บทที่ 3

การออกแบบระบบควบคุมแบบปรับตัวเองโดยการใช้เครือข่ายนิวรอน

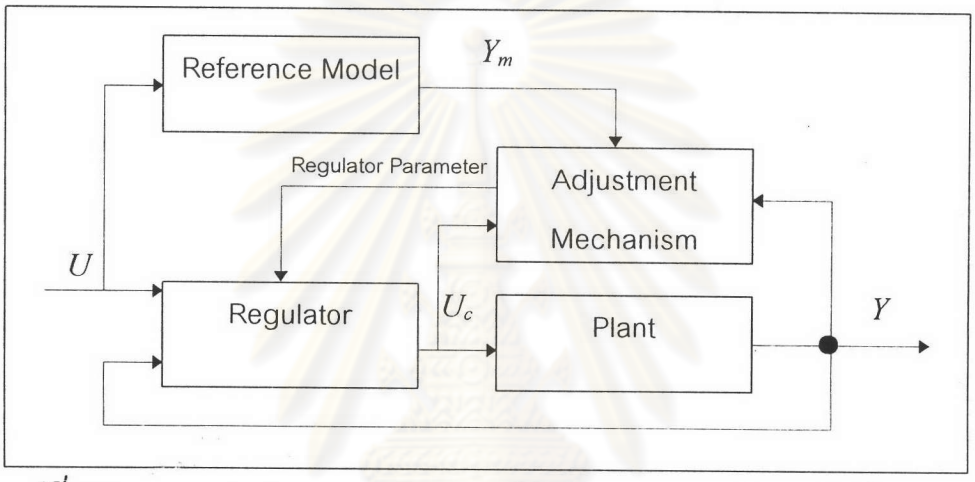
การควบคุมแบบปรับตัวเองเป็นที่นิยมใช้อย่างแพร่หลายในวงการอุตสาหกรรม โดยการออกแบบตัวควบคุมดังกล่าวจำแนกได้เป็น 2 วิธี ได้แก่ Indirect Adaptive Control และ Direct Adaptive Control หลักการของวิธีแรกคือ การปรับค่าพารามิเตอร์ของตัวควบคุมไม่ได้ถูกปรับโดยตรงแต่ได้จากการหาเอกลักษณ์ (Identification) เพื่อหาค่าพารามิเตอร์ที่แท้จริงของระบบที่ต้องการควบคุม และนำค่าพารามิเตอร์ที่ได้ไปใช้ในการปรับค่าพารามิเตอร์ของตัวควบคุมอีกครั้ง ซึ่งวิธีปรับค่าพารามิเตอร์ของตัวควบคุมมีหลายวิธี อาทิเช่น Indirect Self-Tuning Control (Explicit Self-Tuning Control), Minimum Variance, Linear Quadratic Guassian Control ฯลฯ วิธีที่สองคือ Direct Adaptive Control ค่าพารามิเตอร์ของตัวควบคุมจะถูกปรับโดยตรงจากค่าความคลาดเคลื่อนระหว่างสัญญาณออกของรูปแบบจำลองอ้างอิงและสัญญาณออกจากระบบ โดยมีเป้าหมายเพื่อให้สมรรถนะของระบบเป็นไปตามแบบจำลองอ้างอิง ตัวอย่างเช่น วิธี Direct Model Reference Adaptive Control, Direct Self-Tuning Control (Implicit Self-Tuning Control) ฯลฯ

ในบทนี้กล่าวถึงระบบควบคุมแบบปรับตัวเองตามแบบจำลองอ้างอิง (Model-Reference Adaptive Control) ที่นำมาประยุกต์ใช้กับระบบที่ใช้เครือข่ายนิวรอนเป็นตัวควบคุม รวมไปถึงแนวคิดและวิธีในการออกแบบตัวควบคุมดังกล่าว

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

3.1 การควบคุมแบบปรับตัวเองตามแบบจำลองอ้างอิง (Model-Reference Adaptive Control : MRAC) [3]

การควบคุมแบบปรับตัวเองตามแบบจำลองอ้างอิงเป็นระบบควบคุมแบบปรับตัวเองที่สมรรถนะ (Performance) ของระบบมีผลตามแบบจำลองอ้างอิง (Model Reference) ที่กำหนด แสดงได้ดังรูปที่ 3.1

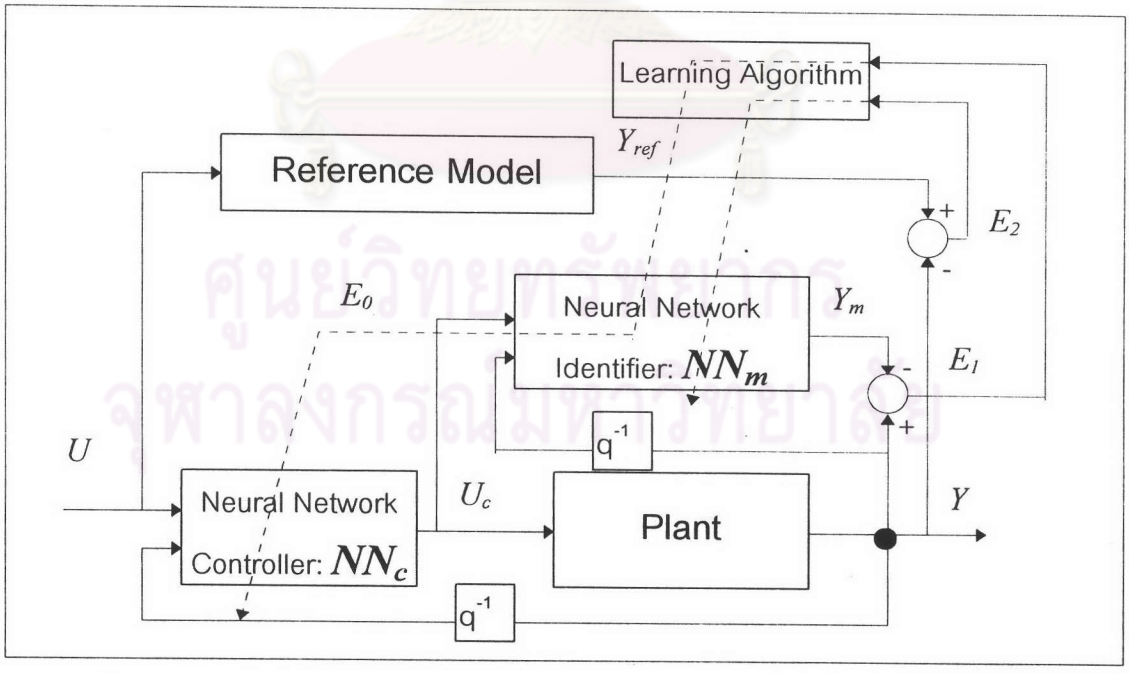


รูปที่ 3.1 *แผนภาพแสดงการควบคุมแบบปรับตัวเองตามแบบจำลองอ้างอิง*

จากรูปที่ 3.1 ระบบควบคุมประกอบไปด้วย 3 ส่วนคือ ส่วนวงภายใน (Inner Loop) ได้แก่ Regulator ซึ่งทำหน้าที่ในการสร้างสัญญาณควบคุมป้อนให้กับระบบเพื่อให้สมรรถนะรวมของระบบมีผลตามแบบจำลองอ้างอิงที่กำหนด และส่วนวงภายนอก (Outer Loop) ได้แก่ Adjustment Mechanism โดยทำหน้าที่ปรับค่าพารามิเตอร์ของ Regulator ขึ้นอยู่กับความคลาดเคลื่อนระหว่างสัญญาณออกจากแบบจำลองอ้างอิง (Model Reference Output Signal) และสัญญาณออกจากระบบ (Plant Output Signal) วิธีในวิเคราะห์และออกแบบการปรับค่าพารามิเตอร์ของตัว Regulator มีหลายวิธี อาทิเช่น วิธี Pole-Placement, วิธี Gradient, วิธี Lyapunov Functions, วิธี Passivity Theory ฯลฯ และส่วนสุดท้ายคือส่วนของแบบจำลองอ้างอิง (Reference Model) ซึ่งเป็นสมการพลวัตที่ใช้เป็นตัวกำหนดลักษณะผลตอบและสมรรถนะของระบบให้เป็นไปตามความต้องการ

3.2 การควบคุมแบบปรับตัวเองตามแบบจำลองอ้างอิงโดยใช้เครือข่ายนิวรอน

เนื่องจากความสามารถของเครือข่ายนิวรอนในการเรียนรู้ จดจำลักษณะความสัมพันธ์ของข้อมูล และสามารถให้ผลตอบที่สอดคล้องกับชุดของข้อมูลที่ป้อนเข้าหลังจากที่ได้รับการฝึกฝนอย่างเพียงพอ จึงสามารถนำไปประยุกต์ใช้งานกับการควบคุมแบบปรับตัวเองตามแบบจำลองอ้างอิง แต่เนื่องจากไม่ทราบค่าสัญญาณควบคุมที่เหมาะสมในการป้อนเข้าสู่ระบบเพื่อให้สมรรถนะของระบบเป็นไปตามแบบจำลองอ้างอิง ดังนั้นการฝึกเครือข่ายนิวรอนเพื่อเป็นตัวควบคุมจึงอาศัยวิธีการส่งผ่านค่าความคลาดเคลื่อนระหว่างสัญญาณที่ออกจากแบบจำลองอ้างอิงและสัญญาณที่ออกจากระบบ ผ่านกลับไปตามชั้นต่างๆของเครือข่ายนิวรอนที่ใช้ในการหาเอกลักษณ์ (Neural Network Identification) ค่าความคลาดเคลื่อนที่แผ่กลับมาถูกใช้ในการฝึกเครือข่ายนิวรอนเพื่อเป็นตัวควบคุม (Neural Network Controller) ให้สามารถเรียนรู้และลดค่าความคลาดเคลื่อนดังกล่าว วิธีนี้เรียกว่า "Differentiating the Model" หรือ "Forward and Inverse Modeling" [26] แสดงดังรูปที่ 3.2



รูปที่ 3.2 *แผนภาพแสดงการควบคุมแบบปรับตัวเองตามแบบจำลองอ้างอิงโดยใช้เครือข่ายนิวรอน*

การประยุกต์เครือข่ายนิวรอนในการควบคุมแบบปรับตัวเองตามแบบจำลองอ้างอิง ดังรูปที่ 3.2 ประกอบด้วยเครือข่ายนิวรอน 2 ส่วน ได้แก่ ส่วนแรกคือส่วนของการหาเอกลักษณ์ (Neural Network Identification Part) หรือเรียกอีกอย่างหนึ่งว่า ส่วนแบบจำลองระบบ (Neural Network Modeling Part : NN_m) หรือส่วนของ Adjustment Mechanism ในรูปที่ 3.1 ส่วนนี้เครือข่ายนิวรอนจะทำหน้าที่ในการจำลองหรือเลียนแบบพฤติกรรมของระบบ โดยใช้ค่าความแตกต่างระหว่างสัญญาณเอาต์พุตจากพลานท์ (Plant Output Signal : Y) และสัญญาณเอาต์พุตจากเครือข่ายนิวรอนที่ใช้ในการหาเอกลักษณ์ (NN Identification Output Signal : Y_m) : E_1 ในการฝึกเครือข่ายดังกล่าว และเป็นตัวสร้างสัญญาณที่ใช้ในการฝึกเครือข่ายนิวรอนเพื่อเป็นตัวควบคุมโดยอาศัยการส่งผ่านค่าความคลาดเคลื่อนระหว่างสัญญาณเอาต์พุตจากแบบจำลองอ้างอิง (Reference Model Output Signal : Y_{ref}) กับสัญญาณเอาต์พุตจากพลานท์ (Plant Output Signal : Y) : E_2 แผ่ย้อนกลับผ่านชั้นต่างๆ จนถึงชั้นอินพุตของเครือข่ายนิวรอนที่ใช้ในการหาเอกลักษณ์ ค่าความคลาดเคลื่อนที่ปมนิวรอนในชั้นอินพุต (E_0) นำไปใช้ในการฝึกเครือข่ายนิวรอนเพื่อเป็นตัวควบคุม (Neural Network Controller : NN_c) ในส่วนที่สองหรือส่วน Regulator ในรูปที่ 3.1 โดยค่าพารามิเตอร์ของเครือข่ายจะถูกปรับเพื่อลดค่าความคลาดเคลื่อนดังกล่าว และสร้างสัญญาณควบคุม (Control Signal : U_c) ป้อนให้กับพลานท์ เพื่อให้สมรรถนะของพลานท์เป็นไปตามแบบจำลองอ้างอิง (Model Reference) ตามที่กำหนด เครือข่ายนิวรอนที่ใช้ทั้งสองส่วนเป็นชนิด Backpropagation เนื่องจากใช้การเรียนรู้แบบอาศัยการแนะนำ (Supervised Learning) ทำให้สามารถฝึกเครือข่ายเพื่อสร้างความสัมพันธ์ขึ้นได้จากข้อมูลที่ให้เรียนรู้ และเนื่องจากเป็นเครือข่ายนิวรอนที่มีจำนวนหลายชั้นทำให้สามารถเพิ่มจำนวนชั้นเพื่อให้สามารถจำรูปแบบที่ซับซ้อนและประมาณค่าฟังก์ชันที่ซับซ้อนได้ดี

3.3 การออกแบบเครือข่ายนิวรอนในส่วนการหาเอกลักษณ์ (Neural Network Identification) หรือส่วนการสร้างแบบจำลองระบบ (NN Modeling : $\mathbf{NN_m}$)

3.3.1 ทฤษฎีการหาเอกลักษณ์เบื้องต้น

จากทฤษฎีการหาเอกลักษณ์ (Identification) [17] โดยทั่วไปจำเป็นต้องกำหนดรูปแบบจำลองของระบบก่อน รูปแบบจำลองของระบบเชิงเส้นไม่เปลี่ยนแปลงตามเวลาแบบอินพุต-เอาต์พุตเดียว (Single Input-Single Output Linear Time-Invariant System) ที่นิยมใช้โดยทั่วไปแสดงดังตารางที่ 3.1

$A(q)y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{Dq)}e(t)$	
Polynomial	Name of Model Structure
B	FIR (Finite Impulse Response)
AB	ARX
ABC	ARMAX
AC	ARMA
ABD	ARARX
$ABCD$	ARARMAX
BF	OE (Output Error)
$BFCD$	BJ (Box-Jenkins)

NOTE: AR = "Autoregressive " , MA = "Moving Average" , X = "Extra Input"

ตารางที่ 3.1 แสดงแบบจำลองสมการที่ใช้ในการหาเอกลักษณ์ของระบบเชิงเส้นไม่เปลี่ยนแปลงตามเวลาแบบอินพุต-เอาต์พุตเดียว

การเลือกใช้รูปแบบจำลองใดขึ้นอยู่กับลักษณะโดยทั่วไปของระบบที่นำไปใช้ ในที่นี้เลือกใช้รูปแบบจำลอง ARX (Autoregressive Extra Input) เนื่องจากระบบที่ใช้ในการทดสอบเป็นแบบอินพุต-เอาต์พุตเดียว และข้อมูลในการสร้างความสัมพันธ์สามารถวัดได้สองค่า คือ สัญญาณอินพุตของระบบ (Input Signal : u) และสัญญาณเอาต์พุตของระบบ (Output Signal : y) สามารถแสดงได้ดังสมการที่ 3.1

$$A(q)y(t) = B(q)u(t) \quad \text{.....(3.1)}$$

เมื่อ y สัญญาณเอาต์พุต (Output Signal)

u สัญญาณอินพุต (Input Signal)

$A(q) = 1 + a_1 q^{-1} + a_2 q^{-2} + \ldots + a_m q^{-m}$

$B(q) = 1 + b_1 q^{-1} + b_2 q^{-2} + \ldots + b_n q^{-n}$

a_m, b_n ค่าพารามิเตอร์ของพลานท์

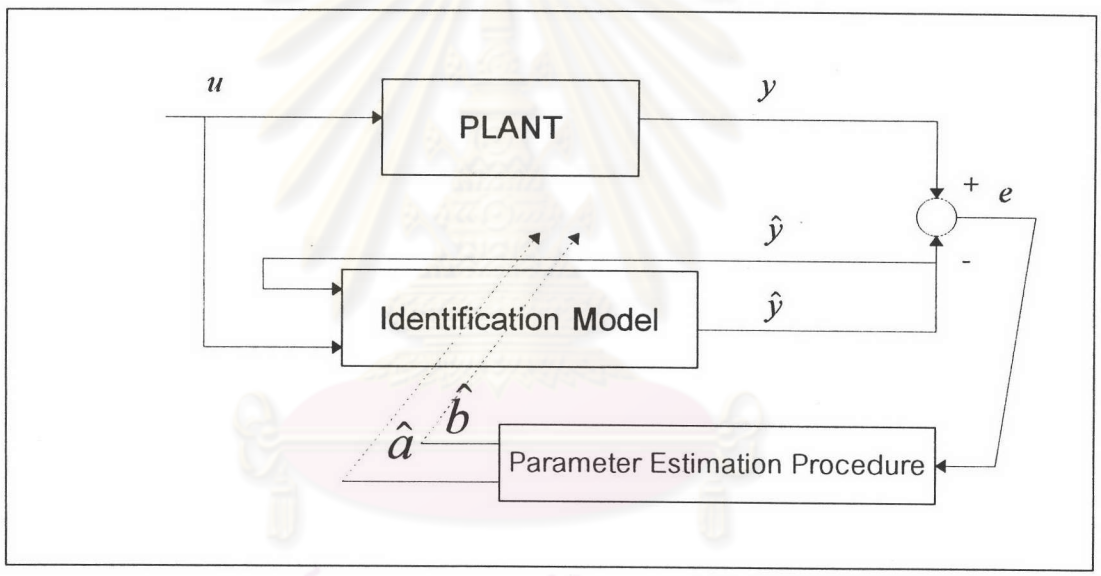
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

I17380908

วิธีการหาค่าพารามิเตอร์ของระบบมีหลายวิธี [16] เช่น วิธี Gradient, วิธี Recursive least Square (RLS), วิธี Extend Least Square (ELS), วิธี Recursive Maximum Likelihood (RML) ฯลฯ การหาค่าพารามิเตอร์ของพลานท์อาศัยค่าความคลาดเคลื่อนระหว่างค่าสัญญาณเอาต์พุตระบบ (y) กับค่าสัญญาณเอาต์พุตที่ประมาณได้จากการหาเอกลักษณ์ ($\hat{y}$) : e

วีธีการหาเอกลักษณ์จากรูปแบบจำลองดังกล่าวแบ่งออกเป็น 2 วิธีคือ [20]

1. แบบจำลองการหาเอกลักษณ์แบบขนาน (Parallel Identification Model)



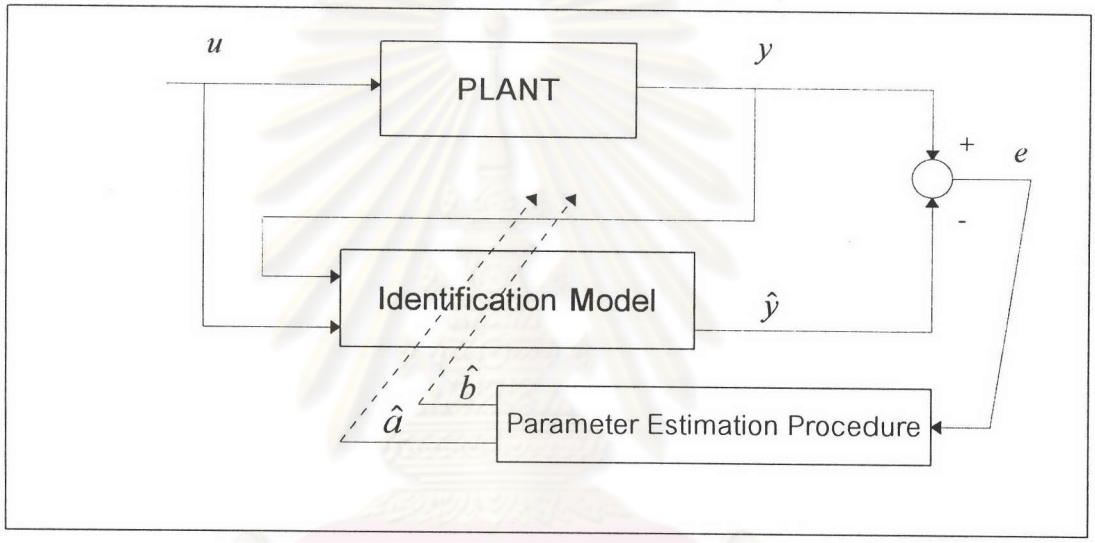
รูปที่ 3.3 แบบจำลองการหาเอกลักษณ์แบบขนาน (Parallel Identification Model)

จากรูปที่ 3.3 วิธีการหาเอกลักษณ์ (Identification) โดยอาศัยชุดข้อมูลของสัญญาณเอาต์พุตที่ประมาณได้ ($\hat{y}$) จากวิธีแสดงเอกลักษณ์ในการคำนวนแบบสืบทอดเพื่อดำเนินการประมาณค่าพารามิเตอร์ของแบบจำลองของระบบ $(\hat{a},\hat{b})$ จากสมการ 3.1 สมการแบบจำลองการหาเอกลักษณ์แบบขนาน แสดงดังสมการที่ 3.2

$$\hat{y}(t+1) = \sum_{i=0}^{m-1} \hat{a}_{i+1}(t)\hat{y}(t-i) + \sum_{j=0}^{n-1} \hat{b}_{j+1}(t)u(t-j) \quad \text{.....(3.2)}$$

เนื่องจากข้อมูลที่ใช้ในการหาค่าพารามิเตอร์ของแบบจำลองเป็นข้อมูลที่ได้จากการประมาณค่าสัญญาณออกจากแบบจำลองที่ลำดับเวลาก่อนหน้า ดังนั้นถ้าการประมาณค่าดังกล่าวมีความคลาดเคลื่อนจะส่งผลให้ค่าพารามิเตอร์ที่ประมาณได้ในลำดับเวลาถัดมามีโอกาสผิดพลาดตามไปด้วย วิธีการนี้จึงไม่เป็นที่นิยมใช้เท่าที่ควร

2. แบบจำลองการหาเอกลักษณ์แบบอนุกรม-ขนาน (Series-Parallel Model)



รูปที่ 3.4 แบบจำลองการหาเอกลักษณ์แบบอนุกรม-ขนาน (Series-Parallel Identification Model)

จากรูปที่ 3.4 เป็นวิธีการหาเอกลักษณ์ที่อาศัยข้อมูลสัญญาณเอาท์พุตจากพลานท์จริง (y) นำมาใช้ในการปรับค่าพารามิเตอร์ของแบบจำลองระบบ ดังนั้นจึงทำให้การหาค่าพารามิเตอร์ที่แท้จริงของระบบเป็นไปอย่างมีประสิทธิภาพ และใกล้เคียงกับค่าพารามิเตอร์จริงของระบบมากขึ้น จากสมการ 3.1 รูปแบบจำลองการหาเอกลักษณ์แบบอนุกรม-ขนาน แสดงดังสมการที่ 3.3

$$\hat{y}(t+1) = \sum_{i=0}^{m-1} \hat{a}_{i+1}(t)y(t-i) + \sum_{j=0}^{n-1} \hat{b}_{j+1}(t)u(t-j) \quad \text{.....(3.3)}$$

3.3.2 การประยุกต์เครือข่ายนิวรอนในการหาเอกลักษณ์

การหาเอกลักษณ์ในงานวิทยานิพนธ์นี้เลือกใช้รูปแบบจำลอง ARX (Autoregressive Extra Input : ARX) ดังเหตุผลที่กล่าวไว้แล้วในหัวข้อ 3.3.1 ดังนั้นรูปแบบสมการการหาเอกลักษณ์โดยเครือข่ายนิวรอนแสดงดังสมการที่ 3.4

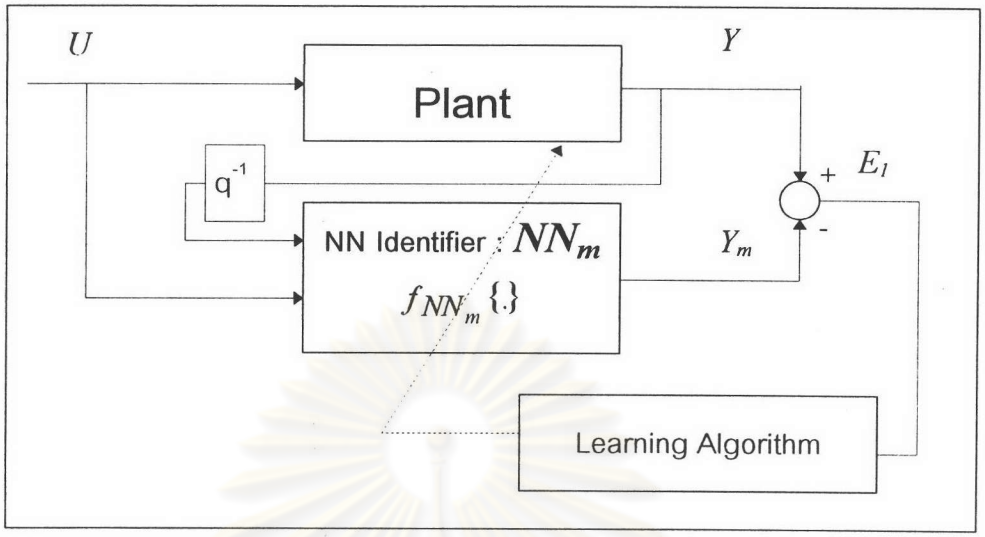
$$Y_m(k+1) = f_{NN_m}\left(Y(k), U(k)\right) \quad \text{.....(3.4)}$$

โดยที่

$f_{NN_m}\{.\}$	เครือข่ายนิวรอนที่ใช้ในการหาเอกลักษณ์ (NN-Identification หรือ NN-Modeling : NN_m)
$\left(Y(k), U(k)\right)$	ชุดข้อมูลอินพุตซึ่งป้อนเข้าสู่เครือข่ายนิวรอน
$Y_m(k+1)$	สัญญาณเอาต์พุตจากเครือข่ายนิวรอนที่ใช้ในการหาเอกลักษณ์
$Y(k)$	ชุดข้อมูลเอาต์พุตจากพลานท์ (Plant Output Signal)
$U(k)$	ชุดข้อมูลอินพุตของพลานท์ (Plant Input Signal)

รูปแบบจำลองดังกล่าวมีข้อดีคือ ไม่จำเป็นต้องทราบลักษณะสมการพลวัตที่แท้จริงของระบบหรือพลานท์ ดังนั้นวิธีการดังกล่าวจะไม่ทราบค่าพารามิเตอร์ที่แท้จริงของระบบแต่ค่าพารามิเตอร์ดังกล่าวจะแฝงอยู่ในรูปของค่าน้ำหนักและค่าไบแอสของเครือข่าย ซึ่งนำไปใช้ในเครือข่ายนิวรอนที่ใช้เป็นตัวควบคุมที่จะกล่าวถึงในลำดับถัดไป

จากสมการที่ (3.4) แสดงรูปการใช้เครือข่ายนิวรอนในการหาเอกลักษณ์แบบอนุกรม-ขนาน ได้ดังรูปที่ 3.5



<u>รูปที่ 3.5</u> แสดงการประยุกต์เครือข่ายนิวรอนในการหาเอกลักษณ์ (Identification)

จากรูปที่ 3.5 การฝึกเครือข่ายนิวรอนในการหาเอกลักษณ์ประกอบไปด้วยชุดข้อมูลอินพุต $(Y(k),U(k))$ และชุดข้อมูลเป้าหมาย ($Y(k+1)$) ค่าความคลาดเคลื่อนระหว่างสัญญาณออกจากเครือข่ายนิวรอนและชุดข้อมูลเป้าหมาย (E_1) ถูกใช้ในการปรับค่าน้ำหนักและค่าไบแอสของเครือข่ายนิวรอน กำหนดดรรชนีสมรรถนะของการหาเอกลักษณ์ด้วยเครือข่ายนิวรอน (J_{NN_m}) ได้ดังสมการที่ 3.5-3.6

$$J_{NN_m} = \frac{1}{2}\sum_{k=0}^{N}\left(Y(k)-Y_m(k)\right)^2 \quad \text{.....(3.5)}$$

หรือ

$$J_{NN_m} = \frac{1}{2}\sum_{k=0}^{N}E_1^2(k) \quad \text{.....(3.6)}$$

โดยที่

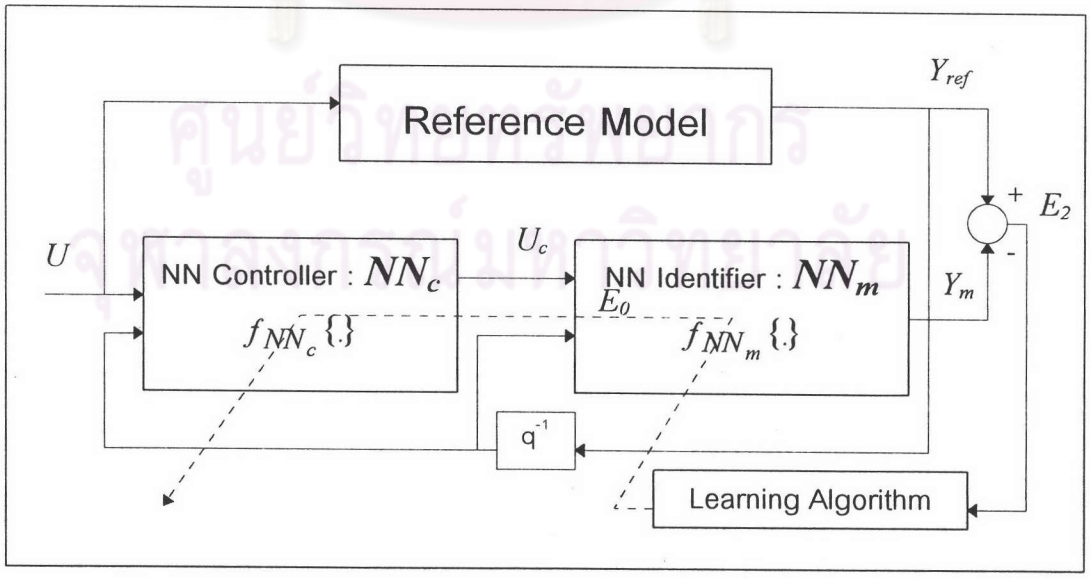
N จำนวนชุดข้อมูลที่ใช้ในการฝึกเครือข่าย (Training Data)

J_{NN_m} ดรรชนีสมรรถนะของการฝึกเครือข่ายนิวรอนเพื่อใช้ในการหาเอกลักษณ์

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

3.4 การออกแบบเครือข่ายนิวรอนเพื่อเป็นตัวควบคุม (NN Controller : $\mathbf{NN_c}$)

โดยทั่วไปการฝึกเครือข่ายนิวรอนเพื่อเป็นตัวควบคุมจำเป็นทราบค่าสัญญาณเป้าหมาย (Target Signal) โดยสัญญาณเป้าหมายในการฝึกคือ สัญญาณควบคุมที่แท้จริงที่ป้อนเข้าสู่ระบบเพื่อให้สมรรถนะหรือผลตอบสนองของระบบเป็นไปตามแบบจำลองอ้างอิง แต่ในทางปฏิบัติไม่ทราบค่าสัญญาณควบคุมที่เหมาะสมจึงเป็นการยากในการฝึกเครือข่ายนิวรอนเพื่อเป็นตัวควบคุมโดยตรง ดังนั้นการฝึกเครือข่ายนิวรอนดังกล่าวจึงใช้การฝึกทางอ้อมอาศัยการกำหนดดรรชนีสมรรถนะของตัวควบคุมเป็นค่าความแตกต่างระหว่างสัญญาณออกจากแบบจำลองอ้างอิงและสัญญาณออกจากเครือข่ายนิวรอนในการหาเอกลักษณ์ วิธีการควบคุมดังกล่าวเรียกว่า "Differentiating The Model หรือ Forward and Inverse Modeling" [26] ค่าความคลาดเคลื่อนดังกล่าวจะถูกส่งไปยังปมนิวรอนในชั้นเอาต์พุตของเครือข่ายนิวรอนที่ฝึกเป็นตัวควบคุม โดยอาศัยการส่งผ่านทางชั้นต่างๆของเครือข่ายนิวรอนที่ใช้ในการหาเอกลักษณ์ (Identification) จนถึงปมนิวรอนในชั้นอินพุตของเครือข่าย ค่าความคลาดเคลื่อนจะถูกนำมาใช้ในการฝึกเครือข่ายนิวรอนเพื่อเป็นตัวควบคุมโดยมีเป้าหมายเพื่อลดค่าความคลาดเคลื่อนดังกล่าว สามารถอธิบายได้ดังรูปที่ 3.6



รูปที่ 3.6 การฝึกเครือข่ายนิวรอนเพื่อเป็นตัวควบคุม (Neural Network Controller : NN_c)

จากรูปที่ 3.6 การฝึกเครือข่ายนิวรอนเพื่อใช้เป็นตัวควบคุมเป็นการฝึกแบบ Off-Line ดังนั้นจึงไม่มีการรวมพลานท์จริงเข้ามาในระหว่างการฝึก แต่อาศัยเครือข่ายนิวรอนที่ได้รับการฝึกในการหาเอกลักษณ์ (Neural Network Identification : NN_m) แทนพลาท์จริง โดยกำหนดค่าน้ำหนัก และค่าไบแอสของเครือข่ายดังกล่าวให้เป็นค่าคงที่โดยมีค่าเท่ากับค่าที่ได้หลังจากการฝึกเครือข่ายนิวรอน

เมื่อป้อนข้อมูลอินพุต (U) ผ่านเครือข่ายนิวรอนที่ต้องการฝึกเป็นตัวควบคุม (Neural Network Controller : NN_c) สามารถแสดงความสัมพันธ์ได้ดังสมการที่ 3.7

$$U_c(k+1) = f_{NN_c}\left(U(k), Y_{ref}(k)\right) \quad \text{.....(3.7)}$$

โดยที่

$U(k)$	สัญญาณอินพุต (Input Signal)
$U_c(k+1)$	สัญญาณควบคุม (Control Signal)
$f_{NN_c}\{.\}$	เครือข่ายนิวรอนที่ใช้เป็นตัวควบคุม
$\left(U(k), Y_{ref}(k)\right)$	ข้อมูลอินพุตซึ่งป้อนเข้าสู่เครือข่ายนิวรอนที่ใช้เป็นตัวควบคุม
$Y_{ref}(k)$	สัญญาณเอาท์พุตจากแบบจำลองอ้างอิง

ชุดข้อมูลอินพุต $\left(U(k), Y_{ref}(k)\right)$ ซึ่งป้อนเข้าสู่เครือข่ายนิวรอนซึ่งใช้เป็นตัวควบคุมจะถูกส่งผ่านไปยังชั้นต่างๆ จนถึงชั้นเอาต์พุตของเครือข่าย สัญญาณออกจากปมนิวรอนในชั้นดังกล่าวจะเป็นสัญญาณควบคุม (U_c) ซึ่งถูกป้อนเข้าสู่เครือข่ายนิวรอนหลังจากได้รับการฝึกในการหาเอกลักษณ์ (Neural Network Identification : NN_m) สามารถแสดงความสัมพันธ์ได้ดังสมการที่ 3.8

$$Y_m(k+1) = f_{NN_m}\left(Y_{ref}(k), U_c(k)\right) \quad \text{.....(3.8)}$$

ค่าความคลาดเคลื่อนระหว่างสัญญาณออกจากเครือข่ายนิวรอนที่ได้รับการฝึกในการหาเอกลักษณ์ (Y_m) กับ สัญญาณที่ออกจากแบบจำลองอ้างอิง (Y_{ref}) : E_2 นำไปใช้ในการ

กำหนดค่าดรรชนีสมรรถนะของการฝึกเครือข่ายนิวรอนเพื่อเป็นตัวควบคุม ดังแสดงในสมการที่ 3.9-3.10

$$J_{NN_c} = \frac{1}{2}\sum_{k=0}^{N}\left(Y_{ref}(k) - Y_m(k)\right)^2 \quad \text{.....(3.9)}$$

หรือ

$$J_{NN_c} = \frac{1}{2}\sum_{k=0}^{N} E_2^2(k) \quad \text{.....(3.10)}$$

โดยที่

J_{NN_c} ดรรชนีสมรรถนะของการฝึกเครือข่ายนิวรอนเพื่อเป็นตัวควบคุม

ค่าความคลาดเคลื่อน (E_2) ถูกถ่ายทอดผ่านทุกชั้นของเครือข่ายนิวรอนที่ใช้ในการหาเอกลักษณ์ (Neural Network Identification : NN_m) พิจารณาทำนองเดียวกับขั้นตอนการเรียนรู้ในบทที่ 2 โดยกำหนดค่าน้ำหนัก และค่าไบแอสหลังจากการฝึกเครือข่ายดังกล่าวไว้เป็นค่าคงที่ และไม่มีการเปลี่ยนแปลงตลอดการฝึกเครือข่ายนิวรอนเพื่อเป็นตัวควบคุม ดังแสดงในสมการต่อไปนี้

สำหรับชั้นเอาต์พุต (Output Layer)

$$\delta_o = \left[Y_{ref} - Y_m\right] f'(Y_in) \quad \text{.....(3.11)}$$

สำหรับชั้นภายใน (Hidden Layer)

$$\delta_h = \left(W\delta_o\right).f'(Z_in) \quad \text{.....(3.12)}$$

สำหรับชั้นอินพุต (Input Layer)

$$E_0 = \left(V\delta_h\right) \quad \text{.....(3.13)}$$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สัญญาณความคลาดเคลื่อนของปมนิวรอนในชั้นอินพุต (E_0) ซึ่งคำนวนได้จากสมการที่ 3.13 ใช้ในการปรับค่าน้ำหนักและค่าไบแอสของเครือข่ายนิวรอนซึ่งใช้เป็นตัวควบคุม (Neural Network Controller: NN_c) โดยใช้วิธีการเรียนรู้ดังกล่าวในบทที่ 2 การปรับค่ามีวัตถุประสงค์เพื่อลดค่าสัญญาณคลาดเคลื่อน : E_0 [26]

3.5 การใช้งานเครือข่ายนิวรอนหลังจากได้รับการฝึกในการหาเอกลักษณ์และการควบคุม

การใช้งานเครือข่ายนิวรอนในระบบควบคุมจริงดังแสดงในรูป 3.2 อาศัยค่าน้ำหนักและค่าไบแอสของเครือข่ายนิวรอนหลังจากฝึกทั้งในส่วนการหาเอกลักษณ์ (NN_m) และในส่วนของตัวควบคุม (NN_c) เป็นจุดเริ่มต้นของการทำงานในเวลาจริง (Real Time) ซึ่งค่าพารามิเตอร์ต่างๆของเครือข่ายนิวรอนทั้งสองจะถูกปรับให้สอดคล้องกับการเปลี่ยนแปลงที่เกิดขึ้นกับระบบเพื่อให้สมรรถนะโดยรวมของระบบเป็นไปตามแบบจำลองอ้างอิง

การทำงานของระบบควบคุมที่ใช้เครือข่ายนิวรอล เริ่มจากการกำหนดดรรชนีสมรรถนะของเครือข่ายนิวรอลที่ใช้เป็นตัวควบคุม ดังแสดงในสมการที่ 3.14

$$J_{NN_c} = \frac{1}{2}\left(Y_{ref}(k) - Y(k)\right)^2 \quad \text{.....(3.14)}$$

เครือข่ายนิวรอลที่ใช้เป็นตัวควบคุม (NN Controller : NN_c) จะรับสัญญาณเข้า (Input Signal : $U(k)$) และสร้างสัญญาณควบคุม (Control Signal: U_c) ดังแสดงในสมการที่ 3.15

$$U_c(k+1) = f_{NN_c}\left(U(k), Y(k)\right) \quad \text{.....(3.15)}$$

สัญญาณควบคุมจะป้อนเข้าสู่พลานท์ (Plant) หรือระบบที่ต้องการควบคุม และเครือข่ายนิวรอนที่ใช้ในการหาเอกลักษณ์ (NN Identification: NN_m) สัญญาณออกที่ได้คำนวนดังสมการที่ 3.16

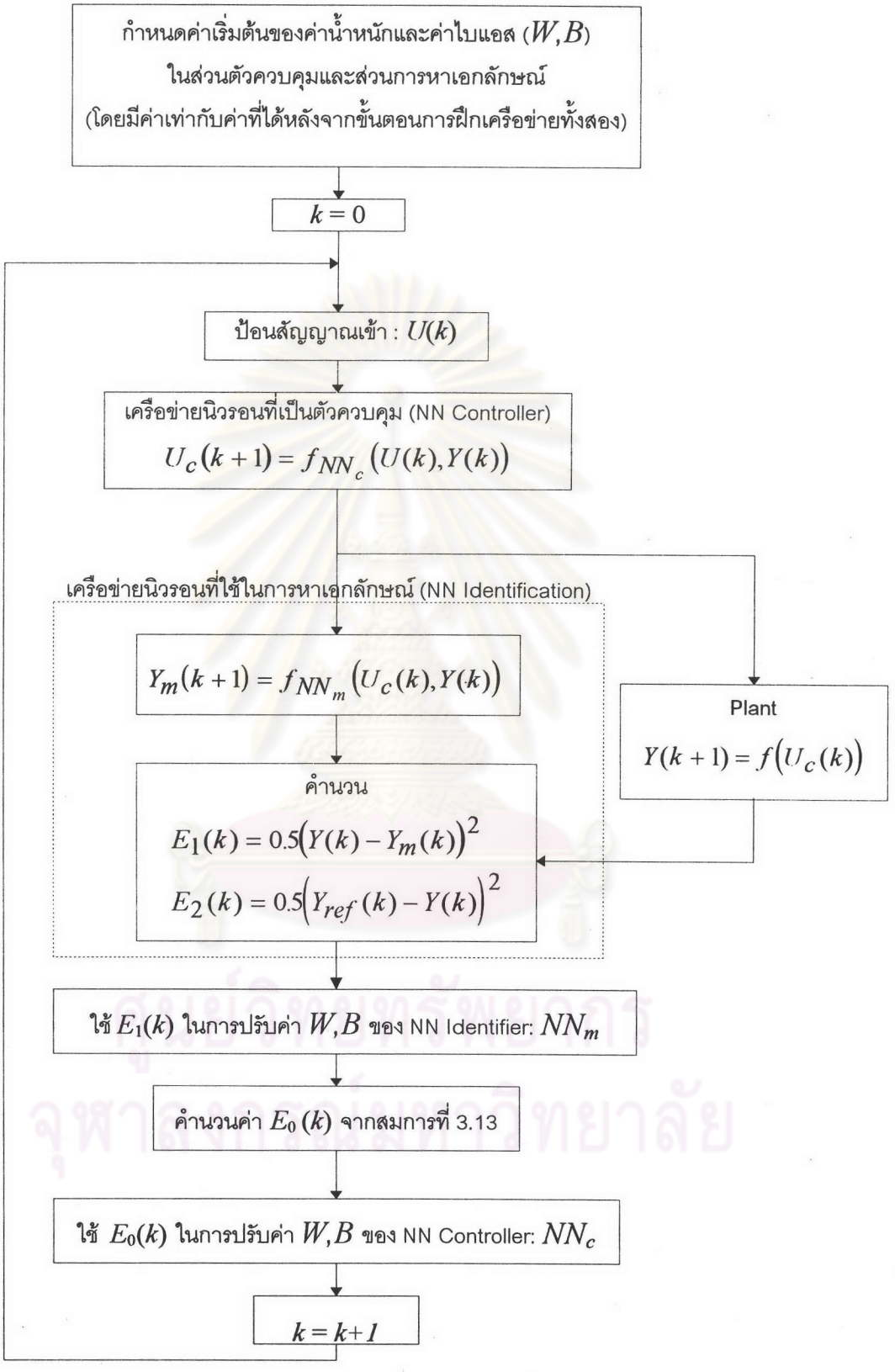
$$Y_m(k+1) = f_{NN_m}\left(Y(k), U_c(k)\right) \quad \text{.....(3.16)}$$

โดยดรรชนีสมรรถนะของเครือข่ายนิวรอนที่ใช้หาเอกลักษณ์ แสดงดังสมการที่ 3.17

$$J_{NN_m} = \frac{1}{2}\left(Y(k) - Y_m(k)\right)^2 \quad \text{.....(3.17)}$$

การปรับค่าพารามิเตอร์ของเครือข่ายนิวรอนทั้งสอง พิจารณาในทำนองเดียวกับขั้นตอนการเรียนรู้ สามารถแสดงแผนภูมิการทำงานดังรูปที่ 3.7

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



<u>รูปที่3.7</u> แสดงแผนภาพของระบบควบคุมโดยใช้เครือข่ายนิวรอน

3.6 การกำหนดแบบจำลองอ้างอิง (Reference Model) [22]

แบบจำลองอ้างอิงที่เลือกใช้ในระบบควบคุมเป็นระบบ Second Order เนื่องจากสามารถกำหนดลักษณะของผลตอบสนอง (Response) อันได้แก่ ช่วงเวลาขึ้น (Rise Time), ค่าพุ่งเกิน (Overshoot), ช่วงเวลาเข้าที่ (Settling Time) ฯลฯ โดยผ่านสมการทางคณิตศาสตร์ได้ง่าย สามารถแสดงในรูปฟังก์ชันถ่ายโอน (Transfer Function) ดังสมการที่ 3.18

$$G_{ref}(s) = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0{}^2} \quad \text{.....(3.18)}$$

กำหนดให้

$G_{ref}(s)$ ฟังก์ชันถ่ายโอนของแบบจำลองอ้างอิง (Transfer Function of Reference Model)

ω_0 การหน่วงความถี่ธรรมชาติต่ำกว่าวิกฤต (Underdamp Natural Frequency)

ζ อัตราส่วนการหน่วง (Damping Ratio)

ลักษณะผลตอบจากการกำหนดแบบจำลองอ้างอิงสามารถคำนวนได้ดังนี้

1. ช่วงเวลาขึ้น (Rise Time : t_r)

คำนวนจาก

$$t_r = \frac{\pi - \beta}{\sigma} \quad \text{.....(3.19)}$$

โดยที่

$$\sigma = \zeta\omega_0 \quad \text{.....(3.20)}$$

$$\beta = \tan^{-1}\left(\frac{\omega_d}{\sigma}\right) \quad \text{.....(3.21)}$$

$$\omega_d = \omega_0\sqrt{1-\varsigma^2} \quad \text{.....(3.22)}$$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

2. เปอร์เซนต์ค่าพุ่งเกินสูงสุด (Maximum Percent Over Shoot : M_p)

คำนวนจาก

$$M_p = e^{-\left(\frac{\sigma}{\omega_0}\right)\pi} \times 100\% \quad \text{.....(3.23)}$$

3. เวลาค่ายอด (Peak Time : t_p)

คำนวนจาก

$$t_p = \frac{\pi}{\omega_d} \quad \text{.....(3.24)}$$

4. ช่วงเวลาเข้าที่ (Settling Time : t_s)

คำนวนจาก

$$t_s = \frac{4}{\varsigma\omega_0} \quad \text{.....(3.25)}$$

ในระบบควบคุมเชิงเลข (Digital Control) โดยอาศัยคอมพิวเตอร์ในการควบคุมในเวลาจริงสามารถแปลงฟังก์ชันถ่ายโอนดังกล่าวให้อยู่ในโดเมนความไม่ต่อเนื่องเชิงเวลากับการคงค่าอันดับศูนย์ (Discrete Time Domain with Zero Order Hold) ดังสมการที่ 3.26

$$H_{ref}(q^{-1}) = \frac{b_1 q^{-1} + b_2 q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2}} \quad \text{.....(3.26)}$$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

โดยที่

$$\omega_d = \omega_0\sqrt{1-\zeta^2} \quad ; (|\zeta| < 1)$$

$$\alpha = e^{-\zeta\omega_0 T_s} \quad ; \beta = \cos(\omega_d T_s) \quad ; \delta = \sin(\omega_d T_s)$$

$$a_1 = -2\alpha\beta \quad ; a_2 = \alpha^2$$

$$b_1 = 1 - \alpha\left(\beta + \delta\frac{\zeta\omega_0}{\omega_d}\right)$$

$$b_2 = \alpha^2 + \alpha\left(\delta\frac{\omega_0}{\omega_d} - \beta\right)$$

กำหนดให้

$H_{ref}(q^{-1})$	ฟังก์ชันถ่ายโอนในรูปโดเมนความไม่ต่อเนื่องเชิงเวลาของแบบจำลองอ้างอิง (Discrete Time Domain Transfer Function of Reference Model)
T_s	คาบเวลาชักตัวอย่าง (Sampling Period)

ฟังก์ชันถ่ายโอนของแบบจำลองอ้างอิงเป็นตัวกำหนดสมรรถนะโดยรวมของระบบควบคุมแบบปรับตัวตามแบบจำลองอ้างอิง (Model-Reference Adaptive Control)

จากขั้นตอนการคำนวนและการประมวลผลของการประยุกต์ใช้เครือข่ายนิวรอน ในระบบควบคุมแบบปรับตัวเองตามแบบจำลองอ้างอิงดังที่กล่าวในบทนี้ นำไปใช้ทดสอบกับระบบควบคุมกระบวนการแลกเปลี่ยนความร้อนในเวลาจริง ซึ่งจะกล่าวถึงในบทถัดไป

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย