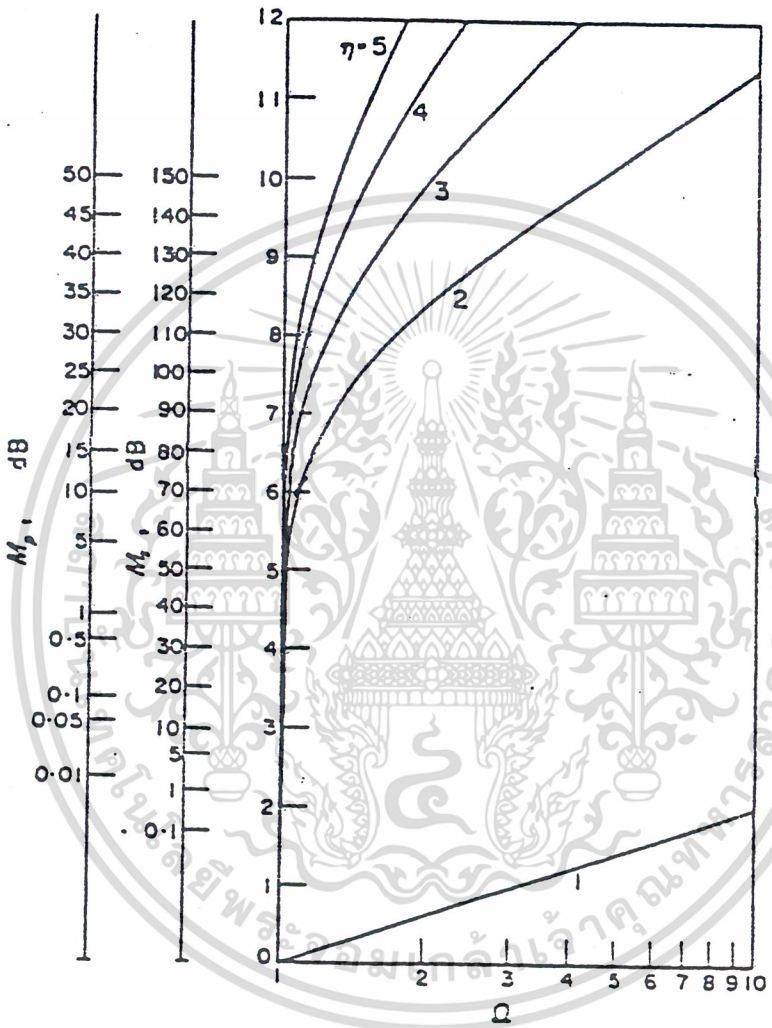
รูป A12. Nomograph ของ -0.90 ultraspherical (-0.90) filters.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป A13. Nomograph ของ -0.95 ultraspherical (-0.95) filters.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป A14. Nomograph ของ -0.9999 ultraspherical (-0.9999) filters.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. Joe N. Hallberg and Claude S. Lindquist, "Nomogramphs and Filters," J. Franklin Inst., Vol. 302, Number 2, pp 111-145, Aug., 1976.
2. Les Thede, "Analog and Digital Filter Design Using C," Prentice Hall New Jersey, 1983.
3. Wai - Kai Chen, "Passive and Active Filters Theory and Implementations," Wiley, New York, 1988.
4. Gobind Daryanani, "Principles of Active Network Synthesis and Desing," Wiley, New York, 1976.
5. Arthur B. Williams, "Electronic Filter Design Handbook," Mc Graw - Hill, New York, 1981.
6. C. Britton Rorabaugh, "Circuit Design and Analysis Featuring C Routines, Mc Graw - Hill, 1992.
7. ธันวา ศรีประมง, "การเขียนโปรแกรมภาษาซีสำหรับวิศวกรรม," พิมพ์ครั้งที่ 4 กรุงเทพมหานคร; มหาวิทยาลัยเทคโนโลยีมหานคร, 2538.
8. มัณฑนา ปราการสมุทร, "การเขียนชุดคำสั่งภาษาซี," พิมพ์ครั้งที่ 3, กรุงเทพมหานคร; ดวงกลมสมัย, 2534.
9. สมพัฒน์ รุ่งตะวันเรืองศรี, "การเรียนรู้คอมพิวเตอร์กราฟิกส์ 2 มิติด้วยภาษาซี," กรุงเทพมหานคร; ซีเอ็ดยูเคชั่น, 2521.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้