

หทัยเทพ วงศ์สุวรรณ

ดร. ชิต เหล่าวัฒนา

ผู้ช่วยศาสตราจารย์

ศูนย์ปฏิบัติการพัฒนาหุ่นยนต์ภาคสนาม

ภาควิชาวิศวกรรมเครื่องกล

สถาบันเทคโนโลยีพระจอมเกล้าธนบุรี

ราษฎร์บูรณะ กรุงเทพฯ 10140

การควบคุม Glycol & Water Chiller Plant โดยใช้ Neural Networks

A Neural Network Based Control for A Glycol & Water Chiller Plant

ในช่วงเกือบทศวรรษที่ผ่านมา ข่ายงานระบบประสาธ ได้แสดงความสามารถสูง ในการใช้งานด้าน ระบบควบคุม โดยเฉพาะในระบบที่ไม่เป็นเชิงเส้น ในบทความนี้ จะแสดงให้เห็นถึง การนำขำ งานระบบประสาธ มาใช้ในการควบคุมระบบ Glycol & Water Chillers Plant ซึ่งมีพฤติกรรม แบบไม่เป็นเชิงเส้น พฤติกรรมดังกล่าว เกิดจาก สภาพอากาศ, อุปกรณ์ต่างในระบบ อาทิเช่น ปั้ม, วาล์วแบบปรับได้ เป็นต้น โดยมีเป้าหมายของการวิจัยคือ เพื่อให้ระบบทำงานโดยมีค่าไฟฟ้าต่ำที่ สุด ในขณะที่ยังรักษาประสิทธิภาพการทำงานไว้สูงสุด สำหรับในขั้นต้นนั้น ได้ทำการทดลอง จำลอง พฤติกรรมของตัว glycol chiller และ water chiller โดยแบบจำลองขำงานระบบประสาธที่ ใช้จะเป็น แบบแพร่กระจายกลับ ซึ่งจากการทดลองพบว่า ขำงานระบบประสาธสามารถเรียนรู้ พฤติกรรมของตัว glycol chiller และ water chiller ได้เป็นอย่างดี ถึงแม้ว่าสัญญาณที่ป้อนเข้ามามีความผิดพลาดไปบ้าง

For almost a decade, neural networks have shown great potentials in solving non-linear control problems. This paper shows how a neural network can control a chiller plant consisting of two glycol & water chillers. Such a chiller plant has nonlinear dynamic characteristics as a result of outdoor air conditions, equipments within the plant such as pumps, modulate valves. The objective of this research is to minimize energy cost while maintaining high performance of plant. In our initial study, model of neural network so called "Backpropagation algorithm" used to study plant behaviour operation was trained. The experimental result indicates that conditions are well recorded by the neural network although the applied input has a certain level of disturbance.

1. บทนำ

กระบวนการต่าง ๆ ในความเป็นจริงจะมีพฤติกรรมแบบไม่เป็นเชิงเส้น ตัวอย่างเช่น ในอุปกรณ์ทางเครื่องกล ส่วนมากจะมีพฤติกรรมในรูปไม่เป็นเชิงเส้น ค่าพารามิเตอร์ที่ถูกต้อง ในกระบวนการนั้นเป็น สิ่งที่เราไม่ทราบค่าพฤติกรรมของกระบวนการ จะเปลี่ยนแปลงไปตามเวลา ซึ่งเกี่ยวเนื่องกับ สภาพแวดล้อม ทำให้ไม่สามารถวัดค่าได้โดยทฤษฎีการควบคุมแบบเดิม ที่ใช้ในการออกแบบตัวควบคุม (Controller) สำหรับระบบพวกนี้ไม่เพียงพอที่จะรักษาประสิทธิภาพของระบบ เมื่อพารามิเตอร์ที่เกี่ยวข้องมีมากมาย

การควบคุมระบบไม่เป็นเชิงเส้นให้ถูกต้องเหมาะสม มักจะประสบปัญหา 2 ประการ คือ

1. ไม่สามารถที่จะสร้างแบบจำลองได้อย่างถูกต้อง เพราะเป็น ระบบที่มีรูปแบบไม่เป็นเชิงเส้นที่มีความซับซ้อนอย่างมาก

2. ความสัมพันธ์ที่เปลี่ยนแปลงไปตามเวลาระหว่าง สัญญาณควบคุม และ พฤติกรรมทางพลวัตเป็นสิ่งที่ทำนายได้ยาก

ดังนั้นเพื่อให้สอดคล้องกับปัญหาที่พบในระบบไม่เป็นเชิงเส้น จึงได้มีผู้คิดค้นรูปแบบการควบคุมแบบใหม่ขึ้น ซึ่งอยู่ในรูปของ Adaptive Control และ Self-tuning Control System

Artificial Neural Network เป็นระบบควบคุมประเภท Adaptive Control แบบหนึ่ง เราสามารถใช้หารูปแบบที่แสดงพฤติกรรมของระบบที่ไม่เป็นเชิงเส้น และ ตัวควบคุมแบบไม่เป็นเชิงเส้นที่เหมาะสมกับ พฤติกรรมทางพลวัตของกระบวนการ เราใช้คำว่า "Neurocontroller" ปรับตัวตาม การเปลี่ยนแปลงทางพลวัตของระบบด้วย นอกจากการควบคุมระบบอย่างถูกต้องเหมาะสมแล้ว การคำนึงถึง การจัดการพลังงาน ก็เป็นอีกหัวข้อที่น่าสนใจอีกด้วย

2. ข่ายงานระบบประสาท (Neural Networks)

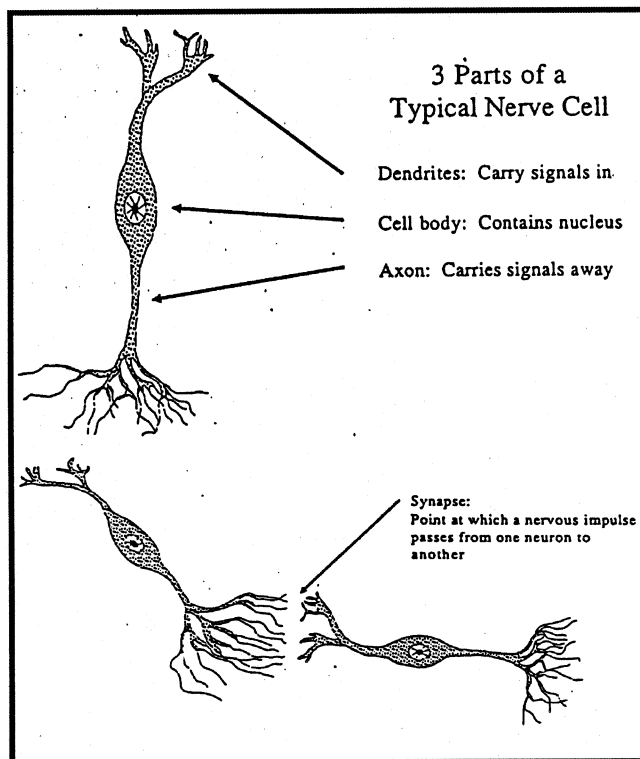
ข่ายงานระบบประสาท (Neural Networks) คือ ระบบที่มีสัญญาณเข้า และ สัญญาณออก ซึ่งประกอบขึ้น จากหลายๆ ฟังก์ชันคณิตศาสตร์ ที่มีรูปแบบอย่างง่าย ๆ แบบเดียวกัน หรือ เรียกอีกอย่างว่า หน่วยประมวลผลพื้นฐาน (Process Element Unit) มาต่อกันเป็นชั้น ๆ (Layer) และแต่ละชั้นมาต่อกันเป็น ลักษณะโครงข่าย (network) โดยแต่ละหน่วย จะมีความสัมพันธ์ต่อกันโดยอาศัยค่าถ่วงน้ำหนัก (weight) เป็นตัวเชื่อม การเปลี่ยนแปลงค่าถ่วงน้ำหนักของแต่ละหน่วยย่อย จะทำให้พฤติกรรมของข่ายงานระบบประสาทเปลี่ยนแปลงไป ดังนั้นการที่เราพยายามจำลองรูปแบบความสัมพันธ์ระหว่าง สัญญาณเข้า และสัญญาณออก โดยใช้ข่ายงานระบบประสาท คือการเลือกค่าถ่วงน้ำหนักที่ถูกต่อนั่นเอง ซึ่งเราเรียกกระบวนการนี้ว่า กระบวนการเรียนรู้ของโครงข่าย (training network)

ในเชิงทฤษฎีแล้ว ข่ายงานระบบประสาทมีความสามารถในการประมาณค่าทุกๆ ฟังก์ชันต่อเนื่องได้อย่างเที่ยงตรง ถึงทุกระดับความต้องการ ถ้ามีจำนวนชั้น และจำนวนโหนด ที่เหมาะสม

2.1 แบบจำลองโครงข่ายเซลล์ประสาทเทียม

สำหรับการทำงานของข่ายงานระบบประสาท จะถอดแบบมาจากการทำงานของระบบสมองมนุษย์ คือจะมีการส่งผ่านข้อมูลโดยการเชื่อมต่อของโหนดกัน เป็นโครงข่ายร่างแหอย่างหนาแน่น และ มีการทำงานในลักษณะขนาน

เซลล์ประสาทแต่ละเซลล์ จะประกอบด้วย ตัวเซลล์ (Cell Body) ทำหน้าที่ประมวลผลข้อมูล และ แขนงที่แยกออกไปจากตัวเซลล์ ซึ่งได้แก่ เดนไดรต์ (Dendrite) เป็นตัวรับข้อมูลเข้ามา และ แอกซอน (Axon) เป็นตัวส่งสัญญาณข้อมูลออกไป ดังรูปที่ 1



รูปที่ 1 โครงสร้างของเซลล์ประสาท และการส่งสัญญาณกระแสประสาท

แบบจำลองโครงข่ายเซลล์ประสาทเทียม ที่จะกล่าวถึงต่อไปนี้เป็นแบบจำลองที่พัฒนามาจาก แบบจำลองของ McCulloch - Pitts ซึ่งเรียกกันว่า "Adaline"[1] ซึ่งมีองค์ประกอบที่สำคัญ 2 ส่วน คือ

1. สมการผลรวมของค่าถ่วงน้ำหนัก (Weight Summer, Σ)
2. ฟังก์ชันการเร่งเร็ว (Activation Function) ซึ่งในที่นี้จะเป็นแบบไม่เป็นเชิงเส้น

ซึ่งแบบจำลองนี้ เป็น แบบจำลองเซลล์ประสาทที่สามารถสอนให้โครงข่ายตัดสินใจเองได้ ดังแสดงในรูปที่ 2 โดยมี

x_i เป็น สัญญาณอินพุต

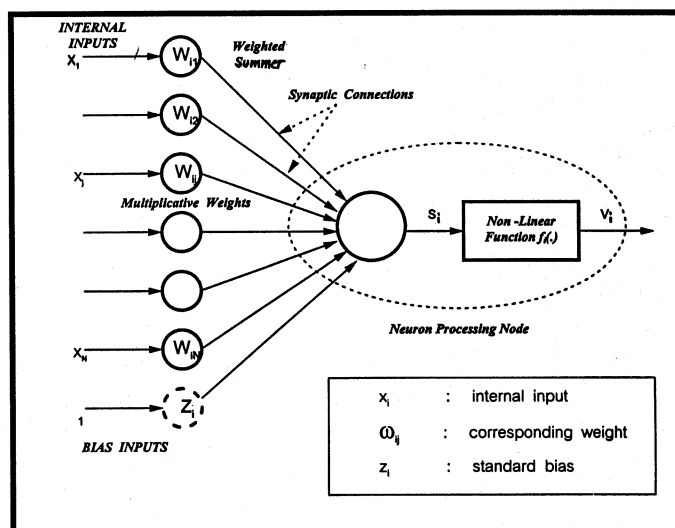
$X = \{ x_i \}$ คือ เวกเตอร์สัญญาณอินพุต ซึ่งประกอบด้วย สัญญาณอินพุต n ตัว

v_i เป็น สัญญาณเอาท์พุต

$V = \{ v_i \}$ คือ เวกเตอร์สัญญาณเอาท์พุตซึ่งประกอบด้วยสัญญาณเอาท์พุต m ตัว

w_i เป็น ค่าถ่วงน้ำหนักที่ได้จากการสอนโครงข่าย

$W = \{ w_i \}$ เป็นเวกเตอร์ของ ค่าถ่วงน้ำหนักโดยมี n ค่า



รูปที่ 2 แบบจำลองโครงข่ายเซลล์ประสาทเทียม

แต่ละโหนดในโครงข่าย จะใช้แทนเซลล์ประสาทแต่ละเซลล์และมีเซ็นแปล ซึ่งจะเชื่อมต่อกันเพื่อใช้ในการส่งผ่านสัญญาณการกระตุ้นหรือ ยับยั้งสัญญาณ จะขึ้นอยู่กับค่าถ่วงน้ำหนัก w_i

$$s(X) = \sum_{i=0}^{n-1} w_i * x_i \quad (1)$$

$$v(X) = f(s(X)) \quad (2)$$

ฟังก์ชันการเร่งเร็ว (Activation Function) ซึ่งเป็นฟังก์ชันไม่เป็นเชิงเส้น $f(s)$ ในที่นี้จะ เป็น Sigmoid Function

$$f(s) = [1 - \exp(-2s)] / [1 + \exp(-2s)] = \tanh(s) \quad (3)$$

ด้วยความไม่เป็นเชิงเส้น Adaline จะมีพฤติกรรมคล้ายๆกับ ตัวกรองเชิงเส้น เมื่อค่าเอาท์พุตของสมการมีค่าน้อยๆ และ จะมีช่วงค่าสัญญาณเอาท์พุต อยู่ระหว่าง -1 ถึง +1 โดยทั่วไป สัญญาณอินพุต ของ Adaline จะมีโหนดหนึ่งที่มีค่าเป็น 1 อยู่ เสมอ เรียกว่า " Bias Node "

การเรียนรู้ของ โครงข่ายเซลล์ประสาทเทียม แบบ Adaline เพื่อให้ปรับตัวเข้าหาพฤติกรรมของระบบ ค่าเวกเตอร์อินพุต X จะทำให้เกิดเอาต์พุต $V(X)$ มีค่าใกล้เคียงหรือ เท่ากับ ค่าเอาต์พุตที่ต้องการ $d(X)$ เรากระทำโดยปรับ ค่าถ่วงน้ำหนัก ซึ่งเราทำการสอนโครงข่าย โดยป้อนอินพุต กับ เอาต์พุต เป็นคู่ๆ เราเรียกรูปแบบการสอนลักษณะนี้ว่า “สอนแบบชี้หน้า (Supervised Learning)”

การวัดสมรรถนะของ Adaline เราใช้ ค่าความผิดพลาดยกกำลัง 2 (mean - square error), J โดยที่ $E(\bullet)$ เป็นเครื่องหมายแสดงถึงค่าความผิดพลาดของทุกๆข้อมูลที่ใช้ในการสอน

$$J = E(error)^2 \quad (4)$$

$$= E(d(X) - V(X))^2 \quad (5)$$

$$= E(d(X) - f(\sum_{i=0}^{n-1} w_i * x_i))^2 \quad (6)$$

การปรับค่าถ่วงน้ำหนัก เพื่อให้ค่า J มีค่าต่ำที่สุด โดยค่าถ่วงน้ำหนักใหม่หาได้จาก

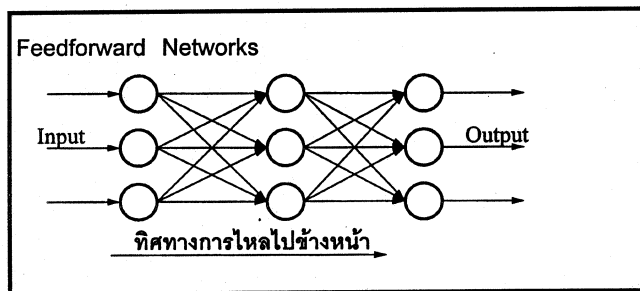
$$w_{i, new} = w_{i, old} + 2\mu\delta x_i \quad (7)$$

$$\delta = (d(X) - V(X)) * f'(s(X)) \quad (8)$$

โดย $f(s(X))$ คือ อนุพันธ์ของฟังก์ชันการกระตุ้น ผู้ออกแบบจะทำการเลือกค่าอัตราเร็วในการเรียนรู้ (Learning Rate), μ ซึ่งจะส่งผลถึงความเร็วในการเรียนรู้ และ เสถียรภาพของการปรับค่าถ่วงน้ำหนัก ค่า δ สามารถที่จะเท่ากับ ค่าความผิดพลาดของ $d(X) - V(X)$ ถ้า $f(s)$ เป็นฟังก์ชันเฉพาะเจาะจง (identity function) ในกรณีนี้สมการ (7) และ (8) จะกลายเป็นวิธีกำลังสองน้อยสุด (least - mean square algorithm(LMS)) ของ Widrow กับ Hoff[1] และ Widrow กับ steams[3] ถ้าเราดำเนินการตามขั้นตอนดังกล่าว โดยใช้ หลายๆ คู่อินพุต - เอาต์พุต ในการปรับค่าถ่วงน้ำหนัก ในที่สุดก็จะได้ค่า J ที่ต่ำที่สุด หรือ ที่ยอมรับได้

2.2 การแพร่กระจายกลับ (Back-Propagation)

จากที่กล่าวมาแล้ว เกี่ยวกับ แบบจำลองโครงข่ายเซลล์ประสาท จะมี การเชื่อมต่อกันของโหนดในลักษณะโครงข่ายอย่างแนบหนา เพื่อให้โครงข่ายสามารถเรียนรู้ และสามารถจดจำสิ่งที่เรียนรู้ มาแล้วได้ซึ่งในบทความนี้จะนำ Adline มาต่อกัน ในรูปแบบที่เรียกว่า “โครงข่ายที่ส่งสัญญาณไปข้างหน้า (Layered feedforward neural network)” ดังรูปที่ 3



รูปที่ 3 โครงข่ายที่ส่งสัญญาณไปข้างหน้า

ชั้นของ Adline จะประกอบด้วย Adline หลายๆตัว โดยมี เวกเตอร์อินพุตเดียวกัน ซึ่งการต่อกันหลายๆชั้น จะเกิดขึ้นจากการที่เอาต์พุตของชั้นก่อน จะเป็น อินพุตของชั้นถัดไป

วิธีการที่ใช้สอนโครงข่ายระบบประสาทหลายชั้น เราเรียกกันว่า “แพร่กระจายกลับ (Back - Propagation)” [2], [5], [6] โดยวิธีการนี้จะปรับค่าถ่วงน้ำหนัก จนค่าความผิดพลาดกำลังสองต่ำสุด (mean - square error), J จากสมการที่ (5) สามารถเขียนได้ใหม่ ดังนี้

$$J = E(||d(X) - V(X)||^2) \quad (9)$$

เมื่อ $V(X)$ คือ เอาต์พุตเวกเตอร์ของชั้นสุดท้ายของโครงข่าย สำหรับชั้นเอาต์พุตสุดท้ายที่มี m ค่า และ V_m คือ เอาต์พุต ที่ได้จากโครงข่ายชั้นที่มี m โหนด, δ_j คือ ค่าความผิดพลาดของชั้นก่อนหน้า J โหนด, w_{jm} คือ ค่าถ่วงน้ำหนักของชั้นเอาต์พุต m โหนด กับชั้นอินพุต j โหนด

$$\delta_m = (d_m(X) - V_m(X))f'(s_m(X)) \quad (10)$$

สำหรับค่าใน ชั้น m โหนด ซึ่งเกี่ยวข้องกับชั้นอื่น

ค่าความผิดพลาดคือ

$$\delta_m = f'(s_m(X)) \sum_j \delta_j * w_{jm} \quad (11)$$

แต่ละค่าถ่วงน้ำหนักจะถูกปรับโดยใช้สมการเดียวกับ กรณี Adline ชั้นเดียว โดย I คือขอบเขตของอินพุตที่เข้ามาในชั้น m โหนด

$$w_{mi, new} = w_{mi, old} + 2\mu\delta_m X_k \quad (12)$$

เราเรียกรูปแบบการเรียนรู้นี้ว่า “แพร่กระจายกลับ” เพราะว่าค่าความผิดพลาดที่เกิดขึ้นในชั้นเอาต์พุต คำนวณได้จากสมการ (10) และ คำนวณย้อนกลับจากชั้นเอาต์พุต ผ่านชั้นฮิดเดนตลอดจนถึงชั้นอินพุต โดยใช้สมการ (11) ค่าถ่วงน้ำหนักจะถูกปรับค่า ตามสมการที่ (12)

วิธีการเรียนรู้แบบแพร่กระจายกลับได้ถูกนำมา ใช้กับโครงข่ายที่มีการเชื่อมต่อกันหลายชั้นที่ใช้ในการ แก้ปัญหาเรื่องการจดจำรูปแบบ (Pattern Recognition), หน่วยความจำช่วยเหลือ (associative memory), ตัวกรองแบบปรับค่าได้ (adaptive filter) แต่สำหรับในบทความนี้จะนำมาใช้ในการศึกษาพฤติกรรมของระบบซึ่งมี พฤติกรรมแบบไม่เป็นเชิงเส้น

2.3 การจำลองระบบโดยใช้ข่ายงานระบบประสาท

(system identification by using neural nets)

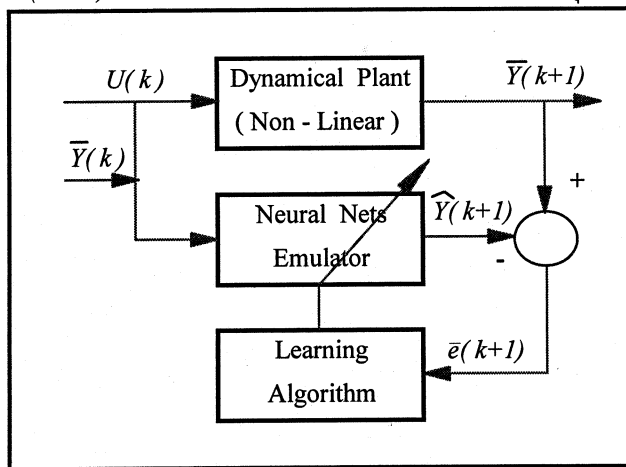
ในปัจจุบันได้มีการนำข่ายงานระบบประสาท มาประยุกต์ใช้ในกิจการต่างๆมากขึ้น อาทิเช่น เตาหลอมโลหะไฟฟ้า (Electric arc furnace), กระบวนการในอุตสาหกรรมเคมี (chemical process industries), การควบคุมระบบบังคับทิศทาง (Vehicular control) เป็นต้น เนื่องมาจากข่ายงานระบบประสาท มีความสามารถในการประมาณค่าฟังก์ชันที่ไม่เป็นเชิงเส้น (nonlinear function mapping) จึงทำให้การศึกษาพฤติกรรมของระบบที่จะถูกควบคุม ให้มีความถูกต้องมากยิ่งขึ้น

โดยรูปแบบทั่วไปที่ทำการคือ ผู้ใช้จะนำข้อมูลในอดีตของขบวนการนั้นๆ มาสร้างตัวทำนายพฤติกรรมของขบวนการ โดยใช้รูปแบบการเรียนรู้ของข่ายงานระบบประสาท หลังจากนั้นนำแบบจำลองที่ได้ มาใช้ในการควบคุมตำแหน่งการทำงาน (Setpoint) ของขบวนการให้มีพฤติกรรมที่เหมาะสม ตัวอย่างเช่น บริษัท Technology and Pavilian's Process ได้นำข่ายงานระบบประสาท มาประยุกต์ ใช้กับขบวนการต่างๆ มากกว่า 200 โครงการ อันประกอบด้วยออกแบบ และ optimization Distillation column, ออกแบบ และ ควบคุมระบบการผลิตพลาสติก, ออกแบบ และ ควบคุมระบบความไม่บริสุทธิ์ของ Boiler, และอื่นๆอีกมาก ทำให้สามารถประหยัดเงินในแต่ละขบวนการเป็นจำนวนหลายล้านดอลลาร์ต่อ

ในการประยุกต์ใช้ข่ายงานระบบประสาทกับขบวนการต่างๆ โดยสรุป มีขั้นตอนดังนี้

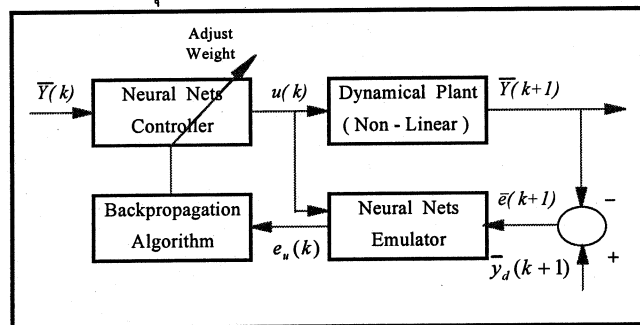
1. การการสร้างแบบจำลองของขบวนการ หรือที่เรียกว่า “**Neural Network Plant Emulator**” ซึ่งโดยส่วนใหญ่ จะประกอบด้วยหลายๆ สัญญาณเข้า (ต.ย.เช่น ความดัน, อุณหภูมิ, อัตราการไหล, คุณสมบัติอื่น) และ สัญญาณออก จำนวนมาก (ต.ย. เช่น ระดับความไม่บริสุทธิ์, ความไม่แน่นอน เป็นต้น) ในรูปที่ 4 แสดงโครงข่ายระบบประสาทแบบปรับค่า (an adaptive neural network) ซึ่งจำลองแบบของขบวนการที่ไม่รู้ค่าได้ ในการเรียนรู้พฤติกรรมทางพลวัตของขบวนการ จากข้อมูลในอดีต

ในบทความนี้เราสมมุติให้สถานะของระบบที่เข้าระบบ (Plant) ปราศจาก สัญญาณรบกวน(noise) ในสภาวะเริ่มต้นสัญญาณที่เข้าสู่ระบบ จะอยู่ในลักษณะสุ่ม ที่เวลา k ข้อมูลที่เข้าสู่โครงข่ายระบบประสาทจะอยู่ในรูปกลุ่มข้อมูล ระหว่าง $U(k)$ และ $\bar{Y}(k)$ โครงข่ายระบบประสาทจะถูกสอนโดย การแพร่กระจายกลับ ตามใช้สมการที่(10),(11),(12) เพื่อทำนายสถานะต่อไปของระบบ $\hat{Y}(k+1)$ นำไปเปรียบเทียบกับข้อมูลเอาท์พุต $\bar{Y}(k+1)$ ของระบบ เพื่อปรับค่าความผิดพลาดให้น้อยที่สุด



รูปที่ 4 ข่ายงานระบบประสาทสำหรับจำลองพฤติกรรมของระบบ

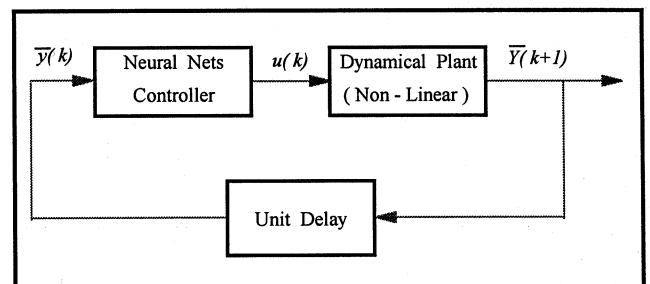
2. เมื่อได้ Plant Emulator ที่เหมาะสมแล้ว จึงนำไปใช้ในการร่วมในการเรียนรู้ของตัวควบคุมแบบข่ายงานระบบประสาท หรือที่เรียกว่า “**Neural Network Controller**” ย่อว่า “**NC**” ดังรูปที่ 5 ซึ่งแสดงถึงการ ใช้แบบจำลองของขบวนการในการสอนตัวควบคุมแบบ 1 Step เพื่อหา ค่าถ่วงน้ำหนักของตัวควบคุมแบบข่ายงานระบบประสาท โดยค่าความผิดพลาดในรูปเวกเตอร์ คือ ความแตกต่างระหว่างค่าสัญญาณออกของ ขบวนการ ($\bar{Y}(k+1)$) กับ ค่าสัญญาณออกที่ต้องการ($\bar{Y}_d(k+1)$) ให้มีค่าน้อยลงที่สุด



รูปที่ 5 ขั้นตอนการเรียนรู้ของตัวควบคุมแบบข่ายงานระบบประสาท

เมื่อเริ่มต้น ตัวควบคุมยังไม่ได้รับการสอน ค่าสัญญาณ $u(k)$ ที่ออกไปยังมีความผิดพลาดอยู่ จะถูกส่งผ่านไปยัง Plant Emulator และ ตัว Plant เอง จะได้ค่าเอาท์พุตที่สถานะต่อไป ค่าความผิดพลาด $(\bar{Y}(k+1) - \bar{Y}_d(k+1))^2$ จะแพร่กระจายกลับผ่านแบบจำลองของขบวนการ เพื่อหา ค่าสัญญาณความผิดพลาด($e_u(k)$) โดยใช้สมการ(10) และ (11) สำหรับ ปรับค่าค่าถ่วงน้ำหนักของตัวควบคุม ค่าถ่วงน้ำหนัก ของตัวควบคุมได้ รับการปรับโดยใช้วิธีการเรียนรู้ แบบแพร่กระจายกลับ โดยใช้สมการ(11) และ (12) ซึ่งนี่เป็นเหตุผลที่ว่าทำไมในการสอนตัวควบคุมต้องใช้ Plant Emulator เนื่องจากตัว Plant จริง แพร่กระจายค่าความผิดพลาดกลับไป ยังตัวควบคุมไม่ได้ ค่าสัญญาณควบคุม (e_u) จะมีความถูกต้องมากน้อย เพียงใด จำเป็นต้องใช้จำนวน Adlines เป็นจำนวนมากในชั้น Hidden Layer [6]

หลังจากการเรียนรู้เสร็จสิ้นลง ตัวจำลองพฤติกรรมระบบ (Neural Network Emulator) จะไม่ถูกนำมาใช้อีกต่อไป เฉพาะตัวควบคุมแบบข่ายงานระบบประสาท (Neural Network Controller) เท่านั้น ที่ถูกนำมาใช้ในการควบคุมระบบ ดังรูปที่ 6



รูปที่ 6 รูปแบบการนำข่ายงานระบบประสาทไปใช้ในการควบคุม

ที่กล่าวมาข้างต้นนี้เป็นวิธีการของ Nguyen and Widrow's approach [6] เพื่อสร้างตัวควบคุมเพื่อใช้ควบคุม non - linear plant จากจุดเริ่มต้นใดๆไปยังจุดเป้าหมายที่กำหนดไว้ตั้งแต่แรก

ในหัวข้อนี้ ได้บรรยายถึง ส่วนเล็กๆในการนำข่ายงานระบบประสาท มาประยุกต์ใช้ในวงการอุตสาหกรรม ในปัจจุบันข้าพเจ้าคิดว่าศาสตร์นี้ยัง เติบโตต่อไปข้างหน้าโดยก่อนที่จะทราบเรื่องนี้จะสงสัยเลยไป ข้าพเจ้าคิดว่า ข่ายงานระบบประสาท จะเข้ามามีบทบาทที่สำคัญอย่างมากในชีวิตประจำวันของมนุษย์ชาติ ดังเช่นทุกวันนี้ที่ญี่ปุ่นที่ **Fuzzy Logic** ได้กลายเป็นเช่นนั้นไปแล้ว

2.4 ขั้นตอนในการสร้างข่ายงานระบบประสาท

(Modeling of ANN)

ขั้นตอนในการสร้างข่ายงานระบบประสาท เราสามารถ สรุปได้เป็น ข้อๆ ดังนี้

1. เลือกรูปแบบของข่ายงานระบบประสาท อันได้แก่ ข้อมูลที่ต้องการ (input signal) และ สิ่งที่ต้องการ (output signal)
2. รวบรวมข้อมูลที่ต้องการ
3. นำข้อมูลที่รวบรวมมาจัดรูปแบบ ให้เหมาะสม
4. ออกแบบ และ สร้างข่ายงานระบบประสาท

5. สอนข่ายงานระบบประสาท
6. วิเคราะห์ข่ายงานระบบประสาท
7. ปรับปรุงข่ายงานระบบประสาท
8. นำข่ายงานระบบประสาทที่สมบูรณ์ไปใช้งานตามที่ ตั้งเป้าหมายที่ตั้งไว้

ต่อไปจะกล่าวถึง การสร้าง Neural Network Plant Emulator ของ Glycol & Water Chiller plant โดยใช้การเรียนรู้แบบแพร่กระจายกลับ สำหรับโครงข่ายระบบประสาทหลายชั้น

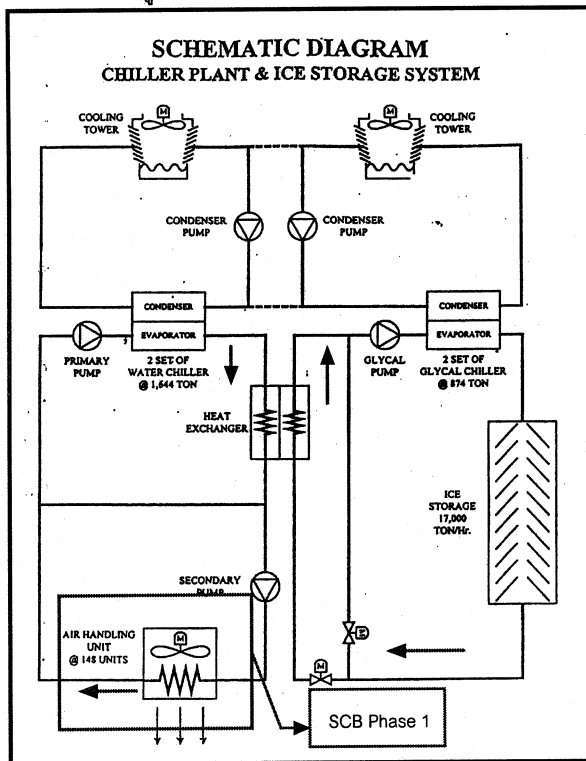
3. A. Glycol & Water Chiller Plant

ในบทความนี้แสดงให้เห็นถึง การนำข่ายงานระบบประสาทมาใช้ในการควบคุม ระบบทำน้ำเย็นแบบ Glycol & Water Chiller Plant ซึ่งมีพฤติกรรมแบบไม่เป็นเชิงเส้น พฤติกรรมดังกล่าวเกิดจาก สภาพอากาศ, อุปกรณ์ต่างในระบบอาทิเช่นปั๊ม, วาล์วแบบปรับได้ เป็นต้น โดยมีเป้าหมายของการวิจัย คือเพื่อให้ระบบทำงานโดยมีค่าไฟฟ้าต่ำที่สุด ในขณะที่ยังรักษาประสิทธิภาพการทำงานไว้สูงสุด

Water Chiller Plant คือ ระบบซึ่งใช้ในการผลิตน้ำเย็น เป็นส่วนหนึ่งของระบบปรับอากาศแบบใช้น้ำเย็น ซึ่งส่งน้ำเย็นไปยังเครื่องเป่าลมเย็นต่าง ๆ ของอาคาร ในปริมาณที่สอดคล้องกับภาระทำความเย็นที่เปลี่ยนแปลงไป

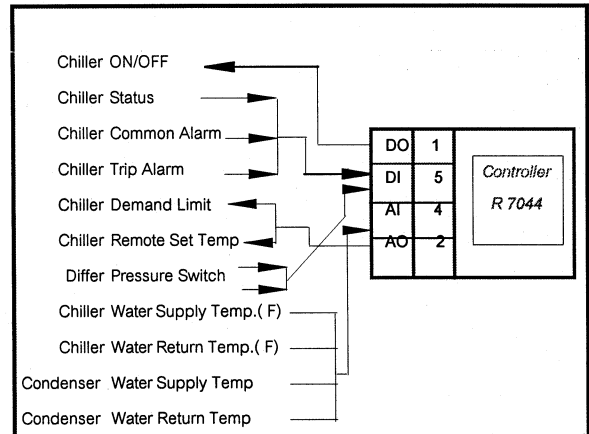
สำหรับระบบของ ธนาคารไทยพาณิชย์ สำนักงานใหญ่ เป็นระบบ Chiller Plant ผลิตน้ำเย็น ร่วมกับ ระบบ ICE Storage เพื่อประหยัดค่าไฟฟ้าที่ใช้ในการทำความเย็น

ลักษณะการทำงานร่วมกันของระบบ Chiller กับ ICE Storage กล่าวคือ ทำการผลิตน้ำแข็งจากระบบ ICE Storage ในเวลากลางคืน และ นำน้ำแข็งที่เก็บไว้มาจ่าย Load ร่วมกับน้ำเย็นที่ผลิตจาก Chiller Plant ในเวลากลางวัน ดังรูปที่ 7



รูปที่ 7 แสดงการทำงานของ ระบบ Glycol & Water Chiller Plant

จากการศึกษาระบบอาคารอัจฉริยะ (Building Automation System (B.A.S.)) ที่ใช้ในการควบคุม Chiller Plant เราพบว่า ข้อมูลที่รับ - ส่ง ของ B.A.S เพื่อใช้ในการควบคุม Chiller Plant มี ดังรูปที่ 8



รูปที่ 8 ข้อมูลที่รับ - ส่ง ของ B.A.S เพื่อใช้ในการควบคุม Chiller Plant

เนื่องจาก ระบบ B.A.S ของอาคารสำนักงานใหญ่ ธนาคารไทยพาณิชย์(มหาชน)จำกัด, SCB phase1 ยังทำงานไม่เต็มที่ ในขณะนี้ ฝ่ายปฏิบัติการเดินเครื่อง จึงต้องเก็บข้อมูลต่างๆโดยใช้พนักงานแทนซึ่งข้อมูลที่เก็บได้มีดังนี้

1. เวลา

2. Chiller No.

WCH - 1 = No. 1 , WCH - 2 = No. 2

GCH - 1 = No. 3 , GCH - 2 = No. 4

3. ความดันของน้ำที่ เข้า ด้าน Cooler (Psi)

4. อุณหภูมิของน้ำที่ เข้า ด้าน Cooler (°F)

5. ความดันของน้ำที่ เข้า ด้าน Condenser (Psi)

6. อุณหภูมิของน้ำที่ เข้า ด้าน Condenser (°F)

7. ความดันของน้ำที่ ออก ด้าน Cooler (Psi)

8. อุณหภูมิของน้ำที่ ออก ด้าน Cooler (°F)

9. ความดันของน้ำที่ ออก ด้าน Condenser (Psi)

10. อุณหภูมิของน้ำที่ ออก ด้าน Condenser (°F)

11. เปอร์เซนต์ความต้องการพลังงานไฟฟ้า

(0 - 100 %)

* ส่วนค่าอัตราการไหลของน้ำในท่อน้ำเย็น ทั้งไป และ กลับจากตัวตึก ประมาณจาก บั้มที่ทำงานอยู่

4. ขั้นตอนในการสอนข่ายงานระบบประสาท

(Training a neural network emulator)

ในบทความนี้ เราเลือกใช้การเรียนรู้แบบแพร่กระจายกลับ สำหรับข่ายงานระบบประสาทหลายชั้น เพื่อใช้ในการสร้าง Neural Network Emulator ในการสร้างข่ายงานระบบประสาท มีขั้นตอนดังนี้

1.) เลือก ข้อมูลที่ใช้ในการศึกษาพฤติกรรมของ ระบบ

Input Data ประกอบด้วย

1. เวลา

2. Chiller No.

WCH - 1 = No. 1 , WCH - 2 = No. 2

GCH - 1 = No. 3 , GCH - 2 = No. 4

3. สถานะของตัว Chiller => on / off => 0 / 1

4. อัตราการไหลของน้ำเข้าสู่ตัว Chiller , gpm

5. ความดันของน้ำที่ เข้า ด้าน Cooler (Psi)

6. อุณหภูมิของน้ำที่ เข้า ด้าน Cooler (°F)

7. ความดันของน้ำที่ เข้า ด้าน Condenser (Psi)

8. อุณหภูมิของน้ำที่ เข้า ด้าน Condenser (°F)

สรุป ในข้อมูลในชั้นอินพุต มี 8 โหนด

Output Data ประกอบด้วย

1. ความดันของน้ำที่ ออก ด้าน Cooler (Psi)

2. อุณหภูมิของน้ำที่ ออก ด้าน Cooler (°F)

3. ความดันของน้ำที่ ออก ด้าน Condenser (Psi)

4. อุณหภูมิของน้ำที่ ออก ด้าน Condenser (°F)

5. สถานะของตัว Chiller => on / off => 0 / 1

6. เปอร์เซ็นต์ความต้องการพลังงานไฟฟ้า

(0 - 100 %)

สรุป ในข้อมูลในชั้นเอาต์พุต มี 6 โหนด

2.) การออกแบบข่ายงานระบบประสาท (Neural network modeling)

จากข้อมูลของ อุปกรณ์ทำน้ำเย็น (Chiller Equipment) ทำให้ทราบถึงจำนวน input node ใน input layer และจำนวน output node ใน output layer นั่นคือ

input layer : 8 input nodes

output layer : 6 output nodes

การหาความสัมพันธ์ระหว่าง ชั้นอินพุต กับ ชั้นเอาต์พุต โดยออกแบบชั้นฮิดเดน (Hidden Layer) ในทางข่ายงานระบบประสาท เพื่อหาความสัมพันธ์ ระหว่าง อินพุต กับ เอาต์พุต ซึ่งไม่เป็นเชิงเส้น ในที่นี้ก็คือค่าถ่วงน้ำหนักที่ได้จากการสอนข่ายงานระบบประสาทนั่นเอง การที่จะกำหนด ชั้นของฮิดเดนว่า โครงข่ายงานระบบประสาท ควรมี กี่ชั้นฮิดเดน และ แต่ละชั้นฮิดเดน ควรมีกี่โหนด นั้น ยังไม่มีกฎเกณฑ์ตายตัว ซึ่งได้ผู้ทำการวิจัยค้นคว้าหาคำตอบว่า ควรมีชั้นฮิดเดนกี่ชั้น อยู่หลายๆท่านด้วยกัน จึงขอยกบางบทความมากล่าวดังนี้

1. การกำหนดจำนวนชั้น และ จำนวนโหนดของแต่ละชั้นของฮิดเดนสามารถทำได้ด้วยการทดลอง[7]

2. การกำหนดจำนวนชั้น และจำนวนโหนดของแต่ละชั้นของฮิดเดนสามารถทำได้ด้วยการทดลอง แต่ถ้าชั้นฮิดเดน 1 ชั้น สามารถหาจำนวนโหนดในชั้นฮิดเดน ได้โดยดูจากความสัมพันธ์ ระหว่าง ชั้นอินพุต กับ ชั้นเอาต์พุต [8]

3. จำนวนโหนดของชั้นฮิดเดน สำหรับฟังก์ชันค่าต่อเนื่องนั้นโดยทั่วไปจะมีค่าเป็น 2 เท่า (โดยประมาณ) ของ จำนวนโหนดของตัวแปรด้านเข้า และมีเพียงชั้นเดียวก็เพียงพอที่จะใช้กับฟังก์ชันที่เป็นค่าต่อเนื่องได้ [9]

5. ผลการทดลองสอนข่ายงานระบบประสาท

(Result of Training a neural network emulator)

จากบทสรุปของหลายบทความจึงได้ทำการทดลองสอนข่ายงานระบบประสาท โดยทำเปลี่ยนชั้น Hidden Layer หลายแบบ เพื่อศึกษาผลของจำนวน Hidden nodes กับค่าความผิดพลาด ในการเรียนรู้

Model	Input Layer (No. of Node)	Hidden Layer			Output Layer (No. of Node)
		L1	L2	L3	
1	8	4	5	4	6
2	8	8	7	6	6
3	8	16			6

• จำนวนข้อมูลที่ใช้สอน มี 519 ข้อมูล

• การวนรอบข้อมูลในการสอน 25,000 ครั้ง

จากการสอนโครงข่ายระบบประสาททั้ง 3 แบบ เราได้ค่าความผิดพลาดกำลังสองโดยเฉลี่ย (Root Mean Square Error (RMSE) ทั้ง 3 แบบ ดังนี้

รูปแบบที่ 1 RMSE = 0.035030

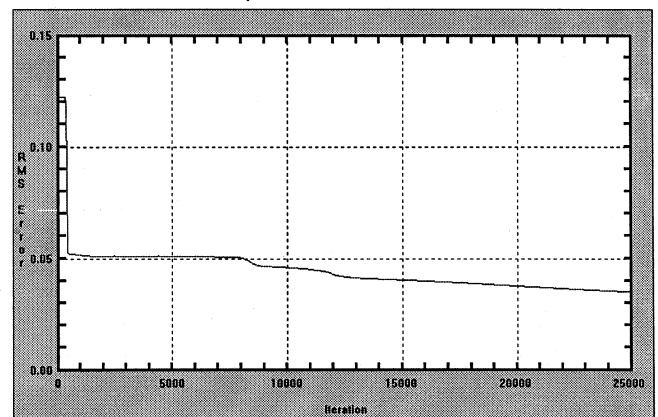
รูปแบบที่ 2 RMSE = 0.031509

รูปแบบที่ 3 RMSE = 0.030099

รูปกราฟแสดงความสัมพันธ์ระหว่างค่าความผิดพลาดกับจำนวนครั้งในการสอนข่ายงานระบบประสาท ทั้ง 3 แบบ เป็นดังต่อไปนี้

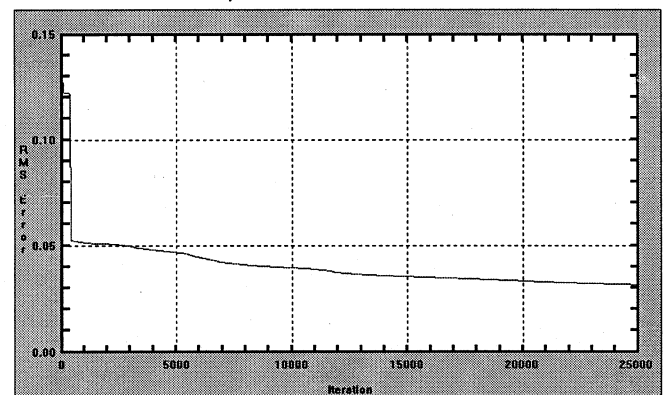
Model 1: Input 8 Node, Output 6 node, Hidden Layer : 4, 5, 4 Nodes

Root Mean Square error = 0.035030

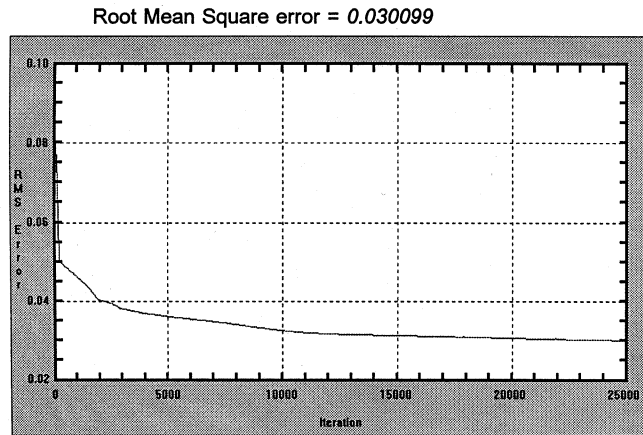


Model 2: Input 8 Node, Output 6 node, Hidden Layer : 8, 7, 6 Nodes

Root Mean Square error = 0.031509



Model 3 : Input 8 Node , Output 6 node , Hidden Layer : 16 Nodes



6. สรุปผลการสอนข่ายงานระบบประสาท

(Conclusion of Training a neural network emulator)

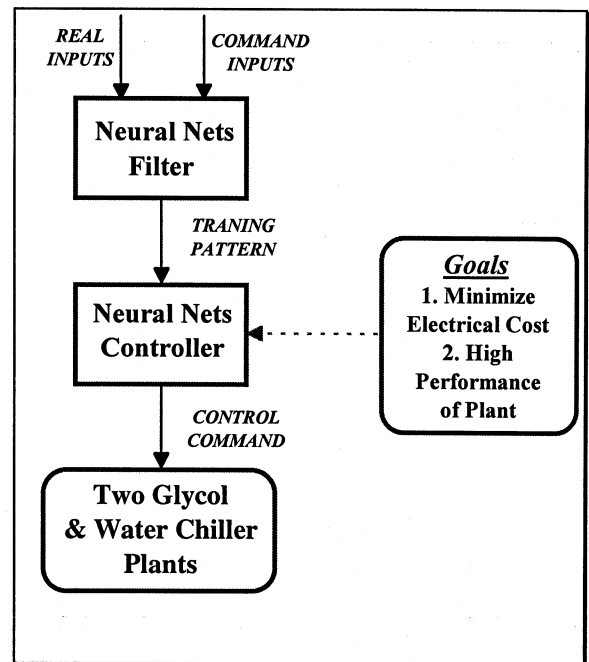
จากผลการทดลองที่ได้สามารถสรุป เป็นข้อๆ ได้ดังนี้

1. จากการทดสอบ พบว่า การเรียนรู้แบบแพร่กระจายกลับ (Back - Propagation) สามารถใช้ได้ทั้งข้อมูลแบบ analog และ Digital
2. ค่าเอพ็อด ได้จากข่ายงานระบบประสาท ถึงแม้ว่าจะมีค่า เป็น 0 หรือ 1 แต่เราสามารถทำการปรับค่าโดย ให้ ค่าอยู่ระหว่าง 0 - 0.5 มีค่า เป็น 0

ค่าอยู่ระหว่าง 0.5 - 1 มีค่าเป็น 1

3. จำนวนโหนดในชั้นฮิดเดน ควรมี ค่าเป็น 2 เท่า ของจำนวน อินพุตโหนด จึงจะเหมาะสม
4. การให้ค่าโหนดในชั้นฮิดเดนมากๆ แม้ว่าจะมีผลให้ค่าความผิดพลาดระหว่างค่าเอพ็อดที่ต้องการ กับค่าเอพ็อดที่ผ่านข่ายงานระบบประสาท น้อยลง แต่เวลาที่ใช้ในการสอนข่ายงานระบบประสาท ก็จะนาน ขึ้นมา ดังนั้น ควรเลือกจำนวนโหนด ที่เหมาะสม ในขณะที่ค่าความผิดพลาด ,J ยอมรับได้

หลังจากที่ได้แบบจำลองพฤติกรรมของอุปกรณ์ทำน้ำเย็น Water และ Glycol Chiller โดยใช้ข่ายงานระบบประสาทที่เหมาะสมแล้ว ต่อไปเราจะทำการสร้างแบบจำลองของระบบทำความเย็นของทั้งระบบ Glycol & Water Chiller Plant โดยให้ข้อมูลอินพุตของระบบ คือ วัน,เวลา การทำความเย็นของตึก ส่วนเอพ็อดของระบบ คือ การเปิด-ปิดอุปกรณ์ต่างๆ ในระบบทำความเย็น Glycol & water Chiller Plant เมื่อได้ แบบจำลองของระบบทำความเย็น Glycol & water Chiller Plant ที่เหมาะสมแล้ว (Neural Network Emulator) ขึ้นต่อไปคือการสอนตัวควบคุมระบบ (Neural Net Controller) โดยมีเป้าหมายการควบคุมตามที่ต้องการ หลังจากที่ได้ตัวควบคุมผ่านกระบวนการเรียนรู้แล้ว จากนั้นนำตัวควบคุม (Neural Network Controller)ที่ได้ ไปควบคุมระบบ Glycol & Water Chiller Plant วิธีการควบคุมระบบทำความเย็น แบบ Glycol & Water Chiller Plant สามารถสรุปได้ดังรูปที่ 9



รูปที่ 9 รูปแบบวิธีการควบคุม Glycol & Water Chiller Plant

งานขั้นต่อไป มีดังต่อไปนี้

- ออกแบบ ข่ายงานระบบประสาท ที่สามารถวิเคราะห์ข้อมูลที่นำมาสอน เพื่อแยกแยะข้อมูลอินพุตที่ถูกต้อง กับข้อมูลที่ผิดเพี้ยนไป
- ออกแบบตัวควบคุมแบบข่ายงานระบบประสาท เพื่อใช้ควบคุมการทำงานของ Glycol & Water Chiller Plant ให้ทำงานโดยมีค่าไฟฟ้าต่ำที่สุด ขณะที่ยังรักษาประสิทธิภาพการทำงานไว้สูงสุด
- นำข่ายงานระบบประสาท ที่เสร็จสมบูรณ์ มาประยุกต์ใช้งานจริงที่ ระบบทำความเย็นแบบ Glycol & Water Chiller ที่ธนาคารไทยพาณิชย์สำนักงานใหญ่

สุดท้ายนี้เราพบว่าการควบคุมระบบโดยใช้ข่ายงานระบบประสาท มีความยืดหยุ่นกว่าการใช้ระบบควบคุมแบบเชิงเส้น กล่าวคือ สามารถทำงานได้แม้ในสภาวะแวดล้อมที่มีความผิดเพี้ยน อีกทั้งยังสามารถเพิ่มความฉลาดให้กลับตัวควบคุมได้ โดยการเพิ่มรูปแบบข้อมูลในการสอนข่ายงานระบบประสาทเข้าไปอีก คุณสมบัติข้อนี้ไม่สามารถหาได้จากโปรแกรมควบคุมชนิดอื่น เนื่องจากการโปรแกรมแบบอื่นจะต้องใช้ความชำนาญในการแก้ไข และเสียเวลามาก

กิตติกรรมประกาศ

บทความนี้ได้รับการสนับสนุน ด้านข้อมูลทางระบบทำความเย็น Glycol & Water Chiller Plant จากบริษัท มหิศร จำกัด และ ขอขอบคุณ ผศ. ดร. ชิต เหล่าวัฒนา ที่ให้การสนับสนุน และผลักดันให้บทความนี้สำเร็จลุล่วงด้วยดี

เอกสารอ้างอิง

- [1] B. Widrow and M.E. Hoff, Jr **"Adaptive Switching Circuits."** In 1960 *IRE WESTCON Conv Record*, Part 4, pp. 96 - 104, 1960
- [2] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, **"Learning Internal Representations by Error Propagation"** in D.E. Rumelhart and J. L. McClelland, Eds., *Parallel Distributed Processing*, vol. 1, chap. 8, Cambridge, MA: MIT Press, 1986.
- [3] B. Widrow and S. D. Stearns, **"Adaptive Signal Processing"**, Englewood Cliffs. NJ: Prentice - Hall, 1985.
- [4] D. B. Parker, **"learning Logic"**, Tech. Rept. TR-47, Center for Comput. Res. Econ. And Manage., Massachusetts Institute of Technology, Cambridge, MA, Aug. 1985.
- [5] P. Werbos, **"Beyond Regression : New Tools for Prediction and analysis in the Behavioral Sciences"**, Ph. D. Thesis, Harvard Univ., Cambridge, MA Aug. 1974.
- [6] Nguyen D.H. and B. Widrow, **"Neural Network for self-learning control systems"**, *IEEE Control Syst. Mag.*, 10, 18-23, 1990.
- [7] James A Freeman, **" Neural Network : Algorithms ,Application and Programming Technoques"** , Addison - Wesley , Reading, MA, 1990, Pages 104.
- [8] J.M.Zurada, **"Introduction to Artificial Neural System"**, west publishing company, Singapore, 1992.
- [9] Hornik, K. M. Stinchcombe, **"Multi - Layer Feedforward Networks are universal Approximators"**, *Neural Network2*, 1989, Pages 359 - 366.
- [10] B. Widrow and E. Walach, **"Adaptive Inverse Control"** , in Appendix H : *Neural Control System*, Englewood Cliffs. NJ: Prentice - Hall, 1985, pages 491 - 493.
- [11] A. M. S. Zalzala and A. S. Morris **"Neural Networks for Robotic Control Theory and Application"** , Ellis Horwood Limited, 1996 , Pages 17 - 20.