

## รายการอ้างอิง

### ภาษาไทย

- นฤพนธ์ มัญญมณี. ตัวควบคุมกระบวนการแบบฟuzzyลอจิก. วิทยานิพนธ์ปริญญาโทบริหารธุรกิจ  
ภาควิชาวิศวกรรมเคมี บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2538
- ธวัชชัย ชยวานิช. ระบบควบคุมแบบฟuzzyลอจิก (ตอนที่ 1). วิศวกรรมสาร. ปีที่ 49 เล่มที่ 12  
ธันวาคม 2539
- ธวัชชัย ชยวานิช. ระบบควบคุมแบบฟuzzyลอจิก (ตอนที่ 2). วิศวกรรมสาร. ปีที่ 50 เล่มที่ 2  
กุมภาพันธ์ 2540

### ภาษาอังกฤษ

- Babuska R., H.B. Verbruggen, "An Overview of Fuzzy Modeling for Control",  
*Control Eng. Practice*, Vol.4, No.11, p.1593-1606, 1996
- Bruce P. Graham, Robert B. Newell., "Fuzzy Adaptive Control of a First-order Process"  
*Fuzzy Sets and Systems*, 31 p.47-65, 1989
- Couillard, D. and S. Zhu, "Control Strategy for Activated Sludge Process Under Shock  
Loading", *Water Research*, Vol.20, No. 5, p.649-655, 1995
- Czogala E., Witold Pedrycz., "On Identification in Fuzzy Systems and Its Applications in  
Control Problems" *Fuzzy Sets and System*, 6 p.73-83, 1981
- George Bojadziev, Maria Bojadziev, "Fuzzy Sets, Fuzzy Logic Applications",  
*World Scientific* 1995
- George J. Klir, Bo Yuan, "Fuzzy Sets and Fuzzy Logic Theory and Applications"  
*Prentice-Hall , Inc.*, 1995
- George J. Klir, Ute H. St. Clair, Bo Yuan, "Fuzzy Set Theory Foundations and Applications"  
*Prentice-Hall , Inc.*, 1997

- Hanakuma, Y., Y. Irizuli, M. Acachi and E. Nakanishi, "Design of a Self-tuning Fuzzy Control System and the Application to a Distillation Column" *International Chemical Engineering*, Vol.34, No.1, p.91-96, 1994
- Igor Skrjanc, Katarina Kavsek-Biasizzo, "Fuzzy Predictive Control Based on Relational Matrix Models", *Computers Chem. Engng.*, Vol.20., p s931-s936, 1996
- Katarina Kavsek-Biasizzo, Igor Skrjanc, Drago Matko, "Fuzzy Predictive Control of Highly Nonlinear pH Process" *Computers chem Engng.* Vol.21, p.S613-618, 1997
- Kim Euntai, Minkee Park, Seunghwan Ji and Mifnon Park, "A New Approach to Fuzzy Modeling" *IEEE Trans. On Syst. Man. and Cybern.*, Vol. 5 No.5, August 1997
- Koffman, S.J., R.R. Fullmer and R.C. Brown, "Fuzzy Logic Control of a Fluidized Bed Combustor", *Prcc. 1989 Am. Control Conf.*, Vol.3, p.3756-3758, 1989
- Kuipers, B. and F. Astroms, "The Composition and Validation of Heterogeneous Control Laws", *Automatica*, Vol.30, p.233-249, 1994
- Henson, M. A., Dale E. Seborg, "Nonlinear Process Control"  
*Prentice-Hall , Inc.*, 1997
- Lyben William L., "Process Modeling, Simulation and Control for Chemical Engineers 2<sup>nd</sup> Edition", *McGraw-Hill International*, 1990
- Lyben William L and Lyben Michael L., "Essentials of Process Control"  
*McGraw-Hill International*, 1997
- Pedrycz W., "Identification in Fuzzy Systems" *IEEE Trans. On Syst. Man. and Cybern.*, Vol. SMC-14, No.2 Mar/Apr 1984
- Postlethwaite Bruce E., "The Development of Model-Based Fuzzy Controller"  
*Ph.D. Thesis. (University of Strathclyde, Glasgow)*. 1991

- Postlethwaite B. E., "Empirical Comparison of Method of Fuzzy Relational Identification",  
*IEE Proceedings-D*, 138(3), p.199-206, 1991
- Postlethwaite B. E., "A Model-based Fuzzy Controller", *Trans. IChemE*, 72(A)  
p.38-46, Jan. 1994
- Ridley. J. N., Shaw, I.S. and Kruger, J.J., "Probabilistic Fuzzy Model for Dynamics Systems"  
*Electronics Letters*, 24(14): 890-892, 1988
- Shah, I., and K. Rajamani, "Fuzzy Logic Controller: Application to Liquid Level System"  
*Miner Metal Process Minerals & Metallurgical Processing*, p.186-192, 1988
- Shah, I., and K. Rajamani, "Self-organizing Controller for Process pH Control" *Control 90*  
*Miner Metal Process Minerals & Metallurgical Processing*, p.45-52, 1990
- Stephanopoulos G., "Chemical Process Control: An Introduction to Theory and Practice"  
*Prentice-Hall , Inc.*, 1997
- Stoll, K.E., P.A. Ralston and S. Ramanganesan, "Simplify Fuzzy Logic Control  
Implementation" *Hydrocarbon Processing*, p 49-55, 1993
- Tomohiro Takagi and Michio Sugeno, "Fuzzy Identification of Systems and Its Applications  
to Modeling and Control", *IEEE Trans. On Syst. Man. and Cybern.*,  
Vol. SMC-15, No.1, Jan.Feb.1985
- Tong R.M., "An Assessment of a Fuzzy Control Algorithm for a Nonlinear Multivariable",  
*Tech. Report EES-MMS-DSFR-76 Queen Mary Collage*, London, 1976
- Tong R.M., M. B., Beck, and A. Laten, "Fuzzy Control of Activated Sludge Waste Water  
Treatment process", *Automatica*, 16, p.659-701 1980
- Tong R.M., "The Evaluation of Fuzzy Models Derived from Experimental Data",  
*Fuzzy Sets and Systems*, 4 p.1-12, 1980

Walter, H.B., J.M. Robert and S.S. Sam, "Design of a Self-tuning Rule-based Controller for a Gasoline Refinery Catalytic Reformer" *IEEE Trans on Automatic Control*, Vol.35, p.156-164, 1990

Wang Li-Xin, "A Course in Fuzzy Systems and Control"  
*Prentice-Hall , Inc.*, 1997

William L. Luyben, "Process Modeling, Simulation and Control for Chemical Engineering 2<sup>nd</sup> Edition", *McGraw-Hill, Inc.*, 1990

Xu Chen-Wei, "Fuzzy Systems Identification" *IEE Proceedings*, Vol.136 Pt. D, No. 4, July 1989

Yamashita Y., S. Mitsumoto and M. Suzuki, "Start-up of a Catalytic Reactor by Fuzzy Controller", *Journal of Chemical Engineering of Japan*, Vol.21, No.3, p.277-282, 1988

Zadeh Lotfi A., "Outline of a New Approach to the Analysis of Complex System and Decision Processes", *IEEE Trans. on Syst. Man. and Cybern.*, Vol. SMC-3, No.1 Jan.1973

## ภาคผนวก ก.

### การอินทิเกรตเชิงตัวเลขสำหรับสมการ ODE

#### Numerical Integration of Ordinary Differential Equation

ในการใช้โปรแกรมคอมพิวเตอร์มาใช้ในการทำงานวิจัยนี้ จะทำการเลียนแบบพลวัตของกระบวนการที่เกิดขึ้นโดยใช้สมการทางคณิตศาสตร์ที่มีอยู่ซึ่งเป็นสมการอนุพันธ์แบบธรรมดาที่ไม่เป็นเชิงเส้น (Non-linear Ordinary Differential Equation) มาเป็นแบบจำลองของกระบวนการที่ต้องการจะควบคุม ในการแก้สมการประเภทนี้ในระบบงานทางด้านวิศวกรรมเคมีจะใช้เทคนิคการอินทิเกรตเชิงตัวเลขสำหรับสมการอนุพันธ์แบบธรรมดา (Ordinary Differential Equations (ODE)) มาเป็นเทคนิคในการแก้สมการ เทคนิคนี้เป็นการนำสมการดิครีต ไฟไนต์-ดิฟเฟอเรนเชียล (Discrete Finite-difference Equations) มาใช้ในการประมาณสมการอนุพันธ์แบบต่อเนื่อง (Continuous Differential Equations)

ในการเลือกใช้เทคนิคการประมาณสมการโดยใช้วิธีการประมาณเชิงตัวเลขนั้น ต้องคำนึงถึงความแม่นยำ เสถียรภาพ และความเร็วในการหาค่าของสมการที่ประมาณออกมาเป็นสำคัญ นอกจากนี้ยังต้องคำนึงถึงความรวดเร็วและความซับซ้อนในการคำนวณ ซึ่งแต่ละเทคนิคก็มีวิธีการความยากง่าย และความเหมาะสมที่แตกต่างกันไป อย่างไรก็ตามอัลกอริธึมของออยเลอร์อันดับหนึ่ง (First-order Explicit Euler Algorithm) เป็นวิธีที่ได้รับความนิยม สำหรับการประยุกต์ใช้ในงานด้านวิศวกรรมเคมีที่ผ่านมามากมาย

ขนาดของขั้น (Step Size) ที่ใช้ในการอินทิเกรตเชิงตัวเลขนั้น ต้องเลือกให้เล็กพอสมควร มิฉะนั้นจะมีผลทำให้ค่าที่คำนวณออกมาได้แม่นยำ ผิดจากความเป็นจริงไป นอกจากนี้แล้วยังทำให้ไม่เสถียรอีกด้วย เช่น เกิดการแกว่งในช่วงที่กว้างกว่าความเป็นจริง และในบางกรณีก็จะผิดไปจากความเป็นจริง เช่น กรณีที่เป็นเศษส่วนโมล จะทำให้เศษส่วนโมลที่คำนวณออกมามีค่าติดลบ หรือมีค่ามากกว่าหนึ่ง ซึ่งไม่ตรงกับความเป็นจริงได้ แต่ถ้าเลือกขนาดของขั้น (Step Size) ที่เล็กเกินไปก็จะทำให้ใช้เวลาคำนวณมากขึ้นอีก ยิ่งมีหลายสมการก็ยิ่งต้องใช้เวลาในการคำนวณมากขึ้น

รูปแบบสมการทั่วไปของสมการอนุพันธ์อันดับที่  $N$  ใดๆ

$$a_N \frac{d^N x}{dt^N} + a_{N-1} \frac{d^{N-1} x}{dt^{N-1}} + \dots + a_1 \frac{dx}{dt} + a_0 x = f(t) \quad (ก 1)$$

โดยที่  $a_0, a_1, a_2, \dots, a_N$  เป็นค่าคงที่

$f(t)$  คือเป็นค่าของอินพุทหรือตัวรบกวนที่ใส่เข้าไปในระบบ (Forcing Function, Disturbance)

$x(t)$  คือตัวแปรสถานะของกระบวนการ

กรณีที่พบบ่อยๆ เป็นกรณีที่มีค่า เป็น 1 และ 2 จะทำให้ได้สมการเป็น

สมการอันดับที่หนึ่ง

$$a_1 \frac{dy}{dt} + a_0 x = f(t) \quad (ก.2)$$

สมการอันดับที่สอง

$$a_2 \frac{d^2 x}{dt^2} + a_1 \frac{dy}{dt} + a_0 x = f(t) \quad (ก.3)$$

โดยที่สามารถแปลงให้อยู่ในรูปแบบมาตรฐานได้ดังสมการที่ (ก.4) และ (ก.5) ดังนี้

สมการอันดับที่หนึ่ง

$$\tau \frac{dy}{dt} + K_p x = f(t) \quad (ก.4)$$

สมการอันดับที่สอง

$$\tau^2 \frac{d^2 x}{dt^2} + 2\tau\zeta \frac{dy}{dt} + x = f(t) \quad (ก.5)$$

เมื่อ

- $K_p$  คือ อัตราขยายเชิงสถิติของกระบวนการ (Process Gain)
- $\tau$  คือ ค่าคงที่เวลาของกระบวนการ (Process Time Constant)
- $\zeta$  คือ สัมประสิทธิ์ความหน่วง (Damp Coefficient)

### ก.1 ออยเลอร์อัลกอริทึม (Euler Algorithm)

ออยเลอร์อัลกอริทึมจัดเป็นวิธีการอินทิเกรตเชิงตัวเลขที่ง่ายไม่ซับซ้อน จึงเป็นประโยชน์ในการนำไปใช้แก้ปัญหา สมมติว่าต้องการแก้สมการ

$$\frac{dx}{dt} = f(x, t) \quad (\text{ก.6})$$

โดยทั่วไปแล้ว  $f(x, t)$  เป็นฟังก์ชันที่ไม่เป็นเชิงเส้นทำให้ไม่สามารถแก้สมการได้ด้วยวิธีการอินทิเกรตธรรมดา ในการอินทิเกรตเชิงตัวเลขจะต้องทราบสภาวะเริ่มต้นของ  $x$  ก่อน โดยทั่วไปจะใช้ที่เวลา เท่ากับ 0 หรือที่สภาวะคงตัวใดๆ

$$x_{(0)} = x_0 \quad \text{ที่เวลา} \quad t = 0 \quad (\text{ก.7})$$

เมื่อสมมติให้ เริ่มต้นที่เวลา  $t_0 = 0$  เมื่อเวลาผ่านไป  $\Delta t$  เราจะสามารถประมาณค่า  $x$  ที่เวลา  $\Delta t$  ได้เท่ากับค่า  $x$  ที่เวลาเดิม  $t_0$  บวกกับผลคูณระหว่างอนุพันธ์ของ  $x$  และ  $\Delta t$  ดังสมการที่ (ก.8)

$$x_{(\Delta t)} = x_{(0)} + \left( \frac{dx}{dt} \right)_{t=0} \cdot \Delta t \quad (\text{ก.8})$$

$$x_1 = x_0 + f(x_0, 0) \cdot \Delta t \quad (\text{ก.9})$$

เมื่อค่า  $\Delta t$  มีค่าเล็กพอก็จะทำให้ค่า  $x_1$  ที่ประมาณได้มีค่าใกล้เคียงกับค่าที่ถูกต้องมากที่สุด และเมื่อต้องการหาค่า  $x$  ที่เวลา  $t = t_2 = 2\Delta t$  ก็จะประมาณ  $x_{(2\Delta t)} \equiv x_2$  ก็จะได้ว่า

$$x_{(2\Delta t)} = x_{(\Delta t)} + \left( \frac{dx}{dt} \right)_{t=\Delta t} \cdot \Delta t \quad (\text{ก.10})$$

$$x_2 = x_1 + f(x_1, t_1) \cdot \Delta t \quad (\text{ก.11})$$

จึงสามารถเขียนในรูปทั่วไปได้ดังนี้

$$x_{(n+1)} = x_n + f(x_n, t_n) \cdot \Delta t = x_n + \left( \frac{dx}{dt} \right)_{t=t_n} \cdot \Delta t \quad (\text{ก.12})$$

โดยที่ 
$$t_{n+1} = t_n + \Delta t \quad (\text{ก.13})$$

ในกรณีที่มีตัวแปรมากกว่า 1 ตัว เช่น มีตัวแปร 2 ตัว คือ  $x_1$  และ  $x_2$  การอินทิเกรตสมการสามารถทำได้เป็น

$$\frac{dx_1}{dt} = f_1(x_1, x_2, t) \quad \frac{dx_2}{dt} = f_2(x_1, x_2, t) \quad (\text{ก.14})$$

จะสามารถประมาณค่า  $x_1$  และ  $x_2$  ได้เป็น

$$x_{1,n+1} = x_{1n} + \Delta t \cdot f_1(x_{1n}, x_{2n}, t_n) \quad (\text{ก.15})$$

$$x_{2,n+1} = x_{2n} + \Delta t \cdot f_2(x_{1n}, x_{2n}, t_n) \quad (\text{ก.16})$$



## ภาคผนวก ข.

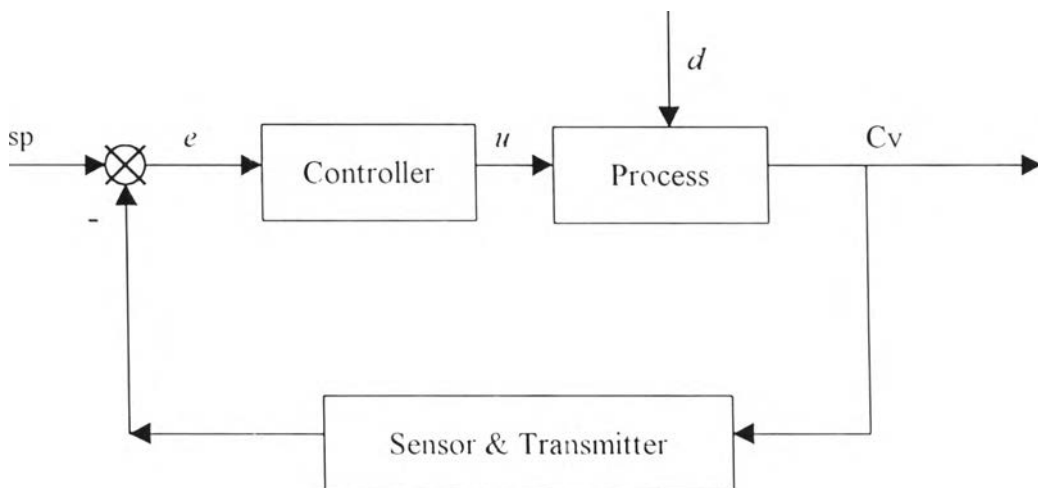
### ตัวควบคุมแบบพีไอดี และการปรับจูน

#### PID CONTROLLER and CONTROLLER TUNING

ในบทนี้จะกล่าวถึงตัวควบคุมแบบป้อนกลับพีไอดี และการปรับจูนตัวควบคุมแบบป้อนกลับพีไอดี ซึ่งได้แก่การปรับจูนแบบทั่วไป และการปรับจูนแบบลองผิดลองถูก และสำหรับตอนท้ายจะพิจารณาการตอบสนองของกระบวนการและ เกณฑ์การพิจารณาสมรรถนะของการควบคุม

##### ข.1 ทฤษฎีของตัวควบคุมแบบป้อนกลับพีไอดี

การควบคุมแบบป้อนกลับพีไอดีเป็นการควบคุมดั้งเดิมแบบพื้นฐานที่ใช้กันทั่วไปในทางอุตสาหกรรม มีโครงสร้างการควบคุมดังแสดงในรูปที่ ข.1 โดยที่ตัวควบคุมแบบป้อนกลับมีหลักการควบคุมโดยจะใช้ค่าตัวแปรควบคุมจากกระบวนการ  $Cv$  เปรียบเทียบกับ ค่าเป้าหมายของกระบวนการ  $sp$  เป็นค่าความผิดพลาดในการควบคุม  $e = sp - Cv$  กระบวนการ นำมาสร้างสัญญาณเอาต์พุต  $u$  ออกมาเพื่อปรับสถานะของกระบวนการให้เข้าสู่เป้าหมายการควบคุม สัญญาณเอาต์พุตของตัวควบคุมจะมีความสัมพันธ์กับค่าความผิดพลาดของการควบคุม ตามลักษณะชนิดของตัวควบคุม



รูปที่ ข.1 ระบบการควบคุมแบบป้อนกลับ (Feedback Control System)

เครื่องควบคุมสามารถแบ่งตามโครงสร้างการทำงานได้เป็น 3 ชนิด คือ เครื่องควบคุมแบบนิวมेटิกส์ ทำงานโดยใช้สัญญาณลม เครื่องควบคุมแบบอิเล็กทรอนิกส์ (Electronics Controller) ทำงานโดยใช้วงจรรีเลย์ทรานซิสเตอร์เชิงเส้น (Linear Circuit) ในการสร้างสัญญาณควบคุม และเครื่องควบคุมแบบดิจิทัล ทำงานโดยใช้วงจรรตรรก (Logic Circuit) หรือไมโครโปรเซสเซอร์สร้างสัญญาณควบคุมโดยการเลียนแบบและจำลองการทำงานของเครื่องควบคุมแบบนิวมेटิกส์และเครื่องควบคุมแบบอิเล็กทรอนิกส์จากแบบจำลองทางคณิตศาสตร์ของเครื่องควบคุมแบบเดิม เครื่องควบคุมแบบพื้นฐานที่ใช้ในระบบการอุตสาหกรรมมี 3 แบบ คือ เครื่องควบคุมแบบพี เครื่องควบคุมแบบพีไอ และเครื่องควบคุมแบบพีไอดี

### ข.1.1 เครื่องควบคุมแบบพี (Proportional Controller, (P Controller))

เครื่องควบคุมแบบพี มีสัญญาณควบคุมแปรผันตามค่าความคลาดเคลื่อนในการควบคุม

$$u(t) = K_C e(t) + u_s \quad (ข.1)$$

เมื่อ	$K_C$	คือเกนสัดส่วน (Proportional Gain) ของตัวควบคุม,
	$u_s$	คือค่าไบแอสหรือสัญญาณหน่วงของตัวควบคุม (Bias Signal)
	$u(t)$	คือค่าเอาต์พุตของตัวควบคุม

เครื่องควบคุมแบบพีสามารถแสดงอัตราการควบคุมได้ 2 วิธี คือ แสดงเป็นสัดส่วนเกน และแบนด์สัดส่วน PB (Proportional Band) โดยที่  $PB = 100/K_C$  แบนด์สัดส่วนหมายถึงค่าความผิดพลาดในการควบคุม  $e$  ที่ทำให้สัญญาณเอาต์พุต  $u(t)$  มีค่าสูงสุด เช่น เมื่อตัวควบคุมแบบพีมีค่าแบนด์สัดส่วน  $PB = 200\%$  จะมีค่า  $K_C = 10/200 = 0.5$  แทนค่า  $K_C$  ลงในสมการที่ (ข.1) โดยสมมติค่า  $u_s = 0$

$$u(y) = 0.5 * e(t) \quad (ข.2)$$

แสดงว่าค่าความผิดพลาดในการควบคุม  $e(t)$  จะต้องมีค่าเป็น 2 หรือ 200% จึงจะทำให้สัญญาณเอาต์พุตมีค่าสูงสุดคือ 100% ในระบบการอุตสาหกรรมทั่วไปสามารถใช้ตัวควบคุมที่มีเกนสัดส่วนอยู่ในช่วง  $0.2 \leq K_C \leq 100$  หรือค่าแบนด์สัดส่วนในช่วง  $1 \leq PB \leq 500$  เครื่องควบคุมที่มีค่าสัดส่วนเกนมากหรือมีค่าแบนด์สัดส่วนน้อย จะมีความไวในการตอบสนองต่อค่าความผิดพลาดสูง

### ข.1.2 เครื่องควบคุมแบบพีไอ (Proportional-Integral Controller,(PI Controller))

เครื่องควบคุมแบบพีไอ มีสัญญาณควบคุมมีความสัมพันธ์กับค่าความคลาดเคลื่อนในการควบคุมตามสมการที่ (ข.3) ดังนี้

$$u(t) = K_C e(t) + \frac{K_C}{\tau_I} \int_0^t e(t) dt + u_S \quad (\text{ข.3})$$

เมื่อ  $\tau_I$  คือ ค่าคงที่เวลาอินทิกรัล (Integral Time Constant) หรือเวลารีเซ็ต (Reset Time) ของตัวควบคุมแบบไอ ตัวควบคุมแบบไอแสดงอัตราการควบคุมโดยใช้ค่าคงที่เวลาอินทิกรัลหรือเวลารีเซ็ต มีหน่วยเป็น ครั้งต่อนาที ซึ่งมีค่าเป็น  $1/\tau_I$

ในกระบวนการอุตสาหกรรมทั่วไปสามารถใช้ตัวควบคุมที่มีค่าคงที่เวลาอินทิกรัล  $0.1 \leq \tau_I \leq 50$  นาที รูปที่ ข.2 แสดงการตอบสนองของตัวควบคุมแบบพีไอเริ่มต้นที่เวลา  $t = 0$  สัญญาณเอาต์พุตจากการควบคุมแบบพีไออย่างเดียวจะมีค่า  $u = K_C * e(t)$  เมื่อเวลาผ่านไป  $\tau_I$  สัญญาณเอาต์พุตจากการควบคุมแบบไอจะมีค่าเท่ากับสัญญาณเอาต์พุตแบบพีไออย่างเดียว

$$\frac{K_C}{\tau_I} \int_0^{\tau_I} e(t) dt = \frac{K_C}{\tau_I} e \tau_I = K_C e \quad (\text{ข.4})$$

ค่าคงที่เวลาอินทิกรัล  $\tau_I$  หมายถึงช่วงเวลาที่ตัวควบคุมใช้ในการเพิ่มสัญญาณเอาต์พุตแบบไอจนมีค่าเท่ากับสัญญาณที่ได้จากการควบคุมแบบพีไอเพียงอย่างเดียวหนึ่งครั้ง หรือ ตัวควบคุมแบบไอต้องใช้เวลา  $\tau_I$  เพื่อเพิ่มสัญญาณเอาต์พุตให้มีค่าเท่ากับสัญญาณที่ได้จากการควบคุมแบบพีไอหนึ่งครั้ง ตัวควบคุมแบบไอมีคุณสมบัติต่างจากตัวควบคุมแบบพีไอคือ สัญญาณเอาต์พุตจะมีการเปลี่ยนแปลงค่าเพิ่มขึ้นหรือลดลงตลอดเวลาที่ยังมีค่าความผิดพลาดในการควบคุมอยู่

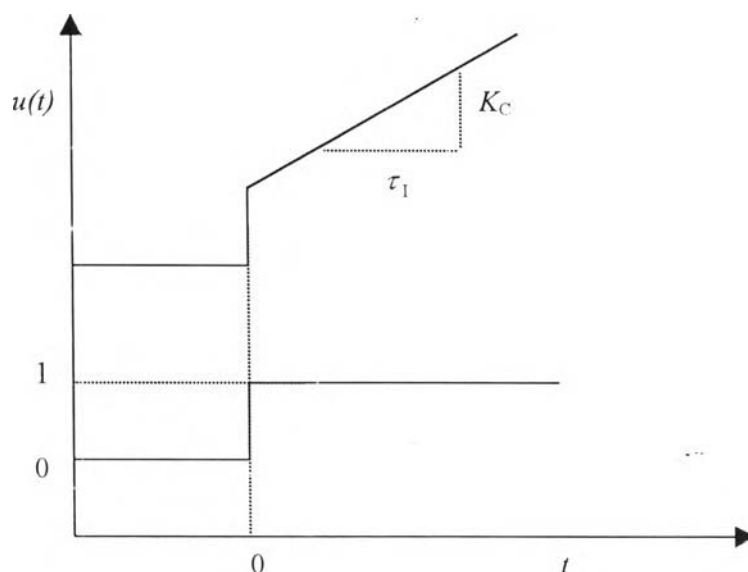
### ข.1.3 เครื่องควบคุมแบบพีไอดี

(Proportional-Integral-Derivative Controller,(PID Controller))

เครื่องควบคุมแบบพีไอดี มีสัญญาณควบคุมมีความสัมพันธ์กับค่าความคลาดเคลื่อนในการควบคุมตามสมการที่ (ข.5) ดังนี้

$$u(t) = K_c e(t) + \frac{K_c}{\tau_I} \int_0^t e(t) dt + K_c \tau_D \frac{de(t)}{dt} + u_s \quad (\text{ข.5})$$

เมื่อ  $\tau_D$  คือค่าคงที่เวลาดอนุพันธ์ (Derivative Time Constant) ของตัวควบคุมแบบตี ตัวควบคุมสร้างสัญญาณเอาต์พุตแบบตีจากเทอม  $\frac{de(t)}{dt}$  ในสมการที่ (ข.5) ซึ่งมีค่าเปลี่ยนแปลงทันทีที่ค่าความผิดพลาดในการควบคุม  $e(t)$  มีการเปลี่ยนแปลงไม่ว่าจะมีค่าเพิ่มขึ้นหรือลดลง และสัญญาณการควบคุมแบบตีจะถูกสร้างเพื่อลดความผิดพลาดในการควบคุมล่วงหน้า



รูปที่ ข.2 การตอบสนองของตัวควบคุมแบบพีไอ

## ข.2 การปรับจูนตัวควบคุมตัวควบคุมป้อนกลับแบบดั้งเดิม

การปรับจูนตัวควบคุมแบบป้อนกลับมีหลายวิธี โดยทั่วไปแล้วเป็นการปรับจูนจากการทดลอง การคาดเดาจากประสบการณ์ของผู้ปรับจูนเอง เนื่องจากงานในวิจัยนี้เป็นการควบคุมที่อยู่ในใหม่โดเมน จึงเสนอวิธีการและหลักเกณฑ์ทั่วไปในการปรับจูนในใหม่โดเมน

### ข.2.1 หลักการจูนโดยทั่วไป

การควบคุมกระบวนการพื้นฐานที่มีอยู่ในอุตสาหกรรมเคมี โดยทั่วไป อันได้แก่ การควบคุมระดับ, การไหล, อุณหภูมิ และความดัน มักจะมีลักษณะที่คล้ายกันไม่ว่าจะเป็นการประยุกต์ใช้กับกระบวนการผลิตแบบใดก็ตาม เช่นในการควบคุมที่เกี่ยวกับการไหลทั้งหมด มักจะให้ตัวควบคุมแบบพีไอเป็นหลัก โดยตั้งค่าแบนด์สัดส่วน PB ให้กว้าง และให้มีเวลารีเซ็ต  $\tau_I$  ที่เร็ว เป็นต้น ซึ่งมี หลักเกณฑ์กว้างๆ ที่น่าสนใจที่ใช้ในทางปฏิบัติทั่วไป ดังต่อไปนี้

- การควบคุมการไหล

การควบคุมการไหล มักใช้การควบคุมแบบ พีไอ (PI Controller) โดยให้มีค่าแบนด์สัดส่วน (Proportional Band (PB)) ที่กว้าง (PB=150) หรือกล่าวอีกนัยหนึ่งว่า ใช้ค่าสัดส่วนเกณฑ์ต่ำเพื่อลดผลกระทบที่เกิดจากสัญญาณรบกวนอันเนื่องมาจากการไหลแบบเทอร์บูเลนซ์ และเลือกใช้ค่าคงที่เวลาอินทิกรัล (Integral Time Constant) หรือค่าเวลารีเซ็ต (Reset Time) ให้มีค่าต่ำ ( $\tau_I=0.1$  นาทีต่อครั้ง) เพื่อให้ตัวแปรควบคุมสามารถเข้าสู่ค่าเป้าหมายได้อย่างรวดเร็ว

- การควบคุมระดับ

การควบคุมระดับของเหลวมักเป็นการควบคุมที่อยู่ในถังกักเก็บหรือถังพัก ก่อนที่ของไหลนั้นๆ จะถูกป้อนเข้าสู่กระบวนการต่อไป ลักษณะการควบคุมก็เพียงแต่เป็นควบคุมให้ระดับของเหลวอยู่ในช่วงสูงสุดและต่ำสุดช่วงหนึ่ง ดังนั้นสำหรับการควบคุมในลักษณะนี้ นิยมใช้การควบคุมแบบ พี (Proportional Controllers (P Controllers)) เพื่อให้มีการเปลี่ยนแปลงอัตราการไหลที่ราบเรียบสม่ำเสมอ และเพื่อไม่ให้เกิดการผันแปรมากต่ออัตราการไหลที่เข้าสู่กระบวนการ (Downstream Units) ต่อไป

แต่ถ้าหากเป็นการควบคุมระดับในลักษณะที่เป็นถังต่ออนุกรมกันหลายถัง (Multiple Unit in Series) ซึ่งต้องการรักษาระดับของของไหลให้คงที่ด้วย มักจะใช้โครงสร้างการควบคุมที่ใช้จะเป็นการควบคุมแบบพีไอ (PI Controller) เพราะว่าถ้าใช้การควบคุมแบบพีเพียงอย่างเดียวเท่านั้น จะทำให้อัตราการไหลเปลี่ยนแปลงไปบ้าง โดยไม่มีการโอเวอร์ชูท แต่ไม่สามารถรักษาระดับให้คงที่ตามค่าเป้าหมายได้

การใช้ตัวควบคุมแบบพีไอสำหรับการควบคุมระดับนั้น เทอมของอินทิกรัลในตัวควบคุมแบบพีไอจะทำให้ระดับเข้าสู่ค่าเป้าหมายได้ และเมื่อเกิดการเปลี่ยนแปลงอัตราการไหลขาเข้าสู่ระบบ จะทำให้อัตราการไหลขาออกจากถังเปลี่ยนแปลงไปทันที ตัวอย่างของกระบวนการควบคุมระดับที่ต้องการใช้

ควบคุมแบบพีไอ ได้แก่ การควบคุมของของเหลวภายในถังปฏิกรณ์ เพื่อให้ของเหลวที่เป็นสารตั้งต้นอยู่ในเครื่องปฏิกรณ์เกิดปฏิกิริยาในเวลาที่ต้องการ

- การควบคุมอุณหภูมิ

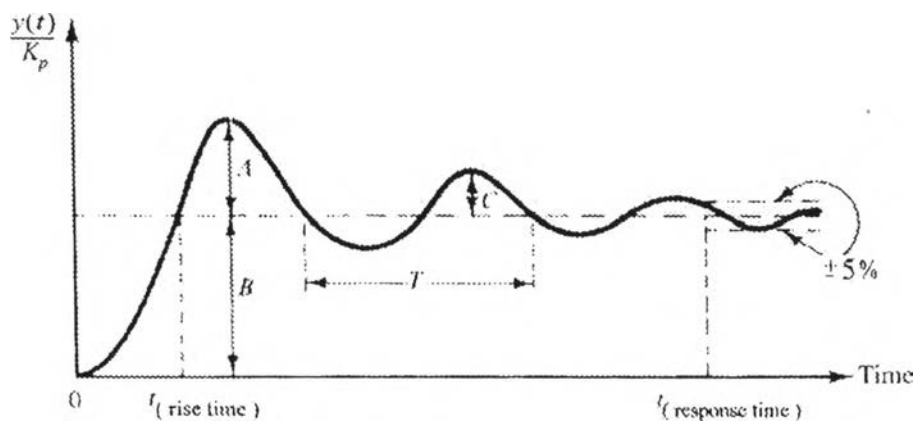
กระบวนการควบคุมอุณหภูมิเป็นการควบคุมที่ค่อนข้างช้า เนื่องจากความล่าช้าในการส่งสัญญาณของตัวเซนเซอร์อันเนื่องมาจากความล่าช้าของกระบวนการถ่ายเทความร้อนของกระบวนการ ตัวควบคุมที่ถูกนำมาใช้ในกระบวนการนี้มักเป็นตัวควบคุมแบบพีไอดี (PID Controllers) โดยตั้งให้ค่าแบนด์ลัดส่วน (PB) มีค่าค่อนข้างน้อย ตามขนาดของวาล์วควบคุมและ ย่านของทรานสมิตเตอร์ และให้ค่าเวลารีเซต  $\tau_I$  มีค่าเป็นไปตามค่าคงที่เวลาของกระบวนการ เช่นถ้า กระบวนการเป็นไปอย่างรวดเร็วก็ให้ค่าเวลารีเซตมีค่าน้อย ส่วนค่าคงที่เวลาอนุพันธ์  $\tau_D$  (Derivative Time Constant) ถูกเซตให้มีค่าประมาณเท่ากับ  $1/4$  เท่าของค่าคงที่เวลาของกระบวนการ ทั้งนี้ขึ้นอยู่กับสัญญาณรบกวนที่อยู่ในตัวทรานสมิตเตอร์

## ข.2.2 การจูนเครื่องควบคุมโดยวิธีทดลองผิดทดลองถูก (On-line Trial and Error)

การปรับจูนเครื่องควบคุมโดยใช้สูตรสำเร็จจากการทดลองบางครั้งไม่สามารถทำให้การควบคุมกระบวนการได้ผลดีเท่าที่ควร เนื่องจากกระบวนการที่ใช้ในการทดลองหาสูตรสำหรับปรับเครื่องควบคุมมีคุณสมบัติบางอย่างต่างจากกระบวนการจริงที่ต้องการจะควบคุม ค่าตัวแปรควบคุมที่ได้จากสูตรสำเร็จเพียงสามารถใช้เป็นค่าเริ่มต้นในการปรับเครื่องควบคุมโดยไม่ทำให้ระบบสูญเสียเสถียรภาพ และทดลองปรับเรื่องควบคุมโดยการสังเกตผลการควบคุมจากการพิจารณาค่าควบคุมของตัวแปรกระบวนการ ซึ่งมีขั้นตอนต่างๆ ดังต่อไปนี้

- 1.) ทำการจูนโดยไม่ให้มีการคิดการทำงานของเทอมอินทิกรัล และเทอมของอนุพันธ์ โดยให้ค่าคงที่เวลาอินทิกรัล  $\tau_I$  มีค่ามากๆ และให้ค่าคงที่เวลาอนุพันธ์  $\tau_D$ , (Derivative Time Constant) มีค่าน้อยๆ
- 2.) ตั้งค่าแบนด์ลัดส่วน PB ไว้ให้สูงๆ อาจให้เป็น 200
- 3.) ติดตั้งตัวควบคุม
- 4.) ให้มีการเปลี่ยนแปลงค่าเป้าหมายหรือโหลดเพียงเล็กน้อย แล้วสังเกตดูการตอบสนองของตัวแปรควบคุม เนื่องจากค่าเกนต่ำดังนั้นจึงทำให้การตอบสนองช้า
- 5.) ลดค่าแบนด์ลัดส่วน (PB) ลงครึ่งหนึ่ง หรือเพิ่มค่าเกนเป็น 2 เท่า ตั้งค่าเป้าหมายใหม่
- 6.) ลดค่าแบนด์ลัดส่วนซ้ำตามข้อ 5 จนกว่าได้การตอบสนองในลักษณะอันเดอร์แดมพ์มาก (Underdamp) และเกิดแกว่ง (Oscillatory) ค่าเกนที่ได้สุดท้ายนี้เรียกว่า  $K_{cu}$  "Ultimate gain" ซึ่งเป็นค่าเกนของตัวควบคุมสูงสุดที่ทำให้กระบวนการแบบวงจรมีเสถียรภาพที่ดีเมื่อใช้ การควบคุมแบบพีไอเพียงอย่างเดียว

- 7.) กลับไปตั้งค่าแบนด์สตัดส่วน PB เป็น 2 เท่าของค่าเดิม หรือให้ ค่าเกน  $K_c$  เป็นครึ่งเท่าของ  $K_{cu}$
- 8.) ปรับค่า  $\tau_I$  ลงทีละครั้ง ให้มีตัวรบกวนกระบวนการเล็กน้อย แล้วสังเกตดูการเปลี่ยนแปลงที่เกิดขึ้น
- 9.) เลือกค่า  $\tau_I$  ที่ทำให้เกิดการตอบสนองในลักษณะอันดับเดมพ์ จากนั้นตั้งค่า  $\tau_I$  ไว้ที่ 2 เท่าของค่าสุดท้ายที่ได้
- 10.) ปรับค่า  $\tau_D$  โดยให้มีค่าเพิ่มขึ้น และให้มีตัวรบกวนเกิดขึ้นในกระบวนการ เลือกค่า  $\tau_D$  ที่ทำให้สามารถควบคุมได้ดีโดยไม่ทำให้สัญญาณรบกวนกระบวนการถูกขยาย
- 11.) ลดค่าแบนด์สตัดส่วนลงอีกทีละ 10% จนกว่าจะได้ค่า (Damping Coefficient) หรือ โอเวอร์ชูทที่เหมาะสม



รูปที่ ข.3 ลักษณะการตอบสนองของกระบวนการ

### ข.3 หลักเกณฑ์ในการพิจารณาสมรรถนะของระบบการควบคุม (Control Performance Criterion)

ในการประเมินผลการควบคุมของระบบการควบคุมที่ออกแบบมา จำเป็นต้องมีการกำหนดหลักเกณฑ์ในการตัดสินเพื่อใช้เป็นมาตรฐานเปรียบเทียบกับระบบการควบคุมอื่นๆ การควบคุมแบบใดให้ผลการควบคุมได้ดีที่สุดตามเป้าหมายการควบคุมและความต้องการที่กำหนดไว้ ซึ่งมี 2 วิธีการด้วยกัน ได้แก่

#### 1. พิจารณาจากผลการตอบสนองของกระบวนการ

ระบบการควบคุมที่ดีต้องมีความสามารถในการลดค่า โอเวอร์ชูท (a) ลดค่าเวลาที่ใช้ในการเข้าสู่สมดุล  $t_s$  ลดเวลาขาขึ้น  $t_r$  ให้น้อยที่สุด ต้องมีอัตราลดทอน (Decay Ratio) หรือ  $c/a$  เป็น  $1/4$  ซึ่งได้แสดงไว้ดังรูปที่ ข.3

#### 2. พิจารณาจากค่าอินทิกรัลของความผิดพลาด

เป็นการตรวจสอบผลการตอบสนองของระบบการควบคุมตามเวลาทุกจุด ตั้งแต่กระบวนการเริ่มต้นเปลี่ยนแปลงไปเมื่อเวลา  $t = 0$  จนกระทั่งกระบวนการเข้าสู่ภาวะสมดุล เมื่อเวลา  $t \rightarrow \alpha$  ได้แก่

#### ก. อินทิกรัลของกำลังสองของความผิดพลาด (Integral Square Error (ISE))

$$ISE = \int_0^{\alpha} \varepsilon^2(t) dt = \sum_{\text{start}}^{\text{end}} (y - y')^2 \quad (\text{ข.6})$$

#### ข. อินทิกรัลของค่าสัมบูรณ์ของความผิดพลาด (Integral Absolute Error (IAE))

$$IAE = \int_0^{\alpha} |\varepsilon(t)| dt = \sum_{\text{start}}^{\text{end}} |y - y'| \quad (\text{ข.7})$$

#### ค. อินทิกรัลเวลาของค่าสัมบูรณ์ของความผิดพลาด (ITAE)

$$ITAE = \int_0^{\alpha} t\varepsilon(t) dt \quad (\text{ข.8})$$

โดยที่  $e(t) = sp(t) - c(t)$



การออกแบบตัวควบคุมที่ดีจะต้องมีค่าอินทิกรัลของความผิดพลาดน้อยที่สุด หลักเกณฑ์ในการตัดสินทั้งหมดนี้จะมีความเหมาะสมสำหรับระบบการควบคุมแต่ละระบบดังนี้

- ก. สำหรับค่าความผิดพลาดที่มีค่ามาก ใช้ ISE จะดีกว่า IAE เนื่องจากค่าความผิดพลาดจะถูกยกกำลังสอง ซึ่งจะทำให้ค่าของอินทิกรัลมีค่ามากตามไปด้วย
- ข. สำหรับความผิดพลาดที่มีค่าน้อย ใช้ IAE จะดีกว่า ISE เนื่องจากเมื่อทำการยกกำลังสอง ค่าความผิดพลาดที่มีค่าน้อยๆ จะทำให้ได้ค่าน้อยลงไปอีกทำให้เปรียบเทียบได้ยาก
- ค. สำหรับระบบที่มีช่วงการทดสอบที่ยาวนาน หลักเกณฑ์ ITEA จะเหมาะสำหรับการออกแบบระบบการควบคุม เนื่องจากค่าความผิดพลาดที่ปรากฏในช่วงที่  $t$  มีค่ามากจะถูกขยายให้มีค่ามากขึ้นไปด้วย ถึงแม้ว่าจะเป็นค่าความผิดพลาดที่น้อยในช่วงเวลาอินทิกรัล

## ภาคผนวก ค.

### ตัวอย่างโค้ดโปรแกรมที่สำคัญ

โปรแกรมเขียนแบบต่างๆ ที่ใช้ในงานวิจัยนี้ อันได้แก่ โปรแกรมซิμούเลชันของกระบวนการ โปรแกรมที่ใช้สร้างข้อมูลอินพุต-เอาต์พุต โปรแกรมที่ใช้สร้างแบบจำลองพีชคณิต และโปรแกรมที่ใช้สำหรับการควบคุมกระบวนการทั้งหมดนี้เขียนขึ้นโดยใช้โปรแกรมเมทแลบ (MATLAB) ทั้งสิ้น

โปรแกรมเมทแลบ (MATLAB) เป็นโปรแกรมที่ใช้ในการคำนวณทางคณิตศาสตร์ขั้นสูงและใช้ในการแก้ปัญหาที่เป็นเมทริกซ์ เป็นอาร์เรย์หลายมิติ การเขียนโปรแกรมด้วยภาษาเมทแลบ เป็นที่นิยมใช้กันมากในปัจจุบัน โดยเฉพาะอย่างยิ่ง ในงานด้านการแก้ปัญหา การคำนวณออกแบบ การสร้างแบบจำลองต่างๆ มากมายในทางวิศวกรรม เนื่องจากไม่ต้องมีการประกาศชื่อและชนิดของตัวแปรเหมือนภาษาระดับกลางและระดับสูงทั่วไป นอกจากนี้ยังเป็นภาษาที่เข้าใจและใช้ได้ง่าย

#### ค.1 ตัวอย่างโปรแกรมการเขียนแบบกระบวนการของเครื่องปฏิกรณ์ถังกวนแบบต่อเนื่อง

```
function [Tnew,Canew] = eu_seborg(T,Tc,Ca)

% operating condition
q=100; Caf=1; Tf=350; V=100; roh=1000; Cp=0.239; DH=5e4;
ER=8750; ko=7.2e10; UA=5e4;
% initial condition
Tc(1)=300; Ca(1)=0.5; T(1)=350;
DELTA=0.01;

% Calculation
CaDOT =q/V*(Caf-Ca)-ko*exp(-ER/T)*Ca;
TDOT =q/V*(Tf-T)+DH/(roh*Cp)*ko*exp(-ER/T)*Ca+...
UA/(V*roh*Cp)*(Tc-T);
Canew=Ca+CaDOT*DELTA;
Tnew=T+TDOT*DELTA;
```

#### ค.2 ตัวอย่างโปรแกรมของกระบวนการสร้างข้อมูลอินพุต-เอาต์พุต สำหรับกระบวนการของเครื่องปฏิกรณ์แบบถังกวนต่อเนื่อง

```
clear all
delta=0.01;
TIME(1)=0;
Tc(1)=300;
Ca(1)=0.5;
T(1)=350;
k=2;i=1;

input('Change Tc = ');
Tcm = 285:0.5:320;
finish=0;
```

```

n=1;

: ***** 1st Period *****
while finish<3
  if T(n)>390
    finish=30
  else
    end

  m=round(rand(1)*71);
  while or(m<1, m>=71)
    m=round(rand(1)*71);
  end

  if n>10
    if T(n)<350                                     * Lower Boundary
      m=round(rand(1)*71);
      while or(m<40, m>=45)
        m=round(rand(1)*45);
      end
    else or(abs(T(n)-T(n-2))>10 , T(n)>=360)       * Upper Boundary
      m=round(rand(1)*15);
      while or(m<1, m>=15)
        m=round(rand(1)*15);
      end
    end
  end

  for i = 1:20
    [Tnew,Canew] = eu_seborg(T(n),Tcm(m),Ca(n));
    n=n+1;
    T(n)=Tnew; Tc(n)=Tcm(m); Ca(n)=Canew; TIME(n)=TIME(n-1)+delta;
  end

  finish=TIME(n)
end

: ***** 2nd Period *****
while finish<6

  if T(n)>390
    finish=30
  else
    end

  m=round(rand(1)*71);
  while or(m<1, m>=71)
    m=round(rand(1)*71)
  end

  if n>10
    if T(n)<340                                     ***** LOWER BOUNDARY *****
      m=round(rand(1)*71);
      while or(m<20, m>=71)
        m=round(rand(1)*71);
      end
    else or(abs(T(n)-T(n-1))>15 , T(n)>=355)       ***** UPPER BOUNDARY *****
      m=round(rand(1)*15);
      while or(m<1, m>=15)
        m=round(rand(1)*15);
      end
    end
  end

  end
end

```

```

for i = 1:20
[Tnew,Canew] = eu_seborg(T(n),Tcm(m),Ca(n));
n=n+1;
T(n)=Tnew; Tc(n)=Tcm(m);Ca(n)=Canew;TIME(n)=TIME(n-1)+delta;
end
finish=TIME(n)
end

```

```

% ***** 3rd Building *****
while finish<9

```

```

    if T(n)>390
        finish=30
    else
    end

```

```

    m=round(rand(1)*71);
    while or(m<1, m>=71)
        m=round(rand(1)*71)
    end

```

```

    if n>10
        if T(n)<350 %*** Lower boundary
            m=round(rand(1)*71);
            while or(m<35, m>=60)
                m=round(rand(1)*71);
            end
        else or(abs(T(n)-T(n-1))>3 , T(n)>=353) % Upper Boundary
            m=round(rand(1)*10);
            while or(m<1, m>=10)
                m=round(rand(1)*10);
            end
        end
    end
end

```

```

for i = 1:20
[Tnew,Canew] = eu_seborg(T(n),Tcm(m),Ca(n));
n=n+1;
T(n)=Tnew; Tc(n)=Tcm(m);Ca(n)=Canew;TIME(n)=TIME(n-1)+delta;
end

```

```

    finish=TIME(n)
end

```

```

% ***** 4th Building *****
while finish<12

```

```

    if T(n)>390
        finish=30
    else
    end

```

```

    m=round(rand(1)*71);
    while or(m<1, m>=71)
        m=round(rand(1)*71)
    end

```

```

    if n>10
        if T(n)<340 %*** Lower boundary
            m=round(rand(1)*71);
            while or(m<35, m>=71)
                m=round(rand(1)*71);
            end
        end
    end
end

```

```

    end
    else or(abs(T(n)-T(n-1))>15 , T(n)>=355)      Upper Boundary
        m=round(rand(1)*15);
        while or(m<1, m>=15)
            m=round(rand(1)*15);
        end
    end
end

for i = 1:20
    [Tnew,Canew] = eu_seborg(T(n),Tcm(m),Ca(n));
    n=n+1;
    T(n)=Tnew; Tc(n)=Tcm(m);Ca(n)=Canew;TIME(n)=TIME(n-1)+delta;
end
finish=TIME(n)
end

% ***** 5th Period *****
while finish<15

    if T(n)>390
        finish=30
    else
        end

    m=round(rand(1)*71);
    while or(m<1, m>=71)
        m=round(rand(1)*71)
    end

    if n>10
        if T(n)<350      ***** Lower Boundary
            m=round(rand(1)*71);
            while or(m<35, m>=71)
                m=round(rand(1)*71);
            end
        else or(abs(T(n)-T(n-1))>1 , T(n)>=354)      Upper Boundary
            m=round(rand(1)*15);
            while or(m<1, m>=15)
                m=round(rand(1)*15);
            end
        end
    end
end

for i = 1:20
    [Tnew,Canew] = eu_seborg(T(n),Tcm(m),Ca(n));
    n=n+1;
    T(n)=Tnew; Tc(n)=Tcm(m);Ca(n)=Canew;TIME(n)=TIME(n-1)+delta;
end
finish=TIME(n)
end

% ***** 6th Period *****
while finish<18

    if T(n)>390
        finish=30
    else
        end

    m=round(rand(1)*71);
    while or(m<1, m>=71)

```

```

    m=round(rand(1)*71)
end

if n>10

    if T(n)<350                                     Lower Boundary
        m=round(rand(1)*71);
        while or(m<35, m>=71)
            m=round(rand(1)*71);
        end
    else or(abs(T(n)-T(n-1))>2 , T(n)>=360)         Upper Boundary
        m=round(rand(1)*15);
        while or(m<1, m >=15)
            m=round(rand(1)*15);
        end
    end

end

for i = 1:20
    [Tnew,Canew] = eu_seborg(T(n),Tcm(m),Ca(n));
    n=n+1;
    T(n)=Tnew; Tc(n)=Tcm(m);Ca(n)=Canew;TIME(n)=TIME(n-1)+delta;
end

finish=TIME(n)
end

figure(1)
subplot(211)
stairs(TIME,Tc)
title('room32.mat data for training')
axis([ 0 25 260 330])
xlabel('Time(min.)')
ylabel('Tc(coolant temp.)')
subplot(212)
plot(TIME,T)
axis([0 25 330 380])
xlabel('Time(min.)')
ylabel('Reactor Temp. (K)')

figure(2)
subplot(211)
plot(TIME,T)
xlabel('Time(min.)')
ylabel('Reactor Temp. (K)')
subplot(212)
plot(TIME,Ca)
axis([0 25 0 1.5])
xlabel('Time(sec)')
ylabel('Conc. A (K)')

```

ค.3 ตัวอย่างโปรแกรมการสร้างแบบจำลองความสัมพันธ์ฟัซซีจากข้อมูลอินพุท-เอาต์พุทของกระบวนการของเครื่องปฏิกรณ์แบบถังกวนต่อเนื่อง

```

clear all

% INITIAL PARAMETER
TIME(1)=0; delta=0.01;

% DATA for building Fuzzy model
load room32.mat
iodata=[no; T; Tc; Ca]';
struct=[2 1;
        2 2;
        2 3;
        3 1;
        2 0];

nvars= length(struct(:,1)); % Get the number of model variables
maxlag = max(struct(:,2)); % Get the maximum lag value
nsamp = length(iodata(:,1)); % Get the number of samples
tdata = zeros(nvars, nsamp-maxlag); % Create the tdata array

for i=1:nvars
    start=maxlag-struct(i,2)+1;
    stop=nsamp-struct(i,2);
    tdata(i,:)=iodata(start:stop,struct(i,1))';
end

%save c:\m_data\pracdata.mat tdata

Tcr = [-1000 280 290 ; 280 290 300 ;
        290 300 310 ; 300 310 320 ; 310 320 1000];

Tr = [-1000 320 330 ; 320 330 340 ;
        330 340 350 ; 340 350 360 ; 350 360 1000];

sets(:,:,1) = Tr;
sets(:,:,2) = Tr;
sets(:,:,3) = Tr;
sets(:,:,4) = Tcr;
sets(:,:,5) = Tr;

type=str2mat('trimf','trimf','trimf','trimf','trimf');

nsamp=length(tdata(1,:));
[nsets,not,nvars] = size(sets);

for i=1:nsamp

    poss(:,:,i)=zeros(nvars,nsets);

    for j=1:nvars
        poss(j,:,i)=evalmmf(tdata(j,i),sets(:,:,j),type);
    end

    prod=1;
    for j=1:(nvars-1)
        prod=kron(prod,poss(j,:,i));
    end
end

```

```

if exist('FSUM')==1
    FSUM=FSUM+prod;
else
    FSUM=prod;
end

if exist('RSUM')==1
    for j=1:nsets
        RSUM(j,:)=RSUM(j,:)+prod*poss(nvars,j,i);
    end
else
    for j=1:nsets
        RSUM(j,:)=prod*poss(nvars,j,i);
    end
end
end
end

```

```

if (exist('R')) & (exist('F'))
    for i=1:nsets
        R(i,:)=(R(i,:).*F + RSUM(i,:))./(F+FSUM);
    end
    F=F+FSUM;
else
    for i=1:nsets
        R(i,:)=RSUM(i,:)./FSUM;
    end
    F=FSUM;
end
end

```

`R(isnan(R))=zeros(size(R(isnan(R))))` This clears any NaN's in the R

### 3. PREDICTED OUTPUT

```

for ii=1:length(tdata)
    datpts(ii)=ii;
    fposs(:,1)=evalmmf(tdata(1,ii),sets(:, :,1),type); %intensity
    fposs(:,2)=evalmmf(tdata(2,ii),sets(:, :,2),type); %suzuki
    fposs(:,3)=evalmmf(tdata(3,ii),sets(:, :,3),type); %suzuki
    fposs(:,4)=evalmmf(tdata(4,ii),sets(:, :,4),type); %suzuki
    fprod=1; %multiplication combination

    for j=1:(nvars-1) %form possibility vectors
        fprod=kron(fprod,fposs(:,j)');
    end
    for j=1:nsets
        foposs(1,j)=sum((R(j,:).*fprod));
    end
    apex=sets(:,2,nvars);
    ym(ii)=sum((apex').*foposs(:, :))/sum(foposs(:, :));
end
clf
figure(1)
plot(datpts,tdata(3,:), 'x', datpts,ym)
Tset=tdata(3,:);
en=length(ym)
MaxErr=max(abs(Tset(1:en)-ym(1:en)))
IAE=sum(abs(Tset(1:en)-ym(1:en)))
ISE=sum(power(Tset(1:en)-ym(1:en),2))

```

save('tm\_resu','\est\train.mat','R','sets','tru



#### ค.4 ตัวอย่างโปรแกรมการควบคุมพีซีแบบฐานแบบจำลองของกระบวนการของเครื่องปฏิกรณ์แบบถังกวนต่อเนื่อง

```

clear all

global sets R struct nvars nsets

% load fuzzy model****
load room2r5.mat
% [nvars not nsets] = size(sets);
% maxlag = max(struct(:,2)); % Get the maximum lag value

% parameter
Kc=10000;
%Tcm=249:0.1:350;
Tc=300;
Ca=0.5;
T=350;

% Setpoint
Tset(1:500) = 345*ones(1,500);
Tset(501:1000)=350*ones(1,500);
Tset(1001:1500)=345*ones(1,500);
Tset(1501:2001) = 350*ones(1,501);

%TIME meter
TIME(1)=0 ;delta=0.01;en=20 ;
simtime=length(TIME(1):delta:en);

% Model-Based controller

Fibonacci number
p=1:20;
F(p)=1/sqrt(5)*(power(((1+sqrt(5))/2),...
(p+1))-power(((1-sqrt(5))/2),(p+1)));

Search parameter
b(1)=285;u(1)=310; %first and last element
K=15;m=1;
e=1/F(K)*(u(1)-b(1)); %accuracy
Kf=1;gain=10000; %filter gain and controller gain

%control parameter
T(1)=350;Tm(1)=350;ym(1)=350;err(1)=0;adj(1)=0;
Tc(1)=300;

for k=1: simtime
    if k<maxlag
        Tlag1=350;
        Tlag2=350;
    else
        Tlag1=T(k-1);
        Tlag2=T(k-2);
    end
    e(k)=T(k)-Tm(k);
    if k==1
        err(k) = 0*(1-Kf)+ Kf*e(k);
    else
        err(k) = err(k-1)*(1-Kf)+ Kf*e(k);
    end
    adj(k)=Tset(k)-err(k);
    yb=T(k);

```

```

search algorithm
m=1;
L(m,1)=F(K-2)/F(K)*(u(m)-b(m))+b(m);
L(m,2)=F(K-1)/F(K)*(u(m)-b(m))+b(m);
A=[power((L(m,1)-Tc(k)),2)/gain,power((L(m,2)-Tc(k)),2)/gain];
  indat1=[yb,Tlag1,Tlag2,L(m,1)];
  indat2=[yb,Tlag1,Tlag2,L(m,2)];
  y1=[cdict23(indat1),cdict23(indat2)];
  yldat=[y1(1)+err(k),y1(2)+err(k)];

  M(m,:)= [A(1)+power((yldat(1)-Tset(k)),2),...
           A(2)+power((yldat(2)-Tset(k)),2)];

for m = 2:K-1
  if M(m-1,1)>M(m-1,2)
    b(m)=L(m-1,1);
    L(m,1)=L(m-1,2);
    u(m)=u(m-1);
    if m<K-1
      L(m,2)=F(K-m)/F(K+1-m)*(u(m)-b(m))+b(m);
      A=[power((L(m,1)-Tc(k)),2)/gain,power((L(m,2)-Tc(k)),2)/gain];
        indat1=[yb,Tlag1,Tlag2,L(m,1)];
        indat2=[yb,Tlag1,Tlag2,L(m,2)];
        y1=[cdict23(indat1),cdict23(indat2)];
        yldat=[y1(1)+err(k),y1(2)+err(k)];

      M(m,:)= [A(1)+power((yldat(1)-Tset(k)),2),...
               A(2)+power((yldat(2)-Tset(k)),2)];
    else
      L(m,2)=(u(m)+b(m))/2;
      A=[power((L(m,1)-Tc(k)),2)/gain,power((L(m,2)-Tc(k)),2)/gain];
        indat1=[yb,Tlag1,Tlag2,L(m,1)];
        indat2=[yb,Tlag1,Tlag2,L(m,2)];
        y1=[cdict23(indat1),cdict23(indat2)];
        yldat=[y1(1)+err(k),y1(2)+err(k)];

      win=[A(1)+power((yldat(1)-Tset(k)),2),...
           A(2)+power((yldat(2)-Tset(k)),2)];
    end
  elseif M(m-1,1)<=M(m-1,2)
    u(m)=L(m-1,2);
    L(m,2)=L(m-1,1);
    b(m)=b(m-1);
    if m<K-1
      L(m,1)=F(K-1-m)/F(K+1-m)*(u(m)-b(m))+b(m);
      A=[power((L(m,1)-Tc(k)),2)/gain,...
         power((L(m,2)-Tc(k)),2)/gain];
        indat1=[yb,Tlag1,Tlag2,L(m,1)];
        indat2=[yb,Tlag1,Tlag2,L(m,2)];
        y1=[cdict23(indat1),cdict23(indat2)];
        yldat=[y1(1)+err(k),y1(2)+err(k)];

      M(m,:)= [A(1)+power((yldat(1)-Tset(k)),2),...
               A(2)+power((yldat(2)-Tset(k)),2)];
    else
      L(m,1)=(u(m)+b(m))/2;
      A=[power((L(m,1)-Tc(k)),2)/gain,...
         power((L(m,2)-Tc(k)),2)/gain];
        indat1=[yb,Tlag1,Tlag2,L(m,1)];
        indat2=[yb,Tlag1,Tlag2,L(m,2)];
        y1=[cdict23(indat1),cdict23(indat2)];
    end
  end
end

```

```

        yldat=[y1(1)+err(k),y1(2)+err(k)];

        win=[A(1)+power((yldat(1)-Tset(k)),2),...
            A(2)+power((yldat(2)-Tset(k)),2)];

    end
end
end

M
[b' L(:,1)L(:,2) u']
clear M win

Tc(k+1)=L(m,1);
indat=[T(k),Tlag1,Tlag2,Tc(k+1)];
Tm(k+1)=cdict23(indat);
[Tnew,Canew] = eu_seborg(T(k),Tc(k+1),Ca(k));
T(k+1)= Tnew;
Ca(k+1)= Canew;
TIME(k+1)=TIME(k)+delta;
k
Tnew
end

% Display Result
figure(1)
subplot(211)
plot(TIME,T,TIME,Tset,'r:')
title('CSTR FIMC Control dt=0.01')
ylabel('Reactor Temp. (K)')
axis([0 20 335 355])
subplot(212),stairs(TIME(1:simtime),Tc(1:simtime))
ylabel('Coolant Temp. (K)')
axis([0 20 260 350])

figure(2)
plot(TIME,T,TIME,Tset,'r:')
ylabel('T (K)')
xlabel('Time (min.)')

figure(3)
stairs(TIME,Tc)
ylabel('Tc (K)')
xlabel('Time (min.)')

figure(4)
plot(TIME,Ca)
ylabel('Ca (Molar)')
xlabel('Time (min.)')

MaxErr=max(abs(Tset(2:2001)-T(2:2001)))
IAE=sum(abs(Tset(2:2001)-T(2:2001)))
ISE=sum(power(Tset(2:2001)-T(2:2001),2))

```

## ประวัติผู้เขียน

นางสาวทัศนีย์ วัฒนเขาว์พิสุทธิ์ เกิดเมื่อวันที่ 8 กันยายน พ.ศ.2516 ที่อำเภอเบตง จังหวัดยะลา สำเร็จการศึกษาในระดับ ปวช. สาขาเคมีปฏิบัติการ คณะเทคนิคเคมี จากสถาบันเทคโนโลยีราชมงคล วิทยาเขตเทคนิค กรุงเทพฯ เมื่อปี พ.ศ.2533 จากนั้นศึกษาต่อในหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมเคมี ในคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า พระนครเหนือ ระหว่างศึกษาในระดับปริญญาตรีได้มีโอกาสฝึกงานภาคฤดูร้อน ในแผนกวิศวกรรม ของบริษัท เชลล์ ประเทศไทย จำกัด (Thai Shell Exploration and Production) เมื่อจบการศึกษาระดับปริญญาตรี ได้มีโอกาสรับทุนแลกเปลี่ยนนักศึกษาฝึกงาน (IEASTE) ไปฝึกงานในแผนก Economic Technology ของโรงกลั่นน้ำมัน CMV Refinery ณ กรุงเวียนนา ประเทศออสเตรีย จากนั้นศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมเคมี เมื่อปี พ.ศ. 2538

