การออกแบบตัวควบคุมพีไอดีชนิดปรับค่าเองได้ โดยใช้เจเนติกอัลกอริธึม ประยุกต์ในกระบวนการปรับค่าพีเอชให้เป็นกลาง

นาย อนุชา ทิพยวัลย์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมเคมี ภาควิชาวิศวกรรมเคมี

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2545

ISBN 974-xxx-xxx-xxx

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

DESIGN OF SELF-TUNING PID CONTROLLER USING
GENETIC ALGORITHMS APPLIED TO PH NEUTRALIZATION PROCESS

Mr. Anucha Thippayawarn

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Chemical Engineering
Department of Chemical Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2002
ISBN 974-XXX-XXX-X

หัวข้อวิทยานิพนธ์	การออกแบบตัวควบคุมพีไอดีชนิดปรับค่าเองได้ โดยใช้เจเนติกอัลกอริธึม ประยุกต์ในกระบวนการปรับค่าพีเอชให้เป็นกลาง
โดย	นาย อนุชา ทิพยวัลย์
สาขาวิชา	วิศวกรรมเคมี
อาจารย์ที่ปรึกษา	อาจารย์ ดร. มนตรี วงศ์ศรี

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

.... คณบดีคณะวิศวกรรมศาสตร์

(ศาสตราจารย์ ดร.สมศักดิ์ ปัญญาแก้ว)

คณะกรรมการสอบวิทยานิพนธ์

.... ประธานกรรมการ

(ศาสตราจารย์ ดร. ปิยะสาร ประเสริฐธรรม)

.... อาจารย์ที่ปรึกษา

(อาจารย์ ดร. มนตรี วงศ์ศรี)

.... กรรมการ

(รองศาสตราจารย์ ดร. ไพศาล กิตติศุภกร)

.... กรรมการ

(อาจารย์ ดร. สุพจน์ พัฒนะศรี)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

อนุชา ทิพยวัลย์ : การออกแบบตัวควบคุมพีไอดีชนิดปรับค่าเองได้ โดยใช้เจเนติกอัลกอริธึมประยุกต์ในกระบวนการปรับค่าพีเอชให้เป็นกลาง (DESIGN OF SELF-TUNING PID CONTROLLER USING GENETIC ALGORITHMS APPLIED TO PH NEUTRALIZATION PROCESS) อ. ที่ปรึกษา : ดร.มนตรี วงศ์ศรี; จำนวนหน้า 117 หน้า. ISBN 974-172-440-3.

งานวิจัยชิ้นนี้ได้มีการนำเจเนติกอัลกอริธึม ซึ่งเป็นวิธีหนึ่งในการหาค่าเหมาะสมที่สุด มาประยุกต์ใช้ในการหาค่าพารามิเตอร์สำหรับตัวควบคุมพีไอดีชนิดปรับค่าเองได้ โดยที่ได้มีการนำเสนอการตรวจสอบความถูกต้องของอัลกอริธึม ด้วยการหาค่าที่เหมาะสมที่สุดของฟังก์ชันทดสอบ คือ ฟังก์ชันโรเซ็นบรอกค์ ซึ่งได้ผลที่มีความถูกต้องเป็นที่น่าพอใจ หลังจากนั้นจึงได้ทดสอบตัวควบคุมที่นำเสนอกับกระบวนการที่สนใจ ซึ่งพบว่าการใช้เจเนติกอัลกอริธึมในการหาค่าพารามิเตอร์ของตัวควบคุมพีไอดีนั้น สามารถปรับค่าพารามิเตอร์ได้อย่างเหมาะสม ดังจะเห็นได้จากตัวควบคุมพีไอดีที่นำเสนอนี้สามารถควบคุมกระบวนการได้ดีการตัวควบคุมพีไอดีที่มีการปรับค่าพารามิเตอร์ตามวิธีของซีกเลอร์-นิโคลส์ โดยที่การทดสอบตัวควบคุมที่นำเสนอนี้ ได้ทำการทดสอบเปรียบเทียบกับตัวควบคุมพีไอดีธรรมดา บนกระบวนการ 2 กระบวนการคือ กระบวนการอันดับสองที่มีเวลาหน่วง และ กระบวนการปรับค่าพีเอชให้เป็นกลาง ที่เกิดขึ้นภายในเครื่องปฏิกรณ์แบบถังกวนต่อเนื่อง กรณีที่ทำการศึกษา เป็นการกำจัดผลของตัวแปรรบกวน คือ อัตราการไหล และความเข้มข้นของสายกรดขาเข้าสู่เครื่องปฏิกรณ์ การหาค่าที่เหมาะสมที่สุดโดยการใช้เจเนติกอัลกอริธึม สามารถปรับค่าพารามิเตอร์ของตัวควบคุมพีไอดีชนิดปรับค่าเองได้ตามที่นำเสนอนี้ ได้อย่างเหมาะสม

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา......วิศวกรรมเคมี.......................ลายมือชื่อนิสิต...
สาขาวิชา......วิศวกรรมเคมี.....................ลายมือชื่ออาจารย์ที่ปรึกษา.......................................
ปีการศึกษา 2545 ลายมือชื่ออาจารย์ทีปรึกษาร่วม................................

437 06033 21 : MAJOR CHEMICAL ENGINEERING

KEY WORD: SELF-TUNING PID CONTROLLER / GENETIC ALGORITHMS / pH NEUTRALIZATION PROCESS

ANUCHA THIPPAYAWARN : DESIGN OF SELF-TUNING PID CONTROLLER USING GENETIC ALGORITHMS APPLIED TO PH NEUTRALIZATION PROCESS. THESIS ADVISOR : DR. MONTREE WONGSRI, 117 pp. ISBN 974-172-440-3.

Genetic Algorithms, one of the optimization techniques, is applied in the self-tuning PID controller in this thesis. Testing the program with the test case optimization problem, Rosenbrock's function, is carried out to validate the algorithms. Then the proposed controller is tested with the process that is relevance to the thesis. The results show that genetic algorithms can be applied to self-tuning PID controller because the proposed self-tuning PID controller gives better control performances over the conventional PID controller which is tuned conform to the Ziegler-Nichols method. The processes, second order process plus dead time and pH neutralization process that takes place in the continuous stirred tank reactor are used to test performance of the proposed controller. The controller are evaluated in the face of disturbances namely, inlet acid flow rate and inlet acid concentration. The genetic algorithms optimization part adjusts the parameters of the self-tuning controller to the proper values.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Department ..CHEMICAL ENGINEERING.. Student's signature...

Field of study ..CHEMICAL ENGINEERING.. Advisor's signature…....................................

Academic year …...2002......... Co-advisor's signature.................................

กิตติกรรมประกาศ

ข้าพเจ้าขอขอบพระคุณ อาจารย์ ดร. มนตรี วงศ์ศรี อาจารย์ที่ปรึกษาวิทยานิพนธ์ และรองศาสตราจารย์ ดร. ไพศาล กิตติศุภกร อาจารย์กรรมการสอบวิทยานิพนธ์ ผู้ซึ่งให้คำปรึกษาและคำแนะนำในการทำงานวิจัย ตลอดจนการตรวจทานและแก้ไขวิทยานิพนธ์ จนกระทั่งสำเร็จลุล่วงอย่างสมบูรณ์

ข้าพเจ้าขอขอบพระคุณ ศาสตราจารย์ ดร. ปิยะสาร ประเสริฐธรรม ประธานกรรมการ และ อาจารย์ ดร. สุพจน์ พัฒนะศรี กรรมการในการสอบวิทยานิพนธ์ ซึ่งได้สละเวลาในการให้คำแนะนำ และความคิดเห็นเพิ่มเติมที่เป็นประโยชน์ต่องานวิจัยนี้

ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และผู้มีอุปการคุณทุกท่านที่ไม่ได้เอ่ยนาม ณ ที่นี้ ที่ได้ทำให้ข้าพเจ้าก้าวมาจนถึงทุกวันนี้ ได้อย่างดี

ขอขอบคุณ พี่ๆ เพื่อนๆ และ น้องๆ ที่ได้เป็นกำลังใจให้กันตลอด

สุดท้ายนี้ ขอขอบพระคุณ จุฬาลงกรณ์มหาวิทยาลัย ที่ได้ปลูกฝังความเป็นบัณฑิตที่สมบูรณ์ให้แก่ข้าพเจ้า ตั้งแต่ระดับปริญญาตรีเป็นต้นมา

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย .... ง

บทคัดย่อภาษาอังกฤษ........ จ

กิตติกรรมประกาศ........ ฉ

สารบัญ .... ช

สารบัญภาพ .... ฌ

สารบัญตาราง........ ฒ

คำอธิบายสัญลักษณ์ .... ณ

บทที่ 1 บทนำ

1.1 มูลเหตุจูงใจและที่มาของงานวิจัย........ 1

1.2 วัตถุประสงค์ของงานวิจัย........ 3

1.3 ขอบเขตของงานวิจัย 3

1.4 ขั้นตอนการดำเนินงานวิจัย 4

1.5 ประโยชน์ที่คาดว่าจะได้รับ 4

1.6 ระยะเวลาในการดำเนินงาน 5

บทที่ 2 ผลงานวิจัยที่เกี่ยวข้อง

2.1 งานวิจัยที่เกี่ยวข้องกับตัวควบคุมพีไอดีชนิดต่างๆ........ 6

2.2 งานวิจัยที่เกี่ยวข้องกับตัวควบคุมชนิดปรับค่าเองได้........ 7

2.3 งานวิจัยที่เกี่ยวข้องกับกระบวนการปรับค่าพีเอช 8

2.4 งานวิจัยที่เกี่ยวข้องกับการประยุกต์ใช้เจเนติกอัลกอริธึมในการควบคุมกระบวนการ..... 9

บทที่ 3 พื้นฐานเจเนติกอัลกอริธึม........ 11

บทที่ 4 ระบบควบคุมชนิดปรับค่าเองได้

4.1 บทนำ 18

4.2 การหาเอกลักษณ์ของกระบวนการ........ 21

4.3 การออกแบบตัวควบคุม........ 24

สารบัญ (ต่อ)

บทที่ หน้า

4.4 การนำไปใช้ในกระบวนการ 25
4.5 อัลกอริธึมของตัวควบคุมพีไอดีชนิดปรับค่าเองได้ที่ใช้ในงานวิจัยนี้ 26

บทที่ 5 กระบวนการปรับค่าพีเอชให้เป็นกลาง
5.1 ภาพรวม 29
5.2 เส้นโค้งของการไตเตรท 32

บทที่ 6 การจำลองการควบคุมกระบวนการ
6.1 การจุดที่ทำให้ฟังก์ชันโรเซ็นบรอกค์ต่ำที่สุด 34
6.2 การทดสอบการหาค่าพารามิเตอร์ของกระบวนการที่ทราบค่าแล้ว 39
6.3 การจำลองการควบคุมบนกระบวนการอันดับสองที่มีค่าพารามิเตอร์เปลี่ยนแปลง 41
6.4 การจำลองการควบคุมกระบวนการที่มีการตอบสนองอันดับสองที่มีเวลาหน่วง 52
6.5 การจำลองการควบคุมกระบวนการที่มีความไม่เป็นเชิงเส้น 66

บทที่ 7 สรุปผลการจำลองการควบคุม 89

รายการอ้างอิง 92

ภาคผนวก
ภาคผนวก ก 94
ภาคผนวก ข 111
ภาคผนวก ค 113

ประวัติผู้เขียนวิทยานิพนธ์ 117

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

หน้า

รูปที่ 3.1 รหัสที่ใช้ในเจเนติกอัลกอริธึม แทนตัวแปรที่มีค่าเท่ากับ 53 12

รูปที่ 3.2 แผนภูมิแสดงกระบวนวิธีในการหาคำตอบที่เหมาะสมที่สุดด้วยการใช้เจเนติกอัลกอริธึม 17

รูปที่ 4.1 การควบคุมแบบป้อนกลับทั่วไป (Typical Feedback Control) 19

รูปที่ 4.2 การควบคุมแบบปรับตัวได้ (Adaptive Control) 19

รูปที่ 4.3 แผนภูมิแสดงการควบคุมกระบวนการปรับค่าพีเอชให้เป็นกลางด้วยตัวควบคุมพีไอดีชนิดปรับค่าเองได้ ที่ใช้เจเนติกอัลกอริธึมในการหาค่าที่เหมาะสมที่สุด 28

รูปที่ 5.1 เครื่องปฏิกรณ์แบบถังกวนต่อเนื่อง (CSTR) 29

รูปที่ 5.2 เส้นกราฟของการไตเตรท 32

รูปที่ 6.1 กราฟความสัมพันธ์ของโรเซ็นบรอกค์ 35

รูปที่ 6.2 การกระจายตัวของประชากรคำตอบที่สุ่มเลือกบนช่วง ([0,2],[0,2]) ในรอบแรก 36

รูปที่ 6.3 กราฟขยายแสดงเส้นความสูงรอบๆ จุดคำตอบ (1,1) 36

รูปที่ 6.4 การกระจายตัวของประชากรคำตอบ ณ รอบการคำนวณสุดท้าย 37

รูปที่ 6.5 ค่าความแข็งแรงของประชากรที่มีความแข็งแรงที่สุด ในแต่ละรอบการคำนวณ 37

รูปที่ 6.6 ค่าเฉลี่ยความแข็งแรงของประชากร ในแต่ละรอบการคำนวณ 38

รูปที่ 6.7 ตอบสนองวงเปิดของสมการอันดับสอง (6.4) เมื่อมีการเปลี่ยนแปลงตัวแปรอินพุทและการประมาณด้วยสมการอันดับหนึ่งที่มีเวลาหน่วง 40

รูปที่ 6.8 การตอบสนองลูพเปิดของกระบวนการดังสมการที่ (6.5) 42

รูปที่ 6.9 การเปลี่ยนแปลงแบบขั้น (Step Input) ของตัวแปรอินพุทเพื่อใช้ในการหาการตอบสนองของกระบวนการอันดับสองดังสมการที่ (6.5) 43

รูปที่ 6.10 การตอบสนองลูพเปิดเมื่อกระบวนการมีค่าเกนของกระบวนการที่ต่างๆ กัน 44

รูปที่ 6.11 การตอบสนองลูพเปิดเมื่อกระบวนการมีค่าเวลาคงที่ของกระบวนการที่ต่างๆ กัน 45

รูปที่ 6.12 ผลการตอบสนองลูพปิดภายใต้ตัวควบคุมพีไอดีธรรมดาที่ (1) ค่าพารามิเตอร์คงที่ (2) ค่าพารามิเตอร์เปลี่ยนแปลง 46

รูปที่ 6.13 ผลการตอบสนองลูพปิดภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ ที่มีการประยุกต์ใช้เจเนติกอัลกอริธึม 46

รูปที่ 6.14 ผลการตอบสนองลูพปิดภายใต้ตัวควบคุมพีไอดีธรรมดา (1) ค่าพารามิเตอร์คงที่ (2) ค่าพารามิเตอร์เปลี่ยนแปลง 48

สารบัญภาพ (ต่อ)

หน้า

รูปที่ 6.15 ผลการตอบสนองลูพปิดภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ ที่มีการประยุกต์ใช้เจเนติกอัลกอริธึม 48
รูปที่ 6.16 ผลการตอบสนองลูพปิดภายใต้ตัวควบคุมพีไอดีธรรมดาที่ (1) ค่าพารามิเตอร์คงที่ (2) ค่าพารามิเตอร์เปลี่ยนแปลง 50
รูปที่ 6.17 ผลการตอบสนองลูพปิดภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ ที่มีการประยุกต์ใช้เจเนติกอัลกอริธึม 50
รูปที่ 6.18 กระบวนการควบคุมความเข้มข้นของสาร A ที่ออกจากเครื่องปฏิกรณ์แบบถังกวนต่อเนื่อง (Continuous Stirred Tank Reactor, CSTR) 52
รูปที่ 6.19 การตอบสนองลูพเปิดของกระบวนการในหัวข้อ 6.4.1 54
รูปที่ 6.20 ตัวแปรอินพุทที่เปลี่ยนแปลงแบบขั้น เพื่อหาผลการตอบสนองลูพเปิดของระบบ 55
รูปที่ 6.21 การตอบสนองของความเข้มข้นของสาร A ที่ออกจากถังที่ 2 ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่ออัตราการไหลของตัวทำละลาย (F) ลดลง 15% 56
รูปที่ 6.22 อัตราการป้อนสาร A (ตัวแปรปรับ) ณ สายขาเข้าของถังที่ 1 ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่ออัตราการไหลของตัวทำละลาย (F) ลดลง 15% 56
รูปที่ 6.23 การตอบสนองของความเข้มข้นของสาร A ที่ออกจากถังที่ 2 ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่ออัตราการไหลของตัวทำละลาย (F) ลดลง 15% 57
รูปที่ 6.24 อัตราการป้อนสาร A (ตัวแปรปรับ) ณ สายขาเข้าของถังที่ 1 ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่ออัตราการไหลของตัวทำละลาย (F) ลดลง 15% 57
รูปที่ 6.25 การตอบสนองของความเข้มข้นของสาร A ที่ออกจากถังที่ 2 ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่ออัตราการไหลของตัวทำละลาย (F) ลดลง 30% 58
รูปที่ 6.26 อัตราการป้อนสาร A (ตัวแปรปรับ) ณ สายขาเข้าของถังที่ 1 ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่ออัตราการไหลของตัวทำละลาย (F) ลดลง 30% 59
รูปที่ 6.27 การตอบสนองของความเข้มข้นของสาร A ที่ออกจากถังที่ 2 ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่ออัตราการไหลของตัวทำละลาย (F) ลดลง 30% 59
รูปที่ 6.28 อัตราการป้อนสาร A (ตัวแปรปรับ) ณ สายขาเข้าของถังที่ 1 ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่ออัตราการไหลของตัวทำละลาย (F) ลดลง 30% 60
รูปที่ 6.29 การตอบสนองของความเข้มข้นของสาร A ที่ออกจากถังที่ 2 ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่ออัตราการไหลของตัวทำละลาย (F) เพิ่มขึ้น 15% 60

สารบัญภาพ (ต่อ)

หน้า

รูปที่ 6.30 อัตราการป้อนสาร A (ตัวแปรปรับ) ณ สายขาเข้าของถังที่ 1 ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่ออัตราการไหลของตัวทำละลาย (F) เพิ่มขึ้น 15% 61
รูปที่ 6.31 การตอบสนองของความเข้มข้นของสาร A ที่ออกจากถังที่ 2 ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่ออัตราการไหลของตัวทำละลาย (F) เพิ่มขึ้น 15% 61
รูปที่ 6.32 อัตราการป้อนสาร A (ตัวแปรปรับ) ณ สายขาเข้าของถังที่ 1 ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่ออัตราการไหลของตัวทำละลาย (F) เพิ่มขึ้น 15% 62
รูปที่ 6.33 การตอบสนองของความเข้มข้นของสาร A ที่ออกจากถังที่ 2 ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่ออัตราการไหลของตัวทำละลาย (F) เพิ่มขึ้น 30% 62
รูปที่ 6.34 อัตราการป้อนสาร A (ตัวแปรปรับ) ณ สายขาเข้าของถังที่ 1 ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่ออัตราการไหลของตัวทำละลาย (F) เพิ่มขึ้น 30% 63
รูปที่ 6.35 การตอบสนองของความเข้มข้นของสาร A ที่ออกจากถังที่ 2 ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่ออัตราการไหลของตัวทำละลาย (F) เพิ่มขึ้น 30% 63
รูปที่ 6.36 อัตราการป้อนสาร A (ตัวแปรปรับ) ณ สายขาเข้าของถังที่ 1 ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่ออัตราการไหลของตัวทำละลาย (F) เพิ่มขึ้น 30% 64
รูปที่ 6.37 การตอบสนองของค่าพีเอช เมื่อมีการเปลี่ยนแปลงตัวแปรอินพุท ดังรูปที่ 6.38 67
รูปที่ 6.38 การเปลี่ยนแปลงอัตราการไหลของสายด่างแบบขั้น เพื่อสังเกตการตอบสนองวงเปิดของค่าพีเอช (ตัวแปรควบคุม) 67
รูปที่ 6.39 การตอบสนองของค่าพีเอช (ตัวแปรควบคุม) ที่ออกจากเครื่องปฏิกรณ์ ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่ออัตราการไหลของกรด (F1) เพิ่มขึ้น 15% 68
รูปที่ 6.40 อัตราการป้อนของสายด่าง (ตัวแปรปรับ) เพื่อควบคุมค่าพีเอช ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่ออัตราการไหลของสายกรด (F1) เพิ่มขึ้น 15% 69
รูปที่ 6.41 การตอบสนองของค่าพีเอช (ตัวแปรควบคุม) ที่ออกจากเครื่องปฏิกรณ์ ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่ออัตราการไหลของกรด (F1) เพิ่มขึ้น 15% 69
รูปที่ 6.42 อัตราการป้อนของสายด่าง (ตัวแปรปรับ) เพื่อควบคุมค่าพีเอช ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่ออัตราการไหลของสายกรด (F1) เพิ่มขึ้น 15% 70
รูปที่ 6.43 การตอบสนองของค่าพีเอช (ตัวแปรควบคุม) ที่ออกจากเครื่องปฏิกรณ์ ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่ออัตราการไหลของกรด (F1) ลดลง 15% 71

สารบัญภาพ (ต่อ)

หน้า

รูปที่ 6.44 อัตราการป้อนของสายด่าง (ตัวแปรปรับ) เพื่อควบคุมค่าพีเอช ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่ออัตราการไหลของสายกรด (F1) ลดลง 15% 71
รูปที่ 6.45 การตอบสนองของค่าพีเอช (ตัวแปรควบคุม) ที่ออกจากเครื่องปฏิกรณ์ ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่ออัตราการไหลของกรด (F1) ลดลง 15%........... 72
รูปที่ 6.46 อัตราการป้อนของสายด่าง (ตัวแปรปรับ) เพื่อควบคุมค่าพีเอช ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่ออัตราการไหลของสายกรด (F1) ลดลง 15% 72
รูปที่ 6.47 การตอบสนองของค่าพีเอช (ตัวแปรควบคุม) ที่ออกจากเครื่องปฏิกรณ์ ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่ออัตราการไหลของกรด (F1) เพิ่มขึ้น 30% 73
รูปที่ 6.48 อัตราการป้อนของสายด่าง (ตัวแปรปรับ) เพื่อควบคุมค่าพีเอช ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่ออัตราการไหลของสายกรด (F1) เพิ่มขึ้น 30% 74
รูปที่ 6.49 การตอบสนองของค่าพีเอช (ตัวแปรควบคุม) ที่ออกจากเครื่องปฏิกรณ์ ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่ออัตราการไหลของกรด (F1) เพิ่มขึ้น 30%.......... 74
รูปที่ 6.50 อัตราการป้อนของสายด่าง (ตัวแปรปรับ) เพื่อควบคุมค่าพีเอช ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่ออัตราการไหลของสายกรด (F1) เพิ่มขึ้น 30% 75
รูปที่ 6.51 การตอบสนองของค่าพีเอช (ตัวแปรควบคุม) ที่ออกจากเครื่องปฏิกรณ์ ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่ออัตราการไหลของกรด (F1) ลดลง 30% 75
รูปที่ 6.52 อัตราการป้อนของสายด่าง (ตัวแปรปรับ) เพื่อควบคุมค่าพีเอช ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่ออัตราการไหลของสายกรด (F1) ลดลง 30% 76
รูปที่ 6.53 การตอบสนองของค่าพีเอช (ตัวแปรควบคุม) ที่ออกจากเครื่องปฏิกรณ์ ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่ออัตราการไหลของกรด (F1) ลดลง 30%........... 76
รูปที่ 6.54 อัตราการป้อนของสายด่าง (ตัวแปรปรับ) เพื่อควบคุมค่าพีเอช ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่ออัตราการไหลของสายกรด (F1) ลดลง 30% 77
รูปที่ 6.55 การตอบสนองของค่าพีเอช (ตัวแปรควบคุม) ที่ออกจากเครื่องปฏิกรณ์ ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่อความเข้มข้นของสารละลายกรด (C1) เพิ่มขึ้น 15%........ 78
รูปที่ 6.56 อัตราการป้อนของสายด่าง (ตัวแปรปรับ) เพื่อควบคุมค่าพีเอช ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่อความเข้มข้นของสารละลายกรด (C1) เพิ่มขึ้น 15% 78

หน้า

รูปที่ 6.57 การตอบสนองของค่าพีเอช (ตัวแปรควบคุม) ที่ออกจากเครื่องปฏิกรณ์ ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่อความเข้มข้นของสารละลายกรด (C1) เพิ่มขึ้น 15% 79
รูปที่ 6.58 อัตราการป้อนของสายด่าง (ตัวแปรปรับ) เพื่อควบคุมค่าพีเอช ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่อความเข้มข้นของสารละลายกรด (C1) เพิ่มขึ้น 15% 79
รูปที่ 6.59 การตอบสนองของค่าพีเอช (ตัวแปรควบคุม) ที่ออกจากเครื่องปฏิกรณ์ ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่อความเข้มข้นของสารละลายกรด (C1) ลดลง 15%........ 80
รูปที่ 6.60 อัตราการป้อนของสายด่าง (ตัวแปรปรับ) เพื่อควบคุมค่าพีเอช ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่อความเข้มข้นของสารละลายกรด (C1) ลดลง 15% 80
รูปที่ 6.61 การตอบสนองของค่าพีเอช (ตัวแปรควบคุม) ที่ออกจากเครื่องปฏิกรณ์ ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่อความเข้มข้นของสารละลายกรด (C1) ลดลง 15% 81
รูปที่ 6.62 อัตราการป้อนของสายด่าง (ตัวแปรปรับ) เพื่อควบคุมค่าพีเอช ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่อความเข้มข้นของสารละลายกรด (C1) ลดลง 15% 81
รูปที่ 6.63 การตอบสนองของค่าพีเอช (ตัวแปรควบคุม) ที่ออกจากเครื่องปฏิกรณ์ ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่อความเข้มข้นของสารละลายกรด (C1) เพิ่มขึ้น 30%........ 82
รูปที่ 6.64 อัตราการป้อนของสายด่าง (ตัวแปรปรับ) เพื่อควบคุมค่าพีเอช ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่อความเข้มข้นของสารละลายกรด (C1) เพิ่มขึ้น 30% 82
รูปที่ 6.65 การตอบสนองของค่าพีเอช (ตัวแปรควบคุม) ที่ออกจากเครื่องปฏิกรณ์ ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่อความเข้มข้นของสารละลายกรด (C1) เพิ่มขึ้น 30% 83
รูปที่ 6.66 อัตราการป้อนของสายด่าง (ตัวแปรปรับ) เพื่อควบคุมค่าพีเอช ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่อความเข้มข้นของสารละลายกรด (C1) เพิ่มขึ้น 30% 83
รูปที่ 6.67 การตอบสนองของค่าพีเอช (ตัวแปรควบคุม) ที่ออกจากเครื่องปฏิกรณ์ ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่อความเข้มข้นของสารละลายกรด (C1) ลดลง 30%........ 84
รูปที่ 6.68 อัตราการป้อนของสายด่าง (ตัวแปรปรับ) เพื่อควบคุมค่าพีเอช ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่อความเข้มข้นของสารละลายกรด (C1) ลดลง 30% 85

สารบัญภาพ (ต่อ)

หน้า

รูปที่ 6.69 การตอบสนองของค่าพีเอช (ตัวแปรควบคุม) ที่ออกจากเครื่องปฏิกรณ์ ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่อความเข้มข้นของสารละลายกรด (C1) ลดลง 30% 85
รูปที่ 6.70 อัตราการป้อนของสายด่าง (ตัวแปรปรับ) เพื่อควบคุมค่าพีเอช ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่อความเข้มข้นของสารละลายกรด (C1) ลดลง 30% 86
รูปที่ ค.1 การตอบสนองของเอาท์พุทของกระบวนการ เมื่อมีอินพุท ดังรูป ค.2...................... 113
รูปที่ ค.2 การเปลี่ยนแปลงแบบขั้นของอินพุท ที่เข้าสู่กระบวนการ....................................... 114
รูปที่ ค.3 วิธีการหาค่าประมาณกระบวนการอันดับหนึ่งและมีเวลาหน่วงแบบกราฟิก 114

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

หน้า

ตารางที่ 1.1 ระยะเวลาในการดำเนินงานวิจัย 5
ตารางที่ 3.1 ประชากรเริ่มต้นและค่าความแข็งแรง 13
ตารางที่ 3.2 ประชากรที่ถูกเลือกขึ้นมาเพื่อทำการครอสโอเวอร์ 14
ตารางที่ 3.3 ประชากรหลังการครอสโอเวอร์ 15
ตารางที่ 3.4 ประชากรหลังจากการกลายพันธุ์. 16
ตารางที่ 4.1 กฎการปรับค่าตัวควบคุมพีไอดีตามหลักของ Ziegler-Nichols. 25
ตารางที่ 5.1 แสดงตัวแปรต่างๆ ในกระบวนการปรับค่าความเป็นกรด-ด่างให้อยู่ในภาวะเป็นกลาง 31
ตารางที่ 6.1 ตารางเปรียบเทียบค่าพารามิเตอร์ที่ได้จากโปรแกรม และค่าพารามิเตอร์ที่ได้จากวิธีฟิต 3 41
ตารางที่ 6.2 เปรียบเทียบค่าดัชนี IAE ในกรณีที่ 6.3.1 47
ตารางที่ 6.3 เปรียบเทียบค่าดัชนี IAE ในกรณีที่ 6.3.2 49
ตารางที่ 6.4 เปรียบเทียบค่าดัชนี IAE ในกรณีที่ 6.3.3 51
ตารางที่ 6.5 ค่าตัวแปรที่ใช้ในแบบจำลองทางคณิตศาสตร์ ณ สภาวะคงที่ 53
ตารางที่ 6.6 สรุปค่าดัชนี IAE ของการควบคุมกระบวนการอันดับสองที่มีเวลาหน่วง 64
ตารางที่ 6.7 ค่าพารามิเตอร์ที่ได้จากการประมาณกระบวนการด้วยวิธีฟิต 3 เมื่อมีการเปลี่ยนแปลงตัวแปรอินพุทแบบขั้น หลายๆ ค่า 66
ตารางที่ 6.8 สรุปค่าดัชนี IAE ของตัวควบคุมบนกระบวนการที่มีความไม่เป็นเชิงเส้นสูง เมื่อมีการเปลี่ยนแปลงของตัวแปรรบกวน (F_1) 86
ตารางที่ 6.9 สรุปค่าดัชนี IAE ของตัวควบคุมบนกระบวนการที่มีความไม่เป็นเชิงเส้นสูง เมื่อมีการเปลี่ยนแปลงของตัวแปรรบกวน (C_1) 87
ตารางที่ ข.1 กฎการปรับค่าตัวควบคุมพีไอดีตามหลักของ Ziegler-Nichols 111
ตารางที่ ข.2 กฎการปรับค่าตัวควบคุมพีไอดีตามหลักของ Cohen-Coon 112
ตารางที่ ข.3 กฎการปรับค่าตัวควบคุมพีไอดีตามหลักของการทำให้ค่า ITAE ต่ำสุด 112

คำอธิบายสัญลักษณ์

a	ค่าพารามิเตอร์
b	ค่าพารามิเตอร์
t	เวลา
k	ค่าคงที่ของปฏิกิริยา
pH	ค่าพีเอช
$u(t)$	อินพุทของกระบวนการ
$y(t)$	เอาท์พุทของกระบวนการ
K	ค่าเกนของกระบวนการ
K_p	ค่าเกนของกระบวนการ
K_c	ค่าเกนของตัวควบคุมพีไอดี
T_s	เวลาในการเก็บตัวอย่าง
V	ปริมาตรของเครื่องปฏิกรณ์
F	อัตราการไหล
F_s	อัตราการไหลของตัวทำละลาย
F_1	อัตราการไหลของสายกรด
F_2	อัตราการไหลของสายด่าง
C_1	ความเข้มข้นของสายกรด
C_2	ความเข้มข้นของสายด่าง
C_{A0}	ความเข้มข้นของสาร A ขาเข้า
C_{A1}	ความเข้มข้นของสาร A ขาออกจากถังที่ 1
C_{A2}	ความเข้มข้นของสาร A ขาออกจากถังที่ 2
H^+	ความเข้มข้นของไฮโดรเจนอิออน
S	ความชันของเส้นโค้ง
ε	ค่าความผิดพลาด
θ	เวลาหน่วง
λ	สัมประสิทธิ์การลืมค่าตัวแปร
ψ	เวคเตอร์ข้อมูล

ξ	ความเข้มข้นของอะซิเตทอิออน
ζ	ความเข้มข้นของโซเดียมอิออน
τ	ค่าเวลาคงที่ของกระบวนการ
τ_I	ค่าพารามิเตอร์อินทีกรัลของตัวควบคุมพีไอดี
τ_D	ค่าพารามิเตอร์ดีริเวทีฟของตัวควบคุมพีไอดี
Δt	ค่าเวลาในการเก็บตัวอย่าง
ΔX	ความแตกต่างของอินพุท
ΔY	ความแตกต่างของเอาท์พุท

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1

บทนำ

1.1 มูลเหตุจูงใจและที่มาของงานวิจัย

การควบคุมกระบวนการในอุตสาหกรรมได้รับการพัฒนามาอย่างต่อเนื่อง เป็นเวลาหลายสิบปีมาแล้ว ด้วยเหตุผลที่ว่าการควบคุมกระบวนการที่มีประสิทธิภาพนั้น สามารถทำให้กระบวนการมีความปลอดภัยเพิ่มขึ้น และสร้างผลกำไรในการผลิตให้สูงขึ้นได้ โดยรูปแบบในระยะเริ่มต้นของการควบคุมกระบวนการ ในช่วงปี 1940 นั้น (Marlin, 2000) เป็นการควบคุมป้อนกลับแบบพีไอดีธรรมดา ซึ่งเป็นตัวควบคุมอนาล็อกแบบอินพุทเดียว-เอาท์พุทเดียว (Single-Input, Single-Output; SISO) ที่มีรูปแบบของการคำนวณที่ใช้ในการควบคุม เป็นแบบอย่างง่ายและมีข้อจำกัดในหลายๆ ด้าน เพราะว่าการคำนวณต่างๆ ต้องทำโดยอาศัยอุปกรณ์ฮาร์ดแวร์ การคำนวณที่ยุ่งยากและซับซ้อน จึงเป็นไปไม่ได้เลย จึงเป็นเหตุให้การควบคุมด้วยตัวควบคุมพีไอดี ซึ่งเป็นตัวควบคุมที่มีอัลกอริธึมไม่ซับซ้อน เป็นที่นิยมและใช้กันโดยทั่วไปในงานอุตสาหกรรมต่างๆ และอีกทั้งกระบวนการผลิตในอุตสาหกรรมในช่วงเวลานั้น ยังไม่มีการผลิตที่ซับซ้อนมากนัก ประสิทธิภาพในการผลิตยังไม่ใช้สิ่งที่มีความจำเป็นมากนัก ผลการควบคุมและประสิทธิภาพของตัวควบคุมพีไอดีธรรมดา จึงอยู่ในเกณฑ์ที่สามารถยอมรับได้ในอุตสาหกรรม

อย่างไรก็ตาม ในเวลาต่อมา เมื่อกระบวนการผลิตในอุตสาหกรรม ได้รับการวิจัยและพัฒนาให้มีกระบวนการที่ซับซ้อนมากขึ้น และมีประสิทธิภาพในการผลิตที่สูงขึ้น ตัวควบคุมพีไอดีธรรมดาที่เคยได้รับการยอมรับ ก็เริ่มให้ผลการควบคุมที่ไม่เป็นที่น่าพอใจในสายตาของวิศวกรมากยิ่งขึ้นกว่าเดิม โดยเฉพาะอย่างยิ่งในการควบคุมกระบวนการที่มีความไม่เป็นเชิงเส้นสูง เช่น กระบวนการควบคุมค่าความเป็นกรด-ด่าง หรือกระบวนการที่มีอินพุท-เอาท์พุทหลายตัว (Multi-Input, Multi-Output; MIMO) ด้วยเหตุนี้เอง จึงเป็นแรงจูงใจให้เกิดการพัฒนา คิดค้นหากลไกที่จะทำให้การควบคุมกระบวนการโดยใช้ตัวควบคุมพีไอดีมีประสิทธิภาพสูงขึ้นกว่าเดิม เป็นไปอย่างกว้างขวาง

ต่อมาในช่วงปี 1960 เมื่อมีการพัฒนาคอมพิวเตอร์ดิจิตอลให้มีความเร็วสูง และมีประสิทธิภาพที่เพิ่มขึ้น การประยุกต์ใช้คอมพิวเตอร์ในการควบคุมกระบวนการในอุตสาหกรรม ก็เติบโตอย่างรวดเร็ว ได้มีการคิดค้นรูปแบบการควบคุมแบบใหม่ๆ ที่มีความซับซ้อนและแม่นยำมากขึ้น ไม่ว่าจะเป็นการควบคุมที่อาศัยแบบจำลองของกระบวนการ เป็นพื้นฐานในการคำนวณ เช่น การควบคุมแบบทำนายด้วยแบบจำลอง (Model Predictive Control; MPC) หรือการควบคุมที่อาศัย

หลักปัญญาประดิษฐ์ (Artificial Intelligent) ดังเช่น การควบคุมโดยใช้โครงข่ายประสาท (Neural Network Control) หรือการควบคุมโดยใช้ตรรกฟัซซี (Fuzzy Logic Control) ที่สามารถตอบสนองต่อระบบที่มีความไม่เป็นเชิงเส้นสูงได้ดี แต่เนื่องจากการควบคุมด้วยตัวควบคุมที่อาศัยการคำนวณในรูปแบบต่างๆ เหล่านี้มีความเข้าใจยากและซับซ้อน จึงทำให้การใช้งานจริงของตัวควบคุมดิจิตอลคอมพิวเตอร์ ยังคงเป็นที่นิยมใช้พื้นฐานการคำนวณตามหลักของตัวควบคุมพีไอดี กันอยู่ในกระบวนการอุตสาหกรรม

การออกแบบตัวควบคุมเพื่อใช้กับกระบวนการที่มีความไม่เป็นเชิงเส้นสูงนั้น ถึงแม้เราจะสามารถสร้างแบบจำลองทางคณิตศาสตร์เพื่อใช้ในการควบคุมขึ้นมาได้ แต่ก็ยังคงมีความแตกต่างของแบบจำลองกับกระบวนการจริง เนื่องจากความเป็นพลวัติของกระบวนการ ทำให้พารามิเตอร์หลายๆ ตัวในระบบ มีค่าเปลี่ยนแปลงไปตลอดเวลา ซึ่งส่งผลให้การควบคุมด้วยตัวควบคุมที่สร้างขึ้นจากแบบจำลองที่มีพารามิเตอร์คงที่มีสมรรถนะในการควบคุมลดลง จึงได้มีการพัฒนาตัวควบคุมแบบปรับตัวได้ (Adaptive Control) ขึ้นมาเพื่อแก้ไขข้อบกพร่องนี้ ทำให้การควบคุมเป็นไปได้อย่างมีประสิทธิภาพมากขึ้น และสามารถทนทานต่อความแตกต่างที่เกิดขึ้นของแบบจำลองกับกระบวนการจริงเพิ่มขึ้นด้วย

สำหรับงานวิจัยนี้ ได้หยิบยกตัวควบคุมพีไอดีขึ้นมาศึกษา เนื่องจากเหตุผลดังที่ได้กล่าวมาแล้วข้างต้น แต่เพื่อให้ตัวควบคุมที่ต้องการนำเสนอนี้ มีความสามารถในการที่จะรองรับกับการทำงานบนกระบวนการที่มีความไม่เป็นเชิงเส้นสูงด้วย จึงใช้ข้อดีของการควบคุมแบบปรับตัวได้เข้ารวมกันด้วย ดังนั้นรูปแบบของตัวควบคุมจึงมีลักษณะเป็นตัวควบคุมพีไอดีที่สามารถจะปรับค่าพารามิเตอร์เองได้ ตามความไม่เป็นเชิงเส้นของกระบวนการ ซึ่งตัวควบคุมนี้ จะเรียกว่า ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ (Self-tuning PID controller) โดยในที่นี้ จะประยุกต์ใช้กระบวนการทางคณิตศาสตร์ที่เรียกว่า เจเนติกอัลกอริธึม (Genetic Algorithms) ในการหาค่าพารามิเตอร์ที่เหมาะสมของตัวควบคุมพีไอดีอีกด้วย

ในการพัฒนารูปแบบของตัวควบคุมหลายครั้งที่เกิดขึ้น กระบวนการควบคุมค่าความเป็นกรด-ด่างหรือค่าพีเอช (pH) ซึ่งเป็นกระบวนการที่มีความไม่เป็นเชิงเส้น ได้ถูกนำมาใช้เป็นเกณฑ์ในการทดสอบตัวควบคุมที่ได้นำเสนอขึ้นมา เนื่องด้วยความไม่เป็นเชิงเส้นสูงของกระบวนการ โดยเฉพาะในช่วงค่าพีเอชที่ใกล้ค่าที่เป็นกลาง และความไวต่อการปรับเปลี่ยนของค่าตัวแปรปรับมาก นอกจากนี้กระบวนการควบคุมค่าพีเอช ยังเป็นกระบวนการที่เกี่ยวข้องอยู่ในอุตสาหกรรมส่วนใหญ่ เช่น ในการควบคุมสภาวะที่เหมาะสมที่สุดของปฏิกิริยา หรือป้องกันการเกิดปฏิกิริยาที่ไม่ต้องการหรือปฏิกิริยาข้างเคียง และโดยเฉพาะอย่างยิ่ง ในการควบคุมกระบวนการบำบัดน้ำเสีย

จากอุตสาหกรรม ฯลฯ ดังนั้น จะเห็นได้ว่าการปรับปรุงการควบคุมกระบวนการควบคุมค่าพีเอช ให้อยู่ในค่าที่ต้องการจึงเป็นประโยชน์อย่างยิ่ง

1.2 วัตถุประสงค์ของงานวิจัย

1.2.1 เพื่อพัฒนาการควบคุมด้วยตัวควบคุมพีไอดี โดยใช้หลักของตัวควบคุมชนิดปรับค่าเองได้ (Self-tuning Controller) ซึ่งตัวควบคุมจะสามารถปรับปรุงค่าพารามิเตอร์ที่ใช้ในการควบคุมคือ K_C, τ_I, τ_D ได้ด้วยตัวเอง โดยใช้เจเนติกอัลกอริธึม (Genetic Algorithms) เพื่อหาค่าที่เหมาะสมที่สุดในการคำนวณในขั้นตอนต่างๆ

1.2.2 สร้างโปรแกรม และทดสอบการหาค่าเหมาะสมที่สุดด้วยเจเนติกอัลกอริธึม กับฟังก์ชันทดสอบที่เป็นมาตรฐาน คือฟังก์ชั่นโรเซ็นบรอกค์ (Rosenbrock) ที่มีสองตัวแปร ดังที่แสดงในสมการที่ (1.1)

$$\min f(x_1, x_2) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2 \qquad (1.1)$$

เมื่อ $-2.048 \le x_i \le 2.048$

1.2.3 ทดสอบตัวประมาณค่าพารามิเตอร์ของกระบวนการ ด้วยเทคนิคกำลังสองน้อยที่สุด (Least Square) กับกระบวนการที่ทราบค่าพารามิเตอร์แน่นอนอื่นๆ เพื่อเปรียบเทียบ

1.3 ขอบเขตของงานวิจัย

1.3.1 กระบวนการที่สนใจคือ กระบวนการปรับค่าความเป็นกรด-ด่าง (ค่าพีเอช) ให้มีสภาพเป็นกลาง ที่เกิดขึ้นในถังปฏิกรณ์แบบกวนต่อเนื่อง โดยที่สายป้อนกรดจะมีอัตราการไหลคงที่ และใช้การปรับอัตราการไหลในสายป้อนด่าง เพื่อปรับสภาพความเป็นกรด-ด่างของสายผลิตภัณฑ์ ให้อยู่ที่สภาวะที่ต้องการ

1.3.2 จำลองกระบวนการ และการควบคุมบนโปรแกรมแม็ทแลบ (MatLAB®)

1.3.3 สร้างแบบจำลองการควบคุมกระบวนการปรับค่าความเป็นกรด-ด่างให้มีค่าเป็นกลาง โดยใช้ตัวควบคุมป้อนกลับพีไอดีชนิดปรับค่าเองได้ ที่มีการประยุกต์ใช้เจเนติกอัลกอริธึมใน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ขั้นตอนต่างๆ แล้วทดสอบผลการตอบสนองเปรียบเทียบกับการควบคุมด้วยวิธีอื่นๆ เช่น การหาค่าพารามิเตอร์ที่เหมาะสมของตัวควบคุมพีไอดีด้วยวิธี ของ Ziegler-Nichols ฯลฯ

1.4 ขั้นตอนการดำเนินงานวิจัย

แบ่งออกได้เป็น 5 ขั้นตอน ดังนี้

1.4.1 ศึกษาและค้นคว้าผลงานวิจัยในช่วงเวลาที่ผ่านมา ที่เกี่ยวข้องกับตัวควบคุมพีไอดีปรับค่าได้เอง และกระบวนการปรับค่าความเป็นกรด-ด่างให้มีสภาพเป็นกลาง รวมทั้งการใช้งานเจเนติกอัลกอริธึมในการควบคุมกระบวนการ

1.4.2 จำลองกระบวนการโดยใช้แบบจำลองทางคณิตศาสตร์ที่สร้างขึ้นมา โดยใช้โปรแกรมสำเร็จรูปแม็ทแลบ (MatLAB®)

1.4.3 ออกแบบโครงสร้างของตัวควบคุมสำหรับการควบคุม

1.4.4 จำลองการควบคุม ของตัวควบคุมและกระบวนการที่ได้สร้างขึ้น โดยใช้โปรแกรมสำเร็จรูปแม็ทแลบ (MatLAB®)

1.4.5 วิเคราะห์ผลการจำลองการควบคุมกระบวนการ และจัดทำวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 ทำให้ผลการควบคุมภายใต้ของตัวควบคุมพีไอดีปรับค่าได้เอง เป็นไปอย่างมีประสิทธิภาพมากขึ้น, มีความทนทานสูงขึ้น

1.5.2 ทำให้การควบคุมกระบวนการปรับค่าความเป็นกรด-ด่างให้มีสภาพเป็นกลาง มีสมรรถนะสูงขึ้น โดยที่ยังสามารถใช้ตัวควบคุมพีไอดีได้

1.6 ระยะเวลาในการดำเนินงาน

ระยะเวลาในการดำเนินงานวิจัยตามขั้นตอนในหัวข้อ 1.4 จะเริ่มตั้งแต่ เดือนพฤศจิกายน พ.ศ. 2544 ถึง เดือนมีนาคม พ.ศ. 2546 ดังแสดงในตารางที่ 1.1

ตารางที่ 1.1 ระยะเวลาในการดำเนินงานวิจัย

ขั้นตอน	พ.ย. 2544	ธ.ค. 2544	ม.ค. 2545	ก.พ. 2545	มี.ค. 2545	เม.ย. 2545	พ.ค. 2545	มิ.ย. 2545	ก.ค. 2545	ส.ค. 2545	ก.ย. 2545	ต.ค. 2545	พ.ย. 2545	ธ.ค. 2545	ม.ค. 2546	ก.พ. 2546	มี.ค. 2546
1.4.1	←	—	—	—	→												
1.4.2					←	—	—	→									
1.4.3								←	—	—	→						
1.4.4											←	—	→				
1.4.5													←	—	—	—	→

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

ผลงานวิจัยที่เกี่ยวข้อง

ผลงานที่เกี่ยวข้องกับงานวิจัยในครั้งนี้มีอยู่มากมาย ทั้งในส่วนที่เกี่ยวข้องกับตัวควบคุมพีไอดีชนิดต่างๆ ส่วนที่เกี่ยวข้องกับตัวควบคุมชนิดปรับค่าเองได้ ส่วนที่เกี่ยวข้องกับกระบวนการปรับค่าพีเอช และส่วนที่เกี่ยวข้องกับการประยุกต์ใช้เจเนติกอัลกอริธึมในการควบคุมกระบวนการ ดังนั้นในบทนี้ จะแยกงานวิจัยที่น่าสนใจออกเป็นกลุ่มๆ ดังที่ได้กล่าวมาข้างต้นคือ

2.1 งานวิจัยที่เกี่ยวข้องกับตัวควบคุมพีไอดีชนิดต่างๆ

2.2 งานวิจัยที่เกี่ยวข้องกับตัวควบคุมชนิดปรับตัวได้เอง

2.3 งานวิจัยที่เกี่ยวข้องกับกระบวนการปรับค่าพีเอช

2.4 งานวิจัยที่เกี่ยวข้องกับการประยุกต์ใช้เจเนติกอัลกอริธึมในการควบคุมกระบวนการ

สำหรับในงานวิจัยที่มีเนื้อหาเกี่ยวข้องกับหลายส่วนด้วยกัน ก็จะจัดไว้ในกลุ่มใดกลุ่มหนึ่งที่เกี่ยวข้องมากที่สุดเท่านั้น เพื่อให้เกิดความสะดวกในการอ้างอิง และค้นคว้าต่อไป

2.1 งานวิจัยที่เกี่ยวข้องกับตัวควบคุมพีไอดีชนิดต่างๆ

Y. K. Yeo และ T. I. Kwon (1998) ได้ศึกษาการควบคุมกระบวนการปรับค่าความเป็นกรด-ด่างให้มีสภาพเป็นกลาง โดยใช้ตัวควบคุมแบบนิวรัลพีไอดี (neural PID) บนแบบจำลองเครื่องปฏิกรณ์ถังกวนต่อเนื่องที่นำเสนอโดย McAvoy et al. (1972) โดยงานวิจัยนี้ได้เสนอตัวควบคุมแบบพีไอดีที่มีการปรับแต่งพารามิเตอร์โดยใช้โครงข่ายประสาท พบว่าการควบคุมโดยใช้ตัวควบคุมที่นำเสนอนี้ให้สมรรถนะที่ดีต่อการเปลี่ยนแปลงค่าเป้าหมาย ถึงแม้จะมีตัวแปรรบกวนและสัญญาณรบกวน และเมื่อนำตัวควบคุมที่ได้ไปใช้กับการทดลองจริง พบว่าให้การควบคุมที่เข้าสู่ค่าเป้าหมายได้

J. H. Kim และ S. J. Oh (2000) ได้ศึกษาถึงการใช้ตัวควบคุมแบบฟัซซีพีไอดี (fuzzy PID controller) เพื่อควบคุมกระบวนการที่มีความไม่เป็นเชิงเส้น โดยในงานวิจัยนี้ได้ศึกษาโดยการใช้ตัวแปรขาเข้า 3 ตัวนั่นคือ ความผิดพลาด (error) อัตราการเปลี่ยนแปลงของความผิดพลาด (rate

of change of error) และความเร่งของการเปลี่ยนแปลงของความผิดพลาด (accelerated rate of change of error) ป้อนเข้าสู่กล่องการควบคุมฟัซซี 2 กล่องที่ขนานและเป็นอิสระจากกัน รวมทั้งใช้กฎของฟัซซีที่แตกต่างกัน เพื่อคำนวณหาค่าพารามิเตอร์ทั้ง 3 ของตัวควบคุมพีไอดี ผลจากการทดสอบตัวควบคุมบนกระบวนการที่มีความไม่เป็นเชิงเส้นเปรียบเทียบกับตัวควบคุมแบบฟัซซีพีไอพบว่า ตัวควบคุมแบบฟัซซีพีไอดีที่นำเสนอนี้ ให้ผลการตอบสนองที่ดีกว่า

2.2 งานวิจัยที่เกี่ยวข้องกับตัวควบคุมชนิดปรับค่าเองได้

Michiyo et al. (2001) ได้นำเสนอตัวควบคุมพีไอดีชนิดปรับค่าเองได้ ที่มีการรวมเทคนิคของโครงข่ายประสาทเข้าไว้ในตัวควบคุม เพื่อให้สามารถปรับค่าเกนของตัวควบคุมได้อย่างรวดเร็วทันการตอบสนองของกระบวนการที่มีความไม่เป็นเชิงเส้น โดยได้มีการแบ่งค่าพารามิเตอร์ของตัวควบคุมพีไอดีออกเป็น 2 ส่วน คือ ส่วนที่เป็นการคำนวณมาจากโครงสร้างของระบบปรับค่าเองได้ที่ใช้วิธีทำซ้ำกำลังสองน้อยที่สุด (recursive least square) และอีกส่วนหนึ่งเป็นส่วนที่คำนวณมาจากเทคนิคของโครงข่ายประสาท ที่มีชั้นของโครงข่าย 3 ชั้นและใช้วิธีถ่วงน้ำหนักคือ วิธีโพรพากันดาย้อนกลับ (back propaganda method) จากการทดสอบตัวควบคุมบนกระบวนการที่มีความไม่เป็นเชิงเส้นพบว่า พารามิเตอร์ของตัวควบคุมมีการปรับค่าได้ตามความเปลี่ยนแปลงของกระบวนการอย่างเหมาะสม ในงานวิจัยนี้ไม่ได้นำเสนอการเปรียบเทียบผลการควบคุมกับตัวควบคุมอื่นๆ เลย

M. A. Henson และ D. E. Seborg (1994) นำเสนอผลการวิจัยเกี่ยวกับวิธีการควบคุมไม่เป็นเชิงเส้นแบบปรับตัวได้ (adaptive nonlinear control) เพื่อใช้ในการควบคุมกระบวนการปรับค่าพีเอชให้เป็นกลาง โดยตัวควบคุมที่นำเสนอนี้ ใช้วิธีทำให้เป็นเชิงเส้นของตัวแปรขาเข้าและตัวแปรขาออกที่ได้รับการปรับปรุงใหม่ (modified input-output linearization) นอกจากนั้นยังมีการใช้ตัวสังเกตแบบวงเปิด ลดอันดับ (reduced-order, open-loop observer) และการหาค่าพารามิเตอร์ด้วยวิธีทำซ้ำกำลังสองน้อยที่สุด ในตัวควบคุมที่สนใจนี้ด้วย ผลจากการทดสอบด้วยการจำลองระบบ และการทดสอบจริงในห้องปฏิบัติการ แสดงให้เห็นว่า ผลการควบคุมของตัวควบคุมไม่เป็นเชิงเส้นแบบปรับตัวได้ให้ผลการตอบสนองที่ดีกว่า เมื่อเปรียบเทียบกับตัวควบคุมเชิงเส้นธรรมดา (ตัวควบคุมพีไอ) และตัวควบคุมไม่เป็นเชิงเส้นทั่วไป

Tae Chul Lee et al. (2001) ได้นำเสนอรูปแบบการควบคุมแบบปรับตัวได้ย้อนกลับไม่ผ่านโดยตรง (indirect adaptive back-stepping control scheme) บนกระบวนการปรับค่าพีเอช

ให้เป็นกลางที่มีตัวแปรไม่ทราบค่าบางตัว อีกทั้งยังมีการประยุกต์ใช้วิธีการทำซ้ำการทำนายความผิดพลาด (recursive prediction error method) เพื่อใช้ในการประมาณค่าพารามิเตอร์ จากผลการทดสอบเชิงตัวเลข พบว่า พารามิเตอร์ที่ได้รับการประมาณค่า จะให้ค่าได้ถูกต้องใกล้เคียงกับค่าพารามิเตอร์จริง ซึ่งยังเป็นผลให้การควบคุมด้วยตัวควบคุมชนิดนี้ควบคุมกระบวนการได้อย่างราบเรียบอีกด้วย

2.3 งานวิจัยที่เกี่ยวข้องกับกระบวนการปรับค่าพีเอช

T. K. Gustafsson และ K. V. Waller (1982) ได้เสนอแบบจำลองสเตท (state space model) ของกระบวนการปรับความเป็นกรด-ด่างให้มีสภาพเป็นกลางในเครื่องปฏิกรณ์แบบถังกวนต่อเนื่อง โดยแบบจำลองที่ได้จะสามารถแบ่งออกเป็น 2 ส่วนคือ ส่วนที่อธิบายลักษณะทางกายภาพของระบบ ซึ่งไม่เกี่ยวข้องกับปฏิกิริยา (reaction invariant) และส่วนที่อธิบายปฏิกิริยาที่เกิดขึ้น (reaction variant) ซึ่งในกรณีที่ปฏิกิริยาระหว่างกรดและด่างเกิดขึ้นรวดเร็ว สามารถที่จะละเลยส่วนที่อธิบายปฏิกิริยาได้ ซึ่งทำให้ตัวแปรสเตทในแบบจำลองนี้เป็นตัวแปรที่ไม่เปลี่ยนแปลงตามปฏิกิริยา ในงานวิจัยนี้ได้มีการจำลองการควบคุมโดยใช้แบบจำลองนี้ กับตัวควบคุมแบบปรับตัวได้ ที่มีการประมาณค่าสเตทและพารามิเตอร์ด้วยตัวกรองคาลมาน พบว่าสามารถควบคุมได้ดี แต่ยังไม่มีการสรุปผลในงานวิจัยนี้

N. Bhat และ T. J. McAvoy (1989) ได้ศึกษาพลวัติของการควบคุมกระบวนการทางเคมีโดยใช้โครงข่ายประสาท (Neural Network) ที่มีการประยุกต์ใช้อัลกอริธึมแบบจำลองพลวัติโพรพาเกชันย้อนกลับ (Back propagation Dynamic Model: BDM) ซึ่งมีความสามารถในการเรียนรู้แบบจำลองที่มีความไม่เป็นเชิงเส้นได้ดี โดย Bhat ใช้การจำลองการตอบสนองของกระบวนการควบคุมค่าความเป็นกรด-ด่าง ในเครื่องปฏิกรณ์ถังกวนต่อเนื่องที่นำเสนอโดย McAvoy et al. (1972) เพื่อศึกษาเรื่องนี้ และเมื่อเปรียบเทียบความสามารถในการจำลองกระบวนการกับวิธีที่ใช้แบบจำลอง ARMA พบว่าการจำลองกระบวนการโดยใช้โครงข่ายประสาทแบบ back propagation ให้ผลการตอบสนองที่ใกล้เคียงกับแบบจำลองต้นแบบมากกว่า นอกจากนี้ ยังได้มีการประยุกต์ใช้การจำลองกระบวนการ BDM กับตัวควบคุมแบบแบบจำลองภายใน (Internal Model Control: IMC) เปรียบเทียบกับการใช้แบบจำลอง ARMA ทั่วไป พบว่าได้ผลการตอบสนองใกล้เคียงกัน

S. J. Norquay, A. Palazoglu และ J. A. Romagnoli (1997) ได้ใช้แบบจำลอง Wiener ที่ประกอบไปด้วยองค์ประกอบส่วนพลวัติเชิงเส้น และส่วนสถิตไม่เป็นเชิงเส้น ในการจำลองกระบวนการปรับค่าความเป็นกรด-ด่างให้มีสภาพเป็นกลาง โดยประยุกต์ใช้ในตัวควบคุมแบบทำนายด้วยแบบจำลอง (model predictive control : MPC) เปรียบเทียบผลการควบคุมกับตัวควบคุมแบบทำนายด้วยแบบจำลองเชิงเส้นทั่วไป และตัวควบคุมพีไอดีที่ปรับพารามิเตอร์ตามวิธีของ Cohen-Coon พบว่าเมื่อเปรียบเทียบโดยใช้ค่า ISE เป็นเกณฑ์ ตัวควบคุม MPC ที่อาศัยแบบจำลอง Wiener จะให้การตอบสนองที่มี ISE น้อยที่สุด โดยที่ตัวควบคุม MPC เชิงเส้นทั่วๆ ไปจะให้ผลที่รองลงมา และตัวควบคุมพีไอดีให้ ISE ที่สูงที่สุด ทั้งในกรณีเปลี่ยนแปลงค่าเป้าหมาย และกรณีตัวแปรรบกวนมีการเปลี่ยนแปลง

ไพศาล และ วนารัตน์ (1999) ได้เสนอการประยุกต์ใช้ตัวควบคุมแบบป้อนกลับสเตท (state feedback controller) ในการควบคุมค่าความเป็นกรด-ด่างของน้ำเสียจากโรงงานเหล็กรีดเย็น โดยที่มีการใช้ตัวกรองคาลมานช่วยในการหาค่าพารามิเตอร์ของกระบวนการ ในงานวิจัยนี้ได้เปรียบเทียบผลการจำลองการควบคุม กับตัวควบคุมพีไอดีที่ปรับค่าโดยวิธีของ Ziegler-Nichole โดยใช้ค่าอินทีกรัลของความผิดพลาดสัมบูรณ์ (IAE) ที่เวลา 0 ถึง 60 นาที เป็นเกณฑ์ โดยแบ่งออกเป็น 4 กรณีคือ ที่สภาวะคงที่, เมื่อมีตัวรบกวน, เมื่อมีความผิดพลาดของแบบจำลอง และเมื่อมีสัญญาณรบกวน พบว่าตัวควบคุมแบบป้อนกลับสเตทมีสมรรถนะสูงกว่าตัวควบคุมพีไอดีในทุกกรณี และในกรณีที่มีสัญญาณรบกวน การใช้ตัวกรองคาลมาน (kalman filter) ยังสามารถช่วยเพิ่มสมรรถนะให้กับตัวควบคุมอีกด้วย

ไพศาล และ นันทนา (2000) ได้ศึกษาถึงการประยุกต์ใช้ตัวควบคุมแบบโกลบอลลิเนียไรซิง (global linearizing control: GLC) ร่วมกับการประมาณค่าตัวแปรสเตท และพารามิเตอร์ด้วยตัวกรองคาลมาน ในการควบคุมค่าความเป็นกรด-ด่าง และควบคุมความเข้มข้นของโลหะหนัก ในกระบวนการบำบัดน้ำเสีย ของโรงงานชุบโลหะด้วยไฟฟ้า ในงานวิจัยนี้ได้มีการเปรียบเทียบสมรรถนะ และความทนทานของตัวควบคุมที่นำเสนอ กับการควบคุมที่ใช้ตัวควบคุมแบบพีไอดี ในกรณีศึกษาต่างๆ คือ กรณีที่มีการเปลี่ยนแปลงค่าของตัวแปรรบกวน, กรณีที่มีสัญญาณรบกวนในตัวแปรวัด และกรณีที่มีความผิดพลาดของพารามิเตอร์ของแบบจำลอง หรือตัวควบคุม ผลการจำลองที่ได้แสดงให้เห็นว่า อัลกอริธึมของการความคุมแบบ GLC ที่ใช้ร่วมกับตัวกรองคาลมาน ให้สมรรถนะและความทนทานดีกว่าการควบคุมแบบพีไอดี

2.4 งานวิจัยที่เกี่ยวข้องกับการประยุกต์ใช้เจเนติกอัลกอริธึมในการควบคุมกระบวนการ

V. Hanagandi และ M. Nikolaou (1995) ได้ศึกษาถึงการประยุกต์ใช้ เจเนติกอัลกอริธึม ในการหาค่าพารามิเตอร์ที่ดีที่สุดของตัวควบคุมแบบพีไอดี ที่จะให้ผลการตอบสนองป้อนกลับต่อการเปลี่ยนแปลงค่าเป้าหมายบนกระบวนการอันดับสอง เป็นไปอย่างรวดเร็วและเสถียร พบว่าเจเนติกอัลกอริธึมสามารถหาค่าพารามิเตอร์ที่ให้ผลการตอบสนองเป็นไปอย่างที่ได้ตั้งเป้าหมายไว้ได้ และดีขึ้นเรื่อยๆ ในทุกรอบของการคำนวณ และนอกจากนี้เขายังทำการศึกษาการใช้เจเนติกอัลกอริธึมในการหาค่าพารามิเตอร์ของการควบคุมในกระบวนการหอกลั่นที่นำเสนอโดย Chien และ Ogunnaike (1992) ที่ใช้ตัวควบคุมแบบพีไอสองตัวในการควบคุมกระบวนการแบบอินพุท-เอาท์พุทหลายตัว พบว่าค่าพารามิเตอร์ที่คำนวณได้ในแต่ละรอบจะให้ผลการตอบสนองที่ดีขึ้นเรื่อยๆ เช่นกัน

Y. Mitsukura, T. Yamamoto และ M. Kaneda (1998) ได้ทำการศึกษาการปรับแต่งค่าพารามิเตอร์ของตัวควบคุมพีไอดี โดยใช้เจเนติกอัลกอริธึม ที่อ้างอิงความสัมพันธ์ของการควบคุมด้วยพีไอดีกับกฎการควบคุมแบบแปรปรวนน้อยที่สุดทั่วไป (generalized minimum variance control) โดยที่มุ่งเน้นไปที่กระบวนการที่มีเวลาหน่วงสูง (2-3 เท่าของเวลาคงที่ของกระบวนการ) โดยเปรียบเทียบผลการควบคุมกับกรณีที่ใช้ตัวควบคุมแบบจำลองภายใน (Internal Model Control: IMC) และวิธีการควบคุมแบบ Chien, Hrones และ Reswick (CHR) (1952) พบว่าการปรับค่าพารามิเตอร์แบบ CHR ให้ผลการตอบสนองที่ดี เมื่อเวลาหน่วงน้อย (θ/τ = 1) แต่จะมีการตอบสนองที่แกว่งค่อนข้างมากขึ้น เมื่อมีเวลาหน่วงเพิ่มขึ้น (θ/τ = 2 และ 3) ส่วนการควบคุมแบบ IMC ให้ผลการตอบสนองที่ยอมรับได้ในทุกกรณี แต่ยังมีการตอบสนองที่พุ่งขึ้นสูง (overshoot) ในตอนต้น ส่วนการควบคุมที่ได้นำเสนอนั้น จะมีการตอบสนองที่ค่อนข้างดีในทุกกรณี

H. Al-Duwaish และ N. Wasif (2001) ได้ทำการศึกษาวิจัยการประยุกต์ใช้เจเนติก อัลกอริธึมในตัวควบคุมแบบทำนายด้วยแบบจำลองไม่เป็นเชิงเส้น (nonlinear model predictive control) โดยที่ใช้ตัวควบคุมที่นำเสนอทดสอบกับระบบไม่เป็นเชิงเส้น 2 ระบบ นั่นคือ ระบบเครื่องแลกเปลี่ยนความร้อนของแฮมเมอร์สเตียน (Hammerstein) ที่มีตัวแปรขาเข้าเป็นอัตราการไหลของสารในกระบวนการ และมีตัวแปรขาออกเป็นอุณหภูมิ ณ ทางออก และระบบวาล์วควบคุมของไวเนอร์ (Wiener) ที่มีตัวแปรขาเข้าเป็นความดันลม และมีตัวแปรขาออกเป็นอัตราการไหลผ่านวาล์ว พบว่าตัวควบคุมสามารถปรับตัวแปรวัดของกระบวนการให้เดินตามค่าเป้าหมายได้อย่างราบเรียบในทั้งสองกระบวนการ

บทที่ 3
พื้นฐานเจเนติกอัลกอริธึม

การหาค่าที่เหมาะสมที่สุด (Optimization) เป็นเครื่องมือทางคณิตศาสตร์อย่างหนึ่ง ที่เราใช้กันทั่วไปในการหาค่าต่ำสุด หรือค่าสูงสุดของความสัมพันธ์ทั้งหลาย วิธีการที่ใช้ในการหาจุดเหมาะสมที่สุด ที่นิยมกันแพร่หลายนั้นสามารถแบ่งได้เป็นสองกลุ่มใหญ่ๆ คือ โปรแกรมเชิงเส้น (Linear Programming) ซึ่งมีตัวอย่างคือ วิธีการซิมเพล็กซ์ (Simplex Method) และ โปรแกรมไม่เชิงเส้น (Non-linear Programming) ตัวอย่างเช่น โปรแกรมควอดราติก (Quadratic Programming), วิธีการลดลงอย่างชันที่สุด (Method of Steepest Descent) ฯลฯ ซึ่งแต่ละวิธีก็สามารถเข้าถึงคำตอบของปัญหาที่ต้องการได้แตกต่างกัน รวมถึงความแตกต่างในด้านความแม่นยำ และความรวดเร็วในการหาคำตอบด้วย

กรรมวิธีการคัดเลือกที่เกิดขึ้นตามธรรมชาติ (Natural Selection) เช่น การสืบพันธุ์ ก็เป็นกระบวนการหนึ่งที่ดำเนินไปเพื่อหาจุดที่เหมาะสมที่สุด โดยสิ่งมีชีวิตที่มีพันธุกรรมเด่น และแข็งแรงจะมีแนวโน้มที่จะสามารถอาศัยอยู่ และดำรงเผ่าพันธุ์ต่อไปได้ ซึ่งกระบวนการสืบทอดจุดเด่นของแต่ละเผ่าพันธุ์นี้ ต้องอาศัยกระบวนการทางพันธุกรรม โดยสมาชิกกลุ่มที่แข็งแรงที่สุดของประชากรในแต่ละยุค (Generation) จะมีโอกาสรอด และเจริญเติบโต เพื่อจับคู่กันสร้างประชากรที่ดีกว่า ในยุคถัดไป

ด้วยแนวคิดนี้เอง จึงได้มีการคิดค้นกรรมวิธี ที่ใช้เลียนแบบการคัดเลือกตามธรรมชาติ ในการหาคำตอบที่เหมาะสมที่สุด เพื่อที่จะให้ได้ผลลัพธ์ที่ถูกต้อง และมีประสิทธิภาพสูง เราเรียกวิธีการนี้ว่า “เจเนติกอัลกอริธึม” (Genetic Algorithms: GAs) โดยได้รับการพัฒนาขึ้นโดย John Holland (1975) โดยมีวัตถุประสงค์สองอย่างคือ (1) เพื่อสรุปและดัดแปลงการใช้กระบวนการทางธรรมชาติให้ถูกต้องมากที่สุด และ (2) เพื่อออกแบบ และสร้างซอฟท์แวร์ที่รักษากลไกสำคัญของธรรมชาติ

เจเนติกอัลกอริธึม เป็นวิธีการหาคำตอบที่สามารถสำรวจลงไปได้ในทุกพื้นที่ ของความเป็นไปได้ของคำตอบอย่างทั่วถึง และดำเนินไปบนเส้นทางที่มีแนวโน้มจะประสบผลสำเร็จ ในขณะที่กำลังสุ่มตรวจสอบคำตอบอื่นๆ ไปด้วย เพื่อที่จะลดโอกาสที่จะติดอยู่กับจุดดีที่สุดท้องถิ่น (Local Optima) (Goldberg,1989) โดยหลักการแล้ว เจเนติกอัลกอริธึมมีกระบวนวิธีที่ง่าย และมีความสามารถในการคำนวณสูง ดังนั้นมันจึงเหมาะสำหรับงานที่มีความซับซ้อนในหลายๆ รูปแบบ ด้วยเหตุนี้เองที่ทำให้เจเนติกอัลกอริธึม แตกต่างจากการสุ่มแบบทั่วไป เนื่องจากเป็นการใช้ข้อมูลใน

อดีตอย่างมี ประสิทธิภาพ เพื่อพิจารณาจุดที่จะต้องค้นหาใหม่ โดยคาดหวังว่าสมรรถนะของการค้นหาจะดีขึ้น

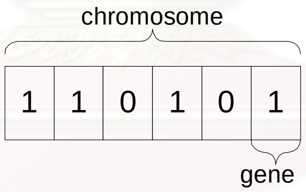
เจเนติกอัลกอริธึมแตกต่างกับวิธีค้นหาและการหาค่าเหมาะสมที่สุดด้วยวิธีอื่นๆ ดังนี้

- เจเนติกอัลกอริธึมไม่ได้ใช้ค่าของตัวแปรในการหาคำตอบโดยตรง แต่จะนำค่าเหล่านั้นมาเปลี่ยนเป็นรหัสก่อน แล้วจึงนำไปใช้ในกระบวนการหาคำตอบ

- การค้นหาเกิดขึ้นจากกลุ่มของข้อมูล (ประชาการ) ไม่ใช้เป็นการค้นหาจากตำแหน่งข้อมูลใดข้อมูลหนึ่ง เพียงจุดเดียว

- เจเนติกอัลกอริธึมเป็นวิธีที่อาศัยพื้นฐานของความน่าจะเป็น (Probabilistic) ไม่ได้หาคำตอบจากการคำนวณค่าจริงๆ (Deterministic)

ด้วยวิธีการนี้ ค่าของตัวแปรจะถูกเก็บไว้ในรูปของ ยีน (Gene) ที่อยู่บนโครโมโซม (Chromosome) ดังรูปที่ 3.1 โดยที่ตัวแปร 1 ตัวเป็น 1 โครโมโซม และมีค่าๆ หนึ่ง เมื่อมีการเปลี่ยนแปลงที่ยีน ณ ตำแหน่งใดๆ ภายในโครโมโซม ตัวแปรเดิมก็จะมีค่าเปลี่ยนไปเป็นค่าอื่นตามรหัสของตำแหน่งของยีนนั้นๆ



รูปที่ 3.1 รหัสที่ใช้ในเจเนติกอัลกอริธึม แทนตัวแปรที่มีค่าเท่ากับ 53

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ขั้นตอนหลักของเจเนติกอัลกอริธึมนั้น สามารถแบ่งได้ดังนี้

1. **ระบุฟังก์ชันวัตถุประสงค์ที่ต้องการ** ในที่นี้เราสมมุติให้เป็นฟังก์ชัน $f(x) = x^2 - 4x - 3$ ซึ่งเราต้องการที่จะหาค่าต่ำสุดของสมการ ในกรณีนี้เราทราบดีอยู่แล้วว่า ค่าต่ำสุดของสมการคือที่จุด $x = 2$ ซึ่ง จะได้ $f(x) = -7$

2. **การเข้ารหัสตัวแปร** เป็นการแปลงค่าพารามิเตอร์ให้อยู่ในรูปแบบที่เหมาะสมกับการทำงานภายใต้เงื่อนไขของ เจเนติกอัลกอริธึมนั่นคือ อยู่ในรูปแถวของตัวเลข (โครโมโซม) ที่มีความยาว (ยีน) แน่นอน โดยทั่วไปจะนิยมทำให้อยู่ในรูปของเลขฐานสอง (Binary Coding)

เช่น ถ้าต้องการที่จะหาค่าต่ำสุดของฟังก์ชันเป้าหมาย $f(x) = x^2 - 4x - 3$ โดยที่ x เป็นจำนวนเต็มที่กำหนดให้ มีค่าอยู่ระหว่าง [0,31] นั่นคือ จะมีสมาชิกจำนวนเต็มที่เป็นไปได้ 32 ตัว ซึ่ง เมื่อเข้ารหัสเป็นเลขฐานสอง จะต้องใช้สตริงที่มีความยาว 5 อักขระ (bit) คือจะมีค่าอยู่ในช่วง 00000 ถึง 11111

3. **การเลือกประชากรเริ่มต้น** จะมีกระบวนการที่เป็นแบบแผนในการเลือกประชากร (กลุ่มของคำตอบ) จากขอบเขตของคำตอบที่เป็นไปได้ เพื่อให้มีการกระจายตัวอย่างทั่วถึง เป็นการลดโอกาสที่จะติดอยู่กับค่าสูงสุดท้องถิ่นลง และหลังจากนั้นจะทำการคำนวณค่าความแข็งแรงของสมาชิกในกลุ่มนั้น ตามฟังก์ชันวัตถุประสงค์ที่ต้องการ สำหรับกระบวนการที่นิยมใช้ในการเลือกประชากรนั้น ก็เช่น วิธีวงล้อรูเลตต์ (Roulette Wheel) หรือ วิธีการเลือกแบบสุ่ม (Random) หรือการเลือกโดยการแข่งขัน (Tournament)

ในกรณีตัวอย่างข้างต้น ถ้าเรากำหนดให้มีการสุ่มจำนวนประชากรเพื่อใช้ในกระบวนการเป็นจำนวน 6 ตัวอย่าง โดยใช้วิธีสุ่ม จะได้ประชากรที่เมื่อคำนวณหาค่าความแข็งแรง ($f(x)$) ของประชากรทั้ง 6 ตัวแล้ว จะได้ผลลัพธ์ดังตารางที่ 3.1

ตารางที่ 3.1 ประชากรเริ่มต้นและค่าความแข็งแรง

ประชากร (ฐานสอง)	ประชากร (ฐานสิบ)	ความแข็งแรง	อันดับความแข็งแรง
00110	6	9	1
01010	10	57	2
11100	28	669	6
10001	17	218	3
10110	22	393	5
10010	18	249	4

ค่าเฉลี่ยของความแข็งแรงของประชากรทั้งหมดคือ 265.83

4. **การจับคู่เพื่อสร้างประชากรใหม่** โดยคู่ของบรรพบุรุษจะผ่านการเลือกมาจากกลุ่มของประชากรอีกครั้งหนึ่ง ซึ่งมักจะเป็นคู่ที่มีความแข็งแรงสูงสุด หรือ ผ่านเกณฑ์ที่ได้กำหนดไว้

หลังจากนั้นก็จะทำครอสโอเวอร์ (Crossover) นั่นคือมีการสลับยีนของพ่อและแม่ ซึ่งลูกที่ได้จากการครอสโอเวอร์นั้น ก็จะถูกนำไปหาค่าความแข็งแรง แล้วจึงแทนลงไปในประชากรเดิม โดยอาจจะแทนประชากรที่มีความแข็งแรงน้อยกว่า หรือแทนคู่ของพ่อกับแม่เลยก็ได้

จากตารางที่ 3.1 ข้างต้น 00110 และ 01010 เป็นประชากรที่มีความแข็งแรงสูงสุด 2 อันดับแรก จึงมีความเหมาะสมที่จะนำมาจับคู่เพื่อสร้างประชากรใหม่ เนื่องจากมีความน่าจะเป็นที่จะให้ผลผลิตที่ออกมามีความแข็งแรงมากขึ้น

ก่อนที่จะทำการครอสโอเวอร์ได้นั้น จะต้องเลือกตำแหน่งของจุดที่จะมีการสลับตำแหน่งเสียก่อน โดยสามารถจะสร้างเงื่อนไขในการเลือกได้หลายรูปแบบ และยังสามารถที่จะสลับโครโมโซมแบบหลายจุดได้ด้วย ในที่นี้ เราจะเลือกตำแหน่งที่ 3 เป็นจุดเริ่มของการสลับตำแหน่ง ซึ่งหลังจากการครอสโอเวอร์แล้ว จะได้ประชากรใหม่ดังตารางที่ 3.2

ตารางที่ 3.2 ประชากรที่ถูกเลือกขึ้นมาเพื่อทำการครอสโอเวอร์

ประชากรเดิม	ประชากรใหม่ (ฐานสอง)	ประชากรใหม่ (ฐานสิบ)	ความแข็งแรงของประชากรใหม่
00 : 110	00 : 010	2	-7
01 : 010	01 : 110	14	137

และถ้าเงื่อนไขในการแทนที่ประชากรเดิมคือ เลือกสองตัวที่แย่ที่สุดออก เราจะได้สมาชิกหลังการครอสโอเวอร์ เป็นดังตารางที่ 3.3

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 3.3 ประชากรหลังการครอสโอเวอร์

ประชากร (ฐานสอง)	ประชากร (ฐานสิบ)	ความแข็งแรง	อันดับความแข็งแรง
00110	6	9	2
01010	10	57	3
00010	2	-7	1
10001	17	218	5
01110	14	137	4
10010	18	249	6

ค่าเฉลี่ยของความแข็งแรงของประชากรทั้งหมดคือ 110.5

จะเห็นว่ามีการปรับปรุงค่าเป้าหมายของกลุ่มตัวอย่าง ไปในทิศทางที่ดีขึ้น

5. การกลายพันธุ์ (Mutation) ในสภาวะธรรมชาติ ประชากรที่มีความแข็งแรงน้อย จะมีโอกาสกลายพันธุ์ได้ เมื่ออยู่ในเงื่อนไขที่เอื้ออำนวย นั่นคือมีความน่าจะเป็นของการกลายพันธุ์ในขณะนั้น สูงเกินกว่าค่าที่ตั้งไว้ ดังนั้นปัจจัยของการ กลายพันธุ์จึงมี 2 กรณี คือ (1) ประชากรมีความแข็งแรงน้อย และ (2) ความน่าจะเป็นของการกลายพันธุ์สูงกว่าค่าที่กำหนด การกลายพันธุ์สามารถเกิดขึ้นได้โดยอาศัยประชากรเพียง 1 ตัว โดยจะมีการ กลายพันธุ์ ณ ตำแหน่งของยีนที่จะถูกกำหนดอีกครั้งหนึ่ง (กำหนดตำแหน่งแบบสุ่ม) หลังจากที่ได้คำนวณหาค่าความแข็งแรงแล้ว ประชากรใหม่นี้ ก็จะถูกแทนกลับไปในกลุ่ม โดยอาศัยเงื่อนไขเช่นเดียวกับการครอสโอเวอร์ หรือกำหนดเงื่อนไขที่ต่างกันก็ได้

วิธีอย่างง่ายของการกลายพันธุ์คือ เมื่อกำหนดตำแหน่งขึ้นมาหนึ่งจุด บนแถวอักขระของโครโมโซม ตัวเลข ณ ตำแหน่งนั้น จะเปลี่ยนเป็นรหัสตรงข้ามกัน คือ จาก 0 เป็น 1 และในทางกลับกันคือเปลี่ยนจาก 1 เป็น 0

จากตัวอย่างที่ผ่านมา ถ้ากำหนดว่าประชากรที่มีความแข็งแรงมากกว่า 200 มีโอกาสที่จะกลายพันธุ์ได้ จะมีสมาชิก 2 ตัวที่เข้าเกณฑ์นี้ คือ 10001 และ 10010 ถ้าเลือกให้ 10010 มี

โอกาสที่จะกลายพันธุ์ได้ ณ ตำแหน่งที่ 4 ของสตริง จะได้ว่า 10010 จะเปลี่ยนเป็น 10000 และจะมีสมาชิกในกลุ่มเป็นดังตารางที่ 3.4

ตารางที่ 3.4 ประชากรหลังจากการกลายพันธุ์

ประชากร (ฐานสอง)	ประชากร (ฐานสิบ)	ความแข็งแรง	อันดับความแข็งแรง
00110	6	9	2
01010	10	57	3
00010	2	-7	1
10001	17	218	6
01110	14	137	4
10000	16	189	5

ค่าเฉลี่ยของความแข็งแรงของประชากรทั้งหมดคือ 100.5

จะเห็นว่ากลุ่มของประชากรที่เหลืออยู่ หลังจากดำเนินการตามขั้นตอนการกลายพันธ์ของเจเนติกอัลกอริธึม มีแนวโน้มที่จะเคลื่อนเข้าใกล้คำตอบมากขึ้น ซึ่งเป็นการแสดงให้เห็นว่าการประยุกต์ใช้เจเนติกอัลกอริธึม ในการแก้ปัญหาที่เกี่ยวกับการหาค่าที่เหมาะสมที่สุดนั้น สามารถนำไปใช้ได้จริง

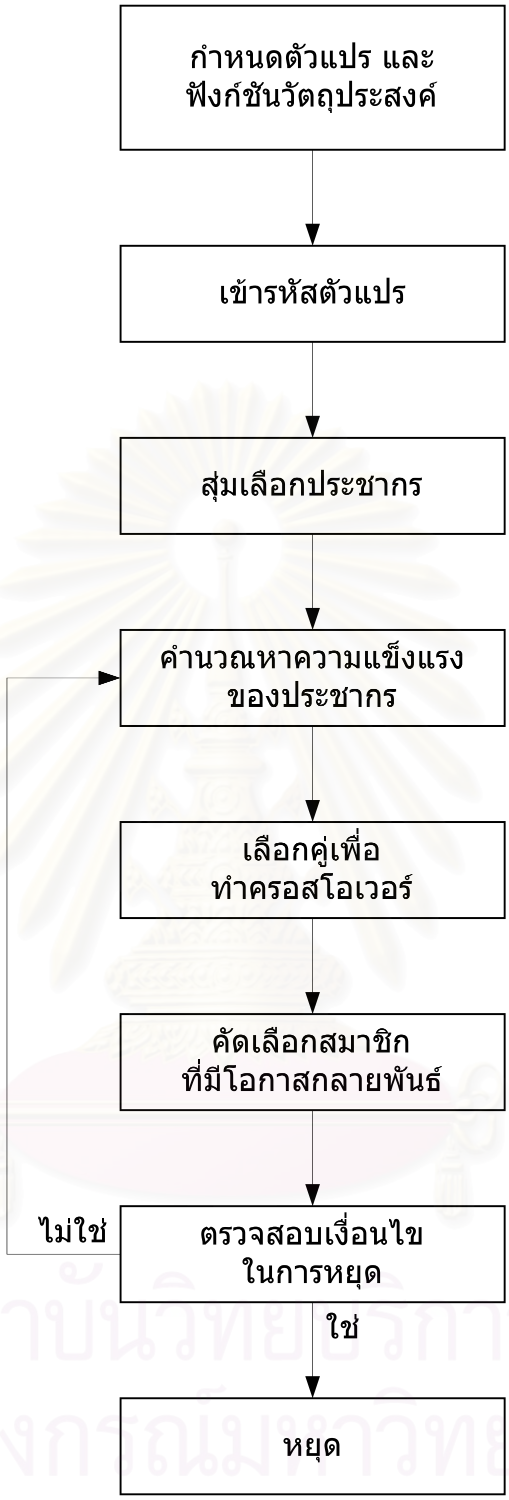
6. **สมาชิกที่เป็นคำตอบของปัญหา** จะเป็นสมาชิกที่มีความแข็งแรงสูงสุดในรุ่นสุดท้าย โดยมีเงื่อนไขในการหยุด อย่างใดอย่างหนึ่งในสองเงื่อนไขคือ การคำนวณจะหยุดลงเมื่อ

6.1 จำนวนรอบของรุ่น (Generation) มีค่ามากกว่าค่าที่ตั้งไว้

6.2 ประชากรที่มีความแข็งแรงสูงสุด (คำตอบที่เหมาะสมที่สุด) มีทศนิยมต่างกันน้อยกว่าที่กำหนดไว้

ดังจะเป็นไปตามแผนภูมิในรูปที่ 3.2

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.2 แผนภูมิแสดงกระบวนวิธีในการหาคำตอบที่เหมาะสมที่สุดด้วยการใช้เจเนติกอัลกอริธึม

บทที่ 4

ระบบการควบคุมชนิดปรับค่าเองได้

4.1 บทนำ

อุตสาหกรรมการผลิตทั่วไปประกอบไปด้วยหน่วยปฏิบัติการ (Unit Operations) ต่างๆ เป็นจำนวนมาก ซึ่งหน่วยปฏิบัติการเหล่านี้จะทำหน้าที่เฉพาะอย่าง เช่น หน่วยที่เพิ่มหรือลดความดัน หน่วยที่เพิ่มหรือลดอุณหภูมิ หน่วยที่เกิดปฏิกิริยา หน่วยที่ทำการแยกหรือรวมสาร ฯลฯ หากกำหนดขอบเขตของการเปลี่ยนแปลงที่สนใจอย่างชัดเจน เช่น การเปลี่ยนแปลงของวัตถุดิบไปเป็นผลิตภัณฑ์ ขอบเขตการเปลี่ยนแปลงนี้มักจะประกอบไปด้วยหน่วยปฏิบัติการมากกว่าหนึ่งหน่วยขึ้นไป จะเรียกขอบเขตที่ประกอบไปด้วยหน่วยปฏิบัติการนี้ว่า กระบวนการ (Process) หรือ ระบบ (System)

เมื่อต้องการศึกษาถึงปรากฏการณ์ต่างๆ ที่เกิดขึ้นภายในกระบวนการ จะพบว่ามีตัวแปรที่สามารถบอกถึงสภาวะของระบบ เข้ามาเกี่ยวข้องกับระบบที่พิจารณาด้วยเสมอ ตัวแปรที่สำคัญสามารถแยกออกได้เป็นสองประเภทคือ (1) อินพุท (2) เอาท์พุท ซึ่งตัวแปรเหล่านี้จะมีการเปลี่ยนแปลงได้ตามเวลา

อินพุท (Input) คือตัวแปรภายนอกที่สามารถส่งผลต่อการเปลี่ยนแปลงของพฤติกรรมของระบบ โดยที่ระบบหนึ่งๆ สามารถมีอินพุทได้มากกว่าหนึ่งตัว อินพุทที่เราสามารถควบคุมได้ จะเรียกว่า ตัวแปรปรับ (Manipulated Variables) ส่วนอินพุทที่เราไม่สามารถควบคุมได้ เรียกว่า ตัวแปรรบกวน (Disturbance Variables)

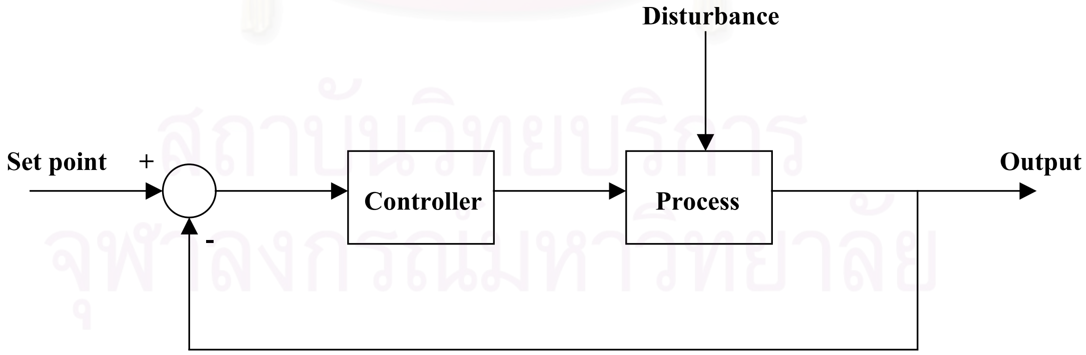
เอาท์พุท (Output) คือตัวแปรที่แสดงการตอบสนองของระบบที่มีต่อการเปลี่ยนแปลงของอินพุท ซึ่งในส่วนของเอาท์พุทที่เราต้องการควบคุมจะเรียกว่าเป็น ตัวแปรควบคุม (Controlled Variables) และจะต้องสามารถวัดค่าได้

การควบคุมกระบวนการเป็นความต้องการที่จะควบคุม “ตัวแปรควบคุม” ที่เราสนใจ ด้วยการปรับ “ตัวแปรปรับ” ที่สำคัญบางตัว โดยที่ตัวแปรทั้งสองชุดนี้จะมีความสัมพันธ์เกี่ยวข้องกันหรือมีความควบคุมได้ (Controllability) ซึ่งรูปแบบการควบคุมที่เป็นที่นิยมใช้กันแพร่หลายในกระบวนการทางอุตสาหกรรมคือ การควบคุมแบบป้อนกลับ (Feedback Control)

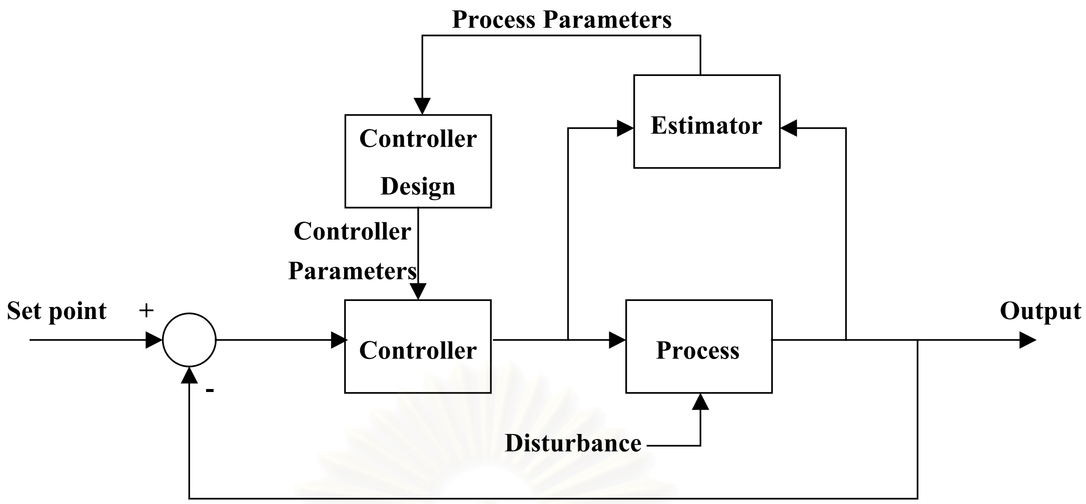
หลักของกระบวนการควบคุมแบบป้อนกลับทั่วๆ ไป คือ มีการใช้ค่าความผิดพลาด (Error) ซึ่งเป็นผลต่างของค่าเป้าหมายที่ต้องการ (Setpoint) กับค่าตัวแปรควบคุมในกระบวนการที่ได้จากการวัด มาผ่านอัลกอริธึมทางคณิตศาสตร์ เพื่อทำการคำนวณหาเอาท์พุทของตัวควบคุม (Controller Output) ซึ่งอัลกอริธึมทางคณิตศาสตร์ที่ใช้ จะมีความซับซ้อน และแตกต่างกันไปตามรูปแบบและชนิดของตัวควบคุมนั้นๆ

ถึงแม้ว่าตัวควบคุมแบบป้อนกลับทั่วๆ ไป เช่น ตัวควบคุมพีไอดี จะเป็นที่นิยมกันอย่างกว้างขวาง เพราะมีความทนทาน (Robustness) สูง และสามารถทำความเข้าใจได้ง่าย แต่เนื่องจากข้อจำกัดในหลายด้านของตัวควบคุมแบบป้อนกลับเหล่านั้น จึงได้มีการพัฒนาตัวควบคุมที่มีความซับซ้อนเพิ่มขึ้น เพื่อให้สามารถรองรับความต้องการ ที่มีต่อประสิทธิภาพในการควบคุมกระบวนการที่สูงขึ้นได้

การควบคุมแบบปรับตัวได้ (Adaptive Control) เป็นหนึ่งในการพัฒนาตัวควบคุมให้มีประสิทธิภาพสูงขึ้น เป็นการควบคุมที่สามารถปรับการตอบสนอง ให้เข้ากับความเปลี่ยนแปลงที่เกิดขึ้นกับลักษณะของตัวแปรรบกวน หรือความเปลี่ยนแปลงเนื่องจากกระบวนการที่เป็นพลวัติได้ ข้อแตกต่างที่เห็นได้ชัดระหว่างตัวควบคุมแบบปรับตัวได้ กับตัวควบคุมแบบป้อนกลับธรรมดานั่นคือตัวควบคุมแบบปรับตัวได้จะมีการปรับค่าพารามิเตอร์ในตัวเอง และมีกลไกในการหาค่าพารามิเตอร์ที่เหมาะสมนั้น ซึ่งตัวควบคุมแบบป้อนกลับทั่วๆ ไป จะมีค่าพารามิเตอร์คงที่ สังเกตความแตกต่างระหว่างการควบคุมแบบป้อนกลับธรรมดา (รูปที่ 4.1) กับการควบคุมแบบปรับตัวได้ (รูปที่ 4.2)



รูปที่ 4.1 การควบคุมแบบป้อนกลับทั่วไป (Typical Feedback Control)



รูปที่ 4.2 การควบคุมแบบปรับตัวได้ (Adaptive Control)

จะเห็นได้ว่าระบบการควบคุมแบบปรับตัวได้ประกอบไปด้วยสองวง โดยวงแรกมีลักษณะเช่นเดียวกันกับวงป้อนกลับ ส่วนอีกวงหนึ่งที่เพิ่มเติมขึ้นมาจากระบบธรรมดา จะเป็นวงของการปรับค่าพารามิเตอร์ โดยทั่วไปวงของการปรับค่า จะดำเนินไปช้ากว่าวงของการป้อนกลับ [1]

งานวิจัยนี้มุ่งเน้นไปที่การศึกษาตัวควบคุมชนิดปรับค่าเองได้ (Self-tuning Controller) ซึ่งเป็นตัวควบคุมหนึ่งในกลุ่มของการควบคุมแบบปรับตัวได้ ซึ่งลักษณะของตัวควบคุมรูปแบบนี้จะประกอบไปด้วย 3 ส่วนหลัก นั่นคือ

- การหาเอกลักษณ์ (Identification) เป็นการสร้างความสัมพันธ์ทางคณิตศาสตร์ขึ้นมาเพื่อที่จะได้ผลลัพธ์ที่เป็นตัวแทนของกระบวนการ

- การออกแบบตัวควบคุม (Controller Design) เพื่อให้ได้ค่าพารามิเตอร์ที่เหมาะสมกับกระบวนการที่จำลองได้ ภายใต้เงื่อนไขการออกแบบที่ได้กำหนดไว้

- การนำไปใช้ (Implementation) จะนำพารามิเตอร์ที่ได้จากการออกแบบ มาปรับปรุงค่าพารามิเตอร์ของตัวควบคุม เพื่อให้ได้การตอบสนองที่เป็นไปตามวัตถุประสงค์ที่ต้องการ

โดยที่จะใช้โครงสร้างของตัวควบคุมพีไอดีเป็นฐานในการคำนวณหาเอาท์พุทของตัวควบคุม

4.2 การหาเอกลักษณ์ของกระบวนการ (Process Identification)

การหาเอกลักษณ์ของกระบวนการ (Process Identification) เป็นกรรมวิธีในการจำลองระบบพลวัติขึ้นมาจากการใช่ข้อมูลอินพุท และเอาท์พุทของกระบวนการที่ได้จากการทดลอง ระบบที่ได้จากการหาเอกลักษณ์ จะแสดงได้โดยการเขียนในรูปของแบบจำลอง (Model) เพื่อที่จะแสดงให้เห็นถึงความสัมพันธ์ของตัวแปรอินพุทและเอาท์พุทที่ได้มานั้น แบบจำลองของระบบมักจะอยู่ในรูปของแบบจำลองทางคณิตศาสตร์ ซึ่งอาศัยพื้นฐานมาจากการหาสมดุลมวล และสมดุลพลังงาน ทำให้แบบจำลองทางคณิตศาสตร์ที่ได้นั้น อยู่ในรูปของสมการผลต่าง หรือสมการอนุพันธ์ (Difference or Differential Equation) เนื่องจากแบบจำลองทางคณิตศาสตร์ในรูปแบบนี้ มีความเหมาะสมต่อการวิเคราะห์ ทำนาย และออกแบบระบบทางพลวัติ

แต่สำหรับกระบวนการที่มีระบบซับซ้อน เช่นกระบวนการที่มีความไม่เป็นเชิงเส้น (Nonlinear Process) การที่จะหาแบบจำลองทางคณิตศาสตร์ที่ถูกต้องเป็นไปได้ยาก ดังนั้นจึงต้องประมาณระบบด้วยแบบจำลองที่มีความซับซ้อนน้อยลง เช่น แบบจำลองเชิงเส้นอันดับหนึ่ง (First Order Linear Model) ถึงแม้ว่าการประมาณระบบเช่นนี้จะให้ค่าการประมาณที่มีความผิดพลาดจากระบบจริงอยู่บ้าง แต่ก็ทำให้ระบบมีความซับซ้อน และยุ่งยากน้อยลง และทำให้การประมาณค่าเป็นได้อย่างรวดเร็ว และง่ายมากขึ้น

โดยทั่วไปแล้วขั้นตอนในการประมาณค่า เพื่อหาเอกลักษณ์ของระบบ จะเป็นไปตามลำดับดังนี้

4.2.1 กระตุ้นระบบ (Exciting the Process) เป็นการเปลี่ยนแปลงค่าอินพุทของกระบวนการในรูปแบบต่างๆ แล้วสังเกตุพฤติกรรมของเอาท์พุทที่เป็นผลมาจากอินพุทที่เปลี่ยนแปลงไปนั้นเทียบกับเวลาที่เปลี่ยนแปลงไป

4.2.2 เลือกแบบจำลองทางคณิตศาสตร์ที่เหมาะสม โดยจะมีพารามิเตอร์ที่ต้องการทราบค่า มากน้อยตามอันดับและความซับซ้อนของระบบ ซึ่งตัวแปรที่ต้องการทราบค่านี้ จะเป็นสัมประสิทธิ์ของสมการอนุพันธ์ที่ได้เลือกมา

4.2.3 ใช้กระบวนการในการหาค่าที่เหมาะสมที่สุด (Optimization) เพื่อที่จะหาค่าพารามิเตอร์ของระบบในข้อ 4.2.2 ที่ทำให้ข้อมูลจากข้อ 4.2.1 มีความผิดพลาดจากแบบจำลองของระบบให้น้อยที่สุด

4.2.4 ทดสอบระบบที่ได้ เพื่อหาความน่าเชื่อถือของแบบจำลอง (Model Validation)

ในงานวิจัยนี้ ได้ใช้การจำลองกระบวนการ โดยการประมาณให้อยู่ในรูปของแบบจำลองเชิงเส้น คือในรูปของกระบวนการอันดับที่หนึ่งที่มีเวลาหน่วง (First Order with Dead Time) ซึ่งนำมาใช้ร่วมกับข้อมูลที่จะได้จากการกระตุ้นกระบวนการ ด้วยการเปลี่ยนแปลงค่าของตัวแปรอินพุทแบบขั้น (Step Input) โดยใช้กระบวนการทำซ้ำกำลังสองน้อยที่สุด (Recursive Least Square, RLS) ซึ่งเหมาะสมที่จะใช้งานกับแบบจำลองกระบวนการที่มีค่าพารามิเตอร์เป็นเชิงเส้น ซึ่งพารามิเตอร์ของกระบวนการที่ได้ จะถูกนำไปใช้ ในการออกแบบตัวควบคุม ชนิดปรับค่าเองได้ตามที่ต้องการ

พิจารณาระบบที่มีทรานสเฟอร์ฟังก์ชัน (Transfer Function)

$$G(s) = \frac{K_p}{\tau s + 1} e^{-\theta s} \tag{4.1}$$

ซึ่งสามารถเขียนให้อยู่ในรูปของสมการอนุพันธ์อันดับหนึ่งได้ดังนี้

$$\tau \frac{dy(t)}{dt} + y(t) = Ku(t-\theta) \tag{4.2}$$

เมื่อทำการประมาณค่าอนุพันธ์ด้วยสมการผลต่าง (Difference Equation) แล้วจะได้ว่า

$$\tau \frac{y(t)-y(t-1)}{\Delta t} + y(t-1) = Ku(t-1-\theta)$$

$$y(t) = \left(1 - \frac{\Delta t}{\tau}\right) y(t-1) + \frac{K\Delta t}{\tau} u(t-1-\theta)$$

$$y(t) = \begin{bmatrix} y(t-1) & u(t-1-\theta) \end{bmatrix} \begin{bmatrix} 1 - \frac{\Delta t}{\tau} \\ \frac{K\Delta t}{\tau} \end{bmatrix}$$

$$y(t) = \psi^T (t-1) \hat{\theta}(t-1) \tag{4.3}$$

เมื่อ

$$\hat{\theta}^T (t-1) = \begin{bmatrix} 1 - \frac{\Delta t}{\tau} & \frac{K\Delta t}{\tau} \end{bmatrix} \tag{4.4}$$

$$\psi^T (t-1) = [y(t-1) \quad u(t-1-\theta)] \tag{4.5}$$

โดยจะเรียก $\hat{\theta}(t-1)$ ว่าเวคเตอร์พารามิเตอร์ (Parameter Vector) และเรียก $\psi(t-1)$ ว่าเวคเตอร์ รีเกรสชัน (Regression Vector) หรือเวคเตอร์ข้อมูล (Data Vector)

อัลกอริธึมของการทำซ้ำกำลังสองน้อยที่สุด จะเป็นไปตามสมการด้านล่าง

$$P(t-1)=\frac{1}{\lambda}[\mathbf{I}-K(t-1)\psi^{T}(t-2)]P(t-2) \tag{4.6}$$

$$K(t)=\frac{P(t-1)\psi(t-1)}{\lambda+\psi^{T}(t-1)P(t-1)\psi(t-1)} \tag{4.7}$$

$$\hat{\theta}(t)=\hat{\theta}(t-1)+K(t)[\Delta y(t)-\psi^{T}(t-1)\hat{\theta}(t-1)] \tag{4.8}$$

เมื่อ *P* เป็นเมตริกซ์ความแปรปรวนร่วม (Covariance Matrix), *K(t)* เป็นส่วนขยายของตัวกรองคาลมาน (Kalman Filter Gain) ส่วน $\mathbf{I}$ เป็นเมตริกซ์หนึ่งหน่วยที่มีมิติเท่ากับ $(n_a+n_b)\times(n_a+n_b)$ โดยที่ในการคำนวณ จะต้องมีการประมาณค่าเริ่มต้นของ *P(0)* และค่าเริ่มต้นของ $\hat{\theta}(0)$

ในที่นี้ λ จะเป็นตัวประกอบการลืมค่าตัวแปร (Variable Data Forgetting Factor) ซึ่งจะมีค่า $0<\lambda\leq 1$, ถ้า λ มีค่าเข้าใกล้ 1 จะเกิดการเปลี่ยนแปลงในการประมาณค่าที่ช้ากว่า λ ที่มีค่าเข้าใกล้ 0 โดยจะสามารถคำนวณค่า λ ได้จากสมการดังนี้

$$\lambda(t)=\frac{n(t)+\sqrt{(n(t))^{2}+4w(t)}}{2} \tag{4.9}$$

เมื่อ

$$n(t)=1-w(t)-(e(t))^{2}/\sigma_{0}$$

$$w(t)=\psi^{T}(t-1)P(t-1)\psi(t-1)$$

และ σ_0 เป็นพารามิเตอร์ปรับแต่งที่มีค่าอยู่ในช่วง 0.1 – 10 และ *e(t)* เป็นความผิดพลาดในการประมาณค่าซึ่งเป็นไปตามสมการ

$$e(t)=\Delta y(t)-\psi^{T}(t-1)\hat{\theta}(t-1) \tag{4.10}$$

ในขั้นตอนสุดท้ายของ RLS คือ ต้องหาค่าที่เหมาะสมที่สุดของเวคเตอร์พารามิเตอร์ $\hat{\theta}$ ที่สอดคล้องกับฟังก์ชันเป้าหมาย นั่นคือหาค่าน้อยที่สุดของสมการ (4.9)

$$J(\theta)=\sum_{t=1}^{N}\lambda^{N-t}[\psi^{T}(t-1)\theta-\Delta y(t)]^{2} \tag{4.11}$$

4.3 การออกแบบตัวควบคุม (Controller Design)

ในที่นี้เราจะกำหนดรูปแบบของตัวควบคุมให้เป็นแบบตัวควบคุมพีไอดี (PID controller) ซึ่งสามารถเขียนให้อยู่ในรูปแบบความเร็ว (Velocity Form) ได้ดังนี้

$$\Delta u(t) = K_c\left[(1+\frac{T_s}{\tau_I}+\frac{\tau_D}{T_s}) - (1+2\frac{\tau_D}{T_s})z^{-1} + (\frac{\tau_D}{T_s})z^{-2}\right]\varepsilon(t) \quad (4.12)$$

เนื่องจากตัวควบคุมพีไอดีในรูปแบบนี้สามารถกำจัดผลของรีเซ็ทไวด์อัพ (Reset Wide-up) ได้

การใช้ตัวควบคุมนี้เพื่อควบคุมกระบวนการที่ต้องการ จะต้องมีการคำนวณค่าพารามิเตอร์ 3 ค่าคือ K_c, τ_I, τ_D ซึ่งต้องอาศัยค่าพารามิเตอร์โดยประมาณของกระบวนการ (K_p, τ, θ) ที่ได้จากขั้นตอนของการสร้างแบบจำลอง จากสมการที่ (4.4) หากกำหนดให้

$$\hat{\theta}^T(t-1) = \begin{bmatrix} a & b \end{bmatrix} \quad (4.13)$$

ซึ่งสามารถเขียนเป็นความสัมพันธ์กับค่าพารามิเตอร์ของกระบวนการได้ดังนี้

$$K_p = \frac{b}{1-a} \quad (4.14)$$

$$\tau = \frac{T_s}{1-a} \quad (4.15)$$

$$\theta = kT_s \quad (4.16)$$

เมื่อ $T_s = \Delta t$ เป็นระยะเวลาในการเก็บตัวอย่าง (Sampling Time)

การคำนวณหาค่าพารามิเตอร์ของตัวควบคุมพีไอดี โดยอาศัยค่าพารามิเตอร์ของแบบจำลองกระบวนการในรูปแบบกระบวนการอันดับหนึ่งและมีเวลาหน่วง สามารถทำได้ในหลายๆ วิธี แต่ในที่นี้จะยกตัวอย่าง กรณีของการคำนวณตามกฎการปรับค่าตัวควบคุมพีไอดีของ Ziegler-Nichols ดังตารางที่ 4.1

ตารางที่ 4.1 กฎการปรับค่าตัวควบคุมพีไอดีตามหลักของ Ziegler-Nichols

ชนิดตัวควบคุม	K_c	τ_I	τ_D
P	$\frac{\tau}{K\theta}$	-	-
PI	$\frac{0.9\tau}{K\theta}$	3.33θ	-
PID	$\frac{1.2\tau}{K\theta}$	2.0θ	0.5θ

สามารถค้นคว้าถึงรูปแบบการปรับค่าอื่นๆ ได้จากภาคผนวก ข.

4.4 การนำไปใช้ในกระบวนการ (Implementation)

ในการควบคุมแบบปรับตัวได้นั้นจะมีขั้นตอนต่างๆ ในลูพของการปรับปรุงค่าในช่วงระหว่างเวลาของการเก็บตัวอย่าง ดังนี้

4.4.1 เก็บข้อมูลอินพุท-เอาท์พุทของกระบวนการ โดยข้อมูลที่เก็บคือ ตัวแปรขาเข้า *u(t)* และขาออก *y(t)* จากกระบวนการ

4.4.2 นำข้อมูลที่ได้จาก 4.4.1 ทั้งในอดีตและปัจจุบัน ไปใช้ในการคำนวณหาค่าพารามิเตอร์ของแบบจำลองของกระบวนการที่ให้ค่าผิดพลาดของแบบจำลองที่น้อยที่สุด

4.4.3 หาค่าพารามิเตอร์ของตัวควบคุม (K_C, τ_I, τ_D)

4.4.4 แทนค่าพารามิเตอร์ที่ได้ ในการควบคุมกระบวนการในรอบการควบคุมถัดไป

4.4.5 เปรียบเทียบผลของการตอบสนองป้อนกลับลูพปิดที่ได้ ว่าค่าตัวแปรควบคุมที่ต้องการลู่เข้าสู่ค่าเป้าหมายหรือไม่

4.4.6 เมื่อค่าที่ได้ใน 4.4.5 มีความเบี่ยงเบนเกินจากที่กำหนดไว้ หรือ เมื่อครบรอบเวลาของการปรับปรุงค่า ก็จะเริ่มคำนวณตามขั้นตอนต่างๆ ตามที่กล่าวมาแล้วอีกครั้งหนึ่ง

4.5 อัลกอริธึมของตัวควบคุมพีไอดีชนิดปรับค่าเองได้ที่ใช้ในงานวิจัยนี้

เนื่องจากการความสามารถ ในการคำนวณหาจุดที่เหมาะสมที่สุด ของความสัมพันธ์ใดๆ โดยการใช้เจเนติกอัลกอริธึม สามารถที่จะหาประมาณของกระบวนการ หรือค่าพารามิเอตร์ของสามการที่ (4.2) ได้โดยตรง ดังนั้นจึงทำให้กระบวนการประมาณค่าพารามิเตอร์ของกระบวนการอันดับหนึ่งที่มีเวลาหน่วง ที่เกิดขึ้นในงานวิจัยนี้ สามารถหาค่า K_p, τ, θ ได้โดยที่ไม่ต้องคำนวณผ่านตัวแปร *a* และ *b* ดังที่แสดงในสมการที่ (4.13) โดยที่ยังมีการใช้ฟังก์ชันวัตถุประสงค์คือ กำลังสองน้อยที่สุดอยู่ และมีการใช้ค่าที่คำนวณได้ในอดีตมาใช้คำนวณซ้ำในรอบการคำนวณปัจจุบันอีกด้วย

นอกจากนี้ ในขั้นตอนของการหาค่าพารามิเตอร์ของตัวควบคุมพีไอดี โดยการใช้วิธีของเจเนติกอัลกอริธึมเพื่อหาค่าที่จะให้ได้ผลของการควบคุมที่ดีที่สุดนั้น สามารถทำได้โดยตรง โดยไม่ต้องอาศัยความสัมพันธ์ของ Ziegler-Nichols หรือความสัมพันธ์ใดๆ

ซึ่งในขั้นตอนนี้ จะใช้การสุ่มเซ็ตของค่าพารามิเตอร์ (K_c, τ_I, τ_D) ขึ้นมาหนึ่งกลุ่ม เป็นประชาสกรเริ่มต้นแล้ว ทำการคำนวณผลของการควบคุมด้วยเซ็ตของพารามิเตอร์เหล่านั้น เพื่อให้ได้ผลของค่าอินทีกรัลของค่าผิดพลาดสัมบูรณ์ (Integral Absolute Error; IAE) ซึ่งจะนำมาใช้เป็นค่าความแข็งแรง (Fitness) ของประชากรนั้นๆ หลังจากนั้นจึงใช้กระบวนวิธีของเจเนติกอัลกอริธึมเพื่อหาค่าที่ดีที่สุด นั่นคือให้ IAE ในการควบคุมกระบวนการที่ได้จากการประมาณต่ำที่สุด และนำค่านั้นไปใช้ในการควบคุมเป็นขั้นตอนสุดท้าย

สามารถสรุปขั้นตอนการปรับค่าพารามิเตอร์ของตัวควบคุมที่นำเสนอได้ดังนี้

4.5.1 เก็บข้อมูลอินพุท-เอาท์พุทของกระบวนการ โดยข้อมูลที่เก็บคือ ตัวแปรขาเข้า *u(t)* และขาออก *y(t)* จากกระบวนการ

4.5.2 นำข้อมูลที่ได้จาก 4.4.1 ทั้งในอดีตและปัจจุบัน ไปใช้ในการคำนวณหาค่าพารามิเตอร์ของแบบจำลองของกระบวนการที่ให้ค่าผิดพลาดของแบบจำลองที่น้อยที่สุด โดยใช้ค่ากำลังสองน้อยที่สุดเพื่อเป็นฟังก์ชันวัตถุประสงค์ในการประมาณค่าพารามิเตอร์ และเจเนติกอัลกอริธึมในการหาค่าที่เหมาะสมที่สุด

4.5.3 หาค่าพารามิเตอร์ของตัวควบคุม (K_c, τ_I, τ_D) ที่ให้ค่า IAE ต่ำที่สุด เมื่อทดสอบด้วยการควบคุมกระบวนการอันดับหนึ่งที่ได้จากการประมาณค่าในข้อ 4.4.2

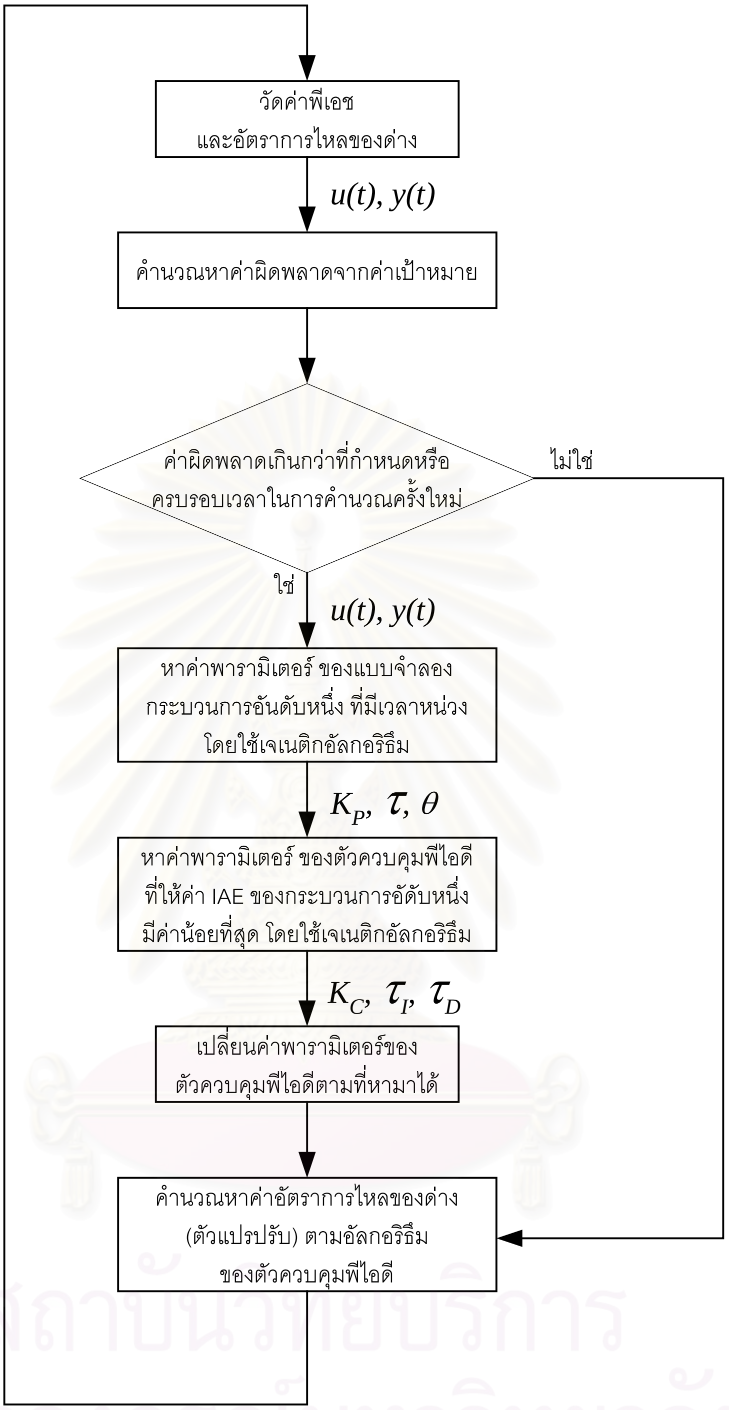
4.5.4 แทนค่าพารามิเตอร์ที่ได้ ในการควบคุมกระบวนการในรอบการควบคุมถัดไป

4.5.5 เปรียบเทียบผลของการตอบสนองป้อนกลับลูพปิดที่ได้ ว่าค่าตัวแปรควบคุมที่ต้องการลู่เข้าสู่ค่าเป้าหมายหรือไม่

4.5.6 เมื่อค่าที่ได้ใน 4.4.5 มีความเบี่ยงเบนเกินจากที่กำหนดไว้ หรือ เมื่อครบรอบเวลาของการปรับปรุงค่า ก็จะเริ่มคำนวณตามขั้นตอนต่างๆ ตามที่กล่าวมาแล้วอีกครั้งหนึ่ง

พิจารณาแผนผังการปรับค่าพารามิเตอร์ได้ในรูปที่ 4.3

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



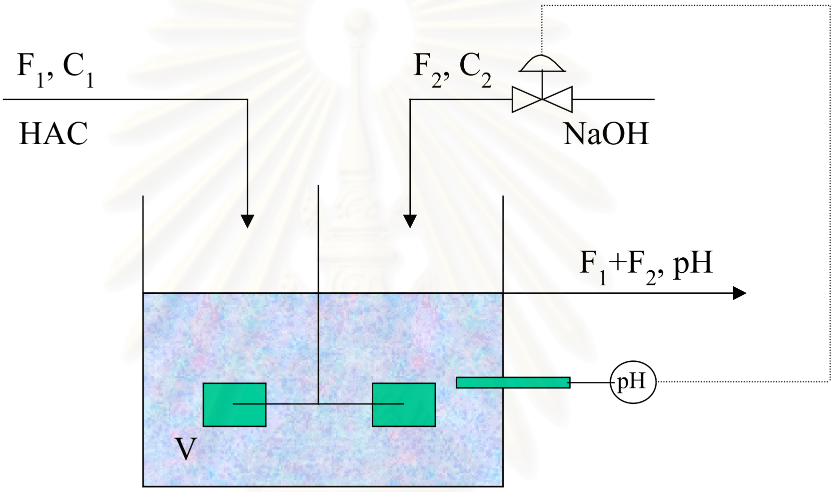
รูปที่ 4.3 แผนภูมิแสดงการควบคุมกระบวนการปรับค่าพีเอชให้เป็นกลาง ด้วยตัวควบคุมพีไอดีชนิดปรับค่าเองได้ ที่ใช้เจเนติกอัลกอริธึมในการหาค่าที่เหมาะสมที่สุด

บทที่ 5

กระบวนการปรับค่าพีเอชให้เป็นกลาง

5.1 ภาพรวม

พิจารณาระบบภายในเครื่องปฏิกรณ์แบบกวนต่อเนื่อง (Continuous Stirred Tank Reactor; CSTR) ที่มีปริมาตรคงที่ และมีการรักษาอุณหภูมิให้คงที่ตลอดเวลาดังรูปที่ 5.1



รูปที่ 5.1 เครื่องปฏิกรณ์แบบกวนต่อเนื่อง (CSTR)

ในการทำงานวิจัยครั้งนี้ เลือกใช้พลวัติของระบบการปรับค่าพีเอชให้อยู่ในภาวะที่เป็นกลาง ดังที่ได้มีการนำเสนอไว้ในงานวิจัยของ McAvoy et al. (1972) ที่มีสมมุติฐานคือ

- ระบบเป็นเครื่องปฏิกรณ์แบบถังกวนต่อเนื่องในอุดมคติ (Ideal CSTR)
- ปริมาตรของเครื่องปฏิกรณ์คงที่
- ปฏิกิริยาของกรดและด่างเกิดขึ้นอย่างรวดเร็ว และสมบูรณ์
- ละเลยเวลาหน่วง (Time Delay) ที่มีในระบบท่อ
- เครื่องปฏิกรณ์เป็นระบบที่มีอุณหภูมิคงที่

เนื่องจากปฏิกิริยาที่เกิดในระบบเครื่องปฏิกรณ์ จะพิจารณาได้ง่ายมากขึ้น โดยที่กำหนดให้ โซเดียมอิออนแทนด้วยตัวแปร $\zeta = [Na^{+}]$ และอะซีเตทอิออนแทนด้วยตัวแปร $\xi = [HAC] + [AC^{-}]$

ซึ่งเมื่อทำการดุลสมการขององค์ประกอบของสารต่างๆ จะได้ชุดสมการดังนี้

ดุลองค์ประกอบอะซีเตท

$$V\frac{d\xi}{dt} = F_1C_1 - (F_1 + F_2)\xi \tag{5.1}$$

ดุลองค์ประกอบโซเดียม

$$V\frac{d\zeta}{dt} = F_2C_2 - (F_1 + F_2)\zeta \tag{5.2}$$

สมดุลของกรดอะซิติก

$$K_a = \frac{[AC^{-}][H^{+}]}{[HAC]} \tag{5.3}$$

ที่ 25 °C, $K_a = 1.978 \times 10^{-5}$

สมดุลการแตกตัวของน้ำ

$$K_w = [H^{+}][OH^{-}] \tag{5.4}$$

ที่ 25 °C, $K_w = 10^{-14}$

ความเป็นกลางทางประจุไฟฟ้า

$$\zeta + [H^{+}] = [OH^{-}] + [AC^{-}] \tag{5.5}$$

ซึ่งเมื่อ แทนสมการที่ (5.3) และ (5.4) ลงในสมการที่ (5.5) แล้วจะสามารถ จัดเรียงสมการได้ใหม่เป็น

$$[H^{+}]^3 + (K_a + \zeta)[H^{+}]^2 + (K_a\zeta - K_a\xi - K_w)[H^{+}] - K_aK_w = 0 \tag{5.6}$$

จากสมการที่ (5.1) และ (5.2) เราสามารถหาค่าของ ξ และ ζ ได้ และเมื่อแทนค่าลงในสมการ (5.6) เราจะสามารถทราบความเข้มข้นของ ไฮโดรเจนอิออน ณ เวลาต่างๆได้โดย การหาคำตอบของสมการพหุนามกำลังสาม

ค่าความเป็นกรด-ด่าง (pH) สามารถหาได้จากสมการความสัมพันธ์ของค่าความเป็นกรด-ด่าง กับความเข้มข้นของไฮโดรเจนอิออน ดังสมการ (5.7)

$$pH = -\log_{10}[H^+] \quad (5.7)$$

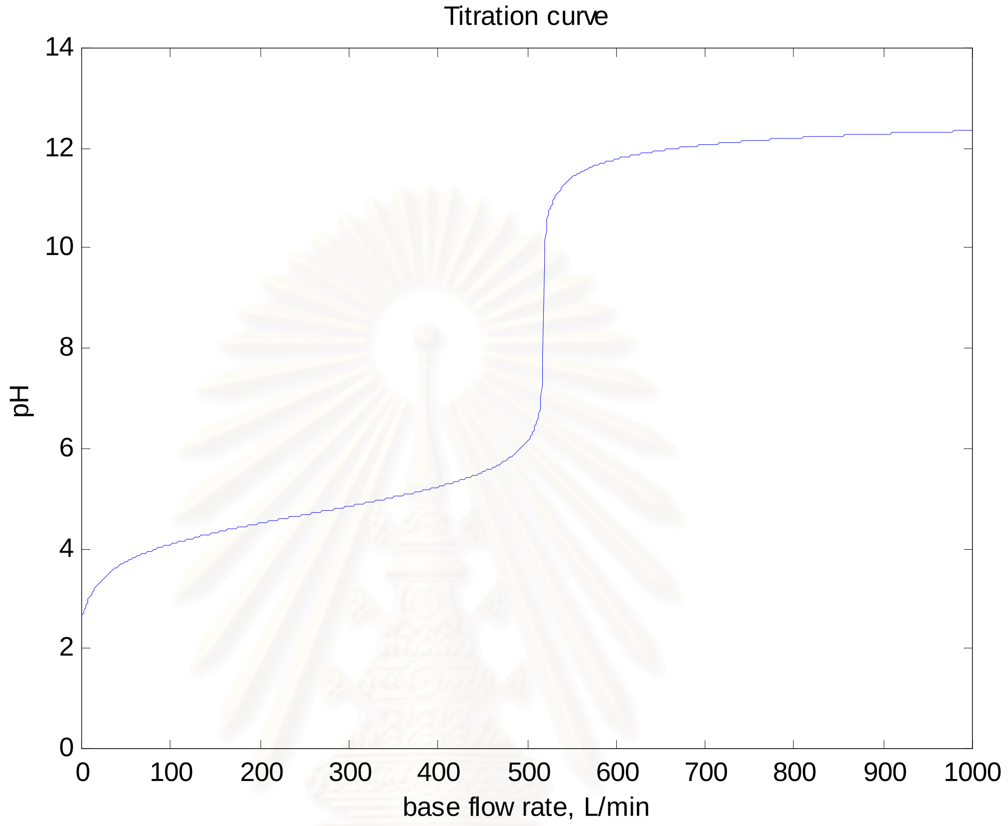
สำหรับค่าคงที่ต่างๆ เช่น ปริมาณการไหล และมิติของเครื่องปฏิกรณ์ กำหนดไว้ ดังตารางที่ 5.1

ตารางที่ 5.1 แสดงตัวแปรต่างๆ ในกระบวนการปรับค่าความเป็นกรด-ด่างให้อยู่ในสภาวะเป็นกลาง

ตัวแปร	ค่าที่กำหนด
ปริมาตรของถัง	$V = 1000\ L$
อัตราการไหลของกรดอะซิติก	$F_1 = 81\ L/min$
อัตราการไหลของโซเดียมไฮดรอกไซด์ที่สภาวะคงที่	$F_2 = 515\ L/min$
ค่าความเป็นกรด-ด่างที่สภาวะคงที่	$pH_{sp} = 7$
ความเข้มข้นของกรดอะซิติก	$C_1 = 0.32\ mol/L$
ความเข้นข้นของโซเดียมไฮดรอกไซด์	$C_2 = 0.05\ mol/L$
ความเข้มข้นเริ่มต้นของอะซิเตทอิออน	$\xi(0) = 0.0432\ mol/L$
ความเข้นข้นเริ่มต้นของโซเดียมไฮดรอกไซด์	$\zeta(0) = 0.0435\ mol/L$
ช่วงเวลาเก็บตัวอย่าง	$T_s = 0.1\ min$

5.2 เส้นกราฟของการไตเตรท (Titration curve)

จากข้อมูลที่ได้กล่าวมาแล้วข้างต้น สามารถสร้างกราฟของการไตเตรทได้ดังรูปที่ 5.2



รูปที่ 5.2 เส้นกราฟของการไตเตรท

จากรูปที่ 5.2 จะเห็นว่าในช่วงค่าพีเอชที่เป็นกลางเท่ากับ 7 ความสัมพันธ์ของการเปลี่ยนแปลงของค่าพีเอชกับการเปลี่ยนแปลงของปริมาณด่างจะเป็นไปอย่างรวดเร็ว และแสดงให้เห็นถึงความไม่เป็นเชิงเส้นของกระบวนการได้อย่างชัดเจน

บทที่ 6

การจำลองการควบคุมกระบวนการ

ในบทนี้ จะเน้นไปที่วิธีการจำลองการควบคุมกระบวนการและผลที่ได้จากการจำลองการควบคุมกระบวนการ โดยที่จะทำการทดสอบตัวควบคุมที่นำเสนอบนสถานการณ์ 3 ลักษณะ คือ

1. กระบวนการที่มีการตอบสนองอันดับสองที่มีการเปลี่ยนแปลงค่าพารามิเตอร์

2. กระบวนการที่มีการตอบสนองอันดับสองและมีเวลาหน่วง

3. กระบวนการที่มีความไม่เป็นเชิงเส้นสูง

ซึ่งแสดงอยู่ในตอนที่ 6.3 ถึงตอนที่ 6.5 การจำลองการควบคุมกระบวนการดังกล่าวข้างต้น จะทำการศึกษาเปรียบเทียบตัวควบคุมพีไอดีชนิดปรับค่าเองได้ โดยใช้เจเนติกอัลกอริธึม ซึ่งสามารถปรับค่าพารามิเตอร์ได้ กับตัวควบคุมพื้นฐาน คือ ตัวควบคุมพีไอดีธรรมดาที่มีค่าพารามิเตอร์คงที่

สำหรับการจำลองการควบคุมที่เกิดขึ้นบนสองกระบวนการแรกนั้น มีวัตถุประสงค์เพื่อที่จะแสดงให้เห็นว่า ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ที่นำเสนอนี้ สามารถใช้ในการควบคุมกระบวนการได้หลากหลาย นอกเหนือจากกระบวนการที่มีความไม่เป็นเชิงเส้นสูงที่ต้องการศึกษาอย่างละเอียดในงานวิจัยนี้ ดังนั้นกรณีที่ใช้ทดสอบ จะแสดงเพื่อเป็นตัวอย่างเพียงเล็กน้อยเท่านั้น

สำหรับการจำลองการควบคุมกระบวนการในสถานการณ์ที่ 3 หรือบนกระบวนการที่มีความไม่เป็นเชิงเส้นสูง จะมีการศึกษาตัวแปรต่างๆ อย่างละเอียด เพื่อให้มีการเปรียบเทียบผลการควบคุมที่ชัดเจนมากขึ้น

ในการเปรียบเทียบประสิทธิภาพของตัวควบคุมที่ใช้ในการศึกษาครั้งนี้ จะเปรียบเทียบด้วยดัชนีคือ ดัชนีค่าอินทีกรัลความผิดพลาดสัมบูรณ์ (Integral Absolute Error, IAE) การจำลองการควบคุมกระบวนการที่เกิดขึ้นในงานวิจัยนี้ ทำการศึกษาบนโปรแกรมคอมพิวเตอร์สำเร็จรูป MATLAB version 5.3.1 ซึ่งสามารถดูรายละเอียดของโปรแกรมเพิ่มเติมได้ในภาคผนวก ง.

แต่ก่อนที่จะถึงการทดสอบการจำลองกระบวนการ เนื่องจากงานวิจัยนี้เป็นการจำลองการควบคุมด้วยโปรแกรมคอมพิวเตอร์ดังนั้น จะแสดงเพิ่มเติมให้เห็นถึงผลการหาค่าที่เหมาะสมที่สุด (Optimization) โดยใช้เจเนติกอัลกอริธึม ซึ่งทำให้มั่นใจได้ว่าความผิดพลาดในการหาคำตอบที่เหมาะสมที่สุดของฟังก์ชันเป้าหมาย ที่เกิดจากโปรแกรมจะมีค่าต่ำที่สุด หรือเป็นการทดสอบความ

ถูกต้องของขั้นตอนต่างๆ ของตัวโปรแกรม เพื่อลดผลที่อาจจะเกิดจากความผิดพลาดของตัวโปรแกรมที่จะทำให้ผลการจำลองการควบคุมเป็นผลที่เบี่ยงเบนไปเนื่องจากการเขียนโปรแกรม

โดยการทดสอบนี้จะมี 2 กรณีคือ การทดสอบหาจุดคำตอบที่ทำให้ฟังก์ชันโรเซนบรอกค์มีค่าต่ำที่สุด และการทดสอบการประมาณค่าพารามิเตอร์ของสมการอันดับสองด้วยแบบจำลองของสมการหนึ่งที่มีเวลาหน่วง

6.1 การหาจุดที่ทำให้ฟังก์ชันโรเซ็นบรอกค์ (Rosenbrock) ต่ำที่สุด

ตอนที่ 6.1 นี้จะเป็นการใช้เจเนติกอัลกอริธึม ในการหาค่าขีดสุด (Extreme) ของฟังก์ชันทดสอบ เพื่อแสดงให้เห็นถึงความสามารถของการหาค่าขีดสุดด้วยกระบวนวิธีนี้ ซึ่งฟังก์ชันที่ใช้ทดสอบครั้งนี้มีชื่อว่า สมการของโรเซ็นบรอกค์ (Rosenbrock's Equation) ที่มี 2 ตัวแปร ซึ่งสามารถเขียนได้ดังสมการที่ (6.1)

$$f(x_1, x_2) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2 \qquad (6.1)$$

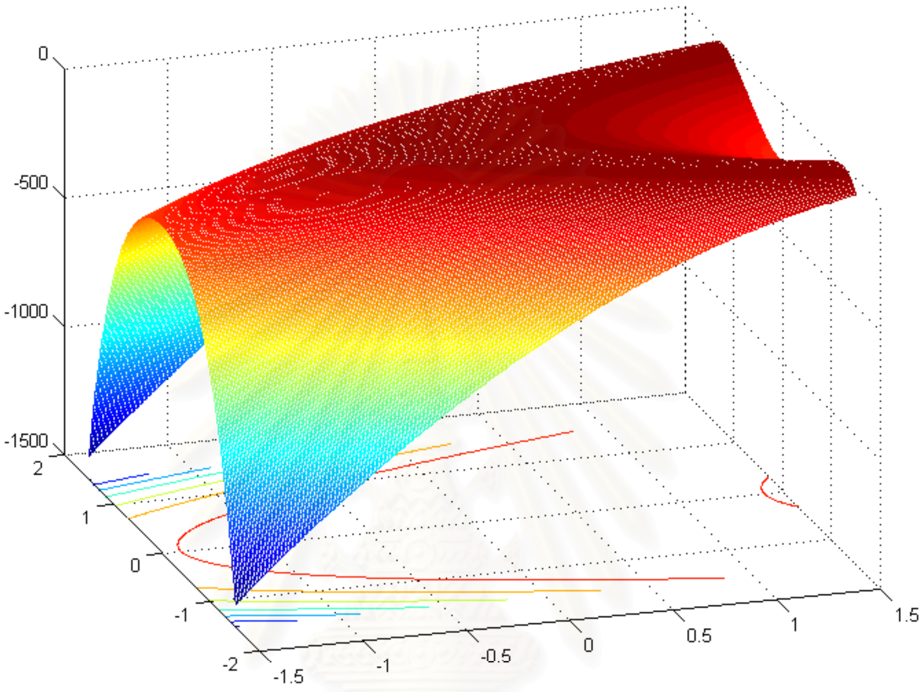
สมการนี้มีค่าต่ำที่สุดอยู่ที่จุด (1,1) ซึ่งจะให้ค่าของฟังก์ชัน $f(x_1, x_2)$ มีค่าเป็นศูนย์

แต่เนื่องจากเจเนติกอัลกอริธึม ที่ใช้ในงานวิจัยนี้ เป็นเจเนติกอัลกอริธึมที่ใช้ในการหาค่าสูงที่สุดเท่านั้น ดังนั้นจึงต้องมีการปรับสมการให้เป็นไปดังสมการ (6.2) เพื่อให้สามารถใช้เป็นฟังก์ชันวัตถุประสงค์ (Objective Function) ในการทดสอบครั้งนี้

$$f(x_1, x_2) = -[100(x_1^2 - x_2)^2 + (1 - x_1)^2] \qquad (6.2)$$

สมการความสัมพันธ์ของโรเซ็นบรอคตามสมการที่ (6.2) สามารถแสดงในรูปของกราฟสามมิติระหว่างตัวแปร x_1 และ x_2 และค่าของฟังก์ชัน ได้ดังรูปที่ 6.1 จะเห็นได้ว่า เป็นการยากที่จะหาคำตอบที่ถูกต้องได้อย่างแม่นยำ เนื่องจากการมีความราบเรียบ (Flat) ของพื้นผิวรอบจุด (1,1) หรือในช่วงที่ $-1 < x_1 < 1$ และ $-1 < x_2 < 1$ ในขณะที่ จุดที่ห่างออกมา ($|x_1, x_2| > 1$) ฟังก์ชันจะมีความชันสูงมาก หรือเรียกได้ว่าเป็นฟังก์ชันที่มีแอ่งหุบเขา (Valley) ซึ่งไม่เหมาะสมสำหรับการหาค่าที่ต่ำที่สุด (เหมาะสมที่สุด) โดยการใช้วิธีเชิงเส้นจำพวก วิธีการซิมเพล็กซ์ (Simplex Method) หรือโปรแกรมควอดราติก (Quadratic Programming), วิธีการลดลงอย่างชันที่สุด (Method of Steepest Descent) ฯลฯ

วิธีการของเจเนติกอัลกอริธึมจะเข้าสู่คำตอบของการหาค่าที่เหมาะสมที่สุดได้ดีกว่า วิธีอื่นๆ เนื่องจากมีการสุ่มตัวอย่างประชากรของคำตอบ เพื่อที่จะมาผ่านกระบวนการดังที่ได้อธิบายไปแล้วในบทที่ 3 ซึ่งจะทำให้ไม่ติดอยู่กับค่าที่เหมาะสมท้องถิ่น (Local Optima) ได้ง่ายๆ ทำให้สามารถเข้าถึงคำตอบที่ถูกต้องได้



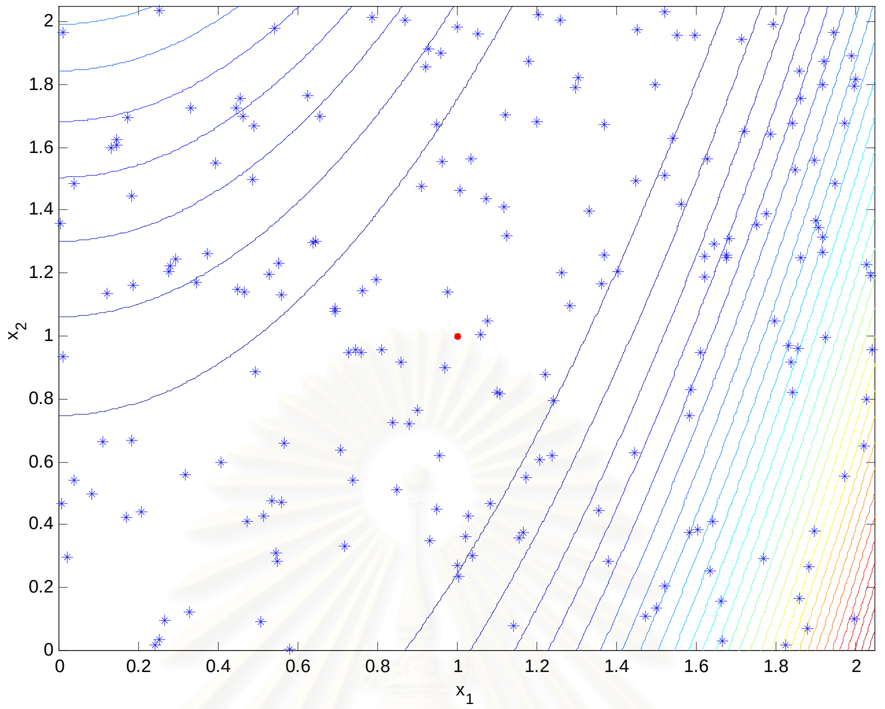
รูปที่ 6.1 กราฟความสัมพันธ์ของโรเซ็นบรอกค์

คำตอบของสมการคือ $x_1 = 1$ และ $x_2 = 1$ ซึ่งจะได้ค่าสูงที่สุดคือ 0

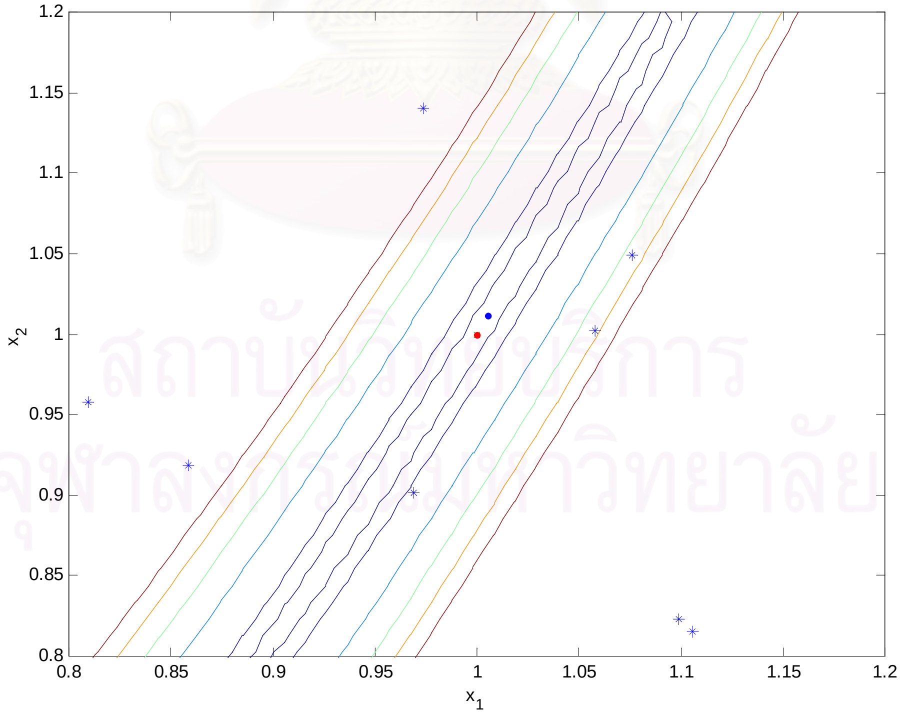
จะเห็นได้จากรูปที่ 6.1 ว่าในบริเวณรอบๆ จุดคำตอบ จะมีความแตกต่างของค่าของฟังก์ชันน้อยมาก เมื่อเทียบกับบริเวณที่ห่างออกไป

ตามขั้นตอนของเจเนติกอัลกอริธึม จะต้องสุ่มเลือกประชากรคำตอบออกมาจากช่วงของคำตอบที่เป็นไปได้ หลังจากนั้นจึงคำนวณหาค่าความแข็งแรง (Fitness) ของประชากรทั้งหมด เพื่อที่จะใช้ในการคัดเลือกหรือ หาคำตอบที่เหมาะสมที่สุดต่อไป ในรูปที่ 6.2 แสดงกราฟที่มีเส้นบอกระดับความสูง และการกระจายตัวของประชากรเริ่มต้น ที่จะใช้ในกลไกกระบวนวิธีของเจเนติกอัลกอริธึมต่อไป

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

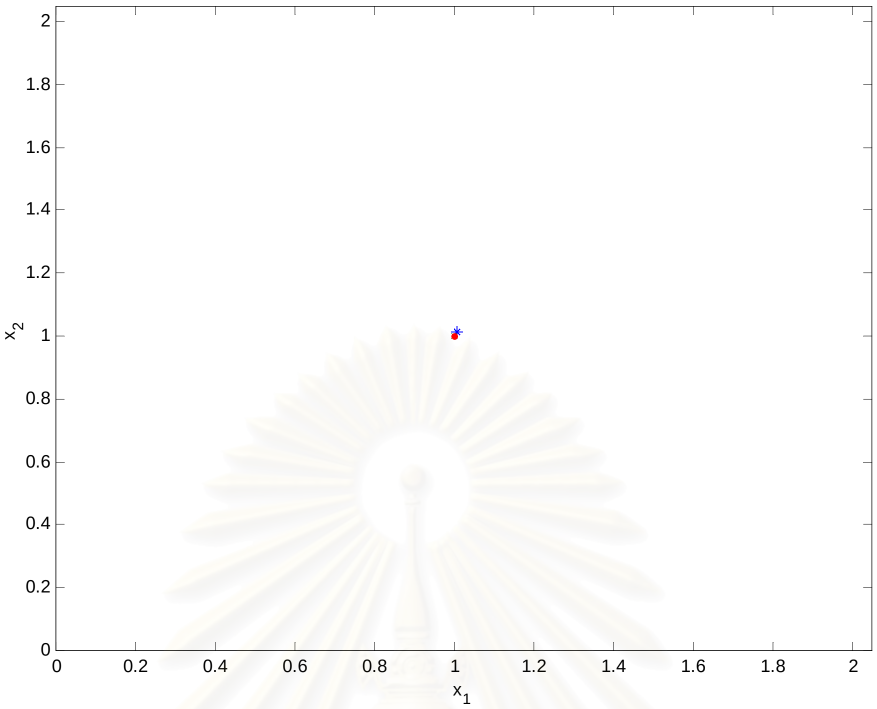


รูปที่ 6.2 แสดงการกระจายตัวของประชากรคำตอบที่สุ่มเลือกบนช่วงที่ ([0,2],[0,2]) ในรอบแรก

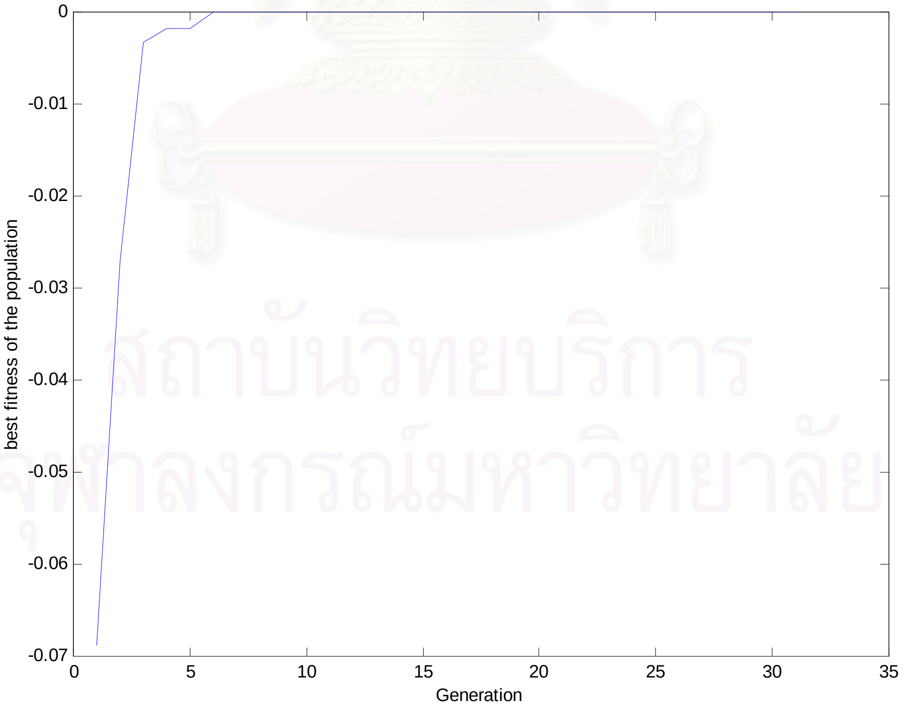


รูปที่ 6.3 กราฟขยายแสดงเส้นความสูงรอบๆ จุดคำตอบ (1,1)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

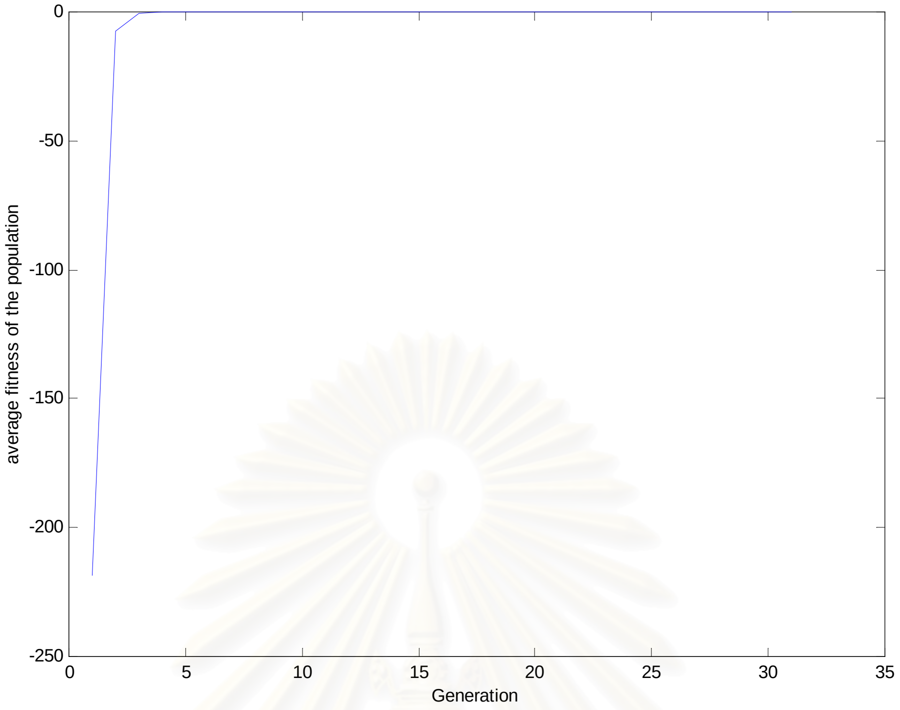


รูป 6.4 การกระจายตัวของประชากรคำตอบ ณ รอบการคำนวณสุดท้าย



รูป 6.5 ค่าความแข็งแรงของประชากรที่มีความแข็งแรงที่สุด ในแต่ละรอบการคำนวณ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 6.6 ค่าเฉลี่ยความแข็งแรงของประชากร ในแต่ละรอบการคำนวณ

คำตอบที่ดีที่สุดในการคำนวณรอบ (First Generation) แรก เป็นดังนี้

x_1 = 0.7389 x_2 = 0.5435 $f(x_1, x_2)$ = -0.0688

หลังจากเข้าสู่กระบวนการหาคำตอบของเจเนติกอัลกอริธึม จะสังเกตถึงความก้าวหน้าในการเข้าสู่คำตอบได้จากกราฟ 2 รูปคือ รูปที่ 6.5 จะเห็นว่าทุกๆ รอบการคำนวณ ค่าที่ดีที่สุด จะให้ $f(x_1, x_2)$ ที่มีค่าเข้าใกล้ศูนย์มากขึ้นเรื่อยๆ และจากรูปที่ 6.6 พบว่าค่าเฉลี่ยของ $f(x_1, x_2)$ ของประชากรทั้งหมดมีค่าเพิ่มขึ้นเรื่อยๆ เช่นกัน

ณ ที่รอบการคำนวณสุดท้ายจะเห็นได้จากรูปที่ 6.4 ว่า ประชากรทั้งหมดมีค่าเข้าใกล้คำตอบคือจุด (1,1) เมื่อเปรียบเทียบกับรูปที่ 6.2 จะเห็นว่าการหาคำตอบของปัญหาค่าเหมาะสมที่สุด ด้วยเจเนติกอัลกอริธึม สามารถเข้าสู่คำตอบได้ โดยที่คำตอบที่ดีที่สุดในการคำนวณรอบสุดท้าย (Final Generation) เป็นดังนี้

x_1 = 1.0055 x_2 = 1.0111 $f(x_1, x_2)$ = -3.0731e-005

ผลการทดสอบในตอนที่ 6.1 นี้แสดงให้เห็นและมั่นใจได้ว่า โปรแกรมการหาค่าที่เหมาะสมที่สุดด้วยเจเนติกอัลกอริธึมที่เขียนขึ้น สามารถหาคำตอบของการหาค่าที่เหมาะสมที่สุดได้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

6.2 การทดสอบการหาค่าพารามิเตอร์ของกระบวนการที่ทราบค่าแล้ว

ในตอนที่ 6.2 นี้ เป็นการทดสอบการหาค่าพารามิเตอร์ของแบบจำลองกระบวนการอันดับหนึ่งที่มีเวลาหน่วง เมื่อใช้ในการประมาณกระบวนการอันดับสองที่ทราบค่าพารามิเตอร์แน่นอน โดยในการหาค่าพารามิเตอร์นี้ จะทำโดยการใช้เจเนติกอัลกอริธึม เพื่อหาคำตอบ (พารามิเตอร์) ที่ทำให้ฟังก์ชันวัตถุประสงค์ (Objective Function) คือ

$$\min \sum_{i=1}^{n} \left(y_{measure} - y_{estimate} \right)_i^2 \qquad (6.3)$$

มีค่าต่ำที่สุด หรือเรียกว่าวิธีกำลังสองน้อยที่สุด (Least Square) ซึ่งขั้นตอนนี้เป็นหนึ่งในส่วนสำคัญของการทำการออกแบบตัวควบคุมชนิดปรับค่าเองได้ ที่จะใช้ในการทำการประมาณแบบจำลอง หรือการหาเอกลักษณ์ของระบบ (System Identification)

สมการของกระบวนการอันดับสองที่ใช้เป็นโจทย์ของการทดลองในตอนนี้ จะมีรูปแบบและค่าพารามิเตอร์ต่างๆ ดังนี้

$$\tau^2 \frac{d^2Y(t)}{dt^2} + 2\xi\tau \frac{dY(t)}{dt} + Y(t) = K_p X(t-\theta) \qquad (6.4)$$

เมื่อกำหนดให้พารามิเตอร์ของกระบวนการมีค่าดังต่อไปนี้

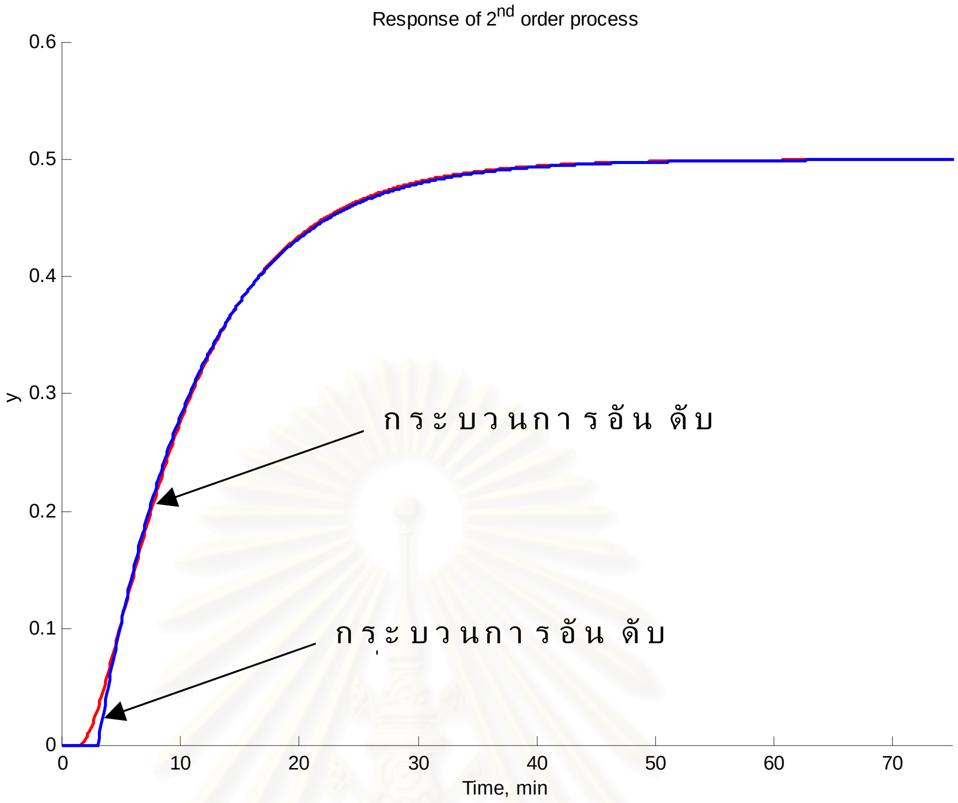
$$K_p = 0.5$$

$$\tau = 4$$

$$\xi = 1.25$$

$$\theta = 0$$

ผลการตอบสนองลูพเปิดของกระบวนการนี้ ที่มีต่อการเปลี่ยนแปลงอินพุท X(t) จาก 0 ไปเป็น 1 แสดงอยู่ในรูปที่ 6.7



รูปที่ 6.7 การตอบสนองวงเปิดของสมการอันดับสอง (6.4) เมื่อมีการเปลี่ยนแปลงตัวแปรอินพุท และการประมาณด้วยสมการอันดับหนึ่งที่มีเวลาหน่วง

รูปที่ 6.7 แสดงให้เห็นว่าเส้นกราฟที่ได้จากการประมาณและเส้นกราฟที่ได้จากแบบจำลองจริงมีความแตกต่างกันน้อยมาก นั่นคือการประมาณค่าด้วยโปรแกรมที่สร้างขึ้น โดยการใช้เจเนติกอัลกอริธึมในการหาค่าที่เหมาะสมที่สุด สามารถให้คำตอบที่เป็นที่น่าพอใจ และทำให้มั่นใจได้ว่า การจำลองการควบคุมกระบวนการในขั้นตอนต่อไป ซึ่งเป็นจุดมุ่งหมายหลักของงานวิจัยนี้ จะเป็นการวัดผลได้อย่างถูกต้อง โดยที่จะเชื่อถือผลการทดลองได้ว่า ไม่มีความผิดพลาดที่เกิดจากตัวโปรแกรม หรือมีผลน้อยที่สุด

เมื่อเปรียบเทียบค่าพารามิเตอร์ที่ได้จากการคำนวณโดยโปรแกรมที่สร้างขึ้น กับค่าพารามิเตอร์ที่ได้จากการประมาณค่าด้วยวิธีฟิต 3 (Fit 3) สามารถเปรียบเทียบผลได้จากตารางด้านล่าง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 6.1 ตารางเปรียบเทียบค่าพารามิเตอร์ที่ได้จากโปรแกรม และค่าพารามิเตอร์ที่ได้จากวิธีฟิต 3

พารามิเตอร์ / วิธีการ	K_p	τ	θ	Error
ฟิต 3	0.5	8.55	1.65	1.43
โปรแกรมที่เขียนขึ้น	0.5	8.50	1.65	1.23

ความแตกต่างของแบบจำลองที่ได้จากโปรแกรมที่เขียนขึ้น กับกระบวนการจริง เมื่อเปรียบเทียบกับความแตกต่างของแบบจำลองที่ได้จากวิธีประมาณฟิต 3 กับกระบวนการจริง พบว่ามีค่าผิดพลาดน้อยกว่าเล็กน้อย นั่นคือ โปรแกรมที่เขียนขึ้นสามารถให้ผลการประมาณค่าพารามิเตอร์ที่เป็นที่น่าพอใจและเชื่อถือได้ สามารถจะใช้ในการจำลองสถานการณ์ในตอนต่อไป (6.3 - 6.5) ได้

ดังที่ได้เกริ่นไว้ในตอนต้นแล้วว่าในตอนที่ 6.3 – 6.5 นี้จะประกอบไปด้วย การทดสอบบนสถานการณ์ที่ต่างๆ กัน 3 แบบ ซึ่งจะเป็นการทดสอบตัวควบคุมพีไอดีชนิดปรับค่าเองได้ที่มีการประยุกต์ใช้เจเนติกอัลกอริธึมในการหาค่าที่เหมาะสมที่สุด ที่ได้ทดสอบสมรรถนะมาแล้วในตอนที่ 6.1 – 6.2

ผลการทดสอบเป็นดังต่อไปนี้

6.3 การจำลองการควบคุมบนกระบวนการที่มีการตอบสนองอันดับสอง ที่มีค่าพารามิเตอร์ เปลี่ยนแปลง (Second Order Process with Parameter Changes)

6.3.1 แบบจำลองของกระบวนการ

กระบวนการอันดับสองที่เลือกใช้เพื่อทดสอบตัวควบคุมพีไอดีชนิดปรับค่าเองได้ที่นำเสนอในงานวิจัยตอนที่ 6.3 นี้ จะเป็นกระบวนการที่มีแบบจำลองทางคณิตศาสตร์ดังนี้

$$\tau^2 \frac{d^2Y(t)}{dt^2} + 2\xi\tau\frac{dY(t)}{dt} + Y(t) = K_p X(t-\theta) \quad (6.5)$$

เมื่อกำหนดให้พารามิเตอร์ของกระบวนการมีค่าดังต่อไปนี้

$$K_p = 1$$

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

$\tau = 2$

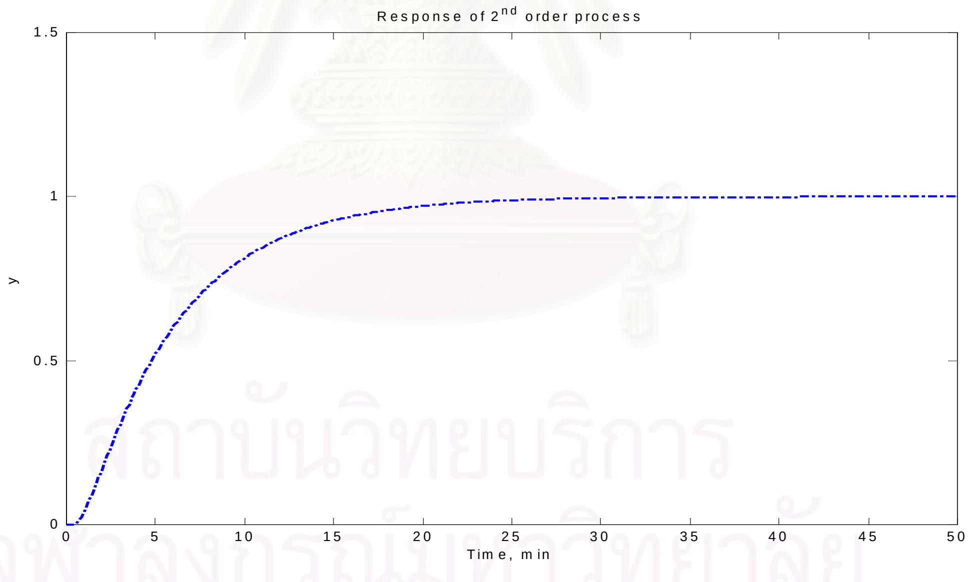
$\xi = 1.5$

$\theta = 0$

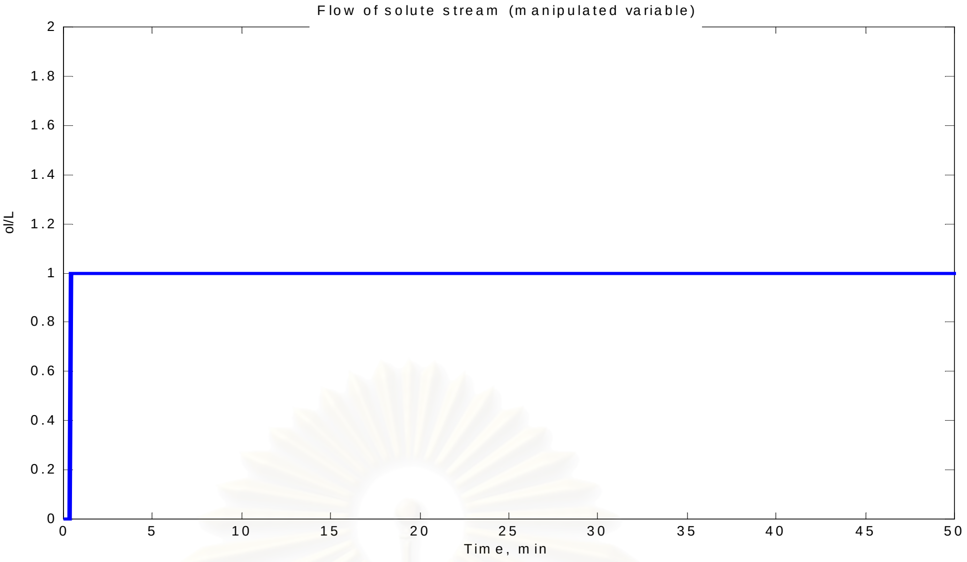
เพื่อที่จะทำการหาค่าพารามิเตอร์ของตัวควบคุมพีไอดี ที่ใช้ในการควบคุมกระบวนการตามสมการที่ (6.5) สามารถทำได้โดยการประมาณค่าพารามิเตอร์ของกระบวนการดังกล่าวให้อยู่ในรูปของกระบวนการอันดับหนึ่งที่มีเวลาหน่วง (First-Order plus Dead-Time Process) ดังสมการที่ (6.2) ด้วยการหาการตอบสนองลูพเปิด (Open-loop Response)

$$\tau \frac{dY(t)}{dt} + Y(t) = K_p X(t-\theta) \tag{6.6}$$

แล้วจึงแทนค่าพารามิเตอร์ของกระบวนการที่ได้ ลงในความสัมพันธ์ของ Ziegler-Nichols [ภาคผนวก ข.] ซึ่งในที่นี้จะทำการประมาณค่าพารามิเตอร์จากการตอบสนองลูพเปิดด้วยวิธีฟิต 3 (Fit 3) (Carlos, 1997)



รูปที่ 6.8 การตอบสนองลูพเปิดของกระบวนการดังสมการที่ (6.5)



รูปที่ 6.9 การเปลี่ยนแปลงแบบขั้น (Step Input) ของตัวแปรอินพุทเพื่อใช้ในการหาการตอบสนองของกระบวนการอันดับสองดังสมการที่ (6.5)

รูปที่ 6.8 เป็นการตอบสนองลูพเปิดของกระบวนการอันดับสองดังสมการที่ (6.5) เมื่อมีการนำการเปลี่ยนแปลงแบบขั้นของตัวแปรอินพุท ดังรูปที่ 6.9 และเมื่อประมาณกระบวนการให้อยู่ในรูปของกระบวนการอันดับหนึ่งที่มีเวลาหน่วงด้วยวิธีฟิต 3 จะได้ค่าพารามิเตอร์ของกระบวนการดังนี้

$$K_p = 0.9999;\ \tau = 5.4;\ \theta = 0.6 \qquad (6.7)$$

และหลังจากแทนค่าพารามิเตอร์ที่ได้นี้ลงในความสัมพันธ์ของ Ziegler-Nichols จะได้ค่าพารามิเตอร์ของตัวควบคุมพีไอดีดังนี้ คือ

$$K_C = 10.8011;\ \tau_I = 1.2;\ \tau_D = 0.3 \qquad (6.8)$$

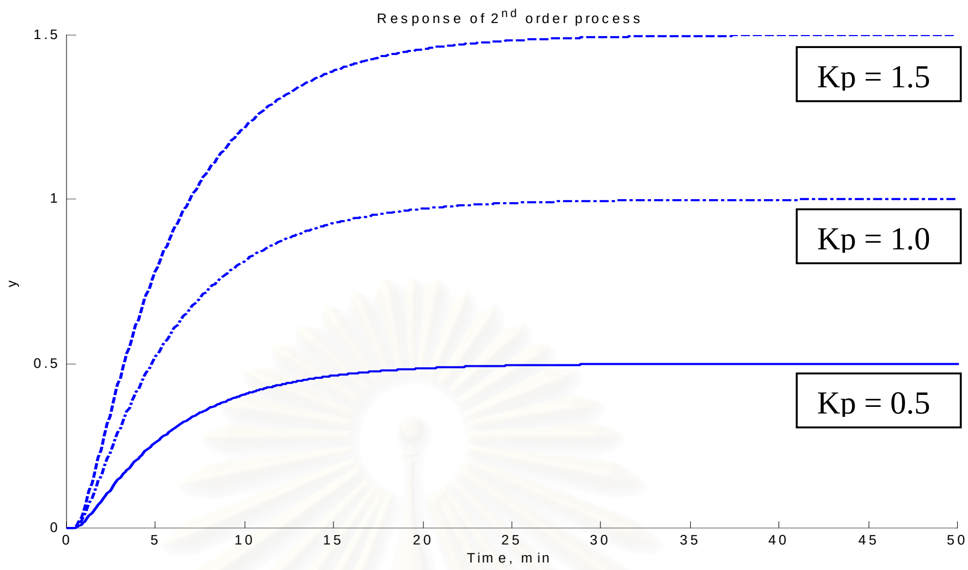
การทดสอบตัวควบคุมในตอนที่ 6.3 นี้ จะแบ่งออกเป็น 3 กรณี ดังนี้

6.3.1 กระบวนการมีค่าเกน (Process Gain, K_p) ลดลง 50% จากสมการที่ (6.5)

6.3.2 กระบวนการมีค่าเวลาคงที่ (Time Constant, τ) เพิ่มขึ้น 25% จากสมการที่ (6.5)

6.3.3 กระบวนการมีค่าเวลาคงที่ (Time Constant, τ) ลดลง 25% จากสมการที่ (6.5)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 6.10 การตอบสนองลูพเปิดเมื่อกระบวนการมีค่าเกนของกระบวนการที่ต่างๆ กัน

โดยมีขั้นตอนการทดลองดังนี้

(1) เริ่มการจำลองกระบวนการตามสมการที่ (6.5) ที่ค่าสภาวะคงตัว

(2) ใส่ตัวแปรรบกวนแบบขั้น เพื่อรบกวนระบบตามสมการที่ (6.5)

(3) หลังจากระบบเข้าสู่สมดุลแล้ว จึงเปลี่ยนแปลงค่าพารามิเตอร์ของกระบวนการดังกรณีที่กำหนด

(4) ใส่ตัวแปรรบกวนแบบขั้น เพื่อรบกวนระบบตามข้อที่ (2)

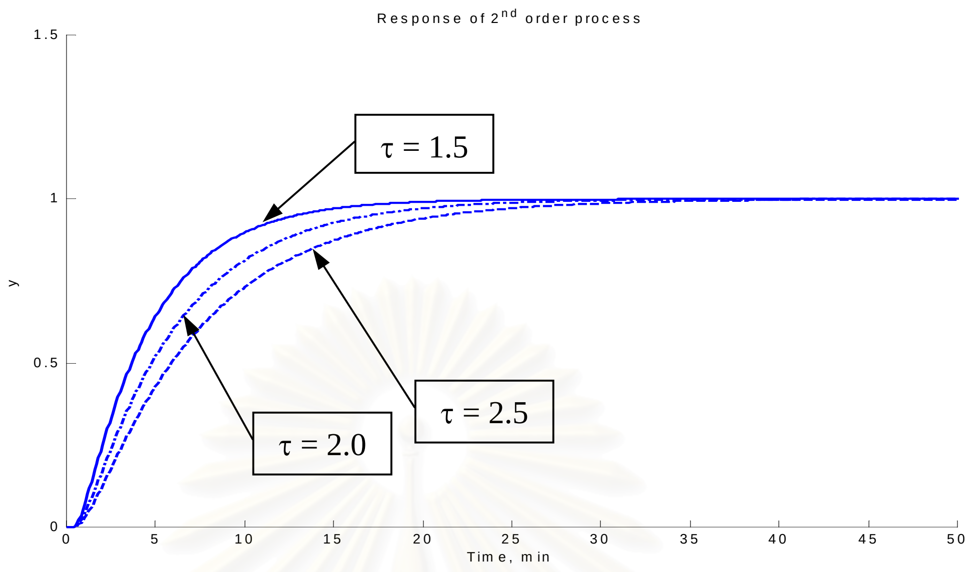
(5) เปรียบเทียบค่าดัชนี IAE

ซึ่งการทดลองตามลำดับที่ (1) ถึง (5) นั้น จะทดสอบด้วยตัวควบคุม 3 ประเภทคือ

() ตัวควบคุมพีไอดีธรรมดาที่มีพารามิเตอร์คงที่

() ตัวควบคุมพีไอดีธรรมดาแต่มีการเปลี่ยนค่าพารามิเตอร์

() ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ที่นำเสนอ



รูปที่ 6.11 การตอบสนองลูพเปิดเมื่อกระบวนการมีค่าเวลาคงที่ของกระบวนการที่ต่างๆ กัน

6.3.1 กระบวนการมีค่าเกน (Process Gain, K_p) ลดลง 50% จากสมการที่ (6.5)

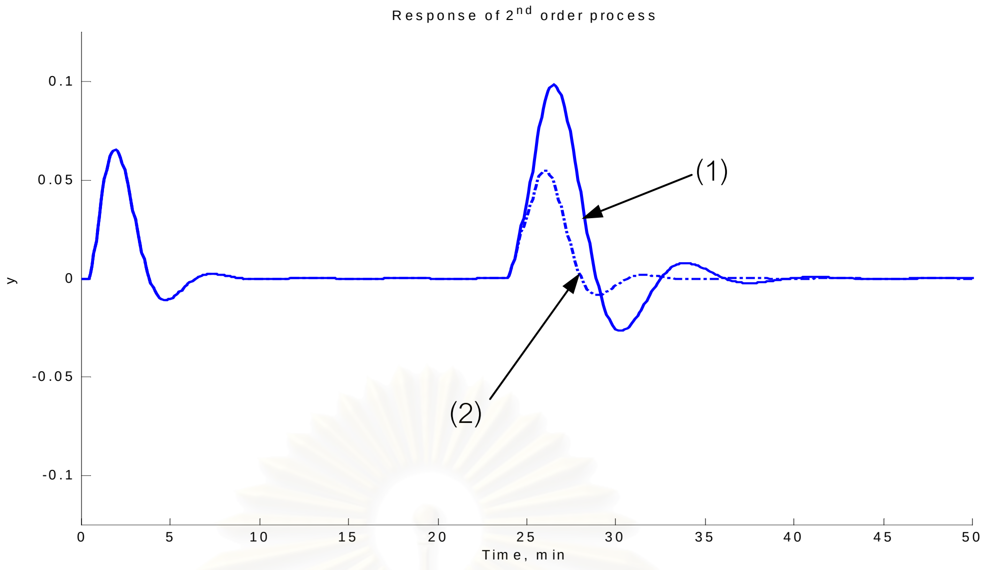
รูปที่ 6.10 แสดงให้เห็นผลการตอบสนองวงเปิด เมื่อกระบวนการมีค่าเกนที่เปลี่ยนแปลงไป จากการประมาณค่ากระบวนการให้อยู่ในรูปกระบวนการอันดับหนึ่งที่มีเวลาหน่วงจะได้ว่า

$$K_p = 0.5;\ \tau = 5.4;\ \theta = 0.6 \qquad (6.9)$$

เมื่อเปรียบเทียบกับพารามิเตอร์ตามสมการที่ (6.7) พบว่าค่า เกนของกระบวนการที่ได้จากการประมาณค่า ลดลงจากเดิม 50% ในขณะที่พารามิเตอร์ของกระบวนการตัวอื่นๆ ที่เหลือมีค่าไม่เปลี่ยนแปลง และเมื่อคำนวณหาค่าพารามิเตอร์ของตัวควบคุมพีไอดี จะได้ว่า

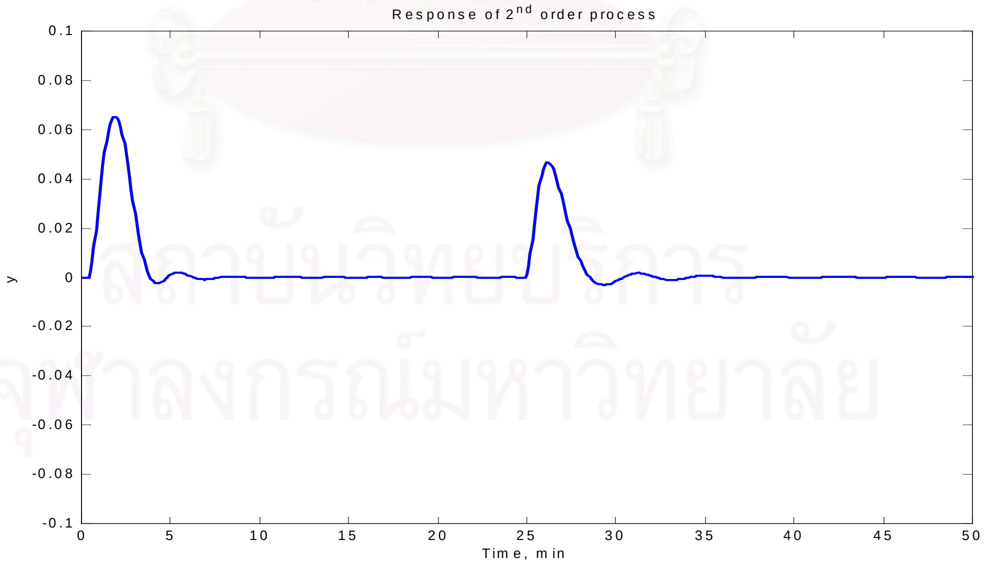
$$K_C = 21.6021;\ \tau_I = 1.2;\ \tau_D = 0.3 \qquad (6.10)$$

ค่าพารามิเตอร์ที่ใช้ในตัวควบคุมพีไอดี เมื่อกระบวนการมีค่าเกนลดลง จำเป็นที่จะต้องมีการเปลี่ยนแปลงไปด้วยเพื่อให้ผลการตอบสนองลูพปิดที่ เป็นไปตามเงื่อนไขเดิมตามความสัมพันธ์ของ Ziegler-Nichols นั่นคือ ค่าผิดพลาดลดลงเป็นอัตราส่วน 1/4 เท่า (Quarter Decay Ratio) รูปที่ 6.12 แสดงให้เห็นถึงความแตกต่างของตัวควบคุมพีไอดีที่มีการเปลี่ยนแปลงค่าพารามิเตอร์และไม่มีการเปลี่ยนแปลงค่าพารามิเตอร์



รูปที่ 6.12 ผลการตอบสนองลูพปิดภายใต้ตัวควบคุมพีไอดีธรรมดาที่ (1) ค่าพารามิเตอร์คงที่ (2) ค่าพารามิเตอร์เปลี่ยนแปลง

รูปที่ 6.12 แสดงให้เห็นว่าการเปลี่ยนแปลงค่าพารามิเตอร์ให้เหมาะสมกับกระบวนการที่เปลี่ยนไปนั้น จะให้ผลการตอบสนองที่ดีกว่า การควบคุมที่มีค่าพารามิเตอร์คงที่ และเมื่อเปรียบเทียบค่า IAE จะได้ดังตารางที่ 6.2



รูปที่ 6.13 ผลการตอบสนองลูพปิดภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ ที่มีการประยุกต์ใช้เจเนติกอัลกอริธึม

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รูปที่ 6.13 แสดงให้เห็นถึงผลการควบคุมกระบวนการที่เกิดขึ้นภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ที่นำเสนอ เมื่อใช้ควบคุมกระบวนการที่มีค่าเกนของกระบวนการลดลง 50% พบว่าให้การตอบสนองที่ดีกว่าเมื่อเปรียบเทียบกับกรณีตัวควบคุมพีไอดี (รูปที่ 6.12)

ตารางที่ 6.2 เปรียบเทียบค่าดัชนี IAE ในกรณีที่ 6.3.1

กรณี	ค่าดัชนี IAE	เปอร์เซ็นต์แตกต่าง
ตัวควบคุมพีไอดีพารามิเตอร์คงที่	0.4974	0.00
ตัวควบคุมพีไอดีพารามิเตอร์เปลี่ยนแปลงได้	0.2878	42.14
ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ที่นำเสนอ	0.2213	55.51

6.3.2 กระบวนการมีค่าเวลาคงที่ (Time Constant, τ) เพิ่มขึ้น 25% จากสมการที่ (6.5)

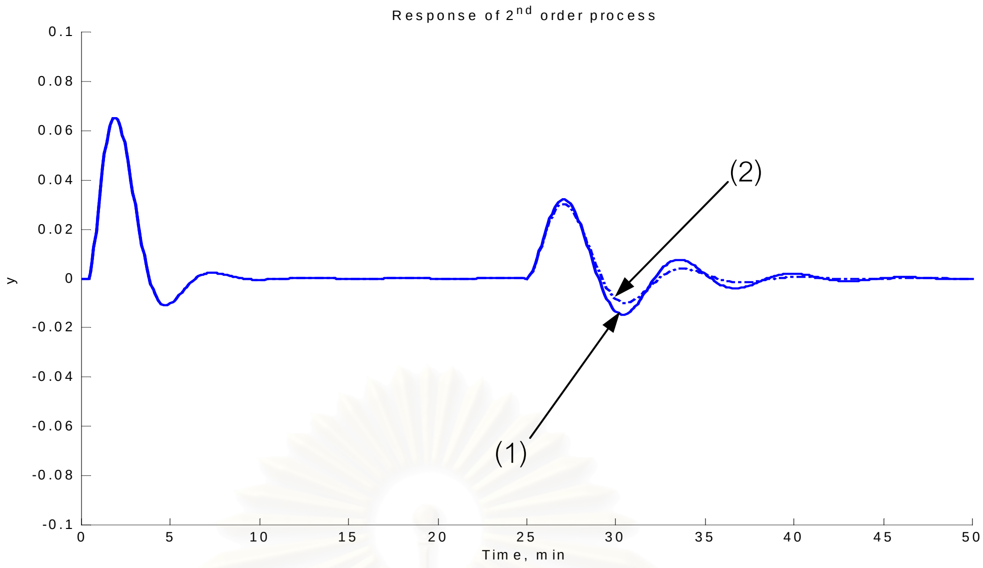
การตอบสนองวงเปิดที่เปลี่ยนแปลงไป เมื่อกระบวนการมีค่าเวลาคงที่แตกต่างไปจากเดิม แสดงอยู่ในรูปที่ 6.11 จากการประมาณค่ากระบวนการให้อยู่ในรูปกระบวนการอันดับหนึ่งที่มีเวลาหน่วงจะได้ว่า

$$K_p = 0.9994;\ \tau = 6.9;\ \theta = 0.7 \tag{6.11}$$

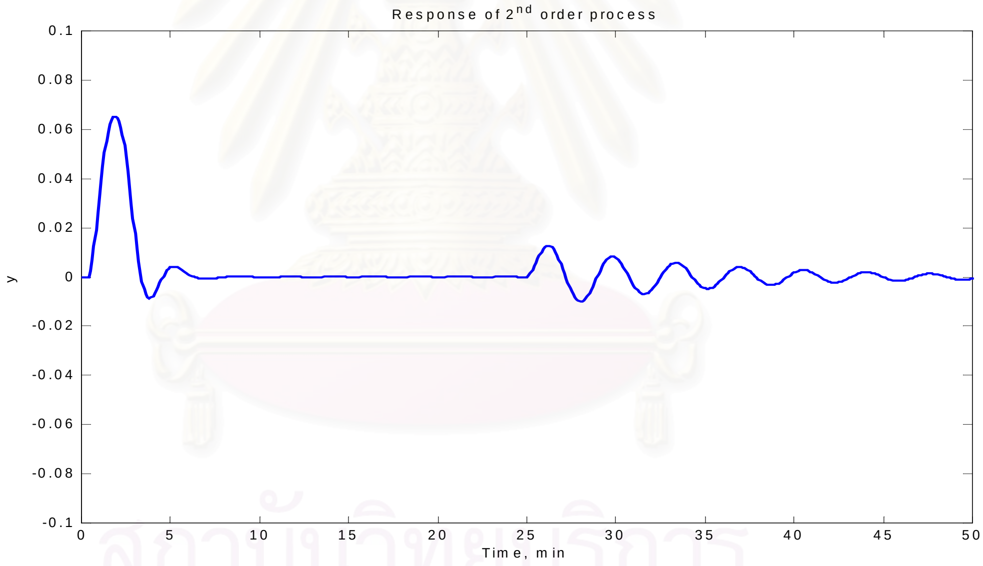
เมื่อเปรียบเทียบกับพารามิเตอร์ตามสมการที่ (6.7) พบว่าค่าเกนของกระบวนการที่ได้จากการประมาณค่าจะไม่แตกต่างจากเดิม ในขณะที่พารามิเตอร์ของกระบวนการตัวอื่นๆ มีค่าเปลี่ยนแปลงไปคือ มีเวลาคงที่และเวลาหน่วงที่เพิ่มขึ้น แสดงให้เห็นว่ากระบวนการมีความเร็วในการตอบสนองที่ลดลงจากเดิม ซึ่งเมื่อคำนวณหาค่าพารามิเตอร์ของตัวควบคุมพีไอดีตามวิธีของ Ziegler-Nichols จะได้ว่า

$$K_C = 11.8361;\ \tau_I = 1.4;\ \tau_D = 0.35 \tag{6.12}$$

ค่าพารามิเตอร์ที่ใช้ในตัวควบคุมพีไอดี เมื่อกระบวนการมีค่าเวลาคงที่เพิ่มขึ้น จำเป็นที่จะต้องมีการเปลี่ยนแปลงไปด้วยเพื่อให้ผลการตอบสนองลูพปิดที่เป็นไปตามเงื่อนไขเดิมตามความสัมพันธ์ของ Ziegler-Nichols นั่นคือ ค่าผิดพลาดลดลงเป็นอัตราส่วน 1/4 เท่า (Quarter Decay Ratio) รูปที่ 6.7 แสดงให้เห็นถึงความแตกต่างของตัวควบคุมพีไอดีที่มีการเปลี่ยนแปลงค่าพารามิเตอร์และไม่มีการเปลี่ยนแปลงค่าพารามิเตอร์



รูปที่ 6.14 ผลการตอบสนองลูพปิดภายใต้ตัวควบคุมพีไอดีธรรมดา (1) ค่าพารามิเตอร์คงที่ (2) ค่าพารามิเตอร์เปลี่ยนแปลง



รูปที่ 6.15 ผลการตอบสนองลูพปิดภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ ที่มีการประยุกต์ใช้เจเนติกอัลกอริธึม

รูปที่ 6.14 แสดงให้เห็นว่า การเปลี่ยนแปลงค่าพารามิเตอร์ให้เหมาะสมกับกระบวนการที่เปลี่ยนไปนั้น จะให้ผลการตอบสนองที่ดีกว่า การควบคุมที่มีค่าพารามิเตอร์คงที่ เช่นเดียวกันกับที่เกิดในกรณี 6.3.1 ส่วนรูปที่ 6.15 แสดงให้เห็นถึงผลการควบคุมกระบวนการที่เกิดขึ้นภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ที่นำเสนอ เมื่อใช้ควบคุมกระบวนการที่มีค่าเวลาคงที่ของกระบวน

การเพิ่มขึ้น 25% พบว่าให้การตอบสนองที่ดีกว่าเมื่อเปรียบเทียบกับกรณีตัวควบคุมพีไอดี และเมื่อเปรียบเทียบค่า IAE จะได้ดังตารางที่ 6.3

ตารางที่ 6.3 เปรียบเทียบค่าดัชนี IAE ในกรณีที่ 6.3.2

กรณี	ค่าดัชนี IAE	เปอร์เซ็นต์แตกต่าง
ตัวควบคุมพีไอดีพารามิเตอร์คงที่	0.2805	0.00
ตัวควบคุมพีไอดีพารามิเตอร์เปลี่ยนแปลงได้	0.2522	10.09
ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ที่นำเสนอ	0.2016	28.13

6.3.3 กระบวนการมีค่าเวลาคงที่ (Time Constant, τ) ลดลง 25% จากสมการที่ (6.5)

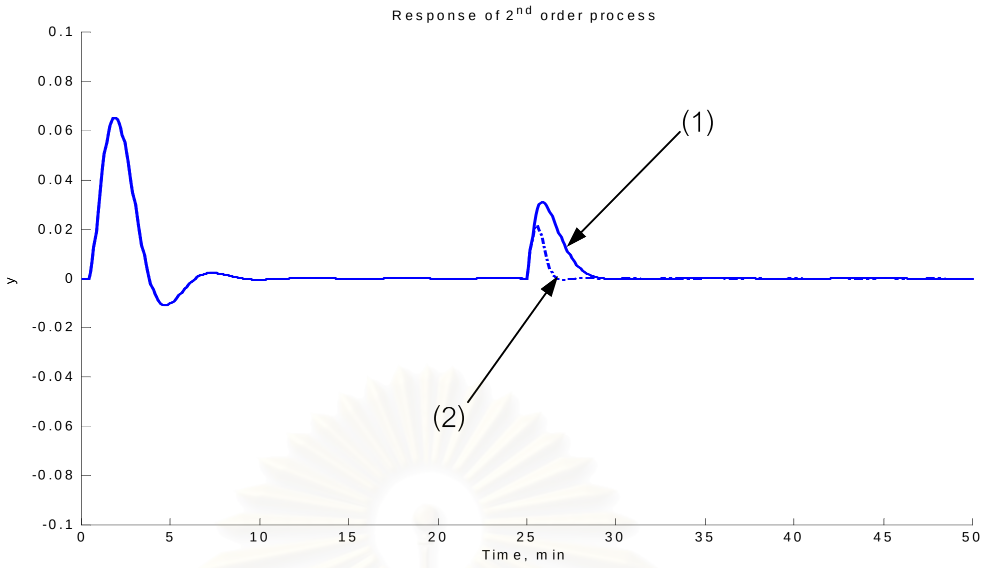
จากการประมาณค่ากระบวนการให้อยู่ในรูปกระบวนการอันดับหนึ่งที่มีเวลาหน่วงจะได้ว่า

$$K_p = 0.9994\,;\ \tau = 4.2\,;\ \theta = 0.3 \qquad (6.13)$$

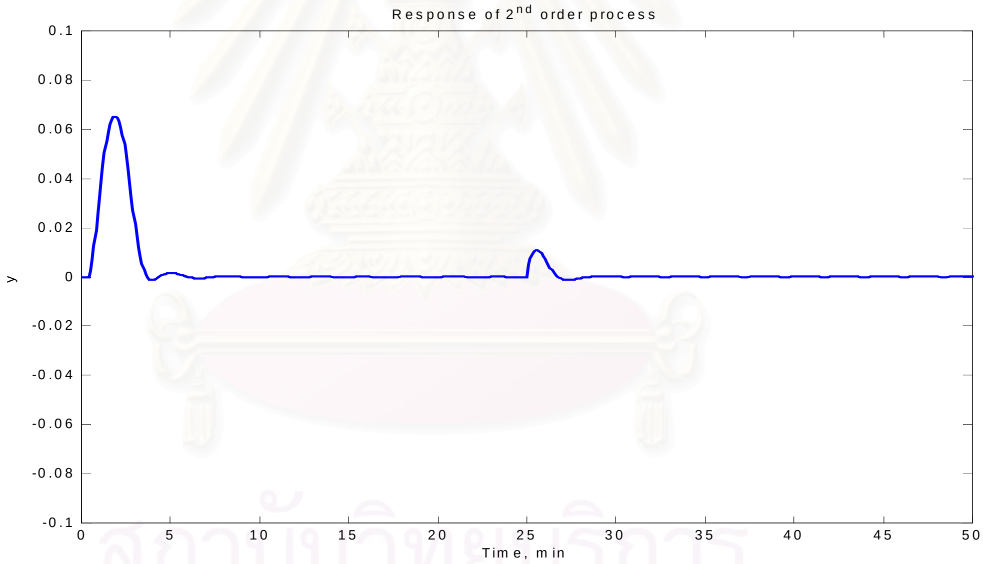
เมื่อเปรียบเทียบกับพารามิเตอร์ตามสมการที่ (6.7) พบว่าค่า เกนของกระบวนการที่ได้จากการประมาณค่าจะไม่แตกต่างจากเดิม ในขณะที่พารามิเตอร์ของกระบวนการตัวอื่นๆ มีค่าเปลี่ยนแปลงไปคือมีเวลาคงที่ และเวลาหน่วงลดลง แสดงให้เห็นว่ากระบวนการมีความเร็วในการตอบสนองมากขึ้น ซึ่งเมื่อคำนวณหาค่าพารามิเตอร์ของตัวควบคุมพีไอดีตามวิธีของ Ziegler-Nichols จะได้ว่า

$$K_C = 16.8001\,;\ \tau_I = 0.6\,;\ \tau_D = 0.15 \qquad (6.14)$$

ค่าพารามิเตอร์ที่ใช้ในตัวควบคุมพีไอดี เมื่อกระบวนการมีค่าเวลาคงที่ลดลง จำเป็นที่จะต้องมีการเปลี่ยนแปลงไปด้วยเพื่อให้ผลการตอบสนองลูพปิดที่เป็นไปตามเงื่อนไขเดิมตามความสัมพันธ์ของ Ziegler-Nichols นั่นคือ ค่าผิดพลาดลดลงเป็นอัตราส่วน 1/4 เท่า (Quarter Decay Ratio) รูปที่ 6.16 แสดงให้เห็นถึงความแตกต่างของตัวควบคุมพีไอดีที่มีการเปลี่ยนแปลงค่าพารามิเตอร์ และไม่มีการเปลี่ยนแปลงค่าพารามิเตอร์



รูปที่ 6.16 ผลการตอบสนองลูพปิดภายใต้ตัวควบคุมพีไอดีธรรมดาที่ (1) ค่าพารามิเตอร์คงที่ (2) ค่าพารามิเตอร์เปลี่ยนแปลง



รูปที่ 6.17 ผลการตอบสนองลูพปิดภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ ที่มีการประยุกต์ใช้เจเนติกอัลกอริธึม

รูปที่ 6.16 แสดงให้เห็นว่า การเปลี่ยนแปลงค่าพารามิเตอร์ให้เหมาะสมกับกระบวนการที่เปลี่ยนไปนั้น จะให้ผลการตอบสนองที่ดีกว่า การควบคุมที่มีค่าพารามิเตอร์คงที่ เช่นเดียวกันกับที่เกิดในกรณี 6.3.1 และ 6.3.2 สำหรับ รูปที่ 6.17 แสดงให้เห็นถึงผลการควบคุมกระบวนการที่เกิดขึ้นภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ที่นำเสนอ เมื่อใช้ควบคุมกระบวนการที่มีค่าเวลาคงที่

ของกระบวนการลดลง 25% พบว่าให้การตอบสนองที่ดีกว่าเมื่อเปรียบเทียบกับกรณีตัวควบคุมพีไอดี และเมื่อเปรียบเทียบค่า IAE จะได้ดังตารางที่ 6.4

ตารางที่ 6.4 เปรียบเทียบค่าดัชนี IAE ในกรณีที่ 6.3.3

กรณี	ค่าดัชนี IAE	เปอร์เซ็นต์แตกต่าง
ตัวควบคุมพีไอดีพารามิเตอร์คงที่	0.2026	0.00
ตัวควบคุมพีไอดีพารามิเตอร์เปลี่ยนแปลงได้	0.1649	18.61
ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ที่นำเสนอ	0.1323	34.70

6.3.4 สรุปผลการทดลองในตอนที่ 6.3

การทดสอบตัวควบคุมที่เกิดขึ้นในตอนที่ 6.3.1 ถึง 6.3.3 แสดงให้เห็นว่า เมื่อกระบวนการมีการเปลี่ยนแปลงค่าพารามิเตอร์ไป การปรับปรุงค่าพารามิเตอร์ของตัวควบคุมมีความจำเป็นมาก เนื่องจากทำให้สามารถรักษาผลการตอบสนอง ให้อยู่ภายใต้เงื่อนไขของการควบคุมที่ต้องการได้ตลอดเวลา ดังจะเห็นได้โดยตรงจากรูปที่แสดงเปรียบเทียบ หรือจากค่าดัชนี IAE ที่ได้สรุปไว้ในตาราง 6.2 ถึง 6.4 ซึ่งพบว่าการใช้พารามิเตอร์ของตัวควบคุมพีไอดีคงที่ให้ค่า IAE ที่สูงที่สุด หรือมีสมรรถนะในการควบคุมแย่ที่สุดนั่นเอง

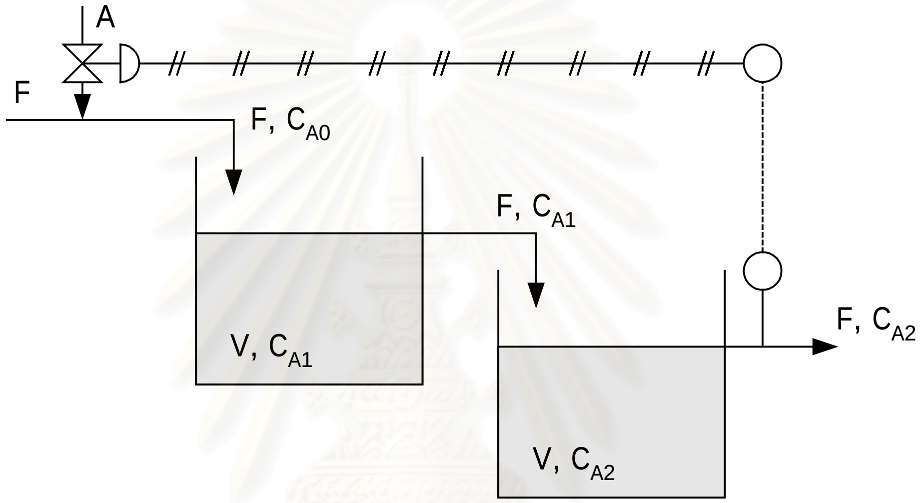
จากการทดสอบ ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ที่มีการประยุกต์ใช้เจเนติกอัลกอริธึมที่ได้นำเสนอนี้ สามารถให้ค่า IAE ที่น้อยกว่าตัวควบคุมพีไอดีธรรมดาที่มีการเปลี่ยนแปลงค่าพารามิเตอร์ นั่นแสดงให้เห็นว่าตัวควบคุมที่นำเสนอมีความสามารถในการปรับค่าพารามิเตอร์ได้เองตามกระบวนการที่เปลี่ยนแปลงไป และสามารถคำนวณหาพารามิเตอร์ที่ให้ผลการตอบสนองที่ดีกว่าในกรณีที่ใช้ความสัมพันธ์ของ Ziegler-Nichols เนื่องจากให้ค่าดัชนี IAE ที่ต่ำกว่า

แต่เนื่องจากการปรับค่าพารามิเตอร์ของตัวควบคุมพีไอดีธรรมดานั้น ต้องอาศัยการประมาณค่ากระบวนการที่หามาจากการกระตุ้นให้เกิดการตอบสนองลูพเปิด แล้วนำผลการตอบสนองที่ได้มาเข้าสู่กระบวนการประมาณค่าด้วยวิธีทางกราฟิก ให้เป็นกระบวนการอันดับหนึ่งที่มีเวลาหน่วง ซึ่งไม่เหมาะสมในทางปฏิบัติที่จะต้องปรับค่าพารามิเตอร์ของกระบวนการบ่อยๆ ดังนั้นจึงเป็นเหตุให้ตัวควบคุมพีไอดีที่นำเสนอ มีความน่าดึงดูดที่จะให้เกิดการนำไปประยุกต์ใช้งานจริงได้มากกว่า และรวมทั้งการประมาณค่าของกระบวนการเกิดขึ้นอย่างต่อเนื่อง แบบออนไลน์ (online) ที่สามารถปรับค่าให้เหมาะสมกับการเปลี่ยนแปลงของกระบวนการได้ในทันที จึงทำให้ผลการตอบสนองมีค่าออกมาดีกว่าที่เกิดขึ้นในตัวควบคุมธรรมดา

6.4 การจำลองการควบคุมบนกระบวนการที่มีการตอบสนองอันดับสองและมีเวลาหน่วง (Second Order Process with Dead Time)

6.4.1 แบบจำลองของกระบวนการ

กระบวนการอันดับสองที่เลือกใช้เพื่อทดสอบตัวควบคุมพีไอดีชนิดปรับค่าเองได้ที่นำเสนอในงานวิจัยตอนที่ 6.4.1 นี้ จะเป็นการควบคุมความเข้มข้นในเครื่องปฏิกรณ์แบบถังกวนต่อเนื่อง ที่ต่ออนุกรมกัน 2 ถัง ดังรูปที่ 6.18



รูปที่ 6.18 กระบวนการควบคุมความเข้มข้นของสาร A ที่ออกจากเครื่องปฏิกรณ์แบบถังกวนต่อเนื่อง (Continuous Stirred Tank Reactor, CSTR)

จากรูปที่ 6.18 สาร A เป็นตัวถูกละลายอยู่ในสถานะของเหลว ถูกป้อนผ่านวาล์วควบคุมด้วยอัตราการไหลที่ต่ำมาก เข้าผสมกับตัวทำละลาย ณ ตำแหน่งป้อนสารเข้าสู่กระบวนการ ซึ่งที่จุดนี้สารละลายมีความเข้มข้น C_{A0} ดังนั้นเราสามารถที่จะเปลี่ยนแปลงความเข้มข้น C_{A0} ได้ด้วยการปรับอัตราการป้อนของสาร A

เมื่อสารละลายผ่านเข้าสู่ในแต่ละเครื่องปฏิกรณ์ สาร A จะเกิดปฏิกิริยาไปเป็นผลิตภัณฑ์ ทำให้ความเข้มข้นของสาร A ลดลง โดยที่สมมุติฐานของระบบ จะเป็นไปตามเงื่อนไขของเครื่องปฏิกรณ์แบบถังกวนต่อเนื่อง ที่มีปริมาตรคงที่ และอุณหภูมิไม่เปลี่ยนแปลง

ในกระบวนการนี้ ความเข้มข้นของสาร A ที่ออกจากเครื่องปฏิกรณ์แบบถังกวนต่อเนื่องถังที่ 2 (C_{A2}) ถูกควบคุมโดยการเปลี่ยนแปลงปริมาณของสาร A ที่จะใช้ผสมกับตัวทำละลายที่มีอัตราการไหลคงที่ ทำให้สรุปได้ว่า

ตัวแปรปรับ (Manipulated Variable) คือ อัตราการไหลของสาร A ขาเข้า (F_A)

ตัวแปรควบคุม(Controlled Variable) คือ ความเข้มข้นของสาร A ที่ออกจากถังที่ 2 (C_{A2}) และกำหนดให้

ตัวแปรรบกวน (Disturbance Variable) คือ อัตราการไหลของตัวทำละลาย (F)

ภายในเครื่องปฏิกรณ์แต่ละถังจะมีการเกิดปฏิกิริยาคือ

$$A \longrightarrow P$$

โดยที่อัตราการเกิดปฏิกิริยาจะเป็นไปตามสมการ

$$r_A = -k\left[C_A\right]$$

เมื่อ k เป็นค่าคงที่ในการเกิดปฏิกิริยาของสาร A

จากกระบวนการตามรูปที่ 6.18 และข้อมูลที่กำหนดให้ ทำให้สามารถเขียนแบบจำลองทางคณิตศาสตร์ของกระบวนการได้ดังนี้

$$\frac{dC_{A1}}{dt} = \frac{F}{V}(C_{A0} - C_{A1}) - k[C_{A1}]$$

$$\frac{dC_{A2}}{dt} = \frac{F}{V}(C_{A1} - C_{A2}) - k[C_{A2}]$$

ในการจำลองกระบวนการนี้ กำหนดค่าคงที่และเงื่อนไขเริ่มต้นของตัวแปรต่างๆ ดังตารางที่ 6.5

ตารางที่ 6.5 ค่าตัวแปรที่ใช้ในแบบจำลองทางคณิตศาสตร์ ณ สภาวะคงที่

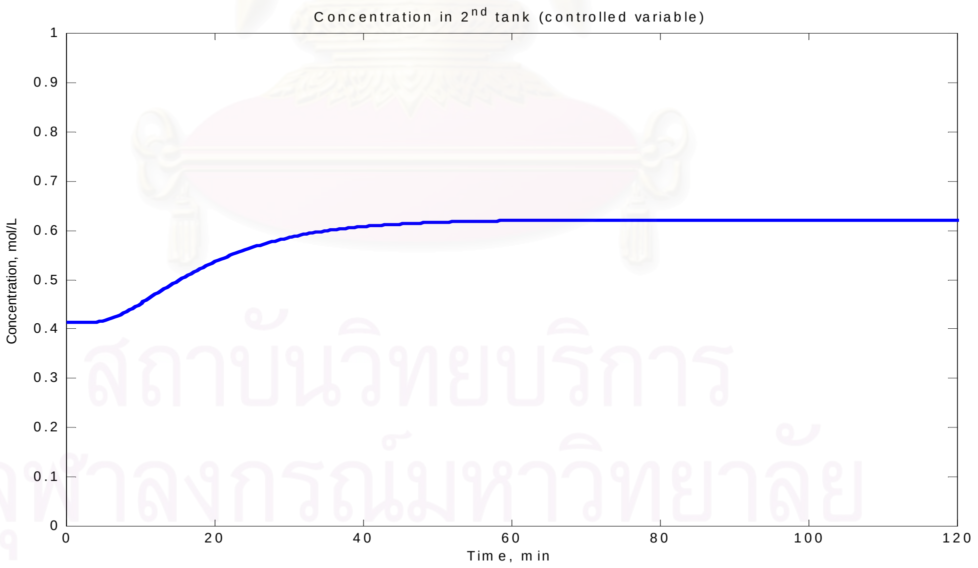
ตัวแปร	หน่วย	ค่าที่ใช้
อัตราการไหล, F	ลิตรต่อนาที	0.085
ปริมาตรของถัง, V	ลิตร	1.05
ความเข้มข้นเริ่มต้นของสาร A ณ สายเข้าถัง 1, C_{A0}	โมลต่อลิตร	0.925
ความเข้มข้นเริ่มต้นของสาร A ณ สายออกจากถัง 1, C_{A1}	โมลต่อลิตร	0.619
ความเข้มข้นเริ่มต้นของสาร A ณ สายออกจากถัง 2, C_{A2}	โมลต่อลิตร	0.414
ค่าคงที่ของการเกิดปฏิกิริยา, k	$(\text{นาที})^{-1}$	0.04

เวลาหน่วง, θ	นาที	4
เวลาในการเก็บตัวอย่าง, T_s	นาที	0.4

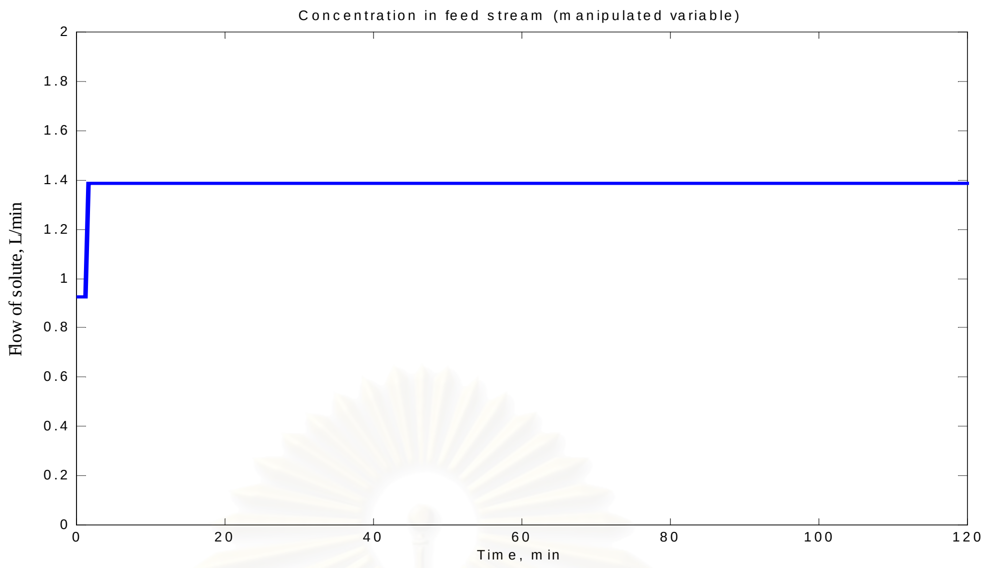
6.4.2 การตอบสนองลูพเปิด (Open-Loop Response)

เพื่อที่จะทำการหาค่าพารามิเตอร์ของตัวควบคุมพีไอดี ที่ใช้ในการควบคุมกระบวนการตามที่กำหนดในหัวข้อ 6.4 สามารถทำได้โดยการประมาณค่าพารามิเตอร์ของกระบวนการดังกล่าวให้อยู่ในรูปของกระบวนการอันดับหนึ่งที่มีเวลาหน่วง (First-Order plus Dead-Time Process) ด้วยการหาการตอบสนองลูพเปิด (Open-loop Response) เช่นเดียวกันกับกรณีที่ 6.3 แล้วจึงแทนค่าพารามิเตอร์ของกระบวนการที่ได้ ลงในความสัมพันธ์ของ Ziegler-Nichols [ภาคผนวก ข.] ซึ่งในที่นี้จะทำการประมาณค่าพารามิเตอร์จากการตอบสนองลูพเปิดด้วยวิธีฟิต 3 ด้วยเช่นกัน

ผลจากการหา ผลการตอบสนองลูพเปิด ของกระบวนการในตอนที่ 6.4 นี้ แสดงอยู่ในรูปที่ 6.19 ซึ่งเป็นการตอบสนองเมื่อที่การเปลี่ยนแปลงตัวแปรอินพุทตามที่แสดงในรูปที่ 6.20



รูปที่ 6.19 การตอบสนองลูพเปิดของกระบวนการในหัวข้อ 6.4.1



รูปที่ 6.20 ตัวแปรอินพุทที่เปลี่ยนแปลงแบบขั้น เพื่อหาผลการตอบสนองลูพเปิดของระบบ

เมื่อประมาณค่าพารามิเตอร์ของกระบวนการ (โดยใช้รูปที่ 6.19 และ 6.20) ให้อยู่ในรูปของกระบวนการอันดับหนึ่งที่มีเวลาหน่วง โดยใช้วิธีฟิต 3 จะได้พารามิเตอร์ของกระบวนการเป็นดังนี้

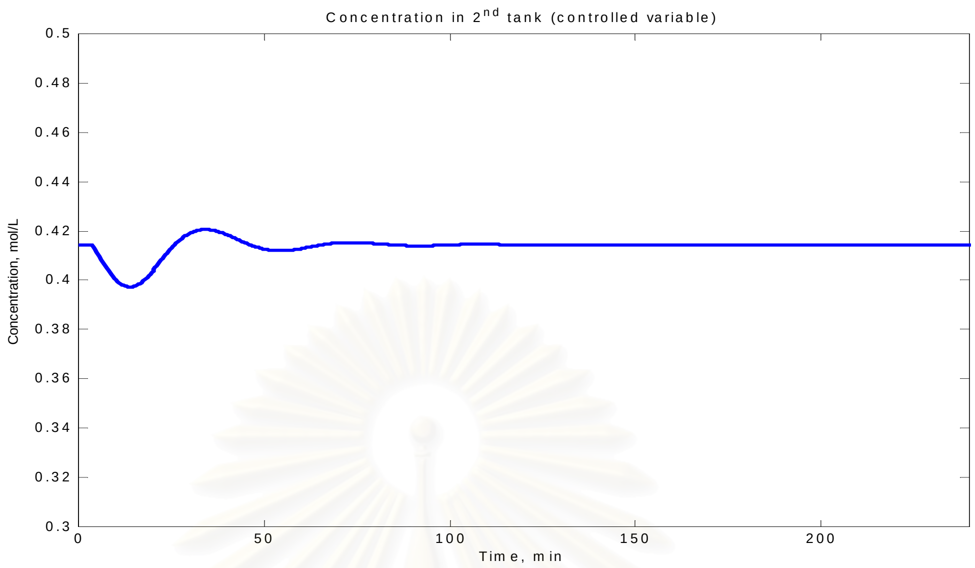
$$K_p = 0.4479\,;\ \tau = 14.4\,;\ \theta = 5.6 \qquad (6.15)$$

ซึ่งเมื่อคำนวณหาค่าพารามิเตอร์ของตัวควบคุมพีไอดีตามวิธีของ Ziegler-Nichols จะได้ว่า

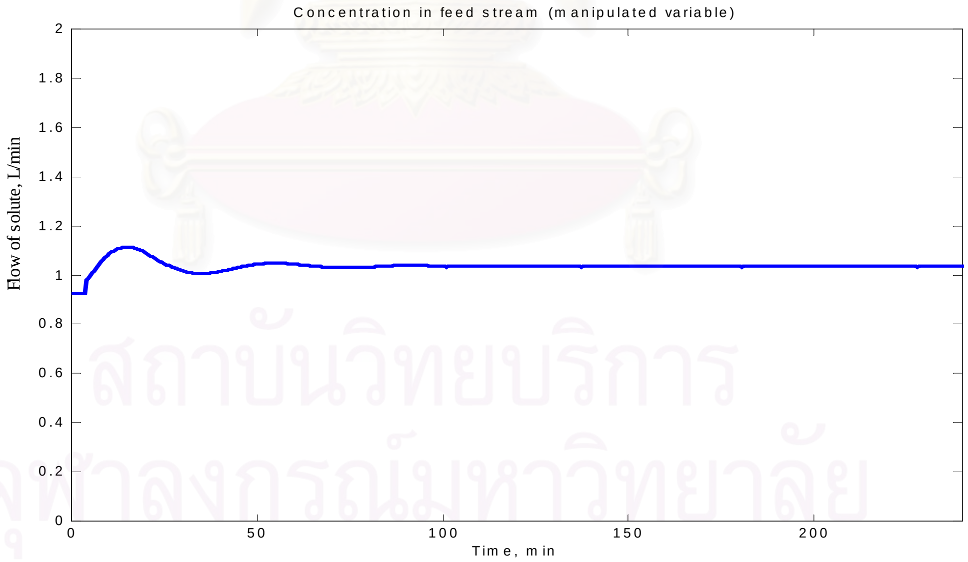
$$K_C = 6.8886\,;\ \tau_I = 11.2\,;\ \tau_D = 2.8 \qquad (6.16)$$

ซึ่งพารามิเตอร์ที่ได้จะใช้ในตัวควบคุมพีไอดี เพื่อควบคุมกระบวนการอันดับสองที่มีเวลาหน่วงในตอนต่อไป

6.4.3 การตอบสนองภายใต้ตัวควบคุมพีไอดี เมื่อตัวแปรรบกวน (F) ลดลง 15%



รูปที่ 6.21 การตอบสนองของความเข้มข้นของสาร A ที่ออกจากถังที่ 2 ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่ออัตราการไหลของตัวทำละลาย (F) ลดลง 15%

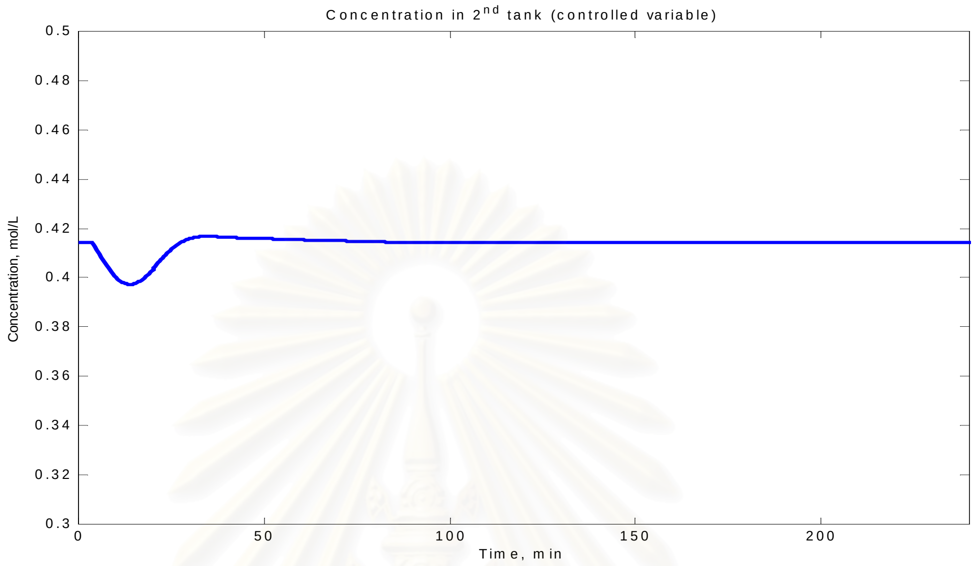


รูปที่ 6.22 อัตราการป้อนสาร A (ตัวแปรปรับ) ณ สายขาเข้าของถังที่ 1 ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่ออัตราการไหลของตัวทำละลาย (F) ลดลง 15%

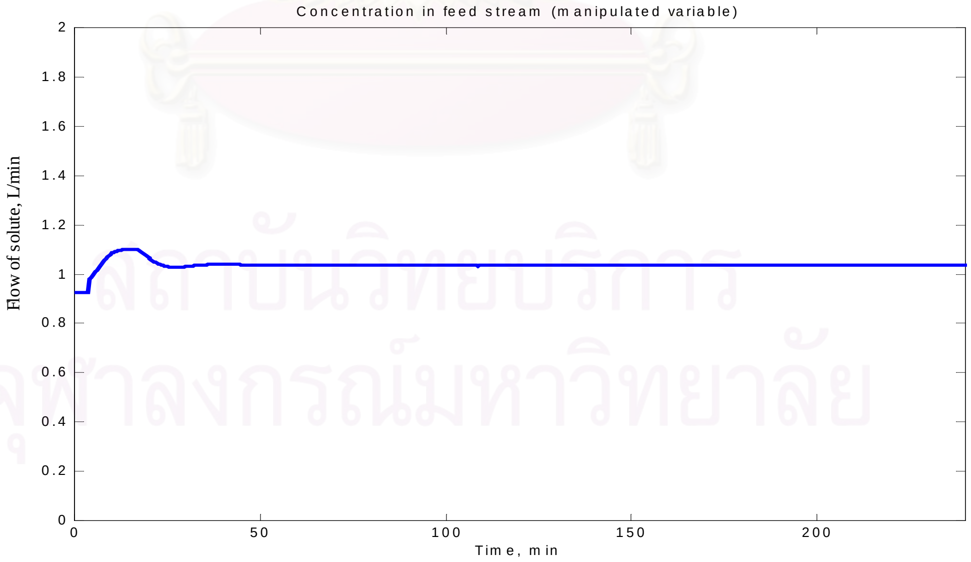
สามารถคำนวณค่าดัชนีอินทีกรัลของค่าผิดพลาดสัมบูรณ์ (IAE) ของกราฟในรูปที่ 6.21 ได้มีค่าเท่ากับ 0.3634

การเปรียบเทียบค่า IAE ของตัวควบคุมทั้งสอง ในแต่ละกรณี แสดงในตอนที่ 6.4.11

6.4.4 การตอบสนองภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่อตัวแปรรบกวน (F) ลดลง 15%



รูปที่ 6.23 การตอบสนองของความเข้มข้นของสาร A ที่ออกจากถังที่ 2 ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่ออัตราการไหลของตัวทำละลาย (F) ลดลง 15%

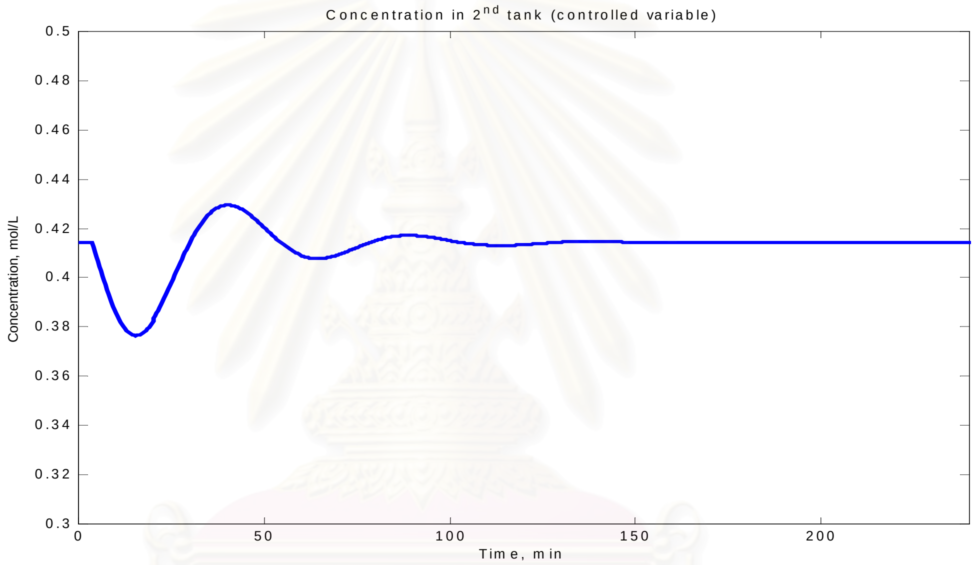


รูปที่ 6.24 อัตราการป้อนสาร A (ตัวแปรปรับ) ณ สายขาเข้าของถังที่ 1 ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่ออัตราการไหลของตัวทำละลาย (F) ลดลง 15%

สามารถคำนวณค่าดัชนีอินทีกรัลของค่าผิดพลาดสัมบูรณ์ (IAE) ของกราฟในรูปที่ 6.23 ได้มีค่าเท่ากับ 0.3097

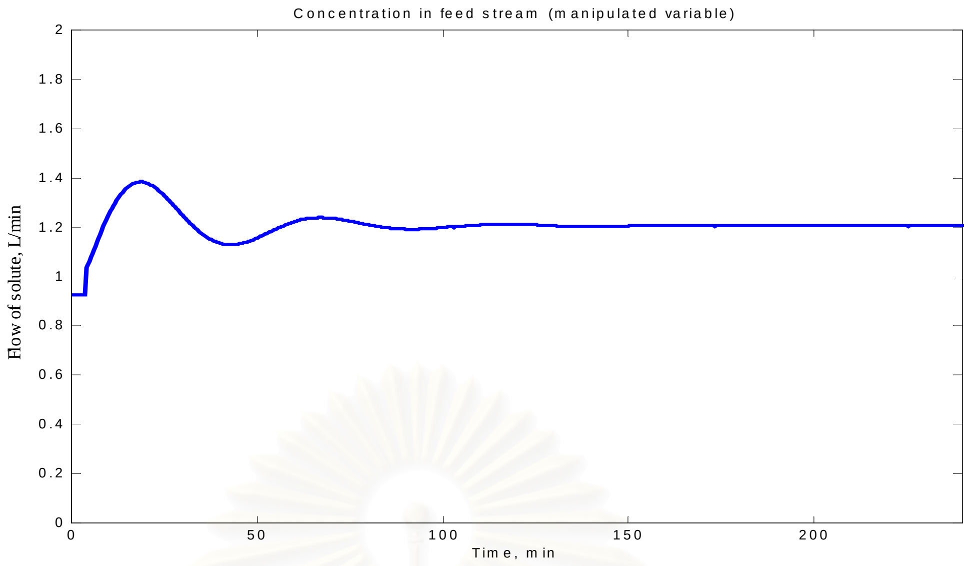
จากค่า IAE ที่คำนวณได้ในกรณีนี้ สามารถเปรียบเทียบได้ว่า ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ดังที่นำเสนอ และตัวควบคุมพีไอดีธรรมดา มีค่า IAE ที่ไม่แตกต่างกันมากนัก หรือกล่าวว่ามีการปรับแต่งค่าพารามิเตอร์ของตัวควบคุม ที่ไม่เอนเอียงไปด้านใดด้านหนึ่ง

6.4.5 การตอบสนองภายใต้ตัวควบคุมพีไอดี เมื่อตัวแปรรบกวน (F) ลดลง 30%



รูปที่ 6.25 การตอบสนองของความเข้มข้นของสาร A ที่ออกจากถังที่ 2 ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่ออัตราการไหลของตัวทำละลาย (F) ลดลง 30%

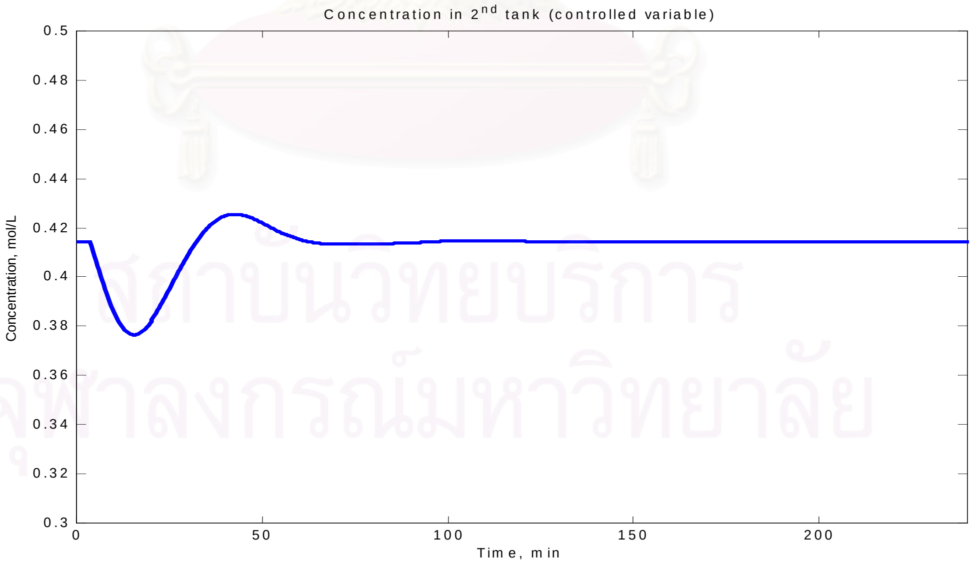
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



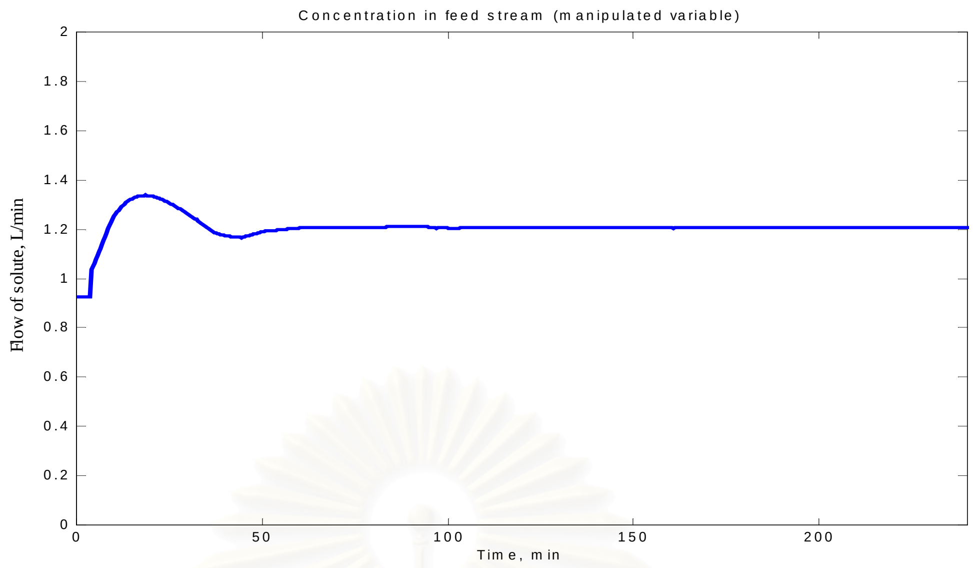
รูปที่ 6.26 อัตราการป้อนสาร A (ตัวแปรปรับ) ณ สายขาเข้าของถังที่ 1 ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่ออัตราการไหลของตัวทำละลาย (F) ลดลง 30%

สามารถคำนวณค่าดัชนีอินทีกรัลของค่าผิดพลาดสัมบูรณ์ (IAE) ของกราฟในรูปที่ 6.25 ได้มีค่าเท่ากับ 1.0216

6.4.6 การตอบสนองภายใต้ตัวควบคุมพีไอดีชนิดปรับตัวเองได้ เมื่อตัวแปรรบกวน (F) ลดลง 30%



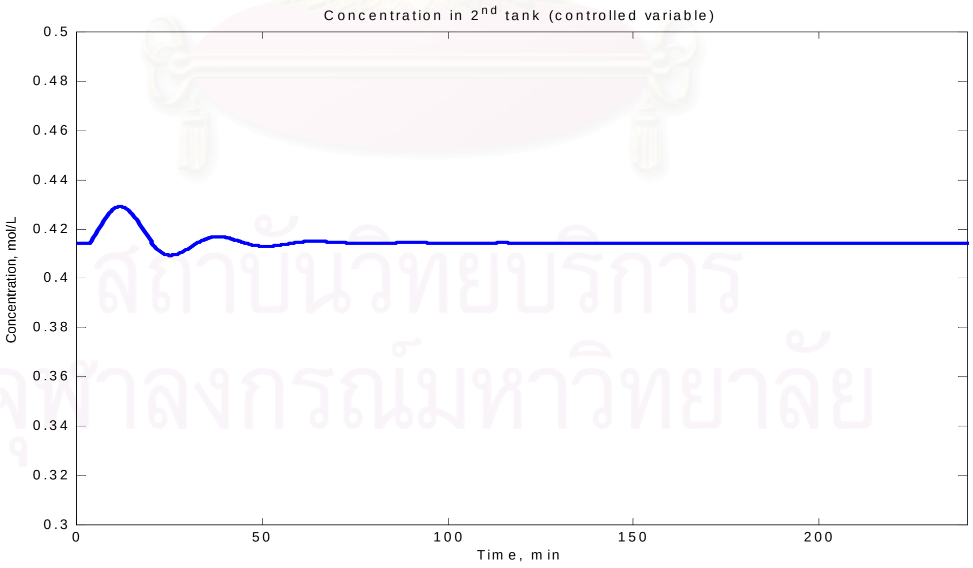
รูปที่ 6.27 การตอบสนองของความเข้มข้นของสาร A ที่ออกจากถังที่ 2 ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่ออัตราการไหลของตัวทำละลาย (F) ลดลง 30%



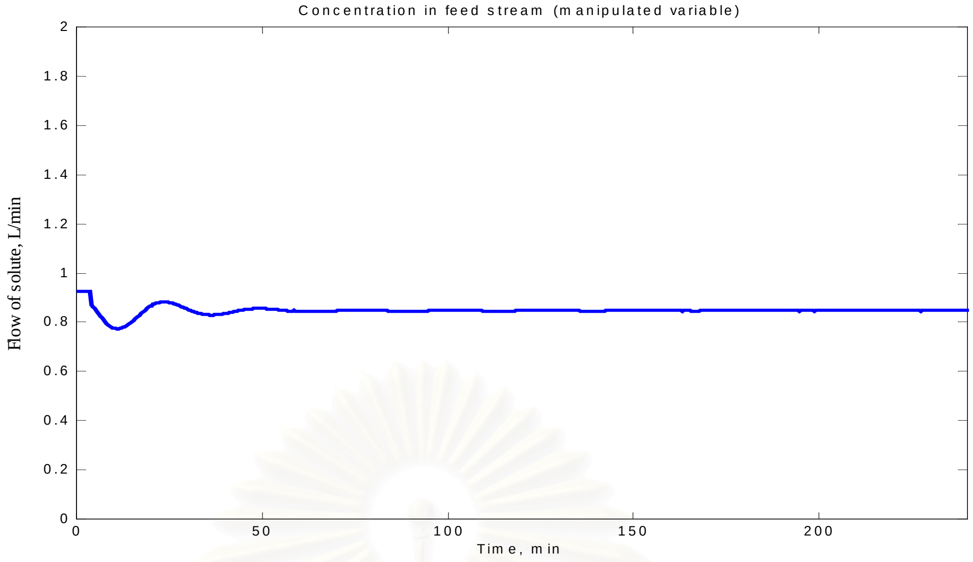
รูปที่ 6.28 อัตราการป้อนสาร A (ตัวแปรปรับ) ณ สายขาเข้าของถังที่ 1 ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่ออัตราการไหลของตัวทำละลาย (F) ลดลง 30%

สามารถคำนวณค่าดัชนีอินทีกรัลของค่าผิดพลาดสัมบูรณ์ (IAE) ของกราฟในรูปที่ 6.27 ได้มีค่าเท่ากับ 0.8783

6.4.7 การตอบสนองภายใต้ตัวควบคุมพีไอดี เมื่อตัวแปรรบกวน (F) เพิ่มขึ้น 15%



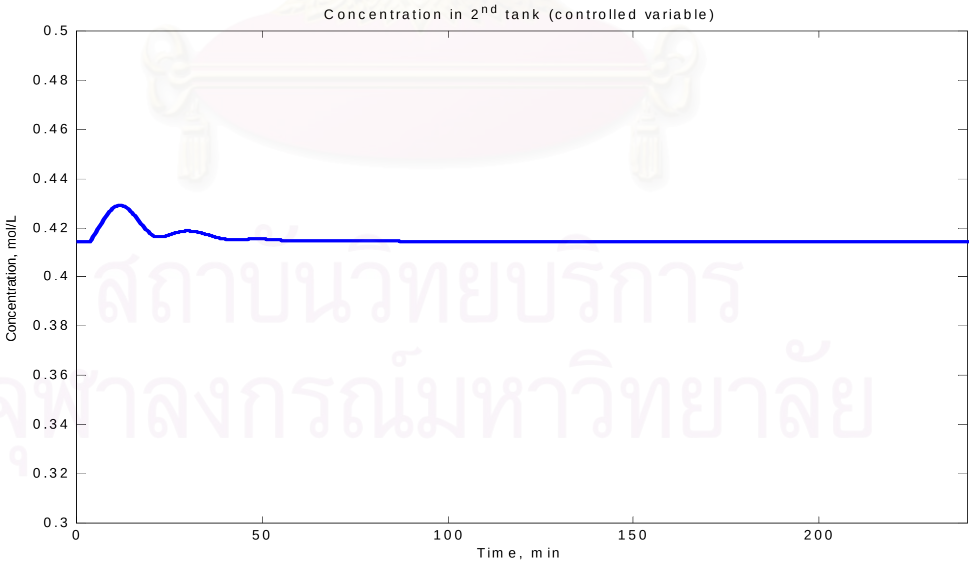
รูปที่ 6.29 การตอบสนองของความเข้มข้นของสาร A ที่ออกจากถังที่ 2 ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่ออัตราการไหลของตัวทำละลาย (F) เพิ่มขึ้น 15%



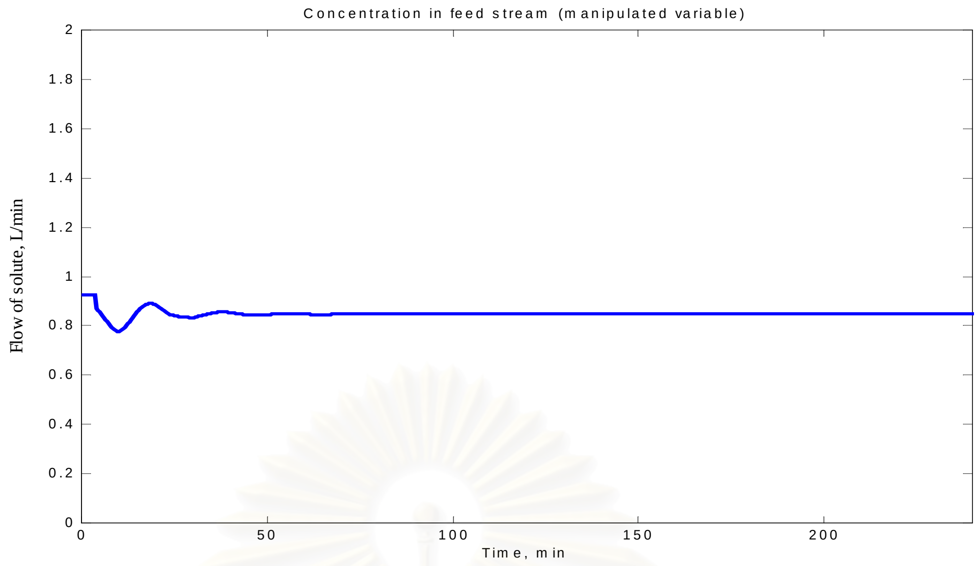
รูปที่ 6.30 อัตราการป้อนสาร A (ตัวแปรปรับ) ณ สายขาเข้าของถังที่ 1 ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่ออัตราการไหลของตัวทำละลาย (F) เพิ่มขึ้น 15%

สามารถคำนวณค่าดัชนีอินทีกรัลของค่าผิดพลาดสัมบูรณ์ (IAE) ของกราฟในรูปที่ 6.29 ได้มีค่าเท่ากับ 0.2362

6.4.8 การตอบสนองภายใต้ตัวควบคุมพีไอดีชนิดปรับตัวเองได้ เมื่อตัวแปรรบกวน (F) เพิ่มขึ้น 15%



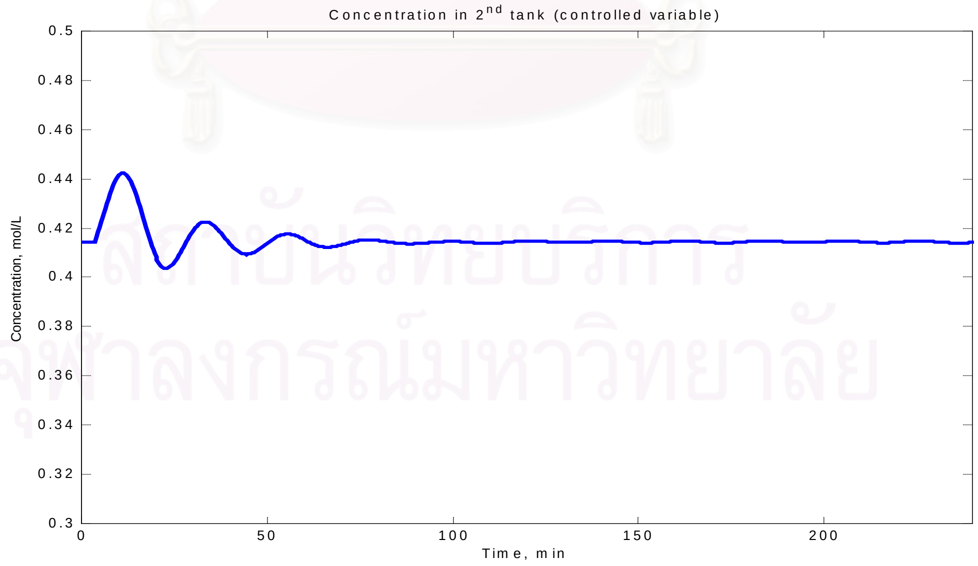
รูปที่ 6.31 การตอบสนองของความเข้มข้นของสาร A ที่ออกจากถังที่ 2 ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่ออัตราการไหลของตัวทำละลาย (F) เพิ่มขึ้น 15%



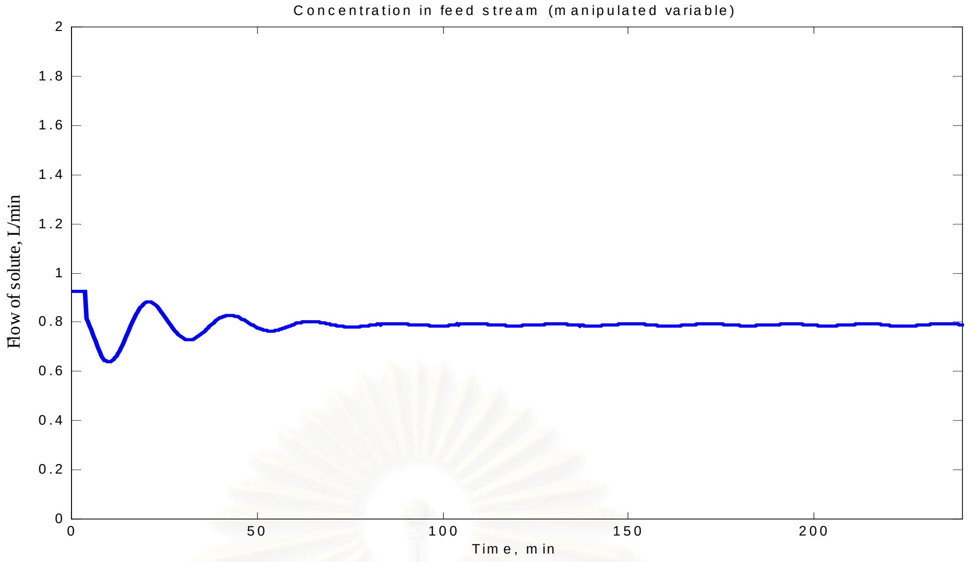
รูปที่ 6.32 อัตราการป้อนสาร A (ตัวแปรปรับ) ณ สายขาเข้าของถังที่ 1 ภายใต้ตัวควบคุมพีไอดี ชนิดปรับค่าเองได้ เมื่ออัตราการไหลของตัวทำละลาย (F) เพิ่มขึ้น 15%

สามารถคำนวณค่าดัชนีอินทิกรัลของค่าผิดพลาดสัมบูรณ์ (IAE) ของกราฟในรูปที่ 6.31 ได้มีค่าเท่ากับ 0.2352

6.4.9 การตอบสนองภายใต้ตัวควบคุมพีไอดี เมื่อตัวแปรรบกวน (F) เพิ่มขึ้น 30%



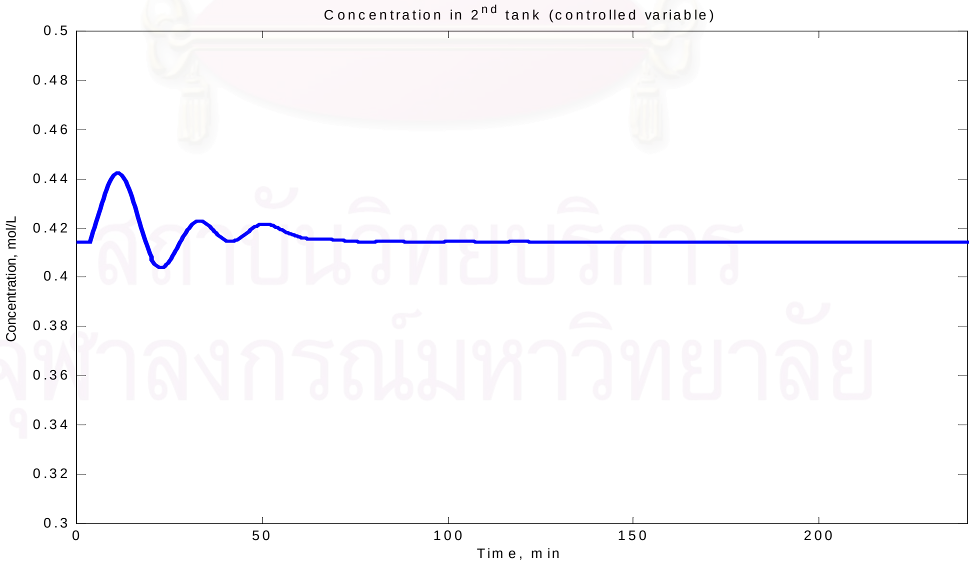
รูปที่ 6.33 การตอบสนองของความเข้มข้นของสาร A ที่ออกจากถังที่ 2 ภายใต้ตัวควบคุมพีไอดี ธรรมดา เมื่ออัตราการไหลของตัวทำละลาย (F) เพิ่มขึ้น 30%



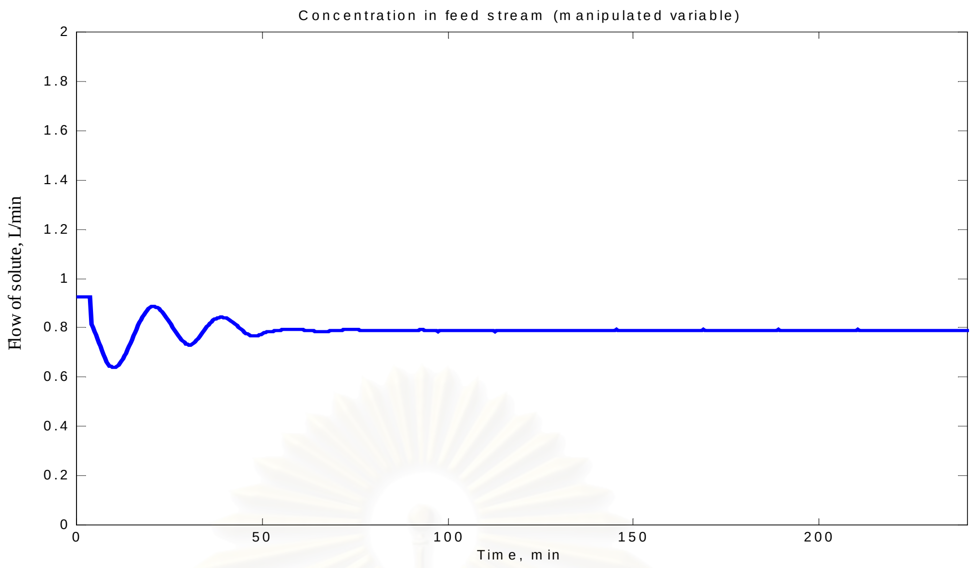
รูปที่ 6.34 อัตราการป้อนสาร A (ตัวแปรปรับ) ณ สายขาเข้าของถังที่ 1 ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่ออัตราการไหลของตัวทำละลาย (F) เพิ่มขึ้น 30%

สามารถคำนวณค่าดัชนีอินทีกรัลของค่าผิดพลาดสัมบูรณ์ (IAE) ของกราฟในรูปที่ 6.33 ได้มีค่าเท่ากับ 0.4937

6.4.10 การตอบสนองภายใต้ตัวควบคุมพีไอดีชนิดปรับตัวเองได้ เมื่อตัวแปรรบกวน (F) เพิ่มขึ้น 30%



รูปที่ 6.35 การตอบสนองของความเข้มข้นของสาร A ที่ออกจากถังที่ 2 ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่ออัตราการไหลของตัวทำละลาย (F) เพิ่มขึ้น 30%



รูปที่ 6.36 อัตราการป้อนสาร A (ตัวแปรปรับ) ณ สายขาเข้าของถังที่ 1 ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่ออัตราการไหลของตัวทำละลาย (F) เพิ่มขึ้น 30%

สามารถคำนวณค่าดัชนีอินทีกรัลของค่าผิดพลาดสัมบูรณ์ (IAE) ของกราฟในรูปที่ 6.35 ได้มีค่าเท่ากับ 0.4856

6.4.11 สรุปการทดลองในตอนที่ 6.4

การทดสอบเปรียบเทียบสมรรถนะลูพปิดของตัวควบคุม บนกระบวนการอันดับสองที่มีเวลาหน่วง ด้วยตัวควบคุมพีไอดีธรรมดา และตัวควบคุมพีไอดีชนิดปรับค่าเองได้ที่มีการประยุกต์ใช้เจเนติกอัลกอริธึมดังที่นำเสนอ โดยการเปรียบเทียบค่าดัชนีอินทีกรัลของความผิดพลาดสัมบูรณ์ (IAE) สรุปไว้ในตารางที่ 6.6

ตารางที่ 6.6 สรุปค่าดัชนี IAE ของการควบคุมกระบวนการอันดับสองที่มีเวลาหน่วง

กรณีที่ศึกษา	ตัวควบคุมพีไอดีธรรมดา	ตัวควบคุมพีไอดีชนิดปรับค่าเองได้	เปอร์เซ็นต์แตกต่าง
ตัวแปรรบกวนเพิ่มขึ้น 30%	0.4937	0.4856	-1.64
ตัวแปรรบกวนเพิ่มขึ้น 15%	0.2362	0.2352	-0.42
ตัวแปรรบกวนลดลง 15%	0.3634	0.3097	-14.78
ตัวแปรรบกวนลดลง 30%	1.0216	0.8783	-14.03

เห็นได้จากตารางที่ 6.6 ว่าตัวควบคุมพีไอดีชนิดปรับค่าเองได้ โดยใช้เจเนติกอัลกอริธึม สามารถควบคุมกระบวนการที่มีผลการตอบสนองอันดับสองที่มีเวลาหน่วง ได้เช่นเดียวกันกับที่ตัวควบคุมพีไอดีธรรมดาสามารถทำได้ โดยที่มีค่าดัชนี IAE แตกต่างกันเพียงเล็กน้อยเท่านั้น ทั้งนี้เป็นเพราะกระบวนการที่มีผลการตอบสนองอันดับสองที่มีเวลาหน่วงนั้น เป็นกระบวนการที่มีแบบจำลองของกระบวนการคงที่ตลอดเวลา ดังนั้นค่าพารามิเตอร์ที่ได้จากการปรับปรุงค่าในแต่ละรอบของการคำนวณ จึงมีค่าที่ใกล้เคียงหรือเท่ากับค่าที่ใช้ในกรณีของการควบคุมด้วยตัวควบคุมพีไอดีธรรมดา ผลที่ได้จากการควบคุมด้วยตัวควบคุมทั้งสองจึงไม่แตกต่างกันมากนัก

ความแตกต่างของผลการควบคุมที่แสดงให้เห็นผ่านทางดัชนี IAE ที่เกิดขึ้น เป็นเพราะจุดเด่นของการค้นหาคำตอบด้วยเจเนติกอัลกอริธึม ที่สามารถจะสุ่มตรวจและค้นหาค่าพารามิเตอร์ที่เหมาะสมมากกว่าค่าพารามิเตอร์ที่ได้จากการคำนวณตามหลักของ Ziegler-Nichols นั่นเอง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

6.5 การทดสอบการควบคุมบนกระบวนการที่มีความไม่เป็นเชิงเส้น (Nonlinear Process)

กระบวนการที่มีความไม่เป็นเชิงเส้น ซึ่งนำมาใช้ในการทดสอบตัวควบคุมนี้ เป็นกระบวนการปรับค่าพีเอชให้มีสภาพเป็นกลางที่ได้มีการอธิบายไว้อย่างละเอียดแล้วในบทที่ 5 ดังนั้นผลการทดลองในตอนที่ 6.5 นี้จึงไม่แสดงในส่วนของแบบจำลองของระบบอีก

6.5.1 การประมาณกระบวนการ

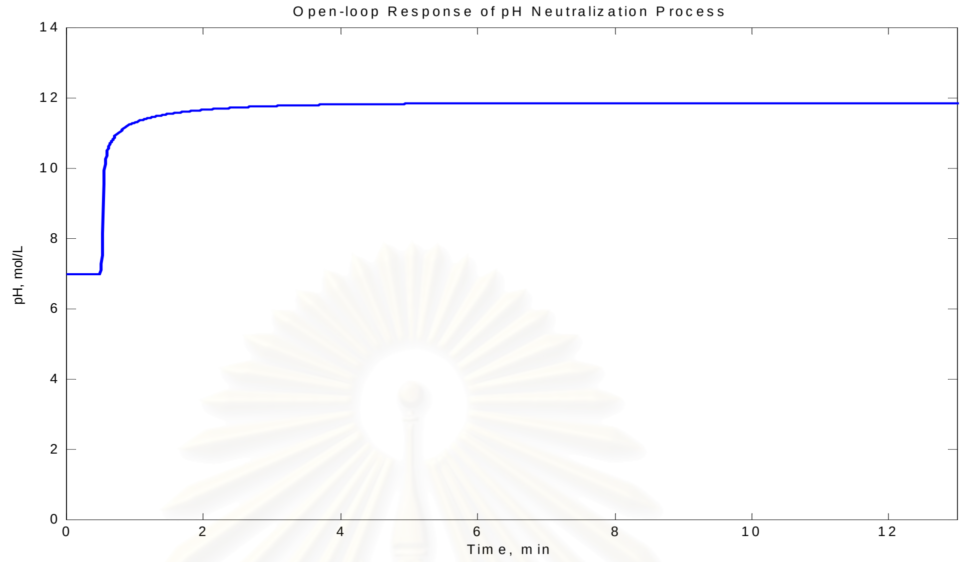
ดังที่ได้แสดงไว้ในรูปที่ 5.2 ถึงความไม่เป็นเชิงเส้นของกระบวนการปรับค่าพีเอชให้มีสภาพเป็นกลางแล้วว่า เป็นกระบวนการที่มีความแตกต่างกันของค่าเกน ของกระบวนการสูงมาก ดังนั้นการประมาณค่าพารามิเตอร์ของกระบวนการ ให้เป็นกระบวนการอันดับหนึ่งที่มีเวลาหน่วงนั้น จะต้องพิจารณาถึงค่าที่สามารถเป็นตัวแทนของกระบวนการได้อีกด้วย พิจารณาตารางที่ 6.7

ตารางที่ 6.7 ค่าพารามิเตอร์ที่ได้จากการประมาณกระบวนการด้วยวิธีฟิต 3 เมื่อมีการเปลี่ยนแปลงตัวแปรอินพุทแบบขั้น หลายๆ ค่า

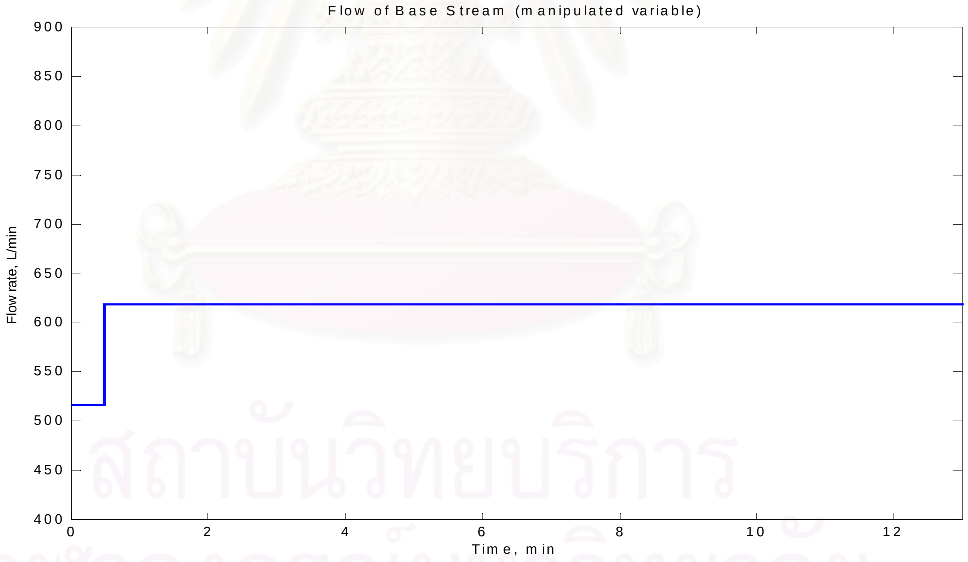
ขนาดของการเปลี่ยนแปลงอินพุท	เกนของกระบวนการจากการประมาณ	เวลาคงที่ของกระบวนการจากการประมาณ	เวลาหน่วงของกระบวนการจากการประมาณ
+5%	0.1656	0.105	0.125
+10%	0.0887	0.060	0.060
+15%	0.0613	0.060	0.030
+20%	0.0471	0.045	0.025
+25%	0.0383	0.045	0.015
+30%	0.0323	0.045	0.005
+35%	0.0280	0.045	0.005
+40%	0.0247	0.030	0.010

และเมื่อพิจารณากราฟของการไตเตรท (รูปที่ 5.2) แล้ว พบว่าที่ขนาดการเปลี่ยนแปลงของอินพุทเพิ่มขึ้น 20% จะให้ค่าพีเอชตกอยู่บนช่วงที่มีการเปลี่ยนแปลงของความชันของเส้นกราฟพอดี ซึ่งเป็นจุดที่น่าจะเป็นตัวแทนของความไม่เป็นเชิงเส้นของกระบวนการที่เห็นได้อย่างชัดเจน ดังนั้นจึงเลือกค่าการประมาณกระบวนการเป็นกระบวนการอันดับหนึ่งที่มีเวลาหน่วง โดยการคำนวณจาก

การตอบสนองลูพเปิด เมื่อมีการเปลี่ยนแปลงอินพุทเพิ่มขึ้น 20% ผลการตอบสนองลูพเปิดจะเป็นดังรูปที่ 6.37



รูปที่ 6.37 การตอบสนองของค่าพีเอช เมื่อมีการเปลี่ยนแปลงตัวแปรอินพุท ดังรูปที่ 6.38



รูปที่ 6.38 การเปลี่ยนแปลงอัตราการไหลของสายด่างแบบขั้น เพื่อสังเกตการตอบสนองวงเปิดของค่าพีเอช (ตัวแปรควบคุม)

และจะสามารถคำนวณค่าพารามิเตอร์ของกระบวนการ ด้วยวิธีฟิต 3 ได้ดังนี้

$$K_p = 0.0471\,;\ \tau = 0.045\,;\ \theta = 0.025 \qquad (6.17)$$

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

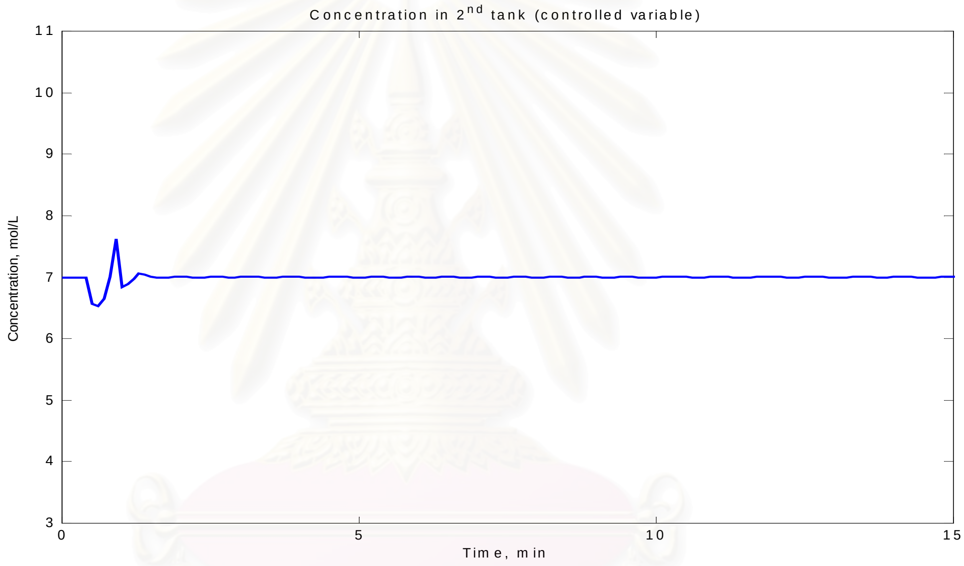
เมื่อคำนวณหาค่าพารามิเตอร์ของตัวควบคุมพีไอดีตามวิธีของ Ziegler-Nichols จะได้ว่า

$$K_C = 45.8839\,;\ \tau_I = 0.05\,;\ \tau_D = 0.0125 \qquad (6.18)$$

ซึ่งพารามิเตอร์ที่ได้จะใช้ในตัวควบคุมพีไอดี เพื่อควบคุมกระบวนการที่มีความไม่เป็นเชิงเส้นสูง (กระบวนการควบคุมค่าพีเอช) ในตอนต่อไป

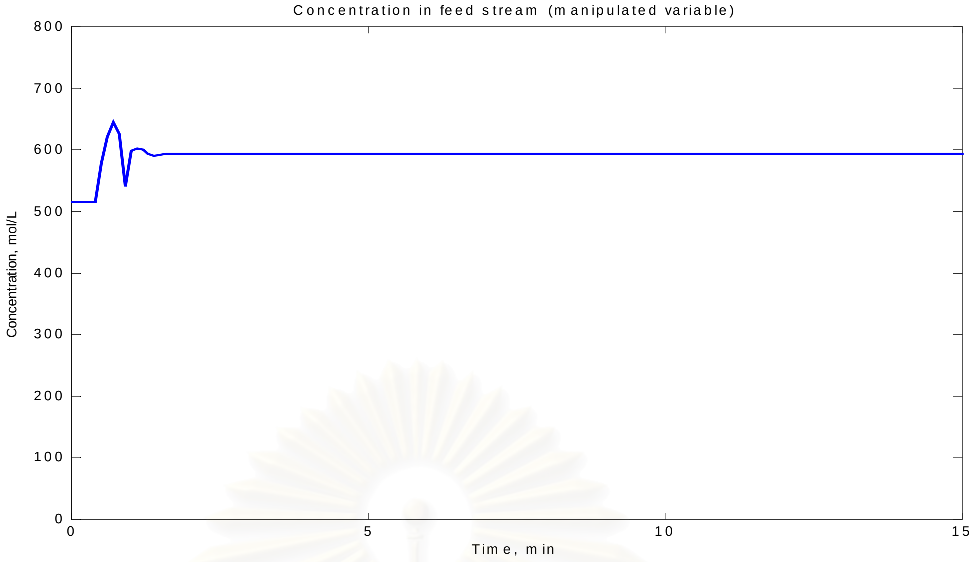
6.5.2 การตอบสนองเมื่อมีตัวแปรรบกวนคือ อัตราการไหลของสารละลายกรด (F_1) เพิ่มขึ้น 15%

(ก) การตอบสนองภายใต้ตัวควบคุมพีไอดีธรรมดา



รูปที่ 6.39 การตอบสนองของค่าพีเอช (ตัวแปรควบคุม) ที่ออกจากเครื่องปฏิกรณ์ ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่ออัตราการไหลของกรด (F1) เพิ่มขึ้น 15%

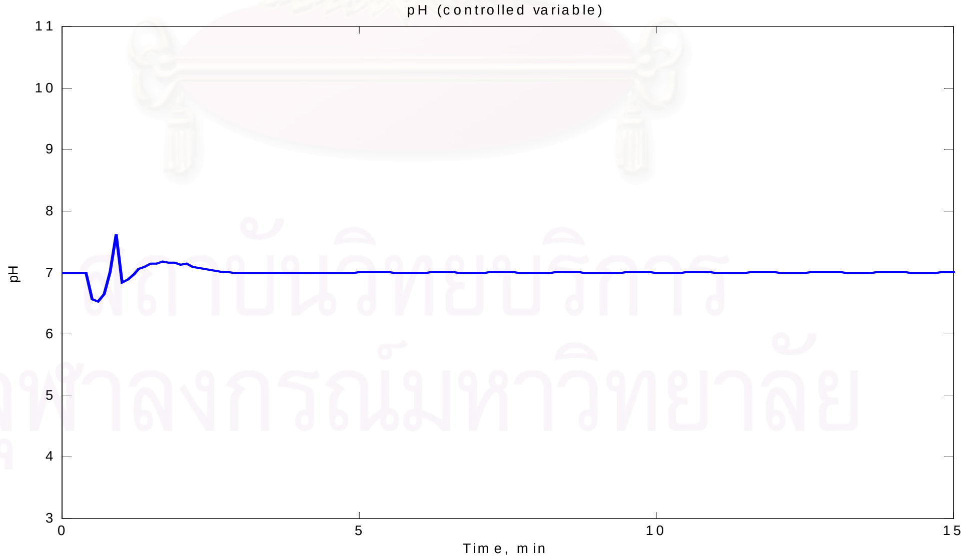
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 6.40 อัตราการป้อนของสายด่าง (ตัวแปรปรับ) เพื่อควบคุมค่าพีเอช ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่ออัตราการไหลของสายกรด (F1) เพิ่มขึ้น 15%

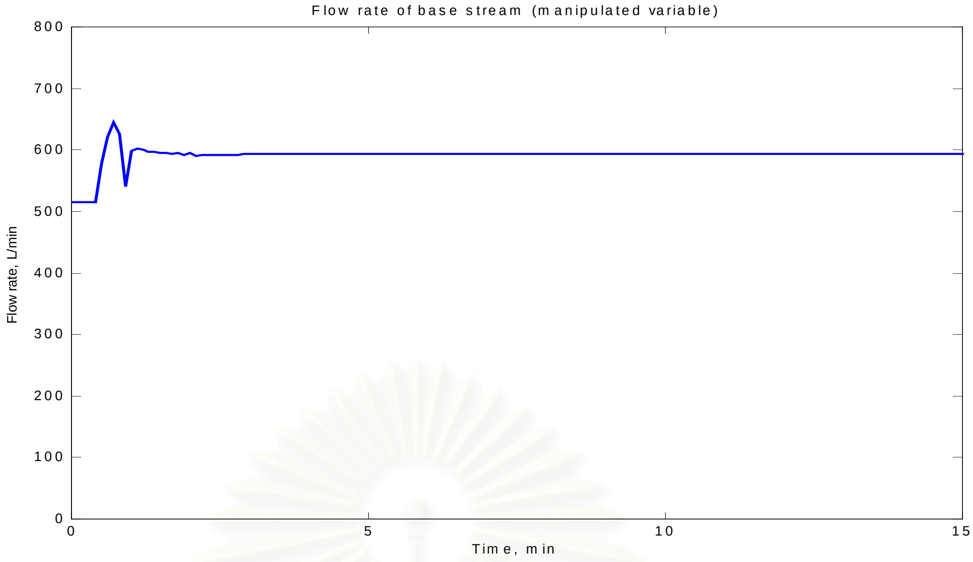
สามารถคำนวณค่าดัชนีอินทีกรัลของค่าผิดพลาดสัมบูรณ์ (IAE) ของกราฟในรูปที่ 6.39 ได้มีค่าเท่ากับ 0.2403

(ข) การตอบสนองภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้



รูปที่ 6.41 การตอบสนองของค่าพีเอช (ตัวแปรควบคุม) ที่ออกจากเครื่องปฏิกรณ์ ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่ออัตราการไหลของกรด (F1) เพิ่มขึ้น 15%

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

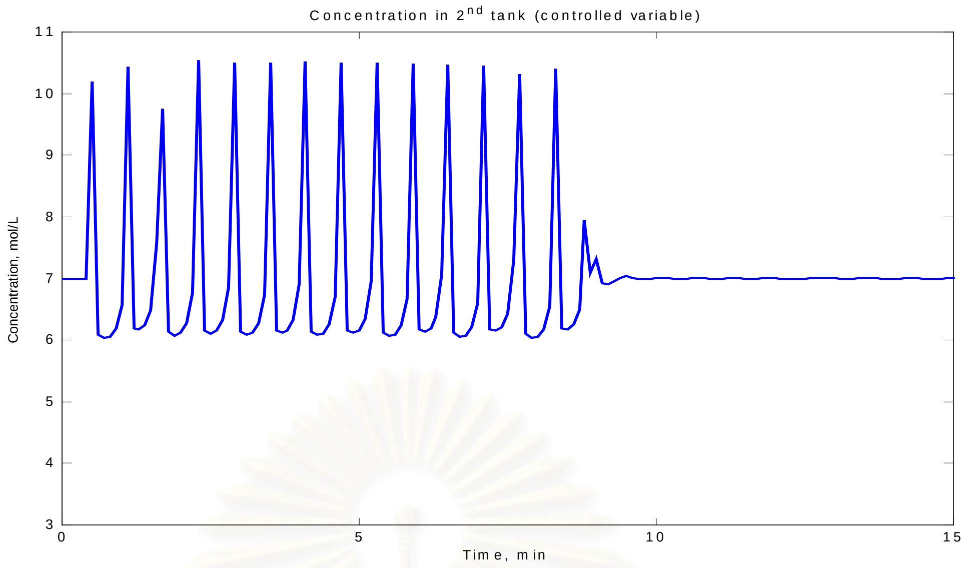


รูปที่ 6.42 อัตราการป้อนของสายด่าง (ตัวแปรปรับ) เพื่อควบคุมค่าพีเอช ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่ออัตราการไหลของสายกรด (F1) เพิ่มขึ้น 15%

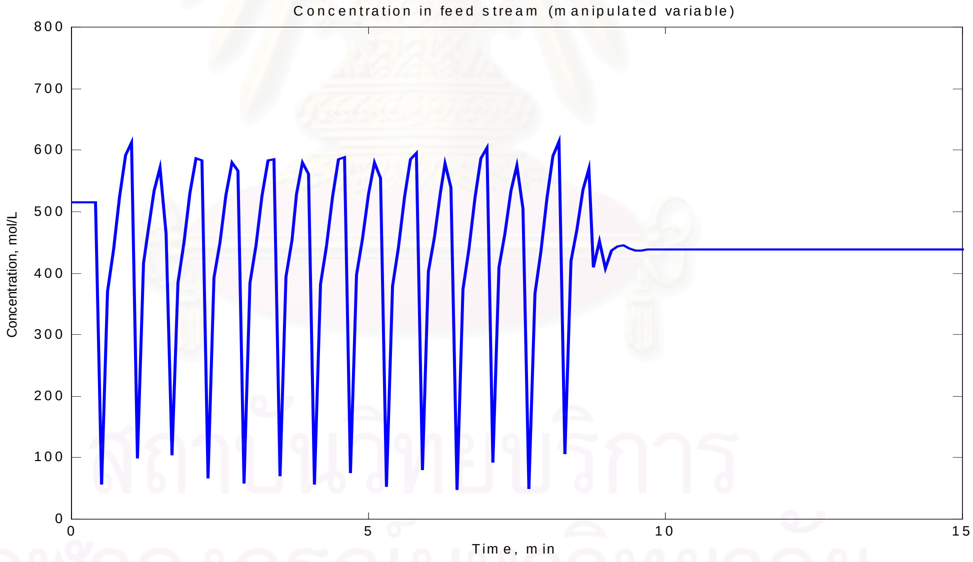
สามารถคำนวณค่าดัชนีอินทีกรัลของค่าผิดพลาดสัมบูรณ์ (IAE) ของกราฟในรูปที่ 6.42 ได้มีค่าเท่ากับ 0.3891

6.5.3 การตอบสนองเมื่อมีตัวแปรรบกวนคือ อัตราการไหลของสารละลายกรด(F_1)ลดลง 15%

(ก) การตอบสนองภายใต้ตัวควบคุมพีไอดีธรรมดา



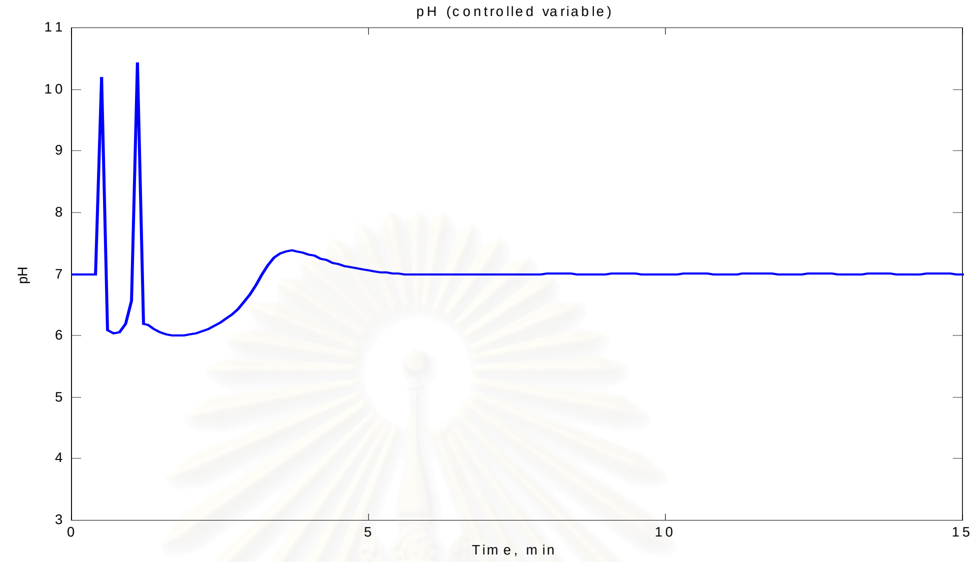
รูปที่ 6.43 การตอบสนองของค่าพีเอช (ตัวแปรควบคุม) ที่ออกจากเครื่องปฏิกรณ์ ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่ออัตราการไหลของกรด (F1) ลดลง 15%



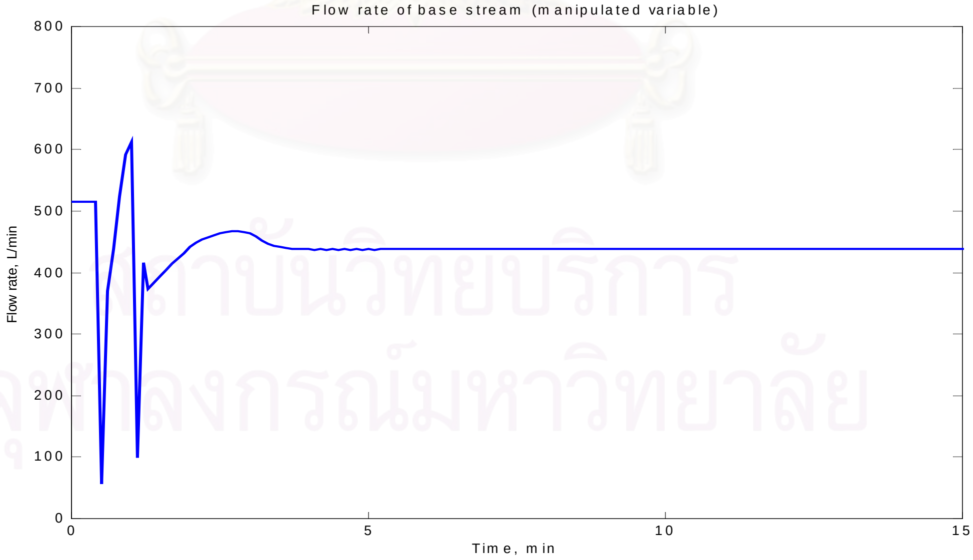
รูปที่ 6.44 อัตราการป้อนของสายด่าง (ตัวแปรปรับ) เพื่อควบคุมค่าพีเอช ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่ออัตราการไหลของสายกรด (F1) ลดลง 15%

สามารถคำนวณค่าดัชนีอินทิกรัลของค่าผิดพลาดสัมบูรณ์ (IAE) ของกราฟในรูปที่ 6.43 ได้มีค่าเท่ากับ 9.9077

(ข) การตอบสนองภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้



รูปที่ 6.45 การตอบสนองของค่าพีเอช (ตัวแปรควบคุม) ที่ออกจากเครื่องปฏิกรณ์ ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่ออัตราการไหลของกรด (F1) ลดลง 15%

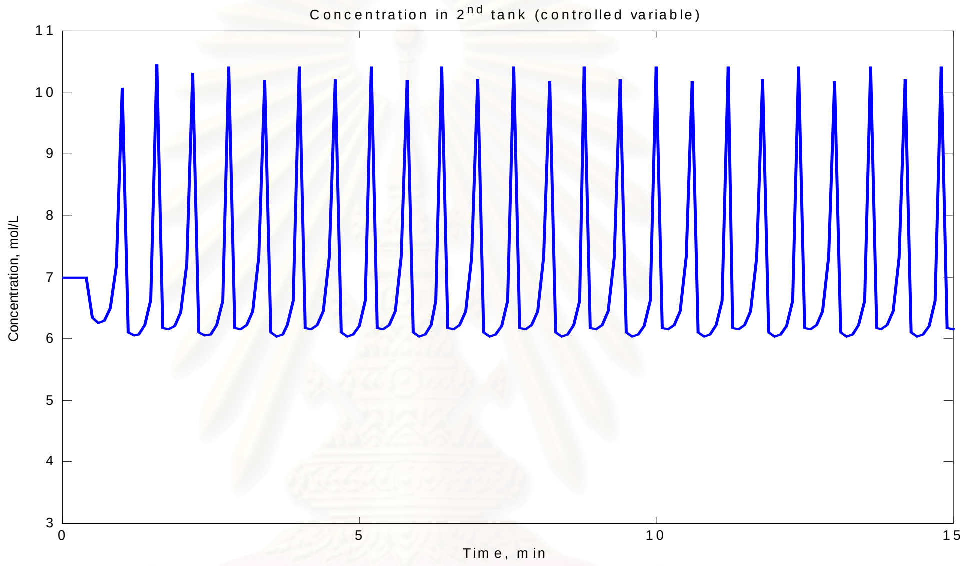


รูปที่ 6.46 อัตราการป้อนของสายด่าง (ตัวแปรปรับ) เพื่อควบคุมค่าพีเอช ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่ออัตราการไหลของสายกรด (F1) ลดลง 15%

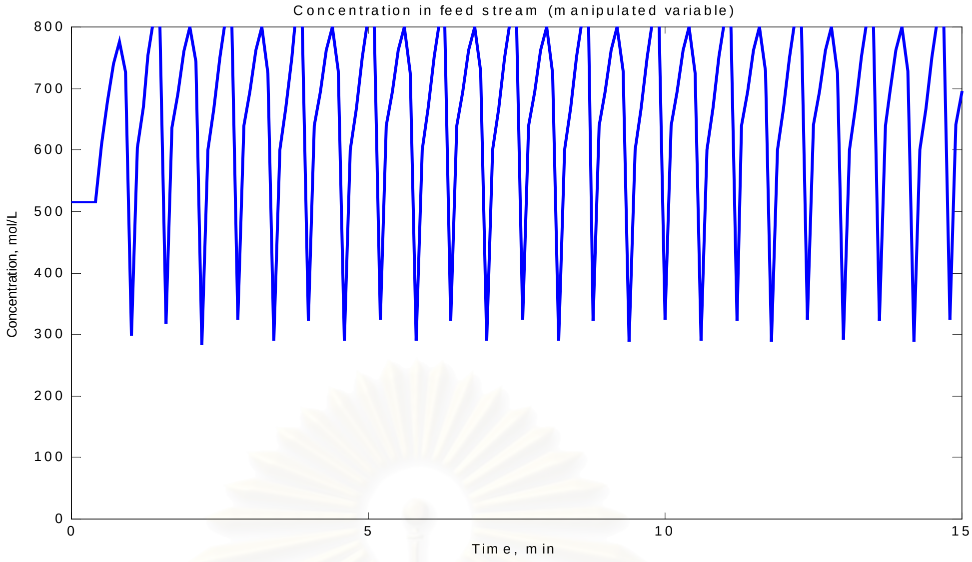
สามารถคำนวณค่าดัชนีอินทีกรัลของค่าผิดพลาดสัมบูรณ์ (IAE) ของกราฟในรูปที่ 6.45 ได้มีค่าเท่ากับ 3.0922

6.5.4 การตอบสนองเมื่อมีตัวแปรรบกวนคือ อัตราการไหลของสารละลายกรด (F_1) เพิ่มขึ้น 30%

(ก) การตอบสนองภายใต้ตัวควบคุมพีไอดีธรรมดา



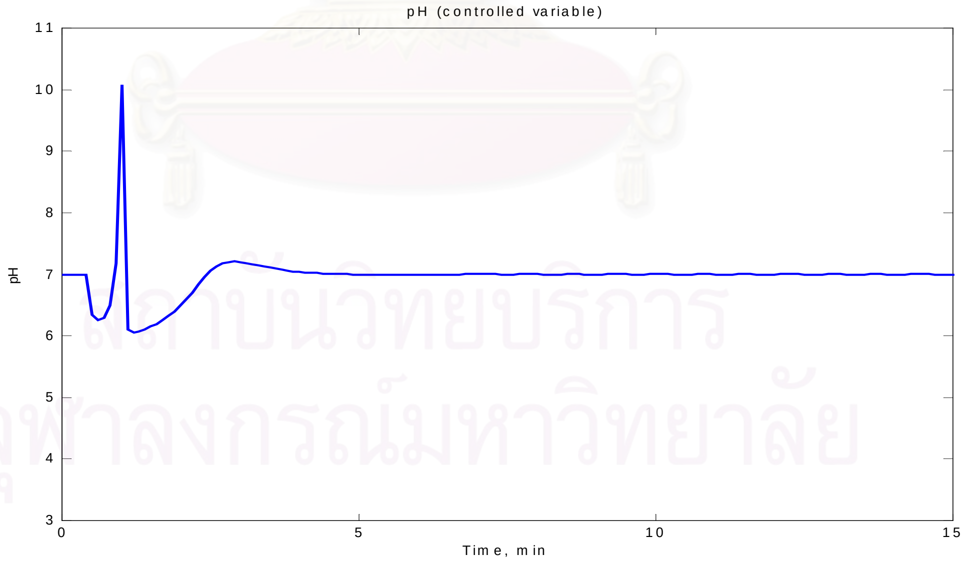
รูปที่ 6.47 การตอบสนองของค่าพีเอช (ตัวแปรควบคุม) ที่ออกจากเครื่องปฏิกรณ์ ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่ออัตราการไหลของกรด (F1) เพิ่มขึ้น 30%



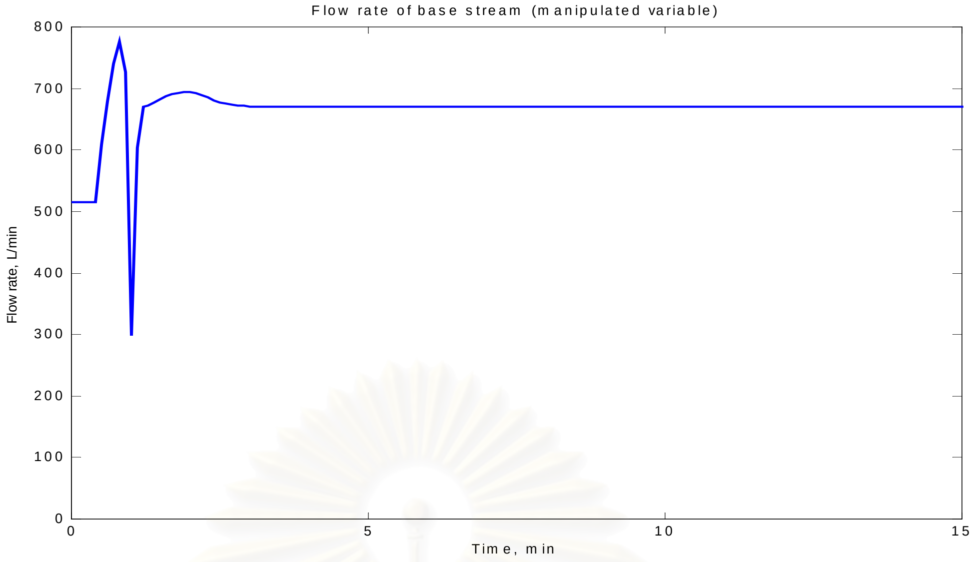
รูปที่ 6.48 อัตราการป้อนของสายด่าง (ตัวแปรปรับ) เพื่อควบคุมค่าพีเอช ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่ออัตราการไหลของสายกรด (F1) เพิ่มขึ้น 30%

สามารถคำนวณค่าดัชนีอินทีกรัลของค่าผิดพลาดสัมบูรณ์ (IAE) ของกราฟในรูปที่ 6.47 ได้มีค่าเท่ากับ 16.7942

(ข) การตอบสนองภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้



รูปที่ 6.49 การตอบสนองของค่าพีเอช (ตัวแปรควบคุม) ที่ออกจากเครื่องปฏิกรณ์ ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่ออัตราการไหลของกรด (F1) เพิ่มขึ้น 30%

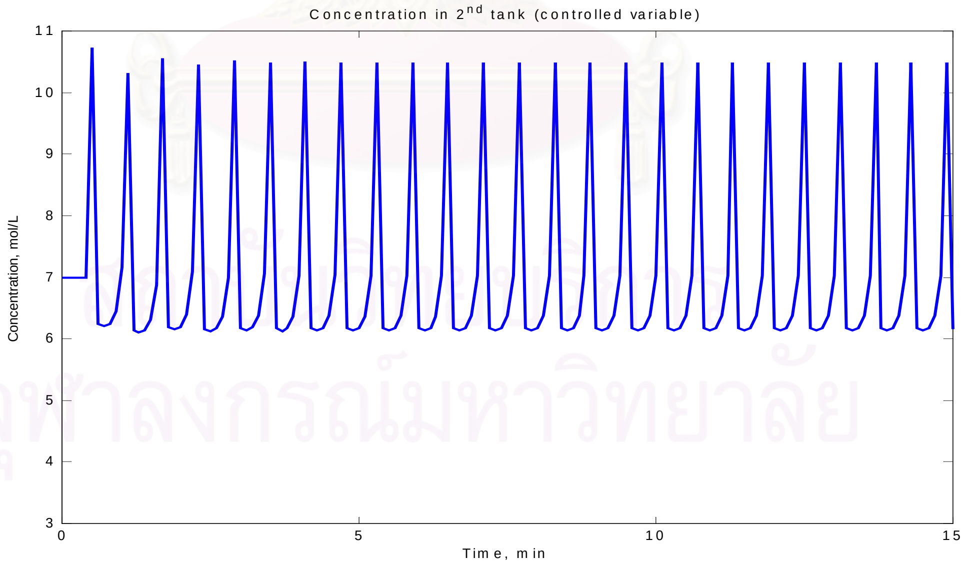


รูปที่ 6.50 อัตราการป้อนของสายด่าง (ตัวแปรปรับ) เพื่อควบคุมค่าพีเอช ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่ออัตราการไหลของสายกรด (F1) เพิ่มขึ้น 30%

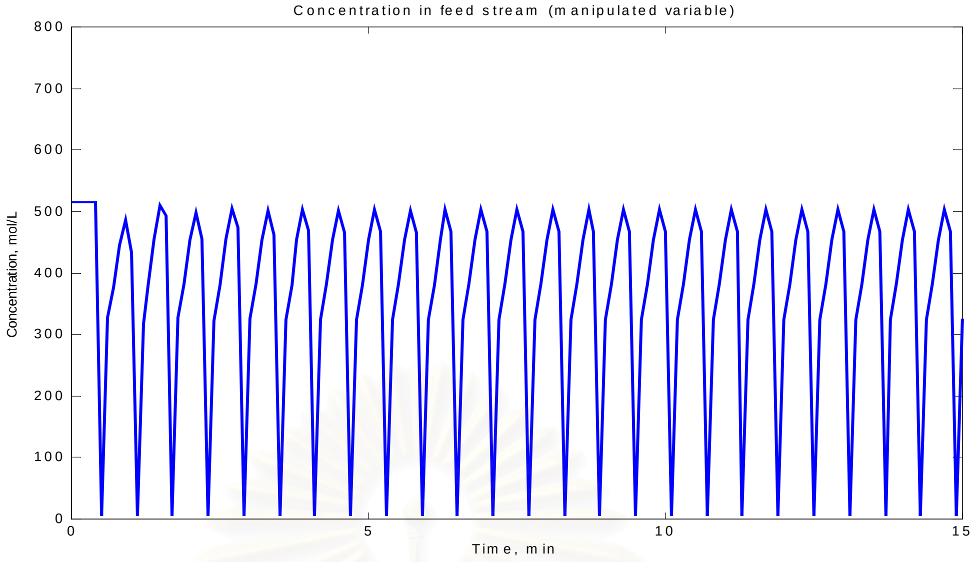
สามารถคำนวณค่าดัชนีอินทีกรัลของค่าผิดพลาดสัมบูรณ์ (IAE) ของกราฟในรูปที่ 6.49 ได้มีค่าเท่ากับ 1.6844

6.5.5 การตอบสนองเมื่อมีตัวแปรรบกวนคือ อัตราการไหลของสารละลายกรด (F_1) ลดลง 30%

(ก) การตอบสนองภายใต้ตัวควบคุมพีไอดีธรรมดา



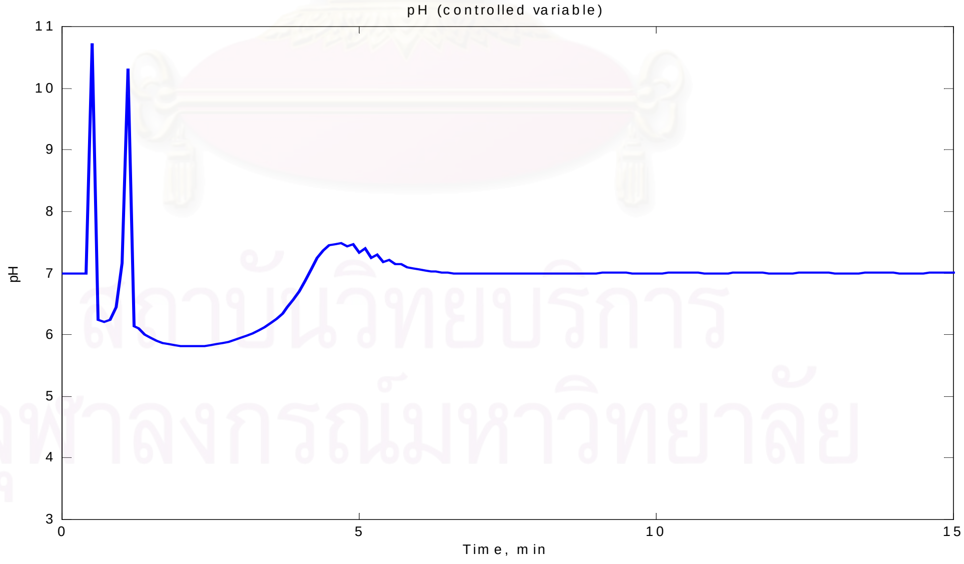
รูปที่ 6.51 การตอบสนองของค่าพีเอช (ตัวแปรควบคุม) ที่ออกจากเครื่องปฏิกรณ์ ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่ออัตราการไหลของกรด (F1) ลดลง 30%



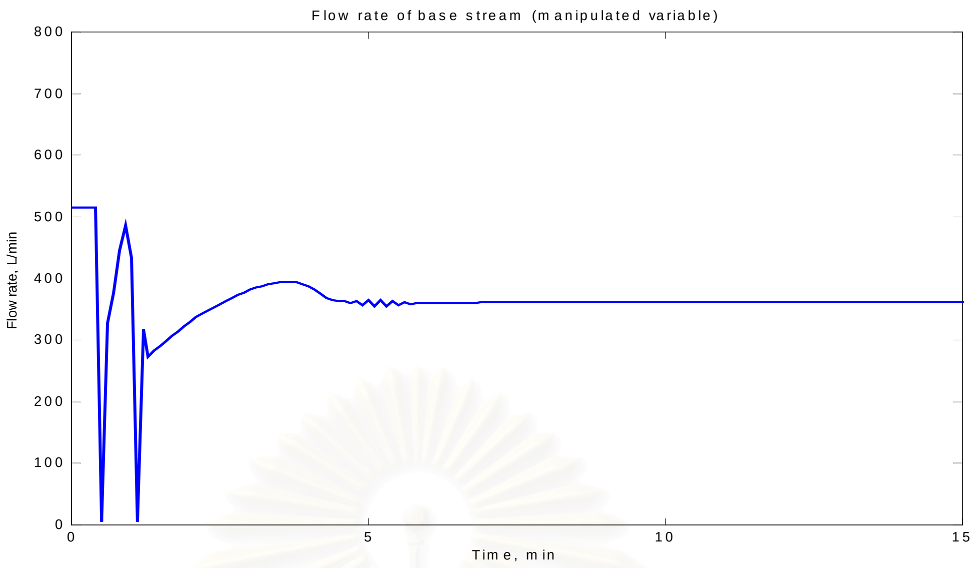
รูปที่ 6.52 อัตราการป้อนของสายด่าง (ตัวแปรปรับ) เพื่อควบคุมค่าพีเอช ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่ออัตราการไหลของสายกรด (F1) ลดลง 30%

สามารถคำนวณค่าดัชนีอินทีกรัลของค่าผิดพลาดสัมบูรณ์ (IAE) ของกราฟในรูปที่ 6.51 ได้มีค่าเท่ากับ 16.4756

(ข) การตอบสนองภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้



รูปที่ 6.53 การตอบสนองของค่าพีเอช (ตัวแปรควบคุม) ที่ออกจากเครื่องปฏิกรณ์ ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่ออัตราการไหลของกรด (F1) ลดลง 30%

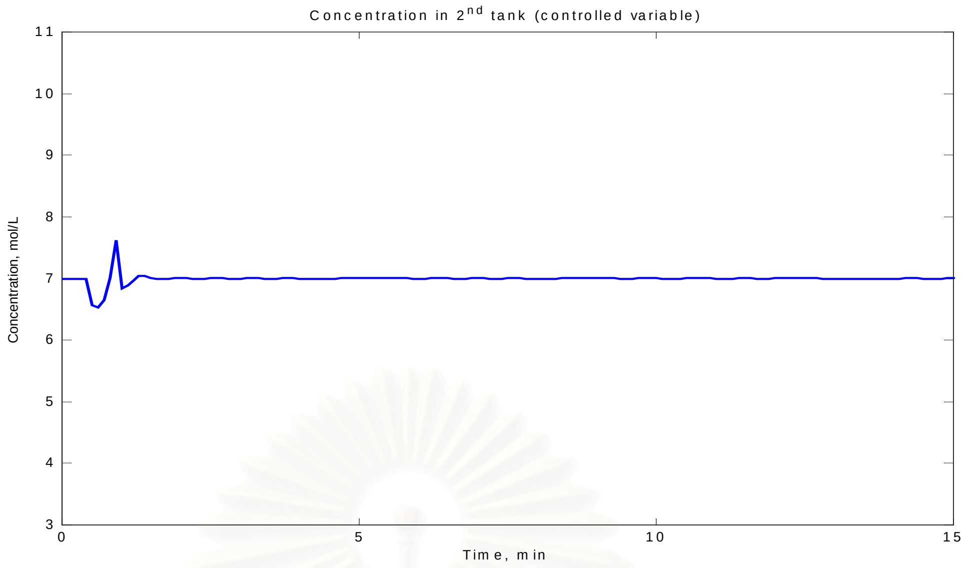


รูปที่ 6.54 อัตราการป้อนของสายด่าง (ตัวแปรปรับ) เพื่อควบคุมค่าพีเอช ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่ออัตราการไหลของสายกรด (F1) ลดลง 30%

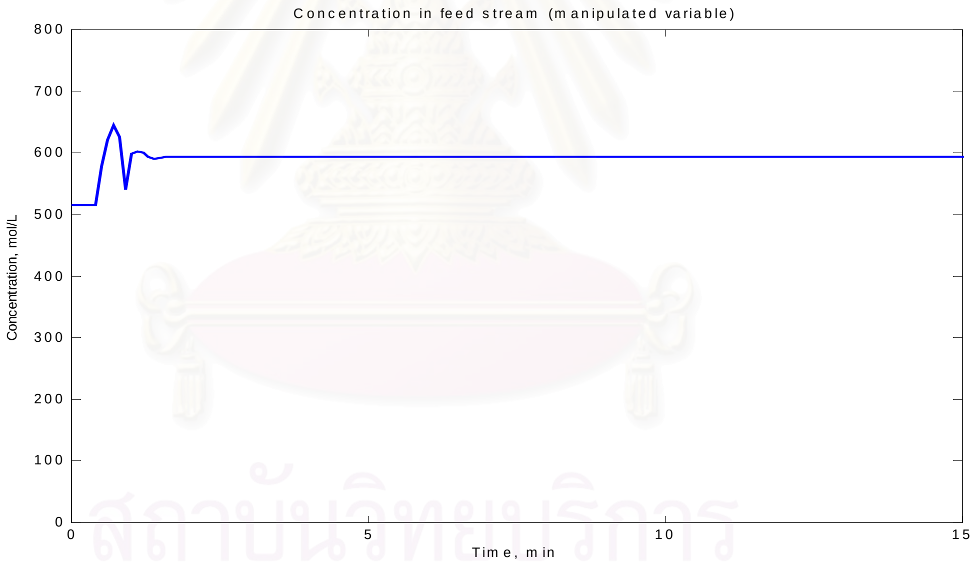
สามารถคำนวณค่าดัชนีอินทีกรัลของค่าผิดพลาดสัมบูรณ์ (IAE) ของกราฟในรูปที่ 6.53 ได้มีค่าเท่ากับ 4.4002

6.5.6 การตอบสนองเมื่อมีตัวแปรรบกวนคือ ความเข้มข้นของสารละลายกรด (C_1) เพิ่มขึ้น 15%

(ก) การตอบสนองภายใต้ตัวควบคุมพีไอดีธรรมดา



รูปที่ 6.55 การตอบสนองของค่าพีเอช (ตัวแปรควบคุม) ที่ออกจากเครื่องปฏิกรณ์ ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่อความเข้มข้นของสารละลายกรด (C1) เพิ่มขึ้น 15%

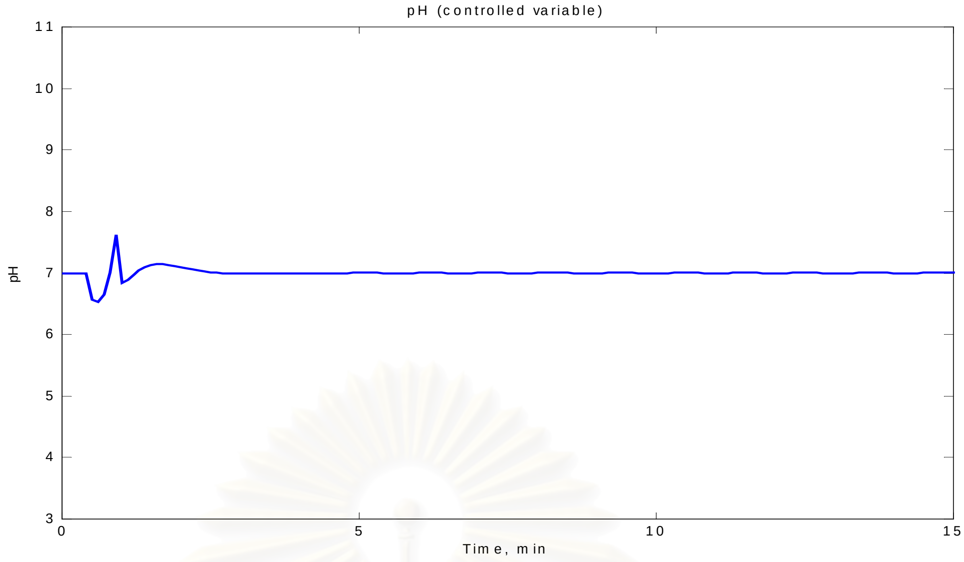


สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

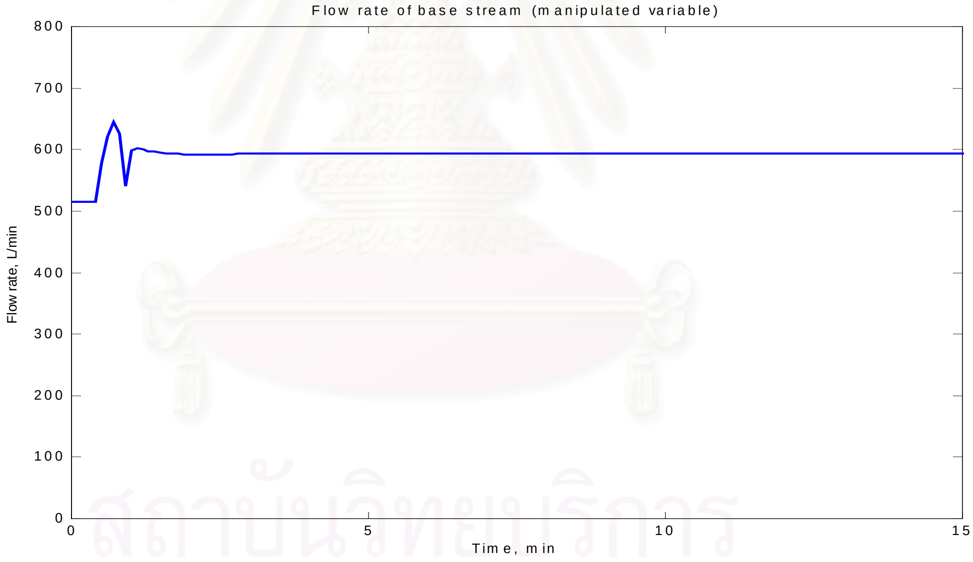
รูปที่ 6.56 อัตราการป้อนของสายด่าง (ตัวแปรปรับ) เพื่อควบคุมค่าพีเอช ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่อความเข้มข้นของสารละลายกรด (C1) เพิ่มขึ้น 15%

สามารถคำนวณค่าดัชนีอินทีกรัลของค่าผิดพลาดสัมบูรณ์ (IAE) ของกราฟในรูปที่ 6.55 ได้มีค่าเท่ากับ 0.2388

(ข) การตอบสนองภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้



รูปที่ 6.57 การตอบสนองของค่าพีเอช (ตัวแปรควบคุม) ที่ออกจากเครื่องปฏิกรณ์ ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่อความเข้มข้นของสารละลายกรด (C1) เพิ่มขึ้น 15%

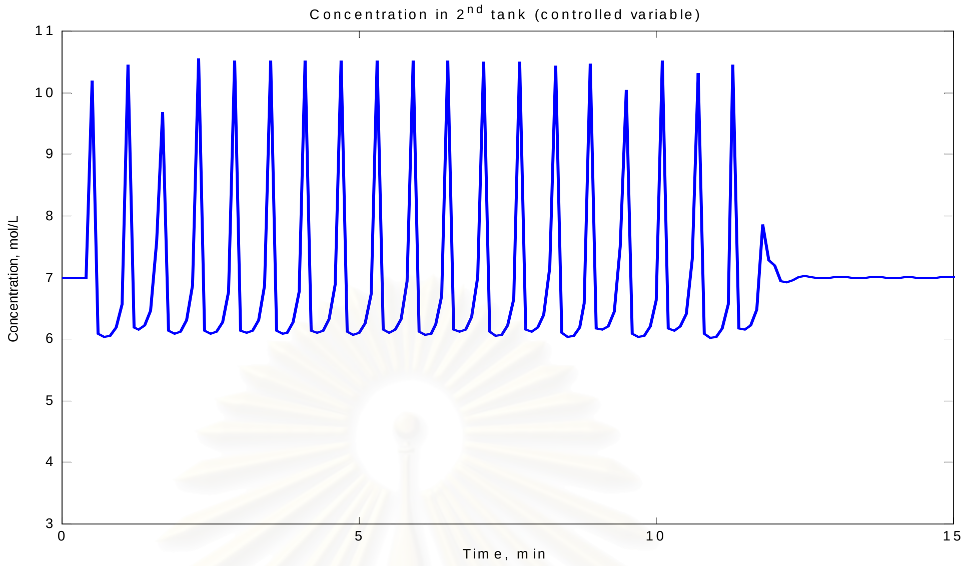


รูปที่ 6.58 อัตราการป้อนของสายด่าง (ตัวแปรปรับ) เพื่อควบคุมค่าพีเอช ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่อความเข้มข้นของสารละลายกรด (C1) เพิ่มขึ้น 15%

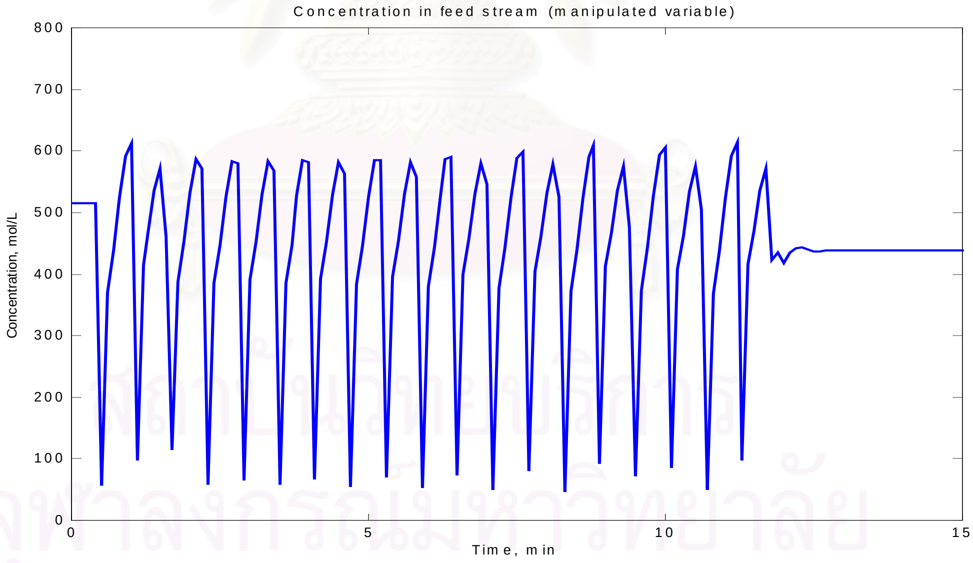
สามารถคำนวณค่าดัชนีอินทีกรัลของค่าผิดพลาดสัมบูรณ์ (IAE) ของกราฟในรูปที่ 6.57 ได้มีค่าเท่ากับ 0.3520

6.5.7 การตอบสนองเมื่อมีตัวแปรรบกวนคือ ความเข้มข้นของสารละลายกรด (C_1) ลดลง 15%

(ก) การตอบสนองภายใต้ตัวควบคุมพีไอดีธรรมดา



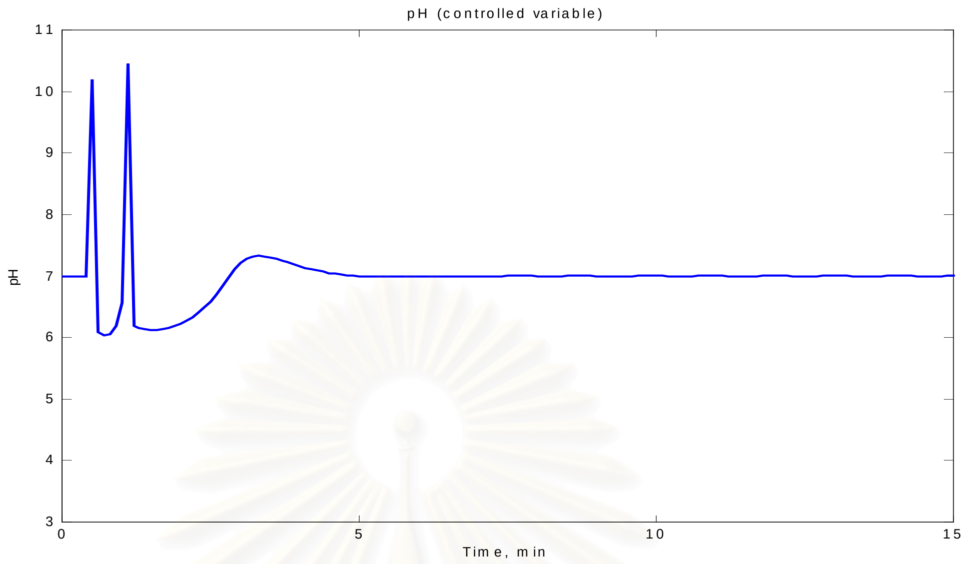
รูปที่ 6.59 การตอบสนองของค่าพีเอช (ตัวแปรควบคุม) ที่ออกจากเครื่องปฏิกรณ์ ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่อความเข้มข้นของสารละลายกรด (C1) ลดลง 15%



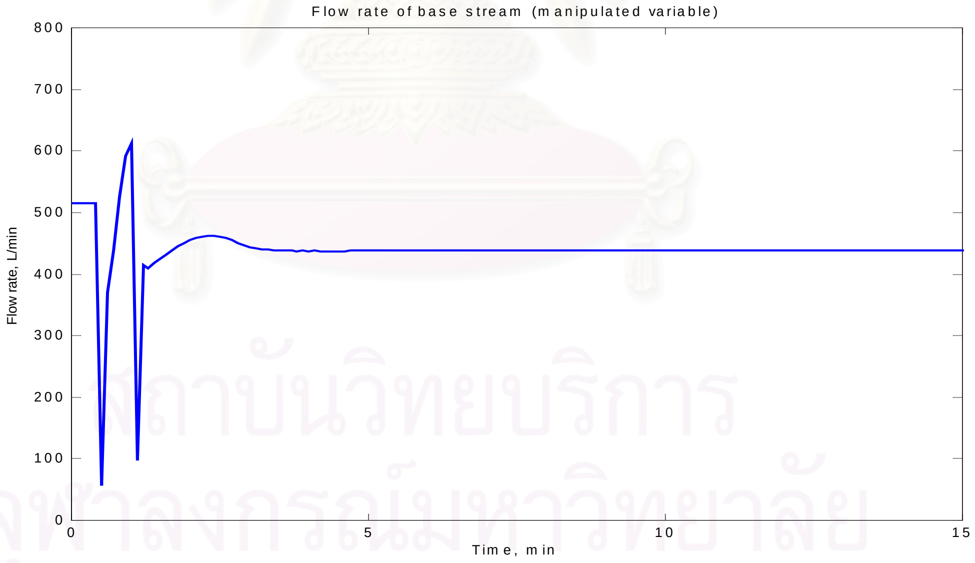
รูปที่ 6.60 อัตราการป้อนของสายด่าง (ตัวแปรปรับ) เพื่อควบคุมค่าพีเอช ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่อความเข้มข้นของสารละลายกรด (C1) ลดลง 15%

สามารถคำนวณค่าดัชนีอินทีกรัลของค่าผิดพลาดสัมบูรณ์ (IAE) ของกราฟในรูปที่ 6.59 ได้มีค่าเท่ากับ 13.4707

(ข) การตอบสนองภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้



รูปที่ 6.61 การตอบสนองของค่าพีเอช (ตัวแปรควบคุม) ที่ออกจากเครื่องปฏิกรณ์ ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่อความเข้มข้นของสารละลายกรด (C1) ลดลง 15%

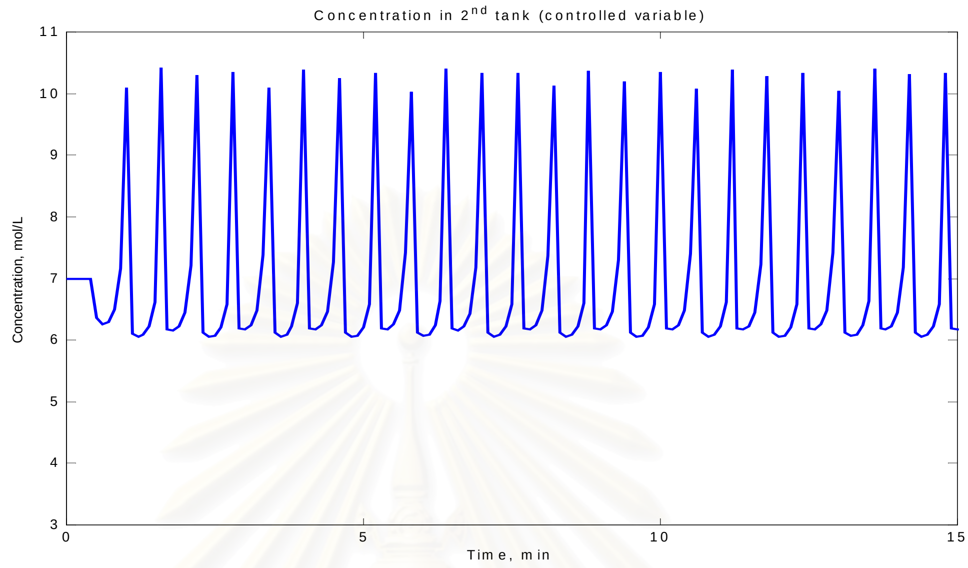


รูปที่ 6.62 อัตราการป้อนของสายด่าง (ตัวแปรปรับ) เพื่อควบคุมค่าพีเอช ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่อความเข้มข้นของสารละลายกรด (C1) ลดลง 15%

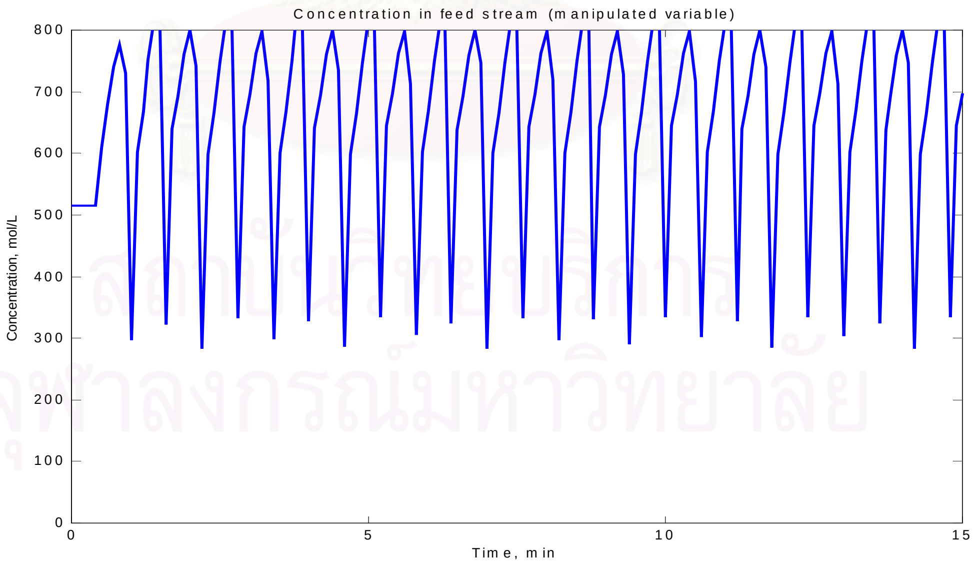
สามารถคำนวณค่าดัชนีอินทิกรัลของค่าผิดพลาดสัมบูรณ์ (IAE) ของกราฟในรูปที่ 6.61 ได้มีค่าเท่ากับ 2.5398

6.5.8 การตอบสนองเมื่อมีตัวแปรรบกวนคือ ความเข้มข้นของสารละลายกรด (C_1) เพิ่มขึ้น 30%

(ก) การตอบสนองภายใต้ตัวควบคุมพีไอดีธรรมดา



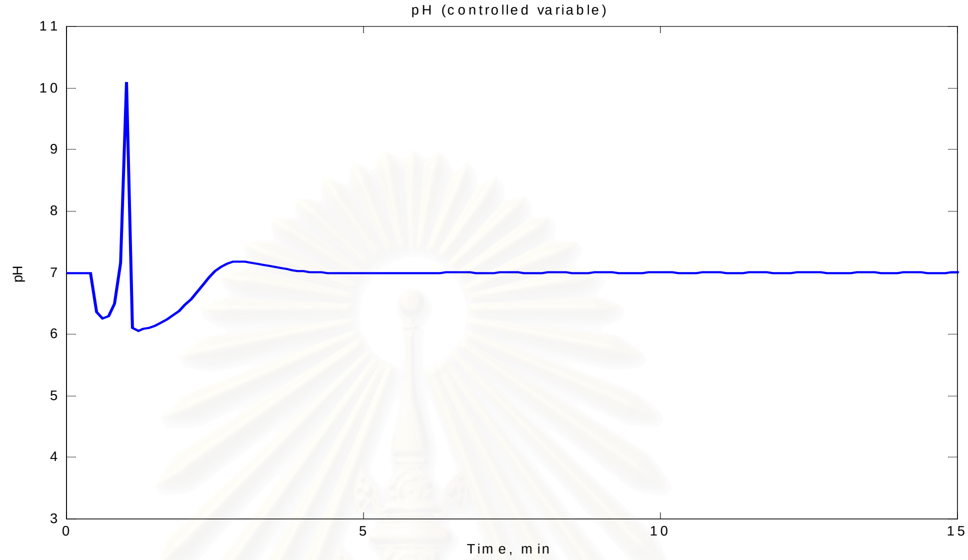
รูปที่ 6.63 การตอบสนองของค่าพีเอช (ตัวแปรควบคุม) ที่ออกจากเครื่องปฏิกรณ์ ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่อความเข้มข้นของสารละลายกรด (C1) เพิ่มขึ้น 30%



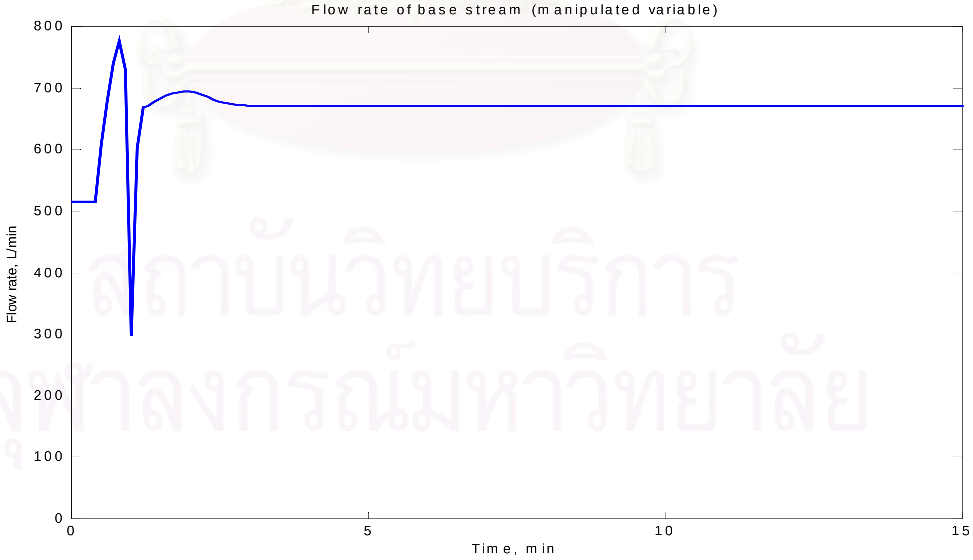
รูปที่ 6.64 อัตราการป้อนของสายด่าง (ตัวแปรปรับ) เพื่อควบคุมค่าพีเอช ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่อความเข้มข้นของสารละลายกรด (C1) เพิ่มขึ้น 30%

สามารถคำนวณค่าดัชนีอินทีกรัลของค่าผิดพลาดสัมบูรณ์ (IAE) ของกราฟในรูปที่ 6.63 ได้มีค่าเท่ากับ 16.5745

(ข) การตอบสนองภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้



รูปที่ 6.65 การตอบสนองของค่าพีเอช (ตัวแปรควบคุม) ที่ออกจากเครื่องปฏิกรณ์ ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่อความเข้มข้นของสารละลายกรด (C1) เพิ่มขึ้น 30%

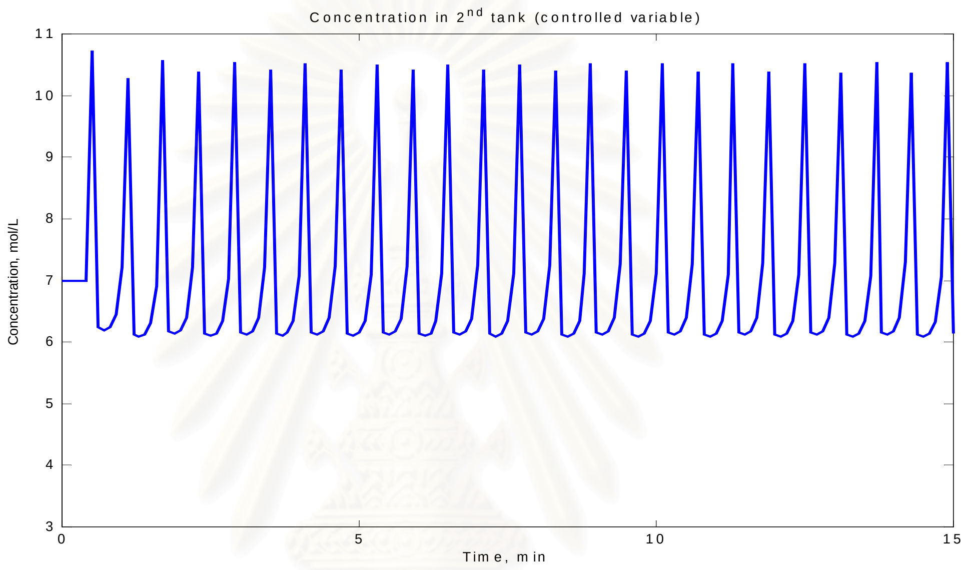


รูปที่ 6.66 อัตราการป้อนของสายด่าง (ตัวแปรปรับ) เพื่อควบคุมค่าพีเอช ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่อความเข้มข้นของสารละลายกรด (C1) เพิ่มขึ้น 30%

สามารถคำนวณค่าดัชนีอินทีกรัลของค่าผิดพลาดสัมบูรณ์ (IAE) ของกราฟในรูปที่ 6.65 ได้มีค่าเท่ากับ 1.6574

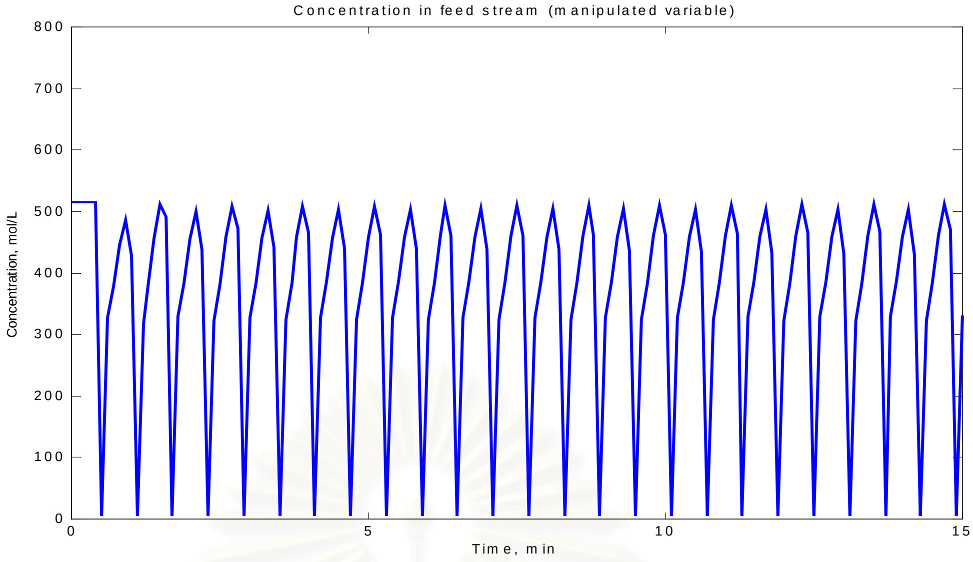
6.5.9 การตอบสนองเมื่อมีตัวแปรรบกวนคือ ความเข้มข้นของสารละลายกรด (C_1) ลดลง 30%

(ก) การตอบสนองภายใต้ตัวควบคุมพีไอดีธรรมดา



รูปที่ 6.67 การตอบสนองของค่าพีเอช (ตัวแปรควบคุม) ที่ออกจากเครื่องปฏิกรณ์ ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่อความเข้มข้นของสารละลายกรด (C1) ลดลง 30%

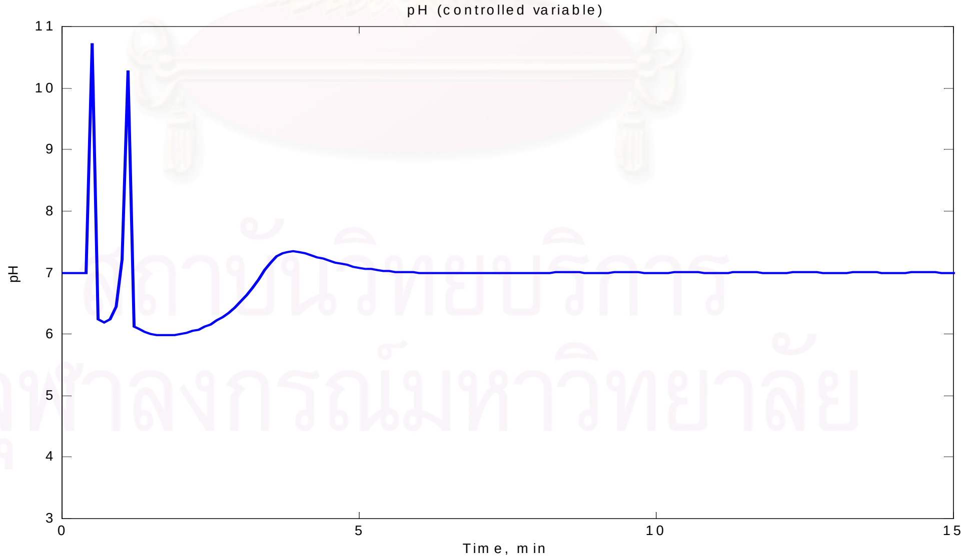
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 6.68 อัตราการป้อนของสายด่าง (ตัวแปรปรับ) เพื่อควบคุมค่าพีเอช ภายใต้ตัวควบคุมพีไอดีธรรมดา เมื่อความเข้มข้นของสารละลายกรด (C1) ลดลง 30%

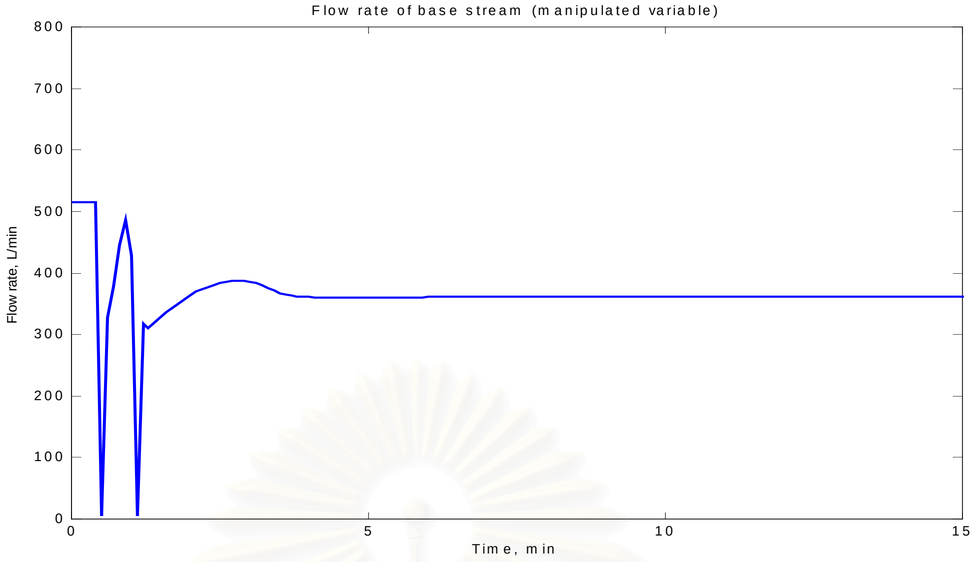
สามารถคำนวณค่าดัชนีอินทีกรัลของค่าผิดพลาดสัมบูรณ์ (IAE) ของกราฟในรูปที่ 6.67 ได้มีค่าเท่ากับ 16.8722

(ข) การตอบสนองภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้



รูปที่ 6.69 การตอบสนองของค่าพีเอช (ตัวแปรควบคุม) ที่ออกจากเครื่องปฏิกรณ์ ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่อความเข้มข้นของสารละลายกรด (C1) ลดลง 30%

สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 6.70 อัตราการป้อนของสายด่าง (ตัวแปรปรับ) เพื่อควบคุมค่าพีเอช ภายใต้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ เมื่อความเข้มข้นของสารละลายกรด (C1) ลดลง 30%

สามารถคำนวณค่าดัชนีอินทีกรัลของค่าผิดพลาดสัมบูรณ์ (IAE) ของกราฟในรูปที่ 6.69 ได้มีค่าเท่ากับ 3.1441

6.5.10 สรุปผลการทดลองตอนที่ 6.5

จากการจำลองสถานการณ์ในกรณีต่างที่ได้กล่าวมาในข้างต้น จะสามารถสรุปค่าดัชนีที่ใช้ในการหาสมรรถนะของตัวควบคุมเพื่อเปรียบเทียบได้ดังตารางที่ 6.8 และ 6.9

ตารางที่ 6.8 สรุปค่าดัชนี IAE ของตัวควบคุมบนกระบวนการที่มีความไม่เป็นเชิงเส้นสูง เมื่อมีการเปลี่ยนแปลงของตัวแปรรบกวน (F_1)

กรณี	พีไอดีธรรมดา	พีไอดีชนิดปรับค่าเองได้	เปอร์เซ็นต์แตกต่าง
ตัวแปรรบกวนเพิ่มขึ้น 30%	16.7942	1.6844	89.97
ตัวแปรรบกวนเพิ่มขึ้น 15%	0.2403	0.3891	-61.92
ตัวแปรรบกวนลดลง 15%	9.9077	3.0922	68.79
ตัวแปรรบกวนลดลง 30%	16.4756	4.4002	73.29

ตารางที่ 6.9 สรุปค่าดัชนี IAE ของตัวควบคุมบนกระบวนการที่มีความไม่เป็นเชิงเส้นสูง เมื่อมีการเปลี่ยนแปลงของตัวแปรรบกวน (C_1)

กรณี	พีไอดีธรรมดา	พีไอดีชนิดปรับค่าเองได้	เปอร์เซ็นต์แตกต่าง
ตัวแปรรบกวนเพิ่มขึ้น 30%	16.5745	1.6574	90.00
ตัวแปรรบกวนเพิ่มขึ้น 15%	0.2388	0.3520	-47.40
ตัวแปรรบกวนลดลง 15%	13.4707	2.5398	81.15
ตัวแปรรบกวนลดลง 30%	16.8722	3.1441	87.29

การจำลองการควบคุมกระบวนการ ควบคุมค่าพีเอชให้อยู่ในสภาวะเป็นกลาง เมื่อมีตัวแปรรบกวนคือ อัตราการไหลของสายกรดขาเข้า (F_1) และความเข้มข้นของสารละลายกรด ณ สายป้อนสารขาเข้า (C_1) จะเห็นได้ว่า ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ ที่มีการประยุกต์ใช้เจเนติกอัลกอริธึมดังที่นำเสนอนั้น สามารถควบคุมกระบวนการ ได้ดีกว่าผลที่ตัวควบคุมพีไอดีธรรมดาสามารถทำได้ โดยเห็นได้จากการที่มีค่าดัชนี IAE ต่ำกว่าตัวควบคุมพีไอดีธรรมดา ในเกือบทุกกรณี และมีผลการตอบสนองที่ราบเรียบมากกว่าด้วย เนื่องจากความสามารถในการปรับค่าพารามิเตอร์ได้อย่างต่อเนื่อง ทำให้สามารถรองรับความไม่เป็นเชิงเส้นของกระบวนการได้ เสมือนกับเป็นตัวควบคุมที่ไม่เป็นเชิงเส้น

ในกรณีที่มีการเปลี่ยนแปลงของตัวแปรรบกวน $\pm 30\%$ จะเห็นได้จากรูปของผลการตอบสนองว่า ตัวควบคุมพีไอดีธรรมดาให้การควบคุมที่ค่อนข้างจะรุนแรง ทำให้เกิดการแกว่งอย่างต่อเนื่องของค่าพีเอช ซึ่งเมื่อพิจารณาร่วมกับเส้นโค้งของการไตเตรท (รูปที่ 5.2) จะพบว่า การแกว่งจะให้ค่าพีเอชอยู่ในช่วงที่มีอัตราการเปลี่ยนแปลงสูงๆ ซึ่งในกรณีนี้สามารถแก้ไขได้ด้วยการลดค่าเกนของตัวควบคุมลงให้มีความแรงของการควบคุมลดลง จะทำให้สามารถควบคุมกระบวนการได้นิ่ง ณ ค่าเป้าหมาย ทั้งนี้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ที่นำเสนอ สามารถปรับค่าพารามิเตอร์ของตัวควบคุม ได้อย่างเหมาะสม ทำให้ผลการตอบสนองมีความราบเรียบไม่เกิดการสั่น

ทั้งนี้ จากการพิจารณาข้อมูลในตารางที่ 6.7 จะเห็นได้ว่ากระบวนการมีแบบจำลองของกระบวนการที่หลากหลาย ทั้งนี้ขึ้นอยู่กับปริมาณของการเปลี่ยนแปลงของอัตราการป้อนสายด่าง (ตัวแปรปรับ) ดังนั้นหากต้องการควบคุมกระบวนการให้ได้ครอบคลุมตลอดช่วงของการทำงาน จะต้องมีการเปลี่ยนแปลงค่าพารามิเตอร์ที่ใช้ในตัวควบคุมให้เหมาะสม ซึ่งในกรณีนี้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ ที่มีการประยุกต์ใช้เจเนติกอัลกอริธึม ในการหาค่าพารามิเตอร์ที่ดีที่สุด สามารถ

เปลี่ยนแปลงค่าพารามิเตอร์ของการควบคุมได้อย่างเหมาะสม ทำให้สามารถใช้ตัวควบคุมเชิงเส้นเข้าควบคุมกระบวนการที่มีความไม่เป็นเชิงเส้นสูงได้อย่างดี

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 7

สรุปผลการจำลองการควบคุม

บทนี้เป็นการสรุปผลที่ได้จากการจำลองการควบคุมกระบวนการต่างๆ ในบทที่ 6 อีกครั้ง พร้อมทั้งเสนอแนะแนวทางการปรับปรุงตัวควบคุมพีไอดีชนิดปรับค่าเองได้ ให้สามารถที่จะควบคุมกระบวนการได้ดีขึ้นในอนาคต

7.1 การควบคุมในกระบวนการที่มีการตอบสนองอันดับสอง (ตอนที่ 6.3 และ 6.4)

การจำลองการควบคุมกระบวนการตอนที่ 6.1 เป็นการเปรียบเทียบให้เห็นว่า เมื่อกระบวนการเปลี่ยนไป การปรับค่าพารามิเตอร์ของตัวควบคุมให้เหมาะสมกับการเปลี่ยนแปลงนั้น จะทำให้ได้ผลการควบคุมที่ดีกว่าการใช้ค่าพารามิเตอร์ของตัวควบคุมที่มีค่าคงที่ตลอดเวลา ซึ่งทำให้สามารถขยายผลออกไปถึงการควบคุมกระบวนการที่มีความไม่เป็นเชิงเส้น ที่มีลักษณะพลวัติของกระบวนการตลอดเวลา หากต้องการควบคุมให้ได้ผลดี ก็จำเป็นที่จะต้องมีการปรับเปลี่ยนค่าพารามิเตอร์ของตัวควบคุมตลอดเวลาด้วยเช่นกัน ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ จึงมีบทบาทสำคัญในการควบคุมกระบวนการประเภทนี้

นอกจากนั้น ผลการทดลองยังสามารถชี้ให้เห็นได้ว่า ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ที่นำเสนอนี้ ยังให้การควบคุมที่ดีกว่า ตัวควบคุมพีไอดีที่มีการเปลี่ยนค่าพารามิเตอร์ตามวิธีการคำนวณของ Ziegler-Nichols อีกด้วย เนื่องจากเจเนติกอัลกอริธึม สามารถหาค่าพารามิเตอร์ของตัวควบคุมพีไอดี ที่เหมาะสมมากกว่าค่าที่ได้จากการคำนวณตามความสัมพันธ์ของ Ziegler-Nichols

การจำลองการควบคุมในตอนที่ 6.4 พบว่า การควบคุมของตัวควบคุมพีไอดีธรรมดา และตัวควบคุมพีไอดีชนิดปรับค่าเองได้ ให้ผลการตอบสนองที่แตกต่างกันไม่มากนัก ซึ่งเป็นการแสดงให้เห็นว่า ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ที่นำเสนอนี้ สามารถควบคุมกระบวนการได้หลากหลาย ไม่เฉพาะแต่กระบวนการที่ต้องการศึกษาเท่านั้น ในส่วนของผลที่ออกมาไม่ต่างกับตัวควบคุมพีไอดีธรรมดานั้น เป็นผลมาจาก กระบวนการอันดับสองที่มีเวลาหน่วง เป็นกระบวนการเชิงเส้นที่มีแบบจำลองคงที่ตลอดเวลา จึงทำให้ค่าพารามิเตอร์ของตัวควบคุมที่คำนวณได้จากการอ้างอิงแบบจำลองนั้น มีค่าที่คงที่ตามไปด้วย ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ จึงมีสมรรถนะเหมือนกับตัวควบคุมพีไอดีธรรมดา

ถึงแม้จากการทดสอบตัวควบคุมพีไอดีที่นำเสนอ จะให้ผลการควบคุมเช่นเดียวกับตัวควบคุมพีไอดี แต่เมื่อพิจารณาถึงกระบวนการที่เกิดขึ้นจริงในอุตสาหกรรม ตัวควบคุมพีไอดีที่นำเสนอสามารถให้ประโยชน์กับกระบวนการเชิงเส้นได้คือ โดยปกติกระบวนการจะมีการเปลี่ยนแปลงที่เกิดขึ้นอย่างช้าๆ เนื่องจากหลายสาเหตุด้วยกัน เช่น เกิดตะกรันในท่อ หรือ เกิดการตกตะกอนในถัง ทำให้กระบวนการมีค่าพารามิเตอร์เปลี่ยนแปลงไปจากที่ออกแบบไว้ในตอนต้น การปรับแต่งตัวควบคุมพีไอดีจะกระทำโดยพนักงานผู้ปฏิบัติงาน แต่ด้วยการใช้ตัวควบคุมพีไอดีชนิดปรับค่าเองได้จะทำให้ การปรับแต่งค่าเกิดขึ้นอย่างอัตโนมัติ ซึ่งลดความผิดพลาดจากการไม่ได้ปรับแต่งตัวควบคุมในระยะยาวลงได้

นอกจากนั้น ตัวควบคุมพีไอดีชนิดปรับค่าเองได้ที่นำเสนอนี้ ช่วยให้สามารถใช้ค่าพารามิเตอร์ของตัวควบคุม ที่คำนวณมากจากความสัมพันธ์ต่างๆ ที่ได้มีการพัฒนามาแล้ว ได้โดยตรง เช่น ความสัมพันธ์ของ Ziegler-Nichols เป็นต้น ไม่ต้องมีการปรับละเอียด (fine tuning) อีก เนื่องจากอัลกอริธึมของตัวควบคุมจะทำการปรับแต่งให้เอง

7.2 การควบคุมในกระบวนการที่มีความไม่เป็นเชิงเส้น

ผลการควบคุมของตัวควบคุมพีไอดีชนิดปรับค่าได้ในตอนที่ 6.5 บนกระบวนการปรับค่าพีเอชให้เป็นกลาง ได้ผลที่มีการตอบสนองที่ดีกว่าตัวควบคุมพีไอดีธรรมดาในการเปลี่ยนแปลงของตัวแปรรบกวนเกือบทุกกรณี เนื่องมากจากความสามารถในการปรับค่าพารามิเตอร์ได้เองของตัวควบคุม ซึ่งปรับค่าพารามิเตอร์ให้เหมาะสมกับความไม่เป็นเชิงเส้นของกระบวนการได้

โดยปรกติกระบวนการที่มีความไม่เป็นเชิงเส้น จะมีค่าเกนของกระบวนการ (Process Gain) ไม่คงที่ ดังนั้น การที่ใช้ตัวควบคุมพีไอดีธรรมดาเข้าควบคุมกระบวนการ จะทำให้ควบคุมได้ดีเฉพาะในช่วงที่การควบคุมเกิดขึ้นใกล้ๆ กับจุดของการดำเนินงาน (Operating Point) ที่ได้มีการหาค่าพารามิเตอร์ไว้เท่านั้น ทั้งนี้เนื่องจากตัวควบคุมพีไอดีธรรมดาเป็นตัวควบคุมเชิงเส้น

การที่ทำให้ตัวควบคุมพีไอดีสามารถปรับค่าพารามิเตอร์เองได้นั้น ไม่ได้ทำให้ตัวควบคุมเปลี่ยนไปเป็นตัวควบคุมแบบไม่เชิงเส้น แต่เป็นการทำให้มีการคำนวณและใช้ค่าพารามิเตอร์ได้เหมาะสมกับจุดของการดำเนินการที่เปลี่ยนไปมากขึ้น แต่ลักษณะที่สำคัญที่สุดที่ยังทำให้ตัวควบคุมพีไอดี ได้มีการใช้กันแพร่หลายในอุตสาหกรรมจนกระทั่งปัจจุบันนั่นคือ ความทนทานของตัวควบคุมพีไอดีที่มีสูงกว่าตัวควบคุมที่มีความไม่เป็นเชิงเส้น เพราะความทนทานของตัวควบคุมเป็นสิ่งจำเป็นมาก เนื่องจากการออกแบบตัวควบคุมไม่สามารถจะออกแบบให้เป็นดังตัวควบคุมอุดม

คติได้ ตัวควบคุมจึงต้องสามารถควบคุมกระบวนการได้ ถึงแม้ว่าค่าพารามิเตอร์จะผิดพลาดไปจากจุดที่ออกแบบก็ตาม ซึ่งตัวควบคุมพีไอดีชนิดปรับค่าเองได้ที่นำเสนอนี้ ไม่ได้เสียคุณสมบัติความทนทานที่ดีข้อนี้ไป

เนื่องจากกรณีที่ได้ทำการทดสอบนี้ เป็นการควบคุมกระบวนการปรับค่าพีเอชให้เป็นกลาง ซึ่งมีความไม่เป็นเชิงเส้นสูง เป็นกรณีที่ใช้ในการทดสอบสมรรถนะของตัวควบคุมต่างๆ ตลอดมา และผลการควบคุมที่ได้ก็ทำให้สามารถมั่นใจได้ว่า สามารถนำตัวควบคุมนี้ ไปควบคุมกระบวนการที่มีความไม่เป็นเชิงเส้นอื่นๆ ได้แน่นอน

7.3 ข้อเสนอแนะ

7.3.1 ทดลองการควบคุมชนิดปรับตัวเองได้ ด้วยการหาค่าพารามิเตอร์จากหลายๆ ความสัมพันธ์ แล้วนำมาเปรียบเทียบกันเพื่อที่จะหาว่าค่าพารามิเตอร์จากความสัมพันธ์ใด ให้ผลการควบคุมที่มีการตอบสนองดีที่สุด แล้วจึงใช้ค่าพารามิเตอร์ค่านั้น ในการควบคุมครั้งต่อไป แทนที่จะใช้ค่าพารามิเตอร์จากความสัมพันธ์ใด ความสัมพันธ์หนึ่ง

7.3.2 ศึกษาและทดลองการคำนวณค่าพารามิเตอร์ ด้วยการคำนวณหาแบบวงปิด (Closed-loop) ซึ่งจะหาค่าเกนอัลติเมท (Ultimate Gain) และค่า คาบอัลติเมท (Ultimate Period) เพื่อที่จะใช้ในการคำนวณหาค่าพารามิเตอร์ของการควบคุม

7.3.3 ทดสอบการควบคุมบนกระบวนการที่ไม่มีสภาวะคงที่ (Unsteady-state) เช่นในการเกิดปฏิกิริยาในเครื่องปฏิกรณ์แบบกะ (Batch Reactor)

7.3.4 ศึกษาความเป็นไปได้ที่จะใช้หลักการนี้ กับตัวควบคุมเชิงเส้นอื่นๆ เช่น ตัวควบคุมป้อนกลับสเตท (State Feedback Controller) ซึ่งมีความสามารถในการควบคุมกระบวนการแบบหลายตัวแปรได้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

ภาษาไทย

ไพศาล กิตติศุภกร และ วนารัตน์ ดวงวัง การประยุกต์ใช้ตัวควบคุมแบบป้อนกลับสเตตสำหรับควบคุมค่าพีเอชในกระบวนการบำบัดน้ำเสีย. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมเคมี บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2542, 145 หน้า

ไพศาล กิตติศุภกร และ นันทนา ศิริพันธ์ Globally linearizing control for pH control of the wastewater treatment process. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมเคมี บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2543, 174 หน้า

ภาษาอังกฤษ

Al-Duwaish, H., and Wasif, N. Nonlinear model predictive control of Hammerstien and Wiener models using genetic algorithms. Proc. IEEE Con. App. 2001(Sep), Mexico

Astrom, K. J., and Wittenmark, B. Adaptive Control. 2nd ed. Addison-Wesley, 1995

Bhat, N., and McAvoy, T. J. Use of neural nets for dynamic modeling and control of chemical process systems. Comp. Chem. Eng. 1990, Vol.14, No.4, 573-583

Carlos, A. S., and Armando, B. C. Principles and Practice of Automatic Process Control. 2nd ed. Wiley, 1997

Goldberg, D. E. Genetic Algorithms in search, optimization, and machine learning. Addison-Wesley, 1989

Gustaffson, T. K., and Waller, K. V. Dynamic modeling and reaction invariant control of pH. Chem. Eng. Sci. 1983, Vol.38, No.3, 389-398

Henson, M. A., and Seborg, D. E. Adaptive nonlinear control of a pH neutralization process. IEEE Trans. Con. Sys. 1994, Vol.2, No.3, 169-182

Isermann, R., Lachmann, K. H., and Matko, D. Adaptive control systems. Prentice Hall, 1992

Karr, C. L. Genetic algorithms for modeling, design, and process control. CIKM'93 Proc., 1993(Nov), DC, USA, 233-238

Kim, J. H., and Oh, S. J. A fuzzy PID controller for nonlinear and uncertain systems. Soft Computing. 4(2000), 123-129

Lee, T. C. et al. Indirect adaptive backstepping control of pH neutralization process based on recursive prediction error method for combined state and parameter estimation. Ind. Eng. Chem. Res. 2001, Vol.40, 4102-4110

Marlin, T. E. Process control: Designing processes and control systems for dynamic performance. 2nd ed. McGraw-Hill, 2000

Mitsukura, Y., Yamamoto, T., and Kaneda, M. Genetic tuning scheme of PID parameters for first-order systems with large dead times. IEICE Trans. Fund. 2000, Vol.E83, No.4, 740-746

Norquay, S. J., Palazoglu, A., and Romagnoli, J. A. Model predictive control based on Wiener models. Chem. Eng. Sci. 1998, Vol.53, No.1, 75-84

Ogunnaike, B. A., and Ray, W. H. Process dynamics, modeling, and control. Oxford University Press, 1994

Roffel, B., Vermeer, P. J., and Chin, P. A. Simulation and implementation of self-tuning controllers. Prentice Hall, 1989

Srivastava, R. P. Use of genetic algorithms for optimization in digital control of dynamic systems. ACM 1992 Proc., 1992, NY, USA, 219-224

Suzuki, M. et al. A design of neural-net based self-tuning PID controllers. ICANN 2001, 2001(Aug), Vienna, Austria

Vijaykumar, H., Michael, N. Practical Handbook of Genetic Algorithms: New Frontiers, Vol.2, 1995, CRC press Inc

Wellstead, P. E., and Zarrop, M. B. Self-tuning systems : Control and signal processing. Wiley, 1991

Yeo, Y. K., and Kwon, T. I. A neural PID controller for the pH neutralization process. Ind. Eng. Chem. Res. 1999, Vol.38, No.3, 978-987.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก.
โปรแกรมที่ใช้ในการทำการจำลองการควบคุม

โปรแกรมที่ใช้ในการจำลองการควบคุมในงานวิจัยนี้ ใช้โปรแกรมที่เขียนขึ้นจากโปรแกรมแม็ทแล็บ เวอร์ชัน 5.3.1 โดยที่จะมีการแบ่งออกเป็น 2 ประเภทคือ โปรแกรมหลัก และ โปรแกรมฟังก์ชัน ซึ่งมีรหัส (Source Codes) ดังต่อไปนี้

1. โปรแกรมหลัก

1.1 โปรแกรมที่ใช้ในตอนที่ 6.1

```
clear all;
%-----------------------------------------------------------------
%          Define variables & matrix used in calculation
%-----------------------------------------------------------------
maxlp               = 751;
Ts                  = 0.1;
tfinal              = Ts*(maxlp-1);
t                   = linspace(0,tfinal,maxlp);
y1(maxlp)           = 0;                %y(t) from 2nd Order Process
y2(maxlp)           = 0;                %y(t) from 1st Order Process estimated from Fit 3
y3(maxlp)           = 0;                %y(t) from 1st Order Process estimated from GAs
x(maxlp)            = 0;                %x(t)
Kp(1:maxlp)         = 0.5;              %Process gain
T(1:maxlp)          = 4;                %Time constant
xi(1:maxlp)         = 1.25;             %Damping factor
dt                  = t(2) - t(1);
%-----------------------------------------------------------------
%          Define step changes in input variables x(t)
%-----------------------------------------------------------------
pt1                 = 15;
x(pt1:maxlp)        = 1;
%-----------------------------------------------------------------
%          main loop calculation
%-----------------------------------------------------------------
for i = 2 : maxlp-1
   y1(i+1) = 2*y1(i)-y1(i-1)+(dt/T(i))^2*( Kp(i)*x(i) - 2*xi(i)*T(i)*(y1(i)-y1(i-1))/dt - y1(i));
end

figure
hold on
plot(t,y1,'LineWidth',2)
axis([0 tfinal 0 0.6])
title('Response of 2^n^d order process')
xlabel('Time, min')
ylabel('y')

%-----------------------------------------------------------------
%          Estimate with Fit 3 method
%-----------------------------------------------------------------
DelX = x(i+1) - x(1);
DelY = y1(i+1) - y1(1);
 Y28 = y1(1) + DelY*0.283;
 Y63 = y1(1) + DelY*0.632;
```

```
i28 = 1;
while y1(i28) <= Y28
  i28 = i28+1;
end
i63 = 1;
while y1(i63) <= Y63
  i63 = i63+1;
end
t28 = t(i28-1) - t(pt1);
t63 = t(i63) - t(pt1);

Kp1     = DelY/DelX;
T1      = 1.5*(t63 - t28);
 Z1     = t63 - T1;

%-------------------------------------------------------------------
%       PID controller parameters calculation based on
%       open-loop Ziegler-Nichols method.
%-------------------------------------------------------------------
Ctrl(1:3) = 0;
Ctrl(1)         = 1.2/Kp1 * (T1/Z1);
Ctrl(2)         = 2*Z1;
Ctrl(3)         = 0.5*Z1;
disp(sprintf('Estimated process has Kp = %g   T  = %g   Z  = %g', Kp1, T1, Z1))
disp(sprintf('PID controller    has Kc = %g   Ti = %g   Td = %g', Ctrl))

%-------------------------------------------------------------------
%       Calculate y(t) of 1st Order from Fit 3 Estimation
%-------------------------------------------------------------------
Z1      = round(Z1/dt);
for i = 1 : maxlp-1
  if i > Z1
    y2(i+1) = y2(i) + (dt/T1)*( Kp1*x(i-Z1) - y2(i) );
  end
end
plot(t,y2,'-r','LineWidth',2)

%-------------------------------------------------------------------
%       Calculate y(t) of 1st Order from GAs Estimation
%-------------------------------------------------------------------
time2           = 15;
count           = 0;
Theta           = [0.5 8.5 1.65];               %First guess
[Theta]         = func_GASelf(x, y1, Theta);            %Find best solution (parameters)
Kp3             = Theta(1);
T3              = Theta(2);
Z3              = round(Theta(3)/dt);
for i = 1 : maxlp-1
  if i > Z3
    y3(i+1) = y3(i) + (dt/T3)*( Kp3*x(i-Z3) - y3(i) );
  end
end
plot(t,y3,'-','LineWidth',2)
```

1.2 โปรแกรมที่ใช้ในตอนที่ 6.2 – PID Controller

```
clear all;
%-------------------------------------------------------------------
%       Define variables & matrix used in calculation
```

```
%------------------------------------------------------------------
maxlp               = 501;
Ts                  = 0.1;
Tfinal              = Ts*(maxlp-1);
T                   = linspace(0,tfinal,maxlp);
y(maxlp)= 0;
x(1:maxlp)          = 0;
Kp(1:maxlp)         = 1;
T(1:maxlp)          = 2;
xi(1:maxlp)         = 1.5;
d(maxlp)            = 0;
e(maxlp)            = 0;
yset                = y(1);
%------------------------------------------------------------------
%       Define parameter & initial conditions
%------------------------------------------------------------------
dt                  = t(2) - t(1);
onCtrl              = 0.0001*yset;
%------------------------------------------------------------------
%       Define disturbance variables
%------------------------------------------------------------------
pt1                 = 5;
d(pt1:maxlp)        = 1;
T(240:maxlp)        = T(1)*0.5;
pt2                 = 251;
d(pt2:maxlp)        = 1.5;

%------------------------------------------------------------------
%       PID controller parameters
%------------------------------------------------------------------
Ctrl(1)             = 10.8011;      %Kc
Ctrl(2)             = 1.2;          %Ti
Ctrl(3)             = 0.3;          %Td

%------------------------------------------------------------------
%       main loop calculation
%------------------------------------------------------------------
e(1:2)              = yset - y(1:2);
for i = 2 : maxlp-1
  y(i+1) = 2*y(i)-y(i-1)+(dt/T(i))^2*( Kp(i)*x(i) - 2*xi(i)*T(i)*(y(i)-y(i-1))/dt - y(i) + d(i) );
  e(i+1) = yset - y(i+1);
  if i > 240
    Ctrl = [16.8001 0.6 0.15];
  end
  if abs(e(i+1)) > onCtrl
    xx = func_CtrlPID([e(i-1) e(i) e(i+1)], Ctrl, dt);
    x(i+1) = x(i) + xx;
  else
    x(i+1) = x(i);
    e(i+1) = 0;
  end
end

IAE = func_IAE(t,y,yset);
disp(sprintf('PID controller has IAE = %g', IAE))
disp('  ')

figure
hold on
plot(t,y,'-.','LineWidth',1.5)
axis([0 tfinal -0.1 0.1])
title('Response of 2^n^d order process')
xlabel('Time, min')
```

```
ylabel('y')
```

1.3 โปรแกรมที่ใช้ในตอนที่ 6.2 – Self-tuning PID Controller

```
clear all;
%-------------------------------------------------------------------
%          Define variables & matrix used in calculation
%-------------------------------------------------------------------
maxlp               = 501;
Ts                  = 0.1;
tfinal              = Ts*(maxlp-1);
t                   = linspace(0,tfinal,maxlp);
y(maxlp)= 0;
x(1:maxlp)          = 0;
Kp(1:maxlp)         = 1;
T(1:maxlp)          = 2;
xi(1:maxlp)         = 1.5;
d(maxlp)            = 0;
e(maxlp)            = 0;
yset                = y(1);
dt                  = t(2) - t(1);
onCtrl              = 0.00001;
%-------------------------------------------------------------------
%          Define disturbance variables
%-------------------------------------------------------------------
pt1                 = 5;
d(pt1:maxlp)        = 1;
T(250:maxlp)        = T(1)*1.5;
pt2                 = 251;
d(pt2:maxlp)        = 1.5;

%-------------------------------------------------------------------
%          PID controller parameters
%-------------------------------------------------------------------
Ctrl(1)             = 10.8011;      %Kc
Ctrl(2)             = 1.2;          %Ti
Ctrl(3)             = 0.3;          %Td


%-------------------------------------------------------------------
%                   main loop calculation
%-------------------------------------------------------------------
e(1:2)              = yset - y(1:2);
time2               = 15;
count               = 0;
Theta               = [1 5.4 0.6];
for i = 2 : maxlp-1
  y(i+1) = 2*y(i)-y(i-1)+(dt/T(i))^2*( Kp(i)*x(i) - 2*xi(i)*T(i)*(y(i)-y(i-1))/dt - y(i) + d(i) );
  e(i+1) = yset - y(i+1);
  if abs(e(i+1)) > onCtrl
    if count > time2
      time1 = round(Theta(3)/Ts)+3;
      [Ctrl, Theta] = func_GASelf([x(i-time1:i) y(i-time1:i+1) y(1)],Ts,Theta);
      count = 0;
    else
      count = count + 1;
    end
    xx = func_CtrlPID([e(i-1) e(i) e(i+1)], Ctrl, dt);
    x(i+1) = x(i) + xx;
  else
    x(i+1) = x(i);
```

```
    e(i+1) = 0;
  end
end

IAE = func_IAE(t,y,yset);
disp(sprintf('PID controller has IAE = %g', IAE))
disp('  ')

figure
plot(t,y,'-','LineWidth',1.5)
axis([0 tfinal -0.1 0.1])
title('Response of 2^n^d order process')
xlabel('Time, min')
ylabel('y')
```

1.4 โปรแกรมที่ใช้ในตอนที่ 6.3 – PID Controller

```
clear all;
%---------------------------------------------------------------------
%         Define variables & matrix used in calculation
%---------------------------------------------------------------------
maxlp             = 601;
Ts                = 0.4;
tfinal            = Ts*(maxlp-1);
t                 = linspace(0,tfinal,maxlp);
F(maxlp)          = 0;
Ca0(maxlp)        = 0;
Ca1(maxlp)        = 0;
Ca2(maxlp)        = 0;
e(maxlp)          = 0;

%---------------------------------------------------------------------
%         Define parameter & initial conditions
%---------------------------------------------------------------------
V                 = 1.05;                 %Tank volume (L)
F(1:maxlp)        = 0.085;                %Flow Rate (L/min)
Ca0(1:maxlp)      = 0.925;        %Concentration of feed (mol/L)
Ca1(1)            = 0.6191;               %Concentration of 1st tank outlet (mol/L)
Ca2(1)            = 0.4144;               %Concentration of 2nd tank outlet (mol/L)
Ca2set            = Ca2(1);               %Setpoint
K                 = 0.040;        %Reaction rate constant
Ca0min  = Ca0(1)/100;
Ca0max = Ca0(1)*2;
delay             = 10;
onCtrl            = 0.0001*Ca2set;

%---------------------------------------------------------------------
%         Define disturbance variables
%---------------------------------------------------------------------
pt1               = 10;
F(pt1:maxlp)      = F(1)*1.3;

%---------------------------------------------------------------------
%         PID controller parameters
%---------------------------------------------------------------------
Ctrl(1)           = 6.8886;               %Kc
Ctrl(2)           = 11.2;                 %Ti
Ctrl(3)           = 2.8;                  %Td

%---------------------------------------------------------------------
%         main loop calculation
```

```
%-------------------------------------------------------------------
e(1)                    = Ca2set - Ca2(1);
for i = 1 : maxlp-1
%-------------------------------------------------------------------
%          concentration measurement
%          estimate Ca1(i+1) and Ca2(i+1) using ode15s method in matlab toolbox
%-------------------------------------------------------------------
  if i < delay + 1
    U(i) = Ca0(1);
  else
    U(i) = Ca0(i-delay);
  end
  tspan    = [t(i) t(i+1)];
  init       = [Ca1(i) Ca2(i)];
  AddOp  = [F(i) U(i) V k];
   [tt yy]   = ode15s('func_2nd_ODE', tspan, init,[],AddOp);
  Ca1(i+1) = yy(size(yy,1),1);
  Ca2(i+1) = yy(size(yy,1),2);
%-------------------------------------------------------------------
%          Concentration controller
%-------------------------------------------------------------------
  e(i+1) = Ca2set - Ca2(i+1);
  if abs(e(i+1)) > onCtrl
    u = func_CtrlPID([e(i-1) e(i) e(i+1)], Ctrl, Ts);
    Ca0(i+1) = Ca0(i) + u;
    if Ca0(i+1) < Ca0min
      Ca0(i+1) = Ca0min;
    elseif Ca0(i+1) > Ca0max
       Ca0(i+1) = Ca0max;
    end
  else
    Ca0(i+1) = Ca0(i);
    e(i+1) = 0;
  end
end

IAE = func_IAE(t,Ca2,Ca2set);
disp(sprintf('PID controller has IAE = %g', IAE))
disp('  ')

figure
plot(t,Ca2,'-','LineWidth',2.5)
axis([0 tfinal 0.3 0.5])
title('Concentration in 2^n^d tank (controlled variable)')
xlabel('Time, min')
ylabel('Concentration, mol/L')

figure
plot(t,Ca0,'-','LineWidth',2.5)
axis([0 tfinal 0 2])
title('Concentration in feed stream (manipulated variable)')
xlabel('Time, min')
ylabel('Flow rate, L/min')
```

1.5 โปรแกรมที่ใช้ในตอนที่ 6.3 – Self-tuning PID Controller

```
clear all;
%-------------------------------------------------------------------
%          Define variables & matrix used in calculation
%-------------------------------------------------------------------
maxlp                 = 601;
```

```
Ts              = 0.4;
tfinal          = Ts*(maxlp-1);
t               = linspace(0,tfinal,maxlp);
F(maxlp)        = 0;
Ca0(maxlp)      = 0;
Ca1(maxlp)      = 0;
Ca2(maxlp)      = 0;
e(maxlp)        = 0;
%--------------------------------------------------------------------
%       Define parameter & initial conditions
%--------------------------------------------------------------------
V               = 1.05;                 %Tank volume (L)
F(1:maxlp)      = 0.085;                %Flow Rate (L/min)
Ca0(1:maxlp)    = 0.925;        %Concentration of feed (mol/L)
Ca1(1)          = 0.6191;               %Concentration of 1st tank outlet (mol/L)
Ca2(1)          = 0.4144;               %Concentration of 2nd tank outlet (mol/L)
Ca2set          = Ca2(1);               %Setpoint
K               = 0.040;        %Reaction rate constant
Ca0min  = Ca0(1)/100;
Ca0max = Ca0(1)*2;
delay           = 10;
onCtrl          = 0.0001*Ca2set;

%--------------------------------------------------------------------
%       Define disturbance variables
%--------------------------------------------------------------------
pt1             = 10;
F(pt1:maxlp)    = F(1)*1.3;

%--------------------------------------------------------------------
%       PID controller parameters
%--------------------------------------------------------------------
Ctrl(1)         = 6.8886;               %Kc
Ctrl(2)         = 11.2;                 %Ti
Ctrl(3)         = 2.8;                  %Td

%--------------------------------------------------------------------
%       main loop calculation
%--------------------------------------------------------------------
e(1)            = Ca2set - Ca2(1);
time2           = 15;
count           = 0;
Theta           = [0.44 14.4 5.6];
for i = 1 : maxlp-1
%--------------------------------------------------------------------
%       concentration measurement
%       estimate Ca1(i+1) and Ca2(i+1) using ode15s method in matlab toolbox
%--------------------------------------------------------------------
  if i < delay + 1
    U(i) = Ca0(1);
  else
    U(i) = Ca0(i-delay);
  end
  tspan      = [t(i) t(i+1)];
  init       = [Ca1(i) Ca2(i)];
  AddOp      = [F(i) U(i) V k];
  [tt yy]    = ode15s('func_2nd_ODE', tspan, init,[],AddOp);
  Ca1(i+1) = yy(size(yy,1),1);
  Ca2(i+1) = yy(size(yy,1),2);
%--------------------------------------------------------------------
%       Self-tuning PID controller
%--------------------------------------------------------------------
```

```
  e(i+1) = Ca2set - Ca2(i+1);
  if abs(e(i+1)) > onCtrl
    if count > time2
      time1 = round(Theta(3)/Ts)+3;
      [Ctrl, Theta] = func_GASelf([U(i-time1:i) Ca2(i-time1:i+1) Ca2(1)],Ts, delay,Theta);
      count = 0;
    else
      count = count + 1;
    end
    u = func_CtrlPID([e(i-1) e(i) e(i+1)], Ctrl, Ts);
    Ca0(i+1) = Ca0(i) + u;
    if Ca0(i+1) < Ca0min
      Ca0(i+1) = Ca0min;
    elseif Ca0(i+1) > Ca0max
      Ca0(i+1) = Ca0max;
    end
  else
    Ca0(i+1) = Ca0(i);
    e(i+1) = 0;
  end
end

IAE = func_IAE(t,Ca2,Ca2set);
disp(sprintf('PID controller has IAE = %g', IAE))
disp('  ')

figure
plot(t,Ca2,'-','LineWidth',2.5)
axis([0 tfinal 0.3 0.5])
title('Concentration in 2^n^d tank (controlled variable)')
xlabel('Time, min')
ylabel('Concentration, mol/L')

figure
plot(t,Ca0,'-','LineWidth',2.5)
axis([0 tfinal 0 2])
title('Concentration in feed stream (manipulated variable)')
xlabel('Time, min')
ylabel('Flow rate, mol/L')
```

1.6 โปรแกรมที่ใช้ในตอนที่ 6.4 – PID Controller

```
clear all;
%--------------------------------------------------------------------
%         Define variables & matrix used in calculation
%--------------------------------------------------------------------
maxlp = 151;
Ts = 0.1;                                    %Sampling time (min)
tfinal = Ts*(maxlp-1);
t = linspace(0,tfinal,maxlp);
F1(maxlp) = 0;
F2(maxlp) = 0;
coxi(maxlp) = 0;
xi(maxlp) = 0;
C1(maxlp) = 0;
C2(maxlp) = 0;
H(maxlp) = 0;
pH(maxlp) = 0;
e(maxlp) = 0;
pHset = 7;
```

```
%--------------------------------------------------------------------
%         Define parameter & initial conditions
%--------------------------------------------------------------------
V = 1000;                                          %Tank volume (L)
F1(1:maxlp) = 81;            %Acid Flow Rate (L/min)
F2(1:maxlp) = 515.7923541247485;                                      %Base Flow Rate (L/min)
C1(1:maxlp) = 0.32;                                      %Acid concentration in feed (mol/L)
C2(1:maxlp) = 0.05;                                      %Base concentration in feed (mol/L)
xi(1) = F1(1)*C1(1)/(F1(1)+F2(1));             %Acetate concentration in tank (mol/L)
coxi(1) = F2(1)*C2(1)/(F1(1)+F2(1));                     %Sodium concentration in tank (mol/L)
Kw = 1e-14;                                        %Water constant
Ka(1:maxlp) = 1.978e-5;                            %Acetic acid constant
F2min = F2(1)/100;
F2max = F2(1)*2;
onCtrl = 0.0001*pHset;

%--------------------------------------------------------------------
%         Define disturbance variables
%--------------------------------------------------------------------
pt1 = 5;
F1(pt1:maxlp) = F1(1)*0.8;

%--------------------------------------------------------------------
%    PID controller parameters
%--------------------------------------------------------------------
Ctrl(1) = 45.8839;%3.7723;                         %Kc
Ctrl(2) = 0.05;%19.6000;   %Ti
Ctrl(3) = 0.0125;%4.9000;              %Td

%--------------------------------------------------------------------
%         Calculate pH at t = 0
%--------------------------------------------------------------------
ans = [1 (Ka(1) + coxi(1)) (Ka(1)*coxi(1)-Ka(1)*xi(1)-Kw) -Ka(1)*Kw];
Hans = roots(ans);
H(1) = max(Hans);
pH(1) = -log10(H(1));
e(1) = pHset - pH(1);

%--------------------------------------------------------------------
%         main loop calculation
%--------------------------------------------------------------------
for i = 1 : maxlp-1
%--------------------------------------------------------------------
%         pH measurement
%         estimate xi(i+1) and coxi(i+1) using ode15s method in matlab toolbox
%--------------------------------------------------------------------
  tspan = [t(i) t(i+1)];
  init = [xi(i) coxi(i)];
  AddOp = [F1(i) F2(i) C1(i) C2(i) V];
  [tt yy] = ode15s('func_pH_ODE', tspan, init,[],AddOp);
  xi(i+1) = yy(size(yy,1),1);
  coxi(i+1) = yy(size(yy,1),2);
  a = [1 (Ka(i+1) + coxi(i+1)) (Ka(i+1)*coxi(i+1)-Ka(i+1)*xi(i+1)-Kw) -Ka(i+1)*Kw];
  Hans = roots(a);
  H(i+1) = max(Hans);
  pH(i+1) = -log10(H(i+1));
  pH(i+1) = round(pH(i+1)*1e6)/1e6;
%--------------------------------------------------------------------
%         pH controller
%--------------------------------------------------------------------
  e(i+1) = pHset - pH(i+1);
  if abs(e(i+1)) > onCtrl
    u = func_CtrlPID([e(i-1) e(i) e(i+1)], Ctrl, Ts);
```

```
    F2(i+1) = F2(i) + u;
    if F2(i+1) < F2min
      F2(i+1) = F2min;
    elseif F2(i+1) > F2max
      F2(i+1) = F2max;
    end
  else
    F2(i+1) = F2(i);
    e(i+1) = 0;
  end
end

IAE = func_IAE(t,pH,pHset);
disp(sprintf('PID controller has IAE = %g', IAE))
disp('  ')

figure
plot(t,pH,'-','LineWidth',1.5)
axis([0 tfinal 3 11])
title('pH (controlled variable)')
xlabel('Time, min')
ylabel('pH')

figure
plot(t,F2,'-','LineWidth',1.5)
axis([0 tfinal 0 800])
title('Flow rate of Acid Stream (manipulated variable)')
xlabel('Time, min')
ylabel('Flow rate, L/min')
```

1.7 โปรแกรมที่ใช้ในตอนที่ 6.4 – Self-tuning PID Controller

```
clear all;
%------------------------------------------------------------------
%        Define variables & matrix used in calculation
%------------------------------------------------------------------
maxlp            = 151;
Ts               = 0.1;                              %Sampling time (min)
tfinal           = Ts*(maxlp-1);
t                = linspace(0,tfinal,maxlp);
F1(maxlp)        = 0;
F2(maxlp)        = 0;
coxi(maxlp)      = 0;
xi(maxlp)        = 0;
C1(maxlp)        = 0;
C2(maxlp)        = 0;
H(maxlp)         = 0;
pH(maxlp)        = 0;
e(maxlp)         = 0;
pHset            = 7;

%------------------------------------------------------------------
%        Define parameter & initial conditions
%------------------------------------------------------------------
V                = 1000;                                     %Tank volume (L)
F1(1:maxlp)      = 81;    %Acid Flow Rate (L/min)
F2(1:maxlp)      = 515.7923541247485;   %Base Flow Rate (L/min)
C1(1:maxlp)      = 0.32;                         %Acid concentration in feed (mol/L)
C2(1:maxlp)      = 0.05;                         %Base concentration in feed (mol/L)
xi(1)            = F1(1)*C1(1)/(F1(1)+F2(1));    %Acetate concentration in tank (mol/L)
coxi(1)          = F2(1)*C2(1)/(F1(1)+F2(1));    %Sodium concentration in tank (mol/L)
```

```
Kw              = 1e-14;                %Water constant
Ka(1:maxlp)     = 1.978e-5;                     %Acetic acid constant
F2min           = F2(1)/100;
F2max           = F2(1)*2;
onCtrl          = 0.0001*pHset;

%-------------------------------------------------------------------
%       Define disturbance variables
%-------------------------------------------------------------------
pt1             = 5;
C1(pt1:maxlp)   = C1(1)*1.15;

%-------------------------------------------------------------------
%    PID controller parameters
%-------------------------------------------------------------------
Ctrl(1)         = 45.8839;              %Kc
Ctrl(2)         = 0.05;                 %Ti
Ctrl(3)         = 0.0125;               %Td

%-------------------------------------------------------------------
%       Calculate pH at t = 0
%-------------------------------------------------------------------
ans             = [1 (Ka(1) + coxi(1)) (Ka(1)*coxi(1)-Ka(1)*xi(1)-Kw) -Ka(1)*Kw];
Hans            = roots(ans);
H(1)            = max(Hans);
pH(1)           = -log10(H(1));
e(1)            = pHset - pH(1);
time2           = 7;
count           = 0;
Theta           = [0.047075 0.045 0.025];

%-------------------------------------------------------------------
%       main loop calculation
%-------------------------------------------------------------------
for i = 1 : maxlp-1
%-------------------------------------------------------------------
%       pH measurement
%       estimate xi(i+1) and coxi(i+1) using ode15s method in matlab toolbox
%-------------------------------------------------------------------
  tspan     = [t(i) t(i+1)];
  init      = [xi(i) coxi(i)];
  AddOp     = [F1(i) F2(i) C1(i) C2(i) V];
  [tt yy]   = ode15s('func_pH_ODE', tspan, init,[],AddOp);
  xi(i+1)   = yy(size(yy,1),1);
  coxi(i+1) = yy(size(yy,1),2);
  a         = [1 (Ka(i+1) + coxi(i+1)) (Ka(i+1)*coxi(i+1)-Ka(i+1)*xi(i+1)-Kw) -Ka(i+1)*Kw];
  Hans      = roots(a);
  H(i+1)    = max(Hans);
  pH(i+1) = -log10(H(i+1));
  pH(i+1) = round(pH(i+1)*1e6)/1e6;
%-------------------------------------------------------------------
%       pH controller
%-------------------------------------------------------------------
  e(i+1) = pHset - pH(i+1);
  if abs(e(i+1)) > onCtrl
    if count > time2
      time1 = round(Theta(3)/Ts)+3;
      [Ctrl, Theta] = func_GASelf([F2(i-time1:i) pH(i-time1:i+1) pH(1)],Ts,Theta,Ctrl);
      count = 0;
    else
      count = count + 1;
    end
    u = func_CtrlPID([e(i-1) e(i) e(i+1)], Ctrl, Ts);
```

```
      F2(i+1) = F2(i) + u;
      if F2(i+1) < F2min
         F2(i+1) = F2min;
      elseif F2(i+1) > F2max
         F2(i+1) = F2max;
      end
   else
      F2(i+1) = F2(i);
      e(i+1) = 0;
   end
end

IAE = func_IAE(t,pH,pHset);
disp(sprintf('PID controller has IAE = %g', IAE))
disp('  ')

figure
plot(t,pH,'-','LineWidth',1.5)
axis([0 tfinal 3 11])
title('pH (controlled variable)')
xlabel('Time, min')
ylabel('pH')

figure
plot(t,F2,'-','LineWidth',1.5)
axis([0 tfinal 0 800])
title('Flow rate of base stream (manipulated variable)')
xlabel('Time, min')
ylabel('Flow rate, L/min')
```

2. โปรแกรมหลัก

2.1 ฟังก์ชัน CrossOver

```
function [c1,c2] = func_CrossOver(pa1,pa2,bounds)
% pa1      - the first parent ( [solution string function value] )
% pa2      - the second parent ( [solution string function value] )
c1 = pa1;
c2 = pa2;

for i = 1:length(pa1)-1
   aa = num2str(pa1(i),'%0.4f');
   bb = length(aa);                                          %number of digit
   j = 1;
   while aa(j) ~= '.'
      j = j+1;
   end
   m = j;                                                    %location of dot

   aaa = num2str(pa2(i),'%0.4f');
   bbb = length(aaa);                                        %number of digit
   j = 1;
   while aaa(j) ~= '.'
      j = j+1;
   end
   mm = j;                                                   %location of dot

   while m > mm
      aaa = ['0' aaa];
      mm = mm+1;
           end
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
            while mm > m
            aa = ['0' aa];
    m = m+1;
  end

  kk = length(aa);
  xPoint = round(rand*(kk-1))+1;
  if xPoint == m
    xPoint = round(rand*(kk-1))+1;
  end

  jj = [aa(1:xPoint) aaa(xPoint+1:kk)];
  jjj = [aaa(1:xPoint) aa(xPoint+1:kk)];
  c1(i) = str2num(jj);
  if c1(i) < bounds(i,1)
    c1(i) = bounds(i,1);
  else if c1(i) > bounds(i,2)
        c1(i) = bounds(i,2);
      end
  end

  c2(i) = str2num(jjj);
  if c2(i) < bounds(i,1)
    c2(i) = bounds(i,1);
  else if c2(i) > bounds(i,2)
        c2(i) = bounds(i,2);
      end
  end

end

c1(length(pa1)) = 0;
c2(length(pa1)) = 0;
```

2.2 ฟังก์ชัน func_GA

```
function [x,Pop] = func_GA(bounds,evalFN,evalOps,Pop,termOps,xOverLoop,mutLoop,accura)

%This function is to perform Genetic Algorithms Optimization
%with Cross Over Operation and Mutation Operation.
e1str=['[c1 c1(xZomeLength)]=' evalFN '(c1, evalOps);'];
e2str=['[c2 c2(xZomeLength)]=' evalFN '(c2, evalOps);'];

xZomeLength = size(Pop,2);              %Length of the xzome = numVars+fitness
numVar      = xZomeLength-1;            %Number of variables
popSize     = size(Pop,1);              %Number of individuals in the pop
c1          = zeros(1,xZomeLength);     %An individual
c2          = zeros(1,xZomeLength);     %An individual
done        = 0;                        %Done with simulated evolution
gen         = 1;                        %Current Generation Number
bPop        = zeros(1,termOps+1);       %history of best values

[Pop, best1, best2, worst2, worst1] = func_Sort(Pop);
bPop(1) = best1(numVar+1);

while(~done)
  %Perform CROSSOVER operation
  for j=1:xOverLoop,
    a = round(rand*(popSize-1)+1);      %Random pick a parent
    b = round(rand*(popSize-1)+1);      %Random pick another parent
    if and(a == b, a ~= popSize)
      b = a + 1;
```

```
    end
    [c1 c2] = feval('func_CrossOver',Pop(a,:),best1,bounds);
    eval(e1str);
    eval(e2str);
    Pop(popSize+2,:) = c1;
    Pop(popSize+1,:) = c2;
    [Pop, best1, best2, worst2, worst1] = func_Sort(Pop);
    Pop = Pop(1:popSize,:);                  %select 2 worst member out
  end
  %Perform MUTATION operation
  for j=1:mutLoop,
    a = round(rand*(popSize-1)+1);           %Random pick a member
    c1 = feval('func_Mutation',Pop(a,:),bounds);
    eval(e1str);
    Pop(popSize+1,:) = c1;
    [Pop, best1, best2, worst2, worst1] = func_Sort(Pop);
    Pop = Pop(1:popSize,:);                  %select the worst member out
  end
  bPop(gen) = best1(numVar+1);
  if or(gen > termOps, -best1(numVar+1) < accura)
    if gen > 3
      done = 1;
    else
      gen = gen + 1;
    end
  else
        gen = gen+1;
  end
end
x = Pop(1,:);
```

2.3 ฟังก์ชัน func_GASelfFit_ctrl

```
function [sol,val] = func_GASelfFit_ctrl(sol,opt)
% This function is to find IAE of 1st order plus dead time process

Kc = sol(1);                                  %Kc
Ti = sol(2);                                  %Ti
Td = sol(3);                                  %Td
l = length(opt)-4;
n = (l-1)/2;                                  % number of U
m = l-n;                                      % number of Y
U = zeros(1,2*n);
Y = zeros(1,2*n+1);
U(1:n)    = opt(1:n);                         % Input
Y(1:n+1)        = opt(m:l);                   % Output
Sp              = opt(l+1);
Kp              = opt(l+2);                   % Estimated Process Gain
T               = opt(l+3);                   % Estimated Process Time Constant
Z               = opt(l+4);                   % Estimated Dead Time
z               = round(Z/Ts)+1;
Ts              = 0.4;
IAE             = 0;
for i = n+1:2*n
  E0     = sp - Y(i-2);
  E1     = sp - Y(i-1);
  E2     = sp - Y(i);
  U      = Kc*( E2-E1 + Ts/Ti*E2 + Td/Ts*(E2 - 2*E1 + E0) );
  U(i)   = U(i-1) + u;
  Y(i+1) = Y(i) + Ts/T*(-Y(i)+Kp*U(i-z));
  IAE    = IAE + abs(sp-Y(i+1))*Ts;
```

```
end
val                    = -IAE;
```

2.4 ฟังก์ชัน func_GASelfFit_plant

```
function [sol,val] = func_GASelfFit_plant(sol,opt)

Kp = sol(1);                                    %Kc
T = sol(2);                                     %Ti
Z = sol(3);                                     %Td
dt = 0.1;
z = round(Z/dt);
l = length(opt);
n = l/2;
m = l-n;
U = opt(1:n);                                   %Process Input
Y = opt(m+1:l);                                 %Process Output
err = 0;
for i = 1:m-1
  if i-z >= 1
    Yest(i+1) = Y(i) + dt/T*(-Y(i) + Kp*U(round(i-z)));
    err = err + (Y(i+1)-Yest(i+1))^2;
  end
end
if err == 0
  err = 1;
end

val = -err;
```

2.5 ฟังก์ชัน func_InitPop

```
function [pop] = func_InitPop(numPop, bounds, evalFN, Options,Theta)

estr            =['[ pop(i,:) pop(i,xZomeLength)]=' evalFN '(pop(i,:),Options);'];
numVars         = size(bounds,1);                        %Number of variables
rng             = (bounds(:,2)-bounds(:,1))';        %Each variable ranges
xZomeLength     = numVars+1;                         %Length of string is numVar + fit
pop             = zeros(numPop,xZomeLength);     %Allocate the new population
pop(:,1:numVars)=(ones(numPop,1)*rng).*(rand(numPop,numVars))+(ones(numPop,1)*bounds(:,1)');

% Assign the guest parameters or the previous loop parameters
m               = length(Theta);
for i = 1:m/3
  pop(i,:) = [Theta(3*i-2:3*i) 0];
end

% Evaluate each parameters set for the fitness
for i=1:numPop
 eval(estr);
end
```

2.6 ฟังก์ชัน func_Mutation

```
function [c1] = func_Mutation(parent,bounds)

% The Mutation operation changes one of the parameters of the parent
```

```
% based on randomize process.
% The effective digit is 3 digits.

g11     = num2str(parent(1),'%0.3f');
g12     = num2str(parent(2),'%0.3f');
g13     = num2str(parent(3),'%0.3f');
k1      = length(g11);
k2      = length(g12);
k3      = length(g13);
mPoint  = round(rand*(k1-1)+1);                    %mPoint must never equal to zero
while or(g11(mPoint) == '.', mPoint > k1)
  mPoint = round(rand*(k1-1)+1);
end
g11(mPoint) = num2str(round(rand*9));

mPoint = round(rand*(k2-1)+1);                     %mPoint must never equal to zero
while or(g12(mPoint) == '.', mPoint > k2)
  mPoint = round(rand*(k2-1)+1);
end
g12(mPoint) = num2str(round(rand*9));

mPoint = round(rand*(k3-1)+1);                     %mPoint must never equal to zero
while or(g13(mPoint) == '.', mPoint > k3)
  mPoint = round(rand*(k3-1)+1);
end
g13(mPoint) = num2str(round(rand*9));

c1 = [str2num(g11) str2num(g12) str2num(g13) 0];
for i = 1:length(c1)-1
    if c1(i) < bounds(i,1)
    c1(i) = bounds(i,1);
  else if c1(i) > bounds(i,2)
        c1(i) = bounds(i,2);
      end
  end
end
```

2.7 ฟังก์ชัน func_Sort

```
function [sortPop, best1, best2, worst2, worst1] = func_Sort(Pop)
%This function will sort population from highest fitness to lowest fitness
D         = size(Pop);
sortPop   = zeros(D);
[y, index] = sort(Pop(:,D(2)));
n         = length(index);
for i = 1:n
  sortPop(i,:) = Pop(index(n+1-i),:);
end
best1     = sortPop(1,:);                          %First best fitness of current pop
best2     = sortPop(2,:);                          %Second best fitness of current pop
worst2    = sortPop(n-1,:);                        %Second worst fitness of current pop
worst1    = sortPop(n,:);                          %First worst fitness of current pop
```

2.9 ฟังก์ชัน func_pH_ODE

```
function dydt = pH_ODE(t,y,option,AddOp)

%------------------------------------------------------------------
%         Define parameter & initial conditions
```

```
%-------------------------------------------------------------------
V          = AddOp(5);                    %Tank volume (L)
F1         = AddOp(1);                    %Acid Flow Rate (L/min)
F2         = AddOp(2);                    %Base Flow Rate (L/min)
C1         = AddOp(3);                    %Acid concentration in feed (mol/L)
C2         = AddOp(4);                    %Base concentration in feed (mol/L)
xi         = y(1);
coxi       = y(2);
dxidt      = ( 1/V )*( F1*C1 - (F1+F2)*xi );
dcoxidt    = ( 1/V )*( F2*C2 - (F1+F2)*coxi );
dydt       = [dxidt; dcoxidt];
```

2.10 ฟังก์ชัน func_2nd_ODE

```
function dydt = func_2nd_ODE(t,y,option,AddOp)

%-------------------------------------------------------------------
%          Define parameter & initial conditions
%-------------------------------------------------------------------
F          = AddOp(1);                    %Tank volume (L)
Ca0        = AddOp(2);                    %Flow Rate (L/min)
V          = AddOp(3);                    %Concentration in feed (mol/L)
k          = AddOp(4);
Ca1        = y(1);
Ca2        = y(2);

dCa1dt     = ( F/V )*(Ca0 - Ca1) - k*Ca1;
dCa2dt     = ( F/V )*(Ca1 - Ca2) - k*Ca2;
dydt       = [dCa1dt; dCa2dt];
```

2.11 ฟังก์ชัน func_CtrlPID

```
function u = func_CtrlPID(E, Para, Ts)

E0 = E(1);
E1 = E(2);
E2 = E(3);
Kc = Para(1);
Ti = Para(2);
Td = Para(3);

%-------------------------------------------------------------------
%          calculate controller output
%          using digital velocity PID control law
%-------------------------------------------------------------------
u = Kc*( E2-E1 + Ts/Ti*E2 + Td/Ts*(E2 - 2*E1 + E0) );          %Proportional Integral Derivative mode
```

2.12 ฟังก์ชัน func_IAE

```
function intgrl = func_IAE(t,value,setpoint)
%IAE calculation

dt         = t(2) - t(1);
intgrl     = 0;
for i = 1:length(t)
  intgrl = intgrl + abs(setpoint - value(i))*dt;
end
```

ภาคผนวก ข.

การปรับค่าพารามิเตอร์ของตัวควบคุมพีไอดีจากแบบจำลองกระบวนการอันดับหนึ่งที่มีเวลาหน่วง

การปรับค่าพารามิเตอร์ของตัวควบคุมพีไอดี โดยอาศัยค่าพารามิเตอร์จากแบบจำลองกระบวนการอันดับหนึ่งและมีเวลาหน่วงดังสมการ (ข.1)

$$G_p = \frac{Ke^{-\theta s}}{\tau s + 1} \qquad (ข.1)$$

เมื่อกำหนด

K เป็นค่าเกน (Process Gain) ของกระบวนการ

τ เป็นค่าเวลาคงที่ (Time Constant) ของกระบวนการ

θ เป็นค่าเวลาหน่วง (Dead Time) ของกระบวนการ

เพื่อใช้ในการคำนวณค่าพารามิเตอร์ของตัวควบคุมพี พีไอ หรือ พีไอดี สามารถคำนวณได้จากความสัมพันธ์ที่ได้มีการวิจัยและปรากฏในงานวิจัยหลายๆ ฉบับ ซึ่งความสัมพันธ์ที่ใช้กันอย่างแพร่หลาย และเป็นต้นกำเนิดของการหาความสัมพันธ์ในลักษณะนี้คือ ความสัมพันธ์ของ Ziegler-Nichols ซึ่งแสดงดังในตารางที่ ข.1

ตารางที่ ข.1 กฎการปรับค่าตัวควบคุมพีไอดีตามหลักของ Ziegler-Nichols

ชนิดตัวควบคุม	K_c	τ_I	τ_D
P	$\frac{\tau}{K\theta}$	-	-
PI	$\frac{0.9\tau}{K\theta}$	3.33θ	-
PID	$\frac{1.2\tau}{K\theta}$	2.0θ	0.5θ

ความสัมพันธ์ของ Ziegler-Nichols จะใช้งานได้ดีเมื่อสัดส่วนของเวลาหน่วงต่อเวลาคงที่ของกระบวนการมีค่าคือ $0.1 < \theta / \tau < 1.0$

สำหรับความสัมพันธ์ที่ได้มีการศึกษาอื่นๆ จะแสดงอยู่ในตารางถัดไป

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข.2 กฎการปรับค่าตัวควบคุมพีไอดีตามหลักของ Cohen-Coon

ชนิดตัวควบคุม	K_c	τ_I	τ_D
P	$\frac{1}{K}\frac{\tau}{\theta}\left[1+\frac{1}{3}\frac{\theta}{\tau}\right]$	-	-
PI	$\frac{1}{K}\frac{\tau}{\theta}\left[0.9+\frac{1}{12}\frac{\theta}{\tau}\right]$	$\theta\left[\frac{30+3\frac{\theta}{\tau}}{9+20\frac{\theta}{\tau}}\right]$	-
PD	$\frac{1}{K}\frac{\tau}{\theta}\left[\frac{5}{4}+\frac{1}{6}\frac{\theta}{\tau}\right]$	-	$\theta\left[\frac{6-2\frac{\theta}{\tau}}{22+3\frac{\theta}{\tau}}\right]$
PID	$\frac{1}{K}\frac{\tau}{\theta}\left[\frac{4}{3}+\frac{1}{4}\frac{\theta}{\tau}\right]$	$\theta\left[\frac{32+6\frac{\theta}{\tau}}{13+8\frac{\theta}{\tau}}\right]$	$\theta\left[\frac{4}{11+2\frac{\theta}{\tau}}\right]$

ความสัมพันธ์ของ Cohen-Coon จะใช้งานได้ดีเมื่อสัดส่วนของเวลาหน่วงต่อเวลาคงที่ของกระบวนการมีค่าคือ $0.1<\theta/\tau<1.0$

ตารางที่ ข.3 กฎการปรับค่าตัวควบคุมพีไอดีตามหลักของการทำให้ค่า ITAE ต่ำสุด

ชนิดตัวควบคุม	K_c	τ_I	τ_D
P (dist.)	$\frac{0.49}{K}\left[\frac{\theta}{\tau}\right]^{1.084}$	-	-
PI (set.)	$\frac{0.586}{K}\left[\frac{\tau}{\theta}\right]^{0.916}$	$\frac{\tau}{1.03-0.165\frac{\theta}{\tau}}$	-
PI (dist.)	$\frac{0.859}{K}\left[\frac{\tau}{\theta}\right]^{0.977}$	$\frac{\tau}{0.674}\left[\frac{\theta}{\tau}\right]^{0.680}$	-
PID (set)	$\frac{0.965}{K}\left[\frac{\tau}{\theta}\right]^{0.855}$	$\frac{\tau}{0.796-0.147\frac{\theta}{\tau}}$	$0.308\tau\left[\frac{\theta}{\tau}\right]^{0.929}$
PID (dist.)	$\frac{1.357}{K}\left[\frac{\tau}{\theta}\right]^{0.947}$	$\frac{\tau}{0.842}\left[\frac{\theta}{\tau}\right]^{0.738}$	$0.381\tau\left[\frac{\theta}{\tau}\right]^{0.995}$

ความสัมพันธ์ในการคำนวณค่าพารามิเตอร์ของตัวควบคุมพีไอดี ตามหลักของการทำให้ค่า ITAE ต่ำสุดจะใช้งานได้ดีเมื่อสัดส่วนของเวลาหน่วงต่อเวลาคงที่ ของกระบวนการมีค่าคือ $0.1<\theta/\tau<1.0$

ภาคผนวก ค.

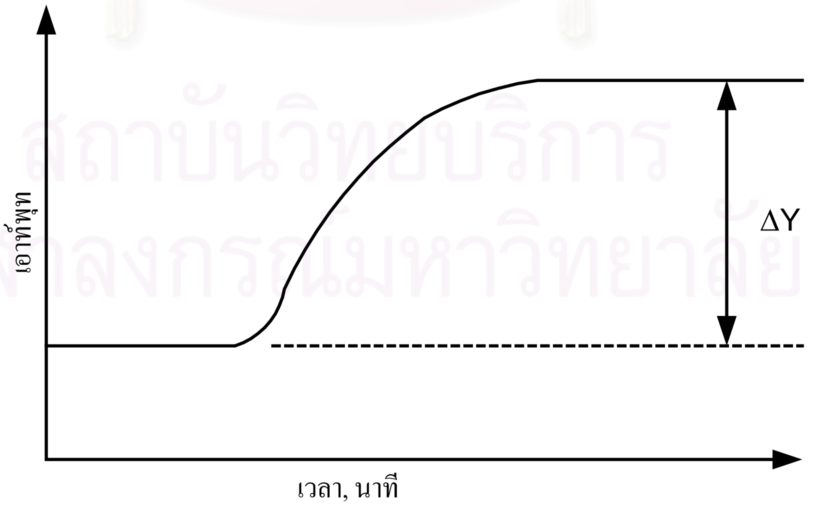
การประมาณกระบวนการด้วยแบบจำลองกระบวนการ

อันดับหนึ่งและมีเวลาหน่วง

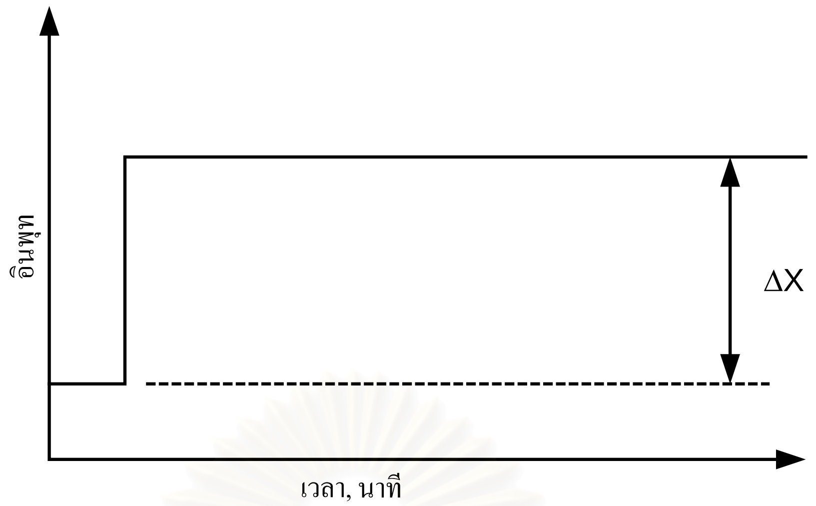
เนื่องจากกระบวนการต่างๆ ที่เกิดขึ้นจริงในอุตสาหกรรม เป็นกระบวนการที่มีการตอบสนองอันดับหนึ่งอยู่น้อยมาก กระบวนการโดยทั่วไปจึงมีอันดับของการตอบสนองของกระบวนการที่สูงกว่าหนึ่งเสมอ แต่จากการศึกษาวิจัยในหลายๆ งานวิจัยที่ผ่านมา พอที่จะสามารถสรุปได้ว่ากระบวนการที่มีการตอบสนองอันดับสูงกว่าหนึ่งนั้น สามารถที่จะประมาณและควบคุมได้ด้วยตัวควบคุมพีไอดีที่มีการปรับค่าพารามิเตอร์ของตัวควบคุม โดยอาศัยค่าพารามิเตอร์ของกระบวนการจากการประมาณแบบจำลองของกระบวนการ ให้อยู่ในรูปของกระบวนการอันดับหนึ่งและมีเวลาหน่วง (First-Order-with-Dead-Time) โดยที่ให้ผลการควบคุมอยู่ในระดับที่ยอมรับได้ และใช้กันโดยแพร่หลาย

ค.1 การประมาณค่าด้วยวิธีกราฟิก

ในตอนนี้จะกล่าวถึงวิธีการหาแบบจำลองโดยประมาณของกระบวนการ ด้วยการประมาณให้อยู่ในรูปของกระบวนการอันดับหนึ่งและมีเวลาหน่วง โดยอาศัยการพิจารณาจากเส้นโค้งการตอบสนองของกระบวนการ (Process Reaction Curve) ดังรูปที่ ค.1 เมื่อมีการปรับอินพุทของกระบวนการแบบขั้น (Step Input) ตามที่แสดงในรูปที่ ค.2

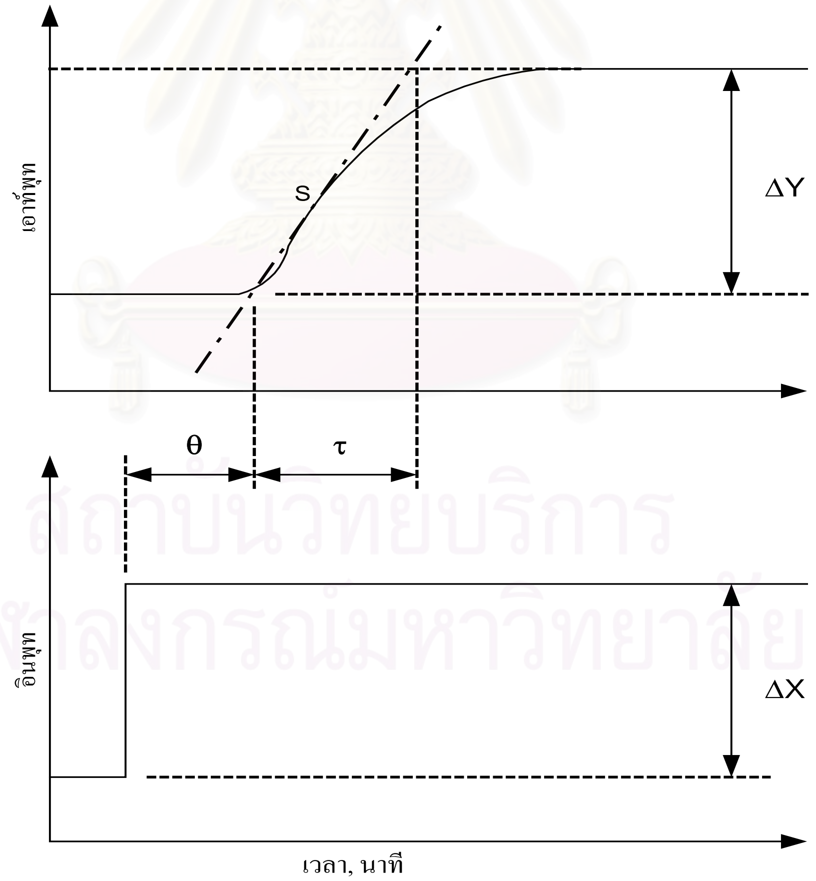


รูปที่ ค.1 การตอบสนองของเอาท์พุทของกระบวนการ เมื่อมีอินพุท ดังรูป ค.2



รูปที่ ค.2 การเปลี่ยนแปลงแบบขั้นของอินพุท ที่เข้าสู่กระบวนการ

จากรูปที่ ค.1 ต้องการหาแบบจำลองโดยประมาณของกระบวนการ สามารถทำได้โดยการลากเส้นตรงให้สัมผัสกับจุดบนเส้นโค้งที่มีความชันสูงที่สุด ดังรูปที่ ค.3 แล้วคำนวณดังนี้



รูปที่ ค.3 วิธีการหาค่าประมาณกระบวนการอันดับหนึ่งและมีเวลาหน่วงแบบกราฟิก

กระบวนการที่มีการตอบสนองอันดับหนึ่งดังสมการ (ค.1)

$$G_p = \frac{K_p e^{-\theta s}}{\tau s + 1} \qquad (ค.1)$$

มีค่าพารามิเตอร์ที่ต้องการทราบค่าอยู่ 3 ตัว นั่นคือ K_p, τ, θ ซึ่งทั้ง 3 ค่านี้ สามารถคำนวณหาได้จากความสัมพันธ์ดังนี้

ค่าเกนของกระบวนการ

$$K_p = \frac{\Delta Y}{\Delta X}$$

ค่าเวลาคงที่ของกระบวนการ

$$\tau = \Delta Y / S$$

= (เวลาที่เส้นสัมผัสตัดกับ 100% ΔY - เวลาที่เส้นสัมผัสตัดกับ 0% ΔY)

เมื่อ S เป็นความชันของเส้นสัมผัสที่ลากขึ้นมา

ค่าเวลาหน่วงของกระบวนการ

$\theta =$ เวลาที่เส้นสัมผัสตัดกับ 0% ΔY – เวลาที่อินพุทมีการเปลี่ยนแปลง

ค.2 การประมาณค่ากระบวนการแบบเวลาไม่ต่อเนื่อง (Discrete Time Model)

การควบคุมกระบวนการด้วยคอมพิวเตอร์ เป็นการควบคุมในระบบดิจิตอล ที่ไม่ได้เกิดขึ้นจากการวัดค่าอินพุทและเอาท์พุทของกระบวนการอย่างต่อเนื่อง แต่เกิดจากการวัดค่าของกระบวนการทุกๆ ช่วงเวลาที่กำหนดไว้ หรือเรียกว่า เวลาในการเก็บตัวอย่าง (Sampling Time) ดังนั้นข้อมูลของกระบวนการจะมีค่าอยู่ ณ เวลาที่มีการเก็บตัวอย่างเท่านั้น ทำให้ข้อมูลจะถูกเก็บอยู่ในรูปของชุดข้อมูล (Series of Data)

แต่อย่างไรก็ตาม ยังสามารถที่จะประมาณค่าพารามิเตอร์ของกระบวนการให้อยู่ในรูปของกระบวนการที่มีการตอบสนองอันดับหนึ่งและมีเวลาหน่วงได้ ซึ่งจะใช้ความสัมพันธ์ของชุดข้อมูลอินพุทและเอาท์พุท เช่นเดียวกันกับในหัวข้อ ค.1

ในโดเมนของเวลาไม่ต่อเนื่อง (Discrete Time Model) กระบวนการอันดับหนึ่งที่มีเวลาหน่วง ดังสมการที่ (ค.1) สามารถประมาณได้ด้วยสมการ

$$\hat{y}(k+1) = \phi y(k) + K_p(1-\phi)u(k-m) \qquad (ค.2)$$

เมื่อ $\phi = e^{-\Delta t/\tau}$

$\theta = m\Delta t$

Δt เป็นเวลาในการเก็บตัวอย่าง

$y(k), \hat{y}(k)$ เป็นค่าเอาท์พุทของกระบวนการที่เวลา k ที่ได้จากการวัดจริง และค่าที่ได้จากการประมาณตามลำดับ

สามารถเขียนสมการที่ (ค.2) ได้ใหม่ว่า

$$\hat{y}(k+1) = ay(k) + bu(k-m) \qquad (ค.3)$$

ดังนั้นจะได้ว่า

$$K_p = \frac{b}{1-a}$$

$$\tau = \frac{-\Delta t}{\ln a}$$

$$\theta = m\Delta t$$

โดยที่ค่า a และ b ในสมการ (ค.3) สามารถหาได้จากการหาคำตอบที่ทำให้ความสัมพันธ์

$$\sum_{i=1}^{n}(y(k+1) - \hat{y}(k+1))_i^2 \qquad (ค.4)$$

หรือ

$$\sum_{i=1}^{n}(y(k+1) - ay(k) - bu(k-m))_i^2 \qquad (ค.5)$$

มีค่าต่ำที่สุด

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นายอนุชา ทิพยวัลย์ เกิดวันที่ 17 มกราคม พ.ศ. 2519 ที่จังหวัดตรัง สำเร็จการศึกษาระดับปริญญาตรีวิศวกรรมศาสตร์บัณฑิต สาขาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2538 และศึกษาต่อในหลักสูตร วิศวกรรมศาสตร์มหาบัณฑิต สาขาวิศวกรรมเคมี คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2543

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย