สำนักหอสมุดกลาง_พระจอมเกล้าลาดกระบัง

ซอฟท์แวร์เลือกค่าพารามิเตอร์ของระบบควบคุมแบบ PID และ ฟัซซี่

SOFTWARE AIDED TUNING FOR PID & FUZZY CONTROL SYSTEMS

โดย

นายเกรียงไกร กัณฐโรจน์

นายไพโรจน์ ปกสุข

นายเสกสรรค์ แร่ทอง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตรบัณฑิต

สาขาเทคโนโลยีอิเล็กทรอนิกส์

ภาควิชาเทคนิคอุตสาหกรรม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

เลขหมู่.....
เลขทะเบียน 34034
วัน, เดือน, ป. 1 ต.ค. 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ซอฟท์แวร์เลือกค่าพารามิเตอร์ของระบบควบคุมแบบ PID และ ฟัซซี่
ชื่อนักศึกษา	นายเกรียงไกร กัณฐโรจน์ รหัสประจำตัว 39013267
	นายไพโรจน์ ปกสุข รหัสประจำตัว 39013283
	นายเสกสรรค์ แร่ทอง รหัสประจำตัว 39013299
อาจารย์ที่ปรึกษา	ดร.ปิติเขต สู้รักษา
	อ. มยุรี เลิศเวชกุล
ระดับการศึกษา	ปริญญาอุตสาหกรรมศาสตรบัณฑิต
	สาขาวิชาเทคโนโลยีอิเล็กทรอนิกส์
ภาควิชา	เทคนิคอุตสาหกรรม
ปีการศึกษา	2541

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตร์บัณฑิต

.....หัวหน้าภาควิชาเทคนิคอุตสาหกรรม

(ผศ.อุทัย ศรีธีระวิโรจน์)

คณะกรรมการสอบปริญญานิพนธ์

.....อาจารย์ที่ปรึกษา

(ดร.ปิติเขต สู้รักษา)

.....อาจารย์ที่ปรึกษาร่วม

(อ.มยุรี เลิศเวชกุล)

.....กรรมการ

(อ.ศิลกาย ธรรมวาโน)

.....กรรมการ

( )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	SOFTWARE AIDED TUNING FOR PID & FUZZY CONTROL SYSTEMS
Student	Mr. Kriengkai Kuntaroj No. 39013267
	Mr. Piroat Pogsuk No. 39013283
	Mr. Saksun Raethong No. 39013299
Thesis Advisor	Dr. Pitikhate Sooraksa
	Mrs. Mayuree Lertwatechakul
Level of Study	Bachelor of Industrial Technology
	Electronics
Department	Industrial Technology
Academic Year	1998

Accepted by the Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang in partial fulfillment of the requirements for the bachelor's degree

.....Chairman
(Asst.Prof. U-thai Sritheeravirojana)

Project Report Committee

.....Advisor
(Dr. Pitikhate Sooraksa)

.....Co Advisor
(Mrs. Mayuree Lertwatechakul)

.....Member
(Ms. Pitak thumwarin)

.....Member
( )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ซอฟท์แวร์เลือกค่าพารามิเตอร์ของระบบควบคุมแบบ PID และ ฟัซซี่
นักศึกษา	นายเกรียงไกร กัณฐโรจน์ รหัสประจำตัว 39013267
	นายไพโรจน์ ปกสุข รหัสประจำตัว 39013283
	นายเสกสรรค์ แร่ทอง รหัสประจำตัว 39013299
อาจารย์ผู้ควบคุมปริญญานิพนธ์	ดร.ปิติเขต สู้รักษา
	อ.มยุรี เลิศเวชกุล
ระดับการศึกษา	ปริญญาอุตสาหกรรมศาสตรบัณฑิต
	สาขาวิชาเทคโนโลยีอิเล็กทรอนิกส์
ภาควิชา	เทคนิคอุตสาหกรรม
ปีการศึกษา	2541

บทคัดย่อ

ปริญญานิพนธ์นี้ เป็นการศึกษาและประยุกต์ใช้ทฤษฎีของตัวควบคุม PID และ ฟัซซี่ เพื่อออกแบบซอฟท์แวร์จำลองการทำงานระบบควบคุม ตระกูล PID ดังต่อไปนี้คือ ตัวควบคุมแบบ P, PI, PD, PID, PID แบบ ซิกเลอร์นิโคล (Zigler Nichols PID) และตัวควบคุมแบบฟัซซี่

ซอฟท์แวร์จำลองการทำงานของตัวควบคุมแบบ PID และ ฟัซซี่ นั้น ได้เขียนโดยใช้โปรแกรมวิชวลเบสิก ซึ่งทำงานบนวินโดว์ ทำให้ใช้งานง่าย ผู้ใช้สามารถปรับแต่งค่าพารามิเตอร์ของตัวควบคุมให้เหมาะสมกับกระบวนการต่างๆได้ตามต้องการ โดยมีการแสดงผลเป็นทั้งกราฟและตัวเลข และจากการทดลอง ได้ผลการจำลองการควบคุมเป็นที่น่าพอใจ โดยตัวควบคุมแต่ละแบบจะเหมาะกับกระบวนการแตกต่างกันไป ซึ่งขึ้นอยู่กับผู้ใช้ว่าจะเลือกตัวควบคุมแบบใดจึงจะเหมาะกับงานมากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	SOFTWARE AIDED TUNING FOR PID & FUZZY CONTROL SYSTEMS		
Student	Mr. Kriengkai	Kuntaroj	No. 39013267
	Mr. Piroat	Pogsuk	No. 39013283
	Mr. Saksun	Raethong	No. 39013299
Thesis Advisor	Dr. Pitikhate	Sooraksa	
	Mrs. Mayuree	Lertwatechakul	
Level of Study	Bachelor of Industrial Technology Electronics		
Department	Industrial Technology		
Academic Year	1998		

ABSTARCT

This thesis studies and implements software for designing PID and fuzzy control systems. The software provides choices of controllers including P, PI, PD, PID, Zigler Nichols' PID and fuzzy controllers.

The software is written in Visual Basic and is a user friendly Windows environment. The users can adjust the control parameters as desired. The simulation results can be displayed numerically and graphically. According to the experiment, this software yietds satisfactory and reliable results and the choices of controllers depend upon the given processes.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การจัดทำปริญญานิพนธ์ฉบับนี้สามารถสำเร็จลุล่วงลงได้ด้วยดี เนื่องจากได้รับความช่วยเหลือเป็นอย่างดีจากอาจารย์ ดร.ปิติเขต สู้รักษา ที่ได้ให้ความกรุณาแนะนำ และให้ข้อคิดเห็นต่าง ๆ แก่ผู้จัดทำเป็นอย่างดีตลอดมา ผู้จัดทำรู้สึกซาบซึ้งและขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ อาจารย์มยุรี เลิศเวชกุล ที่ให้คำแนะนำและดูแลเอาใจใส่ในการจัดทำปริญญานิพนธ์ และให้คำปรึกษาในส่วนของการเขียนโปรแกรม

ขอขอบคุณเพื่อน ๆ ทุกคนที่ช่วยให้คำแนะนำ ให้ยืมเครื่องมือ และเป็นกำลังใจตลอดในการทำปริญญานิพนธ์นี้

สุดท้ายนี้ผู้จัดทำขอกราบขอบพระคุณ บิดา มารดา ที่เคารพรักยิ่ง ที่ได้สนับสนุนด้านการเงินและให้กำลังใจแก่ผู้จัดทำมาตลอด จนปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงลงด้วยดี

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย..... III

บทคัดย่อภาษาอังกฤษ..... IV

กิตติกรรมประกาศ..... V

สารบัญ..... VI

สารบัญรูป..... VIII

บทที่1 บทนำ..... 1

วัตถุประสงค์..... 1

ขอบเขตของโครงงาน..... 2

ขั้นตอนการดำเนินโครงงาน..... 2

บทที่ 2 ทฤษฎี..... 3

ระบบควบคุมแบบ Open - loop..... 3

ระบบควบคุมแบบ Closed - loop..... 4

หลักการออกแบบระบบควบคุม..... 5

ชนิดของตัวควบคุม..... 6

ตัวควบคุมแบบ Proportional Controller (P)..... 6

ตัวควบคุมแบบ ProportionaI Integral Controller (PI)..... 8

ตัวควบคุมแบบ Proportional Derivative Controller (PD)..... 9

ตัวควบคุมแบบ Proportional Integral Derivative Controller (PID)..... 13

State - space..... 17

ทฤษฎีฟัซซี่ลอจิก..... 19

คริชพเซต (Crisp set)..... 19

ฟัซซี่เซต (Fuzzy set)..... 20

ฟัซซี่ลอจิก (Fuzzy logic)..... 23

การประยุกต์ใช้งานของฟัซซี่ลอจิก (Fuzzy logic application)..... 23

กฎการวินิจฉัย (Inference)..... 24

วิธีการ Defuzzifier..... 25

หลักการหา Fuzzy Control Rule..... 28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบตัวควบคุมแบบ Fuzzy PD..... 29
บทที่ 3 การสร้างและพัฒนาโปรแกรม.....32
ความสามารถของโปรแกรม.....32
การออกแบบโปรแกรม.....32
ส่วนของหน้าต่างที่ใช้ติดต่อกับผู้ใช้.....33
ส่วนของโปรมแกรมคำนวณเพื่อหาเอาท์พุท..... 35
บทที่ 4 ส่วนประกอบหลักและวิธีการใช้โปรแกรม..... 42
หน้าต่าง Menu หลัก..... 42
หน้าต่าง Controller.....44
หน้าต่าง Plant..... 45
หน้าต่าง P Controller.....46
หน้าต่าง Controller แบบอื่น ๆ.....47
หน้าต่าง PID Zigler Nichols Controller..... 48
หน้าต่าง Fuzzy PD Controller..... 49
หน้าต่าง Graph..... 51
หน้าต่าง Graph P Controller.....52
หน้าต่าง Graph ของ Controller แบบอื่น ๆ.....53
หน้าต่าง Graph All..... 53
ส่วนของ Help File.....55
บทที่ 5 การทดลองและผลการทดลอง..... 58
การทดลองที่1 ตัวควบคุมแบบ P..... 58
การทดลองที่2 ตัวควบคุมแบบ PI.....61
การทดลองที่3 ตัวควบคุมแบบ PD..... 63
การทดลองที่4 ตัวควบคุมแบบ PID.....65
การทดลองที่5 ตัวควบคุมแบบ PID Zigler Nichols.....67
การทดลองที่6 ตัวควบคุมแบบ Fuzzy PD.....69
บทที่ 6 สรุปและวิจารณ์ผลการทดลอง..... 71
บรรณานุกรม.....75
ภาคผนวก ก Source Code โปรแกรม PID & Fuzzy PD Control System

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 2-1 ระบบควบคุมแบบ Open - Loop..... 3
รูปที่ 2-2 ระบบควบคุมแบบ Closed –Loop..... 4
รูปที่ 2-3 แสดงถึงระบบควบคุมป้อนกลับแบบ Proportional..... 6
รูปที่ 2-4 แสดงถึงระบบควบคุมป้อนกลับแบบ PI..... 8
รูปที่ 2-5 แสดงถึงระบบควบคุมป้อนกลับแบบ PD..... 10
รูปที่ 2-6 Waveforms ของ c(t), e(t), และ de(t) / dt
แสดงถึงผลของการควบคุมแบบดีริเวทีฟ..... 11
รูปที่ 2-7 แสดงถึงระบบควบคุมป้อนกลับแบบ PID..... 13
รูปที่ 2-8 ระบบควบคุมแบบ Proportionnal..... 15
รูปที่ 2-9 แสดงการ Oscillation เพื่อหาค่า Pcr..... 15
รูปที่ 2-10 แสดงการ Inference..... 25
รูปที่ 2-11 แสดงโครงสร้างพื้นฐานของระบบฟัซซี่ลอจิก..... 27
รูปที่ 2-12 โครงสร้างพื้นฐานของระบบควบคุมป้อนกลับด้วย Fuzzy Controller..... 27
รูปที่ 2-13 Block Diagram ของระบบควบคุมแบบ Fuzzy PD..... 29
รูปที่ 2-14 Regions ของ Fuzzy controller..... 30
รูปที่ 3-1 Flow Chart หน้าต่าง Menu..... 33
รูปที่ 3-2 Flow Chart หน้าต่าง Controller..... 34
รูปที่ 3-3 Flow Chart ของ Plant..... 35
รูปที่ 3-4 Flow Chart ของ P Controller..... 36
รูปที่ 3-5 Flow Chart ของ PI Controller..... 37
รูปที่ 3-6 Flow Chart ของ PD Controller..... 38
รูปที่ 3-7 Flow Chart ของ PID Controller..... 39
รูปที่ 3-8 Flow Chart ของ PID Zigler Nichols..... 40
รูปที่ 3-9 Flow Chart แสดงการทำงานของ Fuzzy Controller..... 41
รูปที่ 4-1 หน้าต่าง Menu..... 42
รูปที่ 4-2 หน้าต่าง Controller..... 44
รูปที่ 4-3 หน้าต่าง Plant..... 45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4-4 หน้าต่าง P Controller..... 46
รูปที่ 4-5 หน้าต่าง หน้าต่าง PID Zigler Nichols Controller..... 48
รูปที่ 4-6 หน้าต่าง Fuzzy PD Controller..... 49
รูปที่ 4-7 หน้าต่าง Graph..... 51
รูปที่ 4-8 หน้าต่าง Graph P Controller..... 52
รูปที่ 4-9 หน้าต่าง Graph All..... 53
รูปที่ 4-10 แสดงหน้าต่าง Help Contents..... 55
รูปที่ 4-11 แสดงหน้าต่าง Help Contents..... 56
รูปที่ 4-12 แสดงหน้าต่าง Help Topic..... 56
รูปที่ 4-13 แสดงหน้าต่าง Help Topic..... 57
รูปที่ 4-14 แสดงหน้าต่าง Help Topic..... 57
รูปที่ 5-1 แสดงกราฟเมื่อปรับค่า Kp..... 59
รูปที่ 5-2 แสดงกราฟเอาท์พุทเมื่อเปลี่ยน Process..... 60
รูปที่ 5-3 แสดงกราฟเอาท์พุทเมื่อเปลี่ยนค่า Kp และ Ki..... 61
รูปที่ 5-4 แสดงกราฟเอาท์พุทเมื่อเปลี่ยน Process..... 62
รูปที่ 5-5 แสดงกราฟเอาท์พุทเมื่อเปลี่ยนค่า Kp และ Kd..... 63
รูปที่ 5-6 แสดงกราฟเอาท์พุทเมื่อเปลี่ยน Process..... 64
รูปที่ 5-7 แสดงกราฟเอาท์พุทเมื่อเปลี่ยนค่า Kp ,Ti และ Td..... 65
รูปที่ 5-8 แสดงกราฟเอาท์พุทเมื่อเปลี่ยน Process..... 66
รูปที่ 5-9 แสดงกราฟเอาท์พุทเมื่อเปลี่ยน Process..... 68
รูปที่ 5-10 แสดงกราฟเอาท์พุทเมื่อเปลี่ยนค่า Kp ,Ti และ Td..... 69
รูปที่ 5-11 แสดงกราฟเอาท์พุทเมื่อเปลี่ยน Process..... 70

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ในปัจจุบันการออกแบบระบบควบคุม มีขั้นตอนในการออกแบบค่อนข้างยุ่งยากโดยเฉพาะการปรับค่าพารามิเตอร์ของ Controller นั้นต้องอาศัยผู้มีประสบการณ์ในการปรับแต่ง และต้องมีการลองผิดลองถูกจนกว่าจะได้เอาท์พุทของระบบตามต้องการ ทำให้การปรับแต่งกับระบบควบคุมจริงๆ นั้น เป็นการเสี่ยงที่จะทำความเสียหายให้กับระบบ และสิ้นเปลืองค่าใช้จ่ายอีกด้วย เพราะฉะนั้นจึงต้องมีการจำลองการทำงานของระบบควบคุม เพื่อทดลองเลือก Controller ให้เหมาะกับกระบวนการและทดลองปรับแต่งค่าพารามิเตอร์ของ Controller ให้ได้ผลเป็นที่น่าพอใจก่อนจึงค่อยนำไปใช้งานจริง

สำหรับโครงงานที่นำเสนอนี้เป็นซอฟแวร์จำลองการควบคุมขบวนการ โดยใช้ตัวควบคุมแบบ PID และ Fuzzy PD โดยจำลอง Process ด้วย Transfer function

การควบคุมกระบวนการ ในโครงงานที่นำเสนอนี้ทุกขั้นตอนจะจำลองการทำงานด้วย Computer โดยระบบที่เราจำลองขึ้นทั้งหมดเป็นระบบอินพุทเดียว เอาท์พุทเดียว ซึ่งตัวซอฟแวร์ที่ออกแบบขึ้นนี้จะทำงานบนวินโดว์ ทำให้ใช้งานง่าย สะดวก โดยมีการแสดงผลค่าเอาท์พุทเป็นกราฟและตัวเลข

วัตถุประสงค์

1. เพื่อศึกษาการออกแบบระบบควบคุมด้วยตัวควบคุมแบบ PID และ Fuzzy PD
2. เพื่อศึกษาการปรับแต่งค่าพารามิเตอร์ของตัวควบคุมแบบ PID และ Fuzzy PD ให้เหมาะสมกับกระบวนการ (Process)
3. เพื่อสร้างซอฟแวร์จำลองการทำงานของระบบควบคุม บน PC Computer
4. เพื่อใช้ซอฟแวร์จำลองการทำงานของระบบควบคุม ช่วยในการออกแบบระบบควบคุม
5. เพื่อเปรียบเทียบระบบควบคุมที่ควบคุมด้วยตัวควบคุมแบบ PID และ Fuzzy PD
6. เพื่อศึกษาแนวทางในการนำระบบควบคุมที่ออกแบบได้ ไปประยุกต์ใช้งานจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขอบเขตของโครงงาน

1. สามารถจำลองการทำงานของตัวควบคุมแบบต่างๆ ได้ดังนี้
   - ตัวควบคุมแบบ P
   - ตัวควบคุมแบบ PI
   - ตัวควบคุมแบบ PD
   - ตัวควบคุมแบบ PID
   - ตัวควบคุมแบบ PID Zigler Nichols
   - ตัวควบคุมแบบ Fuzzy PD
2. สามารถปรับแต่งค่า Process ได้ โดยจำลอง Process เป็น Transfer function
3. สามารถปรับแต่งค่าพารามิเตอร์ของตัวควบคุมแต่ละตัวได้
4. การแสดงผลค่าเอาท์พุทสามารถแสดงผลได้ทั้งกราฟ และตัวเลข

ขั้นตอนการดำเนินโครงงาน

1. ศึกษาระบบควบคุม และตัวควบคุมแบบ PID และ Fuzzy PD
2. ศึกษา การเขียนโปรแกรม ภาษา Visual Basic 4.0
3. ออกแบบ Algorithm ของโปรแกรม และการแสดงผลของโปรแกรม
4. เขียนโปรแกรมจำลองการทำงานของระบบควบคุม แบบ PID และ Fuzzy PD
5. ทดสอบการทำงานของโปรแกรม
6. ตรวจสอบความถูกต้องของผลลัพธ์ ถ้าผิดพลาดก็ทำการแก้ไข
7. ทำการทดลองเพื่อเปรียบเทียบตัวควบคุมแบบต่าง ๆ
8. สรุปผล และเขียนรายงานการจัดทำโครงงาน พร้อมทั้งนำเสนอผลงาน

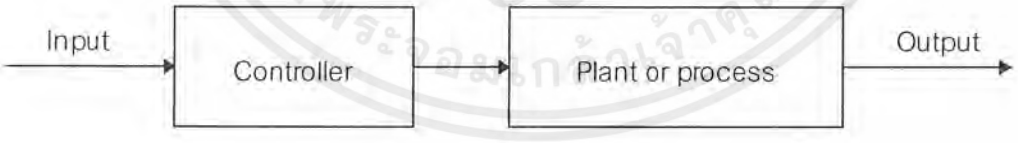
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎี

ในการออกแบบระบบควบคุม จะต้องอาศัยความรู้พื้นฐานเกี่ยวกับระบบควบคุม เพื่อใช้เป็นพื้นฐานในการทำความเข้าใจโครงสร้างและหลักการทำงานของระบบควบคุม และเพื่อใช้ในการการกำหนดโครงสร้างและวิธีการออกแบบให้เหมาะสมกับลักษณะการใช้งาน ดังนั้นเนื้อหาในบทนี้จะกล่าวถึงทฤษฎีทั่วไปเกี่ยวกับระบบควบคุม ซึ่งมีรายละเอียดดังต่อไปนี้

ระบบควบคุมแบบ Open – loop

ระบบควบคุมแบบ Open – loop เป็นระบบควบคุมที่เอาท์พุทของระบบจะไม่มีผลต่อการควบคุมเลย คือเอาท์พุทของระบบจะไม่ถูกป้อนกลับเพื่อนำเทียบกับอินพุท รูปที่ 2-1 แสดง Block diagram ของระบบควบคุมแบบ Open – loop ตัวอย่างที่ง่ายๆของระบบควบคุมแบบ Open – loop ได้แก่ เครื่องซักผ้า กล่าวคือขั้นตอนของการดูดน้ำเข้า การซัก และการปล่อยน้ำทิ้งนั้นจะเป็นไปตามเวลาที่ได้กำหนดไว้ล่วงหน้า แต่เครื่องซักผ้าไม่ได้มีการวัดเอาท์พุทซึ่งก็คือความขาวสะอาดของผ้าออกมาแต่อย่างใด



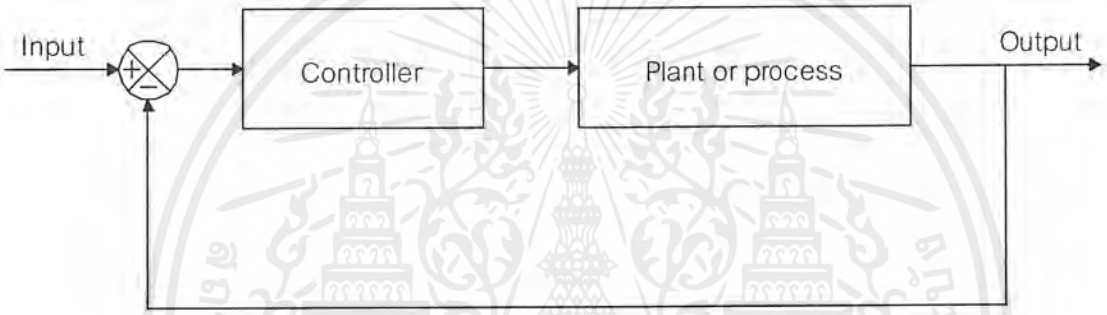
รูปที่ 2-1 ระบบควบคุมแบบ Open – loop

ในระบบควบคุมแบบ Open – loop นั้นเอาท์พุทไม่ได้ถูกนำมาเปรียบเทียบกับอินพุท ดังนั้นความเที่ยงตรงของระบบจะขึ้นอยู่กับการเปรียบเทียบ ในทางปฏิบัติแล้วจะสามารถใช้การควบคุมแบบ Open – loop ได้ถ้าทราบถึงความสัมพันธ์ระหว่างอินพุทและเอาท์พุทของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมแบบ Closed – loop

ระบบควบคุมแบบ Closed – loop เป็นระบบควบคุมที่สัญญาณเอาท์พุทจะมีผลโดยตรงต่อการควบคุม ดังนั้นระบบควบคุมแบบ Closed – loop ก็คือระบบควบคุมป้อนกลับนั่นเอง สัญญาณค่าความคลาดเคลื่อน (Actuating error signal) ซึ่งเป็นสัญญาณความแตกต่างระหว่างสัญญาณอินพุทกับสัญญาณป้อนกลับ (Feedback signal) จะถูกป้อนให้กับตัวควบคุม (Controller) เพื่อที่จะลดค่าความคลาดเคลื่อนให้น้อยลงและทำให้เอาท์พุทของระบบมีค่าตามที่ต้องการ รูปที่ 2-2 แสดง Block diagram ของระบบควบคุมแบบ Closed – loop



รูปที่ 2-2 ระบบควบคุมแบบ Closed – loop

ข้อดีของระบบควบคุมแบบ Closed – loop อย่างหนึ่งก็คือการใช้วิธีการป้อนกลับนำเอาสัญญาณเอาท์พุทมาเทียบกับสัญญาณอินพุทที่ต้องการ ดังนั้นจึงสามารถจะกำจัดหรือลดผลของ Disturbance จากภายนอกหรือผลของ Disturbance จากภายในที่เกิดจากการเปลี่ยนแปลงค่าของพารามิเตอร์ของระบบให้น้อยลงได้ ในบางครั้งจึงไม่จำเป็นที่จะต้องใช้อุปกรณ์ที่มีความเที่ยงตรงสูงและมีราคาแพงแต่อย่างใด แต่ในกรณีของระบบ Open – loop จะไม่เป็นเช่นนี้ กล่าวคือถ้ามี Disturbance เกิดขึ้นก็จะต้องทำการปรับอินพุทหรือพยายามกำจัด Disturbance นั้นทิ้งไปเพื่อให้เอาท์พุทมีค่าคงเดิมตามที่ต้องการ อุปกรณ์ต่างๆที่ใช้ก็ต้องมีความเที่ยงตรงสูง

สำหรับในแง่ของเสถียรภาพแล้ว ปัญหาของเสถียรภาพไม่ใช่ปัญหาใหญ่ของระบบควบคุมแบบ Open – loop เพราะสามารถจะเห็นได้ทันทีว่าระบบ Open – loop จะเสถียร (Stable) หรือไม่เสถียร (Unstable) ต่ออินพุทที่มากระทำ ดังนั้นการทำให้ระบบ Open – loop เป็นระบบที่เสถียรจึงเป็นสิ่งที่ทำได้ไม่ยาก แต่สำหรับระบบควบคุมแบบ Closed – loop นั้นเสถียรเป็นปัญหาที่สำคัญมากอย่างหนึ่ง ทั้งนี้เพราะมีการนำเอาสัญญาณเอาท์พุทมาเปรียบเทียบกับอินพุทเพื่อลดค่าความคลาดเคลื่อน ซึ่งอาจทำให้เกิดการ Oscillate ต่อเนื่องกันไปหรือ Amplitude เปลี่ยนแปลงค่าไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฉนั้นสำหรับระบบที่ทราบว่าอินพุทของระบบจะเปลี่ยนแปลงตามเวลาอย่างไรและแน่ใจว่าไม่มี Disturbance แล้ว การควบคุมระบบนี้จะใช้การควบคุมแบบ Open – loop สำหรับการควบคุมแบบ Closed – loop จะใช้เมื่อไม่ทราบถึงการเปลี่ยนแปลงของอินพุทตามเวลา หรือในกรณีที่มี Disturbance หรือการเปลี่ยนแปลงค่าของอุปกรณ์ในระบบโดยไม่คาดคิดมาก่อน ในบางกรณีระบบใด ๆ อาจจะมีการควบคุมแบบ Open – loop และการควบคุมแบบ Closed – loop ร่วมกันก็ได้เพื่อทำให้ระบบทั้งหมดมีสมรรถนะตามต้องการ

หลักการออกแบบระบบควบคุม

สิ่งที่ต้องการเป็นอันดับแรกสำหรับระบบควบคุมก็คือระบบนั้น ๆ จะต้องเป็นระบบที่เสถียร ซึ่งจะต้องพิจารณาถึงเสถียรภาพสัมบูรณ์ (Absolute stability) และเสถียรภาพสัมพัทธ์ (Relative stability) ด้วย ระบบที่มีเสถียรภาพสัมพัทธ์ที่ดีนั้นความเร็วของผลตอบสนองของระบบจะต้องเร็วพอสมควร และการหน่วงก็ควรมีค่าไม่น้อยจนเกินไป นอกจากนี้แล้วระบบยังจะต้องสามารถลดค่าความคลาดเคลื่อนของระบบให้มีค่าเป็นศูนย์หรือมีค่าน้อยที่สุดที่ยอมรับได้ แต่อย่างไรก็ตามความต้องการทั้งสองกรณีนี้คือต้องการให้เสถียรภาพสัมพัทธ์ดีและมีความเที่ยงตรงสูงที่สภาวะคงที่นั้นเป็นสิ่งที่มักจะขัดแย้งกันเสมอ เช่นถ้าออกแบบให้ระบบมีเสถียรภาพสัมพัทธ์ที่ดี ค่าความเที่ยงตรงที่สภาวะคงที่ก็อาจจะลดน้อยลง เป็นต้น ดังนั้นจึงจำเป็นจะต้องหาวิธีการที่จะประณีประนอมความต้องการทั้งสองกรณีนี้ให้ยอมรับได้

ในการออกแบบระบบควบคุมใด ๆ นั้นมักจะมีวิธีการลองผิดลองถูกมาเกี่ยวข้องด้วยเสมอ สำหรับการ Systhesis ก็เป็นทฤษฎีของระบบควบคุมเชิงเส้นที่เป็นไปได้ และวิศวกรรมระบบควบคุมจะต้องสามารถหาอุปกรณ์ที่จำเป็นจะต้องใช้เพื่อให้ระบบทำงานได้ตามที่คำนวนไว้ แต่อย่างไรก็ตามในทางปฏิบัติแล้ว ระบบนั้นอาจมีเงื่อนไขข้อบังคับหลายอย่างหรืออาจเป็นระบบที่ไม่เป็นเชิงเส้น ซึ่งในแต่ละกรณีจะไม่สามารถใช้วิธีการ Systhesis ได้ และนอกจากนี้การวิเคราะห์หาคุณลักษณะของอุปกรณ์แต่ละชนิดอาจจะทำได้ไม่แม่นยำนัก ดังนั้นโดยทั่วไปจึงนิยมออกแบบระบบโดยใช้วิธีการลองผิดลองถูก

โดยทั่วไปแล้วขั้นตอนในการออกแบบจะเป็นดังนี้คือ ขั้นแรกวิศวกรจะต้องรู้ถึงข้อกำหนดหรือดัชนีแสดงสมรรถนะ คุณสมบัติไดนามิกของ Plant ที่กำหนด คุณสมบัติไดนามิกของอุปกรณ์การประยุกต์ใช้วิธีการ Systhesis ถ้าทำได้ ตลอดจนเทคนิคอื่น ๆ เพื่อสร้างรูปแบบทางคณิตศาสตร์ของระบบทางกายภาพนั้น ๆ ขึ้นมา เมื่อได้รูปแบบทางคณิตศาสตร์แล้ววิศวกรผู้ออกแบบก็สามารถจะหาคำตอบของระบบนั้น ๆ ได้โดยการหาคำตอบทางคณิตศาสตร์ วิธีที่นิยมใช้ก็คือการเลียนแบบ (Simulate) รูปแบบทางคณิตศาสตร์ของระบบนั้น ๆ ด้วยคอมพิวเตอร์เพื่อตรวจสอบหาพฤติกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของระบบต่อสัญญาณอินพุทหลาย ๆ แบบรวมทั้ง Disturbance ด้วย เมื่อการวิเคราะห์ระบบนี้เสร็จลงแล้วจะต้องเริ่มทำการออกแบบระบบใหม่เพื่อให้ระบบมีข้อกำหนดหรือสมรรถนะตามต้องการ จากนั้นนำระบบที่ออกแบบแล้วมาวิเคราะห์ใหม่ และทำเช่นนี้ไปเรื่อย ๆ จนกว่าจะได้ระบบที่ต้องการ

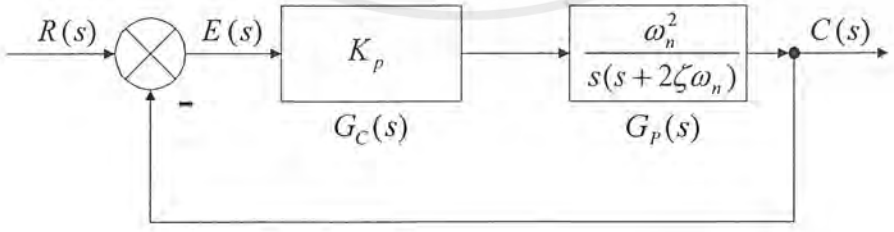
ขั้นตอนต่อไปก็คือการสร้างต้นแบบ (Prototype) ของระบบทางกายภาพขึ้นมาจากรูปแบบทางคณิตศาสตร์ของระบบที่ออกแบบได้ข้างต้น ซึ่งจะเห็นว่าขั้นตอนนี้จะตรงกันข้ามกับการหารูปแบบทางคณิตศาสตร์ในตอนแรก ต้นแบบของระบบทางกายภาพที่สร้างขึ้นมานี้จะใช้แสดงถึงรูปแบบทางคณิตศาสตร์ ซึ่งมีความเที่ยงตรงแม่นยำ อย่างไรก็ตามวิศวกรจะต้องทำการทดสอบการทำงานของระบบทางกายภาพด้วยว่าได้ผลเป็นที่น่าพอใจหรือไม่ ถ้าผลที่ได้เป็นที่น่าพอใจ การออกแบบก็เสร็จสมบูรณ์ แต่ถ้าผลที่ได้ยังไม่ดีพอก็จะต้องทำการแก้ไขต้นฉบับใหม่และทำการทดลองอีกจนกว่าจะได้ต้นแบบที่ให้ผลเป็นที่น่าพอใจ

ชนิดของตัวควบคุม

ต่อไปจะกล่าวถึงชนิดของตัวควบคุมซึ่งมีอยู่หลายแบบ แต่ละแบบมีคุณสมบัติที่แตกต่างกันไป ในการออกแบบจึงต้องเลือกตัวควบคุมที่เหมาะสมจึงจะได้ระบบที่มีเสถียรภาพ

ตัวควบคุมแบบ Proportional Controller (P)

ตัวควบคุมแบบสัดส่วนนี้เป็นชนิดที่ง่ายที่สุดของตัวควบคุม โดยการควบคุมจะใช้ Gain เป็นพารามิเตอร์ในการควบคุม เหมาะกับ Process ที่ไม่ซับซ้อน การปรับพารามิเตอร์ก็ไม่ยุ่งยากเพราะปรับแค่ Gain เพียงอย่างเดียว



รูปที่ 2-3 แสดงถึงระบบควบคุมป้อนกลับแบบ Proportional

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชันถ่ายโอนของตัวควบคุม คือ

$$Gc(s) = Kp$$

ฟังก์ชันถ่ายโอนแบบลูปเปิดของระบบที่ถูกควบคุมโดยรวม คือ

$$\frac{C(s)}{E(s)} = Gc(s)Gp(s) = \frac{\omega_n^2 Kp}{s(s+2\xi\omega_n)}$$

คุณสมบัติของ Proportional Action

- เมื่อลด Kp ลงไปจะทำให้อัตราการขยายสูงขึ้นการควบคุมจะมีความไวขึ้น มีผลทำให้โปรแกรมเกิดการแกว่งขึ้นได้
- เมื่อเพิ่มค่า Kp อัตราขยาย จะลดลง มีผลทำให้ค่าที่วัดได้กับค่าเป้าหมายแตกต่างกันมากขึ้น เราเรียกว่าเกิด Offset
- เมื่อโหลดของ Process มีสภาพแวดล้อมเปลี่ยนไปจากเดิม ซึ่งเราจะมีเรียกว่ามี Disturbance ก็จะเกิด Offset ได้ง่าย

Offset

Offset เป็นชื่อที่ใช้เรียกปรากฏการณ์ที่ตัวแปรโปรเซสหรือค่าวัดมีค่าไม่เท่ากับ ค่าเป้าหมาย ทำให้การควบคุมไม่เป็นไปตามที่ต้องการ แสดงความหมายของ Offset ในระบบการควบคุมแบบป้อนกลับนี้ เมื่อดูอย่างผิวเผิน Offset ไม่น่าจะเกิดขึ้นได้เพราะตัวแปรโปรเซสจะถูกป้อนกลับมาเปรียบเทียบกับค่าเป้าหมายตลอดเวลา อย่างไรก็ตาม Offset มักจะเกิดขึ้นในระบบควบคุมที่มีการใช้ Proportional Control ที่มีค่า Kp ใหญ่ และ ขณะที่เกิด Disturbance ใน Process ซึ่งได้แก่การเปลี่ยนแปลงของโหลดสภาพแวดล้อมเป็นต้น

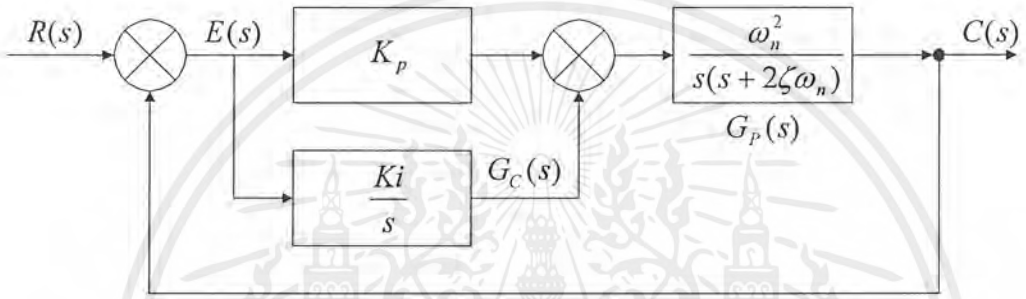
วิธีการแก้ Offset

- ทำให้การลดค่า Kp ให้เล็กลง
- เปลี่ยน BIAS ของการควบคุมด้วยมือ
- เปลี่ยนค่าเป้าหมายใหม่
- ใช้การควบคุมที่มี Reset Action (PI Controller)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวควบคุมแบบ Proportional Integral Controller (PI)

ตัวควบคุมอีกแบบหนึ่งที่นิยมใช้ร่วมกับตัวควบคุมแบบสัดส่วนคือตัวควบคุมแบบอินทริกัล ซึ่งจะสร้างสัญญาณที่มีค่าเท่ากับการอินทริกัลของค่าความคลาดเคลื่อน รูปที่ 2-4 แสดงถึงรูปแบบของการควบคุมแบบอินทริกัลกับระบบอันดับสอง สัญญาณที่ป้อนให้กับระบบจะแบ่งเป็นสองส่วนคือ สัญญาณที่เป็นสัดส่วนกับค่าความคลาดเคลื่อนและสัญญาณที่เป็นค่าอินทริกัลตามเวลาของค่าความคลาดเคลื่อนโดยที่ Kp เป็นค่าคงที่



รูปที่ 2-4 แสดงถึงระบบควบคุมป้อนกลับแบบ PI

ฟังก์ชันถ่ายโอนของตัวควบคุม คือ

$$Gc(s) = Kp + \frac{Ki}{s}$$

ฟังก์ชันถ่ายโอนแบบลูปเปิดของระบบที่ถูกควบคุมโดยรวม คือ

$$\frac{C(s)}{E(s)} = Gc(s)Gp(s) = \frac{\omega_n^2(Kp\,s + Ki)}{s^2(s + 2\xi\omega_n)}$$

ผลของการควบคุมแบบอินทริกรัลก็คือเครื่องอันดับของระบบขึ้นอีกหนึ่งอันดับ หรือเพิ่มชนิดของระบบ (type of system) ขึ้นด้วยหนึ่ง นั่นก็คือค่าความคลาดเคลื่อนที่ภาวะคงทีของระบบเดิมที่ไม่สามารถควบคุมแบบอินทริกัลจะถูกปรับปรุงให้ดีขึ้นด้วยอันดับที่เพิ่มมานี้ หรืออีกนัยหนึ่งถ้าค่าความคลาดเคลื่อนที่ภาวะคงที่ของระบบต่ออินพุทที่กำหนดให้มีค่าคงที่ เมื่อเพิ่มตัวควบคุมแบบอินทริ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กัลเข้าไปจะทำให้มัความคลาดเคลื่อนนี้มีค่าเป็นศูนย์ เช่นระบบอันดับสองในรูปที่ 2-4 นั้น ถ้าไม่มีตัวควบคุมแบบอินทริกัลรวมอยู่ด้วยจะมีค่าความคลาดเคลื่อนที่ภาวะคงที่ต่ออินพุทแบบ Step เท่ากับศูนย์ แต่จะมีค่าความคลาดเคลื่อนที่ภาวะคงที่ต่ออินพุทแบบ Ramp ถ้าระบบอันดับสองมีตัวควบคุมแบบอินทริกัลรวมอยู่ด้วย

ดังแสดงด้วย Open-loop transfer function ค่าความคลาดเคลื่อนที่ภาวะคงที่ต่ออินพุท Step และอินพุทแบบ Ramp จะมีค่าเท่ากับศูนย์ ซึ่งแสดงให้เห็นถึงว่าการควบคุมแบบอินทริกัลสามารถทำให้ค่าความคลาดเคลื่อนที่ภาวะคงที่ลดลงได้นั่นเอง

อย่างไรก็ตาม เนื่องจากตัวควบคุมแบบอินทริกัลนั้นไปทำให้อันดับของระบบเดิมเพิ่มขึ้นดังนั้นระบบจะมีเสถียรภาพน้อยลงเมื่อเทียบกับระบบเดิมหรืออาจจะไม่เสถียรเลยก็ได้

กริยาการควบคุมแบบ Integral นี้บางครั้งเรียกว่า Reset Control การควบคุมแบบนี้สัญญาณควบคุมจะแปรค่าตามอินทิกัลต่อเวลาของผลต่าง เมื่อผลต่างเปลี่ยนเป็นแบบ Step สัญญาณควบคุมจะค่อยๆ เพิ่มขึ้นตามค่าของเวลา ตามความหมายของ Integral Time คือ เวลาที่มีการใช้ไปในการเพิ่มค่าของสัญญาณควบคุมจนมีขนาดเท่ากับผลต่าง e(t) ที่เปลี่ยนไป ดังนั้นเมื่อ Ki มีค่าน้อยผลของ Integral Action จะมากกว่าเมื่อค่าของ Ki มีค่ามาก

คุณสมบัติของ Integral Action

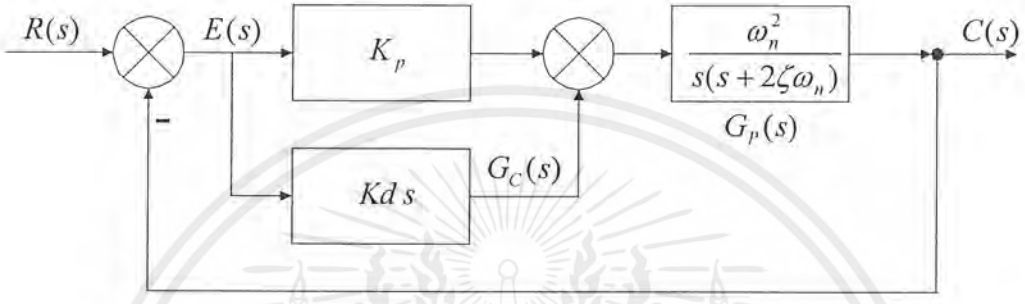
- Offset ที่เกิดขึ้นจาก P Action จะถูก Integral Action แก้ไขหมด
- เมื่อ Ki มีค่าน้อย ผลของ Integral Action จะมากทำให้เกิด การแกว่งได้ซึ่งจะมีผลให้ระบบขาดเสถียรภาพ

ตัวควบคุมแบบ Proportional Derivative Controller (PD)

การควบคุมแบบนี้จะเหมาะกับกระบวนการที่มี Time lag มาก ๆ เพราะสามารถที่จะแก้ข้อผิดพลาดโดยการกระทำล่วงหน้าก่อนที่จะมีคว ามผิดพลาดเกิดขึ้นทั้งนี้เพราะสัญญาณควบคุมจะแปรตามอัตราการเปลี่ยนแปลงของ สัญญาณ Error และในบางครั้งจะเรียกกริยาควบคุมแบบนี้ว่า Rate action การควบคุมแบบ Derivative Action นี้ไม่สามารถที่จะนำไปใช้ตามลำพังได้เพราะว่า Error เป็น 0 หรือมีค่าคงที่ ค่าควบคุมจะไม่ให้ค่าเอาท์พุทออกมาเลย ดังนั้น Derivative Action จึงใช้ร่วมกับการควบคุมแบบ Proportional ซึ่งตัวควบคุมแบบ Proportional จะมีข้อจำกัดซึ่งเป็นข้อเสียคือในบางครั้งจำเป็นจะต้องมีการประนีประนอม (Compromise) ในการเลือกค่า Forward gain เพื่อที่จะทำให้ค่าของ Maximum overshoot และค่าความคลาดเคลื่อนที่ภาวะอยู่ในขอบเขตที่ยอมรับได้ ซึ่งในทางปฏิบัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แล้วทำได้ยากเพราะเมื่อเลือก Gain มากเพื่อที่จะให้ค่าความคลาดเคลื่อนที่ภาวะคงที่น้อย ค่า Maximum overshoot จะมากและระบบอาจจะไม่เสถียรได้ถ้าระบบนั้นเป็นระบบอันดับสูง ด้วยเหตุนี้จึงจำเป็นต้องหาการควบคุมแบบอื่นมาร่วมเพื่อที่จะทำให้ผลตอบสนองชั่วครู่และผลตอบสนองภาวะคงที่ของระบบดีตามที่ต้องการ



รูปที่ 2-5 แสดงถึงระบบควบคุมป้อนกลับแบบ PD

ฟังก์ชันถ่ายโอนของตัวควบคุม คือ

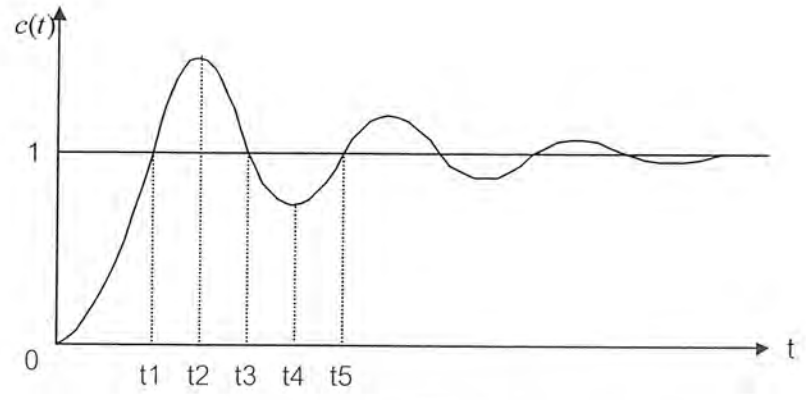
$$Gc(s) = Kp + Kd\,s$$

ฟังก์ชันถ่ายโอนแบบลูปเปิดของระบบที่ถูกควบคุมโดยรวมคือ

$$\frac{C(s)}{R(s)} = Gc(s)Gp(s) = \frac{\omega_n^2 (Kp + Kd\,s)}{s\ (s + 2\xi\omega_n)}$$

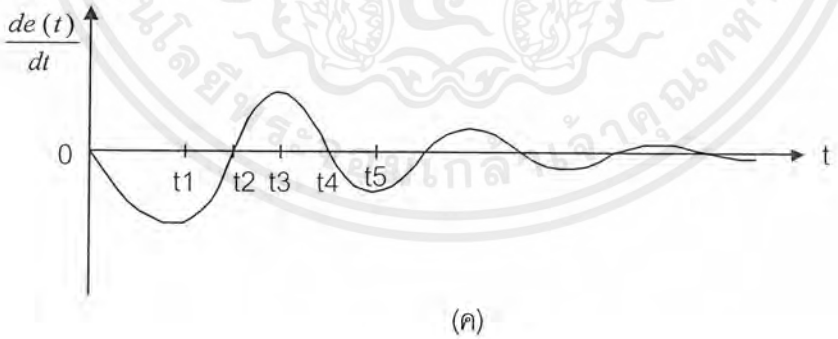
จากสมการข้างบน จะเห็นว่าการควบคุมแบบ PD จะเหมือนกับการเพิ่ม Zero ที่ $s = \frac{-1}{Td}$ ให้กับ Open-loop transfer function ซึ่งทำให้อัตราหน่วงมีค่ามากขึ้น และส่งผลให้ Maximum Overshoot มีค่าลดลง หรือหมดไปเลย อย่างไรก็ตามค่า Kd จะต้องมีค่าที่เหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก)

(ข)



(ค)

รูปที่ 2-6 Waveforms ของ c(t), e(t), และ de(t) / dt
แสดงถึงผลของการควบคุมแบบดีริเวทีฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลของการควบคุมแบบดีริเวทีฟที่มีต่อผลตอบสนองชั่วครู่ของระบบควบคุมสามารถจะพิจารณา ได้จากรูปที่ 2-6 โดยที่รูป (ก) เป็นผลตอบสนองของระบบอันดับสองต่ออินพุทที่เป็น Unit step ซึ่งไม่มีมีตัวควบคุม ดีริเวอร์ทีฟ (Derivative controller) รวมอยู่ รูป (ข) และรูป (ค) เป็นสัญญาณความผิดพลาดระหว่างอินพุทและเอาท์พุท e(t) และค่าอนุพันธ์อันดับหนึ่งของ e(t) เทียบกับเวลาตามอันดับ จากรูป ในช่วงเวลา $0 < t < t1$ นั้นค่าความคลาดเคลื่อนจะเป็นบวกที่มีค่ามากซึ่งจะไปทำให้ผลตอบสนองของระบบเกิด Over shoot ขึ้นมาที่เวลา t2 สำหรับช่วงเวลา $t1 < t < t3$ นั้นค่าความคลาดเคลื่อนจะมีค่าเป็นลบและค่าลบนี้จะเป็นแรงต้านให้เอาท์พุทของระบบลดลง
นั่นคือทำให้ c(t) เกิด Undershoot ในช่วงเวลา $t3 <t<t5$ และในช่วงของเวลา $t3<t<t5$ นี้ค่าความคลาดเคลื่อนจะกลับมีค่าเป็นบวกอีกซึ่งจะส่งผลทำให้ undershoot ของผลตอบสนองลดลง เนื่องจากเราถือว่าระบบนี้เสถียร ดังนั้นค่าความคลาดเคลื่อนจะลดลงทุกช่วงของการ Oscillate เอาท์พุทของระบบจึงสามารถเข้าสู่ค่าสุดท้ายที่ต้องการได้

เพื่อเป็นการลดค่า Overshoot ลง เราจะเพิ่มตัวควบคุมแบบดีริเวทีฟลงในระบบควบคุมดังในรูปที่ 2 ดังนั้นค่าความคลาดเคลื่อนด้ายค่าของสัญญาณที่เป็นสัดส่วนกับอัตราการเปลี่ยนแปลงต่อเวลาของค่าความคลาดเคลื่อน ค่าของสัญญาณที่เพิ่มเข้าไปนี้แสดงดังรูปที่ 3 (ค) ในช่วง $o<t<t1$ ค่าของสัญญาณde(t)/dt จะมีค่าเป็นลบซึ่งจะไปทำให้ค่าความคลาดเคลื่อน e (t) เดิมลดลง ในช่วง $t1<t<t2$ นั้นทั้ง e(t) และ de(t) /dt เป็นลบ ดังนั้นจะเห็นว่าแรงต้านที่เป็นค่าลบนี้จะมากกว่ากรณีที่ไม่มีตัวควบคุมดีริเวทีฟ นั่นก็คือทำให้ Overshoot มีขนาดเล็กลง สำหรับในช่วง $t2<t<t3$ นั้น e(t) และ de (t) จะมีเครื่องหมายตรงกันข้าม และก็ทำให้ Undershoot ลดลงเช่นกัน

การควบคุมแบบดิริเวทีฟจะไม่มีอิทธิพลต่อค่าคลาดเคลื่อนที่ภาวะคงที่ถ้า ค่าคลาดเคลื่อนนี้เป็นค่าคงที่ นั่นคือถ้า e(t) เป็นค่าคงที่ de(t) /dt จะมีค่าเท่ากับศูนย์ ซึ่งแสดงว่าการควบคุมแบบดิริเวทีฟจะไม่มีอิทธิพลต่อค่าความคลาดเคลื่อนที่ภาวะคงที่ แต่ถ้าค่าความคลาดเคลื่อนที่ภาวะคงที่ของระบบแปรตามเวลาแล้ว de(t)/dt จะไม่เท่ากับศูนย์แสดงว่าการควบคุมแบบดิริเวทีฟจะมีอิทธิพลต่อค่าความคลาดเคลื่อนที่ภาวะคงที่

ผลของการควบคุมแบบดิริเวทีฟที่มีต่อผลตอบสนองชั่วครู่ของระบบควบคุมป้อนกลับสามารถจะแสดงได้โดยพิจารณาถึง Open-loop transfer function,เพราะฉะนั้น Closed loop transfer function และสมการคุณลักษณะคือ

$$\frac{C(s)}{R(s)} = \frac{(1+\tau_d s)\omega_n^2}{s^2 + (2\zeta\omega_n + \tau_d\omega)s + \omega_n^2}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการข้างบน จะเห็นว่าการควบคุมแบบดีริเวทีฟไปทำให้สัมประสิทธ์ของ s เพิ่มขึ้นด้วยค่า $Td\,\omega_n^2$ นั่นก็คือทำให้อัตราการหน่วงของระบบเพิ่มขึ้นนั่นเอง ดังนั้นผลตอบสนองของระบบจะมี Overshoot ลดลงหรืออาจจะไม่มี Overshoot เลยก็ได้

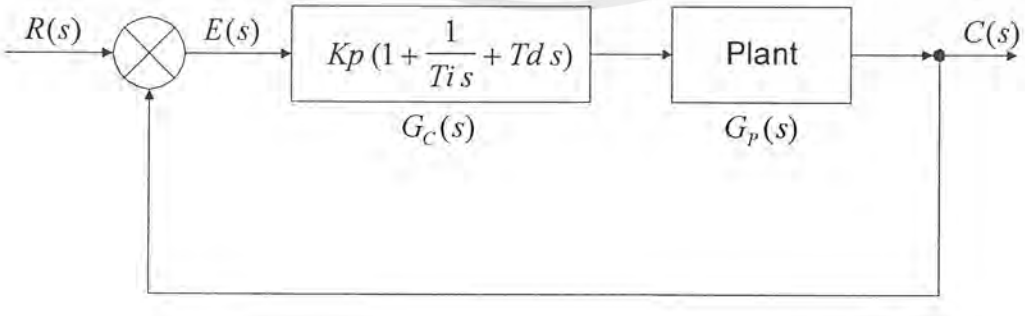
คุณสมบัติของ D Action

- เหมาะสำหรับ Process ที่มี Time lag มาก ๆ จะช่วยให้การควบคุมถึงจุดที่ต้องการเร็วขึ้น
- ถ้า Td มากไป ผลของ D Action จะมาก ทำให้ระบบทั้งระบบไว้ขึ้นขาดเสถียรภาพ
- ไม่เหมาะกับ Process ที่มี Time lag น้อย ๆ และตัวแปร Process เปลี่ยนแปลงได้ง่าย เช่น ระบบควบคุมการไหล ความดัน เป็นต้น

ตัวควบคุมแบบ Proportional Integral Derivative Controller (PID)

ดังที่ทราบแล้วว่า ตัวควบคุมแบบ PD จะทำให้อัตราการหน่วงมีค่ามากขึ้น และส่งผลให้ระบบมีเสถียรภาพสัมพัทธ์ดีขึ้น แต่ไม่มีผลโดยตรงต่อผลตอบสนองที่ภาวะคงที่ ส่วนตัวควบคุมแบบ PI นั้นจะทำให้อัตราการหน่วงมีค่าเพิ่มขึ้นพร้อม ๆ กับทำให้ค่าความคลาดเคลื่อนที่ภาวะคงที่ลดลงหรือหมดไป แต่ทำให้ Rise Time และ Settling Time มีค่ามากขึ้น ดังนั้น บางครั้งจึงจำเป็นต้องใช้ตัวควบคุมแบบ PID เพื่อทำให้ระบบที่ควบคุมมีสมรรถนะตามต้องการ

การควบคุมแบบ PID นั้นจะสามารถที่ทำได้จากการรวมเอากริยาควบคุมแบบ Proportional ซึ่งเป็นอัตราขยาย กริยาควบคุมแบบ Integral ซึ่งสามารถแก้ค่า Offset และ Overshoot ที่จะทำให้เกิดการ Oscillate และกริยาควบคุมแบบ Derivative จะทำให้เกิดผลตอบสนองได้รวดเร็ว



รูปที่ 2-7 แสดงถึงระบบควบคุมป้อนกลับแบบ PID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชันถ่ายโอนของตัวควบคุม แบบPID เป็นดังนี้

$$G_C(s) = Kp\ (1 + \frac{1}{T_i s} + T_d s)$$

โดยที่ Kp = Proprtinal Gain

Ti = Integral Time

Td = deruvative Time

การออกแบบตัวควบคุม PID จะพิจารณาได้ดังต่อไปนี้

- ฟังก์ชันถ่ายโอนของตัวควบคุม PID จะประกอบไปด้วยส่วนของตัวควบคุม PI และ PD
- เทอมของตัวควบคุมแบบ PI จะทำให้ค่าความคลาดเคลื่อนที่ภาวะคงที่ลดลงหรือเป็นศูนย์ ดังนั้น จะพิจารณาเลือกค่าของ Kp ที่ทำให้ Rise Time และ/หรือ Settling Time เป็นไปตามต้องการ โดยจะไม่คำนึงถึง Maximum Overshoot ว่าจะมากหรือน้อยเพียงใด (ส่วนใหญ่จะมาก)
- เทอมของตัวควบคุมแบบ PD จะพิจารณาเลือกค่าของ Kd ที่จะทำให้อัตราการหน่วง หรือ Maximum Overshoot เป็นไปตามต้องการ

วิธีอื่นที่ใช้ได้เช่นกัน คือ เริ่มออกแบบส่วนของตัวควบคุมแบบ PD ก่อนเพื่อหาค่าเหมาะสม ถ้าตัวควบคุมแบบ PD ทำให้ระบบที่ถูกควบคุมมีสมรรถนะตามต้องการ การออกแบบก็เสร็จสิ้น ถ้าตัวควบคุมแบบ PD ยังให้ผลไม่เป็นที่น่าพอใจ จึงเพิ่มตัวควบคุมแบบ PI เข้ามา

การปรับค่าของตัวควบคุม PID

การปรับค่าพารามิเตอร์ของตัวควบคุม PID เป็นสิ่งที่จำเป็นเพื่อให้ได้การควบคุมที่ดีที่สุด ซึ่งทำได้หลายวิธี บางวิธีจะพิจารณาจากผลตอบสนองของระบบและอาศัยประสบการณ์ บางวิธีจะอาศัยคณิตศาสตร์ และส่วนใหญ่จะพิจารณาจากเงื่อนไขในโดเมนเวลามากกว่าในโดเมนความถี่

การปรับค่าของตัวควบคุม PID โดยวิธีของ Ziegler - Nichols

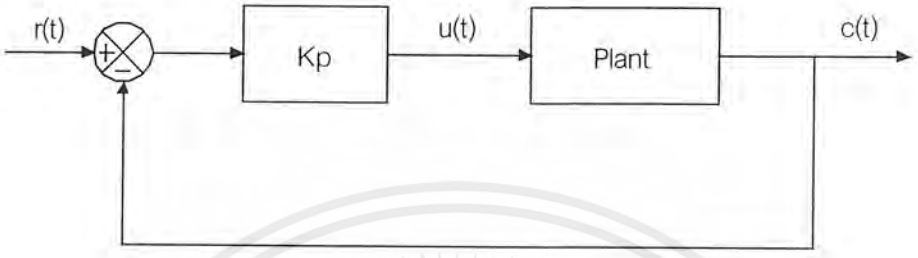
วิธี Ultimate Method

วิธีนี้จะหาค่าพารามิเตอร์ของตัวควบคุม PID จากผลตอบสนองเวลาของระบบหรือกระบวนการที่ถูกควบคุมด้วยตัวควบคุมแบบ P ต่ออินพุทแบบ Unit Step โดยปรับค่าของ Kp ไปเรื่อย ๆ จนผลตอบสนองเวลาเกิดการแกว่งอย่างต่อเนื่อง (Sustained Oscillations) ดังในรูป (ถ้าผลตอบสนองเวลาไม่เกิดการแกว่งอย่างต่อเนื่อง วิธีนี้จะใช้ไม่ได้) จากนั้นหาค่าของ

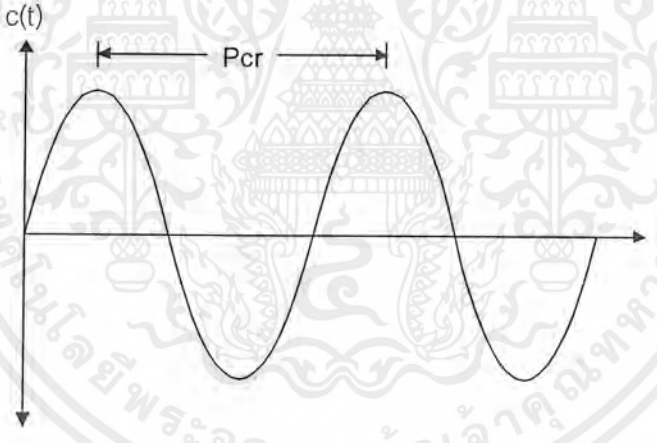
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Kcr (Critical Gain) เป็นอัตราการขยายที่ทำให้ผลตอบสนองเวลาเกิดการแกว่งอย่างต่อเนื่อง

Pcr (Oscillation Period) เป็นคาบเวลาของการแกว่งอย่างต่อเนื่อง



รูปที่ 2-8 ระบบควบคุมแบบ Proportionnal



รูปที่ 2-9 แสดงการ Oscillation เพื่อหาค่า Pcr

Ziegler-Nichols ได้กำหนดค่าของ **Kp** , **Ti** และ **Td** สำหรับตัวควบคุมแบบต่าง ๆ ดังนี้

ตัวควบคุมแบบ P

$$Kp = 0.5\ Kcr$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวควบคุมแบบ PI

$$K_p = 0.45 K_{cr}$$

$$T_i = P_{cr}/1.2$$

ตัวควบคุมแบบ PD

$$K_p = 0.6 K_{cr}$$

$$T_d = 0.125 P_{cr}$$

ตัวควบคุมแบบ PID

$$K_p = 0.6 K_{cr}$$

$$T_i = 0.5 P_{cr}$$

$$T_d = 0.125 P_{cr}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

State - space

State - space เป็นเทคนิคที่ช่วยในการแก้สมการ nth – order differential โดยเราสามารถนำ Transfer function มาคำนวนหา Output ได้

Transfer function จะมีรูปแบบดังนี้

$$\frac{Y(s)}{U(s)} = \frac{b_0 s^n + b_1 s^{n-1} + \cdots + b_{n-1} s + b_n}{s^n + a_1 s^{n-1} + \cdots + a_{n-1} s + a_n}$$

เช่น

$$\frac{Y(s)}{U(s)} = \frac{160s + 640}{s^3 + 18s^2 + 192s + 640}$$

State space matrix

รูปแบบของ State – space matrix เป็นดังนี้

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \cdot \\ \cdot \\ \cdot \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ 0 & 0 & 0 & \cdots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \cdots & -a_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} \beta_1 \\ \beta_2 \\ \cdot \\ \cdot \\ \cdot \\ \beta_{n-1} \\ \beta_n \end{bmatrix} u$$

โดยมีค่าพารามิเตอร์ดังนี้

$$\beta_0 = b_0$$
$$\beta_1 = b_1 - a_1 \beta_0$$
$$\beta_2 = b_2 - a_1 \beta_1 - a_2 \beta_0$$
$$\beta_3 = b_3 - a_1 \beta_2 - a_2 \beta_1 - a_3 \beta_0$$
$$\cdot$$
$$\cdot$$
$$\cdot$$
$$\beta_n = b_n - a_1 \beta_{n-1} - \cdots - a_{n-1} \beta_1 - a_n \beta_0$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

34034

เราสามารถแทนค่าพารามิเตอร์ต่างๆลงใน State – space matrix แล้วคำนวณหาค่า Output ได้จาก

$$y = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} + \beta_0 u$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทฤษฎีฟัซซี่ลอจิก

ระบบควบคุมฟัซซี่ (Fuzzy Control) มีพื้นฐานแนวความคิดมากจากทฤษฎีฟัซซี่ลอจิก (Fuzzy Logic) และ ทฤษฎีฟัซซี่เซต (Fuzzy Set) ฟัซซี่ลอจิกมีพื้นฐานอยู่บนทฤษฎีฟัซซี่เซตซึ่งจะช่วยให้สามารถอธิบายการปฏิบัติการ และการควบคุมของระบบเป็นคำพูดได้ชัดเจนขึ้น หลักสำคัญของทฤษฎีฟัซซี่เซต คือ ยอมรับสมาชิกที่มีลักษณะตามเซตเพียงบางส่วนเข้ามาเป็นสมาชิก ซึ่งแตกต่างจากทฤษฎีเซตดั้งเดิม (Crisp set) ทฤษฎีเซตดั้งเดิม จะเน้นชัดเจนเลยว่าเป็นสมาชิกของเซตหรือไม่เท่านั้น ไม่มีการเป็นสมาชกของเซตเพียงบางส่วน ต่อไปเราจะกล่าวถึงรายละเอียดของทฤษฎีข้างต้น และการนำทฤษฎีดังกล่าวมาประยุกต์ใช้ในการควบคุมระบบตามลำดับ

คริชพเซต (Crisp set)

ทฤษฎีคริชพเซต เป็นทฤษฎีที่เราคุ้นเคยกันดีโดยหลักการพื้นฐานของคริชพเซต มีดังนี้

1. การเป็นสมาชิกของเซต

" ถ้า X เป็นสมาชิกของเซต A " เราจะใช้สัญลักษณ์

$$X \in A$$

" ถ้า X ไม่เป็นสมาชิกของเซต A " เราจะใช้สัญลักษณ์

$$X \notin A$$

ซึ่งในคริชพเซต นี้เราจะระบุชัดเจนว่าสมาชิกของ Universe เป็นสมาชิกของเซต A หรือไม่ เราสามารถแทนค่าระดับการเป็นสมาชิก (Membership function) ซึ่งแทนด้วยสัญลักษณ์ μ_A ดังนี้

$$\mu_A(X) = 1 \text{ ก็ต่อเมื่อ } X \in A$$

$$\mu_A(X) = 0 \text{ ก็ต่อเมื่อ } X \notin A$$

2. การเป็นสับเซต

" ถ้าสมาชิกทุกตัวของเซต A เป็นสมาชิกของเซต B จะเรียกว่า เซต A เป็นสับเซตของเซต B " เราใช้สัญลักษณ์การเป็นสับเซตดังนี้

$$A \subseteq B$$

3. การเท่ากันของเซต

" ถ้าสมาชิกทุกตัวของเซต A เท่ากับสมาชิกทุกตัวของเซต B จะเรียกว่า เซต A เท่ากับเซต B " เราใช้สัญลักษณ์การเท่ากันดังนี้

$$A = B$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การไม่เท่ากันของเซต

" ถ้าสมาชิกทุกตัวของเซต A ไม่เท่ากับสมาชิกทุกตัวของเซต B จะเรียกว่า เซต A ไม่เท่ากับเซต B" เราใช้สัญลักษณ์การไม่เท่ากันเซตดังนี้

$$A \neq B$$

5. การเป็นสับเซตแท้

" ถ้าสมาชิกทุกตัวของเซต A เป็นสมาชิกของเซต B และ เซต A ไม่เท่ากับเซต B จะเรียกว่า เซต A เป็นสับเซตแท้ของเซต B "

เราใช้สัญลักษณ์การเป็นสับเซตแท้ ดังนี้

$$A \subset B$$

6. การปฏิบัติการเบื้องต้นของเซต ได้แก่

- Complement เป็นการสร้างเซตใหม่โดยสมาชิกของเซตใหม่จะไม่เป็นสมาชิกของเซต A เลย

$$\text{Complement } A = A'$$

- Union เป็นการสร้างเซตใหม่โดยสมาชิกของเซตใหม่จะมีสมาชิกของทุกเซตรวมอยู่
- Intersection เป็นการสร้างเซตใหม่โดยสมาชิกของเซตใหม่จะมีสมาชิกซึ่งเป็นสมาชิกของทุกเซต

ฟัซซี่เซต (Fuzzy set)

ทฤษฎีฟัซซี่เซต เป็นการรวมสมาชิกของหลาย ๆ ดริชพเซตที่มีอยู่จริงและพอจะมีลักษณะเข้ารวมกลุ่มได้ ฟัซซี่เซต ยอมรับการเป็นสมาชิกของสมาชิกที่มีลักษณะตามเซตเพียงบางส่วนซึ่งจะมีการเปลี่ยนแปลงทีละน้อยระหว่างการมีคุณสมบัติของการเป็นสมาชิกอย่างครบถ้วน กับไม่มีคุณสมบัติของการเป็นสมาชิกเลย แม้ว่าจะไม่พร้อมกันก็ตาม จึงสามารถใช้ฟัซซี่เซต ไปทำงานกับระบบดริชพเซตได้

1. การเป็นสมาชิกของฟัซซี่เซต

ในระบบดริชพเซตจะกำหนดเพียงว่า X เป็นสมาชิกของเซต A โดยแทนด้วยฟังก์ชันระดับการเป็นสมาชิก เป็น 0 หรือ 1 แต่ฟัซซี่เซตจะยอมรับสมาชิกที่มีลักษณะที่ถูกต้องเพียงบางส่วนและผิดเพียงบางส่วนไม่มีขอบเขตแน่นอน การประยุกต์ใช้ทฤษฎีฟัซซี่เซต จะต้องแสดงค่าระดับ (Degree) ซึ่งเป็นค่าที่เป็นไปได้ที่สมาชิกของเซตหรือฟังก์ชันระดับการเป็นสมาชิก โดยเราใช้สัญลักษณ์ μ แทนค่าระดับความเป็นสมาชิกซึ่งมีค่าระหว่าง 0 ถึง 1 แสดงดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\mu_A(X) \rightarrow [0,1]$$

หมายความว่า ระดับการเป็นสมาชิก (Grade of Membership หรือ Degree of membership) ของ X ในฟัซซี่เซต A อยู่ในช่วงทั้งหมดจาก 0 ถึง 1 เมื่อประยุกต์เข้ากับฟัซซี่ลอจิกค่า μ จะถูกเรียกว่า ค่าความจริงที่แสดงค่าระดับของเซต คือ

$$0 \leq X \leq 1$$

โดยถ้าค่าสูงแสดงว่ามีความเป็นสมาชิกมาก ถ้าค่าต่ำแสดงว่ามีความเป็นสมาชิกน้อย ค่า 0 จะหมายถึงไม่เป็นสมาชิกเลยและ 1 คือ เป็นสมาชิกอย่างสมบูรณ์

ขอเน้นว่าฟังก์ชันระดับการเป็นสมาชิกของฟัซซี่เซต ถึงแม้จะมีค่าระหว่าง 0 ถึง 1 แต่อย่านำไปสับสนกับความน่าจะเป็น (Probability) ฟัซซี่เซต เป็นรูปแบบหนึ่งของความไม่แน่นอน (Uncertainly) และโดยธรรมชาติฟัซซี่เซต เป็นศาสตร์ที่ไม่มีความเกี่ยวข้องกับสถิติ

2. นิยามที่ฟัซซี่เซต ขยายมาจากคริชพเซต

- ฟัซซี่เซต จะว่างก็ต่อเมื่อ ฟังก์ชันระดับการเป็นสมาชิกของฟัซซี่เซต นั้นเป็น 0 ตลอดทั้ง X
- ฟัซซี่เซต A และ B จะเท่ากัน $\mu_A(X) = \mu_B(X)$ สำหรับทุก ๆ X น X เขียนแทนได้ด้วย A = B
- Subset หรือ Containment นิยามโดย

$$A \subset B \leftrightarrow \mu_A(X) \leq \mu_B(X)$$

- Complement ของฟัซซี่เซต A เขียนแทนด้วย A' นิยามโดย

$$\mu_{A'}(X) = 1-\mu_A(X)$$

- Union ของฟัซซี่เซต A และ B ซึ่งมีฟังก์ชันระดับการเป็นสมาชิก $\mu_A(X)$ และ $\mu_B(X)$ ตามลำดับ สามารถแทนด้วย ฟัซซี่เซต C โดยฟังก์ชันระดับการเป็นสมาชิก ของฟัซซี่เซต C กำหนดโดย

$$\mu_C(X) = \text{MAX}[\mu_A(X), \mu_B(X)] \text{ หรืออาจเขียนย่อเป็น}$$

$$\mu_C = \mu_A \vee \mu_B(X)$$

- Intersection ของฟัซซี่เซต A และ B ซึ่งมีฟังก์ชันระดับการเป็นสมาชิก $\mu_A(X)$ และ $\mu_B(X)$ ตามลำดับสามารถแทนด้วยฟัซซี่เซต C โดยฟังก์ชันระดับการเป็นสมาชิกของฟัซซี่เซต C กำหนดโดย

$$\mu_C(X) = \text{MIN}[\mu_A(X), \mu_B(X)] \text{ หรืออาจเขียนย่อเป็น}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\mu_C = \mu_A \wedge \mu_B(X)$$

อย่างไรก็ตามไม่มีหลักตายตัวว่า การ Union ต้องใช้ Max Operation การ Intersection ต้องใช้ Min operation หรือการ Complement ต้องใช้ $\mu_{A'}(X) = 1 - \mu_B(X)$ ดังนั้นเพื่อให้ General มากขึ้นจึงมีนิยามสัจพจน์ (Axiom) ปฏิบัติการแต่ละชนิดดังนี้

- Complement C: [0,1] $\rightarrow$ [1,0]

  Axiom C1: C(0) = 1 และ C(1) = 0 หมายความว่าสามารถครอบคลุมคริชพเซต

  Axiom C2: สำหรับทุก ๆ a, b $\in$ [0,1] ถ้า a $<$ b แล้ว C (a) $\geq$ C(b) หมายความว่า C เป็น Monotonic nonincreasing

ทุก ๆ การปฏิบัติการที่จะถือว่าเป็นการ Complement ได้อย่างน้อยที่สุดจะต้องสอดคล้องกับสัจพจน์ทั้งสอง

- Union U: [0,1] $\times$ [0,1] $\rightarrow$ [0,1]

  Axiom U1:U(0,0) = 0 ; U(0,1) = U(1,0) = U(1,1) = 1 นั่นหมายความว่า จะต้องสามารถครอบคลุมคริชพเซต

  Axiom U2 :U(a,b) = U(b,a); มีคุณสมบัติการสลับที่

  Axiom U3 :ถ้า a $\leq$ a' และ b $\leq$ b' แล้ว U(a,b) $\leq$ U(a' ,b') หมายความว่า U เป็น Monotonic

  Axiom U4 :U(U(a,b),c) = U(a,U(b,c)) ; มีคุณสมบัติการจัดหมู่

ทุก ๆ การปฏิบัติการที่จะถือว่าเป็นการ Union ได้อย่างน้อยจะต้องสอดคล้องกับสัจพจน์ทั้งสี่

- Intersection I : [0,1] $\times$ [0,1] $\rightarrow$ [0,1]

  Axiom I1 :I(1,1) = 1 ; I(0,1) = I(1,0) = I(0,0) = 0 นั่นหมายความว่า จะต้องสามารถครอบคลุมคริชพเซต

  Axiom I2 : I(a,b) = I(b,a) ; มีคุณสมบัติการสลับที่

  Axiom I3 :ถ้า a $\leq$ a' และ b $\leq$ b' แล้ว I(a,b) $\leq$ I(a' ,b') หมายความว่า I เป็นMonotonic

  Axiom I4 : I (I(a,b),c) = I(a,I(b,c)) ; มีคุณสมบัติการจัดหมู่

ทุก ๆ การปฏิบัติการที่จะถือว่าเป็นการ Intersection ได้อย่างน้อยที่สุดจะต้องสอดคล้องกับสัจพจน์ทั้งสี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟัซซี่ลอจิก (Fuzzy logic)

ฟัซซี่ลอจิกจะมีการกระทำทางลอจิกเพื่อรวมค่าลอจิกต่าง ๆ ให้เป็นค่าฟัซซี่ลอจิกคล้ายตัวแปรในระบบลอจิก 2 ระดับ โดยจะต้องมีกำหนดลักษณะความหมายให้แตกต่างกันแต่ใช้คำสั่งทางลอจิกที่เหมือนกันคือ AND, OR และ NOT ความหมายต่าง ๆ ในระบบฟัซซี่ (Fuzzy system)ถูกกำหนดโดย L.A. Zadeh ผู้คิดค้นระบบฟัซซี่ลอจิก

การ AND ของฟัซซี่ลอจิก

ตามคำจำกัดความของ Zedeh คือค่าความจริงที่น้อยที่สุด (Minimum) นั้นคือสำหรับค่าฟัซซี่ A และ B

$$\mu (A \text{ AND } B) = \min (\mu A, \mu B)$$

การ OR ของฟัซซี่ลอจิก

ตามคำจำกัดความของ Zadeh คือค่าความจริงที่มากที่สุด (Maximum)

$$\mu (A \text{ OR } B) = \max (\mu A, \mu B)$$

การ NOT ของค่าฟัซซี่ลอจิก

$$\mu (\text{ NOT } A) = 1 - \mu A$$

ซึ่งการกระทำทั้ง 3 นี้ เป็นสมดุลของกระทำในลอจิก 2 ระดับ สำหรับค่า μ มีค่าอยู่ระหว่าง 0 ถึง 1

การประยุกต์ใช้งานของฟัซซี่ลอจิก (Fuzzy logic application)

โดยทั่วไปจะมีลักษณะโครงสร้างพื้นฐานแบ่งได้ 3 ส่วนหลักคือ

- การเปลี่ยนคริชพเซตเป็นฟัซซี่เซต (Fuzzier)
- กฎการวินิจฉัย (Inference mechanism that employs rules)
- การเปลี่ยนฟัซซี่เป็นคริชพ (Defuzzifier)

ในการใช้กับระบบจะต้องเปลี่ยนให้อยู่ในรูปของฟัซซี่โดเมน (Fuzzy domain) เคลื่อนย้ายประมวลผลข้อมูลแล้วเปลี่ยนกลับให้อยู่ในรูป คริชพโดเมน (Crisp domain) ตามเดิมซึ่งเหมือนกับการกระทำทางอนาลอก (Analog) คือ เปลี่ยนให้อยู่ในรูปของโดเมนความถี่ (Frequency domain) จากข้อมูลเดิมในโดเมนเวลา (Time domain) เพราะว่ากระบวนการในโดเมนความถี่ จะง่ายกว่าโดเมนเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในระบบฟัซซี่กฎพื้นฐานสามารถอธิบายการทำงานของระบบในรูปของฟัซซี่ได้ง่าย ดังนั้นเราจะเปลี่ยนค่าอินพุทในรูปของคริชพไปอยู่ในฟัซซี่ โดเมนมากกว่า จะเปลี่ยนฟัซซี่ไปอยู่ในรูปคริชโดเมน

กฎการวินิจฉัย (Inference)

สำหรับกฎการวินิจฉัยค่าอินพุตและค่าความจริง จะสนับสนุนเงื่อนไขสำหรับการสร้างส่วนกำหนดรูปแบบฟัซซี่ที่ช่วยฟัซซีคอมไพเลอร์ จะสุ่มค่าของอินพุตและนำมาวินิจฉัยค่า เพื่อให้ได้ผลออกมาทางเอาต์พุตของระบบตามทฤษฎี ระบบจะรวมค่าที่เป็นไปได้ทั้งหมดของอินพุทเพื่อนำมาวินิจฉัยและประมวลผล แต่จริง ๆ แล้วการครอบคลุมค่าเหล่านี้ไม่จำเป็นในการใช้งานปกติ กฎการวินิจฉัยนั้นได้มาจากการสร้างความสัมพันธ์ระหว่างอินพุตและเอาต์พุตขึ้นมา เพื่อนำไปสู่เอาต์พุตที่ต้องการ ซึ่งความสัมพันธ์ระหว่างอินพุทและเอาต์พุตก็คือ Rule นั้นเอง ซึ่ง Rule นั้นจะมีได้หลายลักษณะด้วยกัน เช่น เมื่อให้อินพุตของ Controller คือ Error(er) และ Change of error (ce) ของระบบป้อนกลับ และให้เอาต์พุตของ Controller เป็นอินพุตของ Process หรือ Control input (ci) เราสามารถสร้างความสัมพันธ์ได้ดังนี้คือ

IF er = LN AND ce = LN THEN ci = LP

OR

IF er = SN AND ce = SN THEN ci = SP

OR

.

.

.

หมายเหตุ LN = Large negative

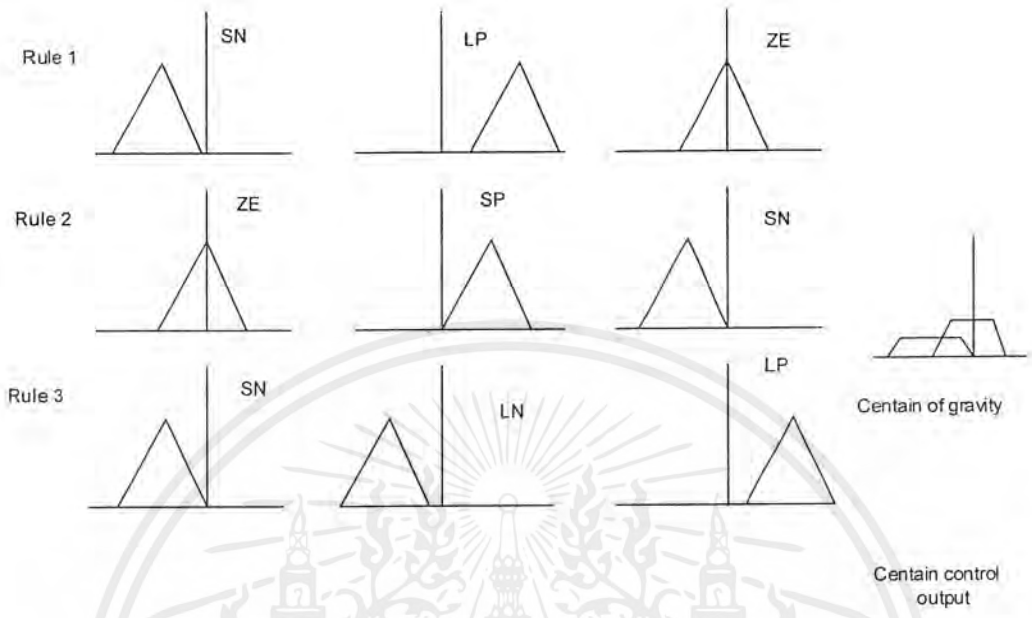
SN = Small negative

LP = Large positive

SP = Small positive

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากความสัมพันธ์ดังกล่าวนำมาแสดงได้ดังรูป



รูปที่ 2-10 แสดงการ Inference

ซึ่งเมื่อนำ Rule ทั้งหมดมารวมเข้าด้วยกัน ก็จะได้รูป ของ Membership function ของ Control input โดยจะนำไปหาค่าของ Control input ที่ต้องการได้ต่อไป

วิธีการ Defuzzifier

มีเทคนิคและวิธีการในการเปลี่ยนฟัซซี่เป็นคลิซ (Defuzzifier) มีอยู่หลายเทคนิค ซึ่งจะกล่าวถึงเป็นบางเทคนิคดังนี้

เทคนิค Maximizier เลือกค่าสูงสุดจากหลาย ๆ แบบมาเพียงหนึ่ง

เป็นการใช้ค่าสูงสุดของค่าระดับการเป็นสมาชิก จากการกระทำหลายๆ แบบ แล้วเลือกกระทำเพียงหนึ่งรูปแบบ ถ้าหากเกิดการกระทำที่มีค่า μ สูงสุดเท่ากัน 2 อย่าง จะต้องใช้รูปการแก้ปัญหาอีกลักษณะหนึ่ง คือ ใช้ค่าเฉลี่ยของเอาท์พุทหรือที่สัมพันธ์กับค่าระดับของระบบพื้นฐาน ถึงแม้เทคนิค Maximizer จะเป็นวิธีที่ง่ายที่สุด แต่ก็ไม่มีประสิทธิภาพเท่าที่ควร

ทฤษฎีค่าน้ำหนักเฉลี่ย

จะใช้ค่าเฉลี่ยของการกระทำหลังจากการกำหนดดค่าระดับของระดับการเป็นสมาชิกไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ล่วงหน้าแล้ว เป็นวิธีที่ง่ายและใช้การคำนวณเพียงเล็กน้อย แต่ก็ยังให้ค่าที่ไม่ค่อยชัดเจน เช่นเดียวกับเทคนิค Maximizer ที่เกิดความไม่ชัดเจนก็เพราะว่าค่าเอาท์พุทของฟังก์ชันระดับการเป็นสมาชิก (Membership function) มีค่าเอาท์พุทมากกว่าหนึ่งค่าต่อค่า μ ที่กำหนดให้ค่าเอาท์พุทฟังก์ชันของการเป็นสมาชิก มีลักษณะคล้ายกับปิระมิด หรือปิระมิดตัดยอดถ้า $\mu = 0.5$ ค่าเอาท์พุทมาจากค่าฟังก์ชันของขอบสัญญาณทั้งด้านขาขึ้นและขาลง ถ้า $\mu = 1.0$ จะมีค่าตรงกับช่วยสัญญาณที่เกิดขึ้นทั้งหมด

วิธีการกำจัดความไม่ชัดเจนสามารถทำได้ด้วยระบวนการแคมปิ้งค่าในฟังก์ชันเอาท์พุทด้วยค่าที่แน่นอนให้กลับไปอยู่ในฟังก์ชั่นอินพุท ซึ่งเป็นวิธีที่น่าเบื่อ และ ไม่สามารถใช้ค่าที่เป็นค่าตรงข้ามของฟังก์ชั่นอินพุทในการวิเคราะห์ด้วย

กรรมวิธีค่าศูนย์กลาง

เป็นการแสดงค่าเอาท์พุทที่สัมพันธ์กับค่าจุดศูนย์กลางมวลของเอาท์พุท ในระดับที่ทำงาน เพราะว่าเราไม่ใช้ค่าของของฟังก์ชันระดับการเป็นสมาชิก และจะไม่เกิดความไม่ชัดเจนอีกต่อไป กรรมวิธีค่าศูนย์กลางเป็นการคำนวณที่แน่นอน และเป็นการแก้ไขข้อบกพร่องที่เกิดขึ้นในวิธีอื่น ๆ

ค่าศูนย์กลางมักจะสัมพันธ์กับค่าเอาท์พุทหนึ่งค่า ผลที่ได้จะเป็นการกระทำอย่างหยาบ ๆ ภายในช่วยของเอาท์พุท กฎทั่วๆไป จะต้องมีการปฏิบัติเพียงจุดเดียวของแต่ระดับ ในกรณีที่ต้องทำพร้อม ๆ กันหลายกฎ ในการกระทำหนึ่งครั้ง จะต้องเกิดการซ้อนทับกัน (Overlap) ของค่าฟังก์ชันอินพุทของระดับการเป็นสมาชิก เพื่อแก้ไขความไม่ต่อเนื่องของเอาท์พุท ถึงแม้ว่าจะมีข้อบกพร่องแต่ก็เป็นวิธีที่ดีที่สุดในการรวมกัน (Combination) และการแปลงค่าฟัซซีกลับคืน (Defuzzification)

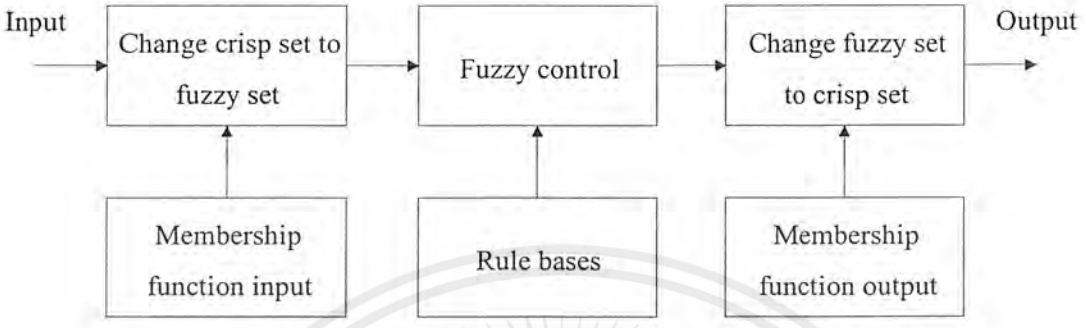
กรรมวิธีนี้จะรวมค่าเอาท์พุทของการกระทำหลาย ๆ ค่า เป็นค่าค่าเดียวสำหรับใช้ในระบบ และค่าเอาท์พุทค่าเดียวนี้เป็นค่าน้ำหนักเฉลี่ยของศูนย์กลาง (Centroid) ของแต่ละฟังก์ชัน ระดับการเป็นสมาชิก

กรรมวิธีซิงเกิลตัน : สังเคราะห์เอาท์พุทเดียว

เป็นเทคนิคพิเศษของกรรมวิธีค่าศูนย์กลาง หรือจะเรียกอีกอย่างหนึ่งคือ กรรมวิธี Remaining combination / defuzzification วิธีนี้เป็นการนำค่าเอาท์พุทของและฟัซซี่เซตมาใช้ใหม่เป็นค่าเอาท์พุทค่าเดียวโดยใช้ค่าน้ำหนักเฉลี่ยจากการกระทำรวมกันหลาย ๆ อย่าง วิธีนี้ได้ค่าความถูกต้องน้อยกว่ากรรมวิธีค่าศูนย์กลาง และยังคงต้องการซ้อนทับกันของอินพุทฟังก์ชัน เพื่อเป็นการหลีกเลี่ยงความไม่ต่อเนื่องของเอาท์พุท ด้วยหลักการและการคำนวณที่ไม่ยุ่งยากประกอบกับยังไม่มีใครคิดค้นวิธีใหม่และดีกว่านี้ กรรมวิธีนี้จึงน่าจะใช้แทนกรรมวิธีค่าศูนย์กลางได้ดีที่สุด

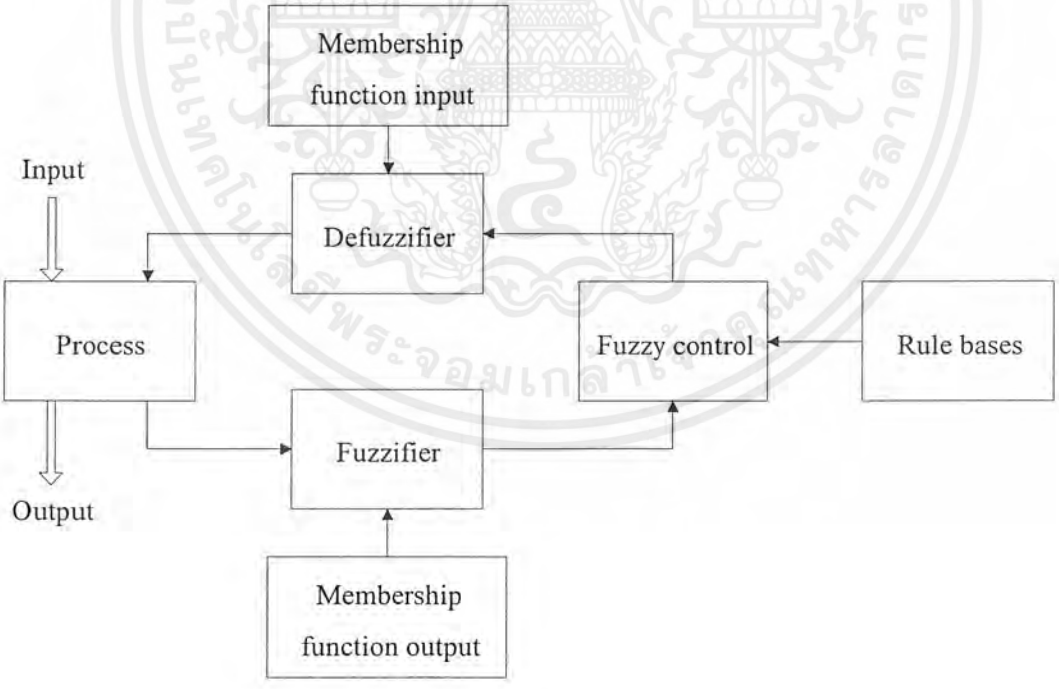
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างพื้นฐานของระบบฟัซซี่ลอจิกแสดงดังรูป



รูปที่ 2-11 แสดงโครงสร้างพื้นฐานของระบบฟัซซี่ลอจิก

สำหรับหลักการ Defuzzifier กรรมวิธีค่าศูนย์กลาง ซึ่งระบบควบคุมป้อนกลับสามารถแสดงดังรูป



รูปที่ 2-12 โครงสร้างพื้นฐานของระบบควบคุมป้อนกลับด้วย Fuzzy Controller

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการหา Fuzzy control rule

ในการออกแบบ Fuzzy Controller นั้น ปัญหาสำคัญอันหนึ่งคือการหา Fuzzy control rule ที่เหมาะสม ซึ่งวิธีที่สามารถนำมาใช้ได้จริงในงาน Process control อยู่ 3 วิธีด้วยกัน คือ

1. จากความรู้และประสบการณ์ของผู้เชี่ยวชาญ

Fuzzy Controller ส่วนมากจะได้รับการออกแบบขึ้นโดยอ้างอิงกับ ความรู้และประสบการณ์ของวิศวกรควบคุมซึ่งอันที่จริงแล้วก็สามารถกล่าวได้ว่า Fuzzy control นั้นเป็น Application ที่ใช้ได้จริงอันแรกด้านระบบผู้เชี่ยวชาญ (Expert system) เลยทีเดียว

การออกแบบด้วยวิธีนี้เหมาะกับระบบที่ผู้ควบคุม ควบคุมเป็นหลักสำคัญ Control rule นั้นก็จะได้มาจากการ List วิธีการควบคุมต่าง ๆโดยวิศวกรควบคุม แต่ข้อเสียของวิธีนี้ก็คือ ส่วนมากผู้ควบคุมจะไม่สามารถถ่ายทอดความรู้ที่มีออกมาได้เป็นลายลักษณ์อักษร โดยเฉพาะเมื่อระบบมีความซับซ้อนมาก

2. จากการจำลองลักษณะการทำงานของผู้ควบคุม

เมื่อทักษะของผู้ควบคุมเป็นสิ่งสำคัญ จึงจำเป็นมากที่จะต้องหา Fuzzy control rule มาจากการจำลองลักษณะการทำงานของผู้ควบคุม ซึ่งวิธีนี้สามารถเป็นไปได้ที่จะแปลงการทำงานของผู้ควบคุมมาให้เป็น Input และ Output ของ Controller

วิธีนี้ค่อนข้างคล้ายคลึงกับการจำลองกระบวนการ แต่ว่าการจำลองการทำงานของผู้ควบคุมนั้นจะง่ายกว่าการจำลองกระบวนการ เพราะ Input ของระบบจะหาได้ง่ายกว่า แต่ในสถานการณ์จริงนั้นก็ควรจะรวมวิธีที่ 1 และ 2 เข้าด้วยกัน

3. จากกการจำลองกระบวนการ

ในวิธีแรกนั้นจะมีพื้นฐานมากจากความคิดคร่าว ๆ ที่เกี่ยวกับคุณลักษณะของกระบวนการ เช่น Output เพิ่มขึ้นเมื่อ Input เพิ่มขึ้น กระบวนการที่มี Timelag ฯลฯ ส่วนวิธีที่ 2 จะใช้เฉพาะตัวแปรที่มีให้แก่ผู้ควบคุมกระบวนการ ซึ่งทั้ง 2 วิธีจะใช้ได้ดีเฉพาะในกรณีที่ผู้ควบคุมมีบทบาทสำคัญในการควบคุมกระบวนการเท่านั้น

แต่ถ้าไม่ต้องการขึ้นอยู่กับผู้ควบคุม และต้องการให้ผลการควบคุมดีกว่าการควบคุมด้วยผู้ควบคุม ก็จะมีการออกแบบอีกวิธีหนึ่งคือ การจำลองกระบวนการ ซึ่งจะเป็นวิธีการซับซ้อนกว่า ซึ่งการออกแบบด้วยวิธีนี้ได้มีการศึกษาวิจัยมาหลายครั้งด้วยกัน การจำลองกระบวนการในที่นี้จะเป็นการแสดงคุณลักษณะของ Process ออกมาด้วย Fuzzy set โดยพิจารณาจาก Input ตัวแปรสถานะ และ Output

มีแนวความคิดสองทางด้วยกันในการออกแบบ Fuzzy Controller จากแบบจำลอง Fuzzy

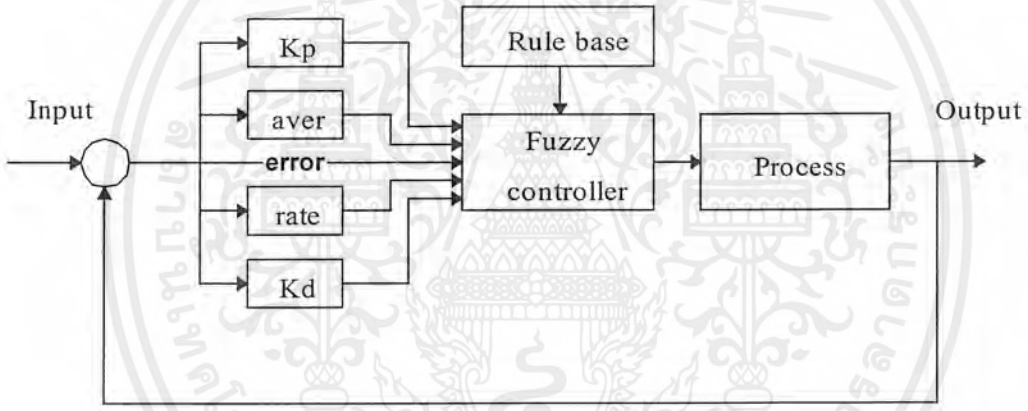
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-ทางแรก คือ การออกแบบให้ Control rule นั้นทำการ Compensate ลักษณะที่ไม่ต้องการของกระบวนการเพื่อให้ได้ตามเป้าหมายที่ต้องการ

-ทางที่สอง คือ เป็นไปตามทฤษฎีของ Optimal control ซึ่งจะให้โครงสร้างและ Parameter ของ Control rule ทำให้ระบบที่มี Fuzzy Controller ควบคุมอยู่นั้นเป็นไปตามเป้าที่ต้องการ

การออกแบบตัวควบคุมแบบ Fuzzy PD

ในโครงงานนี้ได้นำเสนอตัวควบคุมแบบ Fuzzy PD ซึ่งมี Block Diagram ดังนี้



รูปที่ 2-13 Block Diagram ของระบบควบคุมแบบ Fuzzy PD

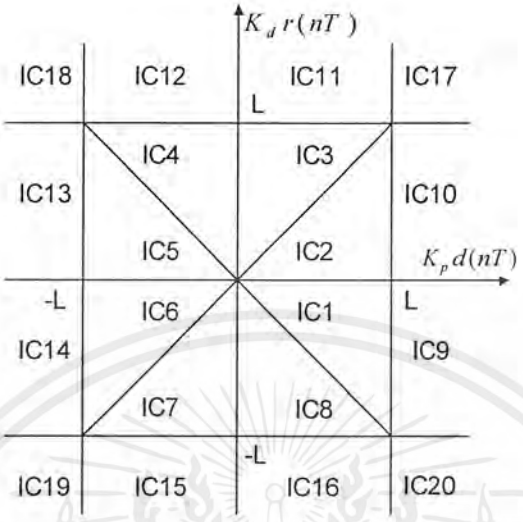
Rulr base

ในการตั่งกฎ ได้มีการตั่งกฎ จากกฎพื้นฐานดังนี้

- Rule 1 If error = ep AND rate = rp THEN output = oz.
- Rule 2 If error = ep AND rate = rn THEN output = op.
- Rule 3 If error = en AND rate = rp THEN output = on.
- Rule 4 If error = en AND rate = rn THEN output = oz.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นจะใช้ประสบการณ์จากผู้เชี่ยวชาญสร้างกฎขึ้นมาจากแผนภาพดังนี้



รูปที่ 2-14 Regions ของ Fuzzy controller

ซึ่ง Output fuzzy จะเป็นไปตามสมการนี้

$$\Delta u(nT) = \frac{rp \times oz + rn \times op + en \times on + en \times oz}{rp + rn + ep + en}$$

โดยค่าต่างๆจะกำหนดโดยผู้เชี่ยวชาญดังนี้

$$ep = \frac{K_p d(nT) + L}{2L}, \qquad en = \frac{-K_p d(nT) + L}{2L}$$
$$rp = \frac{K_d r(nT) + L}{2L}, \qquad rn = \frac{-K_d r(nT) + L}{2L}$$

จะได้กฎที่ใช้ในการออกแบบดังนี้

1. If ((Kd * Abs(rate(m))) <= (Kp * Abs(ave_err(m)))) And ((Kp * Abs(ave_err(m))) <= L) Then
   Fuzzy_out(m) = 0.5 * L * Ku * (-Kd * rate(m) + Kp * ave_err(m)) / (2 * L - Kp * Abs(ave_err(m)))

2. If (((Kp * Abs(ave_err(m))) <= (Kd * Abs(rate(m)))) And ((Kd * Abs(rate(m))) <= L)) Then
   Fuzzy_out(m) = 0.5 * L * Ku * (-Kd * rate(m) + Kp * ave_err(m)) / (2 * L - Kp * Abs(rate(m)))

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. If ((Kp * ave_err(m)) > L) And ((Kd * Abs(rate(m))) <= L) Then

   Fuzzy_out (m) = Ku * (-Kd * rate(m) + L) / 2

4. If ((Kd * rate(m)) > L) And ((Kp * Abs(ave_err(m))) <= L) Then

   Fuzy_out (m) = Ku * (Kp * ave_err(m) - L) / 2

5. If ((Kp * ave_err(m)) < -L) And ((Kd * Abs(rate(m))) < L) Then

   Fuzzy_out (m) = Ku * (-Kd * rate(m) - L) / 2

6. If ((Kd * rate(m)) < -L) And ((Kp * Abs(ave_err(m))) < L) Then

   Fuzzy_out (m) = Ku * (Kp * ave_err(m) + L) / 2

7. If (((Kd * rate(m)) > L) And ((Kp * ave_err(m)) > L)) Or (((Kd * rate(m)) < -L) And ((Kp * ave_err(m)) < L)) Then

   Fuzzy_out (m) = 0

8. If ((Kd * rate(m)) > L) And ((Kp * ave_err(m)) < -L) Then

   Fuzzy_out (m) = -L * Ku

9. If ((Kp * ave_err(m)) > L) And ((Kd * rate(m)) < -L) Then

   Fuzzy_out (m) = L * Ku

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การสร้างและพัฒนาโปรแกรม

โปรแกรมออกแบบระบบควบคุม ที่ได้เขียนขึ้นมานี้เป็นโปรแกรมที่ทำงานบนวินโดว์ ซึ่งเขียนขึ้นจากโปรแกรม Visual Basic โดยมีการออกแบบให้ใช้ง่าย ประสิทธิภาพสูง และยืดหยุ่นสูง

ความสามารถของโปรแกรม

1. สามารถออกแบบ Controller แบบต่างๆ ได้ดังนี้
    1.1 P Control
    1.2 PI Control
    1.3 PD Control
    1.4 PID Control
    1.5 PID Zigler Nichols
    1.6 Fuzzy Control
2. สามารถเปลี่ยน Process ได้โดยง่าย โดยป้อนตัว Process เป็น Transferfunction
3. มีส่วนรับค่าพารามิเตอร์ (ใส่ค่าตาม Mode ที่เลือกไว้) เพื่อเอาไว้ปรับเปลี่ยนค่าพารามิเตอร์เพื่อให้ระบบเสถียรดีที่สุด
4. การแสดงผลสามารถแสดงผลได้ทั้งกราฟ และตัวเลข

การออกแบบโปรแกรม

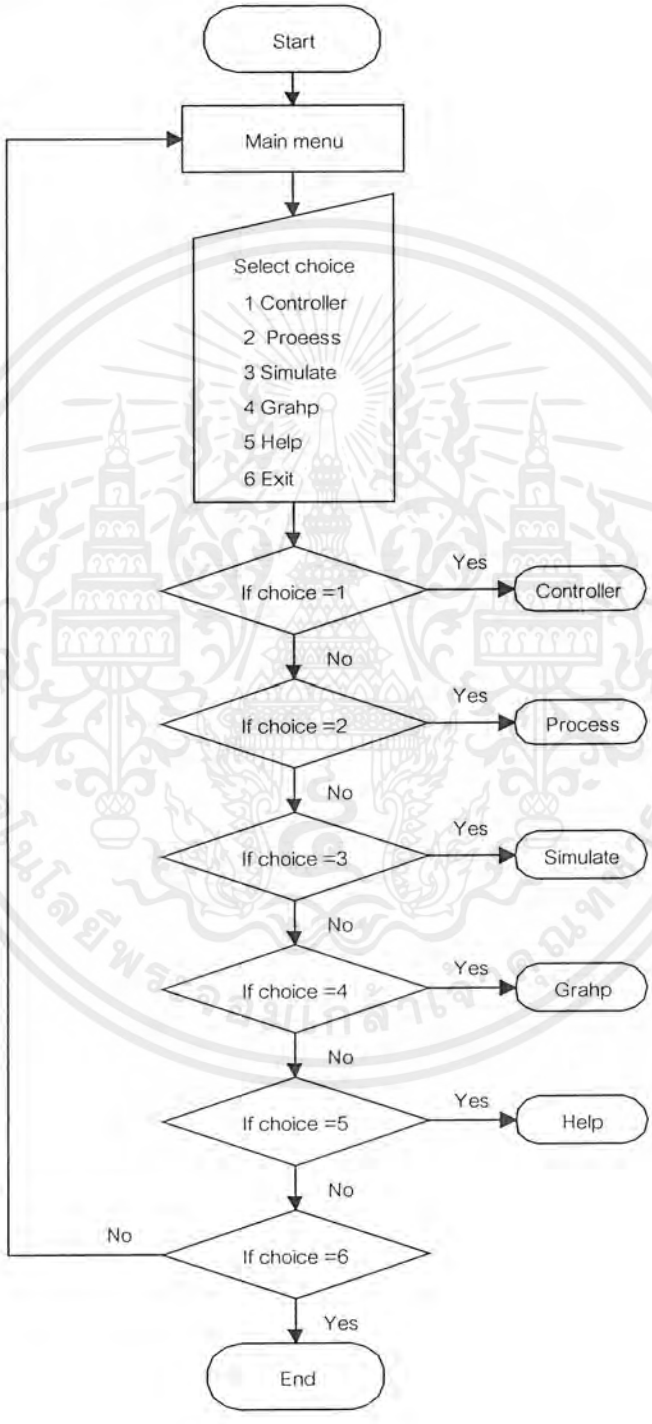
โปรแกรมออกแบบระบบควบคุมนี้ จะมีส่วนสำคัญอยู่ 2 ส่วนคือ

1. ส่วนของหน้าต่างที่ใช้ติดต่อกับผู้ใช้
2. ส่วนของโปรมแกรมคำนวณเพื่อหาเอาท์พุท ซึ่งแยกเป็นตัวควบคุมแบบต่างๆดังนี้
    - P Control
    - PI Control
    - PD Control
    - PID Control
    - PID Zigler Nichols
    - Fuzzy Control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของหน้าต่างที่ใช้ติดต่อกับผู้ใช้

ในส่วนของหน้าต่างที่ใช้ติดต่อกับผู้ใช้นั้น ได้มีการออกแบบให้ใช้งานง่ายโดยหน้าต่างแรกจะเป็นหน้าต่าง Menu โดยมี Flow Chart การทำงานดังนี้

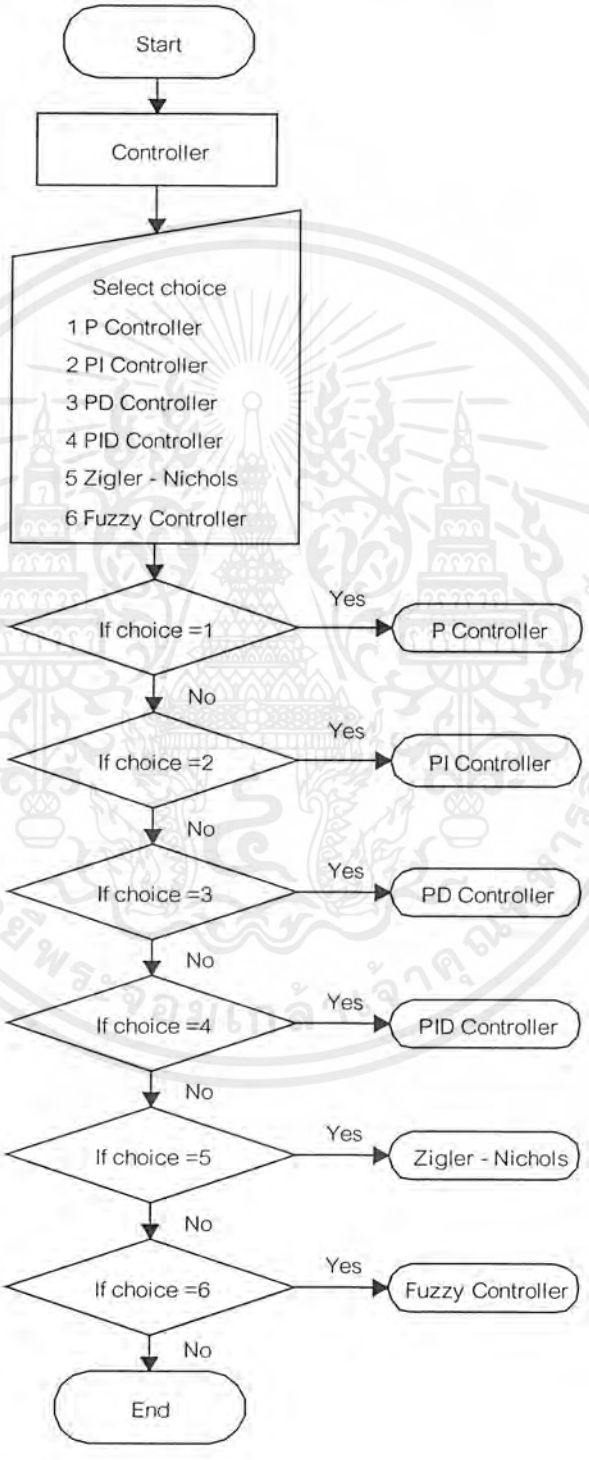


รูปที่ 3-1 Flow chart หน้าต่าง Menu

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าต่าง Controller

ในส่วนของหน้าต่าง Controller ของโปรแกรมนี้จะเป็นหน้าต่างสำหรับเลือกชนิดของตัวควบคุม ซึ่งมีตัวควบคุมให้เลือกตามต้องการ โดยมี Flow Chart การทำงานดังนี้

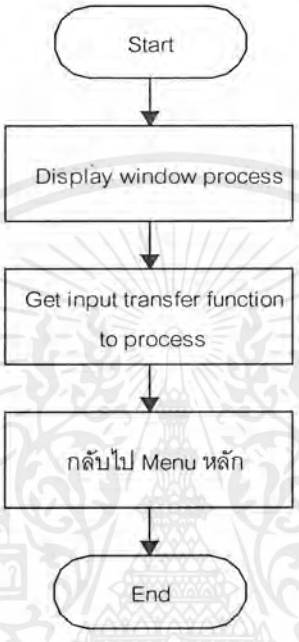


รูปที่ 3-2 Flow chart หน้าต่าง Controller

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าต่าง Plant

ส่วนของหน้าต่าง Plant ของโปรแกรมนี้จะเป็นหน้าต่างสำหรับให้ผู้ใช้ใส่ค่า Plant ที่จะนำมาควบคุม เป็น Transfer function โดยมี Flow Chart การทำงานดังนี้



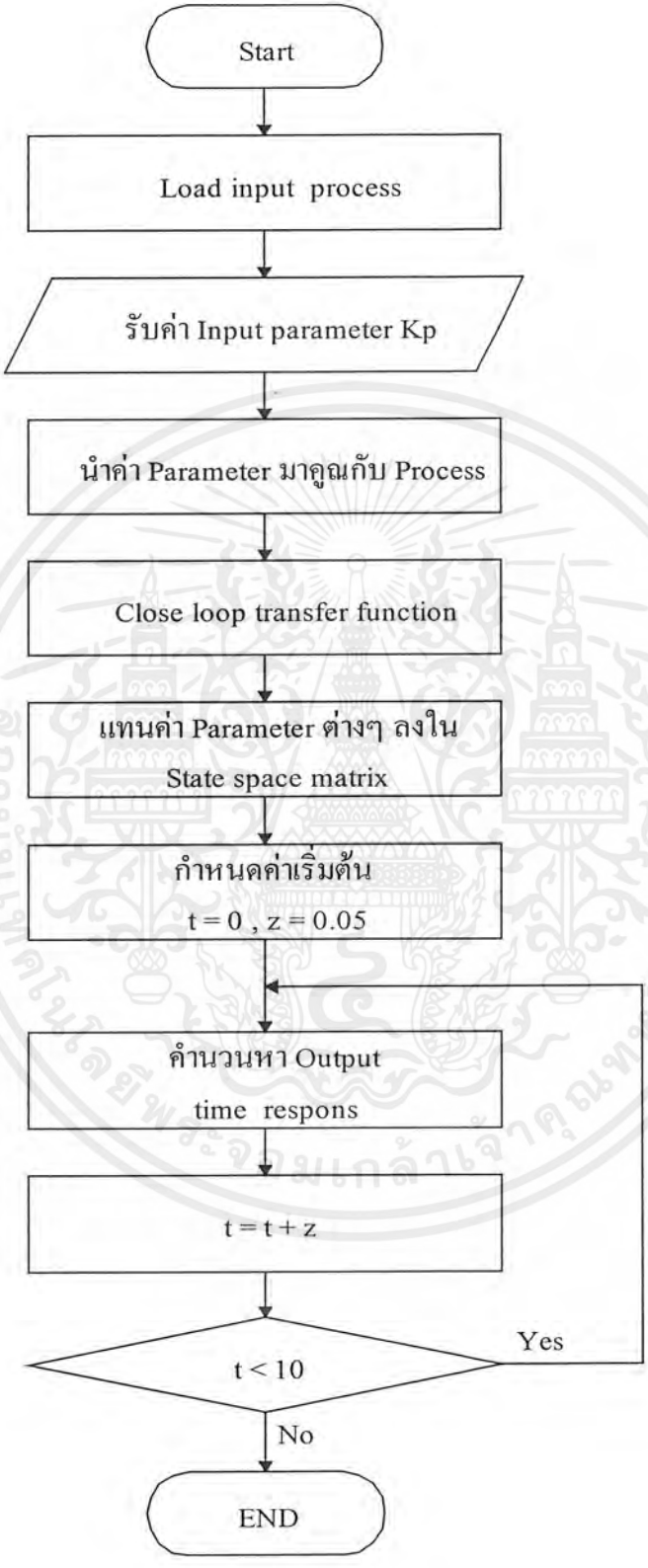
รูปที่ 3-3 Flow Chart ของ Plant

ส่วนของโปรมแกรมคำนวณเพื่อหาเอาท์พุท

ในส่วนของโปรมแกรมคำนวณเพื่อหาเอาท์พุท จะประกอบด้วย Controller แบบต่างๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

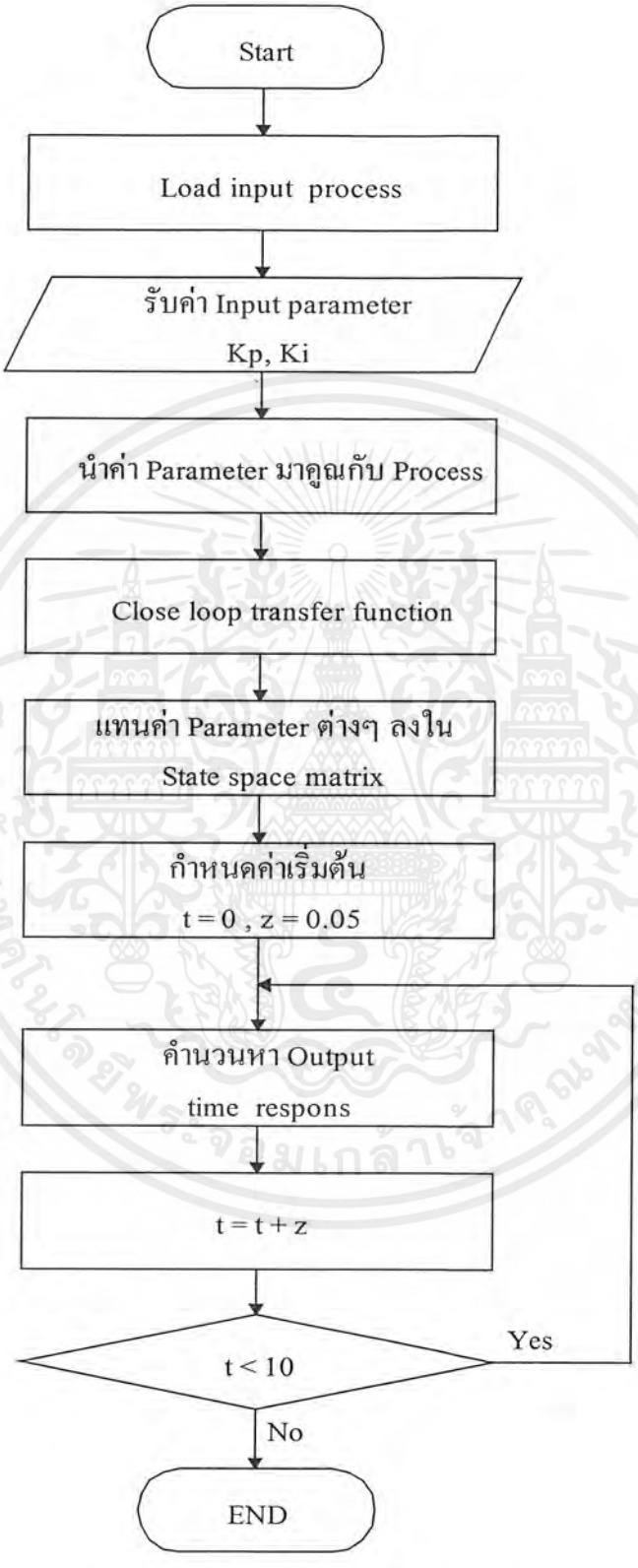
P Controller



รูปที่ 3-4 Flow Chart ของ P Controller

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

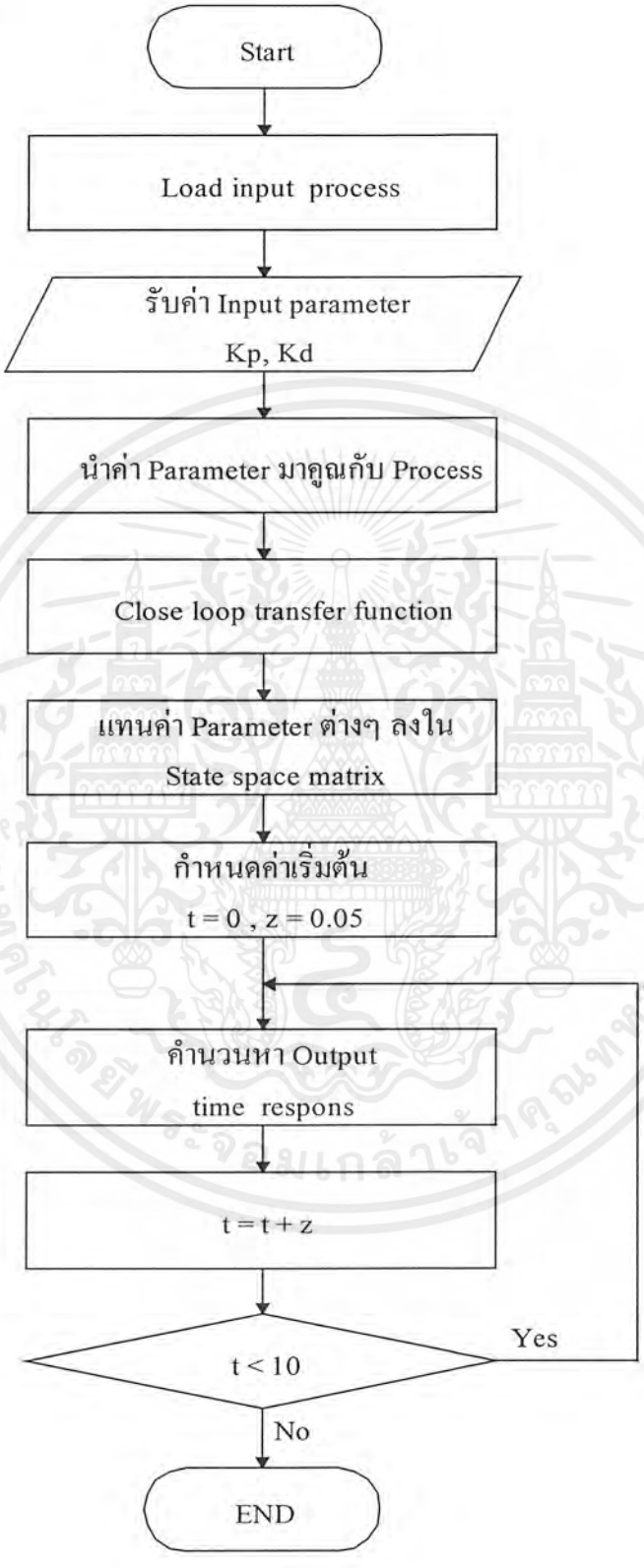
PI Controller



รูปที่ 3-5 Flow Chart ของ PI Controller

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

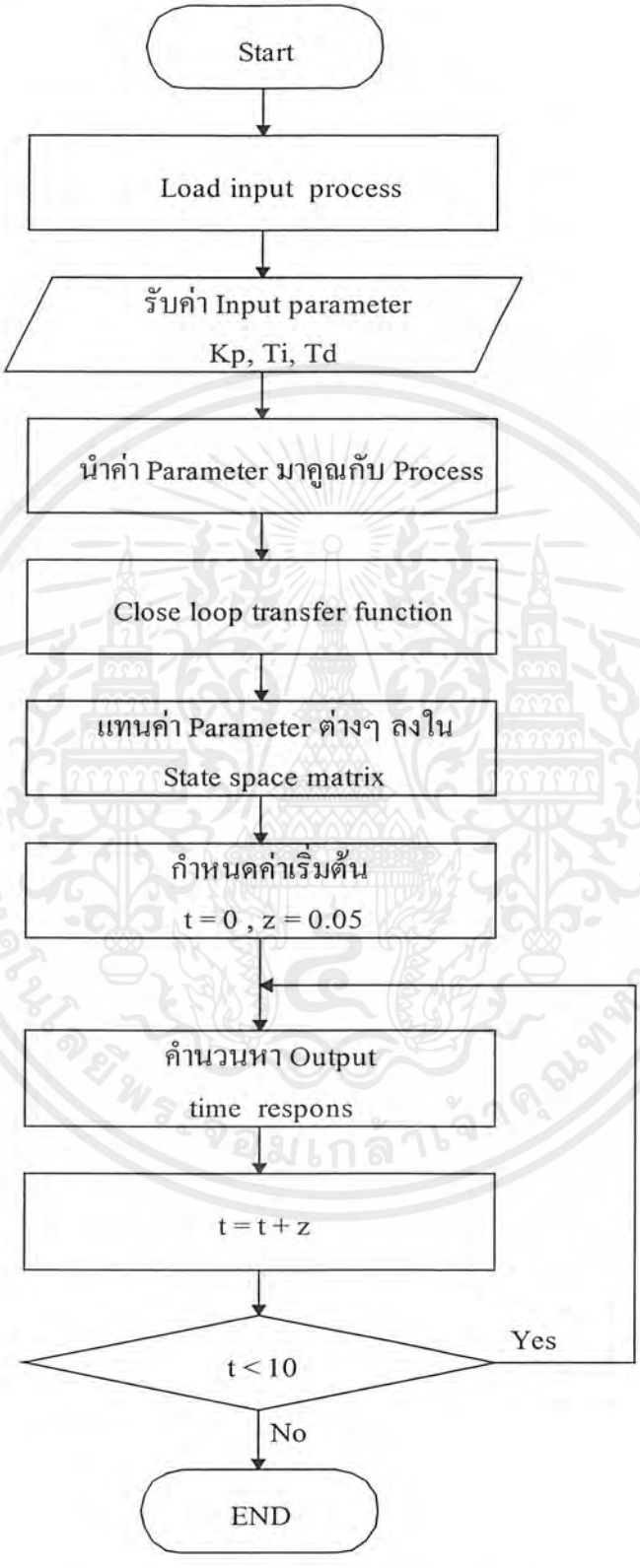
PD Controller



รูปที่ 3-6 Flow Chart ของ PD Controller

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

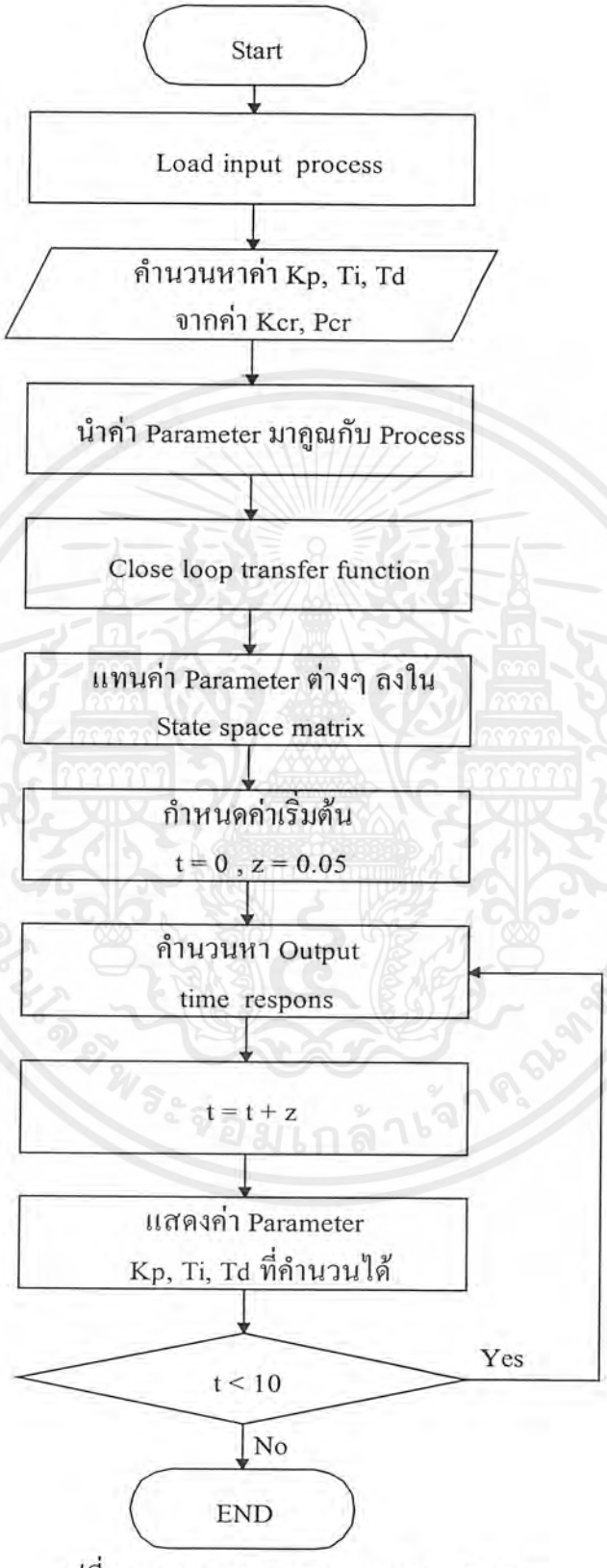
PID Controller



รูปที่ 3-7 Flow Chart ของ PID Controller

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

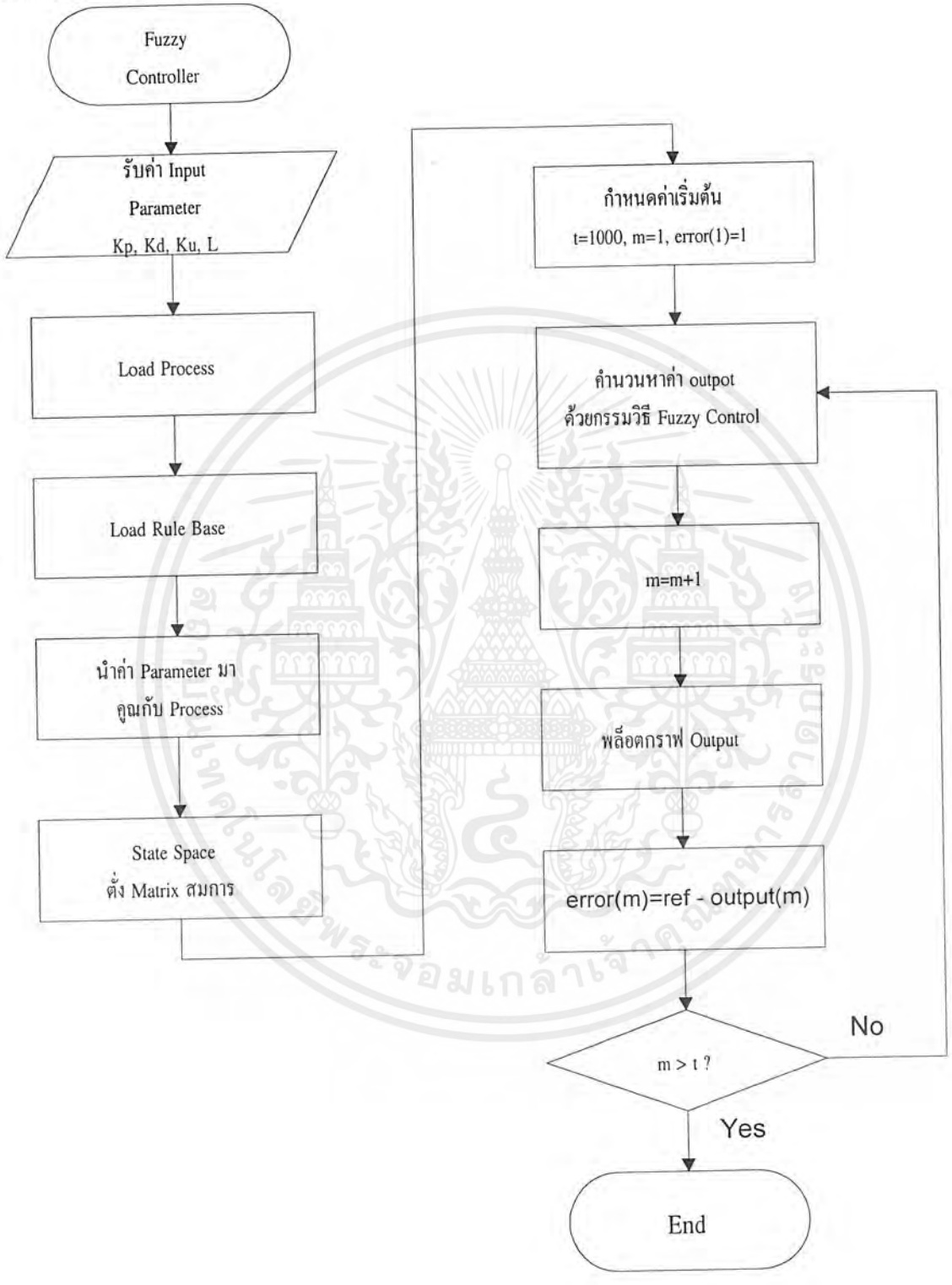
PID Zigler Nichols



รูปที่ 3-8 Flow Chart ของ PID Zigler Nichols

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Fuzzy PD Controller



รูปที่ 3-9 Flow Chart แสดงการทำงานของ Fuzzy Controller

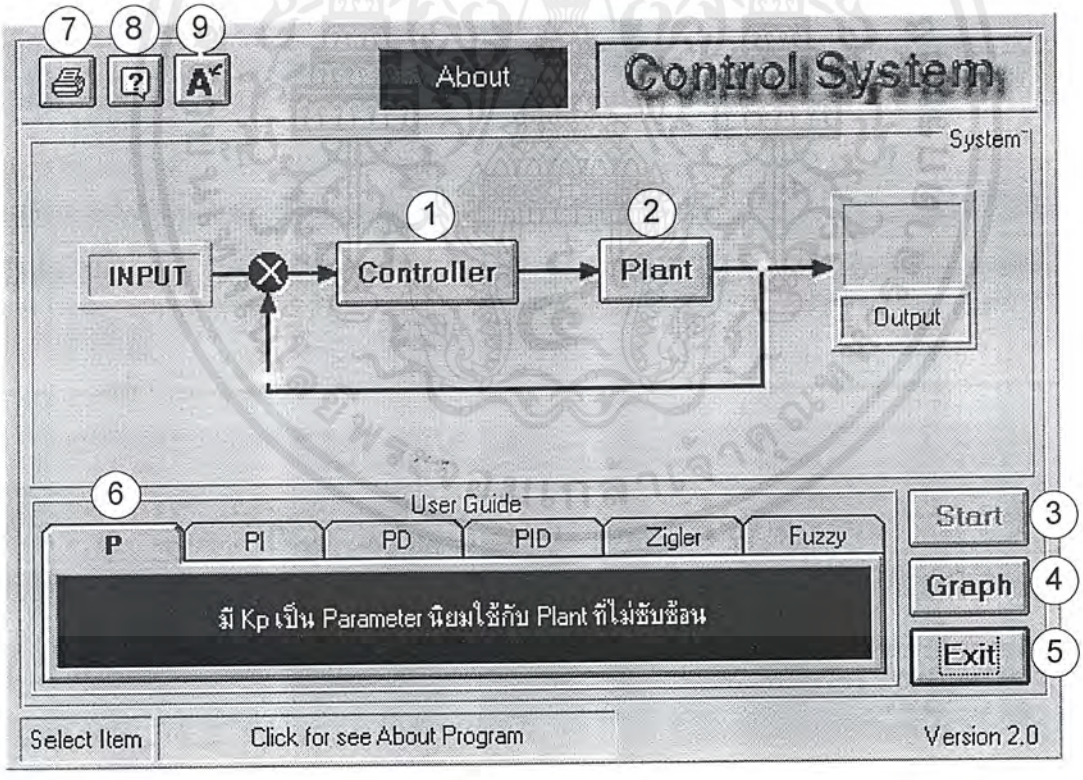
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ส่วนประกอบหลักและวิธีการใช้โปรแกรม

ในบทนี้จะกล่าวถึงส่วนประกอบหลักและวิธีการใช้งานซอฟท์แวร์จำลองกระบวนการที่โครงงานนี้ได้จัดทำขึ้น เพื่อเป็นคู่มือสำหรับการเริ่มต้นใช้งานโปรแกรม

หน้าต่าง Menu หลัก

สำหรับหน้าต่าง Menu หลัก จะเป็นหน้าต่างเริ่มต้นใช้งานโปรแกรม ซึ่งมีส่วนประกอบต่างๆ ดังรูปที่ 4-1



รูปที่ 4-1 หน้าต่าง Menu

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ปุ่ม Controller เมื่อคลิ้กปุ่มนี้จะเรียกหน้าต่าง Controller ขึ้นมาเพื่อเลือกชนิดของตัวควบคุม ซึ่งจะกล่าวโดยละเอียดต่อไป

2. ปุ่ม Plant เมื่อคลิ้กปุ่มนี้จะเรียกหน้าต่าง Plant ขึ้นมาเพื่อใส่ค่า Process ให้กับโปรแกรม ซึ่งจะกล่าวโดยละเอียดต่อไป

3. ปุ่ม Start เมื่อคลิ้กปุ่มนี้จะเรียกหน้าต่าง Simulate ขึ้นมา ซึ่งแล้วแต่ว่าผู้ใช้เลือก Controller แบบใดไว้ ก็จะปรากฏหน้าต่าง Simulate ตาม Controller ที่เลือกไว้ ซึ่งจะกล่าวโดยละเอียดต่อไป

4. ปุ่ม Graph จะเป็น Option แสดงผลเส้นกราฟหลายๆ เส้น เพื่อให้ผู้ใช้ ใช้เปรียบเทียบ Output เมื่อเปลี่ยนค่าพารามิเตอร์ต่างๆ เมื่อคลิ้กปุ่มนี้จะเรียกหน้าต่าง Graph ขึ้นมา ซึ่งจะมีชนิดของ Controllerให้เลือกอีกว่าต้องการเปรียบเทียบ Controller แบบใด ซึ่งจะกล่าวโดยละเอียดต่อไป

5. ปุ่ม Exit คลิ้กปุ่มนี้เมื่อต้องการออกจากโปรแกรม

6. แท็ป User Guide เป็นทิปเล็กๆช่วยในการเลือกใช้ Controller

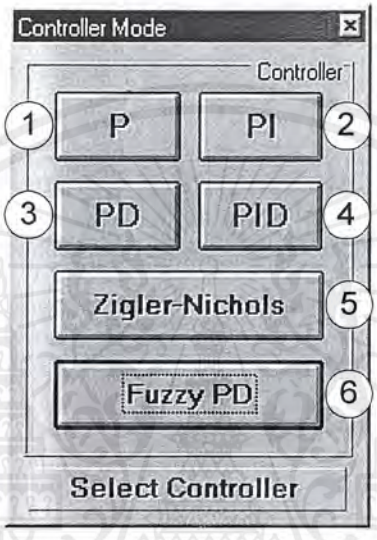
7. ปุ่มรูป Printer คลิ้กปุ่มนี้เมื่อต้องการพิมพ์ภาพออกเครื่องพิมพ์

8. ปุ่มรูป Help เมื่อคลิ้กปุ่มนี้จะเรียกหน้าต่าง Help ออกมา เพื่อใช้ช่วยเหลือการใช้โปรแกรม ซึ่งจะกล่าวโดยละเอียดต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าต่าง Controller

สำหรับหน้าต่าง Controller จะเป็นหน้าต่างสำหรับเลือกชนิดของตัวควบคุม ซึ่งเมื่อคลิ้กปุ่ม Controller ที่หน้าต่าง Menu หลักแล้ว หน้าต่าง Controller นี้จะแสดงขึ้นมา ซึ่งมีส่วนประกอบต่างๆ ดังรูปที่ 4-2



รูปที่ 4-2 หน้าต่าง Controller

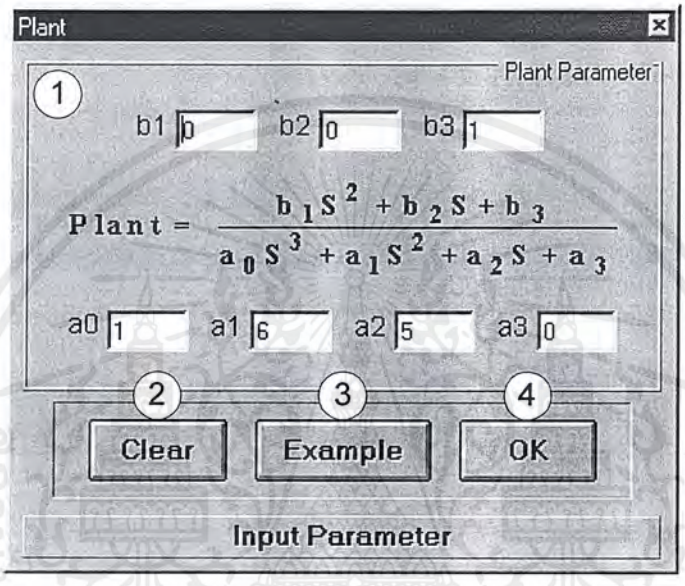
1. ปุ่ม P คลิ้กปุ่มนี้เพื่อเลือกตัวควบคุมแบบ P
2. ปุ่ม PI คลิ้กปุ่มนี้เพื่อเลือกตัวควบคุมแบบ PI
3. ปุ่ม PD คลิ้กปุ่มนี้เพื่อเลือกตัวควบคุมแบบ PD
4. ปุ่ม PID คลิ้กปุ่มนี้เพื่อเลือกตัวควบคุมแบบ PID
5. ปุ่ม PID Zigler Nichols คลิ้กปุ่มนี้เพื่อเลือกตัวควบคุมแบบ PID Zigler Nichols
6. ปุ่ม Fuzzy PD คลิ้กปุ่มนี้เพื่อเลือกตัวควบคุมแบบ Fuzzy PD

หลังจากคลิ้กปุ่มเพื่อเลือกตัวควบคุมแล้ว โปรแกรมจะกลับไปที่หน้าต่าง Menu หลักเองโดยอัตโนมัติ และจะแสดงชื่อของตัวควบคุมที่ผู้ใช้เลือกที่ปุ่ม Controller

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าต่าง Plant

สำหรับหน้าต่าง Plant จะเป็นหน้าต่างสำหรับให้ผู้ใช้ใส่ค่า Process ที่อยู่ในรูปของ Transfer functuon ให้กับโปรแกรม ซึ่งเมื่อคลิ้กปุ่ม Plant ที่หน้าต่าง Menu หลักแล้ว หน้าต่าง Plant นี้จะแสดงขึ้นมา ซึ่งมีส่วนประกอบต่างๆ ดังรูปที่ 4-3



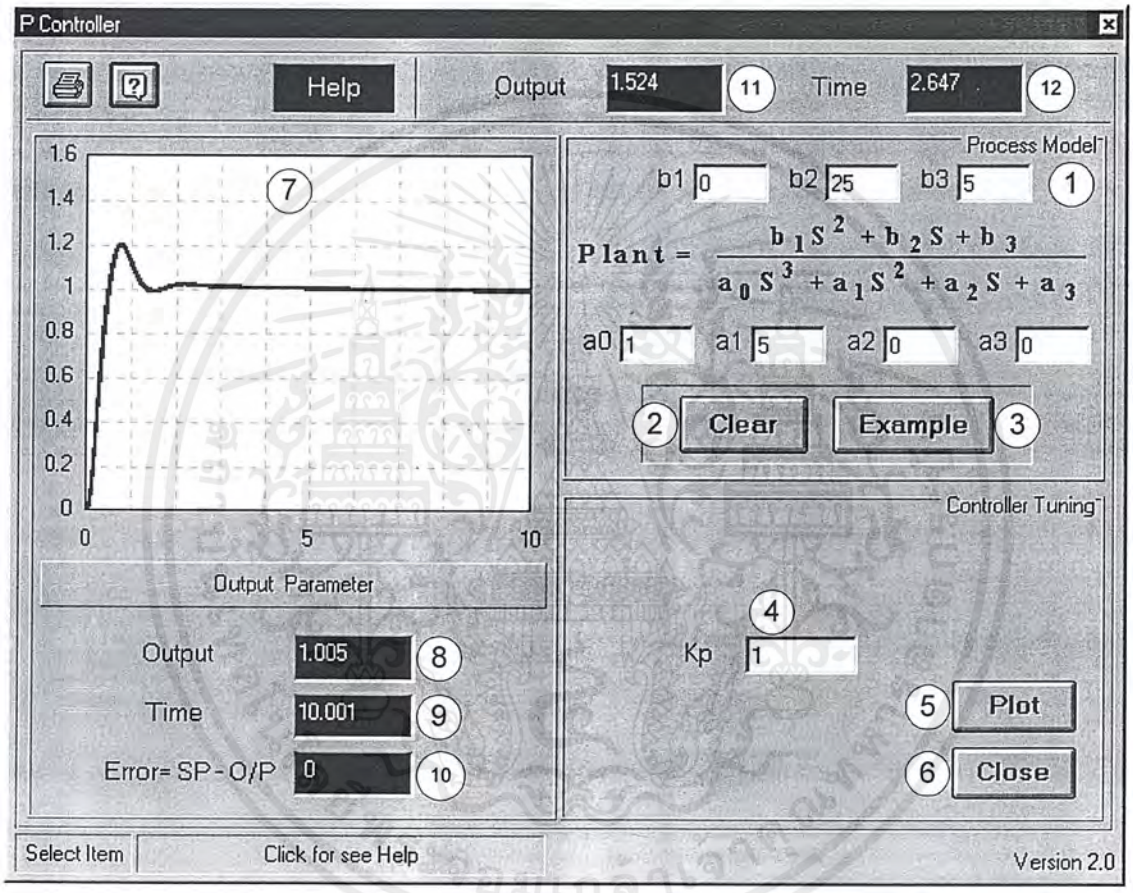
รูปที่ 4-3 หน้าต่าง Plant

1. กรอบ Plant Parameter เป็นกรอบสำหรับให้ผู้ใช้ ใส่ค่า Process ที่อยู่ในรูปของ Transfer functuon ให้กับโปรแกรม โดยใส่ที่ช่องสำหรับใส่ตัวเลข ซึ่งมีรูปแบบตาม Model ที่แสดงตัวอย่างไว้ เช่น a0 คือค่าสัมประสิทธ์ที่อยู่หน้า S^3
2. ปุ่ม Clear คลิ้กปุ่มนี้เพื่อ Clear ค่าตัวเลขที่อยู่ในช่องสำหรับใส่ตัวเลข
3. ปุ่ม Example คลิ้กปุ่มนี้เพื่อใช้ค่า Process ที่กำหนดไว้เป็นตัวอย่าง
4. ปุ่ม OK คลิ้กปุ่มนี้เพื่อยืนยันค่า Process แล้วกลับไปหน้าต่าง Menu หลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าต่าง P Controller

สำหรับหน้าต่าง P Controller จะเป็นหน้าต่างที่ใช้สำหรับออกแบบระบบควบคุม และคำนวณหา Output Time Respons ซึ่งใช้ตัวควบคุมแบบ P โดยมีการแสดงผลการควบคุมเป็นทั้งกราฟและตัวเลข เมื่อคลิ้กปุ่ม Start ที่หน้าต่าง Menu หลักแล้ว หน้าต่าง P Controller นี้จะแสดงขึ้นมา (ซึ่งขึ้นอยู่กับตอนเลือก Controller) และมีส่วนประกอบต่างๆ ดังรูปที่ 4-4



รูปที่ 4-4 หน้าต่าง P Controller

1. กรอบ Process Model เป็นกรอบสำหรับให้ผู้ใช้ ใส่ค่า Process ที่อยู่ในรูปของ Transfer functuon ให้กับโปรแกรม โดยใส่ที่ช่องสำหรับใส่ตัวเลข ซึ่งมีรูปแบบตาม Model ที่แสดงตัวอย่างไว้ เช่น a0 คือค่าสัมประสิทธ์ที่อยู่หน้า S^3 ซึ่งในตอนแรกค่าที่กรอบนี้จะถูกส่งมาจากหน้าต่าง Plant และสามารถเปลี่ยนแปลงค่าที่กรอบนี้ได้ทันที
2. ปุ่ม Clear คลิ้กปุ่มนี้เพื่อ Clear ค่าตัวเลขที่อยู่ในช่องสำหรับใส่ตัวเลข และ Clear รูปกราฟที่อยู่ในกรอบกราฟด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ปุ่ม Example คลิ้กปุ่มนี้เพื่อใช้ค่า Process ที่กำหนดไว้เป็นตัวอย่าง

4. TextBox Input Kp สำหรับใส่ค่า Input Parameter Kp

5. ปุ่ม Plot คลิ้กปุ่มนี้เพื่อ Simulate ดูกราฟ Output

6. ปุ่ม Close คลิ้กปุ่มนี้เพื่อปิดหน้าต่าง P Controller แล้วกลับไปหน้าต่าง Menu หลัก

7. กรอบกราฟ เป็นกรอบสำหรับแสดงกราฟเอาท์พุท

8. TextBox Output สำหรับแสดงค่าเอาท์พุทเป็นตัวเลข

9. TextBox Output Time สำหรับแสดงค่าเอาท์พุทเวลาเป็นตัวเลข

10. TextBox Output Error สำหรับแสดงค่าเอาท์พุท Error เป็นตัวเลข

11. TextBox Output สำหรับแสดงค่าเอาท์พุทตรงที่เมาส์ชี้เป็นตัวเลข

12. TextBox Output Time สำหรับแสดงค่าเอาท์พุทเวลาตรงที่เมาส์ชี้เป็นตัวเลข

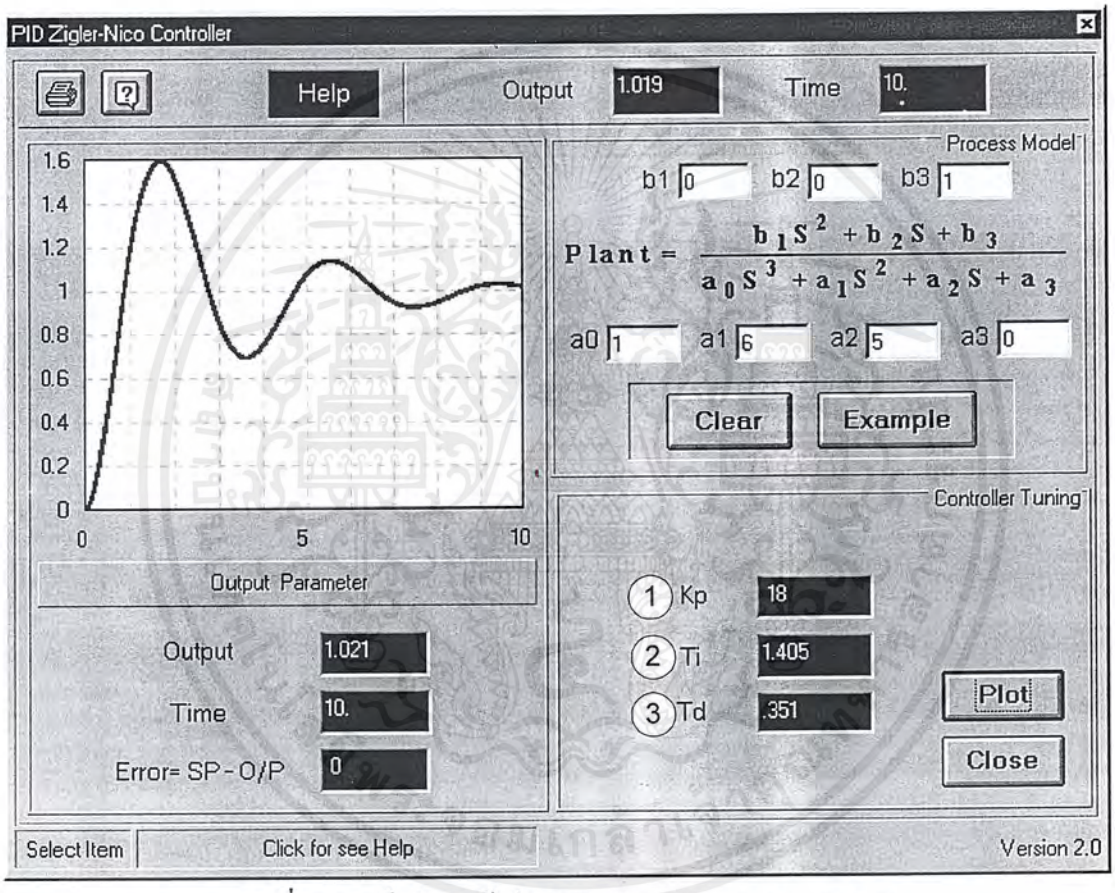
หน้าต่าง Controller แบบอื่น ๆ

สำหรับหน้าต่าง Controller แบบอื่นๆที่มีลักษณะเหมือนกันกับแบบ P Controller ก็คือ แบบ PI Controller, PD Controller, PID Controller จะมีข้อแตกต่างกันตรงที่ อินพุทพารามิเตอร์ของแต่ละ Controller จะไม่เหมือนกันเช่น PI Controller จะมีอินพุทพารามิเตอร์เป็น Kp กับ Ki เป็นต้น จึงขอไม่กล่าวถึงในที่นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าต่าง PID Zigler Nichols Controller

สำหรับหน้าต่าง PID Zigler Nichols Controller จะเป็นหน้าต่างที่ใช้สำหรับออกแบบระบบควบคุม และคำนวณหา Output Time Respons ซึ่งใช้ตัวควบคุมแบบ PID Zigler Nichols โดยมีการแสดงผลการควบคุมเป็นทั้งกราฟและตัวเลข เมื่อคลิ้กปุ่ม Start ที่หน้าต่าง Menu หลักแล้ว หน้าต่าง PID Zigler Nichols Controller นี้จะแสดงขึ้นมา (ซึ่งขึ้นอยู่กับตอนเลือก Controller) และมีส่วนประกอบต่างๆ ดังรูปที่4-5



รูปที่ 4-5 หน้าต่าง หน้าต่าง PID Zigler Nichols Controller

สำหรับรายละเอียดของปุ่มต่างๆ ที่กล่าวมาแล้วจะไม่ขอกล่าวถึงในที่นี้ จะขอกล่าวถึงในส่วนที่เพิ่มเติมเข้ามา

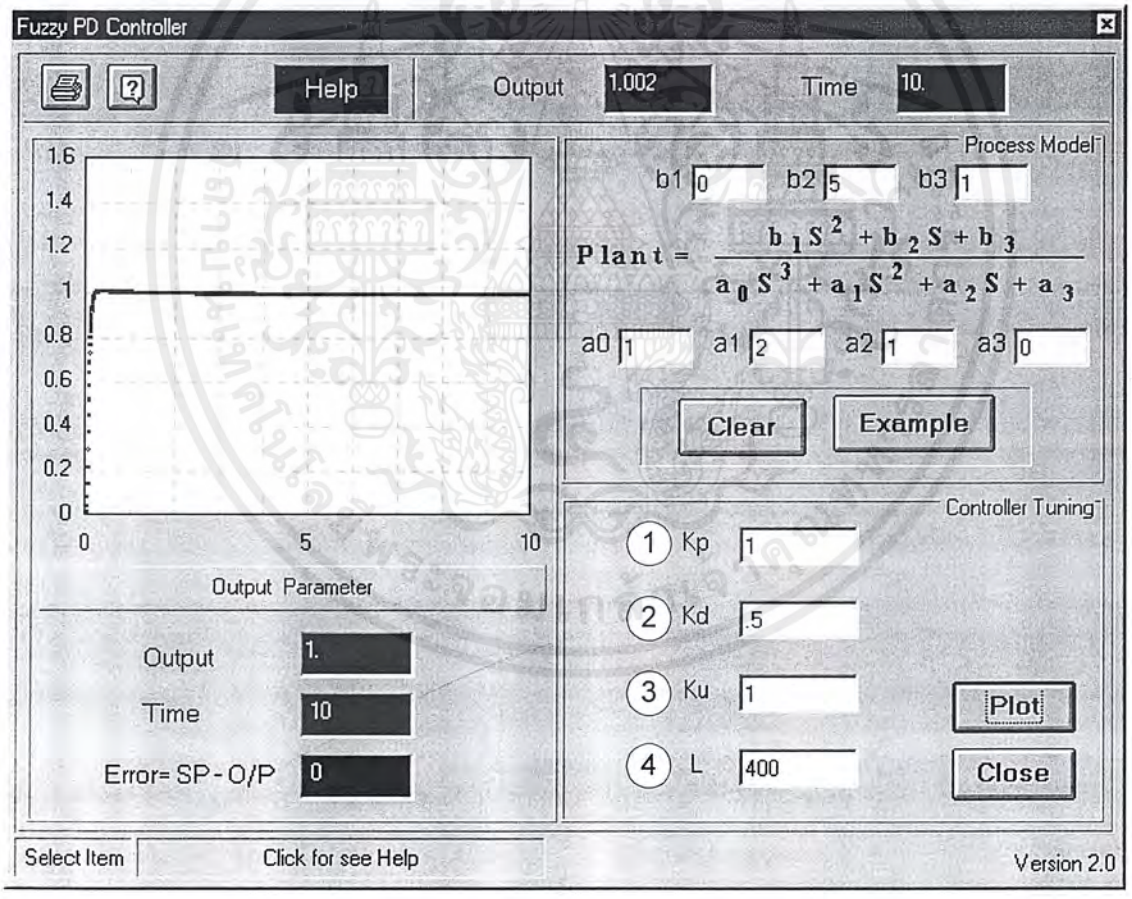
1. TextBox Output Kp สำหรับแสดงค่าเอาท์พุท Kp เป็นตัวเลข
2. TextBox Output Ti สำหรับแสดงค่าเอาท์พุท Ti เป็นตัวเลข
3. TextBox Output Td สำหรับแสดงค่าเอาท์พุท Td เป็นตัวเลข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวควบคุมแบบ PID Zigler Nichols Controller นี้ผู้ใช้ไม่ต้องใส่ค่าพารามิเตอร์ Kp, Ti, Td โดยผู้ใช้ใส่ค่า Process ก็พอ Controller แบบนี้จะคำนวณหาค่าพารามิเตอร์ Kp, Ti, Td ให้เอง และจะแสดงค่าพารามิเตอร์ Kp, Ti, Td ให้ด้วย

หน้าต่าง Fuzzy PD Controller

สำหรับหน้าต่าง Fuzzy PD Controller จะเป็นหน้าต่างที่ใช้สำหรับออกแบบระบบควบคุม และคำนวณหา Output Time Respons ซึ่งใช้ตัวควบคุมแบบ Fuzzy PD โดยมีการแสดงผลการควบคุมเป็นทั้งกราฟและตัวเลข เมื่อคลิ้กปุ่ม Start ที่หน้าต่าง Menu หลักแล้ว หน้าต่าง Fuzzy PD Controller นี้จะแสดงขึ้นมา (ซึ่งขึ้นอยู่กับตอนเลือก Controller) และมีส่วนประกอบต่างๆ ดังรูปที่ 4-6



รูปที่ 4-6 หน้าต่าง Fuzzy PD Controller

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับรายละเอียดของปุ่มต่างๆ ที่กล่าวมาแล้วจะไม่ขอกล่าวถึงในที่นี้ จะขอกล่าวถึงในส่วนที่เพิ่มเติมเข้ามา

1. TextBox Input Kp สำหรับใส่ค่า Input Parameter Kp

2. TextBox Input Kd สำหรับใส่ค่า Input Parameter Kd

3. TextBox Input Ku สำหรับใส่ค่า Input Parameter Ku

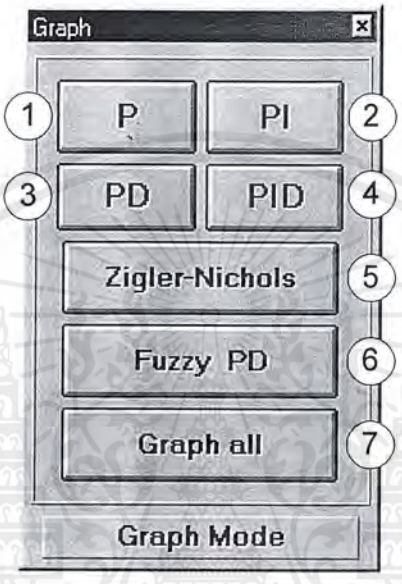
4. TextBox Input L สำหรับใส่ค่า Input Parameter L

5. ปุ่ม Graph จะเป็น Option แสดงผลเส้นกราฟหลายๆ เส้น เพื่อให้ผู้ใช้ ใช้เปรียบเทียบ Output เมื่อเปลี่ยนค่าพารามิเตอร์ต่างๆ เมื่อคลิ้กปุ่มนี้จะเรียกหน้าต่าง Graph ขึ้นมา ซึ่งจะมีชนิดของ Controllerให้เลือกอีกว่าต้องการเปรียบเทียบ Controller แบบใด ซึ่งจะกล่าวโดยละเอียดต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าต่าง Graph

สำหรับหน้าต่าง Graph จะเป็นหน้าต่างสำหรับเลือกการแสดงผลเพื่อเปรียบเทียบกราฟเมื่อเปลี่ยนค่าพารามิเตอร์ ของ Controller แบบต่างๆ เมื่อคลิ้กปุ่ม Graph ที่หน้าต่าง Menu หลักแล้ว หน้าต่าง Graph นี้จะแสดงขึ้นมา และมีส่วนประกอบต่างๆ ดังรูปที่ 4-7



รูปที่ 4-7 หน้าต่าง Graph

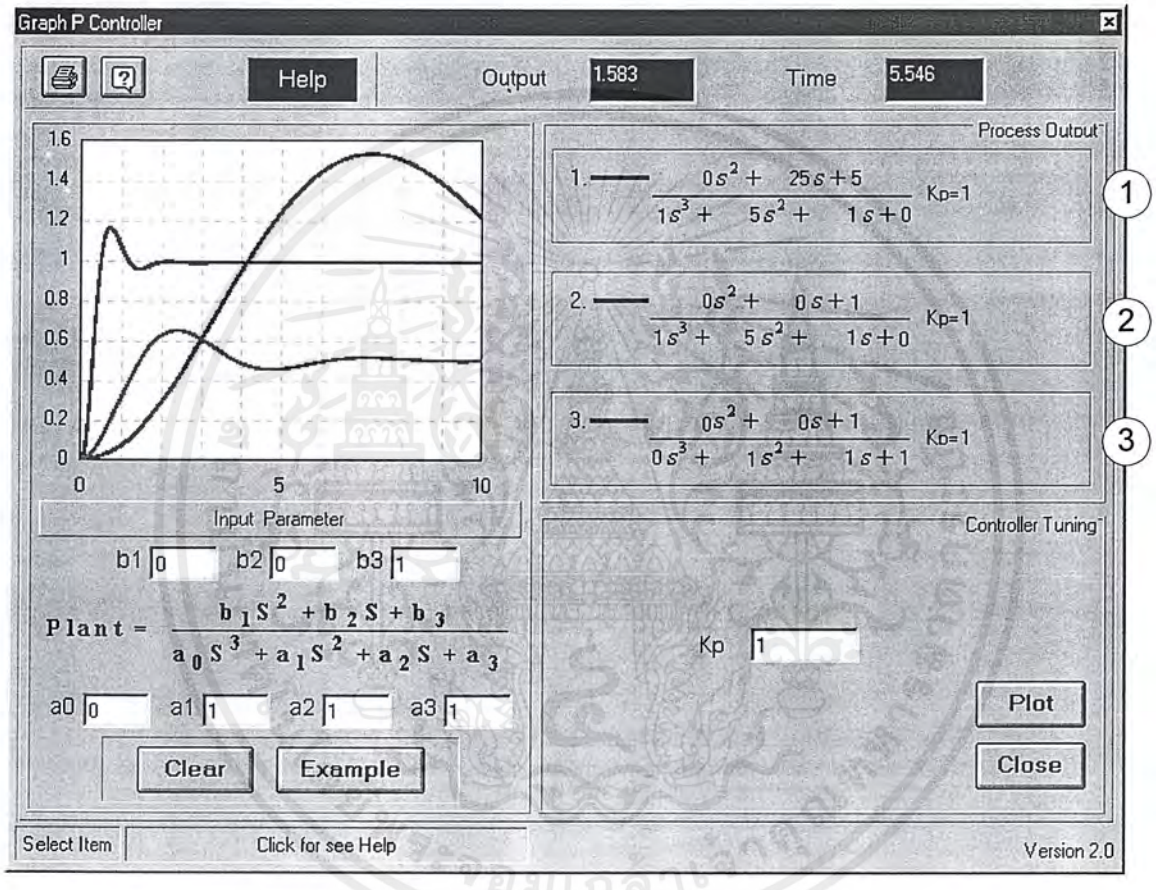
1. ปุ่ม P คลิ้กปุ่มนี้เพื่อเลือกเปรียบเทียบค่าพารามิเตอร์ที่มีผลต่อกราฟของตัวควบคุมแบบ P
2. ปุ่ม PI คลิ้กปุ่มนี้เพื่อเลือกเปรียบเทียบค่าพารามิเตอร์ที่มีผลต่อกราฟของตัวควบคุมแบบ PI
3. ปุ่ม PD คลิ้กปุ่มนี้เพื่อเลือกเปรียบเทียบค่าพารามิเตอร์ที่มีผลต่อกราฟของตัวควบคุมแบบ PD
4. ปุ่ม PID คลิ้กปุ่มนี้เพื่อเลือกเปรียบเทียบค่าพารามิเตอร์ที่มีผลต่อกราฟของตัวควบคุมแบบ PID
5. ปุ่ม Zigler Nichols คลิ้กปุ่มนี้เพื่อเลือกเปรียบเทียบค่าพารามิเตอร์ที่มีผลต่อกราฟของตัวควบคุมแบบ PID Zigler Nichols
6. ปุ่ม Fuzzy PD คลิ้กปุ่มนี้เพื่อเลือกเปรียบเทียบค่าพารามิเตอร์ที่มีผลต่อกราฟของตัวควบคุมแบบ Fuzzy PD
7. ปุ่ม Graph all คลิ้กปุ่มนี้เพื่อเลือกเปรียบเทียบค่าพารามิเตอร์ที่มีผลต่อกราฟของตัวตัวควบคุมทั้งหมด

หลังจากคลิ้กปุ่มเพื่อเลือกตัวควบคุมแล้ว โปรแกรมเรียกหน้าต่างของตัวควบคุมที่เลือกไว้ขึ้นมา เพื่อให้ผู้ใช้ ๆในการเปรียบเทียบต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าต่าง Graph P Controller

สำหรับหน้าต่าง Graph P Controller จะเป็นหน้าต่างที่ใช้สำหรับเปรียบเทียบกราฟเอาท์พุท เมื่อเปลี่ยนค่าพารามิเตอร์ต่างๆ ของ P Controller เช่นค่าพารามิเตอร์ Kp หรือ ผลจากการเปลี่ยน Process เมื่อคลิ้กปุ่ม P Controller ที่หน้าต่าง Graph แล้ว หน้าต่าง Graph P Controller นี้จะแสดงขึ้นมา และมีส่วนประกอบต่างๆ ดังรูปที่ 4-3



รูปที่ 4-8 หน้าต่าง Graph P Controller

สำหรับรายละเอียดของปุ่มต่างๆ ที่กล่าวมาแล้วจะไม่ขอกล่าวถึงในที่นี้ จะขอกล่าวถึงในส่วนที่เพิ่มเติมเข้ามา

1. กรอบ Process Output 1 สำหรับแสดงค่า Process Output และค่าพารามิเตอร์ของระบบที่ 1
2. กรอบ Process Output 2 สำหรับแสดงค่า Process Output และค่าพารามิเตอร์ของระบบที่ 2
3. กรอบ Process Output 3 สำหรับแสดงค่า Process Output และค่าพารามิเตอร์ของระบบที่ 3

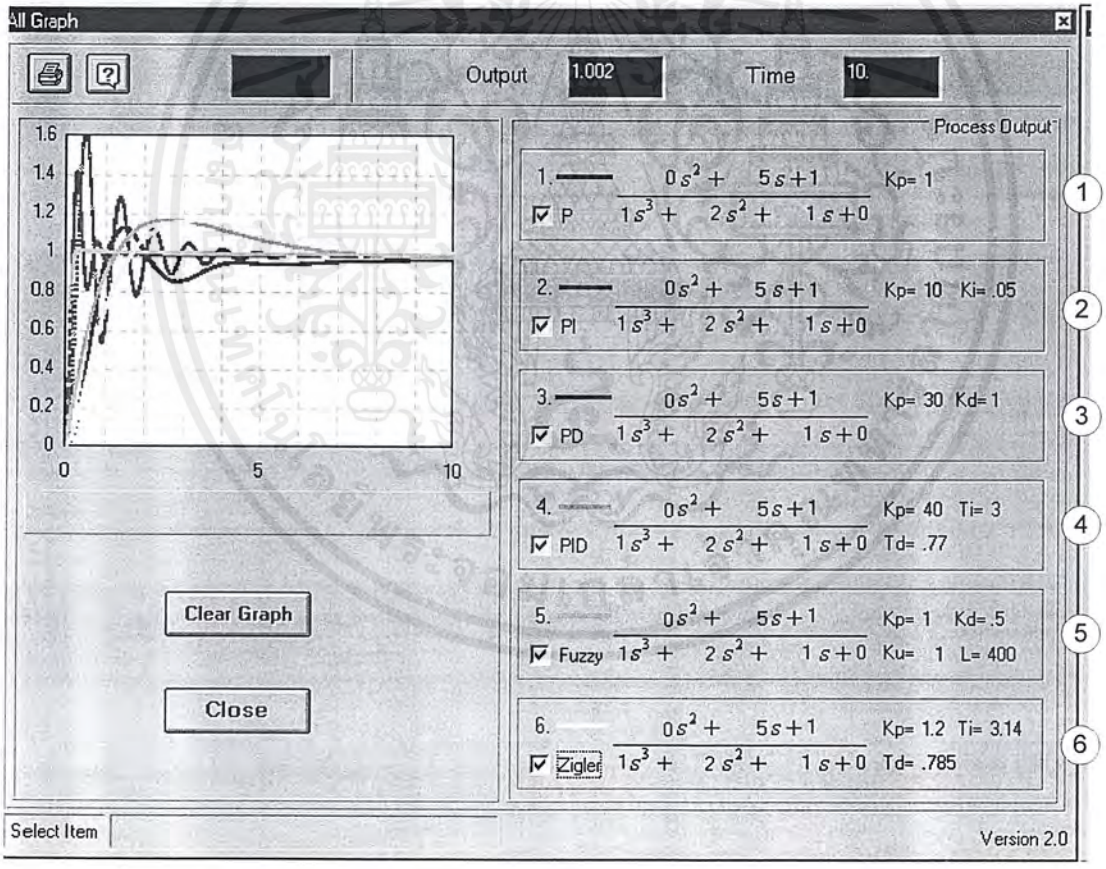
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าต่าง Graph ของ Controller แบบอื่น ๆ

สำหรับหน้าต่าง Graph ชอง Controller แบบอื่นๆที่มีลักษณะเหมือนกันกับแบบ P Controller ก็คือ แบบPI Controller, PD Controller, PID Controller, PID Zigler Nichols และแบบ Fuzzy PD จะมีข้อแตกต่างกันตรงที่ อินพุทพารามิเตอร์ของแต่ละ Controller จะไม่เหมือนกันเช่น PI Controller จะมีอินพุทพารามิเตอร์เป็น Kp กับ Ki เป็นต้น จึงขอไม่กล่าวถึงในที่นี้

หน้าต่าง Graph all

สำหรับหน้าต่าง Graph all จะเป็นหน้าต่างสำหรับเปรียบเทียบค่าพารามิเตอร์ที่มีผลต่อกราฟของตัวตัวควบคุมทั้งหมด เมื่อคลิ้กปุ่ม Graph all ที่หน้าต่าง Graph แล้ว หน้าต่าง Graph all นี้จะแสดงขึ้นมา และมีส่วนประกอบต่างๆ ดังรูปที่ 4-9



รูปที่ 4-9 หน้าต่าง Graph all

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับรายละเอียดของปุ่มต่างๆ ที่กล่าวมาแล้วจะไม่ขอกล่าวถึงในที่นี้ จะขอกล่าวถึงในส่วนที่เพิ่มเติมเข้ามา

1. กรอบ Process Output 1 สำหรับแสดงค่า Process Output และค่าพารามิเตอร์ของ P Controller

2. กรอบ Process Output 2 สำหรับแสดงค่า Process Output และค่าพารามิเตอร์ของ PI Controller

3. กรอบ Process Output 3 สำหรับแสดงค่า Process Output และค่าพารามิเตอร์ของ PD Controller

4. กรอบ Process Output 4 สำหรับแสดงค่า Process Output และค่าพารามิเตอร์ของ PID Controller

5. กรอบ Process Output 5 สำหรับแสดงค่า Process Output และค่าพารามิเตอร์ของ Zigler Nichols

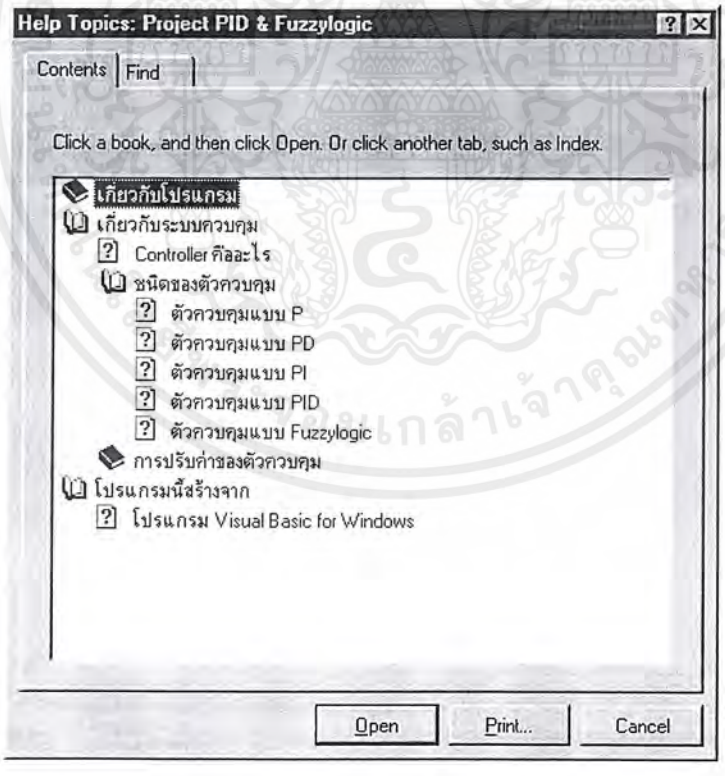
6. กรอบ Process Output 6 สำหรับแสดงค่า Process Output และค่าพารามิเตอร์ของ Fuzzy PD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของ Help File

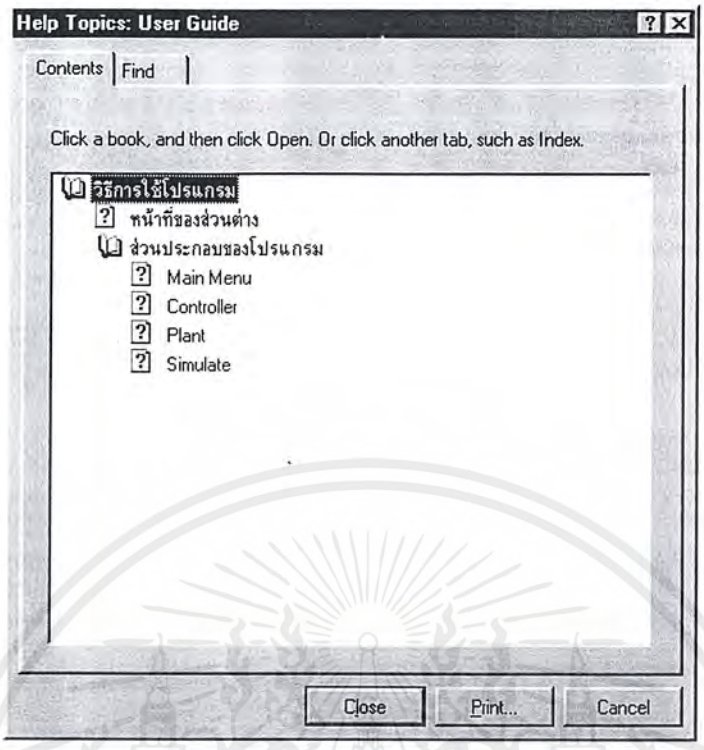
ส่วนของ Help File นี้จัดทำไว้เพื่อช่วยเหลือผู้ใช้โปรแกรม โดยมีข้อมูลตั้งแต่ทฤษฎีพื้นฐาน การใช้งานแต่ละ Menu ตลอดหน้าที่ของปุ่มต่าง โดยในการเขียน Help File นี้ได้มีการนำเอาวิธีการหลายวิธีเพื่อนำมาทดลองในการเขียน Help ของโปรแกรมนี้โดยในตอนแรกนั้นเราได้ใช้วิธีแบบดั้งเดิมในการเขียน Help ก็คือการเขียนด้วย Microsoft Word โดยพิมพ์ Help ลงใน Microsoft Word แล้วทำการ Save เป็น ".RTF" (Rich Text Format) แล้วนำไป Compile ด้วยโปรแกรม Win Help Compiler หลังจากนั้นจะได้ File ".Hlp" หรือ Win Help File นั่นเองแต่หลังจากที่ได้ทดลองใช้วิธีนี้แล้วพบว่าเกิดข้อผิดพลาดที่ตัว Help File หลายตำแหน่งอีกทั้งการเขียนก็ใช้เวลาข้อนข้างมากจึงได้เปลี่ยนวิธีในการเขียนใหม่โดยใช้ Soft Ware เฉพาะด้านมาใช้ในการเขียน ซึ่งในการเขียน Help ครั้งนี้เราได้ใช้โปรแกรมชื่อ Help Pad เขียน แต่ในขั้นตอนที่จะ Compile เป็น ".Hlp" หรือ Win Help File นั้นยังคงต้องใช้ Win Help Compiler เช่นเดิมโดย Help ที่เราเขียนขึ้นมานั้นจะมีลักษณะดังนี้

ในส่วนของ Contents ซึ่งเป็นเหมือนกับ สารบัน ของ Help



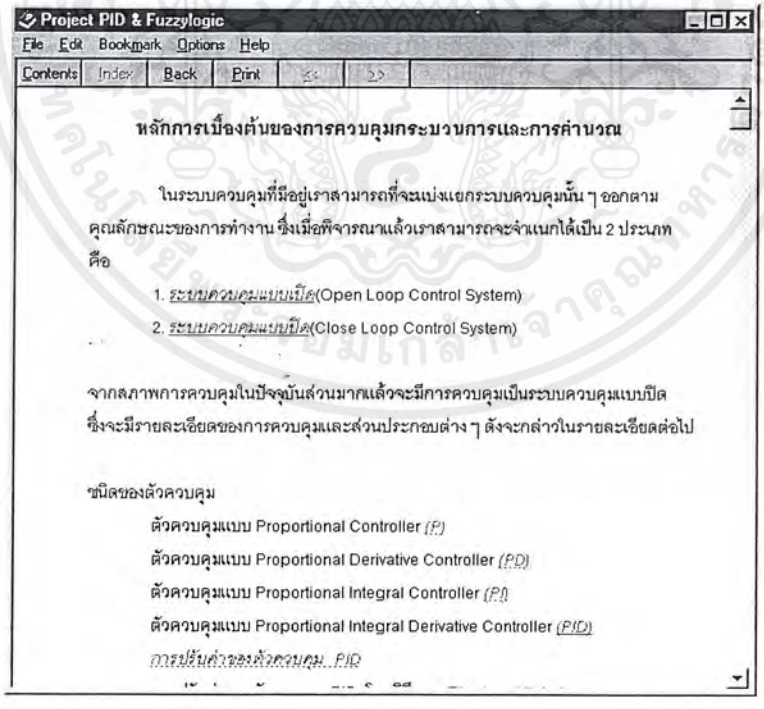
รูปที่ 4-10 แสดงหน้าต่าง Help Contents

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



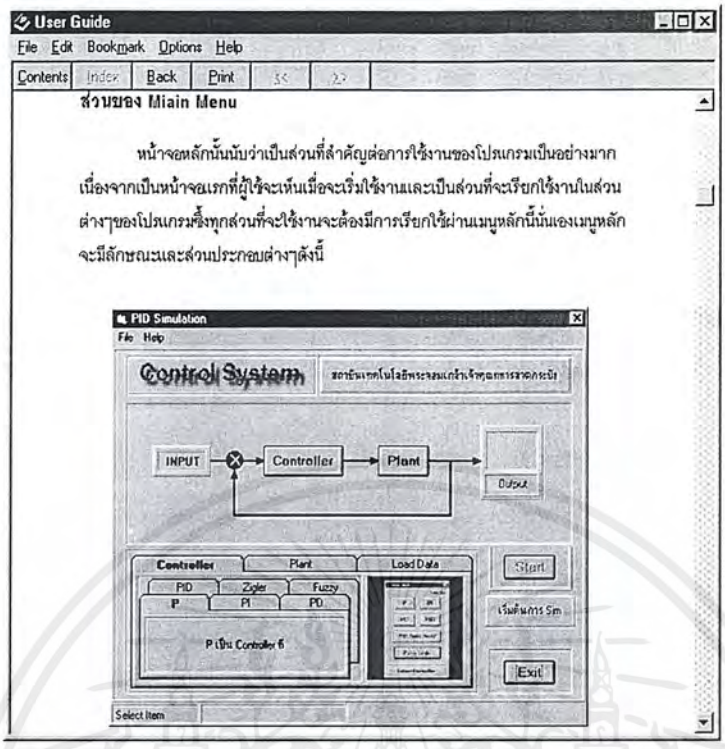
รูปที่ 4-11 แสดงหน้าต่าง Help Contents

และในส่วนของ Topic ซึ่งเป็นส่วนรายระเอียดของแต่ละหัวข้อ

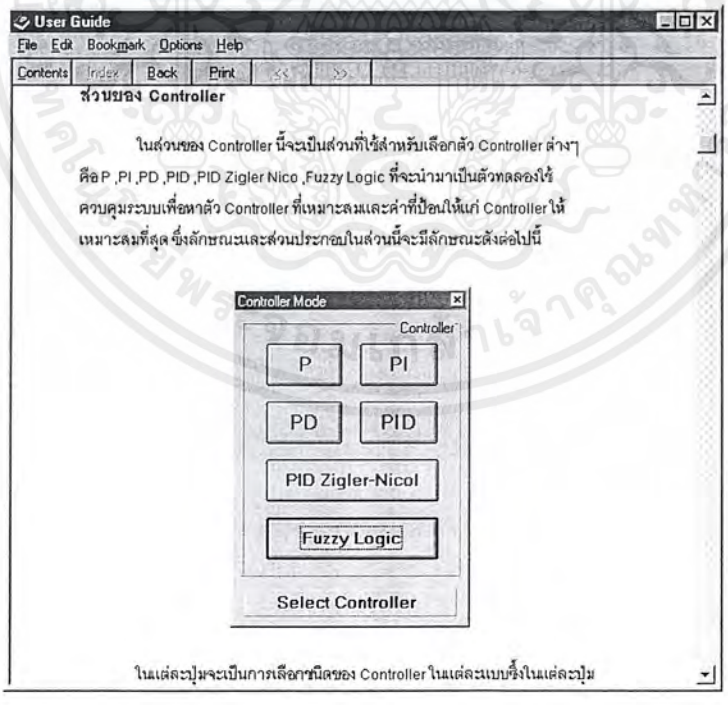


รูปที่ 4-12 แสดงหน้าต่าง Help Topic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-13 แสดงหน้าต่าง Help Topic



รูปที่ 4-14 แสดงหน้าต่าง Help Topic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การทดลองและผลการทดลอง

สำหรับในบทนี้จะเป็นการทดลองออกแบบระบบควบคุม ซึ่งจะทำการเปรียบเทียบผลของการควบคุมจาก Controller แบบต่างๆ โดยเปรียบเทียบจากการทดลองเปลี่ยนค่าพารามิเตอร์ต่างๆ และจากการเปลี่ยน Process เป็นแบบต่างๆ ด้วย และได้มีการสรุปและวิจารณ์ผลการทดลองเพื่อเป็นแนวทางในการออกแบบระบบควบคุมต่อไป .

การทดลองที่1

ตัวควบคุมแบบ P

จุดประสงค์

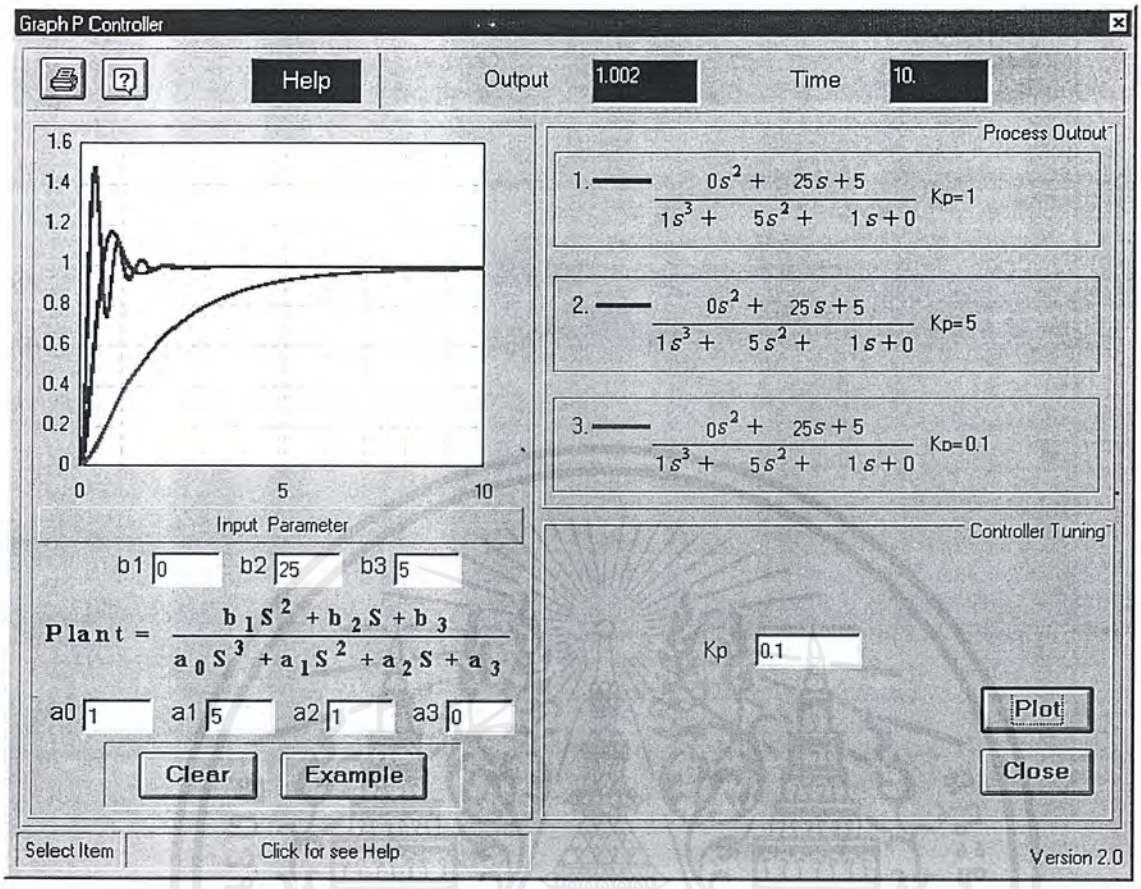
1. ศึกษาถึงผลของค่า Parameter Kp ที่มีต่อระบบควบคุม
2. ศึกษาถึงเสถียรภาพของระบบ ของตัวควบคุมแบบ P Controller ที่มีต่อ Process แบบต่างๆ

ลำดับขั้นการทดลอง

1. เลือกตัวควบคุมแบบ P
2. ใส่ค่า Process เป็น $\frac{25s+5}{s^3+5s^2+s}$
3. ใส่ค่า Kp = 1 แล้ว Simulate ดูกราฟ Output
4. ทดลองเปลี่ยนค่า Kp เป็นค่าต่างๆ เพื่อดู Output ของระบบควบคุม
5. ทดลองเปลี่ยน Process เพื่อดู Output ของระบบควบคุม

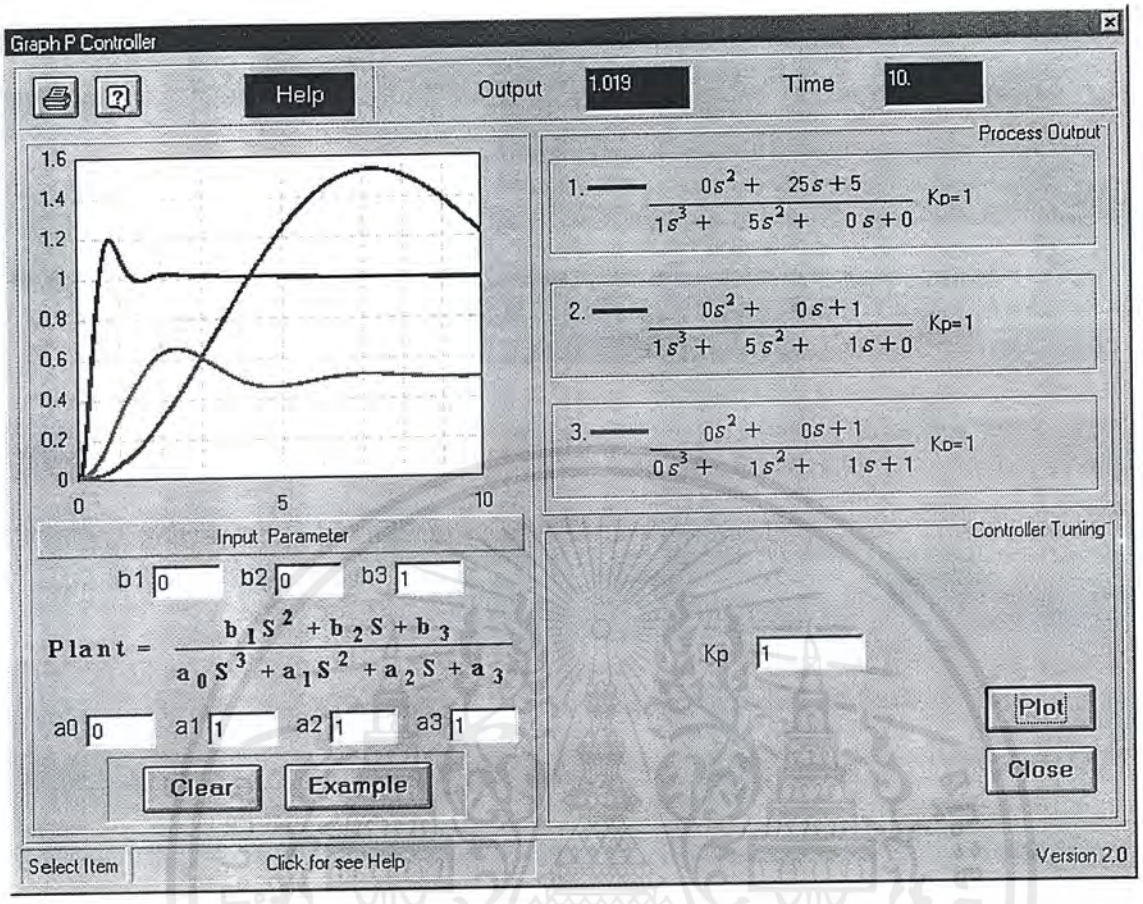
ผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-1 แสดงกราฟเมื่อปรับค่า Kp

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-2 แสดงกราฟเอาท์พุทเมื่อเปลี่ยน Process

สรุปและวิจารณ์ผลการทดลอง

จากการทดลองเปลี่ยนค่าพารามิเตอร์ Kp ปรากฏว่าค่า Kp ควรมีค่าที่เหมาะสม โดยถ้าค่า Kp มากไปจะทำให้เอาท์พุทเกิด over shoot มากและถ้าค่า Kp น้อยไปเอาท์พุทก็จะใช้เวลาเข้าสู่สภาวะคงที่นาน ซึ่งค่า Kp จะไปเพิ่ม Gain ให้กับระบบ

และจากการทดลองเปลี่ยนค่า Process ปรากฏว่าค่า Process ที่มีตัวเศษมีค่ามาก เช่น $\frac{25s+5}{s^3+5s^2+s+1}$ จะเสถียรดีกว่า Process ที่ตัวเศษมีกำลังน้อยเช่น $\frac{1}{s^2+s+1}$ เนื่องจากตัวเศษจะไปเพิ่ม Zeroให้กับระบบทำให้ระบบดีขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 2

ตัวควบคุมแบบ PI

จุดประสงค์

1. ศึกษาถึงผลของค่า Parameter Kp, Ki ที่มีต่อระบบควบคุม

2. ศึกษาถึงเสถียรภาพของระบบ ของตัวควบคุมแบบ PI Controller ที่มีต่อ Process แบบต่างๆ

ลำดับขั้นการทดลอง

1. เลือกตัวควบคุมแบบ PI

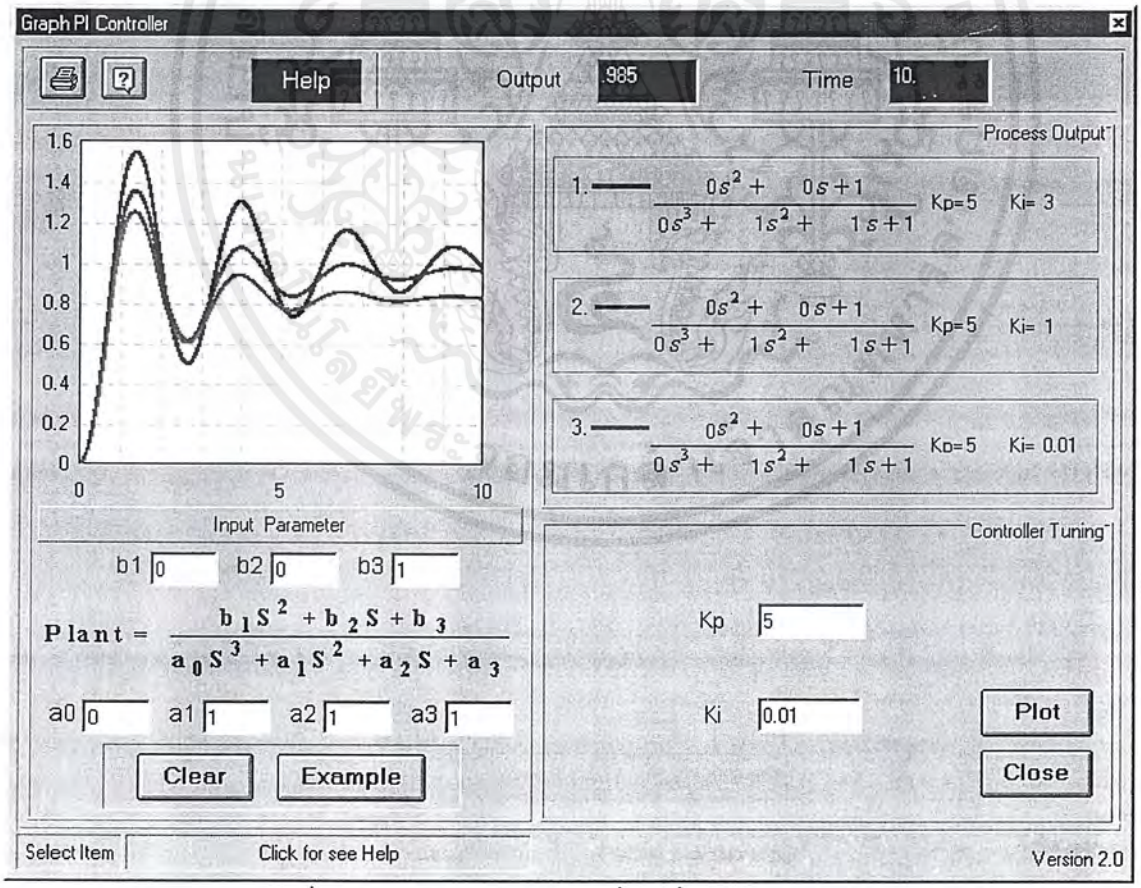
2. ใส่ค่า Process เป็น $\frac{1}{s^2+s+1}$

3. ใส่ค่า Kp = 5 , Ki = 3 แล้ว Simulate ดูกราฟ Output

4. ทดลองเปลี่ยนค่า Kp, Ki เป็นค่าต่างๆ เพื่อดู Output ของระบบควบคุม

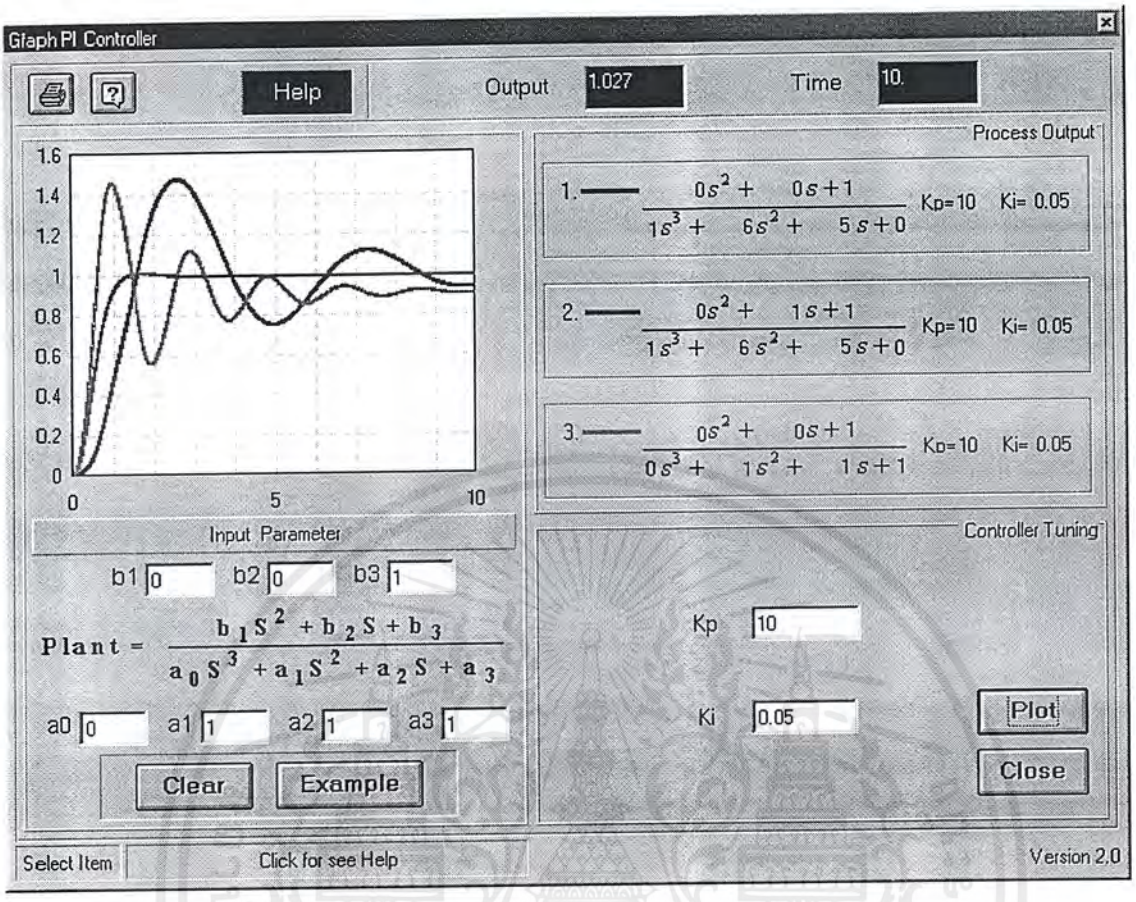
5.ทดลองเปลี่ยน Process เพื่อดู Output ของระบบควบคุม

ผลการทดลอง



รูปที่ 5-3 แสดงกราฟเอาท์พุทเมื่อเปลี่ยนค่า Kp และ Ki

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-4 แสดงกราฟเอาท์พุทเมื่อเปลี่ยน Process

สรุปและวิจารณ์ผลการทดลอง

จากการทดลองเปลี่ยนค่าพารามิเตอร์ Kp , Ki ปรากฏว่าค่า Ki ควรมีค่าที่เหมาะสม โดยถ้าค่า Ki มากไปจะทำให้เอาท์พุทเกิด over shoot มาก และถ้าค่า Ki น้อยไปเอาท์พุทก็จะมี Error มาก ซึ่งเป็นผลมาจาก ตัว Ki จะไปเพิ่ม Pole ให้กับระบบ

และจากการทดลองเปลี่ยนค่า Process ปรากฏว่าค่า Process ที่มีตัวเศษมีค่ามาก เช่น $\frac{25s+5}{s^3+5s^2+s+1}$ จะเสถียรดีกว่า คือเอาท์พุทสามารถเข้าสู่ Set point ได้เร็วและ Overshoot น้อย ส่วน Process ที่ตัวเศษมีกำลังน้อยเช่น $\frac{1}{s^2+s+1}$ เอาท์พุทจะมี Overshoot มาก และใช้เวลาในการเข้าสู่สภาวะคงที่นาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 3

ตัวควบคุมแบบ PD

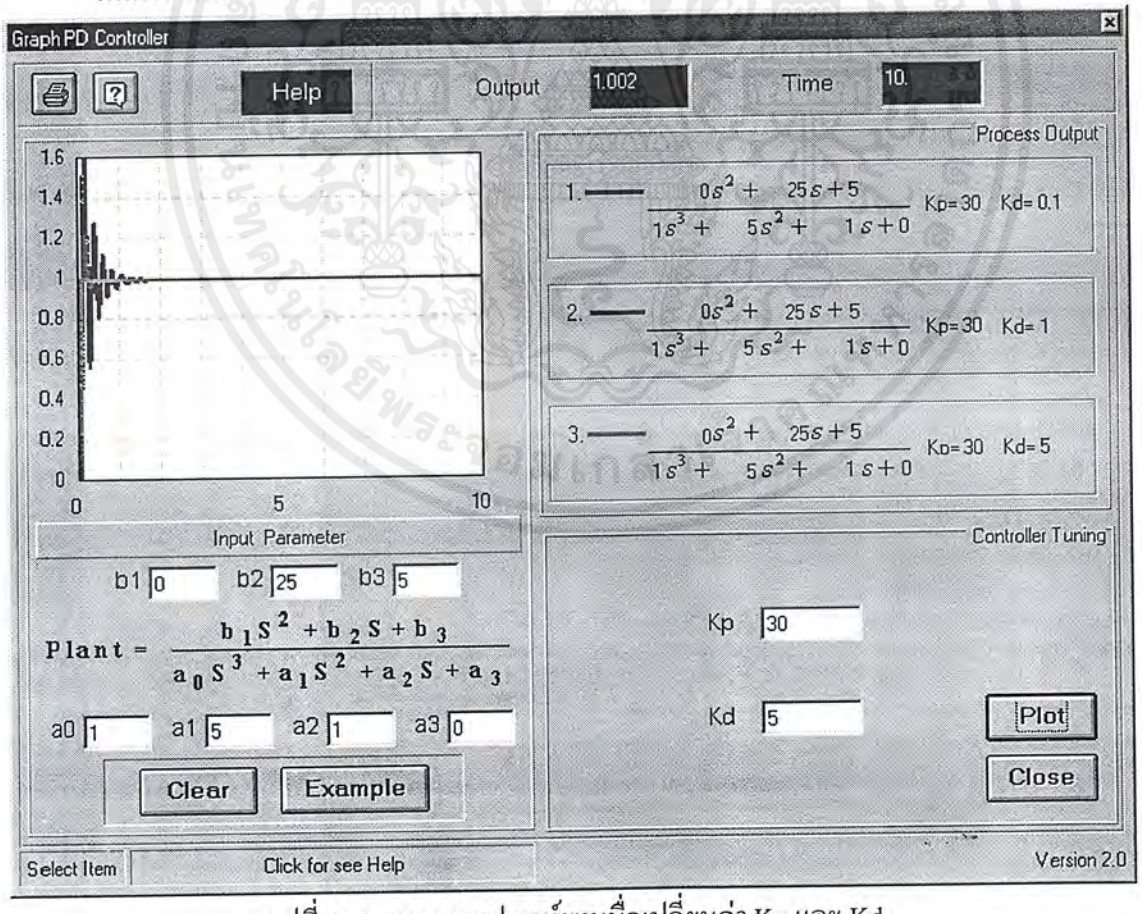
จุดประสงค์

1. ศึกษาถึงผลของค่า Parameter Kp, Kd ที่มีต่อระบบควบคุม
2. ศึกษาถึงเสถียรภาพของระบบ ของตัวควบคุมแบบ PD Controller ที่มีต่อ Process แบบต่างๆ

ลำดับขั้นการทดลอง

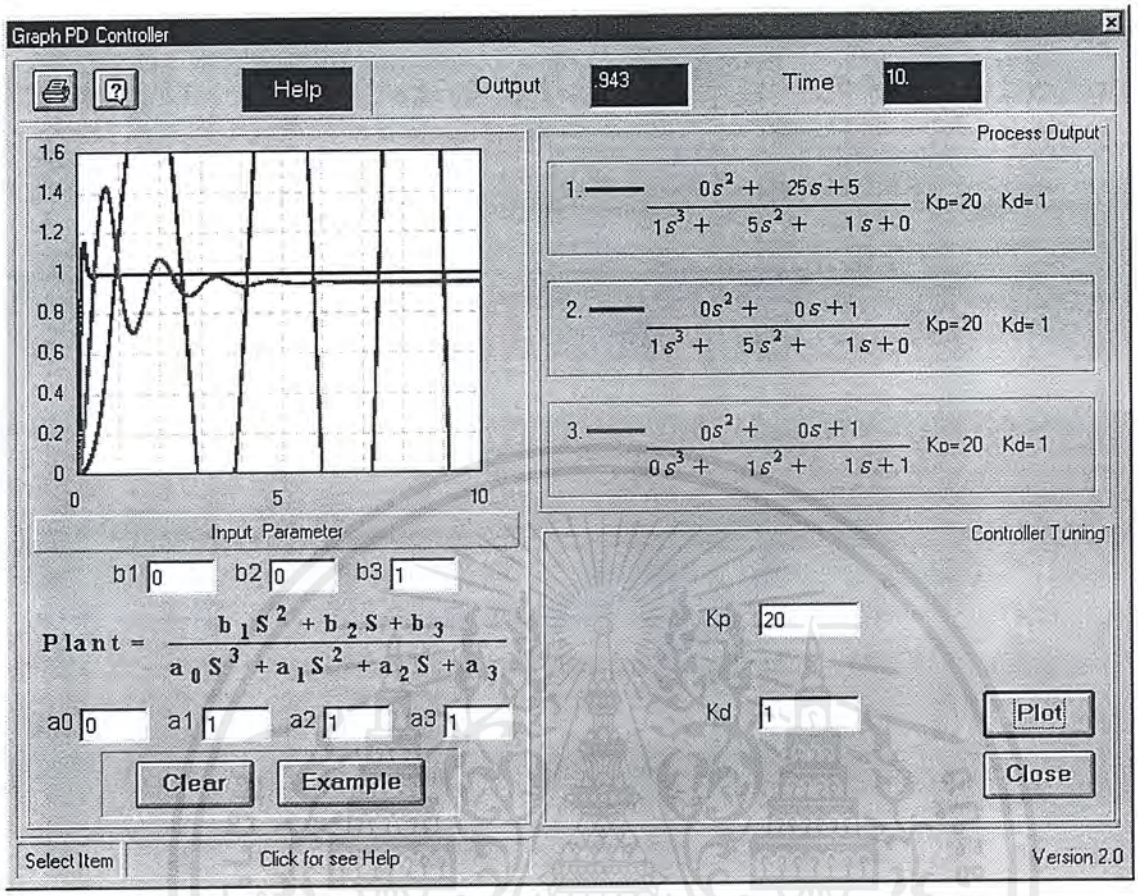
1. เลือกตัวควบคุมแบบ PD
2. ใส่ค่า Process เป็น $\frac{25s+5}{s^3+5s^2+s}$
3. ใส่ค่า Kp = 30 , Kd = 0.1 แล้ว Simulate ดูกราฟ Output
4. ทดลองเปลี่ยนค่า Kp, Kd เป็นค่าต่างๆ เพื่อดู Output ของระบบควบคุม
5. ทดลองเปลี่ยน Process เพื่อดู Output ของระบบควบคุม

ผลการทดลอง



รูปที่ 5-5 แสดงกราฟเอาท์พุทเมื่อเปลี่ยนค่า Kp และ Kd

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-6 แสดงกราฟเอาท์พุทเมื่อเปลี่ยน Process

สรุปและวิจารณ์ผลการทดลอง

จากการทดลองเปลี่ยนค่าพารามิเตอร์ Kp,Kd ปรากฏว่าค่า Kd ควรมีค่ามากๆ ซึ่งจะทำให้ลด Overshoot ได้มากและมีช่วงเวลาไต่ขึ้นน้อย ทำให้เอาท์พุทของระบบเสถียรได้เร็วขึ้น ซึ่งเป็นผลมาจาก ตัว Kd จะไปเพิ่ม Zero ให้กับระบบ

และจากการทดลองเปลี่ยนค่า Process ปรากฏว่า Process ที่มีตัวเศษมีกำลังมาก เช่น $\frac{25s+5}{s^3+5s^2+s}$ จะเสถียรดีกว่า ส่วน Process ที่มีตัวเศษเป็น 1 และตัวส่วนมี Order สูงๆ เช่น $\frac{1}{S^3+5S^2+S}$ เอาท์พุทจะเกิดการแกว่งเป็นออสลิเลท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 4

ตัวควบคุมแบบ PID

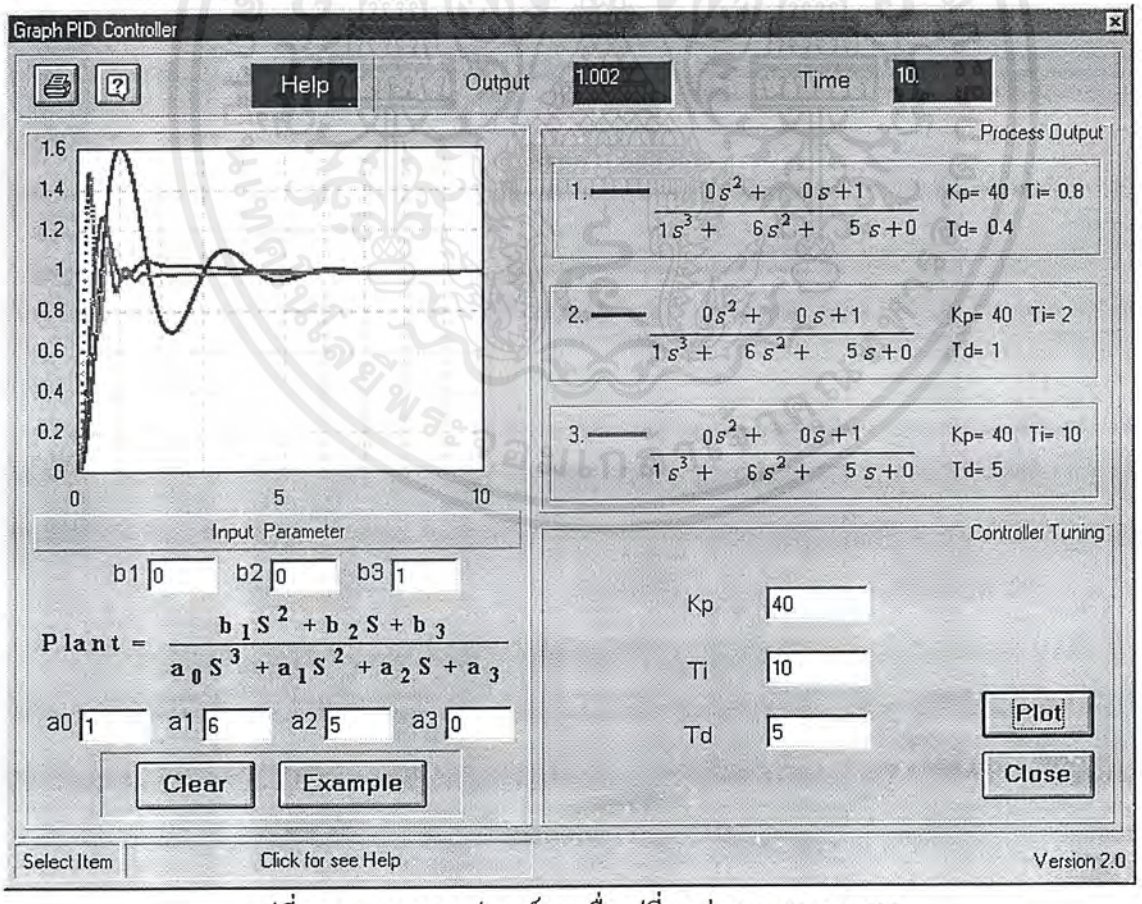
จุดประสงค์

1. ศึกษาถึงผลของค่า Parameter Kp, Ti, Td ที่มีต่อระบบควบคุม
2. ศึกษาถึงเสถียรภาพของระบบ ของตัวควบคุมแบบ PID Controller ที่มีต่อ Process แบบต่างๆ

ลำดับขั้นการทดลอง

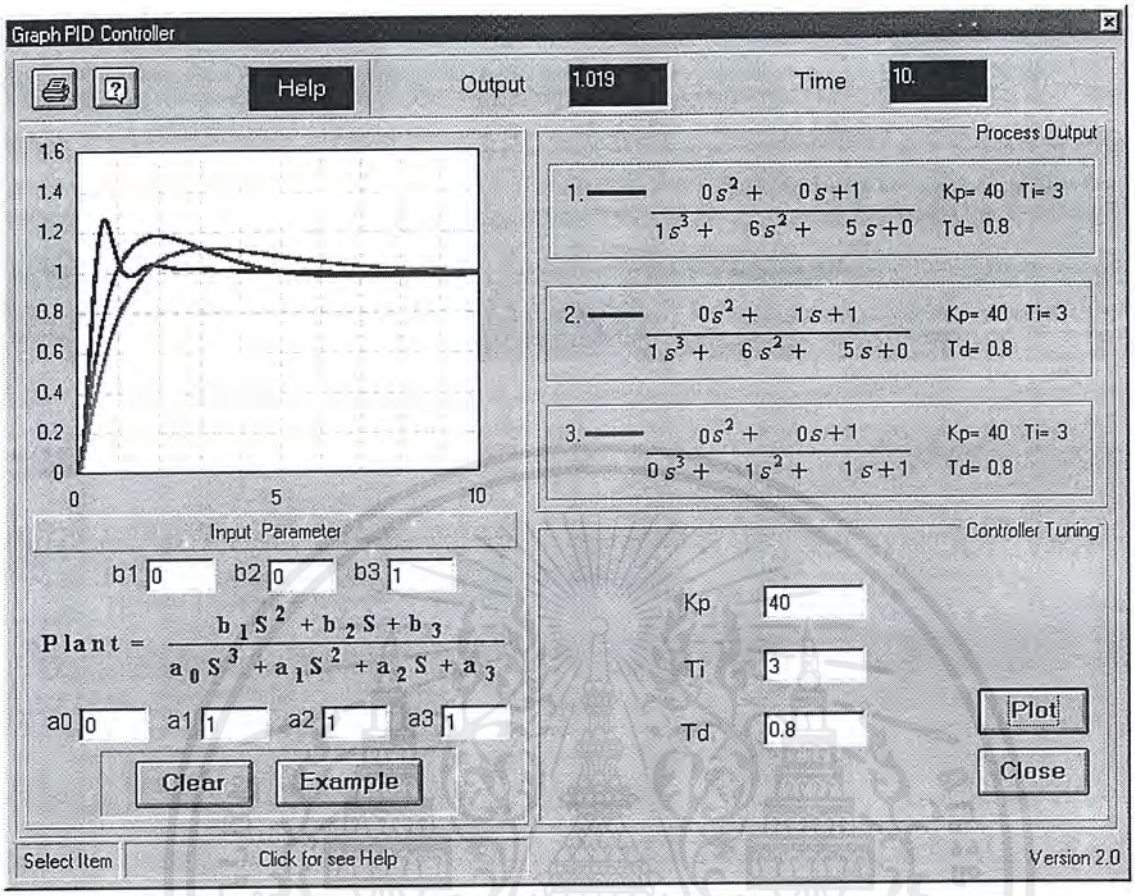
1. เลือกตัวควบคุมแบบ PID
2. ใส่ค่า Process เป็น $\frac{25s+5}{s^3+5s^2+s}$
3. ใส่ค่า Kp = 30 , Kd = 0.1 แล้ว Simulate ดูกราฟ Output
4. ทดลองเปลี่ยนค่า Kp, Kd เป็นค่าต่างๆ เพื่อดู Output ของระบบควบคุม
5. ทดลองเปลี่ยน Process เพื่อดู Output ของระบบควบคุม

ผลการทดลอง



รูปที่ 5-7 แสดงกราฟเอาท์พุทเมื่อเปลี่ยนค่า Kp ,Ti และ Td

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-8 แสดงกราฟเอาท์พุทเมื่อเปลี่ยน Process

สรุปและวิจารณ์ผลการทดลอง

จากการทดลองเปลี่ยนค่าพารามิเตอร์ Kp, Ti, Td ปรากฏว่าค่า Ti และ Td ควรมีค่าที่เหมาะสมโดยให้มีค่ามากจะดีกว่ามีค่าน้อยแต่ไม่ควรมากเกินไป ถ้ามีค่ามาก เอาท์พุทจะเกิด Overshoot มาก แต่เขาสู่สภาวะเสถียรได้เร็ว แต่ถ้ามีค่าน้อยเกินไป ก็ยังคงทำให้ Overshoot สูงขึ้นไปอีก และเข้าสู่สภาวะเสถียรได้ช้าลงไปอีก และจะมีการแก่วงมากด้วย ขึ้น ซึ่งเป็นผลมาจาก ตัว Td จะไปเพิ่ม Zero ให้กับระบบ และตัว Ti จะไปเพิ่ม Pole ให้กับระบบ

และจากการทดลองเปลี่ยนแปลงค่า Process ปรากฏว่าเอาท์พุทก็ได้ไม่ต่างกันมากนัก ไม่ว่าจะเป็น Process ที่มี Order ต่ำหรือสูง ตัวควบคุมแบบ PID ก็ยังคงสามารถควบคุมได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 5

ตัวควบคุมแบบ PID Zigler Nichols

จุดประสงค์

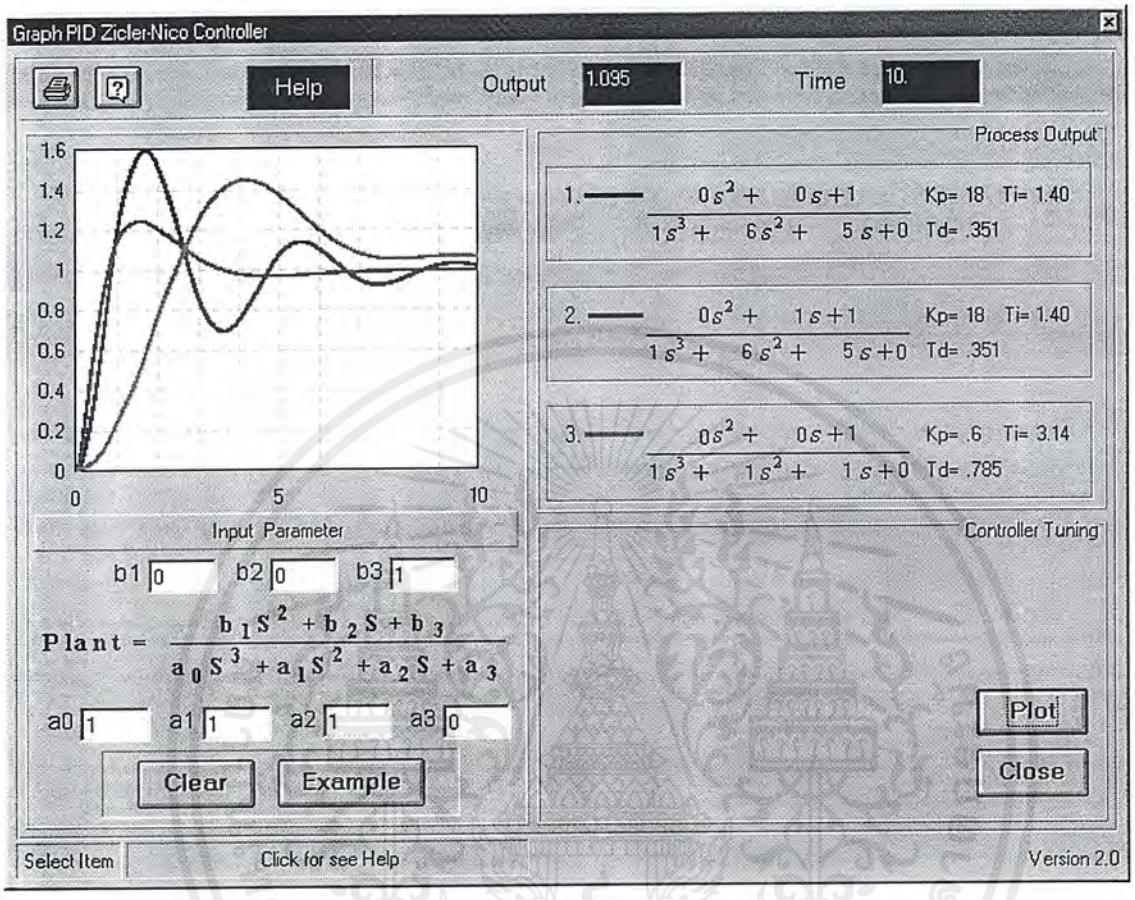
1. ศึกษาถึงเสถียรภาพของระบบ ของตัวควบคุมแบบ PID Zigler Nichols Controller ที่มีต่อ Processแบบต่างๆ

ลำดับขั้นการทดลอง

1. เลือกตัวควบคุมแบบ PID Zigler Nichols
2. ใส่ค่า Process เป็น $\frac{25s+5}{s^3+5s^2+s}$
3. Simulate ดูกราฟ Output
4. ทดลองเปลี่ยน Process เพื่อดู Output ของระบบควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง



รูปที่ 5-9 แสดงกราฟเอาท์พุทเมื่อเปลี่ยน Process

สรุปและวิจารณ์ผลการทดลอง

จากการทดลองเปลี่ยน Process ปรากฏว่าตัวควบคุมแบบนี้ สามารถจูนค่า Kp,Ti,Td ได้ดีพอสมควร ระบบสามารถเข้าสู่สภาวะเสถียรได้ แต่ถ้าต้องการให้เอาท์พุทดีกว่านี้ก็ต้องปรับแต่งค่าพารามิเตอร์ Kp,Ti,Td ต่อไปอีกจึงจะได้เอาท์พุทที่มีเสถียรภาพมากกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 6

ตัวควบคุมแบบ Fuzzy PD

จุดประสงค์

1. ศึกษาถึงผลของค่า Parameter Kp, Kd, Ku, L ที่มีต่อระบบควบคุม

2. ศึกษาถึงเสถียรภาพของระบบ ของตัวควบคุมแบบ Fuzzy PD Controller ที่มีต่อ Process แบบต่างๆ

ลำดับขั้นการทดลอง

1. เลือกตัวควบคุมแบบ Fuzzy

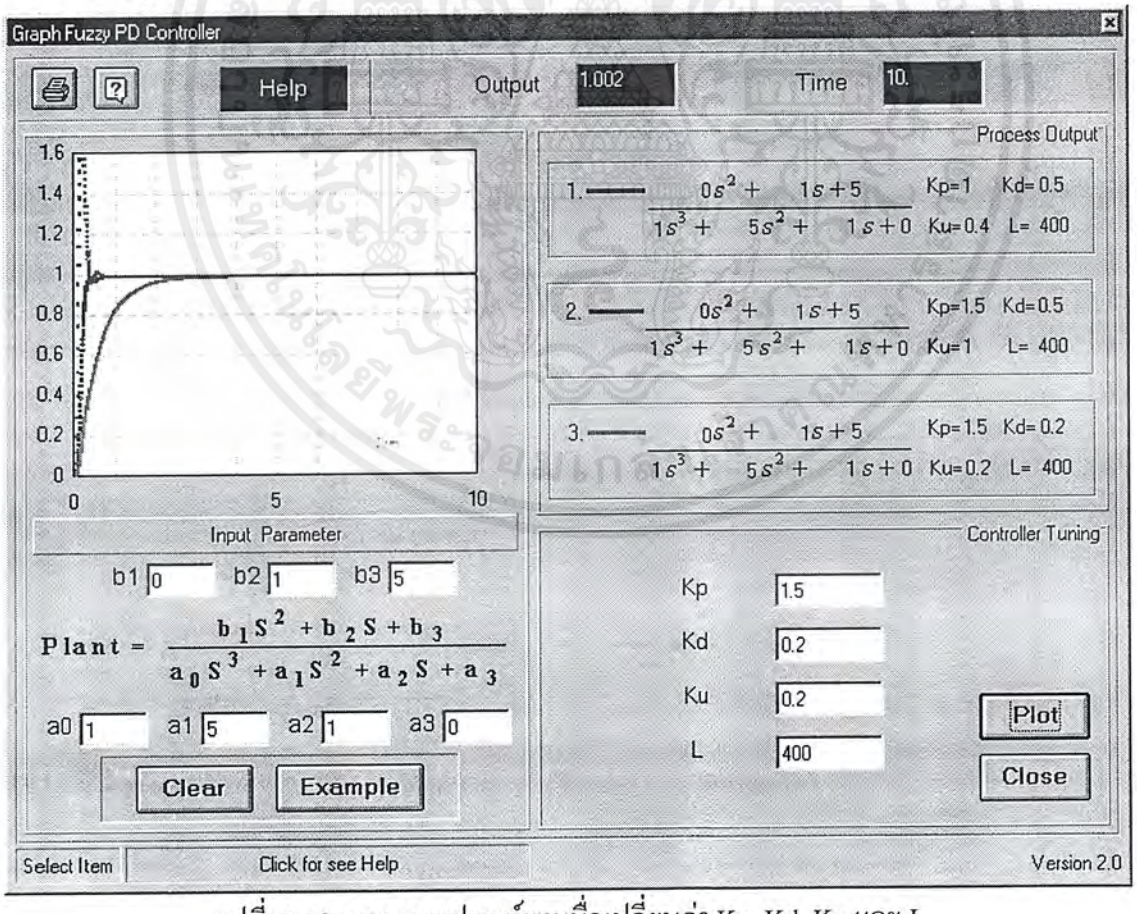
2. ใส่ค่า Process เป็น $\frac{25s+5}{s^3+5s^2+s}$

3. ใส่ค่า Kp = 30 , Kd = 0.1 แล้ว Simulate ดูกราฟ Output

4. ทดลองเปลี่ยนค่า Kp, Kd เป็นค่าต่างๆ เพื่อดู Output ของระบบควบคุม

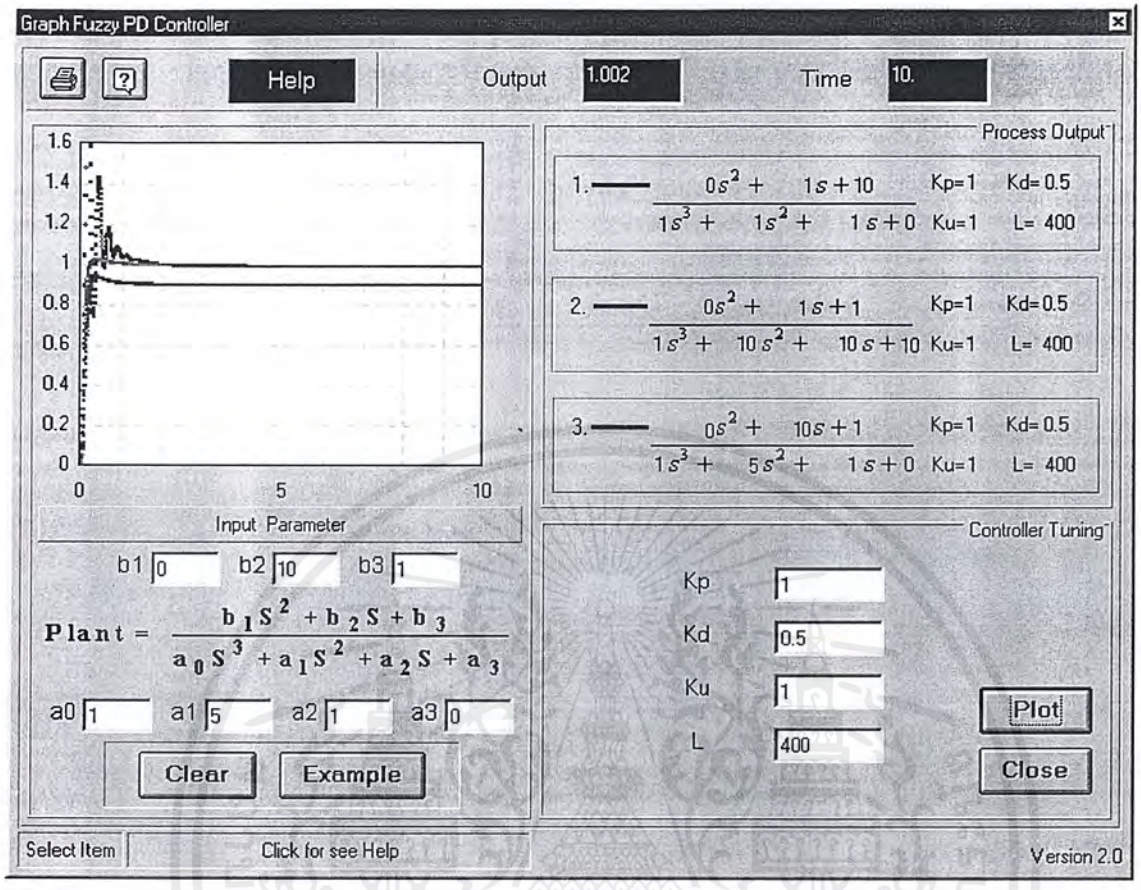
5.ทดลองเปลี่ยน Process เพื่อดู Output ของระบบควบคุม

ผลการทดลอง



รูปที่ 5-10 แสดงกราฟเอาท์พุทเมื่อเปลี่ยนค่า Kp, Kd, Ku และ L

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-11 แสดงกราฟเอาท์พุทเมื่อเปลี่ยน Process

สรุปและวิจารณ์ผลการทดลอง

Fuzzy PD Control

จากการทดลองเปลี่ยน Process ปรากฏว่า ค่า Kd ควรมีค่าน้อยๆ จึงจะทำให้ Overshoot น้อย ค่า Ku ควรอยู่ประมาณ 1 ถ้าค่าน้อยเกินไป ช่วงเวลาไต่ขึ้นจะนาน ส่วน L ไม่ค่อยมีผลต่อเอาท์พุทมากนัก โดยขึ้นอยู่กับการกำหนด Control Rule base

และจากการทดลองเปลี่ยน Process ปรากฏว่า ตัวควบคุมแบบนี้ สามารถควบคุมได้ดีในหลายๆ Process ส่วนในใหญ่ขึ้นอยู่กับค่าตัวเศษของ Process ถ้า b3 มีค่ามาก จะ Overshoot มาก ถ้า b2 มีค่ามาก จะเกิด Error มาก แต่ระบบโดยรวมจะดีกว่าในทุกๆตัวควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
สรุปและวิจารณ์ผลการทดลอง

1. P Controller

ตัวควบคุมแบบนี้ เหมาะกับ Process ที่ตัวเศษมีค่ามาก เช่น $\frac{25s+5}{s^3+5s^2+s+1}$ ระบบจะเสถียรดีกว่า Process ที่ตัวเศษมีกำลังน้อยเช่น $\frac{1}{s^2+s+1}$ ซึ่งเอาท์พุทจะเสถียรช้า เนื่องจากตัวเศษที่มีค่ามากจะไปเพิ่ม Zeroให้กับระบบทำให้ระบบดีขึ้น

ส่วนค่าพารามิเตอร์ Kp ควรมีค่าที่เหมาะสม โดยถ้าค่า Kp มากไปจะทำให้เอาท์พุทเกิด Overshoot มากและถ้าค่า Kp น้อยไปเอาท์พุทก็จะใช้เวลาเข้าสู่สภาวะคงที่นาน ซึ่งค่า Kp จะไปเพิ่ม Gain ให้กับระบบ

2. PI Controller

ตัวควบคุมแบบนี้ เหมาะกับ Process ที่ตัวเศษมีค่ามาก เช่น $\frac{25s+5}{s^3+5s^2+s+1}$ ระบบจะเสถียรดีกว่า คือเอาท์พุทสามารถเข้าสู่ Set Point ได้เร็วและ Overshoot น้อย ส่วน Process ที่ตัวเศษมีกำลังน้อยเช่น $\frac{1}{s^2+s+1}$ เอาท์พุทจะมี Overshoot มาก และใช้เวลาในการเข้าสู่สภาวะคงที่นาน

ส่วนค่าพารามิเตอร์ Ki ควรมีค่าที่เหมาะสม โดยถ้าค่า Ki มากไปจะทำให้เอาท์พุทเกิด Overshoot มาก และถ้าค่า Ki น้อยไปเอาท์พุทก็จะมี Error มาก ซึ่งเป็นผลมาจาก ตัว Ki จะไปเพิ่ม Pole ให้กับระบบ

ข้อเสียของตัวควบคุมแบบนี้ คือค่าเอาท์พุทที่ได้จะต่ำกว่าจุด Set Point คือมี Error นั่นเอง

3. PD Controller

ตัวควบคุมแบบนี้ เหมาะกับ Process ที่ตัวเศษมีค่ามากเช่นกัน ส่วน Process ที่มีตัวเศษเป็น 1 และตัวส่วนมี Order สูงๆ เช่น $\frac{1}{s^3+5s^2+s}$ เอาท์พุทจะเกิดการออสลิเลท เนื่องจากตัว Kd จะไปเพิ่ม Zero ให้กับระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนค่าพารามิเตอร์ Kd ควรมีค่ามากๆ ซึ่งจะทำให้ลด Overshoot ได้มากและมีช่วงเวลาไต่ขึ้นน้อย ทำให้เอาท์พุทของระบบเสถียรได้เร็วขึ้น ซึ่งเป็นผลมาจาก ตัว Kd จะไปเพิ่ม Zero ให้กับระบบ

4. PID Controller

ตัวควบคุมแบบ PID-Controller นี้จะแก้ข้อเสียต่าง ๆ ของตัวควบคุมก่อน ๆ ได้ ซึ่งสามารถควบคุม Process ที่สลับซับซ้อนได้ แต่ในการปรับค่า Kp Ti Td ทำให้ยากพอสมควร ซึ่งเราสามารถปรับแต่งได้ 2 วิธีคือ

1. การสุ่มทางปฏิบัติทำได้โดย
   - 1.1 เลือก Controller แบบ PID Controller ป้อนค่า Kp อย่างเดียวแล้วให้ Ti มีค่าสูงสุดและ Td มีค่าเป็น 0 ต้องป้อน Kp ให้เกิดค่า Offset น้อยที่สุด และในขณะเดียวกันก็ยังเป็นค่าที่ยังคงรักษาไม่ให้กระบวนการเกิดการเปลี่ยนแปลงแบบ Oscillate แล้วลองเปลี่ยนค่าเป้าหมายใหม่ สังเกตดูว่ากระบวนการเปลี่ยนแปลงตามค่าเป้าหมายหรือไม่
   - 1.2 ต่อไปแก้ค่า Offset ที่เกิดขึ้น โดยการป้อน Ti ต้องค่อย ๆ ลดค่า Ti โดยให้เกิด Over Shoot มี Amplitude ต่ำที่สุด
   - 1.3 สุดท้ายป้อนค่า Td ต้องค่อย ๆ เพิ่มจากศูนย์ขึ้นไปตามลำดับ ซึ่งเทอมของ Derivative นี้เป็นการคาดคะเนความเปลี่ยนแปลงของกระบวนการล่วงหน้าซึ่งอยู่ในรูปของความชัน การป้อน Td นั้น จะต้องระมัดระวัง เพราะจะมีผลต่อการควบคุมเป็นอย่างมาก โดยจะทำให้เกิดความไวในการควบคุมอย่างรวมเร็ว จนอาจทำให้การควบคุมเกิดการเปลี่ยนแปลงแบบ Oscillate หรือ ขาดเสถียรภาพในการควบคุมได้
2. การใช้สูตรทางทฤษฎี

   สูตรที่ใช้กันอย่างแพร่หลายและนิยมคือ Ziegler Nichols

   $Kp = 0.6\ Ku$

   $Ti = 0.5\ Tu$

   $Td = 0.125\ Tu$

จะเห็นว่าตัวแปรที่เราต้องทราบคือ Ku (Ulitmate Gain) และ Tu(Ultimate Period Time) ซึ่งวิธีการหานั้นได้จากการทดลองป้อน Kp อย่างเดียว จนทำให้กระบวนการเกิดการเปลี่ยนแปลงแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Oscillate ซึ่งมีค่า Gain ที่เราป้อนนั้นก็คือ Ku และคาบเวลาของการ Oscillate ก็คือ Tu ซึ่งทั้ง Ku และ Tu นั้นขึ้นอยู่กับระบบของกระบวนการในแต่ละกระบวนการก็จะมีค่าแตกต่างกัน

จากการทดลองพบว่าตัวควบคุมแบบ PID นี้เหมาะกับทุกๆ Process ซึ่งเอาท์พุทที่ได้ก็ไม่ต่างกันมากนัก ไม่ว่าจะเป็น Process ที่มี Order ต่ำหรือสูง ตัวควบคุมแบบ PID ก็ยังคงสามารถควบคุมได้

ส่วนค่าพารามิเตอร์ Ti และ Td ควรมีค่าที่เหมาะสมโดยให้มีค่ามากจะดีกว่ามีค่าน้อยแต่ไม่ควรมากเกินไป ถ้ามีค่ามาก เอาท์พุทจะเกิด Overshoot มากแต่เขาสู่สภาวะเสถียรได้เร็ว แต่ถ้ามีค่าน้อยเกินไป ก็ยังคงทำให้ Over Shoot สูงขึ้นไปอีก และเข้าสู่สภาวะเสถียรได้ช้าลงไปอีก และจะมีการแก่วงมากด้วย ขึ้น ซึ่งเป็นผลมาจาก ตัว Td จะไปเพิ่ม Zero ให้กับระบบ และตัว Ti จะไปเพิ่ม Pole ให้กับระบบ

5. PID Zigler nichols

ตัวควบคุมแบบนี้สามารถจูนค่าพารามิเตอร์ Kp, Ti, Td ได้เองโดยอัตโนมัติ ซึ่งสามารถคำนวนได้จากค่า Pole และ Zero ของ Process ก็จะได้ค่า Kp, Ti, Td ที่เหมาะสมออกมาได้

ข้อดีของ Controller แบบนี้คือไม่ต้องปรับแต่งค่าพารามิเตอร์เอง ส่วนข้อเสียก็คือ เอาท์พุทที่ได้ยังไม่เสถียรดีพอต้องปรับแต่งอีกครั้ง

6. PID Fuzzy Control

การควบคุมโดยใช้ทฤษฎีของ Fuzzy เข้ามาช่วยนั้นสิ่งที่ต้องทราบจากระบวนการก็คือ Ku ส่วนค่า Kp, Kd นั้นค่าจะอยู่ที่ประมาณ 1 อยู่แล้วสิ่งที่ต้องปรับมากก็คือ ส่วนทฤษฎี Fuzzy นั้น ได้กล่าวอธิบายมาแล้ว ดังนั้นต่อไปจะเปรียบเทียบวิธีการควบคุมแบบต่าง ๆ ดังที่กล่าวมา

การเปรียบเทียบ การควบคุมแบบ PID และ PID-Fuzzy Controller สิ่งที่เห็นได้ชัดเจนที่สุดก็คือ

การควบคุมแบบ PID ตัวแปรการควบคุม Kp, Ti, Td ผู้ควบคุมจะต้องป้อนให้ตัวควบคุมเอง และค่า Kp, Ti, Td นั้นจะคงที่ตลอดของการควบคุม แม้ว่าการเปลี่ยนแปลงของค่าความผิดพลาดของกระบวนการจะมากหรือน้อยอย่างไรก็ตาม ค่า Kp, Ti, Td ก็ยังคงที่ตลอด ดังนั้น การควบคุมด้วยวิธีนี้ผู้ควบคุมต้องมีความชำนาญ และมีประสบการณ์ในการสังเกตพฤติกรรมของกระบวนการมากพอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมด้วยวิธีใช้สูตร Zeigler Nichols นั้น เมื่อหาค่า Ku และ Tu ได้แล้วก็ป้อน Kp, Ti, Td ที่หาได้เข้าสมการ PID ได้เลย แต่ทั้งนี้การควบคุมด้วยการใช้สูตรนี้ บางครั้งอาจจะไม่ดีเท่าที่ควรนัก เพราะยังเป็นค่าที่ไม่ละเอียดมากพอ ดังนั้น ผู้ควบคุมจำเป็นต้องปรับตัวแปรให้เหมาะสมต่อไปด้วยการสังเกตพฤติกรรมของกระบวนการเช่นกัน

การควบคุมแบบ PID Fuzzy ค่าตัวแปรการควบคุม Kp, Kd, L ผู้ควบคุมไม่ต้องเปลี่ยนค่ามากนัก เปลี่ยนเพียงแต่ค่า Ku เท่านั้น โดยเอาค่าความผิดพลาดของกระบวนการในแต่ละรอบการคำนวณนำมาหาค่า Fuzzy Output ซึ่งความละเอียดและความเหมาะสมของค่านั้นก็ขึ้นอยู่กับการตั้งกฎและการใช้ทฤษฎีของ Fuzzy ได้สร้างรายละเอียดมากน้อยเพียงไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. Katsuhiko Ogata, *Modern Control Engineering*, Second edition, Prentice-Hall International, Inc., 1990.
2. Heidar A. Malki, Huaidong Li, and Guanrong Chen, *New Design and Stability Analysis of Fuzzy Proportional-Derivative Control Systems*, IEEE Transaction on Fuzzy Systems., vol. 2, No. 4, pp. 245-254, 1994.
3. Natban Gurewich & Ori Gurewich, *Visual Basic4 in 21 days*, Sams Publishing, 1995.
4. กิตติ ตีรเศรษฐ, *พื้นฐานวิศวกรรมระบบควบคุม*, คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง พ.ศ. 2539
5. กิตติ ตีรเศรษฐ, *การวิเคราะห์ระบบควบคุมเชิงเส้น*, คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง พ.ศ. 2525
6. กิตติ อินทรานนท์, ศรีรักษ์ ศรีทองชัย, *การตัดสินใจโดยใช้ทฤษฎีฟัซซี่เซต*, วิศวกรรมสาร, ปีที่ 46 เล่มที่ 2, กุมภาพันธ์ 2536
7. สุทธิศักดิ์ พงศ์ธนาพาณิช, *Visual Basic 4.0 Professional*, ซีเอ็ดยูเคชั่น จำกัด (มหาชน), พ.ศ. 2539

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

Source Code

โปรแกรม PID & Fuzzy PD Control Systems

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
---------- pd.Frm----------
Dim a00, a01, a02, a03
Dim b00, b01, b02, b03
Dim a0, a1, a2, a3, a4
Dim b0, b1, b2, b3, b4
Dim ba0, ba1, ba2, ba3, ba4
Dim c1, c2, c3, c4, c5
Dim d1, d2, d3, d4, d5
Dim e1, e2, e3, e4, e5
Dim f1, f2, f3, f4, f5
Dim x1, x2, x3, x4, i1, i2, i3, i4
Dim k1, l1, m1, n1, k2, l2, m2, n2
Dim k3, l3, m3, n3, k4, l4, m4, n4
Dim z, t, tt, tt1, xx1
Dim a, b, c, d, G, Kp, Sp
Dim t20, t30, t40, t50
Dim counter

Dim set1, time1, time, x10, y10
Dim DrawNow As Integer

Dim color, color1, col

Private Sub box_MouseMove(Button As Integer, Shift As
Integer, X As Single, Y As Single)
 time = 10
 x10 = (X - 630)
 y10 = (4010 - Y)

 time1 = (time / 4730)
 set1 = (1 / 2330)
 If DrawNow Then
     Line -(time1, set1)
     Circle (time1, set1), 50
   End If

End Sub

Private Sub Close_Click()
End
End Sub

Private Sub Ex_Click()
Tex1 = 1
Tex2 = 5
Tex3 = 1
Tex4 = 0
Tex5 = 0
Tex6 = 25
Tex7 = 5
Text1 = 1
Text2 = 30
Text3 = 1
End Sub


Private Sub Form_Load()
color = 1
End Sub

Private Sub Plot_Click()

a00 = Tex1.Text
a01 = Tex2.Text
a02 = Tex3.Text
a03 = Tex4.Text
b00 = 0
b01 = Tex5.Text
b02 = Tex6.Text
b03 = Tex7.Text
Kp = Text2.Text
Kd = Text3.Text
Sp = 1

  If color = 1 Then
     color1 = 9
  End If

  If color = 2 Then
     color1 = 12
  End If

  If color = 3 Then
     color1 = 13
  End If

  If color1 = 9 Then
     T11.Text = Tex1.Text
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    T12.Text = Tex2.Text
    T13.Text = Tex3.Text
    T14.Text = Tex4.Text
    T15.Text = Tex5.Text
    T16.Text = Tex6.Text
    T17.Text = Tex7.Text
    T18.Text = Text2.Text
    T19.Text = Text3.Text
  End If

  If color1 = 12 Then
    T21.Text = Tex1.Text
    T22.Text = Tex2.Text
    T23.Text = Tex3.Text
    T24.Text = Tex4.Text
    T25.Text = Tex5.Text
    T26.Text = Tex6.Text
    T27.Text = Tex7.Text
    T28.Text = Text2.Text
    T29.Text = Text3.Text
  End If

  If color1 = 13 Then
    T31.Text = Tex1.Text
    T32.Text = Tex2.Text
    T33.Text = Tex3.Text
    T34.Text = Tex4.Text
    T35.Text = Tex5.Text
    T36.Text = Tex6.Text
    T37.Text = Tex7.Text
    T38.Text = Text2.Text
    T39.Text = Text3.Text
  End If

If Tex1.Text = 0 Then GoTo line1 Else GoTo line2
line1:

  a00 = Tex2.Text
  a01 = Tex3.Text
  a02 = Tex4.Text
  b00 = 0
  b01 = Tex6.Text
  b02 = Tex7.Text
  Kp = Text2.Text
    Kd = Text3.Text

  m00 = (Kp * b01)
  m01 = (Kp * b01) + (Kd * b02)
  m02 = (Kp * b02) + (Kd * b03)

  a0 = a00 + m00
  a1 = a01 + m01
  a2 = a02 + m02

  b0 = m00
  b1 = m01
  b2 = m02

  ba0 = b0
  ba1 = b1
  ba2 = b2 - a1 * ba1

  c1 = 0: c2 = 1: c3 = ba1
  d1 = -a2: d2 = -a1: d3 = ba2

  t = 0: x1 = 0: x2 = 0
  z = 0.001
  tt = 10
  Cls
  For counter = 0 To tt Step z

  i1 = c1 * x1 + c2 * x2 + c3
  i2 = d1 * x1 + d2 * x2 + d3

  k1 = z * i1
  l1 = z * i2

  k2 = z * (c1 * (x1 + 0.5 * k1) + c2 * (x2 + 0.5 * l1) + c3)
  l2 = z * (d1 * (x1 + 0.5 * k1) + d2 * (x2 + 0.5 * l1) + d3)

  k3 = z * (c1 * (x1 + 0.5 * k2) + c2 * (x2 + 0.5 * l2) + c3)
  l3 = z * (d1 * (x1 + 0.5 * k2) + d2 * (x2 + 0.5 * l2) + d3)

  k4 = z * (c1 * (x1 + 0.5 * k3) + c2 * (x2 + 0.5 * l3) + c3)
  l4 = z * (d1 * (x1 + 0.5 * k3) + d2 * (x2 + 0.5 * l3) + d3)

  x1 = x1 + (k1 + 2 * k2 + 2 * k3 + k4) / 6
  x2 = x2 + (l1 + 2 * l2 + 2 * l3 + l4) / 6
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
t = t + z

tt1 = t * 360
xx1 = 2850 - (x1 * 1760)
box.Circle (tt1, xx1), 10, QBColor(color1)
Next
GoTo lastline

line2:

m00 = (Kp * b01)
m01 = (Kp * b01) + (Kd * b02)
m02 = (Kp * b02) + (Kd * b03)
m03 = (Kp * b03)

a0 = a00 + m00
a1 = a01 + m01
a2 = a02 + m02
a3 = a03 + m03

b0 = m00
b1 = m01
b2 = m02
b3 = m03

ba0 = b0
ba1 = b1
ba2 = b2 - a1 * ba1
ba3 = b3 - a1 * ba2 - a2 * ba1

c1 = 0: c2 = 1: c3 = 0: c4 = ba1
d1 = 0: d2 = 0: d3 = 1: d4 = ba2
f1 = -a3: f2 = -a2: f3 = -a1: f4 = ba3
t = 0: x1 = 0: x2 = 0: x3 = 0
z = 0.001
tt = 10
Cls
For counter = 0 To tt Step z

i1 = c1 * x1 + c2 * x2 + c3 * x3 + c4
i2 = d1 * x1 + d2 * x2 + d3 * x3 + d4
i3 = f1 * x1 + f2 * x2 + f3 * x3 + f4

k1 = z * i1
l1 = z * i2
m1 = z * i3

k2 = z * (c1 * (x1 + 0.5 * k1) + c2 * (x2 + 0.5 * l1) + c3 * (x3
+ 0.5 * m1) + c4)
l2 = z * (d1 * (x1 + 0.5 * k1) + d2 * (x2 + 0.5 * l1) + d3 * (x3
+ 0.5 * m1) + d4)
m2 = z * (f1 * (x1 + 0.5 * k1) + f2 * (x2 + 0.5 * l1) + f3 * (x3 +
0.5 * m1) + f4)

k3 = z * (c1 * (x1 + 0.5 * k2) + c2 * (x2 + 0.5 * l2) + c3 * (x3
+ 0.5 * m2) + c4)
l3 = z * (d1 * (x1 + 0.5 * k2) + d2 * (x2 + 0.5 * l2) + d3 * (x3
+ 0.5 * m2) + d4)
m3 = z * (f1 * (x1 + 0.5 * k2) + f2 * (x2 + 0.5 * l2) + f3 * (x3 +
0.5 * m2) + f4)

k4 = z * (c1 * (x1 + 0.5 * k3) + c2 * (x2 + 0.5 * l3) + c3 * (x3
+ 0.5 * m3) + c4)
l4 = z * (d1 * (x1 + 0.5 * k3) + d2 * (x2 + 0.5 * l3) + d3 * (x3
+ 0.5 * m3) + d4)
m4 = z * (f1 * (x1 + 0.5 * k3) + f2 * (x2 + 0.5 * l3) + f3 * (x3 +
0.5 * m3) + f4)

x1 = x1 + (k1 + 2 * k2 + 2 * k3 + k4) / 6
x2 = x2 + (l1 + 2 * l2 + 2 * l3 + l4) / 6
x3 = x3 + (m1 + 2 * m2 + 2 * m3 + m4) / 6

t = t + z

tt1 = t * 360
xx1 = 2850 - (x1 * 1760)
box.Circle (tt1, xx1), 10, QBColor(color1)

Next

lastline:

color = color + 1
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
---------- pi_or2.Frm----------
Dim a00, a01, a02, a03
Dim b00, b01, b02, b03
Dim a0, a1, a2, a3, a4
Dim b0, b1, b2, b3, b4
Dim ba0, ba1, ba2, ba3, ba4
Dim c1, c2, c3, c4, c5
Dim d1, d2, d3, d4, d5
Dim e1, e2, e3, e4, e5
Dim f1, f2, f3, f4, f5
Dim x1, x2, x3, x4, i1, i2, i3, i4
Dim k1, l1, m1, n1, k2, l2, m2, n2
Dim k3, l3, m3, n3, k4, l4, m4, n4
Dim z, t, tt, tt1, xx1
Dim a, b, c, d, G, Kp, Sp
Dim t20, t30, t40, t50
Dim counter

Dim set1, time1, time, x10, y10
Dim DrawNow As Integer

Dim color, color1, col

Private Sub box_MouseMove(Button As Integer, Shift As
Integer, X As Single, Y As Single)
 time = 10
 x10 = (X - 630)
 y10 = (4010 - Y)

 time1 = (time / 4730)
 set1 = (1 / 2330)
 If DrawNow Then
     Line -(time1, set1)
     Circle (time1, set1), 50
   End If

End Sub

Private Sub Close_Click()
End
End Sub

Private Sub Ex_Click()
Tex1 = 1
Tex2 = 6
Tex3 = 5
Tex4 = 0
Tex5 = 0
Tex6 = 0
Tex7 = 1

Text2 = 10
Text3 = 0.05
End Sub

Private Sub Form_Load()
color = 1
End Sub

Private Sub Plot_Click()

a00 = Tex1.Text
a01 = Tex2.Text
a02 = Tex3.Text
a03 = Tex4.Text
b00 = 0
b01 = Tex5.Text
b02 = Tex6.Text
b03 = Tex7.Text
Kp = Text2.Text
Ki = Text3.Text
Sp = 1

 If color = 1 Then
    color1 = 9
 End If

 If color = 2 Then
    color1 = 12
 End If

 If color = 3 Then
    color1 = 13
 End If

 If color1 = 9 Then
    T11.Text = Tex1.Text
    T12.Text = Tex2.Text
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    T13.Text = Tex3.Text
    T14.Text = Tex4.Text
    T15.Text = Tex5.Text
    T16.Text = Tex6.Text
    T17.Text = Tex7.Text
    T18.Text = Text2.Text
    T19.Text = Text3.Text
End If

If color1 = 12 Then
    T21.Text = Tex1.Text
    T22.Text = Tex2.Text
    T23.Text = Tex3.Text
    T24.Text = Tex4.Text
    T25.Text = Tex5.Text
    T26.Text = Tex6.Text
    T27.Text = Tex7.Text
    T28.Text = Text2.Text
    T29.Text = Text3.Text
End If

If color1 = 13 Then
    T31.Text = Tex1.Text
    T32.Text = Tex2.Text
    T33.Text = Tex3.Text
    T34.Text = Tex4.Text
    T35.Text = Tex5.Text
    T36.Text = Tex6.Text
    T37.Text = Tex7.Text
    T38.Text = Text2.Text
    T39.Text = Text3.Text
End If

If Tex1.Text = 0 Then GoTo line1 Else GoTo line2

line1:

    a00 = Tex2.Text
    a01 = Tex3.Text
    a02 = Tex4.Text
    b00 = 0
    b01 = Tex6.Text
    b02 = Tex7.Text
    Kp = Text2.Text
    Ki = Text3.Text

    m01 = (Kp * b01)
    m02 = (Kp * b02) + (Ki * b01)
    m03 = (Ki * b02)

    a0 = a00
    a1 = a01 + m01
    a2 = a02 + m02
    a3 = m03

    b0 = b00
    b1 = m01
    b2 = m02
    b3 = m03

    ba0 = 0
    ba1 = ba1
    ba2 = b2 - a1 * ba1
    ba3 = b3 - a1 * ba2 - a2 * ba1

    c1 = 0: c2 = 1: c3 = 0: c4 = ba1
    d1 = 0: d2 = 0: d3 = 1: d4 = ba2
    e1 = -a3: e2 = -a2: e3 = -a1: e4 = ba3

    t = 0: x1 = 0: x2 = 0: x3 = 0: x4 = 0
    z = 0.005
    tt = 10
    Cls
    For counter = 0 To tt Step z

    i1 = c1 * x1 + c2 * x2 + c3 * x3 + c4
    i2 = d1 * x1 + d2 * x2 + d3 * x3 + d4
    i3 = e1 * x1 + e2 * x2 + e3 * x3 + e4

    k1 = z * i1
    l1 = z * i2
    m1 = z * i3

    k2 = z * (c1 * (x1 + 0.5 * k1) + c2 * (x2 + 0.5 * l1) + c3 * (x3
    + 0.5 * m1) + c4)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
l2 = z * (d1 * (x1 + 0.5 * k1) + d2 * (x2 + 0.5 * l1) + d3 * (x3
+ 0.5 * m1) + d4)
m2 = z * (e1 * (x1 + 0.5 * k1) + e2 * (x2 + 0.5 * l1) + e3 * (x3
+ 0.5 * m1) + e4)

k3 = z * (c1 * (x1 + 0.5 * k2) + c2 * (x2 + 0.5 * l2) + c3 * (x3
+ 0.5 * m2) + c4)
l3 = z * (d1 * (x1 + 0.5 * k2) + d2 * (x2 + 0.5 * l2) + d3 * (x3
+ 0.5 * m2) + d4)
m3 = z * (e1 * (x1 + 0.5 * k2) + e2 * (x2 + 0.5 * l2) + e3 * (x3
+ 0.5 * m2) + e4)

k4 = z * (c1 * (x1 + 0.5 * k3) + c2 * (x2 + 0.5 * l3) + c3 * (x3
+ 0.5 * m3) + c4)
l4 = z * (d1 * (x1 + 0.5 * k3) + d2 * (x2 + 0.5 * l3) + d3 * (x3
+ 0.5 * m3) + d4)
m4 = z * (e1 * (x1 + 0.5 * k3) + e2 * (x2 + 0.5 * l3) + e3 * (x3
+ 0.5 * m3) + e4)

x1 = x1 + (k1 + 2 * k2 + 2 * k3 + k4) / 6
x2 = x2 + (l1 + 2 * l2 + 2 * l3 + l4) / 6
x3 = x3 + (m1 + 2 * m2 + 2 * m3 + m4) / 6

t = t + z

tt1 = t * 360
xx1 = 2850 - (x1 * 1760)
box.Circle (tt1, xx1), 10, QBColor(color1)
Next
GoTo lastline

line2:

m01 = (Kp * b01)
m02 = (Kp * b02) + (Ki * b01)
m03 = (Kp * b03) + (Ki * b02)
m04 = (Ki * b03)

a0 = a00
a1 = a01 + m01
a2 = a02 + m02
a3 = a03 + m03
a4 = m04

b0 = b00
b1 = m01
b2 = m02
b3 = m03
b4 = m04

ba0 = 0
ba1 = ba1
ba2 = b2 - a1 * ba1
ba3 = b3 - a1 * ba2 - a2 * ba1
ba4 = b4 - a1 * ba3 - a2 * ba2 - a3 * ba1

c1 = 0: c2 = 1: c3 = 0: c4 = 0: c5 = ba1
d1 = 0: d2 = 0: d3 = 1: d4 = 0: d5 = ba2
e1 = 0: e2 = 0: e3 = 0: e4 = 1: e5 = ba3
f1 = -a4: f2 = -a3: f3 = -a2: f4 = -a1: f5 = ba4
t = 0: x1 = 0: x2 = 0: x3 = 0: x4 = 0
z = 0.005
tt = 10
Cls
For counter = 0 To tt Step z

i1 = c1 * x1 + c2 * x2 + c3 * x3 + c4 * x4 + c5
i2 = d1 * x1 + d2 * x2 + d3 * x3 + d4 * x4 + d5
i3 = e1 * x1 + e2 * x2 + e3 * x3 + e4 * x4 + e5
i4 = f1 * x1 + f2 * x2 + f3 * x3 + f4 * x4 + f5

k1 = z * i1
l1 = z * i2
m1 = z * i3
n1 = z * i4

k2 = z * (c1 * (x1 + 0.5 * k1) + c2 * (x2 + 0.5 * l1) + c3 * (x3
+ 0.5 * m1) + c4 * (x4 + 0.5 * n1) + c5)
l2 = z * (d1 * (x1 + 0.5 * k1) + d2 * (x2 + 0.5 * l1) + d3 * (x3
+ 0.5 * m1) + d4 * (x4 + 0.5 * n1) + d5)
m2 = z * (e1 * (x1 + 0.5 * k1) + e2 * (x2 + 0.5 * l1) + e3 * (x3
+ 0.5 * m1) + e4 * (x4 + 0.5 * n1) + e5)
n2 = z * (f1 * (x1 + 0.5 * k1) + f2 * (x2 + 0.5 * l1) + f3 * (x3 +
0.5 * m1) + f4 * (x4 + 0.5 * n1) + f5)

k3 = z * (c1 * (x1 + 0.5 * k2) + c2 * (x2 + 0.5 * l2) + c3 * (x3
+ 0.5 * m2) + c4 * (x4 + 0.5 * n2) + c5)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
l3 = z * (d1 * (x1 + 0.5 * k2) + d2 * (x2 + 0.5 * l2) + d3 * (x3
+ 0.5 * m2) + d4 * (x4 + 0.5 * n2) + d5)
m3 = z * (e1 * (x1 + 0.5 * k2) + e2 * (x2 + 0.5 * l2) + e3 * (x3
+ 0.5 * m2) + e4 * (x4 + 0.5 * n2) + e5)
n3 = z * (f1 * (x1 + 0.5 * k2) + f2 * (x2 + 0.5 * l2) + f3 * (x3 +
0.5 * m2) + f4 * (x4 + 0.5 * n2) + f5)

k4 = z * (c1 * (x1 + 0.5 * k3) + c2 * (x2 + 0.5 * l3) + c3 * (x3
+ 0.5 * m3) + c4 * (x4 + 0.5 * n3) + c5)
l4 = z * (d1 * (x1 + 0.5 * k3) + d2 * (x2 + 0.5 * l3) + d3 * (x3
+ 0.5 * m3) + d4 * (x4 + 0.5 * n3) + d5)
m4 = z * (e1 * (x1 + 0.5 * k3) + e2 * (x2 + 0.5 * l3) + e3 * (x3
+ 0.5 * m3) + e4 * (x4 + 0.5 * n3) + e5)
n4 = z * (f1 * (x1 + 0.5 * k3) + f2 * (x2 + 0.5 * l3) + f3 * (x3 +
0.5 * m3) + f4 * (x4 + 0.5 * n3) + f5)

x1 = x1 + (k1 + 2 * k2 + 2 * k3 + k4) / 6
x2 = x2 + (l1 + 2 * l2 + 2 * l3 + l4) / 6
x3 = x3 + (m1 + 2 * m2 + 2 * m3 + m4) / 6
x4 = x4 + (n1 + 2 * n2 + 2 * n3 + n4) / 6

t = t + z

tt1 = t * 360
xx1 = 2850 - (x1 * 1760)
box.Circle (tt1, xx1), 10, QBColor(color1)

Next

lastline:

color = color + 1
End Sub
```

---------- pid_or2.Frm----------

```
Dim a00, a01, a02, a03
Dim b00, b01, b02, b03
Dim a0, a1, a2, a3, a4
Dim b0, b1, b2, b3, b4
Dim ba0, ba1, ba2, ba3, ba4
Dim c1, c2, c3, c4, c5
Dim d1, d2, d3, d4, d5
Dim e1, e2, e3, e4, e5
Dim f1, f2, f3, f4, f5
Dim x1, x2, x3, x4, i1, i2, i3, i4
Dim k1, l1, m1, n1, k2, l2, m2, n2
Dim k3, l3, m3, n3, k4, l4, m4, n4
Dim z, t, tt, tt1, xx1
Dim a, b, c, d, G, Kp, Sp
Dim t20, t30, t40, t50
Dim counter

Dim set1, time1, time, x10, y10
Dim DrawNow As Integer

Dim color, color1, col

Dim m00, m01, m02, m03, m04

Private Sub box_MouseMove(Button As Integer, Shift As
Integer, X As Single, Y As Single)
  time = 10
  x10 = (X - 630)
  y10 = (4010 - Y)

  time1 = (time / 4730)
  set1 = (1 / 2330)
  If DrawNow Then
    Line -(time1, set1)
    Circle (time1, set1), 50
  End If

End Sub
```

```
Private Sub Close_Click()
End
End Sub

Private Sub Ex_Click()
Tex1 = 1
Tex2 = 6
Tex3 = 5
Tex4 = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Tex5 = 0
Tex6 = 0
Tex7 = 1

Text2 = 40
Text3 = 3
Text1 = 0.8
End Sub


Private Sub Form_Load()
color = 1
End Sub

Private Sub Plot_Click()

a00 = Tex1.Text
a01 = Tex2.Text
a02 = Tex3.Text
a03 = Tex4.Text
b00 = 0
b01 = Tex5.Text
b02 = Tex6.Text
b03 = Tex7.Text
Kp = Text2.Text
Ti = Text3.Text
Td = Text1.Text

  If color = 1 Then
     color1 = 9
  End If

  If color = 2 Then
     color1 = 12
  End If

  If color = 3 Then
     color1 = 13
  End If

  If color1 = 9 Then
     T11.Text = Tex1.Text
     T12.Text = Tex2.Text
     T13.Text = Tex3.Text
     T14.Text = Tex4.Text
     T15.Text = Tex5.Text
     T16.Text = Tex6.Text
     T17.Text = Tex7.Text
     T18.Text = Text2.Text
     T19.Text = Text3.Text
     T191.Text = Text1.Text
  End If

  If color1 = 12 Then
     T21.Text = Tex1.Text
     T22.Text = Tex2.Text
     T23.Text = Tex3.Text
     T24.Text = Tex4.Text
     T25.Text = Tex5.Text
     T26.Text = Tex6.Text
     T27.Text = Tex7.Text
     T28.Text = Text2.Text
     T29.Text = Text3.Text
     T292.Text = Text1.Text
  End If

  If color1 = 13 Then
     T31.Text = Tex1.Text
     T32.Text = Tex2.Text
     T33.Text = Tex3.Text
     T34.Text = Tex4.Text
     T35.Text = Tex5.Text
     T36.Text = Tex6.Text
     T37.Text = Tex7.Text
     T38.Text = Text2.Text
     T39.Text = Text3.Text
     T392.Text = Text1.Text
  End If


  If Tex1.Text = 0 Then GoTo line1 Else GoTo line2

  line1:

     a00 = Tex2.Text
     a01 = Tex3.Text
     a02 = Tex4.Text
     b00 = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
b01 = Tex6.Text
b02 = Tex7.Text
Kp = Text2.Text
Ti = Text3.Text
Td = Text1.Text


m00 = (Kp * Td * b01)
m01 = (Kp * Td * b02) + (Kp * b01)
m02 = (Kp * b02) + (Kp * b01 / Ti)
m03 = (Kp * b02 / Ti)

a0 = a00 + m00
a1 = a01 + m01
a2 = a02 + m02
a3 = m03

b0 = m00
b1 = m01
b2 = m02
b3 = m03

ba0 = 0
ba1 = ba1
ba2 = b2 - a1 * ba1
ba3 = b3 - a1 * ba2 - a2 * ba1

c1 = 0: c2 = 1: c3 = 0: c4 = ba1
d1 = 0: d2 = 0: d3 = 1: d4 = ba2
e1 = -a3: e2 = -a2: e3 = -a1: e4 = ba3

t = 0: x1 = 0: x2 = 0: x3 = 0: x4 = 0
z = 0.005
tt = 10
Cls
For counter = 0 To tt Step z

i1 = c1 * x1 + c2 * x2 + c3 * x3 + c4
i2 = d1 * x1 + d2 * x2 + d3 * x3 + d4
i3 = e1 * x1 + e2 * x2 + e3 * x3 + e4

k1 = z * i1
l1 = z * i2
m1 = z * i3

k2 = z * (c1 * (x1 + 0.5 * k1) + c2 * (x2 + 0.5 * l1) + c3 * (x3
+ 0.5 * m1) + c4)
l2 = z * (d1 * (x1 + 0.5 * k1) + d2 * (x2 + 0.5 * l1) + d3 * (x3
+ 0.5 * m1) + d4)
m2 = z * (e1 * (x1 + 0.5 * k1) + e2 * (x2 + 0.5 * l1) + e3 * (x3
+ 0.5 * m1) + e4)

k3 = z * (c1 * (x1 + 0.5 * k2) + c2 * (x2 + 0.5 * l2) + c3 * (x3
+ 0.5 * m2) + c4)
l3 = z * (d1 * (x1 + 0.5 * k2) + d2 * (x2 + 0.5 * l2) + d3 * (x3
+ 0.5 * m2) + d4)
m3 = z * (e1 * (x1 + 0.5 * k2) + e2 * (x2 + 0.5 * l2) + e3 * (x3
+ 0.5 * m2) + e4)

k4 = z * (c1 * (x1 + 0.5 * k3) + c2 * (x2 + 0.5 * l3) + c3 * (x3
+ 0.5 * m3) + c4)
l4 = z * (d1 * (x1 + 0.5 * k3) + d2 * (x2 + 0.5 * l3) + d3 * (x3
+ 0.5 * m3) + d4)
m4 = z * (e1 * (x1 + 0.5 * k3) + e2 * (x2 + 0.5 * l3) + e3 * (x3
+ 0.5 * m3) + e4)

x1 = x1 + (k1 + 2 * k2 + 2 * k3 + k4) / 6
x2 = x2 + (l1 + 2 * l2 + 2 * l3 + l4) / 6
x3 = x3 + (m1 + 2 * m2 + 2 * m3 + m4) / 6

t = t + z

tt1 = t * 360
xx1 = 2850 - (x1 * 1760)
box.Circle (tt1, xx1), 10, QBColor(color1)
Next
GoTo lastline


line2:

m00 = (Kp * Td * b01)
m01 = (Kp * Td * b02) + (Kp * b01)
m02 = (Kp * Td * b03) + (Kp * b02) + (Kp * b01 / Ti)
m03 = (Kp * b03) + (Kp * b02 / Ti)
m04 = (Kp * b03 / Ti)

a0 = a00 + m00
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
a1 = a01 + m01
a2 = a02 + m02
a3 = a03 + m03
a4 = m04

b0 = m00
b1 = m01
b2 = m02
b3 = m03
b4 = m04

ba0 = 0
ba1 = ba1
ba2 = b2 - a1 * ba1
ba3 = b3 - a1 * ba2 - a2 * ba1
ba4 = b4 - a1 * ba3 - a2 * ba2 - a3 * ba1

c1 = 0: c2 = 1: c3 = 0: c4 = 0: c5 = ba1
d1 = 0: d2 = 0: d3 = 1: d4 = 0: d5 = ba2
e1 = 0: e2 = 0: e3 = 0: e4 = 1: e5 = ba3
f1 = -a4: f2 = -a3: f3 = -a2: f4 = -a1: f5 = ba4

t = 0: x1 = 0: x2 = 0: x3 = 0: x4 = 0
z = 0.005
tt = 10

Cls
For counter = 0 To tt Step z
i1 = c1 * x1 + c2 * x2 + c3 * x3 + c4 * x4 + c5
i2 = d1 * x1 + d2 * x2 + d3 * x3 + d4 * x4 + d5
i3 = e1 * x1 + e2 * x2 + e3 * x3 + e4 * x4 + e5
i4 = f1 * x1 + f2 * x2 + f3 * x3 + f4 * x4 + f5

k1 = z * i1
l1 = z * i2
m1 = z * i3
n1 = z * i4

k2 = z * (c1 * (x1 + 0.5 * k1) + c2 * (x2 + 0.5 * l1) + c3 * (x3
+ 0.5 * m1) + c4 * (x4 + 0.5 * n1) + c5)
l2 = z * (d1 * (x1 + 0.5 * k1) + d2 * (x2 + 0.5 * l1) + d3 * (x3
+ 0.5 * m1) + d4 * (x4 + 0.5 * n1) + d5)
m2 = z * (e1 * (x1 + 0.5 * k1) + e2 * (x2 + 0.5 * l1) + e3 * (x3
+ 0.5 * m1) + e4 * (x4 + 0.5 * n1) + e5)
n2 = z * (f1 * (x1 + 0.5 * k1) + f2 * (x2 + 0.5 * l1) + f3 * (x3 +
0.5 * m1) + f4 * (x4 + 0.5 * n1) + f5)

k3 = z * (c1 * (x1 + 0.5 * k2) + c2 * (x2 + 0.5 * l2) + c3 * (x3
+ 0.5 * m2) + c4 * (x4 + 0.5 * n2) + c5)
l3 = z * (d1 * (x1 + 0.5 * k2) + d2 * (x2 + 0.5 * l2) + d3 * (x3
+ 0.5 * m2) + d4 * (x4 + 0.5 * n2) + d5)
m3 = z * (e1 * (x1 + 0.5 * k2) + e2 * (x2 + 0.5 * l2) + e3 * (x3
+ 0.5 * m2) + e4 * (x4 + 0.5 * n2) + e5)
n3 = z * (f1 * (x1 + 0.5 * k2) + f2 * (x2 + 0.5 * l2) + f3 * (x3 +
0.5 * m2) + f4 * (x4 + 0.5 * n2) + f5)

k4 = z * (c1 * (x1 + 0.5 * k3) + c2 * (x2 + 0.5 * l3) + c3 * (x3
+ 0.5 * m3) + c4 * (x4 + 0.5 * n3) + c5)
l4 = z * (d1 * (x1 + 0.5 * k3) + d2 * (x2 + 0.5 * l3) + d3 * (x3
+ 0.5 * m3) + d4 * (x4 + 0.5 * n3) + d5)
m4 = z * (e1 * (x1 + 0.5 * k3) + e2 * (x2 + 0.5 * l3) + e3 * (x3
+ 0.5 * m3) + e4 * (x4 + 0.5 * n3) + e5)
n4 = z * (f1 * (x1 + 0.5 * k3) + f2 * (x2 + 0.5 * l3) + f3 * (x3 +
0.5 * m3) + f4 * (x4 + 0.5 * n3) + f5)

x1 = x1 + (k1 + 2 * k2 + 2 * k3 + k4) / 6
x2 = x2 + (l1 + 2 * l2 + 2 * l3 + l4) / 6
x3 = x3 + (m1 + 2 * m2 + 2 * m3 + m4) / 6
x4 = x4 + (n1 + 2 * n2 + 2 * n3 + n4) / 6

t = t + z

tt1 = t * 360
xx1 = 2850 - (x1 * 1760)
box.Circle (tt1, xx1), 10, QBColor(color1)

Next

lastline:

color = color + 1
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
---------- Zigler_or2.Frm----------
Dim a00, a01, a02, a03
Dim b00, b01, b02, b03
Dim a0, a1, a2, a3, a4
Dim b0, b1, b2, b3, b4
Dim ba0, ba1, ba2, ba3, ba4
Dim c1, c2, c3, c4, c5
Dim d1, d2, d3, d4, d5
Dim e1, e2, e3, e4, e5
Dim f1, f2, f3, f4, f5
Dim x1, x2, x3, x4, i1, i2, i3, i4
Dim k1, l1, m1, n1, k2, l2, m2, n2
Dim k3, l3, m3, n3, k4, l4, m4, n4
Dim z, t, tt, tt1, xx1
Dim a, b, c, d, G, Kp, Sp
Dim t20, t30, t40, t50
Dim counter

Dim set1, time1, time, x10, y10
Dim DrawNow As Integer

Dim color, color1, col

Dim m00, m01, m02, m03, m04

Private Sub box_MouseMove(Button As Integer, Shift As
Integer, X As Single, Y As Single)
 time = 10
 x10 = (X - 630)
 y10 = (4010 - Y)

 time1 = (time / 4730)
 set1 = (1 / 2330)
 If DrawNow Then
     Line -(time1, set1)
     Circle (time1, set1), 50
  End If

End Sub

Private Sub Close_Click()
End
End Sub
```

```
Private Sub Ex_Click()
Tex1 = 1
Tex2 = 6
Tex3 = 5
Tex4 = 0
Tex5 = 0
Tex6 = 0
Tex7 = 1

Text2 = 40
Text3 = 3
Text1 = 0.8
End Sub

Private Sub Form_Load()
color = 1
End Sub

Private Sub Plot_Click()

a00 = Tex1.Text
a01 = Tex2.Text
a02 = Tex3.Text
a03 = Tex4.Text
b00 = 0
b01 = Tex5.Text
b02 = Tex6.Text
b03 = Tex7.Text

Kcr = a01 * a02 / a00
OMAGA = (((((4 * a01 * Kcr) ^ 2) ^ (1 / 2))) ^ (1 / 2)) / (2 *
a01)
Pcr = 2 * 3.142 / OMAGA

Kp = 0.6 * Kcr
Ti = 0.5 * Pcr
Td = 0.125 * Pcr

  If color = 1 Then
     color1 = 9
  End If

  If color = 2 Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    color1 = 12
  End If
    If color = 3 Then
    color1 = 13
  End If

  If color1 = 9 Then
    T11.Text = Tex1.Text
    T12.Text = Tex2.Text
    T13.Text = Tex3.Text
    T14.Text = Tex4.Text
    T15.Text = Tex5.Text
    T16.Text = Tex6.Text
    T17.Text = Tex7.Text
    T18.Text = Kp
    T19.Text = Ti
    T191.Text = Td
  End If

  If color1 = 12 Then
    T21.Text = Tex1.Text
    T22.Text = Tex2.Text
    T23.Text = Tex3.Text
    T24.Text = Tex4.Text
    T25.Text = Tex5.Text
    T26.Text = Tex6.Text
    T27.Text = Tex7.Text
    T28.Text = Kp
    T29.Text = Ti
    T292.Text = Td
  End If

  If color1 = 13 Then
    T31.Text = Tex1.Text
    T32.Text = Tex2.Text
    T33.Text = Tex3.Text
    T34.Text = Tex4.Text
    T35.Text = Tex5.Text
    T36.Text = Tex6.Text
    T37.Text = Tex7.Text
    T38.Text = Kp
    T39.Text = Ti
    T392.Text = Td
  End If
```

```
  m00 = (Kp * Td * b01)
  m01 = (Kp * Td * b02) + (Kp * b01)
  m02 = (Kp * Td * b03) + (Kp * b02) + (Kp * b01 / Ti)
  m03 = (Kp * b03) + (Kp * b02 / Ti)
  m04 = (Kp * b03 / Ti)

  a0 = a00 + m00
  a1 = a01 + m01
  a2 = a02 + m02
  a3 = a03 + m03
  a4 = m04

  b0 = m00
  b1 = m01
  b2 = m02
  b3 = m03
  b4 = m04

  ba0 = 0
  ba1 = ba1
  ba2 = b2 - a1 * ba1
  ba3 = b3 - a1 * ba2 - a2 * ba1
  ba4 = b4 - a1 * ba3 - a2 * ba2 - a3 * ba1

  c1 = 0: c2 = 1: c3 = 0: c4 = 0: c5 = ba1
  d1 = 0: d2 = 0: d3 = 1: d4 = 0: d5 = ba2
  e1 = 0: e2 = 0: e3 = 0: e4 = 1: e5 = ba3
  f1 = -a4: f2 = -a3: f3 = -a2: f4 = -a1: f5 = ba4

  t = 0: x1 = 0: x2 = 0: x3 = 0: x4 = 0
  z = 0.005
  tt = 10

  Cls
  For counter = 0 To tt Step z
  i1 = c1 * x1 + c2 * x2 + c3 * x3 + c4 * x4 + c5
  i2 = d1 * x1 + d2 * x2 + d3 * x3 + d4 * x4 + d5
  i3 = e1 * x1 + e2 * x2 + e3 * x3 + e4 * x4 + e5
  i4 = f1 * x1 + f2 * x2 + f3 * x3 + f4 * x4 + f5

  k1 = z * i1
  l1 = z * i2
  m1 = z * i3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
n1 = z * i4

k2 = z * (c1 * (x1 + 0.5 * k1) + c2 * (x2 + 0.5 * l1) + c3 * (x3
+ 0.5 * m1) + c4 * (x4 + 0.5 * n1) + c5)
l2 = z * (d1 * (x1 + 0.5 * k1) + d2 * (x2 + 0.5 * l1) + d3 * (x3
+ 0.5 * m1) + d4 * (x4 + 0.5 * n1) + d5)
m2 = z * (e1 * (x1 + 0.5 * k1) + e2 * (x2 + 0.5 * l1) + e3 * (x3
+ 0.5 * m1) + e4 * (x4 + 0.5 * n1) + e5)
n2 = z * (f1 * (x1 + 0.5 * k1) + f2 * (x2 + 0.5 * l1) + f3 * (x3 +
0.5 * m1) + f4 * (x4 + 0.5 * n1) + f5)

k3 = z * (c1 * (x1 + 0.5 * k2) + c2 * (x2 + 0.5 * l2) + c3 * (x3
+ 0.5 * m2) + c4 * (x4 + 0.5 * n2) + c5)
l3 = z * (d1 * (x1 + 0.5 * k2) + d2 * (x2 + 0.5 * l2) + d3 * (x3
+ 0.5 * m2) + d4 * (x4 + 0.5 * n2) + d5)
m3 = z * (e1 * (x1 + 0.5 * k2) + e2 * (x2 + 0.5 * l2) + e3 * (x3
+ 0.5 * m2) + e4 * (x4 + 0.5 * n2) + e5)
n3 = z * (f1 * (x1 + 0.5 * k2) + f2 * (x2 + 0.5 * l2) + f3 * (x3 +
0.5 * m2) + f4 * (x4 + 0.5 * n2) + f5)

k4 = z * (c1 * (x1 + 0.5 * k3) + c2 * (x2 + 0.5 * l3) + c3 * (x3
+ 0.5 * m3) + c4 * (x4 + 0.5 * n3) + c5)
l4 = z * (d1 * (x1 + 0.5 * k3) + d2 * (x2 + 0.5 * l3) + d3 * (x3
+ 0.5 * m3) + d4 * (x4 + 0.5 * n3) + d5)
m4 = z * (e1 * (x1 + 0.5 * k3) + e2 * (x2 + 0.5 * l3) + e3 * (x3
+ 0.5 * m3) + e4 * (x4 + 0.5 * n3) + e5)
n4 = z * (f1 * (x1 + 0.5 * k3) + f2 * (x2 + 0.5 * l3) + f3 * (x3 +
0.5 * m3) + f4 * (x4 + 0.5 * n3) + f5)

x1 = x1 + (k1 + 2 * k2 + 2 * k3 + k4) / 6
x2 = x2 + (l1 + 2 * l2 + 2 * l3 + l4) / 6
x3 = x3 + (m1 + 2 * m2 + 2 * m3 + m4) / 6
x4 = x4 + (n1 + 2 * n2 + 2 * n3 + n4) / 6

t = t + z

tt1 = t * 360
xx1 = 2850 - (x1 * 1760)
box.Circle (tt1, xx1), 10, QBColor(color1)

Next

color = color + 1
End Sub
```

----------Zigler_or2.Frm----------

```
Dim a00, a01, a02, a03
Dim b00, b01, b02, b03
Dim a0, a1, a2, a3, a4
Dim b0, b1, b2, b3, b4
Dim ba0, ba1, ba2, ba3, ba4
Dim c1, c2, c3, c4, c5
Dim d1, d2, d3, d4, d5
Dim e1, e2, e3, e4, e5
Dim f1, f2, f3, f4, f5
Dim x1, x2, x3, x4, i1, i2, i3, i4
Dim k1, l1, m1, n1, k2, l2, m2, n2
Dim k3, l3, m3, n3, k4, l4, m4, n4
Dim z, t, tt, tt1, xx1
Dim a, b, c, d, G, Kp, Sp
Dim t20, t30, t40, t50
Dim counter

Dim set1, time1, time, x10, y10
Dim DrawNow As Integer

Dim color, color1, col

Dim m00, m01, m02, m03

Dim pT, L, Kd, Ku

Dim out(5000) As Single
Dim err(5000) As Single
Dim rate(5000) As Single
Dim fuz_out(5000) As Single
Dim del_u(5000) As Single
Dim ave_err(5000) As Single
Dim un(5000) As Single
Dim ref, n, m, h
Dim tran
Dim counter1
Dim X(5000) As Single
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub box_MouseMove(Button As Integer, Shift As
Integer
, X As Single, Y As Single)
time = 10
x10 = (X - 630)
y10 = (4010 - Y)

time1 = (time / 4730)
set1 = (1 / 2330)
If DrawNow Then
    Line -(time1, set1)
    Circle (time1, set1), 50
  End If

End Sub

Private Sub Close_Click()
End
End Sub

Private Sub Ex_Click()
Tex1 = 1
Tex2 = 5
Tex3 = 1
Tex4 = 0
Tex5 = 0
Tex6 = 25
Tex7 = 5

Text2 = 1
Text3 = 0.5
Te1 = 1
Te2 = 400
End Sub

Private Sub Form_Load()
color = 1
End Sub

Private Sub Plot_Click()

a00 = Tex1.Text
a01 = Tex2.Text
a02 = Tex3.Text
a03 = Tex4.Text

b01 = Tex5.Text
b02 = Tex6.Text
b03 = Tex7.Text

Kp = Text2.Text
Kd = Text3.Text
Ku = Te1.Text
L = Te2.Text
ref = 1

If color = 1 Then
  color1 = 9
End If

If color = 2 Then
  color1 = 12
End If

If color = 3 Then
  color1 = 13
End If

If color1 = 9 Then
  T11.Text = Tex1.Text
  T12.Text = Tex2.Text
  T13.Text = Tex3.Text
  T14.Text = Tex4.Text
  T15.Text = Tex5.Text
  T16.Text = Tex6.Text
  T17.Text = Tex7.Text
  T18.Text = Text2.Text
  T19.Text = Text3.Text
  T191.Text = Te1.Text
  T192.Text = Te2.Text
End If

If color1 = 12 Then
  T21.Text = Tex1.Text
  T22.Text = Tex2.Text
  T23.Text = Tex3.Text
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    T24.Text = Tex4.Text
    T25.Text = Tex5.Text
    T26.Text = Tex6.Text
    T27.Text = Tex7.Text
    T28.Text = Text2.Text
    T29.Text = Text3.Text
    T291.Text = Te1.Text
    T292.Text = Te2.Text
End If

If color1 = 13 Then
    T31.Text = Tex1.Text
    T32.Text = Tex2.Text
    T33.Text = Tex3.Text
    T34.Text = Tex4.Text
    T35.Text = Tex5.Text
    T36.Text = Tex6.Text
    T37.Text = Tex7.Text
    T38.Text = Text2.Text
    T39.Text = Text3.Text
    T391.Text = Te1.Text
    T392.Text = Te2.Text
End If


m00 = (100 * b01)
m01 = (100 * b01) + (10 * b02)
m02 = (100 * b02) + (10 * b03)
m03 = (100 * b03)

a0 = a00 + m00
a1 = a01 + m01
a2 = a02 + m02
a3 = a03 + m03

b0 = m00
b1 = m01
b2 = m02
b3 = m03

ba0 = 0
ba1 = 0
ba2 = b2
ba3 = b3 - a1 * ba2

pT = 0.01


err(1) = 1
err(2) = 1
err(3) = 1

fuz_out(1) = 0
fuz_out(2) = 0

rate(1) = 0
rate(2) = 0
rate(3) = 0
rate(4) = 0

ave_err(1) = 0
ave_err(2) = 0
ave_err(3) = 0
ave_err(4) = 0

del_u(1) = 0
del_u(2) = 0

out(1) = 0
out(2) = 0
out(3) = 0

X(1) = 0
X(2) = 0
X(3) = 0

m = 1

t = 0: x1 = 0: x2 = 0: x3 = 0

z = 0.01

For counter = 1 To 1000 Step 1

  ave_err(m) = (err(m) + err(m - 1)) / pT
  rate(m) = (err(m) - err(m - 1)) / pT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    If ((Kd * Abs(rate(m))) <= (Kp * Abs(ave_err(m)))) And
((Kp * Abs(ave_err(m))) <= L) Then

      del_u(m) = 0.5 * L * Ku * (-Kd * rate(m) + Kp * ave_err
(m)) / (2 * L - Kp * Abs(ave_err(m)))
    End If

    If (((Kp * Abs(ave_err(m))) <= (Kd * Abs(rate(m)))) And
((Kd * Abs(rate(m))) <= L)) Then

      del_u(m) = 0.5 * L * Ku * (-Kd * rate(m) + Kp * ave_err
(m)) / (2 * L - Kp * Abs(rate(m)))
    End If

    If ((Kp * ave_err(m)) > L) And ((Kd * Abs(rate(m))) <= L)
Then

      del_u(m) = Ku * (-Kd * rate(m) + L) / 2
    End If

    If ((Kd * rate(m)) > L) And ((Kp * Abs(ave_err(m))) <= L)
Then

      del_u(m) = Ku * (Kp * ave_err(m) - L) / 2
    End If

    If ((Kp * ave_err(m)) < -L) And ((Kd * Abs(rate(m))) < L)
Then

      del_u(m) = Ku * (-Kd * rate(m) - L) / 2
    End If

    If ((Kd * rate(m)) < -L) And ((Kp * Abs(ave_err(m))) < L)
Then

      del_u(m) = Ku * (Kp * ave_err(m) + L) / 2
    End If

    If (((Kd * rate(m)) > L) And ((Kp * ave_err(m)) > L)) Or
(((Kd * rate(m)) < -L) And ((Kp * ave_err(m)) < -L)) Then

      del_u(m) = 0
    End If

    If ((Kd * rate(m)) > L) And ((Kp * ave_err(m)) < -L) Then

      del_u(m) = -L * Ku
    End If

    If ((Kp * ave_err(m)) > L) And ((Kd * rate(m)) < -L) Then

      del_u(m) = L * Ku
    End If

    fuz_out(m) = -fuz_out(m - 1) + Ku * del_u(m)

    un(1) = 0
    tran = 0
    n = 2

      For counter1 = 2 To m - 1 Step 1
        un(n) = 1
        h = pT * 0.5 * (un(n) + un(n - 1))
        tran = tran + fuz_out(m - n) * h
        n = n + 1
      Next

      out(m) = tran

    c1 = 0: c2 = 1: c3 = 0: c4 = ba1
    d1 = 0: d2 = 0: d3 = 1: d4 = ba2 * out(m)
    f1 = -a3: f2 = -a2: f3 = -a1: f4 = ba3 * out(m)

    i1 = c1 * x1 + c2 * x2 + c3 * x3 + c4
    i2 = d1 * x1 + d2 * x2 + d3 * x3 + d4
    i3 = f1 * x1 + f2 * x2 + f3 * x3 + f4

    k1 = z * i1
    l1 = z * i2
    m1 = z * i3

    k2 = z * (c1 * (x1 + 0.5 * k1) + c2 * (x2 + 0.5 * l1) + c3 *
(x3 + 0.5 * m1) + c4)
    l2 = z * (d1 * (x1 + 0.5 * k1) + d2 * (x2 + 0.5 * l1) + d3 *
(x3 + 0.5 * m1) + d4)
    m2 = z * (f1 * (x1 + 0.5 * k1) + f2 * (x2 + 0.5 * l1) + f3 *
(x3 + 0.5 * m1) + f4)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    k3 = z * (c1 * (x1 + 0.5 * k2) + c2 * (x2 + 0.5 * l2) + c3 *
(x3 + 0.5 * m2) + c4)
    l3 = z * (d1 * (x1 + 0.5 * k2) + d2 * (x2 + 0.5 * l2) + d3 *
(x3 + 0.5 * m2) + d4)
    m3 = z * (f1 * (x1 + 0.5 * k2) + f2 * (x2 + 0.5 * l2) + f3 *
(x3 + 0.5 * m2) + f4)

    k4 = z * (c1 * (x1 + 0.5 * k3) + c2 * (x2 + 0.5 * l3) + c3 *
(x3 + 0.5 * m3) + c4)
    l4 = z * (d1 * (x1 + 0.5 * k3) + d2 * (x2 + 0.5 * l3) + d3 *
(x3 + 0.5 * m3) + d4)
    m4 = z * (f1 * (x1 + 0.5 * k3) + f2 * (x2 + 0.5 * l3) + f3 *
(x3 + 0.5 * m3) + f4)

    x1 = x1 + (k1 + 2 * k2 + 2 * k3 + k4) / 6
    x2 = x2 + (l1 + 2 * l2 + 2 * l3 + l4) / 6
    x3 = x3 + (m1 + 2 * m2 + 2 * m3 + m4) / 6

    X(m) = x1

    tt1 = (m * 1.8)

    xx1 = 2850 - (X(m) * 1760)

    box.Circle (tt1, xx1), 10, QBColor(color1)

    err(m) = ref - X(m)

    m = m + 1

Next

color = color + 1
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้