

การออกแบบวงจรกรองความถี่ดิจิทัลโดยใช้ไบเลียร์ทรานสฟอร์ม

Digital Filter Designed by Bilinear Transform



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาคามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมโทรคมนาคม

สถาบัน เทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2536

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Σ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(033189)

การออกแบบวงจรกรองความถี่ดิจิทัลโดยใช้ไบลิเนียร์ทรานสฟอร์ม

Digital Filter Designed by Bilinear Transform

โดย 1. นางสาวสุรีย์ จรัสเนตร 33100456
2. นางสาวฉัตร บุญรอด 33100513

อาจารย์ที่ปรึกษา

อาจารย์ปราโมทย์ วาดเขียน

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ ทำการศึกษาและทดลองการออกแบบวงจรกรองความถี่ดิจิทัล โดยอาศัยหลักการการแปลงทรานสเฟอร์ฟังก์ชันจากสมการ prototype ของวงจรกรองความถี่อนาล็อกไปเป็นทรานสเฟอร์ฟังก์ชันของวงจรกรองความถี่ดิจิทัล ในที่นี้เราเลือกใช้การแปลงแบบไบลิเนียร์ทรานสฟอร์ม โดยการทดลองแบ่งออกเป็น 2 ส่วน คือ ทางด้านซอฟต์แวร์จะเขียนโปรแกรมเพื่อทำหน้าที่ทำงานเป็นวงจรกรองความถี่ดิจิทัลโดยใช้ทรานสเฟอร์ฟังก์ชันของวงจรกรองความถี่ดิจิทัลเอาไว้ แล้วนำเอาสัญญาณดิจิทัลอินพุตป้อนเข้าไปผ่านการประมวลผลภายใน จะได้สัญญาณดิจิทัลเอาต์พุตที่ผ่านวงจรกรองความถี่ดิจิทัลออกมา ทางด้านฮาร์ดแวร์เราสร้างวงจรกรองความถี่ดิจิทัลโดยใช้วงจรสวิทช์คาปาซิเตอร์ คือ การแทนที่ความต้านทานในวงจร RC Active Filter ด้วยวงจรสวิทช์คาปาซิเตอร์ที่ต่อแบบไบลิเนียร์

Abstract

This thesis is learning and experiment about designed of digital filter, usage the transfer function prototype of analog filter transformation into digital filter transfer function by bilinear transform. This thesis is divided into two sections, each one is the software which have function as digital filter to process digital signal, another is the hardware which apply of bilinear transform use the switched capacitor circuit is fabricated resistance in RC Active Filter with switched capacitor using in the bilinear.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดเบี่ยงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการของดิจิทัลเดอเลอร์	3
บทที่ 3 สวิตช์คาปาซิเตอร์	17
3.1 สวิตช์คาปาซิเตอร์	17
3.2 การแทนวงจรสมมูลย์ความต้านทาน ด้วยสวิตช์คาปาซิเตอร์	19
3.3 การออกแบบวงจรสวิตช์คาปาซิเตอร์ ด้วยไบลิเนียร์แซดทรานสฟอร์ม	22
บทที่ 4 ทฤษฎีทั่วไปเกี่ยวกับฟิลเตอร์	25
4.1 การออกแบบวงจรกรองบีตเดอเลอร์เวอร์ธ โพลัสโพลโทด	25
4.2 การออกแบบวงจรกรองอีพีเซฟ โพลัสโพลโทด	30
4.3 การเปลี่ยนแปลงความถี่	34
บทที่ 5 ดิสครีทฟูริเยอร์ทรานสฟอร์ม (Discrete Fourier Transform DFT)	36
5.1 เอ็กโปเนนเชียลเชิงซ้อน	36
5.2 นิยามของดิสครีทฟูริเยอร์ทรานสฟอร์ม	37
5.3 คุณสมบัติบางประการของ DFT	44
บทที่ 6 โปรแกรมการออกแบบวงจรกรองความถี่ดิจิทัล	47
บทที่ 7 การทดลองวงจรกรองความถี่ดิจิทัล	87
7.1 วงจร Infinite Gain Multiple Feedback Low Pass Filter อันดับ 2	88
7.2 วงจร Sallen and Key (finite gain) High Pass Filter อันดับ 2	91

เอกสารนี้เป็นเอกสารที่เอกสารอ้างอิง การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ทฤษฎีเบื้องต้นเกี่ยวกับ Digital Filter

การกรองความถี่ (Filtering) เป็นแขนงหนึ่งของการประมวลผลของสัญญาณ (Signal processing) ซึ่งเราจะต้องทำความเข้าใจความหมายของคำที่เกี่ยวข้องต่อไปนี้คือ

คำว่าสัญญาณ (Signal) คือ ฟังก์ชันที่แทนความหมายของข้อมูล (Data) หรือข่าวสาร (Information) ซึ่งเป็นขนาดทางกายภาพ (Physical quantity) ที่สามารถวัดได้ (Measurable) สัญญาณอาจจะเขียนให้อยู่ในรูปฟังก์ชันทางคณิตศาสตร์ได้หลายตัวแปร ซึ่งขนาดของสัญญาณอาจจะเปลี่ยนไปตามเวลา (Time) ความถี่ (Frequency) และระยะทาง (Distance) เป็นต้น

ตัวแปรอิสระที่ใช้เขียนเป็นคณิตศาสตร์แทนความหมายของสัญญาณ อาจจะเป็นตัวแปรที่ต่อเนื่อง (Continuous) หรือไม่ต่อเนื่อง (Discrete) ก็ได้ ดังนั้นการประมวลผลสัญญาณจึงแบ่งชนิดของสัญญาณออกได้เป็น 2 ชนิดคือ

1. สัญญาณอนาลอกหรือสัญญาณที่มีความต่อเนื่องทางแกนเวลา (Analog or Continuous Signal)

คือสัญญาณที่มีความต่อเนื่องทั้งในแกนเวลาและขนาดของสัญญาณ คณิตศาสตร์พื้นฐานที่ใช้วิเคราะห์คือ สมการอนุพันธ์เชิงเส้น โดยใช้ลาปลาซทรานส์ฟอร์มช่วยแก้ปัญหา

2. สัญญาณดิจิทัลหรือสัญญาณที่ไม่มีความต่อเนื่องทางแกนเวลา (Digital or Discrete Signal)

คือสัญญาณที่ไม่ต่อเนื่องทั้งในแกนเวลาและขนาดของสัญญาณ คณิตศาสตร์พื้นฐานที่ใช้วิเคราะห์คือ สมการแตกต่างเชิงเส้น โดยใช้ Z-transform ช่วยแก้ปัญหา

การประมวลผลสัญญาณดิจิทัล (Digital Signal Processing)

การประมวลผลสัญญาณดิจิทัลหมายถึง การแทนสัญญาณต่างๆด้วยลำดับของตัวเลขและสัญลักษณ์ และทำการประมวลผลลำดับเหล่านี้ออกมา วิธีการประมวลผลอาจจะมีจุดประสงค์แตกต่างกัน เช่น การประมวลคุณสมบัติของสัญญาณ หรือการแปลงสัญญาณหนึ่งให้ไปอยู่

ในอีกรูปแบบหนึ่ง ซึ่งเหมาะสมในการพิจารณาวิเคราะห์มากกว่าสัญญาณเดิม เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขอบข่ายหรือสาขาของการประมวลผลสัญญาณดิจิทัลได้พัฒนาขึ้นอย่างมากมายและได้ช่วยขยายพื้นฐานทางทฤษฎีในสาขาอื่นอีกเป็นจำนวนมาก ซึ่งการกรองสัญญาณดิจิทัล (Digital filter) ก็เป็นแขนงหนึ่งของการประมวลผลสัญญาณดิจิทัล

การกรองสัญญาณดิจิทัล (Digital Filter)

การกรองสัญญาณดิจิทัล คือ กระบวนการคำนวณ (Computation Process) หรืออัลกอริทึม (Algorithm) ซึ่งสัญญาณดิจิทัลหรือลำดับของสัญญาณอินพุต (Input) ถูกแปลงให้เป็นลำดับของตัวเลขใหม่ที่เรียกว่า สัญญาณเอาต์พุตดิจิทัล (Output Digital Signal) กระบวนการคำนวณอาจจะเป็นการกรองความถี่ต่ำ (Low Pass Filter) การกรองความถี่สูง (High Pass Filter) การหาค่าอนุพันธ์ (Derivative) เป็นต้น

Filter หมายถึง อุปกรณ์ที่สามารถเปลี่ยนแปลงสเปกตรัม ซึ่งเป็นองค์ประกอบทางความถี่ของสัญญาณหรือเรียกรวมกันว่า สเปกตรัมความถี่ ให้มีคุณสมบัติตามที่ต้องการได้ ดังนั้น Digital filter จะหมายถึง ตัวประมวลผลสัญญาณดิจิทัลที่ทำให้สเปกตรัมความถี่ของสัญญาณดิจิทัลเปลี่ยนแปลงไป

การกรองสัญญาณดิจิทัล เป็นการประมวลผลสัญญาณขั้นหนึ่งที่มีความสำคัญ โดยที่สัญญาณดิจิทัลมักจะ เกี่ยวเนื่องกับ การกำจัดสัญญาณรบกวน (Noise) และลดความเพี้ยนของสัญญาณ การทำในปัจจุบันระบบการกรองสัญญาณดิจิทัลมีบทบาทสำคัญมากขึ้นก็อาจเนื่องมาจาก

(1) การที่มีวิธีการออกแบบตัวกรอง (Filter) ที่ง่ายแต่ใช้งานได้มีประสิทธิภาพ

(2) พัฒนาการของการออกแบบวงจรร VLSI ที่ทำให้การคูณตัวเลข การเก็บข้อมูลของสัญญาณดิจิทัลทำได้รวดเร็วขึ้นมาก

การกรองสัญญาณดิจิทัล เป็นกระบวนการที่ตัดแปลงสเปกตรัมของสัญญาณให้มีสเปกตรัมเป็นไปตามข้อกำหนดที่ต้องการหรือกำหนดให้ ซึ่งอาจเป็นการเพิ่มค่าหรือลดทอนค่าขนาดของสัญญาณในแถบความถี่ที่กำหนดให้ หรือทำการตัดสัญญาณในแถบความถี่ที่กำหนดให้ ออกไป หรือทำการแยกสัญญาณในแถบความถี่ที่ต้องการออกมาใช้งานก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการของดิจิตอลฟิลเตอร์

Digital Filter มีการนำไปประยุกต์ใช้งานกันอย่างกว้างขวาง อาจมาจากข้อได้เปรียบหลายประการดังนี้

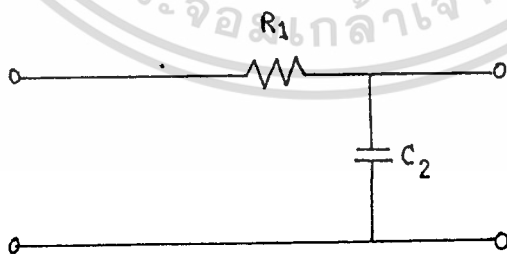
1) ผลตอบสนองความถี่ (Frequency Response) ของตัวกรอง สามารถออกแบบให้มีความถี่ใกล้เคียงกับผลตอบสนองความถี่ที่กำหนดค่าให้ หรือผลตอบสนองความถี่ที่ต้องการได้ นอกจากนี้การออกแบบตัวกรองให้มีผลตอบสนองเฟสเชิงเส้น ก็ทำได้ง่าย

2) คุณสมบัติตัวกรองที่ออกแบบและสร้างแล้วจะไม่ขยับเขยื้อนไปตามสภาวะแวดล้อม หรือตามอุณหภูมิ หรือระยะเวลาการใช้งาน นอกจากนี้ยังสามารถใช้งานในย่านความถี่ต่ำได้เป็นอย่างดี

3) ถ้าพิจารณาในแง่มุมมองของเสถียรภาพของตัวกรอง ความเชื่อถือได้ ราคาหรือขนาดของตัวกรองสัญญาณดิจิตอล ให้ผลลัพธ์ดีกว่าของตัวกรองสัญญาณอนาล็อก

4) มีโปรแกรมซอฟต์แวร์ที่ใช้สร้างและทดสอบการทำงานได้บนเครื่องคอมพิวเตอร์ทั่วไป เข้าใจได้ง่าย Digital Filter อยู่บนพื้นฐานของการ Multiplication, การ Addition, การ Delay ความสัมพันธ์ระหว่าง Analog Signal Processing กับ Digital Signal Processing

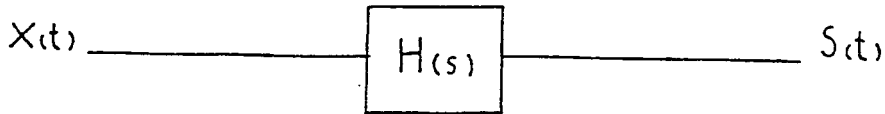
สมมติว่าเรามีวงจร Analog Filter ดังแสดงในรูปที่ 2.1



รูปที่ 2.1 แสดงวงจร Analog Filter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

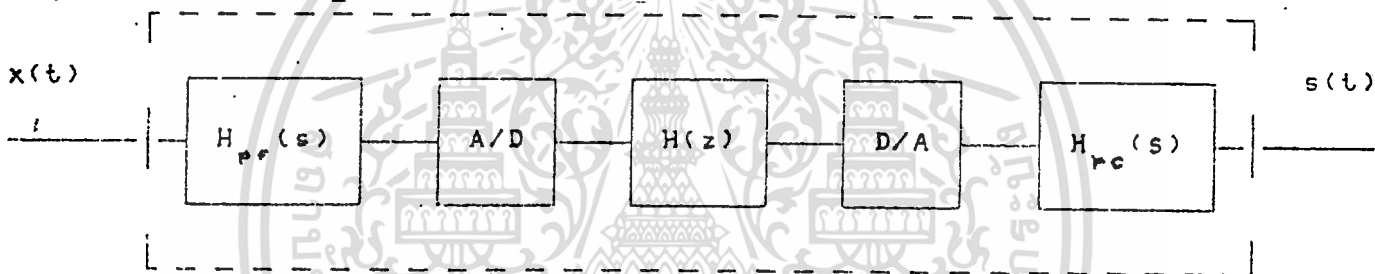
ซึ่งเราสามารถเขียนแทนด้วย Block diagram ได้ดังรูปที่ 2.2



รูปที่ 2.2 แสดง Analog Signal Processing of Analog Signals

เมื่อ $H(s)$ เป็น Transfer Function ของวงจร RC

เราสามารถที่จะแทนวงจรที่มีการประมวลผลสัญญาณแบบอนาล็อก โดยการใช้ประมวลผลสัญญาณแบบดิจิทัลไปแทน ดังรูปที่ 2.3



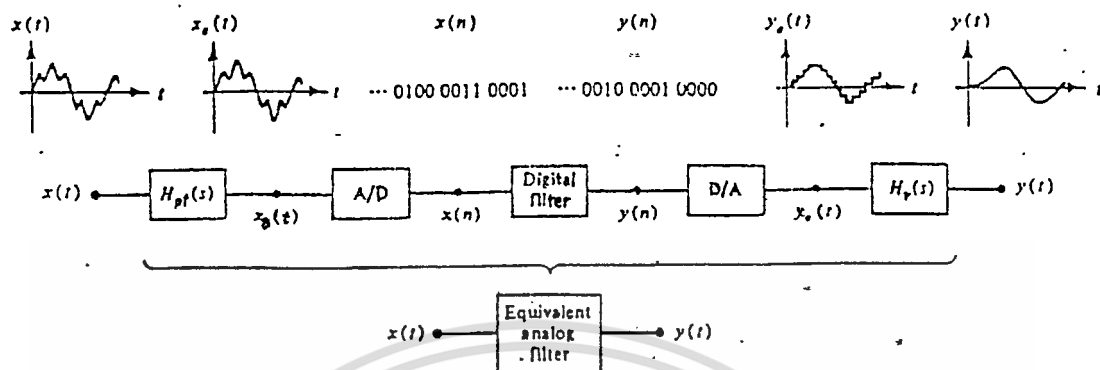
รูปที่ 2.3 แสดง Digital Signal Processing of Analog Signals

Transfer Function $H(s)$ เราสามารถนำไปประมวลผลในรูปแบบ DSP (Digital Signal Processing) โดยใช้ไมโครโปรเซสเซอร์ที่มีความเร็วสูงและเป็นการออกแบบระบบด้วย Software ดังนั้น จึงประยุกต์ใช้งานง่าย ซึ่งการประมวลผลสัญญาณอนาล็อกในรูปแบบ DSP ประกอบด้วยขั้นตอนต่างๆ ดังนี้

1. สุ่มสัญญาณอนาล็อกที่เป็น Input โดยมีมองสัญญาณอนาล็อกเป็นห่างๆ (Discrete Time)
2. เปลี่ยนสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล (A/D)
3. ประมวลผลข้อมูลดิจิทัลนั้นตามอัลกอริทึมทางคณิตศาสตร์
4. แปลงผลลัพธ์จากสัญญาณดิจิทัลกลับเป็นอนาล็อกใหม่ (D/A)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการประมวลผลสัญญาณดิจิทัล



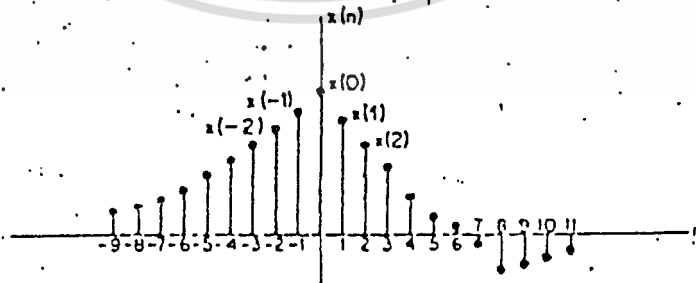
รูปที่ 2.4 Expanded Version of a Digital Signal Processign

ระบบและสัญญาณที่ไม่ต่อเนื่องทางแกนเวลา (Discrete-Time Signal and System)

1. สัญญาณที่ไม่ต่อเนื่องทางแกนเวลา (Discrete-Time Signal Sequences)

ในระบบสัญญาณที่ไม่ต่อเนื่องทางแกนเวลาจะกล่าวถึงการประมวลผลสัญญาณ ซึ่งแทนด้วยลำดับตัวเลข (Sequences) และลำดับตัวเลข X ตัวที่ n เขียนแทนด้วย $x(n)$ ซึ่งเขียนเป็นรูปทางคณิตศาสตร์ ดังนี้

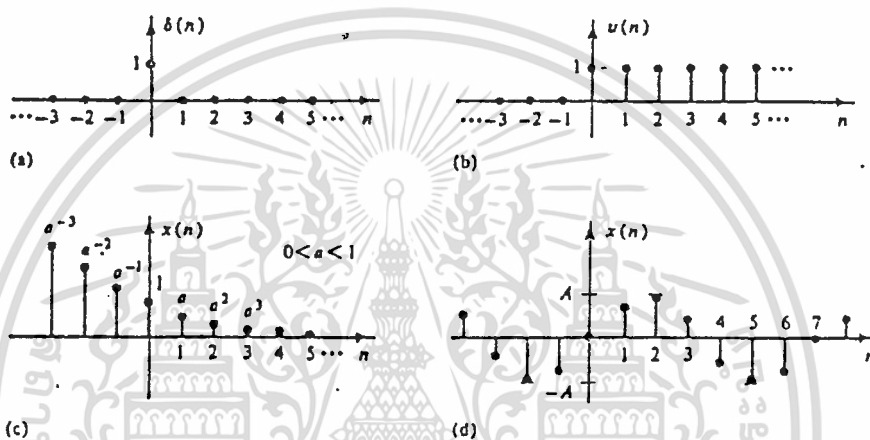
$$X = [x(n)] , \quad -\infty < n < \infty \quad (2.1)$$



รูปที่ 2.5 กราฟแสดงสัญญาณที่ไม่ต่อเนื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น เมื่อผู้ใดเห็นนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณหรือลำดับตัวเลขแทนด้วยกราฟได้ดังตัวอย่าง ในรูปที่ 2.5 แม้ว่าแกน Abscissa เป็นเส้นต่อเนื่อง แต่จริงๆ แล้ว $x(n)$ มีความหมายเฉพาะค่า Integer ของ n ส่วนค่า n ที่ไม่ใช่ตัวเลข Integer ไม่ได้หมายความว่าลำดับ $x(n)$ จะมีค่าเป็นศูนย์ เพียงแต่เราไม่ได้กำหนดค่า $x(n)$ เมื่อ n ไม่ใช่ตัวเลข Integer



รูปที่ 2.6 แสดงกราฟของสัญญาณ

(a) Unit Sample Sequence , (b) Unit Step Sequence

(c) Real Exponential Sequence , (d) sinusoidal Sequence

2. ระบบชีพท์-อินแวเรียนท์เชิงเส้น (Linear Shift-Invariant System)

เป็นระบบที่มีตัวแบบเชิงคณิตศาสตร์ (Mathematical Model) ไม่ซับซ้อนและการวิเคราะห์หาค่าคุณสมบัติของระบบทำได้ง่าย โดยปกติในการประมวลผลสัญญาณ คำว่า ระบบ หมายถึง ขั้นตอนวิธีหรือกระบวนการ ที่ใช้ในการแปลงคุณสมบัติของลำดับสัญญาณอันหนึ่ง

ไป เป็นลำดับสัญญาณอีกอันหนึ่ง เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้า $x(n)$ เป็นลำดับสัญญาณเข้า $y(n)$ เป็นลำดับสัญญาณออก

เขียนได้เป็น

$$y(n) = T [x(n)] \quad (2.2)$$

จากสมการ (2.2) เขียนเป็น Block Diagram ได้ดังรูปที่ 2.7



รูปที่ 2.7 แสดงตัวต้นแบบเชิงคณิตศาสตร์ของระบบ ซึ่งมีลำดับสัญญาณเข้า $x(n)$ ไปเป็นสัญญาณออก $y(n)$

ซึ่ง $T[.]$ เป็นตัวดำเนินการ (Operator) ที่ใช้แทนขั้นตอนวิธีแบบต่างๆ สำหรับระบบเชิงเส้น (Linear System) ถูกกำหนดด้วยรูปแบบทางคณิตศาสตร์ ดังนี้

$$\begin{aligned}
 T [ax_1(n) + bx_2(n)] &= aT [x_1(n)] + bT [x_2(n)] \\
 &= ay_1(n) + by_2(n)
 \end{aligned} \quad (2.3)$$

เมื่อ a และ b เป็นค่าคงที่ใดๆ

$y_1(n)$ และ $y_2(n)$ เป็นผลตอบสนองของสัญญาณเข้า $x_1(n)$ และ $x_2(n)$

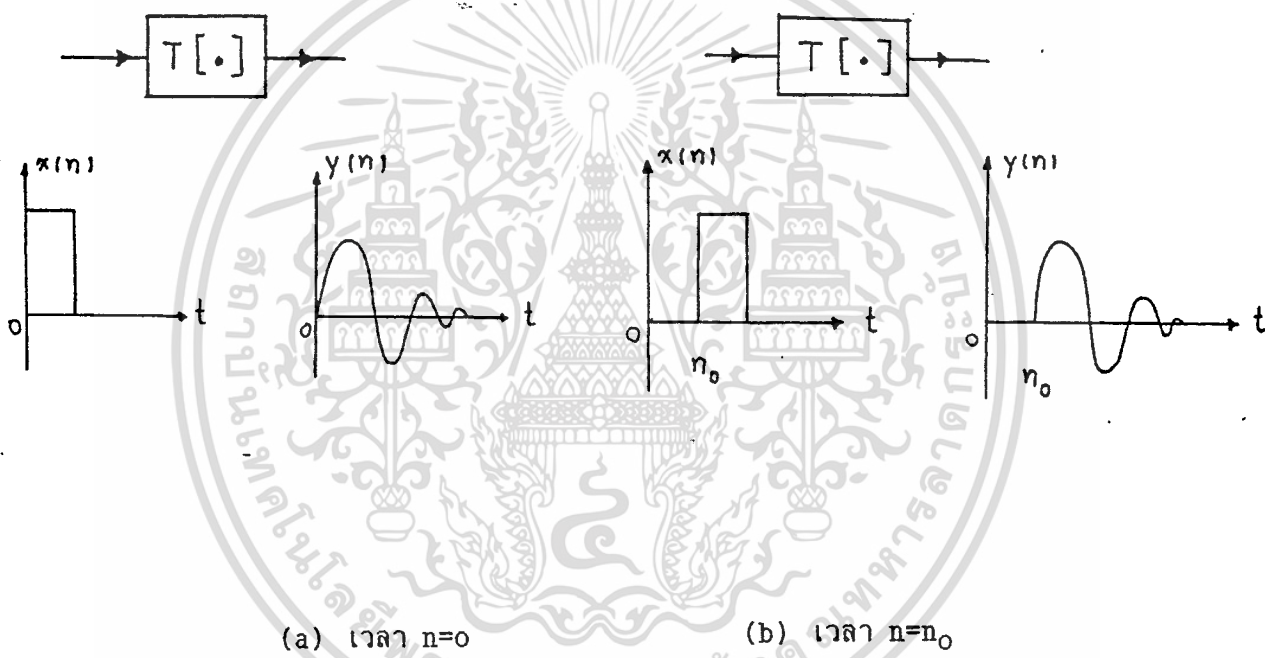
ในกรณี Discrete-Time System คือ $T[.]$ ถูก shift ด้วยเวลา n_0 ซึ่งเรียกว่า มีคุณสมบัติความเป็นเวลายินยง เราจะได้ว่า

$$x(n) \longrightarrow y(n) \text{ หรือ } T[x(n)] = y(n)$$

$$x(n-n_0) \longrightarrow y(n-n_0) \text{ หรือ } T[x(n-n_0)] = y(n-n_0) \quad (2.4)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งคุณสมบัตินี้หมายความว่า รูปคลื่นผลตอบสนองของระบบจะไม่แปรหรือ เปลี่ยน
 รูปร่างไปตาม เวลาที่ป้อนสัญญาณกระตุ้น แต่จะขึ้นกับลักษณะรูปคลื่นของสัญญาณกระตุ้น
 กล่าวคือ จากรูปที่ 2.8 ถ้าป้อนสัญญาณกระตุ้นที่หน่วงเวลาออกไป n_0 วินาที
 ผลตอบสนองของระบบยังคง เหมือนกับผลตอบสนองของระบบที่ป้อนเวลา $n=0$ เพียงแต่มี
 เวลาหน่วงออกไป n_0 วินาที เท่านั้น



(a) เวลา $n=0$ (b) เวลา $n=n_0$

รูปที่ 2.8 แสดงคุณสมบัติความเป็นเวลายินยงของระบบ

สมการแตกต่างเชิงเส้นสัมประสิทธิ์คงที่ (Linear Constant Coefficient Differential Equation)

ระบบ Shift-Invariant สามารถเขียนแทนด้วยสมการแตกต่างเชิงเส้นสัมประสิทธิ์คงที่ N ได้ดังนี้

$$\sum_{k=0}^N a_k y(n-k) = \sum_{r=0}^M b_r x(n-r) ; a_0 \neq 0 \quad (2.5)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$a_0 y(n) + \sum_{k=1}^N a_k y(n-k) = \sum_{r=0}^M b_r x(n-r)$$



เมื่อ a_0, a_1, \dots, a_N และ b_0, b_1, \dots, b_M เป็นค่าคงที่ ที่
ใช้ในการกำหนดคุณสมบัติ (Characteristic) ของระบบ
ถ้าเรากำหนดค่าให้ $a_0 = 1$ เราจะได้

$$y(n) = \sum_{r=0}^M b_r x(n-r) - \sum_{k=1}^N a_k y(n-k) \quad (2.6)$$

เราเรียกสมการ (2.6) ว่าสมการ Digital Filter

โดยที่ a_k และ b_r เป็นค่าสัมประสิทธิ์ จะมีค่าเท่ากับชนิดของ
ฟิลเตอร์ ซึ่งเรียกฟิลเตอร์แบบนี้ว่า รีเคอร์ซีฟฟิลเตอร์ (recursive filter)

ถ้าสัมประสิทธิ์ a_k ของ $y(n-k)$ มีค่าเป็นศูนย์ เราจะเรียกฟิลเตอร์
แบบนี้ว่า นอนรีเคอร์ซีฟฟิลเตอร์ (non-recursive filter) ซึ่งเขียนตามความ
สัมพันธ์ได้ดังสมการ (2.7)

$$y(n) = \sum_{r=0}^M b_r x(n-r) \quad (2.7)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

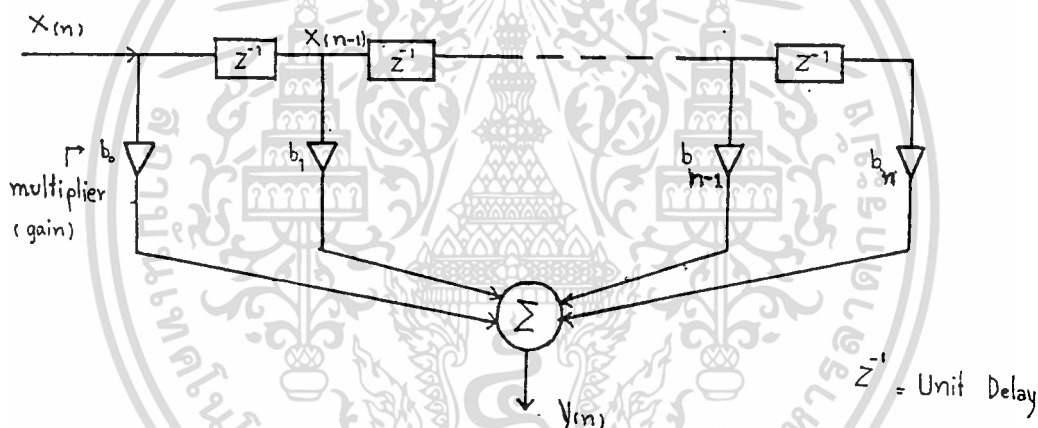
ชนิดของดิจิทัลฟิลเตอร์

แบ่งชนิดของดิจิทัลฟิลเตอร์ตามการประมวลผลของสัญญาณดิจิทัลว่ามีการป้อนกลับ (feedback) ของสัญญาณขาออกหรือไม่ ซึ่งแบ่งได้เป็น 2 ชนิดคือ

1. Nonrecursive Filter หรือเรียกว่า FIR filter (finite impulse response)
2. Recursive Filter หรือเรียกว่า IIR filter (Infinite impulse response)

Nonrecursive filter (FIR)

ไม่มีการป้อนกลับของสัญญาณ นั่นคือ เป็นการประมวลผลที่เกิดจากผลรวมของสัญญาณ input ก่อนหน้าคูณกับค่าสัมประสิทธิ์ (coefficient)



รูปที่ 2.9 แสดง Nonrecursive filter

ซึ่งจากสมการ Digital filter (2.6) เมื่อ a_k และ $y(n-k)$ มีค่าเป็นศูนย์ นั่นคือไม่มีการป้อนกลับของสัญญาณ จะได้ว่าสมการที่ได้จะเป็นสมการของ Nonrecursive filter

$$y(n) = \sum_{k=0}^N b_k x(n-k) \quad (2.8)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Recursive filter (IIR)

มีการนำเอาสัญญาณย้อนกลับ (feedback) ซึ่งการประมวลผลสัญญาณที่ได้ก็คือผลรวมของสัญญาณ input ปัจจุบัน สัญญาณ input ก่อนหน้าและสัญญาณที่ย้อนกลับ ซึ่งสมการของ recursive filter ก็คือสมการ Digital filter ทั่ว ๆ ไปนั่นเอง

$$y(n) = \sum_{k=0}^M b_k x(n-k) - \sum_{k=1}^N a_k y(n-k)$$

การสร้างวงจรของความถี่ดิจิทัลแบบรีเคอร์ซีฟ (Recursive filter Synthesis)

การสร้างวงจรของความถี่ดิจิทัลจะต้องเขียนความสัมพันธ์ระหว่างสัญญาณ input และสัญญาณ output ให้อยู่ในรูปแบบหรือ algorithm ที่ใช้คำนวณก่อน algorithm นี้จะใช้ตัวบวก (adder), ตัวหน่วง (delay) และตัวคูณด้วยค่าคงที่ (constant multiplier) มาช่วยเขียนโครงสร้างของวงจร

จากสมการของ Digital filter สมการที่ (2.6) กรณี $r=k$ จะได้

$$y(n) = \sum_{k=0}^M b_k x(n-k) - \sum_{k=1}^N a_k y(n-k) \quad (2.9)$$

Z-transform

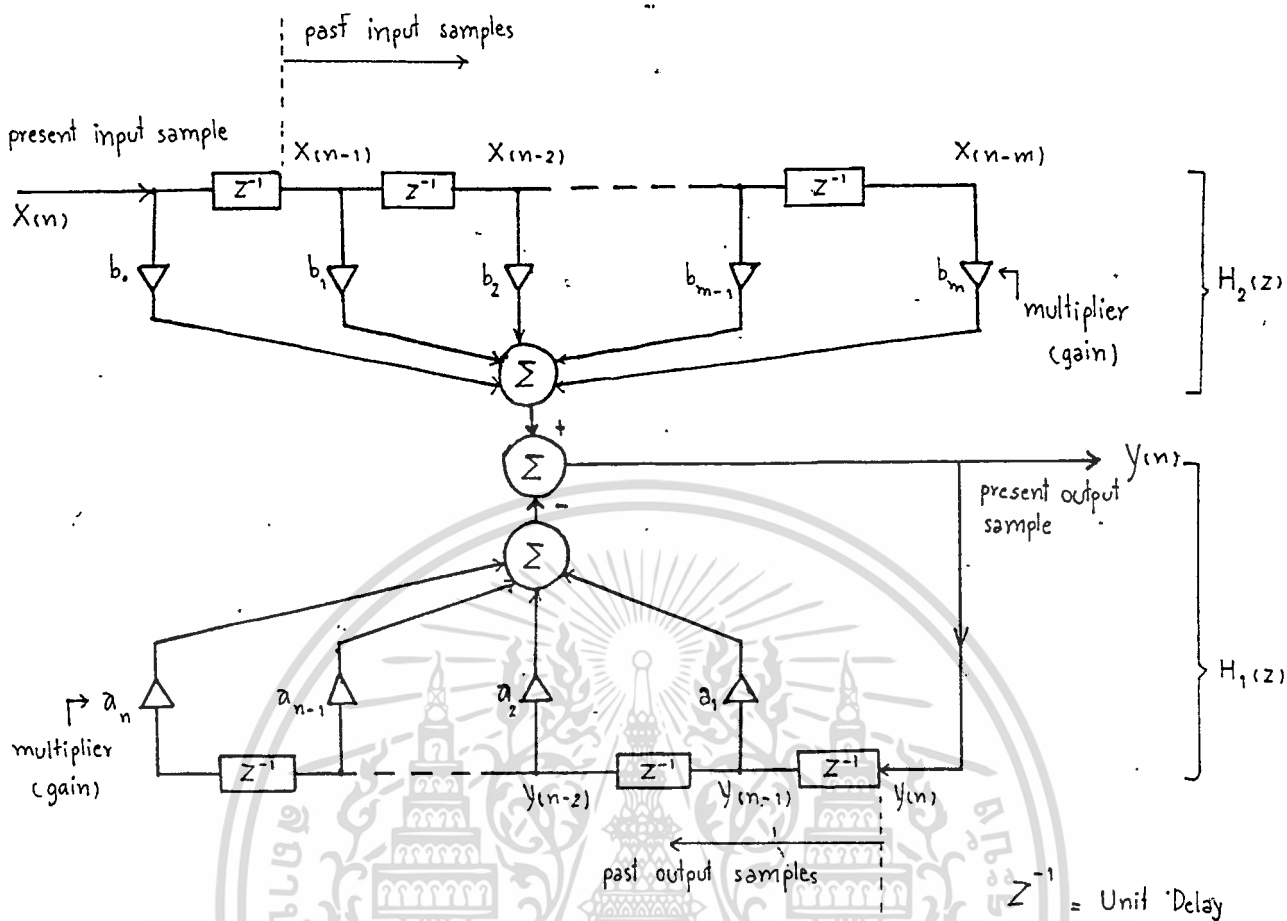
$$Y(z) = \sum_{k=0}^M b_k X(z) z^{-k} - \sum_{k=1}^N a_k Y(z) z^{-k}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$Y(z) \left[1 + \sum_{k=1}^N a_k z^{-k} \right] = X(z) \sum_{k=0}^M b_k z^{-k}$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^M b_k z^{-k}}{1 + \sum_{k=1}^N a_k z^{-k}} \quad (2.10)$$

เมื่อ $H(z)$ เป็น transform function ของสมการ digital filter เราสามารถแทนสมการ (2.9) ด้วย algorithm ในการคำนวณได้โดยตรง นั่นคือเราเอาค่า delay ของสัญญาณ input คูณด้วยสัมประสิทธิ์ b_k และค่า delay ของสัญญาณ output คูณด้วยสัมประสิทธิ์ a_k แล้วนำผลคูณทั้งหมดเอามาเข้าวงจร summing โดยสามารถเขียนเป็น block diagram ดังรูปที่ 2.10 ซึ่งเรียกว่า direct form I



รูปที่ 2.10 แสดง Recursive filter Direct Form I

จาก Direct form I เราจะแยกให้อยู่ในรูปของ Direct form II

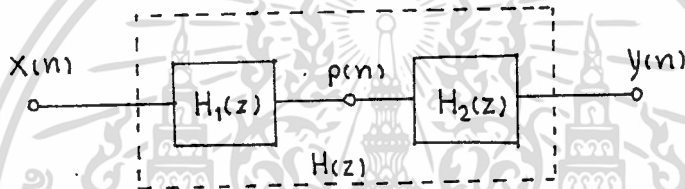
$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^M b_k z^{-k}}{1 + \sum_{k=1}^N a_k z^{-k}}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\text{ให้ } H_1(z) = \frac{1}{1 + \sum_{k=1}^N a_k z^{-k}} \quad (2.11)$$

$$H_2(z) = \sum_{k=0}^M a_k z^{-k} \quad (2.12)$$

$$\text{ดังนั้น } H(z) = H_1(z)H_2(z)$$



$$P(z) = H_1(z)X(z) \quad (2.13)$$

$$Y(z) = H_2(z)P(z) \quad (2.14)$$

แทน $H_1(z)$ และ $H_2(z)$ ในสมการ (2.13) และ (2.14) ได้

$$P(z) + \sum_{k=1}^N a_k P(z) z^{-k} = X(z)$$

$$Y(z) = \sum_{k=0}^M b_k P(z) z^{-k}$$

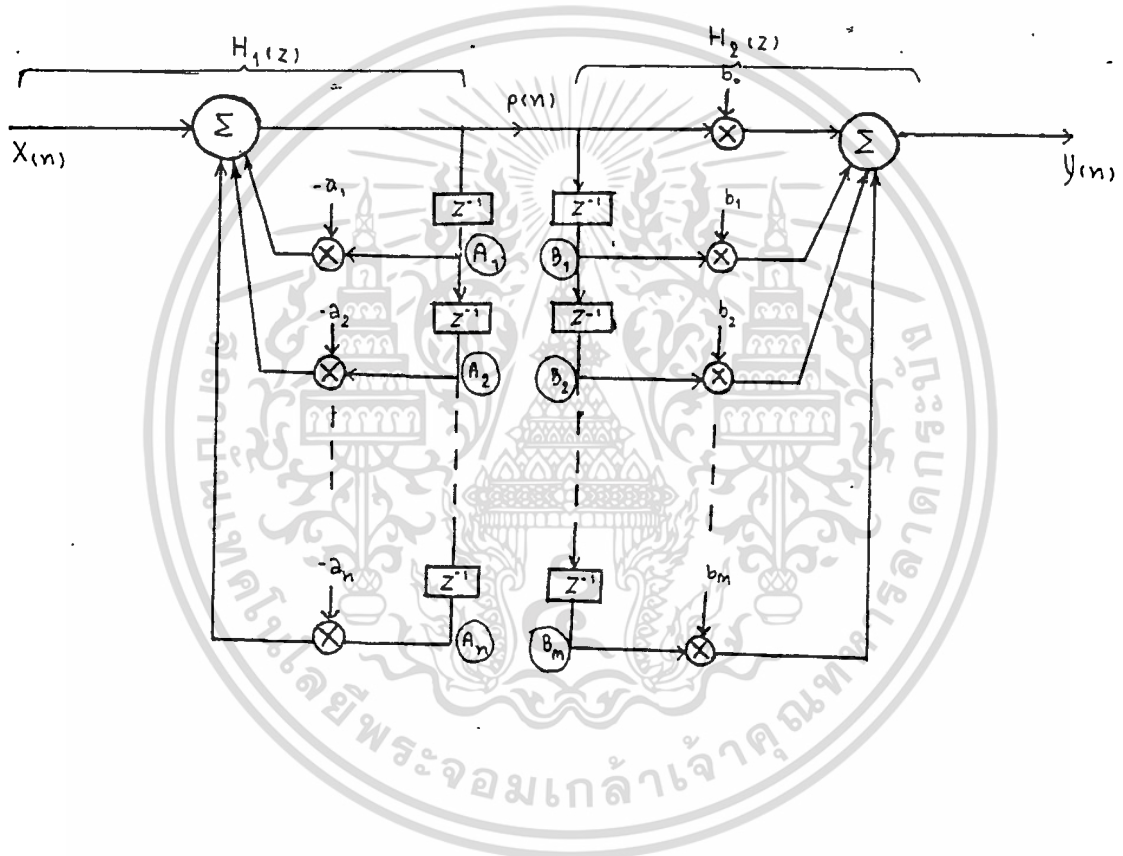
Inverse Z-transform จะได้

$$p(n) = x(n) - \sum_{k=1}^N a_k p(n-k) \quad (2.15)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$y(n) = \sum_{k=0}^M b_k p(n-k) \quad (2.16)$$

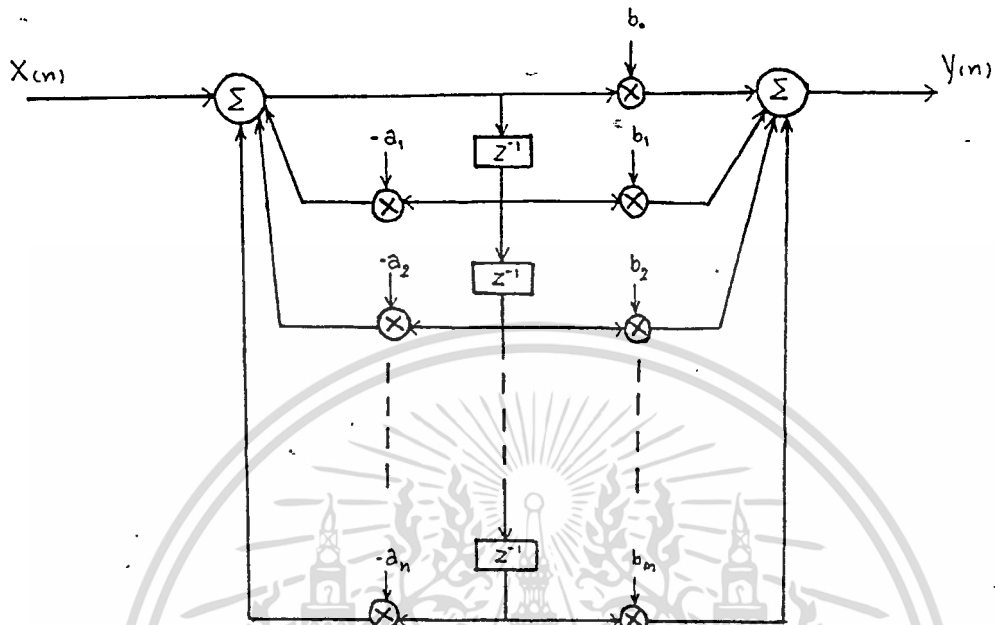
ซึ่งสามารถเขียนเป็น block diagram ของ Direct Form II ได้ดังรูป 2.11



รูปที่ 2.11 แสดง Direct Form II

สัญญาณที่จุด A_1 และ B_1 คือสัญญาณเดียวกันและสัญญาณที่จุด A_2, B_2 ก็คือสัญญาณเดียวกัน ดังนั้นเราสามารถนำ Unit delay รวมกันไว้จึงสามารถเขียน Direct form II ได้ทั้งหมดดังรูปที่ 2.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 แสดง Direct form II

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

สวิตช์คาปาซิเตอร์

การออกแบบวงจรกรองดิจิตอล เราอาศัยการแปลงมาจากสมการของวงจรกรองความถี่อนาล็อกซึ่งอยู่ในรูปของ s -domain แล้วทำการแปลงให้อยู่ในรูปของ z -domain เพื่อที่จะหาสมการของวงจรกรองความถี่แบบดิจิตอลออกมา ในที่นี้เราเลือกใช้การแปลงจาก s -domain ไปเป็น z -domain โดยการใช้นิพจน์ Bilinear transform โดยหลักการแล้วเราจะพบว่าวงจรกรองความถี่อนาล็อกแบบแอกทิฟจะประกอบไปด้วยอุปกรณ์ RC ซึ่งจะพบความไม่สะดวก นั่นก็คือว่า เมื่อเราต้องการเปลี่ยนค่าความถี่ cut off ก็ต้องเปลี่ยนค่า RC ด้วยเหตุนี้เองเราจึงสร้างวงจรกรองความถี่แบบดิจิตอลขึ้นมาเพื่อให้สามารถทำงานแทนวงจรกรองความถี่อนาล็อก เนื่องจากว่าวงจรกรองความถี่แบบดิจิตอลสามารถเปลี่ยนแปลงค่าความถี่ cut off ได้โดยไม่ต้องเปลี่ยนอุปกรณ์ใดๆให้ยุ่งยาก การสร้างวงจรกรองความถี่ดิจิตอลจะใช้ sw. capacitor คือการทำงานร่วมกันของ mos switch กับ capacitor มาทำงานแทนค่าความต้านทาน R ในที่นี้เราเลือก sw. capacitor ที่ต่อกันแบบ bilinear มาทำหน้าที่แทน R ในวงจรกรองความถี่อนาล็อก จะพบว่าถ้าเราต้องการเปลี่ยนแปลงค่าความถี่ cut off ก็เพียงแต่เปลี่ยนแปลงค่าความถี่ของสัญญาณป้อนที่มาจากควบคุมการทำงานของ sw. เท่านั้น

การสร้างวงจรกรองความถี่ดิจิตอล จาก sw. capacitor เรียกอีกอย่างหนึ่งว่า sw. capacitor filter ซึ่งเป็นวิธีการหนึ่งในหลาย ๆ วิธีของการสร้างวงจรความถี่ดิจิตอลทางฮาร์ดแวร์

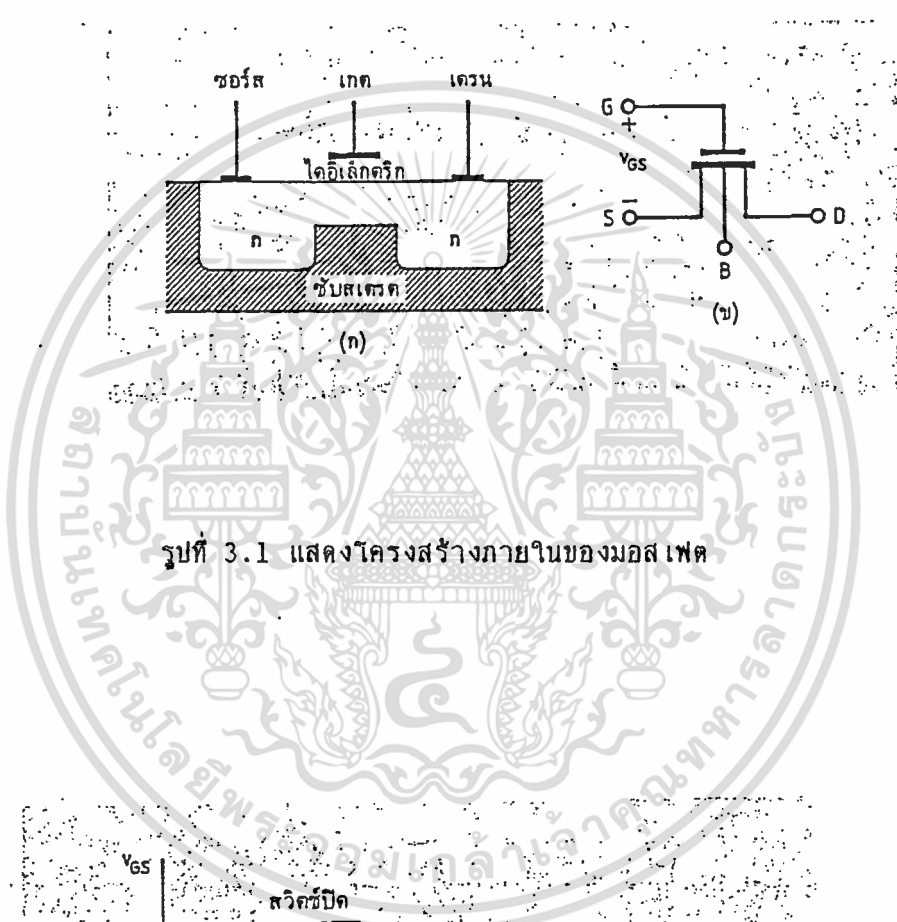
3.1 ทฤษฎีสวิตช์คาปาซิเตอร์

สวิตช์คาปาซิเตอร์ (switch capacitor) คือการทำงานร่วมกันของ มอสสวิตช์กับตัวเก็บประจุ มอสสวิตช์ (Mos-Metal Oxide Semiconductor) จากรูปที่

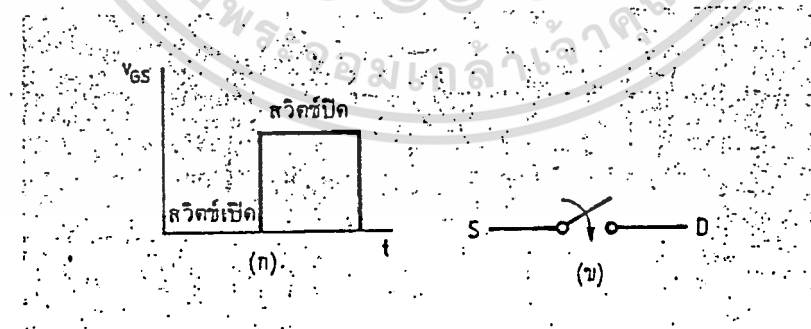
3.1 เป็นการตัดขวางของมอสที่เป็นทรานซิสเตอร์ชนิดหนึ่ง ที่เรียกว่ามอสเฟต (MOSFET : เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ได้รับอนุญาต) ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Metal Oxide Semiconductor Field Effect Transistor)

รูปที่ 3.2 เป็นสัญลักษณ์ของมอสเฟต ที่ขาเกต (Gate) เดรน (Drain) ซอร์ส (Source) และ
 ซับสเตรต (Substrate) V_{gs} คือแรงดันระหว่างเกตกับซอร์ส ปกติมีค่าเป็นศูนย์หรือมีค่าสูง
 กว่า V_{cr} (cut off regions คือ แรงดันที่ทำให้กระแสเดรนคงที่ ปกติมีค่า 1-2 โวลต์)
 V_{gs} จะเป็นตัวควบคุมให้เฟดปิดหรือเปิดระหว่างขาซอร์สกับเดรนจึง เรียกว่า มอสสวิทช์



รูปที่ 3.1 แสดงโครงสร้างภายในของมอส เฟต



รูปที่ 3.2 แสดงสภาวะ เปิดหรือปิดของมอสสวิทช์ซึ่งถูกควบคุมโดย V_{gs}

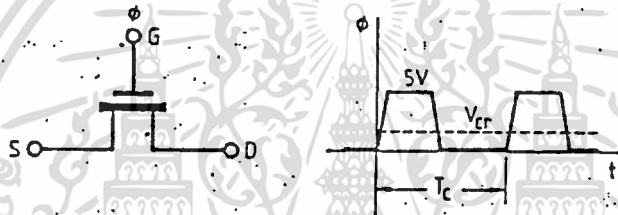
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น เมื่อนำไปเผยแพร่โดยไม่ได้รับอนุญาต
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$V_{gs} > V_{cr}$ ทำให้ FET ปิด R_{ds} (ความต้านทานระหว่างเดรนกับซอร์ส) ประมาณ $10\text{ k}\Omega$ (ค่านี้ขึ้นอยู่กับขนาดของ MOS)

$V_{gs} < V_{cr}$ ทำให้ FET เปิด R_{ds} ประมาณ $100 - 1000\text{ M}\Omega$

อัตราส่วนของความต้านทานทั้งสองสภาวะนี้มีค่าประมาณ 10^5 เท่า

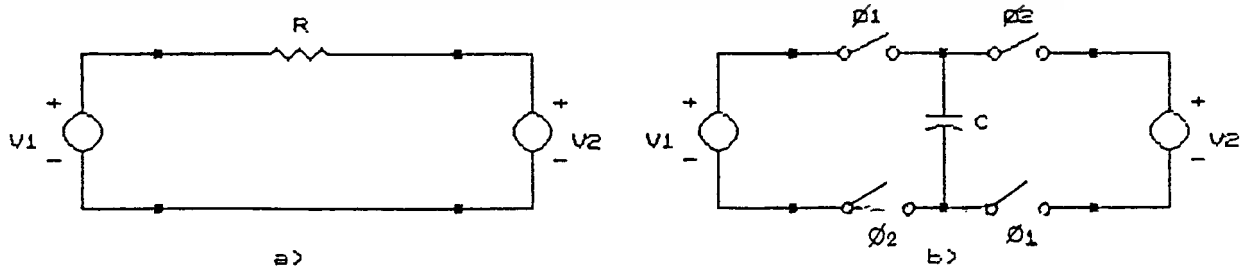
V_{gs} บ้อนด้วยสัญญาณนาฬิกา $T_c = 1/f_c$ ทำให้ mos จะเปิดปิดด้วยคาบเวลาคงที่



รูปที่ 3.3 แสดงรูปร่างสัญญาณนาฬิกาที่ใช้กระตุ้นการทำงานของมอส

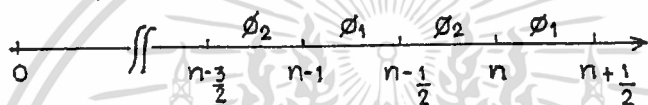
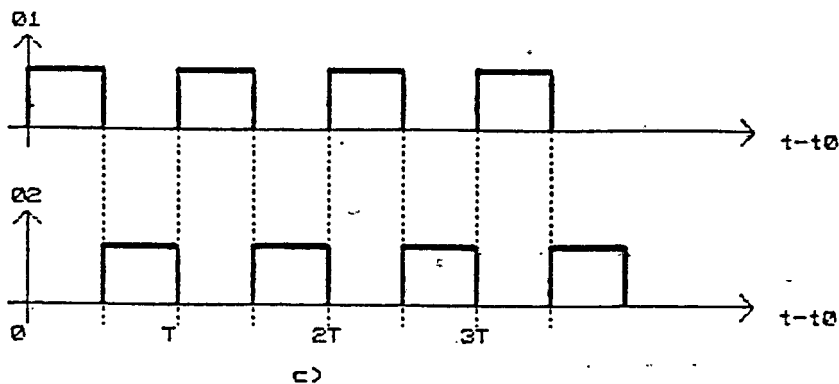
3.2 การแทนวงจรสมมูลของความต้านทานด้วยสวิตช์คาปาซิเตอร์

เมื่อเรานำสวิตช์มาต่อกับตัวเก็บประจุเพื่อทำหน้าที่แทนค่าความต้านทาน มีวิธีการต่ออยู่หลายแบบ แต่ในที่นี้เราเลือกการต่อแบบ bilinear



รูปที่ 3.4 แสดงการใช้ Bilinear sw.cap แทนค่าความต้านทาน R

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูผู้สอนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 แสดง clock phase

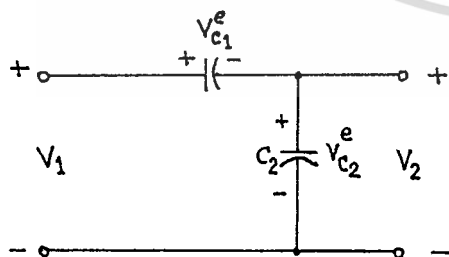
ถ้าเฟสคือ ϕ_2 เกิดขึ้นระหว่าง $(n-1/2)T$ และ nT จะแสดงวงจรเสมือนได้ดังรูปที่ 3.6(a)

จากรูปที่ 3.4 จะพบว่า c_1 ถูก charge มีศักย์ไฟฟ้าดังสมการ 3.1

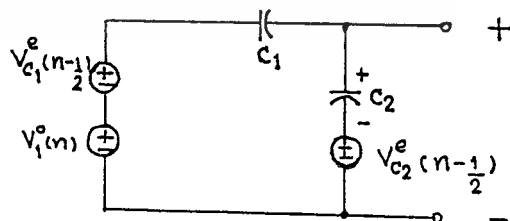
$$V_{c1}^{e(n-1/2)} = V_1^{e(n-1/2)} - V_2^{e(n-1/2)} \quad (3.1)$$

และ c_2 ก็ถูก charge มีศักย์ไฟฟ้าดังสมการ 3.2

$$V_{c2}^{e(n-1/2)} = V_2^{e(n-1/2)} \quad (3.2)$$



รูปที่ 3.6(a)



รูปที่ 3.6(b)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.6(b) แสดงวงจรเสมือนของรูปที่ 3.4 ในเฟสคี่ถัดไป ϕ_1 , $n \leq t/T < (n+1/2)$

$$\begin{aligned} q_L^0(n) &= c_2 v_2^0(n) \\ q_m^e(n-1/2) &= c_2 v_2^e(n-1/2) \end{aligned} \quad \text{----- (3.3)}$$

$$q^{0,e}_c = c_1 v_{c1}^e(n-1/2) + c_1 v_{c1}^0(n) \quad \text{----- (3.4)}$$

$$\text{เมื่อ } v_{c1}^0(n) = v_1^0(n) - v_2^0(n)$$

เปรียบเทียบสมการ 3.3 และ 3.4 และใช้ Z-transform สามารถเขียนใหม่ได้ว่า

$$\begin{aligned} q_L^0(t') &= q_m^e(t) + q^{0,e}_c(t) \quad , \quad t' > t \\ (c_2+c_1)v_2^0(z) &= z^{-1/2}(c_2-c_1)v_2^e(z) + c_1[v_1^e(z)z^{-1/2}+v_1^0(z)] \end{aligned} \quad \text{----- (3.5)}$$

ในทำนองเดียวกันพบว่า

$$(c_2+c_1)v_2^e(z) = z^{-1/2}(c_2-c_1)v_2^0(z) + c_1[v_1^0(z)z^{-1/2}+v_1^e(z)] \quad \text{----- (3.6)}$$

เมื่อบวกสมการ 3.5 และ 3.6 เข้าด้วยกันจะสามารถเขียนสมการทราנסเฟอร์ฟังก์ชันได้ดังนี้

$$\begin{aligned} H(z) &= \frac{V_2(z)}{V_1(z)} = \frac{v_2^e(z) + v_2^0(z)}{v_1^e(z) + v_1^0(z)} \\ &= \frac{1 + z^{-1/2}}{(1+\alpha) + (1-\alpha)z^{-1/2}} \end{aligned}$$

$$\text{เมื่อ } \alpha = c_2/c_1$$

จะพบว่าการแทนค่าความต้านทาน R ด้วย sw.cap ที่ต่อกันแบบ Bilinear

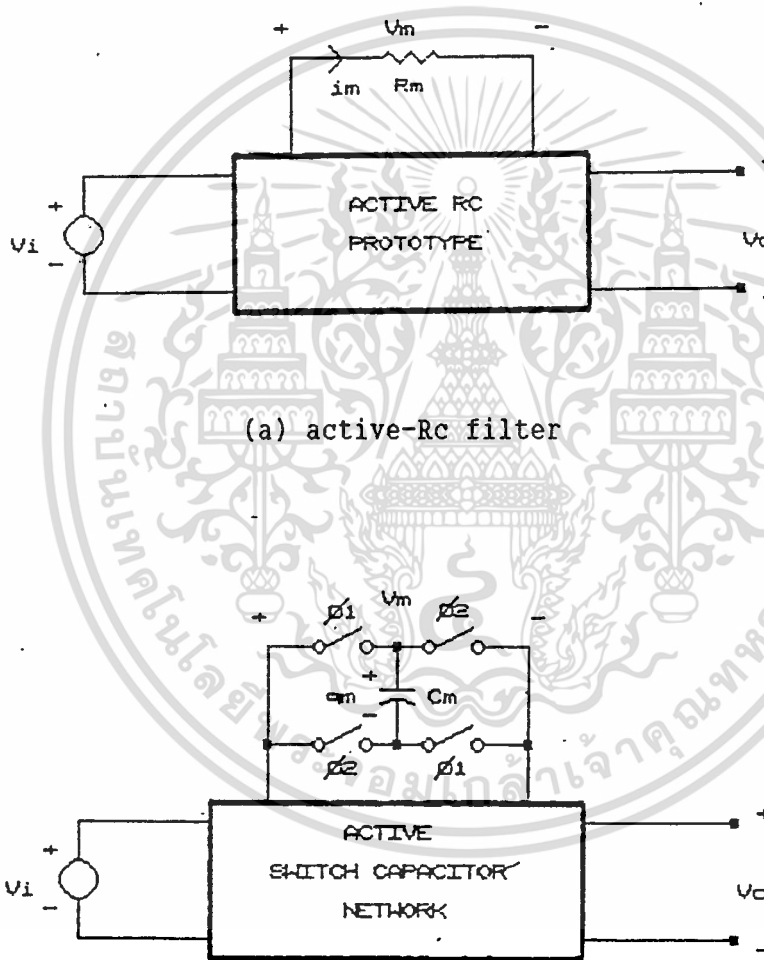
ความถี่ของสัญญาณพิกัดที่ใช้ควบคุมการทำงานของสวิทช์จะเป็น $T_C/2$ เนื่องจาก

ว่าสวิทช์ทุกตัวมีการเปลี่ยนตำแหน่งหรือการทำงาน 2 ครั้งในทุก ๆ T_C ดังนั้นจึงได้คาบ

เอกเวลายของสัญญาณพิกัดใหม่ คือ $T = T_C/2$ การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การออกแบบวงจรสวิตช์คาปาซิเตอร์ฟิลเตอร์โดยใช้การแทนที่ความต้านทาน ด้วยวิธี Bilinear Z-transform

การวิเคราะห์วงจร Active RC Filter เทียบกับ Active SC Filter



รูปที่ 3.6 แสดงวงจรสมมูลของวงจร Active Filter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิจารณาวงจร Active RC filter รูป (a)

$$q_m(t) = \int_{-\infty}^t i_m(t) dt \quad \text{----- (3.7a)}$$

$$= (1/R_m) \int_{-\infty}^t v_m(t) dt \quad \text{----- (3.7b)}$$

ใช้ Laplace transform เพื่อเขียนอยู่ในรูป s-domain

$$Q_m(s) = (1/sR_m) V_m(s) \quad \text{----- (3.8)}$$

ถ้าเราแทน R_m ในวงจรรูป 3.6a ด้วยวงจรรูป 3.6b คือการแทนความต้านทานด้วยการต่อสวิตช์คาปาซิเตอร์ที่ต่อแบบ Bilinear คือประกอบด้วยตัวเก็บประจุ C_m และสวิตช์ 4 ตัว (two toggles switches)

สมมุติให้สวิตช์ทุกตัวทำงานที่ทุก ๆ เวลา T ดังนั้นการเปลี่ยนแปลงของประจุในช่วงเวลา $(n-1) \leq t/T < n$ สามารถเขียนเป็นสมการได้ดังนี้

$$q_m(n) - q_m(n-1) = C_m [V_m(n) + V_m(n-1)] \quad \text{----- (3.9)}$$

ใช้ Z-transform จะได้

$$Q_m(z) - Z^{-1}Q_m(z) = C_m [V_m(z) + Z^{-1}V_m(z)] \quad \text{----- (3.10)}$$

$$Q_m(z) = C_m [(1+Z^{-1})/(1-Z^{-1})] V_m(z) \quad \text{----- (3.11)}$$

ถ้าเราให้ $C_m = T/2R_m$

เปรียบเทียบกับสมการ (3.8) และ (3.11) จะพบว่าการแทนค่าความต้านทาน R_m ด้วยสวิตช์คาปาซิเตอร์ C_m จะให้ความสัมพันธ์ออกมาดังนี้

$$s = \frac{2.1-z^{-1}}{1+z^{-1}} \quad \text{----- (3.12)}$$

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรียกสมการ (3.12) ว่า bilinear transform

$$\text{เมื่อ} \quad T = T_c/2$$

$$\text{ดังนั้น} \quad C_m = T_c/4R_m$$

การออกแบบวงจร active sc filter อยู่บนพื้นฐานของการแทนค่าความต้านทานแต่ละตัว ด้วยการทำงานของสวิตช์คาปาซิเตอร์ในวงจร analog RC filter โดยการใช้ Bilinear transform มีรายละเอียดดังนี้

1. ช่วงความถี่แบบผ่าน (passband) และช่วงความถี่ห้ามผ่าน (stopband)

ขึ้นอยู่กับค่าความถี่สัมพัทธ์ระหว่าง ω_d ของวงจร SC filter และ ω_a ของวงจร active RC prototype ดังนี้

$$\omega_a = \frac{2 \tan(\omega_d T/2)}{T}$$

2. วงจร active filter ออกแบบมาจากความถี่ cut off อนุลอก (ω_a)

3. ค่าความต้านทาน R_m ในวงจร active RC filter แทนด้วยสวิตช์คาปาซิเตอร์ (sc)

$$C_m = T/2R_m$$

แต่สวิตช์ทุกตัวมีการเปลี่ยนตำแหน่งหรือทำงาน 2 ครั้ง ที่ทุกา T_c ดังนั้น

$$T = T_c/2$$

$$C_m = T_c/4R_m$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ทฤษฎีทั่วไป เกี่ยวกับฟิลเตอร์

ดังได้ทราบมาแล้วว่าวงจรรองความถี่ดิจิทัล อาศัยการออกแบบมาจากวงจรรองความถี่อนาล็อก ซึ่งมีอยู่หลายประเภท เช่น Butterworth, Chebyshev และ Elliptic filter ในที่นี้จะขอกล่าวถึงเฉพาะวงจรรองความถี่อนาล็อกชนิด Butterworth และ Chebyshev เท่านั้น

วงจรรองความถี่อนาล็อกมีทรานสเฟอร์ฟังก์ชันดังนี้

$$H(s) = \frac{Y(s)}{X(s)} = \frac{K(s^L + b_{L-1}s^{L-1} + \dots + b_1s + b_0)}{s^N + a_{N-1}s^{N-1} + \dots + a_1s + a_0}$$

เมื่อ N คืออันดับของฟิลเตอร์

4.1 การออกแบบวงจรรองบัตเตอร์เวิร์ธโลพาสไฟโตรีโตนด์ (Butterworth Lowpass Prototype)

บัตเตอร์เวิร์ธฟิลเตอร์ เรียกอีกอย่างหนึ่งว่า maximally flat

ดังแสดงในรูปที่ 4.1

สามารถเขียนฟังก์ชันของ Magnitude-squared ได้ดังนี้

$$|H_{LP}(j\omega)|^2 = \frac{1}{1 + (\omega/\omega_c)^{2N}} \quad \text{----- (4.1)}$$

หรือ
$$|H_{LP}(j\omega)| = \frac{1}{[1 + (\omega/\omega_c)^{2N}]^{1/2}} \quad \text{----- (4.2)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ N คืออันดับของฟิลเตอร์

ω_c คือความถี่คutoff

เมื่อเรากำหนดค่าให้ $\omega_c = 1$ นั่นคือเรานอร์มอลไลซ์ให้ที่ $\omega_c = 1$ เป็นจุดคutoff จะเรียกฟิลเตอร์ชนิดนี้ว่า โพลีโทมัลโลพาสฟิลเตอร์ (Prototype Lowpass Filter)

สามารถเขียนสมการ (4.1) และ (4.2) ได้ใหม่ดังนี้

$$\left| H_{LP_p}(j\omega) \right|^2 = \frac{1}{1 + \omega^{2N}} \quad \text{----(4.3)}$$

$$\left| H_{LP_p}(j\omega) \right| = \frac{1}{[1 + \omega^{2N}]^{1/2}} \quad \text{----(4.4)}$$

พิจารณาสมการ (4.3) และ (4.4) จะพบว่า

1. ถ้า $\omega = 1$ จะพบว่าค่าแมกนิจูด (Magnitude) $M = \left| H_{LP_p}(j\omega) \right| = -3 \text{ dB}$ หรือ $M^2 = \left| H_{LP_p}(j\omega) \right|^2 = 1/2$ ที่ทุก ๆ อันดับของฟิลเตอร์
2. ที่ $\omega = 0$ ทำให้ $\left| H_{LP_p}(j\omega) \right| = 1$ เป็นค่าสูงสุด (maximum value) ทำให้เรียกได้ว่า maximally flat
3. แทน ω^2 ด้วย $-s^2$ ($s=j\omega$) ในสมการ (4.3) จะได้ว่า

$$H(s)H(-s) = \frac{K}{1 + (-s^2)^N}$$

$$H(s) = \frac{K}{Q(s)} \quad \text{----(4.5)}$$

$$Q(s)Q(-s) = 1 + (-s^2)^N$$

ถ้า $N = 2$ จะได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและตั้งอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{aligned}
 Q(s)Q(-s) &= 1 + s^4 \\
 &= 1 + s^4 + 2s^2 - 2s^2 \\
 &= s^4 + 2s^2 + 1 - 2s^2 \\
 &= (s^2 + 1) - (\sqrt{2}s)^2 \\
 &= (s^2 + \sqrt{2}s + 1)(s^2 - \sqrt{2}s + 1)
 \end{aligned}$$

ดังนั้น $Q(s) = s^2 + \sqrt{2}s + 1$ นำไปแทนใน (4.5) ได้

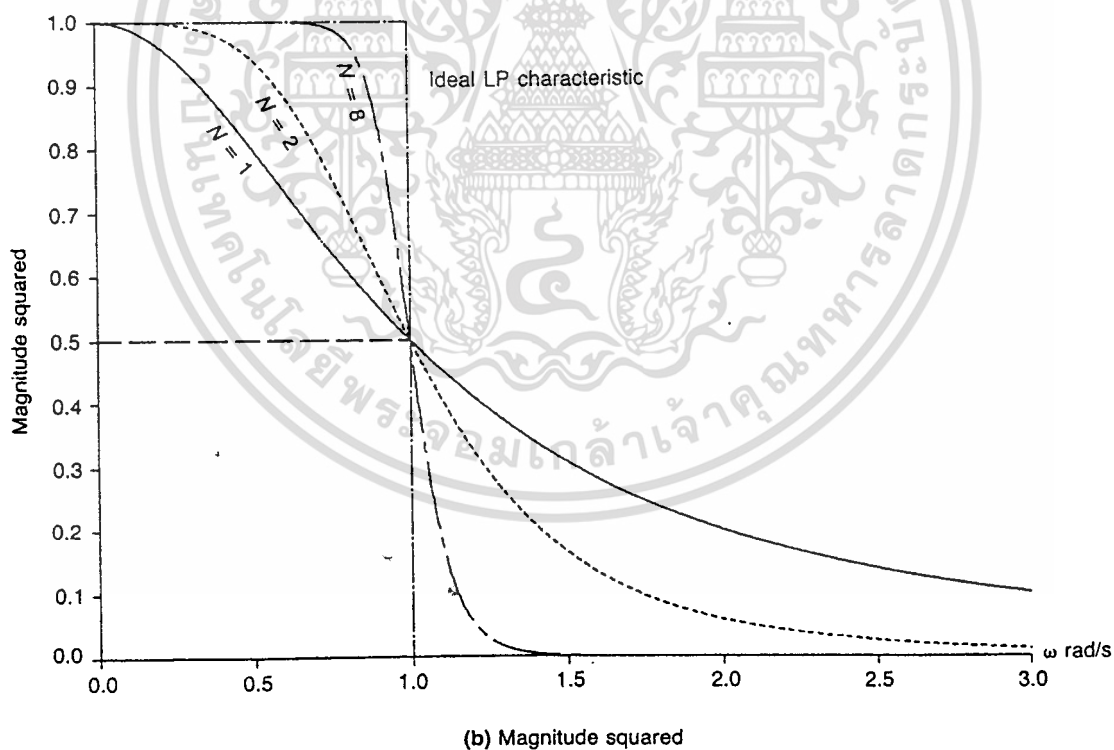
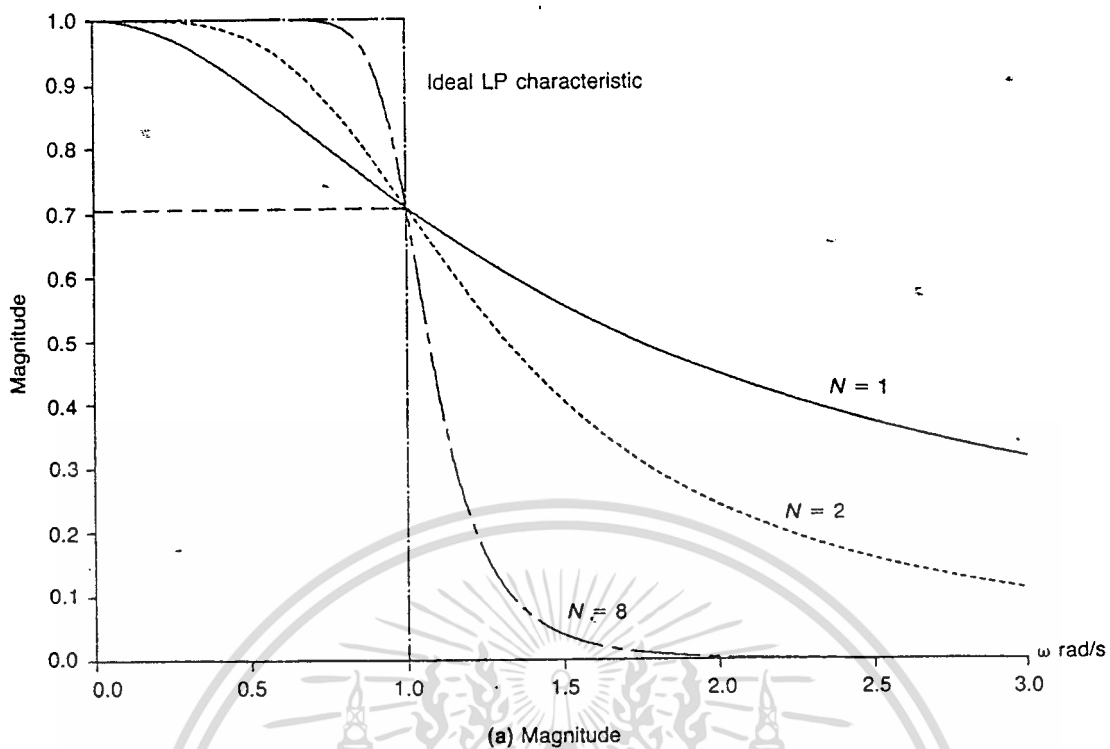
$$H(s) = \frac{K}{s^2 + \sqrt{2}s + 1} \quad \text{-----(4.6)}$$

เรียกสมการที่ (4.6) ว่า Butterworth lowpass filter อันดับที่ 2 และ $s^2 + \sqrt{2}s + 1$ เรียกว่า Butterworth Polynomial of Second order LPF สามารถเขียนรูปทั่วไปของสมการ (4.5) ได้ว่า

$$Q(s) = s^N + b_{N-1}s^{N-1} + \dots + b_1s + b_0$$

ซึ่งแสดงให้เห็นดังตารางที่ 4.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



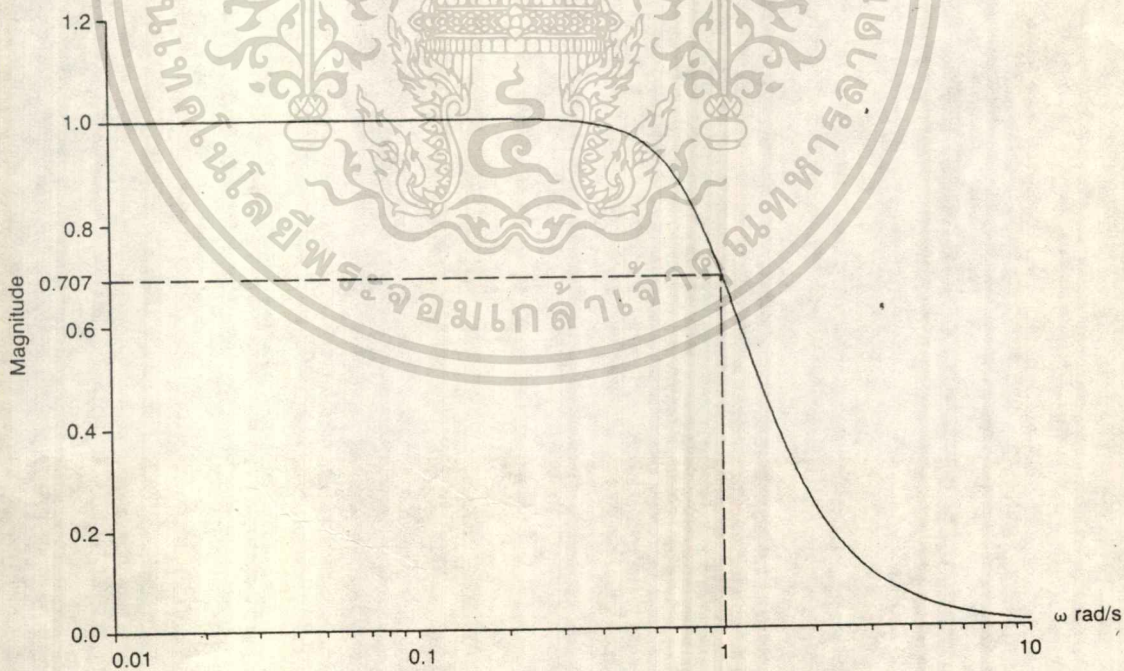
รูปที่ 4.1 แสดง Magnitude ของ Butterworth Lowpass filter ที่อันดับต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

N	a ₁	a ₂	a ₃	a ₄	a ₅	a ₆	a ₇	a ₈
1	1.0000							
2	1.4141	1.0000						
3	2.0000	2.0000	1.0000					
4	2.6131	3.4142	2.6131	1.0000				
5	3.2361	5.2361	5.2361	3.2361	1.0000			
6	3.8637	7.4641	9.1416	7.4641	3.8637	1.0000		
7	4.4940	10.0978	14.5918	14.5918	10.0978	4.4940	1.0000	
8	5.1258	13.1371	21.8462	25.6884	21.8462	13.1371	5.1258	1.0000

$$H_{LP}(s) = \frac{1}{1 + a_1s + a_2s^2 + \dots + a_Ns^N}$$

ตารางที่ 4.1 แสดง Butterworth prototype polynomial ที่อันดับต่าง ๆ



รูปที่ 4.2 แสดง Frequency response magnitude ของ lowpass prototype filter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การออกแบบวงจรฟิลลิปโซโลพาสโฟโตไทด์

(Chebyshev Lowpass Prototype)

ฟิลลิปโซฟิลเตอร์ เรียกอีกอย่างหนึ่งว่า Equal ripple filter เนื่อง
จากในช่วงความถี่แถบผ่านจะมีการกระเพื่อมของแอมพลิจูดด้วยขนาดที่คงที่ ดังแสดงในรูปที่ 4.3
สามารถเขียนฟังก์ชันของ Magnitude-squared ได้ดังนี้

$$\left| H_{LP_p}(j\omega) \right|^2 = \frac{1}{1 + \epsilon^2 C_N^2(\omega)} \quad \text{-----(4.7)}$$

เมื่อ N คืออันดับของฟิลเตอร์

มีค่า $0 < \epsilon < 1$ เป็นตัวกำหนดการกระเพื่อมของแอมพลิจูด
(ripple amplitude) ในช่วงความถี่แถบผ่านมีค่าเท่ากับ

$$\gamma_{dB} = 10 \log(1 + \epsilon^2)$$

ที่ความถี่คutoff จากสมการ (4.7) จะได้ว่า

$$\left| H_{LP_p}(j\omega) \right|^2 = 1/2, \quad \text{นั่นคือ } \epsilon^2 C_N^2(\omega) = 1$$

เมื่อ $C_N(\omega)$ คือ Chebyshev polynomial อันดับที่ N

$$C_N(\omega) = \cos(N \cos^{-1} \omega), \quad 0 \leq \omega \leq 1$$

$$C_N(\omega) = \cosh(N \cosh^{-1} \omega), \quad \omega > 1$$

จาก Magnitude function
$$\left| H_{LP_p}(j\omega) \right| = \frac{1}{[1 + \epsilon^2 C_N^2(\omega)]^{1/2}}$$

พิจารณาที่ $\omega=0$ $C_N^2(0) = 0$, เมื่อ N เป็นจำนวนคี่ ดังนั้น $|H(j0)| = 1$
 $= 1$, เมื่อ N เป็นจำนวนคู่ ดังนั้น $|H(j0)| = 1/[1 + \epsilon^2]^{1/2}$

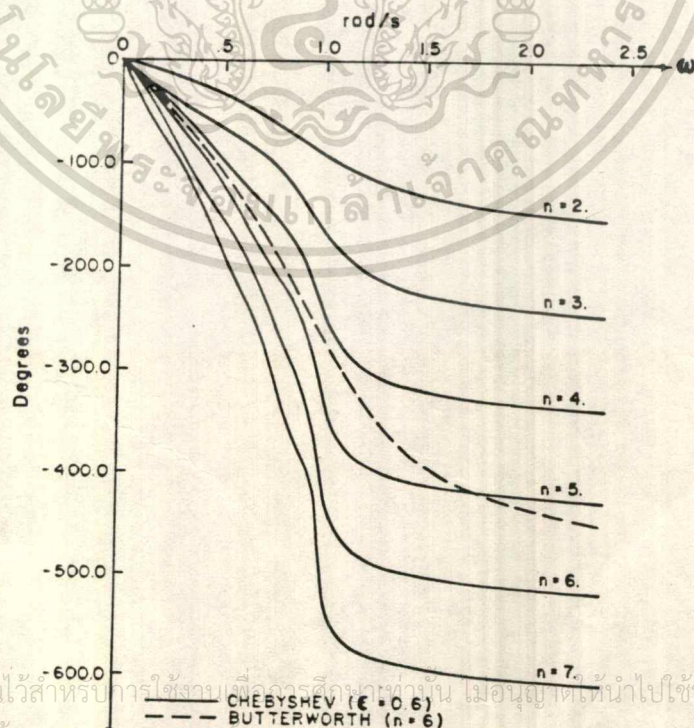
พิจารณาที่ $\omega=1$ $C_N^2(1) = 1$ ที่ทุก ๆ อันดับ ดังนั้น $|H(j1)| = 1/[1 + \epsilon^2]^{1/2}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

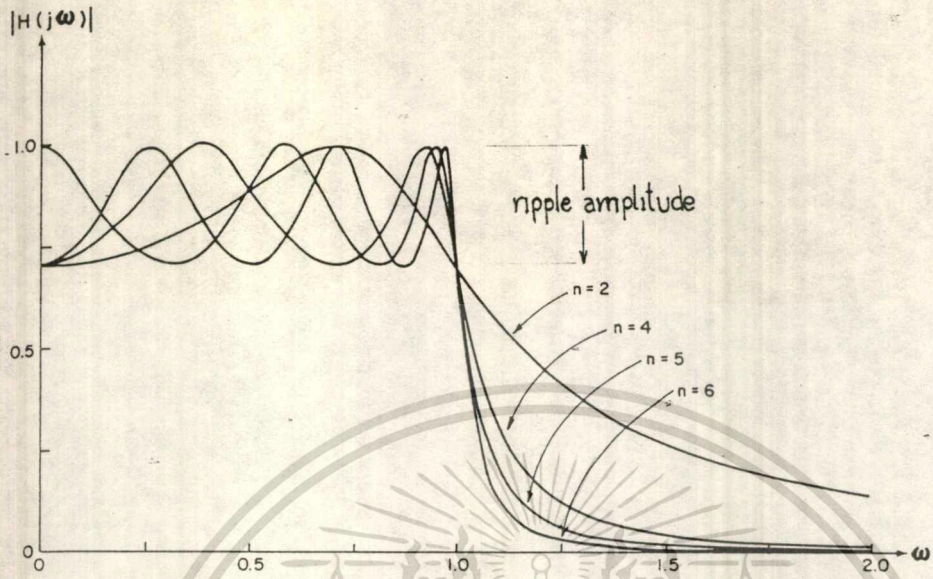
- จะพบว่าที่ $\omega = 1$ $|H(j\omega)| > 1/\sqrt{2}$ หรือมากกว่า -3 dB เนื่องจาก $\epsilon < 1$
- prototype ripple bandwidth อยู่ในช่วง $0 < \omega < 1$
- Frequency Response ของ Chebyshev filter ขึ้นอยู่กับ ϵ และ N
 1. ถ้ากำหนดอันดับที่ N คงที่แล้วเพิ่ม ϵ จะทำให้ขนาดของการกระเพื่อมมากขึ้น (large ripple) เนื่องจาก $\delta_{dB} = 10 \log (1 + \epsilon^2)$ แต่จะทำให้ characteristic ในช่วง stopband ตีขึ้น
 2. ถ้ากำหนดค่า ϵ คงที่แล้วเพิ่มอันดับที่ N ก็จะทำให้ characteristic ในช่วง stopband ตีขึ้นเช่นกันแต่การกระเพื่อมของแอมพลิจูดจะยิ่งมีมากขึ้นเมื่ออันดับที่ N สูงขึ้น

เฟสเรสปอนส์ (Phase response)

เฟสเรสปอนส์ของวงจรรองความถี่เป็นคุณลักษณะหนึ่งที่มีความสำคัญ เช่นเดียวกับ แอมพลิจูดเรสปอนส์ ในการสร้างวงจรรองความถี่ที่มีประสิทธิภาพจะต้องพิจารณาทั้งสองส่วนควบคู่กันไป โดยจะพบว่าวงจรรองความถี่ที่อันดับค่า n จะมีความเป็นเชิงเส้น (linear) ในช่วงความถี่แถบผ่าน ($0 < \omega < 1$) ต่ำกว่าที่อันดับสูง n และพบว่าวงจรรองความถี่แบบบัตเตอร์เวิร์ธจะมีความเป็นเชิงเส้นของเฟสมากกว่าวงจรรองความถี่แบบเชบชีฟด้วย ดังแสดงให้เห็นโดยกราฟข้างล่าง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานที่จำกัดเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ (ω) ใดเปลี่ยนแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 แสดง chebyshev prototype lowpass ที่ $\epsilon = 0.5088$

N	$C_N(\omega)$
1	ω
2	$2\omega^2 - 1$
3	$4\omega^3 - 3\omega$
4	$8\omega^4 - 8\omega^2 + 1$
5	$16\omega^5 - 20\omega^3 + 5\omega$
6	$32\omega^6 - 48\omega^4 + 18\omega^2 - 1$
7	$64\omega^7 - 112\omega^5 + 56\omega^3 - 7\omega$
8	$128\omega^8 - 256\omega^6 + 160\omega^4 - 32\omega^2 + 1$

ตารางที่ 4.2 แสดง chebyshev polynomail

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1/2-dB ripple ($\epsilon = 0.3493$, $\epsilon^2 = 0.1220$)	
N	
1	$s + 2.863$
2	$s^2 + 1.425s + 1.516$
3	$s^3 + 1.253s^2 + 1.535s + 0.716$
4	$s^4 + 1.197s^3 + 1.717s^2 + 1.025s + 0.379$
5	$s^5 + 1.173s^4 + 1.937s^3 + 1.310s^2 + 0.753s + 0.179$

1-dB ripple ($\epsilon = 0.5088$, $\epsilon^2 = 0.2589$)	
N	
1	$s + 1.965$
2	$s^2 + 1.098s + 1.103$
3	$s^3 + 0.988s^2 + 1.238s + 0.491$
4	$s^4 + 0.953s^3 + 1.454s^2 + 0.743s + 0.276$
5	$s^5 + 0.937s^4 + 1.689s^3 + 0.974s^2 + 0.581s + 0.123$

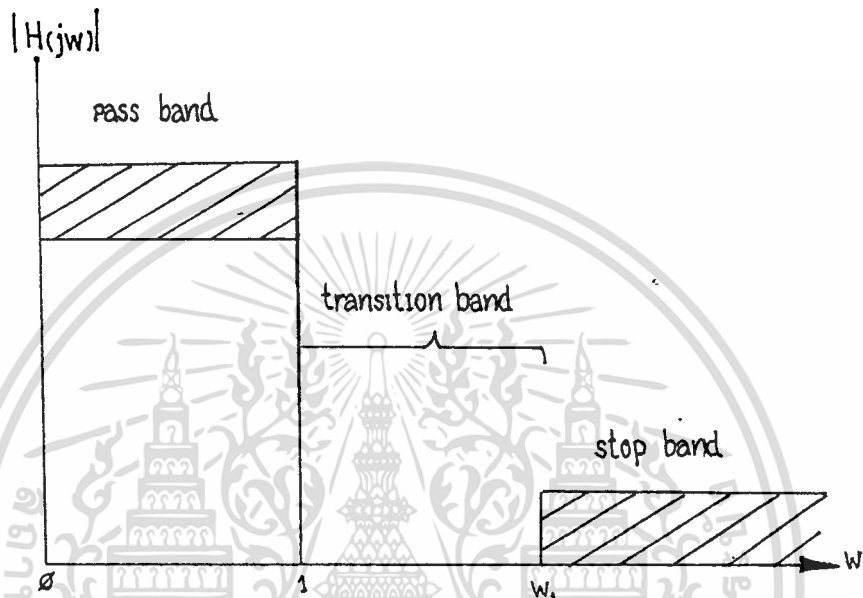
2-dB ripple ($\epsilon = 0.7648$, $\epsilon^2 = 0.5849$)	
N	
1	$s + 1.308$
2	$s^2 + 0.804s + 0.823$
3	$s^3 + 0.738s^2 + 1.022s + 0.327$
4	$s^4 + 0.716s^3 + 1.256s^2 + 0.517s + 0.206$
5	$s^5 + 0.707s^4 + 1.500s^3 + 0.694s^2 + 0.459s + 0.082$

ตารางที่ 4.2 แสดง chebyshev prototype denominator polynomials

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อเปรียบเทียบระหว่าง Butterworth และ chebyshev filter

1. Chebyshev filter มีช่วงความถี่แถบผ่าน (passband) และช่วงความถี่ห้ามผ่าน (stopband) ที่ใกล้เคียงกับอุดมคติมากกว่า Butterworth filter
2. Chebyshev มี transition band ($1 < \omega < \omega_1$) น้อยกว่า



3. แอมพลิจูดของ Butterworth ที่ $\omega = 0$ จะ maximally flat มากกว่า
4. Chebyshev filter อันดับที่ 3 จะมีประสิทธิภาพดีเท่ากับ Butterworth filter อันดับที่ 5
5. Chebyshev filter ที่อันดับต่ำ ๆ จะมีความเป็นเชิงเส้นของเฟสมากกว่าที่อันดับสูง ๆ

4.3 การเปลี่ยนแปลงความถี่ (Frequency Transformation)

เป็นการเลื่อนตำแหน่งของ Frequency Cutoff จากที่นอร์มอลไลซ์เอาไว้ในสมการ prototyp lowpass filter จาก $\omega_c = 1$ rad/sec ไปยังตำแหน่งของ Frequency Cutoff ที่เราต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. แปลงความถี่จาก lowpass prototype ไปเป็น lowpass filter

(unnormalize)

$$\text{จากรานส์เฟอ์ฟังก์ชันของ lowpass prototype } H_{LP}(s) = \frac{Q(s)}{D(s)}$$

$$\text{ไปเป็นทรานส์เฟอ์ฟังก์ชันของ lowpass filter } H_{LP}(s) = H_{LP_p}(s) \Big|_{s=s/w_c}$$

คือแทน s ในสมการ lowpass prototype ด้วย s/w_c ก็จะได้ Lowpass filter ที่มีความถี่คutoff ที่ w_c ใด ๆ

2. แปลงความถี่จาก lowpass prototype ไปเป็น highpass filter

(unnormalize)

$$\text{ทรานส์เฟอ์ฟังก์ชัน ของ highpass filter } H_{HP}(s) = H_{LP_p}(s) \Big|_{s=w_c/s}$$

คือแทน s ในสมการ lowpass prototype ด้วย w_c/s ก็จะได้ highpass filter ที่ความถี่คutoff w_c ใด ๆ นั่นคือเราจะพบว่าการสร้าง lowpass filter, highpass filter อาศัยการแปลงจาก lowpass prototype

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

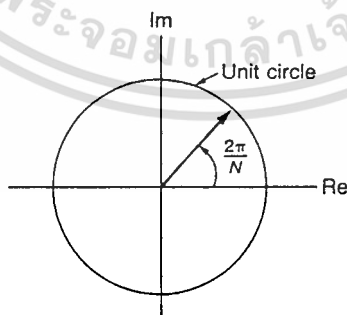
บทที่ 5

DISCRETE FOURIER TRANSFORM

เนื่องจากผลการแปลงฟูรีเยอร์ เป็น เครื่องมือทางคณิตศาสตร์ที่มีประโยชน์ โดยเฉพาะในการวิเคราะห์และศึกษาระบบทางอิเล็กทรอนิกส์ การมีวิธีการคำนวณการแปลงที่มีประสิทธิภาพและเหมาะสมที่ใช้ในการคำนวณโดยใช้คอมพิวเตอร์จึง เป็นสิ่งที่จำเป็น นิยามของการแปลงฟูรีเยอร์การคำนวณ เป็นการหาค่าอินทิกรัล ซึ่งอาจหาค่าตอบได้หรือไม่ได้ขึ้นกับลักษณะของฟังก์ชันโดยใช้คอมพิวเตอร์จะเป็นการทำการวนรอบ เพื่อหาค่าตอบซึ่งใช้ เวลาในการคำนวณนานและไม่แน่ว่าจะมีการลู่เข้าของค่าตอบหรือไม่ ในทางกลับกัน ถ้าหากเราสามารถมีการแปลงแบบใหม่ที่การคำนวณใช้ข้อมูลจากค่าให้ค่าใกล้เคียงกับการแปลงเดิม แต่ใช้เวลาในการคำนวณน้อย การแปลงใหม่นี้จะเป็นประโยชน์อย่างมาก เรียกการแปลงแบบนี้ว่า Discrete Fourier Transform

DFT เป็นผลบวกของ เอ็กโปเนนเชียลเชิงซ้อน (finite sum of complex exponential) ซึ่งเป็นตัวแทนของลำดับตัวเลขความยาวจำกัด (finite duration sequence) เพื่อที่จะบอกว่าในลำดับตัวเลข เหล่านี้ประกอบไปด้วยความถี่ใดบ้าง ซึ่งลำดับตัวเลขนี้ได้มาจากการสุ่มตัวอย่างจากข้อมูลที่เป็นอนาล็อก

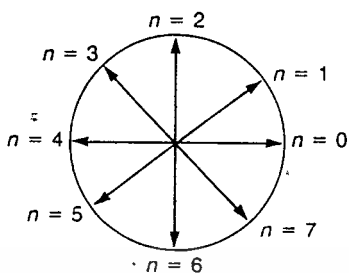
5.1 เอ็กโปเนนเชียลเชิงซ้อน (Complex exponential)



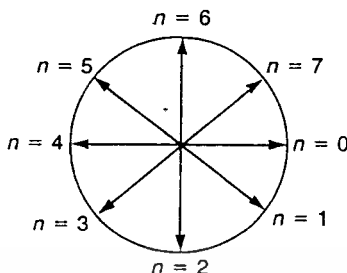
$2\pi/N$ เรเดียนคือ 1 sample ของลำดับสัญญาณที่เป็นรายคาบ (the periodic sequence) นั่นคือ 2π เรเดียนมีจำนวนทั้งหมด N sample

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$P_k(n) = e^{j(2\pi/N)nk} \quad \text{-----(5.1)}$$



$$(a) p_1(n) = e^{j(1 \cdot 2\pi/8)n} \quad k=1$$



$$(b) p_7(n) = e^{j(7 \cdot 2\pi/8)n} \quad k=7$$

เมื่อ $n = 0, 1, 2, \dots, 7$

$N = 8$

จะพบว่าที่ $k = N-1$ จะเป็นจุดที่ sampling มาครบรอบพอดี และพบว่าที่ $k=N$ จะไปซ้ำกับ $k=0$

ดังนั้นจากสมการ (5.1) เขียนได้ใหม่ว่า

$$P_k(n) = e^{j(2\pi/N)nk}, \quad k = 0, 1, 2, \dots, N-1$$

5.2 นิยามของ Discrete Fourier Transform

ถ้าให้ $x_p(t)$ เป็นฟังก์ชันต่อเนื่องและมีคุณสมบัติ เป็นคาบ โดยที่มี

ค่าคาบเป็น T_p หรือเขียนได้ว่า

$$x_p(t) = x_p(t+mT_p) \quad \text{-----(5.2)}$$

โดยที่ m เป็นเลขจำนวนเต็มใด ๆ ในเชิงคณิตศาสตร์ฟังก์ชันแบบนี้อาจเขียนแทนได้ด้วยอนุกรมฟูรีเยอร์ได้ดังนี้

$$x_p(t) = \sum_{k=-\infty}^{\infty} X_p(k) \exp(j2\pi k f_0 t) \quad \text{-----(5.3)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ $f_0 = 1/T_p$ ซึ่งเป็นค่าความถี่หลัก และการเขียนนี้เป็นจริงสำหรับในช่วง $0 < t < T_p$ ส่วนสัมประสิทธิ์ของอนุกรมฟูริเยร์ $X_p(k)$ จะหาได้จาก

$$X_p(k) = (1/T_p) \int_0^{T_p} x_p(t) \exp(-j2\pi k f_0 t) dt \quad \text{-----}(5.4)$$

การเขียนสมการ (5.3) และ (5.4) ถึงแม้ว่า $x_p(t)$ เป็นค่าจริงและ $X_p(k)$ เป็นค่าเชิงซ้อน แต่โดยทั่วไปแล้วไม่ได้จำกัดว่าต้องเป็น เช่นนี้เสมอไป $x_p(t)$ อาจเป็นค่าเชิงซ้อนและ $X_p(k)$ เป็นค่าจริงหรือเป็นค่าเชิงซ้อนทั้งคู่ก็ได้ ข้อที่น่าสนใจก็คือ $x_p(t)$ เป็นฟังก์ชันแบบต่อเนื่องและสเปกตรัมของ $X_p(k)$ พบว่ามีค่าเป็นฟังก์ชันไม่ต่อเนื่อง (discrete)

ถ้าจะพิจารณาอนุกรมฟูริเยร์ในกรณีที่เราทำการสุ่มสัญญาณแบบต่อเนื่องและเป็นคาบ $x_p(t)$ การที่จะทำการสุ่มสัญญาณได้นั้นสัญญาณต้องมีแถบความถี่ปฏิบัติงานที่แน่นอน (ถ้าหากสัญญาณนั้น ๆ ไม่มีแถบความถี่ปฏิบัติงานหรือมีความกว้างมาก เราอาจทำการกำหนดแถบความถี่ปฏิบัติงานของมันได้โดยการป้อนผ่านวงจรกรองความถี่ต่ำเสียก่อน) ถ้าสมมุติว่ามีแถบความถี่ปฏิบัติงานเป็น f_1 หรือให้

$$X_p(k) = 0 \text{ เมื่อ } |k| > k_1$$

โดยที่ k_1 เป็นจำนวนเต็มใด ๆ ที่ทำให้ได้ความสัมพันธ์ของความถี่เป็น

$$k_1 * f_0 = f_1$$

กล่าวอีกนัยหนึ่งคือ เรากำหนดค่าให้สเปกตรัมของความถี่ที่เกินค่า f_1 มีค่าแอมพลิจูดเป็นศูนย์ ดังแสดงในรูป 5.1 (c) โดยอาศัยทฤษฎีของการสุ่มตัวอย่างสัญญาณที่กล่าวไว้ว่า ค่าความถี่ของสัญญาณสุ่มตัวอย่างอย่างน้อยที่สุดต้องเป็น 2 เท่าของแถบความถี่ปฏิบัติงาน หรือในกรณีนี้ต้องเท่ากับ $2f_1$ เพราะฉะนั้นช่วงเวลาสุ่มตัวอย่าง T ที่ใช้ในการสุ่มตัวอย่างสัญญาณนี้มีค่าเท่ากับ

$$T = 1/2f_1 = 1/2k_1 f_0 = T_p/2k_1$$

และถ้ากำหนดจำนวนคาบของสัญญาณเราทำการสุ่มออกเป็น N ช่วงด้วยกัน ทำ
าให้ได้ว่า $2k_1 = N$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นสัญญาณ $x_p(t)$ ที่อยู่ในช่วง $0 < t < T_p$ เมื่อถูก
 สุ่มออกเป็น N ช่วง สามารถเขียนสมการแทนได้เป็น

$$x_p(t/T) = \sum_{n=0}^{N-1} x_p(t/T) \cdot \delta(t/T-n) \quad \text{-----(5.5)}$$

โดยที่ t/T เป็นเวลาอนุกรมลอทซ์ (normalized time) ซึ่งนิยามในช่วง
 $0 \leq t/T < N$ เนื่องจากสัญญาณ $x(t/T)$ ในสมการ (5.5) เป็นสัญญาณแบบเป็นคาบ
 เช่นกัน ดังนั้นจึงอาจหาสัมประสิทธิ์ฟูริเยอร์ของมันได้คือ

$$X(k) = \int_{\emptyset}^{N-1} x(t/T) \exp\{-j2\pi k/N \cdot (t/T)\} d(t/T)$$

เมื่อแทนค่า $x(t/T)$ จาก (5.5) ได้

$$X(k) = \int_{\emptyset}^{N-1} \left\{ \sum_{n=0}^{N-1} x_p(t/T) \cdot \delta(t/T-n) \cdot \exp\{-j2\pi k/N \cdot (t/T)\} d(t/T) \right\}$$

และจากคุณสมบัติของอิมพัลส์ $\delta(t_0)$ ที่นิยามว่า

$$\int_{-\infty}^{\infty} f(t/T) \cdot \delta((t/T)-n) \cdot d(t/T) = f(n)$$

เพราะฉะนั้นทำให้ได้

$$X(k) = \sum_{n=0}^{N-1} x_p(n) \cdot \exp(-j2\pi kn/N) \quad \text{-----(5.6)}$$

สมการที่ (5.6) เรียกว่า สัมประสิทธิ์ของ DFT (DFT coefficient)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และโดยความจริงแล้วลำดับสัญญาณที่ไม่เป็นรายการ nonperiodic sequence $[x(n)]$ ซึ่งเป็นจุดที่ทำการสุ่มสัญญาณมา ก็สามารถพิจารณาได้ว่าเป็นส่วนหนึ่งของลำดับสัญญาณที่เป็นรายการ นั่นคือ

$$\begin{aligned} x(n) &= x_p(n) \quad , \quad 0 \leq n \leq N-1 \\ &= 0 \quad , \quad \text{elsewhere} \end{aligned}$$

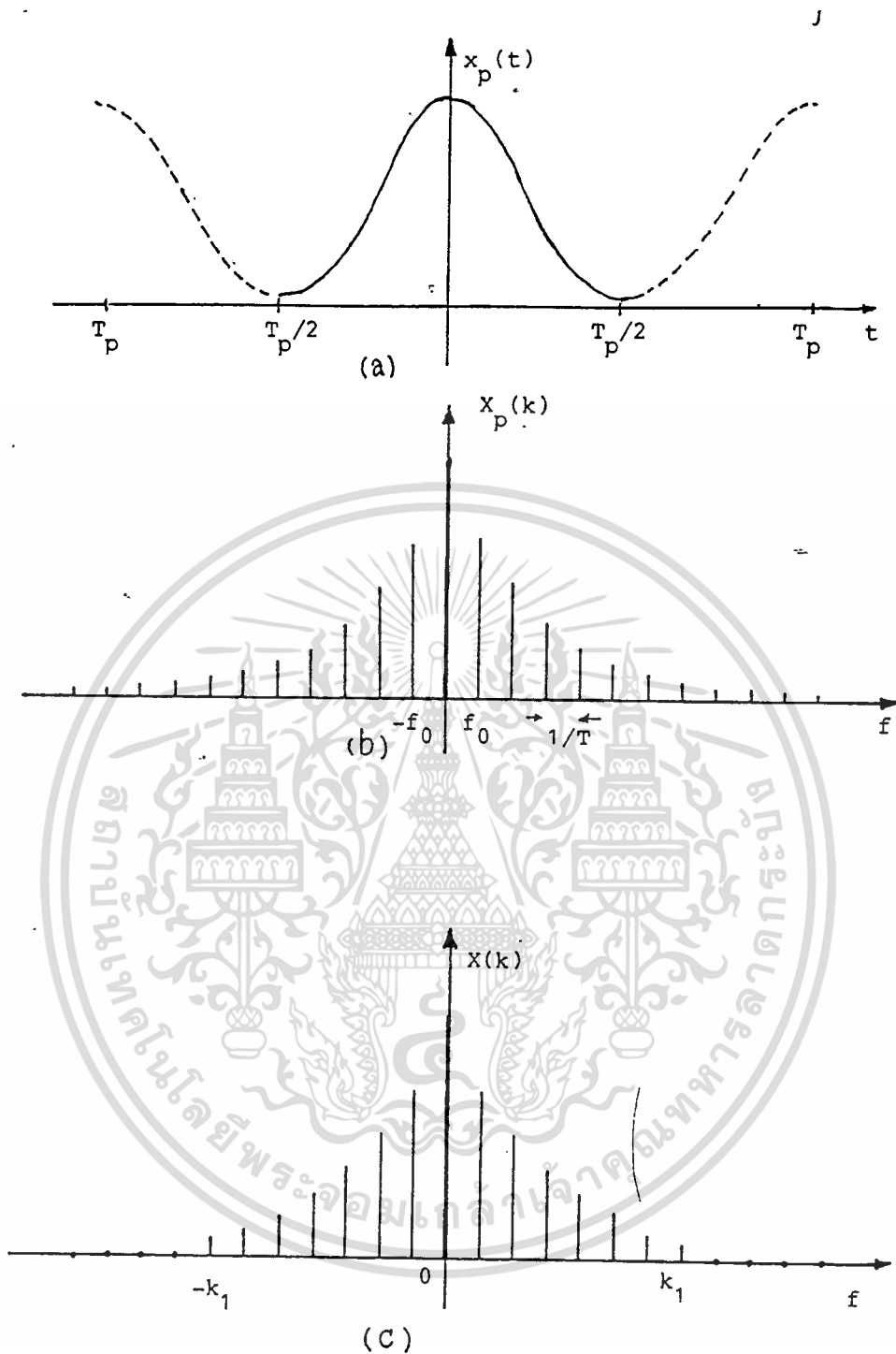
ซึ่งเราสามารถเขียนสมการ DFT ของลำดับสัญญาณ $x(n)$ ได้ดังนี้

$$X(k) = \sum_{n=0}^{N-1} x(n) \exp(-j2\pi kn/N) \quad , \quad k = 0, 1, 2, \dots, N-1$$

$$x(n) = (1/N) \sum_{k=0}^{N-1} X(k) \exp(j2\pi kn/N) \quad , \quad n = 0, 1, 2, \dots, N-1$$

จะพบว่า DFT ที่นิยามและทำการหาจากลำดับสัญญาณที่เป็นรายการนั้นก็
สามารถนำมาใช้คำนวณกับลำดับสัญญาณที่ยาวจำกัด หรือไม่ เป็นรายการได้ด้วยเช่นกัน

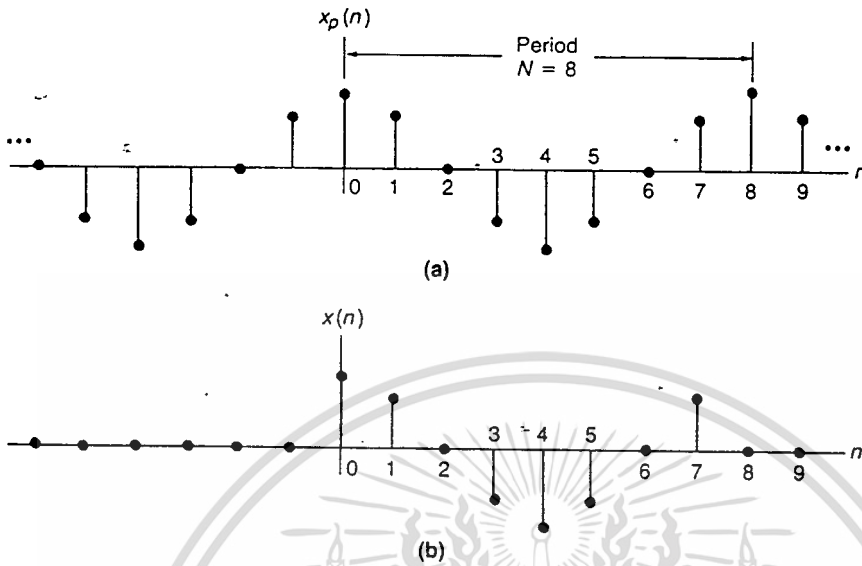
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.1 แสดงสเปกตรัมของสัญญาณที่มีแถบความถี่ปฏิบัติงานจำกัด

(a) สัญญาณเป็นคาบ (b) สเปกตรัม (c) สเปกตรัมเมื่อจำกัดแถบความถี่ปฏิบัติงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.2 (a) แสดงลำดับสัญญาณที่เป็นคาบ (periodic sequence $x_p(n)$)
 (b) แสดงลำดับสัญญาณไม่เป็นคาบ (nonperiodic sequence $x(n)$)

ลำดับสัญญาณที่ไม่เป็นคาบ $x(n)$ ได้จากการสุ่มจากสัญญาณต่อเนื่อง $x_p(t)$ สามารถเขียนได้ใหม่ $x(nT)$ และค่า DFT coefficient ก็สามารเขียนได้ใหม่ว่า $X(k\Delta f)$ เมื่อ Δf คือ frequency spacing of coefficient และขณะเดียวกัน T ใน $x(nT)$ ก็คือ time spacing of sample signal ซึ่งสามารถเขียนสมการ DFT ได้ใหม่ดังนี้

$$X(k\Delta f) = \sum_{n=0}^{N-1} x(nT) \exp(-j2\pi kn/N) \quad , k = 0, 1, 2, \dots, N-1$$

$$x(nT) = (1/N) \sum_{k=0}^{N-1} X(k\Delta f) \exp(j2\pi kn/N) \quad , n = 0, 1, 2, \dots, N-1$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าของ frequency spacing (Δf) ระหว่าง DFT coefficient [$X(k\Delta f)$] สามารถหาได้จาก

$$\Delta f = f_s/N = 1/NT \quad \text{-----(5.7)}$$

เมื่อ f_s คือความถี่ของการ sampling

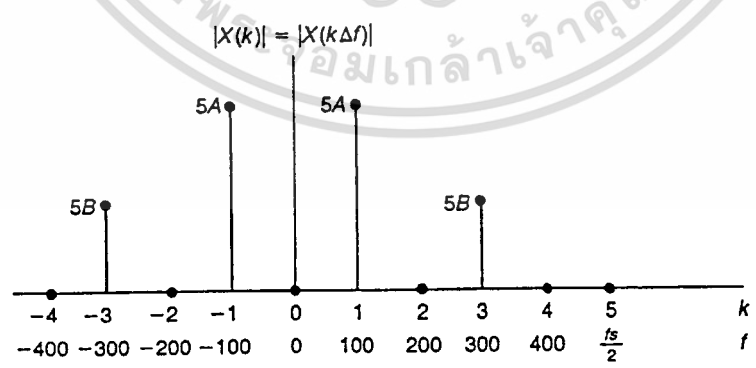
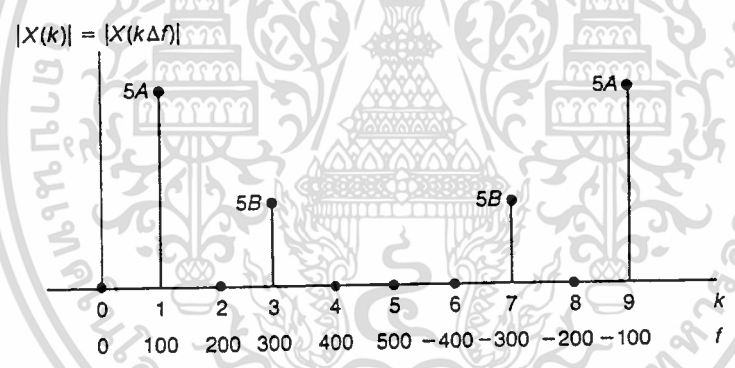
N คือจำนวนลำดับของสัญญาณใน 1 คาบ

T คือคาบเวลาของการ sampling

พบว่าเมื่อ Δf ยิ่งมีขนาดเล็กมาก ๆ จะสามารถแสดงองค์ประกอบทางความถี่ของสัญญาณ $x(n)$ ได้ละเอียดยิ่งขึ้น จากกราฟที่แสดงในรูปที่ 5.3 ซึ่งแสดงความสัมพันธ์ของ DFT coefficient กับความถี่ พบว่าที่จุด $f_s/2$ เป็นจุดที่เรียกว่า folding Frequency ซึ่งแสดงจุดสมมาตรของ DFT coefficient

ถ้า $f_s = 1000 \text{ Hz.}$, $N = 10$

$$\Delta f = f_s/N = 1000/10 = 100 \text{ Hz.}$$



รูปที่ 5.3 กราฟแสดง DFT coefficient กับความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 คุณสมบัติบางประการของ DFT

1. ความสัมพันธ์ระหว่าง DFT กับการแปลง Z

สมมุติว่าเรามีลำดับยาวจำกัด (finite duration sequence) ที่นิยามเฉพาะค่าในช่วง

$$x(n) \begin{cases} = x_p(n) & , 0 \leq n \leq N-1 \\ = 0 & , \text{elsewhere} \end{cases}$$

โดยให้ $x_p(n)$ เป็นลำดับสัญญาณที่มีคุณสมบัติเป็นคาบ และมีช่วงของคาบยาวเท่ากับ N ลำดับ สังเกตว่า $x(n)$ มีค่าเฉพาะในคาบเดียวของ $x_p(n)$ เพราะฉะนั้นการแปลง Z ของ $x(n)$ คือ

$$X(z) = \sum_{n=0}^{N-1} x(n) z^{-n}$$

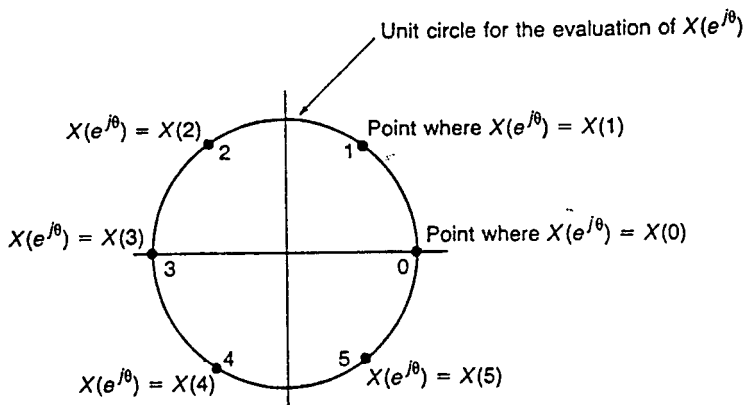
และเมื่อเราทำการหาค่า $X_p(z)$ ที่จุดต่าง ๆ บนเส้นรอบวงกลมหนึ่งหน่วยของระนาบ z ($z = e^{j\theta}$) เมื่อ $\theta = 2\pi k/N$ นั่นคือมุมที่ห่างกันเท่ากับ $2\pi k/N$ จะได้

$$\begin{aligned} X(z) \Big|_{z=e^{j\theta}} &= X(e^{j\theta}) \\ &= \sum_{n=0}^{N-1} x(n) \exp(-j2\pi kn/N) \quad \text{----- (5.8)} \end{aligned}$$

และจากสมการ (5.8) เมื่อกำหนดให้ $x_p(n) = x(n)$ ในช่วง $0 \leq n \leq N-1$ หา DFT ได้

$$X_p(k) = X[\exp(j2\pi k/N)] \quad \text{----- (5.9)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ผลแสดงให้เห็นถึงสมการ (5.8) และ (5.9) ทำให้สรุปได้ว่า สมประสิทธิ์ของ DFT ของลำดับยาวจำกัดนั้นก็คือ ผลการแปลง z ของลำดับนั้นที่ทำการหารอบ จุดที่ห่างเท่า ๆ กัน N จุดรอบวงกลมหนึ่งหน่วยนั่นเอง

2. คุณสมบัติเชิงเส้น (Linearity)

ถ้า $X(k)$ เป็น DFT coefficient ของ $x(n)$ และ $Y(k)$ เป็น DFT coefficient ของ $y(n)$

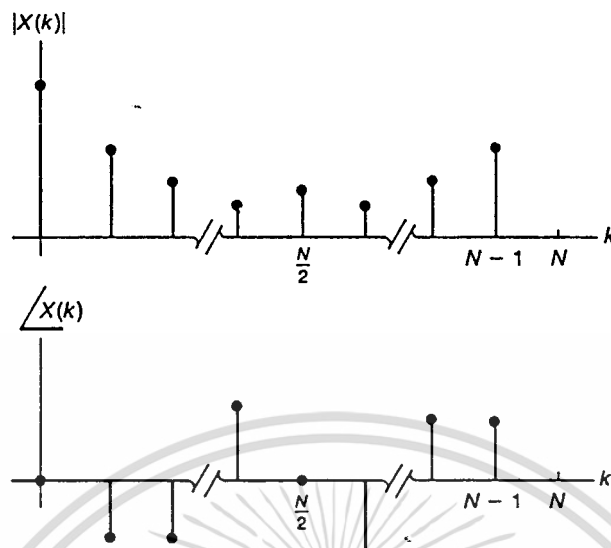
และถ้า $Z(m) = ax(n) + by(n)$ ดังนั้นผลการหา DFT coefficient $Z(k)$ จะเป็น

$$Z(k) = aX(k) + bY(k)$$

3. คุณสมบัติการสมมาตร (Symmetry Propoties)

จุดที่ทำให้แมกนิจูดของ DFT $X(k)$ และ เฟสของ DFT $X(k)$ สมมาตร ก็คือที่ตำแหน่ง $k = N/2$ นั่นคือพบว่าที่ตำแหน่ง $k > N/2$ และ $k < N/2$ จะมีลักษณะของ สัญญาณเป็นแบบเดียวกัน ในกรณีของ $X(k)$ ที่ตำแหน่งที่ $k > N/2$ จะประกอบไปด้วย ความถี่ในช่วงที่ติดลบ ซึ่งคุณสมบัตินี้สามารถใช้ตรวจเช็คความผิดพลาดการคำนวณค่า DFT ได้ ถ้าเกิดการไม่สมมาตรขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.4 (a) แสดงการสมมาตรของ DFT coefficient

(b) แสดงการสมมาตรของ phase DFT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

โปรแกรมการออกแบบ

วงจรกรองความถี่ดิจิทัล

1. การหาทรานส์เฟอร์ฟังก์ชันของวงจรกรองความถี่ดิจิทัล เราสามารถคำนวณหาได้จากสมการ prototype ของวงจรกรองความถี่แบบอนาล็อก ซึ่งมีให้เลือก 2 แบบ คือ สมการ prototype ของ Butterworth และของ Chebyshev แล้วทำการแปลงเป็นทรานส์เฟอร์ฟังก์ชันของวงจรกรองความถี่ดิจิทัล โดยการแปลงจากทรานส์เฟอร์ฟังก์ชันของวงจรกรองความถี่อนาล็อก ที่อยู่ในรูป s-domain ไปเป็น z-domain โดยใช้ Bilinear Transform คือแทน s ด้วย $\frac{2 \cdot 1 - z^{-1}}{1 + z^{-1}}$

$$H(z) = \frac{\sum_{k=0}^N b_k z^{-k}}{1 + \sum_{k=1}^N a_k z^{-k}}$$

ซึ่งก็คือทรานส์เฟอร์ฟังก์ชันของวงจรกรองความถี่ดิจิทัล

2. จากทรานส์เฟอร์ฟังก์ชันของวงจรกรองความถี่ดิจิทัล สามารถค่าสัมประสิทธิ์ $(a_1, a_2, \dots, a_N), (b_0, b_1, \dots, b_N)$ ซึ่งค่าสัมประสิทธิ์เหล่านี้จะเป็นตัวกำหนดคุณลักษณะ (characteristic) ของวงจร เราพบว่าวงจรกรองความถี่ดิจิทัลอันดับเดียวกันไม่ว่าจะเป็น LPF หรือ HPF ทรานส์เฟอร์ฟังก์ชันจะมีรูปแบบเดียวกัน แต่จะต่างกันที่ค่าสัมประสิทธิ์ของทรานส์เฟอร์ฟังก์ชันเท่านั้นเอง

3. ส่วนของโปรแกรมในการคำนวณหาคุณลักษณะของวงจรกรองความถี่ดิจิทัล (characteristic, frequency response) สามารถหาได้จากทรานส์เฟอร์ฟังก์ชันของวงจรกรองความถี่ดิจิทัลนั่นเอง โดยการแทนค่า $z = e^{j\omega T} = \cos(\omega T) + j\sin(\omega T)$ (เพื่อวิเคราะห์ในเทอมของความถี่ได้) แล้วทำการปรับค่า ωT ตั้งแต่ 0 จนถึง 2π หากค่าขนาด

แอมพลิจูดของทรานส์เฟอร์ฟังก์ชันที่แต่ละค่า ωT ออกมา ในที่นี้ที่จุด 2π จะแทนค่าหนึ่งของอาร์คไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความถี่ sampling (F_s) แต่เลือกใช้ ωT ตั้งแต่ 0 ถึง π เท่านั้น (ที่ $\omega T = 2\pi$ แทนตำแหน่งความถี่ sampling) เพราะพบว่าในช่วง π ถึง 2π เมื่อแปลงอยู่ในเทอมของ frequency domain แล้วจะได้ความถี่อยู่ในช่วงคิดลบ เมื่อนำความสัมพันธ์ระหว่างความถี่กับขนาดมาพล็อตกราฟก็จะแสดงให้เห็น characteristic ของวงจรกรองความถี่ดิจิทัลที่ต้องการ

4.4.1 ส่วนของโปรแกรมที่เขียนแบบการทำงานของวงจรกรองความถี่ดิจิทัล เราจะแทนสัญญาณ input ของระบบ $[x(n)]$ ด้วยค่าของตัวเลขที่เป็น real number โดยการสุ่มผ่านทางคีย์บอร์ด กำหนดค่าให้ลำดับของสัญญาณ input มี 20 ลำดับ

4.4.2 นำสัญญาณ input $[x(n)]$ เช่น (0,5,1,2,) มาผ่านระบบใน time domain (หา Difference Equation ของสมการ Digital Filter) จะได้ค่า output $[y(n)]$ ที่แสดง impulse response ของระบบ

4.4.3 นำสัญญาณ input $[x(n)]$ ของระบบมาวิเคราะห์ในเทอมของ Frequency domain โดยการใช้ Discrete Fourier Transform (DFT) เพื่อหาองค์ประกอบทางความถี่ของสัญญาณ input

4.4.4 นำสัญญาณ output $[y(n)]$ ของระบบที่ได้จากการคำนวณใน 4.4.2 ซึ่งอยู่ในเทอมของ time domain มาวิเคราะห์ในเทอมของ frequency domain โดยใช้ DFT เช่นกัน ก็จะได้ output ของระบบที่แสดงองค์ประกอบทางความถี่ของสัญญาณ input ที่สามารถผ่านระบบออกมาได้ ซึ่งก็คือแสดง characteristic ของระบบนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดของโปรแกรม

1. โปรแกรมคำนวณ Difference Equation ของสมการ Digital Filter คือการหา output ของระบบที่อยู่ในเทอมของ time domain นั้นเอง เช่น impulse response

1.1 สัญญาณ input ของระบบ เรากำหนดค่าให้เป็นลำดับของตัวเลข Real number จำนวน 20 ลำดับ ซึ่งรับค่าเข้ามาทางคีย์บอร์ด ลำดับของสัญญาณ input เหล่านี้เสมือนกับว่าเป็นตัวแทนของสัญญาณดิจิทัลที่ได้จากการสุ่ม (sampling) จากนั้นก็เลือกอันดับของวงจรกรองความถี่ที่คิดคือว่าเป็น order 2 หรือ order 4

1.2 นำสัญญาณ input มาผ่านระบบ คือ ผ่านการคำนวณของสมการ Digital Filter ดัง Flow chart ซึ่งมีค่าสัมประสิทธิ์เป็นตัวกำหนดคุณสมบัติของระบบ จะได้สัญญาณ output $[y(n)]$ ออกมาในเทอมของ time domain เช่นแสดง impulse response ของระบบ

1.3 นำค่า $x(n)$ และ $y(n)$ มาพล็อตกราฟแสดงสัญญาณใน เทอมของ time domain

2. โปรแกรมคำนวณ DFT

นำสัญญาณ $x(n), y(n)$ มาผ่านการคำนวณ DFT ที่แสดงดัง Flow chart แล้วนำ $X(k)$ และ $Y(k)$ ที่สัมพันธ์กับความถี่มาพล็อตกราฟ

3. โปรแกรมคำนวณ Frequency Response (Characteristic)

3.1 จากทรานส์เฟอ์ฟังก์ชันของวงจรกรองความถี่ดิจิทัล

$$H(z) = \frac{\sum_{k=0}^N b_k z^{-k}}{1 + \sum_{k=1}^N a_k z^{-k}}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ N เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แทน $z = e^{j\omega T} = \cos(\omega T) + j\sin(\omega T)$ แล้วปรับ ωT ตั้งแต่ 0 ถึง π (วิเคราะห์ในรูป Frequency คือ 0 ถึง $F_s/2$) ผ่านการคำนวณดัง Flow chart โดยไปนำค่าสัมประสิทธิ์ที่คำนวณได้ซึ่งเก็บไว้ในตารางมาใช้ แล้วนำความสัมพันธ์ของขนาดกับ ωT (ความถี่) ไปพล็อตกราฟ

4. โปรแกรมคำนวณหาค่าสัมประสิทธิ์

4.1 เลือกความต้องการฟิลเตอร์ชนิดไหน LPF , HPF

เลือกความต้องการฟิลเตอร์อันดับ 2 , 4

เลือกความต้องการ Butterworth filter หรือ Chebyshev filter

4.2 จากสมการทรานส์เฟอร์ฟังก์ชันของวงจรรวมที่นอกประเภท Chebyshev หรือ Butterworth จะมีค่าสัมประสิทธิ์ที่ต่างกัน จะนำไปสู่การคำนวณหาทรานส์เฟอร์ฟังก์ชันของวงจรรวมความถี่ดิจิทัล นั่นคือเพื่อหาค่าสัมประสิทธิ์ที่แตกต่างกันตามประเภทวงจรรวมความถี่ดิจิทัล ดังแสดงวิธีการทำใน flow chart ของ Transfer function

4.3 นำค่าสัมประสิทธิ์ที่คำนวณได้ไปเก็บไว้ใน File ชื่อ coeff.c หรือ เก็บไว้ในตาราง

5. โปรแกรมการพล็อตกราฟ

5.1 กราฟแสดง Frequency Response

- set driver, mode เขียนแทน x,y ด้วยอักษร (text) บนแกน x,y

- นำค่าขนาดที่ ωT ต่าง ๆ จากการคำนวณมาพล็อตกราฟ line to เป็นส่วนกราฟที่ลากต่อเนื่องกันจากจุดถึงจุด

5.2 กราฟแสดง สัญญาณ input , output ในเทอม time domain

กราฟแสดง สัญญาณ input , output ในเทอม freq domain (DFT)

นำค่า input , output ที่เก็บใน File ในรูปของ Table มาพล็อตกราฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหาค่าสัมประสิทธิ์ของดิจิทัลฟิลเตอร์

1. ฟิลเตอร์อันดับที่ 2

ทรานสเฟอร์ฟังก์ชันของอนาลอกฟิลเตอร์อันดับที่สอง คือ

$$H(s) = \frac{K}{s^2 + As + B} \quad \text{-----(6.1)}$$

ทรานสเฟอร์ฟังก์ชันของดิจิทัลฟิลเตอร์อันดับที่สอง คือ

$$H(z) = \frac{b_0 + b_1z^{-1} + b_2z^{-2}}{1 + a_1z^{-1} + a_2z^{-2}} \quad \text{-----(6.2)}$$

Butterworth prototype polynomial อันดับที่สอง คือ $A = \sqrt{2}$, $B = 1$

Chebyshev prototype polynomial อันดับที่สองที่มี ripple amplitude = 1 dB

คือ $A = 1.097$, $B = 1.1$

จาก Bilinear transform $s = \frac{2.1-z^{-1}}{1+z^{-1}}$

ความสัมพันธ์ระหว่างความถี่อนาลอกและความถี่ดิจิทัลเป็นดังนี้ คือ

$$\omega_a = \frac{2 \tan(\omega_d T/2)}{T}$$

เมื่อ $T = T_s/2 = 1/2F_s$

1.1 กรณี Low Pass Filter แทน $s = s/\omega_a$ ลงในสมการ (6.1)

แล้วจัดพจน์ให้อยู่ในรูปของทรานสเฟอร์ฟังก์ชันของดิจิทัลฟิลเตอร์ ดังสมการ (6.2) จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ได้รับความคุ้มครองออกมาดังนี้
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$b_0 = \frac{K}{\left[\frac{(4^2)Fs^2}{w_a^2} + \frac{(2^2)AFs + B}{w_a^2} \right]}$$

$$b_1 = 2b_0$$

$$b_2 = b_0$$

$$a_1 = \left[\frac{-32Fs^2 + 2B}{w_a^2} \right] \frac{b_0}{K}$$

$$a_2 = \left[\frac{(4^2)Fs^2 - (2^2)AFs + B}{w_a^2} \right] \frac{b_0}{K}$$

1.2 กรณี High Pass Filter แทน $s = w_a/s$ ลงในสมการ (6.1)

แล้วจัดฟอร์มาให้อยู่ในรูปของทรานสเฟอร์ฟังก์ชันของดีจิตอลฟิลเตอร์ ดังสมการที่ (6.2)

จะได้ความสัมพันธ์ออกมาดังนี้

$$b_0 = \frac{K}{\left[\frac{w_a^2 + Aw_a + B}{(4^2)Fs^2} + \frac{Aw_a + B}{(2^2)Fs} \right]}$$

$$b_1 = -2b_0$$

$$b_2 = b_0$$

$$a_1 = \left[\frac{w_a^2 - 2B}{8Fs^2} \right] \frac{b_0}{K}$$

$$a_2 = \left[\frac{w_a^2 - Aw_a + B}{(4^2)Fs^2} - \frac{Aw_a + B}{(2^2)Fs} \right] \frac{b_0}{K}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้... ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า...
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ฟิลเตอร์อันดับที่ 4

ทรานสเฟอ์ฟังก์ชันของอนาลอกฟิลเตอร์อันดับที่ 4

$$H(s) = \frac{K}{s^4 + As^3 + Bs^2 + Cs + D} \quad \text{-----(6.3)}$$

ทรานสเฟอ์ฟังก์ชันของดิจิตอลฟิลเตอร์อันดับที่ 4

$$H(z) = \frac{b_0 + b_1z^{-1} + b_2z^{-2} + b_3z^{-3} + b_4z^{-4}}{1 + a_1z^{-1} + a_2z^{-2} + a_3z^{-3} + a_4z^{-4}} \quad \text{-----(6.5)}$$

Butterworth prototype polynomial อันดับที่ 4 คือ

$$A = 2.616, B = 3.414, C = 2.613, D = 1.0$$

Chebyshev prototype polynomial อันดับที่ 4 ที่มี ripple amplitude = 1 dB

คือ $A = 0.952, B = 1.453, C = 0.742, D = 0.275$

กรณี Low Pass Filter ก็แทน $s = s/\omega_a$ และในกรณี High Pass Filter เราก็แทน $s = \omega_a/s$ เช่นเดียวกันลงในสมการ (6.3) แล้วทำการจัดฟอร์มาให้ อยู่ในรูปของทรานสเฟอ์ฟังก์ชันของดิจิตอลฟิลเตอร์ดังสมการ (6.4) ก็จะหาค่าสัมประสิทธิ์ ออกมาได้เช่นกัน

```

/*****
/*calculate coefficient of digital filter*/
*****/

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

struct {
    float temp[8];
}Temp;

order()
{ int fc,K,type,n,order,N,k;
  float Fs,WA,fca,A,B,C,D,x;
  float a[5] ={0};
  float b[5] ={0};
  FILE *fp;

  clrscr();
  printf("select 1 for LPF ,2 for HPF.");
  scanf("%d",&n);
  printf("select order 2 or 4\n");
  scanf ("%d",&order);
  if(order==2)
  {
    printf (" select type of butterworth or chebyshev filter \n");
    printf (" select number 3 or 4 when 3 is butterworth ,4 is chebyshev.\n");
    scanf (" %d",&type);
    if(type==3)/*coefficient of butterworth analog filter order 2*/
    {
      A = 1.414;
      B = 1;
      K = 1;
    }
    else /*coefficient of Chebyshev analog filter order 2*/
    { A = 1.097;
      B = 1.10;
      K = 1;
    }

    printf (" Enter fc = " ); /*frequency cut off*/
    scanf ( " %d",&fc );
    printf (" Enter Fs = " ); /*sampling frequency*/
    scanf ( " %f",&Fs );
    printf(" fc=%d Fs=%f\n",fc,Fs);
    WA = 2*(2)*Fs*tan((2*M_PI*fc)/(2*2*Fs));
    printf (" WA = %f \n",WA);
    fca = WA/(2*M_PI); /*hertz*/
    printf (" fca = %f \n",fca);

    if(n==1)/*coefficient of LP digital filter order 2*/
    {
      b[0] = K /((4*4*Fs*Fs)/(WA*WA) + (2*2*A*Fs)/WA + B ) ;
      b[1] = 2*b[0];
      b[2] = b[0] ;
      b[3] = 0;
      b[4] = 0;
      a[0] = 0;
      a[1] =((-8*(4)*Fs*Fs)/(WA*WA) + (2*B))*b[0]/K ;
      a[2] =((4*(4)*Fs*Fs)/(WA*WA) - ((2*(2)*A*Fs)/WA)+B )*b[0]/K ;
      a[3] = 0;
      a[4] = 0;
    }
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else /*coefficient of HP digital filter order 2*/
{
    b[0] = K/((WA*WA)/(4*4*Fs*Fs) + (A*WA)/(2*2*Fs) + B );
    b[1] = -2*b[0];
    b[2] = b[0];
    b[3] = 0;
    b[4] = 0;
    a[0] = 0;
    a[1] = ((WA*WA)/(2*4*Fs*Fs)-(2*B))*b[0]/K;
    a[2] = ((WA*WA)/(4*4*Fs*Fs) - (A*WA)/(2*2*Fs) + B)*b[0]/K;
    a[3] = 0;
    a[4] = 0;
}

printf(" A=%f B=%f \n",A,B);
printf(" n=%d\n",n);
printf (" b[0] = %e \n",b[0]);
printf (" b[1] = %e \n",b[1]);
printf (" b[2] = %e \n",b[2]);
printf (" a[0] = %e \n",a[0]);
printf (" a[1] = %e \n",a[1]);
printf (" a[2] = %e \n",a[2]);
}

else/*order 4*/
{
    printf (" select type of butterworth or chebyshev filter \n");
    printf (" select number 3 or 4 when 3 is butterworth ,4 is chebyshev.\n");
    scanf (" %d",&type);
    if(type==3)/*coefficient of Butterworth analog filter order 4*/
    {
        A = 2.613;
        B = 3.414;
        C = 2.613;
        D = 1;
        K = 1;
    }
    else /*coefficient of Chebyshev analog filter order 4*/
    {
        A = 0.952;
        B = 1.453;
        C = 0.742;
        D = 0.275;
        K = 1;
    }

    printf (" Enter fc = " ) ; /*frequency cut off*/
    scanf ( " %d",&fc );
    printf (" Enter Fs = " ) ; /*sampling frequency*/
    scanf ( " %f",&Fs );
    printf(" fc=%d Fs=%f\n",fc,Fs);
    WA = 2*(2)*Fs*tan((2*M_PI*fc)/(2*2*Fs));
    printf (" WA = %f \n",WA);
    fca = WA/(2*M_PI); /*hertz*/
    printf (" fca = %f \n",fca);

    if (n==1)/*coefficient of LP digital filter order 4*/
    {
        b[0] = K/((16*16*Fs*Fs*Fs*Fs)/(WA*WA*WA*WA)+(A*8*8*Fs*Fs*Fs)/(WA*WA*WA)
        +(B*4*4*Fs*Fs)/(WA*WA)+((C*2*2*Fs)/WA)+D);
        b[1] = 4*b[0];
        b[2] = 6*b[0];
        b[3] = 4*b[0];
        b[4] = b[0];
        a[0] = 0;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

a[1] = ((-4*16*16*Fs*Fs*Fs*Fs)/(WA*WA*WA*WA) - (A*16*8*Fs*Fs*Fs)/(WA*WA*WA)
+ ((C*4*2*Fs)/WA) + (D*4))*b[0]/K;
a[2] = ((6*16*16*Fs*Fs*Fs*Fs)/(WA*WA*WA*WA) - (B*8*4*Fs*Fs)/(WA*WA)
+ (D*6))*b[0]/K;
a[3] = ((-4*16*16*Fs*Fs*Fs*Fs)/(WA*WA*WA*WA) + (A*16*8*Fs*Fs*Fs)/(WA*WA*WA)
- ((C*4*2*Fs)/WA) + (D*4))*b[0]/K;
a[4] = ((16*16*Fs*Fs*Fs*Fs)/(WA*WA*WA*WA) - (A*8*8*Fs*Fs*Fs)/(WA*WA*WA)
+ (B*4*4*Fs*Fs)/(WA*WA) - ((C*2*2*Fs)/WA) + D)*b[0]/K;
}
else/*coefficient of HP digital filter order 4*/
{
b[0] = K/((WA*WA*WA*WA)/(16*16*Fs*Fs*Fs*Fs) + (A*WA*WA*WA)/(8*8*Fs*Fs*Fs)
+ (B*WA*WA)/(4*4*Fs*Fs) + (C*WA)/(2*2*Fs) + D);
b[1] = -4*b[0];
b[2] = 6*b[0];
b[3] = -4*b[0];
b[4] = b[0];
a[0] = 0;
a[1] = ((WA*WA*WA*WA)/(4*16*Fs*Fs*Fs*Fs) + (A*WA*WA*WA)/(8*4*Fs*Fs*Fs)
- (C*WA)/(2*Fs) - (D*4))*b[0]/K;
a[2] = ((6*WA*WA*WA*WA)/(16*16*Fs*Fs*Fs*Fs) - (B*WA*WA)/(2*4*Fs*Fs)
+ (D*6))*b[0]/K;
a[3] = ((WA*WA*WA*WA)/(4*16*Fs*Fs*Fs*Fs) - (A*WA*WA*WA)/(4*8*Fs*Fs*Fs)
+ (C*WA)/(2*Fs) - (D*4))*b[0]/K;
a[4] = ((WA*WA*WA*WA)/(16*16*Fs*Fs*Fs*Fs) - (A*WA*WA*WA)/(8*8*Fs*Fs*Fs)
+ (B*WA*WA)/(4*4*Fs*Fs) - (C*WA)/(2*2*Fs) + D)*b[0]/K;
}
printf ("A=%f B=%f C=%f D=%f\n ", A, B, C, D);
printf ("n=%d \n", n);
printf ("b[0]=%e \n", b[0]);
printf ("b[1]=%e \n", b[1]);
printf ("b[2]=%e \n", b[2]);
printf ("b[3]=%e \n", b[3]);
printf ("b[4]=%e \n", b[4]);
printf ("a[0]=%e \n", a[0]);
printf ("a[1]=%e \n", a[1]);
printf ("a[2]=%e \n", a[2]);
printf ("a[3]=%e \n", a[3]);
printf ("a[4]=%e \n", a[4]);
}
x=(b[0]+b[1]+b[2]+b[4]+b[3])/(1+a[1]+a[2]+a[3]+a[4]);
printf ( " x=%f \n", x);
fp = fopen("coeff.ab", "w");/*get coefficient of digital filter in file*/
for (k=0; k<= 4; k++) {
fprintf (fp, "a[%d]=%f b[%d]=%f \n", k, a[k], k, b[k]);
Temp.temp[2*k] = a[k];
Temp.temp[(2*k)+1] = b[k];
}
fclose(fp);

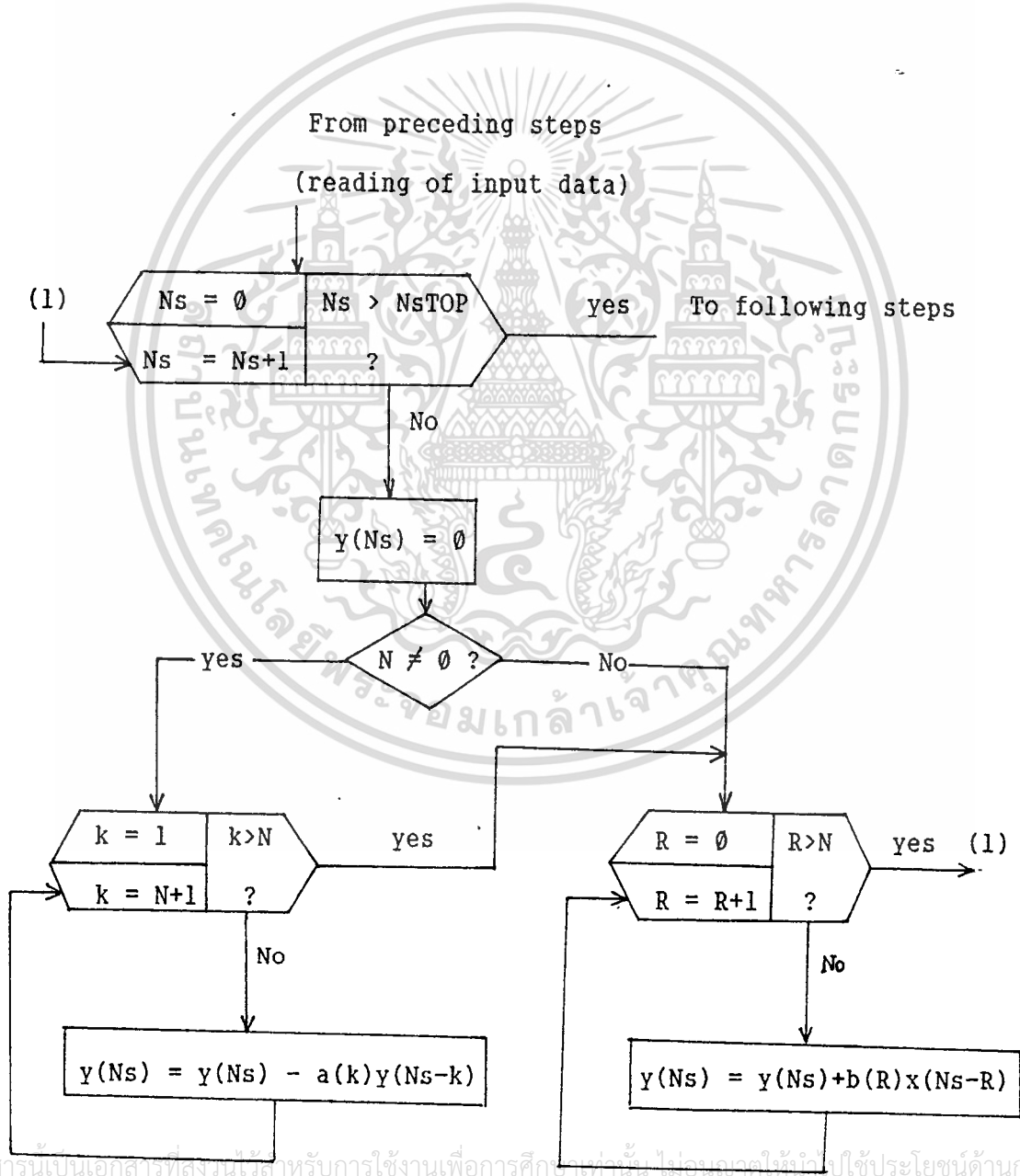
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมการ Digital Filter

$$y(n) = \sum_{k=0}^R b_k x(n-k) - \sum_{k=1}^N a_k y(n-k) \quad \text{Difference Equation}$$

$$y(N_s) = \sum_{R=0}^R b_k x(N_s-R) - \sum_{k=1}^N a_k y(N_s-k)$$



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้แก้ไข ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมการ DFT

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi/N}(nk) \quad , k = 0,1,2, \dots, N-1$$

$$Y(k) = \sum_{n=0}^{N-1} y(n)e^{-j2\pi/N}(nk) \quad , k = 0,1,2, \dots, N-1$$

$$X(k) = \sum_{n=0}^{N-1} \left\{ [x(n)\cos(2\pi/N)(nk)] + [x(n)\sin(2\pi/N)(nk)] \right\}$$

$$Y(k) = \sum_{n=0}^{N-1} \left\{ [y(n)\cos(2\pi/N)(nk)] + [y(n)\sin(2\pi/N)(nk)] \right\}$$

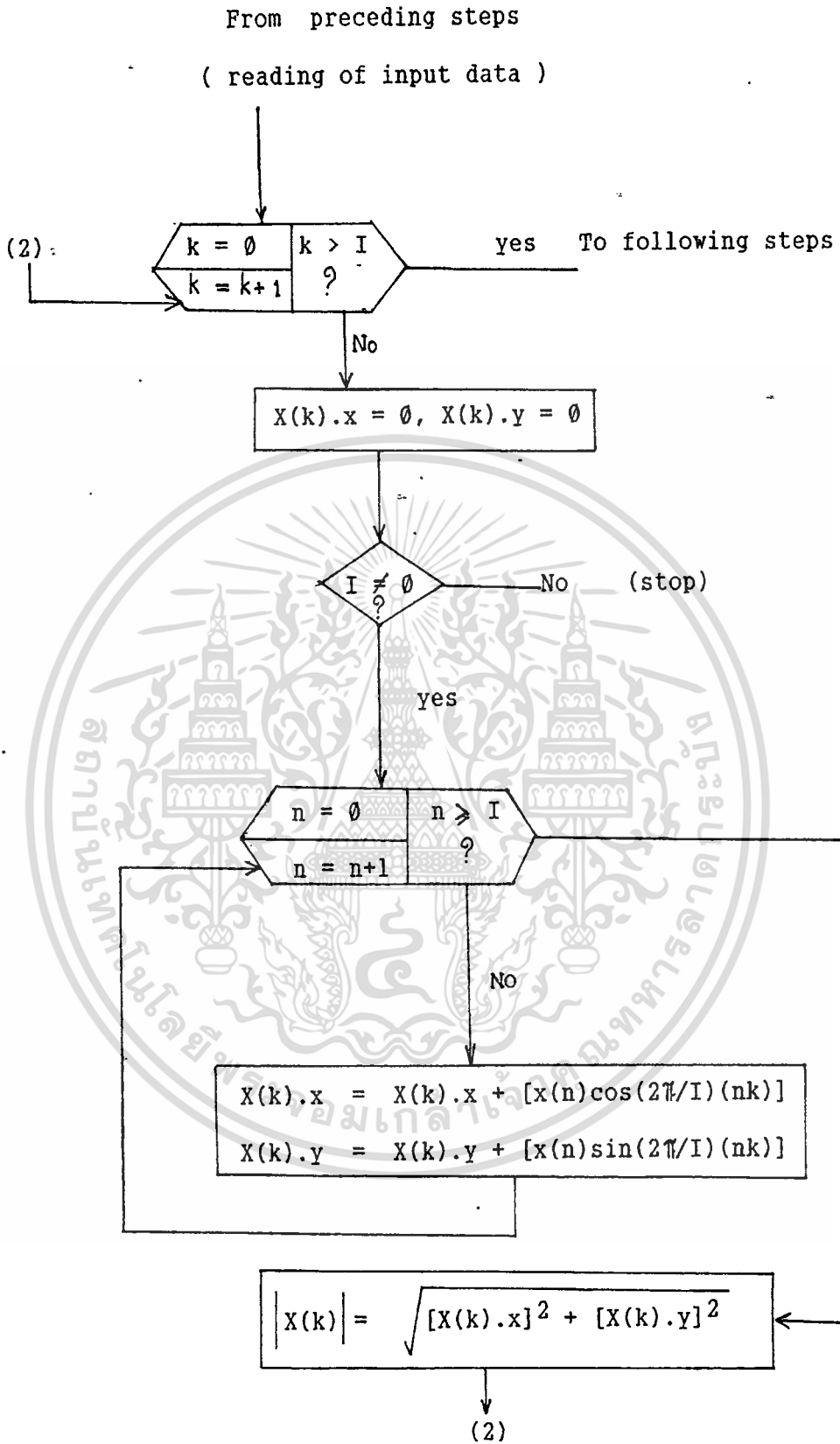
$N = I = N_{\text{STOP}}$ (จำนวนลำดับของ input ที่เข้ามา)

$$X(k) = \sum_{n=0}^{I-1} \left\{ [x(n)\cos(2\pi/I)(nk)] + [y(n)\sin(2\pi/I)(nk)] \right\}$$

$$Y(k) = \sum_{n=0}^{I-1} \left\{ [y(n)\cos(2\pi/I)(nk)] + [y(n)\sin(2\pi/I)(nk)] \right\}$$

$$k = 0,1,2 \dots, I-1$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

K=1;

if ((Ns-K)<0) {
    K++;
    while (K<=N)
    {
        K++;
    }
}
else if((Ns-K)>=0) {
    y[Ns] = y[Ns] - a[K]*y[(Ns-K)];
    K++;

    while(K<=N)
    {
        if((Ns-K)>=0)
            y[Ns] = y[Ns] - a[K]*y[(Ns-K)];
        K++;
    }
}

R= 0 ;
while(R<=N)
{
    if((Ns-R)>=0){
        y[Ns] = y[Ns] + b[R]*x[(Ns-R)] ;
        R++;
    }
    else
        R++;
    printf("\n y[%d]= %f ",Ns,y[Ns]);
}

fp=fopen("table2.tab","w"); /*get y[Ns] in file*/
for(m=0;m<=20;m++)
    fprintf(fp,"\n %f",y[m]);
fclose(fp);

I=NSTOP;
printf ("\n\n\n\t input value =%d",I);

for (K = 0; K <= I; K++) /*find DFT Of input */
{
    for (n = 0; n <= I; n++)
    {
        z[K].x += x[n] * cos ((2*M_PI*K*n)/I);
        z[K].y += x[n] * sin ((2*M_PI*K*n)/I);
    }
}

for (K = 0; K <= I; K++)
{
    X[K] = cabs (z[K]);
    printf (" \n X[%d] = %f " ,K ,X[K] );
}

fp=fopen("table3.tab","w"); /*get DFT of input in file*/
for(m=0;m<=20;m++)
    fprintf(fp,"\n %f",X[m]);
fclose(fp);

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for (K = 0; K <= I; K++) /*find DFT of y[Ns] */
{
    for (n = 0; n <= I; n++)
    {
        w[K].x += y[n] * cos ((2*M_PI*K*n)/I);
        w[K].y += y[n] * sin ((2*M_PI*K*n)/I);
    }
}

for (K = 0; K <= I; K++)
{
    Y[K] = cabs (w[K]);
    printf (" \nX[%d] = %f Y[%d] = %f " ,K,X[K],K ,Y[K] );
}

fp=fopen("table4.tab","w"); /*get DFT of y[Ns] in file*/
for(m=0;m<=20;m++)
fprintf(fp,"\n %f",Y[m]);
fclose(fp);
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Frequency Response, (Characteristic)

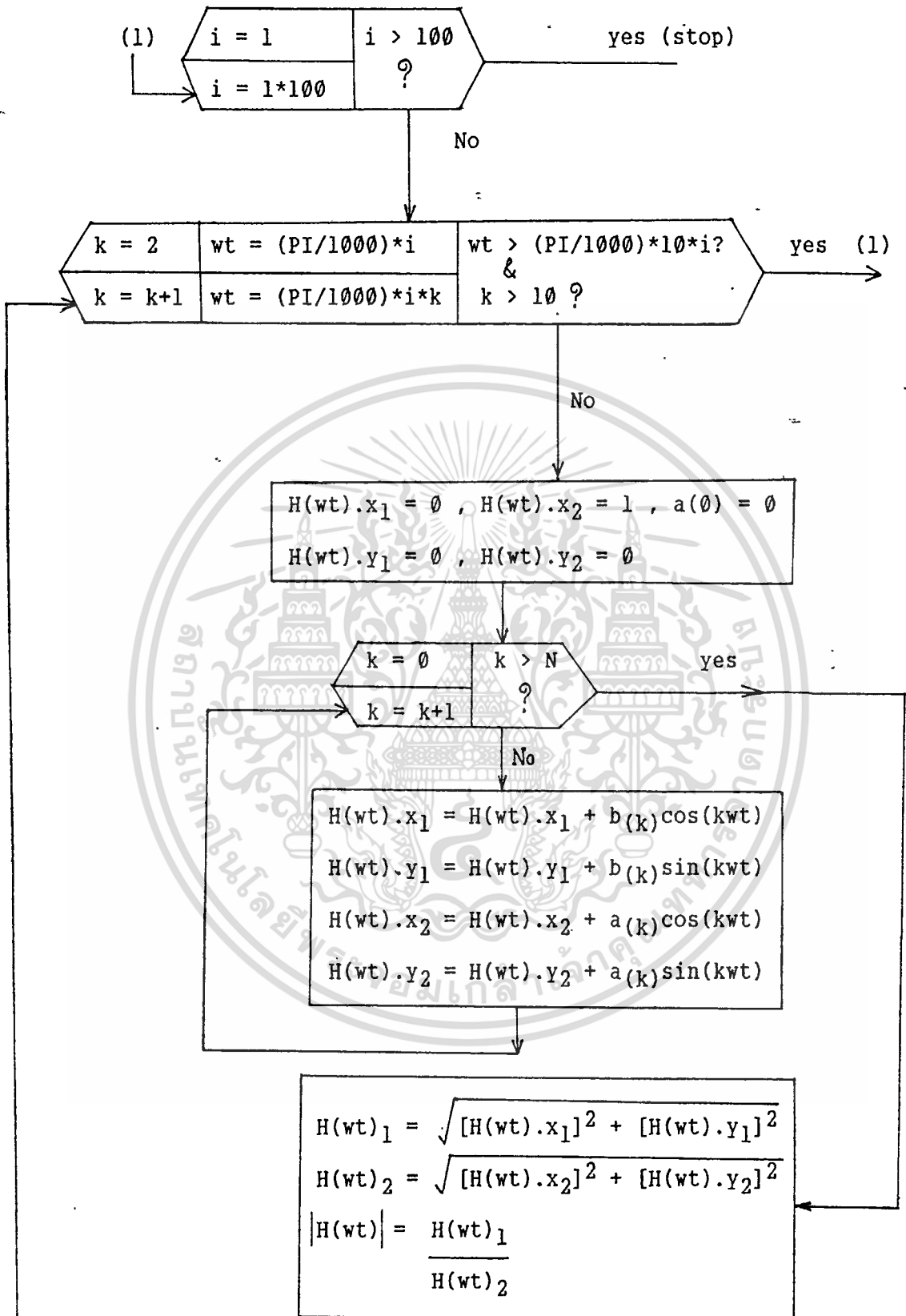
$$H(s) = \frac{\sum_{k=0}^N b_k z^{-k}}{1 + \sum_{k=1}^N a_k z^{-k}}$$

$$z = e^{j\omega t} = \cos(\omega t) + j\sin(\omega t)$$

$$z^{-k} = e^{-jk\omega t} = \cos(k\omega t) - j\sin(k\omega t)$$

$$H(\omega t) = \frac{\sum_{k=0}^N b_k [\cos(k\omega t) + \sin(k\omega t)]}{1 + \sum_{k=1}^N a_k [\cos(k\omega t) + \sin(k\omega t)]}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }/*****/
/* ploat graph of frequency response*/
/*****/

#include <graphics.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <order_2.c>

#define SpaceX 50
#define _X 1.0
#define _Y 1.0
#define MaxY 12
/*#define STEP (float) 0.05*/

#define Z(x) (float) (cos (x) + sin (x))
#define PI 3.14

typedef enum{FALSE,TRUE} STATUS;
STATUS GetFile();
float a[5];
float b[5];

void main ()
{
float Sx,Sy;
float wT;
float AHZ, Temp,STEP,P;
float Re, Im,n,m,p;
char Scale[6],I;
int drv, mode,i,j,N,L,k,s,h;
int ref;
struct complex zn;

clrscr();
order();/*for using coefficient of digital filter*/
clrscr();
GET();
detectgraph (&drv, &mode);
initgraph (&drv, &mode, "");
line(SpaceX,0,SpaceX,Sy=getmaxy()*_Y);
ref=Sy/2;
Sy/=MaxY*2;
Sx=getmaxx()*_X;

line(SpaceX,ref,Sx,ref);
Sx/=8;
for(i=-5,k=0;i<=1,k<=6;i++,k++){
outtextxy(SpaceX+(k*Sx),ref-2,"I");
n=0.5*i;
sprintf(Scale,"% .2f",n);
outtextxy(SpaceX+(k*Sx),ref+10,Scale);
}
for(s=1,k=0;s<=1000,k<=3;s=s*10,k++){
p=(PI/1000)*s;
h=2*k;
sprintf(Scale,"% .4f",p);
outtextxy(SpaceX+(h*Sx),ref+25,Scale);
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(i=MaxY,j=0;i>=0;i--,j++){
    m=i*0.1;
    sprintf(Scale,"%0.2f",m);
    outtextxy(0,(j*Sy),Scale);
}
moveto(SpaceX,ref);
for(i=1;i<=100;i=i*10){
    for (wT = (PI/1000)*i,k=2; wT <= (PI/1000)*10*i,k<=10; wT = STEP,k++){
        STEP=(PI/1000)*i*k;
/*      Hz = (b[0] + b[1]*Z(wT) + b[2]*Z(2.0*wT) +b[3]*Z(3*wT)+b[4]*Z(4*wT)) /
          (1 + a[1]*Z(wT) + a[2]*Z(2.0*wT)+a[3]*Z(3*wT)+a[4]*Z(4*wT));
*/
zn.x = (b[0] + b[1]*cos(wT) + b[2]*cos(2*wT)+b[3]*cos(3*wT)+b[4]*cos(4*wT)
zn.y = (b[1]*sin(wT) + b[2]*sin(2*wT)+b[3]*sin(3*wT)+b[4]*sin(4*wT));
Temp = cabs (zn);

zn.x = (1 + a[1]*cos(wT) + a[2]*cos(2*wT)+a[3]*cos(3*wT)+a[4]*cos(4*wT));
zn.y = (a[1]*sin(wT) + a[2]*sin(2*wT)+a[3]*sin(3*wT)+a[4]*sin(4*wT));
AHZ = Temp / cabs (zn);
P=log10(wT);
/*      printf("\n %f %f %f",wT,AHZ,P);*/
lineto (SpaceX+2*Sx*log10(wT)+(5*Sx), ref - (AHZ*10*Sy));
delay(100);
getch();
}
getch ();
closegraph ();
}
GET()
{
int i;
STATUS flag = TRUE;

for(i=0;i<=4;i++)
    printf("\n a[%d]=%f b[%d]=%f",i,Temp.temp[2*i],i,Temp.temp[(2*i)+1]);

getch();
for(i=0;i<=4;i++) {
    a[i] = Temp.temp[2*i];
    b[i] = Temp.temp[(2*i)+1];
}

return(flag==TRUE)?TRUE:FALSE;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*****
/*_ploat graph with data from file*/
*****/

#if !__MEDIUM__
    #error To be set Medium model
#endif

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

#define _X 0.6 /* Rate of X axis */
#define _Y 1.0 /* Rate of Y axis */
#define MaxY 30 /* Maximun of data in TABLE */
#define TableSize 40 /* Maximun number of data */
#define SpaceX 50

typedef enum { FALSE, TRUE } STATUS;

typedef struct {
    float Table [TableSize];
    int lenght;
    ! BUFFER;

BUFFER Table = {0}; /* Declare buffer */
STATUS GetFile (char *); /* Get data from TABLE */

void main (int argc, char *argv [])
{
    int drv, mode;
    int ref, i, j,u;
    int P1, P2;
    int Sx, Sy; /* Step's Mutiplier (X-axis & Y-axis) */
    char Scale[6]; /* for convert int ==> chars. */
    struct complex zn;

    if (GetFile (argv [1]) == FALSE)
        exit (0);

    detectgraph (&drv, &mode);
    initgraph (&drv, &mode, "");

    if (graphresult () != grOk)
    {
        printf ("\n Graph error");
        exit (0);
    }

    line (SpaceX, 0, SpaceX, Sy = getmaxy ()*_Y); /* Y axis */
    ref = Sy / 2;
    Sx = i = (getmaxx ()*_X);

    Sx /= Table.lenght;
    Sy /= MaxY*2;

    line (SpaceX,ref,i+Sx+SpaceX,ref); /* X axis */
    for (i = 0: i < Table.lenght; i++){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        outtextxy ((i*Sx)+Sx+SpaceX, ref + 10, itoa (i, &Scale[0], 10));
    }

    for (i =10, j=0; i >= -10; i--, j++)
    {
        u=i*3;
        sprintf (Scale, "%4d", u);
        outtextxy (0, (3*j*Sy)+4*Sy, Scale);
    }

    for ( i = 0; i < Table.lenght ; i++)
    {
        P1 = (i * Sx) + Sx + SpaceX;
        P2 = Table.Table [i] * Sy;
        line (P1,ref,P1,ref-(P2));
        /* bar (P1-2,ref-(P2)+2, P1+2, ref-(P2) - 2); */
        circle (P1,ref-(P2),2);
    }

    getch ();
    closegraph ();
}
/*****
/* Routine: GetFile
/* Funcion: Open file and get data from
/* stream of file
/* return: STATUS (Success, Non-success)
*****/
STATUS GetFile (char file [30])
{
    int i;
    FILE *fp;
    STATUS flag = TRUE;

    if (file == NULL)
    {
        printf ("\n Type File Name (xxx.TAB) = ");
        scanf ("%s", file);
    }

    if (( fp = fopen (file,"rt")) == NULL)
    {
        perror ("\nNow, File Error");
        flag = FALSE;
    }

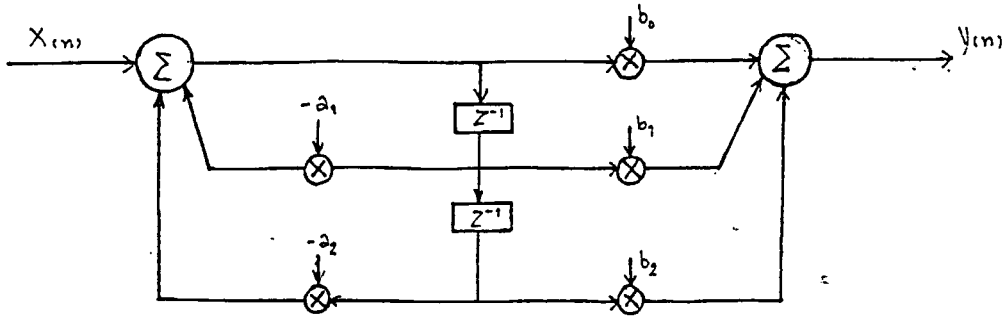
    for ( i = 0; (fscanf (fp,"%f",&Table.Table [i]) != EOF) && (i < TableSize); i++)
    if (i >= (TableSize-3))
    {
        printf ("\n Buffer overflow");
        flag = FALSE;
    }

    Table.lenght = i;
    fclose (fp);

    return (flag == TRUE) ? TRUE : FALSE;
}

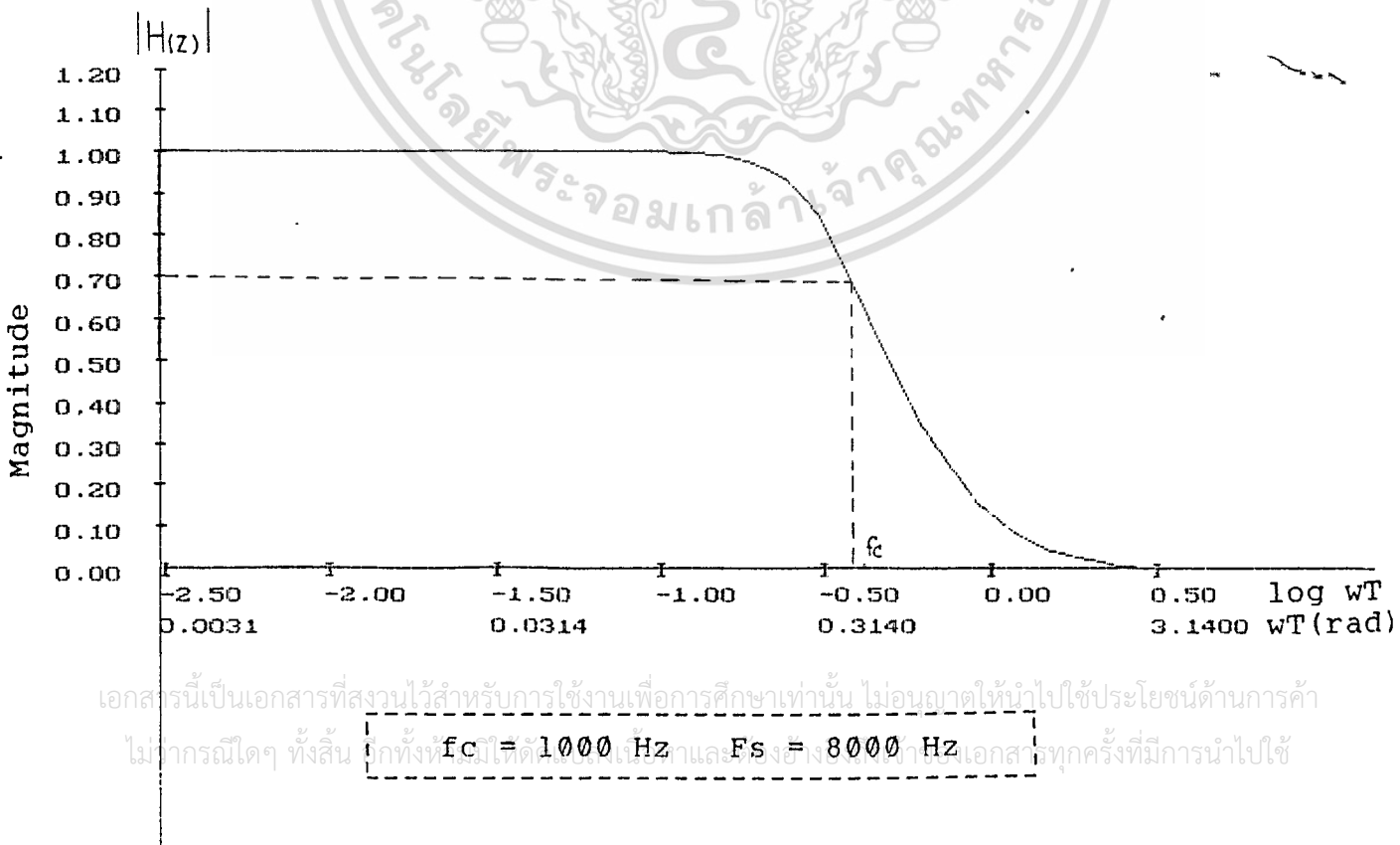
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

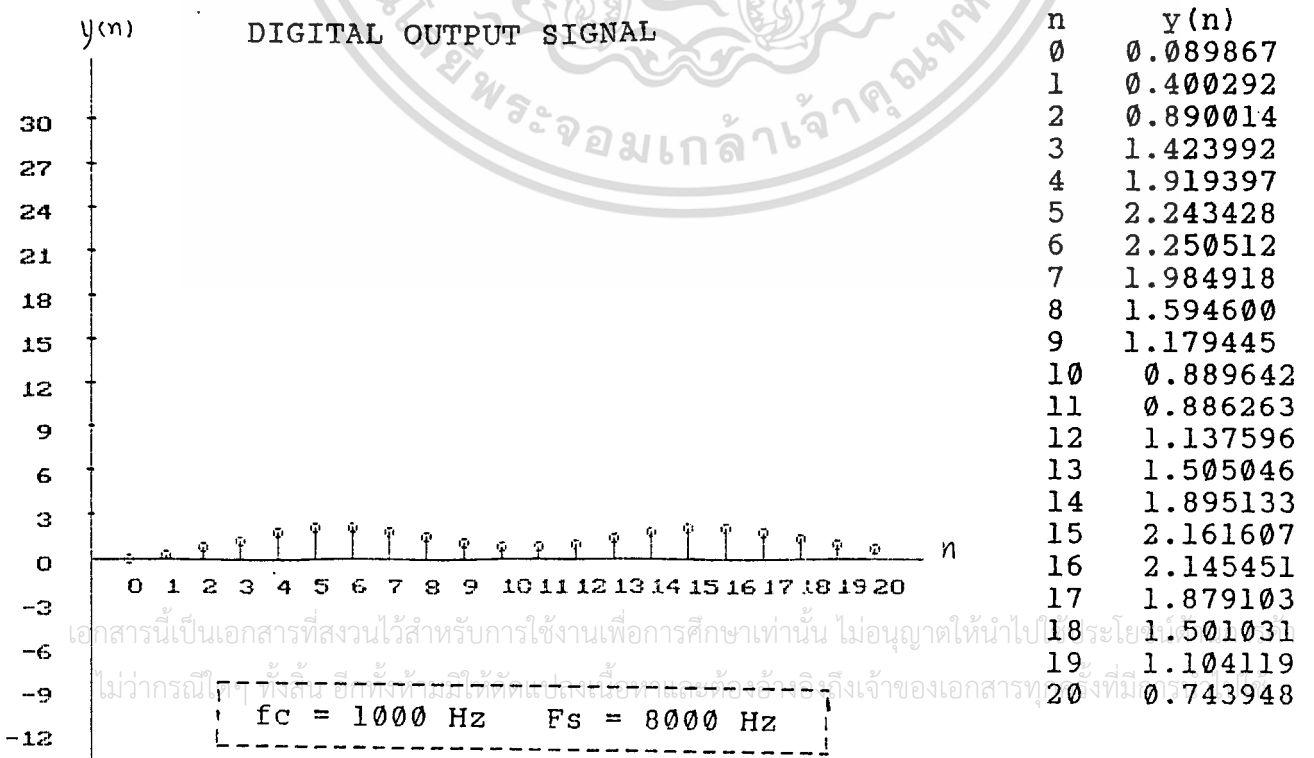
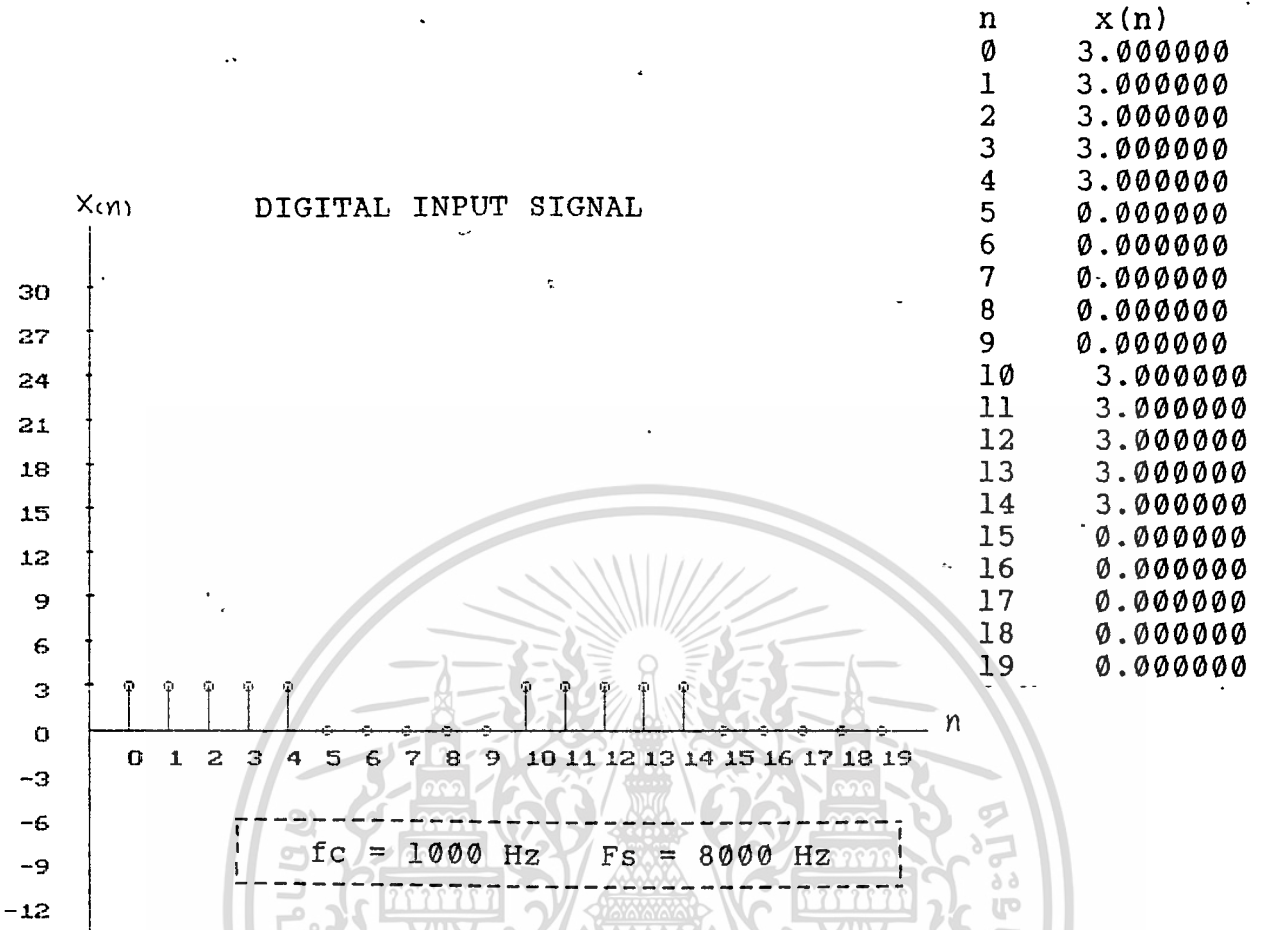


$a[0]=0.000000$ $b[0]=0.029956$
 $a[1]=-1.454290$ $b[1]=0.059911$
 $a[2]=0.574113$ $b[2]=0.029956$

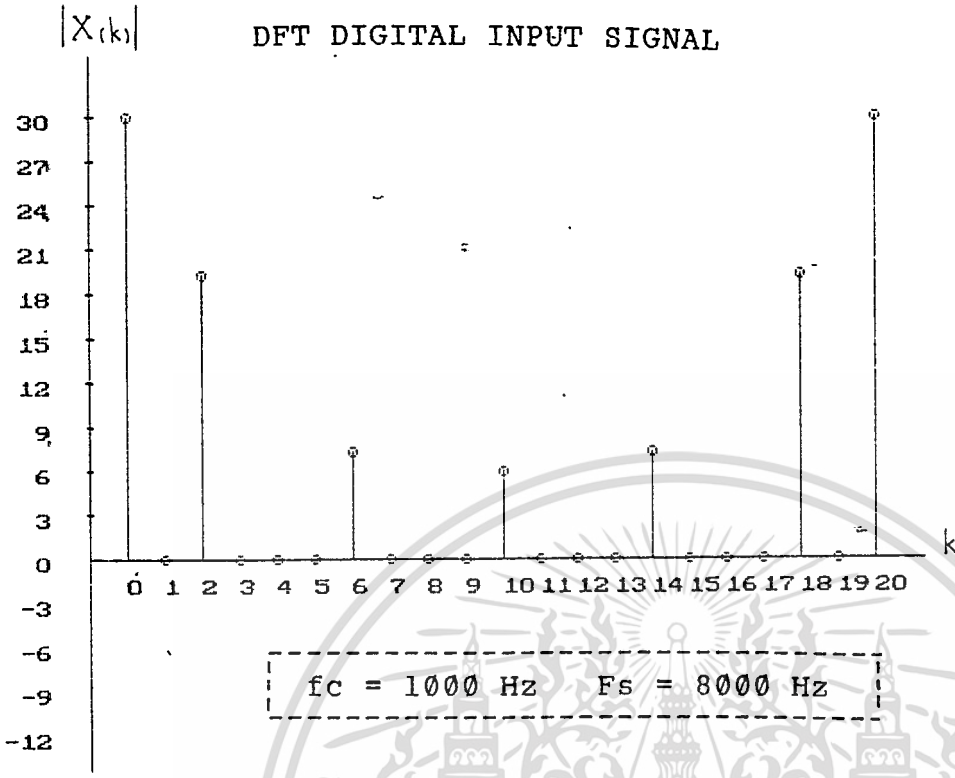
CHARACTERISTIC OF DIGITAL FILTER
 Design by Bilinear Transform
 Butterworth Low Pass Filter
 2-order



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ทำกรณีใดๆ ทั้งสิ้น $f_c = 1000$ Hz $F_s = 8000$ Hz

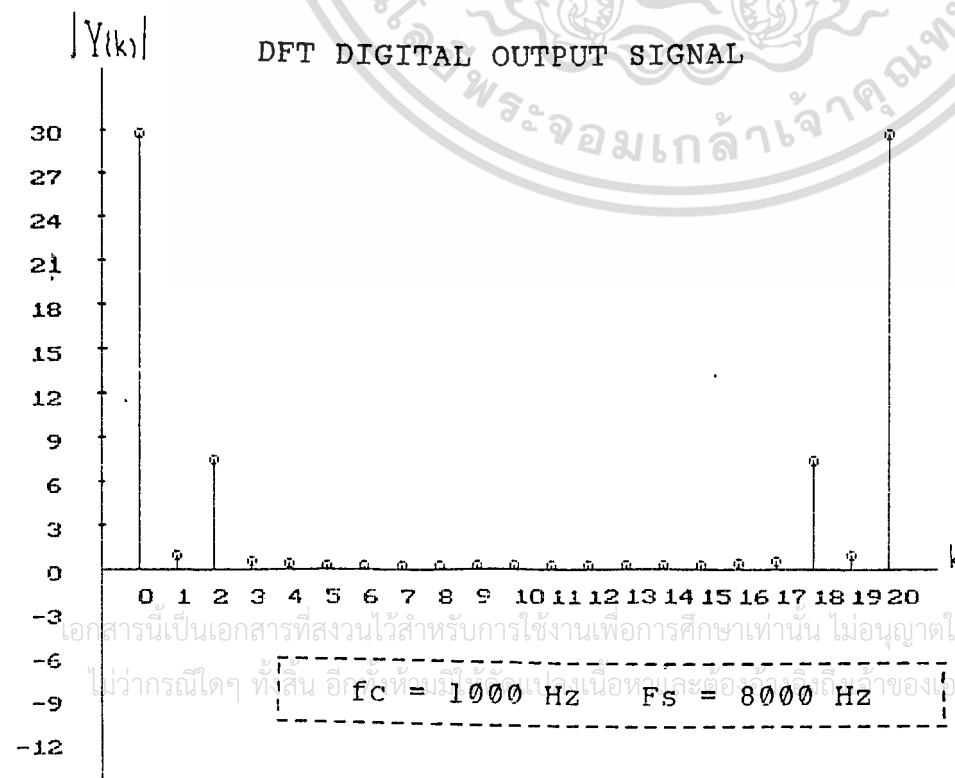


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไป
 18 ระเบียบ
 19
 20



k	X(k)
0	30.000000
1	0.000000
2	19.41640
3	0.000000
4	0.000000
5	0.000000
6	7.416408
7	0.000000
8	0.000000
9	0.000000
10	6.000000
11	0.000000
12	0.000000
13	0.000000
14	7.416408
15	0.000000
16	0.000000
17	0.000000
18	19.41640
19	0.000000
20	30.000000

$f_c = 1000 \text{ Hz}$ $F_s = 8000 \text{ Hz}$

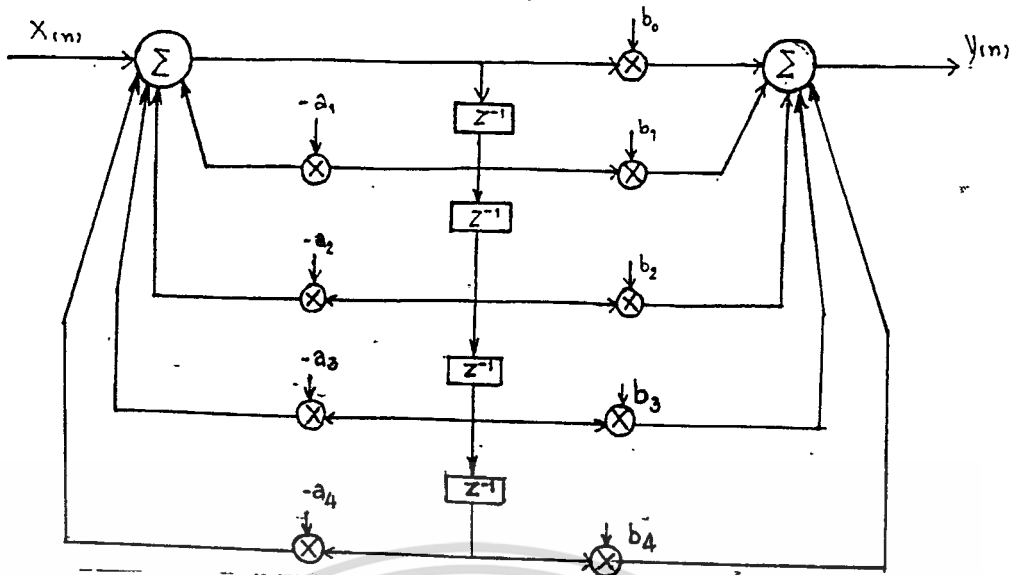


k	Y(k)
0	29.82540
1	1.000482
2	7.506727
3	0.629981
4	0.490538
5	0.408477
6	0.339924
7	0.324334
8	0.303855
9	0.292581
10	0.288977
11	0.292581
12	0.303855
13	0.324334
14	0.339924
15	0.408477
16	0.490538
17	0.629981
18	7.506727
19	1.000482
20	29.82540.

$f_c = 1000 \text{ Hz}$ $F_s = 8000 \text{ Hz}$

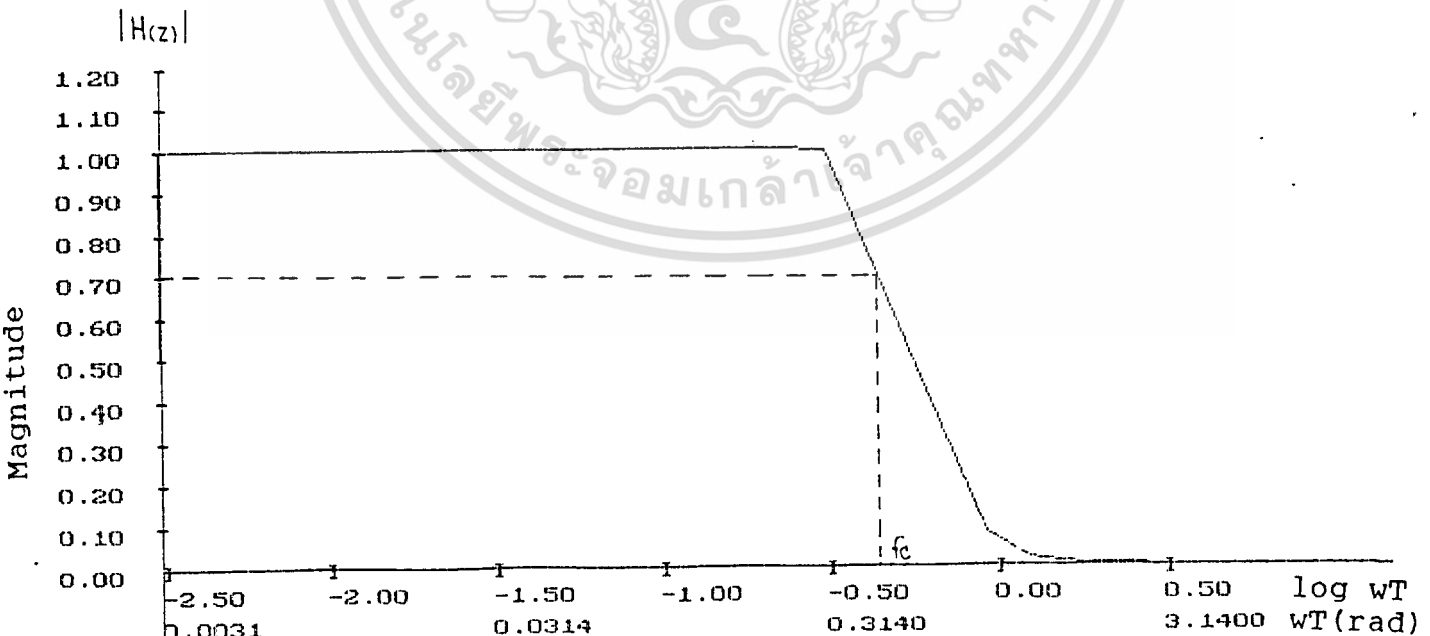
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกส่วนหนึ่งมีลิขสิทธิ์ในเนื้อหาของเอกสารทุกครั้งที่มีการนำไปใช้



$a[0]=0.000000$ $b[0]=0.003141$
 $a[1]=-2.559373$ $b[1]=0.012563$
 $a[2]=2.629454$ $b[2]=0.018845$
 $a[3]=-1.249525$ $b[3]=0.012563$
 $a[4]=0.229697$ $b[4]=0.003141$

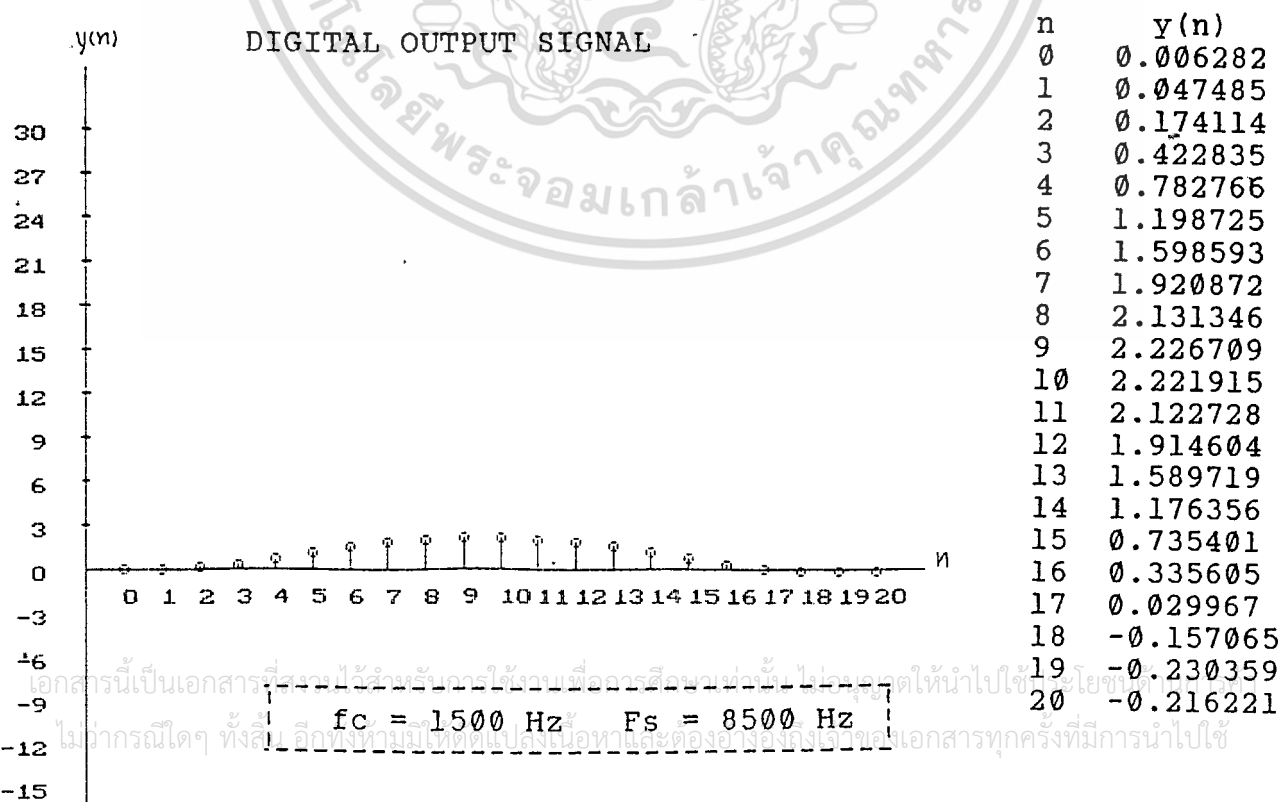
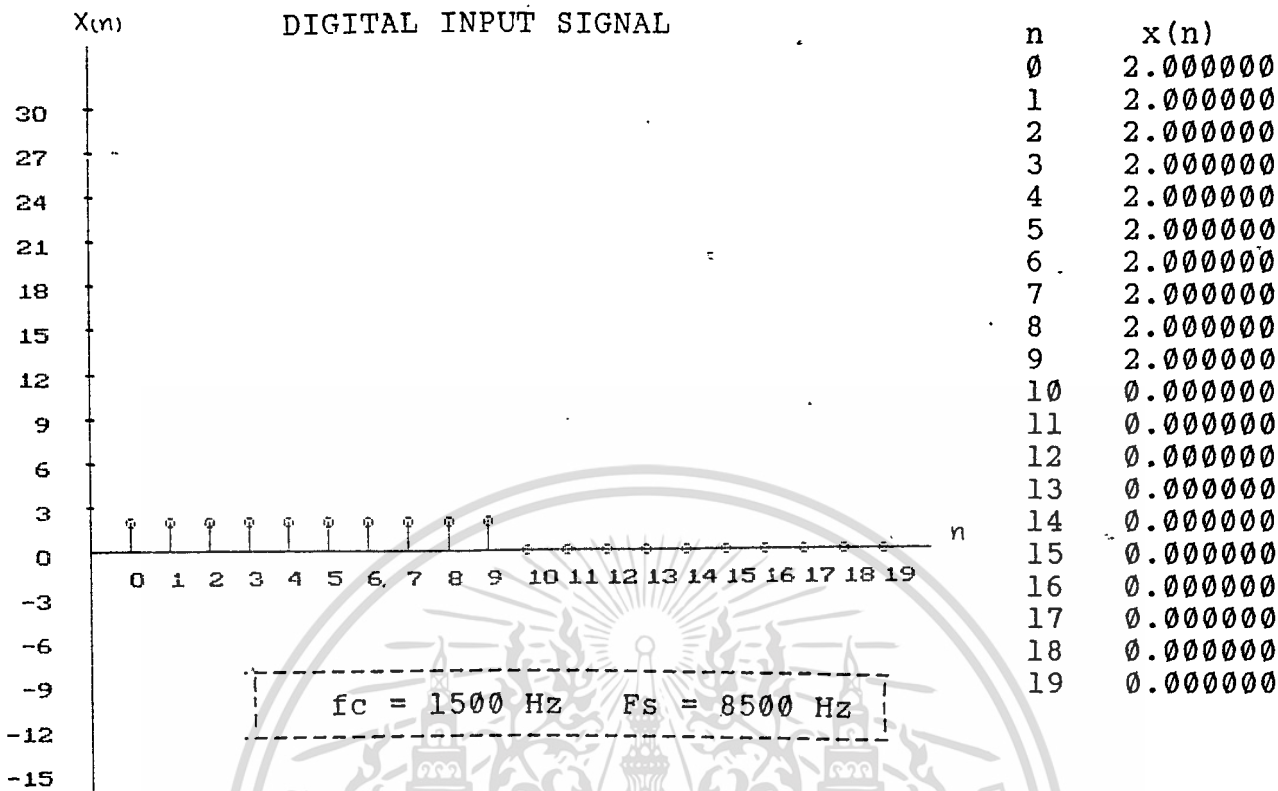
CHARACTERISTIC OF DIGITAL FILTER
 Design by Bilinear Transform
 Butterworth Low Pass Filter
 4-order



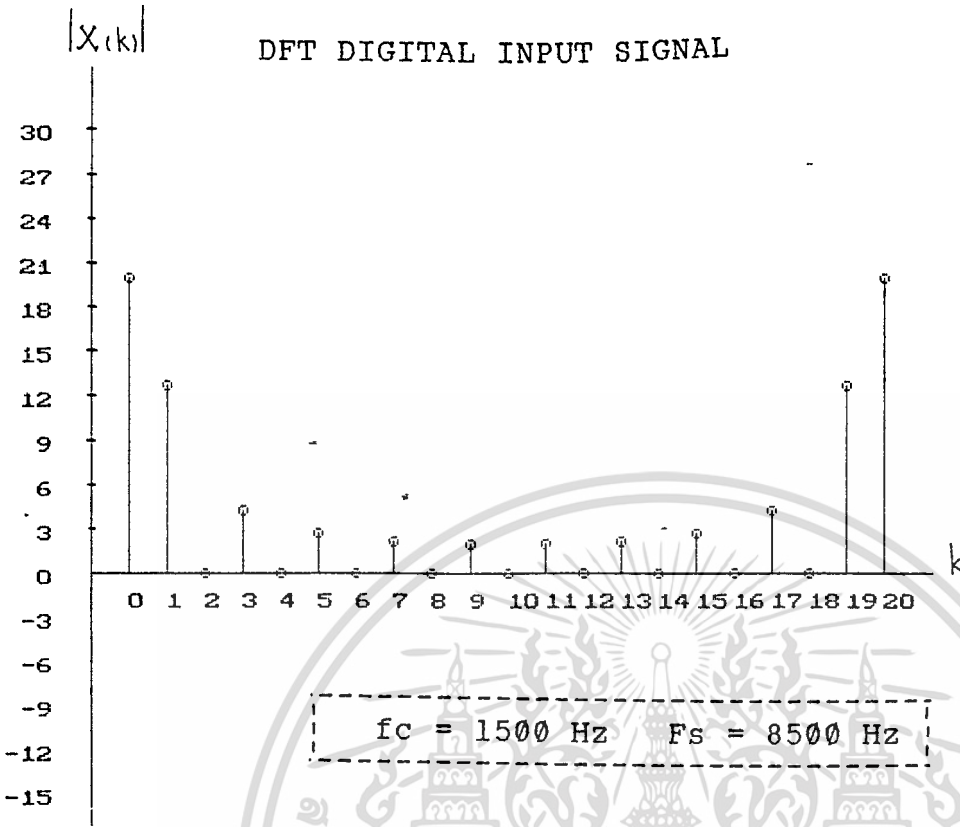
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ทำกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามใช้ข้อมูลไปหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

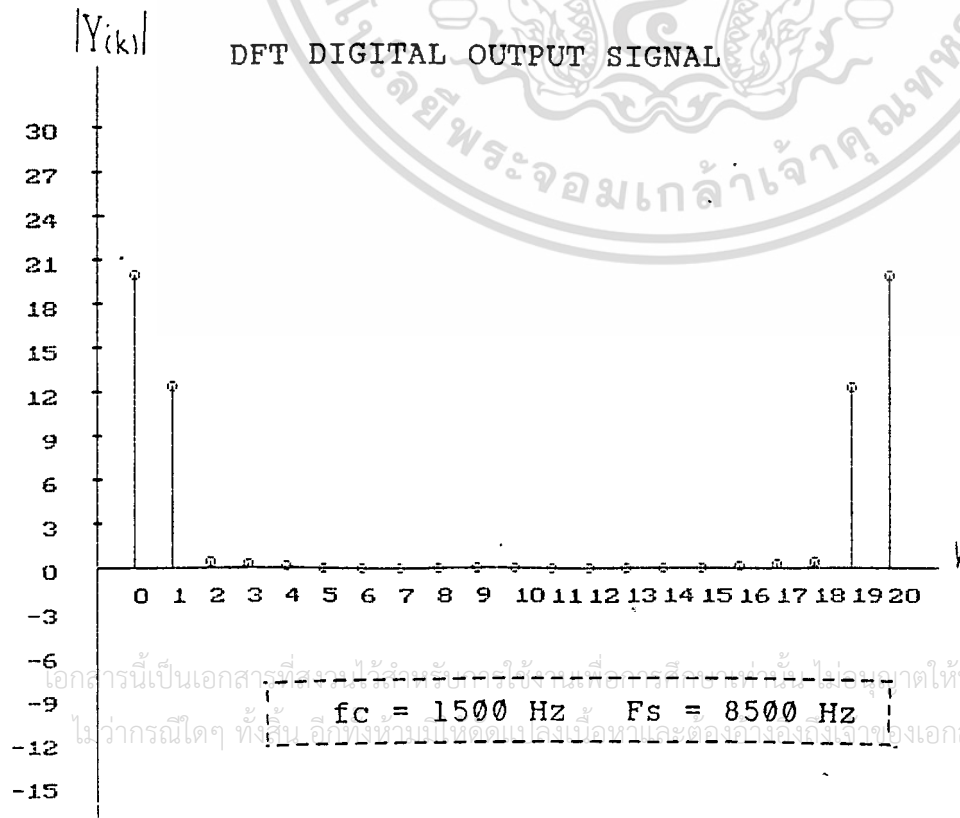
$f_c = 1500 \text{ Hz}$ $F_s = 8500 \text{ Hz}$



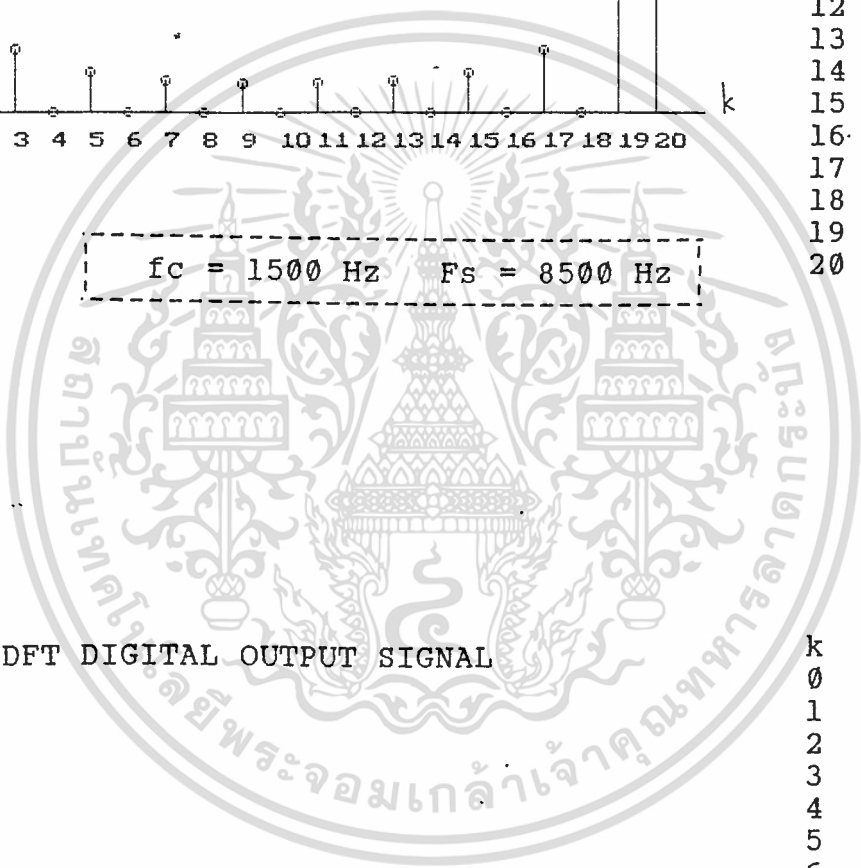
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในวงวิชาการเท่านั้น ไม่ควรนำออกเผยแพร่โดยไม่ได้รับอนุญาตให้ทำซ้ำโดยไม่ได้รับอนุญาต
 ไม่ทำกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกไปเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



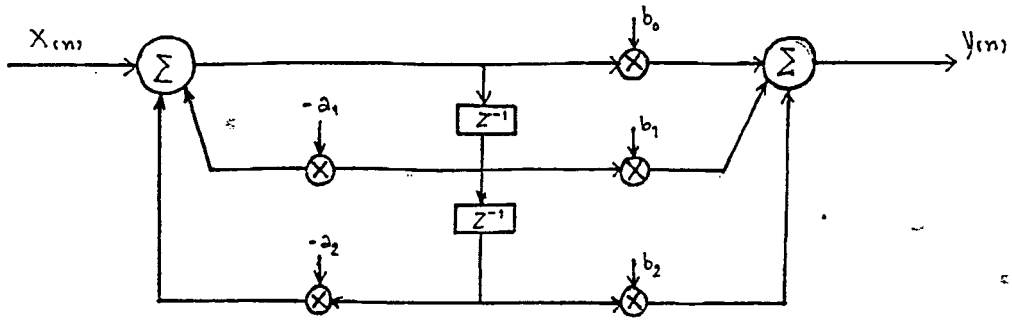
k	X(k)
0	20.000000
1	12.784900
2	0.000000
3	4.405379
4	0.000000
5	2.828427
6	0.000000
7	2.244652
8	0.000000
9	2.024930
10	0.000000
11	2.024930
12	0.000000
13	2.244652
14	0.000000
15	2.828427
16	0.000000
17	4.405379
18	0.000000
19	12.784900
20	20.000000



k	Y(k)
0	20.03237
1	12.52696
2	0.435389
3	0.405301
4	0.181723
5	0.134968
6	0.122115
7	0.108377
8	0.101346
9	0.097119
10	0.095790
11	0.097119
12	0.101346
13	0.108377
14	0.122115
15	0.134968
16	0.181723
17	0.405301
18	0.435389
19	12.52696
20	20.03237

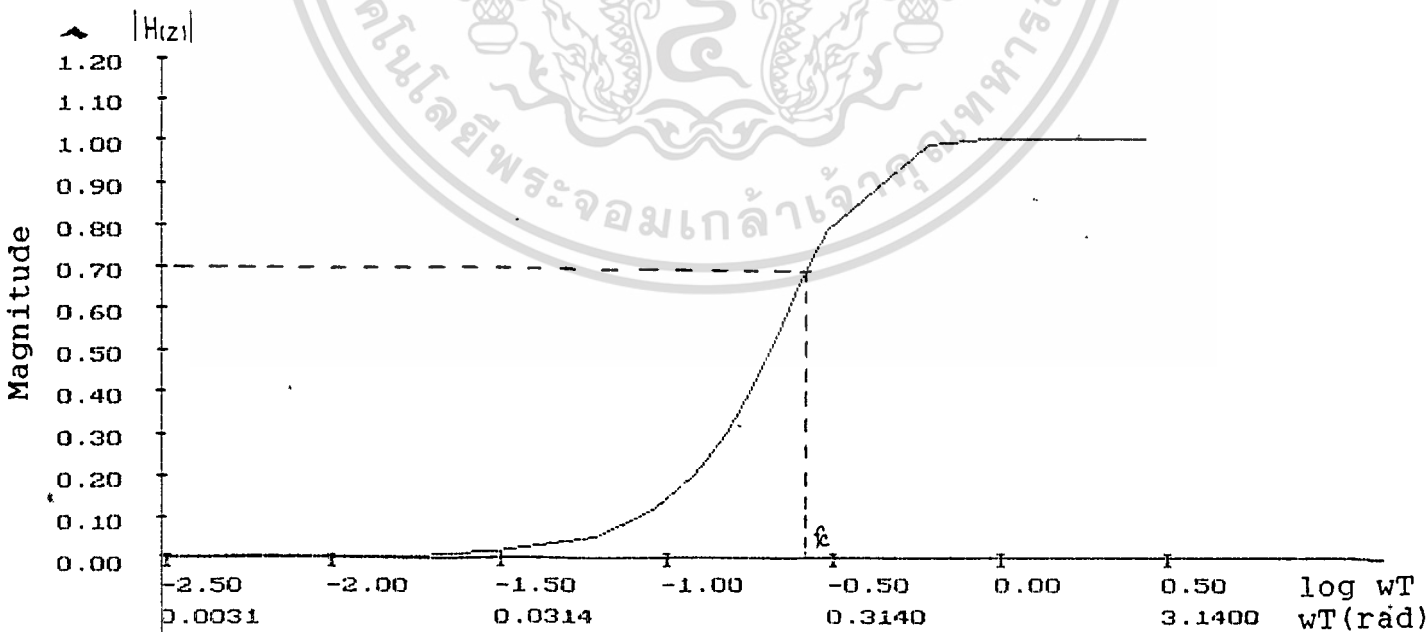


เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์และใช้เฉพาะเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำออก



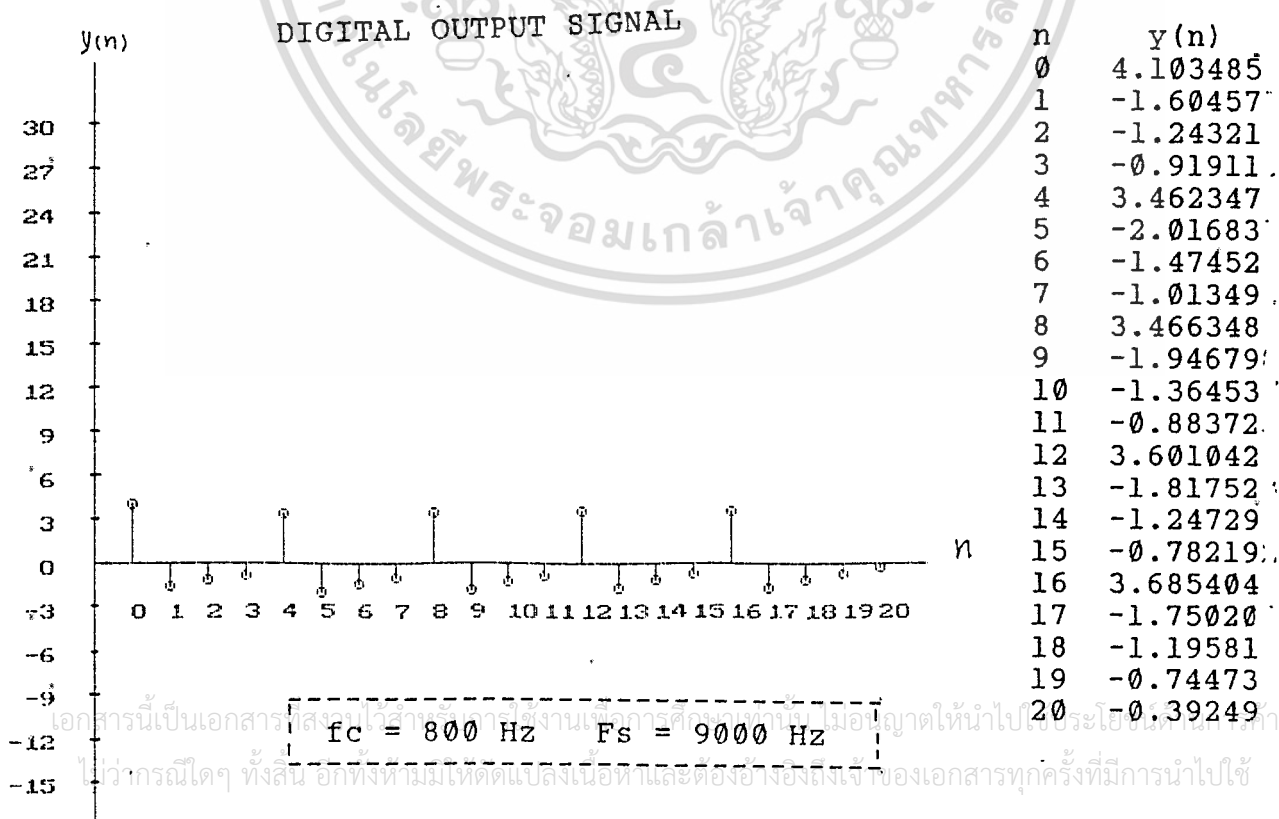
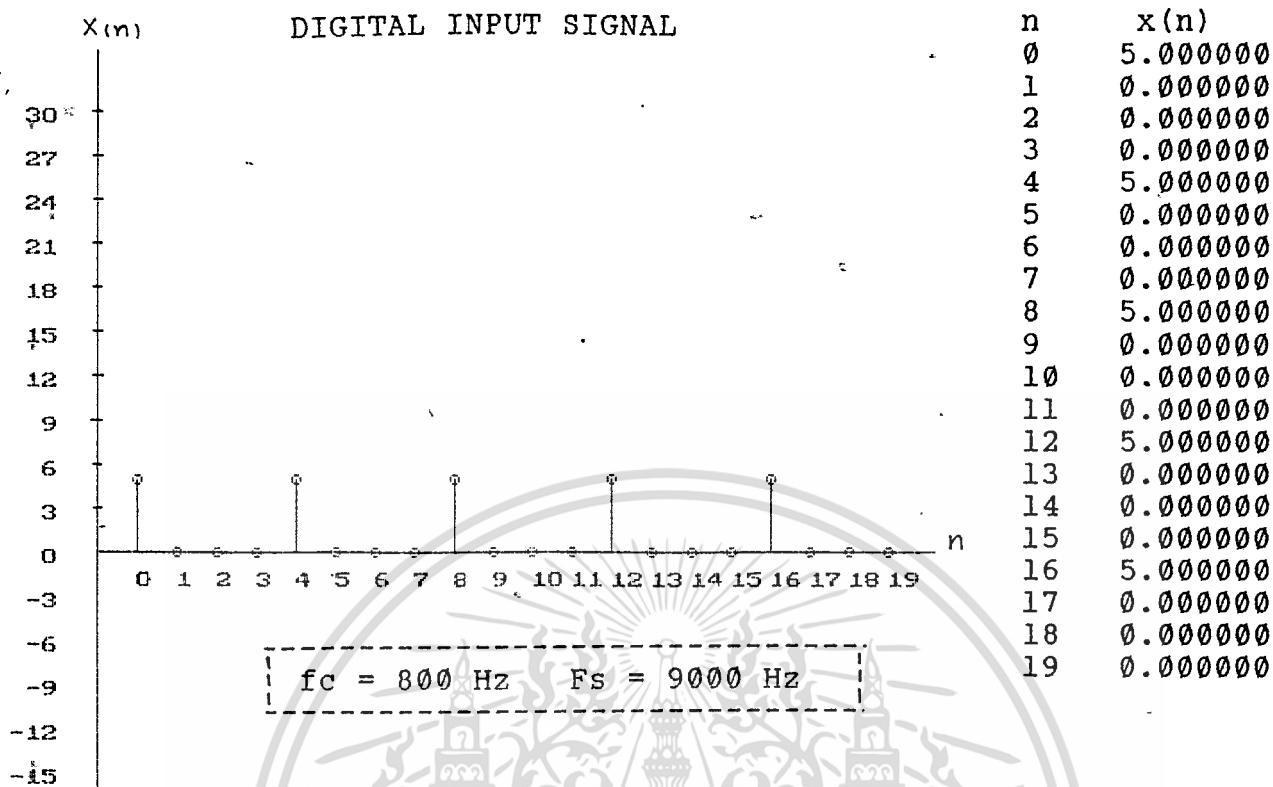
$a[0]=0.000000$ $b[0]=0.820697$
 $a[1]=-1.608974$ $b[1]=-1.641394$
 $a[2]=0.673814$ $b[2]=0.820697$

CHARACTERISTIC OF DIGITAL FILTER
 Design by Bilinear Transform
 Butterworth High Pass Filter
 2-order

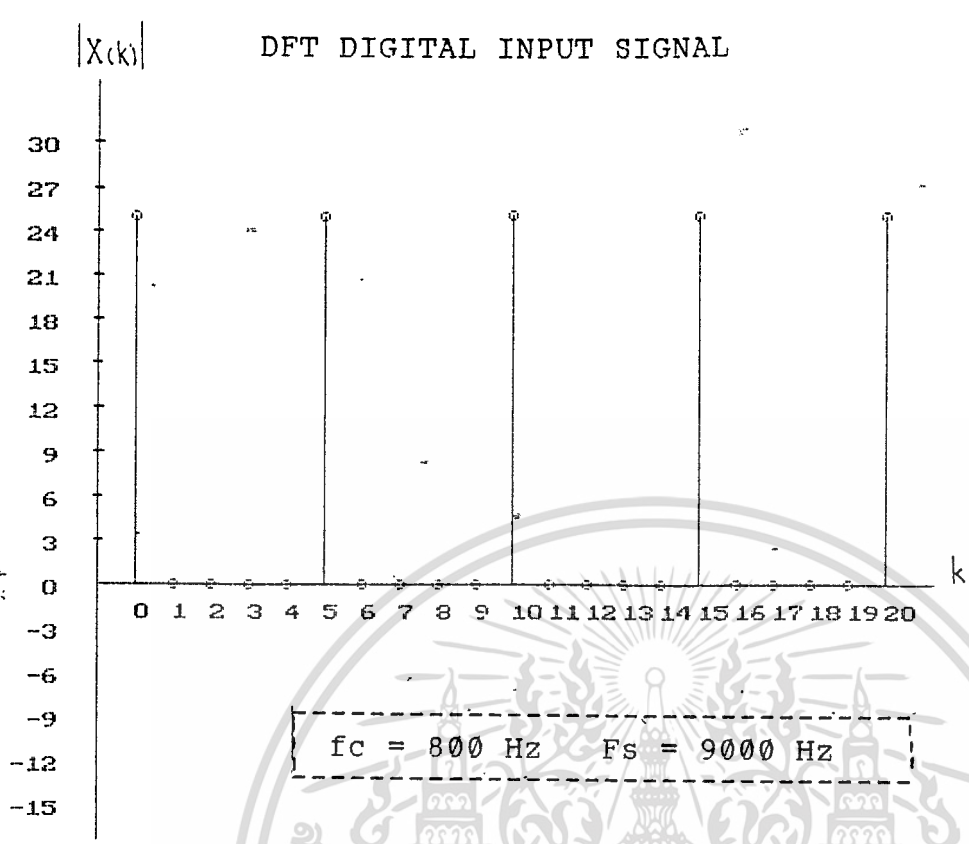


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูผู้สอนที่ขอใช้เอกสารฉบับนี้ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ทำกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่เอกสารฉบับนี้และข้อมูลอ้างอิงของเอกสารทุกครั้งที่มีการนำไปใช้

$f_c = 800 \text{ Hz}$ $F_s = 9000 \text{ Hz}$

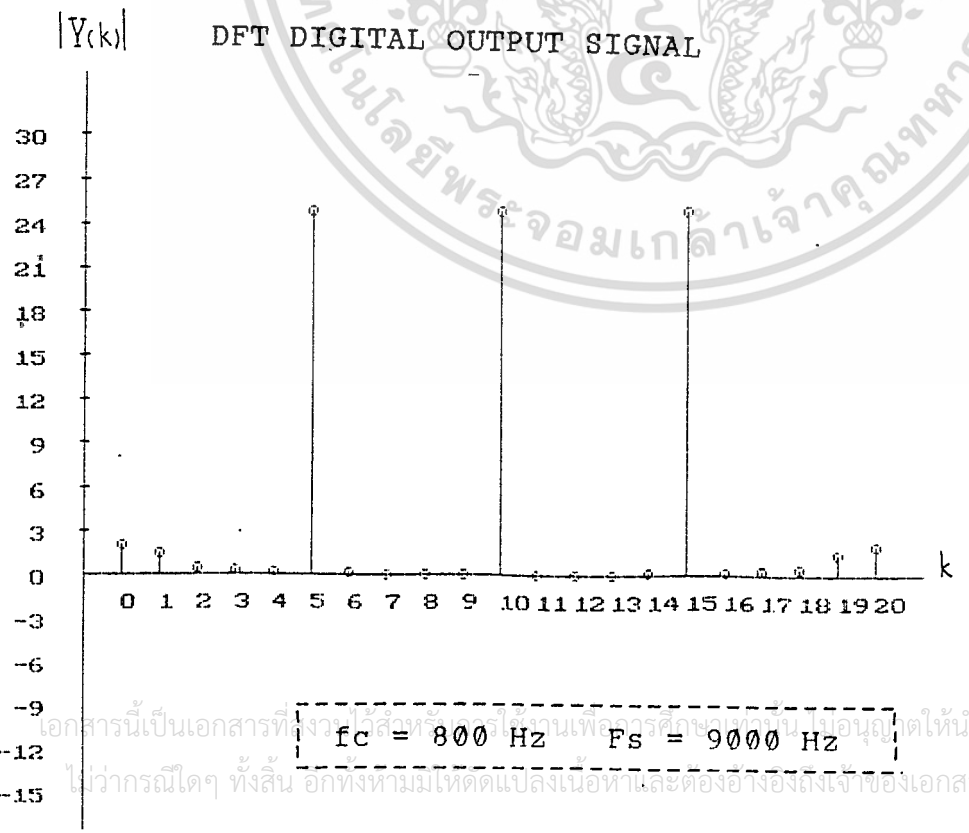


เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ภายใต้การคุ้มครองตามกฎหมาย ไม่อนุญาตให้นำไปใช้ประโยชน์อื่นใด
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



k	X(k)
0	25.000000
1	0.000000
2	0.000000
3	0.000000
4	0.000000
5	25.000000
6	0.000000
7	0.000000
8	0.000000
9	0.000000
10	25.000000
11	0.000000
12	0.000000
13	0.000000
14	0.000000
15	25.000000
16	0.000000
17	0.000000
18	0.000000
19	0.000000
20	25.000000

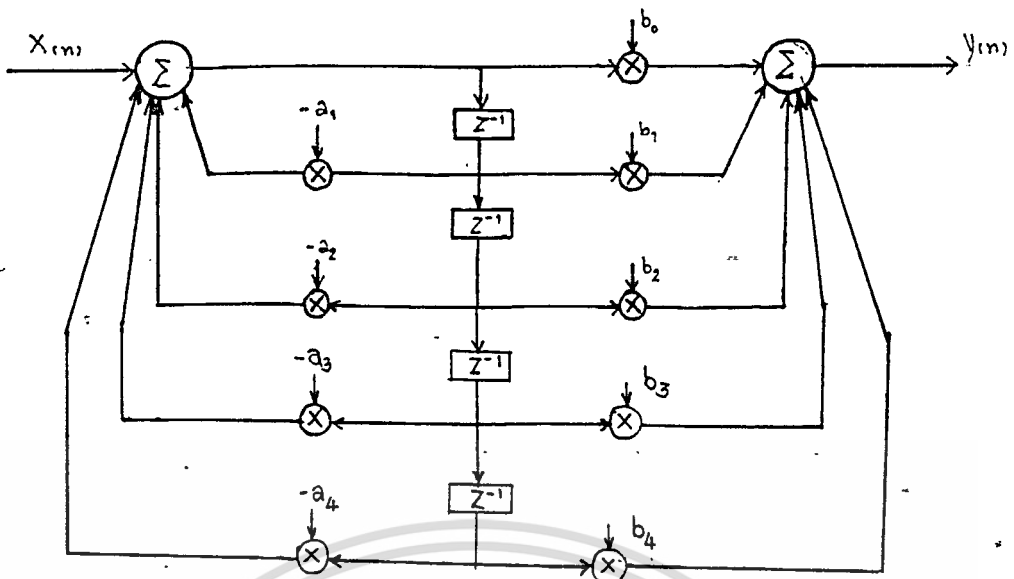
fc = 800 Hz Fs = 9000 Hz



k	Y(k)
0	2.078436
1	1.434907
2	0.554458
3	0.317594
4	0.225611
5	24.91679
6	0.152919
7	0.136787
8	0.127026
9	0.121746
10	24.87993
11	0.121746
12	0.127026
13	0.136787
14	0.152919
15	24.91679
16	0.225611
17	0.317594
18	0.554458
19	1.434907
20	2.078436

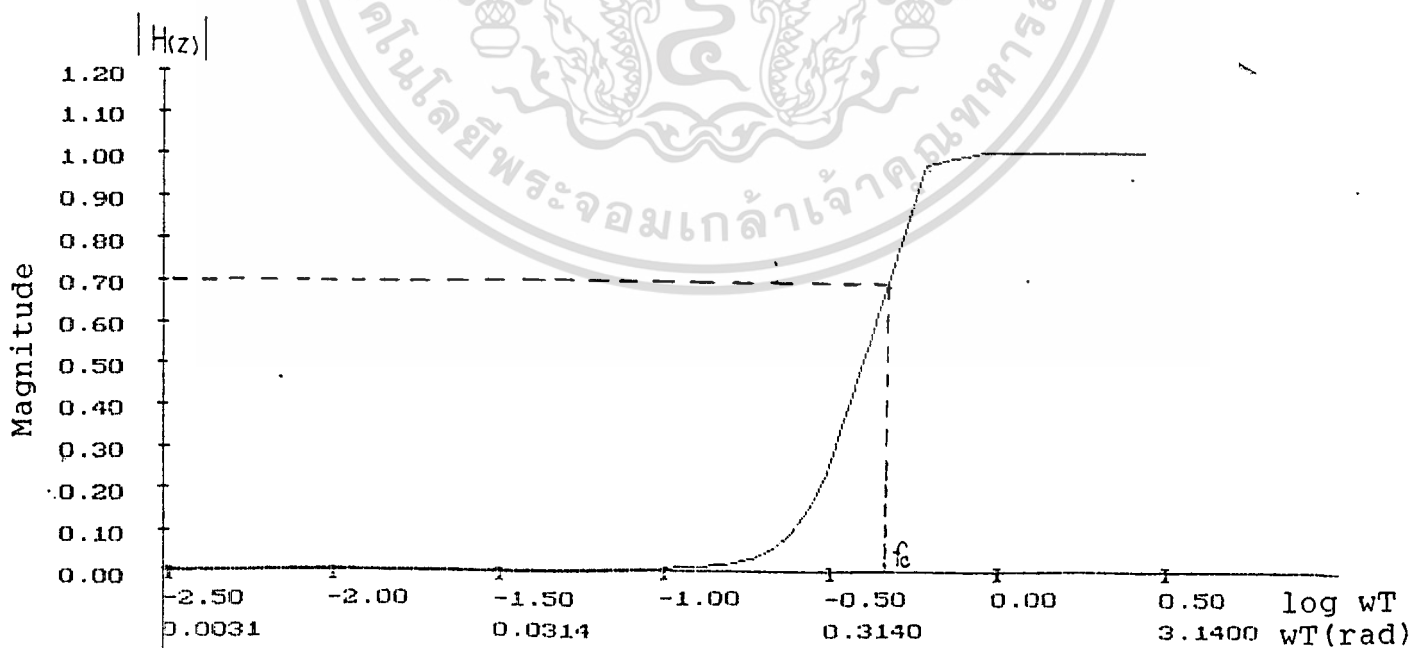
fc = 800 Hz Fs = 9000 Hz

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับงานเพื่อการศึกษาเท่านั้นขออภัยต่อนำไปใช้ประโยชน์ในกรณีที่ไม่ใช่
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



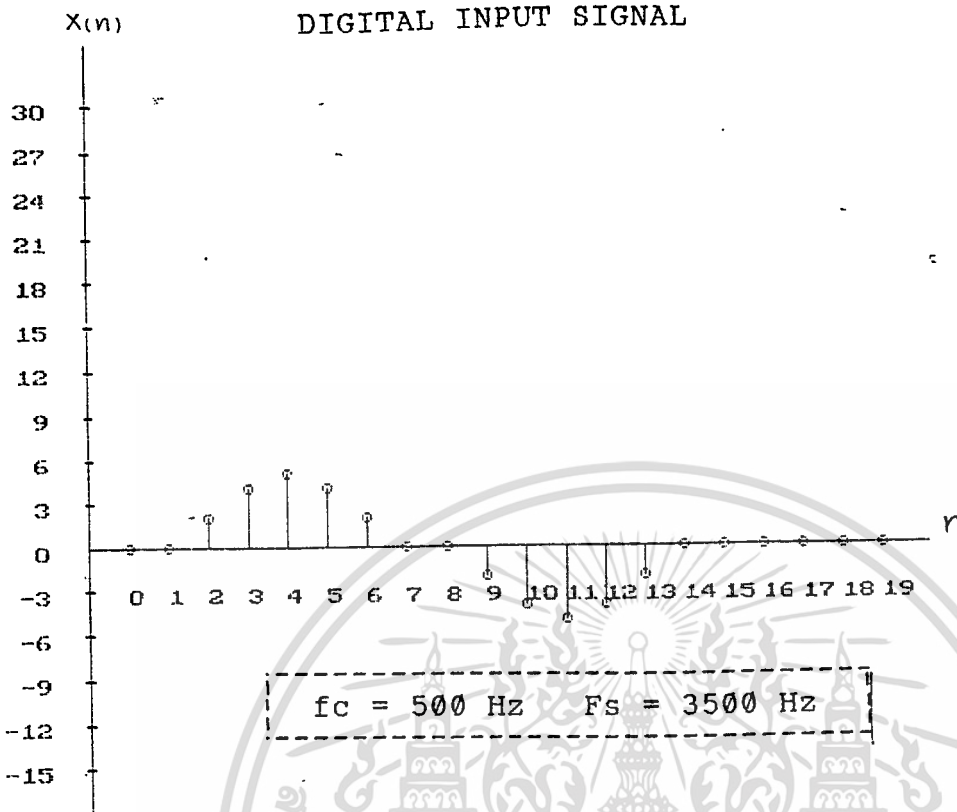
$a[0]=0.000000$ $b[0]=0.553086$
 $a[1]=-2.831694$ $b[1]=-2.212345$
 $a[2]=3.130788$ $b[2]=3.318518$
 $a[3]=-1.580988$ $b[3]=-2.212345$
 $a[4]=0.305910$ $b[4]=0.553086$

CHARACTERISTIC OF DIGITAL FILTER
 Design by Bilinear Transform
 Butterworth High Pass Filter
 4-order

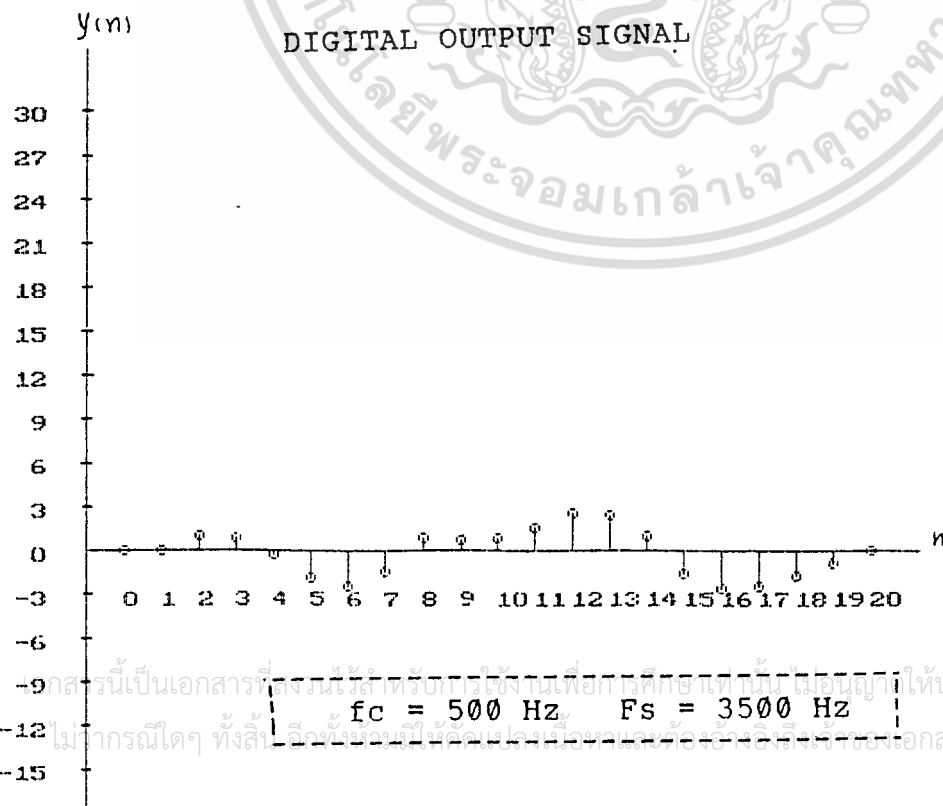


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่หวังกำไรใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$f_c = 500 \text{ Hz}$ $F_s = 3500 \text{ Hz}$

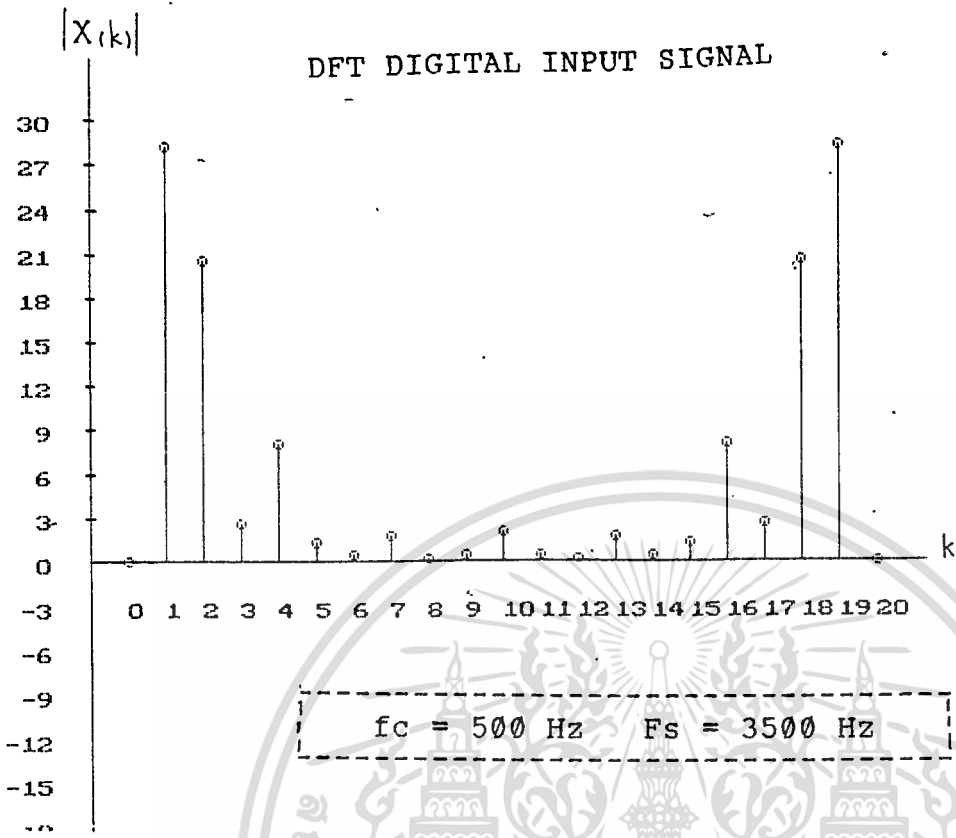


n	x(n)
0	0.000000
1	0.000000
2	2.000000
3	4.000000
4	5.000000
5	4.000000
6	2.000000
7	0.000000
8	0.000000
9	-2.000000
10	-4.000000
11	-5.000000
12	-4.000000
13	-2.000000
14	0.000000
15	0.000000
16	0.000000
17	0.000000
18	0.000000
19	0.000000

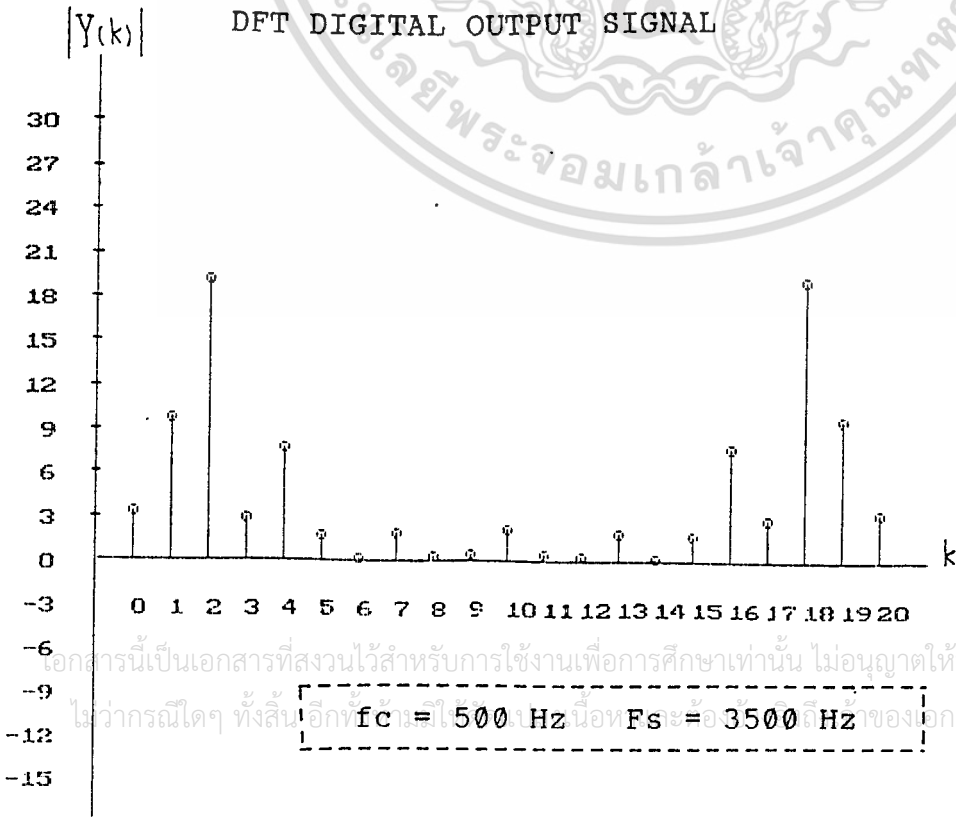


n	y(n)
0	0.000000
1	0.000000
2	1.106173
3	0.919997
4	-0.30495
5	-1.99500
6	-2.47221
7	-1.51818
8	0.933248
9	0.779016
10	0.958696
11	1.662597
12	2.652616
13	2.477365
14	1.045632
15	-1.66313
16	-2.66556
17	-2.55205
18	-1.83056
19	-0.89912
20	-0.03427

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการศึกษาเท่านั้น กรุณาไม่นำไปเผยแพร่โดยไม่ได้รับอนุญาต
 ไม่สามารถใดๆ ทั้งสิ้น ลิขสิทธิ์สงวนไว้ด้วยประการอันดี ขอสงวนสิทธิ์ในเอกสารทุกครั้งที่มีการนำไปใช้

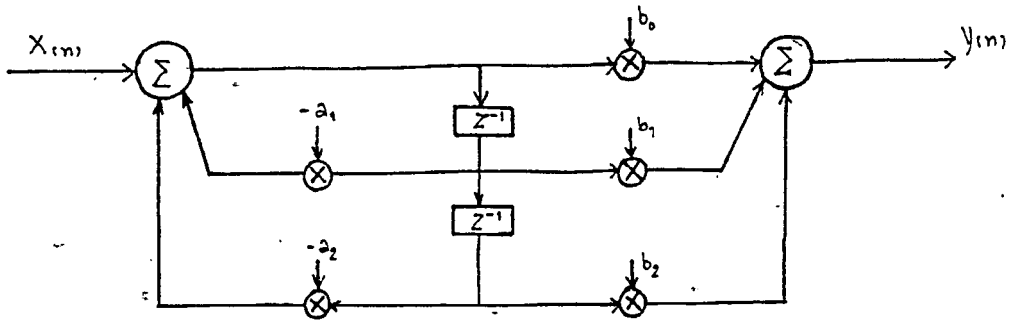


k	X(k)
0	0.000000
1	28.23514
2	20.56230
3	2.648815
4	8.057480
5	1.414214
6	0.437694
7	1.853595
8	0.277515
9	0.569863
10	2.000000
11	0.569863
12	0.277515
13	1.853595
14	0.437694
15	1.414214
16	8.057480
17	2.648815
18	20.56230
19	28.23514
20	0.000000



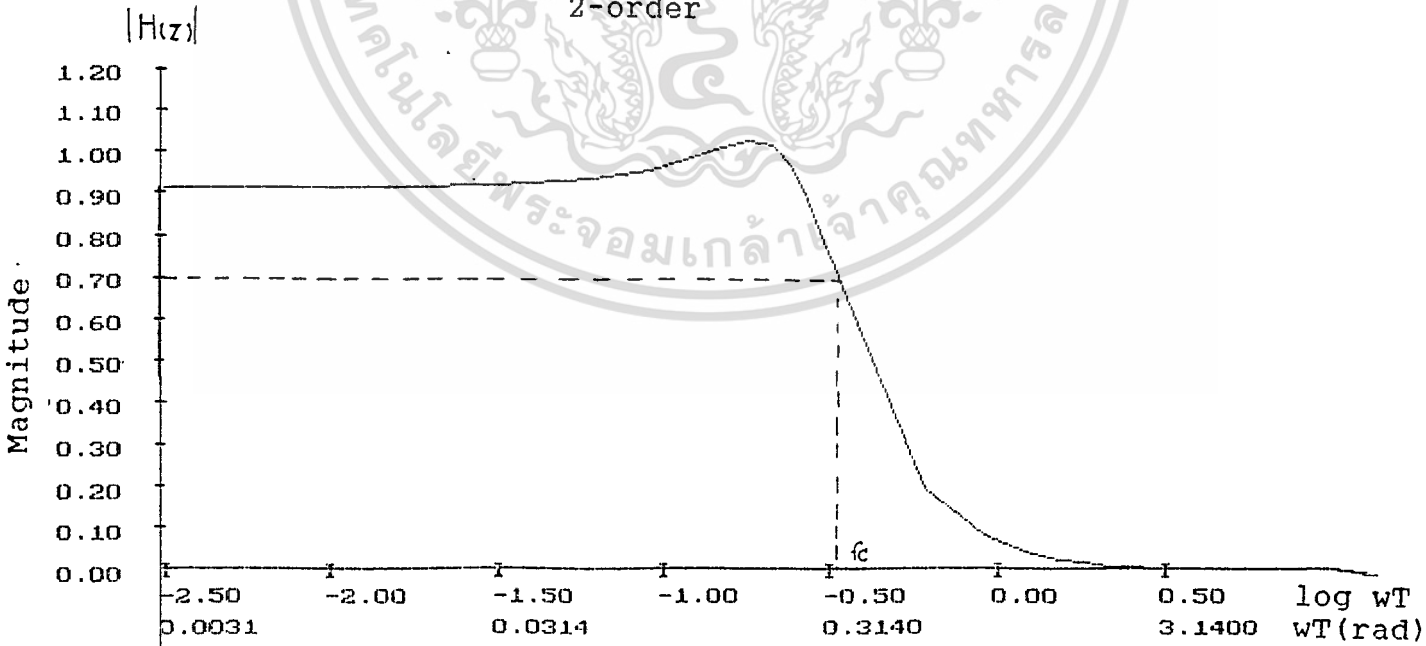
k	Y(k)
0	3.399744
1	9.724009
2	19.15466
3	2.906245
4	7.732569
5	1.785404
6	0.195493
7	1.980454
8	0.366771
9	0.446749
10	2.177309
11	0.446749
12	0.366771
13	1.980454
14	0.195493
15	1.785404
16	7.732569
17	2.906245
18	19.15466
19	9.724009
20	3.399744

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่
 ใ้ว่ากรณีใดๆ ทั้งสิ้น อีกที่ $f_c = 500 \text{ Hz}$ นือ $F_s = 3500 \text{ Hz}$ ของเอกสารทุก

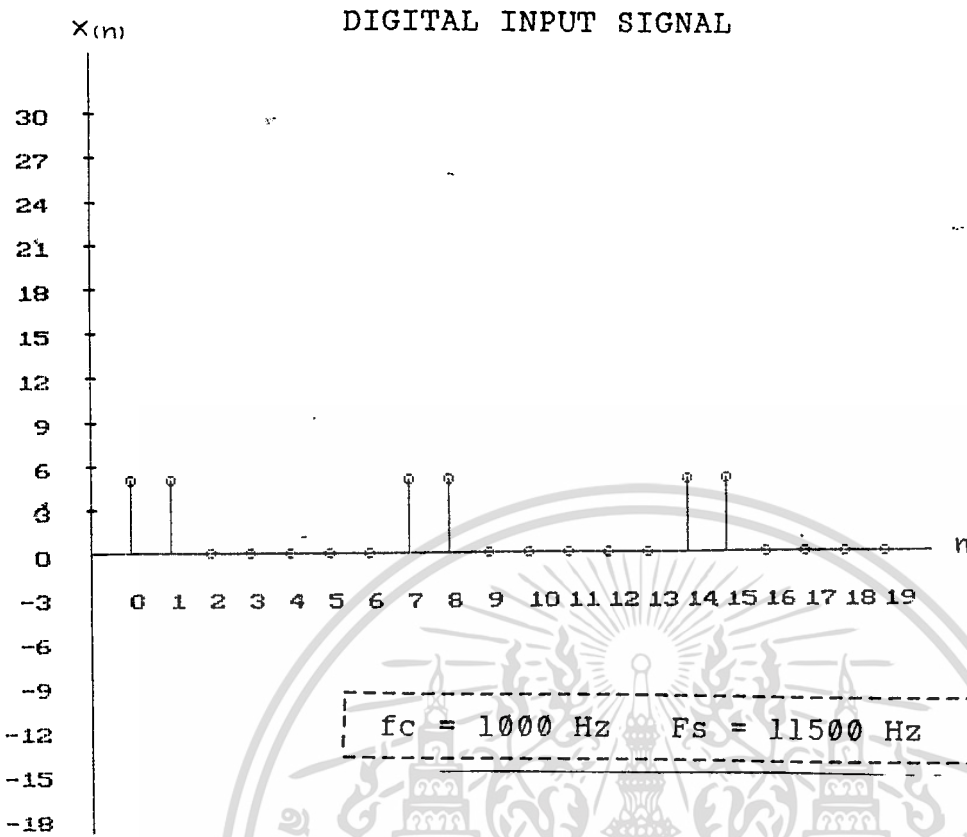


$a[0]=0.000000$ $b[0]=0.016125$
 $a[1]=-1.671650$ $b[1]=0.032250$
 $a[2]=0.742601$ $b[2]=0.016125$

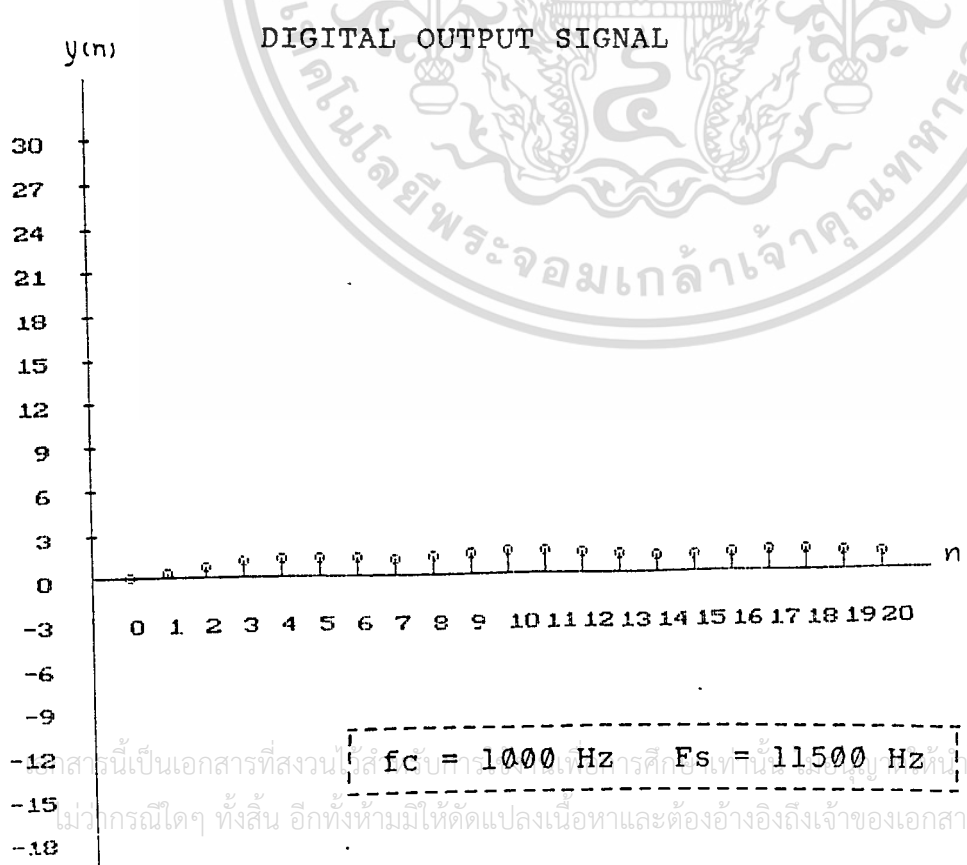
CHARACTERISTIC OF DIGITAL FILTER
 Design by Bilinear Transform
 Chebyshev Low Pass Filter
 2-order



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ทำกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแบบลงเนื้อหาและต้องอ้างอิงถึงชื่อของเอกสารทุกครั้งที่มีการนำไปใช้
 $f_c = 1000 \text{ Hz}$ $F_s = 11500 \text{ Hz}$



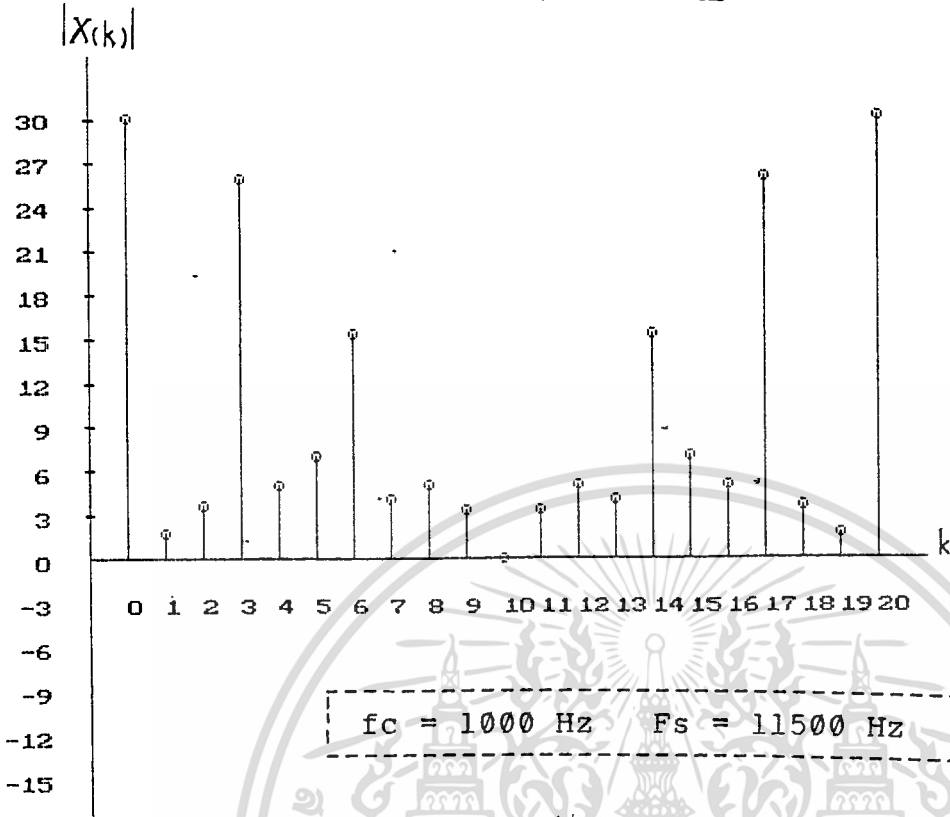
n	x(n)
0	5.000000
1	5.000000
2	0.000000
3	0.000000
4	0.000000
5	0.000000
6	0.000000
7	5.000000
8	5.000000
9	0.000000
10	0.000000
11	0.000000
12	0.000000
13	0.000000
14	5.000000
15	5.000000
16	0.000000
17	0.000000
18	0.000000
19	0.000000



n	y(n)
0	0.080626
1	0.376656
2	0.811643
3	1.157703
4	1.332548
5	1.367843
6	1.297003
7	1.233000
8	1.339866
9	1.566039
10	1.703509
11	1.684729
12	1.551250
13	1.342066
14	1.172131
15	1.204651
16	1.385208
17	1.501634
18	1.481549
19	1.361518
20	1.175781

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ภายใต้การคุ้มครองของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ไม่ทำกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

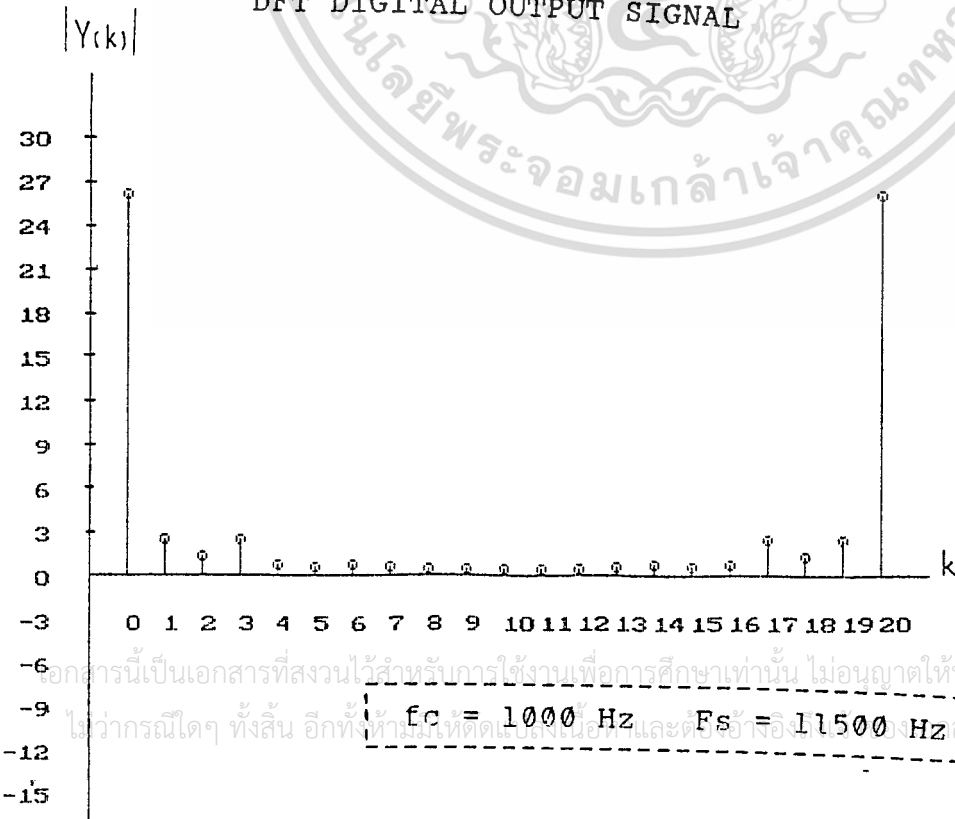
DFT DIGITAL INPUT SIGNAL



k	X(k)
0	30.00000
1	1.734089
2	3.632713
3	25.85801
4	5.000000
5	7.071068
6	15.38841
7	4.095507
8	5.000000
9	3.403342
10	0.000000
11	3.403342
12	5.000000
13	4.095507
14	15.38841
15	7.071068
16	5.000000
17	25.85801
18	3.632713
19	1.734089
20	30.00000

$f_c = 1000 \text{ Hz}$ $F_s = 11500 \text{ Hz}$

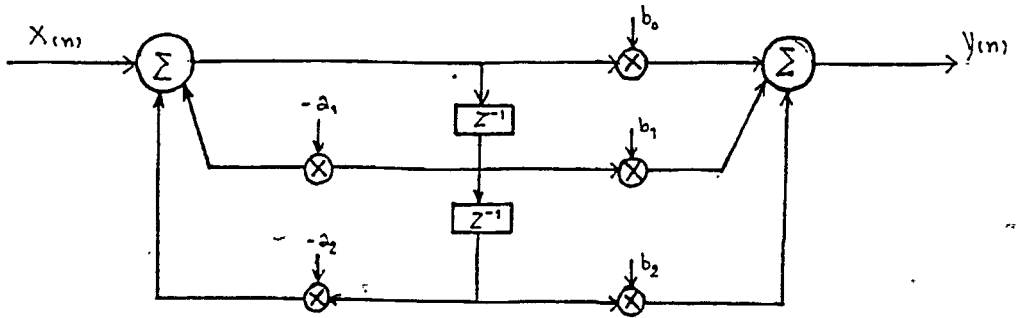
DFT DIGITAL OUTPUT SIGNAL



k	Y(k)
0	26.12695
1	2.453267
2	1.290423
3	2.522850
4	0.824576
5	0.630142
6	0.762580
7	0.596022
8	0.553911
9	0.543389
10	0.535276
11	0.543389
12	0.553911
13	0.596022
14	0.762580
15	0.630142
16	0.824576
17	2.522850
18	1.290423
19	2.453267
20	26.12695

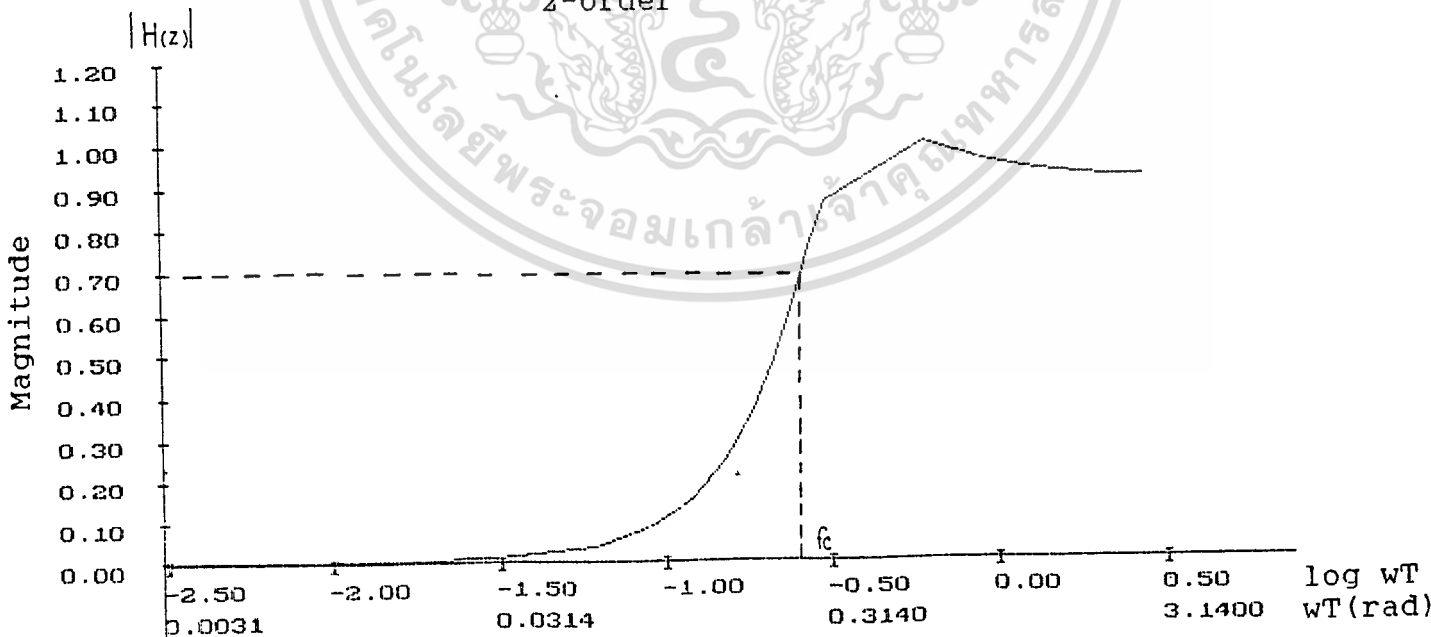
$f_c = 1000 \text{ Hz}$ $F_s = 11500 \text{ Hz}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้
 ใกว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุใดเบี่ยงเบนเนื้อหาและที่ยังอ้างอิงเอกสารทุกครั้งที่มีการนำไปใช้



$a[0]=0.000000$ $b[0]=0.760947$
 $a[1]=-1.630547$ $b[1]=-1.521895$
 $a[2]=0.717622$ $b[2]=0.760947$

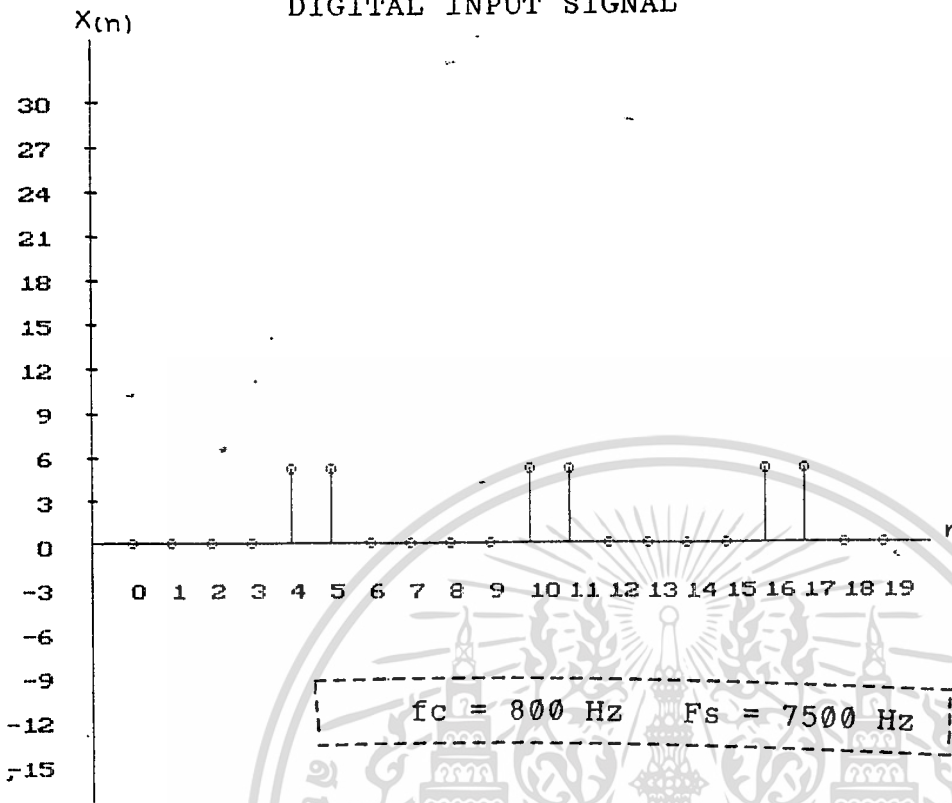
CHARACTERISTIC OF DIGITAL FILTER
 Design by Bilinear Transform
 Chebyshev High Pass Filter
 2-order



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในมหาวิทยาลัยเท่านั้น อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ควรกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ข้อมูลหรือข้อมูลลิขสิทธิ์ของเอกสารทุกครั้งที่มีการนำไปใช้

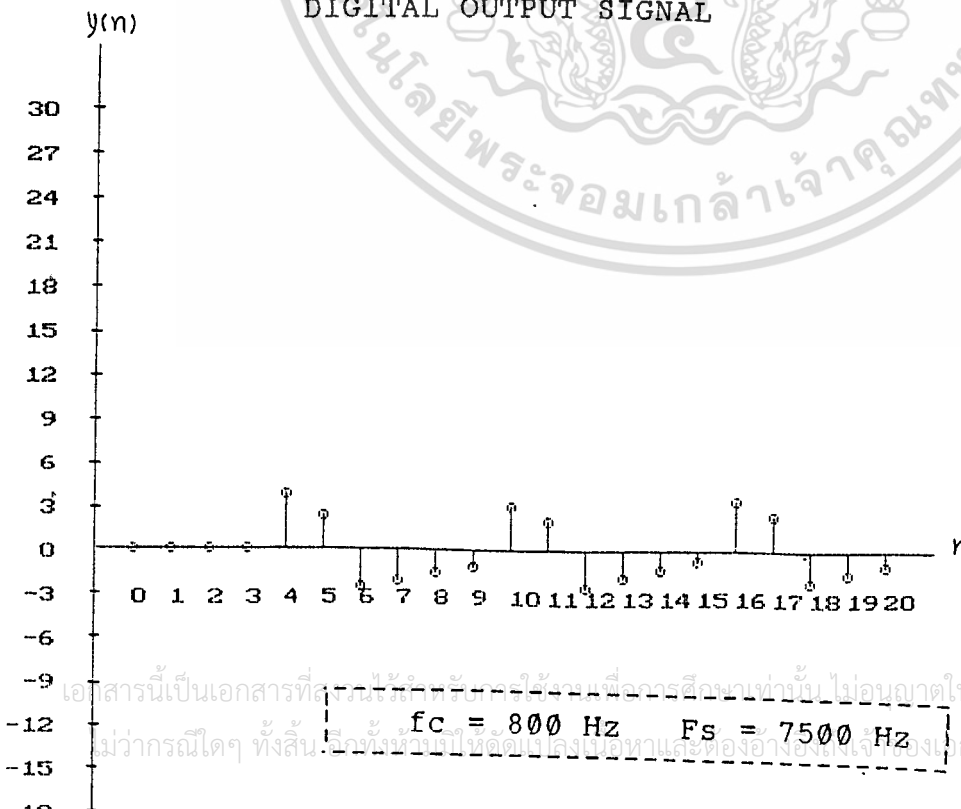
$f_c = 800 \text{ Hz}$ $F_s = 7500 \text{ Hz}$

DIGITAL INPUT SIGNAL



n	x(n)
0	0.000000
1	0.000000
2	0.000000
3	0.000000
4	5.000000
5	5.000000
6	0.000000
7	0.000000
8	0.000000
9	0.000000
10	5.000000
11	5.000000
12	0.000000
13	0.000000
14	0.000000
15	0.000000
16	5.000000
17	5.000000
18	0.000000
19	0.000000

DIGITAL OUTPUT SIGNAL



n	y(n)
0	0.000000
1	0.000000
2	0.000000
3	0.000000
4	3.804737
5	2.399065
6	-2.62331
7	-2.19431
8	-1.69539
9	-1.18972
10	3.081485
11	2.073542
12	-2.63507
13	-1.97988
14	-1.33731
15	-0.75974
16	3.525623
17	2.489168
18	-2.27609
19	-1.69282
20	-1.12685

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์และใช้เฉพาะเพื่อการศึกษานานาชาติเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น. มีข้อยกเว้นที่ชัดเจนในเรื่องเนื้อหาและต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
การทดลอง
วงจรกรองความถี่ดิจิทัล

ในการทดลองนี้ เราใช้วงจรสวิตช์คาปาซิเตอร์ฟิลเตอร์มาสร้างเป็นวงจรกรองความถี่ดิจิทัล โดยเลือกทดลองออกแบบเป็นวงจรกรองความถี่ขั้นดับที่ 2 ดังได้กล่าวไปแล้ว เราสร้างวงจรสวิตช์คาปาซิเตอร์ฟิลเตอร์โดยอาศัยจากวงจร RC active filter แบบ Butterworth โดยการแทนที่ความต้านทานในวงจร RC active filter ด้วยวงจรสวิตช์คาปาซิเตอร์ที่ต่อกันแบบ Bilinear ซึ่งจะพบความสัมพันธ์ระหว่าง analog frequency กับ digital frequency มีค่าใกล้เคียงกันมาก

จาก Bilinear Transform
$$s = \frac{2 \cdot 1 - z^{-1}}{T \cdot 1 + z^{-1}} \quad \text{-----(7.1)}$$

ในกรณีของอนาล็อก
$$s = j\omega_a$$

ในกรณีของดิจิทัล
$$z = \exp(j\omega_d T)$$

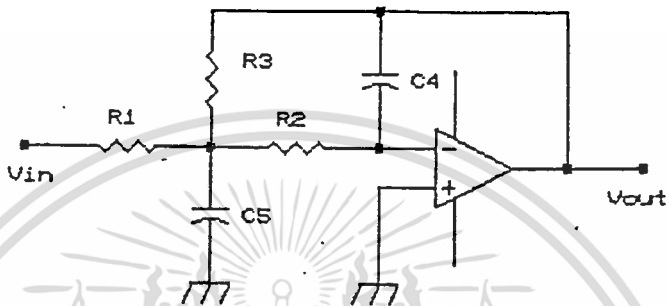
แทนค่า s และ z ลงในสมการ (7.1) จะได้

$$\omega_a = \frac{2 \tan \frac{\omega_d T}{2}}{T} \quad \text{-----(7.2)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.1 วงจร Infinite Gain Multiple Feedback Low Pass Filter

อันดับสอง



รูปที่ 7.1 แสดงวงจร Infinite gain multiple feedback low passfilter
อันดับสอง

ทรานส์เฟอร์ฟังก์ชันของวงจรรูปที่ 7.1 คือ

$$H(s) = \frac{-H\omega_0^2}{s^2 + \frac{\omega_0 s}{Q} + \omega_0^2} \quad \text{----- (7.3)}$$

$$= \frac{-1/R_1 R_2 C_4 C_5}{s^2 + (1/R_1 + 1/R_2 + 1/R_3) \frac{1}{C_5} s + \frac{1}{R_2 R_3 C_4 C_5}} \quad \text{----- (7.4)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปรียบเทียบสมการ (7.3) กับ (7.4) จะได้ว่า

$$\begin{aligned} \omega_0^2 &= 1/(R_2 R_3 C_4 C_5) \\ \omega_0 &= 1/(R_1 + 1/R_2 + 1/R_3) \\ \frac{Q}{C_5} & \\ H &= -R_3/R_1 \end{aligned}$$

จากสมการ prototype ของ Butterworth lowpass filter order 2 มีทรานส์ฟอร์มฟังก์ชันดังนี้

$$H(s) = \frac{K}{s^2 + \sqrt{2}s + 1} \quad (7.5)$$

เมื่อ K คือ เกนของวงจร สมมติให้เป็น 1 และเมื่อต้องการสร้างเป็น lowpass filter ที่ความถี่ cutoff ต่าง ๆ ก็แทน s ด้วย s/ω_a ลงในสมการที่ (7.5) จะได้

$$H(s) = \frac{1}{\frac{s^2}{\omega_a^2} + \frac{\sqrt{2}s}{\omega_a} + 1} \quad (7.6)$$

แล้วทำการเปรียบเทียบการที่ (7.3) กับ (7.6) ก็สามารถหาความสัมพันธ์การออกแบบวงจร Active RC Lowpass Filter ได้ดังนี้

1. เลือก C_4
2. คำนวณ $C_5 = 4C_4(1+H)Q^2$
3. คำนวณ $R_1 = 1/(2H\omega_aQC_4)$
4. คำนวณ $R_2 = 1/[2\omega_aQ(1+H)C_4]$
5. คำนวณ $R_3 = 1/(2\omega_aQC_4)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แปลงจาก R ไปเป็น C โดยใช้สูตร $C_m = 1/(4F_s R_m)$

ทราบนส์เฟอร์ฟังก์ชันของดิจิทัลฟิลเตอร์อันดับที่ 2 คือ

$$H(s) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad \text{-----(7.7)}$$

แทนสมการ (7.1) ซึ่งแสดง Bilinear Transform ลงในสมการ

(7.4) แล้วสามารถจัดรูปให้อยู่ในเทอมต่าง ๆ ดังสมการ (7.7) จะสามารถหาค่าสัมประสิทธิ์ออกมาได้ดังนี้

$$b_0 = \frac{2 * C_2 C_3 / C_4 C_5}{1 + \frac{1(C_1 + C_2 + C_3)}{C_5} + C_2 C_3 / C_4 C_5}$$

$$b_1 = 2 * b_0$$

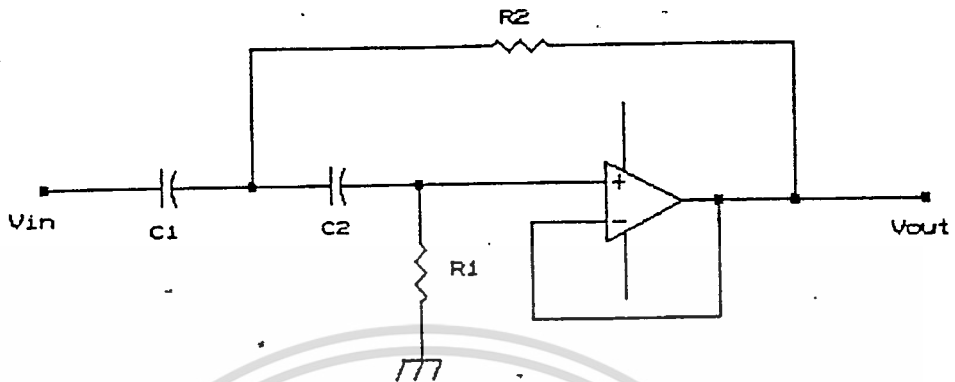
$$b_2 = b_0$$

$$a_1 = \frac{2 * C_2 C_3 / C_4 C_5 - 2}{1 + \frac{1(C_1 + C_2 + C_3)}{C_5} + C_2 C_3 / C_4 C_5}$$

$$a_2 = \frac{1 - [(1/C_5)(C_1 + C_2 + C_3)] + C_2 C_3 / C_4 C_5}{1 + \frac{1(C_1 + C_2 + C_3)}{C_5} + C_2 C_3 / C_4 C_5}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2 วงจร Sallen and Key (finite gain) High Pass Filter อันดับที่ 2



รูปที่ 7.2 แสดงวงจร Sallen and Key (finite gain) High Pass Filter อันดับที่ 2

ทรานส์เฟอร์ฟังก์ชันของวงจรรูปที่ 7.3 คือ

$$H(s) = \frac{Hs^2}{s^2 + \frac{\omega_0 s}{Q} + \omega_0^2} \quad (7.8)$$

$$H(s) = \frac{s^2}{s^2 + \frac{1(1/C_1 + 1/C_2)s + (1/R_1 R_2 C_1 C_2)}{R_1}} \quad (7.9)$$

จากสมการ prototype ของ Butterworth Lowpass Filter อันดับที่ 2 มีทรานส์เฟอร์ฟังก์ชันดังนี้

$$H(s) = \frac{k}{s^2 + \sqrt{2}s + 1} \quad (7.5)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ k คือเกนของวงจร สมมติให้เป็น 1 และเมื่อต้องการสร้างเป็น High pass filter ที่ความถี่ cutoff ต่าง ๆ ก็แทน s ด้วย ω_a/s ลงในสมการที่ (7.5) จะได้

$$H(s) = \frac{1}{\frac{\omega_a^2}{s^2} + \frac{\sqrt{2}\omega_a}{s} + 1} \quad \text{-----(7.10)}$$

ทำการเปรียบเทียบสมการที่ (7.9) กับ (7.10) ก็สามารถหาค่าความสัมพันธ์ การออกแบบวงจร Active RC High Pass Filter ได้ดังนี้

ถ้ากำหนดให้วงจรที่ออกแบบเป็น unity gain และ $C_1 = C_2 = C$

1. ถ้า $C_1 = C_2 = C$ เลือกค่า C ที่เหมาะสม
2. $R_1 = \sqrt{2}/(\omega_a C)$
3. $R_2 = R_1/2$

แปลงจาก R ไปเป็น C โดยใช้สูตร $C_m = 1/(4F_s R_m)$

ทรานส์เฟอร์ฟังก์ชันของดิจิตอลฟิลเตอร์อันดับที่ 2

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

แทนสมการ (7.1) ซึ่งแสดง Bilinear Transform ลงในสมการที่ (7.9) แล้วสามารถจัดรูปให้อยู่ในเทอมต่าง ๆ ของทรานส์เฟอร์ฟังก์ชันของดิจิตอลฟิลเตอร์ จะสามารถหาค่าสัมประสิทธิ์ออกมาได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$b_0 = \frac{1}{1 + 2(c_1/c) + c_1c_2/c^2}$$

$$b_1 = -2*b_0$$

$$b_2 = b_0$$

$$a_1 = \frac{2c_1c_2/c^2 - 2}{1 + 2(c_1/c) + c_1c_2/c^2}$$

$$a_2 = \frac{1 - 2(c_1/c) + c_1c_2/c^2}{1 + 2(c_1/c) + c_1c_2/c^2}$$



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*****
/*_calculate RC equipment in switch capacitor filter(digital filter)*/
/*LPF,HPF order 2 .
/*_calculate coefficient of digital filter.
/*****

#include<stdio.h>
#include<math.h>
main()
{
float H,Q,c1,c2,c3,c4,c5,R1,R2,R3;
float a1,a2,b0,b1,b2,Fc,Fs,W0,WA,c,n,Fca ;
clrscr();
H = 1;
Q = 0.707;

printf("select 1 for LPF ,2 for HPF.");
scanf("%f",&n);
printf("frequency cutoff Fc=");
scanf("%f",&Fc); /*hertz*/
printf("sampling freq Fs=");
scanf("%f",&Fs); /*hertz*/

WA = 2*(2*Fs)*tan((2*M_PI*Fc)/(2*2*Fs));
printf(" WA = %8f \n", WA);
Fca=WA/(2*M_PI);
printf("Fca=%f\n",Fca);
if(n==1)/* LPF order 2 */
{
c5 = 100e-9; /*farad*/
c4 = c5/(4*Q*Q*(1+H));
printf(" c4 = %e \n", c4 );
R1 = 1/(2*H*WA*Q*c4);
printf(" R1 = %8f \n", R1);
c1 = 1/(4*Fs*R1);
printf(" c1 = %e \n", c1);
R2 = 1/(2*WA*Q*(1+H)*c4);
printf(" R2 = %8f \n", R2);
c2 = 1/(4*Fs*R2);
printf(" c2 = %e \n", c2);
R3 = 1/(2*WA*Q*c4);
printf(" R3 = %8f \n", R3);
c3 = 1/(4*Fs*R3);
printf(" c3 = %e \n", c3);

/*find coefficient of digital filter LPF order 2*/
b0 = ((2*(c1*c3))/(c4*c5))/(1+((c1+c2+c3)/c5)+((c2*c3)/(c4*c5)));
b1 = 2*b0;
b2 = b0;
a1 = (((2*c2*c3)/(c4*c5))-2)/(1+((c1+c2+c3)/c5)+((c2*c3)/(c4*c5)));
a2 = (1-((c1+c2+c3)/c5)+((c2*c3)/(c4*c5)))/(1+((c1+c2+c3)/c5)+((c2*c3)/(c4*c5)
printf (" b0 = %8e \n", b0);
printf (" b1 = %8e \n", b1);
printf (" b2 = %8e \n", b2);
printf (" a1 = %8e \n", a1);
printf (" a2 = %8e \n", a2);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else/*HPF order 2 */
{
    c = 50e-09; /*farad*/
    R1= 1.414/(WA*c) ;
    printf ( " R1 = %e \n" , R1 );
    c1 = 1/(4*Fs*R1);
    printf ("c1=%e \n " , c1);
    R2 = R1/2;
    printf ("R2=%e \n.", R2 );
    c2 = 1/(4*Fs*R2);
    printf ("c2=%e \n ",c2);

    /*find coefficient of digital filter HPF order 2*/
    b0 = 1/(1+ (2*(c1/c)) +(( c1*c2)/(c*c)));
    b1 = -2*b0 ;
    b2 = b0 ;
    a1 = ( 2*(-(c1*c2)/(c*c)) -2 )/ (1+(2*(c1/c))+((c1*c2)/(c*c)) ) ;
    a2 = ( 1-(2*(c1/c))+((c1*c2)/(c*c)))/(1+(2*(c1/c))+((c1*c2)/(c*c)))

    printf ("b0 = %8e \n", b0);
    printf ("b1 = %8e \n", b1);
    printf ("b2 = %8e \n", b2);
    printf ("a1 = %8e \n", a1);
    printf ("a2 = %8e \n", a2);
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการทดลอง Lowpass filter

สมมุติว่าต้องการออกแบบวงจรสวิทช์คาปาซิเตอร์ฟิลเตอร์โดย Frequency cutoff อยู่ที่ 1 kHz กำหนดให้ความถี่ของสัญญาณป้อนที่ใช้ควบคุมการทำงานของอนุบาลสวิทช์เท่ากับ 31.25 kHz และ gain เท่ากับ 1

ขั้นตอนการออกแบบ

จากสมการ prototype อันดับ 2 ของ Butterworth

$$H(s) = \frac{1}{s^2 + \sqrt{2}s + 1}$$

จะได้ $Q = 0.707$

1. เลือก $C_4 = 25 \text{ nF}$

2. คำนวณ $C_5 = 4C_4Q^2(1+H)$
 $= 4 \cdot (25 \cdot 10^{-9}) \cdot (0.707)^2 \cdot (1+1)$
 $= 100 \text{ nF}$

3. คำนวณ $\omega_a = \frac{2 \tan(\omega_d T/2)}{T}$
 $= \frac{2 \tan[\omega_d (T_s/2)/2]}{(T_s/2)}$
 $= 2 \cdot 2 \cdot 31.25 \cdot 10^3 \tan[(2\pi \cdot 1 \cdot 10^3) / (2 \cdot 2 \cdot 31.25 \cdot 10^3)]$
 $= 2\pi \cdot 1000.84 \text{ rad/sec}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. คำนวณหาค่า element ต่าง ๆ ดังนี้

$$\begin{aligned}
 R_1 &= 1/(2H\omega_a Q C_4) \\
 &= 1/(2 * 1 * 2\pi * 1000 * 0.84 * (0.707) * 25 * 10^{-9}) \\
 &= 4498 \text{ ohm}
 \end{aligned}$$

$$\begin{aligned}
 C_1 &= 1/(4F_S R_1) \\
 &= 1/(4 * 31.25 * 10^3 * 4498) \\
 &= 1.77 \text{ nF}
 \end{aligned}$$

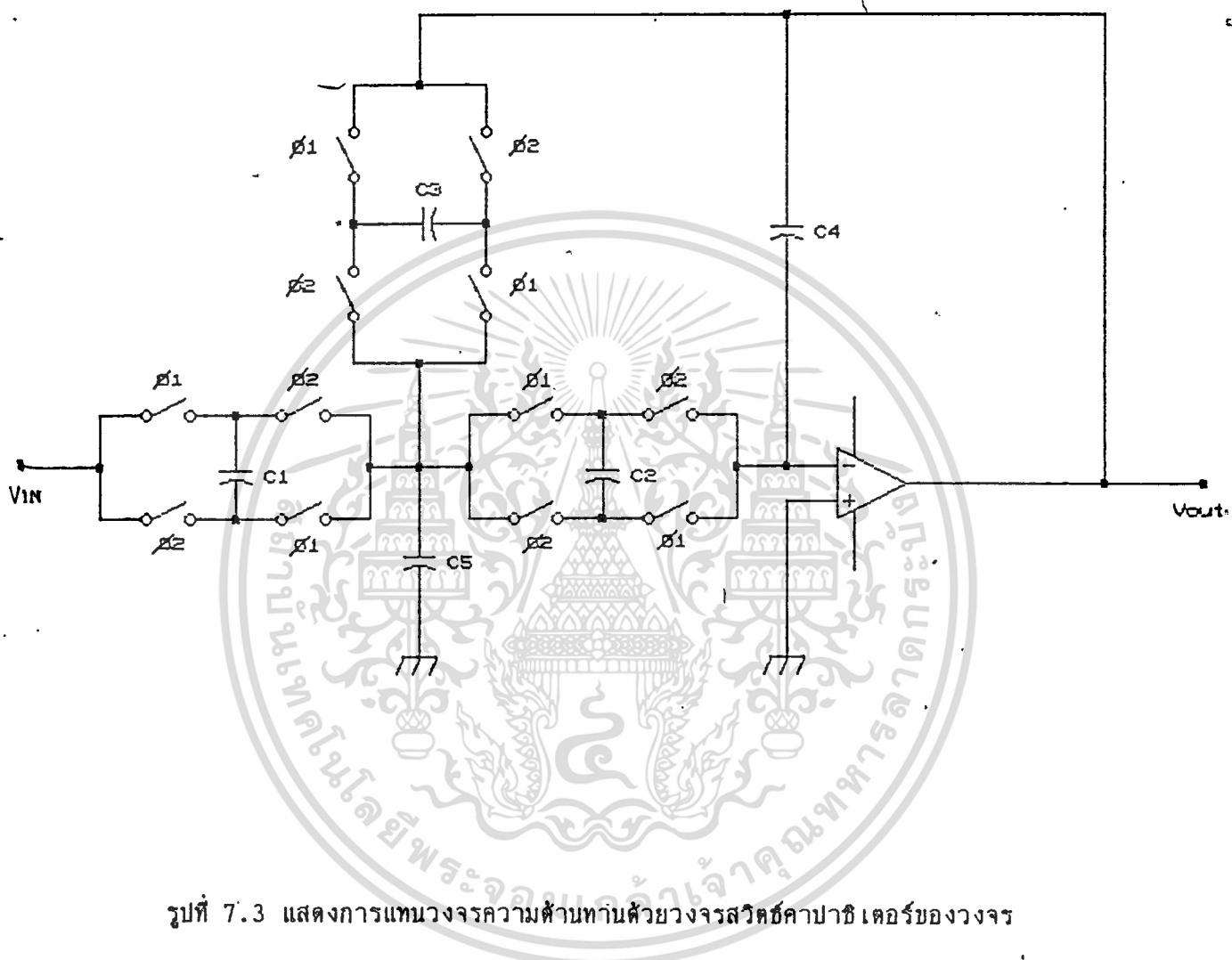
$$\begin{aligned}
 R_2 &= 1/(\omega_a Q (1+H) C_4) \\
 &= 1/(2 * 2\pi * 1000 * 0.84 * (0.707) * (2) * 25 * 10^{-9}) \\
 &= 2249 \text{ ohm}
 \end{aligned}$$

$$\begin{aligned}
 C_2 &= 1/(4F_S R_2) \\
 &= 1/(4 * 31.25 * 10^3 * 2249) \\
 &= 3.55 \text{ nF}
 \end{aligned}$$

$$\begin{aligned}
 R_3 &= 1/(2\omega_a Q C_4) \\
 &= 1/(2 * 2\pi * 1000 * 0.84 * 0.707 * 25 * 10^{-9}) \\
 &= 4498 \text{ ohm}
 \end{aligned}$$

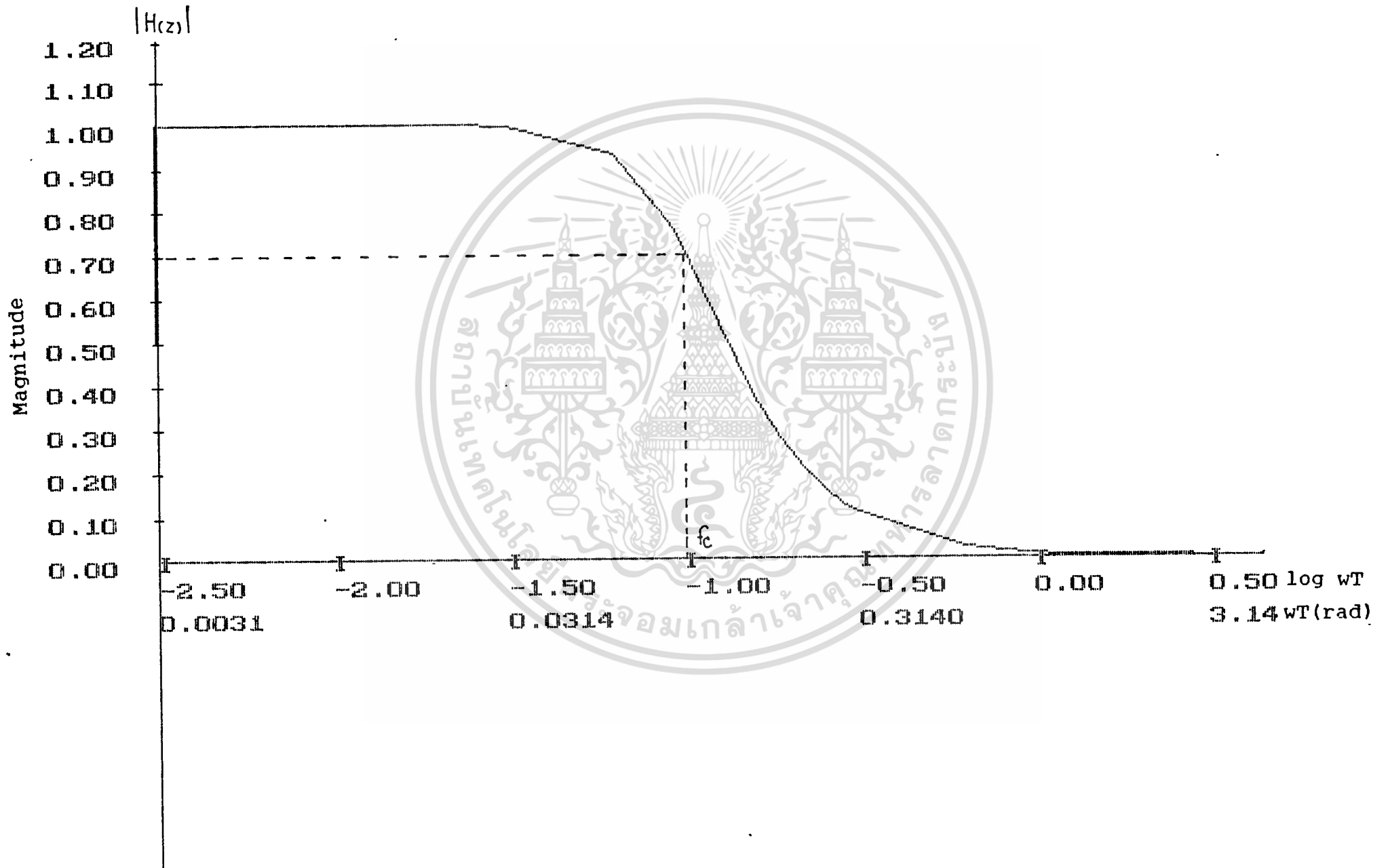
$$\begin{aligned}
 C_3 &= 1/(4F_S R_3) \\
 &= 1/(4 * 31.25 * 10^3 * 4498) \\
 &= 1.77 \text{ nF}
 \end{aligned}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.3 แสดงการแทนวงจรความต้านทานด้วยวงจรสวิตช์คาปาซิเตอร์ของวงจร
Infinite Gain Multiple Feedback Lowpass Filter อันดับที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



สรุป

วงจรกรองความถี่ที่ตัดคอลที่เราออกแบบ โดยสร้างจากวงจรสวิทช์คาปาซิเตอร์ฟิลเตอร์นั้น อาศัยหลักการแทนค่าความต้านทานของวงจร Active RC Filter ด้วยสวิทช์กับคาปาซิเตอร์ที่ต่อกันแบบ Bilinear ซึ่งหาความสัมพันธ์ระหว่าง analog frequency กับ digital frequency มีค่าใกล้เคียงกันมาก จากการทดลองได้ผลเป็นที่น่าพอใจและพบว่าความถี่ของสัญญาณนาฬิกาที่ใช้ควบคุมการทำงานของอนาลอกสวิทช์มีค่าประมาณ 20 - 30 เท่าของความถี่ cutoff จึงจะให้ผลของสัญญาณเข้าทุกออกมาราบเรียบที่สุด ซึ่งความถี่ cutoff ของวงจรจะแปรผันตรงกับความถี่ของสัญญาณนาฬิกาที่ใช้ควบคุมการทำงานของอนาลอกสวิทช์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] กนก เจนจิระพงศ์เวช. " วงจร Active filter ", เอกสารประกอบการสอนวิชา Electronic Laboratory I , คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง
- [2] วัลลภ สุระก่าพลธร, " การประมวลผลสัญญาณเชิงเลข " , วัฒนาพรินทร์, กรุงเทพฯ , 2533
- [3] Charles S.Williams , " Design Digital Filter " , by Prentice - Hall , Inc.USA , 1986.
- [4] David E.Johnson , " Introduction to Filter Theory " , Prentice - Hall , Inc.Englewood Cliffs , New Jersey , 1976.
- [5] G.C.Temes , " The Derivation of Switched Capacitor Filter from Active RC Prototype " , Electronic Letters , Vol.14, June 1978 , pp.361 - 362.
- [6] James A.Codzow , " Foundation of Digital Signal Processing and Data Analysis " , Macmillan Publishing company , 1987.
- [7] Phillip E.Allen , " Switched Capacitor Circuit " .
- [8] Robert D.Strum, Donal E.Kirk, " First Principle of Discrete System and Digital Signal Processing " , 1988.
- [9] Walter G.Jang , " IC-Op-Amp Cookbook " , 1974.
- [10] S.M.Bozic, " Digital and Kalman Filtering " , Edward Arnold Australia , 1979.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

รายงานฉบับนี้สำเร็จลงได้ก็เพราะได้รับความช่วยเหลือจากหลาย ๆ ท่าน
ที่จะขาดเสียมิได้ต้องขอขอบพระคุณท่านอาจารย์ปราโมทย์ วาดเขียน อาจารย์ที่ปรึกษา
ที่ได้ให้คำแนะนำมาโดยตลอด และขอขอบคุณเพื่อน ๆ ที่ได้ให้ความช่วยเหลือดังต่อไปนี้
คือ คุณวารินทร์ ปลื้มรุ่งโรจน์ , คุณรุ่งนภา ไทยประยูร , คุณกิตติเดช นิมวงศ์เจริญสุข
ภาคโทรคมนาคม รุ่น 29

และต้องขอขอบคุณเจ้าหน้าที่ห้องสมุดคณะวิศวกรรมศาสตร์, สโมสรนักศึกษา
คณะวิศวกรรมศาสตร์(ส่วนต่อเนือง) คุณสมศักดิ์ เอื้อนฤกุล(ภาคอิเล็กทรอนิกส์ - ต่อเนือง)
และคุณสมรรถ (ภาคคอมพิวเตอร์ - ต่อเนือง) ที่ได้ให้ความช่วยเหลือและอำนวยความสะดวก
ความสะดวก

และต้องขอขอบคุณน้อง ๆ และเพื่อน ๆ ช่างสมรรถพุทธ ๆ ทุก ๆ ท่านที่ได้
เป็นกำลังใจมาโดยตลอด

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้