

อัลกอริทึมแบบใหม่สำหรับอะแดปทีฟ HR Notch Filter
โดยใช้อัลกอริทึมแบบเจเนติกควบคู่กับอัลกอริทึมแบบปรับเสถียร
และการประยุกต์ใช้งาน

A NEW ALGORITHM FOR ADAPTIVE HR NOTCH FILTER
USING THE COMBINATION OF GENETIC ALGORITHM IN PARALLEL
WITH VARIABLE STEP-SIZE ALGORITHM AND ITS APPLICATIONS

สรินัทพร จอโภชาอุดม
SARINPORN JORPHOCHAUDOM

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของวารสารตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2550

KMITL-2007-EN-M-230-125

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

อัลกอริทึมแบบใหม่สำหรับอะแดปทีฟ IIR นอตช์ฟิลเตอร์
โดยใช้อัลกอริทึมแบบเจเนติกควบคู่กับอัลกอริทึมแบบปรับเสถียรไซส์
และการประยุกต์ใช้งาน

A NEW ALGORITHM FOR ADAPTIVE IIR NOTCH FILTER
USING THE COMBINATION OF GENETIC ALGORITHM IN PARALLEL
WITH VARIABLE STEP-SIZE ALGORITHM AND ITS APPLICATIONS



ศรินทร์ จอโภชาอุดม

SARINPORN JORPHOCHAUDOM

เลขหมู่.....
เลขทะเบียน..... 77964
วัน,เดือน,ปี..... 12 ก.พ. 2551

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

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2550

KMITL-2007-EN-M-230-125

COPYRIGHT 2007

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ อัลกอริทึมแบบใหม่สำหรับอะแดปทีฟ IIR นอตช์ฟิลเตอร์โดยใช้อัลกอริทึมแบบเจเนติกควบคู่กับอัลกอริทึมแบบปรับเสถียรไซส์ และการประยุกต์ใช้งาน
A new Algorithm for Adaptive IIR Notch Filter using the Combination of Genetic Algorithm in Parallel with Variable Step-Size Algorithm and Its Applications

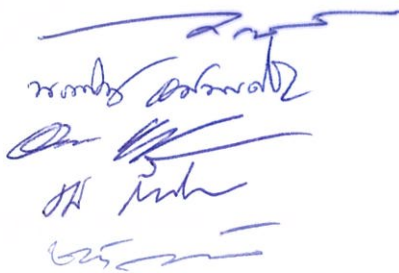
นักศึกษา นางสาวศรินท์พร จอโกษาอุดม

รหัสประจำตัว 45065110

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชา วิศวกรรมสารสนเทศ

อาจารย์ที่ปรึกษาวิทยานิพนธ์ รศ.ดร.ชวลิต เบญจางคประเสริฐ

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
รศ.ดร.กนก	เจนจิระพงศ์เวช	
รศ.นภพินท์	อนันตรศิริชัย	
รศ.อรลภ	แสงอรุณ	
รศ.ดร.ชม	กัมปาน	
รศ.ดร.ชวลิต	เบญจางคประเสริฐ	

วัน / เดือน / ปี ที่สอบ 1 ตุลาคม 2550 เวลา 09.00-11.00 น.

สถานที่สอบ ณ อาคาร A ชั้น 5 ห้องประชุม 3



วันที่.....๗.....เดือน.....๗.....พ.ศ. ๒๕๕๐.....

หัวข้อวิทยานิพนธ์	อัลกอริทึมแบบใหม่สำหรับอะแดปทีฟ IIR นอตซ์ฟิลเตอร์โดยใช้ อัลกอริทึมแบบเจเนติกควบคู่กับอัลกอริทึมแบบปรับเสถียรไปซัส และการประยุกต์ใช้งาน
นักศึกษา	นางสาวศรินทร์พร จอ โภชาอุดม
รหัสประจำตัว	45065110
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมสารสนเทศ
พ.ศ.	2550
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รศ. ดร. ชวลิต เบญจางคประเสริฐ

บทคัดย่อ

วิทยานิพนธ์นี้ได้ทำการศึกษาการทำงานและพัฒนาอัลกอริทึมแบบใหม่ สำหรับตัวกรองความถี่แบบอะแดปทีฟ IIR นอตซ์ฟิลเตอร์ ซึ่งใช้สำหรับประมาณค่าสัญญาณขาเข้าเดี่ยวที่ปะปนอยู่ในสัญญาณรบกวนแบบเกาส์เซียน หรืออิมพัลส์ โดยมีวัตถุประสงค์เพื่อเพิ่มประสิทธิภาพโดยรวมของตัวกรองความถี่ กล่าวคือมีความเร็วเข้าสู่สัมประสิทธิ์ของฟิลเตอร์สูง มีค่าความผิดพลาดต่ำ และทนต่อสัญญาณรบกวนสูง อัลกอริทึมใหม่ที่นำเสนอในวิทยานิพนธ์นี้ ได้ทำการประยุกต์ใช้อัลกอริทึมแบบเจเนติกและอัลกอริทึมแบบปรับเสถียรไปซัสมาทำงานร่วมกันแบบคู่ขนาน โดยให้ทั้งสองอัลกอริทึมทำการหาค่าเสถียรไปซัสไปพร้อม ๆ กัน แล้วเลือกค่าเสถียรไปซัสที่ดีที่สุดที่เกิดจากอัลกอริทึมทั้งสอง ผลที่ได้คือทำให้ได้อัลกอริทึมที่มีความเร็วเข้าสู่ค่าสัมประสิทธิ์ของฟิลเตอร์สูงขึ้น รวมถึงมีค่าความผิดพลาดต่ำลง โดยเฉพาะอย่างยิ่งสามารถทนต่อสัญญาณรบกวนแบบอิมพัลส์สูงขึ้นอีกด้วย นอกจากนี้ได้ทำการทดสอบอะแดปทีฟอัลกอริทึมใหม่ที่พัฒนาขึ้น โดยนำไปประยุกต์ใช้ในการกำจัดสัญญาณรบกวนสัญญาณคลื่นไซน์ความถี่ 50 เฮิร์ตซ์ ของไฟฟ้ากระแสสลับ 220 โวลต์ ที่เหนี่ยวนำเข้าไปรบกวนสัญญาณคลื่นไฟฟ้าหัวใจของผู้ป่วยขณะที่แพทย์กำลังบันทึกสัญญาณ ซึ่งได้ผลเป็นที่น่าพอใจ

Thesis Title	A New Algorithm for Adaptive IIR Notch Filter Using the Combination of Genetic Algorithm in Parallel with Variable Step-Size Algorithm and Its Applications
Student	Ms. Sarinporn Jorphochaudom
Student ID.	45065110
Degree	Master of Engineering
Program	Information Engineering
Year	2007
Thesis Advisor	Assoc. Prof. Dr. Chawalit Benjangkprasert

ABSTRACT

This thesis presents a new algorithm for adaptive IIR notch filter is used for frequency estimation in background noise such as Gaussian noise or impulse noise. The objectives are enhancing 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. A new algorithm in this thesis is using the combination of genetic algorithm in parallel with variable step-size algorithm to find the step-size and select the best step-size from two algorithms for the filter. By this new algorithm, the proposed algorithm gives good performance, such as fast convergence speed, noise robustness, and low MSE. Moreover, this algorithm can also give good stability when used to canceling 50-Hz power line interference, which induce to disturb the signal of electrocardiogram (ECG) of patient while the physician is recording it.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยความช่วยเหลือของบุคคลหลายฝ่าย ผู้วิจัยขอกราบ
ขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้ โดยเฉพาะอย่างยิ่ง รศ. ดร. ชวลิต เบญจางคประเสริฐ ที่ได้ให้
การสนับสนุนด้านข้อมูลต่าง ๆ ที่ใช้ในการดำเนินงานวิจัย รวมทั้งตรวจสอบความถูกต้องตลอดการ
ดำเนินงานวิจัยอีกด้วย

ขอขอบคุณห้องวิจัย Signal transmission laboratory สำนักวิจัยการสื่อสาร และเทคโนโลยี
สารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ให้การสนับสนุนโปรแกรม
MATLAB

ขอกราบขอบพระคุณบิดามารดาที่ให้การสนับสนุน และส่งเสริมด้านการศึกษา และขอขอบคุณ
ผู้ให้การสนับสนุนทุนในการวิจัย รวมทั้งทุกท่านที่ได้ให้คำแนะนำ คำปรึกษา และความช่วยเหลือ
ในทุกด้าน

สรินท์พร จอโกชาอุดม

สารบัญ

หน้า

บทคัดย่อ	
ภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญ.....	1
1.2 จุดมุ่งหมาย และวัตถุประสงค์ของวิทยานิพนธ์.....	3
1.3 สมมติฐานของวิทยานิพนธ์.....	3
1.4 แนวคิดที่ใช้ในการวิจัย.....	4
1.5 ขอบเขตการวิจัย.....	5
บทที่ 2 ทฤษฎีเจเนติก.....	6
2.1 พื้นฐานทางพันธุศาสตร์.....	6
2.2 อัลกอริทึมแบบเจเนติกพื้นฐาน.....	12
2.3 อัลกอริทึมแบบเจเนติกแบบง่าย.....	16
2.4 การประยุกต์เจเนติก อัลกอริทึมแบบง่าย.....	20
2.5 ตัวอย่างการประยุกต์ใช้อัลกอริทึมเจเนติกในการแก้ไขปัญหา.....	27
2.6 สรุป.....	37
บทที่ 3 การประมวลผลสัญญาณดิจิทัลด้วยกระบวนการสุ่มแบบไม่ต่อเนื่องทางเวลา.....	38
3.1 ประเภทของสัญญาณ.....	38
3.2 ตัวแปรสุ่ม (Random Variable) และการวิเคราะห์ตัวแปรสุ่ม.....	40
3.3 การแทนสัญญาณที่มีพลังงานไม่จำกัดด้วยสเปกตรัม.....	56
3.4 สรุป.....	62

สารบัญ (ต่อ)

	หน้า
บทที่ 4 ตัวกรองอะแดปทีฟ.....	63
4.1 ตัวกรองความถี่แบบ IIR (IIR filters).....	63
4.2 โครงสร้างแบบ Direct Forms ของตัวกรองความถี่ IIR.....	65
4.3 ตัวกรองความถี่แบบอะแดปทีฟ.....	67
4.4 สรุป.....	70
บทที่ 5 ตัวกรองอะแดปทีฟแบบนอตช์.....	71
5.1 กล่าวนำเกี่ยวกับตัวกรองอะแดปทีฟแบบนอตช์ (ANF).....	71
5.2 การประมาณค่าตัวกรองแบบนอตช์.....	76
5.3 Least Mean P-Power Error Criterion (LMP).....	78
5.4 Gradient-Based Algorithm แบบ Least Mean p-Power Error Criterion (LMP).....	81
5.5 อะแดปทีฟอัลกอริทึมที่เกี่ยวข้อง.....	83
5.6 อะแดปทีฟอัลกอริทึมที่นำเสนอในวิทยานิพนธ์.....	90
5.7 สรุป.....	100
บทที่ 6 ผลการวิจัย.....	101
6.1 การทดสอบอะแดปทีฟอัลกอริทึมในการประมาณสัญญาณซายน์คลื่นเดี่ยว.....	101
6.2 การทดสอบอะแดปทีฟอัลกอริทึมในการกำจัดสัญญาณรบกวน 50 เฮิร์ตซ์ ที่ปะปน ในสัญญาณคลื่นหัวใจ.....	115
6.3 สรุป.....	116
บทที่ 7 สรุปและวิจารณ์ผลการทดลอง.....	118
เอกสารอ้างอิง.....	120
ภาคผนวก.....	123
ภาคผนวก ก. บทความวิจัยที่ได้รับการตีพิมพ์.....	124

สารบัญ (ต่อ)

	หน้า
ประวัติผู้เขียน.....	125

สารบัญตาราง

ตารางที่	หน้า
2.1 ตัวอย่างโครโมโซมรุ่นแรก.....	21
2.2 ค่าความเหมาะสมของโครโมโซมแต่ละตัว.....	22
2.3 ตัวอย่างการคัดเลือกโครโมโซมต้นแบบ.....	23
2.4 โครโมโซมต้นแบบใน Mating pool.....	23
2.5 การครอสโอเวอร์โครโมโซมต้นแบบเพื่อสร้างโครโมโซมรุ่นใหม่.....	24
2.6 ตัวอย่างการมิวเตชันโครโมโซม.....	26
2.7 โครโมโซมชุดใหม่ หลังผ่านขบวนการทางพันธุศาสตร์ วัฏจักรแรก (gen=0) และวัฏจักรที่ 2 (gen=1).....	27
2.8 การทำงานเพื่อหาค่า x ที่ทำให้สมการ $x^2-6=30$ เป็นจริง ของอัลกอริทึมแบบเจเนติก ในวัฏจักรที่ 1.....	29
2.9 การทำงานเพื่อหาค่า x ที่ทำให้สมการ $x^2-6=30$ เป็นจริง ของอัลกอริทึมแบบเจเนติก ในวัฏจักรที่ 2.....	30
2.10 การทำงานเพื่อหาค่า x ที่ทำให้สมการ $x^2-6=30$ เป็นจริง ของอัลกอริทึมแบบเจเนติก ในวัฏจักรที่ 3.....	31
2.11 การทำงานเพื่อหาค่า x ที่ทำให้สมการ $x^2-6=30$ เป็นจริง ของอัลกอริทึมแบบเจเนติก ในวัฏจักรที่ 4.....	32
2.12 การทำงานเพื่อหาค่า x ที่ทำให้สมการ $x^2-6=30$ เป็นจริง ของอัลกอริทึมแบบเจเนติก ในวัฏจักรที่ 5.....	33
2.13 การทำงานเพื่อหาค่า x ที่ทำให้สมการ $x^2-6=30$ เป็นจริง ของอัลกอริทึมแบบเจเนติก ในวัฏจักรที่ 6.....	34
2.14 การทำงานเพื่อหาค่า x ที่ทำให้สมการ $x^2-6=30$ เป็นจริง ของอัลกอริทึมแบบเจเนติก ในวัฏจักรที่ 7.....	35
2.15 การทำงานเพื่อหาค่า x ที่ทำให้สมการ $x^2-6=30$ เป็นจริง ของอัลกอริทึมแบบเจเนติก ในวัฏจักรที่ 8.....	36
5.1 ตัวอย่างการแปลงค่าโครโมโซมเป็นสตริงไบนารี.....	91

สารบัญรูป

รูปที่	หน้า
1.1 แนวคิดในการกำจัดสัญญาณรบกวนแบบอะแคทีฟของ Widrow [24].....	1
2.1 โครโมโซมทางพันธุศาสตร์.....	8
2.2 การแบ่งตัวของเซลล์ทางพันธุศาสตร์.....	9
2.3 การครอสโอเวอร์ทางพันธุศาสตร์.....	10
2.4 การมิวเตชันทางพันธุศาสตร์.....	11
2.5 วัฏจักรของเจเนติก อัลกอริทึม.....	13
2.6 ไดอะแกรมของเจเนติก อัลกอริทึมแบบง่าย.....	16
2.7 ครอสโอเวอร์แบบ 1 จุด.....	19
2.8 ไบนารีมิวเตชัน.....	20
3.1 สัญญาณ Deterministic แบบมีคาบ.....	39
3.2 สัญญาณแบบ Deterministic แบบไม่มีคาบ.....	39
3.3 สัญญาณแบบสุ่ม.....	40
3.4 สัญญาณแบบสุ่มที่ไม่ต่อเนื่องทางเวลา.....	40
3.5 Pdf ของตัวแปรสุ่ม $x[0]$	41
3.6 สัญญาณแบบสุ่มที่มีค่า 1 และ -1	42
3.7 Pdf ของสัญญาณแบบไบนารี.....	42
3.8 ตัวอย่างสัญญาณแบบสุ่ม.....	43
3.9 Gaussian Distribution ของ $x[2]$	44
3.10 Gaussian Distribution ของ $x[2]$ เมื่อ $N(3,3,1)$	44
3.11 (a) Pdf และ (b) Cdf ของตัวแปรสุ่ม $x[n]$	45
3.12 (a) Pdf และ (b) Cdf ของ $x[n]$	46
3.13 (a) Pdf และ (b) Cdf ของตัวแปรสุ่มที่เป็นแบบ Gaussian Distribution.....	46
3.14 สัญญาณแบบสุ่มไบนารี.....	47
3.15 (a) Pdf ของ $x[n_1]$ และ (b) Pdf ของ $x[n_2]$	47
3.16 Joint Pdf ของ $x[1]$ และ $x[2]$	47
3.17 ตัวแปรสุ่มสองตัว.....	50
3.18 อธิบายความหมายของ Autocorrelation.....	53

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.19 Hand-free Mobile phone.....	54
3.20 บล็อกไดอะแกรมของรูปที่ 3.20.....	54
3.21 ROC และตำแหน่งโพลและซีโรที่ได้จากการแปลง Z ของ $C_{xx}(z)$	57
3.22 ผลตอบสนองทางความถี่ของตัวกรองแบบแถบผ่านในอุดมคติ.....	61
4.1 Signal flow graph ของโครงสร้างแบบ Direct Form I สำหรับระบบอันดับ N	66
4.2 Signal flow graph ของโครงสร้างแบบ Direct Form II สำหรับระบบอันดับ N	67
4.3 ส่วนประกอบโดยทั่วไปของตัวกรองความถี่อะแดปทีฟ IIR	68
4.4 ตัวอย่างการประยุกต์ใช้งานเพื่อทำ System Identification.....	69
5.1 ผลตอบสนองทางขนาดต่อความถี่ $ \hat{H}(e^{j\omega}) $ เมื่อเปลี่ยนค่าสัมประสิทธิ์ ρ และ a	77
5.2 ตำแหน่ง โพล-ซีโร ของ $\hat{H}(z)$ เมื่อ $\rho = 0.95$ และ $a = -1.5$	78
5.3 ค่า Mean p -Power error $J(a)$ สำหรับ $p = 1, 2$ และ 3	80
5.4 การลู่เข้าของพารามิเตอร์ a เมื่อใช้ Steepest decent อัลกอริทึม ที่ $p = 1, 2$ และ 3	80
5.5 โครงสร้างแบบ Direct form ของตัวกรอง ANF.....	83
5.6 สัมประสิทธิ์ฟิลเตอร์ $a(n)$ โดยใช้ อัลกอริทึมแบบเจนเนติก กำหนดให้ $P_m = 0.05$ และ $P_c = 0.5$	87
5.7 สัมประสิทธิ์ฟิลเตอร์ $a(n)$ โดยใช้ อัลกอริทึมแบบเจนเนติก กำหนดให้ $P_m = 0.001$ และ $P_c = 0.01$	87
5.8 สัมประสิทธิ์ของฟิลเตอร์ $a(n)$ โดยใช้ อัลกอริทึมแบบเจนเนติก ที่กำหนดให้ P_m และ P_c แปรค่าได้.....	88
5.9 เงื่อนไขในการทำงานร่วมกันของ อัลกอริทึมแบบเจนเนติกกับ อัลกอริทึมแบบเสถียร [35].....	89
5.10 เปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ (a) VSQMLP แบบเดิม[29], (b) อัลกอริทึมแบบปรับเสถียร [35] (c) อัลกอริทึมแบบเจนเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียร [35]	89
5.11 หลักการหาค่าเสถียรโดยใช้อัลกอริทึมแบบเจนเนติก.....	92
5.12 ความสัมพันธ์ของ P_c และ P_m ที่ส่งผลต่อค่าความแปรปรวน.....	94

สารบัญรูป (ต่อ)

รูปที่	หน้า
5.13 เปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ ของ อัลกอริทึมแบบปรับเสถียรไปไซส์ทำงานร่วมกับอัลกอริทึมแบบเจนเนติก [35] ที่ $\omega = \pi/3$ โดย (a) ใช้ค่า Pc, Pm ที่เหมาะสม (ค่า Pm มีค่าต่ำ) คือ Pc=0.08 Pm=0.02 และ (b) Pc=0.51 Pm=0.84 (ค่า Pm สูง).....	95
5.14 ค่าของสัมประสิทธิ์ฟิลเตอร์ $a(n)$ เมื่อเกิดสัญญาณรบกวนแบบอิมพัลส์ ขนาด 50 หน่วย ที่ตำแหน่ง $n = 500$ (a) อัลกอริทึมแบบเจนเนติกทำงานร่วมกับ อัลกอริทึมแบบปรับเสถียรไปไซส์ [35] (b) อัลกอริทึมแบบเจนเนติกแบบใหม่ทำงานร่วมกับ อัลกอริทึมแบบปรับเสถียรไปไซส์แบบใหม่.....	97
5.15 การทำงานของอัลกอริทึมใหม่.....	99
6.1 บล็อกโคอะแกรมอะแคปทีฟนอตซ์ฟิลเตอร์.....	101
6.2 เปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ ที่ $\mu(0) = 0.2$ โดย (a) อัลกอริทึมแบบเจนเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไปไซส์ [35] (b) อัลกอริทึมแบบเจนเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไปไซส์แบบใหม่ และ (c) อัลกอริทึมใหม่ที่นำเสนอ.....	103
6.3 เปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ ที่ $\mu(0) = 0.1$ โดย (a) อัลกอริทึมแบบเจนเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไปไซส์ [35] (b) อัลกอริทึมแบบเจนเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไปไซส์แบบใหม่ และ (c) อัลกอริทึมใหม่ที่นำเสนอ.....	104
6.4 เปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ ที่ $\mu(0) = 0.01$ โดย (a) อัลกอริทึมแบบเจนเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไปไซส์ [35] (b) อัลกอริทึมแบบเจนเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไปไซส์แบบใหม่ และ (c) อัลกอริทึมใหม่ที่นำเสนอ.....	105
6.5 กราฟเปรียบเทียบค่า MSE ของอัลกอริทึมแบบเจนเนติกทำงานร่วมกับ อัลกอริทึมแบบปรับเสถียรไปไซส์[35] อัลกอริทึมแบบปรับเสถียรไปไซส์แบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไปไซส์แบบใหม่ และอัลกอริทึมใหม่ที่นำเสนอ เมื่อกำหนดให้ $\mu(0) = 0.01$	106

สารบัญญรูป (ต่อ)

รูปที่	หน้า
6.6 เปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ ที่ $\omega = \pi/3$	
(a) อัลกอริทึมแบบเจนเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์[35]	
(b) อัลกอริทึมแบบเจนเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์แบบใหม่	
และ (c) อัลกอริทึมใหม่ที่น่าเสนอ.....	107
6.7 เปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ ที่ $\omega = \pi/4$	
(a) อัลกอริทึมแบบเจนเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์[35]	
(b) อัลกอริทึมแบบเจนเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์แบบใหม่	
และ (c) อัลกอริทึมใหม่ที่น่าเสนอ.....	108
6.8 เปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ ที่ $\omega = \pi/7$	
(a) อัลกอริทึมแบบเจนเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์[35]	
(b) อัลกอริทึมแบบเจนเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์แบบใหม่	
และ (c) อัลกอริทึมใหม่ที่น่าเสนอ.....	109
6.9 เปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ เมื่อเกิดสัญญาณอิมพัลส์ ขนาด 30 หน่วยที่ตำแหน่ง $n=500$	
(a) อัลกอริทึมแบบเจนเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์[35]	
(b) อัลกอริทึมแบบเจนเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์แบบใหม่	
และ (c) อัลกอริทึมใหม่ที่น่าเสนอ.....	110
6.10 เปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ เมื่อเกิดสัญญาณอิมพัลส์ ขนาด 50 หน่วยที่ตำแหน่ง $n=500$	
(a) อัลกอริทึมแบบเจนเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์[35]	
(b) อัลกอริทึมแบบเจนเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์แบบใหม่	
และ (c) อัลกอริทึมใหม่ที่น่าเสนอ.....	111
6.11 เปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ เมื่อเกิดสัญญาณอิมพัลส์ ขนาด 130 หน่วยที่ตำแหน่ง $n=500$	
(a) อัลกอริทึมแบบเจนเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์[35]	
(b) อัลกอริทึมแบบเจนเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์แบบใหม่	
และ (c) อัลกอริทึมใหม่ที่น่าเสนอ.....	112

สารบัญรูป (ต่อ)

รูปที่	หน้า
6.12	
เปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ เมื่อเกิดสัญญาณอิมพัลส์ ขนาด 2,000 หน่วยที่ตำแหน่ง $n=500$	
(a) อัลกอริทึมแบบเจนเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์[35]	
(b) อัลกอริทึมแบบเจนเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์แบบใหม่ และ (c) อัลกอริทึมใหม่ที่น่าเสนอ.....	113
6.13	
เปรียบเทียบค่า MSE ของ อัลกอริทึมแบบเจนเนติกทำงานร่วมกับอัลกอริทึมแบบ ปรับเสถียรไซส์[35] อัลกอริทึมแบบเจนเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึม แบบปรับเสถียรไซส์แบบใหม่ และ อัลกอริทึมใหม่ที่น่าเสนอ เมื่อเกิดสัญญาณ รบกวนแบบอิมพัลส์ 30 หน่วยที่ ตำแหน่ง $n = 500$	114
6.14	
สัญญาณคลื่นหัวใจ ที่ออกจากตัวกรองสัญญาณ (a) อัลกอริทึมแบบ ปรับเสถียรไซส์ทำงานร่วมกับอัลกอริทึมแบบเจนเนติกแบบเดิม [35]	
(b) อัลกอริทึมใหม่ที่น่าเสนอ.....	116

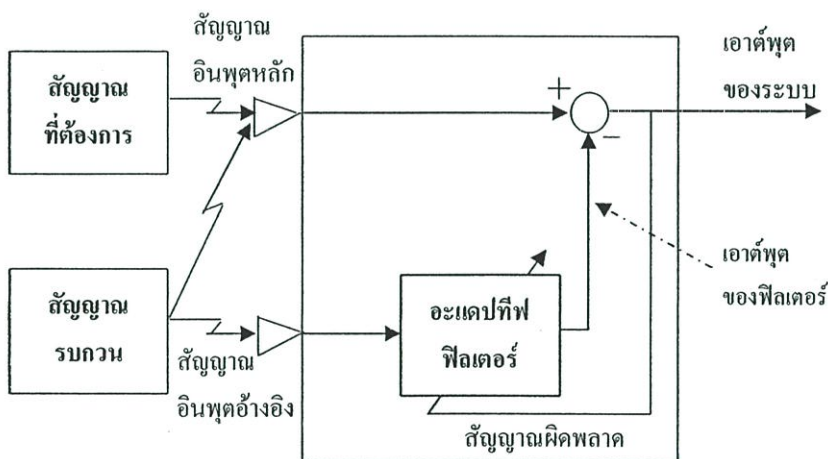
บทที่ 1

บทนำ

1.1 ความเป็นมา และความสำคัญ

ตัวกรองอะแดปทีฟแบบนอตช์ (Adaptive Notch Filter : ANF) ถูกนำมาใช้ประมาณสัญญาณคลื่นไซน์ที่ไม่ทราบความถี่ที่ปะปนอยู่ในสัญญาณรบกวน ซึ่งสัญญาณดังกล่าวจะพบได้ในระบบสื่อสารหลาย ๆ ระบบ ตัวอย่างเช่น ในระบบคิโมคูลต จะใช้สัญญาณคลื่นไซน์เป็นสัญญาณพาห์ (Carrier) มีโอกาสที่สัญญาณคลื่นไซน์จะมีความถี่เปลี่ยนไปเมื่อเดินทางไปถึงเครื่องรับ หรือเกิดการเลื่อนของเฟส และจะเกิดปัญหากับด้านรับเพราะไม่สามารถนำสัญญาณเดิมกลับมาได้ หรือมีความผิดเพี้ยนของสัญญาณเกิดขึ้น การประมาณค่าสัญญาณไซน์เดี่ยวที่ปะปนอยู่ในสัญญาณรบกวนนั้น สามารถทำได้โดยการป้อนสัญญาณดังกล่าว ผ่านตัวกรองความถี่ ซึ่งตัวกรองความถี่จะทำการกดสัญญาณรบกวนให้มีขนาดต่ำลงหรือกำจัดสัญญาณรบกวนออก และยอมให้สัญญาณที่ต้องการผ่านออกไป โดยตัวกรองสัญญาณที่ดี (Optimum filter) ต้องมีคุณสมบัติสำคัญ คือ มีความเร็วในการหาคำตอบสูง มีค่าความคลาดผิดพลาด (Mean Square Error: MSE) ต่ำ และมีความทนทานต่อสัญญาณรบกวนได้ดี

ตัวกรองความถี่แบบอะแดปทีฟได้ถูกนำเสนอครั้งแรก ในปี ค.ศ. 1975 โดย Widrow [24] ได้นำเสนอแนวคิดในการกำจัดสัญญาณรบกวนแบบอะแดปทีฟเป็นครั้งแรก แสดงดังรูปที่ 1.1



รูปที่ 1.1 แนวคิดในการกำจัดสัญญาณรบกวนแบบอะแดปทีฟของ Widrow [24]

จากรูปที่ 1.1 ตัวกรองแบบอะแดปทีฟ ต้องการสัญญาณอินพุตสองสัญญาณ คือ สัญญาณอินพุตหลัก (Primary input) และสัญญาณอินพุตอ้างอิง (Reference input) ซึ่งสัญญาณอินพุตอ้างอิงจะได้จากอุปกรณ์ตรวจจับสัญญาณรบกวน (Sensors) ซึ่งอาจจะมีหลาย ๆ ตัวโดยวางอยู่ในตำแหน่ง

ที่มีสัญญาณรบกวน หรือ ตำแหน่งที่สัญญาณที่ต้องการมีขนาดต่ำ ซึ่งสัญญาณนี้จะถูกรองโดยตัวกรองแบบอะแดปทีฟ แล้วนำไปลบกับสัญญาณอินพุตหลัก ซึ่งประกอบไปด้วยสัญญาณที่ต้องการและสัญญาณรบกวน เป็นผลให้สัญญาณรบกวนถูกลดทอนลง กระบวนการนี้แม้จะดูเหมือนว่าอันตราย เพราะถ้าหากกระทำอย่างไม่เหมาะสมแล้ว จะเป็นการเพิ่มกำลังงานของสัญญาณรบกวนที่สัญญาณเอาต์พุต แต่อย่างไรก็ตาม ถ้ากระบวนการกรอง และการลบถูกควบคุมอย่างเหมาะสมแล้ว กระบวนการทางอะแดปทีฟจะสามารถทำการลดสัญญาณรบกวนได้อย่างสมบูรณ์ โดยทำให้สัญญาณมีความเพี้ยนน้อยที่สุด

ในการกรองสัญญาณของตัวกรองสัญญาณแบบอะแดปทีฟ IIR แบบนอธซ์ (ANF) ซึ่งใช้ใน งานเรดาร์ หรือ โซนาร์ จะต้องมีการปรับค่าสัมประสิทธิ์ของฟิลเตอร์ (Filter Coefficient) ให้เหมาะสม ตัวอย่างอัลกอริทึมที่นำมาใช้ในการหาค่าสัมประสิทธิ์ของฟิลเตอร์ที่เหมาะสม 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 จะมีข้อเสีย คือการหาค่าสัมประสิทธิ์ของฟิลเตอร์ที่ถูกต้องได้ช้า โดยทั่วไปอัลกอริทึมแบบ LMP และ QLMP จะมีค่าเสถียรไซส์แบบคงที่ ซึ่งถ้ากำหนดค่าเสถียรไซส์สูงจะทำให้ความเร็วในการหาค่าตอบเร็วขึ้น แต่ค่า MSE ก็จะสูงตามไปด้วย หากค่าเสถียรไซส์มีค่าต่ำจะทำให้การหาค่าตอบช้าลง แต่ค่า MSE ก็ต่ำลงเช่นกัน

ต่อมาได้มีงานวิจัยอัลกอริทึมที่สามารถหาค่าเสถียรไซส์ที่เหมาะสม โดยไม่ต้องกำหนดค่าเสถียรไซส์แบบคงที่ ทำให้ความเร็วในการหาค่าสัมประสิทธิ์ของฟิลเตอร์ที่ถูกต้องเร็วขึ้น เช่น Variable Step-Size Quantize Least Mean P-Power Error Criterion (VSQ LMP) อัลกอริทึมแบบ VSQ LMP เข้ามาช่วยแก้ปัญหาเรื่องการปรับค่าเสถียรไซส์ ทำให้ตัวกรองสัญญาณมีความเร็วในการหาค่าตอบสูง และค่า MSE ต่ำ ซึ่งอัลกอริทึมแบบ VSQ LMP จะทำการปรับค่าเสถียรไซส์ให้เหมาะสม แต่มีข้อเสียเรื่องความทนทานต่อสัญญาณรบกวนประเภทอิมพัลส์ (Impulse) ได้ไม่คึก ซึ่งอาจทำให้ตัวกรองสัญญาณไม่สามารถลู่เข้าคำตอบได้อีกเมื่อเกิดสัญญาณรบกวนประเภทอิมพัลส์ขึ้น

จากนั้นได้มีผู้วิจัยต่อยอดอัลกอริทึมแบบปรับเสถียรไซส์ (VSQ LMP) เพื่อแก้ปัญหาเรื่องความเร็วในการหาค่าตอบและให้มีความทนทานต่อสัญญาณรบกวนประเภทอิมพัลส์ได้ดียิ่งขึ้น โดยนำเอาหลัก Genetic Search Algorithm (GA) มาประยุกต์ใช้หาค่าเสถียรไซส์ในช่วงแรกเพื่อช่วยแก้ปัญหาในเรื่องความเร็วในการหาค่าตอบ และนำหลักเกรเดียนต์เอาต์พุตมาแก้ปัญหาเรื่องความทนสัญญาณรบกวน โดยอัลกอริทึมแบบนี้เป็นการทำงานร่วมกันระหว่างอัลกอริทึมแบบเจเนติกกับอัลกอริทึมแบบปรับเสถียรไซส์ [35] ทำให้ได้อัลกอริทึมที่มีความเร็วลู่เข้าคำตอบที่เร็ว และทนต่อสัญญาณรบกวนประเภทอิมพัลส์ได้ดีกว่าอัลกอริทึมแบบปรับเสถียรไซส์ (VSQ LMP) แต่จากการศึกษาอัลกอริทึมนี้ แม้ว่าจะเป็นอัลกอริทึมที่มีความเร็วในการลู่เข้าสู่คำตอบสูง แต่ยังพบความแปรปรวนในช่วงแรกของการหาค่าตอบ และรูปแบบความแปรปรวนนั้นยังมีรูปแบบที่ไม่แน่นอน

จากปัญหาดังกล่าว วิทยานิพนธ์นี้จึงได้ทำการวิจัยเพื่อปรับปรุงประสิทธิภาพตัวกรองสัญญาณให้ดีขึ้น โดยให้มีความเร็วผู้เข้าสู่ค่าตอบสูง ค่า MSE ต่ำ ทนต่อสัญญาณรบกวนได้ดีกว่าเดิม และลดความแปรปรวนในช่วงแรกของการกรองสัญญาณ การวิจัยนี้ได้นำหลักของอัลกอริทึมแบบเจนเนติกมาทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์ โดยได้ทำการปรับปรุงการทำงานของทั้งสองอัลกอริทึมใหม่ รวมทั้งปรับเปลี่ยนรูปแบบการทำงานร่วมกันของทั้งสองอัลกอริทึมให้ทำงานกันอย่างสอดคล้อง ทำให้ได้อัลกอริทึมที่มีประสิทธิภาพในการกรองสัญญาณที่ดีขึ้น กล่าวคือมีความเร็วผู้เข้าสู่ค่าตอบสูง ค่า MSE ต่ำ และทนต่อสัญญาณรบกวนประเภทอิมพัลส์ได้ดีมาก รวมทั้งลดความแปรปรวนในช่วงแรกของการทำงานได้เป็นอย่างดี ทั้งนี้สามารถนำไปตรวจจับสัญญาณชายน์แบบคลื่นเดี่ยว และกำจัดสัญญาณรบกวน 50 เฮิร์ตซ์ ที่เกิดจากความถี่ของไฟฟ้ากระแสสลับที่ปะปนมาในสัญญาณคลื่นหัวใจ (ECG) โดยได้ผลเป็นที่น่าพอใจ

1.2 จุดมุ่งหมาย และวัตถุประสงค์ของวิทยานิพนธ์

การพัฒนาตัวกรองสัญญาณดิจิทัลให้มีประสิทธิภาพที่ดีนั้น สิ่งที่ต้องคำนึงถึง คือคุณสมบัติตัวกรองสัญญาณที่ดี จะต้องมีความเร็วผู้เข้าสู่ค่าตอบสูง ค่าความผิดพลาด (MSE) ต่ำ และมีความทนต่อสัญญาณรบกวนได้ดี เพื่อให้ได้ตัวกรองสัญญาณที่มีคุณสมบัติดังกล่าว วิทยานิพนธ์นี้จึงได้ทำการศึกษาวิจัย โดยศึกษาข้อดีข้อเสียของตัวกรองสัญญาณอะแดปทีฟแบบต่าง ๆ เพื่อนำมาปรับปรุงให้มีประสิทธิภาพดีขึ้น จากการศึกษพบว่าตัวกรองอะแดปทีฟที่ใช้การทำงานร่วมกันระหว่างอัลกอริทึมแบบเจนเนติกกับอัลกอริทึมแบบปรับเสถียรไซส์ [35] เป็นอัลกอริทึมที่มีความเร็วในการหาค่าตอบสูง มีความผิดพลาดต่ำ มีสามารถทนต่อสัญญาณรบกวนประเภทอิมพัลส์ได้ดี และเป็นตัวกรองที่ให้ประสิทธิภาพโดยรวมของตัวกรองดีกว่าตัวกรองสัญญาณแบบอื่น แต่พบว่าอัลกอริทึมแบบนี้ยังมีข้อบกพร่องบางอย่าง กล่าวคือ ยังพบความแปรปรวนในช่วงแรกของการทำงานก่อนข้างสูง และรูปแบบความแปรปรวนรวมถึงความเร็วผู้เข้าสู่ค่าตอบที่ทำได้ในแต่ละครั้งไม่แน่นอน จากปัญหาดังกล่าว วิทยานิพนธ์ฉบับนี้จึงได้มีแนวคิดที่จะพัฒนาอะแดปทีฟอัลกอริทึมแบบใหม่ โดยใช้ อัลกอริทึมแบบปรับเสถียรไซส์ และอัลกอริทึมแบบเจนเนติก มาทำงานร่วมกันโดยมีการปรับเปลี่ยนรูปแบบการทำงานร่วมกันของทั้งสองอัลกอริทึมใหม่ เพื่อให้ได้อะแดปทีฟอัลกอริทึมที่มีคุณสมบัติดีขึ้นกว่าเดิม

1.3 สมมติฐานของวิทยานิพนธ์

การนำอะแดปทีฟอัลกอริทึมแบบเจนเนติกมาทำงานร่วมกับ อัลกอริทึมแบบปรับเสถียรไซส์โดยใช้ อัลกอริทึมแบบเจนเนติกมาหาค่าเสถียรไซส์ในช่วงแรก แล้วค่อยเปลี่ยนไปใช้อัลกอริทึมแบบเสถียรไซส์นั้น ถึงแม้จะให้ความเร็วผู้เข้าสู่ค่าตอบสูง แต่พบว่ารูปแบบการผู้เข้าสู่ค่าตอบนั้นไม่แน่นอน และพบความแปรปรวนสูงในช่วงแรกด้วย ซึ่งความแปรปรวนช่วงแรกนี้น่าจะเกิดจากความไม่แน่นอนของ

อัลกอริทึมแบบเจเนติกเนื่องจากการกำหนดฟังก์ชันเป้าหมายของอัลกอริทึมแบบเจเนติก หรืออาจจะเกิดจากการสุ่มต้นแบบโครโมโซมได้ไม่ดี รวมทั้งการกำหนดค่าความน่าจะเป็นของการครอสโอเวอร์ (Pc) และค่าความน่าจะเป็นของการมิวเตชัน (Pm) ที่ไม่เหมาะสมด้วย หลังจากหาค่าเสถียรค่าแล้ว อัลกอริทึมแบบเจเนติกในช่วงแรกแล้ว จะเปลี่ยนการทำงานจากอัลกอริทึมแบบเจเนติกไปเป็นอัลกอริทึมแบบปรับเสถียรค่า ช่วงการเปลี่ยนอัลกอริทึมจะพบว่ามีความแปรปรวนเล็กน้อย ความแปรปรวนนี้น่าจะเป็นผลจากการทำงานที่ไม่ต่อเนื่องกันของทั้งสองอัลกอริทึม

วิทยานิพนธ์นี้จึงได้ตั้งสมมติฐานว่า ถ้าให้อัลกอริทึมแบบเจเนติกทำงานไปพร้อมกับอัลกอริทึมแบบปรับเสถียรค่า โดยเทียบค่าสัมประสิทธิ์ของฟิลเตอร์ที่ได้กับค่าอ้างอิงค่าหนึ่ง ซึ่งเป็นค่าคาดหวังของคำตอบ แล้วเลือกค่าเสถียรค่าที่มีค่าสัมประสิทธิ์ตัวกรองใกล้เคียงกับค่าคาดหวังของคำตอบที่สุด จะสามารถลดความแปรปรวนช่วงเปลี่ยนอัลกอริทึมจากอัลกอริทึมแบบเจเนติกเป็นอัลกอริทึมแบบปรับเสถียรค่า รวมถึงจะสามารถลดความแปรปรวนที่เกิดขึ้นในช่วงแรกลงได้ ทำให้ได้ตัวกรองสัญญาณแบบอะแดปทีฟที่มีความเร็วเข้าสู่คำตอบสูงขึ้น ค่าความผิดพลาด (MSE) ต่ำ อีกประการหนึ่งในกรณีที่เกิดสัญญาณรบกวนประเภทอิมพัลส์ จะใช้ค่าเสถียรค่าที่ได้จากอัลกอริทึมแบบเจเนติก ทำให้ทนต่อสัญญาณรบกวนได้ดีกว่าการใช้อัลกอริทึมแบบปรับเสถียรค่าเพียงอย่างเดียว

1.4 แนวคิดที่ใช้ในการวิจัย

1. อัลกอริทึมแบบปรับเสถียรค่า เป็นอัลกอริทึมที่มีค่าความผิดพลาด (MSE) ต่ำ สามารถทนต่อสัญญาณรบกวนได้ดี แต่มีความเร็วเข้าสู่คำตอบช้า จึงได้นำกรณีศึกษาของเอาต์พุตมาช่วยในการปรับค่าเสถียรค่า เพื่อให้ได้อัลกอริทึมแบบปรับเสถียรค่าแบบใหม่ที่มีความแปรปรวนต่ำ สามารถทนต่อสัญญาณรบกวนแบบอิมพัลส์ได้ดีกว่าเดิม แต่ก็ยังพบว่ามีความเร็วเข้าสู่คำตอบต่ำ

2. อัลกอริทึมแบบเจเนติก เป็นอัลกอริทึมที่สามารถปรับค่าความเร็วในการหาคำตอบได้โดยการปรับ Pc (Probability of Crossover) และ Pm (Probability of Mutation) ดังนั้นการหาค่าเหมาะสมของค่า Pc และ Pm จะได้อัลกอริทึมที่มีความเร็วในการหาคำตอบที่สูงไปด้วย แต่ผลที่ได้คือค่าความผิดพลาด (MSE) จะสูงตามไปด้วย ทั้งนี้เนื่องจากอัลกอริทึมแบบเจเนติกเป็นค่าแบบสุ่ม ดังนั้นความคงที่ในการหาคำตอบจึงไม่ค่อยแน่นอนอนันต์ ก่อนใช้งานจึงต้องทดสอบเพื่อเลือกค่า Pc และ Pm ที่เหมาะสม รวมถึงการปรับเปลี่ยนฟังก์ชันความเหมาะสมและฟังก์ชันเป้าหมายของอัลกอริทึมแบบเจเนติกใหม่ ทำให้ได้อัลกอริทึมแบบเจเนติกที่มีประสิทธิภาพดียิ่งขึ้น

3. การนำข้อดีของอัลกอริทึมแบบเจเนติก ที่มีความเร็วในการหาคำตอบที่สูง และอัลกอริทึมแบบปรับเสถียรค่าแบบใหม่ ซึ่งมีค่าความผิดพลาด (MSE) ต่ำ มาทำงานร่วมกัน จะได้อะแดปทีฟอัลกอริทึมที่ทำงานได้เร็ว และ ค่าสัมประสิทธิ์ของฟิลเตอร์ที่มีความถูกต้องสูง หรือค่า MSE ต่ำ นั่นเอง โดยการเปลี่ยนรูปแบบการทำงานร่วมกันใหม่ให้ทั้งสองอัลกอริทึมทำงานไปพร้อม ๆ กัน

แล้วเลือกค่าเสถียรที่สุดจากอัลกอริทึมทั้งสอง โดยเทียบค่าสัมประสิทธิ์ฟิลเตอร์กับค่าคาดหวังของคำตอบ การทำงานแบบนี้จะได้อะแดปทีฟอัลกอริทึมที่มีประสิทธิภาพในการกรองสัญญาณที่ดีขึ้น กล่าวคือ มีความเร็วเข้าสู่คำตอบสูง ค่าความผิดพลาด (MSF) ต่ำ และทนต่อสัญญาณรบกวนได้ดีกว่าเดิม

1.5 ขอบเขตการวิจัย

วิทยานิพนธ์นี้มีวัตถุประสงค์ เพื่อนำเสนออัลกอริทึมที่ใช้ปรับค่าเสถียรของอะแดปทีฟ IIR นอตซ์ฟิลเตอร์อันดับสอง ให้มีความเร็วในการหาคำตอบหรือค่าสัมประสิทธิ์ของฟิลเตอร์สูงขึ้น รวมถึงมีความทนทานต่อสัญญาณรบกวนประเภทอิมพัลส์ได้ดีกว่า โดยเปรียบเทียบจากบทวิจัยที่ผ่านมา [35] และเพื่อใช้เป็นแนวทางการพัฒนาอะแดปทีฟอัลกอริทึมในอนาคตต่อไป โดยในวิทยานิพนธ์ฉบับนี้ได้ใช้โปรแกรมคอมพิวเตอร์ เข้ามาช่วยสำหรับจำลองการทำงานของอะแดปทีฟอัลกอริทึมแบบต่าง ๆ ที่ได้นำเสนอ

บทที่ 2

ทฤษฎีเจเนติก

การจำลองการทำงานของธรรมชาติ เพื่อนำมาช่วยในการหาคำตอบที่ดีที่สุด (Optimal solution) ได้มีการนำไปศึกษาวิจัย เช่น นิวรอนเน็ตเวิร์ค (Neural network) ฟัซซีลอจิก (Fuzzy logic) เป็นต้น ส่วนกระบวนการเชิงพันธุกรรม (Genetic Algorithm) เป็น Algorithm ที่จำลองการทำงานเชิงพันธุกรรม ซึ่งสามารถนำมาใช้ในการแก้ไขปัญหาที่ไม่สามารถหาวิธีที่ดีที่สุดได้ แต่อาจต้องการวิธีที่เหมาะสมหลาย ๆ วิธี เพื่อนำมาพิจารณาตัดสินใจ โดยจะเลียนแบบการแลกเปลี่ยนระหว่างยีนหรือการเกิดโครโมโซมใหม่ จึงทำให้ได้คำตอบใหม่ที่ดีที่สุด หรือใกล้เคียงที่สุดของแต่ละปัญหา ที่มีลักษณะสืบทอดมาจากคำตอบที่มีอยู่แล้ว

2.1 พื้นฐานทางพันธุศาสตร์

พันธุศาสตร์ (Genetics - มีรากศัพท์มาจากภาษากรีกคำว่า "genno" ซึ่งแปลว่า "การให้กำเนิด") คือ ศาสตร์แขนงหนึ่งของวิทยาศาสตร์ว่าด้วยการศึกษาหน่วยถ่ายทอดพันธุ (ยีน, Gene) การกรรมพันธุ์ (Heredity) และวิวัฒนาการในสิ่งมีชีวิตต่าง ๆ ในอดีตระยะเริ่มต้นของการศึกษาด้านวิทยาศาสตร์ ยังไม่ทราบถึงกลไกการถ่ายทอดลักษณะของสิ่งมีชีวิต จากตัวเดิมไปยังสิ่งมีชีวิตที่เกิดขึ้นใหม่ได้อย่างแน่ชัด จึงเกิดการตั้งสมมติฐานต่าง ๆ ขึ้นมากมาย แต่เมื่อมีการค้นคว้าและการศึกษาด้านพันธุศาสตร์มากขึ้น ทำให้เกิดความเข้าใจเกี่ยวกับการถ่ายทอดลักษณะทางพันธุกรรมของสิ่งมีชีวิตมากขึ้น โดยเริ่มตั้งแต่ ใน ปี ค.ศ. 1875 ออสการ์ เฮร์ทวิก (Oscar Hertwig) [22] ค้นพบว่าการปฏิสนธิเกิดจากการรวมของอสุจิกับนิวเคลียสของไข่ วอร์เตอร์ เฟลมมิง (Walter Flemming) [22] ค้นพบโครโมโซมในนิวเคลียส และการแบ่งตัวแบบไมโทซิส (Mitosis) ในกระบวนการแบ่งเซลล์ แวน เบลนเดน (Van Benden) [22] พบการแบ่งตัวแบบไมโอซิส (Miosis) ในกระบวนการสร้างเซลล์สืบพันธุ์ จนกระทั่ง จอร์จ เมนเดล (Gregor Mendel ค.ศ.1822-1884) [22] ได้สร้างทฤษฎีของการถ่ายทอดลักษณะทางพันธุกรรม จากการศึกษาการผสมพันธุ์ถั่ว (Pisum satium) ที่มีลักษณะทางพันธุกรรมแตกต่างกันอย่างเด่นชัดที่ลักษณะ เช่น ต้นสูง กับต้นเตี้ย เป็นรุ่นพ่อแม่ (Parent generation ; P) ตรวจดูการแสดงลักษณะในรุ่นลูก F1 (First fillial generation) ทำการผสมกันเองในรุ่นลูก คุณลักษณะที่ปรากฏในรุ่นหลาน F2 (Second fillial gnceration) พบว่า การถ่ายทอดลักษณะไม่ได้เกิดจากผลของการนำเอาลักษณะต่าง ๆ มารวมกัน แต่มีหน่วยเฉพาะที่ปรากฏแน่นอนในการผสม เช่น ในการผสมพันธุ์ ลักษณะของลูกรุ่น F1 ที่ได้มีลักษณะเพียงลักษณะเดียวที่ปรากฏลักษณะที่ปรากฏในลูก F1 เรียกว่า ลักษณะเด่น (Dominance) ส่วนลักษณะที่หายไปเรียกว่า ลักษณะด้อย (Recessive) เมื่อผสมรุ่น F1 รุ่นหลาน F2 ที่ได้ ลักษณะด้อยที่หายไปจะปรากฏขึ้น โดย

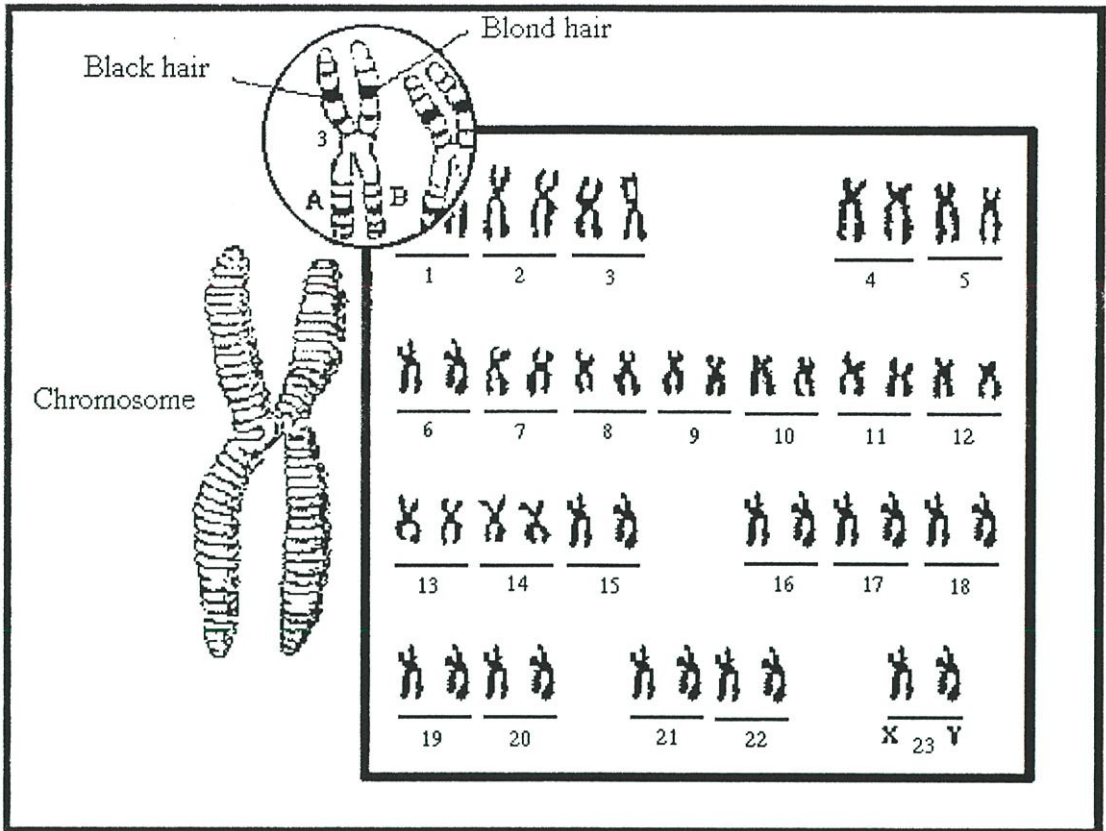
ในรุ่น F2 มีอัตราส่วนที่แสดงออกเป็นลักษณะเด่นต่อลักษณะด้อยเป็น 3 : 1 การถ่ายทอดลักษณะมีหน่วยควบคุมเฉพาะที่เรียกว่า ยีน ซึ่งอยู่เป็นคู่เรียกว่า อัลลีล (Allels) เมื่อเกิดกระบวนการผสมพันธุ์และการถ่ายทอดลักษณะไปยังรุ่นลูก จะเกิดการแยกตัวออก ลูกจะได้รับยีนส่วนหนึ่งจากแม่ และอีกส่วนหนึ่งจากพ่อ

สำหรับการวิจัยทางพันธุศาสตร์ปัจจุบันนี้ ได้ใช้เครื่องมือวิทยาศาสตร์สมัยใหม่ ศึกษาเพื่อให้ทราบถึงระบบภายในของยีน เช่น การวิเคราะห์การตอบสนองทางพันธุกรรม ซึ่งอยู่ภายในสิ่งมีชีวิต ข้อมูลทางพันธุกรรมส่วนใหญ่มักถูกบรรจุไว้ในโครโมโซม (Chromosome) ซึ่งแสดงให้เห็นถึงโครงสร้างทางเคมีของ โมเลกุลดีเอ็นเอ (DNA molecule) สามารถจำแนกรหัสทางพันธุกรรมได้จากยีน ซึ่งถือเป็นข้อมูลที่สำคัญยิ่งในการประกอบลักษณะทางเคมีของ โปรตีน ซึ่งตัวโปรตีนนี้เองกลับมีบทบาทสำคัญอย่างยิ่ง แต่ก็มีในหลาย ๆ กรณีอยู่ด้วยกันที่โปรตีนนั้นไม่อาจกำหนดสารประกอบทางพันธุกรรมได้ทั้งหมด แต่กลับอยู่ในฟีโนไทป์ตัวสุดท้ายของสิ่งมีชีวิตนั้น ๆ วิทยาศาสตร์คำว่า "เพื่อระบุรหัส" มักใช้กับยีนที่สามารถสร้างโปรตีนเองได้ โดยจะถูกเรียกว่า "รหัสถ่ายพันธุ์ของโปรตีน" โครโมโซม เป็นที่อยู่ของหน่วยพันธุกรรม ซึ่งทำหน้าที่ควบคุมและถ่ายทอดข้อมูล เกี่ยวกับ ลักษณะทางพันธุกรรมต่างๆ ของสิ่งมีชีวิต เช่น ลักษณะของเส้นผม ลักษณะดวงตา เพศและผิว การศึกษาลักษณะโครโมโซม จะต้องอาศัยการดูโดยใช้กล้องจุลทรรศน์ที่กำลังขยายสูง จึงจะสามารถ มองเห็นรายละเอียด ของโครโมโซมได้หน่วยพันธุกรรม หรือ ยีน ส่วนของ DNA ที่ทำหน้าที่ถ่ายทอดลักษณะทางพันธุกรรมปรากฏอยู่บนโครโมโซม ทำหน้าที่กำหนดลักษณะ ทางพันธุกรรมต่าง ๆ ของสิ่งมีชีวิต หน่วยพันธุกรรม จะถูกถ่ายทอดจากสิ่งมีชีวิต รุ่นก่อนสู่ลูกหลาน เช่น ความคุมกระบวนการเกี่ยวกับกิจกรรมทั่ว ๆ ไปทางชีวเคมี ภายในเซลล์ของสิ่งมีชีวิต ไปจนถึงลักษณะปรากฏที่พบเห็น หรือสังเกตได้ด้วยตา เช่น รูปร่างหน้าตาของเด็กที่มีบางส่วนเหมือนกับแม่ สีสันของดอกไม้ รสชาติของอาหารนาชนิด ล้วนแล้วแต่เป็นลักษณะที่บันทึกอยู่ในหน่วยพันธุกรรมทั้งสิ้น

การถ่ายทอดลักษณะทางพันธุกรรมเป็นการถ่ายทอดลักษณะต่าง ๆ ของสิ่งมีชีวิตที่เกิดขึ้นเมื่อมีการแบ่งตัวของเซลล์สิ่งมีชีวิต ซึ่งมี 2 แบบ คือ

2.1.1 การเพิ่มจำนวนเซลล์

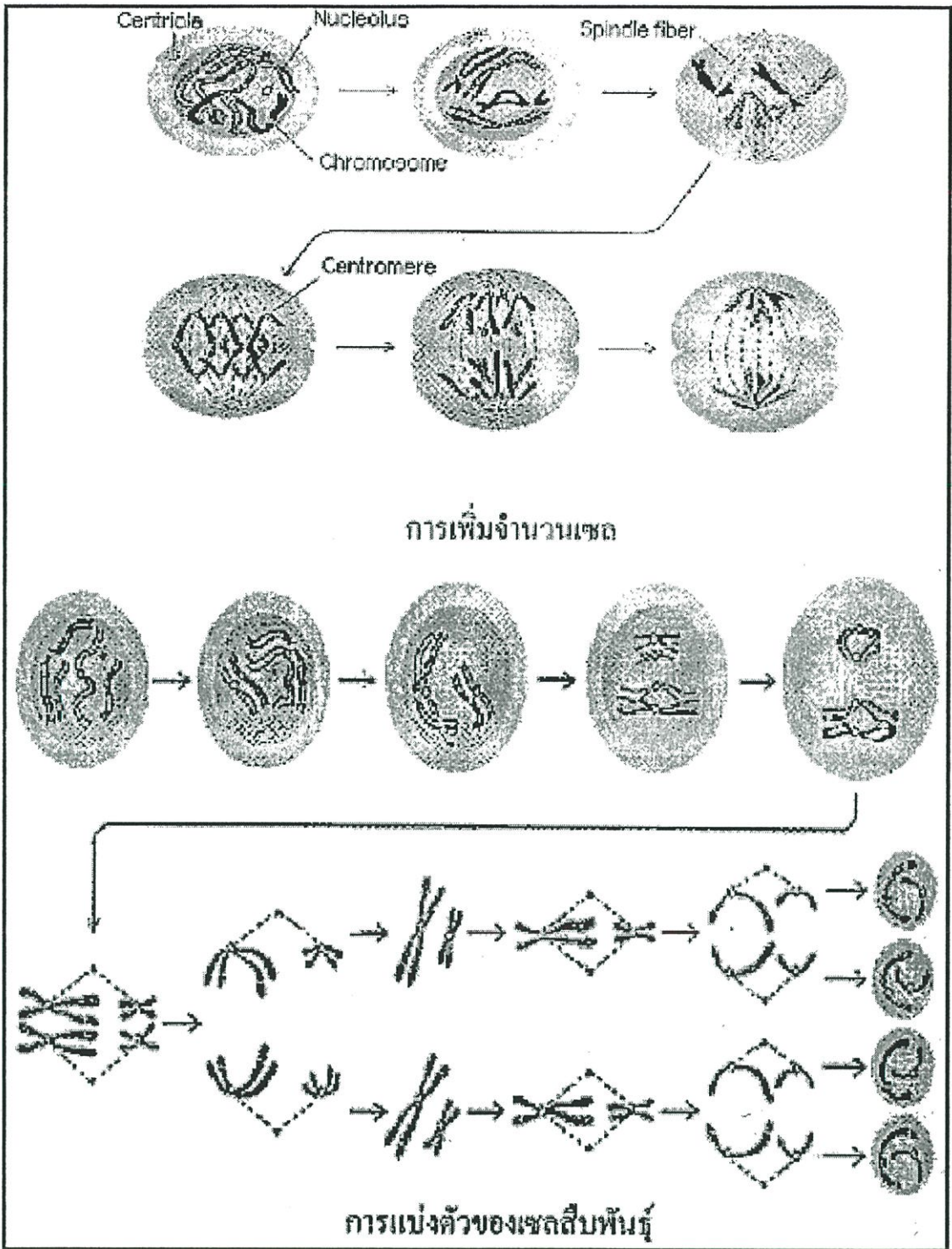
เป็นการแบ่งตัวแบบไมโทซิส (Mitosis) โดยโครโมโซมแต่ละตัวจะขยายตัวเพิ่มจำนวนตัวเองขึ้นเป็นสอง และเยื่อหุ้มนิวเคลียส (Nucleous) จะสลายลง เพื่อดึงแยกโครโมโซมที่เพิ่มจำนวนขึ้นออกจากโครโมโซมเดิมเป็นสองด้าน แล้วเยื่อหุ้มนิวเคลียสจะถูกสร้างขึ้นใหม่กลายเป็นเซลล์ใหม่ 2 เซลล์ที่มีโครโมโซมเหมือนเดิม ดังรูปที่ 2.1



รูปที่ 2.1 โครโมโซมทางพันธุศาสตร์

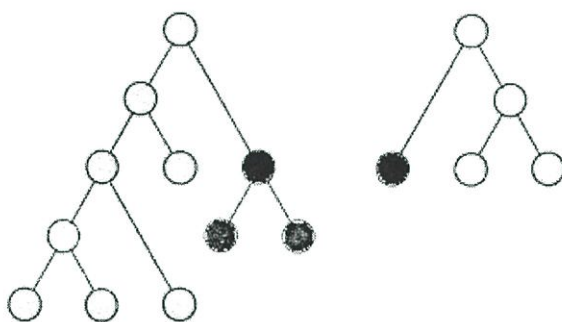
2.1.2 การแบ่งตัวของเซลล์สืบพันธุ์

เป็นการแบ่งตัวแบบไมโอซิส (Meiosis) โดยโครโมโซมจากเซลล์พ่อ 1 โครโมโซม และจากเซลล์แม่ 1 โครโมโซม จะเริ่มจับคู่กันที่โครโมโซมชนิดเดียวกัน ในขณะเดียวกันโครโมโซมแต่ละตัวทั้งที่มาจากพ่อและมาจากแม่ ต่างก็จำลองแบบของตนเพิ่มขึ้นมาอีกแต่ละโครโมโซม ทำให้ได้จำนวนโครโมโซมทั้งหมดเพิ่มขึ้นเป็นสองเท่า และดำเนินการทางพันธุกรรมจนถึงระยะแบ่งตัวโครโมโซมพ่อพร้อมกับแบบจำลองและโครโมโซมแม่พร้อมกับแบบจำลองที่ได้ จะแยกคู่ไปรวมกันเป็น 2 นิวเคลียส กลายเป็นเซลล์ใหม่ 2 เซลล์ ซึ่งจะแบ่งตัวต่อทันที โดยแต่ละโครโมโซมพ่อแยกตัวออกจากแบบจำลอง และโครโมโซมแม่ก็แยกตัวออกจากแบบจำลองรวมกันเป็นเซลล์ใหม่ 4 เซลล์ ดังรูปที่ 2.2

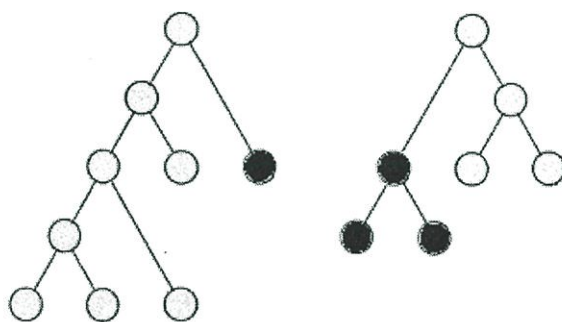


รูปที่ 2.2 การแบ่งตัวของเซลล์ทางพันธุศาสตร์

วิธีการทางพันธุศาสตร์ในระหว่างที่เกิดการแบ่งตัวของไมโอซิสนั้น โครโมโซมจะมีโอกาสแลกเปลี่ยนส่วนบางส่วนซึ่งกันและกัน อันเป็นสาเหตุที่ทำให้เกิดปรากฏการณ์ที่เรียกว่า ครอสโอเวอร์ (Crossover) ของลักษณะต่าง ๆ ขึ้น ซึ่งการครอสโอเวอร์นั้นเกิดขึ้นขณะที่มีการจำลองแบบเพิ่มขึ้น และเกิดขึ้นระหว่างโครโมโซมพ่อกับโครโมโซมแม่ ไม่ใช่เกิดขึ้นระหว่างโครโมโซมพ่อกับแบบจำลอง ซึ่งการครอสโอเวอร์จะทำให้เกิดการเปลี่ยนแปลงลักษณะของยีนส์ต่าง ๆ ของคู่โครโมโซมพ่อกับแม่ โดยเนื่องจากยีนส์เรียงตัวกันบนเส้น โครโมโซมนั้นไม่ได้อยู่กับอย่างหนาแน่น แต่มีระยะห่างกันอย่างไม่สม่ำเสมอ เพราะคุณสมบัติของยีนส์นั้นเป็น โมเลกุลของสารโปรตีนประกอบตัวกันทางเคมี ช่องว่างระหว่างยีนส์นี่เองจะเป็นตำแหน่งที่แตกออกมาได้เวลาจะครอสโอเวอร์ และแลกเปลี่ยนยีนส์ของโครโมโซมโดยส่วนที่อยู่หลังรอยแตก โดยจะถูกย้ายไปอยู่อีกโครโมโซมหนึ่งทั้งหมด นอกจากนี้ยังสามารถแตกอีกที่แห่งก็ได้ ซึ่งผลนั้นขึ้นอยู่กับความสามารถที่เชื่อมกันมากน้อยเพียงไรของช่องว่างระหว่างยีนส์ ดังรูปที่ 2.3



Crossover



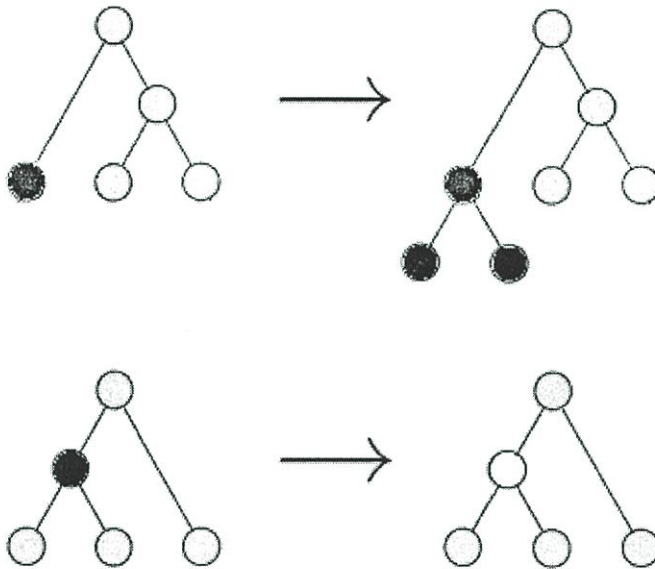
รูปที่ 2.3 การครอสโอเวอร์ทางพันธุศาสตร์

ประโยชน์ที่เกิดจากครอสโอเวอร์ที่เห็นได้ คือ ทำให้มีโอกาสที่จะลักษณะต่าง ๆ กันมาอยู่รวมกันได้หลายแบบมากขึ้น ทำให้สิ่งมีชีวิตรุ่นลูกที่เกิดขึ้นมามีความหลากหลายมากขึ้น และอาจทำให้มีโอกาสเกิดสิ่งมีชีวิตที่มีลักษณะต่าง ๆ ที่ดีพอเหมาะรวมอยู่ด้วยกันได้อย่างพอดี เหมาะสมกับสิ่งแวดล้อม ถ้าการเกิดเซลล์ใหม่โดยถ่ายทอดโครโมโซมไม่มี ครอสโอเวอร์แล้ว โครโมโซมใดเคยมี

ยีนส์ลักษณะใดก็จะมีลักษณะนั้นอยู่เรื่อย ๆ รุ่นลูกก็จะมีลักษณะเช่นเดียวกัน โอกาสที่สิ่งมีชีวิตนั้นจะเจริญ หรือปรับตัวให้ดีขึ้นย่อมมีได้ยากกว่าการเปลี่ยนลักษณะยีนส์ใหม่ หลาย ๆ แบบมากขึ้น

ลักษณะต่าง ๆ ของสิ่งมีชีวิตจะสามารถอยู่รอดได้โดยการคัดเลือกทางธรรมชาติ คือ คัดเลือกโครโมโซมที่มีลักษณะที่ทำให้สิ่งมีชีวิตแข็งแรงเพียงพอ หรือเหมาะสมต่อสภาพแวดล้อม ซึ่งจะสามารถอยู่รอดและถ่ายทอดไปยังลูกหลาน ดังนั้นการคัดเลือกของธรรมชาติเพื่อถ่ายทอดลักษณะทางพันธุกรรมเป็นเพียงส่วนประกอบของการเปลี่ยนแปลงของสิ่งมีชีวิตเท่านั้น มิวเตชัน (Mutation) หรือการผ่าเหล่า คือ การเปลี่ยนแปลงลักษณะของยีนส์ไปจากเดิมที่ควรเป็นไปตามการถ่ายทอด ซึ่งเป็นต้นเหตุของการเกิดลักษณะที่แปลก ๆ ออกไปมากมายสำหรับสิ่งมีชีวิตหนึ่ง ๆ เท่ากับเป็นการให้โอกาสแก่ธรรมชาติในการจะเลือกลักษณะใหม่ ๆ มากขึ้น ดังรูปที่ 2.4

Mutation



รูปที่ 2.4 การมิวเตชันทางพันธุศาสตร์

เนื่องจากกระบวนการวิวัฒนาการ โดยธรรมชาติเองนั้นช้ามาก เพราะกว่าที่ธรรมชาติจะปรับสภาพแวดล้อมให้สิ่งมีชีวิตค่อย ๆ ปรับตัวเองให้เหมาะสมนั้นมีโอกาสน้อยมาก การผ่าเหล่านั้นทุกลักษณะในแต่ละยีนส์ย่อมมีโอกาสที่จะเกิดความเปลี่ยนแปลงไปจากเดิมได้พอ ๆ กัน และถ้าเหมาะสมกับสภาพแวดล้อมขณะนั้นก็จะคงอยู่ต่อไป แต่ถ้าการเปลี่ยนแปลงเป็นลักษณะใดเกิดผิดจังหวะ คือไม่เหมาะสมกับสภาพขณะนั้น ๆ ก็จะไม่ถูก คัดเลือกและหายไป ซึ่งข้อสรุปนี้ได้จากการทดลองโดยการนำยีนส์ของแบคทีเรียมาผสมกัน และจัดสภาพแวดล้อมที่ไม่อำนวยต่อการผสม

กันแล้ว ยีนส์จะปรับตัวเองเพื่อเร่งขบวนการผสมพันธุ์จนได้ผลดีในรุ่นหลัง ๆ ตัวอย่างของการผ่าเหล่าในอดีต คือ การกำเนิดของปลาทองนั้น มีต้นกำเนิดมาจากการกลายพันธุ์หรือการผ่าเหล่าของปลาซิว และในปัจจุบันลักษณะใหม่ที่เกิดจากการผ่าเหล่าก็ยังคงมีให้เห็นอยู่ เช่น ความสามารถของเชื้อโรคแบคทีเรียในการต้านทานต่อยาฆ่าเชื้อ หรือเชลผลิตปกติอันเนื่องมาจากกัมมันตภาพรังสีต่าง ๆ ซึ่งมีผลต่อสารภายในเซลล์ ซึ่งทำให้ลักษณะของยีนส์ในเซลล์เปลี่ยนไป

2.2 อัลกอริทึมแบบเจเนติกพื้นฐาน

การนำเสนอวิธีการแก้ปัญหาโดยเจเนติกนั้น เริ่มขึ้นตั้งแต่ปี ค.ศ. 1975 โดย John Holland [32] โดยมีแนวคิดเลียนแบบการทำงานของธรรมชาติในด้านการคัดเลือกทางธรรมชาติ (Natural selection) โดยมีกฎอยู่ว่าสิ่งมีชีวิตที่แข็งแรงและเหมาะสมกับสภาพแวดล้อมจะมีโอกาสอยู่รอดได้มากกว่า ซึ่งทำให้สิ่งมีชีวิตต่าง ๆ มีการปรับตัวเองให้เข้าสภาพแวดล้อมได้ นั่นก็รวมถึงการเปลี่ยนแปลงส่วนประกอบต่าง ๆ ในโครโมโซม ซึ่งเป็นส่วนประกอบของสิ่งมีชีวิตเพื่อให้ได้โครโมโซมที่ดีขึ้นและส่งผ่านลักษณะที่ดีนั้นสู่รุ่นลูกหลานต่อไป โดยวิธีการที่ใช้ในการปรับปรุงโครโมโซมนั้นเรียกว่า การดำเนินการทางพันธุกรรมศาสตร์ (Genetic operation) ซึ่งประกอบไปด้วย การผสมพันธุ์หรือการครอสโอเวอร์ และการกลายพันธุ์หรือมิวเตชัน เป็นต้น จากแนวความคิดเกี่ยวกับวิวัฒนาการของสิ่งมีชีวิตที่ถ่ายทอดลักษณะที่เหมาะสมกับการอยู่รอดโดยผ่านโครโมโซมนี้ ทำให้ Holland เกิดแนวความคิดขึ้นมาว่าน่าจะนำไปใช้กับการทำงานร่วมกับคอมพิวเตอร์ได้ เขาจึงได้ทำการวิจัยโดยจำลองการทำงานกับแบบทดลองต่าง ๆ โดยมีจุดมุ่งหมายที่จะศึกษาระบบปรับปรุงการประมวลผลเอง (Self adaptive process) และเพื่อสร้างระบบผู้เชี่ยวชาญ (Artificial system software) จึงได้ค้นพบวิธีการใหม่เรียกว่า อัลกอริทึมแบบเจเนติก (Genetic Algorithm)

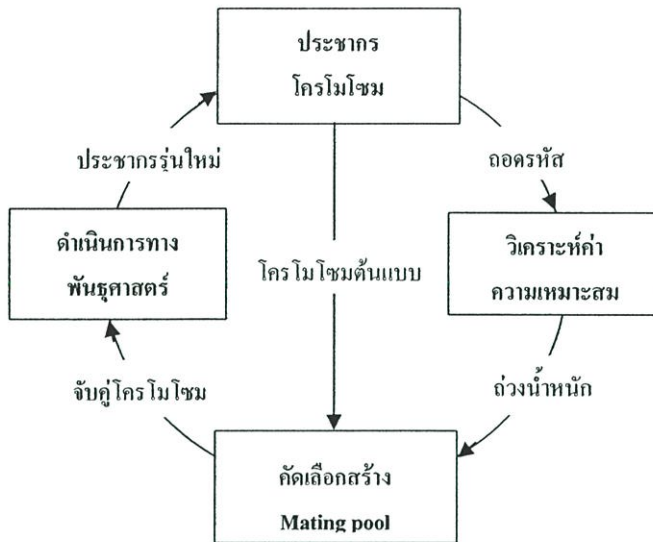
อัลกอริทึมแบบเจเนติก เป็นวิธีการค้นหาคำตอบโดยการเลียนแบบการทำงานของธรรมชาติและพันธุกรรมศาสตร์ ซึ่งอาศัยหลักการสุ่มเพื่อปรับปรุงความสามารถในการหาคำตอบที่ดีขึ้น การค้นหาคำตอบจากรุ่นหนึ่งไปยังรุ่นถัดไป ตามวิวัฒนาการของธรรมชาติคำตอบในรุ่นถัดไปที่เกิดขึ้นจะสร้างความสัมพันธ์ของโครงสร้างต่าง ๆ ที่ประกอบไปด้วยค่าตัวแปรที่เหมาะสมดีในรุ่นก่อน ดังนั้น จึงมีแนวโน้มของคำตอบที่ดีขึ้นเรื่อย ๆ จะเห็นได้ว่า อัลกอริทึมแบบเจเนติก อยู่บนพื้นฐานของวิธีการสุ่มแต่มีหลักการและประสิทธิภาพจากการคาดเดาคำตอบใหม่

2.2.1 หลักการของกระบวนการเชิงพันธุกรรม

กระบวนการเชิงพันธุกรรมเริ่มต้นจากการสุ่มวิธีในการหาคำตอบ แล้วคัดเลือกเอาคำตอบที่มีลักษณะดี กล่าวคือเป็นคำตอบจากวิธีที่ให้ค่าความเหมาะสม (Fitness) สูง จากนั้นจึงนำวิธีที่ผ่านการคัดเลือกมากระทำกันด้วยตัวดำเนินการทางพันธุกรรม (Genetic Operation) กันเพื่อให้ได้ลักษณะซึ่งให้คำตอบใหม่ออกมา คำตอบที่ได้จะถูกนำไปทดสอบค่าความเหมาะสมอีกเช่นกันเพื่อคัดเลือก

คำตอบที่ให้ค่าความเหมาะสมที่ดีขึ้น ค่าความเหมาะสม เป็นค่าความใกล้เคียงของคำตอบที่ต้องการ จากสถิติคำตอบเดิมที่ได้อยู่แล้ว ซึ่งแตกต่างจากวิธีการทั่วไป

เมื่อกำหนดรูปแบบโครโมโซม และฟังก์ชันความเหมาะสมของปัญหาแล้ว อัลกอริทึมแบบเจเนติกจะสามารถประมวลผลหาคำตอบของปัญหาได้ โดยสร้างวิวัฒนาการกลุ่มคำตอบในรุ่นต่อไปตามวัฏจักรการทำงานของ อัลกอริทึมแบบเจเนติก (Genetic Algorithm Cycle) ดังรูปที่ 2.5 ซึ่งมี 4 ขั้นตอนคือ



รูปที่ 2.5 วัฏจักรของเจเนติก อัลกอริทึม

i) สร้างประชากรโครโมโซมรุ่นเก่าตามแบบโครโมโซมที่กำหนดไว้ โดยประชากรต้นกำเนิด (Initial population) เกิดจากการสร้างชุดโครโมโซมต้นกำเนิด จากการสุ่มสร้างค่าแต่ละบิตของแต่ละโครโมโซม

ii) วิเคราะห์ค่าความเหมาะสมแต่ละโครโมโซม โดยถอดรหัสค่าตัวแปร พารามิเตอร์ต่าง ๆ ของแต่ละบิตในโครโมโซม และคำนวณค่าความเหมาะสมจากฟังก์ชันความเหมาะสมที่กำหนดไว้

iii) สร้าง mating pool คือ ชุดโครโมโซมต้นแบบหรือโครโมโซมพ่อแม่ที่สามารถอยู่รอดเป็นต้นแบบซึ่งอาศัยการจำลองการคัดเลือกทางธรรมชาติ โดยพิจารณาถ่วงน้ำหนักจากค่าความเหมาะสมของแต่ละโครโมโซม หากโครโมโซมใดมีค่าความเหมาะสมมากก็มีโอกาสถูกคัดเลือกเป็นต้นแบบมาก

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

การค้นหาคำตอบของ อัลกอริทึมแบบเจเนติก จะประมวลผลซ้ำตามวัฏจักรของ อัลกอริทึมแบบเจเนติก จนกว่าจะได้คำตอบที่พอใจตามกฎเกณฑ์ที่ตั้งไว้ หรือในระยะเวลาตามจำนวนรุ่นที่ดำเนินการที่ต้องการ [33]

2.2.2 ฟังก์ชันเป้าหมาย และฟังก์ชันความเหมาะสม

การหาคำตอบที่ดีที่สุดของปัญหาของอัลกอริทึมแบบเจเนติก มีพื้นฐานอยู่บนผลลัพธ์จากการหาคำตอบที่ผ่านมา ข้อได้เปรียบอัลกอริทึมแบบเจเนติก คือ ไม่มีสมการเชิงอนุพันธ์ที่ซับซ้อนเข้ามาเกี่ยวข้องในขั้นตอนการแก้ปัญหาเหมือนกับอัลกอริทึมแบบอื่น วิธีการของอัลกอริทึมแบบเจเนติกนั้นจะไม่พิจารณาขั้นตอนของการแก้ปัญหา แต่จะพิจารณาโดยตัดสินใจว่าคำตอบใหม่ที่ได้รับดีขึ้นหรือไม่ หรือเป็นคำตอบที่ใกล้เคียงคำตอบที่ต้องการหรือไม่จากฟังก์ชันเป้าหมาย (Object function : f) เนื่องจากแต่ละปัญหาสามารถกำหนดฟังก์ชันเป้าหมาย ซึ่งเป็นฟังก์ชันที่แสดงความสัมพันธ์ของแต่ละค่าพารามิเตอร์ เงื่อนไข หรือข้อกำหนดต่าง ๆ ของปัญหานั้น ระบุคำตอบใดคำตอบหนึ่งที่สามารถเป็นไปได้ ณ ค่าพารามิเตอร์ เงื่อนไข หรือข้อกำหนดชุดดังกล่าว สำหรับฟังก์ชันความเหมาะสม (Fitness Function : F) เป็นฟังก์ชันที่กำหนดค่าความเหมาะสม ของแต่ละโครโมโซมเปรียบเสมือนค่าความสามารถในการอยู่รอดของแต่ละโครโมโซม และเป็นฟังก์ชันที่กำหนดโอกาส หรือสัดส่วนที่แต่ละโครโมโซมที่เหมาะสมมีโอกาสจะถูกคัดเลือกมาน้อยเพียงใด ซึ่งฟังก์ชันความเหมาะสม จะเป็นฟังก์ชันที่แสดงถึง คำคำตอบที่เกิดขึ้นจากชุดตัวแปรของปัญหาของโครโมโซมนั้นดีเพียงใด โดยทั่วไปแล้วเรามักใช้ฟังก์ชันเป้าหมายเป็นฟังก์ชันความเหมาะสม หรืออาจใช้ฟังก์ชันเป้าหมายที่ถูกปรับให้เหมาะสมกับอัลกอริทึมแบบเจเนติกที่นำเสนอ เป็นฟังก์ชันความเหมาะสมก็ได้

2.2.3 รูปแบบโครโมโซม

ทราบกันแล้วว่าวิวัฒนาการทางธรรมชาติ และชีวิตวิทยานั้นเป็นความเปลี่ยนแปลงต่าง ๆ ของสิ่งมีชีวิตซึ่งจะเกิดขึ้นในโครโมโซม ดังนั้นจุดเริ่มต้นของการจำลองแบบทางธรรมชาติของอัลกอริทึมแบบเจเนติก เพื่อใช้แก้ปัญหาจึงเริ่มจาก การมองปัญหาเทียบเท่ากับโครโมโซมชนิดหนึ่งประกอบด้วยยีนส์ลักษณะต่าง ๆ ซึ่งหมายถึงลำดับข้อมูลต่าง ๆ ที่จะแปรความหมายแล้วให้คำตอบของปัญหาค่าหนึ่ง เรียกว่า การถอดรหัส ในการถอดรหัสนั้นจะขึ้นอยู่กับปัญหา และในปัจจุบันปัญหามักมีมากมายจึงทำให้รูปแบบของโครโมโซมมีความแตกต่างกันออกไปตามปัญหานั้น ๆ เช่น

- Binary Encoding [36] เป็นรูปแบบโครโมโซมเริ่มแรกที่น่ามาใช้แก้ปัญหาของเจเนติกอัลกอริทึม จึงทำให้รูปแบบโครโมโซมแบบนี้เป็นเรื่องธรรมดาที่สุด ลักษณะของ Binary Encoding คือ ทุกตำแหน่งของยีนของโครโมโซมจะมีค่าเป็น bit 0 หรือ 1 ตัวอย่างเช่น

โครโมโซม A :	1	0	0	0	1	1	1	0	0
--------------	---	---	---	---	---	---	---	---	---

โครโมโซม B :	1	1	0	0	0	1	0	0	1
--------------	---	---	---	---	---	---	---	---	---

ปัญหาที่ใช้รูปแบบโครโมโซมแบบนี้ในการแก้ไขปัญหา เช่น ปัญหาของ Knapsack [36]

• Value Encoding หรือเรียกว่า Direct Encoding [36] ทุกตำแหน่งของยีนของโครโมโซมจะมีค่าบางค่าซึ่งสามารถเชื่อมโยงไปยังปัญหาได้ เช่น ตัวอักษร จำนวนจริง คำสั่ง หรืออื่น ๆ รูปแบบโครโมโซมแบบนี้สามารถใช้ได้กับปัญหาที่ค่อนข้างซับซ้อนของค่า ตัวอย่างเช่น

โครโมโซม A	1.23	4.51	6.21	0.21	2.87	3.45	1.88	4.33	0.19
------------	------	------	------	------	------	------	------	------	------

โครโมโซม B	a	e	i	y	k	l	m	n	o
------------	---	---	---	---	---	---	---	---	---

โครโมโซม C	back	right	left	back	left	right	back	back	left
------------	------	-------	------	------	------	-------	------	------	------

สำหรับ Value Encoding เป็นตัวเลือกที่ดีที่สุดสำหรับปัญหาพิเศษบางอย่าง รูปแบบโครโมโซมแบบนี้ยังทำให้มีความจำเป็นที่จะต้องพัฒนาการครอสโอเวอร์ และการมิวเตชัน อีกด้วย

• Permutation encoding [36] รูปแบบโครโมโซมแบบนี้ใช้ในการลำดับของปัญหา ทุกตำแหน่งของยีนของโครโมโซม จะเป็นค่าของจำนวนนับที่แทนตำแหน่งในลำดับ เช่น ปัญหาของ Traveling Salesman [36]

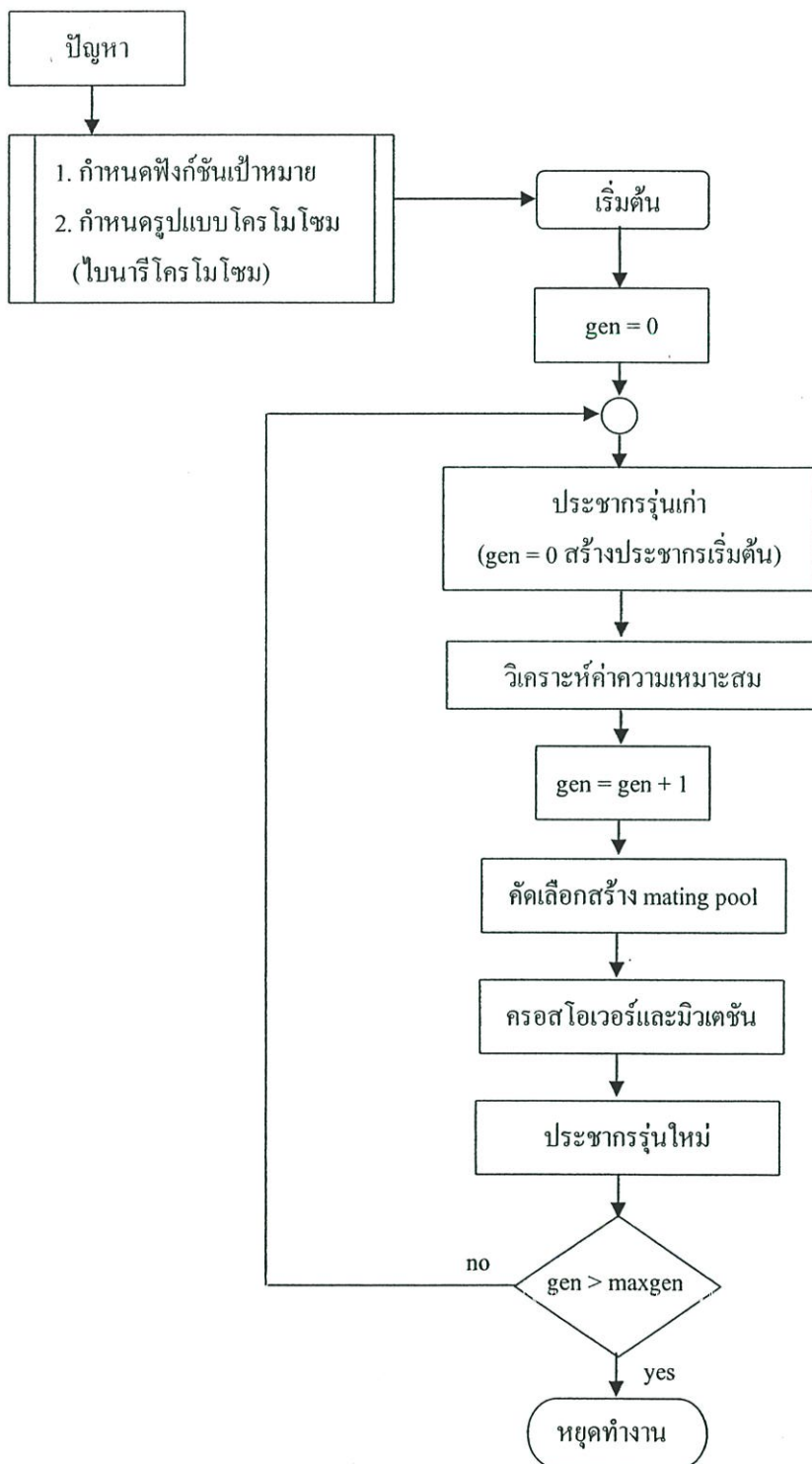
โครโมโซม A :	1	2	3	4	5	6	7	8	9
--------------	---	---	---	---	---	---	---	---	---

โครโมโซม B :	9	5	2	1	4	6	7	8	3
--------------	---	---	---	---	---	---	---	---	---

Permutation encoding ถูกใช้อย่างมากในการลำดับปัญหา สำหรับบางชนิดความถูกต้องของการครอสโอเวอร์ และการมิวเตชัน ต้องสร้างให้ตรงกับปัญหา

2.3 อัลกอริทึมแบบเจเนติกแบบง่าย

อัลกอริทึมแบบเจเนติกในยุคเริ่มแรกของ Holland นั้น [32] คือ อัลกอริทึมแบบเจเนติกแบบง่าย (Simple Genetic Algorithm) ซึ่งมีขั้นตอนพื้นฐานที่มีกระบวนการไม่มากนัก ง่ายต่อการศึกษาและทำความเข้าใจ ซึ่งหลักการทำงานเบื้องต้นของอัลกอริทึมแบบเจเนติกแบบง่าย แสดงดังไคอะแกรมในรูปที่ 2.6



รูปที่ 2.6 ไคอะแกรมของเจเนติก อัลกอริทึมแบบง่าย

จากรูปที่ 2.6 อัลกอริทึมแบบเจเนติกแบบง่ายเริ่มต้นจากการกำหนดฟังก์ชันเป้าหมาย/ฟังก์ชันความเหมาะสม และกำหนดรูปแบบโครโมโซมซึ่งเป็นโครโมโซมแบบไบนารี จากนั้นก็เริ่มต้นหาคำตอบโดยเริ่มจากสร้างโครโมโซมชุดแรก นำโครโมโซมชุดแรกที่ได้แต่ละตัวไปวิเคราะห์หาค่าความเหมาะสมของโครโมโซมแต่ละตัว เมื่อได้ค่าความเหมาะสมแล้วไปผ่านกระบวนการคัดเลือกโครโมโซม (โดยเรียงลำดับตามค่าความเหมาะสม) หลังจากนั้นจะผ่านกระบวนการทางพันธุศาสตร์ (ครอสโอเวอร์และมิวเตชัน) จะได้โครโมโซมชุดใหม่ เพื่อเป็นต้นแบบในการหาคำตอบต่อไป ทำเช่นนี้ไปเรื่อย ๆ จนกว่าจะได้คำตอบตามที่ต้องการ หรือจนกว่าจะครบรอบ (วิถัจกร) ที่กำหนด (Maxgen) ทั้งนี้ได้แบ่งการทำงานเป็น 2 ชั้น คือ ชั้นเตรียมการและชั้นการทำงาน โดยแต่ละชั้นตอนมีรายละเอียดดังนี้

2.3.1 ชั้นเตรียมการ เป็นส่วนของการวิเคราะห์รูปแบบของปัญหา เพื่อกำหนดฟังก์ชันเป้าหมายและฟังก์ชันความเหมาะสม พร้อมทั้งกำหนดรูปแบบโครโมโซมให้เหมาะสมกับแต่ละปัญหา ในชั้นเตรียมการนี้จะประกอบด้วย

i) การกำหนดฟังก์ชันเป้าหมาย (Object function : f) ฟังก์ชันเป้าหมาย คือ ฟังก์ชันที่นำไปใช้งานหรือหาคำตอบของแต่ละปัญหา โดยการแทนค่าด้วยค่าโครโมโซมหรือแทนด้วยรหัสที่ถอดได้จากแต่ละโครโมโซม

ii) กำหนดฟังก์ชันความเหมาะสม (Fitness function : F) เป็นการกำหนดฟังก์ชันหรือสมการเพื่อใช้วิเคราะห์ค่าความเหมาะสมของแต่ละโครโมโซมว่าโครโมโซมใดให้คำตอบ หรือมีความเหมาะสมดีที่สุด ซึ่งคำตอบที่ดีที่สุด คือ ค่าโครโมโซมที่มีค่าความเหมาะสมสูงสุด ($MAX(F)$) โดยฟังก์ชันความเหมาะสมจะเป็นฟังก์ชันเดียวกันกับฟังก์ชันเป้าหมายก็ได้

iii) กำหนดรูปแบบโครโมโซม รูปแบบโครโมโซมของ อัลกอริทึมแบบเจเนติกแบบง่าย นั้นเป็นแบบไบนารี โดยค่าตัวแปรหรือพารามิเตอร์ของปัญหาจะถูกแปลงให้อยู่ในรูปของไบนารีโครโมโซม คือ ประกอบด้วยบิต การกำหนดว่าจะใช้โครโมโซมขนาดกี่บิตนั้นขึ้นกับ ความเหมาะสมของแต่ละปัญหา

2.3.2 ขั้นตอนการทำงาน เป็นขั้นตอนพื้นฐานเบื้องต้นของอัลกอริทึมแบบง่าย ประกอบด้วย

i) ประชากรรุ่นเก่า (Old population)

ii) วิเคราะห์ค่าความเหมาะสม

iii) การคัดเลือก

iv) ดำเนินการทางพันธุศาสตร์

v) ประชากรรุ่นใหม่

แต่ละขั้นตอน มีรายละเอียดดังต่อไปนี้

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

ii) วิเคราะห์ค่าความเหมาะสม เป็นขั้นตอนของการแทนค่าโครโมโซมหรือแทนค่าด้วยรหัสที่ถอดได้ของโครโมโซม ในฟังก์ชันความเหมาะสม (F) ที่กำหนดไว้ เพื่อกำหนดค่าความเหมาะสมตามฟังก์ชันความเหมาะสมของปัญหาของแต่ละโครโมโซม พร้อมทั้งเก็บค่าความเหมาะสมของแต่ละโครโมโซมไว้สำหรับคัดเลือกต่อไป

iii) การคัดเลือก เป็นขั้นตอนที่จำลองแบบการคัดเลือกทางธรรมชาติเพื่อสร้าง Mating pool โดยคัดเลือกชุดโครโมโซมรุ่นเก่าให้เป็นโครโมโซมต้นแบบหรือโครโมโซมพ่อ-แม่ เพื่อใช้สร้างโครโมโซมลูกเป็นรุ่นต่อไป สำหรับการคัดเลือกของ อัลกอริทึมแบบเจเนติกแบบง่าย เป็นแบบอ้างอิงค่าความเหมาะสม (Fitness-based Selection) โดยพิจารณาค่าความเหมาะสมเป็นตัวตัดสินว่าโครโมโซมใดในรุ่นเก่ามีโอกาสจะถูกเลือกเป็นโครโมโซมพ่อ-แม่อย่างน้อยเพียงใด โครโมโซมที่มีค่าความเหมาะสมที่ดีจะถูกกำหนดน้ำหนักค่าความน่าจะเป็นที่จะถูกเลือกแต่ละครั้งสูง การกำหนดค่าความน่าจะเป็นที่จะถูกเลือกต่อการสุ่มเลือกแต่ละครั้ง (Probability of Selected Value : pselect) ของแต่ละโครโมโซม โดยกำหนดจากค่าความเหมาะสม เทียบกับผลรวมของค่าความเหมาะสมทั้งหมด ดังสมการที่ (2.1)

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

ซึ่งสามารถคำนวณค่าที่คาดหวังที่จะสุ่มได้ (Expected Value : E) ของแต่ละโครโมโซมในแต่ละรุ่น ดังสมการที่ (2.2)

$$E_i = pselect_i * popsize \quad (2.2)$$

สำหรับวิธีการสุ่มโครโมโซมต้นแบบของ อัลกอริทึมแบบเจเนติกแบบง่าย เป็นแบบจำลองการหมุนวงล้อถ่วงน้ำหนัก (Roulet Wheel : RW) ซึ่งกำหนดขนาดแต่ละช่องของวงล้อตามค่าความน่าจะเป็นที่จะสุ่มได้ในแต่ละครั้งของแต่ละโครโมโซม ซึ่งมีวิธีการดังนี้

- หาค่าความเหมาะสมของแต่ละโครโมโซม (Fitness value)
- หาค่าความน่าจะเป็นที่จะถูกเลือก (pselect) ในแต่ละครั้งของแต่ละโครโมโซม

- หาความถี่สะสม (q) ของค่าความน่าจะเป็นที่จะถูกเลือก (pselect) ของแต่ละโครโมโซม ดังสมการที่ (2.3)

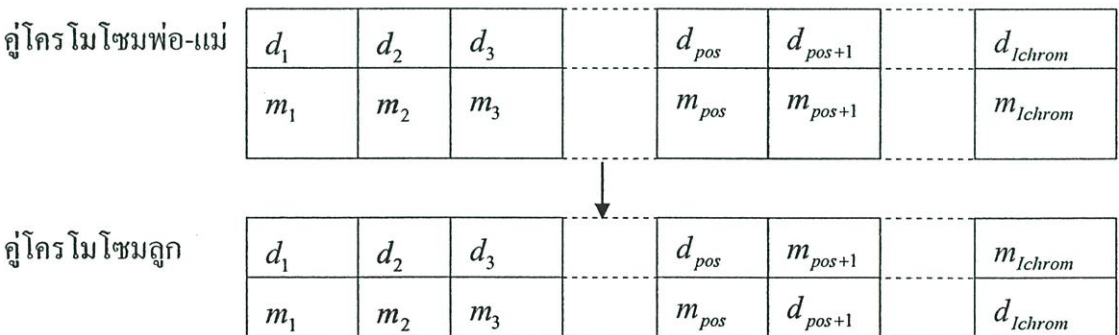
$$q_i = \sum_{j=1}^i pselect_j \tag{2.3}$$

- สร้างเลขสุ่มจำนวนจริง (r) มีค่าอยู่ในช่วง [0.0,1.0]
- เลือกโครโมโซมลำดับที่ r ซึ่ง r มีค่าอยู่ระหว่าง q_{i-1} และ q_i

iv) ดำเนินการทางพันธุศาสตร์ เป็นขั้นตอนที่จำลองแบบธรรมชาติทางพันธุกรรม ซึ่งตัวดำเนินการทางพันธุศาสตร์ของ อัลกอริทึมแบบเจเนติกแบบง่าย คือ ครอสโอเวอร์และมิวเตชัน ซึ่งมียุทธศาสตร์ดังนี้

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

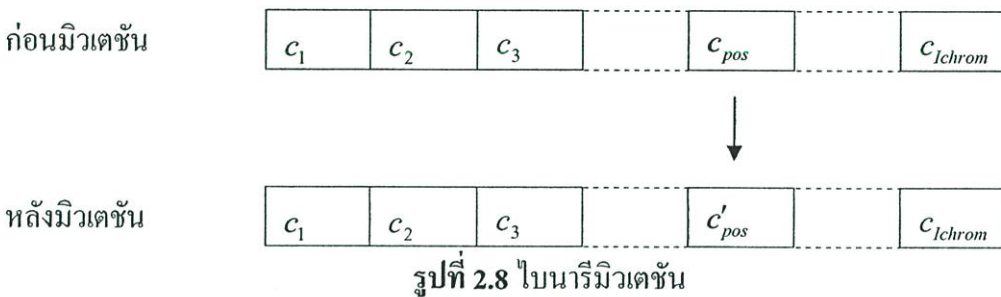
- ขั้นตอนแรก : สุ่มจับคู่โครโมโซมพ่อ-แม่ใน Mating pool ที่สร้างไว้จากการคัดเลือก
- ขั้นตอนที่สอง : สร้างเลขสุ่มจำนวนจริง (r) มีค่าอยู่ในช่วง [0.0 , 1.0] โดยถ้า $r \leq Pc$ แล้วโครโมโซมพ่อ-แม่ นั้นจึงมีการครอสโอเวอร์
- ขั้นตอนที่สาม : ครอสโอเวอร์โดยการแลกเปลี่ยนส่วนของโครโมโซมพ่อ-แม่นั้น ซึ่งการครอสโอเวอร์ของ อัลกอริทึมแบบเจเนติกแบบง่าย เป็นการครอสโอเวอร์แบบ 1 จุด (One-point Crossover) แสดงดังรูปที่ 2.7 ดังนี้
 - สุ่มเลือกตำแหน่ง pos เป็นตำแหน่งที่ครอสโอเวอร์ ซึ่ง pos มีค่าอยู่ในช่วง [1, Ichrom-1]
 - แลกเปลี่ยนค่าในแต่ละบิตของคู่โครโมโซมพ่อ-แม่ตั้งแต่ตำแหน่งที่ pos+1 ถึง Ichrom ซึ่งจะทำให้เกิดโครโมโซมลูกใหม่ 2 โครโมโซม



รูปที่ 2.7 ครอสโอเวอร์แบบ 1 จุด

จำนวนการครอสโอเวอร์ในแต่ละรุ่นขึ้นอยู่กับค่าที่กำหนดค่า P_c ซึ่งแตกต่างกันในแต่ละปัญหา เช่น ถ้าจำนวนประชากรแต่ละรุ่น P_{size} เท่ากับ 30 โครโมโซม และกำหนดให้ $P_c=0.6$ แล้วจำนวนการครอสโอเวอร์ในแต่ละรุ่นเท่ากับ $P_c * (P_{size}/2) = 0.6 * (30/2) = 9$ ครั้ง (การครอสโอเวอร์ 1 ครั้ง เกิดจากโครโมโซม 2 โครโมโซม)

- มิวเตชัน เป็นตัวดำเนินการผ่าเหล่าตัวหนึ่งที่จะช่วยให้โครโมโซม มีค่าความเหมาะสมดีขึ้น หลังจากการครอสโอเวอร์ โดยกลับค่าบิตเป็นค่าใหม่ในตำแหน่งบิตที่สุ่มได้ ตามอัตราความน่าจะเป็นของการมิวเตชันในแต่ละบิต (Probability of Mutation : P_m) ที่กำหนด สำหรับการมิวเตชันของอัลกอริทึมแบบเจเนติกแบบง่าย นั้นเป็นแบบไบนารีมิวเตชัน (Binary Mutation) โดยกลับค่าบิตเป็นค่าคอมพลีเมนต์ คือ จาก 0 เป็น 1 หรือจาก 1 เป็น 0 ดังรูปที่ 2.8



จำนวนการมิวเตชันในแต่ละรุ่นขึ้นอยู่กับค่าที่กำหนดค่า P_m ซึ่งแตกต่างกันในแต่ละปัญหา เช่น ถ้าจำนวนประชากรแต่ละรุ่น P_{size} เท่ากับ 30 โครโมโซมประกอบด้วย 5 บิต และกำหนดให้ $P_m=0.02$ แล้ว จำนวนการมิวเตชันในแต่ละกลุ่มเท่ากับ $P_m * P_{size} * Ichrom = 0.02 * 30 * 5 = 3$ บิต

v) ประชากรรุ่นใหม่ เป็นชุดโครโมโซมลูกที่เกิดจากขั้นตอนของวิวัฒนาการต่าง ๆ ทั้งหมด ซึ่งประชากรรุ่นใหม่ทั้งหมดที่เกิดขึ้นจะถูกถ่ายทอดกลายเป็นประชากรรุ่นเก่าสำหรับวิวัฒนาการในรุ่นถัดไป ซึ่งเรียกววิวัฒนาการแบบนี้ว่า การถ่ายทอดแบบทั่วไปหรือรีโพลิดิกชันแบบทั่วไป (General reproduction) กระบวนการต่างๆ จะถูกทำงานซ้ำ ๆ จนกระทั่งถึงรุ่นที่มากที่สุด (Max generation : maxgen) ที่ต้องการ

2.4 การประยุกต์เจเนติก อัลกอริทึมแบบง่าย

ในหัวข้อนี้เป็นตัวอย่างการประยุกต์ใช้อัลกอริทึมแบบเจเนติกแบบง่าย โดยยกตัวอย่างปัญหาการหาค่าสูงสุดของฟังก์ชัน $f(x) = x^2$ โดยมีขั้นตอนการทำงานดังต่อไปนี้

i) กำหนดฟังก์ชันและฟังก์ชันความเหมาะสม

- ฟังก์ชันเป้าหมาย คือ $f(x) = x^2$
- ฟังก์ชันความเหมาะสม $F = x^2$

- ค่าที่เหมาะสมที่สุด คือ $\text{Max}(F)$

โดยที่ x คือ ค่าตัวเลขจำนวนเต็มของโครโมโซม

ii) กำหนดรูปแบบโครโมโซม

รูปแบบโครโมโซมเป็นโครโมโซมแบบไบนารีขนาด 16 บิต (มีค่าระหว่าง 0-65536)

ตัวอย่างเช่น

0000000000000100 [4] คือ โครโมโซมไบนารีขนาด 16 บิต ของเลขจำนวนเต็ม 4

0000000001001110 [78] คือ โครโมโซมไบนารีขนาด 16 บิต ของเลขจำนวนเต็ม 78

0000000001011011 [91] คือ โครโมโซมไบนารีขนาด 16 บิต ของเลขจำนวนเต็ม 91

iii) เริ่มขั้นตอนการทำงาน

- กำหนดให้จำนวนวิวัฒนาการในการหาคำตอบ (maxgen) = 2 วิวัฒนาการ

- จำนวนโครโมโซมใน 1 ชุด (Popsiz) = 4

- อัตราความน่าจะเป็นของการครอสโอเวอร์ (P_c) = 0.08

- อัตราความน่าจะเป็นของการมิวเตชันในแต่ละบิต (P_m) = 0.05

ขั้นตอนการทำงานมีดังนี้

1. สร้างโครโมโซมรุ่นแรก สำหรับเป็นต้นแบบโดยทำการสุ่ม (Random) โครโมโซมจำนวน 4 ตัว (เท่ากับ Popsiz ที่กำหนด) ตัวอย่างตามตารางที่ 2.1

ตารางที่ 2.1 ตัวอย่างโครโมโซมรุ่นแรก

ลำดับ(i)	โครโมโซมต้นแบบ [ค่าตัวเลข]
1	0000000000000100 [4]
2	0000000000000101 [5]
3	0000000000001001 [9]
4	0000000000000110 [6]

ตารางที่ 2.1 ลำดับ (i) คือ ตัวเลขลำดับของโครโมโซม ซึ่งจะใช้อ้างอิงโครโมโซม

2. วิเคราะห์ค่าความเหมาะสม

ทำการวิเคราะห์หาค่าความเหมาะสม (Fitness value) ของโครโมโซมรุ่นแรก ซึ่งผลการวิเคราะห์แสดงได้ดังตารางที่ 2.2

ตารางที่ 2.2 ตัวอย่างการหาค่าความเหมาะสมของโครโมโซม

ลำดับ(i)	โครโมโซมต้นแบบ	x	$F=x^2$	ค่าความเหมาะสม (Fitness value)
1	0000000000000100 [4]	4	$4 * 4$	16
2	0000000000000101 [5]	5	$5 * 5$	25
3	0000000000001001 [9]	9	$9 * 9$	81
4	000000000000110 [6]	6	$6 * 6$	36

ตารางที่ 2.2 แสดงการคำนวณหาค่าความเหมาะสม (Fitness value) ของโครโมโซมแต่ละตัว โดย ค่าความเหมาะสม (Fitness value) เป็นค่าที่ได้จากการแทนค่า x ในฟังก์ชันความเหมาะสม (Fitness value) ในที่นี้ฟังก์ชันความเหมาะสมคือ $F = x^2$ จากตารางที่ 2.2 พบว่าโครโมโซมลำดับที่ 3 มีค่าความเหมาะสมมากที่สุด คือ เท่ากับ 81 และที่น้อยที่สุดคือโครโมโซมลำดับที่ 1 ซึ่งมีค่าเป็น 16

3. ทำการคัดเลือกโครโมโซม โดยสร้าง Mating Pool โดยคัดเลือกชุดโครโมโซมรุ่นเก่าให้เป็นโครโมโซมต้นแบบ เพื่อใช้สร้างโครโมโซมรุ่นต่อไป โดยการพิจารณาค่าความเหมาะสมของแต่ละโครโมโซมเป็นตัวตัดสินว่า โครโมโซมใดในรุ่นเก่ามีโอกาสจะถูกเลือกเป็นโครโมโซมต้นแบบมากน้อยเพียงใด โครโมโซมที่มีค่าความเหมาะสมที่ดีจะถูกกำหนดน้ำหนักค่าความน่าจะเป็นที่จะถูกเลือกแต่ละครั้งสูง ซึ่งขั้นตอนการคัดเลือกจะประกอบด้วย

- หาค่าความน่าจะเป็นที่จะถูกเลือก (pselect) ของแต่ละ โครโมโซม
- หาค่าคาดหวังที่จะสุ่มได้ (E) ของแต่ละ โครโมโซม
- หาค่าความถี่สะสม (q) ของค่าความน่าจะเป็นที่จะถูกเลือก (pselect) ของแต่ละ โครโมโซม
- สร้างเลขสุ่มจำนวนจริง (r) มีค่าอยู่ในช่วง [0.0, 1.0]
- กำหนดค่าลำดับโครโมโซมที่จะถูกเลือก (Sequence) เป็นค่าที่ได้หลังจากการนำค่า r ไปเปรียบเทียบกับค่าความถี่สะสม (q) ว่าอยู่ในความถี่สะสมช่วงใด ($q_{i-1} < r < q_i$) ซึ่งจะได้ค่าลำดับของโครโมโซมนั้น เป็นโครโมโซมที่จะถูกคัดเลือกในรุ่นต่อไป ตัวอย่างในตารางที่ 2.3 ค่า r ที่สุ่มได้ในลำดับที่ 1 เป็น 0.52934 เมื่อนำค่า 0.52934 ไปเปรียบเทียบกับค่าความถี่สะสม (q) พบว่าอยู่ในช่วง q ลำดับที่ 3 ซึ่งมีค่าความถี่สะสมเท่ากับ 0.77215 จะได้ว่า ค่าลำดับโครโมโซมที่ถูกเลือกในลำดับที่ 1 มีค่าลำดับเท่ากับ 3 และจะได้ว่าโครโมโซมที่ถูกเลือก คือ 000000000001001 [9] ซึ่งเป็นโครโมโซมในลำดับที่ 3

- ทำการเลือกโครโมโซม โดยเลือกโครโมโซมตามค่าลำดับที่จะถูกเลือก (Sequence) เมื่อถึงขั้นตอนนี้จะได้โครโมโซมที่ถูกเลือก (Sequence Chromosome)

ตัวอย่างขั้นตอนการคัดเลือกเพื่อสร้าง Mating pool แสดงได้ดังตารางที่ 2.3 และตารางที่ 2.4

ตารางที่ 2.3 ตัวอย่างการคัดเลือกโครโมโซมต้นแบบ

ลำดับ (i)	โครโมโซมต้นแบบ	x	ค่าความเหมาะสม (Fitness value)	ค่าความน่าจะเป็นที่จะถูกเลือก (pselect)	ค่าคาดหวังที่จะสุ่มได้ (E)	ความถี่สะสม (q)	เลขคู่ (r)	ลำดับโครโมโซมที่ถูกเลือก	โครโมโซมที่ถูกเลือก
1	00000000000100 [4]	4	16	0.10127	0.40506	0.10127	0.52934	3	000000000001001 [9]
2	00000000000101 [5]	5	25	0.15823	0.63291	0.25949	0.83938	4	00000000000110 [6]
3	000000000001001 [9]	9	81	0.51266	2.05063	0.77215	0.63524	3	000000000001001 [9]
4	000000000000110 [6]	6	36	0.22785	0.91139	1	0.62795	3	0000000000001001 [9]
	ผลรวม (Sum of F)		158						
	ค่าน้อยที่สุด (Min of F)		16						
	ค่ามากที่สุด (Max of F)		81						
	ค่าเฉลี่ย (Average of F)		39.5						

ตารางที่ 2.4 โครโมโซมต้นแบบใน Mating pool

ลำดับโครโมโซม (i)	1	2	3	4
โครโมโซมต้นแบบ	00000000000100 [4]	00000000000101 [5]	000000000001001 [9]	00000000000110 [6]
x	4	5	9	6
แทนค่า x ในฟังก์ชันความเหมาะสม $F=x^2$	4 * 4	5 * 5	9 * 9	6 * 6
ค่าความเหมาะสม (Fitness value : F_i)	16	25	81	36
ค่าความน่าจะเป็นที่จะถูกเลือก (pselect) = $F_i / \text{Sum of F}$	16 / 158 = 0.10127	25 / 158 = 0.15823	81 / 158 = 0.51266	36 / 158 = 0.22785
ความถี่สะสม (q)	0.10127	0.25949	0.77215	1
เลขคู่ (r) เป็นเลขจำนวนจริงที่ได้จากการสุ่ม	0.52934	0.83938	0.63524	0.62795
ลำดับโครโมโซมที่ถูกเลือก ($q_{i-1} \leq r \leq q_i$)	3	4	3	3
โครโมโซมที่ถูกเลือกใน Mating pool	000000000001001 [9]	00000000000110 [6]	000000000001001 [9]	000000000001001 [9]

4. ดำเนินการทางพันธุศาสตร์ ขั้นตอนนี้เป็น การนำโครโมโซมต้นแบบที่ได้จากการคัดเลือก ใน Mating pool ผ่านขบวนการทางพันธุศาสตร์ ซึ่งได้แก่ การครอสโอเวอร์ และมิวเตชัน

ครอสโอเวอร์ เป็นการแลกเปลี่ยนโครโมโซมระหว่างโครโมโซมต้นแบบสองตัว คือ โครโมโซมพ่อและแม่ โดยการแลกเปลี่ยนนั้นจะทำการแลกเปลี่ยนโครโมโซมพ่อและแม่ ตามการ กำหนดอัตราความน่าจะเป็นของการครอสโอเวอร์ (Probability of Crossover : Pc) เพื่อสร้างชุด โครโมโซมรุ่นใหม่คือโครโมโซมลูก

ตารางที่ 2.5 การครอสโอเวอร์โครโมโซมต้นแบบเพื่อสร้างโครโมโซมรุ่นใหม่

ลำดับ	โครโมโซมใน Mating pool	r	pos	crossover	โครโมโซมหลัง ครอสโอเวอร์
1	0000000000001 001 [9]	← 0.01	13	crossover	0000000000001 110 [14]
2	0000000000000 110 [6]	←			0000000000000 001 [1]
3	0000000000001001 [9]	0.64	5	no	0000000000001001 [9]
4	0000000000001001 [9]				0000000000001001 [9]

จากตารางที่ 2.5 โครโมโซมต้นแบบ คือ โครโมโซมที่ได้จากการคัดเลือกใน Mating pool การแลกเปลี่ยนโครโมโซมจะทำการแลกเปลี่ยนระหว่างโครโมโซมลำดับที่ 1 กับ 2, 3 กับ 4 โดยการครอสโอเวอร์นี้จะเริ่มจากสุ่มหาค่า r แต่ละคู่ เปรียบเทียบกับค่า Pc ที่กำหนด หากค่า r ที่สุ่มได้น้อยกว่า Pc ก็จะทำให้การแลกเปลี่ยนโครโมโซมกัน ณ ตำแหน่งที่ถัดจาก pos ที่สุ่มได้ไปอีก 1 ตำแหน่ง เช่น โครโมโซมลำดับที่ 1 กับ 2 สุ่มค่า r ได้เป็น 0.01 จะเห็นว่า r (0.01) น้อยกว่า Pc (0.08) ก็จะเกิดการแลกเปลี่ยนโครโมโซมหลังตำแหน่งที่สุ่มได้ (pos = 13) ก็จะแลกเปลี่ยนโครโมโซมตั้งแต่ตำแหน่งที่ 14 เป็นต้นไป โครโมโซมลำดับที่ 3 กับ 4 สุ่มค่า r ได้เป็น 0.64 ซึ่งน้อยกว่ามากกว่า Pc (0.08) จึงไม่เกิดการแลกเปลี่ยนโครโมโซม ดังนั้นโครโมโซมหลังครอสโอเวอร์ก็ยังเป็นโครโมโซมเดิม

- มิวเตชัน เป็นตัวดำเนินการผ่าเหล่าที่อาจช่วยให้โครโมโซม มีค่าความเหมาะสมดีขึ้น หลังจากการครอสโอเวอร์ โดยกลับค่าบิตเป็นค่าใหม่ในตำแหน่งบิตที่สุ่มได้ ตามอัตราความน่าจะเป็นของการมิวเตชันในแต่ละบิต (Probability of Mutation : Pm) ที่กำหนด สำหรับการมิวเตชันของอัลกอริทึมแบบเจเนติกแบบง่าย นั้นเป็นแบบไบนารีมิวเตชัน (Binary Mutation) โดยกลับค่าบิตเป็นค่าคอมพลีเมนต์ คือ จาก 0 เป็น 1 หรือจาก 1 เป็น 0 ในขั้นตอนนี้หลังจากที่ได้โครโมโซมรุ่นใหม่จากการครอสโอเวอร์มาแล้ว จะทำการมิวเตชันโครโมโซมที่ได้นั้น เพื่อปรับปรุงโครโมโซมรุ่นใหม่ขึ้นอีกครั้ง จากตารางที่ 2.6 แสดงการมิวเตชันโครโมโซมที่ได้จากการครอสโอเวอร์ โดยเริ่มจาก สุ่มค่าความน่าจะเป็นที่จะมิวเตชัน (r) ในแต่ละบิตของแต่ละโครโมโซม หากค่า r ที่สุ่มได้นั้น มีค่าน้อยกว่า Pm ที่กำหนด ก็จะทำให้การกลับบิต จาก 0 เป็น 1 หรือ จาก 1 เป็น 0 ตัวอย่างการมิวเตชันในตารางที่ 2.6 เช่น โครโมโซมลำดับที่ 1 คือ 0000000000001110 [14]

บิตที่	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ก่อน	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
r	0.74	(0.03)	0.34	0.60	(0.01)	0.05	0.31	0.32	(0.03)	0.86	0.49	0.65	0.35	0.39	0.48	0.08
หลัง	0	1	0	0	1	0	0	0	1	0	0	0	1	1	1	0

จะเห็นว่าในตำแหน่งบิตที่ 2 ค่า $r(0.03)$ น้อยกว่า $P_m(0.05)$ จะกลับค่าบิต จาก 0 เป็น 1 ตำแหน่งบิตที่ 5 ค่า $r(0.01)$ น้อยกว่า $P_m(0.05)$ จะกลับค่าบิต จาก 0 เป็น 1 และตำแหน่งบิตที่ 9 ค่า $r(0.03)$ น้อยกว่า $P_m(0.05)$ ก็จะกลับค่าบิต จาก 0 เป็น 1 ซึ่งทำให้โครโมโซมใหม่ที่ได้หลังการมีวตชน เปลี่ยนค่าจากเดิม 0000000000001110 [14] เป็น 0100100010001110 [18574]

ตารางที่ 2.6 ตัวอย่างการมีเวตชันโครโมโซม

ลำดับ	โครโมโซมก่อนมีเวตชัน (หลังครอสโอเวอร์)	ค่าตัวเลขคู่ (r)																โครโมโซมหลังมีเวตชัน
		bit 1	bit 2	bit 3	bit 4	bit 5	bit 6	bit 7	bit 8	bit 9	bit 10	bit 11	bit 12	bit 13	bit 14	bit 15	bit 16	
1	000000000001110 [14]	0.74	(0.03)	0.34	0.60	(0.01)	0.05	0.31	0.32	(0.03)	0.86	0.49	0.65	0.35	0.39	0.48	0.08	0100100010001110 [18574]
2	0000000000000001 [1]	0.52	0.47	0.44	0.60	0.73	0.71	0.68	0.49	0.97	0.06	0.94	0.13	0.80	0.28	0.24	0.07	0000000000000001 [1]
3	0000000000001001 [9]	0.26	0.68	0.73	0.09	0.76	0.28	0.42	0.68	0.86	0.76	0.21	0.50	0.94	0.53	0.15	0.33	0000000000001001 [9]
4	0000000000001001 [9]	0.48	0.08	0.24	0.12	0.05	0.91	0.88	0.54	0.84	0.50	0.06	0.70	0.51	0.29	0.58	0.41	0000000000001001 [9]

5. ประชากรรุ่นใหม่ หลังจากผ่านขบวนการทางพันธุศาสตร์ต่าง ๆ แล้วจะได้โครโมโซมรุ่นใหม่ ซึ่งใช้เป็นโครโมโซมต้นแบบสำหรับการดำเนินการในวัฏจักรต่อไป จากตัวอย่างข้างต้นจะได้โครโมโซมรุ่นใหม่ดังตารางที่ 2.7(a) และ (b)

ลำดับ(i)	โครโมโซม
1	0100100010001110 [18574]
2	0000000000000001 [1]
3	0000000000001001 [9]
4	0000000000001001 [9]

(a) หลัง gen=0

ลำดับ(i)	โครโมโซม
1	0100100110001110 [18830]
2	0100100010001010 [18570]
3	0100100010001110 [18574]
4	0100101010001110 [19086]

(b) หลัง gen=1

ตารางที่ 2.7 โครโมโซมชุดใหม่ หลังผ่านขบวนการทางพันธุศาสตร์ วัฏจักรแรก (gen=0) และวัฏจักรที่ 2 (gen=1)

6. ดำเนินการตั้งแต่ขั้นตอนที่ 2 ถึง 5 จนครบจำนวนวัฏจักรที่กำหนด (maxgen) หรือครบเงื่อนไขที่อื่น ๆ ที่กำหนดไว้ ของแต่ละปัญหา

สำหรับปัญหาที่ยกตัวอย่างนี้ เมื่อดำเนินการจนครบ 2 วัฏจักร (maxgen = 2) จะได้ค่าโครโมโซมที่ให้ค่าความเหมาะสมสูงสุดคือ 0100101010001110 [19086] ดังนั้นคำตอบที่ต้องการคือ $x=19086$ นั่นเอง และหากเปรียบเทียบค่าโครโมโซมแต่ละชุดของทั้ง 2 วัฏจักร จะเห็นว่าโครโมโซมในวัฏจักรหลังจะมีการวิวัฒนาการ ให้มีค่าที่ดีขึ้นเรื่อย ๆ ด้วย ข้อนี้เองที่เป็นจุดเด่นของอัลกอริทึมแบบเจเนติก แต่การที่จะให้ได้คำตอบที่ดีและเหมาะสมที่สุดนั้น สิ่งสำคัญอยู่ที่การกำหนดฟังก์ชันเป้าหมาย และฟังก์ชันความเหมาะสมของคำตอบ รวมทั้งจำนวนรอบหรือวัฏจักรที่ต้องการให้แก้ปัญหาหรือหาคำตอบ เป็นหลัก

2.5 ตัวอย่างการประยุกต์ใช้อัลกอริทึมเจเนติกในการแก้ไขปัญหา

หัวข้อนี้จะยกตัวอย่างการใช้อัลกอริทึมแบบเจเนติกในการแก้ไขปัญหา โดยปัญหาในตัวอย่างนี้เป็นการหาค่าคำตอบของสมการอย่างง่าย ๆ เพื่อจำลองให้เห็นภาพการทำงานของอัลกอริทึมแบบเจเนติกให้ชัดเจนยิ่งขึ้น ปัญหาที่กำหนดในตัวอย่างนี้ คือ หาค่า x ที่ทำให้สมการ $x^2 - 6 = 30$ เป็นจริง โดยกำหนดค่าและเงื่อนไขต่าง ๆ ดังนี้

- ฟังก์ชันเป้าหมาย คือ $f = x^2 - 6 = 30$

- ฟังก์ชันความเหมาะสม คือ $F = 1/(30 - (x^2 - 6) + 1) * 10000$

โดยที่ค่าที่เหมาะสมที่สุดคือ $\text{Max}(F)$

หมายเหตุ การทำให้ $F = 1/(30-(x^2 - 6)+1)*10000$ เนื่องจากค่าความเหมาะสมบางตัวที่สุ่มได้ จะมีทศนิยมจำนวนมาก เช่น 0.000082827 ซึ่งทำให้ดูยาก ดังนั้นจึงได้นำ 10000 มาคูณเข้าไปในฟังก์ชันเพื่อดูง่ายเมื่อแสดงบนตาราง และเมื่อนำค่าไปเปรียบเทียบกัน

- $pselect = F_i / \sum F$
- ค่าโครโมโซมที่ถูกเลือก $q_{i-1} < r < q_i$
- $E_i = pselect * popsize$
- q = ความถี่สะสม
- รูปแบบโครโมโซมเป็นโครโมโซมไบนารีขนาด 16 บิต (มีค่าระหว่าง 0-65536)
- จำนวนวิวัฒนาการ (maxgen) = 30 หรือ เมื่อ ได้ค่า x ที่ทำให้ฟังก์ชันเป้าหมายเป็นจริง
- จำนวนโครโมโซมใน 1 ชุด (Popsize) = 20
- อัตราความน่าจะเป็นของการครอสโอเวอร์ (Pc) = 0.08
- อัตราความน่าจะเป็นของการมิวเตชันในแต่ละบิต (Pm) = 0.05

ผลการทดสอบหาค่าคำตอบของสมการ $x^2 - 6 = 30$ แสดงได้ดังตารางที่ 2.8 ถึง 2.15 โดยแต่ละตารางจะแสดงวิวัฒนาการของโครโมโซมในแต่ละวิวัฒนาการ เริ่มตั้งแต่วิวัฒนาการที่ 1 ในตารางที่ 2.8 ทำการหาค่าต่าง ๆ ตามขบวนการเจเนติก ซึ่งได้แก่ การหาค่าความเหมาะสม (Fitness value), ค่าความน่าจะเป็นที่จะถูกเลือก (pselect), ค่าคาดหวังที่จะสุ่มได้ (E), ความถี่สะสม (q) จนได้ค่าโครโมโซมที่ถูกเลือกใน Mating pool จากนั้นนำโครโมโซมที่ได้ไปจะผ่านการครอสโอเวอร์ และมิวเตชัน หลังผ่านการครอสโอเวอร์ และมิวเตชัน จะได้โครโมโซมรุ่นต่อไป ซึ่งโครโมโซมรุ่นต่อไปที่ได้นี้ จะนำไปใช้เป็นโครโมโซมต้นแบบในวิวัฒนาการต่อไป คือ วิวัฒนาการที่ 2 (ตารางที่ 2.9) ทำเช่นนี้ไปเรื่อย ๆ จนกว่าจะครบจำนวนวิวัฒนาการที่กำหนด (maxgen) หรือพบคำตอบตามฟังก์ชันเป้าหมาย ซึ่งตัวอย่างการทดสอบนี้ได้หาค่าตอบพบในวิวัฒนาการที่ 8 ในการทดสอบจะหาค่าตอบได้ในวิวัฒนาการที่เท่าใดนั้น ขึ้นอยู่กับโครโมโซมที่สุ่มได้ ค่า Pc, Pm และปัจจัยอื่น ๆ ประกอบกัน ในบางกรณีอาจจะไม่สามารถหาค่าตอบที่ถูกต้องได้เลยภายในจำนวนวิวัฒนาการที่กำหนด (maxgen)

ตารางที่ 2.8 การทำงานเพื่อหาค่า x ที่ทำให้สมการ $x^2 - 6 = 30$ เป็นจริง ของอัลกอริทึมแบบเจเนติก ในวัฏจักรที่ 1

ลำดับ	โครโมโซมต้นแบบ	x	ค่าความเหมาะสม (Fitness value)	ค่าความน่าจะเป็นที่จะถูกเลือก (pselect)	คาดหวังที่จะสุ่มได้ (E)	ความถี่	เลขสุ่ม (r)	โครโมโซมที่ถูกเลือก	โครโมโซมหลังครอสโอเวอร์	โครโมโซมใหม่	โครโมโซมใหม่	
1	000000000110011 [51]	51	3.89712	0.97904	19.58076	0.97904	0.03363	1 [51]	[51]	[51]	[51]	โครโมโซมรุ่นต่อไป
2	0100011000011001 [17945]	17945	0.00003	0.00001	0.00016	0.97905	0.86303	1 [51]	[51]	[51]	[51]	[51]
3	0110101000010111 [27159]	27159	0.00001	0	0.00007	0.97905	0.49376	1 [51]	[51]	[51]	[51]	[51]
4	0000000110000010 [386]	386	0.06713	0.01686	0.3373	0.99591	0.6469	1 [51]	[51]	[1075]	[1075]	[1075]
5	010000111000001 [17345]	17345	0.00003	0.00001	0.00017	0.99592	0.34587	1 [51]	[51]	[49]	[49]	[49]
6	011010101110000 [27504]	27504	0.00001	0	0.00007	0.99593	0.39033	1 [51]	[51]	[51]	[51]	[51]
7	0101000101001111 [20815]	20815	0.00002	0.00001	0.00012	0.99593	0.48253	1 [51]	[51]	[51]	[51]	[51]
8	010100000110000 [20576]	20576	0.00002	0.00001	0.00012	0.99594	0.07956	1 [51]	[51]	[51]	[51]	[51]
9	0010101011010000 [10960]	10960	0.00008	0.00002	0.00042	0.99596	0.51665	1 [51]	[51]	[307]	[307]	[307]
10	0001011110000100 [6020]	6020	0.00028	0.00007	0.00139	0.99603	0.47346	1 [51]	[51]	[2067]	[2067]	[2067]
11	0001010010001101 [5261]	5261	0.00036	0.00009	0.00182	0.99612	0.44099	1 [51]	[51]	[115]	[115]	[115]
12	011011100110101 [28277]	28277	0.00001	0	0.00006	0.99612	0.59819	1 [51]	[51]	[8755]	[8755]	[8755]
13	010111001100010 [24162]	24162	0.00002	0	0.00009	0.99613	0.72622	1 [51]	[51]	[51]	[51]	[51]
14	0000001110100010 [930]	930	0.01156	0.0029	0.05809	0.99903	0.70678	1 [51]	[51]	[49]	[49]	[49]
15	001010101111010 [11002]	11002	0.00008	0.00002	0.00042	0.99905	0.68029	1 [51]	[51]	[1075]	[1075]	[1075]
16	0100110100011001 [19737]	19737	0.00003	0.00001	0.00013	0.99906	0.48643	1 [51]	[51]	[8243]	[8243]	[8243]
17	010001011111001 [17913]	17913	0.00003	0.00001	0.00016	0.99907	0.97153	1 [51]	[51]	[43]	[43]	[43]
18	0000011010010010 [1682]	1682	0.00353	0.00089	0.01776	0.99995	0.05512	1 [51]	[51]	[563]	[563]	[563]
19	0010100001001111 [10319]	10319	0.00009	0.00002	0.00047	0.99998	0.93912	1 [51]	[51]	[16443]	[16443]	[16443]
20	0010100001101010 [10474]	10474	0.00009	0.00002	0.00046	1	0.12839	1 [51]	[51]	[4147]	[4147]	[4147]
			ผลรวม (Sum of F)	3.98053								
			ค่าน้อยที่สุด (Min of F)	0.00001								
			ค่ามากที่สุด (Max of F)	3.89712								
			ค่าเฉลี่ย (Average of F)	0.199027								

ตารางที่ 2.10 การทำงานเพื่อหาค่า x ที่ทำให้สมการ $x^2 - 6 = 30$ เป็นจริง ของอัลกอริทึมแบบเจเนติก ในวัฏจักรที่ 3

ลำดับ (i)	โครโมโซมต้นแบบ	x	ค่าความเหมาะสม (Fitness value)	ค่าความน่าจะเป็นที่จะถูกเลือก (pselect)	ค่าความถี่ที่จะสุ่มได้ (E)	ความถี่สะสม (q)	เลขสุ่ม (r)	โครโมโซมที่ถูกเลือก	ค่าโครโมโซมหลังครอสโอเวอร์	ค่าโครโมโซมหลังมิวเทชัน	โครโมโซมรุ่นต่อไป
1	0000000000101011 [43]	43	5.51268	0.10346	2.06914	0.10346	0.82592	16 [41]	[41]	[8201]	0010000000010001 [8201]
2	0000000000110011 [51]	51	3.89712	0.07314	1.46275	0.17659	0.167	2 [51]	[51]	[51]	0000000000110011 [51]
3	0000000000110111 [55]	55	3.34448	0.06277	1.25532	0.23936	0.29719	4 [51]	[51]	[51]	0000000000110011 [51]
4	0000000000110011 [51]	51	3.89712	0.07314	1.46275	0.3125	0.81683	16 [41]	[41]	[4137]	0001000000101001 [4137]
5	0000000000110001 [49]	49	4.22654	0.07932	1.5864	0.39182	0.91562	16 [41]	[41]	[41]	0000000000101001 [41]
6	0000000000110011 [51]	51	3.89712	0.07314	1.46275	0.46496	0.98172	19 [51]	[51]	[51]	0000000000110011 [51]
7	0010000000110010 [8242]	8242	0.00015	0	0.00006	0.46496	0.18958	3 [55]	[55]	[55]	0000000000110111 [55]
8	0000000000110011 [51]	51	3.89712	0.07314	1.46275	0.5381	0.36637	5 [49]	[49]	[49]	0000000000110001 [49]
9	0000000000111110 [62]	62	2.62536	0.04927	0.98541	0.58737	0.87744	16 [41]	[41]	[41]	0000000000101001 [41]
10	0010000000110011 [8243]	8243	0.00015	0	0.00006	0.58737	0.39149	5 [49]	[49]	[49]	0000000000110001 [49]
11	0000000000110011 [51]	51	3.89712	0.07314	1.46275	0.66051	0.69359	14 [51]	[51]	[16499]	010000000110011 [16499]
12	0000100000010011 [2067]	2067	0.00234	0.00004	0.00088	0.66055	0.79687	15 [51]	[51]	[51]	0000000000110011 [51]
13	0000010001111011 [1147]	1147	0.0076	0.00014	0.00285	0.66069	0.331	5 [49]	[49]	[49]	0000000000110001 [49]
14	0000000000110011 [51]	51	3.89712	0.07314	1.46275	0.73383	0.07892	1 [43]	[43]	[32811]	1000000000101011 [32811]
15	0000000000110011 [51]	51	3.89712	0.07314	1.46275	0.80697	0.63707	11 [51]	[51]	[51]	0000000000110011 [51]
16	0000000000101001 [41]	41	6.07533	0.11402	2.28033	0.92099	0.49794	8 [51]	[51]	[59]	0000000000110111 [59]
17	0000000010110011 [179]	179	0.31244	0.00586	0.11727	0.92685	0.9866	19 [51]	[51]	[51]	0000000000110011 [51]
18	0001000000110011 [4147]	4147	0.00058	0.00001	0.00022	0.92686	0.77523	15 [51]	[51]	[59]	0000000000110111 [59]
19	0000000000110011 [51]	51	3.89712	0.07314	1.46275	1	0.74084	15 [51]	[51]	[355]	0000000101100011 [355]
20	0010000010010011 [8339]	8339	0.00014	0	0.00005	1	0.06369	1 [43]	[43]	[43]	0000000000101011 [43]
ผลรวม (Sum of F) 53.28475											
ค่าน้อยที่สุด (Min of F) 0.00014											
ค่ามากที่สุด (Max of F) 6.07533											
ค่าเฉลี่ย (Average of F) 2.664238											

ตารางที่ 2.11 การทำงานเพื่อหาค่า x ที่ทำให้สมการ $x^2 - 6x + 30$ เป็นจริง ของอัลกอริทึมแบบจenetik ในวัฏจักรที่ 4

ลำดับ	โครโมโซมต้นแบบ	x	ค่าความเหมาะสม (Fitness value)	ค่าความน่าจะเป็นที่จะถูกเลือก (pselect)	ค่าคาดหวังที่จะสุ่มได้ (E)	ความถี่สะสม (q)	เลขสุ่ม (r)	ค่าลำดับโครโมโซมที่ถูกเลือก	ค่าโครโมโซมหลังการเลือก	ค่าโครโมโซมหลังการเลือก	ค่าโครโมโซมหลังการเลือก	โครโมโซมรุ่นต่อไป	
1	0010000000010001	8201	0.00015	0	0.00005	0	0.24531	6	[51]	[51]	[51]	000000000110011	[51]
2	0000000000110011	51	3.89712	0.0619	1.23808	0.06191	0.00861	2	[51]	[51]	[35]	000000000100011	[35]
3	0000000000110011	51	3.89712	0.0619	1.23808	0.12381	0.53917	10	[49]	[49]	[113]	000000000110001	[113]
4	0001000000101001	4137	0.00058	0.00001	0.00019	0.12382	0.59114	12	[51]	[51]	[51]	0000000000110011	[51]
5	0000000000101001	41	6.07533	0.0965	1.93008	0.22032	0.68941	13	[49]	[49]	[115]	000000000110011	[115]
6	0000000000110011	51	3.89712	0.0619	1.23808	0.28223	0.57686	12	[51]	[51]	[59]	000000000110111	[59]
7	0000000000110111	55	3.34448	0.05313	1.06251	0.33535	0.0333	2	[51]	[51]	[51]	000000000110011	[51]
8	0000000000110001	49	4.22654	0.06714	1.34274	0.40249	0.81097	17	[51]	[51]	[291]	000000100100011	[291]
9	0000000000101001	41	6.07533	0.0965	1.93008	0.49899	0.02686	2	[51]	[51]	[51]	000000000110011	[51]
10	0000000000110001	49	4.22654	0.06714	1.34274	0.56613	0.26337	6	[51]	[51]	[51]	000000000110011	[51]
11	010000000110011	16499	0.00004	0	0.00001	0.56613	0.8088	17	[51]	[51]	[32819]	100000000110011	[32819]
12	0000000000110011	51	3.89712	0.0619	1.23808	0.62804	0.18458	5	[41]	[41]	[41]	000000000101001	[41]
13	0000000000110001	49	4.22654	0.06714	1.34274	0.69517	0.19053	5	[41]	[41]	[41]	000000000101001	[41]
14	1000000000101011	32811	0.00001	0	0	0.69517	0.76424	16	[59]	[59]	[2235]	000010001011011	[2235]
15	0000000000110011	51	3.89712	0.0619	1.23808	0.75708	0.79345	16	[59]	[59]	[2107]	000010000011011	[2107]
16	0000000000110111	59	2.90192	0.0461	0.92191	0.80317	0.38017	8	[49]	[49]	[49]	000000000110001	[49]
17	0000000000110011	51	3.89712	0.0619	1.23808	0.86508	0.32121	7	[55]	[55]	[567]	0000001000110111	[567]
18	0000000000110111	59	2.90192	0.0461	0.92191	0.91117	0.72201	15	[51]	[51]	[51]	000000000110011	[51]
19	000000010100011	355	0.07937	0.00126	0.02522	0.91243	0.36488	8	[49]	[49]	[49]	000000000110001	[49]
20	0000000000101011	43	5.51268	0.08757	1.75133	1	0.84939	17	[51]	[51]	[51]	000000000110011	[51]
			ผลรวม (Sum of F)	62.95415									
			ค่าน้อยที่สุด (Min of F)	0.00001									
			ค่ามากที่สุด (Max of F)	6.07533									
			ค่าเฉลี่ย (Average of F)	3.147708									

ตารางที่ 2.12 การทำงานเพื่อหาค่า x ที่ทำให้สมการ $x^2 - 6x + 30$ เป็นจริง ของอัลกอริทึมแบบเจเนติก ในวัฏจักรที่ 5

ลำดับ	โครโมโซมต้นแบบ	x	ค่าความเหมาะสม (Fitness value)	ค่าความน่าจะเป็นที่จะถูกเลือก (pselect)	ภาคหวังที่จะสุ่มได้ (E)	ความถี่สะสม (q)	เลขสุ่ม (r)	ค่าลำดับโครโมโซมที่ถูกเลือก	ค่าโครโมโซมหลังครอสโอเวอร์	ค่าโครโมโซมหลังมิวเตชัน	โครโมโซมรุ่นต่อไป	
1	0000000000110011	51	3.89712	0.06401	1.28014	0.06401	0.99777	20	[51]	[3]	0000000000000011	[3]
2	0000000000100011	35	8.40336	0.13802	2.76037	0.20203	0.571	12	[41]	[41]	0000000000101001	[41]
3	0000000000110001	113	0.7853	0.0129	0.25796	0.21492	0.58434	12	[41]	[41]	0000000000101001	[41]
4	0000000000110011	51	3.89712	0.06401	1.28014	0.27893	0.68459	13	[41]	[4137]	0001000000101001	[4137]
5	0000000000110011	115	0.75815	0.01245	0.24904	0.29138	0.44499	9	[51]	[51]	0000000000110011	[51]
6	0000000000110111	59	2.90192	0.04766	0.95323	0.33904	0.03446	1	[51]	[59]	0000000000110111	[59]
7	0000000000110011	51	3.89712	0.06401	1.28014	0.40305	0.99149	20	[51]	[8243]	0010000000110011	[8243]
8	0000000010010001	291	0.11814	0.00194	0.03881	0.40499	0.43178	9	[51]	[51]	0000000000110011	[51]
9	0000000000110011	51	3.89712	0.06401	1.28014	0.469	0.21357	3	[113]	[96]	0000000001100000	[96]
10	0000000000110011	51	3.89712	0.06401	1.28014	0.53301	0.95035	20	[51]	[51]	0000000000110011	[51]
11	1000000000110011	32819	0.00001	0	0	0.53301	0.82315	18	[51]	[51]	0000000000110011	[51]
12	0000000000101001	41	6.07533	0.09978	1.99565	0.63279	0.46672	9	[51]	[51]	0000000000110011	[51]
13	0000000000101001	41	6.07533	0.09978	1.99565	0.73257	0.71612	13	[41]	[1065]	0000100000101001	[1065]
14	0000100010110111	2235	0.002	0.00003	0.00066	0.7326	0.02869	1	[51]	[179]	0000000010110011	[179]
15	0000100000111011	2107	0.00225	0.00004	0.00074	0.73264	0.60564	12	[41]	[41]	0000000000101001	[41]
16	0000000000110001	49	4.22654	0.06942	1.38835	0.80206	0.74032	16	[49]	[49]	0000000000110001	[49]
17	0000001000110111	567	0.03111	0.00051	0.01022	0.80257	0.01077	1	[51]	[4151]	0001000000110111	[4151]
18	0000000000110011	51	3.89712	0.06401	1.28014	0.86658	0.86294	18	[51]	[2099]	0000100000110011	[2099]
19	0000000000110001	49	4.22654	0.06942	1.38835	0.93599	0.75469	16	[49]	[20497]	0101000000100001	[20497]
20	0000000000110011	51	3.89712	0.06401	1.28014	1	0.91787	19	[49]	[2353]	0000100100110001	[2353]
ผลรวม (Sum of F) 60.88582												
ค่าน้อยที่สุด (Min of F) 0.00001												
ค่ามากที่สุด (Max of F) 8.40336												
ค่าเฉลี่ย (Average of F) 3.044291												

ตารางที่ 2.15 การทำงานเพื่อหาค่า x ที่ทำให้สมการ $x^2 - 6 = 30$ เป็นจริง ของอัลกอริทึมแบบเจเนติก ในวัฏจักรที่ 8

ลำดับ	โครโมโซมต้นแบบ	x	ค่าความเหมาะสม (Fitness value)	ค่าความน่าจะเป็นที่จะถูกเลือก (pselect)	คาดหวังที่จะสุ่มได้ (E)	ความถี่	ค่าเฉลี่ย (p)	โครโมโซมที่ถูกเลือก	โครโมโซมหลังครอสโอเวอร์	ค่าโครโมโซมหลัง	โครโมโซมใหม่
1	0000000000000110 [6]	6	10000	0.82827	16.56541	0.82827	0.28663	1 [6]	[6]	[17926]	0100011000000110 [17926]
2	0000000100000010 [258]	258	0.15031	0.00001	0.00025	0.82828	0.77166	1 [6]	[6]	[38]	000000000100110 [38]
3	1000000000000111 [32775]	32775	0.00001	0	0	0.82828	0.10474	1 [6]	[6]	[14]	0000000000001110 [14]
4	0000010000000111 [1031]	1031	0.00941	0	0.00002	0.82828	0.78347	1 [6]	[6]	[6]	0000000000000110 [6]
5	0000000000100011 [35]	35	8.40336	0.0007	0.01392	0.82898	0.08594	1 [6]	[6]	[37014]	1001000010010110 [37014]
6	0000000000000011 [3]	3	357.1429	0.02958	0.59162	0.85856	0.54494	1 [6]	[6]	[262]	0000001000000110 [262]
7	1100100000000111 [51207]	51207	0	0	0	0.85856	0.42155	1 [6]	[6]	[6]	0000000000000110 [6]
8	0100000000001011 [16395]	16395	0.00004	0	0	0.85856	0.47206	1 [6]	[6]	[134]	000000010000110 [134]
9	0000000000000011 [3]	3	357.1429	0.02958	0.59162	0.88814	0.45674	1 [6]	[6]	[6]	0000000000000110 [6]
10	0000000000000001 [1]	1	277.7778	0.02301	0.46015	0.91115	0.06238	1 [6]	[6]	[2054]	0000100000000110 [2054]
11	0010000000000011 [8195]	8195	0.00015	0	0	0.91115	0.16138	1 [6]	[6]	[6]	0000000000000110 [6]
12	0100001000000011 [16899]	16899	0.00004	0	0	0.91115	0.83071	6 [3]	[3]	[1]	0000000000000001 [1]
13	1000001000000111 [33287]	33287	0.00001	0	0	0.91115	0.28083	1 [6]	[6]	[6]	0000000000000110 [6]
14	0001100000000001 [6145]	6145	0.00026	0	0	0.91115	0.9856	18 [7]	[7]	[23]	00000000010111 [23]
15	0000000010000001 [129]	129	0.60219	0.00005	0.001	0.9112	0.91852	16 [3]	[3]	[71]	000000001000111 [71]
16	0000000000000011 [3]	3	357.1429	0.02958	0.59162	0.94078	0.84106	6 [3]	[3]	[67]	000000001000011 [67]
17	0000000010000111 [135]	135	0.54975	0.00005	0.00091	0.94083	0.82095	1 [6]	[6]	[6]	0000000000000110 [6]
18	0000000000000111 [7]	7	714.2857	0.05916	1.18324	0.99999	0.40376	1 [6]	[6]	[6]	0000000000000110 [6]
19	0000000100000111 [263]	263	0.14465	0.00001	0.00024	1	0.2924	1 [6]	[6]	[4366]	0001000100001110 [4366]
20	0100000000000011 [16387]	16387	0.00004	0	0	1	0.60137	1 [6]	[6]	[7]	0000000000000111 [7]
			ผลรวม (Sum of F)	12073.35							
			ค่าน้อยที่สุด (Min of F)	0							
			ค่ามากที่สุด (Max of F)	10000							
			ค่าเฉลี่ย (Average of F)	603.6676							

จากตารางที่ 2.8 ถึง 2.15 แสดงการทำงานของอัลกอริทึมแบบเจเนติก โดยแสดงวิวัฒนาการของโครโมโซมในแต่ละวัฏจักร ซึ่งตัวอย่างนี้สามารถหาค่าของ $x = 6$ ได้ในวัฏจักรที่ 8 จากการติดตามคู่วิวัฒนาการของโครโมโซมแต่ละรุ่น พบว่าในวัฏจักรหลัง ๆ จะเริ่มมีโครโมโซมที่ให้ค่าความเหมาะสมเพิ่มสูงขึ้นเรื่อย ๆ แสดงให้เห็นความสามารถในการหาคำตอบ ของอัลกอริทึมแบบเจเนติก

2.6 สรุป

อัลกอริทึมแบบเจเนติก เป็นอัลกอริทึมที่เลียนแบบการทำงานของธรรมชาติ โดยอาศัยหลักการสุ่มซึ่งเป็นการสุ่มคำตอบจากกลุ่มคำตอบรุ่นถัดไป ส่วนการจะทำให้ อัลกอริทึมแบบเจเนติก นั้นหาคำตอบได้เร็วขึ้น ต้องอาศัยการกำหนดฟังก์ชันเป้าหมาย และฟังก์ชันความเหมาะสมให้สอดคล้องกับคำตอบที่ต้องการหา นอกจากนี้ยังมีวิธีการเพิ่มประสิทธิภาพของอัลกอริทึมแบบเจเนติก โดยวิธีที่กล่าวมา เช่น รีโพรดักชัน ครอสโอเวอร์หลายจุด ไบนารีมิวเตชันแบบกำหนดค่าบิต และอินเวอร์ชัน ซึ่งผู้ใช้งาน อัลกอริทึมแบบเจเนติก ควรพิจารณานำมาใช้ให้เหมาะสม เพื่อเพิ่มประสิทธิภาพในการหาคำตอบของ อัลกอริทึมแบบเจเนติก

การประยุกต์ใช้ อัลกอริทึมแบบเจเนติก ในการแก้ปัญหาต่าง ๆ โดยการพัฒนาอัลกอริทึมแบบเจเนติกนั้น จะประกอบด้วย การกำหนดรูปแบบโครโมโซมของปัญหาให้กับ อัลกอริทึมแบบเจเนติก โดยการพิจารณาจากสภาพของปัญหาหรือตัวแปรต่าง ๆ ที่ทำเป็นองค์ประกอบของปัญหา และกำหนดฟังก์ชันเป้าหมายให้กับอัลกอริทึมแบบเจเนติก เพื่อใช้ในการประมวลผลให้ได้แนวทางการหาคำตอบที่ดีขึ้น รวมถึงตัวดำเนินการทางพันธุศาสตร์ที่เหมาะสมกับกระบวนการต่าง ๆ ของอัลกอริทึมแบบเจเนติก การกำหนดรูปแบบของโครโมโซม ฟังก์ชันเป้าหมาย หรือตัวดำเนินการทางพันธุศาสตร์ ของแต่ละปัญหาอาจจะกำหนดในรูปแบบอื่น ๆ ที่แตกต่างกันออกไปได้ อย่างไรก็ตามการปรับปรุงอัลกอริทึมแบบเจเนติก เพื่อหาคำตอบของปัญหาต่าง ๆ ให้มีประสิทธิภาพดีขึ้นนั้น ควรเลือกใช้เทคนิคต่าง ๆ ให้เหมาะสมด้วย

บทที่ 3

การประมวลผลสัญญาณดิจิทัลด้วยกระบวนการสุ่มแบบ ไม่ต่อเนื่องทางเวลา

ในบทนี้จะกล่าวถึงหลักพื้นฐานเกี่ยวกับการประมวลผลสัญญาณดิจิทัลด้วย กระบวนการสุ่มแบบไม่ต่อเนื่องทางเวลา (Discrete-Time Random Processes) ซึ่งมีความสำคัญมากในการประมวลผลสัญญาณดิจิทัล เนื่องจากการประมวลผลสัญญาณแบบอะแดปทีฟ จะเกี่ยวข้องกับสัญญาณสุ่มแบบต่าง ๆ ทั้งนี้การวิเคราะห์คุณสมบัติของสัญญาณแบบสุ่มนั้น จะต้องอาศัยหลักทางสถิติ หลาย ๆ แบบเช่น ค่าเฉลี่ย ค่ากลาง (Mean) ค่าความแปรปรวน (Variance) ค่าความเกี่ยวพัน (Correlation) เป็นต้น และบทนี้ยังกล่าวถึง การแทนสัญญาณที่มีพลังงานไม่จำกัดด้วยสเปกตรัม มีความสำคัญมากต่อการอธิบายความสัมพันธ์ระหว่างสัญญาณอินพุต และสัญญาณเอาต์พุตของระบบเชิงเส้นไม่แปรตามเวลา (LTI) เมื่อสัญญาณที่มีพลังงานไม่จำกัด

3.1 ประเภทของสัญญาณ

สัญญาณ โดยทั่วไปแบ่งออกเป็น 3 ประเภท คือ

3.1.1 สัญญาณแบบ **Deterministic** เป็นสัญญาณที่สามารถอธิบายได้ด้วยสมการทางคณิตศาสตร์ ซึ่งแบ่งเป็นสัญญาณรายคาบ และสัญญาณไม่รายคาบ เช่น

$$x(t) = \sin(2\pi ft + \phi) \quad (3.1)$$

หรือ

$$x(n) = \sin(2\pi fn + \phi) \quad (3.2)$$

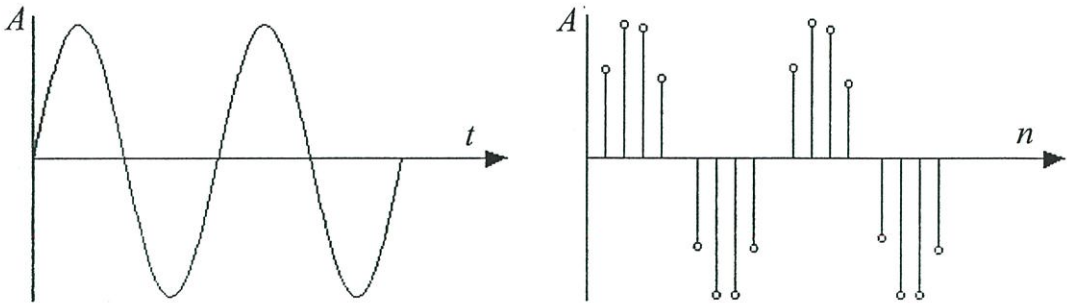
สมการที่ (3.1) คือ สัญญาณไซน์ต่อเนื่องทางเวลา ส่วนสมการที่ (3.2) เป็นสัญญาณไม่ต่อเนื่องทางเวลา ดังแสดงในรูปที่ 3.1 ส่วนสัญญาณแบบ Deterministic แบบไม่มีคาบ เช่น

$$x(t) = \text{rect}(t/T) \quad (3.3)$$

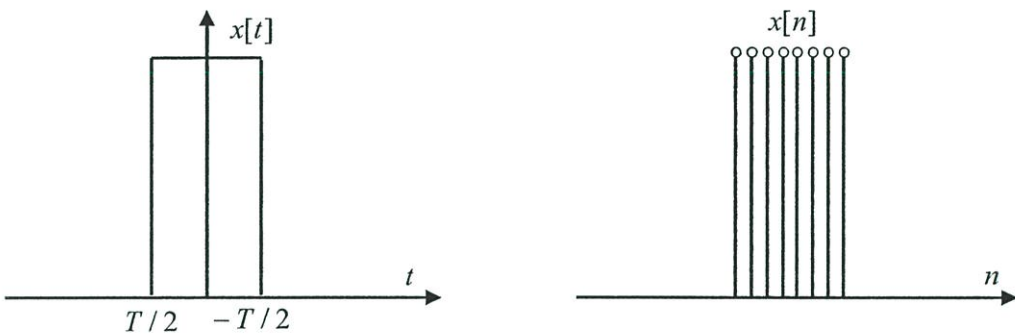
หรือ

$$x(n) = \text{rect}(n/T) \quad (3.4)$$

ดังแสดงในรูปที่ 3.2



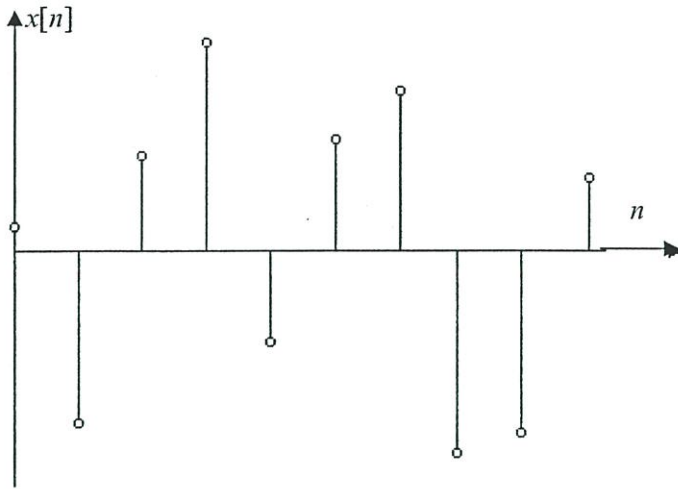
รูปที่ 3.1 สัญญาณ Deterministic แบบมีคาบ



รูปที่ 3.2 สัญญาณแบบ Deterministic แบบไม่มีคาบ

3.1.2 สัญญาณแบบสุ่ม (Random หรือ Stochastic) เป็นสัญญาณที่ไม่สามารถคาดเดาล่วงหน้าได้ว่าจะมีลักษณะเช่นใด เช่น สัญญาณรบกวนที่เกิดจากการควอนไทซ์สัญญาณของวงจร A/D ดังแสดงในรูปที่ 3.3

3.1.3 สัญญาณแบบ Chaotic สัญญาณประเภทนี้จะคล้ายคลึงกับสัญญาณแบบสุ่ม แต่จะมีรูปแบบเฉพาะตัว เช่น การหมุนของน้ำทำให้เกิดสิ่งที่เรียกว่า Swirl ซึ่งสามารถหารูปแบบได้จากสิ่งที่เรียกว่า Attractor หรือตัวที่ทำให้เกิดปรากฏการณ์ Chaos นั่นเอง ซึ่งสัญญาณในลักษณะนี้ยังอยู่ในขั้นทำการวิจัย

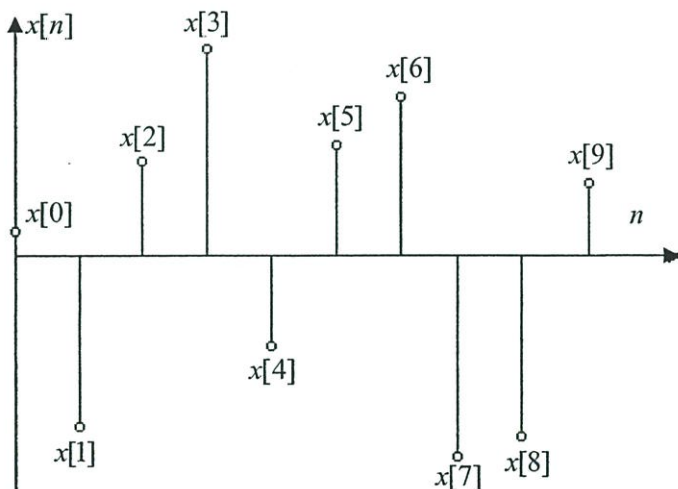


รูปที่ 3.3 สัญญาณแบบสุ่ม

3.2 ตัวแปรสุ่ม (Random Variable) และการวิเคราะห์ตัวแปรสุ่ม

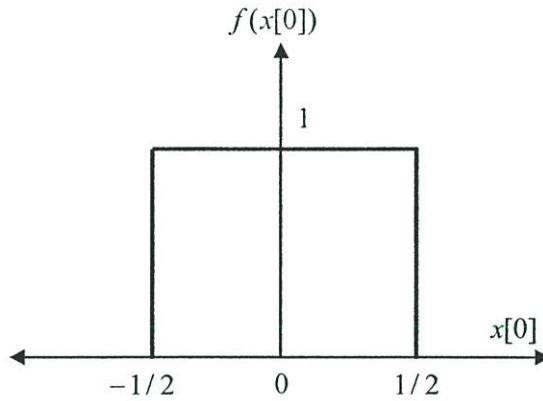
ในหัวข้อนี้จะกล่าวถึง แนวความคิดของตัวแปรสุ่ม และคุณลักษณะเฉพาะตัวต่าง ๆ ซึ่งค่อนข้างมีความสำคัญ เนื่องจากตัวแปรสุ่มมีค่าการกระจายที่กระจายไม่แน่นอน และพบบ่อยในทางปฏิบัติซึ่ง การศึกษาเรื่องตัวแปรสุ่มนี้ เป็นจุดเริ่มต้นสำหรับผู้ที่ศึกษาในเรื่องกระบวนการสุ่มในขั้นสูงต่อไป

สมมติว่า $x[n]$ คือ สัญญาณแบบสุ่มที่ไม่ต่อเนื่องทางเวลา ดังแสดงในรูปที่ 3.4 โดย $x[n]$ แต่ละตัวจะเรียกว่าตัวแปรสุ่ม (Random variable) หรือ ใช้ตัวย่อว่า $r.v.$



รูปที่ 3.4 สัญญาณแบบสุ่มที่ไม่ต่อเนื่องทางเวลา

$x[n]$ หรือ $r.v.$ แต่ละตัวเรียกว่า ตัวแปรสุ่ม สิ่งที่เป็นตัวกำหนดค่าตัวแปรสุ่มแต่ละตัวจะขึ้นกับค่าความเป็นไปได้ ซึ่งอยู่ในรูปของฟังก์ชัน คือ Probability Density Function (Pdf) เช่น $x[0]$ จะมี Pdf ส่วนตัวเป็น $f(x[0])$ หากกำหนดให้ $x[0]$ มีค่าไม่ต่ำกว่า $-1/2$ และมากที่สุดไม่เกิน $1/2$ และทราบว่าค่าความเป็นไปได้ของฟังก์ชันมีค่าน้อยกว่าหรือเท่ากับ 1 เสมอ สิ่งที่ได้ คือ กราฟของ Pdf ของ $x[0]$ หรือ $f(x[0])$ ดังรูปที่ 3.5



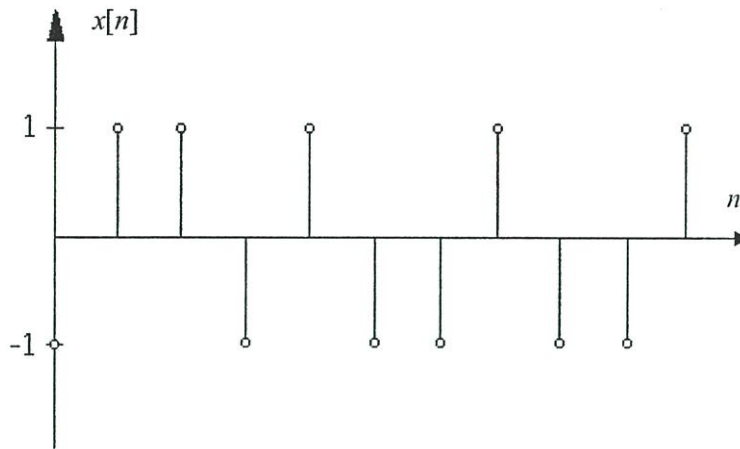
รูปที่ 3.5 Pdf ของตัวแปรสุ่ม $x[0]$

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

3.2.1 คุณสมบัติของ Pdf จากการอินทิเกรตรูปกราฟในรูปที่ 3.5 จะได้พื้นที่รูปกราฟนั้น ซึ่งเขียนเป็นสมการได้ว่า

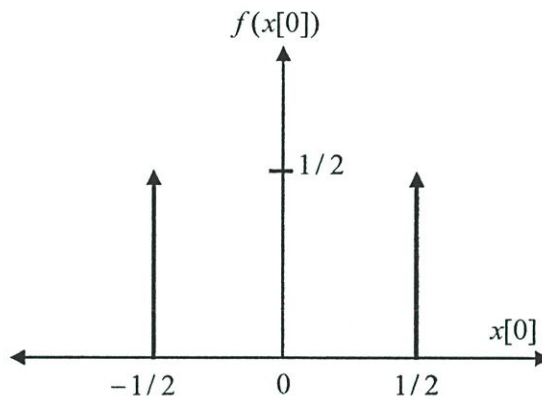
$$\int_{-\infty}^{\infty} f(x)[0] dx[0] = 1 \quad (3.5)$$

ในกรณีที่สัญญาณ $x[n]$ เป็นสัญญาณแบบไบนารี ซึ่งมีค่าเป็น 1 และ -1 ดังแสดงในรูปที่ 3.6



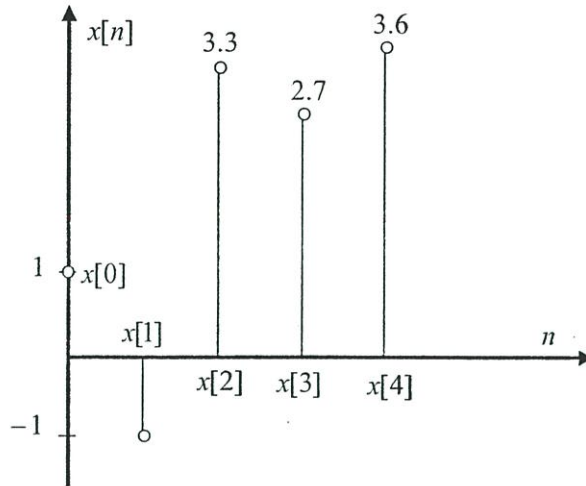
รูปที่ 3.6 สัญญาณแบบสุ่มที่มีค่า 1 และ -1

จะได้ Pdf ของ $x[0]$ เป็นดังรูปที่ 3.7



รูปที่ 3.7 Pdf ของสัญญาณแบบไบนารี

ตัวอย่างเช่น ถ้ามีสัญญาณ $x[n]$ ดังรูปที่ 3.8



รูปที่ 3.8 ตัวอย่างสัญญาณแบบสุ่ม

จากรูปที่ 3.8 $x[0]$ และ $x[1]$ เป็นสัญญาณแบบสุ่มไบนารี ซึ่งจะให้ Pdf ดังรูปที่ 3.7 ส่วน $x[2], x[3], x[4]$ มี Pdf ที่เรียกว่า Normal Distribution หรือ Gaussian Distribution สัญลักษณ์ที่ใช้แทนคือ

$$N\left[\frac{3.3}{\text{Mean}}, \frac{1}{\text{StandardDeviation}}\right]$$

โดย Gaussian Distribution มีสมการเป็น

$$f(x[2]) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(x[2]-\mu)^2}{2\sigma^2}\right) \quad (3.6)$$

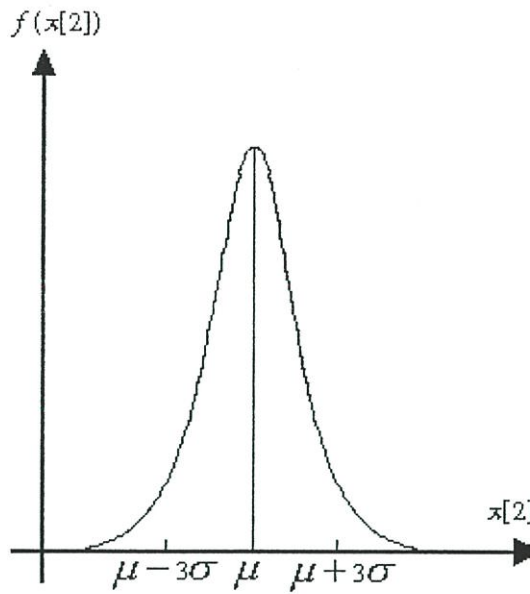
โดยที่

$$\mu = \text{Mean}$$

$$\sigma = \text{Standard Deviation}$$

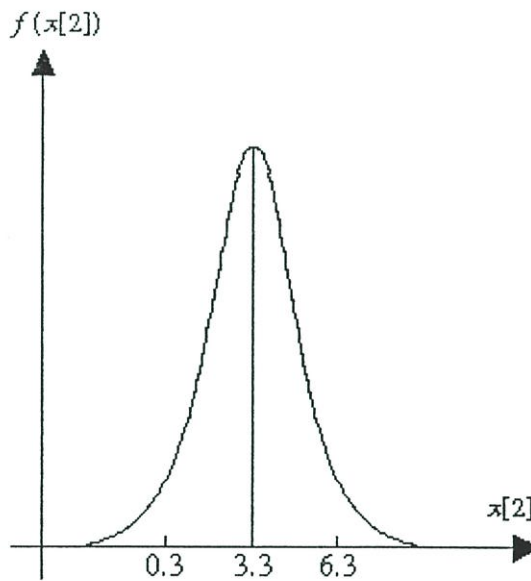
$$\sigma^2 = \text{Variance}$$

การกระจายของ $x[2]$ มีรูปร่างดังรูปที่ 3.9



รูปที่ 3.9 Gaussian Distribution ของ $x[2]$

จากรูปที่ 3.9 พื้นที่ของกราฟ 99% จะอยู่ในบริเวณ $\mu - 3\sigma$ ถึง $\mu + 3\sigma$ ดังนั้น ถ้า $N(3.3,1)$ จะได้ดังรูปที่ 3.10



รูปที่ 3.10 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 $F_x(x[n])$ มีค่าดังสมการ

$$F_x(x[n]) = \text{Probability}(X[n] \leq x[n]) \quad (3.7)$$

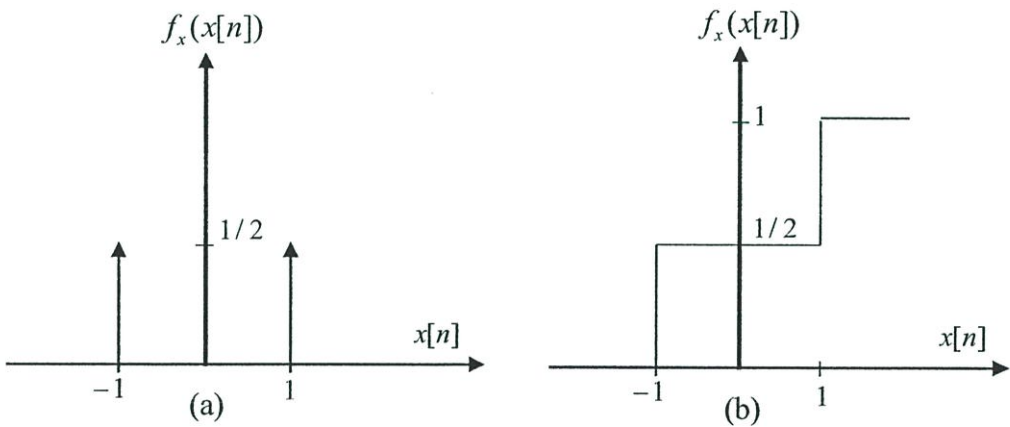
โดยที่ $x[n]$ คือ r.v. และ $x[n]$ คือ ค่าหนึ่งๆ จากสมการ (3.7) จะได้ว่า

$$F_x(x[n]) = \int_{-\infty}^{x[n]} f_x(z) dz \quad (3.8)$$

หรือ

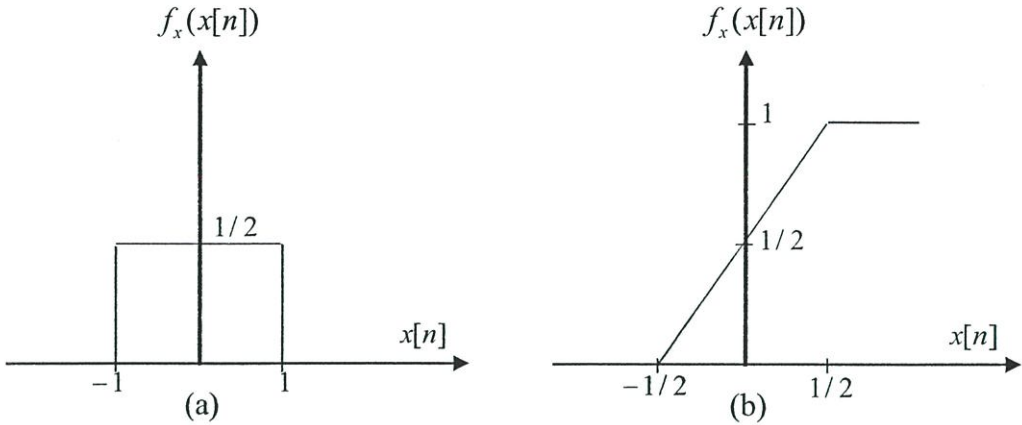
$$f_x(x[n]) = \frac{\partial F_x(x[n])}{\partial x[n]} \quad (3.9)$$

หากมี $f_x(x[n])$ ดังรูปที่ 3.11 (a) จะได้ $F_x(x[n])$ เป็นดังรูปที่ 3.11 (b) ตามลำดับ และผลสุดท้ายจะมีค่าเป็น 1 เสมอ



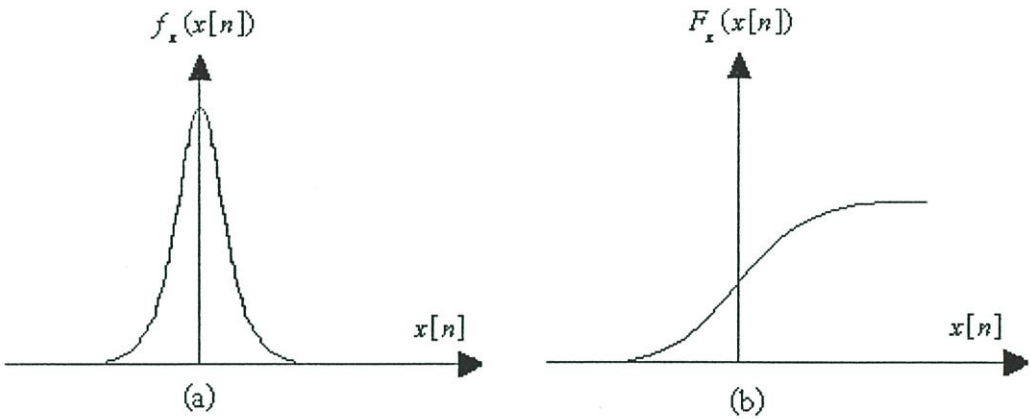
รูปที่ 3.11 (a) Pdf และ (b) Cdf ของตัวแปรสุ่ม $x[n]$

หาก $f_x(x[n])$ เป็นดังรูปที่ 3.12 (a) จะได้ $F_x(x[n])$ ดังรูปที่ 3.12(b) ตามลำดับ



รูปที่ 3.12 (a) Pdf และ (b) Cdf ของ $x[n]$

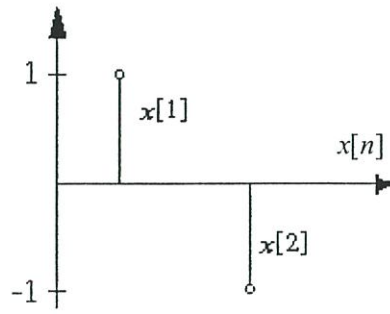
สำหรับ Gaussian Distribution จะได้ดังรูปที่ 3.13



รูปที่ 3.13 (a) Pdf และ (b) Cdf ของตัวแปรสุ่มที่เป็นแบบ Gaussian Distribution

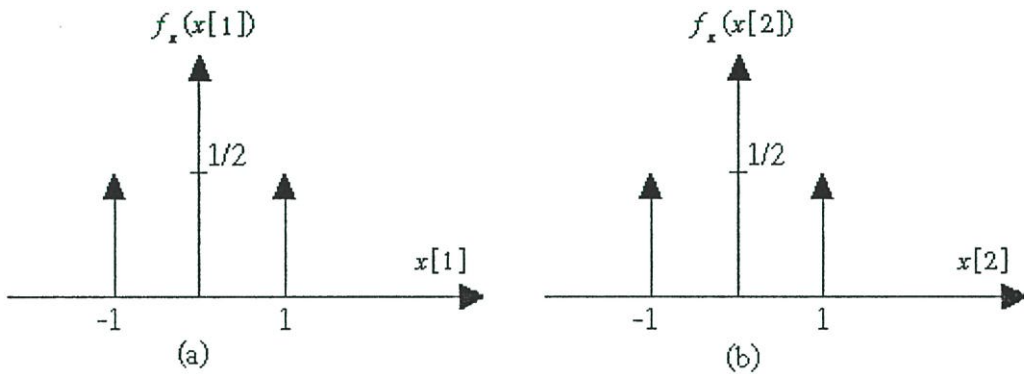
โดยทั่วไปค่า Pdf ของ $x[n_1]$ และ $x[n_2]$ คาดหวังไม่ได้ว่า ค่าทั้งสองจะเหมือนกัน ดังนั้นจึงมีวิธีการหาความสัมพันธ์ของ $x[n_1]$ และ $x[n_2]$ ในรูปของ Pdf ร่วม (Joint Pdf)

ถ้าให้ $x[n_1]$ และ $x[n_2]$ เป็นสัญญาณสุ่มแบบไบนารีแสดงดังรูปที่ 3.14



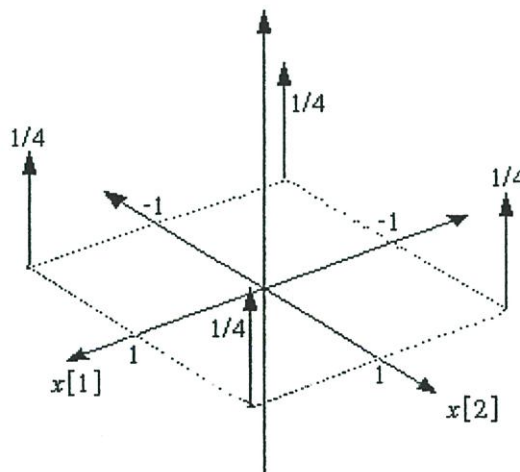
รูปที่ 3.14 สัญญาณแบบสุ่มไบนารี

จะได้ Pdf สำหรับ $x[n_1]$ และ $x[n_2]$ ดังรูปที่ 3.15 (a) และ 3.15 (b) ตามลำดับ



รูปที่ 3.15 (a) Pdf ของ $x[n_1]$ และ (b) Pdf ของ $x[n_2]$

และจะได้ Joint Pdf เป็น $f_x(x[1], x[2])$ แสดงดังรูปที่ 3.16



รูปที่ 3.16 Joint Pdf ของ $x[1]$ และ Joint Pdf ของ $x[2]$

สามารถเขียนความสัมพันธ์ในรูปสมการ Joint Cdf ดังนี้

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_x(x[1], x[2]) dx[1] dx[2] = 1 \quad (3.10)$$

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

และจะได้ Joint Pdf ดังสมการ

$$f(x[n_1], x[n_2]) = \frac{\partial^2 F(x[n_1], x[n_2])}{\partial x[n_1] \partial x[n_2]} \quad (3.11)$$

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

$$F(x[n_1], x[n_2], \dots, x[n_k]) = \text{Prob}(X[n_1] \leq x[n_1], \dots, X[n_k] \leq x[n_k]) \quad (3.12)$$

และได้ Pdf ดังสมการ

$$f(x[n_1], \dots, x[n_k]) = \frac{\partial^k F(x[n_1], \dots, x[n_k])}{\partial x[n_1] \dots \partial x[n_k]} \quad (3.13)$$

3.2.2 ค่าเฉลี่ยแบบ Ensemble ที่ผ่านมาได้ทำการพิจารณาสัญญาณแบบสุ่ม $x[n]$ และได้ Pdf ของ $x[n_1]$ นั้น ๆ หากมี $x[n_1]$ มากกว่า 1 จะต้องพิจารณา $x[n_1]$ ตั้งแต่ $x_1[n]$ ถึง $x_k[n]$ ที่เวลา $n = n_0$ เดียวกัน โดย ถ้าหากกำหนดให้ Probability ของ $x[n]$ ทุกตัวมีค่าเท่ากันจะได้ว่า

$$\text{Prob ของ } X_k[n] = \frac{1}{k}, \forall k \quad (3.14)$$

ค่าเฉลี่ยแบบ Ensemble เป็นการหาค่าเฉลี่ยของ $f(x[n_0])$ โดยที่ $x[n_0]$ เป็น $x_1[n_0], x_2[n_0], \dots, x_k[n_0]$ หรือ

$$\begin{aligned}
 \text{Ensemble average} &= x_1[n_0] \cdot \text{Prob}(x_k[n]) \\
 &+ x_2[n_0] \cdot \text{Prob}(x_2[n]) \\
 &+ \dots \\
 &+ x_k[n_0] \cdot \text{Prob}(x_k[n])
 \end{aligned} \tag{3.15}$$

3.2.3 ค่ากลาง (Mean value) มีข้อกำหนดในการหาค่าตัวแปรสุ่มหลายตัวดังนี้

$$\mu[n] = E\{x[n]\} \tag{3.16}$$

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

$$E\{x[n]\} = \int_{-\infty}^{\infty} x[n] f(x[n]) dx[n] \tag{3.17}$$

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

$$\mu[n] = \lim_{N \rightarrow \infty} \left\{ \frac{1}{N} \sum_{i=1}^N x_i[n] \right\} \tag{3.18}$$

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

$$\mu[n] \neq \mu[m] \quad \text{ที่ } n \neq m \tag{3.19}$$

ตัวกระทำ Expectation ที่น่าสนใจ สามารถสรุปได้ดังนี้คือ

i) ความเป็นเชิงเส้น

$$E\{ax[n] + by[m]\} = aE\{x[n]\} + bE\{y[m]\} \tag{3.20}$$

ii) ความเป็นอิสระต่อกัน

$$E\{x[n]y[m]\} \neq E\{x[n]\} \cdot E\{y[m]\} \quad (3.21)$$

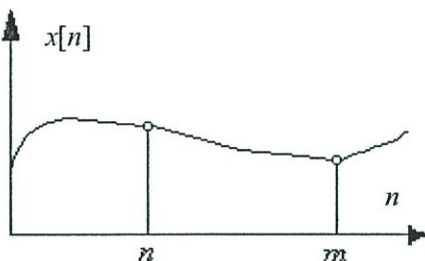
คุณสมบัติข้อนี้จะเท่ากันก็ต่อเมื่อ $x[n]$ และ $y[m]$ นั้นเป็นอิสระต่อกัน

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

$$E\{y[n]\} = \int_{-\infty}^{\infty} g(x[n]) \cdot f(x[n]) dx[n] \quad (3.22)$$

3.2.4 ค่าความเกี่ยวพัน (Correlation) เป็นตัวบอกความเหมือนของ r.v. 2 ตัว ดังตัวอย่างในรูปที่ 3.17 ตัวแปรสุ่มมีสองตัวคือ $x[n]$ และ $x[m]$ ความเกี่ยวเนื่องกันของตัวแปรทั้งสองจะสามารถเขียนให้อยู่ในรูปของสมการดังนี้

$$r(m, n) = E\{x[m] \cdot x[n]\} \quad (3.23)$$



รูปที่ 3.17 ตัวแปรสุ่มสองตัว

หรือ

$$r(m, n) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x[m]x[n]f(x[m], x[n])dx[m]dx[n] \quad (3.24)$$

และในรูปของ Relative frequency จะได้ว่า

$$r(m, n) = \lim_{N \rightarrow \infty} \left\{ \frac{1}{N} \sum_{i=1}^N x_i[m], x_i[n] \right\} \quad (3.25)$$

กรณีพิเศษเมื่อ $n = m$ จะได้

$$r(m, n) = E\{x^2[n]\} \quad (3.26)$$

ซึ่งเป็นค่ากำลังงาน หรือ พลังงานเฉลี่ยของสัญญาณใด ๆ ค่าในสมการที่ (3.23-3.29) โดยทั่วไป จะเรียกว่า Autocorrelation

3.2.5 ค่าความแปรปรวนร่วม (Covariance) ค่า Covariance กำหนดได้ดังสมการ

$$c(m, n) = E\{(x[m] - \mu[m])(x[n] - \mu[n])\} \quad (3.27)$$

ถ้า $\mu[n] = \mu[m] = 0$ จะได้ว่า Covariance จะมีค่าเท่ากับ Autocorrelation จากสมการที่ (3.27) จะได้ว่า

$$\begin{aligned} c(m, n) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x[m] - \mu[m])(x[n] - \mu[n]) f(x[m], x[n]) dx[m] dx[n] \\ &= E\{x[m]x[n]\} - \mu[n]\mu[m] \end{aligned} \quad (3.28)$$

จากสมการที่ (3.27) ถ้าหากว่า $m = n$ แล้วจะได้ว่า

$$\begin{aligned} c(n, n) &= E\{(x[n] - \mu[n])^2\} \\ &= \sigma_n^2 \end{aligned} \quad (3.29)$$

จากสมการที่ (3.29) เรียกว่า ค่าความแปรปรวน หรือ Variance นั้นเอง และจากสมการที่ (3.29) จะสามารถหาค่า Variance ได้ดังนี้

$$\begin{aligned} \sigma_n^2 &= \int (x[n] - \mu[n])^2 f[x] dx \\ &= E\{x^2[n]\} - (E\{x[n]\})^2 \end{aligned} \quad (3.30)$$

ในกรณีของกระบวนการสุ่มแบบ Gaussian จะได้ว่า

$$\sigma_n^2 = 1 \quad (3.31)$$

3.2.6 ความเป็นอิสระ (Independence) จะกล่าวว่า $\{x_1[n]\}$ และ $\{x_2[n]\}$ เป็นอิสระต่อกันก็ต่อเมื่อ

$$f(x_1[n], x_2[n]) = f(x_1[n]) \cdot f(x_2[n]) \quad (3.32)$$

และ

$$E\{x_1[n], x_2[n]\} = E\{x_1[n]\} \cdot E\{x_2[n]\} \quad (3.33)$$

หรือแม้กระทั่ง $x_1[n]$ และ $x_2[n]$ จะเป็นสัญญาณ $x[n]$ เดียวกันก็ได้ โดยที่ $x_1[n] = x_2[m]$ และ $x_2[n] = x[m+n]$ จากสมการที่ (3.28) หากกล่าวว่า $x[m]$ และ $x[n]$ เป็นอิสระต่อกันแล้วจะได้ว่า

$$c(m, n) = E\{x[m]x[n]\} - \mu[n]\mu[m] = 0 \quad (3.34)$$

หรือกล่าวอีกนัยหนึ่งก็คือ ตัวแปรสุ่มทั้งสองไม่เกี่ยวข้องกัน (Uncorrelated) สรุปก็คือ Independent samples จะ Uncorrelated ส่วน Uncorrelated samples อาจจะไม่ Independent หากมันไม่เป็น Gaussian Distribution

3.2.7 Orthogonality หาก $x[m]$ และ $x[n]$ นั้น Uncorrelated กัน และ $\mu = 0$ จะกล่าวว่ามัน Orthogonal กัน สัญลักษณ์ที่ใช้คือ \perp หรือเขียนในเทอมของ $E\{\cdot\}$ จะได้ว่า

$$E\{x[m] \cdot x[n]\} = 0 \quad (3.35)$$

3.2.8 Stationarity ความหมายที่แท้จริงของ Stationary คือ ค่าของ $f_x(x[n])$ จะต้องมีค่าเท่ากัน สำหรับทุก ๆ ค่าของ n หรือ จะมีชื่อเรียกอีกอย่างหนึ่งว่า Strictly Stationary ซึ่งสามารถเขียนเป็นสมการได้ดังนี้

$$f(x[n_1 + n_0], x[n_2 + n_0], \dots, x[n_k + n_0]) = f(x[n_1], \dots, x[n_k]) \quad (3.36)$$

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

i)

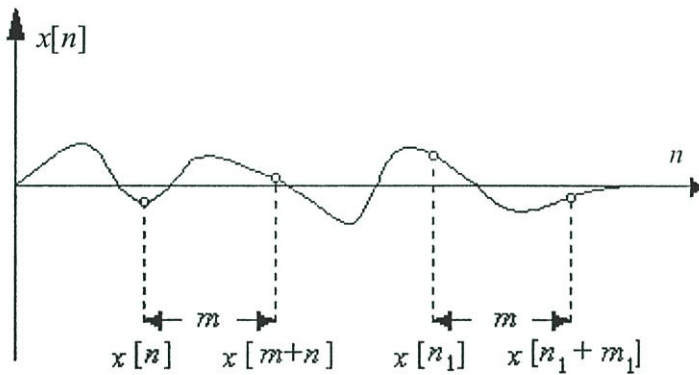
$$E\{x[m]\} = E\{x[n]\} = \mu \quad (3.37)$$

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

ii)

$$r(m) = E\{x[n] \cdot x[n+m]\} \quad (3.38)$$

หมายความว่า ค่า Autocorrelation จะขึ้นกับค่าของ m เท่านั้น ดังรูปที่ 3.18 จะได้ว่า Autocorrelation ของ $x[n]$ กับ $x[n+m]$ มีค่าเท่ากับ Autocorrelation ของ $x[n_1]$ กับ $x[n_1+m]$

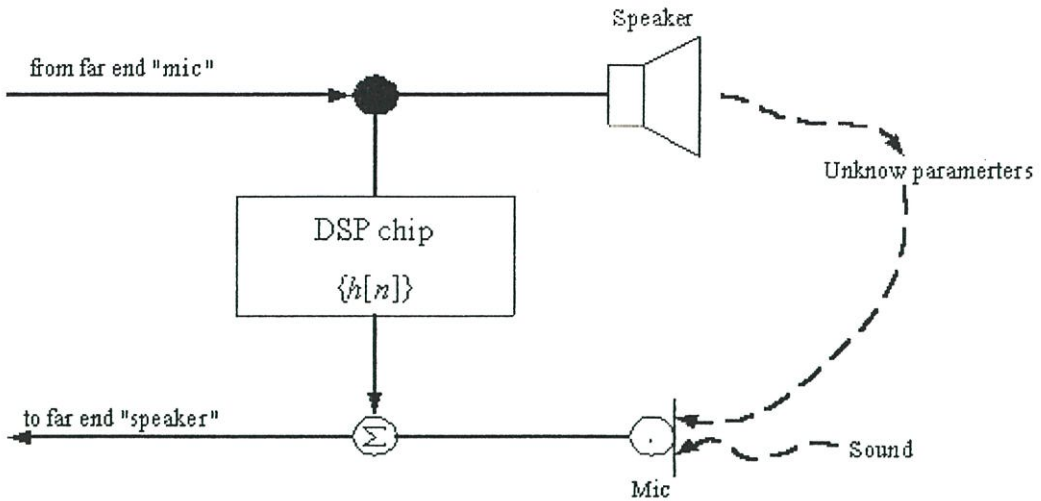


รูปที่ 3.18 อธิบายความหมายของ Autocorrelation

3.2.9 ความสัมพันธ์แบบไขว้ (Cross-correlation) Cross-correlation จะมีนิยามตามสมการ
คือ

$$r_{xy}(m) = E\{x[n]y[n+m]\} = r_{xy}[-m] \quad (3.39)$$

เพื่อให้สามารถเข้าใจได้โดยง่ายจะขอยกตัวอย่างการประยุกต์ใช้งาน Cross-correlation โดยนำมาทำ System Identification ของ Hand-free mobile phone ดังแสดงในรูปที่ 3.19 และรูปที่ 3.20



รูปที่ 3.19 Hand-free Mobile phone

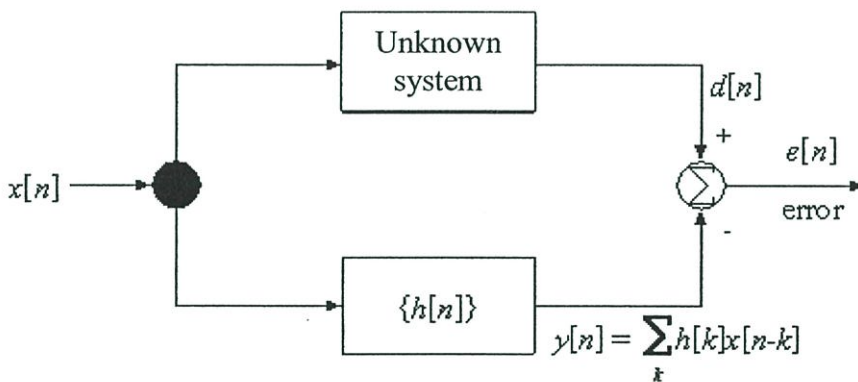
สิ่งที่ต้องการคือ $\{h[n]\}$ ที่ทำให้ $y[n]$ เหมือน $d[n]$ มากที่สุด โดยตัววัดความเหมือนกัน คือ $e[n]$ โดยจะใช้วิธีการลด $E\{e^2[n]\}$ ให้มีค่าน้อยที่สุด หรือหมายความว่า \underline{h} จะได้จาก

$$\underline{h} = R_{xx}^{-1} r_{dx} \quad (3.40)$$

เมื่อ

\underline{h} = Impulse Response ของ Unknown System

R_{xx} = Autocorrelation ของสัญญาณอินพุต



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

รูปที่ 3.20 บล็อกไดอะแกรมของรูปที่ 3.19

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

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

$$r_{xx}(m) = E\{x[n]x[n+m]\} \quad (3.41)$$

$$c_{xx}(m) = E\{x[n] - \mu_x)(x[n+m] - \mu_x)\} \quad (3.42)$$

$$r_{xy}(m) = E\{x[n]y[n+m]\} \quad (3.43)$$

$$c_{xy}(m) = E\{x[n] - \mu_x)(y[n+m] - \mu_y)\} \quad (3.44)$$

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

i)

$$c_{xx}(m) = r_{xx}(m) - \mu_x^2 \quad (3.45)$$

$$c_{xy}(m) = r_{xy}(m) - \mu_x\mu_y \quad (3.46)$$

จากสมการที่ (3.45) และ (3.46) ถ้า $\mu_x = 0$ จะทำให้ Correlation และ Covariance ค่าเท่ากัน

ii)

$$r_{xx}(0) = E\{x^2[n]\} \text{ (ค่า กำลังสองเฉลี่ย)} \quad (3.47)$$

$$c_{xx}(0) = \sigma_x^2 \quad (3.48)$$

iii)

$$r_{xx}(m) = r_{xx}(-m) \quad (3.49)$$

$$c_{xx}(m) = c_{xx}(-m) \quad (3.50)$$

$$r_{xy}(m) = r_{xy}(-m) \quad (3.51)$$

$$c_{xy}(m) = c_{xy}(-m) \quad (3.52)$$

iv) ถ้า $y[n]=x[n-n_0]$ แล้วจะได้

$$r_{yy}(m) = r_{xx}(m) \quad (3.53a)$$

$$c_{yy}(m) = c_{xx}(m) \quad (3.53b)$$

v) สำหรับกระบวนการสุ่มใด ๆ เมื่อ $m \rightarrow \infty$ แล้วจะทำให้ตัวแปรสุ่มมีการ Uncorrelated กัน น้อยมากดังนั้นจะได้

$$\lim_{m \rightarrow \infty} r_{xx}(m) = (E\{x[n]\})^2 = \mu_x^2 \quad (3.54)$$

$$\lim_{m \rightarrow \infty} c_{xx}(m) = 0 \quad (3.55)$$

$$\lim_{m \rightarrow \infty} r_{xy}(m) = \mu_x \mu_y \quad (3.56a)$$

$$\lim_{m \rightarrow \infty} c_{xy}(m) = 0 \quad (3.56b)$$

3.3 การแทนสัญญาณที่มีพลังงานไม่จำกัดด้วยสเปกตรัม [8]

แม้ว่าจะไม่สามารถทำการแปลง z (z -transform) กับสัญญาณที่มีพลังงานไม่จำกัด (Infinite-Energy Signals) ได้เนื่องจากสัญญาณดังกล่าวมีลักษณะไม่เป็นรายคาบ (Aperiodic) อย่างไรก็ตาม เมื่อสัญญาณนี้มาทำ Autocorrelation และ ทำ Autocovariance ค่าที่ได้จะมีลักษณะเป็นรายคาบเกิดขึ้น จึงทำให้สามารถ นำเอาการแปลง z และการแปลงฟูเรียร์ (Fourier Transform) มาใช้สำหรับ หาสเปกตรัมของสัญญาณที่มีพลังงานไม่จำกัดเหล่านี้ได้ การแทนสัญญาณเหล่านี้ด้วยสเปกตรัม มีความสำคัญมากต่อการอธิบายความสัมพันธ์ระหว่างสัญญาณอินพุต และสัญญาณเอาต์พุตของระบบเชิงเส้นไม่แปรตามเวลา หรือ LTI เมื่อสัญญาณที่มีพลังงานไม่จำกัด

3.3.1 การแปลง Z ของ Correlation และ Covariance กำหนดให้ $R_{xx}(z), C_{xx}(z), R_{xy}(z)$ และ $C_{xy}(z)$ คือ การแปลง Z ของ $r_{xx}(m), c_{xx}(m), r_{xy}(m)$ และ $c_{xy}(m)$ ตามลำดับ จากสมการที่ (3.54) และ (3.56) จะทราบได้ทันทีว่า การแปลง Z จะมีค่าก็ต่อเมื่อ $\mu_x = 0$ เท่านั้น ซึ่งจะทำให้ $R_{xx}(z) = C_{xx}(z)$ และ $R_{xy}(z) = C_{xy}(z)$ ซึ่งมีคุณสมบัติที่น่าสนใจดังนี้

i)

$$\sigma_x^2 = \frac{1}{2\pi j} \oint_{close} C_{xx}(z) z^{-1} dz \quad (3.57)$$

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

ii)

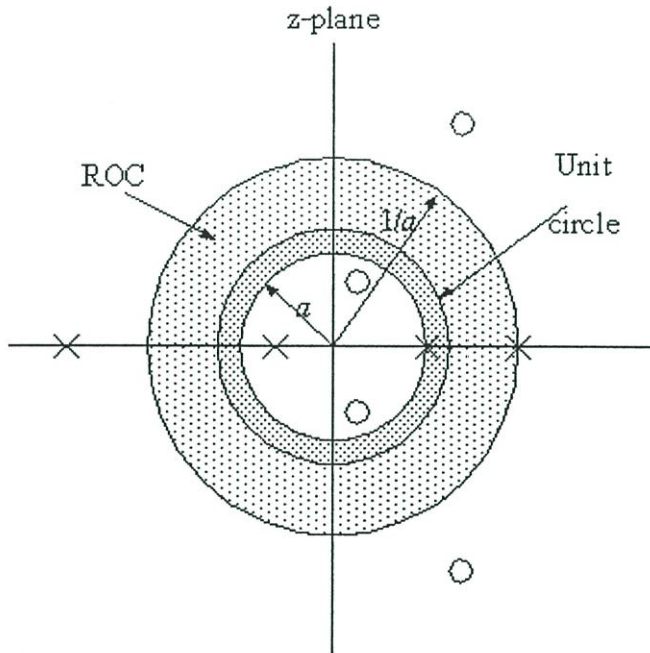
$$C_{xx}(z) = C_{xx}(1/z) \quad (3.58a)$$

$$C_{xy}(z) = C_{xy}(1/z) \quad (3.58b)$$

สมการที่ (3.58) ได้รับโดยตรงจากการแปลง Z สมการที่ (3.49) ถึง (3.52) และ ROC ของ $C_{xx}(z)$ จะเป็นไปตามเงื่อนไขดังนี้ คือ

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

เมื่อ $0 < a < 1$ เนื่องจาก $C_{xx}(z) \rightarrow 0$ เมื่อ $m \rightarrow \infty$ ROC จะต้องอยู่ภายในวงกลมหนึ่งหน่วย ในกรณีที่ $C_{xx}(z)$ เป็นฟังก์ชันแบบเศษส่วน (Rational function) ของตัวแปร Z ดูเหมือนว่า โพลและซีโรของ $C_{xx}(z)$ จะเกิดเป็นคู่ Complex conjugate กัน ดังแสดงในรูปที่ 3.21



รูปที่ 3.21 ROC และตำแหน่งโพลและซีโรที่ได้จากการแปลง Z ของ $C_{xx}(z)$

3.3.2 สเปกตรัมกำลังงาน เนื่องจากว่า ROC จะอยู่ในวงกลมหนึ่งหน่วย จะทำให้สามารถเขียนสมการที่ (3.57) ใหม่ได้ดังนี้

$$\sigma_x^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} P_{xx}(\omega) d\omega \quad (3.59)$$

โดยที่

$$P_{xx}(\omega) = C_{xx}(e^{j\omega}) \quad (3.60)$$

เมื่อ $\mu_x = 0$ จะเรียกว่า variance ซึ่งก็คือค่า mean-square หรือ กำลังงานเฉลี่ยนั่นเอง ดังนั้นพื้นที่ภายใน $P_{xx}(\omega)$ สำหรับ $-\pi \leq \omega \leq \pi$ จะเป็นสัดส่วนโดยตรงกับกำลังงานเฉลี่ยของสัญญาณและเช่นเดียวกัน 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 $r_{xx}(m) \rightarrow \mu_x^2$ เมื่อ $m \rightarrow \infty$ ในกรณีเช่นนี้ จะเลือกใช้สมการที่ (3.61) มาใช้แทนสมการที่ (3.60) จากคุณสมบัติข้อ ii) ในหัวข้อการแปลง Z ของ Correlation และ Covariance ทำให้ทราบว่า $P_{xx}(\omega)$ เป็นฟังก์ชันแบบสมมาตร คือ $P_{xx}(\omega) = P_{xx}(-\omega)$ และก็เป็นความจริงที่ว่า ความหนาแน่นกำลังงานไม่มีค่าเป็นลบแน่นอน

ในทำนองเดียวกันจะสามารถนิยาม ความหนาแน่นกำลังงานแบบไขว้ได้ดังนี้

$$P_{xy}(\omega) = C_{xy}(e^{j\omega}) \quad (3.61)$$

ในทำนองเดียวกัน

$$P_{xx}(\omega) = P_{xx}(-\omega) \quad (3.62)$$

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

$$y[n] = \sum_{k=-\infty}^{\infty} h[n-k]x[k] = \sum_{k=-\infty}^{\infty} h[k]x[n-k] \quad (3.63)$$

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 ค่า Mean ค่า Variance และ Autocorrelation ซึ่งจะสามารถหาความสัมพันธ์ระหว่างปริมาณทั้งสองได้ดังนี้

กำหนดให้ค่า mean ของสัญญาณเอาต์พุตเป็น

$$\begin{aligned} \mu_y = E\{y[n]\} &= \sum_{k=-\infty}^{\infty} h[k]E\{x[n-k]\} \\ &= \mu_x \sum_{k=-\infty}^{\infty} h[k] \end{aligned} \quad (3.64)$$

จากความจริงที่ว่าค่าคาดหวังของผลบวก จะเท่ากับผลบวกของค่าคาดหวัง ในเทอมของฟังก์ชันระบบ จะเขียนได้ดังสมการ

$$\mu_y = H(e^{j0})\mu_x \quad (3.65)$$

จากสมการที่ (3.65) เป็นจริงทั้งนี้เพราะ สัญญาณอินพุตเป็น Stationary จึงทำให้สัญญาณเอาต์พุตคงที่ด้วย

จากนั้นสมมติว่า สัญญาณเอาต์พุตมีลักษณะเป็น Nonstationary ชั่วคราว จะสามารถกำหนดค่าฟังก์ชัน Autocorrelation ของสัญญาณเอาต์พุตได้ดังสมการ

$$\begin{aligned}
r'_{xx}(n, n+m) &= E\{y[n]y[n+m]\} \\
&= E\left\{\sum_{k=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} h[k]h[j]x[n-k]x[n+m-j]\right\} \\
&= \sum_{k=-\infty}^{\infty} h[k] \sum_{j=-\infty}^{\infty} h[j]E\{x[n-k]x[n+m-j]\}
\end{aligned} \tag{3.66}$$

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

$$r_{yy}(n, n+m) = \sum_{k=-\infty}^{\infty} h[k] \sum_{j=-\infty}^{\infty} h[j]r_{xx}[m+k-j] = r_{yy}[m] \tag{3.67}$$

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

ถ้าแทน $l = j - k$ ลงในสมการที่ (3.67) แล้วจะสามารถเขียนใหม่ได้ดังสมการ

$$\begin{aligned}
r_{yy}(m) &= \sum_{l=-\infty}^{\infty} r_{xx}[m-l] \sum_{k=-\infty}^{\infty} h[k]h[l+k] \\
&= \sum_{k=-\infty}^{\infty} r_{xx}[m-l]v[l]
\end{aligned} \tag{3.68}$$

เมื่อกำหนดให้

$$v[l] = \sum_{k=-\infty}^{\infty} h[k]h[l+k] \tag{3.69}$$

สมการที่ (3.69) เรียกว่า Aperiodic autocorrelation sequence หรือเรียกง่าย ๆ ว่า Autocorrelation ที่จริงแล้ว สมการที่ (3.69) ก็คือ การคอนโวลูชันแบบไม่ต่อเนื่องทางเวลาของ $h[n]$ กับ $h[-n]$ นั่นเอง

สมมติว่า $\mu_x = 0$ ทำการแปลง Z สมการที่ (3.68) และ สมการที่ (3.69) จะได้ว่า

$$\begin{aligned}
R_{yy}(z) &= V(z)R_{xx}(z) \\
&= H(z)H(z^{-1})R_{xx}(z)
\end{aligned} \tag{3.70}$$

ในเทอมของความหนาแน่นกำลังงานจะได้ว่า

$$P_{yy} = |H(e^{j\omega})|^2 P_{xx}(\omega) \quad (3.71)$$

สมมติว่า $\mu_x = 0$ และจากสมการที่ (3.69) $\mu_y = 0$ เหมือนกัน ดังนั้นจะได้ว่า

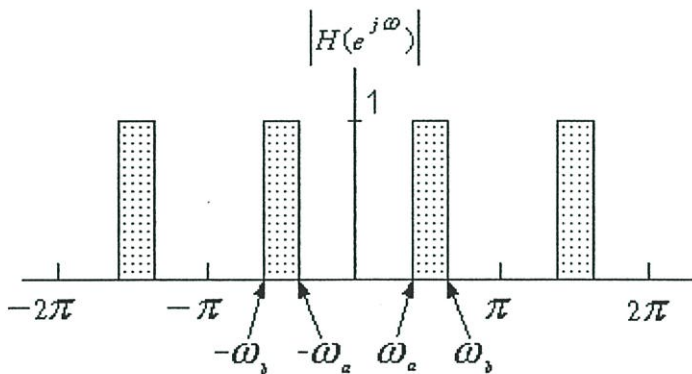
$$\begin{aligned} r_{yy}(0) &= \frac{1}{2\pi} \int_{-\pi}^{\pi} P_{yy}(\omega) d\omega \\ &= \text{กำลังงานเฉลี่ยรวมที่เอาต์พุต} \end{aligned} \quad (3.72)$$

แทนสมการที่ (3.71) ลงในสมการที่ (3.72) จะได้

$$r_{yy}(0) = \frac{1}{2\pi} \int_{-\pi}^{\pi} |H(e^{j\omega})|^2 P_{xx}(\omega) d\omega \quad (3.73)$$

สมมติว่า $H(e^{j\omega})$ เป็นตัวกรองแบบแถบผ่านในอุดมคติ ดังแสดงในรูปที่ 3.22 จากที่ทราบว่ $r_{xx}(m)$ เป็นฟังก์ชันคู่ ดังนั้น

$$P_{xx}(\omega) = P_{xx}(-\omega) \quad (3.74)$$



รูปที่ 3.22 ผลตอบสนองทางความถี่ของตัวกรองแบบแถบผ่านในอุดมคติ

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

$$\begin{aligned} r_{yy}(0) &= \text{กำลังเฉลี่ยเอาต์พุต} \\ &= \frac{1}{\pi} \int_{\omega_c}^{\omega_s} P_{xx}(\omega) d\omega \end{aligned} \quad (3.75)$$

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

3.4 สรุป

ในบทนี้ได้กล่าวถึงทฤษฎีพื้นฐานของกระบวนการสุ่มแบบไม่ต่อเนื่องทางเวลา ซึ่งเกี่ยวข้องโดยตรงกับการประมวลผลสัญญาณดิจิทัล โดยเริ่มจากการกล่าวทั่วๆ ไปเกี่ยวกับตัวแปรสุ่ม และกระบวนการสุ่มหลายตัว ถึงแม้ว่าการที่จะอธิบายถึงคุณสมบัติของกระบวนการสุ่มได้อย่างสมบูรณ์นั้น จำเป็นจะต้องทราบ Joint distribution หรือ Density function อย่างไรก็ตาม จะสามารถอธิบายกระบวนการสุ่มได้ด้วย ค่า Mean ค่า Variance และ Autocorrelation โดยการใช้ค่าเฉลี่ยแบบ Ensemble แทน Density function ได้ จากนั้นได้กล่าวถึง กระบวนการสุ่มที่มีคุณสมบัติเป็น Stationary และแบบ Wide-Sense Stationary (WSS) ซึ่งพบว่า ค่า Mean ของกระบวนการสุ่มนี้จะมีค่าคงที่ไม่ขึ้นกับเวลา นอกจากนี้ค่า Autocorrelation $E\{x[n]x[n+m]\}$ จะขึ้นกับค่าความแตกต่างทางเวลา m เท่านั้น ถ้าพิจารณาเวลาช่วงสั้น ๆ ก็อาจจะสมมติได้ว่า กระบวนการสุ่มนั้นมีคุณสมบัติเป็น WSS ได้ ในบางปัญหาที่มีความจำเป็นจะต้องทราบค่าทางสถิติอันดับ 1 (Mean, Variance) และค่าทางสถิติอันดับ 2 (Correlation, Covariance) ของกระบวนการสุ่ม ซึ่งเป็นไปไม่ได้ที่จะทราบล่วงหน้า ดังนั้นจึงจำเป็นต้องใช้วิธีการประมาณค่าทั้งสอง ซึ่งมีด้วยกันสองแบบคือ ใช้การเฉลี่ยแบบ Ensemble และ ค่าเฉลี่ยทางเวลา ในกรณีที่ค่าเฉลี่ยทั้งสองเท่ากัน จะเรียกกระบวนการสุ่มนั้นว่าเป็น Ergodic

ปริมาณทางสถิติที่สำคัญอีกอันหนึ่งที่ได้กล่าวถึงในบทนี้ คือ สเปกตรัมกำลังงาน ซึ่งเป็นการแปลงฟูเรียร์ของกระบวนการสุ่มแบบ WSS และยังสามารถแสดงให้เห็นถึงความสัมพันธ์ระหว่างสัญญาณอินพุต และสัญญาณเอาต์พุตจากระบบ LTI ให้เห็นอีกด้วย ซึ่งทฤษฎีทั้งหมดที่ได้กล่าวถึงในบทนี้ จะเป็นทฤษฎีที่ต้องใช้สำหรับการศึกษา การประมวลผลสัญญาณแบบอะแคปทีฟในบทต่อไป

บทที่ 4

ตัวกรองความถี่แบบอะแดปทีฟ

ตัวกรองความถี่แบบอะแดปทีฟ (Adaptive filtering) ได้เข้ามามีบทบาทเป็นอย่างมากในงานประมวลผลสัญญาณดิจิทัล ทั้งนี้เพราะสามารถปรับเปลี่ยนคุณลักษณะของตัวกรองให้สอดคล้องกับสัญญาณที่เข้ามาได้ด้วยตัวเอง ในบทนี้จะกล่าวถึงพื้นฐานเกี่ยวกับตัวกรองความถี่ดิจิทัล และตัวกรองความถี่แบบอะแดปทีฟ ซึ่งจะใช้เป็นพื้นฐานในการพัฒนาตัวกรองความถี่แบบอะแดปทีฟแบบใหม่ที่น่าสนใจในวิทยานิพนธ์นี้

ปัจจุบันการประมวลผลสัญญาณดิจิทัล ได้มีการนำตัวกรองความถี่ดิจิทัล (Digital filtering) เข้ามาประยุกต์ใช้ในงานประมวลผลสัญญาณดิจิทัล และระบบการควบคุม ตัวกรองความถี่แบบดิจิทัล แบ่งออกได้เป็นสองชนิด [9] คือ

1. ตัวกรองความถี่แบบ Finite Impulse Response (FIR filter) เป็นตัวกรองความถี่ชนิดที่มีฟังก์ชันถ่ายโอน (Transfer function) มีเฉพาะซีโร (Zero) ไม่มีโพล (Pole) ทำให้ระบบมีความเสถียรภาพอย่างแน่นอน และมีผลตอบสนองทางเฟสเป็นเชิงเส้น (Linear phase) แต่ก็มีข้อเสียตรงที่ต้องใช้จำนวนอันดับ (Order) สูง จึงจะทำให้ตัวกรองทำงานได้ดี ทำให้การคำนวณเพิ่มมากขึ้นตามไปด้วย

2. ตัวกรองความถี่แบบ Infinite Impulse Response (IIR filter) ฟังก์ชันถ่ายโอน (Transfer function) จะมีทั้งซีโร (Zero) และโพล (Pole) ทำให้มีปัญหาทางด้านเสถียรภาพ แต่ถ้ามีการออกแบบที่ดี จะทำให้ปัญหาดังกล่าวลดลง หรือไม่เกิดขึ้นเลย ข้อดีของตัวกรองความถี่แบบ IIR เมื่อเทียบกับแบบ FIR คือ ที่คุณภาพของการทำงาน (Performance) เท่ากัน ตัวกรองความถี่แบบ IIR จะใช้จำนวนอันดับน้อยกว่า ซึ่งทำให้การคำนวณน้อยกว่าด้วย ความซับซ้อนในการคำนวณนี้มีความสำคัญมาก เมื่อนำตัวกรองความถี่ไปใช้งานที่เวลาจริง (Real time) ด้วยไมโครโปรเซสเซอร์ ซึ่งมีข้อจำกัดทางด้านจำนวนบิต และความเร็วในการทำงานตัวกรองความถี่ที่มีจำนวนการคำนวณน้อยกว่า จะทำให้ไมโครโปรเซสเซอร์ทำงานน้อยลง และทำงานได้เร็วกว่า

ในหัวข้อนี้จะกล่าวถึงทฤษฎีทั่วไปที่เกี่ยวกับตัวกรองแบบ IIR เพียงอย่างเดียว เนื่องจากในวิทยานิพนธ์นี้จะนำพื้นฐานของตัวกรองความถี่แบบ IIR ไปประยุกต์ในการพัฒนาตัวกรองความถี่แบบใหม่ ที่จะนำเสนอในบทต่อไป

4.1 ตัวกรองความถี่แบบ IIR (IIR filters)

ตัวกรองความถี่แบบ IIR เป็นส่วนประกอบที่สำคัญอันหนึ่งในระบบการประมวลผลแบบไม่ต่อเนื่องทางเวลา (Discrete-time processing) มีข้อดีหลาย ๆ ข้อที่เหนือกว่าตัวกรองแบบ 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 จะมีสมการของสัญญาณเอาต์พุตที่เป็นฟังก์ชันของสัญญาณอินพุต ปัจจุบัน อินพุตในอดีต และสัญญาณเอาต์พุตในอดีต (สัญญาณเอาต์พุตในอดีตได้จากการป้อนกลับ) ซึ่งสามารถเขียนให้อยู่ในรูปสมการผลต่าง (Difference equation) ได้ดังนี้

$$y(n) = -a_1y(n-1) - a_2y(n-2) - \dots - a_Ny(n-N) + b_0x(n) + b_1x(n-1) + \dots + b_Mx(n-M) \quad (4.1)$$

เมื่อ $x[n]$ คือ ลำดับของสัญญาณอินพุต และ $y[n]$ คือ ลำดับของสัญญาณเอาต์พุต N คือ จำนวนตัวอย่าง (Samples) ทั้งหมดของสัญญาณเอาต์พุตที่ทราบค่า สมการที่ (4.1) สามารถนำไปใช้ในการคำนวณหาสัญญาณเอาต์พุตต่อ ๆ กันไปได้ รูปแบบของการนำเอาสัญญาณเอาต์พุตก่อนหน้ามาใช้สำหรับคำนวณหาสัญญาณเอาต์พุตตัวต่อไปจะเรียกว่า รีเคอร์ซีฟ (Recursive) โดยทั่วไป ตัวกรองความถี่แบบ IIR และตัวกรองความถี่แบบรีเคอร์ซีฟ มักจะนำมาใช้ในความหมายอย่างเดียวกัน ทั้งนี้เพราะจากสมการที่ (4.1) สามารถนำไปใช้สร้างตัวกรองความถี่ทั้งสองแบบได้เหมือนกัน คำว่า IIR จะหมายถึงรูปแบบของผลตอบสนองอิมพัลส์ (Impulse response) ของตัวกรองความถี่ ในขณะที่คำว่า รีเคอร์ซีฟ จะหมายถึงตัวกรองความถี่ที่ถูกสร้างขึ้นอย่างไร ตัวกรองความถี่แบบ FIR จะสามารถสร้างในรูปแบบของรีเคอร์ซีฟ ได้ด้วย ในขณะเดียวกันตัวกรองความถี่แบบ IIR ก็จะสามารถสร้างในรูปแบบนอนรีเคอร์ซีฟ (Nonrecursive) ได้เหมือนกัน

เพื่อความสะดวกจะนิยามตัวกรองความถี่แบบรีเคอร์ซีฟด้วย ฟังก์ชันถ่ายโอน หรือฟังก์ชันของระบบ (System function) ฟังก์ชันของระบบก็คือการแปลง Z (Z-transform) ของผลตอบสนองอิมพัลส์ของตัวกรองความถี่ โดยฟังก์ชันของระบบนี้จะอยู่ในรูปเศษส่วน (Rational function) ในตัวแปร z^{-1} ระบบตามสมการที่ (4.1) จะมีฟังก์ชันของระบบเป็นไปตามสมการ คือ

$$H(z) = \frac{\sum_{k=0}^M b_k z^{-k}}{1 + \sum_{k=1}^N a_k z^{-k}} \quad (4.2)$$

จากสมการที่ (4.2) ถ้ากำหนดให้เงื่อนไขเริ่มต้นเป็นศูนย์ จะสามารถเขียนใหม่ได้คือ

$$H(z) = G \frac{\prod_{k=0}^M (1 - \beta_k z^{-1})}{\prod_{k=1}^N (1 - \alpha_k z^{-1})} \quad (4.3)$$

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

ตัวกรองความถี่ที่มีความเป็นเชิงเส้นไม่แปรตามเวลา (Linear Time-Invariance: LTI) จะมีคุณสมบัติเป็นคอซอล (Causal) ถ้าผลตอบสนองอิมพัลส์มีค่าเท่ากับศูนย์เมื่อ $x(n) = 0$ ตัวอย่างสัญญาณเอาต์พุตของตัวกรองความถี่ที่เป็นคอซอลจะขึ้นกับตัวอย่างสัญญาณอินพุตปัจจุบัน และในอดีตเท่านั้น ถ้าจำกัดสัญญาณอินพุตให้มีค่าเป็นศูนย์ $x(n) = 0$ สำหรับ $n < 0$ และค่าเริ่มต้นของ $y(-1) = y(-2) = \dots = y(-N) = 0$ จะทำให้ตัวกรองความถี่แบบรีเคอร์ซีฟตามสมการที่ (4.1) มีคุณสมบัติเป็นคอซอล ความเป็นคอซอลของตัวกรองความถี่ มีความสำคัญมากในการประยุกต์ใช้งานที่เวลาจริง (Real time) เมื่อมีการทิก (Tick) ของสัญญาณนาฬิกาจะได้ตัวอย่างอินพุต 1 ตัวอย่าง จากนั้นตัวกรองความถี่จะต้องสร้างตัวอย่างของเอาต์พุตออกมา 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

4.2 โครงสร้างแบบ Direct Forms ของตัวกรองความถี่ IIR (Structures for IIR filter)

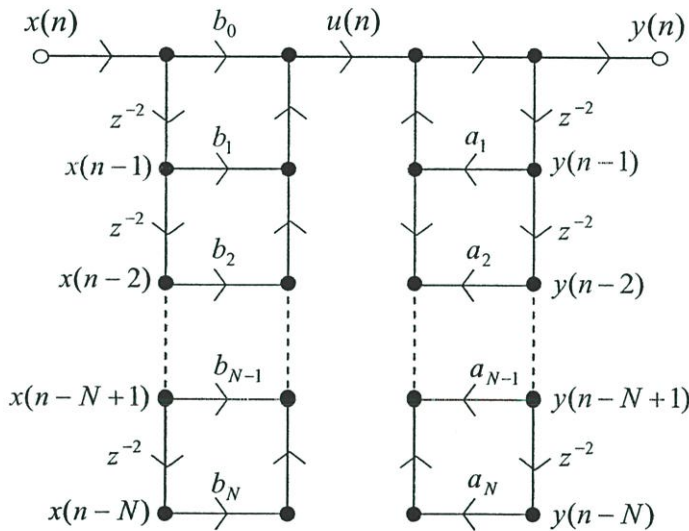
หัวข้อนี้จะกล่าวถึงโครงสร้างของตัวกรองความถี่แบบ IIR โดยแสดงในรูปของ Signal flow graph [15] โดยจะกล่าวถึงเฉพาะโครงสร้างแบบ Direct Forms เนื่องจากเป็นโครงสร้างที่ใช้ในวิทยานิพนธ์ฉบับนี้ โดยทั่วไปแล้วโครงสร้างที่ดีและเหมาะที่จะนำไปใช้งานจะต้องเป็นโครงสร้างที่มีความซับซ้อนในการคำนวณน้อย คือ มีการคูณและตัวหน่วง (Delay) ที่น้อย ทั้งนี้เพราะโดยทั่วไปแล้วการคูณ จะต้องใช้เวลามาก และมีราคาสูงเมื่อนำไปทำเป็นฮาร์ดแวร์ จำนวนตัวหน่วงหมายถึง จำนวนหน่วยความจำ ซึ่งหากมีตัวหน่วงน้อย หน่วยความจำที่ใช้ก็น้อยลงด้วย และสิ่งที่ได้จากลดจำนวนการคูณอีกอย่างก็คือจะทำให้ความเร็วในการทำงานของ CPU เร็วขึ้น

จากสมการที่ (4.1) และ (4.2) ประกอบไปด้วยการบวกจำนวน $M + 1$ ครั้ง สำหรับพจน์อินพุต $x(n), x(n-1), \dots, x(n-M)$ และ N ครั้งสำหรับพจน์เอาต์พุต $y(n-1), y(n-2), \dots, y(n-N)$ แต่ละพจน์จะถูกถ่วงน้ำหนักด้วยสัมประสิทธิ์ของตัวกรองความถี่ จากสมการที่ (4.1) สามารถเขียนใหม่ได้ว่า

$$u(n) = b_0x(n) + b_1x(n-1) + b_2x(n-2) + \dots + b_Mx(n-M) \quad (4.4)$$

$$y(n) = -a_1y(n-1) - a_2y(n-2) - \dots - a_Ny(n-N) + u(n) \quad (4.5)$$

จากสมการที่ (4.4) และ (4.5) จะเห็นว่าตัวกรองความถี่แบบ IIR จะประกอบด้วยระบบย่อย (Subsystem) สองระบบต่อกันเป็นชั้น (Cascade) กัน กล่าวคือ ชั้นที่หนึ่งเป็นของฟังก์ชันเศษ ในสมการที่ (4.2) และชั้นที่สองเป็นของหนึ่งหารด้วยฟังก์ชันส่วนของสมการที่ (4.2) เมื่อนำสองส่วนนี้มารวมกันจะได้โครงสร้างแบบ Direct Form I ซึ่งสามารถเขียนเป็น Signal flow graph สำหรับตัวกรองความถี่อันดับ N ตามรูปที่ 4.1



รูปที่ 4.1 Signal flow graph ของโครงสร้างแบบ Direct Form I สำหรับระบบอันดับ N

จากรูปที่ 4.1 สังเกตเห็นว่าโครงสร้างแบบ Direct Form I จะมีจำนวนพจน์ทั้งหมด $M+N+1$ พจน์ คือ $x(n), x(n-1), \dots, x(n-M), y(n-1), y(n-2), \dots, y(n-N)$ แต่ละพจน์จะถูกคูณด้วยสัมประสิทธิ์ และนำบวกกัน ซึ่งจำนวนของการคูณสำหรับตัวอย่างเอาต์พุตแต่ละค่า จะเท่ากับจำนวนสัมประสิทธิ์ที่เป็น Nontrivial เท่านั้น (การคูณด้วย 1, -1 และ 0 จะไม่นำมาคิดเป็นการคูณ)

จากสมการที่ (4.2) ตัวเศษของฟังก์ชันระบบ ต้องการสัญญาณอินพุตเป็น $x(n)$ และให้สัญญาณเอาต์พุตเป็น $u(n)$ และตัวส่วน ต้องการสัญญาณอินพุตเป็น $u(n)$ และให้สัญญาณเอาต์พุตเป็น $y(n)$ จากรูปที่ 4.1 ถ้าทำการพลิก (reverse) แต่ละระบบย่อย จะทำให้ตัวอย่างข้อมูลที่ถูเก็บ

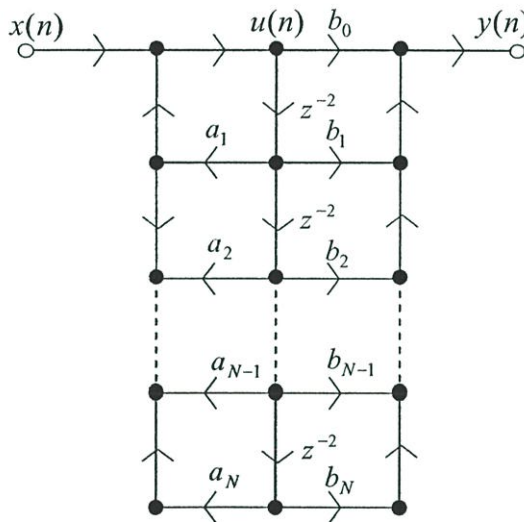
ในระบบย่อยที่ 1 สามารถเก็บไว้ร่วมกับระบบย่อยที่ 2 ได้ ซึ่งโครงสร้างลักษณะนี้จะเรียกว่า Direct Form II หรือ Canonic direct form ซึ่งมีข้อดีคือต้องการจำนวนตัวหน่วงน้อยกว่า โดยจะมีค่าสูงสุดเท่ากับ M, N และการคูณเท่ากับ $M+N+1$ ครั้ง โครงสร้างแบบ Direct Form II จะมีสมการผลต่างคือ

$$u(n) = x(n) - a_1u(n-1) - a_2u(n-2) - \dots - a_Nu(n-N) \quad (4.6)$$

$$y(n) = b_0u(n) + b_1u(n-1) + b_2u(n-2) + \dots + b_Mu(n-M) \quad (4.7)$$

และ Signal flow graph แสดงดังรูปที่ 4.2

ตัวกรองความถี่อะแดปทีฟแบบนอกรีตที่นำเสนอในวิทยานิพนธ์เป็นแบบ IIR อันดับสอง (2^{nd} order) และใช้โครงสร้างแบบ Direct form I และ เนื่องจากตัวกรองความถี่แบบ IIR จะมีปัญหาทางด้านความมีเสถียรภาพกล่าวคือ หากมีโพลเพียง 1 ตัวอยู่นอกวงกลมหนึ่งหน่วยก็จะทำให้ตัวกรองความถี่ไร้เสถียรภาพทันที

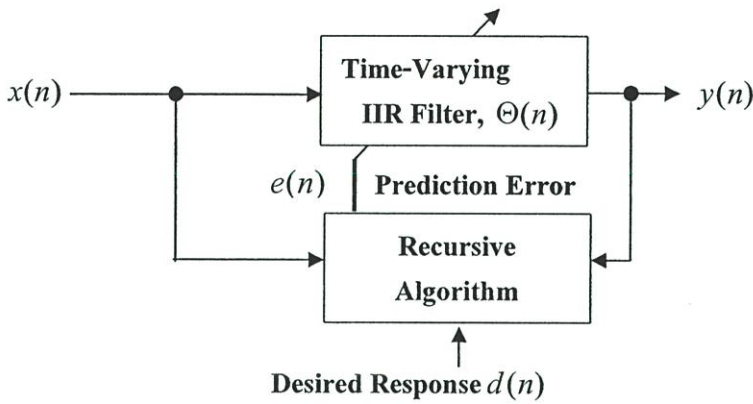


รูปที่ 4.2 Signal flow graph ของ โครงสร้างแบบ Direct Form II สำหรับระบบอันดับ N

4.3 ตัวกรองความถี่แบบอะแดปทีฟ

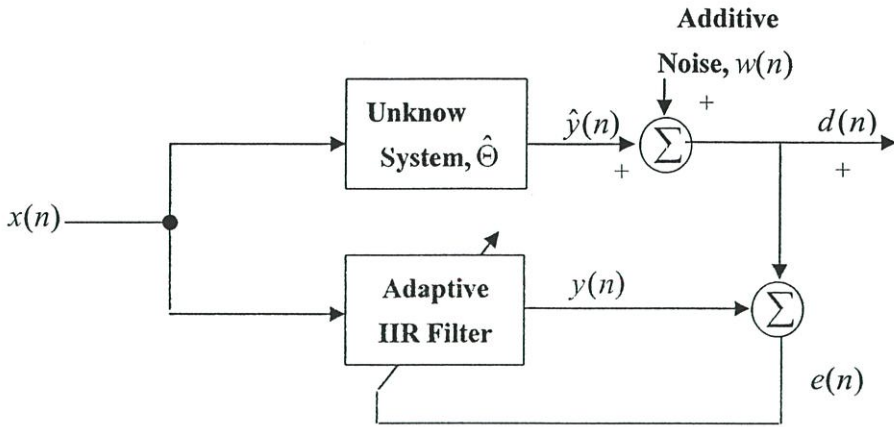
ตัวกรองความถี่แบบอะแดปทีฟ (Adaptive filtering) มีคุณสมบัติเด่นที่สามารถปรับเปลี่ยนคุณลักษณะ (Characteristics) ประจำตัวของระบบให้สอดคล้องกับสัญญาณที่รับเข้ามาได้ด้วยตัวเอง ซึ่งจะสามารถทำงานกับสัญญาณที่มีลักษณะเป็น nonstationary ซึ่งเป็นสัญญาณที่เกิดขึ้นจริงในทางปฏิบัติ ตัวกรองความถี่แบบอะแดปทีฟนั้นมีสองชนิดคือ ตัวกรองอะแดปทีฟแบบ FIR และแบบตัวกรองอะแดปทีฟแบบ IIR ตัวกรองอะแดปทีฟแบบ FIR จะปรับเฉพาะซีโรเท่านั้น เพราะไม่มีโพล ส่วนแบบ IIR จะปรับทั้งซีโรและโพลในเวลาเดียวกัน ตัวกรองอะแดปทีฟแบบ IIR นั้นได้ถูก

นำมาใช้งานต่าง ๆ แทนตัวกรองแบบ FIR มากขึ้น ทั้งนี้เนื่องจากตัวกรองอะแดปทีฟแบบ IIR จะให้สมรรถนะที่ดีกว่า FIR เมื่อเปรียบเทียบที่จำนวนอันดับเท่ากัน และจากหลักการสำคัญของ IIR ที่จะมีการใช้สัญญาณเอาต์พุตป้อนกลับไปใช้ในการประมวลผล จะทำให้ได้ผลตอบสนองต่อสัญญาณอิมพัลส์ที่มีค่าเป็นอนันต์ ทั้งที่จำนวนสัมประสิทธิ์มีค่าจำกัด นอกจากนี้ยังสามารถประมาณผลตอบสนองทางความถี่ได้อย่างมีประสิทธิภาพมากกว่า เมื่อใช้สัญญาณเอาต์พุตจากระบบที่มีทั้งโพล และซีโร และเนื่องจากตัวกรองอะแดปทีฟ IIR ใช้จำนวนสัมประสิทธิ์น้อยกว่าแบบ FIR ทำให้ลดความซับซ้อนในการคำนวณลงได้ ดังนั้นการประยุกต์ใช้งานหลาย ๆ อย่าง จึงหันมาใช้ตัวกรองอะแดปทีฟแบบ IIR มากขึ้น



รูปที่ 4.3 ส่วนประกอบโดยทั่วไปของตัวกรองความถี่อะแดปทีฟ IIR

รูปที่ 4.3 แสดงส่วนประกอบโครงสร้างทั่วไปของตัวกรองความถี่อะแดปทีฟแบบ IIR โดยที่ $x(n)$ เป็นสัญญาณอินพุต $y(n)$ เป็นสัญญาณเอาต์พุต และประกอบด้วยตัวตัวกรองความถี่แบบเปลี่ยนตามเวลา (Time-Varying Filter) โดยคุณลักษณะของตัวกรองจะถูกกำหนดด้วยพารามิเตอร์ $\Theta(n)$ และ Recursive Algorithm ใช้สำหรับปรับ $\Theta(n)$ จนทำให้สัญญาณเอาต์พุต $y(n)$ ใกล้เคียงกับ $d(n)$ ให้มากที่สุด ตัวอย่างการนำไปประยุกต์ใช้งานแสดงดังรูป 4.4



รูปที่ 4.4 ตัวอย่างการประยุกต์ใช้งานเพื่อทำ System Identification

จากรูปที่ 4.4 เป็นการทำให้ System Identification เมื่อ $\hat{\Theta}$ คือ พารามิเตอร์ที่ไม่ทราบค่า และ $d(n)$ คือ สัญญาณที่วัดได้จากเอาต์พุตของระบบ ซึ่งโดยทั่วไปจะถูกรบกวนด้วยสัญญาณรบกวน $w(n)$ วัตถุประสงค์หลักของอัลกอริทึม คือ พยายามทำให้ค่าผิดพลาด $e(n)$ มีค่าต่ำที่สุด และถือเป็นหลักพื้นฐานของการทำ Prediction Error (PE) บางครั้งเรียกว่า Estimation Error ซึ่งกำหนดได้ดังสมการ

$$e(n) = d(n) - y(n) \quad (4.8)$$

และข้อตัดสิน (Criterion) ที่นิยมใช้ คือ ค่า Mean Squar Error (MSE) ของสมการที่ (4.8) ซึ่งกำหนดได้ดังสมการ

$$\xi = E\{e^2(n)\} \quad (4.9)$$

เมื่อ $E\{\bullet\}$ คือ ตัวหาค่าคาดหวังทางสถิติ (Statistical expectation) โดยพารามิเตอร์ $\hat{\Theta}$ จะถูกปรับด้วยอัลกอริทึมแบบ Stochastic Gradient หรือ อัลกอริทึมของ Gauss-Newton นอกจากนี้ยังมีข้อตัดสินอย่างอื่นอีก เช่น วิธี Least Square (LS) หรือ Recursive Least Square (RLS) [25]

โดยพื้นฐานแล้ว จะสามารถสร้างสมการค่าผิดพลาด (Equation Error) เพื่อทำ Prediction Error ได้สองวิธี คือ Equation Error และ Output Error

1) Equation Error มีหลักการพื้นฐาน คือ สัมประสิทธิ์ที่ถูกรบกวนกลับของตัวกรอง IIR จะถูกปรับในรูปแบบ FIR (All-zero) จากนั้นจะถูกคัดลอก (Copy) ให้กับตัวกรองที่สอง ซึ่งจะสร้างในรูปแบบ IIR ที่เป็น all-pole วิธีนี้จะทำให้ตัวกรองอะแดปทีฟ IIR มีคุณสมบัติเป็น FIR ซึ่งง่ายต่อการ

ทำความเข้าใจ และสามารถวิเคราะห์หาประสิทธิภาพของตัวกรอง แต่ก็มีข้อเสีย คือ อาจจะทำให้พารามิเตอร์ $\hat{\Theta}$ มีค่า bias เกิดขึ้นได้

2) Output Error มีหลักการ คือ สัมประสิทธิ์ที่ใช้ป้อนกลับจะอยู่ในรูปของ recursive pole-zero ซึ่งไม่ทำให้เกิดค่า bias แต่จะทำให้อะแดปทีฟอัลกอริทึมลู่เข้าสู่จุด local minimum ของ ξ ทำให้ $\hat{\Theta}$ ผิดไปจากค่าที่ต้องการ และการวิเคราะห์คุณสมบัติการลู่เข้าทำได้ยาก

อัลกอริทึมแบบอะแดปทีฟที่ใช้กันมีหลายแบบ เช่น Least Mean Square (LMS), Normalised LMS (NLMS), Recursive Least Square (RLS) เป็นต้น ซึ่งจะไม่ขอกล่าวถึงรายละเอียดในที่นี้

4.4 สรุป

ในบทนี้ได้กล่าวทั่วไป เกี่ยวกับพื้นฐานตัวกรองความถี่ดิจิทัล ซึ่งถือเป็นหลักการพื้นฐานของตัวกรองความถี่แบบต่าง ๆ โครงสร้างตัวกรองความถี่ที่ใช้โครงสร้างแบบ Direct Form ซึ่งจะนำไปประยุกต์ใช้ในหัวข้อต่อไป สำหรับตัวกรองความถี่แบบอะแดปทีฟ IIR ได้กล่าวถึงโครงสร้างโดยทั่วไปของตัวกรองอะแดปทีฟ ซึ่งมีหลักสำคัญ คือ จะใช้การพิจารณาค่าความผิดพลาด $e(n)$ เป็นหลักโดยพยายามทำให้ค่า $e(n)$ มีค่าน้อยที่สุด โดยใช้ข้อตัดสิน (Criterion) หลาย ๆ แบบมาช่วย เช่น Mean Squar Error (MSE), Least Square (LS) หรือ Recursive Least Square (RLS) เป็นต้น และเนื่องจากทฤษฎีตัวกรองอะแดปทีฟ IIR ยังไม่สมบูรณ์ แต่ได้มีการนำมาใช้พอสมควร โดยทั่วไปการศึกษาตัวกรองอะแดปทีฟ IIR จะนำโปรแกรมคอมพิวเตอร์มาใช้เพื่อจำลองการทำงานแทนการวิเคราะห์ตามหลักคณิตศาสตร์ เนื่องจากความไม่เป็นเชิงเส้นของระบบนั่นเอง

บทที่ 5

ตัวกรองอะแดปทีฟ IIR แบบนอตช์

ในบทนี้จะกล่าวถึงการประยุกต์ใช้งานตัวกรองอะแดปทีฟแบบนอตช์ (Adaptive Notch Filter: ANF) ซึ่งเป็นอะแดปทีฟอัลกอริทึมเพื่อการตรวจวัดความถี่ของสัญญาณคลื่นซายน์ที่ไม่ทราบค่าในสัญญาณรบกวน ที่มีการใช้ในระบบเรดาร์ หรือ โซนา ใช้สำหรับกำจัดสัญญาณแบนด์แคบ (Narrow band) ที่ปนมากับสัญญาณแบนด์กว้าง (Broad band) ในระบบสื่อสาร หรือระบบควบคุม รวมทั้งใช้สำหรับกำจัดสัญญาณคลื่นซายน์ความถี่ 50 เฮิร์ตซ์ ในสัญญาณคลื่นไฟฟ้าหัวใจ (Electrocardiogram : ECG) ของผู้ป่วยซึ่งเหนี่ยวนำมาจากไฟฟ้ากระแสสลับ 220 โวลต์ ซึ่งทำให้สัญญาณคลื่นไฟฟ้าหัวใจ มีความผิดเพี้ยน โดยจะกล่าวเฉพาะในส่วนของอัลกอริทึมที่ใช้การปรับค่าสัมประสิทธิ์ของตัวกรองอะแดปทีฟแบบนอตช์อันดับสอง และการประยุกต์ใช้งาน อัลกอริทึมแบบเจนเนติก เพื่อหาค่าเสถียรไซส์ที่เหมาะสม ที่นำไปใช้ในการปรับค่าของสัมประสิทธิ์ของตัวกรอง และทำงานร่วมกับอะแดปทีฟอัลกอริทึมแบบใหม่ที่น่าสนใจในวิทยานิพนธ์ฉบับนี้

5.1 กล่าวนำเกี่ยวกับ ANF

ตัวกรองอะแดปทีฟแบบนอตช์ (Adaptive Notch Filter : ANF) ถูกนำมาใช้ประมาณสัญญาณคลื่นซายน์ที่ไม่ทราบความถี่ที่ปะปนอยู่ในสัญญาณรบกวน ซึ่งสัญญาณดังกล่าวจะพบได้ในระบบสื่อสารหลาย ๆ ระบบ ตัวอย่างเช่น ในระบบคิโมคูเลต จะใช้สัญญาณคลื่นซายน์เป็นสัญญาณพาห้ (Carrier) มีโอกาสเป็นไปได้ที่สัญญาณคลื่นซายน์จะมีความถี่เปลี่ยนแปลงไปเมื่อเดินทางไปถึงเครื่องรับ หรือเกิดการเลื่อนของเฟส ซึ่งจะเกิดปัญหากับด้านรับเพราะไม่สามารถกู้สัญญาณเดิมกลับมาได้ หรือกู้ได้แต่เกิดความผิดเพี้ยน ในทางปฏิบัติจำเป็นจะต้องให้สัญญาณคลื่นพาห้ที่เครื่องส่ง มีความสัมพันธ์กัน (Synchronized) ซึ่งรูปแบบของสัญญาณกำหนดได้ตามสมการ

$$x_k(n) = \sum_{k=1}^N A_k \sin(\omega_k n + \theta_k) + w(n) \quad (5.1)$$

เมื่อ A_k คือ ขนาดของสัญญาณคลื่นซายน์ตัวที่ k , ω_k คือ ความถี่ที่ไม่ทราบค่าและมีค่าอยู่ในช่วง $[0, \pi]$, θ_k คือ เฟส มีค่าอยู่ในช่วง $[0, 2\pi]$ และ $w(n)$ คือ สัญญาณรบกวน โดยสมมติว่าเป็นอิสระจากสัญญาณซายน์ ซึ่งโดยทั่วไปจะสมมติให้เป็นสัญญาณรบกวนแบบขาว มี Pdf แบบเกาส์เซียน (Gaussian white noise) วัตถุประสงค์ที่ต้องการ คือ ต้องการทราบความถี่ ω_k และ/หรือกำจัดสัญญาณซายน์ทิ้งไป หลังจากตรวจวัดความถี่ได้แล้ว A_k และ θ_k จะมีค่าเหมือนกับสัญญาณที่ด้านส่งทุกประการ (ในอุดมคติ)

ถ้าพิจารณาสัญญาณขาเข้าเพียงคลื่นเดียว และไม่มีสัญญาณรบกวนปน จะเขียนได้ดังสมการ

$$x(n) = \sin(\omega_1 n + \theta) \quad (5.2)$$

จะสามารถหาฟังก์ชัน Autocorrelation ของสมการที่ (5.2) ได้ดังนี้ คือ

$$\begin{aligned} r_{xx}(m) &= E\{x(n)x(n-m)\} \\ &= \frac{1}{2\pi} \int_0^{2\pi} \sin(\omega_1 n + \theta) \sin[\omega_1(n-m) + \theta] d\theta \\ &= \frac{\cos(\omega_1 m)}{2} \end{aligned} \quad (5.3)$$

และฟังก์ชันความหนาแน่นสเปกตรัม คือ

$$\begin{aligned} P_{xx}(e^{j\omega}) &= \sum_{m=-\infty}^{\infty} r_{xx}(m) e^{-jm\omega} \\ &= \frac{\delta(\omega - \omega_1) + \delta(\omega + \omega_1)}{2} \end{aligned} \quad (5.4)$$

เมื่อ Dirac delta function $\delta(\cdot)$ ถูกนิยามจาก kernel property คือ

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} \delta(\omega - \omega_1) f(e^{j\omega}) d\omega = f(e^{j\omega_1}) \quad (5.5)$$

โดยที่ $f(e^{j\omega_1})$ จะต้องถูกกำหนดมาก่อน

ทำการป้อนสัญญาณในสมการที่ (5.1) ให้กับตัวกรอง $H(z)$ จะได้สัญญาณเอาต์พุตจากตัวกรองสัญญาณตามสมการดังนี้

$$y(n) = H(z)[x(n) + w(n)] \quad (5.6)$$

ค่า variance ของสัญญาณเอาต์พุตมีค่าตามสมการ คือ

$$E\{y^2(n)\} = \sum_{k=1}^N \left(\frac{A_k^2}{4\pi} \int_{-\pi}^{\pi} \delta(\omega \pm \omega_k) |H(e^{j\omega})|^2 d\omega \right) + \frac{1}{2\pi} \int_{-\pi}^{\pi} P_w(e^{j\omega}) |H(e^{j\omega})|^2 d\omega \quad (5.7)$$

เมื่อ

$$P_w(z) = \sum_{m=-\infty}^{\infty} E\{w(n)w(n-m)\}z^{-k} \quad (5.8)$$

สมการที่ (5.8) คือความหนาแน่นสเปกตรัมของกระบวนการสุ่ม $[w(\cdot)]$ โดยทั่วไปจะเขียนสมการค่า variance ของสัญญาณเอาต์พุตในรูปต่อไปนี้ คือ

$$E\{y^2(n)\} = \langle H(z), H(z) \rangle_{P_{xx}} + \langle H(z), P_w(z)H(z) \rangle \quad (5.9)$$

เมื่อ $\langle \cdot, \cdot \rangle_{P_{xx}}$ คือ inner product

เพื่อให้ง่าย จะกำหนดให้สัญญาณอินพุต เป็นสัญญาณซายน์คลื่นเดี่ยวที่ปนอยู่กับสัญญาณรบกวนดังสมการ

$$x(n) + w(n) = A_1 \sin(\omega_1 n + \theta_1) + w(n) \quad (5.10)$$

ให้ $y(n) = H(z)[x(n) + w(n)]$ จะได้ variance ของสัญญาณเอาต์พุต คือ

$$E\{y^2(n)\} = A_1^2 |H(e^{j\omega})|^2 + \frac{1}{2\pi} \int_{-\pi}^{\pi} P_w(e^{-j\omega}) |H(e^{j\omega})|^2 d\omega \quad (5.11)$$

สมมติว่า ผลตอบสนองทางขนาดต่อความถี่ของ $H(e^{j\omega})$ จะเป็นดังนี้

$$|H(e^{j\omega})| = \begin{cases} 0, & \omega = \omega_0 \text{ and } \omega = -\omega_0 \\ 1, & \text{otherwise} \end{cases} \quad (5.12)$$

ซึ่งเป็นผลตอบสนองทางขนาดต่อความถี่ของตัวกรองแบบนอตช์ในอุดมคติ ผลตอบสนองทางขนาดของตัวกรองไปตามวงกลมหนึ่งหน่วยมีค่าเป็น 1 เกือบทุกความถี่ยกเว้นที่ $\omega = \pm\omega_0$ มีค่าเท่ากับ 0 ซึ่ง ω_0 นี้เรียกว่า ความถี่นอตช์ (Notch frequency) และจะได้ว่า

$$E\{y^2(n)\} = \begin{cases} E\{w^2(n)\}, & \omega_0 - \omega_1 \\ A_1^2 + E\{w^2(n)\}, & \text{otherwise} \end{cases} \quad (5.13)$$

ดังนั้น จะทำให้ output error cost function จะมีจุดนอตช์จุดเดียว โดยการทำให้ ω_0 เลื่อนติดตามตามความถี่ ω_1

ถ้าตัวกรองแบบแถบผ่าน (bandpass filter) จากตัวกรองนอกรีตในสมการที่ (5.12) ดังนี้ คือ

$$|G(e^{j\omega})|^2 = 1 - |H(e^{j\omega})|^2 = \begin{cases} 1, \omega = \pm\omega_0 \\ 0, \text{otherwise} \end{cases} \quad (5.14)$$

และสัญญาณเอาต์พุตของตัวกรองกำหนดได้ คือ

$$y(n) = G(n)[x(n) + w(n)] \quad (5.15)$$

ซึ่งจะมีเฉพาะสัญญาณชานี้เท่านั้น จะสังเกตเห็นว่า ผลตอบสนองทางความถี่ของ $|H(e^{-j\omega})|$ และ $|G(e^{j\omega})|$ นั้นเป็นฟังก์ชันแบบไม่ต่อเนื่อง (ในอุดมคติ) และเนื่องจากไม่สามารถใช้ฟังก์ชันแบบเศษส่วนสร้างตัวกรองในอุดมคติได้ จึงต้องทำให้ $H(e^{j\omega})$ มีความสัมพันธ์กับผลตอบสนองความถี่ของ stable causal filter ก่อน กล่าวคือ ด้วยความเป็น causal ทำให้สามารถเขียน $H(z)$ ในของอนุกรมได้ดังนี้ คือ

$$H(z) = h_0 + h_1 z^{-1} + h_2 z^{-2} + \dots, |z| > 1 \quad (5.16)$$

เสถียรภาพของสมการที่ (5.16) หาได้จากการพิจารณาที่ L_2 norm ดังนี้คือ

$$\sum_{m=0}^{\infty} h^2(m) < \infty \quad (5.17)$$

จากความสัมพันธ์ของ Parseval ด้วยกับความจริงที่ว่า

$$\sum_{m=0}^{\infty} h^2(m) = \frac{1}{2\pi} \int_{-\pi}^{\pi} |H(e^{j\omega})|^2 d\omega = 1 \quad (5.18)$$

ซึ่งจะเป็นจริงถ้าผลตอบสนองทางความถี่เป็นดังสมการที่ (5.12) พิจารณาฟังก์ชันต่อไปนี้

$$H_0(z) = \exp\left(\frac{z^{-1} + 1}{z^{-1} - 1}\right) \quad (5.19)$$

สำหรับ $|z| > 1$ ฟังก์ชัน $(z^{-1} + 1)/(z^{-1} - 1)$ จะมีส่วนจริงเป็นลบ ทำให้ $H_0(z)$ มีขอบเขต (bound) ดังนั้นอนุกรมตามสมการที่ (5.16) จะลู่เข้าเฉพาะกรณีที่ $|z| > 1$

ผลตอบสนองทางความถี่ของสมการที่ (5.19) หาได้โดยการกำหนดให้ $z = \rho e^{j\omega}$ โดยที่ $0 < \rho < 1$ จากนั้นทำการ take ลิมิต ให้ $\rho \rightarrow 1$ ดังนั้น กรณีที่ $\omega \neq 0$ พบว่า

$$|H_0(e^{j\omega})| = \left| \lim_{\rho \rightarrow 1} H_0(\rho e^{j\omega}) \right| = 1, \omega \neq 0 \quad (5.20)$$

เพราะว่า $(\rho e^{-j\omega} + 1)/(\rho e^{-j\omega} - 1)$ จะเข้าสู่ค่าจินตภาพ เมื่อ $\rho \rightarrow 1$ ถ้า $\omega \neq 0$ ซึ่งค่า exponential ของส่วนจินตภาพนี้ จะมีค่า modulus เป็น 1

แต่ในกรณีที่ $\omega = 0$ พบว่า

$$\lim_{\rho \rightarrow 1} \frac{\rho + 1}{\rho - 1} \rightarrow -\infty$$

ดังนั้น

$$H_0(e^{j0}) = \lim_{\rho \rightarrow 1} H(\rho) = \exp(-\infty) = 0 \quad (5.21)$$

ดังนั้นผลตอบสนองความถี่ เมื่อทำการ take ลิมิตที่ค่า $\rho \rightarrow 1$ จะได้ว่า

$$|H_0(e^{j\omega})| = \begin{cases} 0, \omega = 0; \\ 1, \text{otherwise} \end{cases} \quad (5.22)$$

ทำการแปลง $z \rightarrow e^{-j\omega_0} z$ และทำการเลื่อน ω_0 ไปตามรัศมีวงกลมในระนาบ z ดังนั้น ฟังก์ชัน $H_0(e^{-j\omega_0} z)$ มี modulus เป็น 1 เกือบทุกจุดบนวงกลมหนึ่งหน่วย ยกเว้นที่ $z = e^{j\omega_0}$ จะมีค่าเท่ากับ 0 ดังสมการ คือ

$$\begin{aligned} H(z) &= H_0(e^{-j\omega_0} z) H_0(e^{j\omega_0} z) \\ &= \exp\left(\frac{e^{-j\omega_0} z^{-1} + 1}{e^{-j\omega_0} z^{-1} - 1} + \frac{e^{j\omega_0} z^{-1} + 1}{e^{j\omega_0} z^{-1} - 1}\right) \\ &= \exp\left(\frac{2(z^{-2} - 1)}{1 - 2\cos(\omega_0)z^{-1} + z^{-2}}\right) \end{aligned} \quad (5.23)$$

ทำการวิเคราะห์เฉพาะที่ $|z| > 1$ จะได้ผลตอบสนองทางความถี่ของสมการที่ (5.23) หลังจากทำการ take ลิมิตให้ $\rho \rightarrow 1$ คือ

$$|H_0(e^{j\omega})| = \left| \lim_{\rho \rightarrow 1} H(\rho e^{-j\omega}) \right| = \begin{cases} 0, \omega = \pm\omega_0; \\ 1, \text{otherwise} \end{cases} \quad (5.24)$$

จะพบว่า $H(z)$ มีรูปแบบที่ไม่เป็นฟังก์ชันแบบเศษส่วนของตัวแปร z แต่มีความต้องการที่จะสร้างตัวกรองให้อยู่ในรูปของฟังก์ชันแบบเศษส่วน ซึ่งสามารถทำได้โดยใช้การประมาณแบบ L_2 norm พิจารณาฟังก์ชันต่อไปนี้ คือ

$$\hat{H}(z) = \frac{N(z)}{D(z)} \quad (5.25)$$

เมื่อตัวเศษ $N(z)$ กำหนดให้เป็นโพลีโนเมียลในตัวแปร z อันดับสอง มีซีโรที่ $z = e^{\pm j\omega_0}$ และ $D(z)$ มีโพลอยู่ใกล้กับ $z = e^{\pm j\omega_0}$ ซึ่งวิธีการนี้ จะทำให้มี modulus ประมาณ 1 ยกเว้นในบริเวณที่ $z = e^{\pm j\omega_0}$

5.2 การประมาณค่าตัวกรองแบบนอตร์

หัวข้อนี้จะกล่าวถึงการประมาณค่าตัวกรองสัญญาณแบบนอตร์ ซึ่งใช้ในการประมาณสัญญาณคลื่นชาวน์ที่ไม่ทราบความถี่ซึ่งปะปนอยู่กับสัญญาณรบกวน การประมาณตัวกรองแบบนอตร์ จากผลตอบสนองทางอุดมคติ ให้อยู่ในรูปของฟังก์ชันแบบเศษส่วน (Rational function) ที่มีความสัมพันธ์กับโครงสร้างของตัวกรองดิจิทัลแบบ Direct form [16-18] โดยจะสมมติให้สัญญาณอินพุต เป็นสัญญาณชาวน์ความถี่เดียวปนมากับสัญญาณรบกวน ส่วนตัวกรองแบบนอตร์ที่ใช้โครงสร้างแบบ Direct form ทำการประมาณค่าตัวกรองสัญญาณแบบนอตร์ แบบ Direct form ด้วยสมการต่อไปนี้ คือ

$$\hat{H}(z) = \frac{N(z)}{N(\rho z)} \quad (5.26)$$

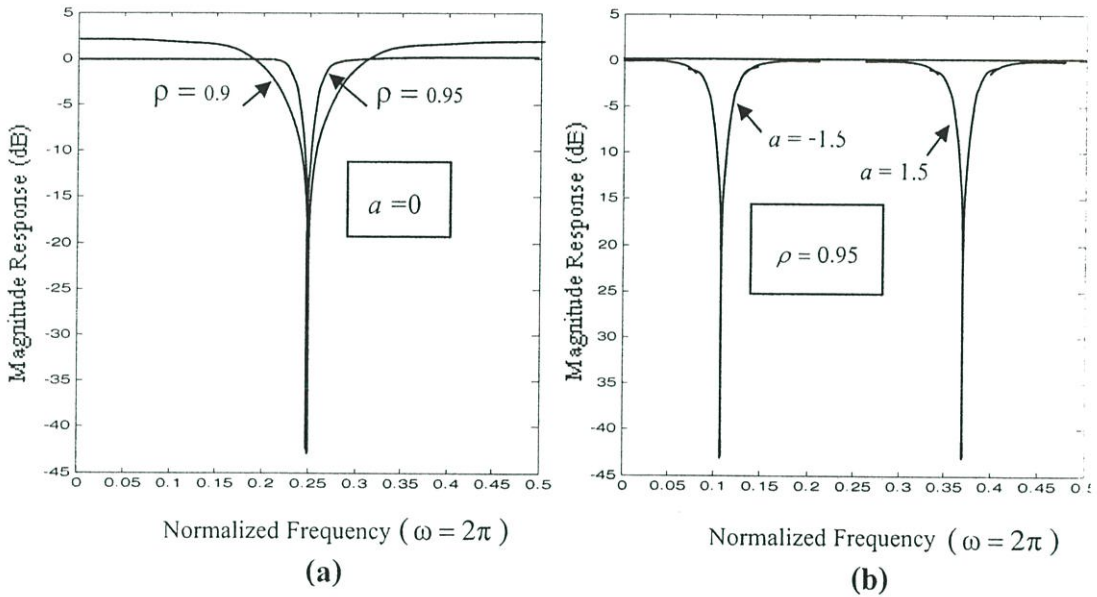
เมื่อ $N(z)$ เป็นโพลีโนเมียลที่มีซีโรอยู่บนวงกลมหนึ่งหน่วย ในขณะที่ ρ เป็นค่าคงที่ มีค่าใกล้เคียงแต่น้อยกว่า 1 ดังนั้น โพลของตัวกรองจะยังคงอยู่ในบริเวณที่มีความเสถียรภาพ การออกแบบตัวกรองความถี่ด้วยฟังก์ชันแบบเศษส่วนนั้น จะนิยมสร้างจาก ตัวกรองความถี่อันดับสองถ้าต้องการจำนวนอันดับที่สูงขึ้น จะนำเอาตัวกรองอันดับสองหลาย ๆ ตัวมาต่อ Cascade กัน ดังนั้น จะได้ ฟังก์ชันถ่ายโอน (Transfer function) ของตัวกรองแบบนอตร์อันดับสองดังสมการ [8] คือ

$$\hat{H}(z) = \frac{1 + az^{-1} + z^{-2}}{1 + \rho az^{-1} + \rho^2 z^{-2}} \quad (5.27)$$

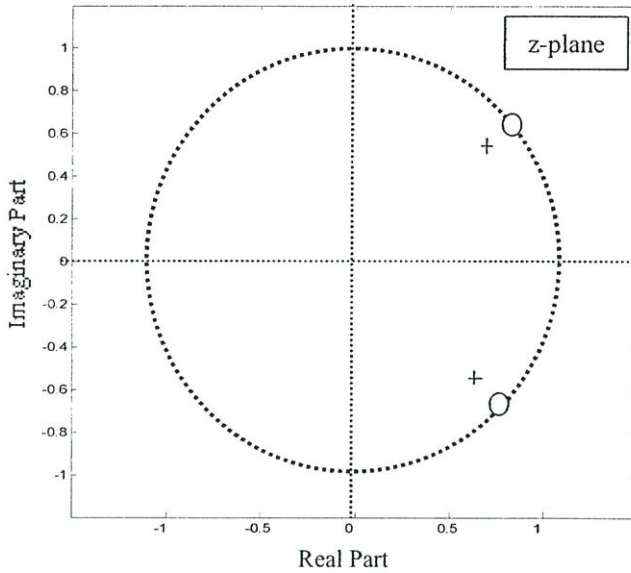
เมื่อ ρ เป็นพารามิเตอร์สำหรับกำหนดแบนด์วิดท์ของตัวกรอง ($0 < \rho < 1$) ความถี่นอตช์จะกำหนดได้ตามสมการ คือ

$$\omega_0 = \cos^{-1}(-a/2) \quad (5.28)$$

จะได้ว่า $-2 < a < 2$ รูปที่ 5.1 แสดงผลตอบสนองทางขนาดต่อความถี่ $|\hat{H}(e^{j\omega})|$ โดยการปรับเปลี่ยนพารามิเตอร์ a และ ρ จากรูปแสดงให้เห็นว่าเมื่อ ρ เข้าใกล้ 1 จะทำให้แบนด์วิดท์ของตัวกรองลดลง หรือค่า Q-factor สูงขึ้น ทำให้ $|\hat{H}(e^{j\omega})|$ มีความใกล้เคียงกับผลตอบสนองทางความถี่ในอุดมคติ a เป็นพารามิเตอร์ที่ใช้กำหนดความถี่นอตช์ตามสมการที่ (5.28) และรูปที่ 5.2 แสดงตำแหน่ง โพล-ซีโร ของสมการที่ (5.27) เมื่อกำหนดให้ $\rho = 0.95$ และ $a = -1.5$



รูปที่ 5.1 ผลตอบสนองทางขนาดต่อความถี่ $|\hat{H}(e^{j\omega})|$ เมื่อเปลี่ยนค่าสัมประสิทธิ์ ρ และ a



รูปที่ 5.2 ตำแหน่ง โพล-ซีโร ของ $\hat{H}(z)$ เมื่อ $\rho=0.95$ และ $a=-1.5$

5.3 Least Mean P-Power Error Criterion (LMP)

ในหัวข้อนี้ได้พิจารณาหา Cost ฟังก์ชัน $J(a)$ ของ ANF อันดับสอง ซึ่ง Cost ฟังก์ชันที่ได้ จะเป็นฟังก์ชันของพารามิเตอร์ a และ a เป็นพารามิเตอร์ที่ใช้กำหนดความถี่นอตช์ของตัวกรอง คำตอบของ a ที่ต้องการ คือ ค่า a ที่ทำให้ Cost ฟังก์ชัน $J(a)$ มีค่าต่ำสุด (Minimized) นั่นคือ จะทำการ Minimize ค่าคาดหวังทางสถิติของสัญญาณเอาต์พุตยกกำลังสอง $E\{\hat{y}^2(n)\}$ ของตัวกรอง นอตช์ ซึ่งข้อตัดสิน (Criterion) ดังกล่าวนี้นี้ จะเป็นที่รู้จักกันโดยทั่วไป นั่นคือ Least Mean Square (LMS) error sense เหตุผลที่ LMS เป็นที่นิยมใช้กันเนื่องจากความง่ายในทางคณิตศาสตร์

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) จะดีกว่า โดยเฉพาะในกรณีที่สัญญาณรบกวนเป็นแบบอิมพัลส์ (Impulse noise) [21] เนื่องจากการ Minimize แบบ L_p มีทฤษฎีรองรับอย่างสมบูรณ์แล้ว จึงเป็นสิ่งที่น่าสนใจ หากนำมาใช้พัฒนาเป็นอะแดปทีฟ อัลกอริทึมสำหรับ ANF ซึ่งโดยปกติใช้ข้อตัดสินแบบ LMS แต่จะใช้ข้อตัดสินแบบ Least Mean p-Power (LMP) Error Criterion แทน [3] ซึ่งพบว่าถ้าเลือกค่า p ที่เหมาะสมแล้ว เมื่อนำมาประยุกต์ใช้ในทางปฏิบัติหลาย ๆ อย่าง อะแดปทีฟอัลกอริทึมแบบ LMP จะมี สมรรถนะ ที่ดีกว่า อะแดปทีฟอัลกอริทึมแบบ LMS

เพื่อจะอธิบายข้อตัดสินใจแบบ LMP ให้สัญญาณอินพุตเป็นสัญญาณไซน์ความถี่เดียว ปนมากับสัญญาณรบกวนแบบขาว และ/หรือ สัญญาณรบกวนแบบอิมพัลส์ คือ

$$x(n) = A \sin(\omega_1 n + \theta) + v(n) \quad (5.29)$$

เมื่อ $v(n)$ คือ สัญญาณรบกวนแบบขาว หรือ อิมพัลส์ จากฟังก์ชันถ่ายโอนของตัวกรองอะแดปทีฟแบบนอกรีตอันดับสอง พารามิเตอร์ a จะถูกปรับ ดังนั้นค่า Mean p-Power error ของสัญญาณเอาต์พุตหรือ Cost ฟังก์ชัน จะถูกปรับจนมีค่าต่ำสุด

$$J(a) = E\{|\hat{y}(n)|^p\} \quad (5.30)$$

สมการที่ (5.30) จะกลายเป็น LMS เมื่อให้ $p=2$ การหาค่าของสมการที่ (5.30) ทำได้โดยใช้การประมาณด้วยวิธีการของ Batch (Batch method) [23] คือ

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

$$\hat{J}(a) = \frac{1}{N} \sum_{n=1}^N |\hat{y}(n)|^p \quad (5.31)$$

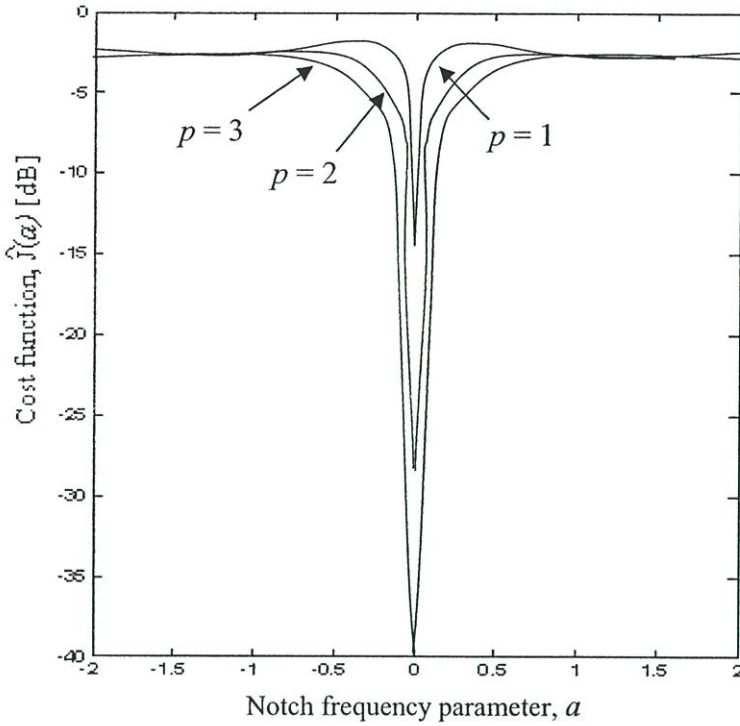
ดังนั้น จะสามารถประมาณค่าตอบของพารามิเตอร์ a ได้ตามสมการ คือ

$$\hat{a} = \arg \min_a \hat{J}(a) \quad (5.32)$$

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

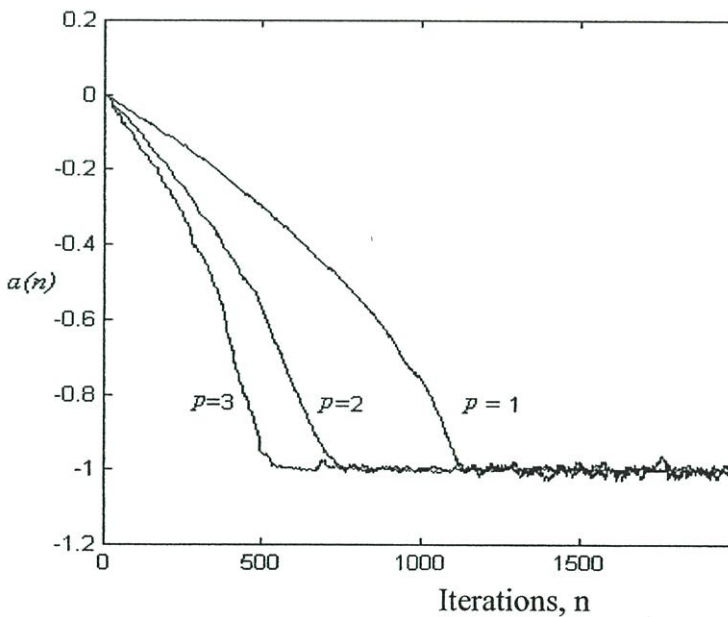
$$x(n) = \sin\left(\frac{\pi}{2}n + \frac{\pi}{2}\right) + v(n), n = 0, 1, \dots, N \quad (5.33)$$

กำหนดให้ $v(n)$ เป็นสัญญาณรบกวนขาว มี Pdf เป็นแบบเกาส์เซียน มีค่า $\sigma_v^2 = 0.5$ ป้อนเข้าสู่ตัวกรองแบบนอกรีต โดยกำหนดให้ $\rho = 0.9$ รูปที่ 5.3 แสดงค่า $\hat{J}(a)$ เทียบกับ a ที่ค่า $p = 1, 2$ และ 3



รูปที่ 5.3 ค่า Mean p -Power error $J(a)$ สำหรับ $p = 1, 2$ และ 3

จากรูปที่ 5.3 จะเห็นได้ชัดว่า ที่ $p = 3$ จะให้ค่า $J(a)$ ต่ำสุด คือ -40 dB และ Dip shape จะกว้างกว่าที่ค่า $p = 2$ และ $p = 1$ นั้นหมายความว่า เมื่อนำ Steepest decent อัลกอริทึมมาใช้เพื่อหาจุดต่ำของ $J(a)$ ที่ p มีค่าสูง จะทำให้หาค่าที่พอลกอริทึมทำงานได้เร็วเพิ่มขึ้น ดังแสดงในรูปที่ 5.4



รูปที่ 5.4 การลู่เข้าของพารามิเตอร์ a เมื่อใช้ Steepest decent อัลกอริทึม ที่ $p = 1, 2$ และ 3

5.4 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] ซึ่งมีรูปแบบสมการดังนี้ คือ

$$a(n+1) = a(n) - \mu \frac{\partial J(a)}{\partial a} \quad (5.34)$$

เมื่อ μ คือ ค่าสเกลปไซส์ ใช้กำหนดอัตราการเร็วของอะแดปทีฟอัลกอริทึม ขอบเขตของ μ สำหรับตัวกรองอะแดปทีฟแบบ IIR ยังไม่มีข้อสรุปที่แน่นอน ยังอยู่ในขั้นการค้นคว้าวิจัย ทั้งนี้ เพราะความไม่เป็นเชิงเส้นของ Cost ฟังก์ชัน ซึ่งจะไม่สามารถทำนายหรือทราบล่วงหน้าได้เลย จึงเป็นการยากที่จะสร้างเป็นสูตรสำเร็จเหมือนอย่างกับตัวกรองอะแดปทีฟแบบ FIR [25] ดังนั้น จึงต้องใช้วิธีการ ลองผิดลองถูก (Trial and error) และ ปรับเปลี่ยนไปจนกว่าจะพอใจ

ในทางปฏิบัติ จะไม่สามารถสร้างอะแดปทีฟอัลกอริทึมตามสมการที่ (5.34) ได้ ทั้งนี้เพราะ Cost ฟังก์ชัน $J(a) = \{|\hat{y}(n)|^p\}$ เป็นค่าคาดหวังทางสถิติ ซึ่งจะต้องคิดค่า $n = 0, 1, 2, \dots, \infty$ และ เนื่องจาก 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

$$a(n+1) = a(n) - \mu \frac{\partial |\hat{y}(n)|^p}{\partial a} \quad (5.35)$$

ในกรณีของสมการที่ (5.35) p มีโอกาสเป็นเลขคี่ ซึ่งถ้าช่วงเวลาใดที่ $\hat{y}(n)$ มีค่าเป็นลบ และ p เป็นเลขคี่ก็ทำให้ทิศทางของเกรเดียนต์ เป็นบวก - ลบ สลับกัน ขึ้นกับสัญญาณ $\hat{y}(n)$ จะทำให้อะแดปทีฟอัลกอริทึม ไม่สามารถค้นหาจุดต่ำสุดของ cost ฟังก์ชันได้เลย ดังนั้น จึงต้องใช้ตัวกระทำ $|\cdot|$ เพื่อบังคับให้ $\hat{y}^p(n)$ มีค่าเป็นบวก สำหรับทุกค่าของ p เพื่อทำให้ทิศทางของเกรเดียนต์ เป็นลบตลอดการทำงาน ดังนั้นจะได้ว่า

$$|\hat{y}(n)|^p = \begin{cases} [\hat{y}(n)]^p & p : \text{even} \\ \text{sgn}[\hat{y}(n)] \cdot [\hat{y}(n)]^p & p : \text{odd} \end{cases} \quad (5.36)$$

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

$$\frac{\partial |\hat{y}(n)|^p}{\partial a} = \begin{cases} p \cdot [\hat{y}(n)]^{p-1} \cdot \frac{\partial \hat{y}(n)}{\partial a} & p : \text{even} \\ p \cdot \text{sgn}[\hat{y}(n)] \cdot [\hat{y}(n)]^{p-1} \cdot \frac{\partial \hat{y}(n)}{\partial a} & p : \text{odd} \end{cases} \quad (5.37)$$

โดยที่ $\text{sgn}[x] \equiv \frac{x}{|x|}$ ดังนั้นสมการที่ (5.37) จะกลายเป็น

$$\begin{aligned} a(n+1) &= a(n) - \mu \cdot U_1(n) & p : \text{even} \\ a(n+1) &= a(n) - \mu \cdot U_2(n) & p : \text{odd} \end{aligned} \quad (5.38)$$

เมื่อ

$$\begin{aligned} U_1(n) &= p \cdot [\hat{y}(n)]^{p-1} \cdot \hat{e}(n) \\ U_2(n) &= p \cdot \text{sgn}[\hat{y}(n)] \cdot [\hat{y}(n)]^{p-1} \cdot \hat{e}(n) \end{aligned} \quad (5.39)$$

และ

$$\hat{e}(n) = \frac{\partial \hat{y}(n)}{\partial a} \quad (5.40)$$

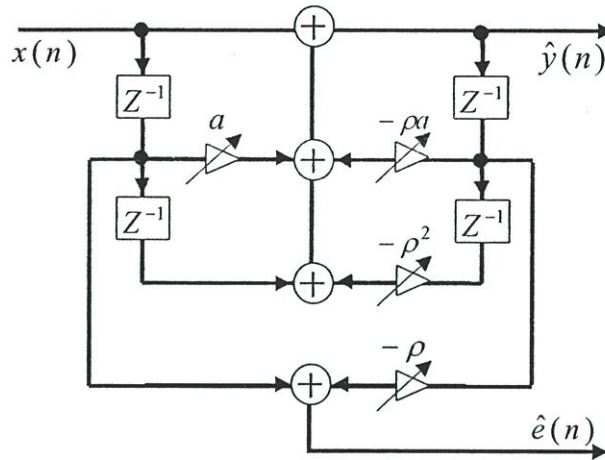
สมการของ $\hat{y}(n)$ หาได้โดยทำการแปลง Z กลับ (Inverse Z - transform) ของสมการที่ (5.27) ได้ดังนี้

$$\hat{y}(n) = x(n) + ax(n-1) + x(n-2) + a\rho y(n-1) - \rho^2 y(n-2) \quad (5.41)$$

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

$$\hat{e}(n) = \frac{\partial y(n)}{\partial a} \approx x(n-1) - \rho y(n-1) \quad (5.42)$$

จากที่กล่าวมานี้ คือ ะแดปทีฟอัลกอริทึมที่นำเสนอในบทวิจัย [3] รูปที่ 5.5 แสดงโครงสร้างแบบ Direct form ของ ANF



รูปที่ 5.5 โครงสร้างแบบ Direct form ของตัวกรอง ANF

5.5 ะแดปทีฟอัลกอริทึมที่เกี่ยวข้อง

วิทยานิพนธ์ [35] นำเสนออะแดปทีฟอัลกอริทึม ที่ได้ทำการดัดแปลงจาก LMP คือ QLMP (Quantized LMP) [27-28] และ VSQ LMP (Variable Step-size Quantized LMP) [29] อะแดปทีฟอัลกอริทึม แบบต่าง ๆ จะมีข้อดี ข้อเสีย แตกต่างกันไป ขึ้นกับงานที่จะไปใช้ อย่างไรก็ตาม การจะเลือกว่าจะนำอะแดปทีฟอัลกอริทึมแบบใดมาใช้ นั้น จะขึ้นกับเหตุผล 4 ประการ คือ

- i) ความเร็วในการหาคำตอบ
- ii) ความถูกต้องของคำตอบ (ดูจากค่า MSE มีค่าต่ำ)
- iii) ความสามารถในการติดตามสัญญาณสุ่มแบบ Nonstationary
- iv) ความซับซ้อนในการคำนวณ

ดังนั้นหาก อะแดปทีฟอัลกอริทึมแบบใด ที่ให้คุณสมบัติทั้ง 4 ข้อนี้ ก็จะเลือกนำมาใช้งาน

5.5.1 อะแดปทีฟอัลกอริทึมแบบ QLMP

สมการสำหรับปรับพารามิเตอร์ a ของอะแดปทีฟอัลกอริทึมแบบ QLMP ที่ได้ดัดแปลงจาก LMP คือ

$$\begin{aligned} a(n+1) &= a(n) - \mu \cdot \text{sgn}[U_1(n)] & p: \text{even} \\ a(n+1) &= a(n) - \mu \cdot \text{sgn}[U_2(n)] & p: \text{odd} \end{aligned} \quad (5.43)$$

โดยที่ $U_i(n), i=1,2$ มีค่าตามสมการที่ (5.39) วิธีการนี้ อาศัยตัวกระทำ $\text{sgn}[\cdot]$ สำหรับตรวจสอบเครื่องหมายของ $U_i(n)$ ถ้า $U_i(n)$ มีค่าเป็นบวก จะได้ค่า $\text{sgn}[U_i(n)] = 1$ และถ้า $U_i(n)$ มีค่าเป็นลบ จะได้ค่า $\text{sgn}[U_i(n)] = -1$ 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

$$U_i(n) = U_2(n) = \text{sgn}[\hat{y}(n)] \cdot \hat{e}(n) \quad (5.44)$$

และสมการสำหรับปรับพารามิเตอร์ a คือ

$$a(n+1) = a(n) - \mu \cdot \text{sgn}[\text{sgn}[\hat{y}(n)] \cdot \hat{e}(n)] \quad (5.45)$$

ซึ่งจะลดการคูณลงได้ 2 ครั้ง (เทียบกับ LMP ที่ $p \neq 1$) นอกจากนี้ ยังพบว่า ความเร็วในการทำงานของอะแดปทีฟอัลกอริทึมจะขึ้นกับค่า 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 สัญญาณรบกวนเกรเดียนต์ [24-25] สูงกว่าแบบ LMP เมื่อค่า μ เท่ากัน นั้นหมายความว่า variance ของ พารามิเตอร์ a มีค่าสูงกว่านั่นเอง

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

5.5.2 อะแดปทีฟอัลกอริทึมแบบ VSQMLP [29]

อะแดปทีฟอัลกอริทึมแบบ VSQMLP นี้ได้นำข้อดีของการทำให้เสถียรไซส์ เปลี่ยนแปลงตามเวลา คือ จะทำงานได้เร็วขึ้น โดยจะสร้างสมการสำหรับปรับเสถียรไซส์ แบบรีเคอร์ซีฟโดยตรง ไม่ต้องใช้การหารการทำเช่นนี้สามารถลดสัญญาณรบกวนเกรเดียนต์ ได้ เมื่อเทียบกับ QLMP นั้นหมายความว่าค่า Variance ของคำตอบมีค่าต่ำกว่าด้วย จากสมการที่ (5.43) ที่แทน $\mu \rightarrow \mu(n)$ ดังนั้นสมการที่ใช้สำหรับปรับพารามิเตอร์ a คือ

$$\begin{aligned} a(n+1) &= a(n) - \mu(n) \cdot \text{sgn}[U_1(n)] & p: \text{even} \\ a(n+1) &= a(n) - \mu(n) \cdot \text{sgn}[U_2(n)] & p: \text{odd} \end{aligned} \quad (5.46)$$

เมื่อ

$$\mu(n+1) = \gamma \cdot \mu(n) + \lambda \psi^2(n) \quad (5.47)$$

โดยที่

$$\psi(n) = \alpha \cdot \psi(n-1) + (1-\alpha)y^2(n) \quad (5.48)$$

γ, λ เป็นค่าคงที่ที่อยู่ในช่วง $[0,1]$ ส่วน α ประมาณ 1 วิธีการนี้จะใช้ค่าพลังงานของสัญญาณเอาต์พุต $\hat{y}(n)$ มาใช้สำหรับควบคุมค่าเสถียรไซส์ ให้เปลี่ยนแปลงตามเวลา

จากนั้นได้มีการนำแนวความคิดอัลกอริทึมแบบ VSQ LMP มาใช้กับตัวกรองอะแดปทีฟแบบ IIR โดยอยู่บนพื้นฐานของอะแดปทีฟอัลกอริทึมแบบ LMP และเนื่องจากค่า เสถียรไซส์ จะปรับตัวตามค่าพลังงานของสัญญาณเอาต์พุต $\hat{y}(n)$ ในช่วงแรก ขณะที่ ANF เริ่มทำงาน ยังไม่สามารถตรวจวัดความถี่ได้ ค่าพลังงานของ $\hat{y}(n)$ มีค่าสูง เป็นผลให้ $\mu(n)$ มีค่าสูง จึงทำงานได้เร็ว หลังจากเข้าสู่แล้ว ค่าพลังงาน $\hat{y}(n)$ มีค่าต่ำโดยเหลือเฉพาะพลังงานของสัญญาณรบกวนเท่านั้น ดังนั้นเมื่อพลังงานของสัญญาณรบกวนมีค่าสูงขึ้นทันทีทันใด หรือ เกิดสัญญาณอิมพัลส์ จะทำให้ค่าสัมประสิทธิ์ของฟิลเตอร์เกิดความแปรปรวนสูงมาก และอาจทำให้ไม่สามารถเข้าสู่ค่าตอบได้อีก

5.5.3 อะแดปทีฟอัลกอริทึมแบบปรับเสถียรไซส์ [35]

วิทยานิพนธ์ [35] ได้ใช้เกรเดียนต์ของเอาต์พุตเข้ามาช่วยเพื่อแก้ปัญหาในกรณีที่เกิดสัญญาณรบกวนแบบอิมพัลส์ จากสมการที่ (5.48) จะได้ว่า

$$\psi(n) = \alpha \cdot \psi(n-1) + (1-\alpha)y(n) \cdot e(n) \quad (5.49)$$

จากสมการที่ (5.49) ทำให้ได้อะแดปทีฟอัลกอริทึมแบบปรับเสถียรไซส์สำหรับตัวกรองสัญญาณ ที่สามารถทนต่อสัญญาณอิมพัลส์ได้ดีขึ้น

5.5.4 อะแดปทีฟอัลกอริทึมที่เกิดจากการทำงานร่วมกันของอัลกอริทึมแบบเจนเนติกกับอัลกอริทึมแบบปรับเสถียรไซส์[35]

ในหัวข้อนี้ได้นำเสนอ อะแดปทีฟอัลกอริทึมที่เกิดจากการทำงานร่วมกันของอัลกอริทึมแบบปรับเสถียรไซส์กับอัลกอริทึมแบบเจนเนติก ซึ่งจะนำไปหาค่าเสถียรไซส์ที่เหมาะสม โดยในช่วงแรกใช้ อัลกอริทึมแบบเจนเนติก เพียงอย่างเดียวในการหาเสถียรไซส์ ซึ่งจะแสดงให้เห็นถึงผลของการ

ปรับค่าของความน่าจะเป็นของมิวเดชัน (Pm) และ ความน่าจะเป็นของครอสโอเวอร์ (Pc) ที่จะส่งผลต่อค่าสัมประสิทธิ์ของฟิลเตอร์เพียงใด

1. ฟังก์ชันเป้าหมายและฟังก์ชันความเหมาะสม

ในปัญหานี้ได้ใช้ฟังก์ชันเป้าหมายเป็นอפקทฟังก์ชันของอัลกอริทึมแบบปรับเสถียรไซส์ คือ

$$a(n+1) = a(n) - \mu(n) \cdot \text{sgn}[\text{sgn}(y(n)) \cdot e(n)] \quad (5.50)$$

โดยพารามิเตอร์ต่าง ๆ มีค่าดังนี้

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

ส่วนฟังก์ชันความเหมาะสมคือ

$$e_j^2 = \frac{1}{t_e} \sum_{n=1}^{t_e} [d(n) - y_j(n)]^2 \quad (5.51)$$

t_e คือ Window Size ใช้ค่าเท่ากับ 4

$d(n)$ คือ สัญญาณเอาต์พุตที่ต้องการ 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} คือค่าบิตต่ำสุด หรือ LSB (Least significant bit) ส่วน B_1 คือ ค่าบิตสูงสุด หรือ MSB (Most significant bit)

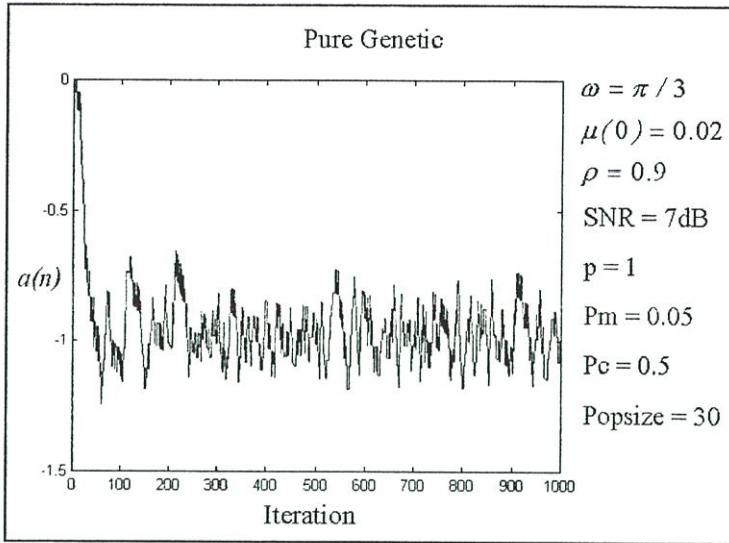
3. การดำเนินการทางพันธุศาสตร์

3.1 จำนวนประชากร 30

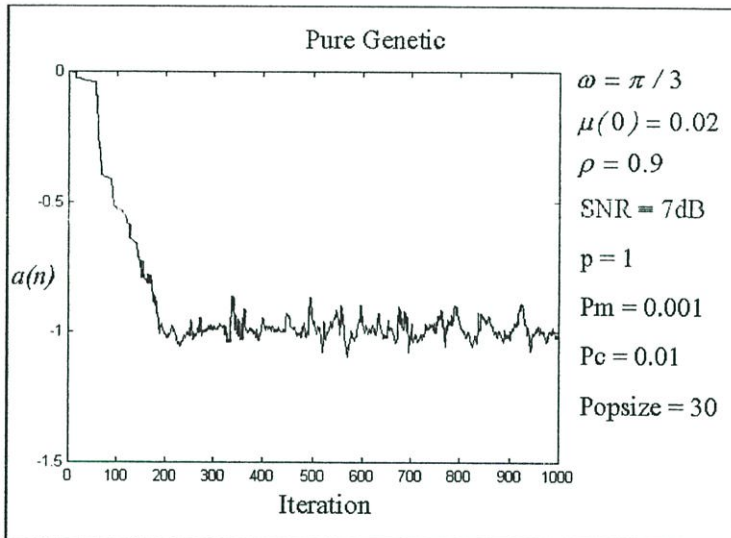
3.2 ความน่าจะเป็นของครอสโอเวอร์ $Pc = 0.5, 0.01$

3.3 ความน่าจะเป็นของมิวเดชัน $Pm = 0.05, 0.001$

3.4 รีโพรดักชัน 1 โครโมโซม



รูปที่ 5.6 สัมประสิทธิ์ฟิลเตอร์ $a(n)$ โดยใช้ อัลกอริทึมแบบเจเนติก กำหนดให้ $P_m = 0.05$ และ $P_c = 0.5$

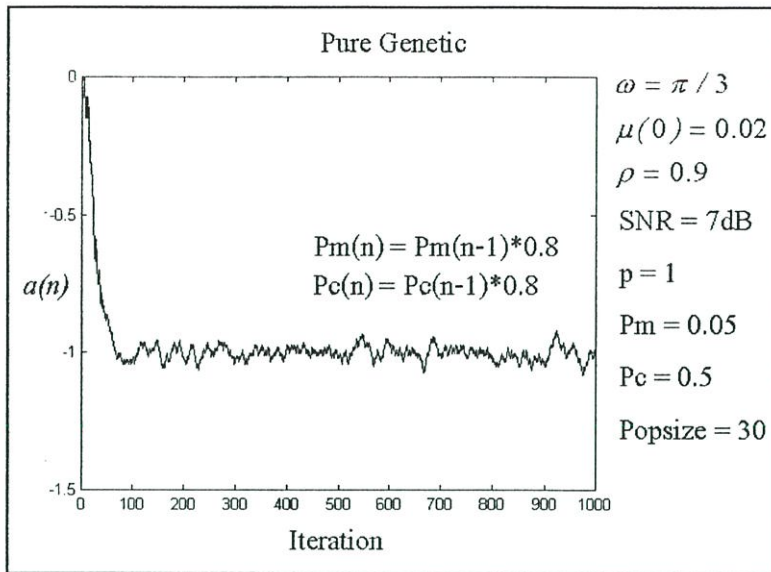


รูปที่ 5.7 สัมประสิทธิ์ฟิลเตอร์ $a(n)$ โดยใช้ อัลกอริทึมแบบเจเนติก กำหนดให้ $P_m = 0.001$ และ $P_c = 0.01$

จากรูปที่ 5.6 และ 5.7 แสดงให้เห็นว่า กรณีที่ค่า P_m และ P_c มีค่าสูงดังรูปที่ 5.6 นั้นทำให้ค่าสัมประสิทธิ์ของฟิลเตอร์เข้าสู่ค่าตอบได้เร็ว แต่มีค่าความแปรปรวนสูง ส่วนกรณีที่ค่า P_m และ P_c มีค่าต่ำลง จะทำให้สัมประสิทธิ์ของฟิลเตอร์เข้าสู่ค่าตอบได้ช้าลง และมีค่าความแปรปรวนที่ต่ำลงด้วยเช่นกัน

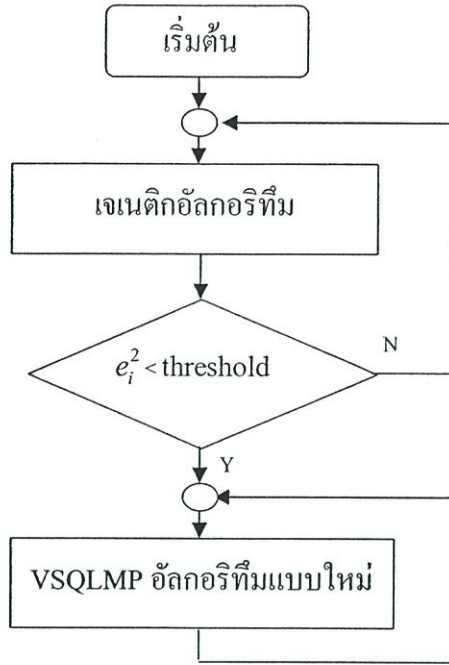
อีกกรณีหนึ่งจากผลที่ได้จากรูปที่ 5.6 และ 5.7 ทำให้เกิดสมมติฐานว่า ถ้าทำให้ค่า P_m และ P_c สูงในช่วงแรก ๆ จากนั้นค่อย ๆ ลดลงเรื่อย ๆ ผลที่เกิดขึ้นจะเป็นอย่างไร

จากรูปที่ 5.8 โดยที่พารามิเตอร์ต่าง ๆ เหมือนเดิมกำหนดให้ค่าเริ่มต้นของ $P_m = 0.05$, $P_c = 0.5$ จากนั้นทุก ๆ วัฏจักรให้ค่า P_m และ P_c ลดลงครึ่งละ 20 % ผลที่ได้ คือ ค่าสัมประสิทธิ์ของฟิลเตอร์สามารถลู่เข้าสู่ค่าตอบได้เร็วโดยที่ค่าความแปรปรวนไม่สูงมากนัก

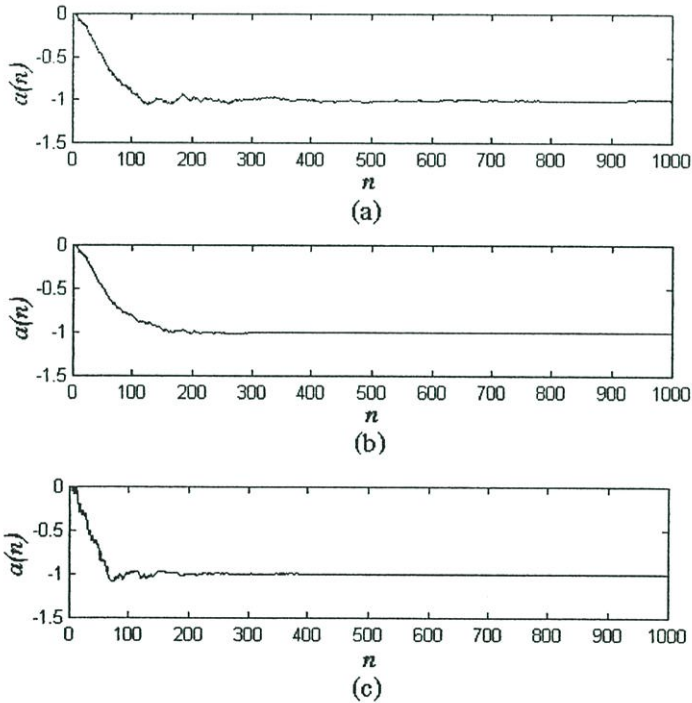


รูปที่ 5.8 สัมประสิทธิ์ของฟิลเตอร์ $a(n)$ โดยใช้ อัลกอริทึมแบบเจเนติก กำหนดให้ P_m และ P_c แปรค่าได้

จากผลที่ได้การหาค่าเสถียรไซส์ สำหรับอะแดปทีฟ IIR นอตซ์ฟิลเตอร์ ทำให้เกิดแนวคิดในการพัฒนาอะแดปทีฟอัลกอริทึมคือ อัลกอริทึมแบบเจเนติกสามารถหาค่าตอบได้เร็วในกรณีที่มีค่า P_m และ P_c สูง แต่จะทำให้ค่าความแปรปรวนสูงตามไปด้วย ดังนั้นในวิทยานิพนธ์ [35] จึงใช้ อะแดปทีฟอัลกอริทึมแบบปรับเสถียรไซส์ เข้ามาทำงานร่วมกับอัลกอริทึมแบบเจเนติก โดยมีเงื่อนไขดังรูปที่ 5.9 ผลที่ได้จะทำให้ได้ตัวกรองที่มีความเร็วในการทำงานสูงและสามารถทนต่อสัญญาณรบกวนได้ดี ซึ่งเป็นผลมาจากการทำงานของอัลกอริทึมแบบเจเนติกกับอัลกอริทึมแบบปรับเสถียรไซส์ แสดงผลดังรูปที่ 5.10



รูปที่ 5.9 เงื่อนไขในการทำงานร่วมกันของ อัลกอริทึมแบบเจเนติกกับอัลกอริทึมแบบปรับเสถียร [35]



รูปที่ 5.10 เปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ (a) VSQMLP แบบเดิม[29], (b) อัลกอริทึมแบบปรับเสถียร [35] (c) อัลกอริทึมแบบเจเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียร [35]

5.6 อะแดปทีฟอัลกอริทึมที่นำเสนอในวิทยานิพนธ์

ในวิทยานิพนธ์นี้ได้มีการศึกษาผลกระทบจากการปรับเสถียรไซส์ ที่มีผลต่ออะแดปทีฟ IIR นอตซ์ฟิลเตอร์ เพื่อพัฒนาให้ฟิลเตอร์มีประสิทธิภาพดียิ่งขึ้น โดยหลักการ คือ การนำเสนอการควบคุมขนาดของเอาต์พุตเข้ามาช่วยในอ็อปเทฟิงก์ชัน ทำให้ได้ผลดีขึ้น

5.6.1 อะแดปทีฟอัลกอริทึมที่เกิดจากการทำงานร่วมกันของอัลกอริทึมแบบเจนเนติกกับ

อัลกอริทึมแบบปรับเสถียรไซส์

จากหัวข้อ 5.5.4 หลักการ คือ การนำอัลกอริทึมแบบเจนเนติกมาช่วยหาค่าเสถียรไซส์ที่เหมาะสมในช่วงแรก เพื่อทำให้ความเร็วเข้าสู่ค่าสัมประสิทธิ์ที่ต้องการเร็วยิ่งขึ้น และหลังจากนั้นใช้ อัลกอริทึมแบบปรับเสถียรไซส์ทำงานต่อ เพื่อลดค่า MSE ของฟิลเตอร์ ทำให้ประสิทธิภาพของอะแดปทีฟ IIR นอตซ์ฟิลเตอร์ ดีขึ้นทั้งด้านความเร็วที่เข้าสู่ค่าตอบ และค่าความผิดพลาดน้อยลง ฟังก์ชันเป้าหมายของอัลกอริทึมแบบปรับเสถียรไซส์ คือ

$$a(n+1) = a(n) - \mu_{vss}(n) \cdot \text{sgn}[\text{sgn}(y(n)) \cdot e(n)] \quad (5.52)$$

ฟังก์ชันเป้าหมายของอัลกอริทึมแบบเจนเนติก คือ

$$a(n+1) = a(n) - \mu_{ga}(n) \cdot \text{sgn}[\text{sgn}(y(n)) \cdot e(n)] \quad (5.53)$$

ฟังก์ชันความเหมาะสมของอัลกอริทึมแบบเจนเนติก เป็นไปตามสมการที่ (5.51) คือ

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

โดยที่ μ_{vss} คือ ค่าเสถียรไซส์ได้จาก อัลกอริทึมแบบปรับเสถียรไซส์ และ μ_{ga} คือ ค่าเสถียรไซส์ได้จาก อัลกอริทึมแบบเจนเนติกซึ่งสามารถกำหนดได้ดังสมการ

$$\mu_{vss}(n+1) = \gamma \cdot \mu_{vss}(n) + \lambda \psi^2(n) \quad (5.54)$$

$$\mu_{ga}(n+1) = (\text{Select}[\text{chromosome}]) / 2^{16} \quad (5.55)$$

โดยที่

$$\psi(n) = \alpha.\psi(n-1) + (1-\alpha)y(n).e(n) \quad (5.56)$$

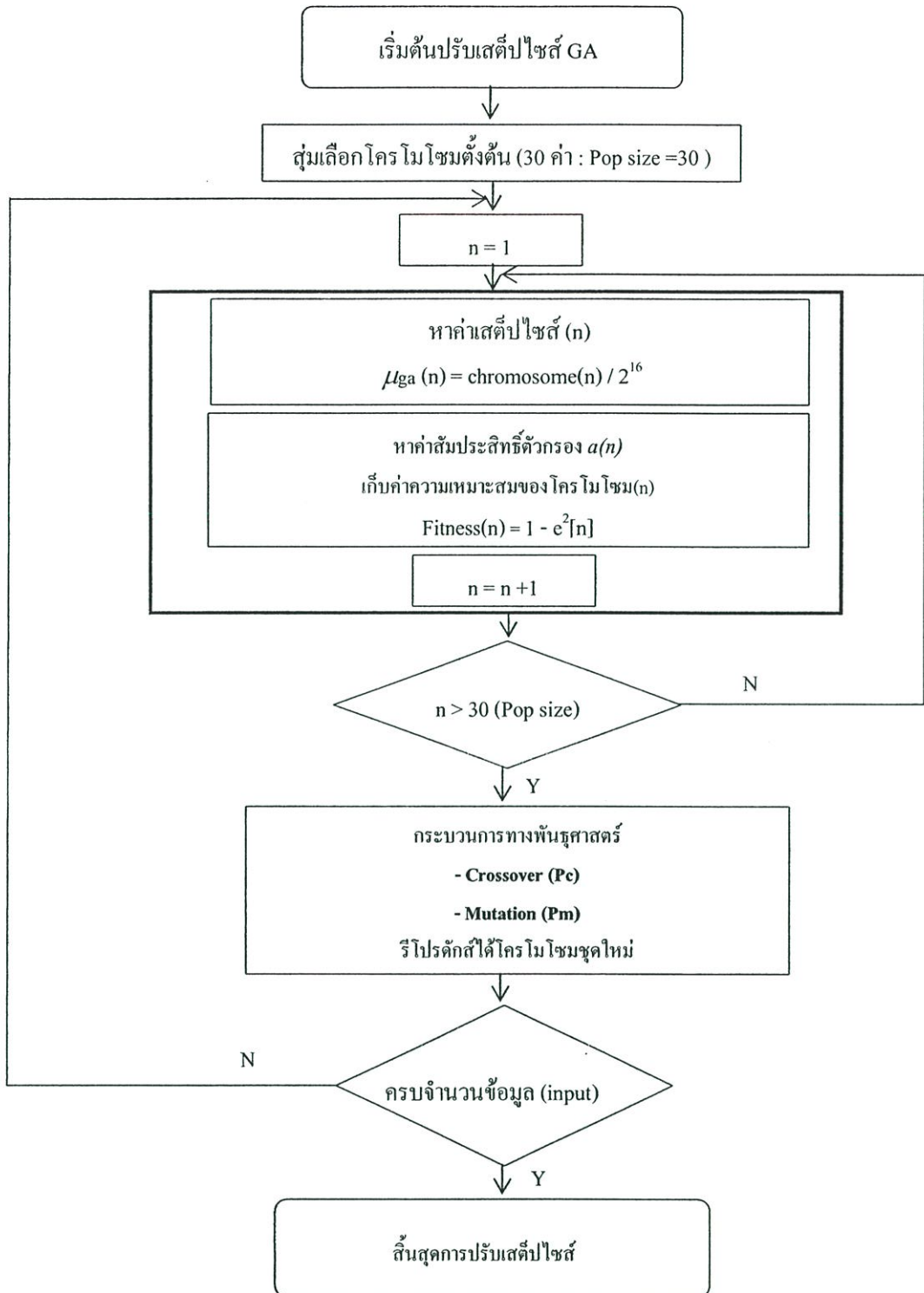
Select[chromosome] เป็นโครโมโซมขนาด 16 บิต ที่ได้จากการเลือกค่าโครโมโซมที่เหมาะสม จากชุดโครโมโซมขนาด 16 บิต ซึ่งโครโมโซม 1 ชุดจะประกอบด้วย 30 โครโมโซม (Pop size=30) ตามขบวนการทางพันธุศาสตร์ โดยกำหนดค่าความน่าจะเป็นของครอสโอเวอร์ P_c เท่ากับ 0.08 และ ความน่าจะเป็นของมิวเตชัน P_m เท่ากับ 0.02 ซึ่งวิธีการเลือกค่าโครโมโซมที่เหมาะสมได้กล่าวไว้แล้วในบทที่ 2 เนื่องจากค่าเสตีปไซส์จะมีค่าเป็นทศนิยมที่อยู่ระหว่าง 0.0 ถึง 1.0 การแปลงค่าโครโมโซมไปเป็น ค่าเสตีปไซส์ของอัลกอริทึมแบบเจเนติก จึงต้องนำโครโมโซมที่ได้หารด้วยค่าสูงสุดที่โครโมโซมจะเป็นไปได้ คือ 2^{16}

ตัวอย่างการแปลงค่าโครโมโซมเป็นเสตีปไซส์ แสดงได้ตามตารางที่ 5.1

ตารางที่ 5.1 ตัวอย่างการแปลงค่าโครโมโซมเป็นค่าเสตีปไซส์

ลำดับที่(n)	Chromosome(n)	$\mu_{ga}(n) = \text{Chromosome}(n) / 2^{16}$	$\mu_{ga}(n)$
1	51	51 / 65536	0.000778198
2	17945	17945 / 65536	0.27381897
3	27159	27159 / 65536	0.414413452
.	.	.	.
.	.	.	.
27	15811	15811 / 65536	0.241256714
28	2607	2607 / 65536	0.039779663
29	16929	16929 / 65536	0.25831604
30	15514	15514 / 65536	0.236724854

หลักการเบื้องต้นในการนำอัลกอริทึมแบบเจเนติกหาค่าเสตีปไซส์ สามารถแสดงได้ ดังรูปที่



รูปที่ 5.11 หลักการหาค่าเสถียรค่าโดยใช้อัลกอริทึมแบบเจเนติก

จากรูปที่ 5.11 แสดงหลักการทำงานเบื้องต้นของการปรับค่าเสถียรค่า ด้วยอัลกอริทึมแบบเจเนติกพอสรุปได้ดังนี้

1. สุ่มโครโมโซมจำนวน 30 ตัว (1 ชุด)

2. วนรอบหาค่าเสถียรไซส์ โดยค่าเสถียรไซส์ $\mu_{ga}(n) = \text{Chromosome}/2^{16}$ และเลือกค่าโครโมโซมที่ให้ค่าเสถียรไซส์ที่ดีที่สุด โดยดูจากความผิดพลาด $e^2[n]$ มีค่าน้อยที่สุดแล้วเก็บค่าความเหมาะสม (Fitness value) ของโครโมโซมแต่ละตัวไว้ นำค่าเสถียรไซส์ที่ดีที่สุดนั้น ไปหาค่าสัมประสิทธิ์ตัวกรอง $a(n)$ ต่อไป
3. จากนั้นทำการจัดเรียงโครโมโซมใหม่ ตามค่าความเหมาะสมในข้อ 2 ตามขบวนการทางพันธุศาสตร์ แล้วนำโครโมโซมที่จัดเรียงแล้วไปผ่านขบวนการครอสโอเวอร์ P_c และขบวนการมิวเตชัน P_m เพื่อสร้างโครโมโซมชุดใหม่ สำหรับใช้ในการหาเสถียรไซส์ต่อไป เนื่องจากกำหนดค่า P_c P_m ในขบวนการครอสโอเวอร์และมิวเตชัน จะมีผลต่อการจัดเรียงหรือคัดเลือกโครโมโซมชุดใหม่ เพื่อนำมาเป็นค่าเสถียรไซส์โดยใช้อัลกอริทึมแบบเจเนติก (μ_{ga}) การใช้งานจึงต้องทำการหาค่า P_c และ P_m ที่เหมาะสมก่อน

5.6.2 การหาค่า P_c , P_m ที่เหมาะสมให้กับอัลกอริทึมแบบเจเนติก

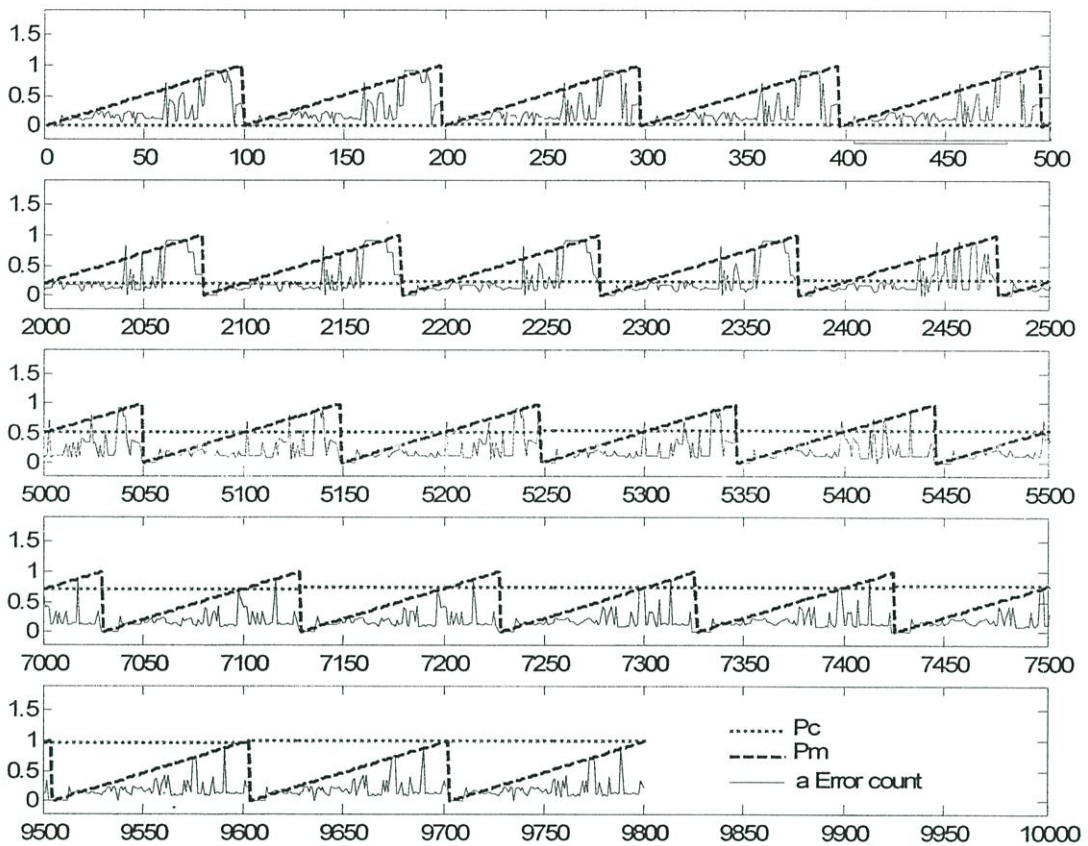
การทดสอบหาค่า P_c และ P_m ที่เหมาะสม โดยทดสอบกับอะแดปทีฟอัลกอริทึม ที่เกิดจากการทำงานร่วมกันของอัลกอริทึมแบบปรับเสถียรไซส์และอัลกอริทึมแบบเจเนติก โดยใช้สัญญาณอินพุตเดียวกัน โครโมโซมเริ่มต้นชุดเดียวกัน แต่เปลี่ยนค่า P_c และ P_m โดยใช้โปรแกรมคอมพิวเตอร์มาช่วยในการสร้างและวิเคราะห์ค่าสัมประสิทธิ์ตัวกรอง (a) ที่ได้

วิธีการทดสอบเพื่อหาค่า P_c และ P_m ที่เหมาะสม ที่ให้ค่าสัมประสิทธิ์ของตัวกรอง (a) ดีที่สุด

1. วนรอบเปลี่ยนค่า P_c และ P_m โดยให้ P_c และ P_m มีค่าตั้งแต่ 0.01 ถึง 0.99 เพื่อสร้างไฟล์สัมประสิทธิ์ตัวกรอง (a) ในแต่ละครั้งจะมีการ Reset ค่าในตาราง Random เพื่อให้ค่า Random ที่ใช้ในการหาค่าสัมประสิทธิ์ (a) เป็นค่าเดียวกันทุกครั้ง รวมถึง โครโมโซมเริ่มต้นเป็นโครโมโซม ชุดเดียวกันด้วย
2. แต่ละ P_c และ P_m จะมีการหาสัมประสิทธิ์ของตัวกรอง (a) 100 ครั้ง เพื่อหาค่าความเร็วการลู่เข้าค่าตอบ และเก็บจำนวนครั้งที่ค่าความเร็วลู่เข้าสู่ค่าตอบ (a) มีความเปลี่ยนแปลง หลักการหาค่าสัมประสิทธิ์ตัวกรอง (a) คือ
 - กำหนดให้ $a_compare = \text{average}(a[800..999])$ (ค่าเฉลี่ยของสัมประสิทธิ์ของตัวกรอง (a) ช่วงท้ายของกราฟ)
 - วนรอบเพื่อเปรียบเทียบค่าสัมประสิทธิ์ของตัวกรอง (a) กับค่า $a_compare$ เพื่อหาจุดเริ่มต้นของความเร็วลู่เข้าสู่สัมประสิทธิ์ (a) หากค่า a มีค่าต่างจาก $a_compare$ เกินค่าที่กำหนด ณ จุดใด จุดนั้นจะเป็นจุดเริ่มลู่เข้าสู่สัมประสิทธิ์ โดยวนรอบตั้งแต่ค่าสัมประสิทธิ์ของตัวกรอง (a) ที่ตำแหน่ง 200 วนย้อนกลับมาที่ค่าสัมประสิทธิ์ของตัวกรอง (a) ค่าแรก โดยจะมีการเก็บค่าความเร็วลู่เข้า และจำนวนครั้งที่ค่าสัมประสิทธิ์ของตัวกรอง (a) มีความเปลี่ยนแปลง

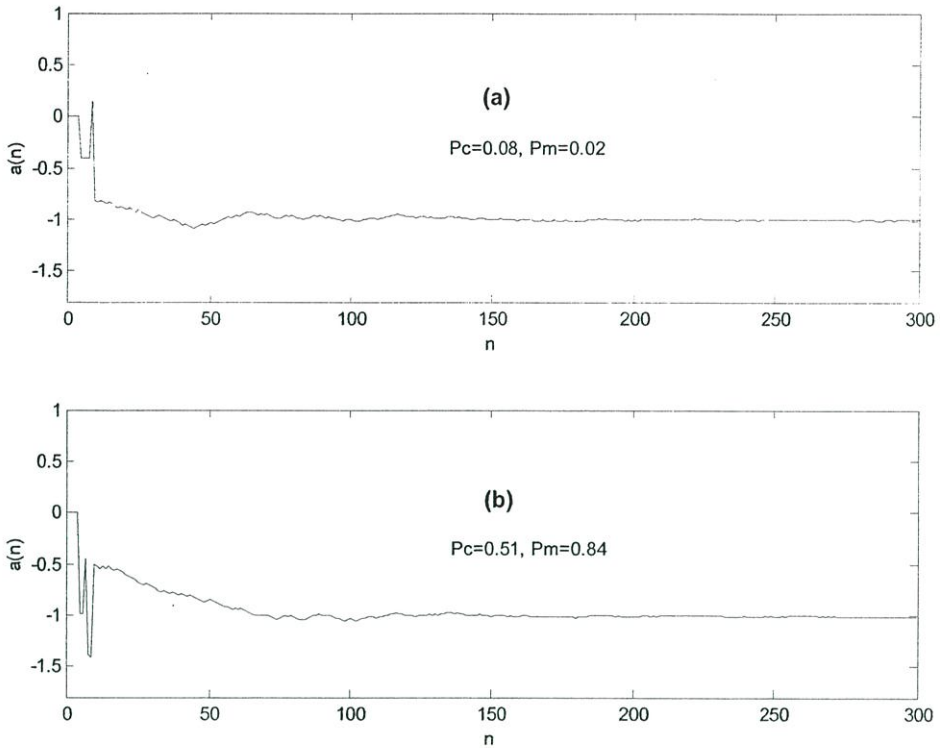
- นำค่าความเร็วการลู่เข้า และจำนวนครั้งที่พบว่าสัมประสิทธิ์ของตัวกรอง (a) มีความเปลี่ยนแปลงของแต่ละ P_c และ P_m เพื่อหาค่าความเร็วการลู่เข้าที่น้อยสุด (Speed_Min) และ ค่าความเร็วลู่เข้าเฉลี่ย (Speed_Avg)
- เก็บผลลัพธ์ P_c , P_m , Speed_min, Speed_Avg และจำนวนครั้งที่ค่าสัมประสิทธิ์ของตัวกรอง (a) มีการเปลี่ยนแปลง เพื่อนำไปใช้ในการวิเคราะห์เลือกค่า P_c และ P_m ที่ดีที่สุด

3. นำค่าผลลัพธ์ที่ได้จากข้อ 2 ของทุก P_c และ P_m มาวิเคราะห์ จะได้ผลดังรูปที่ 5.12



รูปที่ 5.12 ความสัมพันธ์ของ P_c และ P_m ที่ส่งผลต่อค่าความแปรปรวน

จากรูปที่ 5.12 พบว่าจุดที่ค่า P_m ต่ำ จะมีค่าความแปรปรวนต่ำด้วย ซึ่งจากการดูเทียบกับข้อมูลพบว่า ค่า P_m ที่ดี คือ P_m มีค่าระหว่าง 0.01 ถึง 0.03 จากนั้นได้เลือกค่า P_c และ P_m ที่เหมาะสม คือ P_m มีค่าระหว่าง 0.01 ถึง 0.03 มาทำการทดสอบกับตัวกรองสัญญาณ โดยทดสอบเปรียบเทียบกับค่า P_m ที่มีค่าสูง ได้ผลการทดสอบเป็นดังรูปที่ 5.13



รูปที่ 5.13 เปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ ของ อัลกอริทึมแบบปรับเสถียรไซส์ ทำงานร่วมกับอัลกอริทึมแบบเจนเนติก [35] ที่ $\omega = \pi/3$ โดย (a) ใช้ค่า P_c, P_m ที่เหมาะสม (ค่า P_m มีค่าต่ำ) คือ $P_c=0.08, P_m=0.02$ และ (b) $P_c=0.51, P_m=0.84$ (ค่า P_m สูง)

จากรูปที่ 5.13 แสดงให้เห็นว่ากราฟค่าสัมประสิทธิ์ของตัวกรองที่ $P_m = 0.84$ มีความแปรปรวนสูงกว่า กราฟค่าสัมประสิทธิ์ของตัวกรองที่ $P_m = 0.02$ ซึ่งในวิทยานิพนธ์นี้ จะใช้ค่า P_c และ P_m ที่เหมาะสม โดยจะใช้ค่า $P_c = 0.08$ และ $P_m = 0.02$ ในการหาค่าเสถียรไซส์ กับอัลกอริทึมแบบใหม่

5.6.3 อะแดปทีฟอัลกอริทึมปรับเสถียรไซส์แบบใหม่ และอัลกอริทึมเจนเนติกแบบใหม่

ในหัวข้อนี้กล่าวถึงแนวคิดที่ใช้ในการปรับปรุงอัลกอริทึมแบบปรับเสถียรไซส์แบบใหม่ และอัลกอริทึมแบบเจนเนติกแบบใหม่ ที่จะใช้ในการพัฒนาอะแดปทีฟอัลกอริทึมแบบใหม่

1. อะแดปทีฟอัลกอริทึมแบบปรับเสถียรไซส์แบบใหม่

วิทยานิพนธ์นี้ได้นำแนวความคิดอัลกอริทึมแบบ VSQMP [29] มาใช้ในการปรับปรุงอัลกอริทึมแบบปรับเสถียรไซส์ โดยพบว่าค่าเสถียรไซส์จะปรับตัวตามค่าพลังงานของสัญญาณเอาต์พุต $\hat{y}(n)$ ในช่วงแรก ขณะที่ตัวกรองเริ่มทำงานนั้นไม่สามารถตรวจวัดความถี่ได้ ค่าพลังงานของเอาต์พุต $\hat{y}(n)$ มีค่าสูง เป็นผลให้ค่าเสถียรไซส์มีค่าสูง จึงทำงานได้เร็ว หลังจากการลู่เข้าสู่ค่าตอบแล้ว ค่าพลังงานของเอาต์พุต $\hat{y}(n)$ มีค่าต่ำ โดยเหลือเฉพาะพลังงานของสัญญาณรบกวนเท่านั้น ดังนั้นเมื่อเกิดสัญญาณรบกวนแบบอิมพัลส์ จะทำให้เกิดความแปรปรวนของค่าสัมประสิทธิ์

ตัวกรองสูงมาก และอาจทำให้ไม่สามารถลู่เข้าสู่คำตอบได้อีก จากปัญหาดังกล่าวจึงได้นำค่าความผิดพลาดของเอาต์พุตมาช่วยแก้ปัญหาเรื่องสัญญาณรบกวน จากสมการ (5.48) คือ

$$\psi(n) = \alpha \cdot \psi(n-1) + (1 - \alpha) y^2(n)$$

เมื่อนำค่าความผิดพลาดของเอาต์พุตเข้ามาช่วยจะได้ดังสมการ

$$\psi(n) = \alpha \cdot \psi(n-1) + (1 - \alpha) y(n) \cdot e(n-1) \quad (5.57)$$

จะเห็นว่าค่าความผิดพลาดของเอาต์พุตที่นำมาใช้ เป็นค่าความผิดพลาดของเอาต์พุตก่อนหน้า คือ $e(n-1)$ ผลที่ได้ คือทำให้ได้อะแดปทีฟอัลกอริทึมแบบใหม่ ที่ทนต่อสัญญาณรบกวนประเภทอิมพัลส์ได้ดีกว่าอัลกอริทึมแบบ VSQ LMP [29] และทนต่อสัญญาณรบกวนได้ใกล้เคียงกับอัลกอริทึมแบบปรับเสถียร [35]

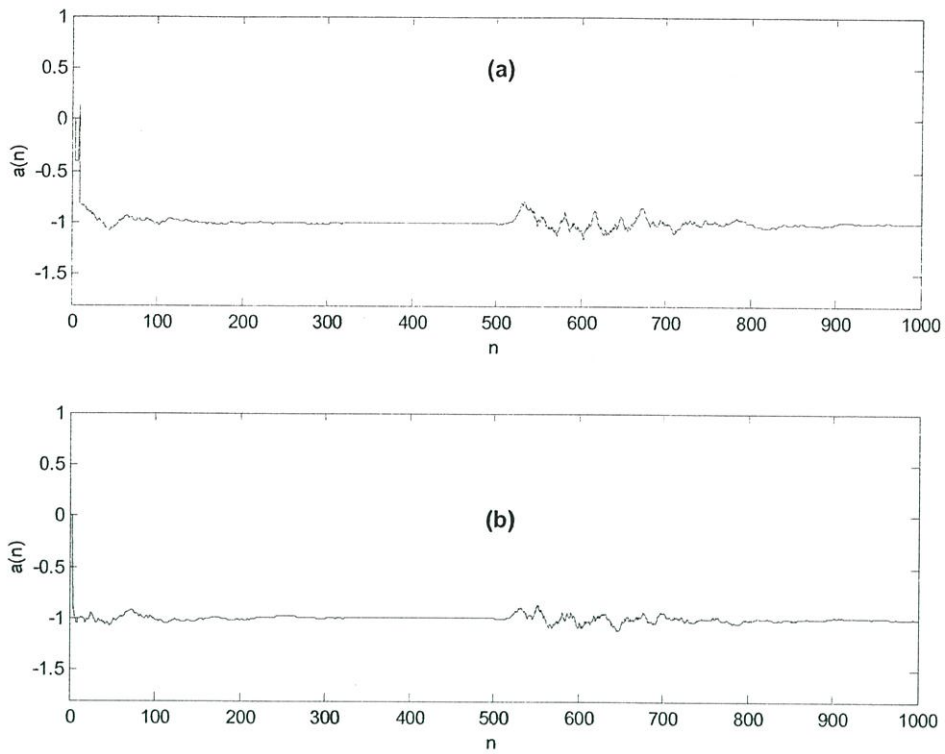
2. อัลกอริทึมเจเนติกแบบใหม่

อัลกอริทึมเจเนติกแบบใหม่นี้ได้นำอัลกอริทึมแบบเจเนติก [35] มาทำการปรับปรุงให้ดีขึ้น กล่าวคือ จากรูปที่ 5.11 ใช้ค่า $e^2[n]$ ในการหาค่า fitness ของโครโมโซม แต่พบว่าคำตอบที่ได้บางครั้งไม่แน่นอน ดังนั้นจึงนำค่า Bias มาหาค่า fitness แทนดังสมการ

$$Fitness(n) = 1 - BIAS[n] \quad (5.58)$$

กำหนดจุดเปลี่ยนจากอัลกอริทึมแบบเจเนติกไปเป็นอัลกอริทึมแบบปรับเสถียร ซึ่งเดิมใช้ค่า e^2 เทียบกับค่า Theshold (0.35) หาก $e^2 > Theshold$ ก็จะเปลี่ยนไปใช้อัลกอริทึมแบบปรับเสถียร สำหรับอัลกอริทึมแบบใหม่จะใช้ค่า Bias(GA) เทียบกับค่า Theshold2 (0.0001) หาก $Bias(GA) < Theshold2$ จะเปลี่ยนไปใช้อัลกอริทึมแบบปรับเสถียร

ผลจากการทำงานร่วมกันของทั้งสองอัลกอริทึมนี้ จะได้ตัวกรองสัญญาณที่มีความเร็วลู่เข้าสู่คำตอบสูงขึ้น ความแปรปรวนต่ำลง ค่าความผิดพลาดลดต่ำลงด้วย แต่ระดับความทนสัญญาณรบกวนแบบอิมพัลส์จะใกล้เคียงกับอัลกอริทึมแบบเจเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียร [35] แต่ค่าความผิดพลาดและความแปรปรวนจะต่ำกว่ามาก ผลที่ได้สามารถแสดงดังรูปที่ 5.14



รูปที่ 5.14 ค่าของสัมประสิทธิ์ฟิลเตอร์ $a(n)$ เมื่อเกิดสัญญาณรบกวนแบบอิมพัลส์ ขนาด 50 หน่วย ที่ตำแหน่ง $n = 500$ (a) อัลกอริทึมแบบเจเนติกทำงานร่วมกับ อัลกอริทึมแบบปรับเสถียรไซส์ [35] (b) อัลกอริทึมแบบเจเนติกแบบใหม่ทำงานร่วมกับ อัลกอริทึมแบบปรับเสถียรไซส์แบบใหม่

5.6.4 หลักการทำงานของอัลกอริทึมใหม่ที่น่าสนใจ

จากหัวข้อ 5.6.3 ได้นำเสนออัลกอริทึมแบบเจนเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์แบบใหม่ แม้ว่าจะเป็นอัลกอริทึมที่มีความเร็วเข้าสู่ค่าตอบสูงและความแปรปรวนต่ำกว่าอัลกอริทึมแบบเจนเนติกทำงานร่วมกับอัลกอริทึมแบบเสถียรไซส์ [35] แต่ยังคงพบว่ายังมีความแปรปรวนในช่วงแรกของการปรับตัวของอัลกอริทึม และช่วงที่มีการเปลี่ยนการทำงานจากอัลกอริทึมแบบเจนเนติกไปเป็นอัลกอริทึมแบบปรับเสถียรไซส์

อัลกอริทึมใหม่ที่น่าสนใจ เป็นอัลกอริทึมที่นำเอาข้อดีของอัลกอริทึมแบบปรับเสถียรไซส์ ที่มีความแปรปรวนต่ำ แต่มีข้อเสียคือมีความเร็วเข้าสู่ค่าตอบช้า และ อัลกอริทึมแบบเจนเนติกที่มีข้อดีคือ มีความเร็วเข้าสู่ค่าตอบสูง แต่ข้อเสียคือ มีความแปรปรวนสูงด้วย ซึ่งจะเห็นว่าหากนำข้อดีของทั้งสองอัลกอริทึมมาทำงานร่วมกัน โดยใช้ทั้งสองอัลกอริทึมทำการหาเสถียรไซส์ไปพร้อม ๆ กันแบบคู่ขนาน อัลกอริทึมแบบใหม่ที่น่าสนใจนี้จะเลือกใช้อัลกอริทึมใดให้ค่าเสถียรไซส์ที่ดีที่สุดจะใช้ค่าเสถียรไซส์ของอัลกอริทึมนั้น และทำงานด้วย 2 อัลกอริทึมไปจนกว่าอัลกอริทึมแบบปรับเสถียรไซส์จะเริ่มเข้าสู่ค่าตอบ จะเปลี่ยนไปใช้อัลกอริทึมแบบปรับเสถียรไซส์เพียงอัลกอริทึมเดียวเพื่อลดการคำนวณที่ค่อนข้างช้าของอัลกอริทึมแบบเจนเนติก ซึ่งหลักการของอัลกอริทึมแบบใหม่ที่น่าสนใจนี้ แสดงไว้ดังรูปที่ 5.15

จากรูปที่ 5.15 แสดงหลักการทำงานของอัลกอริทึมใหม่ เริ่มต้นขบวนการหาเสถียรไซส์ด้วยการหาค่าคาดหวัง ($E[a]$) โดยที่ค่าคาดหวังกำหนดได้ดังสมการ

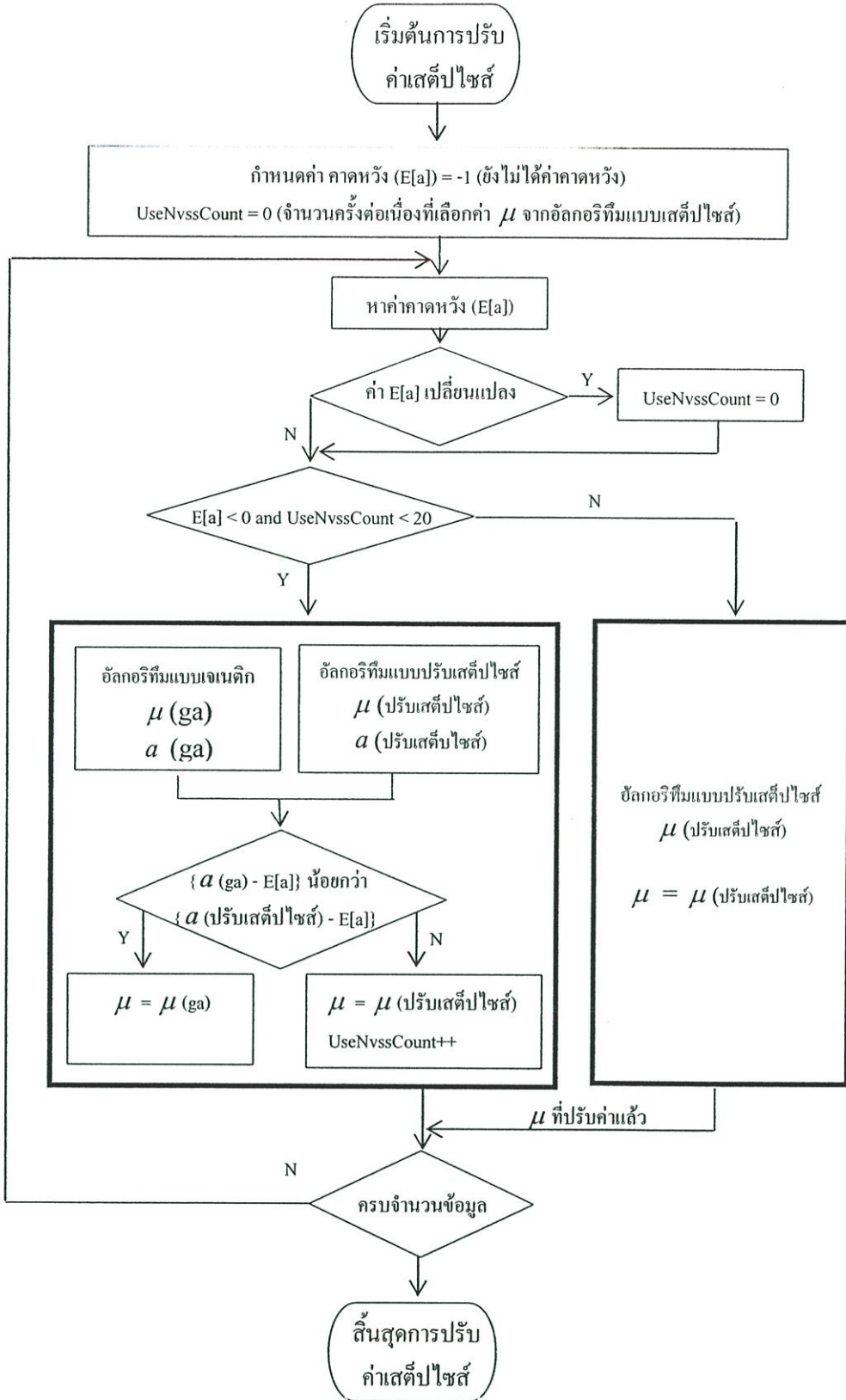
$$E[a] = \cos(\pi) * |\cos(\omega)| * 2 \quad (5.59)$$

เมื่อ $E[a]$ คือ ค่าคาดหวัง

ω = ค่าความถี่นอตรัสที่ใช้งาน มีค่าเท่ากับ π/n

n = เป็นค่าคงที่ (2, 3, 4, ..., n)

ถ้ายังไม่ได้ค่าคาดหวังจะหาค่า μ ด้วยอัลกอริทึมแบบปรับเสถียรไซส์ เมื่อได้ค่าคาดหวังแล้วจะเริ่มหาค่าเสถียรไซส์ μ ด้วยอัลกอริทึมแบบเจนเนติก และอัลกอริทึมแบบปรับเสถียรไซส์ไปพร้อม ๆ กัน ซึ่งจะมีการเปรียบเทียบเลือกค่าเสถียรไซส์ที่ดีที่สุด โดยเทียบค่าสัมประสิทธิ์ของตัวกรอง (a) กับค่าคาดหวัง ถ้าค่าสัมประสิทธิ์ของตัวกรอง (a) ของอัลกอริทึมใดใกล้เคียงกว่าก็จะใช้ค่าเสถียรไซส์ μ จากอัลกอริทึมนั้น เพื่อใช้หาค่าเสถียรไซส์รอบต่อไป ซึ่งระหว่างการหาค่าเสถียรไซส์ด้วยทั้งสองอัลกอริทึมนี้ อัลกอริทึมแบบปรับเสถียรไซส์จะค่อย ๆ ปรับตัวเองเพื่อให้ได้ค่าคำตอบที่ดีที่สุด (เริ่มเข้าสู่ค่าตอบ)



รูปที่ 5.15 หลักการทำงานของอัลกอริทึมใหม่

หากพบค่าเสถียรภาพ μ ที่ได้ มาจากอัลกอริทึมแบบปรับเสถียรภาพต่อเนื่องกันเกิน 20 ครั้ง จะให้ใช้อัลกอริทึมแบบปรับเสถียรภาพเพียงอย่างเดียว เพื่อประหยัดเวลาในการเปรียบเทียบ ทั้งนี้ ระหว่างการหาค่าเสถียรภาพ μ หากพบว่าสัญญาณอินพุตที่เข้ามามีค่าสูงหรือต่ำกว่าปกติ (เช่น เกิดสัญญาณรบกวนอิมพัลส์) จะกลับเข้าสู่การหาค่าเสถียรภาพ μ แบบสองอัลกอริทึมไปพร้อม ๆ กัน อีกครั้ง และทำงานด้วยสองอัลกอริทึมนี้จนกว่าค่าเสถียรภาพ μ ที่เลือกมาจากอัลกอริทึมแบบปรับเสถียรภาพต่อเนื่องกัน 20 ครั้ง จะใช้ค่าเสถียรภาพ μ แบบอัลกอริทึมเสถียรภาพอย่างเดียวย่อกรอบ

จากวิทยานิพนธ์ [35] กำหนดค่า P_c และ P_m ด้วยวิธีลองผิด ลองถูก ซึ่งอาจจะทำให้ไม่ได้การทำงานที่ดีที่สุด ดังนั้นจึงมีความจำเป็นที่จะต้องหาค่า P_c และ P_m ที่ใช้ในขั้นตอนการครอส โอเวอร์ และ การมิวเตชัน ที่เหมาะสมที่สุด ซึ่งเป็นกระบวนการหนึ่งในการหาเสถียรภาพของอัลกอริทึมแบบเจเนติก เพราะหากเลือกค่า P_c และ P_m ไม่ดี จะมีผลต่อประสิทธิภาพการกรองสัญญาณด้วย จึงได้ทำการทดสอบ เพื่อหาค่า P_c และ P_m ที่เหมาะสมที่สุดดังหัวข้อที่ 5.6.2 ทำให้การกรองสัญญาณมีประสิทธิภาพสูงที่สุดตามไปด้วย โดยค่าความสัมพันธ์ใหม่นี้ จะมีผลทำให้ความเร็วผู้เข้าสู่ค่าสัมประสิทธิ์ที่ต้องการสูงขึ้น และค่าความผิดพลาดลดลง

5.7 สรุป

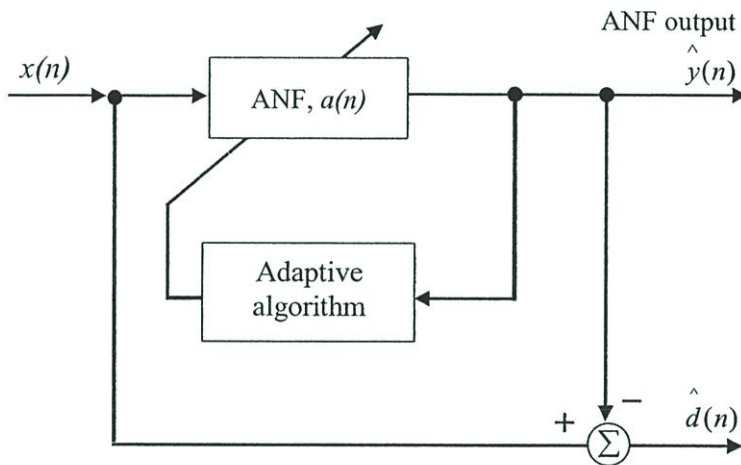
สำหรับอะแดปทีฟอัลกอริทึมที่นำเสนอในวิทยานิพนธ์นี้ได้ปรับปรุงจากอะแดปทีฟอัลกอริทึมแบบปรับเสถียรภาพ และอัลกอริทึมแบบเจเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรภาพ [35] ซึ่งการปรับปรุงอัลกอริทึมนี้ ส่วนใหญ่มักจะทำในลักษณะที่ต้องการเพิ่มความเร็วในการลู่เข้าสู่คำตอบ หรือ เพิ่มความทนทานต่อสัญญาณรบกวน ในการปรับปรุงอะแดปทีฟอัลกอริทึมแบบใหม่นี้ได้ใช้ข้อดีของอัลกอริทึมแบบปรับเสถียรภาพ คือ มีความทนต่อสัญญาณรบกวนแบบอิมพัลส์ และให้ค่าความผิดพลาดต่ำ มาทำงานร่วมกับอัลกอริทึมแบบเจเนติก ที่มีค่าความเร็วผู้เข้าสู่คำตอบสูง โดยการปรับเปลี่ยนรูปแบบการทำงานร่วมกันของอัลกอริทึมแบบเจเนติกกับอัลกอริทึมแบบปรับเสถียรภาพ ให้ทำงานแบบคู่ขนานกัน จึงได้อัลกอริทึมใหม่ที่มีความเร็วผู้เข้าสู่คำตอบสูง ความแปรปรวนต่ำ และทนทานต่อสัญญาณรบกวนแบบอิมพัลส์ได้สูง

ส่วนการทดสอบการทำงานของ อัลกอริทึมแบบใหม่นั้นจะนำเสนอในบทถัดไป

บทที่ 6

ผลการวิจัย

ในบทนี้จะแสดงผลการทำงานของอะแดปทีฟอัลกอริทึมแบบเจนเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์ [35] อะแดปทีฟอัลกอริทึมแบบเจนเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์แบบใหม่ และอัลกอริทึมใหม่ที่น่าสนใจ ซึ่งกล่าวถึงแล้วในบทที่ 5 โดยการทดสอบทั้งหมดนี้ได้อาศัยการจำลองการทำงานผ่าน โปรแกรม MATLAB และ ภาษาซี ส่วนประสิทธิภาพของอะแดปทีฟอัลกอริทึมนี้ สามารถดูได้จากความเร็วในการลู่เข้าหาคำตอบและค่า MSE ของสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ กล่าวคืออะแดปทีฟอัลกอริทึมที่ดีนั้นต้องมีความเร็วในการหาคำตอบสูง ค่าMSE ต่ำ และ ทนต่อสัญญาณรบกวนได้ดี และสามารถประยุกต์นำไปใช้งานได้จริง ดังที่กล่าวไว้ในบทที่ 4 บล็อกไดอะแกรมสำหรับการประมาณค่าความถี่ แสดงได้ดังรูปที่ 6.1



รูปที่ 6.1 บล็อกไดอะแกรมอะแดปทีฟฟอนดซ์ฟิลเตอร์

จากรูปที่ 6.1 $x(n)$ คือ สัญญาณอินพุต ประกอบด้วยสัญญาณชายน้ก่ล้นเดี่ยว และสัญญาณรบกวน $\hat{y}(n)$ คือ สัญญาณเอาต์พุตของอะแดปทีฟฟอนดซ์ฟิลเตอร์ $\hat{d}(n)$ คือ สัญญาณเอาต์พุตของอะแดปทีฟแบนด์พาสฟิลเตอร์

6.1 การทดสอบอะแดปทีฟอัลกอริทึมในการประมาณสัญญาณชายน้ก่ล้นเดี่ยว

หัวข้อนี้เป็นการทดสอบการทำงานของอะแดปทีฟอัลกอริทึมในวิทยานิพนธ์ฉบับนี้ โดยนำมาประมาณสัญญาณชายน้ก่ที่มีสัญญาณรบกวนแบบเกาส์เซียน (White Gaussian noise) หรือ

สัญญาณรบกวนแบบอิมพัลส์ (Impulse noise) ปะปนอยู่ อัลกอริทึมที่นำมาเปรียบเทียบ คือ อัลกอริทึมแบบเจนเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียร [35] โดยนำมาเปรียบเทียบกับ อัลกอริทึมแบบเจนเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรแบบใหม่ และ อัลกอริทึมแบบใหม่ที่น่าเสนอ

6.1.1 การทดสอบอะแดปทีฟอัลกอริทึมโดยที่มีค่า $\mu(0)$ ที่แตกต่างกัน

ในหัวข้อนี้จะเป็นการทดสอบตัวกรองสัญญาณโดยใช้อัลกอริทึม 3 แบบ คือ อัลกอริทึมแบบเจนเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียร [35] อัลกอริทึมแบบเจนเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรแบบใหม่ และอัลกอริทึมใหม่ที่น่าเสนอ กำหนดสัญญาณอินพุตตามสมการ

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

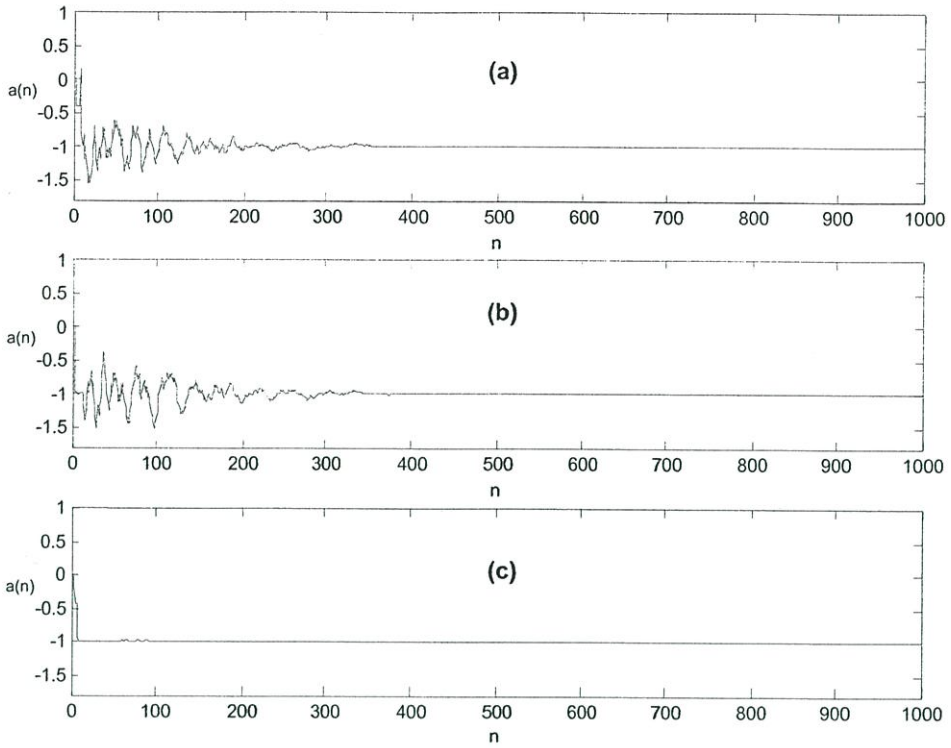
โดยที่ A คือ แอมพลิจูด

ω_0 คือ ความถี่เชิงมุม

ϕ คือ เฟสของความถี่เชิงมุม และ

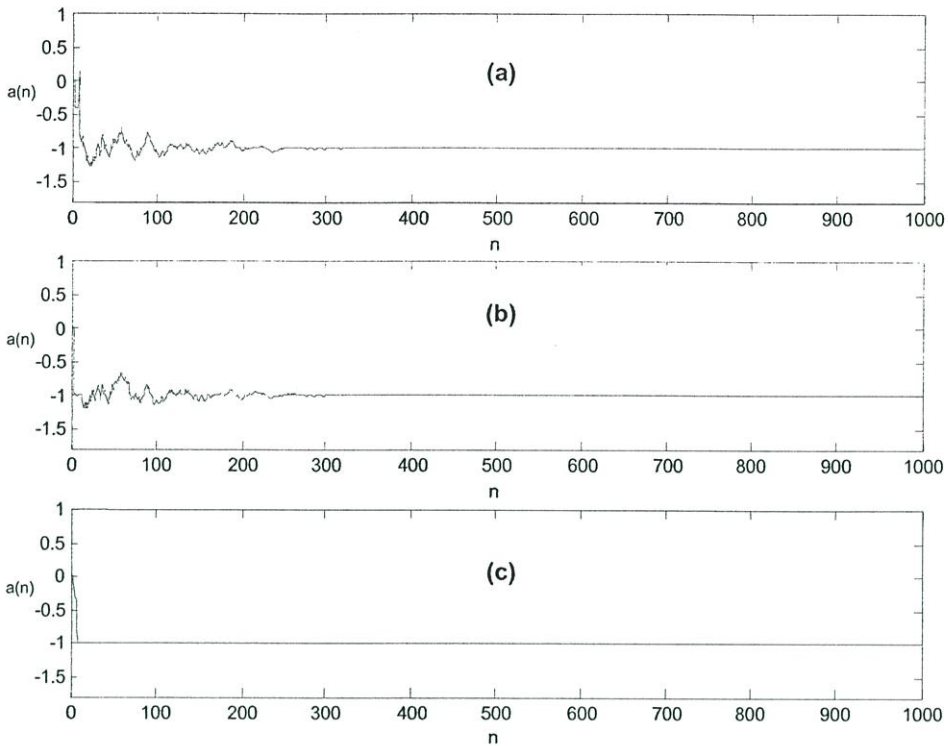
$v(n)$ คือ สัญญาณรบกวนแบบเกาส์เซียน

กำหนดพารามิเตอร์ของ อัลกอริทึมแบบปรับเสถียร มีค่าดังนี้ $A = 1$, $\phi = 0$, $\omega_0 = \pi/3$, $\rho = 0.9$, $\gamma = 0.99$, $\alpha = 0.99$, $\lambda = 0.001$, SNR = 7 dB และในส่วนของอัลกอริทึมแบบเจนเนติก กำหนดค่าพารามิเตอร์ดังนี้ Pm = 0.02, Pc = 0.08 และ Popsize = 30 สำหรับค่า $\mu(0)$ ที่ใช้ในการทดสอบจะมีค่า 0.2, 0.1 และ 0.01 ตามลำดับ โดยในรูปที่ 6.2 ให้ค่าเสถียรเริ่มต้นเท่ากับ 0.2 รูปที่ 6.3 ให้ค่าเสถียรเริ่มต้นเท่ากับ 0.1 และ รูปที่ 6.4 ให้ค่าเสถียรเริ่มต้นเท่ากับ 0.02 ตามลำดับ จากรูปที่ 6.2 ถึง 6.4 แสดงให้เห็นว่า ค่าเสถียรเริ่มต้น มีผลต่ออัลกอริทึมแบบปรับเสถียร กล่าวคือ เมื่อค่าเสถียรเริ่มต้นมีค่าสูงจะทำให้สามารถหาคำตอบได้เร็ว แต่ถ้ามีค่าสูงมากเกินไปทำให้เกิดความแปรปรวนสูงในช่วงแรก ดังรูปที่ 6.2 หรือ ในกรณีที่ค่าเสถียรเริ่มต้นมีค่าสูงมาก อาจทำให้ไม่สามารถหาคำตอบได้เลย เนื่องจากค่าเสถียร สำหรับตัวกรองอะแดปทีฟแบบ IIR ยังไม่มีข้อสรุปที่แน่นอนดังที่กล่าวไว้ในบทที่ 4 ดังนั้นจึงไม่มีข้อสรุปสำหรับค่าเสถียรเริ่มต้น ด้วย การกำหนดค่าเสถียรเริ่มต้น จึงเป็นการลองผิดลองถูก แต่ในกรณีของอัลกอริทึมแบบใหม่ที่น่าเสนอนี้ ค่าเสถียรเริ่มต้นแทบจะไม่ส่งผลกับการหาคำตอบเลย



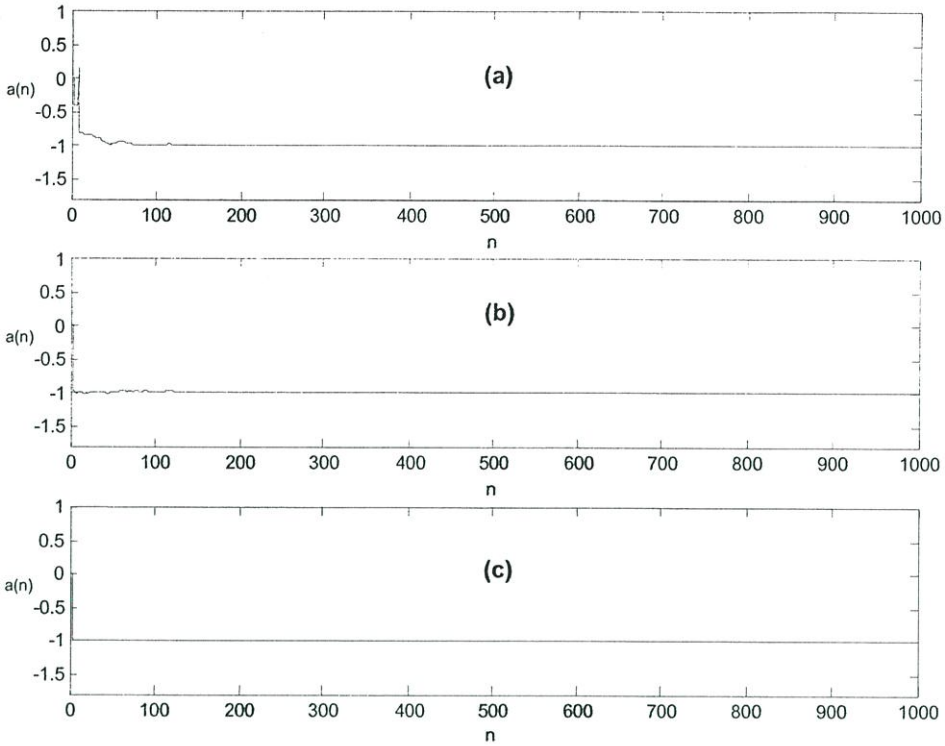
รูปที่ 6.2 เปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ ที่ $\mu(0) = 0.2$ (a) อัลกอริทึมแบบเจนเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์ [35] (b) อัลกอริทึมแบบเจนเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์แบบใหม่ และ (c) อัลกอริทึมใหม่ที่นำเสนอ

จากรูปที่ 6.2 เป็นการเปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ ที่ $\mu(0) = 0.2$ จะเห็นว่ารูปที่ 6.2 (a) และ 6.2 (b) มีความแปรปรวนค่อนข้างสูงช่วงต้นของการทำงานของอัลกอริทึม หากเปรียบเทียบกับรูปที่ 6.2 (c) จะเห็นว่ารูปที่ 6.2 (c) มีความแปรปรวนน้อยกว่ารูปที่ 6.2 (a) และ 6.2 (b)



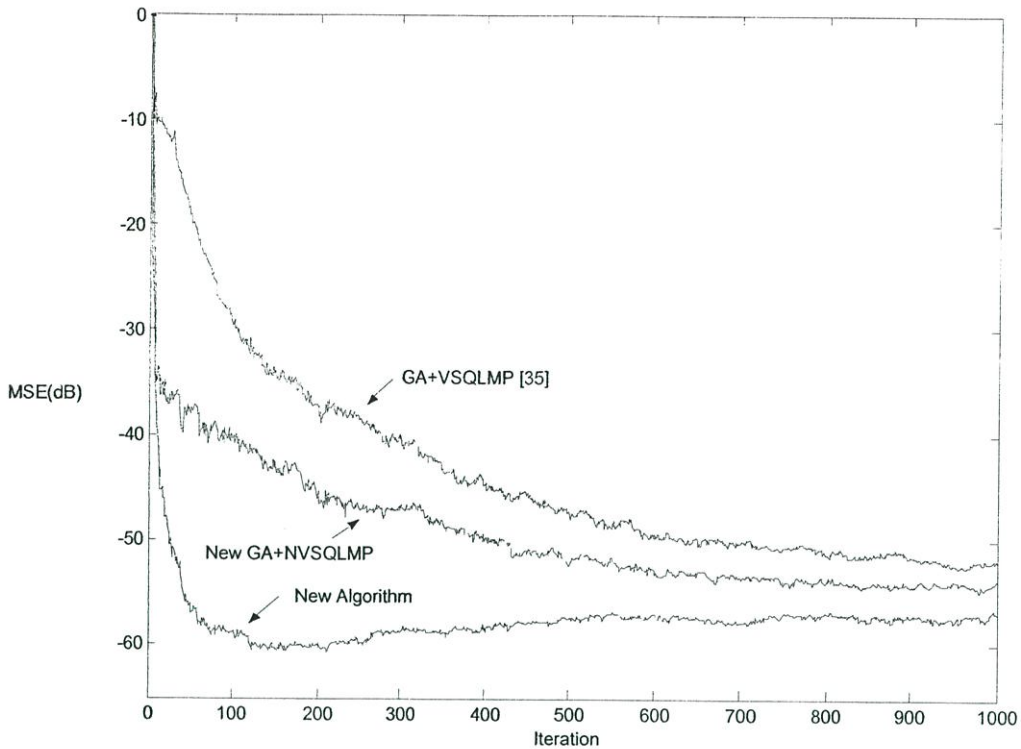
รูปที่ 6.3 เปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ ที่ $\mu(0) = 0.1$ (a) อัลกอริทึมแบบเจนเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์ [35] (b) อัลกอริทึมแบบเจนเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์แบบใหม่ และ (c) อัลกอริทึมใหม่ที่นำเสนอ

จากรูปที่ 6.3 เป็นรูปเปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ ที่ $\mu(0) = 0.1$ จะเห็นว่ารูปที่ 6.3 (a) และ 6.3 (b) มีความแปรปรวนค่อนข้างสูง แต่จะเห็นว่า ความเร็วเข้าสู่ค่าตอบของรูปที่ 6.3 (a) ช้ากว่ารูปที่ 6.3 (b) หากดูเทียบกับรูปที่ 6.2 นั้นแสดงว่าค่า $\mu(0)$ สูง จะทำให้อัลกอริทึมแบบปรับเสถียรไซส์มีความแปรปรวนสูงขึ้นไปด้วย หากเปรียบเทียบกับรูปที่ 6.3 (c) จะเห็นว่ารูปที่ 6.3 (c) มีความแปรปรวนน้อยกว่ารูปที่ 6.3 (a) และ 6.3 (b)



รูปที่ 6.4 เปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ ที่ $\mu(0)=0.01$ (a) อัลกอริทึมแบบเจนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์ [35] (b) อัลกอริทึมแบบเจนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์แบบใหม่ และ (c) อัลกอริทึมใหม่ที่น่าเสนอ

จากรูปที่ 6.4 เป็นรูปเปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ ที่ $\mu(0) = 0.01$ จะเห็นว่า รูปที่ 6.4 (a) มีความเร็วเข้าสู่ค่าตอบที่ช้าลงมาก ส่วนรูปที่ 6.4 (b) และ 6.4 (c) มีความเร็วเข้าสู่ค่าตอบสูงเมื่อเทียบกับรูปที่ 6.2 และ 6.3 สำหรับความแปรปรวนของทั้งสามอัลกอริทึมจะลดลงมาก นั่นแสดงว่าค่า $\mu(0)$ ต่ำ จะส่งผลโดยตรงกับความเร็เข้าสู่ค่าตอบของอัลกอริทึมแบบปรับเสถียรไซส์ แต่ไม่มีผลกับความเร็เข้าสู่ค่าตอบของอัลกอริทึมแบบเจนติก จากรูปที่ 6.4 นี้ ได้แสดงให้เห็นว่า อัลกอริทึมแบบใหม่ที่น่าเสนอ จะมีความเร็วในการหาค่าตอบสูงกว่า และค่าความแปรปรวนต่ำกว่า ด้วย ซึ่งทำการเปรียบเทียบค่า MSE ของทั้งสามอัลกอริทึม จะได้ดังรูปที่ 6.5



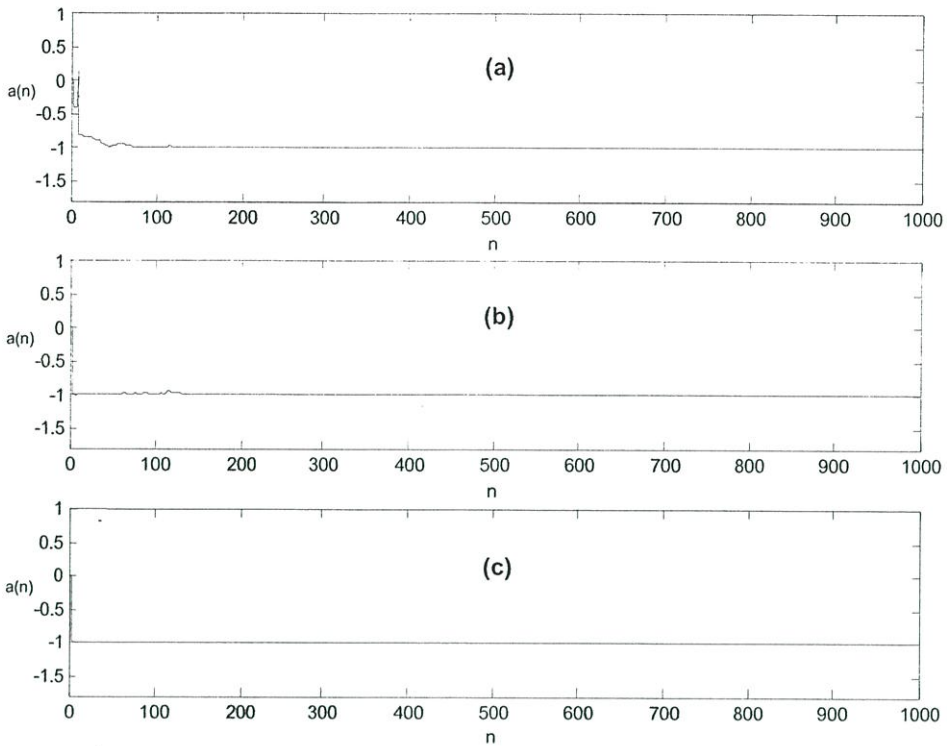
รูปที่ 6.5 กราฟเปรียบเทียบค่า MSE ของอัลกอริทึมแบบเจเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์ [35] อัลกอริทึมแบบปรับเจเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์แบบใหม่ และอัลกอริทึมใหม่ที่น่าเสนอ เมื่อกำหนดให้ $\mu(0) = 0.01$

จากรูปที่ 6.5 แสดงให้เห็นว่าอัลกอริทึมแบบใหม่ที่น่าเสนอนี้ มีค่า MSE ต่ำกว่า อัลกอริทึมแบบเจเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์ [35] และต่ำกว่าอัลกอริทึมแบบเจเนติกแบบใหม่ที่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์แบบใหม่

6.1.2 การทดสอบอะแดปทีฟอัลกอริทึมโดยที่มีค่า ω_0 ในสมการอินพุตที่แตกต่างกัน

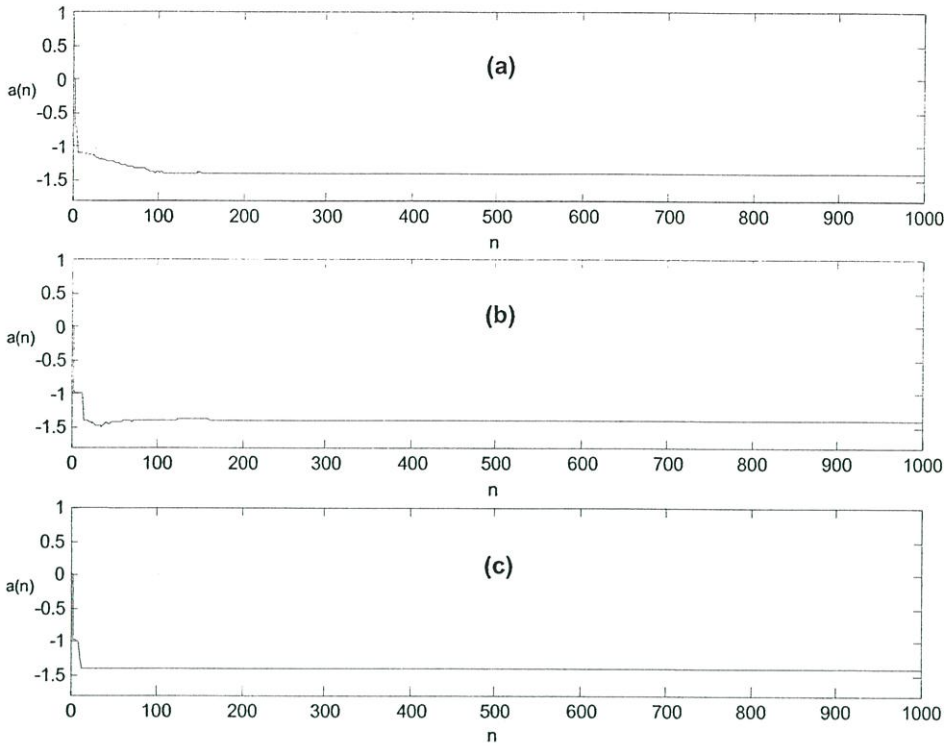
ในหัวข้อนี้จะเป็นการทดสอบตัวกรองสัญญาณโดยใช้อัลกอริทึมสามแบบ คือ อัลกอริทึมแบบเจเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์ [35] อัลกอริทึมแบบเจเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์แบบใหม่ และอัลกอริทึมแบบใหม่ที่น่าเสนอ โดยที่กำหนดสัญญาณอินพุตตามสมการที่ (6.1)

กำหนดพารามิเตอร์ของ อัลกอริทึมแบบปรับเสถียรไซส์ มีค่าดังนี้ $A = 1$, $\phi = 0$, $\mu(0) = 0.01$, $\rho = 0.9$, $\gamma = 0.99$, $\alpha = 0.99$, $\lambda = 0.001$, $\text{SNR} = 7$ dB และในส่วนของอัลกอริทึมแบบเจเนติก กำหนดค่าพารามิเตอร์ดังนี้ $P_m = 0.02$, $P_c = 0.08$ และ $\text{Popsiz} = 30$ สำหรับค่า ω_0 ที่ใช้ในการทดสอบจะมีค่า $\pi/3$, $\pi/4$ และ $\pi/7$ จากการทดสอบได้ผลดังรูปที่ 6.6 ถึง รูปที่ 6.8



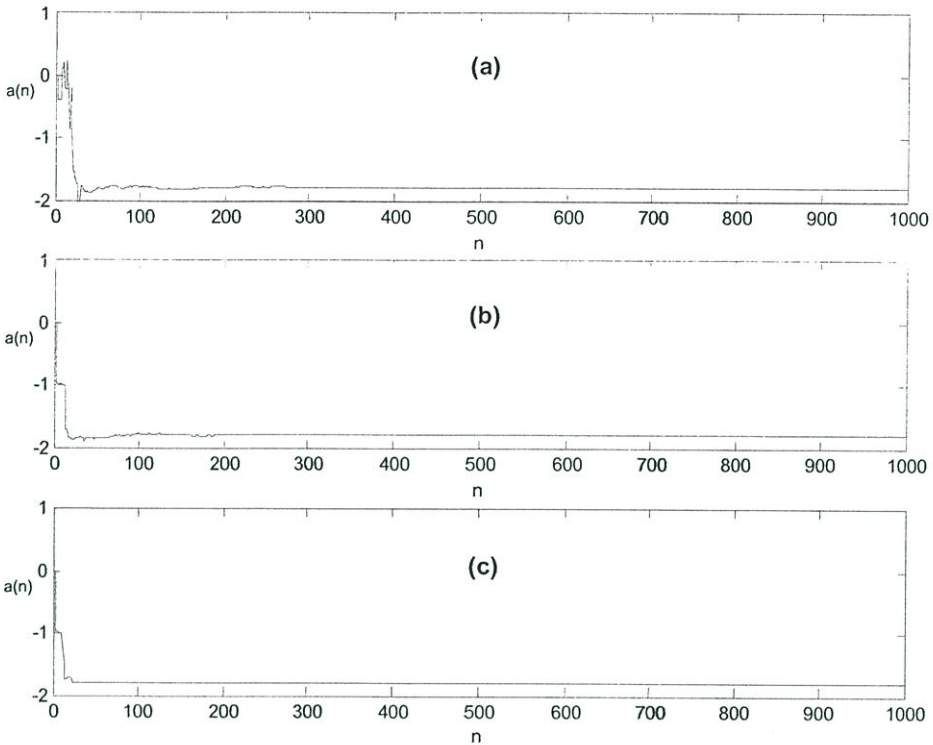
รูปที่ 6.6 เปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ ที่ $\omega = \pi/3$ (a) อัลกอริทึมแบบเจนเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์[35] (b) อัลกอริทึมแบบเจนเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์แบบใหม่ และ (c) อัลกอริทึมใหม่ที่นำเสนอ

จากรูปที่ 6.6 จะเห็นว่ารูปที่ 6.6 (a) มีความเร็วเข้าสู่ค่าตอบช้าสุด ประมาณข้อมูลที่ 40 ส่วนรูปที่ 6.6 (b) และ 6.6 (c) มีความเร็วเข้าสู่ค่าตอบใกล้เคียงกัน คือ ประมาณข้อมูลที่ 5 แต่รูปที่ 6.6 (a) และ 6.6 (b) จะพบความแปรปรวนในช่วงแรก ในขณะที่รูปที่ 6.6 (c) มีความแปรปรวนต่ำมากหรือแทบไม่เกิดความแปรปรวนเลย



รูปที่ 6.7 เปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ ที่ $\omega = \pi/4$ (a) อัลกอริทึมแบบเจนเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์[35] (b) อัลกอริทึมแบบเจนเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์แบบใหม่ และ (c) อัลกอริทึมใหม่ที่นำเสนอ

จากรูปที่ 6.7 เมื่อเปลี่ยนค่า ω เป็น $\pi/4$ จะเห็นว่ารูปที่ 6.7 (a) มีความเร็วเข้าสู่ค่าตอบช้าลงเมื่อเทียบกับรูปที่ 6.6 ที่ใช้ค่า ω เป็น $\pi/3$ ส่วนรูปที่ 6.7 (b) และ 6.7 (c) มีความเร็วเข้าสู่ค่าตอบที่ช้าลงเล็กน้อย คือ ประมาณข้อมูลที่ 10-15 จะเห็นว่าหากค่า ω เปลี่ยนไปจะส่งผลให้การหาค่าคำตอบช้าลงด้วย แต่หากเทียบความเร็วเข้าสู่ค่าตอบระหว่างรูปที่ 6.7 (a), 6.7 (b) และ 6.7 (c) ที่ $\omega = \pi/4$ จะเห็นว่าความเร็วเข้าสู่ค่าตอบของรูปที่ 6.7 (c) สูงกว่ารูปที่ 6.7 (a) อย่างเห็นได้ชัด



รูปที่ 6.8 เปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ ที่ $\omega = \pi/7$ (a) อัลกอริทึมแบบเจนเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไพไซส์[35] (b) อัลกอริทึมแบบเจนเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไพไซส์แบบใหม่ และ (c) อัลกอริทึมใหม่ที่น่าเสนอ

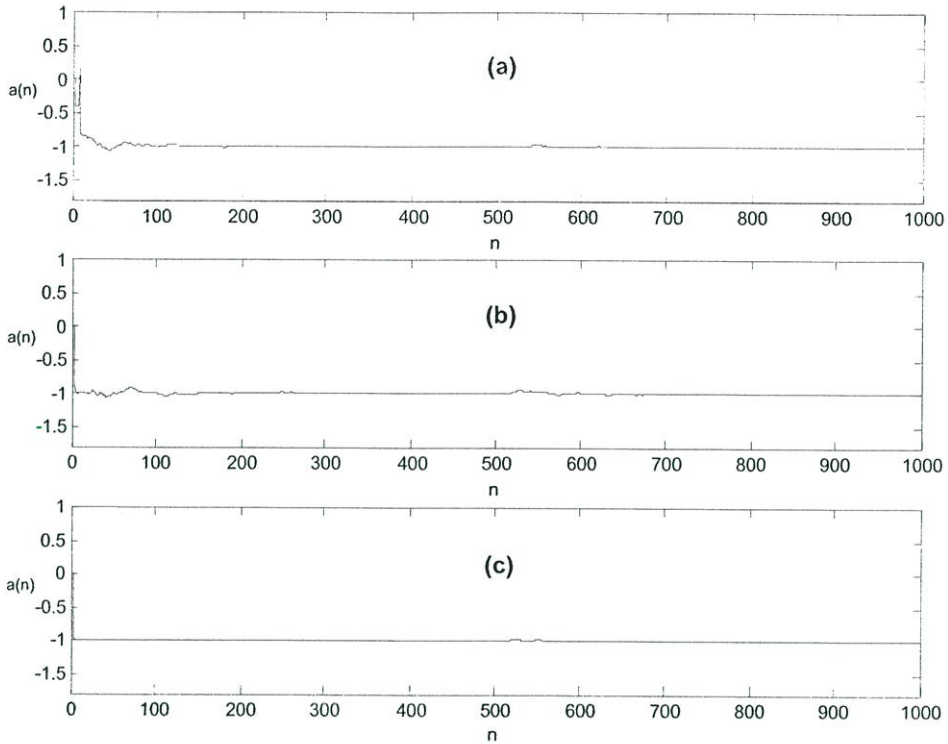
จากรูปที่ 6.8 เมื่อเปลี่ยนค่า ω เป็น $\pi/7$ จะเห็นว่ารูปที่ 6.8 (a) มีความเร็วเข้าสู่ค่าตอบช้าลงเมื่อเทียบกับรูปที่ 6.6 ซึ่งใช้ค่า ω เป็น $\pi/3$ ส่วนรูปที่ 6.8 (b) และ 6.8 (c) มีความเร็วเข้าสู่ค่าตอบที่ช้าลงกว่ารูปที่ 6.6 คือ ประมาณข้อมูลที่ 15 จะเห็นได้ว่า หากค่า ω เปลี่ยนไปในทิศทางที่ลดต่ำลงจะส่งผลให้การหาค่าคำตอบช้าลงด้วย แต่หากเทียบความเร็วเข้าสู่ค่าตอบระหว่างรูปที่ 6.8 (a), 6.8 (b) และ 6.8 (c) ที่ $\omega = \pi/7$ จะเห็นว่าความเร็วเข้าสู่ค่าตอบของรูปที่ 6.8 (c) สูงกว่ารูปที่ 6.8 (a) อย่างเห็นได้ชัดยิ่งขึ้น

6.1.3 การทดสอบอะแดปทีฟอัลกอริทึมในกรณีที่เกิดสัญญาณรบกวนแบบอิมพัลส์

ในหัวข้อนี้จะทำการทดสอบตัวกรองสัญญาณ โดยใช้อัลกอริทึมทั้ง 3 แบบ คือ อัลกอริทึมแบบปรับเสถียรไพไซส์ และ อัลกอริทึมแบบปรับเสถียรไพไซส์ทำงานร่วมกับ อัลกอริทึมแบบเจนเนติกแบบเดิม และ อัลกอริทึมแบบใหม่ที่น่าเสนอ กำหนดสัญญาณอินพุตเป็นไปตามสมการที่ (6.1) และเพิ่มสัญญาณรบกวนแบบอิมพัลส์

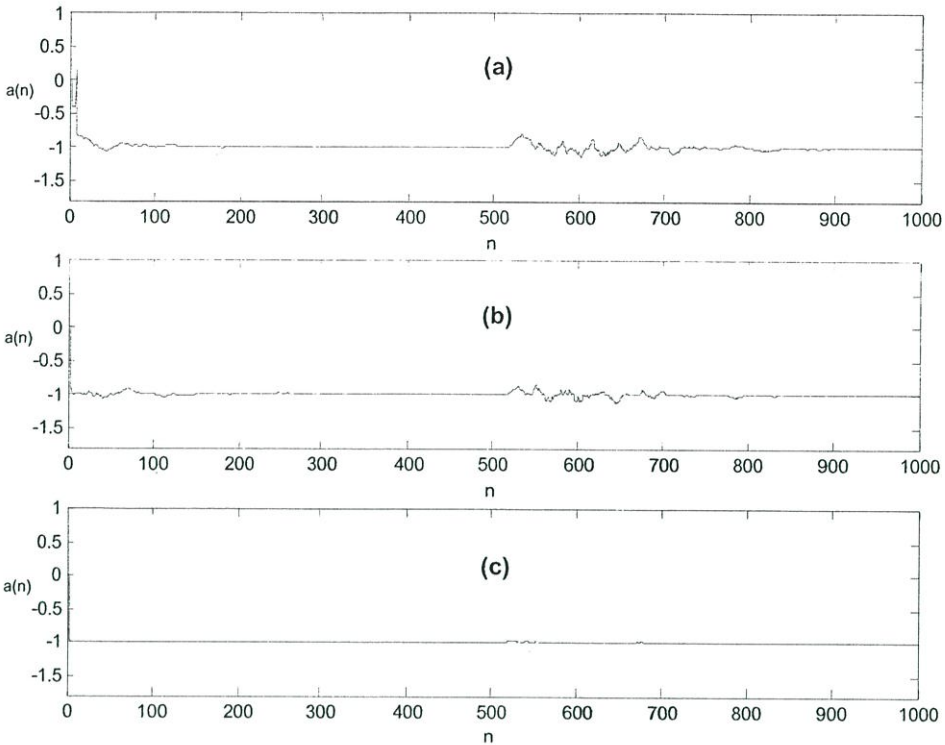
กำหนดพารามิเตอร์ของอัลกอริทึมแบบปรับเสถียรไพไซส์ มีค่าดังนี้ $A = 1$, $\phi = 0$, $\mu(0) = 0.01$, $\omega_0 = \pi/3$, $\rho = 0.9$, $\gamma = 0.99$, $\alpha = 0.99$, $\lambda = 0.001$, SNR = 7 dB และในส่วนของอัลกอริทึม

แบบเจเนติก กำหนดค่าพารามิเตอร์ดังนี้ $P_m = 0.02$, $P_c = 0.08$ และ $\text{Popsiz} = 30$ จากนั้นเพิ่มสัญญาณรบกวนแบบอิมพัลส์เข้าไปที่ตำแหน่ง $n = 500$ โดยมีขนาดที่แตกต่างกัน ผลที่ได้แสดงดังรูปที่ 6.9 ถึง รูปที่ 6.12 ตามลำดับ



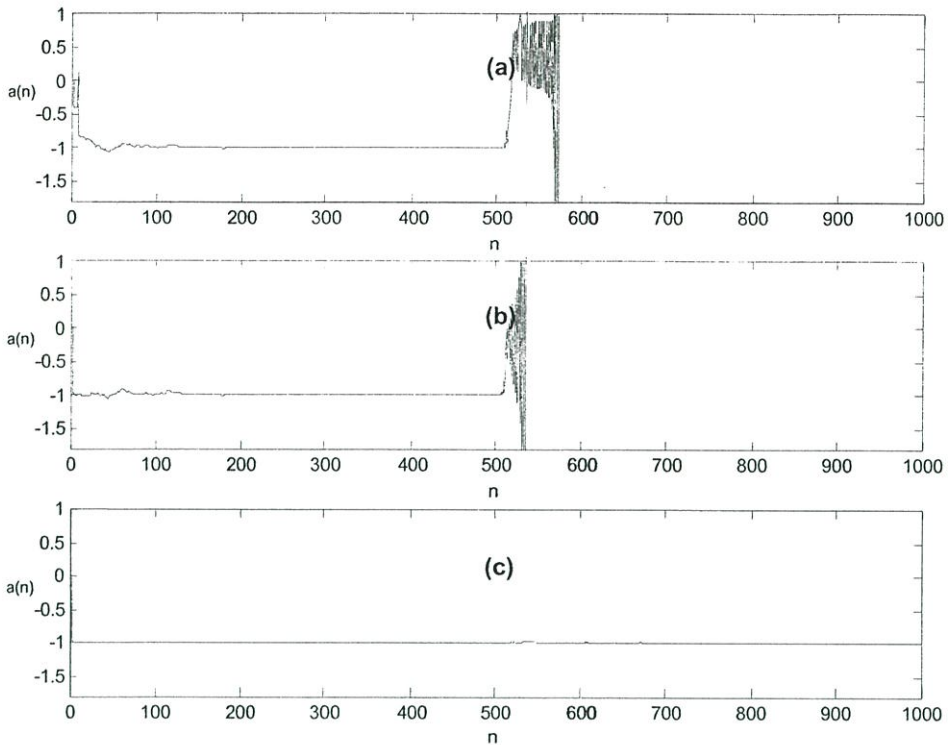
รูปที่ 6.9 เปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ เมื่อเกิดสัญญาณอิมพัลส์ขนาด 30 หน่วยที่ตำแหน่ง $n=500$ (a) อัลกอริทึมแบบเจเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์ [35] (b) อัลกอริทึมแบบเจเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์แบบใหม่ และ (c) อัลกอริทึมใหม่ที่นำเสนอ

จากรูปที่ 6.9 เมื่อเกิดสัญญาณรบกวนแบบอิมพัลส์ที่ตำแหน่ง $n = 500$ ขนาด 30 หน่วย จะเห็นว่าค่าสัมประสิทธิ์ $a(n)$ ของฟิลเตอร์ที่ใช้อัลกอริทึมแบบเจเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์ [35] และ อัลกอริทึมแบบเจเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์แบบใหม่นั้นเกิดความแปรปรวนเล็กน้อย ในขณะที่ อัลกอริทึมแบบใหม่ที่นำเสนอ จะมีความแปรปรวนน้อยกว่า



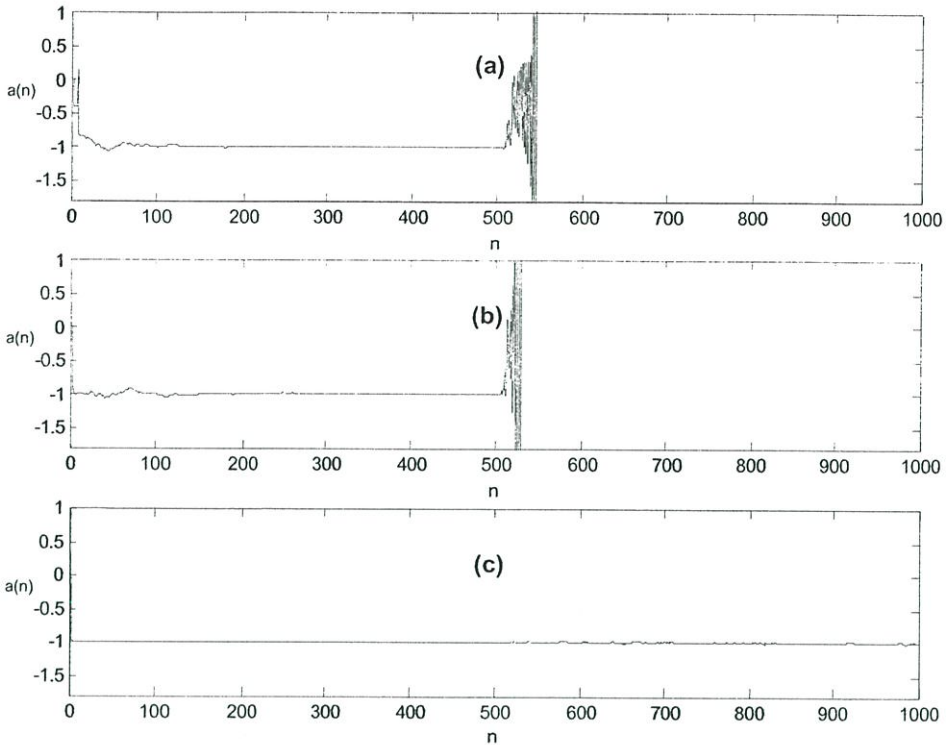
รูปที่ 6.10 เปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ เมื่อเกิดสัญญาณอิมพัลส์ขนาด 50 หน่วยที่ตำแหน่ง $n=500$ (a) อัลกอริทึมแบบเจนเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์ [35] (b) อัลกอริทึมแบบเจนเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์แบบใหม่ และ (c) อัลกอริทึมใหม่ที่นำเสนอ

จากรูปที่ 6.10 เมื่อเกิดสัญญาณรบกวนแบบอิมพัลส์ที่ตำแหน่ง $n = 500$ ขนาด 50 หน่วย จะเห็นว่าค่าสัมประสิทธิ์ $a(n)$ ของฟิลเตอร์ที่ใช้อัลกอริทึมแบบเจนเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์ [35] และ อัลกอริทึมแบบเจนเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์แบบใหม่นั้นเริ่มเกิดความแปรปรวนสูงขึ้นใกล้เคียงกัน ในขณะที่ อัลกอริทึมแบบใหม่ที่นำเสนอ จะมีความแปรปรวนน้อยกว่ามาก



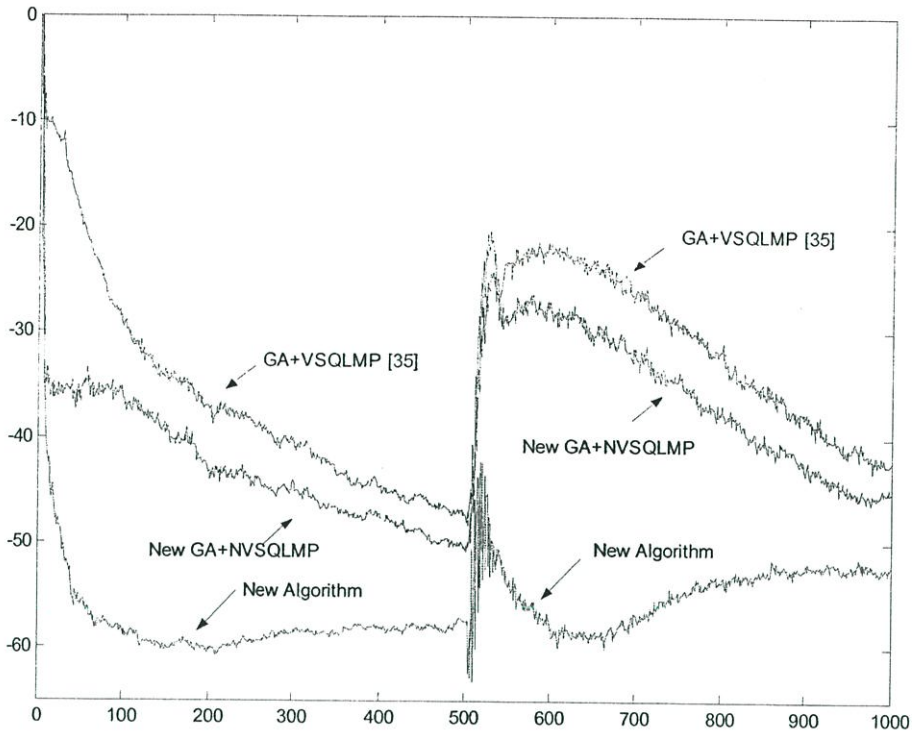
รูปที่ 6.11 เปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ เมื่อเกิดสัญญาณอิมพัลส์ เมื่อเกิด สัญญาณรบกวนแบบอิมพัลส์ขนาด 130 หน่วย ที่ตำแหน่ง $n = 500$ (a) อัลกอริทึมแบบเจนเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์[35] (b) อัลกอริทึมแบบเจนเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์แบบใหม่ และ (c) อัลกอริทึมใหม่ที่นำเสนอ

จากรูปที่ 6.11 แสดงให้เห็นว่า ในกรณีที่เกิดสัญญาณรบกวนแบบอิมพัลส์ที่มีขนาดสูงขึ้น จะทำให้อัลกอริทึมแบบเจนเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์ [35] และ อัลกอริทึมแบบเจนเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์แบบใหม่ หยุดการทำงานได้ ส่วนอัลกอริทึมใหม่ที่นำเสนอสามารถทนสัญญาณรบกวนขนาด 130 หน่วยได้



รูปที่ 6.12 เปรียบเทียบสัมประสิทธิ์ของฟิลเตอร์ $a(n)$ เมื่อเกิดสัญญาณอิมพัลส์ เมื่อเกิด สัญญาณรบกวนแบบอิมพัลส์ขนาด 2,000 หน่วย ที่ตำแหน่ง $n = 500$ (a) อัลกอริทึมแบบเจนเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์[35] (b) อัลกอริทึมแบบเจนเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์แบบใหม่ และ (c) อัลกอริทึมใหม่ที่นำเสนอ

จากรูปที่ 6.12 ทำการทดสอบโดยทำการทดสอบกับสัญญาณรบกวนขนาด 2,000 หน่วย พบว่า อัลกอริทึมแบบใหม่ที่นำเสนอสามารถทนสัญญาณรบกวนถึง 2,000 หน่วยได้ แต่จะพบความแปรปรวนหลังจากเกิดสัญญาณรบกวนขนาดสูง ๆ



รูปที่ 6.13 เปรียบเทียบค่า MSE ของ อัลกอริทึมแบบเจเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียร [35] อัลกอริทึมแบบเจเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรแบบใหม่ และ อัลกอริทึมใหม่ที่นำเสนอ เมื่อเกิดสัญญาณรบกวนแบบอิมพัลส์ 30 หน่วยที่ตำแหน่ง $n = 500$

จากรูปที่ 6.13 ได้แสดงค่า MSE ของแต่ละอัลกอริทึมเมื่อเกิดสัญญาณรบกวนแบบอิมพัลส์ตำแหน่ง $n = 500$ ขนาด 30 หน่วย จะเห็นว่า อัลกอริทึมแบบเจเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียร [35] และอัลกอริทึมแบบเจเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรแบบใหม่ นั้น มีค่า MSE สูงกว่าอัลกอริทึมใหม่ที่นำเสนอ โดยเฉพาะช่วงที่เกิดสัญญาณรบกวนแบบอิมพัลส์ นั้นแสดงให้เห็นว่าอัลกอริทึมแบบใหม่ที่นำเสนอ นอกจากจะให้ค่า MSE ต่ำกว่าแล้ว ยังทนต่อสัญญาณรบกวนแบบอิมพัลส์ได้ดีกว่า อัลกอริทึมแบบเจเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียร [35] และอัลกอริทึมแบบเจเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรแบบใหม่

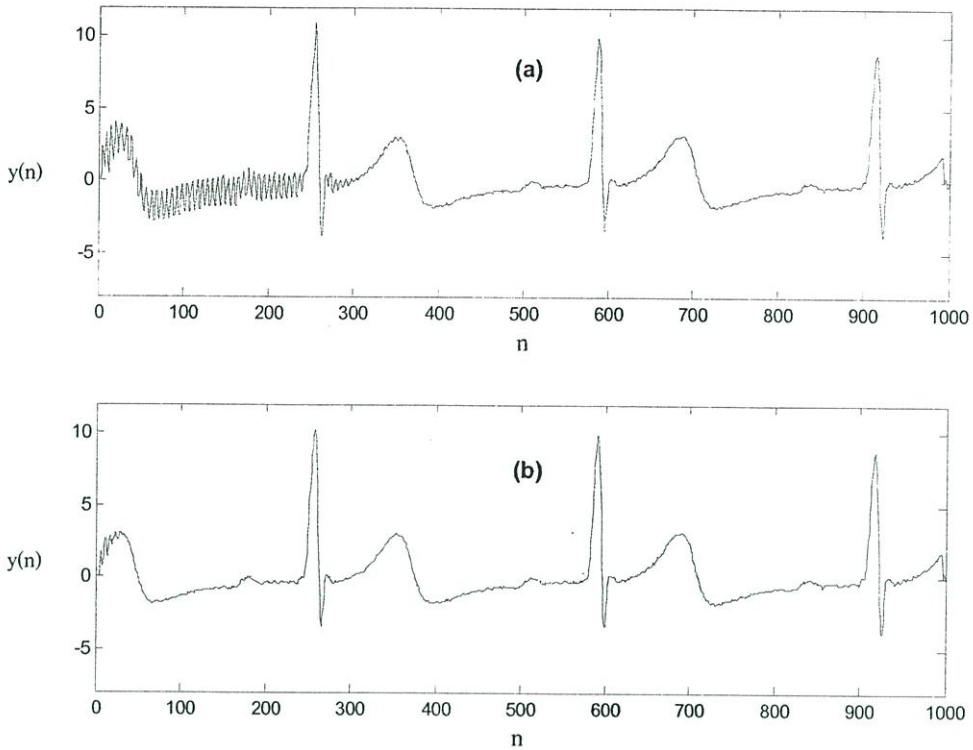
เมื่อเปรียบเทียบผลการทำงานระหว่างอัลกอริทึมแบบเดิม ที่ใช้อัลกอริทึมแบบเจเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียร [35] และ อัลกอริทึมแบบใหม่ที่นำเสนอ พบว่า อัลกอริทึมแบบเดิม [35] เริ่มต้นด้วยการหาค่าเสถียร [35] ด้วยอัลกอริทึมแบบเจเนติกเพียงอย่างเดียว แล้วค่อยเปลี่ยนเป็นอัลกอริทึมแบบปรับเสถียร เมื่อถึงจุดที่ได้ค่าตามที่คาดหวัง ซึ่งการทำงานดังกล่าวถึง

จะให้ความเร็วเข้าสู่คำตอบได้สูง แต่จะมีความแปรปรวนสูงในช่วงแรก และ ช่วงที่เปลี่ยนการทำงานไปเป็นอัลกอริทึมแบบปรับเสถียรไซส์นั้น อัลกอริทึมแบบปรับเสถียรไซส์ จะต้องค่อย ๆ ปรับค่าตัวเองเข้าสู่คำตอบอีกครั้ง จึงพบว่ามีความแปรปรวนช่วงรอยต่อที่เปลี่ยนอัลกอริทึมด้วย สำหรับอัลกอริทึมแบบใหม่ที่น่าเสนอ จะใช้ทั้งอัลกอริทึมแบบเจเนติก และ อัลกอริทึมแบบปรับเสถียรไซส์ทำงานไปพร้อม ๆ กันแบบคู่ขนาน เพื่อเลือกค่าเสถียรไซส์ที่ดีที่สุด ทำให้นอกจากจะได้รับความเร็วเข้าสู่คำตอบที่เร็วจากอัลกอริทึมแบบเจเนติกแล้ว ยังได้ความเรียบ ซึ่งเกิดจากการเลือกค่าเสถียรไซส์ที่ดีที่สุด จากทั้งสองอัลกอริทึมด้วย และ ขณะเดียวกันอัลกอริทึมแบบปรับเสถียรไซส์ มีการปรับตัวเองสู่คำตอบตามไปด้วย ทำให้อัลกอริทึมแบบปรับเสถียรไซส์ทำงานอย่างต่อเนื่อง และให้ค่าสัมประสิทธิ์ที่ดีตามมา

6.2 การประยุกต์ใช้งานอะแดปทีฟอัลกอริทึมที่น่าเสนอโดยนำไปทดสอบการกำจัด

สัญญาณรบกวน 50 เฮิร์ตซ์ ที่ปะปนในสัญญาณคลื่นหัวใจ

ในหัวข้อนี้จะทำการทดสอบตัวกรองสัญญาณ โดยใช้อัลกอริทึมสองแบบ คือ อัลกอริทึมแบบปรับเสถียรไซส์ทำงานร่วมกับอัลกอริทึมแบบเจเนติกแบบเดิม [35] และ อัลกอริทึมแบบใหม่ที่น่าเสนอ ในการทดสอบ สัญญาณอินพุต คือ สัญญาณคลื่นหัวใจที่มีสัญญาณรบกวนคลื่นความถี่ ที่เกิดแหล่งจ่ายพลังงานไฟฟ้า 50 เฮิร์ตซ์ ปะปนอยู่ด้วย โดยกำหนดค่าพารามิเตอร์ต่าง ๆ ดังนี้ $\mu(0) = 0.01$, $\rho = 0.9$, $\gamma = 0.99$, $\alpha = 0.99$, $\lambda = 0.001$, $\text{SNR} = 7 \text{ dB}$, $P_m = 0.02$, $P_c = 0.08$ และ $\text{Popsiz} = 30$



รูปที่ 6.14 สัญญาณคลื่นหัวใจที่ออกจากตัวกรองสัญญาณ (a) อัลกอริทึมแบบปรับเสถียรไปไซส์ทำงานร่วมกับอัลกอริทึมแบบเจเนติกแบบเดิม [35] (b) อัลกอริทึมใหม่ที่นำเสนอ

จากรูปที่ 6.14 แสดงถึงสัญญาณคลื่นหัวใจ (ECG) ที่ได้จากการกรองสัญญาณรบกวน 50 เฮิรตซ์ โดย อัลกอริทึมทั้งสองแบบ คือ อัลกอริทึมแบบเจเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไปไซส์ [35] และ อัลกอริทึมแบบใหม่ที่นำเสนอ ตามลำดับ จะเห็นว่า อัลกอริทึมแบบเจเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไปไซส์ [35] นั้น เมื่อทำงานได้สักระยะหนึ่ง หรือ ประมาณข้อมูลที่ 250 จึงจะกำจัดสัญญาณรบกวนได้หมด แต่ อัลกอริทึมแบบใหม่ที่นำเสนอนี้ จะเริ่มกำจัดสัญญาณรบกวนได้เร็วกว่า คือ สามารถกำจัดสัญญาณรบกวนได้หมด ตั้งแต่ช่วงข้อมูลประมาณที่ 40

6.3 สรุป

ในบทนี้ ได้นำเสนอผลการทดสอบอัลกอริทึมสำหรับตัวกรองสัญญาณแบบอะแดปทีฟนอซ์ฟิลเตอร์ เพื่อประมาณสัญญาณชาวน์ที่ไม่ทราบความถี่ โดยในการทดสอบแรก เป็นการนำอัลกอริทึมทั้งสามแบบ คือ อัลกอริทึมแบบเจเนติกทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไปไซส์ [35] อัลกอริทึมแบบเจเนติกแบบใหม่ทำงานร่วมกับอัลกอริทึมแบบปรับเสถียรไปไซส์แบบใหม่ และ อัลกอริทึมแบบใหม่ที่นำเสนอ ซึ่งเป็นอัลกอริทึมที่เกิดจากการทำงานร่วมกันแบบคู่ขนานของอัลกอริทึม

แบบปรับเสถียรไซส์กับอัลกอริทึมแบบเจเนติก ในการกำจัดสัญญาณรบกวนประเภทเกาส์เซียน ซึ่งผลที่ได้นั้นจะเห็นว่า อัลกอริทึมใหม่ที่น่าเสนอ มีคุณสมบัติที่ดีกว่า อัลกอริทึมแบบอื่น กล่าวคือ ในด้านความเร็วในการหาคำตอบสูงกว่า และค่า MSE ต่ำกว่า นอกจากนี้การเปลี่ยนแปลงค่าเสถียรไซส์เริ่มต้น แทบจะไม่มีผลต่อการทำงานของอัลกอริทึมใหม่ที่น่าเสนออีกด้วย ต่อจากนั้น เป็นการทดสอบเมื่อเกิดสัญญาณรบกวนแบบอิมพัลส์ ผลที่ได้ คือ อัลกอริทึมแบบใหม่ที่น่าเสนอนั้น มีความทนทานต่อสัญญาณรบกวนแบบอิมพัลส์ได้ดีกว่าอัลกอริทึมแบบเดิมที่นำมาทดสอบ และสุดท้ายเป็นการทดสอบโดยการประยุกต์นำอัลกอริทึมที่น่าเสนอไปใช้งาน โดยกำจัดสัญญาณรบกวนในสัญญาณคลื่นหัวใจ ที่มีสัญญาณรบกวนที่เกิดแหล่งจ่ายพลังงานไฟฟ้า (Power Line) 50 เฮิร์ตซ์ ปะปนอยู่ด้วย อัลกอริทึมแบบใหม่ที่น่าเสนอนี้ สามารถกำจัดสัญญาณรบกวนได้ดีกว่า อัลกอริทึมแบบเดิม จากที่ได้ทำการทดสอบอัลกอริทึมทั้งสามแบบ จะเห็นว่า อัลกอริทึมแบบใหม่ที่น่าเสนอ ซึ่งเกิดจากการทำงานร่วมกันแบบคู่ขนานของอัลกอริทึมแบบปรับเสถียรไซส์ กับ อัลกอริทึมแบบเจเนติกนั้น จะมีคุณสมบัติที่ดีกว่าอัลกอริทึมแบบเดิม ในด้านความเร็วในการหาคำตอบ ค่า MSE ต่ำกว่า และ มีความทนทานต่อสัญญาณรบกวนแบบอิมพัลส์ที่สูงอีกด้วย

สรุปและวิจารณ์ผลการทดลอง

ในวิทยานิพนธ์นี้ได้นำเสนออะแดปทีฟอัลกอริทึมใหม่ คือ อัลกอริทึมที่เกิดจากการทำงานร่วมกันของอัลกอริทึมแบบเจนเนติกกับอัลกอริทึมแบบปรับเสถียรไซส์แบบคู่ขนาน โดยอัลกอริทึมที่นำเสนอในวิทยานิพนธ์ฉบับนี้ มีข้อดีตรงที่มีความเร็วผู้คำนวณสูงกว่า อะแดปทีฟอัลกอริทึมแบบเดิม นอกจากนี้ยังมีค่า MSE ต่ำ โดยเฉพาะอย่างยิ่งทนทานต่อสัญญาณรบกวนแบบอิมพัลส์ได้ดี เหตุที่อัลกอริทึมที่นำเสนอในวิทยานิพนธ์นี้ มีความเร็วการเข้าสู่ค่าตอบสูง เพราะจะใช้ข้อดีของทั้งสองอัลกอริทึม คือ อัลกอริทึมแบบปรับเสถียรไซส์ กับอัลกอริทึมแบบเจนเนติก มาช่วยกันหาค่าเสถียรไซส์ที่เหมาะสม ดังนั้นค่าเสถียรไซส์ที่ได้จะทำให้หาค่าตอบได้เร็ว ความแปรปรวนต่ำ และทนต่อสัญญาณรบกวนแบบอิมพัลส์ได้ดี เปรียบเสมือนการรวมข้อดีของ อัลกอริทึมแบบเจนเนติก และอัลกอริทึมแบบปรับเสถียรไซส์เข้าด้วยกันนั่นเอง

จากผลการทดลองที่ใช้โปรแกรม MATLAB และ ภาษา C นั้น จะเห็นได้ว่า การประมาณค่าสัญญาณขาขึ้นคลื่นเดี่ยว ที่ถูกสัญญาณรบกวนแบบเกาส์เซียน และ สัญญาณรบกวนแบบอิมพัลส์ อัลกอริทึมที่นำเสนอในวิทยานิพนธ์นี้มีความเร็วเข้าสู่ค่าตอบสูงกว่า มีความทนทานมากกว่า และค่า MSE ต่ำกว่าอีกด้วย ส่วนในกรณีที่น่าจะเอาเดปทีฟฟอสเตอร์ ไปประยุกต์ใช้กับการกำจัดสัญญาณรบกวน 50 เฮิร์ตซ์ ที่เกิดจากการเหนี่ยวนำของไฟฟ้ากระแสสลับที่ปะปนมากับสัญญาณคลื่นหัวใจ (ECG) ในขณะที่ทำการบันทึก อัลกอริทึมใหม่ที่น่าเสนอนี้สามารถกำจัดสัญญาณรบกวนได้ดีกว่าเช่นกัน

ข้อสังเกตค่าเสถียรไซส์เริ่มต้น หรือ ค่าเสถียรไซส์สูงสุดนั้น จะมีผลต่ออัลกอริทึมแบบปรับเสถียรไซส์เป็นอย่างมาก ซึ่งในการทดสอบจะเห็นว่า ถ้าค่าเสถียรไซส์เริ่มต้นมีค่ามาก จะทำให้อัลกอริทึมแบบปรับเสถียรไซส์ หาค่าตอบได้เร็วขึ้น แต่ในกรณีที่ค่าเสถียรไซส์เริ่มต้นมีค่ามากเกินไป จะทำให้เกิดความแปรปรวนสูง ในวิทยานิพนธ์นี้จึงไม่กล่าวถึงวิธีการเลือกค่าเสถียรไซส์เริ่มต้นที่เหมาะสม เพราะ ค่าเสถียรไซส์เริ่มต้นจะมีผลน้อยมากในอัลกอริทึมที่นำเสนอนี้ อีกส่วนที่สำคัญ คือ การกำหนดฟังก์ชันความเหมาะสมของ อัลกอริทึมแบบเจนเนติกต้องกำหนดอย่างเหมาะสม นอกจากนี้ อัลกอริทึมแบบเจนเนติก ยังสามารถประยุกต์ใช้งานได้อีกมาก เพราะการทำงานของอัลกอริทึมแบบเจนเนติกนั้น ขึ้นอยู่กับการสุ่มตัวเลข ซึ่งเป็นส่วนที่ต้องศึกษาและพัฒนาต่อไป ส่วนข้อเสียของอัลกอริทึมแบบเจนเนติก คือ การคำนวณซึ่งการคำนวณนั้น จะมากกว่าอัลกอริทึมแบบปรับเสถียรไซส์เป็นจำนวนเท่าของโครโมโซม เพราะว่า ต้องนำโครโมโซมทุกตัวมาหาค่าความเหมาะสมก่อน เนื่องจากทฤษฎีเกี่ยวกับตัวกรอง IIR ยังอยู่ในขั้นของการวิจัย ยังไม่มีสูตรสำเร็จเหมือน FIR เพราะ IIR นั้นทำงานในลักษณะ Nonlinear ซึ่งทำนายพฤติกรรมได้ค่อนข้างยาก ต้องอาศัยการประมาณจึง

ต้องใช้ความรู้ด้านคณิตศาสตร์ และประสบการณ์ ดังนั้นในการวิเคราะห์ IIR ฟิวเตอร์นั้นจะต้องนำคอมพิวเตอร์เข้ามาช่วยในการจำลองการทำงานเช่นเดียวกับอัลกอริทึมที่ได้นำเสนอในวิทยานิพนธ์นี้

เอกสารอ้างอิง

- [1] S. Phuvasitkul, C. Benjangkprasert, and N. Anantrasirichai, “A new class of gradient-based algorithm by using variable step-size technique”, **2002 6th International conference on signal processing proceedings**, Aug. 2002, pp. 85-88.
- [2] S. Phuvasitkul, C. Benjangkprasert, N. Anantrasirichai and S. Jorphochaudom, “A new gradient-based algorithm-based algorithm using variable step-size technique and its applicatio”, **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 error criterion”, **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 approach a new learning algorithm for adaptive IIR filtering”, **IEEE Signal processing magazine**, Nov 1996, pp. 38-46.
- [5] R. Puchalard, P. chumchu, C. Benjangkprasert, N. Anantrasirichai, O. Sangaroon and K. Jenjerapongvej “The reduction of gradient noise in gradient – based algorithm by using variable step-size technique”, **IEEE Asia-Pacific conference on Circuit and System**, Dec. 2000, pp. 415-418.
- [6] กิติ ไพฑูริย์วัฒนกิจ กาญจนี วงศ์วิภาพร “การปรับปรุงสมรรถนะ จีเนติก อัลกอริทึม” วารสารสำนักงานคณะกรรมการวิจัยแห่งชาติ ปีที่ 27 เล่มที่ 1 มกราคม-มิถุนายน 2538 หน้า 1-33
- [7] M H. Hayes, **Statistical Digital Signal Processing and Modeling**, Canada : John Wiley & Sons, 1996.
- [8] A. V. Oppenheim and R. W. Schaffer, **Digital Signal Processing**, Prentice-Hall. 1975.
- [9] R. M. Mersereau and M. J. T. Smith, **Digital Filtering A Computer Laboratory Textbook**. John Wiley & Sons., 1994, pp. 59-61.
- [10] Robert D. Strum and Donald E. Kirk, **First Principles of Discrete System and Digital Signal Processing**, Addison-Wesley, 1988.
- [11] L. C. Ludeman, **Fundamentals of Digital Signal Processing**, John wiley & Sons, 1987.
- [12] J. G. Proakis and D. G. Manolakis, **Digital Signal Processing Principles : Algorithm and Applications**, Prentice-Hall, 1996.

- [13] R. M. Mersereau and M. J.T. Smith, **Digital filtering**, John Wiley & Sons, 1994.
- [14] E. C. Lfeachor and B. W. Jervis, **Digital signal processing a practical approach**, **Addison-Wesley Publishing Company**, 1996, pp.255-257.
- [15] A. V. Oppenheim and R. W. Schaffer, **Discrete-time signal processing**, Prentice-Hall, 1999.
- [16] A. Anehorai, "A minimum 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", **IEEE Trans. Acoustics Speech and Signal Processing** vol. ASSP-36, no. 6, June 1988, pp.911-919.
- [19] D. G. Manolakis, V. K. Ingle and S. 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. Schroeder, Rao Yarlagadda and 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] สถาบันบัณฑิตวิทยาศาสตร์และเทคโนโลยีไทย สวทช. พันธุกรกรรม และการถ่ายทอดลักษณะทางพันธุกรรม [Online]. Available :
<http://www.learn.in.th/gmos/sample/chapter1/chapter1.3.html>. 2550.
- [23] L. Ljung, "Asymptotic gain and search direction for recursive identification algorithms", **Proc. 19th IEEE Trans. on automatic control**, Dec. 1980, pp. 981-985.
- [24] B. Widrow and S. D. Stearns, **Adaptive Signal Processing**, New Jersey : 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] R. Panchalard *et. al.*, "Simplify adaptive IIR notch filter based on least mean p -power error criterion", **Proc. IEEE APCCAS**, Nov. 1988, pp. 335-338.
- [28] ราชู พันธุ์ฉลาด และคณะ. "อะแดปทีฟ IIR นอตช์ตัวกรองบนพื้นฐานการพิจารณาค่าผิดพลาดเฉลี่ยกำลัง p น้อยที่สุดโดยใช้วิธีการควอนไทซ์เกรเดียนต์" การประชุมวิชาการวิศวกรรมไฟฟ้า ครั้งที่ 21 ภาควิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี พฤศจิกายน 2541 หน้า 122-125
- [29] ราชู พันธุ์ฉลาด "อะแดปทีฟอัลกอริทึมสำหรับ IIR นอตช์ฟิลเตอร์ด้วยวิธีเกรเดียนต์และการประยุกต์ใช้งาน" วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชา วิศวกรรมไฟฟ้า บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 2543
- [30] ไพศาล เหล่าสุวรรณ พันธุศาสตร์ ไทยวัฒนาพานิช 2535
- [31] ชิดชนก เหลือสินทรัพย์ "คอมพิวเตอร์กับการแก้ปัญหาโดยวิธีพันธุกรรม" ไมโครคอมพิวเตอร์ ฉบับที่ 97 หน้า 192-198
- [32] J. H. Holland, **Genetic Algorithms**, Scientific American, July 1992, pp. 45-50.
- [33] Z. Michalewicz, **Genetic algorithms + data structures = Evolutionary programs**, Springer-Verlag, Berlin, 1992.
- [34] M. Morrow, "Genetic algorithms". **Dr. Dobb's Journal**, April 1991, pp. 26-32.
- [35] สิริศักดิ์ ภูวสิทธิ์กุล "เกรเดียนต์อัลกอริทึมแบบใหม่สำหรับอะแดปทีฟ IIR นอตช์ฟิลเตอร์โดยใช้อัลกอริทึมแบบเจเนติกร่วมกับอัลกอริทึมแบบปรับเสถียรไซส์" วิทยานิพนธ์ วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 2544
- [36] นางสาวปริศนา แซ่มสุขชี, 2547 "เอกสารสัมมนาทางคอมพิวเตอร์ Genetic Algorithm (GA)" ใน วิชา 310492 สัมมนาทางคอมพิวเตอร์ คณะวิทยาศาสตร์ จันทบุรี : มหาวิทยาลัยบูรพา

ภาคผนวก

ภาคผนวก ก.

บทความวิจัยที่ได้รับการตีพิมพ์

- [1] S. Jorphochaudom, C. Benjangkprasert, S. Teerasakworakun and K. Janchitrapongvej. "Adaptive lattice step-size algorithm for narrowband interference suppression in DS/CDMA system", Proceeding of International Conference on Control, Automation and Systems, pp. 2087-2089, Oct. 2003.

ประวัติผู้เขียน

นางสาวศรินทร์พร จอโกษาอุดม เกิดเมื่อวันที่ 26 สิงหาคม 2523 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษา วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ จาก มหาวิทยาลัยเทคโนโลยีสุรนารี ปีการศึกษา 2544 และเข้าศึกษาต่อในหลักสูตร วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมสารสนเทศ ที่ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2545

ปัจจุบันทำงานที่ บริษัท ไปรษณีย์ไทย จำกัด