

การเข้ารหัสลับของสัญญาณเสียงโดยใช้สัญญาณเสียงเคออสติก

Voice Encryption Using Discrete Chaotic Signal



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่.....

เลขทะเบียน.....50273.....

วัน,เดือน,ปี. 2 8 เม.ย. 2547

ปีการศึกษา 2545

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Voice Encryption Using Discrete Chaotic Signal



**A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENT FOR THE DEGREE OF
BACHELOR IN DEPARTMENT OF INFORMATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

2002

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์

การเข้ารหัสลับของสัญญาณเสียงโดยใช้สัญญาณเสียงเคออสติกแบบดิสครีต

Title

Voice Encryption Using Discrete Chaotic Signal

นักศึกษา

1.นายจักรพันธ์ รัชมียุงทอง รหัสนักศึกษา 42010053
2.นายวรวิทย์ ธรรมาเจริญราช รหัสนักศึกษา 42010310

อาจารย์ผู้ควบคุมปริญญานิพนธ์

ผศ. ดร. ปิติเชต สุรักษา

ระดับการศึกษา

อาจารย์กฤดากร กล่อมการ
ปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชา

สาขาวิศวกรรมสารสนเทศ

ปีการศึกษา

วิศวกรรมสารสนเทศ

2545

ปริญญานิพนธ์ฉบับนี้ ได้รับการอนุมัติให้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตร
วิศวกรรมศาสตรบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร
ลาดกระบัง


.....
(อาจารย์กฤดากร กล่อมการ)
อาจารย์ผู้ควบคุมปริญญานิพนธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์

การเข้ารหัสลับของสัญญาณเสียงโดยใช้สัญญาณเสียงเคออสติกแบบดิสครีต

นักศึกษา

1. นายจักรพันธ์ รัศมีบุงทอง รหัสนักศึกษา 42010053
2. นายวรวิทย์ ธรรมาเจริญราช รหัสนักศึกษา 42010310

อาจารย์ผู้ควบคุมปริญญานิพนธ์ ผศ. ดร. ปิติเขต สุรักษา

ระดับการศึกษา

อาจารย์กฤดากร กล่อมการ
ปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมสารสนเทศ

ภาควิชา

วิศวกรรมสารสนเทศ

ปีการศึกษา

2545

บทคัดย่อ

โปรเจกต์นำเสนอเกี่ยวกับการรักษาความปลอดภัยในการติดต่อสื่อสารบนพื้นฐานของดิสครีตไดนามิก, วงจรการเข้ารหัสสัญญาณเสียง โดยใช้ 2 dimension (Henon map) ในการสร้าง ซึ่งใช้เทคนิคการ synchronize และ การใช้คีย์พารามิเตอร์ที่ได้ศึกษามา ซึ่งจะแสดงการส่งสัญญาณจริงที่ทำการเข้ารหัสบนโทรศัพท์และประเมินคุณภาพของเสียงที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

THESIS TITLE	Voice Encryption Using Discrete Chaotic Signal		
STUDENT	Mr. Jukkraphun Ratsameeyoongthong	No. 42010053	
	Mr. Worawit Thammacharoenrach	No. 42010310	
ADVISOR	Asst. Prof. D.r. Pitikhate Sooraksa		
	Mr. Kiddakorn Klomkarn		
COURSE	Bachelor Degree of Information Engineering		
DEPARTMENT	Information Engineering		
YEAR	2002		

Abstract

This project present secure communication based on discrete dynamic ,Voice encryption circuit by using 2 dimension (Henon map) are constructed. The synchronization technique and key parameter are also study ,Finally the demonstrated in real transmission encryption signal over telephone and voice quality are evaluated.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ ทางผู้จัดทำได้รับความอนุเคราะห์ทางด้านต่าง ๆ ทั้งคำปรึกษาทางด้านวิชาการ และคำแนะนำในการปฏิบัติงาน รวมถึงความช่วยเหลือทางด้านเครื่องมือ และอุปกรณ์ต่าง ๆ จากอาจารย์กฤตดากร กล่อมการ จนกระทั่งสำเร็จเป็นปริญญานิพนธ์ฉบับนี้

สุดท้ายนี้ขอกราบขอบพระคุณ คุณพ่อ คุณแม่ ที่คอยห่วงใยและให้การสนับสนุนด้านการศึกษาเป็นอย่างดีมาโดยตลอด รวมทั้งขอบคุณเพื่อน ๆ นักศึกษาทุกคนที่ช่วยเหลือให้คำแนะนำทางด้านวิชาการ และการดำเนินงานในหลายๆ ด้าน

คุณประโยชน์ที่เกิดจากปริญญานิพนธ์ฉบับนี้ ผู้จัดทำขอบอບแด่ผู้มีพระคุณทุกท่าน

นายจักรพันธ์ รัศมีขุมทอง

นายวรวิทย์ ธรรมาเจริญราช

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย

บทคัดย่อภาษาอังกฤษ

สารบัญ

สารบัญรูป

บทที่ 1 บทนำ

1.1 แนวคิดและที่มา	1
1.2 จุดประสงค์	1
1.3 ขอบเขตของโครงการ	1
1.4 ผลที่คาดว่าจะได้รับ	1

บทที่ 2 ทฤษฎีและหลักการที่ใช้ในโครงการ

2.1 ทฤษฎี Chaos	2
2.1.1 The Logistic map	2
2.1.2 Bifurcation Diagram	3
2.1.3 Hyper Chaos	4
2.2 Java	7
2.2.1 Introducing Java	7
2.2.2 Java Virtual Machine	8
2.2.3 Java Developer Kit (JDK)	9
2.2.4 JDK Installation	9
2.2.5 Creating Java Application	10
2.2.6 Creating Java Applet	11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
บทที่ 3 การออกแบบ	
3.1 Software	13
3.1.1 การทดลองการเข้ารหัสโดยใช้สัญญาณ chaotic จากสมการ logistic โดยโปรแกรม Matlab	14
3.1.2 การทดลองการเข้ารหัสโดยใช้สัญญาณ chaotic จากสมการ Hyper Henon โดยโปรแกรม Matlab	16
3.1.3 การสร้างโปรแกรมที่ใช้แสดงสัญญาณของสมการ Logistic ในรูป Applet	18
3.1.4 การสร้างโปรแกรมที่ใช้แสดงสัญญาณของสมการ Hyper Henon ในรูป Applet	19
3.2 Hardware	19
3.2.1 Logistic Block Diagram	19
3.2.1.1 วงจรคูณสัญญาณ	20
3.2.2 Hyper Henon Block Diagram	21
บทที่ 4 ผลการทดลอง	
4.1 Software	
4.1.1 ผลการทดลองจากโปรแกรมสร้างสัญญาณจากสมการ Logistic โดยใช้ Java	22
4.2 ผลการทดลองจากโปรแกรมสร้างสัญญาณจากสมการ Hyper Henon โดยใช้ Java	23
4.2 Hardware	
4.2.1 ผลการทดลองวงจรสร้างสัญญาณ Chaos แบบ logistic ของวงจรตามรูป 3-9	24
4.2.1 ผลการทดลองวงจรสร้างสัญญาณ Chaos แบบ Hyper Henon ของวงจรตามรูป 3-12	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

บทที่ 5 สรุปผลการทดลอง

5.1 สรุปผลการทดลอง

28

5.2 ปัญหาที่เกิดขึ้นในการทดลอง

28

5.3 แนวทางในการพัฒนา

28

บรรณานุกรม

ภาคผนวก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

	หน้า
รูปที่ 2-1 แสดง การเปลี่ยนแปลง parameter โดยการเพิ่มค่า r จาก 3.3 เป็น 3.8	3
รูปที่ 2-2 แสดง Bifurcation Diagram ของสมการ Logistic ที่ r ระหว่าง 0 ถึง 4	3
รูปที่ 2-3 แสดง การเปลี่ยนแปลง parameter โดยการเพิ่มค่า r จาก 0.5 เป็น 1.79	5
รูปที่ 2-4 แสดง Bifurcation Diagram ของสมการ Hyper Henon ที่ r ระหว่าง 0 ถึง 4	6
รูปที่ 3-1 แสดง Block Diagram การเข้ารหัสลับ	13
รูปที่ 3-2 แสดง Original Signal	15
รูปที่ 3-3 แสดง Coded Signal [Logistic]	15
รูปที่ 3-4 แสดง Recovered Signal	15
รูปที่ 3-5 แสดง Original Signal	17
รูปที่ 3-6 แสดง Coded Signal [Hyper Henon]	17
รูปที่ 3-7 แสดง Recovered Signal	17
รูปที่ 3-8 แสดง logistic block diagram	19
รูปที่ 3-9 วงจร logistic map	19
รูปที่ 3-10 วงจร คูณ	20
รูปที่ 3-11 แสดง hyper henon block diagram	21
รูปที่ 3-12 แสดง hyper henon map	21
รูปที่ 4-1 ตัวอย่างโปรแกรม Logistic Signal เมื่อมีการปรับค่า $r=3.68$ โดย plot ทุก 100 ms โดย scan ใน mode single-scan และ plot ใน mode line graph	22
รูปที่ 4-2 ตัวอย่างโปรแกรม Hyper Henon Signal เมื่อมีการปรับค่า $r=1.5$ โดย plot ทุก 100 ms โดย scan ใน mode single-scan และ plot ใน mode line graph	23
รูปที่ 4-3 รูปสัญญาณจากวงจร logistic เมื่อกำหนด $R_F = 2.5 \text{ k}\Omega$ โดยค่า $R_i = 975 \Omega$, $r = 3.57$	24
รูปที่ 4-4 รูปสัญญาณจากวงจร logistic เมื่อกำหนด $R_F = 2.5 \text{ k}\Omega$ โดยค่า $R_i = 960 \Omega$, $r = 3.6$	24
รูปที่ 4-5 รูปสัญญาณจากวงจร logistic เมื่อกำหนด $R_F = 2.5 \text{ k}\Omega$ โดยค่า $R_i = 925 \Omega$, $r = 3.7$	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ(ต่อ)

หน้า

รูปที่ 4-6 รูปสัญญาณจากวงจร logistic	
เมื่อกำหนด $R_F = 2.5\text{ k}\Omega$ โดยค่า $R_i = 890\Omega$, $r = 3.8$	25
รูปที่ 4-7 รูปสัญญาณจากวงจร Hyper Henon เมื่อกำหนด $V=-1.8$	25
รูปที่ 4-8 รูปสัญญาณจากวงจร Hyper Henon เมื่อกำหนด $V=-1.5$	26
รูปที่ 4-9 รูปสัญญาณจากวงจร Hyper Henon เมื่อกำหนด $V=-1.2$	26
รูปที่ 4-10 รูปสัญญาณจากวงจร Hyper Henon เมื่อกำหนด $V=-1.05$	26
รูปที่ 4-11 รูปสัญญาณจากวงจร Hyper Henon เมื่อกำหนด $V=-0.94$	27
รูปที่ 4-12 รูปสัญญาณจากวงจร Hyper Henon เมื่อกำหนด $V=-0.67$	27
รูปที่ 4-13 รูปสัญญาณจากวงจร Hyper Henon เมื่อกำหนด $V=-1.09$	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 แนวความคิดและที่มา

ในปัจจุบันความสำคัญในด้านความปลอดภัยได้รับความสนใจเป็นอย่างมาก ซึ่งจะเห็นได้จากการนำเสนอรูปแบบในการเข้ารหัสสัญญาณต่างๆ แต่ในการเข้ารหัสสัญญาณไม่ได้มีแค่ในรูปแบบของโปรแกรมการเข้ารหัสเท่านั้น แต่รวมไปถึงการเข้ารหัสที่ซ่อนสัญญาณด้วย ซึ่งรูปแบบของโครงการนี้เป็นการนำ สัญญาณ Chaotic ที่สร้างจากสมการ Logistic มาประยุกต์ใช้จริง โดยนำมาแสดงในรูปของวงจรอิเล็กทรอนิกส์

1.2 จุดประสงค์

เพื่อศึกษาค้นคว้าและทดลองสร้างอุปกรณ์การเข้ารหัสสัญญาณเสียงโดยใช้สัญญาณเคอสติกแบบดิสครีต

1.3 ขอบเขตของโครงการ

- 1.3.1 สร้างสัญญาณ Chaotic จากสมการ logistic โดยวงจรอิเล็กทรอนิกส์
- 1.3.2 ทดลองการ Synchronize สัญญาณ Chaotic
- 1.3.3 นำสัญญาณที่ได้ไปใช้งาน Encryption

1.4 ผลที่คาดว่าจะได้รับ

สร้างวงจรเข้ารหัสลับของสัญญาณเสียงโดยสัญญาณ Chaotic แบบ Discrete

บทที่ 2

ทฤษฎีและหลักการที่ใช้ในโครงการ

2.1 ทฤษฎี Chaos

Chaos คือระบบ ไม่เชิงเส้นที่เราสามารถหาค่าได้ยาก เมื่อเราเปลี่ยนค่าพารามิเตอร์ สัญญาณที่ได้จะเปลี่ยนไปไม่สามารถหาค่าได้ จึงเหมาะแก่การนำไปใช้ในการรักษาความปลอดภัยของข้อมูล ในการสร้างสัญญาณ chaotic นั้น เราสามารถสร้างได้จากสมการ Logistic

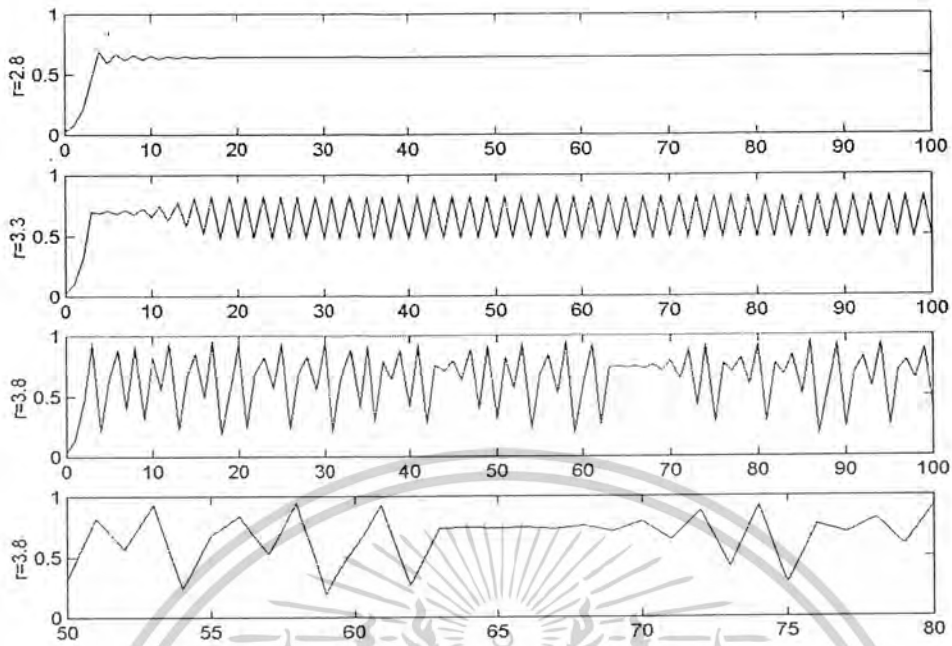
2.1.1 The Logistic map

สมการ Logistic map เป็นสมการแบบ ไม่เชิงเส้น ซึ่ง output คือ (x_{n+1}) ขึ้นกับ x_n โดยสมการแบบ logistic เป็นลักษณะ recursive ซึ่งหมายความว่าเทอมที่สามเป็นฟังก์ชันของตัวที่สอง , เทอมที่สี่เป็นฟังก์ชันของตัวที่สาม ไปเรื่อยๆ โดยแสดงสมการได้ ดังนี้

$$x_{n+1} = rx_n(1 - x_n) \quad (2.1)$$

r คือค่าคงที่ เมื่อเปลี่ยนค่า r ไปเรื่อยๆ ผลจากกราฟ เมื่อค่า $r = 2.8$ กราฟที่ได้จะเป็นแบบ damped harmonic oscillator ตัวอย่าง เช่น การแกว่งของลูกตุ้ม ซึ่งลูกตุ้มก็จะแกว่งช้าลงๆ จนหยุด สำหรับ $r = 3.3$ สัญญาณจะเป็นสัญญาณ oscillator โดยไม่มีการ damping , oscillator นี้จะดำเนินต่อเนื่องไปไม่มีที่สิ้นสุด

เมื่อค่า r มากกว่า 3.3 Logistic map จะเริ่มแสดงปรากฏการณ์ Chaotic ดูจากกราฟ ข้างล่างจะเห็นความแตกต่างของจุด peak และการแตกต่างกันเล็กน้อยของกราฟ โดยสมการจะให้ความแตกต่างกันของรูปสำหรับค่า x_n ใดๆ

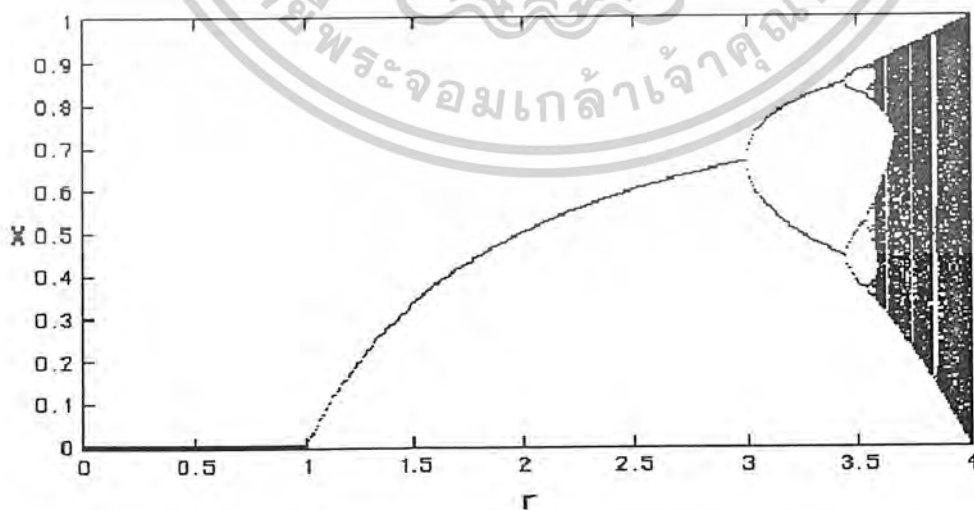


รูปที่ 2-1 แสดง การเปลี่ยนแปลง parameter โดยการเพิ่มค่า r จาก 3.3 เป็น 3.8

2.1.2 Bifurcation Diagram

Bifurcation คือ คาบที่เพิ่มขึ้นเป็นเท่าตัว การเปลี่ยนจาก N -point attractor เป็น $2N$ -point attractor ซึ่งเกิดเมื่อการควบคุมพารามิเตอร์นั้นเปลี่ยน

Bifurcation Diagram คือ การมองผลสรุปของการสร้างคาบที่เป็น 2 เท่าเมื่อมีการเพิ่มค่า r ไปเรื่อยๆ โดยรูปเป็นการแสดง bifurcation diagram ของ logistic map, ค่า r ตามแนวแกน x



รูปที่ 2-2 แสดง Bifurcation Diagram ของสมการ Logistic ที่ r ระหว่าง 0 ถึง 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป Bifurcation Diagram อธิบายได้ว่าเมื่อ

$r < 0$ ค่าสัญญาณที่เกิดจากสมการ Chaotic จะมีค่า $x=0$ เมื่อเวลาผ่านไปช่วงหนึ่ง

$r < 3$ ค่าสัญญาณที่เกิดจากสมการ Chaotic จะมีค่า x =ค่าคงที่

$r = 3.45$ ค่าสัญญาณที่เกิดจากสมการ Chaotic จะมีลักษณะเป็น 2 ค่า หรือสภาวะ Oscillate

$r < 3.54$ ค่าสัญญาณที่เกิดจากสมการ Chaotic จะมีลักษณะเป็น 4 ค่า หรือ Double Period

$r > 3.57$ สัญญาณจะลักษณะเป็น Chaotic

2.1.3 Hyper Chaos

คือสัญญาณ chaotic ที่มีความยุ่งเหยิง คาดเดาได้ยากกว่า สัญญาณ chaotic ธรรมดา ตัวอย่างสมการที่ใช้ในการสร้างสัญญาณนี้ได้แก่ สมการ Hyper Henon ผู้ค้นพบสมการ Hyper Henon คือนักวิจัยชื่อ G. Baier และ M. Klein ชาวเยอรมัน ซึ่งตีพิมพ์ในวารสาร PHYSICS LETTERS A Volume 151, Number 6,7 วันที่ 17 December 1990 มีสมการดังนี้

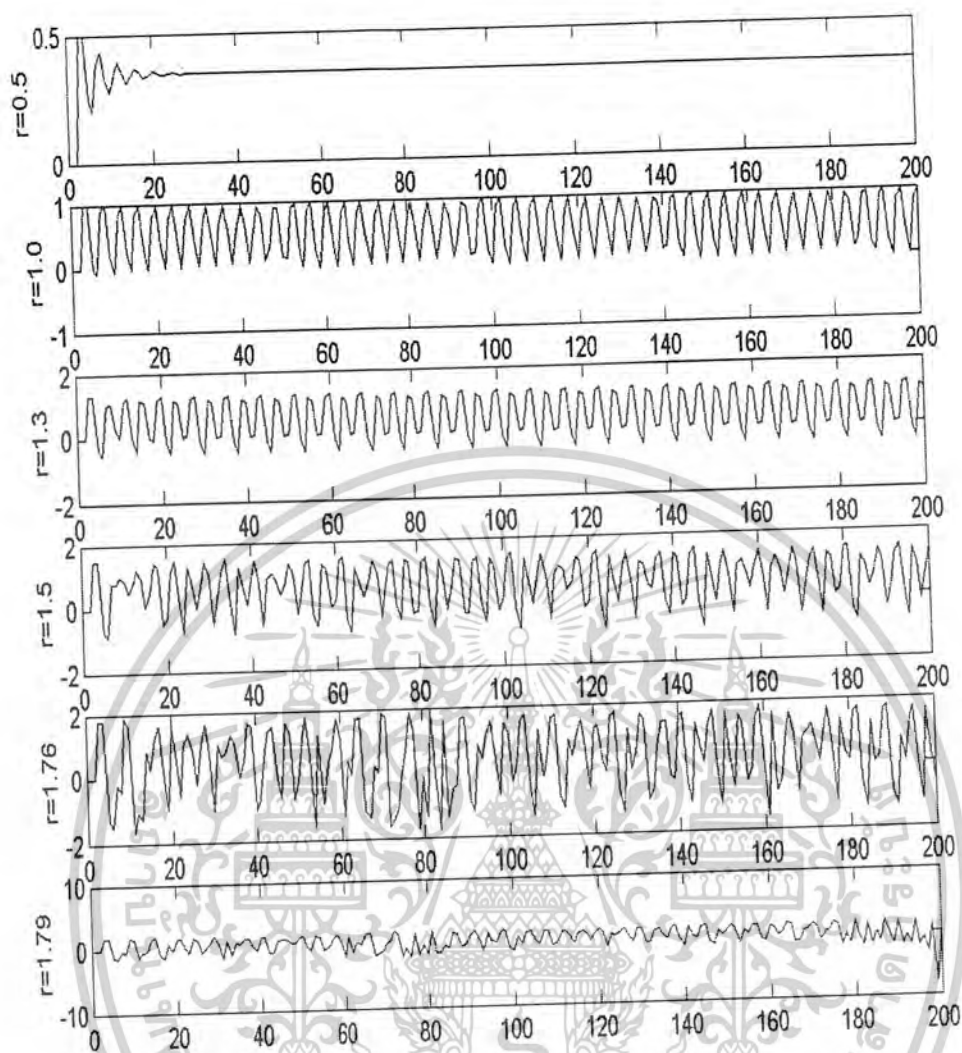
$$\left. \begin{aligned} x_1(k+1) &= r - x_2^2(k) - 0.1x_3(k) \\ x_2(k+1) &= x_1(k) \\ x_3(k+1) &= x_2(k) \end{aligned} \right\} \quad (2.2)$$

ค่าพารามิเตอร์สำหรับปรับให้เกิดสัญญาณ Chaotic คือค่า r

r คือค่าคงที่ เมื่อเปลี่ยนค่า r ไปเรื่อยๆ ผลจากกราฟ เมื่อค่า $r = 0.5$ กราฟที่ได้จะเป็นแบบ damped harmonic oscillator ตัวอย่าง เช่น การแกว่งของลูกตุ้ม ซึ่งลูกตุ้มก็จะแกว่งช้าลงๆ จนหยุด สำหรับ $r = 1.3$ สัญญาณจะเป็นสัญญาณ oscillator โดยไม่มีการ damping, oscillator นี้จะดำเนินต่อเนื่องไปไม่มีที่สิ้นสุด

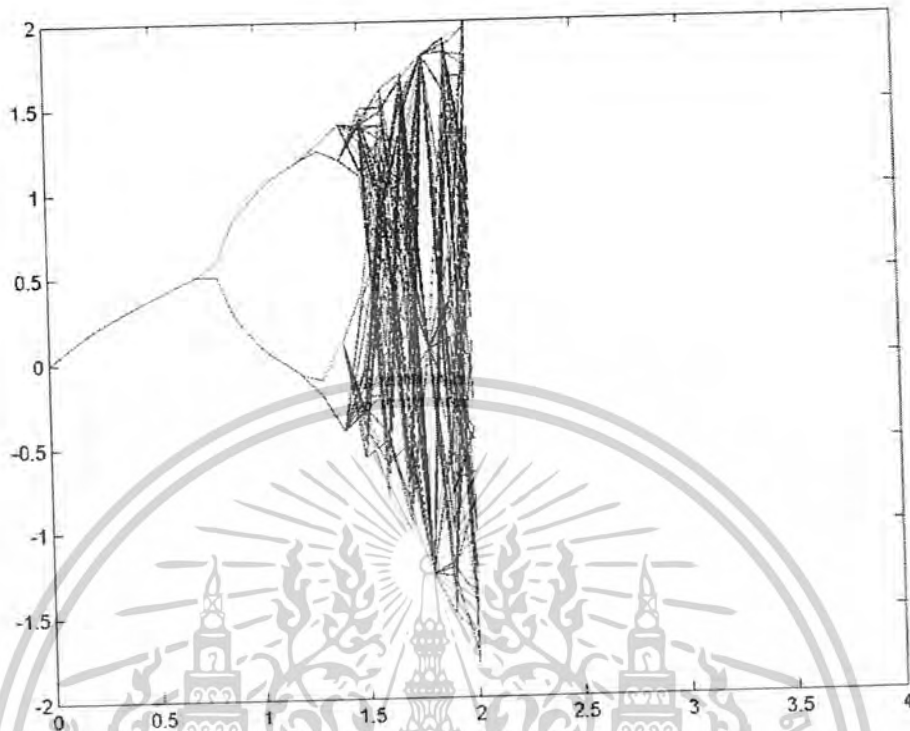
เมื่อค่า r มากกว่า 1.3 Logistic map จะเริ่มแสดงปรากฏการณ์ Chaotic ดูจากกราฟข้างล่างจะเห็นความแตกต่างของจุด peak และการแตกต่างกันเหลื่อมคมของกราฟ โดยสมการจะให้ความแตกต่างกันของรูปสำหรับค่า x_n ใด ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-3 แสดง การเปลี่ยนแปลง parameter โดยการเพิ่มค่า r จาก 0.5 เป็น 1.79

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-4 แสดง Bifurcation Diagram ของสมการ Hyper Henon ที่ r ระหว่าง 0 ถึง 4

จากรูป Bifurcation Diagram อธิบายได้ว่าเมื่อ
 $r < 0$ ค่าสัญญาณที่เกิดจากสมการ Chaotic จะมีค่า $x=0$ เมื่อเวลาผ่านไปช่วงหนึ่ง
 $r < 0.6$ ค่าสัญญาณที่เกิดจากสมการ Chaotic จะมีค่า x =ค่าคงที่
 $r = 0.6$ ค่าสัญญาณที่เกิดจากสมการ Chaotic จะมีลักษณะเป็น 2 ค่า หรือสถานะ Oscillate
 $r < 1.3$ ค่าสัญญาณที่เกิดจากสมการ Chaotic จะมีลักษณะเป็น 4 ค่า หรือ Double Period
 $r > 1.3$ สัญญาณจะลักษณะเป็น Chaotic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 JAVA

2.2.1 Introducing Java

ภาษา Java เริ่มเป็นที่สนใจ เมื่อบริษัท Sun Microsystems ประกาศให้เป็นภาษาสำหรับการสร้างโปรแกรมเพื่อใช้งานบนอินเทอร์เน็ต (Internet) โปรแกรมภาษา Java อาจจะถูกสร้างขึ้นบนเครื่องคอมพิวเตอร์เครื่องหนึ่ง แล้วนำไปทำงานบนเครื่องคอมพิวเตอร์ต่างระบบได้ โดยไม่ต้องคอมไพล์โปรแกรมใหม่ ความสามารถนี้จะเปลี่ยนโฉมหน้าและบทบาทของอินเทอร์เน็ตไปอย่างสิ้นเชิง เมื่อเรามีโปรแกรมที่สามารถทำงานบนเครื่องคอมพิวเตอร์ใดๆ ได้ อินเทอร์เน็ตจะกลายเป็นหนึ่งเดียว สิ่งที่จะส่งไปมาในอินเทอร์เน็ตไม่จำเป็นต้องแบ่งแยกระหว่างข้อมูลกับโปรแกรมอีกต่อไป นั่นคือ web pages ทั้งหลายจะไม่ใช่อเอกสารที่ถูกกระทำ (passive) แต่จะกลายเป็นเอกสารที่ทำงานได้ (active)

ผลกระทบเหล่านี้ทำให้ภาษา Java เป็นที่น่าสนใจอย่างยิ่ง ในขณะที่บริษัทผู้ผลิตซอฟต์แวร์ทั้งหลายได้ให้การต้อนรับภาษานี้ในทิศทางที่ดีเกินความคาดหมาย ประกอบด้วยภาษา Java มีประสิทธิภาพในการสร้างโปรแกรมอย่างยิ่ง จึงเป็นเครื่องมือชั้นดีสำหรับเทคโนโลยีการสร้างโปรแกรมด้วย components และ visual programming

ภาษา Java ถูกออกแบบให้มีลักษณะดังต่อไปนี้

1. เป็นภาษาที่ง่าย (simple) ในการเรียนรู้และใช้งาน ความหมายของคำว่า “ง่าย” อาจจะถูกตีความได้หลายอย่างในแง่ต่างๆ ดังนี้
 - ภาษา Java นำไวยากรณ์ภาษาส่วนใหญ่มาจากภาษา C และ C++ ผู้ที่คุ้นเคยกับภาษา C หรือ C++ อยู่แล้ว จะเข้าใจไวยากรณ์ภาษา Java ได้ง่าย หรือใช้เวลาศึกษาไม่นานนัก
 - ภาษา Java มีกลไกของภาษาจำนวนไม่มากและไม่ซับซ้อน โดยคัดกลไกของภาษา C และ C++ ส่วนที่ทำให้ยุ่งยากออกไป
 - ภาษา Java ประสบความสำเร็จอย่างมาก ในการใช้เทคนิคของภาษาเชิงวัตถุ ช่วยให้สร้างโปรแกรมที่ยุ่งยากเป็นไปได้ง่ายขึ้น
2. โปรแกรมที่สร้างขึ้นด้วยภาษา Java จะไม่มีความผิดพลาดจากข้อบกพร่องของภาษา นั่นคือ โปรแกรมจะต้องไม่ล้มเหลวลง ด้วยความผิดพลาดเพียงเล็กน้อยที่ไม่เกี่ยวกับตรรกะของโปรแกรม คุณสมบัติของภาษาอย่างนี้เรียกว่าความคงทน (robust) ภาษา Java ถูกออกแบบให้มีความคงทน ด้วยวิธีการดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ภาษา Java เน้นการใช้กลไก exception handling เพื่อให้โปรแกรมสามารถจัดการกับความผิดปกติบางอย่างที่เกิดขึ้นในขณะที่โปรแกรมทำงานให้โปรแกรมทำงานต่อไปได้โดยไม่ต้องหยุดลง

- ภาษา Java ตัดกลไกบางอย่างในภาษา C และ C++ ที่อาจจะทำให้เกิดความผิดพลาด หากใช้อย่างไม่ระวัง

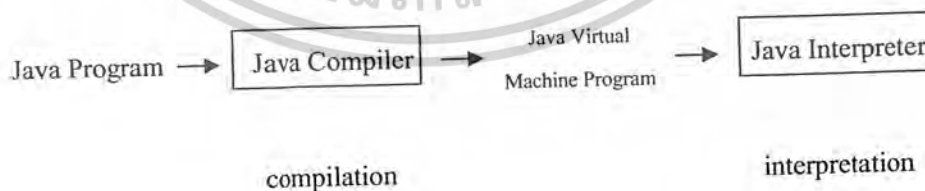
- ภาษา Java ไม่มีกลไกสำหรับคืน (de-allocation) หน่วยความจำที่ขอมมาในขณะที่โปรแกรมทำงาน (dynamic memory allocation) อย่างที่มีในภาษา C และ C++ คือ free() และ delete() ภาษา Java อาศัย automatic garbage collector ทำหน้าที่เก็บหน่วยความจำที่ไม่สามารถอ้างถึงได้แล้วกลับไปใช้งานใหม่

- ภาษา Java เป็นภาษาประเภท strongly typed หมายถึง ภาษาที่เน้นความถูกต้องของชนิดข้อมูล (type) ที่ใช้ใน โปรแกรม คอมไพเลอร์ของภาษาประเภทนี้จะทำการตรวจสอบว่าโปรแกรมการจัดการกับชนิดข้อมูลของตัวแปรถูกต้องหรือไม่

3. โปรแกรมภาษา Java มักถูกส่งผ่านระบบเครือข่ายไปทำงานบนเครื่องคอมพิวเตอร์ของผู้อื่น ดังนั้น ภาษา Java ต้องมีหลักประกันให้แก่ผู้รับ โปรแกรมนั้นไปทำงานว่าจะไม่ก่อให้เกิดอันตรายต่อเครื่องหรือระบบของเขา ภาษา Java จึงมีข้อกำหนดหลายอย่าง เพื่อให้โปรแกรมไม่สามารถทำอันตราย หรือสิ่งที่ไม่สมควรทำ ต่อระบบที่รับ โปรแกรมนั้นไปทำงาน คุณสมบัติของภาษาอย่างนี้เรียกว่า ปลอดภัย (secure)

2.2.2 Java Virtual Machine

ภาษา Java นำความคิดการสร้างเครื่องจักรสมมติมาใช้ เพื่อให้โปรแกรมไม่ขึ้นกับระบบ โดยมีคอมไพเลอร์ทำการแปลภาษาให้เป็นโปรแกรมของ Java Virtual Machine (JVM) แล้วนำโปรแกรมนั้นมาทำงานด้วยเครื่องจักรสมมติที่จำลองขึ้น โดย Java interpreter สรุปได้ดังรูปนี้



ในวิธีนี้โปรแกรมภาษา Java จะถูกคอมไพล์โดย Java compiler ได้เป็นโปรแกรมของ JVM แล้วสามารถนำไปทำงานบนเครื่องใดๆที่มี Java interpreter ได้จึงมีคุณสมบัติไม่ขึ้นกับระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 Java Developer Kit (JDK)

ในการสร้างโปรแกรมภาษา Java อย่างน้อยเราจะต้องมี Java compiler และ Java interpreter ปัจจุบันมีหลายบริษัทผลิตชุดโปรแกรมสำหรับนักพัฒนาโปรแกรม Java ออกมา เช่น

- Java Developer Kit (JDK) ของบริษัท JavaSoft
- Visual Café ของบริษัท Symantec
- J Builder ของบริษัท Borland
- JDeveloper ของบริษัท Oracle
- Visual Age for Java ของบริษัท IBM
- Visual J++ ของบริษัท Microsoft

2.2.4 JDK Installation

เมื่อเราติดตั้ง JDK จะเห็นว่ามีโปรแกรมและไฟล์ถูกติดตั้งลงไปจำนวนมาก ดังนั้นเพื่อไม่ให้การแนะนำยาวเกินไป จึงขอกล่าวเฉพาะโปรแกรมที่จำเป็นในการสร้างโปรแกรมภาษา Java ขึ้นพื้นฐานเท่านั้น หากดูในไดเรกทอรี C:\jdk\bin จะพบโปรแกรมที่เราต้องใช้งานบ่อยๆดังนี้

Javac.exe คือ Java compiler

Java.exe คือ Java interpreter

Appletviewer.exe คือ โปรแกรมสำหรับดู applet

โปรแกรมติดตั้ง JDK ไม่ได้เปลี่ยนแปลง autoexec.bat ดังนั้นในตอนนี้อย่างน้อย Java compiler (javac.exe) และ Java interpreter (java.exe) จะถูกเรียกใช้ได้จากไดเรกทอรี c:\jdk\bin เท่านั้น ปกติเราจะไม่ทำงานในไดเรกทอรีที่เก็บคอมไพเลอร์ จึงควรสร้างและทำงานในไดเรกทอรีที่เก็บเฉพาะโปรแกรมที่กำลังถูกพัฒนา ดังนั้นเพื่อให้สามารถเรียกใช้ javac.exe และ java.exe ได้จากไดเรกทอรีอื่น จึงควรตั้งค่า directory path ไปที่ c:\jdk\bin โดยเพิ่มประโยคต่อไปนี้ลงในไฟล์ autoexec.bat

```
SET PATH=%PATH%;C:\JDK\BIN;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระหว่างที่คอมไพล์โปรแกรมหรือขณะที่โปรแกรมทำงาน หากโปรแกรมมีการอ้างถึงคลาสๆหนึ่ง Java compiler และ Java interpreter จะเริ่มค้นหาคลสนั้นจากไฟล์

C:\JDK\LIB\CLASSES.ZIP

และในไคเรททอรีปัจจุบัน โดย default แสดงว่าหากเราใช้เฉพาะคลาสที่อยู่ในสองที่นี้ก็ไม่ต้องการกำหนดค่า CLASSPATH ใดเพิ่ม แต่หากเรามีคลาสที่ต้องเรียกใช้เก็บไว้ในที่ สมมติเป็น MyClasses ก็ต้องกำหนดค่าให้แก่ตัวแปร CLASSPATH โดยเพิ่มประโยชน์ด้านล่างนี้ไปในไฟล์ autoexec.bat

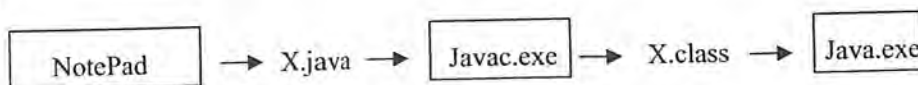
SET CLASSPATH=.;C:\JDK\LIB\CLASSES.ZIP;C:\MyClasses

เพื่อบอกให้ค้นหาคลสเริ่มต้นจาก ไคเรททอรีปัจจุบัน (คือ ".") หากไม่พบให้ไปหาต่อในไฟล์ classes.zip และถ้ายังไม่เจออีกให้ไปหาต่อในไคเรททอรี C:\MyClasses

2.2.5 Creating Java Application

Java application คือ โปรแกรมที่สร้างขึ้นด้วยภาษา Java เพื่อถูกทำงานโดย Java interpreter ต้องเป็นโปรแกรมที่สมบูรณ์ มี main() เป็นจุดเริ่มต้น และสามารถควบคุมการดำเนินไปของตัวเองได้ ทำงานภายใต้ Java interpreter โดยไม่ต้องมีคปรแกรมอื่นช่วย บางคนจึงเรียกว่า stand alone program

ขั้นตอนการสร้างและการทำงาน Java Application มีดังนี้ เริ่มจากใช้ editor (อย่างเช่น notepad) เขียนโปรแกรมสำหรับ Java application นั้นเก็บไว้ในไฟล์ที่มี extension เป็น java เช่น X.java จากนั้นใช้ javac.exe ทำการคอมไพล์ X.java จะได้ผลลัพธ์เป็นไฟล์ X.class ซึ่งเป็นโปรแกรม JVM ของ class X เมื่อต้องการให้ X.class นั้นทำงานก็ส่งให้แก่ java.exe ดังแสดงในรูป



Compilation

Interpretation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

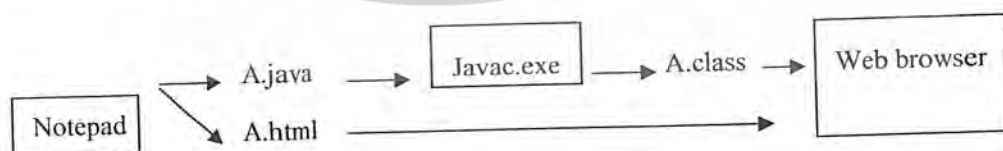
2.2.6 Creating Java Applets

Applets คือโปรแกรมภาษา Java ที่สร้างขึ้นเพื่อส่งไปกับ HTML page ให้ไปทำงานภายใต้ web browser ที่มี java interpreter โดยสามารถติดต่อกับผู้ใช้และแสดงผลผ่านทาง web page นั้น หรือกล่าวได้ว่าเป็นโปรแกรมที่ฝากมากับ HTML page นั่นคือ จากใน HTML page จะสามารถเรียก applet มาทำงานและแสดงผลในพื้นที่ของ page นั้น โดยที่ web browser จะเป็นผู้โหลด applet ที่ถูกเรียกมาจากเซิร์ฟเวอร์ที่ ให้ HTML page นั้นมาและจัดการทำงานโดยอัตโนมัติ วิธีทำให้เราสามารถส่งโปรแกรมไปทำงานบนเครื่องใดๆ ในระบบอินเทอร์เน็ต โดยใช้ HTML page เป็นตัวกลาง

ขั้นตอนการสร้างทำงาน applet มีดังนี้ ใช้ editor เขียนโปรแกรมสำหรับ applet นั้นเก็บไว้ในไฟล์ที่มี extension เป็น .java เช่น A.java แล้วใช้คอมไพเลอร์ javac.exe คอมไพล์ A.java จะได้ผลลัพธ์เป็นไฟล์ A.class ซึ่งเป็นโปรแกรม JVM ของ applet นั้น สังเกตว่า java.exe จะไม่สามารถทำงาน A.class เพราะไม่ใช่ java application ต่อจากนั้นให้ใช้ editor เขียน HTML page สมมติว่า เป็น a.html ที่มีประโยค

```
<applet code="A.class" width=100 height=100></applet>
```

ซึ่งจะเรียก A.class มาทำงานและแสดงผลในพื้นที่ขนาด 100x100 โดย default ไฟล์ของ applet นั้นจะถูกมองหาในไดเรกทอรีและเครื่องเดียวกันกับ HTML page นั้น หากกรณีที่ applet อยู่ที่เครื่องอื่นก็ต้องอ้าง URL ให้ถึงไฟล์ของ applet นั้น สุดท้ายเมื่อต้องการดูผลของ applet นี้ก็ใช้ web browser ที่มี Java interpreter อย่างเช่น Netscape หรือ Microsoft IE เรียกดูไฟล์ a.html เมื่อไฟล์นี้ถูกจัดการถึงประโยคที่เรียก applet นั้น A.class ก็จะถูกโหลดมาทำงาน ดังแสดงในรูปนี้



ปัจจุบัน web browsers ส่วนใหญ่สามารถทำงาน applet ได้ แต่มีของบางบริษัทที่ทำงานได้ไม่ตรงมาตรฐาน Java หรือไม่สนับสนุนกลไกใหม่ๆ บางอย่าง ดังนั้นเพื่อให้ผู้ใช้สามารถพัฒนาและทำการตรวจสอบ applet ได้จึงให้โปรแกรม appletviewer.exe มากับ JDK สำหรับดูผลของ applet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Appletviewer.exe จะไม่สามารถทำงานได้เหมือนกับ Web browser ทั่วไป คือจะแสดงเฉพาะ applet ที่ระบุระหว่าง <applet></applet> โคนไม่สนใจข้อความอื่นๆ ในไฟล์ และไม่สามารถติดต่อกับระบบเครือข่ายได้เหมือนกับ web browser แต่อย่างไรก็ตาม appletviewer.exe มีขนาดเล็กและทำงานเร็วกว่า web browser ทั่วไปมาก

การใช้ web browser ดู applet จะมีปัญหาอย่างหนึ่งคือ web browser จะจำชื่อคลาสที่โหลดมาแล้วไว้ และจะไม่โหลดใหม่เมื่อมีการอ้างถึงคลาสนั้นอีก เพื่อความเร็ว เพราะการโหลดคลาสอาจเสียเวลามาก แต่หากเราอยู่ในระหว่างการพัฒนา applet เมื่อใช้ web browser ดู applet หนึ่งแล้ว ถ้าทำการเปลี่ยนแปลงโปรแกรมของ applet และคอมไพล์ใหม่

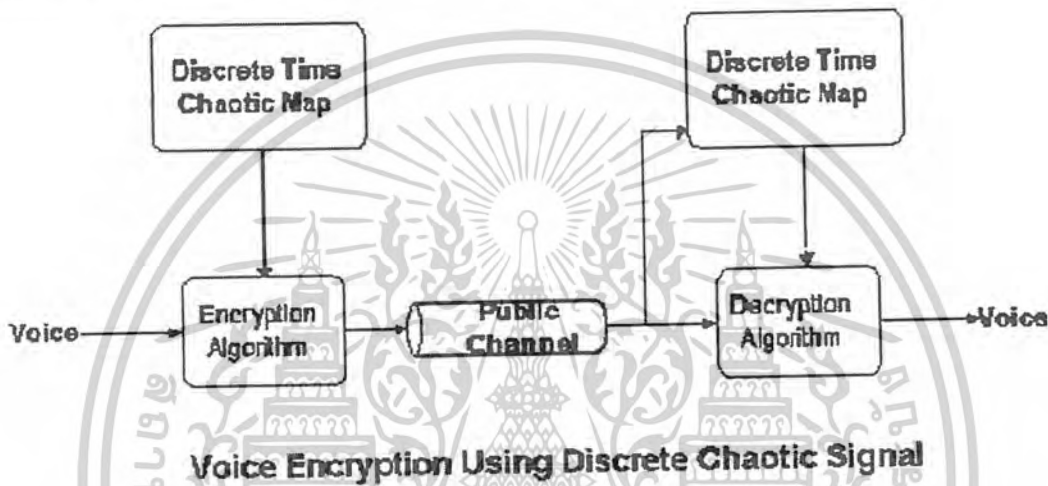
แม้ว่าเรา refresh ให้โหลด HTML page นั้นใหม่ แต่ applet ใหม่นั้นจะไม่ถูกโหลด เราจะต้องออกจาก web browser นั้น แล้วเรียกกลับเข้ามาใหม่ ซึ่งเสียเวลามาก เพราะการเริ่มต้น web browser แต่ครั้งใช้เวลาานานมาก เราจึงควรใช้ apple viewer ในการพัฒนา applets จะสะดวกกว่า



บทที่ 3

การออกแบบและการทดลอง

ในการเข้ารหัสเราจะทำตามขั้นตอนของ Block Diagram ดังรูป



รูปที่ 3-1 Block Diagram การเข้ารหัสลับ

3.1 Software

ในด้าน Software เราจะใช้โปรแกรม MatLab ในการทดลองเข้ารหัสเสียงตาม Block Diagram ข้างต้น ซึ่งการทดลองจะใช้สมการ Logistic ในการสร้างสัญญาณ chaotic และ Hyper Henon ในการสร้างสัญญาณ hyper chaotic

3.1.1 การทดลองการเข้ารหัสโดยใช้สัญญาณ chaotic จากสมการ logistic โดยโปรแกรม

Matlab

เราจะเขียน code ได้ดังนี้ โดยในที่นี้เราจะกำหนดค่า $r=3.7$

```
[x,fs]=wavread('h:\wav\all_done.wav');
sound(x,20911,8);
wavwrite(x,20911,8,'h:\wav\copyall_done.wav');

z=(1:20911);for i=1:20911,z(i)=0;end
r=3.7;
d=0.5*ones(1,size(r,2));
for i=1:20911,d=r.*d.*(1-d);end
y=[];
for i=1:20911,d=r.*d.*(1-d),y=[y,d'];end
for i=1:20911,z(i)=x(i)+y(i);end
wavwrite(z,20911,8,'h:\wav\Encryptall_done.wav');

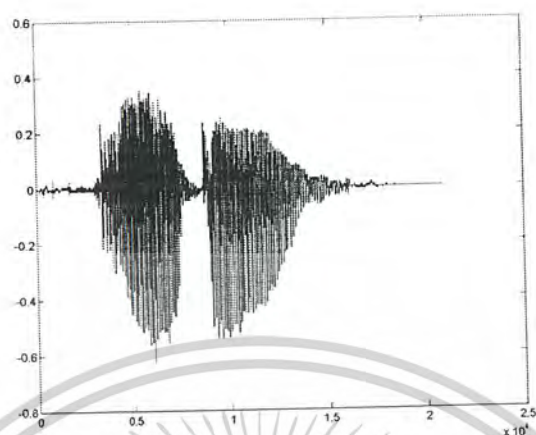
a=(1:20911);for i=1:20911,a(i)=0;end
for i=1:20911,a(i)=z(i)-y(i);end
wavwrite(a,20911,8,'h:\wav\Oriall_done.wav');
```

โดยจากการทดลองเราจะได้ไฟล์เสียงขึ้นมาใหม่ 3 ไฟล์คือ

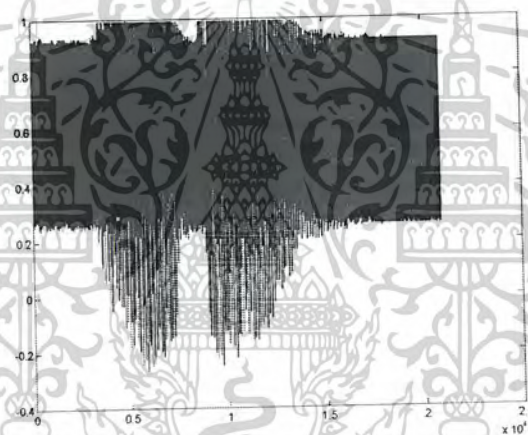
Copyall_done.wav	เป็นไฟล์เสียงต้นแบบที่ผ่านการ Sampling
Encrypall_done.wav	เป็นไฟล์เสียงที่ได้รับการเข้ารหัสแล้ว
Oriall_done.wav	เป็นไฟล์เสียงที่ได้รับการถอดรหัสแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

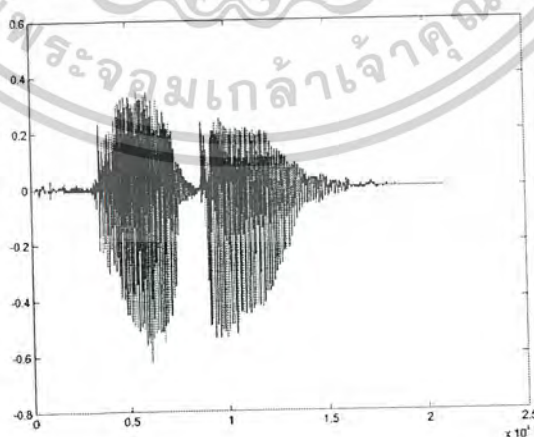
ซึ่งจะได้สัญญาณดังนี้



รูปที่ 3-2 แสดง Original Signal



รูปที่ 3-3 แสดง Coded Signal [Logistic]



รูปที่ 3-4 แสดง Recovered Signal

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 การทดลองการเข้ารหัสโดยใช้สัญญาณ chaotic จากสมการ Hyper Henon โดย

โปรแกรม Matlab

เราจะเขียน code ได้ดังนี้ โดยในที่นี้เราจะกำหนดค่า $r=1.7$

```
[x,fs]=wavread('h:\wav\hyper\all_done.wav');
sound(x,20911,8);
wavwrite(x,20911,8,'h:\wav\hyper\copyall_done.wav');

z=(1:20911);for i=1:20911,z(i)=0;end
r=1.7;
d=0.5*ones(1,size(r,2));
for i=1:20911,d=r-d.*(d-0.1);end
y=[];
for i=1:20911,d=r-d.*(d-0.1);y=[y,d'];end
for i=1:20911,z(i)=x(i)+y(i);end
wavwrite(z,20911,8,'h:\wav\hyper\Encryptall_done.wav');

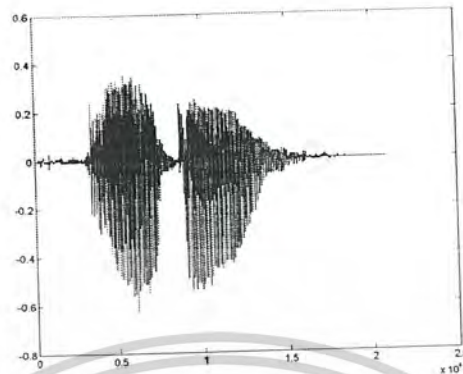
a=(1:20911);for i=1:20911,a(i)=0;end
for i=1:20911,a(i)=z(i)-y(i);end
wavwrite(a,20911,8,'h:\wav\hyper\Oriall_done.wav');
```

โดยจากการทดลองเราจะได้ไฟล์เสียงขึ้นมาใหม่ 3 ไฟล์คือ

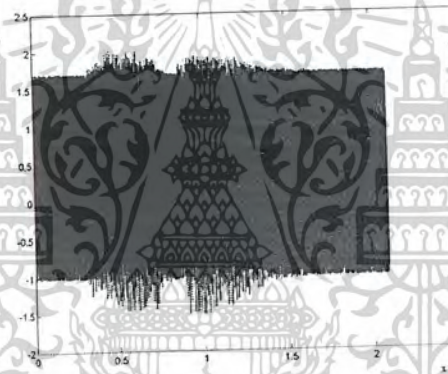
Copyall_done.wav	เป็นไฟล์เสียงต้นแบบที่ผ่านการ Sampling
Encrypall_done.wav	เป็นไฟล์เสียงที่ได้รับการเข้ารหัสแล้ว
Oriall_done.wav	เป็นไฟล์เสียงที่ได้รับการถอดรหัสแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

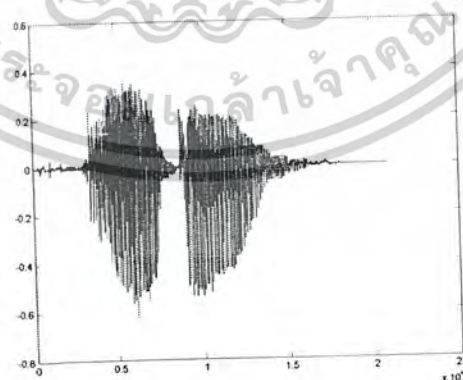
ซึ่งจะได้สัญญาณดังนี้



รูปที่ 3-5 แสดง Original Signal



รูปที่ 3-6 แสดง Coded Signal [Hyper Henon]



รูปที่ 3-7 แสดง Recovered Signal

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 การสร้างโปรแกรมที่ใช้แสดงสัญญาณของสมการ Logistic ในรูป Applet

Source Code ที่ใช้ในการสร้างประกอบไปด้วย Class ไฟล์ต่างๆดังนี้

- DifEqnAp - class ที่ใช้แสดง applet หลัก
- TimeGr - class ที่ใช้สร้างและพลอตกราฟเวลา
- BouceGr - class ที่ใช้สร้างและพลอตกราฟ
- CtrlPanl - class ที่ใช้แสดงส่วนควบคุมต่างๆ
- SelPer - class เกี่ยวกับปุ่มเกี่ยวกับเวลา the iteration period selector radio buttons
- SelPlot - class ที่เกี่ยวกับการเลือกโหมดการพลอต
- SelEqn - class ที่ใช้ในการเลือก สมการ
- SelScan - class ในการเลือก scan mode
- SetCval - class ที่ใช้แสดงส่วนค่าคงที่ c
- IncCval - class เกี่ยวกับปุ่ม + และ - buttons สำหรับการเปลี่ยนค่า c
- SetButt - class เกี่ยวกับปุ่ม Start, Stop และ Clear buttons

ซึ่ง Source Class ต่างๆที่ใช้จะแสดงในภาคผนวก

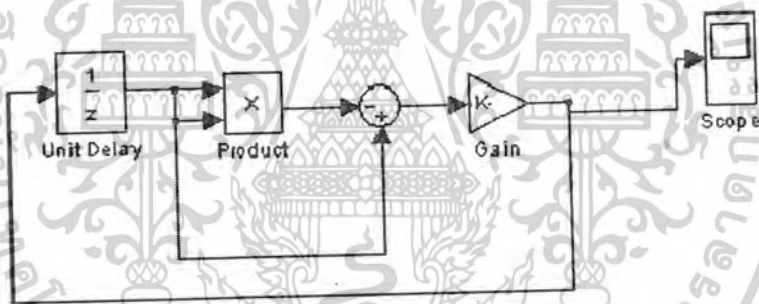
3.1.4 การสร้างโปรแกรมที่ใช้แสดงสัญญาณของสมการ Hyper Henon ในรูป Applet

Source Code ที่ใช้ในการสร้างประกอบไปด้วย Class ไฟล์ที่คล้ายกับ Logistic จะต่างกันที่ไฟล์ BouceGr.class ที่ใช้ในการสร้างสมการ ซึ่งจะแสดงในภาคผนวก

3.2 Hardware

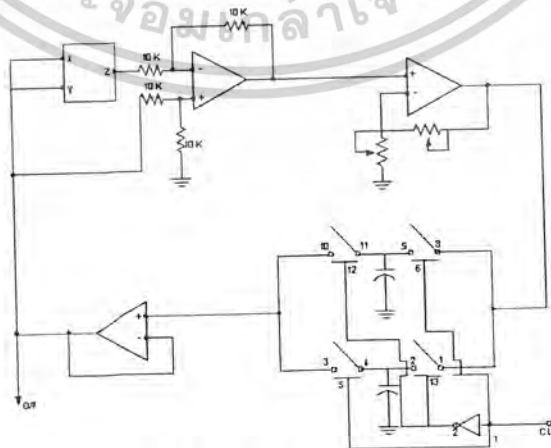
ในการออกแบบวงจรเพื่อสร้างภาคส่งตามรูป 3.1 จากสมการ Logistic สามารถที่จะนำมาเขียนเป็นบล็อกไดอะแกรมได้ดังนี้

3.2.1 Logistic Block Diagram



รูปที่ 3-8 แสดง logistic block diagram

การปรับ gain คือการปรับค่า r ซึ่งเป็นค่า parameter key ซึ่งทำเป็นวงจรได้ดังนี้

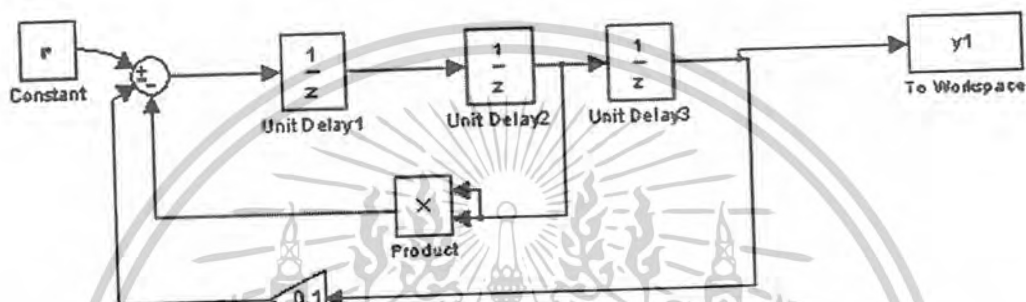


รูปที่ 3-9 วงจร logistic map

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการออกแบบวงจรเพื่อสร้างภาคส่งตามรูป 3.1 จากสมการ Hyper Henon สามารถที่จะนำมาเขียนเป็นบล็อกโคอะแกรมได้ดังนี้

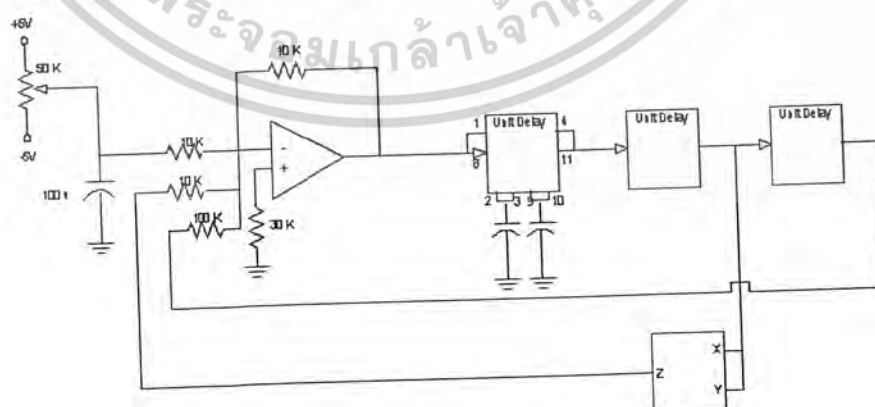
3.2.2 Hyper Henon Block Diagram



รูปที่ 3-11 แสดง hyper henon block diagram

Unit Delay 1 กับ Unit Delay 2 จะหน่วงเวลาของสัญญาณช่วงเวลาหนึ่งแล้วส่งต่อไปที่ วงจรคูณและ Unit Delay 3 สัญญาณจาก Unit Delay 3 จะนำไปขยาย 0.1 เท้าลกับสัญญาณจาก วงจรคูณและบวกกับค่าแรงดันคงที่ค่าหนึ่งค่าใดแล้วส่งสัญญาณย้อนกลับไปที่ Unit Delay 1 โดย เราจะจับสัญญาณจากสัญญาณที่ออกจาก Unit Delay 3

การปรับ gain คือการปรับค่า r ซึ่งเป็นค่า parameter key ซึ่งทำเป็นวงจรได้ดังนี้



รูปที่ 3-12 แสดง hyper henon map

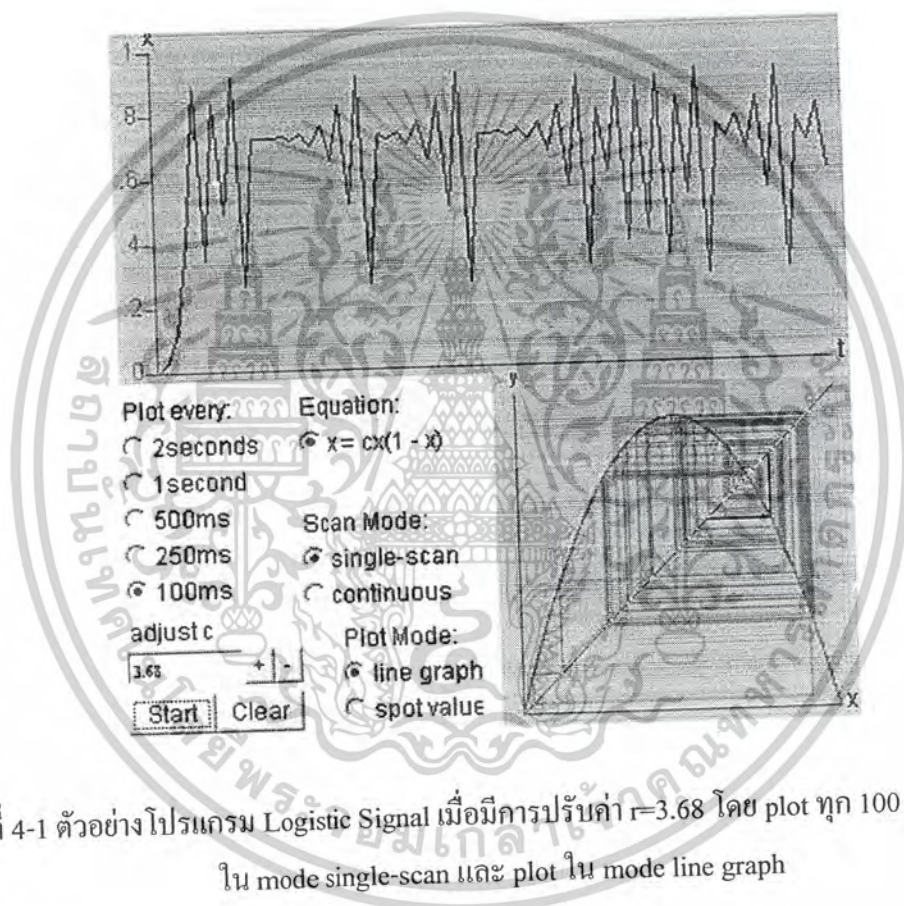
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

4.1 Software

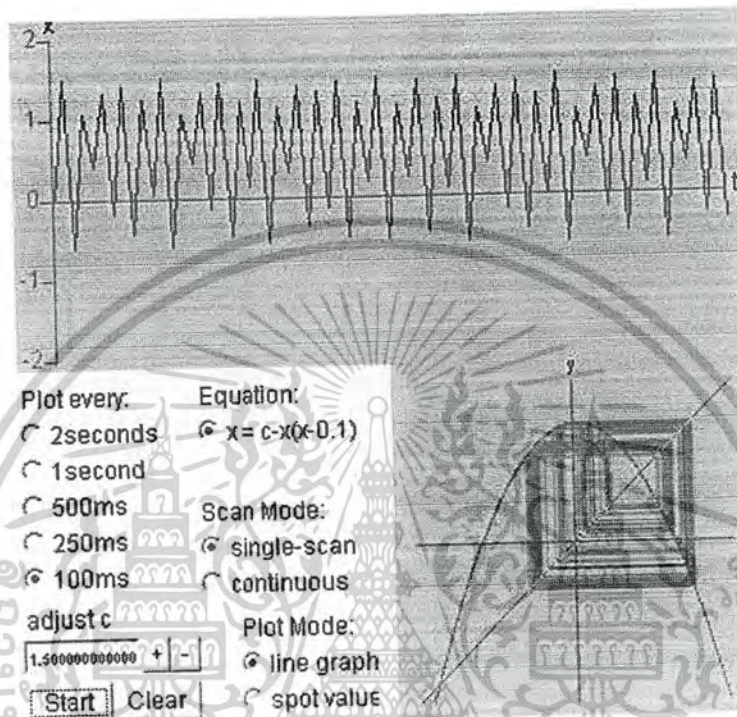
4.1.1 ผลการทดลองจากโปรแกรมสร้างสัญญาณจากสมการ Logistic โดยใช้ Java



รูปที่ 4-1 ตัวอย่าง โปรแกรม Logistic Signal เมื่อมีการปรับค่า $r=3.68$ โดย plot ทุก 100 ms โดย scan ใน mode single-scan และ plot ใน mode line graph

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 ผลการทดลองจากโปรแกรมสร้างสัญญาณจากสมการ Hyper Henon โดยใช้ Java

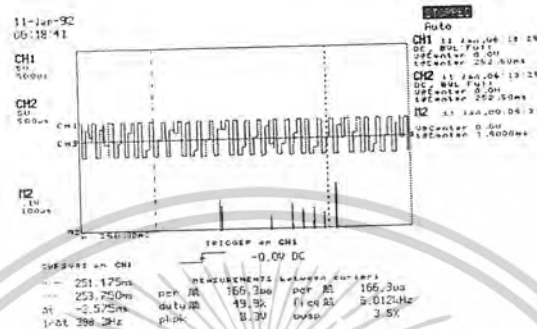


รูปที่ 4-2 ตัวอย่าง โปรแกรม Hyper Henon Signal เมื่อมีการปรับค่า $c=1.5$ โดย plot ทุก 100 ms โดย scan ใน mode single-scan และ plot ใน mode line graph

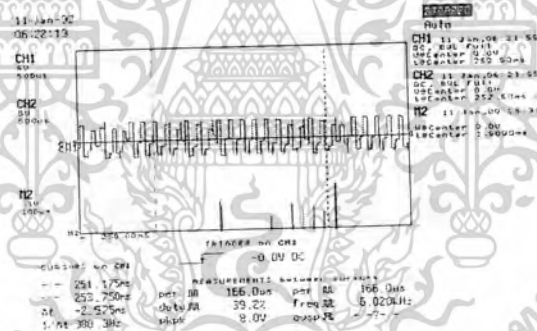
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 Hardware

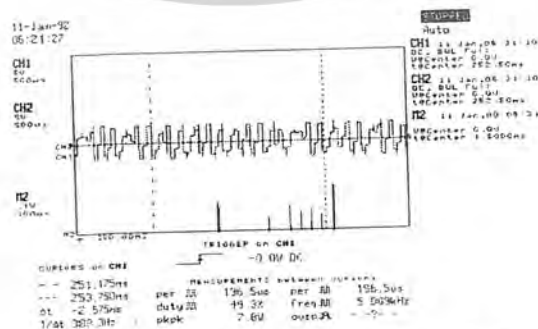
4.2.1 ผลการทดลองวงจรสร้างสัญญาณ Chaos แบบ logistic ของวงจรตามรูป 3-9



รูปที่ 4-3 รูปสัญญาณจากวงจร logistic เมื่อกำหนด $R_F = 2.5\text{ k}\Omega$ โดยค่า $R_i = 975\Omega$, $r = 3.57$

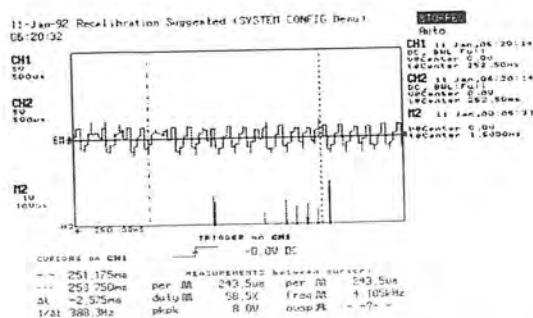


รูปที่ 4-4 รูปสัญญาณจากวงจร logistic เมื่อกำหนด $R_F = 2.5\text{ k}\Omega$ โดยค่า $R_i = 960\Omega$, $r = 3.6$



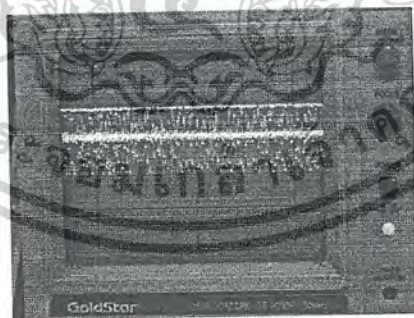
รูปที่ 4-5 รูปสัญญาณจากวงจร logistic เมื่อกำหนด $R_F = 2.5\text{ k}\Omega$ โดยค่า $R_i = 925\Omega$, $r = 3.7$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



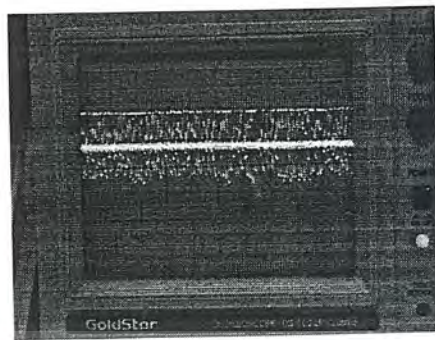
รูปที่ 4-6 รูปสัญญาณจากวงจร logistic เมื่อกำหนด $RF = 2.5 \text{ k}\Omega$ โดยค่า $R_i = 890 \Omega$, $r = 3.8$

4.2.2 ผลการทดลองวงจรสร้างสัญญาณ Chaos แบบ Hyper Henon ของวงจรตามรูป 3-12



รูปที่ 4-7 รูปสัญญาณจากวงจร Hyper Henon เมื่อกำหนด $V = -1.8$

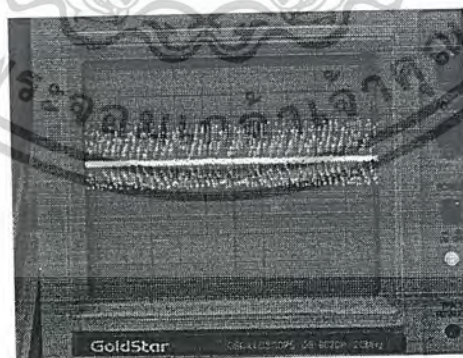
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-8 รูปสัญญาณจากวงจร Hyper Henon เมื่อกำหนด $V=-1.5$

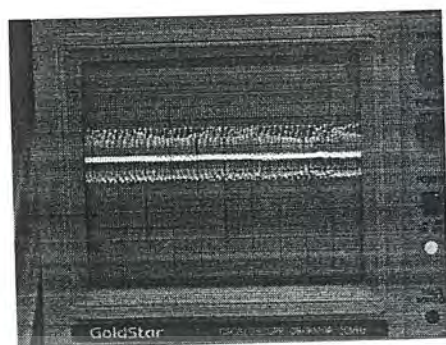


รูปที่ 4-9 รูปสัญญาณจากวงจร Hyper Henon เมื่อกำหนด $V=-1.2$



รูปที่ 4-10 รูปสัญญาณจากวงจร Hyper Henon เมื่อกำหนด $V=-1.05$

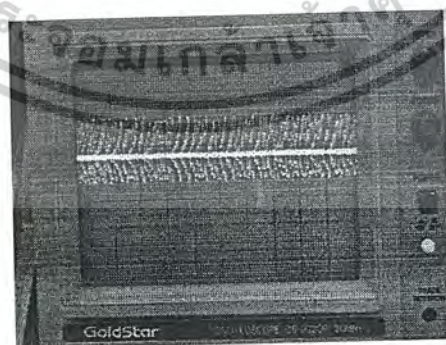
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-11 รูปสัญญาณจากวงจร Hyper Henon เมื่อกำหนด $V=-0.94$



รูปที่ 4-12 รูปสัญญาณจากวงจร Hyper Henon เมื่อกำหนด $V=-0.67$



รูปที่ 4-13 รูปสัญญาณจากวงจร Hyper Henon เมื่อกำหนด $V=-1.09$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลอง

5.1 สรุปผลการทดลอง

โครงการนี้เป็น การนำสัญญาณเคออสติกจากสมการ Logistic และ Hyper Henon มาใช้ในการประยุกต์ใช้งานในด้านการรักษาความปลอดภัยของข้อมูล ซึ่งสัญญาณเคออสติกที่ได้จากสมการ Hyper Henon มีความยุ่งเหยิงกว่าสัญญาณเคออสติกที่ได้จากสมการ Logistic ทั้งนี้ถ้าอยากให้เกิดสัญญาณเคออสติกที่มีความยุ่งเหยิงมากขึ้นเพียงใดก็สามารถที่จะทำการปรับค่าพารามิเตอร์ r ได้

5.2 ปัญหาที่เกิดขึ้นในการทดลอง

ในการทดลองการเข้ารหัสลับสัญญาณเสียงด้าน Software นั้น ปัญหาที่พบได้ทั่วไปคือ ภาษาที่ใช้ควรจะ เป็นแบบไหนจึงเหมาะสม และออกแบบอย่างไรจึงได้เสียงที่ถูกต้อง ซึ่งในการแก้ปัญหานั้นได้เลือกใช้โปรแกรมเมทแลป (MATLAB) เข้ามาช่วยในการสร้างส่วนติดต่อกับผู้ใช้ และเนื่องจากในตัวของเมทแลปเองนั้น มีฟังก์ชันการทำงานในเชิงคณิตศาสตร์มาก จึงสะดวกต่อการพัฒนา โดยไม่จำเป็นต้องเขียนฟังก์ชันขึ้นมาใช้เองมากนัก ส่วนในการออกแบบนั้นได้นำหลักการทางคณิตศาสตร์มาใช้ โดยการ plot กราฟสัญญาณก่อนเข้ารหัสลับเทียบกับกราฟสัญญาณ หลังจากการถอดรหัส

ส่วนในการสร้างวงจรทางอิเล็กทรอนิกส์นั้น ปัญหาที่พบคืออุปกรณ์บางประเภทไม่ได้คุณภาพจึงทำให้ผลที่ได้มีความคลาดเคลื่อนเล็กน้อย อีกทั้งเครื่องมือที่ใช้ในการวัดผลการทดลอง ไม่มีความเที่ยงตรงจึงทำให้เป็นปัญหาในการเก็บผลการทดลอง

5.3 แนวทางในการพัฒนา

สัญญาณเคออสติกเป็นสัญญาณที่มีลักษณะคล้ายสัญญาณรบกวน แต่เราสามารถควบคุมการเกิดของสัญญาณได้โดยกำหนดค่าตัวแปรต่าง ๆ และค่าเริ่มต้นให้กับตัวกำเนิดสัญญาณ

การใช้งานสัญญาณเคออสติกนั้น ในที่นี้ขอยกตัวอย่าง การใช้งานสัญญาณเคออสติกในระบบสื่อสาร ซึ่งสัญญาณเคออสติกนั้นมีลักษณะคล้ายสัญญาณรบกวน หากนำไปทำการเข้ารหัสลับกับข้อมูลสื่อสารที่ต้องการนั้น ทำให้ได้ชุดข้อมูลซับซ้อนยากต่อการถอดรหัส ซึ่งผู้ที่จะถอดรหัสได้นั้น จะต้องทราบค่าตัวแปรและค่าเริ่มต้นที่ตรงกันทั้งสองฝ่าย คือ ทั้งด้านรับและส่ง ในที่นี้จึงถือได้ว่าตัวแปรและค่าเริ่มต้นเป็นกุญแจที่ไว้สำหรับถอดรหัสลับ ซึ่งขึ้นกับผู้ใช้งานจะเป็นกุญแจสาธารณะ หรือกุญแจส่วนตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] International Journal Of Bifurcation And Chaos , Vol. 12 , No. 1 (2000)
- [2] ดร. วีระศักดิ์ ชิงฉาวร , Java Programming Vol. I , ซีเอ็ดยูเคชั่น , 2545.
- [3] ดร. วีระศักดิ์ ชิงฉาวร , Java Programming Vol. II , ซีเอ็ดยูเคชั่น , 2545.

<http://home.clara.net/robmorton/projects/index.htm#java>

http://www.wikipedia.org/wiki/Bifurcation_diagram

<http://mathworld.wolfram.com/LogisticEquation.html>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

Source Code ที่ใช้ในการสร้างโปรแกรม Logistic Signal ประกอบไปด้วย Class ไฟล์ต่างๆดังนี้

- DifEqnAp - class ที่ใช้แสดง applet หลัก
- TimeGr - class ที่ใช้สร้างและพลอตกราฟเวลา
- BouceGr - class ที่ใช้สร้างและพลอตกราฟ
- CtrlPanl - class ที่ใช้แสดงส่วนควบคุมต่างๆ
- SelPer - class เกี่ยวกับปุ่มเกี่ยวกับเวลา the iteration period selector radio buttons
- SelPlot - class ที่เกี่ยวกับการเลือกโหมดการพลอต
- SelEqn - class ที่ใช้ในการเลือก สมการ
- SelScan - class ในการเลือก scan mode
- SetCval - class ที่ใช้แสดงส่วนค่าคงที่ c
- IncCval - class เกี่ยวกับปุ่ม + และ - buttons สำหรับการเปลี่ยนค่า c
- SetButt - class เกี่ยวกับปุ่ม Start, Stop และ Clear buttons

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

difeqnap.java

```
import java.applet.Applet;
import java.awt.*;

public class difeqnap extends Applet
    implements Runnable
{

    public void init()
    {
        setLayout(null);
        ctrlPanl = new ctrlpanl();
        add(ctrlPanl);
        ctrlPanl.reshape(0, 180, 180, 170);
        font = new Font("Dialog", 0, 12);
        fm = getFontMetrics(font);
        d = size();
        TG = createImage(d.width, 175);
        gt = TG.getGraphics();
        tg = new timmgr(gt, fm, bgClr);
        BG = createImage(180, 180);
        gb = BG.getGraphics();
        bg = new bouncegr(gb, bgClr);
        getCtrlPanlState();
        ctrlPanl.start();
        RunFlag = true;
        lter = new Thread(this);
        lter.start();
        lter.suspend();
        T = System.currentTimeMillis() + Period;
    }

    void getCtrlPanlState()
    {
        Period = ctrlPanl.getPeriod();
        EqnFlag = ctrlPanl.getEqnFlag();
        PlotMode = ctrlPanl.getPlotMode();
        ScanMode = ctrlPanl.getScanFlag();
        c = ctrlPanl.getCval();
        RunFlag = ctrlPanl.getRunFlag();
        boolean flag = false;
        if(ctrlPanl.getReset())
        {
            StatusMsg = "Graphs cleared, iteration process stopped and reset.";
            flag = true;
        }
        if(EqnFlag)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        x = 0.0D;
    else
        x = 0.01D;
    tg.Reset(EqnFlag);
    bg.Reset(EqnFlag, c);
    RepaintAfterReset = true;
    repaint();
}

if(!flag)
    if(!RunFlag)
        StatusMsg = "Applet is iterating the equation...";
    else
        StatusMsg = "Iteration process halted by user command.";
    showStatus(StatusMsg);
}

public void paint(Graphics g)
{
    g.drawImage(TG, 0, 0, null);
    g.drawImage(BG, 190, 175, null);
}

public void update(Graphics g)
{
    if(RepaintAfterReset)
    {
        paint(g);
        RepaintAfterReset = false;
        return;
    }
    x = bg.DoFormula(x, c, EqnFlag);
    if(bg.OutOfRange)
    {
        showStatus("Stopped because x has gone out of range of the graph.");
        StopEqn();
        return;
    }
    bg.Plot(g, c, x);
    if(!tg.Plot(g, x, ScanMode, PlotMode))
    {
        StopEqn();
        showStatus("Stopped because the single scan has been completed.");
    }
}

void StopEqn()
{
    ctrlPanel.Stop();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RunFlag = false;
}

public void run()
{
    do
    {
        if(RunFlag)
            repaint();
        long l = T - System.currentTimeMillis();
        if(l < 5L)
            l = 5L;
        try
        {
            Thread.currentThread();
            Thread.sleep(l);
        }
        catch(InterruptedException _ex) {}
        T = System.currentTimeMillis() + Period;
    } while(true);
}

public boolean action(Event event, Object obj)
{
    if(!ctrlPanel.action(event))
    {
        return false;
    }
    else
    {
        getCtrlPanelState();
        return true;
    }
}

public void start()
{
    Iter.resume();
}

public void stop()
{
    Iter.suspend();
}

public void destroy()
{
    if(Iter != null)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Iter.stop();
        Iter = null;
    }
}

public difeqnap()
{
    StatusMsg = "";
    ScanMode = false;
    RunFlag = true;
    PlotMode = false;
    EqnFlag = true;
    OutOfRange = false;
    RepaintAfterReset = false;
    Period = 500L;
    bgClr = Color.lightGray;
}

String StatusMsg;
boolean ScanMode;
double x;
double c;
boolean RunFlag;
boolean PlotMode;
boolean EqnFlag;
boolean OutOfRange;
boolean RepaintAfterReset;
timegr tg;
Image TG;
Graphics gt;
bouncegr bg;
Image BG;
Graphics gb;
ctrlpanl ctrlPanl;
Font font;
FontMetrics fm;
Thread Iter;
long Period;
long T;
Color bgClr;
Dimension d;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

timegr.java

```
import java.awt.*;

class timegr
{
    timegr(Graphics g, FontMetrics fontmetrics, Color color)
    {
        yMAX = 160;
        tMAX = 340;
        HB = 20;
        VB = 10;
        axis = Color.gray;
        gi = g;
        fm = fontmetrics;
        bgClr = color;
        trace = new Color(0, 128, 64);
    }

    void Reset(boolean flag)
    {
        t = 1;
        u = 61;
        gi.setColor(bgClr);
        gi.fillRect(0, 0, 390, 175);
        int i = 160;
        int j = 218;
        char c = '\u0160';
        if(flag)
        {
            i = 40;
            j = 298;
            char cl = '\u0110';
        }
        k = i;
        gi.setColor(axis);
        gi.drawLine(HB, VB, HB, VB + yMAX);
        Y = 160;
        if(flag)
            Y = 80;
        gi.drawLine(HB, VB + Y, HB + tMAX, VB + Y);
        String s = "";
        j = HB - 6;
        if(flag)
        {
            int j1 = 2;
            for(int l = 0; l < 161;)
            {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int i1 = VB + 1;
gi.drawLine(j, i1, HB, i1);
String s1 = String.valueOf(j1);
gi.drawString(s1, j - fm.stringWidth(s1), i1 + 3);
j = 40;
j1--;
}

} else
{
int i1 = 10;
for(int i1 = 0; i1 < 161; i1 += 32)
{
int i2 = VB + i1;
gi.drawLine(j, i2, HB, i2);
String s2;
if(k1 == 10)
s2 = "1";
else
if(k1 == 0)
s2 = "0";
else
s2 = "." + String.valueOf(k1);
gi.drawString(s2, j - fm.stringWidth(s2), i2 + 3);
i1 = 2;
}

}

char c2 = "254";
if(flag)
c2 = "W";
gi.setColor(Color.black);
gi.drawString("x", 18, 6);
gi.drawString("r", HB + tMAX + 4, c2);
y2 = Y;
}

boolean Plot(Graphics g, double d, boolean flag, boolean flag1)
{
y1 = y2;
y2 = Y - (int)(k * d);
Line(g, flag1);
Line(g, flag1);
if(flag)
{
Wipe(g);
Wipe(gi);
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if((u += 5) > tMAX)
    u = 1;
if((t += 5) > tMAX)
    if(flag)
    {
        t = 5;
    } else
    {
        if(!flag)
            return false;
        t = 1;
    }
return true;
}

void Line(Graphics g, boolean flag)
{
    int i = HB + 1;
    int j = VB + y2;
    g.setColor(trace);
    if(flag)
    {
        g.drawLine(i, j, i + 5, j);
        return;
    } else
    {
        g.drawLine(i, VB + y1, i + 5, j);
        return;
    }
}

void Wipe(Graphics g)
{
    g.setColor(bgClr);
    int i = VB + Y;
    g.fillRect(HB + u, VB, 7, yMAX);
    g.setColor(axis);
    g.drawLine(HB, i, HB + tMAX, i);
}

double k;
int t;
int u;
int y1;
int y2;
int Y;
int yMAX;
int tMAX;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int HB;

int VB;

Color bgClr;

Color axis;

Color trace;

Graphics gi;

FontMetrics fm;
}

```

bouncegr.java

```

import java.awt.Color;
import java.awt.Graphics;

class bouncegr
{
    bouncegr(Graphics g, Color color)
    {
        FTTF = false;
        MAX = 161;
        HB = 200;
        VB = 185;
        Hb = 10;
        Vb = 10;

        gi = g;
        bgClr = color;
        P = new int[MAX];
        Q = new int[MAX];
        R = new int[MAX];
        S = new int[MAX];
        axisColor = Color.gray;
        atClr = new Color(0, 255, 128);
        fgClr = new Color(0, 128, 64);
    }

    void Reset(boolean flag, double d)
    {
        FTTF = false;
        gi.setColor(bgClr);
        gi.fillRect(0, 0, 180, 180);
        if(flag)
        {
            X = 80;
            Y = 80;
            k = 40D;
            Amax = 80;
            Amin = 80;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

pa = 0;
P1 = -2D;
P2 = 2D;
dP = 0.0050000000000000001D;
} else
{
X = 0;
Y = 160;
k = 160D;
Amax = 160;
Amin = 0;
pa = 80;
P1 = 0.0D;
P2 = 1.0D;
dP = 0.001D;
}
gi.setColor(axisColor);
int i = Hb + X;
int l = Vb + Y;
gi.drawLine(i + Amin, l, i + Amax, l);
gi.drawLine(i - Amin, l, i - Amax, l);
gi.setColor(Color.black);
gi.drawString("x", i + Amax + 3, l + 2);
gi.drawString("y", i - 2, Vb - 5);
gi.setColor(axisColor);
gi.drawLine(i + Amin, l - Amin, i + Amax, l - Amax);
for(int i1 = 0; i1 < MAX; i1++)
{
R[i1] = Amax;
S[i1] = pa;
P[i1] = pa;
Q[i1] = Amin;
}

OutOfRange = false;
Pmax = 0;
Pmin = 160;
for(double d1 = P1; d1 <= P2; d1 += dP)
{
double d2 = DoFormula(d1, d, flag);
if(!OutOfRange)
{
int j = (int)(k * d1);
int j1 = (int)(k * d2);
int k1 = Y - j1;
if(k1 > Pmax)
Pmax = k1;
if(k1 < Pmin)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    Pmin = k1;
    if(j <= pa)
    {
        if(j < P[k1])
            P[k1] = j;
        if(j > Q[k1])
            Q[k1] = j;
    }
    if(j >= pa)
    {
        if(j < R[k1])
            R[k1] = j;
        if(j > S[k1])
            S[k1] = j;
    }
    j = Hb + X + j;
    j1 = Vb + k1;
    gi.drawLine(j, j1, j, j1);
}
OutOfRange = false;

double DoFormula(double d, double d1, boolean flag)
{
    if(flag)
        d = d * d + d1;
    else
        d = d1 * d * (1.0D - d);
    if(d < P1 || d > P2)
        OutOfRange = true;
    return d;
}

void Plot(Graphics g, double d, double d1)
{
    if(FTTF)
    {
        Lines(g, atClr);
        Lines(g, atClr);
    }
    x = y;
    y = (int)(k * d1);
    if(FTTF)
    {
        Lines(g, fgClr);
        Lines(g, fgClr);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

    FTTF = true;
}

void Lines(Graphics g, Color color)
{
    int i = HB + X;
    int j = VB + Y;
    int l = VB;
    if(g == gi)
    {
        i = Hb + X;
        j = Vb + Y;
        l = Vb;
    }

    int il = i + x;
    int jl = i + y;
    int kl = j - x;
    int ll = j - y;
    g.setColor(color);
    g.drawLine(il, kl, il, ll);
    g.drawLine(il, ll, jl, ll);
    if(color == atClr)
    {
        g.setColor(axisColor);
        for(int i2 = y - 3; i2 < y + 4; i2++)
        {
            int j2 = Y - i2;
            int k2 = l + j2;
            if(j2 <= Pmax && j2 >= Pmin)
            {
                g.drawLine(i + P[j2], k2, i + Q[j2], k2);
                g.drawLine(i + R[j2], k2, i + S[j2], k2);
            }
        }

        g.setColor(axisColor);
        g.drawLine(i + Amin, j, i + Amax, j);
        g.drawLine(i, j - Amin, i, j - Amax);
        g.setColor(Color.white);
        g.drawLine(il, kl, il, kl);
        g.drawLine(jl, ll, jl, ll);
    }
}

boolean FTTF;
boolean OutOfRange;
int P[];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int Q[];
int R[];
int S[];
int MAX;
int Amax;
int Amin;
int Pmax;
int Pmin;
int pa;
double xx;
double P1;
double P2;
double dP;
double k;
int x;
int y;
int X;
int Y;
int HB;
int VB;
int Hb;
int Vb;
Color axisColor;
Color atClr;
Color bgClr;
Color fgClr;
Graphics gi;
}

```

ctrlpanl.java

```

import java.awt.*;

public class ctrlpanl extends Panel
{

    ctrlpanl()
    {
        setLayout(null);
        setFont(new Font("Dialog", 0, 12));
        selPer = new selper();
        add(selPer);
        selPer.reshape(0, 0, 80, 110);
        selScan = new selscan();
        add(selScan);
        selScan.reshape(90, 58, 90, 56);
        selButt = new selbutt();
        add(selButt);
        selButt.reshape(0, 150, 90, 20);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

selPlot = new selplot(setButt);
add(selPlot);
selPlot.reshape(110, 116, 90, 56);
setCval = new setcval(setButt);
add(setCval);
setCval.reshape(0, 110, 60, 37);
incCval = new incCval(setCval);
add(incCval);
incCval.reshape(60, 128, 30, 18);
selEqn = new seleqn(setCval);
add(selEqn);
selEqn.reshape(90, 0, 90, 56);
setCval.SE = selEqn;
setCval.initCval();
}

boolean action(Event event)
{
    if((event.target instanceof Checkbox) && (selPer.event(event) || selScan.event(event) || selPlot.event(event) || selEqn.event(event)))
        return true;
    if((event.target instanceof TextField) && setCval.event(event))
        return true;
    return (event.target instanceof Button) && (incCval.event(event) || setButt.event(event));
}

long getPeriod()
{
    return selPer.getPeriod();
}

boolean getEqnFlag()
{
    return selEqn.getEqnFlag();
}

boolean getPlotMode()
{
    boolean flag = selPlot.getPlotMode();
    if(flag)
        selScan.setScanMode(false);
    return flag;
}

boolean getScanFlag()
{
    return selScan.getScanMode();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

double getCval()
{
    return setCval.getCval();
}

boolean getReset()
{
    return setButt.getResetStatus();
}

boolean getRunFlag()
{
    return setButt.getRunFlag();
}

void Start()
{
    setButt.Start();
}

void Stop()
{
    setButt.Stop();
}

selper selPer;
seleqn selEqn;
selscan selScan;
selplot selPlot;
setcval setCval;
incCval incCval;
setbutt setButt;
}

```

selper.java

```

import java.awt.*;

public class selper extends Panel
{
    selper()
    {
        Period = 100L;
        B = new Checkbox[5];
        setLayout(new GridLayout(6, 0));
        L = new Label("Plot every:", 0);
        G = new CheckboxGroup();
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

a = new Checkbox("2seconds", G, false);
b = new Checkbox("1second ", G, false);
c = new Checkbox("500ms", G, false);
d = new Checkbox("250ms", G, false);
e = new Checkbox("100ms", G, true);
add(L);
B[0] = (Checkbox)add(a);
B[1] = (Checkbox)add(b);
B[2] = (Checkbox)add(c);
B[3] = (Checkbox)add(d);
B[4] = (Checkbox)add(e);
}

boolean event(Event event1)
{
    Checkbox checkbox = G.getCurrent();
    if(event1.target == checkbox)
    {
        int i;
        for(i = 0; i < 5; i++)
            if(checkbox == B[i])
                break;

        Period = T[i];
        return true;
    } else
    {
        return false;
    }
}

long getPeriod()
{
    return Period;
}

Label L;
CheckboxGroup G;
Checkbox a;
Checkbox b;
Checkbox c;
Checkbox d;
Checkbox e;
Checkbox B[];
private long Period;
private long T[] = {
    2000L, 1000L, 500L, 250L, 100L
};
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

selper.java

```
import java.awt.*;

public class selper extends Panel
{
    selper()
    {
        Period = 100L;
        B = new Checkbox[5];
        setLayout(new GridLayout(6, 0));
        L = new Label("Plot every:", 0);
        G = new CheckboxGroup();
        a = new Checkbox("2seconds", G, false);
        b = new Checkbox("1second ", G, false);
        c = new Checkbox("500ms", G, false);
        d = new Checkbox("250ms", G, false);
        e = new Checkbox("100ms", G, true);
        add(L);
        B[0] = (Checkbox)add(a);
        B[1] = (Checkbox)add(b);
        B[2] = (Checkbox)add(c);
        B[3] = (Checkbox)add(d);
        B[4] = (Checkbox)add(e);
    }

    boolean event(Event event1)
    {
        Checkbox checkbox = G.getCurrent();
        if(event1.target == checkbox)
        {
            int i;
            for(i = 0; i < 5; i++)
                if(checkbox == B[i])
                    break;

            Period = T[i];
            return true;
        }
        else
        {
            return false;
        }
    }

    long getPeriod()
    {
        return Period;
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Label L;
CheckboxGroup G;
Checkbox a;
Checkbox b;
Checkbox c;
Checkbox d;
Checkbox e;
Checkbox B[];
private long Period;
private long T[] = {
    2000L, 1000L, 500L, 250L, 100L
};
}

```

seleqn.java

```

// Source File Name: seleqn.java

import java.awt.*;

public class seleqn extends Panel
{
    seleqn(setcval setcval1)
    {
        EqnFlag = false;
        SC = setcval1;
        setLayout(new GridLayout(3, 0));
        L = new Label("Equation:", 0);
        G = new CheckboxGroup();
        a = new Checkbox("x = cx(1 - x)", G, true);
        b = new Checkbox("x = x**2 + c", G, false);
        add(L);
        add(a);

        boolean event(Event event1)
        {
            Checkbox checkbox = G.getCurrent();
            if(event1.target != checkbox)
                return false;
            boolean flag = EqnFlag;
            if(checkbox == a)
                EqnFlag = false;
            else
                EqnFlag = true;
            if(flag != EqnFlag)
                SC.initCval();

            return true;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

boolean getEqnFlag()
{
    return EqnFlag;
}

setEval SC;
Label L;
CheckboxGroup G;
Checkbox a;
Checkbox b;
private boolean EqnFlag;
}

```

selscan.java

```

import java.awt.*;

public class selscan extends Panel
{
    selscan()
    {
        ScanMode = true;
        setLayout(new GridLayout(3, 0));
        L = new Label("Scan Mode:", 0);
        G = new CheckboxGroup();
        a = new Checkbox("single-scan", G, false);
        b = new Checkbox("continuous", G, true);
        add(L);
        add(a);
        add(b);
    }

    boolean event(Event event1)
    {
        Checkbox checkbox = G.getCurrent();
        if(event1.target == checkbox)
        {
            if(checkbox == a)
                ScanMode = false;
            else
                ScanMode = true;
            return true;
        }
        else
        {
            return false;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}

boolean getScanMode()
{
    return ScanMode;
}

void setScanMode(boolean flag)
{
    ScanMode = flag;
    if(ScanMode)
    {
        G.setCurrent(b);
        return;
    } else
    {
        G.setCurrent(a);
        return;
    }
}

Label L;
CheckboxGroup G;
Checkbox a;
Checkbox b;
private boolean ScanMode;
}

```

setcval.java

```

import java.awt.*;

public class setcval extends Panel
{
    setcval(setbutt setbutt1)
    {
        SB = setbutt1;
        setLayout(new GridLayout(2, 1));
        L = new Label("adjust c");
        add(L);
        t = new TextField();
        add(t);
    }

    boolean event(Event event1)
    {
        if(event1.target != t)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

return false;

double d = c;

try
{
    c = Double.valueOf(t.getText()).doubleValue();
}
catch(NumberFormatException _ex)
{
    c = d;
}

CheckLimits();

return true;
}

void UpDn(double d)
{
    c += d;
    CheckLimits();
}

void CheckLimits()
{
    if(SE.getEqnFlag())
    {
        if(c > 2.5D)
            c = 2.5D;
        if(c < -2.5D)
            c = -2.5D;
    } else
    {
        if(c > 4.0999999999999996D)
            c = 4.0999999999999996D;
        if(c < 0.10000000000000001D)
            c = 0.10000000000000001D;
    }

    ShowReset();
}

void initCval()
{
    if(SE.getEqnFlag())
        c = -1.5D;
    else
        c = 3.6800000000000002D;

    ShowReset();
}

void ShowReset()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    L.setText(String.valueOf(c));
    SB.Reset();
}

double getCval()
{
    return c;
}

setbutt SB;
seleqn SE;
Label L;
TextField t;
private double c;
}

```

inccval.java

```

import java.awt.*;

public class inccval extends Panel
{
    inccval(setcval setcval1)
    {
        SC = setcval1;
        setLayout(new GridLayout(1, 2));
        a = new Button("+");
        add(a);
        b = new Button("-");
        add(b);
    }

    boolean event(Event event1)
    {
        double d;
        if(event1.target == a)
            d = 0.10000000000000001D;
        else
            if(event1.target == b)
                d = -0.10000000000000001D;
            else
                return false;
        SC.UpDn(d);
        return true;
    }

    setcval SC;
    Button a;
    Button b;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

setbutt.java

```
import java.awt.*;

public class setbutt extends Panel
{

    setbutt()
    {
        RunFlag = true;
        ResetFlag = true;
        setLayout(new GridLayout(1, 3));
        b = new Button("Stop");
        add(b);
        c = new Button("Clear");
        add(c);
    }

    boolean event(Event event1)
    {
        if(event1.target == b)
            Relabel(RunFlag = !RunFlag);
        else
            if(event1.target == c)
                Reset();
            else
                return false;
        return true;
    }

    boolean getResetStatus()
    {
        if(ResetFlag)
        {
            ResetFlag = false;
            return true;
        }
        else
        {
            return false;
        }
    }

    boolean getRunFlag()
    {
        return RunFlag;
    }

    void Start()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Relabel(RunFlag = true);
}

void Stop()
{
    Relabel(RunFlag = false);
}

void Reset()
{
    RunFlag = false;
    Relabel(false);
    ResetFlag = true;
}

private void Relabel(boolean flag)
{
    if(flag)
    {
        b.setLabel("Stop");
        return;
    } else
    {
        b.setLabel("Start");
        return;
    }
}

Button b;
Button c;
Button d;
private boolean RunFlag;
private boolean ResetFlag;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Source Code ที่ใช้ในการสร้างโปรแกรม Hyper Henon Signal ประกอบไปด้วย Class ไฟล์ต่างๆ เหมือนกับโปรแกรม Logistic Signal โดยที่ต่างกันคือไฟล์ BouceGr.class ซึ่งมี source code ดังนี้

BouceGr.class

```
import java.awt.Color;
import java.awt.Graphics;

class bouncegr
{
    bouncegr(Graphics g, Color color)
    {
        FTTF = false;
        MAX = 161;
        HB = 200;
        VB = 185;
        Hb = 10;
        Vb = 10;
        gi = g;
        bgClr = color;
        P = new int[MAX];
        Q = new int[MAX];
        R = new int[MAX];
        S = new int[MAX];
        axisColor = Color.gray;
        atClr = new Color(0, 255, 128);
        fgClr = new Color(0, 128, 64);
    }

    void Reset(boolean flag, double d)
    {
        FTTF = false;
        gi.setColor(bgClr);
        gi.fillRect(0, 0, 180, 180);
        if(flag)
        {
            X = 80;
            Y = 80;
            k = 40D;
            Amax = 80;
            Amin = -80;
            pa = 0;
            P1 = -2D;
            P2 = 2D;
            dP = 0.0050000000000000001D;
        }
        else
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    X = 0;
    Y = 160;
    k = 160D;
    Amax = 160;
    Amin = 0;
    pa = 80;
    P1 = 0.0D;
    P2 = 1.0D;
    dP = 0.001D;
}

gi.setColor(axisColor);
int i = Hb + X;
int l = Vb + Y;
gi.drawLine(i + Amin, l, i + Amax, l);
gi.drawLine(i, l - Amin, i, l - Amax);
gi.setColor(Color.black);
gi.drawString("x", i + Amax + 3, l + 2);
gi.drawString("y", i - 2, l - 5);
gi.setColor(axisColor);
gi.drawLine(i + Amin, l - Amin, i + Amax, l - Amax);
for(int il = 0; il < MAX; il++)
{
    R[il] = Amax;
    S[il] = pa;
    P[il] = pa;
    Q[il] = Amin;
}

OutOfRange = false;
Pmax = 0;
Pmin = 160;
for(double d1 = P1; d1 <= P2; d1 += dP)
{
    double d2 = DoFormula(d1, d, flag);
    if(!OutOfRange)
    {
        int j = (int)(k * d1);
        int j1 = (int)(k * d2);
        int k1 = Y - j1;
        if(k1 > Pmax)
            Pmax = k1;
        if(k1 < Pmin)
            Pmin = k1;
        if(j <= pa)
        {
            if(j < P[k1])

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        P[k1] = j;
        if(j > Q[k1])
            Q[k1] = j;
    }
    if(j >= pa)
    {
        if(j < R[k1])
            R[k1] = j;
        if(j > S[k1])
            S[k1] = j;
    }
    j = Hb + X + j;
    j1 = Vb + k1;
    gi.drawLine(j, j1, j, j1);
}
OutOfRange = false;
}
}

double DoFormula(double d, double d1, boolean flag)
{
    if(flag)
        d = d * d + d1;
    else
        d = d1 * d * (1.0D - d);
    if(d < P1 || d > P2)
        OutOfRange = true;
    return d;
}

void Plot(Graphics g, double d, double d1)
{
    if(FTTF)
    {
        Lines(g, atClr);
        Lines(gi, atClr);
    }
    x = y;
    y = (int)(k * d1);
    if(FTTF)
    {
        Lines(g, fgClr);
        Lines(gi, fgClr);
    }
    FTTF = true;
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void Lines(Graphics g, Color color)
{
    int i = HB + X;
    int j = VB + Y;
    int l = VB;
    if(g == gi)
    {
        i = Hb + X;
        j = Vb + Y;
        l = Vb;
    }
    int il = i + x;
    int jl = i + y;
    int kl = j - x;
    int ll = j - y;
    g.setColor(color);
    g.drawLine(il, kl, il, ll);
    g.drawLine(il, ll, jl, ll);
    if(color == atClr)
    {
        g.setColor(axisColor);
        for(int i2 = y - 3; i2 < y + 4; i2++)
        {
            int j2 = Y - i2;
            int k2 = l + j2;
            if(j2 <= Pmax && j2 >= Pmin)
            {
                g.drawLine(i + P[j2], k2, i + Q[j2], k2);
                g.drawLine(i + R[j2], k2, i + S[j2], k2);
            }
        }
    }

    g.setColor(axisColor);
    g.drawLine(i + Amin, j, i + Amax, j);
    g.drawLine(i, j - Amin, i, j - Amax);
    g.setColor(Color.white);
    g.drawLine(il, kl, il, kl);
    g.drawLine(jl, ll, jl, ll);
}

boolean FTTF;
boolean OutOfRange;
int P[];
int Q[];
int R[];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int S[];
int MAX;
int Amax;
int Amin;
int Pmax;
int Pmin;
int pa;
double xx;
double P1;
double P2;
double dP;
double k;
int x;
int y;
int X;
int Y;
int HB;
int VB;
int Hb;
int Vb;
Color axisColor;
Color atClr;
Color bgClr;
Color fgClr;
Graphics gi;
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้