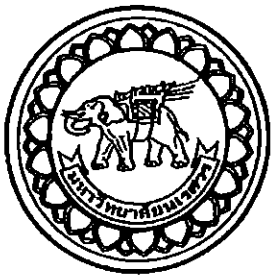


การเรียนรู้ด้วยวิธีเคอร์เนลแบบออนไลน์โดยใช้ส่วนต่อประสานกราฟิกกับผู้ใช้

LEARNING WITH ONLINE KERNEL METHOD USING GUI

นายอนุสรณ์ เริญไธสง รหัส 50362955

นายอภิวัฒน์ โค่นบ่อรัง รหัส 50362986

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 19 ส.ค. 2555
เลขทะเบียน 1973940
เลขเรียกหนังสือ ฟร.
มหาวิทยาลัยนเรศวร

A 2319
2553

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2553

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน การเรียนรู้ด้วยวิธีเดอร์เนลแบบออนไลน์โดยใช้ส่วนต่อประสานกราฟิกกับผู้ใช้

ผู้ดำเนินโครงงาน นายอนุสรณ์ เริญไชสง รหัส 50362955

นายอภิวัฒน์ โค่นบ่อรัง รหัส 50362986

ที่ปรึกษาโครงงาน ดร. ศุภวรรณ พลพิทักษ์ชัย

สาขาวิชา วิศวกรรมไฟฟ้า

ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์

ปีการศึกษา 2553

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

.....ที่ปรึกษาโครงงาน

(ดร. ศุภวรรณ พลพิทักษ์ชัย)

.....กรรมการ

(ดร. นิพัทธ์ จันทรมินทร์)

.....กรรมการ

(ดร. มุฑิตา สงฆ์จันทร์)

ชื่อหัวข้อโครงงาน	การเรียนรู้ด้วยวิธีเคอร์เนลแบบออนไลน์โดยใช้ส่วนต่อประสานกราฟิกกับผู้ใช้
ผู้ดำเนินโครงงาน	นายอนุสรณ์ เริญไชสง รหัส 50362955
	นายอภิวัฒน์ โค่นบ่อรัง รหัส 50362986
ที่ปรึกษาโครงงาน	ดร. ศุภวรรณ พลพิทักษ์ชัย
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2553

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ได้ศึกษาการเรียนรู้ด้วยวิธีเคอร์เนลแบบออนไลน์โดยใช้ส่วนต่อประสานกราฟิกกับผู้ใช้ ซึ่งนำมาประยุกต์ใช้สำหรับหาฟังก์ชันการประมาณค่าแบบออนไลน์ด้วยวิธีเคอร์เนลผ่านกลุ่มข้อมูลที่เรียกว่ากลุ่มสอน ส่วนประสิทธิภาพของการเรียนรู้จะถูกวัดด้วยค่าความผิดพลาดเฉลี่ยกำลังสองผ่านกลุ่มข้อมูลที่เรียกว่ากลุ่มทดสอบ ซึ่งในการทดลองนั้นจะมีการใช้รูปแบบของข้อมูลและฟังก์ชันเคอร์เนลที่หลากหลาย เมื่อนำส่วนต่อประสานกราฟิกกับผู้ใช้เข้ามาช่วยในการหาค่าความผิดพลาดเฉลี่ยกำลังสองแล้ว จะทำให้สามารถเลือกรูปแบบของข้อมูลได้และสามารถเปลี่ยนฟังก์ชันเคอร์เนลง่าย ทำให้การใช้งานทำได้รวดเร็วและสะดวกมากยิ่งขึ้น นอกจากนั้นการทดลองยังแสดงให้เห็นถึงผลของค่าพารามิเตอร์แต่ละตัว เช่น อัตราการเรียนรู้ เรกกูลาไรเซชันพารามิเตอร์ ความกว้างของเคอร์เนลฟังก์ชัน ว่ามีผลต่อความผิดพลาดเฉลี่ยกำลังสองอย่างไรบ้าง

Project title	Learning with Online Kernel Method using GUI
Name	Mr. Anusorn Remthaisong ID. 50362955
	Mr. Apiwat Koneborrang ID. 50362986
Project advisor	Ms. Supawan Ponpitakchai, Ph.D.
Major	Electrical Engineering
Department	Electrical and Computer Engineering
Academic year	2010

Abstract

This project studies online learning with kernel method using GUI (Graphical User Interface). Approximation function can be investigated with online kernel method through training data. The performance of learning is measured with MSE (Mean Square Error) of validation data. Using GUI in the experiments, types of training data and kernel function can be selected easier than previous. Moreover, the experiments also show the affects of parameters such as learning rate, regularization parameter and kernel width to MSE.

กิตติกรรมประกาศ

โครงงานวิศวกรรมเรื่องการเรียนรู้ด้วยวิธีเดอร์เนลแบบออนไลน์โดยใช้ส่วนต่อประสานกราฟิกกับผู้ใช้สำเร็จไปได้ด้วยดีนั้น ทั้งนี้ต้องขอขอบคุณ ดร. ศุภวรรณ พลพิทักษ์ชัย ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงานที่คอยให้คำแนะนำ คำปรึกษา ตลอดจนข้อคิดเห็นต่าง ๆ อันเป็นประโยชน์อย่างมากในการทำโครงงานในครั้งนี้

เหนือสิ่งอื่นใด คณะผู้ดำเนินโครงงานขอกราบขอบพระคุณบิดามารดาที่ได้มอบความรักความอบอุ่น ตลอดจนค่าเล่าเรียน และเป็นกำลังใจให้รวมทั้งเป็นผู้ให้ทุกสิ่งทุกอย่างตั้งแต่เยาว์วัยจวบจนถึงปัจจุบัน จนทำให้ได้รับความสำเร็จอย่างทุกวันนี้ ทางคณะผู้ดำเนินโครงงานจะไม่ลืมพระคุณของทุกท่านที่ได้กล่าวมาข้างต้นตลอดไป

นายอนุสรณ์ เริญไชสง
นายอภิวัฒน์ โค่นบ่อรัง

สารบัญ

หน้า

ใบรับรองปริญญานิพนธ์ ..... ก
บทคัดย่อภาษาไทย ..... ข
บทคัดย่อภาษาอังกฤษ ..... ค
กิตติกรรมประกาศ ..... ง
สารบัญ ..... จ
สารบัญตาราง ..... ช
สารบัญรูป ..... ซ

บทที่ 1 บทนำ 1
1.1 ที่มาและความสำคัญของโครงงาน 1
1.2 วัตถุประสงค์ของโครงงาน 2
1.3 ขอบเขตของโครงงาน 2
1.4 ขั้นตอนและแผนการดำเนินงาน 3
1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงงาน 3
1.6 งบประมาณ 3

บทที่ 2 การประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนล 4
2.1 การประมาณค่าฟังก์ชันในอาร์เคเอชเอส 4
2.2 การประมาณค่าฟังก์ชันในอาร์เคเอชเอสแบบออนไลน์ 6
2.3 ส่วนต่อประสานกราฟิกกับผู้ใช้ 8
2.4 กระบวนการออกแบบส่วนต่อประสานกราฟิกผู้กับใช้ 10

บทที่ 3 วิธีการประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนลโดยใช้ส่วนต่อประสานกราฟิกกับผู้ใช้ 11
3.1 หลักการออกแบบการประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนล 11
3.2 การออกแบบและการใช้โปรแกรม 13

สารบัญ (ต่อ)

หน้า

บทที่ 4 ผลการทดลองและการวิเคราะห์ผล 18

4.1 ผลการทดลองการประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนล 18

4.1.1 การประมาณค่าโดยใช้เคอร์เนลฟังก์ชันเป็นฟังก์ชันเกาเสียนเรเดียลเบสิส 20

4.1.2 การประมาณค่าโดยใช้เคอร์เนลฟังก์ชันเป็นซิงค์ฟังก์ชัน 29

บทที่ 5 สรุปและข้อเสนอแนะ 39

5.1 สรุปผลการทดลอง 39

5.1.1 การหาค่าพารามิเตอร์ที่ดีที่สุดของการประมาณค่าโดยใช้เคอร์เนล ฟังก์ชันเป็นฟังก์ชันเกาเสียนเรเดียลเบสิส 39

5.1.2 การหาค่าพารามิเตอร์ที่ดีที่สุดของการประมาณค่าโดยใช้เคอร์เนล ฟังก์ชันเป็นซิงค์ฟังก์ชัน 40

5.2 ปัญหาและแนวทางแก้ไข 40

เอกสารอ้างอิง 41

ภาคผนวก 42

ประวัติผู้ดำเนินโครงงาน 61

สารบัญตาราง

ตารางที่ หน้า

4.1 ค่าพารามิเตอร์สำหรับเคอร์เนลฟังก์ชันแบบเกาเสียนเรเดียลเบสิส 20
4.2 ค่าพารามิเตอร์สำหรับเคอร์เนลฟังก์ชันแบบซิงค์ฟังก์ชัน 30
5.1 ค่าพารามิเตอร์ที่เหมาะสมโดยใช้เคอร์เนลฟังก์ชันเป็นฟังก์ชันเกาเสียนเรเดียลเบสิส 39
5.2 ค่าพารามิเตอร์ที่เหมาะสมโดยใช้เคอร์เนลฟังก์ชันเป็นซิงค์ฟังก์ชัน 40

สารบัญรูป

รูปที่ หน้า

2.1 แผนภาพลักษณะการทำงานของส่วนต่อประสานกราฟิกกับผู้ใช้ 8
2.2 ลำดับขั้นตอนการออกแบบส่วนต่อประสานกราฟิกกับผู้ใช้ 10
3.1 ลำดับขั้นตอนการประมาณค่าฟังก์ชัน 12
3.2 หน้าต่างการทำงานของโปรแกรมเมื่อเลือก Selecting data 13
3.3 หน้าต่างการทำงานของโปรแกรมเมื่อเลือก Reproducing kernel 14
3.4 หน้าต่างการทำงานของโปรแกรมเมื่อเลือก Learning 15
3.5 หน้าต่างการทำงานของโปรแกรมเมื่อเลือก Training model 16
3.6 หน้าต่างการทำงานของโปรแกรมเมื่อเลือก Saving data 17
4.1 ข้อมูลที่ใช้สอน 19
4.2 การประมาณค่าด้วยวิธีเคอร์เนลเป็นฟังก์ชันเกาเสียนเรเดียลเบสิส
สำหรับข้อมูลที่ใช้สอนแบบ Sinc function 22
4.3 การประมาณค่าด้วยวิธีเคอร์เนลเป็นฟังก์ชันเกาเสียนเรเดียลเบสิส
สำหรับข้อมูลที่ใช้สอนแบบ Exponential function 24
4.4 การประมาณค่าด้วยวิธีเคอร์เนลเป็นฟังก์ชันเกาเสียนเรเดียลเบสิส
สำหรับข้อมูลที่ใช้สอนแบบ Product of exponential and sine function 25
4.5 การประมาณค่าด้วยวิธีเคอร์เนลเป็นฟังก์ชันเกาเสียนเรเดียลเบสิส
สำหรับข้อมูลที่ใช้สอนแบบ Polynomial function 27
4.6 การประมาณค่าด้วยวิธีเคอร์เนลเป็นฟังก์ชันเกาเสียนเรเดียลเบสิส
สำหรับข้อมูลที่ใช้สอนแบบ Paley-Wiener function 28
4.7 การประมาณค่าด้วยวิธีเคอร์เนลเป็นซิงค์ฟังก์ชัน
สำหรับข้อมูลที่ใช้สอนแบบ Sinc function 31
4.8 การประมาณค่าด้วยวิธีเคอร์เนลเป็นซิงค์ฟังก์ชัน
สำหรับข้อมูลที่ใช้สอนแบบ Exponential function 33
4.9 การประมาณค่าด้วยวิธีเคอร์เนลเป็นซิงค์ฟังก์ชัน
สำหรับข้อมูลที่ใช้สอนแบบ Product of exponential and sine function 34
4.10 การประมาณค่าด้วยวิธีเคอร์เนลเป็นซิงค์ฟังก์ชัน
สำหรับข้อมูลที่ใช้สอนแบบ Polynomial function 36

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.11 การประมาณค่าด้วยวิธีเคอร์เนลเป็นซิงค์ฟังก์ชัน
สำหรับข้อมูลที่ใช้สอนแบบ Paley-Wiener function 37

มหาวิทยาลัยนเรศวร

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

เครื่องการเรียนรู้ (Machine learning) เป็นการเรียนรู้เกี่ยวกับการพัฒนาเทคนิควิธีเพื่อให้คอมพิวเตอร์สามารถเรียนรู้ โดยเน้นที่วิธีการเพื่อสร้างความสัมพันธ์ของข้อมูลขาเข้าและขาออก การวิเคราะห์ข้อมูลที่กำหนดให้ที่กล่าวถึงนี้ก็มีการจำแนกประเภท (Classification) การทำนายค่า (Prediction หรือ Regression) การคำนวณการกระจายของข้อมูล (Density estimation) และการเลือกค่าแบบจำลอง (Model selection) การหาความสัมพันธ์ของข้อมูลด้านทฤษฎีส่วนมากในเครื่องการเรียนรู้จะสนใจการวิเคราะห์ข้อมูลที่มีจำนวนจำกัด

การเรียนรู้หมายถึงการหาความสัมพันธ์ระหว่างข้อมูลขาเข้าและขาออก ซึ่งข้อมูลที่ได้นั้นจะต้องมีค่าความผิดพลาดน้อยที่สุดและเป็นค่าที่ดีที่สุดสามารถนำไปใช้งานได้จริง โดยทั่วไปแล้วการเรียนรู้จากข้อมูลจะใช้แบบข้อมูลทั้งหมด (Batch learning) ในการสอน (Train) เพียงครั้งเดียว อย่างไรก็ตามวิธีนี้จะมีข้อเสียในกรณีที่ข้อมูลมีจำนวนมากคือมีข้อมูลเท่าไรก็จะป้อนเท่านั้น การทำแบบนี้จะทำให้คอมพิวเตอร์มีการประมวลผลช้าและอาจจะทำให้มีค่าความผิดพลาดที่มากหรืออาจจะทำให้ระบบหยุดการทำงานได้เลย ดังนั้นเราจึงใช้การเรียนรู้แบบออนไลน์ (Online learning) ซึ่งวิธีนี้จะใช้ข้อมูลเข้าไปทีละตัว โดยจะมีการสร้างแบบจำลอง (Model) ของแต่ละข้อมูลเอาไว้เพื่อจะหาค่าความผิดพลาดว่ามีค่ามากน้อยเพียงใด และจะทำให้คอมพิวเตอร์ของเราทำงานได้อย่างมีประสิทธิภาพเนื่องจากการป้อนข้อมูลไปทีละตัว นอกจากนั้นยังมีการเรียนรู้ชนิดการเรียนรู้แบบมีผู้สอน (Supervised learning) และการเรียนรู้แบบไม่มีผู้สอน (Unsupervised learning) การเรียนรู้แบบมีผู้สอน (Supervised learning) เป็นเทคนิคหนึ่งของการเรียนรู้ของเครื่องซึ่งสร้างฟังก์ชันจากข้อมูลสอน (Training data) ข้อมูลสอนประกอบด้วยข้อมูลขาเข้าและผลที่ต้องการ ผลจากการเรียนรู้จะเป็นฟังก์ชันที่ให้ค่าต่อเนื่อง (เรียกวิธีการว่าการถดถอยหรือ Regression) หรือฟังก์ชันการทำนายประเภทของวัตถุ (เรียกว่าการแบ่งประเภท หรือ Classification) ส่วนการเรียนรู้แบบไม่มีผู้สอน (Unsupervised learning) เป็นเทคนิคหนึ่งของเครื่องการเรียนรู้ โดยการสร้างโมเดลที่เหมาะสมกับข้อมูล การเรียนรู้แบบนี้แตกต่างจากการเรียนแบบมีผู้สอน คือ จะไม่มีการระบุผลที่ต้องการหรือประเภทไว้ก่อน และจะพิจารณาวัตถุเป็นเซตของตัวแปรสุ่ม แล้วจึงสร้างโมเดลความหนาแน่นร่วมของชุดข้อมูล การเรียนรู้แบบไม่มีผู้สอนสามารถนำไปใช้ร่วมกับการอนุมานแบบเบย์ เพื่อหาความน่าจะเป็นแบบมีเงื่อนไขของตัวแปรสุ่มโดยกำหนดตัวแปรที่เกี่ยวข้องให้ นอกจากนี้ยังสามารถนำไปใช้ในการบีบอัดข้อมูลได้

ในปี ค.ศ. 1980 ได้มีผู้เสนอการเรียนรู้วิธีเคอร์เนล (Kernel method) ซึ่งมีข้อแตกต่างกับวิธีการเรียนรู้แบบเดิมเช่น วิธีโครงข่ายในแง่ประสาท (Neural Network) หรือโครงข่ายใยประสาทเสมือน (Artificial Neural Network: ANN) การหาค่าที่เหมาะสมที่สุด (Optimization) สามารถทำได้ด้วยวิธีเคอร์เนลจะไม่มีปัญหาด้านค่าที่ต่ำสุดเฉพาะที่ (local minima) ในปัจจุบันนี้การเรียนรู้วิธีเคอร์เนล ถูกนำไปใช้ในงานต่าง ๆ อาทิเช่น เอสวีเอ็ม (SVM-support vector machine) การวิเคราะห์ของความสัมพันธ์ของข้อมูลต่าง ๆ เช่น การจัดกลุ่ม การจัดอันดับ องค์ประกอบหลัก การจำแนกประเภท เป็นต้น

โครงงานนี้เป็นการศึกษาทฤษฎีของการเรียนรู้แบบออนไลน์ด้วยวิธีเคอร์เนล โดยจะมุ่งเน้นในการศึกษาการนำข้อมูลต่าง ๆ ซึ่งเป็นข้อมูลจริงมาวิเคราะห์ประสิทธิภาพที่ได้โดยใช้โปรแกรมแมทแลป (Matlab) โดยการสร้างส่วนต่อประสานกราฟิกกับผู้ใช้เช่น การป้อนค่าพารามิเตอร์ต่าง ๆ การเลือกเคอร์เนลฟังก์ชัน ซึ่งประสิทธิภาพของการเรียนรู้จะถูกแสดงให้เห็นว่าขึ้นอยู่กับการปรับค่าพารามิเตอร์ต่าง ๆ ของแบบจำลอง

1.2 วัตถุประสงค์ของโครงงาน

โครงงานนี้มีวัตถุประสงค์เพื่อศึกษาทฤษฎีและประสิทธิภาพของการเรียนรู้แบบออนไลน์ด้วยวิธีเคอร์เนล โดยการนำไปใช้วิเคราะห์ข้อมูลต่าง ๆ และศึกษาถึงพารามิเตอร์ต่าง ๆ ของแบบจำลองเคอร์เนล และสร้างส่วนต่อประสานกราฟิกกับผู้ใช้เพื่อให้การวิเคราะห์ข้อมูลและศึกษาถึงค่าพารามิเตอร์ต่าง ๆ ทำได้ง่ายขึ้น

1.3 ขอบเขตของโครงงาน

1) ศึกษาเรื่องทฤษฎีของการเรียนรู้ด้วยวิธีเคอร์เนลแบบออนไลน์
2) ศึกษาปริญญานิพนธ์เรื่องการเรียนรู้ด้วยวิธีเคอร์เนลแบบออนไลน์
3) นำข้อมูลมาวิเคราะห์ด้วยวิธีเคอร์เนลโดยสร้างขั้นตอนวิธีบนโปรแกรมแมทแลป
4) ศึกษาผลกระทบที่ได้จากการปรับค่าของพารามิเตอร์แต่ละตัว
5) สร้างส่วนต่อประสานกราฟิกกับผู้ใช้สำหรับการป้อนค่าพารามิเตอร์ต่าง ๆ

1.4 ขั้นตอนและแผนการดำเนินงาน

รายละเอียด	ปี 2553						ปี 2554		
	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1) รวบรวมข้อมูล									
2) ศึกษาทฤษฎีทั่วไปของการเรียนรู้แบบออนไลน์ด้วยวิธีเดอร์เนล									
3) ดำเนินการวิเคราะห์ผลที่ได้จากวิธีการเรียนรู้แบบออนไลน์ด้วยวิธีเดอร์เนล									
4) สรุปผลการดำเนินงาน									
5) จัดทำปริญญานิพนธ์ฉบับสมบูรณ์									

1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงงาน

1) เข้าใจถึงเรื่องทฤษฎีของการเรียนรู้ด้วยวิธีเดอร์เนลแบบออนไลน์
2) เข้าใจถึงผลกระทบของพารามิเตอร์แต่ละตัวในการเรียนรู้ด้วยวิธีเดอร์เนลแบบออนไลน์
3) สามารถใช้โปรแกรมแมทแลปในการวิเคราะห์ข้อมูลได้
4) สามารถนำความรู้ด้านทฤษฎีของการเรียนรู้ด้วยวิธีเดอร์เนลแบบออนไลน์ไปใช้ในขั้นสูงต่อไป
5) ได้ส่วนต่อประสานกราฟิกกับผู้ใช้สำหรับป้อนค่าพารามิเตอร์ต่าง ๆ
6) สามารถนำไปประยุกต์ใช้กับข้อมูลจริงได้

1.6 งบประมาณ

1) ค่าถ่ายเอกสารและค่าเข้าเล่มปริญญานิพนธ์ฉบับสมบูรณ์	1,000	บาท
2) ค่าพิมพ์เอกสาร	500	บาท
3) วัสดุอุปกรณ์คอมพิวเตอร์	500	บาท
รวมเป็นเงินทั้งสิ้น (สองพันบาทถ้วน)	2,000	บาท

หมายเหตุ: ถัวเฉลี่ยทุกรายการ

บทที่ 2
การประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนล

ในบทนี้จะนำเสนอแนวทางในการแก้ไขปัญหาที่สามารถใช้ในงานแตกต่างกัน อาทิเช่น การประมวลผลสัญญาณการควบคุมเครื่องกลและการประมาณค่าฟังก์ชัน โดยส่วนใหญ่แล้วเราจะมองถึงปัญหาเหล่านี้ในรูปแบบของการประมาณการระบบที่ไม่รู้จักโดยอาศัยข้อมูลตัวอย่าง ในปัจจุบันนี้วิธีอาร์เคเอชเอส (RKHS – Reproducing kernel HilbertSpaces) ได้ถูกศึกษาและนำไปใช้กับปัญหาต่าง ๆ ที่ได้กล่าวมาแล้ว ดังนั้นในบทนี้จะเป็นการแสดงทฤษฎีที่เกี่ยวข้องกับการแก้ปัญหาด้วยวิธีเคอร์เนลแบบออนไลน์ นั้นคือเป็นการหาฟังก์ชันการประมาณค่าที่เป็นตัวแทนของระบบที่เราไม่รู้จักจากจำนวนข้อมูลจำกัด ซึ่งสามารถเก็บค่าได้จากระบบนั้น ๆ

2.1 การประมาณค่าฟังชันในอาร์เคเอชเอส [3]

สมมติว่าฟังก์ชันไม่รู้ค่า f ซึ่งสามารถสังเกตค่าจำนวนจำกัดได้ เป็นส่วนหนึ่งของอาร์เคเอชเอส $\mathcal{F}$ และ f นิยามบนเซต $\mathcal{X}$ ที่สามารถถือได้ว่าเป็นข้อมูลขาเข้าโดยที่ $x \in \mathcal{X}$ ดังนั้น $f(x)$ จะสามารถแสดงถึงการประมาณค่าของ f ที่ x และเซตข้อมูลขาเข้า $\mathcal{X}$ จะถูกมองว่าเป็นซับเซตปริภูมิยูคลิเดียน (Euclidian space) นั่นคือ $\mathcal{X} \subseteq R^n$ โดยที่ x แต่ละตัวเป็นเวกเตอร์มิติ n

เซตจำกัดของการสังเกตหรือข้อมูลขาออกของฟังก์ชัน $\{z_i\}_{i=1}^N$ จะสอดคล้องกับข้อมูลขาเข้า $\{x_i\}_{i=1}^N$ โดยเราสามารถกล่าวได้ว่า เซตของข้อมูลขาออกอยู่ในปริภูมิ Z ถ้าไม่มีข้อผิดพลาดค่าการสังเกตจะแสดงได้ดังนี้

$$z_i = L_i f \tag{2.1}$$

เมื่อ $\{L_i\}_{i=1}^N$ เป็นเซตของการประเมินผลฟังก์ชันนัลแบบเชิงเส้น (Linear evaluation functional) กำหนดบน $\mathcal{F}$ ซึ่งสอดคล้องกับ f เราสามารถแสดงค่าทั้งหมดของการสังเกต $\{z_i\}_{i=1}^N$ ได้ดังนี้

$$z = Lf = \sum_{i=1}^{N} (L_i f) s_i \tag{2.2}$$

โดยที่ $s_i \in R^N$ เป็นเวกเตอร์มาตรฐาน (Standard basis vector) ที่ลำดับ i โดยในทางปฏิบัติสามารถเขียนได้เป็น

$$z_i = f(x_i) \tag{2.3}$$

ซึ่งสามารถนำไปใช้ในปัญหาการประมาณค่า

ปัญหาฟังก์ชันการประมาณค่า ซึ่งเป็นตัวแทนฟังก์ชันที่ไม่รู้ค่า สามารถกระทำได้โดยกำหนดคลาสของ $\mathcal{F}$ ของฟังก์ชัน และค่าสังเกต $\{z_i\}_{i=1}^N$ ของฟังก์ชันนัล L_i ซึ่งถูกกำหนดบน $\mathcal{F}$ ดังนั้นจะสามารถหาฟังก์ชัน f บน $\mathcal{F}$ ซึ่งสอดคล้องกับสมการที่ (2.1) และ (2.3)ได้

ตามหลักแล้วเราสามารถกำหนดอาร์เคเอชเอสให้เป็นปริภูมิฮิลเบิร์ต (Hilbert space) ของฟังก์ชันบน $\mathcal{X}$ ด้วยคุณสมบัติที่ว่าในแต่ละ $x \in \mathcal{X}$ ฟังก์ชันนัล L_i ซึ่งเชื่อมโยง f ด้วย $f(x_i)$ นั่นคือ $L_i \to f(x_i)$ จะเป็นฟังก์ชันนัลแบบเส้นตรงที่ถูกกำหนดขอบเขตและการมีขอบเขตหมายถึงการมีของค่า M โดยที่

$$|L_i f| = |f(x_i)| \le M\|f\| \quad \text{สำหรับทุกๆ } f \text{ ในอาร์เคเอชเอส}$$

เมื่อ $\|\cdot\|$ เป็นค่านอร์มในอาร์เคเอชเอส

เมื่อฟังก์ชันนัล L_i ถูกกำหนดขอบเขต ตามทฤษฎีบทของรีซ (Riesz representation theorem) เราสามารถแสดงค่าการสังเกตได้ดังนี้

$$L_i f = \langle f, k_i \rangle, \quad i = 1,\ldots,N \tag{2.4}$$

เมื่อ $\langle \cdot,\cdot \rangle$ หมายถึงผลคูณภายใน (Inner product) ใน $\mathcal{F}$ และ $\{k_i\}_{i=1}^N$ เป็นชุดของฟังก์ชัน ที่เรียกว่ารีโพรดิวซิ่งเคอร์เนล (Reproducing kernels) ซึ่งแต่เป็นส่วนหนึ่งของ $\mathcal{F}$ และจะไม่ถูกกำหนดซ้ำกันโดยฟังก์ชันนัล L_i

ดังนั้นการแก้ปัญหาฟังก์ชันการประมาณค่าจะสามารถกำหนดใหม่ได้อีกแบบนั่นคือ กำหนดปริภูมิฮิลเบิร์ต $(\mathcal{F})$ ของเซตของฟังก์ชัน $\{k_i\}_{i=1}^N \subset \mathcal{F}$ และค่าการสังเกต $\{z_i\}_{i=1}^N$ จากเงื่อนไขดังกล่าวมาแล้วสามารถหาฟังก์ชันการประมาณค่า $f \in \mathcal{F}$ ที่สอดคล้องกับสมการที่ (2.4)

ในทุกอาร์เคเอชเอสจะมีฟังก์ชันบวกแน่นอน (Positive definite function) ที่เรียกว่ารีโพรดิวซิ่งเคอร์เนล (k) ที่กำหนดบน $\mathcal{X} \times \mathcal{X}$ ในการให้คำนิยามของอาร์เคเอชเอสในรูปฟังก์ชันนัลที่จำกัดเขตเชิงเส้นจะใช้คุณสมบัติดังนี้ ซึ่งกำหนดบนอาร์เคเอชเอส นั่นคือทุก ๆ $x, x' \in \mathcal{X}$จะได้ $k(\cdot, x')$ เป็นฟังก์ชันที่กำหนดไว้บน $\mathcal{X}$ โดยที่ค่าที่ x มีค่าเท่ากับ $k(x, x')$ สำหรับทุก f ใน $\mathcal{F}$ จะได้

(i) $k(\cdot, x') \in \mathcal{F}$; และ

(ii) $\langle f, k(\cdot, x') \rangle_{\mathcal{F}} = f(x')$

จากคุณสมบัติที่กล่าวมาแล้วแสดงให้เห็นว่า ฟังก์ชันรีโพรดิวซิ่งเคอร์เนล k_i เป็นฟังก์ชันบน $\mathcal{X} \times \mathcal{X}$ แต่อย่างไรก็ตาม เราสามารถแสดงเคอร์เนลฟังก์ชันในรูปแบบของฟังก์ชันบน $\mathcal{X}$ เท่านั้นได้ด้วยการพิจารณาฟังก์ชันเคอร์เนล $k(x, x_i)$ เมื่อ x_i เป็นจุดที่ไม่เปลี่ยนแปลงค่าและ

$x_i \in \mathcal{X}$ ดังนั้นการเขียน $k(x, x_i) = k_i(x)$ (หรือ k_i) จะสามารถสรุปได้ว่า $k_i \in \mathcal{F}$ บน $\mathcal{X}$ มีจุดศูนย์กลางบน x_i โดยส่วนใหญ่แล้วจุด x_i มักจะถูกกำหนดให้เป็นศูนย์กลางของฟังก์ชันเคอร์เนลและยังเป็นไฮเปอร์พารามิเตอร์ (Hyper parameter) ของฟังก์เคอร์เนลอีกด้วย ซึ่งโดยทั่วไปแล้วศูนย์ กลางเหล่านี้จะสอดคล้องกับข้อมูลขาเข้า ดังนั้นเราจึงสามารถแสดงฟังก์ชัน f ทั่ว ๆ ไปบนอาร์เคเอชเอส $\mathcal{F}$ ด้วยรีโพรดิวซิ่งเคอร์เนล k ได้ดังนี้

$$f(x) = \sum_i \alpha_i k(x, x_i) = \sum_i \alpha_i k_i(x) \tag{2.5}$$

เมื่อ $\alpha_i \in R, i \in N$ เพราะฉะนั้นการแก้ปัญหาการประมาณค่าฟังก์ชัน จะเปลี่ยนรูปเป็นการประมาณค่าที่เหมาะสมสำหรับตัวแปร α_i ในสมการที่ (2.5)

ฟังก์ชันที่ถือว่าเป็นฟังก์ชันเคอร์เนลนั้นมีหลายชนิดที่นิยมใช้ยกตัวอย่างเช่น ฟังก์ชันเกาเสียนเรเดียลเบสิสที่ใช้ในระบบข่ายประสาท [6], [7]

$$k(x, x') = \exp(-\beta \|x - x'\|^2) \tag{2.6}$$

โดยที่ $\beta > 0$

ฟังก์ชันโพลิโนเมียลของค่าชุดของค่า x กำลัง d แสดงได้โดย

$$k(x, x') = \sum_{k=0}^{d} \frac{1}{k! \rho_k} (xx')^k \tag{2.7}$$

เมื่อ ρ_k คือเซตของค่าคงที่ที่เรียกว่าค่าน้ำหนักและมีค่าเท่ากับ

$$k(x, x') = (1 + xx')^d \tag{2.8}$$

ฟังก์ชันเคอร์เนลนั้นสามารถแสดงได้ด้วยปริภูมิพาเลย์–เวียเนอร์ของฟังก์ชันจำกัดแถบ (Band limited function) หรือซิงค์ฟังก์ชัน (Sinc function)

$$k(x, x') = \frac{\sin \beta\pi(x - x')}{\pi(x - x')} \tag{2.9}$$

2.2 การประมาณค่าฟังก์ชันในอาร์เคเอชเอสแบบออนไลน์ [3], [6]

ในกรณีนี้ได้ตั้งสมมติฐานว่าในแต่ละรอบของการทำซ้ำ เราจะทราบค่าสังเกตเพียงแค่ค่าเดียวนั่นคือ z_n ดังนั้น

$$L_n f = z_n \tag{2.10}$$

จากนั้นเราจะได้ฟังก์ชันนัล $\hat{g}_{reg}$ ที่เวลา n ที่เป็นฟังก์ชันไม่เป็นค่าลบ โดยที่ $\hat{g}_{reg}$: $Z \to R$

$$\hat{g}_{reg}(f_n) = \frac{1}{2}\|L_{n+1}f_n - z_{n+1}\|^2 + \frac{\rho}{2}\|f_n\|^2 \tag{2.11}$$

ตั้งค่าเริ่มต้น f_0 ซึ่งโดยส่วนใหญ่จะให้มีค่าเท่ากับศูนย์ จากนั้นเราจะหาค่า f_n ที่จะทำให้สมการที่ (2.11) มีค่าต่ำที่สุดโดยอาศัยวิธีสโตแคสติกเกรเดียนเดสเซนต์ (Stochastic gradient descent, SGD) ที่แสดงโดย

$$f_{n+1} = f_n - \eta_n \nabla \hat{g}_{reg}(f_n) \tag{2.12}$$

เมื่อที่ $\nabla \hat{g}_{reg}(f_n)$ เป็นเกรเดียน ณ เวลาปัจจุบันของ $\hat{g}_{reg}$ เทียบกับ f_n ดังนั้น

$$\begin{aligned}\hat{g}_{reg}(f_n) &= L^*_{n+1}L_{n+1}f_n - L^*_{n+1}z_{n+1} + \rho f_n \\ &= L^*_{n+1}(L_{n+1}f_n - z_{n+1}) + \rho f_n\end{aligned}$$

แทนค่าในสมการที่ (2.12) จะได้ว่า

$$f_{n+1} = (1-\eta_n\rho)f_n - \eta_n L^*_{n+1}(L_{n+1}f_n - z_n) \tag{2.13}$$

โดยที่สำหรับค่าคงที่ใด ๆ a จะสามารถแสดงสมการได้ว่า $L^*_{n+1}a = k_{n+1}a$ และ $L_{n+1}f_n = f_n(x_{n+1})$ ดังนั้น

$$\nabla \hat{g}_{reg}(f_n) = k_{n+1}[f_n(x_{n+1}) - z_{n+1}] + \rho f_n \tag{2.14}$$

เพราะฉะนั้นสมการที่ (2.12) สามารถเขียนได้เป็น

$$f_{n+1} = (1-\eta_n\rho)f_n - \eta_n k_{n+1}[f_n(x_{n+1}) - z_{n+1}] \tag{2.15}$$

สมมติว่าการประมาณค่าฟังก์ชัน ณ เวลาปัจจุบันมีเคอร์เนลจำนวน p เทอม ฟังก์ชันการประมาณค่าเหล่านี้สามารถเขียนใหม่ได้เป็น

$$\begin{aligned}f_{n+1}(x) &= (1-\eta_n\rho)\sum_{i=1}^{p}\alpha_n^i k_i(x) - \eta_n e_{n+1}k_{n+1}(x) \\ &= \sum_{i=1}^{p+1}\alpha_{n+1}^i k_i(x)\end{aligned} \tag{2.16}$$

เมื่อ α_{n+1}^{i} ถูกคำนวณมาก่อนแล้วดังสมการต่อไปนี้

$$\alpha_{n+1}^{i} = (1-\eta_n \rho)\alpha_n^{i} \qquad \text{โดยที่ } i \le p \tag{2.17}$$

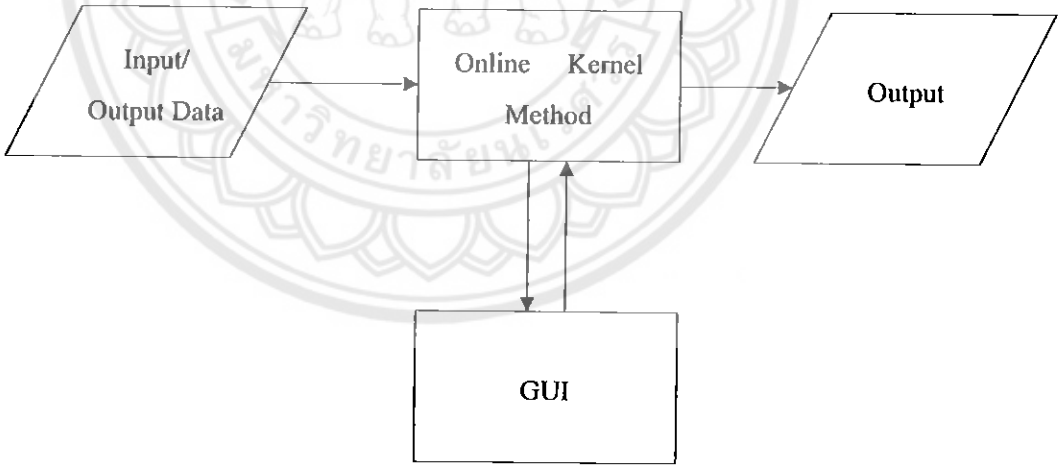
และ

$$\alpha_{n+1}^{i} = -\eta_n e_{n+1} \qquad \text{โดยที่ } i = p+1 \tag{2.18}$$

จากสมการที่ผ่านมาเราสามารถสรุปได้ว่าเมื่อค่าน้ำหนัก (α) ค่าใหม่ถูกเพิ่มให้กับแบบจำลอง (f_{n+1}) โดยมีการคูณเข้ากับฟังก์ชันเคอร์เนล ค่าน้ำหนักค่าเก่าจะถูกปรับปรุงโดยการคูณด้วยเทอม $(1-\eta_n \rho)$ ซึ่งหมายถึงการลดลงของค่าน้ำหนักค่านั้น ๆ

2.3 ส่วนต่อประสานกราฟิกกับผู้ใช้ [1], [2]

ส่วนต่อประสานกราฟิกผู้กับใช้ (Graphical User Interface, GUI) เป็นวิธีการใช้งานคอมพิวเตอร์ผ่านทางสัญลักษณ์หรือภาพนอกเหนือจากทางตัวอักษร ส่วนต่อประสานกราฟิกกับผู้ใช้มีส่วนประกอบต่าง ๆ เช่น ไอคอน หน้าต่างการใช้งาน เมนู ปุ่มเลือก และการใช้เมาส์ หรือแม้แต่ในระบบทัชสกรีน ลักษณะการทำงานดังแสดงรูปที่ 2.1



รูปที่ 2.1 ลักษณะการทำงานของส่วนต่อประสานกราฟิกกับผู้ใช้

โปรแกรมแมทแลปจะสร้างส่วนต่อประสานกราฟิกกับผู้ใช้อยู่บนหน้าต่างรูปภาพ (Figure window) ซึ่งภายในหน้าต่างนี้จะมีส่วนประกอบต่าง ๆ อยู่ไม่ว่าจะเป็น Axes Uicontrol การควบคุมและเมนูแบบต่าง ๆ หรือวัตถุอื่น ๆ ที่ผู้ใช้ต้องการที่จะสร้างลงในหน้าต่างรูปภาพตามที่ผู้ใช้ต้องการ

GUIDE จะมีหน้าที่ในการวางส่วนประกอบที่ต้องการให้มีลงในส่วนต่อประสานกราฟิกกับผู้ใช้จากนั้น GUIDE จะสร้าง M-file ของหน้าต่างที่ได้ออกแบบไว้รวมทั้งคำสั่งต่าง ๆ ซึ่งเป็นโครงร่างโปรแกรมที่ใช้ควบคุมการทำงานของส่วนต่อประสานกราฟิกกับผู้ใช้และจะถูกสร้างขึ้นโดยอัตโนมัติพร้อมกับ Fig-file เมื่อใช้ GUIDE ในการสร้างส่วนต่อประสานกราฟิกกับผู้ใช้ โปรแกรมหรือ Code ทุกส่วนรวมถึง Function จะรวมอยู่ใน Application M-file โดย Callback ทั้งหมดจะถูกเขียนเป็น Function และสามารถปรับค่าเริ่มต้นหรือทำการ Initialize ให้กับส่วนต่อประสานกราฟิกกับผู้ใช้ได้สะดวกและรวดเร็วมากขึ้น โดยที่ Application M-file จะให้แนวทางในการเขียนฟังก์ชันและกระบวนการต่าง ๆ

การสร้างส่วนต่อประสานกราฟิกกับผู้ใช้จะประกอบด้วยขั้นตอนสองขั้นตอน

1) กำหนดและวางส่วนประกอบต่าง ๆ ลงบนส่วนต่อประสานกราฟิกกับผู้ใช้
2) เขียนโปรแกรมเพื่อกำหนดการทำงานของแต่ละส่วนประกอบในส่วนต่อประสานกราฟิกกับผู้ใช้

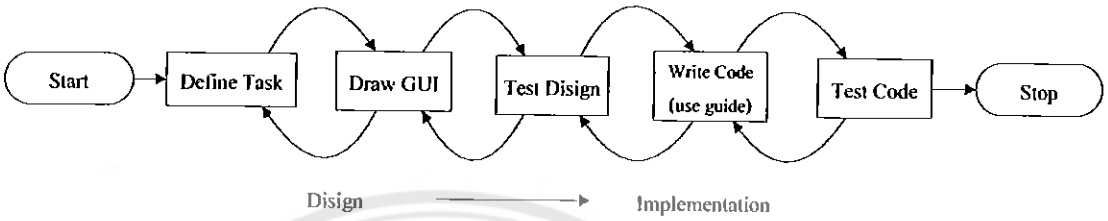
2.3.1 ส่วนประกอบของส่วนต่อประสานกราฟิกกับผู้ใช้ในแมทแลป

การใช้ส่วนต่อประสานกราฟิกกับผู้ใช้เขียนเป็น M-file ขึ้นมาจะทำให้การทำงานง่ายขึ้น ทำให้ผู้ใช้กำหนดตำแหน่งของวัตถุต่าง ๆ ได้โดยง่ายเมื่อทำการ Save File แล้วส่วนต่อประสานกราฟิกกับผู้ใช้จะสร้างไฟล์ขึ้นมาสองไฟล์เพื่อเก็บและนำมาใช้ต่อไปซึ่งจะประกอบด้วย

1) FIG-file ซึ่งจะบรรจุรายละเอียดของวัตถุต่าง ๆ ที่เป็นองค์ประกอบอยู่ในหน้าต่างรูปภาพที่เป็นส่วนต่อประสานกราฟิกกับผู้ใช้
2) M-file ที่จะบรรจุฟังก์ชันที่กำหนดการทำงานของส่วนต่อประสานกราฟิกกับผู้ใช้ รวมถึง Callback ทั้งหมดซึ่งจะถูกบรรจุเป็นฟังก์ชันอยู่ใน M-file และจะเรียกส่วนนี้ว่า M-file ที่ควบคุมการทำงานของส่วนต่อประสานกราฟิกกับผู้ใช้นี้ว่า Application M-file ในขณะที่รูปแบบของส่วนประกอบที่บรรจุอยู่ในส่วนต่อประสานกราฟิกกับผู้ใช้เช่น สี ขนาด ตำแหน่ง หรือค่าอื่น ๆ จะถูกบรรจุอยู่ใน FIG-file

2.4 กระบวนการออกแบบส่วนต่อประสานกราฟิกกับผู้ใช้

การออกแบบส่วนต่อประสานกราฟิกกับผู้ใช้มีความสำคัญต่อเป็นอย่างมากเพื่อความสะดวกของผู้ใช้เองและง่ายต่อการเข้าใจและการนำไปใช้งานในด้านต่าง ๆ ขั้นตอนการออกแบบสามารถแสดงดังในรูปต่อไปนี้ดังแสดงรูปที่ 2.2



รูปที่ 2.2 ขั้นตอนการออกแบบส่วนต่อประสานกราฟิกกับผู้ใช้

2.4.1 การออกแบบให้ง่ายต่อการใช้งาน

ในการออกแบบส่วนต่อประสานกราฟิกกับผู้ใช้นั้นจะต้องคำถึงความง่ายในการใช้งานเพราะการออกแบบที่ซับซ้อนโดยไม่จำเป็นนั้นจะทำให้ผู้ใช้มีความรู้สึกว่าส่วนต่อประสานกราฟิกกับผู้ใช้นี้ยุ่งยากเกินกว่าที่จะสามารถนำมาใช้ได้อย่างมีประสิทธิภาพ ดังนั้นการออกแบบควรจะมีเฉพาะปุ่มที่จำเป็นที่จะต้องควบคุมไม่ควรใช้หน้าต่างหลาย ๆ หน้าต่างในการแสดงผลและรับข้อมูลเพราะจะทำให้สับสนได้ง่าย

2.4.2 การออกแบบให้มีรูปแบบที่ชัดเจน

การออกแบบให้มีรูปแบบที่จัดเจนหมายความว่าในกรณีที่สร้างส่วนต่อประสานกราฟิกกับผู้ใช้ขึ้นมาใช้เป็นลำดับขั้นคือหลังจากการกรอกข้อมูลหน้าต่างแรกเสร็จก็จะปรากฏหน้าต่างที่สองขึ้นมาและอาจมีหน้าต่างอื่น ๆ ตามขึ้นมาอีกการออกแบบแต่ละหน้าต่างควรจะให้มีการกำหนดลักษณะของแถบเมนูที่ใกล้เคียงกันและควรจะมีสีพื้นที่เหมือนกันคำอธิบายต่าง ๆ ก็ควรจะวางไว้ในตำแหน่งที่ใกล้เคียงกันในทุกหน้าต่าง

บทที่ 3
วิธีการประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนล
โดยใช้ส่วนต่อประสานกราฟิกกับผู้ใช้

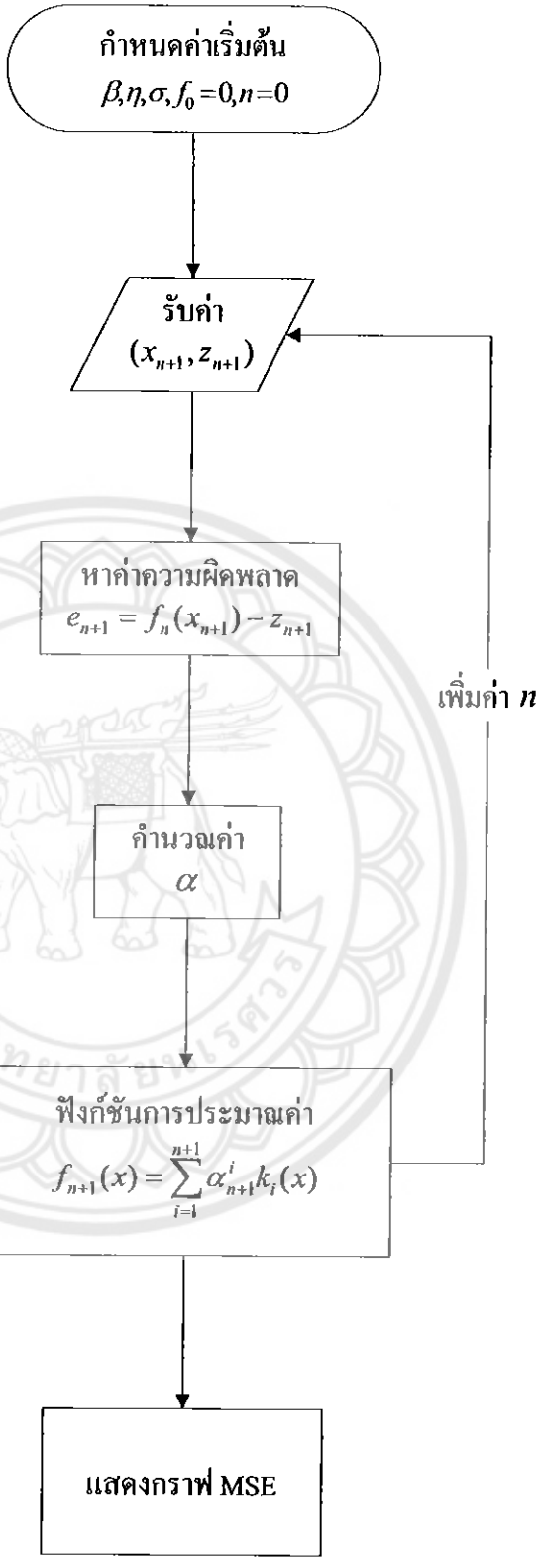
โครงงานนี้ได้มุ่งเน้นศึกษาการออกแบบวิธีการประมาณค่าฟังก์ชันของระบบที่ไม่รู้ค่าด้วยวิธีเคอร์เนล โดยอาศัยทฤษฎีที่ได้กล่าวมาแล้วในบทที่ 2 จากนั้นฟังก์ชันการประมาณค่าหรือแบบจำลองที่ได้ จะถูกหาค่าความผิดพลาดโดยใช้ค่าความผิดพลาดเฉลี่ยกำลังสอง (MSE-mean square error) และใช้ส่วนต่อประสานกราฟิกกับผู้ใช้ในการกำหนดและสามารถเปลี่ยนแปลงค่าพารามิเตอร์ β, η, ρ ได้โดยตรงจากหน้าต่างส่วนต่อประสานกราฟิกกับผู้ใช้ เพื่อดูความแตกต่างของค่าที่เปลี่ยนแปลงว่ามีผลต่อการแสดงผลของค่าความผิดพลาดอย่างไร

3.1 หลักการออกแบบการประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนล

การออกแบบระบบด้วยวิธีการหาค่าความผิดพลาดของโมเดลโดยการสุ่มตัวอย่าง เป็นการระบุค่าพารามิเตอร์ของระบบจากข้อมูลหลาย ๆ ข้อมูลมาทำการหาค่าเฉลี่ยความผิดพลาด ซึ่งจะมีหลักการในการเลือกแบบจำลองที่เหมาะสมของระบบ โดยพิจารณาจากค่าที่ผิดพลาดน้อยที่สุดซึ่งการประมาณค่าฟังก์ชันมีขั้นตอนการทำดังนี้

1) กำหนดค่าเริ่มต้นค่า β, η, ρ และฟังก์ชัน $f_0 = 0$
2) กำหนดค่ารูปแบบข้อมูลที่ใช้สอน
3) กำหนดค่า n เริ่มที่ $n = 0$
4) รับค่าข้อมูล (x_{n+1}, z_{n+1}) จากชุดข้อมูลที่ใช้สอน
5) หาค่าความผิดพลาดโดยใช้สมการ $e_{n+1} = f_n(x_{n+1}) - z_{n+1}$
6) หาค่า α จากสมการที่ (2.17) และ (2.18)
7) หาค่าฟังก์ชันการประมาณค่าที่ได้จากสมการ $f_{n+1}(x) = \sum_{i=1}^{n+1} \alpha_{n+1}^i k_i(x)$ โดยใช้ฟังก์ชันเคอร์เนล $k(x,x') = \exp(-\beta\|x - x'\|^2)$ และ $k(x,x') = \dfrac{\sin \beta\pi(x-x')}{\pi(x-x')}$
8) ทำซ้ำตามขั้นตอนที่ 3-6

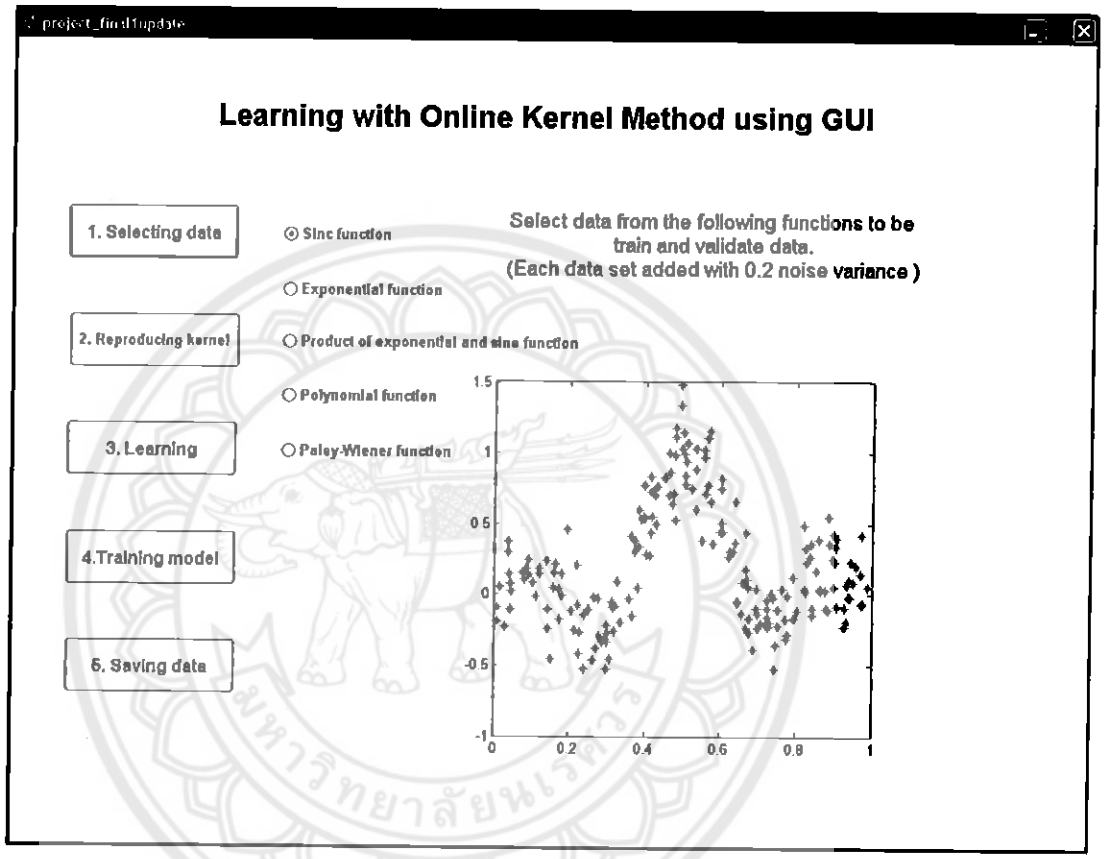
ดังนั้นเมื่อเสร็จสิ้นกระบวนการจะได้ฟังก์ชันการประมาณค่า f เพื่อที่จะนำไปวิเคราะห์หาข้อผิดพลาดจากข้อมูลที่ใช้ทดสอบต่อไป ลำดับการประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนลแสดงได้ดังรูปที่ 3.1



รูปที่ 3.1 ขั้นตอนการประมาณค่าฟังก์ชัน

3.2 การออกแบบและการใช้โปรแกรม

เมื่อเริ่มการทำงานของโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้แล้ว จะมีปุ่มการทำงานหลัก ๆ อยู่ 5 ปุ่ม ดังแสดงรูปที่ 3.2

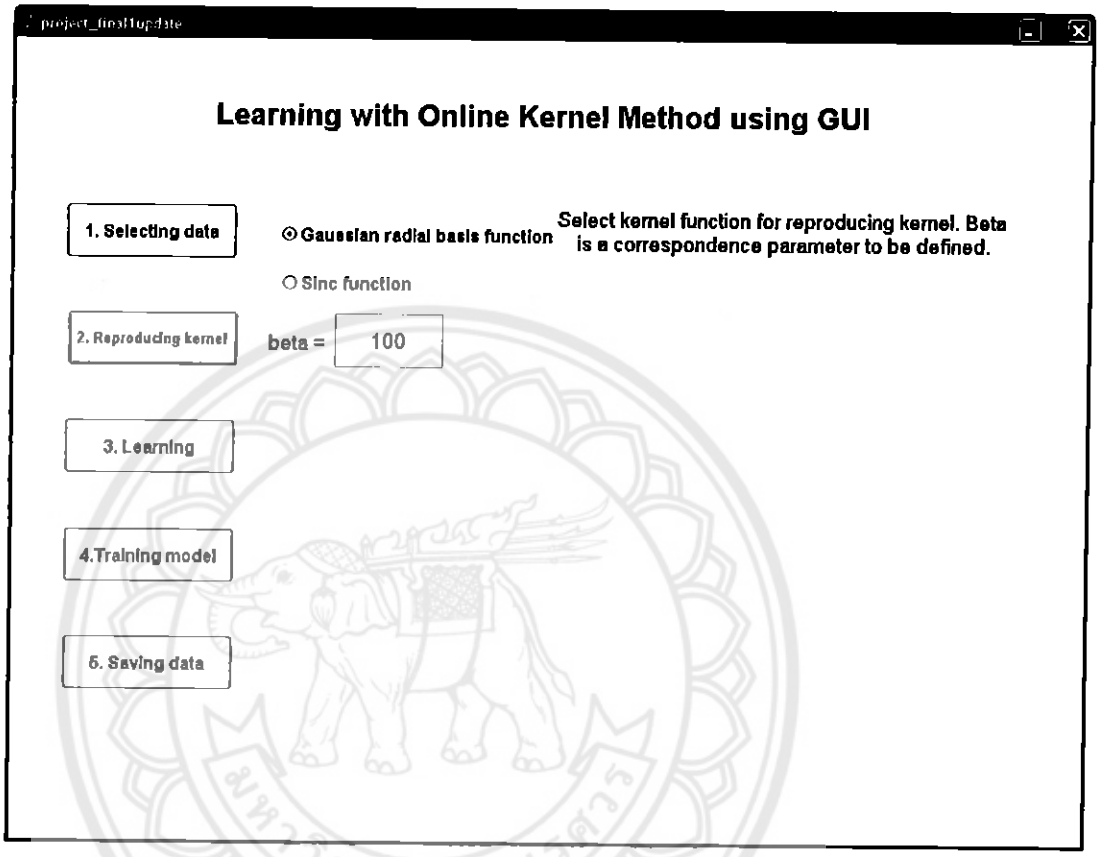


รูปที่ 3.2 หน้าต่างการทำงานของโปรแกรมเมื่อเลือก Selecting data

จากรูปที่ 3.2 ลำดับที่ 1 ทำการเลือกปุ่ม

ซึ่งจะเป็นการเลือกรูปแบบข้อมูลที่ใช้สอนโดยมีทั้งหมด 5 ข้อมูล โดยในแต่ละข้อมูลจะมีลักษณะของกราฟที่แตกต่างกันของแต่ละข้อมูล

เมื่อเลือกรูปแบบข้อมูลที่ใช้สอนแล้ว ต่อไปเป็นการเลือกฟังก์ชันเคอร์เนลสำหรับการประมาณค่าฟังก์ชันดังแสดงรูปที่ 3.3

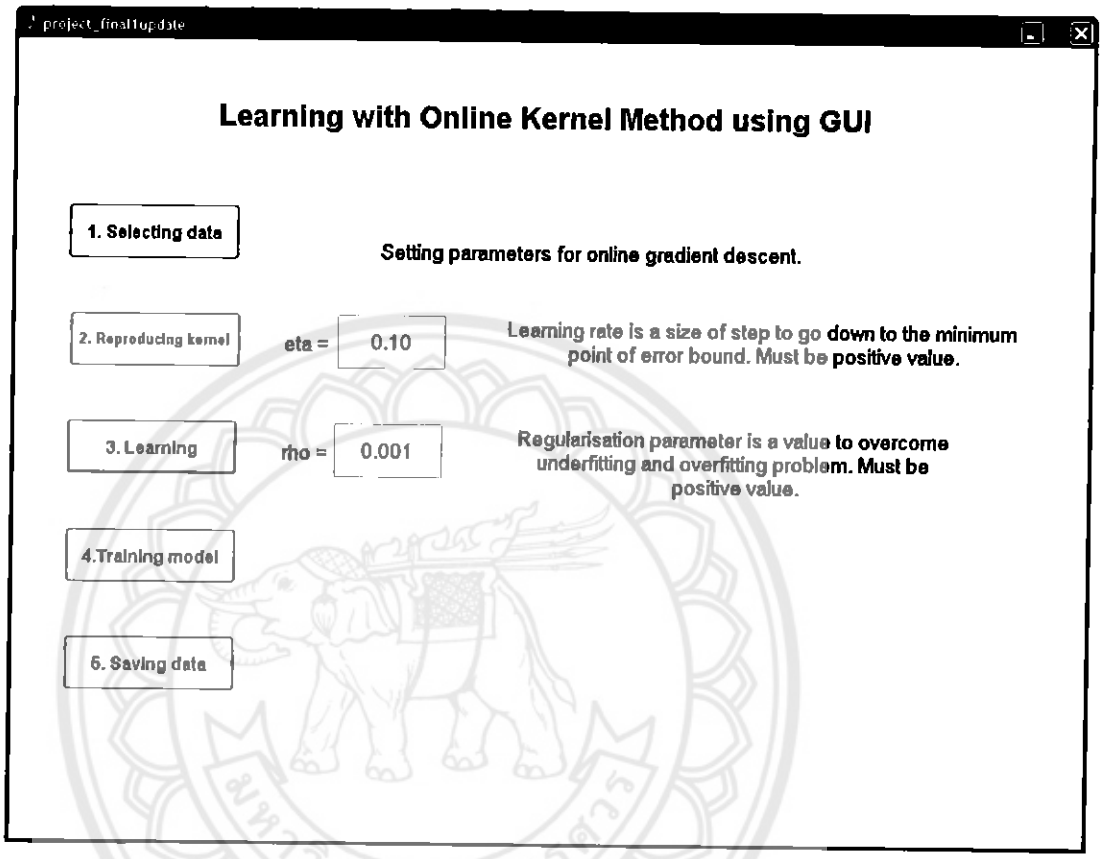


รูปที่ 3.3 หน้าต่างการทำงานของโปรแกรมเมื่อเลือก Reproducing kernel

ลำดับที่ 2 เมื่อทำการเลือกปุ่ม

จะแสดงรูปที่ 3.3 ซึ่งเป็นการเลือกฟังก์ชันเคอร์เนลมีฟังก์ชันเคอร์เนลที่ใช้สองฟังก์ชันคือ ฟังก์ชันเกาเสียนเรเดียลเบสิสและซิงค์ฟังก์ชันและการป้อนค่าเบต้าซึ่งเป็นค่าพารามิเตอร์ในที่นี้ให้เท่ากับ 100

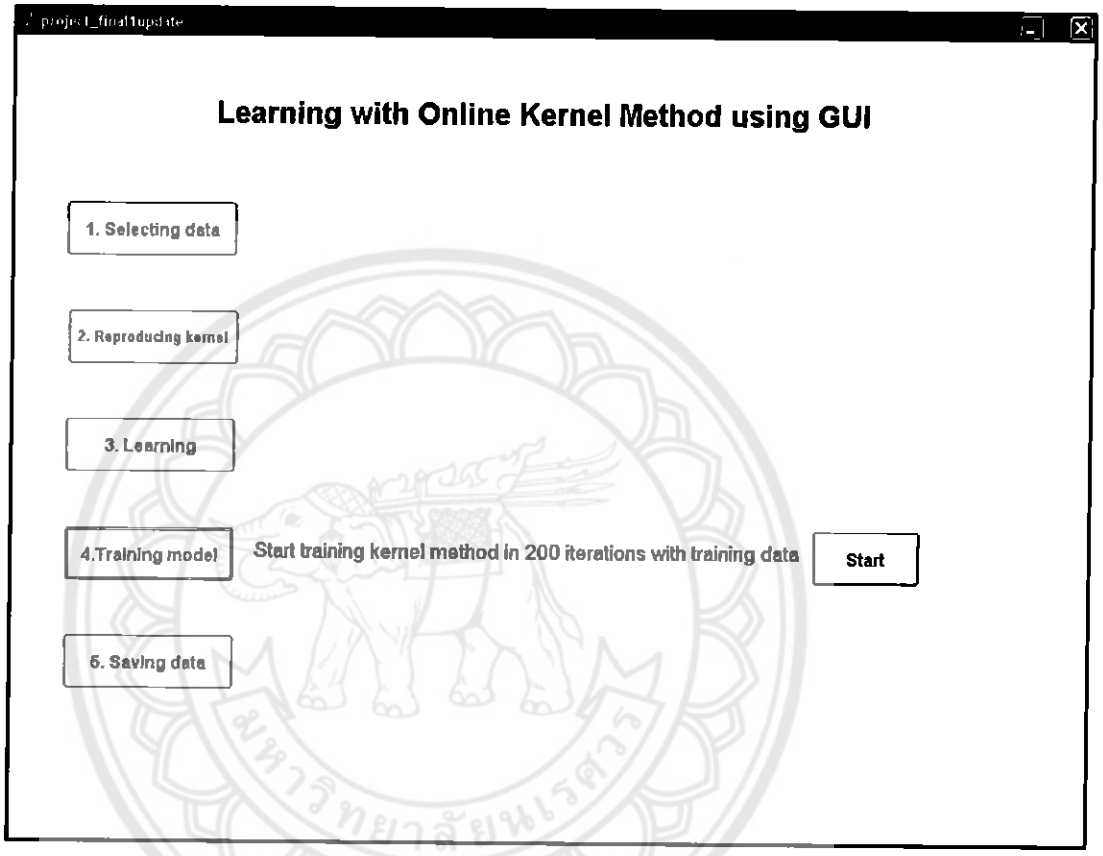
ค่าพารามิเตอร์ที่ใช้ในการประมาณค่าฟังก์ชันเคอร์เนลนั้น ในแต่ละฟังก์ชันเคอร์เนลจะใช้ค่าพารามิเตอร์ที่แตกต่างกัน การป้อนค่าพารามิเตอร์ต่าง ๆ แสดงดังรูปที่ 3.4



รูปที่ 3.4 หน้าต่างการทำงานของโปรแกรมเมื่อเลือก Learning

ลำดับที่ 3 เลือกปุ่ม 3. Learning จะแสดงดังรูปที่ 3.4 ซึ่งเป็นการป้อนค่าพารามิเตอร์คือค่าอัตราการเรียนรู้ซึ่งควรจะมีค่าบวกและค่าเรกกูลาไรเซชัน ในที่นี้ให้เท่ากับ 0.10, 0.001 ตามลำดับ

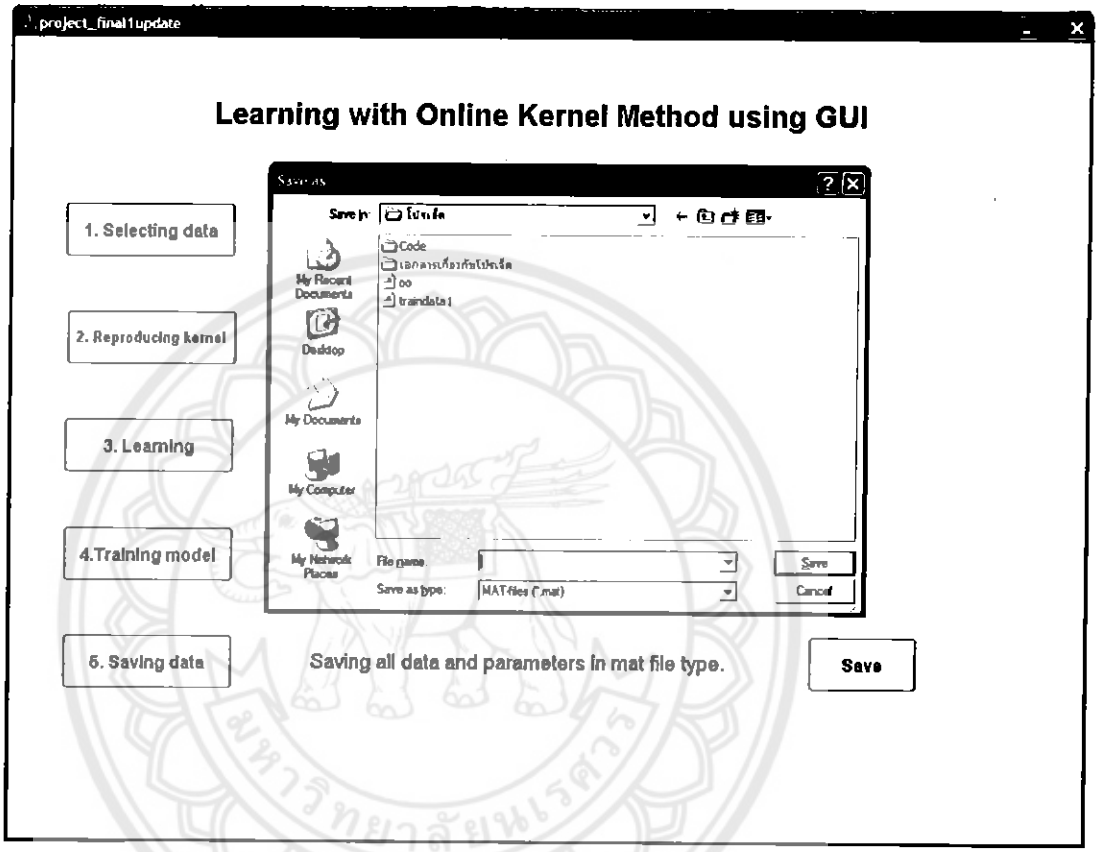
เมื่อเลือกข้อมูลที่ใช้สอน เลือกฟังก์ชันเคอร์เนลและป้อนค่าพารามิเตอร์ทั้งหมดแล้วก็จะทำการประมาณค่าฟังก์ชันเคอร์เนล วิธีการประมาณค่าฟังก์ชันเคอร์เนลจากส่วนต่อประสานกราฟิกกับผู้ใช้มีวิธีการดังแสดงรูปที่ 3.5



รูปที่ 3.5 หน้าต่างการทำงานของโปรแกรมเมื่อเลือก Training model

ลำดับที่ 4 เลือกปุ่ม 4.Training model จะแสดงดังรูปที่ 3.5 ซึ่งเป็นการเริ่มกระบวนการประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนลจำนวน 200 รอบ เมื่อทำกดปุ่ม Start โปรแกรมก็จะทำการคำนวณโดยใช้ค่าพารามิเตอร์ทั้ง 3 ตัวตามที่ได้ป้อนไว้ในข้างต้น

เมื่อทำการประมาณค่าฟังก์ชันเคอร์เนลแล้ว การบันทึกข้อมูลของการประมาณค่าในแต่ละข้อมูลที่ใช้สอนและแต่ละฟังก์ชันเคอร์เนล สามารถนำไปประยุกต์ใช้ต่อไปได้ มีลักษณะการทำงานดังแสดงรูปที่ 3.6



รูปที่ 3.6 หน้าต่างการทำงานของโปรแกรมเมื่อเลือก Save data

ลำดับที่ 5 หลังจากทำการประมาณค่าฟังก์ชัน แล้วเมื่อเลือกปุ่ม 5. Saving data หลังจากนั้นเลือกปุ่ม Save จะแสดงรูปที่ 3.6 ขึ้นมาเพื่อทำการบันทึกข้อมูลทุกอย่างไว้ในรูป mat ไฟล์ เพื่อสามารถเรียกกลับมาดูข้อมูลได้ในภายหลัง

ผลการทดลองจากการป้อนค่าพารามิเตอร์ทั้ง 3 ตัว และการแสดงลักษณะค่าความผิดพลาดที่ได้ของแต่ละรูปแบบข้อมูลในแต่ละฟังก์ชันเคอร์เนลจะแสดงในบทที่ 4

บทที่ 4

ผลการทดลองและการวิเคราะห์ผล

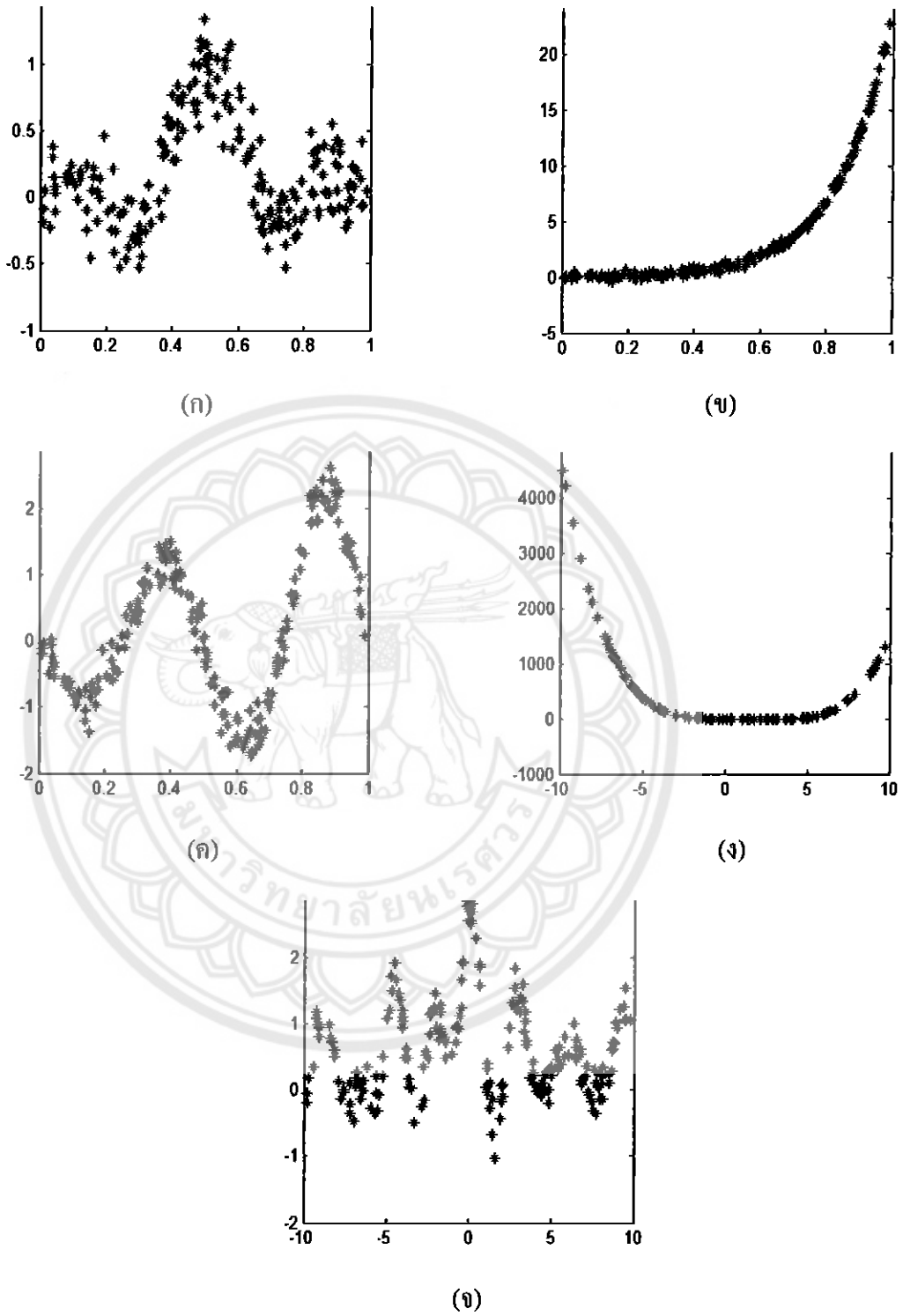
ในบทนี้จะกล่าวถึงผลการทดลองของวิธีการประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนล ซึ่งจะทำการปรับค่าพารามิเตอร์ต่าง ๆ และดูผลที่เกิดขึ้น โดยแต่ละค่าพารามิเตอร์ที่แตกต่างกันจะให้ผลของค่าความผิดพลาด (ค่าความผิดพลาดเฉลี่ยกำลังสองหรือ MSE) ที่แตกต่างกันออกไป แต่สิ่งที่ต้องการคือการลดลงของค่าความผิดพลาดต้องค่อย ๆ ลดลงและค่าความผิดพลาดที่ได้ต้องน้อยที่สุด ซึ่งเป็นสิ่งที่ระบบต้องการ

4.1 ผลการทดลองการประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนล

การประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนลนั้นสามารถทำได้โดยใช้สมการที่ (2.16) และจะใช้ฟังก์ชันเกาเสียนเรเดียลเบสิสและซิงค์ฟังก์ชันเป็นฟังก์ชันเคอร์เนลในการทดลองหาการประมาณค่าฟังก์ชัน จะมีขั้นตอนตามที่ได้กล่าวมาแล้วในบทที่ 3 หัวข้อที่ 3.1 ในการทดลองนี้ได้สร้างข้อมูลที่ใช้สอนจำนวน 5 รูปแบบข้อมูลคือแบบ

1) Sinc function
2) Exponential function
3) Product of exponential and sine function
4) Polynomial function
5) Paley-Wiener function

ซึ่งจะมีรูปแบบกราฟที่แตกต่างกันออกไปดังแสดงรูปที่ 4.1



รูปที่ 4.1 ข้อมูลที่ใช้สอน ก) ข้อมูลแบบ Sinc function ข) ข้อมูลแบบ Exponential ค) ข้อมูลแบบ Product of exponential and sine function ง) ข้อมูลแบบ Polynomial function จ) ข้อมูลแบบ Paley-Wiener function

4.1.1 การประมาณค่าโดยใช้เคอร์เนลฟังก์ชันเป็นฟังก์ชันเกาเสียนเรเดียลเบสิส

การประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนลโดยข้อมูลที่ใช้สอนมีทั้งหมด 5 รูปแบบ และมีการเลือกใช้เคอร์เนลฟังก์ชันเป็นฟังก์ชันเกาเสียนเรเดียลเบสิส จากนั้นจะทดลองปรับค่าพารามิเตอร์ต่าง ๆ ให้ได้ค่าความผิดพลาดที่ดีที่สุด โดยจะมีการปรับค่าพารามิเตอร์ 3 ตัว คือค่าเบต้า ค่าอัตราการเรียนรู้ และค่าแรกกูลาไรเซชันซึ่งทำการเปรียบเทียบแสดงได้ดังตารางที่ 4.1 ดังต่อไปนี้

ตารางที่ 4.1 ค่าพารามิเตอร์สำหรับเคอร์เนลฟังก์ชันแบบเกาเสียนเรเดียลเบสิส

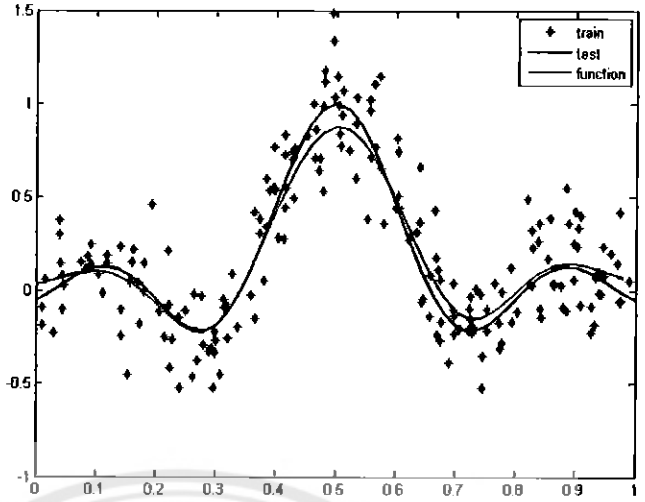
ข้อมูลที่ใช้สอน	β	η	ρ
Sinc function	50	0.1	0.001
	100	0.1	0.001
	150	0.1	0.001
Exponential function	150	0.5	0.001
	150	1	0.001
	150	1.5	0.001
Product of exponential and sine function	200	0.5	0.001
	200	1	0.001
	200	1.5	0.001
Polynomial function	0.5	0.5	0.001
	0.5	1	0.001
	0.5	1.5	0.001
Paley-Wiener function	0.5	0.5	0.001
	1	0.5	0.001
	1.5	0.5	0.001

จากตารางที่ 4.1 การปรับค่าพารามิเตอร์ที่ใช้เคอร์เนลฟังก์ชันเป็นฟังก์ชันเกาเสียนเรเดียลเบสิสโดยในแต่ละข้อมูลที่ใช้สอนนั้นมีลักษณะการปรับค่าที่แตกต่างกันออกไป ในข้อมูลที่ใช้สอนแต่ละข้อมูลมีค่าพารามิเตอร์บางตัวคงที่ บางตัวเปลี่ยนค่า ซึ่งในแต่ละข้อมูลที่ใช้สอนก็จะมีตัวคงที่และตัวที่ปรับค่าแตกต่างกัน ซึ่งจากการทดลองปรับค่าพารามิเตอร์ของแต่ละข้อมูลนั้น ค่าพารามิเตอร์บางตัวเมื่อทำการเปลี่ยนค่าพารามิเตอร์แล้ว ผลการประมาณค่าที่แสดงออกมาในรูปของค่าเฉลี่ยความผิดพลาดกำลังสองนั้นมีการเปลี่ยนแปลงที่น้อยมากซึ่งจะไม่ค่อยเห็นถึงความแตกต่างที่เกิดขึ้น ดังนั้นจึงทำให้แต่ละข้อมูลที่ใช้สอนมีการปรับค่าพารามิเตอร์ที่แตกต่างกันออกไป จะเห็นได้ว่าจะมีค่าพารามิเตอร์ที่คงที่อยู่สองตัวและตัวที่เปลี่ยนค่าอยู่หนึ่งตัว จากการทดลองตัวพารามิเตอร์ที่เปลี่ยนค่านั้นมีผลต่อค่าความผิดพลาดเฉลี่ยกำลังสองและเห็นถึงความแตกต่างที่เห็นได้ชัดซึ่งเป็นสิ่งที่ต้องการ จากการทดลองนี้มีลักษณะการปรับค่าพารามิเตอร์ต่าง ๆ ดังนี้

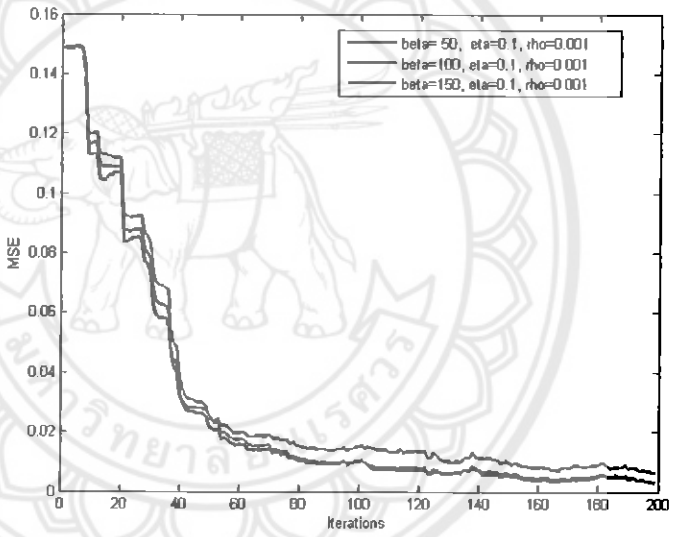
1) ข้อมูลที่ใช้สอนแบบ Sinc function นั้นการปรับค่าพารามิเตอร์จะปรับค่าเบต้า ส่วนค่าอัตราการเรียนรู้และค่าเรกกูลาไรเซชันนั้นคงที่
2) ข้อมูลที่ใช้สอนแบบ Exponential function นั้นการปรับค่าพารามิเตอร์จะปรับค่าอัตราการเรียนรู้ ส่วนค่าเบต้าและค่าเรกกูลาไรเซชันนั้นคงที่
3) ข้อมูลที่ใช้สอนแบบ Product of exponential and sine function นั้นการปรับค่าพารามิเตอร์จะปรับค่าอัตราการเรียนรู้ ส่วนค่าเบต้าและค่าเรกกูลาไรเซชันนั้นคงที่
4) ข้อมูลที่ใช้สอนแบบ Polynomial function นั้นการปรับค่าพารามิเตอร์จะปรับค่าอัตราการเรียนรู้ ส่วนค่าเบต้าและค่าเรกกูลาไรเซชันนั้นคงที่
5) ข้อมูลที่ใช้สอนแบบ Paley-Wiener function นั้นการปรับค่าพารามิเตอร์จะปรับค่าเบต้า ส่วนค่าอัตราการเรียนรู้และค่าเรกกูลาไรเซชันนั้นคงที่

การประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนลโดยใช้ฟังก์ชันเกาเสียนเรเดียลเบสิสโดยใช้ค่าพารามิเตอร์ต่าง ๆ ตามตารางที่ 4.1 ค่าความผิดพลาดของแต่ละรูปแบบข้อมูลที่ใช้สอนแสดงได้ดังนี้

1) ผลของการประมาณค่าฟังก์ชันเมื่อใช้เคอร์เนลฟังก์ชันเป็นเกาเสียนเรเดียลเบสิสโดยใช้ข้อมูลที่ใช้สอนแบบ Sinc function จะได้กราฟเปรียบเทียบระหว่างฟังก์ชันจริงของข้อมูลและแบบจำลองที่ได้จากการทดลอง และกราฟค่าความผิดพลาดดังแสดงรูปที่ 4.2



(ก)



(ข)

รูปที่ 4.2 การประมาณค่าด้วยวิธีเคอร์เนลเป็นฟังก์ชันเกาเสียนเรเดียลเบสิสสำหรับข้อมูลที่ใช้สอนแบบ Sinc function ก) ข้อมูลที่ใช้สอนและได้จากการทดลอง ข) ผลของความผิดพลาด

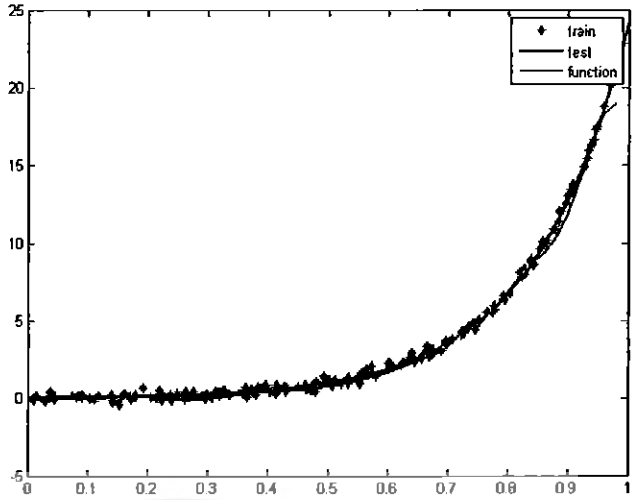
จากรูปที่ 4.2 (ก) เป็นการแสดงการเปรียบเทียบฟังก์ชันจริงของข้อมูลที่ใช้สอนและแบบจำลองที่ได้จากการทดลองปรับค่าพารามิเตอร์ต่าง ๆ เมื่อใช้เคอร์เนลฟังก์ชันเป็นฟังก์ชันเกาเสียนเรเดียลเบสิสและใช้รูปแบบข้อมูลที่ใช้สอนเป็น Sinc function โดยจุดสีแดงหมายถึงข้อมูลที่ใช้สอน เส้นสีน้ำเงินหมายถึงข้อมูลที่ใช้ทดสอบ เส้นสีเขียวหมายถึงฟังก์ชันการประมาณค่าที่โมเดลสร้างขึ้น ซึ่งจะเห็นได้ว่าฟังก์ชันการประมาณค่าโมเดลที่สร้างขึ้นมีค่าใกล้เคียงกับฟังก์ชันจริงมาก

จากรูปที่ 4.2 (ข) เป็นผลของค่าความผิดพลาดที่ได้จากการปรับค่าพารามิเตอร์ทั้ง 3 ตัว คือ ค่าเบต้า ค่าอัตราการเรียนรู้และค่าแรกกูลาไรเซชัน เมื่อทดลองปรับค่าพารามิเตอร์ทั้ง 3 ตัวแล้ว จะเห็นได้ว่าเส้นสีแดงที่ได้จากการปรับค่าเบต้า ค่าอัตราการเรียนรู้และค่าแรกกูลาไรเซชันเท่ากับ 100, 0.1 และ 0.001 ตามลำดับ นั้นมีค่าความผิดพลาดที่ดีที่สุด โดยในช่วงแรกจะลดลงอย่างฉับพลันตั้งแต่รอบที่ 0 – 40 แล้วหลังจากนั้นจะค่อย ๆ ลดลงจนค่าความผิดพลาดนั้นมีลักษณะคงที่

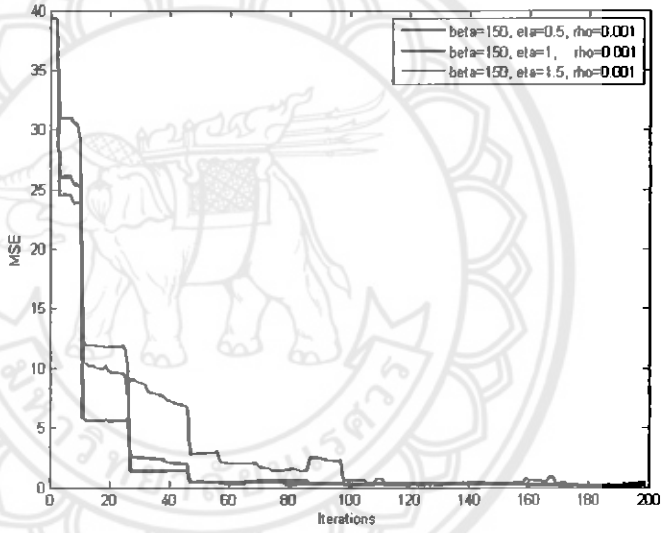
2) ผลของการประมาณค่าฟังก์ชันเมื่อใช้เคอร์เนลฟังก์ชันเป็นเกาเสียนเรเดียลเบสิสโดยใช้ข้อมูลที่ใช้สอนแบบ Exponential function จะได้กราฟเปรียบเทียบระหว่างฟังก์ชันจริงของข้อมูลและแบบจำลองที่ได้จากการทดลอง และกราฟค่าความผิดพลาดดังแสดงรูปที่ 4.3

จากรูปที่ 4.3 (ก) เป็นการแสดงการเปรียบเทียบฟังก์ชันจริงของข้อมูลที่ใช้สอนและแบบจำลองที่ได้จากการทดลองปรับค่าพารามิเตอร์ต่าง ๆ เมื่อใช้เคอร์เนลฟังก์ชันเป็นฟังก์ชันเกาเสียนเรเดียลเบสิสและใช้รูปแบบข้อมูลที่ใช้สอนเป็น Exponential function โดยจุดสีแดงหมายถึงข้อมูลที่ใช้สอน เส้นสีน้ำเงินหมายถึงข้อมูลที่ใช้ทดสอบ เส้นสีเขียวหมายถึงฟังก์ชันการประมาณค่าที่โมเดลสร้างขึ้น ซึ่งจะเห็นได้ว่าฟังก์ชันการประมาณค่าโมเดลที่สร้างขึ้นมีค่าใกล้เคียงกับฟังก์ชันจริงมาก

จากรูปที่ 4.3 (ข) เป็นผลของค่าความผิดพลาดที่ได้จากการปรับค่าพารามิเตอร์ทั้ง 3 ตัว คือ ค่าเบต้า ค่าอัตราการเรียนรู้และค่าแรกกูลาไรเซชัน เมื่อทดลองปรับค่าพารามิเตอร์ทั้ง 3 ตัวแล้ว จะเห็นได้ว่าเส้นสีแดงที่ได้จากการปรับค่าเบต้า ค่าอัตราการเรียนรู้และค่าแรกกูลาไรเซชันเท่ากับ 150, 1 และ 0.001 ตามลำดับ นั้นมีค่าความผิดพลาดที่ดีที่สุด โดยในช่วงแรกจะลดลงอย่างฉับพลันตั้งแต่รอบที่ 0 – 20 แล้วหลังจากนั้นจะค่อย ๆ ลดลงจนค่าความผิดพลาดนั้นมีลักษณะคงที่



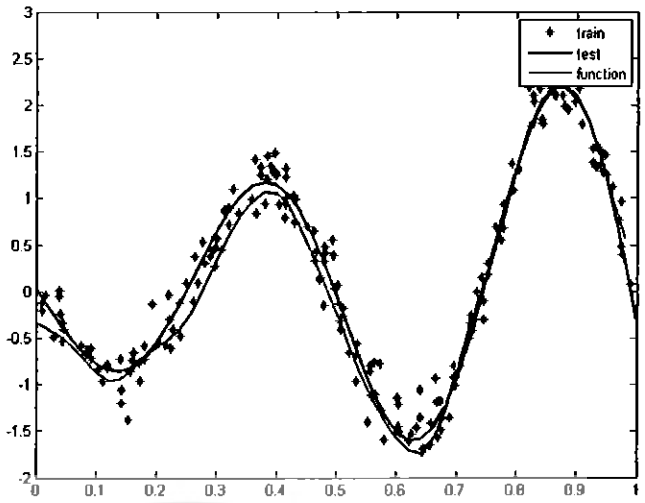
(ก)



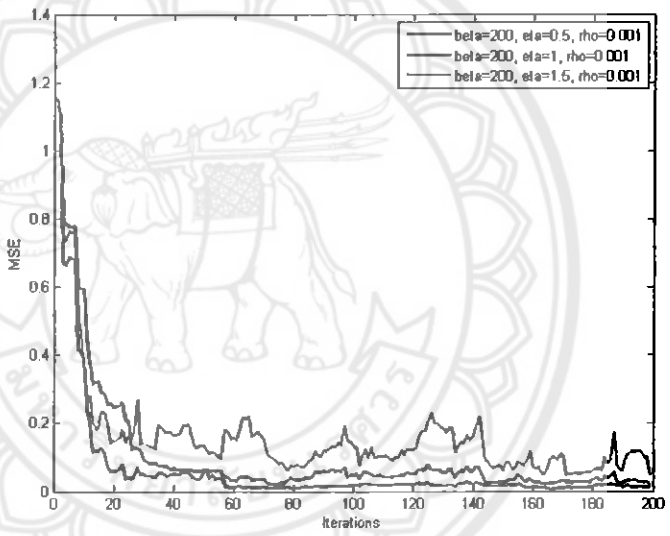
(ข)

รูปที่ 4.3 การประมาณค่าด้วยวิธีเคอร์เนลเป็นฟังก์ชันเกาเสียนเรเดียลเบสิสสำหรับข้อมูลที่ใช้สอนแบบ Exponential function ก) ข้อมูลที่ใช้สอนและได้จากการทดลอง ข) ผลของความผิดพลาด

3) ผลของการประมาณค่าฟังก์ชันเมื่อใช้เคอร์เนลฟังก์ชันเป็นเกาเสียนเรเดียลเบสิสโดยใช้ข้อมูลที่ใช้สอนแบบ Product of exponential and sine function จะได้กราฟเปรียบเทียบระหว่างฟังก์ชันจริงของข้อมูลและแบบจำลองที่ได้จากการทดลอง และกราฟค่าความผิดพลาดดังแสดงรูปที่ 4.4



(ก)



(ข)

รูปที่ 4.4 การประมาณค่าด้วยวิธีเคอร์เนลเป็นฟังก์ชันเกาเสียนเรเดียลเบสิสสำหรับข้อมูลที่ใช้สอนแบบ Product of exponential and sine function ก) ข้อมูลที่ใช้สอนและได้จากการทดลอง ข) ผลของความผิดพลาด

จากรูปที่ 4.4 (ก) เป็นการแสดงการเปรียบเทียบฟังก์ชันจริงของข้อมูลที่ใช้สอนและแบบจำลองที่ได้จากการทดลองปรับค่าพารามิเตอร์ต่าง ๆ เมื่อใช้เคอร์เนลฟังก์ชันเป็นฟังก์ชันเกาเสียนเรเดียลเบสิสและใช้รูปแบบข้อมูลที่ใช้สอนเป็น Product of exponential and sine function โดยจุดสีแดงหมายถึงข้อมูลที่ใช้สอน เส้นสีน้ำเงินหมายถึงข้อมูลที่ใช้ทดสอบ เส้นสีเขียวหมายถึงฟังก์ชันการประมาณค่าที่โมเดลสร้างขึ้น ซึ่งจะเห็นได้ว่าฟังก์ชันการประมาณค่าโมเดลที่สร้างขึ้นมีค่าใกล้เคียงกับฟังก์ชันจริงมาก

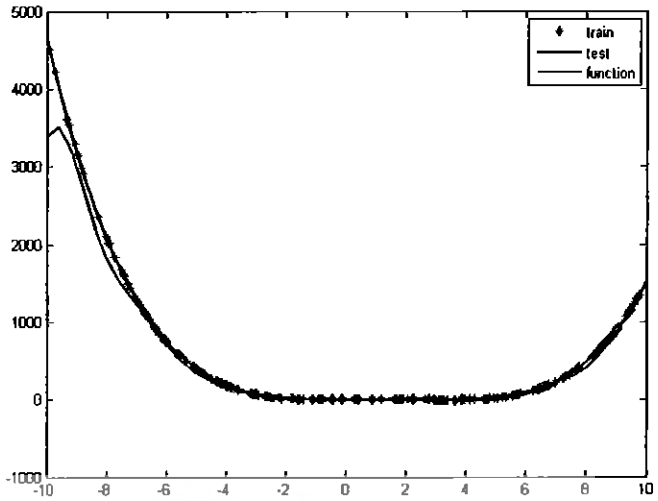
15739640
ว.ร.
ว23/17
2553

จากรูปที่ 4.4 (ข) เป็นผลของค่าความผิดพลาดที่ได้จากการปรับค่าพารามิเตอร์ทั้ง 3 ตัว คือ ค่าเบต้า ค่าอัตราการเรียนรู้และค่าเรกกูลาไรเซชัน เมื่อทดลองปรับค่าพารามิเตอร์ทั้ง 3 ตัวแล้ว จะเห็นได้ว่าเส้นสีแดงที่ได้จากการปรับค่าเบต้า ค่าอัตราการเรียนรู้และค่าเรกกูลาไรเซชันเท่ากับ 200, 1 และ 0.001 ตามลำดับ นั้นมีค่าความผิดพลาดที่ดีที่สุด โดยในช่วงแรกจะลดลงอย่างฉับพลันตั้งแต่รอบที่ 0 – 10 แล้วหลังจากนั้นจะค่อย ๆ ลดลงจนค่าความผิดพลาดนั้นมีลักษณะคงที่

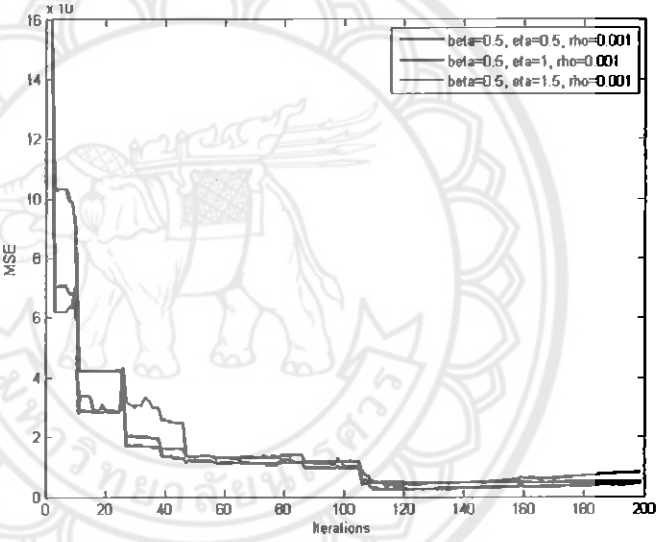
4) ผลของการประมาณค่าฟังก์ชันเมื่อใช้เคอร์เนลฟังก์ชันเป็นเกาเสียนเรเดียลเบสิสโดยใช้ข้อมูลที่ใช้สอนแบบ Polynomial function จะได้กราฟเปรียบเทียบระหว่างฟังก์ชันจริงของข้อมูลและแบบจำลองที่ได้จากการทดลอง และกราฟค่าความผิดพลาดดังแสดงรูปที่ 4.5

จากรูปที่ 4.5 (ก) เป็นการแสดงการเปรียบเทียบฟังก์ชันจริงของข้อมูลที่ใช้สอนและแบบจำลองที่ได้จากการทดลองปรับค่าพารามิเตอร์ต่าง ๆ เมื่อใช้เคอร์เนลฟังก์ชันเป็นฟังก์ชันเกาเสียนเรเดียลเบสิสและใช้รูปแบบข้อมูลที่ใช้สอนเป็น Polynomial function โดยจุดสีแดงหมายถึงข้อมูลที่ใช้สอน เส้นสีน้ำเงินหมายถึงข้อมูลที่ใช้ทดสอบ เส้นสีเขียวหมายถึงฟังก์ชันการประมาณค่าที่โมเดลสร้างขึ้น ซึ่งจะเห็นได้ว่าฟังก์ชันการประมาณค่าโมเดลที่สร้างขึ้นมีค่าใกล้เคียงกับฟังก์ชันจริงมาก

จากรูปที่ 4.5 (ข) เป็นผลของค่าความผิดพลาดที่ได้จากการปรับค่าพารามิเตอร์ทั้ง 3 ตัว คือ ค่าเบต้า ค่าอัตราการเรียนรู้และค่าเรกกูลาไรเซชัน เมื่อทดลองปรับค่าพารามิเตอร์ทั้ง 3 ตัวแล้ว จะเห็นได้ว่าเส้นสีแดงที่ได้จากการปรับค่าเบต้า ค่าอัตราการเรียนรู้และค่าเรกกูลาไรเซชันเท่ากับ 0.5, 1 และ 0.001 ตามลำดับ นั้นมีค่าความผิดพลาดที่ดีที่สุด โดยในช่วงแรกจะลดลงอย่างฉับพลันตั้งแต่รอบที่ 0 – 10 แล้วหลังจากนั้นจะค่อย ๆ ลดลงจนค่าความผิดพลาดนั้นมีลักษณะคงที่



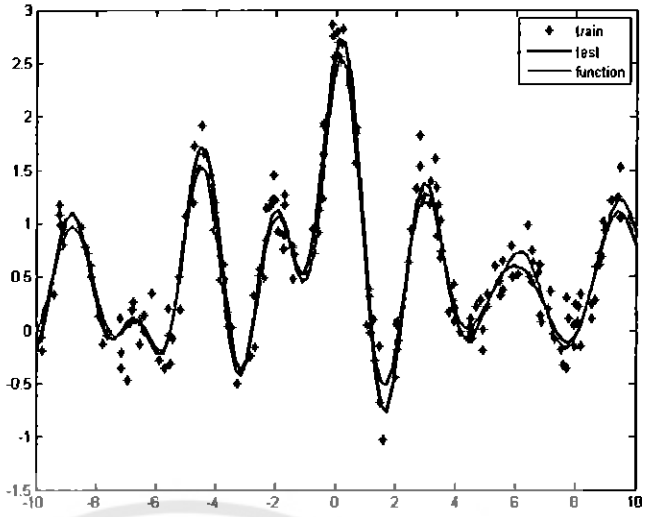
(ก)



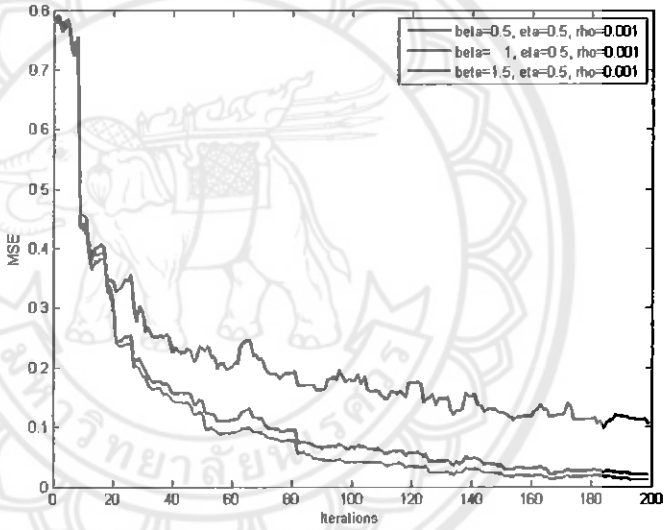
(ข)

รูปที่ 4.5 การประมาณค่าด้วยวิธีเคอร์เนลเป็นฟังก์ชันเกาเสียนเรเดียลเบสิสสำหรับข้อมูลที่ใช้สอนแบบ Polynomial function ก) ข้อมูลที่ใช้สอนและได้จากการทดลอง ข) ผลของความผิดพลาด

5) ผลของการประมาณค่าฟังก์ชันเมื่อใช้เคอร์เนลฟังก์ชันเป็นเกาเสียนเรเดียลเบสิสโดยใช้ข้อมูลที่ใช้สอนแบบ Paley-Wiener function จะได้กราฟเปรียบเทียบระหว่างฟังก์ชันจริงของข้อมูลและแบบจำลองที่ได้จากการทดลอง และกราฟค่าความผิดพลาดดังแสดงรูปที่ 4.6



(ก)



(ข)

รูปที่ 4.6 การประมาณค่าด้วยวิธีเคอร์เนลเป็นฟังก์ชันเกาเสียนเรเดียลเบสิสสำหรับข้อมูลที่ใช้สอนแบบ Paley-Wiener function ก) ข้อมูลที่ใช้สอนและได้จากการทดลอง ข) ผลของความผิดพลาด

จากรูปที่ 4.6 (ก) เป็นการแสดงการเปรียบเทียบฟังก์ชันจริงของข้อมูลที่ใช้สอนและแบบจำลองที่ได้จากการทดลองปรับค่าพารามิเตอร์ต่าง ๆ เมื่อใช้เคอร์เนลฟังก์ชันเป็นฟังก์ชันเกาเสียนเรเดียลเบสิสและใช้รูปแบบข้อมูลที่ใช้สอนเป็น Paley-Wiener function โดยจุดสีแดงหมายถึงข้อมูลที่ใช้สอน เส้นสีน้ำเงินหมายถึงข้อมูลที่ใช้ทดสอบ เส้นสีเขียวหมายถึงฟังก์ชันการประมาณค่าที่โมเดลสร้างขึ้น ซึ่งจะเห็นได้ว่าฟังก์ชันการประมาณค่าโมเดลที่สร้างขึ้นมีค่าใกล้เคียงกับฟังก์ชันจริงมาก

จากรูปที่ 4.6 (ข) เป็นผลของค่าความผิดพลาดที่ได้จากการปรับค่าพารามิเตอร์ทั้ง 3 ตัว คือ ค่าเบต้า ค่าอัตราการเรียนรู้และค่าเรกกูลาไรเซชัน เมื่อทดลองปรับค่าพารามิเตอร์ทั้ง 3 ตัวแล้ว จะเห็นได้ว่าเส้นสีเขียวที่ได้จากการปรับค่าเบต้า ค่าอัตราการเรียนรู้และค่าเรกกูลาไรเซชันเท่ากับ 1.5, 0.5 และ 0.001 ตามลำดับ นั้นมีค่าความผิดพลาดที่ดีที่สุด โดยในช่วงแรกจะลดลงอย่างฉับพลันตั้งแต่รอบที่ 0 – 10 แล้วหลังจากนั้นจะค่อย ๆ ลดลงจนค่าความผิดพลาดนั้นมีลักษณะคงที่

4.1.2 การประมาณค่าโดยใช้เคอร์เนลฟังก์ชันเป็นซิงค์ฟังก์ชัน

การประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนลโดยข้อมูลที่ใช้สอนมีทั้งหมด 5 รูปแบบ และมีการเลือกใช้เคอร์เนลฟังก์ชันเป็นซิงค์ฟังก์ชัน จากนั้นจะทดลองปรับค่าพารามิเตอร์ต่าง ๆ ให้ได้ค่าความผิดพลาดที่ดีที่สุด โดยจะมีการปรับค่าพารามิเตอร์ 3 ตัว คือค่าเบต้า ค่าอัตราการเรียนรู้ และค่าเรกกูลาไรเซชันซึ่งทำการเปรียบเทียบแสดงได้ในตารางที่ 4.2 ดังต่อไปนี้

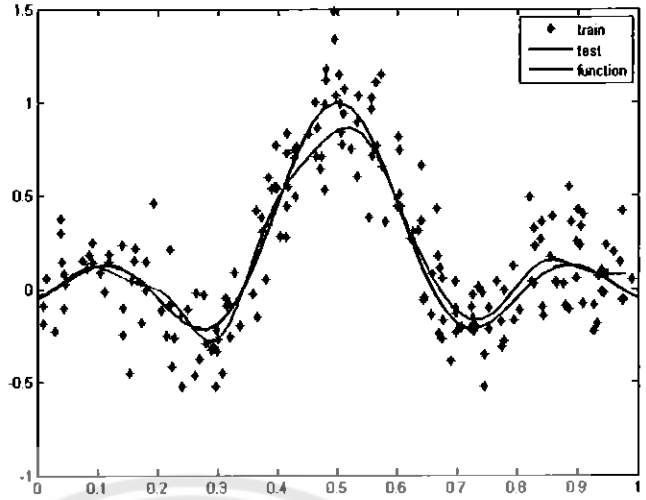
จากตารางที่ 4.2 การปรับค่าพารามิเตอร์ในแต่ละข้อมูลที่ใช้สอนโดยใช้เคอร์เนลฟังก์ชันเป็นซิงค์ฟังชั้น ลักษณะการปรับค่าพารามิเตอร์แต่ละตัวนั้นมีลักษณะเหมือนกับการปรับค่าพารามิเตอร์ที่ใช้เคอร์เนลฟังก์ชันเป็นเกาเสียนเรเดียลเบสิสโดยการปรับค่าพารามิเตอร์ในแต่ละข้อมูลที่ใช้สอนมีลักษณะเหมือนกันดังที่กล่าวไว้ในการปรับค่าในแต่ละข้อมูลที่ใช้สอนของฟังก์ชันเกาเสียนเรเดียลเบสิสที่ได้อธิบายไว้ในข้างต้น

ตารางที่ 4.2 ค่าพารามิเตอร์สำหรับเคอร์เนลฟังก์ชันแบบซิงค์ฟังก์ชัน

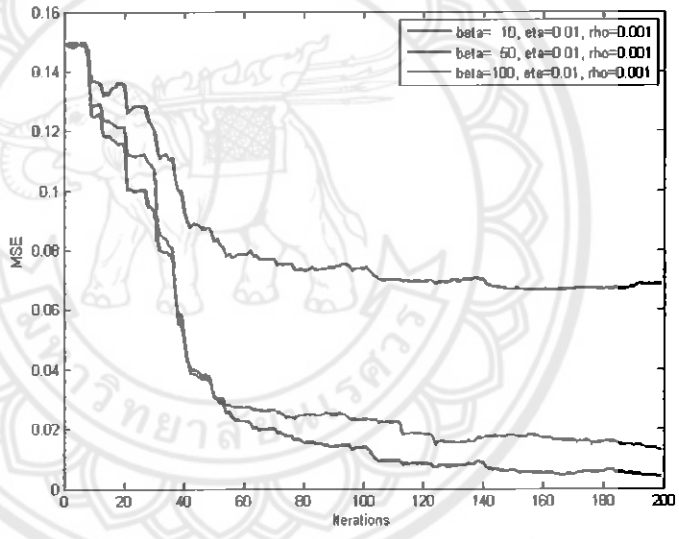
ข้อมูลที่ใช้สอน	β	η	ρ
Sinc function	10	0.01	0.001
	50	0.01	0.001
	100	0.01	0.001
Exponential function	50	0.01	0.001
	50	0.03	0.001
	50	0.05	0.001
Product of exponential and sine function	50	0.01	0.001
	50	0.03	0.001
	50	0.05	0.001
Polynomial function	1.5	0.5	0.001
	1.5	1	0.001
	1.5	1.5	0.001
Paley-Wiener function	1	0.5	0.001
	5	0.5	0.001
	10	0.5	0.001

การประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนลโดยใช้ซิงค์ฟังก์ชันเป็นเคอร์เนลฟังก์ชันโดยใช้ค่าพารามิเตอร์ต่าง ๆ ตามตารางที่ 4.2 ค่าความผิดพลาดของแต่ละรูปแบบข้อมูลที่ใช้สอนแสดงได้ดังนี้

1) ผลของการประมาณค่าฟังก์ชันเมื่อใช้เคอร์เนลฟังก์ชันเป็นซิงค์ฟังก์ชัน โดยใช้ข้อมูลที่ใช้สอนแบบ Sinc function จะได้กราฟเปรียบเทียบระหว่างฟังก์ชันจริงของข้อมูลและแบบจำลองที่ได้จากการทดลอง และกราฟค่าความผิดพลาดดังแสดงรูปที่ 4.7



(ก)



(ข)

รูปที่ 4.7 การประมาณค่าด้วยวิธีเคอร์เนลเป็นซิงค์ฟังก์ชันสำหรับข้อมูลที่ใช้สอนแบบ Sinc function ก) ข้อมูลที่ใช้สอนและได้จากการทดลอง ข) ผลของความผิดพลาด

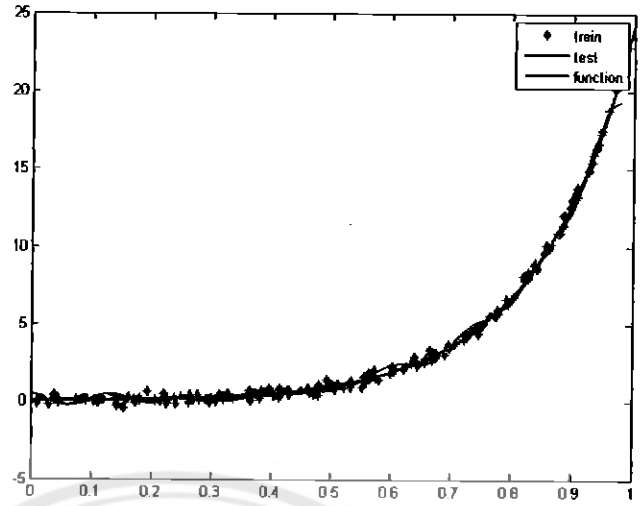
จากรูปที่ 4.7 (ก) เป็นการแสดงการเปรียบเทียบฟังก์ชันจริงของข้อมูลที่ใช้สอนและแบบจำลองที่ได้จากการทดลองปรับค่าพารามิเตอร์ต่าง ๆ เมื่อใช้เคอร์เนลฟังก์ชันเป็นซิงค์ฟังก์ชันและใช้รูปแบบข้อมูลที่ใช้สอนเป็น Sinc function โดยจุดสีแดงหมายถึงข้อมูลที่ใช้สอน เส้นสีน้ำเงินหมายถึงข้อมูลที่ใช้ทดสอบ เส้นสีเขียวหมายถึงฟังก์ชันการประมาณค่าที่โมเดลสร้างขึ้น ซึ่งจะเห็นได้ว่าฟังก์ชันการประมาณค่าโมเดลที่สร้างขึ้นมีค่าใกล้เคียงกับฟังก์ชันจริงมาก

จากรูปที่ 4.7 (ข) เป็นผลของค่าความผิดพลาดที่ได้จากการปรับค่าพารามิเตอร์ทั้ง 3 ตัว คือ ค่าเบต้า ค่าอัตราการเรียนรู้และค่าเรกกูลาไรเซชัน เมื่อทดลองปรับค่าพารามิเตอร์ทั้ง 3 ตัวแล้ว จะเห็นได้ว่าเส้นสีแดงที่ได้จากการปรับค่าเบต้า ค่าอัตราการเรียนรู้และค่าเรกกูลาไรเซชันเท่ากับ 50, 0.01 และ 0.001 ตามลำดับ นั้นมีค่าความผิดพลาดที่ดีที่สุด โดยในช่วงแรกจะลดลงอย่างฉับพลันตั้งแต่รอบที่ 0 – 40 แล้วหลังจากนั้นจะค่อย ๆ ลดลงจนค่าความผิดพลาดนั้นมีลักษณะคงที่

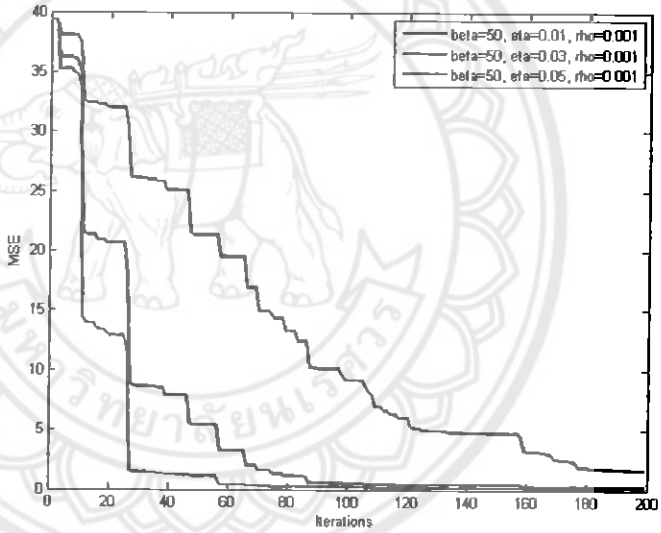
2) ผลของการประมาณค่าฟังก์ชันเมื่อใช้เคอร์เนลฟังก์ชันเป็นซิงค์ฟังก์ชันโดยใช้ข้อมูลที่ใช้สอนแบบ Exponential function จะได้กราฟเปรียบเทียบระหว่างฟังก์ชันจริงของข้อมูลและแบบจำลองที่ได้จากการทดลอง และกราฟค่าความผิดพลาดดังแสดงรูปที่ 4.8

จากรูปที่ 4.8 (ก) เป็นการแสดงการเปรียบเทียบฟังก์ชันจริงของข้อมูลที่ใช้สอนและแบบจำลองที่ได้จากการทดลองปรับค่าพารามิเตอร์ต่าง ๆ เมื่อใช้เคอร์เนลฟังก์ชันเป็นซิงค์ฟังก์ชันและใช้รูปแบบข้อมูลที่ใช้สอนเป็น Exponential function โดยจุดสีแดงหมายถึงข้อมูลที่ใช้สอน เส้นสีน้ำเงินหมายถึงข้อมูลที่ใช้ทดสอบ เส้นสีเขียวหมายถึงฟังก์ชันการประมาณค่าที่โมเดลสร้างขึ้น ซึ่งจะเห็นได้ว่าฟังก์ชันการประมาณค่าโมเดลที่สร้างขึ้นมีค่าใกล้เคียงกับฟังก์ชันจริงมาก

จากรูปที่ 4.8 (ข) เป็นผลของค่าความผิดพลาดที่ได้จากการปรับค่าพารามิเตอร์ทั้ง 3 ตัว คือ ค่าเบต้า ค่าอัตราการเรียนรู้และค่าเรกกูลาไรเซชัน เมื่อทดลองปรับค่าพารามิเตอร์ทั้ง 3 ตัวแล้ว จะเห็นได้ว่าเส้นสีเขียวที่ได้จากการปรับค่าเบต้า ค่าอัตราการเรียนรู้และค่าเรกกูลาไรเซชันเท่ากับ 50, 0.05 และ 0.001 ตามลำดับ นั้นมีค่าความผิดพลาดที่ดีที่สุด โดยในช่วงแรกจะลดลงอย่างฉับพลันตั้งแต่รอบที่ 0 – 25 แล้วหลังจากนั้นจะค่อย ๆ ลดลงจนค่าความผิดพลาดนั้นมีลักษณะคงที่



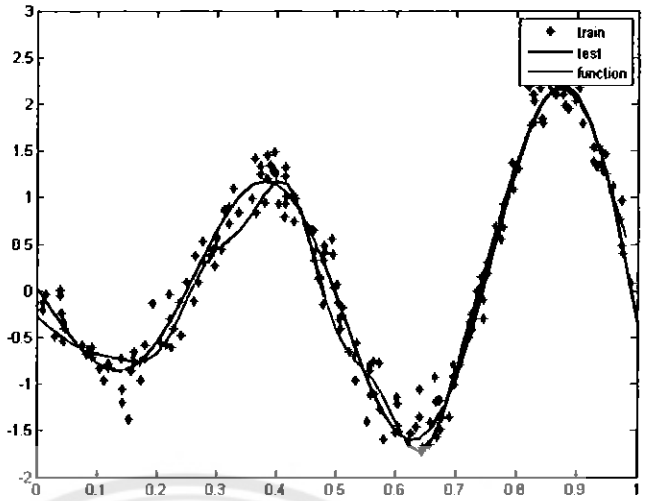
(ก)



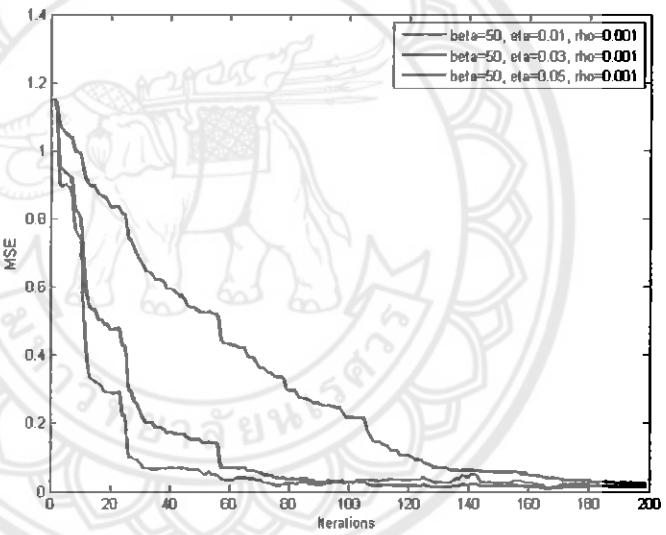
(ข)

รูปที่ 4.8 การประมาณค่าด้วยวิธีเคอร์เนลเป็นซิงค์ฟังก์ชันสำหรับข้อมูลที่ใช้สอนแบบ Exponential function ก) ข้อมูลที่ใช้สอนและได้จากการทดลอง ข) ผลของความผิดพลาด

3) ผลของการประมาณค่าฟังก์ชันเมื่อใช้เคอร์เนลฟังก์ชันเป็นซิงค์ฟังก์ชัน โดยใช้ข้อมูลที่ใช้สอนแบบ Product of exponential and sine function จะได้กราฟเปรียบเทียบระหว่างฟังก์ชันจริงของข้อมูลและแบบจำลองที่ได้จากการทดลอง และกราฟค่าความผิดพลาดดังแสดงรูปที่ 4.9



(ก)



(ข)

รูปที่ 4.9 การประมาณค่าด้วยวิธีเคอร์เนลเป็นซิงค์ฟังก์ชันสำหรับข้อมูลที่ใช้สอนแบบ Product exponential and sine function ก) ข้อมูลที่ใช้สอนและได้จากการทดลอง ข) ผลของความผิดพลาด

จากรูปที่ 4.9 (ก) เป็นการแสดงการเปรียบเทียบฟังก์ชันจริงของข้อมูลที่ใช้สอนและแบบจำลองที่ได้จากการทดลองปรับค่าพารามิเตอร์ต่าง ๆ เมื่อใช้เคอร์เนลฟังก์ชันเป็นซิงค์ฟังก์ชันและใช้รูปแบบข้อมูลที่ใช้สอนเป็น Product exponential and sine function โดยจุดสีแดงหมายถึงข้อมูลที่ใช้สอน เส้นสีน้ำเงินหมายถึงข้อมูลที่ใช้ทดสอบ เส้นสีเขียวหมายถึงฟังก์ชันการประมาณ

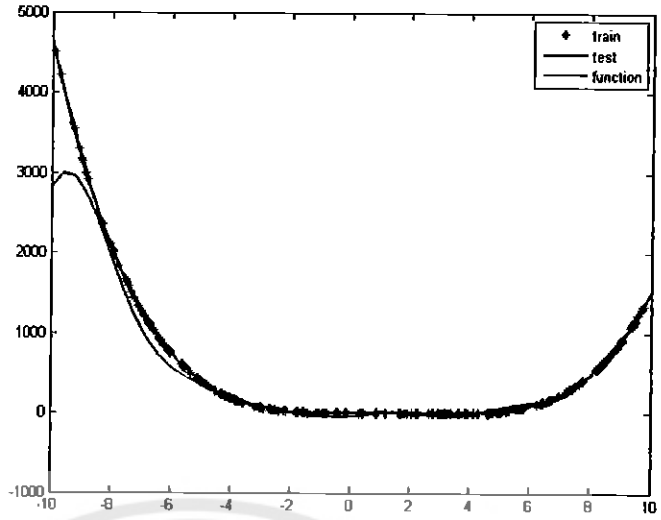
ค่าที่โมเดลสร้างขึ้น ซึ่งจะเห็นได้ว่าฟังก์ชันการประมาณค่าโมเดลที่สร้างขึ้นมีค่าใกล้เคียงกับฟังก์ชันจริงมาก

จากรูปที่ 4.9 (ข) เป็นผลของค่าความผิดพลาดที่ได้จากการปรับค่าพารามิเตอร์ทั้ง 3 ตัว คือ ค่าเบต้า ค่าอัตราการเรียนรู้และค่าแรกกูลาไรเซชัน เมื่อทดลองปรับค่าพารามิเตอร์ทั้ง 3 ตัวแล้ว จะเห็นได้ว่าเส้นสีเขียวที่ได้จากการปรับค่าเบต้า ค่าอัตราการเรียนรู้และค่าแรกกูลาไรเซชันเท่ากับ 50, 0.05 และ 0.001 ตามลำดับ นั้นมีค่าความผิดพลาดที่ดีที่สุด โดยในช่วงแรกจะลดลงอย่างฉับพลันตั้งแต่รอบที่ 0 – 20 แล้วหลังจากนั้นจะค่อย ๆ ลดลงจนค่าความผิดพลาดนั้นมีลักษณะคงที่

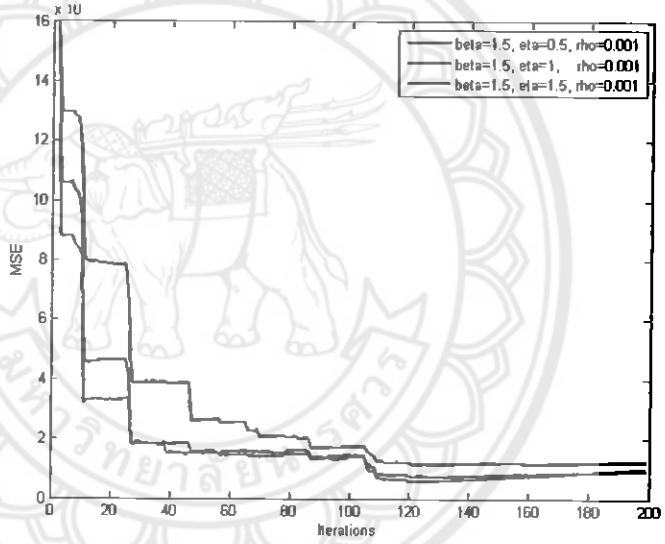
4) ผลของการประมาณค่าฟังก์ชันเมื่อใช้เคอร์เนลฟังก์ชันเป็นซิงค์ฟังก์ชันโดยใช้ข้อมูลที่ใช้สอนแบบ Polynomial function จะได้กราฟเปรียบเทียบระหว่างฟังก์ชันจริงของข้อมูลและแบบจำลองที่ได้จากการทดลอง และกราฟค่าความผิดพลาดดังแสดงรูปที่ 4.10

จากรูปที่ 4.10 (ก) เป็นการแสดงการเปรียบเทียบฟังก์ชันจริงของข้อมูลที่ใช้สอนและแบบจำลองที่ได้จากการทดลองปรับค่าพารามิเตอร์ต่าง ๆ เมื่อใช้เคอร์เนลฟังก์ชันเป็นซิงค์ฟังก์ชันและใช้รูปแบบข้อมูลที่ใช้สอนเป็น Polynomial function โดยจุดสีแดงหมายถึงข้อมูลที่ใช้สอน เส้นสีน้ำเงินหมายถึงข้อมูลที่ใช้ทดสอบ เส้นสีเขียวหมายถึงฟังก์ชันการประมาณค่าที่โมเดลสร้างขึ้น ซึ่งจะเห็นได้ว่าฟังก์ชันการประมาณค่าโมเดลที่สร้างขึ้นมีค่าใกล้เคียงกับฟังก์ชันจริงมาก

จากรูปที่ 4.10 (ข) เป็นผลของค่าความผิดพลาดที่ได้จากการปรับค่าพารามิเตอร์ทั้ง 3 ตัว คือ ค่าเบต้า ค่าอัตราการเรียนรู้และค่าแรกกูลาไรเซชัน เมื่อทดลองปรับค่าพารามิเตอร์ทั้ง 3 ตัวแล้ว จะเห็นได้ว่าเส้นสีเขียวที่ได้จากการปรับค่าเบต้า ค่าอัตราการเรียนรู้และค่าแรกกูลาไรเซชันเท่ากับ 1.5, 1.5 และ 0.001 ตามลำดับ นั้นมีค่าความผิดพลาดที่ดีที่สุด โดยในช่วงแรกจะลดลงอย่างฉับพลันตั้งแต่รอบที่ 0 – 10 แล้วหลังจากนั้นจะค่อย ๆ ลดลงจนค่าความผิดพลาดนั้นมีลักษณะคงที่



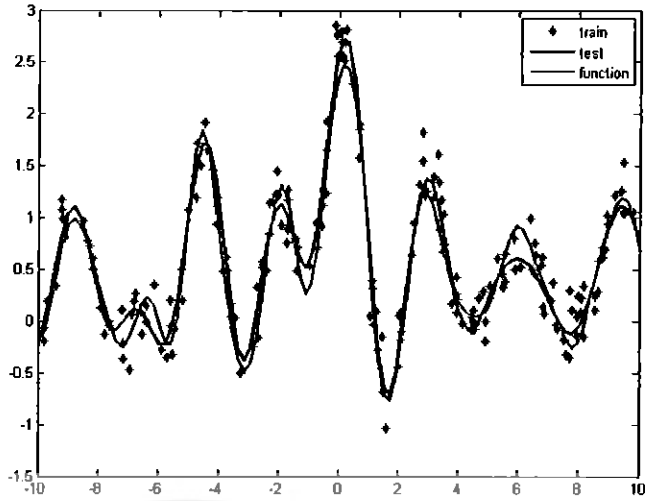
(ก)



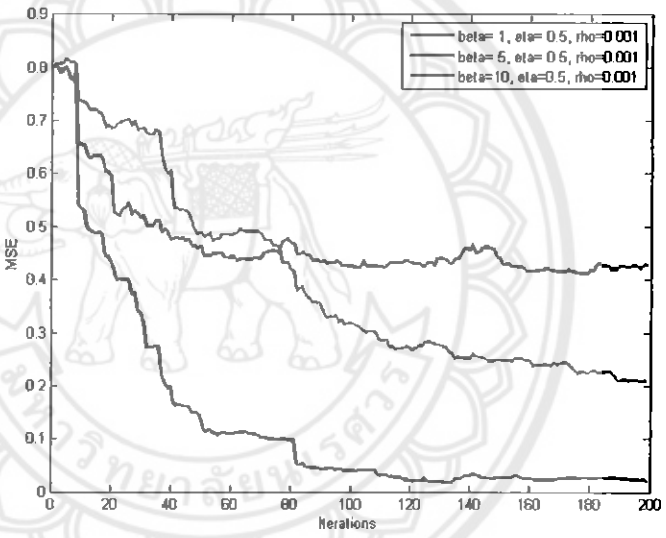
(ข)

รูปที่ 4.10 การประมาณค่าด้วยวิธีเคอร์เนลเป็นซิงค์ฟังก์ชันสำหรับข้อมูลที่ใช้สอนแบบ Polynomial function ก) ข้อมูลที่ใช้สอนและได้จากการทดลอง ข) ผลของความผิดพลาด

5) ผลของการประมาณค่าฟังก์ชันเมื่อใช้เคอร์เนลฟังก์ชันเป็นซิงค์ฟังก์ชัน โดยใช้ข้อมูลที่ใช้สอนแบบ Paley-Wiener function จะได้กราฟเปรียบเทียบระหว่างฟังก์ชันจริงของข้อมูลและแบบจำลองที่ได้จากการทดลอง และกราฟค่าความผิดพลาดดังแสดงรูปที่ 4.11



(ก)



(ข)

รูปที่ 4.11 การประมาณค่าด้วยวิธีเคอร์เนลเป็นซิงค์ฟังก์ชันสำหรับข้อมูลที่ใช้สอนแบบ Paley-Wiener function ก) ข้อมูลที่ใช้สอนและได้จากการทดลอง ข) ผลของความผิดพลาด

จากรูปที่ 4.11 (ก) เป็นการแสดงการเปรียบเทียบฟังก์ชันจริงของข้อมูลที่ใช้สอนและแบบจำลองที่ได้จากการทดลองปรับค่าพารามิเตอร์ต่าง ๆ เมื่อใช้เคอร์เนลฟังก์ชันเป็นซิงค์ฟังก์ชัน และใช้รูปแบบข้อมูลที่ใช้สอนเป็น Paley-Wiener โดยจุดสีแดงหมายถึงข้อมูลที่ใช้สอน เส้นสีน้ำเงินหมายถึงข้อมูลที่ใช้ทดสอบ เส้นสีเขียวหมายถึงฟังก์ชันการประมาณค่าที่โมเดลสร้างขึ้น ซึ่งจะเห็นได้ว่าฟังก์ชันการประมาณค่าโมเดลที่สร้างขึ้นมีค่าใกล้เคียงกับฟังก์ชันจริงมาก

จากรูปที่ 4.11 (ข) เป็นผลของค่าความผิดพลาดที่ได้จากการปรับค่าพารามิเตอร์ทั้ง 3 ตัว คือ ค่าเบต้า ค่าอัตราการเรียนรู้และค่าเรกกูลาไรเซชัน เมื่อทดลองปรับค่าพารามิเตอร์ทั้ง 3 ตัวแล้ว จะเห็นได้ว่าเส้นสีแดงที่ได้จากการปรับค่าเบต้า ค่าอัตราการเรียนรู้และค่าเรกกูลาไรเซชันเท่ากับ 5, 0.5 และ 0.001 ตามลำดับ นั้นมีค่าความผิดพลาดที่ดีที่สุด โดยในช่วงแรกจะลดลงอย่างฉับพลัน ตั้งแต่รอบที่ 0 – 10 แล้วหลังจากนั้นจะค่อย ๆ ลดลงจนค่าความผิดพลาดนั้นมีลักษณะคงที่

บทที่ 5
สรุปและข้อเสนอแนะ

ในบทนี้จะกล่าวถึงผลการทดลองที่เกิดขึ้นในการหาค่าความผิดพลาดด้วยวิธีการประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนลแบบออนไลน์ โดยใช้ส่วนต่อประสานกราฟิกกับผู้ใช้ว่าในแต่ละเคอร์เนลฟังก์ชันและค่าพารามิเตอร์ต่าง ๆ กันจะมีผลต่อความผิดพลาดที่เพิ่มขึ้นหรือลดลงอย่างไร เพื่อที่จะหาค่าที่เหมาะสมเพื่อให้ได้ค่าความผิดพลาดที่ดีที่สุด

5.1 สรุปผลการทดลอง

โครงงานนี้ได้ศึกษาการประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนลโดยใช้ซิงค์ฟังก์ชันและฟังก์ชันเกาเสียนเรเดียลเบสิสเป็นเคอร์เนลฟังก์ชันในแต่ละรูปแบบข้อมูลที่ใช้สอนต่าง ๆ เพื่อหาค่าความผิดพลาดจากข้อมูลที่ใช้ทดลอง แล้วเลือกค่าความผิดพลาดที่มีค่าน้อยที่สุดมาใช้เป็นแบบจำลองต่อไป

5.1.1 ค่าพารามิเตอร์ที่ดีที่สุดของการประมาณค่าโดยใช้เคอร์เนลฟังก์ชันเป็นฟังก์ชันเกาเสียนเรเดียลเบสิส

จากการทดลองในบทที่ 4 ในแต่ละรูปแบบข้อมูลทดสอบจะมีการปรับค่าพารามิเตอร์ 3 ตัวคือ ค่าเบต้า ค่าอัตราการเรียนรู้ และค่าเรกกูลาไรเซชัน เพื่อหาค่าความผิดพลาดที่ดีที่สุด ซึ่งในแต่ละรูปแบบข้อมูลทดสอบก็จะมีพารามิเตอร์ที่เหมาะสมแตกต่างกันออกไป ค่าพารามิเตอร์ที่เหมาะสมในแต่ละรูปแบบข้อมูลทดสอบแสดงในตารางที่ 5.1

ตารางที่ 5.1 ค่าพารามิเตอร์ที่เหมาะสมโดยใช้เคอร์เนลฟังก์ชันเป็นฟังก์ชันเกาเสียนเรเดียลเบสิส

ข้อมูลที่ใช้สอน	β	η	ρ
Sinc function	100	0.1	0.001
Exponential function	150	1	0.001
Product of exponential and sine function	200	1	0.001
Polynomial function	0.5	1	0.001
Paley-Wiener function	1.5	0.5	0.001

5.1.2 การหาค่าพารามิเตอร์ที่ดีที่สุดของการประมาณค่าโดยใช้เคอร์เนลฟังก์ชันเป็นซิงค์ฟังก์ชัน

จากการทดลองในบทที่ 4 ในแต่ละรูปแบบข้อมูลทดสอบจะมีการปรับค่าพารามิเตอร์ 3 ตัวคือ ค่าเบต้า ค่าอัตราการเรียนรู้ และค่าเรกกูลาไรเซชัน เพื่อหาค่าความผิดพลาดที่ดีที่สุด ซึ่งในแต่ละรูปแบบข้อมูลก็จะมีพารามิเตอร์ที่เหมาะสมแตกต่างกันออกไป ค่าพารามิเตอร์ที่เหมาะสมในแต่ละรูปแบบข้อมูลมีดังนี้

ตารางที่ 5.2 ค่าพารามิเตอร์ที่เหมาะสมโดยใช้เคอร์เนลฟังก์ชันเป็นซิงค์ฟังก์ชัน

ข้อมูลที่ใช้สอน	β	η	ρ
Sinc function	50	0.01	0.001
Exponential function	50	0.05	0.001
Product of exponential and sine function	50	0.05	0.001
Polynomial function	1.5	1.5	0.001
Paley-Wiener function	5	0.5	0.001

5.2 ปัญหาและแนวทางแก้ไข

ในการทดลองการประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนลแบบออนไลน์โดยใช้ส่วนต่อประสานกราฟิกกับผู้ใช้นั้น ควรจะมีการทดสอบกับข้อมูลจริงเพื่อที่จะได้รู้ว่าฟังก์ชันเคอร์เนลในแต่ละฟังก์ชันมีค่าพารามิเตอร์ใดที่เหมาะสม ซึ่งในแต่ละฟังก์ชันก็จะมีค่าพารามิเตอร์ที่แตกต่างกันออกไป เมื่อมีการทดสอบกับข้อมูลจริงแล้วก็จะทำให้รู้ว่าฟังก์ชันแบบไหนเหมาะกับข้อมูลแบบไหน ค่าพารามิเตอร์แต่ละตัวนั้นมีผลอย่างไร ทำให้การนำไปประยุกต์ใช้งานมีความน่าเชื่อถือมากยิ่งขึ้น

เอกสารอ้างอิง

[1] สัญจร วุฒิสิทธิกุลกิจ และคณะ, "MATLAB การประยุกต์ใช้งานทางวิศวกรรมไฟฟ้า", พิมพ์ครั้งที่ 3, สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย. 2551

[2] ปริญญา สงวนสัตย์. "คู่มือ MATLAB ฉบับสมบูรณ์",กรุงเทพ: สำนักพิมพ์ไอดีซี พรีเมียร์. 2554

[3] T.J. Dodd, "Gradient descent approach to approximation in reproducing kernel Hilbertspaces, Department of Automatic Control and Systems Engineering, University of Sheffield, UK, Tech. Rep. 821, 2002.

[4] N. Aronszajn, "Theory of reproducing kernels," Transactions of the American Mathermatical society, vol. 68, pp. 337-404, 1950.

[5] พิเชษฐ์ เทือกถา และภูษิต พิสกุล, "การเรียนรู้แบบออนไลน์ด้วยวิธีเคอร์เนล", ปริญญานิพนธ์ วศ.บ., มหาวิทยาลัยนเรศวร, พิษณุโลก. 2553

[6] J. Kivinen, A. Smola, and R. Williamson, "Online learning with kernels", in Advances in Neural Processing Systems, 2001.

[7] S. Smale and Y. Yoa, "Online learning algorithms", October 2004, Unpublished.

ภาคผนวก

โปรแกรมที่ใช้ในการทดลอง

```
function varargout = project_final1update(varargin)
% PROJECT_FINAL1UPDATE M-file for project_final1update.fig
%      PROJECT_FINAL1UPDATE, by itself, creates a new PROJECT_FINAL1UPDATE or raises the existing
%      singleton*.
%
%      H = PROJECT_FINAL1UPDATE returns the handle to a new PROJECT_FINAL1UPDATE or the handle to
%      the existing singleton*.
%
%      PROJECT_FINAL1UPDATE('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in PROJECT_FINAL1UPDATE.M with the given input arguments.
%
%      PROJECT_FINAL1UPDATE('Property','Value',...) creates a new PROJECT_FINAL1UPDATE or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before project_final1update_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to project_final1update_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help project_final1update

% Last Modified by GUIDE v2.5 06-May-2011 00:46:21

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @project_final1update_OpeningFcn, ...
                   'gui_OutputFcn',  @project_final1update_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
```

```
                'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before project_final1update is made visible.
function project_final1update_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to project_final1update (see VARARGIN)

% Choose default command line output for project_final1update
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes project_final1update wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = project_final1update_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
showm(handles.radiobutton3)
showm(handles.radiobutton4)
showm(handles.radiobutton5)
showm(handles.radiobutton6)
showm(handles.radiobutton7)
hidem(handles.radiobutton1)
hidem(handles.radiobutton2)
showm(handles.text1)
hidem(handles.text2)
hidem(handles.text3)
hidem(handles.text4)
hidem(handles.text5)
hidem(handles.text6)
hidem(handles.text7)
hidem(handles.text8)
hidem(handles.text9)
hidem(handles.text10)
hidem(handles.text11)
hidem(handles.text12)
hidem(handles.beta)
hidem(handles.eta)
hidem(handles.rho)
hidem(handles.start)
hidem(handles.pushbutton6)
cla(handles.axes1,'reset')
set(handles.axes1,'visible','off')
```

```
function beta_Callback(hObject, eventdata, handles)
% hObject    handle to beta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of beta as text
%        str2double(get(hObject,'String')) returns contents of beta as a double


d1= str2double(get(hObject, 'String'));
if or(isnan(d1),d1<0)
   set(hObject, 'String', 0);
   errordlg('Input must be a number >= 0','Error');
end


% Save the new density value
handles.metricdata.d1 = d1;
guidata(hObject,handles)


% --- Executes during object creation, after setting all properties.
function beta_CreateFcn(hObject, eventdata, handles)
% hObject    handle to beta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called


% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end



% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
showm(handles.radiobutton1)
showm(handles.radiobutton2)
hidem(handles.radiobutton3)
hidem(handles.radiobutton4)
hidem(handles.radiobutton5)
hidem(handles.radiobutton6)
hidem(handles.radiobutton7)
showm(handles.beta)
showm(handles.text2)
showm(handles.text10)
hidem(handles.text1)
hidem(handles.text3)
hidem(handles.text4)
hidem(handles.text5)
hidem(handles.text6)
hidem(handles.text7)
hidem(handles.text8)
hidem(handles.text9)
hidem(handles.text11)
hidem(handles.text12)
hidem(handles.start)
hidem(handles.pushbutton6)
hidem(handles.eta)
hidem(handles.rho)
hidem(handles.axes1)
cla(handles.axes1,'reset')
set(handles.axes1,'visible','off')
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

showm(handles.eta)
showm(handles.rho)
showm(handles.text3)
```

```
showm(handles.text4)
showm(handles.text5)
showm(handles.text11)
showm(handles.text12)
hidem(handles.radiobutton1)
hidem(handles.radiobutton2)
hidem(handles.radiobutton3)
hidem(handles.radiobutton4)
hidem(handles.radiobutton5)
hidem(handles.radiobutton6)
hidem(handles.radiobutton7)
hidem(handles.text1)
hidem(handles.text2)
hidem(handles.text6)
hidem(handles.text7)
hidem(handles.text8)
hidem(handles.text9)
hidem(handles.text10)
hidem(handles.start)
hidem(handles.pushbutton6)
hidem(handles.beta)
hidem(handles.axes1)
cla(handles.axes1,'reset')
set(handles.axes1,'visible','off')

% --- Executes on button press in radiobutton1.
function radiobutton1_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of radiobutton1

set(handles.radiobutton2,'value',0)

% --- Executes on button press in radiobutton2.
function radiobutton2_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to radiobutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of radiobutton2

set(handles.radiobutton1,'value',0)

function eta_Callback(hObject, eventdata, handles)
% hObject    handle to eta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of eta as text
%        str2double(get(hObject,'String')) returns contents of eta as a double

d1= str2double(get(hObject, 'String'));
if or(isnan(d1),d1<0)
   set(hObject, 'String', 0);
   errordlg('Input must be a number >= 0','Error');
end

% Save the new density value
handles.metricdata.d1 = d1;
guidata(hObject,handles)

% --- Executes during object creation, after setting all properties.
function eta_CreateFcn(hObject, eventdata, handles)
% hObject    handle to eta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
```

```
function rho_Callback(hObject, eventdata, handles)
% hObject    handle to rho (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of rho as text
%        str2double(get(hObject,'String')) returns contents of rho as a double
d1= str2double(get(hObject, 'String'));
if or(isnan(d1),d1<0)
   set(hObject, 'String', 0);
   errordlg('Input must be a number >= 0','Error');
end

% Save the new density value
handles.metricdata.d1 = d1;
guidata(hObject,handles)

% --- Executes during object creation, after setting all properties.
function rho_CreateFcn(hObject, eventdata, handles)
% hObject    handle to rho (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in start.
function start_Callback(hObject, eventdata, handles)
% hObject    handle to start (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

beta=str2num(get(handles.beta,'string'));
eta=str2num(get(handles.eta,'string'));
```

```
rho=str2num(get(handles.rho,'string'));

% clear all; clc
rand('seed',1032423);
randn('seed',42434123);
cc1=get(handles.radiobutton3,'value');
cc2=get(handles.radiobutton4,'value');
cc3=get(handles.radiobutton5,'value');
cc4=get(handles.radiobutton6,'value');
cc5=get(handles.radiobutton7,'value');

if cc1==1
   ccase=1;
elseif cc2==1
   ccase=2;
elseif cc3==1
   ccase=3;
elseif cc4==1
   ccase=4;
elseif cc5==1
   ccase=5;
end
cc6=get(handles.radiobutton1,'value');
cc7=get(handles.radiobutton2,'value');
if cc6==1
    ccase1=1;
 elseif cc7==1
   ccase1=2;
 end


if ccase==1
  x_train = rand(200,1); %ข้อมูลที่ใช้สอน x
  y_train = sinc((20*x_train-10)/pi);
  y_train = y_train + 0.2*randn(size(x_train )); %ข้อมูลที่ใช้สอน y
  x_test  = [0:0.02:1]';%ข้อมูลที่ใช้ทดสอบ x
```

```
    y_idealtest = sinc((20*x_test-10)/pi); %ข้อมูลที่ใช้ทดสอบ y
    nodata=(1:50);

  elseif ccase==2
    x_train = rand(200,1); %ข้อมูลที่ใช้สอน x
    y_train = exp((20*x_train-10)/pi);
    y_train = y_train + 0.2*randn(size(x_train )); %ข้อมูลที่ใช้สอน y
    x_test  = [0:0.02:1]';%ข้อมูลที่ใช้ทดสอบ x
    y_idealtest = exp((20*x_test-10)/pi); %ข้อมูลที่ใช้ทดสอบ y
    nodata=(1:50);

  elseif ccase==3
    x_train = rand(200,1); %ข้อมูลที่ใช้สอน x
    a=((40*x_train-10)/pi);
    y_train = exp(0.1*a).*sin(a);
    y_train = y_train + 0.2*randn(size(x_train )); %ข้อมูลที่ใช้สอน y
    x_test  = [0:0.02:1]';%ข้อมูลที่ใช้ทดสอบ x
    y_idealtest = exp(0.1*(40*x_test-10)/pi).*(sin((40*x_test-10)/pi)); %ข้อมูลที่ใช้ทดสอบ y
    nodata=(1:50);

  elseif ccase==4
    x_train = rand(200,1); %ข้อมูลที่ใช้สอน x
    xx(1:100) = x_train(1:100)*(-10);
    xx(101:200) = x_train(101:200)*(10);
    x_train=xx';
    p = [0.3 -1.6 0.8 2 0.9];
    y_train = polyval(p,x_train);
    y_train = y_train + 0.2*randn(size(x_train)); %ข้อมูลที่ใช้สอน y
    x_test  = [0:0.02:1]';%ข้อมูลที่ใช้ทดสอบ x
    y_idealtest = [0.3 -1.6 0.8 2 0.9];
    x_test=20*x_test-10;
    y_idealtest = polyval(y_idealtest,(x_test));
  %    y_idealtest = polyval(y_idealtest,(20*x_test-10)/pi);
    nodata=(1:50);
```

```
elseif ccase==5
   load traindata1
   x_train;
   y_train;
   x_test;
   y_idealtest;
   nodata=(1:100);
end

no_data = size(x_train,1);
%กำหนดค่าและปรับค่าของพารามิเตอร์
 for n=1
    f=[0];%กำหนดค่าเริ่มต้น
    e(n)=-y_train(n);
    alpha=(-eta)*e(n);
    if cc5==1
       for i=1:100
     x_test(i);
     yidealtest = y_idealtest(i);
     f(i)=alpha*ker1(x_train(n),x_test(i),beta,ccase1);
     ff=[f(i)]';
     error(i)=((ff-yidealtest).^2);
  end
     MSE=sum(error)/length(y_idealtest);
     f_predict=alpha*ker1(x_train(n),x_train(n+1),beta,ccase1);

   else
  for i=1:50
     x_test(i);
     yidealtest = y_idealtest(i);
     f(i)=alpha*ker1(x_train(n),x_test(i),beta,ccase1);
     ff=[f(i)]';
     error(i)=((ff-yidealtest).^2);
  end
     MSE=sum(error)/length(y_idealtest);
     f_predict=alpha*ker1(x_train(n),x_train(n+1),beta,ccase1);
```

```
  end
end
 h=waitbar(0,'Please wait...');
for n=2:199
   e(n)=f_predict-y_train(n);
   alpha=[alpha*(1-eta*rho); -eta*e(n)];

   if cc5==1
    for i=1:100
      x_test(i);
      yidealtest = y_idealtest(i);
      for j=1:n
       ff(j)=alpha(j)*ker1(x_train(j),x_test(i),beta,ccase1);
      end
      y_predict(i)= sum(ff);

      error(i)=((y_predict(i)-y_idealtest(i)).^2);

   end
      MSE(n)=sum(error)/length(y_idealtest);
   else
   for i=1:50
      x_test(i);
      yidealtest = y_idealtest(i);
      for j=1:n
       ff(j)=alpha(j)*ker1(x_train(j),x_test(i),beta,ccase1);
      end
      y_predict(i)= sum(ff);

      error(i)=((y_predict(i)-y_idealtest(i)).^2);

   end
      MSE(n)=sum(error)/length(y_idealtest);
   end
   for j=1:n
       ff(j)=alpha(j)*ker1(x_train(n+1),x_train(j),beta,ccase1);
```

```
    end
      f_predict=sum(ff);
      waitbar(n/200,h)
  end
  close(h)

  MSE; %ค่าเฉลี่ยความผิดพลาดกำลังสอง
 figure(2)
  plot(MSE,'b','LineWidth',1.5) %กราฟของค่าความผิดพลาด
  xlabel('Iterations');
  ylabel('MSE')

 figure(1)
  plot(x_train,y_train,'r*',x_test,y_idealtest,'b',x_test(nodata)',y_predict,'g','LineWidth',1.2)
  legend('train','test','function')
 handles.x_train=x_train;
 handles.y_train=y_train;
 guidata(hObject, handles);
 showm(handles.text8)
 showm(handles.text9)
 function kernel=ker1(x_train,x_test,beta,m)
 if m==1
  kernel=exp(-beta*(abs((x_train-x_test))).^2);
 elseif m==2
    r=norm(x_train-x_test);
    if r==0
       kernel=beta/pi;
    else
       kernel=sin(beta*r)/(pi*r);
    end

 end

 % --- Executes on button press in pushbutton4.
 function pushbutton5_Callback(hObject, eventdata, handles)
 % hObject    handle to pushbutton4 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
showm(handles.pushbutton6)
showm(handles.text7)
hidem(handles.radiobutton1)
hidem(handles.radiobutton2)
hidem(handles.radiobutton3)
hidem(handles.radiobutton4)
hidem(handles.radiobutton5)
hidem(handles.radiobutton6)
hidem(handles.radiobutton7)
hidem(handles.text1)
hidem(handles.text2)
hidem(handles.text3)
hidem(handles.text4)
hidem(handles.text5)
hidem(handles.text6)
hidem(handles.text8)
hidem(handles.text9)
hidem(handles.text10)
hidem(handles.text11)
hidem(handles.text12)
hidem(handles.start)
hidem(handles.beta)
hidem(handles.eta)
hidem(handles.rho)
cla(handles.axes1,'reset')
set(handles.axes1,'visible','off')

% --- Executes on button press in radiobutton3.
function radiobutton3_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of radiobutton3
```

```
set(handles.radiobutton4,'value',0)
set(handles.radiobutton5,'value',0)
set(handles.radiobutton6,'value',0)
set(handles.radiobutton7,'value',0)
rand('seed',1032423);
randn('seed',42434123);
x_train = rand(200,1); %ข้อมูลที่ใช้สอน x
y_train = sinc((20*x_train-10)/pi);
y_train = y_train + 0.2*randn(size(x_train ));
axes(handles.axes1)
plot(x_train,y_train,'r*')

% --- Executes on button press in radiobutton4.
function radiobutton4_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of radiobutton4

set(handles.radiobutton3,'value',0)
set(handles.radiobutton5,'value',0)
set(handles.radiobutton6,'value',0)
set(handles.radiobutton7,'value',0)
rand('seed',1032423);
randn('seed',42434123);
x_train = rand(200,1);
y_train = exp((20*x_train-10)/pi);
y_train = y_train + 0.2*randn(size(x_train ));
axes(handles.axes1)
plot(x_train,y_train,'r*')

% --- Executes on button press in radiobutton3.
function radiobutton5_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of radiobutton3

set(handles.radiobutton3,'value',0)
set(handles.radiobutton4,'value',0)
set(handles.radiobutton6,'value',0)
set(handles.radiobutton7,'value',0)
rand('seed',1032423);
randn('seed',42434123);
x_train = rand(200,1);
    a=((40*x_train-10)/pi);
y_train = exp(0.1*a).*sin(a);
y_train = y_train + 0.2*randn(size(x_train ));
axes(handles.axes1)
plot(x_train,y_train,'r*')

% --- Executes on button press in radiobutton6.
function radiobutton6_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of radiobutton6

set(handles.radiobutton3,'value',0)
set(handles.radiobutton4,'value',0)
set(handles.radiobutton5,'value',0)
set(handles.radiobutton7,'value',0)
rand('seed',1032423);
randn('seed',42434123);
   x_train = rand(100,1);
   xx(1:50) = x_train(1:50)*(-10);
   xx(51:100) = x_train(51:100)*(10);
   x_train=xx';
   p = [0.3 -1.6 0.8 2 0.9];
   y_train = polyval(p,x_train);
y_train = y_train + 0.2*randn(size(x_train)); %ข้อมูลที่ใช้สอน y
```

```
axes(handles.axes1)
plot(x_train,y_train,'r*')

% --- Executes on button press in radiobutton7.
function radiobutton7_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of radiobutton7
set(handles.radiobutton3,'value',0)
set(handles.radiobutton4,'value',0)
set(handles.radiobutton5,'value',0)
set(handles.radiobutton6,'value',0)
load traindata1
x_train;
y_train;
axes(handles.axes1)
plot(x_train,y_train,'r*')

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
showm(handles.start)
showm(handles.text6)
hidem(handles.radiobutton1)
hidem(handles.radiobutton2)
hidem(handles.radiobutton3)
hidem(handles.radiobutton4)
hidem(handles.radiobutton5)
hidem(handles.radiobutton6)
hidem(handles.radiobutton7)
hidem(handles.text1)
hidem(handles.text2)
hidem(handles.text3)
```

```
hidem(handles.text4)
hidem(handles.text5)
hidem(handles.text7)
hidem(handles.text8)
hidem(handles.text9)
hidem(handles.text10)
hidem(handles.text11)
hidem(handles.text12)
hidem(handles.pushbutton6)
hidem(handles.beta)
hidem(handles.eta)
hidem(handles.rho)
cla(handles.axes1,'reset')
set(handles.axes1,'visible','off')

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% --- Executes on button press in pushbutton6.
x_train=handles.x_train;
y_train=handles.y_train;
[filename, pathname] = uiputfile(...
 {'*.mat';'*.*'},...
 'Save as');
dir1=cd;
cd(pathname)
save(filename)
cd(dir1)
```

ประวัติผู้ดำเนินโครงงาน

ชื่อ นายอนุสรณ์ เริญไธสง

ภูมิลำเนา 154 หมู่ 7 ต. ถาวรวัฒนา อ. ทรายทองวัฒนา จ. กำแพงเพชร

ประวัติการศึกษา

- จบการศึกษาจากโรงเรียนทุ่งทรายวิทยา
- ปัจจุบันกำลังศึกษาระดับปริญญาตรี ชั้นปีที่ 4 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail: anusorn_nu_ee@hotmail.com

ชื่อ นายอภิวัฒน์ โค่นบ่อรัง

ภูมิลำเนา 112 หมู่ 4 ต. บ่อรัง อ. วิเชียรบุรี จ. เพชรบูรณ์

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนนิยมศิลป์อนุสรณ์
- ปัจจุบันกำลังศึกษาระดับปริญญาตรี ชั้นปีที่ 4 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail: oat_c.ronaldo@hotmail.com