



การควบคุมระบบไฟฟ้ากำลังโดยใช้โครงข่ายนิเวศ Power System Controlled by Neural Network



โดย

นายวิศ อิงคณิสร

นายฤทธิ แสงเมล์

นายวฤทธิ์ อาทิตย์กุล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมไฟฟ้า

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2535

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

032636

ปริญญาานิพนธ์ปีการศึกษา 2535

ภาควิชาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การควบคุมระบบไฟฟ้ากำลังโดยใช้โครงตาข่ายนิรอรล

ผู้จัดทำ

- | | | | |
|----|-----------|------------|---------|
| 1. | นายวิศ | อิงคณิศร | 32.1255 |
| 2. | นายฤทธิ์ | แสงเมล์ | 32.1264 |
| 3. | นายวฤทธิ์ | อาทิตย์กุล | 32.1285 |



อาจารย์ที่ปรึกษา

(ผศ. มณฑล สีสัจจินดาไกรฤกษ์)

การควบคุมระบบไฟฟ้ากำลัง โดยใช้โครงตาข่ายนิเวรอล

รวีศ อิงคณิศร

ฤทธิ แสงเมล์

วฤทธิ อาทิตย์กุล

ผศ.มลทล ลีลาจินดาไกรฤกษ์ อาจารย์ที่ปรึกษา

ปีการศึกษา 2535

บทคัดย่อ

รายงานฉบับนี้ ได้แสดงถึงการควบคุมสภาวะเสถียรของระบบไฟฟ้ากำลัง โดยใช้ อุปกรณ์ควบคุมความเร็วรอบ (GOV) อุปกรณ์ควบคุมแรงดัน (AVR) และตัวควบคุมเสถียรภาพด้วยกฎพื้นฐาน (Rule-Based Stabilizer) อุปกรณ์ควบคุมความเร็วรอบใช้ควบคุมความถี่ให้คงที่ โดยการปรับวาล์วที่เป็นอินพุทของใบพัด อุปกรณ์ควบคุมความเร็วรอบใช้ควบคุมแรงดันให้มีค่าคงที่ โดยการควบคุมแหล่งจ่ายไฟฟ้ากระแสตรงที่สนามแม่เหล็กกระตุ้น ตัวควบคุมเสถียรภาพด้วยกฎพื้นฐานเป็นตัวควบคุมที่มีการคำนวณบนเครื่องคณิตกรณ์ (Computer) เนื่องจากมีการคำนวณซับซ้อน เงื่อนไขของกฎพื้นฐานจะใช้เงื่อนไขของสภาวะของเครื่องกำเนิดไฟที่เวลาสุ่มใดๆ (Sampling Time) แล้วนำไปวิเคราะห์บนคณิตกรณ์ แล้วจะส่งสัญญาณกลับเข้าไปในลูปควบคุมกระตุ้น (Excitation Control Loop) ที่เวลาสุ่มนั้นๆ และมีการนำเอาทฤษฎีการไหลของภาวะมาประยุกต์ใช้เพื่อสามารถควบคุมเครื่องจักรหลาย ๆ เครื่องได้

การนำเอาโครงตาข่ายนิเวรอล (Artificial Neural Network) มาประยุกต์ใช้ในการควบคุมระบบไฟฟ้ากำลัง ซึ่งจะต้องมีการฝึก (Train) ให้ ANN มีการเรียนรู้ถึงการทำงานของตัวควบคุมเสถียรภาพด้วยกฎพื้นฐาน ซึ่งจะทำให้ ANN สามารถเลียนแบบการทำงานของตัวควบคุมเสถียรภาพด้วยกฎพื้นฐาน และสามารถนำไปใช้ควบคุมระบบไฟฟ้ากำลังได้

Power System Controlled by Neural Network

Ravis Ingkanisorn

Ritte Saengmay

Warit Attitkul

Asst.Prof.Monthol Leelachindakraierk

Advisor

1992

ABSTRACT

The paper presents power system stabilizing control by using GOV, AVR, and Rule-based Stabilizer. Adjusting valve which is turbine's input, GOV is controlled for constant frequency. Adjusting DC. generator at excited field winding, AVR is controlled for constant voltage. Rule-Based stabilizing can be constructed by using a microcomputer for hard calculating. Rule-based Stabilizing is determined generating condition which is measured at every sampling time and analysis on computer. Computer supply stabilizing signal which is added to the excitation control loop at sampling time and load flow method is applied to control multi-machine.

Using artificial neural network to control power system stability . It must be trained for learning rule-base stabilizer algorithm. ANN has capability to immitate rule base stabilizer procedure and can control power system stability.

สารบัญ

บทที่	หน้า
1. บทนำ	1
2. ตัวเสถียรภาพของเครื่องจักรไฟฟ้าในระบบไฟฟ้ากำลัง	3
2.1 อุปกรณ์ควบคุมแรงดันอัตโนมัติ	3
2.1.1 เครื่องควบคุมแรงดันอัตโนมัติ	4
2.1.2 ชนิดของอุปกรณ์ควบคุมแรงดัน	6
2.2 อุปกรณ์ควบคุมความเร็ว	7
2.2.1 หลักการทำงานทั่วไปของอุปกรณ์ควบคุมความเร็ว	8
2.2.2 ขั้นตอนการทำงานของอุปกรณ์ควบคุมความเร็ว	9
2.2.3 กลไกในการควบคุมความเร็วรอบ	10
2.3 การควบคุมเสถียรภาพโดยกฎพื้นฐาน	11
3. การไหลของภาระ (Load flow Methods)	15
4. โครงตาข่ายนิรอรล	22
4.1 ความรู้เบื้องต้น	22
4.2 โครงตาข่ายนิรอรลและตัวจัดกลุ่ม	25
4.3 หลักการเบื้องต้นของทฤษฎีโครงตาข่ายนิรอรล	26
4.3.1 ต้นแบบทางชีววิทยา	26
4.3.2 นิรอรนเทียม	27
4.3.3 ฟังก์ชันกระตุ้น	28
4.3.4 โครงตาข่ายนิรอรลแบบเลเยอร์เดียว	31
4.3.5 โครงตาข่ายนิรอรลแบบหลายชั้น	32
4.3.6 ฟังก์ชันการกระตุ้นที่ไม่เป็นเชิงเส้น	33
4.3.7 โครงตาข่ายแบบรีเคอร์รนท์	33
4.3.8 ลักษณะการใช้ถ้อยคำ	34
4.3.9 การแสดงตัวอย่าง	34
4.3.10 การฝึกของโครงตาข่ายนิรอรล	35
4.3.11 การฝึกแบบตรวจสอบ	35
4.3.12 การฝึกแบบไม่มีการตรวจสอบ	35
4.3.13 ขั้นตอนการทำงานของกรฝึก	36

บทที่	หน้า	
4.4	เปอร์เซ็ปตรอน	36
4.4.1	เปอร์เซ็ปตรอนและจุดเริ่มต้นของ ANN	37
4.4.2	การนำเสนอเปอร์เซ็ปตรอน	39
4.4.3	การเรียนรู้เปอร์เซ็ปตรอน	39
4.4.4	ขั้นตอนการฝึกเปอร์เซ็ปตรอน	40
4.4.5	ซิงเกิลเลเยอร์เปอร์เซ็ปตรอน	42
4.4.6	มัลติเลเยอร์เปอร์เซ็ปตรอน	45
4.4.7	กฎของเซลล์ด้า	48
4.5	การกระจายแบบย้อนกลับ	49
4.5.1	ขั้นตอนการฝึกการกระจายแบบย้อนกลับ	50
4.5.2	โครงตาข่ายชั้นซับซ้อน	52
4.5.3	ภาพโดยรวมของการฝึก	53
4.5.4	ขบวนการผ่านไปข้างหน้า	54
4.5.5	ขบวนการผ่านย้อนกลับ	55
4.5.6	โมเมนต์ัม	57
4.5.7	ขั้นตอนการทำงานขั้นสูงสุด	58
4.5.8	การประยุกต์ใช้งาน	59
5	ขั้นตอนและผลการวิเคราะห์	60
6	สรุปผลและข้อแนะนำ	70
ภาคผนวก 1		72
ภาคผนวก 2		76
กิตติกรรมประกาศ		98
บรรณานุกรม		99

สารบัญรูป

รูปที่		หน้า
2-1	การควบคุมเครื่องกำเนิดไฟฟ้า	3
2-2	แบบจำลองของอุปกรณ์ควบคุมแรงดัน	5
2-3	บล็อกไดอะแกรมของอุปกรณ์ควบคุมแรงดัน	6
2-4	ผลของการควบคุมขดลวด	6
2-5	แสดงทรานสเฟอ์ฟังก์ชันของอุปกรณ์ควบคุมความเร็วรอบ	9
2-6	บล็อกไดอะแกรมของชุดเครื่องกำเนิดไฟฟ้าเทอร์โบ	9
2-7	ระบบที่ศึกษา	12
2-8	6 ชิกในระนาบของมุม	12
2-9	Gain factor $g(k)$ กับ distance $D(k)$	13
3-1	แสดงระบบไฟฟ้าชนิดที่มี 3 บัส	16
4-1	แสดงโนดและแสดงความเป็นเชิงเส้นไว้ 3 ลักษณะ	22
4-2	บล็อกไดอะแกรมของตัวจัดกลุ่ม	25
4-3	แสดงลักษณะของนิวรอลในทางชีววิทยา	26
4-4	นิวรอลที่สร้างขึ้น	27
4-5	นิวรอลที่สร้างขึ้นกับฟังก์ชันกระตุ้น	28
4-6	ฟังก์ชันแบบโลจิสติกซิกมอยด์	30
4-7	ฟังก์ชันไฮเปอร์โบลิคแทนเจนท์	31
4-8	โครงตาข่ายนิวรอลชนิดชั้นเดียว	31
4-9	โครงตาข่ายนิวรอลชนิดสองชั้น	32
4-10	แสดงนิวรอลเปอร์เซ็ปตรอน	37
4-11	แสดงเปอร์เซ็ปตรอนที่มีเอาต์พุทหลายค่า	38
4-12	การทราบภาพพจน์เปอร์เซ็ปตรอน	40
4-13	ซิงเกิลเลเยอร์เปอร์เซ็ปตรอน	42
4-14	กล่องการทำงานของเปอร์เซ็ปตรอน	43
4-15	ตัวอย่างของเขตการตัดสินใจ	44
4-16	ชนิดของบริเวณการตัดสินใจ	45
4-17	เปอร์เซ็ปตรอนแบบ 3 ชั้น	46
4-18	นิวรอนที่ถูกสร้างขึ้นกับฟังก์ชันกระตุ้น	50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่	หน้า	
4-19	ฟังก์ชันซิกมอยด์กับไบอัสและการดัดแปลงรูปต่าง ๆ	51
4-20	โครงตาข่ายการกระจายแบบย้อนกลับแบบ 2 ชั้น	52
4-21	ภาพจินตนาการการเรียนรู้	53
4-22	การฝึกน้ำหนักในชั้นเอาต์พุท	55
4-23	การฝึกของน้ำหนักในชั้นภายใน	57
5-1	ระบบที่ 1 ซึ่งมีการควบคุมด้วยอุปกรณ์ควบคุมอัตโนมัติ	60
5-2	ระบบที่ 2 ซึ่งมีการควบคุมด้วยอุปกรณ์ควบคุมอัตโนมัติ	61
5-3	ระบบที่ 1 ซึ่งมีการควบคุมโดยตัวควบคุมเสถียรภาพ	62
5-4	ระบบที่ 2 ซึ่งมีการควบคุมโดยตัวควบคุมเสถียรภาพ	62
5-5	ระบบจำลองที่ใช้ในการวิเคราะห์	62
5-6	ขั้นตอนการทำงานของโปรแกรมควบคุมการไหลของไหล	63
5-7	ผลการควบคุมเครื่องกำเนิดไฟฟ้า	64
5-8	ค่าคงที่ที่ใช้ในระบบ	65
5-9	ขั้นตอนการปรับน้ำหนักของทฤษฎีโครงตาข่ายนิรอร	66
5-10	ขั้นตอนการฝึกของทฤษฎีโครงตาข่ายนิรอร	67
5-11	ผลการนำเอา ANN มาใช้ควบคุมเครื่องกำเนิดไฟฟ้า	68
5-12	ผลการนำเอา ANN ไปใช้ควบคุมระบบจำลอง	68

บทที่ 1

บทนำ

เนื่องมาจากระบบไฟฟ้ากำลังมีขนาดที่เพิ่มขึ้น และมีความซับซ้อนมากขึ้น มีผลข้างเคียงเพิ่มขึ้นในระบบเสถียรของระบบกำลังขนาดใหญ่ ก่อนหน้านี้ระบบเสถียรกำลังถูกสร้างกับวงจรแอนะล็อก เช่น การชดเชยแบบมูมนำ และการรีเซทตัวกรองสัญญาณ (filter) เครื่องกำเนิดไฟฟ้าในปัจจุบันนี้ ได้ถูกจัดเตรียมกับ สภาวะเสถียรแบบแอนะล็อก เพื่อที่จะปรับปรุงพัฒนาพฤติกรรมทางพลศาสตร์ อย่างไรก็ตาม ความพยายามที่จะพัฒนาระบบควบคุมดิจิทัลบนคณิตกรณ์ เช่น การปรับสภาวะเสถียรของไฟฟ้ากำลัง เพื่อที่จะกำเนิดสัญญาณเสถียรที่น่าพึงพอใจ แล้วไปพัฒนาความเสถียรของระบบกำลังทั้งหมด ยกตัวอย่าง เช่นระบบควบคุมคณิตกรณ์ สามารถสร้างโครงร่างง่าย ๆ เพื่อที่จะได้อุปกรณ์ควบคุมขั้นสูงที่น่าเชื่อถือ

จุดประสงค์ข้อแรกในการทำปริญาณิพนธ์ฉบับนี้ของรายงานฉบับนี้ ก็เพื่อที่จะจำลองประสิทธิภาพของเสถียรภาพที่เปลี่ยนแปลงของระบบ ซึ่งเหมาะสำหรับการควบคุมบนคอมพิวเตอร์ เสถียรภาพที่ตั้งไว้สามารถสร้างได้ง่ายโดยใช้คณิตกรณ์ กับ ฟังก์ชันของ A/D และ D/A เงื่อนไขของเครื่องกำเนิดไฟฟ้า จะถูกแสดงกับปริมาณอนุพันธ์ของความ เร็ว และ อัตราเร่งในระนาบของมุมเฟส (phase plane) โดยระนาบดังกล่าว จะถูกแบ่งเป็น 6 ซีก แต่ละซีกจะมีกฎการ ควบคุมเฉพาะ สัญญาณเสถียรภาพที่จ่ายออกไป จากกฎเหล่านี้กับเงื่อนไขที่เวลาสุ่มใดๆนั้น ถูกกำหนด และประยุกต์เพื่อที่จะใช้รูปควบคุมกระตุ้น (Excitation Control Loop) สัญญาณออก (output) ที่ถูกกำหนดให้เสถียร ยกตัวอย่างเช่น อนุพันธ์ของความเร็วมอเตอร์กำเนิดไฟฟ้า จะถูกวันที่เวลาสุ่มใดๆ เพียงส่วนหนึ่งของข้อมูลที่เพิ่มขึ้นมาจากการวัดใหม่ จะนำมาคำนวณอัตราเร่งของเครื่องกำเนิดไฟฟ้า การคำนวณที่ซับซ้อนซึ่งจะได้จากการคำนวณบนคณิตกรณ์ ไม่เกี่ยวข้องกับระบบเสถียรพื้นฐานที่ต้องการ เพราะว่าเป็นการควบคุมอย่างง่าย

เพื่อที่จะสาธิตประสิทธิภาพของระบบเสถียรที่ต้องการการจำลองทั้งหลาย จะถูกแสดงโดยใช้เครื่องจักร 1 ตัวต่ออยู่กับบัส (One-Machine Infinite-Busbar System) การแสดงระบบที่ดีจะต้องสังเกตตลอดการจำลอง ทั้งหมดโดยการประยุกต์ระบบเสถียรที่ต้องการจะดูว่า เป็นกฎการควบคุมแบบง่าย ๆ ที่เดียว ผลของการ จำลองบ่งชี้ให้เห็นว่า การกำหนดค่าพารามิเตอร์เสถียรภาพไม่ขึ้นอยู่กับจุดปฏิบัติการของเครื่องกำเนิดไฟฟ้า เพื่อสิ่งนั้นหลังจากการปรับพารามิเตอร์เสถียรภาพที่จุดปฏิบัติการที่เจาะจง การปรับปรุง พารามิเตอร์เหล่านี้ไม่จำเป็นยิ่งไปกว่านั้นความสำเร็จของการพัฒนาเสถียรโดยใช้ตัวเสถียรภาพพื้นฐาน (Stabilizer) ตัวเสถียรภาพพื้นฐานจะถูกเปรียบเทียบกับความสำเร็จโดยใช้ตัวเสถียรภาพ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เสถียรภาพพื้นฐานจะถูกเปรียบเทียบกับความสำเร็จโดยการใช้ตัวเสถียรภาพทางระบบกำลังที่ตกลงกันได้ และตัวเสถียรภาพซึ่งปรับด้วยตัวเอง

สำหรับจุดประสงค์ที่สำคัญของปริญญาโทฉบับนี้ก็เพื่อที่จะศึกษาถึงโครงสร้างของทฤษฎีโครงตาข่ายนิรอรรวมทั้งการวิเคราะห์และการเขียนโปรแกรมจำลองทฤษฎีเพื่อนำมาใช้ควบคุมระบบเดิมซึ่งประกอบด้วยเครื่องจักรต่ออยู่กับบัส

ทฤษฎีโครงตาข่ายนิรอรนี้ได้วิเคราะห์ค้นคว้าขึ้นมาใหม่เพื่อนำมาวิเคราะห์เสถียรภาพในสภาวะคงที่ (Steady State Stability) ของเครื่องจักรซินโครนัสซึ่งค่าอินพุตที่ป้อนเข้าไปในโครงตาข่ายก็คือตัวแปรของระบบซึ่งมีบทบาทสำคัญต่อเสถียรภาพของระบบในสภาวะคงที่ดังที่ได้กล่าวมาข้างต้นซึ่งจุดประสงค์ที่ต้องการควบคุมเสถียรภาพในสภาวะคงที่ ก็เพื่อทำการตรวจสอบพฤติกรรมทางไดนามิกเมื่อมีสิ่งรบกวนเข้าไปในระบบ (Small Perturbation) เนื่องจากว่าเครื่องจักรซินโครนัสจะต้องสามารถทำงานได้อย่างเป็นปกติในกรณีที่ภาวะของระบบเกิดการเปลี่ยนแปลงไปเป็นระยะ ๆ จากความรู้ที่ว่าแรงบิดหน่วง (Damping Torque) ของเครื่องจักรซินโครนัสจะได้รับผลกระทบจากปัจจัยหลายอย่าง เช่น ภาวะของเครื่องจักร ค่าคงที่สำหรับการกระตุ้น (Exciter Constant) ตัวปรับเสถียรภาพของระบบ (Power System Stabilizer, PSS) เป็นต้น เพราะฉะนั้นสำหรับแบบจำลองทางคณิตศาสตร์ ที่เราจะนำไปใช้ในการประเมินเสถียรภาพของระบบในสภาวะคงที่จะต้องประกอบด้วยพารามิเตอร์ที่กล่าวมาแล้วข้างต้น

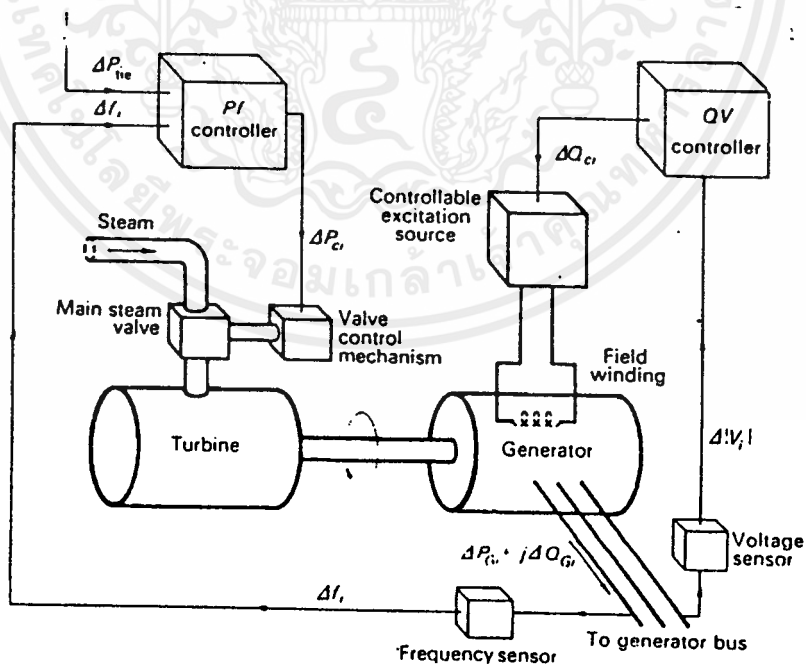
เราได้นำทฤษฎีโครงตาข่ายนิรอรมาประยุกต์ใช้ในการแก้ปัญหาทางระบบไฟฟ้าซึ่งสามารถนำไปใช้แก้ปัญหาที่ซับซ้อนได้ในเวลาอันสั้นและมีความสามารถสูงในการวิเคราะห์เสถียรภาพที่เป็นลักษณะออนไลน์ (ON-line Stability Analysis) เช่นการวิเคราะห์เสถียรภาพในขณะทรานเซียน การประเมินความปลอดภัยทางสเตติกส์ การกำหนดภาระฮาร์โมนิก (Harmonic Load Identification) การควบคุมด้วยตัวเก็บประจุ (Capacitor Control)

บทที่ 2

ตัวเสถียรภาพของเครื่องจักรในระบบไฟฟ้ากำลัง (Stabilizer for Machine in Power System)

2.1 อุปกรณ์ควบคุมแรงดันอัตโนมัติ (Automatic Voltage Regulator)

ในการทำงานของเครื่องกำเนิดไฟฟ้าซิงโครนัส (Synchronous Generator) ในระบบไฟฟ้ากำลังที่มีมุมของภาระสูงๆ (Load Angle) และในสภาวะทรานเซียน (Transient Condition) จะขึ้นอยู่กับแหล่งจ่ายไฟฟ้าที่ทำหน้าที่จ่ายไฟให้กับขดลวดกระตุ้นของเครื่องกำเนิดไฟฟ้า



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 2-1 การควบคุมเครื่องกำเนิดไฟฟ้าให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 เครื่องควบคุมแรงดันไฟฟ้า (Voltage Regulator)

การควบคุม (Regulation) เป็นการควบคุมแรงดันที่จ่ายออกให้จากเครื่องกำเนิดไฟฟ้าที่เทอร์มินอล เช่น ในประเทศอังกฤษกำหนดให้มีการเปลี่ยนแปลงของแรงดันที่เทอร์มินอลได้ $\pm 5\%$ ซึ่งการเปลี่ยนแปลงของแรงดันจะขึ้นอยู่กับระบบและการปฏิบัติการ การเปลี่ยนแปลงของระบบจะมีค่าเพิ่มขึ้นเมื่อมีการเปลี่ยนแปลงอุณหภูมิ ในค่าที่เป็นส่วนประกอบ การเลื่อนในเครื่องขยาย และแรงดันที่จ่ายให้ ในค่าที่ไม่เป็นเชิงเส้นในส่วนประกอบ เช่น ตัวขยายทางแม่เหล็ก หรือ เซอร์โว ซึ่งค่าที่มีการเปลี่ยนแปลงไปนี้จะสามารถกำจัดได้ด้วยการออกแบบที่ดีและการเลือกส่วนประกอบอย่างระมัดระวัง การเปลี่ยนแปลงในการทำงานจะมีค่าที่เพิ่มขึ้นเมื่อมีการเปลี่ยนแปลงในสภาวะของภาระ ซึ่งต้องการแรงดันที่ขดลวดกระตุ้นที่มีค่าแตกต่างกันออกไป

ในเครื่องควบคุมแรงดันไฟฟ้าพื้นฐาน จะมีการทำงานโดยใช้การเปรียบเทียบแรงดันที่มีการปรับเป็นกระแสตรงแล้ว (Rectifier) จากแรงดันที่เทอร์มินอลของเครื่องกำเนิดไฟฟ้า V_f กับแรงดันอ้างอิง V_r ค่าความผิดพลาดจะถูกขยายโดยเครื่องขยายและป้อนเข้าสู่ขดลวดกระตุ้น เครื่องขยายอาจจะประกอบด้วยเครื่องขยายแม่เหล็ก หรือ เครื่องขยายไทรริสเตอร์ หรือ ตัวกระตุ้นกระแสตรงที่ป้อนเข้าสู่กระแสตรงหลัก (D.C. pilot excitor feeding into a main D.C.) หรือตัวกระตุ้นกระแสสลับกับเครื่องปรับ (A.C. excitor with rectifier)

ซึ่งจะมีสมการการกระตุ้นในสภาวะคงที่ดังต่อไปนี้

$$V_f = \mu (V_r - V_f) \quad (2.1)$$

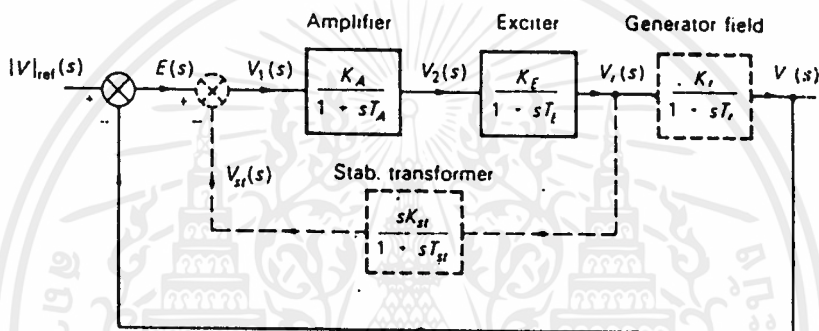
เมื่อ V_f เป็นแรงดันที่จ่ายให้กับขดลวดกระตุ้นที่โรเตอร์ และ μ เป็นตัวขยายทั้งหมดของระบบกระตุ้นซึ่งจะเป็นการง่ายขึ้นถ้าทำให้อยู่ในรูปเปอร์ยูนิต (per-unit)

รูปที่ 2-2 จะแสดงถึงแบบจำลองที่ในระบบนี้ ซึ่งจะมีส่วนประกอบหลัก ๆ ของอุปกรณ์ควบคุมแรงดันดังต่อไปนี้

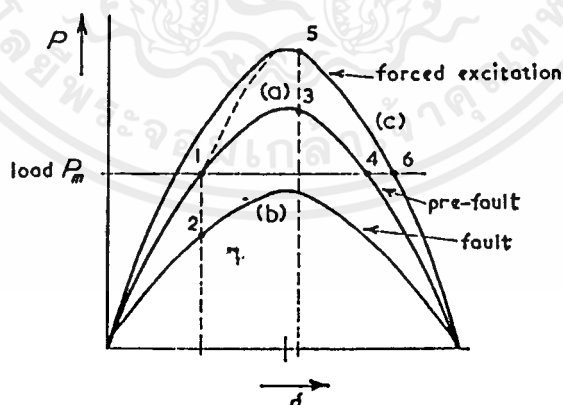
อุปกรณ์เปรียบเทียบแรงดัน (Voltage Comparator) เป็นอุปกรณ์ที่ทำการเปรียบเทียบแรงดันที่เทอร์มินอลของเครื่องกำเนิดไฟฟ้ากับแรงดันอ้างอิงซึ่งมีการตั้งค่าให้เท่ากับแรงดันที่กำหนดไว้ ดังนั้นค่าผิดพลาดของแรงดันจะเท่ากับ

$$\begin{aligned} e &= \Delta |V|_{ref} - |V| \\ &= |V^0| - (|V^0| - \Delta |V|) \end{aligned} \quad (2.2)$$

การทำงานอยู่ที่จุด 1 เมื่อเกิดการผิดปกติดังกราฟบี โรเตอร์จะเกิดการเร่งและไม่มีการควบคุมการกระตุ้นหลังจากที่การเกิดการผิดปกติได้ถูกจัดการไปหมดแล้วการทำงานของเครื่องกำเนิดไฟฟ้าจะกลับเข้าสู่กราฟเอที่จุด 3 โดยคิดว่าไม่มีการเปลี่ยนแปลงในค่ารีแอคแตนซ์ การทำงานในช่วงนี้จะมีการหน่วงเกิดขึ้นซึ่งจะมีการทำงานอยู่ที่จุด 4 ที่จุดนี้โรเตอร์จะเกิดการเร่งอีกครั้งหนึ่งและจะทำให้เกิดการสูญเสียสภาวะซิงโครนิสซึม แต่เมื่อมีการควบคุมการกระตุ้น ซึ่งมีการทำงานที่จุด 5 ของกราฟ C และมีการกำหนดการแกว่งให้อยู่ในจุด B กำลังที่ลดลงจะมีค่ามากและโอกาสในการสูญเสียสภาวะซิงโครนิสซึมมีค่าลดลง



รูปที่ 2-3 บล็อกไดอแกรมของอุปกรณ์ควบคุมแรงดัน



รูปที่ 2-4 ผลของการควบคุมขดลวดกระตุ้น

2.1.2 ชนิดของอุปกรณ์ควบคุมแรงดัน

ในส่วนที่แล้วจะแสดงให้เห็นถึงการออกแบบอุปกรณ์ควบคุมแรงดันซึ่งได้รับอิทธิพลจากความต้องการในทางปฏิบัติแต่ก็ยังมีสิ่งที่ยึดแย้งกันอยู่เช่น การหน่วงเวลาอาจจะไม่เหมาะสมเป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่บนสื่อออนไลน์
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประโยชน์ในแง่เสถียรภาพในสภาวะคงที่แต่ จะขัดกับสภาวะทรานเซียน ในขณะที่ค่าอัตราขยายที่สูงจะดีสำหรับการควบคุมเสถียรภาพในสภาวะคงที่ การพัฒนาของอุปกรณ์ควบคุมในหลายปีที่ผ่านมาจึงมุ่งหมายเพื่อลดสิ่งที่ขัดแย้งกันเหล่านี้

การพัฒนาระบบกระตุ้นสามารถแบ่งได้ดังนี้

1. ตัวกระตุ้นกระแสตรงที่ควบคุมด้วยมือ (DC Exciter with Manual Field Control)
2. ตัวกระตุ้นกระแสตรงที่ควบคุมด้วยกลไกทางไฟฟ้า (DC Exciter with Electromechanical Reg.)
3. ตัวกระตุ้นกระแสตรงที่ควบคุมด้วยแอมป์ไดนา (Amplidyne) (DC Exciter with Amplidyne Control)
4. ตัวกระตุ้นกระแสตรงที่ควบคุมด้วยตัวขยายแม่เหล็ก (Magnetic Amplifier Control)
5. ตัวกระตุ้นกระแสสลับ การควบคุมด้วยตัวขยายแม่เหล็ก และจัดเรียงกระแสด้วยไอโอดแบบสแตติก
6. ตัวกระตุ้นกระแสสลับที่มีการควบคุมขดลวดกระตุ้นด้วยไทรสเตอร์ และจัดเรียงกระแสด้วยไดโอดทางสถิตย์
7. ตัวกระตุ้นกระแสสลับที่มีการควบคุมขดลวดกระตุ้นด้วยไทรสเตอร์และจัดเรียงกระแสด้วยโรเตตติ้งไอโอด (Rotating Diode)
8. ตัวกระตุ้นกระแสสลับที่มีโรเตตติ้งไทรสเตอร์ (Rotating Thyristor)

การควบคุมสนามกระตุ้นด้วยมือแทนที่ด้วยตัวควบคุมระดับแรงดันด้วยกลไกทางไฟฟ้า แต่ไม่สามารถปรับปรุงเสถียรภาพทางไดนามิกส์ได้ด้วยการควบคุมแรงดันเพียงอย่างเดียว อุปกรณ์ควบคุมด้วยแอมพลิดายน์และตัวขยายแม่เหล็กไฟฟ้าสำหรับตัวกระตุ้นกระแสตรงได้ถูกพัฒนาขึ้นในช่วงเดียวกัน แต่ก็มีปัญหาทางด้านเวลาที่ช้าและความต้องการรูปที่มีเสถียรภาพ

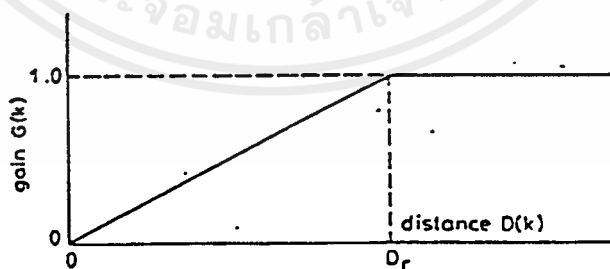
2.2 อุปกรณ์ควบคุมความเร็ว

ในอดีตตัวควบคุม (Governor) จะใช้ในการรักษาเสถียรภาพในสภาวะคงที่ของระบบ ไฟฟ้ากำลังภายใต้การเปลี่ยนแปลงของภาระ และใช้ป้องกันการดำเนินงานที่ความเร็วรอบสูงเกิน (Over Speed) จนทำให้เกิดการทรูปอย่างฉุกเฉิน (Emergency Trip) และทำให้เกิดการไม่ยอมรับภาระทางไฟฟ้า (Load Rejection)

จากกฎจะสามารถอธิบายได้ดังต่อไปนี้

- ก. ควบคุมด้วยความหน่วงอย่างรุนแรงจากจุด a ชิก A เพื่อที่จะป้องกันไม่ให้อุปกรณ์ของความเร็วจนเคลื่อนที่ไปทางขวามากไปกว่านี้
- ข. ควบคุมด้วยความหน่วงที่มีขนาดเล็ก จากจุด b ในชิก B เพื่อที่จะเลื่อนความเร็วของเครื่องกำเนิดไฟฟ้า (Generator Speed) ไปยังความเร็วที่ต้องการ (ความเร็วเชิงโคโรนัส)
- ค. ควบคุมด้วยความเร่งที่มีขนาดเล็กจากจุด c ในชิก C เพื่อที่จะป้องกันการเลื่อนของอุปกรณ์ความเร็วเคลื่อนที่ไปทางด้านลบ
- ง. ควบคุมด้วยความเร่งอย่างรุนแรงจากจุด d ชิก D เพื่อที่จะป้องกันไม่ให้อุปกรณ์ของความเร็วจนเคลื่อนที่ไปทางขวามากไปกว่านี้
- จ. ควบคุมด้วยความเร่งที่มีขนาดเล็ก จากจุด e ในชิก E เพื่อที่จะเลื่อนความเร็วของเครื่องกำเนิดไฟฟ้าไปยังความเร็วที่ต้องการ (ความเร็วเชิงโคโรนัส)
- ฉ. ควบคุมด้วยความหน่วงที่มีขนาดเล็กจากจุด f ในชิก F เพื่อที่จะป้องกันการเลื่อนของอุปกรณ์ความเร็วเคลื่อนที่ไปทางด้านลบ

จากสมการที่กล่าวมานี้ $G(k)$ บ่งชี้อัตราขยาย (Gain Factor) ที่เวลา $t = kDt$ และ $G(k)$ เป็นฟังก์ชันที่ไม่เป็นเชิงเส้น ดังรูปที่ 5 จะเห็นได้ว่า $G(k)$ เป็นสัดส่วนกับ ระยะทาง $D(k)$ จากจุด สมดุลที่พอใจ เพราะเป็นตัวบ่งชี้ว่า ขณะนั้นจุด $p(k)$ อยู่ห่างจากจุดกำเนิดเท่าใด



รูปที่ 2-9 Gain factor $g(k)$ กับ distance $D(k)$

การปรับตั้งค่าพารามิเตอร์ (Parameter Setting)

การแสดงในหัวข้อก่อนหน้านี้มีพารามิเตอร์หลายค่าที่จะปรับเป็นค่าจริงการเลือกเส้น (Switching line) L1 ระหว่างสัญญาณเสถียรภาพบวก และลบถูกเจาะจงแยกออกมาจากตำแหน่งบวก ทำมุม 45 ทิศทาง ทวนเข็มนาฬิกา กับแกนตั้ง การเลื่อนของการเลือกเส้น L1 นี้อาจจะจำเป็นเพื่อที่จะพัฒนา พฤติกรรมของเครื่องกำเนิดไฟฟ้าให้มากขึ้น ตำแหน่งของการเลือกเส้นอื่นๆ L2 และ L3 เพื่อที่จะเลือก U_{min} และ U_{max} ถูกกลั่นกรองโดยการเลื่อนของเฟส (Phase Shift) จากแกนตั้ง หรือแกนนอน ในรูปที่ 7

ค่าสูงสุดของสัญญาณเสถียรภาพ U_{max} ขึ้นอยู่กับธรรมชาติของระบบ เพราะฉะนั้นค่าของ U_{max} ควรจะเจาะจงที่ค่าสูงสุด ซึ่งสามารถประยุกต์ไปเป็นหน่วยของเครื่องกำเนิดไฟฟ้า (The Generating Unit) ที่จุด เสถียรภาพที่ต้องการจะถูกติดตั้งค่าของ U_r ควรที่จะเลือกโดยไม่ให้เกิน U_{max} ที่กำหนดไว้ ยิ่งไปกว่านั้น การตั้งค่าของ D_r ในช่วงที่สั้นเกินไป จะทำให้เกิดการแกว่ง (Hunting) รอบๆจุดสมดุล (จุดกำเนิดในระนาบของมุม)

บทที่ 3

การไหลของภาระ (Load Flow Methods)

การไหลของภาระหรือการไหลของกำลัง (Power Flow) เป็นผลซึ่งได้มาจากระบบไฟฟ้ากำลังภายใต้การทำงานในสภาวะสถิตย์ (Static) การศึกษาการไหลของภาระจะทำให้ทราบถึง

- 1.เส้นทางการไหล (Line Flow)
- 2.แรงดันที่บัสและรูปด้านนอกของแรงดันของระบบ (System Voltage Profile)
- 3.ผลการเปลี่ยนแปลงในรูปร่างของวงจรรวมกันของวงจรรวมใหม่ในภาระของระบบ
- 4.ผลของการสูญเสียชั่วคราวของความจุในการส่งจ่าย (Transmission Capacity) และ (หรือ) การผลิตบนภาระของระบบ (Generation On System Loading) และผลที่เกี่ยวข้อง
- 5.ผลของแรงดันบัสที่ควอดราเจอร์และในเฟส (In-Phase and Quadrature Boost Voltage) ของภาระในระบบ
- 6.การปฏิบัติการของระบบในเชิงเศรษฐศาสตร์
- 7.การลดลงของการสูญเสียในระบบสายส่ง
- 8.การตั้งแทปของหม้อแปลงในการปฏิบัติการเชิงเศรษฐศาสตร์
- 9.การปรับปรุงที่เป็นไปได้ในระบบเดิมโดยการเปลี่ยนขนาดตัวนำและแรงดันของระบบ

สาเหตุที่มีการศึกษาการไหลของภาระเพราะการจำลองระบบให้อยู่ในรูปเฟสเดี่ยวนั้น จะใช้ได้ในระบบที่สมดุลเท่านั้น แต่ในระบบที่ไม่สมดุลจำใช้ไม่ได้ ในปัจจุบันการจำลองของการไหลของภาระได้เปลี่ยนจากระบบอนาล็อกไปเป็นดิจิทัลเพราะเครื่องคอมพิวเตอร์ในปัจจุบันสามารถเก็บข้อมูลได้จำนวนมากและทำงานได้อย่างรวดเร็ว ในวงจรโครงตาข่ายจะมีตัวแปรทั้งหมด 4 ตัวในแต่ละบัสหรือโหนด ตัวแปรทั้งหมดคือ

1. ขนาดของแรงดัน (Voltage-Magnitude)
2. มุมเฟสของแรงดัน (Voltage Phase Angle)
3. กำลังจริง (Real Power)
4. กำลังจินตภาพ (Reactive Power)

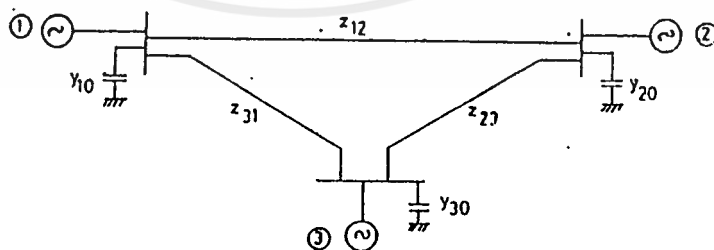
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในตัวแปร 4 ตัวนี้จะมีการกำหนดในแต่ละบัสเพียง 2 ตัวและอีก 2 ตัวจะได้จากการคำนวณการไหลของภาระที่สแลคบัส (Slack Bus) หรือสวิงบัส (Swing Bus) จะมีการกำหนดขนาดของแรงดันและมุมเฟสของแรงดัน เพื่อใช้ในการหาค่ากำลังที่สูญเสีย (Power Loss) ที่จะทำให้เกิดความสมดุลในการผลิต ที่บัสเครื่องกำเนิดไฟฟ้า (Generator Bus) ทั่ว ๆ ไปจะมีการกำหนด ขนาดของแรงดันและกำลังจริง และที่บัสภาระ (Load Bus) จะมีการกำหนดค่าความต้องการของภาระ (Real & Reactive Load Demand)

ตารางที่ 3.1

บัส	ตัวแปรที่กำหนด	ตัวแปรที่คำนวณ
บัสสแลค (Slack-Bus)	ขนาดของแรงดันและมุมเฟส	กำลังจริงและกำลังจินตภาพ
บัสเครื่องกำเนิด (Generator Bus) (PV-Bus หรือ บัสควบคุมแรงดัน)	ขนาดของแรงดันที่บัส และกำลังจริง (จะกำหนดโดยกำลังจินตภาพ)	มุมเฟสของแรงดันและกำลังจินตภาพ
บัสภาระ (Load Bus)	กำลังจริงและกำลังจินตภาพ	ขนาดและมุมเฟสของแรงดันที่บัส

แบบจำลองที่ใช้ในการศึกษา load flow



รูปที่ 3-1 แสดงระบบจ่ายไฟฟ้าชนิดที่มี 3 บัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่ารวมของ (capacitance susceptances) ที่แต่ละบัสจะแทนโดย y_{10} , y_{20} และ y_{30} เมื่อใช้กฎของเคอร์ชอฟที่แต่ละบัสจะได้ว่า

$$I_1 = V_1 y_{10} + (V_1 - V_2) y_{12} + (V_1 - V_3) y_{13}$$

$$I_2 = V_2 y_{20} + (V_2 - V_1) y_{21} + (V_2 - V_3) y_{23}$$

$$I_3 = V_3 y_{30} + (V_3 - V_1) y_{31} + (V_3 - V_2) y_{32}$$

ซึ่งจะเขียนให้อยู่ในรูปเมตริกซ์ได้ดังนี้

$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \begin{bmatrix} y_{10} + y_{12} + y_{13} & -y_{12} & -y_{13} \\ -y_{12} & y_{20} + y_{12} + y_{13} & -y_{23} \\ -y_{13} & -y_{23} & y_{30} + y_{13} + y_{23} \end{bmatrix} \times \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix}$$

$$= \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} \\ Y_{21} & Y_{22} & Y_{23} \\ Y_{31} & Y_{32} & Y_{33} \end{bmatrix} \times \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix}$$

เมื่อ $Y_{11} = y_{10} + y_{12} + y_{13}$

$Y_{22} = y_{20} + y_{12} + y_{23}$

$Y_{33} = y_{30} + y_{13} + y_{23}$

เป็นค่า self admittance ที่สร้างรูปแบบทางเส้นแท่งมุมและเมตริกซ์ของค่าแอดมิตแตนซ์

$Y_{12} = Y_{21} = -y_{12}$

$Y_{13} = Y_{31} = -y_{13}$

$Y_{23} = Y_{32} = -y_{23}$

สำหรับระบบที่ประกอบด้วย n บัส องค์ประกอบของเมตริกซ์ของค่าแอดมิตแตนซ์สามารถที่เขียนให้อยู่ในรูปของรูปแบบทางเส้นแท่งมุมได้ดังนี้

$$Y_{ii} = y_{i0} + \sum y_{ik} \quad (k=1 \text{ to } n, k \neq i)$$

$$Y_{ik} = -y_{ik}$$

ถ้าองค์ประกอบของวงจรมีค่าแอดมิตแตนซ์เกี่ยวข้อง (Mutual Admittance) เราจะไม่ใช้สมการข้างต้น สำหรับรูปแบบทางระบบของ y บัส เราจะใช้ทฤษฎีกราฟที่เป็นเชิงเส้นพร้อมด้วยวิธีการแปลงแบบเดียวกัน

แบบจำลองของระบบในการศึกษาการไหลของภาระ (System Model For Load Flow Studies)

ค่าตัวแปรและค่าพารามิเตอร์ต่าง ๆ ที่สัมพันธ์กับบัส i และบัส k ซึ่งเป็นบัสโกล์ เคียงสามารถที่จะถูกแสดงได้รูปแบบทั่ว ๆ ไปได้ดังนี้

$$V_i = |V_i| \exp j\delta_i = V_i (\cos\delta_i + j\sin\delta_i) \quad (3.1)$$

สำหรับค่าแอดมิตแตนซ์ของบัส

$$Y_{ik} = |Y_{ik}| (\cos\theta_{ik} + j\sin\theta_{ik}) \quad (3.2)$$

และค่ากำลัง VA

$$\begin{aligned} S_i &= P_i + jQ_i \\ &= V_i I_i^* \end{aligned} \quad (3.3)$$

เมื่อใช้อินติส G และ L สำหรับต้นกำเนิดและภาระ

$$\begin{aligned} P_i &= P_{gi} - P_{li} \\ &= \text{Re}[V_i I_i^*] \end{aligned} \quad (3.4)$$

$$Q_i = Q_{gi} - Q_{li} = \text{Im}[V_i I_i^*] \quad (3.5)$$

และค่าของกระแสบัสคือ

$$I_{bus} = Y_{bus} * V_{bus} \quad (3.6)$$

ดังนั้นจากสมการที่ 3.3 และ 3.4 สำหรับระบบที่มี n บัส

$$I_i = (P_i - jQ_i) / V_i^* = Y_{ii} V_i + \sum Y_{ik} V_k \quad (3.7)$$

และจากสมการที่ 7 เราจะได้ว่า

$$V_i = -1/Y_{ii} * [(P_i - jQ_i) / V_i^* - \sum Y_{ik} V_k] \quad (3.8)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่สามารถนำออกเผยแพร่โดยไม่ผ่านการคัดค้าน
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$P_i + jQ_i = V_i \sum Y_{ik}^* V_{ik}^* \quad (3.9) \quad \square$$

สามารถแสดงในรูปของโพล่าได้เป็น

$$P_i + jQ_i = \sum |V_i V_k Y_{ik}| \exp(\delta_i - \delta_k - \theta_{ik}) \quad (3.10)$$

ดังนั้น

$$P_i = \sum |V_i V_k Y_{ik}| \cos(\delta_i - \delta_k - \theta_{ik}) \quad (3.11)$$

และ

$$Q_i = \sum |V_i V_k Y_{ik}| \sin(\delta_i - \delta_k - \theta_{ik}) \quad (3.12)$$

สมการการไหลของพลังงาน (3.11) และ (3.12) มีลักษณะที่ไม่เป็นเชิงเส้นจึงมีความจำเป็นที่จะต้องทำการแก้สมการที่ $2(n-1)$ รวมทั้ง $|V_i|$, P_i และ Q_i สำหรับในแต่ละบัสที่ i สำหรับการแก้สมการการไหลของภาวะ ท้ายสุดค่าของกำลังที่สแลคบัสนี้จะต้องถูกคำนวณจากค่าของการสูญเสียและปัจจัยอื่น ๆ วิธีการอินเทอเรทีฟเมตริกซ์ Y (Y -Matrix Iterative Methods) อยู่บนพื้นฐานของการแก้สมการเกี่ยวกับการไหลของกำลังโดยใช้ค่าของการแสมที่มิสแมชดังสมการที่

$$\Delta I_i = I_i - \sum V_k Y_{ik} \quad (3.13)$$

$$\Delta V_i = I_i / Y_{ii} \quad (3.14)$$

การลู่เข้าของวิธีการดังกล่าวขึ้นอยู่กับโดเมนซ์ทางเส้นแท่งมุมของเมตริกซ์ของค่าบัสนัดมิตแดนซ์ ค่าแอดมิสแดนซ์ที่เหนี่ยวนำขึ้นเองของบัสนี้จะมีค่ามากเมื่อเทียบกับค่าแอดมิสแดนซ์ที่ถูกเหนี่ยวนำขึ้นมา จุดเชื่อมของค่าอิมพีแดนซ์ที่มีค่าสูงและต่ำที่ต่ออนุกรมและค่าคาปาซิแตนซ์ซึ่งจะมีในสายส่งที่มีระยะทางไกล (Cable Circuit Long Line) EHV การเลือกสแลคบัสนี้มีผลต่อการลู่เข้าอย่างมาก ลักษณะที่เป็นขาลื่นวิธีการอินเทอเรทีฟเมตริกซ์ Y ซึ่งก็คือองค์ประกอบในรูปของการบวกในสมการที่ 3.7 และสมการที่ 3.8 ในกรณีที่เป็นระบบที่มีเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ณที่เป็นระบบที่มี n บัสจำนวนของการทำซ้ำจะเท่ากับจำนวนเท่าของ n และเวลาที่ใช้ในการคำนวณจะประมาณเท่ากับ n^2

$$V = |Y^{-1}| = Z \cdot I \quad (3.15)$$

วิธี Z เมตริกซ์โดยปกติแล้วจะไม่ใช้สำหรับการเลือกสแลคบัส ส่วนปัญหาที่สามารถใช้วิธี Z เมตริกซ์ และวิธี Y เมตริกซ์จึงมักจะใช้วิธีหลังมากกว่า

วิธีการทำซ้ำของเกาส์-ไซเดล (Gauss-Seidel Iterative Method)

ในวิธีนี้ เราจะทำการสมมุติค่าโวลต์เดจของบัสทุกบัสยกเว้นที่สแลคบัส ส่วนค่าของบัสโวลต์เดจจะถูกกำหนดและจำค่าไว้โดยไม่มีการเปลี่ยนแปลง เราสามารถเขียนสมการแรงดันที่บัส $n-1$ คือ

$$V_i = \frac{1}{Y_{ii}} [(P_i - jQ_i)/V_i^* - \sum_{k=1 \text{ to } n, k \neq i} Y_{ik} V_k] \quad (3.16)$$

$i \leftrightarrow \text{slack bus}$

สมการข้างต้นจะถูกแก้สมการเพื่อนำมาใช้แก้ปัญหาให้ดีขึ้น เพื่อที่จะทำการเร่งการลู่เข้า ค่าของแรงดันบัสที่ถูกคำนวณขึ้นมาใหม่จะถูกแทนในสมการที่ 3.16 และสมการแรงดันบัสสำหรับการทำซ้ำครั้งที่ $(m + 1)^{\text{th}}$ สามารถเขียนได้เป็น

$$V_i^{(m+1)} = \frac{1}{Y_{ii}} [(P_i - jQ_i)/V_i^{(m)*} - \sum_{k=1}^n Y_{ik} V_k^{(m+1)}] \quad (3.17)$$

$$\Delta V_i = \alpha \Delta S_i^* / V_i^* Y_{ii} \quad (3.18)$$

ในการใช้แฟลเตอร์ความเร่งปริมาณเท่ากับเอ็กซ์ตรีวาโพลชันที่ไม่เป็นเชิงเส้นน้อย ๆ ของ V_i สำหรับระบบที่ให้มา เรามักจะพบบ่อย ๆ ว่าการเลือกที่ดีที่สุดของค่า α ที่มีอยู่ซึ่งใช้ได้ในช่วงของสภาพการทำงาน แฟลเตอร์ความเร่งที่แตกต่างกันอาจจะถูกใช้ อาจจะถูกใช้สำหรับส่วนที่เป็นค่าจริงและค่าจินตภาพของค่าแรงดัน

$$|V_i^{(m)}| \text{ ang}(\delta_i) = |\alpha| |V_i^{(m)}| \text{ ang}(\delta_i) \quad (3.19)$$

วิธีการสำหรับสำหรับการจัดการกับบัส PV จำเป็นต้องใช้การนำเสนอที่เป็นเรคแทนกล่าวเราจะให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$V_i = v'_i + jv''_i \quad (3.20)$$

โดยที่ V_i และ V''_{ii} เป็นค่าจริงและค่าอิมเมจินของ V_1

$$v'_i{}^2 + v''_i{}^2 = |V_i|_{\text{scheduled}}^2 \quad (3.21)$$

เราจำเป็นต้องใช้ค่ากำลังเสมือนของบัสเพื่อที่จะคำนวณแรงดันของบัส การประมาณค่าส่วนประกอบของแรงดัน $v'_i(m)$ และ $v''_i(m)$ หลังจากการทำซ้ำ m ครั้งจะต้องถูกปรับเพื่อให้สอดคล้องกับสมการที่ 4.21 มุมเฟสของค่าแรงดันที่ประมาณคือ

$$\delta_i(m) = \tan^{-1}[v''_i(m) / v'_i(m)] \quad (3.22)$$

สมมุติว่าค่ามุมเฟสของแรงดันที่ประมาณและค่าโวลต์เดจที่วางไว้ (Schedule Voltage) มีค่าเท่ากัน เราสามารถปรับค่า $v'_i(m)$ และ $v''_i(m)$ เป็น

$$v'_i(m)_{\text{(new)}} = |V_i|_{\text{scheduled}} \cos \delta_i(m) \quad (3.23)$$

และจะได้ว่า

$$v''_i(m)_{\text{(new)}} = |V_i|_{\text{scheduled}} \sin \delta_i(m) \quad (3.24)$$

เราจะใช้ค่านี้ในการคำนวณค่ากำลังเสมือน $Q_i(m)$ และจะทำการคำนวณค่า $V_i(m+1)$ โดยใช้ค่ากำลังเสมือนและค่าแรงดัน $V_{\text{(new)}}(m)$ โดยได้แสดงขั้นตอนการทำงานของโปรแกรมสำหรับการคำนวณการไหลของภาระโดยใช้วิธีการของเกาส์ไซเดลไว้ในบทที่ 5 ขั้นตอนและการทำงาน

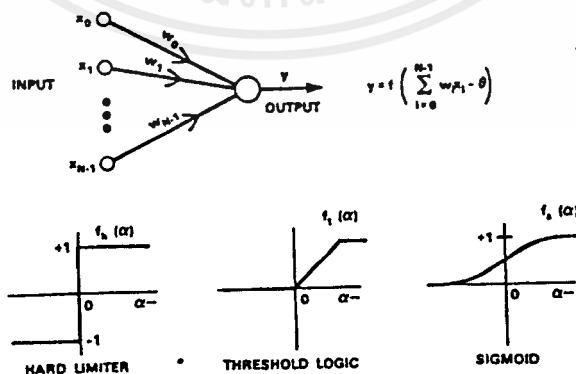
ในขณะที่คำนวณค่ากำลังเสมือนเราจะต้องพิจารณาขีดจำกัดของแหล่งกำลังเสมือนด้วยถ้าค่ากำลังเสมือนที่คำนวณมาได้มากกว่าค่าที่จำกัดไว้ เราจะกำหนดค่าไว้ที่ค่าจำกัดนั้น และถ้าไม่มีความจำเป็นที่จะต้องคงขนาดของแรงดันไว้ บัสจะเป็นลักษณะ PQ บัสหรือโหลดบัส

บทที่ 4

โครงตาข่ายนิวรอล (ARTIFICIAL NEURAL NETWORK)

4.1 ความรู้เบื้องต้น (Introduction)

แบบจำลองของโครงตาข่ายนิวรอลที่สร้างขึ้น (Artificial Neural Net) หรือที่เรียกกันง่าย ๆ ว่า โครงตาข่ายนิวรอล (Neural net) หรืออาจจะเรียกได้หลายชื่อเช่น แบบจำลองของการต่อเชื่อม (Connectionist Model) แบบจำลองของกระบวนการการกระจายแบบขนาน (Parallel Distributed Processing Model) หรือระบบแบบนิวโรโมฟิก (Neuromorphic System) โดยโครงสร้างของโครงตาข่ายนิวรอลอยู่บนพื้นฐานของความเข้าใจในระบบประสาทในทางชีววิทยา แบบจำลองของโครงตาข่ายนิวรอลจะมีศักยภาพสูงสำหรับในบางพื้นที่ เช่น สปีชและอิมเมจเรคognition (Speech and Image Recognition) และในเวลาเดียวกันได้มีการกำหนดสมมุติฐานต่าง ๆ ตามไปด้วย แทนที่จะแสดงโปรแกรมในรูปของคำสั่งเหมือนในคอมพิวเตอร์ของ วอนนุแมน (Von Neumann) แต่ในขณะเดียวกันแบบจำลองของโครงตาข่ายนิวรอลได้พยายามค้นหาสมมุติฐานหลาย ๆ อย่าง โดยการใช้โครงตาข่ายแบบขนาน (Parallel Net) ซึ่งประกอบด้วยส่วนที่เกี่ยวกับการคำนวณ (Computational Elements) หลายชนิด ซึ่งถูกเชื่อมต่อกับด้วยค่าน้ำหนักต่าง ๆ



รูปที่ 4-1 แสดงโนดซึ่งจะรวมค่าผลคูณของอินพุท และค่าน้ำหนักจำนวน N อินพุท แล้วผ่านผลลัพธ์ในลักษณะที่ไม่เป็นเชิงเส้น โดยได้แสดงความไม่เป็นเชิงเส้นไว้ 3 ลักษณะ

ส่วนที่เกี่ยวข้องกับการคำนวณหรือที่เรียกว่าโนด (Node) ซึ่งใช้ในแบบจำลองของโครงตาข่ายนิวรอลมีลักษณะที่ไม่เป็นเชิงเส้น โดยมีลักษณะเป็นแอนนาลอก ซึ่งจะซ้ำเมื่อเปรียบเทียบกับวงจรทางดิจิทัล (Digital Circuitry) และในรูปที่ 1 ได้แสดงแบบจำลองอย่างง่ายซึ่งประกอบด้วยอินพุต N ตัวคือ (x_1, x_2, \dots, x_n) ถูกป้อนเข้าไปในวงจรโดยผ่านทางน้ำหนักที่สอดคล้องกันคือ (w_1, w_2, \dots, w_n) โดยโนดจะถูกกำหนดคุณลักษณะโดยค่าทริสโพลหรือออฟเซ็ทและชนิดของความไม่เป็นเชิงเส้น (Nonlinearity) ในรูปที่ 1 ได้แสดงถึงชนิดของความไม่เป็นเชิงเส้นที่มีลักษณะทั่ว ๆ ไป 3 แบบคือ Hard Limiter , องค์ประกอบทางโลจิกทริสโพล (Threshold Logic Element) และความไม่เป็นเชิงเส้นแบบซิกมอยด์ (Sigmoidal Nonlinearity) และในกรณีของโนดที่มีความซับซ้อนมากกว่านี้อาจจะประกอบด้วยการรวมแบบเทมเพอรอล (Temporal Intigation & Time Dependency) ชนิดอื่น ๆ รวมทั้งมีการคำนวณทางคณิตศาสตร์ที่ซับซ้อนขึ้น

แบบจำลองของโครงตาข่ายนิวรอลจะถูกกำหนดโดยเน็ตโทโพลยี (Net Topology), คุณลักษณะของโนดและกฎของการฝึกและการสอน (Training & Learning Rule) และจะเป็นตัวกำหนดกลุ่มของน้ำหนักเริ่มต้น ซึ่งจะเป็นตัวที่บ่งชี้ว่าควรจะมีการปรับค่าน้ำหนักหรือไม่ รวมทั้งวิธีการออกแบบและกฎของการฝึก

ศักยภาพของโครงตาข่ายนิวรอลสูงกว่าในส่วนของอัตราการคำนวณ (Computation Rate) เมื่อเปรียบเทียบกับแมสซีฟพาราเรลลิซึม (Massive Parallelism) นอกจากนี้โครงตาข่ายนิวรอลสามารถให้ระดับของฟอลท์เทอโรแลนซ์ (Fault Tolerance) เนื่องจากมีจำนวนโนดที่จะใช้ในการโพรเซส (Processing Node) มากกว่าโดยแต่ละโนดจะมีการต่อเชื่อมระหว่างกัน ดังนั้นเมื่อเกิดความเสียหายตรงจุดเชื่อมต่อหรือที่โนดใด ๆ ก็จะไม่ทำความเดือดร้อนต่อระบบทั้งหมด ขั้นตอนการทำงานของโครงตาข่ายนิวรอลส่วนใหญ่จะทำการปรับค่าน้ำหนักเพื่อที่จะปรับระบบการทำงานของโครงตาข่ายให้ดีขึ้นโดยการใช้การเรียนรู้ ซึ่งเป็นหัวข้อสำคัญสำหรับการวิจัยโครงตาข่ายนิวรอล ในส่วนของความสามารถที่จะทำการปรับค่าน้ำหนักหรือการเรียนรู้นั้นเป็นจุดที่เราจะต้องให้ความสำคัญและมีความจำเป็นอย่างยิ่งสำหรับบางบริเวณ เช่น สปีชรีคอกนิชัน (Speech Recognition) ซึ่งข้อมูลสำหรับการฝึกจะถูกกำหนดขอบเขต

ในการออกแบบโครงตาข่ายนิวรอลสำหรับการแก้ปัญหาและการเรียนรู้โครงตาข่ายทางชีววิทยาที่แท้จริงได้นำไปสู่ความเข้าใจและการปรับปรุงขั้นตอนการทำงาน

การศึกษาการออกแบบโครงตาข่ายนิวรอลได้ทำมาเป็นระยะเวลาอันนานในการพัฒนาเกี่ยวกับแบบจำลองทางคณิตศาสตร์ได้เริ่มต้นมานานกว่า 40 ปียกตัวอย่างเช่นงานของ McCulloch และ Pitts งานของ Hebb , Rosenblatt, Widrow และคนอื่น ๆ อีก และในส่วนงานที่ผ่านมานี้ของ Hopfield, Rumelhart และ Mclelland, Sejnowsky, Feldman, Grossberg เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้ทำให้งานในสาขานี้ได้กลับมาได้เกิดความสนใจอีกครั้ง และความสนใจนี้ได้นำไปสู่การพัฒนาขั้นตอนการทำงานและโครงตาข่ายโทโพโลยี (Topology Net) รวมทั้งเทคนิค VSLI แบบแอนะล็อก (Analog VSLI Implementation Techniques) รวมทั้งได้มีการศึกษาค้นคว้าการทำงานของระบบสมองของมนุษย์เพิ่มขึ้นด้วย

ทฤษฎีโครงตาข่ายนิรพลนี้ได้รับการวิเคราะห์ค้นคว้าขึ้นมาใหม่เพื่อนำมาวิเคราะห์เสถียรภาพในสภาวะคงที่ (Steady State Stability) ของเครื่องจักรซินโครนัสซึ่งค่าอินพุทที่ป้อนเข้าไปในโครงตาข่ายก็คือตัวแปรของระบบซึ่งมีบทบาทสำคัญต่อเสถียรภาพของระบบในสภาวะคงที่ดังที่ได้กล่าวมาข้างต้นซึ่งจุดประสงค์ที่ต้องมีการควบคุมเสถียรภาพในสภาวะคงที่ ก็เพื่อทำการตรวจสอบพฤติกรรมทางไดนามิกเมื่อมีสิ่งรบกวนเข้าไปในระบบ (Small Perturbation) เนื่องจากว่าเครื่องจักรซินโครนัสจะต้องสามารถทำงานได้อย่างเป็นปกติในกรณีที่เกิดการเปลี่ยนแปลงไปเป็นระยะ ๆ จากความรู้ที่ว่าแรงบิดหน่วง (Damping Torque) ของเครื่องจักรซินโครนัสจะได้รับผลกระทบจากปัจจัยหลายอย่าง เช่น ภาระของเครื่องจักร ค่าคงที่สำหรับการกระตุ้น (Exciter Constant) ตัวปรับเสถียรภาพของระบบ (Power System Stabilizer, PSS) เป็นต้น เพราะฉะนั้นสำหรับแบบจำลองทางคณิตศาสตร์ ที่เราจะนำไปใช้ในการประเมินเสถียรภาพของระบบในสภาวะคงที่ที่จะต้องประกอบด้วยพารามิเตอร์ที่กล่าวมาแล้วข้างต้น

ในช่วง 2-3 ปีที่ผ่านมาเราใช้พยายามอย่างมากในการที่จะวิเคราะห์เสถียรภาพในสภาวะคงที่ของระบบไม่ว่าจะเป็นวิธีโดเมนความถี่ (Frequency Domain) หรือวิธีการจำลองโดเมนของเวลา (Time Domain) ซึ่งได้นำวิธีดังกล่าวมาใช้โดยวิศวกรในการประเมินค่าของภาระและพารามิเตอร์ PSS แต่วิธีการดังที่ได้กล่าวมาเหมาะสำหรับวิเคราะห์เสถียรภาพแบบออฟไลน์ (Off-Line) เพื่อใช้ในการวางแผนระบบ (System Planning) แต่ไม่เหมาะที่จะนำวิธีนี้มาใช้ประเมินเสถียรภาพที่เป็นออนไลน์ (On-line) ซึ่งถ้าหากนำวิธีการเหล่านี้มาวิเคราะห์เสถียรภาพที่เป็นออนไลน์ (On-Line) เพราะจะต้องใช้เวลาคำนวณนานในการแก้สมการดิฟเฟอเรนเชียล

เราได้นำทฤษฎีโครงตาข่ายนิรพลได้นำมาประยุกต์ใช้ในการแก้ปัญหาทางระบบไฟฟ้าซึ่งสามารถนำไปใช้แก้ปัญหาที่ซับซ้อนได้ในเวลาอันสั้นและมีความสามารถสูงในการวิเคราะห์เสถียรภาพที่เป็นลักษณะออนไลน์ (ON-Line Stability Analysis) เช่นการวิเคราะห์เสถียรภาพในขณะทรานเซียน การประเมินความปลอดภัยทางสแตติกส์ การกำหนดภาระฮาโมนิก (Harmonic Load Identification) การควบคุมด้วยตัวเก็บประจุ (Capacitor Control)

Aggoune ได้พัฒนาโครงตาข่ายนิรพลแบบป้อนไปข้างหน้าแบบขั้นซับซ้อนเพื่อนำมาใช้วิเคราะห์เสถียรภาพทางไดนามิกในลักษณะที่เป็นออนไลน์แต่ก็ยังมีข้อเสียเปรียบในโครงสร้างเมื่อนำไปใช้ในการวิเคราะห์เสถียรภาพในสภาวะทรานเซียนหรืองานการควบคุม

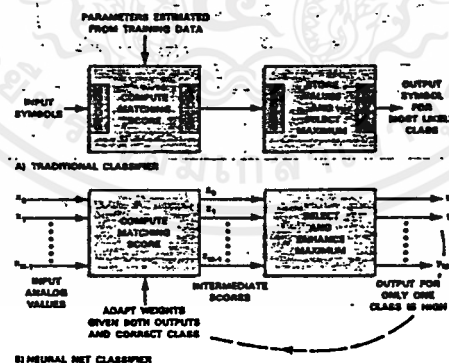
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยตัวเก็บประจุ ซึ่งยังมีการขาดเทคนิคที่มีประสิทธิภาพเพียงพอสำหรับโครงตาข่ายนิรवलลชนิดนี้

4.2 โครงตาข่ายนิรवलลและตัวจัดกลุ่ม

(Neural Network and Traditional Classifier)

สำหรับบล็อกไดอะแกรมของตัวจัดกลุ่มของโครงตาข่ายนิรवलล ได้แสดงไว้ในรูปที่ 2 โดยตัวจัดกลุ่มทั้งสองชนิดจะเป็นตัวกำหนดคลาสซึ่งจะเป็นตัวที่แสดงถึงรูปแบบของอินพุทในทางสถิติ ในรูปที่ 2 ตัวจัดกลุ่มจะประกอบด้วยส่วนสำคัญ 2 ส่วนในส่วนแรกจะเป็นส่วนที่คำนวณค่าที่แสดงการเข้ากัน (Matching Score) สำหรับแต่ละคลาส และในส่วนที่ 2 จะทำหน้าที่เลือกคลาสที่มีแต้มสูงสุด ค่าอินพุทที่ถูกป้อนเข้าไปในส่วนที่ 1 ของตัวแบ่งกลุ่มจะมีลักษณะที่เป็นสัญลักษณ์ โดยค่าเหล่านี้จะถูกถอดรหัสจากแบบสัญลักษณ์ดังกล่าวไปเป็นรูปแบบสัญลักษณ์ภายใน (Internal Symbolic Form) จะมีขั้นตอนที่จะทำการคำนวณค่าการเข้ากันได้สำหรับแต่ละคลาสโดยใช้วิธีการกระจายของเกาส์เซียน (Gaussian Distribution) ซึ่งตัวแต้มที่เข้ากันได้นี้จะเป็นตัวบ่งชี้ว่าค่าอินพุทจะเข้ากันได้ (Match) กับแต่ละคลาสอย่างไร หลังจากนั้นค่าที่แสดงถึงการเข้ากันจะถูกเข้ารหัสให้อยู่ในรูปของสัญลักษณ์และจะถูกส่งผ่านไปยังส่วนที่ 2 ของตัวจัดกลุ่มซึ่งจะถูกถอดรหัสและจะทำการเลือกคลาสที่มีสคอร์สูงสุด เอาท์พุทซึ่งเป็นสัญลักษณ์ที่แสดงถึงคลาส จะถูกส่งออกมาเพื่อให้กระบวนการแยกคลาส (Classification) สำเร็จอย่างสมบูรณ์



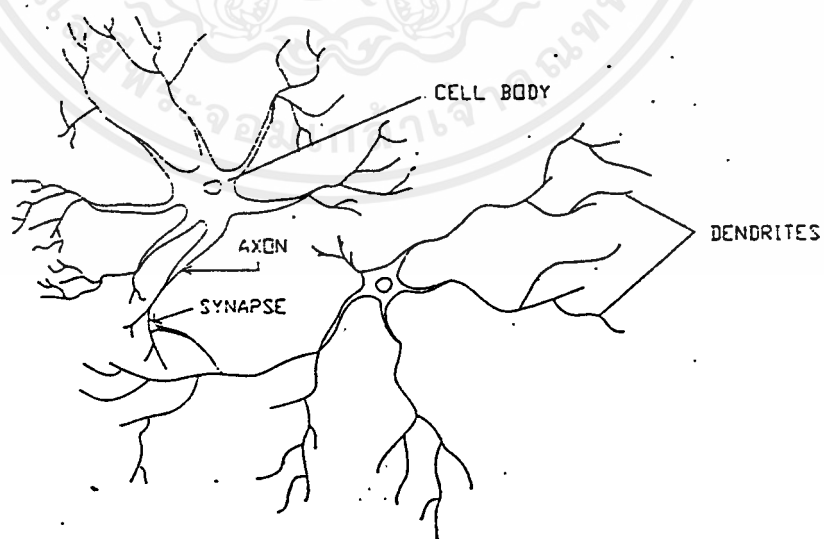
รูปที่ 4-2 บล็อกไดอะแกรมของตัวจัดกลุ่ม A และ B ค่าอินพุท และอินพุทของตัวจัดกลุ่มที่ใช้อยู่จะถูกผ่านเข้าไปในลักษณะอนุกรม หลังจากนั้นจะมีการคำนวณภายในบล็อกไดอะแกรมโดยจะประมาณค่าพารามิเตอร์ต่างๆ จากข้อมูลการฝึก และจะคงค่านั้นไว้สอน อินพุทและเอาท์พุทของตัวจัดกลุ่มของโครงตาข่ายนิรवलลจะถูกผ่านในลักษณะขนาน โดยค่าพารามิเตอร์ และน้ำหนักจะถูกปรับ หรือฝึก ในขณะที่มีการใช้ค่า เอาท์พุท หากคลาสที่ต้องการ

4.3 หลักการเบื้องต้นของทฤษฎีโครงตาข่ายนิวรอล (Fundamental Neural Network)

ทฤษฎีโครงตาข่ายนิวรอลได้มีการพัฒนามาในรูปแบบที่แตกต่างกันหลายแบบ โดยเราจะมาพิจารณาหลักพื้นฐานเบื้องต้นเกี่ยวกับทฤษฎีโครงตาข่ายนี้ เพื่อให้เกิดความเข้าใจเบื้องต้น ก่อนที่จะทำการศึกษารายละเอียดอื่น ๆ ในตอนหลัง

4.3.1 ต้นแบบทางชีววิทยา (The Biological Prototype)

ทฤษฎีโครงตาข่ายนิวรอล ได้ถูกสร้างขึ้นโดยเลียนแบบทางชีววิทยา ซึ่งมีการเทียบเคียงกับระบบสมองของมนุษย์ เมื่อเราพิจารณาถึงขั้นตอนการทำงานและลักษณะของโครงตาข่าย แต่เนื่องจากเรามีความรู้จำกัดในเรื่องของการทำงานของระบบสมอง จึงมีความสามารถที่จะเลียนแบบหลักการของโครงตาข่ายกับการทำงานของระบบสมองได้อย่างจำกัด ดังนั้นผู้ที่ต้องการศึกษาถึงโครงตาข่ายนิวรอลจะต้องค้นคว้าหาความรู้ทางชีววิทยาให้มากกว่าที่มีอยู่ในปัจจุบัน และจะต้องคิดโครงสร้างรวมทั้งแสดงการทำงานซึ่งสามารถนำมาใช้ประโยชน์ได้ โดยการทำงานของโครงสร้างจะเทียบเคียงกับวิธีการรับรู้ของคนเรา จึงจำเป็นต้องมีการศึกษาการเปรียบเทียบ(analogy) ระหว่างโครงตาข่ายนิวรอลกับระบบสมองซึ่งจะต้องทำการเรียนรู้และทำความเข้าใจระบบประสาทของมนุษย์ โดยเราสามารถสรุปโครงสร้างและขั้นตอนการทำงานของระบบประสาทที่เราสามารถนำมาใช้สอดคล้องกับทฤษฎีโครงตาข่ายนิวรอลได้ดังนี้

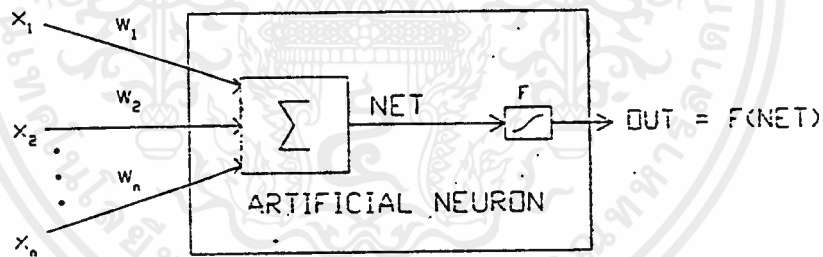


รูปที่ 4-3 แสดงลักษณะของนิวรอนในทางชีววิทยา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบประสาทของมนุษย์สร้างขึ้นและประกอบรวมกันด้วยจำนวนเซลล์ที่เราเรียกว่านิวรอน (Neuron) มีการเรียงตัวที่ซับซ้อน ประมาณ 10^{11} ตัวจะประกอบด้วย การเชื่อมต่อระหว่างกันประมาณ 10^{15} เส้นทาง นิวรอนแต่ละตัวและจะมีการแลกเปลี่ยนคุณลักษณะกับเซลล์อื่น ๆ ในร่างกาย แต่มีคุณสมบัติเฉพาะตัวในการรับ การประมวล รวมทั้งการส่งสัญญาณไฟฟ้าเคมี (Electrochemical Signal) ไปตามเส้นทางต่าง ๆ ซึ่งประกอบกันเข้าเป็นระบบการติดต่อภายในสมอง

ในรูปที่ 4-3 ได้แสดงถึงโครงสร้างทางชีววิทยาของนิวรอน 2 ตัว เดนไรท์จะแผ่ขยายจากเซลล์หนึ่งไปยังนิวรอนตัวอื่น ๆ โดยจะมีการรับสัญญาณที่จุดต่อซึ่งเรียกว่าไซแนปส์ (synapse) อินพุทเหล่านี้จะถูกนำไปยังเซลล์ต่าง ๆ เพื่อทำการรวมอินพุท โดยอินพุทบางส่วนจะกระตุ้นเซลล์หรืออาจจะทำการยับยั้งการไฟร์ริง (Firing) ในกรณีที่การกระตุ้นที่เกิดขึ้นเกินกว่าค่าทริสโฮล เซลล์จะทำการส่งสัญญาณผ่านทางแอกซอน (Axon) ไปยังนิวรอนตัวอื่น ๆ เราจะเห็นว่ารูปแบบและขั้นตอนการทำงานมีความสลับซับซ้อนซึ่งเราจะต้องนำมาใช้เพื่อออกแบบโครงตาข่ายนิวรอลต่อไป



รูปที่ 4-4 นิวรอนที่สร้างขึ้น

4.3.2 นิวรอนเทียม (Artificial Neuron)

โครงตาข่ายนิวรอน (The Artificial Neuron) ได้ถูกออกแบบไว้สำหรับการเลียนแบบคุณลักษณะอันดับหนึ่งของนิวรอนทางชีววิทยา (First Order Characteristics of Biological Neurons) โดยจะใช้กลุ่มของอินพุท ซึ่งแต่ละตัวจะเป็นเอาท์พุทของนิวรอนตัวอื่น ๆ อินพุทแต่ละตัวจะถูกคูณด้วยน้ำหนักที่สอดคล้องกัน ซึ่งเปรียบเสมือนความเครียดซินแนปส์ (Synaptic Strength) และค่าดังกล่าวทั้งหมดจะถูกบวกเข้าด้วยกัน เพื่อนำมาใช้กำหนดระดับของการกระตุ้น (Activation Level) ของนิวรอน โดยในรูปที่ 3-4 ได้แสดงถึงแบบจำลองที่สนับสนุนความคิดดังกล่าวนี้ ทั้งที่มีความหลากหลายของโครงสร้างของแบบจำลอง แต่ก็อยู่บนพื้นฐานของแบบจำลองในรูปนี้ จากรูปที่ กลุ่มของอินพุทซึ่งก็คือ x_1, x_2, \dots, x_n ถูกป้อนเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์อื่นใด ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เข้าไปในโครงข่ายนิเวศโดยเราจะอ้างอิงอินพุทเหล่านี้ในรูปของเวกเตอร์ X ซึ่งจะเปรียบเสมือนกับสัญญาณที่เข้าไปสู่ซินแนปส์ของนิวรอนในทางชีววิทยา สัญญาณดังกล่าวจะถูกคูณด้วยน้ำหนักที่สอดคล้องกันโดยใช้สัญลักษณ์ w_1, w_2, \dots, w_n ก่อนจะเข้าสู่ส่วนที่จะทำหน้าที่รวมสัญญาณ (Summation Box) ซึ่งใช้สัญลักษณ์ Σ ซึ่งน้ำหนักดังกล่าวจะสอดคล้องกับความเครียดของการเชื่อมต่อทางซินแนปส์ทางชีววิทยา (Biological Synaptic Connection) ส่วนของบล็อกที่ทำหน้าที่รวมสัญญาณก็สามารถเทียบได้โดยหยาบ ๆ กับตัวเซลล์ในทางชีววิทยา (Biological Cell Body) จะทำหน้าที่รวมน้ำหนักอินพุท (Input Weight) ตามเครื่องหมายทางพีชคณิตและจะได้สัญญาณเอาต์พุท ที่เรียกว่า NET ซึ่งสามารถเขียนในรูปของสมการเวกเตอร์ได้เป็น

$$NET = XW \tag{4.1}$$

4.3.3 ฟังก์ชันกระตุ้น (Activation Function)

สัญญาณเอาต์พุทดังกล่าวหรือ NET โดยทั่ว ๆ ไป แล้วจะถูกผ่านไปยังกระบวนการต่อไปโดยฟังก์ชันกระตุ้น F เพื่อทำการสร้างสัญญาณเอาต์พุทของนิวรอนหรือที่เรียกว่า OUT โดยสามารถเขียนเป็นสมการที่เป็นเชิงเส้นง่าย ๆ คือ

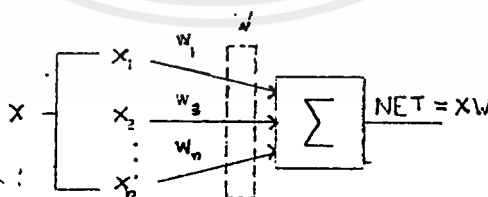
$$WT = K(NET)$$

โดยค่า K เป็นค่าคงที่ และทรานส์โวลต์ฟังก์ชันสามารถเขียนได้เป็น

$$OUT = 1 \text{ ถ้า } NET > T$$

$$ELSE \text{ } OUT = 0$$

โดยค่า T คือค่าคงที่ทรานส์โวลต์ หรืออาจจะเป็นฟังก์ชันที่จำลองถึงคุณลักษณะในการทรานสเฟอร์ที่ไม่เป็นเชิงเส้น (Nonlinear Transfer Characteristic) ของนิวรอนทางชีววิทยาได้แม่นยำกว่า



$$NET = x_1 w_1 + x_2 w_2 + \dots + x_n w_n$$

รูปที่ 4-5 นิวรอนเทียมที่สร้างขึ้นกับฟังก์ชันการกระตุ้น

โดยในรูปที่ 4-5 ส่วนของบล็อก F ที่ได้เพิ่มเติมเข้าไปในส่วนของโครงข่ายนิเวศจะรับค่าเน็ต (NET) และจะสร้างสัญญาณ OUT ออกมา ในกรณีที่กระบวนการในส่วนบล็อกเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอฟ (F) ได้บิบบัญญัติแวนเน็ต (NET) เพื่อว่าสัญญาณเอาต์พุตดังกล่าวจะได้มีค่าไม่เกินค่าต่ำสุดที่กำหนดไว้โดยไม่คำนึงถึงค่าเน็ต อย่างไรก็ตามเราจะเรียกบล็อกเอฟนี้ว่าฟังก์ชันการบดอัด (Squashing Function) โดยฟังก์ชันตัวนี้จะถูกเลือกใช้เป็นโลจิสติกส์ฟังก์ชัน (Logistic Function) หรือซิกมอยด์ (Sigmoid) มีลักษณะเป็นรูปตัวเอส (S) และได้แสดงไว้ในรูปที่ 4-6 โดยสามารถเขียนแสดงเป็นสมการทางคณิตศาสตร์

$$F(X) = 1/(1+e^X) \quad (4.2)$$

ดังนั้นจะได้ค่าเอาต์ (OUT)

$$OUT = 1/(1+e^{-net}) \quad (4.3)$$

เมื่อทำการเปรียบเทียบกับระบบในทางอิเล็กทรอนิกส์ เราสามารถพิจารณาฟังก์ชันกระตุ้น ได้กับอัตราขยายที่ไม่เชิงเส้น (Nonlinear Gain) ซึ่งอัตราขยายนี้สามารถที่จะคำนวณได้โดยการหาอัตราส่วนระหว่างการเปลี่ยนแปลงในเอาต์ (OUT) กับการเปลี่ยนแปลงจำนวนเล็กน้อยที่เกิดขึ้นในเน็ต (NET) ดังนั้นค่าอัตราขยายก็จะเป็นค่าความชันของกราฟ ที่ระดับการกระตุ้น (Excitation Level) ที่กำหนดโดยจะเปลี่ยนแปลงจากค่าน้อย ๆ ที่มีการกระตุ้นที่เป็นลบ (Negative Excitation) กราฟจะมีลักษณะเป็นแนวนอน (Horizontal) ไปเป็นค่าสูง ๆ เมื่อการกระตุ้นเป็น 0 (Zero Excitation) และจะตกลงในขณะที่การกระตุ้นมีค่ามาก ๆ และเป็นมีค่าบวก GROSS BERGE (1973) ได้ค้นพบว่าคุณลักษณะของอัตราขยายที่ไม่เป็นเชิงเส้นนี้ สามารถใช้ในการแก้ไขการเกิดการอิมิตัวของสัญญาณรบกวน (saturable Noise) และจากคำถามที่ว่าโครงตาข่ายเดียวกันสามารถใช้ได้กับสัญญาณขนาดเล็กและขนาดใหญ่ได้อย่างไร โดยสัญญาณอินพุตที่มีขนาดเล็กจะต้องการอัตราขยายที่มีขนาดใหญ่ผ่านทางโครงตาข่ายถ้าจะเอาสัญญาณนี้สร้างเอาต์พุต อย่างไรก็ตามเสด็จอัตราขยายสูงแบบคาสเคด (Cascaded High Gain Stage) จำนวนมากนี้สามารถทำให้เอาต์พุตเกิดการอิมิตัวพร้อมการเพิ่มสูงขึ้นของสัญญาณรบกวน สำหรับการแปรแบบสุ่ม (Random Variation) ซึ่งได้แสดงไว้ในโครงตาข่าย ส่วนสัญญาณอินพุตที่มีขนาดใหญ่จะทำให้เสด็จอัตราขยายสูงเกิดการอิมิตัว ในบริเวณที่อัตรากำลังขยายมีขนาดใหญ่ (High Gain Region) ของฟังก์ชันสามารถใช้แก้ปัญหาการประมวลสัญญาณขนาดเล็ก ในขณะที่บริเวณของอัตรากำลังขยายมีขนาดเล็กจะเหมาะสมกับการกระตุ้นที่ขนาดใหญ่ (Large Excitation) จากที่กล่าวมาสามารถแสดงให้เห็นว่านิรอลสามารถที่จะทำงานได้ด้วยอัตราขยายที่เหมาะสมในช่วงกว้างของระดับอินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

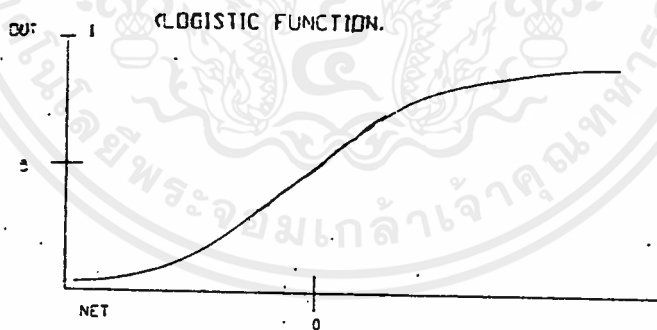
สำหรับการใช้ฟังก์ชันการกระตุ้นอีกวิธีหนึ่งคือเป็นในลักษณะของไฮเปอร์โพลิคแทนเจนท์ (Hyperpolic tangent) .ซึ่งจะมีแบบโครงสร้างที่เหมือนกับฟังก์ชันแบบโลจิสติก (Logistic Function) และมักจะถูกใช้โดยนักชีววิทยาเพื่อใช้เป็นแบบจำลองในทางคณิตศาสตร์ของการกระตุ้นของระบบเซลล์ประสาท (Nerve Cell)

กรณีของฟังก์ชันการกระตุ้นของทฤษฎีโครงตาข่ายนิวรอลเราสามารถเขียนสมการได้ดังนี้

$$\text{OUT} = \tanh(x)$$

จะเห็นว่ามึลักษณะที่คล้ายกับฟังก์ชันแบบโลจิสติก ส่วนลักษณะของเส้นกราฟฟังก์ชันแบบไฮเปอร์โพลิคแทนเจนท์จะเป็นรูปคล้ายตัว S โดยจะมีลักษณะที่สมมาตร เมื่อ NET = 0 จะทำให้สัญญาณ OUT เป็น 0 ดังได้แสดงไว้ในรูปที่ 4-7 ซึ่งจะมีจุดที่แตกต่างจากฟังก์ชันแบบโลจิสติกตรงที่ว่า ค่า OUT จะเป็นค่าที่เป็นไบโพลาร์ (Bipolar Value) ซึ่งคุณสมบัติตรงจุดนี้จะมีประโยชน์สำหรับการใช้โครงตาข่ายวงจรบางชนิด

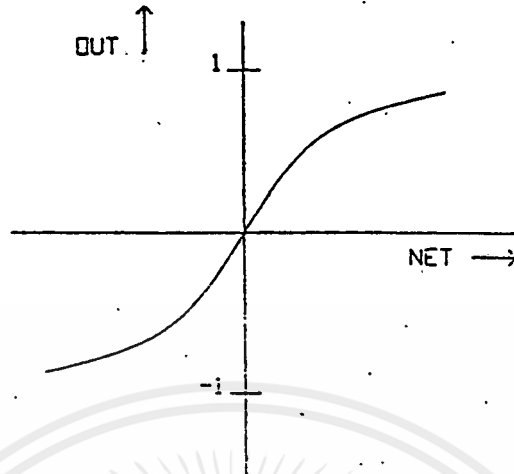
$$\text{Out} = 1/(1+\exp(-\text{net})) = F(\text{net})$$



รูปที่ 4-6 ฟังก์ชันแบบโลจิสติกซิกมอยด์

จากที่ได้กล่าวมาแบบจำลองของนิวรอลได้ขาดคุณสมบัติทางชีววิทยาบางจุด ยกตัวอย่างเช่น เราไม่ได้นำค่าของเวลา (Time Delay) ซึ่งมีผลต่อไดนามิกส์ของระบบมาประกอบการพิจารณาด้วย เพียงแต่พิจารณาว่าค่าอินพุตที่เราป้อนเข้าไปจะเกิดเอาท์พุตออกมาในทันที และที่สำคัญไปกว่านี้ เราไม่ได้รวมเอาผลการซินโครไนซ์ หรือ ฟังก์ชันการเปลี่ยนความถี่ของคลื่น (Frequency Modulation Function) ของนิวรอลในทางชีววิทยาซึ่งเป็นคุณสมบัติที่นักชีววิทยาบางคนให้ความสนใจมากมาพิจารณาด้วยเช่นกัน

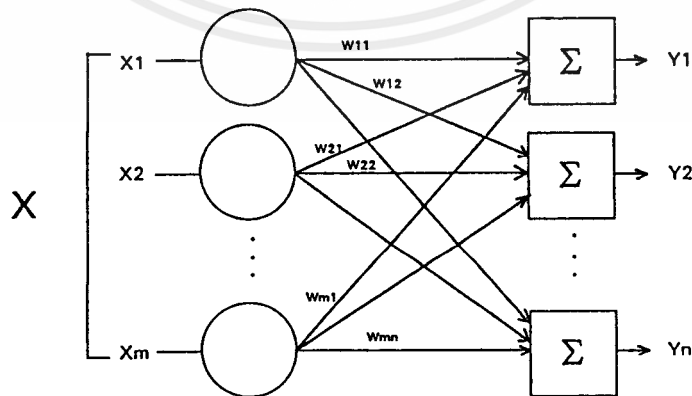
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-7 ฟังก์ชันไฮเปอร์โบลิกแทนเจนต์

4.3.4 โครงข่ายนิวรอลแบบเลเยอร์เดียว (Single Layer Artificial Neural Network)

แม้ว่านิวรอลเพียงหนึ่งตัวสามารถที่จะแสดงรูปแบบของฟังก์ชันบางชนิดได้ อย่างไรก็ตามความสามารถในการคำนวณนิวรอนได้มาจากการเชื่อมต่อหรือรวมกลุ่มของนิวรอนเข้าไปในโครงข่ายวงจรโดยโครงข่ายที่ง่ายที่สุดจะประกอบไปด้วยกลุ่มของนิวรอนที่มีการจัดเรียงตัวใน 1 เลเยอร์ โดยได้แสดงภาพไว้ด้านขวามือในภาพที่ ส่วนโหนดทรงกลมทางด้านซ้ายใช้สำหรับเป็นตัวกระจายอินพุตเท่านั้น ซึ่งส่วนนี้จะไม่มีการคำนวณใด ๆ ทั้งสิ้น จึงแสดงไว้ในรูปทรงกลมเพื่อให้ถึงความแตกต่าง กับส่วนนิวรอนทางขวามือ ซึ่งแสดงไว้เป็น รูปสี่เหลี่ยม



รูปที่ 4-8 โครงข่ายนิวรอลชนิดชั้นเดียว

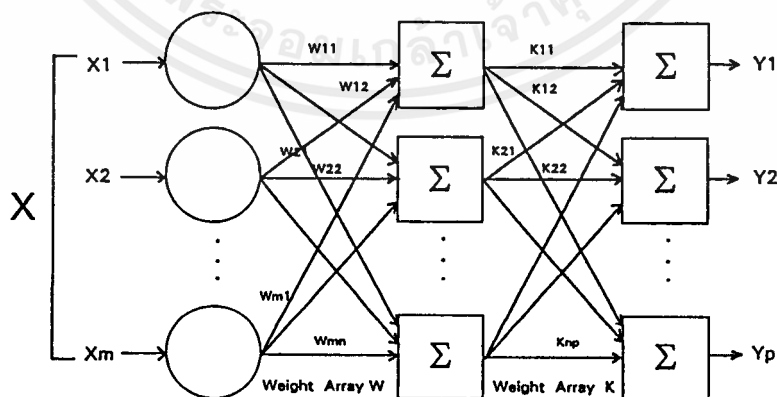
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มของอินพุต x จะมีการต่อเชื่อมกับนิวรอนโดยผ่านค่าน้ำหนัก ในช่วงแรก ๆ นั้นโครงตาข่ายนิวรอลไม่ได้มีความซับซ้อนอย่างที่เห็น ซึ่งนิวรอนแต่ละตัวจะให้ค่าเอาต์พุตที่เป็นค่ารวมของผลคูณของอินพุต และค่าน้ำหนัก

สำหรับโครงตาข่ายทางชีววิทยาที่แท้จริงจะประกอบไปด้วยการต่อเชื่อมที่มีความซับซ้อนน้อยกว่านี้ ซึ่งอาจจะมีการต่อเชื่อมระหว่างอินพุตและเอาต์พุตในเลเยอร์นั้น ๆ ในส่วนของค่าน้ำหนัก เราจะแสดงไว้ให้อยู่ในรูปของเมตริกซ์ W โดยมีขนาด m แถวและ n คอลัมน์ โดยค่า m คือจำนวนของอินพุต และ n คือจำนวนนิวรอนทั้งหมด ยกตัวอย่างเช่น ค่าน้ำหนักที่เชื่อมต่อกับอินพุตตัวที่ 3 กับนิวรอนตัวที่ 2 ก็สามารถเขียนด้วยสัญลักษณ์ W_{32} ในส่วนนี้เราจะเห็นว่าการคำนวณ กลุ่มของนิวรอน NET เอาต์พุต N กรณีที่เป็นหนึ่งชั้นนั้นเป็นการคูณเมตริกซ์อย่างง่าย ๆ โดยอาจเขียนว่า $N = XW$ โดยที่ N และ X เป็น row vector

4.3.5 โครงตาข่ายนิวรอลแบบหลายชั้น (Multi Layer Neural Network)

โครงตาข่ายวงจรแบบนี้จะมีขนาดใหญ่และมีความซับซ้อนกว่า และ ประสิทธิภาพในการคำนวณสูงกว่า แม้ว่าโครงตาข่ายจะถูกสร้างขึ้นตามแบบจำลองหลายแบบหรือแม้แต่การเรียงตัวของนิวรอน ในชั้นจะเลียนแบบมาจากโครงสร้างที่เป็นชั้นในบางส่วนของสมอง ซึ่งโครงตาข่ายวงจรที่มีหลายชั้นนี้ได้รับการพิสูจน์แล้วว่ามีประสิทธิภาพมากกว่าโครงตาข่ายแบบชั้นเดียวจากที่ได้กล่าวมาแล้วข้างต้น และเมื่อไม่นานมานี้ได้มีการพัฒนาขั้นตอนการฝึก (Train)



รูปที่ 4-9 ทฤษฎีโครงข่ายนิวรอลแบบ 2 ชั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงตาข่ายแบบนี้ ซึ่งโครงตาข่ายแบบนี้ (Multilayer Network) อาจจะได้จากการต่อเชื่อมแบบคาสเคด (Cascaded) กลุ่มของชั้นเดี่ยว ๆ อย่างซ้ำ ๆ โดยเอาท์พุทที่ได้จากชั้นหนึ่ง ๆ จะเป็นอินพุทป้อนให้กับเลเยอร์ในชั้นต่อไป ในรูปที่ 4-9 ได้แสดงถึงโครงตาข่ายแบบนี้และแสดงถึงการต่อเชื่อมไว้โดยละเอียด

4.3.6 ฟังก์ชันการกระตุ้นที่ไม่เป็นเชิงเส้น

(Nonlinear Activation Function)

โครงตาข่ายแบบหลายชั้น จะไม่ให้ประสิทธิภาพในการคำนวณได้สูงกว่าโครงตาข่ายแบบชั้นเดี่ยวถ้าไม่มีฟังก์ชันการกระตุ้นที่ไม่เป็นเชิงเส้น (Nonlinear Activation Function) ระหว่างชั้นของเลเยอร์ การคำนวณเอาท์พุทของชั้นหนึ่ง ๆ ซึ่งประกอบด้วยการคูณเวกเตอร์อินพุทกับเมตริกซ์ของค่าน้ำหนักในชั้นนั้น ๆ และ (ถ้าไม่มีฟังก์ชันการกระตุ้นที่ไม่เป็นเชิงเส้น) ก็จะคูณเวกเตอร์ผลลัพธ์ กับ เมตริกซ์ของน้ำหนักในชั้นที่สอง โดยสามารถแสดงสมการในเชิงคณิตศาสตร์ ได้เป็น

$$(XW_1) W_2$$

โดยใช้ความรู้ในเรื่องของการจัดกลุ่มทางคณิตศาสตร์เราสามารถเปลี่ยนรูปสมการได้เป็น

$$X(W_1W_2)$$

จากสมการเราจะเห็นได้ว่า โครงตาข่ายแบบเชิงเส้นขนาด 2 ชั้น (Two Layer Linear Network) จะมีความคล้ายคลึงกับโครงตาข่ายแบบชั้นเดี่ยว ซึ่งมีเมตริกซ์ของน้ำหนักเท่ากับผลคูณเมตริกซ์น้ำหนัก 2 เมตริกซ์ ดังนั้น โครงตาข่ายแบบเชิงเส้นที่เป็นหลายชั้นสามารถเขียนแทนด้วยโครงตาข่ายแบบชั้นเดี่ยวที่สอดคล้องกัน เนื่องจากโครงตาข่ายแบบชั้นเดี่ยวมีข้อจำกัดในเรื่องของประสิทธิภาพในการคำนวณ ดังนั้น ฟังก์ชันการกระตุ้นที่ไม่เป็นเชิงเส้นจึงมีความสำคัญในการเพิ่มประสิทธิภาพและความสามารถของระบบซึ่งมากกว่าประสิทธิภาพแบบชั้นเดี่ยว

4.3.7 โครงตาข่ายแบบรีเคอร์เรนท์ (Recurrent Network)

สำหรับโครงตาข่ายที่ได้กล่าวถึงตั้งแต่ต้นจนมาถึงจุดนี้ไม่มีการต่อเชื่อมแบบป้อนกลับ (Feedback Connection) ซึ่งก็คือการต่อเชื่อมเอาท์พุทของชั้นหนึ่งไปยังอินพุทของชั้นในชั้นเดียวกันหรือชั้นถัดมาโดยผ่านค่าน้ำหนัก โดยเราสามารถลักษณะโครงตาข่ายแบบนี้ว่า

โครงตาข่ายนอนรีเคอร์เรนต์ (Nonrecurrent Net.) ซึ่งในกรณีนี้จะไม่มีการใช้หน่วยความจำ โดยค่าเอาต์พุตที่ได้จะเกิดจากอินพุตและน้ำหนัก ในแบบจำลองบางแบบที่เป็นแบบนอนรีเคอร์เรนต์จะทำการป้อนสัญญาณเอาต์พุตกลับไปเป็นอินพุตใหม่ ดังนั้นค่าเอาต์พุตจะถูกกำหนดโดยค่าอินพุตปัจจุบันและเอาต์พุตก่อนหน้า ในลักษณะอย่างนี้เราจะเรียกว่าเป็นโครงตาข่ายแบบรีเคอร์เรนต์ (Recurrent Network) โดยมีลักษณะคุณสมบัติคล้ายคลึงกับหน่วยความจำระยะสั้น (Short-Term Memory)

ลักษณะการใช้ถ้อยคำ การใช้สัญลักษณ์ และการแสดงตัวอย่าง ของโครงตาข่ายนิเวรอล (Terminology, Notation and Representative of Artificial Neural Network)

เป็นที่น่าเสียดายที่ยังไม่มีข้อกำหนดในเรื่องของมาตรฐานต่าง ๆ ที่เกี่ยวกับข้อตกลงโดยทั่ว ๆ ไป เช่น เครื่องหมาย คำ สัญลักษณ์ รูปแบบการแสดงทางกราฟฟิก ของโครงตาข่ายนิเวรอลจึงจะทำให้แบบจำลองของโครงตาข่ายวงจรเดียวกันจะมีลักษณะที่แตกต่างกันเมื่อเขียนด้วยบุคคลต่าง ๆ กัน

4.3.8 ลักษณะการใช้ถ้อยคำ (Terminology)

ขั้นตอนการทำงานของโครงตาข่ายนิเวรอลโดยทั่ว ๆ ไป สามารถที่จะแสดงได้ในรูปของสมการดิฟเฟอเรนเชียล (Differential Equation) หรืออาจแสดงไว้ในรูปสมการดิฟเฟอเรนซ์ (Difference Equation) สำหรับการแสดงไว้ในรูปของดิฟเฟอเรนเชียล จะใช้สมมุติฐานที่ว่า กระบวนการทำงานเป็นขั้นตอนที่ต่อเนื่อง ทำงานคล้ายกับโครงตาข่ายแบบแอนนาลอก ขนาดใหญ่ สามารถเห็นระบบในทางชีววิทยาได้ในระดับ ไมโครสโคปิก (Microscopic Level) ซึ่งในทางปฏิบัติจริง ๆ แล้วไม่สามารถเป็นไปได้ ระดับการกระตุ้นของนิเวรอลในทางชีววิทยาจะถูกกำหนดโดยอัตราเฉลี่ยของการปล่อยพัลส์ไปตามแอกซอน (Axon) โดยอัตราเฉลี่ยนี้จะถูกทรีท (Treat) เสมือนเป็นปริมาณทางแอนนาลอก นอกจากนี้ถ้าหากต้องการจำลองโครงตาข่ายนิเวรอลบนคอมพิวเตอร์แอนนาลอก จะต้องแสดงด้วยสมการดิฟเฟอเรนเชียล แต่อย่างไรก็ตามส่วนมากในปัจจุบัน จะปฏิบัติบนดิจิทัลคอมพิวเตอร์ ดังนั้นการแสดงขั้นตอนการทำงานของโครงตาข่ายนิเวรอลให้อยู่ในรูปสมการดิฟเฟอเรนซ์ (Difference Equation) จะให้ความเหมาะสมกว่า เพราะสมการนี้สามารถเปลี่ยนไปเป็นโปรแกรมทางคอมพิวเตอร์ได้โดยง่าย

4.3.9 การแสดงตัวอย่าง (Representation)

มีข้อตกลงเพียงเล็กน้อยเกี่ยวกับวิธีการนับจำนวนของเลเยอร์ในโครงตาข่ายแบบหลายชั้น ในรูปที่ได้แสดงถึงโครงตาข่ายวงจรที่ประกอบด้วยกลุ่มของนิเวรอน (Neuron Set) และค่าน้ำหนักซึ่งได้มีการอธิบายถึงการต่อเชื่อมในรูปที่ จะไม่มีการรวมชั้นของอินพุต (No summation) โดยนิเวรอนเหล่านี้จะเป็นเสมือนจุดแฟนเอาต์ (Fan Out Point) และจะไม่

เกิดผลกระทบกับประสิทธิภาพและความสามารถในการคำนวณของโครงข่าย ด้วยเหตุนี้ เลเยอร์ในชั้นแรก ๆ จะไม่ถูกรวมเข้าไปสำหรับการนับจำนวนชั้น และโครงข่ายที่ได้แสดงไว้ในรูปที่ 1-6 เป็นโครงข่ายแบบ 2 ชั้นเนื่องจากมีการคำนวณใน 2 ชั้นเท่านั้น

4.3.10 การฝึกของโครงข่ายนิเวรอล

(Training of Artificial Neural Network)

การฝึกจะแสดงให้เห็นถึงความเข้าใจเบื้องต้นเกี่ยวกับกระบวนการเหล่านี้ แต่อย่างไรก็ตามการเรียนรู้เกี่ยวกับโครงข่ายนิเวรอลมีขีดจำกัด และยังมีปัญหาอีกหลายข้อที่จะต้องทำการแก้ไขก่อนที่จะเราจะบอกได้ว่าเราได้ศึกษามาตามขั้นตอนที่ถูกต้อง จุดประสงค์ของการฝึกโครงข่ายวงจรถูกเพื่อแสดงให้เห็นว่าการป้อนกลุ่มของอินพุตสามารถทำให้เกิดเอาต์พุตที่ต้องการ โดยเราจะทำการอ้างอินพุตและเอาต์พุตอยู่ในรูปของเวกเตอร์ สำหรับการฝึกสามารถทำได้โดยการป้อนเวกเตอร์อินพุตและจะต้องมีการปรับค่าน้ำหนักไปเรื่อย ๆ จนกว่าจะได้ค่าน้ำหนักของโครงข่ายวงจรถูกที่สามารถให้ค่าเอาต์พุตเวกเตอร์ที่ต้องการ

4.3.11 วิธีการฝึกแบบตรวจสอบ (Supervised Training)

ขั้นตอนการทำงานของวิธีการฝึกสามารถถูกแบ่งแยกออกเป็น การฝึกแบบตรวจสอบ (Supervised Training) และการฝึกแบบไม่มีการตรวจสอบ (Unsupervised Training) ซึ่งการฝึกในแบบแรก จะต้องมีการใช้คู่ของเวกเตอร์อินพุตและเวกเตอร์เป้าหมาย ซึ่งแสดงถึงค่าเอาต์พุตที่ต้องการโดยเรียกว่า คู่สำหรับการฝึก (Training Pair) เมื่อทำการป้อนเวกเตอร์ อินพุตจะมีการคำนวณค่าเอาต์พุตของโครงข่ายและจะนำค่าดังกล่าวไปเปรียบเทียบกับเวกเตอร์เป้าหมาย (Target Vector) ซึ่งจะได้ค่าความคลาดเคลื่อนออกมา และค่าความคลาดเคลื่อนดังกล่าวจะถูกป้อนกลับเข้าไปในวงจรเพื่อทำการปรับค่าน้ำหนักที่เหมาะสมตามขั้นตอนการทำงาน โดยมีจุดประสงค์เพื่อลดค่าความคลาดเคลื่อนให้น้อยที่สุด เวกเตอร์ของกลุ่มที่ใช้ฝึก (Vector Of Training Set) จะถูกป้อนเข้าเรื่อย ๆ และจะมีการคำนวณค่าความคลาดเคลื่อนพร้อมทั้งการปรับน้ำหนักให้ได้ค่าที่เหมาะสมจนกว่าค่าความคลาดเคลื่อนของกลุ่มที่ใช้ฝึกทั้งหมดอยู่ในระดับต่ำพอที่จะยอมรับได้

4.3.12 การฝึกแบบไม่มีการตรวจสอบ (Unsupervised Training)

แม้ว่าวิธีการข้างต้นสามารถนำมาใช้ได้เป็นอย่างดีแต่ได้ถูกวิจารณ์ว่าเป็นไปไม่ได้และขาดเหตุผลทางชีววิทยา โดยเป็นการยากที่นักภาพถึงการเปลี่ยนแปลงระหว่างเอาต์พุตที่คำนวณได้และเอาต์พุตที่ต้องการภายในสมอง เราจะสามารถตั้งเอาต์พุตที่ต้องการได้อย่างไร จึงเป็นคำถามที่เกิดขึ้นจึงได้เกิดวิธีการฝึกอีกวิธีหนึ่งที่เรียกว่าการฝึกแบบไม่มีการตรวจสอบ

จึงเป็นคำถามที่เกิดขึ้นจึงได้เกิดวิธีการฝึกอีกวิธีหนึ่งที่เรียกว่าการฝึกแบบไม่มีการตรวจสอบ (Unsupervised Training) ซึ่งสามารถเรียนระบบในทางชีววิทยาได้ โดยได้ถูกค้นคว้าและพัฒนาโดยโคโฮเนน (Cohonen) ในปี 1984 เป็นวิธีการที่ไม่จำเป็นต้องใช้เวกเตอร์เป้าหมายสำหรับเปรียบเทียบค่าเอาต์พุต จึงไม่จำเป็นต้องมีการเปรียบเทียบค่าเอาต์พุตกับค่าเอาต์พุตที่ต้องการ โดยกลุ่มสำหรับการฝึกจะประกอบด้วยเวกเตอร์อินพุตเท่านั้น และขั้นตอนการทำงานของ การฝึกจะทำการปรับค่าน้ำหนักของโครงข่ายวงจรเพื่อสร้างค่าเวกเตอร์เอาต์พุตที่สอดคล้องกันและค่อนข้างคงที่ กระบวนการดังกล่าวข้างต้นจะดึงเอาคุณสมบัติทางสถิติของกลุ่มการฝึกและกลุ่มของเวกเตอร์ที่คล้ายกันเข้าไปในคลาส ซึ่งการป้อนเวกเตอร์จากคลาสที่ให้มาไปยังอินพุตจะทำให้ได้เอาต์พุตเวกเตอร์ที่กำหนด แต่ในทางปฏิบัติจริง ๆ เป็นไปไม่ได้ที่จะกำหนดรูปแบบของเอาต์พุตก่อนการฝึกเพราะค่าเอาต์พุตจะถูกสร้างจากคลาสของเวกเตอร์อินพุตที่ให้มา

4.3.13 ขั้นตอนการทำงานของ การฝึก (Train Algorithms)

ขั้นตอนการทำงานของ การฝึกที่ใช้กันอยู่ในปัจจุบันได้พัฒนามาจากหลักการของ ฮอป (D.O Hebb) ใน ปี 1961 ซึ่งได้เสนอแบบจำลองสำหรับการเรียนแบบไม่ตรวจสอบ โดยค่าน้ำหนักหรือความเครียดซินแนปติกส์ (Synaptic Strength) จะถูกเพิ่มค่าถ้านิวรอนที่หมาย (Destination Neuron) และนิวรอนจากแหล่งเริ่มต้น (Source Neuron) ถูกกระตุ้น

สำหรับโครงข่ายนิวรอลที่ใช้พื้นฐานการเรียนแบบเฮบบเบียน (Hebbian Learning) จะเพิ่มค่าน้ำหนักตามระดับการกระตุ้นของนิวรอนที่หมายและนิวรอนเริ่มต้น โดยสามารถเขียนสมการได้ดังต่อไปนี้

$$W_{ij} (n+1) = W_{ij} (n) + \alpha OUT_i OUT_j$$

โดยที่

W_{ij} คือค่าของน้ำหนักจากนิวรอน i ไปยังนิวรอน j ก่อนที่จะทำการปรับ

$W_{ij}(n+1)$ คือค่าของน้ำหนักจากนิวรอน i ไปยังนิวรอน j หลังจากการปรับ

α คือสัมประสิทธิ์ของอัตราการเรียน (Learning Rate Coefficient)

OUT_i คือเอาต์พุตของนิวรอน i ซึ่งเป็นอินพุตของนิวรอน j

OUT_j คือเอาต์พุตของนิวรอน j

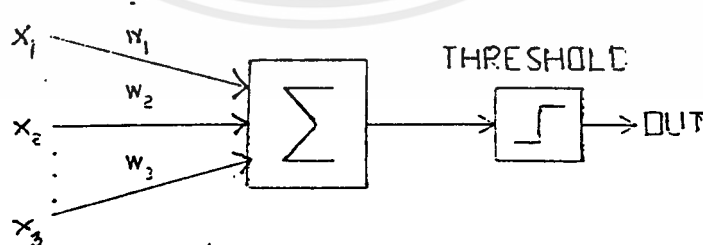
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 เปอร์เซ็ปตรอน (Perceptron)

4.4.1 เปอร์เซ็ปตรอน และจุดเริ่มต้นของ ANN

การศึกษาทางวิทยาศาสตร์เกี่ยวกับ ANN ได้เริ่มปรากฏชัดในช่วงทศวรรษที่ 1940 นักวิจัยที่ต้องการแบ่งการทำงานของระบบของมนุษย์เป็น 2 ส่วน ได้พัฒนากระด้าง ภัณฑ์ (Hardware) และละมุนภัณฑ์ (Software) ของแบบจำลองนิวรอนทางชีววิทยา และระบบการเชื่อมต่อกันของนิวรอน และนิวโรไซซิโอลอจิส (Neurophysiologist) ได้ให้ความสนใจในระบบนิวรอลของมนุษย์ (Human Neuron System) เพิ่มขึ้น ซึ่งทำให้ได้ผลลัพธ์ที่น่าพอใจ สนับสนุนการค้นคว้าและทำให้เกิดโครงข่ายวงจรที่ได้รับการพัฒนามากขึ้น

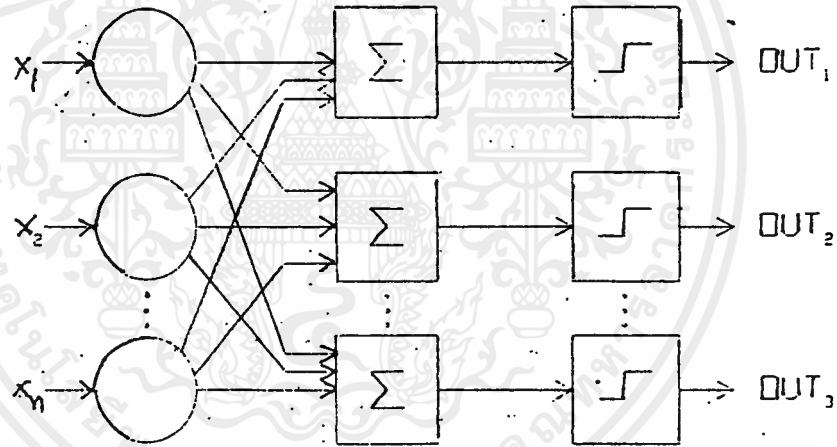
McCulloch และ Pitts (1943) ได้จัดพิมพ์งานเกี่ยวกับโครงตาข่ายวงจรนิวรอน แต่ในปี 1947 McCulloch และ Pitts ได้ค้นคว้ารูปแบบของโครงตาข่ายสำหรับการจำรูปแบบ (Pattern Recognition) งานส่วนใหญ่ของทั้งสองได้รวมไปกับการออกแบบ แบบจำลองนิวรอลอย่างง่าย ๆ ดังแสดงไว้ในรูป 4-10 ส่วน Σ จะเป็นตัวคูณค่าระหว่างอินพุต X กับ ค่าน้ำหนักของตัวมัน และจะบอกอินพุตน้ำหนักมากกว่าทั้งหมดเข้าด้วยกัน ในกรณีที่ผลบวกของอินพุตน้ำหนักมากกว่าทริสโฮล ค่าเอาต์พุตจะเป็น 1 แต่ถ้าน้อยกว่าทริสโฮล ค่าเอาต์พุตจะเป็น 0 เราจะเรียกระบบเหล่านี้ว่า เปอร์เซ็ปตรอน โดยทั่วไปแล้วระบบจะประกอบด้วยนิวรอนแบบหนึ่งชั้น (Single layer of artificial neurons) ต่อเชื่อมกับน้ำหนักของกลุ่มของอินพุต (ดูรูปที่ 4-10) หรือแม้แต่ในกรณีที่เป็หลายชั้น และมีความซับซ้อนมากกว่านี้ ก็จะใช้ชื่อเดียวกัน



รูปที่ 4-10 แสดงนิวรอลเปอร์เซ็ปตรอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในช่วงปี 1960 นักวิจัยได้ให้ความสนใจอย่างมากกับเปอร์เซ็ปตรอน งานวิจัยของ Rosenblatt (1962) ได้พิสูจน์ทฤษฎีเกี่ยวกับการเรียนรู้เปอร์เซ็ปตรอน (Perceptron Learning) Widrow (1961, 1963), Widrow และ Angell (1962), Widrow และ Hoff (1960) ได้แสดงให้เห็นว่าระบบที่มีลักษณะคล้ายเปอร์เซ็ปตรอนที่น่าเชื่อถือ และทำให้นักวิชาการมีความกระตือรือร้นที่จะศึกษาและค้นคว้าถึงศักยภาพของระบบเหล่านี้ ซึ่งต่อมาได้เกิดการค้นพบว่า เปอร์เซ็ปตรอนไม่สามารถใช้กับงานการเรียนรู้ (Learning Task) ง่ายๆ บางอย่าง ได้โดย Minsky และ Papert (1969) ได้วิเคราะห์ว่าปัญหานี้เกิดเนื่องมาจากมีข้อจำกัดในกรณีที่เป็นเปอร์เซ็ปตรอนชั้นเดียว เพราะว่าในสมัยนั้นความรู้ในการฝึกหัดแบบใช้การเรียนรู้โครงข่ายวงจรแบบหลายชั้นยังไม่มี



รูปที่ 4-11 แสดงเปอร์เซ็ปตรอนที่มีเอาต์พุตหลายค่า

นักวิจัยได้พยายามค้นคว้าโครงข่ายนิเวรอล ต่อมาไม่นานได้ค้นพบเทคนิคการฝึกหัดสำหรับโครงข่ายนิเวรอลแบบหลายชั้น โดยผลักดันให้เกิดความสนใจ และเกิดการค้นคว้าเพิ่มเติมต่อไป ทั้งๆ ที่มีข้อจำกัดในกรณีของเปอร์เซ็ปตรอน แต่ก็ยังได้มีการค้นคว้าเพิ่มเติมซึ่งทฤษฎีของเปอร์เซ็ปตรอนได้ใช้เป็นรากฐานสำหรับโครงข่ายนิเวรอลอื่นๆ และทฤษฎีเหล่านี้ได้ก่อให้เกิดหลักการที่สำคัญๆ มากมาย ด้วยเหตุผลนี้จึงเป็นจุดเริ่มต้นที่ดีสำหรับการศึกษาโครงข่ายนิเวรอลต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.2 การนำเสนอเปอร์เซ็ปตรอน (Perceptron Representation)

จากทฤษฎีการเลียนแบบเปอร์เซ็ปตรอนของ Rosenblatt 1962 ได้แสดงให้เห็นว่า เปอร์เซ็ปตรอนสามารถใช้เรียนในสิ่งที่ตัวมันมีความจำเป็นที่จะต้องแยกความแตกต่างระหว่างการแสดงและการเรียน โดยการแสดงหมายถึง ความสามารถของเปอร์เซ็ปตรอน (หรือโครงข่ายวงจรอื่น ๆ) ที่จะจำลองฟังก์ชันที่กำหนด ส่วนการเรียนจำเป็นต้องมีขั้นตอนของระบบ สำหรับใช้ในการปรับน้ำหนักของโครงข่ายวงจรเพื่อสร้างฟังก์ชันนั้น

เพื่อที่จะแสดงให้เห็นถึงปัญหาของการแสดง สมมติว่า เรามีกลุ่มของแฟลชการ์ด (Flash card) มีตัวเลข 0-9 และสมมติว่ามีเครื่องจักร (Hypertherical machine) ซึ่งสามารถที่จะแยกแยะการ์ด ที่เป็นเลขคู่กับเลขคี่ ซึ่งได้แสดงระบบไว้ในรูปที่ 4-12 จึงมีคำถามที่ว่า เครื่องจักรตัวนี้สามารถใช้แทนด้วยเปอร์เซ็ปตรอนซึ่งใช้การปรับน้ำหนักได้หรือไม่ ซึ่งเราสามารถใช้อเปอร์เซ็ปตรอนแทนเครื่องจักรดังกล่าวได้ และเราจะเห็นว่าเปอร์เซ็ปตรอนแบบขั้นเดียวมีข้อจำกัดของประสิทธิภาพในการแสดง และมีเครื่องจักรหลายแบบที่ไม่สามารถแสดงได้ ไม่ว่าจะใช้วิธีการปรับน้ำหนักอย่างไร

4.4.3 การเรียนเปอร์เซ็ปตรอน (Perceptron Learning)

คุณสมบัติที่ดีเด่นของโครงข่ายนิวรอนก็คือความสามารถในการเรียน (Learning Capability) ซึ่งเหมือนกับระบบในทางชีววิทยา โครงข่ายจะปรับตัวเองเพื่อสร้างรูปแบบทางพฤติกรรมที่ต้องการ

โดยการใช้การเกณฑ์ชนิดแยกออกเชิงเส้น (Linear-Separability Criterion) จึงเป็นไปได้ที่ว่าโครงข่ายแบบเลเยอร์เดียวสามารถที่จะแสดงฟังก์ชันที่ต้องการได้แต่ก็ยังไม่มียประสิทธิภาพเพียงพอเพราะเราไม่สามารถหาค่าที่ต้องการสำหรับค่าน้ำหนักและทริสโพล ได้ จึงจำเป็นต้องใช้วิธีการทางระบบ (Systematic Methods) หรือขั้นตอนการทำงานสำหรับการคำนวณค่าซึ่งโรเซนบลัท (Rosenblatt, 1962) ได้แสดงไว้ในขั้นตอนการฝึก นอกจากนี้เปอร์เซ็ปตรอนสามารถใช้ฝึกทุกฟังก์ชันที่มันสามารถแทนได้

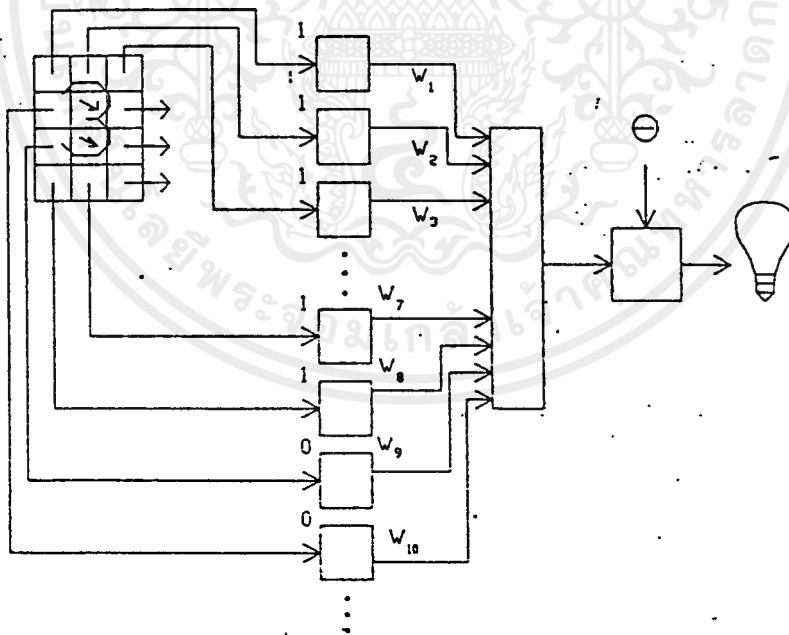
สำหรับการเรียนนั้นเราสามารถใช้อทั้งวิธีการที่เป็นแบบตรวจสอบ (Supervise) และไม่ตรวจสอบ (Unsupervise) การเรียนแบบตรวจสอบจำเป็นต้องใช้ "ครูจากภายนอก" (External Teacher) ซึ่งจะทำการคำนวณค่าพฤติกรรมของระบบ ส่วนการเรียนแบบไม่ตรวจสอบซึ่งได้มีการกล่าวไว้ในตอนแรก ไม่จำเป็นต้องมีครู โครงข่ายวงจรจะจัดระบบในตัวเองเพื่อทำให้เกิดการเปลี่ยนแปลงที่ต้องการ สำหรับการเรียนเปอร์เซ็ปตรอนจะเป็นการเรียนแบบตรวจสอบ

ขั้นตอนการทำงานของเปอร์เซ็ปตรอนสามารถทำได้โดยใช้เครื่องคอมพิวเตอร์หรือ กระด้างภัณฑ์ (Hardware) ทางอิเล็กทรอนิกส์อื่น ๆ และทางโครงข่ายวงจรจะสามารถปรับได้

ด้วยตัวมันเอง (Self-adjusting) ด้วยเหตุนี้เราจะเรียกรวธีการปรับค่านำหนักว่าการฝึกและสำหรับในส่วนของโครงตาข่ายว่า "การเรียนรู้"

4.4.4 ขั้นตอนการฝึกเปอร์เซ็ปตรอน (Perceptrons Training Algorithm)

เปอร์เซ็ปตรอนจะถูกฝึกโดยการเสนอกลุ่มของรูปแบบให้กับอินพุทของมันและจะทำการปรับน้ำหนักจนกว่าจะได้ค่าเอาต์พุทที่ต้องการสำหรับแต่ละอินพุทที่ต้องการ สมมติว่ารูปแบบอินพุทอยู่บนแฟลชการ์ด แฟลชการ์ดแต่ละแผ่นจะแทนด้วยสัญลักษณ์ที่เป็นสี่เหลี่ยมซึ่งจะให้ค่าอินพุทไปยังเปอร์เซ็ปตรอน ในกรณีที่มีเส้นผ่านเข้าไปในสี่เหลี่ยมค่าเอาต์พุทของมันก็จะเป็น 1 ถ้าไม่มีเส้นผ่านเข้าไป ค่าเอาต์พุทก็จะเป็นศูนย์ กลุ่มของสี่เหลี่ยมดังกล่าวจะแสดงถึงกลุ่มของ 1 และ 0 ซึ่งหมายถึงอินพุทที่ป้อนให้กับเปอร์เซ็ปตรอน จุดประสงค์ที่ต้องทำการฝึกหัดเปอร์เซ็ปตรอนก็เพื่อว่า เมื่อทำการป้อนกลุ่มของอินพุทที่แสดงถึงตัวเลขที่เป็นเลขคี่ จะทำให้หลอดไฟติด และถ้าตัวเลขเป็นเลขคู่ไฟจะดับ



รูปที่ 4-12 การทราบภาพพจน์เปอร์เซ็ปตรอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 4-12 ได้แสดงถึงองค์ประกอบของเปอร์เซ็ปตรอน สมมติว่าเวกเตอร์ X แสดง ถึงรูปแบบของแฟลชการ์ด สำหรับสี่เหลี่ยม หรือส่วนประกอบของเวกเตอร์ของน้ำหนักที่สอดคล้องกันซึ่งก็คือ W_1, W_2, W_3, \dots ผลลัพธ์ที่ได้จากข้างต้นจะนำมาบวกกัน ถ้าผลบวกที่ได้เกินกว่าค่า ทริสโฮล Threshold เอาท์พุทของนิวรอน Y จะเป็น 1 (และไฟจะเปิด) นอกจากนี้ ไฟจะดับ (เอาท์พุทเป็น 0) ซึ่งสามารถเขียนขั้นตอนดังกล่าวในรูปของสมการเป็น $Y = XW$ ตามด้วยขั้นตอนกระบวนการทริสโฮล

เพื่อทำการฝึกหัดโครงตาข่าย รูปแบบ X จะถูกป้อนเข้าไปที่อินพุท และจะมีการคำนวณเอาท์พุท ถ้า Y ถูกต้อง จะไม่มีการเปลี่ยนแปลงใดๆ แต่ถ้าเอาท์พุท Y ผิด จะต้องเกิดการปรับค่าน้ำหนักที่เชื่อมต่อกับอินพุทเพื่อให้ความผิดพลาดลดลง

เพื่อให้เห็นวิธีการเหล่านี้โดยละเอียด เราจะสมมติว่าให้แฟลชการ์ดที่เขียนหมายเลข 3 ถูกป้อนเข้าไปในระบบ และเมื่อคำนวณค่าเอาท์พุทได้เป็น 1 (เป็นเลขคี่) ซึ่งก็คือผลลัพธ์ที่ถูกต้อง จึงไม่ต้องทำการปรับน้ำหนัก แต่ถ้าในกรณีของแฟลชการ์ดหมายเลข 4 ถูกป้อนเข้าไปในระบบเปอร์เซ็ปตรอน และได้เอาท์พุทออกมาเป็น 1 (เป็นเลขคี่) จะต้องมีการปรับค่าน้ำหนักที่เชื่อมต่อกับอินพุท เพื่อให้เอาท์พุทลดลงจาก 1 ไปเป็น 0 ในทำนองเดียวกัน ถ้าในกรณีของการ์ดหมายเลข 3 เข้าไปในระบบ แต่เอาท์พุทที่ได้เป็น 0 จึงต้องทำการปรับน้ำหนักเพื่อให้ค่าเอาท์พุทเพิ่มขึ้นเป็น 1

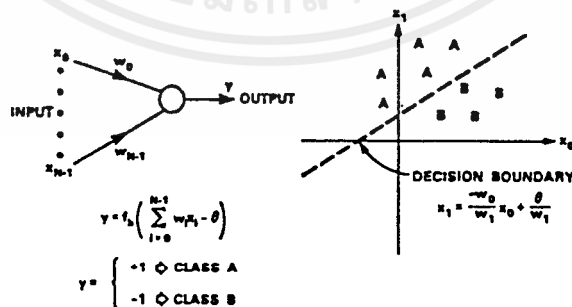
เราสามารถสรุปวิธีการฝึกหัดได้ดังนี้

1. ป้อนรูปแบบอินพุท และทำการคำนวณเอาท์พุท Y
2. a. ถ้าเอาท์พุทที่ได้ถูกต้องให้ทำ ข้อ 1
- b. ถ้าเอาท์พุทที่ได้ไม่ถูกต้อง และเป็น 0 ต้องเพิ่มน้ำหนัก
- c. ถ้าเอาท์พุทที่ได้ไม่ถูกต้อง และเป็น 1 ต้องลดน้ำหนัก
3. ไปทำข้อ 1

จากขั้นตอนทั้ง 3 โครงตาข่ายวงจรจะทำการเรียนเพื่อแยกการ์ดไปเป็นกลุ่มที่เป็นเลขคู่ และกลุ่มที่เป็นเลขคี่ ซึ่งก็คือ สำหรับการ์ดที่เป็นเลขคี่ ค่าเอาท์พุทสูงกว่าค่าทริสโฮล และสำหรับการ์ดที่เป็นเลขคู่ ค่าเอาท์พุทจะต่ำกว่าค่าทริสโฮล ซึ่งการเรียนที่กล่าวมานี้เป็นการเรียนด้านอินพุท ซึ่งจะทำการเรียนกลุ่มของการ์ดทั้งหมด ทำให้เกิดคำถามที่ว่าเราจะเลือกการ์ดเป็นชั้นๆ หรือเป็นแบบสุ่ม ซึ่งมีทฤษฎีที่สามารถให้คำตอบนี้ได้

4.4.5 ชิงเกิลเลเยอร์เปอร์เซปตรอน

ชิงเกิลเลเยอร์เปอร์เซปตรอน คือโครงข่ายแรกที่สามารถใช้ได้ทั้งค่าต่อเนื่องและค่าที่เป็นฐานสอง โครงข่ายแบบนี้ ผลักดันให้เกิดความสนใจ และการพัฒนาเพิ่มขึ้น เพราะมีประสิทธิภาพในการใช้เรียนรูแบบง่าย ๆ สำหรับในรูปที่ 4-13 ได้แสดงให้เห็นว่าเปอร์เซปตรอน จะเป็นตัวตัดสินใจว่า อินพุตจะมาจาก 1 หรือ 2 คลาส (Class) (A หรือ B) ในส่วนของโนดเดี่ยว (Single Node) จะเป็นตัวคำนวณผลรวมของน้ำหนักของอินพุต ลบด้วยค่า ทริสโฮล (Threshold) และทำการส่งผลลัพธ์ผ่านขอบเขตที่ไม่เป็นเชิงเส้น (limiting nonlinearity) และเอาท์พุท y ที่ออกมาจะเป็น +1 หรือ -1 ซึ่งถ้าเอาท์พุทเป็น 1 ก็จะเป็นคลาส A และถ้าเป็น -1 ก็จะเป็นคลาส B ซึ่งเป็นเทคนิคที่มีประโยชน์สำหรับการวิเคราะห์พฤติกรรมของโครงข่ายอย่างเช่น เปอร์เซปตรอนนี้ ก็เพื่อที่จะทำการพล็อตแผนภาพของบริเวณการตัดสินใจ (Decision region) ที่ถูกสร้างให้เป็นแบบภาพหลายมิติ ซึ่งโดยแทนเป็นตัวแปรอินพุต บริเวณการตัดสินใจนี้จะกำหนดว่าค่าของอินพุตตัวไหนที่ทำให้เกิดการตอบสนองในคลาส A และถ้าอินพุตที่ทำให้การตอบสนองในคลาส B เปอร์เซปตรอนจะสร้างบริเวณการตัดสินใจ 2 พื้นที่โดยมี ไฮเปอร์เพลน (Hyperplane) เป็นตัวแยก บริเวณเหล่านี้ได้แสดงไว้ในด้านขวามือของรูปที่ 4-13 เมื่อมีอินพุต 2 ชนิด คือ X_1 และ X_2 และ ไฮเปอร์เพลน เป็นเส้นตรง ในกรณีนี้อินพุตที่อยู่เหนือเส้นอาณาเขต (Boundary Line) จะนำไปสู่การตอบสนองในคลาส A และอินพุตที่อยู่ด้านล่างเส้นดังกล่าวจะนำไปสู่การตอบสนองในคลาส B โดยสมการของเส้นอาณาเขตจะขึ้นอยู่กับน้ำหนัก และทริสโฮล



รูปที่ 4-13 ชิงเกิลเลเยอร์เปอร์เซปตรอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

น้ำหนัก และทริสโฮลในเปอร์เซ็ปตรอนหนึ่งๆ สามารถที่จะถูกกำหนดไว้ หรือดัดแปลงโดยใช้ขั้นตอนการทำงานที่แตกต่างกัน ขั้นตอนการลู่เข้าของเปอร์เซ็ปตรอน (Perceptron Convergence Procedure) สำหรับการปรับค่าน้ำหนักได้ถูกพัฒนาโดย Rosenbalt ได้อธิบายไว้ใน กล่อง ถ้าน้ำหนักค่าแรก และทริสโฮล จะถูกกำหนดให้มีค่าเริ่มต้น เป็นค่าน้อยๆ ที่ไม่เท่ากับ 0 (Non-Zero Values) จากนั้นอินพุตใหม่ซึ่งประกอบด้วยค่าต่อเนื่อง N จะถูกป้อนเข้าไปเพื่อทำการคำนวณเอาต์พุตเหมือนในรูปที่ 4-13 ค่าน้ำหนักจะถูกปรับเมื่อเกิดค่าคลาดเคลื่อนโดยใช้สมการในขั้นตอนที่ 4 ของ กล่องรูป 14 ซึ่งสูตรดังกล่าว ประกอบด้วยอัตราขยาย η (0.0 - 1.0) ซึ่งจะทำให้การอัตราควบคุมการปรับ (Adaptation Rate) อัตราขยายนี้จะต้องถูกปรับเพื่อทำให้เกิดการประมาณค่าน้ำหนักที่แน่นอน

กล่อง ขั้นตอนการทำงานการลู่เข้าของเปอร์เซ็ปตรอน

ขั้นที่ 1. เลือกน้ำหนัก และทริสโฮล (Threshold) เริ่มต้น ตั้ง $w_i(0)$ ($0 \leq i \leq N-1$) และ θ ด้วยการสุ่มค่าน้อยๆ ซึ่งเป็นน้ำหนักจากอินพุต i ที่เวลา t และ θ คือทริสโฮลในโนดเอาต์พุต

ขั้นที่ 2. ป้อนอินพุตใหม่ และเอาต์พุตที่ต้องการ ป้อนค่าต่อเนื่องใหม่ เอาต์พุตใหม่ไปเรื่อยๆ

ขั้นที่ 3. คำนวณเอาต์พุตที่แท้จริง

$$y(t) = f_h \left(\sum_{i=0}^{n-1} w_i(t) x_i(t) - \theta \right)$$

ขั้นที่ 4. ปรับน้ำหนัก

$$w_i(t+1) = w_i(t) + \eta [d(t) - y(t)] x_i(t),$$

$$0 \leq i \leq N-1$$

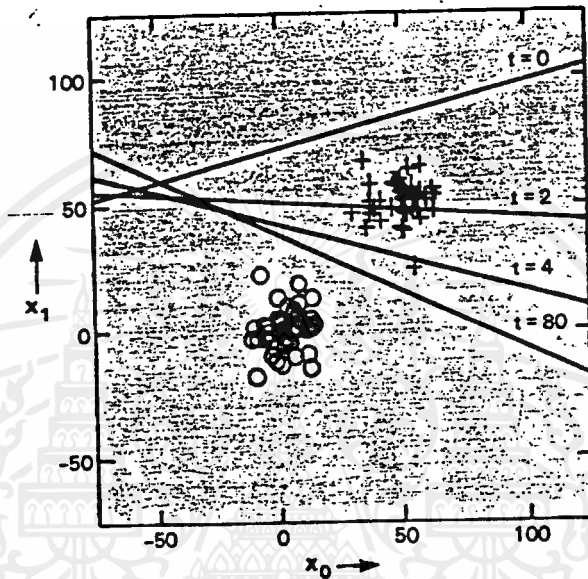
$d(t) = +1$ ถ้า อินพุตมาจากคลาส A
 $d(t) = -1$ ถ้า อินพุตมาจากคลาส B
 η คือค่าคงที่บวกที่น้อยกว่า 1

ขั้นที่ 5. กลับไปทำขั้นที่ 2

รูปที่ 4-14 แสดงกล่องการทำงานของเปอร์เซ็ปตรอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับตัวอย่างของการใช้ขั้นตอนการลู่เข้าของเปอร์เซ็ปตรอนได้แสดงไว้ในรูปที่ 4-15 โดยวงกลมแสดงค่าตัวอย่าง (Example Value) จากคลาส A และกากบาทแสดงตัวอย่าง จากคลาส B เส้นทั้ง 4 เส้นได้แสดงถึงขอบเขตการตัดสินใจ (Four Decision Boundaries) หลังจากที่ได้ทำการปรับน้ำหนักตามค่าผิดพลาด เมื่อ $t = 0, 2, 4, 80$ จากตัวลู่เข้าอัตรา การขยายเท่ากับ 0.01



รูปที่ 4-15 ตัวอย่างของเขตการตัดสินใจ ซึ่งถูกสร้างมาจากการลู่เข้าของเปอร์เซ็ปตรอนที่ใช้ 2 คลาส ข้อมูลที่สุ่มจากคลาส A จะแสดงด้วย กากบาท เส้นตรงที่เห็นแสดงถึงของเขตการตัดสินใจ หลังจากที่ได้ค่าความคลาดเคลื่อน และน้ำหนักถูกปรับแล้ว

Rosenblatt ได้พิสูจน์ว่าถ้าอินพุตที่ถูกแสดงทั้งสองชนิดแยกจากกันได้ จะทำให้ขั้นตอนการลู่เข้าของเปอร์เซ็ปตรอนจะลู่เข้า และจะวางตำแหน่งไฮเปอร์เพลนระหว่างคลาสทั้งสองนี้ ซึ่งไฮเปอร์เพลนดังกล่าวได้ถูกแสดงไว้ด้านขวามือรูปที่ 4-13 ขอบเขตการตัดสินใจนี้จะแยกตัวอย่างจาก คลาส A และ B ซึ่งวิธีการดังกล่าวเกิดปัญหาขึ้นตรงที่อาณาเขตการตัดสินใจ อาจเกิดการเหลื่อมกันจากการกระจาย จึงได้ดัดแปลงพัฒนาวิธีการดังกล่าวใหม่ซึ่งจะช่วยลดค่าผิดพลาดระหว่างเอาท์พุตที่ต้องการของโครงตาข่ายที่คล้ายเปอร์เซ็ปตรอนกับ เอาท์พุตจริง ซึ่งเรียกว่า Widrow-Hoff หรือ LMS Algorithms

โดยขั้นตอนการทำงาน LMS จะคล้ายกับในขั้นตอนการลู่เข้าของเปอร์เซ็ปตรอน ซึ่งได้ อธิบายไว้ในรูปที่ 14 แต่ขอบเขตความไม่เป็นเชิงเส้นจำกัด (Limiting Nonlinearity) ได้ถูกทำให้เป็นเชิงเส้น หรือถูกแทนที่โดยทริสโฮลลอจิก (Threshold-logic Nonlinearity) คำนำ-

หนักจะถูกปรับในทุกการทดลอง (Trial) โดยขึ้นอยู่กับความแตกต่างระหว่างอินพุตที่ต้องการและอินพุตจริง

STRUCTURE	TYPES OF DECISION REGIONS	EXCLUSIVE OR PROBLEM	CLASSES WITH MESHED REGIONS	MOST GENERAL REGION SHAPES
SINGLE-LAYER	HALF PLANE BOUNDED BY HYPERPLANE			
TWO-LAYER	CONVEX OPEN OR CLOSED REGIONS			
THREE-LAYER	ARBITRARY (Complexity Limited By Number of Model)			

รูปที่ 4-16 ชนิดของบริเวณการตัดสินใจซึ่งสามารถสร้างจากเปอร์เซ็ปตรอน แบ่งชั้นเดียว หรือชั้นซับซ้อน โดยมี 2 อินพุต และชั้นภายใน 1 และ 2 ชั้น บริเวณที่แรเงาหมายถึง บริเวณการตัดสินใจของคลาส A โหนดในโครงตาข่ายใช้แบบความไม่เป็นเชิงเส้นแบบจำกัดสูง (Hard limiting nonlinearities)

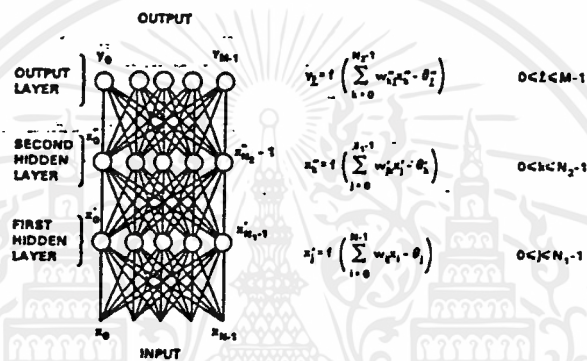
ตัวจำแนกประเภท (Classifier) ที่ใช้ขั้นตอนการทำงานของกรฝึกแบบ LMS สามารถที่จะใช้เอาต์พุตที่ต้องการเป็น 1 ในกรณีที่เป็นคลาส A และ 0 ในกรณีคลาส B ในระหว่างการปฏิบัติ ค่าอินพุตจะถือเป็นคลาส A ถ้าเอาต์พุตมีค่ามากกว่า 0.5

4.4.6 มัลติเลเยอร์เปอร์เซ็ปตรอน (Multilayer Perceptron)

มัลติเลเยอร์ คือ โครงตาข่ายเดินทางหน้าซึ่งมีโน้ตระหว่างอินพุต และเอาต์พุต มากกว่า 1 ชั้นไป ยังมีชั้นภายในมากขึ้นเท่าไร ระบบก็จะมีขอบเขตกว้างมากขึ้นเท่านั้น แต่การพัฒนาที่จะยุ่งยากและช้าลง ถึงแม้ว่ามันยังพิสูจน์ว่าขั้นตอนการทำงานจะลู่เข้า (Converge) เจกเช่น ซิงเกิลเลเยอร์เปอร์เซ็ปตรอน แต่มัลติเลเยอร์ก็แสดงถึงความสำเร็จของการแก้ปัญหาหลายปัญหาลงได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสามารถของมัลติเลเยอร์เพอร์เซ็ปตรอนจากที่ไม่เป็นเชิงเส้นจะถูกใช้เข้าไปใน โหนด ถ้าโหนดเป็นอุปกรณ์เชิงเส้นแล้ว โครงตาข่ายธรรมดาซึ่งมีการเลือกน้ำหนักที่เหมาะสม สามารถเทียบเท่ากับรูปแบบการคำนวณโดยโครงตาข่ายซับซ้อนได้เป็นอย่างดี ความสามารถของ เพอร์เซ็ปตรอนซึ่งมีจำนวนเลเยอร์ 1 2 และ 3 ซึ่งใช้ความไม่เป็นเชิงเส้นแบบขอบเขตยาก (Hard Limiting) ได้ถูกแสดงไว้ในรูปที่ 4-17 ในแถวที่ 2 เป็นค่าตัวเลขที่บ่งชี้จำนวนของขอบ เขตพื้นที่การตัดสินใจ ซึ่งสามารถแสดงด้วยลักษณะโครงตาข่ายที่แตกต่างกัน ส่วน 2 แถวถัด ไปได้แสดงตัวอย่างของบริเวณการตัดสินใจ ซึ่งสามารถจัดรูปได้ สำหรับปัญหาที่เป็น เอ็กซ์คลูซีฟ ออ (Exclusive OR) และปัญหาเกี่ยวกับ เมชเรจัน (Mesh Region)



รูปที่ 4-17 แสดงเพอร์เซ็ปตรอนแบบ 3 ชั้น โดยมีเอาท์พุทเป็น ค่าที่ต่อเนื่อง เอาท์พุท M ตัว และชั้นภายใน 2 ชั้น เราใช้กฎของ การตัดสินใจในการเลือกคลาสที่สอดคล้องกับโหนด เอาท์พุท ที่มี เอาท์พุทที่มีค่ามากที่สุด จากรูป X', X_k'' เป็นเอาท์พุทของโหนดใน ชั้นภายในที่ 1 และ 2 θ_k'' และ θ_j'' เป็นค่าออฟเซตพายใน w_{ij} คือความเครียดเชื่อมต่อ (connection strength) จากอินพุทที่เบี ोनเข้าไปในชั้นภายในชั้นแรก และ w_{ij} 'และ w_{ij} ' เป็นความ เครียดเชื่อมต่อระหว่างชั้นแรก และชั้นที่ 2 และระหว่างชั้นที่สอง และชั้นเอาท์พุทตามลำดับ

เพอร์เซ็ปตรอนแบบซิงเกิลเลเยอร์ สามารถสร้างบริเวณการตัดสินใจแบบครึ่งเพลน (Half-plane decision region) ส่วนเพอร์เซ็ปตรอนแบบ 2 ชั้น สามารถสร้างบริเวณที่มี ลักษณะนูนโค้ง (Convex Region) ได้ บริเวณดังกล่าวซึ่งประกอบด้วย 'Convex polygons' บางครั้งอาจเรียกว่า Convex hull และในรูปที่ 14 แถวกลาง ได้แสดงบริเวณ Convex ที่ไม่มี ขอบเขต (Unbounded convex region) โดย Convex หมายถึง จุดต่อของเส้นใด ๆ (Line Joining Points) ภายในบริเวณนั้น บริเวณโค้งนูน จะถูกสร้างจากการตัดกันบริเวณที่เป็นครึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพลน ซึ่งถูกสร้างโดยโนดแต่ละโนดในชั้นแรกของเปอร์เซ็ปตรอนแบบหลายชั้น โหนดแต่ละโนดในชั้นแรก จะมีลักษณะคล้ายกับเปอร์เซ็ปตรอน ที่มีชั้นเดียว และมีเอาท์พุทที่เป็น 'สูง' (High) สำหรับจุดที่อยู่บนด้านหนึ่งของไฮเปอร์เพลนที่ถูกสร้างขึ้นจากน้ำหนักของตัวมันและออฟเซต (Offset) ถ้าน้ำหนักจากโนดในชั้นแรก N_1 ที่ไปยังโนดเอาท์พุท เป็น 1 ทั้งหมด และทริสโสล ที่โนดเอาท์พุทโนด คือ $N_1 - \epsilon$ โดยที่ $0 < \epsilon < 1$ ค่าโนดเอาท์พุทก็จะเป็น 'สูง' (High)

การวิเคราะห์ที่กล่าวมานี้ จะให้ความเข้าใจในปัญหาของการเลือกจำนวนของโนดเพื่อใช้ในเปอร์เซ็ปตรอน จำนวนของโนดจะต้องมีขนาดใหญ่เพียงพอสำหรับสร้างบริเวณการตัดสินใจ แต่ขนาดจะต้องไม่ใหญ่จนกระทั่งทำให้จำนวนน้ำหนักที่ต้องการไม่สามารถคาดคะเนได้จากข้อมูลสำหรับการฝึก ยกตัวอย่างเช่น โหนด 2 โหนด สามารถที่แก้ปัญหา เอ็กซ์คลูซีฟ ออ ดิงได้แสดงไว้ใน แถวที่ 2 ของรูปที่ 4-16 แต่จำนวนโนดในรูปที่ 4-16 เปอร์เซ็ปตรอนแบบ 2 ชั้น สามารถแยกขอบเขตคลาสได้

สำหรับเปอร์เซ็ปตรอนแบบ 3 ชั้น สามารถสร้างบริเวณการตัดสินใจที่ซับซ้อน และสามารถแยกคลาสดังที่ได้แสดงไว้ในรูปที่ 4-16 ตอนล่างซึ่งเป็นเปอร์เซ็ปตรอน 3 ชั้นสามารถสร้างบริเวณการตัดสินใจเหมือนกับการสร้างโดยตัวคลาสแบบเพื่อนบ้าน (Nearest Neighbour) ซึ่งสามารถพิสูจน์ได้โดยแบ่งบริเวณการตัดสินใจออกเป็นได้โดยแบ่งบริเวณการตัดสินใจออกเป็นไฮเปอร์เล็ก ๆ (ถ้าเป็น 2 อินพุทจะเป็นรูปสี่เหลี่ยม) ในแต่ละไฮเปอร์คิวบ์ (Hypercube) จะต้องใช้จำนวนโนด $2N$ ในชั้นแรก (4 โหนด ถ้าเป็น 2 อินพุท) ถ้าเอาท์พุทของโนดในเลเยอร์ที่ 2 จะเป็น 'สูง' (High) เมื่ออินพุทอยู่ภายในไฮเปอร์คิวบ์ ไฮเปอร์เพลนจะถูกส่งไปยังบริเวณการตัดสินใจที่เหมาะสม โดยการต่อเชื่อมเอาท์พุทของแต่ละโนดในชั้นที่ 2 กับโนดเอาท์พุทที่สอดคล้องกับบริเวณการตัดสินใจซึ่งมีไฮเปอร์คิวบ์ของโนดอยู่ภายใน และแสดงเป็นการกระทำ "หรือ" (Logical OR Operation) ในแต่ละโนดเอาท์พุทโดยการกระทำทางตรรกศาสตร์แบบ "หรือ" นี้ จะถูกแสดงถ้าน้ำหนักจากชั้นภายในที่ 2 ที่ไปยังชั้นเอาท์พุทเป็น 1 และ ทริสโสลในโนดเอาท์พุทเป็น 0.5 ชั้นตอนดังกล่าวสามารถใช้บริเวณโค้งนูน แทนการใช้ไฮเปอร์คิวบ์เล็ก ๆ และสามารถสร้าง ขอบเขตไม่โค้งนูน (Non Convex Region) ดังแสดงได้ในรูปที่ 4-16

จากการวิเคราะห์ดังกล่าวได้แสดงให้เห็นว่าโครงตาข่ายขบวนการผ่านไปข้างหน้าที่คล้ายเปอร์เซ็ปตรอน ไม่จำเป็นต้องใช้จำนวนชั้นที่มากกว่า 3 เพราะ 3 ชั้นสามารถสร้างบริเวณการตัดสินใจที่ซับซ้อนได้ และได้เกิดความเข้าใจในปัญหาการเลือกจำนวนของโนดที่จะใช้ในเปอร์เซ็ปตรอนแบบ 3 ชั้น จำนวนของโนดในชั้นที่ 2 จะต้องมากกว่า 1 เมื่อบริเวณการตัดสินใจมีลักษณะเป็นแบบไม่ติดต่อกัน (Disconnect) และไม่สามารถสร้างได้จากบริเวณโค้งนูนบริเวณเดียว จำนวนของโนดในชั้นที่ 2 จะต้องเท่ากับจำนวนของบริเวณที่ไม่ติดต่อกันในการ

ติดต่อกัน ในการกระจายของอินพุต (Input Distribution) จำนวนของโนดในชั้นแรกจะต้องพอเพียงที่จะให้ขอบที่มากกว่า หรือเท่ากับ 3 สำหรับแต่ละบริเวณโค้งนูนที่ถูกสร้างโดยทุกๆ โนดในชั้นที่ 2

จากที่กล่าวมาข้างบนได้เน้นหนักไปที่เปอร์เซ็ปตรอนแบบหลายชั้นเมื่อเอาท์พุทมีค่าเดียว และมีการใช้ความไม่เป็นเส้นแบบ Hard Limiting พฤติกรรมอย่างง่ายๆ สามารถแสดงได้โดยเปอร์เซ็ปตรอน แบบหลายชั้น และมีโนดเอาท์พุทหลายโนด และกฎของการตัดสินใจก็เพื่อทำการเลือกชั้นก็สอดคล้องกับโนดเอาท์พุท และเอาท์พุทค่าสูงๆ พฤติกรรมดังกล่าวของโครงตาข่ายจะมีความซับซ้อนมาก เพราะว่าบริเวณการตัดสินใจจะถูกกำหนดขอบเขตโดยเส้นโค้งเรียบ (Smooth Curve) แทนที่จะเป็นส่วนของเส้นตรง และการวิเคราะห์ก็มีความยุ่งยากมากกว่า อย่างไรก็ตาม โครงตาข่ายดังกล่าวสามารถฝึกหัดโดยการใช้ลำดับขั้นตอนฝึกหัดแบบการกระจายย้อนกลับ

4.4.7 กฎของเดลต้า (Delta Rule)

ความสำคัญโดยทั่วไปของขั้นตอนในการฝึกเปอร์เซ็ปตรอน ซึ่งเรียกว่ากฎของเดลต้าได้ขยายเทคนิค เพื่อความต่อเนื่องของอินพุต และเอาท์พุท และเพื่อที่จะดูว่า การพัฒนาเกิดขึ้นได้อย่างไร เราจะเริ่มพิจารณาจาก δ ซึ่งเป็นค่าความแตกต่างระหว่างค่าที่ต้องการ หรือเป้าหมาย T กับค่าเอาท์พุทจริง A เขียนแทนได้ดังสมการนี้

$$\delta = (T - A) \quad (4.4)$$

ในกรณีที่ $\delta = 0$ จะเป็นไปตามขั้นตอน 2 a. ซึ่งแสดงว่าค่าเอาท์พุทถูกต้อง และไม่ต้องทำอะไร ขั้นตอน 2 b. จะสอดคล้องกับ $\delta > 0$ และขั้นตอน 2 c. จะสอดคล้องกับ $\delta < 0$ ในกรณีต่างๆ เหล่านี้ ขั้นตอนของการฝึกเปอร์เซ็ปตรอน จะเป็นที่พึงพอใจถ้า δ ถูกคูณด้วยค่าอินพุต X_i และผลที่ได้จะนำไปควบคุมขนาดโดยเฉลี่ยของการเปลี่ยนแปลงน้ำหนัก

$$\Delta_i = \eta \delta X_i \quad (4.5)$$

$$w_i(n+1) = w_i(n) + \Delta_i \quad (4.6)$$

เมื่อ

Δ_i เป็นค่าเปลี่ยนแปลงที่สอดคล้องกับ X_i

$w_i(n+1)$ เป็นค่าของน้ำหนัก i หลังจากการปรับ

$w_i(n)$ เป็นค่าของน้ำหนัก i ก่อนการปรับ

กฎของเดลต้า จะทำการปรับน้ำหนักให้เป็นไปด้วยความเหมาะสม สำหรับค่าเป้าหมายและค่าเอาต์พุตจริง ทั้งค่าที่เป็นบวก และลบ และสำหรับเอาต์พุตกับอินพุตที่เป็นฐานสองให้มีความต่อเนื่อง

ปัญหาขั้นตอนการฝึกเปอร์เซ็ปตรอน

เนื่องจากในสภาพที่เป็นจริงค่าอินพุตจะแปรตามเวลา และจะแยกกันได้ทีละเวลาหนึ่งๆ เท่านั้น และยังไม่มีข้อความในบทพิสูจน์ของขั้นตอนการฝึกเปอร์เซ็ปตรอน ที่จะเป็นตัวบ่งชี้ว่า เราจะต้องใช้ขั้นตอนการฝึกเป็นจำนวนเท่าไร เพียงแต่มีข้อสรุปเล็กน้อยว่า การฝึกจะใช้จำนวนขั้นตอนที่มีค่าใดค่าหนึ่ง จึงไม่มีข้อพิสูจน์ว่า ขั้นตอนการฝึกเปอร์เซ็ปตรอน จะเร็วกว่าการใช้เวลาในการปรับน้ำหนักทั้งหมด ปัญหาทั้งหมดนี้ยังหาคำตอบที่พึงพอใจไม่ได้ และยังเกี่ยวข้องอยู่กับกลุ่มของข้อมูลที่ใช้ในการเรียน จึงเป็นปัญหาที่ต้องมีการค้นคว้ากันต่อไป

4.5 การกระจายแบบย้อนกลับ (Back-Propagation)

ในช่วงหลาย ๆ ปีที่ผ่านมา ยังไม่มีการกล่าวทางทฤษฎีถึงขั้นตอนการฝึกทฤษฎีโครงตาข่ายนิเวรอลแบบชั้นซ้อนเลย เนื่องจากทฤษฎีโครงตาข่ายแบบชั้นเดียวได้รับการพิสูจน์ว่ามีขีดจำกัดในการนำไปใช้งาน

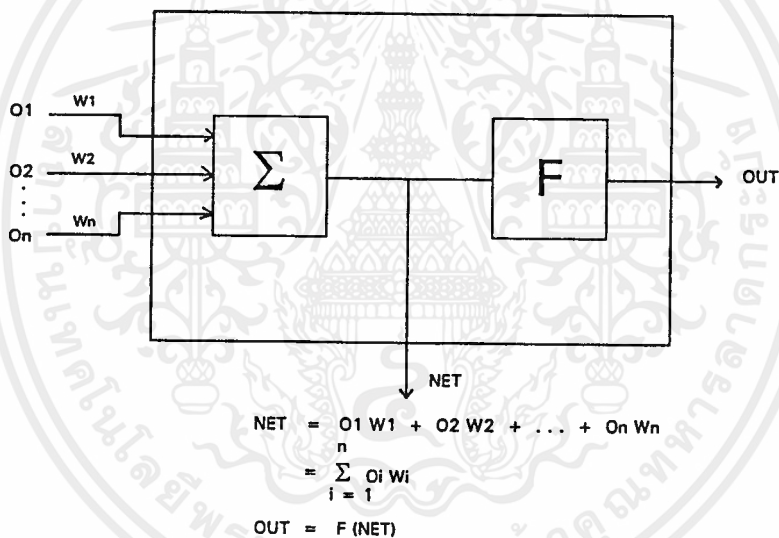
การสร้างขั้นตอนการทำงานแบบการกระจายย้อนกลับได้มีบทบาทสำคัญ ทำให้มีการกลับมาให้ความสนใจกับทฤษฎีโครงตาข่ายนิเวรอล การกระจายแบบย้อนกลับเป็นหลักการที่เป็นระบบเพื่อใช้ในการฝึกทฤษฎีโครงตาข่ายนิเวรอลแบบชั้นซ้อน ซึ่งมีการกำหนดทางคณิตศาสตร์ที่ไม่ยากต่อการปฏิบัติจริง และถึงแม้ว่าจะมีขีดจำกัดก็ตามการกระจายแบบย้อนกลับก็สามารถขยายช่วงของขีดจำกัดทำให้สามารถนำเอาโครงตาข่ายนิเวรอลมาใช้ได้ ซึ่งการกระจายแบบย้อนกลับก็ได้แสดงถึงความสามารถในการวิเคราะห์จนประสบความสำเร็จ

ประวัติของการศึกษาการกระจายแบบย้อนกลับที่น่าสนใจมีดังต่อไปนี้ Rumelhart, Hinton และ Williams (1986) ได้แสดงถึงขั้นตอนการทำงานของการกระจายแบบย้อนกลับอย่างรวบรัดและชัดเจน ซึ่งก่อนหน้านั้นก็เคยมีงานตีพิมพ์ในปี 1982 ของ Parker ที่กล่าวถึงในเรื่องเดียวกันนี้ Werbos ได้ค้นพบวิธีการในการอธิบายหลักการนี้ตั้งแต่ปี 1974 ซึ่งการวิจัย

ทฤษฎีโครงตาข่ายนิรพลได้มีการตีพิมพ์ในหนังสือและวารสาร (Journal) จากหลายแห่งและนำไปใช้ได้หลายๆ ด้าน

4.5.1 ขั้นตอนในการฝึกหัดการกระจายแบบย้อนกลับ

รูปที่ 4-18 จะแสดงนิรพลซึ่งจะใช้เป็นส่วนประกอบพื้นฐานในโครงข่ายแบบการกระจายแบบย้อนกลับ เมื่อมีอินพุตเข้ามาที่ด้านหนึ่งของชั้น อินพุตต่างๆ จะถูกคูณด้วยน้ำหนัก (Weight) และถูกรวมเข้าด้วยกัน ซึ่งผลรวมนี้จะเรียกว่า เนท (NET) และจะต้องมีการคำนวณสำหรับแต่ละนิรพลในโครงตาข่าย หลังจากที่ได้คำนวณได้ค่า NET ออกมาแล้ว จะใช้ฟังก์ชัน F (Activation Function) ในการปรับค่า และทำให้ได้สัญญาณออกมา (OUT)



รูปที่ 4-18 จะแสดงนิรพลที่ถูกสร้างขึ้นกับฟังก์ชันกระตุ้น

ผลรวมอินพุตที่โนดในชั้น j คือ

$$NET_j = \sum w_{ji} o_i \quad (4.7)$$

เอาต์พุตของโนด j คือ

$$o_j = F(NET_j) \quad (4.8)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

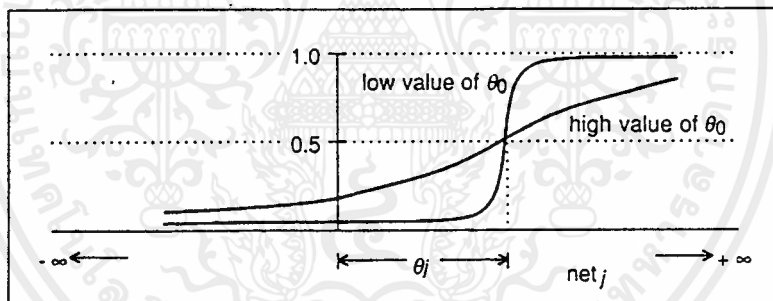
เมื่อ F คือ ฟังก์ชันกระตุ้น (Activation Function)

$$o_j = 1 / (1 + \exp(-(\text{net}_j + \theta_j) / \theta_0)) \quad (4.9)$$

ในสมการ (4.9) พารามิเตอร์ θ_j คือ ทริสโฮล (Threshold) หรือ ไบอัส (Bias) ผลของ θ_j ฟังก์ชันกระตุ้น ไปทางทางซ้ายหรือขวา และผลของ θ_0 จะเป็นตัวเปลี่ยนแปลงรูปร่างของซิกมอยด์ ค่า θ_0 ต่ำๆ จะทำให้ซิกมอยด์มีคุณลักษณะเป็นแบบทริสโฮลลอจิกยูนิต (Threshold-logic unit) และค่า θ_0 สูง ๆ จะทำให้ฟังก์ชันเปลี่ยนแปลงแบบเรียบๆ ดังรูปที่โน้ดในชั้น k เราจะได้

$$\text{NET}_k = \sum w_{kj} o_j \quad (4.10)$$

$$o_k = F(\text{NET}_k) \quad (4.11)$$



รูป4-19 ฟังก์ชันซิกมอยด์กับไบอัส และการตัดแปลงรูปร่างต่างๆ

สมการที่ (4.9) เรียกว่า ซิกมอยด์ (Sigmoid) อยู่ในรูปการอนุพันธ์อย่างง่าย ๆ ซึ่งเราจะใช้ซิกมอยด์ เป็นเครื่องมือในการสร้างขั้นตอนการทำงาน ของการกระจายแบบย้อนกลับ

ซิกมอยด์จะมีค่าอยู่ในช่วงของเน็ต ดังนั้น O_j จะมีค่าระหว่าง 0 กับ 1 ซึ่งบางครั้งเรียกว่า ฟังก์ชันการบดอัด (Logistic or Squashing Function) เนื่องจากโครงตาข่ายชั้นซ่อน มีความสามารถในการวิเคราะห์ได้ดีกว่า โครงตาข่ายชั้นธรรมดา สามารถนำไปวิเคราะห์สมการที่ไม่เป็นเชิงเส้น โดยฟังก์ชันการบดอัดจะทำให้เกิดการไม่เป็นเชิงเส้น

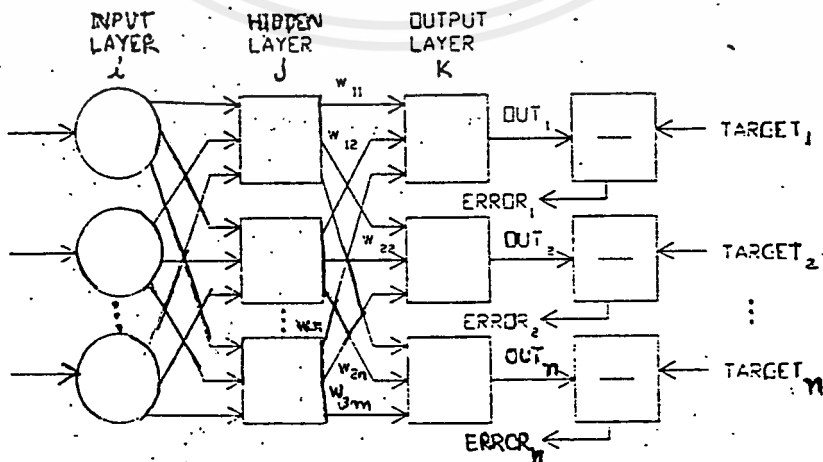
$$\delta O_j / \delta \text{net} = O_j (1 - O_j) \quad (4.12)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชันที่ใช้ในขั้นตอนการทำงานของโครงข่ายย้อนกลับ จะต้องเป็นฟังก์ชันที่สามารถดิฟเฟอเรนทิเอตได้ ซึ่งซิกมอยด์มีคุณสมบัตินี้ และยังมีข้อได้เปรียบอีกคือ การเตรียมพร้อมสำหรับการควบคุมอัตราขยาย (Gain) โดยอัตโนมัติ ในสัญญาณต่างๆ (เนทมีค่าใกล้ 0) ความชันของกราฟอินพุตต่อเอาต์พุตจะชัน ซึ่งจะทำให้มีการปรับอัตราขยายให้สูง และเมื่อขนาดของสัญญาณมีค่ามากขึ้น อัตราขยายจะมีค่าลดลง ซึ่งวิธีนี้จะทำให้โครงข่ายสามารถปรับสัญญาณที่มีค่ามากๆ โดยไม่เกิดการอิ่มตัว ในขณะที่สัญญาณต่ำๆ ได้รับอนุญาตให้ผ่านโดยไม่มีการปรับเพิ่มขึ้น

4.5.2 โครงข่ายชั้นขั้วซ่อน

รูปที่ 4-20 จะแสดงถึงโครงข่ายชั้นขั้วซ่อนที่มีความเหมาะสมจะใช้ในการกระจายแบบย้อนกลับ (รูปจะถูกแปลงให้อยู่ในรูปอย่างง่าย เพื่อจะแสดงให้เห็นอย่างเด่นชัด) กลุ่มของนิวรอนกลุ่มแรก (ต่ออยู่กับอินพุต) จะเป็นเสมือนในการกระจายที่จุดนี้จะไม่มีการรวมค่าอินพุตสัญญาณอินพุตจะสามารถผ่านจุดนี้ไปหาน้ำหนักต่างๆ และออกไปเป็นค่าเอาต์พุตได้โดยง่าย ในแต่ละชั้นจะมีการสร้างสัญญาณ NET และ OUT ดังที่กล่าวมาแล้ว ตามเอกสารทางวิชาการไม่ได้มีการระบุถึงจำนวนชั้นใน ผู้เขียนบางท่านหมายถึงจำนวนของชั้นในโครงข่ายนั้นๆ (รวมไปถึงชั้นอินพุตที่ไม่มีการรวมกันด้วย) และบางท่านก็หมายถึงจำนวนชั้นของน้ำหนัก แต่ในที่นี้ตามรูปที่ 4-20 จะแสดงถึงโครงข่ายที่ประกอบไปด้วยชั้นทั้งหมด 2 ชั้น (Two Layer) นิวรอนจะถูกเชื่อมโยงด้วยกลุ่มของน้ำหนักซึ่งต่ออยู่กับอินพุต ดังนั้นน้ำหนักในชั้นที่ 1 จะสิ้นสุดในนิวรอนชั้นที่ 1 โดยอินพุต หรือชั้นกระจายจะเป็นชั้นที่ 0 การกระจายแบบย้อนกลับนี้ จะสามารถใช้กับโครงข่ายที่มีจำนวนชั้นเท่าไรก็ได้ ในที่นี้จะอธิบายถึง 2 ชั้น

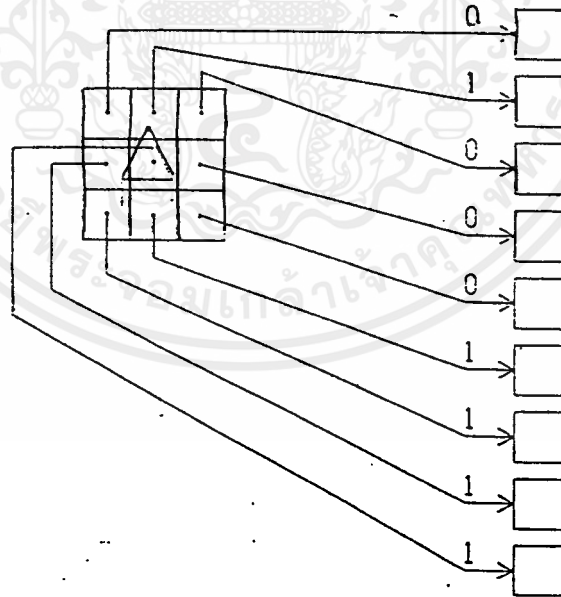


รูป 4-20 โครงข่ายการกระจายย้อนกลับแบบ 2 ชั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.3 ภาพโดยรวมของการฝึก (An Overview of training)

จุดประสงค์ในการฝึกโครงข่ายก็เพื่อที่จะปรับน้ำหนักเพื่อใช้กับกลุ่มของอินพุท และให้ได้กลุ่มของเอาต์พุตตามต้องการ เพื่อให้มีความกะทัดรัด กลุ่มของอินพุท และเอาต์พุท จะปรับให้อยู่ในรูปของเวกเตอร์ ในการฝึกจะสมมติว่า เวกเตอร์อินพุทแต่ละตัว ถูกนำให้เป็นคู่กับเวกเตอร์เป้าหมาย ซึ่งเวกเตอร์เป้าหมายนี้จะไปตามเอาต์พุทที่ต้องการ ซึ่งที่กล่าวมานี้ จะเรียกว่า คู่ของการฝึก (Training Pair) โดยทั่วไป โครงข่ายจะถูกฝึกมากกว่าจำนวนคู่ของการฝึก ตัวอย่างเช่นส่วนอินพุทของคู่ของการฝึกจะประกอบด้วยรูปแบบ และ ซึ่งแสดงในการจำลองเป็นฐานสองของตัวอักษร ในรูป 4-21 จะแสดงกลุ่มของอินพุทของตัวอักษร A ซึ่งเขียนอยู่บนกริด ถ้ามีเส้นลากผ่านรูปสี่เหลี่ยม อินพุทของ นิวรอน จะมีค่าเป็น 1 และถ้าไม่มี จะเป็น 0 บาทเอาต์พุทจะเป็นตัวเลขซึ่งแสดงถึงตัวอักษร A หรือในบางที่อาจเป็นกลุ่มของ 1 และ 0 ซึ่งสามารถสร้างรูปแบบเอาต์พุทออกมาได้ ถ้ามีความต้องการจะฝึกโครงข่ายให้รู้จักตัวอักษรทั้งหมด จะต้องมี 26 คู่ของการฝึก ซึ่งกลุ่มของการฝึกจะเรียกว่า กลุ่มการฝึก (Training Set)



รูปที่ 4-21 การจินตนาการการเรียนรู้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก่อนจะเริ่มขบวนการฝึก น้ำหนักทุกตัวจะต้องถูกกำหนดค่าเริ่มต้น โดยค่าน้อยๆที่ได้จากการสุ่ม ซึ่งเงื่อนไขนี้จะทำให้แน่ใจได้ว่า โครงตาข่ายจะไม่เกิดการอิมิตัวจากค่ามากๆของน้ำหนัก และเป็นการป้องกันเส้นทางที่แน่นอน ซึ่งถ้าน้ำหนักทั้งหมดมีค่าเริ่มต้นเท่ากัน และค่าที่ต้องการมีค่าไม่เท่ากัน โครงตาข่ายจะไม่สามารถเรียนรู้ได้

ขั้นตอนในการฝึกของการกระจายแบบย้อนกลับจะมีดังต่อไปนี้

1. เลือกคู่ฝึกตัวต่อไปจากกลุ่มที่ฝึก ให้เวกเตอร์อินพุตกับอินพุตของโครงตาข่าย
2. คำนวณค่าเอาต์พุตของโครงตาข่าย
3. คำนวณค่าผิดพลาดระหว่างค่าเอาต์พุตของโครงตาข่ายกับค่าที่ต้องการ (เวกเตอร์เป้าหมายจากคู่ฝึก)
4. ปรับค่าน้ำหนักของโครงตาข่ายในหนทางที่จะทำให้ค่าผิดพลาดน้อยที่สุด
5. นำขั้นตอนที่ 1-4 ซ้ำ จนกระทั่งค่าผิดพลาดมีค่าต่ำลงพอจนยอมรับได้

หลังจากทำขั้นตอนที่ 1-4 แล้ว ค่าผิดพลาดระหว่างค่าเอาต์พุตที่คำนวณได้กับค่าเอาต์พุตเป้าหมายจะมีค่าลดลงจนกระทั่งเป็นค่าที่ยอมรับได้ และจะสามารถกล่าวได้ว่า โครงตาข่ายได้รับการฝึกแล้ว ณ จุดนี้ โครงตาข่ายจะสามารถยอมรับได้ และค่าน้ำหนักจะไม่เปลี่ยนแปลงไปอีก สามารถนำไปใช้งานได้เลย เพียงแต่ป้อนค่าอินพุตเข้าไป ก็จะได้ค่าเอาต์พุตที่ต้องการออกมาทันที

4.5.4 ขบวนการผ่านไปข้างหน้า (Forward Pass)

ขั้นตอนที่ 1 และ 2 จะแสดงอยู่ในรูปเวกเตอร์ดังต่อไปนี้ เวกเตอร์อินพุต X จะเป็นค่าที่ป้อนเข้าไป และเวกเตอร์เอาต์พุต Y เป็นผลที่ได้ออกมา คู่ของเวกเตอร์เป้าหมายอินพุตคือ X และ T จะได้มาจากกลุ่มการฝึก การคำนวณจะถูกดำเนินการจาก X เพื่อให้ได้รับเวกเตอร์เอาต์พุต Y

เราจะเห็นว่าการคำนวณ โครงตาข่ายชั้นซับซ้อน จะมีการทำทีละชั้น โดยเริ่มจากชั้นที่ใกล้อินพุตมากที่สุด ค่า NET ของแต่ละนิวรอนในชั้นแรกจะถูกคำนวณจากผลรวมตามน้ำหนักของค่าอินพุตที่เข้ามาของนิวรอน ฟังก์ชัน F (Activation Function) จะทำการลดค่า NET ลงเพื่อผลิตค่าเอาต์พุตสำหรับแต่ละนิวรอนในชั้นนั้น หลังจากที่ไดกลุ่มของเอาต์พุตของชั้นนั้น ค่าเอาต์พุตนั้นก็ทำหน้าที่เป็นอินพุตของชั้นต่อไป และมีการทำขั้นตอนซ้ำเดิมต่อไปจากชั้นหนึ่งไปสู่อีกชั้นหนึ่ง จนกระทั่งได้ค่าเอาต์พุตของโครงตาข่ายออกมา

ขั้นตอนนี้สามารถแสดงอย่างรวบรัดด้วยเครื่องหมายทางเวกเตอร์ ค่าน้ำหนัก

ระหว่างนิวรอนสามารถพิจารณาอยู่ในรูปเมตริกซ์ W ตัวอย่างเช่น น้ำหนักจากนิวรอน 8 ในเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

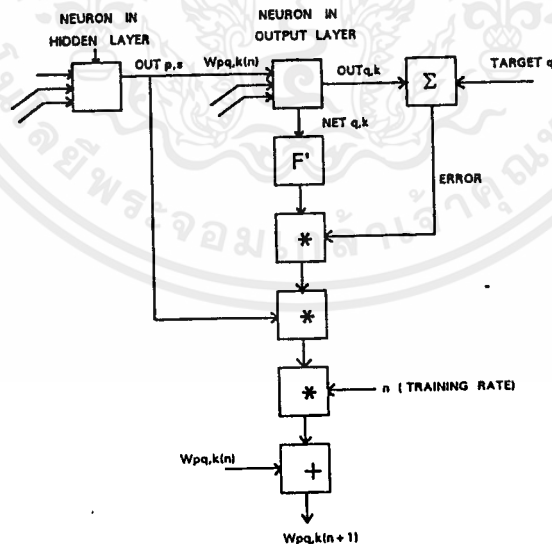
ชั้นที่ 2 ไปหา นิวรอน 5 ในชั้นที่ 3 จะเขียนแทนด้วย W_{85} เวกเตอร์ NET ของชั้น N จะแสดงในรูปของผลที่ได้จาก X และ W ด้วยสัญลักษณ์ทางเวกเตอร์ $N = XW$ และเวกเตอร์เอาต์พุต O จะได้จากการใส่ฟังก์ชัน F กับ N

$$O = F(XN) \quad (4.13)$$

เวกเตอร์เอาต์พุตในชั้นนี้จะเป็นเวกเตอร์อินพุตในชั้นต่อไป ดังนั้นการคำนวณเอาต์พุตของชั้นสุดท้ายจะต้องมีการประยุกต์ใช้สมการ ในแต่ละชั้น จากอินพุตของโครงข่ายไปสู่เอาต์พุตของมัน

4.5.5 ขบวนการผ่านย้อนกลับ (Reverse Pass)

ในการปรับน้ำหนักของเอาต์พุตให้ใช้การได้กับนิวรอนแต่ละตัวเพื่อให้ได้เป้าหมายตามที่เรากำลังต้องการนั้น จะทำได้โดยการใช้การปรับแตงน้ำหนักตามกฎของเดลต้า ชั้นภายในจะเรียกว่าเป็น Hidden Layer ซึ่งถ้าค่าเอาต์พุตไม่มีค่าเป้าหมาย เพื่อที่จะใช้ในการเปรียบเทียบแล้ว การฝึกสอนก็จะมีคามยุ่งยากมากขึ้น



รูปที่ 4-22 การฝึกน้ำหนักในชั้นเอาต์พุต

รูปที่ 4-22 แสดงถึงขั้นตอนการฝึกสำหรับน้ำหนักตัวเดียวจากนิรอน P ในชั้นภายใน J ไปสู่นิรอน Q ในชั้นเอาต์พุต K เอาต์พุตของนิรอนในชั้น K จะถูกลบออกจากค่าเป้าหมาย เพื่อที่จะหาค่าสัญญาณผิดพลาดซึ่งจะถูกคูณด้วยอนุพันธ์ของฟังก์ชัน $(O_j)(1-O_j)$ ซึ่งถูกคำนวณสำหรับชั้นของนิรอน K และจะได้ค่า δ ออกมา

$$\delta = O_j(1-O_j)(\text{Target} - O_j) \quad (4.14)$$

$$\Delta w_{pq,k} = \eta \delta_{q,k} O_{pj} \quad (4.15)$$

$$w_{pq,k}(n+1) = w_{pq,k}(n) + \Delta w_{pq,k} \quad (4.16)$$

เมื่อ

$$w_{pq,k}(n) = \text{เป็นค่าของน้ำหนักจากนิรอน } p \text{ ในชั้นภายในไปหา}$$

$$\text{นิรอน } q \text{ ในชั้นเอาต์พุตที่ชั้น } n \text{ (ต้องการปรับ) ตัว}$$

$$\text{ห้อย } k \text{ จะแสดงถึงชั้นที่เป็นเป้าหมาย}$$

$$w_{pq,k}(n+1) = \text{ค่าของน้ำหนักที่ปรับใหม่}$$

$$\delta_{q,k} = \text{ค่าของ } \delta \text{ สำหรับนิรอน } q \text{ ไปหาชั้นเอาต์พุต } k$$

$$O_{p,j} = \text{ค่าของ } O_j \text{ ของนิรอน } p \text{ ในชั้นภายใน}$$

$$\eta = \text{ค่า Coefficient } 0.01 \text{ to } 1.0$$

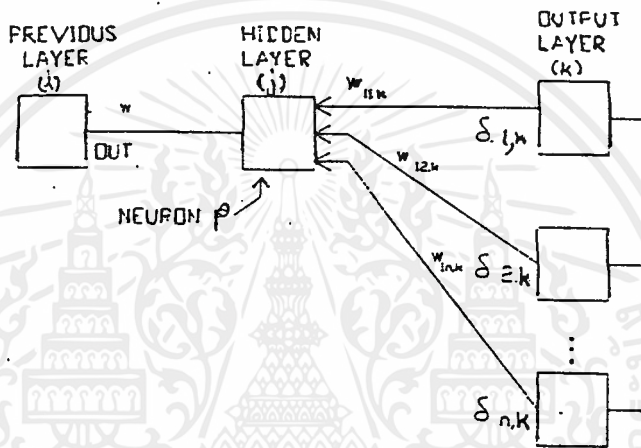
การปรับน้ำหนักในชั้นภายในซึ่งไม่มีเวกเตอร์เป้าหมาย จะไม่สามารถใช้ขั้นตอนการฝึกดังที่กล่าวมาจึงไม่สามารถใช้ได้ทั้งหมด แต่เรายังคงใช้สมการ กับทุกๆชั้น เพียงแต่จะมีการเปลี่ยนแปลงของตัวแปรบางตัวเท่านั้น รูป 4-23 จะแสดงถึงการฝึกน้ำหนักในชั้นภายใน ชั้นแรก δ จะถูกคำนวณสำหรับนิรอนแต่ละตัวที่ชั้นเอาต์พุต ตามสมการ ซึ่งน้ำหนักที่ป้อนเข้าสู่ชั้นเอาต์พุตจะถูกปรับ หลังจากนั้น δ จะถูกกระจายย้อนกลับผ่านน้ำหนักเดิม เพื่อที่จะสร้างค่า δ สำหรับแต่ละนิรอนในชั้นภายในแรกค่าของ δ จะถูกใช้ในการปรับน้ำหนักของชั้นภายใน และจะทำวิธีเดียวกันนี้กับชื่อต่อไป

เมื่อพิจารณาถึงนิรอนเดียวในชั้นภายใน ก่อนที่จะถึงชั้นเอาต์พุต ในขบวนการไปข้างหน้า นิรอนจะกระจายค่าเอาต์พุตให้กับนิรอนในชั้นเอาต์พุตผ่านน้ำหนักที่เชื่อมต่อ ในระหว่างการฝึกน้ำหนักเหล่านี้จะมีการปฏิบัติในทางตรงกันข้ามกัน และส่งผ่านค่าของ δ จากชั้นเอาต์พุตมายังชั้นภายใน น้ำหนักแต่ละตัวจะถูกคูณด้วยค่า δ ของนิรอนซึ่งมันต่ออยู่

ในชั้นเอาต์พุต ค่าของ δ จะได้มาจากการรวมผลทั้งหมด และคูณด้วยอนุพันธ์ของฟังก์ชันตามสมการนี้

$$\delta = o_{pj}(1-o_{pj})(\sum \delta_{q,k} w_{pq,k}) \quad (4.17)$$

(ดูรูปที่ 4-23) เมื่อมีค่า δ เดิมอยู่ น้ำหนักที่ป้อนเข้าให้กับชั้นภายในแรกจะถูกปรับโดยใช้สมการ และการปรับจะแสดงชั้นที่ถูกต้อง



รูปที่ 4-23 การฝึกของน้ำหนักในชั้นภายใน

4.5.6 โมเมนตัม (Momentum)

Rumelhart, Hinto และ Williams (1986) ได้อธิบายถึงวิธีการในการปรับปรุงเวลาในการฝึกของขั้นตอนแบบการกระจายย้อนกลับ เรียกวิธีการนี้ว่า โมเมนตัม วิธีการนี้จะมีการปรับน้ำหนักเป็นสัดส่วนกับการเปลี่ยนแปลงน้ำหนักในครั้งที่แล้ว สมการในการปรับจะมีดังนี้

$$\Delta w_{pq,k(n+1)} = \eta \delta_{q,k} o_{pj} + \alpha [\Delta w_{pq,k}] \quad (4.18)$$

$$w_{pq,k(n+1)} = w_{pq,k(n)} + \Delta w_{pq,k(n+1)} \quad (4.19)$$

เมื่อ ค่า α คือค่าสัมประสิทธิ์โมเมนตัม (Momentum efficient) ซึ่งโดยปกติจะตั้งที่ 0.9 การใช้วิธีการโมเมนตัมนี้ โคจรตาข่ายมีแนวโน้มที่จะเป็นไปตามค่าที่ต้องการ แทนที่จะมีการตัดและลดลงอย่างรวดเร็วจากด้านหนึ่งไปสู่อีกด้านหนึ่ง วิธีการนี้จะเหมาะสมกับเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาบางปัญหาเท่านั้น แต่จะมีผลกระทบเล็กน้อยกับปัญหาอื่นๆ และในทางปฏิบัติแล้ว ค่าสัมประสิทธิ์โมเมนต์นั้น สามารถตั้งได้ตั้งแต่ 0.01 ถึง 1 ขึ้นอยู่กับความเหมาะสมกับงานนั้นๆ

Sejnowski และ Rosenberg (1987) ได้อธิบายวิธีการคล้ายคลึงกันแต่มีพื้นฐานอยู่บน การทำให้เรียบด้วย เอ็กซ์โปเนนเชียล (Exponential Smoothing) ซึ่งเหมาะกับการนำไปใช้ดังสมการ

$$\Delta w_{pq,k(n+1)} = \alpha [\Delta w_{pq,k}] + (1 - \alpha) \delta_{q,k} O_{pj} \quad (4.20)$$

$$w_{pq,k(n+1)} = w_{pq,k(n)} + \eta \Delta w_{pq,k(n+1)} \quad (4.21)$$

เมื่อค่า α คือค่าสัมประสิทธิ์ความเรียบ มีช่วงตั้งแต่ 0 ถึง 1 ถ้าค่า α เท่ากับ 0 ความเรียบจะต่ำ และจะมีการปรับน้ำหนักใหม่ แต่ถ้า α มีค่าเท่ากับ 1 การปรับใหม่จะไม่เกิดขึ้น และจะใช้ค่าเดิมซ้ำต่อไป ช่วงระหว่าง 0 กับ 1 จะเป็นทั้งที่ชื่อทำให้การปรับน้ำหนักถูกทำให้เรียบขึ้นดีขึ้นโดยเป็นสัดส่วนกับ α และค่า η คือสัมประสิทธิ์อัตราการเรียนรู้ซึ่งจะใช้ในการปรับขนาดของการเปลี่ยนแปลงน้ำหนักโดยเฉลี่ย

4.5.7 ขั้นตอนการทำงานขั้นสูง (Advanced Algorithms)

นักวิจัยหลายคนได้ปรับปรุงขั้นตอนการทำงานแบบการกระจายย้อนกลับ ในส่วนนี้เราจะกล่าวถึงงานของนักวิจัยบางท่าน เช่น Parker (1987) ได้อธิบายถึงการปรับปรุงความเร็วในการลู่เข้า (Speed of convergence) ของขั้นตอนการทำงานแบบการกระจายย้อนกลับ ที่เรียกว่าการกระจายแบบย้อนกลับอันดับ 2 (Second-order backpropagation) ซึ่งใช้ออนุพันธ์อันดับ 2 เพื่อทำให้การประมาณเกี่ยวกับการปรับค่าน้ำหนัก มีความแม่นยำมากขึ้น ซึ่งต้องใช้วิธีการคำนวณที่ซับซ้อนขึ้นเมื่อเปรียบเทียบกับวิธีการกระจายแบบย้อนกลับอันดับ 1 (First order backpropagation)

Stornetta และ Huberman (1987) ได้กล่าวถึงวิธีการง่ายๆ สำหรับการปรับปรุงคุณลักษณะของการฝึกโครงข่ายแบบการกระจายแบบย้อนกลับ ซึ่งได้กล่าวว่า ช่วงทางไดนามิกระหว่าง 0-1 ของชั้นภายใน และอินพุทไม่ใช่ค่าที่ดี เนื่องจากขนาดของการปรับน้ำหนัก $\Delta w_{pq,k}$ เป็นสัดส่วนกับระดับเออร์ทพุทของนิวรอนซึ่งสร้าง OUT ออกมา ในกรณีระดับ 0 ทำให้ไม่เกิดการปรับน้ำหนัก ถ้าเวกเตอร์ของอินพุทที่เป็นฐาน 2 ครั้งหนึ่งของอินพุทโดยเฉลี่ยแล้วจะเท่ากับ 0 และค่าน้ำหนักที่มันต่อเชื่อมอยู่จะไม่มีการฝึก ซึ่งการแก้ปัญหในการ

เปลี่ยนช่วงของอินพุทเป็น $\pm 1/2$ และการเพิ่มไบอัสเข้าไปในฟังก์ชันเพื่อปรับช่วงของนิวรอนเอาต์พุทเป็น $\pm 1/2$ จะทำให้ได้สมการฟังก์ชันบดอัดขึ้นมาใหม่ คือ

$$\text{OUT} = -1/2 + 1/(1+(\exp(-\text{net}))) \quad (4.22)$$

เวลาในการลู่เข้าจะลดลงประมาณ 30-50% เป็นตัวอย่างของการปรับปรุงประสิทธิภาพของขั้นตอนการทำงานให้ดีขึ้น

วิธีการนี้จะทำให้ซิกมอยด์มีค่าตั้งแต่ -0.5 ถึง 0.5 ซึ่งต่างจากสมการเดิม (มีค่า 0 ถึง 1) สามารถแก้ปัญหาที่ต้องการเอาต์พุทที่เป็นค่าลบได้

4.5.8 การประยุกต์ใช้งาน

การกระจายแบบย้อนกลับมีการนำไปประยุกต์ใช้งานกันอย่างกว้างขวาง ซึ่งจะมีตัวอย่างดังต่อไปนี้

NEC ในประเทศญี่ปุ่นได้ประกาศเมื่อเร็ว ๆ นี้ว่า ได้นำเอาทฤษฎีการกระจายแบบย้อนกลับไปใช้ในระบบชนิดใหม่ (New Optical -Character-Recognition System) ซึ่งสามารถปรับปรุงให้มีความถูกต้องสูงกว่า 99% การปรับปรุงนี้จะถูกทำสำเร็จภายใต้การรวมกับของกันของขั้นตอนการทำงานเดิน กับโครงตาข่ายแบบย้อนกลับ

Sejnowski และ Rosenberg (1987) ได้ผลิต Nettetalk สำเร็จ ซึ่งระบบนี้จะแปลงตัวอักษรภาษาอังกฤษที่พิมพ์ออกมาให้เป็นคำพูดที่สละสลวย ซึ่งเทปที่บันทึกนี้จะเก็บขั้นตอนการฝึก และใช้ประโยชน์ในการสอนภาษาอังกฤษสำหรับเด็กหัดพูดช่วงอายุต่าง ๆ

Burr (1987) ได้ใช้การกระจายแบบย้อนกลับในเครื่องมือตรวจสอบคำที่เขียนด้วยมือเป็นภาษาอังกฤษ โดยใช้ตัวอักษรที่มีขนาดธรรมดา วางลงบนกริด แล้วตรวจสอบเส้นที่ลากผ่านรูปลี่เหลี่ยมในกริดนั้น การตรวจสอบนี้จะทำเป็นอินพุทเข้าไปสู่โครงตาข่ายการกระจายแบบย้อนกลับ ซึ่งเขาบอกว่ามีความถูกต้องถึง 99.7% เมื่อตัวอักษรเป็นตัวพิมพ์

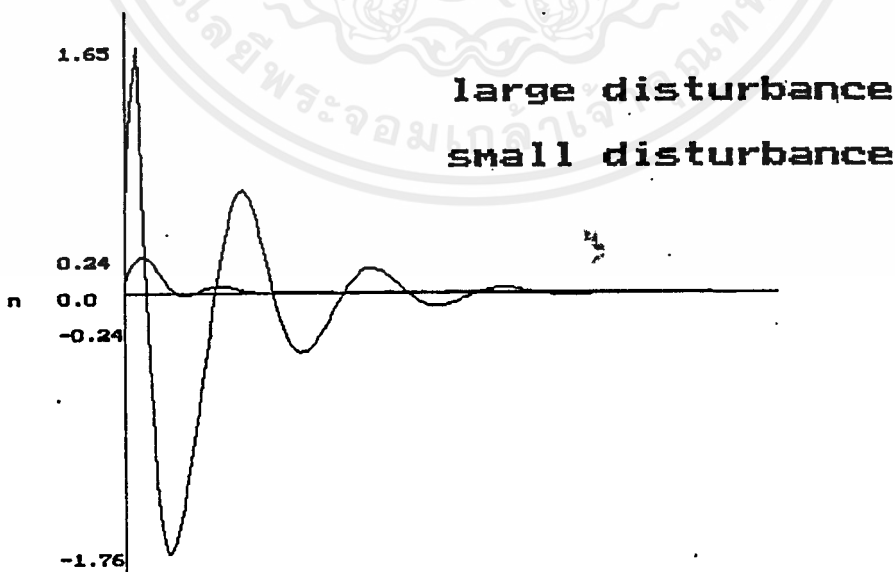
บทที่ 5

ขั้นตอน และผลการวิเคราะห์

เมื่อนำโปรแกรมใช้ควบคุมระบบจำลองซึ่งมีการต่อกับบัสอนันต์ (Infinite Bus) (ซึ่งมีค่ากำลังจริงและกำลังจินตภาพที่คงที่) ซึ่งเมื่อมีการรบกวนระบบอย่างมากและมีการรบกวนระบบน้อยจะได้ผลการควบคุมดังรูปที่ 5-1 (ระบบที่ 1) และเมื่อทำการเปลี่ยนเครื่องจักรที่ใช้ทำการจำลอง (ระบบที่ 2) จะทำให้ค่าพารามิเตอร์ต่างๆ เปลี่ยนแปลง ไป จะได้ผลการควบคุมดังรูปที่ 5-2

เมื่อทำการใส่ตัวควบคุมโดยกฎพื้นฐานเข้าไปในระบบ จะทำให้ได้ผลการควบคุมดังรูปที่ 5-3 สำหรับระบบที่ 1 และจะได้ผลการควบคุมดังรูปที่ 5-4 สำหรับระบบที่ 2

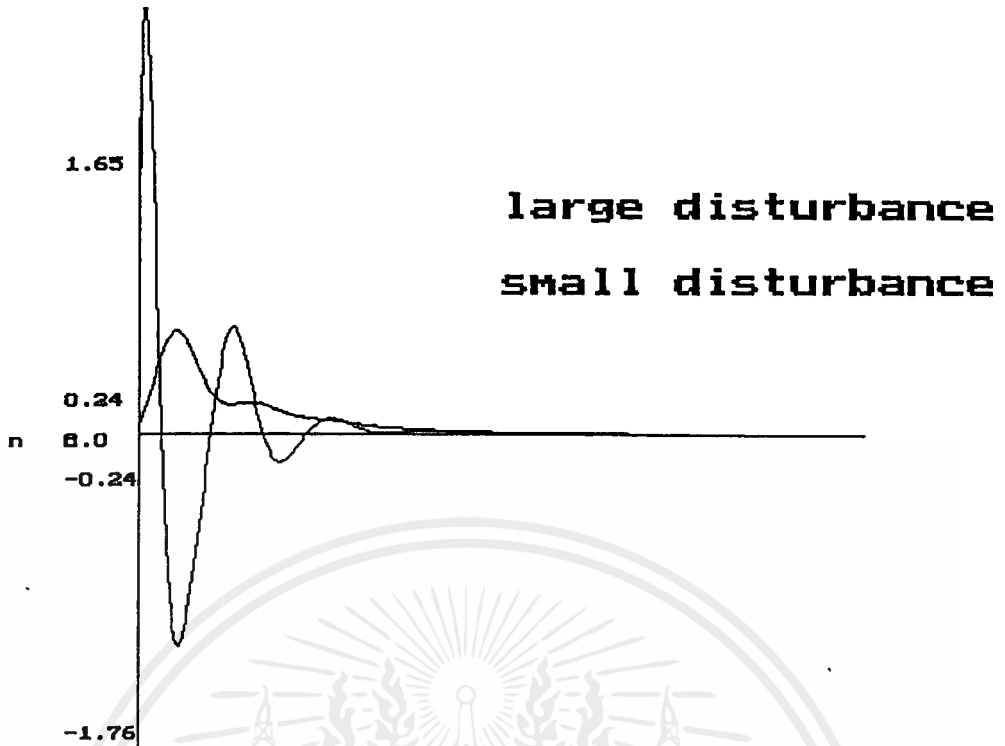
เมื่อทำการพิจารณาจากกราฟ สำหรับระบบที่ 1 ซึ่งเป็นระบบที่เล็ก เมื่อทำการใส่ตัวควบคุมโดยกฎพื้นฐานเข้าไปในระบบ จะไม่เห็นผลของการเปลี่ยนแปลงมากนัก เพียงแต่ลักษณะการแกว่งของกราฟก่อนเข้าสู่เสถียรภาพจะน้อยลง และเมื่อทำการเปลี่ยนเป็นระบบที่ 2 โดยการตั้งค่าพารามิเตอร์สำหรับเครื่องจักรที่มีขนาดใหญ่ ผลของตัวควบคุมโดยกฎพื้นฐานจะทำให้การแกว่งของกราฟก่อนเข้าสู่เสถียรภาพจะไม่มี ความแตกต่างมากนัก และเมื่อทำการทดลองรวมการควบคุมของและการควบคุมโดยกฎพื้นฐาน ลักษณะของกราฟจะมีลักษณะเรียบมากขึ้น แต่จุดเสถียรภาพของระบบ จะเลื่อนต่ำลง



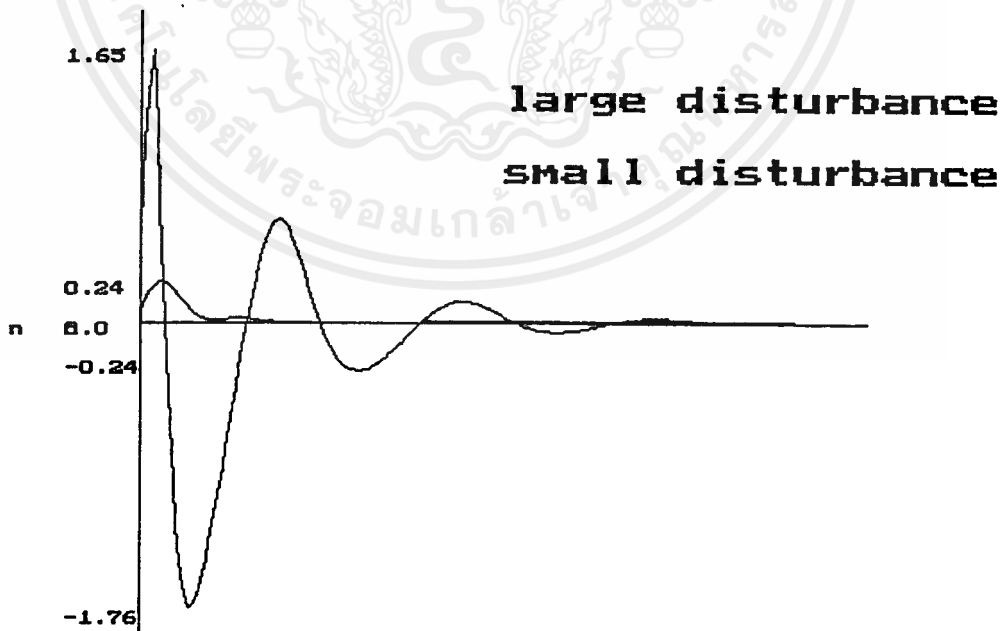
รูปที่ 5-1 ระบบที่ 1 ซึ่งมีการควบคุมโดยอุปกรณ์ควบคุมแรงดัน

และอุปกรณ์ควบคุมความเร็วรอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาและอุปกรณืควบคุมความเร็วนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

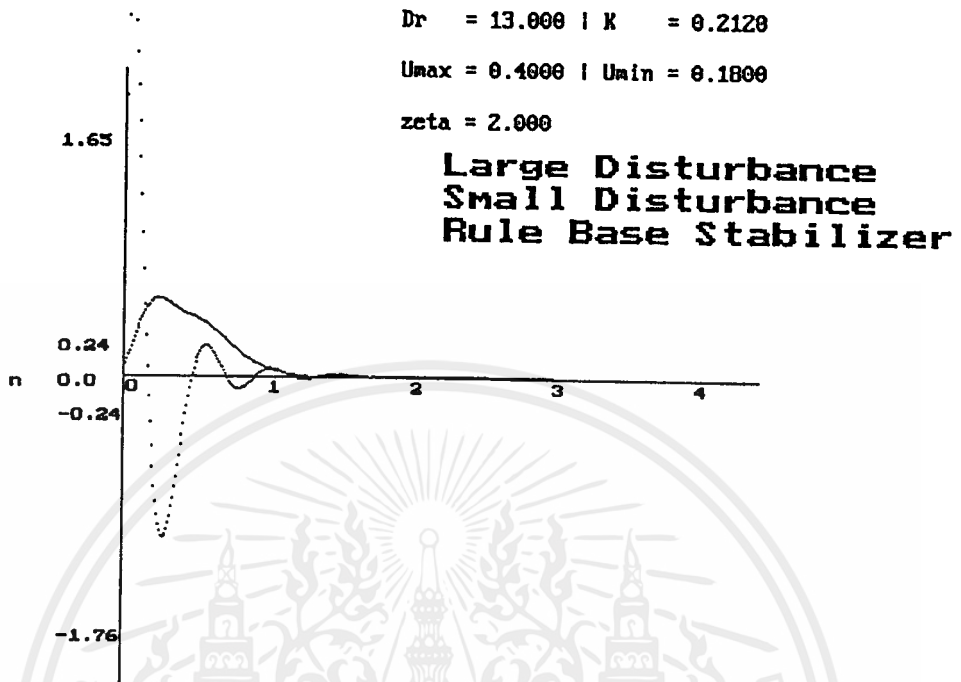


รูปที่ 5-2 ระบบที่ 2 ซึ่งมีการควบคุมโดยอุปกรณ์ควบคุมแรงดัน
และอุปกรณ์ควบคุมความเร็วรอบ

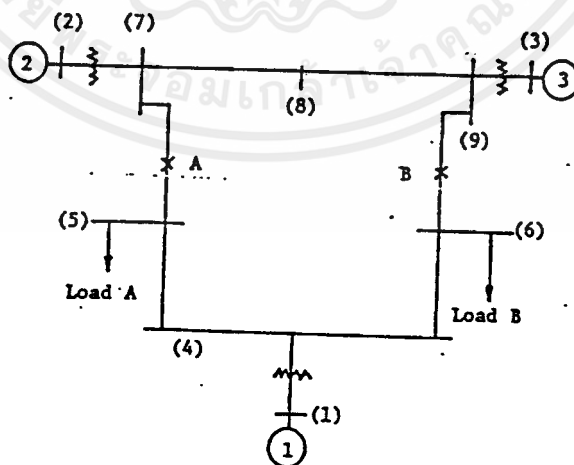


รูปที่ 5-3 ระบบที่ 1 ซึ่งมีการควบคุมโดยตัวควบคุมเสถียรภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-4 ระบบที่ 2 ซึ่งมีการควบคุมโดยตัวควบคุมเสถียรภาพ

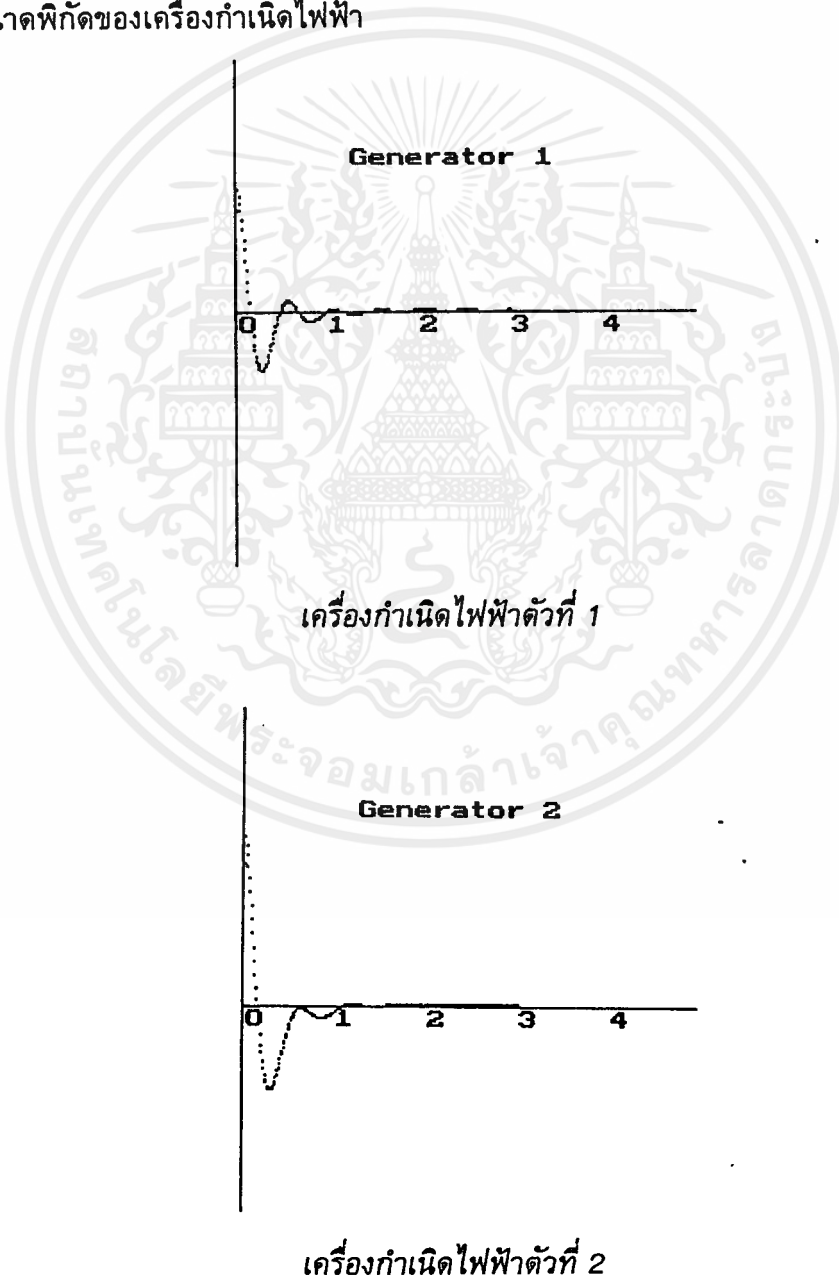


รูปที่ 5-5 ระบบจำลองที่ใช้ในการวิเคราะห์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

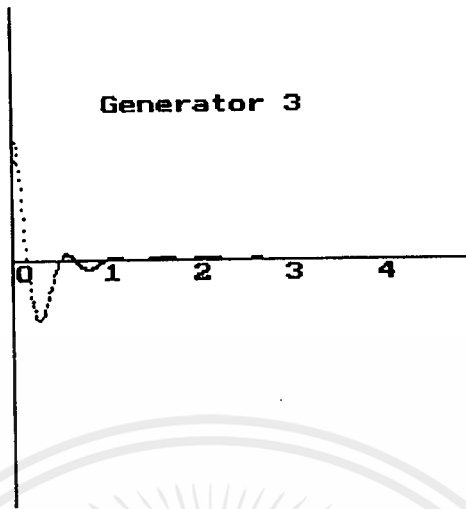
เมื่อนำโปรแกรมไปใช้ควบคุมระบบจำลองที่มีเครื่องจักรหลาย ๆ ตัว (Multi-Machine) เครื่องซึ่งได้แสดงขั้นตอนการทำงานของโปรแกรมดังในรูปที่ 5-6 ซึ่งจะทำให้เครื่องจักรสามารถจ่ายกำลังได้เหมาะสมกับภาระ โดยที่ค่ากำลังไฟฟ้าจริงและกำลังไฟฟ้าจินตภาพจะมีค่าไม่คงที่ โดยจะใช้การคำนวณหาค่ากำลังไฟฟ้าจริงและค่ากำลังไฟฟ้าจินตภาพจากการไหลของภาระหรือการไหลของกำลัง จะทำให้ได้กราฟดังรูปที่ 5-7

เมื่อทำการพิจารณาจากกราฟในรูปที่ 5-7 จะเห็นว่าเครื่องกำเนิดไฟฟ้าสามารถเข้าสู่สภาวะเสถียรได้เมื่อเกิดการผิดปกติ ซึ่งภาระของเครื่องกำเนิดไฟฟ้าจะต้องมีความเหมาะสมกับขนาดพิกัดของเครื่องกำเนิดไฟฟ้า



รูปที่ 5-7 แสดงผลการคุมเครื่องกำเนิดไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



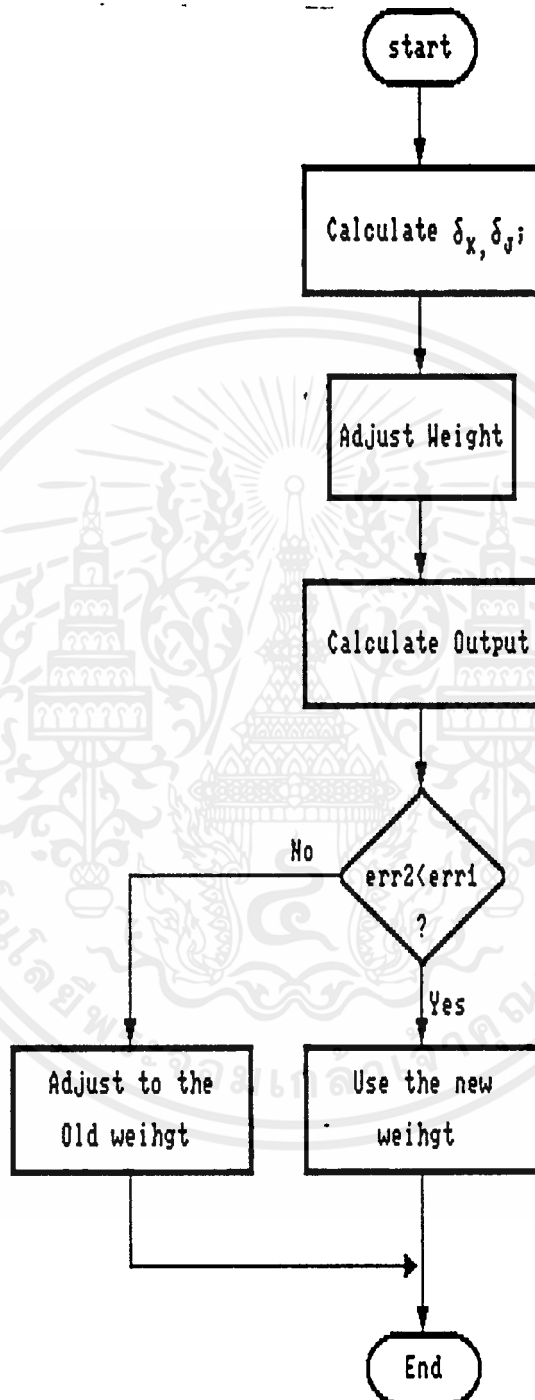
เครื่องกำเนิดไฟฟ้าตัวที่ 3
รูปที่ 5-7(ต่อ) แสดงผลการคุมเครื่องกำเนิดไฟฟ้า

การพัฒนาทฤษฎีนิรอรลเพื่อนำไปใช้เพื่อนำไปใช้ควบคุมระบบไฟฟ้า ได้ทำการฝึกโครงตาข่ายนิรอรลในรูปแบบการกระจายแบบย้อนกลับให้มีการปรับน้ำหนักเลียนแบบการทำงานของตัวควบคุมเสถียรภาพโดยกฎพื้นฐานซึ่งขั้นตอนในการฝึกโปรแกรมจะใช้เวลาในการฝึกนานประมาณ 100 ชั่วโมง ซึ่งในการฝึกบางครั้งก็เกิดความล้มเหลว เนื่องจากมีการตั้งขนาดในแต่ละชั้น η ที่สูงเกินไปมีผลทำให้เกิดความไม่ต่อเนื่องและไม่สมบูรณ์ นอกจากนี้ในการฝึกจะต้องทำการฝึกอย่างต่อเนื่องเพื่อให้โครงตาข่ายจำการฝึกในครั้งที่แล้วได้

		System 1	System 2
Generator Constant			
Inertia constant	M	0.0191 s	0.0314 s
Damping coefficient	D	0.00531 pus	0.01 pus
Time constant of d-axis circuit	Tdo	5.6 s	7.76 s
d-axis synchronous reactance	Xd	0.905 p.u.	0.973 p.u.
d-axis transient reactance	Xd'	0.25 p.u.	0.19 p.u.
q-axis synchronous reactance	Xq	0.550 p.u.	0.550 p.u.
AVR constants			
AVR gain	Ka	30	50
AVR time constant	Ta	0.1 s	0.1 s
Governor constants			
Governor gain	Kg	0.02	0.027
Governor time constant	Tg	0.1 s	0.1 s
Turbine time constant	Tt	0.2 s	0.3 s
Line constants			
Line resistance	R	0.03 p.u.	0.02 p.u.
Line reactance	X	0.6 p.u.	0.26 p.u.

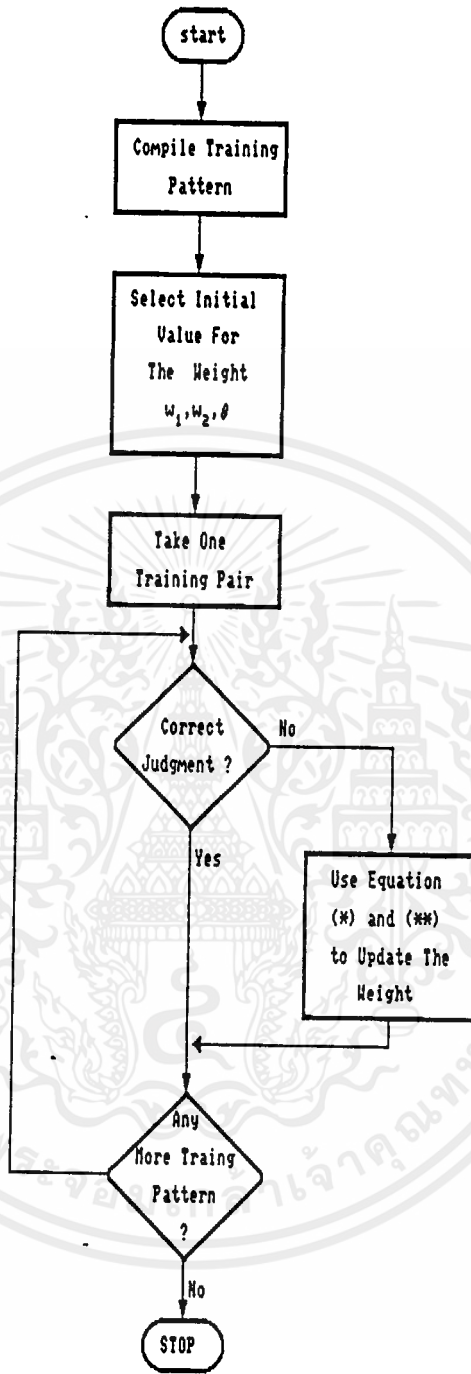
รูปที่ 5-8 แสดงค่าคงที่ซึ่งใช้ในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-9 แสดงขั้นตอนการปรับน้ำหนักของทฤษฎีโครงตาข่ายนิรอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



$$\text{Equ. } \bar{w}(1) = [w'_1 \ w'_2] \quad (*)$$

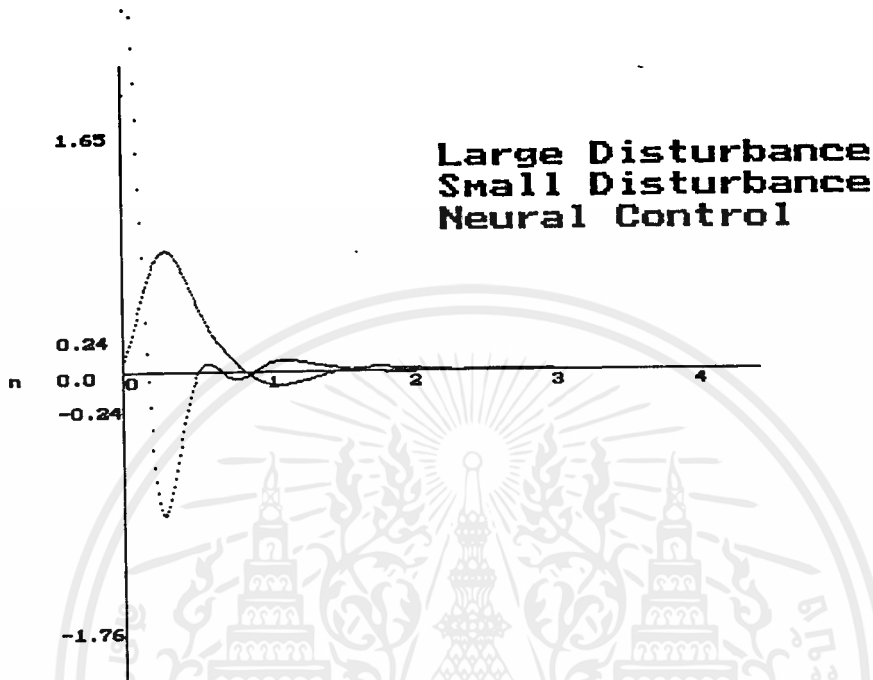
$$= \bar{w}(0) + \eta \bar{X}_p$$

$$\theta(1) = \theta(0) - \eta \bar{X}_p \bar{X}_p \quad (**)$$

รูปที่ 5-10 แสดงขั้นตอนการฝึกของทฤษฎีโครงตาข่ายนิรอร

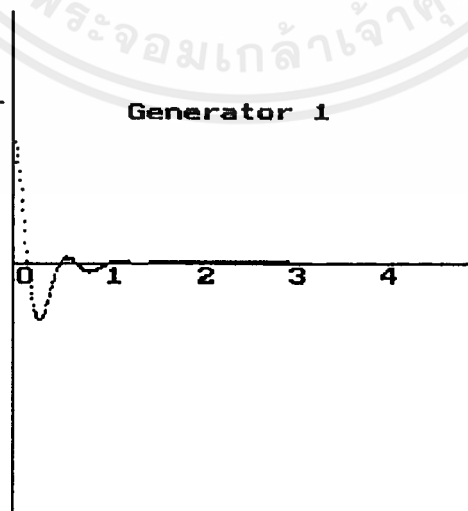
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การนำเอาโปรแกรมที่ได้ผ่านการฝึกแล้วไปใช้ควบคุมระบบจำลองที่มีการต่อกับบัลลจํากัด จะเห็นว่ามี การควบคุมดังรูปที่ 5-11



รูปที่ 5-11 แสดงผลการนำเอา ANN ผ่านการฝึกแล้วไปใช้ควบคุมเครื่องกำเนิดไฟฟ้า

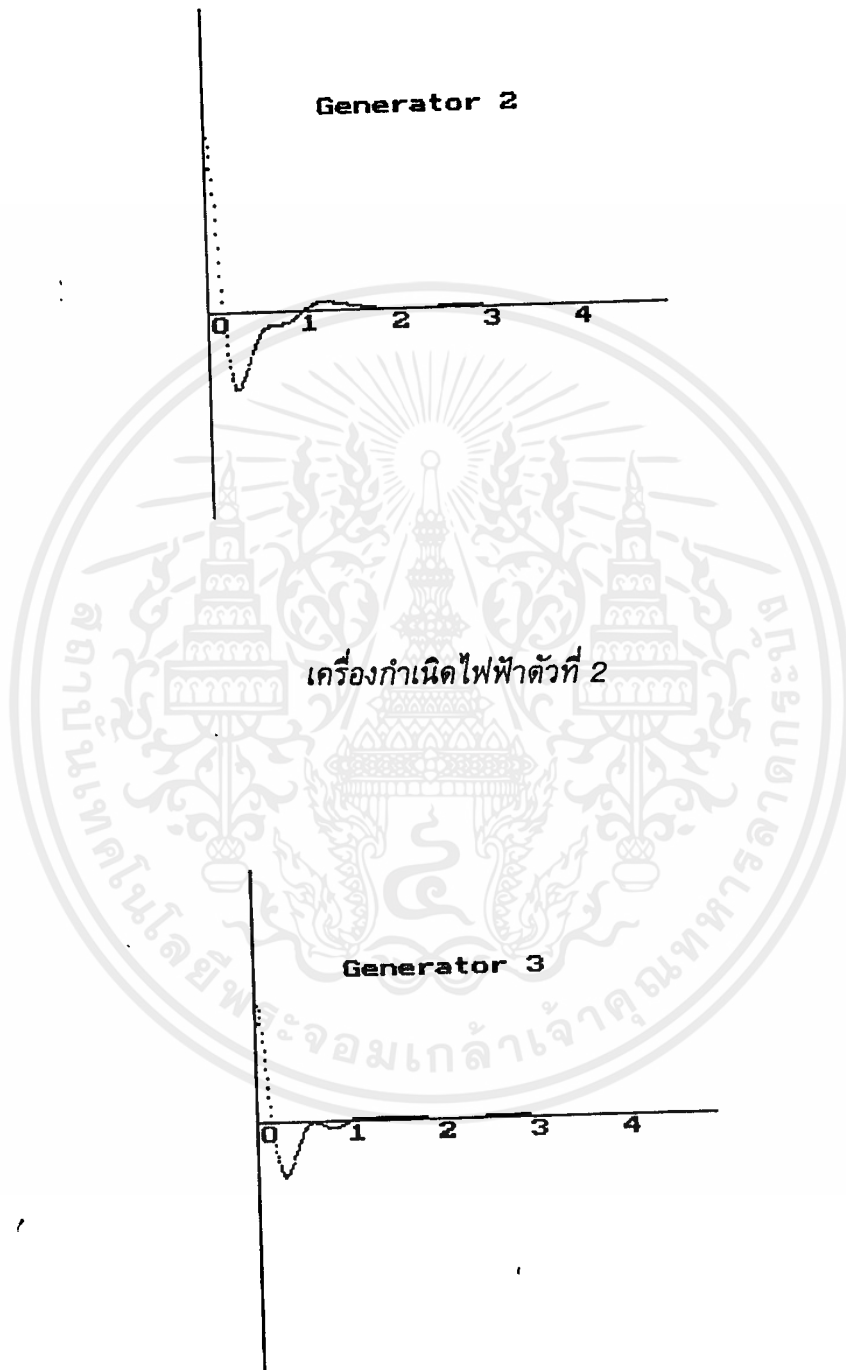
การนำเอาโปรแกรมที่ได้ผ่านการฝึกแล้วไปใช้ควบคุมระบบจำลองที่มีเครื่องจักรหลายเครื่องจักร จะเห็นว่ามี การควบคุมดังรูปที่ 5-12



เครื่องกำเนิดไฟฟ้าตัวที่ 1

รูปที่ 5-12 แสดงผลการนำเอา ANN ผ่านการฝึกแล้วไปใช้ควบคุมระบบจำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เครื่องกำเนิดไฟฟ้าตัวที่ 3

รูปที่ 5-12 (ต่อ) แสดงผลการนำเอา ANN ผ่านการฝึกแล้วไปใช้ควบคุมระบบจำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

สรุปผล และข้อเสนอแนะ

การควบคุมด้วย GOV และ AVR จะทำให้ระบบที่ถูกรบกวนกลับเข้าสู่สภาวะเสถียรได้อย่างรวดเร็ว

ตัวควบคุมระบบโดยใช้กฎพื้นฐาน (Rule-Based Power System Stabilizer) ที่ได้เสนอไปสามารถนำมาใช้สร้างเสถียรภาพให้กับระบบได้ กับจุดทำงานในช่วงกว้างของเครื่องกำเนิดไฟฟ้า ทั้งในกรณีของการรบกวนระบบอย่างและอย่างน้อย (Large and Small Disturbance) เราสามารถตั้งพารามิเตอร์ของตัวควบคุมได้โดยอิสระโดยไม่เกี่ยวข้องกับจุดทำงานของเครื่องกำเนิดไฟฟ้า เพื่อที่จะทำให้ได้ค่าที่เหมาะสมที่สุด สำหรับตัวควบคุมก็ได้เสมอไว้สามารถสร้างได้โดยใช้คณิตกรรม และฟังก์ชันของการเปลี่ยน A/D และ D/A ซึ่งเป็นอุปกรณ์ที่มีโครงสร้างง่าย ๆ และได้มีการพัฒนาให้สามารถควบคุมระบบที่มีเครื่องจักรหลายตัวได้อีกด้วย

จากผลการจำลองตัวควบคุมระบบโดยใช้กฎพื้นฐานจะเหมาะสมกับระบบที่มีขนาดใหญ่ แต่ถ้าเป็นระบบขนาดเล็กการใช้การควบคุมโดย GOV และ AVR จะไม่เห็นความแตกต่างกับตัวควบคุมระบบโดยใช้กฎพื้นฐานมากนัก

ในส่วนที่เราได้นำเอาทฤษฎีโครงตาข่ายนิเวศแบบการกระจายแบบย้อนกลับมาประยุกต์ใช้งานในการควบคุมระบบยังคงมีปัญหาหลายประการคือ ขั้นตอนที่ยาวนานในการฝึกและความไม่แน่นอนในการฝึก ซึ่งปัญหาที่ซับซ้อนเหล่านี้บางทีอาจจะต้องใช้ระยะเวลาในการฝึกนานเป็นวันหรืออาจจะใช้ระยะเวลาเป็นสัปดาห์ ซึ่งการใช้ระยะเวลาในการฝึกที่ยาวนานนี้จะทำให้ได้ค่าน้ำหนักที่เหมาะสม ในส่วนของความล้มเหลวที่เกิดจากการฝึกนั้นอาจจะมาจากสาเหตุที่สำคัญได้แก่ ความไม่สมบูรณ์ของทฤษฎีโครงตาข่าย (Paralysis Network) และโลคอลมินิมา (Local Minima)

สำหรับความไม่สมบูรณ์ของโครงตาข่ายเราสามารถอธิบายได้ดังนี้

ในการฝึกโครงตาข่าย เราจะทำการปรับค่าน้ำหนักอย่างมากซึ่งจะเป็นการกำหนดให้นิเวศทำงานที่ค่ามาก ๆ ของ OUT และในกรณีที่ค่าอนุพันธ์ของฟังก์ชันสควอชซิง (Squashing Function) มีค่าน้อย ๆ ขั้นตอนการฝึกจะมีการเปลี่ยนแปลงน้อยเพราะ การลดลงของขนาดในแต่ละขั้น (Step Size) จะมีผลทำให้เวลาในการฝึกมากขึ้น จากบทพิสูจน์ของ Rumelheart, Hinton และ Williams(1986) แสดงให้เห็นว่าในการเลือกขนาดในแต่ละขั้นจะต้องใช้ประสิทธิภาพในการเลือก ถ้าขนาดในแต่ละขั้นมีค่าน้อยเกินไป การลู่เข้าจะเป็นไปอย่างช้ามาก แต่ถ้าขนาดมีค่ามากเกินไปจะทำให้เกิดผลที่ไม่สมบูรณ์และไม่มีเสถียรภาพที่ต่อเนื่อง หรืออีกอาจจะเกิดปัญหาการไม่มีเสถียรภาพในช่วงเวลา (Temperal Stability) โดยขั้นตอนในการ

สอนโครงตาข่ายให้เกิดการเรียนรู้กลุ่มที่ต้องการฝึกนั้น จะต้องมีความต่อเนื่องไม่มีการขาดตอน อีกปัญหาหนึ่งที่ได้พบคือการเลือกตัวอย่างที่ใช้การสุ่มมานั้นจะไม่ทราบว่าเป็นตัวอย่างที่เหมาะสมที่จะนำมาใช้ในการฝึก ซึ่งถ้าใช้ตัวอย่างทั้งหมดจะทำให้เสียเวลาในการฝึกมาก โดยจริง ๆ แล้วไม่มีความจำเป็นต้องนำตัวอย่างทั้งหมดมาใช้ในการฝึก อย่างไรก็ตามอาจจะพบปัญหาอื่นอีกแต่ที่ได้อธิบายมาเป็นปัญหาที่มักจะพบบ่อยเท่านั้น



ภาคผนวก 1

ในการเรียนโครงตาข่าย เมื่อต้องการพิจารณาค่าผิดพลาด จะพิจารณาดังนี้

$$E_p = \frac{1}{2} \sum_k (t_{pk} - o_{pk})^2 \quad (\text{A.1})$$

t_{pk} คือ เป้าหมายตัวที่ k ตัวอย่างที่ p

o_{pk} คือ เอาท์พุทตัวที่ k ตัวอย่างที่ p

และค่าผิดพลาดเฉลี่ยของระบบ ก็คือ

$$E = \frac{1}{2P} \sum_p \sum_k (t_{pk} - o_{pk})^2 \quad (\text{A.2})$$

ที่มาของสมการปรับน้ำหนักมีดังต่อไปนี้

$$\Delta w_{kj} = -\eta E/dw_{kj} \quad (\text{A.3})$$

อย่างไรก็ตามเราสามารถเขียนค่า E ซึ่งเป็นค่าความคลาดเคลื่อนให้อยู่ในรูปของค่าเอาท์พุท o_k ซึ่งเป็นค่าเอาท์พุทที่ไม่เป็นเชิงเส้นของโนด k ได้เป็น

$$\text{net}_k = f(\text{net}_k) \quad (\text{A.4})$$

โดยที่ net_k คือค่าอินพุทที่ป้อนเข้าไปที่โนด k th ซึ่งก็คือค่าผลรวมน้ำหนักของค่าเอาท์พุทจากชั้นที่แล้ว

$$\text{net}_k = \sum w_{kj} o_j \quad (\text{A.5})$$

ซึ่งค่าอนุพันธ์ dE/dw_{kj} สามารถหาได้โดยใช้กฎลูกโซ่ (Chain Rule)

$$dE/dw_{kj} = (dE/d\text{net}_k)(d\text{net}_k/dw_{kj}) \quad (\text{A.6})$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และเมื่อใช้สมการที่ 3 เราจะได้

$$dnet_k/dw_{kj} = d/dw_{kj} (\sum w_{kj}o_j) = o_j \quad (A.7)$$

และเมื่อกำหนด

$$\delta_k = -dE/dnet_k \quad (A.8)$$

ทำให้ได้

$$\Delta w_{kj} = \eta \delta_k o_j \quad (A.9)$$

เพื่อที่จะคำนวณค่าของ $\delta_k = -dE/dnet_k$ เราใช้กฎลูกโซ่ เพื่อแสดงอนุพันธ์ในรูปของ 2 แฟลคเตอร์ ซึ่งค่าแฟลคเตอร์แรกจะแสดงค่าของอัตราการเปลี่ยนของค่าความเคลื่อนไหวเมื่อเปรียบเทียบกับค่าเอาท์ o_k ส่วนอีกแฟลคเตอร์จะแสดงถึงอัตราการเปลี่ยนของค่าเอาท์พุทของโนด k เมื่อเทียบกับค่าอินพุทที่โนดเดียวกัน เราจึงได้

$$\delta_k = -dE/dnet_k \quad (A.10)$$

$$= -(dE/do_k)(do_k/dnet_k) \quad (A.11)$$

และสามารถหาค่าของแฟลคเตอร์ทั้งสองได้ดังนี้

$$dE/do_k = -(t_k - o_k) \quad (A.12)$$

$$do_k/dnet_k = f'_k(net_k) \quad (A.13)$$

และจากสมการข้างต้นทำให้เราได้

$$\delta_k = (t_k - o_k) f'_k(net_k) \quad (A.14)$$

และสำหรับชั้นเอาร์ทพุทโนด k ใด ๆ เราจะได้

$$\begin{aligned}\Delta w_{kj} &= \eta(t_k - o_k) f'_k(\text{net}_k) o_j \\ &= \eta \delta_k o_j\end{aligned}\quad (\text{A.15})$$

ถ้าค่าน้ำหนักไม่มีผลเอาร์ทพุทโนดโดยตรงเราจะได้

$$\begin{aligned}\Delta w_{kj} &= -\eta (dE/dw_{kj}) \\ &= -\eta (dE/d\text{net}_k)(d\text{net}_j/dw_{kj}) \\ &= -\eta (dE/d\text{net}_j) o_j \\ &= \eta (-dE/do_j * do_j/d\text{net}_j) o_j \\ &= \eta (-dE/do_j) f'_j(\text{net}_j) o_j\end{aligned}\quad (\text{A.16})$$

อย่างไรก็ตามค่าแฟคเตอร์ dE/do_j ไม่สามารถหาค่าได้โดยตรง เราจึงเขียนให้อยู่
ในรูปของปริมาณซึ่งสามารถที่จะค่าออกมาได้ จะได้

$$\begin{aligned}-dE/do_j &= -\sum (d\text{net}_k/do_j)(dE/d\text{net}_k) \\ &= \sum (-dE/d\text{net}_k) * d\sum w_{km} o_m / do_j \\ &= \sum (-dE/d\text{net}_k) w_{kj} = \sum \delta_k w_{kj}\end{aligned}\quad (\text{A.17})$$

เราจะเห็นได้ว่า

$$\delta_j = f'_j(\text{net}_j) \sum \delta_k w_{kj}\quad (\text{A.18})$$

ซึ่งก็คือ ค่าเดลต้าที่โนดภายในสามารถที่จะหาค่าออกมาได้ในรูปของค่าเดลต้าที่ชั้น
สูงกว่า (Upper Layer) ดังนั้นเมื่อเริ่มที่ชั้นที่สูงที่สุด (ชั้นของเอาร์ทพุท) เราจะสามารถหาค่า
ของ δ_k และสามารถ (Propagate) ค่าความคลาดเคลื่อนกลับไปสู่ชั้นต่ำกว่า และเมื่อใช้ p
เพื่อแสดงรูปแบบของจำนวนเราจะได้ว่า

$$\Delta p_{w_{ji}} = \eta \delta_j o_{pj}\quad (\text{A.19})$$

ถ้าโนด j เป็นโนดของชั้นเอาร์ทพุทเราจะได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\delta_{pj} = (t_{pj} - o_{pj}) f'_j(\text{net}_{pj}) \quad (\text{A.20})$$

อย่างไรก็ตามถ้าโนด j เป็นโนดภายใน เราจำเป็นต้องหาค่า δ_{pj} ในรูปของ δ_s ที่โนดที่สูงกว่า ซึ่งก็คือ

$$\delta_{pj} = f'_j(\text{net}_{pj}) \sum (\delta_{pk} w_{kj}) \quad (\text{A.21})$$

และถ้า

$$o_j = 1 / (1 + \exp(-(\sum_i w_{ji} o_i + \theta_j))) \quad (\text{A.22})$$

ดังนั้น

$$do_j/d\text{net}_j = o_j(1 - o_{pk}) \quad (\text{A.23})$$

ซึ่งสามารถหาค่าเคลต้าโดย

$$\delta_{pk} = (t_{pk} - o_{pk}) o_{pk}(1 - o_{pk}) \quad (\text{A.24})$$

$$\delta_{pj} = o_{pj}(1 - o_{pj}) \sum \delta_{pk} w_{kj} \quad (\text{A.25})$$

สำหรับส่วนของชั้นเอาต์พุตและชั้นภายในตามลำดับและสามารถเรียนค่าทริสโพลได้เหมือนกับการเรียนค่าน้ำหนักอื่น ๆ โดยเราจะคิดอย่างง่าย ๆ ว่า θ_j เป็นค่าน้ำหนักจากหน่วยที่มักจะให้ค่าเอาต์พุตเป็น 1

ภาคผนวก 2

Main Program Part

```

/*-----
I E E Nonlinear model system - One machine infinite bus
Condition Variables
x 0 : e Torque angle
x 1 : w Frequency
x 2 : f f Field flux
x 3 : v f Exciter voltage
x 4 : g Gate Opening
x 5 : g f Governer Feedback Loop signal
x 6 : h

Input Variables
u 0 : G O V input
u 1 : A V R input

Output Variables
y 0 : P e Real Power
y 1 : V t Terminal Voltage
y 2 : Q e Reactive voltage

#include <stdio.h>
#include <stdlib.h>
#include <float.h>
#include <math.h>
#include <graphics.h>
#include <conio.h>
#include <b:pro21.c>

#define Ue 50.00 /* Exciter gain */
#define Te 0.1 /* Exciter time constant */
#define SIGM 0.05 /* Permanent drop */
#define Tg 0.010 /* Gate time constant */
#define Ta 0.010 /*Governor actuator time constant*/
#define Tw 0.3
#define Tf 0.1
#define Vt0 1.0 /* Terminal Voltage */
#define W0 377.0 /* Synchronous speed */
#define BETA 1.0 /* Time scaling factor */
#define BETA2 BETA*BETA
#define Xdd ((1.0-xe*B)*xdd+xe) /* Transmission reactances */
#define Xq ((1.0-xe*B)*xd+xe)
#define Xd ((1.0-xe*B)*xd+xe)
#define x10 1.0*xl /* short circuit impedance */
#define T 0.01 /* Sampling time */
#define N 301 /* Number of data */
#define LIMIT T*(DOUBLE)(N-1) /* Max. number of data */
#define NA 8 /*Number of differential equation*/

/*----- Initial condition variables -----*/

```

```

static double vd0, vq0, id0, iq0, del0,
                v0, fif0, vf0, xe,
                B = -0.1, /* System susceptance */
                xt = 0.6, /* Transmission reactance */
                xl = 0.14,
                xee = 0.05;
static double xd = 0.973, /* d axis reactance */
                xdd = 0.190, /* d axis transient reactance*/
                xq = 0.550, /* q axis reactance */
                Td0d = 7.760, /*d axis Open circuit time
constant*/
                M = 0.0191, /* Inertia constant */
                D = 0.01; /* Damping coefficient */

static double u1[2][N],
                dy1[N],
                dy3[N],
                ti;

static double Pd; /* Change of Torque angle */

/*-- new parameter --*/

/*static double K=0.002;*/ /* for phase shifter equipment */
static double Jn,K; /* for phase shifter equipment */

static int DD1,DD2,DD3,DXD1,DXD2,DXD3;
static int AVRL;

int kpt,CH,vy (double),jj;
float P0,Q0,pp[6],qq[6];
void d_out(),
        runge4(),
        sys1(),
        x_stand(),
        shoki();

unsigned _stack = 50000;

/*-----*
                main Program
*-----*/
void main()
{
    static double GD;
    static double x1[NA], y1[3], f[NA];
    char buf[15];
    int jo;
    float vx;
    float x,y,cx,sx;
    double time[N],freq[N];
    int device = DETECT,mode;
    loadflow();

    /*----- Open file for Output -----*/
    initgraph (&device,&mode,NULL); /*initial the graphic system*/

```

```

{
    P0 = pp[jo];
    Q0 = qq[jo];
    for (K=0.412;K<0.414;K+=0.002)
    {
/*----- Disturbance Large (0,1) , Small(1,2) -----*/
        for (CH=0;CH<1;CH+=1)
        {
            xe = xt + xl + xee;
            if(CH==1)
                Pd = 0.05;
            else
                Pd = 0.0;
/*----- Initial Condition of System -----*/
            shoki();
/*----- Initial Condition of State Equation -----*/
            x_stand(x1);
/*----- Graphic -----
            if(CH<1)
            {
                outtextxy(60,-1.65*90+230, "1.65");
                outtextxy(60,230,"0.0");
                outtextxy(30,230,"n");
                outtextxy(60,1.76*90+230,"-1.76");
                settextstyle(0,0,2);
                outtextxy(300,100, "large disturbance");
                moveto(100,-40.7+230);
                settextstyle(0,0,1);
            }
            else
            {
                outtextxy(60,-0.24*90+230, "0.24");
                outtextxy(60,230,"0.0");
                outtextxy(60,230,"n");
                outtextxy(60,0.24*90+230,"-0.24");
                settextstyle(0,0,2);
                outtextxy(300,140, "small disturbance");
                moveto(100,-0.06*90+230);
                settextstyle(0,0,1);
            }
        }
/*--- short circuit 3 phase to ground ---*/
        if(CH<1)
        {
            xe=( (xt+0.5*x10) * (xee+0.5*(2.0*xl-x10)) * (4.0*xl)+
                (xt+0.5*x10) * (x10*(2.0*xl-x10)) +
                (xee+0.5*(2.0*xl-x10)) * (x10*(2.0*xl-
                x10)))/(x10+(2.0*xl-x10));
            B = -((xt+0.5*x10) * (xee+0.5*(2.0*xl-x10)) *
                (4.0*xl)+
                (xt+0.5*x10) * (x10*(2.0*xl-x10)) +
                (xee+0.5*(2.0*xl-x10)) * (x10*(2.0*xl-
                x10)))/(xee+(0.5*(2.0*xl-x10)));
            B = 1.0/B;
        }
/*--- on line 1 ---*/
        for (kpt = 0;kpt <201; kpt++)

```

```

ti = (double)kpt * T;
if(CH==0)
{
    if(kpt==5)
    {
        xe = x1+2.0*x1+xee ;
        B = 0.0;
    }
}
DD1=DXD1;
DD2=DXD2;
DD3=DXD3;

/*--- Phase plane Control ---*/
phase_plane(x1 ,u1);
/*--- System state equations calculation using Rungekutta ---*/
runge4(x1, u1, T, NA, dy3);
/*--- Limiter of A V R ---*/
if(x1[3]>3 * vf0)
{
    AVRL = 1;
    x1[3] = 3 * vf0;
}
else
if(x1[3]<-2 * vf0)
{
    AVRL = -1;
    x1[3] = -2 * vf0;
}
else
    AVRL = 0;

/*--- Limiter of G O V ---*/
GD = (-x1[1]/W0-x1[6]-SIGM*x1[5])/Tg;
if(GD>0.1)
    DXD1=1;
else
if(GD<-0.1)
    DXD1=-1;
else
    DXD1 = 0;
if(x1[5]>(1.05-P0))
    DXD2=1;
else
if(x1[5]<-P0)
    DXD2=-1;
else
    DXD2 = 0;
if(x1[7]>20.0)
    DXD3=1;
else
if(x1[7]<-20.0)
    DXD3=-1;
else
    DXD3=0;
sys1(x1, u1, f, y1, dy3);
/*--- Calculation of Deviation of Output ---*/
d_out(y1);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อ d_out(y1); เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*--- Graphic ---*/
/* gen1 @*/
if(CH<1 && jo == 0)
{
    axis1();
    putpixel((ti*40+100),(x1[1]*50+130),3);
}
/*draw line of graph*/
/* gen2 */
if(CH<1 && jo == 1)
{
    axis2();
    putpixel((ti*40+100),(x1[1]*50+390),3);
}
/*draw line of graph*/
/* gen3 @*/
if(CH<1 && jo == 2)
{
    axis3();
    putpixel((ti*40+350),(x1[1]*50+130),3);
}
/*draw line of graph*/
/* gen4 @*/
if(CH<1 && jo == 3)
{
    axis4();
    putpixel((ti*40+350),(x1[1]*50+390),3);
}
/*draw line of graph*/
}
Jn = Jn + fabs(x1[1]) * ti;
}
Jn = 0.0;
}
}
}
getch();
closegraph(); /*shut down the graphic system*/
for (i=0;i<=20;i++)
    printf("          I love you so much.\n");
getch();
/*          listvalue(time,freq);
savefile(time,freq); */
}

/*-----*
Initial Condition of state variables
*-----*/
void x_stand(x)
double x[7];
{
    x[0] = del0 + Pd;
    x[1] = 0.0;
    x[2] = fif0;
    x[3] = 0.0;
    x[4] = 0.0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        x[5] = 0.0;
        x[6] = 0.0;
        x[7] = 0.0;
    }
    /*-----*
        Deviation of Output
    *-----*/
void d_out(y1)
double y1[];
{
    dy1[kpt+1] = y1[0]-P0;
    dy3[kpt+1] = y1[1]-Vt0;
}
/*-----*
    System Equations
    *-----*/
void sys1(x, u, f, y, dy)
double x[], u[][N], f[], y[], dy[N];
{
    double vd, /* d axis voltage */
            vq, /* q axis voltage */
            id, /* d axis current */
            iq; /* q axis current */

    /*---
    Operating Point Calculations ---*/
    /*---
    d axis voltage Calculations ---*/
        vd = xq * v0 * sin((x[0]-x[7])) / Xq;
    /*---
    q axis voltage Calculations ---*/
        vq = (xe * x[2]) / (Xdd * Td0d)
            + xdd * v0 * cos((x[0]-x[7])) / Xdd;
    /*---
    d axis current Calculations ---*/
        id = (1.0 - xe * B) * x[2] / (Xdd * Td0d)
            - v0 * cos((x[0]-x[7])) / Xdd;
    /*---
    q axis current Calculations ---*/
        iq = v0 * sin((x[0]-x[7])) / Xq;
    /*---
    Real Power Calculations ---*/
        y[0] = vd * id + vq * iq;
    /*---
    Terminal Voltage Calculations ---*/
        y[1] = sqrt(vd * vd + vq * vq);
    /*---
    Reactive Power Calculations ---*/
        y[2] = vq * id - vd * iq;
    /*---
    Nonlinear differential equations ---*/
    /*---
    0 : ð ---*/
        f[0] = x[1];

```

```

f[1] = 1.0 / M * (x[6] - D * BETA * x[1]
               - (v0 * sin((x[0]-x[7])) * x[2] / (Xdd * Td0d)
               + (xdd - xq) * v0 * v0 * sin(2.0 * (x[0]-x[7]))
               / (2.0 * Xdd * Xq) - P0));
/*--- 2 : if ---*/
switch(AVRL)
{
    case 1 : f[2] = (4 * vf0) / BETA - Xd * x[2] / (BETA * Xdd *
Td0d)
               + (xd-xdd) * v0 * cos((x[0]-x[7])) / (BETA * Xdd);
               break;
    case 0 : f[2] = (vf0 + x[3]) / BETA - Xd * x[2] / (BETA * Xdd *
Td0d)
               + (xd-xdd) * v0 * cos((x[0]-x[7])) / (BETA * Xdd);
               break;

    case -1 : f[2] = (-vf0) / BETA - Xd * x[2] / (BETA * Xdd *
Td0d)
               + (xd-xdd) * v0 * cos((x[0]-x[7])) / (BETA * Xdd);
               break;
}
/*--- 3 : vf ---*/
f[3] = -x[3] / (BETA * Te)
       + Ue / (BETA * Te) * (-dy[kpt] - u[1][kpt]);
/*--- 5 : g ---*/
switch(DD1)
{
    case 1 : f[4] = 0.1;
               break;
    case 0 : f[4] = -SIGM * x[4] / (BETA * Tg)
               + 1.0 / (BETA * Tg) * (-BETA * x[1]
/W0 - x[5]);
               break;
    case -1 : f[4] = -0.1;
               break;
}

/*--- 6 : gf ---*/
f[5] = -x[5] / (BETA * Ta) + u[0][kpt];
/*--- 7 : delta Pi ---*/
switch(DD2)
{
    case 1 : f[6] = (-x[6] + (1.05 - P0)) / Tw;
               break;
    case 0 : f[6] = (-x[6] + x[4]) / Tw;
               break;
    case -1 : f[6] = (-x[6] - P0) / Tw;
               break;
}
switch(DD3)
{
    case 1 : f[7] = ((-K * x[1]) - 20.0) / Tf;
               break;
    case 0 : f[7] = ((-K * x[1]) - x[7]) / Tf;
               break;
    case -1 : f[7] = ((-K * x[1]) - 20.0) / Tf;
               break;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

break;
    }
}
/*-----*
                        Rungekutta Method
*-----*/
void runge4(x, u, t, n, dy)
double x[], u[][N], t, dy[N];
int n;
{
    double k1[NA], k2[NA], k3[NA],
           k4[NA], y[3], z[NA];
    int i;

    sys1(x, u, k1, y, dy);
    for(i=0;i<n;i++)
        z[i] = x[i] + t / 2 * k1[i];
    sys1(z, u, k2, y, dy);
    for(i=0;i<n;i++)
        z[i] = x[i] + t / 2 * k2[i];
    sys1(z, u, k3, y, dy);
    for(i=0;i<n;i++)
        z[i] = x[i] + t * k3[i];
    sys1(z, u, k4, y, dy);
    for(i=0;i<n;i++)
        x[i] = x[i] + t * (k1[i] + 2 * k2[i]
                        + 2 * k3[i] + k4[i]) / 6;
}
/*-----*
                        System Initial Conditions
*-----*/
void shoki()
{
    static double A, C;

    /*--- 1. v d 0 Calculations ---*/
    A = (Vt0 * Vt0 / xq + Q0);
    A = A * A;
    A = A + P0 * P0;
    vd0 = P0 * Vt0 / sqrt(A);

    /*--- 2. i q 0 Calculations ---*/
    C = Vt0 * Vt0 + xq * Q0;
    C = C * C;
    C = C + xq * xq * P0 * P0;
    iq0 = P0 * Vt0 / sqrt(C);

    /*--- 3. v q 0 Calculations ---*/
    A = Vt0 * Vt0 - vd0 * vd0;
    vq0 = sqrt(A);

    /*--- 4. i d 0 Calculations ---*/
    id0 = (P0 - vq0 * iq0) / vd0;

    /*--- 5. d e l 0 Calculations ---*/
    A = iq0 * xe + vd0 - vd0 * B * xe;
    C = -id0 * xe + vq0 - vq0 * B * xe;
    del0 = atan(A/C);

    /*--- 6. v 0 Calculations ---*/
    A = iq0 * xe + vd0 - vd0 * B * xe;
    C = sin(del0);
}

```

```

        v0 = A/C;
/*--- 7. f i f 0 Calculations ---*/
        fif0 = (xdd * v0 * cos(del0) / Xdd);
        fif0 = (vq0 - fif0) * Xdd * Td0d / xe;
/*--- 8. v f 0 Calculations ---*/
        vf0 = (Xd * fif0) / (Xdd * Td0d) -
              (xd -xdd) * v0 * cos(del0) / Xdd;
}

720*348 */

16 color 1 page */
void axis1()
{
    setbkcolor(1);
    setcolor(3);
    setlinestyle(0,0,0); /*set current line style in norm_width*/
    line(100,20,100,240); /*draw axis Y*/
    moveto(100,130);
    lineto(300,130); /*draw axis X*/
    setcolor(60);
}

void axis2()
{
    setcolor(3);
    setlinestyle(0,0,0); /*set current line style in norm_width*/
    line(100,260,100,520); /*draw axis Y*/
    moveto(100,390);
    lineto(300,390); /*draw axis X*/
    setcolor(60);
}

void axis3()
{
    setcolor(3);
    setlinestyle(0,0,0); /*set current line style in norm_width*/
    line(350,20,350,240); /*draw axis Y*/
    moveto(350,130);
    lineto(550,130); /*draw axis X*/
    setcolor(60);
}

void axis4()
{
    setcolor(3);
    setlinestyle(0,0,0); /*set current line style in norm_width*/
    line(350,260,350,520); /*draw axis Y*/
    moveto(350,390);
    lineto(550,390); /*draw axis X*/
    setcolor(60);
}

/*listvalue(tt,ff)
double tt[],ff[];
{
    int i;

```

```

static int c = 9, m = 1;
for(i = 0; i<201; i++)
{
    gotoxy(c,m++);printf("ti = %4.2f ->x1[1] = %4.2f",tt[i],ff[i]);
    if( m == 25)
    {
        c = c + 40;
        m = 1;
    }
    if( c>80)
    {
        getch();
        c = 9;
        m = 1;
        clrscr();
    }
}
}

void savefile(tt,ff)
double tt[],ff[];
{
    int i;
    FILE *fp;
    fp = fopen("da3","w");
    for (i = 0;i<201;i++)
    {
        fprintf(fp,"ti = %5.2e -> X1[1] = %5.2e /n",tt[i],ff[i]);
    }
    fclose(fp);
}

/*/

phase_plane(x,u)
double x[], u[][N];
{
    long float Dr,xx1,xx2,px,py,Dk,l1,l2,l3,J1,J2,J3,a,b,Gk,zeta,Umax,Umin,Uk;
    int i;
    Dr = 20;
    Umax = 0.035;
    Umin = 0.025;
    xx1 = xx2;
    xx2 = x[1] - 0;
    px = xx2;
    py = (xx2-xx1)/T;
    Dk = sqrt(px*px+py*py);
    if(Dk<Dr)
        Gk = Dk/Dr;
    if(Dk>=Dr)
        Gk = 1;

    /*find J1,J2
    for(i=0; i<=200;i++)
    {
        a = (x[0]-0);
        a = a * a;
        J1= a + J1;
        b = (ti*xx2);

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ไม่ว่ากรรมใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        b = b * b;
        J2= b + J2;
    }
    */

zeta = 0.1745;

l1 = -px;
/*l2 = -tan(90-zeta)*px;
l3 = -tan(zeta)*px;*/
if(py>l1 && px >0 && py>0) Uk = Gk*Umax;
if(py<0 && py >l1 && px>0) Uk = Gk*Umin;
if(px<0 && py <l1 && py<0) Uk = -Gk*Umin;
if(py<l1 && px <0 && py<0) Uk = -Gk*Umax;
if(py>0 && py <l1 && px<0) Uk = -Gk*Umin;
if(py>0 && py >l1 && px<0) Uk = Gk*Umin;
u[1][kpt] = Uk;
}

```

Load Flow Part

```

#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <stdlib.h>
#define radtodeg 57.29577951
typedef float busrange[6];
typedef float complex[6][6];
int n,i,j,v,m,k,L,jj;
busrange d,p,smr,q,smi,vr,vi,qmin,qmax,ar,ai,cr,pp,qq;
busrange ci,vcr,vci,vspec,ang,pl,ql,pg,qg,av;
complex y,ylr,yi,yli,br,sr,r,x,bi,si,bx;
int b[11];
float vtemp,r,vtempi,dvmax,dv,pb,P0,Q0;
int ee;
void setpow()
{
    int f;

    for (f=0; f <= n-1; f++) {
        pg[f]=0;qg[f]=0;
        if ((b[f]==1) || (b[f]==2)) {
            pg[f] =p[f]+pl[f];
            qg[f] =q[f]+ql[f];
        }
    }
}

void ragtopol()
{
    int g;

    for (g=0; g<=n-1; g++) {
        av[g]=sqrt((vr[g]*vr[g])+(vi[g]*vi[g]));
        ang[g]=atan2(vi[g],vr[g]);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง วัตถุประสงค์เพื่อให้นักศึกษาได้ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
}

void mismatch()
{
    float rm[11],im[11];

    for (i=0;i<=n-1;i++) {
        rm[i]=0;im[i]=0;
        for (j=0; j <= n-1; j++)
            if ((x[i][j] != 0) && (i!=j)) {
                rm[i]=rm[i]+sr[i][j];
                im[i]=im[i]+si[i][j];
            }
        smr[i]=p[i]-rm[i];
        smi[i]=q[i]-im[i];
    }
}

void lflow()
{
    int s,t;
    float lcr[11],lci[11],xr[11],xi[11];
    float vtr,vti,ab,c,ld;

    for (s=0; s <= n-1; s++)
        for (t=0; t <= n-1; t++)
            if ((x[s][t]!=0) && (s!=t)) {
                ab=(r[s][t]*r[s][t])+x[s][t]*x[s][t];
                lcr[t]=-bx[s][t]*vi[s]/2;
                lci[t]=-bx[s][t]*vr[s]/2;
                vtr=vr[s]-vr[t];
                vti=-vi[s]+vi[t];
                xr[t]=(vtr*r[s][t]-vti*x[s][t])/ab;
                xi[t]=(vti*r[s][t]+vtr*x[s][t])/ab;
                c=xr[t]+lcr[t];
                ld=xi[t]+lci[t];
                sr[s][t]=vr[s]*c-vi[s]*ld;
                si[s][t]=vi[s]*c+vr[s]*ld;
            }
}

void slack()
{
    float sci[11];
    float scr[11];

    sci[0]= 0;
    scr[0]= 0;
    for (j=0;j<=n-1;j++) {
        sci[j+1]=sci[j]+yi[0][j]*vr[j]+y[0][j]*vi[j];
        scr[j+1]=scr[j]+y[0][j]*vr[j]-yi[0][j]*vi[j];
    }

    q[0]=-(vr[0]*sci[j]-vi[0]*scr[j]);
    p[0]=vr[0]*scr[j]+vi[0]*sci[j];
}

void out()

```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    printf("          K= %d\n", k);
    printf(" Vbus  real  imagin  reactive power\n");
    for (j=0; j <= n-1; j++)
        printf("Vbus%d %8.6f %8.6f %8.6f\n",j,vr[j],vi[j],q[j]);
}

void out()
{
    clrscr();
    printf("          Program converge in %d iteration. \n", k);
    printf("          generation power load power\n");
    printf(" bus  volts  angle  (MW)  (MVar)  (MW)  (MVar)\n");
    printf("-----\n");
    for (j=0; j <= n-1; j++)
        printf("bus%d %8.4f %8.2f %10.3f %10.3f %10.3f %10.3f\n", j+1,
av[j], ang[j]*radtodeg, pg[j]*pb, qg[j]*pb, pl[j]*pb, ql[j]*pb);
    printf("-----\n");
    printf("press any key to show line flow.\n");
    getch();
    clrscr();
    printf("          Line flow\n");
    putchar('\n');
    printf("send  receive MW  MVar\n");
    printf("-----\n");
    i=0;
    j=0;
    for (i=0; i <= n-1; i++)
        for (j=0; j <= n-1; j++)
            if ((x[i][j]!=0) && (i!=j))
                printf("%d %d %10.3f
%10.3f\n",i+1,j+1,sr[i][j]*pb,si[i][j]*pb);
    printf("-----\n");
    printf("press any key to show power mismatch. \n");
    getch();
    clrscr();
    printf("          Power Mismatch\n");
    putchar('\n');
    printf(" Bus  MW  Mvar\n");
    printf("-----\n");
    i=1;
    for (i=0; i <= n-1; i++)
        printf(" %d%9.3f%9.3f\n", i+1, smr[i]*pb, smi[i]*pb);
    printf("-----\n");
    k = k+1000;
}

void diff()
{
    float dr[11],di[11],dav[11];

    dv=0;
    for (j=1;j<=n-1;j++) {
        dr[j]=vr[j]-vcr[j];
        di[j]=vi[j]-vci[j];
        dav[j]=sqrt((dr[j]*dr[j])+(di[j]*di[j]));
        if (dav[j] > dv)

```

```

        dv=dav[j];
    }
}

void specv()
{
    float crp[11];
    float cip[11];
    float buf;

    crp[i]=vspec[i]*vr[i];
    cip[i]=vspec[i]*vi[i];
    buf=sqrt((vr[i]*vr[i])+(vi[i]*vi[i]));
    vr[i]=crp[i]/buf;
    vi[i]=cip[i]/buf;
}

void qgen()
{
    float c1[6];
    float c2[6];
    c1[0]=0;
    c2[0]=0;
    c1[i]=0;
    c2[i]=0;
    for (j= 0; j <= n-1; j++) {
        c1[j+1]=c1[j]+yi[i][j]*vr[j]+y[i][j]*vi[j];
        c2[j+1]=c2[j]+y[i][j]*vr[j]-yi[i][j]*vi[j];
    }
    q[i]=-(vr[i]*c1[j]-vi[i]*c2[j]);
}

void aigen()
{
    cr[i] = p[i]*y[i][i]-q[i]*yi[i][i];
    ci[i] = -(q[i]*y[i][i]+p[i]*yi[i][i]);
    ar[i] = cr[i]/d[i];
    ai[i] = ci[i]/d[i];
}

void vbus()
{
    float c1[11];
    float c2[11];
    float wr,wi,avr,avi;

    c1[0]=0; c2[0]=0;
    c1[i+1]=0; c2[i+1]=0;
    for (j=0; j <= i-1; j++) {
        c1[j+1] = c1[j]+bi[i][j]*vr[j]+br[i][j]*vi[j];
        c2[j+1] = c2[j]+br[i][j]*vr[j]-bi[i][j]*vi[j];
    }
    for (j=i+1; j <= n-1; j++) {
        c1[j+1] = c1[j]+bi[i][j]*vr[j]+br[i][j]*vi[j];
        c2[j+1] = c2[j]+br[i][j]*vr[j]-bi[i][j]*vi[j];
    }
    wr=c2[i]+c2[n];
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

wi=c1[i]+c1[n];
avr=(ar[i]*vr[i]-ai[i]*vi[i])/((vr[i]*vr[i])+(vi[i]*vi[i]));
avi=(ai[i]*vr[i]+ar[i]*vi[i])/((vr[i]*vr[i])+(vi[i]*vi[i]));
vr[i] =avr-wr;
vi[i] =avi-wi;
}

void loadflow()
{
clrscr();
printf("Program Calculation Load Flow by Gauss Seidel Method.\n");
putchar('\n');
printf("How many buses ?\n");
scanf("%d",&n);
printf("Base power (MVA) = ?\n");
scanf("%f", &pb);
printf("_____ \n");
printf("Send   Receive   Series   impedance   Shunt suscept \n");
printf(" bus     bus       RS       XS         BSH\n");
printf("_____ \n");
for (i=0; i <= n-1; i++) {
for (j=0; j <= n-1; j++) {
y1r[i][j] = 0;
y1i[i][j] = 0;
r[i][j] = 0;
x[i][j] = 0;
bx[i][j] = 0;
vr[i] = 0;
vi[i] = 0;
p[i] = 0;
q[i] = 0;
qmin[i] = 0;
qmax[i] = 0;
}
}
v = 1;
for(i=0;i<=n-1;i++) {
for(j=0;j<=n-1;j++) if(i<j){
printf("%d ",i+1);
gotoxy(10,10+v);printf("%d",j+1);
gotoxy(19,10+v);scanf("%f",&r[i][j]);
gotoxy(28,10+v);scanf("%f",&x[i][j]);
gotoxy(42,10+v);scanf("%f",&bx[i][j]);
v = v+1;
}
}
printf("_____ \n");
clrscr();
printf("Program Calculated Load Flow by Gauss Seidel Method.\n");
putchar('\n');
printf("type 1. slack bus\n");
printf(" 2. generator bus\n");
printf(" 3. load bus\n");

printf("_____ \n");
printf(" Bus Type      Generation      Load      V,per unit \n");

```

```

printf("      P,MW      Q,Mvar      P,MW      Q,MVar      real imagine\n");

printf("
_____
\n");
v = 1;
for (i=0; i <= n-1; i++) {
    printf("%d\n",i+1);
    gotoxy(8,9+v);
    scanf("%d", &b[i]);
    gotoxy(14,9+v);
    scanf("%f", &pg[i]);
    pg[i] = pg[i]/pb;
    gotoxy(21,9+v);
    scanf("%f", &qg[i]);
    qg[i] = qg[i]/pb;
    gotoxy(30,9+v);
    scanf("%f", &pl[i]);
    pl[i] = pl[i]/pb;
    gotoxy(37,9+v);
    scanf("%f", &ql[i]);
    ql[i] = ql[i]/pb;
    gotoxy(45,9+v);
    scanf("%f", &vr[i]);
    gotoxy(53,9+v);
    scanf("%f", &vi[i]);
    gotoxy(61,9+v);
    scanf("%f", &qmin[i]);
    gotoxy(69,9+v);
    scanf("%f", &qmax[i]);
    p[i] = pg[i]-pl[i];
    q[i] = qg[i]-ql[i];
    v=v+1;
}

printf("
_____
\n");
for (i =0; i <= n-1; i++) {
    for (j=0; j <= n-1; j++)
        if (i<j) {
            r[i][j]=r[j][i];
            x[i][j]=x[j][i];
            bx[i][j]=bx[j][i];
        }
}

for (i=0; i <= n-1; i++)
    if (b[i]==2)
        vspec[i]=vr[i];
for (i=0; i <= n-1; i++)
    for (j= 0; j <= n-1; j++)
        if ((i<j) && (r[i][j]!=0)) {
            ylr[i][j]= r[i][j]/((r[i][j]*r[i][j])+(x[i][j]*x[i][j]));
            yli[i][j]=-x[i][j]/((r[i][j]*r[i][j])+(x[i][j]*x[i][j]));
            ylr[j][i]= ylr[i][j];
            yli[j][i]= yli[i][j];
        }
}

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

y[i][i] = 0; yi[i][i] = 0;
for (j=0; j <= n-1; j++)
    if (i!=j) {
        y[i][j]=y[i][i]+ylr[i][j];
        yi[i][j]=yi[i][i]+yli[i][j]+bx[i][j]/2;
    }
}

for (i=0; i <= n-1; i++)
    for (j=0; j <= n-1; j++)
        if (i<j) {
            y[i][j] = -ylr[i][j];
            y[j][i] = -ylr[i][j];
            yi[i][j] = -yli[i][j];
            yi[j][i] = -yli[i][j];
        }
for (i = 0; i <= n-1; i++) {
    for (j=0; j <= n-1; j++) {
        d[i] = (y[i][i]*y[i][i])+(yi[i][i]*yi[i][i]);
        br[i][j] = (y[i][j]*y[i][i]+yi[i][j]*yi[i][i])/d[i];
        bi[i][j] = (yi[i][j]*y[i][i]-y[i][j]*yi[i][i])/d[i];
    }
    if (b[i] == 3) {
        cr[i] = p[i]*y[i][i]-q[i]*yi[i][i];
        ci[i] = -(q[i]*y[i][i]+p[i]*yi[i][i]);
        ar[i] = cr[i]/d[i];
        ai[i] = ci[i]/d[i];
    }
}
k=0;
do {
    dvmax = 0;
    for (i= 1; i <= n-1; i++) {
        for (j=1; j <= n-1; j++) {
            vcr[j] = vr[j];
            vci[j] = vi[j];
        }
        if (b[i]==2) {
            vtemp=vr[i];
            vtempi=vi[i];
            specv0;
            qgen0;
            if (q[i] < qmin[i]) {
                q[i]=qmin[i];
                vr[i]=vtemp;
                vi[i]=vtempi;
            }
            if (q[i] > qmax[i]) {
                q[i]=qmax[i];
                vr[i]=vtemp;
                vi[i]=vtempi;
            }
        }
        aigen0;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ **vbus0**; ซึ่งงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

diff();
if (dv > dvmax)
    dvmax=dv;
}
outt();
if (dvmax <= 0.0001) {
    slack();
    lflow();
    mismatch();
    setpow();
    ragtopol();
    out();
}
k=k+1;
}
while (!(k>=1000));
k = k-1000;
printf("Load Flow calculation not obtained in %d iterations.\n", k);
jj=0;
i=0;
for (i=0; i <=n;i++)
{
    if((b[i]==2)||b[i]==1) {
        pp[jj] = pg[i];
        qq[jj] = qg[i];
        jj = jj+1;
    }
}
}□

```

Training Algorithm

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
#define n 0.1
#define mmm 60
#define nnn 2
float wkj[2][5],wnkj[2][5],wnji[5][3],wji[5][3];
float err1,err2,dk[5],dj[5],o[5],on[5];
double oj[5],h[5],hn[5],oni[5],onj[5],oi[5];
int jj; /* 2 , 105 0 = x1[1] | 1 = a */
static float in[nnn][mmm] = {
{-0.889591,-1.234610,-1.491089,-1.653064,-1.181407,0.974204,1.721590
,1.648055,1.235307,0.727206,0.185557,-0.204966,-0.705853,-0.678013
,-0.483131,-0.035407,0.192283,0.341611,0.394330,0.410963,0.387955
,0.368308,0.344142,0.139946,0.026947,-0.072714,-0.163229 ,-0.184889
,-0.186336,-0.168665
,-0.121489,-0.081836,-0.000524,0.034523,0.98480,0.089957,0.084806,0.052078,0.022565
,-0.013805,-0.033846,-0.045244,-0.046393,-0.034427,-0.025367
,-0.005467,0.007901,0.02191,0.023449,0.010418,0.003012
,-0.00967,-0.011712,-0.008411,-0.001268,0.004475,.00609},
{-0.470196,-0.889591,-1.234610,-1.491089,-1.653064,0.626425,1.629677
,1.714800,1.352053,0.858814,0.321615,-0.079693,-0.663439,-0.713539

```

เอกสารนี้, 1.714800, 1.352053, 0.858814, 0.321615, -0.079693, -0.663439, -0.713539 ใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

,-0.560182,-0.123848,0.124109,0.301325,0.372498,0.412929
,0.402405,0.387955,0.368308,0.177967,0.063860,-0.041841
,-0.146200,-0.176157,-0.189844,-0.179163,-0.139359,-0.102145,-0.019987
,0.017743,0.082996
,0.097632,0.090475,0.06137,0.032481,-0.005516,-0.028057,-0.042468,0.047520
,-0.038349,-0.030063,-0.0104,0.00376,0.020101,0.024078,0.012852,0.005449
,-0.008507,-0.011792,-0.009385,-0.002484,0.003774,0.006019}
};
static float tar[mmm] = {
-0.02,-0.02,-0.017295,0.01291,-0.008266,0.02,0.007803,0,0,0,0
,-0.006545,-0.003139,0.000559,0.001671,0.002291,0.003817,0.002509,0.001589
,0.000491,0,0,0,0,-0.001669,-0.001058,-0.00067,-0.000191,0.0002,0.000403
,0.000486,0.000508,0.000914,0.000461,0.000171,0,0,0,-0.000446,-0.000336
,-0.000195,-0.000024,0.000086,0.00011,0.000126,0.000224,0.000188,0.000013
,0,0,-0.000071,-0.000014,0.000022,0.000031,0.000041,0.000011};

main()
{
int v,z,i,qq;
char mt;
chw();
z = 0;
for(jj=0;jj<56;jj++)
tar[jj] = tar[jj]*20;
/*-- inputdata(); ---*/
for(qq=0;qq<=1200;qq++) {
for(jj=0;jj<56;jj++)
{
z = z+1;
printf(" %d \n",z);
out1();
while (err1 > 0.001 || err1 < -0.001)
{
adw();
out1();
printf(" %f %f %f %f %f \n",in[0][jj],in[1][jj],tar[jj],o[0],err1);
}
pw();
}
}
}
getch();
}

adw()
{
int i,j,k;
float p,q;

for(k=0;k<=0;k++)
for(j=0;j<=3;j++)
wnkj[k][j] = wkj[k][j];
for(j=0;j<=3;j++)
for(i=0;i<=1;i++)
wnji[j][i] = wji[j][i];
for (k=0;k<=0;k++)
for(j=0;j<=3;j++)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        dcal();
        wkj[k][j] = wkj[k][j] + 0.2*dk[k]*oj[j];
        for (i=0; i <= 1 ; i++)
        {
            wji[j][i] = wji[j][i] + 0.2 *dj[j]*oi[i];
        }
    }
    out1();
}
out1()
{
    int i,j,k;
    double a,b,inp;

    a = 0;
    b = 0;
    o[0] = 0;
    for(j=0;j<=3;j++)
        h[j] = 0;
    for(i=0;i<=1;i++)
        oi[i] = 0;

    for(i=0;i<=1;i++)
    {
        inp = in[i][j];
        oi[i] = 1/(1+exp(-inp/0.9));
    }
    for(j=0;j<=3;j++)
        for(i=0;i<=1;i++)
        {
            a = wji[j][i]*oi[i];
            h[j] = h[j] + a;
        }
    for(j=0;j<=3;j++)
        oj[j] = 1/(1+exp(-h[j]));
        for(k=0;k<=0;k++)
        {
            for(j=0;j<=3;j++)
            {
                b = wkj[k][j] * oj[j];
                o[k] = o[k] + b;
            }
        }
    o[k] = -1/2 + 1/(1+exp(-o[k]));
    err1 = tar[jj] - o[0];
}

out2()
{
    int i,j,k ;
    double x,y,inp ;

    x = 0;
    y = 0;
    on[0] = 0;
    for(j=0;j<=3;j++)

```

```

hn[j] = 0;
for(i=0;i<=1;i++)
  oni[i] = 0;
for(i=0;i<=1;i++)
  {
  inp = in[i][j];
  oni[i] = 1/(1+exp(-inp/0.9));
  }
for(j=0;j<=3;j++)
  for(i=0;i<=1;i++)
    {
    x = wnj[i][j]*oni[i];
    hn[j] = hn[j] + x;
    }
for(j=0;j<=3;j++)
  onj[j] = 1/(1+exp(-hn[j]));
for(k=0;k<=0;k++)
  {
  for(j=0;j<=3;j++)
    {
    y = wnkj[k][j] * onj[j];
    on[k] = on[k] + y;
    }
  }
on[k] = -1/2 + 1/(1+exp(-on[k]));
err2 = tar[j] - on[0];
}

dcal0
{
int i,j,k;
float sdw[4];
dk[0] = 0;
for(j=0;j<=3;j++)
  {
  sdw[j]=0;
  dj[j]=0;
  }
for(k=0;k<=0;k++)
  {
  o[k] = 1 / (1+exp(o[k]));
  dk[k] = o[k]*(1-o[k])*(err1);
  }
for(j=0;j<=3;j++)
  for(k=0;k<=0;k++)
    sdw[j] = sdw[j] + dk[k]*wkj[k][j];
for(j=0;j<=3;j++)
  dj[j] = oj[j]*(1-oj[j])*sdw[j];
}

/*----inputdata0
{
int i,j;
for (i=0;i<=301;i++)
  for (j=1;j<=2;j++)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยดี เพราะได้รับความช่วยเหลือ และสนับสนุนจากหลาย ๆ ฝ่าย โดยจะขอเอ่ยนามดังต่อไปนี้

ผศ. มณฑล	ลีลาจินดาไกรฤกษ์	อาจารย์ที่ปรึกษาโครงการ
ดร. กิตติ	ไพฑูรย์วัฒนกิจ	ให้คำแนะนำเกี่ยวกับโครงการด้านนิเวศ ทั้งหมด
อ. วิบูลย์	พร้อมพาณิชย์	ให้ความช่วยเหลือทางด้านโปรแกรม
อ. วัชระ	ฉัตรวิริยะ	ให้ความช่วยเหลือทางด้านโปรแกรม

และอาจารย์ท่านอื่น ๆ ซึ่งมีสามารถกล่าวชื่อได้หมดในที่นี้ ที่คอยให้คำแนะนำช่วยเหลือ ซึ่งทางคณะผู้จัดทำขอขอบคุณมา ณ โอกาสนี้

คณะผู้จัดทำ

บรรณานุกรม

1. A.E. Guile ,and W.Paterson ,Electrical P.S. Vol2 ,Pergamon Press ,1977 ,pp.88-143
2. J.Arrillaga ,C.D. Arnold ,Computer Modeling of Electrical of E.P.S. ,John Wiley & Son ,1983 ,pp.206-249
3. Hopfield ,Neural Networks and Physical System with Emergent Collective Computational Ability ,Proc. Natl. Acad. Sci. USA, Vol.79 ,April 1982. ,pp.2554-2558
4. Kreyszic ,Advance Engineering Mathematic ,6th Edition,Wiley,1989.
5. Herbert Schildt, C the Complete Reference ,2th Edition ,Mc Grawhill ,1987.
6. McCulloch ,w.w.,and Pitts ,W. 1943. A Logical Calculus of The Ideas Imminent in Nervous Activity Bolletin of Mathematical Biophysical 5:115-33
7. Minsky ,and S. Papert ,Perceptron : An Introdiction to Computational Geometry ,MIT Press ,1969
8. Phillip D. Wasserman ,Neural Computing:Theory and Practice. Van Nostrand Reinhold ,New York ,1989
9. Professor M.V. Deshpande ,Element of Eelectical Power Station Design ,Wheeler Publishing ,1978
10. Paul Gorman ,and Terrence J. Sejnowsky ,Analysis of Hidden Units in a Layer Network Trained to Classify Sonar Tarkets. Neural Networks ,1(1):76-90,1988
11. Yao-Nan YU, Electrical power system dynamics, ACADEMIC PRESS, INC.,1983.
12. Yoh-Han Pao ,Adaptive Pattern Recognition and Neural Networks. ,Addison-Wesley Publishing Company,Inc. ,1989

13. Chao-Rong Chen, and Yuan-Yih Hsu, Senior Member, Synchronous Machine Steady -State Stability Analysis Using An Artificial Neural NetWork, IEEE Trans. on Energy Conversion, Vol. 6, No. 1, 1989, pp.12-20
14. D.C.Park, M.A. El-Sharkawi, R.J. Marks II, L.E. Atlas, and M.J. Damborg, Electrical Load Forecasting Using Artificial Neural Network, IEEE Trans. on Power System, Vol.6, No.2, May 1991, pp.442-448
15. Dean J. Sobajic, and Yao-Han Pao, Artificial Neural Net Based Dynamic Security Assesment for Electric Power System, IEEE Transection Power System, Vol. 4, No.1, February 1989, pp.220-228
16. Isabelle Guyon, Neural Network Systems :Proceeding of the fifth International Symposium on Numerical Methods in Engineering, Vol 1, Bookcraft.(bath) Ltd., Avon, 1989, pp.203-210
17. Aggoune, M.A. El-Sharkawi, D.C. Park, M.J. Damborg and R.J. Marks II, Preliminary Results on Using Artificial Neural Networks for Security Assesment, IEEE Transactions on Power Systems, Vol.6, No.2, May 1991
18. Richard P. Lippmann, An Introduction to Computing with Neural Nets, IEEE ASSP MAGAZINE APRIL 1987, pp.4-22
19. Hartana, and G.G. Richards, Harmonis Source Monitoring And Identification Using Neural Networks, IEEE Trans. on Power System, Vol.5, No.4, November 1990, pp.1698-1104
20. Hiyama, Application of Rule-Based Stabilizing Controller to Electrical Power System, IEE Proceedings Vol.136, Pt.C, No.3, May 1989, pp.175-181
21. Hiyama, Rule-Based Stabilizer For Multi-Machine Power System, IEEE Transactions on Power Systems, Vol.,5, No.2, May 1990, pp.403-411
22. Yao-Nan YU, Senior Member, Khien Vongsuriya and Leonard N. Wedman, Application of an Optimal Control Theory To a Power System, IEEE TRansactions on Power Apparatus and Systems, Vol Pas-89, No. 1, January 1970, pp.55-62