

การออกแบบนิเวศเน็ตเวิร์กเพื่อประมาณค่าตำแหน่งของสถานีเคลื่อนที่



นายศรัณย์ เกียรติศิริโรจน์

สถาบันวิทยบริการ

จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2548

ISBN 974-17-5702-6

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

DESIGN OF NEURAL NETWORK FOR ESTIMATING LOCATION OF MOBILE STATION

Mr. Saran Kietsiroje

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Electrical Engineering

Department of Electrical Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2005

ISBN 974-17-5702-6

ศรัณย์ เกียรติศิริโรจน์: การออกแบบนิรอรอลเน็ตเวิร์กเพื่อประมาณค่าตำแหน่งของสถานีเคลื่อนที่ (DESIGN OF NEURAL NETWORK FOR ESTIMATING LOCATION OF MOBILE STATION) อ. ที่ปรึกษา: รองศาสตราจารย์ ดร.วาทิต เบญจพลกุล, 106 หน้า. ISBN 974-17-5702-6.

ในช่วงทศวรรษที่ผ่านมา บริการประมาณค่าตำแหน่งในระบบโทรศัพท์เคลื่อนที่ได้รับความนิยมเพิ่มขึ้นอย่างรวดเร็ว ไม่ว่าจะเป็นจากภาครัฐหรือภาคเอกชน จนได้มีการออกเป็นข้อกำหนดเพื่อสร้างเป็นมาตรฐานโดยหน่วยงาน Federal Communications Commission (FCC) กล่าวคือ ตั้งแต่เดือนตุลาคม ค.ศ. 2001 สำหรับระบบสื่อสารเคลื่อนที่ CDMA หรือ GSM จะต้องมีความสามารถรายงานค่าตำแหน่งให้มีความถูกต้องในระยะความผิดพลาด 125 เมตร อย่างน้อย 67% ของจำนวนครั้งการประมาณค่า

วิทยานิพนธ์ฉบับนี้นำเสนอเทคนิคในการออกแบบนิรอรอลเน็ตเวิร์กเพื่อประมาณค่าตำแหน่งของสถานีเคลื่อนที่โดยใช้ค่าความแรงของสัญญาณและตำแหน่งของสถานีฐานที่ติดต่อกับสถานีเคลื่อนที่เป็นข้อมูลเข้าในการฝึกและทดสอบนิรอรอลเน็ตเวิร์ก เพื่อแก้ไขปัญหาค่าความไม่แม่นยำของการประมาณค่าที่ทำให้ไม่สามารถนำแบบจำลองทางคณิตศาสตร์มาใช้ได้อย่างเหมาะสม และเพื่อยืนยันว่านิรอรอลเน็ตเวิร์กที่ออกแบบสามารถใช้ในการประมาณค่าตำแหน่งของสถานีเคลื่อนที่ได้จริง วิทยานิพนธ์ฉบับนี้จึงได้ออกแบบนิรอรอลเน็ตเวิร์กโดยใช้ภาษาบรรยายฮาร์ดแวร์ประเภทหนึ่งที่เรียกว่า ภาษาวีเอชดีแอล (Very high speed integrated circuit Hardware Description Language: VHDL) เนื่องจากในการออกแบบสามารถแก้ไข ทดสอบประสิทธิภาพ และนำมาสร้างเป็นอุปกรณ์ต้นแบบได้ โดยง่ายด้วยการดาวน์โหลดนิรอรอลเน็ตเวิร์กที่ออกแบบลงบนบอร์ดเอฟพีจีเอ (Field Programmable Gate Array: FPGA) ซึ่งจากผลการทดสอบในการจำลองแบบ และการทดสอบอุปกรณ์ต้นแบบที่ทดลองสร้างขึ้น แสดงให้เห็นว่านิรอรอลเน็ตเวิร์กที่ออกแบบสามารถประมาณค่าตำแหน่งของสถานีเคลื่อนที่ได้ถูกต้องตามข้อกำหนดของ FCC อย่างไรก็ดีตาม อุปกรณ์ต้นแบบดังกล่าวยังต้องมีการพัฒนาให้มีประสิทธิภาพมากขึ้นต่อไปในอนาคต

ภาควิชา.....วิศวกรรมไฟฟ้า.....ลายมือชื่อนิสิต.....ศรัณย์ เกียรติศิริโรจน์.....
 สาขาวิชา.....วิศวกรรมไฟฟ้า.....ลายมือชื่ออาจารย์ที่ปรึกษา.....
 ปีการศึกษา.....2548.....

4570556721: MAJOR ELECTRICAL ENGINEERING

KEY WORD: POSITION LOCATION / MOBILE STATION / SIGNAL STRENGTH (SS) / NEURAL NETWORK / VERY HIGH SPEED INTEGRATED CIRCUIT HARDWARE DESCRIPTION LANGUAGE (VHDL) / FIELD PROGRAMMABLE GATE ARRAY (FPGA)

SARAN KIETSIROJE: DESIGN OF NEURAL NETWORK FOR ESTIMATING LOCATION OF MOBILE STATION. ASSOC. PROF. DR. WATIT BENJAPOLKUL, 106 pp. ISBN 974-17-5702-6.

Position location services in mobile telephone systems have been receiving increased interest in the past decades. The applications are included in both commercial and governmental interests. At present, the Federal Communications Commission (FCC) has launched the specification for E-911 service. These requirements proposed that by October 2001, operators must be able to determine accurate location of mobile stations with accuracy within the radius of 125 meters to achieve at least 67 percent of all trial locations.

This thesis proposes a technique to design neural network to approximate the location of Mobile Station (MS) by using real data of the Signal Strength (SS) and the position of Base Stations (BSs) which communicate with the MS as inputs to train and test the neural network for solving the non-linear problem of approximation which is not suitable for using mathematical models. To prove that the designed neural network can be in practice, this thesis focuses on implementing the neural network using Very high speed integrated circuit Hardware Description Language (VHDL) because of its simplicity in debugging, testing the performance and implementing the prototype by downloading the designed neural network to Field Programmable Gate Array (FPGA). Simulation model and prototype testing show that the designed neural network has the ability to approximate the location of MS with accuracy in compliance with FCC. However, the implemented prototype in this thesis is still to be improved in the future.

Department..... Electrical Engineering..... Student's Signature *Saran Kietsiroje*
 Field of study..... Electrical Engineering..... Advisor's signature *Watit*
 Academic year..... 2005.....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี กระผมใคร่ขอกราบขอบพระคุณอย่างสูงสำหรับความช่วยเหลืออย่างดียิ่งของ รองศาสตราจารย์ ดร.วาทิต เบญจพลกุล อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งได้ให้คำแนะนำและข้อคิดเห็นต่าง ๆ รวมทั้งข้อมูลด้านวิชาการในการทำวิจัยมาโดยตลอด

ขอขอบคุณอาจารย์ทุกท่านที่ได้ให้ความรู้ในการศึกษาในการนำไปใช้เกี่ยวกับงานวิจัยนี้

ขอขอบคุณโครงการวิจัยร่วมระหว่างภาควิชาวิศวกรรมไฟฟ้า จุฬาลงกรณ์มหาวิทยาลัย และภาคเอกชนทางด้านการวิจัยและพัฒนา ที่ให้การสนับสนุนเกี่ยวกับการทำงานวิจัยนี้

ขอขอบคุณ พี่เจนวิทย์ จากศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC) สำหรับคำแนะนำดี ๆ และเทคนิคต่าง ๆ ในการทำงานวิจัยนี้

ขอขอบคุณ รุ่นพี่ รุ่นน้อง เพื่อน ๆ ทุกคน ทั้งที่ห้องปฏิบัติการวิจัยระบบโทรคมนาคม จุฬาลงกรณ์มหาวิทยาลัย และมหาวิทยาลัยสงขลานครินทร์ สำหรับความช่วยเหลือและกำลังใจในการทำวิจัย ทำให้ปัญหาต่าง ๆ ในการวิจัยผ่านไปได้อย่างราบรื่น

สุดท้ายขอกราบขอบพระคุณบิดามารดาและครอบครัว ที่เป็นกำลังใจและให้การสนับสนุนอย่างดียเยี่ยมแก่ผู้วิจัยมาโดยตลอดจนสำเร็จการศึกษา

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	ฅ
สารบัญภาพ.....	ญ
บทที่	
1 บทนำ.....	1
1.1 ความเป็นมาและแนวเหตุผล.....	1
1.2 ภาพรวมของงานวิจัยที่เกี่ยวข้อง.....	2
1.3 แนวทางของวิทยานิพนธ์.....	4
1.4 เป้าหมายและขอบเขตของวิทยานิพนธ์.....	5
1.5 ขั้นตอนและวิธีการดำเนินงาน.....	6
1.6 ประโยชน์ที่คาดว่าจะได้รับ.....	6
1.7เค้าโครงวิทยานิพนธ์.....	6
2 ความรู้พื้นฐาน.....	8
2.1 ทฤษฎีพื้นฐานของนิเวศอินเทอร์เน็ต.....	8
2.1.1 ส่วนประกอบของแบบจำลองนิเวศอินเทอร์เน็ต.....	8
2.1.2 ฟังก์ชันถ่ายโอนรูปแบบต่าง ๆ ที่ใช้กับแบบจำลองนิเวศอินเทอร์เน็ต.....	10
2.1.3 การทำงานโดยใช้แบบจำลองหลายชั้น.....	12
2.1.4 สถาปัตยกรรมโครงข่ายชนิดป้อนไปหน้าและการฝึกแบบแพร่กระจายย้อนกลับ.....	13
2.2 ข้อกำหนดของ FCC สำหรับบริการ E-911 ในระบบสื่อสารไร้สาย.....	15
3 นิเวศอินเทอร์เน็ตเพื่อประมาณค่าตำแหน่งของสถานีเคลื่อนที่ที่นำเสนอ.....	17
3.1 หลักการเบื้องต้นในการออกแบบสร้างอุปกรณ์ต้นแบบเพื่อประมาณค่าตำแหน่งของสถานีเคลื่อนที่ที่นำเสนอ.....	17

	หน้า
3.2 รูปแบบของนิรอลเน็ตเวิร์กเพื่อประมาณค่าตำแหน่งของสถานีเคลื่อนที่ที่ ถูกออกแบบในวิทยานิพนธ์ฉบับนี้.....	20
3.3 การออกแบบและสร้างอุปกรณ์ต้นแบบนิรอลเน็ตเวิร์กเพื่อประมาณค่า ตำแหน่งของสถานีเคลื่อนที่ที่นำเสนอ.....	21
3.3.1 การออกแบบนิรอลเน็ตเวิร์กเพื่อประมาณค่าตำแหน่งของสถานี เคลื่อนที่ด้วยภาษาวีเอชดีแอล.....	24
3.3.2 การสร้างอุปกรณ์ต้นแบบสำหรับประมาณค่าตำแหน่งของสถานี เคลื่อนที่.....	40
4 ผลการทดสอบ.....	44
4.1 ผลการทดสอบการรู้เข้าของนิรอลเน็ตเวิร์ก.....	45
4.2 ผลการทดสอบนิรอลเน็ตเวิร์กที่ออกแบบด้วยข้อมูลภาคสนาม.....	47
4.3 ผลการทดสอบนิรอลเน็ตเวิร์กที่ออกแบบด้วยข้อมูลจำลอง.....	80
4.4 ผลการทดสอบอุปกรณ์ต้นแบบเพื่อประมาณค่าตำแหน่งของสถานี เคลื่อนที่.....	86
5 บทสรุปและข้อเสนอแนะ.....	89
5.1 บทสรุป.....	89
5.2 ข้อเสนอแนะ.....	90
รายการอ้างอิง.....	92
ภาคผนวก.....	93
ประวัติผู้เขียนวิทยานิพนธ์.....	106

สารบัญตาราง

ตารางประกอบ	หน้า
ตารางที่ 3.1 การกำหนดขาของชิพเอฟพีจีเอเข้ากับสัญญาณนาฬิกา.....	43
ตารางที่ 3.2 การกำหนดขาของชิพเอฟพีจีเอเข้ากับสวิตช์ rst และสวิตช์ start.....	43
ตารางที่ 3.3 การกำหนดขาของชิพเอฟพีจีเอเข้ากับสวิตช์ lat_sel และสวิตช์ int_sel.....	43
ตารางที่ 3.4 การกำหนดขาของชิพเอฟพีจีเอเข้ากับตัวแสดงผลเจ็ดส่วน.....	43
ตารางที่ 4.1 รูปแบบของชุดข้อมูลภาคสนามสำหรับฝึกและทดสอบนิเวศน์เน็ตเวิร์ก.....	48



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

ภาพประกอบ		หน้า
รูปที่ 1.1	รูปแบบของนิรวัลเน็ตเวิร์กในงานวิจัยที่เกี่ยวข้อง.....	3
รูปที่ 1.2	รูปแบบของนิรวัลเน็ตเวิร์กในวิทยานิพนธ์.....	5
รูปที่ 2.1	โครงสร้างของนิรวัลเน็ตเวิร์กโดยทั่วไป 1 ชั้น.....	8
รูปที่ 2.2	ฟังก์ชันถ่ายโอนชั้นบันได.....	11
รูปที่ 2.3	ฟังก์ชันถ่ายโอนเชิงเส้น.....	11
รูปที่ 2.4	ฟังก์ชันถ่ายโอนลอการิทึมซิกมอยด์.....	11
รูปที่ 2.5	ฟังก์ชันถ่ายโอนไฮเพอร์โบลิกแทนเจนต์ซิกมอยด์.....	12
รูปที่ 2.6	แบบจำลองนิรวัลเน็ตเวิร์กหลายชั้น.....	12
รูปที่ 3.1	ขั้นตอนการออกแบบวงจรบนชิพเอฟพีจีเอโดยใช้ภาษาวีเอชดีแอล.....	18
รูปที่ 3.2	รูปแบบของนิรวัลเน็ตเวิร์กที่ใช้ในวิทยานิพนธ์ฉบับนี้.....	21
รูปที่ 3.3	ลักษณะการออกแบบวงจรโดยแบ่งเป็นวิธีข้อมูลและวิธีควบคุม.....	22
รูปที่ 3.4	โฟลตติงพอยนต์ 32 บิต.....	22
รูปที่ 3.5	จำนวนจริง -3.625 หลังถูกแปลงให้อยู่ในรูปโฟลตติงพอยนต์ 32 บิต.....	23
รูปที่ 3.6	คุณลักษณะของฟังก์ชันประมาณของฟังก์ชันถ่ายโอนลอการิทึมซิกมอยด์ เปรียบเทียบกับฟังก์ชันถ่ายโอนลอการิทึมซิกมอยด์.....	23
รูปที่ 3.7	ภาพรวมของนิรวัลเน็ตเวิร์กที่ออกแบบในวิทยานิพนธ์ฉบับนี้.....	24
รูปที่ 3.8	ขั้นตอนการทำงานร่วมกันของวงจรร้อยต่าง ๆ.....	25
รูปที่ 3.9	วิธีข้อมูลของวงจรเลือกค่าถ่วงน้ำหนักและค่าไบแอส.....	26
รูปที่ 3.10	วิธีควบคุมของวงจรเลือกค่าถ่วงน้ำหนักและค่าไบแอส.....	27
รูปที่ 3.11	วิธีข้อมูลของวงจรขั้นที่ 1.....	28
รูปที่ 3.12	วิธีควบคุมของวงจรขั้นที่ 1.....	29
รูปที่ 3.13	วิธีข้อมูลของวงจรขั้นที่ 2.....	30
รูปที่ 3.14	วิธีข้อมูลของวงจรขั้นที่ 3.....	31
รูปที่ 3.15	วิธีข้อมูลของวงจรขั้นที่ 4.....	32
รูปที่ 3.16	วิธีข้อมูลของวงจรขั้นที่ 5.....	33
รูปที่ 3.17	วิธีข้อมูลของวงจรขั้นที่ 6.....	34
รูปที่ 3.18	วิธีข้อมูลของวงจรขั้นที่ 7.....	35
รูปที่ 3.19	วิธีข้อมูลของวงจรขั้นที่ 8.....	36

ภาพประกอบ	หน้า
รูปที่ 3.20	วิธีข้อมูลของวงจรชั้นที่ 9..... 37
รูปที่ 3.21	วิธีข้อมูลของวงจรชั้นที่ 10..... 39
รูปที่ 3.22	บอร์ดเอฟพีจีเอรุ่น Discovery-III XC3S200..... 40
รูปที่ 3.23	รายละเอียดของบอร์ดเอฟพีจีเอรุ่น Discovery-III XC3S200..... 41
รูปที่ 3.24	ส่วนควบคุมและส่วนแสดงผลของอุปกรณ์ต้นแบบ..... 42
รูปที่ 4.1	ผลการทดสอบการลู่เข้าของนิรอลเน็ตเวิร์กที่มีจำนวนนิรอลในชั้นซ่อน เส้นไม่เท่ากัน..... 46
รูปที่ 4.2	ผลการทดสอบการลู่เข้าของนิรอลเน็ตเวิร์กที่มีค่าอัตราการเรียนรู้ไม่ เท่ากัน..... 46
รูปที่ 4.3	ตำแหน่งการเคลื่อนที่ของสถานีเคลื่อนที่ที่เหมาะสม..... 48
รูปที่ 4.4	ชุดข้อมูลสำหรับฝึกและทดสอบนิรอลเน็ตเวิร์กไม่อยู่ในบริเวณเดียวกัน..... 49
รูปที่ 4.5	ค่าความผิดพลาดของการประมาณค่าเมื่อชุดข้อมูลสำหรับฝึกและทดสอบ นิรอลเน็ตเวิร์กไม่อยู่ในบริเวณเดียวกัน..... 49
รูปที่ 4.6	ชุดข้อมูลสำหรับฝึกและทดสอบนิรอลเน็ตเวิร์กบางส่วนไม่อยู่ในบริเวณ เดียวกัน..... 50
รูปที่ 4.7	ค่าความผิดพลาดของการประมาณค่าเมื่อชุดข้อมูลสำหรับฝึกและทดสอบ นิรอลเน็ตเวิร์กบางส่วนไม่อยู่ในบริเวณเดียวกัน..... 51
รูปที่ 4.8	ชุดข้อมูลสำหรับฝึกและทดสอบนิรอลเน็ตเวิร์กทั้งหมดอยู่ในบริเวณ เดียวกัน..... 51
รูปที่ 4.9	ผลการทดสอบกรณีที่ 1 (ชุดข้อมูลทดสอบที่ 1 ถึง 10)..... 53
รูปที่ 4.10	ผลการทดสอบกรณีที่ 1 (ชุดข้อมูลทดสอบที่ 11 ถึง 20)..... 53
รูปที่ 4.11	ผลการทดสอบกรณีที่ 1 (ชุดข้อมูลทดสอบที่ 21 ถึง 30)..... 54
รูปที่ 4.12	ผลการทดสอบกรณีที่ 1 (ชุดข้อมูลทดสอบที่ 31 ถึง 40)..... 54
รูปที่ 4.13	ค่าความผิดพลาดของการประมาณค่าในกรณีที่ 1 โดยใช้นิรอลเน็ตเวิร์กที่ ออกแบบ..... 55
รูปที่ 4.14	ค่าความผิดพลาดของการประมาณค่าในกรณีที่ 1 โดยใช้ NNTool..... 55
รูปที่ 4.15	ค่าความผิดพลาดที่เกิดขึ้นของชุดข้อมูลทดสอบแต่ละชุดจากการประมาณ ค่ากรณีที่ 1 ของนิรอลเน็ตเวิร์กที่ออกแบบเปรียบเทียบกับค่าการประมาณค่า โดยใช้ NNTool..... 56

ภาพประกอบ	หน้า
รูปที่ 4.16 ผลการทดสอบกรณีที่ 2 (ชุดข้อมูลทดสอบที่ 1 ถึง 10).....	57
รูปที่ 4.17 ผลการทดสอบกรณีที่ 2 (ชุดข้อมูลทดสอบที่ 11 ถึง 20).....	57
รูปที่ 4.18 ผลการทดสอบกรณีที่ 2 (ชุดข้อมูลทดสอบที่ 21 ถึง 30).....	58
รูปที่ 4.19 ผลการทดสอบกรณีที่ 2 (ชุดข้อมูลทดสอบที่ 31 ถึง 40).....	58
รูปที่ 4.20 ค่าความผิดพลาดของการประมาณค่าในกรณีที่ 2 โดยใช้นิเวศเน็ตเวิร์กที่ ออกแบบ.....	59
รูปที่ 4.21 ค่าความผิดพลาดของการประมาณค่าในกรณีที่ 2 โดยใช้ NNTool.....	59
รูปที่ 4.22 ค่าความผิดพลาดที่เกิดขึ้นของชุดข้อมูลทดสอบแต่ละชุดจากการประมาณ ค่ากรณีที่ 2 ของนิเวศเน็ตเวิร์กที่ออกแบบเปรียบเทียบกับค่าการประมาณค่า โดยใช้ NNTool.....	60
รูปที่ 4.23 ผลการทดสอบกรณีที่ 3 (ชุดข้อมูลทดสอบที่ 1 ถึง 10).....	61
รูปที่ 4.24 ผลการทดสอบกรณีที่ 3 (ชุดข้อมูลทดสอบที่ 11 ถึง 20).....	61
รูปที่ 4.25 ผลการทดสอบกรณีที่ 3 (ชุดข้อมูลทดสอบที่ 21 ถึง 30).....	62
รูปที่ 4.26 ผลการทดสอบกรณีที่ 3 (ชุดข้อมูลทดสอบที่ 31 ถึง 40).....	62
รูปที่ 4.27 ค่าความผิดพลาดของการประมาณค่าในกรณีที่ 3 โดยใช้นิเวศเน็ตเวิร์กที่ ออกแบบ.....	63
รูปที่ 4.28 ค่าความผิดพลาดของการประมาณค่าในกรณีที่ 3 โดยใช้ NNTool.....	63
รูปที่ 4.29 ค่าความผิดพลาดที่เกิดขึ้นของชุดข้อมูลทดสอบแต่ละชุดจากการประมาณ ค่ากรณีที่ 3 ของนิเวศเน็ตเวิร์กที่ออกแบบเปรียบเทียบกับค่าการประมาณค่า โดยใช้ NNTool.....	64
รูปที่ 4.30 ชุดข้อมูลภาคสนามสำหรับฝึกนิเวศเน็ตเวิร์กถูกเรียงกันอย่างเป็นลำดับ จากจุดหนึ่งไปยังอีกจุดหนึ่ง.....	65
รูปที่ 4.31 ชุดข้อมูลภาคสนามสำหรับฝึกนิเวศเน็ตเวิร์กที่ถูกสลับตำแหน่งของชุด ข้อมูลจากหน้าไปหลัง.....	65
รูปที่ 4.32 ผลการทดสอบกรณีที่ 3 (ชุดข้อมูลทดสอบที่ 1 ถึง 10) เมื่อสลับตำแหน่ง ของชุดข้อมูลภาคสนามสำหรับฝึกนิเวศเน็ตเวิร์กจากหน้าไปหลัง.....	66
รูปที่ 4.33 ผลการทดสอบกรณีที่ 3 (ชุดข้อมูลทดสอบที่ 11 ถึง 20) เมื่อสลับตำแหน่ง ของชุดข้อมูลภาคสนามสำหรับฝึกนิเวศเน็ตเวิร์กจากหน้าไปหลัง.....	66

ภาพประกอบ	หน้า
รูปที่ 4.34 ผลการทดสอบกรณีที่ 3 (ชุดข้อมูลทดสอบที่ 21 ถึง 30) เมื่อสลับตำแหน่งของชุดข้อมูลภาคสนามสำหรับฝึกนิรอลเน็ตเวิร์กจากหน้าไปหลัง.....	67
รูปที่ 4.35 ผลการทดสอบกรณีที่ 3 (ชุดข้อมูลทดสอบที่ 31 ถึง 40) เมื่อสลับตำแหน่งของชุดข้อมูลภาคสนามสำหรับฝึกนิรอลเน็ตเวิร์กจากหน้าไปหลัง.....	67
รูปที่ 4.36 ค่าความผิดพลาดของการประมาณค่าในกรณีที่ 3 โดยใช้นิรอลเน็ตเวิร์กที่ออกแบบ (สลับตำแหน่งของชุดข้อมูลภาคสนามสำหรับฝึกนิรอลเน็ตเวิร์กจากหน้าไปหลัง).....	68
รูปที่ 4.37 ค่าความผิดพลาดของการประมาณค่าในกรณีที่ 3 โดยใช้ NNTool (สลับตำแหน่งของชุดข้อมูลภาคสนามสำหรับฝึกนิรอลเน็ตเวิร์กจากหน้าไปหลัง).....	68
รูปที่ 4.38 ค่าความผิดพลาดที่เกิดขึ้นของชุดข้อมูลทดสอบแต่ละชุดจากการประมาณค่ากรณีที่ 3 ของนิรอลเน็ตเวิร์กที่ออกแบบเปรียบเทียบกับค่าการประมาณค่าโดยใช้ NNTool (สลับตำแหน่งของชุดข้อมูลภาคสนามสำหรับฝึกนิรอลเน็ตเวิร์กจากหน้าไปหลัง).....	69
รูปที่ 4.39 ชุดข้อมูลภาคสนามสำหรับฝึกนิรอลเน็ตเวิร์กที่ถูกสลับตำแหน่งอย่างไม่มีระเบียบ.....	69
รูปที่ 4.40 ผลการทดสอบกรณีที่ 3 (ชุดข้อมูลทดสอบที่ 1 ถึง 10) เมื่อสลับตำแหน่งของชุดข้อมูลภาคสนามสำหรับฝึกนิรอลเน็ตเวิร์กอย่างไม่มีระเบียบ.....	71
รูปที่ 4.41 ผลการทดสอบกรณีที่ 3 (ชุดข้อมูลทดสอบที่ 11 ถึง 20) เมื่อสลับตำแหน่งของชุดข้อมูลภาคสนามสำหรับฝึกนิรอลเน็ตเวิร์กอย่างไม่มีระเบียบ.....	71
รูปที่ 4.42 ผลการทดสอบกรณีที่ 3 (ชุดข้อมูลทดสอบที่ 21 ถึง 30) เมื่อสลับตำแหน่งของชุดข้อมูลภาคสนามสำหรับฝึกนิรอลเน็ตเวิร์กอย่างไม่มีระเบียบ.....	72
รูปที่ 4.43 ผลการทดสอบกรณีที่ 3 (ชุดข้อมูลทดสอบที่ 31 ถึง 40) เมื่อสลับตำแหน่งของชุดข้อมูลภาคสนามสำหรับฝึกนิรอลเน็ตเวิร์กอย่างไม่มีระเบียบ.....	72
รูปที่ 4.44 ค่าความผิดพลาดของการประมาณค่าในกรณีที่ 3 โดยใช้นิรอลเน็ตเวิร์กที่ออกแบบ (สลับตำแหน่งของชุดข้อมูลภาคสนามสำหรับฝึกนิรอลเน็ตเวิร์กอย่างไม่มีระเบียบ).....	73

ภาพประกอบ	หน้า	
รูปที่ 4.45	ค่าความผิดพลาดของการประมาณค่าในกรณีที่ 3 โดยใช้ NNTool (สลับตำแหน่งของชุดข้อมูลภาคสนามสำหรับฝึกนิรอลเน็ตเวิร์กอย่างไม่มีระเบียบ).....	73
รูปที่ 4.46	ค่าความผิดพลาดที่เกิดขึ้นของชุดข้อมูลทดสอบแต่ละชุดจากการประมาณค่ากรณีที่ 3 ของนิรอลเน็ตเวิร์กที่ออกแบบเปรียบเทียบกับค่าการประมาณค่าโดยใช้ NNTool (สลับตำแหน่งของชุดข้อมูลภาคสนามสำหรับฝึกนิรอลเน็ตเวิร์กอย่างไม่มีระเบียบ).....	74
รูปที่ 4.47	ค่าความผิดพลาดที่เกิดขึ้นจากการประมาณค่าของนิรอลเน็ตเวิร์กที่ออกแบบเมื่อมีการสลับตำแหน่งของชุดข้อมูลภาคสนามสำหรับฝึกนิรอลเน็ตเวิร์กในลักษณะต่าง ๆ.....	75
รูปที่ 4.48	ค่าความผิดพลาดที่เกิดขึ้นจากการประมาณค่าของ NNTool เมื่อมีการสลับตำแหน่งของชุดข้อมูลภาคสนามสำหรับฝึกนิรอลเน็ตเวิร์กในลักษณะต่าง ๆ.....	76
รูปที่ 4.49	ผลการทดสอบการลู่เข้าของนิรอลเน็ตเวิร์กในกรณีที่ 3 เมื่อเปลี่ยนวิธีการฝึกนิรอลเน็ตเวิร์กเป็นแบบปรับค่าครั้งเดียว.....	77
รูปที่ 4.50	ค่าความผิดพลาดของการประมาณค่าตำแหน่งของสถานีเคลื่อนที่หลังผ่านการฝึกด้วยวิธีการฝึกแบบปรับค่าครั้งเดียวจำนวน 800 รอบ.....	78
รูปที่ 4.51	ค่าความผิดพลาดของการประมาณค่าตำแหน่งของสถานีเคลื่อนที่หลังผ่านการฝึกด้วยวิธีการฝึกแบบปรับค่าครั้งเดียวจำนวน 100 รอบ.....	79
รูปที่ 4.52	ระบบสถานีฐานและสถานีเคลื่อนที่ที่ใช้ในการจำลองแบบ.....	80
รูปที่ 4.53	ผลการทดสอบกรณีชุดข้อมูลฝึก 50 ชุด และข้อมูลทดสอบ 40 ชุด.....	82
รูปที่ 4.54	ผลการทดสอบกรณีชุดข้อมูลฝึก 75 ชุด และข้อมูลทดสอบ 40 ชุด.....	83
รูปที่ 4.55	ผลการทดสอบกรณีชุดข้อมูลฝึก 100 ชุด และข้อมูลทดสอบ 40 ชุด.....	83
รูปที่ 4.56	ค่าความผิดพลาดกรณีชุดข้อมูลจำลองสำหรับฝึก 100 ชุด และสำหรับทดสอบ 40 ชุด เมื่อเปลี่ยนวิธีการฝึกนิรอลเน็ตเวิร์กเป็นแบบปรับค่าครั้งเดียว.....	85
รูปที่ 4.57	ค่าความผิดพลาดเมื่อเปลี่ยนวิธีการฝึกนิรอลเน็ตเวิร์กเป็นแบบปรับค่าครั้งเดียว และเพิ่มจำนวนนิรอลเน็ตเวิร์กในชั้นซ่อนเร้นเป็น 50 นิรอลเน็ตเวิร์ก.....	85

ภาพประกอบ		หน้า
รูปที่ 4.58	ค่าความผิดพลาดเมื่อเปลี่ยนวิธีการฝึกนิเวรอลเนตเวิร์กเป็นแบบปรับค่าครั้งเดียว และเพิ่มจำนวนนิเวรอนในชั้นซ่อนเร้นเป็น 100 นิเวรอน.....	86
รูปที่ 4.59	การแสดงผลเมื่อกดปุ่ม rst.....	87
รูปที่ 4.60	การแสดงผลค่าจำนวนเต็มของตำแหน่งละติจูด.....	88
รูปที่ 4.61	การแสดงผลค่าจำนวนทศนิยมของตำแหน่งละติจูด.....	88
รูปที่ 4.62	การแสดงผลค่าจำนวนเต็มของตำแหน่งลองจิจูด.....	88
รูปที่ 4.63	การแสดงผลค่าจำนวนทศนิยมของตำแหน่งลองจิจูด.....	88



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1

บทนำ

ในบทนี้จะกล่าวถึงความเป็นมาและแนวเหตุผลของวิทยานิพนธ์ฉบับนี้ ภาพรวมของงานวิจัยที่เกี่ยวข้อง แนวทางของวิทยานิพนธ์ เป้าหมายและขอบเขตของวิทยานิพนธ์ ขั้นตอนและวิธีการดำเนินงาน ประโยชน์ที่ได้รับ และเค้าโครงวิทยานิพนธ์

1.1 ความเป็นมาและแนวเหตุผล

ในยุคปัจจุบัน ระบบสื่อสารเคลื่อนที่ได้เข้ามามีบทบาทอย่างมากในชีวิตประจำวันของมนุษย์ ทำให้เกิดการคิดค้นและปรับปรุงเทคโนโลยีต่าง ๆ ตลอดเวลาเพื่อรองรับความต้องการที่เพิ่มขึ้นเรื่อย ๆ ได้ หนึ่งในเทคโนโลยีดังกล่าว ได้แก่ เทคโนโลยีการประมาณค่าตำแหน่งของสถานีเคลื่อนที่ ซึ่งถูกคิดค้นขึ้นเพื่ออำนวยความสะดวก โดยเฉพาะอย่างยิ่งทางด้านความปลอดภัยและการนำมาใช้สำหรับกรณีฉุกเฉิน ยกตัวอย่างเช่น การค้นหาบุคคลสูญหายจากตำแหน่งของสถานีเคลื่อนที่ที่บุคคลนั้นพกพา หรือการรายงานสถานที่เกิดอุบัติเหตุจากสถานีเคลื่อนที่ เป็นต้น จนกระทั่งได้มีการกำหนดเป็นมาตรฐานสำหรับบริการฉุกเฉิน 911 (E-911) ของหน่วยงาน U. S. Federal Communications Commission (FCC) ในหัวข้อที่ 94-102 เมื่อวันที่ 15 กันยายน ค.ศ. 1999 ซึ่งกำหนดไว้ว่า สำหรับระบบสื่อสารเคลื่อนที่ซีดีเอ็มเอ (Code Division Multiple Access: CDMA) หรือจีเอสเอ็ม (Global System for Mobile communications: GSM) ระบบการประมาณค่าตำแหน่งของสถานีเคลื่อนที่ จะต้องมีความสามารถในการรายงานค่าพิกัดตำแหน่งให้มีความถูกต้องถึง 67% ของจำนวนครั้งของการประมาณค่าในระยะความผิดพลาด 125 เมตร [1, 2]

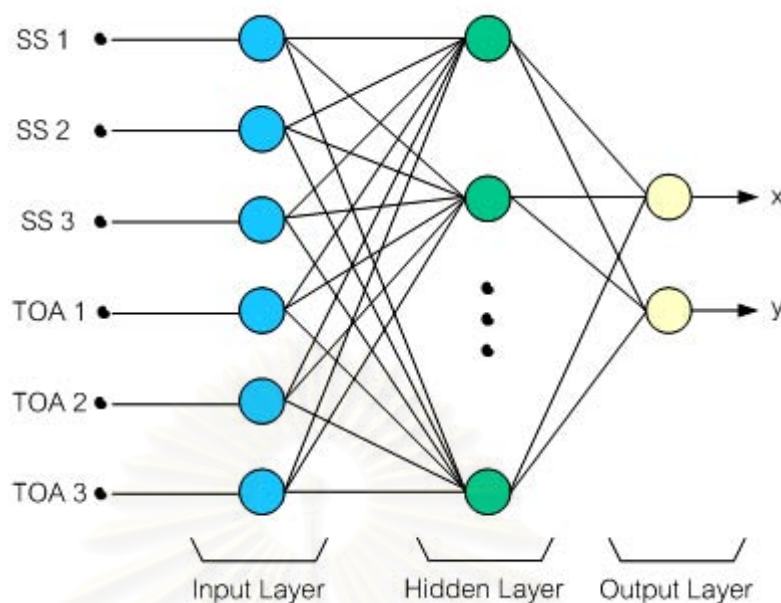
ในการประมาณค่าตำแหน่งของสถานีเคลื่อนที่ จำเป็นต้องมีการวัดสัญญาณเพื่อนำมาเป็นข้อมูลในการคำนวณหาพิกัดตำแหน่งของสถานีเคลื่อนที่ ซึ่งมีอยู่หลายวิธี [2, 3] ยกตัวอย่างเช่น การวัดค่าความแรงของสัญญาณ (Signal Strength Method: SS Method), การวัดมุมการมาถึงของสัญญาณ (Angle of Arrival Method: AOA Method), การวัดเวลาการมาถึงของสัญญาณ (Time of Arrival Method: TOA Method) เป็นต้น อย่างไรก็ตาม วิธีดังกล่าวยังมีความยุ่งยากในการคำนวณเนื่องจากต้องแก้สมการที่ไม่เป็นเชิงเส้น ประกอบกับค่าความแรงของสัญญาณ, มุมการมาถึงของสัญญาณ รวมทั้งเวลาที่วัดได้อาจมีความผิดพลาดเนื่องจากผลกระทบจากสภาพภูมิประเทศ, เฟดดิ้งพหุวิถี (Multipath fading), การอยู่นอกแนวสายตา (Non Line of Sight: NLOS) ฯลฯ จึงทำให้ผลลัพธ์ของการประมาณค่าผิดพลาดพอสมควร

จากปัญหาดังที่ได้กล่าวมา ระบบที่ใช้ในการประมาณค่าตำแหน่งของสถานีเคลื่อนที่จึงต้องมีความสามารถในการเลียนแบบหรือรู้จำสภาพภูมิประเทศ เพื่อให้สามารถทำนายค่าพิกัดตำแหน่งของสถานีเคลื่อนที่ได้ได้อย่างถูกต้อง ซึ่งวิธีที่ทำให้ระบบสามารถรู้จำได้นั้นมีหลายวิธี โดยที่วิธีแต่ละวิธีนั้นมีข้อเด่นที่แตกต่างกัน เช่น Hidden Markov เหมาะสำหรับระบบที่ต้องการรู้จำเพื่อแยกแยะข้อมูล แต่การประมาณค่าตำแหน่งของสถานีเคลื่อนที่นั้นเป็นระบบที่ต้องการรู้จำเพื่อประมาณค่า ซึ่งวิธีที่ได้รับความนิยมและแพร่หลาย ได้แก่ การใช้โครงข่ายประสาทเทียม (Neural Network) ซึ่งเป็นเทคนิคการเรียนรู้รูปแบบแบบหนึ่ง ทำงานโดยเลียนแบบระบบประสาทของมนุษย์ หลักการทำงานประกอบด้วยหน่วยประมวลผลย่อยหลาย ๆ หน่วยทำงานเชื่อมต่อกัน แต่ละหน่วยสามารถปรับค่าพารามิเตอร์ประจำหน่วยได้จากกระบวนการเรียนรู้โดยใช้ตัวอย่างแบบจำลองที่ผ่านการฝึกฝนจะสามารถนำไปใช้แก้ปัญหาจริงได้ โครงข่ายประสาทเทียมมีคุณสมบัติที่ดีและเหมาะสมหลายประการ เช่น สามารถสร้างแบบจำลองสำหรับงานที่ซับซ้อนมาก ๆ ได้ สามารถทำงานกับข้อมูลที่มีความผิดพลาด หรือข้อมูลที่ไม่มีโครงสร้างได้ สามารถนำไปใช้กับปัญหาในการประมาณค่าต่าง ๆ และมีความสามารถในการแก้ปัญหาแบบไม่เป็นเชิงเส้น โดยใช้เวลาในการคำนวณน้อยกว่าการนำอัลกอริทึมอื่น ๆ ที่ใช้แบบจำลองทางคณิตศาสตร์มาคำนวณ

1.2 ภาพรวมของงานวิจัยที่เกี่ยวข้อง

ในหัวข้อนี้จะกล่าวถึงภาพรวมของงานวิจัยที่เกี่ยวข้อง ได้แก่ งานวิจัยเรื่องการประมาณค่าตำแหน่งของสถานีโดยใช้โครงข่ายประสาทเทียม [4]

งานวิจัยเรื่องการประมาณค่าตำแหน่งของสถานีเคลื่อนที่ด้วยโครงข่ายประสาทเทียมนี้ ผู้วิจัยใช้โครงข่ายประสาทเทียมมาประยุกต์ใช้กับการประมาณค่าตำแหน่งของสถานีเคลื่อนที่ โดยโครงข่ายประสาทเทียมที่ใช้ในงานวิจัยมีสถาปัตยกรรมแบบโครงข่ายชนิดป้อนไปหน้าและมีการฝึกแบบแพร่กระจายย้อนกลับ (Feedforward Network – Backpropagation Training Architecture) ที่มีชั้นข้อมูลเข้า (Input Layer) ประกอบด้วย ค่าข้อมูลความแรงของสัญญาณที่สถานีฐาน 3 สถานีรับได้จากสถานีเคลื่อนที่ที่พิจารณา (SS1, SS2, SS3) และค่าเวลาการมาถึงของสัญญาณที่สถานีฐาน 3 สถานีรับได้จากสถานีเคลื่อนที่ (TOA1, TOA2, TOA3) เป็นข้อมูลเข้า (Input) ที่จะฝึกโครงข่ายประสาทเทียม มีชั้นซ่อนเร้น (Hidden Layer) จำนวน 1 ชั้น ซึ่งในงานวิจัยนี้ได้ทดสอบเพื่อหาจำนวนนิวรอน (Neuron) ในชั้นซ่อนเร้นที่เหมาะสม และชั้นข้อมูลออก (Output Layer) โดยโครงข่ายประสาทเทียมจะให้ค่าผลลัพธ์ออกมาเป็นค่าตำแหน่งละติจูด – ลองจิจูดของสถานีเคลื่อนที่ที่พิจารณา (x, y) ดังแสดงในรูปที่ 1.1



รูปที่ 1.1 รูปแบบของนิเวศเน็ตเวิร์กในงานวิจัยที่เกี่ยวข้อง

การจำลองระบบการประมาณค่าตำแหน่งของสถานีเคลื่อนที่โดยใช้นิเวศเน็ตเวิร์กในงานวิจัยนี้มีสมมุติฐานที่ใช้ในแบบจำลองดังนี้

1. ระบบที่พิจารณาเป็นระบบสื่อสารเคลื่อนที่ CDMA หรือ GSM
2. ใช้สายอากาศแบบรอบทิศทางในระนาบเดียว (Omnidirection antenna)
3. สถานีเคลื่อนที่ที่อยู่ภายในบริเวณสามเหลี่ยมที่สร้างขึ้นจากสถานีฐานทั้ง 3 สถานี
4. สถานีเคลื่อนที่ที่ต้องอยู่ในตำแหน่งที่นิ่ง หรือเคลื่อนไหวน้อยมากในขณะที่พิจารณาและไม่มี การส่งต่อ (Hand-off) ในขณะนั้น
5. ไม่มีการควบคุมกำลัง (Power Control) ในขณะใช้งานระบบการประมาณค่าตำแหน่ง
6. จำลองระบบและชุดข้อมูลสำหรับฝึกและทดสอบนิเวศเน็ตเวิร์ก และทดสอบนิเวศเน็ตเวิร์ก โดยจำลองค่าข้อมูลของความแรงสัญญาณ, เวลาการมาถึงของสัญญาณ และตำแหน่งของสถานีเคลื่อนที่ ซึ่งการจำลองข้อมูลของเวลาจะเริ่มจากการจำลองอย่างง่ายโดยไม่คำนึงถึงผลของ เฟดดิ้งพหุวิถีและการอยู่นอกแนวสายตา

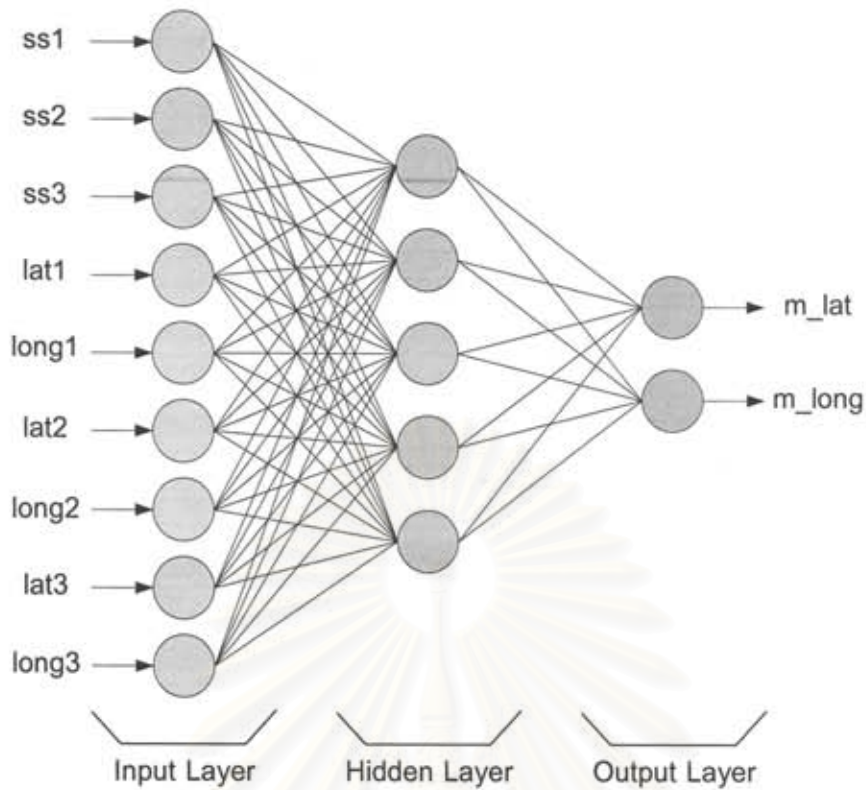
งานวิจัยข้างต้นได้จำลองการทำงานบนโปรแกรมสำเร็จรูป Matlab ซึ่งผลการทดสอบระบบในหลาย ๆ กรณี แสดงให้เห็นว่านิเวศเน็ตเวิร์กมีความสามารถในการประมาณค่าตำแหน่งของสถานีเคลื่อนที่ที่ต้องทำตามข้อกำหนดของ FCC อย่างไรก็ตาม ในงานวิจัยข้างต้นได้ใช้การจำลองค่าความแรงของสัญญาณและเวลาการมาถึงของสัญญาณ เป็นข้อมูลเพื่อฝึกและทดสอบ

นิเวศน์เน็ตเวิร์ก แต่ในความเป็นจริง ข้อมูลภาคสนามหรือข้อมูลจริงที่ได้มาจากบริษัทเอกชนที่ให้บริการโทรศัพท์เคลื่อนที่ส่วนใหญ่มักไม่มีค่าเวลาการมาถึงของสัญญาณ ทำให้ไม่สามารถนำแบบจำลองในงานวิจัยนี้มาใช้กับข้อมูลจริงได้ ดังนั้นวิทยานิพนธ์นี้ซึ่งมุ่งเน้นที่จะออกแบบนิเวศน์เน็ตเวิร์กที่สามารถประมาณค่าตำแหน่งของสถานีเคลื่อนที่ได้จริง จึงเสนอให้แก้ไขและปรับปรุงระบบ รวมทั้งออกแบบระบบการประมาณค่าตำแหน่งของสถานีเคลื่อนที่ใหม่ เพื่อให้เหมาะสมกับข้อมูลจริงและสามารถนำไปพัฒนาเป็นเครื่องต้นแบบการประมาณค่าตำแหน่งของสถานีเคลื่อนที่ด้วยนิเวศน์เน็ตเวิร์กได้

1.3 แนวทางของวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้ เสนอการออกแบบนิเวศน์เน็ตเวิร์กเพื่อประมาณค่าตำแหน่งของสถานีเคลื่อนที่โดยใช้ภาษาบรรยายฮาร์ดแวร์ (Hardware Description Language: HDL) ประเภทหนึ่งเรียกว่า วีเอสดีแอล (VHSIC (Very High Speed Integrated Circuits) Hardware Description Language: VHDL) ในการออกแบบ เนื่องจากภาษาวีเอสดีแอลสามารถแก้ไข, ทดสอบประสิทธิภาพ และนำมาพัฒนาสร้างเป็นอุปกรณ์ต้นแบบเพื่อประมาณค่าตำแหน่งของสถานีเคลื่อนที่ได้โดยการดาวน์โหลดวงจรที่ออกแบบลงบนชิพหรือบอร์ดเอฟพีจีเอ (Field Programmable Gate Array: FPGA) โดยในวิทยานิพนธ์นี้ได้นำทั้งข้อมูลจำลองและข้อมูลภาคสนาม (ข้อมูลจริง) มาใช้ฝึกและทดสอบประสิทธิภาพของนิเวศน์เน็ตเวิร์กเพื่อยืนยันว่านิเวศน์เน็ตเวิร์กที่ออกแบบสามารถประมาณค่าตำแหน่งของสถานีเคลื่อนที่ได้จริง โดยใช้ข้อมูลความแรงของสัญญาณที่สถานีฐาน 3 สถานี รับได้จากสถานีเคลื่อนที่ที่พิจารณา (ss1, ss2, ss3) และตำแหน่งละติจูด - ลองจิจูดของสถานีฐานทั้ง 3 สถานี (lat1, long1, lat2, long2, lat3, long3) เป็นข้อมูลเข้าของนิเวศน์เน็ตเวิร์ก เพื่อให้ค่าผลลัพธ์เป็นตำแหน่งละติจูด - ลองจิจูดของสถานีเคลื่อนที่ที่พิจารณา (m_lat, m_long) ดังแสดงในรูปที่ 1.2

หลังจากออกแบบและทดสอบการทำงานของนิเวศน์เน็ตเวิร์กแล้ว ในวิทยานิพนธ์ฉบับนี้ได้เสนอให้ทดลองนำวงจรที่ออกแบบดาวน์โหลดลงบนบอร์ดเอฟพีจีเอรุ่น Discovery-III XC3S200 ของบริษัท เอเพค อินสตรูเมนต์ จำกัด เพื่อแสดงให้เห็นว่านิเวศน์เน็ตเวิร์กที่ออกแบบสามารถนำมาพัฒนาเป็นอุปกรณ์ต้นแบบสำหรับประมาณค่าตำแหน่งของสถานีเคลื่อนที่ได้จริง



รูปที่ 1.2 รูปแบบของนิรอลเน็ตเวิร์กในวิทยานิพนธ์

1.4 เป้าหมายและขอบเขตของวิทยานิพนธ์

1. ออกแบบและจำลองการทำงานของนิรอลเน็ตเวิร์กโดยใช้ค่าความแรงของสัญญาณที่สถานีฐานรับได้จากสถานีเคลื่อนที่ และตำแหน่งละติจูด - ลองจิจูดของสถานีฐาน เป็นข้อมูลสำหรับฝึกและทดสอบนิรอลเน็ตเวิร์ก เพื่อแสดงความสามารถในการประมาณค่าตำแหน่งของสถานีเคลื่อนที่
2. นิรอลเน็ตเวิร์กที่ออกแบบต้องสามารถประมาณค่าพิกัดของสถานีเคลื่อนที่ได้อย่างมีประสิทธิภาพ โดยเปรียบเทียบกับมาตรฐาน E-911 ของหน่วยงาน FCC
3. นิรอลเน็ตเวิร์กที่ออกแบบต้องสามารถนำมาพัฒนาเป็นอุปกรณ์ต้นแบบเพื่อนำมาใช้ในการใช้งานจริงได้

1.5 ขั้นตอนและวิธีการดำเนินงาน

1. ศึกษาวิธีการประมาณค่าตำแหน่งของสถานีเคลื่อนที่โดยใช้นิรवलเน็ตเวิร์กจากงานวิจัยที่เกี่ยวข้อง
2. ศึกษาวิธีการออกแบบระบบดิจิทัลด้วยภาษาวีเอสดีแอล
3. คัดเลือกและจัดรูปแบบข้อมูลภาคสนาม (ข้อมูลจริง) ให้สามารถนำมาใช้สำหรับฝึกและทดสอบนิรवलเน็ตเวิร์กได้
4. ออกแบบนิรवलเน็ตเวิร์กเพื่อประมาณค่าตำแหน่งของสถานีเคลื่อนที่ด้วยภาษาวีเอสดีแอล
5. ทดสอบและวิเคราะห์ผลการทำงานของนิรवलเน็ตเวิร์กที่ออกแบบ
6. นำนิรवलเน็ตเวิร์กที่ออกแบบดาวนโหลดบนบอร์ดเอฟพีจีเอเพื่อพัฒนาเป็นอุปกรณ์ต้นแบบ
7. สรุปผลการทดสอบและเขียนวิทยานิพนธ์

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1. เข้าใจแนวความคิดและความรู้พื้นฐานรวมทั้งวิธีการนำนิรवलเน็ตเวิร์กมาประยุกต์ใช้ในการประมาณค่าพิกัดตำแหน่งของสถานีเคลื่อนที่
2. สามารถนำความรู้ที่ได้มาออกแบบและสร้างระบบประมาณค่าตำแหน่งของสถานีเคลื่อนที่ที่สามารถนำมาใช้งานกับข้อมูลจริงได้
3. ระบบที่สร้างขึ้นสามารถนำมาปรับปรุงและพัฒนาเพื่อสร้างเป็นเครื่องต้นแบบสำหรับใช้งานจริงในอนาคตต่อไป

1.7 คำโครงวิทยานิพนธ์

สำหรับเนื้อหาของวิทยานิพนธ์ฉบับนี้แบ่งรายละเอียดออกเป็น 5 บทดังต่อไปนี้

บทที่ 1 บทนำ ซึ่งได้กล่าวถึง ความเป็นมาและแนวเหตุผลของวิทยานิพนธ์ฉบับนี้ ภาพรวมของงานวิจัยที่เกี่ยวข้อง แนวทางของวิทยานิพนธ์ เป้าหมายและขอบเขตของวิทยานิพนธ์ ขั้นตอนและวิธีการดำเนินงาน ประโยชน์ที่ได้รับ และคำโครงวิทยานิพนธ์

บทที่ 2 ความรู้พื้นฐาน เนื้อหาในบทนี้จะกล่าวถึง ทฤษฎีพื้นฐานของนิรवलเน็ตเวิร์ก ซึ่งประกอบด้วย ส่วนประกอบของแบบจำลองนิรवलเน็ตเวิร์ก ฟังก์ชันถ่ายโอนรูปแบบต่าง ๆ ที่ใช้กับแบบจำลองนิรवलเน็ตเวิร์ก การทำงานโดยใช้แบบจำลองหลายชั้น และสถาปัตยกรรมโครงข่ายชนิดป้อนไปหน้าและการฝึกแบบแพร่กระจายย้อนกลับ รวมทั้งรายละเอียดพอสังเขปเกี่ยวกับ

ข้อกำหนดของ FCC สำหรับบริการ E-911 ในระบบสื่อสารไร้สาย (The FCC Ruling on Wireless E-911)

บทที่ 3 การออกแบบนิรลเนตเวิร์กที่นำเสนอ เนื้อหาในบทนี้กล่าวถึง เทคนิคและวิธีการในการออกแบบนิรลเนตเวิร์กและทดลองสร้างอุปกรณ์ต้นแบบสำหรับประมาณค่าตำแหน่งของสถานีเคลื่อนที่

บทที่ 4 ผลการทดสอบ ในบทนี้จะแสดงผลการทดสอบและการวิเคราะห์ประสิทธิภาพการจำลองแบบของนิรลเนตเวิร์กที่ได้นำเสนอในบทที่ 3 ในการประมาณค่าตำแหน่งของสถานีเคลื่อนที่ โดยข้อมูลที่ใช้ฝึกและทดสอบนิรลเนตเวิร์กมีทั้งข้อมูลสมมุติที่จำลองขึ้น และข้อมูลจริงที่ได้มาจากการเก็บข้อมูลภาคสนาม พร้อมทั้งเปรียบเทียบกับการประมาณค่าตำแหน่งของสถานีเคลื่อนที่ด้วยนิรลเนตเวิร์กโดยใช้กรรมวิธีอื่น รวมถึงการทดสอบอุปกรณ์ต้นแบบเพื่อประมาณค่าตำแหน่งของสถานีเคลื่อนที่ที่ทดลองสร้างขึ้น

บทที่ 5 บทสรุป ซึ่งเป็นการสรุปผลการวิจัยของวิทยานิพนธ์และข้อเสนอแนะเพิ่มเติมสำหรับงานวิจัยในอนาคต



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

ความรู้พื้นฐาน

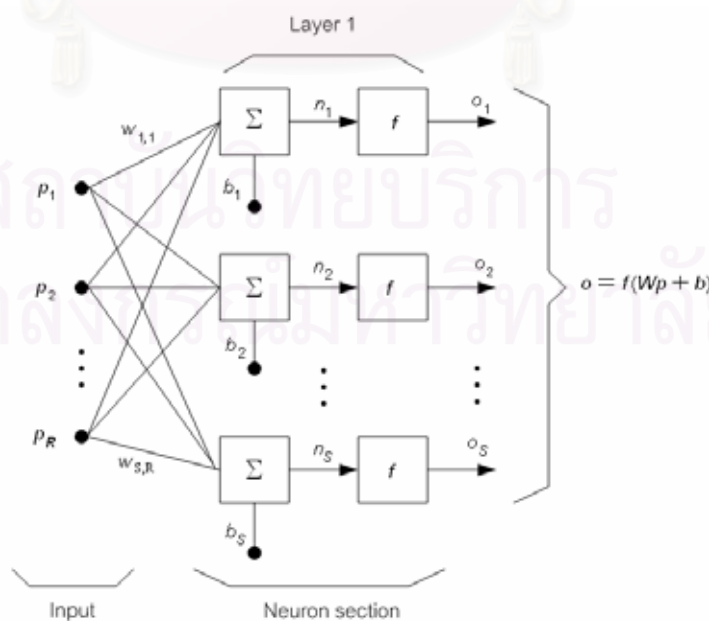
ในบทนี้จะกล่าวถึง ความรู้พื้นฐานต่าง ๆ ที่ต้องนำมาใช้ในวิทยานิพนธ์ ซึ่งประกอบด้วย ทฤษฎีพื้นฐานของนิวรอลเน็ตเวิร์ก และรายละเอียดของซอฟต์แวร์เกี่ยวกับข้อกำหนดของ FCC สำหรับ บริการ E-911 ในระบบสื่อสารไร้สาย

2.1 ทฤษฎีพื้นฐานของนิวรอลเน็ตเวิร์ก

นิวรอลเน็ตเวิร์กเป็นวิธีการจำลองระบบให้เสมือนเป็นเซลล์ประสาทของมนุษย์ โดยมี จุดมุ่งหมายที่จะสร้างวิธีการคำนวณให้ระบบสามารถรู้จักการประมาณค่า หรือจดจำคุณลักษณะ ของสิ่งใดสิ่งหนึ่ง เนื่องจากไม่สามารถนำแบบจำลองทางคณิตศาสตร์ใด ๆ มาใช้อธิบาย หรือแก้ไข ปัญหาได้อย่างมีประสิทธิภาพ อีกทั้งยังช่วยลดความยุ่งยากซับซ้อนในการใช้แบบจำลองทาง คณิตศาสตร์มาคำนวณ นอกจากนี้ นิวรอลเน็ตเวิร์กยังมีความสามารถและเป็นที่ยอมรับในการ แก้ปัญหาระบบที่ไม่เป็นเชิงเส้นอีกประการหนึ่งด้วย

2.1.1 ส่วนประกอบของแบบจำลองนิวรอลเน็ตเวิร์ก [5, 6]

แบบจำลองนิวรอลเน็ตเวิร์กประกอบด้วยส่วนประกอบต่าง ๆ ดังแสดงในรูปที่ 2.1



รูปที่ 2.1 โครงสร้างของนิวรอลเน็ตเวิร์กโดยทั่วไป 1 ชั้น

จากรูปที่ 2.1 โครงสร้างของนิวรอนเน็ตเวิร์กโดยทั่วไป 1 ชั้น ประกอบด้วยส่วนประกอบต่าง ๆ ดังนี้

- 1) ข้อมูลเข้า (Input: \vec{p}): เป็นค่าข้อมูลที่ป้อนเข้าสู่ระบบ
- 2) ค่าถ่วงน้ำหนัก (Weight: \vec{w}): เป็นค่าที่นำมาคูณเข้ากับข้อมูลเข้าเพื่อเป็นการกำหนดค่าความสำคัญให้กับข้อมูลแต่ละตัว จากนั้นจะส่งผ่านเข้าสู่ส่วนนิวรอนต่อไป ในรูปที่ 2.1 นิยามให้ $w_{s,r}$ เป็นค่าถ่วงน้ำหนักในชั้นข้อมูลเข้า (Input layer weight) จากข้อมูลเข้าตำแหน่งที่ r ไปยังการคำนวณตำแหน่งที่ s
- 3) นิวรอน (Neuron): เป็นหน่วยที่เปรียบเสมือนเซลล์ประสาทแต่ละเซลล์ของมนุษย์ที่ใช้ในการคำนวณเพื่อประมาณค่า แยกแยะ หรือรู้จำ โดยในนิวรอนแต่ละนิวรอนประกอบไปด้วยส่วนประกอบย่อย ได้แก่
 - ค่าไบแอส (Bias: b): เป็นค่าคงที่ที่บวกเพิ่มเข้าไปในนิวรอนแต่ละตัวเพื่อเป็นค่าออฟเซต (Offset) ในการคำนวณ
 - ฟังก์ชันการรวม (Summation: Σ): เป็นการรวมค่าข้อมูลเข้าที่ถูกคูณด้วยค่าถ่วงน้ำหนักกับค่าไบแอสเข้าด้วยกัน ซึ่งจะได้ผลลัพธ์ออกมาเป็นข้อมูลเข้ารวม
 - ข้อมูลเข้ารวม (Net Input: n): เป็นข้อมูลที่ได้จากฟังก์ชันการรวมก่อนที่จะผ่านฟังก์ชันถ่ายโอนซึ่งจากรูปที่ 2.1 จะได้ค่าข้อมูลเข้ารวมเป็นไปตามสมการที่ (2.1)

$$n = wp + b \quad (2.1)$$

- ฟังก์ชันถ่ายโอน (Transfer Function: f): เป็นฟังก์ชันการคำนวณที่นิวรอนใช้แบ่งข้อมูลเข้าออกเป็นกลุ่ม ๆ เพื่อจับคู่ (Mapping) ข้อมูลเข้าไปสู่ค่าเป้าหมายที่ป้อนให้ โดยฟังก์ชันถ่ายโอนที่นิยมใช้กันมาก ได้แก่ ฟังก์ชันถ่ายโอนขั้นบันได (Hard Limit Transfer Function) ฟังก์ชันถ่ายโอนเชิงเส้น (Linear Transfer Function) และฟังก์ชันถ่ายโอนซิกมอยด์ (Sigmoid Transfer Function)
- 4) ข้อมูลออก (Output: \vec{o}): เป็นผลลัพธ์ที่ได้จากการประมาณค่าผ่านแบบจำลอง 1 ชั้น ซึ่งมีลักษณะดังสมการที่ (2.2)

$$\vec{o} = f(\vec{w}\vec{p} + \vec{b}) \quad (2.2)$$

- 5) ค่าเป้าหมาย (Target: t^-): เป็นค่าที่ต้องการให้นิวรอลเน็ตเวิร์กประมาณค่าให้ใกล้เคียงมากที่สุด ยกตัวอย่างเช่น ในการใช้งานในการรู้จำตัวอักษร ค่าของเป้าหมายจะเป็น ตัวอักษรที่ต้องการให้คอมพิวเตอร์รู้จำ

รูปที่ 2.1 แสดงชุดข้อมูลป้อนเข้าจำนวน R ข้อมูลป้อนเข้าสู่นิวรอน S ตัว โดยผ่านค่าถ่วงน้ำหนัก W ซึ่งจะเป็นการคูณแบบสเกลาร์ระหว่างข้อมูลเข้ากับค่าถ่วงน้ำหนัก จากนั้นข้อมูลที่ได้จะถูกนำไปรวมกับค่าไบแอสของนิวรอนแต่ละตัว และผ่านฟังก์ชันถ่ายโอนออกมาเป็นค่าข้อมูลออกเป็นลักษณะของแบบจำลองนิวรอลเน็ตเวิร์ก 1 ชั้น

2.1.2 ฟังก์ชันถ่ายโอนรูปแบบต่าง ๆ ที่ใช้กับแบบจำลองนิวรอลเน็ตเวิร์ก [5, 6]

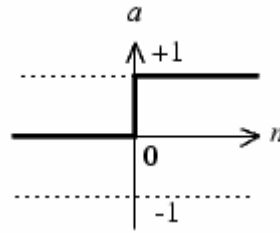
ฟังก์ชันถ่ายโอนในแบบจำลองนิวรอลเน็ตเวิร์กเป็นสิ่งสำคัญที่จะบอกความสามารถในการประมาณ แยกแยะ หรือรู้จำ โดยการนำมาใช้ให้เหมาะสมกับปัญหาจะทำให้ผลลัพธ์ที่ได้จากการคำนวณถูกจัดกลุ่มอยู่ด้วยกันอย่างเหมาะสม ส่งผลให้ความถูกต้องในการทำงานมีสูง เนื่องจากค่าที่ได้จากการคำนวณด้วยนิวรอลเน็ตเวิร์กจะมีค่าลู่เข้าสู่ค่าเป้าหมายที่ตั้งไว้ ดังนั้นการเลือกฟังก์ชันถ่ายโอนมาใช้กับแบบจำลองที่สร้างขึ้นให้มีความเหมาะสมกับปัญหาจึงเป็นปัจจัยหลักที่ต้องพิจารณาในการออกแบบนิวรอลเน็ตเวิร์ก ก่อนที่จะฝึกและทดสอบนิวรอลเน็ตเวิร์กต่อไป โดยทั่วไปได้มีการกำหนดฟังก์ชันถ่ายโอนหลัก ๆ ที่ใช้กับงานต่าง ๆ ดังนี้

- 1) ฟังก์ชันถ่ายโอนขั้นบันได (Hard Limit Transfer Function)

เป็นฟังก์ชันถ่ายโอนที่ใช้กับแบบจำลองอย่างง่าย เช่น ปัญหาที่ต้องการเพียงแบบจำลองขั้นเดียวในการทำงาน การแบ่งข้อมูลของฟังก์ชันถ่ายโอนชนิดนี้จะแบ่งข้อมูลออกเป็น 2 ส่วน คือ มากกว่าค่า 0 และน้อยกว่าค่า 0 ดังแสดงในสมการที่ (2.3)

$$\text{hard lim}(x) = \begin{cases} 0, & x \leq 0 \\ 1, & x > 0 \end{cases} \quad (2.3)$$

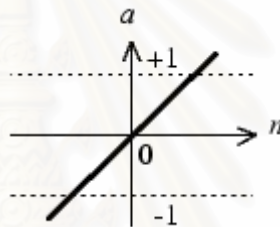
ฟังก์ชันถ่ายโอนขั้นบันไดแสดงได้ดังรูปที่ 2.2 และเนื่องจากการแบ่งข้อมูลของฟังก์ชันถ่ายโอนชนิดนี้เป็นดังสมการที่ (2.3) ดังนั้นข้อมูลที่ฟังก์ชันถ่ายโอนชนิดนี้สามารถแบ่งแยกได้จึงต้องเป็นข้อมูลที่สามารถแบ่งแยกได้เป็นเชิงเส้นชัดเจน (linearly separable)



รูปที่ 2.2 ฟังก์ชันถ่ายโอนขั้นบันได

2) ฟังก์ชันถ่ายโอนเชิงเส้น (Linear Transfer Function)

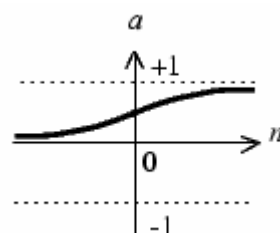
เป็นฟังก์ชันถ่ายโอนที่ใช้กับการแก้ปัญหาที่มีลักษณะเป็นเชิงเส้น ซึ่งการใช้ฟังก์ชันถ่ายโอนเชิงเส้นนี้จะให้ผลลัพธ์ในการแยกแยะข้อมูลที่ดีเมื่อชุดข้อมูลที่ป้อนเข้าสามารถแบ่งแยกหรือมีความสัมพันธ์กันในลักษณะเชิงเส้น ฟังก์ชันถ่ายโอนเชิงเส้นมีลักษณะดังแสดงในรูปที่ 2.3



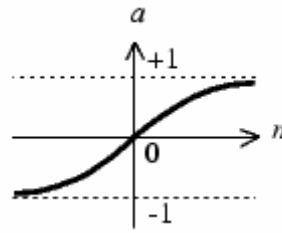
รูปที่ 2.3 ฟังก์ชันถ่ายโอนเชิงเส้น

3) ฟังก์ชันถ่ายโอนซิกมอยด์ (Sigmoid Transfer Function)

ฟังก์ชันถ่ายโอนชนิดนี้เหมาะกับการแก้ปัญหาแบบไม่เป็นเชิงเส้น และเป็นฟังก์ชันถ่ายโอนที่นิยมใช้กับการฝึกโครงข่ายประสาทเทียมแบบแพร่กระจายย้อนกลับ (Back Propagation Training) เนื่องจากเป็นฟังก์ชันที่สามารถหาอนุพันธ์ได้ ฟังก์ชันถ่ายโอนชนิดนี้สามารถแบ่งได้เป็น 2 ลักษณะ ได้แก่ ฟังก์ชันถ่ายโอนลอการิทึมซิกมอยด์ (Logarithm Sigmoid Transfer Function) และฟังก์ชันถ่ายโอนไฮเพอร์โบลิกแทนเจนต์ซิกมอยด์ (Hyperbolic Tangent Sigmoid Function) ดังแสดงในรูปที่ 2.4 และ 2.5 ตามลำดับ



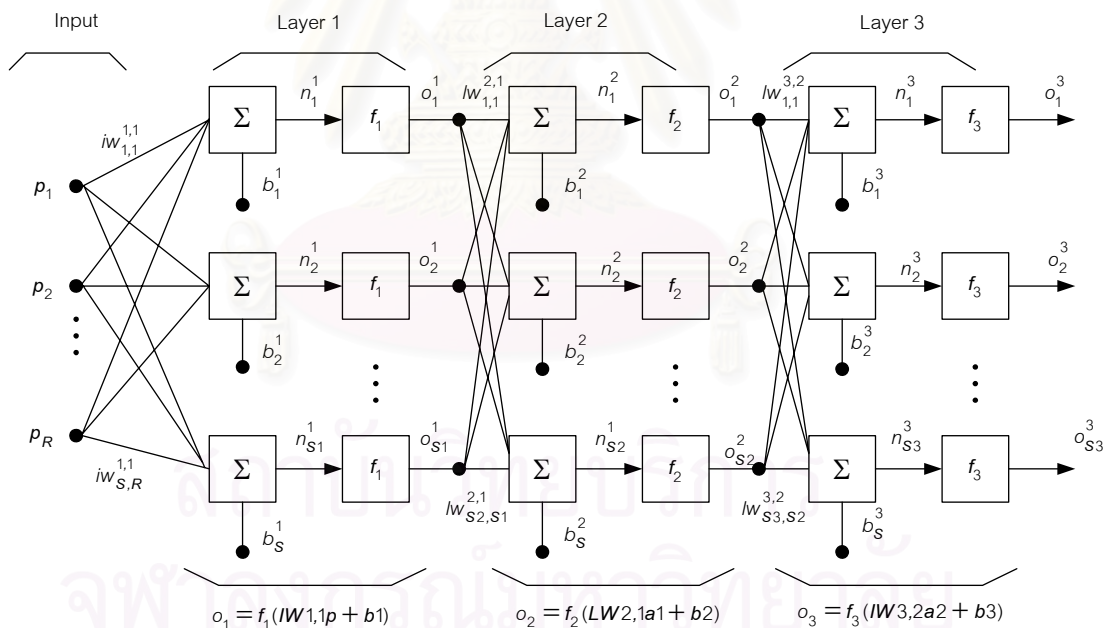
รูปที่ 2.4 ฟังก์ชันถ่ายโอนลอการิทึมซิกมอยด์



รูปที่ 2.5 ฟังก์ชันถ่ายโอนไฮเพอร์โบลิกแทนเจนต์ซิกมอยด์

2.1.3 การทำงานโดยใช้แบบจำลองหลายชั้น [5, 6]

ในการใช้นิวรอลเน็ตเวิร์กเพื่อแก้ปัญหาที่มีความซับซ้อนและต้องการความถูกต้องสูง การใช้งานแบบจำลองนิวรอลเน็ตเวิร์กอย่างง่ายในรูปที่ 2.1 ไม่สามารถคำนวณและให้ค่าผลลัพธ์ที่ต้องการได้ ดังนั้นจึงจำเป็นต้องออกแบบนิวรอลเน็ตเวิร์กให้มีจำนวนชั้นของแบบจำลอง และจำนวนนิวรอนที่มากขึ้นเพื่อให้ระบบมีความสามารถในการคำนวณให้ค่าผลลัพธ์เป็นไปตามที่ต้องการได้ แบบจำลองหลายชั้นจะเกิดจากการต่อแบบจำลองชั้นเดียวมากกว่าหนึ่งชุดเข้าด้วยกัน ดังแสดงในรูปที่ 2.6



รูปที่ 2.6 แบบจำลองนิวรอลเน็ตเวิร์กหลายชั้น

วิธีการต่อแบบจำลองหลายชั้นนั้นจะทำให้สามารถเลือกใช้ฟังก์ชันถ่ายโอนที่แตกต่างกันในแต่ละชั้นเพื่อให้มีความสามารถในการแก้ปัญหาที่เหมาะสม โดยเฉพาะอย่างยิ่งความสามารถในการแก้ปัญหาแบบไม่เป็นเชิงเส้น

ผลลัพธ์ของแบบจำลองนิวรอลเน็ตเวิร์กหลายชั้นจะมีค่าเป็นไปตามสมการที่ (2.4)

$$O_3 = f_3(LW_{3,2}f_2(LW_{2,1}(IW_{1,1}p + b_1) + b_2) + b_3) \quad (2.4)$$

เมื่อ IW คือเมตริกซ์ของค่าน้ำหนักที่ถ่ายเชื่อมโยงขาเข้าสู่ระบบ (Input Weight Matrix)

LW คือเมตริกซ์ของค่าน้ำหนักที่ถ่ายเชื่อมโยงระหว่างชั้นของแบบจำลอง (Layer Weight Matrix)

f_i เป็นค่าฟังก์ชันถ่ายโอนของแบบจำลองชั้นที่ i , $i = 1, 2$, และ 3

2.1.4 สถาปัตยกรรมโครงข่ายชนิดป้อนไปหน้าและการฝึกแบบแพร่กระจายย้อนกลับ (Feedforward Network - Backpropagation Training Architecture) [5, 6]

สถาปัตยกรรมโครงข่ายชนิดป้อนไปหน้ามีลักษณะดังรูปที่ 2.6 ซึ่งจะเป็นการต่อนิรอรลเน็ตเวิร์กอย่างง่ายในรูปที่ 2.1 หลาย ๆ ชั้นเข้าด้วยกัน เพื่อให้ทำให้นิรอรลเน็ตเวิร์กที่ได้มีความสามารถในการแก้ปัญหาที่ซับซ้อนมากขึ้น นิรอรลเน็ตเวิร์กชนิดป้อนไปหน้าจะประกอบไปด้วยชั้นข้อมูลเข้า (Input Layer) 1 ชั้น, ชั้นซ่อนเร้น (Hidden Layer) อย่างน้อย 1 ชั้น และชั้นข้อมูลออก (Output Layer) 1 ชั้น การต่อกันของนิรอรลในแต่ละชั้นจะเป็นการต่อแบบไปข้างหน้า ไม่มีการต่อย้อนกลับมา ดังนั้นการต่อนิรอรลเน็ตเวิร์กแบบนี้จึงถูกเรียกว่า “โครงข่ายชนิดป้อนไปหน้า”

ในการฝึกนิรอรลเน็ตเวิร์กชนิดนี้จะนิยมใช้วิธีการฝึกแบบแพร่กระจายย้อนกลับ กล่าวคือมีการป้อนกลับของค่าข้อมูลที่ได้จากการประมาณค่าเพื่อปรับค่าค่าถ่วงน้ำหนักและค่าไบแอสจากนิรอรลในชั้นข้อมูลออกกลับไปสู่นิรอรลในชั้นข้อมูลเข้า ในลักษณะของการมองย้อนกลับจากแบบจำลองชั้นปลายมาสู่ชั้นต้น โดยการฝึกนิรอรลเน็ตเวิร์กสามารถแบ่งได้เป็น 2 ประเภทตามลักษณะของการปรับค่าถ่วงน้ำหนักและค่าไบแอสดังนี้

1. การฝึกนิรอรลเน็ตเวิร์กแบบปรับค่าตลอดเวลา (Adaptive Training)

การฝึกนิรอรลเน็ตเวิร์กแบบปรับค่าตลอดเวลา เป็นการปรับค่าถ่วงน้ำหนักและค่าไบแอสให้กับนิรอรลเน็ตเวิร์กทุก ๆ ครั้งที่มีการป้อนข้อมูลเข้าแต่ละตัวเข้าสู่ระบบ

2. การฝึกนิรอรลเน็ตเวิร์กแบบปรับค่าครั้งเดียว (Batch Training)

การฝึกนิรอรลเน็ตเวิร์กแบบปรับค่าครั้งเดียว เป็นการปรับค่าถ่วงน้ำหนักและค่าไบแอสให้กับนิรอรลเน็ตเวิร์กแต่ละครั้งหลังจากการป้อนชุดข้อมูลเข้าให้กับนิรอรลเน็ตเวิร์กทั้งหมดทุกชุดแล้ว

พิจารณานิรอรลเน็ตเวิร์กในรูปที่ 2.6 การฝึกนิรอรลเน็ตเวิร์กแบบแพร่กระจายย้อนกลับที่ใช้ในวิทยานิพนธ์ฉบับนี้สามารถเขียนเป็นอัลกอริทึมได้ดังนี้

- 1) กำหนดค่าเริ่มต้นของค่าถ่วงน้ำหนักและค่าไบแอสทั้งหมดในนิวรอนเดวิด
- 2) พิจารณาที่นิวรอนที่ j ใด ๆ ค่าผลลัพธ์เริ่มต้นของนิวรอนที่ j (o_j) จะเป็นดังสมการที่ (2.5)

$$o_j = f(\sum w_{ji}o_i + b_j) \quad (2.5)$$

เมื่อ w_{ji} เป็นค่าถ่วงน้ำหนักจากค่าข้อมูลเข้า o_i มาที่นิวรอนที่ j
 b_j เป็นค่าไบแอสของนิวรอนที่ j
 นิวรอนที่ i คือนิวรอนในชั้นที่อยู่ด้านซ้ายมือของนิวรอนที่ j

- 3) ฝึกนิวรอนเดวิดแบบปรับค่าตลอดเวลา โดยเริ่มพิจารณาจากนิวรอนในชั้นข้อมูลออกมายังชั้นซ่อนเร้นและปรับค่าถ่วงน้ำหนักและค่าไบแอสตามสมการที่ (2.6) และ (2.7)

$$w_{ji}(t+1) = w_{ji}(t) + \Delta w_{ji} \quad (2.6)$$

$$b_j(t+1) = b_j(t) + \Delta b_j \quad (2.7)$$

เมื่อ $w_{ji}(t)$ คือค่าถ่วงน้ำหนักจากค่าข้อมูลเข้าที่ i ไปยังนิวรอนที่ j ในการวนซ้ำรอบที่ t
 $b_j(t)$ คือค่าไบแอสของนิวรอนที่ j ในการวนซ้ำรอบที่ t
 Δw_{ji} คือค่าส่วนปรับของค่าถ่วงน้ำหนักจากนิวรอนที่ i ไปยังนิวรอนที่ j
 Δb_j คือค่าส่วนปรับของค่าไบแอสของนิวรอนที่ j

สมการที่ (2.6) และ (2.7) เป็นสมการวนซ้ำที่จะปรับค่าถ่วงน้ำหนักและค่าไบแอสสำหรับทุก ๆ นิวรอน ในการคำนวณจากชั้นข้อมูลออกกลับเข้าสู่ชั้นข้อมูลเข้า ส่วนปรับของค่าถ่วงน้ำหนักและค่าส่วนปรับของค่าไบแอส เป็นไปตามสมการที่ (2.8) และ (2.9)

$$\Delta w_{ji} = \eta \delta_j o_i \quad (2.8)$$

$$\Delta b_j = \eta \delta_j \quad (2.9)$$

โดยที่ δ_j มีค่าตามสมการที่ (2.10)

$$\delta_j = \begin{cases} f'_j(\text{net}_j)(t_j - o_j) & \text{if neuron } j \text{ is an output neuron} \\ f'_j(\text{net}_j) \sum_k \delta_k w_{kj} & \text{if neuron } j \text{ is a hidden neuron} \end{cases} \quad (2.10)$$

เมื่อ η คือค่าอัตราการเรียนรู้ (Learning rate) ปกติมีค่าอยู่ระหว่าง 0 ถึง 1
 o_i คือค่าผลลัพธ์ที่นิวรอนเน็ตเวิร์กทำนายได้ที่นิวรอนที่ i
 o_j คือค่าผลลัพธ์ที่นิวรอนเน็ตเวิร์กทำนายได้ที่นิวรอนที่ j
 t_j คือค่าเป้าหมายที่ส่งเข้าที่นิวรอนที่ j
 δ_j คือค่าเกรเดียนต์ของค่าความผิดพลาด (Error gradient) ที่นิวรอนที่ j
 δ_k คือค่าเกรเดียนต์ของค่าความผิดพลาดที่นิวรอนที่ k ไปยังนิวรอนที่ j
 นิวรอนที่ k คือนิวรอนในชั้นที่อยู่ด้านขวามือของนิวรอนที่ j
 net_j คือข้อมูลก่อนผ่านฟังก์ชันถ่ายโอนที่นิวรอนที่ j
 $f'_j(net_j)$ คือค่าอนุพันธ์ของฟังก์ชันถ่ายโอนของข้อมูลเข้ารวมที่นิวรอนที่ j เทียบกับค่า net_j

- 4) วนซ้ำขั้นใหม่โดยเริ่มที่ขั้นตอนที่ 2) จนกว่าจะได้ค่าผลลัพธ์ที่ทำให้ค่าความผิดพลาดมีค่าเข้าสู่ค่าที่ต้องการ

2.2 ข้อกำหนดของ FCC สำหรับบริการ E-911 ในระบบสื่อสารไร้สาย (The FCC Ruling on Wireless E-911) [1, 2]

สำหรับการประมาณค่าตำแหน่งของสถานีเคลื่อนที่ที่มีความถูกต้องของการประมาณค่าเป็นไปตามข้อกำหนดของ FCC สำหรับบริการ E-911 ในระบบสื่อสารไร้สาย ซึ่งมีรายละเอียดโดยสังเขปดังต่อไปนี้

- 1) ระบบสื่อสารไร้สายจะต้องมีการทำงานที่สามารถประสานงานกับระบบข่ายสายได้อย่างไม่มีปัญหา ในเรื่องของการจัดเส้นทางในการส่งข้อมูลจากสถานีฐานไปยังศูนย์ 911 หรือพีเอสเอพี (Public Safety Answering Point: PSAP) ได้อย่างลงตัว และต้องมีความสามารถในการส่งข้อมูลที่เพียงพอที่พีเอสเอพีจะติดต่อกลับไปยังผู้เรียกใช้บริการนี้ได้ ซึ่ง FCC ได้กำหนดระยะเวลาจากวันที่เริ่มใช้ข้อกำหนดนี้ไปเป็นเวลา 18 เดือน และเรียกเป็นข้อกำหนดระยะที่ 1
- 2) ความถูกต้องของการประมาณค่าพิกัดตำแหน่งของผู้เรียกใช้บริการต้องมีค่าอยู่ในระยะ 125 เมตร ใน 67 % ของจำนวนครั้งของการประมาณค่าภายในระยะเวลา 5 ปีแรกของการเริ่มใช้ข้อกำหนด และค่าความถูกต้องนี้จะต้องมีรายละเอียดเพียงพอที่จะทำให้พีเอสเอพีสามารถระบุพิกัดตำแหน่งของสถานีเคลื่อนที่ได้อย่างเจาะจงมากกว่าการระบุเป็นตำแหน่งของถนน หรือ สถานีที่กว้าง ๆ และเรียกเป็นข้อกำหนดระยะที่ 2
- 3) บริการ E-911 สำหรับระบบสื่อสารไร้สายจะต้องครอบคลุมถึงสถานีเคลื่อนที่ที่ถูกยกเลิกการใช้บริการไปแล้ว เช่น กรณีแจ้งการสูญหาย หรือถูกจารกรรม เป็นต้น

- 4) บริการ E-911 สำหรับระบบสื่อสารไร้สายจะต้องรองรับระบบการทำงานที่หลากหลาย เช่น ระบบเซลลูลาร์, ระบบพีซีเอสแบนด์กว้าง (Broadband Public Communications System: Broadband PCS), ระบบโทรศัพท์เคลื่อนที่มาตรฐานจีเอสเอ็ม (Global System for Mobile Communications: GSM) และระบบโทรศัพท์เคลื่อนที่มาตรฐานซีดีเอ็มเอ (Code Division Multiple Access: CDMA) เป็นต้น แต่จะไม่รวมถึงระบบเคลื่อนที่ผ่านดาวเทียม (Mobile Satellite System)
- 5) ข้อกำหนดของ FCC ทั้งในระยะเวลาที่ 1 และ ระยะเวลาที่ 2 เริ่มใช้งานตั้งแต่วันที่ 1 ตุลาคม ค.ศ. 1996



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

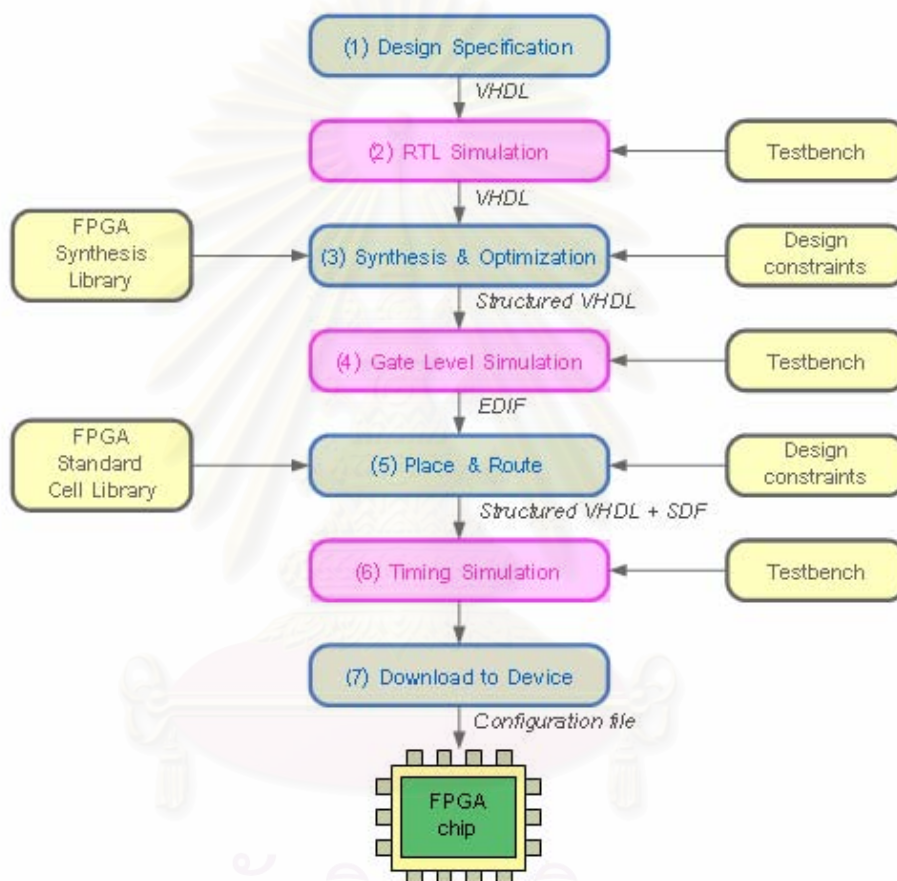
นิเวศน์เน็ตเวิร์กเพื่อประมาณค่าตำแหน่งของสถานีเคลื่อนที่ที่นำเสนอ

ในบทนี้จะกล่าวถึง เทคนิคและวิธีการออกแบบนิเวศน์เน็ตเวิร์กและทดลองสร้างอุปกรณ์ต้นแบบสำหรับประมาณค่าตำแหน่งของสถานีเคลื่อนที่ที่นำเสนอ

3.1 หลักการเบื้องต้นในการออกแบบสร้างอุปกรณ์ต้นแบบเพื่อประมาณค่าตำแหน่งของสถานีเคลื่อนที่ที่นำเสนอ [7]

จากเดิมการออกแบบสร้างอุปกรณ์ต่าง ๆ จะเป็นการออกแบบโดยใช้การวาดวงจร (Capture Schematic) โดยใช้โปรแกรมช่วยในการวาด (Schematic entry tools) โดยใช้ไอซีลอจิก เช่น ไอซีตระกูล 74xxx บัฟเฟอร์ รวมทั้งไอซีหน่วยความจำต่าง ๆ เชื่อมต่อกันเข้าเป็นวงจรตามฟังก์ชันการทำงาน ซึ่งวิธีการออกแบบค่อนข้างยาก และต้องใช้เวลามากในการออกแบบระบบจำลองการทำงาน (Simulation) ตลอดจนการแก้ไขความถูกต้องของอุปกรณ์ซึ่งทำได้ค่อนข้างยาก รวมทั้งใช้พื้นที่ของแผงวงจรพิมพ์ (Printed Circuit Board) ที่ใหญ่ ทำให้เป็นอุปสรรคในการออกแบบวงจรที่มีความซับซ้อนสูง ภาระเหล่านี้จึงตกเป็นหน้าที่ของไมโครคอนโทรลเลอร์ (Microcontroller) ไมโครโปรเซสเซอร์ (Microprocessor) หรือไอซีประมวลผลสัญญาณดิจิทัล (Digital Signal Processing) แต่ไอซีสำหรับประมวลผลเหล่านี้ถูกออกแบบมาสำหรับการทำงานตามชุดคำสั่งที่ผู้ใช้งานออกแบบ ไม่ได้ออกแบบมาสำหรับวงจรใดวงจรหนึ่งโดยเฉพาะ จึงมีข้อจำกัดด้านความเร็วในการทำงาน ดังนั้นจึงไม่เหมาะกับงานที่ต้องการความละเอียดในการคำนวณสูงหรืองานที่ต้องการเวลาในการประมวลผลสั้น แต่ในการออกแบบปัจจุบันได้มีกระบวนการออกแบบอีกรูปแบบหนึ่งที่มีประสิทธิภาพสูง รวดเร็ว และไม่ยึดติดกับเทคโนโลยีที่ใช้ออกแบบ (Technology Independent) ได้แก่ กระบวนการออกแบบที่มีลักษณะการออกแบบจากบนลงล่าง (Top-down Design) ซึ่งใช้ภาษาบรรยายฮาร์ดแวร์ (Hardware Description Language: HDL) ในการออกแบบ จำลองการทำงาน สังเคราะห์วงจร (Synthesis) ในรูปแบบของเทคโนโลยีที่เราต้องการ และสามารถพัฒนาเป็นอุปกรณ์ฮาร์ดแวร์ได้โดยดาวน์โหลดวงจรที่ออกแบบลงบนชิพหรือบอร์ดเอฟพีจีเอ (Field Programmable Gate Array: FPGA) ดังนั้นการออกแบบสามารถทำได้โดยง่ายและมีความสะดวกรวดเร็วมากยิ่งขึ้น นอกจากนี้ การออกแบบสร้างอุปกรณ์บนเอฟพีจีเอยังสามารถแก้ปัญหาเรื่องความเร็วในการประมวลผล รวมทั้งขนาดของวงจรทั้งหมดที่ถูกออกแบบบนแผงวงจรพิมพ์จะมีขนาดเล็ก เนื่องจากวงจรดิจิทัลทั้งหมดถูกรวมอยู่ในไอซีตัวเดียว ทำให้ไม่มีปัญหาเรื่องการออกแบบแผงวงจรพิมพ์ที่ซับซ้อน ลดสายสัญญาณ และลดสัญญาณรบกวน

ภาษาวีเอชดีแอล (VHSIC (Very High Speed Integrated Circuit) Hardware Description Language: VHDL) เป็นภาษาบรรยายฮาร์ดแวร์ประเภทหนึ่ง มีความสามารถในการบรรยายพฤติกรรมของวงจรที่ต้องการออกแบบ สามารถออกแบบให้มีพอร์ตเข้าออก (I/O port) มากเท่าที่จำนวนขาเอพพีจีเอรองรับ และวงจรทั้งหมดสามารถบรรจุในไอซีเพียงตัวเดียว ซึ่งปัจจุบันสามารถรองรับได้ถึงหลักล้านลอจิกเกต ในขณะที่ราคาที่ไม่สูงเมื่อเทียบกับความสามารถของวงจรที่ได้ การออกแบบวงจรบนเอพพีจีเอโดยใช้ภาษาวีเอชดีแอลมีขั้นตอนดังแสดงในรูปที่ 3.1



รูปที่ 3.1 ขั้นตอนการออกแบบวงจรบนชิพเอพพีจีเอโดยใช้ภาษาวีเอชดีแอล

ขั้นตอนที่ 1 การสร้างข้อกำหนดของการออกแบบ (Design Specification)

เป็นขั้นตอนการสร้างข้อกำหนดต่าง ๆ ของวงจร เช่น วงจรทำงานที่ความถี่เท่าไร ฟังก์ชันการทำงานมีอะไรบ้าง ซึ่งเป็นรายละเอียดของวงจรที่ต้องการออกแบบ และเขียนฟังก์ชันการทำงานของวงจรตามที่ถูกออกแบบกำหนดด้วยภาษาวีเอชดีแอลในระดับอาร์ทีแอล (Register Transfer Level: RTL) ซึ่งเป็นการใส่รายละเอียดลงไปเป็นหน่วยบันทึก (Register) หน่วยความจำ (Memory) หน่วยคำนวณและตรรกะ (Arithmetic Logic Unit: ALU) และเครื่องสถานะ (State machine)

ขั้นตอนที่ 2 จำลองการทำงานแบบจำลองวงจรระดับอาร์ทีแอล (RTL Simulation)

เป็นขั้นตอนตรวจสอบการทำงานของแบบจำลองวงจรในระดับอาร์ทีแอล โดยการจำลองการทำงานจะถูกทดสอบโดยการสร้างระบบการทดสอบของแบบจำลองที่ต้องการทดสอบหรือเทสต์เบนช์ (Testbench) ซึ่งในขั้นตอนนี้จะจำลองการทำงานเพียงฟังก์ชัน ไม่คำนึงถึงค่าการประวิง (Delay) ของวงจรแต่อย่างใด

ขั้นตอนที่ 3 สังเคราะห์และการทำงานจริงให้เหมาะสมที่สุด (Synthesis & Optimization)

เป็นขั้นตอนการสร้างแผนภาพวงจร (Schematic) จากแบบจำลองวงจรระดับอาร์ทีแอล ให้อยู่ในรูปของลอจิกเกต โดยอาศัยซอฟต์แวร์ช่วยในการสังเคราะห์วงจร โดยขั้นตอนนี้จะต้องมีการเลือกใช้เทคโนโลยีเอพพีจีเอที่ผู้ออกแบบต้องการเลือกใช้ ซึ่งบริษัทผู้ผลิตเอพพีจีเอจะมีเทคโนโลยีไลบรารี (Technology library) เตรียมไว้ให้ผู้ออกแบบไว้ในซอฟต์แวร์ที่ใช้พัฒนาชิพหรือบอร์ดเอพพีจีเอของแต่ละบริษัทไว้เรียบร้อยแล้ว โดยเมื่อสังเคราะห์ได้ผังวงจร ซอฟต์แวร์จะทำงานจริงให้เหมาะสมที่สุดตามข้อกำหนดหรือเงื่อนไขของการสังเคราะห์ (Design constraints) ที่ผู้ออกแบบกำหนดขึ้นตามขั้นตอนแรกของการออกแบบ ซึ่งผลลัพธ์ที่ได้จากการสังเคราะห์วงจรจะอยู่ในรูปแบบของไฟล์วีเอชดีแอลโครงสร้าง (Structured VHDL) และไฟล์เน็ตลิสต์มาตรฐาน (Netlist) ประเภทอีดีไอเอฟ (Electronic Design Interchange Format: EDIF) ที่จะนำไปใช้ในขั้นตอนการวางและเชื่อมต่อเซลล์ภายในของเอพพีจีเอ (Place & Route) ต่อไป

ขั้นตอนที่ 4 การจำลองการทำงานของวงจรระดับลอจิกเกต (Gate Level Simulation)

เป็นขั้นตอนที่ผู้ออกแบบจะต้องทดสอบไฟล์เน็ตลิสต์ที่เป็นแบบจำลองของวงจรระดับลอจิกเกต โดยใช้เทสต์เบนช์ตัวเดิมที่ใช้จำลองการทำงานระดับอาร์ทีแอลมาแล้ว ซึ่งในการจำลองการทำงานในระดับนี้ จะมีเรื่องของความประวิงของเกต (Gate delay) เข้ามาเกี่ยวข้องในผลการจำลองการทำงาน ซึ่งจะแตกต่างจากการจำลองการทำงานในระดับอาร์ทีแอล ดังนั้นผู้ออกแบบจะต้องจำลองการทำงานเพื่อตรวจสอบไทม์มิง (Timing) อีกครั้งหนึ่งว่ายังถูกต้องตามข้อกำหนดของวงจรหรือไม่ ถ้าไม่ตรงตามข้อกำหนดจะต้องกลับไปขั้นตอนที่ 3 เพื่อสังเคราะห์และทำงานจริงให้เหมาะสมที่สุดใหม่

ขั้นตอนที่ 5 การวางและเชื่อมต่อเซลล์ภายในของเอพพีจีเอ (Place & Route)

เมื่อตรวจสอบการทำงานของวงจรระดับลอจิกเกตเป็นที่เรียบร้อยแล้ว จะต้องนำไฟล์เน็ตลิสต์ที่อยู่ในรูปแบบอีดีไอเอฟมาแปลงลงสู่เทคโนโลยีเซลล์ภายในของเอพพีจีเอ และเชื่อมต่อเซลล์ภายในเข้าด้วยกันตามรูปแบบการเชื่อมต่ออุปกรณ์ต่าง ๆ ภายในเน็ตลิสต์ โดยขั้นตอนนี้จะมีการเรียกใช้เทคโนโลยีเซลล์ของเอพพีจีเอ เนื่องจากเทคโนโลยีเซลล์ของเอพพีจีเออาจมีมาโครเซลล์ (Macro

cells) สำหรับสร้างฟังก์ชันต่าง ๆ ให้กับผู้ออกแบบใช้งาน เพราะมาโครเซลล์ต่าง ๆ นี้ถูกทางผู้ผลิต ออกแบบไว้สำหรับซีพียูแต่ละตัว หรืออาจกล่าวได้ว่าเป็นวงจรที่ถูกทำให้เหมาะสมที่สุด สำหรับซีพียูแต่ละเทคโนโลยี ซึ่งในการวางและเชื่อมต่อเซลล์หรือมาโครเซลล์ต่าง ๆ จะถูกควบคุม ด้วยข้อกำหนดในการวางและเชื่อมต่อเซลล์ เพื่อให้ซอฟต์แวร์สามารถวางและเชื่อมต่อกันตาม ความต้องการของผู้ออกแบบ (Design constraints) โดยผลลัพธ์จากขั้นตอนนี้จะอยู่ในรูปของไฟล์ วิเอสดีแอลโครงสร้างของเซลล์ภายในของซีพียูและไฟล์ประเภทเอสดีเอฟ (Standard Delay Format: SDF) ซึ่งเป็นไฟล์รูปแบบมาตรฐานที่มีข้อมูลเกี่ยวกับค่าการประวิงของเส้นทางการ เชื่อมต่อภายในเซลล์ (Routing delay) และไฟล์สำหรับใช้โปรแกรมลงซีพียูซีเอ ซึ่งส่วนใหญ่จะ อยู่ในรูปแบบของไฟล์รูปลักษณะ (Configuration file)

ขั้นตอนที่ 6 การจำลองการทำงานวงจรระดับฐานเวลาจริง (Timing Simulation)

เป็นขั้นตอนสุดท้ายของการตรวจสอบความถูกต้อง ก่อนจะนำวงจรที่ออกแบบไป โปรแกรมหรือดาวน์โหลดลงสู่ชิพหรือบอร์ดจริงต่อไป โดยผลลัพธ์ที่ได้จากขั้นตอนนี้จะมีความ ใกล้เคียงกับไทม์มิงการทำงานจริงบนชิพหรือบอร์ดเอชพีซีเอ เนื่องจากมีข้อมูลเกี่ยวกับการประวิง ของเซลล์ภายในซีพียูและการประวิงของการเชื่อมต่อเซลล์เข้ามาเกี่ยวข้อง

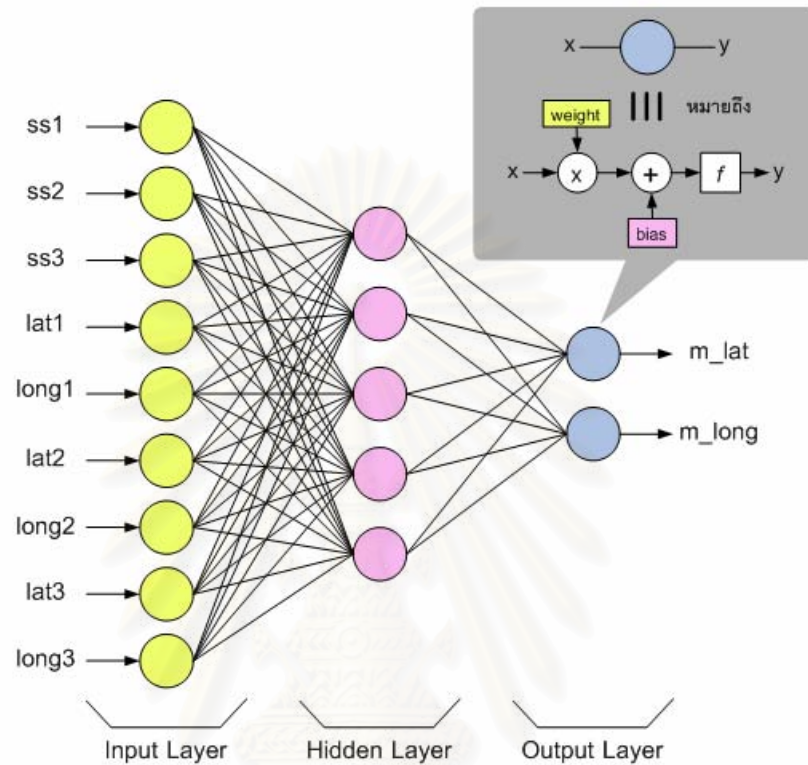
ขั้นตอนที่ 7 โปรแกรมลงสู่ชิพจริง (Download to Device)

เป็นขั้นตอนสุดท้ายสำหรับการออกแบบวงจรนั้นคือ การโปรแกรมไฟล์รูปลักษณะลงสู่ชิพ หรือบอร์ดเอชพีซีเอเพื่อทดสอบการทำงานจริงต่อไป

3.2 รูปแบบของนิวรอลเน็ตเวิร์กเพื่อประมาณค่าตำแหน่งของสถานีเคลื่อนที่ที่ถูก ออกแบบในวิทยานิพนธ์ฉบับนี้

วิทยานิพนธ์ฉบับนี้ได้ออกแบบนิวรอลเน็ตเวิร์กเพื่อนำมาประยุกต์ใช้ในการประมาณค่า ตำแหน่งของสถานีเคลื่อนที่ โดยนิวรอลเน็ตเวิร์กมีสถาปัตยกรรมเป็นแบบโครงข่ายป้อนไปข้างหน้า และมีการฝึกแบบแพร่กระจายย้อนกลับ (Feedforward Network - Backpropagation Training Architecture) ซึ่งมีชั้นข้อมูลเข้า (Input Layer) จำนวน 1 ชั้น จำนวนนิวรอน (Neuron) ในชั้นนี้ เท่ากับจำนวนข้อมูลเข้า (Input) ได้แก่ ค่าข้อมูลความแรงของสัญญาณที่สถานีฐาน 3 สถานีรับได้ จากสถานีเคลื่อนที่ที่พิจารณา (ss1, ss2, ss3) และตำแหน่งละติจูด-ลองจิจูดของสถานีฐานทั้ง 3 สถานี (lat1, long1, lat2, long2, lat3, long3) ฟังก์ชันถ่ายโอนในชั้นนี้เป็นฟังก์ชันถ่ายโอนเชิงเส้น (Linear Transfer Function) มีชั้นซ่อนเร้น (Hidden Layer) จำนวน 1 ชั้น โดยมีการทดสอบหา จำนวนนิวรอนในชั้นซ่อนเร้นที่เหมาะสม ฟังก์ชันถ่ายโอนในชั้นนี้เป็นฟังก์ชันถ่ายโอนลอการิทึมซิก

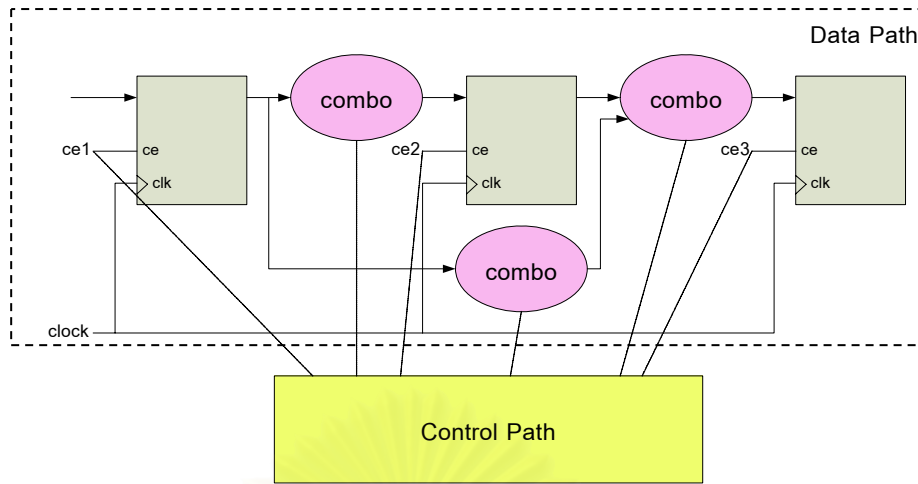
มอดยด์ (Logarithm Sigmoid Transfer Function) และชั้นข้อมูลออก (Output Layer) จำนวน 1 ชั้น จะให้ค่าผลลัพธ์ออกมาเป็นค่าตำแหน่งละติจูด- ลองจิจูด (m_lat, m_long) ของสถานีเคลื่อนที่ ฟังก์ชันถ่ายโอนในชั้นนี้เป็นฟังก์ชันถ่ายโอนเชิงเส้น ดังแสดงในรูปที่ 3.2



รูปที่ 3.2 รูปแบบของนิวรอลเน็ตเวิร์กที่ใช้ในวิทยานิพนธ์ฉบับนี้

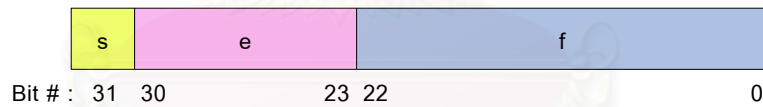
3.3 การออกแบบและสร้างอุปกรณ์ต้นแบบนิวรอลเน็ตเวิร์กเพื่อประมาณค่าตำแหน่งของสถานีเคลื่อนที่ที่น่าเสนอ

การออกแบบนิวรอลเน็ตเวิร์กเพื่อประมาณค่าตำแหน่งของสถานีเคลื่อนที่ที่น่าเสนอในวิทยานิพนธ์ฉบับนี้ มีลักษณะการออกแบบวงจร โดยแบ่งนิวรอลเน็ตเวิร์กเป็นวงจรย่อย ๆ หลายวงจร โดยที่แต่ละวงจรย่อยจะถูกออกแบบให้เป็นวงจรแบบซิงโครนัส และประกอบไปด้วยวิถีข้อมูล (Data Path) และวิถีควบคุม (Control Path) โดยที่วิถีข้อมูลเป็นการทำงานร่วมกันของวงจรชุดผสม (Combination Circuit) และรีจิสเตอร์ (Register) ซึ่งวงจรชุดผสมจะมีหน้าที่ประมวลผลสัญญาณให้ได้ผลลัพธ์ตามที่ต้องการ ส่วนรีจิสเตอร์จะทำหน้าที่รับผลลัพธ์ ณ เวลาต่าง ๆ และเก็บไว้ในระยะเวลาที่ต้องการ ในขณะที่วิถีควบคุมจะสร้างสัญญาณที่จะเข้ามาควบคุมการทำงานของวงจรชุดผสมและรีจิสเตอร์ทั้งหมด ดังแสดงในรูปที่ 3.3



รูปที่ 3.3 ลักษณะการออกแบบวงจรโดยแบ่งเป็นวิถีข้อมูลและวิถีควบคุม

เนื่องจากขีดความสามารถในการออกแบบโดยใช้ภาษาวีเอสดีแอลนั้น สามารถออกแบบได้เฉพาะวงจรที่มีลักษณะเป็นระบบดิจิทัลเท่านั้น สัญญาณต่าง ๆ ภายในวงจรจึงต้องอยู่ในรูปแบบดิจิทัล ซึ่งในวิทยานิพนธ์ฉบับนี้ ค่าของสัญญาณอินพุตต่าง ๆ อยู่ในรูปแบบของจำนวนจริง (เลขทศนิยม) จึงมีความจำเป็นต้องแปลงค่าสัญญาณที่เป็นจำนวนจริงให้เป็นสัญญาณดิจิทัล โดยวิทยานิพนธ์นี้เลือกใช้สัญญาณดิจิทัลแบบฟลอยติงพอยนต์ (Floating point format) ขนาด 32 บิต [8] ซึ่งสามารถแทนค่าจำนวนจริงได้ตั้งแต่ $\pm 6.80565 \times 10^{-38}$ ถึง $\pm 8.81625 \times 10^{-39}$ ดังแสดงในรูปที่ 3.4



รูปที่ 3.4 ฟลอยติงพอยนต์ 32 บิต

สาเหตุที่ใช้สัญญาณดิจิทัลแบบฟลอยติงพอยนต์ที่มีจำนวนบิตมากนั้น เนื่องจากในวิทยานิพนธ์ฉบับนี้ ค่าผลลัพธ์ที่ต้องการคือตำแหน่งละติจูด – ลองจิจูดของสถานีเคลื่อนที่ ซึ่งการประมาณค่าผิดพลาดแค่เพียงทศนิยมตำแหน่งที่ 3 ถือว่าเป็นความผิดพลาดที่ไม่สามารถยอมรับได้ (ประมาณค่าผิดพลาดเพียง 0.001 องศา เท่ากับประมาณค่าผิดพลาดประมาณ 111 เมตร) จึงต้องใช้สัญญาณที่ให้ความละเอียดของข้อมูลสูง โดยการหาค่า s e และ f ทำได้ดังตัวอย่างต่อไปนี้

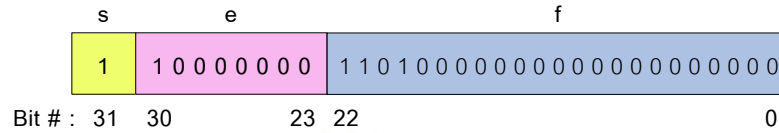
- ตัวอย่าง การแปลงเลขจำนวนจริง -3.625 ให้อยู่ในรูปแบบฟลอยติงพอยนต์ 32 บิต
- s = '1' (เนื่องจาก -3.625 มีค่าเป็นลบ โดยที่ s จะเท่ากับ '0' หากมีค่าเป็นบวก)
- 3.625 (เลขฐานสิบ) = 11.101 (เลขฐานสอง)
- ทำ 11.101 ให้อยู่ในรูป 1.f ได้เป็น 1.1101 (เลื่อนจุดไปทางซ้าย 1 ตำแหน่ง)

ค่า e สามารถหาได้จาก $e^{-127} =$ จำนวนการเลื่อนจุดไปทางซ้าย

จะได้ $e^{-127} = 1$ ดังนั้น $e = 128$ (เลขฐานสิบ) = 10000000 (เลขฐานสอง)

จะได้ $f = 110100000000000000000000$ (เติม 0 ให้ครบ 23 บิต)

ดังนั้น -3.625 สามารถแปลงให้อยู่ในรูปแบบฟลอยตติ้งพอยนต์ 32 บิต ดังแสดงในรูปที่ 3.5



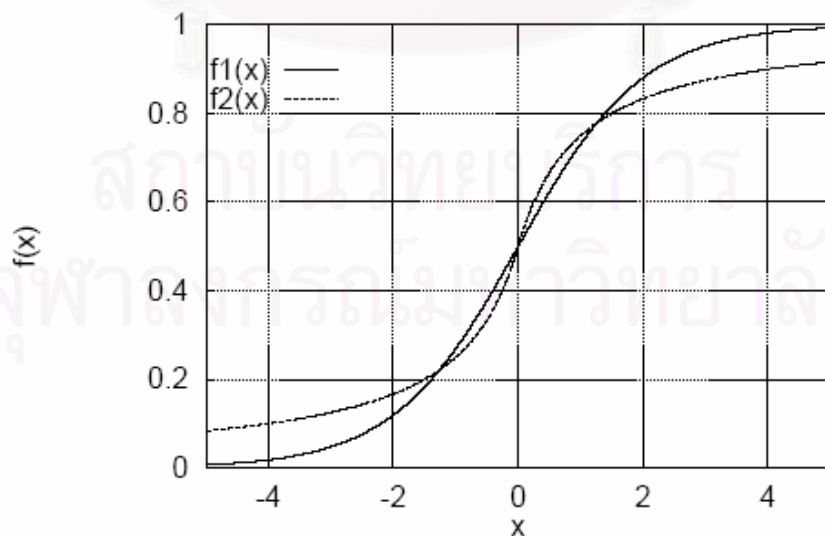
รูปที่ 3.5 จำนวนจริง -3.625 หลังถูกแปลงให้อยู่ในรูปแบบฟลอยตติ้งพอยนต์ 32 บิต

และเนื่องจากในชั้นซ่อนเร้นของนิวรอลเน็ตเวิร์กที่ออกแบบไว้ มีฟังก์ชันถ่ายโอนเป็นฟังก์ชันถ่ายโอนลอการิทึมซิกมอยด์ ซึ่งเป็นไปตามสมการที่ (3.1)

$$f_1(x) = \frac{1}{1 + e^{-x}} \quad (3.1)$$

ซึ่งในภาษาวีเอชดีแอล การเขียนฟังก์ชันลักษณะนี้ไม่สามารถทำได้โดยง่าย จึงได้นำฟังก์ชันอีกแบบหนึ่งที่สามารถใช้แทนฟังก์ชันถ่ายโอนลอการิทึมซิกมอยด์ได้อย่างใกล้เคียง [9] และสามารถเขียนด้วยภาษาวีเอชดีแอลได้ง่ายกว่ามาก ซึ่งเป็นไปตามสมการที่ (3.2)

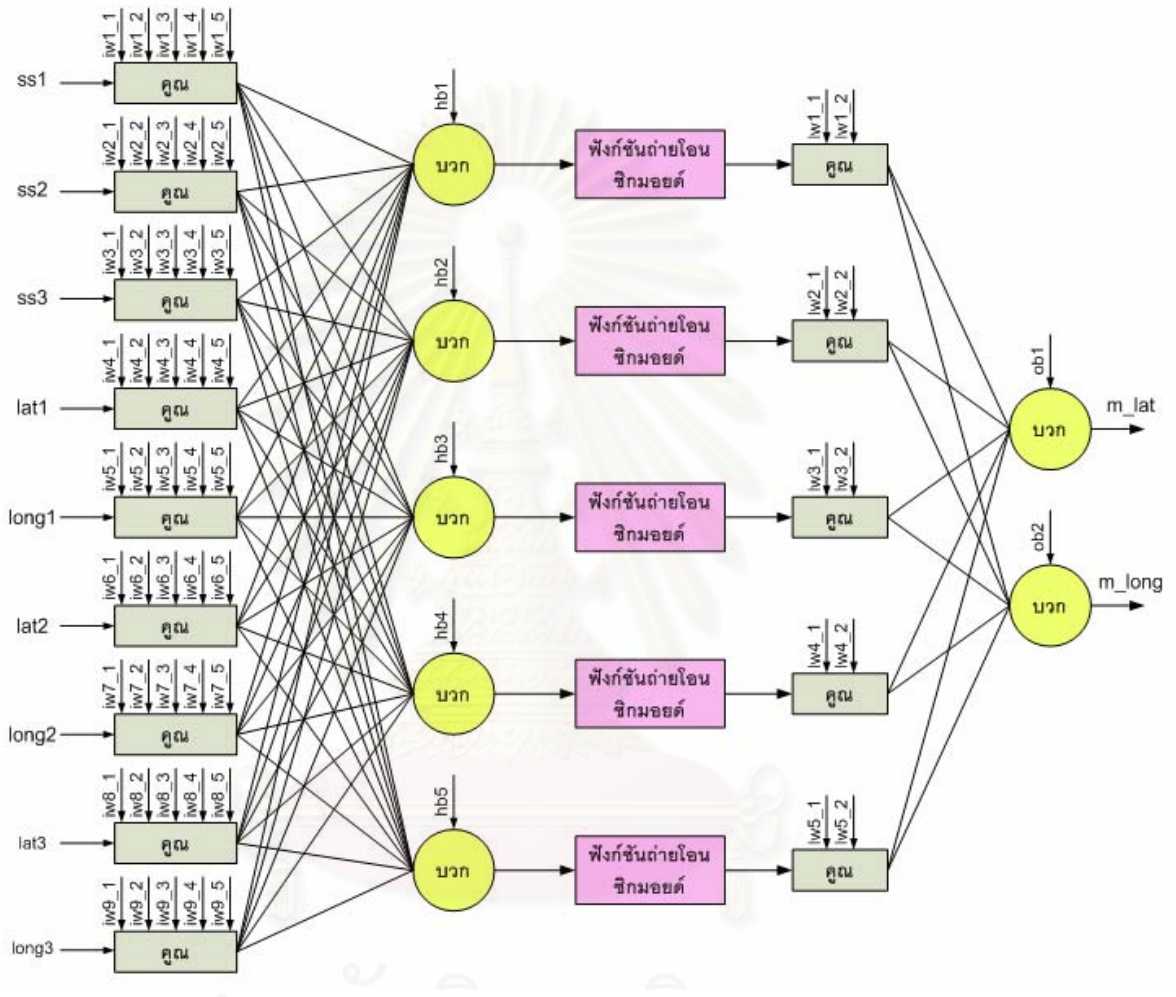
$$f_2(x) = \frac{1}{2} \left(\frac{x}{1 + |x|} + 1 \right) \quad (3.2)$$



รูปที่ 3.6 คุณลักษณะของฟังก์ชันประมาณของฟังก์ชันถ่ายโอนลอการิทึมซิกมอยด์
เปรียบเทียบกับฟังก์ชันถ่ายโอนลอการิทึมซิกมอยด์ [9]

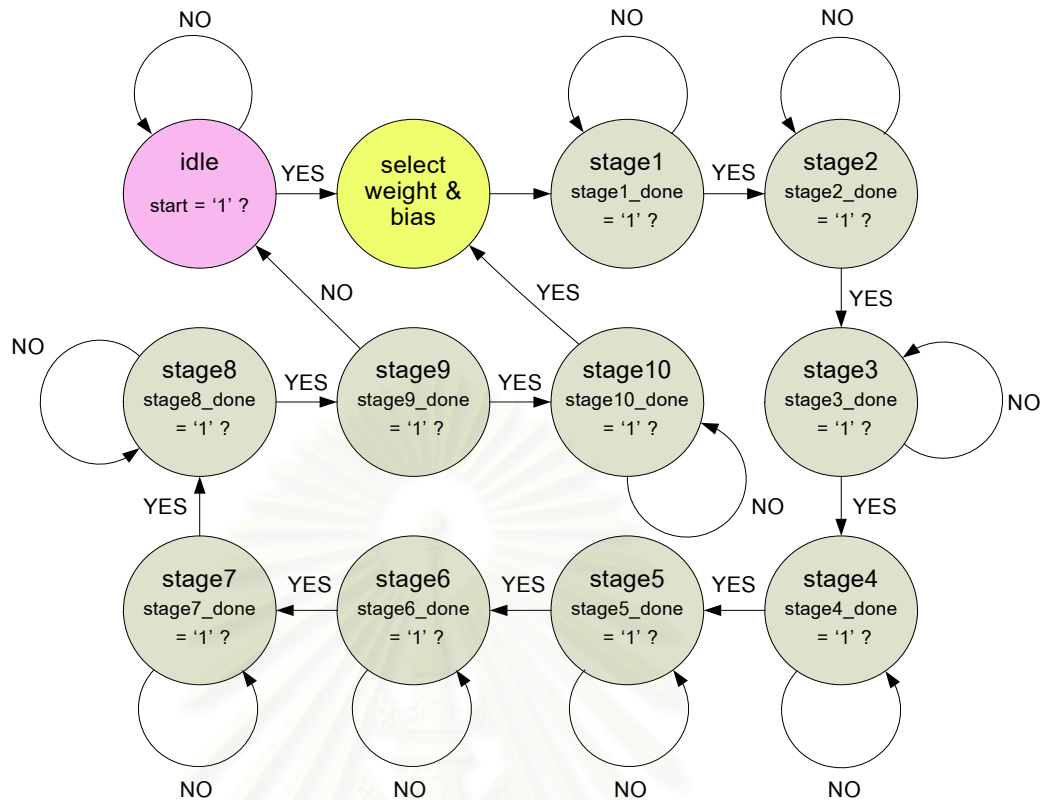
3.3.1 การออกแบบนิรอลเน็ตเวิร์กเพื่อประมาณค่าตำแหน่งของสถานีเคลื่อนที่ด้วยภาษาวีเอสดีแอล

นิรอลเน็ตเวิร์กในรูปที่ 3.2 สามารถแสดงให้อยู่ในรูปของสัญลักษณ์ต่าง ๆ ที่ใช้ในการออกแบบดังแสดงในรูปที่ 3.7



รูปที่ 3.7 ภาพรวมของนิรอลเน็ตเวิร์กที่ออกแบบในวิทยานิพนธ์ฉบับนี้

วิทยานิพนธ์ฉบับนี้ได้ออกแบบนิรอลเน็ตเวิร์กเพื่อประมาณค่าตำแหน่งของสถานีเคลื่อนที่ โดยออกแบบเป็นวงจรย่อยหลาย ๆ วงจรเพื่อลดความซับซ้อนในการออกแบบและง่ายสำหรับตรวจสอบจุดผิดพลาด ซึ่งนิรอลเน็ตเวิร์กในรูปที่ 3.7 สามารถออกแบบเป็นวงจรย่อย 11 วงจร ได้แก่ วงจรเลือกค่าถ่วงน้ำหนักและค่าไบแอส (Select weight and bias) วงจรชั้นที่ 1 (Stage1) วงจรชั้นที่ 2 (Stage2) วงจรชั้นที่ 3 (Stage3) วงจรชั้นที่ 4 (Stage4) วงจรชั้นที่ 5 (Stage5) วงจรชั้นที่ 6 (Stage6) วงจรชั้นที่ 7 (Stage7) วงจรชั้นที่ 8 (Stage8) วงจรชั้นที่ 9 (Stage9) และวงจรชั้นที่ 10 (Stage10) โดยมีขั้นตอนการทำงานร่วมกันดังแสดงในรูปที่ 3.8

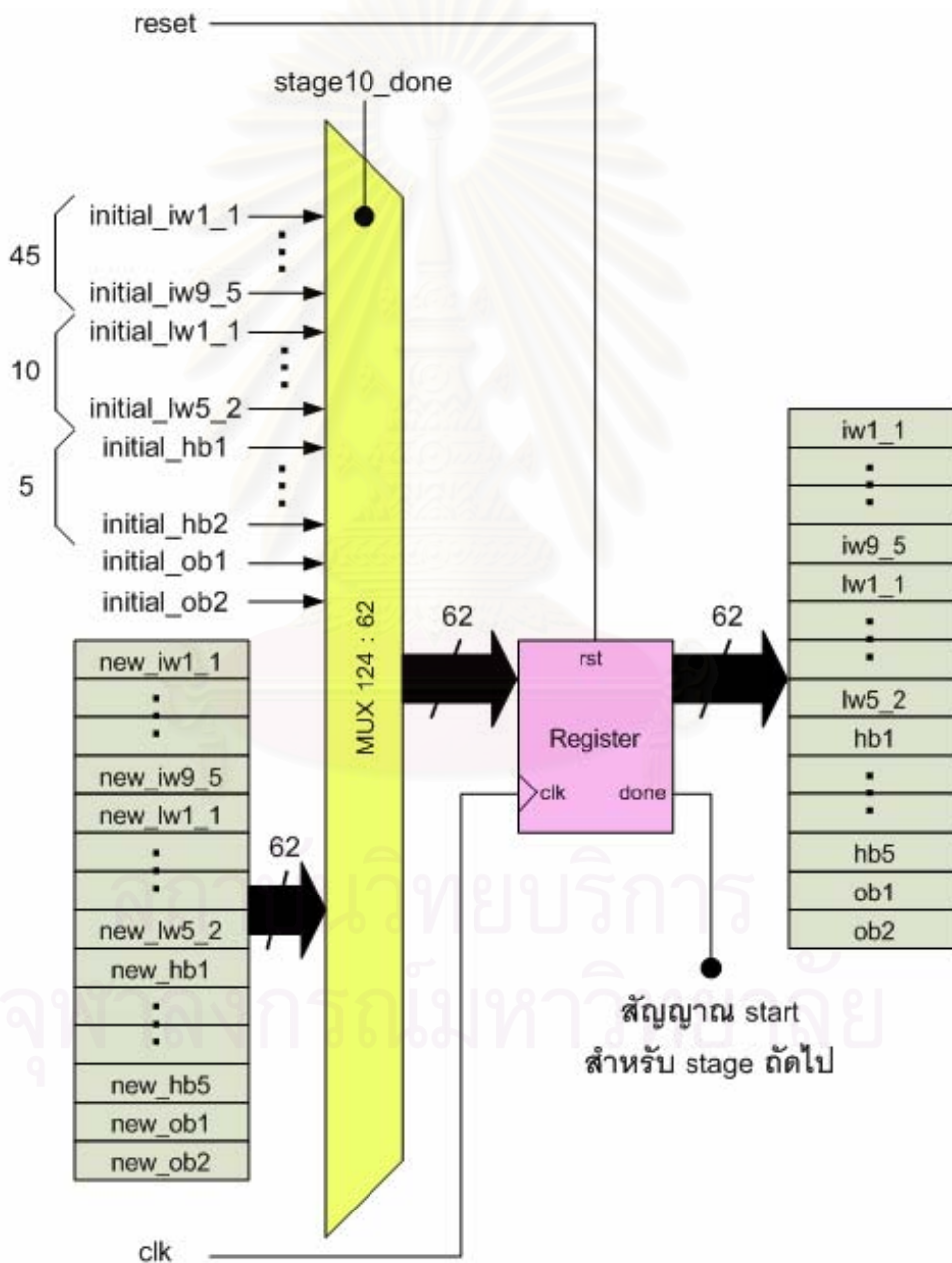


รูปที่ 3.8 ขั้นตอนการทำงานร่วมกันของวงจรรย่อยต่าง ๆ

เริ่มต้น นิวรอลเน็ตเวิร์กจะอยู่ในสถานะว่าง (Idle) ไม่มีการทำงานใด ๆ เกิดขึ้น จนกระทั่งผู้ใช้กดปุ่มเริ่มต้นการทำงาน (สัญญาณ start = '1') วงจรเลือกค่าถ่วงน้ำหนักและค่าไบแอสจะเริ่มทำงาน และส่งค่าถ่วงน้ำหนักและค่าไบแอสไปยังวงจรขั้นที่ 1 เมื่อวงจรขั้นที่ 1 ทำงานเสร็จสิ้น (สัญญาณ stage1_done = '1') วงจรขั้นที่ 2 จึงเริ่มทำงาน นั่นคือ สัญญาณ stage1_done จะกลายเป็นสัญญาณ start ของวงจรขั้นที่ 2 นั่นเอง ลักษณะการทำงานจะเป็นเช่นนี้ไปเรื่อย ๆ จนถึงการทำงานของวงจรขั้นที่ 4 ซึ่งให้ผลลัพธ์เป็นค่าตำแหน่งของสถานะเคลื่อนที่ที่พิจารณา หลังจากนั้นนิวรอลเน็ตเวิร์กจะเข้าสู่กระบวนการคำนวณเพื่อหาค่าถ่วงน้ำหนักและค่าไบแอสใหม่ เพื่อเป็นค่าพารามิเตอร์ในการคำนวณรอบต่อไป วงจรขั้นที่ 5 วงจรขั้นที่ 6 วงจรขั้นที่ 7 วงจรขั้นที่ 8 และวงจรขั้นที่ 9 จะทำงานต่อ ๆ กันตามลำดับ ในวงจรขั้นที่ 9 ซึ่งให้ผลลัพธ์เป็นส่วนต่างระหว่างค่าถ่วงน้ำหนักและค่าไบแอสรอบใหม่กับค่าถ่วงน้ำหนักและค่าไบแอสรอบก่อนหน้า หากค่าส่วนต่างดังกล่าวเท่ากับศูนย์ถือว่าการทำงานทั้งหมดเสร็จสิ้น (สัญญาณ stage9_done = '0') นิวรอลเน็ตเวิร์กจะกลับสู่สถานะว่างอีกครั้ง แต่หากไม่เท่ากับศูนย์ (สัญญาณ stage9_done = '1') วงจรขั้นที่ 10 จะเริ่มต้นทำงาน นิวรอลเน็ตเวิร์กจะได้ค่าถ่วงน้ำหนักและค่าไบแอสใหม่ และส่งค่าไปยังวงจรเลือกค่าถ่วงน้ำหนักและค่าไบแอสเพื่อเริ่มต้นการทำงานทั้งหมดใหม่อีกครั้ง รายละเอียดในการออกแบบของวงจรรย่อยต่าง ๆ แสดงได้ดังต่อไปนี้

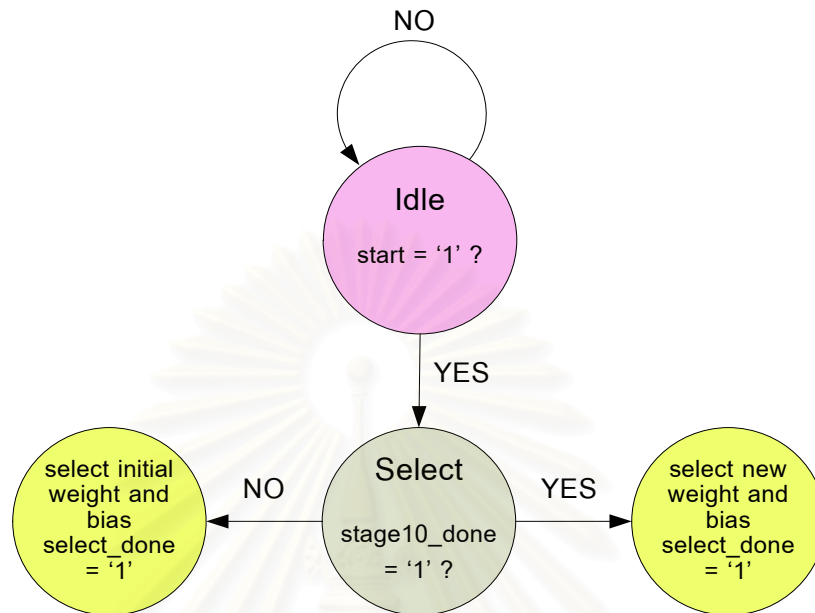
วงจรเลือกค่าถ่วงน้ำหนักและค่าไบแอส (Select weight and bias)

ตามปกติ ในการฝึกนิรอรอลเน็ตเวิร์ก ต้องมีการกำหนดค่าเริ่มต้นให้กับค่าถ่วงน้ำหนักและค่าไบแอสเพื่อใช้เป็นพารามิเตอร์เริ่มต้นในการคำนวณของนิรอรอลเน็ตเวิร์ก แต่เมื่อการทำงานของวงจรย่อยทั้งหมดเสร็จสิ้น จะได้ค่าถ่วงน้ำหนักและค่าไบแอสใหม่เพื่อใช้ในการคำนวณรอบต่อไป วงจรนี้ทำหน้าที่ตัดสินใจว่าจะเลือกค่าถ่วงน้ำหนักและค่าไบแอสที่เป็นค่าเริ่มต้น หรือค่าถ่วงน้ำหนักและค่าไบแอสใหม่ วิธีข้อมูลของวงจรนี้แสดงได้ดังรูปที่ 3.9



รูปที่ 3.9 วิธีข้อมูลของวงจรเลือกค่าถ่วงน้ำหนักและค่าไบแอส

วิธีควบคุมของวงจรเลือกค่าถ่วงน้ำหนักและค่าไบแอสออกแบบโดยใช้ Finite-State Machine แสดงได้ดังรูปที่ 3.10



รูปที่ 3.10 วิธีควบคุมของวงจรเลือกค่าถ่วงน้ำหนักและค่าไบแอส

หลักการการทำงานของวงจรเลือกค่าถ่วงน้ำหนักและค่าไบแอส คือ เริ่มต้นวงจรจะอยู่ในสถานะว่าง (Idle) เมื่อสัญญาณ start = '1' วงจรจะเข้าสู่สถานะเลือก (Select) วงจรจะพิจารณาว่าหากสัญญาณ stage10_done ที่ได้รับจากวงจรชั้นที่ 10 เป็นลอจิก '0' วงจรจะเลือกค่าเริ่มต้นของค่าถ่วงน้ำหนักและค่าไบแอส (initial_iw1_1, initial_iw1_2, ..., initial_ob2) แต่หากสัญญาณ stage10_done ที่ได้รับจากวงจรชั้นที่ 10 เป็นลอจิก '1' วงจรจะเลือกค่าถ่วงน้ำหนักและค่าไบแอสใหม่ (new_iw1_1, new_iw1_2, ..., new_ob2) เพื่อใช้เป็นค่าถ่วงน้ำหนักและค่าไบแอสในการคำนวณในวงจรย่อยอื่น ๆ (iw1_1, iw1_2, ..., ob2) และสัญญาณ select_done จะมีลอจิกเป็น '1' เพื่อส่งค่าไปยังวงจรชั้นที่ 1 ต่อไป

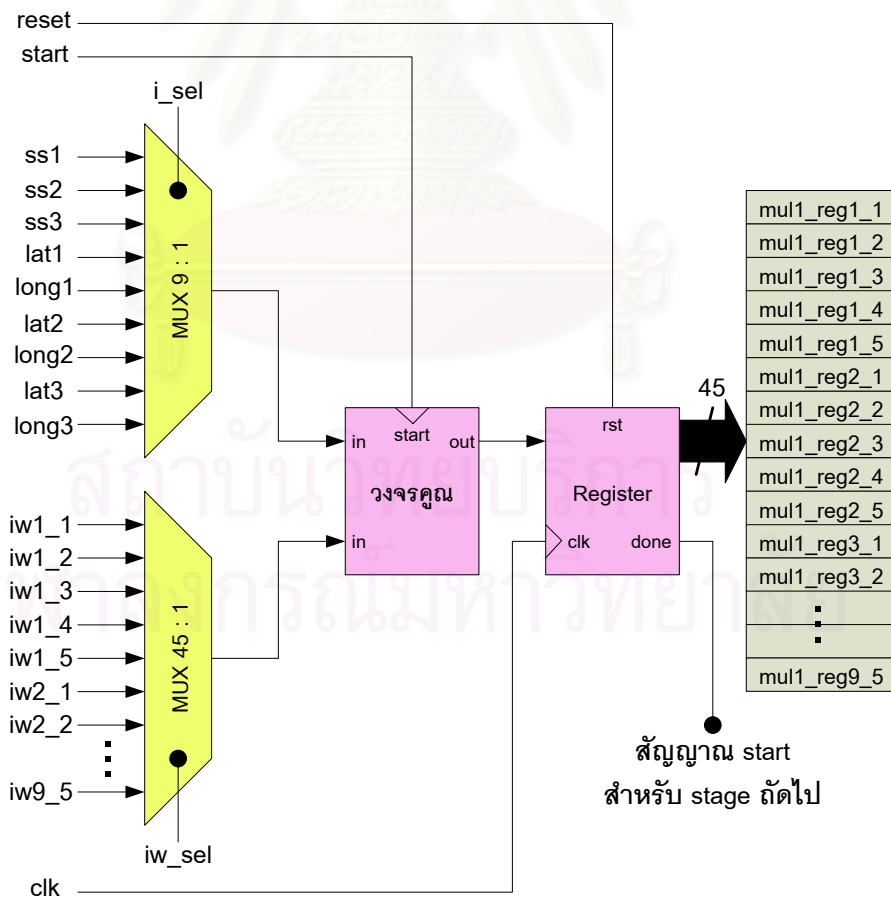
สำหรับวงจรชั้นที่ 1 ถึงวงจรชั้นที่ 10 จะเป็นวงจรในส่วนการคำนวณต่าง ๆ ตามหลักการทำงานของนิวรอนเน็ตเวิร์กที่กล่าวไว้แล้วในบทที่ 2 ซึ่งมีรายละเอียดดังต่อไปนี้

วงจรขั้นที่ 1 (Stage1)

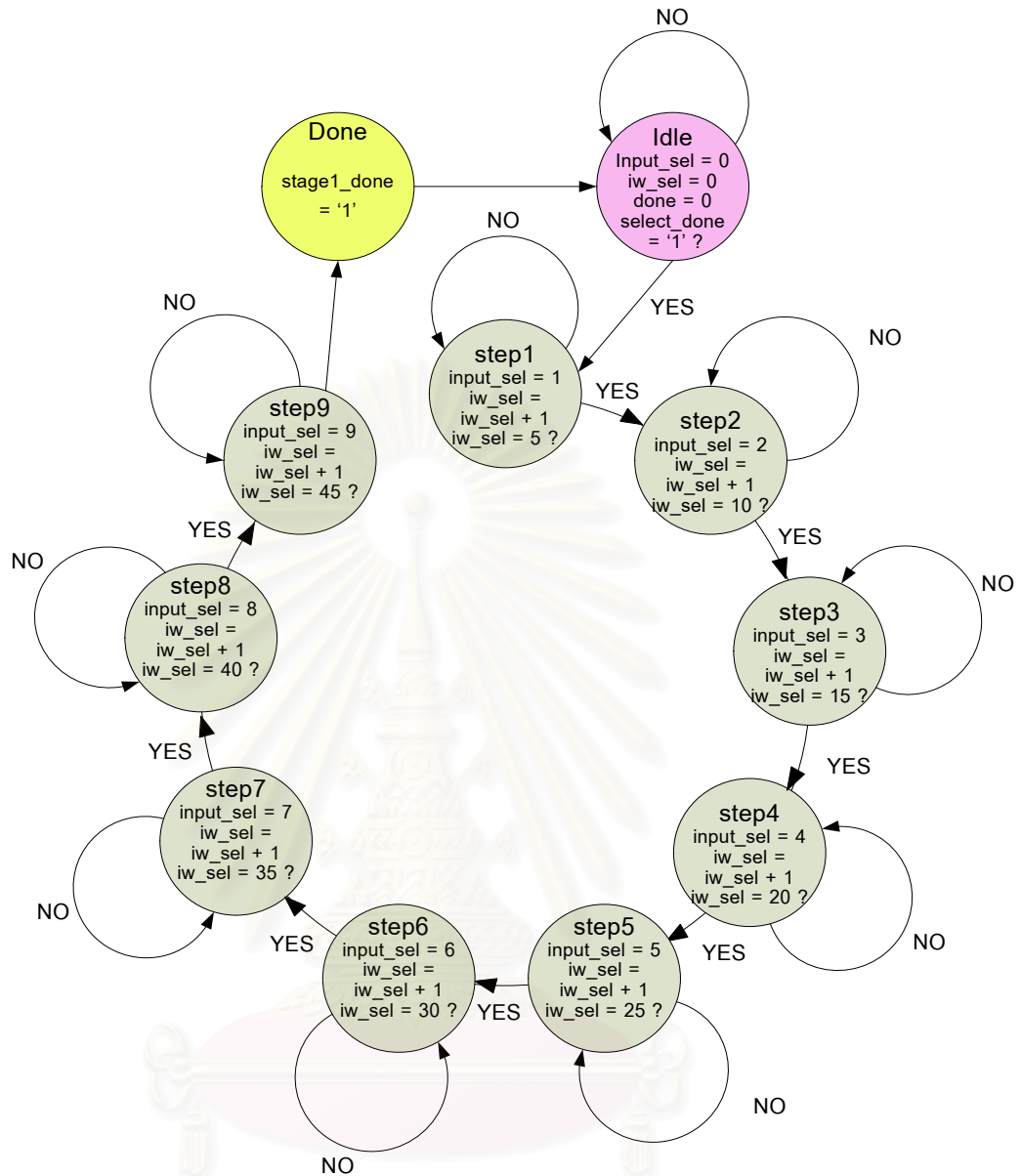
วงจรขั้นที่ 1 มีหน้าที่นำสัญญาณอินพุตทั้งหมดของนิวรอลเน็ตเวิร์กมาคูณกับค่าถ่วงน้ำหนักบางส่วนตามสมการที่ (3.3)

$$\begin{aligned}
 ss1 \times \begin{bmatrix} iw1_1 \\ iw1_2 \\ iw1_3 \\ iw1_4 \\ iw1_5 \end{bmatrix} &= \begin{bmatrix} mul1_reg1_1 \\ mul1_reg1_2 \\ mul1_reg1_3 \\ mul1_reg1_4 \\ mul1_reg1_5 \end{bmatrix}, & \quad ss2 \times \begin{bmatrix} iw2_1 \\ iw2_2 \\ iw2_3 \\ iw2_4 \\ iw2_5 \end{bmatrix} &= \begin{bmatrix} mul1_reg2_1 \\ mul1_reg2_2 \\ mul1_reg2_3 \\ mul1_reg2_4 \\ mul1_reg2_5 \end{bmatrix}, \\
 ss3 \times \begin{bmatrix} iw3_1 \\ iw3_2 \\ iw3_3 \\ iw3_4 \\ iw3_5 \end{bmatrix} &= \begin{bmatrix} mul1_reg3_1 \\ mul1_reg3_2 \\ mul1_reg3_3 \\ mul1_reg3_4 \\ mul1_reg3_5 \end{bmatrix}, \dots, & \quad long3 \times \begin{bmatrix} iw9_1 \\ iw9_2 \\ iw9_3 \\ iw9_4 \\ iw9_5 \end{bmatrix} &= \begin{bmatrix} mul1_reg9_1 \\ mul1_reg9_2 \\ mul1_reg9_3 \\ mul1_reg9_4 \\ mul1_reg9_5 \end{bmatrix}
 \end{aligned}
 \tag{3.3}$$

วงจรขั้นที่ 1 มีวิธีข้อมูลและวิธีควบคุมแสดงได้ดังรูปที่ 3.11 และ 3.12 ตามลำดับ



รูปที่ 3.11 วิธีข้อมูลของวงจรขั้นที่ 1



รูปที่ 3.12 วิธีควบคุมของวงจรขั้นที่ 1

หลักการการทำงานของวงจรขั้นที่ 1 คือ เริ่มต้นวงจรจะอยู่ในสถานะว่าง จนกระทั่งสัญญาณ select_done ที่ถูกส่งค่ามาจากวงจรเลือกค่าถ่วงน้ำหนักและค่าไบแอสเป็นลอจิก '1' วงจรจะเริ่มต้นคูณค่าอินพุตเข้ากับค่าถ่วงน้ำหนักตามสมการที่ 3.1 (step1 ถึง step10) โดยมีสัญญาณ input_sel เป็นสัญญาณสำหรับเลือกค่าอินพุต และสัญญาณ iw_sel เป็นสัญญาณสำหรับเลือกค่าถ่วงน้ำหนัก เมื่อการคูณทั้งหมดเสร็จสิ้น สัญญาณ stage1_done จะมีลอจิกเป็น '1' และถูกส่งค่าไปยังวงจรขั้นที่ 2

วงจรขั้นที่ 2 (Stage2)

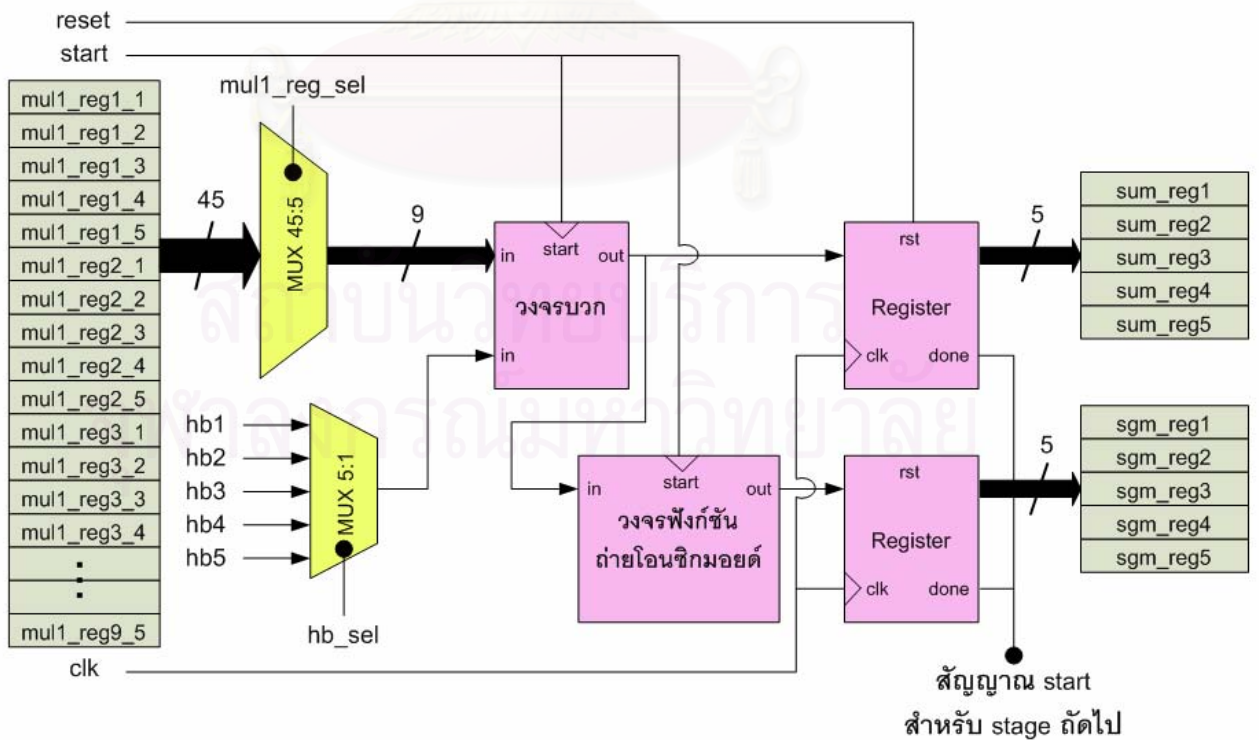
วงจรขั้นที่ 2 มีหน้าที่นำสัญญาณค่าต่าง ๆ และค่าไบแอสบางส่วนมาคำนวณตามสมการที่ (3.4) (3.5) และ (3.6)

$$\begin{bmatrix} mul1_reg1_1 + mul1_reg2_1 + mul1_reg3_1 + mul1_reg4_1 + mul1_reg5_1 + hb1 \\ mul1_reg1_2 + mul1_reg2_2 + mul1_reg3_2 + mul1_reg4_2 + mul1_reg5_2 + hb2 \\ mul1_reg1_3 + mul1_reg2_3 + mul1_reg3_3 + mul1_reg4_3 + mul1_reg5_3 + hb3 \\ mul1_reg1_4 + mul1_reg2_4 + mul1_reg3_4 + mul1_reg4_4 + mul1_reg5_4 + hb4 \\ mul1_reg1_5 + mul1_reg2_5 + mul1_reg3_5 + mul1_reg4_5 + mul1_reg5_5 + hb5 \end{bmatrix} = \begin{bmatrix} sum_reg1 \\ sum_reg2 \\ sum_reg3 \\ sum_reg4 \\ sum_reg5 \end{bmatrix} \quad (3.4)$$

$$\begin{bmatrix} f(sum_reg1) \\ f(sum_reg2) \\ f(sum_reg3) \\ f(sum_reg4) \\ f(sum_reg5) \end{bmatrix} = \begin{bmatrix} sgm_reg1 \\ sgm_reg2 \\ sgm_reg3 \\ sgm_reg4 \\ sgm_reg5 \end{bmatrix} \quad (3.5)$$

$$f(x) = \frac{1}{2} \left(\frac{x}{1+|x|} + 1 \right) \quad (3.6)$$

วงจรขั้นที่ 2 มีวิธีข้อมูลแสดงได้ดังรูปที่ 3.13 ส่วนวิธีกควบคุมของวงจรขั้นที่ 2 มีลักษณะการออกแบบเช่นเดียวกับวงจรขั้นที่ 1 นั่นคือ มีสัญญาณสำหรับเลือกค่าต่าง ๆ เพื่อนำมาคำนวณ เมื่อการคำนวณเสร็จสิ้น สัญญาณ stage2_done จะมีลอจิกเป็น '1' และถูกส่งค่าสู่วงจรขั้นที่ 3 ต่อไป



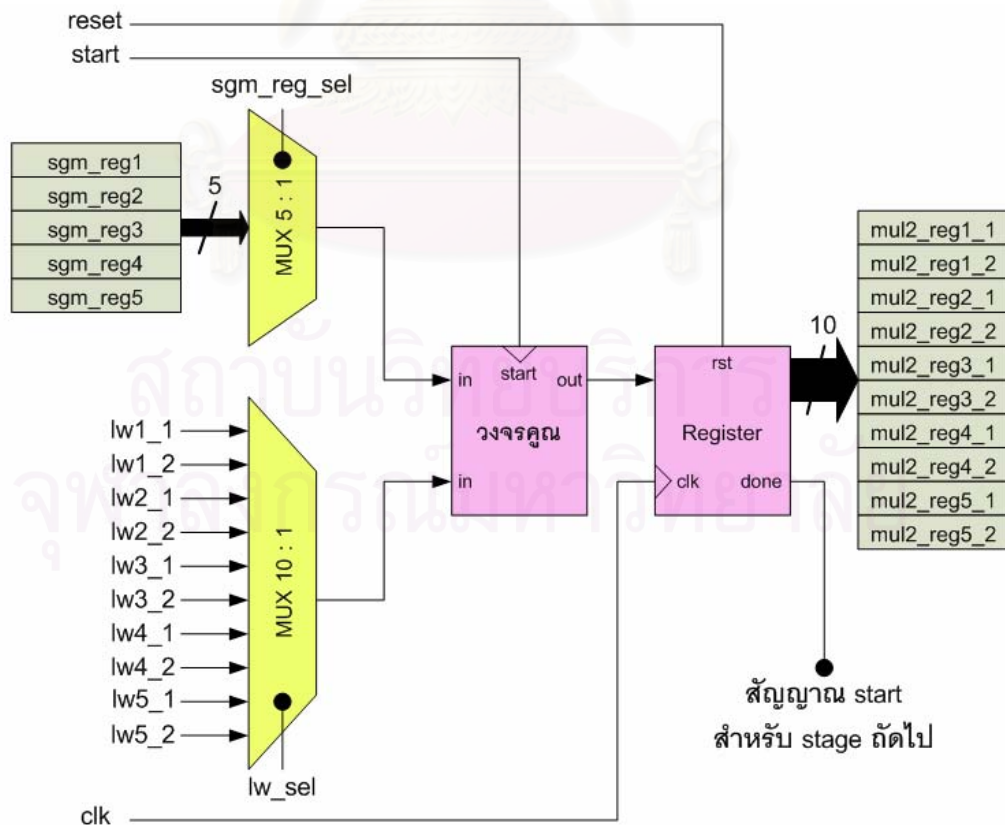
รูปที่ 3.13 วิธีข้อมูลของวงจรขั้นที่ 2

วงจรชั้นที่ 3 (Stage3)

วงจรชั้นที่ 3 มีหน้าที่นำสัญญาณค่าต่าง ๆ บางส่วนที่ได้มาจากวงจรชั้นที่ 2 มาคูณกับค่าถ่วงน้ำหนักบางส่วนตามสมการที่ (3.7)

$$\begin{aligned}
 sgm_reg1 \times \begin{bmatrix} lw1_1 \\ lw1_2 \end{bmatrix} &= \begin{bmatrix} mul2_reg1_1 \\ mul2_reg1_2 \end{bmatrix}, \\
 sgm_reg2 \times \begin{bmatrix} lw2_1 \\ lw2_2 \end{bmatrix} &= \begin{bmatrix} mul2_reg2_1 \\ mul2_reg2_2 \end{bmatrix}, \\
 sgm_reg3 \times \begin{bmatrix} lw3_1 \\ lw3_2 \end{bmatrix} &= \begin{bmatrix} mul2_reg3_1 \\ mul2_reg3_2 \end{bmatrix}, \\
 sgm_reg4 \times \begin{bmatrix} lw4_1 \\ lw4_2 \end{bmatrix} &= \begin{bmatrix} mul2_reg4_1 \\ mul2_reg4_2 \end{bmatrix}, \\
 sgm_reg4 \times \begin{bmatrix} lw5_1 \\ lw5_2 \end{bmatrix} &= \begin{bmatrix} mul2_reg5_1 \\ mul2_reg5_2 \end{bmatrix}
 \end{aligned} \tag{3.7}$$

วงจรชั้นที่ 3 มีวิธีข้อมูลแสดงได้ดังรูปที่ 3.14 ส่วนวิธีกควบคุมของวงจรชั้นที่ 3 มีลักษณะการออกแบบเช่นเดียวกับวงจรชั้นที่ 1 นั่นคือ มีสัญญาณสำหรับเลือกค่าต่าง ๆ เพื่อนำมาคำนวณเมื่อดีคำนวณเสร็จสิ้น สัญญาณ stage3_done จะมีลอจิกเป็น '1' และถูกส่งค่าสู่วงจรชั้นที่ 4 ต่อไป



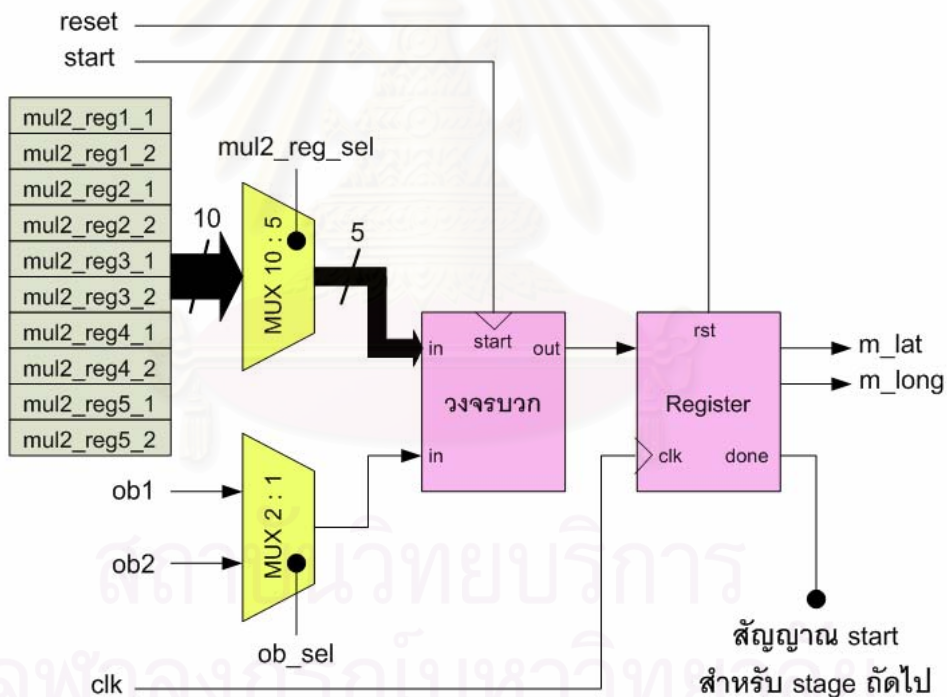
รูปที่ 3.14 วิธีข้อมูลของวงจรชั้นที่ 3

วงจรชั้นที่ 4 (Stage4)

วงจรชั้นที่ 4 มีหน้าที่นำสัญญาณค่าต่าง ๆ ที่ได้จากวงจรชั้นที่ 3 และค่าไบแอสที่เหลือ มาคำนวณตามสมการที่ (3.8) ผลลัพธ์ที่ได้จากวงจรมีคือ ค่าตำแหน่งของสถานีเคลื่อนที่ที่พิจารณา (m_lat , m_long)

$$\begin{bmatrix} mul2_reg1_1 + mul2_reg2_1 + mul2_reg3_1 + \\ mul2_reg4_1 + mul2_reg5_1 + ob1 \\ mul2_reg1_2 + mul2_reg2_2 + mul2_reg3_2 + \\ mul2_reg5_2 + mul2_reg4_2 + ob2 \end{bmatrix} = \begin{bmatrix} m_lat \\ m_long \end{bmatrix} \quad (3.8)$$

วงจรชั้นที่ 4 มีวิธีข้อมูลแสดงได้ดังรูปที่ 3.15 ส่วนวิธีควบคุมของวงจรชั้นที่ 4 มีลักษณะการออกแบบเช่นเดียวกับวงจรชั้นที่ 1 เช่นกัน นั่นคือ มีสัญญาณสำหรับเลือกค่าต่าง ๆ เพื่อนำมาคำนวณ เมื่อการคำนวณเสร็จสิ้น สัญญาณ $stage4_done$ จะมีลอจิกเป็น '1' และถูกส่งค่าสู่วงจรชั้นที่ 5 ต่อไป



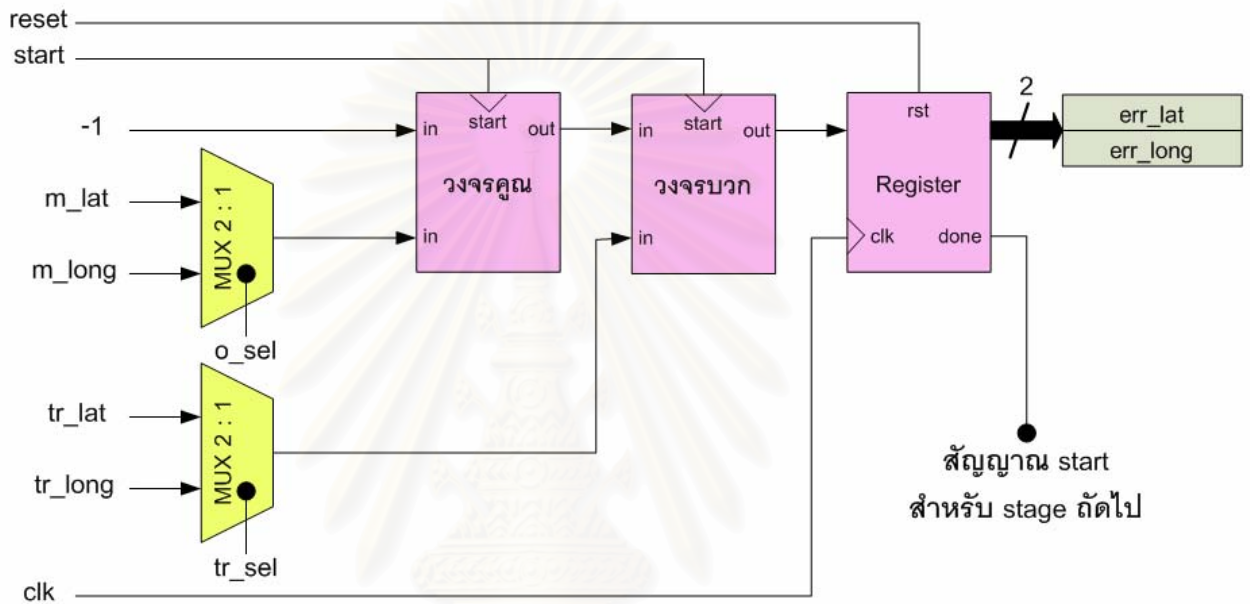
รูปที่ 3.15 วิธีข้อมูลของวงจรชั้นที่ 4

วงจรชั้นที่ 5

วงจรชั้นที่ 5 มีหน้าที่นำค่าตำแหน่งของสถานีเคลื่อนที่ที่ถูกส่งค่ามาจากวงจรชั้นที่ 4 มาเปรียบเทียบกับค่าเป้าหมายของนิรลเนตเวิร์ก (tr_lat , tr_long) ตามสมการที่ (3.9) ผลลัพธ์ที่ได้จะเป็นค่าความผิดพลาดของละติจูดและลองจิจูด (err_lat , err_long)

$$\begin{bmatrix} err_lat \\ err_long \end{bmatrix} = \begin{bmatrix} tr_lat \\ tr_long \end{bmatrix} - \begin{bmatrix} m_lat \\ m_long \end{bmatrix} \quad (3.9)$$

วงจรชั้นที่ 5 มีวิธีข้อมูลแสดงได้ดังรูปที่ 3.16 ส่วนวิธีควบคุมของวงจรชั้นที่ 5 มีลักษณะการออกแบบเช่นเดียวกับวงจรชั้นที่ 1 เช่นกัน นั่นคือ มีสัญญาณสำหรับเลือกค่าต่าง ๆ เพื่อนำมาคำนวณ เมื่อการคำนวณเสร็จสิ้น สัญญาณ stage5_done จะมีลอจิกเป็น '1' และถูกส่งค่าสู่วงจรชั้นที่ 6 ต่อไป



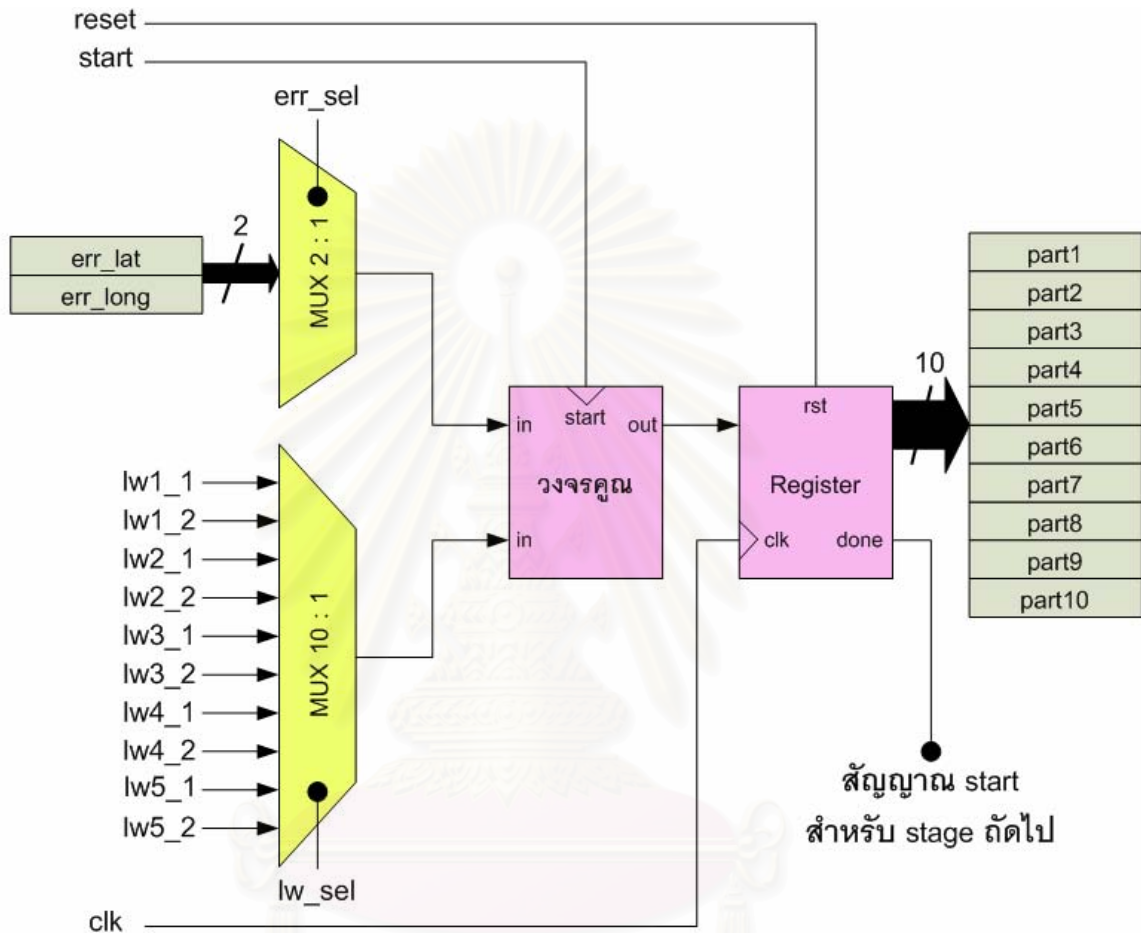
รูปที่ 3.16 วิธีข้อมูลของวงจรชั้นที่ 5

วงจรชั้นที่ 6 (Stage6)

วงจรชั้นที่ 6 เป็นส่วนหนึ่งของวงจรที่ทำหน้าที่คำนวณเพื่อหาค่าถ่วงน้ำหนักและค่าไบแอสใหม่ที่ใช้ในการคำนวณของนิวรอลเน็ตเวิร์กต่อไป เพื่อให้ค่าความผิดพลาดในการประมาณค่าตำแหน่งของสถานีเคลื่อนที่ลดน้อยลง วงจรชั้นที่ 6 จะรับค่าความผิดพลาดจากวงจรชั้นที่ 5 มาคูณกับค่าถ่วงน้ำหนักบางส่วนตามสมการที่ (3.10)

$$err_lat \times \begin{bmatrix} lw1_1 \\ lw2_1 \\ lw3_1 \\ lw4_1 \\ lw5_1 \end{bmatrix} = \begin{bmatrix} part1 \\ part3 \\ part5 \\ part7 \\ part9 \end{bmatrix}, \quad err_long \times \begin{bmatrix} lw1_2 \\ lw2_2 \\ lw3_2 \\ lw4_2 \\ lw5_2 \end{bmatrix} = \begin{bmatrix} part2 \\ part4 \\ part6 \\ part8 \\ part10 \end{bmatrix} \quad (3.10)$$

วงจรรุ่นที่ 6 มีวิธีข้อมูลแสดงได้ดังรูปที่ 3.17 ส่วนวิธีควบคุมของวงจรรุ่นที่ 6 มีลักษณะการออกแบบเช่นเดียวกับวงจรรุ่นที่ 1 เช่นกัน นั่นคือ มีสัญญาณสำหรับเลือกค่าต่าง ๆ เพื่อนำมาคำนวณ เมื่อการคำนวณเสร็จสิ้น สัญญาณ stage6_done จะมีลอจิกเป็น '1' และถูกส่งค่าสู่วงจรรุ่นที่ 7 ต่อไป



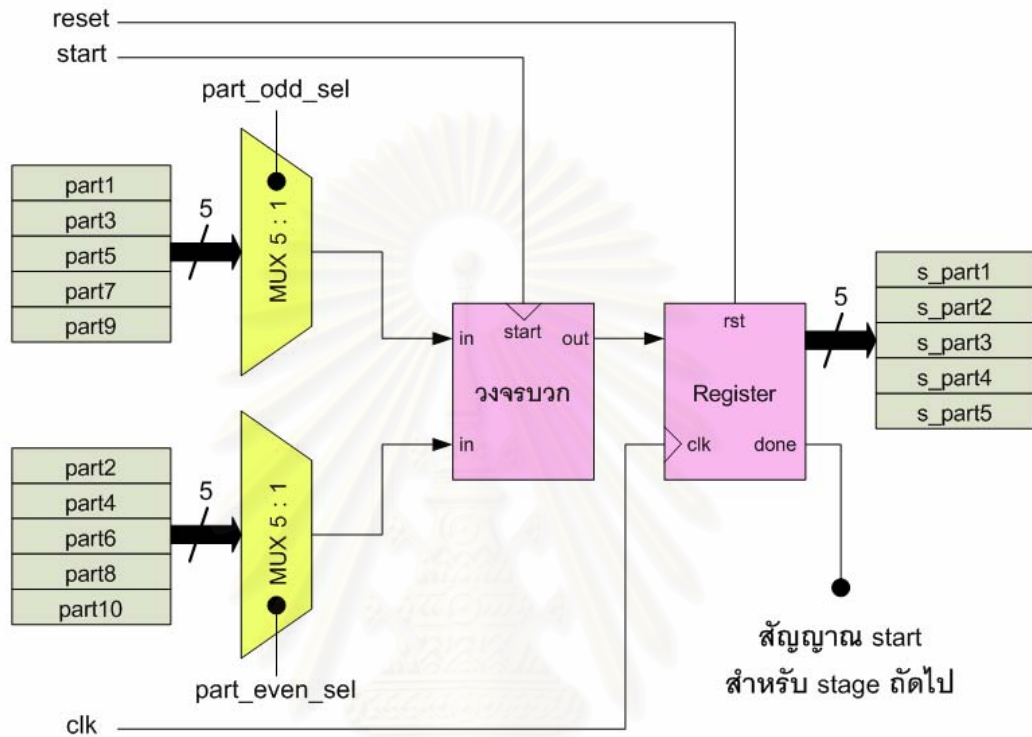
รูปที่ 3.17 วิธีข้อมูลของวงจรรุ่นที่ 6

วงจรรุ่นที่ 7 (Stage7)

วงจรรุ่นที่ 7 มีหน้าที่นำสัญญาณค่าต่าง ๆ ที่ได้จากวงจรรุ่นที่ 6 มาคำนวณตามสมการที่ (3.11)

$$\begin{bmatrix} part1 + part2 \\ part3 + part4 \\ part5 + part6 \\ part7 + part8 \\ part9 + part10 \end{bmatrix} = \begin{bmatrix} s_part1 \\ s_part2 \\ s_part3 \\ s_part4 \\ s_part5 \end{bmatrix} \quad (3.11)$$

วงจรชั้นที่ 7 มีวิธีข้อมูลแสดงได้ดังรูปที่ 3.18 ส่วนวิธีกควบคุมของวงจรชั้นที่ 7 มีลักษณะการออกแบบเช่นเดียวกับวงจรชั้นที่ 1 เช่นกัน นั่นคือ มีสัญญาณสำหรับเลือกค่าต่าง ๆ เพื่อนำมาคำนวณ เมื่อการคำนวณเสร็จสิ้น สัญญาณ stage7_done จะมีลอจิกเป็น '1' และถูกส่งค่าสู่วงจรชั้นที่ 8 ต่อไป



รูปที่ 3.18 วิธีข้อมูลของวงจรชั้นที่ 7

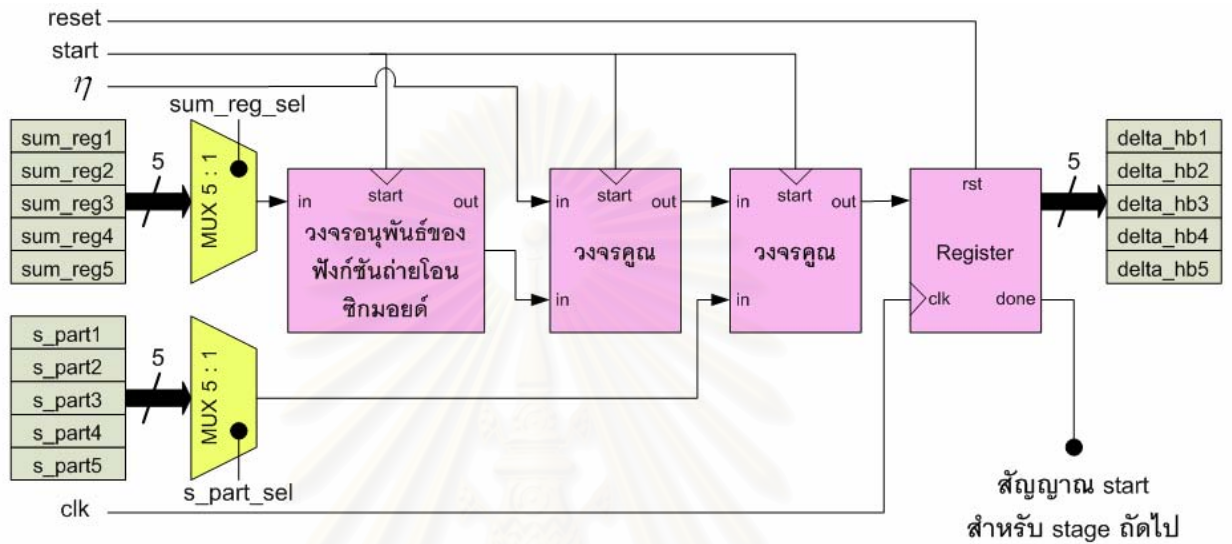
วงจรชั้นที่ 8 (Stage8)

วงจรชั้นที่ 8 มีหน้าที่นำสัญญาณค่าต่าง ๆ จากวงจรชั้นที่ 2 และวงจรชั้นที่ 7 มาคำนวณตามสมการที่ (3.12) และ (3.13)

$$\eta \times f' \begin{bmatrix} \text{sum_reg1} \\ \text{sum_reg2} \\ \text{sum_reg3} \\ \text{sum_reg4} \\ \text{sum_reg5} \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \\ d \\ e \end{bmatrix}, \quad f'(x) = \begin{cases} \frac{1}{2} \left[\frac{1}{(1+x)^2} \right], & x > 0 \\ \frac{1}{2} \left[\frac{1}{(1-x)^2} \right], & x < 0 \end{cases} \quad (3.12)$$

$$\begin{bmatrix} s_part1 \times a \\ s_part2 \times b \\ s_part3 \times c \\ s_part4 \times d \\ s_part5 \times e \end{bmatrix} = \begin{bmatrix} \text{delta_hb1} \\ \text{delta_hb2} \\ \text{delta_hb3} \\ \text{delta_hb4} \\ \text{delta_hb5} \end{bmatrix} \quad (3.13)$$

วงจรชั้นที่ 8 มีวิธีข้อมูลแสดงได้ดังรูปที่ 3.19 ส่วนวิธีควบคุมของวงจรชั้นที่ 8 มีลักษณะการออกแบบเช่นเดียวกับวงจรชั้นที่ 1 เช่นกัน นั่นคือ มีสัญญาณสำหรับเลือกค่าต่าง ๆ เพื่อนำมาคำนวณ เมื่อการคำนวณเสร็จสิ้น สัญญาณ stage8_done จะมีลอจิกเป็น '1' และถูกส่งค่าสู่วงจรชั้นที่ 9 ต่อไป



รูปที่ 3.19 วิธีข้อมูลของวงจรชั้นที่ 8

วงจรชั้นที่ 9 (Stage9)

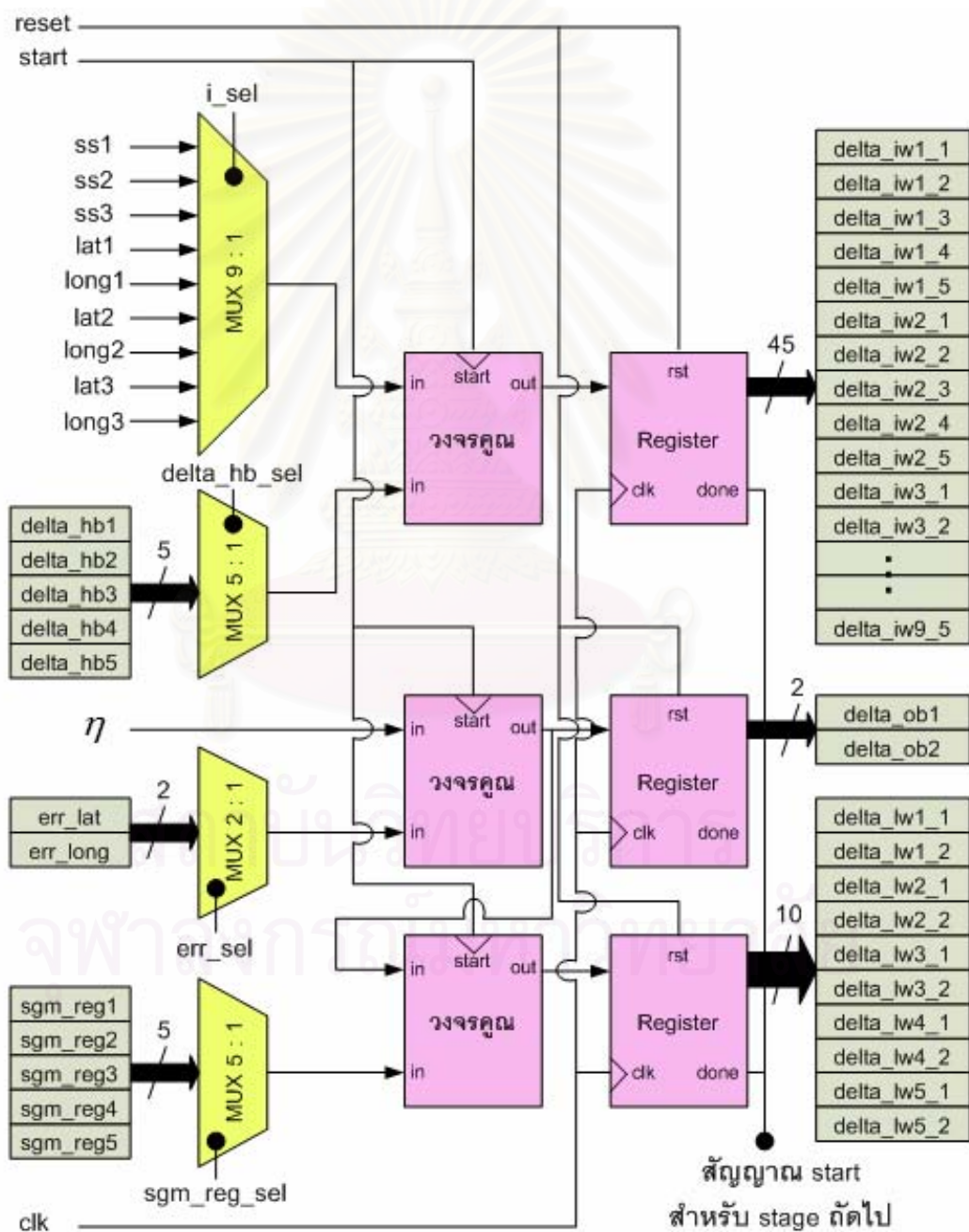
วงจรชั้นที่ 9 มีหน้าที่นำสัญญาณอินพุตทั้งหมด มาคูณกับสัญญาณค่าต่าง ๆ ที่ได้จากวงจรชั้นที่ 8 และนำสัญญาณบางส่วนที่ได้จากวงจรชั้นที่ 2 มาคูณกับค่าความผิดพลาดที่ได้จากวงจรชั้นที่ 5 ตามสมการที่ (3.14) (3.15) และ (3.16)

$$\begin{bmatrix} ss1 \\ ss2 \\ ss3 \\ lat1 \\ long1 \\ lat2 \\ long2 \end{bmatrix} \times \begin{bmatrix} \delta_{hb1} \\ \delta_{hb2} \\ \delta_{hb3} \\ \delta_{hb4} \\ \delta_{hb5} \end{bmatrix}^T = \begin{bmatrix} \delta_{iw1_1} & \delta_{iw1_2} & \delta_{iw1_3} & \delta_{iw1_4} & \delta_{iw1_5} \\ \delta_{iw2_1} & \delta_{iw2_2} & \delta_{iw2_3} & \delta_{iw2_4} & \delta_{iw2_5} \\ \delta_{iw3_1} & \delta_{iw3_2} & \delta_{iw3_3} & \delta_{iw3_4} & \delta_{iw3_5} \\ \delta_{iw4_1} & \delta_{iw4_2} & \delta_{iw4_3} & \delta_{iw4_4} & \delta_{iw4_5} \\ \delta_{iw5_1} & \delta_{iw5_2} & \delta_{iw5_3} & \delta_{iw5_4} & \delta_{iw5_5} \\ \delta_{iw6_1} & \delta_{iw6_2} & \delta_{iw6_3} & \delta_{iw6_4} & \delta_{iw6_5} \\ \delta_{iw7_1} & \delta_{iw7_2} & \delta_{iw7_3} & \delta_{iw7_4} & \delta_{iw7_5} \\ \delta_{iw8_1} & \delta_{iw8_2} & \delta_{iw8_3} & \delta_{iw8_4} & \delta_{iw8_5} \\ \delta_{iw9_1} & \delta_{iw9_2} & \delta_{iw9_3} & \delta_{iw9_4} & \delta_{iw9_5} \end{bmatrix} \quad (3.14)$$

$$\eta \times [err_lat \quad err_long] = [\delta_{ob1} \quad \delta_{ob2}] \quad (3.15)$$

$$\begin{bmatrix} sgm_reg1 \\ sgm_reg2 \\ sgm_reg3 \\ sgm_reg4 \\ sgm_reg5 \end{bmatrix} \times [\delta_{ob1} \quad \delta_{ob2}] = \begin{bmatrix} \delta_{lw1_1} & \delta_{lw1_2} \\ \delta_{lw2_1} & \delta_{lw2_2} \\ \delta_{lw3_1} & \delta_{lw3_2} \\ \delta_{lw4_1} & \delta_{lw4_2} \\ \delta_{lw5_1} & \delta_{lw5_2} \end{bmatrix} \quad (3.16)$$

วงจรชั้นที่ 9 มีวิธีข้อมูลแสดงได้ดังรูปที่ 3.20 ส่วนวิธีกควบคุมของวงจรชั้นที่ 9 มีลักษณะการออกแบบคล้ายกับวงจรชั้นที่ 1 เช่นกัน นั่นคือ มีสัญญาณสำหรับเลือกค่าต่าง ๆ เพื่อนำมาคำนวณ เมื่อการคำนวณเสร็จสิ้น สัญญาณ stage9_done จะมีลอจิกเป็น '1' และถูกส่งค่าสู่วงจรชั้นที่ 10 ต่อไป แต่หากค่าผลลัพธ์ของวงจรมีเท่ากับศูนย์ทั้งหมด ซึ่งหมายความว่าค่าถ่วงน้ำหนักและค่าไบแอสรอบปัจจุบันมีค่าเท่ากับค่าถ่วงน้ำหนักและค่าไบแอสรอบก่อนหน้า สัญญาณ stage9_done จะมีลอจิกเป็น '0' ถือว่าสิ้นสุดการทำงานทั้งหมดของนิวรอนเน็ตเวิร์ก จะไม่มีการส่งค่าใด ๆ ไปยังวงจรชั้นที่ 10



รูปที่ 3.20 วิธีข้อมูลของวงจรชั้นที่ 9

วงจรขั้นที่ 10 (Stage10)

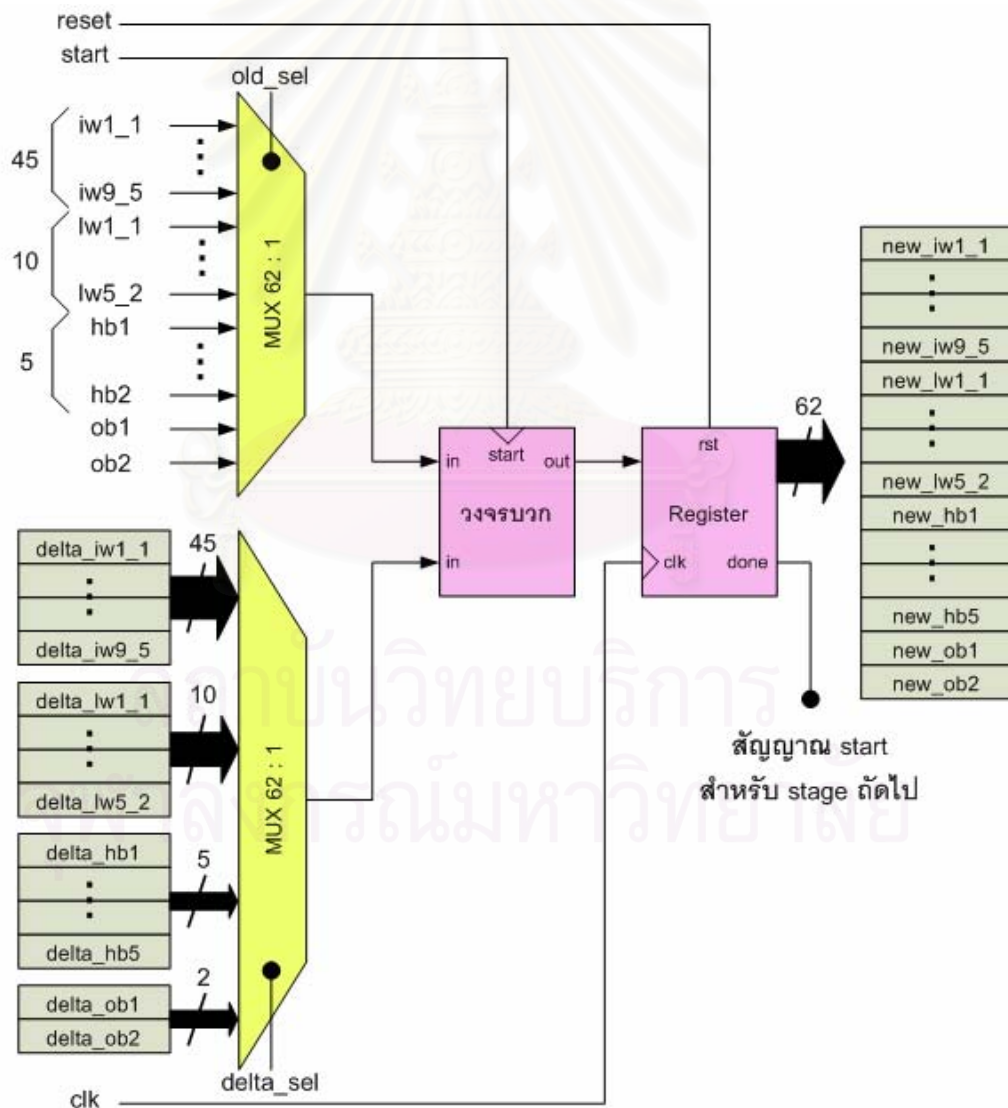
วงจรขั้นที่ 10 มีหน้าที่นำสัญญาณค่าต่าง ๆ จากวงจรขั้นที่ 8 และวงจรขั้นที่ 9 มาบวกกับค่าถ่วงน้ำหนักและค่าไบแอสทั้งหมด เพื่อคำนวณหาค่าถ่วงน้ำหนักและค่าไบแอสใหม่ที่นิรพอลเนตเวิร์กจะใช้ในการคำนวณรอบต่อไปตามสมการที่ (3.17) (3.18) และ (3.19)

$$\begin{aligned}
 & \begin{bmatrix} iw1_1 & iw1_2 & iw1_3 & iw1_4 & iw1_5 \\ iw2_1 & iw2_2 & iw2_3 & iw2_4 & iw2_5 \\ iw3_1 & iw3_2 & iw3_3 & iw3_4 & iw3_5 \\ iw4_1 & iw4_2 & iw4_3 & iw4_4 & iw4_5 \\ iw5_1 & iw5_2 & iw5_3 & iw5_4 & iw5_5 \\ iw6_1 & iw6_2 & iw6_3 & iw6_4 & iw6_5 \\ iw7_1 & iw7_2 & iw7_3 & iw7_4 & iw7_5 \\ iw8_1 & iw8_2 & iw8_3 & iw8_4 & iw8_5 \\ iw9_1 & iw9_2 & iw9_3 & iw9_4 & iw9_5 \end{bmatrix} + \\
 & \begin{bmatrix} \delta_{iw1_1} & \delta_{iw1_2} & \delta_{iw1_3} & \delta_{iw1_4} & \delta_{iw1_5} \\ \delta_{iw2_1} & \delta_{iw2_2} & \delta_{iw2_3} & \delta_{iw2_4} & \delta_{iw2_5} \\ \delta_{iw3_1} & \delta_{iw3_2} & \delta_{iw3_3} & \delta_{iw3_4} & \delta_{iw3_5} \\ \delta_{iw4_1} & \delta_{iw4_2} & \delta_{iw4_3} & \delta_{iw4_4} & \delta_{iw4_5} \\ \delta_{iw5_1} & \delta_{iw5_2} & \delta_{iw5_3} & \delta_{iw5_4} & \delta_{iw5_5} \\ \delta_{iw6_1} & \delta_{iw6_2} & \delta_{iw6_3} & \delta_{iw6_4} & \delta_{iw6_5} \\ \delta_{iw7_1} & \delta_{iw7_2} & \delta_{iw7_3} & \delta_{iw7_4} & \delta_{iw7_5} \\ \delta_{iw8_1} & \delta_{iw8_2} & \delta_{iw8_3} & \delta_{iw8_4} & \delta_{iw8_5} \\ \delta_{iw9_1} & \delta_{iw9_2} & \delta_{iw9_3} & \delta_{iw9_4} & \delta_{iw9_5} \end{bmatrix} \\
 & = \begin{bmatrix} new_iw1_1 & new_iw1_2 & new_iw1_3 & new_iw1_4 & new_iw1_5 \\ new_iw2_1 & new_iw2_2 & new_iw2_3 & new_iw2_4 & new_iw2_5 \\ new_iw3_1 & new_iw3_2 & new_iw3_3 & new_iw3_4 & new_iw3_5 \\ new_iw4_1 & new_iw4_2 & new_iw4_3 & new_iw4_4 & new_iw4_5 \\ new_iw5_1 & new_iw5_2 & new_iw5_3 & new_iw5_4 & new_iw5_5 \\ new_iw6_1 & new_iw6_2 & new_iw6_3 & new_iw6_4 & new_iw6_5 \\ new_iw7_1 & new_iw7_2 & new_iw7_3 & new_iw7_4 & new_iw7_5 \\ new_iw8_1 & new_iw8_2 & new_iw8_3 & new_iw8_4 & new_iw8_5 \\ new_iw9_1 & new_iw9_2 & new_iw9_3 & new_iw9_4 & new_iw9_5 \end{bmatrix} \quad (3.17)
 \end{aligned}$$

$$\begin{aligned}
 & \begin{bmatrix} lw1_1 & lw1_2 \\ lw2_1 & lw2_2 \\ lw3_1 & lw3_2 \\ lw4_1 & lw4_2 \\ lw5_1 & lw5_2 \end{bmatrix} + \begin{bmatrix} \delta_{lw1_1} & \delta_{lw1_2} \\ \delta_{lw2_1} & \delta_{lw2_2} \\ \delta_{lw3_1} & \delta_{lw3_2} \\ \delta_{lw4_1} & \delta_{lw4_2} \\ \delta_{lw5_1} & \delta_{lw5_2} \end{bmatrix} = \begin{bmatrix} new_lw1_1 & new_lw1_2 \\ new_lw2_1 & new_lw2_2 \\ new_lw3_1 & new_lw3_2 \\ new_lw4_1 & new_lw4_2 \\ new_lw5_1 & new_lw5_2 \end{bmatrix} \quad (3.18)
 \end{aligned}$$

$$\begin{bmatrix} hb1 \\ hb2 \\ hb3 \\ hb4 \\ hb5 \\ ob1 \\ ob2 \end{bmatrix} + \begin{bmatrix} \Delta_{hb1} \\ \Delta_{hb2} \\ \Delta_{hb3} \\ \Delta_{hb4} \\ \Delta_{hb5} \\ \Delta_{ob1} \\ \Delta_{ob2} \end{bmatrix} = \begin{bmatrix} new_hb1 \\ new_hb2 \\ new_hb3 \\ new_hb4 \\ new_hb5 \\ new_ob1 \\ new_ob2 \end{bmatrix} \quad (3.19)$$

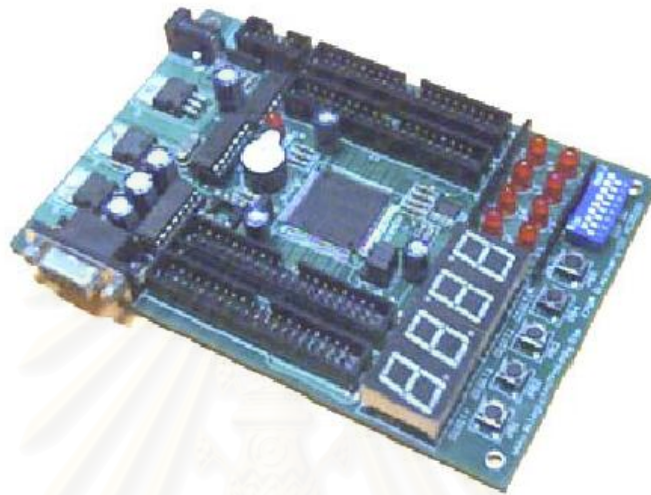
วงจรรุ่นที่ 10 มีวิธีข้อมูลแสดงได้ดังรูปที่ 3.21 ส่วนวิธีควบคุมของวงจรรุ่นที่ 10 มีลักษณะการออกแบบเช่นเดียวกับวงจรรุ่นที่ 1 เช่นกัน นั่นคือ มีสัญญาณสำหรับเลือกค่าต่าง ๆ เพื่อนำมาคำนวณ เมื่อการคำนวณเสร็จสิ้น สัญญาณ stage10_done จะมีลอจิกเป็น '1' และถูกส่งค่าสู่วงจรถูกเลือกต่างหน้าหน้าและค่าไบแอสต่อไป



รูปที่ 3.21 วิธีข้อมูลของวงจรรุ่นที่ 10

3.3.2 การสร้างอุปกรณ์ต้นแบบสำหรับประมาณค่าตำแหน่งของสถานีเคลื่อนที่

เมื่อออกแบบนิรอลเน็ตเวิร์กด้วยภาษาวีเอชดีแอลเรียบร้อยแล้ว วิทยานิพนธ์ฉบับนี้ได้ทดลองสร้างอุปกรณ์ต้นแบบสำหรับประมาณค่าตำแหน่งสถานีเคลื่อนที่โดยการดาวน์โหลดวงจรนิรอลเน็ตเวิร์กที่ออกแบบด้วยภาษาวีเอชดีแอลลงบนบอร์ดเอฟพีจีเอรุ่น Discovery-III XC3S200 ของบริษัท เอเพก อินสตรูเมนต์ จำกัด (รายละเอียดในภาคผนวก) ดังแสดงในรูปที่ 3.22



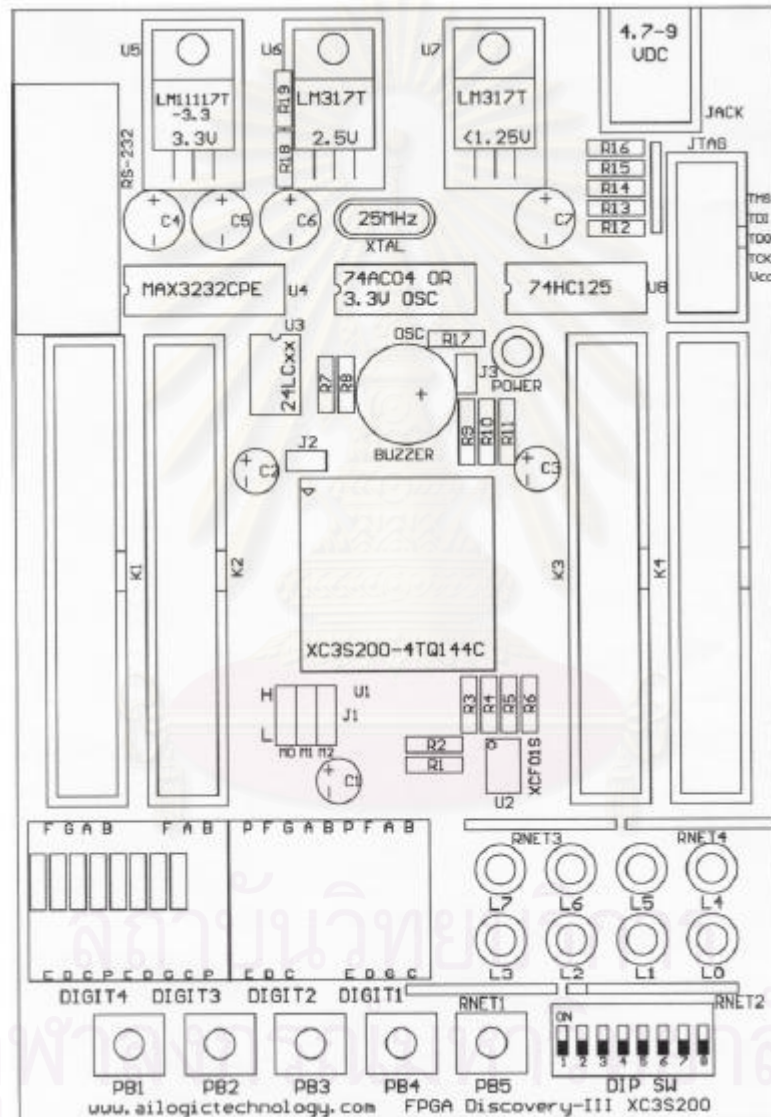
รูปที่ 3.22 บอร์ดเอฟพีจีเอรุ่น Discovery-III XC3S200

รายละเอียดของบอร์ดแสดงดังรูปที่ 3.23 และคุณสมบัติทั่วไปของบอร์ด มีดังต่อไปนี้

- ชิพเอฟพีจีเอตระกูล Spartan-3 ของบริษัท Xilinx เบอร์ XC3S200 ขนาด 200,000 เกต แพ็กเกจแบบ TQ144 Speed Grade: 4
- Platform Flash PROM เบอร์ XCF01S สามารถโปรแกรมข้อมูลวงจรซ้ำได้ 20,000 ครั้ง
- ตัวแสดงผลเจ็ดส่วน (7-Segment) จำนวน 4 หลัก (ถอดออกได้)
- LED จำนวน 8 ดวง (สามารถแยกออกจาก I/O ได้ โดยการถอดหรือหักเอา RNET3 และ RNET4 ออก)
- ออกความถี่เสียง (Buzzer) จำนวน 1 ตัว
- สวิตช์กดติด (Push Button Switch) จำนวน 5 ตัว สวิตช์เลื่อน (DIP Switch) 8 บิต 1 ตัว
- พอร์ตใช้เชื่อมต่อสัญญาณไปยังบอร์ดหรืออุปกรณ์ภายนอก (Expansion Port) ที่มี I/O เป็น 3.3 โวลต์ จำนวน 4 พอร์ต 80 บิต
- พอร์ต RS-232C จำนวน 1 พอร์ต
- 12C Socket สำหรับ EEPROM
- ออกสวิตช์เลเตอร์ 25 MHz (เปลี่ยนเป็นความถี่อื่น ๆ ได้โดยใช้ Digital Frequency Synthesizer ที่มีอยู่ในเอฟพีจีเอ)

คุณสมบัติของซีพียูฟิจีเอตตระกูล Spartan-3 เบอร์ XC3S200 สรุปได้ดังต่อไปนี้

- ความจุวงจร 200,000 เกต
- 18 Kb block RAMs จำนวน 12 ชุด (รวม 216 Kb)
- 18x18 ฮาร์ดแวร์มัลติพลายเออร์ (Hardware Multiplier) จำนวน 12 ชุด
- Digital Clock Manager (DCM) จำนวน 12 ชุด
- Digitally Controlled Impedance (DCI)



รูปที่ 3.23 รายละเอียดของบอร์ดเฟฟิจีเอรุ่น Discovery-III XC3S200

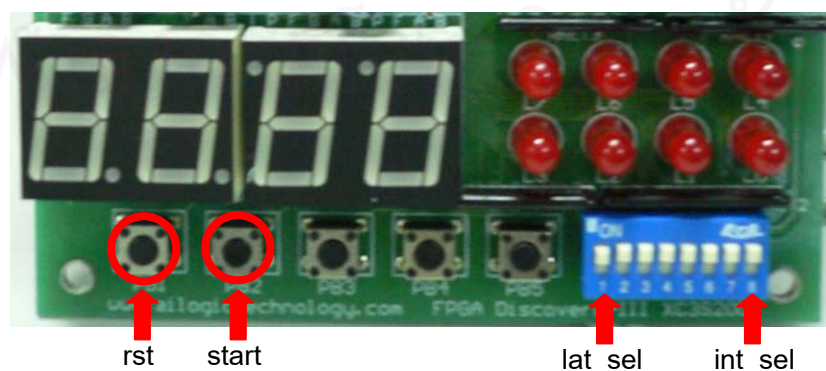
วิทยานิพนธ์ฉบับนี้ ได้ทดลองดาวน์โหลดวงจรนิรลเนตเวิร์กที่ออกแบบเพื่อประมาณค่าตำแหน่งของสถานีเคลื่อนที่โดยใช้ภาษาวีเอชดีแอล ดังที่ได้เสนอไว้ในบทที่ 3 ลงบนบอร์ดเฟฟิจีเอรุ่น Discovery-III XC3S200 ของบริษัท เอเพก อินสตรูเมนต์ จำกัด อย่างไรก็ตาม วงจรนิรลเนต

เวิร์กที่ถูกดาวน์โหลดนั้น เป็นเพียงส่วนหนึ่งของวงจรทั้งหมด เนื่องจากวงจรทั้งหมดมีขนาดใหญ่ มาก จำเป็นต้องใช้ความจุของบอร์ด (จำนวนลอจิกเกต จำนวนขาอินพุตและเอาต์พุต ฯลฯ) จำนวนมาก ซึ่งในปัจจุบันในประเทศไทยไม่มีบอร์ดสำเร็จรูปที่สามารถรองรับวงจรขนาดใหญ่เช่นนี้ได้ วิทยานิพนธ์ฉบับนี้ จึงได้นำเอาวงจรบางส่วนของนิรอลเน็ตเวิร์กมาทดสอบบนบอร์ดดังกล่าว เพื่อเป็นการยืนยันในระดับหนึ่งว่านิรอลเน็ตเวิร์กสามารถทำงานบนบอร์ดได้จริง ส่วนวงจรมินิรอลเน็ตเวิร์กทั้งหมดจะได้มีการพัฒนาและศึกษาเพิ่มเติมเพื่อออกแบบให้วงจรมีขนาดเล็กกลง และเลือกใช้บอร์ดที่มีความจุและราคาที่เหมาะสมต่อไปในอนาคต

วงจรส่วนที่นำมาดาวน์โหลดลงบนบอร์ดเฟฟพีจีเอ ได้แก่ วงจรขั้นที่ 1 วงจรขั้นที่ 2 วงจรขั้นที่ 3 และวงจรขั้นที่ 4 โดยไม่รวมเอาวงจรส่วนสำหรับปรับค่าถ่วงน้ำหนักและค่าไบแอส ซึ่งได้แก่ วงจรเลือกค่าถ่วงน้ำหนักและค่าไบแอส วงจรขั้นที่ 5 วงจรขั้นที่ 6 วงจรขั้นที่ 7 วงจรขั้นที่ 8 วงจรขั้นที่ 9 และวงจรขั้นที่ 10 (รายละเอียดของวงจรแต่ละวงจรถูกกล่าวไว้แล้วในบทที่ 3) กล่าวคือ อุปกรณ์ต้นแบบจะสามารถประมาณค่าตำแหน่งของสถานีได้ก็ต่อเมื่อได้มีการฝึกนิรอลเน็ตเวิร์กในการจำลองแบบ เพื่อนำค่าถ่วงน้ำหนักและค่าไบแอสมาใช้

อุปกรณ์ต้นแบบที่ทดลองสร้างขึ้นมีส่วนแสดงผลเป็นตัวแสดงผลเจ็ดส่วน (7-Segment) 4 หลัก และมีสวิตช์สำหรับควบคุมการทำงาน 4 สวิตช์ ดังแสดงในรูปที่ 3.24

- สวิตช์กดติด rst มีหน้าที่สำหรับสั่งให้อุปกรณ์ต้นแบบปรับตั้งค่าพารามิเตอร์ต่าง ๆ ใหม่ทั้งหมด
- สวิตช์กดติด start มีหน้าที่สำหรับสั่งให้อุปกรณ์ต้นแบบเริ่มต้นการประมาณค่าตำแหน่งของสถานีเคลื่อนที่
- สวิตช์เลื่อน lat_sel มีหน้าที่สำหรับสั่งให้อุปกรณ์ต้นแบบแสดงค่าตำแหน่งละติจูดเมื่อสวิตช์อยู่ที่ตำแหน่ง ON และแสดงค่าตำแหน่งลองจิจูดเมื่อสวิตช์อยู่ที่ตำแหน่ง OFF
- สวิตช์เลื่อน int_sel มีหน้าที่สำหรับสั่งให้อุปกรณ์ต้นแบบแสดงค่าจำนวนเต็มเมื่อสวิตช์อยู่ที่ตำแหน่ง ON และแสดงค่าจำนวนทศนิยมเมื่อสวิตช์อยู่ที่ตำแหน่ง OFF



รูปที่ 3.24 ส่วนควบคุมและส่วนแสดงผลของอุปกรณ์ต้นแบบ

สำหรับการกำหนดขาของชิพเฟลพฟี่จีเอเพื่อให้อุปกรณ์ต้นแบบสามารถทำงานได้ตามที่
ออกแบบไว้ แสดงดังตารางที่ 3.1 ตารางที่ 3.2 ตารางที่ 3.3 และตารางที่ 3.4

ตารางที่ 3.1 การกำหนดขาของชิพเฟลพฟี่จีเอเข้ากับสัญญาณนาฬิกา

Oscillator	FPGA Pinout	Description
OSC	p127	25 MHz, GCLK6

ตารางที่ 3.2 การกำหนดขาของชิพเฟลพฟี่จีเอเข้ากับสวิตช์ rst และสวิตช์ start

Push Button	FPGA Pinout	Neural Network Input
PB1	p44	rst
PB2	p46	start

ตารางที่ 3.3 การกำหนดขาของชิพเฟลพฟี่จีเอเข้ากับสวิตช์ lat_sel และสวิตช์ int_sel

Dip SW	FPGA Pinout	Neural Network Input
1	p52	lat_sel
2	p68	int_sel

ตารางที่ 3.4 การกำหนดขาของชิพเฟลพฟี่จีเอเข้ากับตัวแสดงผลเจ็ดส่วน

7-Segment	FPGA Pinout	Description
a	p40	a
b	p35	b
c	p32	c
d	p30	d
e	p27	e
f	p25	f
g	p23	g
DG1	p31	DIGIT1, COMMON CATHODE
DG2	p33	DIGIT2, COMMON CATHODE
DG3	p36	DIGIT3, COMMON CATHODE
DG4	p41	DIGIT4, COMMON CATHODE

บทที่ 4

ผลการทดสอบ

ในบทนี้จะแสดงผลการทดสอบและการวิเคราะห์ประสิทธิภาพการจำลองแบบของนิรอลเนตเวิร์กที่ได้นำเสนอในบทที่ 3 ในการประมาณค่าตำแหน่งของสถานีเคลื่อนที่ โดยข้อมูลที่ใช้ฝึกและทดสอบนิรอลเนตเวิร์กเน้นไปที่ข้อมูลจริงที่ได้มาจากการเก็บข้อมูลภาคสนามเพื่อทำให้นิรอลเนตเวิร์กได้เรียนรู้และสามารถนำไปใช้ในทางปฏิบัติได้ นอกจากนี้ยังได้จำลองข้อมูลลักษณะอื่นขึ้น เพื่อวิเคราะห์ผลการทดสอบผลที่เกิดขึ้นหากมีการเปลี่ยนแปลงรูปแบบของข้อมูล พร้อมทั้งเปรียบเทียบกับ การประมาณค่าตำแหน่งของสถานีเคลื่อนที่ด้วยนิรอลเนตเวิร์กโดยใช้กรรมวิธีอื่น รวมถึงการทดสอบอุปกรณ์ต้นแบบเพื่อประมาณค่าตำแหน่งของสถานีเคลื่อนที่ที่ทดลองสร้างขึ้น

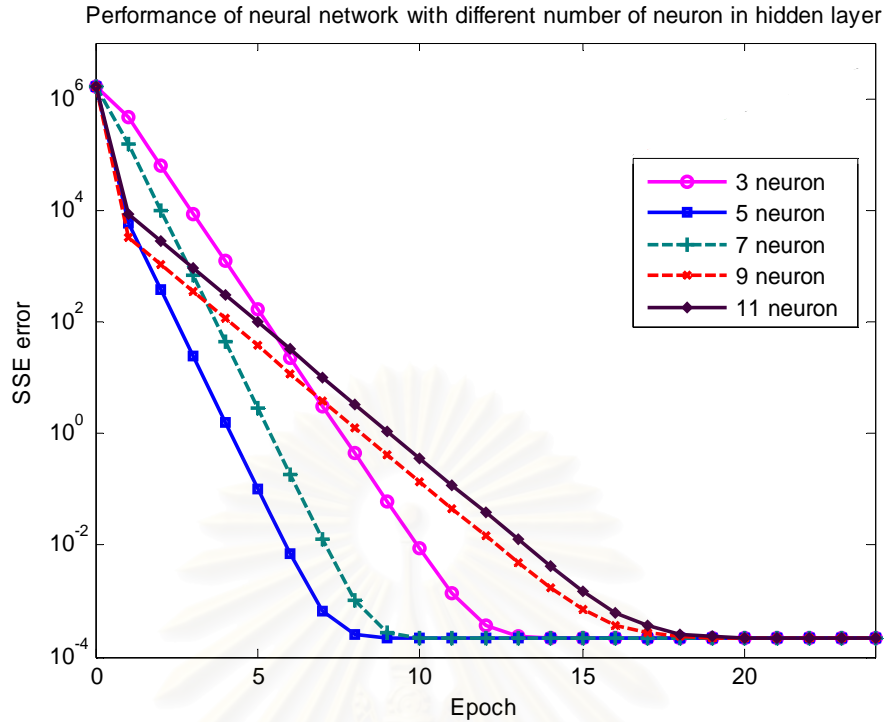
กำหนดสัญลักษณ์ที่ใช้ในรูปและตารางต่าง ๆ ดังนี้

ss1	แทนค่าความแรงของสัญญาณที่สถานีฐานสถานีที่ 1 รับได้จากสถานีเคลื่อนที่ที่พิจารณา มีหน่วยเป็นเดซิเบล (dB)
ss2	แทนค่าความแรงของสัญญาณที่สถานีฐานสถานีที่ 2 รับได้จากสถานีเคลื่อนที่ที่พิจารณา มีหน่วยเป็นเดซิเบล (dB)
ss3	แทนค่าความแรงของสัญญาณที่สถานีฐานสถานีที่ 3 รับได้จากสถานีเคลื่อนที่ที่พิจารณา มีหน่วยเป็นเดซิเบล (dB)
lat1	แทนตำแหน่งละติจูดของสถานีฐานสถานีที่ 1 มีหน่วยเป็นองศาเหนือ
long1	แทนตำแหน่งลองจิจูดของสถานีฐานสถานีที่ 1 มีหน่วยเป็นองศาตะวันออก
lat2	แทนตำแหน่งละติจูดของสถานีฐานสถานีที่ 2 มีหน่วยเป็นองศาเหนือ
long2	แทนตำแหน่งลองจิจูดของสถานีฐานสถานีที่ 2 มีหน่วยเป็นองศาตะวันออก
lat3	แทนตำแหน่งละติจูดของสถานีฐานสถานีที่ 3 มีหน่วยเป็นองศาเหนือ
long3	แทนตำแหน่งลองจิจูดของสถานีฐานสถานีที่ 3 มีหน่วยเป็นองศาตะวันออก
tr_lat	แทนค่าเป้าหมายของตำแหน่งละติจูดของสถานีเคลื่อนที่ที่พิจารณา มีหน่วยเป็นองศาเหนือ
tr_long	แทนค่าเป้าหมายของตำแหน่งลองจิจูดของสถานีเคลื่อนที่ที่พิจารณา มีหน่วยเป็นองศาตะวันออก
m_lat	แทนตำแหน่งละติจูดของสถานีเคลื่อนที่ที่พิจารณาที่ได้จากการประมาณค่าของนิรอลเนตเวิร์กที่ออกแบบ มีหน่วยเป็นองศาเหนือ

m_long	แทนตำแหน่งละติจูดของสถานีเคลื่อนที่ที่พิจารณาที่ได้จากการประมาณค่าของนิวรอลเน็ตเวิร์กที่ออกแบบ มีหน่วยเป็นองศาตะวันออก
sign_tr_lat	แทนเครื่องหมายของค่าเป้าหมายของตำแหน่งละติจูดของสถานีเคลื่อนที่
int_tr_lat	แทนจำนวนเต็มของค่าเป้าหมายของตำแหน่งละติจูดของสถานีเคลื่อนที่
frac_tr_lat	แทนจำนวนทศนิยมของค่าเป้าหมายของตำแหน่งละติจูดของสถานีเคลื่อนที่
sign_tr_long	แทนเครื่องหมายของค่าเป้าหมายของตำแหน่งลองจิจูดของสถานีเคลื่อนที่
int_tr_long	แทนจำนวนเต็มของค่าเป้าหมายของตำแหน่งลองจิจูดของสถานีเคลื่อนที่
frac_tr_long	แทนจำนวนทศนิยมของค่าเป้าหมายของตำแหน่งลองจิจูดของสถานีเคลื่อนที่
sign_m_lat	แทนเครื่องหมายของตำแหน่งละติจูดของสถานีเคลื่อนที่ที่ได้จากการประมาณค่าของนิวรอลเน็ตเวิร์กที่ออกแบบ
int_m_lat	แทนจำนวนเต็มของตำแหน่งละติจูดของสถานีเคลื่อนที่ที่ได้จากการประมาณค่าของนิวรอลเน็ตเวิร์กที่ออกแบบ
frac_m_lat	แทนจำนวนทศนิยมของตำแหน่งละติจูดของสถานีเคลื่อนที่ที่ได้จากการประมาณค่าของนิวรอลเน็ตเวิร์กที่ออกแบบ
sign_m_long	แทนเครื่องหมายของตำแหน่งลองจิจูดของสถานีเคลื่อนที่ที่ได้จากการประมาณค่าของนิวรอลเน็ตเวิร์กที่ออกแบบ
int_m_long	แทนจำนวนเต็มของตำแหน่งลองจิจูดของสถานีเคลื่อนที่ที่ได้จากการประมาณค่าของนิวรอลเน็ตเวิร์กที่ออกแบบ
frac_m_long	แทนจำนวนทศนิยมของตำแหน่งลองจิจูดของสถานีเคลื่อนที่ที่ได้จากการประมาณค่าของนิวรอลเน็ตเวิร์กที่ออกแบบ

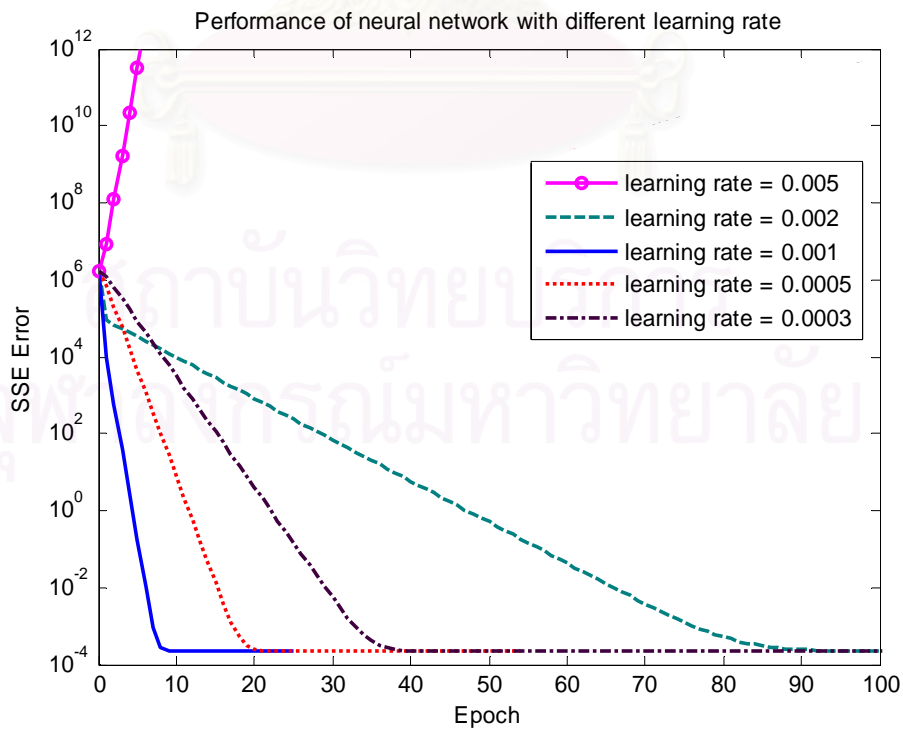
4.1 ผลการทดสอบเพื่อหาจำนวนนิวรอนในชั้นซ่อนเร้นและค่าอัตราการเรียนรู้ที่เหมาะสม

พารามิเตอร์สำคัญที่มีส่วนเกี่ยวข้องกับการลู่เข้าในการฝึกนิวรอลเน็ตเวิร์ก ได้แก่ จำนวนนิวรอนในชั้นซ่อนเร้นของนิวรอลเน็ตเวิร์ก และค่าอัตราการเรียนรู้ของนิวรอลเน็ตเวิร์ก ซึ่งในวิทยานิพนธ์ฉบับนี้ได้ทดลองใช้จำนวนนิวรอน และอัตราการเรียนรู้ของนิวรอลเน็ตเวิร์กที่แตกต่างกันหลาย ๆ กรณี เพื่อทดสอบหาจำนวนนิวรอนในชั้นซ่อนเร้นและค่าอัตราการเรียนรู้ที่เหมาะสมที่สุดกับระบบที่จะนำนิวรอลเน็ตเวิร์กไปประยุกต์ใช้งาน โดยการทดสอบจะใช้ชุดข้อมูลภาคสนามจำนวน 100 ชุด เป็นชุดข้อมูลสำหรับทดสอบ และแบ่งการทดสอบออกเป็นสองส่วน ได้แก่ ส่วนแรก เป็นการทดสอบหาจำนวนนิวรอนในชั้นซ่อนเร้นที่เหมาะสม ผลการทดสอบแสดงได้ดังรูปที่ 4.1 และส่วนที่สอง เป็นการหาค่าอัตราการเรียนรู้ที่เหมาะสม ผลการทดสอบแสดงได้ดังรูปที่ 4.2



รูปที่ 4.1 ผลการทดสอบการลู่เข้าของนิวรอลเน็ตเวิร์กที่มีจำนวนนิวรอนในชั้นซ่อนเร้นไม่เท่ากัน

จากรูปที่ 4.1 จะเห็นว่าการเปลี่ยนจำนวนนิวรอนในชั้นซ่อนเร้นไม่ส่งผลต่อค่าความผิดพลาดในการประมาณค่าของนิวรอลเน็ตเวิร์ก แต่จะส่งผลต่อการลู่เข้าของนิวรอลเน็ตเวิร์ก โดยจำนวนนิวรอนในชั้นซ่อนเร้นเท่ากับ 5 จะทำให้นิวรอลเน็ตเวิร์กเกิดการลู่เข้าที่เร็วที่สุด



รูปที่ 4.2 ผลการทดสอบการลู่เข้าของนิวรอลเน็ตเวิร์กที่มีค่าอัตราการเรียนรู้ไม่เท่ากัน

ในทางทฤษฎี ค่าอัตราการเรียนรู้จะมีค่าอยู่ระหว่าง 0 ถึง 1 หากค่าอัตราการเรียนรู้ยิ่งมีค่ามากจะทำให้นิรอลเน็ตเวิร์กเกิดการลู่เข้าได้เร็ว แต่ถ้าค่าอัตราการเรียนรู้สูงเกินไปอาจทำให้นิรอลเน็ตเวิร์กลู่ออก หรือลู่เข้าช้ากว่าที่ควรจะเป็น จากรูปที่ 4.2 จะเห็นว่ากรณีที่ค่าอัตราการเรียนรู้เท่ากับ 0.005 จะทำให้นิรอลเน็ตเวิร์กเกิดการลู่ออก และค่าอัตราการเรียนรู้ที่ทำให้นิรอลเน็ตเวิร์กลู่เข้าเร็วที่สุดมีค่าเท่ากับ 0.001

จากรูปที่ 4.1 และ 4.2 แสดงให้เห็นว่าสำหรับการประมาณค่าตำแหน่งของสถานีเคลื่อนที่ในกรณีที่ใช้ข้อมูลภาคสนามนี้ จำนวนนิรอลในชั้นซ่อนเร้นและค่าอัตราการเรียนรู้ที่เหมาะสมมีค่าเท่ากับ 5 และ 0.001 ตามลำดับ ดังนั้นในวิทยานิพนธ์ฉบับนี้ จึงออกแบบให้นิรอลเน็ตเวิร์กมีจำนวนนิรอลในชั้นซ่อนเร้นเท่ากับ 5 และค่าอัตราการเรียนรู้เท่ากับ 0.001

สำหรับการจำลองแบบที่ใช้ทดสอบประสิทธิภาพของนิรอลเน็ตเวิร์กที่ออกแบบในการประมาณค่าตำแหน่งของสถานีเคลื่อนที่ มีรายละเอียดดังต่อไปนี้

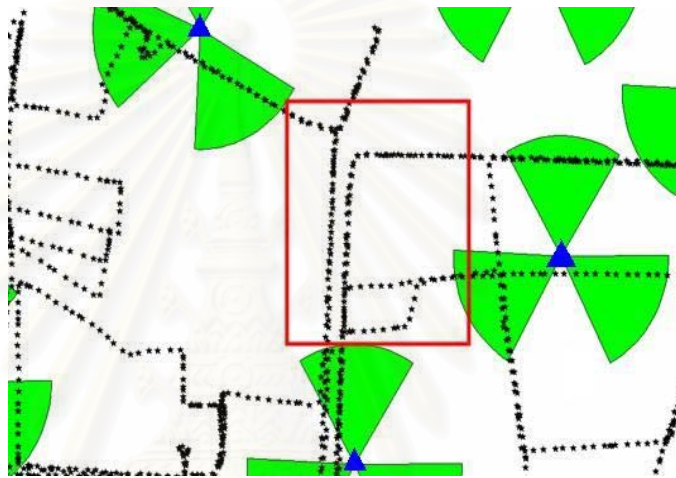
- นิรอลเน็ตเวิร์กที่ออกแบบมีสถาปัตยกรรมแบบป้อนไปข้างหน้าและมีการแพร่กระจายย้อนกลับ โดยมีชั้นข้อมูลเข้า 1 ชั้น จำนวนนิรอลในชั้นนี้เท่ากับ 9 นิรอลตามจำนวนของข้อมูลเข้าที่ประกอบด้วยเป็นค่าข้อมูลความแรงของสัญญาณที่สถานีฐาน 3 สถานีรับได้จากสถานีเคลื่อนที่ที่พิจารณา (ss1, ss2 และ s3) และตำแหน่งละติจูด - ลองจิจูดของสถานีฐานที่ 1, 2, และ 3 ตามลำดับ (lat1, long1, lat2, long2, lat3 และ long3) ฟังก์ชันถ่ายโอนในชั้นนี้เป็นฟังก์ชันถ่ายโอนเชิงเส้น
- นิรอลเน็ตเวิร์กที่ออกแบบมีชั้นซ่อนเร้น 1 ชั้น จำนวนนิรอลในชั้นนี้เท่ากับ 5 นิรอล ฟังก์ชันถ่ายโอนในชั้นนี้เป็นฟังก์ชันถ่ายโอนลอกการิทึมซิกมอยด์
- นิรอลเน็ตเวิร์กที่ออกแบบมีชั้นข้อมูลออก 1 ชั้น จำนวนนิรอลในชั้นนี้เท่ากับ 2 นิรอลตามจำนวนของข้อมูลออกที่ประกอบด้วยค่าตำแหน่งละติจูด - ลองจิจูดของสถานีเคลื่อนที่ที่พิจารณา (m_lat และ m_long) ฟังก์ชันถ่ายโอนในชั้นนี้เป็นฟังก์ชันถ่ายโอนเชิงเส้น

4.2 ผลการทดสอบนิรอลเน็ตเวิร์กที่ออกแบบด้วยข้อมูลภาคสนาม

การฝึกและทดสอบนิรอลเน็ตเวิร์กที่ออกแบบโดยใช้ข้อมูลภาคสนามมีรายละเอียดดังต่อไปนี้

- 1) ชุดข้อมูลภาคสนามที่ใช้สำหรับฝึกและทดสอบนิรอลเน็ตเวิร์กทั้งค่าความแรงของสัญญาณที่สถานีฐาน 3 สถานีรับได้จากสถานีเคลื่อนที่ที่พิจารณาและตำแหน่งของสถานีฐานที่ติดต่อกับสถานีเคลื่อนที่ เป็นข้อมูลจริงที่ได้รับความอนุเคราะห์มาจากบริษัท แอดวานซ์ อินโฟ เซอร์วิส จำกัด (มหาชน) ซึ่งเป็นข้อมูลที่บริษัทเก็บรวบรวม

ขณะมีการเคลื่อนที่ไปตามถนนในเขตกรุงเทพมหานครดังแสดงในรูปที่ 4.3 โดยที่เส้นจุดสีดำที่บแทนแนวการเคลื่อนที่ที่สถานีเคลื่อนที่เคลื่อนที่ผ่าน รูปสามเหลี่ยมสีน้ำเงินแทนสถานีฐาน และบริเวณภายในสี่เหลี่ยมสีแดง คือ ตำแหน่งของสถานีเคลื่อนที่ที่เหมาะสมสำหรับนำมาฝึกและทดสอบนิรอลเน็ตเวิร์ก โดยหลักการในการเลือกตำแหน่ง คือ เลือกตำแหน่งของสถานีเคลื่อนที่ที่มีข้อมูลความแรงสัญญาณและตำแหน่งแหล่งละติจูด-ลองจิจูดของสถานีฐานครบทั้ง 3 สถานีฐานที่กำลังพิจารณา ซึ่งหลังจากเลือกตำแหน่งของสถานีเคลื่อนที่ที่เหมาะสมแล้ว จะได้รูปแบบของชุดข้อมูลที่จำเป็นในการฝึกและทดสอบนิรอลเน็ตเวิร์กดังตารางที่ 4.1

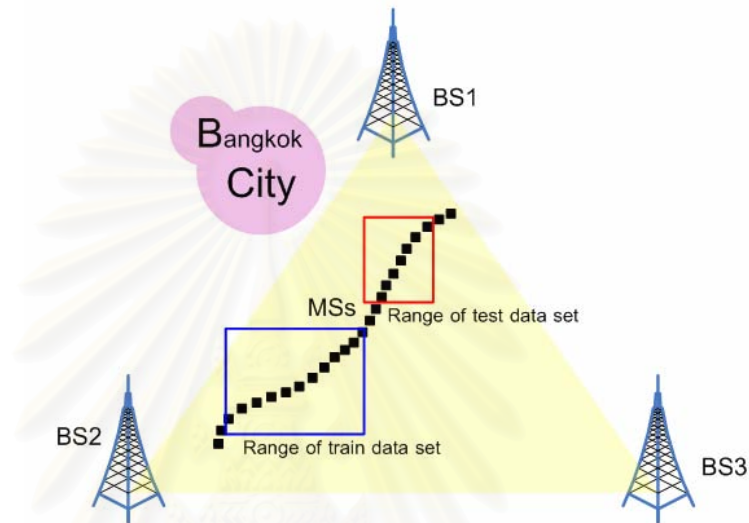


รูปที่ 4.3 ตำแหน่งการเคลื่อนที่ของสถานีเคลื่อนที่ที่เหมาะสม

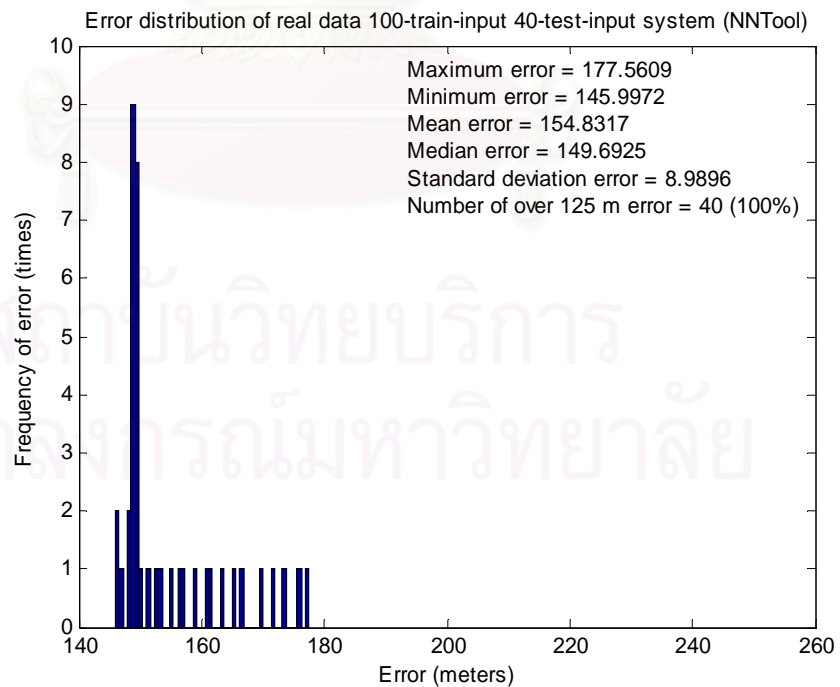
ตารางที่ 4.1 รูปแบบของชุดข้อมูลภาคสนามสำหรับฝึกและทดสอบนิรอลเน็ตเวิร์ก

ข้อมูลเข้า	ชุดข้อมูลที่						
	1	2	3	4	5	6	...
ss1	-72	-72	-73	-77	-82	-83	.
ss2	-63	-67	-71	-73	-76	-74	.
ss3	-78	-76	-78	-77	-80	-81	.
lat1	18.787	18.787	18.787	18.787	18.787	18.787	
long1	98.987	98.987	98.987	98.987	98.987	98.987	
lat2	16.936	16.936	18.797	16.936	16.936	16.936	
long2	100.35	100.35	98.987	100.35	100.35	100.35	
lat3	18.792	18.797	16.936	18.797	15.02	15.02	
long3	98.993	98.987	100.35	98.987	99.62	99.62	

หลังจากเลือกตำแหน่งของสถานีเคลื่อนที่ที่เหมาะสม และได้ชุดข้อมูลสำหรับฝึกและทดสอบนิรวลเน็ตเวิร์กแล้ว สิ่งที่ต้องพิจารณาอีกประการหนึ่ง ได้แก่ การเลือกขอบเขตของชุดข้อมูลที่นำมาฝึกและทดสอบนิรวลเน็ตเวิร์ก หากชุดข้อมูลสำหรับฝึกและทดสอบนิรวลเน็ตเวิร์กไม่ได้อยู่ในอาณาบริเวณเดียวกัน ดังแสดงในรูปที่ 4.4 จะทำให้นิรวลเน็ตเวิร์กไม่สามารถประมาณค่าตำแหน่งของสถานีเคลื่อนที่ได้ถูกต้อง ดังแสดงในรูปที่ 4.5



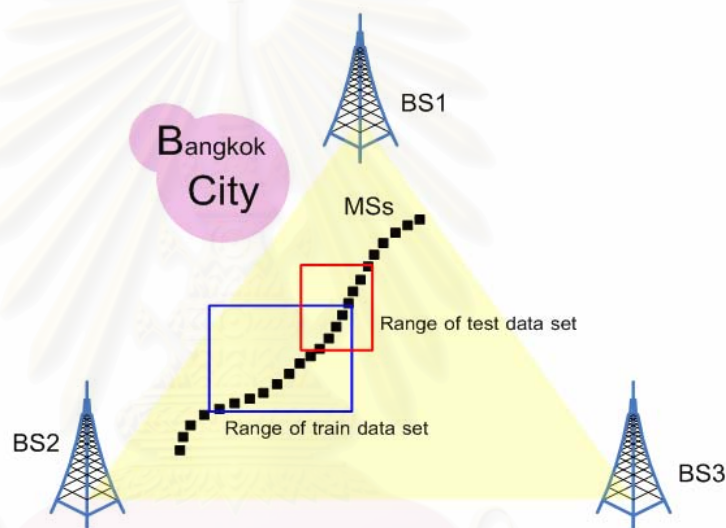
รูปที่ 4.4 ชุดข้อมูลสำหรับฝึกและทดสอบนิรวลเน็ตเวิร์กไม่อยู่ในบริเวณเดียวกัน



รูปที่ 4.5 ค่าความผิดพลาดของการประมาณค่าเมื่อชุดข้อมูลสำหรับฝึกและทดสอบนิรวลเน็ตเวิร์กไม่อยู่ในบริเวณเดียวกัน

จากรูปที่ 4.5 ซึ่งเป็นการทดสอบความสามารถในการประมาณค่าของนิรพลเน็ตเวิร์ก เมื่อชุดข้อมูลสำหรับฝึกและทดสอบนิรพลเน็ตเวิร์กไม่ได้อยู่ในบริเวณเดียวกัน โดยใช้ชุดข้อมูลภาคสนามจำนวน 100 ชุดสำหรับฝึกนิรพลเน็ตเวิร์ก และ 40 ชุดสำหรับทดสอบนิรพลเน็ตเวิร์ก ผลปรากฏว่านิรพลเน็ตเวิร์กไม่สามารถประมาณค่าตำแหน่งของสถานีได้อย่างถูกต้องตามข้อกำหนดของ FCC กล่าวคือ ค่าความผิดพลาดในการประมาณค่าทุกครั้งมากกว่า 125 เมตร

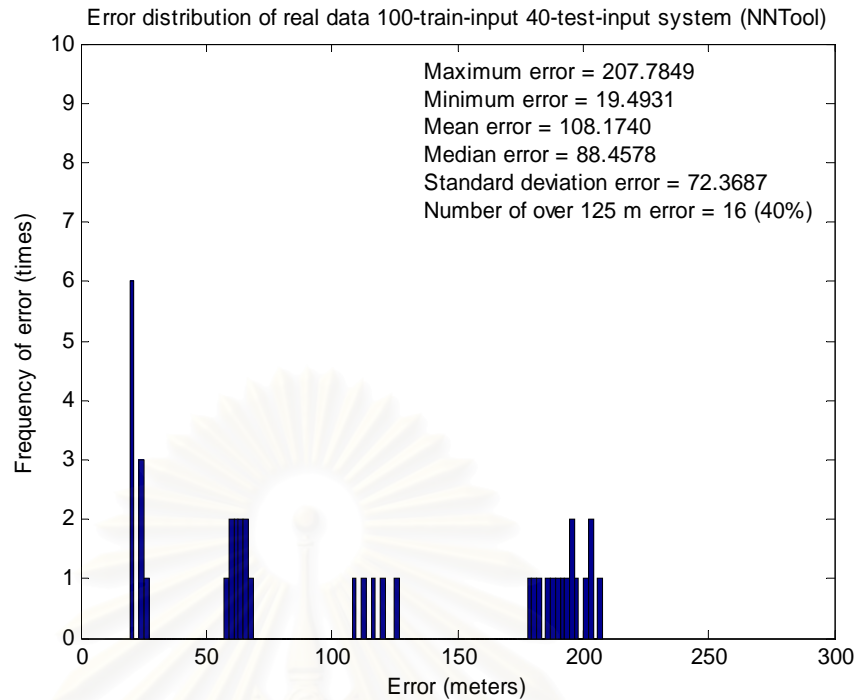
เพื่อให้เกิดความชัดเจนมากยิ่งขึ้น จึงได้ลองทดสอบอีกกรณีหนึ่ง กล่าวคือ เลือกชุดข้อมูลภาคสนามสำหรับทดสอบนิรพลเน็ตเวิร์กบางส่วนอยู่ในบริเวณเดียวกันกับชุดข้อมูลสำหรับฝึกนิรพลเน็ตเวิร์ก ดังแสดงในรูปที่ 4.6



รูปที่ 4.6 ชุดข้อมูลสำหรับฝึกและทดสอบนิรพลเน็ตเวิร์กบางส่วน
ไม่อยู่ในบริเวณเดียวกัน

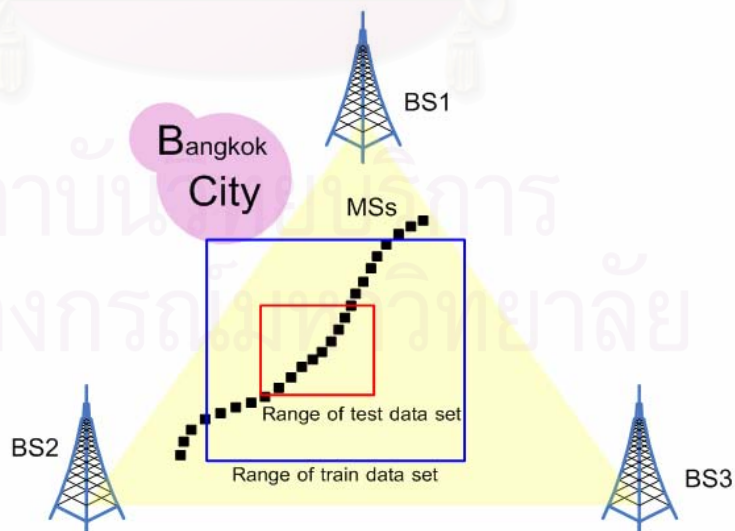
ผลการทดสอบความสามารถในการประมาณค่าของนิรพลเน็ตเวิร์กในกรณีนี้แสดงได้ดังรูปที่ 4.7 ซึ่งเป็นการทดสอบโดยใช้ข้อมูลภาคสนามจำนวน 100 ชุดสำหรับฝึกนิรพลเน็ตเวิร์ก และ 40 ชุดสำหรับทดสอบนิรพลเน็ตเวิร์ก โดยชุดข้อมูลสำหรับทดสอบนิรพลเน็ตเวิร์กนั้นประกอบด้วย ชุดข้อมูลที่อยู่ในบริเวณเดียวกันกับชุดข้อมูลสำหรับฝึกจำนวน 25 ชุด ชุดข้อมูลที่ไม่ได้อยู่ในบริเวณเดียวกันกับชุดข้อมูลสำหรับฝึกอีก 15 ชุด

จากรูปที่ 4.7 จะเห็นว่านิรพลเน็ตเวิร์กสามารถประมาณค่าตำแหน่งของสถานีเคลื่อนที่โดยให้ค่าความผิดพลาดมากกว่า 125 เมตร จำนวน 16 ครั้ง ซึ่งเป็นผลมาจากชุดข้อมูลสำหรับทดสอบนิรพลเน็ตเวิร์กบางส่วนไม่ได้อยู่ในบริเวณเดียวกันกับชุดข้อมูลสำหรับฝึกนิรพลเน็ตเวิร์กนั่นเอง



รูปที่ 4.7 ค่าความผิดพลาดของการประมาณค่าเมื่อชุดข้อมูลสำหรับฝึกและทดสอบ
 นีวรอลเน็ตเวิร์กบางส่วนไม่อยู่ในบริเวณเดียวกัน

- 2) สำหรับการฝึกนีวรอลเน็ตเวิร์กใช้วิธีการฝึกแบบปรับค่าตลอดเวลา และแบ่งการฝึกเป็น 3 กรณี ได้แก่ กรณีมีชุดข้อมูลภาคสนามสำหรับฝึกนีวรอลเน็ตเวิร์กจำนวน 50 ชุด 75 ชุด และ 100 ชุด และกำหนดให้ค่าเริ่มต้นของค่าถ่วงน้ำหนักทั้งหมดเท่ากับ 0 ค่าเริ่มต้นของค่าไบแอสทั้งหมดเท่ากับ 1 และค่าอัตราการเรียนรู้เท่ากับ 0.001



รูปที่ 4.8 ชุดข้อมูลสำหรับฝึกและทดสอบนีวรอลเน็ตเวิร์กทั้งหมด
 อยู่ในบริเวณเดียวกัน

- 3) สำหรับการทดสอบนิเวศเน็ตเวิร์ก ชุดข้อมูลภาคสนามสำหรับทดสอบนิเวศเน็ตเวิร์กทั้งหมดอยู่ในบริเวณเดียวกันกับชุดข้อมูลสำหรับฝึกนิเวศเน็ตเวิร์ก ดังแสดงในรูปที่ 4.8 โดยมีชุดข้อมูลสำหรับทดสอบจำนวน 40 ชุด ซึ่งเป็นชุดข้อมูลคนละชุดกับชุดข้อมูลสำหรับฝึกนิเวศเน็ตเวิร์ก
- 4) แสดงผลการทดสอบความสามารถในการประมาณค่าของนิเวศเน็ตเวิร์กที่ออกแบบผ่านโปรแกรม ModelSim XE
- 5) ในหัวข้อนี้ การตรวจสอบประสิทธิภาพการประมาณค่าของนิเวศเน็ตเวิร์กที่ออกแบบ นอกจากจะทำโดยการเปรียบเทียบกับข้อกำหนดของ FCC แล้ว ยังได้นำค่าความผิดพลาดจากนิเวศเน็ตเวิร์กที่ออกแบบโดยภาควิชาวิศวกรรมไฟฟ้า มาเปรียบเทียบกับค่าความผิดพลาดจากการประมาณค่าโดยใช้เครื่องมือนิเวศเน็ตเวิร์ก (NNTool) ที่มีอยู่ในโปรแกรม Matlab โดยเลือกประเภทของนิเวศเน็ตเวิร์กเป็นโครงข่ายชนิดบ่อน้ำและทำการฝึกแบบแพร่กระจายย้อนกลับ รวมถึงการตั้งค่าพารามิเตอร์ให้มีลักษณะเดียวกันกับนิเวศเน็ตเวิร์กที่ออกแบบ และใช้ชุดข้อมูลสำหรับฝึกและทดสอบนิเวศเน็ตเวิร์กชุดเดียวกัน

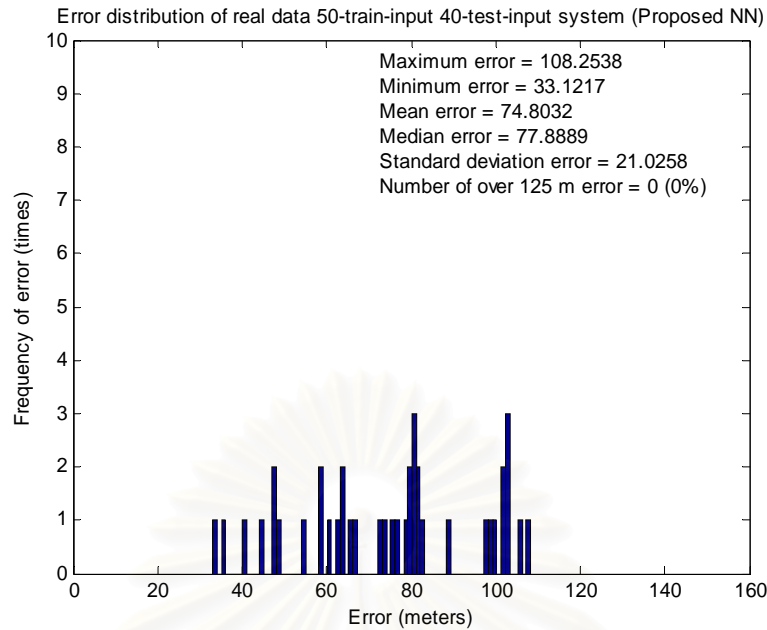
การทดสอบกรณีที่ 1 ใช้ข้อมูลภาคสนามสำหรับฝึกนิเวศเน็ตเวิร์กจำนวน 50 ชุด และชุดข้อมูลภาคสนามสำหรับทดสอบนิเวศเน็ตเวิร์กจำนวน 40 ชุด

สำหรับการทดสอบกรณีที่ 1 ตำแหน่งของสถานีเคลื่อนที่ที่นิเวศเน็ตเวิร์กที่ออกแบบประมาณค่าได้ และค่าเป้าหมายของตำแหน่ง ถูกแสดงผลผ่านโปรแกรม ModelSim XE ได้ดังรูปที่ 4.9 4.10 4.11 และ 4.12

จากรูปที่ 4.9 4.10 4.11 และ 4.12 สามารถหาค่าความผิดพลาดในการประมาณค่าตำแหน่งของสถานีเคลื่อนที่ได้ตามสมการที่ (4.1) และ (4.2) มาแสดงได้ดังรูปที่ 4.13

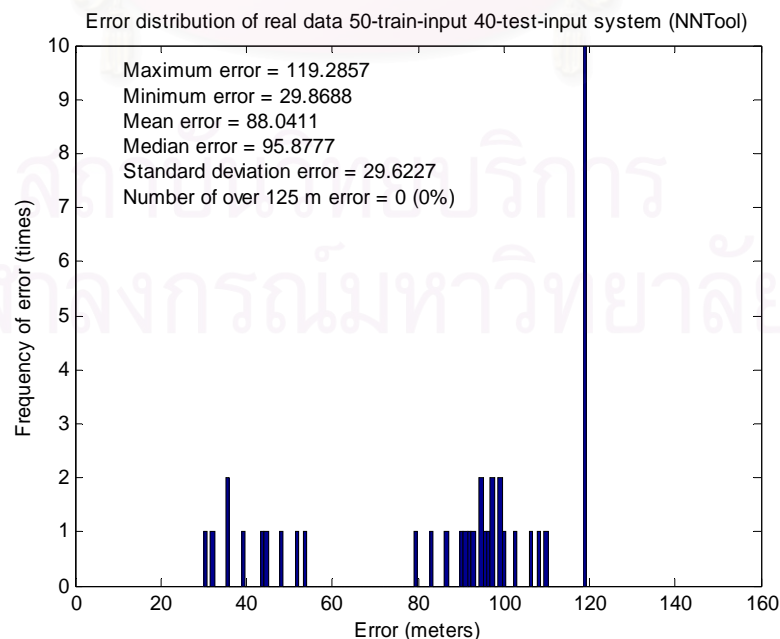
$$x = \frac{|tr_lat - m_lat| \times 10^6}{8.99322}, \quad y = \frac{|tr_long - m_long| \times 10^6}{9.013244} \quad (4.1)$$

$$error = \sqrt{x^2 + y^2} \quad \text{หน่วย: เมตร} \quad (4.2)$$



รูปที่ 4.13 ค่าความผิดพลาดของการประมาณค่าในกรณีที่ 1 โดยใช้โครงข่ายประสาทเทียมที่ออกแบบ

จากรูปที่ 4.13 จะเห็นว่าโครงข่ายประสาทเทียมที่ออกแบบสามารถประมาณค่าตำแหน่งของสถานีโดยมีค่าความผิดพลาดเฉลี่ยในการประมาณค่าเท่ากับ 74.8032 เมตร และไม่มี การประมาณค่าครั้งใดมีค่าความผิดพลาดเกิน 125 เมตร ซึ่งถือว่าถูกต้องตามที่ FCC กำหนดไว้ กล่าวคือ ระบบการประมาณค่าตำแหน่งของสถานีเคลื่อนที่ที่มีความสามารถในการรายงานค่า พิกัดตำแหน่งให้มีความถูกต้องคิดเป็น 67% ของจำนวนครั้งของการประมาณค่าในระยะความ ผิดพลาด 125 เมตร และเมื่อนำชุดข้อมูลสำหรับฝึกและทดสอบโครงข่ายประสาทเทียมชุดเดียวกันมา ทดสอบโดยใช้ NNTool ค่าความผิดพลาดในการประมาณค่าของ NNTool แสดงดังรูปที่ 4.14

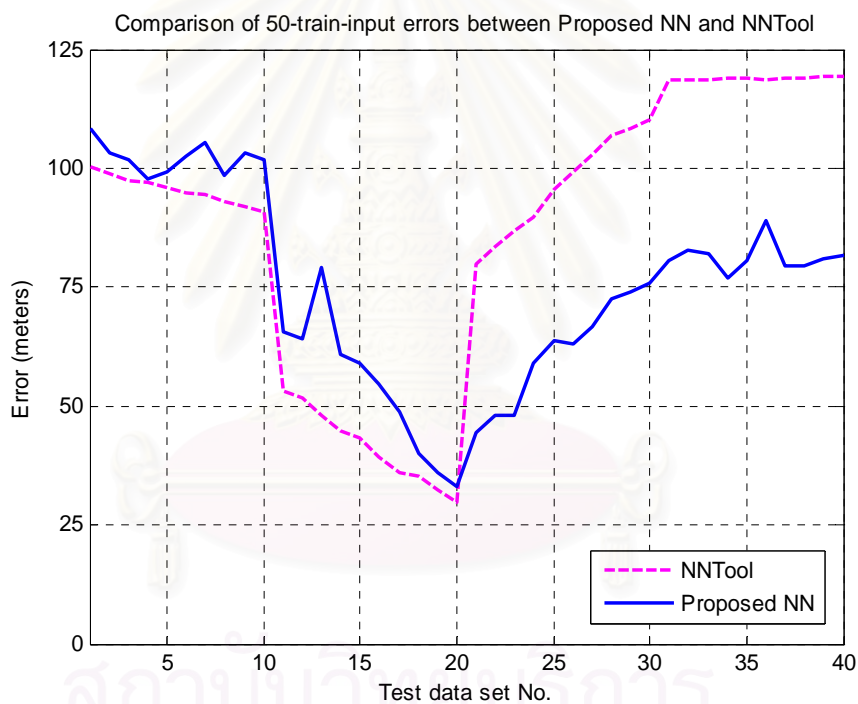


รูปที่ 4.14 ค่าความผิดพลาดของการประมาณค่าในกรณีที่ 1 โดยใช้ NNTool

จากรูปที่ 4.14 จะเห็นว่า NNTool สามารถประมาณค่าตำแหน่งของสถานีเคลื่อนที่ได้ ถูกต้องตามข้อกำหนดของ FCC เช่นกัน โดยให้ค่าความผิดพลาดเฉลี่ยในการประมาณค่าเท่ากับ 88.0411 เมตร และไม่มี การประมาณค่าครั้งใดให้ค่าความผิดพลาดสูงเกิน 125 เมตร

จากรูปที่ 4.13 และ 4.14 เมื่อเปรียบเทียบค่าความผิดพลาดในการประมาณค่าของทั้ง 2 วิธี จะเห็นว่าการประมาณค่าโดยใช้นิวรอลเน็ตเวิร์กที่ออกแบบและ NNTool มีประสิทธิภาพใกล้เคียงกัน (แตกต่างกันประมาณ 13 เมตร ซึ่งถือว่าไม่ส่งผลอย่างมีนัยสำคัญในทางปฏิบัติ) และ เพื่อให้การพิจารณา มีความชัดเจนมากยิ่งขึ้น จึงได้แสดงค่าความผิดพลาดที่เกิดขึ้นของชุดข้อมูล ทดสอบแต่ละชุดจากการประมาณค่าทั้ง 2 วิธี ดังรูปที่ 4.15

จากรูปที่ 4.15 จะเห็นว่ารูปแบบของค่าความผิดพลาดที่เกิดขึ้นจากการประมาณค่าทั้ง 2 วิธี มีแนวโน้มการเกิดค่าความผิดพลาดในลักษณะเดียวกัน

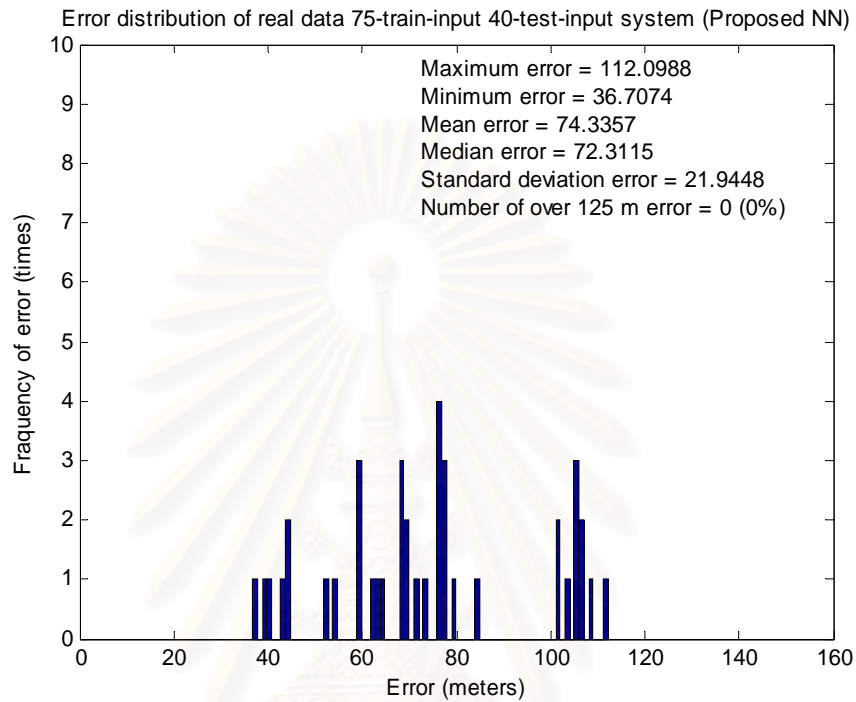


รูปที่ 4.15 ค่าความผิดพลาดที่เกิดขึ้นของชุดข้อมูลทดสอบแต่ละชุดจากการประมาณค่ากรณีที่ 1 ของนิวรอลเน็ตเวิร์กที่ออกแบบเปรียบเทียบกับ การประมาณค่าโดยใช้ NNTool

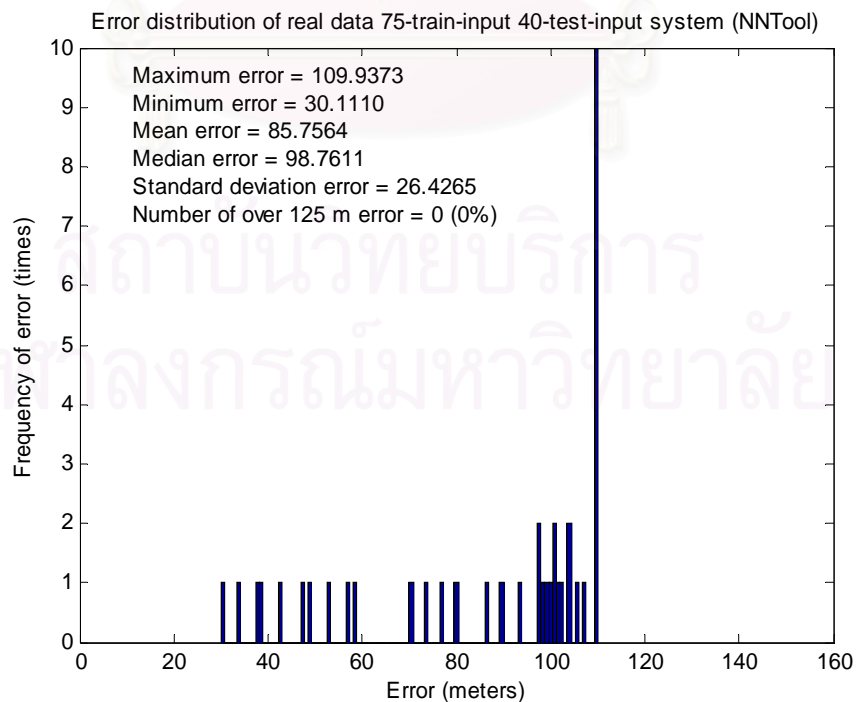
การทดสอบกรณีที่ 2 ใช้ข้อมูลภาคสนามสำหรับฝึกนิวรอลเน็ตเวิร์กจำนวน 75 ชุด และชุดข้อมูลภาคสนามสำหรับทดสอบนิวรอลเน็ตเวิร์กจำนวน 40 ชุด

สำหรับการทดสอบกรณีที่ 2 ตำแหน่งของสถานีเคลื่อนที่ที่นิวรอลเน็ตเวิร์กที่ออกแบบ ประมาณค่าได้ และค่าเป้าหมายของตำแหน่ง ถูกแสดงผลผ่านโปรแกรม ModelSim XE ได้ดังรูปที่ 4.16 4.17 4.18 และ 4.19

จากรูปที่ 4.16 4.17 4.18 และ 4.19 สามารถนำค่าความผิดพลาดในการประมาณค่าตำแหน่งของสถานีเคลื่อนที่มาแสดงได้ดังรูปที่ 4.20 โดยนิเวรอลเน็ตเวิร์กที่ออกแบบให้ค่าความผิดพลาดเฉลี่ยเท่ากับ 74.3357 เมตร ส่วนค่าความผิดพลาดในการประมาณค่าโดยใช้ NNTool แสดงได้ดังรูปที่ 4.21 โดย NNTool ให้ค่าความผิดพลาดเฉลี่ยเท่ากับ 85.7564 เมตร

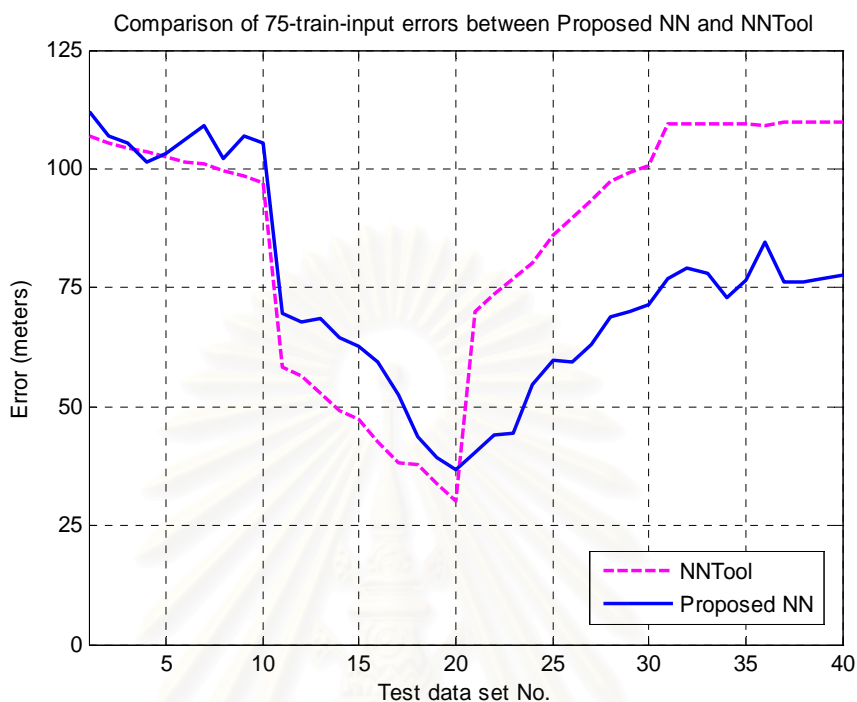


รูปที่ 4.20 ค่าความผิดพลาดของการประมาณค่าในกรณีที่ 2 โดยใช้นิเวรอลเน็ตเวิร์กที่ออกแบบ



รูปที่ 4.21 ค่าความผิดพลาดของการประมาณค่าในกรณีที่ 2 โดยใช้ NNTool

ส่วนค่าความผิดพลาดที่เกิดขึ้นของชุดข้อมูลทดสอบแต่ละชุดจากการประมาณค่าทั้ง 2 วิธีในกรณีนี้ สามารถแสดงได้ดังรูปที่ 4.22



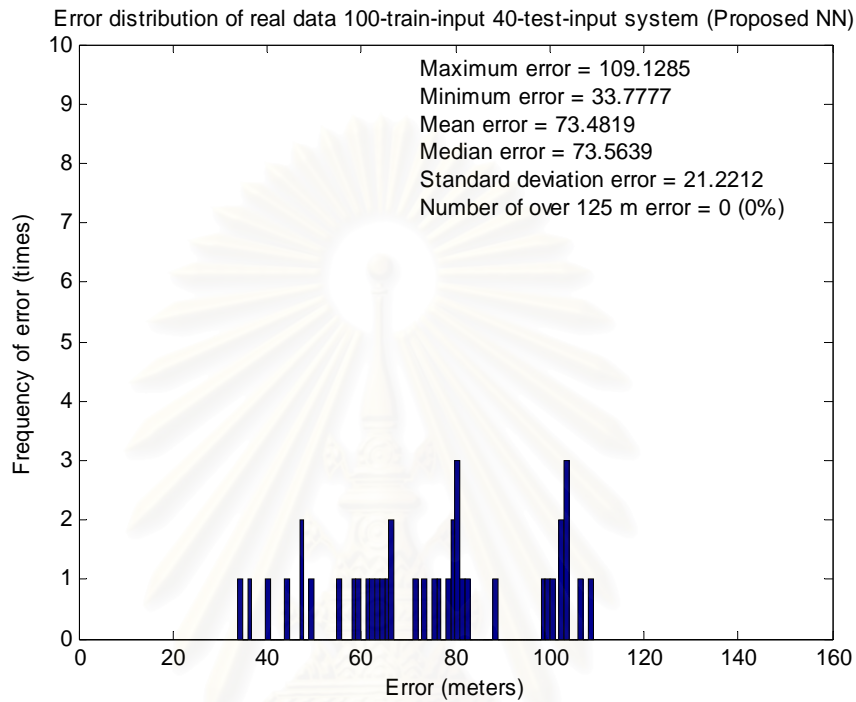
รูปที่ 4.22 ค่าความผิดพลาดที่เกิดขึ้นของชุดข้อมูลทดสอบแต่ละชุดจากการประมาณค่ากรณีที่ 2 ของนิวรอลเน็ตเวิร์กที่ออกแบบเปรียบเทียบกับค่าประมาณค่าโดยใช้ NNTool

จากรูปที่ 4.20 4.21 และ 4.22 จะเห็นได้ว่าการประมาณค่าของทั้ง 2 วิธี สามารถประมาณค่าตำแหน่งของสถานีเคลื่อนที่ได้ใกล้เคียงกันเช่นเดียวกับการทดสอบในกรณีที่ 1 แต่ค่าความผิดพลาดที่เกิดขึ้นลดลง เนื่องมาจากการเพิ่มจำนวนชุดข้อมูลสำหรับฝึกนิวรอลเน็ตเวิร์ก

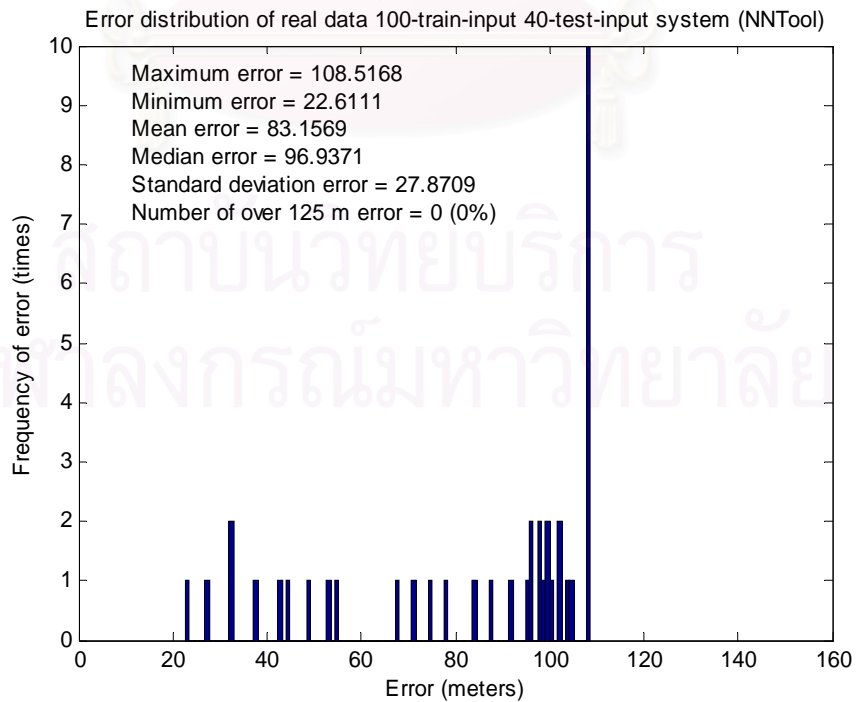
การทดสอบกรณีที่ 3 ใช้ข้อมูลภาคสนามสำหรับฝึกนิวรอลเน็ตเวิร์กจำนวน 100 ชุด และชุดข้อมูลภาคสนามสำหรับทดสอบนิวรอลเน็ตเวิร์กจำนวน 40 ชุด

สำหรับการทดสอบกรณีที่ 3 ตำแหน่งของสถานีเคลื่อนที่ที่นิวรอลเน็ตเวิร์กที่ออกแบบประมาณค่าได้ และค่าเป้าหมายของตำแหน่ง ถูกแสดงผลผ่านโปรแกรม ModelSim XE ได้ดังรูปที่ 4.23 4.24 4.25 และ 4.26

จากรูปที่ 4.23 4.24 4.25 และ 4.26 สามารถนำค่าความผิดพลาดในการประมาณค่าตำแหน่งของสถานีเคลื่อนที่มาแสดงได้ดังรูปที่ 4.27 โดยนิเวรอลเน็ตเวิร์กที่ออกแบบให้ค่าความผิดพลาดเฉลี่ยเท่ากับ 73.4819 เมตร ส่วนค่าความผิดพลาดในการประมาณค่าโดยใช้ NNTool แสดงได้ดังรูปที่ 4.28 โดย NNTool ให้ค่าความผิดพลาดเฉลี่ยเท่ากับ 83.1569 เมตร

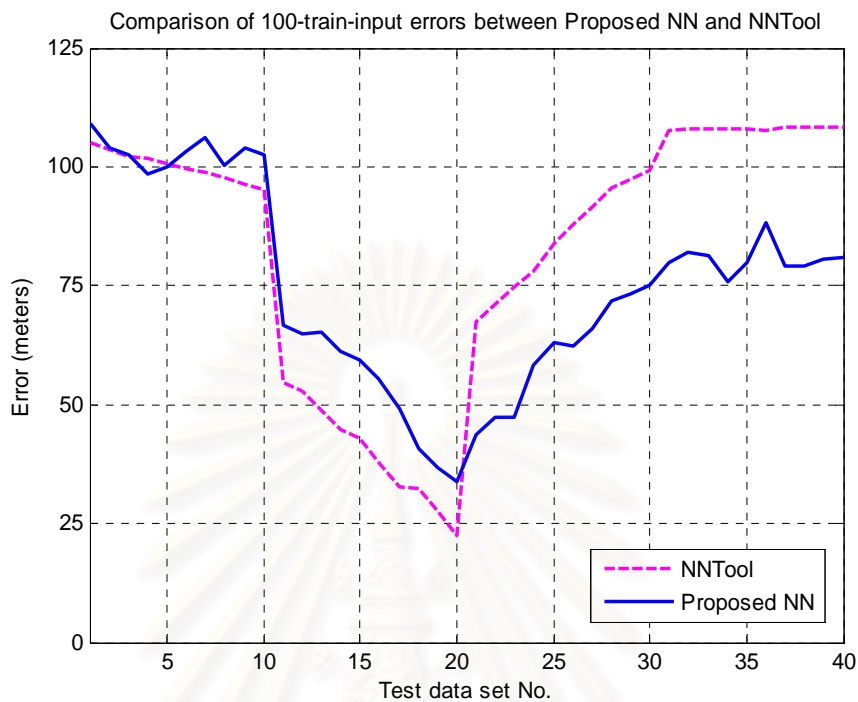


รูปที่ 4.27 ค่าความผิดพลาดของการประมาณค่าในกรณีที่ 3 โดยใช้นิเวรอลเน็ตเวิร์กที่ออกแบบ



รูปที่ 4.28 ค่าความผิดพลาดของการประมาณค่าในกรณีที่ 3 โดยใช้ NNTool

ส่วนค่าความผิดพลาดที่เกิดขึ้นของชุดข้อมูลทดสอบแต่ละชุดจากการประมาณค่าทั้ง 2 วิธีในกรณีนี้ สามารถแสดงได้ดังรูปที่ 4.29

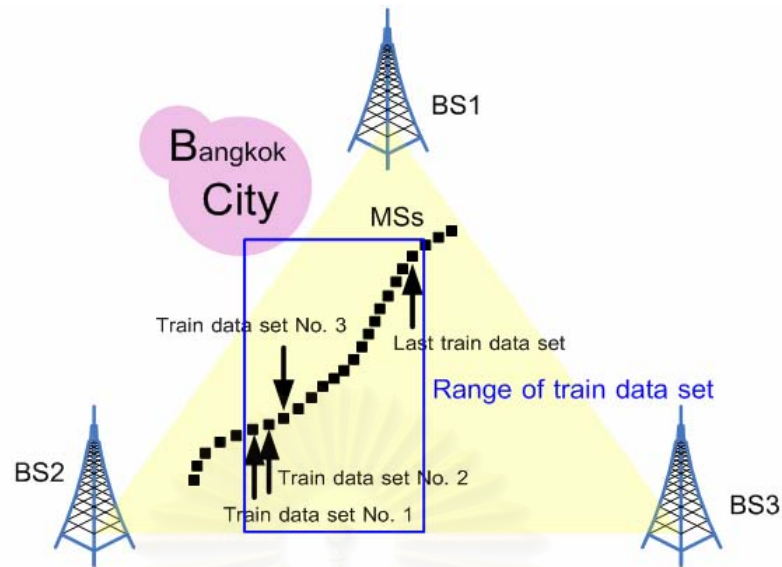


รูปที่ 4.29 ค่าความผิดพลาดที่เกิดขึ้นของชุดข้อมูลทดสอบแต่ละชุดจากการประมาณค่ากรณีที่ 3 ของนิวรอลเน็ตเวิร์กที่ออกแบบเปรียบเทียบกับค่าประมาณค่าโดยใช้ NNTool

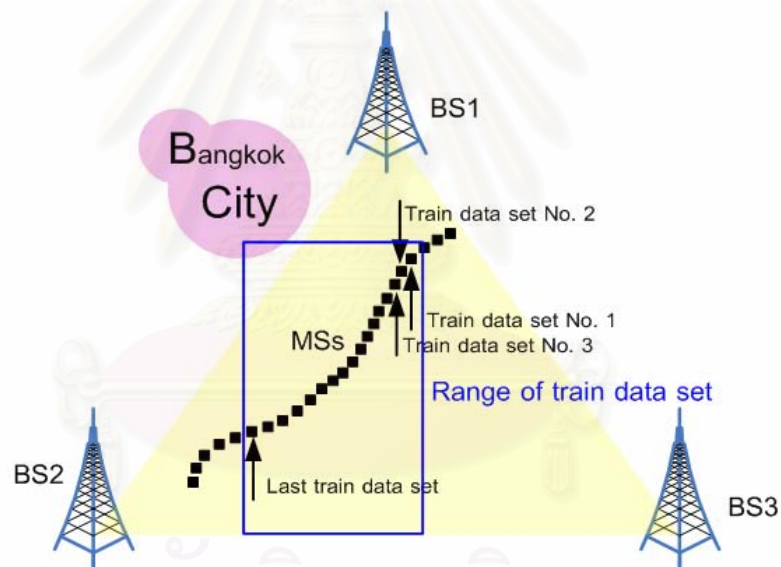
จากรูปที่ 4.27 4.28 และ 4.29 จะเห็นได้ว่าการประมาณค่าทั้ง 2 วิธี สามารถประมาณค่าตำแหน่งของสถานีเคลื่อนที่ได้ใกล้เคียงกันเช่นกัน และมีค่าความผิดพลาดในการประมาณค่าน้อยกว่า 2 กรณีก่อนหน้านี้ เนื่องจากจำนวนชุดข้อมูลสำหรับฝึกนิวรอลเน็ตเวิร์กที่มีมากกว่านั่นเอง

จากผลการทดสอบทั้ง 3 กรณี จะเห็นได้ว่านิวรอลเน็ตเวิร์กที่ออกแบบสามารถนำมาใช้ประมาณค่าตำแหน่งของสถานีเคลื่อนที่ได้ โดยมีความถูกต้องตรงตามข้อกำหนดของ FCC และเมื่อเปรียบเทียบกับค่าประมาณค่าโดยใช้ NNTool ซึ่งเป็นเครื่องมือที่เป็นที่ยอมรับและมีความน่าเชื่อถือ นิวรอลเน็ตเวิร์กที่ออกแบบสามารถประมาณค่าได้อย่างใกล้เคียง

อย่างไรก็ตาม การทดสอบทั้ง 3 กรณีที่ผ่านมา ชุดข้อมูลภาคสนามสำหรับฝึกนิวรอลเน็ตเวิร์กถูกเรียงกันอย่างเป็นลำดับ (Sequence) จากจุดหนึ่งไปยังอีกจุดหนึ่ง ดังแสดงในรูปที่ 4.30 ซึ่งการฝึกนิวรอลเน็ตเวิร์กแบบปรับค่าตลอดเวลานั้น ลำดับของชุดข้อมูลสำหรับฝึกอาจส่งผลกระทบต่อความสามารถในการประมาณค่าของนิวรอลเน็ตเวิร์ก จึงได้ลองสลับตำแหน่งของชุดข้อมูลภาคสนามสำหรับฝึกนิวรอลเน็ตเวิร์กจากหน้าไปหลัง ดังแสดงในรูปที่ 4.31 และทดสอบความสามารถในการประมาณค่าของนิวรอลเน็ตเวิร์กใหม่อีกครั้ง



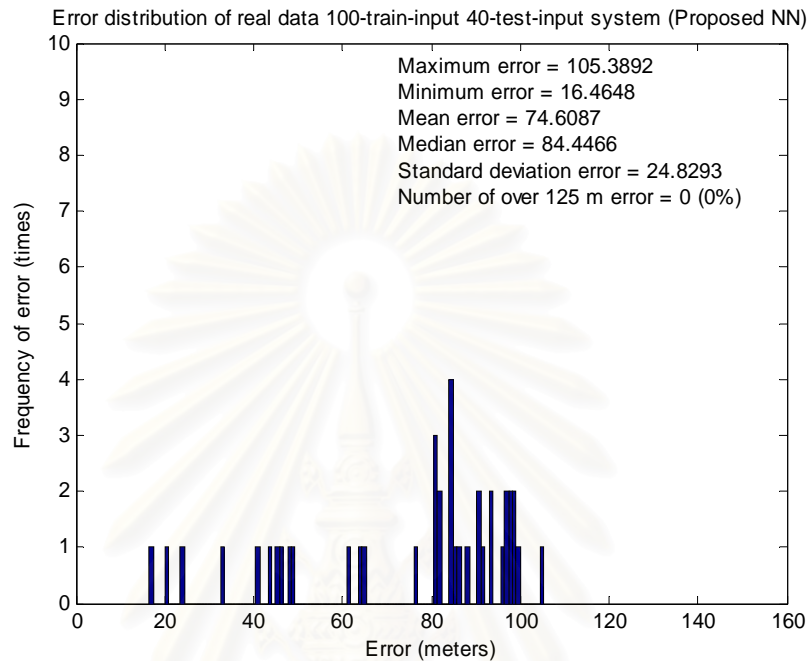
รูปที่ 4.30 ชุดข้อมูลภาคสนามสำหรับฝึกนินรอลเน็ตเวิร์ก ถูกเรียงกันอย่างเป็นลำดับจากจุดหนึ่งไปยังอีกจุดหนึ่ง



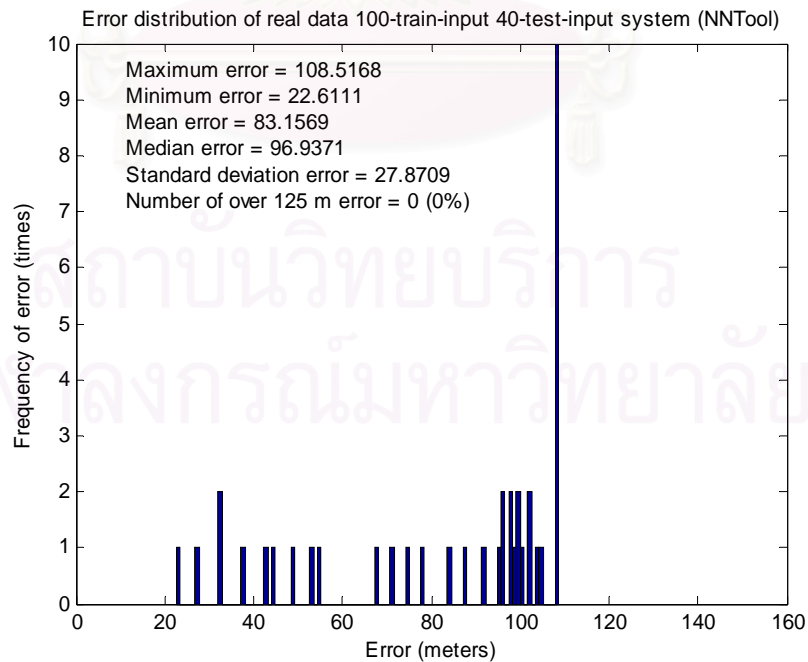
รูปที่ 4.31 ชุดข้อมูลภาคสนามสำหรับฝึกนินรอลเน็ตเวิร์ก ที่ถูกสลับตำแหน่งของชุดข้อมูลจากหน้าไปหลัง

เมื่อสลับตำแหน่งของชุดข้อมูลสำหรับฝึกนินรอลเน็ตเวิร์กแล้ว ได้ลองทดสอบนินรอลเน็ตเวิร์กในลักษณะเดียวกันกับการทดสอบกรณีที่ 3 ก่อนหน้านี้ นั่นคือ ใช้ชุดข้อมูลภาคสนามสำหรับฝึกนินรอลเน็ตเวิร์กจำนวน 100 ชุด และสำหรับทดสอบนินรอลเน็ตเวิร์กจำนวน 40 ชุด ซึ่งนินรอลเน็ตเวิร์กที่ออกแบบสามารถประมาณค่าตำแหน่งของสถานีเคลื่อนที่ได้ดังรูปที่ 4.32 4.33 4.34 และ 4.35

จากรูปที่ 4.32 4.33 4.34 และ 4.35 สามารถนำค่าความผิดพลาดในการประมาณค่าตำแหน่งของสถานีเคลื่อนที่มาแสดงได้ดังรูปที่ 4.36 โดยนิเวรอลเน็ตเวิร์กที่ออกแบบให้ค่าความผิดพลาดเฉลี่ยเท่ากับ 74.6087 เมตร ส่วนค่าความผิดพลาดในการประมาณค่าโดยใช้ NNTool แสดงได้ดังรูปที่ 4.37 โดย NNTool ให้ค่าความผิดพลาดเฉลี่ยเท่ากับ 83.1569 เมตร

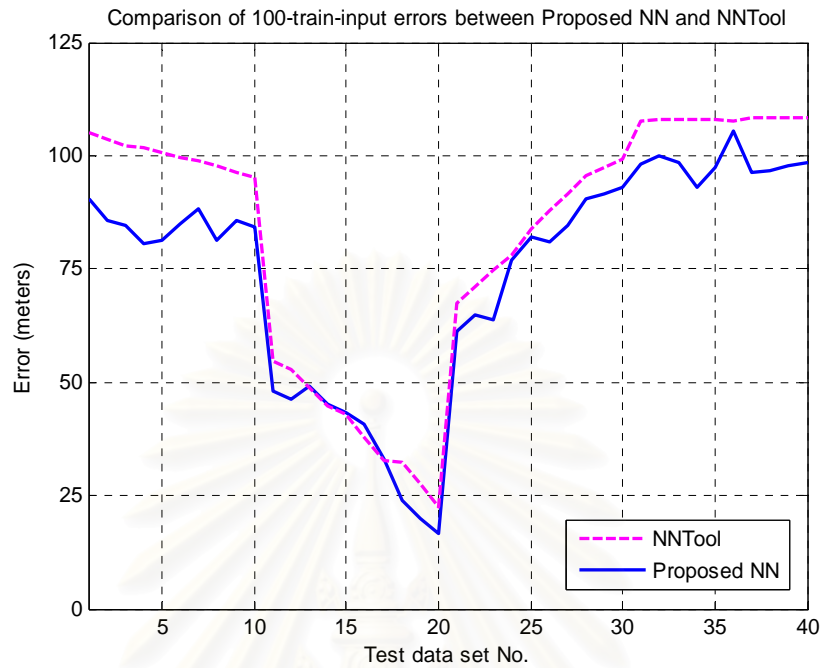


รูปที่ 4.36 ค่าความผิดพลาดของการประมาณค่าในกรณีที่ 3 โดยใช้นิเวรอลเน็ตเวิร์กที่ออกแบบ (สลับตำแหน่งของชุดข้อมูลภาคสนามสำหรับฝึกนิเวรอลเน็ตเวิร์กจากหน้าไปหลัง)



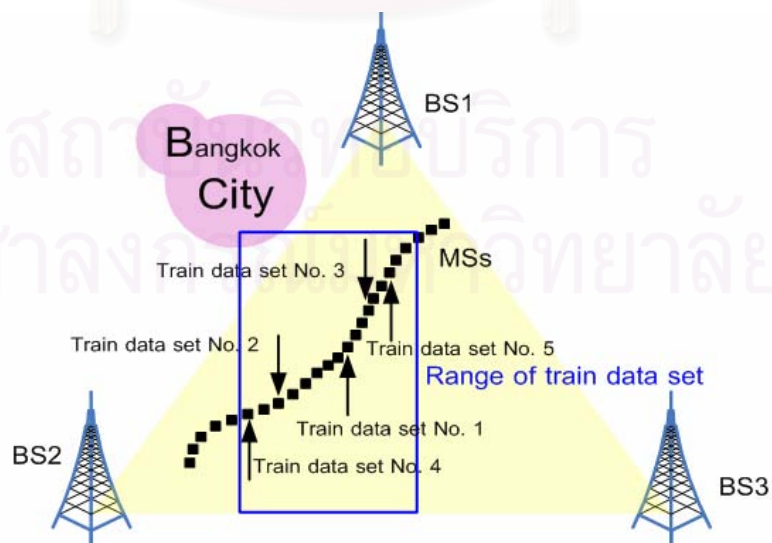
รูปที่ 4.37 ค่าความผิดพลาดของการประมาณค่าในกรณีที่ 3 โดยใช้ NNTool (สลับตำแหน่งของชุดข้อมูลภาคสนามสำหรับฝึกนิเวรอลเน็ตเวิร์กจากหน้าไปหลัง)

ส่วนค่าความผิดพลาดที่เกิดขึ้นของชุดข้อมูลทดสอบแต่ละชุดจากการประมาณค่าทั้ง 2 วิธีในกรณีนี้ สามารถแสดงได้ดังรูปที่ 4.38



รูปที่ 4.38 ค่าความผิดพลาดที่เกิดขึ้นของชุดข้อมูลทดสอบแต่ละชุดจากการประมาณค่ากรณีที่ 3 ของนิรลเน็ตเวิร์กที่ออกแบบเปรียบเทียบกับค่าประมาณค่าโดยใช้ NNTool (สลับตำแหน่งของชุดข้อมูลภาคสนามสำหรับฝึกนิรลเน็ตเวิร์กจากหน้าไปหลัง)

จากรูปที่ 4.36 4.37 และ 4.38 จะเห็นว่าค่าประมาณค่าตำแหน่งของสถานีเคลื่อนที่โดยใช้นิรลเน็ตเวิร์กที่ออกแบบ มีค่าความผิดพลาดเฉลี่ยใกล้เคียงกับการประมาณค่าโดยใช้ NNTool และมีรูปแบบของค่าความผิดพลาดของชุดข้อมูลทดสอบแต่ละชุดที่คล้ายคลึงกันมาก



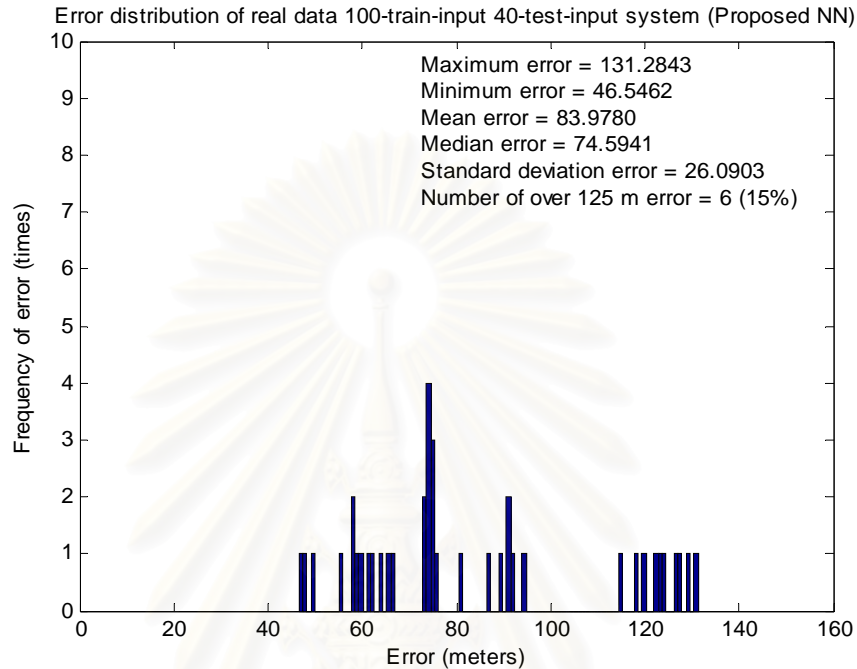
รูปที่ 4.39 ชุดข้อมูลภาคสนามสำหรับฝึกนิรลเน็ตเวิร์กที่ถูกสลับตำแหน่งอย่างไม่มีการเปรียบเทียบ

ถึงแม้จะมีการสลับตำแหน่งของชุดข้อมูลสำหรับฝึกนิรอลเน็ตเวิร์ก อย่างไรก็ตาม ชุดข้อมูลที่ถูกลบยังคงมีลักษณะเป็นลำดับเช่นเดิม จึงได้ลองทดสอบโดยการสลับตำแหน่งของชุดข้อมูลสำหรับฝึกนิรอลเน็ตเวิร์กอย่างไม่ระเบียบ เพื่อให้ชุดข้อมูลสำหรับฝึกไม่มีลักษณะเป็นลำดับ ดังแสดงในรูปที่ 4.39 และลองทดสอบนิรอลเน็ตเวิร์กในลักษณะเดียวกันกับการทดสอบกรณีที่ 3 ก่อนหน้านี้เช่นกัน นั่นคือ ใช้ชุดข้อมูลภาคสนามสำหรับฝึกนิรอลเน็ตเวิร์กจำนวน 100 ชุด และสำหรับทดสอบนิรอลเน็ตเวิร์กจำนวน 40 ชุด ซึ่งนิรอลเน็ตเวิร์กที่ออกแบบสามารถประมาณค่าตำแหน่งของสถานีเคลื่อนที่ได้ดังรูปที่ 4.40 4.41 4.42 และ 4.43

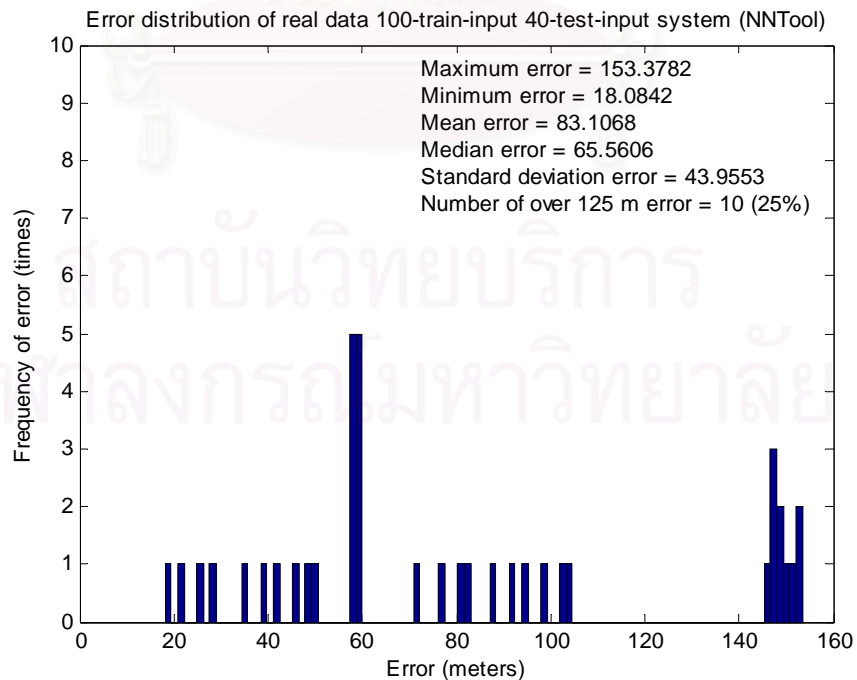


สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

จากรูปที่ 4.40 4.41 4.42 และ 4.43 สามารถนำค่าความผิดพลาดในการประมาณค่าตำแหน่งของสถานีเคลื่อนที่มาแสดงได้ดังรูปที่ 4.44 โดยนิเวรอลเน็ตเวิร์กที่ออกแบบให้ค่าความผิดพลาดเฉลี่ยเท่ากับ 83.9780 เมตร ส่วนค่าความผิดพลาดในการประมาณค่าโดยใช้ NNTool แสดงได้ดังรูปที่ 4.45 โดย NNTool ให้ค่าความผิดพลาดเฉลี่ยเท่ากับ 83.1068 เมตร

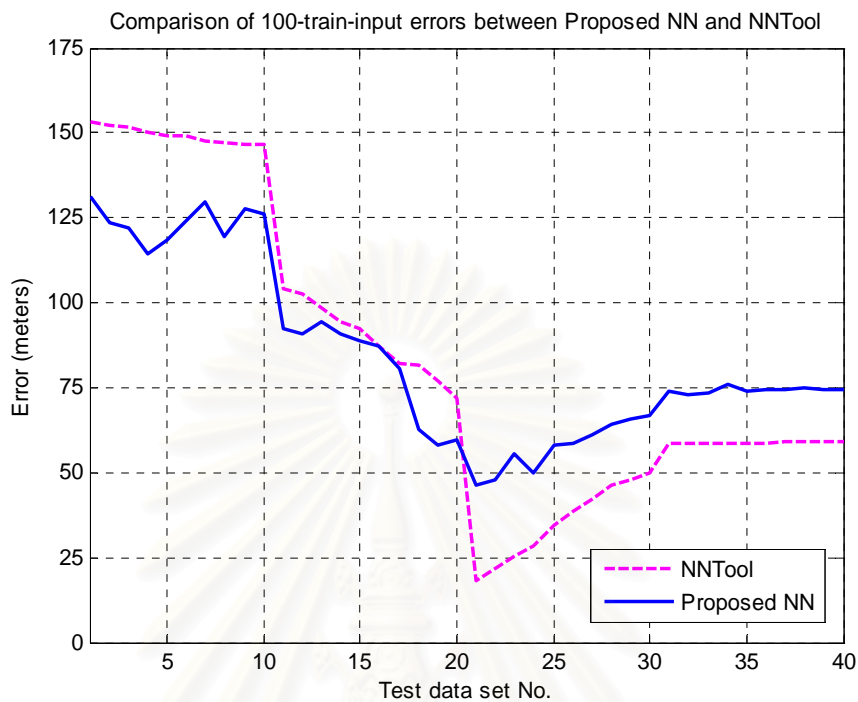


รูปที่ 4.44 ค่าความผิดพลาดของการประมาณค่าในกรณีที่ 3 โดยใช้นิเวรอลเน็ตเวิร์กที่ออกแบบ (สลับตำแหน่งของชุดข้อมูลภาคสนามสำหรับฝึกนิเวรอลเน็ตเวิร์กอย่างไม่มีระเบียบ)



รูปที่ 4.45 ค่าความผิดพลาดของการประมาณค่าในกรณีที่ 3 โดยใช้ NNTool (สลับตำแหน่งของชุดข้อมูลภาคสนามสำหรับฝึกนิเวรอลเน็ตเวิร์กอย่างไม่มีระเบียบ)

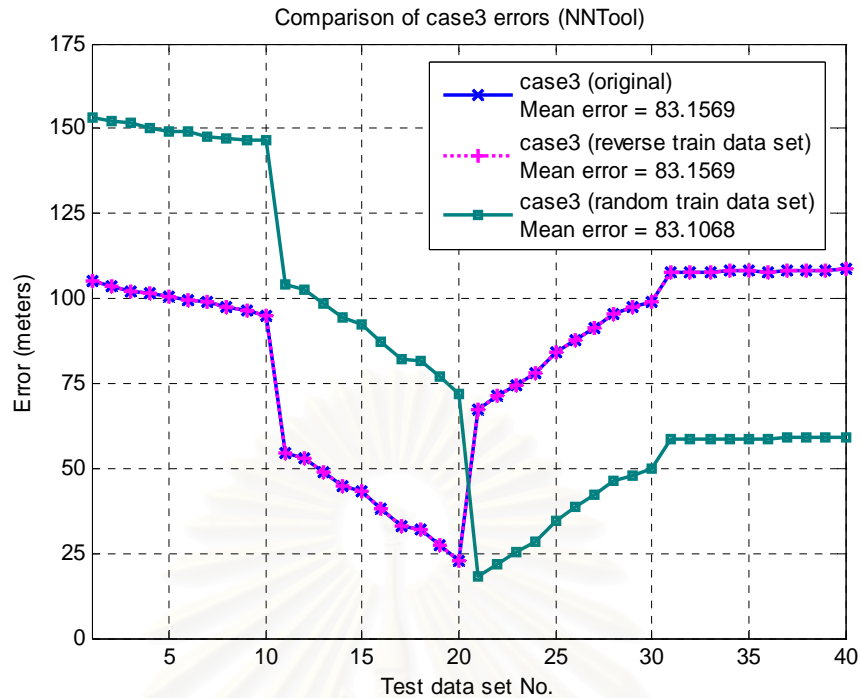
ส่วนค่าความผิดพลาดที่เกิดขึ้นของชุดข้อมูลทดสอบแต่ละชุดจากการประมาณค่าทั้ง 2 วิธีในกรณีนี้ สามารถแสดงได้ดังรูปที่ 4.46



รูปที่ 4.46 ค่าความผิดพลาดที่เกิดขึ้นของชุดข้อมูลทดสอบแต่ละชุดจากการประมาณค่ากรณีที่ใช้ของนิวรอลเน็ตเวิร์กที่ออกแบบเปรียบเทียบกับค่าการประมาณค่าโดยใช้ NNTool (สลับตำแหน่งของชุดข้อมูลภาคสนามสำหรับฝึกนิวรอลเน็ตเวิร์กอย่างไม่มีการเปรียบเทียบ)

จากรูปที่ 4.36 4.37 และ 4.38 จะเห็นว่าการประมาณค่าตำแหน่งของสถานีเคลื่อนที่โดยใช้นิวรอลเน็ตเวิร์กที่ออกแบบ มีค่าความผิดพลาดเฉลี่ยใกล้เคียงกับการประมาณค่าโดยใช้ NNTool และมีรูปแบบของค่าความผิดพลาดของชุดข้อมูลทดสอบแต่ละชุดที่คล้ายคลึงกันมาก อย่างไรก็ตาม การประมาณค่าเมื่อมีการสลับตำแหน่งของชุดข้อมูลภาคสนามสำหรับฝึกนิวรอลเน็ตเวิร์กอย่างไม่มีการเปรียบเทียบของทั้ง 2 วิธี ถึงแม้จะให้ค่าความผิดพลาดในการประมาณค่าถูกต้องอยู่ในขอบเขตที่ FCC กำหนดไว้ แต่จะสังเกตเห็นได้ว่าการประมาณค่าโดยใช้นิวรอลเน็ตเวิร์กที่ออกแบบมีจำนวนครั้งที่เกิดค่าความผิดพลาดในการประมาณค่ามากกว่า 125 เมตร จำนวน 6 ครั้ง และสำหรับ NNTool จำนวน 10 ครั้ง

จากผลการทดสอบที่ผ่านมาแสดงให้เห็นว่าการสลับตำแหน่งของชุดข้อมูลภาคสนามสำหรับฝึกนิวรอลเน็ตเวิร์กสามารถส่งผลกระทบต่อความสามารถในการประมาณค่าของนิวรอลเน็ตเวิร์ก โดยผลการเปรียบเทียบค่าความผิดพลาดที่เกิดขึ้นของชุดข้อมูลทดสอบแต่ละชุดเมื่อมีการสลับตำแหน่งของข้อมูลภาคสนามในลักษณะต่าง ๆ แสดงได้ดังรูปที่ 4.47 และ 4.48



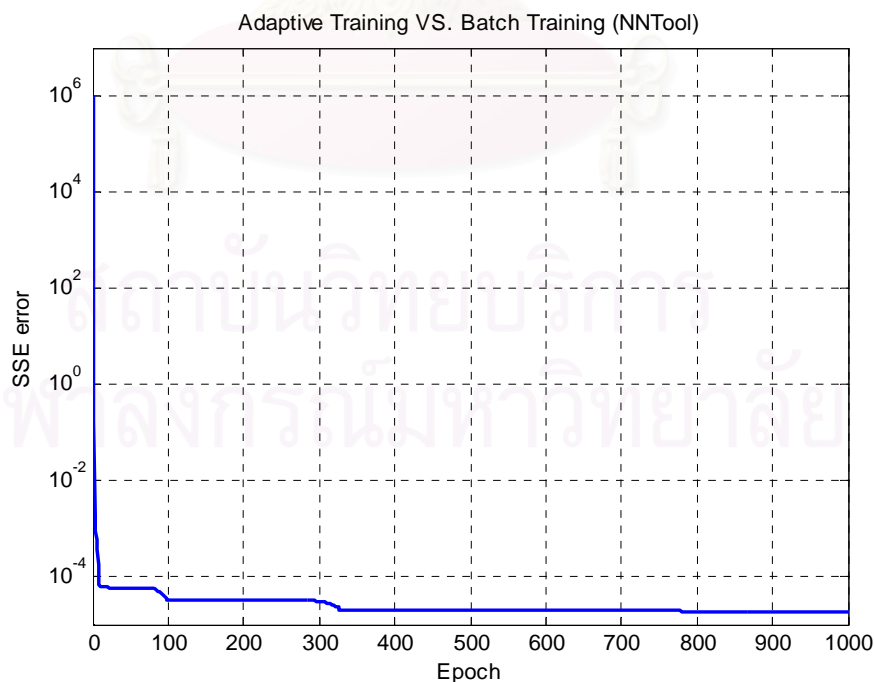
รูปที่ 4.48 ค่าความผิดพลาดที่เกิดขึ้นจากการประมาณค่าของ NNTool เมื่อมีการสลับตำแหน่งของชุดข้อมูลภาคสนามสำหรับฝึกนินวอลเน็ตเวิร์ก

จากรูปที่ 4.48 ซึ่งเป็นการประมาณค่าตำแหน่งของสถานีเคลื่อนที่โดยใช้ NNTool จะเห็นว่าค่าความผิดพลาดเฉลี่ยในการประมาณค่าของ NNTool ทั้งในกรณีไม่มีการสลับตำแหน่งชุดข้อมูลภาคสนามสำหรับฝึกนินวอลเน็ตเวิร์ก (กรณีที่ 3 เดิม) กรณีสลับตำแหน่งชุดข้อมูลสำหรับฝึกจากหน้าไปหลัง และกรณีสลับตำแหน่งชุดข้อมูลสำหรับฝึกอย่างไม่มีการสลับตำแหน่งชุดข้อมูลสำหรับฝึกและกรณีสลับตำแหน่งชุดข้อมูลสำหรับฝึกจากหน้าไปหลังมีลักษณะค่อนข้างสมมาตรและเหมือนกันทุกประการ แตกต่างกับกรณีสลับตำแหน่งชุดข้อมูลสำหรับฝึกอย่างไม่มีการสลับ รูปแบบของค่าความผิดพลาดที่เกิดขึ้นมีลักษณะไม่สมมาตรอย่างเห็นได้ชัด ทั้งนี้ NNTool ก็สามารถประมาณค่าตำแหน่งของสถานีเคลื่อนที่ได้ถูกต้องตามข้อกำหนดของ FCC ไม่ว่าจะสลับชุดข้อมูลสำหรับฝึกในลักษณะใดก็ตามเช่นเดียวกัน แต่ในการประมาณค่าเมื่อมีการสลับตำแหน่งชุดข้อมูลสำหรับฝึกอย่างไม่มีการสลับ ผลปรากฏว่าเกิดค่าความผิดพลาดในการประมาณค่าที่มากกว่า 125 เมตร จำนวน 10 ครั้งด้วยกัน คิดเป็น 25% ของจำนวนครั้งการประมาณค่า ซึ่งในทางปฏิบัติ ถือเป็นความเสี่ยงที่ค่อนข้างสูง

จากผลการทดสอบในรูปที่ 4.47 และ 4.48 อาจกล่าวได้ว่า หากไม่ต้องการให้เกิดความเสี่ยงในทางปฏิบัติ การฝึกนินวอลเน็ตเวิร์กควรใช้ชุดข้อมูลภาคสนามที่มีลักษณะเป็นลำดับมากกว่าชุดข้อมูลที่ไม่เป็นระเบียบ

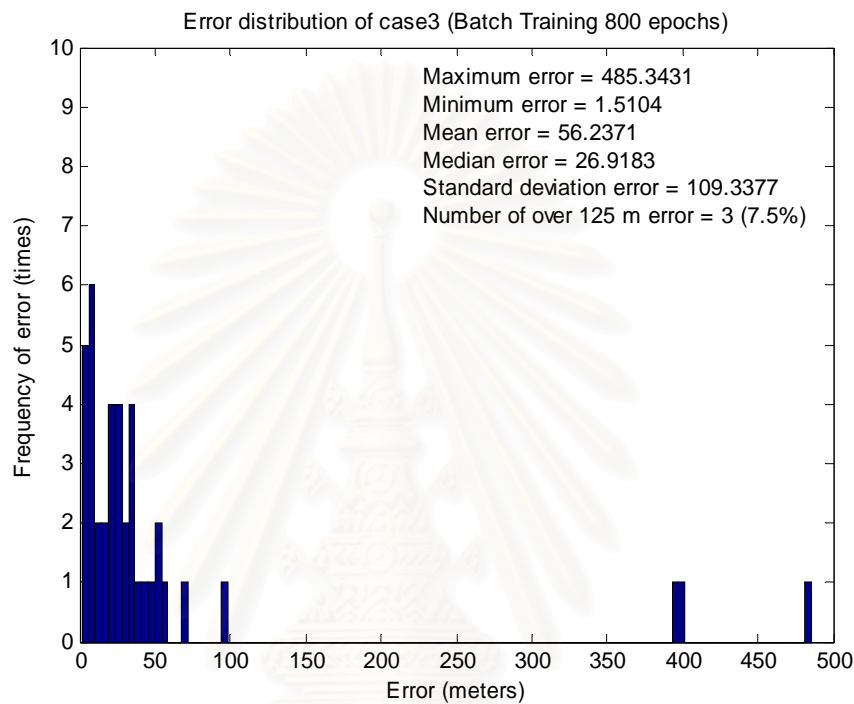
จากที่ได้กล่าวมาแล้วในบทที่ 3 ว่าการฝึกนิรอลเน็ตเวิร์กสามารถแบ่งได้เป็น 2 ประเภทตามลักษณะของการปรับค่าถ่วงน้ำหนักและค่าไบแอส ได้แก่ การฝึกนิรอลเน็ตเวิร์กแบบปรับค่าตลอดเวลา (Adaptive Training) และการฝึกนิรอลเน็ตเวิร์กแบบปรับค่าครั้งเดียว (Batch Training) ซึ่งในวิทยานิพนธ์นี้ การออกแบบนิรอลเน็ตเวิร์กเพื่อประมาณค่าตำแหน่งของสถานีเคลื่อนที่ใช้วิธีการฝึกแบบปรับค่าตลอดเวลา ซึ่งประสิทธิภาพของนิรอลเน็ตเวิร์กที่ถูกฝึกโดยวิธีการนี้ได้แสดงให้เห็นดังการทดสอบหลาย ๆ การทดสอบก่อนหน้าแล้ว

สาเหตุของการเลือกวิธีฝึกแบบนี้ในการฝึกนิรอลเน็ตเวิร์กที่ออกแบบเนื่องจาก ก่อนออกแบบนิรอลเน็ตเวิร์กได้มีการทดสอบประสิทธิภาพของนิรอลเน็ตเวิร์กที่ใช้วิธีการฝึกทั้ง 2 แบบ ซึ่งจากการทดสอบผลปรากฏว่าวิธีการฝึกแบบปรับค่าตลอดเวลาทำให้นิรอลเน็ตเวิร์กเกิดการลู่เข้าเร็วกว่าแต่ให้ค่าความผิดพลาดเฉลี่ยสูงกว่าวิธีการฝึกแบบปรับค่าครั้งเดียว ในที่นี้จะแสดงผลการทดสอบโดยยกการทดสอบกรณีที่ 3 มาใช้อีกครั้งหนึ่ง นั่นคือใช้ข้อมูลภาคสนามสำหรับฝึกนิรอลเน็ตเวิร์กจำนวน 100 ชุด และชุดข้อมูลภาคสนามสำหรับทดสอบนิรอลเน็ตเวิร์กจำนวน 40 ชุด โดยเปลี่ยนวิธีการฝึกนิรอลเน็ตเวิร์กจากวิธีการฝึกแบบปรับค่าตลอดเวลาเป็นวิธีการฝึกแบบปรับค่าครั้งเดียว และใช้ NNTool เป็นเครื่องมือในการประมาณค่าเพื่อแสดงผลการทดสอบ (สำหรับนิรอลเน็ตเวิร์กที่ออกแบบด้วยภาษาเวซดีแอส การเปลี่ยนวิธีการฝึกทำให้ต้องออกแบบนิรอลเน็ตเวิร์กใหม่ในหลาย ๆ ส่วน ซึ่งในขณะนี้กำลังอยู่ในระหว่างการศึกษาเพื่อออกแบบเพิ่มเติม) ซึ่งการลู่เข้าของนิรอลเน็ตเวิร์กแสดงได้ดังรูปที่ 4.49



รูปที่ 4.49 ผลการทดสอบการลู่เข้าของนิรอลเน็ตเวิร์กในกรณีที่ 3 เมื่อเปลี่ยนวิธีการฝึกนิรอลเน็ตเวิร์กเป็นแบบปรับค่าครั้งเดียว

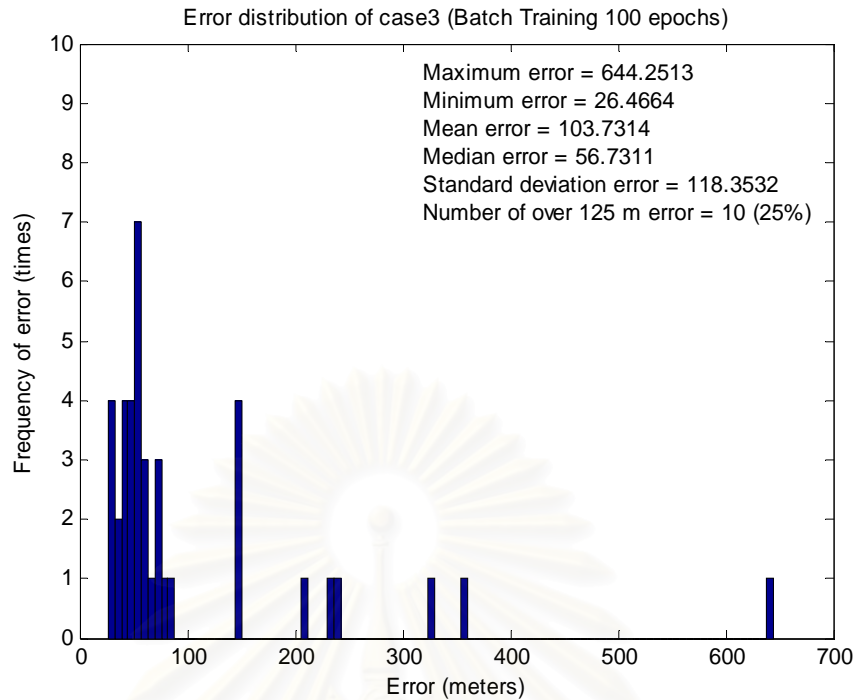
จากรูปที่ 4.49 จะเห็นว่านิรอลเน็ตเวิร์กเกิดการลู่เข้าที่ค่อนข้างช้า ต้องฝึกนิรอลเน็ตเวิร์กด้วยชุดข้อมูลสำหรับฝึกทั้งหมด (100 ชุดข้อมูล) เป็นจำนวนมากกว่า 800 รอบ (800 epochs) นิรอลเน็ตเวิร์กจึงจะลู่เข้าอย่างสมบูรณ์ ซึ่งในทางปฏิบัติต้องใช้เวลาพอสมควร ซึ่งค่าความผิดพลาดในการประมาณค่าของนิรอลเน็ตเวิร์กที่ผ่านการฝึกด้วยวิธีการฝึกแบบปรับค่าครั้งเดียวจำนวน 800 รอบ แสดงได้ดังรูปที่ 4.50



รูปที่ 4.50 ค่าความผิดพลาดของการประมาณค่าตำแหน่งของสถานีเคลื่อนที่
 หลังผ่านการฝึกด้วยวิธีการฝึกแบบปรับค่าครั้งเดียวจำนวน 800 รอบ

จากรูปที่ 4.50 เมื่อนำค่าความผิดพลาดเฉลี่ยในการประมาณค่าเมื่อใช้วิธีการฝึกแบบนี้ ซึ่งมีค่าเท่ากับ 56.2371 เมตร ไปเปรียบเทียบกับค่าความผิดพลาดเฉลี่ยเมื่อใช้วิธีการฝึกแบบปรับค่าตลอดเวลา (รูปที่ 4.28) ซึ่งมีค่าเท่ากับ 83.1569 เมตร จะเห็นว่าค่าความผิดพลาดเฉลี่ยเมื่อใช้วิธีการฝึกแบบปรับค่าครั้งเดียวจะน้อยกว่า อย่างไรก็ตาม การฝึกนิรอลเน็ตเวิร์กด้วยวิธีนี้นอกจากต้องใช้เวลาในการฝึกที่นานกว่ามาก ยังเกิดค่าความผิดพลาดที่มากกว่า 125 เมตร เป็นจำนวน 3 ครั้ง (7.5%) ด้วยกัน ซึ่งถือเป็นความเสี่ยงหากมีการนำไปใช้งานจริง นอกจากนี้วิธีการฝึกแบบนี้ยังทำให้การออกแบบนิรอลเน็ตเวิร์กเพื่อนำไปพัฒนาเป็นอุปกรณ์สำหรับประมาณค่าตำแหน่งของสถานีเคลื่อนที่ได้ยากกว่าวิธีการฝึกแบบปรับค่าตลอดเวลาอีกประการหนึ่งด้วย

เมื่อลองนำนิรอลเน็ตเวิร์กที่ผ่านการฝึกด้วยวิธีการฝึกแบบปรับค่าครั้งเดียวจำนวนเพียง 100 รอบมาประมาณค่าตำแหน่งของสถานีเคลื่อนที่ ค่าความผิดพลาดในการประมาณค่าสามารถแสดงได้ดังรูปที่ 4.51



รูปที่ 4.51 ค่าความผิดพลาดของการประมาณค่าตำแหน่งของสถานีเคลื่อนที่
หลังผ่านการฝึกด้วยวิธีการฝึกแบบปรับค่าครั้งเดียวจำนวน 100 รอบ

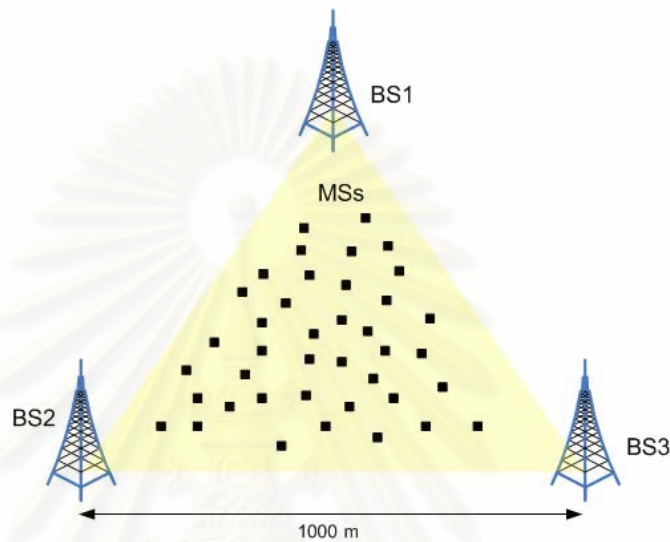
จากรูปที่ 4.50 จะเห็นว่าหากต้องการความรวดเร็วในการทำงาน และรีบร้อนนำนิรอลเน็ตเวิร์กที่ยังไม่ผ่านการฝึกให้เกิดการลู่เข้าอย่างสมบูรณ์มาประมาณค่าตำแหน่งของสถานีเคลื่อนที่ ค่าความผิดพลาดเฉลี่ยในการประมาณค่าเมื่อใช้วิธีการฝึกแบบปรับค่าครั้งเดียว ซึ่งมีค่าเท่ากับ 103.7314 เมตร จะสูงกว่าค่าความผิดพลาดเฉลี่ยเมื่อใช้วิธีการฝึกแบบปรับค่าตลอดเวลา (รูปที่ 4.28) ซึ่งมีค่าเท่ากับ 83.1569 เมตร นอกจากนี้ยังเกิดค่าความผิดพลาดที่มากกว่า 125 เมตร เป็นจำนวน 10 ครั้ง (25%) ด้วยกัน ซึ่งถือเป็นความเสี่ยงที่ค่อนข้างสูงหากมีการนำนิรอลเน็ตเวิร์กไปใช้งานจริง

เนื่องจากการฝึกและทดสอบนิรอลเน็ตเวิร์กด้วยข้อมูลภาคสนามในหัวข้อที่ผ่านมา การเคลื่อนที่ของสถานีเคลื่อนที่มีลักษณะเป็นเส้นตามแนวของถนน ทำให้ชุดข้อมูลที่นำมาฝึกและทดสอบนิรอลเน็ตเวิร์กมีลักษณะเป็นแบบรูป (Pattern) ในหัวข้อต่อไปจึงได้จำลองชุดข้อมูลสำหรับฝึกและทดสอบนิรอลเน็ตเวิร์กขึ้นมาใหม่ โดยชุดข้อมูลไม่มีลักษณะเป็นแบบรูปเหมือนข้อมูลภาคสนาม แต่มีลักษณะกระจายตัวอย่างสม่ำเสมอ และทดสอบในลักษณะเดียวกันเพื่อวิเคราะห์ผลการทดสอบต่อไป

4.3 ผลการทดสอบนิรवलเน็ตเวิร์กที่ออกแบบด้วยข้อมูลจำลอง

การฝึกและทดสอบนิรवलเน็ตเวิร์กที่ออกแบบโดยใช้ข้อมูลจำลองมีรายละเอียดดังต่อไปนี้

- 1) สถานีฐาน 3 สถานี ซึ่งตั้งเป็นรูป 3 เหลี่ยม ที่มีสถานีฐานแต่ละสถานีเป็นจุดมุม แต่ละสถานีห่างกัน 1000 เมตร มีสถานีเคลื่อนที่ที่ตั้งอยู่ภายในบริเวณ 3 เหลี่ยมดังกล่าวและมีการกระจายตัวอยู่อย่างสม่ำเสมอ ดังแสดงในรูปที่ 4.52



รูปที่ 4.52 ระบบสถานีฐานและสถานีเคลื่อนที่ที่ใช้ในการจำลองแบบ

- 2) การจำลองค่าความแรงของสัญญาณเพื่อใช้เป็นข้อมูลเข้าในการฝึกและทดสอบ มีรายละเอียดดังต่อไปนี้

การจำลองค่าความแรงของสัญญาณจากสถานีเคลื่อนที่ที่สถานีฐานรับได้นั้นจะเป็นไปตามสมการที่ (4.3)

$$P_s = P_r + L \quad (4.3)$$

เมื่อ P_s เป็น ค่ากำลังของสัญญาณที่ส่งออกจากเครื่องส่ง (dB)

P_r เป็น ค่ากำลังของสัญญาณที่เครื่องรับรับได้ (dB)

L เป็นค่าการสูญเสียตามระยะทางของคลื่น (dB)

สมการที่ (4.3) เป็นกฎทรงพลังงาน ซึ่งแสดงถึงกำลังของสัญญาณที่ส่งจากเครื่องส่ง (สถานีเคลื่อนที่) จะสูญเสียไปส่วนหนึ่งตามระยะทางการเดินทางของคลื่นก่อนที่จะถึงเครื่องรับ (สถานีฐาน) การจำลองค่าการสูญเสียตามระยะทางการเดินทางของคลื่น (Path Loss/Propagation Loss) และผลของบริเวณเงา (shadowing) เป็นดังสมการที่ (4.4) [10]

$$L = 69.55 + 26.16 \log f_c - 13.82 \log h_t - a(h_r) + (44.9 - 6.55 \log h_t) \log d \quad (4.4)$$

โดยที่ L คือ ค่าการสูญเสียตามระยะทางของคลื่น (dB)

f_c คือค่าความถี่ของคลื่นพาห้ที่ส่ง

h_t คือค่าความสูงของเสาอากาศของเครื่องส่ง (m)

h_r คือค่าความสูงของเสาอากาศของเครื่องรับ (m)

$a(h_r)$ คือค่าแฟกเตอร์ความสัมพันธ์ (Correlation factor) ระหว่างสถานีเคลื่อนที่กับสภาพภูมิประเทศ

d คือค่าระยะทางจากเครื่องส่งถึงเครื่องรับ (m)

ค่า $a(h_r)$ จะมีค่าขึ้นกับขนาดของสภาพภูมิประเทศ ดังนี้

สำหรับภูมิประเทศที่เป็นชนบทหรือชานเมือง ค่า $a(h_r)$ จะเป็นไปตามสมการที่

(4.5)

$$a(h_r) = (1.1 \log f_c - 0.7)h_r - (1.56 \log f_c - 0.8) \quad (4.5)$$

สำหรับภูมิประเทศที่เป็นเขตเมืองหนาแน่น ค่า $a(h_r)$ จะเป็นไปตามสมการที่

(4.6)

$$a(h_r) = \begin{cases} 8.29(\log 1.54 h_r)^2 - 1.1 & : f \leq 200 \text{ MHz} \\ 3.2(\log 11.75 h_r)^2 - 4.97 & : f \geq 400 \text{ MHz} \end{cases} \quad (4.6)$$

ในหัวข้อนี้จะพิจารณาระบบที่เป็นเขตเมืองหนาแน่น เพื่อให้มีลักษณะคล้ายคลึงกับการทดสอบนิรอลเนตเวิร์กด้วยข้อมูลภาคสนามในหัวข้อที่ผ่านมา ซึ่งสภาพภูมิประเทศอยู่ในเขตกรุงเทพมหานคร ดังนั้นค่า $a(h_r)$ จะเป็นไปตามสมการที่ (4.6)

จากสมการที่ (4.3) และ (4.4) เมื่อทราบกำลังส่งของเครื่องส่งก็จะสามารถคำนวณหากำลังของสัญญาณที่มาถึงเครื่องรับได้ ซึ่งค่ากำลังที่มาถึงเครื่องรับก็คือค่าความแรงของสัญญาณนั่นเอง เมื่อพิจารณาสมการที่ (4.3) จะเห็นได้ค่าความแรงของสัญญาณที่ป้อนเป็นข้อมูลเข้าของนิรอลเนตเวิร์กสามารถใช้ค่าการสูญเสียตามระยะทางป้อนเป็นข้อมูลเข้าแทนได้เช่นกันเนื่องจากค่าความแรงของสัญญาณที่เครื่องรับจะมีค่าเท่ากับค่ากำลังส่งลบด้วยค่าการสูญเสียตามระยะทาง ดังสมการที่ (4.7)

$$|P_r| = |P_s - L| = |L - P_s| \quad \text{dB} \quad (4.7)$$

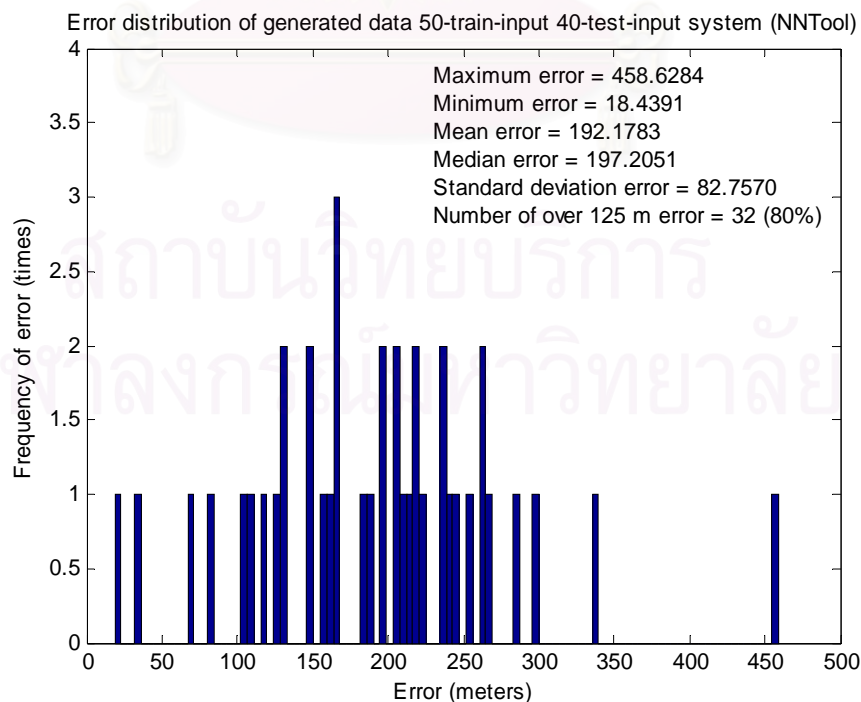
จะเห็นได้จากสมการที่ (4.7) ว่าค่ากำลังส่งของสัญญาณจากเครื่องส่งจะเป็นค่าคงที่ เสมือนค่าออฟเซต หรือ ค่าไบแอสคงที่ที่บวกเข้ากับค่าการสูญเสียตามระยะทางเมื่อไม่คิดผลของกำลังส่งซึ่งเป็นค่าคงที่ออกไป ค่าข้อมูลความแรงของสัญญาณที่

เครื่องรับ รั่วได้จะมีความสัมพันธ์อยู่กับค่าการสูญเสียตามระยะทาง ตามที่กล่าวมาข้างต้น ดังนั้นจึงสามารถใช้ค่าของการสูญเสียตามระยะทางป้อนเป็นค่าข้อมูลเข้าของนิรลเนตเวิร์กได้เช่นกันโดยรายละเอียดของการจำลองค่าความแรงของสัญญาณที่ใช้ในหัวข้อนี้จะมีลักษณะดังนี้

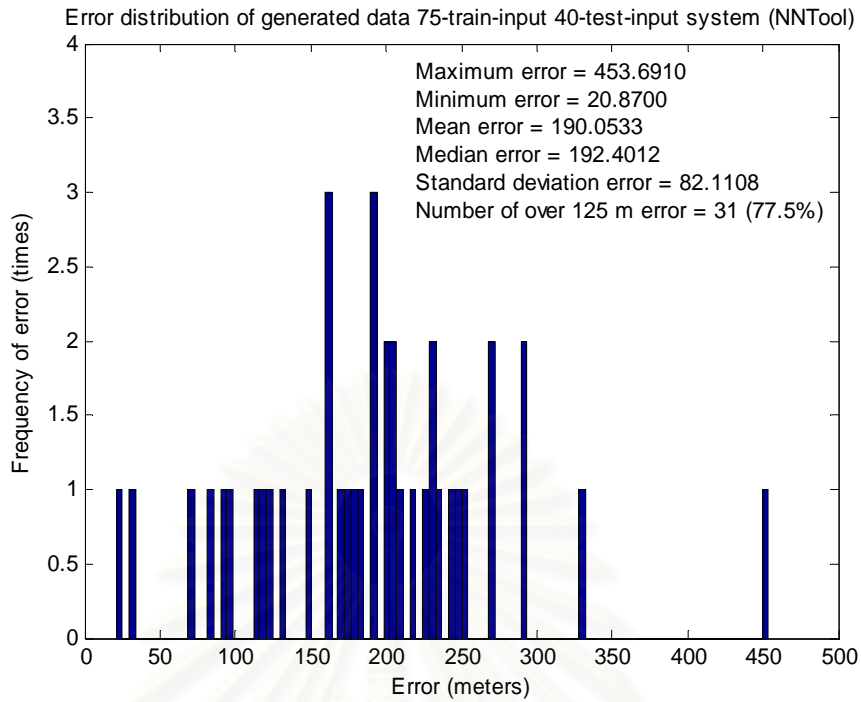
- ค่าความถี่คลื่นพาหะที่ส่ง = 900 MHz
- ค่าความสูงของเสาอากาศของสถานีฐาน = 200 m
- ค่าความสูงของเสาอากาศของสถานีเคลื่อนที่ = 2 m (รวมความสูงของผู้ใช้)
- พิจารณาระบบที่เป็นเมืองขนาดเล็กจนถึงขนาดกลาง

3) การทดสอบนิรลเนตเวิร์กด้วยข้อมูลจำลองแบ่งได้เป็น 3 กรณี เช่นเดียวกับการทดสอบนิรลเนตเวิร์กด้วยข้อมูลภาคสนาม ได้แก่ กรณีที่ 1 การทดสอบกรณีมีชุดข้อมูลจำลองสำหรับฝึกนิรลเนตเวิร์กจำนวน 50 ชุด และสำหรับทดสอบนิรลเนตเวิร์กจำนวน 40 ชุด กรณีที่ 2 การทดสอบกรณีมีชุดข้อมูลจำลองสำหรับฝึกนิรลเนตเวิร์กจำนวน 75 ชุด และสำหรับทดสอบนิรลเนตเวิร์กจำนวน 40 ชุด และกรณีที่ 3 การทดสอบกรณีมีชุดข้อมูลจำลองสำหรับฝึกนิรลเนตเวิร์กจำนวน 100 ชุด และสำหรับทดสอบนิรลเนตเวิร์กจำนวน 40 ชุด

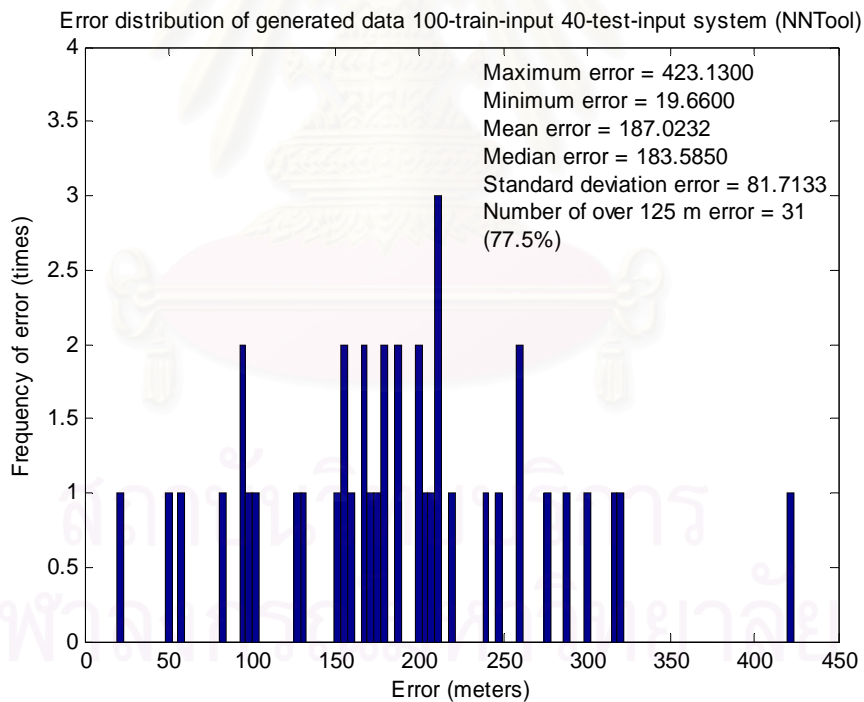
ผลการทดสอบนิรลเนตเวิร์กด้วยข้อมูลจำลองในกรณีที่ 1 กรณีที่ 2 และกรณีที่ 3 โดยใช้ NNTool สามารถแสดงได้ดังรูปที่ 4.53 รูปที่ 4.54 และรูปที่ 4.55 ตามลำดับ



รูปที่ 4.53 ค่าความผิดพลาดกรณีชุดข้อมูลจำลองสำหรับฝึก 50 ชุด และสำหรับทดสอบ 40 ชุด



รูปที่ 4.54 ค่าความผิดพลาดกรณีชุดข้อมูลจำลองสำหรับฝึก 75 ชุด และสำหรับทดสอบ 40 ชุด



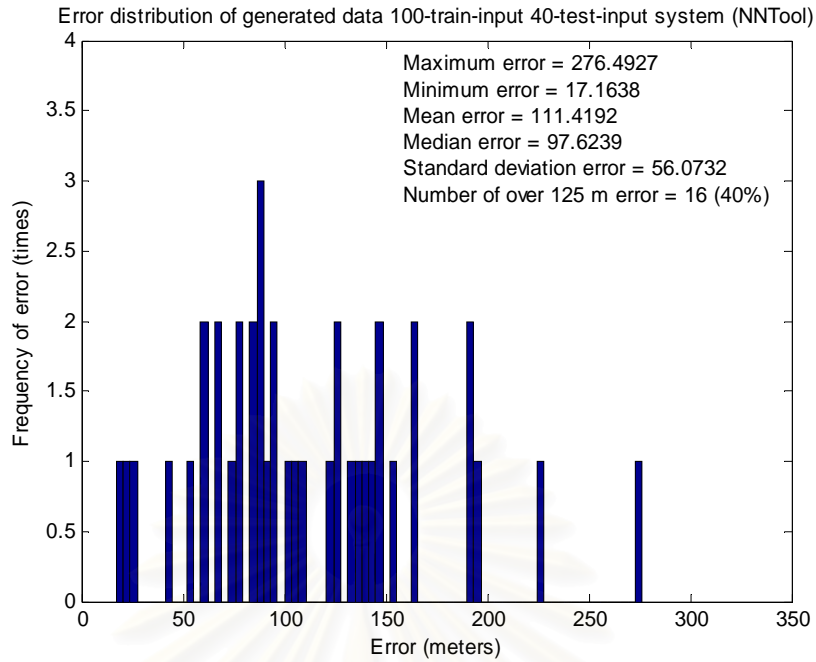
รูปที่ 4.55 ค่าความผิดพลาดกรณีชุดข้อมูลจำลองสำหรับฝึก 100 ชุด และสำหรับทดสอบ 40 ชุด

จากรูปที่ 4.53 4.54 และ 4.55 จะเห็นว่าเมื่อชุดข้อมูลสำหรับฝึกและทดสอบนิวรอลเน็ตเวิร์กมีการเปลี่ยนแปลง ในที่นี้คือการเปลี่ยนแปลงรูปแบบของชุดข้อมูลสำหรับฝึกและทดสอบนิวรอลเน็ตเวิร์กจากชุดข้อมูลที่มีลักษณะเป็นแบบรูปเป็นชุดข้อมูลที่มีลักษณะกระจายตัวอย่างสม่ำเสมอ ถึงแม้การเพิ่มชุดข้อมูลสำหรับฝึกนิวรอลเน็ตเวิร์ก จะทำให้นิวรอลเน็ตเวิร์กประมาณค่า

ได้ถูกต้องมากขึ้น แต่อย่างไรก็ตาม นีวรอลเน็ตเวิร์กไม่สามารถประมาณค่าตำแหน่งของสถานีเคลื่อนที่ได้อย่างถูกต้องตามข้อกำหนดของ FCC โดยค่าความผิดพลาดเฉลี่ยที่เกิดขึ้นเท่ากับ 192.1783 เมตร 190.0533 เมตร และ 187.0232 เมตร ในการทดสอบกรณีที่ 1 กรณีที่ 2 และกรณีที่ 3 ตามลำดับ และมีค่าความผิดพลาดเกิน 125 เมตรมากกว่าที่ FCC กำหนดไว้ทั้ง 3 กรณี สาเหตุเนื่องมาจาก จำนวนนิวรอนในชั้นซ่อนเร้นและค่าอัตราการเรียนรู้ของนีวรอลเน็ตเวิร์กที่นำมาใช้ในการประมาณค่าถูกหามาจากการใช้ข้อมูลภาคสนามที่มีลักษณะเป็นแบบรูป ไม่ได้ถูกทำให้เหมาะสมกับชุดข้อมูลจำลองที่มีลักษณะกระจายตัวเช่นนี้ สาเหตุอีกประการหนึ่งอาจเกิดจากวิธีการฝึกนีวรอลเน็ตเวิร์กที่ไม่เหมาะสม ดังนั้นหากต้องการลดค่าความผิดพลาดที่เกิดขึ้นในการทดสอบด้วยข้อมูลจำลองนี้ จำเป็นต้องมีการทดสอบเพื่อหาจำนวนนิวรอนในชั้นซ่อนเร้นและค่าอัตราการเรียนรู้ที่เหมาะสมใหม่ หรืออาจต้องเปลี่ยนวิธีการฝึกนีวรอลเน็ตเวิร์กเป็นวิธีการฝึกแบบปรับค่าครั้งเดียว

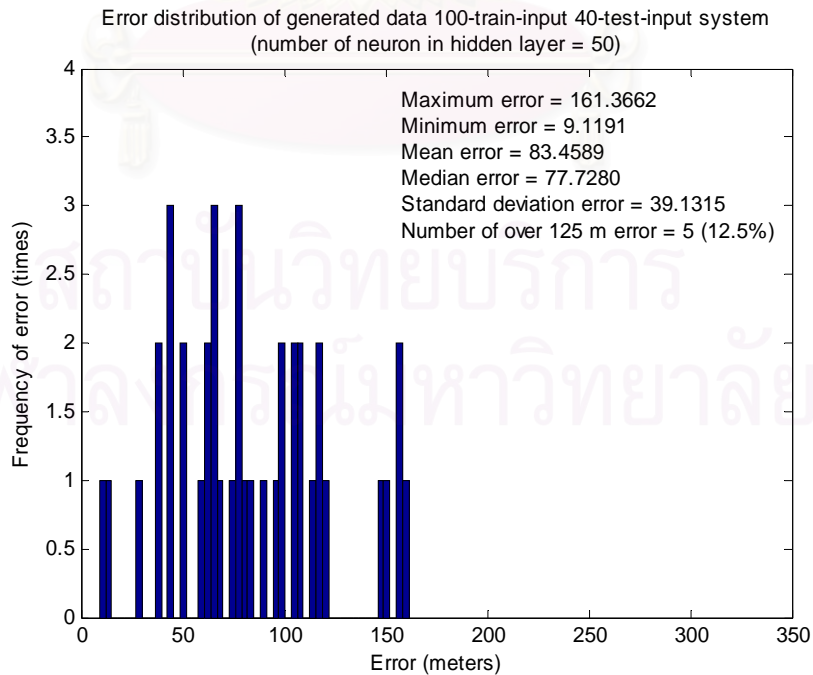
สำหรับนีวรอลเน็ตเวิร์กเพื่อประมาณค่าตำแหน่งของสถานีเคลื่อนที่ที่ออกแบบด้วยภาษาวีเอชดีแอลนั้น จากการฝึกและทดสอบนีวรอลเน็ตเวิร์กด้วยชุดข้อมูลจำลองปรากฏว่า ในการฝึกนีวรอลเน็ตเวิร์ก นีวรอลเน็ตเวิร์กไม่สามารถปรับค่าถ่วงน้ำหนักและค่าไบแอสให้คงที่ที่ค่า ๆ หนึ่งได้ จึงไม่สามารถหาค่าถ่วงน้ำหนักและค่าไบแอสที่เหมาะสมมาทดสอบนีวรอลเน็ตเวิร์กได้ สาเหตุเนื่องมาจากสมมุติฐานหลาย ๆ ประการดังที่กล่าวมาแล้วเช่นเดียวกัน

จากสมมุติฐานที่ทำให้นีวรอลเน็ตเวิร์กไม่สามารถประมาณค่าตำแหน่งของสถานีเคลื่อนที่ได้ถูกต้องมากขึ้น จึงได้ลองทดสอบโดยใช้ NNTool เพื่อหาจำนวนนิวรอนในชั้นซ่อนเร้นและค่าอัตราการเรียนรู้ที่เหมาะสมสำหรับการนำชุดข้อมูลจำลองมาฝึกและทดสอบนีวรอลเน็ตเวิร์กใหม่อีกครั้งหนึ่ง ผลปรากฏว่า ไม่ว่าจะเปลี่ยนจำนวนนิวรอนในชั้นซ่อนเร้นและปรับค่าอัตราการเรียนรู้เท่าใดก็ตาม ไม่สามารถทำให้นีวรอลเน็ตเวิร์กสามารถประมาณค่าตำแหน่งของสถานีเคลื่อนที่ได้ถูกต้องมากขึ้น จึงได้ลองเปลี่ยนวิธีการฝึกนีวรอลเน็ตเวิร์กจากวิธีการฝึกแบบปรับค่าตลอดเวลาเป็นวิธีการฝึกแบบปรับค่าครั้งเดียว และทดสอบประสิทธิภาพในการประมาณค่าของนีวรอลเน็ตเวิร์กในกรณีใช้ชุดข้อมูลจำลองสำหรับฝึก 100 ชุด และสำหรับทดสอบ 40 ชุดอีกครั้งหนึ่ง ค่าความผิดพลาดในการประมาณค่าที่เกิดขึ้นสามารถแสดงได้ดังรูปที่ 4.56 โดยนีวรอลเน็ตเวิร์กมีค่าความผิดพลาดเฉลี่ยในการประมาณค่าเท่ากับ 111.4192 เมตร และเกิดจำนวนครั้งที่มีความผิดพลาดมากกว่า 125 เมตร จำนวน 16 ครั้ง (40%) เมื่อเปรียบเทียบกับนีวรอลเน็ตเวิร์กที่ถูกฝึกด้วยวิธีการฝึกแบบปรับค่าตลอดเวลาซึ่งมีค่าความผิดพลาดแสดงดังรูปที่ 4.55 โดยนีวรอลเน็ตเวิร์กมีค่าความผิดพลาดเฉลี่ยในการประมาณค่าเท่ากับ 187.0232 เมตร และเกิดจำนวนครั้งที่มีความผิดพลาดมากกว่า 125 เมตร จำนวน 31 ครั้ง (75%) จะเห็นว่านีวรอลเน็ตเวิร์กที่ถูกฝึกด้วยวิธีการฝึกแบบปรับค่าครั้งเดียวมีความสามารถในการประมาณค่าที่ดีกว่ามาก

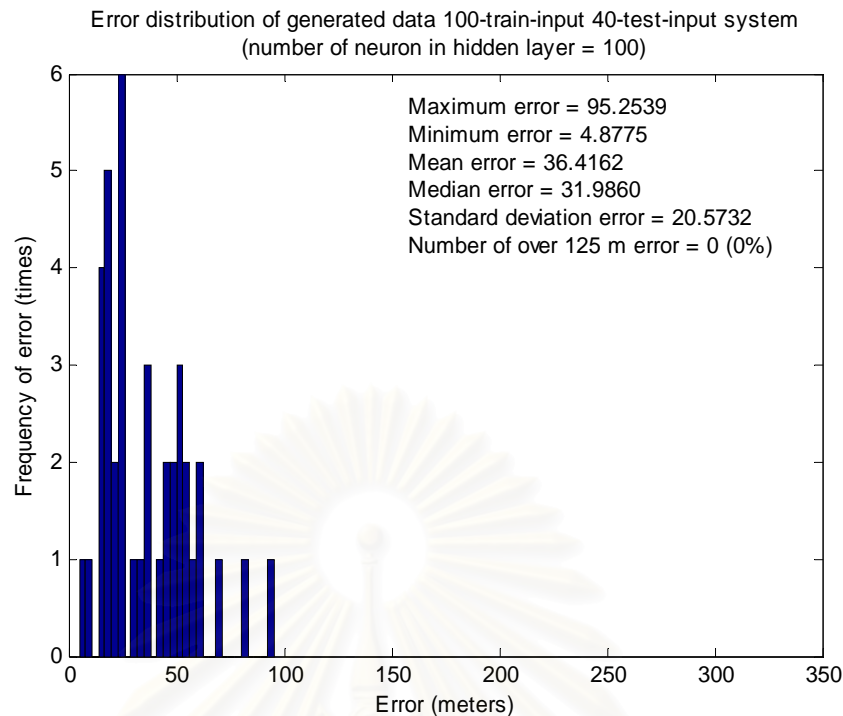


รูปที่ 4.56 ค่าความผิดพลาดกรณีชุดข้อมูลจำลองสำหรับฝึก 100 ชุด และสำหรับทดสอบ 40 ชุด เมื่อเปลี่ยนวิธีการฝึกนิวรอลเน็ตเวิร์กเป็นแบบปรับค่าครั้งเดียว

อย่างไรก็ตาม จากรูปที่ 4.56 นิวรอลเน็ตเวิร์กยังไม่สามารถประมาณค่าตำแหน่งของสถานีเคลื่อนที่ได้ถูกต้องตามข้อกำหนดของ FCC ทั้งนี้เนื่องจากจำนวนนิวรอนในชั้นซ่อนเร้นที่ไม่เหมาะสม จึงได้ลองเพิ่มจำนวนนิวรอนในชั้นซ่อนเร้นจาก 5 นิวรอนเป็น 50 นิวรอน และ 100 นิวรอน ซึ่งค่าความผิดพลาดในการประมาณค่าแสดงได้ดังรูปที่ 4.57 และ 4.58 ตามลำดับ



รูปที่ 4.57 ค่าความผิดพลาดเมื่อเปลี่ยนวิธีการฝึกนิวรอลเน็ตเวิร์กเป็นแบบปรับค่าครั้งเดียว และเพิ่มจำนวนนิวรอนในชั้นซ่อนเร้นเป็น 50 นิวรอน



รูปที่ 4.58 ค่าความผิดพลาดเมื่อเปลี่ยนวิธีการฝึกนิวรอลเน็ตเวิร์กเป็นแบบปรับค่าครั้งเดียว และเพิ่มจำนวนนิวรอนในชั้นซ่อนเร้นเป็น 100 นิวรอน

จากรูปที่ 4.57 และ 4.58 จะเห็นว่าเมื่อเพิ่มจำนวนนิวรอนในชั้นซ่อนเร้นเป็น 50 นิวรอน และ 100 นิวรอน (ค่าอัตราการเรียนรู้ในกรณีนี้เท่ากับ 0.001 ไม่มีการเปลี่ยนแปลง) นิวรอลเน็ตเวิร์กที่ถูกฝึกด้วยวิธีฝึกแบบปรับค่าครั้งเดียวสามารถประมาณค่าได้ถูกต้องมากขึ้น โดยมีค่าความผิดพลาดเฉลี่ยเท่ากับ 83.4589 เมตร และ 36.4162 เมตร ตามลำดับ ส่วนค่าความผิดพลาดที่มากกว่า 125 เมตรเกิดขึ้น 5 ครั้ง (12.5%) และ 0 ครั้ง (0%) ตามลำดับ

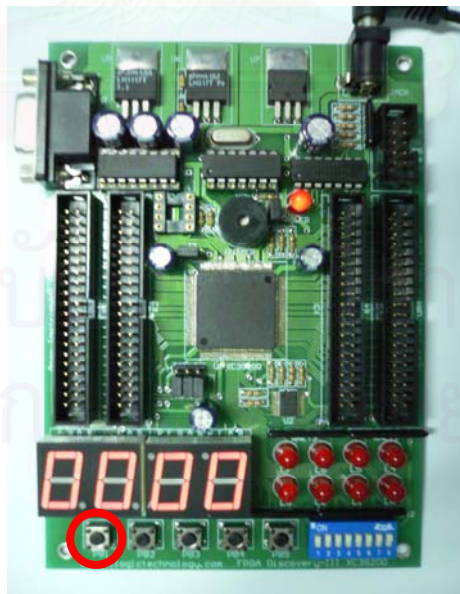
จากการทดสอบนิวรอลเน็ตเวิร์กด้วยชุดข้อมูลจำลองทั้งหมดที่กล่าวมา ทำให้ได้ข้อสังเกตว่า สำหรับชุดข้อมูลที่มีลักษณะกระจายตัว วิธีการฝึกนิวรอลเน็ตเวิร์กด้วยวิธีฝึกแบบปรับค่าครั้งเดียว น่าจะทำให้นิวรอลเน็ตเวิร์กสามารถประมาณค่าตำแหน่งของสถานีเคลื่อนที่ได้ถูกต้องมากกว่า ซึ่งจะได้นำไปแก้ไขและพัฒนานิวรอลเน็ตเวิร์กที่ออกแบบเพื่อให้สามารถประมาณค่าตำแหน่งของสถานีเคลื่อนที่ในสถานการณ์จริง ซึ่งชุดข้อมูลมีลักษณะกระจายตัวได้ถูกต้องมากยิ่งขึ้น

4.4 ผลการทดสอบอุปกรณ์ต้นแบบเพื่อประมาณค่าตำแหน่งของสถานีเคลื่อนที่

การทดสอบหัวข้อนี้ เป็นการทดสอบความสามารถในการประมาณค่าของอุปกรณ์ต้นแบบ เพื่อประมาณค่าตำแหน่งของสถานีเคลื่อนที่ โดยใช้ชุดข้อมูลภาคสนามในการทดสอบ ซึ่งอุปกรณ์ต้นแบบถูกสร้างขึ้นมาโดยนำวงจรบางส่วนของนิวรอลเน็ตเวิร์กที่ออกแบบด้วยภาษาวีเซตดีแอลมา

ดาวนโหลดลงบอร์ดเอฟพีจีเอรุ่น Discovery XC3S200 ของบริษัท เอเพก อินสตรูเมนต์ จำกัด ทั้งนี้เนื่องจากข้อจำกัดทางด้านฮาร์ดแวร์ ดังที่ได้กล่าวไว้แล้วในบทที่ 3 หัวข้อที่ 3.3.2 จึงไม่สามารถดาวนโหลดวงจรทั้งหมดลงบนบอร์ดได้

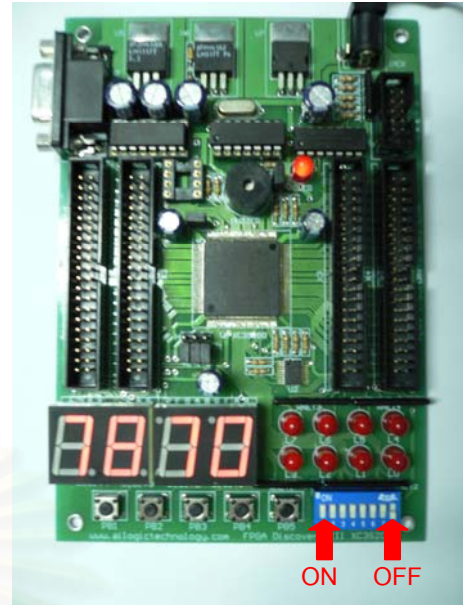
ก่อนเริ่มต้นการประมาณค่าตำแหน่งของสถานีเคลื่อนที่ควรกดปุ่ม rst ทุกครั้ง และเมื่อกดปุ่ม rst อุปกรณ์ต้นแบบจะแสดงค่า “0000” ดังแสดงในรูปที่ 4.59 ในช่วงเริ่มต้น สวิตช์ lat_sel และ int_sel ทั้งสองสวิตช์ จะถูกตั้งไว้ที่ตำแหน่ง ON ดังนั้นเมื่อกดปุ่ม start เพื่อเริ่มต้นการประมาณค่า อุปกรณ์ต้นแบบจะแสดงค่าจำนวนเต็มของตำแหน่งละติจูด ได้แก่ “0018” ดังแสดงในรูปที่ 4.60 และเมื่อต้องการตรวจสอบค่าทศนิยมของตำแหน่งละติจูด สามารถทำได้โดยเลื่อนสวิตช์ int_sel ไปที่ตำแหน่ง OFF อุปกรณ์ต้นแบบจะแสดงค่าทศนิยมของตำแหน่งละติจูด ได้แก่ “7870” ดังแสดงในรูปที่ 4.61 เมื่อต้องการตรวจสอบค่าจำนวนเต็มของตำแหน่งลองจิจูด สามารถทำได้โดยเลื่อนสวิตช์ int_sel กลับไปที่ตำแหน่ง ON และเลื่อนสวิตช์ lat_sel ไปที่ตำแหน่ง OFF อุปกรณ์ต้นแบบจะแสดงค่าจำนวนเต็มของตำแหน่งลองจิจูด ได้แก่ “0098” ดังแสดงในรูปที่ 4.62 และเมื่อต้องการตรวจสอบค่าทศนิยมของตำแหน่งลองจิจูด สามารถทำได้โดยเลื่อนสวิตช์ int_sel ไปที่ตำแหน่ง OFF อีกครั้ง อุปกรณ์ต้นแบบจะแสดงค่าทศนิยมของตำแหน่งลองจิจูด ได้แก่ “9869” ดังแสดงในรูปที่ 4.63 นั่นคืออุปกรณ์ต้นแบบได้ประมาณค่าตำแหน่งของสถานีเคลื่อนที่อยู่ที่ตำแหน่งละติจูด 18.787 องศาเหนือ และลองจิจูด 98.9869 องศาตะวันออก โดยที่ค่าเป้าหมายในการทดสอบครั้งนี้อยู่ที่ละติจูด 18.787 องศาเหนือ และลองจิจูด 98.987 องศาตะวันออก



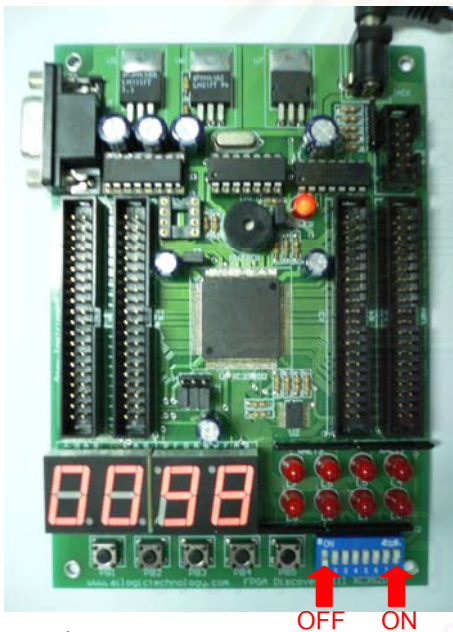
รูปที่ 4.59 การแสดงผลเมื่อกดปุ่ม rst



รูปที่ 4.60 การแสดงผลค่าจำนวนเต็ม
ของตำแหน่งละติจูด



รูปที่ 4.61 การแสดงผลค่าจำนวนทศนิยม
ของตำแหน่งละติจูด



รูปที่ 4.62 การแสดงผลค่าจำนวนเต็ม
ของตำแหน่งลองจิจูด



รูปที่ 4.63 การแสดงผลค่าจำนวนทศนิยม
ของตำแหน่งลองจิจูด

จากผลการทดสอบแสดงให้เห็นว่า นิรอรอลเน็ตเวิร์กที่ออกแบบโดยใช้ภาษาวีเอชดีแอล สามารถนำมาพัฒนาเป็นอุปกรณ์ต้นแบบเพื่อประมาณค่าตำแหน่งของสถานีเคลื่อนที่ได้จริง และสามารถประมาณค่าได้ถูกต้องตรงตามข้อกำหนดของ FCC อย่างไรก็ตาม เนื่องจากอุปกรณ์ต้นแบบที่ได้ทดลองสร้างขึ้นนั้น เป็นเพียงส่วนหนึ่งของนิรอรอลเน็ตเวิร์กทั้งหมดที่ออกแบบ จึงยังไม่สามารถทำงานได้อย่างสมบูรณ์ จึงต้องอาศัยเวลาในการพัฒนาประสิทธิภาพต่อไปในอนาคต

บทที่ 5

บทสรุปและข้อเสนอแนะ

บทนี้กล่าวถึงบทสรุปของการออกแบบและทดสอบนิรอลเน็ตเวิร์กเพื่อประมาณค่าตำแหน่งของสถานีเคลื่อนที่ รวมทั้งการพัฒนาอุปกรณ์ต้นแบบ และข้อเสนอแนะเพิ่มเติม

5.1 บทสรุป

วิทยานิพนธ์ฉบับนี้เสนอการออกแบบนิรอลเน็ตเวิร์กเพื่อประมาณค่าตำแหน่งของสถานีเคลื่อนที่ โดยมุ่งเน้นให้นิรอลเน็ตเวิร์กที่ออกแบบสามารถนำมาใช้งานได้จริง จึงได้ออกแบบโดยใช้ภาษาวีเอสดีแอลและดาวนโหลดวงจรที่ออกแบบบนบอร์ดเอฟพีจีเอ โดยแบ่งการทดสอบออกเป็น 2 ส่วน ได้แก่ ส่วนที่หนึ่ง การทดสอบความถูกต้องของนิรอลเน็ตเวิร์กที่ออกแบบ โดยแสดงผลการจำลองแบบผ่านโปรแกรม ModelSim XE และนำผลที่ได้มาเปรียบเทียบกับเครื่องมือนิรอลเน็ตเวิร์ก (NNTool) ที่มีอยู่ในโปรแกรม Matlab ซึ่งชุดข้อมูลสำหรับฝึกและทดสอบนิรอลเน็ตเวิร์กมีทั้งข้อมูลสมมุติที่จำลองขึ้น และข้อมูลจริงที่ได้มาจากการเก็บข้อมูลภาคสนาม จากนั้นจึงเข้าสู่การทดสอบส่วนที่สอง นั่นคือการทดสอบความสามารถในการประมาณค่าตำแหน่งของสถานีเคลื่อนที่ ซึ่งถูกสร้างขึ้นโดยดาวนโหลดวงจรที่ออกแบบลงบนบอร์ดเอฟพีจีเอ จากผลการทดสอบในบทที่ 4 สามารถสรุปได้ดังนี้

1. จำนวนนิรอลเน็ตเวิร์กในชั้นซ่อนเร้นของนิรอลเน็ตเวิร์ก และค่าอัตราการเรียนรู้ของนิรอลเน็ตเวิร์ก รวมถึงวิธีการฝึกนิรอลเน็ตเวิร์ก เป็นพารามิเตอร์สำคัญที่มีความเกี่ยวข้องกับการลู่เข้าของนิรอลเน็ตเวิร์ก ทั้งนี้จำนวนนิรอลเน็ตเวิร์กในชั้นซ่อนเร้น ค่าอัตราการเรียนรู้ และวิธีการฝึกที่เหมาะสมจะแตกต่างกันตามระบบที่ต้องการนำนิรอลเน็ตเวิร์กไปประยุกต์ใช้ ซึ่งหากนำไปใช้อย่างไม่เหมาะสม อาจทำให้การฝึกนิรอลเน็ตเวิร์กไม่เกิดการลู่เข้า ส่งผลให้การประมาณค่าของนิรอลเน็ตเวิร์กมีความผิดพลาดสูงได้

2. เมื่อระบบที่นำนิรอลเน็ตเวิร์กไปใช้ในการประมาณค่ามีการเปลี่ยนแปลง เช่น มีการเปลี่ยนแปลงลักษณะของชุดข้อมูลสำหรับฝึกและทดสอบนิรอลเน็ตเวิร์กจากชุดข้อมูลที่มีลักษณะเป็นแบบรูป (Pattern) เป็นชุดข้อมูลที่มีลักษณะกระจาย จำเป็นต้องมีการทดสอบเพื่อหาจำนวนนิรอลเน็ตเวิร์กและค่าอัตราการเรียนรู้ที่เหมาะสม รวมถึงวิธีการฝึกนิรอลเน็ตเวิร์กใหม่ทุกครั้ง

3. วิธีการฝึกนิรอลเน็ตเวิร์กแบบปรับค่าตลอดเวลาเหมาะสำหรับการนำไปใช้กับชุดข้อมูลที่มีลักษณะเป็นแบบรูป และควรฝึกนิรอลเน็ตเวิร์กโดยจัดชุดข้อมูลสำหรับฝึกที่เคลื่อนที่จากจุดหนึ่งไปยังอีกจุดหนึ่งโดยไม่มีการสลับตำแหน่ง

4. วิธีการฝึกนิรอลเน็ตเวิร์กแบบปรับค่าครั้งเดียวเหมาะสำหรับการนำไปใช้กับชุดข้อมูลที่มีลักษณะกระจายตัว แต่ใช้เวลาในการฝึกนานกว่า และทำให้ระบบมีความซับซ้อนกว่าวิธีการฝึกแบบปรับค่าตลอดเวลา

5. ความถูกต้องในการประมาณค่าตำแหน่งของสถานีเคลื่อนที่ขึ้นอยู่กับจำนวนชุดข้อมูลสำหรับฝึกนิรอลเน็ตเวิร์กเป็นสำคัญ หากชุดข้อมูลสำหรับฝึกยังมีจำนวนมาก ยิ่งทำให้นิรอลเน็ตเวิร์กสามารถรู้จำสภาพภูมิประเทศโดยรวม และประมาณค่าตำแหน่งของสถานีเคลื่อนที่ได้ถูกต้องมากยิ่งขึ้น

6. นิรอลเน็ตเวิร์กที่นำเสนอสามารถประมาณค่าตำแหน่งของสถานีเคลื่อนที่ในกรณีที่สุดข้อมูลสำหรับฝึกและทดสอบนิรอลเน็ตเวิร์กมีลักษณะเป็นแบบรูป โดยให้ค่าความผิดพลาดใกล้เคียงกับการประมาณค่าโดยใช้ NNTool ซึ่งเป็นเครื่องมือที่ได้รับการยอมรับและมีความน่าเชื่อถือ และสามารถประมาณค่าตำแหน่งของสถานีเคลื่อนที่ได้ถูกต้องตามข้อกำหนดของ FCC โดยไม่มีการประมาณค่าครั้งใดที่มีความผิดพลาดมากกว่า 125 เมตร

7. นิรอลเน็ตเวิร์กที่ออกแบบสามารถนำมาพัฒนาเป็นเครื่องต้นแบบเพื่อประมาณค่าตำแหน่งของสถานีเคลื่อนที่ได้จริง และเครื่องต้นแบบดังกล่าวสามารถประมาณค่าตำแหน่งของสถานีเคลื่อนที่โดยมีค่าความผิดพลาดอยู่ในขอบเขตที่ FCC กำหนดไว้

5.2 ข้อเสนอแนะ

1. ถึงแม้ว่านิรอลเน็ตเวิร์กที่นำเสนอจะสามารถประมาณค่าตำแหน่งของสถานีเคลื่อนที่ได้ถูกต้องตามข้อกำหนดของ FCC แต่อย่างไรก็ตาม ค่าความผิดพลาดในการประมาณค่ายังคงสูงอยู่ หากมีชุดข้อมูลภาคสนามสำหรับฝึกนิรอลเน็ตเวิร์กมากขึ้นกว่าที่มีอยู่ ณ ปัจจุบัน จะทำให้นิรอลเน็ตเวิร์กสามารถประมาณค่าตำแหน่งได้ถูกต้องมากยิ่งขึ้น

2. ในการพัฒนาความสามารถในการประมาณค่าตำแหน่งของสถานีเคลื่อนที่ของนิรอลเน็ตเวิร์กที่ออกแบบ ควรนำข้อมูลภาคสนามที่มีลักษณะกระจายตัวมากขึ้นมาใช้ฝึกและทดสอบนิรอลเน็ตเวิร์ก เนื่องจากในความเป็นจริง หากมีการนำนิรอลเน็ตเวิร์กที่ออกแบบไปใช้สำหรับประมาณค่าตำแหน่งของสถานีเคลื่อนที่ ตำแหน่งของสถานีเคลื่อนที่จะไม่มีลักษณะเป็นแบบรูป ดังเช่นข้อมูลภาคสนามที่ใช้ในวิทยานิพนธ์ฉบับนี้

3. นิรอลเน็ตเวิร์กที่ออกแบบเหมาะสำหรับการนำไปใช้กับชุดข้อมูลที่มีลักษณะเป็นแบบรูป จึงควรพัฒนาให้สามารถนำไปใช้กับข้อมูลทุกรูปแบบ ซึ่งอาจต้องปรับปรุงให้นิรอลเน็ตเวิร์กที่ออกแบบสามารถฝึกนิรอลเน็ตเวิร์กได้ทั้งแบบปรับค่าตลอดเวลา และแบบปรับค่าครั้งเดียว รวมถึงสามารถเพิ่มหรือลดจำนวนนิรอลเน็ตเวิร์กในชั้นซ่อนเร้น โดยไม่ต้องแก้ไขวงจรมากนัก

4. อุปกรณ์ต้นแบบที่ทดลองสร้างขึ้นเป็นเพียงการดาวน์โหลดวงจรบางส่วนของนิรอลเน็ตเวิร์กทั้งหมดที่ออกแบบด้วยภาษาวีเอชดีแอลลงบนบอร์ดเอฟพีจีเอเท่านั้น ซึ่งได้แก่วงจรในส่วนที่ใช้สำหรับการทดสอบนิรอลเน็ตเวิร์ก แต่ยังไม่มีส่วนที่ใช้สำหรับฝึกนิรอลเน็ตเวิร์ก เนื่องจากข้อจำกัดด้านฮาร์ดแวร์ของบอร์ดที่ไม่สามารถดาวน์โหลดทั้งหมดลงไปได้ จึงต้องมีการศึกษาเพิ่มเติมเพื่อพัฒนาเป็นอุปกรณ์สำหรับประมวลผลค่าตำแหน่งของสถานีเคลื่อนที่ที่สมบูรณ์ต่อไปในอนาคต



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

1. J. J. Caffery, Jr. L. Stuber and G. L. Stuber. Overview of Radiolocation in CDMA Cellular Systems. IEEE Communications Magazine. 36, (Apr. 1998) : 38-45.
2. S. S. Wang, M. Green and M. Malkawi. E-911 Location Standards and Location Commercial Services. Emerging Technologies Symposium: Broadband, Wireless Internet Access, 2000 IEEE. (2000) : 1-5.
3. T. S. Rappaport, J. H. Reed and B. D. Woerner. Position Location Using Wireless Communications on Highways of the Future. IEEE Communication Magazine. 34, (Oct. 1996) : 33-41.
4. ปิติ เล็กอุทัย. การประมาณค่าตำแหน่งของสถานีเคลื่อนที่โดยใช้นิวรอลเน็ตเวิร์ก. วิทยานิพนธ์ปริญญาโทบริหารธุรกิจ สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2544.
5. C. Lau. Neural Networks Theoretical Foundations and Analysis. New York : IEEE PRESS, 1992.
6. Robert J. Schalkoff Pattern Recognition: Statistical, Structural and Neural Approaches. USA : John Wiley & Sons, 1992.
7. ชำนาญ ปัญญาไส และ วัชรกร หนูทอง. ภาษา VHDL สำหรับการออกแบบวงจรดิจิทัล. ซีเอ็ดยูเคชั่น, 2547.
8. N. Shirazi, A. Walters and P. Athanas. Quantitative Analysis of Floating Point Arithmetic on FPGA Based Custom Computing Machines. IEEE Symposium on FPGAs for Custom Computing Machines. (Apr. 1995) : 155 – 162.
9. A. Steven and Mario J. Gonzalez. A Neural Network Implementation Using Reconfigurable Architectures. Department of Electrical and Computer Engineering, University of Texas at Austin, USA.
10. J. D. Parsons. The Mobile Radio Propagation Channel. London : Pentech Press, 1992.



ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทความทางวิชาการที่ได้รับการเผยแพร่

1. W. Benjapolakul, P. Lek-uthai and S. Kietsiroje. Locating Mobile Station Position Using Neural Network. ECTI 2005 (May 2005): 287-290.



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Locating Mobile Station Position Using Neural Network

Watit Benjapolakul, Piti Lek-uthai, and Saran Kietsiroje

Department of Electrical Engineering, Faculty of Engineering,
Chulalongkorn University, Bangkok 10330, Thailand

E-mail: watit.b@eng.chula.ac.th, pitilek@hotmail.com, saran_k360@hotmail.com

ABSTRACT

Wireless Position Location has received an enormous attention during this decade. Recently, the Federal Communications Commission (FCC) developed requirements for enhanced wireless 9-1-1 service (E-911) which state that in July 1996 all wireless service providers are required to provide location information for E-911 public safety service. Moreover, FCC also requires that by October 2001, operators must be able to determine accurate location of mobile stations (MSs) which request emergency assistance with accuracy within the radius of 125 meters to achieve at least 67 percent of all trial locations. This paper presents an application of neural network technique to approximate the location of MS with the measurements of the Signal Strength (SS) and the Time of Arrival (TOA) from several base stations (BSs).

1. INTRODUCTION

Position Location services in mobile telephone systems have received increased attention in the past decades. The applications are both included in commercial and governmental interests. Recently, the Federal Communications Commission (FCC) has launched the specification for E-911 service [1]. These requirements have been proposed for two-phase implementation timeline. Phase 1 requires that the wireless network provides the user's call-back number and the geographic area associated with the cell/sector handling the call. Phase 2 requires that the caller's location is reported with a much higher degree of accuracy. For phase 2, the FCC mandate requires wireless operators to locate the position of emergency callers with a location error lower than 125 meters for 67% of the time from October 2001 to December 2005.

Position estimation technologies that have been proposed to meet FCC 94-102 fall into two broad categories:

- Network-based solutions by which network itself calculates the position of MS from the received signal measurements.
- Handset-based solutions which rely on a modified handset to calculate its own position. This technique includes a GPS receiver embedded in the handset and

requires a return data path to report the location to the network for relaying to the emergency operator.

Several methods [2,3] have been proposed to perform location estimation of MS based on triangulation signal measurements as follows:

- Global Positioning System (GPS) in the mobile telephones which use three satellites to provide the measurements.
- Angle of Arrival (AOA). This method estimates the MS location by measuring the AOAs of a signal from an MS at several BSs through antenna arrays or sectors antenna (Figure 1). However, there are still the effects of Non-Line of Sight (NLOS) and multipath interfering with the angle measurements.
- Time of Arrival (TOA). In this approach, the distance between an MS and a BS is calculated by the time measurements (one-way propagation time between an MS and a BS). Geometrically, this provides a circle, centered at the BS, on which the MS must lie. For two dimensional position (x, y) , it requires at least three BSs to resolve the ambiguities, and the location of the MS is at the intersection of the circles shown in Figure 2(a).
- Time Difference of Arrival (TDOA). This method uses the differences in the TOAs which define a hyperbola, with foci at the BSs, on which the MS must lie as shown in Figure 2(b). The advantage of using TDOA method is that the synchronization between any BSs is not necessary. However, in time-based measurements, it should be noted that the effects of NLOS and multipath are still the main problem to achieve high accuracy.
- Signal Strength (SS). The location of the MS is calculated from the signal strength (in dB) of the MS received at the BSs. With the received signal strength, it is possible to calculate the propagation distance which is obtained from the path loss (the difference between the received signal power (measured) and the transmitted signal power). With propagation distance and probabilistic calculation [4], the expectation of the position of MS (which is the estimated location the network calculates) is calculated and reported. Due to the hearability problem caused by power control in cellular CDMA systems, less than three BSs can receive the signal from the same MS. Only one or two, instead of three SS informations are obtained. Thus, it may be insufficient to calculate for the MS position.

The main problems for radiolocation methods are multipath propagation, NLOS signal component, multiple access interference and shadowing. For this reason, using only one measurement to calculate for the location of MS (such as SS, TOA, or AOA etc.) with high accuracy seems to be difficult. Thus, the combinations of different radiolocation measurements have been proposed.

However, with the combinations of different measurements and the mathematical calculation of non-linear equation during the process, some linearizations of non-linear functions (such as using Taylor's series) make the approximation of the mobile location not highly accurate enough for least square error approximation (LSE).

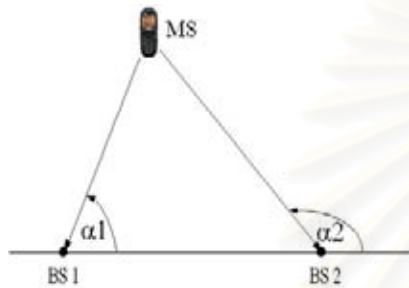


Fig. 1: Angle of Arrival (AOA) method

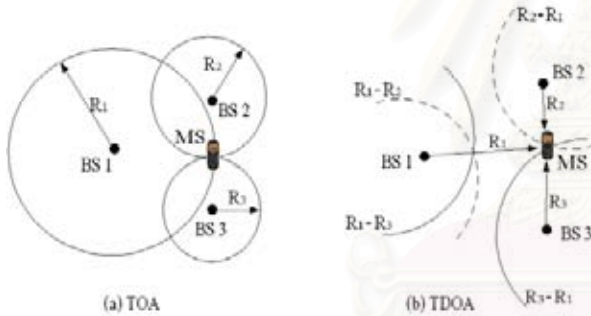


Fig. 2: Time-based positioning method

This paper presents an idea of combining the information of signal strength and time of arrival to be used in neural network and let neural network calculate the position of MSs with high accuracy. Section 2 describes the neural network structure and the algorithm for locating MS position. Sections 3 and 4 show the simulation results of the performance of the proposed algorithm, and summary, respectively.

2. NEURAL NETWORK

Neural network is widely used for the non-linear problem, and also is a robust tool for applications like classification, recognition and approximation of functions. The architecture [5, 6] of a simple neural network is shown in Figure 3. The inputs which are some features of the object or function to be estimated are fed into the neuron where the weights must be multiplied to the input and then each of them is passed through a summation function to be a net-input (bias is sometimes

included). The net-input then is fed to a specific transfer function. This transfer function acts as a classifier which groups or classifies each input into region (called mapping function). The transfer functions used in neural network are generally step transfer function, linear transfer function and sigmoid transfer function. With non-linear problem, the sigmoid transfer function is chosen. The output of a single layer neural network is described in (1)

$$O = f(Wp + b) \quad (1)$$

Where O is the output from neural network (result of neural network approximation).

f is the transfer function stated above.

W is the weight matrix.

p is the input vector (feature vector).

b is the bias vector.

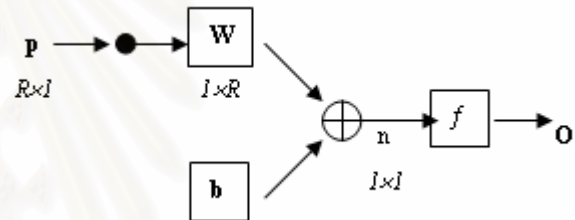


Fig. 3: Structure of a simple neural network

Before operating the neural network in a specific problem, the neural network must be trained. In the training phase, the measured features (inputs) are fed into the network whose weights have already been initialized. Then the network produces the output which is compared to the real target (t) to find error (e). Then the error is used to update the weights and bias in the next epoch (iteration).

The updating components to update the weights and bias are described in (2) and (3).

$$\Delta W = (t - O)(p)^T = e(p)^T \quad (2)$$

$$\Delta b = (t - O) = e \quad (3)$$

Where t is the target matrix (real output).

New weights and bias are as follows:

$$W^{new} = W^{old} + \Delta W \quad (4)$$

$$b^{new} = b^{old} + \Delta b \quad (5)$$

The algorithm for training the simple neural network described above is called *supervised learning*. Supervised

learning is used in the application such as recognition and approximation. If there is no target supplied to neural network, it is called *unsupervised learning* which is used in the classification application.

However, a simple neural network does not suit for some applications. In this application, a series of simple neural network is used which is called *feed-forward neural network* or *multi-layer perceptron*. This type of network performs well in many applications, also in position location, as will be seen in this paper. The next section will explain about the simulations and the results of the method proposed in this paper.

3. SIMULATION OF USING NEURAL NETWORK IN LOCATING MS POSITION

3.1 Neural network structure for locating MS position

Figure 4 shows the structure of BSs considered in this paper. There are three BSs locating at triangular position and the MSs are uniformly distributed in the triangle area with each BS being a corner of the triangle. Each BS has an antenna which is 200 meters high and MS is 2 meters high from the ground. The considered system is CDMA system with perfect time synchronization between BS and MS. In the training phase, the training data is simulated from the empirical formula for signal strength information considering that the system is in a small/medium sized city (suburban area) and the signal strength is affected by shadowing. The one-way propagation time delay is also simulated. Both of the data are mapped to be training sequence input and the exact location simulated is the training target. All the training sequence inputs and the training targets are fed into neural network of the multi-layer perceptron with back-propagation training type used in this application. The neural network has three layers consisting of one input layer, one hidden layer and one output layer. The hidden layer consists of 20 neurons as shown in Figure 5.

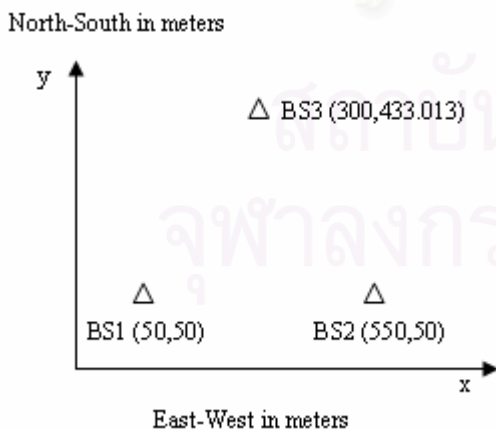


Fig. 4: Structure of BSs

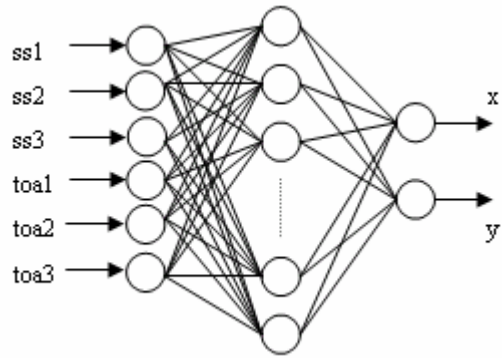


Fig. 5: Structure of neural network for locating MS position

The simulated data file for training contains more than 250 values of information. In addition, the simulated data file for testing contains more than 100 values.

The result of training converges into a specific error threshold. With a specific error (which is 30 meters in this simulation), the result of the neural network described above can converge to lower than 30 meters which seems to be satisfied, compared with the FCC requirement of 125 meters.

3.2 Result and source of error

In testing phase, a new set of data is simulated and fed into neural network. The error calculated from the output of the neural network is shown in Figure 6.

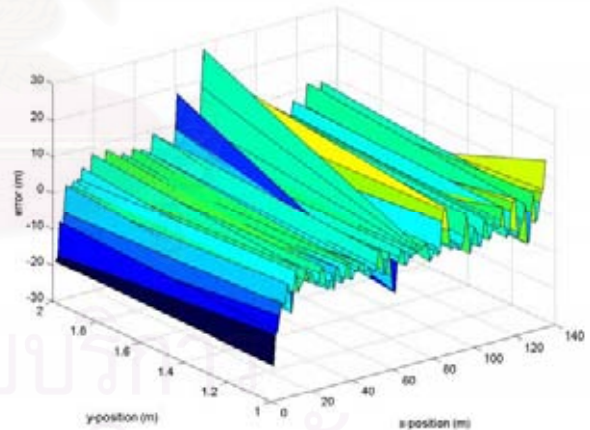


Fig. 6: Result of testing the neural network

From Figure 6, the maximum error of MS position is nearly 30 meters which seems to be a very satisfactory result.

Figure 7 shows the error distribution of the testing result. The result shows that the error obtained from the trained network is below the error threshold of 30 meters.

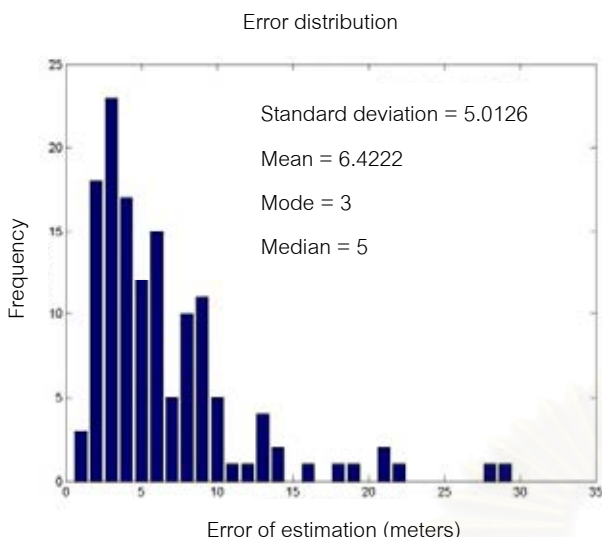


Fig. 7: Error distribution in the testing phase

Though, using neural network in locating the position of the MS is one of the good methods, some subjects must be considered as the sources of error:

- The reliability of measured information. Ambiguous and unreliable data (obtained from the drive test in practice) can make the neural network not perform well and this leads to error in estimating the location.
- The hearability problem. If three BSs can not hear signal from the MS, then a poor performance of neural network is obtained.
- The configuration of BS. This paper considers only the configuration of the BSs shown in Figure 4. If BSs locate in the other ways, the neural network must be trained to cover all types of configurations of the BSs or else it will lead to poor performance.

4. SUMMARY

This paper presents an application of neural network in positioning systems. This algorithm has been tested using simulations in a suburban environment. Clearly, neural network can be used to improve the estimation of the MS position. The simulation results show that the approximation of the position is compatible with the FCC requirements, and the advantages of using a neural network are high performance and low complexity of the embedded algorithm. However, using neural network with high accuracy and low complexity in MS location and testing its performance when applied to the real data from a drive test are still under study.

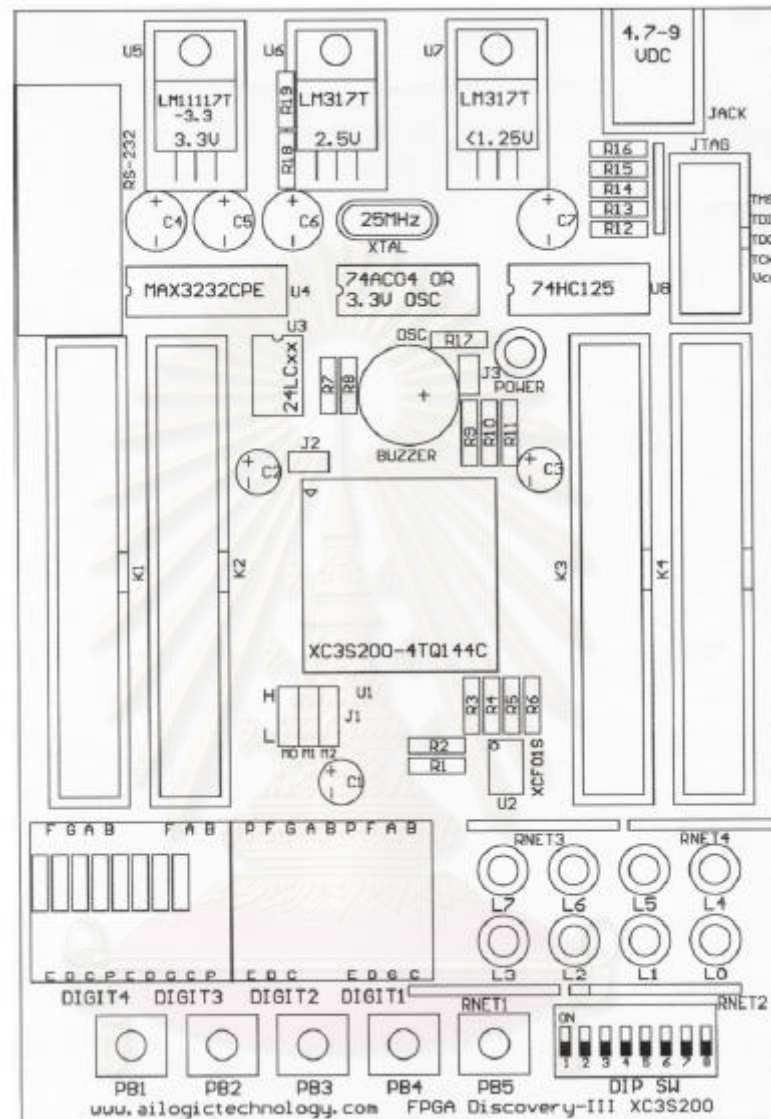
5. ACKNOWLEDGEMENT

The authors wish to thank the Cooperative Project of Research and Development between Department of Electrical Engineering, Chulalongkorn University and Private Sector for supporting this work.

6. REFERENCES

- [1] Wang S., Green M., and Malkawi M. "E-911 Location Standards and Location Commercial Services". *IEEE Emerging Technologies Symposium: Broadband, Wireless Internet Access*, 2000, pp. 1-5.
- [2] Caffery J., Stuber Jr., and Stuber G. "Overview of Radiolocation in CDMA Cellular Systems". *IEEE Communication Magazine*. vol. 36, Apr. 1998, pp. 38-45.
- [3] Rappaport T., Reed J., and Woerner B. "Position Location Using Wireless Communications on Highways of the Future". *IEEE Communication Magazine*. vol. 34, Oct. 1996, pp. 33-41.
- [4] Masato A., Manabu K., and Takeshi H. "A New Location Estimation Method Based on Maximum Likelihood Function in Cellular Systems". *IEEE Vehic. Tech. Conf.*, 2001, pp. 106-110.
- [5] Robert J. Schalkoff. *Pattern Recognition: Statistical and Neural Approaches*. John Wiley & Sons, Clemson, 1992.
- [6] Demuth H. and Beale M. *Neural Network Toolbox for Use with MATLAB*. The MATH Works, 1998.

FPGA Discovery-III XC3S200 Board Manual



1. Connector and Jumper

1.1 Expansion connector (K1 – K4)

เป็นหัวต่อที่ใช้เชื่อมต่อสัญญาณ I/O จาก FPGA ไปยังบอร์ดหรืออุปกรณ์ภายนอกที่มี I/O เป็น 3.3 V โดยจะต่ออยู่กับขา CPLD ดังตารางด้านล่าง ในกรณีที่ I/O ของ FPGA เป็น Output สามารถต่อออกจากบอร์ดไปขับ Input ของอุปกรณ์ที่เป็นระบบ 3.3 V และ 5 V ได้โดยตรง แต่ถ้า I/O ของ FPGA เป็น Input นั้นจะรับได้เฉพาะ Input ที่เป็นระบบ 3.3 V เท่านั้น (ถ้ารับมาจากระบบ 2.5 V ต้องใช้ตัวความต้านทาน (R) มา Pull up) แต่ถ้า Input เป็นระบบ 5 V จะต้องใช้บัฟเฟอร์ที่เป็นระบบ 3.3 V มากั้นเพื่อป้องกันไม่ให้ I/O ของ

FPGA ได้รับความเสียหาย บัฟเฟอร์ที่เป็นระบบ 3.3 V อาจใช้ไอซีตระกูล 74HCxx หรือ 74ACxx (เร็วกว่า) มากินไฟเลี้ยง (Vcc) 3.3 VDC และต้องต่อความต้านทาน 200 โอห์ม ที่ Input ของบัฟเฟอร์ทุกตัว (เพื่อจำกัดกระแสไม่ให้ Input ของบัฟเฟอร์เสียหาย คือ < 10 mA)

1.2 JTAG connector

เป็นหัวต่อที่ใช้ต่อกับสายที่ใช้โปรแกรมข้อมูลลงตัว FPGA และ PROM โดยผ่านทาง JTAG Cable

1.3 J1

จัมเปอร์ J1 ประกอบด้วย M0, M1, M2 โดยปกติให้เซทไว้ที่ลอจิก "L" หรืออยู่ในโหมด Master serial เนื่องจากเราสามารถโปรแกรม PROM หรือ FPGA โดยใช้สาย JTAG ได้อยู่แล้ว โดยไม่ต้องสนใจตำแหน่งของจัมเปอร์แต่อย่างใด

1.4 J2

จัมเปอร์ที่ใช้ควบคุมให้ FPGA ทำการ Pull up I/O ของ FPGA ทุกขาเมื่อใส่จัมเปอร์ และจะเป็น Hi Impedance เมื่อถอดจัมเปอร์ออก

1.5 J3

จัมเปอร์ที่ใช้ตัด BUZZER ออกจาก FPGA เมื่อถอดจัมเปอร์ออก

2. Input

2.1 DIP switch (DIP SW)

เป็นชุดของสวิตช์เลื่อนขนาดเล็กที่ใช้ป้อนข้อมูลเข้าสู่ FPGA โดยถ้าเลื่อนลง (Off) จะเป็น "1"  ถ้าเลื่อนขึ้น (On) จะเป็น "0"  โดยเชื่อมต่ออยู่กับขา FPGA ดังตาราง ด้านท้าย DIP switch ทุกตัวจึงทำงานแบบ Active Low

2.2 Push button switch (PB1 – PB5)

เป็นสวิตช์กดติดปลั๊กดัดที่ให้สัญญาณ Output เป็นระดับลอจิก "0" เมื่อกดสวิตช์ และเป็นระดับลอจิก "1" เมื่อปล่อยสวิตช์ โดยเชื่อมต่ออยู่กับขา FPGA ดังตาราง ด้านท้าย Push button switch ทุกตัวจึงทำงานแบบ Active Low

2.3 Changeable oscillator (OSC)

เป็นตัวกำเนิดสัญญาณนาฬิกาที่สามารถเปลี่ยนค่าความถี่ที่ต้องการได้ โดยการถอดเปลี่ยนออสซิลเลเตอร์เดิม (3.3 V) ที่ให้มาบนบอร์ดออก แล้วใส่ตัวใหม่ (3.3 V) เข้าไปแทนที่ที่ Socket ไอซีเบอร์ 74AC04 โดยที่ Output ของ OSC จะต่ออยู่กับขา FPGA ดัง

ตารางด้านท้าย ซึ่งเป็นขา Global clock เหมาะสำหรับวงจรที่ต้องการความถี่ในการทำงานสูง ๆ

3. Output

3.1 7-Segment (DIGIT1 – DIGIT4)

เป็นตัวแสดงผลเจ็ดส่วนจำนวน 4 หลักที่สามารถถอดออกได้ (หากต้องการใช้ I/O ที่ Connector K1 และ K2 ส่วนที่แชร์ I/O อยู่กับตัวแสดงผลเจ็ดส่วนทั้ง 4 หลัก) โดยเรียงจากซ้ายไปขวาคือ DIGIT4, DIGIT3, DIGIT2 และ DIGIT1 โดยที่ตัวที่ 2 และ 1 จะทำการกลับตัวแสดงผลเจ็ดส่วนให้เพื่อใช้จุด (Dot) ในการทำนาฬิกาหรือแสดงองศาในการวัดอุณหภูมิ เช่น 11:39 หรือ 20°

ตัวแสดงผลเจ็ดส่วนทั้งหมดจะต่อขา Data เข้าด้วยกัน โดยมีขาไฟร่วม (Common cathode) แยกกันสี่ขา ดังนั้นผู้ใช้จึงจำเป็นต้องใช้เทคนิคในการสแกน (Scan) เพื่อให้ตัวแสดงผลเจ็ดส่วนสามารถแสดงผลพร้อมกันได้ทั้งหมด และตัวแสดงผลทั้งหมดเป็นแบบไฟร่วม (Common cathode) โดยจะต่ออยู่กับขา FPGA ดังตารางด้านท้าย

3.2 LED แสดงผล

LED แสดงผล L0 – L7 จะต่อแชร์กับ I/O ของ Connector K3 และ K4 โดยที่ L2, L3, L6 และ L7 จะต่อแชร์กับ I/O ของ Connector K3 โดยมี Register “RNET3” 8P4R 470 Ohm จำกัดกระแส และ L0, L1, L4 และ L5 จะต่อแชร์กับ I/O ของ Connector K4 โดยมี Register “RNET3” 8P4R 470 Ohm จำกัดกระแส

3.3 Buzzer

เป็นฮอดความถี่เสียง (Buzzer) โดยที่จะมีเสียงดังเมื่อบิตสัญญาณเป็น High “1” โดยจะต่ออยู่กับขา FPGA ดังตารางด้านท้าย กรณีที่ต้องการใช้ I/O ของ Connector K4 ที่แชร์อยู่กับฮอดหรือไม่ต้องการใช้ฮอดให้ถอดจัมเปอร์ J3 ออก

4. Misc

4.1 Jack สำหรับ DC Adaptor

เป็นหัวต่อไฟเลี้ยงเพื่อป้อนให้แก้อบอร์ดในการทำงาน ต่ออยู่กับอะแดปเตอร์ที่มีไฟออกมาเป็น 4.7 V – 9 V โดยมีขั้วด้านในเป็นบวก “+” ด้านนอกเป็นลบ “-”

4.2 Power LED (POWER)

เป็นไดโอดเปล่งแสงว่าในขณะนั้น ๆ มีไฟเลี้ยงบอร์ดอยู่หรือไม่

4.3 RS-232C Port

เป็นพอร์ต RS-232C ซึ่งหากไม่ต้องการใช้พอร์ต RS-232C แต่ต้องการใช้เป็น I/O ที่ Connector K1 และ K2 (ส่วนที่แชร์ I/O อยู่กับพอร์ต RS-232C) ให้ถอดไอซี MAX3232CPE ออกจาก Socket

หมายเหตุ ขา 15 ของไอซี MAX3232CPE ต้องบัดกรีลงกราวนด์ด้วย

4.4 Platform Flash PROM เบอร์ XCF01S

เป็น Serial PROM ที่สามารถโปรแกรมได้โดยตรงผ่านสาย JTAG สามารถโปรแกรมซ้ำได้ประมาณ 20,000 ครั้ง

4.5 12C Socket

เป็น Socket สำหรับใส่ไอซีแบบ 12C Serial EEPROM เบอร์ 24LCxx เช่น 24LC256 เป็นต้น ซึ่งสามารถถอดออกได้ (หากต้องการใช้ I/O ที่ Connector K1 และ K2 ส่วนที่แชร์ I/O อยู่กับ 12C Serial EEPROM) I/O 2 ขานี้จะมี Pull up resistor 4.7 kOhm ต่ออยู่

5. ตาราง I/O ของ FPGA

ดูจากไฟล์ K1, K2, K3, K4 และ 7-Segment การดูขาของ Connector K1 – K4 ให้สังเกตด้านรอยบากลึกรูปสี่เหลี่ยมจะเป็นแถวของขาที่เป็นตัวเลขคี่ทั้งหมดเริ่มจาก 1 ถึง 39 โดยที่ด้านซ้ายสุดเป็นขา 1 ซึ่งจะมี Mark รูปสามเหลี่ยมปรากฏอยู่ด้านหลัง

7-Segment	FPGA Pinout	Descriptions
a	p40	a
b	p35	b
c	p32	c
d	p30	d
e	p27	e
f	p25	f
g	p23	g
dp	p20	Decimal Point
DG1	p31	DIGIT1, COMMON CATHODE
DG2	p33	DIGIT2, COMMON CATHODE
DG3	p36	DIGIT3, COMMON CATHODE
DG4	p41	DIGIT4, COMMON CATHODE

Push Button	FPGA Pinout	Descriptions
PB1	p44	Push Button No. 1
PB2	p46	Push Button No. 2
PB3	p47	Push Button No. 3
PB4	p50	Push Button No. 4
PB5	p51	Push Button No. 5

EEPROM	FPGA Pinout	Descriptions
12C-SCL	p128	24LCxx
12C-SDA	p129	24LCxx

RS-232	FPGA Pinout	Descriptions
TX	p131	MAX3232CPE
RX	p132	MAX3232CPE

LED	FPGA Pinout	Descriptions
L0	p70	L0
L1	p77	L1
L2	p69	L2
L3	p76	L3
L4	p74	L4
L5	p79	L5
L6	p73	L6
L7	p78	L7

Dip SW	FPGA Pinout	Descriptions
1	p52	Dip Switch No. 1
2	p53	Dip Switch No. 2
3	p55	Dip Switch No. 3
4	p56	Dip Switch No. 4
5	p59	Dip Switch No. 5
6	p60	Dip Switch No. 6
7	p63	Dip Switch No. 7
8	p68	Dip Switch No. 8

Oscillator	FPGA Pinout	Descriptions
OSC	p127	25MHz, GCLK6

BUZZER	FPGA Pinout	Descriptions
BUZZER	p124	BUZZER

K1 Pinout	FPGA Pinout	Descriptions
1	p40	I/O, a
2		3.3 V
3	p35	I/O, b
4		GND
5	p32	I/O, c
6		GND
7	p30	I/O, d
8		GND
9	p27	I/O, e
10		GND
11	p25	I/O, f
12		GND
13	p23	I/O, g
14		GND
15	p20	I/O, dp
16		GND
17	p17	I/O
18		GND
19	p14	I/O
20		GND

K1 Pinout	FPGA Pinout	Descriptions
21	p12	I/O
22		GND
23	p10	I/O
24		GND
25	p7	I/O
26		GND
27	p5	I/O
28		GND
29	p2	I/O
30		GND
31	p141	I/O
32		GND
33	p137	I/O
34		GND
35	p132	I/O, RS-232 (RX)
36		GND
37	p130	I/O
38		GND
39	p128	I/O, GCLK7, 12C-SCL
40		GND

K2 Pinout	FPGA Pinout	Descriptions
1	p41	I/O, DIGIT4
2		3.3 V
3	p36	I/O, DIGIT3
4		GND
5	p33	I/O, DIGIT2
6		GND
7	p31	I/O, DIGIT1
8		GND
9	p28	I/O
10		GND
11	p26	I/O
12		GND
13	p24	I/O
14		GND
15	p21	I/O
16		GND
17	p18	I/O
18		GND
19	p15	I/O
20		GND

K2 Pinout	FPGA Pinout	Descriptions
21	p13	I/O
22		GND
23	p11	I/O
24		GND
25	p8	I/O
26		GND
27	p6	I/O
28		GND
29	p4	I/O
30		GND
31	p1	I/O
32		GND
33	p140	I/O
34		GND
35	p135	I/O
36		GND
37	p131	I/O, RS-232 (TX)
38		GND
39	p129	I/O, 12C-SDA
40		GND

ประวัติผู้เขียนวิทยานิพนธ์

นายศรัณย์ เกียรติศิริโรจน์ เกิดเมื่อวันที่ 21 สิงหาคม 2523 สำเร็จการศึกษาปริญญา
วิศวกรรมศาสตรบัณฑิต ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์
ในปีการศึกษา 2544 จากนั้นได้เข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชา
วิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อ พ.ศ.
2545



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย