รายการอ้างอิง

ภาษาอังกฤษ

Chua, L.O., and Lin, G.N. 1984. Nonlinear Programming without Computation. **IEEE Transaction on Circuit and System**. 31: 182-188.

Cichocki, A., and Unbehauen, R. 1993. **Neural Networks for Optimization and Signal Processing. Chichester**: Wiley & Sons.

Fausett, L. 1994. **Fundamentals of Neural Networks Architecture, Algorithm, and Applications**. USA.: Prentice Hall International Inc.

Glasos, M.P., Hui, S., and Zak, S.H. 1994. On Solving Constrained Optimization Problems with Neural Networks. **IEEE Conference on Neural Network**. 7: 4547-4552.

Hopfield, J.J., and Tank, D.W. 1986. Simple 'neural' optimization network and a/d converter signal decision circuit and a linear programming circuit. **IEEE Transaction on Circuit and System**. 33: 533-541.

Kai, H. 1993. **Advanced Computer Architecture**. Singapore: McGraw-Hill International Inc.

Kennedy, M.P., and Chua, L.O. 1988. Neural Networks for Nonlinear Programming. **IEEE Transaction on Circuit and System**. 35: 554-562.

Prithviraj, B. 1994. **Parallel Algorithms for VLSI Computer-Aided Design**. USA.: Prentice Hall International Inc.

Rao, S.S. 1989. **Optimization Theory and Application**. India: Wiley Eastern Limited.

Rodri′gues-Va′zquez, A., Domi′nguez-Castro, R., Rueda, A., Huertas, J.L., and Sa′nchez-Sinencio, E. 1990.Nonlinear Switched-Capacitor Neural Networks for Optimization Problems. **IEEE Transaction on Circuit and System**. 37: 384-398.

Schalkoff, R. 1992. **Pattern Recognition**. Canada: Wiley & Sons.

Utkin, V.I. 1978. **Sliding Modes and Their Application in Variable Structure Systems**. Moscow: MIR Publishers.

Zurada, J.M. 1992. **Introduction to Artificial Neural Systems**. Singapore: West Publishing Company.

ภาคผนวก

ภาคผนวก ก.

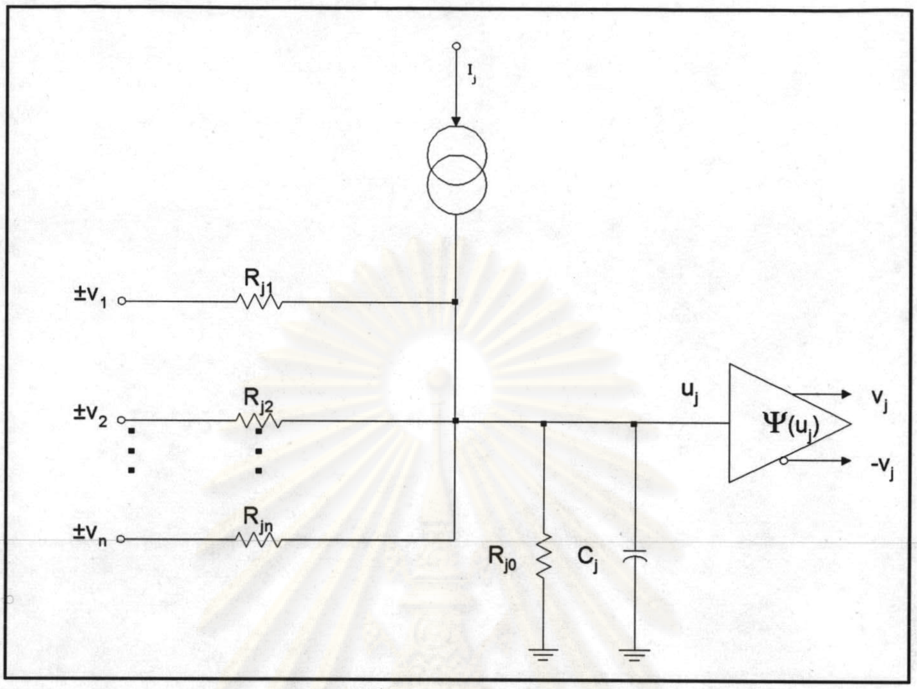
รูปแบบพื้นฐานของรูปแบบฮอปฟิลด์

ความนำ

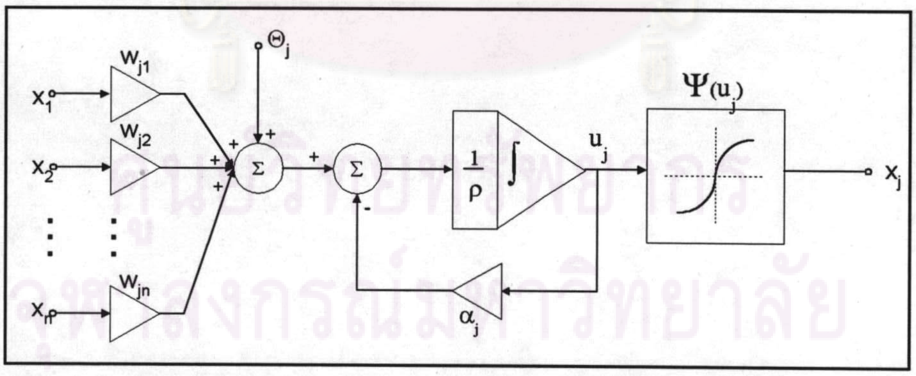
ในภาคผนวกนี้ กล่าวถึงรูปแบบพื้นฐานของรูปแบบฮอปฟิลด์ซึ่งเป็นรูปแบบหนึ่งของเครือข่ายนิวรอลที่นิยมใช้ในการแก้ปัญหาการหาค่าออปติมัม รูปแบบฮอปฟิลด์นี้มีโครงสร้างที่ง่ายและมีลักษณะการทำงานที่ขนานกัน ซึ่งในที่นี้แสดงในรูปของโครงสร้างวงจรไฟฟ้าและโครงสร้างทางบล็อคไดอะแกรม

รูปแบบพื้นฐานของรูปแบบฮอปฟิลด์

Hopfield Model เป็นรูปแบบหนึ่งของเครือข่ายนิวรอล (Neural Networks) ที่นิยมใช้ในการแก้ปัญหาการหาค่าออปติมัม ซึ่งแสดงในรูปที่ ก.1 และรูปที่ ก.2 โดยแสดงถึงโครงสร้างทางวงจรไฟฟ้าและโครงสร้างทางบล็อคไดอะแกรม โครงสร้างทางวงจรไฟฟ้าประกอบด้วยตัวความจุไฟฟ้า (Capacitor) c_j , ตัวความต้านทาน (Resistor) R_{ji} และตัวขยายไม่เชิงเส้น (Nonlinear Amplifier) หรือ Sigmoid activation function ซึ่งถ้าค่าน้ำหนัก (Weight) มีค่าเป็นบวก ตัวความต้านทาน R_{ji} จะต่อกับ $+v$ และถ้าค่าน้ำหนักมีค่าเป็นลบ ตัวความต้านทาน R_{ji} จะต่อกับ $-v$ โดยให้ กระแส I_j เป็น Bias



รูปที่ ก.1 วงจรไฟฟ้าของเซลนิวรอลแบบฮอปฟิลด์



รูปที่ ก.2 บล็อคไดอะแกรมของเซลนิวรอลแบบฮอปฟิลด์

จาก Ohm's Law และ Kirchoff's Current Law จะได้ว่า

$$c_j \frac{du_j}{dt} = \frac{-u_j}{R_j} + \sum_{i=1}^{n} \frac{v_i}{R_{ji}} + I_j \qquad (ก.1)$$

โดยที่

$$v_j = \Psi(u_j) \qquad , j = 1,2,\ldots,n$$

$$\frac{1}{R_j} = \frac{1}{R_{j0}} + \sum_{i=1}^{n} \frac{1}{R_{ji}}$$

$$= G_{jo} + \sum_{i=1}^{n} G_{ji}$$

จากสมการที่ (ก.1) สามารถเขียนให้อยู่ในรูปทั่วไปได้ดังนี้คือ

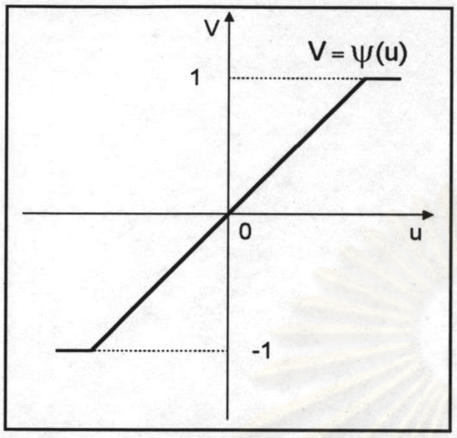
$$\rho \frac{du_j}{dt} = -\alpha_j u_j + \left(\sum_{i=1}^{n} w_{ji} x_i + \Theta_j \right)$$

โดยที่

$$x_j = \Psi(u_j) \qquad , j = 1,2,\ldots,n$$

ซึ่ง Ψ คือ Sigmoid activation function โดยมีรูปแบบตามรูปที่ ก.3(1) -ก.3(4)

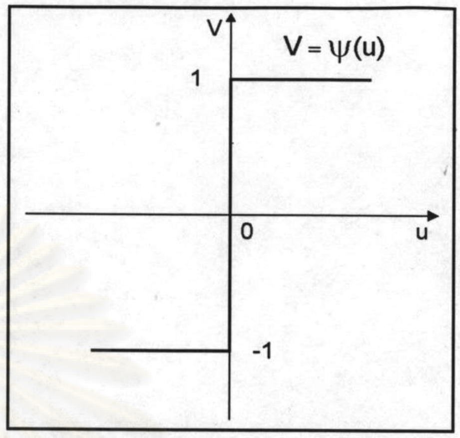
(1)



Saturation limiter

(Linear Lamp)

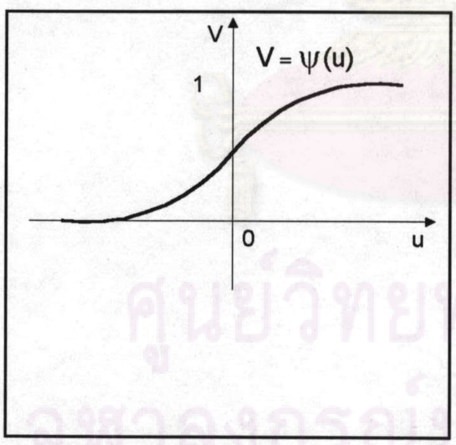
(2)



Hard limiter

(Signum function)

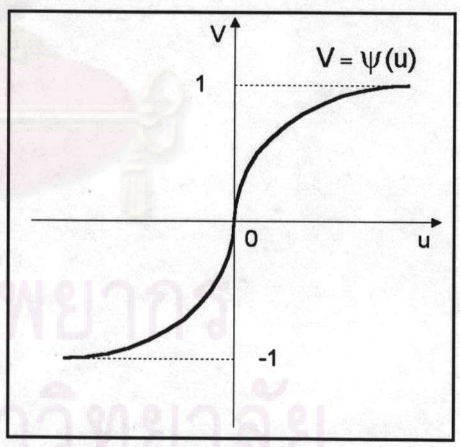
(3)



Sigmoid limiter

(Unipolar)

(4)

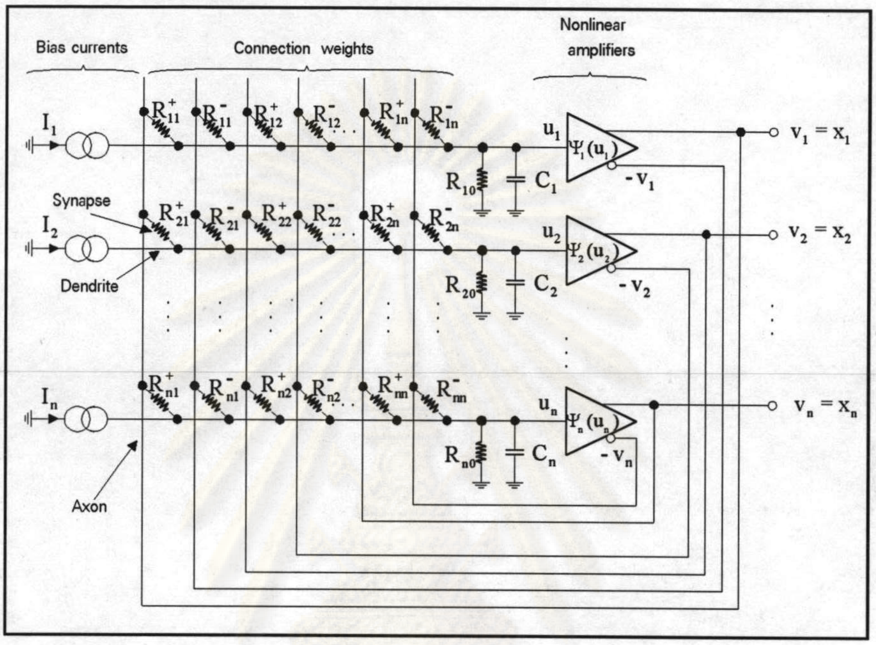


Sigmoid limiter

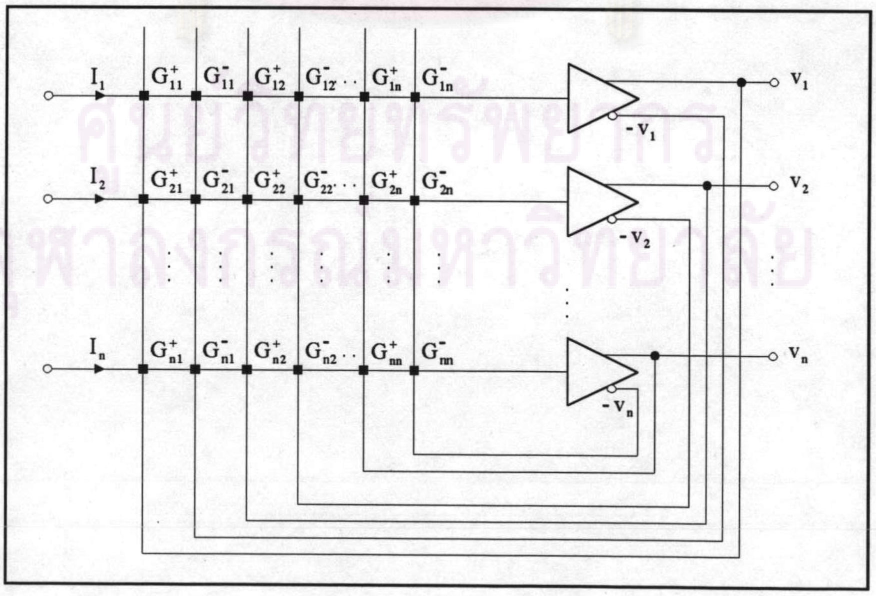
(Bipolar)

รูปที่ ก.3 Sigmoid activation functions แบบต่างๆที่ใช้ในเครือข่ายนิวรอล

รูปแบบพื้นฐานของ Hopfield Model ประกอบด้วยตัวความจุไฟฟ้า,ตัวความต้านทาน, ตัวขยายไม่เชิงเส้น และแหล่งกำเนิดกระแสไฟฟ้า (Bias current sources) ดังแสดงในรูปที่ ก.4 และรูปที่ ก.5



รูปที่ ก.4 เครือข่ายนิวรอลแบบฮอปฟิลด์



รูปที่ ก.5 โครงสร้างแบบง่ายๆของรูปแบบฮอปฟิลด์

จากรูปที่ ก.4 จะได้ว่า

$$c_j \frac{du_j}{dt} = \sum_{i=1}^{n} G_{ji}(x_i - u_j) + I_j - \frac{u_j}{R_{j0}}$$

โดยที่

$$x_j = \Psi_j(u_j)$$

ซึ่งสามารถเขียนใหม่เป็น

$$c_j \frac{du_j}{dt} = -\frac{u_j}{R_j} + \sum_{i=1}^{n} G_{ji} x_i + I_j \quad , j = 1,2,\ldots,n$$

ซึ่ง $c_j > 0$ เป็นตัวความจุไฟฟ้า

u_j และ x_j เป็นแรงดันไฟฟ้าเข้า และแรงดันไฟฟ้าออกของนิวรอลตัวที่ j

G_{ji} เป็น Conductance จากนิวรอลตัวที่ i ไปยังนิวรอลตัวที่ j

$$G_{ji} = \frac{1}{R_{ji}^{+}} - \frac{1}{R_{ji}^{-}} \quad \text{ซึ่ง } R_{ji}^{\pm} > 0$$

หรือ

$$G_{ji} = \begin{cases} \dfrac{1}{R_{ji}^{+}} & \text{if } G_{ji} > 0 \text{ with } R_{ji}^{-} = \infty \\ -\dfrac{1}{R_{ji}^{-}} & \text{if } G_{ji} < 0 \text{ with } R_{ji}^{+} = \infty \end{cases}$$

$$\frac{1}{R_j} = \frac{1}{R_{j0}} + \sum_{i=1}^{n}\left(\frac{1}{R_{ji}^{+}} + \frac{1}{R_{ji}^{-}}\right)$$

และ $\Psi_j(u_j)$ คือ Monotonically increasing activation function สามารถเขียนเป็น

$$\Psi_j(u_j) = \left[1 + e^{-\gamma_j u_j}\right]^{-1} \quad \text{or} \quad \Psi_j(u_j) = \tanh(\gamma_j u_j)$$

ตารางที่ ก.1 การเปรียบเทียบข้อดีและข้อเสียของ Sigmoid activation function **ที่**นิยมใช้ทั่วไป

Activation Function	ข้อดี	ข้อเสีย
Hard Limter (Signum Function)	-เร็วและง่ายต่อการนำไปใช้ในรูปฮาร์ดแวร์	-ไม่สามารถหาส่วนกลับได้ -ไม่มีส่วนที่เป็นเชิงเส้น -ไม่สามารถสร้างฟังก์ชันที่เรียบได้
Saturation Limiter (Linear Lamp)	-มีส่วนที่เป็นเชิงเส้น -สามารถสร้างฟังก์ชันที่เรียบได้ -ง่ายต่อการไปใช้งานจริง	-ไม่สามารถหาส่วนกลับได้
Sigmoid Limiter (Unipolar)	-สามารถหาส่วนกลับได้ -มีส่วนที่สามารถสร้างฟังกชันที่เรียบได้	-ยากต่อการนำไปใช้ในทางปฏิบัติ -ค่าสัญญาณออกจะมีแต่เฉพาะค่าบวก
Sigmoid Limiter (Bipolar)	-สามารถหาส่วนกลับได้ -มีส่วนที่สามารถสร้างฟังก์ชันที่เรียบได้ -ค่าสัญญาณออกจะมีแต่เฉพาะค่าบวกและค่าลบ	-ยากต่อการนำไปใช้ในทางปฏิบัติ

ภาคผนวก ข.

การหาค่าออปติมัมแบบพื้นฐาน

ความนำ

ในภาคผนวกนี้แบ่งเป็น 2 ส่วนคือ ส่วนแรกกล่าวถึงการหาค่าออปติมัมแบบพื้นฐานซึ่งวิธีที่ใช้ในการหาค่าออปติมัมมี 4 วิธีคือ วิธี Steepest Descent , วิธี Newton , วิธี Quasi-Newton และวิธี Conjugate Gradient และส่วนที่สองกล่าวถึงรูปแบบทางคณิตศาสตร์ของเครือข่ายนิวรอลซึ่งสามารถจัดให้อยู่ในรูปแบบที่ต่อเนื่องทางเวลาคือในรูปของสมการอนุพันธ์ และรูปแบบที่ไม่ต่อเนื่องทางเวลาคือในรูปของสมการผลต่าง

การหาค่าออปติมัมแบบพื้นฐาน

ถ้าปัญหาการหาค่าออปติมัมอยู่ในรูปของ

หา $\mathbf{x}$ ซึ่งทำให้ $f(\mathbf{x})$ มีค่าต่ำสุด

โดยที่ $\mathbf{x} \in \Re^n$ (ข.1)

ถ้าสมการที่ (ข.1) สามารถหา Gradient vector และ Hessian matrix ของ Cost function ได้และถ้าให้ $\mathbf{x}^{(k)}$ เป็นจุดค้นหา (Search point) จะได้ว่า

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \eta^{(k)}\mathbf{d}_k, \quad k = 0,1,2,\ldots$$

โดยที่ $\mathbf{x}^{(0)} = \mathbf{x}(0)$ และพารามิเตอร์ $\eta^{(k)} > 0$ โดยก้าวไปในทิศทางของเวคเตอร์ $\mathbf{d}_k \cong \Delta\mathbf{x}^{(k)}$

การหาค่าออปติมัมแบบพื้นฐานมีวิธีที่ใช้ในการหาค่าออปติมัม 4 วิธีคือ

1. วิธี Steepest Descent (วิธี Gradient) ซึ่ง

$$\mathbf{d}_k = -\nabla f(\mathbf{x}^{(k)})$$

2. วิธี Newton ซึ่ง

$$\mathbf{d}_k = -\left[\nabla f(\mathbf{x}^{(k)})\right]^{-1} \nabla f(\mathbf{x}^{(k)})$$

3. วิธี Quasi-Newton ซึ่ง

$$\mathbf{d}_k = -\left[\nabla^2 f(\mathbf{x}^{(k)})\right]^{-1} \nabla f(\mathbf{x}^{(k)})$$

4. วิธี Conjugate Gradient ซึ่ง

$$\mathbf{d}_k = -\nabla f(\mathbf{x}^{(k)}) + \beta_k \mathbf{d}_{k-1} \quad , \; k = 1,2,\ldots$$

โดยที่

$$\mathbf{d}_0 = -\nabla f(\mathbf{x}^{(0)})$$

และ

$$\beta_k = \left(\frac{\|\nabla f(\mathbf{x}^{(k)})\|}{\|\nabla f(\mathbf{x}^{(k-1)})\|} \right)^2$$

วิธีการหาค่าออปติมัมแบบพื้นฐานนี้มีข้อดีข้อเสียคือ วิธี Steepest Descent เป็นวิธีที่ง่ายมากและประหยัดในการที่จะ Implement จริง แต่มีข้อเสียคือลู่เข้าหาค่าออปติมัมช้ามาก ส่วนวิธี Newton มีการเพิ่มอัตราการลู่เข้าหาค่าออปติมัมแต่การนำไปใช้ในการ Implement จริงนั้นสิ้นเปลือง และต้องสามารถหาค่า Inverse Hessian matrix ซึ่งยากและบางปัญหาเป็น Singular matrix ส่วนวิธี Quasi-Newton ต้องการหน่วยความจำในการเก็บค่าต่างๆ ของ Hessian matrix ในแต่ละรอบ ส่วนวิธี Conjugate Gradient ต้องการหน่วยความจำในการเก็บข้อมูลน้อยกว่าวิธี Quasi-Newton แต่ต้องหาค่าพารามิเตอร์ $\eta^{(k)}$,คำนวณค่า $\beta_{(k)}$ และต้องหาค่า Gradient 2 ครั้งหรือมากกว่านั้น แต่วิธีนี้ประหยัดเวลาและหน่วยความจำ

Basic Gradient System ของเครือข่ายนิวรอล

จากสมการที่ (ข.1) สามารถจัดให้อยู่ในรูปแบบทางคณิตศาสตร์ของเครือข่ายนิวรอลโดยอยู่ในรูปของสมการอนุพันธ์ (Differential equation) คือ

$$\frac{dx_j}{dt} = -\lambda(t)\frac{\partial f}{\partial x_j} \qquad (ข.2)$$

ซึ่ง

$$x_j(0) = x_j^{(0)} \quad , \; j = 1,2,\ldots,n$$

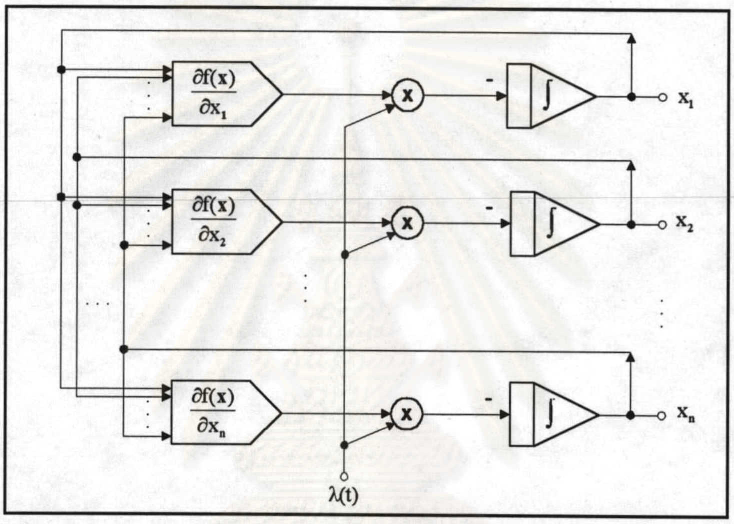
$$\lambda(t) > 0 \quad , \; \lambda(t) \in C^1$$

โดยที่ $\lambda(t)$ เป็นฟังก์ชันสเกลาร์บวก (Positive scalar function) ซึ่งเป็นค่าคงที่ และสมการที่ (ข.2) เรียกว่า basic dynamic gradient system โดยที่รูปที่ ข.1 เป็นบล็อคไดอะแกรมของเครือข่ายนิวรอลสำหรับการหาค่าออปติมัมแบบต่อเนื่องทางเวลา (Continue-time) และรูปที่ ข.2 เป็นบล็อคไดอะแกรมของเครือข่ายนิวรอลสำหรับการหาค่าออปติมัมแบบไม่ต่อเนื่องทางเวลา (Discrete-time) ซึ่งการหาค่าออปติมัมแบบไม่ต่อเนื่องทางเวลาสามารถเขียนในรูปของสมการผลต่าง (Difference equations) คือ

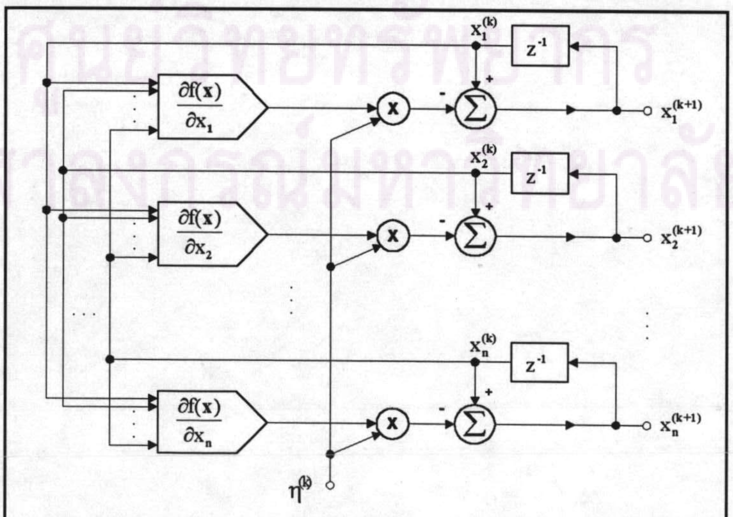
$$x_j^{(k+1)} = x_j^{(k)} - \eta^{(k)} \frac{\partial f(\mathbf{x}^{(k)})}{\partial x_j} \qquad (ข.3)$$

ซึ่ง $\eta^{(k)} > 0 \quad , k = 0,1,2,\ldots$

การใช้อัลกอริทึมการทำซ้ำของสมการที่ (ข.3) อาจทำให้เกิดการซิกแซก (zigzag) ซึ่งสามารถแก้ไขได้โดยใช้ Continuous-time gradient system



รูปที่ ข.1 เครือข่ายนิวรอลแบบต่อเนื่องทางเวลา



รูปที่ ข.2 เครือข่ายนิวรอลแบบไม่ต่อเนื่องทางเวลา

ภาคผนวก ค.

การหาค่าออปติมัมแบบมีเงื่อนไขด้วยเครือข่ายนิวรอล

ความนำ

ในภาคผนวกนี้แบ่งเป็น 2 ส่วนคือ ส่วนแรกกล่าวถึงวิธีการหาค่าออปติมัมแบบมีเงื่อนไขด้วยเครือข่ายนิวรอล การหาค่าออปติมัมนี้ใช้วิธี Penalty function แปลงจากปัญหาการหาค่าออปติมัมแบบมีเงื่อนไขให้เป็นปัญหาการหาค่าออปติมัมแบบไม่มีเงื่อนไข จากนั้นแปลงปัญหาการหาค่าออปติมัมแบบไม่มีเงื่อนไขให้เป็นปัญหาของการแก้สมการอนุพันธ์ของตัวแปรสถานะ ซึ่งมี Forcing function เป็นฟังก์ชันของ Steepest descent direction ผลตอบของสมการอนุพันธ์เป็นผลตอบของปัญหาการหาค่าออปติมัมแบบไม่มีเงื่อนไข และผลตอบของสมการอนุพันธ์นี้หาได้จากการจำลองสมการอนุพันธ์ด้วยวงจรไฟฟ้า โดยที่สมการอนุพันธ์ดังกล่าวสามารถทำเป็น Parallel processing ได้ นั้นคือสามารถจัดให้อยู่ในรูปแบบทางคณิตศาสตร์ของเครือข่ายนิวรอล และวงจรไฟฟ้านี้เปรียบเสมือนเครือข่ายนิวรอล โดยที่ Integrators คือตัวนิวรอล(Neurons) และ Functional nonlinear generators สร้างขึ้นมาเพื่อเชื่อมต่อระหว่างตัวนิวรอลเหล่านี้ และส่วนที่สองกล่าวถึงการพิสูจน์การลู่เข้าหาค่าออปติมัมของวิธี Steepest Descent

การหาค่าออปติมัมแบบมีเงื่อนไขด้วยเครือข่ายนิวรอล

วิธีหนึ่งในการแก้ปัญหาการหาค่าออปติมัมแบบมีเงื่อนไขคือวิธี Penalty function ซึ่งแปลงจากปัญหาการหาค่าออปติมัมแบบมีเงื่อนไขให้เป็นปัญหาการหาค่าออปติมัมแบบไม่มีเงื่อนไข

ถ้าปัญหาการหาค่าออปติมัมแบบมีเงื่อนไขอยู่ในรูปของ

หา $\mathbf{x}$ ซึ่งทำให้ $f(\mathbf{x})$ มีค่าต่ำสุด

โดยที่ $g_i(\mathbf{x}) \le 0 \quad , i = 1,2,\ldots,m$

สามารถแปลงให้เป็นปัญหาการหาค่าออปติมัมแบบไม่มีเงื่อนไขคือ

หา $\mathbf{x}$ ซึ่งทำให้ $p(\mathbf{x})$ มีค่าต่ำสุด (ค.1)

โดยที่ $\mathbf{x} \in \Re^n$, $f : \Re^n \to \Re$, $\mu_i > 0$, $p(\mathbf{x}) = f(\mathbf{x}) + \sum_{i=1}^{m} \mu_i q_{ig}(g_i(\mathbf{x}))$ และ $g_i : \Re^n \to \Re$, $i = 1,2,\ldots,m$

สังเกตได้ว่าฟังก์ชัน $p(\mathbf{x})$ ประกอบด้วย Objective function $f(\mathbf{x})$ และพจน์ของ $q_{ig}(g_i(\mathbf{x}))$ ซึ่งค่า $p(\mathbf{x})$ จะเพิ่มขึ้นเมื่อมีการละเมิดเงื่อนไขนั้นคือ $\{g_i(\mathbf{x}) > 0\}$, $i = 1,2,\ldots,m$ โดยถ้ายิ่งมีการละเมิดเงื่อนไขมากเท่าไร ค่าฟังก์ชัน $p(\mathbf{x})$ ยิ่งเพิ่มขึ้นเท่านั้น ซึ่งรูปแบบทั่วไปของ $q_{ig}(g_i(\mathbf{x}))$ มีดังต่อไปนี้คือ

$$q_{ig}(g_i(\mathbf{x})) = -\min\{0,-g_i(\mathbf{x})\} ,\ i = 1,2,\ldots,m \qquad (ค.2)$$

หรือ

$$q_{ig}(g_i(\mathbf{x})) = \frac{1}{2}\left[\min\{0,-g_i(\mathbf{x})\}\right]^2 ,\ i = 1,2,\ldots,m \qquad (ค.3)$$

สังเกตได้ว่าถ้ามีการละเมิดเงื่อนไขจะทำให้ $\min\{0,-g_i(\mathbf{x})\} = -g_i(\mathbf{x}) < 0$, $i = 1,2,\ldots,m$ ดังนั้นสมการที่ (ค.2) เท่ากับ $q_{ig}(g_i(\mathbf{x})) = g_i(\mathbf{x}) > 0$, $i = 1,2,\ldots,m$ และสมการที่ (ค.3) เท่ากับ $q_{ig}(g_i(\mathbf{x})) = \frac{1}{2}\left[\min\{0,-g_i(\mathbf{x})\}\right](-g_i(\mathbf{x})) = \frac{1}{2}(g_i(\mathbf{x}))^2 > 0$, $i = 1,2,\ldots,m$ และถ้าไม่มีการละเมิดเงื่อนไขจะทำให้ $\min\{0,-g_i(\mathbf{x})\} = 0$, $i = 1,2,\ldots,m$ ซึ่งทำให้สมการที่ (ค.2) และสมการที่ (ค.3) เท่ากับ $q_{ig}(g_i(\mathbf{x})) = 0$, $i = 1,2,\ldots,m$

เมื่อนำสมการที่ (ค.3) แทนลงในสมการที่ (ค.1) จะได้ว่า

$$p(\mathbf{x}) = f(\mathbf{x}) + \frac{1}{2}\sum_{i=1}^{m}\mu_i\left[\min\{0,-g_i(\mathbf{x})\}\right]^2 \qquad (ค.4)$$

โดยที่ $\mu_i > 0$ และถ้าสมการเงื่อนไข $g_i(\mathbf{x})$ สามารถหาอนุพันธ์ย่อยอันดับหนึ่งได้

$$\tilde{g}_i(\mathbf{x}) = \frac{1}{2}\left(\min\{0,-g_i(\mathbf{x})\}\right)^2 := \frac{1}{2}\left(\left[-g_i(\mathbf{x})\right]_-\right)^2$$

$$\frac{\partial \tilde{g}_i(\mathbf{x})}{\partial \mathbf{x}_j} = -\left[-g_i(\mathbf{x})\right]_- \frac{\partial g_i(\mathbf{x})}{\partial \mathbf{x}_j} \quad \text{สำหรับทุกๆค่า} \quad \mathbf{x} \in \Re^n \qquad (ค.5)$$

จากสมการที่ (ค.4) สามารถใช้ Gradient หาค่าออปติมัม ซึ่งสามารถจัดให้อยู่ในรูปของสมการอนุพันธ์ โดยจากสมการอนุพันธ์ดังกล่าวและจากสมการที่ (ค.5) สามารถเขียนได้ดังนี้คือ

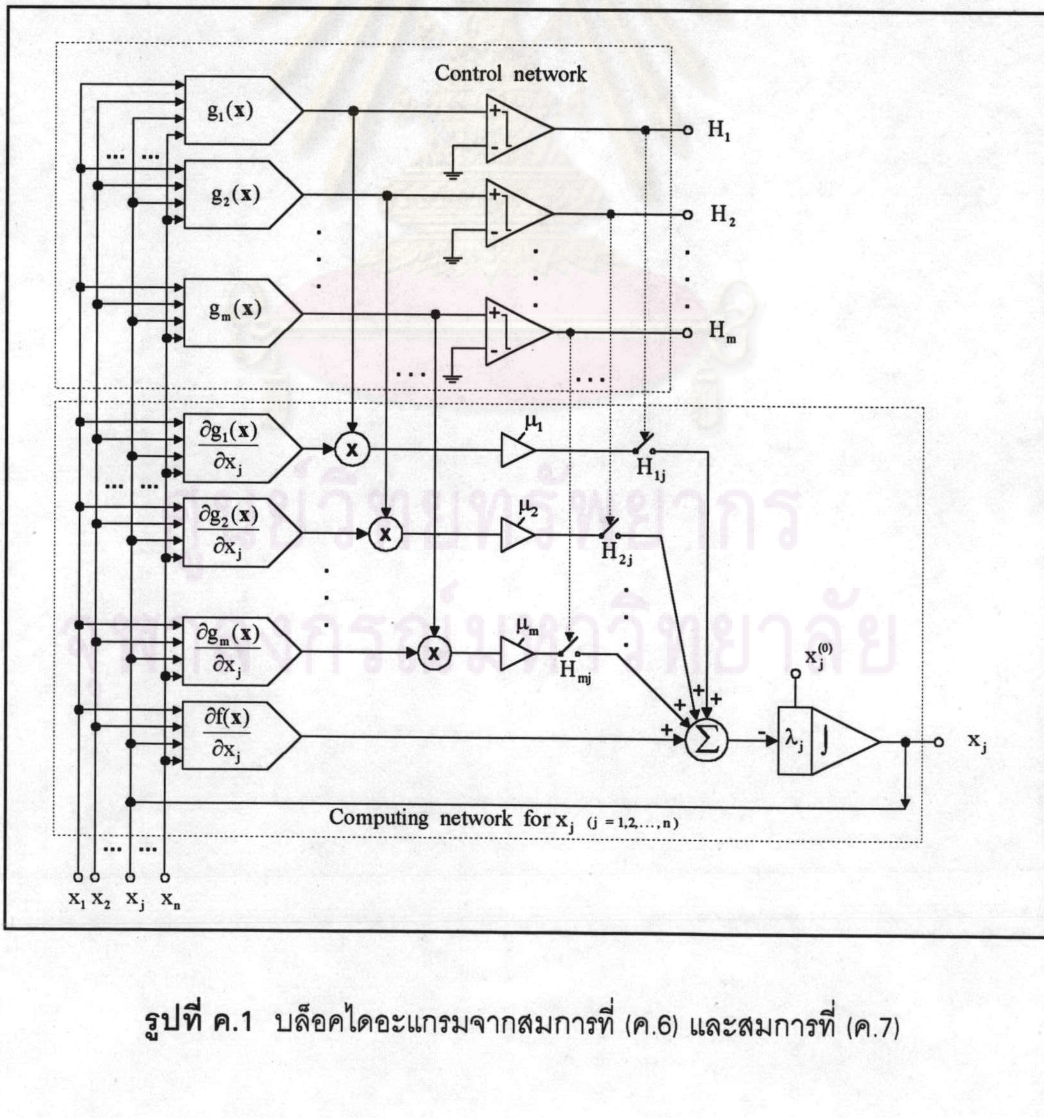
$$\frac{dx_j}{dt} = -\lambda_j\left[\frac{\partial f(\mathbf{x})}{\partial x_j} + \sum_{i=1}^{m}\mu_i H_i g_i(\mathbf{x})\frac{\partial g_i(\mathbf{x})}{\partial x_j}\right] \quad , \; j = 1,2,\ldots,n \qquad \text{(ค.6)}$$

ซึ่ง $\lambda_j > 0 \;\; , \; \mu_j > 0$

และ

$$H_i = \begin{cases} 1 & \text{if } g_i(\mathbf{x}) > 0 \\ 0 & \text{if } g_i(\mathbf{x}) \le 0 \end{cases} \qquad \text{(ค.7)}$$

จากสมการข้างบนสามารถนำมา Implement ในรูปที่ ค.1 โดยที่บล็อคไดอะแกรมนี้ได้มาจากสมการที่ (ค.6) และสมการที่ (ค.7) วงจรไฟฟ้านี้เปรียบเสมือนเครือข่ายนิวรอล โดยที่ Integrators คือตัวนิวรอล (Neurons) และ Functional nonlinear generators สร้างขึ้นมาเพื่อเชื่อมต่อระหว่างตัวนิวรอลเหล่านี้



รูปที่ ค.1 บล็อคไดอะแกรมจากสมการที่ (ค.6) และสมการที่ (ค.7)

จากสมการที่ (ค.1) สามารถใช้ gradient หาค่าออปติมัม โดยจัดให้อยู่ในรูปของสมการอนุพันธ์ จากนั้นนำสมการที่ (ค.2) แทนลงในสมการอนุพันธ์นั้น ซึ่งสามารถเขียนได้ดังนี้คือ

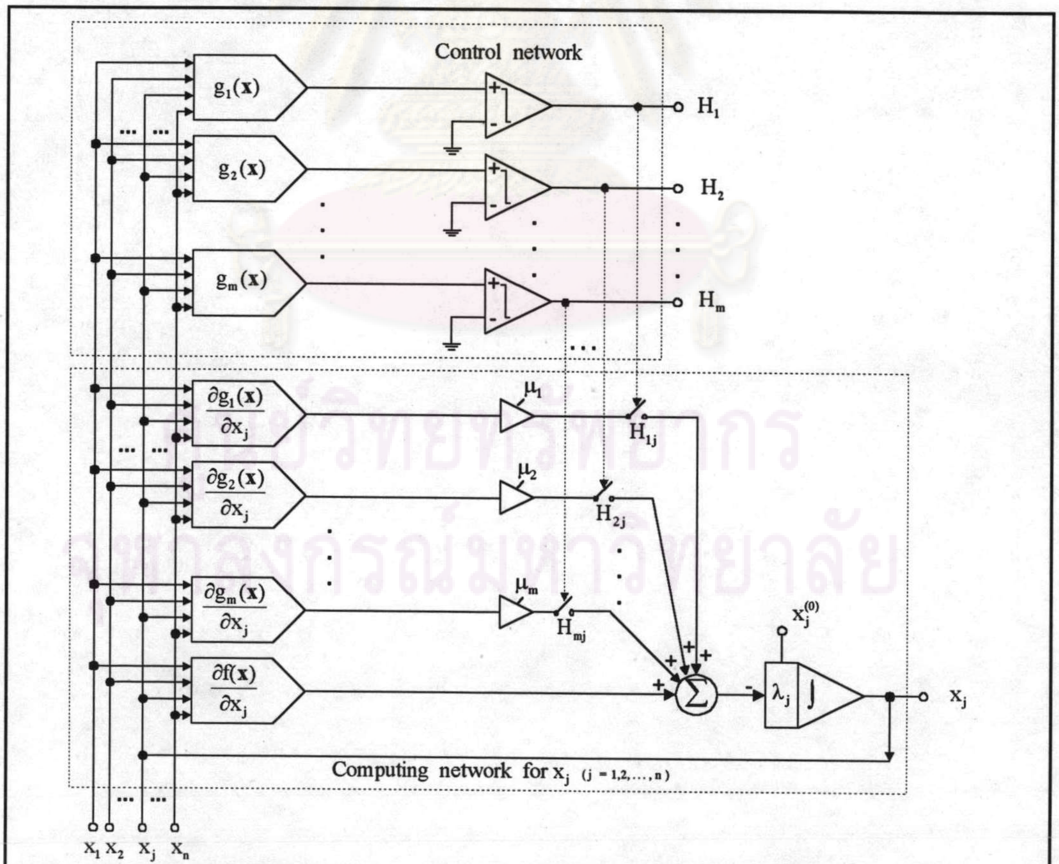
$$\frac{dx_j}{dt} = -\lambda_j\left[\frac{\partial f(\mathbf{x})}{\partial x_j} + \sum_{i=1}^{m}\mu_i H_i \frac{\partial g_i(\mathbf{x})}{\partial x_j}\right] \quad , j = 1,2,\ldots,n \qquad (ค.8)$$

ซึ่ง

$$\lambda_j > 0 \quad , \mu_j > 0 \quad \text{และ}$$

$$H_i = \begin{cases} 1 & \text{if} \quad g_i(\mathbf{x}) > 0 \\ 0 & \text{if} \quad g_i(\mathbf{x}) \le 0 \end{cases} \qquad (ค.9)$$

จากสมการที่ (ค.8) และสมการที่ (ค.9) สามารถนำมาเขียนในรูปบล็อคไดอะแกรมดังรูปที่ ค.2



รูปที่ ค.2 บล็อคไดอะแกรมจากสมการที่ (ค.8) และสมการที่ (ค.9)

การพิสูจน์การลู่เข้าหาค่าออปติมัมของวิธี Steepest Descent

การพิสูจน์การลู่เข้าหาค่าออปติมัมของวิธี Steepest Descent สามารถพิสูจน์ได้ดังนี้คือ ถ้าให้ $\mathbf{s}_p(\mathbf{x}(k))$ เป็น Steepest descent direction ก้าวที่ k และ $p(\mathbf{x}(k))$ เป็น Penalty Function จะได้ว่า

จากสมการที่ (2.17) และ (2.19) สามารถเขียนใหม่ได้เป็น

$$\tau\left\{\frac{\mathbf{x}(k+1)-\mathbf{x}(k)}{T}\right\} = -\nabla p(\mathbf{x}(k)) \qquad , k=0,1,2,\ldots$$

$$\nabla p(\mathbf{x}(k)) = -\tau\left\{\frac{\mathbf{x}(k+1)-\mathbf{x}(k)}{T}\right\} \quad , k=0,1,2,\ldots \qquad (ค.10)$$

โดยที่พารามิเตอร์ τ และ μ มีค่าคงที่เป็นจำนวนจริงบวก , $T = t_{k+1} - t_k$, $k=0,1,2,\ldots$

ในการพิสูจน์การลู่เข้าหาค่าออปติมัม เราสามารถพิสูจน์ได้โดยการพิจารณา $\frac{dp}{dt}$ ถ้า $\frac{dp}{dt}$ น้อยกว่าหรือเท่ากับศูนย์แสดงว่า p มีการเปลี่ยนแปลงในแนวทางที่ลดลงหรือไม่มีการเปลี่ยนแปลงตามลำดับ เหตุการณ์ดังกล่าวแสดงว่า p ลู่เข้าหาค่าๆหนึ่งหรือค่า minimum นั้นเอง ซึ่งเขียนได้ดังนี้คือ

$$\frac{dp}{dt} = \frac{\partial p}{\partial \mathbf{x}}^{T}\frac{d\mathbf{x}}{dt}$$

$$= \nabla p(\mathbf{x})^{T}\dot{\mathbf{x}} \qquad (ค.11)$$

จากสมการที่ (ค.11)) สามารถจัดให้อยู่ในรูปแบบที่ไม่ต่อเนื่องทางเวลาได้ดังนี้คือ

$$\left\{\frac{p(\mathbf{x}(k+1))-p(\mathbf{x}(k))}{T}\right\} = \nabla p(\mathbf{x}(k))^{T}\left\{\frac{\mathbf{x}(k+1)-\mathbf{x}(k)}{T}\right\} \quad , k=0,1,2,\ldots \qquad (ค.12)$$

และเมื่อแทน $\nabla p(\mathbf{x}(k))$ ในสมการที่ (ค.12) ด้วยสมการที่ (ค.10) จะได้ว่า

$$\left\{\frac{p(\mathbf{x}(k+1))-p(\mathbf{x}(k))}{T}\right\} = \left[-\tau\left\{\frac{\mathbf{x}(k+1)-\mathbf{x}(k)}{T}\right\}\right]^T\left\{\frac{\mathbf{x}(k+1)-\mathbf{x}(k)}{T}\right\}$$

$$= -\tau\left\|\frac{\mathbf{x}(k+1)-\mathbf{x}(k)}{T}\right\|^2 \qquad , k=0,1,2,\ldots \quad \text{(ค.13)}$$

ซึ่ง $\because$ พารามิเตอร์ $\tau > 0$ และ $\left\|\frac{\mathbf{x}(k+1)-\mathbf{x}(k)}{T}\right\|^2 \geq 0$, $k=0,1,2,\ldots$ ดังนั้นจากสมการที่ (ค.13) จะได้ว่า

$$\left\{\frac{p(\mathbf{x}(k+1))-p(\mathbf{x}(k))}{T}\right\} \leq 0$$

จากสมการที่ (ค.13) สามารถพิสูจน์ได้ว่าการหาค่าออปติมัมโดยใช้วิธี Steepest Descent นั้นสามารถหาค่า minimum ได้ เนื่องจากพจน์ทางด้านขวาน้อยกว่าหรือเท่ากับศูนย์เสมอ

ประวัติผู้เขียน

นางสาวยุพาพร จารุศิริพจน์ เกิดวันที่ 21 กันยายน พ.ศ 2513 ที่อำเภอเมือง จังหวัดนครปฐม จบการศึกษาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้าระบบควบคุม ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เมื่อปีการศึกษา พ.ศ 2535

พ.ศ 2536 ได้เข้าทำงานในตำแหน่งวิศวกรระบบ ฝ่ายวางแผนระบบ แผนกวิศวกรรม บมจ.โทเทิ่ล แอ็คแซ็ส คอมมูนิเคชั่น

พ.ศ 2537 ได้เข้าศึกษาต่อระดับปริญญามหาบัณฑิต สาขาวิชาระบบควบคุม ภาควิชาวิศวกรรมไฟฟ้า จุฬาลงกรณ์มหาวิทยาลัย โดยได้รับอนุญาตให้พักงานมาเรียนจาก บมจ.โทเทิ่ล แอ็คแซ็ส คอมมูนิเคชั่น

ปัจจุบันทำงานในตำแหน่งวิศวกรระบบอาวุโส ฝ่ายวางแผนระบบ แผนกวิศวกรรม บมจ.โทเทิ่ล แอ็คแซ็ส คอมมูนิเคชั่น