สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การออกแบบวงจรกรองสัญญาณเชิงเลขด้วยเจเนติกอัลกอริทึม

DESIGN OF DIGITAL FILTER USING GENETIC ALGORITHM

โดย

นางสาวศริญญา ปะสะกวี

นางสาวสุพรรณิการ์ พัฒนวลัญช์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เลขหมู่.....
เลขทะเบียน 54962
วัน,เดือน,ปี - 1 มิ.ย. 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

การออกแบบวงจรกรองสัญญาณเชิงเลขด้วยเจเนติกอัลกอริทึม

DESIGN OF DIGITAL FILTER USING GENETIC ALGORITHM

โดย

นางสาวศริญญา ปะสะกวี 43010423

นางสาวสุพรรณิการ์ พัฒนวลัญช์ 43010486

อาจารย์ที่ปรึกษา

รศ.ดร. กอบชัย เดชหาญ

อ.สมปอง วิเศษพานิชกิจ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2546

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การออกแบบวงจรกรองสัญญาณเชิงเลขด้วยเจเนติก อัลกอริทึม

DESIGN OF DIGITAL FILTER USING GENETIC ALGORITHM

ผู้จัดทำ

1. นางสาวศริญญา ปะสะกวี 43010423
2. นางสาวสุพรรณิการ์ พัฒนวลัญช์ 43010486

..... อาจารย์ที่ปรึกษา

(รศ.ดร. กอบชัย เดชหาญ)

..... อาจารย์ที่ปรึกษา

(อ.สมปอง วิเศษพานิชกิจ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบวงจรกรองสัญญาณเชิงเลขด้วยเจเนติก อัลกอริทึม

DESIGN OF DIGITAL FILTER USING GENETIC ALGORITHM

โดย นางสาวศริญญา ปะสะกวี 43010423
นางสาวสุพรรณิการ์ พัฒนวลัญช์ 43010486

อาจารย์ที่ปรึกษา รศ.ดร. กอบชัย เดชหาญ
อ. สมปอง วิเศษพานิชกิจ

บทคัดย่อ

โครงงานนี้นำเสนอการออกแบบวงจรกรองสัญญาณเชิงเลขชนิดผลตอบสนองอิมพัลส์จำกัดด้วยวิธีการเจเนติกอัลกอริทึม ซึ่งเป็นวิธีการค้นหาคำตอบที่ดีที่สุดโดยใช้หลักการคัดเลือกแบบธรรมชาติ และหลักการทางสายพันธุ์ ซึ่งจัดว่าเป็นการคำนวณอย่างหนึ่งที่มีวิวัฒนาการอยู่ในขั้นตอนของการค้นหาคำตอบ ส่วนผลการทดสอบจะใช้บอร์ดประมวลผลสัญญาณเชิงเลข TMS320C31 สำหรับเปรียบเทียบผลที่ได้ทางทฤษฎี

Abstract

This project presents a design of finite impulse response (FIR) digital filter by genetic algorithm. This algorithm aims at the search of optimization based on natural selection and genetic operation which classified to evaluate the computation. Experiment is used TMS320C31 DSP board for comparing with theory.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	**1**
บทที่ 2 ทฤษฎีและหลักการของเจเนติก	**5**
2.1 เจเนติก อัลกอริทึม	9
2.1.1 เจเนติก อัลกอริทึมแบบพื้นฐาน	11
2.1.2 เจเนติก โอเปอเรเตอร์	17
2.1.2.1 การเลือก	17
2.1.2.2 ครอสโอเวอร์	19
2.1.2.3 มิวเตชั่น	23
2.1.3 ตัวอย่างของเจเนติก อัลกอริทึมแบบพื้นฐาน	24
2.1.4 การประยุกต์เจเนติก อัลกอริทึมแบบพื้นฐาน	26
บทที่ 3 ทฤษฎีและหลักการทางวงจรกรองความถี่ดิจิตอล	**34**
3.1 ทฤษฎีดิจิตอลซิกแนลโปรเซสซิ่ง	36
3.2 หลักการเบื้องต้นของตัวกรองความถี่ดิจิตอล	47
3.3 โครงสร้างของตัวกรองความถี่ดิจิตอล	48
3.4 ชนิดของตัวกรองความถี่ดิจิตอล	49
3.5 สรุปคุณสมบัติของตัวกรองความถี่แบบเอฟไออาร์	50
3.6 สรุปคุณสมบัติของตัวกรองความถี่แบบไอไอออาร์	51
บทที่ 4 การออกแบบและการทดลอง	**55**
บทที่ 5 ผลการทดลอง	**59**
บทที่ 6 บทวิจารณ์และบทสรุป	**89**
ภาคผนวก	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

หน้า

รูปที่ 1.2.1 ระบบที่ใช้ไมโครโปรเซสเซอร์ 2

รูปที่ 1.1.2 ส่วนประกอบของ DSP 3

รูปที่ 2.1.1 โครโมโซมทางพันธุศาสตร์ 5

รูปที่ 2.1.2 การแบ่งตัวของเซลทางพันธุศาสตร์ 6

รูปที่ 2.1.3 การครอสโอเวอร์ทางพันธุศาสตร์ 7

รูปที่ 2.1.4 การมิวเตชันทางพันธุศาสตร์ 8

รูปที่ 2.1.5 หลักการเบื้องต้นของ GA 10

รูปที่ 2.1.6 เจเนติกอัลกอริทึมแบบพื้นฐาน 11

รูปที่ 2.1.7 วัฏจักรเจเนติก อัลกอริทึม 13

รูปที่ 2.1.8 รายละเอียดทางพันธุศาสตร์ กับ เจเนติกอัลกอริทึม 15

รูปที่ 2.1.9 ไดอะแกรมการทำงานของ เจเนติก อัลกอริทึมแบบง่าย 16

รูปที่ 2.1.10 การเลือกโดยใช้วิธีวงล้อรูเล็ท 18

รูปที่ 2.1.11 ครอสโอเวอร์แบบ 1 จุด 20

รูปที่ 2.1.12 การครอสโอเวอร์แบบหนึ่งจุด 22

รูปที่ 2.1.13 ออร์เดอร์ครอสโอเวอร์ 22

รูปที่ 2.1.14 ไบนารีมิวเตชัน 23

รูปที่ 2.1.15 มิวเตชั่น 24

รูปที่ 2.1.16 รายละเอียดผลการทำงานของ SGA ในการหาค่าสูงสุดของฟังก์ชัน $y = x^n$ ที่ n = 10 27

รูปที่ 2.1.17 (ต่อ) รายละเอียดผลการทำงานของ SGA ในการหาค่าสูงสุดของ ฟังก์ชัน $y = x^n$ ที่ n = 10 28

รูปที่ 2.1.18 กราฟค่าความเหมาะสมสูงสุดของ SGA ในการหาค่าสูงสุดของฟังก์ชัน $y = x^n$ ที่ n = 10 29

รูปที่ 2.1.19 ครอสโอเวอร์แบบ 2 จุด 31

รูปที่ 2.1.20 อินเวอร์ชัน 32

รูปที่ 3.1 กราฟสัญญาณไม่ต่อเนื่อง 37

รูปที่ 3.2 ตัวอย่างของลำดับสัญญาณ ซึ่งมีความหมายในการวิเคราะห์ และการแทน ความหมายในระบบและสัญญาณที่ไม่ต่อเนื่องตามแกนเวลา 37

รูปที่ 3.3 การแปลงลำดับสัญญาณเข้า X (n) ไปเป็นสัญญาณออก Y (n) 38

รูปที่ 3.4 ความสัมพันธ์ระหว่าง s-plane และ z-plane 41

รูปที่ 3.5 สัญลักษณ์และบล็อกไดอะแกรมสำหรับวงจรดิจิตอล 42

รูปที่ 3.6 บล็อกไดอะแกรมสำหรับแทนสมการแตกต่างทั่ว ๆ ไป อันดับที่ N 42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

หน้า

รูปที่ 3.7 การประมาณวงจรกรองความถี่ในอุดมคติในช่วงกว้างที่กำหนดขอบเขตให้ 43

รูปที่ 3.8 การ mapping ของ bilinear transformation 47

รูปที่ 3.9 ความสัมพันธ์ในแกนความถี่ โดยวิธีไบลีเนียร์ 47

รูปที่ 3.10 บล็อคไดอะแกรมของ Transversal Filter โดยที่มี h(k),(k=0,1,2,…,n-1) 48

รูปที่ 3.11 การวิเคราะห์สมการ Difference Equation ในรูปแบบบล็อคไดอะแกรม 48

รูปที่ 3.12 การวิเคราะห์สมการ Difference Equation ในรูปแบบ Signal Flow Graph Diagram 49

รูปที่ 3.13 บล็อคไดอะแกรมแสดงถึงหลักการของตัวกรองความถี่ดิจิตอล 49

รูปที่ 3.14 โครงสร้างของตัวกรองความถี่ดิจิตอลแบบเอฟไอาร์ 51

รูปที่ 3.15 โครงสร้างของตัวกรองความถี่ดิจิตอลแบบไอไอาร์ 52

รูปที่ 4.1.1 ผลตอบสนองแอมปลิจูดของวงจรกรองสัญญาณความถี่ต่ำผ่านในอุดมคติ 56

รูปที่ 4.1.2 การจัดโครงสร้างของวงจรกรองสัญญาณดิจิตอลแบบเอฟไอาร์ 57

รูปที่ 4.1.3 โครงสร้างของวงจรกรองในอันดับที่ 5 โดยลดจำนวนตัวคูณลง 58

รูปที่ 5.1.1 กราฟผลตอบสนองทางความถี่ ของตัวกรองแบบเอฟไอาร์โดยใช้เจเนติก อัลกอริทึมในการหาค่าสัมประสิทธิ์ของตัวกรองความถี่และใช้ Matlab เขียนกราฟ 59

รูปที่ 5.1.2 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไอาร์ 59

รูปที่ 5.1.3 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไอาร์ 60

รูปที่ 5.1.4 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไอาร์ 60

รูปที่ 5.1.5 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไอาร์ เทียบกับสัญญาณอินพุทที่เป็นแบบสวีฟ 61

รูปที่ 5.1.6 ผลตอบสนองทางความถี่ ของสัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่าน แบบเอฟไอาร์ซึ่งใช้ไดนามิกซิกแนลอนาไลเซอร์ ในการบันทึกผล 62

รูปที่ 5.1.7 ผลตอบสนองทางเฟสของสัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่าน แบบเอฟไอาร์ซึ่งใช้ไดนามิกซิกแนลอนาไลเซอร์ ในการบันทึกผล 62

รูปที่ 5.1.8 การเปรียบเทียบกราฟผลตอบสนองทางความถี่ที่ได้จากวิธีเจเนติก อัลกอริทึม และกราฟผลตอบสนองทางความถี่ที่ได้จากการใช้วินโดว์ฟังก์ชัน 63

รูปที่ 5.2.1 กราฟผลตอบสนองทางความถี่ ของตัวกรองแบบเอฟไอาร์โดยใช้เจเนติก อัลกอริทึมในการหาค่าสัมประสิทธิ์ของตัวกรองความถี่และใช้ Matlab เขียนกราฟ 65

รูปที่ 5.2.2 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไอาร์ 65

รูปที่ 5.2.3 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไอาร์ 66

รูปที่ 5.2.4 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไอาร์ 66

รูปที่ 5.2.5 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไอาร์ เทียบกับสัญญาณอินพุทที่เป็นแบบสวีฟ 67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

หน้า

รูปที่ 5.2.6 ผลตอบสนองทางความถี่ ของสัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ซึ่งใช้ไดนามิกซิกแนลอนาไลเซอร์ ในการบันทึกผล 68

รูปที่ 5.2.7 ผลตอบสนองทางเฟสของสัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ซึ่งใช้ไดนามิกซิกแนลอนาไลเซอร์ ในการบันทึกผล 68

รูปที่ 5.2.8 การเปรียบเทียบกราฟผลตอบสนองทางความถี่ที่ได้จากวิธีเจเนติก อัลกอริทึม และกราฟผลตอบสนองทางความถี่ที่ได้จากการใช้วินโดว์ฟังก์ชัน 69

รูปที่ 5.3.1 กราฟผลตอบสนองทางความถี่ ของตัวกรองแบบเอฟไออาร์โดยใช้เจเนติกอัลกอริทึมในการหาค่าสัมประสิทธิ์ของตัวกรองความถี่และใช้ Matlab เขียนกราฟ 71

รูปที่ 5.3.2 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ 71

รูปที่ 5.3.3 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ 72

รูปที่ 5.3.4 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ 72

รูปที่ 5.3.5 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์เทียบกับสัญญาณอินพุทที่เป็นแบบสวีฟ 73

รูปที่ 5.3.6 ผลตอบสนองทางความถี่ ของสัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ซึ่งใช้ไดนามิกซิกแนลอนาไลเซอร์ ในการบันทึกผล 74

รูปที่ 5.3.7 ผลตอบสนองทางเฟสของสัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ซึ่งใช้ไดนามิกซิกแนลอนาไลเซอร์ ในการบันทึกผล 74

รูปที่ 5.3.8 การเปรียบเทียบกราฟผลตอบสนองทางความถี่ที่ได้จากวิธีเจเนติก อัลกอริทึม และกราฟผลตอบสนองทางความถี่ที่ได้จากการใช้วินโดว์ฟังก์ชัน 75

รูปที่ 5.4.1 กราฟผลตอบสนองทางความถี่ ของตัวกรองแบบเอฟไออาร์โดยใช้เจเนติกอัลกอริทึมในการหาค่าสัมประสิทธิ์ของตัวกรองความถี่และใช้ Matlab เขียนกราฟ 77

รูปที่ 5.4.2 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ 77

รูปที่ 5.4.3 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ 78

รูปที่ 5.4.4 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ 78

รูปที่ 5.4.5 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์เทียบกับสัญญาณอินพุทที่เป็นแบบสวีฟ 79

รูปที่ 5.4.6 ผลตอบสนองทางความถี่ ของสัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ซึ่งใช้ไดนามิกซิกแนลอนาไลเซอร์ ในการบันทึกผล 80

รูปที่ 5.4.7 ผลตอบสนองทางเฟสของสัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ซึ่งใช้ไดนามิกซิกแนลอนาไลเซอร์ ในการบันทึกผล 80

รูปที่ 5.4.8 การเปรียบเทียบกราฟผลตอบสนองทางความถี่ที่ได้จากวิธีเจเนติก อัลกอริทึม และกราฟผลตอบสนองทางความถี่ที่ได้จากการใช้วินโดว์ฟังก์ชัน 81

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

หน้า

รูปที่ 5.5.1 กราฟผลตอบสนองทางความถี่ ของตัวกรองแบบเอฟไอาร์โดยใช้เจเนติก อัลกอริทึมในการหาค่าสัมประสิทธิ์ของตัวกรองความถี่และใช้ Matlab เขียนกราฟ 83

รูปที่ 5.5.2 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไอาร์ 83

รูปที่ 5.5.3 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไอาร์ 84

รูปที่ 5.5.4 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไอาร์ 84

รูปที่ 5.5.5 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไอาร์ เทียบกับสัญญาณอินพุทที่เป็นแบบสวีฟ 85

รูปที่ 5.5.6 ผลตอบสนองทางความถี่ ของสัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่าน แบบเอฟไอาร์ซึ่งใช้ไดนามิกซิกแนลอนาไลเซอร์ ในการบันทึกผล 86

รูปที่ 5.5.7 ผลตอบสนองทางเฟสของสัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่าน แบบเอฟไอาร์ซึ่งใช้ไดนามิกซิกแนลอนาไลเซอร์ ในการบันทึกผล 86

รูปที่ 5.5.8 การเปรียบเทียบกราฟผลตอบสนองทางความถี่ที่ได้จากวิธีเจเนติก อัลกอริทึม และกราฟผลตอบสนองทางความถี่ที่ได้จากการใช้วินโดว์ฟังก์ชัน 87

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 2.1.1 คำศัพท์ทางพันธุศาสตร์กับทางเจเนติกอัลกอริทึม 15

ตารางที่ 2.1.2 ตัวอย่างของการเลือก 18

ตารางที่ 2.1.3 ตัวอย่างของการสุ่ม 19

ตารางที่ 2.1.4 ตัวอย่างของการครอสโอเวอร์ 21

ตารางที่ 2.1.5 ตัวอย่างของการมิวเตชั่นภายหลังจากการครอสโอเวอร์ 23

ตารางที่ 2.1.6 การกำหนดรูปโครโมโซมของ SGA 25

ตารางที่ 2.1.7 จำนวนประชากรแต่ละรุ่น 25

ตารางที่ 2.1.8 ค่าความเหมาะสมและรูปแบบของโครโมโซม 26

ตารางที่ 4.1.1 ผลตอบสนองอิมพัลส์ในอุดมคติของวงจรกรองชนิดต่างๆ 57

ตารางที่ 5.1.1 ขนาดของสัญญาณที่ความถี่ต่างๆของสัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำแบบเอฟไออาร์ 63

ตารางที่ 5.1.2 ค่าความเหมาะสมของค่าสัมประสิทธิ์ในแต่ละรุ่นของตัวกรองความถี่แบบเอฟไออาร์ที่ได้จากวิธีเจเนติกอัลกอริทึม 64

ตารางที่ 5.2.1 ขนาดของสัญญาณที่ความถี่ต่างๆของสัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำแบบเอฟไออาร์ 69

ตารางที่ 5.2.2 ค่าความเหมาะสมของค่าสัมประสิทธิ์ในแต่ละรุ่นของตัวกรองความถี่แบบเอฟไออาร์ที่ได้จากวิธีเจเนติกอัลกอริทึม 70

ตารางที่ 5.3.1 ขนาดของสัญญาณที่ความถี่ต่างๆของสัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำแบบเอฟไออาร์ 75

ตารางที่ 5.3.2 ค่าความเหมาะสมของค่าสัมประสิทธิ์ในแต่ละรุ่นของตัวกรองความถี่แบบเอฟไออาร์ที่ได้จากวิธีเจเนติกอัลกอริทึม 76

ตารางที่ 5.4.1 ขนาดของสัญญาณที่ความถี่ต่างๆของสัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำแบบเอฟไออาร์ 81

ตารางที่ 5.4.2 ค่าความเหมาะสมของค่าสัมประสิทธิ์ในแต่ละรุ่นของตัวกรองความถี่แบบเอฟไออาร์ที่ได้จากวิธีเจเนติกอัลกอริทึม 82

ตารางที่ 5.5.1 ขนาดของสัญญาณที่ความถี่ต่างๆของสัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำแบบเอฟไออาร์ 87

ตารางที่ 5.5.2 ค่าความเหมาะสมของค่าสัมประสิทธิ์ในแต่ละรุ่นของตัวกรองความถี่แบบเอฟไออาร์ที่ได้จากวิธีเจเนติกอัลกอริทึม 88

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1. Genetic Algorithms

ในการออกแบบวงจรกรองสัญญาณดิจิตอลโดยทั่วไปจะทำการออกแบบจากอนุกรมฟูเรียร์ (Fourier Series) จุดเด่นของวิธีการนี้คือ สามารถกำหนดความถี่ที่ต้องการเลยได้จากนั้นจึงทำการคำนวณหาค่าของ Impulse Response หรือค่าของสัมประสิทธิ์ของตัวกรองสัญญาณต่อไป หลักการง่าย ๆ ของการออกแบบก็คือเราจะกำหนดความถี่ที่เราต้องการก่อนจากนั้นก็คำนวณหาค่าของ Coefficient filters แต่ก็มีจุดที่ยุ่งยากขึ้นมาคือการแก้สมการหาค่าสัมประสิทธิ์ของตัวกรองสัญญาณนั้นเป็นการยากมากถ้าหากสมการอยู่ในรูปที่ยุ่งยาก แต่ในโครงงานนี้ได้เสนอการออกแบบวงจรกรองสัญญาณดิจิตอลโดยใช้แนวความคิดแบบเจเนติกอัลกอริทึมหรือเรียกโดยย่อว่า " GA " เพื่อหาสัมประสิทธิ์ของตัวกรองสัญญาณดิจิตอล โดยแนวความคิดแบบเจเนติกอัลกอริทึม (genetic algorithm) เป็นการจำลองเอาแนวความคิดของการวิวัฒนาการของสิ่งมีชีวิตในระบบชีววิทยามาใช้ในการคำนวณด้วยคอมพิวเตอร์ จากหลักการที่ว่าสิ่งมีชีวิตที่แข็งแรงมากกว่าจะมีโอกาสอยู่รอดและเป็นผู้ผลิตทายาทในรุ่นต่อไปได้มากกว่า ดังนั้นเจเนติกอัลกอริทึมจึงเป็นการค้นหาคำตอบที่ดีที่สุดโดยใช้หลักการคัดเลือกแบบธรรมชาติและหลักการทางสายพันธุ์ วิธีการนี้มีความสามารถในการค้นหาคำตอบอย่างชาญฉลาดและลดความยุ่งยากในขั้นตอนของการค้นหาของปัญหาที่มีรูปแบบซับซ้อน ปัญหาที่มีหลายคำตอบ หรือปัญหาที่ไม่สามารถหาอนุพันธ์ได้ วิธีเจเนติกอัลกอริทึมได้ถูกนำมาใช้งานอย่างกว้างขวางเนื่องจากมีข้อได้เปรียบและมีความแตกต่างไปจากวิธีดั้งเดิม เช่น การคำนวณเชิงเลข การโปรแกรมเชิงเส้น หรือวิธีการค้นหาผลเฉลย ปัจจุบันจึงเห็นได้ว่ามีการนำเอา GA ไปใช้ในเกือบจะทุกสาขาวิชา เช่น การประมวลผลสัญญาณดิจิตอล การประมวลผลสัญญาณภาพและการมองเห็นด้วยคอมพิวเตอร์ ระบบควบคุม การสื่อสารและโทรคมนาคม อิเล็กทรอนิกส์ ไฟฟ้ากำลัง คอมพิวเตอร์และอินเตอร์เน็ต การแพทย์ การเงิน และการขนส่ง และอื่น ๆ อีกมากมาย จะเห็นได้ว่าประสิทธิภาพและสมรรถนะของ GA ได้เป็นที่ยอมรับและได้มีการประยุกต์ใช้กันอย่างแพร่หลาย ในโครงงานนี้จึงขอนำเสนอรายละเอียดของ GA ในทุกองค์ประกอบเพื่อความเข้าใจในหลักการทำงานของ GA อันจะนำไปสู่การนำไปประยุกต์ใช้ในการออกแบบวงจรกรองสัญญาณดิจิตอลได้อย่างมีประสิทธิภาพ

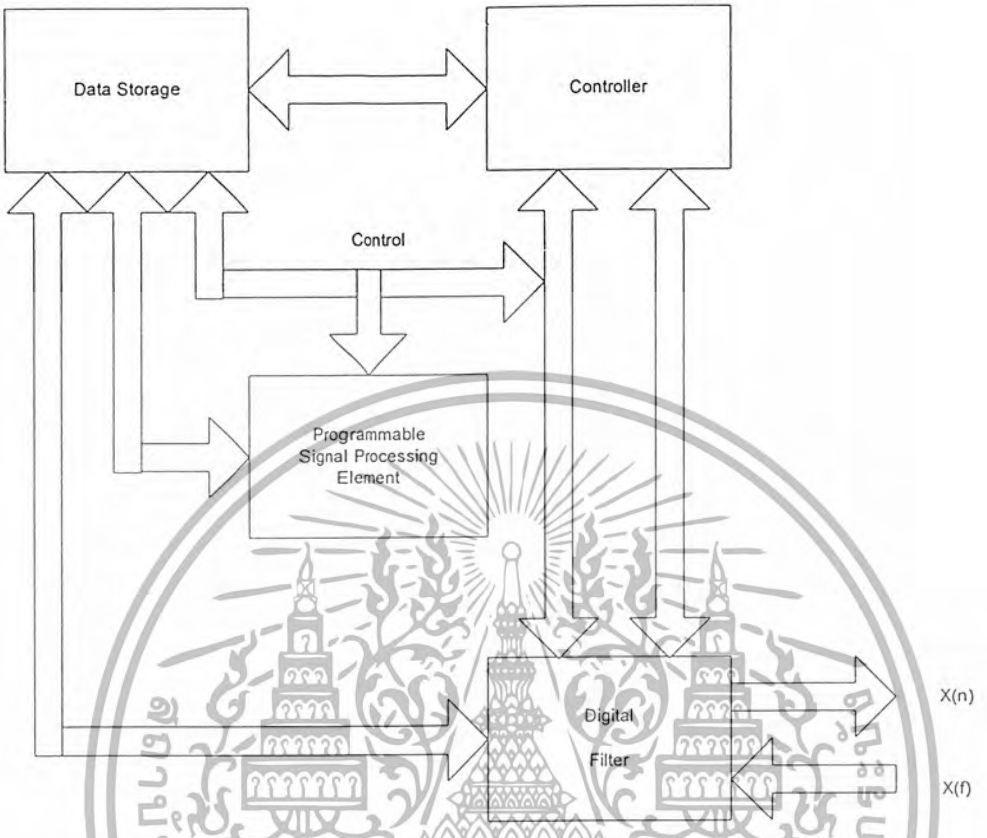
2. Digital Signal Processing

ดิจิตอลซิกแนลโปรเซสซิง (DSP) คือการเปลี่ยนรูปแบบของสัญญาณโดยการใช้การคำนวณทางคณิตศาสตร์ด้วยดิจิตอลซิกแนลโปรเซสเซอร์ ซึ่งจะเห็นได้ว่าเราสามารถจะนำสัญญาณนั้นใส่ลงไปในสมการคณิตศาสตร์ แล้วได้ผลลัพธ์ออกมาโดยตรงเลยทีเดียว

ในช่วง 10 กว่าปีที่ผ่านมา ความก้าวหน้าทางด้านดิจิตอลซิกแนลโปรเซสซิงได้เกิดขึ้นเป็นอย่างมากมีการค้นพบอัลกอริทึม (Algorithms) ใหม่ๆเพิ่มขึ้นเรื่อยๆ และการพัฒนาเครื่องมือใหม่ๆ เพื่อช่วยในการออกแบบตามอัลกอริทึมนั้นอีกด้วย ระบบดิจิตอลซิกแนลโปรเซสซิงนั้นจะมีอยู่ 2 แบบใหญ่ๆคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ระบบที่ออกมาเพื่อให้ทำงานอย่างใดอย่างหนึ่งโดยเฉพาะ
2. ระบบที่ใช้ไมโครโปรเซสเซอร์ ซึ่งจะสามารถทำงานเป็นอะไรก็ได้ขึ้นอยู่กับโปรแกรมที่ใส่



รูปที่ 1.2.1 แสดงระบบที่ใช้ไมโครโปรเซสเซอร์ ซึ่งจะสามารถทำงานเป็นอะไรก็ได้ขึ้นอยู่กับโปรแกรมที่ใส่

ความก้าวหน้าทางอิเล็กทรอนิกส์ครั้งใหญ่ เกิดขึ้นเมื่อมีการสร้างดิจิตอลซิกแนลโปรเซสเซอร์แบบความเร็วสูงขึ้น (High-Speed Signal Processor) ซึ่งปัจจุบันนี้ได้มีใช้กันอย่างแพร่หลายในราคาไม่แพงนัก

ดิจิตอลซิกแนลโปรเซสเซอร์จำเป็นจะต้องมีความเร็วสูงและถูกออกแบบมาโดยเฉพาะให้สามารถทำงานได้อย่างมีประสิทธิภาพ โดยใช้ข้อดีของการประมวลผลแบบขนาน และใช้ชุดคำสั่งที่สร้างขึ้นมาโดยเฉพาะ ทั้งนี้ก็เพื่อการประมวลผลแบบเวลาจริง (Real time) นั่นเอง การประมวลผลแบบเวลาจริงคือเมื่อเราใส่สัญญาณอินพุทเข้าไป ก็จะได้สัญญาณเอาท์พุทออกมาเลย โดยจะมีเวลาหน่วง (Delay) น้อยมาก ซึ่งจะมีประโยชน์ในการใช้งานมากกว่าการประมวลผลแบบไม่ใช่เวลาจริง

ด้วยความสามารถของดิจิตอลซิกแนลโปรเซสชิง จึงได้นำไปใช้ในรูปแบบต่างๆ ซึ่งให้ผลลัพธ์ที่ดีกว่าใช้วงจรอนาลอก เช่น ความน่าเชื่อถือ ความยืดหยุ่น การทดแทนได้ง่าย ขนาดที่เล็กกว่า และเสถียรภาพในระยะยาว และความสามารถของดิจิตอลซิกแนลโปรเซสเซอร์ที่สามารถเปลี่ยนโปรแกรมการทำงานได้ง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในระยะยาว และความสามารถของดิจิตอลซิกแนลโปรเซสเซอร์ที่สามารถเปลี่ยนโปรแกรมการทำงานได้ง่าย ทำให้การใช้งานดิจิตอลซิกแนลโปรเซสเซอร์นั้นน่าสนใจ เช่น สามารถเปลี่ยนให้ดีขึ้นเมื่ออัลกอริทึมใหม่ๆถูกค้นพบ หรือความสามารถทำงานได้หลายๆอย่างโดยเพียงแค่เปลี่ยนโปรแกรมให้กับมันเท่านั้น

ในทางอิเล็กทรอนิกส์ โดยส่วนใหญ่มักเกี่ยวข้องกับสัญญาณทางไฟฟ้า เช่นการกรองสิ่งที่ไม่ต้องการออกจากสัญญาณอินพุท การแยกเอาข้อมูลออกมา หรือไม่ก็เป็นการสร้างรูปคลื่นที่ต้องการขึ้นมา หรือการเปลี่ยนลักษณะทางแอมปลิจูดของสัญญาณ ซึ่งในอดีตการกระทำเหล่านี้จะทำโดยใช้วงจรอนาลอกที่ออกแบบมาเพื่อทำงานนั้นๆ นอกจากนั้นวงจรอนาลอกยังไม่เหมาะที่จะสร้างเป็นวงจรขนาดใหญ่ (VLSI) อีกด้วย คุณสมบัติของวงจรอนาลอกนั้นจะขึ้นอยู่กับค่าของอุปกรณ์ที่ใช้ เช่น R, C ที่มีค่าไม่แน่นอน ที่สำคัญจะมีกระแสรั่วไหลที่แปรตามเวลา และยังขึ้นอยู่กับโวลเตจและอุณหภูมิอีกด้วย การออกแบบวงจรทางอนาลอกนั้นก็ยังน่าเบื่อ เพราะวงจรหนึ่งๆจะทำงานได้อย่างเดียวเท่านั้น แต่ถ้าต้องการให้ทำงานอื่นๆ ก็ต้องเปลี่ยนอุปกรณ์และก็ต้องเดินสายกันใหม่ นอกจากนี้แล้วการทดสอบวงจรอนาลอกอาจเป็นปัญหาได้เนื่องจากอิมพิแดนซ์ของเครื่องมือที่วัดต่อเข้าไปในวงจร ปัญหาพื้นฐานของวงจรอนาลอกเหล่านี้ สามารถแก้ไขได้โดยใช้ DSP ข้อได้เปรียบอีกประการหนึ่งของการใช้ DSP แทนวงจรอนาลอกคือ การเปลี่ยนคุณสมบัติของการประมวลผลทำได้โดยง่าย ซึ่งจะทำให้เราสามารถทดลองในทุกทางที่เป็นไปได้เพื่อเลือกเอาผลที่ดีที่สุดที่เราพอใจ

โดยทั่วไประบบ DSP จะประกอบด้วยส่วนต่างๆ ดังรูป

รูปที่ 1.2.2 แสดงส่วนประกอบของ DSP

สัญญาณอนาลอกอินพุท จะถูกป้อนเข้าสู่ Low Pass Filter เพื่อจะจำกัดช่วงความถี่ของสัญญาณอินพุท จากนั้นก็จะส่งต่อไปยัง อนาลอก/ดิจิตอล คอนเวอร์เตอร์ (Analog to Digital Converter: A/D) แล้วส่งให้ระบบประมวลสัญญาณดิจิตอลต่อไป ระบบประมวลสัญญาณดิจิตอลก็จะนำเอาอัลกอริทึมที่มีอยู่มากระทำกับข้อมูล จากนั้นก็จะส่งสัญญาณข้อมูลที่ถูกแก้ไขแล้ว ไปให้ส่วนดิจิตอล/อนาลอก คอนเวอร์เตอร์ (Digital to Analog Converter:D/A) เพื่อเปลี่ยนสัญญาณข้อมูลกลับเป็นสัญญาณอนาลอกตามต้องการ แต่จากเอาท์พุทของ D/A ซึ่งจะมีลักษณะเป็นขั้นๆแยกกัน ดังนั้นเราจำเป็นต้องใช้วงจรกรองสัญญาณให้เรียบ (Low Pass Filter) เพื่อที่จะขจัดสัญญาณความถี่สูงที่ไม่ต้องการออกไป

หากเราจะมองข้อเสียของ DSP ก็มี แต่ข้อเสียหรือปัญหาที่เกิดขึ้นจะแตกต่างจากที่พบในวงจรอนาลอก เช่น การเกิดควอนไตเซชัน (Quantization), การเกิดน้อยส์ (Noise), Idle Channel Noise, ความไม่ราบเรียบของสัญญาณ, ความละเอียดของสัญญาณ เป็นต้น ตัวแปรที่สำคัญอีกตัวแปรหนึ่งที่จะแยกระบบ DSP ต่างๆออกตามการใช้งานคือ ค่าอัตราการสุ่มข้อมูล (Sampling Rate) ยิ่งอัตราการสุ่มข้อมูลมาก ยิ่งต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้ระบบประมวลผลที่มีความเร็วสูงยิ่งขึ้นด้วย อัตราการสุ่มข้อมูลนี้จะมีผลต่อการนำไปใช้งาน ถ้ามีอัตราการสุ่มข้อมูลที่สูง ก็จะใช้งานในย่านความถี่สูงขึ้นได้ โดยคุณภาพของสัญญาณข้อมูลยังคงยอมรับได้

บริษัทเทกซัสอินสตูเมนต์ ได้ผลิตไอซีที่เป็นตัวประมวลผลออกมาสำหรับใช้งานทางด้านของดิจิตอลซิกแนลโปรเซสซิงโดยเฉพาะ ซึ่งเป็นตระกูลที่มีชื่อว่า TMS320 ไอซีตระกูล TMS320 นี้จะเป็นลักษณะไมโครโปรเซสเซอร์แบบชิพเดี่ยว (Single Chip) สามารถทำงานได้กว้างขวาง และมีความเร็วในการประมวลผลสูง

เราจึงนำเอา TMS320C31 มาใช้ในงานวิเคราะห์สัญญาณแบบตัวกรองความถี่ โดยเราจะออกแบบให้สามารถเปลี่ยนค่าความถี่ผ่านและอื่นๆโดยคอมพิวเตอร์ ทำให้สามารถเลือกใช้และเปลี่ยนแปลงค่าความถี่ผ่านได้โดยง่ายทางคอมพิวเตอร์ ทั้งยังสามารถปรับปรุงและเปลี่ยนแปลงหรือพัฒนาโปรแกรมเพิ่มเติมได้อย่างง่ายดาย

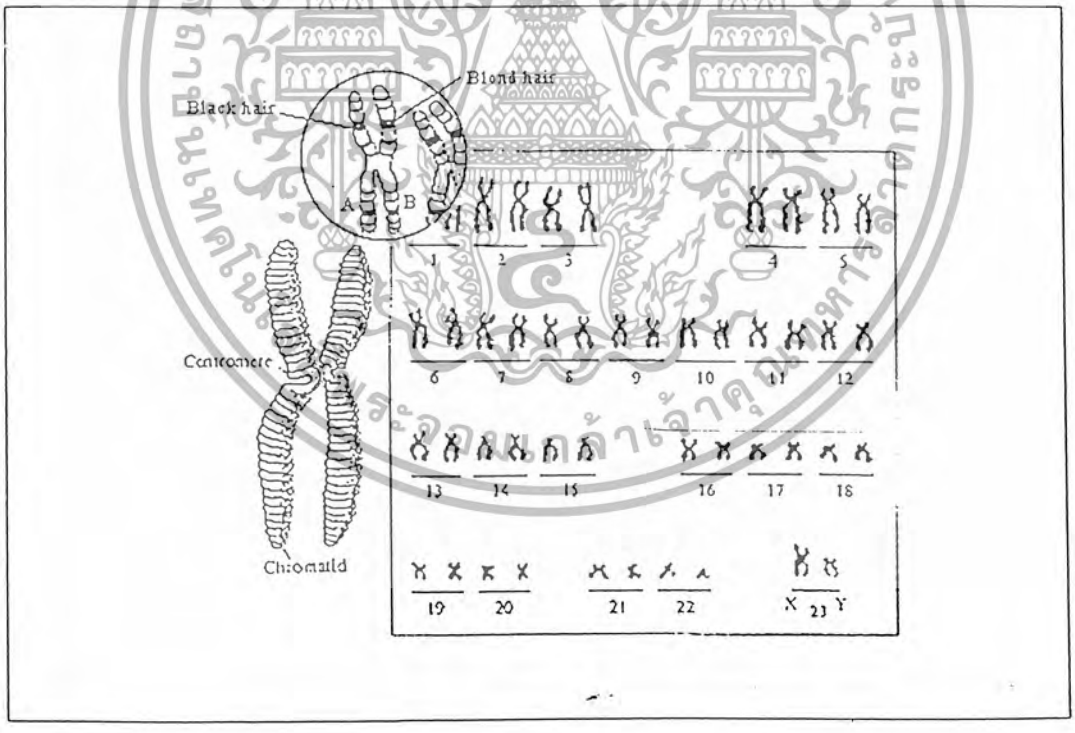
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่2
ทฤษฎีและหลักการของเจเนติก

ภาคทฤษฎีเจเนติก อัลกอริทึม

ในปัจจุบันนี้ปัญหาที่ต้องการคำตอบที่ดีที่สุด (Optimal Solution) ทางวิทยาศาสตร์ วิศวกรรมศาสตร์ คอมพิวเตอร์ หรือในการทำงานต่างๆที่เกิดขึ้นมากมายนั้น สามารถหาคำตอบได้หลายๆวิธี ซึ่งแตกต่างกันไปตามชนิดของปัญหา ความคิด เทคนิค วิธีการวิเคราะห์ปัญหานั้นๆและความแพร่หลายในการพัฒนาศักยภาพของคอมพิวเตอร์ให้รู้จักเรียนรู้เพื่อช่วยหาคำตอบหรือช่วยตัดสินคำตอบในขั้นต้นมีมากขึ้น โดยปัจจุบันนี้นักวิทยาศาสตร์ได้เริ่มนำความรู้เกี่ยวกับทฤษฎีหรือกฎเกณฑ์ทางธรรมชาติมาช่วยในการศึกษาวิจัย เช่นนิวรอลเน็ตเวิร์ค (Neural Network) ฟัซซี่ลอจิก (Fuzzy Logic) เป็นต้น เจเนติกอัลกอริทึมเป็นอีกวิธีการหนึ่งที่จำลองรูปแบบวิธีการทางชีววิทยา ในการให้กำเนิดประชากรรุ่นใหม่หรือขยายเผ่าพันธุ์ในรุ่นลูก รุ่นหลานต่อไป ซึ่งอาศัยพื้นฐานความคิดของวิวัฒนาการทางธรรมชาติถ่ายทอดลักษณะต่างๆทางพันธุกรรม โดยปฏิบัติตามกระบวนการทางพันธุศาสตร์ เพื่อจะใช้ในการหาคำตอบที่ดีที่สุดหรือใกล้เคียงที่สุดของปัญหาโดยคอมพิวเตอร์

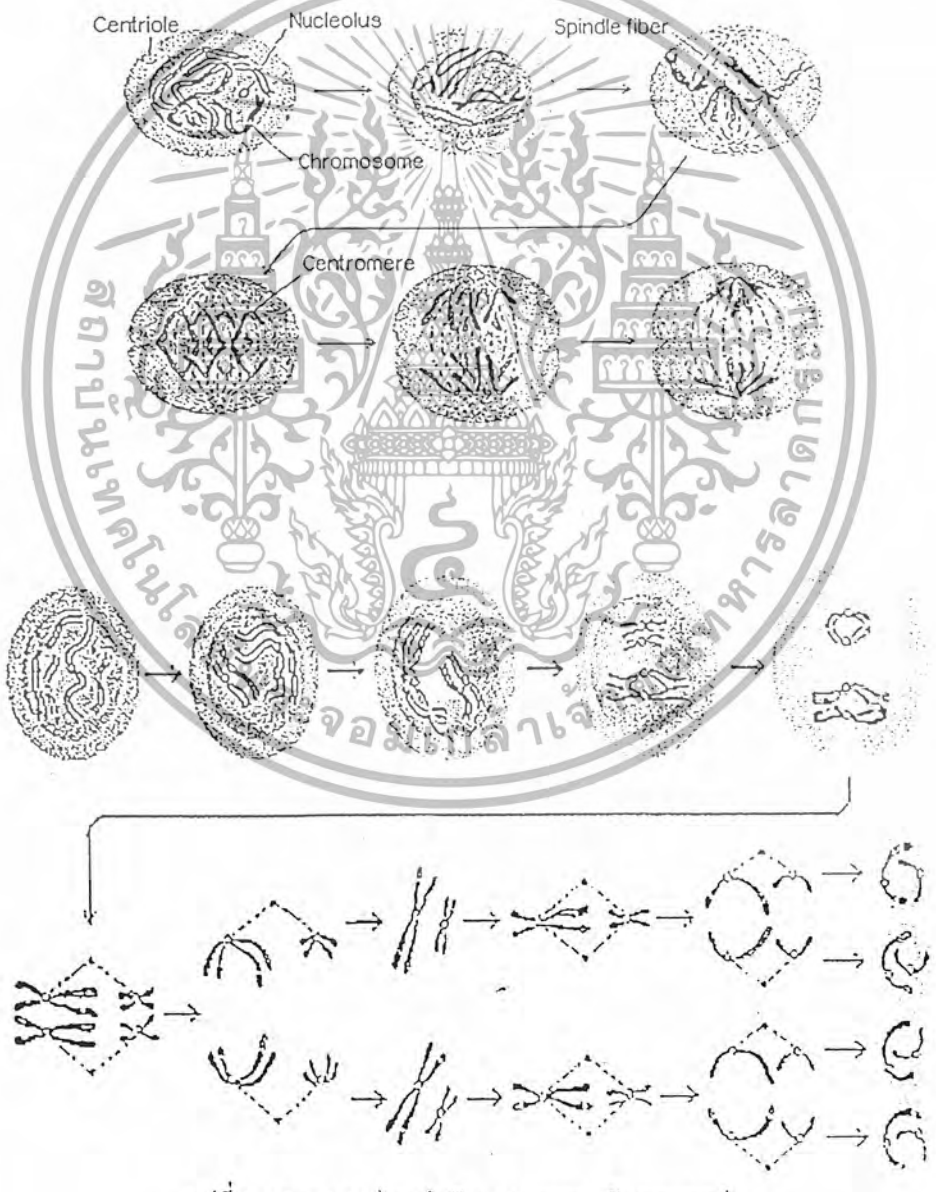
พันธุศาสตร์ทางชีววิทยา



รูปที่ 2.1.1 แสดงโครโมโซมทางพันธุศาสตร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมนเดล (Mendel) บิดาแห่งวิชาพันธุศาสตร์ ค้นพบว่า ยีนส์ (Genes) หน่วยเก็บลักษณะทางกรรมพันธุ์เป็นตัวกำหนดลักษณะภายนอก ซึ่งยีนหลายๆยีนจะเรียงตัวกันอยู่บนเส้นโครโมโซม (Chromosome) อีกทีหนึ่งในเซลของสิ่งมีชีวิต โดยจะอยู่กันเป็นคู่ๆ แต่จะแตกต่างกันที่ค่าลักษณะต่างๆในแต่ละยีน เรียกว่า แอลลี (Allele) ซึ่งแบบต่างๆของยีนที่มีแอลลีต่างกันในแต่ละตำแหน่งยีนเดียวกันเรียกว่า จีโนไทป์ (Genotype) สำหรับลักษณะภายนอกที่ปรากฏออกมาให้เห็น เรียกว่า ฟีโนไทป์ (Phenotype) เช่น ในคนจะมีโครโมโซม 23 คู่ 46 โครโมโซม ซึ่งแต่ละโครโมโซมจะประกอบด้วยยีนต่างๆกันราว 1250 ยีน ตัวอย่างคู่โครโมโซมที่ 1 ในคนดังรูปที่ 2.1.1 ซึ่งประกอบด้วยยีนลักษณะสีผม สีผิว สีตา และอื่นๆอีกประมาณ 1247 ลักษณะ และโครโมโซม 1A มีแอลลีของยีนลักษณะสีผมคือ ผมสีดำ ส่วนโครโมโซม 1B มีแอลลีของยีนลักษณะสีผมคือ ผมสีบลอนด์



รูปที่ 2.1.2 แสดงการแบ่งตัวของเซลทางพันธุศาสตร์

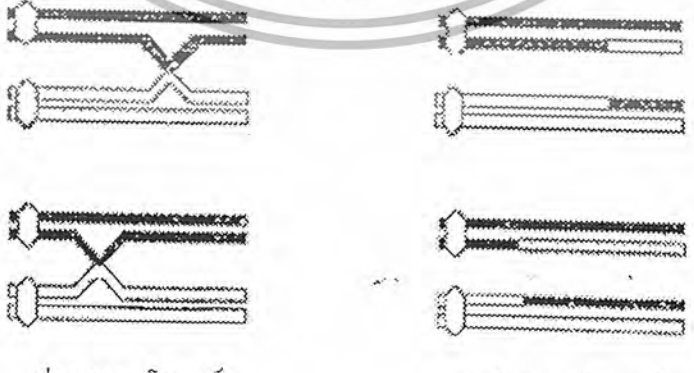
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การถ่ายทอดลักษณะทางพันธุกรรมเป็นการถ่ายทอดลักษณะต่างๆของสิ่งมีชีวิตที่เกิดขึ้นเมื่อมีการแบ่งตัวของเซลสิ่งมีชีวิต ซึ่งมี 2 แบบ คือ

(1) การเพิ่มจำนวนเซล เป็นการแบ่งตัวแบบไมโตสิส (Mitosis) โดยโครโมโซมแต่ละตัวจะขยายตัวเพิ่มจำนวนตัวเองขึ้นเป็นสอง และเยื่อหุ้มนิวเคลียส (Nucleus) จะสลายลงเพื่อดึงแยกโครโมโซมที่เพิ่มจำนวนขึ้นออกจากโครโมโซมเดิมเป็นสองด้าน แล้วเยื่อหุ้มนิวเคลียสจะถูกสร้างขึ้นใหม่กลายเป็นเซลใหม่ 2 เซล ที่มีโครโมโซมเหมือนเดิม ดังรูปที่ 2.1.2(a)

(2) การแบ่งตัวของเซลสืบพันธุ์ เป็นการแบ่งตัวแบบไมโอสิส (Meiosis) โดยโครโมโซมจากเซลพ่อ 1 โครโมโซมและโครโมโซมจากเซลแม่ 1 โครโมโซม จะเริ่มจับคู่กันที่โครโมโซมชนิดเดียวกัน ในขณะเดียวกันโครโมโซมแต่ละตัวทั้งที่มาจากพ่อและมาจากแม่ ต่างก็จำลองแบบของตนเพิ่มขึ้นมาอีกแต่ละโครโมโซม ทำให้ได้จำนวนโครโมโซมทั้งหมดเพิ่มขึ้นเป็นสองเท่า และดำเนินการทางพันธุกรรมจนถึงระยะแบ่งตัว โครโมโซมพ่อพร้อมกับแบบจำลองและโครโมโซมแม่พร้อมกับแบบจำลองที่ได้ จะแยกคู่ไปรวมกันเป็น 2 นิวเคลียส กลายเป็นเซลใหม่ 2 เซล ซึ่งจะแบ่งตัวต่อทันที โดยแต่ละโครโมโซมพ่อแยกตัวออกจากแบบจำลอง และโครโมโซมแม่ก็แยกตัวออกจากแบบจำลองรวมกันเป็นเซลใหม่ 4 เซล ดังรูปที่ 2.1.2(b)

วิธีการทางพันธุศาสตร์ในระหว่างที่เกิดการแบ่งตัวของไมโอสิสนั้น โครโมโซมจะมีโอกาสแลกเปลี่ยนส่วนบางส่วนซึ่งกันและกัน อันเป็นสาเหตุที่ทำให้เกิดปรากฏการณ์ที่เรียกว่า ครอสโอเวอร์ (Crossover) ของลักษณะต่างๆขึ้น ซึ่งการครอสโอเวอร์นั้นเกิดขึ้นขณะที่มีการจำลองแบบเพิ่มขึ้น และเกิดขึ้นระหว่างโครโมโซมพ่อกับโครโมโซมแม่ ไม่ใช่เกิดขึ้นระหว่างโครโมโซมพ่อกับแบบจำลอง ซึ่งการครอสโอเวอร์จะทำให้เกิดการแลกเปลี่ยนลักษณะของยีนต่างๆของคู่โครโมโซมพ่อกับแม่ โดยเนื่องจากยีนแต่ละยีนเรียงตัวกันบนเส้นโครโมโซมนั้นไม่ได้อยู่กันอย่างหนาแน่น แต่มีระยะห่างกันอย่างไม่สม่ำเสมอ เพราะคุณสมบัติของยีนนั้นเป็นโมเลกุลของสารโปรตีนประกอบตัวกันทางเคมี ช่องว่างระหว่างยีนนี่เองจะเป็นตำแหน่งที่แตกออกมาได้เวลาจะครอสโอเวอร์ และแลกเปลี่ยนยีนของโครโมโซมโดยส่วนที่อยู่หลังรอยแตก โดยจะถูกย้ายไปอยู่อีกโครโมโซมหนึ่งทั้งหมด นอกจากนี้ยังจะสามารถแตกอีกกี่แห่งก็ได้ ซึ่งผลนั้นขึ้นอยู่กับความสามารถที่จะเชื่อมกันมากน้อยเพียงไรของช่องว่างระหว่างยีนดังรูปที่ 2.1.3

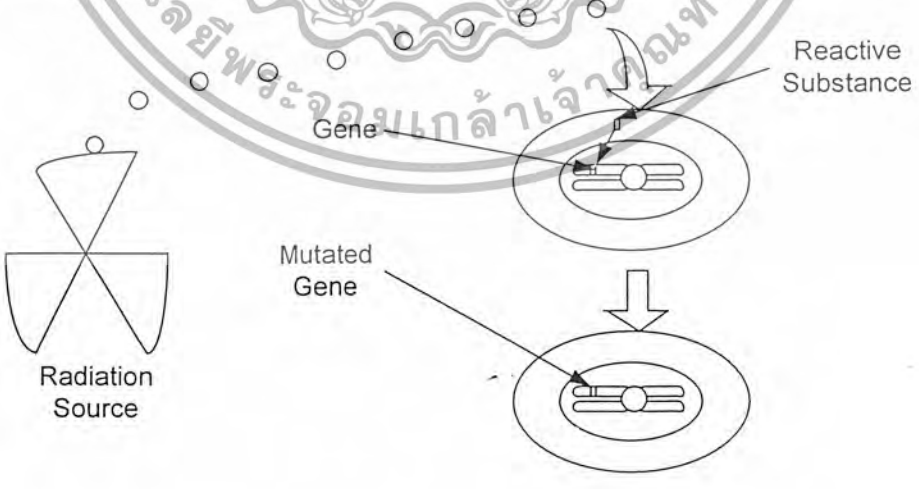


รูปที่ 2.1.3 แสดงการครอสโอเวอร์ทางพันธุศาสตร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประโยชน์ที่เกิดจากครอสโอเวอร์ที่เราจะเห็นได้คือ ทำให้มีโอกาสที่จะได้ลักษณะต่างๆกันมาอยู่รวมกันได้หลายแบบมากขึ้น ทำให้สิ่งมีชีวิตรุ่นลูกที่เกิดขึ้นมามีความหลากหลายมากขึ้น และอาจทำให้มีโอกาสเกิดสิ่งมีชีวิตที่มีลักษณะต่างๆที่ดีพอเหมาะรวมอยู่ด้วยกันได้อย่างพอดี เหมาะสมกับสิ่งแวดล้อม ถ้าการเกิดเซลใหม่โดยถ่ายทอดโครโมโซมไม่มีการครอสโอเวอร์แล้ว โครโมโซมใดเคยมียีนลักษณะใดก็จะมีลักษณะนั้นอยู่เรื่อยๆ รุ่นลูกก็จะมีลักษณะเช่นเดียวกัน โอกาสที่สิ่งมีชีวิตนั้นจะเจริญ หรือปรับตัวให้ดีขึ้นย่อมมีได้ยากกว่าการเปลี่ยนลักษณะยีนใหม่หลายๆแบบมากขึ้น

ลักษณะต่างๆของสิ่งมีชีวิตจะสามารถอยู่รอดได้โดยการคัดเลือกทางธรรมชาติคือ คัดเลือกโครโมโซมที่มีลักษณะที่ทำให้สิ่งมีชีวิตแข็งแรงเพียงพอ หรือเหมาะสมต่อสภาพแวดล้อม ซึ่งจะสามารถอยู่รอดและถ่ายทอดไปยังลูกหลาน ดังนั้นการคัดเลือกของธรรมชาติเพื่อถ่ายทอดลักษณะทางพันธุกรรมเป็นเพียงส่วนประกอบของการเปลี่ยนแปลงของสิ่งมีชีวิตเท่านั้นมิวเตชัน (Mutation) หรือการผ่าเหล่า คือการเปลี่ยนแปลงลักษณะของยีนไปจากเดิมที่ควรเป็นไปตามการถ่ายทอด ซึ่งเป็นต้นเหตุของการเกิดลักษณะที่แปลกๆออกไปมากมายสำหรับสิ่งมีชีวิตหนึ่งๆ ซึ่งเท่ากับเป็นการให้โอกาสแก่ธรรมชาติในการที่จะเลือกลักษณะแปลกๆมากขึ้น เนื่องจากกระบวนการวิวัฒนาการโดยธรรมชาติเองนั้นช้ามาก เพราะกว่าที่ธรรมชาติจะปรับสภาพแวดล้อมให้สิ่งมีชีวิตค่อยๆปรับตัวเองให้เหมาะสมนั้นมีโอกาสน้อยมาก การผ่าเหล่านั้นทุกลักษณะในแต่ละยีนย่อมมีโอกาสที่จะเกิดความเปลี่ยนแปลงไปจากเดิมได้พอๆกัน และถ้าเหมาะสมกับสภาพแวดล้อมขณะนั้นๆก็จะไม่ถูกคัดเลือกและหายไป ซึ่งข้อสรุปนี้ได้จากการทดลองโดยการนำเอายีนของแบคทีเรียมาผสมกัน และจัดสภาพแวดล้อมที่ไม่อำนวยต่อการผสมกันแล้ว ยีนจะปรับตัวเองเพื่อเร่งขบวนการผสมพันธุ์จนได้ผลดีในรุ่นหลังๆ ตัวอย่างของการผ่าเหล่าในอดีตคือการกำเนิดของปลาทองนั้น มีต้นกำเนิดมาจากการกลายพันธุ์หรือผ่าเหล่าของปลาชิว และในปัจจุบันลักษณะใหม่ที่เกิดจากการผ่าเหล่าก็ยังคงมีให้เห็นอยู่ เช่น ความสามารถของเชื้อโรคแบคทีเรียในการต้านทานต่อยาฆ่าเชื้อ หรือเซลผิดปกติอันเกิดจากถั่มมันตภาพรังสีต่างๆ ซึ่งมีผลต่อสารภายในเซลทำให้ลักษณะของยีนในเซลเปลี่ยนไป ซึ่งแสดงดังรูปที่ 2.1.4 เป็นต้น



รูปที่ 2.1.4 การมิวเตชันทางพันธุศาสตร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากความเชื่อในวิวัฒนาการทางธรรมชาติ ที่แสดงถึงคุณลักษณะที่เป็นอยู่ของสิ่งมีชีวิตโดยการถ่ายทอดลักษณะต่างๆบนโครโมโซมนั้นมีคุณสมบัติทั่วไปที่ยอมรับกันคือ

(1) วิวัฒนาการเป็นผลที่เกิดขึ้นเนื่องจากความเปลี่ยนแปลงบนโครโมโซมที่เป็นอยู่ ซึ่งแสดงลักษณะของสิ่งมีชีวิตนั้นๆ

(2) ธรรมชาติทางการคัดเลือกมีความสัมพันธ์กับโครโมโซมที่แสดงถึงประสิทธิภาพของโครงสร้างที่ดี ที่จะคัดเลือกเพื่อถ่ายทอดส่วนของโครงสร้างที่ดี

(3) การถ่ายทอดในขณะที่เกิดวิวัฒนาการนั้น โครโมโซมพ่อ-แม่ มีการแลกเปลี่ยนส่วนโครงสร้างกันเพื่อสร้างโครโมโซมลูก และเหตุผลที่ทำให้เกิดโครโมโซมลูกที่แตกต่างออกไปคือขบวนการผ่าเหล่า

(4) วิวัฒนาการทางธรรมชาติมิได้เป็นสิ่งที่เกิดจากความจดจำ แต่เป็นกระบวนการที่เกิดจากโครงสร้างต่างๆในโครโมโซมที่เหมาะสมกับสภาพแวดล้อมที่เกิดขึ้นในขณะนั้น

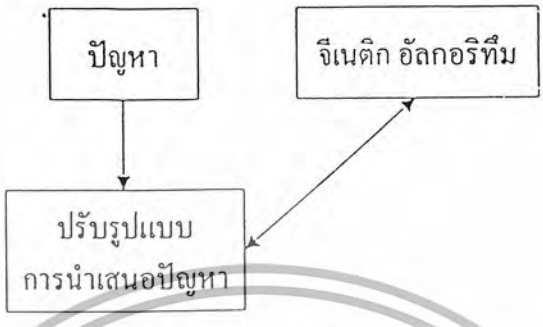
John Holland ในปี ค.ศ.1975 ร่วมกับเพื่อนและนักศึกษา ณ มหาวิทยาลัยของมิชิแกน คิดว่าแนวความคิดจากคุณสมบัติเหล่านี้ น่าจะนำมาปรับใช้กับคอมพิวเตอร์ให้ช่วยแก้ปัญหาที่ยุ่งยากต่างๆในการหาคำตอบที่ดีที่สุดหรือใกล้เคียงที่สุด เขาจึงได้ทำการวิจัยโดยจำลองแบบเพื่อทดลองกับปัญหาแบบต่างๆ โดยมีจุดมุ่งหมายที่จะศึกษาระบบปรับปรุงการประมวลผลเอง (Self Adaptive Process) และเพื่อสร้างโปรแกรมระบบผู้เชี่ยวชาญ (Artificial System Software) เพื่อแก้ปัญหา โดยอาศัยแนวความคิดของระบบทางธรรมชาติ และค้นพบวิธีการใหม่ซึ่งเรียกว่า เจเนติก อัลกอริทึม (Genetic Algorithms: A)

เจเนติก อัลกอริทึม

เจเนติกอัลกอริทึม เป็นอัลกอริทึมที่ค้นหาคำตอบ (Search Algorithm) ที่มีประสิทธิภาพสูง โดยอัลกอริทึมจะเลียนแบบวิวัฒนาการของสิ่งมีชีวิตในธรรมชาติในการหาคำตอบ จากหลักการที่ว่า สิ่งมีชีวิตที่แข็งแรงมากกว่าจะมีโอกาสอยู่รอดและเป็นผู้ผลิตทายาทในรุ่นต่อไปได้มากกว่า โดยในสิ่งมีชีวิต ภายในเซลจะมีส่วนที่เก็บลักษณะเฉพาะตัวของสิ่งมีชีวิตนั้นๆ ซึ่งเรียกว่า โครโมโซม (Chromosome) โดยโครโมโซมจะอยู่เป็นคู่ๆ ซึ่งจะมีอยู่หลายคู่ในแต่ละเซล ภายในโครโมโซมจะมีการเรียงตัวยาวของยีน (Gene) การเรียงตัวของยีนจะมีตำแหน่งเฉพาะสำหรับสิ่งมีชีวิตแต่ละชนิด โดยยีนจะเป็นตัวเก็บข้อมูลต่างๆของสิ่งมีชีวิต ซึ่งจะเป็นตัวถ่ายทอดลักษณะเด่นและด้อยไปสู่ลูกหลาน การผสมของยีนจะทำการ ครอสโอเวอร์ (Crossover) หรือ มิวเตชั่น (Mutation) ซึ่งยีนในโครโมโซมของสิ่งมีชีวิตลูกจะมีการสับเปลี่ยนตำแหน่งการเรียงตัวของยีนใหม่ การทำครอสโอเวอร์ คือการตัดบางส่วนของกลุ่มยีนในโครโมโซมของพ่อแม่แล้วผสมกัน ซึ่งจะได้โครโมโซมใหม่สองโครโมโซม ส่วนการทำมิวเตชั่น คือการเรียงสับเปลี่ยนตำแหน่งของยีนในโครโมโซมโดยการสุ่มเลือกตำแหน่งการสับเปลี่ยน ทำให้ได้การเรียงของยีนในโครโมโซมใหม่ที่อาจจะไม่เหมือนกับโครโมโซมของพ่อแม่ ดังนั้นประชากรของสิ่งมีชีวิตในรุ่นหลังจะไม่ค่อยได้รับยีนของประชากรรุ่นแรกๆที่ไม่ค่อยแข็งแรง ทำให้ประชากรในรุ่นหลังๆมีความแข็งแรงมากขึ้น ในปัจจุบันวิธีเจเนติกอัลกอริทึมได้นำมาใช้งานอย่างกว้างขวาง ในลักษณะการคำนวณหาค่าที่เหมาะสมและปัญหาการเรียนรู้ วิธีเจเนติกอัลกอริทึมประสบความสำเร็จอย่างสูงสำหรับการแก้ปัญหาที่มีรูปแบบที่ซ้ำซ้อน ปัญหาที่มีหลายคำตอบโล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอล(ที่ใกล้เคียงกับคำตอบโกลบอล) หรือปัญหาที่ไม่สามารถหาอนุพันธ์ได้ เมื่อเทียบกับเทคนิคการคำนวณที่มีพื้นฐานจากการหาอนุพันธ์และวิธีการไต่เขา(การหาคำตอบที่ดีขึ้นจากตำแหน่งคำตอบเดิม)



รูปที่ 2.1.5 แสดงหลักการเบื้องต้นของ GA

รูปที่ 2.1.5 แสดงหลักการเบื้องต้นในการใช้ GA แก้ปัญหา โดยจะต้องมีการปรับปรุงรูปแบบปัญหาในการนำเสนอ GA ในลักษณะที่เหมาะสม เพราะ GA เป็นวิธีการค้นหาคำตอบโดยอาศัยวิธีการเลียนแบบการคัดเลือกทางธรรมชาติ และธรรมชาติทางพันธุกรรมโดยการรวมกันหรือสลับเปลี่ยนตัวแปรต่างๆ อันเป็นองค์ประกอบโครงสร้างของปัญหาที่ให้คำตอบที่ต้องการ ซึ่งอาศัยหลักการสุ่ม เพื่อปรับปรุงความสามารถในการค้นหาคำตอบที่ดีขึ้น การค้นหาคำตอบจากรุ่นหนึ่งไปรุ่นถัดไปตามวิวัฒนาการทางธรรมชาตินั้น คำตอบในรุ่นใหม่เกิดขึ้นจากการสร้างความสัมพันธ์ของโครงสร้างต่างๆที่ประกอบด้วยค่าตัวแปรที่เหมาะสมดีในรุ่นก่อน ดังนั้นจึงทำให้ได้คำตอบที่ดีขึ้น

จุดเด่นของวิธีเจเนติก อัลกอริทึมที่เหนือกว่าวิธีการค้นหาแบบเดิม เช่น วิธีที่มีพื้นฐานบนการคำนวณแคลคูลัสทั้งวิธีตรงและวิธีอ้อม วิธีการค้นหาโดยการแจกแจงคำตอบ และวิธีการค้นหาโดยการสุ่ม คือ

1. วิธีเจเนติก อัลกอริทึม ทำงานโดยอาศัยการเข้ารหัสของค่าพารามิเตอร์ของปัญหา ซึ่งไม่ใช้การค้นหาคำตอบจาก ค่าพารามิเตอร์เหล่านั้นโดยตรง
2. วิธีเจเนติก อัลกอริทึม ค้นหาผลลัพธ์ในการคำนวณจากกลุ่มของคำตอบ ซึ่งไม่ใช้การค้นหาจากคำตอบเดียว
3. วิธีเจเนติก อัลกอริทึม ใช้เฉพาะค่าความเหมาะสมจากฟังก์ชั่นเป้าหมาย มาใช้เป็นข้อมูลสำหรับการค้นหาคำตอบ ซึ่งไม่จำเป็นต้องใช้ความรู้ในการหาค่าอนุพันธ์ หรือความรู้ในทางคณิตศาสตร์อื่นๆ
4. วิธีเจเนติก อัลกอริทึม ค้นหาผลลัพธ์จากวิธีการจากความน่าจะเป็น ซึ่งไม่ใช้กรรมวิธีการคำนวณโดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 เจเนติก อัลกอริทึมแบบพื้นฐาน (Simple Genetic Algorithm: SGA)

เจเนติก อัลกอริทึมแบบพื้นฐาน จะมีลำดับในการทำงานดังรูปที่ 2.1.6 โดยเริ่มต้นจากการสุ่มประชากร (Population) เริ่มต้นแล้วหาค่าแต่ละตัวของประชากร ทำการเลือกประชากร N_p ตัวเพื่อสร้างประชากรรุ่นใหม่ โดยการเลือกประชากรนั้นจะมีหลายวิธี เช่น อาจจะเลือกโดยขึ้นอยู่กับอัตราส่วนของค่าฟิตเนส (Fitness) จากนั้นสุ่มจับคู่ประชากรที่เลือกเป็นประชากรรุ่นพ่อแม่เพื่อสร้างประชากรรุ่นลูกหลานโดยทำการครอสโอเวอร์ สำหรับการที่จะนำประชากรมาทำการครอสโอเวอร์นั้นจะขึ้นอยู่กับค่า P_c ซึ่งคือค่า Probability ของการทำ ครอสโอเวอร์ ถ้าประชากรตัวไหนที่ไม่ได้ทำการครอสโอเวอร์ประชากรตัวนั้นจะถูกแทนเป็นประชากรรุ่นถัดไปโดยอัตโนมัติ จากนั้นจะนำประชากรไปทำการมิวเตชัน ตามค่า P_m ซึ่งคือค่า Probability ของการมิวเตชัน จากนั้นประชากรทุกตัวจะเป็นตัวแทนรุ่นต่อไป

1. Initialization
   สุ่มประชากรรุ่นแรก
2. Selection
   2.1 ใช้การสุ่มเลือกประชากร เช่น อาจใช้การเลือกแบบ Roulette Wheel แล้วแทนที่ลงในประชากรปัจจุบัน
   2.2 สุ่มจับคู่ประชากรที่ได้
3. Crossover
   พิจารณาอัตราการเกิดครอสโอเวอร์ว่าคู่ของโครโมโซมจากขั้นตอนที่ 2 ควรเกิดครอสโอเวอร์หรือไม่ ถ้าควรเกิดการเกิดครอสโอเวอร์ให้สุ่มตำแหน่งที่จะทำการเกิดครอสโอเวอร์แล้วสร้างโครโมโซมคู่ใหม่แทนที่โครโมโซมที่เป็นพ่อแม่ ถ้าไม่ควรเกิดครอสโอเวอร์ให้คงโครโมโซมคู่เดิมไว้
4. Mutation
   สำหรับทุกโครโมโซมให้พิจารณาตามอัตราการเกิดมูเตชั่นว่าควรเกิดมูเตชั่นหรือไม่ ถ้าควรเกิด ให้สุ่มตำแหน่งที่จะเกิดการเปลี่ยนแปลง
5. Reproduction
   เป็นการสร้างประชากรรุ่นใหม่โดยทำตามขั้นตอนที่ 2-4 ซ้ำจนกระทั่งได้ผลตามเงื่อนไข

รูปที่ 2.1.6 เจเนติกอัลกอริทึมแบบพื้นฐาน

ฟังก์ชันเป้าหมายกับฟังก์ชันความเหมาะสม

การหาคำตอบที่ดีที่สุดของปัญหาของ GA มีพื้นฐานอยู่บนผลลัพธ์จากการหาคำตอบที่ผ่านมา วิธีการของ GA จะไม่พิจารณาจากขั้นตอนของการแก้ปัญหา แต่จะพิจารณาโดยตัดสินว่าคำตอบใหม่ที่ได้รับดีขึ้นหรือไม่ หรือเป็นคำตอบที่ใกล้เคียงคำตอบที่ต้องการหรือไม่ จากฟังก์ชันเป้าหมาย (Object Function :f) เนื่องจากแต่ละปัญหาจะสามารถกำหนดฟังก์ชันเป้าหมายซึ่งเป็นฟังก์ชันที่แสดงความสัมพันธ์ของแต่ละตัวแปร พารามิเตอร์ เงื่อนไข หรือข้อกำหนดต่างๆของปัญหานั้นๆที่ระบุคำตอบใดคำตอบหนึ่งที่สามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นไปได้ ณ ค่าพารามิเตอร์ เงื่อนไข หรือข้อกำหนดชุดดังกล่าว สำหรับฟังก์ชันความเหมาะสม (Fitness Function :F) เป็นฟังก์ชันที่กำหนดค่าความเหมาะสมของแต่ละโครโมโซมเปรียบเสมือนค่าความสามารถในการอยู่รอดของแต่ละโครโมโซมและเป็นฟังก์ชันที่กำหนดโอกาส หรือสัดส่วนที่แต่ละโครโมโซมเหมาะสมจะถูกคัดเลือกมากน้อยเพียงใด นั่นคือฟังก์ชันความเหมาะสมจะเป็นฟังก์ชันที่แสดงถึงค่าคำตอบที่เกิดขึ้นจากชุดตัวแปรของปัญหาของโครโมโซมนั้นดีเพียงใด โดยทั่วไปแล้วเรามักใช้ฟังก์ชันเป้าหมายเป็นฟังก์ชันความเหมาะสม หรืออาจใช้ฟังก์ชันเป้าหมายที่ถูกปรับให้เหมาะสมกับการนำเสนอ GA เป็นฟังก์ชันความเหมาะสมก็ได้

รูปแบบโครโมโซม

เราทราบกันแล้วว่าวิวัฒนาการทางธรรมชาติชีววิทยานั้นเป็นความเปลี่ยนแปลงต่างๆของสิ่งมีชีวิตเกิดขึ้นในโครโมโซม ดังนั้นจุดเริ่มต้นของการจำลองแบบทางธรรมชาติของ GA เพื่อใช้แก้ปัญหาจึงเริ่มจากการมองปัญหาเทียบเท่ากับโครโมโซมชนิดหนึ่ง ประกอบด้วยยีนลักษณะต่างๆ ซึ่งหมายถึงลำดับข้อมูลต่างๆที่จะแปลความหมายแล้วให้ค่าคำตอบของปัญหาค่าหนึ่ง การมองภาพยีนของ GA ให้ถือเสมือนยีนทางพันธุกรรมที่แสดงความหมายหรือเป็นตัวแทนคำตอบใดคำตอบหนึ่ง หรือลักษณะใดลักษณะหนึ่งทางกรรมพันธุ์ ในทางพันธุศาสตร์นั้นยีนเป็นตัวแสดงลักษณะที่อยู่รอดในสภาพแวดล้อมขณะนั้น สำหรับ GA นั้นยีนเป็นตัวแสดงค่าคำตอบของปัญหาที่แปรผันไปตามการประยุกต์ใช้งาน ซึ่งโดยทั่วไปยีนหมายถึงตัวแปร พารามิเตอร์ เงื่อนไข หรือข้อกำหนดต่างๆที่เป็นองค์ประกอบของปัญหา ดังนั้นการกำหนดรูปแบบโครโมโซมของแต่ละปัญหาโดยการแปลงตัวแปร พารามิเตอร์ เงื่อนไข หรือข้อกำหนดต่างๆให้อยู่ในรูปลำดับของยีนบนโครโมโซมหรือเรียกว่าสตริง (String) อันประกอบด้วยบิต (Bit) หรือเรียกว่าอักขระ (Character) ซึ่งลักษณะต่างๆที่เป็นได้ของแต่ละยีนคือค่าของบิต (Bit Value) หรือค่าตัวแปร พารามิเตอร์ต่างๆที่เป็นไปได้ และรูปแบบของค่าบิตที่จัดเรียงบนโครโมโซม คือ จีโนไทป์ (Genotype) ที่จะแสดงถึงค่าของตัวแปร พารามิเตอร์ต่างๆที่เป็นไปได้ชุดหนึ่ง หรือฟีโนไทป์ (Phenotype) นั่นเอง การกำหนดรูปแบบโครโมโซมของปัญหาให้เป็นตามแบบธรรมชาติ โดยกำหนดรหัสในรูปแบบตัวเลขหรือตัวอักษรในช่วงที่จำกัดตามค่าตัวแปรหรือพารามิเตอร์ และประกอบรวมกันในจำนวนยีนหรือความยาวของโครโมโซมที่คงที่ เช่น หากต้องการหาค่าสูงสุดของฟังก์ชันทางคณิตศาสตร์ $y=x^2$ ที่ x เป็นจำนวนเต็มอยู่ในช่วง [0,31] แล้ววิธีการของ GA ในการแก้ปัญหาโดยกำหนดรูปแบบโครโมโซมจากการกำหนดรหัสตัวแปร x เป็นตัวเลขไบนารี 0 หรือ 1 จำนวน 5 ตำแหน่ง ซึ่ง x จะมีค่าตั้งแต่ 00000 ถึง 11111 เป็นค่า 0 ถึง 31 ตามต้องการ เป็นต้น

วัฏจักรเจเนติก อัลกอริทึม

เมื่อกำหนดรูปแบบโครโมโซมและฟังก์ชันความเหมาะสมของปัญหาแล้ว GA จะสามารถประมวลผลหาคำตอบของปัญหาได้ โดยสร้างวิวัฒนาการกลุ่มคำตอบในรุ่นต่อไปตามวัฏจักรการทำงานของ GA (Genetic Algorithm Cycle) ดังรูปที่ 2.1.7 ซึ่งมี 4 ขั้นตอน คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(1) สร้างประชากรโครโมโซมรุ่นเก่าตามรูปแบบโครโมโซมที่กำหนดไว้ โดยประชากรต้นกำเนิด (Initial Population) เกิดจากการสร้างชุดโครโมโซมต้นกำเนิด จากการสุ่มสร้างค่าแต่ละบิทของแต่ละโครโมโซม

(2) วิเคราะห์ค่าความเหมาะสมแต่ละโครโมโซม โดยถอดรหัสค่าตัวแปร พารามิเตอร์ต่างๆของแต่ละบิทในโครโมโซม และคำนวณค่าความเหมาะสมจากฟังก์ชันความเหมาะสมที่กำหนดไว้

(3) สร้าง mating pool คือชุดโครโมโซมต้นแบบหรือชุดโครโมโซมพ่อ-แม่ ที่สามารถอยู่รอดเป็นต้นแบบ ซึ่งอาศัยการจำลองการคัดเลือกตามธรรมชาติ โดยพิจารณาถ่วงน้ำหนักจากค่าความเหมาะสมของแต่ละโครโมโซม หากโครโมโซมใดมีค่าความเหมาะสมมากก็จะมีโอกาสถูกคัดเลือกเป็นต้นแบบมาก

(4) ดำเนินการทางพันธุศาสตร์โดยสุ่มจับคู่โครโมโซมต้นแบบใน mating pool เพื่อสร้างประชากรโครโมโซมรุ่นใหม่ ซึ่งตัวดำเนินการทางพันธุศาสตร์ประกอบด้วย ครอสโอเวอร์ โดยการแลกเปลี่ยนค่าบิทบางส่วนของโครโมโซมซึ่งกันและกัน หรือมิวเตชัน โดยสุ่มเปลี่ยนค่าบิทบางบิทของแต่ละโครโมโซม เป็นต้น

รูปที่ 2.1.7 แสดงวัฏจักรเจเนติก อัลกอริทึม

การค้นหาคำตอบของ GA จะประมวลผลซ้ำตามวัฏจักร GA จนกว่าจะได้รับคำตอบที่พอใจตามกฎเกณฑ์ที่ตั้งไว้ หรือในระยะเวลาตามจำนวนรุ่นที่ดำเนินการที่ต้องการ ซึ่งแสดงอัลกอริทึมการทำงานของ GA ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

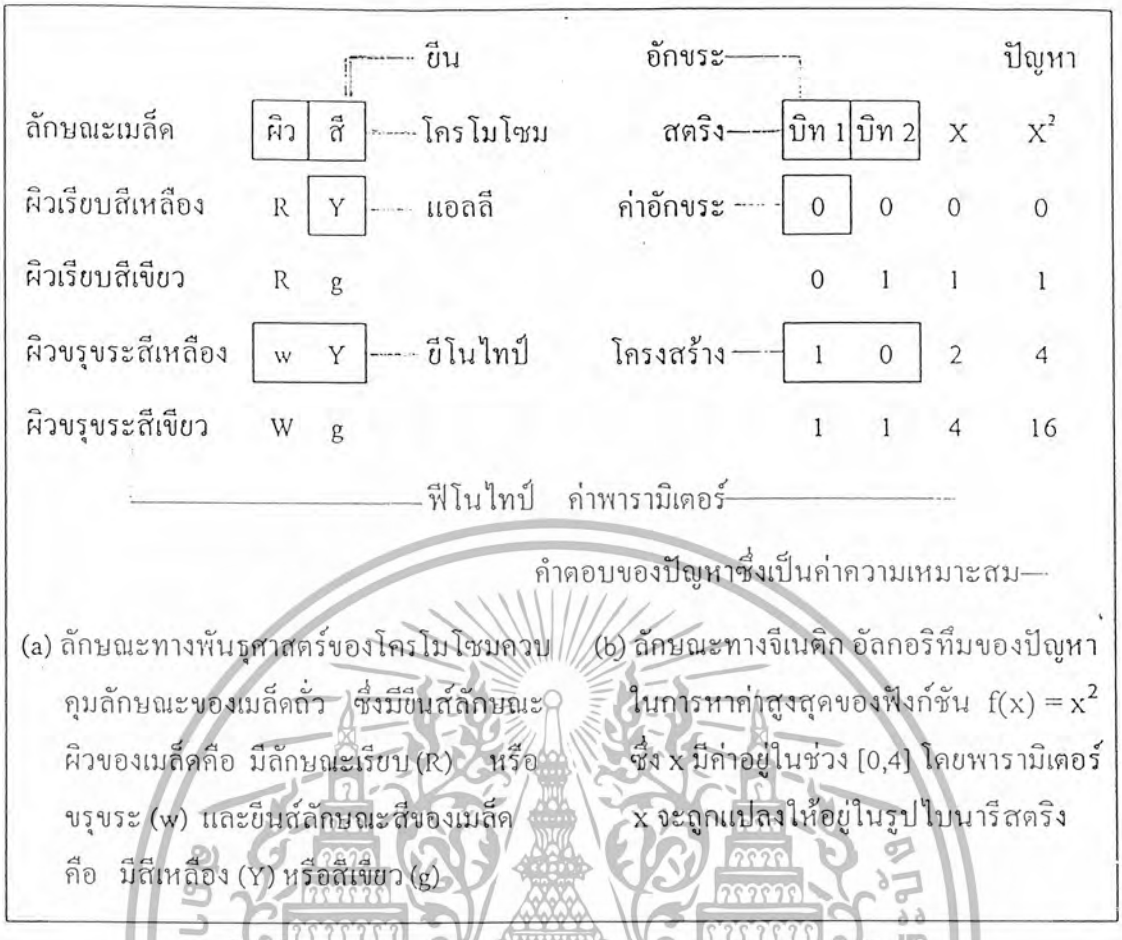
อัลกอริทึม GA

```
BEGIN
    t := 0;
    // สร้างประชากรโครโมโซมต้นกำเนิดโดยการสุ่ม
    Initial population P(t);
    // วิเคราะห์ค่าความเหมาะสมแต่ละโครโมโซมประชากรต้นกำเนิด
    Evaluate P(t);
    ตรวจสอบเงื่อนไขความพอใจ (เช่น เวลา, ค่าความเหมาะสม เป็นต้น)
    while not terminate
    begin
        t := t+1;
        // คัดเลือกโครโมโซมต้นแบบจากประชากรรุ่นก่อน
        P'(t) := Select parents P(t-1);
        // แลกเปลี่ยนส่วนยีนภายในโครโมโซมต้นแบบ
        Recombine P'(t);
        // มิวเตชันโครโมโซมต้นแบบ
        Mutate P'(t);
        // วิเคราะห์ค่าความเหมาะสมของประชากรรุ่นใหม่
        Evaluate P'(t);
        // ประชากรรุ่นใหม่กลายเป็นประชากรรุ่นเก่าต่อไป
        P(t) := P'(t);
    end;
END.
```

พันธุศาสตร์ทางชีววิทยา กับ เจเนติค อัลกอริทึม

เพื่อเปรียบเทียบลักษณะโครงสร้างทางพันธุศาสตร์ กับ เจเนติค อัลกอริทึม แล้วเรากล่าวโดยสรุปได้คือ ในทางพันธุศาสตร์ แต่ละโครโมโซม ประกอบด้วย หน่วยเก็บลักษณะ หรือยีน ซึ่งเก็บค่าแสดงลักษณะ หรือแอลลี และแต่ละแบบของชุดยีน เรียกว่าจีโนไทป์ ซึ่งแสดงลักษณะภายนอกที่ปรากฏเรียกว่า ฟีโนไทป์ ดังรูปที่ 2.1.8(a)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1.8 แสดงรายละเอียดทางพันธุศาสตร์ กับ เจเนติกอัลกอริทึม

สำหรับในทางเจเนติกอัลกอริทึม ตัวแปรหรือพารามิเตอร์ของปัญหาจะถูกแปลงให้อยู่ในรูปของสตริง ซึ่งมักเรียกกันว่า โครโมโซม ประกอบด้วย อักขระ หรือค่าของบิต ที่แสดงถึงโครงสร้างของแต่ละโครโมโซมที่มีค่าตัวแปรหรือพารามิเตอร์ของปัญหาแตกต่างกัน และเป็นตัวกำหนดค่าความเหมาะสมตามฟังก์ชันความเหมาะสมของแต่ละปัญหา ดังรูปที่ 2.1.8(b) ซึ่งสรุปความหมายเปรียบเทียบคำศัพท์ ที่ใช้ทางพันธุศาสตร์กับทางเจเนติก อัลกอริทึม ดังตารางที่ 2.1.1

Natural Genetic	Genetic Algorithm
chromosome	string
gene	Character , bit
allele	string position
genotype	structure
phenotype	a decode structure

ตารางที่ 2.1.1 แสดงคำศัพท์ทางพันธุศาสตร์ กับทางเจเนติก อัลกอริทึม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เจเนติก อัลกอริทึมแบบง่าย

รูปที่ 2.1.9 แสดงไดอะแกรมการทำงานของ เจเนติก อัลกอริทึมแบบง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

2.1.2 เจเนติก โอเปอเรเตอร์ (Genetic Operator)

เราจะกล่าวถึง เจเนติก โอเปอเรเตอร์ ของเจเนติก อัลกอริทึมแบบพื้นฐาน เช่น การเลือก (Selection),ครอสโอเวอร์ และมิวเตชั่น

2.1.2.1 การเลือก (Selection)

เป็นขั้นตอนที่จำลองแบบการคัดเลือกทางธรรมชาติเพื่อสร้าง Mating pool โดยคัดเลือกชุดโครโมโซมรุ่นเก่าให้เป็นโครโมโซมต้นแบบหรือโครโมโซมพ่อ-แม่ เพื่อใช้สร้างโครโมโซมลูกเป็นรุ่นต่อไป

• การเลือก จะมีหลายวิธีการ เช่น Roulette Wheel Selection, Binary Tournament Selection เป็นต้น สำหรับการคัดเลือกของ SGA เป็นแบบอ้างอิงค่าความเหมาะสม (Fitness-based Selection) โดยพิจารณาค่าความเหมาะสมเป็นตัวตัดสินว่า โครโมโซมใดในรุ่นเก่ามีโอกาสจะถูกเลือกเป็นโครโมโซมพ่อ-แม่มากน้อยเพียงใด โครโมโซมที่มีค่าความเหมาะสมที่ดีจะถูกกำหนดน้ำหนักค่าความน่าจะเป็นที่จะถูกเลือกแต่ละครั้งสูง การกำหนดค่าความน่าจะเป็นที่จะถูกเลือกต่อการสุ่มเลือกในแต่ละครั้ง (Probability of Selected Value : pselect) ของแต่ละโครโมโซมโดยกำหนดจากค่าความเหมาะสม เทียบกับผลรวมของค่าความเหมาะสมทั้งหมด ดังสมการที่ 2.1

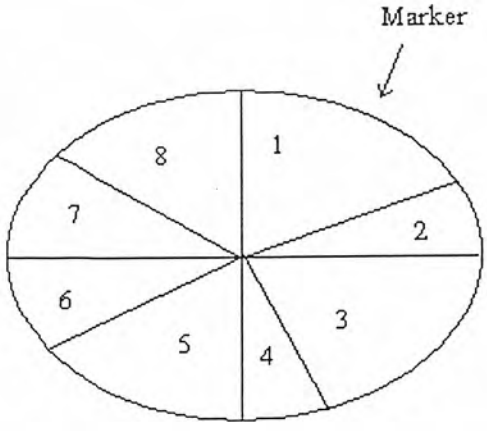
$$pselect_i = \frac{F_i}{\sum F} \qquad (2.1)$$

ซึ่งสามารถคำนวณค่าที่คาดหวังว่าจะสุ่มได้ (Expected Value: E) ของแต่ละโครโมโซมในแต่ละรุ่น ดังสมการที่ 2.2

$$E_i = pselect_i * popsize = \frac{F_i}{F} \qquad (2.2)$$

• Roulette Wheel Selection ดังรูปที่ 2.1.10 วงล้อจะเป็นตัวแทนของประชากรทุกตัวสำหรับ Slot ของวงล้อคือประชากรแต่ละตัว โดยขนาดของ Slot จะเป็นอัตราส่วนกับค่าความเหมาะสมของประชากรแต่ละตัว กรรมวิธีในการเลือกจะใช้วิธีการหมุนวงล้อ ถ้า Maker ตรงกับ Slot ของประชากรตัวนั้นจะได้รับการคัดเลือก ค่าProbability ของการที่จะถูกเลือกนั้นขึ้นกับอัตราส่วนของค่าความเหมาะสม สำหรับ Binary Tournament Selection ประชากร 2 ตัวจะถูกเลือกโดยการสุ่มและค่าของประชากรที่มีค่าดีกว่า 1 ตัวจะถูกเลือก ถ้า Binary Tournament Selection ถูกกระทำโดยไม่ได้แทนที่ของประชากร แล้วประชากร 2 ตัวจะถูกลบออกจากกลุ่มของประชากรทุกๆครั้งที่มีการเลือก โดยจำนวนของประชากรจะมีขนาดคงที่จากรุ่นหนึ่งไปยังอีกรุ่นหนึ่ง ประชากรเริ่มต้น (Original Population) จะถูกเรียกนำกลับคืนมา หลังจากประชากรใหม่ถูกแทนที่ครึ่งหนึ่ง ดังนั้นประชากรตัวที่ดีที่สุดอาจจะถูกเลือกถึง 2 ครั้ง และประชากรตัวที่แย่ที่สุดจะไม่ถูกเลือกเลยก็ได้ สำหรับ Binary Tournament Selection กับการแทนที่ของประชากร ค่าประชากร 2 ตัวจะถูกแทนที่ในประชากรรุ่นเดิมเพื่อสำหรับการเลือกครั้งต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
54962



รูปที่ 2.1.10 Roulette Wheel Selection

สำหรับวิธีการสุ่มโครโมโซมต้นแบบของ SGA เป็นแบบจำลองการหมุนวงล้อถ่วงน้ำหนัก (Roulette Wheel: RW) ซึ่งกำหนดขนาดในแต่ละช่องของวงล้อนั้นตามค่าความน่าจะเป็นที่จะสุ่มได้ในแต่ละครั้งของแต่ละโครโมโซม ซึ่งมีวิธีการดังนี้

(1) หาค่าความเหมาะสมของแต่ละโครโมโซม

(2) หาค่าความน่าจะเป็นที่จะสุ่มได้ในแต่ละครั้งของแต่ละโครโมโซม

(3) หาค่าความถี่สะสม (q) ของค่าความน่าจะเป็นของแต่ละโครโมโซม ดังสมการที่ 2.3

$$q_i = \sum_{j=1}^{i} pselect_j \qquad (2.3)$$

(4) สร้างเลขสุ่มจำนวนจริง (r) มีค่าอยู่ในช่วง [0.0, 1.0]

(5) เลือกโครโมโซมลำดับที่ i ซึ่ง r มีค่าอยู่ระหว่าง q_{i-1} และ q_i

ตัวอย่าง :

ลำดับ	โครโมโซม	x	ค่าความเหมาะสม (F)	ค่าความน่าจะเป็น ($pselect_i$)	จำนวนที่คาดหวัง (E_i)	จำนวนที่สุ่มได้จาก RW
1	01110	14	196	0.157	0.628	1
2	11001	25	625	0.502	2.008	2
3	01000	8	64	0.051	0.204	0
4	10011	19	361	0.290	1.160	1
	รวม		1246	1.000	4.000	
	ค่าเฉลี่ย		312	0.250	1.000	
	ค่าสูงสุด		625	0.502	2.008	

ตารางที่ 2.1.2 แสดงตัวอย่างของการเลือก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการกำหนดค่าความน่าจะเป็น โดยกำหนดจากค่าความเหมาะสม เทียบกับผลรวมของค่าความเหมาะสมทั้งหมด จะเห็นได้ว่าในการคัดเลือกโครโมโซมต้นแบบจาก 4 โครโมโซมนี้ โอกาสที่จะสุ่มได้โครโมโซมลำดับที่ 1 ต่อการสุ่มแต่ละครั้งเท่ากับ 0.157 และโอกาสที่จะสุ่มได้โครโมโซมลำดับที่ 2, 3, 4 ต่อการสุ่มแต่ละครั้งเท่ากับ 0.502, 0.051 และ 0.290 ตามลำดับ และจำนวนโครโมโซมต้นแบบที่สุ่มได้โดยจำลองการหมุนวงล้อ ดังนี้

ลำดับโครโมโซม	1	2	3	4
ค่าความเหมาะสม (F)	196	625	64	361
ค่าความน่าจะเป็นที่สุ่มได้ในแต่ละครั้ง ($pselect_i$)	0.157	0.502	0.051	0.290
ความถี่สะสมค่าความน่าจะเป็น (q_i)	0.157	0.659	0.710	1.000
สร้างเลขสุ่มในการหมุนวงล้อแต่ละครั้ง (r)	0.333	0.844	0.456	0.128
ลำดับโครโมโซมที่ถูกเลือก ($q_{i-1} \leq r \leq q_i$)	2	4	2	1

ตารางที่ 2.1.3 แสดงตัวอย่างของการสุ่ม

ซึ่งจำนวนที่สุ่มได้เป็นโครโมโซมต้นแบบใน Mating pool ของแต่ละโครโมโซมเป็น 1, 2, 0 และ 1 ตามลำดับ จะเห็นได้ว่าโครโมโซมลำดับที่ 2 มีค่าความเหมาะสมสูงที่สุด จะมีโอกาสถูกคัดเลือกในจำนวนที่มากที่สุด ส่วนโครโมโซมลำดับที่ 3 มีค่าความเหมาะสมต่ำมากจึงมีโอกาสที่จะไม่ถูกคัดเลือกเลย

วัตถุประสงค์ของเจเนติกอัลกอริทึม คือการลู่เข้าสู่ค่าที่เป็นคำตอบที่ Optimal โดยการเลือกจะเป็นตัวที่กำหนดอัตราของการลู่เข้าสู่ค่าประชากรมีค่า Optimal การที่มีอัตราการเลือกสูงจะนำไปสู่การที่ประชากรลู่เข้าสู่ค่า Suboptimal

Roulette Wheel Selection จะมีการเน้นการเลือกแบบอัตราการเลือกสูงในรุ่นต้นๆของประชากร โดยเฉพาะเมื่อประชากรมีค่าความเหมาะสมสูงกว่าประชากรตัวอื่นๆ สำหรับ Tournament Selection จะเน้นในประชากรรุ่นหลังๆ โดยเมื่อค่าความเหมาะสมของประชากรไม่แตกต่างกันมากนัก ดังนั้น Roulette Wheel Selection จะลู่เข้าสู่ Suboptimal ถ้าประชากรมีค่าความเหมาะสมที่แตกต่างกันมาก

2.1.2.2 ครอสโอเวอร์ (Crossover)

เป็นขั้นตอนที่จำลองแบบธรรมชาติทางพันธุกรรมขั้นตอนหนึ่งโดยการแลกเปลี่ยนส่วนของโครโมโซมพ่อ – แม่ตามการกำหนดอัตราความน่าจะเป็นของการครอสโอเวอร์ (Probability of Crossover :P_c) เพื่อสร้างชุดโครโมโซมรุ่นใหม่หรือโครโมโซมลูก การครอสโอเวอร์จะสร้างประชากรรุ่นลูกหลาน 2 ตัว โดยการครอสโอเวอร์จะมีหลายแบบ เช่น ครอสโอเวอร์ 1 จุด หรือ 2 จุด โครโมโซมจะถูกเลือกตำแหน่งโดยการสุ่มระหว่างความยาวของโครโมโซม และโครโมโซมของพ่อแม่จะถูกตัดที่จุดตำแหน่งที่สุ่ม จากรูปที่ 2.1.11 โครโมโซมรุ่นลูกตัวแรกจะเหมือนกับโครโมโซมพ่อแม่ตัวแรกจนถึงจุดตัดหลังจากจุดตัดนั้นจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

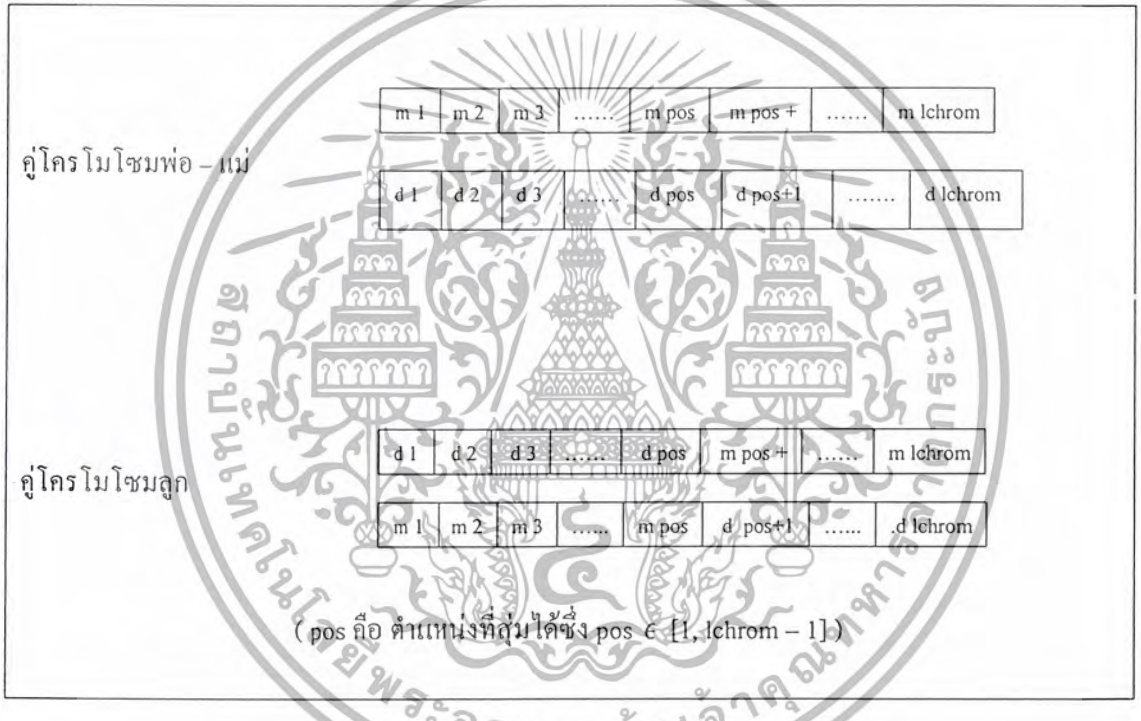
เหมือนกับโครโมโซมของลูกตัวที่สอง ใน Uniform Crossover จุดตัดของโครโมโซมจะถูกตัดตามค่า Probability ขั้นตอนการทำงานของครอสโอเวอร์ มีดังนี้

ขั้นตอนแรก : สุ่มจับคู่โครโมโซมพ่อ – แม่ใน mating pool ที่สร้างไว้จากการคัดเลือก

ขั้นตอนที่สอง : สร้างเลขสุ่มจำนวนจริง (r) มีค่าอยู่ในช่วง [0.0,1.0] โดยถ้า $r \leqslant P_c$ แล้ว โครโมโซมพ่อ – แม่นั้นจึงมีการครอสโอเวอร์

ขั้นตอนที่สาม : ครอสโอเวอร์โดยแลกเปลี่ยนส่วนของคู่โครโมโซมพ่อ – แม่นั้น ซึ่งการครอสโอเวอร์ของ SGA เป็นการครอสโอเวอร์แบบ 1 จุด (One – point Crossover) แสดงดังรูปที่ 2.1.11 ดังนี้

- สุ่มเลือกตำแหน่ง pos เป็นตำแหน่งที่จะครอสโอเวอร์ ซึ่ง pos มีค่าอยู่ในช่วง [1, lchrom – 1]
- แลกเปลี่ยนค่าในแต่ละบิทของคู่โครโมโซมพ่อ – แม่ตั้งแต่ตำแหน่งที่ pos+1 ถึง lchrom ซึ่งจะทำให้เกิดโครโมโซมลูกใหม่ 2 โครโมโซม



รูปที่ 2.1.11 ครอสโอเวอร์แบบ 1 จุด

จำนวนการครอสโอเวอร์ในแต่ละรุ่นดำเนินการขึ้นอยู่กับการกำหนดค่า P_c ซึ่งแตกต่างกันในแต่ละปัญหา เช่น ถ้าจำนวนประชากรในแต่ละรุ่น popsize เท่ากับ 30 โครโมโซม และกำหนดให้ $P_c = 0.6$ แล้วจำนวนการครอสโอเวอร์ในแต่ละรุ่นเท่ากับ P_c *(popsize / 2) = 0.6 * (30 / 2) = 9 ครั้ง (การครอสโอเวอร์ 1 ครั้ง เกิดจากโครโมโซม 2 โครโมโซม)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง : กำหนด $P_c = 0.5$ โครโมโซมพ่อ – แม่ ใน mating pool จากการคัดเลือกครอสโอเวอร์ ดังนี้

ลำดับที่คัดเลือก	mating pool	สุ่มจับคู่พ่อ – แม่	เลขสุ่ม (r)	ก่อนครอสโอเวอร์	สุ่มตำแหน่ง (pos)	หลังครอสโอเวอร์	X	ค่าความเหมาะสม (F)	ลำดับโครโมโซมลูก
2	11001	1,2	0.321	0 1 1 1 0	2	0 1 0 0 1	9	81	1
4	10011		≤ 0.5	1 1 0 0 1		1 1 1 1 0	30	900	2
2	11001	2,4	0.654	ไม่ครอสโอเวอร์			25	625	3
1	01110		>0.5				19	361	4
				รวม				1967	
				ค่าเฉลี่ย				492	
				ค่าสูงสุด				900	

ตารางที่ 2.1.4 แสดงตัวอย่างของการครอสโอเวอร์

จากการสุ่มจับคู่โครโมโซมพ่อ – แม่ ใน mating pool ได้โครโมโซมลำดับที่ 1 คู่ลำดับที่ 2 และลำดับที่ 2 คู่ลำดับที่ 4 แต่เฉพาะโครโมโซมคู่แรกจะเกิดครอสโอเวอร์ เนื่องจากเลขสุ่ม $r \leq 0.5$ ตามอัตราการครอสโอเวอร์ที่กำหนด โดยตำแหน่งในการครอสโอเวอร์ที่สุ่มได้คือ pos = 2 จะเห็นได้ว่าโครโมโซมลูกลำดับที่ 2 ที่เกิดขึ้นหลังจากครอสโอเวอร์มีค่าความเหมาะสมดีขึ้นกว่าโครโมโซมพ่อ – แม่ทั้งหมดในรุ่นก่อน เป็น 900 ซึ่งแสดงให้เห็นถึงการจำลองแบบกระบวนการครอสโอเวอร์ตามธรรมชาติทางพันธุศาสตร์ของ SGA ช่วยสร้างคำตอบที่ดีขึ้น

เนื่องจากการครอสโอเวอร์เป็นกระบวนการแบบสุ่ม ซึ่งผลของกระบวนการจะรวมโครโมโซมพ่อแม่ที่ไม่ดี ซึ่งจะส่งผลไปสู่โครโมโซมลูกหลานที่ไม่ดีด้วย แต่จะถูกกำจัดโดยโอเปอเรเตอร์ของการเลือก ในรุ่นถัดไปจำนวนของครอสโอเวอร์จะถูกควบคุมโดย Crossover Probability (P_c) ซึ่งกำหนดโดยอัตราส่วนของจำนวนของประชากรรุ่นลูกที่จะถูกสร้างในแต่ละรุ่นต่อจำนวนของประชากรทั้งรุ่น ค่า Crossover Probability สูงๆ จะทำให้หาค่าที่ดีใน Solution space และลดโอกาสที่จะได้รับค่าที่ไม่ดีได้ สำหรับค่า Crossover Probability ต่ำๆจะทำให้ สามารถใช้ประโยชน์จากประชากรที่มีค่าความเหมาะสมสูงๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Parent1:  1 0 1 1 0 1 1 0 1  |  1 1 1 0 0 1 1 0 0
Parent2:  0 0 1 1 0 1 1 0 0  |  1 0 0 1 0 1 0 0 0

                             |

Offspring1:  1 0 1 1 0 1 1 0 1  |  1 0 0 1 0 1 0 0 0
Offspring2:  0 0 1 1 0 1 1 0 0  |  1 1 1 0 0 1 1 0 0
```

รูปที่ 2.1.12 การครอสโอเวอร์แบบหนึ่งจุด

นอกจากนี้ยังมีการครอสโอเวอร์หลายแบบ เช่น ออร์เดอร์ครอสโอเวอร์ (Order Crossover), ไซเคิล ครอสโอเวอร์ (Cycle Crossover), พาร์เชียล แมป ครอสโอเวอร์ (Partially-Mapped Crossover) ซึ่งได้มีการประยุกต์กับการแก้ปัญหาหลายๆแบบ ในที่นี้จะยกตัวอย่างออร์เดอร์ ครอสโอเวอร์ ดังรูปที่ 2.1.13 สมมติว่าโครโมโซมแรกเรียงลำดับดังนี้ ABCDEFGHI และโครโมโซมตัวที่สองคือ DEBAHGFIC ก่อนที่จะสลับตำแหน่ง เราจะกำหนดตำแหน่งของกลุ่มโครโมโซมในยีนทั้งสองตำแหน่งที่กำหนดจะแบ่งยีนแต่ละอันออกเป็น 2 กลุ่ม โครโมโซมย่อยตำแหน่งที่กำหนดนี้เป็นแบบสุ่ม ไม่มีความแน่นอน จากรูปที่ 2.1.13 แสดงตัวอย่างการแบ่งยีนทั้งสองออกเป็น 4 กลุ่มโครโมโซม เส้นที่ลากผ่านยีนทั้งสองเป็นตัวกำหนดตำแหน่งการแบ่งกลุ่ม ยีนอันใหม่ของโครโมโซมลูกตัวแรกได้จากการเอากลุ่มของโครโมโซมทางซ้ายของเส้นตั้งที่แบ่งกลุ่มในยีนอันแรกมาใส่ในยีนอันใหม่ กลุ่มโครโมโซมทางซ้ายของเส้นแบ่งกลุ่มในยีนอันแรกคือ ABCDEF จากนั้นเราจะพิจารณาโครโมโซมแต่ละอันจากซ้ายไปขวาในยีนอันที่ 2 เราเอาโครโมโซมที่ยังไม่ได้ปรากฏในยีนอันที่ 3 มาใส่ในยีนอันที่ 3 ตามลำดับทีละตัวจากซ้ายไปขวา โครโมโซมแรกที่ไม่ได้อยู่ในกลุ่ม ABCDEF คือ 1 เราเอา 1 มาใส่ต่อจาก F จากนั้นเอา G และ H มาใส่ต่อจาก 1 เราข้ามโครโมโซม A, C, D ในยีนอันที่ 2 เพราะว่าปรากฏอยู่แล้วในยีนอันที่ 3 เราข้ามไปจนถึงโครโมโซม K และ J หยิบ K และ J ในยีนอันที่ 2 มาใส่ในยีนอันที่ 3 ถัดจาก H ตามลำดับ สำหรับโครโมโซมลูกตัวที่สองก็เกิดจากการสร้างเช่นเดียวกัน

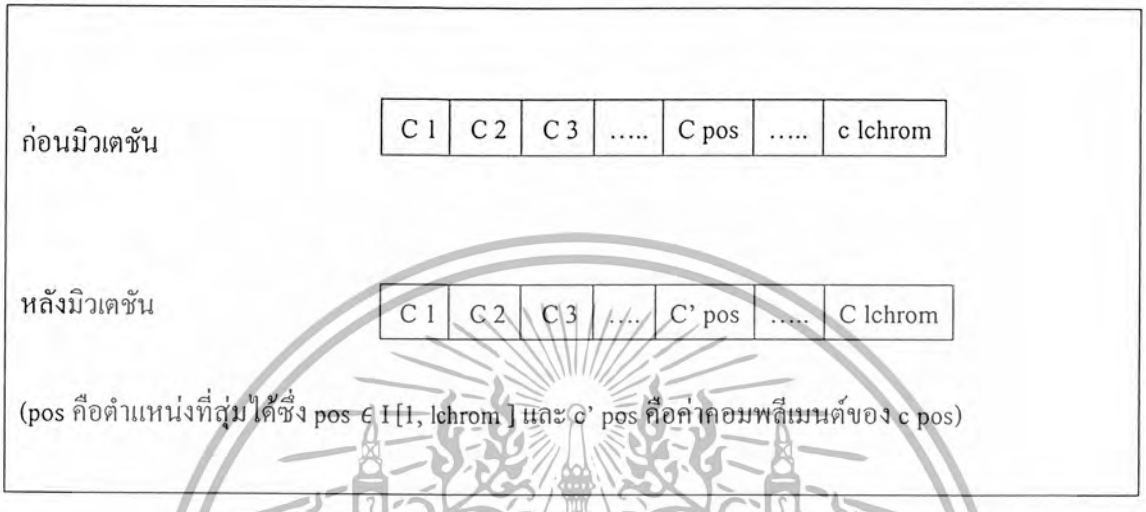
```
Parent 1: A B C D E F  |  G H I J K
Parent 2: B E I G H F  |  A C D K J

                       |

Offspring 1: A B C D E F  |  _ _ _ _ _
Offspring 2: B E I G H F  |  _ _ _ _ _

                       |

Offspring 1: A B C D E F  |  I G H K J
Offspring 2: B E I G H F  |  A C  D J K
```

รูปที่ 2.1.13 ออร์เดอร์ครอสโอเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2.3 มิวเตชั่น (Mutation)

เป็นตัวดำเนินการผ่าเหล่าตัวหนึ่งที่อาจช่วยให้โครโมโซม มีค่าความเหมาะสมดีขึ้นหลังจากครอสโอเวอร์ โดยกลับค่าของบิทเป็นค่าใหม่ในตำแหน่งบิทที่สุ่มได้ ตามอัตราความน่าจะเป็นของการมิวเตชันในแต่ละบิท (Probability of Mutation: P_m) ที่กำหนด สำหรับการมิวเตชันของ SGA นั้นเป็นแบบไบนารีมิวเตชัน (Binary Mutation) โดยกลับค่าบิทเป็นค่าคอมพลีเมนต์ คือ จาก 0 เป็น 1 หรือ จาก 1 เป็น 0 ดังรูป



รูปที่ 2.1.14 ไบนารีมิวเตชัน

จำนวนการมิวเตชันในแต่ละรุ่นขึ้นอยู่กับการกำหนดค่า P_m ซึ่งแตกต่างกันในแต่ละปัญหา เช่น ถ้าจำนวนประชากรแต่ละรุ่น popsize เท่ากับ 30 โครโมโซม ซึ่งแต่ละโครโมโซมประกอบด้วย 5 บิท และกำหนดให้ $P_m = 0.2$ แล้ว จำนวนการมิวเตชันในแต่ละรุ่นเท่ากับ P_m * popsize * lchrom = 0.02 * 30 * 5 = 3 บิท

ตัวอย่าง : กำหนด $P_m = 0.1$ ดำเนินการมิวเตชันโครโมโซมลูกที่ได้จากการครอสโอเวอร์ ดังนี้

ลำดับ	ก่อนมิวเตชัน	เลขสุ่ม (r)	หลังมิวเตชัน	X	ค่าความเหมาะสม (F)
1	0 1 0 0 1	0.581 0.346 0.062 0.785 0.401	0 1 1 0 1	13	169
2	1 1 1 1 0	0.829 0.534 0.947 0.308 0.277	1 1 1 1 0	30	900
3	1 1 0 0 1	0.398 0.646 0.494 0.765 0.029	0 1 0 0 0	24	576
4	1 0 0 1 1	0.175 0.335 0.837 0.577 0.308	1 1 1 1 0	19	361
	รวม				2006
	ค่าเฉลี่ย				502
	ค่าสูงสุด				900

ตารางที่ 2.1.5 แสดงตัวอย่างของการมิวเตชันภายหลังจากการครอสโอเวอร์แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการสุ่มตำแหน่งที่จะมิวเตชันโดยสร้างเลขสุ่ม r ของแต่ละตำแหน่งบิทในแต่ละโครโมโซมแล้ว ตำแหน่งบิทที่ 3 ของโครโมโซมลำดับที่ 1 และตำแหน่งบิทที่ 4 ของโครโมโซมลำดับที่ 3 เป็นตำแหน่งที่ $r \leq 0.1$ ตามอัตราการที่กำหนดจึงเกิดมิวเตชัน ทำให้โครโมโซมมีค่าความเหมาะสมจาก 81 และ 625 เป็น 169 และ 576 ตามลำดับ จะเห็นได้ว่ามิวเตชันเป็นตัวดำเนินการที่อาจทำให้โครโมโซมมีค่าความเหมาะสมสูงขึ้นหรือลดลงได้ แต่อย่างไรก็ตามค่าเฉลี่ยของค่าความเหมาะสมดีขึ้นจาก 492 เป็น 502 แสดงถึงการหาคำตอบของ SGA โดยส่วนมากดีขึ้น และความสำคัญของการหาคำตอบของ GA นั้นเป็นความต้องการได้คำตอบโดยพิจารณาจากคำตอบที่ดีขึ้นเกิดขึ้น ซึ่งจะมีโอกาสอยู่รอดเพื่อถ่ายทอดส่วนที่ดีในรุ่นต่อไป

ประชากรรุ่นลูกที่ถูกสร้างใหม่จะถูกการมิวเตชั่นตามค่า Mutation Probability (P_m) การมิวเตชั่นจะกระทำโดยการสลับขีน ดังรูปที่ 2.1.14 ซึ่งจะสร้างการเปลี่ยนแปลงแบบสุ่มหลังจากการครอสโอเวอร์ Pm จะเป็นตัวกำหนดอัตราส่วนของจำนวนของประชากรรุ่นลูกที่จะถูกสร้างใหม่ในแต่ละรุ่นต่อจำนวนของประชากรทั้งรุ่น ถ้า P_m มีค่าน้อยประชากรรุ่นลูกจะมีอัตราการเปลี่ยนค่าต่ำ ถ้า P_m มีค่ามาก ประชากรรุ่นลูกจะมีการเปลี่ยนแปลงค่าสูง แต่ส่งผลต่อประชากรรุ่นพ่อแม่และรุ่นลูกจะมีความแตกต่างกันมาก ซึ่งถ้ามีการปรับค่าที่ดีขึ้นก็จะเหมาะสม แต่ถ้ามีการปรับค่าที่แย่ลงโดยประชากรรุ่นลูกมีค่าแย่แก่รุ่นพ่อแม่

Before Mutation: 1 1 0 1 0 0 0 1 0 0 1 1

After Mutation: 1 1 0 0 0 0 0 1 0 0 1 1

รูปที่ 2.1.15 มิวเตชั่น

2.1.3 ตัวอย่างของเจเนติก อัลกอริทึมแบบพื้นฐาน (Simple Genetic Algorithm: SGA)

GA ในยุคเริ่มแรกของ Holland นั้นคือ เจเนติก อัลกอริทึมแบบง่าย (Simple Genetic Algorithm: SGA) ซึ่งมีขั้นตอนพื้นฐานที่มีกระบวนการไม่มากนักและง่ายในการศึกษาความเข้าใจแต่ละขั้นตอนการทำงานของ GA เพื่อแก้ปัญหาในการหาคำตอบ แบ่งออกเป็น 2 ส่วน คือ ขั้นเตรียมการและขั้นตอนการทำงาน สำหรับในส่วนของขั้นเตรียมการนั้นเป็นส่วนของการปรับรูปแบบของปัญหาให้เหมาะสมสำหรับการนำเสนอ GA เพื่อใช้ในการแก้ปัญหานั้นๆ ประกอบด้วย

1. กำหนดฟังก์ชั่นความเหมาะสม เพื่อความสะดวกและง่ายต่อความเข้าใจในขั้นตอนการทำงานต่างๆ จะกำหนดตัวอย่างปัญหาสำหรับอธิบายรายละเอียดการหาคำตอบของ SGA คือ ปัญหาการหาค่าสูงสุดของฟังก์ชั่น $y = x^2$ ที่ x มีค่าระหว่างจำนวนเต็ม I [0, 31] ดังนั้น

ตัวอย่าง : ฟังก์ชั่นเป้าหมาย คือ $f = x^2$

และกำหนดให้ฟังก์ชั่นความเหมาะสม คือ $F = x^2$

ซึ่งคำตอบที่ดีที่สุด คือ ค่า X ที่มีค่าความเหมาะสมสูงสุด (Max (F))

2. กำหนดรูปแบบโครโมโซม รูปแบบโครโมโซมของ SGA นั้นเป็นแบบไบนารี โดยค่าตัวแปรหรือพารามิเตอร์ของปัญหาจะถูกแปลงให้อยู่ในรูปของไบนารีโครโมโซม คือประกอบด้วยบิทที่มีค่าเป็น 0 หรือ 1 ซึ่งเป็นค่าในเลขฐานสอง และมีความยาว (Chromosome Length : lchrom) ตามแต่จะกำหนด ซึ่งแสดงด้วยสัญลักษณ์ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

B1	B2	B3	.....	B lchrom

ซึ่ง $B_i \in I[0,1]$

ตารางที่ 2.1.6 การกำหนดรูปแบบโครโมโซมของ SGA

ตัวอย่าง : วิธีการเข้ารหัสแบบไบนารีโดยแปลงค่าพารามิเตอร์ X ให้อยู่ในรูปไบนารีบิท 5 บิท (lchrom = 5)

ดังนั้นโครโมโซมของปัญหาจะมีค่าอยู่ในช่วง 00000 ถึง 11111 ซึ่งเมื่อถอดรหัสแล้วจะทำให้ X มีค่าอยู่ในช่วง 0 ถึง 31 ตามที่ต้องการ

ในส่วนของรายละเอียดขั้นตอนการทำงานของ SGA จะเป็นขั้นตอนพื้นฐานเบื้องต้นแบบง่าย ประกอบด้วย

1. ประชากรรุ่นเก่า (Old Population) เป็นชุดโครโมโซมที่จะถูกคัดเลือกไปเป็นต้นแบบสำหรับสร้างประชากรรุ่นใหม่ (New Population) ในวิวัฒนาการรุ่น (generation :gen) ต่อไป โดยประชากรเริ่มต้นที่ gen = 0 จะถูกสร้างขึ้นโดยการสุ่มตามจำนวนโครโมโซมในแต่ละรุ่น (Population Size :popsize) ที่กำหนด

ตัวอย่าง :

ลำดับ	โครโมโซม
1	01110
2	11001
3	01000
4	10011

ตารางที่ 2.1.7 แสดงจำนวนประชากรในแต่ละรุ่น

ชุดโครโมโซมในรุ่นเริ่มต้นนี้เป็นชุดโครโมโซมที่กำหนดให้ ในแต่ละรุ่นประกอบด้วย 4 โครโมโซม ซึ่งแต่ละโครโมโซมเกิดจากการสุ่มค่าไบนารี 0 หรือ 1 จำนวน 5 ครั้ง

2. วิเคราะห์ค่าความเหมาะสม เป็นขั้นตอนของการถอดรหัสจากรูปแบบโครโมโซมที่กำหนดไว้เพื่อคำนวณค่าความเหมาะสมตามฟังก์ชันความเหมาะสมของปัญหาของแต่ละโครโมโซม ในที่นี้ฟังก์ชันเป้าหมายหรือฟังก์ชันความเหมาะสม คือ $F=x^2$ ดังนั้นการวิเคราะห์ค่าความเหมาะสมของ SGA โดยถอดรหัสเลขฐานสองของแต่ละโครโมโซมเป็นค่าของตัวแปร X และคำนวณค่าความเหมาะสมคือค่า x^2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง :

ลำดับ	โครโมโซม	%	ค่าความเหมาะสม
1	01110	14	196
2	11001	25	625
3	01000	8	64
4	10011	19	361

ตารางที่ 2.1.8 แสดงค่าความเหมาะสมและรูปแบบของโครโมโซม

ชุดโครโมโซมในรุ่นเริ่มต้นมีค่าความเหมาะสมเป็น 196, 625, 64 และ 361 ตามลำดับ

3. ประชากรรุ่นใหม่ เป็นชุดโครโมโซมลูกที่เกิดจากขั้นตอนของวิวัฒนาการต่างๆ ทั้งหมด ซึ่งประชากรรุ่นใหม่ทั้งหมดที่เกิดขึ้น จะถูกถ่ายทอดกลายเป็นประชากรรุ่นเก่าสำหรับวิวัฒนาการในรุ่นถัดไป ซึ่งเรียกวิวัฒนาการแบบนี้ว่า การถ่ายทอดแบบทั่วไป หรือรีโพรดักชันแบบทั่วไป (General Reproduction) กระบวนการต่างๆ จะถูกปฏิบัติซ้ำๆ จนกระทั่งถึงรุ่นที่มากที่สุด (Max Generation: maxgen) ที่ต้องการ

การทดลองเพื่อทดสอบการทำงานของSGAจะทดสอบโดยใช้ในการหาค่าสูงสุดของฟังก์ชัน$y = x^n$ ที่ x มีค่าระหว่าง I [1, $2^l - 1$] ซึ่ง l คือ จำนวนบิตของโครโมโซม ด้วยวิธีการของ SGA กำหนดให้ฟังก์ชันเป้าหมาย คือ $f = x^n$ เพื่อความสะดวกในการวิเคราะห์การทำงานของ SGA จึงปรับให้ค่าความเหมาะสมอยู่ในช่วง [0, 1] ซึ่งคำตอบที่ดีที่สุดจะมีค่าความเหมาะสมเป็น 1 ดังนั้น

ฟังก์ชันความเหมาะสม คือ $$F = \left(\frac{x}{2^l - 1}\right)^n$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SGA Parameters

Population size (popsize)	=	30
Chromosome length (l)	=	30
Maximum # of generation (maxgen)	=	60
Crossover probability (Pc)	=	8.00000000000E-01
Mutation probability (Pm)	=	3.33000000000E-02

Initial Generation Statistics

Initial population max fitness	=	9.4811791207E-01
Initial population average fitness	=	1.0300239870E-01
Initial population minimum fitness	=	5.3711099438E-21
Initial population sum fitness	=	3.0900719609E+00

Population Report

#	Generation 0 string	x	fitness	#	pos parents 1	2	Generation 1 string	x	fitness
1)	111111101010001111000110101100	1.068E+09	0.9481	1)	(1, 2)	5 30	111001101010001111000110101100	9.674E+08	0.3523
2)	111101110011111100110101101111	1.037E+09	0.7062	2)	(1, 2)	5 30	111101111011111001101011100101	1.039E+09	0.7206
3)	111011000011011100101110001111	9.908E+08	0.4474	3)	(3, 3)	16 30	111101000011011110101110001111	1.024E+09	0.6243
4)	111000111100010110000110110010	9.553E+08	0.3109	4)	(3, 3)	16 30	111011000011011100101110001111	9.908E+08	0.4474
5)	110111111011000110101100100001	9.382E+08	0.2595	5)	(1, 4)	18 30	111111101010001111000110101100	1.068E+09	0.9481
6)	110101110011111110111010111011	9.028E+08	0.1766	6)	(1, 4)	18 30	101000111100001011000011010010	6.869E+08	0.0115
7)	110011101000000001101110001010	8.661E+08	0.1166	7)	(3, 6)	13 30	111011000011011100101110000111	9.908E+08	0.4474
8)	101111101100010111110001000111	8.002E+08	0.0528	8)	(3, 6)	13 30	110100111001111111011110101011	8.860E+08	0.1464
9)	101100000010001111110111000110	7.388E+08	0.0236	9)	(6, 4)	11 30	110101110011111110111110001011	9.028E+08	0.1766
10)	101100000000011001110100111111	7.383E+08	0.0236	10)	(6, 4)	11 30	111000111100010110000110110010	9.553E+08	0.3109
11)	101001010001010111110101101000	6.924E+08	0.0124	11)	(2, 5)	1 30	111101110011111101110101101111	1.037E+09	0.7062
12)	100111101011110011110011101110	6.658E+08	0.0084	12)	(2, 5)	1 30	110111110011000110101101001001	9.361E+08	0.2538
13)	100001010111001101111111111001	5.597E+08	0.0015	13)	(3, 7)	12 30	111010000101101110010111010111	9.761E+08	0.3853
14)	011111010010001001101010010001	5.309E+08	0.0009	14)	(3, 7)	12 30	110011101010000001101110001010	8.667E+08	0.1173
15)	011111000000010111110010010001	5.202E+08	0.0007	15)	(6, 1)	6 30	110101111001111111011110101011	9.028E+08	0.1766
16)	011101010011100101000001001100	4.917E+08	0.0004	16)	(6, 1)	6 30	111111101010001111000110101100	1.068E+09	0.9481
17)	011011100111010100101011110010	4.633E+08	0.0002	17)	(1, 1)	15 30	111111111010001111000110101100	1.072E+09	0.9860
18)	010010110000011110000101001000	3.148E+08	0.0000	18)	(1, 1)	15 30	111111101010001111000110101101	1.068E+09	0.9481
19)	001111001001000101110001110000	2.540E+08	0.0000	19)	(1, 2)	5 30	111111101010001111000110101100	1.068E+09	0.9481
20)	001101110101001001011100010111	2.320E+08	0.0000	20)	(1, 2)	5 30	111101110011101100110101111111	1.037E+09	0.7057
21)	001100110011010000110111111011	2.148E+08	0.0000	21)	(3, 6)	1 30	111011000011011100101110001111	9.908E+08	0.4474
22)	001100110010111111100110001111	2.147E+08	0.0000	22)	(3, 6)	1 30	110101110011111110111101011111	9.028E+08	0.1766
23)	001100010110001010000101010100	2.071E+08	0.0000	23)	(1, 3)	18 30	111110101010001111000110101100	1.051E+09	0.8093
24)	001011100001100001101101110000	1.933E+08	0.0000	24)	(1, 3)	18 30	111011000011011100101110011111	9.908E+08	0.4474
25)	000111110110001111101001100111	1.317E+08	0.0000	25)	(1, 4)	3 30	111111101100001111000101101100	1.069E+09	0.9528
26)	000100101101001000001110110111	7.894E+07	0.0000	26)	(1, 4)	3 30	111000111100010110000010110010	9.553E+08	0.3109
27)	000011010100111011000000000011	5.582E+07	0.0000	27)	(1, 5)	1 30	111111101010001111000110101100	1.068E+09	0.9481
28)	000001100001110011011100000010	2.564E+07	0.0000	28)	(1, 5)	1 30	110111111111000110101100100001	9.393E+08	0.2624
29)	000001011011011001110111100010	2.396E+07	0.0000	29)	(6, 2)	21 30	110101110011111110111101011011	9.028E+08	0.1766
30)	000000100110011111011101101111	1.009E+07	0.0000	30)	(6, 2)	21 30	111101110011111100110101100111	1.037E+09	0.7062

Note: Generation 1 & Accumulated Statistics: max=0.9860, min=0.0115, avg=0.5199, sum=15.5984, nmutation=32, ncross= 12

(a) วิวัฒนาการรุ่นที่ 1

รูปที่ 2.1.16 รายละเอียดผลการทำงานของ SGA ในการหาค่าสูงสุดของฟังก์ชัน $y = x^n$ ที่ $n = 10$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Population Report

#	string	Generation 5 x	fitness	# parents	pos 1	2	string	Generation 6 x	fitness
1)	11111111101000111100011010110	1.072E+09	0.9860	1) (4,20)	1	30	11100111101000111100011010110	9.716E+08	0.3679
2)	11111111101000111100011010110	1.072E+09	0.9860	2) (4,20)	1	30	01111100100001111100001010101	5.223E+08	0.0007
3)	11111111101000111100011010110	1.072E+09	0.9860	3) (18, 3)	27	30	11111010001000011100011010110	1.049E+09	0.7930
4)	11111111101000111100011010110	1.072E+09	0.9860	4) (18, 3)	27	30	11111111101000111100011010110	1.072E+09	0.9860
5)	11111111100000111100100101101	1.072E+09	0.9812	5) (16, 6)	11	30	11111100001101101010111000010	1.058E+09	0.8516
6)	11111111100000111100011010110	1.072E+09	0.9812	6) (16, 6)	11	30	01111111100000111100011010110	5.348E+08	0.0009
7)	11111111100000111100011000110	1.072E+09	0.9812	7) (21, 5)	20	30	11111100001000110100010010100	1.058E+09	0.8590
8)	11111110100101111010110101100	1.071E+09	0.9728	8) (21, 5)	20	30	11111111100000111111010010010	1.072E+09	0.9812
9)	11111101100001111000100100101	1.069E+09	0.9528	9) (26,16)	7	30	11101100001101111010111000010	9.908E+08	0.4474
10)	11111101010001011000110100100	1.068E+09	0.9480	10) (26,16)	7	30	01111100001001101010111000010	5.207E+08	0.0007
11)	11111101000101111000110101101	1.065E+09	0.9446	11) (26, 4)	5	30	11101100001101111010111000010	9.908E+08	0.4474
12)	11111011000000111100011101100	1.063E+09	0.9071	12) (26, 4)	5	30	11111111101000111100011010110	1.072E+09	0.9860
13)	11111100101000111100110011000	1.060E+09	0.8762	13) (1, 2)	14	30	11111111101000111100011010110	1.072E+09	0.9860
14)	11111100101000111100110010100	1.060E+09	0.8762	14) (1, 2)	14	30	11111111111000111100111010110	1.073E+09	0.9957
15)	11111100101000111100110010100	1.060E+09	0.8762	15) (17,12)	1	30	11111010101100011100011010110	1.051E+09	0.8110
16)	11111100000110110101011100001 10	1.058E+09	0.8616	16) (17,12)	1	30	11111101100000111100011101100	1.063E+09	0.9071
17)	11111010101100011100011010110	1.051E+09	0.8110	17) (15,21)	27	30	11111100101000111001110010100	1.060E+09	0.8762
18)	11111010001000011100011010110	1.049E+09	0.7930	18) (15,21)	27	30	11110100001000111100110010100	1.024E+09	0.6223
19)	11111001001110100011010101111	1.045E+09	0.7648	19) (1,14)	6	30	11111111101000111100011010100	1.072E+09	0.9860
20)	11110100100001111100001010100	1.026E+09	0.6323	20) (1,14)	6	30	11110100101000111100110010100	1.026E+09	0.6351
21)	11110100001000111100110010100	1.024E+09	0.6223	21) (21,29)	7	30	11110100001000111100110010100	1.024E+09	0.6223
22)	11110000100010111100000010100	1.009E+09	0.5365	22) (21,29)	7	30	11100100011101110010111000101 1	9.583E+08	0.3205
23)	11101110110100111100010110010 0	1.002E+09	0.4994	23) (21,13)	22	30	11110100001010111100110010100	1.024E+09	0.6231
24)	11101110100000111100010010101	1.000E+09	0.4929	24) (21,13)	22	30	11011100101010111100110011000	9.256E+08	0.2265
25)	11101101100000111100011001100	9.962E+08	0.4726	25) (13, 6)	22	30	11111100101000111100110011000	1.060E+09	0.8762
26)	11101100000110111010111000010	9.908E+08	0.4474	26) (13, 6)	22	30	11111111100000111100111010101	1.072E+09	0.9812
27)	11101100000110111001011000010	9.908E+08	0.4474	27) (19,11)	5	30	11111001001111100011010111111	1.045E+09	0.7653
28)	11101011111001011001101010010	9.894E+08	0.4414	28) (19,11)	5	30	11111101000010111100110101101	1.068E+09	0.9446
29)	11100100011101110010111000101 1	9.583E+08	0.3205	29) (19, 3)	1	30	11111001001110100011010111111	1.045E+09	0.7646
30)	01100110110100111100110110010 0	4.313E+08	0.0001	30) (19, 3)	1	30	11111011101000111100011010100	1.055E+09	0.8422

Note: Generation 6 & Accumulated Statistics: max=0.9957, min=0.0007, avg=0.6839, sum=20.5181, nmutation=185, ncross= 74

(b) วิวัฒนาการรุ่นที่ 6

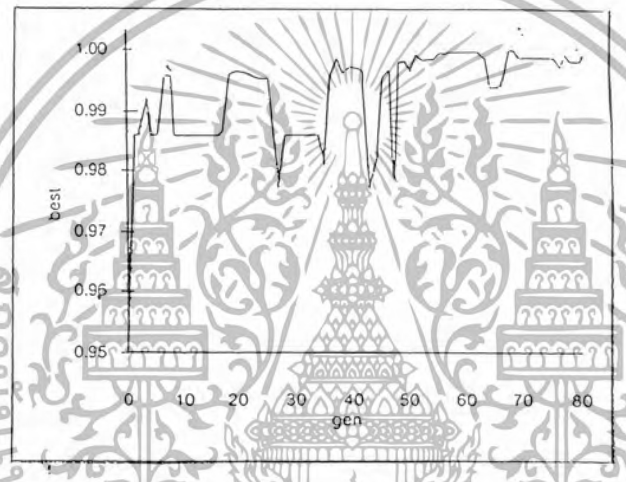
รูปที่ 2.1.17 (ต่อ) รายละเอียดผลการทำงานของ SGA ในการหาค่าสูงสุดของฟังก์ชัน $y = x^n$ ที่ n = 10

ผลการทดสอบการทำงานของSGAในการหาคำตอบ เมื่อกำหนดให้ n=10 และกำหนดค่า SGA พารามิเตอร์คือ l = 30, popsize = 30, maxgen = 30, Pc = 0.8 และ Pm = 0.0333 จะเห็นว่าในรุ่นเริ่มต้น (gen=0) ซึ่งประชากรโครโมโซมเกิดจากการสุ่มนั้นคำตอบที่ดีที่สุดคือ 0.9481 ดังรูปที่ 2.1.17(a) และผลการหาคำตอบของ SGA ในรุ่นที่ 1 ซึ่งแสดงรายละเอียดการจับคู่โครโมโซมพ่อ – แม่ในคู่ลำดับของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

parents และตำแหน่งในการครอสโอเวอร์แลกเปลี่ยนค่าบิทตั้งแต่ตำแหน่ง pos1 ที่สุ่มได้ถึง pos2 = 30 ซึ่งหาก pos1 มีค่าเป็น 1 และ pos2 มีค่าเป็น 30 นั่นหมายถึงคู่โครโมโซมพ่อ – แม่คู่นั้นไม่มีการครอสโอเวอร์ จะเห็นว่าการครอสโอเวอร์ทำให้โครโมโซมมีความเปลี่ยนแปลงรูปแบบของบิทให้คล้ายกันมากขึ้นในแบบของโครโมโซมที่ดีซึ่งมีค่าความเหมาะสมสูง. และแสดงผลจำนวนการครอสโอเวอร์และจำนวนบิทที่เกิดมิวเตชันทั้งหมดคือ ncross และ nmutation ตามลำดับ รวมทั้งสรุปผลค่าความเหมาะสมสูงสุด ต่ำสุด และ ค่าเฉลี่ย คือ max , min และ avg ตามลำดับ ซึ่งในรุ่นที่ 1 นี้ SGA สามารถหาคำตอบที่ดีที่สุดคือ max เท่ากับ 0.9860 รูปที่ 2.1.17(b) แสดงผลวิวัฒนาการของ SGA ในรุ่นที่ 6 จะเห็นได้ว่า รูปแบบโครโมโซมที่ดีเกิดมากยิ่งขึ้น และ SGA สามารถหาคำตอบได้ดีขึ้นเป็น 0.9957

2.1.4 การประยุกต์เจเนติก อัลกอริทึมแบบพื้นฐาน



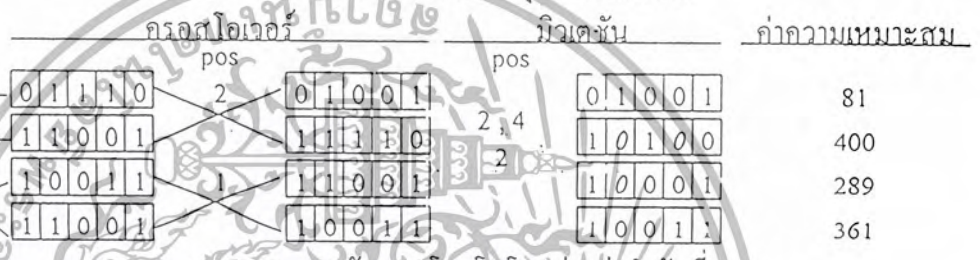
รูปที่ 2.1.18 แสดงกราฟค่าความเหมาะสมสูงสุดของ SGA ในการหาค่าสูงสุดของฟังก์ชัน $y = x^n$ ที่ $n = 10$

เมื่อวิเคราะห์การทำงานของ SGA ซึ่งเป็น GA ในยุคแรกๆ แล้วจะเห็นว่า วิธีการของ GA เป็นการหาคำตอบแบบสุ่ม ซึ่งเป็นวิธีการที่ไม่มีการบันทึกหรือจดจำคำตอบที่ดีที่สุดของรุ่นก่อน จึงทำให้การหาคำตอบของ SGA ได้คำตอบที่ดีมากขึ้นหรือน้อยลงได้ ดังรูปที่ 2.1.18 แสดงกราฟคำตอบค่าความเหมาะสมสูงสุดแต่ละรุ่นภายใน 30 รุ่นในการหาค่าสูงสุดของฟังก์ชัน $y = x^n$ ที่ $n = 10$ จะเห็นได้ว่าคำตอบค่าความเหมาะสมสูงสุดของ SGA มีค่าสูงขึ้นและลดลงด้วยวิธีการทำงานที่อาศัยการสุ่ม ดังนั้นหากสามารถพัฒนาวิธีการค้นหาคำตอบของ GA ให้ดีขึ้นแล้วก็จะเป็นการปรับปรุงสมรรถนะของ GA ยิ่งขึ้น สำหรับหาค่าสูงสุดของฟังก์ชัน $y = x^n$ เราปรับปรุงการค้นหาคำตอบของ GA โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับ	โครโมโซม รุ่นเก่า	ค่าความ เหมาะสม	จำนวนที่ สุ่มได้	โครโมโซม รุ่นใหม่	ค่าความ เหมาะสม
1	01110	196	1		
2	11001	625	2		
3	01000	64	0		
4	10011	361	1		

1.) สร้างโครโมโซมรุ่นใหม่ทั้งหมด



2.) คัดลอกโครโมโซมรุ่นเก่าลำดับที่ 2

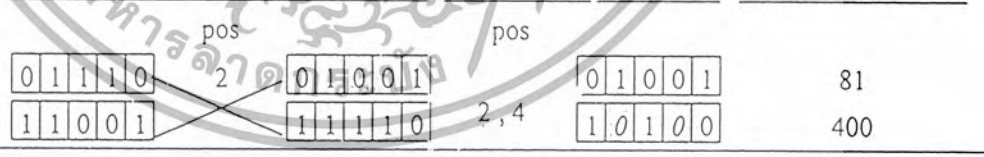
				11001	625

(a) รักษาค่าความเหมาะสมที่ดี 1 โครโมโซม

ลำดับ	โครโมโซม รุ่นเก่า	ค่าความ เหมาะสม	จำนวนที่ สุ่มได้	โครโมโซม รุ่นใหม่	ค่าความ เหมาะสม
1	01110	196	1	11001	625
2	11001	625	2	10011	361
3	01000	64	0		
4	10011	361	1		

1.) คัดลอกโครโมโซมรุ่นเก่าลำดับที่ 2 และ 4

2.) สร้างโครโมโซมรุ่นใหม่ทั้งหมด



				01001	81
				10100	400

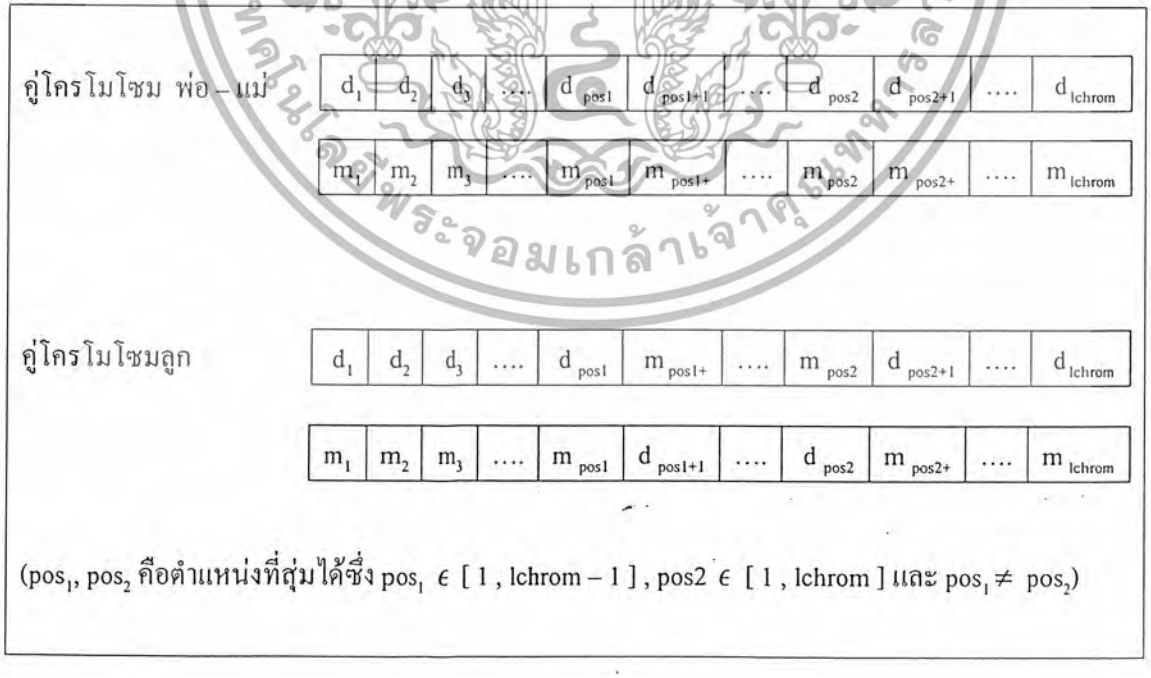
รักษาค่าความเหมาะสมที่ดี 2 โครโมโซม

1. รีโพรดักชันแบบรักษาค่าความเหมาะสมที่ดี เนื่องจากในการค้นหาคำตอบของ SGA นั้นมีโอกาสที่จะสูญเสียโครโมโซมในรุ่นเก่า ที่มีค่าความเหมาะสมที่ดีไปได้ ซึ่งจะทำให้คำตอบในรุ่นถัดไปนั้นดีมากขึ้นหรือน้อยลง ดังนั้นหากปรับปรุง SGA ให้ควบคุมการค้นหาคำตอบ โดยรักษาโครโมโซมที่ดีไว้แล้ว จะช่วยให้วิวัฒนาการคำตอบในรุ่นถัดไปดีขึ้นเรื่อยๆ โดยมีวิธีการดังนี้

- กำหนดจำนวนโครโมโซมที่ดีที่สุด (# Best) ของรุ่นเก่าที่ต้องการรักษา เป็น 1, 2, 4, ...
- ถ้าจำนวนโครโมโซมที่กำหนดเป็น 1 ให้สร้างชุดโครโมโซมรุ่นใหม่ทั้งหมด แล้วจึงคัดลอก (Copy) โครโมโซมที่ดีที่สุดของรุ่นเก่า มาแทนที่โครโมโซมรุ่นใหม่ที่มีค่าความเหมาะสมน้อยที่สุด
- ถ้าจำนวนโครโมโซมที่กำหนดเป็น 2 , 4, ... ให้คัดลอกโครโมโซมที่ดีที่สุดในรุ่นเก่า ตามจำนวนที่กำหนดมาเป็นโครโมโซมรุ่นใหม่ แล้วจึงสร้างโครโมโซมรุ่นใหม่ส่วนที่เหลือต่อไป

ดังรูปที่ 2.1.18 a แสดงวิธีการรีโพรดักชันแบบรักษาค่าความเหมาะสมที่ดี 1 โครโมโซม และรูปที่ 2.1.18 b แสดงวิธีการรีโพรดักชันแบบรักษาค่าความเหมาะสมที่ดี 2 โครโมโซม

2. ครอสโอเวอร์แบบ 2 จุด การแลกเปลี่ยนส่วนของโครโมโซมพ่อ – แม่นั้น บางครั้งหากแลกเปลี่ยนค่าบิทเพียงบางช่วงของโครโมโซมแล้วจะสร้างโครโมโซมที่ดีกว่า เช่น การหาค่าสูงสุดของฟังก์ชัน $y = x^2$ ของคู่โครโมโซมพ่อ – แม่ 01110 และ 11001 ซึ่งมีค่าความเหมาะสมเป็น 196 และ 625 หากแลกเปลี่ยนค่าบิตตำแหน่งที่ 3 และ 4 เท่านั้นจะทำให้เกิดโครโมโซมลูกคือ 01*000* และ 11*111* มีค่าความเหมาะสมเป็น 64 และ 961 ซึ่งโครโมโซม 11111 เป็นโครโมโซมที่ให้คำตอบที่ดีที่สุดที่ต้องการ ดังนั้นการประยุกต์ SGA โดยพัฒนาตัวดำเนินการครอสโอเวอร์ให้เป็นแบบ 2 จุด (Two – point Crossover) จะทำให้ SGA ค้นหาคำตอบที่ดีขึ้นได้ ดังรูปที่ 2.1.19 มีวิธีการดังนี้



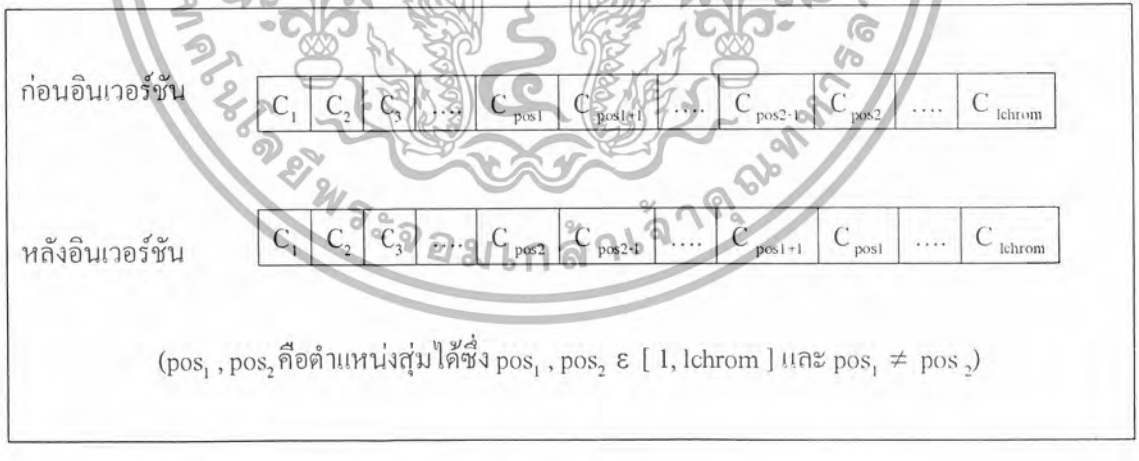
รูปที่ 2.1.19 ครอสโอเวอร์แบบ 2 จุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สุ่มเลือกตำแหน่ง pos_1, pos_2 คือตำแหน่งเริ่มต้นและตำแหน่งสุดท้ายที่จะครอสโอเวอร์ตามลำดับ ซึ่ง pos_1 มีค่าอยู่ในช่วง [1, lchrom – 1] และ pos_2 มีค่าอยู่ในช่วง [1, lchrom] โดยที่ pos_1 มีค่าน้อยกว่า pos_2
- แลกเปลี่ยนค่าในแต่ละบิตของคู่โครโมโซมพ่อ – แม่ตั้งแต่ตำแหน่งที่ pos_1+1 ถึง pos_2

3. ไบนารีมิวเตชันแบบกำหนดค่าบิต เนื่องจากการหาคำตอบของ SGA นั้น กระบวนการไบนารีมิวเตชันอาจทำให้โครโมโซมที่เปลี่ยนแปลงไปให้คำตอบที่ดีลดลงและทำให้สูญเสียโครโมโซมที่ดีไป เช่น โครโมโซม 11110 มีค่าความเหมาะสมเป็น 900 หากสุ่มได้บิตตำแหน่งที่ 1 เกิดมิวเตชันแล้วโครโมโซมที่เกิดขึ้นจากการมิวเตชันคือ 01110 ทำให้มีค่าความเหมาะสมลดลงเป็น 196 แต่ในบางครั้งข้อดีหรือจุดเด่นของปัญหาจะสามารถนำมาปรับให้เข้ากับการค้นหาคำตอบที่ดีขึ้นได้ สำหรับในการหาค่าสูงสุดของฟังก์ชัน $y = x^n$ นี้ค่าบิตของโครโมโซมที่เป็น 1 จะทำให้ค่าความเหมาะสมสูงขึ้นเสมอ ดังนั้นหากปรับปรุงไบนารีมิวเตชันให้เป็นแบบกำหนดค่าแน่นอนให้กับบิตที่เกิดมิวเตชัน โดยกำหนดให้บิตที่เกิดมิวเตชันมีค่าบิตเป็น 1 เสมอ จะช่วยปรับแนวทางการค้นหาคำตอบของ SGA ดีขึ้น เช่น หากประยุกต์ไบนารีมิวเตชันที่กำหนดค่าบิตให้เป็น 1 เสมอกับโครโมโซม 11110 ในตำแหน่งที่ 1 แล้วโครโมโซมที่เกิดขึ้นจะเหมือนเดิม และยังเป็นการรักษาโครโมโซมที่มีค่าความเหมาะสมที่ดีไว้ด้วย

4.อินเวอร์ชัน (Inversion) เป็นตัวดำเนินการที่ประยุกต์เพิ่มเติมใน SGA โดยจำลองแบบลักษณะของการอินเวอร์ชันในทางพันธุศาสตร์ที่เป็นลักษณะของการกลับหัวกลับหางส่วนของยีนภายในโครโมโซม ที่อาจช่วยให้เกิดโครโมโซมที่ดีขึ้นได้ โดยการกลับส่วนค่าบิตภายในช่วงตำแหน่งของโครโมโซมที่สุ่มได้ตามอัตราค่าความน่าจะเป็นของการอินเวอร์ชันแต่ละโครโมโซม (Probability of Inversion: Pi) ที่กำหนด ดังรูปที่ 2.1.20 มีขั้นตอนดังนี้



รูปที่ 2.1.20 อินเวอร์ชัน

- สุ่มเลือกตำแหน่ง pos_1, pos_2 คือตำแหน่งเริ่มต้นและตำแหน่งสุดท้ายที่จะอินเวอร์ชันตามลำดับ ซึ่ง pos_1 และ pos_2 มีค่าอยู่ในช่วง [1, lchrom] โดยที่ pos_1 มีค่าน้อยกว่า pos_2
- กลับค่าบิตในช่วงของตำแหน่งที่ pos_1 ถึง pos_2 ของโครโมโซม โดยสลับค่าบิต pos_1 กับ pos_2, pos_1+1 กับ pos_2-1, pos_1+2 กับ pos_2-2,....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่น สุ่มโครโมโซมที่จะอินเวอร์ชัน คือ 01010 มีค่าความเหมาะสมเป็น 100 โดยสุ่มตำแหน่ง $pos_1 = 1$ และ $pos_2 = 4$ แล้ว จะเห็นว่าการอินเวอร์ชันทำให้เกิดโครโมโซม ***10100*** ซึ่งมีค่าความเหมาะสมดีขึ้นเป็น 400 เป็นต้น สำหรับจำนวนการอินเวอร์ชันในแต่ละรุ่นขึ้นอยู่กับการกำหนดค่า Pi ซึ่งแตกต่างกันในแต่ละปัญหา เช่น ถ้าจำนวนประชากรแต่ละรุ่น popsize เท่ากับ 30 โครโมโซม และกำหนดให้ Pi = 0.1 แล้ว จำนวนการอินเวอร์ชันในแต่ละรุ่นเท่ากับ Pi * popsize = 0.1 * 30 = 3 ครั้ง

สรุปภาคทฤษฎีเจเนติก อัลกอริทึม

เจเนติก อัลกอริทึมเป็นทฤษฎีที่เลียนแบบกระบวนการวิวัฒนาการทางธรรมชาติ ที่นำมาประยุกต์ใช้กับคอมพิวเตอร์เพื่อช่วยแก้ปัญหาในการหาคำตอบต่างๆ ซึ่งมีพื้นฐานการทำงานเบื้องต้นเป็นเจเนติก อัลกอริทึมแบบง่าย โดยมีรูปแบบโครโมโซมเป็นแบบไบนารี และตัวดำเนินการทางพันธุศาสตร์ คือ การครอสโอเวอร์และมิวเตชันที่ไม่ซับซ้อน แต่สามารถปรับปรุงให้เข้ากับปัญหา เพื่อช่วยให้เจเนติก อัลกอริทึมมีแนวทางการค้นหาคำตอบที่ดีขึ้น ในบทต่อไปจะกล่าวถึงการประยุกต์ใช้เจเนติก อัลกอริทึม ในการแก้ปัญหาการออกแบบตัวกรองความถี่แบบ (Finite Impulse Response: FIR) โดยจะต้องมีวิธีการปรับรูปแบบโครโมโซม หรือตัวดำเนินการทางพันธุศาสตร์เพื่อให้เหมาะสม ซึ่งจะเป็นแนวทางในการประยุกต์ใช้เจเนติก อัลกอริทึมแก้ปัญหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่3

ทฤษฎีและหลักการทางวงจรกรองความถี่ดิจิตอล

การประมวลผลสัญญาณเชิงเลข (Digital Signal Processing)

การประมวลสัญญาณเชิงเลขหมายถึง การแทนสัญญาณต่าง ๆ ด้วยลำดับของตัวเลขหรือสัญลักษณ์ และทำการประมวลลำดับเหล่านี้ออกมา วิธีการประมวลผลอาจมีจุดประสงค์ต่าง ๆ กัน เช่น การประมาณคุณสมบัติของสัญญาณ หรือการแปลงสัญญาณหนึ่งให้ไปอยู่ในอีกรูปแบบหนึ่ง ซึ่งเหมาะสมในการพิจารณาวิเคราะห์มากกว่าสัญญาณเดิม

สามารถแบ่งได้เป็น 2 หัวข้อใหญ่ๆ คือ

1) **การกรองเชิงเลข (Digital Filtering)** การกรองเชิงเลข มีอยู่หลายประเภทด้วยกัน อย่างไรก็ตาม สำหรับในหัวข้องานวิจัยนี้ เราสนใจเฉพาะการกรองเชิงเลข แบบเป็นระบบเชิงชนิดตัวกรองไม่ป้อนกลับเชิงเลข (Non-recursive Digital Filter) ซึ่งเป็นวงจรพื้นฐานของวงจรกรองเชิงเลขแบบอื่นๆ รายละเอียดเกี่ยวกับคุณสมบัติ หลักการออกแบบ หลักการสร้าง จะได้กล่าวถึงในบทต่อๆไป

2) **การแปลงเชิงเลข (Digital Transform)** การแปลงเชิงเลขโดยเฉพาะการแปลงฟูริเยร์เต็มหน่วย (Discrete Fourier Transform หรือ DFT) ที่ทำการประมวลผลโดยใช้ขั้นตอนวิธีที่เรียกว่า การแปลงฟาสท์ฟูริเยร์ (Fast Fourier Transform หรือ FFT) มีส่วนทำให้การประมวลผลสัญญาณเชิงเลขได้รับความนิยมมากขึ้น

ตัวอย่างการนำไปประยุกต์ใช้งานของการประมวลผลสัญญาณเชิงเลข มีให้เห็นอยู่มากในปัจจุบัน อาทิเช่น เทคนิคการสื่อสารเชิงเลข ซึ่งอาจเป็น การมอดูเลต (Modulation) หรือการลงรหัส (Coding) สัญญาณเชิงเลข นอกจากนี้ยังมีใช้ในระบบควบคุมเชิงเลข (Digital Control System) และอุปกรณ์เครื่องมือวัดเชิงเลข เราอาจหมายถึง

ก. การประมวลผลเชิงเลขของสัญญาณใดๆที่ไม่ได้อยู่ในรูปแบบสัญญาณเชิงเลขหรือ

ข. การประมวลผล(ด้วยวิธีใดก็ตาม) ของสัญญาณที่อยู่ในรูปแบบสัญญาณเชิงเลขแล้ว

ความหมายในข้อ (ก) นั้นหมายถึง การประมวลผลสัญญาณในธรรมชาติทั่วไป ซึ่งสัญญาณในธรรมชาตินั้นปกติอยู่ในรูปแบบ สัญญาณเชิงอุปมาน (Analog Signal) ก่อนการประมวลผลเราต้องทำการแปลงสัญญาณเชิงอุปมานให้เป็นสัญญาณเชิงเลขเสียก่อน ส่วนความหมายในข้อ (ข) เป็นการประมวลผลสำหรับสัญญาณที่มาจากอุปกรณ์เชิงเลข เช่น คอมพิวเตอร์ หรือ ระบบควบคุมเชิงเลข สัญญาณจากอุปกรณ์เหล่านี้เป็นแบบสัญญาณเชิงเลขอยู่แล้ว จึงไม่จำเป็นต้องมีการแปลงรูปสัญญาณอีก

เหตุผลสำคัญที่ทำให้การประมวลผลสัญญาณเชิงเลขเริ่มได้รับความสนใจมากขึ้นก็เนื่องมาจากเทคโนโลยีในการสร้าง วงจรรวม หรือ วงจรไอซี (Integrated Circuit) สามารถสร้างเป็นวงจรรวมที่ผลิตบนสารกึ่งตัวนำชิ้นเดียวกันได้ และสามารถผลิตครั้งละจำนวนมากชิ้นได้ จึงทำให้ราคาต่อวงจรถูกลงเป็นอย่างมาก โดยเฉพาะอย่างยิ่งวงจรรวมแบบแอลเอสไอ (LSI=Large Scale Integrated Circuit) ที่สำคัญๆ เช่น ไมโครโปรเซสเซอร์ (Microprocessor) ซึ่งมีราคาถูกลงมากและมีผู้สนใจนำมาทำเป็นตัวประมวลสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มากขึ้น นอกจากนี้การประมวลสัญญาณเชิงเลขยังให้ความเชื่อถือได้และความแม่นยำในการคำนวณดีกว่าการประมวลผลสัญญาณเชิงอุปมานมาก

เนื่องจากการประมวลผลสัญญาณเชิงอุปมานได้มีการศึกษา วิจัย และมีการพัฒนากันมานาน ทำให้อุปกรณ์ และระบบวงจรที่ใช้กันอยู่ในปัจจุบันในรูปแบบของ การประมวลผลสัญญาณเชิงอุปมานเกือบทั้งหมด การที่จะออกแบบสร้างระบบประมวลผลสัญญาณใหม่ ให้อยู่ในรูปแบบของการประมวลผลสัญญาณเชิงเลข อุปกรณ์ใหม่นั้นก็ต้องสามารถเชื่อมโยงกับอุปกรณ์เดิม ซึ่งเป็นระบบการประมวลผลสัญญาณเชิงอุปมานได้ ดังนั้น ความพยายามของนักวิจัยสมัยแรก ก็คือ การวิจัย และศึกษาเพื่อหาวิธีการนำเอาอุปกรณ์เชิงเลขและอุปกรณ์เชิงอุปมานมาเชื่อมโยงกัน ดังสามารถเห็นได้จากสาขาที่แสดงถึงความสัมพันธ์กันนั้นได้แก่ สาขาวิชา ทฤษฎีโครงข่าย (Network Theory) กับทฤษฎีกราฟ (Graph Theory) สาขาวิชา การทำนายข้อมูล (Data Prediction) กับการอัดแถบความถี่ปฏิบัติงาน (Bandwidth Compression) สาขาวิชาการแปลงฟูริเยร์ (Fourier Transform) กับการแปลงอาดามาห์ (Hadasard Transform) หรือสาขาวิชาการกรองเชิงเลขกับการกรองเชิงอุปมาน เป็นต้น ซึ่งในแต่ละสาขาวิชาที่กล่าวถึงนี้เป็นที่รู้จักกันอย่างดี และมีการประยุกต์ใช้มากในปัจจุบัน

บทนิยามของสัญญาณและระบบ

ก่อนที่จะกล่าวถึงส่วนสำคัญต่างๆของระบบประมวลผลสัญญาณเชิงเลข เรามาทำความรู้จักกับคำนิยามของศัพท์เทคนิคบางคำเสียก่อน เพื่อให้การสื่อความหมายและความเข้าใจเป็นแบบอย่างเดียวกัน โดยที่นิยามของคำศัพท์เหล่านี้มาจากบทความเรื่อง Terminology in digital signal processing ตีพิมพ์ในวารสาร IEEE Transaction on Audio and Electro acoustic, vol. AU-20, pp.323-337, Dec 1972. ซึ่งนำเฉพาะคำที่สำคัญมากล่าวถึงคือ

1) สัญญาณเชิงอุปมาน ใช้กับการกล่าวถึงสัญญาณที่มีรูปคลื่น (Waveform) แปรไปอย่างต่อเนื่องกับพิสัยเวลา โดยที่แอมปลิจูด (Amplitude) หรือค่าขนาดของสัญญาณก็มีการแปรไปอย่างต่อเนื่องด้วย เช่น สัญญาณไซน์ หรือ sin (ω t) เป็นต้น คำว่า อุปมาน มีต้นกำเนิดมาจากการพัฒนาของคอมพิวเตอร์เชิงอุปมานที่เลียนแบบปริมาณทางฟิสิกส์ต่างๆ (ซึ่งเป็นปริมาณต่อเนื่องกับเวลา) ด้วย ค่าขนาดศักย์ หรือ กระแสไฟฟ้า และใช้อุปกรณ์เชิงอุปมาน เช่น วงจรขยายออป-แอมป์ (Op-Amp) มาช่วยในการเลียนแบบ

2) สัญญาณเชิงเวลาต่อเนื่อง (Continuous-Time Signal) สัญญาณแบบนี้รูปคลื่นของสัญญาณแปรค่าไปอย่างต่อเนื่องกับพิสัยของเวลา แต่แอมปลิจูดไม่ได้เจาะจงว่าจะต้องแปรไปอย่างต่อเนื่อง ซึ่งก็หมายถึงว่าอาจแปรอย่างไม่ต่อเนื่องก็ได้ เพราะฉะนั้นอาจกล่าวได้ว่า สัญญาณเชิงอุปมานเป็นชนิดหนึ่งของสัญญาณเชิงเวลาต่อเนื่องได้ อย่างไรก็ตามในหนังสือทั่วไปมักใช้คำทั้งสองนี้ปนเปกันไป ดังนั้นโดยทั่วไปการพูดถึงสัญญาณเชิงอุปมาน จึงหมายถึงสัญญาณเชิงเวลาต่อเนื่องด้วย

3) สัญญาณเชิงเวลาเต็มหน่วย (Discrete-Time Signal) เป็นสัญญาณ x (t) ที่ค่าของฟังก์ชันกำหนดเฉพาะเซตของเวลาที่แน่นอนอันหนึ่งเท่านั้น สัญญาณแบบนี้อาจแบ่งตามลักษณะของแอมพลิจูดได้เป็น 2 แบบคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 **สัญญาณเชิงข้อมูลเต็มหน่วย (Discrete-Time Signal)** สัญญาณแบบนี้แอมปลิจูดจะต่อเนื่องหรือกล่าวได้ว่า แอมปลิจูดมีค่าเท่ากันทุกประการกับสัญญาณเชิงอุปมาน ที่เป็นตัวต้นแบบในการสุ่มตัวอย่าง (Sampling) ตัวอย่างเช่น สัญญาณออกของวงจรสุ่มและค่าคงสัญญาณ จะจัดเป็นสัญญาณเชิงเต็มหน่วย สัญญาณแบบนี้อาจเรียกว่า ข้อมูล หรือ สัญญาณเชิงเต็มหน่วย

3.2 **สัญญาณเชิงเลข** สัญญาณแบบนี้แอมปลิจูดของสัญญาณมีค่าเฉพาะเซตของค่าที่แน่นอนเซตหนึ่งเท่านั้น เช่นสัญญาณที่ออกจากวงจร A/D เป็นต้น

4) **ระบบเวลาจริง (Real-Time System)** โดยทั่วไปคำนี้ใช้กับ ระบบการประมวลผลสัญญาณที่ผลในการคำนวณประมวลผลทำได้เสร็จสิ้น ก่อนจะมีการสุ่มตัวอย่างสัญญาณลำดับใหม่ อย่างไรก็ตามคาบเวลาในการสุ่มตัวอย่างสัญญาณควรมีค่าน้อย ไม่ควรมีค่า วัน เดือน หรือ ปี

5) ลำดับ (Sequence) หนังสือบางเล่มอาจเรียกว่าเป็นลำดับข้อมูล หรือลำดับสัญญาณ หรือลำดับก็ได้ ซึ่งคำว่าลำดับอาจใช้แทนสัญญาณเชิงเต็มหน่วย หรือสัญญาณเชิงเลขก็ได้

3.1ทฤษฎีดิจิตอลซิกแนลโปรเซสซิ่ง

คำว่าสัญญาณ (Signal)เราจะให้คำจำกัดความว่า คือฟังก์ชันที่แทนความหมายข้อมูล (data) หรือข่าวสาร (Information) เกี่ยวกับภาวะของการประพฤติตัวของระบบทางกายภาพ (Physical System) สัญญาณอาจเขียนอยู่ในรูปที่ตัวแปรเป็นแกนเวลา ส่วนสัญญาณรูปภาพจะใช้ระบบมิติเป็นตัวแปร (Spatial Variables)

ตัวแปรอิสระที่ใช้เขียนเป็นคณิตศาสตร์แทนความหมายของสัญญาณ อาจจะเป็นตัวแปรที่ต่อเนื่อง (Continuous) หรือไม่ต่อเนื่อง (Discrete) ก็ได้ ดังนั้นการประมวลสัญญาณจึงแบ่งชนิดสัญญาณออกได้สองชนิด คือ

1. สัญญาณที่มีความต่อเนื่องทางแกนเวลาหรือสัญญาณอนาลอก (Continuous or Analog Signal) คือสัญญาณที่มีความต่อเนื่องทั้งในแนวแกนเวลาและขนาดของสัญญาณ คณิตศาสตร์พื้นฐานที่ใช้วิเคราะห์คือ สมการอนุพันธ์เชิงเส้น โดยใช้ลาปลาซทรานฟอร์มช่วยแก้ปัญหา
2. สัญญาณที่ไม่มีความต่อเนื่องทางแกนเวลาหรือสัญญาณดิจิตอล (Discrete or Digital Signal) คือสัญญาณที่ไม่ต่อเนื่องกันทั้งในแนวแกนเวลาและขนาดของสัญญาณ คณิตศาสตร์พื้นฐานที่ใช้วิเคราะห์คือ สมการแตกต่างเชิงเส้น โดยใช้ Z- ทรานฟอร์มช่วยแก้ปัญหา

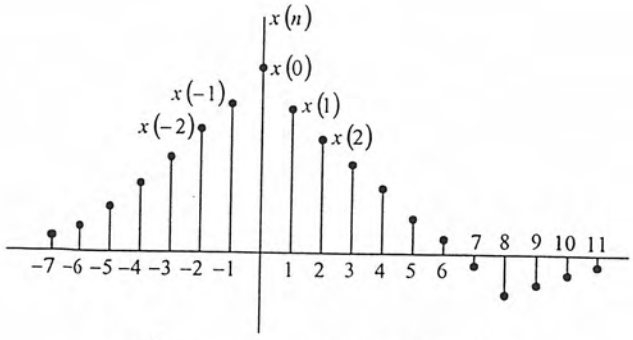
ระบบและสัญญาณที่ไม่ต่อเนื่องตามแกนเวลา (Discrete – Time Signal and Systems)

1. **สัญญาณที่ไม่ต่อเนื่องตามแกนเวลา (Discrete – Time Signal Sequences)**

ในระบบที่ไม่ต่อเนื่องทางแกนเวลาจะกล่าวถึงการประมวลสัญญาณ ซึ่งแทนด้วยลำดับตัวเลข (sequences) และลำดับตัวเลข X ตัวที่ n เขียนแทนด้วย X(n) ซึ่งเขียนเป็นรูปแบบทางคณิตศาสตร์ได้

$$X = \{X(n)\} \quad , \quad -\infty < n < \infty \quad \text{.....(3.1)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3. 1 กราฟแสดงสัญญาณไม่ต่อเนื่อง

สัญญาณหรือลำดับตัวเลขแทนด้วยกราฟได้ดังตัวอย่างในรูปที่ 3.1 แม้ว่าแกนแอบซิสซ่า (Abscissa) เป็นเส้นที่ต่อเนื่อง แต่จริง ๆ แล้ว X(n) มีความหมายเฉพาะค่าจำนวนเต็ม (Integer) ของ n ส่วนค่า n ที่ไม่ใช่ตัวเลขจำนวนเต็มไม่ได้หมายความว่าลำดับ X(n) จะมีค่าเป็นศูนย์ เพียงแต่ว่าเราไม่ได้กำหนดค่า X(n) เมื่อ n ไม่ใช่ตัวเลขจำนวนเต็ม

ตัวอย่างลำดับตัวเลขหรือสัญญาณที่สำคัญ ๆ แสดงดังรูปที่3.2

รูปที่ 3.2 แสดงตัวอย่างของลำดับสัญญาณ ซึ่งมีความหมายในการวิเคราะห์และการแทนความหมายในระบบและสัญญาณที่ไม่ต่อเนื่องตามแกนเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

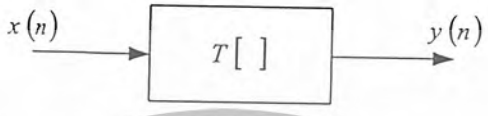
2. ระบบชีฟท์- อินแวเรียนท์เชิงเส้น (Linear Shift-Invariant Systems)

ระบบ (System) มีความหมายทางคณิตศาสตร์ว่าเป็นการแปลงหรือเป็นตัวกระทำ ซึ่งเปลี่ยนลำดับของสัญญาณเข้า X (n) ไปเป็นลำดับของสัญญาณออก Y(n) เขียนได้เป็น

$$Y(n) = T[X(n)] \qquad \text{.....(3.2)}$$

T เป็นตัวแปลง (Transform) ทางคณิตศาสตร์

สมการที่ (3.2) เขียนบล็อกไดอะแกรมได้ดังรูปที่ 3.3



รูปที่ 3.3 แสดงการแปลงลำดับสัญญาณเข้า X(n) ไปเป็นสัญญาณออก Y(n)

สำหรับระบบเชิงเส้น (Linear Systems) ถูกกำหนดด้วยรูปแบบทางคณิตศาสตร์ดังนี้

$$T[ax_1(n) + bx_2(n)] = aT[x_1(n)] + bT[x_2(n)]$$
$$= ay_1(n) + by_2(n) \qquad \text{.....(3.3)}$$

เมื่อ $y_1(n)$ และ $y_2(n)$ เป็นผลตอบสนองของสัญญาณเข้า $x_1(n)$ และ $x_2(n)$ a และ b เป็นค่าคงที่ใด ๆ

ส่วนระบบชีฟท์- อินแวเรียนท์ มีคุณสมบัติว่า ถ้า $y(n)$ เป็นผลตอบสนองของ $x(n)$ และ $x(n-k)$เมื่อ k เป็นเลขบวกหรือลบก็ได้

ความสัมพันธ์ระหว่างลำดับสัญญาณเข้าและสัญญาณออกของระบบชีฟท์- อินแวเรียนท์เชิงเส้น เขียนได้เป็น

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k)$$

หรือ

$$= \sum_{k=-\infty}^{\infty} h(k)x(n-k) \qquad \text{.....(3.4)}$$

เมื่อ $h(n)$เป็นผลตอบนองทางอิมพัลส์ของระบบ สมการที่ (3.4) เรียกว่า ผลรวมของคอนโวลูชั่น (Convolution Sum)

3. เสถียรภาพและภาวะเหตุ (Stability and Causality)

เรากำหนดระบบที่มีเสถียรภาพว่าเมื่อป้อนสัญญาณเข้าที่มีขอบเขตจะได้สัญญาณออกที่มีขอบเขตเช่นกัน ซึ่งระบบจะมีเสถียรภาพก็ต่อเมื่อ

$$S \overset{\Delta}{=} \sum_{k=-\infty}^{\infty} |h(k)| < \infty \qquad \text{.....(3.5)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบตามภาวะเหตุ (Causal System) หมายความว่า การเปลี่ยนแปลงที่สัญญาณออกจะไม่เกิดก่อนการเปลี่ยนแปลงที่สัญญาณเข้า สำหรับระบบชีฟท์- อินแวเรียนท์ที่เป็นระบบตามภาวะเหตุผลตอบสนองทางอิมพัลส์เป็นศูนย์เมื่อ $n < 0$

4. สมการแตกต่างเชิงเส้นสัมประสิทธิ์คงที่ (Linear Constant Coefficient Difference Equations)

อินแวเรียนท์สามารถเขียนแทนด้วยสมการแตกต่างเชิงเส้นสัมประสิทธิ์คงที่อันดับที่ N ดังนี้

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

หรือ $$y(n) = \sum_{r=0}^{M} b_r x(n-r) - \sum_{k=1}^{N} a_k y(n-k) \quad \text{.....(3.6)}$$

$b_0, b_1, ..., b_M$ และ $a_0, a_1, ..., a_N$ เป็นค่าคงที่ที่ใช้ในการกำหนดคุณสมบัติ (characteristic) ของระบบ สมการที่ (6) ตับแปรอยู่ในแนวแกนเวลา แต่สามารถหาความสัมพันธ์ในแกนความถี่ได้ด้วย Z – ทรานส์ฟอร์ม (Z – Transform)

5. การแทนระบบสัญญาณที่ไม่ต่อเนื่องทางแกนเวลาด้วยแกนความถี่ (Frequency Domain Representation of Discrete-Time Systems and Signals)

สมมติเราป้อนสัญญาณเข้าด้วย $x(n) = e^{j\omega n}$ เมื่อ $-\infty < n < \infty$ เราสามารถหาผลตอบสนองของสัญญาณออกด้วยสมการที่ (3.4) ได้

$$y(n) = \sum_{k=-\infty}^{\infty} h(k) e^{j\omega(n-k)}$$

$$= e^{j\omega n} \sum_{k=-\infty}^{\infty} h(k) e^{-j\omega k} \quad \text{.....(3.7)}$$

ถ้าเรากำหนด

$$H(e^{j\omega}) = \sum_{k=-\infty}^{\infty} h(k) e^{-j\omega k} \quad \text{.....(3.8)}$$

เราจะได้

$$y(n) = H(e^{j\omega}) e^{j\omega n} \quad \text{.....(3.9)}$$

จากสมการที่ (3.9) จะเห็นว่า $H(e^{j\omega})$ เป็นฟังก์ชันของความถี่ ω เราจะเรียก $H(e^{j\omega})$ ว่าผลตอบสนองทางความถี่ (Frequency Response) ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. Z – TRANSFORM

Z-Transform เป็นการแปลงอีกชนิดหนึ่งซึ่งนักออกแบบตัวกรองสัญญาณทั้งหลายนิยมใช้อ้างถึง วิธีการนี้จะเป็นการแปลงลำดับซึ่งเป็นตัวเลขที่เขียนเรียงกันให้เป็นฟังก์ชันของตัวแปรเชิงซ้อน (Complex Variable) หรือกล่าวง่ายๆคือการเอาลำดับชุดหนึ่งไปสร้างฟังก์ชันตามกฎเกณฑ์ทางคณิตศาสตร์ที่กำหนดไว้ ดังนั้นคุณสมบัติต่างๆที่มีอยู่ในลำดับจะถูกส่งเข้าไปอยู่ในฟังก์ชันที่สร้างขึ้นด้วย และจะใช้อักษร Z ในภาษาอังกฤษเป็นสัญลักษณ์ของตัวแปรในฟังก์ชันที่สร้างขึ้น

Z-Transform เป็นสิ่งที่สำคัญที่สุดอย่างหนึ่งในระบบดิจิตอลซิกแนลโปรเซสซิ่ง นั่นคือใช้ในการออกแบบเพื่อวิเคราะห์สัญญาณ Error ใน ตัวกรองความถี่ดิจิตอล โดยเฉพาะในระบบตัวกรองความถี่ดิจิตอลที่มีอิมพัลส์ไม่จำกัด (IIR Filter) ซึ่งมักจะถูกใช้ในการคำนวณหาค่าสัมประสิทธิ์ของตัวกรองความถี่ดิจิตอล และวิเคราะห์ผลของการ Quantization Errors คือใช้ในการแสดงโครงสร้างของตัวกรองความถี่ดิจิตอล

ลำดับสัญญาณ $x(0), x(1), ..., x(n)$ เป็นสัญญาณที่ไม่ต่อเนื่องทางแกนเวลาสามารถแปลงให้อยู่ในรูปแกนความถี่เป็น $X(z)$ z เป็นตัวแปรคอมเพล็ก (Complex Variable) หรือแปลงสัญญาณจากแกนความถี่ไปอยู่ในแกนเวลาด้วยการใช้ความสัมพันธ์

$$X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n} \quad \text{..... (3.10)}$$

$$x(k) = \frac{1}{2\pi j} \oint X(z) z^{k-1} dz \quad \text{..... (3.11)}$$

สมการที่ (3.10) เรียกว่า Z – ทรานส์ฟอร์ม ส่วนสมการที่ (3.11) เรียกว่า อินเวอร์ส Z – ทรานส์ฟอร์ม โดยที่ $z = e^{j\omega}$

คุณสมบัติที่สำคัญของ Z – ทรานส์ฟอร์ม (Elementary Properties of Z- Transform)

1. ความเป็นเชิงเส้น (Linearity)

   Z – ทรานส์ฟอร์มของ $\{ax(n) + by(n)\} = aX(z) + bY(z)$

2. การเลื่อน (Shifts)

   Z – ทรานส์ฟอร์มของ $\{x(n+k)\} = z^k X(z)$

3. คอนโวลูชัน (Convolution)

   ถ้า $g(m) = \sum_{m=-\infty}^{\infty} x(m) y(n-m)$

   แล้ว $G(z) = X(z)Y(z)$

ความสัมพันธ์ระหว่าง S-plane และ z-plane

จากข้อกำหนดที่ให้ $e^{-ST} = z^{-1}$

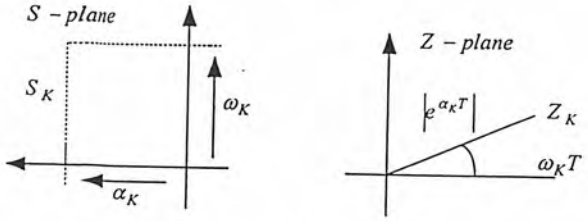
จะเขียนได้เป็น $Z = e^{(\alpha - j\omega)T}$

จะพบว่า

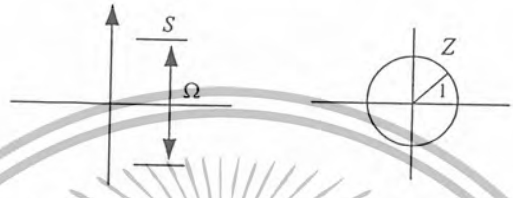
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$|Z| = e^{(\omega T)} \qquad \angle Z = \omega T$$

ซึ่งเมื่อนำมาเขียนในรูป s และ z plane จะได้ดังรูปที่ (3.4)



(ก) ความสัมพันธ์ของ S_K และ Z_K



(ข) วงกลมหนึ่งหน่วยใน Z- plane

รูปที่3.4 แสดงความสัมพันธ์ระหว่าง s-plane และ z-plane

ก. แสดงความสัมพันธ์ของขนาด

ข. แสดงถึงวงกลมหนึ่งหน่วยใน z-plane

จาก $|Z| = e^{\alpha i}$

ถ้า $\alpha = 0$ แล้ว $Z = 1$ จะได้วงกลมหนึ่งหน่วยในรูป ข. ส่วนของ Ω ทั้งหมดจะปรากฏก็จะปรากฏในพื้นที่ของวงกลม 1 หน่วย ใน z-plane

รูปแบบโครงสร้างของวงจรกรองความถี่ดิจิตอล (Network Structures for Digital Filter)

การสร้างวงจรกรองความถี่ดิจิตอลไม่ว่าวิธีฮาร์ดแวร์ หรือซอฟท์แวร์จะต้องเขียนความสัมพันธ์ระหว่างสัญญาณเข้า – สัญญาณออกให้อยู่ในรูปแบบหรืออัลกอริทึมที่ใช้คำนวณก่อน อัลกอริทึมนี้จะใช้ตัวบวก (Adder) ตัวหน่วงเวลา (Delay) และตัวคูณด้วยค่าคงที่ (Constant Multiplier) มาช่วยเขียนโครงสร้างของวงจร ตัวอย่างเช่นเราพิจารณาระบบซึ่งมีทรานส์เฟอร์ฟังก์ชันในรูป

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 - \sum_{k=1}^{N} a_k z^{-k}} = \frac{Y(z)}{X(z)} \qquad \text{.....(3.12)}$$

เราจะเขียนสมการแตกต่างได้เป็น

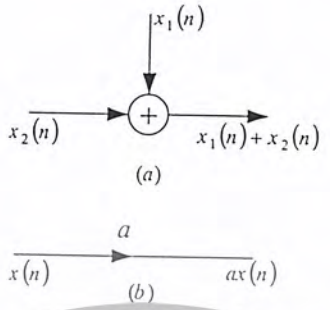
$$y(n) = \sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k) \qquad \text{.....(3.13)}$$

เราสามารถแทนสมการที่ (3.12) ด้วยอัลกอริทึมในการคำนวณโดยตรงนั่นคือเอาค่าหน่วงเวลาของสัญญาณคูณด้วยสัมประสิทธิ์ a_k แล้วนำผลคูณทั้งหมดเอามาบวกกัน

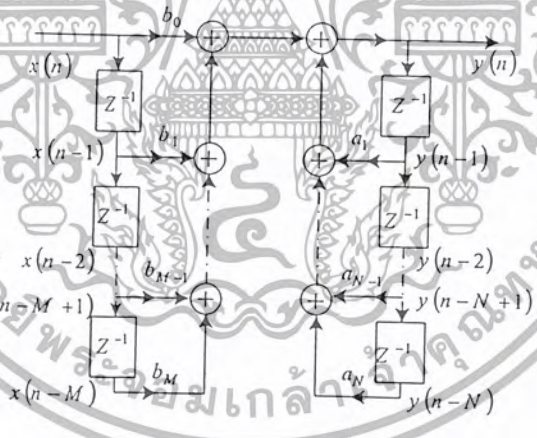
สัญลักษณ์ส่วนใหญ่ที่ใช้แทนสมการแตกต่าง เช่น ตัวบวก ตัวคูณ ตัวหน่วงเวลาแสดงดังรูปที่ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับสมการที่ (3.13) เมื่อนำมาเขียนเป็นวงจรจะได้ดังรูปที่ (3.6) ส่วนวิธีการเขียนบล็อกไดอะแกรมเพื่อแทนวงจรกรองความถี่ในแบบและชนิดต่าง ๆ ก็มีเทคนิคการเขียนแตกต่างกันออกไป แล้วแต่ความเหมาะสม เช่นในด้านความประหยัด ความรวดเร็วในการคำนวณ หรือความเที่ยงตรงในการคำนวณเป็นต้น



รูปที่ 3.5 สัญลักษณ์และบล็อกไดอะแกรมสำหรับวงจรดิจิตอล
(a) การบวกลำดับสัญญาณสองสัญญาณ
(b) การคูณลำดับสัญญาณด้วยค่าคงที่
(c) ค่าหน่วงหนึ่งหน่วย



รูปที่ 3.6 บล็อกไดอะแกรมสำหรับแทนสมการแตกต่างทั่ว ๆ ไป อันดับที่ N

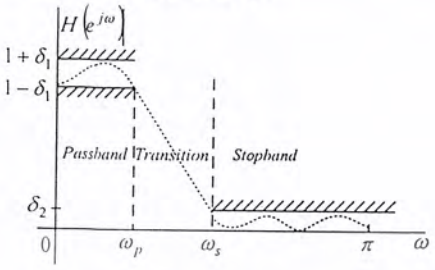
การประมาณในการออกแบบวงจรกรองความถี่ดิจิตอล (Approximation of Digital Filter)

วงจรกรองความถี่ดิจิตอลแบ่งออกได้เป็นสองแบบคือ แบบไม่ป้อนกลับ (Non – Recursive) และแบบป้อนกลับ (Recursive) ทั้งสองแบบนี้มีเทคนิคในการประมาณค่าที่คล้ายกัน บางเทคนิคก็แตกต่างกันออกไป ส่วนการกำหนด (Specification) ในการออกแบบขึ้นอยู่กับการนำไปใช้งาน เช่นอาจต้องการผลตอบสนองทางเฟส (phase) ที่เป็นเชิงเส้น (linear) หรืออาจต้องการผลตอบสนองทางขนาดเป็นแบบที่มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยอดคลื่นเท่ากัน (Equip Ripple) สิบยอดในแถบความถี่ต่ำผ่าน และแถบปลอดสัญญาณให้มีขนาดราบเรียบมากที่สุด (Maximally flat) ฯลฯ

รูปตัวอย่างในการออกแบบผลตอบสนองทางขนาดในรูปที่ 3.7 ซึ่งเป็นการออกแบบให้ความถี่ต่ำผ่าน โดยที่ช่วงกว้างของผลตอบสนองทางขนาดไม่เกินข้อกำหนดที่ให้



รูปที่ 3.7 แสดงการประมาณวงจรกรองความถี่ในอุดมคติในช่วงกว้างที่กำหนดขอบเขตให้

เส้นโค้งประ แทนผลตอบสนองทางความถี่ของระบบ ซึ่งออกแบบให้ตรงข้อกำหนดได้อย่างแน่นอน ในกรณีในแถบความถี่ผ่านค่าผลตอบสนองทางขนาดให้ประมาณเท่ากับ 1 และมีความผิดพลาด (error) ไม่เกิน $\pm\delta_1$ เขียนในรูปคณิตศาสตร์ได้ว่า

$$1-\delta_1 \leq \left|H\left(e^{j\omega}\right)\right| \leq 1+\delta_1 \quad , \quad |\omega| \leq \omega_p$$

แถบปลอดสัญญาณซึ่งผลตอบสนองทางขนาดให้ประมาณเท่ากับศูนย์และให้ผิดพลาดได้น้อยกว่า δ_2 เขียนได้เป็น

$$\left|H\left(e^{j\omega}\right)\right| \leq \delta_2 \quad , \quad \omega_s \leq |\omega| \leq \pi$$

โดยที่ ω_p เป็นจุดกำจัดความถี่ในแถบสัญญาณผ่าน ω_s เป็นจุดกำจัดความถี่ในแถบปลอดสัญญาณทั้งสองจุดเป็นมุมในระนาบ Z ส่วนในแถบ $(\omega_p - \omega_s)$ ให้มีค่ามากกว่า 0 เรียกว่า แถบทรานซิชัน (Transition Band) แถบนี้ผลตอบสนองทางขนาดลดลงอย่างราบเรียบ จากแถบความถี่ผ่านจนถึงแถบปลอดสัญญาณ

วงจรกรองความถี่ดิจิตอลแบบไม่ป้อนกลับ (Nonrecursive Digital Filter)

วงจรกรองความถี่ชนิดนี้สามารถออกแบบให้มีเฟสลักษณะเป็นเชิงเส้นโดยวิธีง่าย ๆ แล้วยังมีเสถียรภาพ (Stable) เสมอ ในที่นี้ผู้เขียนขออธิบายคุณสมบัติของเฟสที่เป็นเชิงเส้นของวงจรกรองความถี่ดิจิตอลแบบไม่ป้อนกลับ เพื่อสะดวกในการอ้างอิงบทต่อไป

ทรานส์เฟอร์ฟังก์ชัน แบบนอนรีเคอร์ซีฟเขียนได้เป็น

$$\begin{aligned} H(z) &= \sum_{n=0}^{N-1} h(n)\, z^{-n} \\ &= h(0) + h(1)z^{-1} + K + h(N-1)z^{-(N-1)} \end{aligned} \quad \text{.....(3.14)}$$

ผลตอบสนองทางความถี่ของสมการที่ (3.14)

$$H\left(e^{j\omega}\right) = \sum_{n=0}^{N-1} h(n)\, e^{-j\omega n}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$H\left(e^{j\omega}\right) = \sum_{n=0}^{N-1} h(n)\, e^{-j\omega n}$$

$$= \left|H\left(e^{j\omega}\right)\right| e^{j\theta(\omega)} \qquad \text{.....(3.15)}$$

เมื่อ $\left|H\left(e^{j\omega}\right)\right|$ คือผลตอบสนองทางขนาด $\theta(\omega)$ คือมุมเฟส

เฟสและกรุ๊ฟดีเลย์ (Group Delay) มีคำจำกัดความว่า

$$\tau_p = -\frac{\theta(\omega)}{\omega} \quad \text{และ} \quad \tau_g = -\frac{d\theta(\omega)}{d\omega} \qquad \text{.....(3.16)}$$

ถ้าจะมีเฟสลักษณะเป็นเชิงเส้นต้องมีเงื่อนไขว่า

$$\theta(\omega) = -\tau\omega \quad , \quad \tau \text{ เป็นค่าคงที่}$$

ดังนั้นจากสมการที่ (3.15) จะเขียนให้อยู่ในรูปของเฟสได้

$$\theta(\omega) = -\tau\omega = \tan^{-1}\frac{-\sum_{n=0}^{N-1} h(n)\sin\omega n}{\sum_{n=0}^{N-1} h(n)\cos\omega n} \qquad \text{.....(3.17)}$$

หรือ

$$\tan\omega\tau = \frac{\sum_{n=0}^{N-1} h(n)\sin\omega n}{\sum_{n=0}^{N-1} h(n)\cos\omega n} \qquad \text{.....(3.18)}$$

หรือ

$$\tan\omega\tau = \frac{\sum_{n=0}^{N-1} h(n)\sin\omega n}{h(0)+\sum_{n=1}^{N-1} h(n)\cos\omega n} \qquad \text{.....(3.19)}$$

พิจารณาจากสมการที่ (3.19) จะมีสองกรณีที่เป็นไปได้คือ กรณีที่หนึ่ง เมื่อ $\tau = 0$ สมการที่ (3.19) จะเป็น

$$0 = \frac{\sum_{n=0}^{N-1} h(n)\sin\omega n}{h(0)+\sum_{n=1}^{N-1} h(n)\cos\omega n} \qquad \text{.....(3.20)}$$

สมการที่ (3.20) จะเป็นจริงได้เมื่อ $h(n)=0$ และ $n \neq 0$ ส่วน $h(0)$ แม้จะมีค่าแต่ก็ไม่มีประโยชน์ เพราะผลตอบสนองออกมาเป็นแบบอิมพัลส์ เหลืออีกกรณีหนึ่งคือเมื่อ $\tau \neq 0$ สมการที่ (3.18) เขียนใหม่ได้เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\sum_{n=0}^{N-1} h(n)\cos(\omega n)\sin(\omega\tau) - \sum_{n=0}^{N-1} h(n)\sin(\omega n)\cos(\omega\tau) = 0 \quad \text{.....(3.21)}$$

หรือรวมด้านซ้ายมือของสมการที่ (3.21) ได้เป็น

$$\sum_{n=0}^{N-1} h(n)\sin\left[(\tau - n)\omega\right] = 0 \quad \text{.....(3.22)}$$

สมการที่ (3.22) จะเป็นจริงได้เมื่อมีเงื่อนไข

$$\tau = \frac{(N-1)}{2} \quad \text{.....(3.23)}$$

$$h(n) = h(N-1-n) \qquad 0 \le n \le N-1$$

ในกรณีที่ต้องการให้ผลตอบสนองทางความถี่มีรูปแบบเป็น

$$H(e^{j\omega}) = \left|H(e^{j\omega})\right| e^{j(\beta-\tau\omega)}$$

และให้เฟสเป็นเชิงเส้น เราสามารถหาได้ว่า

$$\tau = \frac{(N-1)}{2}$$

$$\beta = \pm\frac{\pi}{2}$$

$$h(n) = -h(N-1-n) \qquad 0 \le n \le N-1 \quad \text{.....(3.24)}$$

เมื่อหาซีโร่ (Zero) ของสมการที่ (3.24) โดยใช้เงื่อนไขที่ (3.23) และ (3.24) จะได้ซีโร่ซึ่งมีลักษณะเป็นกระจกเงา (Mirror – Image) ในแนวแกน X และจะเรียกโพลีโนเมียลในลักษณะนี้ว่า โพลีโนเมียลกระจกเงา (Mirror – Image Polynomial)

เทคนิคการออกแบบวงจรกรองความถี่แบบไม่ป้อนกลับที่รู้จักกันดีมี

ก. วิธีการใช้วินโดว์ฟังก์ชัน (Window Function)

ข. การแซมปลิ้งความถี่ (Frequency Sample)

ค. วิธีออพติไมเซชัน (Optimization Technique)

2. วงจรกรองความถี่ดิจิตอลแบบป้อนกลับ (Recursive Digital Filter)

วงจรกรองความถี่ชนิดนี้มีข้อได้เปรียบกว่าแบบนอนรีเคอร์ซีฟที่ค่าผลตอบสนองทางขนาดมีความคมกว่าในแถบทรานส์ชิชัน เมื่ออันดับ (Order) ของวงจรมีจำนวนเท่ากัน แต่เทคนิคการออกแบบค่อนข้างซับซ้อนเมื่อต้องการให้เฟสมีลักษณะเป็นเชิงเส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทคนิคการออกแบบวงจรกรองความถี่แบบป้อนกลับที่นิยมใช้กันมี

ก. วิธีการออกแบบจากวงจรกรองแบบอนาลอก **(Design of Recursive Digital Filter From Analog Filter)** โดยใช้เทคนิคอิมพัลส์อินแวเรียนซ์ (Impulse Invariance) และเทคนิคการแปลงแบบไบลีเนียร์ (Bilinear Transform)

ข. วิธีออพติไมเซชัน

สรุปแล้วการออกแบบจากวงจรกรองความถี่ดิจิตอลทั้งสองแบบมีวิธีการใหญ่ ๆ สองวิธีคือ การออกแบบให้อยู่ในรูปสูตรสำเร็จ (Closed – Form Solution) และการออกแบบโดยใช้คอมพิวเตอร์ช่วย (Computer Aided Designed)

ข. การสร้างวงจรกรองความถี่ดิจิตอล **(Implementation of Digital Filter)**

วงจรกรองความถี่ดิจิตอลสร้างได้ 2 รูปแบบคือ

1. วิธีซอฟท์แวร์ **(Software)**

   สร้างโดยการเลียนแบบ (Simulation) โครงสร้างวงจรดิจิตอล ด้วยดิจิตอลคอมพิวเตอร์ คือการเขียนโปรแกรมทรานส์เฟอร์ฟังก์ชันของวงจรกรองความถี่ดิจิตอลเก็บไว้ นำเอาสัญญาณดิจิตอลที่จะกรองป้อนเข้าไปทำการคำนวณกับโปรแกรมของทรานส์เฟอร์ฟังก์ชัน ผลลัพธ์ที่ได้คือ สัญญาณดิจิตอลที่ผ่านวงจรกรองความถี่ดิจิตอลนั่นเอง

2. วิธีฮาร์ดแวร์ **(Hardware)**

   โดยการนำเอาอุปกรณ์ทางดิจิตอลเช่น ชิฟรีจีสเตอร์ (Shift – Registe) ตัวบวก (Adder) ตัวลบ (Subtracter) และตัวคูณ (Multiplier) มาต่อเป็นวงจร

ทฤษฎีการแซมปลิ้ง (Sampling Theorem)

ในการแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอลนั้น จะต้องมีการสุ่มตัวอย่างของสัญญาณนั้น เข้ามาและเพื่อให้ได้สัญญาณที่มีคุณสมบัติครบตามสัญญาณต้นแบบนั้น ๆ จะต้องทำการสุ่มตาม Uniform Sampling Theorem โดยจะต้องมีความถี่ในการสุ่มอย่างน้อย 2 เท่า ของความถี่สูงสุดของสัญญาณนั้น ๆ ช่วงเวลา T (Sampling Interval) จะต้องไม่มากเกินกว่า $T = \frac{1}{2f_m}$ ซึ่งช่วงกว้างนี้เรียกว่า Nyquist Interval

The bilinear transformation

ในการหาความสัมพันธ์ระหว่างวงจรกรองความถี่แบบอนาลอกและดิจิตอล สามารถหาได้โดยการใช้วิธีของ Bilinear Transformation ซึ่งมีสมการเป็น

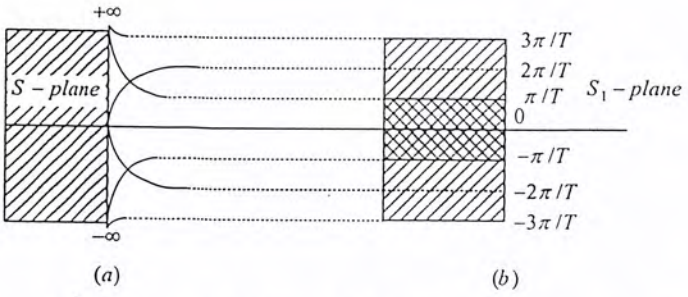
$$s = \frac{z-1}{z+1}$$

ให้ $s = j\omega_A$ A หมายถึง Analog

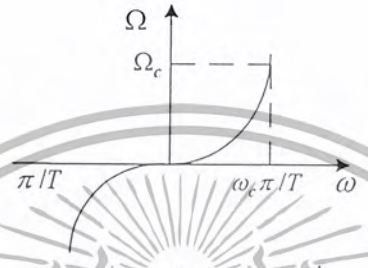
$z = e^{j\omega_D T}$ D หมายถึง Digital

แก้สมการแสดงรูปการ Mapping ได้ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 แสดงการ Mapping ของ Bilinear Transformation



รูปที่3.9 แสดงความสัมพันธ์ในแกนความถี่โดยวิธีไบลีเนียร์

3.2 หลักการเบื้องต้นของตัวกรองความถี่ดิจิตอล

สัญญาณเต็มหน่วย (Discrete-Time Signal) ที่ได้จากการวัดอาจจะมีสัญญาณอย่างอื่นที่ไม่ต้องการ ซึ่งเรียกว่า สัญญาณรบกวน (Noise) ปะปนผสมเข้ามาด้วย จึงจำเป็นต้องนำสัญญาณนี้ผ่านเข้าไปในอุปกรณ์อย่างหนึ่งซึ่งเรียกว่า ตัวกรองสัญญาณซึ่งทำหน้าที่กรองเอาสัญญาณรบกวนออก ที่ด้านออกของตัวกรองเราจะได้สัญญาณที่ต้องการ ในการกรองสัญญาณนั้นก่อนอื่นจะต้องรู้จักคุณลักษณะเฉพาะตัวของสัญญาณที่ต้องการจะวัดและของสัญญาณรบกวนที่เกิดขึ้น เพื่อให้สามารถสร้างตัวกรองสัญญาณที่มีคุณสมบัติยอมให้เฉพาะสัญญาณที่ต้องการผ่านออกมาด้านนอกได้เท่านั้น กระบวนการกรองสัญญาณเต็มหน่วยนี้หากใช้กระบวนการทางดิจิตอลเช่น ใช้คอมพิวเตอร์จะเรียกว่า ตัวกรองสัญญาณดิจิตอล

ตัวกรองสัญญาณดิจิตอลเป็นสิ่งที่สำคัญมากที่สุดอย่างหนึ่งในการนำไปใช้ในระบบดิจิตอลซิกแนลโปรเซสซิ่ง เป็นการนำ Algorithm ทางคณิตศาสตร์มาใช้ในระบบ Hardware หรือ Software กล่าวคือ สัญญาณอินพุตดิจิตอลจะทำการสร้างสัญญาณเอาต์พุตโดยใช้ Filtering Algorithm เพื่อให้ได้รับสัญญาณจากการ Filtering ตามความต้องการเพื่อนำไปใช้ในตัวกรองสัญญาณดิจิตอล สำหรับชนิดของตัวกรองสัญญาณ ที่สำคัญจะถูกกำหนดโดย

$$y(n) = \sum_{k=0}^{\infty} h(k)x(n-k) \qquad (3.25)$$

โดยที่ h(k) คือ สัมประสิทธิ์ของ Filter

k คือ 0,1,2,...,n-1

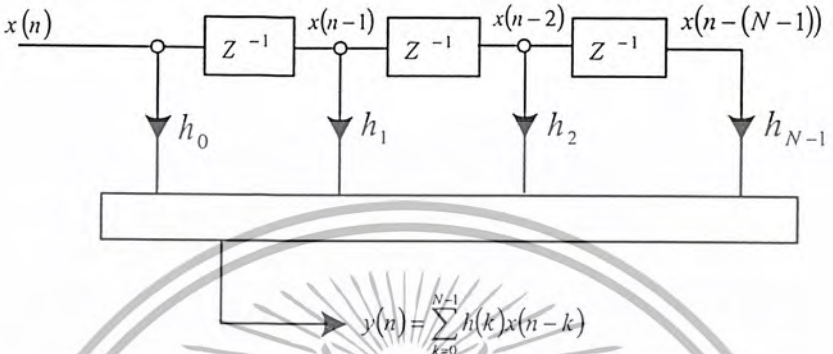
x(n) คือ Input Filter

y(n) คือ Output Filter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งสำหรับตัวกรองสัญญาณที่ให้มา ค่าสัมประสิทธิ์จะกำหนด Characteristic ของตัวกรองสัญญาณ Filtering นี้จะขึ้นอยู่ในการคอนโวลูชั่นที่แท้จริงของสัญญาณ และการตอบสนองต่ออิมพัลส์ของตัวกรองสัญญาณใน Time Domain ซึ่งก็คือ h(k)

รูปที่จะกล่าวถึงแสดงถึงบล็อกไดอะแกรม ซึ่งจะแสดงนิยามของตัวกรองความถี่ดิจิตอลที่กำหนดไว้ก่อนหน้านี้ โดยที่รูปแบบตัวกรองสัญญาณชนิดนี้เป็นที่รู้จักอย่างแพร่หลาย เรียกว่า Transversal Filter จากรูป 3.10 z^{-1} จะแสดงถึง Delay ของช่วงเวลาในการสุ่มตัวอย่างหนึ่งครั้ง

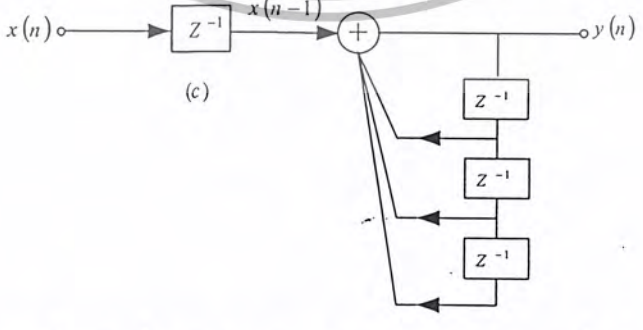


รูปที่ 3.10 บล็อกไดอะแกรมของ Transversal Filter โดยที่มี h(k), (k=0,1,2,...,n-1)

วัตถุประสงค์ของตัวกรองสัญญาณโดยรวม ๆ แล้วเป็นการกำจัดสัญญาณรบกวน จากสัญญาณที่ต้องการดังตัวอย่างดังรูปโดยการดึงข้อมูลของกลุ่มที่สนใจในช่วงหนึ่งออกมาซึ่งสัญญาณรบกวน ในระบบอื่น ๆอาจจะปะปนมาด้วยดังนั้นเราจึงทำการกรองสัญญาณที่แท้จริงออกมาเพื่อที่จะสามารถนำไปใช้ในการวิเคราะห์ได้ถูกต้อง

3.3 โครงสร้างของตัวกรองความถี่ดิจิตอล

Discrete-Time Filter จะถูกนำมาแสดงบ่อยๆในรูปแบบของบล็อกไดอะแกรมหรือ Signal Flow Graph ซึ่งบล็อกไดอะแกรมจะเป็นการสะดวกกว่าที่ใช้ในการแสดง Difference Equations หรือ Equivalent ของ Transfer Function ซึ่งสามารถพิจารณาดังตัวอย่าง Discrete Filter ด้วย Difference Equations ดังรูปต่อไปนี้



รูปที่ 3.11 การวิเคราะห์สมการ Difference Equation ในรูปแบบบล็อกไดอะแกรม

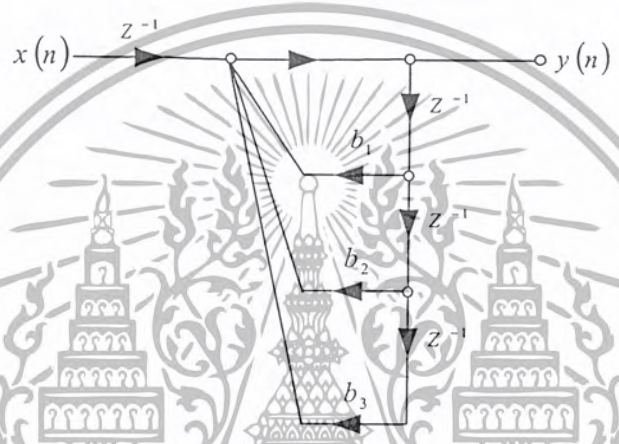
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยการสมมติสมการดังนี้

$$y(n) = x(n-1) - b_1 y(n-1) + b_2 y(n-2) + b_3 y(n-3) \qquad (3.26)$$

บล็อกไดอะแกรมแสดงถึงสมการดังกล่าวจากรูปสัญลักษณ์ Z จะแสดงถึง Delay ของช่วงเวลาหนึ่งหน่วย ซึ่งอาจพิจารณาใน Node ต่างๆ ลูกศรแสดงถึงการ Multiplies และค่าคงตัวที่ตัวถัดมา ทำการคูณกับตัวประกอบความสัมพันธ์ระหว่าง Difference Equations และ Block Diagram จะค่อยๆปรากฏขึ้นและ Signal Flow Graph Diagram ก็ใช้แสดงกับ Difference Equations เช่นเดียวกันดังรูป 3.11 โดยสมมติสมการดังนี้

$$y(n) = x(n-1) - b_1 y(n-1) + b_2 y(n-2) + b_3 y(n-3)$$

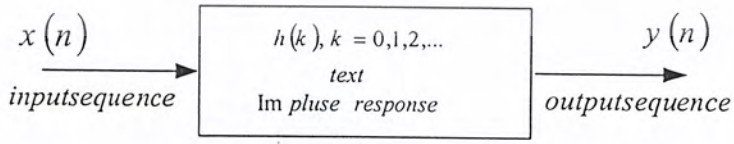


รูปที่ 3.12 การวิเคราะห์สมการ Difference Equation ในรูปแบบ Signal Flow Graph Diagram

3.4 ชนิดของตัวกรองความถี่ดิจิตอล

ตัวกรองความถี่ดิจิตอลที่ใช้อย่างกว้างขวางแบ่งออกเป็น 2 ชนิด

1. Finite Impulse Response (FIR)
2. Infinite Impulse Response (IIR)



รูปที่ 3.13 บล็อกไดอะแกรมแสดงถึงหลักการของตัวกรองความถี่ดิจิตอล

ตัวกรองสัญญาณแต่ละชนิด จะมีรูปแบบพื้นฐานแต่ละตัวซึ่งสามารถแสดงลำดับของการตอบสนองต่อ Impulse h(k) ;(k=0,1,2,...) ดังรูปที่ 3.13 สัญญาณอินพุตและเอาท์พุตที่ต่อตัวกรองสัญญาณจะถูกนำมาเกี่ยวพันโดยผลรวมของมัน ซึ่งสมการแต่ละชนิดแสดงได้ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$y(n) = \sum_{k=0}^{n-1} h(k)x(n-k) \qquad \text{;IIR digital filter (3.27)}$$

$$y(n) = \sum_{k=0}^{\infty} h(k)x(n-k) \qquad \text{;FIR digital filter (3.28)}$$

จากสมการข้างต้น สำหรับตัวกรองความถี่แบบไอไออาร์นั้น ผลการตอบสนองต่ออิมพัลส์จะอยู่ในช่วงที่ไม่จำกัด แต่สำหรับตัวกรองความถี่แบบเอฟไออาร์จะอยู่ในช่วงที่กำหนด โดยเริ่มที่ h(k) สำหรับ FIR จะมีเพียง N ค่า ส่วนในการคำนวณหาค่าเอาต์พุตของตัวกรองความถี่แบบไอไออาร์ไม่สามารถเป็นไปได้ เนื่องจากสมการมีความยาวของผลการตอบสนองต่อ อิมพัลส์นั้นยาวมาก (∞: Infinite) ดังนั้นเราจึงแทนสมการของตัวกรองความถี่แบบไอไออาร์ ในรูปของ Recursive Form

$$y(n) = \sum_{k=0}^{\infty} h(k)x(n-k) = \sum_{k=0}^{\infty} a_k x(n-k) - \sum_{k=1}^{\infty} a_k x(n-k) \qquad (3.29)$$

โดยที่ a_k และ b_k เป็นสัมประสิทธิ์ของตัวกรองสัญญาณ ดังนั้นจากสมการข้างต้นคือสมการที่แตกต่างกันระหว่างตัวกรองความถี่แบบเอฟไออาร์และไอไออาร์ จากสมการเหล่านั้นจะมีค่า h(k) สำหรับ FIR และ a_k และ b_k สำหรับ IIR ซึ่งสมการทั้งสองเป็นสิ่งที่สำคัญมากที่ใช้ในการออกแบบตัวกรองสัญญาณเป็นส่วนใหญ่ จากสมการกระแสการสุ่มตัวอย่าง เอาต์พุต y(n) เป็นฟังก์ชันของเอาต์พุตที่ผ่านมา ซึ่งค่าที่ถูกต้องจะเท่ากับตัวอย่างอินพุตที่สุ่มมาทั้งในอดีตและปัจจุบัน ซึ่งนั่นคือ IIR จะมีการ Feedback ซึ่งจะทำการเปรียบเทียบกับสมการ FIR แล้วกระแสการสุ่มตัวอย่างเอาต์พุต y(n) จะเป็นเพียงฟังก์ชันของค่าอินพุตของอดีตและปัจจุบันเท่านั้นเอง แต่อย่างไรก็ตามเมื่อ b_k มีค่าเท่ากับ 0 สมการ IIR คือ สมการ FIR นั่นเอง

และจากสมการข้างล่างนี้คือ Transfer Function ของตัวกรองความถี่แบบเอฟไออาร์และไอไออาร์ ซึ่งค่า Transfer Function เหล่านี้จะมีประโยชน์มากในการคำนวณหาค่าผลตอบสนองทางความถี่

$$H(z) = \frac{\sum_{k=0}^{n} a_k z^{-k}}{1 + \sum_{k=1}^{m} b_k z^{-k}} \qquad \text{;IIR digital filter (3.30)}$$

$$H(z) = \sum_{k=0}^{n-1} h(k) z^{-k} \qquad \text{;FIR digital filter (3.31)}$$

3.5 สรุปรูปแบบของตัวกรองความถี่แบบเอฟไออาร์

3.5.1 สรุปรูปแบบของตัวกรองความถี่แบบเอฟไออาร์จะแสดงดังสมการ

$$y(n) = \sum_{k=0}^{n-1} h(k)x(n-k) \qquad (3.32)$$

$$H(z) = \sum_{k=0}^{n-1} h(k) z^{-k} \qquad (3.33)$$

โดยที่ h(k);k=0,1,2,…,n-1 เป็นสัมประสิทธิ์ในการตอบสนองต่อของตัวกรองสัญญาณ

H(z) เป็น Transfer Function ของตัวกรองสัญญาณ

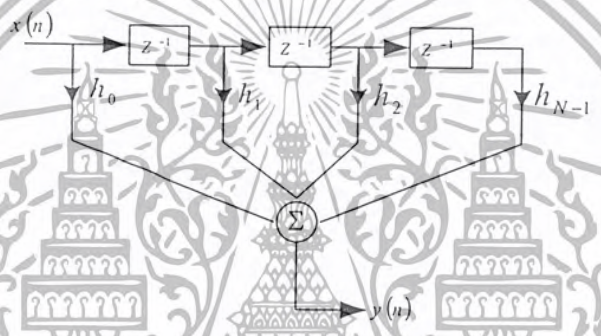
N เป็น Filter Length

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นั่นก็คือจำนวนสัมประสิทธิ์ของตัวกรองสัญญาณของสมการแรกเป็น FIR Difference Equation ซึ่งมันจะเป็นสมการใน Time Domain และจะพิจารณาตัวกรองความถี่แบบเอฟไอาร์ในรูปของ Non Recursive Form ซึ่งจะทำการสุ่มตัวอย่างเอาต์พุต y(n) เป็นฟังก์ชันเพียงค่าของอินพุตในอดีตและปัจจุบันของ x(n) เท่านั้น เมื่อตัวกรองสัญญาณถูกนำไปใช้ในรูปแบบนี้นั่นคือ จะเป็นการหาค่าโดยตรง จะทำให้ได้ค่าที่คงที่ และ Transfer Function ของตัวกรองสัญญาณ จะใช้ในการวิเคราะห์และหาค่าตัวกรองสัญญาณในทางผลตอบสนองทางความถี่

3.5.2 ตัวกรองความถี่แบบเอฟไอาร์มีผลตอบสนองทางเฟสเป็นเชิงเส้นที่แท้จริง

3.5.3 ตัวกรองความถี่แบบเอฟไอาร์ง่ายที่จะนำไปใช้ในระบบ DSP Processor ทั้งหมด มีความเหมาะสมที่จะใช้กับ FIR Filtering Non-Recursive ตัวกรองความถี่แบบเอฟไอาร์จะมีข้อผิดพลาดน้อยมากจากผลของความยาวที่จำกัดกว่า



รูปที่ 3.14 โครงสร้างของตัวกรองความถี่แบบเอฟไอาร์

3.6 สรุปรูปแบบของตัวกรองความถี่แบบไอไอาร์

รูปแบบของตัวกรองความถี่แบบไอไอาร์ถูกกำหนดตาม Recursive ดังนี้

$$y(n) = \sum_{k=0}^{\infty} h(k)x(n-k) = \sum_{k=0}^{\infty} a_k x(n-k) - \sum_{k=1}^{\infty} a_k x(n-k) \qquad (3.34)$$

โดยที่ h(k) เป็นการตอบสนองต่อ Impulse ของตัวกรองสัญญาณ ซึ่งจะมีช่วงเวลาที่ไม่จำกัด

a_k และ b_k เป็นสัมประสิทธิ์ของตัวกรองความถี่

x(n) และ y(n) เป็นอินพุตและเอาต์พุตของตัวกรองความถี่

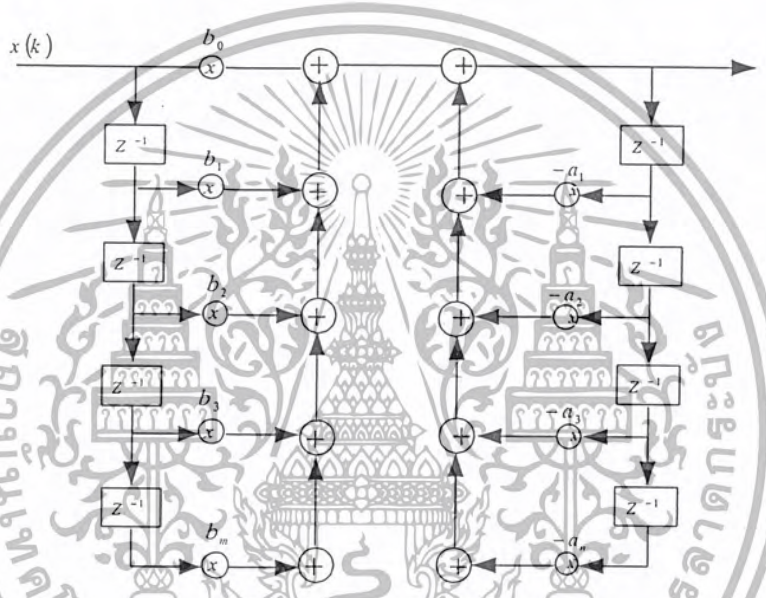
Transfer Function สำหรับตัวกรองความถี่แบบไอไอาร์ถูกกำหนดโดย

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + ... + a_n z^{-n}}{1 + b_1 z^{-1} + b_2 z^{-2} + ... + b_m z^{-m}} = \frac{\sum_{k=0}^{n} a_k z^{-k}}{1 + \sum_{k=0}^{m} b_k z^{-k}} \qquad (3.35)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่สำคัญในการออกแบบตัวกรองความถี่แบบไอไออาร์คือการค้นหาค่าที่เหมาะสม สำหรับค่าสัมประสิทธิ์และให้สอดคล้องกับลักษณะของตัวกรองสัญญาณในผลตอบสนองทางความถี่ ดังนั้นสมการดังกล่าวทั้งสองจึงเป็น Characteristic Equation สำหรับตัวกรองความถี่แบบไอไออาร์

จากสมการแรก กล่าวได้ว่ากระแสการสุ่มตัวอย่างเอาต์พุต, y(n) เป็นฟังก์ชันของเอาต์พุตที่ผ่านมา y(n) ซึ่งจะเท่ากับค่าการสุ่มตัวอย่างอินพุตทั้งในปัจจุบันและอดีต นั่นคือตัวกรองความถี่แบบไอไออาร์เป็นระบบ Feedback ซึ่งปกติแล้วตัวกรองความถี่แบบไอไออาร์ต้องการเพียงสัมประสิทธิ์ 2-3 ตัว มากกว่าตัวกรองความถี่แบบเอฟไออาร์สำหรับข้อกำหนดเดียวกัน ซึ่งตัวกรองความถี่แบบไอไออาร์จะถูกใช้เมื่อจุดคัตออฟแคบ และมี Throughput ที่สูง ซึ่งเป็นความต้องการที่สำคัญและเป็นเหตุผลที่จะทำให้ตัวกรองความถี่แบบไอไออาร์ไม่คงที่ (Unstable)



รูปที่ 3.15 โครงสร้างตัวกรองความถี่แบบไอไออาร์

ข้อได้เปรียบและข้อด้อยของการประมวลผลสัญญาณเชิงเลข

อาจกล่าวได้ว่า ข้อเด่นเป็นอย่างมากของการประมวลผลสัญญาณเชิงเลขก็คือ ความเที่ยงตรง ความแม่นยำ ในการประมวลผล อย่างไรก็ตามการนำเอาระบบประมวลผลสัญญาณเชิงเลขมาใช้ร่วมกับการประมวลผลสัญญาณเชิงอุปมานที่ยังมีการใช้กันมากในปัจจุบัน ก็ต้องมีข้อเสียเปรียบอยู่มากเช่นกัน ดังนั้นการกล่าวถึงข้อได้เปรียบ และข้อด้อย บางประการของการประมวลผลสัญญาณเชิงเลขจึงอาจเป็นประโยชน์บ้าง และนอกจากนี้ยังเป็นการให้ภาพกว้างๆแก่ผู้ที่จะนำเอาระบบประมวลผลสัญญาณเชิงเลขไปประยุกต์ใช้ได้อย่างเหมาะสมยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ข้อดีของการประมวลผลสัญญาณเชิงเลข

ข้อได้เปรียบของการประมวลผลสัญญาณเชิงเลขอาจกล่าวได้เป็นข้อๆดังนี้

1) เหมาะสำหรับอุปกรณ์ที่ข้อมูลอยู่ในรูปแบบสัญญาณเชิงเลขอยู่แล้ว เช่น ผลลัพธ์จากคอมพิวเตอร์ ไมโครโปรเซสเซอร์ หรือ ข้อมูลจากระบบควบคุมเชิงเลข เป็นต้น ทั้งนี้เนื่องจากสัญญาณหรือข้อมูลจากอุปกรณ์เหล่านี้ ถ้าหากต้องการนำไปประมวลผลสัญญาณเชิงอุปมาน ก็จำเป็นต้องมีวงจรแปลงรูปแบบสัญญาณเชิงเลขเป็นเชิงอุปมาน (Digital to Analog Converter หรือ D/A) เพื่อแปลงสัญญาณเชิงเลขให้เป็นสัญญาณเชิงอุปมานก่อน หลังจากประมวลผลด้วยระบบประมวลผลเชิงอุปมานแล้วก็ต้องแปลงผลลัพธ์ที่เป็นสัญญาณเชิงอุปมานเป็นเชิงเลข (Analog to Digital Converter หรือ A/D) เพื่อส่งกลับไปยังอุปกรณ์เชิงเลขต่อไป ซึ่งจะเห็นได้ว่าระบบประมวลผลมีความซับซ้อนมากขึ้น นอกจากนี้ถ้าหากเป็นระบบที่ต้องการทำการประมวลผลสัญญาณพร้อมกันหลายสัญญาณ (ซึ่งในคอมพิวเตอร์มักทำในลักษณะนี้) ก็จะทำให้ระบบยุ่งยากมากขึ้นไปอีกทั้งในปัจจุบัน A/D หรือ D/A ที่มีประสิทธิภาพสูงและมีความเร็วในการแปลงสูงยังมีราคาแพงอยู่

2) อุปกรณ์ทางด้านเชิงเลขหรือดิจิตอลมีราคาถูก ขนาดเล็ก มีประสิทธิภาพสูง ความแม่นยำและความแน่นอนสูง นอกจากนี้การพัฒนาให้มีประสิทธิภาพสูงขึ้นก็เป็นไปอย่างรวดเร็ว ข้อนี้เป็นข้อได้เปรียบที่สำคัญ เพราะว่าการประมวลผลสัญญาณเชิงเลขสามารถคำนวณได้อย่างแม่นยำและต้องการให้มีความละเอียดเท่าใดก็ได้ แต่การประมวลผลเชิงอุปมานการคำนวณที่ให้ความละเอียดเกินกว่าหนึ่งในพันส่วน หรือ 0.1 นั้น ทำได้ยากมาก นอกจากนี้ชิ้นอุปกรณ์ที่ใช้ในระบบของการประมวลผลเชิงอุปมาน เช่น ตัวความต้านทาน ตัวขยายสัญญาณ ยังมีคุณสมบัติแปรค่าได้ตามสภาพแวดล้อม เช่น อุณหภูมิ ความชื้น อายุการใช้งาน ทำให้ความแม่นยำและความน่าเชื่อถือของระบบการประมวลผลเชิงอุปมานต่ำ อย่างไรก็ตามในปัจจุบันการประมวลผลเชิงเลขที่ให้ความแม่นยำสูงต้องใช้อุปกรณ์ราคาแพง ถ้ามีการพัฒนาให้มีราคาถูกก็จะเป็นผลดีอย่างยิ่ง

3) การรับ และส่งข้อมูลหรือสัญญาณเชิงเลขทำได้แน่นอนกว่า ทั้งนี้เนื่องจากสัญญาณเชิงเลขมีแค่สองระดับ คือ 0 กับ 1 เท่านั้น ถ้าหากรูปคลื่นสัญญาณผิดเพี้ยนไปก็สามารถแก้ไขและสร้างขึ้นมาใหม่ให้เหมือนเดิมได้โดยง่าย

4) การประมวลผลสัญญาณเชิงเลขทำได้ง่าย เนื่องจาก Algorithm การประมวลผลสัญญาณมักจะประกอบด้วย การบวก การลบ การคูณ การหาร และการเลื่อนตัวเลขเท่านั้น ซึ่งข้อความนี้สามารถแสดงให้เห็นจริงได้ในบทต่อๆไป

5) ระบบประมวลผลสัญญาณเชิงเลข สามารถทำเป็นแบบระบบแบ่งกันใช้เวลา (Time-Shared System) ได้ ดังนั้นจึงสามารถทำการประมวลผลพร้อมๆกันหลายช่องสัญญาณได้ และนอกจากนี้ในระบบเดียวกันยังสามารถโปรแกรมให้ทำงานได้หลายรูปแบบด้วย

6) ระบบประมวลผลสัญญาณเชิงเลข มีความคล่องตัวสูง ทั้งนี้เนื่องจากสามารถทำการมัลติเพล็กซ์แบบแบ่งเวลา (Time Division Multiplex หรือ TDM) การมัลติเพล็กซ์แบบแบ่งรหัส (Code Division

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Division Multiplex หรือ CDM) ได้ นอกจากนี้การส่งสัญญาณยังส่งได้ในอัตราบิต (Bit Rate) หรืออัตราบอร์ด (Baud Rate) ต่างๆกันได้

ข้อด้อยของการประมวลผลสัญญาณเชิงเลข

นอกจากข้อดีที่กล่าวแล้ว การประมวลผลสัญญาณเชิงเลขทำให้เกิดข้อด้อยหลายประการคือ

1) ระบบประมวลผลสัญญาณเชิงเลขจะต้องมีสัญญาณสำหรับการซิงโครไนซ์ (Synchronize) การจับเวลา (Timing) และการกำหนดกรอบ (Framing) ของข้อมูลไว้ด้วย ถ้าหากสัญญาณเหล่านี้สูญหายหรือผิดพลาดไป การทำงานของระบบก็จะผิดพลาดไปด้วย

2) มีปัญหาการเชื่อมโยง (Interfacing) กับระบบการประมวลผลสัญญาณเชิงอุปมาน ทั้งนี้เนื่องมาจากระบบประมวลผลสัญญาณแบบเดิมส่วนมากอยู่ในระบบเชิงอุปมาน ดังนั้นระบบที่สร้างขึ้นมาใหม่ก็ต้องสามารถเชื่อมโยงกับระบบเดิมให้ได้ ซึ่งอาจทำให้ระบบประมวลสัญญาณซับซ้อนขึ้น

3) สัญญาณหรือข้อมูลเชิงเลขไม่ใช่สัญญาณตามธรรมชาติที่แท้จริง แต่ถูกสร้างขึ้นมา ดังนั้นการส่งสัญญาณชนิดนี้ไปในตัวกลางตามธรรมชาติทั่วๆไป ซึ่งมักมีแถบความถี่ปฏิบัติงาน (Band Width) จำกัดและยังมีผลตอบสนองเฟสไม่เป็นเชิงเส้น (Non-Linear Phase Response) จึงอาจทำให้สัญญาณเกิดความผิดเพี้ยนได้ เช่น เมื่อเราส่งสัญญาณรูปเหลี่ยมไปบนสายส่ง สัญญาณที่ได้รับไม่เป็นรูปเหลี่ยมเหมือนเดิม

4) เนื่องจากการที่เราต้องการออกแบบระบบประมวลผลสัญญาณเชิงเลขให้มีความคล่องตัวสูง สามารถใช้งานได้หลายรูปแบบ ดังนั้นอาจทำให้การออกแบบระบบประมวลผลสัญญาณเชิงเลขมีความซับซ้อนมาก ซึ่งรวมทั้งด้านการซ่อมแซม บำรุงรักษา และจัดทำคู่มือใช้งานด้วย

5) แถบความถี่ปฏิบัติงานของระบบประมวลผลสัญญาณเชิงเลขจะต่ำกว่าแถบความถี่ปฏิบัติงานของระบบประมวลผลเชิงอุปมานมาก ข้อจำกัดนี้เนื่องมาจากอุปกรณ์ที่ใช้หรือประกอบขึ้นเป็นระบบประมวลผลสัญญาณเชิงเลข เช่น วงจรเกต วงจรชิฟต์รีจิสเตอร์ วงจรสุ่มและคงค่าสัญญาณ (Sampling and Hold Circuit หรือ S/H) หรือ วงจร A/D และ D/A เป็นต้น วงจรเหล่านี้ต่างมีความเร็วสูงสุดในการทำงานจำกัดอยู่ค่าหนึ่ง ซึ่งในปัจจุบันยังมีค่าต่ำมาก จึงเป็นผลทำให้ระบบประมวลผลสัญญาณเชิงเลขมีความเร็วต่ำ ตัวอย่างเช่น ถ้าไมโครโปรเซสเซอร์ที่มีสัญญาณนาฬิกาขนาด 1 เมกกะเฮิรตซ์ เมื่อนำไปสร้างเป็นระบบประมวลผลสัญญาณเชิงเลขจะใช้ได้กับสัญญาณที่มีความถี่สูงสุดประมาณ 10 ถึง 20 กิโลเฮิรตซ์เท่านั้น

อย่างไรก็ตาม เมื่อพิจารณาข้อดีและข้อด้อยต่างๆเหล่านี้แล้ว ถ้าเราต้องการระบบประมวลผลสัญญาณที่มีประสิทธิภาพดี ความแม่นยำสูง โดยไม่คำนึงว่าราคาสูงเท่าใดแล้ว ระบบประมวลผลสัญญาณเชิงเลขก็ให้ประโยชน์ได้มาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การออกแบบและการทดลอง

4.1 การออกแบบวงจรกรองสัญญาณดิจิตอลแบบเอฟไออาร์

ในการออกแบบวงจรกรองสัญญาณดิจิตอลชนิดผลตอบสนองอิมพัลส์จำกัดโดยทั่วไปจะทำการออกแบบจากอนุกรมฟูเรียร์ (Fourier Series) โดยผลตอบสนองที่ได้นี้จะเป็นผลตอบสนองในทางอุดมคติ ซึ่งในทางปฏิบัติ ค่าสัมประสิทธิ์หรือผลตอบสนองอิมพัลส์ที่จะนำมาใช้งานนั้นจะต้องมีจำนวนที่จำกัด ซึ่งผลจากการตัดค่าผลตอบสนองอิมพัลส์ให้มีจำนวนที่จำกัดนี้จะทำให้เกิดความพริ้ว (Ripples) ขึ้นที่ย่านผ่านและหยุดสัญญาณ รวมทั้งเกิดโอเวอร์ชูท (Overshoot) ขึ้นที่ขอบของย่านผ่านสัญญาณ ซึ่งผลที่เกิดขึ้นนี้รวมเรียกว่าปรากฏการณ์กิปส์ (Gibbs's Phenomenon) การลดผลที่เกิดขึ้นนี้สามารถทำได้โดยการถ่วงน้ำหนัก (Weighted) ด้วยฟังก์ชั่นหน้าต่าง (Windows Function) ชนิดต่างๆ

ผลตอบสนองทางความถี่ของวงจรกรองสัญญาณที่มีผลตอบสนองอิมพัลส์จำกัดจำนวน N ตัว สามารถคำนวณได้จากการแปลงฟูเรียร์เวลาเต็มหน่วย (Discrete-Time Fourier Transform: DTFT) ของผลตอบสนองอิมพัลส์ ได้ดังนี้

$$H(\omega) = \sum_{n=0}^{N-1} h(n)\, e^{-j\omega n} \tag{4.1}$$

สมมุติให้เฟสเป็นเชิงเส้นจะสามารถเขียนได้เป็น

$$H(\omega) = A(\omega)\, e^{-jM\omega} \tag{4.2}$$

โดย $A(\omega)$ คือ ผลตอบสนองแอมปลิจูด

$M = \dfrac{(N-1)}{2}$ คือ ค่าคงที่ความหน่วงกลุ่ม (Constant Group Delay) ของวงจร

สำหรับ N เป็นจำนวนคี่ จะได้

$$A(\omega) = \sum_{n=-M}^{M} \hat{h}(n)\cos(\omega n) \tag{4.3}$$

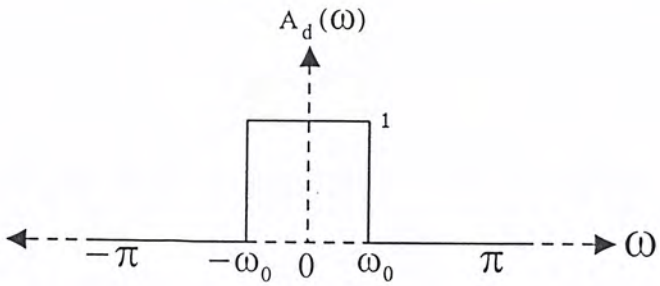
โดย $\hat{h}(n) = h(n+M)$ ซึ่งคือค่า $h(n)$ ที่ถูกเลื่อนไปทางซ้ายเพื่อทำให้เกิดความสมมาตร (Symmetric) ที่ $n = 0$

ผลการแปลงกลับฟูเรียร์เวลาเต็มหน่วย (Inverse DTFT) ของ $A(\omega)$ จะทำให้ได้ผลตอบสนองอิมพัลส์ดังนี้

$$\hat{h}(n) = \frac{1}{\pi}\int_{0}^{\pi} A(\omega)\cos(\omega n)\, d\omega \tag{4.4}$$

สำหรับวงจรกรองสัญญาณความถี่ต่ำผ่านในอุดมคติ ผลตอบสนองทางแอมปลิจูด สามารถแสดงได้ดังรูปที่ 4.1.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1.1 ผลตอบสนองแอมปลิจูดของวงจรกรองสัญญาณความถี่ต่ำผ่านในอุดมคติ

ผลตอบสนองในทางอุดมคติจะมีค่าเป็น 1 ถ้าความถี่อยู่ในย่านผ่านสัญญาณ คือระหว่าง 0 ถึง ω_0 และมีค่าเป็น 0 ถ้าความถี่อยู่ในย่านหยุดสัญญาณ คือระหว่าง ω_0 ถึง π (พิจารณาเป็นค่า Normalized Frequency คือ $\omega_s = 2\pi$) ซึ่งเขียนเป็นสมการได้โดย

$$A_d(\omega) = \begin{cases} 1 & 0 < \omega < \omega_0 \\ 0 & \omega_0 < \omega < \pi \end{cases} \tag{4.5}$$

อาศัยสมการที่ (4.4) และ สมการที่ (4.5) จะทำให้ได้ผลตอบสนองอิมพัลส์ในอุดมคติ ดังนี้

$$\begin{aligned} \hat{h}_d(n) &= \frac{1}{\pi}\int_0^{\pi} A_d(\omega)\cos(\omega n)\, d\omega \\ &= \frac{1}{\pi}\int_0^{\omega_0} \cos(\omega n)\, d\omega = \frac{\sin(\omega_0 n)}{\pi n} \end{aligned} \tag{4.6}$$

ซึ่งค่า $\hat{h}_d(n)$ ที่เป็นผลของสมการที่ (4.6) โดยทั่วไปจะมีจำนวนเป็นอนันต์ เพื่อที่จะให้ได้ผลตอบสนองในทางอุดมคติ แต่ในทางปฏิบัติจะต้องทำการตัดให้เหลือจำนวนเพียง N ตัว และเลื่อนไปทางขวา M ตำแหน่ง เพื่อที่จะทำให้ระบบคอซอล (Causal) ซึ่งจะทำให้ได้ผลตอบสนองอิมพัลส์เป็น

$$h(n) = \begin{cases} \dfrac{\sin(\omega_0(n-M))}{\pi(n-M)} & \text{สำหรับ } 0 \le n \le N-1 \\ 0 & \text{อื่นๆ} \end{cases} \tag{4.7}$$

ในทำนองเดียวกันเราสามารถหา $\hat{h}_d(n)$ สำหรับวงจรกรองชนิดอื่นๆได้โดยใช้วิธีการข้างต้น กระทำกับผลตอบสนองทางความถี่ของวงจรกรองนั้นๆ ซึ่งตารางที่ 1 ได้สรุปค่าของ $\hat{h}_d(n)$ สำหรับวงจรกรองชนิดอื่นๆไว้ดังนี้

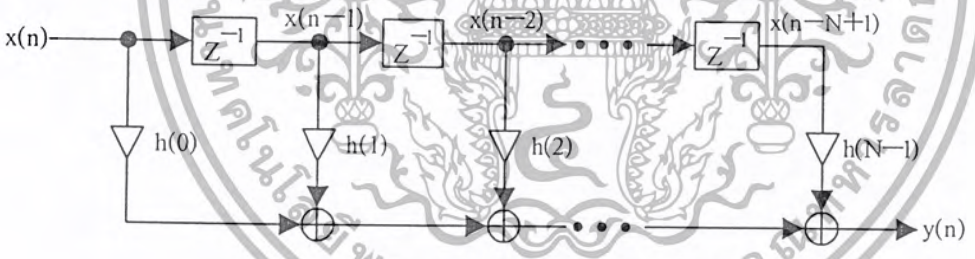
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชนิดของวงจรกรอง	$\hat{h}_d(n), -\infty < n < \infty$	$\hat{h}_d(0)$
ความถี่ต่ำผ่าน (LPF)	$\frac{\sin(\omega_0 n)}{\pi n}$	$\frac{\omega_0}{\pi}$
ความถี่สูงผ่าน (HPF)	$-\frac{\sin(\omega_0 n)}{\pi n}$	$1-\frac{\omega_0}{\pi}$
แถบความถี่ผ่าน (BPF)	$\frac{\sin(\omega_{02} n)-\sin(\omega_{01} n)}{\pi n}$	$\frac{\omega_{02}-\omega_{01}}{\pi}$
แถบความถี่หยุด (BSF)	$\frac{\sin(\omega_{01} n)-\sin(\omega_{02} n)}{\pi n}$	$1-\frac{\omega_{02}-\omega_{01}}{\pi}$

ตารางที่ 4.1.1 ผลตอบสนองอิมพัลส์ในอุดมคติของวงจรกรองชนิดต่างๆ

การจัดโครงสร้างสำหรับวงจรกรองสัญญาณดิจิตอลแบบเอฟไอาร์

สำหรับวงจรกรองสัญญาณดิจิตอลแบบ FIR จะใช้ผลตอบสนองอิมพัลส์ หรือ h(n) เพื่อสร้างวงจร ซึ่งกระบวนการทำงานของวงจรกรองในที่นี้ก็คือการทำคอนโวลูชั่นระหว่าง h(n) กับสัญญาณอินพุท x(n) โดยสามารถจัดเป็นโครงสร้างได้ดังรูป



รูปที่ 4.1.2 การจัดโครงสร้างของวงจรกรองสัญญาณดิจิตอลแบบเอฟไอาร์

การทำ คอนโวลูชั่นระหว่าง h(n) กับสัญญาณอินพุท x(n) แสดงได้ดังสมการ

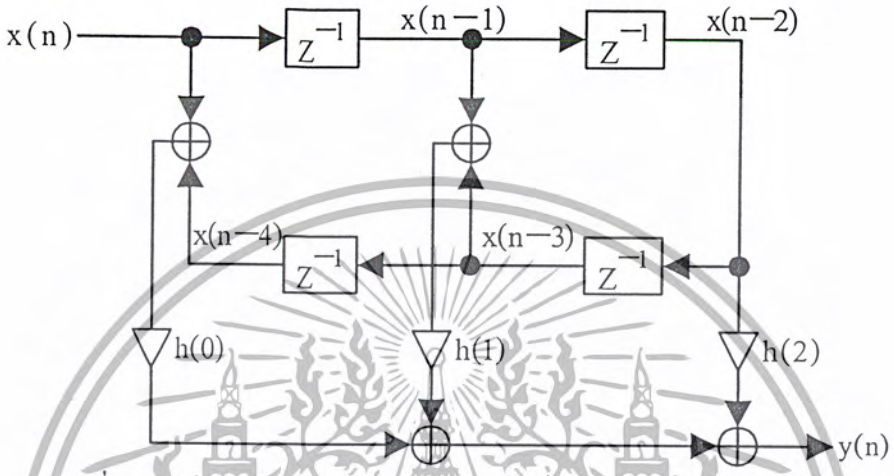
$$y(n) = x(n) * h(n) = \sum_{i=0}^{N-1} h(i)x(n-i) \tag{4.8}$$

ซึ่งสามารถแสดงเป็นสมการผลต่างสืบเนื่อง (Difference Equation) ได้ดังนี้

$$y(n) = h(0)x(n) + h(1)x(n-1) + h(2)x(n-2) + \ldots + h(N-1)x(n-N+1) \tag{4.9}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับในกรณีที่อันดับของวงจรกรองหรือจำนวนผลตอบสนองอิมพัลส์ ซึ่งก็คือตัวแปร N เป็นจำนวนคี่ จะทำให้ผลตอบสนองอิมพัลส์มีลักษณะสมมาตรคี่ คือ $h(n) = h(N-1-n)$ เช่นกรณี N=5 จะได้ว่า $h(0) = h(4), h(1) = h(3)$ โดยที่ $h(2)$ ไม่มีคู่สมมาตร ซึ่งทำให้การจัดโครงสร้างสามารถลดจำนวนตัวคูณลงไปได้ทำให้ขนาดของวงจรกรองเล็กลงซึ่งโดยปกติในทางดิจิตอลตัวคูณจะมีขนาดของวงจรที่ใหญ่และใช้เวลาในการทำงานมากเมื่อเทียบกับส่วนอื่นๆ โดยวงจรกรองในอันดับที่ 5 ที่ได้จัดโครงสร้างใหม่แสดงได้ดังรูป

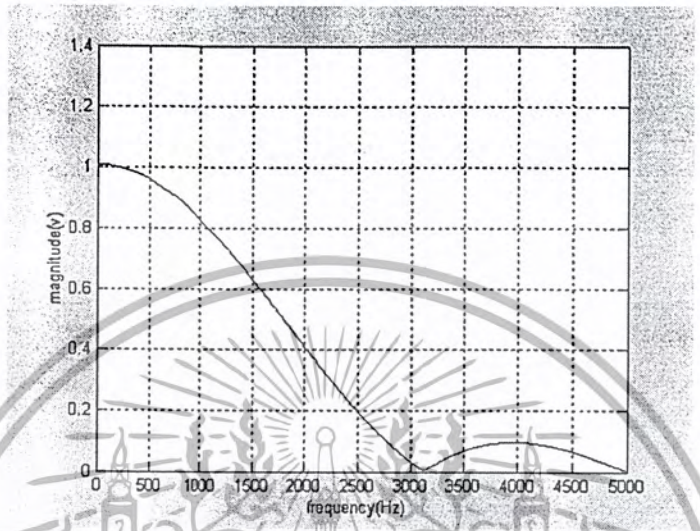


รูปที่ 4.1.3 โครงสร้างของวงจรกรองในอันดับที่ 5 โดยลดจำนวนตัวคูณลง

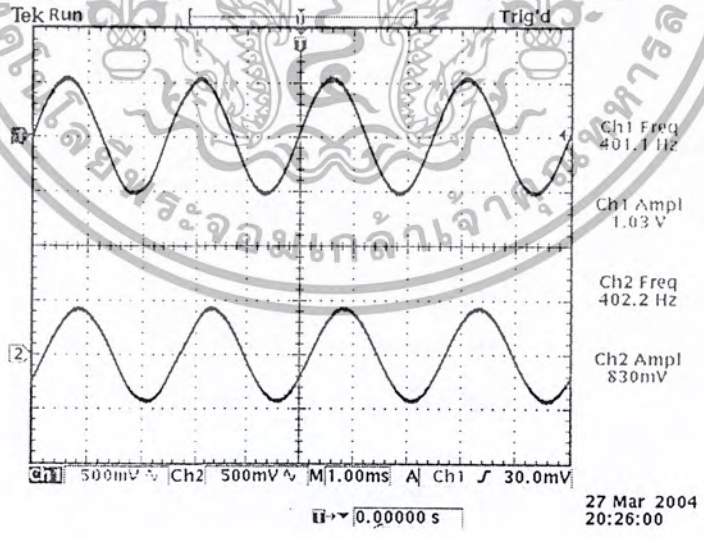
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
ผลการทดลอง

5.1 วงจรกรองความถี่ต่ำผ่านอันดับ 5 ความถี่คัทออฟ 1 KHz



รูปที่ 5.1.1 กราฟผลตอบสนองทางความถี่ของตัวกรองแบบเอฟไอาร์โดยใช้เจเนติก อัลกอริทึมในการหาค่าสัมประสิทธิ์ของตัวกรองความถี่ และใช้ Matlab เขียนกราฟ

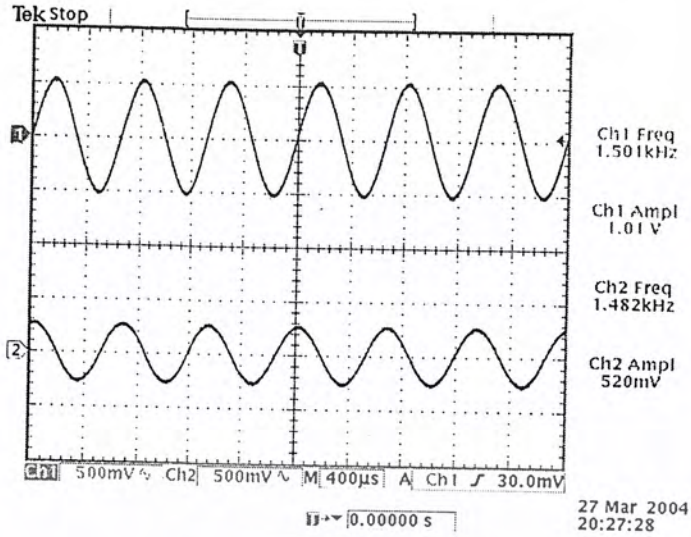


รูปที่ 5.1.2 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไอาร์ เทียบกับสัญญาณอินพุท

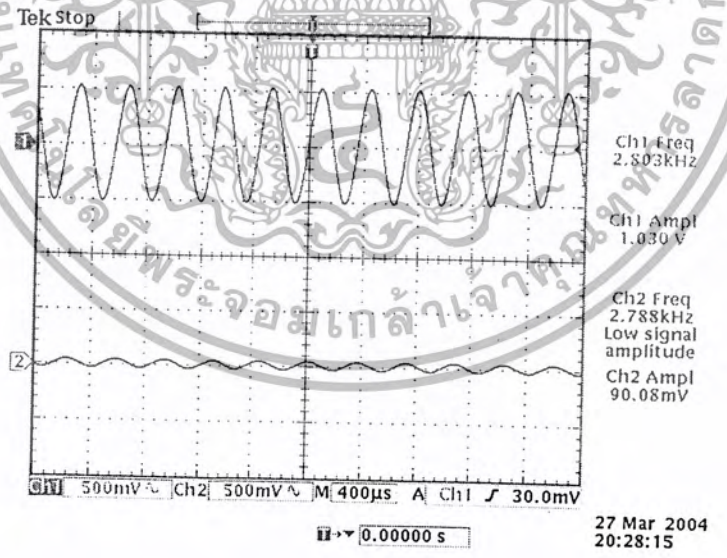
CH1 สัญญาณอินพุทที่เข้าวงจรกรองความถี่ต่ำผ่าน

CH2 สัญญาณเอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

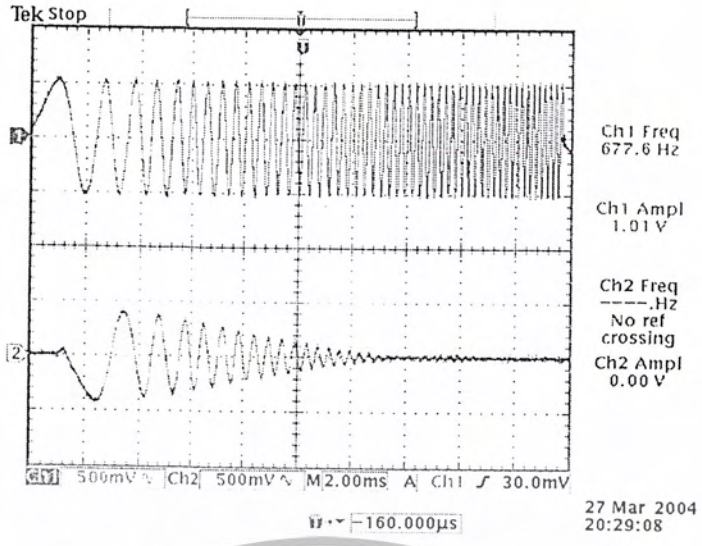


รูปที่ 5.1.3 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ เทียบกับสัญญาณอินพุท
CH1 สัญญาณอินพุทที่เข้าวงจรกรองความถี่ต่ำผ่านเมื่อปรับความถี่ให้สูงขึ้น
CH2 สัญญาณเอาท์พุท



รูปที่ 5.1.4 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ เทียบกับสัญญาณอินพุท
CH1 สัญญาณอินพุทที่เข้าวงจรกรองความถี่ต่ำผ่านเมื่อปรับความถี่ให้สูงขึ้น
CH2 สัญญาณเอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

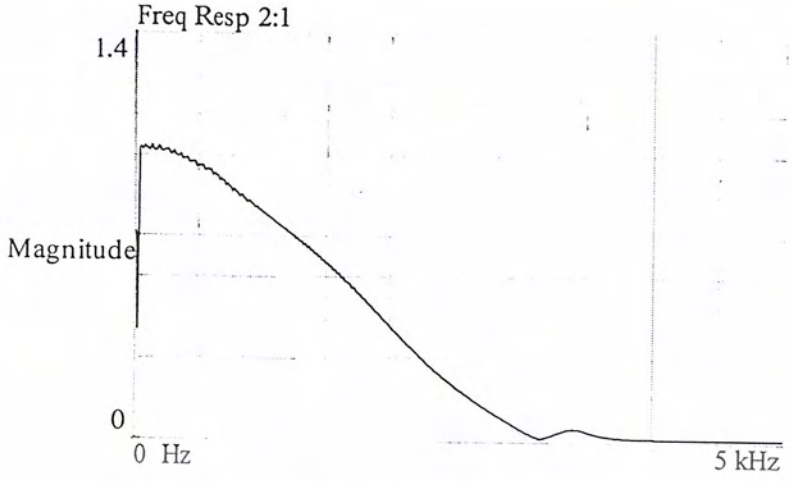


รูปที่ 5.1.5 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ เทียบกับสัญญาณอินพุทที่เป็นแบบสวีฟ

CH1 สัญญาณอินพุทที่เข้าวงจรกรองความถี่ต่ำผ่าน

CH2 สัญญาณเอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



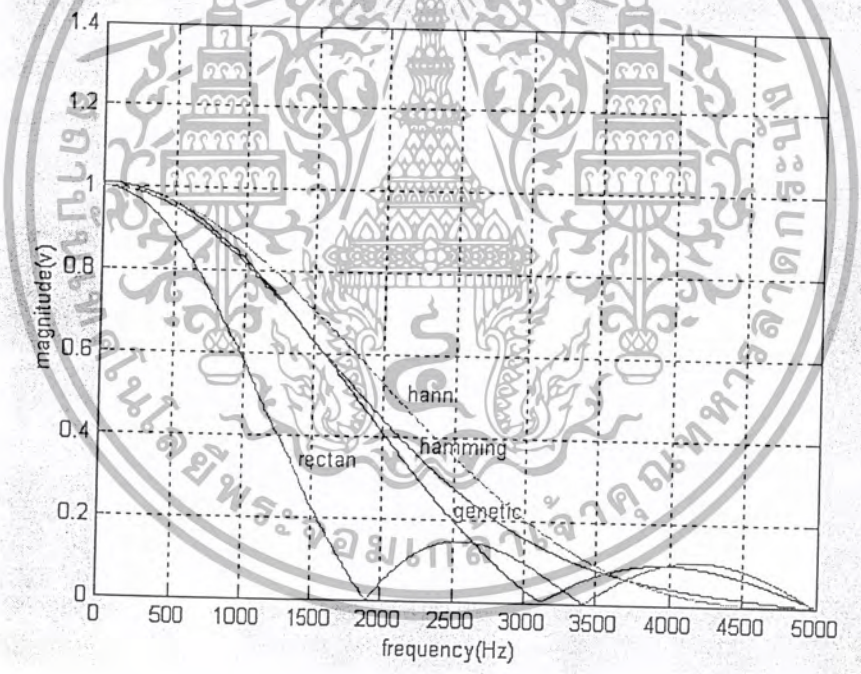
รูปที่ 5.1.6 ผลตอบสนองทางความถี่ของสัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ซึ่งใช้ไดนามิกซิกแนลอนาไลเซอร์ (Dynamic Signal Analyzer) ในการบันทึกผล

รูปที่ 5.1.7 ผลตอบสนองทางเฟสของสัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ซึ่งใช้ไดนามิกซิกแนลอนาไลเซอร์ในการบันทึกผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FREQZ(HZ)	MAGNITUDE(V)	FREQZ(HZ)	MAGNITUDE(V)	FREQZ(HZ)	MAGNITUDE(V)
400	0.968982	2000	0.3867452	3600	0.01513559
800	0.8595407	2400	0.216103	4000	0.001197199
1200	0.7171181	2800	0.09005222	4400	0.000246351
1600	0.5717686	3200	0.006879495	4800	0.000147509

ตารางที่ 5.1.1 ขนาดของสัญญาณที่ความถี่ต่างๆของสัญญาณเอาท์พุทที่ได้จากวงจรวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์



รูปที่ 5.1.8 การเปรียบเทียบกราฟผลตอบสนองทางความถี่ที่ได้จากวิธีเจเนติก อัลกอริทึม และกราฟผลตอบสนองทางความถี่ที่ได้จากการใช้วินโดว์ฟังก์ชัน

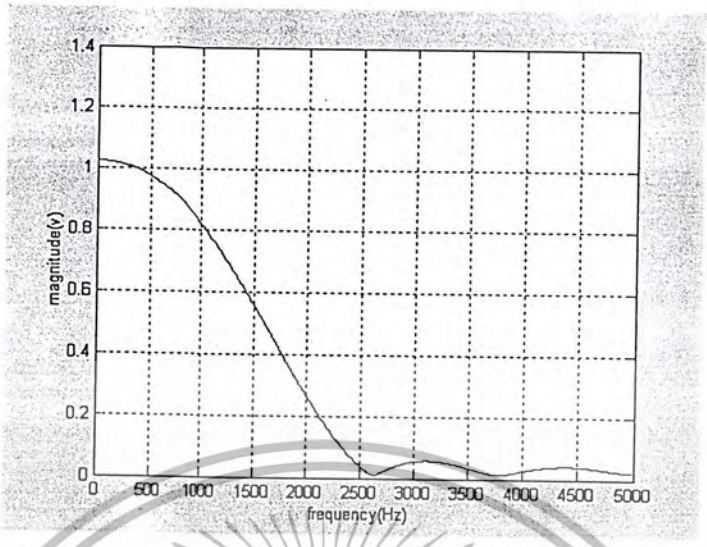
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รุ่นที่	ค่าความเหมาะสม	รุ่นที่	ค่าความเหมาะสม
10	415.799	260	459.671
20	420.117	270	459.671
30	423.839	280	459.672
40	448.982	290	459.701
50	449.127	300	459.701
60	449.903	310	459.758
70	450.827	320	459.758
80	450.966	330	459.811
90	451.51	340	459.811
100	453.769	350	459.818
110	454.081	360	459.819
120	454.083	370	459.82
130	454.084	380	459.82
140	454.657	390	459.823
150	454.695	400	459.831
160	456.022	410	459.832
170	456.108	420	459.833
180	458.097	430	459.838
190	458.83	440	459.848
200	459.122	450	459.849
210	459.565	460	459.85
220	459.579	470	459.85
230	459.624	480	459.851
240	459.624	490	459.851
250	459.658	500	459.851

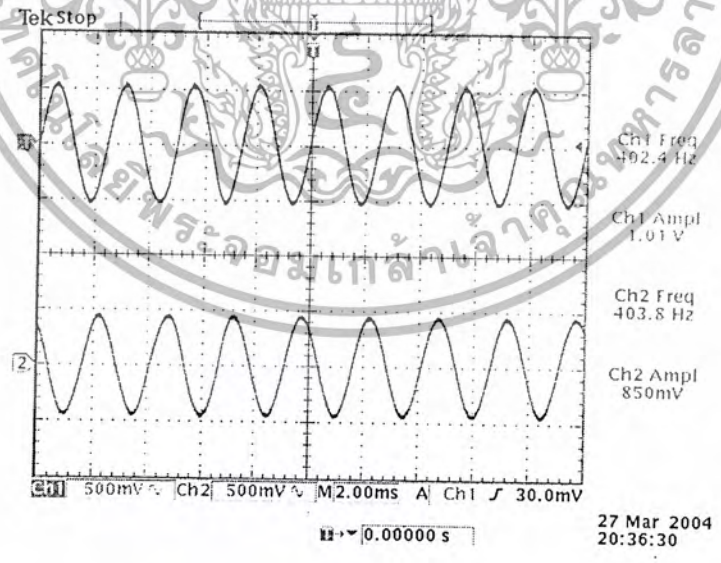
ตารางที่ 5.1.2 ค่าความเหมาะสมของค่าสัมประสิทธิ์ในแต่ละรุ่นของตัวกรองความถี่แบบเอฟไอาร์ที่ได้จากวิธีเจเนติกอัลกอริทึม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 วงจรกรองความถี่ต่ำผ่านอันดับ 7 ความถี่คัทออฟ 1 KHz

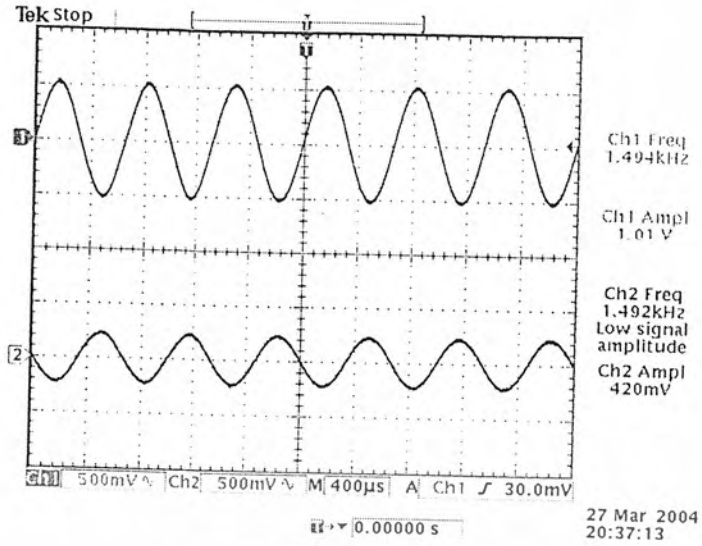


รูปที่ 5.2.1 กราฟผลตอบสนองทางความถี่ของตัวกรองแบบเอฟไอาร์โดยใช้เจเนติก อัลกอริทึมในการหาค่าสัมประสิทธิ์ของตัวกรองความถี่ และใช้ Matlab เขียนกราฟ

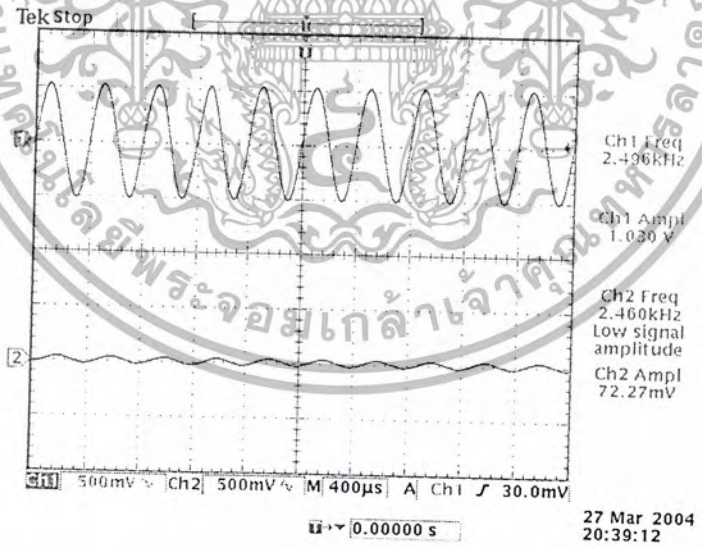


รูปที่ 5.2.2 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไอาร์ เทียบกับสัญญาณอินพุท
CH1 สัญญาณอินพุทที่เข้าวงจรกรองความถี่ต่ำผ่าน
CH2 สัญญาณเอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

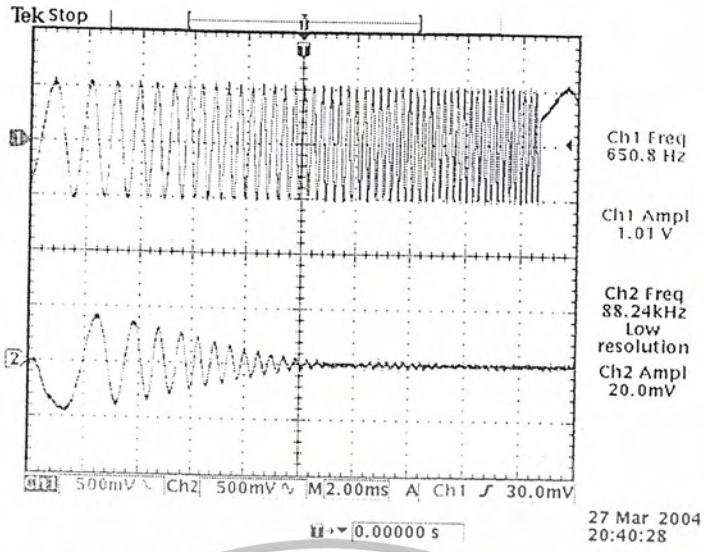


รูปที่ 5.2.3 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ เทียบกับสัญญาณอินพุท
CH1 สัญญาณอินพุทที่เข้าวงจรกรองความถี่ต่ำผ่านเมื่อปรับความถี่ให้สูงขึ้น
CH2 สัญญาณเอาท์พุท



รูปที่ 5.2.4 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ เทียบกับสัญญาณอินพุท
CH1 สัญญาณอินพุทที่เข้าวงจรกรองความถี่ต่ำผ่านเมื่อปรับความถี่ให้สูงขึ้น
CH2 สัญญาณเอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

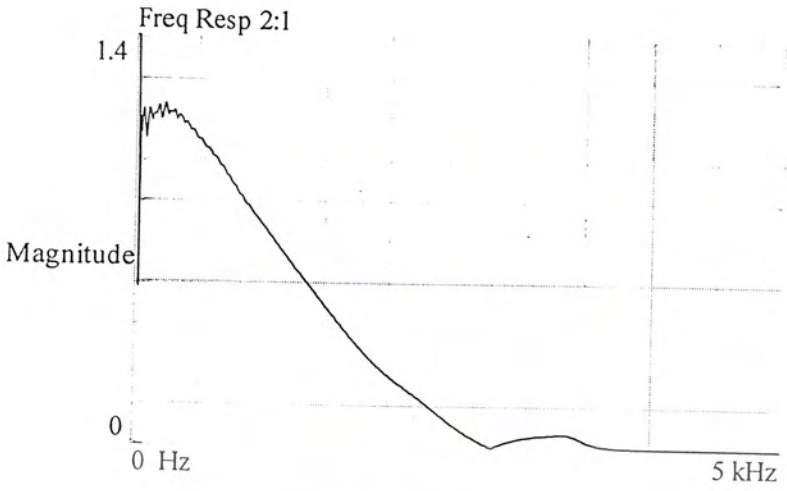


รูปที่ 5.2.5 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไอาร์ เทียบกับสัญญาณอินพุทที่เป็นแบบสวีฟ

CH1 สัญญาณอินพุทที่เข้าวงจรกรองความถี่ต่ำผ่าน

CH2 สัญญาณเอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.2.6 ผลตอบสนองทางความถี่ของสัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ซึ่งใช้ไดนามิกซิกแนลอนาไลเซอรในการบันทึกผล

รูปที่ 5.2.7 ผลตอบสนองทางเฟสของสัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ซึ่งใช้ไดนามิกซิกแนลอนาไลเซอร์ในการบันทึกผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FREQZ(HZ)	MAGNITUDE(V)	FREQZ(HZ)	MAGNITUDE(V)	FREQZ(HZ)	MAGNITUDE(V)
400	0.9553053	2000	0.4033531	3600	0.007583259
800	0.9577881	2400	0.15695	4000	0.000591717
1200	0.8987603	2800	0.01860586	4400	0.000934812
1600	0.6897272	3200	0.04204159	4800	0.000849764

ตารางที่ 5.2.1 ขนาดของสัญญาณที่ความถี่ต่างๆของสัญญาณเอาท์พุทที่ได้จากวงจรวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์

รูปที่ 5.2.8 การเปรียบเทียบกราฟผลตอบสนองทางความถี่ที่ได้จากวิธีเจเนติก อัลกอริทึม และกราฟผลตอบสนองทางความถี่ที่ได้จากการใช้วินโดว์ฟังก์ชัน

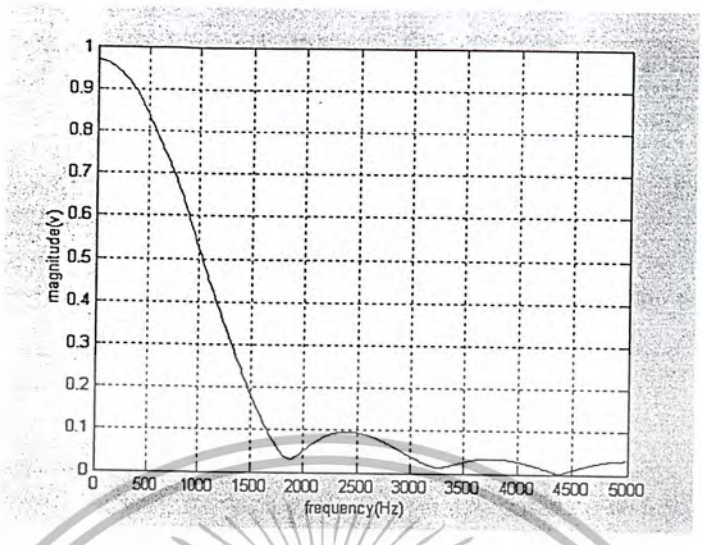
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รุ่นที่	ค่าความเหมาะสม	รุ่นที่	ค่าความเหมาะสม
10	409.444	260	456.031
20	413.879	270	456.031
30	414.886	280	456.176
40	427.808	290	456.176
50	430.308	300	456.176
60	430.317	310	456.179
70	439.552	320	456.18
80	439.552	330	456.18
90	441.518	340	456.204
100	445.49	350	456.242
110	451.012	360	456.244
120	453.139	370	456.245
130	453.582	380	456.245
140	454.304	390	456.254
150	454.478	400	456.29
160	455.589	410	456.303
170	455.754	420	456.762
180	455.755	430	458.018
190	455.819	440	458.62
200	455.948	450	458.62
210	456.011	460	458.734
220	456.029	470	458.738
230	456.03	480	459.129
240	456.031	490	459.129
250	456.031	500	459.359

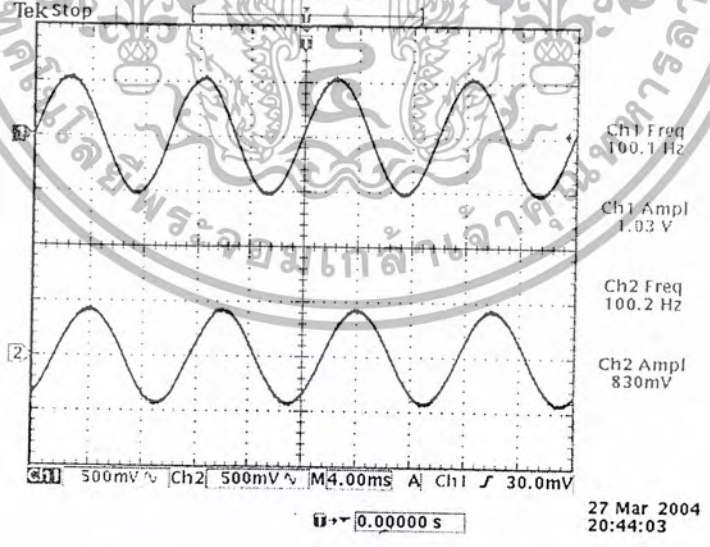
ตารางที่ 5.2.2 ค่าความเหมาะสมของค่าสัมประสิทธิ์ในแต่ละรุ่นของตัวกรองความถี่แบบเอฟไอาร์ที่ได้จากวิธีเจเนติกอัลกอริทึม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 วงจรกรองความถี่ต่ำผ่านอันดับ 9 ความถี่คัทออฟ 1 KHz



รูปที่ 5.3.1 กราฟผลตอบสนองทางความถี่ของตัวกรองแบบเอฟไออาร์โดยใช้เจเนติก อัลกอริทึมในการหาค่าสัมประสิทธิ์ของตัวกรองความถี่ และใช้ Matlab เขียนกราฟ

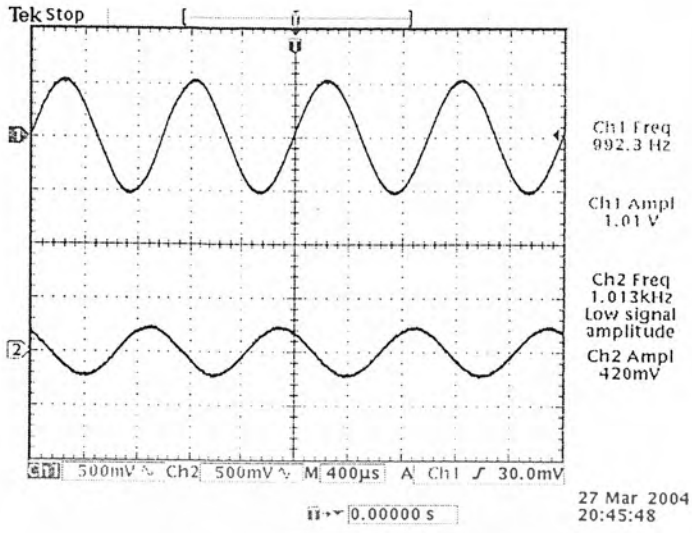


รูปที่ 5.3.2 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ เทียบกับสัญญาณอินพุท

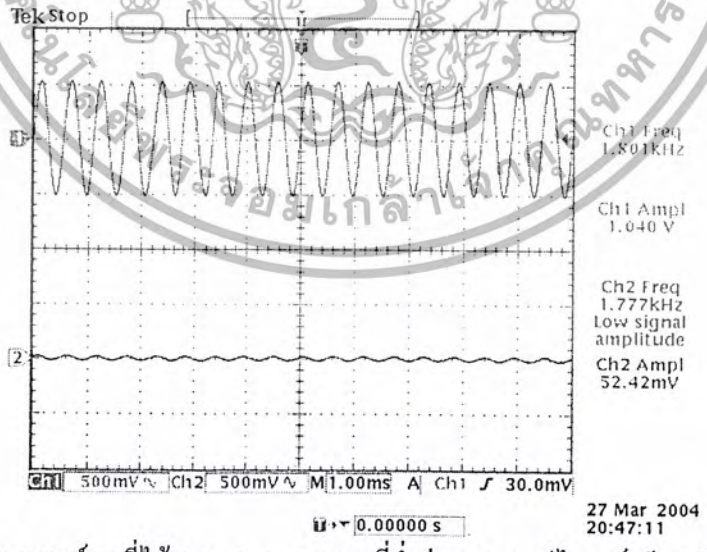
CH1 สัญญาณอินพุทที่เข้าวงจรกรองความถี่ต่ำผ่าน

CH2 สัญญาณเอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

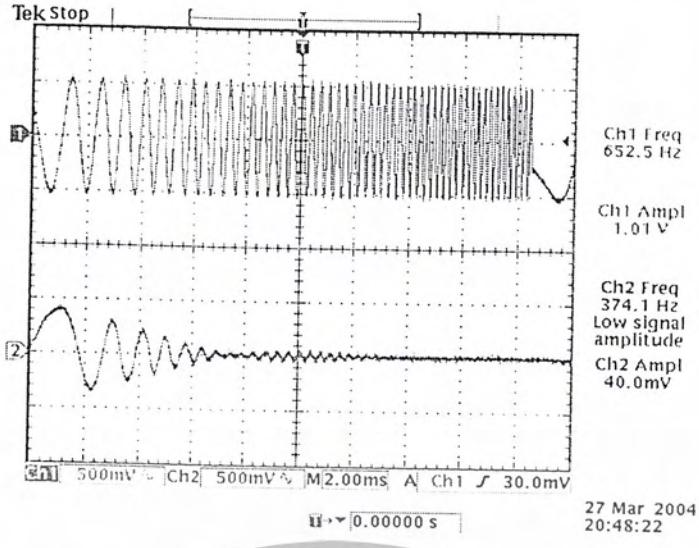


รูปที่ 5.3.3 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ เทียบกับสัญญาณอินพุท
CH1 สัญญาณอินพุทที่เข้าวงจรกรองความถี่ต่ำผ่านเมื่อปรับความถี่ให้สูงขึ้น
CH2 สัญญาณเอาท์พุท



รูปที่ 5.3.4 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ เทียบกับสัญญาณอินพุท
CH1 สัญญาณอินพุทที่เข้าวงจรกรองความถี่ต่ำผ่านเมื่อปรับความถี่ให้สูงขึ้น
CH2 สัญญาณเอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

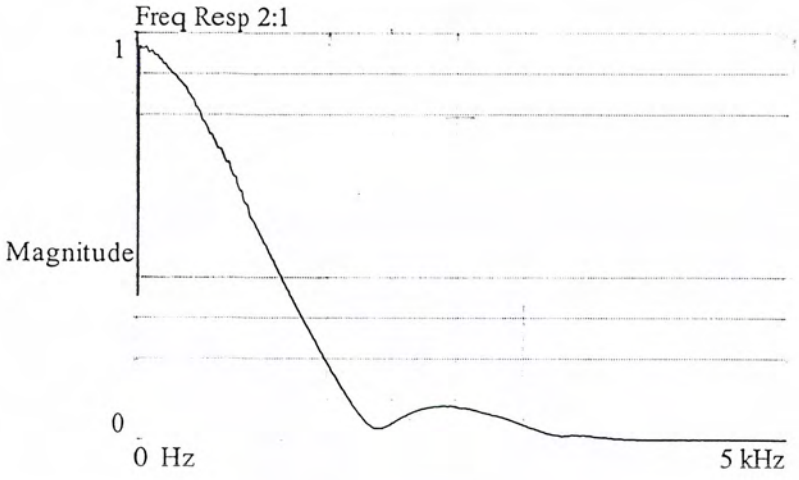


รูปที่ 5.3.5 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไอาร์ เทียบกับสัญญาณอินพุทที่เป็นแบบสวีฟ

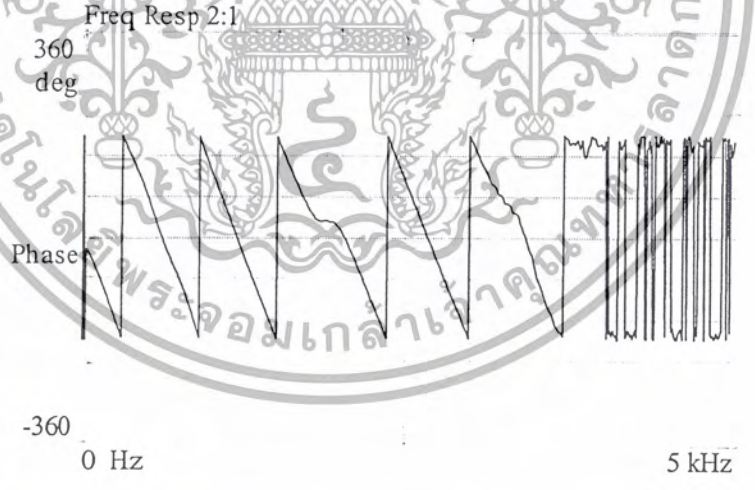
CH1 สัญญาณอินพุทที่เข้าวงจรกรองความถี่ต่ำผ่าน

CH2 สัญญาณเอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.3.6 ผลตอบสนองทางความถี่ของสัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ซึ่งใช้ไดนามิกซิกแนลอนาไลเซอรในการบันทึกผล

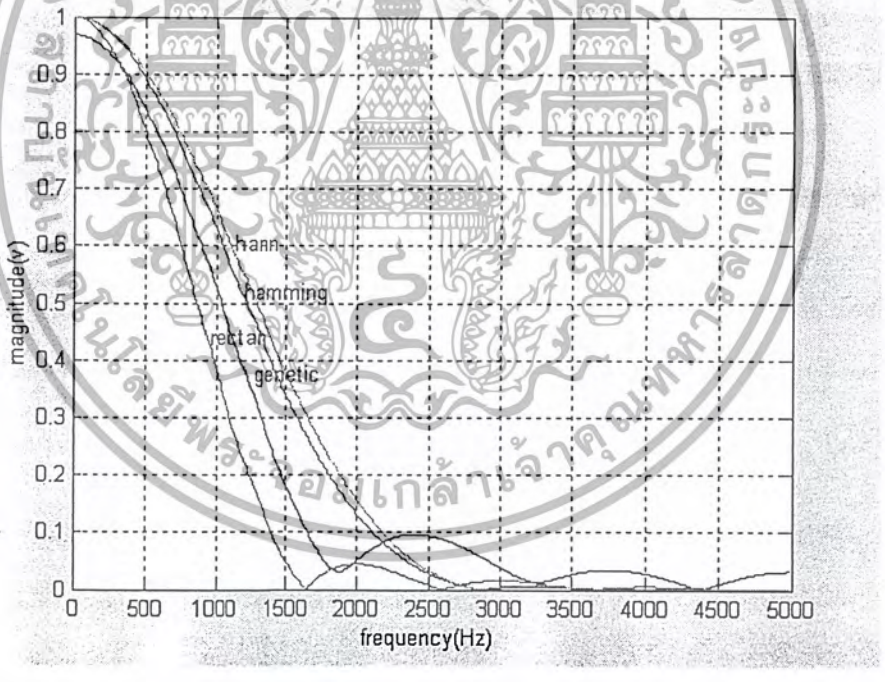


รูปที่ 5.3.7 ผลตอบสนองทางเฟสของสัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ซึ่งใช้ไดนามิกซิกแนลอนาไลเซอร์ในการบันทึกผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FREQZ(HZ)	MAGNITUDE(V)	FREQZ(HZ)	MAGNITUDE(V)	FREQZ(HZ)	MAGNITUDE(V)
400	0.8528893	2000	0.04060644	3600	0.007241216
800	0.620539	2400	0.08443012	4000	0.001362259
1200	0.3694893	2800	0.05995111	4400	0.001509635
1600	0.1373853	3200	0.01646882	4800	0.001672778

ตารางที่ 5.3.1 ขนาดของสัญญาณที่ความถี่ต่างๆของสัญญาณเอาท์พุทที่ได้จากวงจรวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์



รูปที่ 5.3.8 การเปรียบเทียบกราฟผลตอบสนองทางความถี่ที่ได้จากวิธีเจเนติก อัลกอริทึม และกราฟผลตอบสนองทางความถี่ที่ได้จากการใช้วินโดว์ฟังก์ชัน

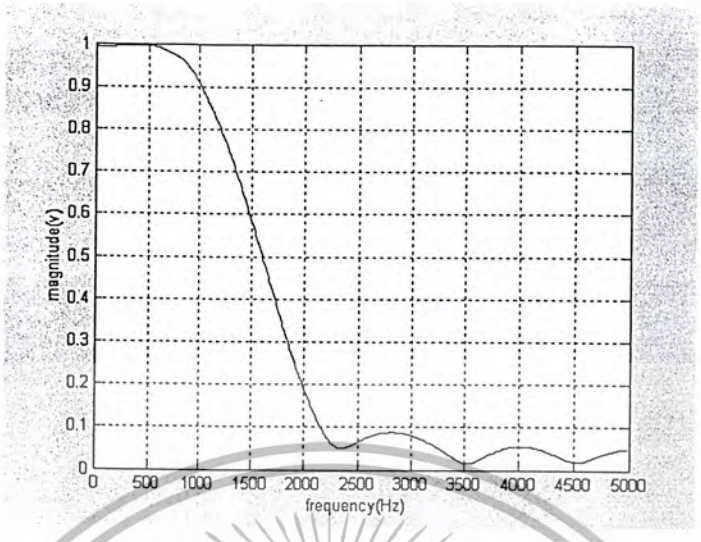
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รุ่นที่	ค่าความเหมาะสม	รุ่นที่	ค่าความเหมาะสม
10	343.324	260	471.954
20	369.707	270	472.471
30	383.828	280	472.471
40	385.214	290	472.471
50	400.287	300	472.471
60	405.685	310	472.471
70	408.694	320	472.479
80	415.596	330	472.492
90	429.067	340	472.777
100	430.082	350	472.829
110	434.587	360	472.909
120	442.713	370	474.039
130	454.806	380	474.1
140	455.927	390	474.108
150	455.928	400	474.108
160	456.043	410	474.589
170	457.829	420	474.633
180	469.894	430	474.633
190	470.809	440	474.679
200	470.809	450	474.704
210	471.296	460	474.715
220	471.296	470	474.786
230	471.296	480	474.79
240	471.794	490	474.79
250	471.884	500	474.79

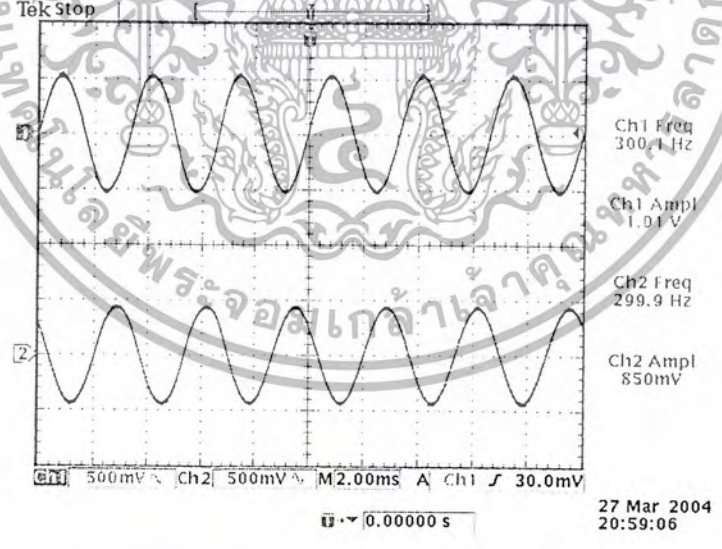
ตารางที่ 5.3.2 ค่าความเหมาะสมของค่าสัมประสิทธิ์ในแต่ละรุ่นของตัวกรองความถี่แบบเอฟไอาร์ที่ได้จากวิธีเจเนติกอัลกอริทึม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 วงจรกรองความถี่ต่ำผ่านอันดับ 11 ความถี่คัทออฟ 1 KHz



รูปที่ 5.4.1 กราฟผลตอบสนองทางความถี่ของตัวกรองแบบเอฟไอาร์โดยใช้เจเนติก อัลกอริทึมในการหาค่าสัมประสิทธิ์ของตัวกรองความถี่ และใช้ Matlab เขียนกราฟ

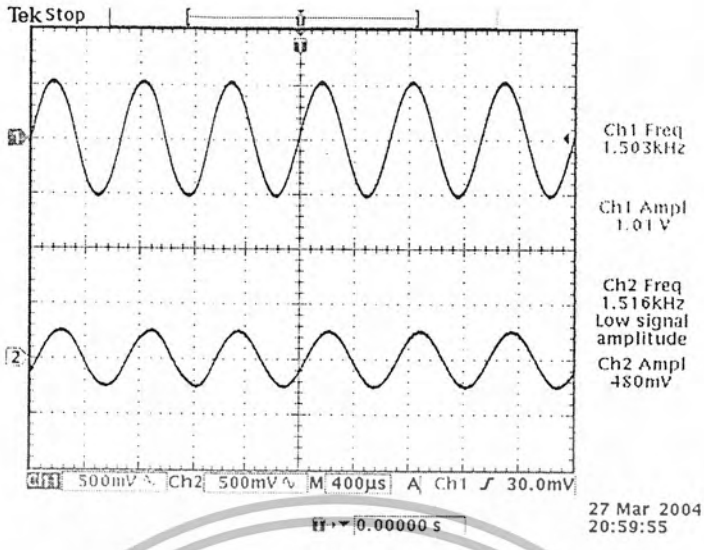


รูปที่ 5.4.2 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไอาร์ เทียบกับสัญญาณอินพุท

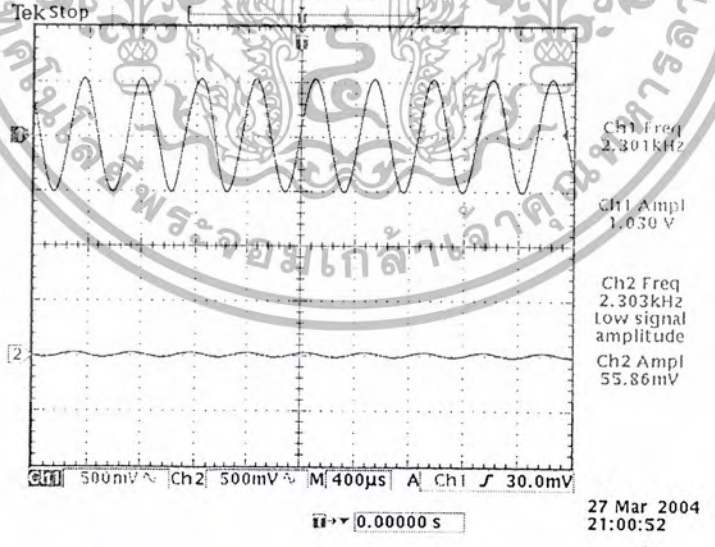
CH1 สัญญาณอินพุทที่เข้าวงจรกรองความถี่ต่ำผ่าน

CH2 สัญญาณเอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

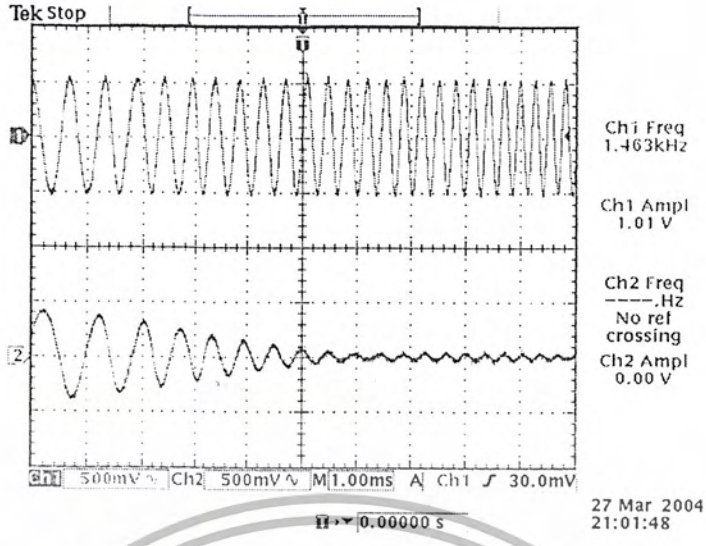


รูปที่ 5.4.3 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ เทียบกับสัญญาณอินพุท
CH1 สัญญาณอินพุทที่เข้าวงจรกรองความถี่ต่ำผ่านเมื่อปรับความถี่ให้สูงขึ้น
CH2 สัญญาณเอาท์พุท



รูปที่ 5.4.4 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ เทียบกับสัญญาณอินพุท
CH1 สัญญาณอินพุทที่เข้าวงจรกรองความถี่ต่ำผ่านเมื่อปรับความถี่ให้สูงขึ้น
CH2 สัญญาณเอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

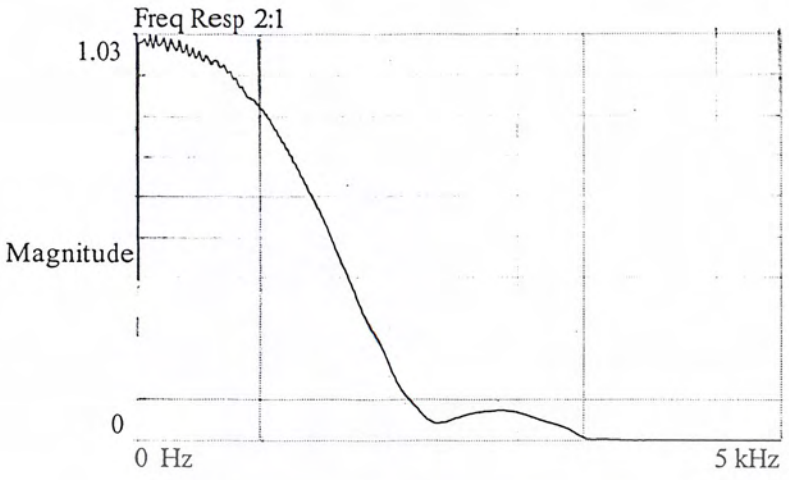


รูปที่ 5.4.5 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบแอคทีฟไอซี เทียบกับสัญญาณอินพุทที่เป็นแบบสวีฟ

CH1 สัญญาณอินพุทที่เข้าวงจรกรองความถี่ต่ำผ่าน

CH2 สัญญาณเอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



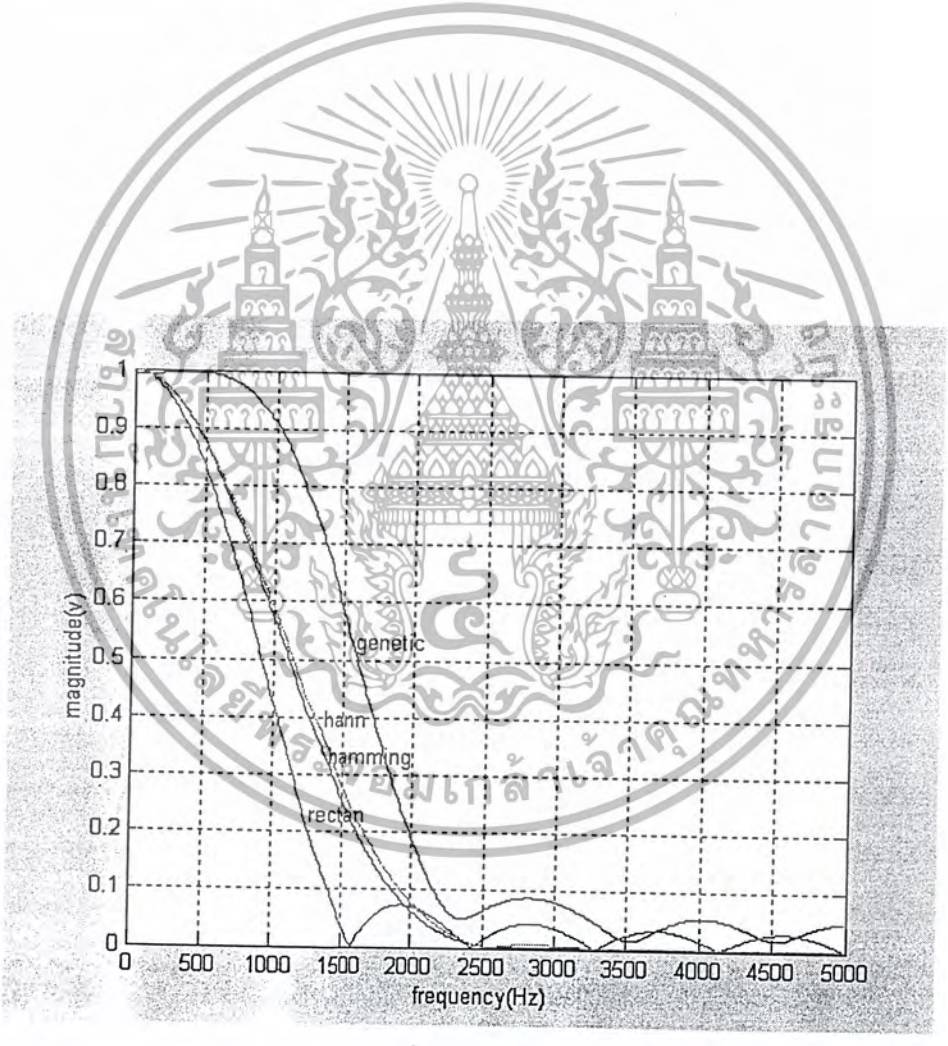
รูปที่ 5.4.6 ผลตอบสนองทางความถี่ของสัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ซึ่งใช้ไดนามิกซิกแนลอนาไลเซอรในการบันทึกผล

รูปที่ 5.4.7 ผลตอบสนองทางเฟสของสัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ซึ่งใช้ไดนามิกซิกแนลอนาไลเซอร์ในการบันทึกผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FREQZ(HZ)	MAGNITUDE(V)	FREQZ(HZ)	MAGNITUDE(V)	FREQZ(HZ)	MAGNITUDE(V)
400	0.9909549	2000	0.1905706	3600	0.003369061
800	0.9172781	2400	0.04546934	4000	0.001162688
1200	0.7521472	2800	0.07508202	4400	0.001105856
1600	0.4816023	3200	0.04986203	4800	0.001220086

ตารางที่ 5.4.1 ขนาดของสัญญาณที่ความถี่ต่างๆของสัญญาณเอาท์พุทที่ได้จากวงจรวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์



รูปที่ 5.4.8 การเปรียบเทียบกราฟผลตอบสนองทางความถี่ที่ได้จากวิธีเจเนติก อัลกอริทึม และกราฟผลตอบสนองทางความถี่ที่ได้จากการใช้วินโดว์ฟังก์ชัน

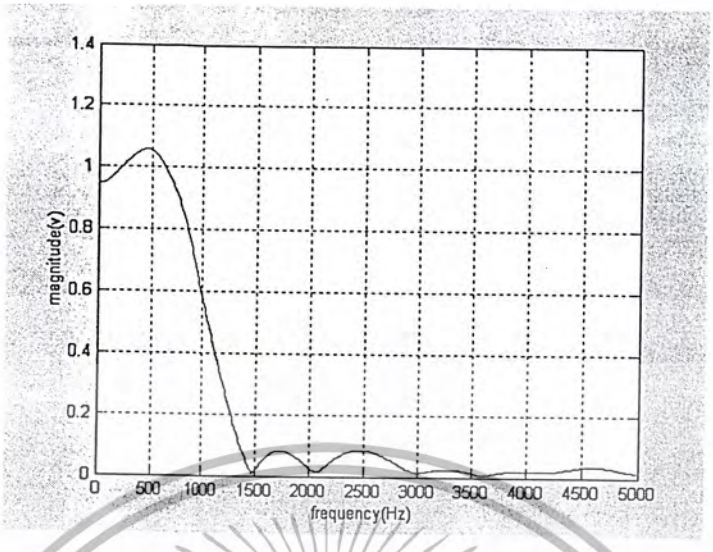
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รุ่นที่	ค่าความเหมาะสม	รุ่นที่	ค่าความเหมาะสม
10	334.24	260	458.365
20	381.872	270	458.79
30	393.12	280	460.169
40	393.955	290	460.177
50	394.057	300	461.312
60	400.023	310	461.417
70	403.188	320	463.836
80	404.298	330	464.417
90	409.237	340	464.417
100	430.472	350	464.417
110	431.877	360	464.418
120	433.789	370	464.701
130	434.36	380	465.189
140	438.458	390	465.467
150	439.099	400	465.825
160	444.34	410	466.13
170	448.056	420	466.495
180	448.317	430	466.495
190	451.171	440	467.3
200	451.492	450	467.691
210	452.859	460	469.061
220	452.859	470	469.977
230	453.792	480	470.546
240	456.578	490	470.546
250	458.279	500	470.607

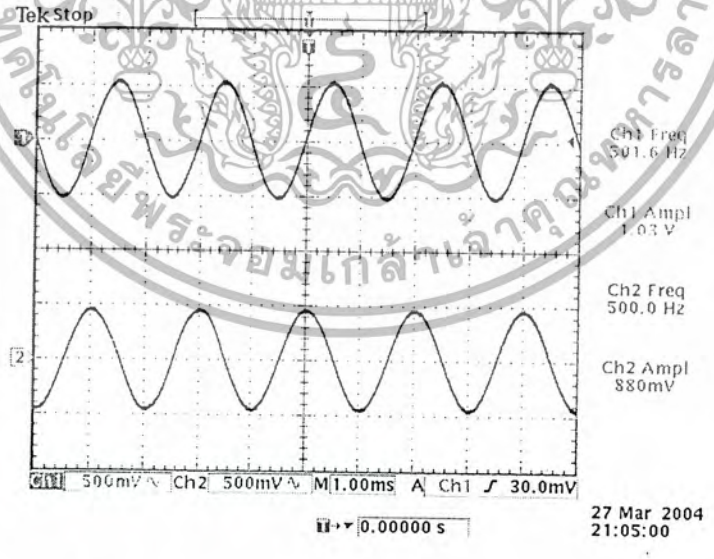
ตารางที่ 5.4.2 ค่าความเหมาะสมของค่าสัมประสิทธิ์ในแต่ละรุ่นของตัวกรองความถี่แบบเอฟไอาร์ที่ได้จากวิธีเจเนติกอัลกอริทึม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5 วงจรกรองความถี่ต่ำผ่านอันดับ 13 ความถี่คัทออฟ 1 KHz



รูปที่ 5.5.1 กราฟผลตอบสนองทางความถี่ของตัวกรองแบบเอฟไอาร์โดยใช้เจเนติก อัลกอริทึมในการหาค่าสัมประสิทธิ์ของตัวกรองความถี่ และใช้ Matlab เขียนกราฟ

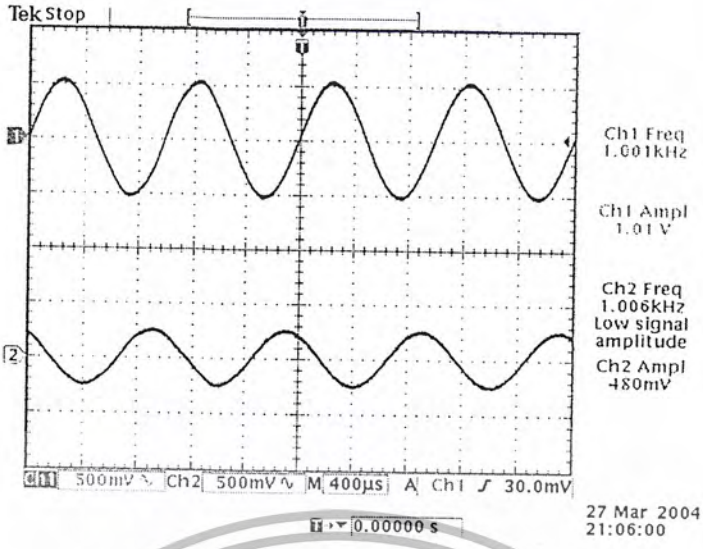


รูปที่ 5.5.2 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไอาร์ เทียบกับสัญญาณอินพุท

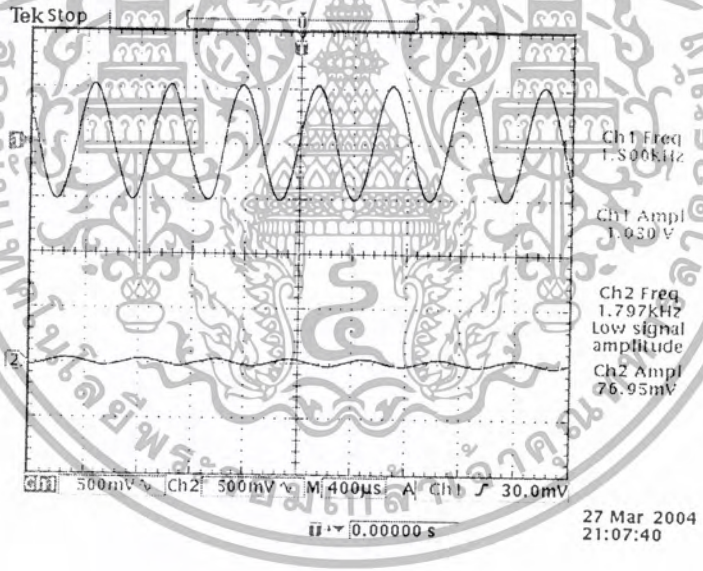
CH1 สัญญาณอินพุทที่เข้าวงจรกรองความถี่ต่ำผ่าน

CH2 สัญญาณเอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

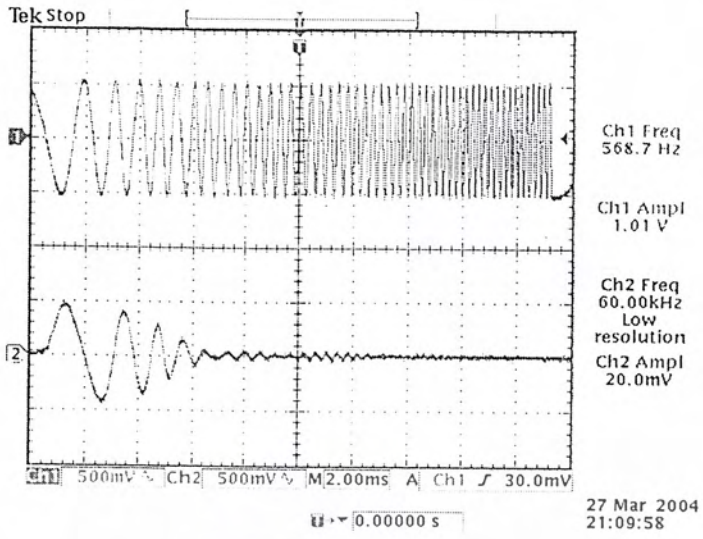


รูปที่ 5.5.3 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ เทียบกับสัญญาณอินพุท
CH1 สัญญาณอินพุทที่เข้าวงจรกรองความถี่ต่ำผ่านเมื่อปรับความถี่ให้สูงขึ้น
CH2 สัญญาณเอาท์พุท



รูปที่ 5.5.4 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ เทียบกับสัญญาณอินพุท
CH1 สัญญาณอินพุทที่เข้าวงจรกรองความถี่ต่ำผ่านเมื่อปรับความถี่ให้สูงขึ้น
CH2 สัญญาณเอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

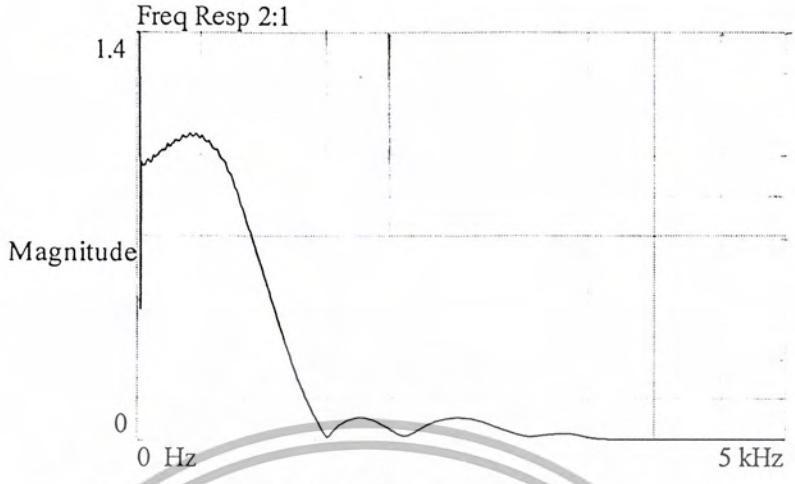


รูปที่ 5.5.5 สัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ เทียบกับสัญญาณอินพุทที่เป็นแบบสวีฟ

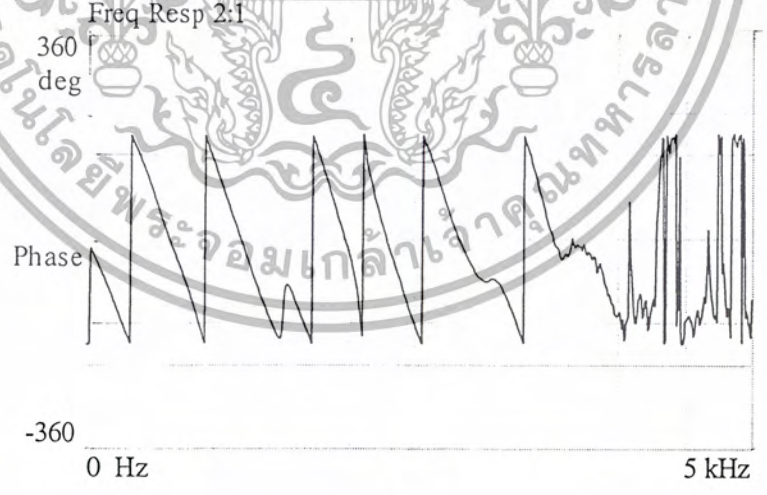
CH1 สัญญาณอินพุทที่เข้าวงจรกรองความถี่ต่ำผ่าน

CH2 สัญญาณเอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.5.6 ผลตอบสนองทางความถี่ของสัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ซึ่งใช้ไดนามิกซิกแนลอนาไลเซอร์ในการบันทึกผล

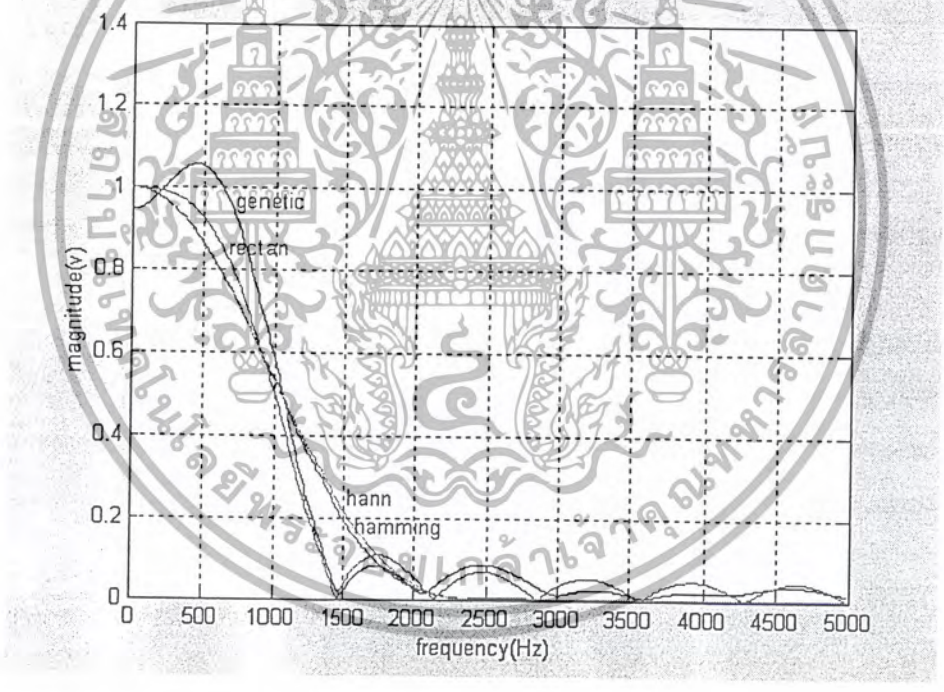


รูปที่ 5.5.7 ผลตอบสนองทางเฟสของสัญญาณเอาท์พุทที่ได้จากวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์ซึ่งใช้ไดนามิกซิกแนลอนาไลเซอร์ในการบันทึกผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FREQZ(HZ)	MAGNITUDE(V)	FREQZ(HZ)	MAGNITUDE(V)	FREQZ(HZ)	MAGNITUDE(V)
400	1.049816	2000	0.03001357	3600	0.001741459
800	0.8352123	2400	0.06884027	4000	0.000310302
1200	0.2926473	2800	0.04312752	4400	0.000140534
1600	0.04981179	3200	0.01699963	4800	9.83E-05

ตารางที่ 5.5.1 ขนาดของสัญญาณที่ความถี่ต่างๆของสัญญาณเอาท์พุทที่ได้จากวงจรวงจรกรองความถี่ต่ำผ่านแบบเอฟไออาร์



รูปที่ 5.5.8 การเปรียบเทียบกราฟผลตอบสนองทางความถี่ที่ได้จากวิธีเจเนติก อัลกอริทึม และกราฟผลตอบสนองทางความถี่ที่ได้จากการใช้วินโดว์ฟังก์ชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รุ่นที่	ค่าความเหมาะสม	รุ่นที่	ค่าความเหมาะสม
10	337.888	260	483.17
20	337.934	270	483.265
30	337.934	280	484.293
40	338.513	290	485.708
50	387.987	300	487.333
60	387.987	310	488.058
70	396.958	320	488.29
80	397.298	330	489.438
90	397.298	340	489.469
100	397.903	350	489.584
110	413.14	360	489.584
120	413.519	370	489.584
130	414.018	380	491.402
140	418.382	390	491.402
150	435.23	400	491.856
160	444.091	410	492.178
170	444.091	420	492.178
180	451.558	430	492.178
190	465.636	440	492.836
200	465.636	450	492.847
210	466.328	460	492.847
220	472.909	470	492.847
230	473.476	480	493.083
240	480.104	490	493.083
250	482.238	500	493.089

ตารางที่ 5.5.2 ค่าความเหมาะสมของค่าสัมประสิทธิ์ในแต่ละรุ่นของตัวกรองความถี่แบบเอฟไออาร์ที่ได้จากวิธีเจเนติกอัลกอริทึม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

บทวิจารณ์และบทสรุป

1. การวนหาค่าสัมประสิทธิ์ของตัวกรองความถี่เมื่อเราวนหาค่าสัมประสิทธิ์ด้วยจำนวนรุ่นที่มากขึ้น หรือที่อันดับที่ของตัวกรองมากขึ้น จะใช้เวลาในการประมวลผลนานขึ้น แต่จะได้ค่าสัมประสิทธิ์ที่ดีขึ้น จากการทดลองที่อันดับ 7 ค่าสัมประสิทธิ์รุ่นที่ 2000 ใช้เวลาในการประมวลผลนานประมาณ 45 นาที และมีรูปผลตอบสนองความถี่ที่ใกล้เคียงกับอุดมคติมากขึ้น

2. สำหรับในกรณีที่รูปกราฟที่ได้คล้ายกับรูปกราฟผลตอบสนองทางอุดมคติแต่ไม่เหมือนเป็นเพราะการเกิดค่าการผิดพลาดจากการ mapping จากสัมประสิทธิ์ที่เป็นค่าเลขฐาน 2 ที่ใช้พิจารณาเป็นเลขฐาน 10 ซึ่งอาจได้ค่าใกล้เคียงแต่อาจไม่เหมือนกับเลขฐาน 10 ที่แท้จริง อาจแก้ได้โดยการเพิ่มจำนวนบิทขึ้น

3. เจเนติก อัลกอริทึมแบบ simple genetic จะเหมาะกับการหาค่าสัมประสิทธิ์ของตัวกรองที่มีอันดับต่ำ ๆ และไม่ควรเกินอันดับที่ 13 ถ้าเราต้องการหาค่าสัมประสิทธิ์ของตัวกรองความถี่ที่มีอันดับสูงๆ จะต้องเลือกรูปแบบของเจเนติกรูปแบบอื่น เพื่อช่วยในการหาค่าสัมประสิทธิ์ที่เหมาะสมกับตัวกรองความถี่มากขึ้น

สรุปผลการทดลอง

จากการทดลองพบว่าการใช้วิธีการเจเนติกอัลกอริทึม ซึ่งเป็นวิธีการค้นหาคำตอบที่ดีที่สุดมาทำการหาสัมประสิทธิ์ของวงจรกรองสัญญาณเชิงเลขชนิดผลตอบสนองอิมพัลส์จำกัดชนิดนั้น เมื่อนำสัมประสิทธิ์ดังกล่าวที่ได้มา plot เป็นกราฟโดยใช้โปรแกรม matlab และเปรียบเทียบกราฟดังกล่าวกับกราฟในทางอุดมคติซึ่งจะเห็นว่ามีลักษณะใกล้เคียงกันมากขึ้นเมื่อใช้รุ่นของตัวกรองความถี่ที่มากขึ้น แสดงว่าวิธีเจเนติกอัลกอริทึมนั้นสามารถใช้เป็นวิธีการหนึ่งในการออกแบบวงจรกรองสัญญาณเชิงเลขชนิดผลตอบสนองอิมพัลส์จำกัดได้จริงและให้ผลการทดลองใกล้เคียงกับค่าทางทฤษฎี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. กิตติ ไพฑูรย์วัฒนกิจ . กาญจนี วงศ์วิภาพร , การจัดตารางเรียนแบบจำลองโดยเจเนติกอัลกอริทึม วารสารวิศวกรรมสาร , ปีที่ 46 , เล่มที่ 12 , ธันวาคม 2536 , หน้า 81-84.
2. กิตติ ไพฑูรย์วัฒนกิจ . กาญจนี วงศ์วิภาพร , การปรับปรุงสมรรถนะ เจเนติก อัลกอริทึม วารสารสำนักงานคณะกรรมการวิจัยแห่งชาติ , ปีที่ 27 , เล่มที่ 1 , ธันวาคม – มิถุนายน 2538 , หน้า 1-33.
3. อาทิตย์ ศรีแก้ว เจเนติกอัลกอริทึม ตอนที่ 1 วารสารเทคโนโลยีสุรนารี , ปีที่ 19 ฉบับที่ 1 , มกราคม – มีนาคม 2545, หน้า 69-83.
4. อาทิตย์ ศรีแก้ว เจเนติกอัลกอริทึม ตอนที่ 2 : การค้นหาสัมประสิทธิ์ของอนุกรมฟูเรียร์ วารสารเทคโนโลยีสุรนารี , ปีที่ 9 ฉบับที่ 2 , มกราคม – มีนาคม 2545 , หน้า 139-146.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

ก. ส่วนของโปรแกรม Matlab ที่ใช้ในการหาค่าผลตอบสนองทางความถี่ โดยใช้ Window Function

```
b=fir1(4,0.2,rectwin(5));
[mag,q]=freqz(b,1,512,10000);
plot(q,abs(mag));
hold on;
b=fir1(4,0.2,hamming(5));
[mag,q]=freqz(b,1,512,10000);
plot(q,abs(mag));
hold on;
b=fir1(4,0.2,hann(5));
[mag,q]=freqz(b,1,512,10000);
plot(q,abs(mag));
gtext('rectan');
gtext('hamming');
gtext('hann');
xlabel('frequency(Hz)');
ylabel('magnitude(V)');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข. ส่วนของโปรแกรมภาษา C ที่ใช้ในการหาค่าสัมประสิทธิ์ของตัวกรองความถี่โดยใช้วิธีเจเนติก อัลกอริทึม

```
#include <stdio.h>
#include <iostream.h>
#include <fstream.h>
#include <math.h>
#define NOW_ORDER   5
#define MAX_RESULT   512
#define NUM_OF_POPULATION 50
#define NUM_OF_GENERATION 500
#define PI 3.141592653589793
#define FS 2
#define FC 0.2
      struct myc {
            double r;
            double i;
      };
      struct mc {
            double i;
            double j;
      };
void convertmc2myc (struct myc* mycout, struct mc* mcin){
    mycout->r = mcin->i * cos(mcin->j);
    mycout->i = mcin->i * sin(mcin->j);
}
void addmyc (struct myc* mycout, struct myc* cin1, struct myc* cin2){
    mycout->r = cin1->r + cin2->r;
    mycout->i = cin1->i + cin2->i;
}
void setmyc (struct myc* mycout, double r, double i){
    mycout->r = r;
    mycout->i = i;
}

void setmc (struct mc* mycout, double i, double j){
    mycout->i = i;
    mycout->j = j;
}
double myabs (struct myc* mycin){
    return (sqrt(((mycin->r)*(mycin->r))+((mycin->i)*(mycin->i))));
}
float Objective(GAGenome &);
int
main(int argc, char **argv)
{
  cout << "Example \n\n";
  cout << "This program generates a sequence of random numbers then
uses\n";
  cout << "a simple GA and binary-to-decimal genome to match the\n";
  cout << "sequence.\n\n";
  unsigned int seed = 0;
  for(int ii=1; ii<argc; ii++) {
    if(strcmp(argv[ii++],"seed") == 0) {
      seed = atoi(argv[ii]);
    }
  }
  int popsize  = NUM_OF_POPULATION;
  int ngen     = NUM_OF_GENERATION;
  float pmut   = 0.01;
  float pcross = 0.9;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  GARandomSeed(seed);
  float *target = new float[MAX_RESULT];
  int i;
  double gap;
  gap = (1.0*FS/2)/MAX_RESULT;
  for(i=0; i<MAX_RESULT; i++)
    target[i] = i*gap;
  cout << "input sequence:\n";

  for(i=0; i<MAX_RESULT; i++){
    cout.width(10);
    cout << target[i] << " ";
  }
  cout << "\n"; cout.flush();
  GABin2DecPhenotype map;
  for(i=0; i<NOW_ORDER; i++)
    map.add(31, -1, 1);
  GABin2DecGenome genome(map, Objective, (void *)target);
  GASimpleGA ga(genome);
  ga.populationSize(popsize);
  ga.nGenerations(ngen);
  ga.pMutation(pmut);
  ga.pCrossover(pcross);
  ga.scoreFilename("bog.dat");
  ga.flushFrequency(50);
  ga.evolve(seed);
  char filename1[128] = "outh.txt";
  ofstream outfile;
  outfile.open(filename1, ios :: out);
  if(!outfile){
    cerr << "Cannot open " << filename1 << " for output.\n";
    exit(1);
  }
  outfile << map.nPhenotypes() <<" "<< MAX_RESULT <<" "<<
NUM_OF_GENERATION <<" "<< FS <<" "<< FC <<" : ";
  genome = ga.statistics().bestIndividual();
  cout << "the ga generated:\n";
  for(i=0; i<map.nPhenotypes(); i++){
    cout.width(10); cout << genome.phenotype(i) << " ";
    outfile << genome.phenotype(i) << " , ";
  }
  cout << "\n\n"; cout.flush();
  outfile.close();
  cout << "best of generation data are in 'outh.txt'\n";
// Clean up by freeing the memory we allocated.
  delete [ ] target;
  return 0;
}
float
Objective(GAGenome& g)
{
  GABin2DecGenome & genome = (GABin2DecGenome &)g;
  float *sequence = (float *)g.userData();
  float value=genome.nPhenotypes();
      myc c;
      mc cc;
      myc tresult;
    int i;
    int j;
      double t;
      for (i=0;i<MAX_RESULT;i++){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            t = sequence[i] * 2 * PI / FS;
        setmyc(&tresult,0,0);
          for (j=0;j<NOW_ORDER;j++){
              setmc(&cc,genome.phenotype(j),(j*t));
              convertmc2myc(&c,&cc);
              addmyc(&tresult,&tresult,&c);
              }
           value += 1.0 / (1.0 + fabs(myabs(&tresult) - (sequence[i] >
FC ? 0 : 1)));
        }
  return value;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค. ส่วนของโปรแกรมภาษา C ซึ่งใช้ในการทดสอบการนำสัญญาณเข้าออก

```
#include "c:\dsk\dsk3\aiccomc.c"                    /*AIC comm routines */
#include "c:\dsk\dsk3\math.h"                      /*math library function*/
int AICSEC[4]={0x162c,0x1,0x3872,0x63};
void ad()
{
    int input,output,n,x[13];
      float acc;
    float H[13]={ 0.0643336 , 0.116581 , 0.166626 , 0.191553 ,
0.210153,0.185914, 0.124935, 0.0628307, 0.0104448, -0.029099,
-0.0633121, -0.0625393, -0.0326356};
      while(1)
      {
            input=UPDATE_SAMPLE(output);
            x[0]=input;
            acc=0;
                  for(n=0;n<=12;n++)
                acc+=H[n]*x[n];
                  for(n=12;n>0;n--)
                x[n]=x[n-1];
                output=acc;
      }
}
main()
{
      AICSET();
      ad();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Texas Instruments

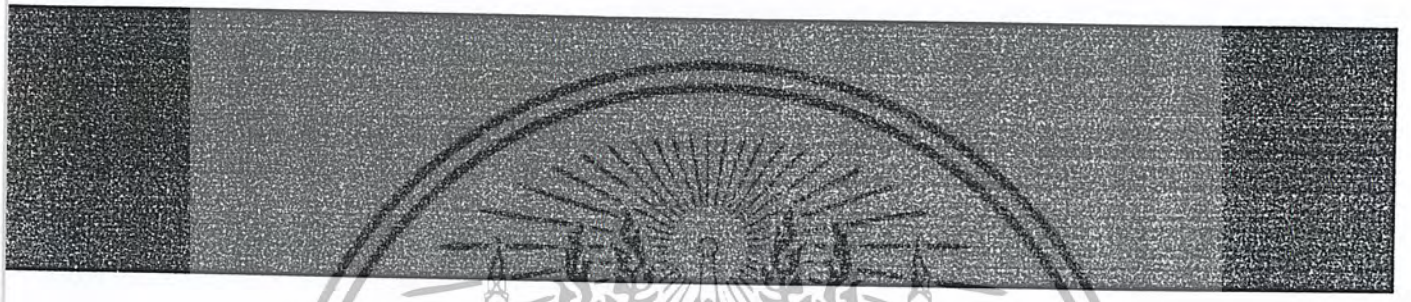
EDRAM™ Memory Controller for the TMS320C31 DSP

Application Report

1998

Digital Signal Processor Solutions

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



TEXAS INSTRUMENTS

Printed in U.S.A., January 1998

SPRA172

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EDRAM™ Memory Controller for the TMS320C31 DSP

Enhanced Memory Systems
1850 Ramtron Drive, Colorado Springs, CO 80921
web: http://www.edram.com

SPRA172
January 1998

Printed on Recycled Paper

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

Copyright © 1998, Texas Instruments Incorporated

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Contents

1 EDRAM Controller Design 2

2 EDRAM Controller Functional Description 5

3 Summary 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

List of Figures

1 TMS320C31 System Block Diagram 3
2 EDRAM Controller Block Diagram 5
3 Address Multiplexer Using Two 74FCT541CT Buffers 6
4 EDRAM Controller State Machine 6
5 TMS320C31 Read-Miss and Read-Hit Cycles @ 50 MHz 14
6 TMS320C31 Refresh After Idle Cycle @ 50 MHz 16
7 TMS320C31 Write Cycle – 50 MHz 18

List of Tables

1 TMS320C31 Read-Miss and Read-Hit Cycles @ 50 MHz 15
2 TMS320C31 Refresh After Idle Cycle @ 50 MHz 17
3 TMS320C31 Write Cycle – 50 MHz 19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EDRAM™ Memory Controller for the TMS320C31 DSP

ABSTRACT

This report provides an overview of a controller chip set to improve processor memory management and transactions while using the Texas Instruments (TI™) TMS320C31 digital signal processor (DSP). This application uses a minimum number of wait states which increases processor throughput.

TI is a trademark of Texas Instruments Incorporated
EDRAM is a registered trademark of Enhanced Memory Systems, Inc..

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1 EDRAM Controller Design

This Enhanced DRAM controller design supports TMS320C31 DSP memory transactions with a minimum of wait states. This EDRAM controller is designed to support one EDRAM bank of 2M bytes; however, memory size can expanded by adding additional control outputs to the logic. Figure 1 shows a functional block diagram of the TMS320C31 DSP System. The TMS320C31 DSP primary-bus interface can be programmed to control wait states and hold operations. The memory-control register for the local bus is programmed with the following parameters:

- SWW is programmed to respond only to the external $\overline{RDY}$ input for wait states (SWW=00). The internal $\overline{RDY}_{wtcnt}$ is ignored.
- WTCNT does not need to be programmed since software programmable wait states are not used.
- BNKCMP is set as shown below:

EDRAM	MSBs Defining a Bank	Bank Size	BNKCMP
512K x 8	23–8	256	10000

The TMS320C31 supports the following memory transactions:

- 32-bit page-read hit
- 32-bit page-read miss
- 32-bit write (hit or miss)

To support these bus transactions, the EDRAM controller must interface with the following processor control and address signals:

- A18–A0 – address bus
- $\overline{STRB}$ – external-access strobe
- R/$\overline{W}$ – read-write-mode outputs
- $\overline{RDY}$ – ready input
- $\overline{RESET}$ – reset input
- H1CLK – processor-clock output
- TCLK0 – timer zero output clock/pulse

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

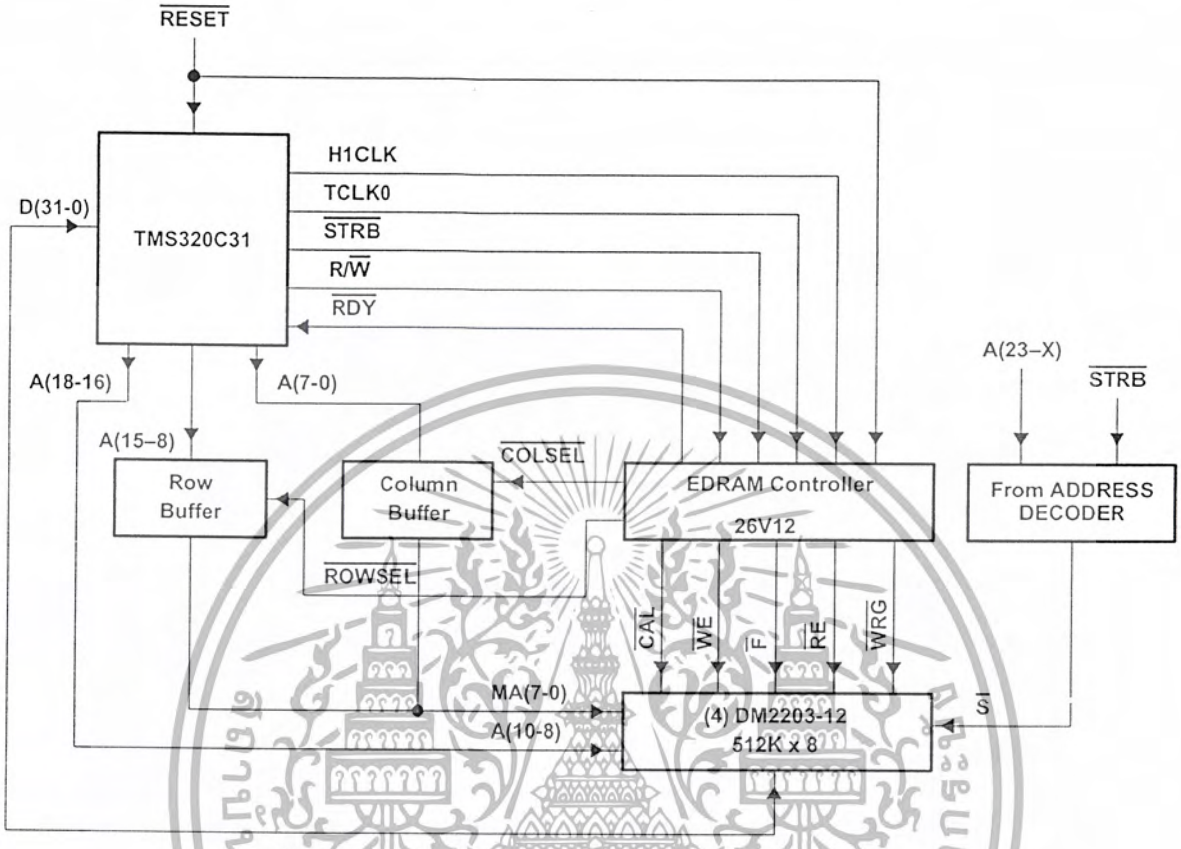


Figure 1. TMS320C31 System Block Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The controller generates the following signals to control the EDRAM:

- $\overline{\text{ROWSEL}}$ – multiplexed row-address enable
- $\overline{\text{COLSEL}}$ – multiplexed column-address enable
- $\overline{\text{RE}}$ – row enable
- $\overline{\text{CAL}}$ – column-address strobe
- $\overline{\text{WE}}$ – write enable
- $\overline{\text{WRG}}$ – combined W/$\overline{\text{R}}$ and $\overline{\text{G}}$
- $\overline{\text{F}}$ – refresh
- $\overline{\text{S}}$ – chip select

The EDRAM W/$\overline{\text{R}}$ and $\overline{\text{G}}$ control lines are tied together to form $\overline{\text{WRG}}$. The $\overline{\text{ROWSEL}}$, $\overline{\text{COLSEL}}$, $\overline{\text{CAL}}$, $\overline{\text{WE}}$, $\overline{\text{WRG}}$, and $\overline{\text{F}}$ logic signals are supplied from the 26V12 PLD.

$\overline{\text{S}}$ is an enable pin for the EDRAM and should be active for the desired page(s) of DSP memory. Because of the limited number of outputs of the PAL26V12, $\overline{\text{S}}$ is not included within the PAL. A simple decode might be to connect $\overline{\text{STRB}}$ to $\overline{\text{S}}$ where all external memory accesses enable the EDRAM, or in combination with an upper address line. Note that disabling the EDRAM will also lower the EDRAM power consumption. Optionally, the $\overline{\text{S}}$ input signal can be tied to ground if EDRAM power consumption is not an issue.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2 EDRAM Controller Functional Description

This section describes the EDRAM controller block diagram shown in Figure 2. The refresh counter is formed by using an internal timer in the TMS320C31 DSP to generate a refresh signal every 64 µs for the state machine. This refresh signal triggers an $\overline{F}$ refresh on the next available bus cycle. Up to three wait states can be inserted for the processor while a bus transaction is occurring.

Figure 3, the address multiplexer, selects a row and a column address for the EDRAM multiplex address lines under control of the state machine. The multiplexer is implemented using the two 74FCT541CT 8-bit buffer chips. These chips were selected for their short throughput delay time (4.1 ns) and short select-delay time (5.8 ns), respectively. A column address is connected to pins A7–A0 for 512K x 8 EDRAM. The row address comprises the 11 local address bits above the column address. Address lines A10–A8 of the EDRAM are connected directly to A18–A16 of the TMS320C31 DSP. During column select times, the upper address lines are ignored so there is no need to feed these address lines through a buffer.

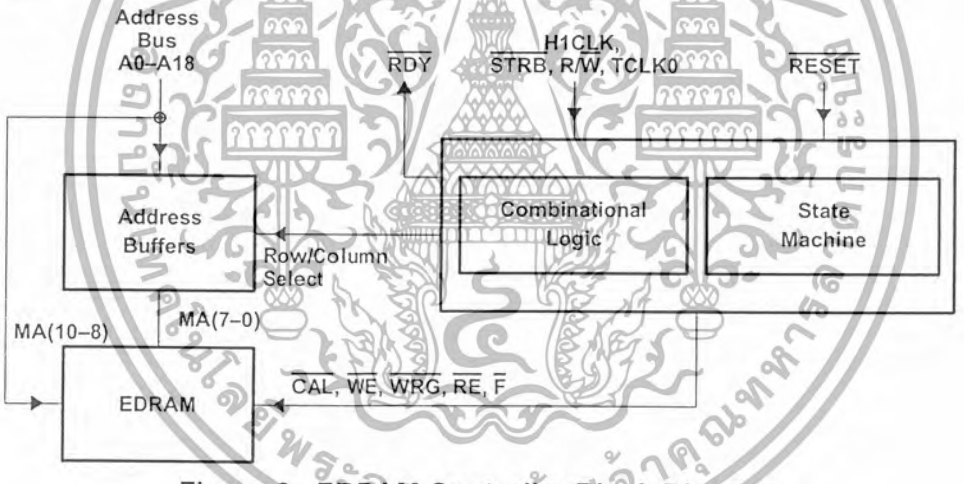


Figure 2. EDRAM Controller Block Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

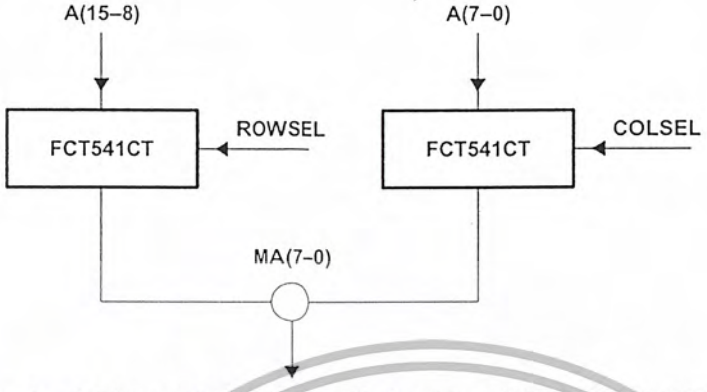


Figure 3. Address Multiplexer Using Two 74FCT541CT Buffers

The state machine operates synchronously with the rising edge of H1CLK.

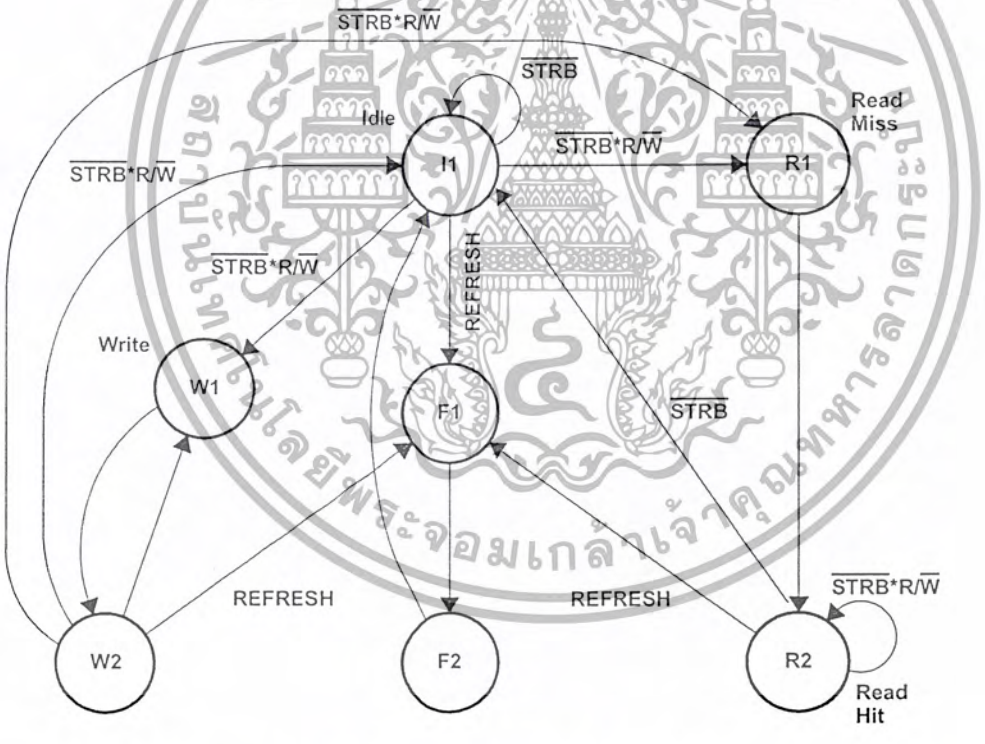


Figure 4. EDRAM Controller State Machine

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The following memory-control sequences are selected, based on control-input-line status. Read and write sequences are described. All logic states are determined at the rising edge of H1CLK.

- Reset Sequence – When $\overline{\text{RESET}}$ is at logic 0, the processor is in the reset state. The EDRAM controller continuously executes $\overline{\text{F}}$ refresh cycles until $\overline{\text{RESET}}$ returns to logic 1 level. This meets the EDRAM initialization requirements of eight $\overline{\text{F}}$ refresh cycles during start up. It is assumed that a startup control program is run immediately following a reset. This program must configure the TMS320C31 to perform two read-miss cycles to different row addresses for each internal EDRAM bank. Row address bits A8 and A9 define the four internal EDRAM banks. This startup procedure must be performed for proper EDRAM operation.
- Page-read-miss sequence – Starting from the idle state, when $\overline{\text{STRB}}$ is at logic 0 and R/$\overline{\text{W}}$ is at logic 1, a page-read-miss sequence is executed. The row address is selected, and $\overline{\text{WRG}}$ signal is set to logic 0. The $\overline{\text{RE}}$ signal is then clocked to load a new page of memory into the internal DRAM cache column, which takes 30-ns. $\overline{\text{RDY}}$ is brought to logic 1 at the start of cycle R1 to insert a wait state for the processor. Data is available to the processor at the end of cycle two. Because of the TMS320C31 addressing system, one wait state is inserted when a read cycle follows a write cycle.
- Page-read-hit sequence – Starting from the R1 state, when $\overline{\text{STRB}}$ is at logic 0 and R/$\overline{\text{W}}$ is logic 1, a page-read-hit sequence is executed. The column address is selected, and $\overline{\text{WRG}}$ is brought to logic 0 to gate data onto the data bus. The EDRAM can support back-to-back page reads from cache in zero wait states. The TMS320C31 processor holds $\overline{\text{STRB}}$ at logic 0 during back-to-back page hit reads since the programmable bank switching feature is being used. On a page-read miss, $\overline{\text{STRB}}$ goes to logic 1, which also causes an idle state to be inserted.
- Write sequence – If $\overline{\text{STRB}}$ is at logic 0 and R/$\overline{\text{W}}$ is also at logic 0, a write sequence is executed. The row address is selected and write-data operation is stable and the $\overline{\text{WE}}$ signal is clocked to post the write data operation into an internal latch. When $\overline{\text{CAL}}$ falls, write data is written to the EDRAM array. Following the write-data operation, $\overline{\text{RE}}$ is brought to logic 1 to terminate the write-data cycle. The write operation is completed in two cycles or zero wait states. The EDRAM supports back-to-back writes in two cycles or zero wait states.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Refresh sequence – If TCKL0 input is at logic 1, an $\overline{F}$ refresh sequence is performed after any current memory cycles are completed. The TCKL0 output is at logic 1 for 40 ns and it then goes to logic 0 and remains at logic 0 until the next refresh period. The $\overline{F}$ pin is brought to logic 0 and $\overline{RE}$ is clocked to perform an internal refresh using the internal-refresh counter. A refresh sequence is executed every 62-µs. During a refresh sequence, $\overline{RDY}$ is held at logic 1 to cause the processor to wait. The processor may need to wait for up to three wait states to complete the current memory transaction.
- Idle – If $\overline{STRB}$ is at logic 1 (except at the start of the W2 state), this indicates the cycle is an idle cycle. The $\overline{WRG}$ is brought to logic 1 to disable the EDRAM outputs from the data bus until the next active cycle.
- Chip select ($\overline{S}$) – The $\overline{S}$ is generated using a NAND, NOR or equivalent gate to decode the Q2–Q0 state machine outputs. $\overline{S}$ is brought to logic 1 during the idle and refresh states to reduce power consumption. This circuit is optional and $\overline{S}$ can be tied to ground if reduced power consumption is not desired.
- Programming the TMS320C31 DSP Timer – The timer global-control register must be programmed immediately after power up so that the refresh counter can be initialized properly. The TCLK0 output is used to generate a 40-ns minimum high pulse every 62-µs. The bit values to be programmed are as follows:

CLKSRC = 1
FUNC = 1
C/P = 0
I/O = 1
HLD = 1
INV = 0

$$\text{period register} = \frac{f(\text{Timer clock})}{fINT} = \frac{12.5\text{MHz}}{16\text{KHz}} = 780$$

Period Register Value = 780d

The remaining bit fields can remain at their default values after power up. The interrupt-enable register does not need to be programmed to generate an interrupt when timer zero generates a high pulse every 62-µs. The timer must be started immediately after power up by setting the GO/HLD bits in the timer-global-control register to 11. When the counter value equals the value set in the period register, a 40-ns logic 1 pulse is output on TCLK0 pin. This logic 1 pulse causes the REF_PENDING register output-logic level to remain at logic 1 starting with the next rising edge of H1CLK. This level is cleared only after a refresh sequence is performed. The TMS320C31 DSP timer does not need to be restarted and no interrupt is generated. A refresh sequence is not started until after the second cycle of a read-miss or write sequence, if one is in progress.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Timing of the EDRAM controller is shown in Figure 5, Figure 6, and Figure 7. The EDRAM controller operates with a 50-MHz TMS320C31 DSP system using the 12-ns version of the EDRAM as shown. The EDRAM–controller parameters are for a 26V12-7 programmable logic device (PLD) and Pericom 74FCT541CT 8-bit buffers. The TMS320C31DSP timing parameters are from the TMS320C31 DSP Data Manual (literature number SPRU031D).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3 Summary

An EDRAM controller for the 50 MHz and 60 MHz versions of the TI TMS320C31 DSP can be implemented using a simple PLD and two Pericom 74FCT541CT data buffers. The controller supports one EDRAM bank of 2M bytes (using DM2203 components) and can be expanded easily to include more memory. The EDRAM system as shown in this application report achieves zero wait states on read-hit cycles and on all write cycles.

System cost is reduced significantly by using an internal timer in the TMS320C31 to generate the refresh signals. The EDRAM achieves near 12 ns static RAM (SRAM) performance, while providing much lower memory cost at 4M-bit density. The EDRAM/TMS320C31 DSP provides an excellent ratio of cost/performance for DSP applications.

Example 1. PLD Equations

```
* TMS320C31 PLD Equations - 25 MHz Bus
* 26V12-7 PLD
* Revision 2.0
* Vantis MACH-XL 5.0 Compiler

INPUT H1CLK;
INPUT TCLK0;
LOW_TRUE INPUT RESET;
LOW_TRUE INPUT RW;
LOW_TRUE INPUT STRB;

LOW_TRUE OUTPUT RDY;
LOW_TRUE OUTPUT RE;
LOW_TRUE OUTPUT CAL;
LOW_TRUE OUTPUT WE;
LOW_TRUE OUTPUT WRG;
LOW_TRUE OUTPUT F;
LOW_TRUE OUTPUT ROWSEL;
LOW_TRUE OUTPUT COLSEL;

NODE q2..q0 CLOCKED_BY H1CLK;
D_FLOP NODE REF_PENDING CLOCKED_BY H1CLK;

MACRO ON      1;
MACRO OFF     0;

* Equations

* We need to store the current cycle state to generate EDRAM
* timings
STATE_MACHINE TMS320C31
       STATE_BITS [q2..q0]
       CLOCKED_BY H1CLK;
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
STATE IDLE:
        REF_PENDING = OFF;
        RDY = STRB * RW;
        RE = OFF;
        CAL = OFF;
        WE = OFF;
        WRG = OFF;
        F = OFF;
        ROWSEL = ON;
        COLSEL = OFF;

        IF (RESET) THEN
           GOTO REFRESH1;
           ELSE
            IF (TCLK0) THEN
               GOTO REFRESH1;
           ELSE
                IF (STRB) THEN
                   IF (RW) THEN
                      GOTO WRITE1;
                     ELSE
                      GOTO READ1;
                   END IF;
                END IF;
          END IF;
      END IF;

STATE REFRESH1 :
        REF_PENDING = OFF;
        RDY = OFF;
        RE = /H1CLK;
        CAL = OFF;
        WE = OFF;
        WRG = OFF;
        F = ON;
        ROWSEL = OFF;
        COLSEL = OFF;
        GOTO REFRESH2;

STATE REFRESH2 :
        REF_PENDING = OFF;
        RDY = OFF;
        RE = H1CLK;
        CAL = OFF;
        WE = OFF;
        WRG = OFF;
        F = OFF;
        ROWSEL = OFF;
        COLSEL = OFF;
        GOTO IDLE;
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
STATE READ1 :
        RDY = /H1CLK;
        RE = /H1CLK;
        CAL = OFF;
        WE = OFF;
        WRG = ON;
        F = OFF;
        ROWSEL = H1CLK;
        COLSEL = /H1CLK;
        REF_PENDING = TCLK0;
        GOTO READ2;

STATE READ2 :
        RDY = /H1CLK;
        RE = /H1CLK;
        CAL = OFF;
        WE = OFF;
        WRG = ON;
        F = OFF;
        ROWSEL = OFF;
        COLSEL = ON;
        REF_PENDING = TCLK0

        IF (RESET) THEN
          GOTO IDLE
        ELSE IF (TCLK0 + REF_PENDING) THEN
          GOTO REFRESH1;
        ELSE IF (STRB * /RW) THEN
          GOTO READ2;
        ELSE
          GOTO IDLE;
        END IF;

STATE WRITE1 :
        RDY = OFF;
        RE = /H1CLK;
        CAL = OFF;
        WE = /H1CLK;
        WRG = OFF;
        F = OFF;
        ROWSEL = H1CLK;
        COLSEL = /H1CLK;
        REF_PENDING = TCLK0;
        GOTO WRITE2;

STATE WRITE2 :
        RDY = STRB * RW;
        RE = H1CLK;
        CAL = H1CLK;
        WE = H1CLK;
        WRG = OFF;
        F = OFF;
        ROWSEL = OFF;
        COLSEL = ON;
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        REF_PENDING = TCLK0;

IF (RESET) THEN
        GOTO IDLE;
      ELSE IF (TCLK0 + REF_PENDING)
        THEN GOTO REFRESH1;

      ELSE IF (/STRB * RW)
        THEN GOTO WRITE1;

      ELSE IF (STRB * /RW)
        THEN GOTO READ1;

      ELSE IF (/STRB * /RW)
        THEN GOTO IDLE;

      ELSE GOTO IDLE;

END IF;

END TMS320C31;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

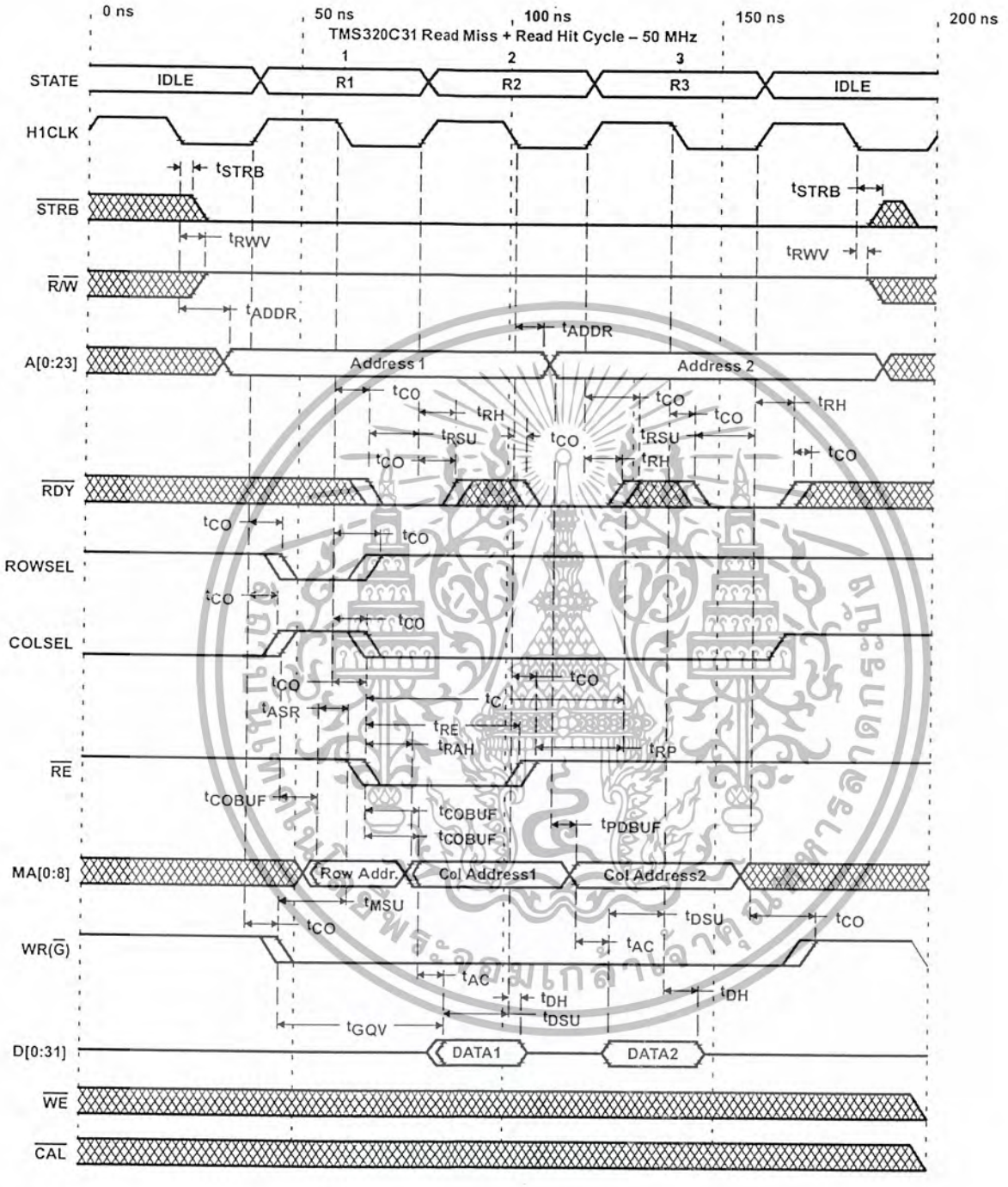


Figure 5. TMS320C31 Read-Miss and Read-Hit Cycles @ 50 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 1. TMS320C31 Read-Miss and Read-Hit Cycles @ 50 MHz

NO.		NAME	FORMULA	MIN	MAX	MARGIN	COMMENTS
1	V	t_{ASR}	5	5			Setup time, row address
3	C	t_{RSU}	6	5		<8.><8.><8.>	Setup time, $\overline{RDY}$ before H1 high
2	V	t_{RAH}	1,5	1,5			Hold time, row address
4	C	t_{RH}	0	6		<4.><4.><4.>	Hold time, $\overline{RDY}$ after H1 high
5	C	t_{ASR}	5	5		<7.>	Setup time, row address
6	D	t_{STRB}	4	4			Valid time, H1CLK to $\overline{STRB}$
7	D	t_{ADDR}	9	9			Valid time, H1CLK to address
8	D	t_{RWV}	4	4			Valid time, H1CLK to R/W
9	C	t_{RE}	30	30		<8.>	Active time, row enable
10	C	t_{RAH}	1	1		<3.>	Hold time, row enable
11	C	t_C	55	55		<7.>	Cycle time, row enable
12	D	t_{AC}	12	12	12		Access time, column address
13	D	t_{GQV}	5		5		Access time, output enable
14	C	t_{RP}	20	20		<2.>	Precharge time, row
15	C	t_{MSU}	5	5		<13.>	Setup time, $\overline{F}$ and W/R mode select
16	D	t_{COBUF}	6	6	6		Delay time, clock to output
17	C	t_{DSU}	10	10		<6.><5.>	Setup time, 'C31 data
18	C	t_{DH}	0	0		<3.>	Hold time, 'C31 data
19	D	t_{PDBUF}	4	4	4	<3.><3.>	Delay time, propagation
20	D	t_{CO}	4,6	4	6		Delay time, clock to output

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

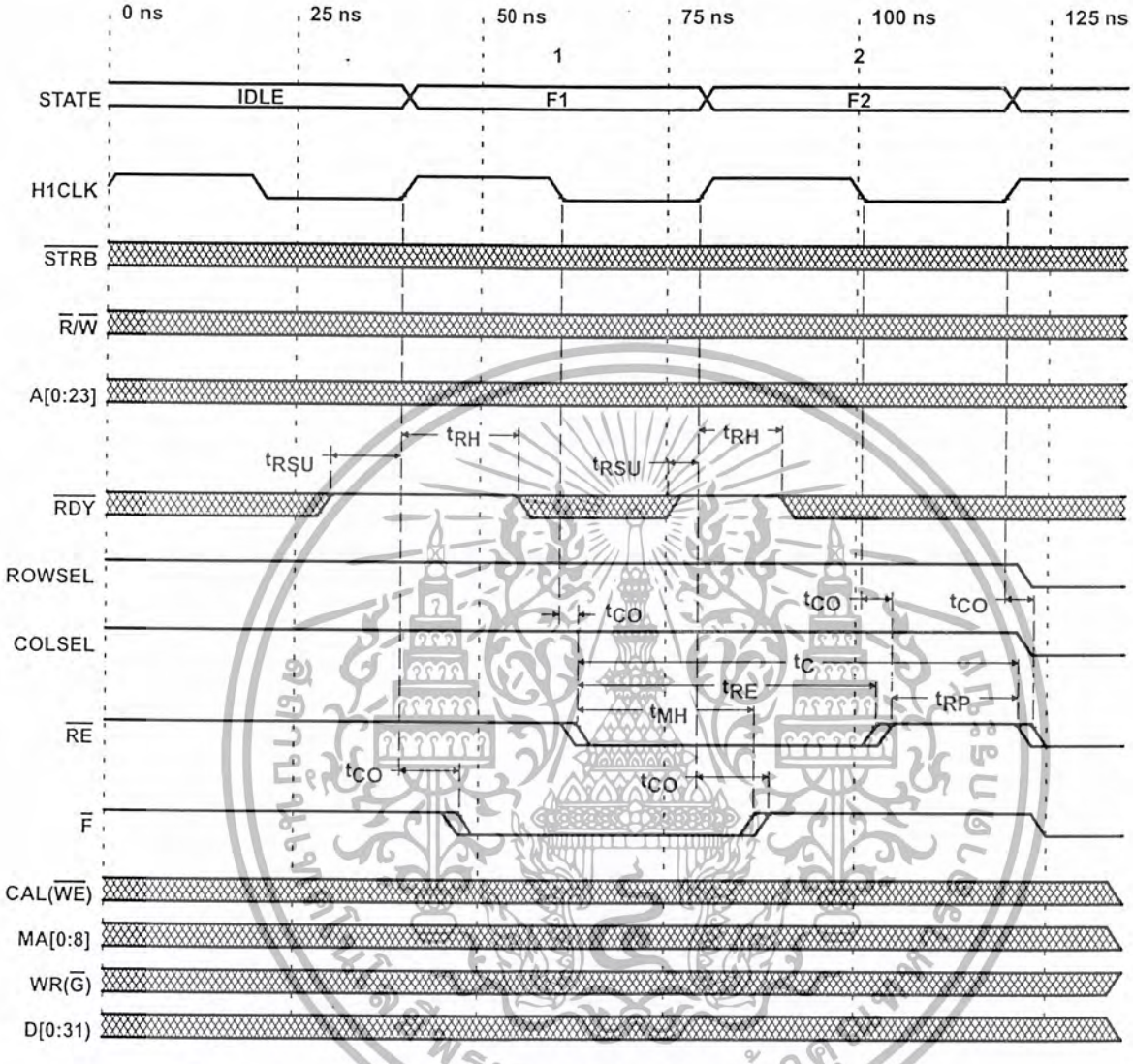


Figure 6. TMS320C31 Refresh After Idle Cycle @ 50 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 2. TMS320C31 Refresh After Idle Cycle @ 50 MHz

NO.		NAME	FORMULA	MIN	MAX	MARGIN	COMMENTS
1	V	t_{ASR}	5	5			Setup time, row address
2	V	t_{RAH}	1	1			Hold time, row address
3	C	t_{RSU}	6	6		<3.><2.>	Setup time, $\overline{RDY}$ before H1 high
4	C	t_{RH}	0	0		<14><8.>	Hold time, $\overline{RDY}$ after H1 high
5	C	t_{RE}	30	30		<8.>	Active time, row enable
6	C	t_C	55	55		<3.>	Cycle time, row enable
7	C	t_{MSU}	5	5		<13.>	Setup time, $\overline{F}$ and W/R mode select
8	C	t_{MH}	5	5		<13.>	Hold time, $\overline{F}$ and W/R mode select
9	C	t_{RP}	20	20		<–2.>	Precharge time, row
10	D	t_{CO}	4,6	4	6		Delay time, clock to output

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

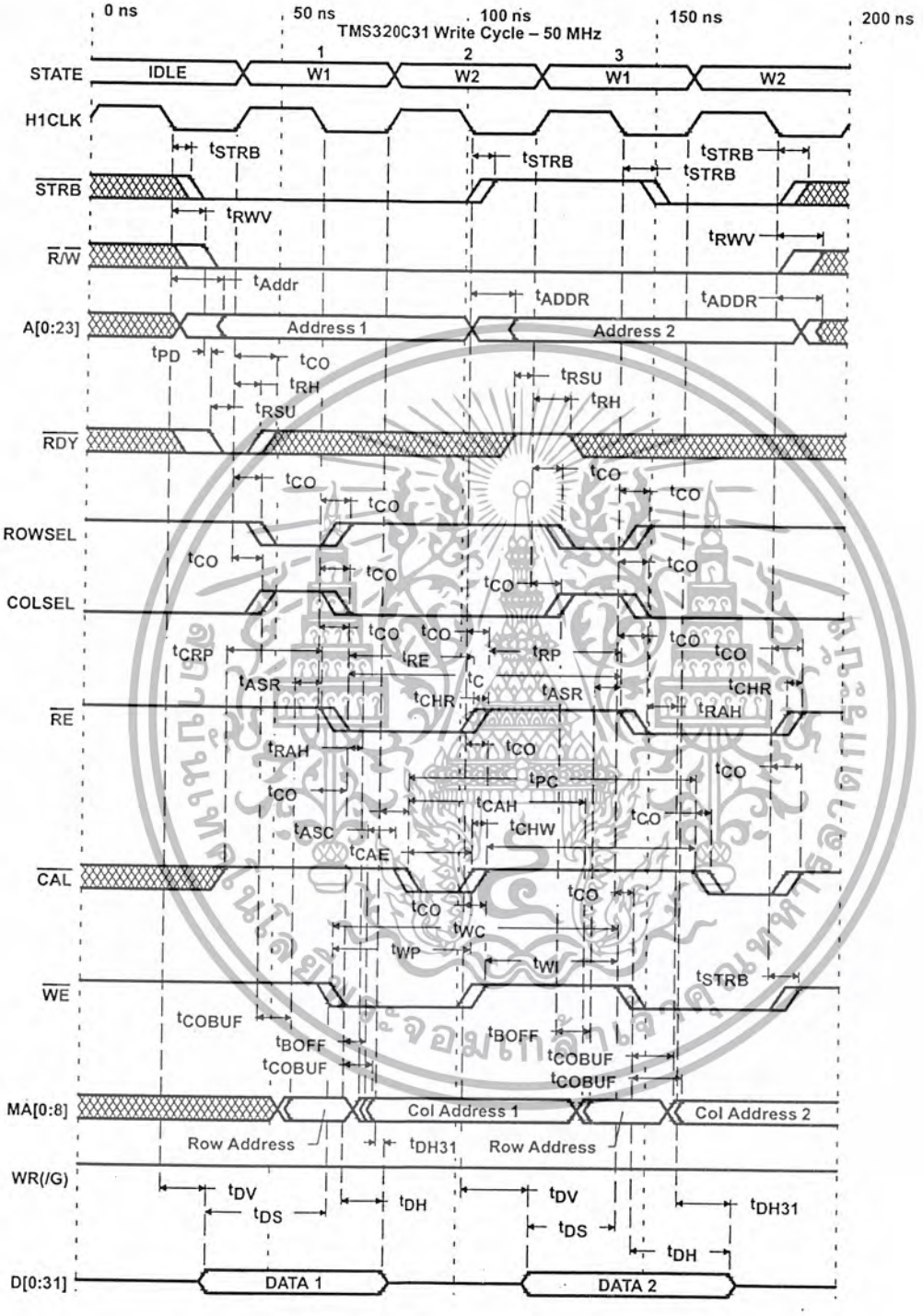


Figure 7. TMS320C31 Write Cycle – 50 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 3. TMS320C31 Write Cycle – 50 MHz

		NAME	FORMULA	MIN	MAX	MARGIN	COMMENTS
1	C	t_{RSU}	6	6		<0.><1.>	Setup time, $\overline{RDY}$ before H1 high
2	C	t_{RAH}	0	0		<4.><8.>	Hold time, $\overline{RDY}$ after H1 high
3	C	t_{ASR}	5	5		<7.><7.>	Setup time, row address
4	D	t_{STRB}	2,4	2	4		Valid time, H1CLK to $\overline{STRB}$
5	C	tR_E	30	30		<8.>	Hold time, row address
6	C	t_{RAH}	1	1		<1.><3.>	Cycle time, row enable
7	C	t_C	55	55		<23.>	Cycle time, row enable
8	C	t_{ASC}	5	5		(7.>	Setup time, column address
9	C	t_{DH31}	0	0		<12.><2.>	Hold time, 'C31 data
10	C	t_{CRP}	5	5		<22.>	Setup time, column address latch to row enable
11	D	t_{RWV}	2,7	2	7		Valid time, H1CLK to R/W
12	D	t_{ADDR}	2,9	2	9		Valid time, H1CLK to address
13	D	t_{BOFF}	4	4	4		Buffer output invalid
14	D	t_{COBUF}	6	6	6		Delay time, clock to output
15	D	t_{DV}	12	14	14		Valid time, 'C31 data
16	C	t_{BP}	20	20		<18.>	Precharge time, row
17	C	t_{CHB}	–2	–2		<0.><0.>	Setup time, $\overline{CAL}$ high to $\overline{RE}$ high
18	D	t_{CO}	4,6	4			Delay time, clock to output
19	D	t_{PD}	3,7	3	7		Delay time, propagation
20	C	t_{DS}	5	5	6	<25.><25.>	Setup time, 'C31 data
21	C	t_{DH}	0	0		<16.><26.>	Hold time, 'C31 data
22	C	t_{CAH}	0	0		<44.>	Hold time, column address
23	C	t_{CAE}	5	5		<13.>	Active time, column address latch
24	C	t_{PC}	12	12		<66.>	Cyce time, column address latch
25	C	t_{WP}	5	5		<33.>	Active time, write enable
26	C	t_{WC}	12	12		<66.>	Cycle time, write enable
27	C	t_{CH}	5	5		<53.>	Column address latch high time (latch transparent)
28	C	t_{WI}	5	5		<33.>	Inactive time, write enable
29	C	t_{CHW}	0	0		<–2.>	Column address latch high to write enable low

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้