เกรเดียนต์อัลกอริทึมแบบใหม่สำหรับอะแดปทีฟ IIR นอตช์ฟิลเตอร์โดยใช้อัลกอริทึมแบบเจเนติกร่วมกับอัลกอริทึมแบบปรับเสต็ปไซส์

A NEW CLASS OF GRADIENT-BASED ALGORITHM FOR ADAPTIVE IIR NOTCH FILTER BY USING THE COMBINATION OF GENETIC ALGORITHM AND VARIABLE STEP-SIZE

สิทธิศักดิ์ ภูวสิทธิ์กุล
SITTISAK PHUVASITKUL

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2546

ISBN 974-324-575-8

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เกรเดียนต์อัลกอริทึมแบบใหม่สำหรับอะแดปทีฟ IIR นอตช์ฟิลเตอร์โดยใช้อัลกอรทึมแบบเจเนติกร่วมกับอัลกอริทึมแบบปรับเสต็ปไซส์

A NEW CLASS OF GRADIENT-BASED ALGORITHM FOR ADAPTIVE IIR NOTCH FILTER BY USING THE COMBINATION OF GENETIC ALGORITHM AND VARIABLE STEP-SIZE

สิทธิศักดิ์ ภูวสิทธิ์กุล

SITTISAK PHUVASITKUL

เลขหมู่.....
เลขทะเบียน 47663
วัน, เดือน, ปี 21 ส.ค. 2546

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

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2546

ISBN 974-324-575-8

A NEW CLASS OF GRADIENT-BASED ALGORITHM FOR ADAPTIVE IIR NOTCH FILTER BY USING THE COMBINATION OF GENETIC ALGORITHM AND VARIABLE STEP-SIZE

SITTISAK PHUVASITKUL

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE
MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING
SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2003

ISBN 974-324-575-8

COPYRIGHT 2003

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ เกรเดียนต์อัลกอริทึมแบบใหม่สำหรับอะแดปทีฟ IIR นอตช์ฟิลเตอร์โดยใช้อัลกอริทึมแบบเจเนติก ร่วมกับอัลกอริทึม แบบปรับเสต็ปไซส์

A NEW CLASS OF GRADIENT-BASED ALGORITHM FOR ADAPTIVE IIR NOTCH FILTER BY USING THE COMBINATION OF GENETIC ALGORITHM AND VARIABLE STEP-SIZE

ชื่อนักศึกษา นายสิทธิศักดิ์ ภูวสิทธิ์กุล

รหัสประจำตัว 43061106

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชา วิศวกรรมไฟฟ้า

อาจารย์ผู้ควบคุมวิทยานิพนธ์ รศ.ชวลิต เบญจางคประเสริฐ

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
รศ.ดร.กนก	เจนจิระพงศ์เวช	
รศ.อรลาภ	แสงอรุณ	
รศ.ดร.วันชัย	ริ้วรุจา	
ผศ.นภพินทุ์	อนันตรศิริชัย	
รศ.ชวลิต	เบญจางคประเสริฐ	

วัน/เดือน/ปี ที่สอบ 23 พฤษภาคม 2546 เวลา 9.30 น. เป็นต้นไป

สถานที่สอบ ณ อาคาร 12 ชั้น ชั้น 4 (ห้อง E12-404)

บัณฑิตวิทยาลัยรับรองแล้ว

(รศ.ดร.บุญวัฒน์ อัตชู)

คณบดีบัณฑิตวิทยาลัย

วันที่ 27 **เดือน** พฤษภาคม **พ.ศ.** 2546

หัวข้อวิทยานิพนธ์	เกรเดียนต์อัลกอริทึมแบบใหม่สำหรับอะแดปทีฟ IIR นอตช์ฟิลเตอร์โดยใช้อัลกอริทึมแบบเจเนติกร่วมกับอัลกอริทึมแบบปรับเสต็ปไซส์
นักศึกษา	นายสิทธิศักดิ์ ภูวสิทธิ์กุล
รหัสนักศึกษา	43061106
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2546
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ. ชวลิต เบญจางคประเสริฐ

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้นำเสนอการศึกษาและการพัฒนาการทำงานของอะแดปทีฟ IIR นอตช์ฟิลเตอร์โดยใช้อัลกอริทึมแบบปรับเสต็ปไซส์ สำหรับการประมาณค่าความถี่ในสัญญาณที่ถูกรบกวนด้วยสัญญาณรบกวนแบบเกาส์เซียน หรือ อิมพัลซ์ โดยมีเป้าหมายให้อะแดปทีฟ IIR นอตช์ฟิลเตอร์มีประสิทธิภาพดีขึ้น กล่าวคือ ความเร็วในการลู่เข้าสู่คำตอบที่ต้องการเร็วขึ้น ค่าความผิดพลาดต่ำ หรือ ค่า MSE น้อยลง

นอกจากนี้แล้วในวิทยานิพนธ์ฉบับนี้ยังได้นำ เจเนติกอัลกอริทึม ซึ่งเป็นการจำลองการทำงานของธรรมชาติมาประยุกต์ใช้ร่วมกับอัลกอริทึมแบบปรับค่าเสต็ปไซส์ เพื่อช่วยหาค่าเสต็ปไซส์ที่เหมาะสมตอนเริ่มต้นด้วย เป็นผลให้อะแดปทีฟอัลกอริทึมที่นำเสนอนี้ สามารถเพิ่มความเร็วในการลู่เข้าหาคำตอบที่ต้องการ มีความทนทานต่อสัญญาณรบกวนสูง และยังคงให้ค่าความผิดพลาดต่ำ เพื่อเป็นการยืนยันว่าอะแดปทีฟอัลกอริทึมที่พัฒนาขึ้นนี้ สามารถนำไปใช้งานได้ จึงนำไปประยุกต์ใช้สำหรับ กำจัดสัญญาณรบกวนสัญญาณคลื่นซายน์ความถี่ 50 Hz ของไฟฟ้ากระแสสลับ 220โวลต์ ที่เหนี่ยวนำเข้าไปรบกวนสัญญาณคลื่นไฟฟ้าหัวใจของผู้ป่วยขณะที่แพทย์กำลังบันทึกสัญญาณ

Thesis Title	A New Class of Gradient-Based Algorithm for Adaptive IIR Notch filter by using the combination of Genetic Algorithm and Variable Step-Size
Student	Mr. Sittisak Phuvasitkul
Student ID.	43061106
Degree	Master of Engineering
Programme	Electrical Engineering
Year	2003
Thesis Advisor	Assoc. Prof. Chawalit Benjangkaprasert

ABSTRACT

This thesis presents new algorithms for adaptive IIR notch filter using variable step-size algorithm. This application is used for frequency estimation in background noise such as Gaussian noise or impulse noise. The objectives are enhance the performance of adaptive IIR notch filter, fast convergence speed, low mean square error (MSE). Moreover, to increase the processing speed and obtains optimal filter coefficients, the combination of genetic algorithms and variable step-size techniques are used. By this technique, the proposed algorithm gives good performance, such as fast convergence speed, noise robustness, and low MSE. From the experiment results show that the proposed adaptive IIR notch filter can used to eliminating 50 Hz power line interference, which induce to disturb the signal of electrocardiogram (ECG) of patient while the physician is recording it.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยความช่วยเหลือของบุคคลหลายฝ่ายผู้วิจัยขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้ โดยเฉพาะอย่างยิ่ง รองศาสตราจารย์ ชวลิต เบญจางคประเสริฐ ที่ได้ให้การสนับสนุนด้านข้อมูลต่างๆ ที่ใช้ในการดำเนินงานวิจัยรวมทั้ง ตรวจสอบความถูกต้องตลอดการดำเนินงานวิจัยอีกด้วย

ขอขอบคุณบิดา มารดาที่ให้การสนับสนุน และส่งเสริมด้านการศึกษา และขอขอบพระคุณผู้ให้การสนับสนุนทุนในการวิจัย รวมทั้งทุกท่านที่ได้ให้คำแนะนำ คำปรึกษา และความช่วยเหลือในทุกด้าน

สิทธิศักดิ์ ภูวสิทธิ์กุล

สารบัญ

หน้า

สารบัญ(ต่อ)

หน้า

สารบัญตาราง

สารบัญภาพ

สารบัญภาพ(ต่อ)

สารบัญภาพ(ต่อ)

สารบัญภาพ(ต่อ)

บทที่ 1
บทนำ

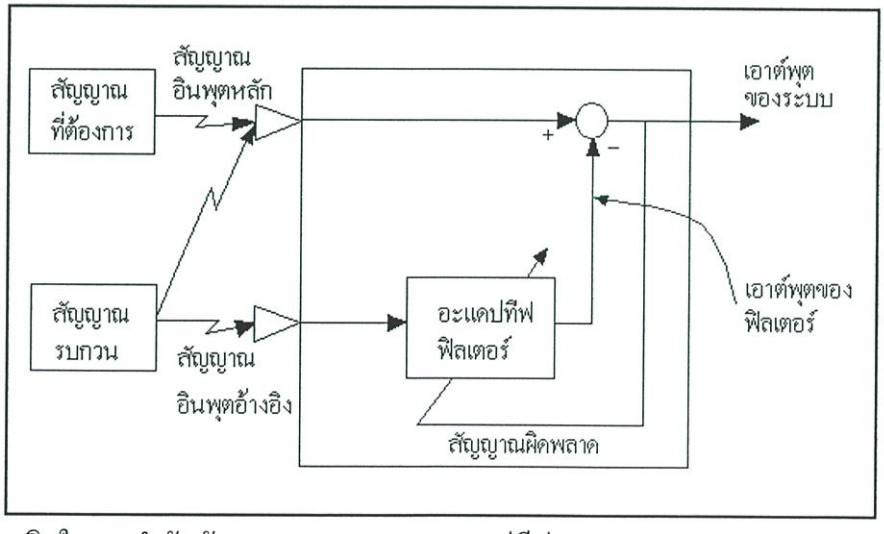
1.1 ความเป็นมา และความสำคัญ

สัญญาณรบกวนต่าง ๆ ที่เข้ามาปะปนในสัญญาณที่ต้องการจะส่งผลให้เกิดความผิดเพี้ยนของสัญญาณ ทำให้ไม่ได้รับสัญญาณที่ถูกต้อง ดังนั้นการนำตัวกรองสัญญาณแบบดิจิตอลเข้ามาช่วยกำจัดสัญญาณรบกวนก็เพื่อให้ได้สัญญาณที่ต้องการผ่านไป โดยที่ตัวกรองสัญญาณที่ดี (Optimum filter) ต้องมีคุณสมบัติ คือ มีค่าความคลาดเคลื่อนต่ำ (Mean Square Error: MSE) มีความทนทานต่อสัญญาณรบกวนสูง และมีความเร็วในการหาคำตอบสูง

ตัวกรองความถี่โดยทั่วไปจะแบ่งเป็น 2 ชนิด คือ

1. ตัวกรองความถี่แบบคงที่ (Fixed filter)
2. ตัวกรองความถี่แบบปรับค่าได้ (Adaptive filter)

ตัวกรองความถี่แบบปรับค่าได้มีข้อได้เปรียบตัวกรองความถี่แบบคงที่ คือไม่จำเป็นต้องทราบคุณสมบัติของสัญญาณ หรือต้องการเพียงเล็กน้อยเท่านั้น ทำให้สามารถนำไปประยุกต์ใช้กับงานได้หลายด้านกันอย่างกว้างขวาง เช่น ระบบสื่อสาร ระบบการประมวลผลสัญญาณดิจิตอล หรือระบบควบคุม ในปี ค.ศ. 1975 Widrow [24] ได้นำเสนอแนวคิดในการกำจัดสัญญาณรบกวนแบบอะแดปทีฟเป็นครั้งแรก แสดงดังรูปที่ 1.1



รูปที่ 1.1 แนวคิดในการกำจัดสัญญาณรบกวนแบบอะแดปทีฟ

จากรูปที่ 1.1 ตัวกรองแบบอะแดปทีฟ ต้องการสัญญาณอินพุตสองสัญญาณ คือสัญญาณอินพุตหลัก (Primary input) และสัญญาณอินพุตอ้างอิง (Reference input) ซึ่งสัญญาณอินพุตอ้างอิงจะได้จากอุปกรณ์ตรวจจับสัญญาณรบกวน (Sensors) ซึ่งอาจจะมีหลาย ๆ ตัวโดยวางอยู่ในตำแหน่งที่มี

สัญญาณรบกวน หรือตำแหน่งที่สัญญาณที่ต้องการมีขนาดต่ำ ซึ่งสัญญาณนี้จะถูกกรองโดยตัวกรองแบบอะแดปทีฟ และนำไปหักล้างกับสัญญาณอินพุตหลัก ซึ่งประกอบไปด้วยสัญญาณที่ต้องการและสัญญาณรบกวน เป็นผลให้สัญญาณรบกวนถูกลดทอนลง กระบวนการนี้แม้จะดูเหมือนว่าอันตราย เพราะถ้าหากกระทำอย่างไม่เหมาะสมแล้ว จะเป็นการเพิ่มกำลังงานของสัญญาณรบกวนที่สัญญาณเอาต์พุต แต่อย่างไรก็ตาม ถ้ากระบวนการกรอง และการลบถูกควบคุมอย่างเหมาะสมแล้ว กระบวนการทางอะแดปทีฟจะสามารถทำการลดสัญญาณรบกวนให้น้อยลงลงหรือหมดไปเลย โดยทำให้สัญญาณมีความเพี้ยนน้อยที่สุด

ตัวกรองสัญญาณแบบดิจิตอลแบ่งออกเป็น 2 ชนิดด้วยกัน คือ

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

โดยที่ฟังก์ชันถ่ายโอน (Transfer function) IIR นั้นประกอบไปด้วยโพล (Pole) และ ซีโร่ (Zero) แต่ฟังก์ชันถ่ายโอน FIR มีเพียงซีโร่เท่านั้น ในการตรวจวัดสัญญาณซายน์แบบคลื่นเดี่ยวนั้น นิยมใช้ตัวกรองสัญญาณแบบ Adaptive IIR Notch Filter (ANF) ซึ่งใช้ในงานระบบสื่อสาร ระบบการประมวลผลสัญญาณดิจิตอล หรือระบบควบคุม กันอย่างกว้างขวาง โดยจะต้องมีการปรับค่าสัมประสิทธิ์ของฟิลเตอร์ (Filter Coefficient) ให้เหมาะสม โดยอัลกอริทึมที่นำมาใช้ในการหาค่าสัมประสิทธิ์ของฟิลเตอร์ที่เหมาะสม เช่น Least Mean P-Power Error Criterion (LMP) หรือ Quantize Least Mean P-Power Error Criterion (QLMP) ซึ่งอยู่บนพื้นฐานของเกรเดียนต์ (Gradient-Based algorithm) ซึ่งทั้ง 2 อัลกอริทึม คือ LMP และ QLMP จะมีข้อเสียคือการหาค่าสัมประสิทธิ์ของฟิลเตอร์ที่ถูกต้องได้ช้า ต่อมาได้มีงานวิจัยอัลกอริทึมที่สามารถนำไปหาค่าเสต็ปไซส์ที่เหมาะสม ทำให้ความเร็วในการหาค่าสัมประสิทธิ์ของฟิลเตอร์ที่ถูกต้องเร็วขึ้นเช่น Variable Step-Size Quantize Least Mean P-Power Error Criterion (VSQLMP) โดยทั่วไปอัลกอริทึมแบบ LMP และ QLMP จะมีค่าเสต็ปไซส์ที่คงที่ ซึ่งถ้าค่าเสต็ปไซส์มากจะทำให้ความเร็วในการหาคำตอบเร็วขึ้น แต่ค่า MSE ก็จะสูงตามไปด้วย เช่นเดียวกัน หากค่าเสต็ปไซส์มีค่าน้อยจะทำให้การหาคำตอบช้าลง แต่ค่า MSE ก็จะต่ำลงไป การแก้ปัญหาโดยใช้อัลกอริทึมแบบ VSQLMP เข้ามาช่วยจะทำให้ตัวกรองสัญญาณมีความเร็วในการหาคำตอบสูงและค่า MSE ต่ำ ซึ่งอัลกอริทึมแบบ VSQLMP จะทำการปรับค่าเสต็ปไซส์ให้เหมาะสมแต่มีความทนทานต่อสัญญาณรบกวนประเภทอิมพัลส์ (Impulse) ได้ไม่ดีนัก ซึ่งอาจทำให้ตัวกรองสัญญาณไม่สามารถลู่เข้าหาคำตอบได้อีกเมื่อเกิดสัญญาณรบกวนประเภทอิมพัลส์ขึ้น ฉะนั้นในวิทยานิพนธ์ฉบับนี้จึงได้มีการนำเกรเดียนต์ของเอาต์พุตเข้ามาช่วยเพื่อให้ อัลกอริทึมแบบ VSQLMP ใหม่ (NVSQLMP) มีความทนทานต่อสัญญาณรบกวนประเภทอิมพัลส์ได้ดีขึ้น และเพื่อเพิ่มความเร็วในการหาค่าสัมประสิทธิ์ของฟิลเตอร์ที่ถูกต้องให้เร็วยิ่งขึ้น จึงได้ประยุกต์นำเอา Genetic Search Algorithm (GA) มาใช้ซึ่งเป็นการลอกเลียนแบบการทำงานของธรรมชาติ โดยอาศัยการดำเนินการทางพันธุกรรมศาสตร์เพื่อหาค่าโครโมโซมที่เหมาะ

สม (ค่าเสต็ปไซส์ที่เหมาะสม) และอาศัยการทำงานร่วมกันของอัลกอริทึมแบบ NVSQLMP และ GA ซึ่งทำให้ได้อะแดปทีฟอัลกอริทึมที่มีการหาคำตอบได้เร็ว และให้ค่า MSE ต่ำ ทำให้สามารถนำไปตรวจจับสัญญาณซายน์แบบคลื่นเดี่ยว และกำจัดสัญญาณรบกวน 50 Hz ที่เกิดจากความถี่ของไฟฟ้ากระแสสลับ ที่ปะปนมาในสัญญาณคลื่นหัวใจ (ECG) โดยได้ผลเป็นที่น่าพอใจ

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

วิทยานิพนธ์ฉบับนี้นำเสนออะแดปทีฟอัลกอริทึมแบบใหม่ ซึ่งเกิดจากอัลกอริทึมที่พัฒนามาจาก VSQLMP ใน [5] และ GA เพื่อนำมาใช้ในการหาค่าเสต็ปไซส์ที่เหมาะสม เพื่อให้ได้อัลกอริทึมที่มีคุณสมบัติที่ดีขึ้นโดยการรวมข้อดีของ NVSQLMP คือมีความทนทานต่อสัญญาณรบกวนประเภทอิมพัลส์ได้ดี และนำ GA ซึ่งมีความเร็วลู่เข้าสูงมาใช้กับตัวกรองสัญญาณ IIR แบบนอตซ์อันดับสอง (Second order) เพื่อการตรวจสัญญาณซายน์ความถี่เดี่ยวที่มีสัญญาณรบกวนปะปนอยู่ และยังนำไปกำจัดสัญญาณรบกวน 50 Hz ที่ปะปนมาในสัญญาณ ECG ด้วย

1.3 แนวคิดที่ใช้ในการวิจัย

1. VSQLMP เป็นอัลกอริทึมที่มีความเร็วลู่เข้าสูง แต่ไม่สามารถทนทานสัญญาณรบกวนแบบอิมพัลส์ จึงได้มีการนำเกรเดียนต์ของเอาต์พุตมาช่วยเพื่อทำให้ ได้อัลกอริทึมแบบ NVSQLMP สามารถทนทานต่อสัญญาณรบกวนได้ดีขึ้น

2. GA เป็นอัลกอริทึมที่สามารถปรับค่าความเร็วในการหาคำตอบได้โดยการปรับ Probability of Crossover (Pc) และ Probability of Mutation (Pm) ดังนั้นการปรับค่า Pc และ Pm ให้สูงจะได้อัลกอริทึมที่มีความเร็วในการหาคำตอบที่สูงไปด้วย แต่ผลที่ได้ Pm และ Pc ก็จะแปรผันตามค่า MSE ด้วย

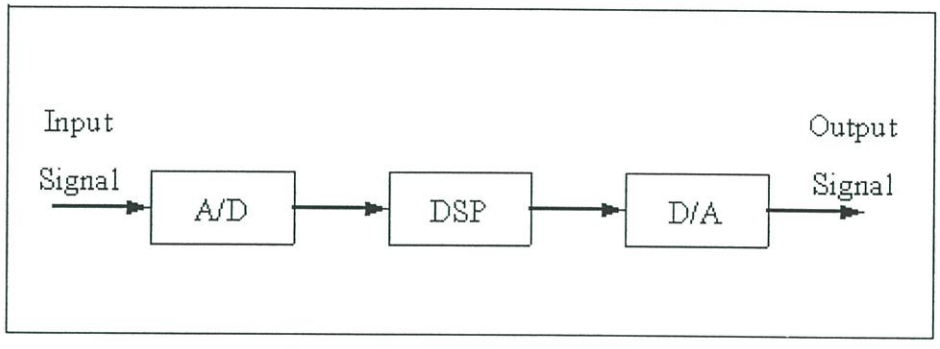
3. การนำข้อดีของ GA คือ ความเร็วในการหาคำตอบที่สูง มาทำงานร่วมกับอัลกอริทึม NVSQLMP ซึ่งมีค่า MSE ต่ำก็จะได้อะแดปทีฟอัลกอริทึมที่ทำงานได้เร็ว และ ค่าสัมประสิทธิ์ของฟิลเตอร์ที่มีความถูกต้องสูง หรือค่า MSE ต่ำนั่นเอง

1.4 ขอบเขตการวิจัย

วิทยานิพนธ์นี้มีวัตถุประสงค์ คือการนำเสนออัลกอริทึมที่ใช้ปรับค่าเสต็ปไซส์ของอะแดปทีฟ IIR นอตซ์ฟิลเตอร์อันดับสองให้มีความเร็วในการหาค่าสัมประสิทธิ์ของฟิลเตอร์ดีขึ้นจากบทวิจัยที่ผ่านมา [5] และมีความทนทานต่อสัญญาณรบกวนแบบอิมพัลส์ และเพื่อเป็นแนวทางในการพัฒนาในอนาคต โดยใช้โปรแกรมคอมพิวเตอร์สำหรับจำลองการทำงานของอะแดปทีฟอัลกอริทึมแบบต่าง ๆ ที่ได้นำเสนอในบทวิจัยนี้

บทที่ 2
กระบวนการสุ่มแบบไม่ต่อเนื่องทางเวลา

การประมวลผลสัญญาณความถี่ด้านดิจิตอล สัญญาณทางไฟฟ้าต้องถูกแปลงเป็นสัญญาณดิจิตอลก่อน โดยการ Sampling หรือ การทำ Analog to digital แล้วนำไปประมวลผลด้านดิจิตอล จากนั้นก็ทำการแปลงสัญญาณทางดิจิตอลเป็นสัญญาณทางไฟฟ้าโดย Digital to analog ดังรูปที่ 2.1



รูปที่ 2.1 การประมวลผลสัญญาณดิจิตอล

ในบทนี้จะกล่าวถึงกระบวนการสุ่มแบบไม่ต่อเนื่องทางเวลา (Discrete-Time Random Processes) ซึ่งเป็นส่วนสำคัญในการประมวลผลสัญญาณดิจิตอล โดยทั่วไปแล้วสามารถแบ่งสัญญาณออกได้เป็น 3 ประเภท

2.1 ประเภทของสัญญาณ

2.1.1 สัญญาณแบบ Deterministic เป็นสัญญาณที่สามารถอธิบายได้ด้วยสมการทางคณิตศาสตร์ ซึ่งแบ่งเป็นสัญญาณรายคาบ และสัญญาณไม่รายคาบ เช่น

$$x(t) = sin(2\pi ft + \phi) \tag{2.1}$$

หรือ

$$x(n) = \sin(2\pi fn + \phi) \tag{2.2}$$

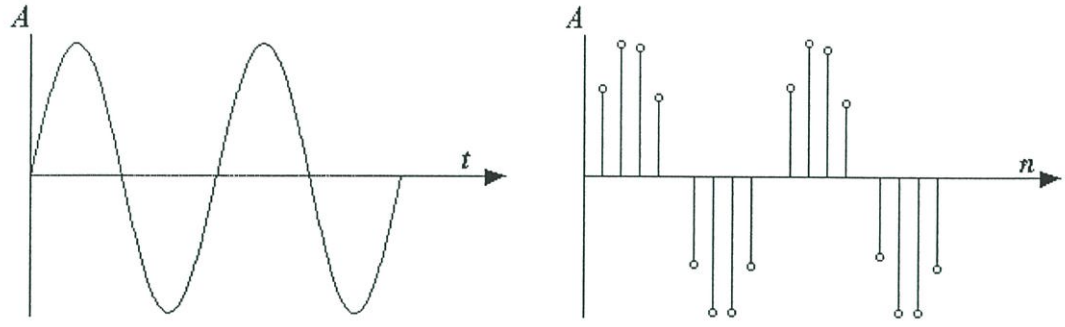
สมการที่ (2.1) คือสัญญาณซายน์ต่อเนื่องทางเวลา ส่วนสมการที่ (2.2) เป็นสัญญาณไม่ต่อเนื่องทางเวลา ดังแสดงในรูปที่ 2.2 ส่วนสัญญาณแบบ Deterministic แบบไม่มีคาบ เช่น

$$x(t) = rect(t/T) \tag{2.3}$$

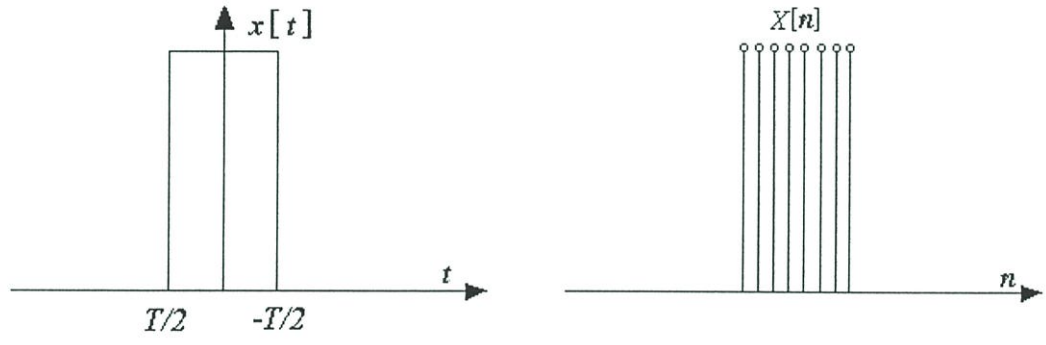
หรือ

$$x(t) = rect(n/T) \tag{2.4}$$

ดังแสดงในรูปที่ 2.3



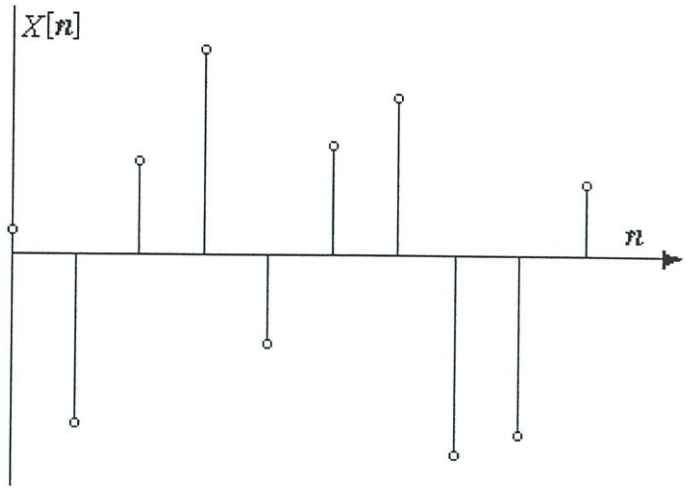
รูปที่ 2.2 สัญญาณ Deterministic แบบมีคาบ



รูปที่ 2.3 สัญญาณแบบ Deterministic แบบไม่มีคาบ

2.1.2. สัญญาณแบบสุ่ม (Random หรือ Stochastic) เป็นสัญญาณที่ไม่สามารถคาดเดาล่วงหน้าได้ว่าจะมีลักษณะเช่นใด เช่น สัญญาณรบกวนที่เกิดจากการควอนไตซ์สัญญาณของวงจร A/D ดังแสดงในรูปที่ 2.4

2.1.3. สัญญาณแบบ Chaotic สัญญาณประเภทนี้จะคล้ายคลึงกับสัญญาณแบบสุ่ม แต่จะมีรูปแบบเฉพาะตัว เช่น การหมุนของน้ำทำให้เกิดสิ่งที่เรียกว่า Swirl สัญญาณในลักษณะนี้ยังอยู่ในขั้นทำการวิจัย

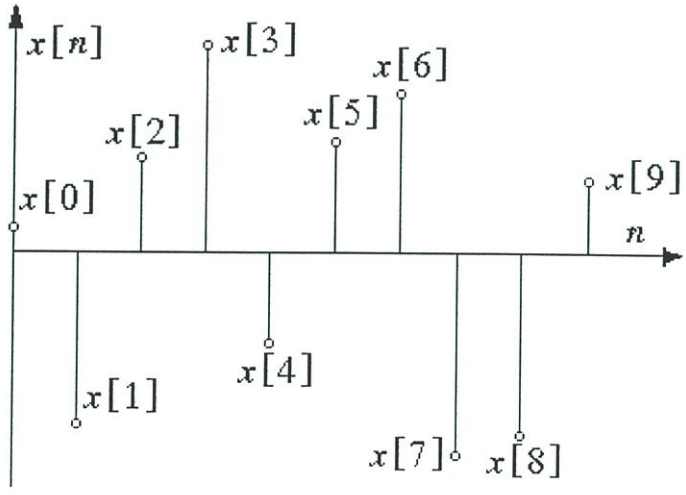


รูปที่ 2.4 สัญญาณแบบสุ่ม

2.2 ตัวแปรสุ่ม (Random Variable)

ในหัวข้อนี้จะกล่าวถึง แนวความคิดของตัวแปรสุ่ม และคุณลักษณะเฉพาะตัวต่าง ๆ ซึ่งค่อนข้างมีความสำคัญ เนื่องจากตัวแปรสุ่มมีค่าการกระจัดกระจายไม่แน่นอน และการศึกษาเรื่องตัวแปรสุ่มเป็นจุดเริ่มต้นสำหรับผู้ที่จะศึกษาในเรื่องกระบวนการสุ่มในขั้นสูง

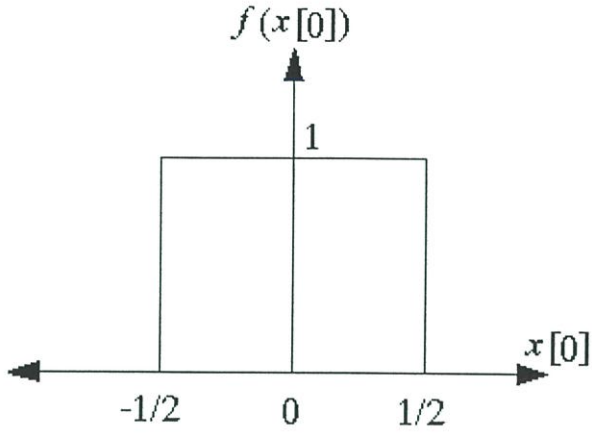
สมมติว่า $x[n]$ คือสัญญาณแบบสุ่มที่ไม่ต่อเนื่องทางเวลา ดังแสดงในรูปที่ 2.5 โดย $x[n]$ แต่ละตัวจะเรียกว่าตัวแปรสุ่ม (random variable) หรือใช้ตัวย่อว่า $r.v.$



รูปที่ 2.5 สัญญาณแบบสุ่มที่ไม่ต่อเนื่องทางเวลา

$x[n]$ หรือ $r.v.$ แต่ละตัวเรียกว่า ตัวแปรสุ่ม (random variable) สิ่งที่เป็นตัวกำหนดค่าตัวแปรสุ่มแต่ละตัวจะขึ้นอยู่กับค่าความเป็นไปได้ ซึ่งอยู่ในรูปของฟังก์ชัน คือ Probability Density Function (Pdf) เช่น $x[0]$ จะมี Pdf ส่วนตัวเป็น $f(x[0])$ หากกำหนดให้ $x[0]$ มีค่าไม่ต่ำกว่า $-1/2$

และมากสุดไม่เกิน ½ โดยที่ค่าความเป็นไปได้ของฟังก์ชันมีค่าน้อยกว่าหรือเท่ากับ 1 เสมอ สิ่งที่ได้คือ กราฟของ Pdf ของ $x[0]$ หรือ $f(x[0])$ ดังรูปที่ 2.6



รูปที่ 2.6 Pdf ของตัวแปรสุ่ม $x[0]$

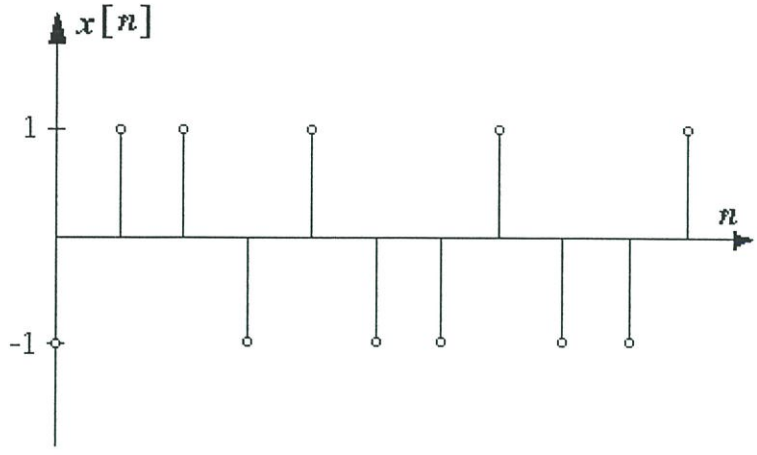
$x[0]$ จะมีค่าเท่าใดก็ได้ภายในกรอบสี่เหลี่ยม ซึ่งพื้นที่ภายในกรอบจะมีค่าเท่ากับ 1 เสมอ ซึ่งจะสามารถนำไปสู่การหาคุณสมบัติของ Pdf ได้

2.2.1 คุณสมบัติของ Pdf

จากการอินทิเกรตรูปกราฟในรูปที่ 2.6 จะได้พื้นที่รูปกราฟนั้น ซึ่งเขียนเป็นสมการได้ว่า

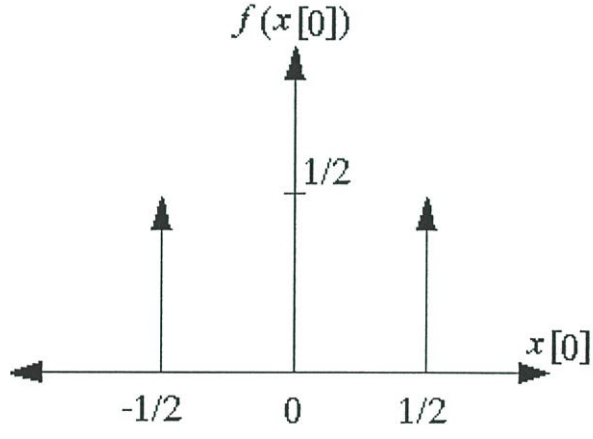
$$\int_{-\infty}^{\infty} f(x)[0]dx[0] = 1 \tag{2.5}$$

ในกรณีที่สัญญาณ $x[n]$ เป็นสัญญาณแบบไบนารี่ ซึ่งมีค่าเป็น 1 และ –1 ดังแสดงในรูปที่ 2.7



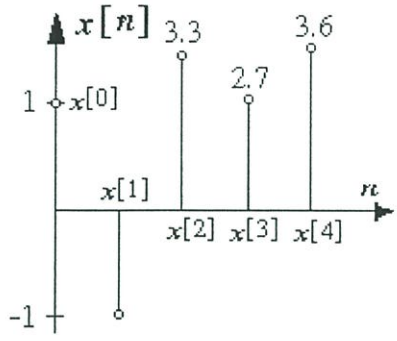
รูปที่ 2.7 สัญญาณแบบสุ่มที่มีค่า 1 และ –1

จะได้ Pdf ของ $x[0]$ เป็นดังรูปที่ 2.8



รูปที่ 2.8 Pdf ของสัญญาณแบบไบนารี

ถ้ามีสัญญาณ $x[n]$ ดังรูปที่ 2.9



รูปที่ 2.9 ตัวอย่างสัญญาณแบบสุ่ม

จากรูป $x[0]$ และ $x[1]$ เป็นสัญญาณแบบสุ่มไบนารี่ ซึ่งจะให้ Pdf ดังรูป 2.8 ส่วน $x[2],x[3],x[4]$ มี Pdf ที่เรียกว่า Normal Distribution หรือ Gaussian Distribution สัญลักษณ์ที่ใช้แทนคือ

$$N(\underset{Mean}{3.3},\underset{standardDeviation}{1})$$

โดย Gaussian distribution มีสมการเป็น

$$f(x[2]) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(x[2]-\mu)^2}{2\sigma^2}\right) \tag{2.6}$$

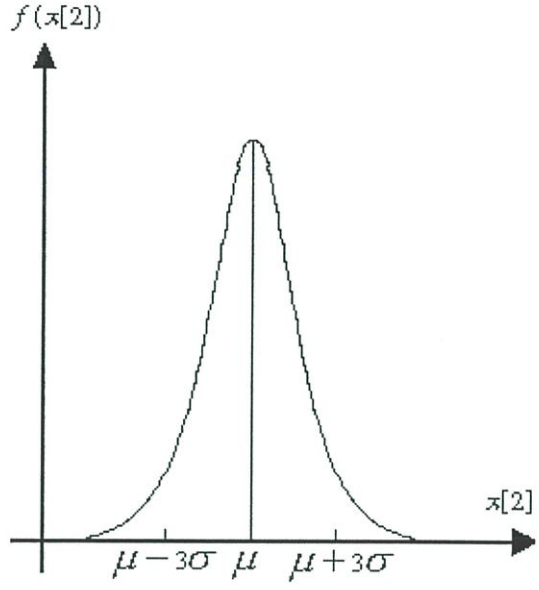
โดยที่

μ = Mean

σ = Standard Deviation

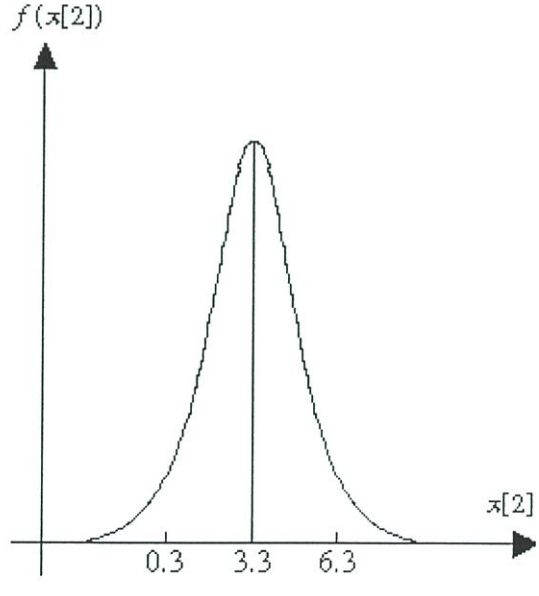
σ^2 = Variance

การกระจายของ $x[2]$ มีรูปร่างดังรูปที่ 2.10



รูปที่ 2.10 Gaussian Distribution ของ $x[2]$

จากรูปที่ 2.10 พื้นที่ของกราฟ 99% จะอยู่ในบริเวณ $\mu-3\sigma$ ถึง $\mu+3\sigma$ ดังนั้น ถ้า $N(3.3,1)$ จะได้ดังรูปที่ 2.11



รูปที่ 2.11 Gaussian Distribution ของ $x[2]$ เมื่อ $N(3.3,1)$

ในกรณีที่ต้องการหาค่า Pdf ของ $r.v.$ ตัวหนึ่งที่มีค่าน้อยมาก ๆ $(\rightarrow -\infty)$ จนถึงค่าหนึ่งของ $x[n]$ ก็จะได้ผลรวมของ Pdf เหล่านั้นเป็นอีกฟังก์ชันหนึ่ง ซึ่งจะเรียกว่า Cumulative Density Function หรือ Cdf ใช้สัญลักษณ์คือ $F_x(x[n])$ โดยที่

$$F_x(x[n]) = Probability(X[n] \leq x[n]) \tag{2.7}$$

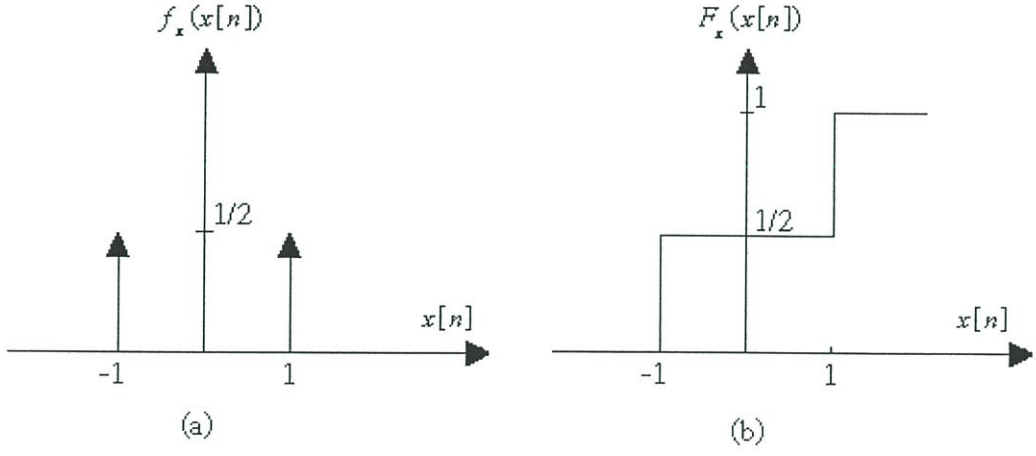
โดยที่ $X[n]$ คือ $r.v.$ และ $x[n]$ คือ ค่าหนึ่งๆ จากสมการ (2.7) จะได้ว่า

$$F_x(x[n]) = \int_{-\infty}^{x[n]} f_x(z)dz \tag{2.8}$$

หรือ

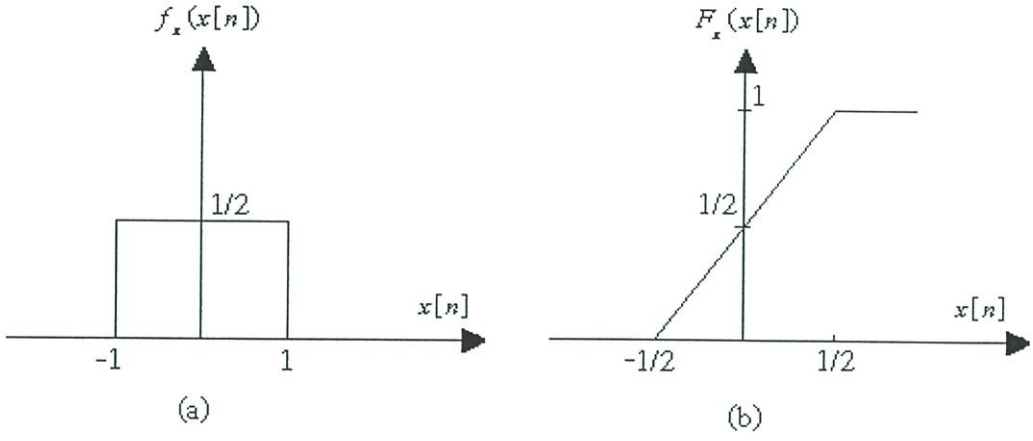
$$f_x(x[n]) = \frac{\partial F_x(x[n])}{\partial x[n]} \tag{2.9}$$

หากมี $f_x(x[n])$ ดังรูปที่ 2.12 (a) จะได้ $F_x(x[n])$ เป็นดังรูปที่ 2.12 (b) ตามลำดับ และผลสุดท้ายจะมีค่าเป็น 1 เสมอ



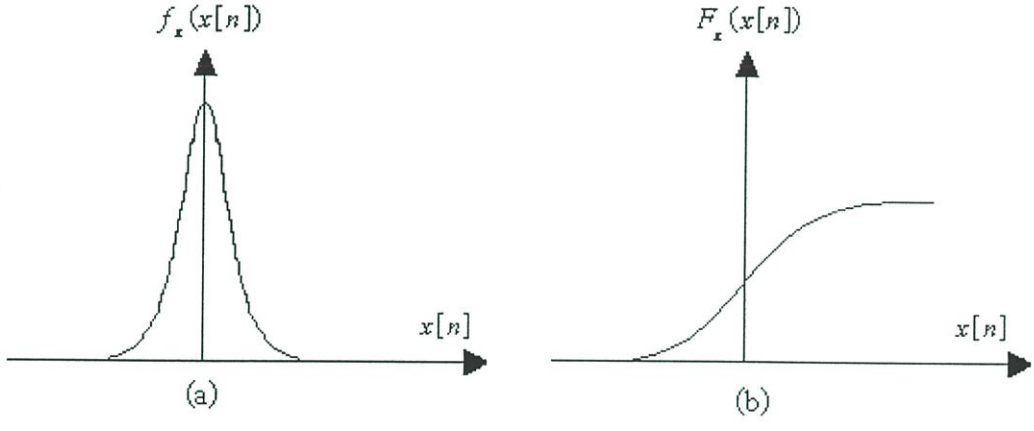
รูปที่ 2.12 (a) และ รูป (b) Cdf ของตัวแปรสุ่ม $x[n]$

และหาก $f_x(x[n])$ เป็นดังรูป 2.13(a) จะได้ $F_x(x[n])$ 2.13(b) ตามลำดับ



รูปที่ 2.13 (a) Pdf และ (b) Cdf ของ $x[n]$

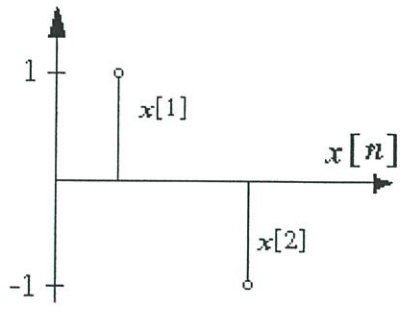
สำหรับ Gaussian Distribution จะได้ดังรูปที่ 2.14



รูปที่ 2.14 (a) Pdf และ (b) Cdf ของตัวแปรสุ่มที่เป็นแบบ Gaussian Distribution

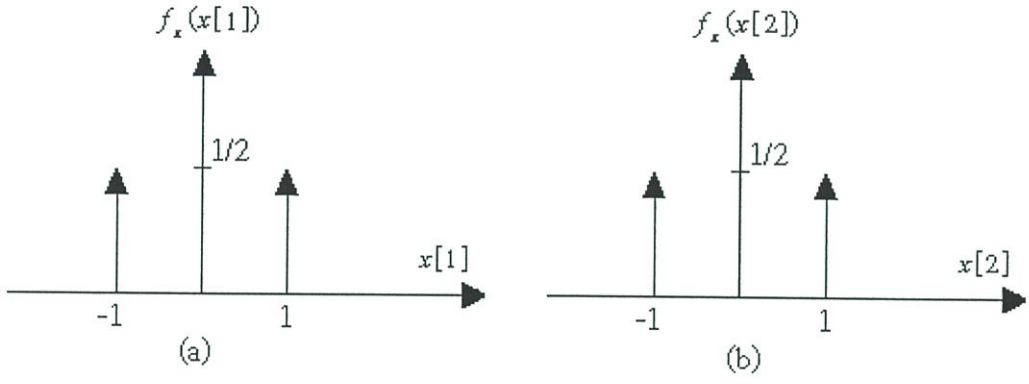
โดยทั่วไปค่า Pdf ของ $x[n_1]$ และ $x[n_2]$ อาจจะไม่เหมือนกัน ดังนั้นจะมีวิธีการหาความสัมพันธ์ของ $x[n_1]$ และ $x[n_2]$ ในรูปของ Pdf ร่วม (Joint Pdf)

ถ้าหาก $x[n_1]$ และ $x[n_2]$ เป็นสัญญาณสุ่มแบบไบนารี่แสดงดังรูปที่ 2.15



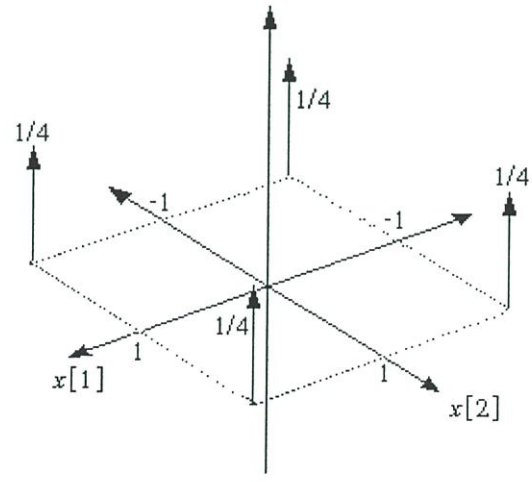
รูปที่ 2.15 สัญญาณแบบสุ่มไบนารี่

จะได้ Pdf สำหรับ $x[n_1]$ และ $x[n_2]$ ดังรูปที่ 2.16 (a) และ 2.16 (b) ตามลำดับ



รูปที่ 2.16 (a) Pdf ของ $x[n_1]$ และ (b) Pdf ของ $x[n_2]$

และจะได้ Joint Pdf เป็น $f_x(x[1],x[2])$ แสดงดังรูปที่ 2.17



รูปที่ 2.17 Joint Pdf ของ $x[1]$ และ $x[2]$

สามารถเขียนความสัมพันธ์ของ $x[1]$ และ $x[2]$ ในรูปสมการ Joint Cdf ดังนี้

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_x(x[1],x[2])dx[1]dx[2]=1 \qquad (2.10)$$

$$=F(x[n_1],x[n_2])$$

และจะได้ Joint Pdf ดังสมการ

$$f(x[n_1],x[n_2])=\frac{\partial^2 F(x[n_1],x[n_2])}{\partial x[n_1]\partial x[n_2]} \qquad (2.11)$$

ดังนั้นหากเป็นการ Distribution แบบ $r.v.$ หลายตัว ก็จะได้เป็น k-th order multivariate distribution เป็น

$$F(x[n_1],x[n_2],...x[n_k])=\Pr ob(X[n_1]\le x[n_1],...,X[n_k]\le x[n_k]) \qquad (2.12)$$

จากสมการที่ (2.12) จะได้สมการ Pdf เป็น

$$f(x[n_1],...,x[n_k])=\frac{\partial^2 F(x[n_1],...,x[n_k])}{\partial x[n_1]...\partial x[n_k]} \qquad (2.13)$$

2.2.2 ค่าเฉลี่ยแบบ Ensemble

ที่ผ่านมาได้ทำการพิจารณาสัญญาณแบบสุ่ม $x[n]$ และได้ Pdf ของ $x[n_1]$ นั้น ๆ ถ้าหากมี $x[n_1]$ มากกว่า 1 จะต้องพิจารณา $x[n_1]$ ตั้งแต่ $x_1[n]$ ถึง $x_k[n]$ ที่เวลา $n=n_0$ เดียวกันโดย ถ้าหากกำหนดให้ Probability ของ $x[n]$ ทุกตัวมีค่าเท่ากันจะได้ว่า

$$\text{Prob ของ } X_k[n]=\frac{1}{k},\forall k \qquad (2.14)$$

ค่าเฉลี่ยแบบ Ensemble (Ensemble average) เป็นการหาค่าเฉลี่ยของ $f(x[n_0])$ โดยที่ $x[n_0]$ เป็น $x_1[n_0],x_2[n_0],...,x_k[n_0]$ ซึ่งหาค่าได้ดังนี้

$$\begin{aligned}\text{Ensemble average} &= x_1[n_0]\cdot \text{Prob}(x_k[n]) \\ &+ x_2[n_0]\cdot \text{Prob}(x_2[n]) \\ &+ \dots \\ &+ x_k[n_0]\cdot \text{Prob}(x_k[n])\end{aligned} \tag{2.15}$$

2.2.3 ค่ากลาง (mean value) มีข้อกำหนดในการหาค่าตัวแปรสุ่มหลายตัว

$$\mu[n] = E\{x[n]\} \tag{2.16}$$

โดยที่ $E\{x[n]\}$ เป็นค่าเฉลี่ย Ensemble หรือ ค่าคาดหวัง (Expected value) คือ

$$E\{x[n]\} = \int_{-\infty}^{\infty} x[n] f(x[n]) dx[n] \tag{2.17}$$

โดยที่ $f(x[n])$ เป็น Pdf ค่า Mean มีชื่อเรียกอีกอย่าง First moment และจะสามารถเขียนค่า Mean ในรูปของ Relative frequency ได้เป็น

$$\mu[n] = \lim_{N\to\infty}\left\{\frac{1}{N}\sum_{i=1}^{N} x_i[n]\right\} \tag{2.18}$$

ซึ่งจะใช้แทนค่าเฉลี่ย Ensemble ได้ แต่โดยทั่วไป หาก Pdf ของ $x[n]$ ไม่เท่ากับ Pdf ของ $x[m]$ ก็จะได้ว่า

$$\mu[n] \neq \mu[m] \quad \text{ที่} \quad n \neq m \tag{2.19}$$

ตัวกระทำ Expectation ที่น่าสนใจ สามารถสรุปได้ดังนี้คือ

i) ความเป็นเชิงเส้น

$$E\{ax[n] + by[m]\} = aE\{x[n]\} + bE\{y[m]\} \tag{2.20}$$

ii) ความเป็นอิสระต่อกัน

$$E\{x[n]y[m]\} \neq E\{x[n]\} \cdot E\{y[m]\} \tag{2.21}$$

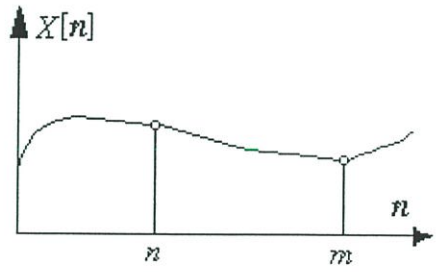
คุณสมบัติข้อนี้จะเท่ากันก็ต่อเมื่อ $x[n]$ และ $y[m]$ นั้นเป็นอิสระต่อกัน

iii) หาก $y[n]=g(x[n])$ และ Pdf ของ $x[n]$ เป็น $f(x[n])$ แล้วจะสามารถหา $E\{y[n]\}$ โดยที่ Pdf ที่ต้องการทราบมีเพียง $f(x[n])$ เท่านั้น จะได้ว่า

$$E\{y[n]\} = \int_{-\infty}^{\infty} g(x[n]) \cdot f(x[n]) dx[n] \tag{2.22}$$

2.2.4 ค่าความเกี่ยวพันธ์ (Correlation) เป็นตัวบอกความเหมือนของ $r.v.$ 2 ตัว ดังตัวอย่างในรูปที่ 2.18 ตัวแปรสุ่มมีสองตัวคือ $x[n]$ และ $x[m]$ ความเกี่ยวเนื่องกันของตัวแปรทั้งสองจะสามารถเขียนให้อยู่ในรูปของ

$$r(m,n) = E\{x[m] \cdot x[n]\} \tag{2.23}$$



รูปที่ 2.18 ตัวแปรสุ่มสองตัว

หรือ

$$r(m,n) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x[m]x[n] f(x[m],x[n]) dx[m] dx[n] \tag{2.24}$$

และในรูปของ Relative frequency จะได้ว่า

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

$$r(m,n) = \lim_{N\to\infty}\left\{\frac{1}{N}\sum_{i=1}^{N} x_i[m], x_i[n]\right\} \tag{2.25}$$

กรณีพิเศษเมื่อ $n = m$ จะได้

$$r(m,n) = E\{x^2[n]\} \tag{2.26}$$

ซึ่งเป็นค่ากำลังงาน หรือ พลังงานเฉลี่ยของสัญญาณใด ๆ สมการที่ (2.23-2.26) โดยทั่วไปจะเรียกว่า Autocorrelation

2.2.5 ค่าความแปรปรวนร่วม (Covariance)

ค่า Covariance กำหนดได้ตามสมการ

$$c(m,n) = E\{(x[m]-\mu[m])(x[n]-\mu[n])\} \tag{2.27}$$

ถ้า $\mu[n] = \mu[m] = 0$ จะได้ว่า Covariance จะมีค่าเท่ากับ Autocorrelation จากสมการที่ (2.27) จะได้ว่า

$$\begin{aligned} c(m,n) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(x[m]-\mu[m])(x[n]-\mu[n]) f(x[m],x[n])dx[m]dx[n] \\ &= E\{x[m]x[n]\} - \mu[n]\mu[m] \end{aligned} \tag{2.28}$$

จากสมการที่ (2.27) ถ้าหากว่า $m = n$ แล้วจะได้ว่า

$$\begin{aligned} c(n,n) &= E\{(x[n]-\mu[n])^2\} \\ &= \sigma_n^2 \end{aligned} \tag{2.29}$$

จากสมการที่ (2.29) เรียกว่า ค่าความแปรปรวน หรือ Variance นั่นเอง และจากสมการที่ (2.29) จะสามารถหาค่า Variance ได้ดังนี้

$$\begin{aligned} \sigma_n^2 &= \int (x[n]-\mu[n])^2 f[x]dx \\ &= E\{x^2[n]\} - (E\{x[n]\})^2 \end{aligned} \tag{2.30}$$

47663

ในกรณีของกระบวนการสุ่มแบบ Gaussian จะได้ว่า

$$\sigma_n^2 = 1 \tag{2.31}$$

2.2.6 ความเป็นอิสระ (Independence)

การที่ $\{x_1[n]\}$ และ $\{x_2[n]\}$ เป็นอิสระต่อกันก็ต่อเมื่อ

$$f(x_1[n], x_2[n]) = f(x_1[n]) \cdot f(x_2[n]) \tag{2.32}$$

และ

$$E\{x_1[n], x_2[n]\} = E\{x_1[n]\} \cdot E\{x_2[n]\} \tag{2.33}$$

$x_1[n]$ และ $x_2[n]$ จะเป็นสัญญาณ $x[n]$ เดียวกันได้ ก็ต่อเมื่อ $x_1[n] = x_2[m]$ และ $x_2[n] = x[m+n]$ จากสมการที่ (2.28) หากกล่าวว่า $x[m]$ และ $x[n]$ เป็นอิสระต่อกันแล้วจะได้ว่า

$$c(m,n) = E\{x[m]x[n]\} - \mu[n]\mu[m] = 0 \tag{2.34}$$

หรือกล่าวอีกนัยหนึ่งก็คือ ตัวแปรสุ่มทั้งสองไม่เกี่ยวพันธ์กัน (Uncorrelated) สรุปก็คือ Independent samples จะ Uncorrelated samples อาจจะไม่ Independent หากมันไม่เป็น Gaussian distribution

2.2.7 Orthogonality หาก $x[m]$ และ $x[n]$ นั้น Uncorrelated กัน และ $\mu = 0$ จะกล่าวว่ามันตั้งฉากกัน สัญลักษณ์ที่ใช้ คือ $\perp$ หรือเขียนในเทอมของ $E\{\cdot\}$ จะได้ว่า

$$E\{x[m] \cdot x[n]\} = 0 \tag{2.35}$$

2.2.8 Stationarity

ความหมายที่แท้จริงของ Stationary คือ ค่าของ $f_x(x[n])$ จะต้องมีค่าเท่ากัน สำหรับทุก ๆ ค่าของ n หรือจะมีชื่อเรียกอีกชื่อหนึ่งว่า Strictly Stationary จะสามารถเขียนเป็นสมการได้ว่า

$$f(x[n_1+n_0],x[n_2+n_0],...,x[n_k+n_0])=f(x[n_1],...x[n_k]) \tag{2.36}$$

โดยทั่วไปในทางปฏิบัติจะไม่ค่อยมี Strict stationarity ดังนั้นจึงลดข้อกำหนดลงเหลือเพียงการดูที่ Mean และ Variance ซึ่งเป็น 1^{st} และ 2^{nd} moments เท่านั้น และเรียกชื่อใหม่ว่า Wide-Sense Stationary หรือ WSS สำหรับ $x[n]$ ซึ่งเป็น WSS จะมีคุณสมบัติดังต่อไปนี้ คือ

i)

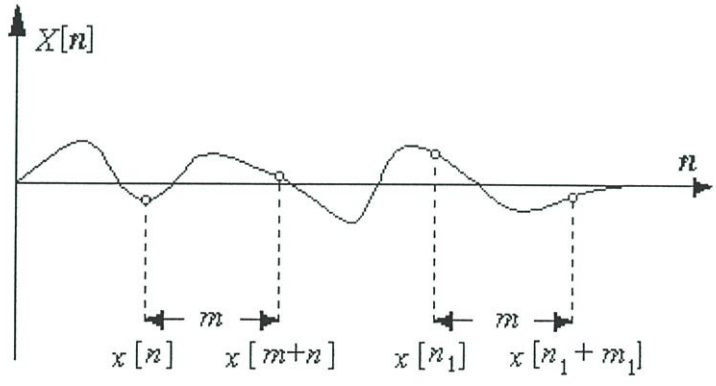
$$E\{x[m]\}=E\{x[n]\}=\mu \tag{2.37}$$

เช่น $x[n]$ เป็น Gaussian จะได้ว่า $\mu=0$

ii)

$$r(m)=E\{x[n]\cdot x[n+m]\} \tag{2.38}$$

จากสมการที่ (2.38) แสดงว่าค่า Autocorrelation จะขึ้นอยู่กับค่าของ m เท่านั้น ดังรูปที่ 2.19 จะได้ว่า Autocorrelation ของ $x[n]$ กับ $x[n+m]$ มีค่าเท่ากับ Autocorrelation ของ $x[n_1]$ กับ $x[n_1+m]$



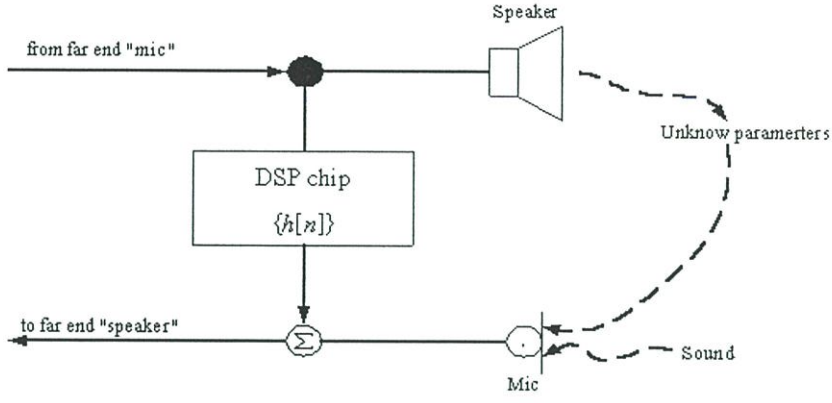
รูปที่ 2.19 อธิบายความหมายของ Autocorrelation

2.2.9 ความสัมพันธ์แบบไขว้ (Cross-correlation)

Cross-correlation จะมีนิยามตามสมการ คือ

$$r_{xy}(m)=E\{x[n]y[n+m]\}=r_{xy}[-m] \tag{2.39}$$

เพื่อให้สามารถเข้าใจได้โดยง่ายจะขอยกตัวอย่างการประยุกต์ใช้งาน Cross-correlation โดยนำมาทำ System Identification ของ Hand-free mobile phone ดังแสดงในรูปที่ 2.20 และรูปที่ 2.21



รูปที่ 2.20 Hand-free Mobile phone

สิ่งที่ต้องการคือ $\{h[n]\}$ ที่ทำให้ $y[n]$ เหมือน $d[n]$ มากที่สุด โดยตัววัดความเหมือนกันคือ $e[n]$ โดยจะใช้วิธีการลด $E\{e^2[n]\}$ ให้มีค่าน้อยที่สุด หรือหมายความว่า $\underline{h}$ จะได้จาก

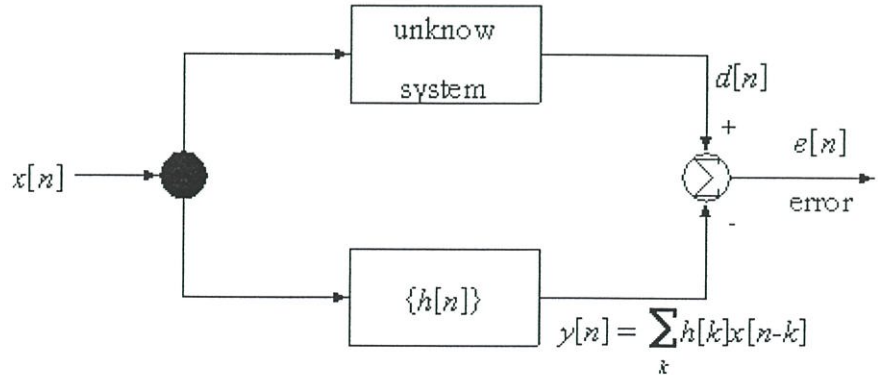
$$\underline{h} = R_{xx}^{-1} r_{dx} \tag{2.40}$$

เมื่อ

h = Impulse Response ของ Unknown System

R_{xx} = Autocorrelation ของสัญญาณอินพุต

r_{dx} = Cross-correlation ของ $d[n]$ และ $x[n]$



รูปที่ 2.21 บล๊อกไดอะแกรมของรูปที่ 2.20

สามารถศึกษารายละเอียดได้จากเอกสารอ้างอิง [7] และเนื่องจาก Correlation และ Covariance จะอธิบายความสัมพันธ์กันระหว่างสัญญาณสองสัญญาณ โดยที่ ถ้าสัญญาณทั้งสองเป็นสัญญาณเดียวกันแล้วจะได้ Autocorrelation และ Autocovariance และถ้าสัญญาณทั้งสองมาจากคนละแห่งแล้วจะได้ Cross-correlation และ Cross-covariance ซึ่งสามารถสรุปได้ดังนี้คือ

สมมติว่ามีสัญญาณแบบสุ่มที่มีคุณสมบัติเป็น Stationary สองสัญญาณคือ $x[n]$ และ $y[n]$ จะสามารถหาค่า Autocorrelation, Autocovariance, Cross-correlation และ Cross-covariance ดังสมการตามลำดับได้ดังนี้ คือ

$$r_{xx}(m) = E\{x[n]x[n+m]\} \tag{2.41}$$

$$c_{xx}(m) = E\{x[n]-\mu_x)(x[n+m]-\mu_x)\} \tag{2.42}$$

$$r_{xy}(m) = E\{x[n]x[n+m]\} \tag{2.43}$$

$$c_{xy}(m) = E\{x[n]-\mu_x)(x[n+m]-\mu_y)\} \tag{2.44}$$

เมื่อ μ_x และ μ_y คือค่า mean ของสัญญาณ $x[n]$ และ $y[n]$ ตามลำดับ ซึ่งมีคุณสมบัติที่น่าสนใจดังนี้ คือ

i)

$$c_{xx}(m) = r_{xx}(m) - \mu_x^2 \tag{2.45}$$

$$c_{xy}(m) = r_{xy}(m) - \mu_x\mu_y \tag{2.46}$$

จากสมการที่ (2.45) และ (2.46) ถ้า $\mu_x = 0$ จะทำให้ Correlation และ Covariance ค่าเท่ากัน

ii)

$$r_{xx}(0) = E\{x^2[n]\} \text{ (ค่า กำลังสองเฉลี่ย)} \tag{2.47}$$

$$c_{xx}(0) = \sigma_x^2 \tag{2.48}$$

iii)

$$r_{xx}(m) = r_{xx}(-m) \tag{2.49}$$

$$c_{xx}(m) = c_{xx}(-m) \tag{2.50}$$

$$r_{xy}(m) = r_{xy}(-m) \tag{2.51}$$

$$c_{xy}(m) = c_{xy}(-m) \tag{2.52}$$

iv) ถ้า $y[n]=x[n-n_0]$ แล้วจะได้

$$r_{yy}(m)=r_{xx}(m) \tag{2.53a}$$

$$c_{yy}(m)=c_{xx}(m) \tag{2.53b}$$

v) สำหรับกระบวนการสุ่มใดๆ เมื่อ $m\rightarrow\infty$ แล้วจะทำให้ตัวแปรสุ่มมีการ Uncorrelated กันน้อยมากดังนั้นจะได้

$$\lim_{m\rightarrow\infty} r_{xx}(m)=(E\{x[n]\})^2=\mu_x^2 \tag{2.54}$$

$$\lim_{m\rightarrow\infty} c_{xx}(m)=0 \tag{2.55}$$

$$\lim_{m\rightarrow\infty} r_{xy}(m)=\mu_x\mu_y \tag{2.56a}$$

$$\lim_{m\rightarrow\infty} c_{xy}(m)=0 \tag{2.56b}$$

2.3 การแทนสัญญาณที่มีพลังงานไม่จำกัดด้วยสเปกตรัม [8]

แม้ว่าจะไม่สามารถทำการแปลง z (z-transform) กับสัญญาณที่มีพลังงานไม่จำกัด (Infinite-Energy Signals) ได้เนื่องจากสัญญาณดังกล่าวมีลักษณะไม่เป็นรายคาบ (Aperiodic) อย่างไรก็ตาม เมื่อนำสัญญาณนี้มาทำ Autorcorrelation และ ทำ Autocovariance ค่าที่ได้จะมีลักษณะเป็นรายคาบเกิดขึ้น จึงทำให้สามารถ นำเอาการแปลง z และการแปลงฟูเรียร์ (Fourier Transform) มาใช้สำหรับหาสเปกตรัมของสัญญาณที่มีพลังงานไม่จำกัดเหล่านี้ได้ การแทนสัญญาณเหล่านี้ด้วยสเปกตรัม มีความสำคัญมากต่อการอธิบายความสัมพันธ์ระหว่างสัญญาณอินพุตและสัญญาณเอาต์พุตของระบบเชิงเส้นไม่แปรตามเวลา หรือ LTI เมื่อสัญญาณที่มีพลังงานไม่จำกัด

2.3.1 การแปลง Z ของ Correlation และ Covariance

การกำหนดให้ $R_{xx}(z),C_{xx}(z),R_{xy}(z)$ และ $C_{xy}(z)$ คือการแปลง Z ของ $r_{xx}(m),c_{xx}(m),r_{xy}(m)$ และ $c_{xy}(m)$ ตามลำดับ จากสมการที่ (2.54) และ (2.56) จะทราบได้ทันทีว่า การแปลง Z จะมีค่าก็ต่อเมื่อ $\mu=0$ เท่านั้น ซึ่งจะทำให้ $R_{xx}(z)=C_{xx}(z)$ และ $R_{xy}(z)=C_{xy}(z)$ ซึ่งมีคุณสมบัติที่น่าสนใจดังนี้

i)

$$\sigma_x^2=\frac{1}{2\pi j}\oint_{close} C_{xx}(z)z^{-1}dz \tag{2.57}$$

เมื่อ Closed เป็นเส้นทางปิดในบริเวณที่ $C_{xx}(z)$ ลู่เข้า (Region of Convergence : ROC)

ii)

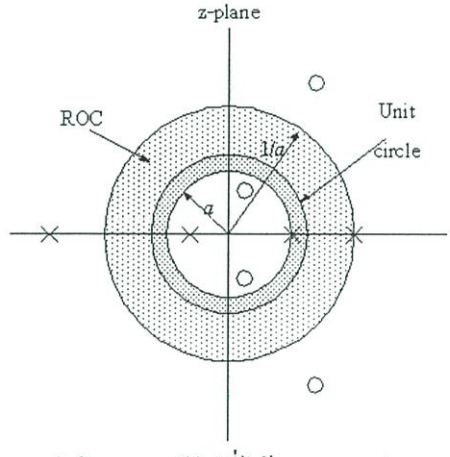
$$C_{xr}(z) = C_{xr}(1/z) \tag{2.58a}$$

$$C_{xr}(z) = C_{xr}(1/z) \tag{2.58b}$$

สมการที่ (2.58) ได้รับโดยตรงจากการแปลง Z สมการที่ (2.49) - (2.52) และ ROC ของ $C_{xx}(z)$ จะเป็นไปตามเงื่อนไขดังนี้ คือ

$$a < |z| < \frac{1}{a}$$

เมื่อ $0 < a < 1$ เนื่องจาก $C_{xx}(z) \to 0$ เมื่อ $m \to \infty$ ROC จะต้องอยู่ภายในวงกลมหนึ่งหน่วย ในกรณีที่ $C_{xx}(z)$ เป็นฟังก์ชันแบบเศษส่วน (rational function) ของตัวแปร z นัยว่า โพลและซีโร่ของ $C_{xx}(z)$ จะเกิดเป็นคู่ Complex conjugate กัน ดังแสดงในรูปที่ 2.23



รูปที่ 2.22 ROC และตำแหน่งโพลและซีโร่ที่ได้จากการแปลง Z ของ $C_{xx}(z)$

2.3.2 สเปกตัมกำลังงาน

เนื่องจากว่า ROC จะอยู่ภายในวงกลมหนึ่งหน่วย จะทำให้สามารถเขียนสมการที่ (2.57) ใหม่ได้ดังนี้

$$\sigma_x^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} P_{xx}(\omega) d\omega \tag{2.59}$$

โดยที่

$$P_{xx}(\omega)=C_{xx}(e^{j\omega}) \tag{2.60}$$

เมื่อ $\mu_x = 0$ จะเรียกว่า variance ซึ่งคือค่า mean-square หรือ กำลังงานเฉลี่ยนั่นเอง ดังนั้น พื้นที่ภายใน $P_{xx}(\omega)$ สำหรับ $-\pi \le \omega \le \pi$ จะเป็นสัดส่วนโดยตรงกับกำลังงานเฉลี่ยของสัญญาณ และเช่นเดียวกัน การอินทิเกรต $P_{xx}(\omega)$ ตลอดช่วงความถี่หนึ่ง ก็จะเป็นสัดส่วนโดยตรงกับกำลังงานเฉลี่ยของสัญญาณตลอดช่วงความถี่นั้นด้วย จากเหตุผลนี้เอง จึงเรียก $P_{xx}(\omega)$ ว่าสเปกตรัมความหนาแน่นกำลังงาน (power density spectrum) หรือเรียกย่อ ๆ ว่า สเปกตรัม (spectrum) จะพบว่าสามารถคำนวณหาค่าความหนาแน่นกำลังงานได้จากการแปลงฟูเรียร์ ของ Autocorrelation หรือ Autocovariance อย่างใดอย่างหนึ่ง อย่างไรก็ตามการใช้วิธีนี้จะทำได้ยากเมื่อ $\mu_x \neq 0$ เนื่องจาก $r_{xx}(m) \to \mu_x^2$ เมื่อ $m \to \infty$ ในกรณีเช่นนี้ จะเลือกใช้สมการที่ (2.61) มาใช้แทน (2.60) จากคุณสมบัติข้อ ii) ในหัวข้อการแปลง Z ของ Correlation และ Covariance ทำให้ทราบว่า $P_{xx}(\omega)$ เป็นฟังก์ชันแบบสมมาตร คือ $P_{xx}(\omega)=P_{xx}(-\omega)$ และก็เป็นความจริงที่ว่า ความหนาแน่นกำลังงานไม่มีค่าเป็นลบแน่นอน

ในทำนองเดียวกันจะสามารถนิยาม ความหนาแน่นกำลังงานแบบไขว้ได้ดังนี้ คือ

$$P_{xx}(\omega)=C_{xy}(e^{j\omega}) \tag{2.61}$$

ในทำนองเดียวกัน

$$P_{xx}(\omega)=P_{xx}(-\omega) \tag{2.62}$$

2.2.3 ผลตอบสนองของระบบเชิงเส้นต่อสัญญาณสุ่ม

พิจารณาระบบ LTI ซึ่งมีความเสถียรภาพ มีผลตอบสนองของระบบต่อสัญญาณอิมพัลส์เป็น $h[n]$ และให้ $x[n]$ คือ ลำดับของสัญญาณอินพุต เป็นสัญญาณแบบสุ่ม มีคุณสมบัติเป็น WSS แล้วจะได้สัญญาณเอาต์พุตของระบบ LTI ดังสมการ คือ

$$y[n]=\sum_{k=-\infty}^{\infty} h[n-k]x[k]=\sum_{k=-\infty}^{\infty} h[k]x[n-k] \tag{2.63}$$

สัญญาณอินพุตอาจกำหนดได้ด้วยค่า Mean คือ μ_x และฟังก์ชัน Autocorrelation $r_{xx}(m)$ และจะแสดงให้เห็นว่า เมื่อสัญญาณอินพุตเป็น Stationary แล้ว สัญญาณเอาต์พุต $y[n]$ ก็จะเป็น

Stationary ด้วย ซึ่งจะสามารถกำหนดได้ด้วยค่าทางสถิติเช่นเดียวกับสัญญาณอินพุต ในการประยุกต์ใช้งานหลาย ๆ อย่างเพียงพอแล้วที่จะกำหนดคุณลักษณะของสัญญาณอินพุต และสัญญาณเอาต์พุต ในเทอมของค่าเฉลี่ยอย่างง่าย ๆ เช่น ค่าMean, ค่าVariance และ Autocorrelation ซึ่งจะสามารถหาความสัมพันธ์ระหว่างปริมาณทั้งสองได้ดังนี้

กำหนดให้ค่า mean ของสัญญาณเอาต์พุตเป็น

$$\mu_y = E\{y[n]\} = \sum_{k=-\infty}^{\infty} h[k]E\{x[n-k]\} = \mu_x \sum_{k=-\infty}^{\infty} h[k] \tag{2.64}$$

จากความจริงที่ว่าค่าคาดหวังของผลบวก จะเท่ากับผลบวกของค่าคาดหวัง ในเทอมของฟังก์ชันระบบ จะเขียนได้ดังสมการ

$$\mu_y = H(e^{j0})\mu_x \tag{2.65}$$

จากสมการที่ (2.65) เป็นจริงทั้งนี้เพราะ สัญญาณอินพุตเป็น stationary จึงทำให้สัญญาณเอาต์พุตคงที่ด้วย

ต่อไปจะสมมติว่า สัญญาณเอาต์พุตมีลักษณะเป็น Nonstationary ชั่วคราวจะสามารถกำหนดค่าฟังก์ชัน Autocorrelation ของสัญญาณเอาต์พุตได้ดังสมการ คือ

$$\begin{aligned} r_{xx}(n,n+m) &= E\{y[n]y[n+m])\} \\ &= E\left\{\sum_{k=-\infty}^{\infty}\sum_{j=-\infty}^{\infty} h[k]h[j]x[n-k]x[n+m-j]\right\} \\ &= \sum_{k=-\infty}^{\infty} h[k]\sum_{j=-\infty}^{\infty} h[j]E\{x[n-k]x[n+m-j]\} \end{aligned} \tag{2.66}$$

เนื่องจากสมมติว่า $x[n]$ เป็น Stationary แล้ว $E\{x[n-k]x[n+m-j]\}$ จะขึ้นกับความแตกต่างของ $m+k-j$ ดังนั้น

$$r_{yy}(n,n+m) = \sum_{k=-\infty}^{\infty} h[k]\sum_{j=-\infty}^{\infty} h[j]r_{xx}[m+k-j] = r_{yy}[m] \tag{2.67}$$

นั่นคือ ฟังก์ชัน Autocorrelation ของสัญญาณเอาต์พุตก็จะขึ้นกับค่าความต่างของเวลา m ด้วย ดังนั้นสรุปได้ว่า สำหรับระบบที่มีคุณสมบัติเป็น LTI ถ้าสัญญาณอินพุตเป็น stationary แล้ว จะได้สัญญาณเอาต์พุตเป็น stationary ด้วย

ถ้าแทน $l = j - k$ ลงในสมการที่ (2.67) แล้วจะสามารถเขียนใหม่ได้ดังสมการ คือ

$$\begin{aligned} r_{yy}(m) &= \sum_{l=-\infty}^{\infty} r_{xx}[m-l] \sum_{k=-\infty}^{\infty} h[k]h[l+k] \\ &= \sum_{k=-\infty}^{\infty} r_{xx}[m-l]v[l] \end{aligned} \tag{2.68}$$

เมื่อกำหนดให้

$$v[l] = \sum_{k=-\infty}^{\infty} h[k]h[l+k] \tag{2.69}$$

สมการที่ (2.69) เรียกว่า Aperiodic autocorrelation sequence หรือเรียกง่าย ๆ ว่า autocorrelation ที่จริงแล้ว สมการที่ (2.69) ก็คือ การคอนโวลูชันแบบไม่ต่อเนื่องทางเวลาของ $h[n]$ กับ $h[-n]$ นั่นเอง

สมมติว่า $\mu_x = 0$ ทำการแปลง Z สมการที่ (2.68) และ สมการที่ (2.69) จะได้ว่า

$$\begin{aligned} R_{yy}(z) &= V(z)R_{xx}(z) \\ &= H(z)H(z^{-1})R_{xx}(z) \end{aligned} \tag{2.70}$$

ในเทอมของความหนาแน่นกำลังงานจะได้ว่า

$$P_{yy} = H(e^{j\omega})^2 P_{xx}(\omega) \tag{2.71}$$

สมมติว่า $\mu_x = 0$ และจากสมการที่ (2.69), $\mu_y = 0$ เหมือนกัน ดังนั้นจะได้ว่า

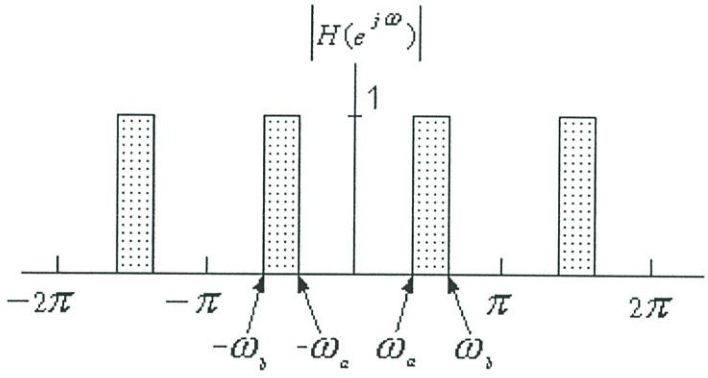
$$\begin{aligned} r_{yy}(0) &= \frac{1}{2\pi} \int_{-\pi}^{\pi} P_{yy}(\omega)d\omega \\ &= \text{กำลังงานเฉลี่ยรวมที่เอาต์พุต} \end{aligned} \tag{2.72}$$

แทนสมการที่ (2.71) ลงในสมการที่ (2.72) จะได้

$$r_{yy}(0)=\frac{1}{2\pi}\int_{-\pi}^{\pi}\left|H(e^{j\omega})\right|^2 P_{xx}(\omega)d\omega \tag{2.73}$$

สมมติว่า $H(e^{j\omega})$ เป็นตัวกรองแบบแถบผ่านในอุดมคติ ดังแสดงในรูปที่ 2.23 จากที่ทราบว่า $r_{xx}(m)$ เป็นฟังก์ชันคู่ดังนั้น

$$P_{xx}(\omega)=P_{xx}(-\omega) \tag{2.74}$$



รูปที่ 2.23 ผลตอบสนองทางความถี่ของตัวกรองแบบแถบผ่านในอุดมคติ

และในทำนองเดียวกัน $\left|H(e^{j\omega})\right|^2$ ก็เป็นฟังก์ชันคู่ในตัวแปร ω ดังนั้นจะได้ว่า

$$\begin{aligned} r_{xx}(0) &= \text{กำลังเฉลี่ยเอาต์พุต} \\ &= \frac{1}{\pi}\int_{\omega_a}^{\omega_b} P_{xx}(\omega)d\omega \end{aligned} \tag{2.75}$$

ดังนั้นพื้นที่ภายใต้ $P_{xx}(\omega)$ ระหว่าง ω_a ถึง ω_b จะสามารถใช้แทน กำลังงานเฉลี่ยของสัญญาณอินพุต ในช่วงความถี่หนึ่งนั่นเอง

2.4 สรุป

ในบทนี้ได้กล่าวถึงทฤษฎีพื้นฐานของกระบวนการสุ่มแบบไม่ต่อเนื่องทางเวลาโดยเริ่มจากการกล่าวทั่ว ๆ ไปเกี่ยวกับตัวแปรสุ่ม และกระบวนการสุ่มหลายตัว ถึงแม้ว่าการที่จะอธิบายถึงคุณสมบัติของกระบวนการสุ่มได้อย่างสมบูรณ์นั้นจำเป็นจะต้องทราบ Joint distribution หรือ Density function อย่างไรก็ตาม จะสามารถอธิบายกระบวนการสุ่มได้ด้วย ค่า Mean, ค่า Variance และ

Autocorrelation โดยการใช้ค่าเฉลี่ยแบบ Ensemble แทน Density function ได้ จากนั้นได้กล่าวถึงกระบวนการสุ่มที่มีคุณสมบัติเป็น Stationary และแบบ Wide-Sense Stationary (WSS) ซึ่งพบว่า ค่า Mean ของกระบวนการสุ่มนี้จะมีค่าคงที่ไม่ขึ้นกับเวลา นอกจากนี้ค่า Autocorrelation $E\{x[n]x[n+m]\}$ จะขึ้นกับค่าความแตกต่างทาง m เวลาสั้น ๆ ก็อาจจะสมมติได้ว่า กระบวนการสุ่มนั้นมีคุณสมบัติเป็น WSS ได้ ในบางปัญหามีความจำเป็นจะต้องทราบค่า 1^{st} (Mean, Variance) และ 2^{nd} (Correlation, Covariance) order ของกระบวนการสุ่ม ซึ่งเป็นไปไม่ได้ที่จะทราบล่วงหน้าได้ ดังนั้นจำเป็นจะต้องใช้วิธีการประมาณค่าทั้งสอง ซึ่งมีด้วยกันสองแบบคือ โดยใช้การเฉลี่ยแบบ Ensemble และ ค่าเฉลี่ยทางเวลา ในกรณีที่ค่าเฉลี่ยทั้งสองเท่ากัน จะเรียกกระบวนการสุ่มนั้นว่าเป็น Ergodic

ปริมาณทางสถิติที่สำคัญอีกอันหนึ่งที่ได้กล่าวถึงในบทนี้คือ สเปกตรัมกำลังงาน ซึ่งเป็นการทำการแปลงฟูเรียร์ของกระบวนการสุ่มแบบ WSS และยังได้แสดงให้เห็นถึงความสัมพันธ์ระหว่างสัญญาณอินพุต และสัญญาณเอาต์พุตจากระบบ LTI ให้เห็นอีกด้วย ซึ่งทฤษฎีทั้งหมดที่ได้กล่าวถึงในบทนี้ จะเป็นทฤษฎีที่ต้องใช้สำหรับการศึกษา การประมวลผลสัญญาณแบบอะแดปทีฟ

บทที่ 3
ตัวกรองแบบอะแดปทีฟ

ตัวกรองความถี่แบบอะแดปทีฟ (Adaptive filtering) ได้เข้ามามีบทบาทเป็นอย่างมากในงานประมวลผลสัญญาณดิจิตอล และระบบการควบคุม ทั้งนี้เพราะสามารถปรับเปลี่ยนคุณลักษณะ ของตัวกรองให้สอดคล้องกับสัญญาณที่เข้ามาได้ด้วยตัวเอง ตัวกรองความถี่แบบอะแดปทีฟ แบ่งได้เป็นสองชนิด [9] คือ ตัวกรองความถี่แบบ Finite Impulse Response (FIR filter) และตัวกรองความถี่แบบ Infinite Impulse Response (IIR filter) ตัวกรองความถี่แบบ FIR นั้น ฟังก์ชันถ่ายโอน (Transfer function) จะมีเฉพาะซีโร่ (Zero) ไม่มีโพล (Pole) ทำให้ระบบมีความเสถียรภาพอย่างแน่นอน และมีผลตอบสนองทางเฟสเป็นเชิงเส้น (Linear phase) แต่มีข้อเสียคือ ต้องใช้จำนวนอันดับ (Order) สูงจึงจะทำให้ตัวกรองทำงานได้ดี ส่วนตัวกรองความถี่แบบ IIR นั้น ฟังก์ชันถ่ายโอนจะประกอบด้วย ทั้งซีโร่และโพล ทำให้มีปัญหาทางด้านเสถียรภาพ แต่ถ้ามีการออกแบบที่ดี จะทำให้ปัญหาดังกล่าวน้อยลง หรือไม่เกิดขึ้นเลย ข้อดีของตัวกรองความถี่แบบ IIR เมื่อเทียบกับแบบ FIR คือ ที่คุณภาพของการทำงาน (Performance) เท่ากัน ตัวกรองความถี่แบบ IIR จะใช้จำนวนอันดับน้อยกว่า ซึ่งทำให้การคำนวณน้อยกว่าด้วย ซึ่งความซับซ้อนในการคำนวณนี้ มีความสำคัญมากเมื่อนำตัวกรองความถี่ไปใช้งานที่เวลาจริง (Real time) ด้วยไมโครโปรเซสเซอร์ ซึ่งมีข้อจำกัดทางด้านจำนวนบิต และความเร็วในการทำงานตัวกรองความถี่ที่มีจำนวนการคำนวณน้อยกว่า จะทำให้ไมโครโปรเซสเซอร์ทำงานน้อยลง และทำงานได้เร็วกว่า ในวิทยานิพนธ์ฉบับนี้ได้นำเอาตัวกรองความถี่แบบ IIR มาใช้ ดังนั้นเนื้อหาที่จะได้กล่าวต่อไป จะได้กล่าวเฉพาะตัวกรองความถี่แบบ IIR เท่านั้น ส่วนตัวกรองความถี่แบบ FIR สามารถดูได้จากเอกสารอ้างอิง [10-13] การเลือกใช้งานระหว่าง FIR และแบบ IIR สามารถสรุปได้ [14] ดังต่อไปนี้คือ

i) การประยุกต์ใช้งานทางด้านการสื่อสารข้อมูล (data transmission) ทางการแพทย์ และการประมวลผลสัญญาณภาพ จำเป็นต้องใช้ตัวกรองความถี่ที่ให้ผลตอบสนองทางเฟสเป็นเชิงเส้น ดังนั้น FIR จะเหมาะสมกว่าแบบ IIR

ii) สำหรับการใช้งานที่เวลาจริงซึ่งต้องมีการจำกัดจำนวนบิตนั้น การคำนวณสัมประสิทธิ์ของตัวกรองจะต้องทำการปัดเศษ (round-off) ซึ่งจะทำให้เกิด round-off noise ขึ้นซึ่ง round-off noise นี้จะเกิดขึ้นใน FIR น้อยกว่าที่เกิดขึ้นใน IIR

iii) ในกรณีที่มีความต้องการความชันในช่วงแถบหยุด (cutoff) สูงนั้น FIR จะต้องใช้จำนวนสัมประสิทธิ์มากกว่า IIR

iv) IIR สามารถออกแบบได้โดยตรงจากตัวกรองต้นแบบในตัวกรองความถี่แบบแอนะลอกแต่ FIR ทำไม่ได้

v) ตัวกรองความถี่แบบ FIR จะสังเคราะห์ได้ค่อนข้างยากถ้าหากไม่ใช้คอมพิวเตอร์ (CAD) ช่วยในการออกแบบ

จากทั้งหมดที่กล่าวมาพอจะสรุปเป็นแนวทางสำหรับการใช้งานได้ดังนี้

- จะใช้ IIR ก็ต่อเมื่อในงานที่มีความต้องการความชันในช่วงแถบหยุดสูง ซึ่ง IIR จะใช้จำนวนสัมประสิทธิ์น้อยกว่า FIR
- จะใช้ FIR ถ้าจำนวนสัมประสิทธิ์ไม่มากจนเกินไป โดยเฉพาะสำหรับงานที่ต้องการความผิดเพี้ยนทางเฟสน้อยจะเหมาะสมอย่างยิ่ง

3.1 ตัวกรองความถี่แบบ IIR (IIR filters)

ตัวกรองความถี่แบบ IIR เป็นส่วนประกอบที่สำคัญอันหนึ่งในระบบการประมวลผลแบบไม่ต่อเนื่องทางเวลา (Discrete-time processing) มีข้อดีหลาย ๆ ข้อที่เหนือกว่าตัวกรองกรองแบบ FIR โดยเฉพาะอย่างยิ่ง ด้านผลตอบสนองทางขนาด เช่น เมื่อมีความต้องการให้ความกว้างของช่วงความถี่แถบผ่าน (pass band) หรือ ช่วงแถบความถี่ไม่ผ่าน (stop band) มีขนาดแคบมาก ๆ หรือต้องการใช้ช่วงแถบเปลี่ยน (transition band) มีค่าแคบมาก ๆ หรือมีความต้องการให้มีอัตราการลดทอนที่สูง ตัวกรองความถี่แบบ IIR เหมาะที่จะนำมาใช้งานมากกว่าแบบ FIR

ตัวกรองความถี่แบบ IIR จะมีสมการของสัญญาณเอาต์พุตที่เป็นฟังก์ชันของสัญญาณอินพุตปัจจุบัน อินพุตในอดีตและสัญญาณเอาต์พุตในอดีต (สัญญาณเอาต์พุตในอดีตได้จากการป้อนกลับ) ซึ่งสามารถเขียนให้อยู่ในรูปสมการผลต่าง (Difference equation) ได้ดังนี้

$$y(n) = -a_1 y(n-1) - a_2 y(n-2) - \ldots - a_N y(n-N) + b_0 x(n) + b_1 x(n-1) + \ldots + b_M x(n-M) \qquad (3.1)$$

เมื่อ $x(n)$ คือลำดับของสัญญาณอินพุต และ $y(n)$ คือลำดับของสัญญาณเอาต์พุต N คือจำนวนตัวอย่าง (Samples) ทั้งหมดของสัญญาณเอาต์พุตที่ทราบค่า สมการที่ (3.1) สามารถนำไปใช้ในการคำนวณหาสัญญาณเอาต์พุตต่อ ๆ กันไปได้ รูปแบบของการนำเอาสัญญาณเอาต์พุตก่อนหน้ามาใช้สำหรับคำนวณหาสัญญาณเอาต์พุตตัวต่อไปจะเรียกว่า รีเคอร์ซีฟ (Recursive) โดยทั่วไป ตัวกรองความถี่แบบ IIR และตัวกรองความถี่แบบรีเคอร์ซีฟ มักจะนำมาใช้ในความหมายอย่างเดียวกันทั้งนี้เพราะจากสมการที่ (3.1) สามารถนำไปใช้สร้างตัวกรองความถี่ทั้งสองแบบได้เหมือนกัน คำว่า IIR จะหมายถึงรูปแบบของผลตอบสนองอิมพัลส์ (Impulse response) ของตัวกรองความถี่ ในขณะที่คำว่า รีเคอร์ซีฟ จะหมายถึงตัวกรองความถี่นี้ถูกสร้างขึ้นอย่างไร ตัวกรองความถี่แบบ FIR จะสามารถ

สร้างในรูปแบบของรีเคอร์ซีฟ ได้ด้วยในขณะเดียวกันตัวกรองความถี่แบบ IIR ก็จะสามารสร้างในรูปแบบนอนรีเคอร์ซีฟ (Nonrecursive) ได้เหมือนกัน

เพื่อความสะดวกจะนิยามตัวกรองความถี่แบบรีเคอร์ซีฟด้วยกับฟังก์ชันถ่ายโอน หรือฟังก์ชันของระบบ (System function) ฟังก์ชันของระบบก็คือการแปลง z (z-transform) ของผลตอบสนองอิมพัลส์ของตัวกรองความถี่ โดยฟังก์ชันของระบบนี้จะอยู่ในรูปเศษส่วน (Rational function) ในตัวแปร z^{-1} ระบบตามสมการที่ (3.1) จะมีฟังก์ชันของระบบเป็นไปตามสมการ คือ

$$H(z)=\frac{\sum_{k=0}^{M} b_k z^{-k}}{i+\sum_{k=1}^{N} a_k z^{-k}} \tag{3.2}$$

จากสมการที่ (3.2) ถ้ากำหนดให้เงื่อนไขเริ่มต้นเป็นศูนย์ จะสามารถเขียนใหม่ได้คือ

$$H(z)=G\frac{\prod_{k=0}^{M}(1-\beta_k z^{-1})}{\prod_{k=1}^{N}(1-\alpha_k z^{-1})} \tag{3.3}$$

ราก (Roots) ของโพลิโนเมียลตัวเศษ β_k เรียกว่าซีโร่ของตัวกรองความถี่ และรากของตัวส่วน α_k เรียกว่าโพล และ G เป็นอัตราการขยาย ซึ่งมีค่าคงที่ โดยทั่วไป จำนวนของซีโร่ และโพลจะขึ้นกับความต้องการของผู้ใช้งาน จำนวนอันดับที่ N (Order N) ของตัวกรองความถี่แบบ IIR จะหาได้จากจำนวนรากของโพลที่อยู่ในระนาบ z ที่มีค่าจำกัด (Finite z-plane)

ตัวกรองความถี่ที่มีความเป็น เชิงเส้นไม่แปรตามเวลา (Linear Time-Invariance:LTI) จะมีคุณสมบัติเป็นคอซอล (Causal) ถ้าผลตอบสนองอิมพัลส์มีค่าเท่ากับศูนย์เมื่อ $x(n)=0$ ตัวอย่างสัญญาณเอาต์พุตของตัวกรองความถี่ ที่เป็นคอซอลจะขึ้นกับตัวอย่างสัญญาณอินพุตปัจจุบัน และในอดีตเท่านั้น ถ้าจำกัดสัญญาณอินพุตให้มีค่าเป็นศูนย์ $x(n)=0$ สำหรับ $n<0$ และค่าเริ่มต้นของ $y(-1)=y(-2)=...=y(-N)=0$ จะทำให้ตัวกรองความถี่แบบรีเคอร์ซีฟตามสมการที่ (3.1) มีคุณสมบัติเป็นคอซอล ความเป็นคอซอลของตัวกรองความถี่ มีความสำคัญมากในการประยุกต์ใช้งานที่เวลาจริง (Real time) เมื่อมีการทิก (Tick) ของสัญญาณนาฬิกาจะได้ตัวอย่างอินพุต 1 ตัวอย่างจากนั้นตัวกรองความถี่จะต้องสร้างตัวอย่างของเอาต์พุตออกมา 1 ตัวอย่างด้วย (ในความเป็นจริงจะต้องไม่มีสัญญาณอินพุตก่อนเวลา $n=0$ แน่นอน)

ข้อควรคำนึงถึงอีกอันหนึ่งของตัวกรองความถี่แบบ IIR ก็คือความมีเสถียรภาพ (stable) ถ้าตัวกรองความถี่ไม่เสถียรภาพ (Unstable) จะทำให้ลำดับของสัญญาณเอาต์พุตเพิ่มขึ้นอย่างไม่มีขอบเขต ถ้าสัญญาณอินพุตยังคงป้อนให้อยู่ ซึ่งเสถียรภาพของตัวกรองความถี่แบบ IIR จะขึ้นอยู่กับตำแหน่งโพลของฟังก์ชันระบบในระนาบ z ตัวกรองความถี่แบบ IIR ที่คุณสมบัติเป็น causal LTI จะมีเสถียรภาพ ถ้าค่าโพลเป็นไปตามเงื่อนไข $|\alpha_k|<1$ ซึ่งหมายความว่าตำแหน่งโพลทุกตัวจะต้องอยู่ในวงกลมหนึ่งหน่วย (Unit circle) บนระนาบ z นั่นเอง

3.2 โครงสร้างของตัวกรองความถี่แบบ IIR (Structures for IIR filter)

ในหัวข้อนี้จะกล่าวถึงโครงสร้างของตัวกรองความถี่แบบ IIR โดยแสดงในรูปของ Signal flow graph [15] โดยจะกล่าวถึงเฉพาะโครงสร้างแบบ Direct Forms เนื่องจากเป็นโครงสร้างที่ใช้ในวิทยานิพนธ์ฉบับนี้ โครงสร้างที่ดีและเหมาะจะนำไปใช้งานจะต้องเป็นโครงสร้างที่มีความซับซ้อนน้อย คือ มีการคูณและตัวหน่วง (delay) ที่น้อย ทั้งนี้เพราะโดยทั่วไปแล้วการคูณ จะต้องใช้เวลามากและมีราคาสูงเมื่อนำไปทำเป็นฮาร์ดแวร์ และจำนวนตัวหน่วง หมายถึง จำนวนหน่วยความจำซึ่งหากมีตัวหน่วงน้อย หน่วยความจำที่ใช้ก็น้อยลงด้วย และสิ่งที่ได้จากลดจำนวนการคูณอีกอย่างก็คือจะทำให้ความเร็วในการทำงานของ CPU เร็วขึ้น ซึ่งจะได้กล่าวในรายละเอียดของแต่ละแบบ ดังต่อไปนี้

3.2.1 โครงสร้างแบบ Direct Forms

จากสมการที่ (3.1) และ (3.2) ประกอบไปด้วยการบวกจำนวน $M+1$ ครั้ง สำหรับพจน์อินพุต $x(n),x(n-1),...,x(n-M)$ และ N ครั้งสำหรับพจน์เอาต์พุต $y(n-1),y(n-2),...,y(n-N)$ และแต่ละพจน์จะถูกถ่วงน้ำหนักด้วยสัมประสิทธิ์ของตัวกรองความถี่ จากสมการที่ (3.1) สามารถเขียนใหม่ได้ว่า

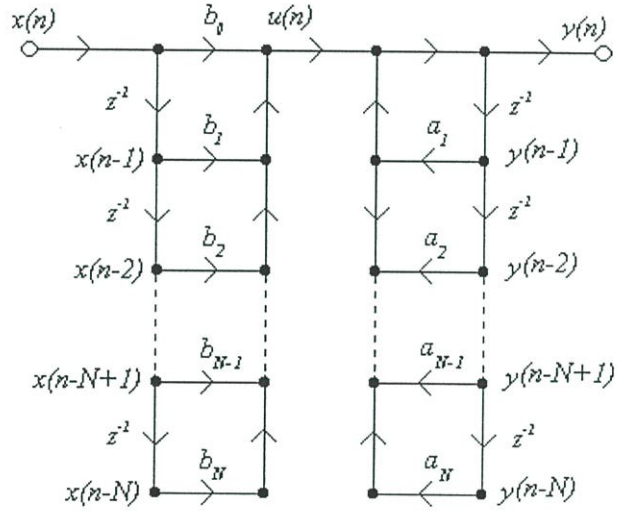
$$u(n)=b_0x(n)+b_1x(n-1)+b_2x(n-2)+...+b_Mx(n-M) \tag{3.4}$$

$$y(n)=-a_1y(n-1)-a_2y(n-2)+...+a_Ny(n-N)+u(n) \tag{3.5}$$

จากสมการ (3.4) และ (3.5) จะเห็นว่าตัวกรองความถี่แบบ IIR จะประกอบด้วยระบบย่อย (Subsystem) สองระบบต่อกันเป็นชั้น (Cascade) กัน กล่าวคือ ชั้นที่หนึ่งเป็นของฟังก์ชันเศษ ในสมการที่ (3.2) และชั้นที่สองเป็นของหนึ่งหารด้วยฟังก์ชันส่วนของสมการที่ (3.2) เมื่อนำสองส่วนนี้มารวมกันจะได้โครงสร้างแบบ Direct Form I ซึ่งสามารถเขียนเป็น signal flow graph สำหรับตัวกรองความถี่อันดับ N ตามรูปที่ 3.1

จะสังเกตเห็นว่าโครงสร้างแบบ Direct Form I จะมีจำนวนพจน์ทั้งหมด $M+N+1$ พจน์ คือ $x(n),x(n-1),...,x(n-M),y(n-1)y(n-2),...,y(n-N)$ แต่ละพจน์จะถูกคูณด้วยสัมประสิทธิ์ และนำบวกกัน ซึ่งจำนวนของการคูณสำหรับตัวอย่างเอาต์พุตแต่ละค่าจะเท่ากับจำนวนสัมประสิทธิ์ที่เป็นNontrivial เท่านั้น (การคูณด้วย 1, -1 และ 0 จะไม่นำมาคิดเป็นการคูณ)

จากสมการที่ (3.2) ตัวเศษของฟังก์ชันระบบ ต้องการสัญญาณอินพุตเป็น $x(n)$ และให้สัญญาณเอาต์พุตเป็น $u(n)$ และตัวส่วน ต้องการสัญญาณอินพุตเป็น $u(n)$ และให้สัญญาณเอาต์พุตเป็น $y(n)$ จากรูปที่ 3.1 ถ้าทำการพลิก (reverse) แต่ละระบบย่อย จะทำให้ตัวอย่างข้อมูลที่ถูกเก็บในระบบย่อยที่ 1 สามารถเก็บไว้ร่วมกับระบบย่อยที่ 2 ได้ ซึ่งโครงสร้างลักษณะนี้จะเรียกว่า Direct Form II หรือ Canonic direct form ซึ่งมีข้อดีคือต้องการจำนวนตัวหน่วงน้อยกว่า โดยจะมีค่าสูงสุดเท่ากับ M, N และการคูณเท่ากับ $M+N+1$ ครั้ง โครงสร้างแบบ Direct Form II จะมีสมการผลต่างคือ



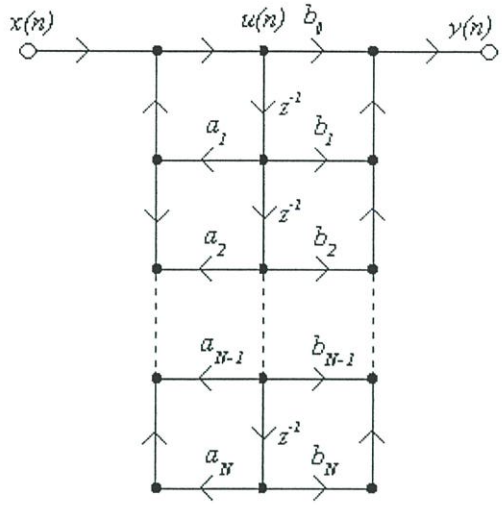
รูปที่ 3.1 Signal flow graph ของโครงสร้างแบบ Direct Form I สำหรับระบบอันดับ N

$$u(n)=x(n)-a_1u(n-1)-a_2u(n-2)-...-a_Nu(n-N) \tag{3.6}$$

$$y(n)=b_0u(n)+b_1u(n-1)-b_2u(n-2)-...-b_Mu(n-M) \tag{3.7}$$

และ signal flow graph แสดงดังรูปที่ 3.2

ตัวกรองความถี่อะแดปทีฟแบบนอตช์ที่นำเสนอในวิทยานิพนธ์เป็นแบบ IIR อันดับสอง (2^{nd} order) และใช้โครงสร้างแบบ direct form I และเนื่องจากตัวกรองความถี่แบบ IIR จะมีปัญหาทางด้านความมีเสถียรภาพกล่าวคือ หากมีโพลเพียง 1 ตัวอยู่นอกวงกลมหนึ่งหน่วยก็จะทำให้ตัวกรองความถี่ไร้เสถียรภาพทันที



รูปที่ 3.2 Signal flow graph ของโครงสร้างแบบ Direct From II สำหรับระบบอันดับ N

3.3 การประมาณค่าตัวกรองแบบนอตช์

หัวข้อนี้จะกล่าวถึง Adaptive Notch Filter (ANF) ซึ่งใช้ในการประมาณสัญญาณคลื่นซายน์ที่ไม่ทราบความถี่ซึ่งปะปนอยู่กับสัญญาณรบกวน การประมาณตัวกรองแบบนอตช์ จากผลตอบสนองทางอุดมคติ ให้อยู่ในรูปของฟังก์ชันแบบเศษส่วน (Rational function) ที่มีความสัมพันธ์กับโครงสร้างของตัวกรองดิจิตอลแบบ Direct form [16-18] โดยจะสมมติให้สัญญาณอินพุต เป็นสัญญาณซายน์ความถี่เดียวปนมากับสัญญาณรบกวน ส่วนตัวกรองแบบนอตช์ที่ใช้โครงสร้างแบบ Direct form ทำการประมาณ notch filter แบบ Direct form ด้วยกับสมการต่อไปนี้ คือ

$$\hat{H}(z)=\frac{N(z)}{N(\rho z)} \tag{3.8}$$

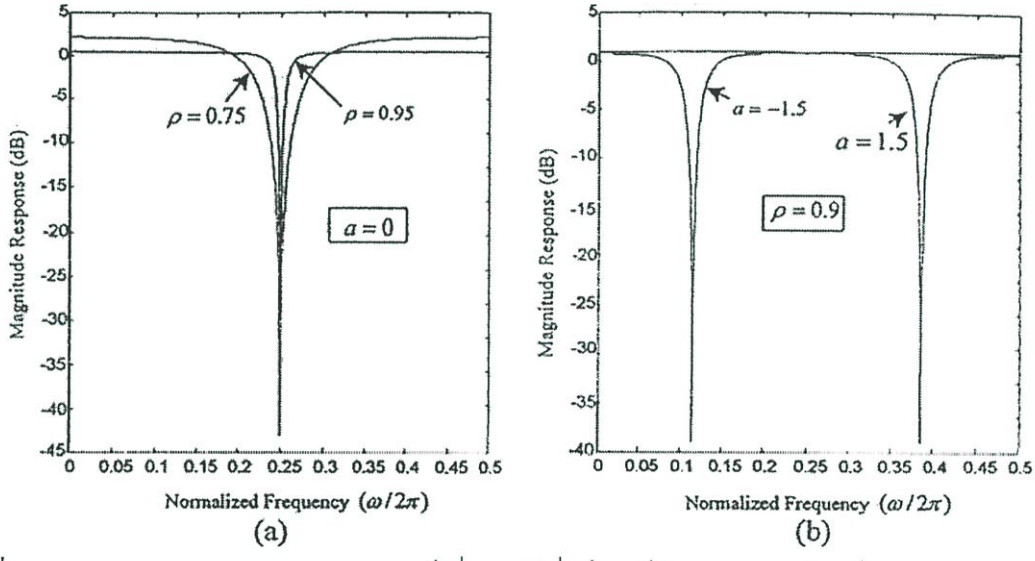
เมื่อ $N(z)$ เป็นโพลีโนเมียลที่มีซีโร่อยู่บนวงกลมหนึ่งหน่วย ในขณะที่ ρ เป็นค่าคงที่ มีค่าใกล้เคียงแต่น้อยกว่า 1 ดังนั้น โพลของตัวกรองจะยังคงอยู่ในบริเวณที่มีความเสถียรภาพ การออกแบบตัวกรองความถี่ด้วยกับฟังก์ชันแบบเศษส่วนนั้น จะนิยมสร้างจาก ตัวกรองความถี่อันดับสองถ้าต้องการจำนวนอันดับที่สูงขึ้น ก็จะนำเอาตัวกรองอันดับสองหลาย ๆ ตัวมาต่อ cascade กัน ดังนั้นจะได้ ฟังก์ชันถ่ายโอน (Transfer function) ของ notch filter อันดับสองดังสมการ คือ

$$\hat{H}(z)=\frac{1+az^{-1}+z^{-2}}{1+\rho az^{-1}+\rho^2 z^{-2}} \quad ,0<\rho<1 \tag{3.9}$$

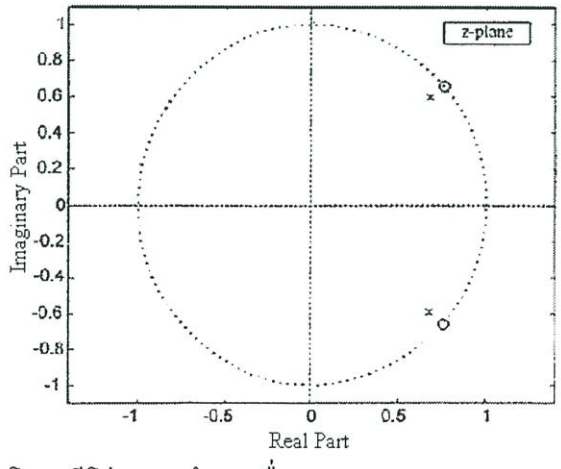
ความถี่นอตช์จะกำหนดได้ตามสมการ คือ

$$\omega_0 = \cos^{-1}(-a/2) \tag{3.10}$$

จะได้ว่า $-2<a<2$ รูปที่ 3.3 แสดงผลตอบสนองทางขนาดต่อความถี่ $\left|\hat{H}(e^{j\omega})\right|$ โดยการปรับเปลี่ยนพารามิเตอร์ a และ ρ กราฟแสดงให้เห็นว่าเมื่อ ρ เข้าใกล้ 1 จะทำให้แบนด์วิดของตัวกรองลดลง ทำให้ $\left|\hat{H}(e^{j\omega})\right|$ มีความใกล้เคียงกับผลตอบสนองทางความถี่ในอุดมคติ a เป็นพารามิเตอร์ที่ใช้กำหนดความถี่นอตช์ตามสมการที่ (3.10) และรูปที่ 3.4 แสดงตำแหน่ง โพล-ซีโร่ ของสมการที่ (3.9) เมื่อกำหนดให้ $\rho = 0.9$ และ $a = -1.5$



รูปที่ 3.3 ผลตอบสนองทางขนาดต่อความถี่ $\left|\hat{H}(e^{j\omega})\right|$ เมื่อเปลี่ยนค่าสัมประสิทธิ์ ρ และ a



รูปที่ 3.4 ตำแหน่ง โพล-ซีโร่ ของ $\hat{H}(z)$ เมื่อ $\rho = 0.9$ และ $a = -1.5$

3.4 สรุป

ในบทนี้กล่าวถึงอะแดปทีฟ IIR นอตช์ฟิลเตอร์ที่ใช้โครงสร้างแบบ Direct form โดยใช้ฟังก์ชันถ่ายโอน $H(z)$ ซึ่งฟังก์ชันถ่ายโอนนี้จะบังคับให้ซีโร่อยู่บนวงกลมหนึ่งหน่วย และโพลถูกบังคับให้อยู่ในวงกลมหนึ่งหน่วยและอยู่ในแนวรัศมีเดียวกันกับซีโร่ การออกแบบลักษณะนี้มีข้อดีหลายประการ เช่น มีเสถียรภาพสูง หมายความว่า โพลจะถูกคูณด้วยแฟกเตอร์ที่ทำให้ไม่สามารถออกจากวงกลมหนึ่งหน่วยได้เลย คำตอบที่ได้มีความถูกต้องสูง หมายความว่ามี Variance และ Bias ต่ำ และทำงานได้เร็ว ซึ่งทำให้ ANF มีความใกล้เคียงในอุดมคติมากขึ้น

บทที่ 4
ทฤษฎีเจเนติก

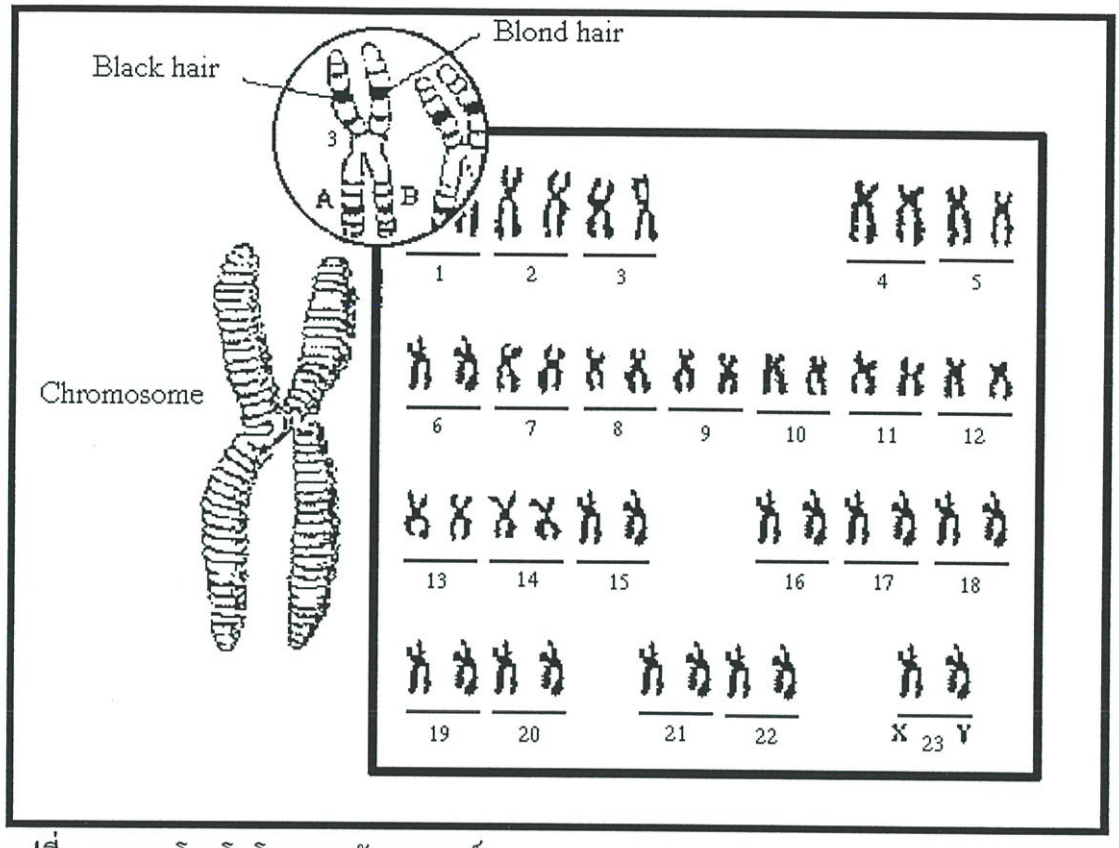
การจำลองการทำงานของธรรมชาติ เพื่อนำมาช่วยในการหาคำตอบที่ดีที่สุด (Optimal solution) ได้มีการนำไปศึกษาวิจัย เช่น นิวรอนเน็ตเวิร์ค (Neural network) ฟัซซี่ลอจิก (Fuzzy logic) เป็นต้น ส่วนเจเนติกอัลกอริทึมก็เป็นอีกวิธีการหนึ่งที่จำลองการทำงานทางชีววิทยา ในการกำเนิดประชากรรุ่นใหม่หรือขยายเผ่าพันธุ์ในรุ่นลูก รุ่นหลานต่อไป ซึ่งอาศัยพื้นฐานความคิดของวิวัฒนาการทางธรรมชาติถ่ายทอดลักษณะต่าง ๆ ทางพันธุกรรม โดยปฏิบัติตามกระบวนการทางพันธุศาสตร์ เพื่อจะใช้ในการหาคำตอบที่ดีที่สุด หรือใกล้เคียงที่สุดของปัญหา

4.1 พื้นฐานด้านชีววิทยา

เมนเดล (Mendel) บิดาแห่งวิชาพันธุศาสตร์ ค้นพบว่า ยีนส์ (Genes) หน่วยเก็บลักษณะทางกรรมพันธุ์เป็นตัวกำหนดลักษณะภายนอก ซึ่งยีนส์หลาย ๆ ยีนส์จะเรียงตัวกันอยู่บนเส้นโครโมโซม (Chromosome) หรือก็คือในเซลของสิ่งมีชีวิตโดยจะอยู่กันเป็นคู่ ๆ แต่จะแตกต่างกันที่ค่าลักษณะต่าง ๆ ในแต่ละยีนส์เรียกว่า แอลลี (Allele) ซึ่งแบบต่าง ๆ ของยีนส์ที่มีแอลลีต่างกันในแต่ละตำแหน่งยีนส์เดียวกันเรียกว่า ยีโนไทป์ (Genotype) เช่น ในคนจะมีโครโมโซม 23 คู่ 46 โครโมโซม ซึ่งแต่ละโครโมโซมจะประกอบด้วยยีนส์ต่าง ๆ กันราว 1,250 ยีนส์ ตัวอย่างคู่โครโมโซมที่ 1 ในคน ดังรูปที่ 4.1 ซึ่งประกอบด้วยยีนส์ลักษณะสีผม สีผิว สีตา และอื่น ๆ อีกประมาณ 1,247 ลักษณะ และโครโมโซม 1A มีแอลลีของยีนส์ลักษณะสีผม คือ ผมสีดำ ส่วนโครโมโซม 1B มีแอลลีของยีนส์ลักษณะสีผมคือ ผมสีบลอนด์ [28-29]

การถ่ายทอดลักษณะทางพันธุกรรมเป็นการถ่ายทอดลักษณะต่าง ๆ ของสิ่งมีชีวิตที่เกิดขึ้นเมื่อมีการแบ่งตัวของเซลสิ่งมีชีวิต ซึ่งมี 2 แบบ คือ

i) การเพิ่มจำนวนเซล เป็นการแบ่งตัวแบบไมโตสิส (Mitosis) โดยโครโมโซมแต่ละตัวจะขยายตัวเพิ่มจำนวนตัวเองขึ้นเป็นสอง และเยื่อหุ้มนิวเคลียส (Nucleous) จะสลายลงเพื่อดึงแยกโครโมโซมที่เพิ่มจำนวนขึ้นออกจากโครโมโซมเดิมเป็นสองด้าน แล้วเยื่อหุ้มนิวเคลียสจะถูกสร้างขึ้นใหม่กลายเป็นเซลใหม่ 2 เซล ที่มีโครโมโซมเหมือนเดิม ดังรูปที่ 4.2



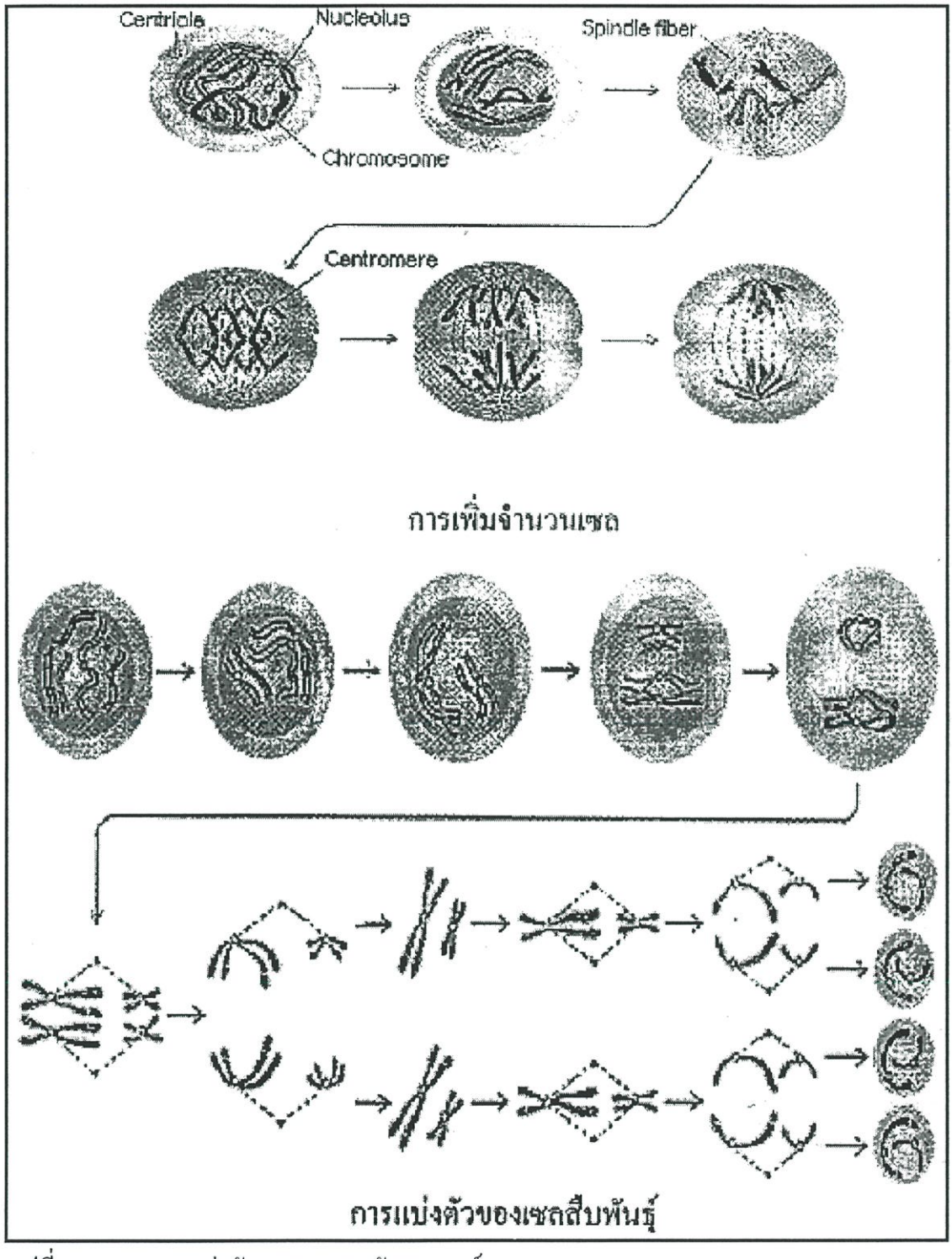
รูปที่ 4.1 แสดงโครโมโซมทางพันธุศาสตร์

ii) การแบ่งตัวของเซลสืบพันธุ์ เป็นการแบ่งตัวแบบไมโอสิส (Meiosis) โดยโครโมโซมจากเซลพ่อ 1 โครโมโซม และจากเซลแม่ 1 โครโมโซม จะเริ่มจับคู่กันที่โครโมโซมชนิดเดียวกัน ในขณะเดียวกันโครโมโซมแต่ละตัวทั้งที่มาจากพ่อและมาจากแม่ ต่างก็จำลองแบบของตนเพิ่มขึ้นมาอีกแต่ละโครโมโซม ทำให้ได้จำนวนโครโมโซมทั้งหมดเพิ่มขึ้นเป็นสองเท่า และดำเนินการทางพันธุกรรมจนถึงระยะแบ่งตัว โครโมโซมพ่อพร้อมกับแบบจำลองและโครโมโซมแม่พร้อมกับแบบจำลองที่ได้ จะแยกคู่ไปรวมกันเป็น 2 นิวเคลียส กลายเป็นเซลใหม่ 2 เซล ซึ่งจะแบ่งตัวต่อทันที โดยแต่ละโครโมโซมพ่อแยกตัวออกจากแบบจำลอง และโครโมโซมแม่ก็แยกตัวออกจากแบบจำลองรวมกันเป็นเซลใหม่ 4 เซล ดังรูปที่ 4.2

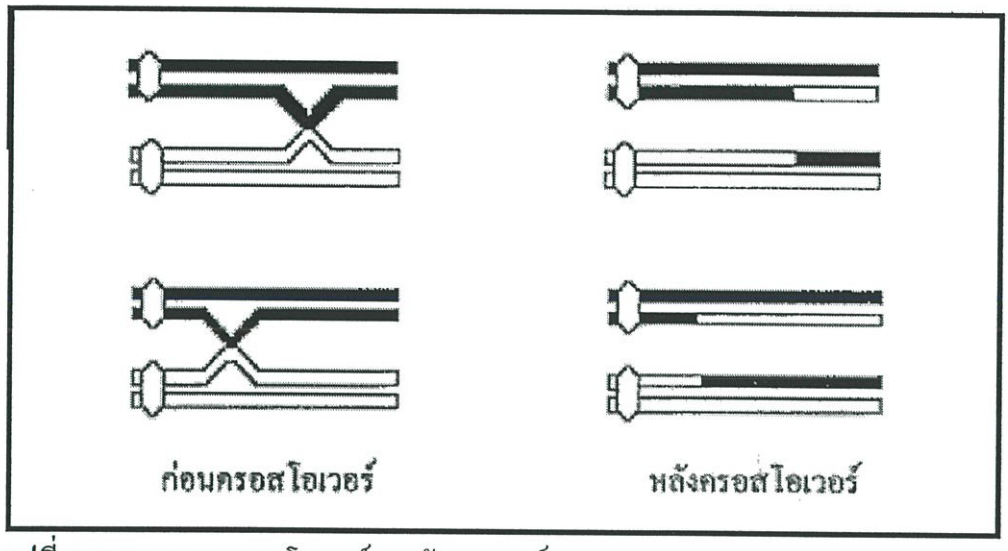
วิธีการทางพันธุศาสตร์ในระหว่างที่เกิดการแบ่งตัวของไมโอสิสนั้น โครโมโซมจะมีโอกาสแลกเปลี่ยนส่วนบางส่วนซึ่งกันและกัน อันเป็นสาเหตุที่ทำให้เกิดปรากฏการณ์ที่เรียกว่า ครอสโอเวอร์ (Crossover) ของลักษณะต่าง ๆ ขึ้น ซึ่งการครอสโอเวอร์นั้นเกิดขึ้นขณะที่มีการจำลองแบบเพิ่มขึ้นและเกิดขึ้นระหว่างโครโมโซมพ่อกับโครโมโซมแม่ ไม่ใช่เกิดขึ้นระหว่างโครโมโซมพ่อกับแบบจำลอง ซึ่งการครอสโอเวอร์จะทำให้เกิดการเปลี่ยนแปลงลักษณะของยีนส์ต่าง ๆ ของคู่โครโมโซมพ่อกับแม่ โดยเนื่องจากยีนส์เรียงตัวกันบนเส้นโครโมโซมนั้นไม่ได้อยู่กันอย่างหนาแน่น แต่มีระยะห่างกันอย่างไม่สม่ำเสมอ เพราะคุณสมบัติของยีนส์นั้นเป็นโมเลกุลของสารโปรตีนประกอบตัวกัน

ทางเคมี ช่องว่างระหว่างยีนส์นี่เองจะเป็นตำแหน่งที่แตกออกมาได้เวลาจะครอสโอเวอร์ และแลกเปลี่ยนยีนส์ของโครโมโซมโดยส่วนที่อยู่หลังรอยแตก โดยจะถูกย้ายไปอยู่อีกโครโมโซมหนึ่งทั้งหมด นอกจากนี้ยังจะสามารถแตกอีกกี่แห่งก็ได้ ซึ่งผลนั้นขึ้นอยู่กับความสามารถที่เชื่อมกันมากน้อยเพียงไรของช่องว่างระหว่างยีนส์ ดังรูปที่ 4.3 ประโยชน์ที่เกิดจากครอสโอเวอร์ที่เราเห็นได้คือทำให้มีโอกาสที่จะลักษณะต่าง ๆ กันมาอยู่รวมกันได้หลายแบบมากขึ้น ทำให้สิ่งมีชีวิตรุ่นลูกที่เกิดขึ้นมามีความหลากหลายมากขึ้น และอาจทำให้มีโอกาสเกิดสิ่งมีชีวิตที่มีลักษณะต่าง ๆ ที่ดีพอเหมาะรวมอยู่ด้วยกันได้อย่างพอดี เหมาะสมกับสิ่งแวดล้อม ถ้าการเกิดเซลใหม่โดยถ่ายทอดโครโมโซมไม่มีครอสโอเวอร์แล้ว โครโมโซมใดเคยมียีนส์ลักษณะใดก็จะมีลักษณะนั้นอยู่เรื่อย ๆ รุ่นลูกก็จะมีลักษณะเช่นเดียวกัน โอกาสที่สิ่งมีชีวิตนั้นจะเจริญ หรือปรับตัวให้ดีขึ้นย่อมมีได้ยากกว่าการเปลี่ยนลักษณะยีนส์ใหม่ หลาย ๆ แบบมากขึ้น

ลักษณะต่าง ๆ ของสิ่งมีชีวิตจะสามารถอยู่รอดได้โดยการคัดเลือกทางธรรมชาติ คือ คัดเลือกโครโมโซมที่มีลักษณะที่ทำให้สิ่งมีชีวิตแข็งแรงเพียงพอหรือเหมาะสมต่อสภาพแวดล้อมซึ่งจะสามารถอยู่รอดและถ่ายทอดไปยังลูกหลาน ดังนั้นการคัดเลือกของธรรมชาติเพื่อถ่ายทอดลักษณะทางพันธุกรรมเป็นเพียงส่วนประกอบของการเปลี่ยนแปลงของสิ่งมีชีวิตเท่านั้น มิวเตชัน (Mutation) หรือการผ่าเหล่า คือการเปลี่ยนแปลงลักษณะของยีนส์ไปจากเดิมที่ควรเป็นไปตามการถ่ายทอด ซึ่งเป็นต้นเหตุของการเกิดลักษณะที่แปลก ๆ ออกไปมากมายสำหรับสิ่งมีชีวิตหนึ่ง ๆ ซึ่งเท่ากับเป็นการให้โอกาสแก่ธรรมชาติในการจะเลือกลักษณะใหม่ ๆ มากขึ้น เนื่องจากกระบวนการวิวัฒนาการโดยธรรมชาติเองนั้นช้ามาก เพราะกว่าที่ธรรมชาติจะปรับสภาพแวดล้อมให้สิ่งมีชีวิตค่อย ๆ ปรับตัวเองให้เหมาะสมนั้นมีโอกาสน้อยมาก การผ่าเหล่านั้นทุกลักษณะในแต่ละยีนส์ย่อมมีโอกาสที่จะเกิดความเปลี่ยนแปลงไปจากเดิมได้พอ ๆ กัน และถ้าเหมาะสมกับสภาพแวดล้อมขณะนั้นก็จะคงอยู่ต่อไป แต่ถ้าการเปลี่ยนแปลงเป็นลักษณะใดเกิดผิดจังหวะ คือไม่เหมาะสมกับสภาพขณะนั้น ๆ ก็จะไม่ถูก คัดเลือกและหายไป ซึ่งข้อสรุปนี้ได้จากการทดลองโดยการนำยีนส์ของแบคทีเรียมาผสมกันและจัดสภาพแวดล้อมที่ไม่อำนวยต่อการผสมกันแล้ว ยีนส์จะปรับตัวเองเพื่อเร่งขบวนการผสมพันธุ์จนได้ผลดีในรุ่นหลัง ๆ ตัวอย่างของการผ่าเหล่าในอดีตคือ การกำเนิดของปลาทองนั้น มีต้นกำเนิดมาจากการกลายพันธุ์หรือการผ่าเหล่าของปลาชิว และในปัจจุบันลักษณะใหม่ที่เกิดจากการผ่าเหล่าก็ยังคงมีให้เห็นอยู่ เช่น ความสามารถของเชื้อโรคแบคทีเรียในการต้านทานต่อยาฆ่าเชื้อหรือเซลผิดปกติอันเนื่องมาจากกัมมันตภาพรังสีต่าง ๆ ซึ่งมีผลต่อสารภายในเซล ซึ่งทำให้ลักษณะของยีนส์ในเซลเปลี่ยนไป



รูปที่ 4.2 แสดงการแบ่งตัวของเซลทางพันธุศาสตร์



รูปที่ 4.3 แสดงการครอสโอเวอร์ทางพันธุศาสตร์

4.2 เจเนติกอัลกอริทึมพื้นฐาน

การนำเสนอวิธีการแก้ปัญหาโดยเจเนติกนั้น เริ่มขึ้นตั้งแต่ปี ค.ศ. 1975 โดย John Holland [29] โดยมีแนวคิดเลียนแบบการทำงานของธรรมชาติในด้านการคัดเลือกทางธรรมชาติ (Natural selection) โดยมีกฎอยู่ว่าสิ่งมีชีวิตที่แข็งแรงและเหมาะสมกับสภาพแวดล้อมจะมีโอกาสอยู่รอดได้มากกว่า ซึ่งทำให้สิ่งมีชีวิตต่าง ๆ มีการปรับตัวเองให้เข้าสภาพแวดล้อมได้ นั้นก็รวมถึงการเปลี่ยนแปลงส่วนประกอบต่าง ๆ ในโครโมโซม ซึ่งเป็นส่วนประกอบของสิ่งมีชีวิตเพื่อให้ได้โครโมโซมที่ดีขึ้นและส่งผ่านลักษณะที่ดีนั้นสู่รุ่นลูกหลานต่อ ๆ ไป โดยวิธีการที่ใช้ในการปรับปรุงโครโมโซมนั้นเรียกว่าการดำเนินการทางพันธุกรรมศาสตร์ (Genetic operation) ซึ่งประกอบไปด้วย การผสมพันธุ์หรือการครอสโอเวอร์ และการกลายพันธุ์หรือมิวเตชัน เป็นต้น จากแนวความคิดเกี่ยวกับวิวัฒนาการของสิ่งมีชีวิตที่ถ่ายทอดลักษณะที่เหมาะสมกับการอยู่รอดโดยผ่านโครโมโซมนี้ ทำให้ Holland เกิดแนวความคิดขึ้นมาว่าน่าจะนำไปใช้กับการทำงานร่วมกับคอมพิวเตอร์ได้ เขาจึงได้ทำการวิจัยโดยจำลองการทำงานกับแบบทดลองต่าง ๆ โดยมีจุดมุ่งหมายที่จะศึกษาระบบปรับปรุงการประมวลผลเอง (Self adaptive process) และเพื่อสร้างระบบผู้เชี่ยวชาญ (Artificial system software) จึงได้ค้นพบวิธีการใหม่เรียกว่า เจเนติกอัลกอริทึม (Genetic Algorithm : GA)

GA เป็นวิธีการค้นหาคำตอบโดยการเลียนแบบการทำงานของธรรมชาติและพันธุกรรมศาสตร์ ซึ่งอาศัยหลักการสุ่มเพื่อปรับปรุงความสามารถในการหาคำตอบที่ดีขึ้น การค้นหาคำตอบจากรุ่นหนึ่งไปยังรุ่นถัดไป ตามวิวัฒนาการของธรรมชาติคำตอบในรุ่นถัดไปที่เกิดขึ้นจะสร้างความสัมพันธ์ของโครงสร้างต่าง ๆ ที่ประกอบไปด้วยค่าตัวแปรที่เหมาะสมดีในรุ่นก่อน ดังนั้น จึงมีแนวโน้มของคำตอบที่ดีขึ้นเรื่อย ๆ จะเห็นได้ว่า GA อยู่บนพื้นฐานของวิธีการสุ่มแต่มีหลักการและประสิทธิภาพจากการคาดเดาคำตอบใหม่ จากสถิติคำตอบเดิมที่ดีอยู่แล้วซึ่งแตกต่างจากวิธีการทั่วไป คือ

i) GA ค้นหาคำตอบภายใต้โครงสร้างของปัญหาอันเกิดจากการกำหนดรหัส (coding) รูปแบบโครงสร้างจากกลุ่มตัวแปรต่าง ๆ ของปัญหานั้น ไม่ใช่ค้นหาคำตอบจากค่าของกลุ่มตัวแปรนั้น

ii) GA ค้นหาคำตอบโดยพิจารณาจากประชากรคำตอบ หรือกลุ่มคำตอบ ไม่ใช่พิจารณาจากคำตอบใดคำตอบหนึ่ง

iii) GA ค้นหาคำตอบจากผลลัพธ์ของกลุ่มค่าตัวแปรที่เป็นฟังก์ชันเป้าหมายของปัญหา

iv) GA ค้นหาคำตอบโดยอาศัยการถ่วงน้ำหนักความเหมาะสมของแต่ละคำตอบจากกลุ่มคำตอบนั้น ๆ

4.2.1 ฟังก์ชันเป้าหมายกับฟังก์ชันความเหมาะสม

การหาคำตอบที่ดีที่สุดของปัญหาของ GA มีพื้นฐานอยู่บนผลลัพธ์จากการหาคำตอบที่ผ่านมา วิธีการของ GA จะไม่พิจารณาขั้นตอนของการแก้ปัญหา แต่จะพิจารณาโดยตัดสินว่าคำตอบใหม่ที่ได้รับดีขึ้นหรือไม่ หรือเป็นคำตอบที่ใกล้เคียงคำตอบที่ต้องการหรือไม่ จากฟังก์ชันเป้าหมาย (Object function : f) เนื่องจากแต่ละปัญหาจะสามารถกำหนดฟังก์ชันเป้าหมายซึ่งเป็นฟังก์ชันที่แสดงความสัมพันธ์ของแต่ละตัวแปร พารามิเตอร์ เงื่อนไข หรือข้อกำหนดต่าง ๆ ของปัญหานั้น ๆ ที่ระบุคำตอบใดคำตอบหนึ่งที่สามารถเป็นไปได้ ณ ค่าพารามิเตอร์ เงื่อนไข หรือข้อกำหนดชุดดังกล่าว สำหรับฟังก์ชันความเหมาะสม (Fitness Function : F) เป็นฟังก์ชันที่กำหนดค่าความเหมาะสม (Fitness) ของแต่ละโครโมโซมเปรียบเสมือนค่าความสามารถในการอยู่รอดของแต่ละโครโมโซม และเป็นฟังก์ชันที่กำหนดโอกาส หรือสัดส่วนที่แต่ละโครโมโซมที่เหมาะสมจะถูกคัดเลือกมากน้อยเพียงใด นั่นคือ ฟังก์ชันความเหมาะสม ซึ่งจะเป็นฟังก์ชันที่แสดงถึงค่าคำตอบที่เกิดขึ้นจากชุดตัวแปรของปัญหาของโครโมโซมนั้นดีเพียงใด โดยทั่วไปแล้วเรามักใช้ฟังก์ชันเป้าหมายเป็นฟังก์ชันความเหมาะสม หรืออาจใช้ฟังก์ชันเป้าหมายที่ถูกปรับให้เหมาะสมกับการนำเสนอ GA เป็นฟังก์ชันความเหมาะสมก็ได้

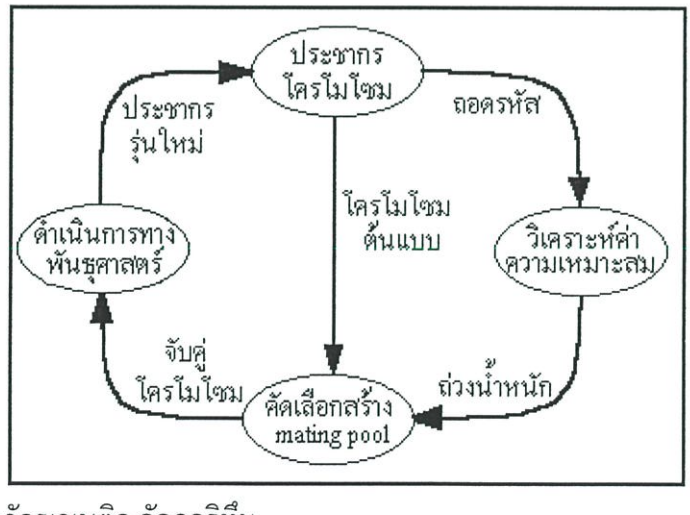
4.2.2 รูปแบบโครโมโซม

ทราบกันแล้วว่าวิวัฒนาการทางธรรมชาติ และชีววิทยานั้นเป็นความเปลี่ยนแปลงต่าง ๆ ของสิ่งมีชีวิตซึ่งจะเกิดขึ้นในโครโมโซม ดังนั้นจุดเริ่มต้นของการจำลองแบบทางธรรมชาติของ GA เพื่อใช้แก้ปัญหาจึงเริ่มจากการมองปัญหาเทียบเท่ากับโครโมโซมชนิดหนึ่ง ประกอบด้วยยีนส์ลักษณะต่าง ๆ ซึ่งหมายถึงลำดับข้อมูลต่าง ๆ ที่จะแปรความหมายแล้วให้ค่าคำตอบของปัญหาค่าหนึ่ง การมองค่ายีนส์ของ GA ให้ถือเสมือนยีนส์ทางพันธุกรรมที่แสดงความหมายหรือเป็นตัวแทนคำตอบใดคำตอบหนึ่ง หรือลักษณะใดลักษณะหนึ่งทางพันธุกรรม ในทางพันธุศาสตร์นั้นยีนส์เป็นตัวแสดงลักษณะที่อยู่รอดในสภาพแวดล้อมขณะนั้น สำหรับ GA นั้น ยีนส์เป็นตัวแสดงค่าคำตอบของปัญหาที่แปรผันไปตามการประยุกต์ใช้งาน ซึ่งโดยทั่วไปยีนส์หมายถึงตัวแปร พารามิเตอร์ เงื่อนไข หรือข้อกำหนดต่าง ๆ ที่เป็นองค์ประกอบของปัญหา ดังนั้น การกำหนดรูปแบบโครโมโซมของแต่ละปัญหาโดย

การแปลงตัวแปร พารามิเตอร์ เงื่อนไข หรือข้อกำหนดต่าง ๆ ให้อยู่ในรูปลำดับของยีนส์บนโครโมโซม หรือเรียกว่า สตริง (String) อันประกอบไปด้วย บิต (Bit) หรือเรียกว่าอักขระ (Character) ซึ่งลักษณะต่าง ๆ ที่เป็นได้ของแต่ละยีนส์คือค่าของบิต (Bit Value) หรือค่าตัวแปรพารามิเตอร์ต่าง ๆ ที่เป็นไปได้ และรูปแบบของค่าบิตที่จัดเรียงบนโครโมโซมคือ ยีโนไทป์ (Genotype) ที่แสดงถึงค่าของตัวแปร พารามิเตอร์ต่าง ๆ ที่เป็นไปได้ชุดหนึ่งหรือฟีโนไทป์ (Phenotype) นั่นเอง การกำหนดรูปแบบโครโมโซมของปัญหาให้เป็นตามแบบธรรมชาติโดยกำหนดรหัสในรูปแบบตัวเลข หรือตัวอักษรในช่วงที่จำกัดตามค่าตัวแปรหรือพารามิเตอร์ และประกอบรวมกันในจำนวนยีนส์หรือความยาวของโครโมโซมที่คงที่ เช่น หากต้องการหาค่าสูงสุดของฟังก์ชันทางคณิตศาสตร์ $y=x^2$ ที่ x เป็นจำนวนเต็มอยู่ในช่วง [0, 31] แล้ววิธีการของ GA ในการแก้ปัญหาโดยการกำหนดรูปแบบโครโมโซมจากการกำหนดรหัสตัวแปร x เป็นตัวเลขไบนารี 0 หรือ 1 จำนวน 5 ตำแหน่ง ซึ่ง x จะมีค่าตั้งแต่ 00000 ถึง 11111 คือ ค่า 0 ถึง 31 นั่นเอง

4.2.3 วัฏจักรเจเนติก อัลกอริทึม

เมื่อกำหนดรูปแบบโครโมโซม และฟังก์ชันความเหมาะสมของปัญหาแล้ว GA จะสามารถประมวลผลหาคำตอบของปัญหาได้ โดยสร้างวิวัฒนาการกลุ่มคำตอบในรุ่นต่อไปตามวัฏจักรการทำงานของ GA (Genetic Algorithm Cycle) ดังรูป 4.4 ซึ่งมี 4 ขั้นตอนคือ



รูปที่ 4.4 แสดงวัฏจักรเจเนติก อัลกอริทึม

i) สร้างประชากรโครโมโซมรุ่นเก่าตามแบบโครโมโซมที่กำหนดไว้ โดยประชากรต้นกำเนิด (Initial population) เกิดจากการสร้างชุดโครโมโซมต้นกำเนิด จากการสุ่มสร้างค่าแต่ละบิตของแต่ละโครโมโซม

ii) วิเคราะห์ค่าความเหมาะสมแต่ละโครโมโซม โดยถอดรหัสค่าตัวแปร พารามิเตอร์ต่าง ๆ ของแต่ละบิตในโครโมโซม และคำนวณค่าความเหมาะสมจากฟังก์ชันความเหมาะสมที่กำหนดไว้

iii) สร้าง Mating pool คือ ชุดโครโมโซมต้นแบบหรือโครโมโซมพ่อ-แม่ที่สามารถอยู่รอดเป็นต้นแบบ ซึ่งอาศัยการจำลองการคัดเลือกทางธรรมชาติ โดยพิจารณาถ่วงน้ำหนักจากค่าความเหมาะสมของแต่ละโครโมโซม หากโครโมโซมใดมีค่าความเหมาะมากก็มีโอกาสถูกคัดเลือกเป็นต้นแบบมาก

iv) ดำเนินการทางพันธุศาสตร์โดยการสุ่มจับคู่โครโมโซมต้นแบบใน Mating pool เพื่อสร้างประชากรโครโมโซมรุ่นใหม่ ซึ่งตัวดำเนินการทางพันธุศาสตร์ประกอบด้วย ครอสโอเวอร์ โดยการแลกเปลี่ยนค่าบิตบางส่วนของโครโมโซมซึ่งกันและกัน หรือมิวเตชัน โดยสุ่มเปลี่ยนบิตบางบิตของแต่ละโครโมโซม เป็นต้น

การค้นหาคำตอบของ GA จะประมวลผลซ้ำตามวัฏจักรของ GA จนกว่าจะได้คำตอบที่พอใจตามกฎเกณฑ์ที่ตั้งไว้ หรือในระยะเวลาตามจำนวนรุ่นที่ดำเนินการที่ต้องการ [31]

4.3 เจเนติก อัลกอริทึมแบบง่าย

GA ในยุคเริ่มแรกของของ Holland นั้นคือ เจเนติกอัลกอริทึมแบบง่าย (Simple Genetic Algorithm : SGA) ซึ่งมีขั้นตอนพื้นฐานที่มีกระบวนการไม่มากนักและง่ายในกาศึกษาและทำความเข้าใจในแต่ละขั้นตอนการทำงานของ GA เพื่อแก้ปัญหาในการหาคำตอบ แสดงดังไดอะแกรมในรูปที่ 4.5 คือแบ่งออกเป็นสองส่วนคือ ขั้นเตรียมการและขั้นตอนการทำงาน

สำหรับในส่วนของขั้นเตรียมการนั้นเป็นส่วนของการปรับรูปแบบของปัญหาให้เหมาะสมสำหรับการนำเสนอ GA เพื่อใช้ในการแก้ปัญหานั้น ๆ ประกอบด้วย

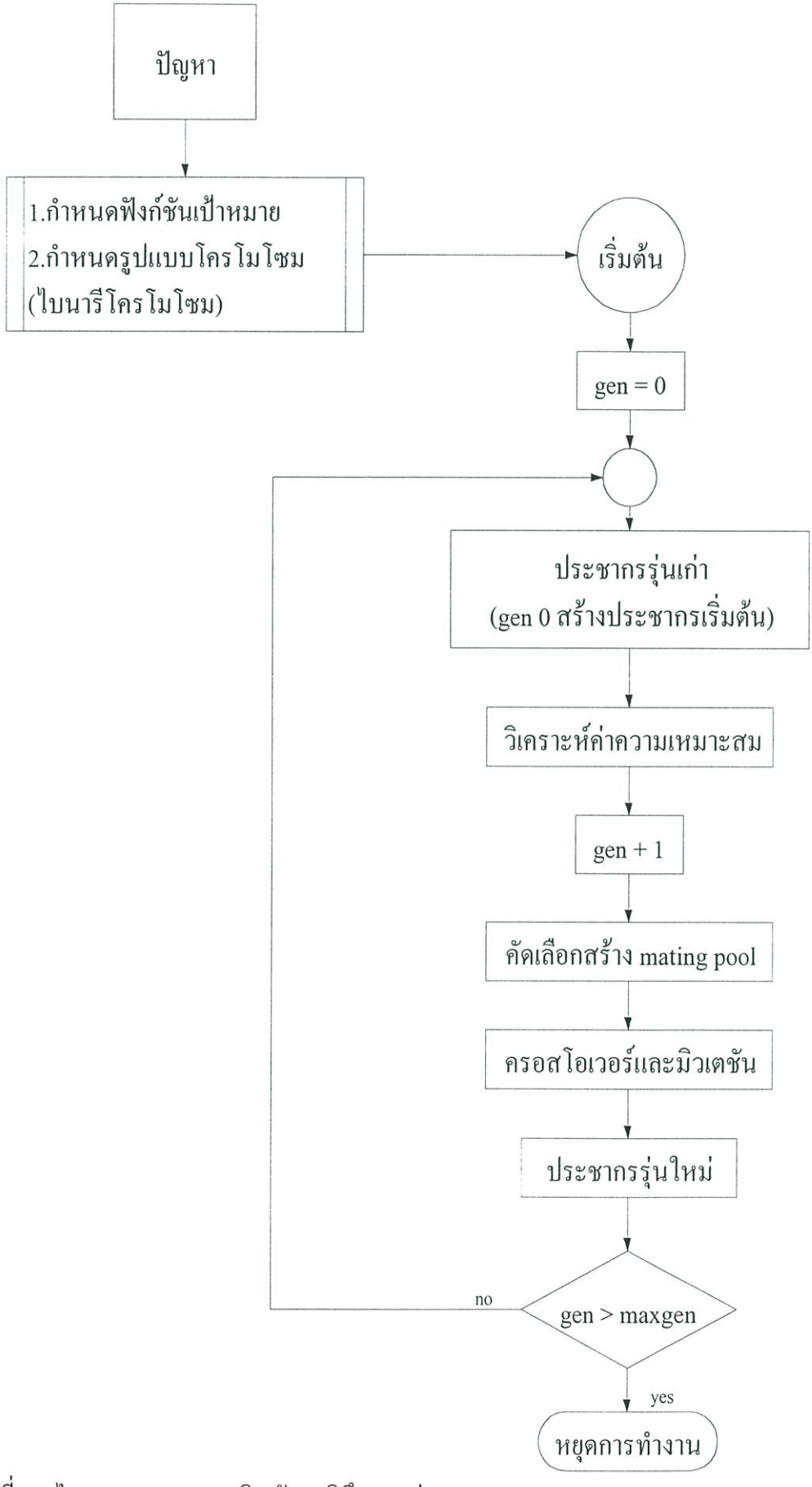
i) กำหนดฟังก์ชันความเหมาะสม เพื่อความสะดวกและง่ายต่อความเข้าใจในขั้นตอนการทำงานต่าง ๆ โดยจะกำหนดตัวอย่างปัญหาสำหรับอธิบายรายละเอียดการหาคำตอบของ SGA คือปัญหาการหาค่าสูงสุดของฟังก์ชัน $y = x^2$ ที่ x เป็นจำนวนเต็มอยู่ในช่วง [0, 31]

ตัวอย่าง : ฟังก์ชันเป้าหมาย คือ $f = x^2$

และกำหนดให้ฟังก์ชันความเหมาะสมคือ $F = x^2$

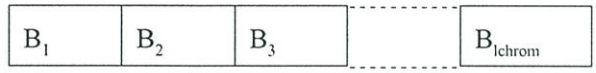
ซึ่งคำตอบที่ดีที่สุด คือ ค่า x ที่มีค่าความเหมาะสมสูงสุด (MAX(F))

ii) กำหนดรูปแบบโครโมโซม รูปแบบโครโมโซมของ SGA นั้นเป็นแบบไบนารี โดยค่าตัวแปรหรือพารามิเตอร์ของปัญหาจะถูกแปลงให้อยู่ในรูปของไบนารีโครโมโซม คือประกอบด้วยบิต



รูปที่ 4.5 ไดอะแกรมของเจเนติก อัลกอริทึมแบบง่าย

ที่มีค่าเป็น 0 หรือ 1 ซึ่งเป็นค่าในเลขฐานสอง และมีความยาว (Chromosome Length : Ichrom) ตามแต่จะกำหนด ซึ่งแสดงด้วยสัญลักษณ์ได้ดังนี้



ซึ่ง $B_l \in I\,[0, 1]$

ตัวอย่าง : วิธีการเข้ารหัสแบบไบนารีโดยแปลงค่าพารามิเตอร์ x ให้อยู่ในรูปไบนารีบิต 5 บิต (Ichrom=5) ดังนั้นโครโมโซมของปัญหาจะมีค่าอยู่ในช่วง 00000 ถึง 11111 ซึ่งเมื่อถอดรหัสแล้วจะทำให้ x มีค่าอยู่ในช่วง 0 ถึง 31 ตามที่ต้องการ

ในส่วนของรายละเอียดขั้นตอนการทำงานของ SGA จะเป็นขั้นตอนพื้นฐานเบื้องต้นแบบง่าย ประกอบด้วย

4.3.1 ประชากรรุ่นเก่า (Old population) เป็นโครโมโซมที่จะถูกคัดเลือกไปเป็นต้นแบบสำหรับสร้างประชากรรุ่นใหม่ (New population) ในวิวัฒนาการรุ่น (Generation : gen) ต่อไป โดยประชากรเริ่มต้นที่ gen = 0 จะถูกสร้างขึ้นโดยการสุ่มตามจำนวนโครโมโซมในแต่ละรุ่น (Population Size: popsize) ที่กำหนด

ตัวอย่าง : ลำดับ โครโมโซม

1	0 1 1 1 0
2	1 1 0 0 1
3	0 1 0 0 0
4	1 0 0 1 1

ชุดโครโมโซมในรุ่นเริ่มต้นนี้เป็นโครโมโซมที่กำหนดให้ในแต่ละรุ่นประกอบด้วย 4 โครโมโซม ซึ่งแต่ละโครโมโซมเกิดจากการสุ่มค่าไบนารี 0 หรือ 1 จำนวน 5 ครั้ง

4.3.2 วิเคราะห์ค่าความเหมาะสม เป็นขั้นตอนของการถอดรหัสจากรูปแบบโครโมโซมที่กำหนดไว้เพื่อคำนวณค่าความเหมาะสมตามฟังก์ชันความเหมาะสมของปัญหาของแต่ละโครโมโซม ในที่นี้ฟังก์ชันเป้าหมายหรือฟังก์ชันความเหมาะสม คือ $F = x^2$ ดังนั้นการวิเคราะห์ค่าความเหมาะสมของ SGA โดยถอดรหัสเลขฐาน 2 ของแต่ละโครโมโซมเป็นค่าของตัวแปร x และคำนวณค่าความเหมาะสม คือค่า x^2

ตัวอย่าง : ลำดับ	โครโมโซม	x	ค่าความเหมาะสม
1	0 1 1 1 0	14	196
2	1 1 0 0 1	25	625
3	0 1 0 0 0	8	64
4	1 0 0 1 1	19	361

ชุดโครโมโซมในรุ่นเริ่มต้นมีค่าความเหมาะสมเป็น 196, 625, 64 และ 361 ตามลำดับ

4.3.3 การคัดเลือก เป็นขั้นตอนที่จำลองแบบการคัดเลือกทางธรรมชาติเพื่อสร้าง Mating pool โดยคัดเลือกชุดโครโมโซมรุ่นเก่าให้เป็นโครโมโซมต้นแบบหรือโครโมโซมพ่อ-แม่ เพื่อใช้สร้างโครโมโซมลูกเป็นรุ่นต่อไป สำหรับการคัดเลือกของ SGA เป็นแบบอ้างอิงค่าความเหมาะสม (Fitness-based Selection) โดยพิจารณาค่าความเหมาะสมเป็นตัวตัดสินว่า โครโมโซมใดในรุ่นเก่ามีโอกาสจะถูกเลือกเป็นโครโมโซมพ่อ-แม่มากน้อยเพียงใด โครโมโซมที่มีค่าความเหมาะสมที่ดีจะถูกกำหนดน้ำหนักค่าความน่าจะเป็นที่จะถูกเลือกแต่ละครั้งสูง การกำหนดค่าความน่าจะเป็นที่จะถูกเลือกต่อการสุ่มเลือกแต่ละครั้ง (Probability of Selected Value : pselect) ของแต่ละโครโมโซม โดยกำหนดจากค่าความเหมาะสม เทียบกับผลรวมของค่าความเหมาะสมทั้งหมด ดังสมการที่ 4.1

$$pselect_i = \frac{F_i}{\sum F} \tag{4.1}$$

ซึ่งสามารถคำนวณค่าที่คาดหวังว่าจะสุ่มได้ (Expected Value :E) ของแต่ละโครโมโซมในแต่ละรุ่น ดังสมการที่ 4.2

$$E_i = pselect_i * popsize = \frac{F_i}{F} \tag{4.2}$$

สำหรับวิธีการสุ่มโครโมโซมต้นแบบของ SGA เป็นแบบจำลองการหมุนวงล้อถ่วงน้ำหนัก (Roullet Wheel : RW) ซึ่งกำหนดขนาดแต่ละช่องของวงล้อตามค่าความน่าจะเป็นที่จะสุ่มได้ในแต่ละครั้งของแต่ละโครโมโซม ซึ่งมีวิธีการดังนี้

i) หาค่าความเหมาะสมของแต่ละโครโมโซม

ii) หาค่าความน่าจะเป็นที่จะสุ่มได้ในแต่ละครั้งของแต่ละโครโมโซม

iii) หาความถี่สะสม (q) ของค่าความน่าจะเป็นของแต่ละโครโมโซมดังสมการที่ 4.3

$$q_i = \sum_{j=1}^{i} pselect_j \tag{4.3}$$

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

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

ตัวอย่างการกำหนดความน่าจะเป็น โดยกำหนดจากค่าความเหมาะสม เทียบกับผลรวมของค่าความเหมาะสมทั้งหมด จะเป็นได้ว่าในการคัดเลือกโครโมโซมต้นแบบจาก 4 โครโมโซมนี้ โอกาสที่จะได้โครโมโซมลำดับที่ 1 ต่อการสุ่มแต่ละครั้งเท่ากับ 0.157 และโอกาสที่จะสุ่มได้โครโมโซมลำดับที่ 2, 3, 4 ต่อการสุ่มแต่ละครั้งเท่ากับ 0.502, 0.051 และ 0.290 ตามลำดับ และจำนวนโครโมโซมต้นแบบที่สุ่มได้โดยจำลองการหมุนวงล้อ ดังนี้

ตารางที่ 4.1 ตัวอย่างการคัดเลือกโครโมโซมต้นแบบ

ตัวอย่าง :

ลำดับ	โครโมโซม	x	ค่าความเหมาะสม (F)	ค่าความน่าจะเป็น ($pselect_i$)	จำนวนที่คาดหวัง (E_i)	จำนวนที่สุ่มได้จาก RW
1	0 1 1 1 0	14	196	0.157	0.628	1
2	1 1 0 0 1	25	625	0.502	2.008	2
3	0 1 0 0 0	8	64	0.051	0.204	0
4	1 0 0 1 1	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 4.2 โครโมโซมต้นแบบใน Mating pool

ลำดับโครโมโซม	1	2	3	4
ค่าความเหมาะสม (F)	196	625	64	361
ค่าความน่าจะเป็นที่สุ่มได้แต่ละครั้ง ($pselect_i$)	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

ซึ่งจำนวนที่สุ่มได้เป็นโครโมโซมต้นแบบใน Mating pool ของแต่ละโครโมโซมเป็น 1, 2, 0 และ 1 ตามลำดับ จะเห็นได้ว่าโครโมโซมลำดับที่ 2 มีค่าความเหมาะสมสูงที่สุดจะมีโอกาสถูกคัดเลือกในจำนวนที่มากที่สุด ส่วนโครโมโซมลำดับที่ 3 มีค่าความเหมาะสมต่ำมากจึงมีโอกาสที่จะไม่ถูกเลือกเลย

4.3.4 ดำเนินการทางพันธุศาสตร์ เป็นขั้นตอนที่จำลองแบบธรรมชาติทางพันธุกรรม ซึ่งตัวดำเนินการทางพันธุศาสตร์ของ SGA คือ ครอสโอเวอร์และมิวเตชัน ซึ่งมีรายละเอียดดังนี้

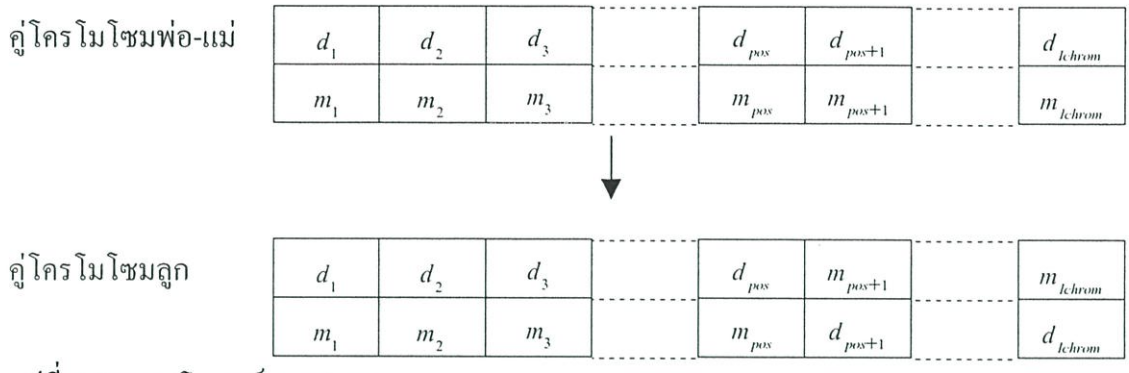
i) ครอสโอเวอร์ เป็นตัวดำเนินการในการแลกเปลี่ยนส่วนโครโมโซมพ่อ-แม่ ตามการกำหนดอัตราความน่าจะเป็นของการครอสโอเวอร์ (Probability of Crossover : Pc) เพื่อสร้างชุดโครโมโซมรุ่นใหม่คือโครโมโซมลูก มีขั้นตอนการทำงานคือ

ขั้นตอนแรก : สุ่มจับคู่โครโมโซมพ่อ-แม่ใน mating pool ที่สร้างไว้จากการคัดเลือก

ขั้นตอนที่สอง : สร้างเลขสุ่มจำนวนจริง (r) มีค่าอยู่ในอยู่ในช่วง [0.0 , 1.0] โดยถ้า $r \leq Pc$ แล้ว โครโมโซมพ่อ-แม่ นั้นจึงมีการครอสโอเวอร์

ขั้นตอนที่สาม : ครอสโอเวอร์โดยการแลกเปลี่ยนส่วนของโครโมโซมพ่อ-แม่นั้น ซึ่งการครอสโอเวอร์ของ SGA เป็นการครอสโอเวอร์แบบ 1 จุด (One-point Crossover) แสดงดังรูปที่ 4.6 ดังนี้

- สุ่มเลือกตำแหน่งที่ใช้ดำเนินการทางพันธุศาสตร์ (pos) เป็นตำแหน่งที่ครอสโอเวอร์ ซึ่ง pos มีค่าอยู่ในช่วง [1, Ichrom-1]
- แลกเปลี่ยนค่าในแต่ละบิตของคู่โครโมโซมพ่อ-แม่ตั้งแต่ตำแหน่งที่ post+1 ถึง Ichrom ซึ่งจะทำให้เกิดโครโมโซมลูกใหม่ 2 โครโมโซม



รูปที่ 4.6 ครอสโอเวอร์แบบ 1 จุด

จำนวนการครอสโอเวอร์ในแต่ละรุ่นดำเนินการขึ้นอยู่กับการกำหนดค่า Pc ซึ่งแตกต่างกันในแต่ละปัญหา เช่น ถ้าจำนวนประชากรแต่ละรุ่น Popsize เท่ากับ 30 โครโมโซม และกำหนดให้ Pc=0.6 แล้วจำนวนการ ครอสโอเวอร์ในแต่ละรุ่นเท่ากับ Pc*(popsize/2)=0.6*(30/2)=9 ครั้ง (การครอสโอเวอร์ 1 ครั้ง เกิดจากโครโมโซม 2 โครโมโซม)

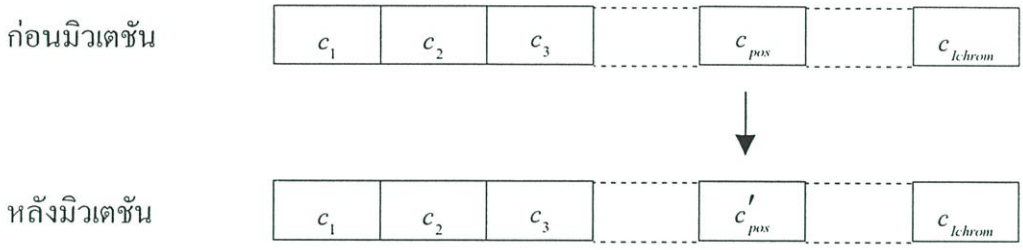
จากการสุ่มจับคู่โครโมโซมพ่อ-แม่ใน mating pool ได้โครโมโซมลำดับที่ 1 คู่ลำดับที่ 2 และลำดับที่ 2 คู่กับลำดับที่ 4 แต่เฉพาะโครโมโซมคู่แรกจะเกิดการครอสโอเวอร์ เนื่องจากเลขสุ่ม $t \leq 0.5$ ตามอัตราการครอสโอเวอร์ที่กำหนด โดยตำแหน่งในการครอสโอเวอร์ที่สุ่มได้คือ pos = 2 จะเห็นได้ว่าโครโมโซมลูกลำดับที่ 2 ที่เกิดขึ้นหลังจากครอสโอเวอร์มีค่าความเหมาะสมดีขึ้นกว่าโครโมโซมพ่อ-แม่ทั้งหมดในรุ่นก่อนเป็น 900 ซึ่งแสดงให้เห็นถึงการจำลองแบบกระบวนการครอสโอเวอร์ตามธรรมชาติทางพันธุศาสตร์ของ SGA ช่วยสร้างคำตอบที่ดีขึ้น

ตัวอย่าง : กำหนด Pc=0.5 โครโมโซมพ่อ-แม่ใน mating pool จากการคัดเลือกครอสโอเวอร์ดังนี้

ตารางที่ 4.3 แสดงการครอสโอเวอร์ของโครโมโซมใน Mating pool

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

ii) มิวเตชัน เป็นตัวดำเนินการผ่าเหล่าตัวหนึ่งที่อาจช่วยให้โครโมโซม มีค่าความเหมาะสมดีขึ้นหลังจากการครอสโอเวอร์ โดยกลับค่าบิตเป็นค่าใหม่ในตำแหน่งบิตที่สุ่มได้ ตามอัตราความน่าจะเป็นของการมิวเตชันในแต่ละบิต (Probabilty of Mutation : Pm) ที่กำหนด สำหรับการมิวเตชันของ SGA นั้นเป็นแบบไบนารีมิวเตชัน (Binary Mutation) โดยกลับค่าบิตเป็นค่าคอมพลีเมนต์ คือ จาก 0 เป็น 1 หรือจาก 1 เป็น 0 ดังรูปที่ 4.7



รูปที่ 4.7 ไบนารีมิวเตชัน

จำนวนการมิวเตชันในแต่ละรุ่นขึ้นอยู่กับการกำหนดค่า Pm ซึ่งแตกต่างกันในแต่ละปัญหา เช่น ถ้าจำนวนประชากรแต่ละรุ่น popsize เท่ากับ 30 โครโมโซมประกอบด้วย 5 บิต และกำหนดให้ Pm=0.02 แล้ว จำนวนการมิวเตชั่นในแต่ละกลุ่มเท่ากับ Pm*popsize*Ichrom=0.02*30*5=3 บิต

ตัวอย่าง : กำหนด Pm=0.1 ดำเนินการมิวเตชันโครโมโซมลูกที่ได้จากการครอสโอเวอร์ดังนี้

ตารางที่ 4.4 การดำเนินการมิวเตชั่น

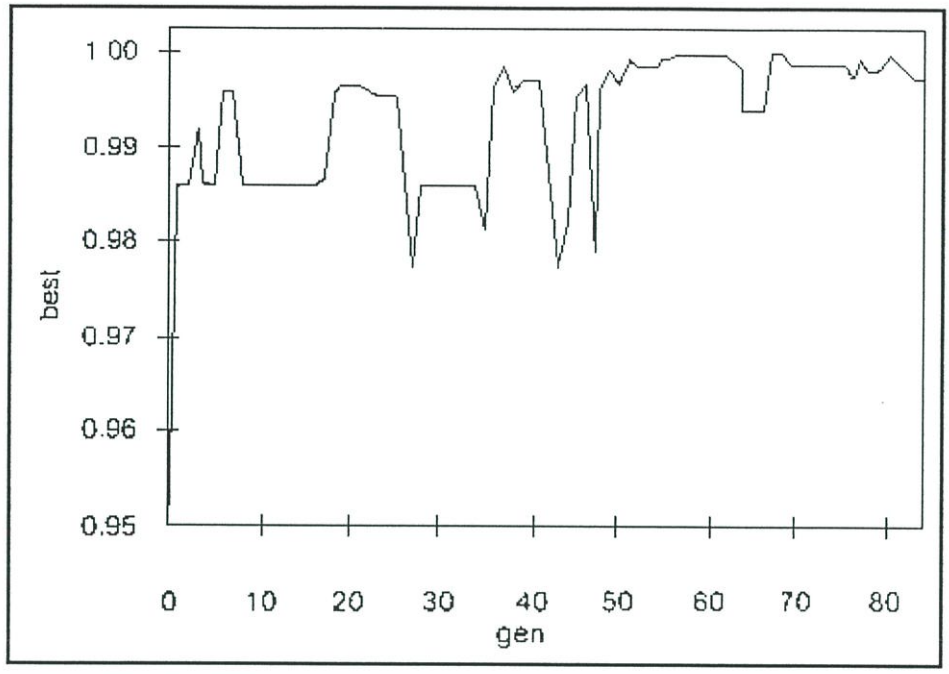
ลำดับ	ก่อนมิวเตชัน					เลขสุ่ม (r)					หลังมิวเตชัน					x	ค่าความเหมาะสม (F)
1	0	1	0	0	1	0.581	0.346	*0.062*	0.785	0.401	0	1	*1*	0	1	13	169
2	1	1	1	1	0	0.829	0.534	0.947	0.308	0.277	1	1	1	1	0	30	900
3	1	1	0	0	1	0.398	0.646	0.494	0.765	*0.029*	0	1	0	0	*0*	24	576
4	1	0	0	1	1	0.175	0.335	0.837	0.577	0.308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

จากการสุ่มตำแหน่งที่มิวเตชันโดยสร้างเลขสุ่ม r ของแต่ละตำแหน่งบิตในแต่ละโครโมโซมแล้ว ตำแหน่งบิตที่ 3 ของโครโมโซมลำดับที่ 1 และตำแหน่งที่ 4 ของโครโมโซมลำดับที่ 3 เป็นตำแหน่งที่ $r \leq 0.1$ ตามอัตราการที่กำหนดจึงเกิดมิวเตชั่น ทำให้โครโมโซมมีค่าความเหมาะสมจาก 81 และ 625 เป็น 169 และ 576 ตามลำดับ จะเห็นได้ว่ามิวเตชันเป็นตัวดำเนินการที่อาจทำให้โครโมโซมมีค่าความเหมาะสมสูงขึ้นหรือลดลงได้ แต่อย่างไรก็ตามเฉลี่ยของค่าความเหมาะสมดีขึ้นจาก 492 เป็น 502 แสดงถึงการหาคำตอบของ SGA โดยส่วนมากดีขึ้น และความสำคัญของการหาคำตอบของ GA นั้นเป็นความต้องการได้คำตอบโดยพิจารณาจากคำตอบที่ดีขึ้นเกิดขึ้น ซึ่งจะมีโอกาสอยู่รอดเพื่อถ่ายทอดส่วนที่ดีในรุ่นต่อไป

4.3.5 ประชากรรุ่นใหม่ เป็นชุดโครโมโซมลูกที่เกิดจากขั้นตอนของวิวัฒนาการต่าง ๆ ทั้งหมด ซึ่งประชากรรุ่นใหม่ทั้งหมดที่เกิดขึ้นจะถูกถ่ายทอดกลายเป็นประชากรรุ่นเก่าสำหรับวิวัฒนาการในรุ่นถัดไป ซึ่งเรียกวิวัฒนาการแบบนี้ว่า การถ่ายทอดแบบทั่วไปหรือรีโพดักชันแบบมั่วไป (General reproduction) กระบวนการต่าง ๆ จะถูกปฏิบัติซ้ำ ๆ จนกระทั่งถึงรุ่นที่มากที่สุด (Max generation : maxgen) ที่ต้องการ

4.4 การประยุกต์เจเนติก อัลกอริทึมแบบง่าย

เมื่อวิเคราะห์การทำงานของ SGA ซึ่งเป็น GA ในยุคแรก ๆ แล้วจะเห็นว่า วิธีการ GA เป็นการหาคำตอบแบบสุ่ม ซึ่งเป็นวิธีการที่ไม่มีการบันทึกหรือจดจำที่ดีที่สุดของรุ่นก่อน จึงทำให้การหาคำตอบของ SGA ได้คำตอบที่ดีมากขึ้นหรือน้อยลงได้ ดังรูปที่ 4.12 แสดงกราฟคำตอบค่าความเหมาะสมสูงสุดแต่ละรุ่นภายใน 30 รุ่นในการหาค่าสูงสุดของฟังก์ชัน $y = x^n$ ที่ $n = 10$ จะเห็นได้ว่าคำตอบความเหมาะสมสูงสุดของ SGA มีค่าสูงขึ้นและลดลงด้วยวิธีการทำงานที่อาศัยการสุ่ม ดังนั้นหากสามารถพัฒนาวิธีการค้นหาคำตอบของ GA ให้ดีขึ้นแล้วก็จะเป็นการปรับปรุงสมรรถนะของ GA ยิ่งขึ้น สำหรับหาค่าสูงสุดของฟังก์ชัน $y = x^n$ เราปรับปรุงการค้นหาคำตอบของ GA



รูปที่ 4.8 แสดงกราฟค่าความเหมาะสมสูงสุดของ SGA ในการหาค่าสูงสุดของฟังก์ชัน $y = x^n$ ที่ $n = 10$

4.4.1 รีโพรดักชันแบบรักษาค่าความเหมาะสมที่ดี เนื่องจากในการค้นหาคำตอบของ SGA นั้น มีโอกาสที่จะสูญเสียโครโมโซมในรุ่นเก่า ที่มีความเหมาะสมที่ดีไปได้ ซึ่งจะทำให้คำตอบในรุ่นถัดไปนั้นดีมากขึ้นหรือน้อยลง ดังนั้นหากปรับปรุง SGA ให้ควบคุมการค้นหาคำตอบ โดยรักษาโครโมโซมที่ดีไว้แล้ว จะช่วยให้วิวัฒนาการคำตอบในรุ่นถัดไปดีขึ้นเรื่อยๆ โดยมีวิธีการดังนี้

- กำหนดจำนวนโครโมโซมที่ดีที่สุด (#best) ของรุ่นเก่าที่ต้องรักษา เป็น 1, 2, 4, ...
- ถ้าจำนวนโครโมโซมที่กำหนดเป็น 1 ให้สร้างชุดโครโมโซมรุ่นใหม่ทั้งหมด แล้วจึงคัดลอก (copy) โครโมโซมที่ดีที่สุดของรุ่นเก่า มาแทนที่โครโมโซมรุ่นใหม่ที่มีค่าความเหมาะสมน้อยที่สุด
- ถ้าจำนวนโครโมโซมที่กำหนดเป็น 2, 4, ... ให้คัดลอกโครโมโซมที่ดีทีสุดในรุ่นเก่า ตามจำนวนที่กำหนดมาเป็นโครโมโซมรุ่นใหม่ แล้วจึงสร้างโครโมโซมรุ่นใหม่ส่วนที่เหลือต่อไป

ดังรูปที่ 4.9(a) แสดงวิธีการรีโพรดักชันแบบรักษาค่าความเหมาะสมที่ดี 2 โครโมโซม

ลำดับ	โครโมโซมรุ่นเก่า	ค่าความเหมาะสม	จำนวนที่สุ่มได้						โครโมโซมรุ่นใหม่	ค่าความเหมาะสม
				1.) สร้างโครโมโซมรุ่นใหม่ทั้งหมด						
				ครอสโอเวอร์			มิวเตชัน	ค่าความเหมาะสม		
1	01110	196	1	01110	pos 2	01001	pos	01001	81	
2	11001	625	2	11001		11110	2,4	10100	400	10100 400
3	01000	64	0	10011	1	11001	2	10001	289	10001 289
4	10011	361	1	11001		10011		10011	361	10011 361
				2.) คัดลอกโครโมโซมรุ่นเก่าลำดับที่ 2						**11001 625**

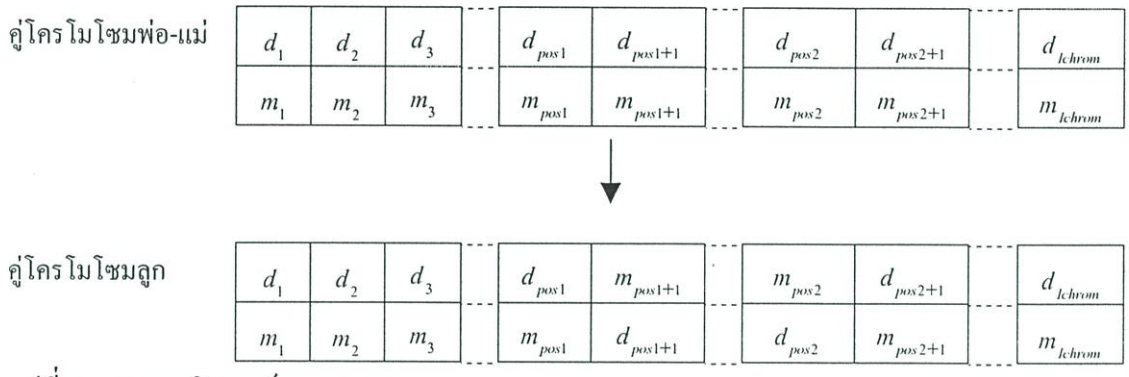
(a) รักษาค่าความเหมาะสมที่ดี 1 โครโมโซม

ลำดับ	โครโมโซมรุ่นเก่า	ค่าความเหมาะสม	จำนวนที่สุ่มได้						โครโมโซมรุ่นใหม่	ค่าความเหมาะสม
				1.) คัดลอกโครโมโซมรุ่นเก่าลำดับที่ 2 และ 4					**11001**	**625**
				2.) สร้างโครโมโซมรุ่นใหม่ทั้งหมด					**10011**	**361**
1	01110	196	1	ครอสโอเวอร์			มิวเตชัน	ค่าความเหมาะสม		
2	11001	625	2		pos		pos			
3	01000	64	0	01110	2	01001		01001	81	01001 81
4	10011	361	1	11001		11110	2,4	10001	400	10100 400

(b) รักษาค่าความเหมาะสมที่ดี 2 โครโมโซม

รูปที่ 4.9 รีโพรดักชันแบบรักษาค่าความเหมาะสมที่ดี

4.4.2 ครอสโอเวอร์แบบ 2 จุด การแลกเปลี่ยนส่วนโครโมโซมพ่อ-แม่นั้น บางครั้งหากแลกเปลี่ยนค่าบิตเพียงบางช่วงของโครโมโซมแล้วจะสร้างโครโมโซมที่ดีกว่า เช่น การหาค่าสูงสุดของฟังก์ชัน $y = x^2$ ของคู่โครโมโซมพ่อ-แม่ 01110 และ 11001 ซึ่งมีค่าความเหมาะสมเป็น 196 และ 625 หากแลกเปลี่ยนค่าบิตตำแหน่งที่ 3 และ 4 เท่านั้นจะทำให้เกิดโครโมโซมลูกคือ 01*00*0 และ 11*11*1 มีค่าความเหมาะสมเป็น 64 และ 961 ซึ่งโครโมโซม 11111 เป็นโครโมโซมที่ให้คำตอบที่ดีที่สุดที่ต้องการ ดังนั้นการประยุกต์ SGA โดยพัฒนาตัวดำเนินการครอสโอเวอร์ให้เป็นแบบ 2 จุด (Two-point crossover) จะทำให้ SGA ค้นหาคำตอบที่ดีขึ้นได้ ดังรูปที่ 4.10 มีวิธีการดังนี้

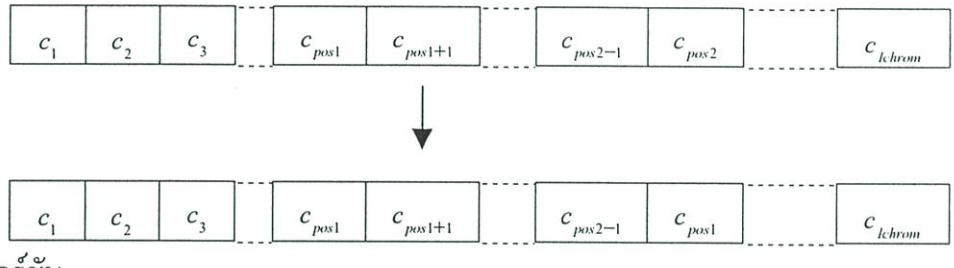


รูปที่ 4.10 ครอสโอเวอร์แบบ 2 จุด

- สุ่มเลือกตำแหน่ง pos_1, pos_2 คือตำแหน่งเริ่มต้นและตำแหน่งสุดท้ายที่จะครอสโอเวอร์ตามลำดับ ซึ่ง pos_1 มีค่าอยู่ในช่วง [1, Ichrom-1] และ pos_2 มีค่าอยู่ในช่วง [1, Ichrom] โดยที่ pos_1 มีค่าน้อยกว่า pos_2
- แลกเปลี่ยนค่าในแต่ละบิตของคู่โครโมโซมพ่อ-แม่ ตั้งแต่ตำแหน่งที่ $pos_1 + 1$ ถึง pos_2

4.4.3 ไบนารีมิวเตชันแบบกำหนดบิต เนื่องจากการหาคำตอบของ SGA นั้น กระบวนการไบนารีมิวเตชัน อาจทำให้โครโมโซมที่เปลี่ยนแปลงไปให้คำตอบที่ดีลดลง และ ทำให้สูญเสียโครโมโซมที่ดีไป เช่น โครโมโซม 11110 มีค่าความเหมาะสมเป็น 900 หากสุ่มได้บิตตำแหน่งที่ 1 เกิดมิวเตชันแล้ว โครโมโซมที่เกิดขึ้นจากการมิวเตชันคือ *0*1110 ทำให้มีค่าความเหมาะสมลดลงเป็น 196 แต่ในบางครั้งข้อดีหรือจุดเด่นของปัญหาจะสามารถนำมาปรับให้เข้ากับการค้นหาคำตอบที่ดีขึ้นได้ สำหรับในการหาค่าสูงสุดของฟังก์ชัน $y = x^n$ นี้ค่าบิตของโครโมโซมที่เป็น 1 จะทำให้ค่าความเหมาะสมสูงขึ้นเสมอ ดังนั้นหากปรับปรุงไบนารีมิวเตชันให้เป็นแบบกำหนดค่าแน่นอนให้กับบิตที่เกิดมิวเตชัน โดยกำหนดให้บิตที่เกิดมิวเตชันมีค่าบิตเป็น 1 เสมอจะช่วยปรับแนวทางการค้นหาคำตอบของ SGA ดีขึ้น เช่น หากประยุกต์ไบนารีมิวเตชันที่กำหนดค่าบิตให้เป็น 1 เสมอ กับโครโมโซม 11110 ในตำแหน่งที่ 1 แล้วโครโมโซมทีเกิดขึ้นจะเหมือนเดิม และยังเป็นการรักษาโครโมโซมที่มีค่าความเหมาะสมที่ดีไว้ด้วย

4.4.4 อินเวอร์ชัน (Inversion) เป็นตัวดำเนินการที่ประยุกต์เพิ่มเติมใน SGA โดยจำลองแบบลักษณะของการอินเวอร์ชันในทางพันธุศาสตร์ที่เป็นลักษณะของการกลับหัวกลับหางส่วนยีนส์ภายในโครโมโซม ที่อาจช่วยให้เกิดโครโมโซมที่ดีขึ้นได้ โดยการกลับส่วนค่าบิตในช่วงตำแหน่งของโครโมโซมที่สุ่มได้ตามอัตราค่าความน่าจะเป็นของการอินเวอร์ชันแต่ละโครโมโซมที่สุ่มได้ (Probability of Inversion : Pi) ที่กำหนด ดังรูปที่ 4.11 มีขั้นตอนดังนี้



รูปที่ 4.11 อินเวอร์ชัน

- สุ่มเลือกตำแหน่ง pos_1, pos_2 คือตำแหน่งเริ่มต้นและตำแหน่งสุดท้ายที่จะอินเวอร์ชันตามลำดับ ซึ่ง pos_1 และ pos_2 มีค่าอยู่ในช่วง [1, Ichrom] 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,....

เช่น สุ่มโครโมโซมที่จะอินเวอร์ชันคือ 01010 มีค่าความเหมาะสมเป็น 100 โดยสุ่มตำแหน่ง $pos_1=1$ และ $pos_2=4$ แล้ว จะเห็นว่าการอินเวอร์ชันในแต่ละรุ่นขึ้นอยู่กับการกำหนดค่า Pi ซึ่งแตกต่างกันในแต่ละปัญหา เช่น ถ้าจำนวนประชากรแต่ละรุ่น popsize เท่ากับ 30 โครโมโซม และกำหนดให้ Pi=0.1 แล้วจำนวนการอินเวอร์ชันในแต่ละรุ่นเท่ากับ Pi*popsize=0.1*30=3 ครั้ง

ผลการทำงานโดยสรุปของการประยุกต์ SGA ในการหาค่าสูงสุดของฟังก์ชัน $y=x^n$ ที่ n=10 ภายใน 30 รุ่น ซึ่งกำหนดค่า SGA พารามิเตอร์ต่าง ๆ เปรียบเทียบดังตารางที่ 4.2 แสดงดังรูปที่ 4.12 (a) ถึง 4.12 (g) โดยเปรียบเทียบการหาคำตอบที่มีกำหนดให้มีการมิวเตชันในรูปที่ 4.12 (b) นั้นดีกว่าการหาคำตอบเพียงครอสโอเวอร์เม่านั้นในรูปที่ 4.12 (a) และจะเห็นได้ว่า SGA สามารถหาคำตอบได้ดีขึ้นเมื่อมีการปรับปรุงวิธีการต่าง ๆ คือ หากรีโพรดักชันแบบรักษาค่าที่ดีจำนวน 4 โครโมโซม และครอสโอเวอร์แบบ 2 ตำแหน่งในรูปที่ 4.12 (e) แล้ว ผลการทำงานของ SGA สามารถหาคำตอบที่ดีที่สุดเท่ากับ 1.0 ได้ ซึ่งดีกว่าในรูปที่ 4.12 (d) เมื่อรีโพรดักชันแบบทั่วไปในรูปที่ 4.12 (a), 4.12 (b), 4.12 (c) และ 4.12 (d) ที่ยังไม่สามารถหาคำตอบที่ดีที่สุดได้ภายในการดำเนินการ 30 รุ่น สำหรับในรูปที่ 4.12 (f) นั้นเมื่อเพิ่มตัวดำเนินการอินเวอร์ชันแล้ว ผลการดำเนินงานนั้นสามารถหาคำตอบที่ดีที่สุดเท่ากับ 1.0 ได้เร็วขึ้นในรุ่นที่ 20 และเมื่อเพิ่มเทคนิคการกำหนดค่าบิตของการมิวเตชันให้เป็น 1 เสมอแล้ว ก็ยิ่งทำให้ SGA ทำงานได้ดียิ่งขึ้น โดยสามารถหาคำตอบที่ดีที่สุดได้เมื่อดำเนินงานผ่านไปเพียง 9 รุ่นเท่านั้น ดังรูปที่ 4.12 (g)

ตารางที่ 4.5 การกำหนด SGA พารามิเตอร์เปรียบเทียบผลการทำงานในการหาค่าสูงสุดของฟังก์ชัน $y = x^n$ ที่ n=10

รูปที่	Pc	Pm	Pi	จำนวนการรักษาโครโมโซมรุ่นเก่า (#best)	จำนวนตำแหน่งในการครอสโอเวอร์ (pt.cross)	การกำหนดค่าบิตมิวเตชัน (fix value)
4.12 (a)	0.8000	0.0000	0.0000	0	1	0->1 และ 1->0
4.12 (b)	0.8000	0.0333	0.0000	0	1	0->1 และ 1->0
4.12 (c)	0.8000	0.0333	0.0000	0	2	0->1 และ 1->0
4.12 (d)	0.8000	0.0333	0.0000	4	2	0->1 และ 1->0
4.12 (e)	0.8000	0.0333	0.0000	1	2	0->1 และ 1->0
4.12 (f)	0.8000	0.0333	0.3000	4	2	0->1 และ 1->0
4.12 (g)	0.8000	0.0333	0.3000	4	2	0, 1->1

Pop.size = 30 Pc = 0.8000 # best = 0
Chrom.length = 30 Pm = 0.0000 pt.crs = 1
Madx.gen = 30 Pi = 0.0000 fix val. = 2

generation	max	avg	min
0	0.9481	0.103	0.0000
1	0.9481	0.5498	0.1166
2	0.9481	0.7032	0.1766
3	0.9481	0.7352	0.3109
4	0.9481	0.8152	0.4474
5	0.9481	0.8819	0.4474
6	0.9481	0.9153	0.7062
7	0.9481	0.9320	0.7062
8	0.9481	0.9401	0.7062
9	0.9481	0.9401	0.9481
10	0.9481	0.9481	0.9481
11	0.9481	0.9481	0.9481
12	0.9481	0.9481	0.9481
13	0.9481	0.9481	0.9481
14	0.9481	0.9481	0.9481
15	0.9481	0.9481	0.9481
16	0.9481	0.9481	0.9481
17	0.9481	0.9481	0.9481
18	0.9481	0.9481	0.9481
19	0.9481	0.9481	0.9481
20	0.9481	0.9481	0.9481
21	0.9481	0.9481	0.9481
22	0.9481	0.9481	0.9481
23	0.9481	0.9481	0.9481
24	0.9481	0.9481	0.9481
25	0.9481	0.9481	0.9481
26	0.9481	0.9481	0.9481
27	0.9481	0.9481	0.9481
28	0.9481	0.9481	0.9481
29	0.9481	0.9481	0.9481
30	0.9481	0.9481	0.9481

(a)

Pop.size = 30 Pc = 0.8000 # best = 0
Chrom.length = 30 Pm = 0.0333 pt.crs = 1
Madx.gen = 30 Pi = 0.0000 fix val. = 2

generation	max	avg	min
0	0.9481	0.1030	0.0000
1	0.9860	0.5199	0.0115
2	0.9860	0.5905	0.0150
3	0.9921	0.5721	0.1563
4	0.9860	0.6123	0.0009
5	0.9860	0.7462	0.0001
6	0.9957	0.6839	0.0007
7	0.9957	0.6596	0.0007
8	0.9860	0.5824	0.0000
9	0.9859	0.6774	0.0553
10	0.9859	0.6368	0.0000
11	0.9859	0.7170	0.0007
12	0.9859	0.8187	0.3933
13	0.9859	0.7007	0.0006
14	0.9859	0.6491	0.0007
15	0.9860	0.8273	0.0455
16	0.9860	0.8119	0.0009
17	0.9865	0.8549	0.0205
18	0.9957	0.8328	0.0005
19	0.9963	0.8734	0.2589
20	0.9963	0.8912	0.2354
21	0.9963	0.7875	0.0089
22	0.9957	0.7705	0.0002
23	0.9954	0.8397	0.0009
24	0.9954	0.7015	0.0002
25	0.9953	0.7313	0.0009
26	0.9883	0.7610	0.0000
27	0.9771	0.7299	0.0000
28	0.9860	0.7715	0.0001
29	0.9859	0.8475	0.2015
30	0.9859	0.7080	0.0005

(b)

Pop.size = 30 Pc = 0.8000 # best = 0
Chrom.length = 30 Pm = 0.0333 pt.crs = 2
Madx.gen = 30 Pi = 0.0000 fix val. = 2

generation	max	avg	min
0	0.9481	0.1030	0.0000
1	0.9575	0.5559	0.0005
2	0.9957	0.5741	0.0009
3	0.9969	0.7093	0.0358
4	0.9960	0.7513	0.2475
5	0.9971	0.7064	0.0008
6	0.9971	0.8505	0.0330
7	0.9770	0.7278	0.0001
8	0.9911	0.7542	0.0002
9	0.9911	0.7146	0.0007
10	0.9952	0.7890	0.0009
11	0.9960	0.7329	0.0000
12	0.9956	0.8780	0.3621
13	0.9956	0.7700	0.0000
14	0.9956	0.7285	0.0501
15	0.9958	0.7667	0.0009
16	0.9958	0.7655	0.0003
17	0.9958	0.7429	0.0006
18	0.9957	0.7776	0.0007
19	0.9982	0.7790	0.0009
20	0.9982	0.7279	0.0009
21	0.9982	0.7525	0.0432
22	0.9982	0.7452	0.0009
23	0.9982	0.7096	0.0002
24	0.9976	0.8281	0.2432
25	0.9861	0.6068	0.0000
26	0.9923	0.8072	0.0009
27	0.9887	0.7854	0.0006
28	0.9958	0.7924	0.0005
29	0.9959	0.8595	0.4295
30	0.9959	0.7752	0.0008

(c)

Pop.size = 30 Pc = 0.8000 # best = 1
Chrom.length = 30 Pm = 0.0333 pt.crs = 2
Madx.gen = 30 Pi = 0.0000 fix val. = 2

generation	max	avg	min
0	0.9481	0.1030	0.0000
1	0.9575	0.5875	0.0238
2	0.9957	0.606	0.0194
3	0.9963	0.7498	0.0524
4	0.9969	0.7991	0.2620
5	0.9969	0.7512	0.0512
6	0.9969	0.896	0.4962
7	0.9969	0.7753	0.0005
8	0.998	0.8075	0.0450
9	0.998	0.7623	0.0050
10	0.9981	0.8387	0.0308
11	0.9981	0.7796	0.0005
12	0.9981	0.9174	0.4760
13	0.9981	0.8194	0.2426
14	0.9987	0.7726	0.0554
15	0.9987	0.823	0.0219
16	0.9987	0.8139	0.2426
17	0.9987	0.7908	0.0009
18	0.9987	0.8296	0.2605
19	0.9987	0.8628	0.2441
20	0.9987	0.7987	0.0367
21	0.9987	0.8101	0.0526
22	0.9987	0.8145	0.0425
23	0.9988	0.7884	0.0010
24	0.9938	0.9105	0.4667
25	0.9988	0.6898	0.0005
26	0.9988	0.8623	0.2399
27	0.9988	0.8495	0.0476
28	0.9988	0.853	0.0007
29	0.9999	0.8977	0.6664
30	0.9999	0.8486	0.0009

(d)

รูปที่ 4.12 ผลการทำงานโดยสรุปของ GA ในการหาค่าสูงสุดของฟังก์ชัน $y=x^n$ ที่ n=10

Pop.size = 30 Pc = 0.8000 # best = 4
Chrom.length = 30 Pm = 0.0333 pt.crs = 2
Max.gen = 30 Pi = 0.000 fix val. = 2

generation	max	avg	min
0	0.9481	0.1030	0.0000
1	0.9575	0.5857	0.0005
2	0.9957	0.6655	0.0009
3	0.9963	0.7556	0.0524
4	0.9963	0.7705	0.0524
5	0.9963	0.8069	0.0524
6	0.9975	0.7939	0.0009
7	0.9975	0.8738	0.0454
8	0.9975	0.8026	0.0001
9	0.9987	0.8697	0.2170
10	0.9987	0.7446	0.0002
11	0.9997	0.8911	0.0347
12	0.9997	0.8241	0.0005
13	0.9997	0.8020	0.0000
14	0.9998	0.9362	0.2589
15	0.9998	0.8406	0.0000
16	0.9998	0.7955	0.0546
17	0.9998	0.8405	0.0010
18	0.9998	0.9286	0.2590
19	0.9998	0.7849	0.0007
20	0.9998	0.8373	0.0007
21	0.9998	0.8608	0.0008
22	0.9998	0.8767	0.0010
23	0.9998	0.7984	0.0010
24	0.9998	0.8458	0.0456
25	0.9998	0.8151	0.0010
26	0.9998	0.8314	0.0020
27	0.9998	0.8100	0.0527
28	0.9998	0.9182	0.0475
29	1.0000	0.6918	0.0000
30	1.0000	0.8837	0.0010

(e)

Pop.size = 30 Pc = 0.8000 # best = 4
Chrom.length = 30 Pm = 0.0333 pt.crs = 2
Max.gen = 30 Pi = 0.000 fix val. = 2

generation	max	avg	min
0	0.9481	0.1030	0.0000
1	0.9648	0.5539	0.0001
2	0.9648	0.6905	0.0004
3	0.9803	0.7530	0.0009
4	0.9803	0.7691	0.0224
5	0.9900	0.7140	0.0004
6	0.9900	0.7182	0.0042
7	0.9900	0.7613	0.0350
8	0.9997	0.7830	0.0009
9	0.9997	0.7269	0.0009
10	0.9998	0.8061	0.0352
11	0.9998	0.7000	0.0181
12	0.9998	0.7294	0.0455
13	0.9998	0.8265	0.0563
14	0.9998	0.8679	0.0009
15	0.9998	0.8829	0.0236
16	0.9998	0.8733	0.0006
17	0.9998	0.7911	0.0010
18	0.9998	0.8801	0.0009
19	0.9998	0.8065	0.0009
20	1.0000	0.8335	0.0009
21	1.0000	0.8084	0.0006
22	1.0000	0.8212	0.0009
23	1.0000	0.8201	0.0008
24	1.0000	0.8541	0.1786
25	1.0000	0.8065	0.0202
26	1.0000	0.8479	0.0559
27	1.0000	0.8045	0.0008
28	1.0000	0.7436	0.0003
29	1.0000	0.7853	0.0028
30	1.0000	0.8771	0.0008

(f)

Pop.size = 30 Pc = 0.8000 # best = 4
Chrom.length = 30 Pm = 0.0333 pt.crs = 2
Max.gen = 30 Pi = 0.000 fix val. = 2

generation	max	avg	min
0	0.9481	0.1030	0.0000
1	0.9481	0.5959	0.0156
2	0.9860	0.8297	0.0862
3	0.9860	0.9738	0.1108
4	0.9860	0.9068	0.6612
5	0.9951	0.9356	0.7049
6	0.9951	0.9670	0.7062
7	0.9956	0.9723	0.7062
8	0.9997	0.9875	0.9481
9	1.0000	0.9818	0.8139
10	1.0000	0.9484	0.0427
11	1.0000	0.9837	0.9521
12	1.0000	0.9572	0.5236
13	1.0000	0.9582	0.5236
14	1.0000	0.9702	0.5496
15	1.0000	0.9574	0.5496
16	1.0000	0.9374	0.5496
17	1.0000	0.9078	0.5245
18	1.0000	0.9413	0.3657
19	1.0000	0.9774	0.4798
20	1.0000	0.9799	0.4798
21	1.0000	0.9884	0.6714
22	1.0000	0.9595	0.5245
23	1.0000	0.9621	0.5030
24	1.0000	0.9678	0.6714
25	1.0000	0.9794	0.8543
26	1.0000	0.9754	0.8543
27	1.0000	0.9801	0.8543
28	1.0000	0.9796	0.8543
29	1.0000	0.9817	0.8543
30	1.0000	0.9625	0.0456

(g)

รูปที่ 4.12 (ต่อ) ผลการทำงานโดยสรุปของ GA ในการหาค่าสูงสุดของฟังก์ชัน $y=x^n$ ที่ n=10

4.5 การแก้ปัญหาโดยเจเนติกอัลกอริทึม

ในหัวข้อนี้จะนำเสนอตัวอย่างในการใช้ GA เข้ามาแก้ปัญหาต่าง ๆ ซึ่งสิ่งที่จำเป็นอย่างยิ่งในการปรับรูปแบบของปัญหาและขั้นตอนต่าง ๆ ให้เหมาะสมกับการทำงานของ GA โดยอาจใช้การประยุกต์เจเนติกแบบง่าย ในที่กล่าวมาแล้วในบทนี้เข้ามาช่วยหรืออาจเป็นการเพิ่มขั้นตอนหรือเงื่อนไขต่าง ๆ ของ GA

เนื่องจาก GA นั้นทำงานโดยอาศัยหลักการสุ่ม โดยการสุ่มครั้งแรกก็คือการสุ่มเลือกโครโมโซมต้นแบบจากโครโมโซมทั้งหมดที่เป็นไปได้ ดังนั้นในบางกรณีอาจจะสุ่มได้กรณีที่ดี ซึ่งจะส่งผลให้สามารถหาคำตอบที่ดีของปัญหานั้นได้ภายในการทำงานของ GA เพียงไม่กี่วัฏจักร แต่ในกรณีที่โครโมโซมต้นแบบมีค่าความเหมาะสมต่ำก็ทำให้การทำงานของ GA ช้าลง ดังนั้นอาจจะใช้การกำหนดค่าของโครโมโซมต้นแบบขึ้นมาเอง เพื่อลดปัญหาในการที่ได้โครโมโซมต้นแบบที่ดีไม่ดี ดังเช่นปัญหาดังต่อไปนี้

4.5.1 ปัญหาการหาค่าสูงสุดของฟังก์ชัน $f(x) = x^2$

ในปัญหานี้ได้มีการใช้ GA เข้ามาช่วยในการหาค่าของ x ที่ทำให้ f (x) มีค่าสูงสุด โดยได้แบ่งการทดสอบเป็น 2 ชุด โดยชุดแรกได้ใช้การสุ่มในการเลือกโครโมโซมต้นแบบ ส่วนในชุดที่ 2 นั้นได้เลือกโครโมโซมต้นแบบ โดยการเลือกมาจากทุก ๆ ช่วงของค่าคำตอบ

1. กำหนดฟังก์ชันความเหมาะสม

   ฟังก์ชันเป้าหมาย คือ $f = x^2$

   ฟังก์ชันความเหมาะสม คือ $F = x^2$

   ซึ่งคำตอบที่ดีที่สุด คือ x ที่มีค่าความเหมาะสมสูงสุด (MAX (F))

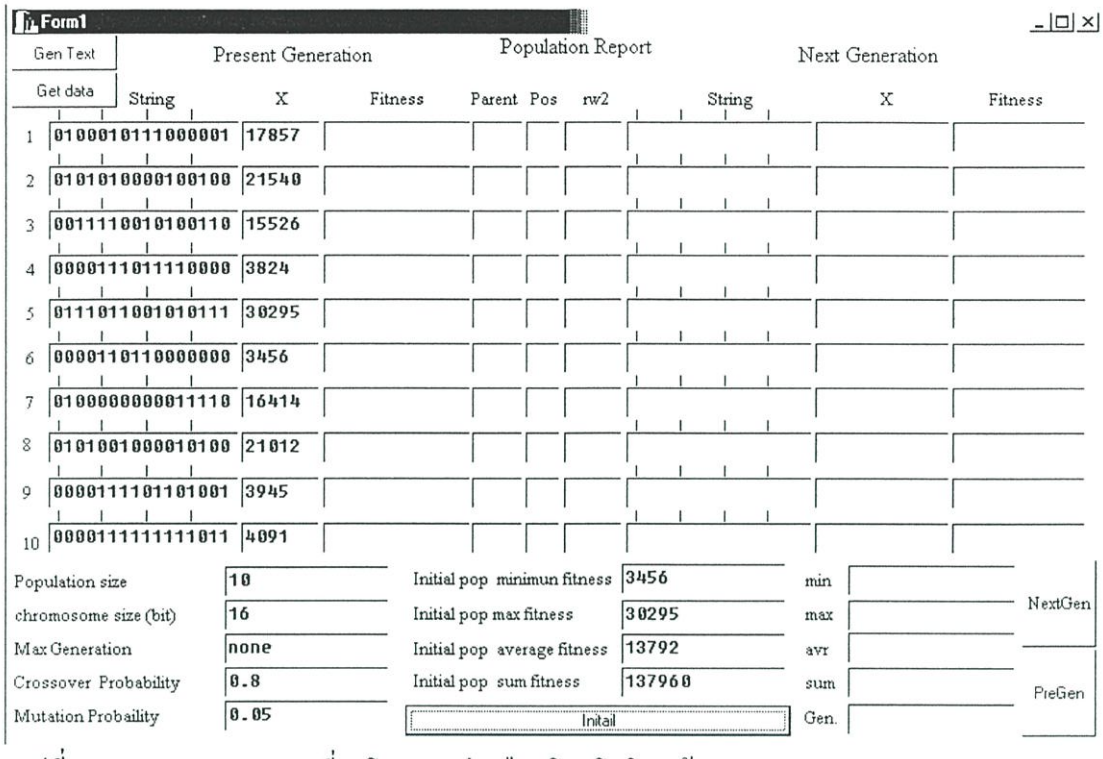
2. การกำหนดรูปแบบโครโมโซมเป็น กำหนดรูปแบบโครโมโซมเป็นแบบไบนารี 16 บิต ซึ่งมีค่าต่ำสุด คือ 0 และค่าสูงสุดคือ 65535

B_1	B_2	B_3	...	B_{16}

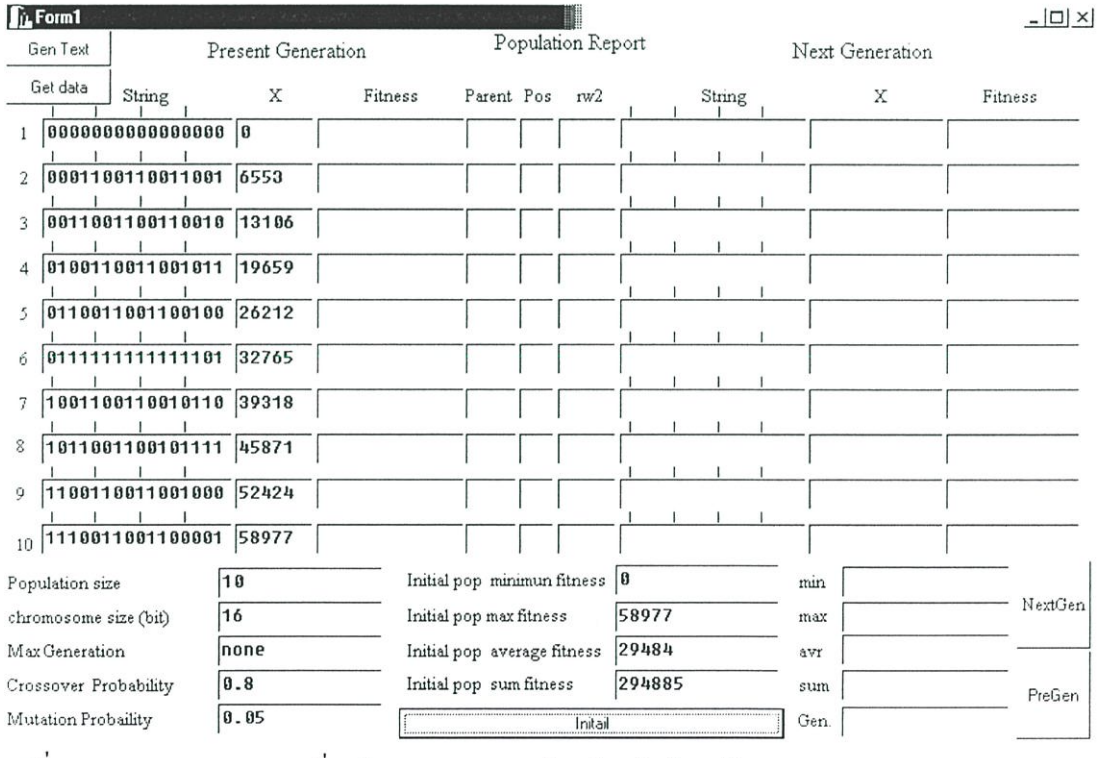
   $B_i \in [0,1]$ โดยที่ B_{16} นั้นคือค่าบิตต่ำสุด หรือ LSB ส่วน B_1 นั้น คือ ค่าบิตสูงสุด หรือ MSB

3. การดำเนินการทางพันธุกรรม

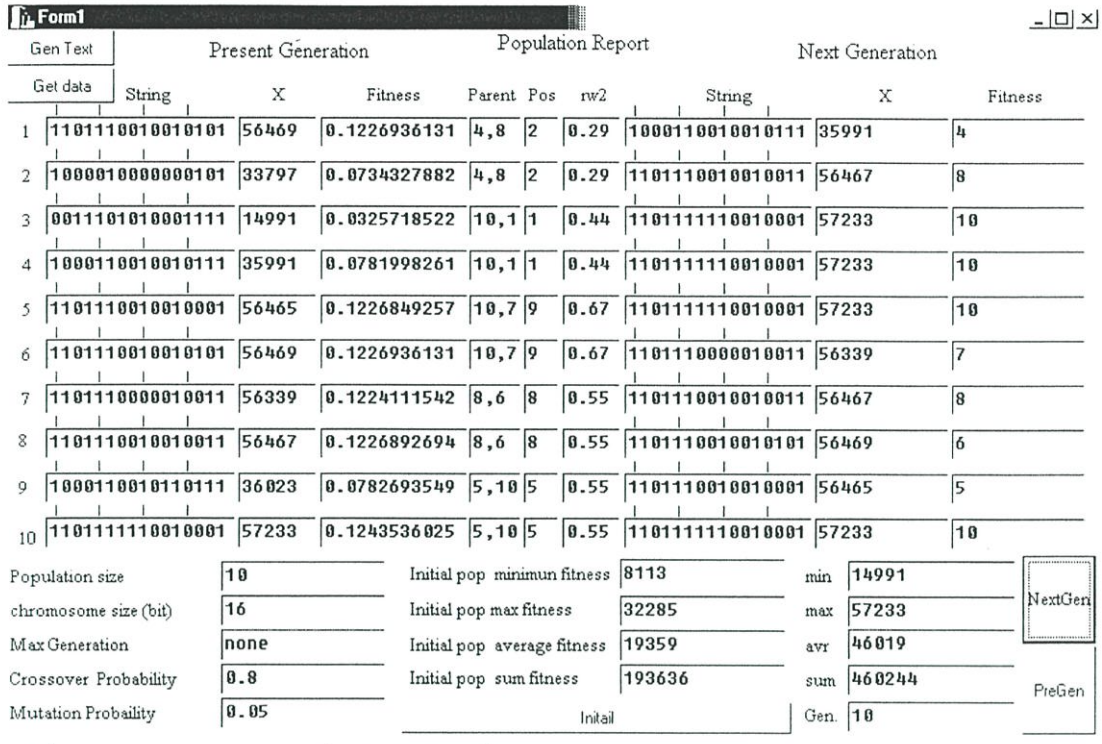
   1. จำนวนประชากร 10 โครโมโซม
   2. ครอสโอเวอร์ Pc = 0.8
   3. มิวเตชั่น Pm = 0.05
   4. รีโพรดัก 1 โครโมโซม



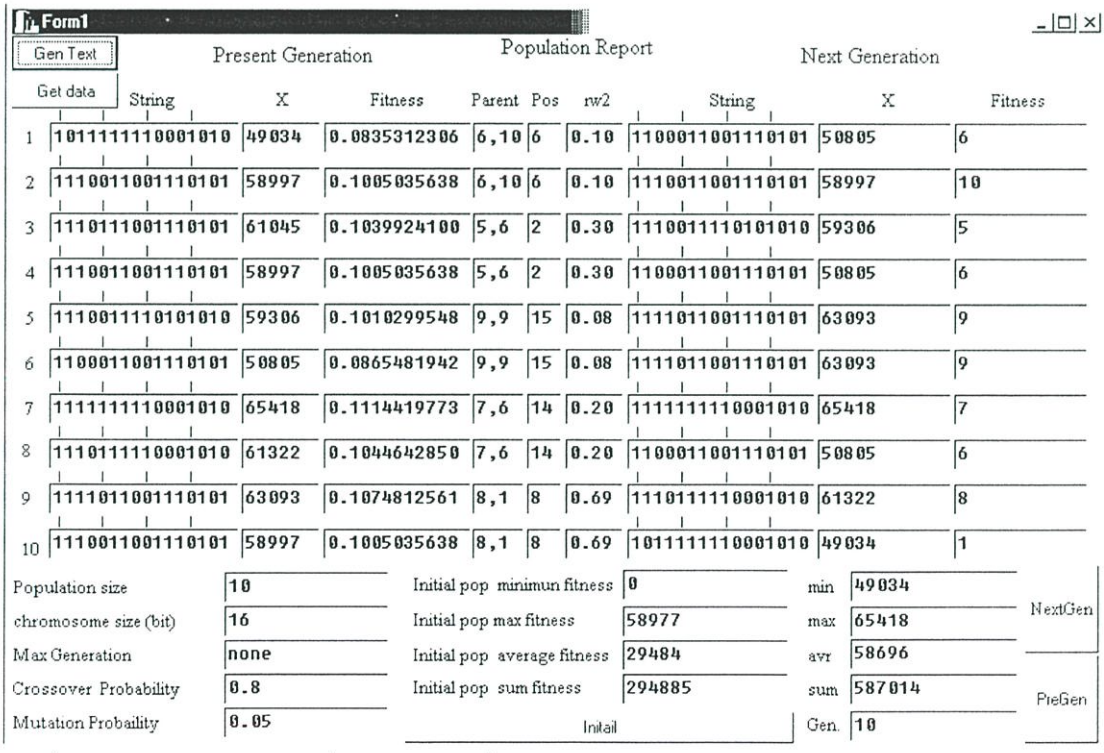
รูปที่ 4.13 การทดสอบแบบที่ 1 โดยการสุ่มเลือกโครโมโซมต้นแบบ



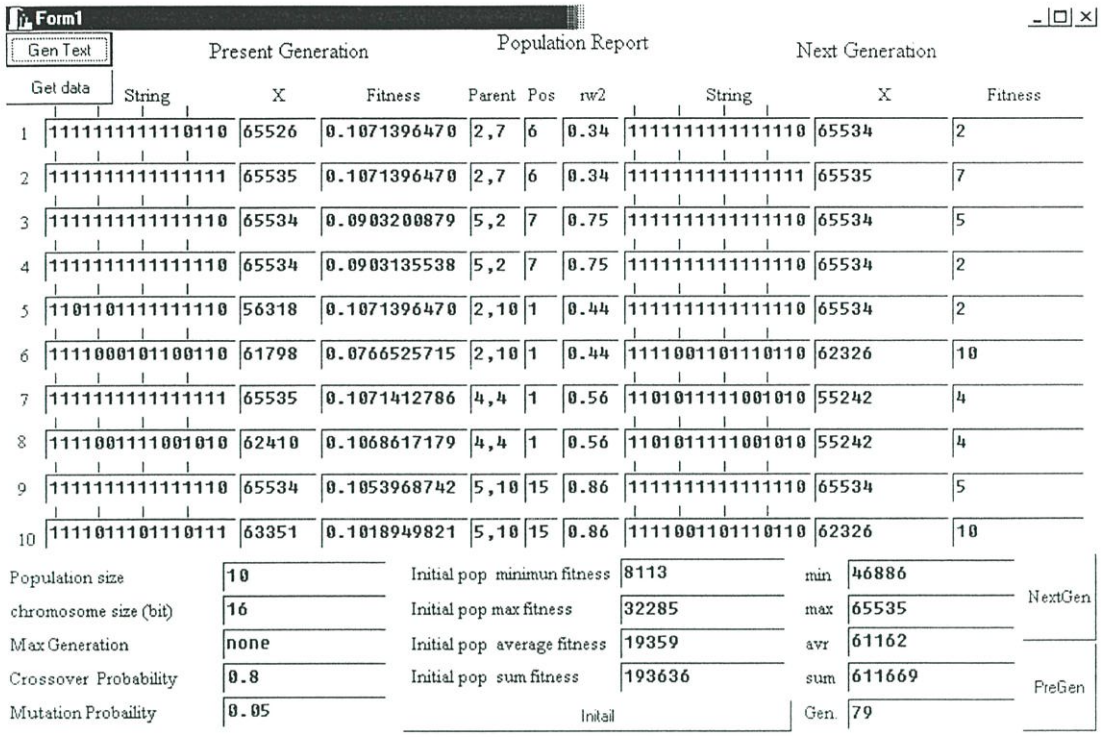
รูปที่ 4.14 การทดสอบแบบที่ 2 โดยการกระจายเลือกโครโมโซมต้นแบบ



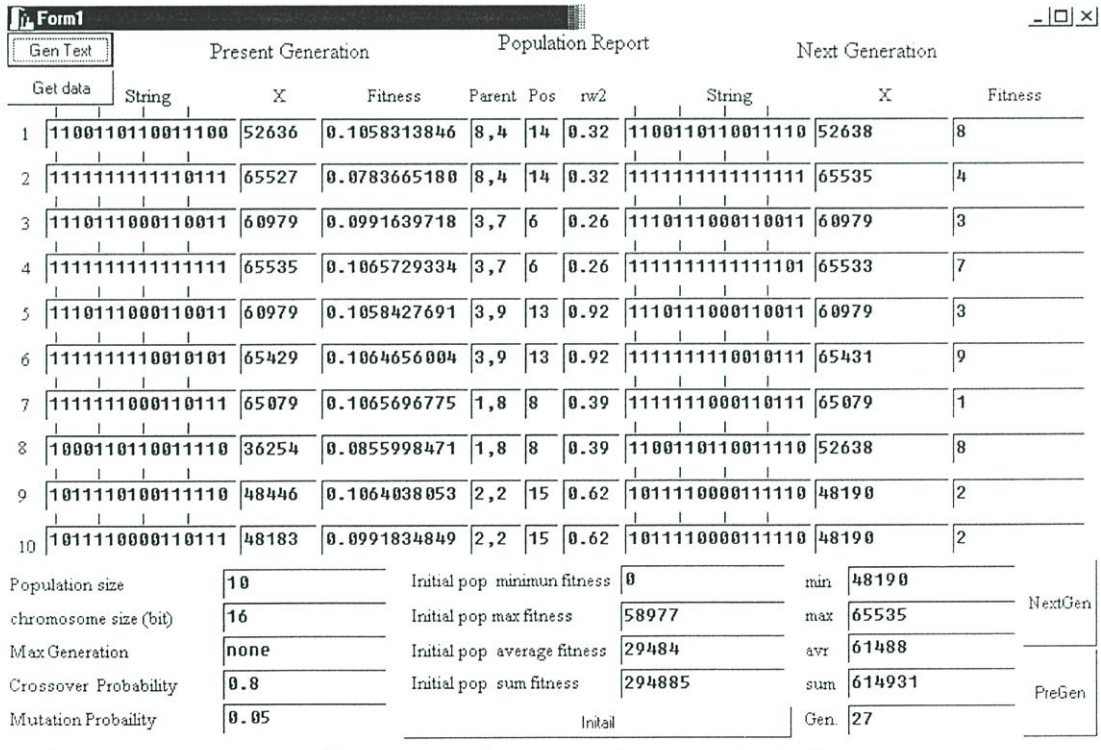
รูปที่ 4.15 การทดสอบแบบที่ 1 ในวัฏจักรที่ 10 (Gen=10)



รูปที่ 4.16 การทดสอบแบบที่ 2 ในวัฏจักรที่ 10 (Gen=10)



รูปที่ 4.17 การทดสอบแบบที่ 1 หาคำตอบที่เหมาะสมที่สุดได้ในวัฏจักรที่ 79 (Gen=79)



รูปที่ 4.18 การทดสอบแบบที่ 2 หาคำตอบที่เหมาะสมที่สุดได้ในวัฏจักรที่ 27 (Gen=27)

สรุปผล

จากการทำงานของ GA จะเห็นได้ว่ารูปที่ 4.13 ได้ทำการสุ่มโครโมโซมต้นแบบขึ้นมานั้นได้คำตอบของโครโมโซมต้นแบบที่ดีที่สุด คือ 30295 แต่ในรูปที่ 4.14 นั้นได้เลือกโครโมโซมต้นแบบจากการกระจายเป็นช่วง ๆ ที่เท่ากันทำให้ได้โครโมโซมต้นแบบที่ให้ค่าความเหมาะสมสูงสุด คือ 58977 จากนั้นในรูปที่ 4.15 กับ 4.16 เป็นการเปรียบเทียบผลของ GA ทั้ง 2 ในวัฏจักรที่ 10 ซึ่งแสดงให้เห็นว่า GA ในการทดสอบแบบที่ 2 นั้นให้ผลดีกว่า และสุดท้าย GA ในการทดสอบแบบที่ 2 นั้นสามารถหาคำตอบที่ดีที่สุด ๆ ได้ในวัฏจักรที่ 27 ขณะที่ GA ในการทดสอบแบบที่ 1 ต้องทำงานถึงวัฏจักรที่ 79 เพื่อให้ได้คำตอบที่ดีที่สุด

4.5.2 การแก้ปัญหาทายคำ

ปัญหาการทายคำ (Word Guessing Problem) [30] เป็นการเล่นเกมทายตัวอักษร เช่น การทายศัพท์ที่มีเงื่อนไขเป็นจำนวนครั้งในการทาย เพื่อตรวจสอบอักษรทุกตัวในคำนั้นต้องถูกต้องตามคำที่ต้องการ

1. ฟังก์ชันเป้าหมายของปัญหาทายคำ

ฟังก์ชันเป้าหมายของปัญหาทายคำ เป็นฟังก์ชันตรวจสอบความถูกต้องของแต่ละตัวอักษรของคำที่ต้องการหาคำตอบ โดยไม่คำนึงถึงอักษรตัวพิมพ์เล็ก หรืออักษรตัวพิมพ์ใหญ่ ดังสมการที่ (4.4)

$$f = \sum_{i=1}^{c} correct(chr_i) \tag{4.4}$$

โดยที่ c คือตัวอักษรทั้งหมด

correct (x) = 1 ถ้าตัวอักษรถูกต้อง, 0 ถ้าตัวอักษรไม่ถูกต้อง

ดังนั้นคำตอบคำตอบที่ดีที่สุดของปัญหาทายคำ คือ คำตอบที่ตัวอักษรถูกต้องทั้งหมด (MAX (f)) ซึ่งในที่นี้มีค่าเท่ากับ 16

2. โครโมโซมของปัญหาทายคำ

ตารางที่ 4.6 การกำหนดค่าบิตของโครโมโซมปัญหาทายคำ

ค่าบิต	อักษร	ค่าบิต	อักษร
1	A	27	A
2	b	28	B
:	:	:	:
26	z	52	Z

ลักษณะโครโมโซมของปัญหาทายคำประกอบด้วย บิตที่เป็นตัวอักษร (Character) ภายในคำที่ต้องการหาคำตอบ ดังนั้นรูปแบบโครโมโซมของปัญหาทายคำจึงเป็นการกำหนดรหัสค่าบิตของตัวอักษรแทนด้วยตัวเลข ดังตารางที่ 4.6 คือ อักษร a ถึง z มีค่า 1 ถึง 26 และ ตัวอักษร A ถึง Z มีค่าตั้งแต่ 27 ถึง 52 และจำนวนบิตทั้งหมดของโครโมโซมจะเท่ากับจำนวนตัวอักษรทั้งหมดของคำที่ต้องการหาคำตอบซึ่งลักษณะโครโมโซมของปัญหาทายคำคือ

$Charater_1$	$Charater_2$	$Charater_3$	$Charater_4$	$Charater_{c=5}$
H	E	L	L	O

ซึ่งแสดงในรูปของสัญลักษณ์ได้ดังนี้

$B_1 B_2 B_3 \ldots B_c$ ซึ่ง $B_i \in I[1,52]$

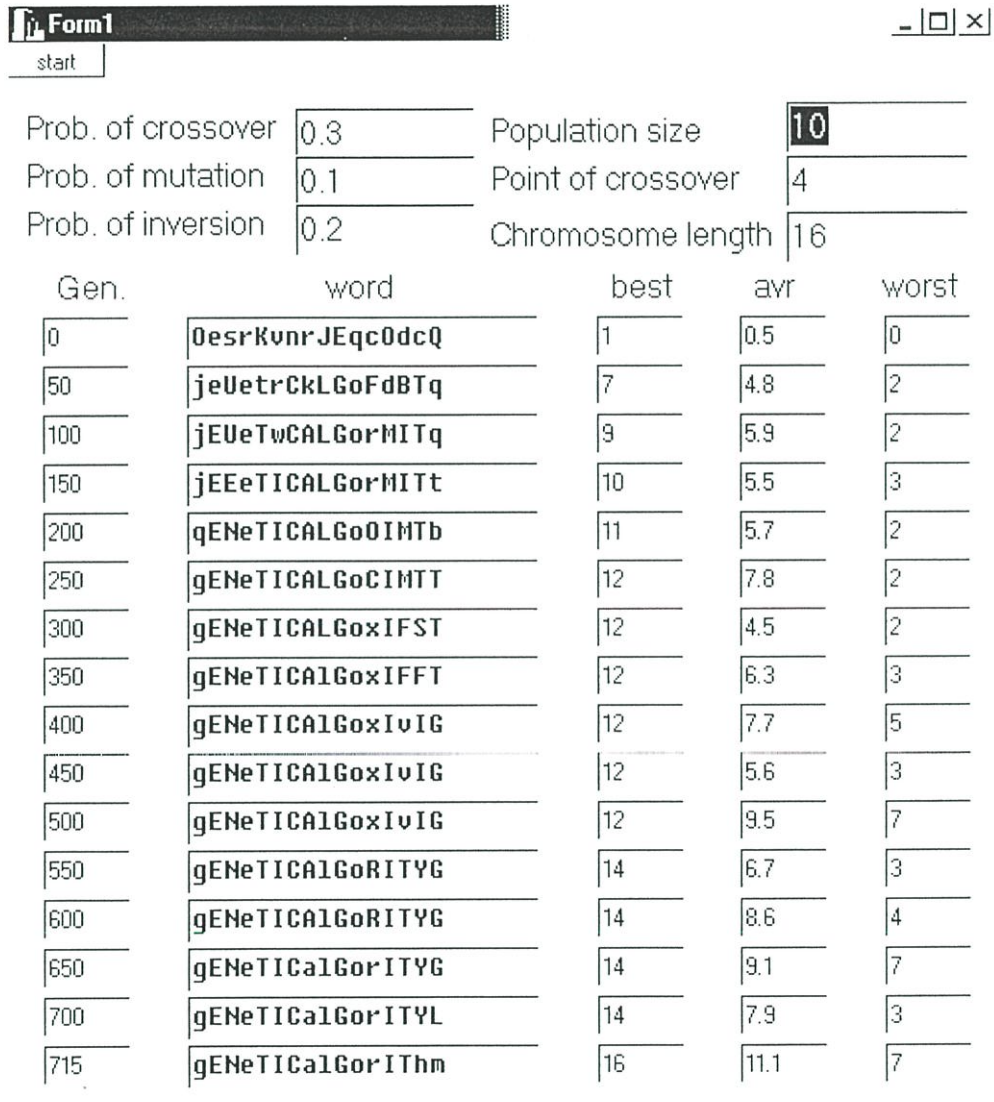
1. การดำเนินการทางพันธุกรรม
   1.1 ขนาดโครโมโซม (Chromosome length) = 16
   1.2 จำนวนประชากร 10 โครโมโซม
   1.3 ครอสโอเวอร์ Pc = 0.3,0.6
   1.4 มิวเตชั่น Pm = 0.1
   1.5 อินเวอร์ชัน Pi = 0.2
   1.6 รีโพรดัก 1 โครโมโซม
   1.7 การครอสโอเวอร์หลายจุด (Point of Crossover) = 4,2

นอกจากนี้ยังใช้การมิวเตชันแบบตัวเลข (integer mutation) คือ การเปลี่ยนค่าบิตเดิมเป็นค่าบิตใหม่ที่สามารถเป็นไปได้ในตำแหน่งบิตนั้น เช่น ในกรณีนี้อาจจะเป็นค่าอะไรก็ได้ในช่วง 1-52 (a-z และ A-Z)

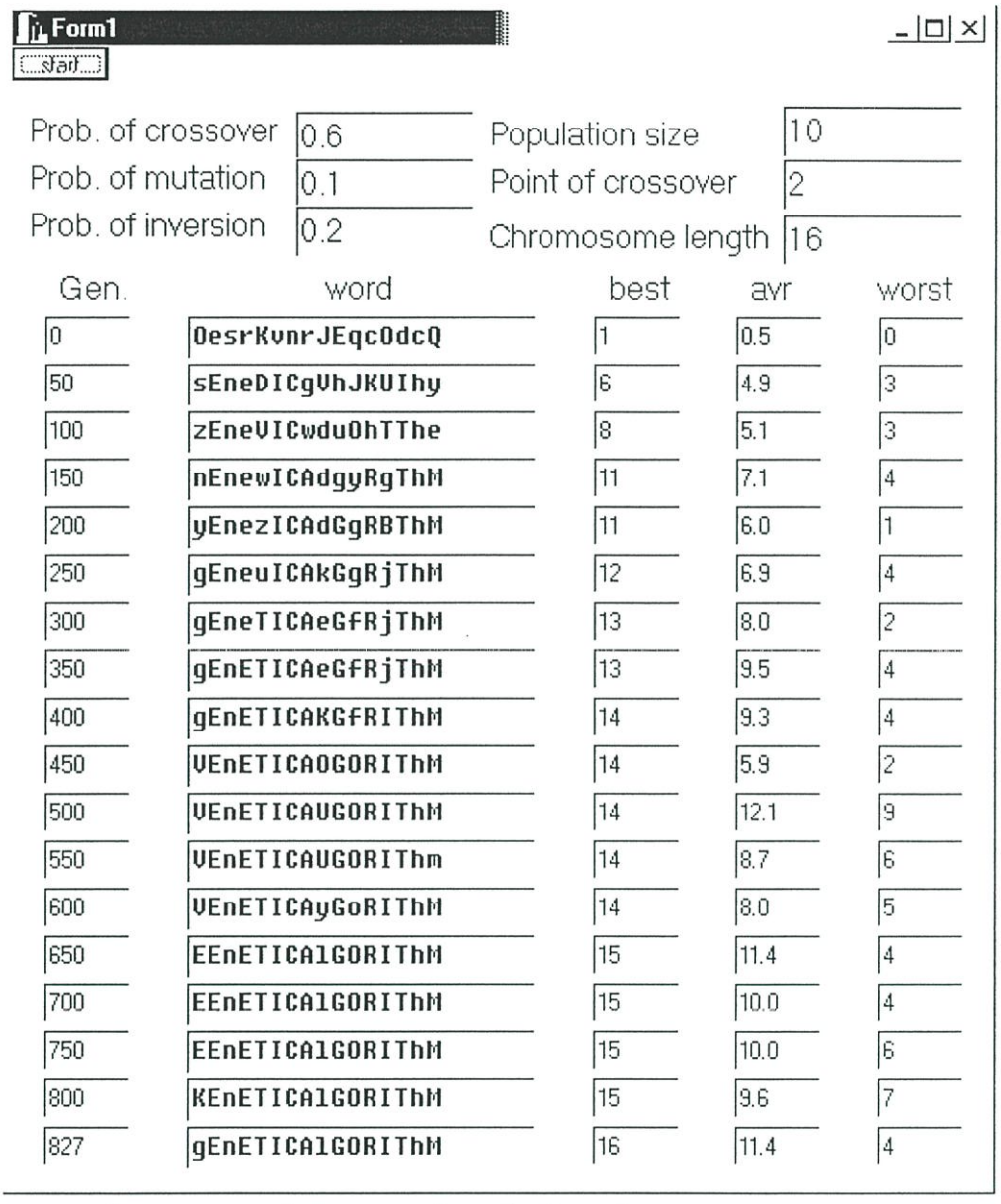
สรุปผล

ผลการทำงานของ GA ในปัญหาทายคำโดยทดสอบให้ทายคำ “GENETICalgorithm” โดยแบบแรกกำหนดให้ popsize = 10, Pc=0.3 และตำแหน่งในการครอสโอเวอร์เป็น 4 ตำแหน่ง Pm=0.1, Pi=0.2 และการรักษาค่าความเหมาะสม หรือรีโพรดักชันเท่ากับ 1 โครโมโซม ส่วนในแบบที่ 2 กำหนดให้ popsize = 10, Pc=0.6และตำแหน่งในการครอสโอเวอร์เป็น 2 ตำแหน่ง Pm=0.1, Pi=0.2 และการรักษาค่าความเหมาะสม หรือรีโพรดักชันเท่ากับ 1 โครโมโซม จากการทดสอบจะเห็นว่า

GA ในแบบแรกสามารถหาคำตอบที่มีค่าความเหมาะสมสูงสุดคือ 16 ในวัฏจักรที่ 715 ส่วนในแบบที่ 2 นั้นสามารถหาคำตอบที่เหมาะสมที่สุดในวัฏจักรที่ 827 ดังรูปที่ 4.19 และ 4.20 ซึ่งแสดงให้เห็นว่าการเพิ่มจำนวนจุดในการครอสโอเวอร์ก็สามารถทำให้การใช้งาน GA มีประสิทธิภาพดียิ่งขึ้นในการแก้ปัญหาทายคำนี้



รูปที่ 4.19 การทำงานของ GA ด้วยการครอสโอเวอร์ 4 ตำแหน่ง



รูปที่ 4.20 การทำงานของ GA ด้วยการครอสโอเวอร์ 2 ตำแหน่ง

4.6 สรุป

GA เป็นอัลกอริทึมที่เลียนแบบการทำงานของธรรมชาติ โดยอาศัยหลักการสุ่มซึ่งเป็นการสุ่มคำตอบจากกลุ่มคำตอบรุ่นถัดไป ส่วนการจะทำให้ GA นั้นหาคำตอบได้เร็วขึ้น ต้องอาศัยการกำหนดฟังก์ชันเป้าหมาย และฟังก์ชันความเหมาะสมให้สอดคล้องกับคำตอบที่ต้องการหา นอกจากนี้ยังมีวิธีการเพิ่มประสิทธิภาพของ GA โดยวิธีที่กล่าวมา เช่น รีโพรดักชัน ครอสโอเวอร์หลายจุด ไบนารีมิวเตชั่นแบบกำหนดค่าบิต และอินเวอร์ชัน ซึ่งผู้ใช้งาน GA ควรพิจารณานำมาใช้ให้เหมาะสม เพื่อเพิ่มประสิทธิภาพในการหาคำตอบของ GA

การประยุกต์ใช้ GA ในการแก้ปัญหาต่าง ๆ โดยการพัฒนาเจเนติกอัลกอริทึมนั้นจะประกอบด้วย การกำหนดรูปแบบโครโมโซมของปัญหาให้กับ GA โดยการพิจารณาจากสภาพของปัญหาหรือตัวแปรต่าง ๆ ที่ทำเป็นองค์ประกอบของปัญหา และกำหนดฟังก์ชันเป้าหมายให้กับ GA เพื่อใช้ในการประมวลผลให้ได้แนวทางการหาคำตอบที่ดีขึ้น รวมถึงตัวดำเนินการทางพันธุศาสตร์ที่เหมาะสมกับกระบวนการต่าง ๆ ของ GA ซึ่งรูปแบบของโครโมโซมฟังก์ชันเป้าหมาย หรือตัวดำเนินการทางพันธุศาสตร์ ในแต่ละปัญหาอาจจะกำหนดในรูปแบบอื่น ๆ ที่แตกต่างกันออกไปได้ อย่างไรก็ตามการปรับปรุง GA เพื่อหาคำตอบของปัญหาต่าง ๆ ให้มีประสิทธิภาพดีขึ้นนั้น ควรเลือกใช้เทคนิคต่าง ๆ ให้เหมาะสมด้วย

บทที่ 5
ตัวกรองอะแดปทีฟแบบนอตช์

ในบทนี้จะกล่าวถึงตัวกรองอะแดปทีฟแบบนอตช์ โดยจะกล่าวในส่วนของอัลกอริทึมที่ใช้ในการปรับค่าของสัมประสิทธิ์ของตัวกรองอะแดปทีฟแบบนอตช์อันดับสอง และการประยุกต์ใช้งาน GA เพื่อหาค่าเสต็ปไซส์ที่เหมาะสม ที่นำไปใช้ในการปรับเปลี่ยนค่าสัมประสิทธิ์ของฟิลเตอร์ทำงานร่วมกับอะแดปทีฟอัลกอริทึมแบบ NVSQLMP ที่นำเสนอในวิทยานิพนธ์ฉบับนี้

5.1 Least Mean P-Power Error Criterion (LMP)

ในหัวข้อนี้ได้พิจารณาหา cost ฟังก์ชัน $J(a)$ ของ ANF อันดับสอง ซึ่ง cost ฟังก์ชันที่ได้ จะเป็นฟังก์ชันของพารามิเตอร์ a และ a เป็นพารามิเตอร์ที่ใช้กำหนดความถี่นอตช์ของตัวกรองคำตอบของ a ที่ต้องการ คือ ค่า a ที่ทำให้ cost ฟังก์ชัน $J(a)$ มีค่าต่ำสุด (Minimized) นั่นคือ จะทำการ Minimize ค่าคาดหวังทางสถิติของสัญญาณเอาต์พุตยกกำลังสอง $E\{\hat{y}^2(n)\}$ ของตัวกรองนอตช์ ซึ่งข้อตัดสิน (Criterion) ดังกล่าวนี้ จะเป็นที่รู้จักกันโดยทั่วไป นั่นคือ Least Mean Square (LMS) error sense เหตุผลที่ LMS เป็นที่นิยมใช้กันเนื่องจากความง่ายในทางคณิตศาสตร์ (Mathematical simplicity)

ในช่วงหลายปีที่ผ่านมา การทำ Minimize แบบ L_p -norm ($\|\bullet\|_p$:p-norm) [7, 19] และได้ถูกทดสอบถึงการทำงานค่อนข้างสมบูรณ์ และนำมาใช้กับการประยุกต์ใช้งานต่าง ๆ มากมาย เช่น การออกแบบตัวกรองความถี่, การทำ Deconvolution, การเข้ารหัสเสียง และ การประมาณค่าความถี่ เป็นต้น โดยทั่วไป การออกแบบตัวกรองความถี่ จะต้องใช้การ Minimize แบบ L_∞ - norm [20] แต่เมื่อใช้สำหรับ ประมาณค่าความถี่ การ Minimize แบบ L_1 -norm ($\|\bullet\|_1$:1-norm) จะดีกว่าโดยเฉพาะในกรณีที่สัญญาณรบกวนเป็นแบบอิมพัลส์ (Impulse noise) [21] เนื่องจากการ Minimize แบบ L_p มีทฤษฎีรองรับอย่างสมบูรณ์แล้ว จึงเป็นสิ่งที่น่าสนใจ หากนำมาใช้พัฒนาเป็นอะแดปทีฟอัลกอริทึมสำหรับ ANF ซึ่งโดยปกติใช้ข้อตัดสินแบบ LMS แต่จะใช้ ข้อตัดสินแบบ Least Mean p-Power (LMP) Error Criterion แทน [22] ซึ่งพบว่าถ้าเลือกค่า p ที่เหมาะสมแล้ว เมื่อนำมาประยุกต์ใช้ในทางปฏิบัติหลาย ๆ อย่าง อะแดปทีฟอัลกอริทึมแบบ LMP จะมี สมรรถนะ ที่ดีกว่า อะแดปทีฟอัลกอริทึมแบบ LMS

เพื่อจะอธิบายข้อตัดสินแบบ LMP ให้สัญญาณอินพุตเป็นสัญญาณซายน์ความถี่เดียว ปนมากับสัญญาณรบกวนแบบขาว และ/หรือ สัญญาณรบกวนแบบอิมพัลส์ คือ

$$x(n) = A\sin(\omega_1 n + \theta) + v(n) \tag{5.1}$$

เมื่อ $v(n)$ คือ สัญญาณรบกวนแบบขาว หรือ อิมพัลส์ จากฟังก์ชันถ่ายโอนของตัวกรองอะแดปทีฟแบบนอตช์อันดับสอง พารามิเตอร์ a จะถูกปรับ ดังนั้นค่า mean p-Power error ของสัญญาณเอาต์พุตหรือ cost ฟังก์ชัน

$$J(a) = E\{|\hat{y}(n)|^p\} \tag{5.2}$$

จะถูกปรับจนมีค่าต่ำสุด สมการที่ (5.2) จะกลายเป็น LMS เมื่อให้ $p=2$ การหาค่าของสมการที่ (5.2) ทำได้โดยใช้การประมาณด้วยวิธีการของ Batch (Batch method) [23] คือ

ให้ N คือ จำนวนความยาวของข้อมูลอินพุต $x(n)$ และ $\hat{y}(n)$ คือ สัญญาณเอาต์พุตจากตัวกรองแล้ว $J(a)$ จะสามารถประมาณได้ตามสมการ คือ

$$\hat{J}(a) = \frac{1}{N}\sum_{n=1}^{N}|\hat{y}(n)|^p \tag{5.3}$$

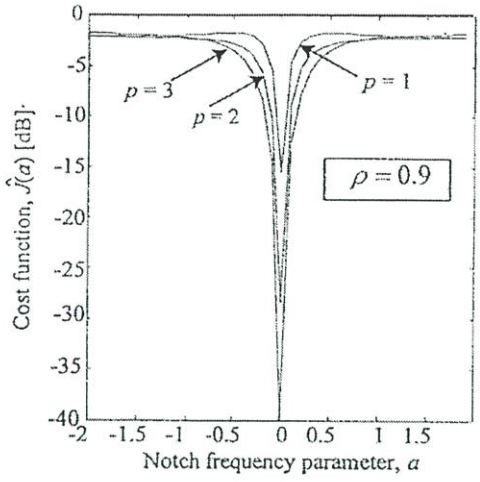
ดังนั้น จะสามารถประมาณคำตอบของพารามิเตอร์ a ได้ตามสมการ คือ

$$\hat{a} = \arg\min_{a}\hat{J}(a) \tag{5.4}$$

$\hat{a}$ หาได้โดยใช้ Nonlinear programming optimization เช่น อัลกอริทึมแบบ Steepest decent และ Conjugate gradient เป็นต้น ต่อไปจะพิจารณาตัวอย่าง เมื่อสัญญาณอินพุตมีค่าตามสมการคือ

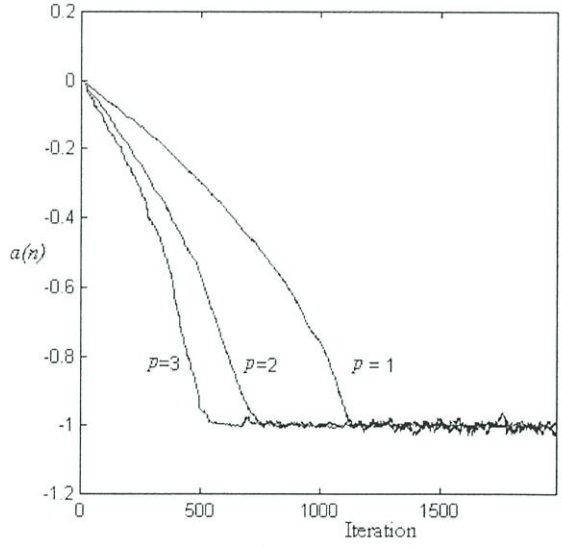
$$x(n) = \sin(\frac{\pi}{2}n + \frac{\pi}{2}) + v(n), n = 0,1,..., N \tag{5.5}$$

กำหนดให้ $v(n)$ เป็นสัญญาณรบกวนขาว มี Pdf เป็นแบบเกาส์เซียน มีค่า $\sigma_v^2 = 0.5$ ป้อนเข้าสู่ Notch filter โดยกำหนดให้ ρ=0.9 รูปที่ 5.1 แสดงค่า $\hat{J}(a)$ เทียบกับ a ที่ค่า p = 1, 2 และ 3



รูปที่ 5.1 ค่า mean p-Power error $\hat{J}(a)$ สำหรับ $p = 1, 2$ และ 3

จากรูปที่ 5.1 เห็นได้ชัดว่า ที่ $p = 3$ จะให้ค่า $\hat{J}(a)$ ต่ำสุด คือ –40 dB และ dip shape จะกว้างกว่าที่ค่า p=2 และ p=1 นั้นหมายความว่า เมื่อนำ steepest decent อัลกอริทึมมาใช้เพื่อหาจุดต่ำของ $\hat{J}(a)$ ที่ p มีค่าสูง จะทำให้อะแดปทีฟอัลกอริทึมทำงานได้เร็วเพิ่มขึ้น ดังแสดงในรูปที่ 5.2



รูปที่ 5.2 การลู่เข้าของพารามิเตอร์ a เมื่อใช้ Steepest decent อัลกอริทึม ที่ p=1, 2 และ 3

5.2 Gradient-Based Algorithm แบบ Least Mean p-Power Error Criterion (LMP)

อะแดปทีฟอัลกอริทึมแบบ Gradient-Based เป็นรูปแบบหนึ่งของ Recursive Prediction Error (RPE) อัลกอริทึม อาศัยหลักการค้นหาจุดต่ำสุดของ Cost ฟังก์ชันด้วยวิธีการแบบ Steepest decent อัลกอริทึม [24] ซึ่งมีรูปแบบสมการดังนี้คือ

$$a(n+1)=a(n)-\mu\frac{\partial J(a)}{\partial a} \tag{5.6}$$

เมื่อ μ คือ ค่าสเต็ปไซส์ ใช้กำหนดอัตราความเร็วของอะแดปทีฟอัลกอริทึม ขอบเขตของ μ สำหรับตัวกรองอะแดปทีฟแบบ IIR ยังไม่มีข้อสรุปที่แน่นอน ยังอยู่ในขั้นการค้นคว้าวิจัย ทั้งนี้เพราะความไม่เป็นเชิงเส้นของ Cost ฟังก์ชัน ซึ่งจะไม่สามารถทำนายหรือทราบล่วงหน้าได้เลยจึงเป็นการยากที่จะสร้างเป็นสูตรสำเร็จเหมือนอย่างกับตัวกรองอะแดปทีฟแบบ FIR [25] ดังนั้น จึงต้องใช้วิธีการ ลองผิดลองถูก (Trial and error) และ ปรับเปลี่ยนไปจนกว่าจะพอใจ

ในทางปฏิบัติ จะไม่สามารถสร้างอะแดปทีฟอัลกอริทึมตามสมการที่ (5.6) ได้ ทั้งนี้เพราะ Cost ฟังก์ชัน $J(a)=\{|\hat{y}(n)|^p\}$ เป็นค่าคาดหวังทางสถิติ ซึ่งจะต้องคิดค่า n = 0, 1, 2, ..., ∞ และเนื่องจาก สัญญาณในทางปฏิบัติ เป็นสัญญาณแบบ Nonstationary ซึ่งไม่สามารถคาดเดาอะไรได้มากนัก ดังนั้นจึงต้องใช้วิธีการประมาณแทน โดยการใช้ค่าของสัญญาณปัจจุบัน มาทำนายสัญญาณในอนาคต ค่าปัจจุบัน ณ เวลาหนึ่ง มักจะเรียกว่า Instantaneous value โดย จะใช้ค่า Instantaneous ของ $J(a)$ เพื่อแทนค่าเฉลี่ย Ensemble ของมัน ดังนั้น จะได้สมการสำหรับ ปรับค่าพารามิเตอร์ $\hat{a}$ ที่เป็นค่าประมาณของ Steepest decent อัลกอริทึม ซึ่งเรียกชื่อใหม่ว่า Gradient-Based อัลกอริทึม ดังนี้ คือ

$$a(n+1)=a(n)-\mu\frac{\partial|\hat{y}(n)|^p}{\partial a} \tag{5.7}$$

ในกรณีของสมการที่ (5.7) p มีโอกาสเป็นเลขคี่ ซึ่งถ้าช่วงเวลาใดที่ $\hat{y}(n)$ มีค่าเป็นลบ และ p เป็นเลขคี่ก็ทำให้ทิศทางของ Gradient เป็นบวก – ลบ สลับกัน ขึ้นกับสัญญาณ $\hat{y}(n)$ จะทำให้อะแดปทีฟอัลกอริทึม ไม่สามารถค้นหาจุดต่ำสุดของ cost ฟังก์ชันได้เลย ดังนั้น จึงต้องใช้ตัวกระทำ $|\,.\,|$ เพื่อบังคับให้ $\hat{y}^p(n)$ มีค่าเป็นบวก สำหรับทุกค่าของ p เพื่อทำให้ทิศทางของ Gradient เป็นลบตลอดการทำงาน ดังนั้นจะได้ว่า

$$|\hat{y}(n)|^p=\begin{cases}[\hat{y}(n)]^p & p:even\\ \mathrm{sgn}[\hat{y}(n)]\cdot[\hat{y}(n)]^p & p:odd\end{cases} \tag{5.8}$$

ดังนั้นอนุพันธ์ย่อยของ $|\hat{y}(n)|^p$ เทียบกับพารามิเตอร์ a คือ

$$\frac{\partial|\hat{y}(n)|^p}{\partial a}=\begin{cases}p\cdot[\hat{y}(n)^{p-1}\cdot\dfrac{\partial\hat{y}(n)}{\partial a} & p:even\\ p\cdot\mathrm{sgn}[\hat{y}(n)]\cdot[\hat{y}(n)]^{p-1}\cdot\dfrac{\partial\hat{y}(n)}{\partial a} & p:odd\end{cases} \tag{5.9}$$

โดยที่ $\text{sgn}[x] \equiv \frac{x}{|x|}$ ดังนั้นสมการที่ (5.7) จะกลายเป็น

$$\begin{aligned} a(n+1) &= a(n) - \mu \cdot U_1(n) \qquad p: even \\ a(n+1) &= a(n) - \mu \cdot U_2(n) \qquad p: odd \end{aligned} \tag{5.10}$$

เมื่อ

$$\begin{aligned} U_1(n) &= p \cdot [\hat{y}(n)]^{p-1} \cdot \hat{e}(n) \\ U_2(n) &= p \cdot sng[\hat{y}(n)] \cdot [\hat{y}(n)]^{p-1} \cdot \hat{e}(n) \end{aligned} \tag{5.11}$$

และ

$$\hat{e}(n) = \frac{\partial \hat{y}(n)}{\partial a} \tag{5.12}$$

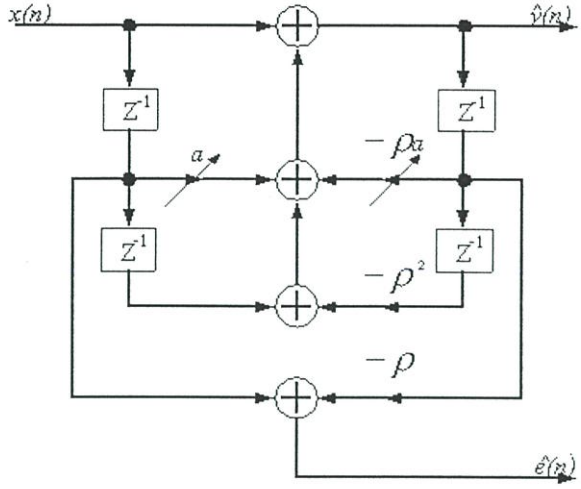
สมการของ $\hat{y}(n)$ หาได้โดยทำการแปลง Z กลับ (Inverse Z-transform) ของสมการที่ (3.9) ในบทที่ 3 ได้ดังนี้

$$\hat{y}(n) = x(n) + ax(n-1) + x(n-2) - a\rho y(n-1) - \rho^2 y(n-2) \tag{5.13}$$

และสมการของ $\hat{e}(n)$ หาได้โดยใช้วิธีการของ Pseudolinear regression algorithm [26] ดังนี้คือ

$$\hat{e}(n) = \frac{\partial y(n)}{\partial a} \approx x(n-1) - \rho y(n-1) \tag{5.14}$$

จากที่กล่าวมานี้ คืออะแดปทีฟอัลกอริทึมที่นำเสนอในบทวิจัย [22] รูปที่ 5.3 แสดงโครงสร้างแบบ Direct form ของ ANF



รูปที่ 5.3 โครงสร้างแบบ Direct form ของตัวกรอง ANF

5.3 อะแดปทีฟอัลกอริทึมที่นำเสนอในวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้ นำเสนออะแดปทีฟอัลกอริทึมที่ได้ทำการดัดแปลงจาก LMP อัลกอริทึม แบบ [5] อะแดปทีฟอัลกอริทึม แบบต่าง ๆ จะมีข้อดี ข้อเสีย แตกต่างกัน ขึ้นกับงานที่จะไปใช้ อย่างไรก็ตาม การจะเลือกว่าจะนำอะแดปทีฟอัลกอริทึมแบบใดมาใช้นั้น จะขึ้นกับเหตุผล 4 ประการ คือ

i) ความเร็วในการหาคำตอบ

ii) ความถูกต้องของคำตอบ (ดูจากค่า MSE มีค่าต่ำ)

iii) ความสามารถในการติดตามสัญญาณสุ่มแบบ Nonstationary

iv) ความซับซ้อนในการคำนวณ

ดังนั้นหาก อะแดปทีฟอัลกอริทึมแบบใด ที่ให้คุณสมบัติทั้ง 4 ข้อนี้ ก็จะเลือกนำมาใช้งาน

5.3.1 อะแดปทีฟอัลกอริทึมแบบ QLMP

สมการสำหรับปรับพารามิเตอร์ a ของอะแดปทีฟอัลกอริทึมแบบ QLMP ที่ได้ดัดแปลงจาก LMP คือ

$$\begin{aligned} a(n+1) &= a(n) - \mu \cdot \mathrm{sgn}[U_1(n)] \quad p: even \\ a(n+1) &= a(n) - \mu \cdot \mathrm{sgn}[U_2(n)] \quad p: odd \end{aligned} \tag{5.15}$$

โดยที่ $U_i(n), i = 1,2$ มีค่าตามสมการที่ (5.11) วิธีการนี้ อาศัยตัวกระทำ sgn[·] สำหรับตรวจสอบเครื่องหมายของ $U_i(n)$ ถ้า $U_i(n)$ มีค่าเป็นบวก จะได้ค่า $\mathrm{sgn}[U_i(n)] = 1$ และถ้า $U_i(n)$ มีค่าเป็นลบ จะได้ค่า $\mathrm{sgn}[U_i(n)] = -1$ หรือกล่าวอีกนัยหนึ่ง คือ จะทำการ Quantize ค่า $U_i(n)$ ให้เพียงสองระดับ คือ 1 และ –1 ข้อดี คือ จะสามารถลดการคูณลงได้ 1 ครั้ง ต่อ 1 ตัวอย่าง (Sample)

สัญญาณเอาต์พุต (เทียบกับ LMP ที่ $p>1$) เพราะ μ จะถูกคูณด้วย 1 หรือ –1 ซึ่งปกติแล้วจะไม่นับว่าเป็นการคูณ ในกรณีที่เลือกให้ $p=1$ จะได้

$$U_1(n) = U_2(n) = \text{sgn}[\hat{y}(n)] \cdot \hat{e}(n) \tag{5.16}$$

และสมการสำหรับปรับพารามิเตอร์ a คือ

$$a(n+1) = a(n) - \mu \cdot \text{sgn}[\text{sgn}\{\hat{y}(n)\} \cdot \hat{e}(n)] \tag{5.17}$$

ซึ่งจะลดการคูณลงได้ 2 ครั้ง (เทียบกับ LMP ที่ $p \neq 1$) นอกจากนี้ ยังพบว่า ความเร็วในการทำงานของอะแดปทีฟอัลกอริทึมจะขึ้นกับ ค่า μ เท่านั้น ดังนั้นในกรณีที่ใช้ μ เท่ากัน อะแดปทีฟอัลกอริทึมแบบ QLMP จะทำงานได้เร็วกว่า แบบ LMP และความเร็วจะไม่ขึ้นกับค่า p อีกด้วย ซึ่งจะต่างจากแบบ LMP เพราะเมื่อ p สูงขึ้น จะทำให้ความเร็วในการทำงานเพิ่มขึ้น แต่อย่างไรก็ตาม อะแดปทีฟอัลกอริทึมแบบ QLMP จะมีข้อเสีย คือ พารามิเตอร์ a จะมีค่า Gradient noise [24-25] สูงกว่าแบบ LMP เมื่อค่า μ เท่ากัน นั่นหมายความว่า variance ของ พารามิเตอร์ a มีค่าสูงกว่านั่นเอง

สรุปได้ว่า ข้อดีสองประการของอะแดปทีฟอัลกอริทึมแบบ QLMP คือ (i) ลดความซับซ้อนในการคำนวณลงได้ โดยเฉพาะที่ $p=1$ จะดีที่สุด ดังนั้น ในการใช้งานควรใช้ $p=1$ (ii) คือ ค้นหาคำตอบได้เร็ว

5.3.2 อะแดปทีฟอัลกอริทึมแบบ VSQLMP

อะแดปทีฟอัลกอริทึมแบบ VSQLMP นี้ได้นำข้อดีของการทำให้ค่าเสต็ปไซส์ เปลี่ยนแปลงตามเวลาคือ จะทำงานได้เร็วขึ้น โดยจะสร้างสมการสำหรับปรับค่าเสต็ปไซส์ แบบรีเคอร์ซีฟโดยตรง ไม่ต้องใช้การหารการทำเช่นนี้สามารถลด Gradient noise ได้ เมื่อเทียบกับ QLMP นั่นหมายความว่า ค่า Variance ของคำตอบมีค่าต่ำกว่าด้วย จากสมการที่ (5.15) ทำการแทน $\mu \rightarrow \mu(n)$ ดังนั้นสมการที่ใช้สำหรับปรับพารามิเตอร์ a คือ

$$\begin{aligned} a(n+1) &= a(n) - \mu(n) \cdot \text{sgn}[U_1(n)] \quad p: even \\ a(n+1) &= a(n) - \mu(n) \cdot \text{sgn}[U_2(n)] \quad p: odd \end{aligned} \tag{5.18}$$

เมื่อ

$$\mu(n+1) = \gamma \cdot \mu(n) + \lambda \Psi^2(n) \tag{5.19}$$

โดยที่

$$\Psi(n)=\alpha\cdot\Psi(n-1)+(1-\alpha)y^{2}(n) \tag{5.20}$$

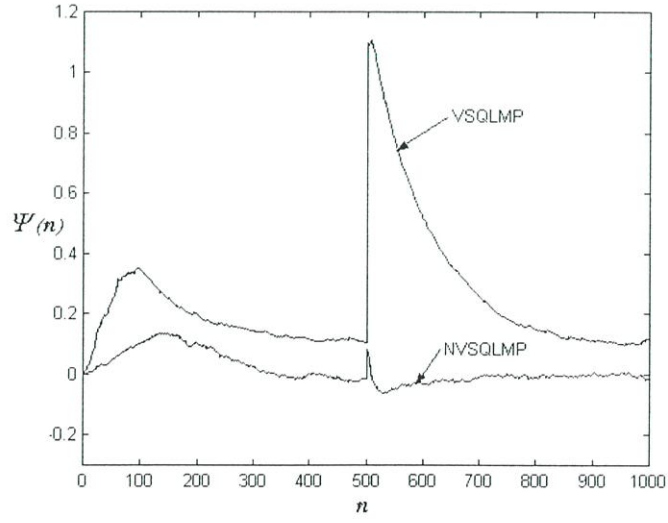
$\gamma.\lambda$ และ α เป็นค่าคงที่อยู่ในช่วง 0 ถึง 1 วิธีการนี้จะใช้ค่าพลังงานประมาณ $\hat{y}(n)$ มาใช้สำหรับควบคุมค่าเสต็ปไซส์ ให้เปลี่ยนแปลงตามเวลาเนื่องจากค่าเสต็ปไซส์ จะปรับตัวตามค่าพลังงานของสัญญาณเอาต์พุต $\hat{y}(n)$ ในช่วงแรกขณะที่ ANF เริ่มทำงาน ยังไม่สามารถตรวจวัดความถี่ได้ ค่าพลังงานของ $\hat{y}(n)$ จะมีค่าสูง เป็นผลให้ $\mu(n)$ มีค่าสูงตาม อะแดปทีฟอัลกอริทึมจึงทำงานได้เร็ว หลังจากลู่เข้าแล้ว ค่าพลังงาน $\hat{y}(n)$ มีค่าต่ำโดยเหลือเฉพาะพลังงานของสัญญาณรบกวนเท่านั้น ดังนั้นเมื่อพลังงานของสัญญาณรบกวนมีค่าสูงขึ้นทันทีทันใด หรือ เกิดสัญญาณรบกวนอิมพัลส์ขึ้น จะทำให้ค่าสัมประสิทธิ์ของฟิลเตอร์เกิดความแปรปรวนสูงมากและอาจทำให้ไม่สามารถลู่เข้าสู่คำตอบได้อีก ในวิทยานิพนธ์นี้ได้ใช้เกรเดียนต์ของเอาต์พุตเข้ามาช่วยเพื่อแก้ปัญหาในเรื่องอิมพัลส์ เพื่อความสะดวกต่อไปขอเรียกอัลกอริทึมที่นำเสนอนี้ว่า NVSQLMP อัลกอริทึม จากสมการที่ (5.20) จะได้ว่า

$$\Psi(n)=\alpha\cdot\Psi(n-1)+(1-\alpha)y(n)\frac{\partial y(n)}{\partial a} \tag{5.21}$$

หรือ

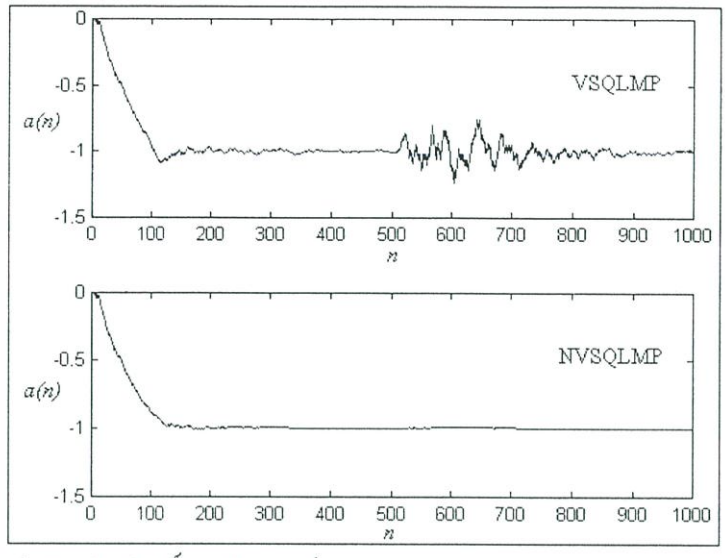
$$\Psi(n)=\alpha\cdot\Psi(n-1)+(1-\alpha)y(n)e(n) \tag{5.22}$$

จากรูปที่ 5.4 ได้แสดงค่า $\Psi(n)$ จากสมการที่ (5.21) โดยกำหนดให้เกิดสัญญาณรบกวนอิมพัลส์ขึ้นที่ n = 500 ซึ่งจะเห็นได้ว่าค่า $\Psi(n)$ ของอัลกอริทึมแบบ NVSQLMP นั้นมีค่าต่ำกว่า อัลกอริทึมแบบ VSQLMP



รูปที่ 5.4 เปรียบเทียบค่า $\Psi(n)$ ของอัลกอริทึมแบบ NVSQLMP และ VSQLMP เมื่อเกิดสัญญาณรบกวนอิมพัลส์ขึ้นที่ n = 500

จากสมการที่ (5.22) ทำให้ได้อะแดปทีฟอัลกอริทึมแบบ VSQLMP ที่สามารถทนต่อสัญญาณอิมพัลส์ได้ดีขึ้น



รูปที่ 5.5 การปรับค่าสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ จาก VSQLMP และ NVSQLMP อัลกอริทึมเมื่อเกิดสัญญาณรบกวนอิมพัลส์ขึ้นที่ n = 500

รูปที่ 5.5 ผลการจำลองการทำงาน การปรับค่าสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ โดยเปรียบเทียบระหว่าง อัลกอริทึมแบบ VSQLMP และ NVSQLMP เมื่อเกิดสัญญาณรบกวนอิมพัลส์ขึ้นที่ n = 500 โดยพารามิเตอร์ต่าง ๆ มีค่าดังนี้ $\mu(0) = 0.02$, $\omega_0 = \pi/3$, $\rho = 0.9$, $\gamma = 0.99$, $\alpha = 0.99$, $\lambda = 0.001$, SNR = 7dB และ p = 1

5.3.3 อะแดปทีฟอัลกอริทึมที่ทำงานร่วมกันของ NVSQLMP กับเจเนติก

ในหัวข้อนี้ได้นำเสนอ อะแดปทีฟอัลกอริทึมที่เกิดจากการทำงานร่วมกันของ NVSQLMP กับเจเนติกซึ่งจะนำไปหาค่าเสต็ปไซส์ ที่เหมาะสม โดยในช่วงแรกใช้ GA เพียงอย่างเดียวในการหาค่าเสต็ปไซส์ ซึ่งจะแสดงให้เห็นถึงผลของการปรับค่าของ Pm และ Pc จะส่งผลต่อค่าสัมประสิทธิ์ของฟิลเตอร์เพียงใด

1. ฟังก์ชันเป้าหมายและฟังก์ชันความเหมาะสม

ในปัญหานี้ได้ใช้ฟังก์ชันเป้าหมายเป็นอัพเดทฟังก์ชันของ NVSQLMP คือ

$$a(n+1) = a(n) - \mu(n) \cdot sgn\left[sgn(y(n)) \cdot \frac{\partial y(n)}{\partial a} \right]$$

โดยพารามิเตอร์ต่างๆ มีค่าดังนี้

$$\omega = \pi/3, \quad \mu(0) = 0.02, \quad \rho = 0.9 \text{ และ } p = 1$$

ส่วนฟังก์ชันความเหมาะสมคือ

$$e_j^2 = \frac{1}{t_c}\sum_{n=1}^{t_c}\left[d(n) - y_j(n)\right]^2$$

t_c คือ Window Size ใช้ค่าเท่ากับ 4

$d(n)$ คือ สัญญาณเอาต์พุตที่ต้องการ (Desired Output) กรณีนี้ใช้สัญญาณอินพุตโดยตรง

$y_j(n)$ คือ เอาต์พุตที่ได้จากโครโมโซม n ในวัฏจักร j

ดังนั้นจะได้คำตอบที่ดีที่สุด คือ $e_j^2 = 0$ (MIN(F))

2. ลักษณะโครโมโซม เป็นโครโมโซมแบบไบนารีขนาด 16 บิต

B_1	B_2	B_3	...	B_{16}

$B_i \in [0,1]$

โดยที่ B_{16} นั้นคือค่าบิตต่ำสุด หรือ LSB ส่วน B_1 นั้น คือ ค่าบิตสูงสุด หรือ MSB

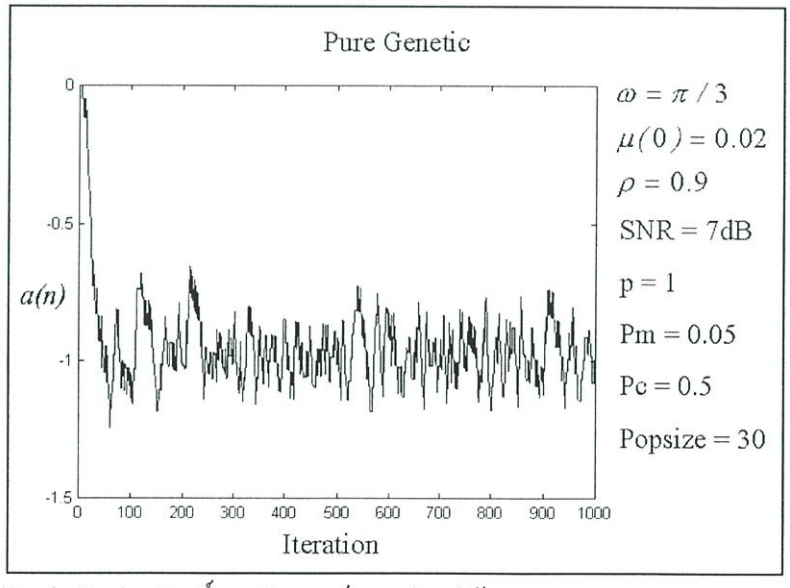
3. การดำเนินการทางพันธุศาสตร์

3.1. จำนวนประชากรเท่ากับ 30

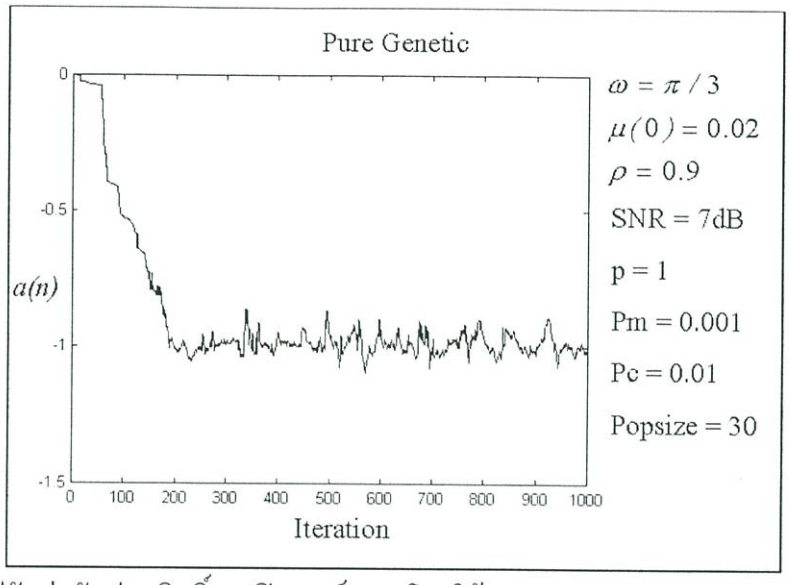
3.2. ครอสโอเวอร์ Pc = 0.5, 0.01

3.3. มิวเตชั่น Pm = 0.05, 0.01

3.4. รีโพรดักชัน 1 โครโมโซม



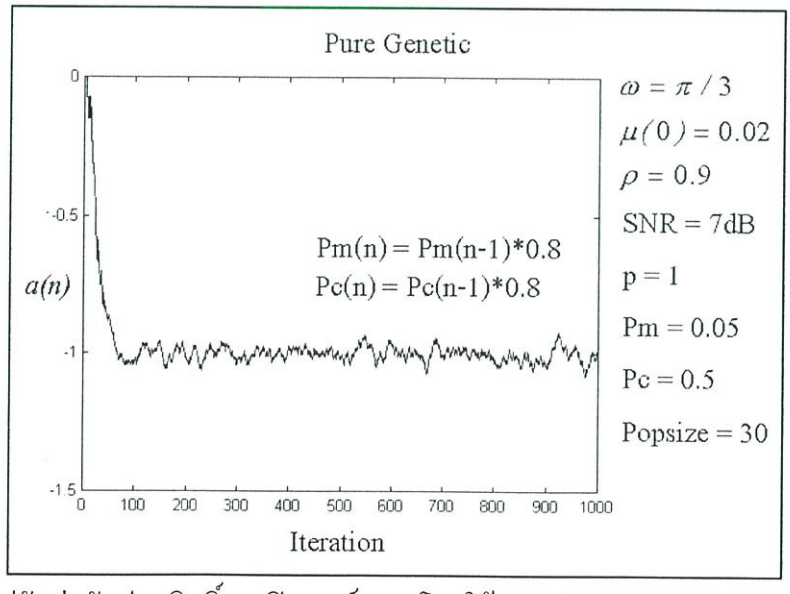
รูปที่ 5.6 การปรับค่าสัมประสิทธิ์ของฟิลเตอร์ *a(n)* โดยใช้ GA กำหนด Pm = 0.05, Pc = 0.5



รูปที่ 5.7 การปรับค่าสัมประสิทธิ์ของฟิลเตอร์ *a(n)* โดยใช้ GA กำหนด Pm = 0.001, Pc = 0.01

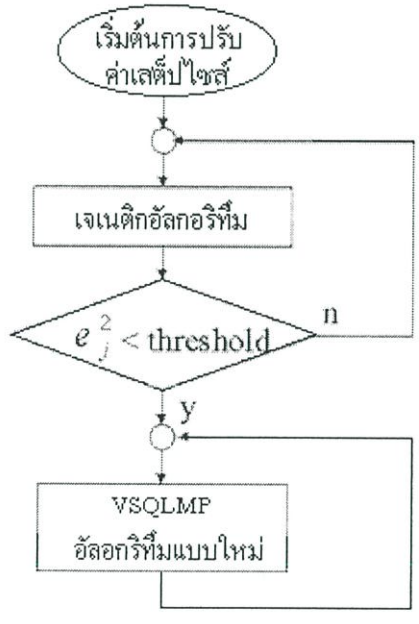
จากรูปที่ 5.6 และ 5.7 แสดงให้เห็นว่า กรณีที่ค่า Pm และ Pc มีค่าสูงดังรูปที่ 5.6 นั้นทำให้การปรับค่าสัมประสิทธิ์ของฟิลเตอร์ลู่เข้าสู่คำตอบได้เร็ว แต่มีค่าความแปรปรวนสูง ส่วนกรณีที่ค่า Pm และ Pc มีค่าต่ำลงจะทำให้การปรับค่าสัมประสิทธิ์ของฟิลเตอร์ลู่เข้าสู่คำตอบได้ช้าลง และมีค่าความแปรปรวนก็ต่ำลงด้วยเช่นกัน

อีกกรณีหนึ่งจากผลที่ได้จากรูปที่ 5.6 และ 5.7 ทำให้เกิดสมมติฐานว่า ถ้าทำให้ค่า Pm และ Pc สูงในช่วงแรก ๆ จากนั้นค่อย ๆ ลดลงเรื่อย ๆ ผลที่เกิดขึ้นจะเป็นอย่างไร จากรูปที่ 5.8 โดยที่พารามิเตอร์ต่าง ๆ เหมือนเดิมกำหนดให้ค่าเริ่มต้นของ Pm – 0.05, Pc – 0.5 จากนั้นทุก ๆ วัฏจักรก็ให้ค่า Pm และ Pc ลดลงครั้งละ 20 % ผลที่ได้คือ ค่าสัมประสิทธิ์ของฟิลเตอร์สามารถลู่เข้าสู่คำตอบได้เร็ว โดยที่ค่าความแปรปรวนไม่สูงมากนัก

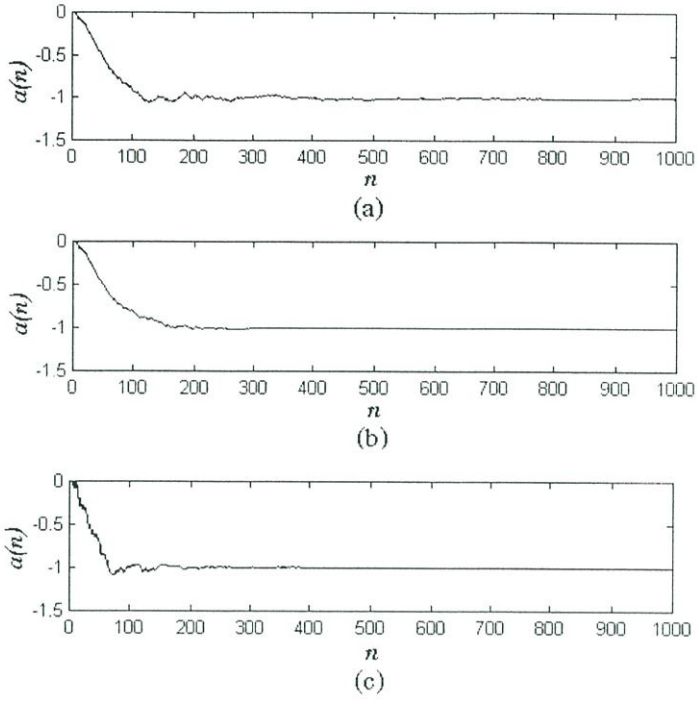


รูปที่ 5.8 การปรับค่าสัมประสิทธิ์ของฟิลเตอร์ *a(n)* โดยใช้ GA กำหนด Pm และ Pc แปรค่าได้

จากผลที่ได้การหาค่าเสต็ปไซส์ สำหรับอะแดปทีฟ IIR นอตช์ฟิลเตอร์ ทำให้เกิดแนวคิดในการพัฒนาอะแดปทีฟอัลกอริทึมคือ เจเนติกอัลกอริทึมสามารถหาคำตอบได้เร็วในกรณีที่ มีค่า Pm และ Pc สูง แต่ก็จะทำให้ค่าความแปรปรวนสูงตามไปด้วย ดังนั้นในวิทยานิพนธ์นี้จึงใช้ อะแดปทีฟอัลกอริทึมแบบ NVSQLMP เข้ามาทำงานร่วมกับเจเนติกอัลกอริทึม โดยมีเงื่อนไขดังรูปที่ 5.9 คือเริ่มต้นการหาค่าเสต็ปไซส์โดย GA จะทำงานไปจนกว่าค่า e_j^2 มีค่าน้อยกว่าที่กำหนด (threshold) จึงเริ่มการทำงานของ NVSQLMP อัลกอริทึมเพื่อหาค่าเสต็ปไซส์ต่อ ๆ ไป ซึ่งผลที่ได้จะทำให้ได้ตัวกรองที่มีความเร็วในการทำงานสูงและสามารถทนต่อสัญญาณรบกวนได้ดี ซึ่งเป็นผลมาจากการทำงานของ เจเนติกอัลกอริทึมกับอัลกอริทึม NVSQLMP ตามลำดับ และได้จำลองการทำงานของอัลกอริทึมทั้ง 3 แบบผลดังรูปที่ 5.10



รูปที่ 5.9 เงื่อนไขในการทำงานร่วมกันของ เจเนติกอัลกอริทึม กับ NVSQLMP อัลกอริทึม



รูปที่ 5.10 เปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ *a(n)* (a) VSQLMP อัลกอรทึม, (b) NVSQLMP อัลกอรทึม (c) NVSQLMP อัลกอรทึม ทำงานร่วมกับเจเนติกอัลกอริทึม

5.4 สรุป

สำหรับอะแดปทีฟอัลกอริทึมที่นำเสนอในวิทยานิพนธ์นี้ได้ปรับปรุงจากอะแดปทีฟอัลกอริทึมแบบ VSQLMP ซึ่งการปรับปรุงอัลกอริทึมนั้น ส่วนใหญ่มักจะทำในลักษณะที่ต้องการเพิ่มความเร็วในการทำงาน หรือเพิ่มความทนทานต่อสัญญาณรบกวน ในการปรับปรุงอะแดปทีฟอัลกอริทึมแบบ VSQLMP นี้ได้ใช้เกรเดียนต์เอาต์พุตเพื่อทำให้ฟังก์ชันถ่ายโอนพลังงานมีค่าลดลง จึงทำให้ได้อะแดปทีฟอัลกอริทึมแบบ NVSQLMP ที่ทนทานต่อสัญญาณรบกวนแบบอิมพัลส์ ส่วนในการหาค่าเสต็ปไซส์สำหรับอะแดปทีฟ IIR นอตช์ฟิลเตอร์ด้วย GA แสดงให้เห็นว่า

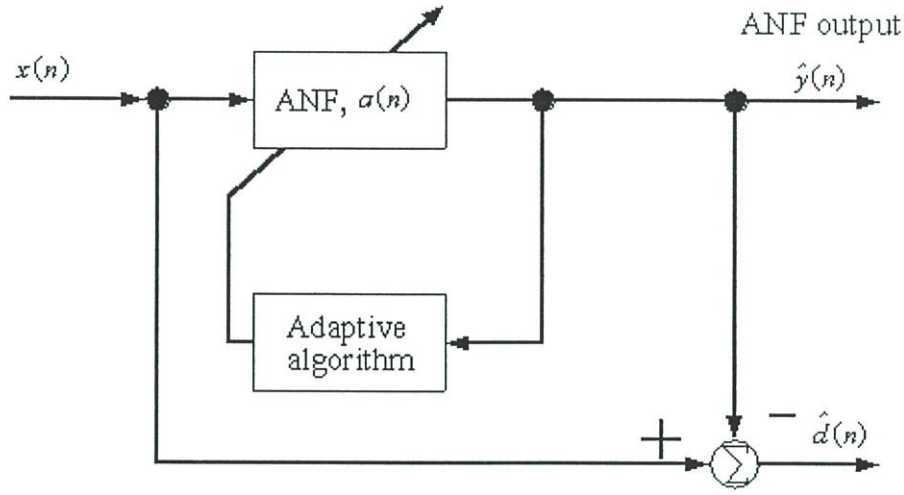
1. Pm และ Pc มีค่าสูงทำให้ GA หาคำตอบได้เร็วแต่ค่าความแปรปรวนสูงตามไปด้วย
2. การใช้เจเนติกอัลกอริทึมโดยตรงนั้นไม่สามารถหาค่าเสต็ปไซส์ที่เหมาะสมได้ ต้องอาศัยเงื่อนไขต่างๆเข้ามาช่วย เช่น การลดค่า Pm และ Pc ลงเรื่อยๆในทุกวัฏจักรดังรูป 5.8
3. เจเนติกอัลกอริทึมทำงานร่วมกับ NVSQLMP อัลกอริทึม ทำให้ได้ตัวกรองความถี่ที่มีความเร็วในการทำงานสูงและสามารถทนต่อสัญญาณรบกวนได้ดี

ส่วนการทดสอบการทำงานต่างๆ ของ GA และ NVSQLMP อัลกอริทึม นั้นจะนำเสนอในบทถัดไป

บทที่ 6
ผลการวิจัย

ในบทนี้จะแสดงผลการทำงานของอะแดปทีฟอัลกอริทึมแบบ VSQLMP [5] และ NVSQLMP อัลกอริทึมที่นำเสนอในวิทยานิพนธ์นี้ และยังรวมถึงอะแดปทีฟอัลกอริทึมที่เกิดจากการทำงานร่วมกันของ NVSQLMP อัลกอริทึมกับเจเนติกอัลกอริทึม โดยการทดสอบทั้งหมดนี้ได้อาศัยโปรแกรมภาษาซีจำลองการทำงานในส่วนของ GA และเก็บผลการทดสอบที่ได้ไว้ในรูปแบบของเท็กซ์ไฟล์ซึ่งสามารถนำไฟล์นี้ไปแสดงผลโดยใช้โปรแกรม MATLAB

ส่วนประสิทธิภาพของอะแดปทีฟอัลกอริทึมนั้นสามารถดูได้จากความเร็วในการลู่เข้าหาคำตอบและค่า MSE ของสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ กล่าวคืออะแดปทีฟอัลกอริทึมที่ดีนั้นต้องมีความเร็วในการหาคำตอบสูง ค่าMSE ต่ำ และ ทนต่อสัญญาณรบกวนได้ดี ดังที่กล่าวไว้ในบทที่ 3



รูปที่ 6.1 บล๊อกไดอะแกรม ANF

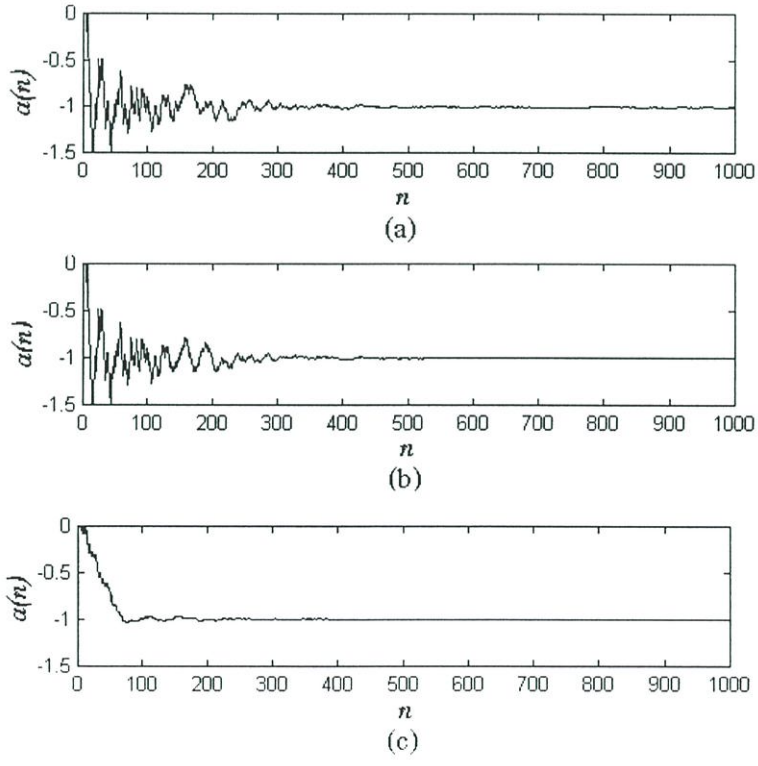
6.1 การทดสอบอะแดปทีฟอัลกอริทึมในการประมาณสัญญาณซายน์คลื่นเดี่ยว

หัวข้อนี้เป็นการทดสอบการทำงานของอะแดปทีฟอัลกอริทึมในวิทยานิพนธ์ฉบับนี้ โดยนำมาประมาณสัญญาณซายน์ที่มีสัญญาณรบกวนแบบเกาส์เซียน (White gaussian noise) หรือสัญญาณรบกวนแบบอิมพัลส์ (Impulse noise) ปะปนอยู่

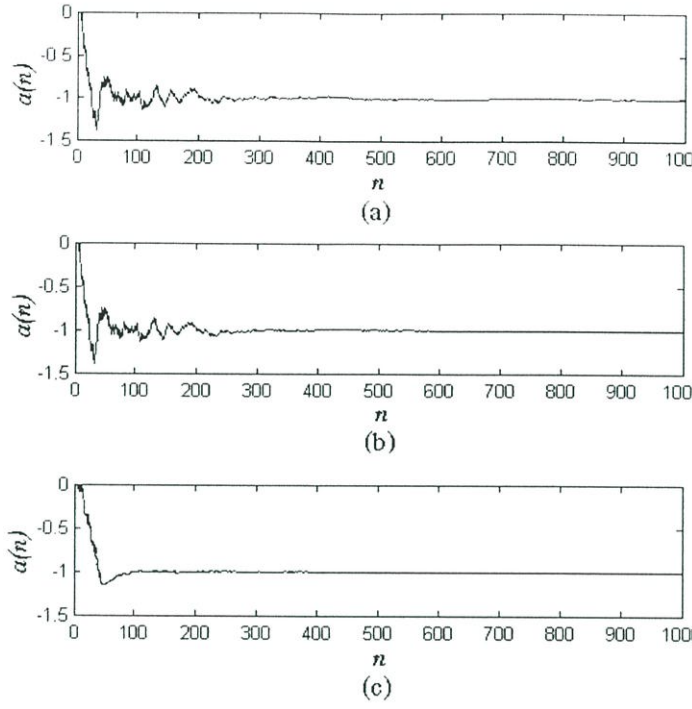
6.1.1 เป็นการทดสอบอะแดปทีฟอัลกอริทึมโดยที่มีค่า $\mu(0)$ ที่แตกต่างกัน คือ 0.2, 0.1 และ 0.02 ซึ่งมีอินพุตตามดังนี้

$$x(n) = A\sin(\omega_0 n + \phi) + v(n) \qquad (6.1)$$

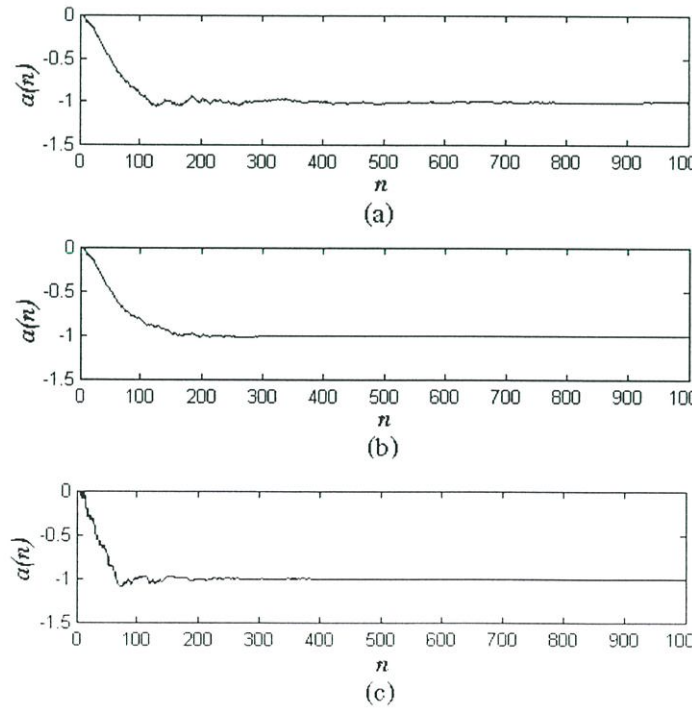
โดยที่ A คือแอมพลิจูด ω_0 คือความถี่เชิงมุม ϕ คือ เฟส (Phase) ของความถี่เชิงมุม และ $v(n)$ คือสัญญาณรบกวน กำหนดพารามิเตอร์ของ VSQLMP และ NVSQLMP มีค่าดังนี้ A = 1, ϕ = 0, $\omega_0 = \pi/3$, $\rho = 0.9$, $\gamma = 0.99$, $\alpha = 0.99$, $\lambda = 0.001$, SNR = 7dB, p = 1 และในส่วนของ GA กำหนดค่าพารามิเตอร์ดังนี้ Pm = 0.05, Pc = 0.5, Popsize = 30 และ Threshold = 0.35 โดยในรูปที่ 6.2 กำหนด $\mu(0) = 0.2$ รูปที่ 6.3 กำหนด $\mu(0) = 0.1$ รูปที่ 6.4 กำหนด $\mu(0) = 0.02$ ตามลำดับ จากรูปที่ 6.2 ถึง 6.4 แสดงให้เห็นว่าค่า $\mu(0)$ มีผลต่อ VSQLMP ทั้งสองแบบกล่าวคือ $\mu(0)$ มีค่าสูงก็จะทำให้สามารถหาคำตอบได้เร็ว แต่ถ้ามีค่าสูงมากเกินไปก็ทำให้เกิดความแปรปรวนสูงในช่วงแรกดังรูปที่ 6.2 หรือในกรณีที่ $\mu(0)$ มีค่าสูงมากอาจทำให้ไม่สามารถหาคำตอบได้เลย เนื่องจากค่า $\mu(0)$ ที่เหมาะสมสำหรับตัวกรองอะแดปทีฟแบบ IIR นั้นยังไม่มีข้อสรุปที่แน่นอนดังที่กล่าวไว้ในบทที่ 3 ดังนั้นการกำหนดค่า $\mu(0)$ จึงเป็นการลองผิดลองถูก แต่ในกรณีของ GA ที่ทำงานร่วมกับ NVSQLMP อัลกอริทึมนั้นการเปลี่ยนค่า $\mu(0)$ จะไม่ส่งผลกับการหาคำตอบ



รูปที่ 6.2 เปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ $\mu(0)$ = 0.2 โดย (a) VSQLMP อัลกอริทึม, (b) NVSQLMP อัลกอริทึม (c) NVSQLMP อัลกอริทึมทำงานร่วมกับ GA

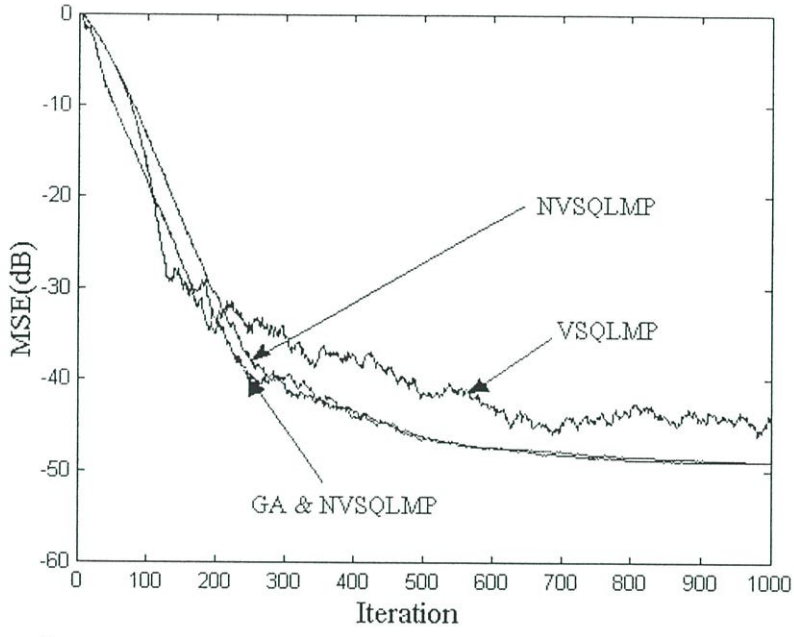


รูปที่ 6.3 เปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ ที่ $\mu(0) = 0.1$ (a) VSQLMP อัลกอริทึม, (b) NVSQLMP อัลกอริทึม (c) NVSQLMP อัลกอริทึมทำงานร่วมกับ GA

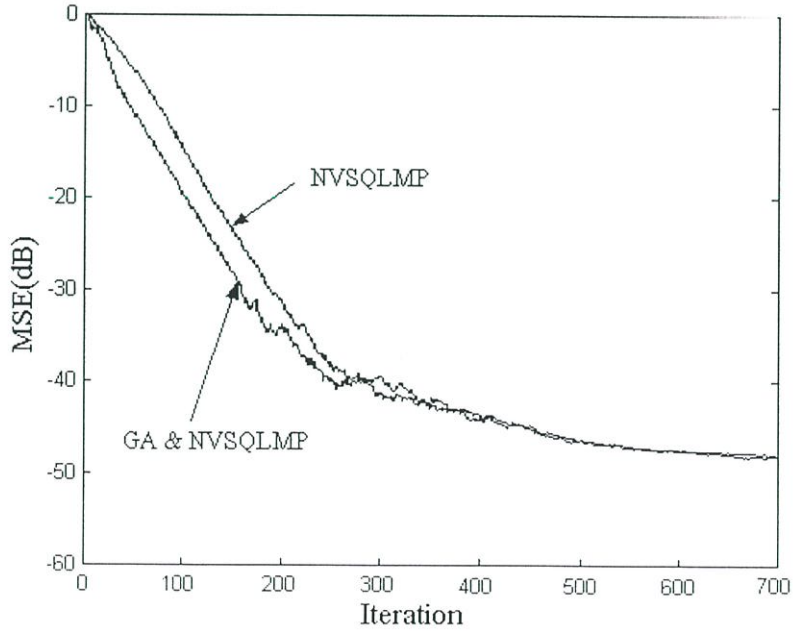


รูปที่ 6.4 เปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ ที่ $\mu(0) = 0.02$ (a) VSQLMP อัลกอริทึม, (b) NVSQLMP อัลกอริทึม (c) NVSQLMP อัลกอริทึมทำงานร่วมกับ GA

จากรูปที่ 6.4 ได้แสดงให้เห็นว่า NVSQLMP อัลกอริทึมทำงานร่วมกับ GA จะมีความเร็วในการหาคำตอบสูงกว่า VSQLMP อัลกอริทึม และ NVSQLMP อัลกอริทึม นอกจากมีความเร็วสูงแล้วค่า MSE ยังต่ำกว่าด้วย จากรูปที่ 6.4 จะสามารถแสดงค่า MSE ได้ดังรูปที่ 6.5 และ 6.6



รูปที่ 6.5 เปรียบเทียบค่า MSE ของ VSQLMP อัลกอริทึม NVSQLMP อัลกอริทึม และ NVSQLMP อัลกอริทึมทำงานร่วมกับ GA กำหนด $\mu_{(0)} = 0.02$



รูปที่ 6.6 รูปขยายเปรียบเทียบค่า MSE ระหว่าง NVSQLMP อัลกอริทึม และ NVSQLMP อัลกอริทึมทำงานร่วมกับ GA กำหนด $\mu_{(0)} = 0.02$

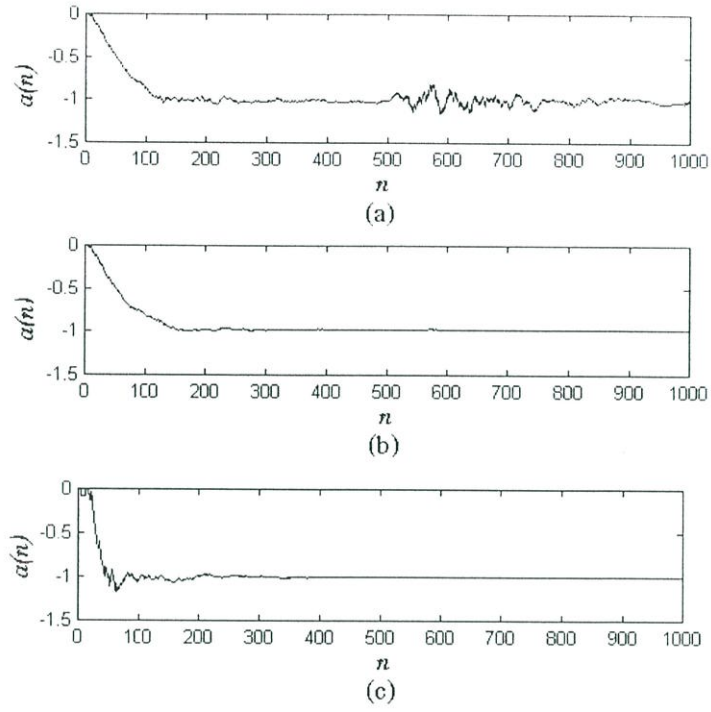
จากรูปที่ 6.5 แสดงให้เห็นว่า VSQLMP อัลกอริทึมมีค่า MSE สูงกว่า NVSQLMP อัลกอริทึม และ NVSQLMP อัลกอริทึมทำงานร่วมกับ GA ที่ $\mu(0) = 0.02$ และรูปที่ 6.6 แสดงให้เห็นว่า NVSQLMP อัลกอริทึมจะมีค่า MSE สูงกว่า NVSQLMP อัลกอริทึมทำงานร่วมกับ GA ที่ $\mu(0) = 0.02$ ในช่วงแรก โดยในช่วงนี้อัลกอริทึมแบบ NVSQLMP อัลกอริทึมทำงานร่วมกับ GA ได้ใช้ GA ที่มีค่า Pc Pm สูงในการทำงาน และหลังจากนั้นจะเห็นว่าอัลกอริทึมทั้ง 2 มีค่า MSE ที่ใกล้เคียงกันมากเนื่องจากช่วงหลังนี้เป็นการใช้อัลกอริทึมเดียวกันคือ NVSQLMP อัลกอริทึมนั่นเอง

6.1.2 การทดสอบอะแดปทีฟอัลกอริทึมในกรณีที่เกิดสัญญาณรบกวนแบบอิมพัลส์

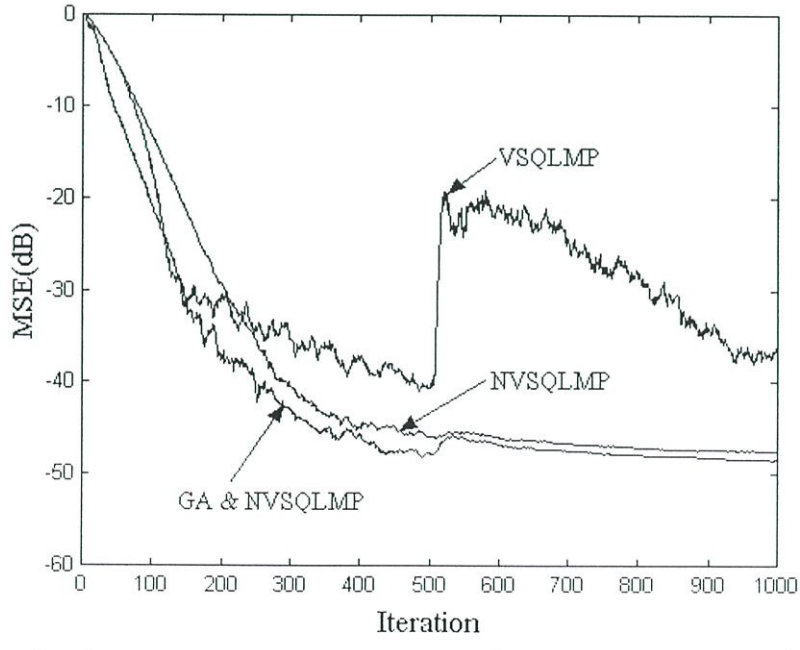
ในหัวข้อนี้จะทำการทดสอบตัวกรองสัญญาณโดยใช้อัลกอริทึมทั้ง 3 แบบ คือ VSQLMP, NVSQLMP อัลกอริทึม และ NVSQLMP อัลกอริทึมทำงานร่วมกับ GA และกำหนดอินพุตจากสมการ (6.1) คือ

$$x(n) = A\sin(\omega_0 n + \phi) + v(n)$$

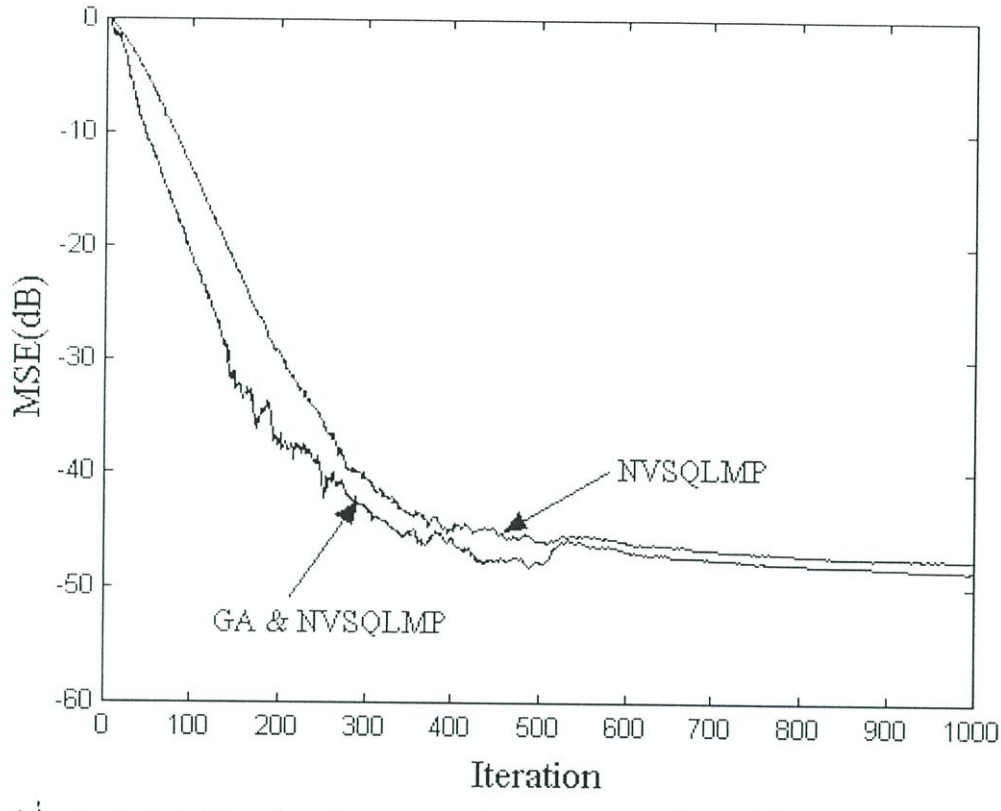
กำหนดพารามิเตอร์ของ VSQLMP และ NVSQLMP มีค่าดังนี้ $A = 1$, $\phi = 0$, $\mu(0) = 0.02$, $\omega_0 = \pi/3$, $\rho = 0.9$, $\gamma = 0.99$, $\alpha = 0.99$, $\lambda = 0.001$, SNR = 7dB, p = 1 และในส่วนของ GA กำหนดค่าพารามิเตอร์ดังนี้ Pm = 0.05, Pc = 0.5, Popsize = 30 และ Threshold = 0.35 จากนั้นเพิ่มสัญญาณรบกวนแบบอิมพัลส์เข้าไปที่ตำแหน่ง n = 500 โดยมีขนาดเท่ากับ 10 ผลที่ได้แสดงดังรูปที่ 6.7 โดยรูปที่ 6.7 (a), (b) และ (c) แสดงถึงค่า $a(n)$ ที่ได้จากอัลกอริทึมแบบ VSQLMP แบบ NVSQLMP อัลกอริทึม และ NVSQLMP อัลกอริทึมทำงานร่วมกับ GA ตามลำดับ เมื่อเกิดสัญญาณรบกวนแบบอิมพัลส์ที่ตำแหน่ง n = 500 จะเห็นว่าค่าของสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ ของ VSQLMP อัลกอริทึมนั้นเกิดความแปรปรวนสูงในขณะที่ NVSQLMP อัลกอริทึม และ NVSQLMP อัลกอริทึมทำงานร่วมกับ GA อัลกอริทึมแทบจะไม่เกิดการเปลี่ยนแปลงเลย นั่นหมายความว่า NVSQLMP อัลกอริทึม และ NVSQLMP อัลกอริทึมทำงานร่วมกับ GA อัลกอริทึมนั้นมีความทนทานต่อสัญญาณรบกวนแบบอิมพัลส์สูงนั่นเอง ส่วนเรื่องความเร็วในการหาคำตอบนั้นจะเห็นได้ว่า NVSQLMP อัลกอริทึมทำงานร่วมกับ GA อัลกอริทึมนั้นมีความมีความเร็วในการหาคำตอบได้สูงกว่า VSQLMP อัลกอริทึม และ NVSQLMP อัลกอริทึม



รูปที่ 6.7 เปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ เมื่อเกิดสัญญาณอิมพัลส์ขนาด 10 หน่วยที่ตำแหน่ง $n = 500$ (a) VSQLMP (b) NVSQLMP อัลกอริทึม และ (c) NVSQLMP อัลกอริทึม ทำงานร่วมกับ GA

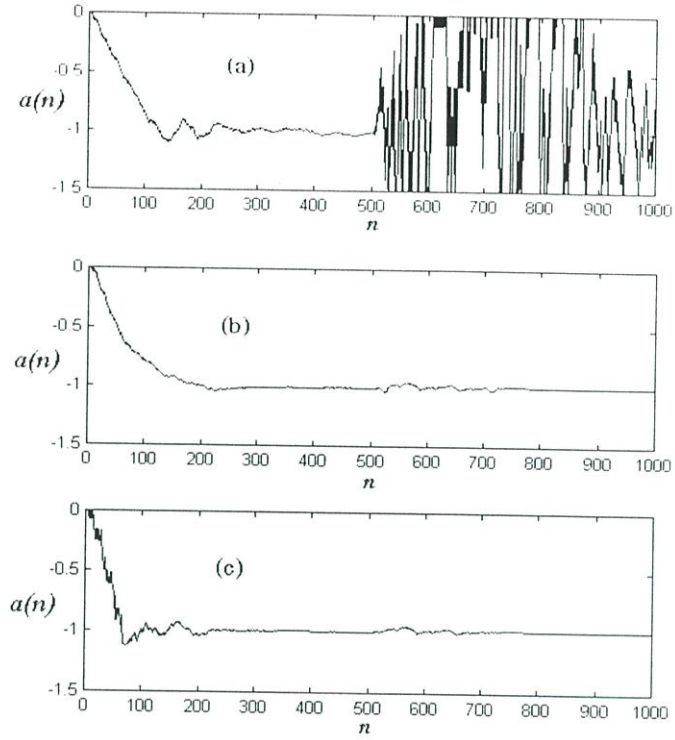


รูปที่ 6.8 เปรียบเทียบค่า MSE ระหว่าง VSQLMP อัลกอริทึม NVSQLMP อัลกอริทึม และ NVSQLMP อัลกอริทึม ทำงานร่วมกับ GA

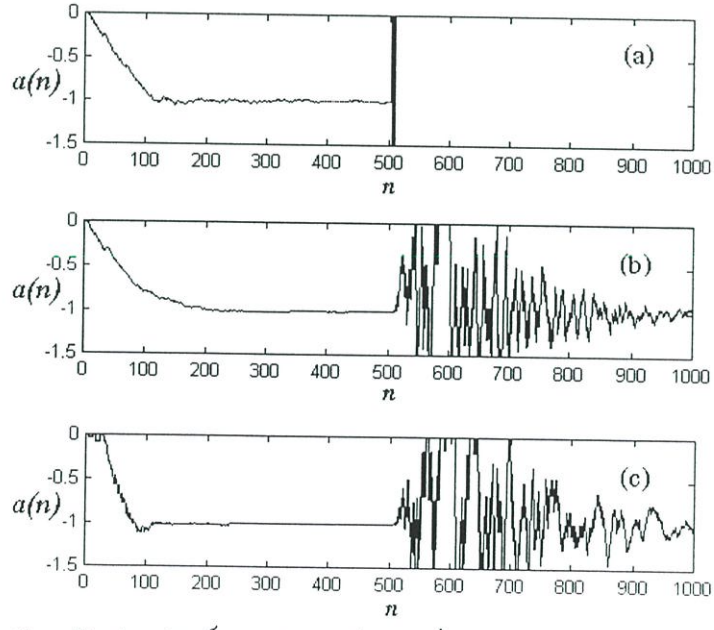


รูปที่ 6.9 รูปขยายเปรียบเทียบค่า MSE ระหว่าง NVSQLMP อัลกอริทึม และ NVSQLMP อัลกอริทึม ทำงานร่วมกับ GA

จากรูปที่ 6.8 และ 6.9 ได้แสดงค่า MSE ของแต่ละอัลกอริทึม ซึ่งแสดงให้เห็นว่า VSQLMP อัลกอริทึม นั้นมีค่า MSE ที่สูงโดยเฉพาะช่วงที่เกิดสัญญาณรบกวนแบบอิมพัลส์ นั่นหมายความว่า VSQLMP อัลกอริทึมไม่สามารถทนต่อสัญญาณรบกวนแบบอิมพัลส์ส่วน NVSQLMP อัลกอริทึม และ NVSQLMP อัลกอริทึมทำงานร่วมกับ GA นั้นสามารถทนต่อสัญญาณรบกวนแบบอิมพัลส์ได้ดี ส่วนในรูปที่ 6.9 นั้นทำให้เห็นได้ชัดเจนว่าในช่วงแรกนั้น NVSQLMP อัลกอริทึมทำงานร่วมกับ GA มีค่า MSE ต่ำกว่า NVSQLMP อัลกอริทึม ส่วนในรูปที่ 6.10 นั้นได้แสดงให้เห็นว่า ในกรณีที่เกิดสัญญาณรบกวนแบบอิมพัลส์ขนาด 30 หน่วยนั้นอาจทำให้ VSQLMP อัลกอริทึมหยุดการทำงานได้ และในกรณีที่เกิดสัญญาณรบกวนแบบอิมพัลส์สูง ๆ เช่นสัญญาณรบกวนแบบอิมพัลส์ขนาด 100 หน่วยนั้นทำให้ NVSQLMP อัลกอริทึม และ NVSQLMP อัลกอริทึมทำงานร่วมกับ GA เกือบจะหยุดการทำงานเช่นกันดังแสดงในรูปที่ 6.11



รูปที่ 6.10 เปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ เมื่อเกิดสัญญาณอิมพัลส์ขนาด 30 หน่วยที่ตำแหน่ง $n = 500$ (a) VSQLMP (b) NVSQLMP อัลกอริทึม (c) NVSQLMP อัลกอริทึมทำงานร่วม กับ GA

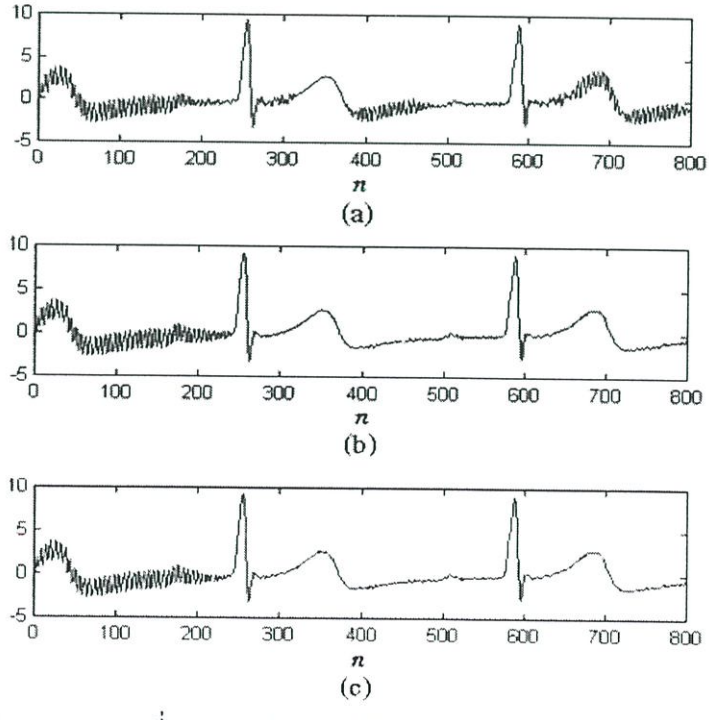


รูปที่ 6.11 เปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ เมื่อเกิดสัญญาณอิมพัลส์ขนาด 100 หน่วยที่ตำแหน่ง $n = 500$ (a) VSQLMP (b) NVSQLMP อัลกอริทึม (c) NVSQLMP อัลกอริทึมทำงานร่วม กับ GA

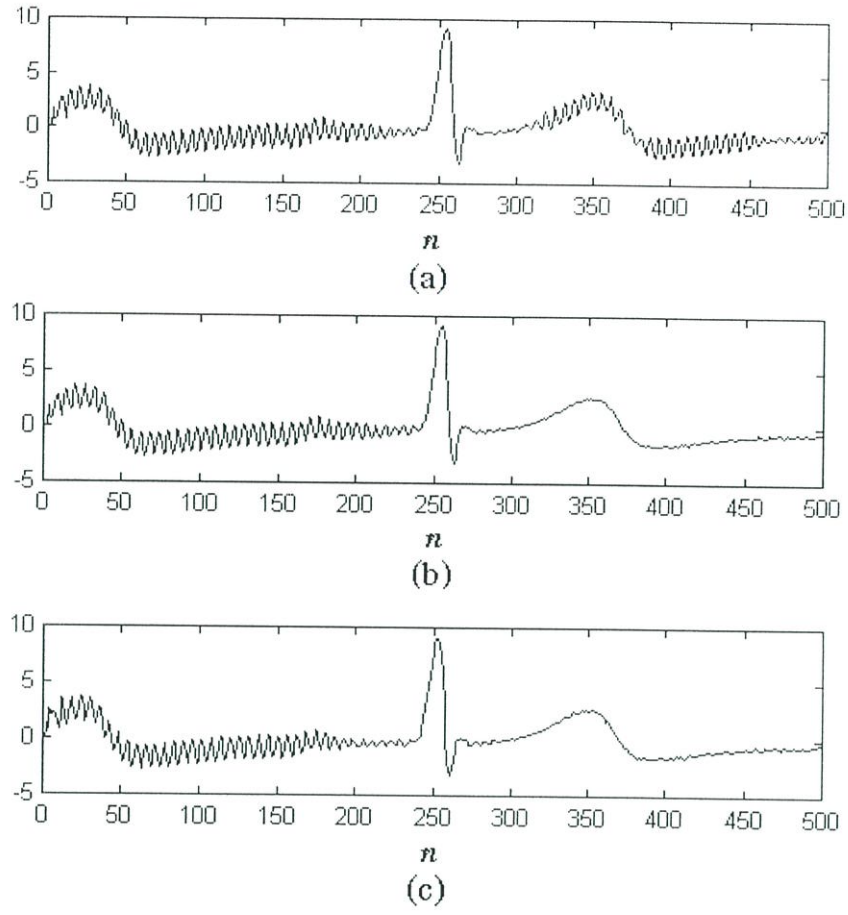
6.2 การทดสอบอะแดปทีฟอัลกอริทึมในการกำจัดสัญญาณรบกวน 50 Hz ที่ปะปนในสัญญาณ ECG

ในหัวข้อนี้จะทำการทดสอบตัวกรองสัญญาณโดยใช้อัลกอริทึมทั้ง 3 แบบคือ VSQLMP อัลกอริทึม NVSQLMP อัลกอริทึม และ NVSQLMP อัลกอริทึมทำงานร่วมกับ GA ในการทดสอบกำหนดอินพุตคือสัญญาณคลื่นหัวใจที่มีสัญญาณรบกวนคลื่นความถี่ที่เกิดจากไฟฟ้ากระแสสลับความถี่ 50 Hz ปะปนอยู่ด้วย กำหนดพารามิเตอร์ของ VSQLMP และ NVSQLMP มีค่าดังนี้ $\mu(0) = 0.05$, $\rho = 0.9$, $\gamma = 0.99$, $\alpha = 0.99$, $\lambda = 0.005$, SNR = 7dB, p = 1 และในส่วนของ GA กำหนดค่าพารามิเตอร์ดังนี้ Pm = 0.05, Pc = 0.5, Popsize = 30 และ Threshold = 0.35

จากรูปที่ 6.12 แสดงถึงสัญญาณ ECG ที่ได้จากการกำจัดสัญญาณรบกวน 50 Hz ด้วยอัลกอริทึมทั้ง 3 แบบรูปที่ 6.12 (a), (b) และ (c) แสดงถึงสัญญาณเอาต์พุตของสัญญาณคลื่นหัวใจ ที่ได้จาก VSQLMP อัลกอริทึม, NVSQLMP อัลกอริทึม และ NVSQLMP อัลกอริทึมทำงานร่วมกับ GA ตามลำดับ จากรูปจะเห็นได้ว่า VSQLMP อัลกอริทึมไม่สามารถกำจัดสัญญาณรบกวนได้หมด ส่วน NVSQLMP อัลกอริทึม และ NVSQLMP อัลกอริทึมทำงานร่วมกับ GA นั้นสามารถกำจัดสัญญาณรบกวนได้หมดไป จากรูปที่ 6.13 ได้ทำการขยายสัญญาณ ECG ที่ออกจากตัวกรองสัญญาณที่ได้จาก NVSQLMP อัลกอริทึม และ NVSQLMP อัลกอริทึมทำงานร่วมกับ GA โดยที่เงื่อนไขและพารามิเตอร์ต่าง ๆ ยังคงเหมือนรูปที่ 6.12 ซึ่งผลที่ได้ในรูปที่ 6.13 นั้นจะเห็นได้ว่า NVSQLMP อัลกอริทึมทำงานร่วมกับ GA สามารถทำงานได้เร็วกว่า NVSQLMP อัลกอริทึม



รูปที่ 6.12 แสดงสัญญาณ ECG ที่ออกจากตัวกรองสัญญาณ (a) VSQLMP อัลกอริทึม (b) NVSQLMP อัลกอริทึม และ (c) NVSQLMP อัลกอริทึมทำงานร่วมกับ GA



รูปที่ 6.13 รูปขยายสัญญาณ ECG ที่ออกจากตัวกรองสัญญาณ (a) VSQLMP อัลกอริทึม (b) NVSQLMP อัลกอริทึม และ (c) NVSQLMP อัลกอริทึมทำงานร่วมกับ GA

6.3 สรุป

ในบทนี้ได้นำเสนอผลการทดสอบอัลกอริทึมต่าง ๆ สำหรับตัวกรองสัญญาณแบบ ANF เพื่อประมาณสัญญาณซายน์ที่ไม่ทราบความถี่โดยในการทดสอบแรกคือการนำอัลกอริทึมทั้ง 3 แบบ คือ ที่ VSQLMP อัลกอริทึม NVSQLMP อัลกอริทึม และ NVSQLMP อัลกอริทึมทำงานร่วมกับ GA ใช้สำหรับการกำจัดสัญญาณรบกวนประเภทเกาส์เซียน ซึ่งผลที่ได้นั้นจะเห็นว่า NVSQLMP อัลกอริทึมทำงานร่วมกับ GA มีคุณสมบัติที่ดีกว่า VSQLMP อัลกอริทึม กล่าวคือในด้านความเร็วในการหาคำตอบสูงกว่าและค่า MSE ต่ำกว่าด้วย นอกจากนี้การเปลี่ยนแปลงค่า $\mu(0)$ ยังไม่มีผลต่อการทำงานของ NVSQLMP อัลกอริทึมทำงานร่วมกับ GA อีกด้วย การทดสอบเมื่อเกิดสัญญาณรบกวนแบบอิมพัลส์ผลที่ได้คือ NVSQLMP อัลกอริทึม และ NVSQLMP อัลกอริทึมทำงานร่วมกับ GA นั้นมีความทนทานต่อสัญญาณรบกวนแบบอิมพัลส์ได้ดีกว่า VSQLMP อัลกอริทึม สุดท้ายคือการทดสอบโดยนำตัวกรองประยุกต์ไปใช้สำหรับกำจัดสัญญาณรบกวนคลื่นความถี่ 50 Hz ที่ปะปนหรือรบกวนสัญญาณคลื่นหัวใจ จะเห็นได้ว่า NVSQLMP อัลกอริทึม และ NVSQLMP อัลกอริทึมทำงานร่วมกับ GA นั้นสามารถกำจัดสัญญาณรบกวนได้ดีกว่า จากผลที่ได้ทำการ

ทดสอบอัลกอริทึมทั้ง 3 แบบนั้นสรุปได้ว่า อัลกอริทึมที่ใช้การทำงานร่วมกัน NVSQLMP อัลกอริทึมทำงานร่วมกับ GA นั้นจะมีคุณสมบัติที่ดีกว่า VSQLMP อัลกอริทึมในด้านความเร็วในการหาคำตอบ ค่า MSE ต่ำกว่า และมีความทนทานต่อสัญญาณรบกวนแบบอิมพัลส์อีกด้วย

บทที่ 7

สรุปผลการทดลอง และข้อเสนอแนะ

ในวิทยานิพนธ์นี้ได้นำเสนออะแดปทีฟอัลกอริทึมใหม่ 2 แบบ คือ NVSQLMP อัลกอริทึม และ อัลกอริทึมที่เกิดจากการทำงานร่วมกันของ NVSQLMP อัลกอริทึมกับ GA โดยในแบบแรก คือ NVSQLMP อัลกอริทึมนั้น ขั้นตอนการทำงานต่าง ๆ ยังคล้ายคลึงกับ VSQLMP อัลกอริทึม แต่ว่ามีข้อได้เปรียบคือ มีค่า MSE ต่ำกว่าและความทนทานต่อสัญญาณรบกวนแบบอิมพัลส์ได้ดีกว่า เนื่องจากได้ใช้เกรเดียนต์ของเอาต์พุตเข้าไปช่วยลดพลังงานของเอาท์พุตของอะแดปทีฟฟิลเตอร์ จึงทำให้ค่าเสต็ปไซส์ ไม่เปลี่ยนแปลงมากเกินไป ซึ่งทำให้ค่า $\alpha(n)$ นั้นจะไม่เกิดความแปรปรวนสูงเหมือนกับ VSQLMP อัลกอริทึม ส่วนอัลกอริทึมแบบที่ 2 นั้นคือ NVSQLMP อัลกอริทึมทำงานร่วมกับ GA นี้มีข้อดีตรงที่มีความเร็วสูงกว่า VSQLMP อัลกอริทึมและ NVSQLMP อัลกอริทึม นอกจากนี้ยังมีค่า MSE ที่ต่ำ และทนทานต่อสัญญาณรบกวนแบบอิมพัลส์ได้ดี เหตุที่อัลกอริทึมในวิทยานิพนธ์นี้มีความเร็วสูงก็เพราะใช้ GA ในการเริ่มต้นหาค่าเสต็ปไซส์ ก่อนโดยที่ใช้ GA ที่มี Pm และ Pc สูง ดังนั้นค่าเสต็ปไซส์ ที่ได้จะทำให้หาคำตอบได้เร็ว แต่ทว่า GA ที่มี Pm และ Pc สูงนั้นจะมีข้อเสียตรงที่ทำให้เกิดความแปรปรวนสูงไปด้วย ดังนั้นจึงนำ NVSQLMP อัลกอริทึมเข้ามาทำงานต่อจาก GA ซึ่งจะทำให้เกิดผลดีต่ออัลกอริทึม คือค่า MSE ต่ำ และทนต่อสัญญาณรบกวนแบบอิมพัลส์ได้ดี เปรียบเสมือนการรวมข้อดีของ GA และ NVSQLMP อัลกอริทึมเข้าด้วยกันนั่นเอง

จากผลการทดลองที่ใช้โปรแกรม MATHLAB และ ภาษา C++ นั้นก็จะเห็นได้ว่า การประมาณค่าสัญญาณซายน์คลื่นเดี่ยวที่ถูกสัญญาณรบกวนแบบเกาส์เซียน และสัญญาณรบกวนอิมพัลส์ อัลกอริทึมที่เสนอในวิทยานิพนธ์นี้สามารถลู่เข้าหาคำตอบได้ดีกว่า มีความทนทานมากกว่า และ ค่า MSE ต่ำกว่าอีกด้วย ส่วนในกรณีที่นำอะแดปทีฟนอตช์ฟิลเตอร์ไปประยุกต์ใช้กับการกำจัดสัญญาณรบกวน 50 Hz ที่เกิดจากการเหนี่ยวนำของไฟฟ้ากระแสสลับที่ปะปนมากับสัญญาณคลื่นหัวใจ (ECG) ในขณะที่ทำการบันทึก อัลกอริทึมที่เสนอนี้สามารถกำจัดสัญญาณรบกวนได้ดีเช่นกัน

ข้อสังเกต ค่าเสต็ปไซส์เริ่มต้น ($\mu(0)$ หรือ $\mu_{\max}$) นั้นจะมีผลต่ออัลกอริทึมแบบ VSQLMP เป็นอย่างมาก ซึ่งในการทดสอบจะเห็นว่า ถ้าค่าเสต็ปไซส์เริ่มต้น มีค่ามากจะทำให้อัลกอริทึมหาคำตอบได้เร็วขึ้น แต่ในกรณีที่มีค่าเสต็ปไซส์เริ่มต้น มากเกินไปจะทำให้เกิดความแปรปรวนสูง ในวิทยานิพนธ์นี้ยังไม่มีการกล่าวถึงวิธีการเลือกค่าเสต็ปไซส์เริ่มต้น ที่เหมาะสม แต่ในส่วนของ GA นั้นไม่จำเป็นต้องมีค่าเสต็ปไซส์เริ่มต้น ก็ได้เนื่องจาก ค่าเสต็ปไซส์เริ่มต้น ชุดแรกของ GA นั้นได้มาจากการสุ่มค่าเสต็ปไซส์ ในช่วงที่กำหนดไว้ และจากการทดสอบ GA นั้น จะหาคำตอบได้เร็วหรือช้าจะขึ้นอยู่กับค่า Pm และ Pc นอกจากค่า Pm และ Pc แล้ว ยังมีอีกส่วนที่สำคัญคือ การกำหนด

ฟังก์ชันความเหมาะสมของ GA อีกด้วย นอกจากนี้ GA ยังสามารถประยุกต์ใช้งานได้อีกมาก เพราะการทำงานของ GA นั้นขึ้นอยู่กับการสุ่มตัวเลข ซึ่งเป็นส่วนที่ต้องศึกษาและพัฒนาต่อไป ส่วนข้อเสียของ GA คือ การคำนวณซึ่งการคำนวณนั้นจะมากกว่าอัลกอริทึมแบบ VSQLMP เป็นจำนวนเท่าของโครโมโซม เพราะว่าต้องนำโครโมโซมทุกตัวมาหาค่าความเหมาะสมก่อนด้วย เนื่องจากทฤษฎีเกี่ยวกับตัวกรอง IIR ยังอยู่ในขั้นของการวิจัย ยังไม่มีสูตรสำเร็จเหมือน FIR เพราะ IIR นั้นทำงานในลักษณะ Nonlinear ซึ่งทำนายพฤติกรรมได้ค่อนข้างยาก ต้องอาศัยการประมาณจึงต้องใช้ความรู้ด้านคณิตศาสตร์ และประสบการณ์ ดังนั้นในการวิเคราะห์ IIR นั้นจะต้องนำคอมพิวเตอร์เข้ามาช่วยในการจำลองการทำงานเช่นเดียวกับอัลกอริทึมที่ได้นำเสนอในวิทยานิพนธ์นี้

เอกสารอ้างอิง

[1] Phuvasitkul S., Benjangkaprasert C., Anantrasirichai N., "A New Class of Gradient-Based Algorithm by Using Variable Step-Size Technique", Proceedings of 6th International Conference on Signal Processing, Sep. 2002 , pp. 85-88.

[2] Phuvasitkul S., Benjangkaprasert C., Anantrasirichai N., Jorphochaudom S., "A New Gradient-Based Algorithm Using Variable Step-Size Technique and Its Application", Proceeding of IEEE APCCAS, Oct. 2002, pp.309-312.

[3] S.C. Pei and C.C. Tseng, "Adaptive IIR Notch Filter Based on Least Mean *p*-Power ErrorCriterion", IEEE Transaction on circuits and systems-II Analog and digital signal Processing, vol. 40, No.8, Aug, 1993, pp. 525-529.

[4] S.C. Ng, S.H. Leung, C.Y. Chung, A. Luk and W.H. Lau, "The Genetic Search Approch A New Learning Algorithm for Adaptive IIR Filtering" IEEE Signal Processing Magazine, Nov 1996, pp. 38-46.

[5] ราชู พันธ์ฉลาด, 2543. "อะแดปทีฟอัลกอริทึมสำหรับ IIR นอตช์ฟิลเตอร์ ด้วยวิธีเกรเดียน และการประยุกต์ใช้งาน." วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า คณะบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

[6] กิติ ไพฑูรย์วัฒนกิจ., กาญจนี วงศ์วิภาพร, *การปรับปรุงสมรรถนะ จีเนติก อัลกอริทึม*, วารสารสำนักงานคณะกรรมการวิจัยแห่งชาติ, ปีที่ 27, เล่มที่ 1, มกราคม-มิถุนายน 2538, หน้า 1-33.

[7] Monson H. Hayes, **Statistical Digital Signal Processing and Modeling**, Canada : John Wiley & Sons, 1996.

[8] Alan V. Oppenheim, Ronald W. Schafer, **Digital Signal Processing**, Prentice-Hall, 1975.

[9] R. M. Mersereau, M. J. T. Smith, **Digital Filtering A Computer Laboratory Textbook**, John Wiley & Sons, 1994, pp. 59-61.

[10] Robert D. Strum, Donald E. Kirk **First Principles of Discrete System and Digital Signal Processing.** Addison-Wesley. 1988.

[11] Lonnie C. Ludeman. **Fundamentals of Digital Signal Processing.** John wiley & Sons. 1987.

[12] John G. Proakis, Dimitris G. Manolakis, **Digital Signal Processing Principles, Algorithm, and Applications,** Prentice-Hall, 1996.

[13] Russell M. Mersereau, Mark J.T. Smith, **Digital Filtering,** John Wiley & Sons, 1994.

[14] E. C. Ifeachor and B. W. Jervis, **Digital Signal Processing A Practical Approach**, Addison-Wesley Publishing Company. 1996. pp.255-257.

[15] A. V. Oppenheim, R. W. Schafer, **Discrete-Time Signal Processing,** Prentice-Hall, 1999.

[16] A. Anchorai, "A minimun parameter adaptive notch filter with constrained poles and zeros", IEEE Trans. Acoustics, Speech, and Signal Processing, vol. ASSP-33, no. 4, Aug. 1985, pp. 983-996.

[17] T. S. Ng, "Some aspects of an adaptive digital notch filter with constrained poles and zeros.", IEEE Trans. Acoustics, Speech, and Signal Processing, vol. ASSP-35, no. 2, Feb. 1979, pp.158-161.

[18] P. Stoica and Nehorai, " Performance analysis of an adaptive notch filter with constrained poles and zeros.", Acoustics, Speech, and Signal Processing, vol. ASSP-36, no. 6, June, 1988, pp.911-919.

[19] Dimitris G. Manolakis, Vinay K. Ingle, Stephen M. Kogon, **Statistical and Adaptive Signal Processing,** McGRAW-HILL, 2000.

[20] C. S. Burrus and J. A. Barreto, "Least p-Power error design of FIR filters.", Proc. IEEE int. Symp. Circuits and Systems, May 1992, pp. 545-548.

[21] J. Schrooeder, Rao Yarlagadda, J. Hershey, " L_p normed minimization with applications to linear predictive modeling for sinusoidal frequency estimation.", Signal Processing, vol. 24, Aug. 1991, pp. 193-216.

[22] Soo-Chang pei and chien-Cheng Tseng, "Adaptive IIR Notch Filter Based on Least Mean p-Power Error Criterion.", IEEE Trans. Circuits syst. vol. 40, Aug. 1993, pp. 525-529.

[23] L. Ljung, "Asymptotic gain and search direction for recursive identification algorithms.", Proc. 19th IEEE Conf. Dec. 1980.

[24] B. Widrow, Samuel D. Stearns, **Adaptive Signal Processing,** New Jersy : Prentice-Hall, 1985.

[25] S. Haykin, **Adaptive Filter Theory,** Prentice-Hall, Englewood Cliffs, N. J. 1987.

[26] J. Shynk, "Adaptive IIR filtering.", IEEE ASSP Magazine, April 1989, pp. 4-21.

[27] ไพศาล เหล่าสุวรรณ, พันธุศาสตร์, ไทยวัฒนาพานิช, 2535.

[28] ชิดชนก เหลือสินทรัพย์, คอมพิวเตอร์กับการแก้ปัญหาโดยวิธีพันธุกรรม, ไมโครคอมพิวเตอร์, ฉบับที่ 97 หน้า 192-198.

[29] Holland, J. H., *Genetic Algorithms*, Scientific American, July 1992, pp. 45-50.

[30] Michaleuiez, Z., *Genetic Algorithms + Data Structures = Evolutionary Programs*, Springer-Verlag, Berlin, 1992.

[31] Morrow, M., *Genetic Algorithms*, Dr. Dobb's Journal April 1991, pp. 26-32.

ภาคผนวก

ภาคผนวก ก.
บทความวิจัยที่ได้รับการตีพิมพ์

[1] Phuvasitkul S., Benjangkaprasert C., Anantrasirichai N., "A New Class of Gradient-Based Algorithm by Using Variable Step-Size Technique", Proceedings of 6th International Conference on Signal Processing, pp. 85-88, Sep. 2002.

[2] Phuvasitkul S., Benjangkaprasert C., Anantrasirichai N., Jorphochaudom S., "A New Gradient-Based Algorithm Using Variable Step-Size Technique and Its Application", Proceeding of IEEE APCCAS, pp.309-312, Oct. 2002.

APCCAS 2002
Asia-Pacific Conference on Circuits and Systems
Bali - Indonesia

PROCEEDINGS

PROCEEDINGS

VOL 1

October 28th - 31st

APCCAS 2002

Asia-Pacific Conference on Circuits And Systems

Denpasar, Bali - Indonesia
Discovery Kartika Plaza

IEEE Catalog Number : 02EX636

A NEW GRADIENT-BASED ALGORITHM USING VARIABLE STEP-SIZE TECHNIQUE AND ITS APPLICATION

Chawalit Benjangkaprasert, Sarinporn Jorphochaudom, Sittisak Phuvasitkul, Noppin Anantrasirichai

Research Center for Communications and Information Technology (ReCCIT), and
Department of Information Engineering, Faculty of Engineering,
King Mongkut's Institute of Technology Ladkrabang,
Chalongkrung Rd., Ladkrabang District, Bangkok 10520, Thailand

Email: kbchawal@kmitl.ac.th and pikajuz@lycos.com

ABSTRACT

In this paper, a new class of gradient-based algorithm by using variable step-size technique for a second-order adaptive IIR notch filter is presented. An adaptation step-size parameter for the algorithm is worked by the output signal and the gradient signal. The adaptive algorithm is used to detection of a sinusoid with additive white Gaussian noise, impulse noise and cancel 50-Hz interference in the recording of electrocardiograms signal. The performance of this algorithm proved here that gives high convergence speed, high impulse noise robustness and good efficiency of cancellation 50-Hz. Finally, the results of computer simulation are given to demonstrate the performance of the proposed algorithm for adaptive IIR notch filter.

1. INTRODUCTION

Digital notch filters have many applications such as the detection of sinusiods in noise or eliminating sinusoidal disturbances in communication and others. The second-order adaptive notch filters have been used for the elimination or enhancement of sine wave components with unknown frequencies[1]-[7]. For adaptive IIR notch filter (ANF) analysis, a good algorithm requires fast convergence speed and low variance and bias [1]. One approaches of using the adaptive algorithm of quantize least mean *p*-power error criterion (QLM*p*) and variable step-size quantize least mean *p*-power error criterion (VSQLM*p*) were proposed in references[3][4], respectively. However, this algorithm performs high variance.

In this paper, we have modified the algorithm by using the variable step-size was proposed in reference[4]. The new algorithm increases the noise robustness and reduces the computational complexity when compare with the previous one. Assuming a sinusoidal signal with Gaussian white noise as input signal of the second-order adaptive IIR notch filter, the filter is analyzed for various parameters from the derivation of the transfer function. This algorithm is used to estimate frequency of the sinusoidal embedded in impulse noise and cancel 50-Hz interference in the recording of electrocardiograms (ECG) signal. The simulation results for several situations are tested by using the computer programs to show that the proposed algorithm is an attractive adaptive algorithm.

2. A NEW VARIABLE STEP-SIZE ALGORITHM

In this section, we first describe the QLM*p* algorithm that was proposed in ref. [3] and then the proposed algorithm. To simplify, assuming that the input signal to the ANF is a single sinusoidal corrupted by Gaussian white noise, i.e.,

$$x(n) = A\sin(\omega_0 n + \phi) + v(n), n = 0,1,2,... \qquad (1)$$

where A is the amplitude, ϕ is the phase, ω_0 is an unknown frequency, n is time index and $v(n)$ is the zero mean unit variance white Gaussian noise, the input signal to noise ratio (SNR) is in the SNR (dB) = $10\log A^2/2\sigma^2$ form (σ^2 = noise variance). The objective is to estimate unknown frequency, ω_0 by applying $x(n)$ to the ANF, the transfer function of the second-order adaptive IIR notch filter is given by:

$$H(z) = \frac{1 + a(n)z^{-1} + z^{-2}}{1 + \rho a(n)z^{-1} + \rho^2 z^{-2}} \qquad (2)$$

where $a(n)$ is the filter coefficient or notch parameter, which should converge to $-2\cos\omega_0$ to reject a sinusoid with frequency ω_0 , ρ is the pole zero contraction factor

0-7803-7690-0/02/$17.00 ©2002 IEEE

which is to unity to ensure the narrow notch filter. The estimate output signal of ANF can be express as:

$$y(n) = x(n) + a(n)x(n-1) + x(n-2) - a(n)\rho y(n-1) - \rho^2 y(n-2) \quad (3)$$

The LMp criterion of the ANF can be expressed in terms of notch parameter " a (n)" i.e.,

$$J(a) = E(|y(n)|^p) \quad (4)$$

Eq. (4) is called "Cost function", the ensemble average value can be use to replace the expectation value in Eq. (4) that is:

$$\hat{J}(a) = \frac{1}{N}\sum_{n=0}^{N-1}|y(n)|^p \quad (5)$$

where $\hat{J}(a)$ is estimated version of $J(a)$, to simplify and reduce the computational complexity, we will set the value of p equal to one. Thus, the Eq. (5) can be rewritten as follows:

$$\hat{J}(a) = \frac{1}{N}\sum_{n=0}^{N-1}|y(n)| \quad (6)$$

The QLMp attempts to adjust notch parameter a (n) as long as minimum magnitude output signals. Hence, adaptive algorithm can be express as:

$$a(n+1) = a(n) - \mu \cdot \mathrm{sgn}\left(\frac{\partial|y(n)|^p}{\partial a(n)}\right) \quad (7)$$

where μ is the step-size constant parameter used to control the speed of convergence of an algorithm and $sgn(x) = x/|x|$.

Since

$$|y(n)| = sgn(y(n)) \cdot y(n) \quad (8)$$

substituting Eq. (8) into Eq.(7), where $p = 1$, we get

$$a(n+1) = a(n) - \mu \cdot \mathrm{sgn}\left[\mathrm{sgn}(y(n) \cdot \frac{\partial y(n)}{\partial a(n)}\right] \quad (9)$$

and by pseudolinear assumption, we have

$$\frac{\partial y(n)}{\partial a(n)} \approx x(n-1) - \rho y(n-1) \quad (10)$$

It is evident from Eq. (9) providing that we choose value of p equal to one, the QLMp algorithm will be low complicating in the calculation.

Next, if the notch parameter of update recursion is large when the algorithm is far from the optimum that give the fast convergence speed in the initial and decreasing as we approach the optimum that give low mean square error (MSE). To approach this objective is to ensure large $\mu(n)$ step-size parameter when the algorithm is far from the optimum and $\mu(n)$ decreasing as we approach the optimum. The proposed algorithm achieves the objective by using an estimate of energy of output signal and gradient signal of the ANF to control step-size update. The estimate is a time average signal that is described as

$$\Psi(n) = \alpha \cdot \Psi(n-1) + (1-\alpha)y(n)\frac{\partial y(n)}{\partial a(n)} \quad (11)$$

where α is positive constant, the range of values for α between 0 and 1. This constant is called an exponential weighting parameter that governs the averaging time constant, i.e., the quality of the estimation.

In the early stages of adaptation, the estimate of output signal energy $\Psi(n)$ is large, resulting in a large $\mu(n)$. As we approaches the optimum, the output signal approach zero, resulting in a smaller step-size parameter. This provides the fast convergence due to large initial $\mu(n)$ while ensuring low steady-state MSE due to the small final $\mu(n)$. Thus, the proposed step-size update is given by:

$$\mu(n+1) = \gamma \cdot \mu(n) + \lambda\Psi^2(n) \quad (12)$$

where γ is positive constant, the range of values for γ between 0 and 1, for the quality of the estimation should be close to unity and λ is small positive constant. Thus, the adaptation step-size is adjusted using the energy of the instantaneous output signal. The notch parameter update recursion is changed in Eq. (9), a variable $\mu(n)$ instead of a fixed μ. The proposed algorithm is as follow:

$$a(n+1) = a(n) - \mu(n) \cdot \mathrm{sgn}\left[\mathrm{sgn}(y(n)) \cdot \frac{\partial y(n))}{\partial a(n)}\right] \quad (13)$$

3. THE SIMULATION RESULTS

3.1 Sinusoidal estimation in Gaussian white noise and Impulse noise

In this simulation, the parameters chosen are $A = 1$, $\omega_0 = \pi/3$, $\alpha = 0.99$, $\gamma = 0.99$, $\lambda = 0.001$, $p = 1$, $\mu_{max}(0) = 0.02$, $a(0) = 0$, SNR = 6 dB for Gaussian noise environment and additive $50\delta(n-500)$ for impulse noise. Figure 1 shows the learning curves of notch parameter $a(n)$. As indicated in curves of Figure 1, the convergence rate and noise robustness of the proposed algorithm is better than the previous one.

In Figure 2, shows the MSE of the algorithms. From Figure 2 shows that the proposed algorithm yields significantly MSE improvement over the previous algorithm in Ref.[4].

3.2 50-Hz interference canceling in ECG signal

A major problem in the recording of ECG signal is that the measurement signals degraded by additive 50-Hz power line interference. The method used to canceling 50-Hz interference in the ECG signal is ANF.

Figure 3 (a) shows an ECG waveform with excessive amount of 50-Hz interference. Figure 3 (b) and 3 (c) show the output waveform from an ANF [4] and proposed algorithm respectively. Test condition are $\alpha = 0.99$, $\gamma = 0.99$, $\lambda = 0.001$, $p = 1$, $\mu_{max}(0) = 0.02$, $a(0) = 0$ and SNR = 10 dB. From Figure 3(c), we see that the proposed algorithm gives a good cancellation of 50-Hz ECG interference.

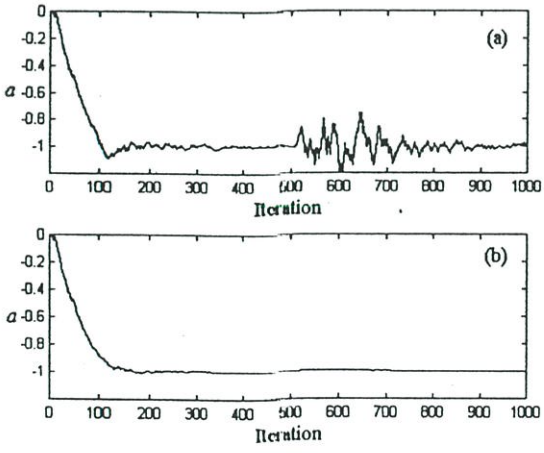


Figure 1 (a) The learning curves of coefficient $a(n)$ by Ref.[4] algorithm
(b) The learning curves of coefficient $a(n)$ by the new algorithm

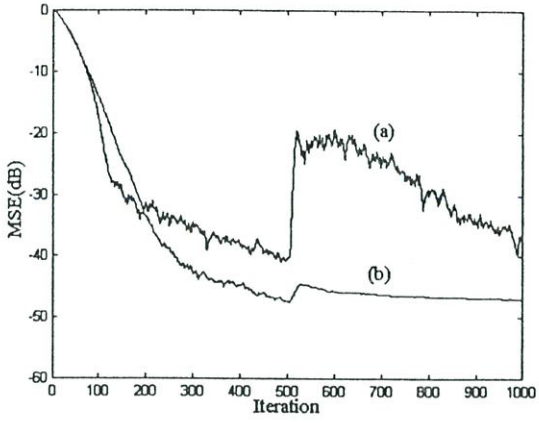


Figure 2 (a) The MSE curves of coefficient $a(n)$ by Ref. [4] algorithm
(b) The MSE curves of coefficient $a(n)$ by the new algorithm

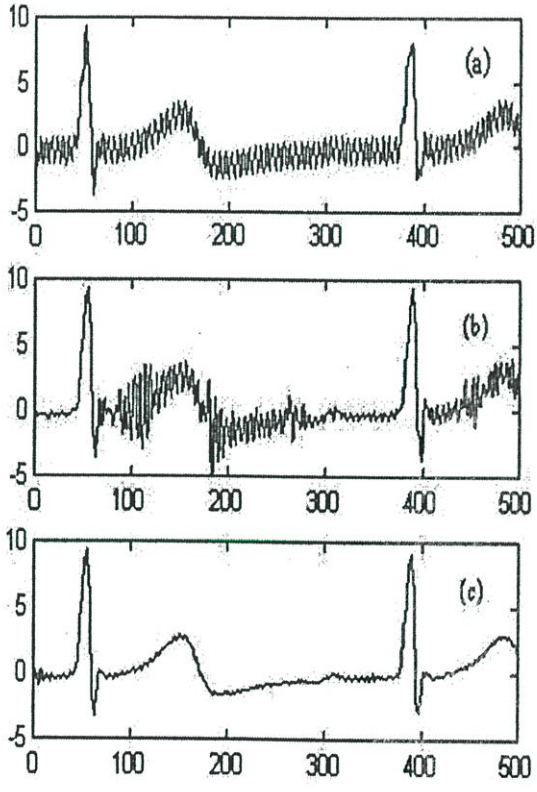


Figure 3 (a) Shows an ECG waveform with excessive amount of 50-Hz interference
(b) The output signal using Ref.[4] algorithm
(c) The output signal using the new algorithm

4. CONCLUSIONS

We have proposed a new gradient-based algorithm for a second-order IIR adaptive notch filter. From the results of applications, the convergence rate, the noise cancellation and accuracy of the proposed algorithm are better than does the previous one. The result of simulation by computer can show the proposed algorithm is consistent. In addition, it is of interesting to analyze the theoretically performance. This work will be studied and investigated in the future.

REFERENCES

[1] Simon Haykin,"Adaptive Filter Theory", Prentice-Hall International.Inc, 1996.

[2] Soo-Chang Pei and Chien-Cheng Tseng, "Adaptive IIR notch filter based on least mean *p*-power error criterion", IEEE Transaction on circuits and systems-II Analog and digital signal processing, pp. 525-529, Vol. 40, No.8, Aug, 1993.

[3] Rachu Punchalard, Chawalit Benjangkaprasert, Kanok Jenjerapongvej and Prawit chumchu , " Adaptive IIR notch filter based on least mean *p*-power error criterion by using quantized gradient technique", 21st Electrical Engineering Conference (EECON-21), pp.122-125, Nov. 1998.

[4] Rachu Punchalard, Prawit chumchu, Chawalit Benjangkaprasert, Noppin Anantrasirichai, Ornlarp Sangaroon and Kanok Jenjerapongvej, "The reduction of gradient noise in gradient – based algorithm by using variable step-size technique", IEEE Asia-Pacific conference on Circuit and System, pp. 415-418, Dec. 2000.

[5] D.V.B. Rao and S.Y. Kung, "Adaptive notch filtering for the retrieval of sinusoids in noise", IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-32, pp.791-802, Aug. 1984.

[6] K. Martin and M. T. Sun, "Adaptive filters suitable for real-time spectral analysis", IEEE Trans. Circuits Syst., vol. CAS-33, pp.218-229, Feb. 1986.

[7] S. Nishimura, J. K. Kim, and K. Hirano, "Mean-squared error analysis of an adaptive notch filters", Proc. of IEEE ISCAS, pp.732-735, May 1989.

ประวัติผู้เขียน

นายสิทธิศักดิ์ ภูวสิทธิ์กุล เกิดเมื่อวันที่ 27 มกราคม 2520 ที่จังหวัดกรุงเทพฯ สำเร็จการศึกษาวิศวกรรมศาสตรบัณฑิต สาขา วิศวกรรมไฟฟ้า จากมหาวิทยาลัย ศรีนครินทรวิโรฒ ปีการศึกษา 2542 และเข้าศึกษาต่อในหลักสูตร วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า ที่สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2543