

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การวิเคราะห์พีไอดีพารามิเตอร์ด้วยทฤษฎีทางเดินรากโดย MATLAB

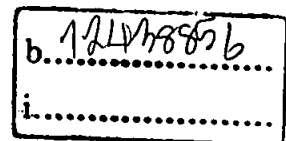
ANALYSIS PID PARAMETERS VIA ROOT LOCUS THEORY BY MATLAB



T123050

อัชฌ์ภณวุฒิ ไกรสุนทรเลิศภพ
AHUTPHONNAWUT KRAISUNTRONLERTPHOP

จพ.
๑53417
๑๕๕๕
เลขหมู่.....
เลขทะเบียน 123050
วัน,เดือน,ปี 18 ต.ค. 2555



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2555

KMITL-2012-EN-M-060-103

ANALYSIS PID PARAMETERS VIA ROOT LOCUS THEORY BY MATLAB

AHUTPHONNAWUT KRAISUNTRONLERTPHOP

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN INSTRUMENTATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2012
KMITL-2012-EN-M-060-103


COPYRIGHT 2012

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การวิเคราะห์ PID พารามิเตอร์ด้วยทฤษฎีทางเดินรากโดย MATLAB
Thesis Title Analysis PID Parameters via Root Locus Theory by MATLAB
นักศึกษา นายอัครวัฒน์ ไรสุทรเลิศภพ
รหัสประจำตัว 51060805
ปริญญา วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา วิศวกรรมการวัดคุม
อาจารย์ที่ปรึกษาวิทยานิพนธ์ รศ.สุพรรณ กุลพานิชย์
หมายเลขวิทยานิพนธ์ KMITL-2012-EN-M-060-103

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
รศ.ดร.พุฒศักดิ์	ชีวิสุวิทย์	
รศ.วิริยะ	กองรัตน์	
ดร.ยุทธพงศ์	ทัฬหะ	
รศ.ดร.ทวีพล	เชื้อสัจย์	
รศ.สุพรรณ	กุลพานิชย์	

วัน / เดือน / ปี ที่สอบ วันพฤหัสบดีที่ 31 พฤษภาคม พ.ศ. 2555 เวลา 10.00-12.00 น.
สถานที่สอบ ณ ห้องประชุมสาขาวิชาวิศวกรรมการวัดคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

คณะวิศวกรรมศาสตร์ รับรองแล้ว



(รองศาสตราจารย์ ดร.สุชัชวีร์ สุวรรณสวัสดิ์)

คณบดี คณะวิศวกรรมศาสตร์

วันที่ 31 พฤษภาคม พ.ศ. 2555

หัวข้อวิทยานิพนธ์	การวิเคราะห์ PID พารามิเตอร์ด้วยทฤษฎีทางเดินทางโดย MATLAB
นักศึกษา	นายอัษฎมภณวุฒิ ไกรสุนทรเลิศภพ
รหัสประจำตัว	51060805
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมการวัดและควบคุม
พ.ศ.	2555
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รองศาสตราจารย์ สุพรรณ กุลพานิश्य

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้เป็นการนำเสนอวิธีการควบคุมแบบ พีไอดี ที่สามารถกำหนดตัวแปรสำคัญได้เช่น ค่าฟังก์ชันการถ่ายโอน เปอร์เซนต์การพุ่งเกิน และช่วงเวลาเข้าที่ ให้สอดคล้องกับกระบวนการ โดยอาศัยทฤษฎีทางเดินราก เพื่อนำไปสู่การหาค่าพารามิเตอร์ K_p , K_i และ K_d ที่ได้มาจากการคำนวณของโปรแกรม MATLAB โดยมีส่วนของโปรแกรมที่สร้างขึ้นที่เรียกว่า M-file เพื่อความสะดวกในการกำหนดคุณสมบัติของกระบวนการ จากนั้นนำค่าตัวแปรกระบวนการไปจำลองการทำงานบนโปรแกรมคอมพิวเตอร์ Simulink เพื่อสังเกตผลตอบสนองของกระบวนการว่าเป็นไปตามค่าตัวแปรที่กำหนดหรือไม่ และนำค่าพารามิเตอร์ K_p , K_i และ K_d ไปควบคุมกระบวนการจริงและเปรียบเทียบผลของการจำลองและผลของการควบคุมกระบวนการจริง โดยจะใช้การควบคุมอุณหภูมิเป็นกระบวนการทดลอง

Thesis Title	Analysis PID Parameter via Root locus theory by MATLAB
Student	Mr. Ahutphonawut Kraisuntronlertphop
Student ID.	51060805
Degree	Master of Engineering
Program	Instrumentation and Control Engineering
Year	2012
Thesis Advisor	Associate Professor Supan Gunpanich

ABSTRACT

This thesis is presented the PID control method. It can be set important parameters is transfer function, percent overshoot, settling time and consistent base on root locus theory. In order to determine the parameters K_p , K_i and K_d obtained from the calculation of the program MATLAB. The part of the program that created the M-file to facilitate determine competence process. And then, it used to simulate the processes running on a computer program Simulink, it to observe the response of the process is determine the variables or not. The parameters K_p , K_i and K_d to control the process and compare the results of the simulation and the actual process of regulation, It is used to control the experimental process.

กิตติกรรมประกาศ

งานวิจัยและการเรียบเรียงวิทยานิพนธ์ฉบับนี้จะไม่สำเร็จลุล่วงลงได้ ต้องประสบกับปัญหามากมายประการ ทั้งในส่วนของวิทยานิพนธ์และส่วนของผู้จัดทำเอง แต่ด้วยความช่วยเหลือและความอนุเคราะห์จากบุคคลหลายท่าน ทำให้ผู้จัดทำสามารถฟันฝ่าอุปสรรคเหล่านั้นไปได้ด้วยดี จึงต้องขอกล่าวขอบพระคุณทุกท่านมา ณ ที่นี้

ขอขอบพระคุณ รองศาสตราจารย์ สุพรรณ กุลพานิชย์ อาจารย์ผู้ควบคุมวิทยานิพนธ์ ข้าพเจ้าผู้รู้สึกรับซึ่ง ในความอนุเคราะห์จากท่าน

ขอขอบพระคุณ คณาจารย์สาขาวิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกท่านที่ประสิทธิ์ประสาทวิชาให้กับข้าพเจ้า

ขอขอบคุณเพื่อนๆ พี่ๆ และน้องๆ ภาควิชาวิศวกรรมการวัดและควบคุม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกท่านที่ให้คำแนะนำต่างและคอยให้กำลังใจที่ดีเสมอมา

ขอขอบคุณบัณฑิตวิทยาลัย คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ให้ความช่วยเหลือในเรื่องต่างๆ

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และพี่สาวของข้าพเจ้าที่เป็นกำลังใจ และให้การสนับสนุนในทุกเรื่องๆตลอดมา ซึ่งเป็นผลทำให้ข้าพเจ้าสามารถทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี

คุณค่าและประโยชน์อันพึงมีจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบแต่บิดา มารดา ตลอดจนผู้มีพระคุณทุกท่าน

อัทธมภณวุฒิ ไกรสุนทรเลิศภาพ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 จุดมุ่งหมายและความสำคัญของการศึกษา	1
1.3 ขอบเขตการวิจัย	1
1.4 ส่วนประกอบของวิทยานิพนธ์	1
บทที่ 2 กระบวนการและทฤษฎีที่เกี่ยวข้อง	3
2.1 การควบคุมแบบ พีไอดี (PID Control)	3
2.1.1 การควบคุมแบบสัดส่วน	3
2.1.2 การควบคุมแบบปริพันธ์	4
2.1.3 การควบคุมแบบอนุพันธ์	4
2.1.4 การควบคุมแบบ PI	6
2.1.5 การควบคุมแบบ PD	6
2.1.6 การควบคุมแบบ PID	7
2.2 ทฤษฎีทางเดินราก (Root Locus)	8
2.2.1 แนวคิดวิเคราะห์ทางเดินราก	8
2.2.2 การเขียนทางเดินของราก	8
2.3 ชุดจำลองอนุกรม	13
2.4 โปรแกรม MatLab	15
2.4.1 การวิเคราะห์เกี่ยวกับโพลีโนเมียล	16
2.4.1.1 การคูณของโพลีโนเมียลด้วยโพลีโนเมียล	16
2.4.1.2 การหารโพลีโนเมียลด้วยโพลีโนเมียล	16
2.4.2 การใช้ MatLAB วิเคราะห์เกี่ยวกับระบบควบคุม	16

สารบัญ (ต่อ)

	หน้า
2.4.2.1 Transfer Function กับ zero-pole-gain	16
2.4.2.2 การหา system response จาก transfer function	17
2.4.2.3 การหา Root-Locus Analysis	18
2.5 ทฤษฎีโปรแกรม LabVIEW	18
อุปกรณ์ ดาต้า แอควิซชัน	20
บทที่ 3 การออกแบบตัวควบคุมและโปรแกรม M-file.....	22
3.1 การหาค่าสมการคุณลักษณะขุดจำลองอนุหภูมิ	22
3.2 หลักการออกแบบโปรแกรมและกระบวนการทำงาน M-file	23
3.3 การออกแบบการจำลองการควบคุมแบบ พีไอดี (PID Control)	26
3.4 การออกแบบการควบคุมระบบจริงโดยใช้โปรแกรม Labview	27
บทที่ 4 ผลการจำลองและผลการทดลองกระบวนการ	31
4.1 ผลการหาค่าพารามิเตอร์ควบคุมแบบ พีไอดี	31
4.2 ผลการจำลองกระบวนการแบบ	31
Proportional Integral Derivative (PID)	
4.3 ผลการควบคุมกระบวนการจริงแบบ	42
Proportional Integral Derivative (PID)	
4.4 สรุปผลการทดลอง	44
บทที่ 5 สรุปผลการวิจัย	45
5.1 สรุปผลการวิจัย	45
5.2 ปัญหาและแนวทางแก้ไขปัญหางานวิจัย	45
5.3 แนวทางในการพัฒนาวิจัย	45
บรรณานุกรม	46

สารบัญ (ต่อ)

หน้า

ภาคผนวก	47
ภาคผนวก ก	48
ภาคผนวก ข	52
ภาคผนวก ค	54
ภาคผนวก ง	59
ภาคผนวก จ	66
ประวัติผู้เขียน	72

สารบัญรูป

รูปที่	หน้า
2.1 การตอบสนองของคอนโทรลเลอร์แบบสัดส่วน	3
2.2 การตอบสนองของคอนโทรลเลอร์แบบปริพันธ์	4
2.3 การตอบสนองของคอนโทรลเลอร์แบบอนุพันธ์	5
2.4 ผลการควบคุมแบบพีไอ	6
2.5 ผลของการควบคุมแบบพีดี	7
2.6 ผลของการควบคุมแบบพีไอดี	7
2.7 ระบบควบคุมที่มีการป้อนกลับ	8
2.8 SOLIT STATE RELAY	14
2.9 SOLIT STATE RELAY	14
2.10 Omron Temperature Simulation	15
2.11 วงจร Closed loop transfer function	18
2.12 Front Panel	20
3.1 แสดงวิธีการ Open loop เพื่อหาสมการคุณลักษณะ	22
3.2 แสดงการจำลองหาค่าสมการคุณลักษณะของระบบ	22
3.3 ผลการปรับหาค่าสมการคุณลักษณะของระบบ	23
3.4 แสดงกระบวนการทำงานของโปรแกรม M-file	26
3.5 แสดงฟังก์ชัน While Loop	27
3.6 แสดงฟังก์ชัน Format into file	28
3.7 แสดงฟังก์ชัน พีไอดี	29
3.8 แสดงฟังก์ชัน Wait Until Next ms Multiple	29
3.9 แสดงการทำงานและควบคุมกระบวนการ	30
4.1 แสดงผลของ Pole บนเส้นทางเดินราก	31
4.2 แสดงตำแหน่งโพลที่ $0.974+0.0336i$ ค่าพารามิเตอร์ $K_p = 0.3486$, $K_i = 0.0069$, $K_d = 0.4671$, $OS\% = 10\%$, $T_s = 300$ sec	32
4.3 แสดงผลการจำลองที่ 40 องศา $K_p = 0.3486$, $K_i = 0.0069$, $K_d = 0.4671$, $OS\% = 10\%$, $T_s = 300$ sec	32
4.4 แสดงผลการจำลองที่ 45 องศา $K_p = 0.3486$, $K_i = 0.0069$, $K_d = 0.4671$ $OS\% = 10\%$, $T_s = 300$ sec	33

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.5 แสดงผลการจำลองที่ 50 องศา $K_p = 0.3486$, $K_i = 0.0069$, $K_d = 0.4671$ OS% = 10%, $T_s = 300$ sec	33
4.6 แสดงตำแหน่งโพลที่ $0.974+0.0531i$ ค่าพารามิเตอร์ $K_p = 0.5424$, $K_i = 0.0137$, $K_d = 1.1713$, OS% = 20%, $T_s = 300$ sec	34
4.7 แสดงผลการจำลองที่ 35 องศา $K_p = 0.5424$, $K_i = 0.0137$, $K_d = 1.1713$, OS% = 20%, $T_s = 300$ sec	34
4.8 แสดงผลการจำลองที่ 40 องศา $K_p = 0.5424$, $K_i = 0.0137$, $K_d = 1.1713$, OS% = 20%, $T_s = 300$ sec	35
4.9 แสดงผลการจำลองที่ 45 องศา $K_p = 0.5424$, $K_i = 0.0137$, $K_d = 1.1713$, OS% = 20%, $T_s = 300$ sec	35
4.10 แสดงผลการจำลองที่ 50 องศา $K_p = 0.5424$, $K_i = 0.0137$, $K_d = 1.1713$, OS% = 20% $T_s = 300$ sec	36
4.11 แสดงตำแหน่งโพลที่ $0.988+0.0161i$ ค่าพารามิเตอร์ $K_p = 0.1474$, $K_i = 0.0028$, $K_d = 0.9854$, OS% = 10%, $T_s = 700$ sec	36
4.12 แสดงผลการจำลองที่ 35 องศา $K_p = 0.1474$, $K_i = 0.0028$, $K_d = 0.9854$, OS% = 10%, $T_s = 700$ sec	37
4.13 แสดงผลการจำลองที่ 40 องศา $K_p = 0.1474$, $K_i = 0.0028$, $K_d = 0.9854$, OS% = 10%, $T_s = 700$ sec	37
4.14 แสดงผลการจำลองที่ 45 องศา $K_p = 0.1474$, $K_i = 0.0028$, $K_d = 0.9854$, OS% = 10%, $T_s = 700$ sec	38
4.15 แสดงผลการจำลองที่ 50 องศา $K_p = 0.1474$, $K_i = 0.0028$, $K_d = 0.9854$, OS% = 10% $T_s = 700$ sec	38
4.16 แสดงตำแหน่งโพลที่ $0.988+0.0161i$ ค่าพารามิเตอร์ $K_p = 0.1474$, $K_i = 0.0028$, $K_d = 0.9854$, OS% = 20%, $T_s = 700$ sec	39
4.17 แสดงผลการจำลองที่ 35 องศา $K_p = 0.1474$, $K_i = 0.0028$, $K_d = 0.9854$, OS% = 20%, $T_s = 700$ sec	40
4.18 แสดงผลการจำลองที่ 40 องศา $K_p = 0.1474$, $K_i = 0.0028$, $K_d = 0.9854$, OS% = 20%, $T_s = 700$ sec	40

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.19 แสดงผลการจำลองที่ 45 องศา $K_p = 0.1474$, $K_i = 0.0028$, $K_d = 0.9854$, OS% = 20%, $T_s = 700$ sec	41
4.20 แสดงผลการจำลองที่ 50 องศา $K_p = 0.1474$, $K_i = 0.0028$, $K_d = 0.9854$, OS% = 20%, $T_s = 700$ sec	41
4.21 แสดงการควบคุมกระบวนการที่ Set point = 35 °C	42
4.22 แสดงการควบคุมกระบวนการที่ Set point = 40 °C	42
4.23 แสดงการควบคุมกระบวนการที่ Set point = 45 °C	43
4.24 แสดงการควบคุมกระบวนการที่ Set point = 50 °C	43

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

ในปัจจุบันนี้การควบคุมกระบวนการเป็นสิ่งจำเป็นสำหรับงานอุตสาหกรรมหลายอย่างทำให้ การศึกษาการควบคุมเป็นที่แพร่หลาย โดยการควบคุมที่รู้จักโดยทั่วไป คือ การควบคุมแบบพีไอดี (PID)

ในการควบคุมนี้จะใช้ชุดทดลองอุณหภูมิ OMRON TEMPERATURE MODEL เพื่อใช้ในการศึกษาของผลตอบสนองของการควบคุมแบบพีไอดี (PID) โดยการเขียน M-file หาค่าพารามิเตอร์ต่างๆ และใช้โปรแกรม MathLab เพื่อจำลองผลการตอบสนองของระบบ จากนั้นจึงใช้โปรแกรม LabVIEW เพื่อควบคุมกระบวนการจริงโดยใช้ค่าพารามิเตอร์จากโปรแกรม M-file

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1. ใช้สำหรับศึกษาและทดลองการควบคุมแบบพีไอดี
2. เพื่อศึกษาการหาค่าพารามิเตอร์ของการควบคุมแบบพีไอดีโดยใช้โปรแกรม m-file
3. เพื่อศึกษาการใช้โปรแกรม Simulink เพื่อจำลองการควบคุมกระบวนการ
4. ใช้โปรแกรม LabVIEW เพื่อควบคุมกระบวนการจริง โดยใช้ค่าพารามิเตอร์ที่ได้จากโปรแกรม m-file

1.3 ขอบเขตและขั้นตอนการวิจัย

1. ทำการหาสมการคุณลักษณะของชุดจำลองอุณหภูมิ
2. สร้างแบบจำลองทางคณิตศาสตร์ของชุดทดลองอุณหภูมิ
3. ทำการหาค่าพารามิเตอร์ที่ใช้ควบคุมระบบ โดยโปรแกรม M-file
4. จำลองผลการตอบสนองของระบบการควบคุมแบบพีไอดีด้วยโปรแกรม Simulink
5. เปรียบเทียบผลการทดลองที่ได้จากการทดสอบกระบวนการจริงโดยใช้โปรแกรม LabView กับการจำลองกระบวนการโดยใช้โปรแกรม MathLab

1.4 รายละเอียดของวิทยานิพนธ์

วิทยานิพนธ์ แบ่งเป็น 5 บท รายละเอียดแต่ละบท ดังนี้

บทที่ 1 ความเป็นมาและความสำคัญของหัวข้องานวิจัย ความมุ่งหมาย วัตถุประสงค์ของการศึกษาวิจัย ขอบเขตและขั้นตอนในการวิจัยรวมถึงรายละเอียดของวิทยานิพนธ์แต่ละบท

บทที่ 2 รายละเอียดและกระบวนการของชุดจำลองอุณหภูมิ องค์ประกอบทั้งหมดของกระบวนการ และโปรแกรมที่ใช้ในการทดลอง รวมถึงทฤษฎีพื้นฐานที่เกี่ยวข้องกับงานวิจัย

บทที่ 3 การออกแบบตัวควบคุมแบบพีไอดี ใช้ทฤษฎีพื้นฐานและการเขียนโปรแกรม M-file เพื่อใช้หาค่าพารามิเตอร์ที่ใช้ควบคุมกระบวนการ

บทที่ 4 การจำลองและการทดลอง กล่าวถึงขั้นตอนการจำลองกระบวนการและวิธีการทดลองทั้งหมดพร้อมนำเสนอผลการทดลองของกระบวนการ

บทที่ 5 สรุปผลการทดลอง ข้อเสนอแนะ ปัญหาที่พบในงานวิจัย พร้อมทั้งเสนอแนะ
แนวทางแก้ปัญหาและแนวทางการพัฒนา

บทที่ 2

กระบวนการและทฤษฎีที่เกี่ยวข้อง

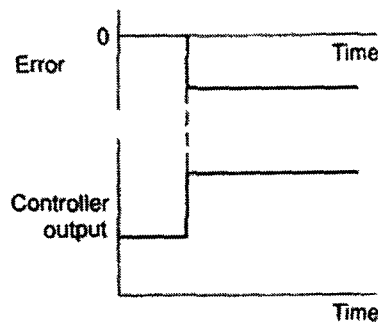
ในบทนี้กล่าวถึงกระบวนการทำงาน อุปกรณ์ภายในชุดจำลองอุณหภูมิ ทฤษฎีการควบคุม โปรแกรมที่ใช้ในการจำลอง ทฤษฎีการควบคุมแบบพีไอดี[11],[12] รวมถึงการใช้ทฤษฎีแบบทางเดิน ราก[3],[5],[6] รวมถึงโปรแกรมที่ใช้ควบคุมระบบจริงของวิทยานิพนธ์นี้

2.1 การควบคุมแบบพีไอดี [11], [12]

การควบคุมแบบพีไอดี (PID Control) เป็นการควบคุมแบบรวมกัน โดยอาศัยสัดส่วนของการขยายสัญญาณความผิดพลาด (Proportional Control) ปริพันธ์ (Integral Control) และอนุพันธ์ (Derivative Control) เพื่อใช้เป็นตัวควบคุมระบบ โดยมีจุดประสงค์เพื่อให้การตอบสนองของระบบมีเสถียรภาพที่ดีขึ้น ในการควบคุมแบบพีไอดีจะเป็นการควบคุมแบบลูปปิด (Close Loop Control) หรือการควบคุมแบบป้อนกลับ (Feedback Control) ซึ่งประกอบด้วยส่วนการควบคุมที่สำคัญ 3 ส่วนด้วยกัน คือ

2.1.1 การควบคุมแบบสัดส่วน [15]

การควบคุมแบบสัดส่วน (Proportional Control) เป็นเพียงการขยายสัญญาณความผิดพลาดเท่านั้น การที่เราได้สัญญาณความผิดพลาดขนาดใหญ่ที่เวลาหนึ่ง จะทำให้เกิดเอาต์พุตที่มีขนาดใหญ่จากคอนโทรลเลอร์ในเวลาหนึ่ง อย่างไรก็ตามการที่เรากำหนดให้อัตราขยาย (Gain) คงที่ ในทางปฏิบัติเราอาจจะกำหนดไว้ในบางช่วงของสัญญาณความผิดพลาดเท่านั้น ซึ่งเราอาจกำหนดให้คอนโทรลเลอร์ของเรามีค่าเอาต์พุตไม่น้อยกว่าค่าหนึ่งและไม่มากเกินไปกว่าค่าหนึ่งก็ทำให้ได้เอาต์พุตที่ออกจากคอนโทรลเลอร์แบบขยายสัญญาณความผิดพลาด โดยจะขึ้นกับขนาดของความผิดพลาดในขณะที่เรากำลังพิจารณา เป็นการกำหนดการทำงานของเอาต์พุตให้เป็นสัดส่วนเปอร์เซ็นต์กับค่าความผิดพลาด (Error) หรือการเปลี่ยนแปลงของค่าที่วัดได้

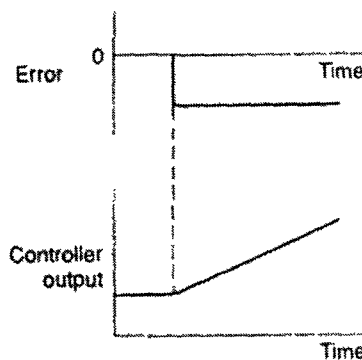


รูปที่ 2.1 การตอบสนองของคอนโทรลเลอร์แบบสัดส่วน

จากรูปที่ 2.1 นั้น ในทางปฏิบัติผลตอบสนองของคอนโทรลเลอร์แบบสัดส่วนจะให้ค่าเอาต์พุตเข้าใกล้ค่าหนึ่งที่ไม่ใช่ค่าเป้าหมายจริง เราเรียกค่านี้ว่า ค่าออฟเซต (Offset) ซึ่งการเพิ่มค่าของการควบคุมแบบสัดส่วน (Proportional-Gain) ทำเพื่อลดไรซ์ไทม์ (Rise Time) ของระบบ

2.1.2 การควบคุมแบบปริพันธ์ [15]

การควบคุมแบบปริพันธ์ (Integral Control) ระบบควบคุมแบบปริพันธ์ จะเพิ่มจำนวนโพลให้กับระบบควบคุมและเพิ่มอันดับ ของระบบ จาก อันดับ 0 เป็น อันดับ 1 ซึ่งทำให้ระบบมีความผิดพลาดที่สภาวะคงตัวเป็นศูนย์เทียบต่ออินพุตแบบขั้นบันได อย่างไรก็ตามการเพิ่มโพลที่ $S = 0$ และไม่มี การเพิ่มศูนย์ให้กับระบบควบคุม จะทำให้ความแตกต่างระหว่างจำนวนโพล (n) และจำนวนศูนย์ (m) เพิ่มขึ้นอีก 1 ซึ่งจะมีผลทำให้ความเสถียรสัมพันธ์ของระบบลดลง ซึ่งระบบควบคุมแบบปริพันธ์ จะเข้าไปช่วยกำจัดค่าออฟเซต (Offset) ที่ยังคงมีอยู่ให้ระบบเข้าสู่เป้าหมาย โดยค่าเอาต์พุตที่ออกมา จะขึ้นอยู่กับค่าเวลาการปริพันธ์ (Integral Time) ที่กำหนดขึ้นมาตั้งแต่ต้น หากกำหนดให้ค่าเวลาการปริพันธ์ (Integral Time) น้อย ระบบจะเข้าสู่ค่าเป้าหมาย ได้อย่างรวดเร็วแต่จะเกิดการกระเพื่อมของระบบมากด้วย และหากกำหนดค่าเวลาการปริพันธ์ (Integral Time) มากจะเกิดการกระเพื่อม น้อย แต่จะใช้เวลานานกว่าระบบจะเข้าสู่ค่าเป้าหมาย

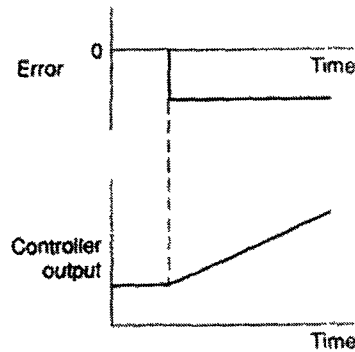


รูปที่ 2.2 การตอบสนองของคอนโทรลเลอร์แบบปริพันธ์

จากรูปที่ 2.2 หลังจากเลือกค่าการควบคุมแบบสัดส่วน (Proportional-Gain) ที่ให้ค่าไรซ์ไทม์ (Rise Time) ที่เหมาะสมแล้ว จึงเพิ่มส่วนของปริพันธ์ (Integral-Gain) เข้าไปเพื่อกำจัดค่าผิดพลาดที่ภาวะคงตัว (Steady State Error) โดยค่าปริพันธ์ที่น้อยเกินไปจะทำให้ระบบไม่เสถียร ส่วนการเพิ่มค่าปริพันธ์จะช่วยลดไรซ์ไทม์ของระบบ แต่ค่าการพุ่งเกิน (Overshoot) จะสูงขึ้น

2.1.3 การควบคุมแบบอนุพันธ์ [15]

การควบคุมแบบอนุพันธ์ (Derivative Control) การควบคุมแบบอนุพันธ์นี้ เอาต์พุตจะเป็นสัดส่วนกับอัตราการเปลี่ยนแปลงความผิดพลาดเทียบกับเวลา ในกรณีที่มีการรบกวนระบบจากภายนอก (Disturbance) เป็นผลให้ระบบมีการเปลี่ยนแปลงอย่างทันทีทันใด เราควรจะใช้การควบคุมแบบอนุพันธ์ ซึ่งจะมีการตอบสนองที่รวดเร็วเป็นผลให้ระบบเข้าสู่เป้าหมายได้รวดเร็วขึ้น



รูปที่ 2.3 การตอบสนองของคอนโทรลเลอร์แบบอนุพันธ์

จากรูปที่ 2.3 การเพิ่มค่าอนุพันธ์ (Derivative-Gain) จะช่วยลดค่าการพุ่งเกิน (Overshoot) ของระบบแต่ไรส์ไทม์ (Rise Time) จะมากขึ้น ถ้าหากว่าเป็นระบบอันดับ 1 หรือสูงกว่า การควบคุมแบบอนุพันธ์จะลด s ในเทอมส่วนลง และลดอันดับของระบบลง 1 อันดับอย่างไรก็ตามเราได้กล่าวก่อนหน้านี้แล้วว่า การควบคุมแบบอนุพันธ์นี้มักจะไม่ใช้เพียงลำพัง แต่เราจะใช้ร่วมกับการควบคุมแบบอื่น เพราะเมื่อเราใช้การควบคุมแบบอนุพันธ์ จะทำให้เราเพิ่มความเร็วในการตอบสนองของระบบต่อความผิดพลาดที่เกิดขึ้นค่าที่ใช้ในการวิเคราะห์ผลตอบสนองชั่วคราว (Transient Response) ของระบบควบคุมแบบป้อนกลับ (Feedback Control System) ที่สำคัญ มีดังนี้

- 1) ค่าผิดพลาดในภาวะคงตัว (Steady State Error) คือ ความแตกต่างของค่าเอาต์พุตเมื่อเทียบกับเป้าหมาย (Set Point) ที่เวลาเข้าสู่สภาวะนิ่ง หรือเวลาที่ระบบอยู่ในภาวะคงตัว

$$\text{Error} = \text{ค่าเป้าหมาย (Set Point)} - \text{ค่าเอาต์พุตที่วัดได้ (Output)} \quad (2.1)$$

- 2) ค่าเปอร์เซ็นต์การพุ่งเกิน (Percent Overshoot) คือ ค่าเอาต์พุตสูงสุดเมื่อเทียบกับค่าเป้าหมาย (Set Point) ซึ่งการเพิ่มค่าอนุพันธ์ (Derivative-Gain) นั้นจะช่วยลดค่าเปอร์เซ็นต์การพุ่งเกินของระบบ แต่ไรส์ไทม์ (Rise Time) จะมากขึ้น ส่วนการเพิ่มค่าปริพันธ์ (Integral-Gain) จะช่วยลดไรส์ไทม์ของระบบแต่ค่าการพุ่งเกิน (Overshoot) จะสูงขึ้น

$$\% OS = \frac{\text{Maximum overshoot}}{\text{Final desired value}} \times 100 \quad (2.2)$$

- 3) ไรส์ไทม์ (Rise Time) เป็นช่วงเวลาตั้งแต่เอาต์พุตมีค่าเป็น 10 เปอร์เซ็นต์จนถึง 90 เปอร์เซ็นต์ ของค่าเป้าหมาย (Set Point) ซึ่งการเพิ่มส่วนของการขยายสัญญาณความผิดพลาด โดยทำเพื่อลดไรส์ไทม์ (Rise Time) ของระบบ ส่วนการเพิ่มค่าการปริพันธ์ (Integral-Gain) นั้นก็จะช่วยลดไรส์ไทม์ของระบบได้ แต่ค่าการพุ่งเกิน (Overshoot) จะสูงขึ้น

- 4) ช่วงเวลาที่เข้าสู่ค่าเป้าหมาย (Settling Time) เป็นช่วงเวลาตั้งแต่เริ่มต้นจนกระทั่งการแกว่งของเอาต์พุตลดลงอยู่ในขอบเขตที่กำหนด ซึ่งโดยปกติแล้วขอบเขตนี้จะอยู่ในช่วง 2% ของค่าเป้าหมาย (Set Point)

2.1.4 การควบคุมแบบ PI [15]

การควบคุมแบบพีไอ (Proportional Integral Control) มี Integral Action หรือ Reset Action เพิ่มเข้ากับการควบคุมแบบพี ระบบควบคุมส่วนมากมักจะใช้การควบคุมแบบพีไอ ในการควบคุมอัตราการไหลและแรงดันของของเหลวสมการของสัญญาณควบคุมของการควบคุมแบบพีไอ แสดงได้ ดังนี้

$$MV(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt \quad (2.1)$$

เมื่อ T_i คือ Integral Time หรือ Reset Time



รูปที่ 2.4 ผลการควบคุมแบบพีไอ

จากรูปที่ 2.4 ขนาดของ Integral Action ขณะใดขณะหนึ่งจะไม่สัมพันธ์กับค่าความผิดพลาดขณะนั้น แต่เป็นสัดส่วนโดยตรงกับขนาดของความผิดพลาดสะสม ผลของ Integral Action เหมือนการปรับค่าไบแอสจนกระทั่งค่าความผิดพลาดหมดไป และผลการตอบสนองของ Integral Action ขึ้นอยู่กับการปรับค่า T_i ถ้า T_i มีค่าน้อยการตอบสนองจะมีค่ามาก

ในระบบควบคุมจะพบว่า Integral Action สามารถกำจัดออฟเซตให้หมดไป โดยมีความไวในการกำจัดออฟเซตขึ้นอยู่กับการปรับค่า T_i ถ้า T_i มีค่าน้อยการกำจัดออฟเซตจะทำได้รวดเร็วแต่อย่างไรก็ตาม ถ้าค่า T_i มีค่าน้อยเกินไปจะทำให้ตัวแปรโปรเซสเกิดการแกว่ง และการเพิ่ม Integral Action จะเหมือนเป็นการเพิ่ม Capacity Lag เข้าไปในระบบควบคุม ทำให้ตัวแปรโปรเซสเปลี่ยนแปลงเข้าสู่ค่าเป้าหมายได้ช้าลง

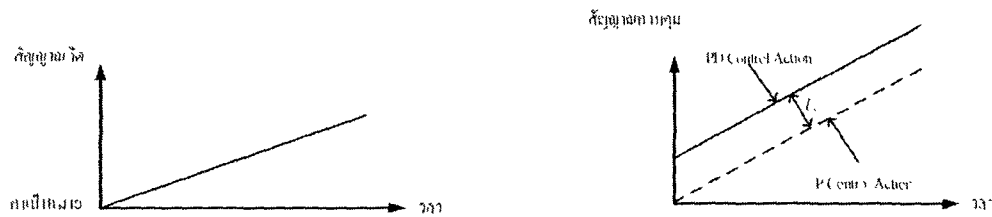
2.1.5 การควบคุมแบบ PD [15]

การควบคุมแบบพีดี (Proportional Derivative Control) จะช่วยปรับปรุงผลตอบสนองชั่วขณะของระบบ ทำให้การควบคุมเข้าสู่เป้าหมายที่ต้องการรวดเร็วยิ่งขึ้น เนื่องจากการควบคุมแบบพีดีมีคุณสมบัติในการคาดการณ์ล่วงหน้าถึงการเปลี่ยนแปลงของค่าความผิดพลาดที่เกิดขึ้น และ

สร้างสัญญาณควบคุมเพื่อลดค่าผิดพลาดล่วงหน้าทันทีที่สมการของสัญญาณควบคุมของการควบคุมแบบพีดี แสดงได้ดังนี้

$$MV(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt} \quad (2.2)$$

เมื่อ T_d คือ Derivative Time หรือ Rate Time



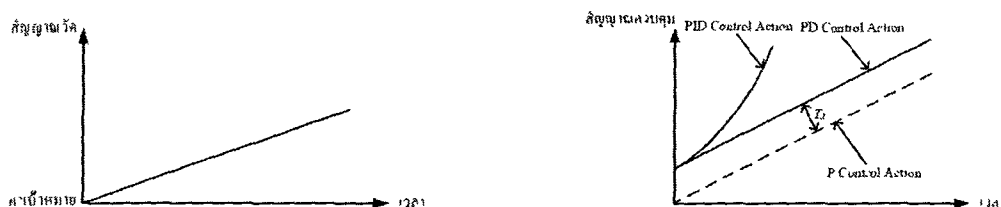
รูปที่ 2.5 ผลของการควบคุมแบบพีดี

จากรูปที่ 2.5 ผลตอบสนองของ Derivative Action จะเป็นสัดส่วนโดยตรงกับอัตราการเปลี่ยนแปลงของค่าความผิดพลาด กรณีที่ค่าเป้าหมายมีค่าคงที่และสัญญาณวัดไม่เปลี่ยนแปลง Derivative Action จะไม่มีผล เมื่อสัญญาณวัดเกิดการเปลี่ยนแปลง Derivative Action จะเพิ่มหรือลดสัญญาณควบคุมตามอัตราการเปลี่ยนแปลงของสัญญาณวัด

2.1.6 การควบคุมแบบ PID [15]

ในการควบคุมแบบพีไอ จะทำให้ตัวควบคุมตอบสนองต่อความผิดพลาดช้าลง กรณีที่โปรเซสช้าตอบสนองช้าจำเป็นต้องเพิ่ม D หรือ derivative Action เพื่อลดความช้าของระบบควบคุม การควบคุมแบบนี้เรียกว่า การควบคุมแบบพีไอดี จะพบการควบคุมแบบนี้ในงานควบคุมอุณหภูมิเป็นส่วนใหญ่สมการของสัญญาณควบคุมของการควบคุมแบบพีไอดี (Proportional Integral Derivative Control) แสดงได้ดังนี้

$$MV(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt} \quad (2.3)$$



รูปที่ 2.6 ผลของการควบคุมแบบพีไอดี

จากรูปที่ 2.6 เมื่อเพิ่ม Derivative Action เข้าไปในการควบคุมแบบพีไอ จะทำให้ผลตอบสนองต่อสิ่งรบกวนไวขึ้นและสัญญาณวัดเปลี่ยนแปลงสู่ค่าเป้าหมายไว อย่างไรก็ตามถ้าปรับตั้งค่า T_d มากเกินไปสัญญาณวัดหรือตัวแปรกระบวนการอาจแกว่งได้

2.2 ทฤษฎีทางเดินราก [16]

เส้นทางเดินของราก (Root-Locus Technique) คือ การตรวจสอบสมรรถนะของระบบควบคุม เมื่อพารามิเตอร์ของระบบหนึ่งตัวหรือมากกว่านั้นแปรค่าไปเกินกว่าขอบเขตที่กำหนดให้เป็นสิ่งจำเป็นอย่างหนึ่งในการพิจารณาออกแบบระบบควบคุม ซึ่งทราบกันดีอยู่แล้วว่าสมการคุณลักษณะของระบบมีบทบาทสำคัญในการวิเคราะห์ระบบเชิงเส้น วิธีเส้นทางเดินของราก (Root-Locus) จึงเป็นวิธีที่ศึกษาถึงพฤติกรรมทางเดินของรากของสมการคุณลักษณะของระบบ เมื่อพารามิเตอร์แปรค่าไปจึงเป็นวิธีที่มีประสิทธิภาพวิธีหนึ่งในการศึกษาระบบควบคุม

นอกจากนี้เรายังสามารถนำวิธีทางเดินของรากไปประยุกต์เพื่อหารากของสมการทางพีชคณิตที่มีอันดับสูงๆ ได้อีกด้วย โดยทั่วไปวิธีของเส้นทางเดินของรากจะใช้สำหรับพิจารณาทางเดินของรากเมื่อพารามิเตอร์ตัวหนึ่งแปรค่าไป แต่ในกรณีที่พารามิเตอร์มากกว่าหนึ่งตัวแปรก็สามารถใช้วิธีทางเดินของรากได้เช่นกัน

2.2.1 แนวคิดของการวิเคราะห์ทางเดินของราก

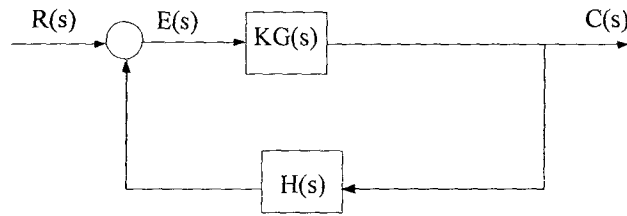
ทางเดินของรากของสมการคุณลักษณะหรือโพลของระบบแบบปิดจะพิจารณาจากการแปรค่าของแฟคเตอร์อัตราขยาย (Gain) ของฟังก์ชันถ่ายโอนแบบเปิดจากค่าเริ่มต้นเท่ากับศูนย์ไปจนถึงค่าอนันต์ เนื่องจากตัวแปร s ซึ่งใช้แทนตำแหน่งเป็นตัวแปรเชิงซ้อน ดังนั้นการวิเคราะห์ทางเดินของรากจึงเป็นวิธีที่ทำให้ทราบถึงตำแหน่งของโพล และตำแหน่งของซีโรของระบบแบบเปิดจากการเขียนตำแหน่งของโพลและซีโรของระบบลงในระนาบ s ด้วยการแทรกค่าแฟคเตอร์อัตราขยายแบบเปิด

ข้อดีของการวิเคราะห์ทางเดินของรากโดยวิธีเส้นทางเดินของรากที่เห็นได้ชัดเจน คือ การเขียนหาค่าตำแหน่งของโพลและซีโรของระบบควบคุมแบบปิดโดยตรงเป็นกระบวนการที่ยู่ยากมาก วิธีเส้นทางเดินของรากจะเขียนค่าของตำแหน่งโพลและซีโรของระบบเปิดได้ง่ายกว่า

ในการออกแบบระบบเชิงเส้น วิธีเส้นทางเดินของรากมีประโยชน์มาก เพราะการทราบตำแหน่งของโพลและซีโรของระบบควบคุมแบบเปิดสามารถอธิบายข้อกำหนดของสมรรถนะของการควบคุมระบบแบบปิดได้เป็นอย่างดี เป็นวิธีที่หาค่าโดยประมาณด้วยกราฟ การวิเคราะห์ผลจึงง่ายและรวดเร็ว

2.2.2 การเขียนทางเดินของราก

เพื่อให้เข้าใจการวิเคราะห์ระบบโดยอาศัยบล็อกไดอะแกรมที่เคยกล่าวมาแล้ว พิจารณาระบบควบคุมที่มีการป้อนกลับที่อธิบายเป็นบล็อกไดอะแกรมดังรูปที่ 2.7



รูปที่ 2.7 ระบบควบคุมที่มีการป้อนกลับ

จากรูปที่ 2.7 จะได้

$$\text{ฟังก์ชันถ่ายโอนแบบเปิด} = \frac{C(s)}{R(s)} = \frac{G(s)}{1+G(s)\cdot H(s)} \quad (2.4)$$

และ

$$\text{ฟังก์ชันถ่ายโอนแบบเปิด} = G(s)\cdot H(s) \quad (2.5)$$

$$\text{สมการคุณลักษณะ} = 1+G(s)\cdot H(s) = 0 \quad (2.6)$$

พิจารณาสมการคุณลักษณะโดยจัดรูปแบบใหม่ เพื่อแสดงให้เห็นค่าแพลคเตอร์ขยาย K ดังนี้

$$1+KG(s)\cdot H(s) = 0 \quad (2.7)$$

เมื่อ K คือ แพลคเตอร์อัตราขยาย ดังนั้น

$$KG(s)\cdot H(s) = -1 \quad (2.8)$$

จากสมการ หมายความว่า $|KG(s)\cdot H(s)| = 1$ และมีมุมเป็น 180°

หากพิจารณาเฉพาะขนาด จะได้

$$|K|\cdot|G(s)\cdot H(s)| = 1 \quad (2.9)$$

ดังนั้น

$$|G(s) \cdot H(s)| = \frac{1}{|K|} \quad (2.10)$$

พิจารณาเฉพาะมุม จะได้

$$\angle KG(s) \cdot H(s) = 180 + K \cdot 360 = 2(K+1) \cdot 180 \quad (2.11)$$

เมื่อ $0 < K < \infty$

เพื่อให้เข้าใจพื้นฐาน สมมติให้ฟังก์ชันถ่ายโอน $G(s)$ และ $H(s)$ มีค่าดังนี้

$$G(s) = \frac{K(s+Z_1)}{(s+P_1)(s+P_2)} \quad (2.12)$$

$$H(s) = \frac{1}{(s+P_3)} \quad (2.13)$$

ดังนั้น ฟังก์ชันถ่ายโอนแบบเปิดจึงได้เป็น

$$G(s) \cdot H(s) = \frac{K(s+Z_1)}{(s+P_1)(s+P_2)(s+P_3)} \quad (2.14)$$

พิจารณาโพลของฟังก์ชันถ่ายโอนแบบเปิดจะอยู่ที่ตำแหน่ง $s=-P_1$, $s=-P_2$ และ $s=-P_3$ เมื่อนำไปหาสมการคุณลักษณะ จะได้

$$1+G(s) \cdot H(s) = 0$$

$$1 + \frac{K(s+Z_1)}{(s+P_1)(s+P_2)(s+P_3)} = 0$$

$$(s+P_1)(s+P_2)(s+P_3) + K(s+Z_1) = 0 \quad (2.15)$$

ดังนั้น ฟังก์ชันถ่ายโอนแบบปิดจึงได้เป็น

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1+G(s)\cdot H(s)}$$

$$\frac{C(s)}{R(s)} = \frac{\frac{K(s+Z_1)}{(s+P_1)(s+P_2)}}{1 + \frac{K(s+Z_1)}{(s+P_1)(s+P_2)} \cdot \frac{1}{(s+P_3)}}$$

ดังนั้น

$$\frac{C(s)}{R(s)} = \frac{K(s+Z_1)(s+P_3)}{(s+P_1)(s+P_2)(s+P_3)+K(s+Z_1)} \quad (2.16)$$

กรณีค่า $K = 0$ พิจารณาเทียบกับสมการ 6.4 และ 6.5 จะได้ว่า

$$G(s)\cdot H(s) = \frac{K(s+Z_1)}{(s+P_1)(s+P_2)(s+P_3)} \Leftrightarrow (s+P_1)(s+P_2)(s+P_3)+K(s+Z_1) = 0$$

ตำแหน่งโพลของฟังก์ชันถ่ายโอนแบบเปิด = ตำแหน่งรากของสมการคุณลักษณะ

และหากพิจารณาสมการ 2.14 และ 2.16 จะได้ว่า

เมื่อ $K = 0$

$$G(s)\cdot H(s) = \frac{K(s+Z_1)}{(s+P_1)(s+P_2)(s+P_3)} \Leftrightarrow \frac{K(s+Z_1)(s+P_3)}{(s+P_1)(s+P_2)(s+P_3)+K(s+Z_1)}$$

จะได้ว่า

ตำแหน่งโพลของฟังก์ชันถ่ายโอนแบบเปิด = ตำแหน่งโพลของฟังก์ชันถ่ายโอนแบบปิด

กรณีที่ค่า $K = \infty$ จะมีผลต่อสมการคุณลักษณะ ดังนี้

เมื่อ $K = \infty$ พิจารณาสมการที่ 2.15

$$(s+P_1)(s+P_2)(s+P_3)+K(s+Z_1) = 0$$

จะพบว่าเทอมของ $(s+P_1)$, $(s+P_2)$ และ $(s+P_3)$ มีค่าน้อยมากจนสามารถตัดทิ้งได้ เมื่อเทียบกับเทอมของ $K(s+Z_1)$ ดังนั้นสมการที่ 2.15 เขียนใหม่ได้เป็น

$$K(s+Z_1) \cong 0$$

ซึ่งสรุปได้ว่า เมื่อค่า K มีค่ามากขึ้น รากของสมการคุณลักษณะจะย้ายไปอยู่ที่ตำแหน่งของซีโรของฟังก์ชันถ่ายโอนแบบเปิด ดังนี้
เมื่อ $K = \infty$

$$\frac{C(s)}{R(s)} = \frac{K(s+Z_1)(s+P_3)}{(s+P_1)(s+P_2)(s+P_3)+K(s+Z_1)} \Leftrightarrow G(s) \cdot H(s) = \frac{K(s+Z_1)}{(s+P_1)(s+P_2)(s+P_3)}$$

ตำแหน่งของโพลของฟังก์ชันถ่ายโอนแบบปิด = ตำแหน่งซีโรของฟังก์ชันถ่ายโอนแบบเปิด

จากสมการข้อสรุปกรอบบน พิจารณาสมการในรูปทั่วไปได้ ดังนี้

$$\frac{1}{|K|} = |G(s) \cdot H(s)| = \frac{|s+Z_1| \cdot |s+Z_2| \cdot \dots \cdot |s+Z_m|}{|s+P_1| \cdot |s+P_2| \cdot \dots \cdot |s+P_n|} \quad (2.17)$$

เมื่อ $s = -Z_1, -Z_2, \dots, -Z_m$ คือ ตำแหน่งซีโรของฟังก์ชันถ่ายโอนแบบเปิด
 $s = -P_1, -P_2, \dots, -P_n$ คือ ตำแหน่งโพลของฟังก์ชันถ่ายโอนแบบปิด

จากสมการที่ 2.16 ถ้า s เข้าใกล้ตำแหน่งโพลใดๆ ของฟังก์ชันถ่ายโอนแบบเปิด ขนาดของฟังก์ชันถ่ายโอนแบบเปิด $|G(s) \cdot H(s)|$ จะมีค่าไปสู่อนันต์ ซึ่งหมายความว่า

$$\frac{1}{|K|} \rightarrow \infty \text{ เมื่อ } s \text{ เข้าใกล้โพลใดๆ ของฟังก์ชันถ่ายโอนแบบเปิด } G(s) \cdot H(s)$$

ดังนั้น $|K| \rightarrow 0$ เมื่อ s เข้าใกล้โพลใดๆ ของฟังก์ชันถ่ายโอนแบบเปิด $G(s) \cdot H(s)$

นั่นคือ เมื่อค่าของตัวแปรเชิงซ้อน (Complex variable) s เข้าใกล้ตำแหน่งโพลใดๆ ของฟังก์ชันถ่ายโอนแบบเปิด $G(s) \cdot H(s)$ แล้ว ขนาดของแฟคเตอร์ $|K|$ จะมีค่าไปสู่ศูนย์ แต่ทราบมาจาก

“ตำแหน่งโพลของฟังก์ชันถ่ายโอนแบบเปิด = ตำแหน่งโพลของฟังก์ชันถ่ายโอนแบบปิด” ว่าเมื่อ $|K| = 0$ ตำแหน่งของโพลของฟังก์ชันถ่ายโอนแบบเปิดจะอยู่ที่ตำแหน่งโพลของฟังก์ชันถ่ายโอนแบบเปิด ดังนั้นจุดเริ่มต้นของทางเดินรากของสมการคุณลักษณะหรือโพลของฟังก์ชันถ่ายโอนแบบเปิด $\frac{C(s)}{R(s)}$ จะเริ่มต้นที่ $|K| = 0$ อยู่ที่ตำแหน่งของโพลของฟังก์ชันถ่ายโอนแบบเปิด $G(s) \cdot H(s)$

จากสมการ 2.17 ถ้า s เข้าใกล้ตำแหน่งซีโรใดๆ ของฟังก์ชันถ่ายโอนแบบเปิด $G(s) \cdot H(s)$ ค่า $|K|$ จะไปสู่อนันต์

$|K| \rightarrow \infty$ เมื่อ s เข้าใกล้ซีโรใดๆ ของฟังก์ชันถ่ายโอนแบบเปิด $G(s) \cdot H(s)$

จาก “ตำแหน่งของโพลของฟังก์ชันถ่ายโอนแบบปิด = ตำแหน่งซีโรของฟังก์ชันถ่ายโอนแบบเปิด” สรุปได้ว่า เมื่อ $|K| \rightarrow \infty$ ตำแหน่งโพลของฟังก์ชันถ่ายโอนแบบปิดจะเข้าใกล้ตำแหน่งซีโรของฟังก์ชันถ่ายโอนแบบเปิด

ดังนั้น จุดสุดท้ายของทางเดินรากของสมการคุณลักษณะหรือโพลของฟังก์ชันถ่ายโอนแบบปิด $\frac{C(s)}{R(s)}$ เมื่อ $|K| = \infty$ จะอยู่ที่ตำแหน่งซีโรของฟังก์ชันถ่ายโอนแบบเปิด $G(s) \cdot H(s)$

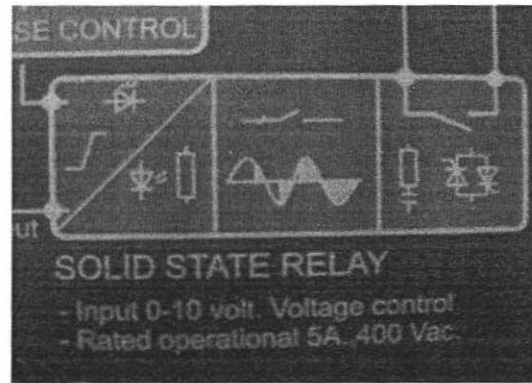
จากการวิเคราะห์ตำแหน่งของรากของสมการคุณลักษณะหรือโพลของฟังก์ชันถ่ายโอนแบบเปิด โดยการแปรค่าของ $|K|$ สรุปเป็นเส้นทางเดินของรากได้ ดังนี้

ทางเดินของราก คือ ทางเดินรากของสมการคุณลักษณะหรือโพลของฟังก์ชันถ่ายโอนแบบปิด บนระนาบ s เมื่อค่าของแอมพลิจูดขยาย K แปรค่าจากศูนย์ไปสู่อนันต์ โดยรากจะเริ่มต้นออกจากโพลของฟังก์ชันถ่ายโอนแบบเปิด $G(s) \cdot H(s)$ ไปสู่ตำแหน่งของซีโรของฟังก์ชันถ่ายโอนแบบเปิด $G(s) \cdot H(s)$

2.3 ชุดจำลองอุณหภูมิ

ในวิทยานิพนธ์นี้จะใช้ชุดจำลองอุณหภูมิ OMRON TEMPERATURE SIMULATION ในชุดจำลองอุณหภูมิ จะแบ่งออกเป็น 2 ช่วงคือด้านสัญญาณอินพุท และ เอาท์พุท ในการควบคุมจะแบ่งการควบคุมอุณหภูมิได้ 2 วิธีคือ

- 1) การควบคุมด้วย Manual Phase Control โดยทำการปรับเฟสของแรงดันเพื่อเพิ่มอุณหภูมิโดยการปรับด้วยวิธีการนี้จะไม่สามารถควบคุมอุณหภูมิได้
- 2) Linear Phase Control โดยจะใช้อุปกรณ์ควบคุมด้วย Solid State Relay โดยจะใช้แรงดันไฟตรง (Vdc) เป็นสัญญาณควบคุมเฟส



รูปที่ 2.8 SOLIT STATE RELAY

และในภาคของสัญญาณเอาต์พุตจะใช้ Thermocouple Transmitter เพื่อรับสัญญาณจากเทอร์โมคัปเปิลส์ โดยจะมีสัญญาณเอาต์พุต ที่ 1-10 Vdc

3) สายละเอียดภายในชุดจำลองอุณหภูมิ

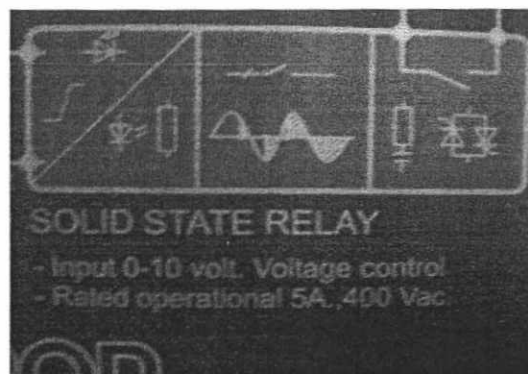
3.1) Solid State Relay , Control Input 1-10 Vdc , Rate Operational 5A., 400Vac

3.2) Heater 220 Vac , 50 Hz , 20 Watts

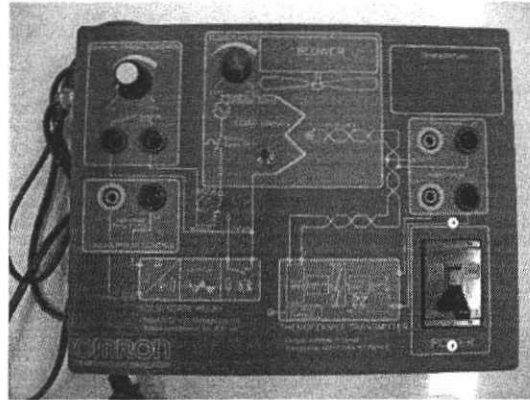
3.3) พัดลมระบายอากาศขนาด 12Vdc , 1.4W , ชุดแสดงอุณหภูมิ seven segment

3.4) Thermocouple Type K

3.5) ชุด Thermocouple Transmitter Output Voltage 0-10 Vdc Rangeable within Type K 100mv/C



รูปที่ 2.9 SOLIT STATE RELAY



รูปที่ 2.10 Omron Temperature Simulation

2.4 โปรแกรม MatLab [8],[9]

โดยโปรแกรม MATLAB ต้องการใช้ 2 หน้าต่าง คือ หน้าต่างคำสั่ง (Command Windows) และหน้าต่างกราฟ (Graphics Windows)

หน้าต่างคำสั่งเป็นหน้าต่างแรกที่พบ เมื่อเปิดโปรแกรม MatLAB การป้อนคำสั่งต่าง ๆ จะป้อนลงในหน้าต่างนี้ เพื่อทำการคำนวณและประมวลผลต่าง ๆ และเป็นหน้าต่างที่ใช้แสดงผลที่ได้จากการประมวลผลต่าง ๆ ยกเว้นรูปภาพซึ่งจะแยกไปแสดงผลในหน้าต่างกราฟ

ในการป้อนคำสั่งสำหรับการประมวลผลของโปรแกรม MatLAB สามารถป้อนได้ 2 วิธีด้วยกัน คือ หนึ่งเป็นการป้อนคำสั่ง หรือ ค่าตัวแปรต่าง ๆ บนหน้าต่างคำสั่งทีละคำสั่งตามขั้นตอนของการแก้ปัญหา หรือสองเป็นการกำหนดค่าตัวแปรต่าง ๆ และชุดคำสั่งต่าง ๆ ที่ใช้ทำการประมวลผลตามขั้นตอนของการคำนวณเพื่อแก้ปัญหาลงในไฟล์ ซึ่งจะต้องเก็บไว้ด้วยนามสกุล .m โดยที่จะเรียกชุดตัวแปรและชุดคำสั่งทั้งหมดที่เขียนในลักษณะนี้ว่า โปรแกรม M-file หรือเรียกว่า Script-file ซึ่งหมายถึงไฟล์ที่ถูกเก็บในรูปแบบของรหัส ASCII Code โดย M-file หรือ Script-file นี้ สามารถสร้างมาจากโปรแกรม Editor ใด ๆ หรือโปรแกรม Word Processor ใด ๆ ก็ได้ ในการเรียกใช้โปรแกรม M-file นี้ สามารถทำได้ง่าย ๆ โดยเพียงแต่พิมพ์ชื่อของ M-file โดยไม่ต้องตามด้วยนามสกุล (.m) ลงในหน้าต่างคำสั่ง โปรแกรม MatLAB ก็จะไปทำการอ่านคำสั่งใน M-file ชื่อนั้น ๆ และทำการประมวลผลไปที่ละบรรทัดจนจบโปรแกรม

สำหรับโปรแกรมคำสั่ง MatLAB นั้นโปรแกรม MatLAB มีคำสั่งซึ่งช่วยดูลักษณะการใช้งานและความสามารถในการทำงานของโปรแกรม MatLAB และคำสั่ง computer ซึ่งจะแสดงชนิดของเครื่อง Computer ที่เรากำลังใช้งานอยู่ในขณะนั้น นอกจากนี้ ยังมีคำสั่ง help ซึ่งจะอธิบายการใช้คำสั่งต่าง ๆ รวมถึงรูปแบบคำสั่งที่มีใช้ในโปรแกรม MatLAB

สัญลักษณ์หนึ่งตัวที่ใช้บ่อยในโปรแกรม MatLAB คือ % สัญลักษณ์ % เมื่อเป็นตัวแรกของคำสั่งหรือข้อความใด ๆ จะเป็น เครื่องหมายที่บอกให้เครื่องทราบว่าไม่ต้องทำการประมวลผลใด ๆ ต่อคำสั่ง หรือข้อความที่อยู่หลัง เครื่องหมายนี้ วัตถุประสงค์ของการใช้เครื่องหมายนี้ก็เพื่อเป็นการเขียนคำอธิบายโปรแกรมในคำสั่ง ของบรรทัดต่อ ๆ ไปที่เราได้เขียนไว้ ทั้งนี้เป็นการเตือนความจำของเราเองถึงขั้นตอนการทำงานของ โปรแกรม หรือผู้ที่จะนำโปรแกรมไปใช้ให้เข้าใจถึงตัวโปรแกรมที่เราเขียนนั้น

สำหรับโปรแกรม MatLAB คำสั่งต่าง ๆ ที่เกี่ยวข้องกับ Control System เป็นคำสั่งที่ใช้คำนวณระบบที่ถือว่าเป็น Linear System และ Time-invariant (LTI) เท่านั้น ส่วนระบบที่เป็น Nonlinear และ Time-variant นั้น จะต้องมีการเปลี่ยนระบบให้อยู่ในรูป Time-invariant ก่อนที่จะทำการประมวลผลโดยใช้โปรแกรมของ MatLAB

2.4.1 การวิเคราะห์เกี่ยวกับโพลีโนเมียล

ในการคำนวณค่าโพลีโนเมียล หากเราใช้ฟังก์ชัน polyval ในการคำนวณหาค่าโพลีโนเมียล สิ่งที่ใช้ในฟังก์ชันนี้ คือ สัมประสิทธิ์ของ polynomial ซึ่งสัมประสิทธิ์นี้เป็นเวกเตอร์ และเวกเตอร์จะบวกหรือลบกันได้ต้องมีขนาดเท่ากัน

2.4.1.1 การคูณของโพลีโนเมียลด้วยโพลีโนเมียล

การคูณนี้ไม่ต้องทำให้เวกเตอร์สัมประสิทธิ์เท่ากัน ดังการบวกและการลบฟังก์ชันที่ใช้คือ conv ซึ่ง conv(a,b) ฟังก์ชันนี้มี 2 arguments ทั้งตัวแรกและตัวที่สอง คือ สัมประสิทธิ์ของโพลีโนเมียลที่นำมาคูณกันตามลำดับ ผลลัพธ์ออกมาเป็นค่าสัมประสิทธิ์ของโพลีโนเมียลเช่นเดียวกัน

2.4.1.2 การหารโพลีโนเมียลด้วยโพลีโนเมียล

การหารนี้กำลังของโพลีโนเมียลที่เป็นเศษต้องมากกว่า ส่วนเสมอ ผลจะได้เป็น โพลีโนเมียลของจำนวนเต็ม และโพลีโนเมียลของเศษ เช่น ในจำนวนเต็มใด ๆ หาร 5 ด้วย 3 ผลที่ได้ คือ $1\frac{2}{3}$ หนึ่ง (1) คือ ค่าของจำนวนเต็มที่ได้จากการ หาร สอง (2) คือ ค่าของเศษที่ได้จากการหาร

ฟังก์ชันที่ใช้ในการหารนี้ คือ deconv ซึ่ง [q,r] = deconv (a,b) ฟังก์ชันนี้มี 2 arguments: ตัวที่หนึ่งคือ เวกเตอร์ของสัมประสิทธิ์ของโพลีโนเมียลของตัวที่ถูกหาร ตัวที่สองคือ เวกเตอร์สัมประสิทธิ์ของโพลีโนเมียลของตัวที่จะหาร ผลลัพธ์จะถูกส่งไปเก็บไว้ 2 เวกเตอร์ (ที่สมมติคือ q และ r) โดยที่เวกเตอร์ผลลัพธ์แรกจะมีขนาดยาวเท่ากับกำลังของโพลีโนเมียลที่เป็นเศษลบด้วยกำลังของโพลีโนเมียล ซึ่งผลลัพธ์ คือ สัมประสิทธิ์ของโพลีโนเมียลของจำนวนเต็มที่เกิดจากการหาร และเวกเตอร์ผลลัพธ์ที่สองคือ สัมประสิทธิ์ของโพลีโนเมียลของเศษที่เหลือจากการหารแต่ขนาดที่ได้จะยาวเท่ากับกำลังของโพลีโนเมียลที่เป็นเศษบวกด้วยกำลังของโพลีโนเมียลที่เป็นส่วน บวกด้วย 1 โดยค่าสัมประสิทธิ์ที่กำลังมากกว่าหรือเท่ากับโพลีโนเมียลที่เป็นตัวหารจะเป็นศูนย์

2.4.2 การใช้ MatLAB วิเคราะห์เกี่ยวกับระบบควบคุม

2.4.2.1 Transfer Function กับ zero-pole-gain

สำหรับโปรแกรม MatLAB คำสั่งต่างๆ ที่เกี่ยวข้องกับ Control System เป็นคำสั่งที่ใช้คำนวณระบบที่ถือว่าเป็น Linear System และ Time-invariant (LTI) เท่านั้น ส่วน

ระบบที่เป็น Nonlinear และ Time-variant นั้น จะต้องมีการเปลี่ยนระบบให้อยู่ในรูป Time-invariant ก่อนที่จะทำการประมวลผลโดยใช้โปรแกรมของ MatLAB ซึ่งระบบที่เราพิจารณา คือ

$$H(s) = \frac{x(s)}{u(s)} \quad (2.18)$$

จากค่า Transfer Function นี้ เราจะนำไปใช้ในการคำนวณต่างๆ ของ Control System โดยขั้นแรกเมื่อต้องการทราบของระบบเราจำเป็นต้องแยกสมการของ Transfer Function ในส่วนของ Numerator และ Denominator ออก เป็นสมการ zero-pole-gain function ซึ่งจะทำให้ได้สมการอยู่ในรูป

$$H(s) = \frac{k(s-z_1)(s-z_2)\cdots(s-z_n)}{(s-p_1)(s-p_2)\cdots(s-p_n)} \quad (2.19)$$

คำสั่งของโปรแกรม MATLAB ที่ใช้ในการเปลี่ยนสมการ Transfer Function ให้เป็น zero-pole-gain function คือ [z, p, k] = tf2zp(num, den) ในทางตรงกันข้าม ถ้าเราทราบ poles และ zero เราสามารถหา Transfer Function ได้จากคำสั่ง [NUM, DEN] = zp2tf(z, p, k)

2.4.2.2 การหา system response จาก transfer function

สำหรับ MatLAB มีฟังก์ชันพื้นฐานในการหา system response ของแต่ละชนิดของอินพุตแต่อินพุตที่เราสนใจคือ step function

[y,x,t] = step (num,den) % ใช้สำหรับ H(s) ที่เป็นสมการโพลีโนเมียลโดย MATLAB จะเลือกเวลา t

[y,x,t] = step (num,den,t) % ผู้ใช้สามารถเลือกช่วงเวลา t ได้เอง

[y,x,t] = step (a,b,c,d) % สำหรับระบบที่อยู่ในรูป state-model และ MATLAB จะเลือกช่วงเวลา t

[y,x,t] = step (a,b,c,d,iu) % สำหรับ multivariable system ซึ่งผู้ใช้สามารถหา Output ที่ขึ้นกับ Input ลำดับที่ iu

[y,x,t] = step (a,b,c,d,iu,t) % ผู้ใช้เลือกเวลา t เอง

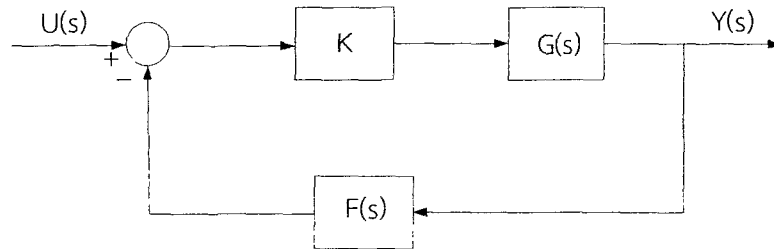
สำหรับ Input ที่เป็น impulse function ใน MatLAB ก็จะมีวิธีการเช่นเดียวกับ step นั่นคือ [y,x,t] = impulse (เหมือนของ step)

นอกจากนี้ ถ้าผู้ใช้ต้องการอินพุตที่ต่างจาก step function และ impulse function ก็สามารถเลือกใช้ฟังก์ชัน LSIM ซึ่งมีรูปแบบการใช้ดังนี้คือ [y,x] = lsim(num,den,u,t) ใช้สำหรับระบบที่อยู่ในรูป polynomial eq. และ matrix u จะมีจำนวนคอลัมน์เท่ากับจำนวนลำดับของอินพุต แต่ความยาวของคอลัมน์ของ u ต้องมีขนาดเดียวกันกับของเวกเตอร์ t ซึ่งค่าในแต่ละคอลัมน์ของ u คือค่าของอินพุต ณ เวลาที่อยู่ในเวกเตอร์ t

2.4.2.3 การหา Root-Locus Analysis

Root-locus-method คือ วิธีการหา closed-loop poles จาก zero และ pole ของ loop transfer function ตามค่า gain k ที่เปลี่ยนแปลงของ loop transfer function ซึ่ง Function ที่สามารถใช้ได้ใน MatLAB คือ rlocus

พิจารณาระบบโดยทั่วไป ของ feedback system



รูปที่ 2.11 วงจร Closed loop transfer function

Closed loop transfer function คือ

$$H(s) = \frac{Y(s)}{R(s)} = \frac{KG(s)}{1+KG(s)F(s)} \quad (2.20)$$

ถ้าเขียน Characteristic equation $1+KG(s)F(s)$ ให้อยู่ในรูป

$$1+KG(s)F(s) = 1+K \frac{\text{num}(s)}{\text{den}(s)} \quad (2.21)$$

เมื่อทั้ง $\text{num}(s)$ และ $\text{den}(s)$ ต่างก็เป็น polynomial function ในเทอมของ s เราสามารถใช้ function rlocus ใน MatLAB ช่วยในการหาค่าตำแหน่งของ closed-loop-poles ของระบบได้ ณ gain K และ rlocus จะ plot locus ของ $H(s)$ ให้อัตโนมัติเช่นกัน รูปแบบการใช้ rlocus ใน MatLAB ให้เลือก ดังนี้

`[r,k] = rlocus(num,den) % สำหรับ $G(s)F(s)$ ที่อยู่ในรูป polynomial function โดยMATLAB จะเลือก gain k ให้`

`[r,k] = rlocus(num,den,K) % ใช้กรณีเลือกค่า gain K เอง`

2.5 โปรแกรม LabVIEW

โปรแกรม LabVIEW (Laboratory virtual instrument engineering workbench) หมายความว่า LabVIEW เป็นโปรแกรมที่สร้างเครื่องวัดเสมือนจริงในห้องปฏิบัติการทางวิศวกรรม โดยมีจุดประสงค์หลักของการทำงาน คือ จัดการด้านการวัดและเครื่องมือวัดได้อย่างมีประสิทธิภาพ

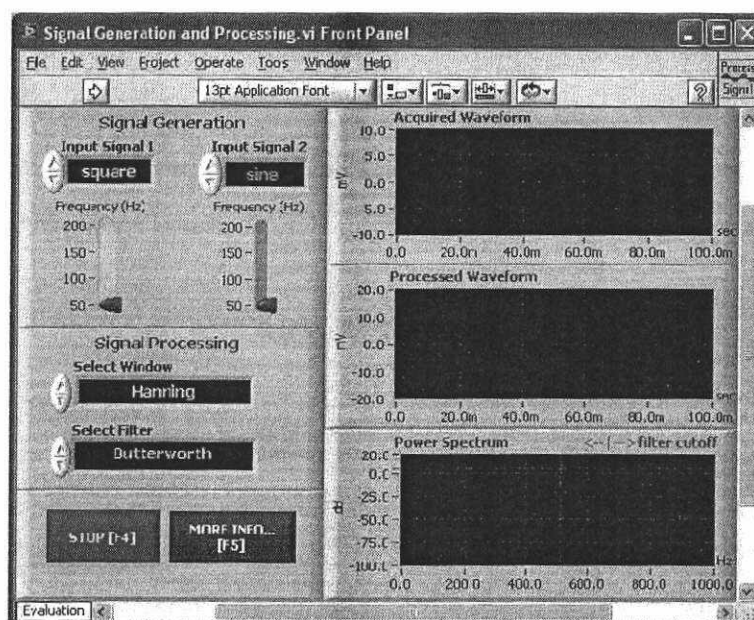
โปรแกรมจะประกอบด้วยฟังก์ชันที่ช่วยในการวัดมากมายซึ่งโปรแกรมนี้จะมีประโยชน์อย่างสูงเมื่อใช้ร่วมกับเครื่องมือวัดทางวิศวกรรมต่างๆ

โปรแกรม LabVIEW แตกต่างจากโปรแกรมอื่นอย่างเห็นได้ชัดที่สุดคือเป็นโปรแกรมประเภท (Graphic user interface : GUI) โดยสมบูรณ์ โดยมีแนวคิดแบบ Dataflow ต่างกันกับแนวคิดของโปรแกรมประเภท Text base programming ซึ่งทำงานจากบรรทัดบนไปสู่บรรทัดล่างสามารถเขียนเป็นสมการต่างๆ ที่ซับซ้อนได้เหมือน Text base programming เช่น ภาษาซี โดยใช้ฟังก์ชัน Formula node การทำงานจะเน้นติดต่อกับเครื่องมือหรือ อุปกรณ์ Data acquisition (DAQ) ต่อเชื่อมเข้ากับคอมพิวเตอร์เพื่อเก็บข้อมูลและส่งต่อไปเก็บไว้ที่หน่วยความจำของคอมพิวเตอร์ หรือ เก็บอยู่ในรูปของไฟล์ข้อมูลเพื่อนำไปประมวลผลอีกครั้ง นอกจากนี้ยังมีอุปกรณ์บางส่วนสามารถส่งผ่านข้อมูลไปยังคอมพิวเตอร์ โดยผ่านทาง Parallel port, Serial port หรือ USB port เป็นต้น การเก็บข้อมูลจากอุปกรณ์วัดด้วยคอมพิวเตอร์สามารถเรียกอีกอย่างว่า PC-Based DAQ System [12]

ในงานวิจัยฉบับนี้ได้อาศัยความสามารถของโปรแกรม LabVIEW ในการเก็บค่าและแสดงผล โดยเมื่อรับค่าอุณหภูมิจาก IC LM 35DZ Analog Transformer ผ่านการ์ด National Instrument PCI 6014 จะการแปลงสัญญาณ A/D เข้าฟังก์ชัน DAQ Assistant ของโปรแกรม LabVIEW โดยรับค่าจาก IC LM 35DZ ทั้งสี่ตัวเข้ามาในโปรแกรม ต่อจากนั้น ฟังก์ชัน Split signals จะทำการแยกสัญญาณที่ส่งมาจาก ฟังก์ชัน DAQ Assistant เพื่อแสดงผลของแต่ละค่าในรูปแบบ Waveform graph and chart และจะถูกเรียกข้อมูลโดยฟังก์ชัน Like to text file และเก็บข้อมูลใน Data base ที่เราสร้างไว้

LabVIEW เป็นซอฟต์แวร์ที่พัฒนาขึ้นโดย บริษัท เนชั่นแนลอินสตรูเมนต์ จำกัด (National Instruments) หรือ NI ซึ่งเป็นผู้พัฒนาอุปกรณ์และซอฟต์แวร์ที่ใช้งานวัดและควบคุมอัตโนมัติชั้นนำ บริษัทหนึ่ง LabVIEW จัดเป็นซอฟต์แวร์ประเภท Graphical Programming Language คล้ายกับ C++ แต่ต่างกันตรงที่ LabVIEW เป็นการสร้างโปรแกรมโดยใช้ภาษารูปภาพ คือจะใช้บล็อกฟังก์ชันซึ่งแทนด้วยรูปไอคอนแทนการเขียนโปรแกรม และใช้เส้นเชื่อมระหว่างบล็อกฟังก์ชันแทนการไหลของข้อมูลระหว่างโปรแกรมนั้นๆ จะมีคล้ายกันกับการเขียนโฟลชาร์ตหรือบล็อกไดอะแกรมของโปรแกรมด้วยเหตุนี้เองการพัฒนาโปรแกรมด้วย LabVIEW จึงช่วยลดเวลาลงอย่างมากเมื่อเทียบกับการเขียนโปรแกรมด้วยภาษาที่ใช้อักษรแบบเดิมประมาณ 4 เท่า ถึง 10 เท่า สิ่งที่ทำให้ LabVIEW ต่างจากซอฟต์แวร์อื่นทั่วไปอีกอย่างหนึ่ง คือ ความสามารถในการพัฒนางานทางด้านการวัดและควบคุมแบบอัตโนมัติ ซึ่งเป็นเป้าหมายสำคัญของ LabVIEW โดยมีเครื่องมือที่สนับสนุนการใช้งานทางด้านนี้ไว้อย่างมากมาย ให้ผลลัพธ์ออกมาในรูปแบบของเครื่องมือเสมือนจริง นอกจากนี้ในเวอร์ชันตั้งแต่ LabVIEW 6i ขึ้นไป ยังเพิ่มความสามารถทางการสนับสนุนการทำงานเครือข่ายและเทคโนโลยีอินเทอร์เน็ตให้กับโปรแกรมที่พัฒนาด้วย LabVIEW ทำให้โปรแกรมเหล่านี้มีการใช้งานที่สะดวกมากยิ่งขึ้น

LabVIEW จะมี Front Panel ซึ่งเปรียบเสมือนกับสิ่งที่ผู้ใช้งานจะเห็นและควบคุม Front Panel ประกอบด้วยส่วนที่รับคำสั่ง (Control) เช่น การป้อนข้อมูล ปุ่ม สวิตช์ และส่วนแสดงผล (Indicator) เช่น การแสดงผลด้วยตัวเลข มิเตอร์ บาร์สเกล และกราฟ ส่วนต่างๆ เหล่านี้สามารถเลือกได้จาก Controls Palette แล้วนำมาวางประกอบกัน ผู้ใช้งานสามารถควบคุมการทำงานของโปรแกรมที่สร้างขึ้นผ่าน Front Panel ด้วยการกดปุ่ม เลื่อนแถบสไลด์หรือป้อนข้อมูลทางคีย์บอร์ด พร้อมทั้งแสดงผลข้อมูลได้อีกด้วย



รูปที่ 2.12 Front Panel

โปรแกรมที่พัฒนาขึ้นโดย LabVIEW จะเรียกว่า VI (Virtual Instrument) ซึ่งสามารถสร้างขึ้นเองได้อย่างรวดเร็ว เพราะมีส่วนประกอบต่างๆ ที่ใช้ออกแบบ Front Panel ให้เลือกมากมาย

อุปกรณ์ดาด้า แอคควิซิชัน (Data Acquisition)

มีลักษณะเป็นการดอินเทอร์เฟซหรือแบบที่เชื่อมต่อด้วยพอร์ต USB ทำหน้าที่เชื่อมต่อกับคอมพิวเตอร์เพื่ออ่านค่าจากการวัดแล้วส่งข้อมูลให้กับคอมพิวเตอร์หรือในทางกลับกันจะทำหน้าที่รับข้อมูลที่ได้จากคอมพิวเตอร์ แล้วสร้างเป็นสัญญาณควบคุมเพื่อใช้ควบคุมอุปกรณ์ภายนอก

โครงสร้างของ DAQ อาจแตกต่างกันไปบ้างตามลักษณะของงานที่จะใช้ ซึ่งรวมไปถึงการทำงานของ DAQ ก็แตกต่างกันตามชนิดสัญญาณและฟังก์ชันการทำงาน เช่น จำนวนช่องสัญญาณ อินพุต เอาต์พุต อัตราการสุ่ม (Sampling Rate) ระดับการแยกแยะสัญญาณ (Resolution) ความแม่นยำและย่านของอินพุตที่รับได้ (Range) โดยใช้อุปกรณ์ของบริษัท National Instruments รุ่น PCI-6014 ขนาด 2 Output 24 bits เป็นต้น

ดังนั้น ดาต้า แอควิซิชัน เป็นอุปกรณ์ที่ใช้ในการเก็บข้อมูลในรูปแบบของข้อมูลดิจิทัล
ซึ่งนิยมเรียกทับศัพท์ว่า DAQ

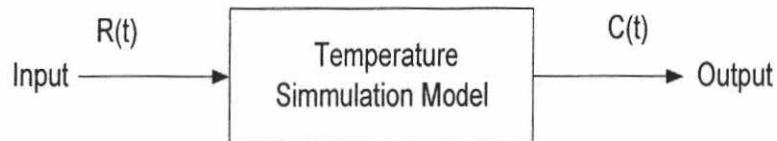
บทที่ 3

การออกแบบตัวควบคุมและโปรแกรม M-file

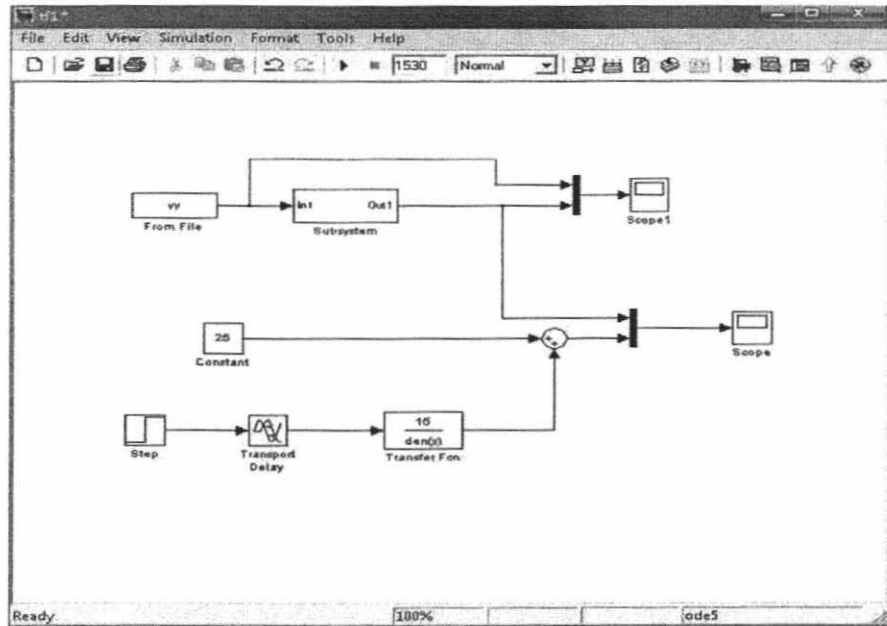
บทนี้กล่าวถึงการออกแบบตัวควบคุมชุดจำลองอุณหภูมิรวมถึงการหาสมการคุณลักษณะและค่าพารามิเตอร์ต่างๆที่จะใช้เป็นตัวแปรในการควบคุมโดยจะแบ่งขั้นตอนการหาค่าพารามิเตอร์ พีโอดี โดยใช้วิธีทางเดินรากในการการเขียนโปรแกรม M-file และใช้โปรแกรม Mat Lab เป็นโปรแกรมจำลองค่าพารามิเตอร์โดยเพื่อตรวจสอบค่าพารามิเตอร์ของโปรแกรม โดยใช้โปรแกรม LabVIEW ควบคุมกระบวนการจริง โดยใช้ค่าพารามิเตอร์ที่ได้มา

3.1 การหาค่าสมการคุณลักษณะชุดจำลองอุณหภูมิ [15]

จากชุดจำลองอุณหภูมิในรูปที่ 2.10 นำมาหาสมการคุณลักษณะโดยการใช้วิธีการ Open loop ดังรูปที่ 3.1 โดยจะใช้วิธีกำหนดแรงดันให้คงที่ค่าหนึ่งจ่ายให้ชุดจำลองอุณหภูมิ โดยรอจนอุณหภูมิเข้าสู่สภาวะคงตัว (steady state) ที่ค่าหนึ่งและนำค่าที่ได้ไปแสดงผลโดยใช้โปรแกรม MatLab โดยสร้างแบบจำลองเพื่อเปรียบเทียบผลดังรูปที่ 3.2

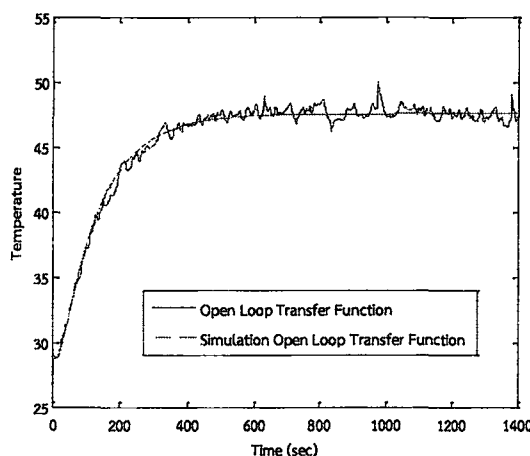


รูปที่ 3.1 แสดงวิธีการ Open loop เพื่อหาสมการคุณลักษณะ



รูปที่ 3.2 แสดงการจำลองหาค่าสมการคุณลักษณะของระบบ

จากแบบจำลองบน Simulink Mat lab ในรูปที่ 3.2 จะใช้ฟังก์ชัน Unit Step หาค่าสมการคุณลักษณะโดยนำผล Open Loop ในรูปที่ 3.1 มาเป็นตัวเปรียบเทียบ โดยจะทำการปรับสมการที่บล็อก Transfer Fun เพื่อให้กราฟข้อมูลมีค่าใกล้เคียงกับกราฟ Open Loop โดยปรับค่าที่สมการทั้งในเทอมของเศษและเทอมของส่วนเพื่อปรับกราฟจำลองคุณลักษณะให้มีความใกล้เคียงกันที่สุดดังรูปที่ 3.3



รูปที่ 3.3 ผลการปรับหาค่าสมการคุณลักษณะของระบบ

จากรูปที่ 3.3 ผลจากการปรับสมการเพื่อเปรียบเทียบกับผลของ Open Loop จะเห็นว่ากราฟสมการทั้งมีค่าใกล้เคียงกัน ทำให้สามารถประมาณสมการคุณลักษณะ (Characteristic equation) ของระบบจากสมการที่ปรับในโปรแกรมจำลอง Simulink จะได้ค่าดังสมการที่ 3.1

$$G(S) = \frac{14.5}{1700s^2 + 135s + 1} \quad (3.1)$$

3.2 หลักการออกแบบโปรแกรมและกระบวนการทำงาน M-file

ในการออกแบบโปรแกรมนี้จะใช้ทฤษฎีทางเดินราก (Root Locus Theory) โดยใช้สมการในบทที่สองเป็นพื้นฐานในการเริ่มเขียนโปรแกรมโดยเริ่มกระบวนการตามโฟลว์ชาร์ต (Flow chart diagram) ดังกระบวนการดังรูปที่ 3.4 โดยจะมีโปรแกรมดังภาคผนวก ก

จากโปรแกรม M-file ในภาคผนวก ก ต้องทำการใส่ค่าเริ่มต้นของสมการโดยเป็นส่วนที่สำคัญของโปรแกรม จะแบ่งออกเป็นสามค่าคือ สมการคุณลักษณะ เปอร์เซ็นโอเวอร์ชูต และ ค่าเซตตั้งทาร์ม เป็นการกำหนดค่าเริ่มต้นของโปรแกรม ในการออกแบบโปรแกรมนี้จะแบ่งออกเป็นสองส่วนคือ

1) ออกแบบตัวควบคุมแบบ พีโอ เพื่อชดเชยผลตอบสนองช่วงเวลาที่สภาวะคงตัวจากสมการอนุพันธ์การควบคุมแบบ พีโอ แบบอนาล็อก คือ

$$MV(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt \quad (3.2)$$

เมื่ออินทิเกรตของทางสัญญาณดิสครีตมีค่าเป็น $\frac{1}{1-Z^{-1}}$ ดังนั้นตัวควบคุมแบบ พีไอ จะมีแบบจำลอง ดังสมการที่ 3.3

$$M(z) = \left[K_p + K_i \frac{1}{1-Z^{-1}} \right] E(z) \quad (3.3)$$

ทำการจัดรูปแบบสมการที่ 3.3 ใหม่จะได้

$$G_{ctrl}(z) = K_p \frac{K_i z}{z-1} = \frac{(K_p + K_i)z - K_p}{z-1} \quad (3.4)$$

โดยที่ตำแหน่งของโพลที่ปรากฏของตัวควบคุมแบบ พีไอ ในระนาบ z จะปรากฏที่ $+1$ และมีตำแหน่งซีโรปรากฏที่ $z = \frac{K_p K_i}{K_i + K_p}$ ที่อยู่ในช่วงของ 0 และ 1 หรือในวงกลมหนึ่งหน่วย โดยจะสามารถปรับตำแหน่งซีโรของตัวควบคุมแบบ พีไอ เพื่อให้เฟสที่เกิดขึ้นสำหรับตัวควบคุมแบบ พีไอ มีค่าเป็นลบหรือเป็นเฟสที่ตามหลังดังสมการที่ 3.5

$$\alpha_{pi} - \theta_{pi} = \theta_{req} \quad (3.5)$$

โดยที่ θ_{pi} เป็นมุมของตัวควบคุมแบบ พีไอ และมักจะมีค่าที่มากกว่า α_{pi} โดยจากผลรวมภายในของสามเหลี่ยมจะเท่ากับมุม 180 องศาจะได้ดังสมการที่ 3.6

$$\begin{aligned} \alpha_{pi} + (180^\circ - \theta_{pi}) + \phi &= 180 \\ \phi &= \alpha_{pi} - \theta_{pi} = \theta_{req} \end{aligned} \quad (3.6)$$

โดยแสดงรูปของทรานสเฟอร์ฟังก์ชัน พีไอแบบดิสครีตจะมีค่าดังสมการที่ 3.7

$$G_{ctrl}(z) = (K_p + K_i) \frac{z - \left(\frac{K_p}{K_p + K_i} \right)}{z-1} = K \left(\frac{z - \phi}{z-1} \right) \quad (3.7)$$

ดังนั้นการกำหนดผลตอบสนองที่ต้องการโดยกำหนดตำแหน่งโพลวงปิดในระนาบ z มีค่าเป็น $\sigma + j\beta$ จะสามารถหามุมที่เกิดจากเพิ่มตัวควบคุมแบบดีสครีตได้ ดังนี้

$$\theta_{req} = \tan^{-1}\left(\frac{\beta}{\sigma-\alpha}\right) - \tan^{-1}\left(\frac{\beta}{\sigma-1}\right) \quad (3.8)$$

จากสมการที่กล่าวข้างต้นจะสามารถเขียนโปรแกรมควบคุมโดยใช้ M-file คำนวณหามุมและตำแหน่งของโพลแบบ พีไอ แล้วนำค่าที่ได้มาหาค่าการควบคุมแบบพีไอดี และในส่วนนี้จะอยู่ในส่วนแรกของกระบวนการคำนวณ คือ Calculate PI case 1 ตามขั้นตอนการทำงานดังแสดงในรูปที่ 3.4

2) ออกแบบตัวควบคุมแบบ พีดี เพื่อชดเชยผลตอบสนองช่วงเวลาที่สภาวะชั่วขณะ และจากสมการทรานสเฟอร์ฟังก์ชัน ตัวควบคุมแบบดีสครีตจะมีแบบจำลองของการดิฟเฟอเรนทิเอชัน คือ $1-z^{-1}$ ดังนั้นตัวควบคุมแบบ พีดี จะมีรูปแบบดังสมการที่ 3.9

$$M(z) = [K_p + K_d(1-z^{-1})]F(z) \quad (3.9)$$

โดยที่สมการทรานสเฟอร์ฟังก์ชันของตัวควบคุม พีดี จะได้ดังสมการที่ 3.10

$$G_{pd}(z) = K_p + \frac{K_d(z-1)}{z} = (K_p + K_d) \left(\frac{z-\alpha}{z} \right) \quad (3.10)$$

จากทรานสเฟอร์ฟังก์ชันที่ปรากฏ จะเห็นได้ว่าตัวควบคุม พีดี จะมีตำแหน่งโพลอยู่ที่ 0 ในระนาบ z และมีตำแหน่งซีโรที่ปรากฏในตำแหน่ง $\frac{K_d}{K_d + K_p}$ บนแกนจำนวนจริงระหว่าง 0 กับ 1

โดยผลของการเพิ่มตัวควบคุม พีดี แบบดีสครีตเป็นการเพิ่มตำแหน่งโพล และ ซีโรเข้าไปในระบบวงปิดแต่ไม่ใช่การเปลี่ยนอันดับของระบบ โดยการออกแบบตัวควบคุมนี้ จะสามารถทำได้โดยการปรับตำแหน่งซีโรของตัวควบคุมเมื่อ α_{pd} และ θ_{pd} เป็นมุมของซีโรกับโพลของตัวควบคุม พีดี ตามลำดับ โดยมุมที่ปรากฏขึ้นของตัวควบคุมนี้จะเป็นดังสมการ

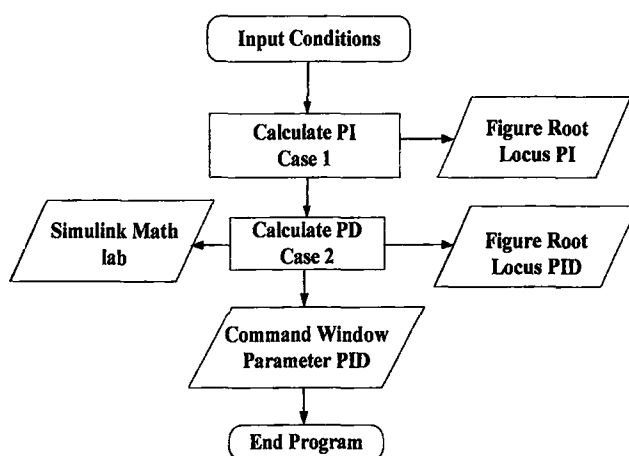
$$\alpha_{pd} - \theta_{pd} = \theta_{req} \quad (3.11)$$

การปรับเปลี่ยนนี้จะทำภายใต้เงื่อนไขของมุมจะมีค่าเป็นบวกเสมอ ดังนั้น ในการกำหนดผลตอบสนองที่ต้องการคือการ กำหนดตำแหน่ง โพลในระนาบ z มีค่าเป็น $\sigma + j\beta$ โดยมุมของระบบที่เกิดจากการเพิ่มตัวควบคุม พีดี แบบดีสครีตจะได้ดังสมการที่ 3.12

$$\theta_{req} = \tan^{-1}\left(\frac{\beta}{\sigma-\alpha}\right) - \tan^{-1}\left(\frac{\beta}{\sigma}\right) \quad (3.12)$$

จากสมการที่กล่าวข้างต้นจะสามารถเขียนโปรแกรมควบคุมโดยใช้ M-file คำนวณหามุมและตำแหน่งของโพลแบบ พีไอ แล้วนำค่าที่ได้มาหาค่าการควบคุมแบบพีไอดี และในส่วนนี้จะอยู่ในส่วนแรกของกระบวนการคำนวณ คือ Calculate PD case 2 ตามขั้นตอนการทำงานดังแสดงในรูปที่ 3.4

หลังจากที่ผ่านกระบวนการทั้งสอง จะสามารถหาค่าพารามิเตอร์ควบคุมแบบ พีไอดี ได้โดยจะตรวจสอบค่าโดยใช้ค่าตัวแปรที่ Command window หรือตรวจสอบโพลที่ทางเดินรากว่าเป็นไปตามค่ากำหนดเริ่มต้นหรือไม่ ดังแสดงในขั้นตอนดังรูปที่ 3.4



รูปที่ 3.4 แสดงกระบวนการทำงานของโปรแกรม M-file

3.3 การออกแบบการจำลองการควบคุมแบบ พีไอดี (PID Control) [10], [8]

โดยผลการจำลองในงานวิจัยนี้จะใช้การเขียนโปรแกรมจำลองการทำงานโดยใช้โปรแกรม Simulink Mat Lab โดยจะจำลองกระบวนการควบคุมแบบ พีไอดี (PID) ดังภาคผนวก ข ในกระบวนการนี้จะใช้ค่าพารามิเตอร์ที่ได้จากโปรแกรม M-file ในภาคผนวก ก เพื่อใช้ตรวจสอบค่าเริ่มต้นของโปรแกรมว่าเป็นไปตามเงื่อนไขของโปรแกรมหรือไม่ โดยค่าพารามิเตอร์จะถูกส่งไปยังโปรแกรม Simulink เพื่อจำลองกระบวนการและใช้สังเกตพฤติกรรมของระบบดังกล่าว

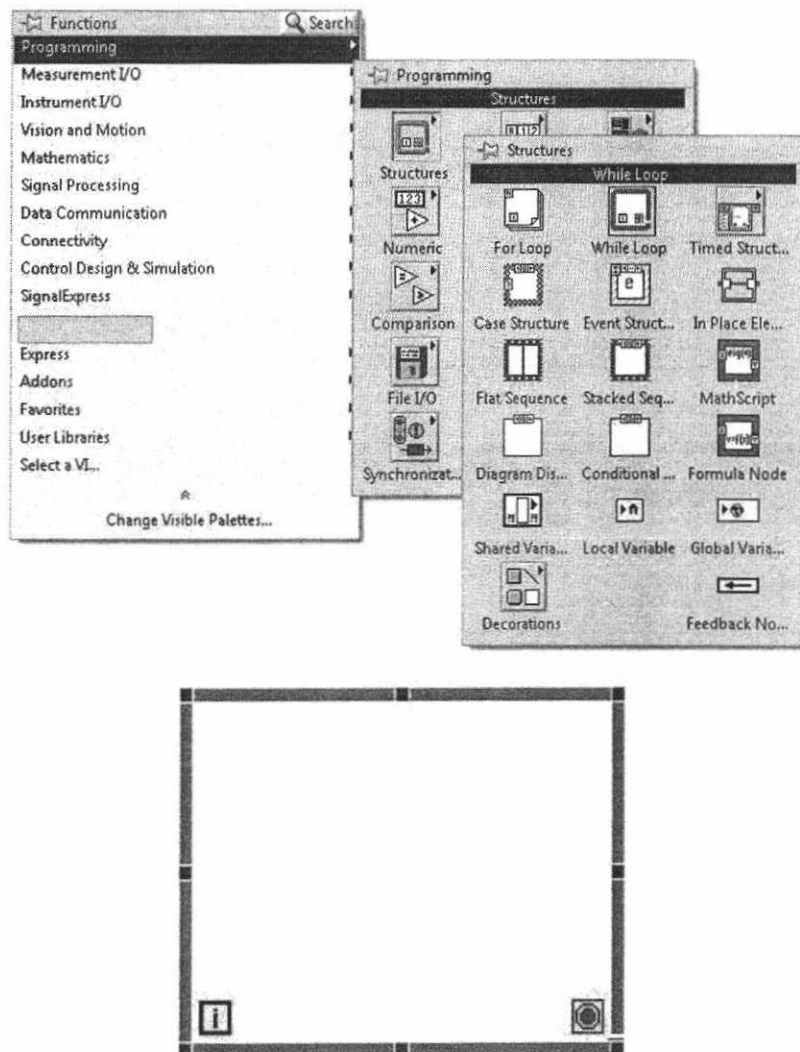
ในการจำลองนี้จะใช้ฟังก์ชัน ยูนิตส텝 เป็นสัญญาณจำลองกระบวนการ โดยจะใช้ร่วมกับฟังก์ชันดีสครีตทราสเฟอร์ฟังก์ชัน ฟังก์ชันซีโรออเดอร์โฮล และจะมีสมการคุณลักษณะของระบบเพื่อจำลองกระบวนการ โดยค่าที่นำมาจำลองจะได้มาจากโปรแกรม M-file จากภาคผนวก ก โดยผลการจำลองจะสังเกตได้จาก ผลตอบสนองเอาต์พุทของระบบ

3.4 การออกแบบการควบคุมระบบจริงโดยใช้โปรแกรม LabVIEW [17]

ในงานวิจัยนี้จะใช้การเขียนโปรแกรมควบคุมกระบวนการจริงโดยใช้โปรแกรม LabVIEW ทดสอบค่าพารามิเตอร์ที่ได้จากโปรแกรม M-file จากโปรแกรมในภาคผนวก ก โดยจะใช้โปรแกรมในภาคผนวก ค การควบคุมกระบวนการจริงพร้อมทั้งแสดงกราฟฟิการควบคุมและสามารถปรับเปลี่ยนค่าพารามิเตอร์ในการควบคุมกระบวนการจริง และสามารถกำหนดค่าของอุณหภูมิ และเก็บข้อมูลผ่านโปรแกรม Excel โดยจะเก็บข้อมูลผ่านการ์ด PCI 6014

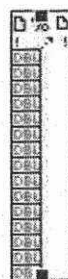
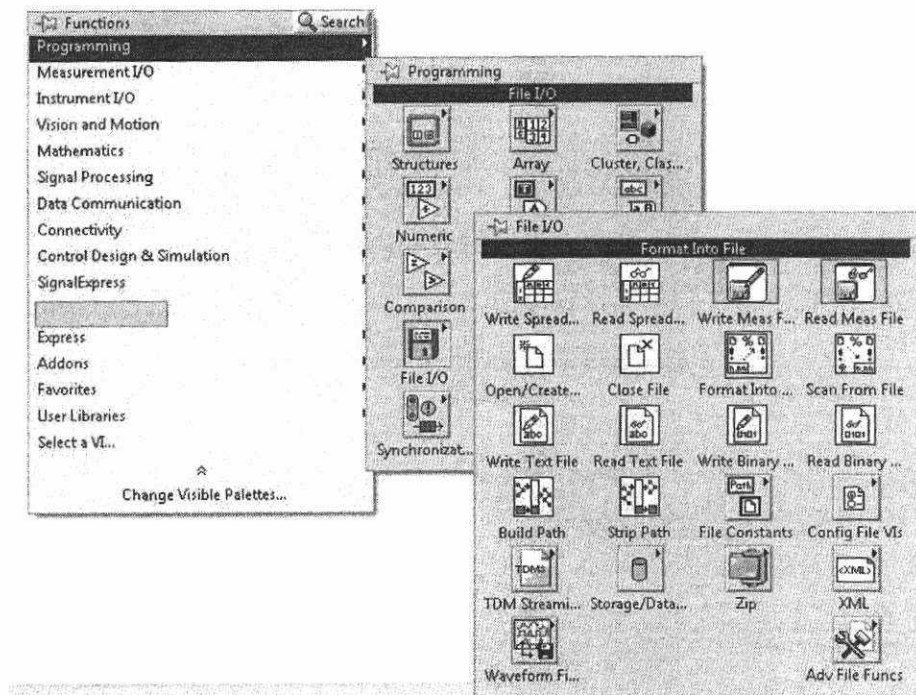
3.4.1 ฟังก์ชันการออกแบบโปรแกรม LabView

- 1) ฟังก์ชัน While Loop คือ ฟังก์ชันที่กำหนดให้ Code หรือ โปรแกรมในกระบวนการทำงานแบบวนซ้ำจนเป็นไปตามเงื่อนไขที่กำหนดไว้



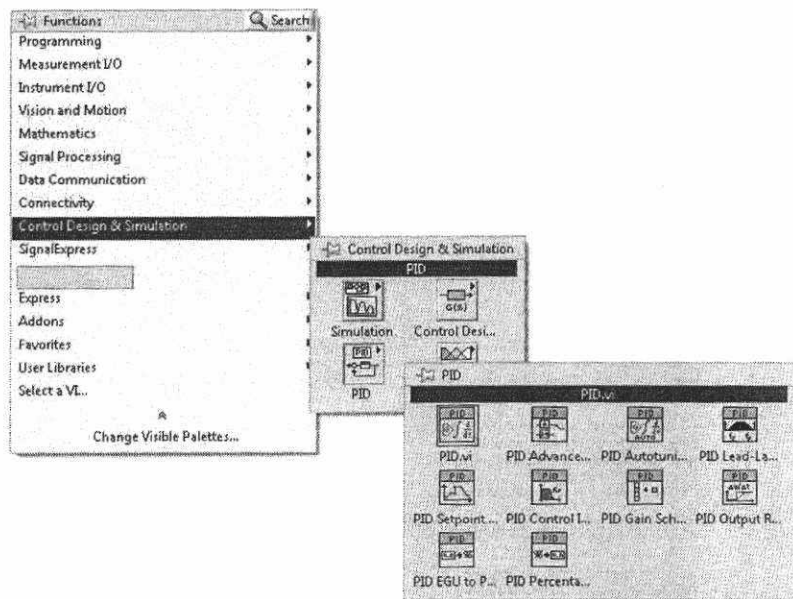
รูปที่ 3.5 แสดงฟังก์ชัน While Loop

- 2) ฟังก์ชัน Format into file คือฟังก์ชันที่ใช้เก็บข้อมูลด้าน อินพุตและ เอาท์พุทโดยจะเก็บผลอยู่ในรูปไฟล์ Excel



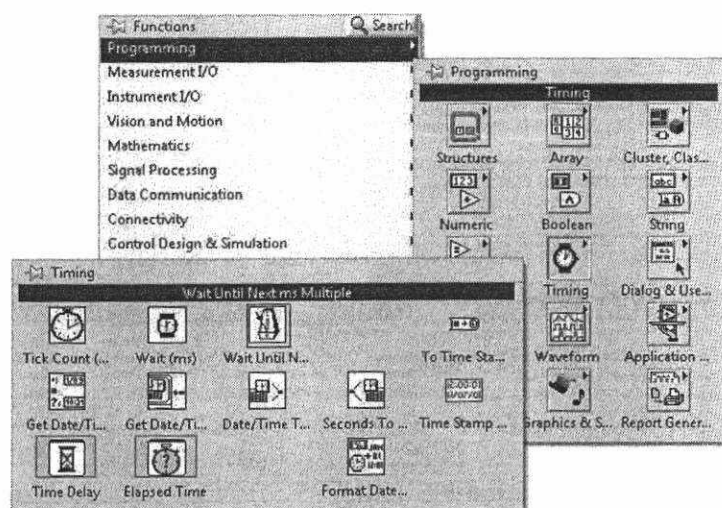
รูปที่ 3.6 แสดงฟังก์ชัน Format into file

- 3) ฟังก์ชันควบคุมแบบ พีไอดี ในฟังก์ชันนี้จะใช้การควบคุมระบบโดยจะต้องมีการอินพุตค่าพารามิเตอร์ที่จำเป็นสำหรับการควบคุม จะมีค่า Gain Parameter และ Input Parameter โดยจะมี Set Point เป็นตัวกำหนดค่าการทำงาน



รูปที่ 3.7 แสดงฟังก์ชัน พีไอดี

- 4) ฟังก์ชัน Wait Until Next ms Multiple โดยจะใช้เพื่อกำหนดการ Sampling ค่าในการควบคุมโดยจะกำหนดค่าไว้ที่ 1วินาที



รูปที่ 3.8 แสดงฟังก์ชัน Wait Until Next ms Multiple

จากฟังก์ชันการทำงานข้างต้นเราสามารถที่จะกำหนดการควบคุมโดยใช้โปรแกรม LabView โดยจะใช้การ์ด PCI 6012 เป็นอุปกรณ์ส่งผ่านข้อมูล โดยจะสามารถกำหนดค่าพารามิเตอร์และ Set Point โดยจะแสดงดังรูปที่ 3.9



รูปที่ 3.9 แสดงการทำงานและควบคุมกระบวนการ

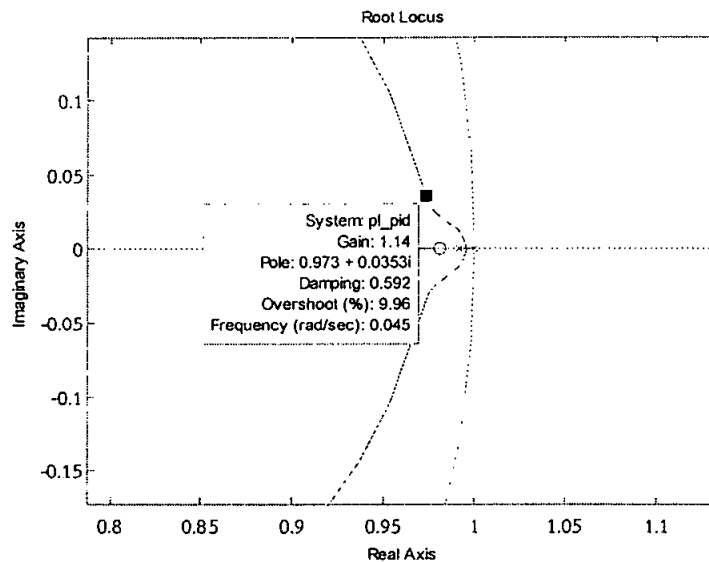
บทที่ 4

ผลการจำลองและผลการทดลองกระบวนการ

สำหรับเนื้อหาในบทนี้จะว่าด้วยเรื่องเกี่ยวกับการจำลองและการทดลองของกระบวนการ โดยนำเอาทฤษฎีทั้งหมดที่ได้กล่าวมารวมทั้งฮาร์ดแวร์และซอฟต์แวร์ใช้หาจุดทำงานจากนั้นจึงทำการจำลองกระบวนการโดยใช้โปรแกรม Simulink และทำการทดลองควบคุมกระบวนการจริงด้วยโปรแกรม LabVIEW โดยใช้ชุดจำลองอุณหภูมิ Omron Temperature Model ทำการทดสอบและเก็บผลการทดลอง

4.1 ผลการหาค่าพารามิเตอร์ควบคุมแบบพีไอดี

จากรูปแบบตั้งโปรแกรมตาม ภาคผนวก ก โดยจะใช้ทฤษฎีทางเดินราก ดังแสดงในรูปที่ 4.1 โดยใส่ค่าเริ่มต้นของลงในโปรแกรม M-file ซึ่งจะแสดงผลตำแหน่งของ Pole และเส้นทางเดินรากโดยกำหนดให้มีค่าเปอร์เซ็นต์การพุ่งเกิน (% Overshoot) เท่ากับ 10%

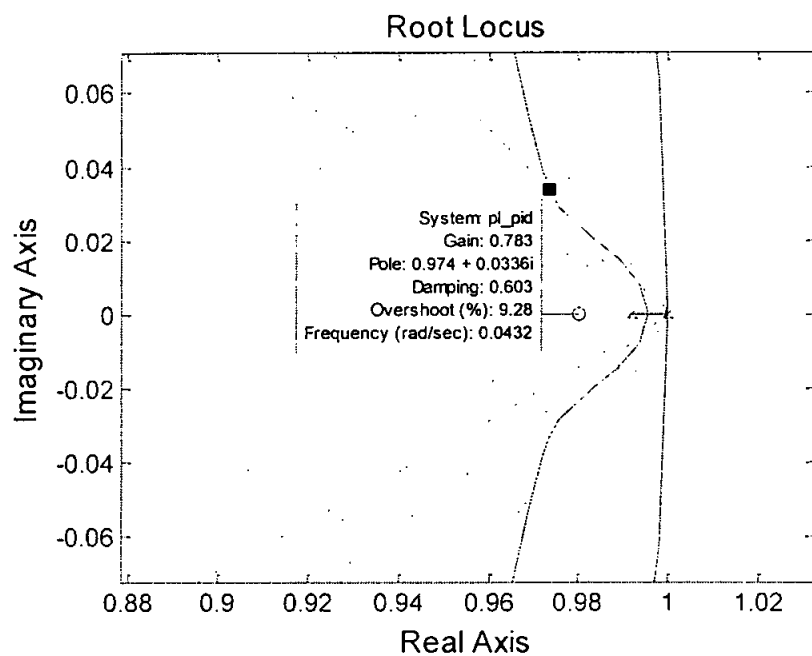


รูปที่. 4.1 แสดงผลของ Pole บนเส้นทางเดินราก

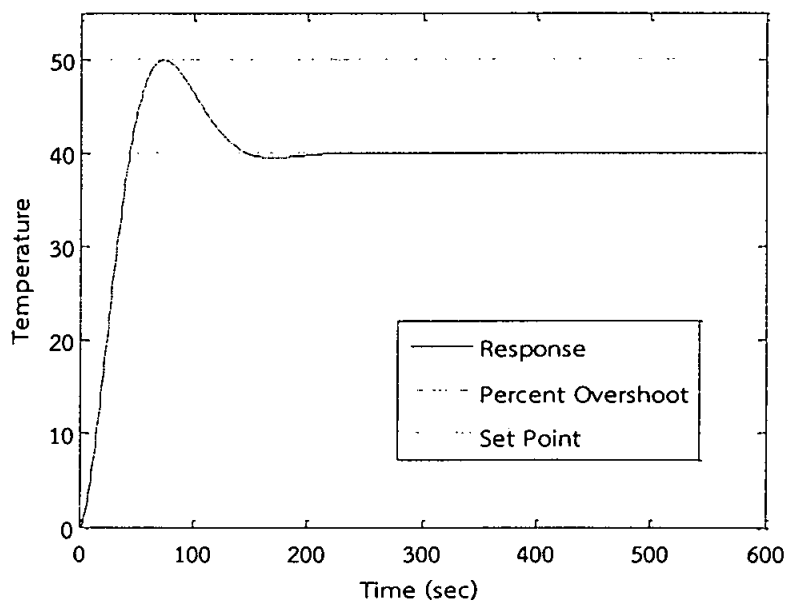
4.2 ผลการจำลองกระบวนการแบบ Proportional Integral Derivative (PID)

จากผลของโปรแกรมในภาคผนวก ก ทำให้ได้ค่าพารามิเตอร์ของระบบ โดยจะใช้ค่าที่ได้ทำการจำลองผลการตอบสนองของระบบ และในการจำลองกระบวนการจะใช้โปรแกรม Simulink ดังที่แสดงในภาคผนวก ข จากรูปที่ 4.2 ถึง 4.20 แสดงการจำลองผลตอบสนองของกระบวนการจะเห็นว่าจากโปรแกรม M-file สามารถที่จะหาค่าพารามิเตอร์ควบคุม พีไอดี ดังแสดงในผลการจำลองข้างต้นและรูปที่ 4.21 ถึง 4.24 เป็นการควบคุมกระบวนการจริง

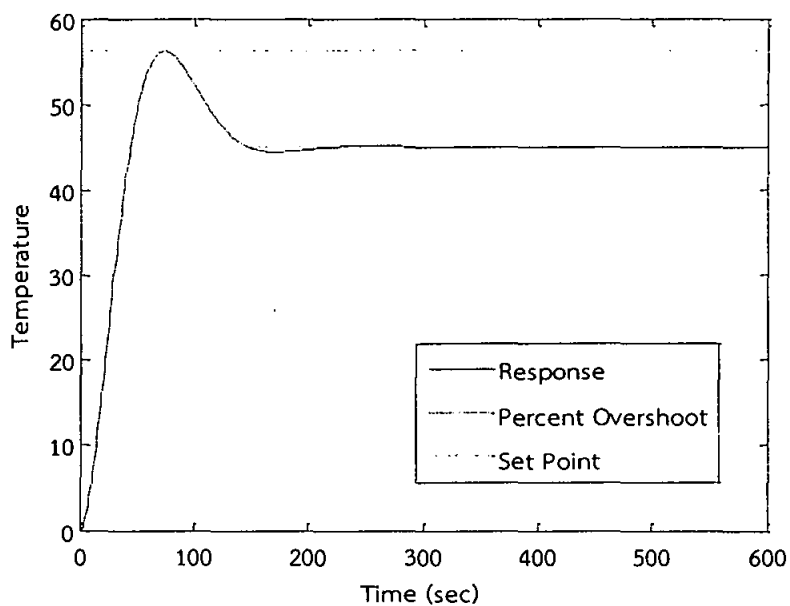
- 1) ผลการจำลองโดยกำหนดค่า Setting time=300 sec , Percent Oveshoot=10% จากค่าเริ่มต้นข้างต้นจะได้ค่าพารามิเตอร์ควบคุม $K_p = 0.3486$, $K_i = 0.0069$, $K_d = 0.4671$



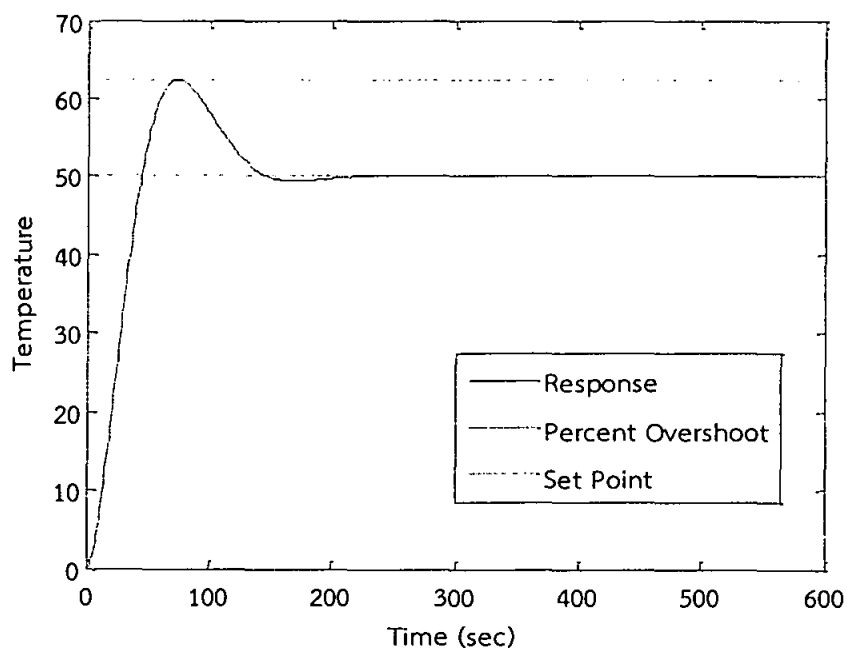
รูปที่ 4.2 แสดงตำแหน่งโพลที่ $0.974 + 0.0336i$ ค่าพารามิเตอร์ $K_p = 0.3486$,
 $K_i = 0.0069$, $K_d = 0.4671$, $OS\% = 10\%$, $T_s = 300$ sec



รูปที่ 4.3 แสดงผลการจำลองที่ 40 องศา $K_p = 0.3486$, $K_i = 0.0069$,
 $K_d = 0.4671$, $OS\% = 10\%$, $T_s = 300$ sec

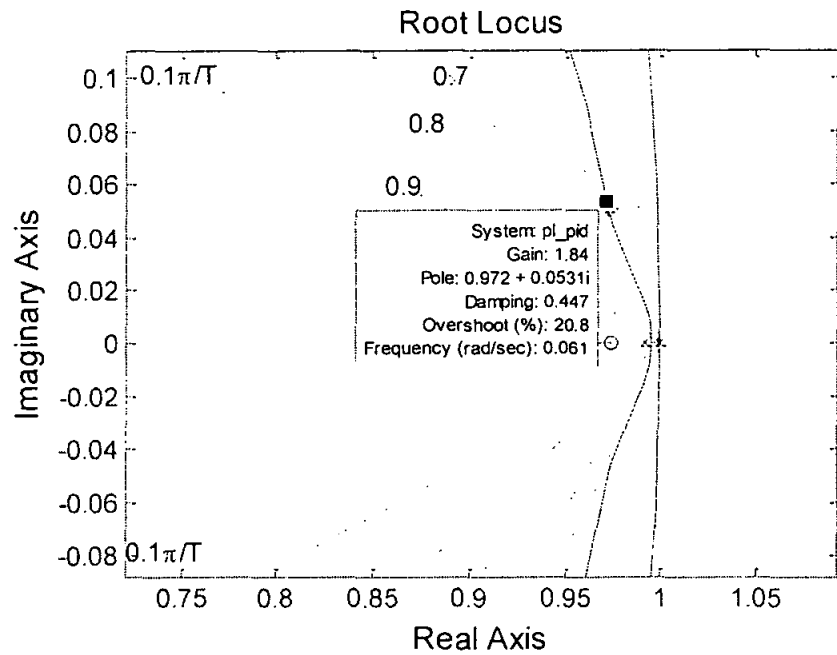


รูปที่ 4.4 แสดงผลการจำลองที่ 45 องศา $K_p = 0.3486$, $K_i = 0.0069$,
 $K_d = 0.4671$ OS% = 10%, $T_s = 300$ sec

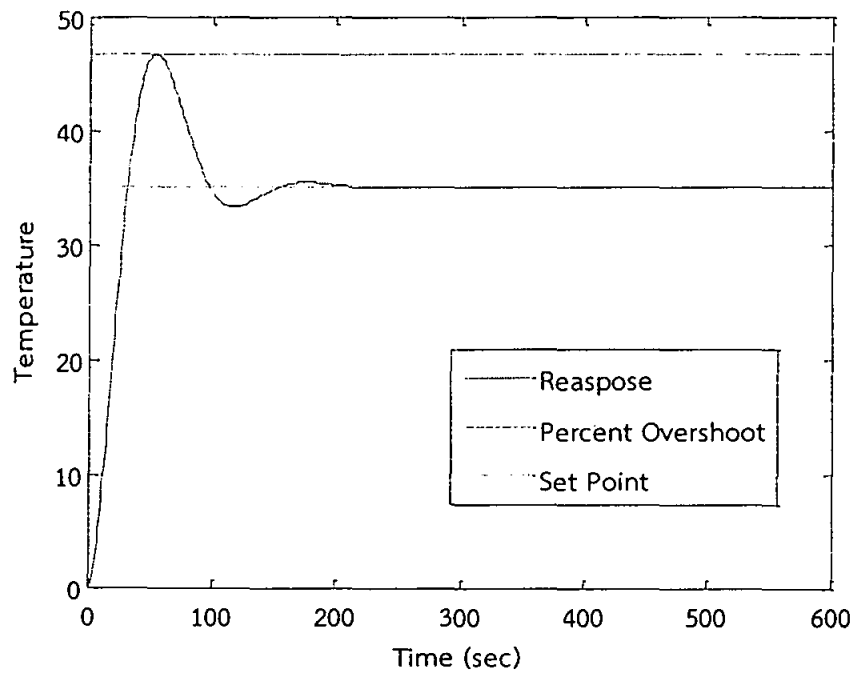


รูปที่ 4.5 แสดงผลการจำลองที่ 50 องศา $K_p = 0.3486$, $K_i = 0.0069$,
 $K_d = 0.4671$ OS% = 10%, $T_s = 300$ sec

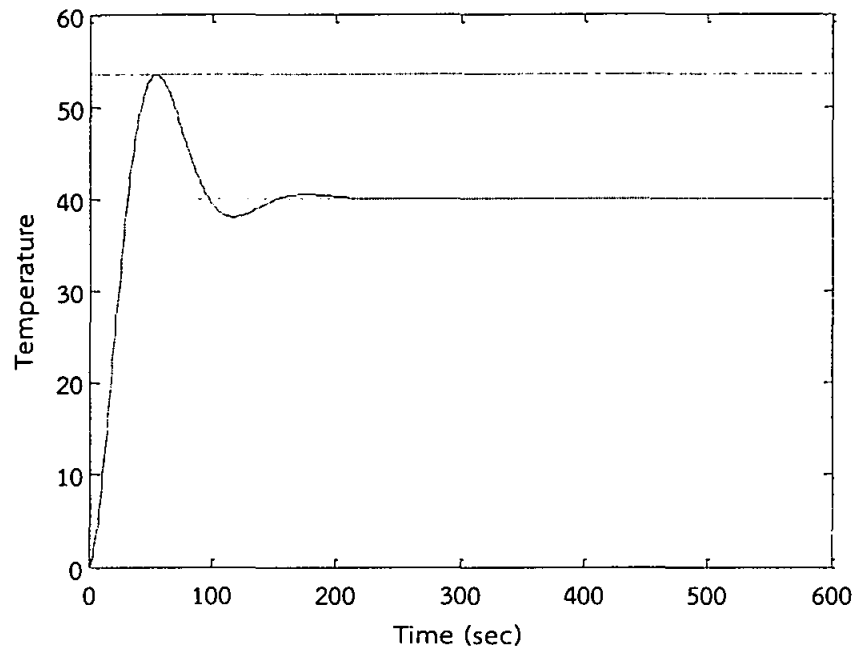
- 2) ผลการจำลองโดยกำหนดค่า Setting time=300 sec , Percent Overshoot=20% จากค่าเริ่มต้นข้างต้นจะได้ค่าพารามิเตอร์ควบคุม $K_p = 0.5424$, $K_i = 0.0137$, $K_d = 1.1713$



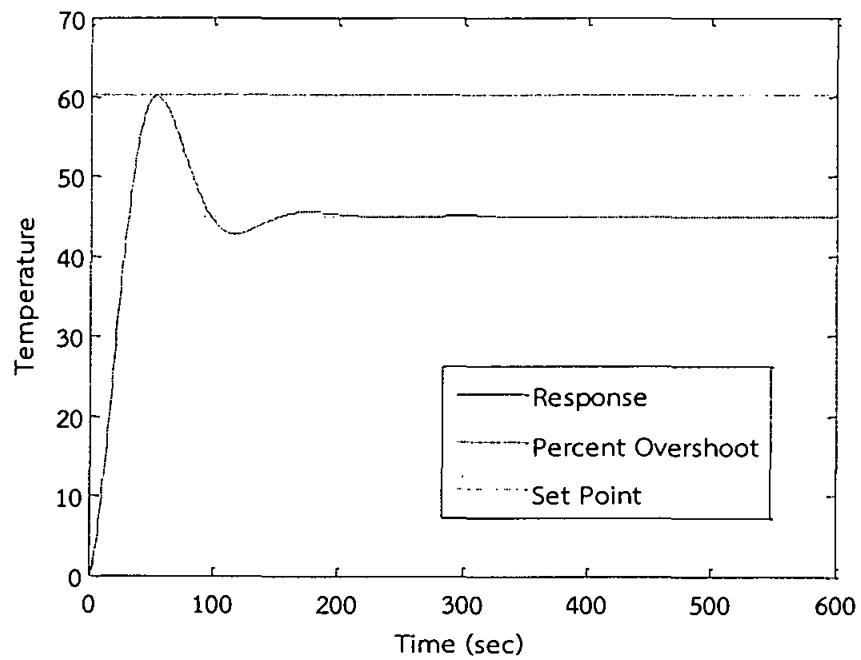
รูปที่ 4.6 แสดงตำแหน่งโพลที่ $0.974 + 0.0531i$ ค่าพารามิเตอร์ $K_p = 0.5424$, $K_i = 0.0137$, $K_d = 1.1713$, $OS\% = 20\%$, $T_s = 300$ sec



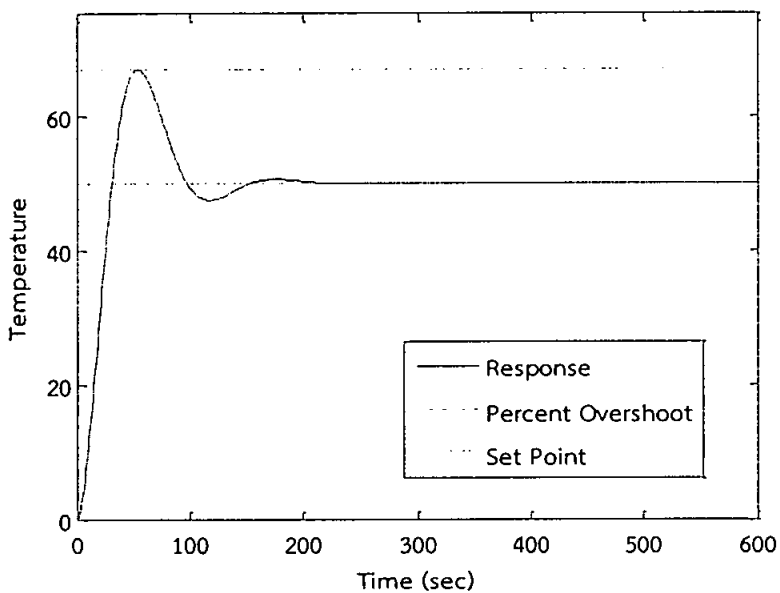
รูปที่ 4.7 แสดงผลการจำลองที่ 35 องศา $K_p = 0.5424$, $K_i = 0.0137$, $K_d = 1.1713$, $OS\% = 20\%$, $T_s = 300$ sec



รูปที่ 4.8 แสดงผลการจำลองที่ 40 องศา $K_p = 0.5424$, $K_i = 0.0137$,
 $K_d = 1.1713$, $OS\% = 20\%$, $T_s = 300$ sec

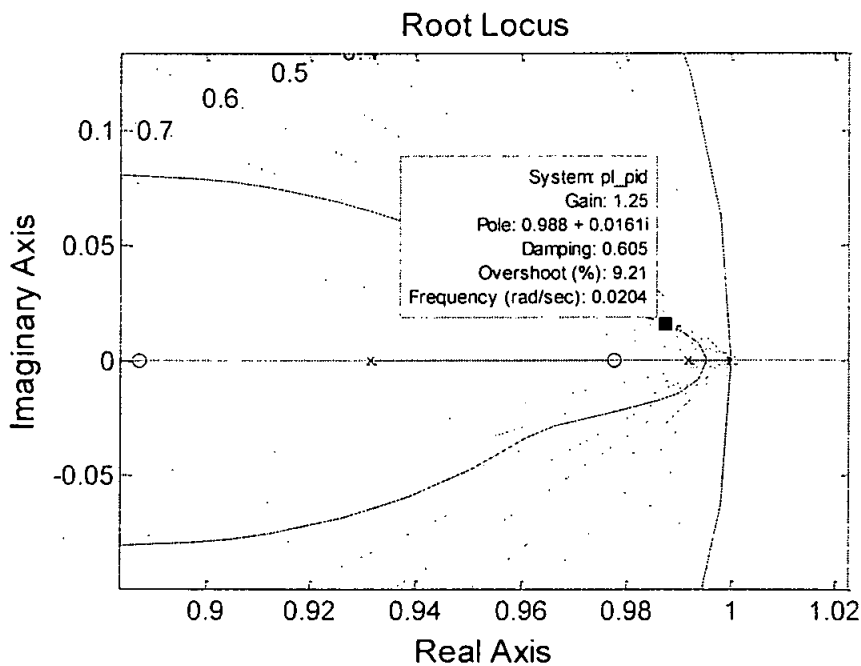


รูปที่ 4.9 แสดงผลการจำลองที่ 45 องศา $K_p = 0.5424$, $K_i = 0.0137$,
 $K_d = 1.1713$, $OS\% = 20\%$, $T_s = 300$ sec

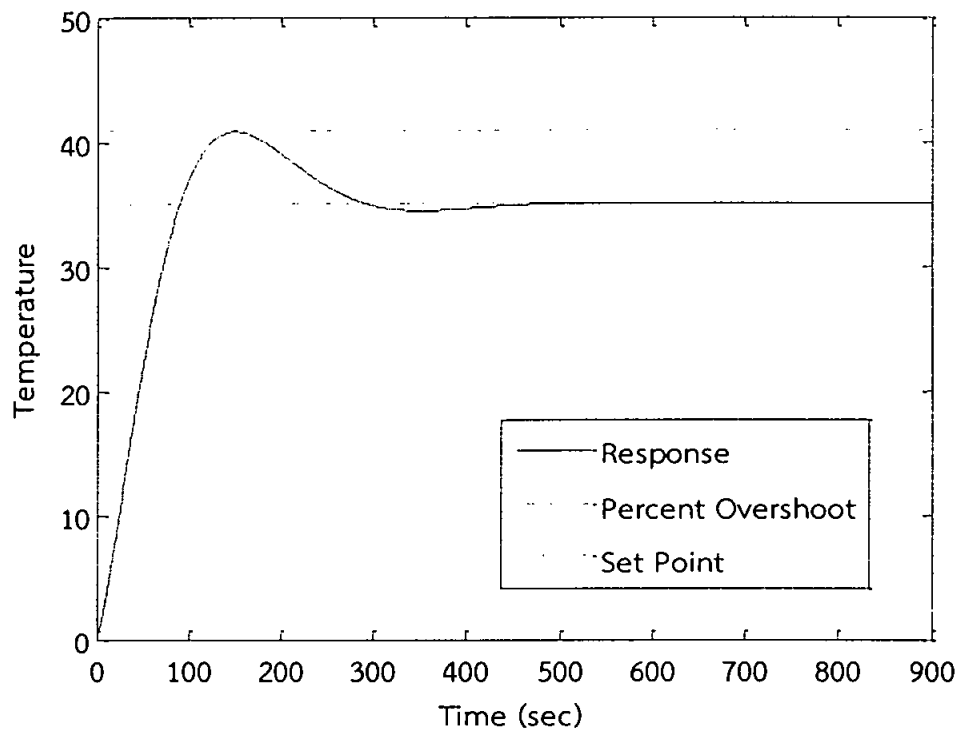


รูปที่ 4.10 แสดงผลการจำลองที่ 50 องศา $K_p = 0.5424$, $K_i = 0.0137$, $K_d = 1.1713$, $OS\% = 20\%$, $T_s = 300$ sec

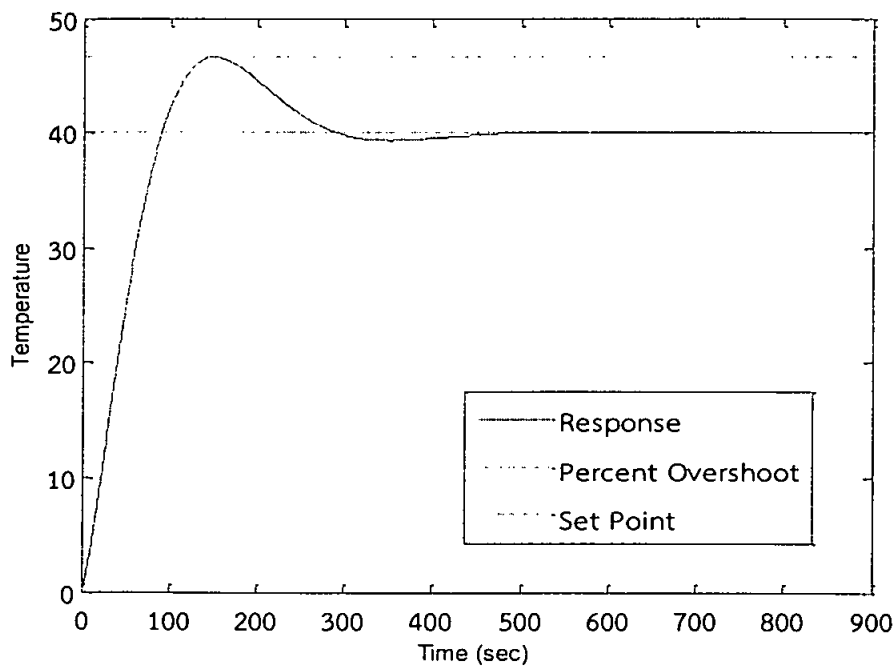
- 3) ผลการจำลองโดยกำหนดค่า Setting time=700 sec , Percent Overshoot=10% จากค่าเริ่มต้นข้างต้นจะได้ค่าพารามิเตอร์ควบคุม $K_p = 0.1474$, $K_i = 0.0028$, $K_d = 0.9854$, $T_s=700$ sec



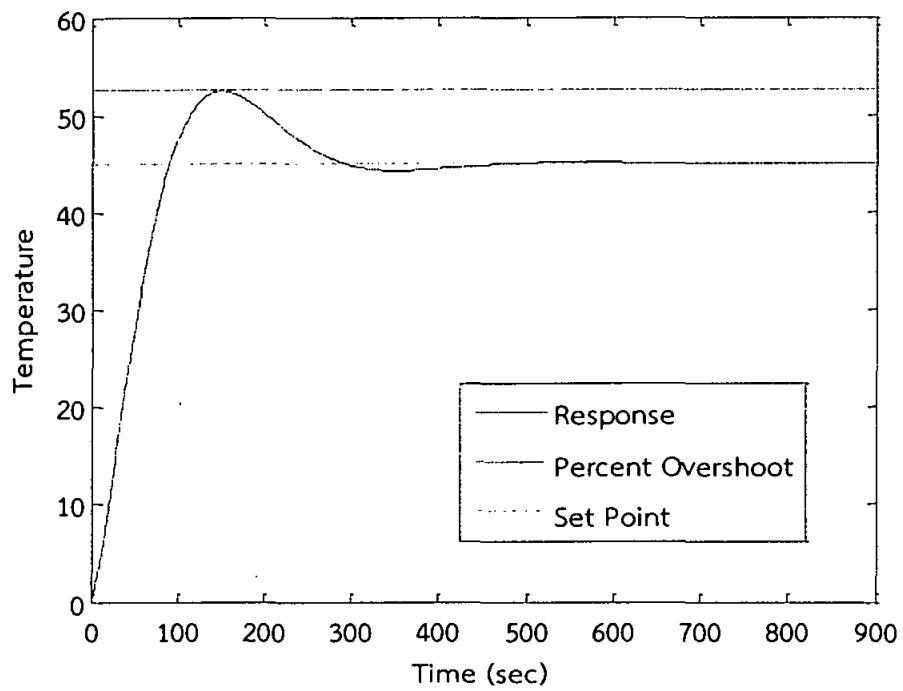
รูปที่ 4.11 แสดงตำแหน่งโพลที่ $0.988+0.0161i$ ค่าพารามิเตอร์ $K_p = 0.1474$, $K_i = 0.0028$, $K_d = 0.9854$, $OS\% = 10\%$, $T_s = 700$ sec



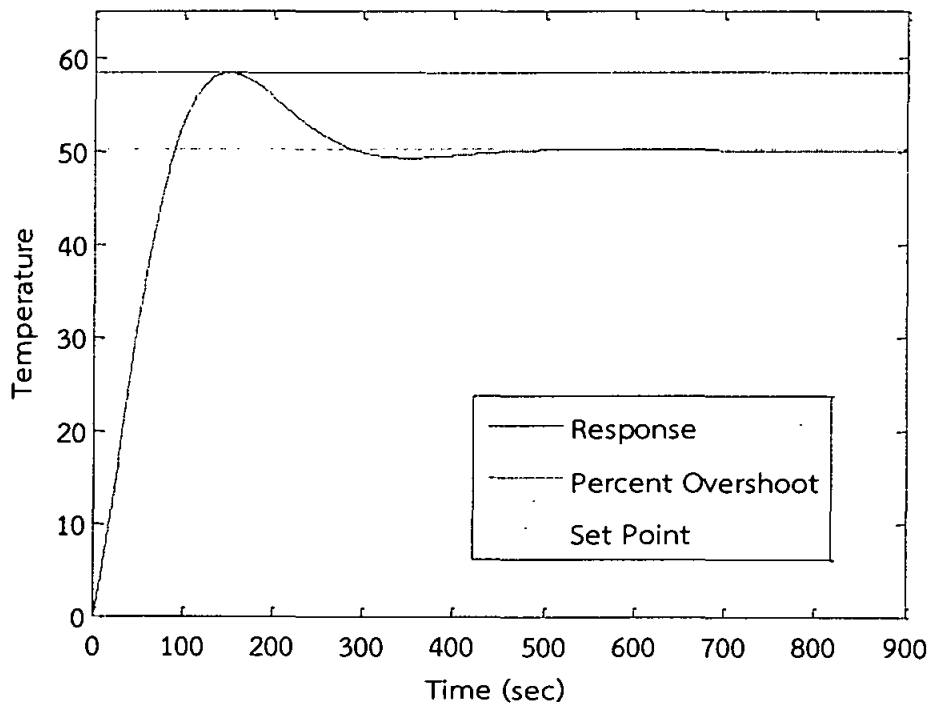
รูปที่ 4.12 แสดงผลการจำลองที่ 35 องศา $K_p = 0.1474$, $K_i = 0.0028$,
 $K_d = 0.9854$, $OS\% = 10\%$, $T_s = 700$ sec



รูปที่ 4.13 แสดงผลการจำลองที่ 40 องศา $K_p = 0.1474$, $K_i = 0.0028$,
 $K_d = 0.9854$, $OS\% = 10\%$, $T_s = 700$ sec

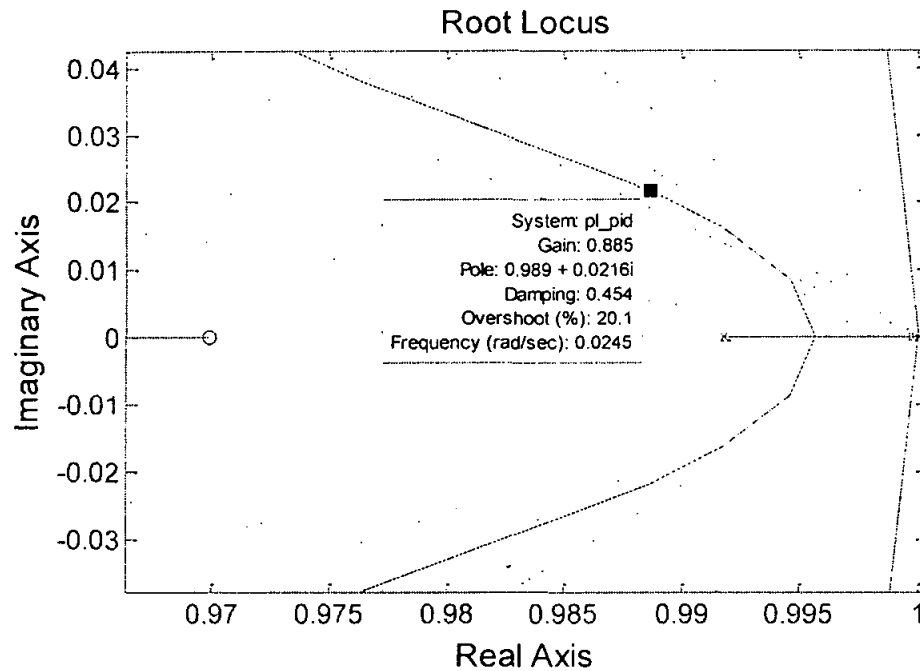


รูปที่ 4.14 แสดงผลการจำลองที่ 45 องศา $K_p = 0.1474$, $K_i = 0.0028$,
 $K_d = 0.9854$, $OS\% = 10\%$, $T_s = 700$ sec

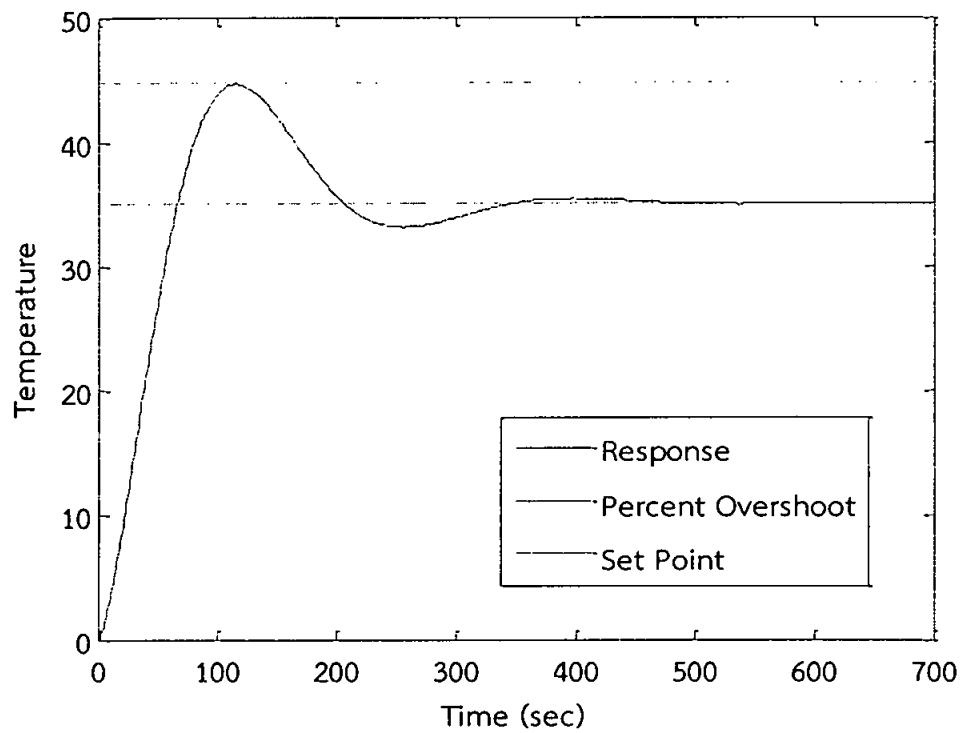


รูปที่ 4.15 แสดงผลการจำลองที่ 50 องศา $K_p = 0.1474$, $K_i = 0.0028$,
 $K_d = 0.9854$, $OS\% = 10\%$, $T_s = 700$ sec

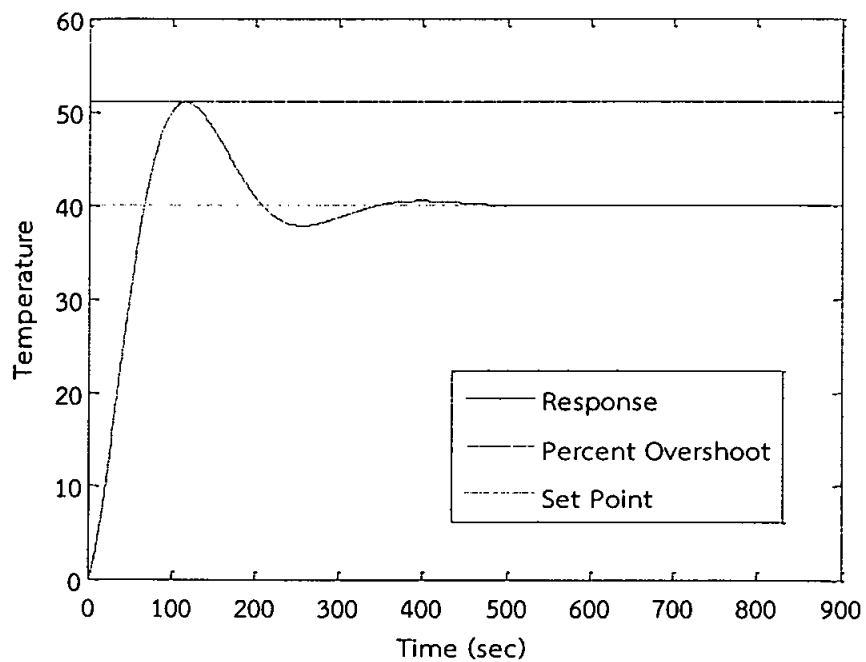
- 4) ผลการจำลองโดยกำหนดค่า Setting time=700 sec , Percent Overshoot=20% จากค่าเริ่มต้นข้างต้นจะได้ค่าพารามิเตอร์ควบคุม $K_p = 0.1712$, $K_i = 0.0046$, $K_d = 0.7443$, $T_s=700$ sec



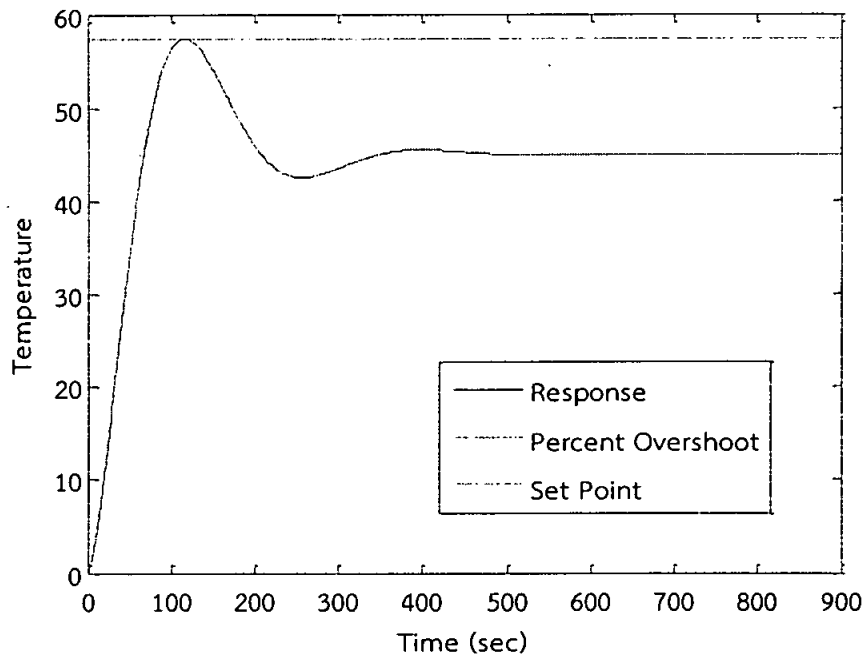
รูปที่ 4.16 แสดงตำแหน่งโพลที่ $0.988+0.0161i$ ค่าพารามิเตอร์ $K_p = 0.1474$, $K_i = 0.0028$, $K_d = 0.9854$, $OS\% = 20\%$, $T_s = 700$ sec



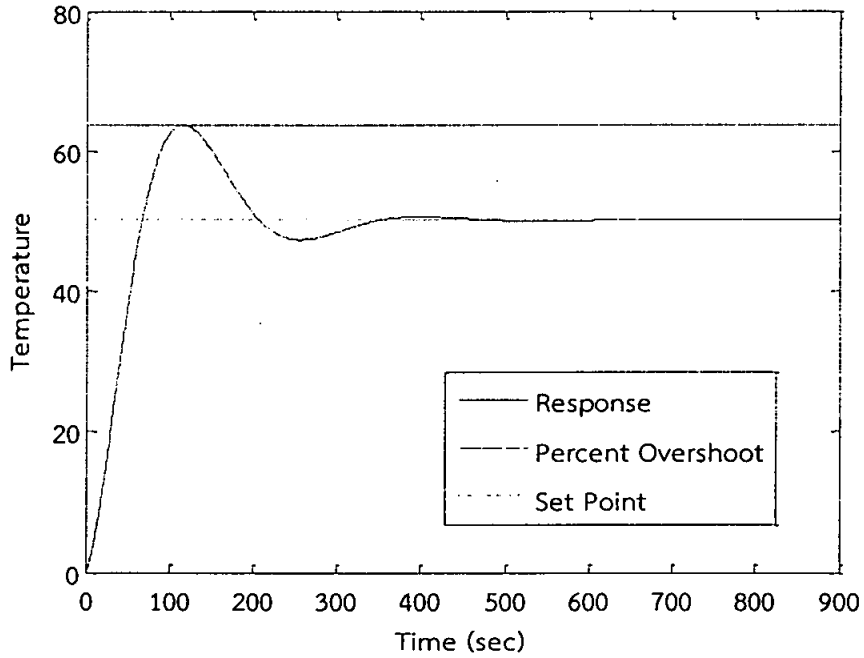
รูปที่ 4.17 แสดงผลการจำลองที่ 35 องศา $K_p = 0.1474$, $K_i = 0.0028$,
 $K_d = 0.9854$, $OS\% = 20\%$, $T_s = 700$ sec



รูปที่ 4.18 แสดงผลการจำลองที่ 40 องศา $K_p = 0.1474$, $K_i = 0.0028$,
 $K_d = 0.9854$, $OS\% = 20\%$, $T_s = 700$ sec



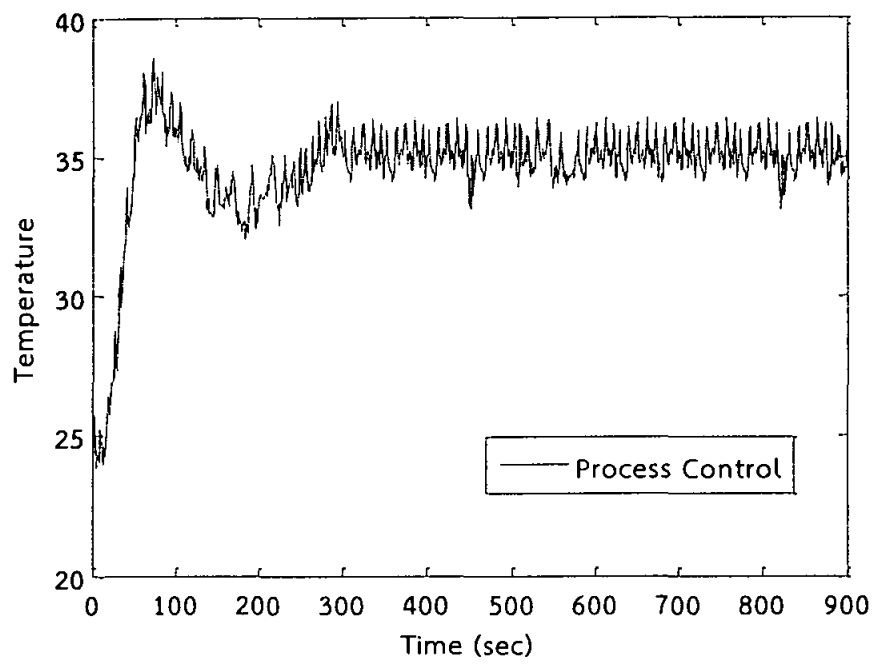
รูปที่ 4.19 แสดงผลการจำลองที่ 45 องศา $K_p = 0.1474$, $K_i = 0.0028$,
 $K_d = 0.9854$, $OS\% = 20\%$, $T_s = 700$ sec



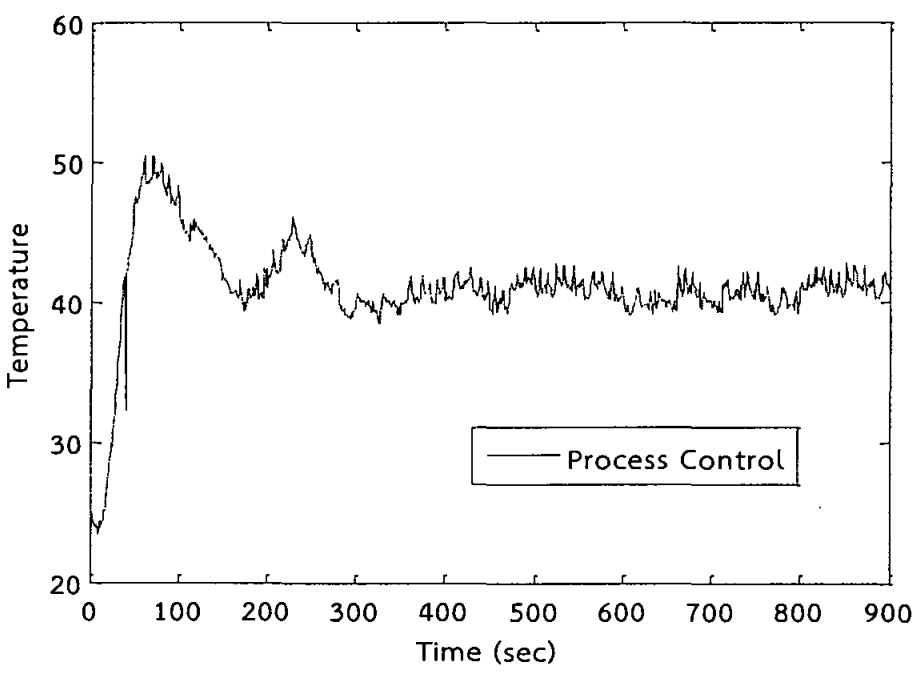
รูปที่ 4.20 แสดงผลการจำลองที่ 50 องศา $K_p = 0.1474$, $K_i = 0.0028$,
 $K_d = 0.9854$, $OS\% = 20\%$, $T_s = 700$ sec

4.3 ผลการควบคุมกระบวนการจริงแบบ Proportional Integral Derivative (PID)

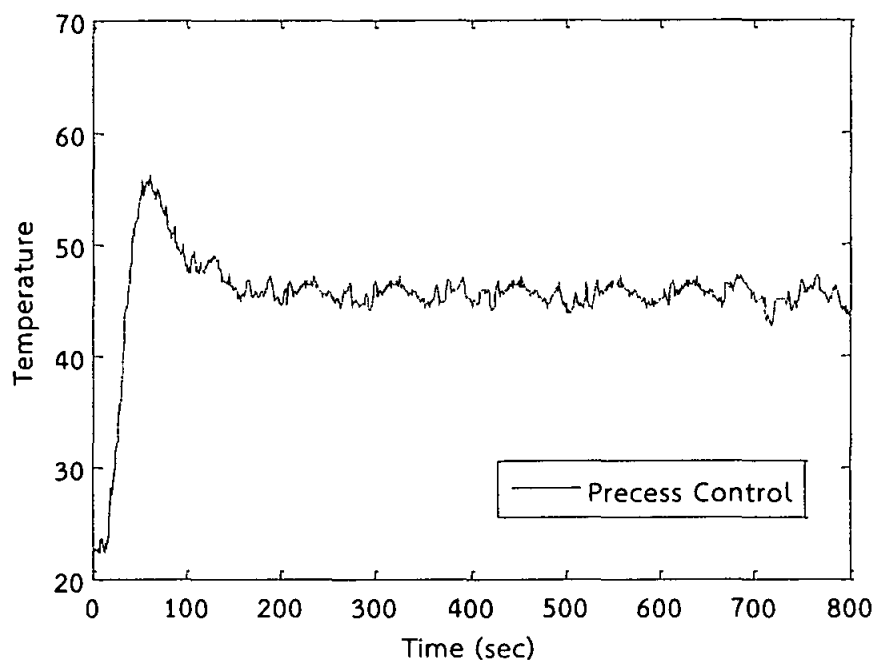
จากผลของโปรแกรมในภาคผนวก ก ทำให้ได้ค่าพารามิเตอร์ของระบบ โดยจะใช้ค่าที่ได้จากโปรแกรม M-file มาใช้ควบคุมกระบวนการจริงโดยจะใช้โปรแกรม LabView ดังที่แสดงในภาคผนวก ค และจากรูปที่ 4.6 ถึง 4.9 แสดงผลการตอบสนองของการควบคุมกระบวนการจริง



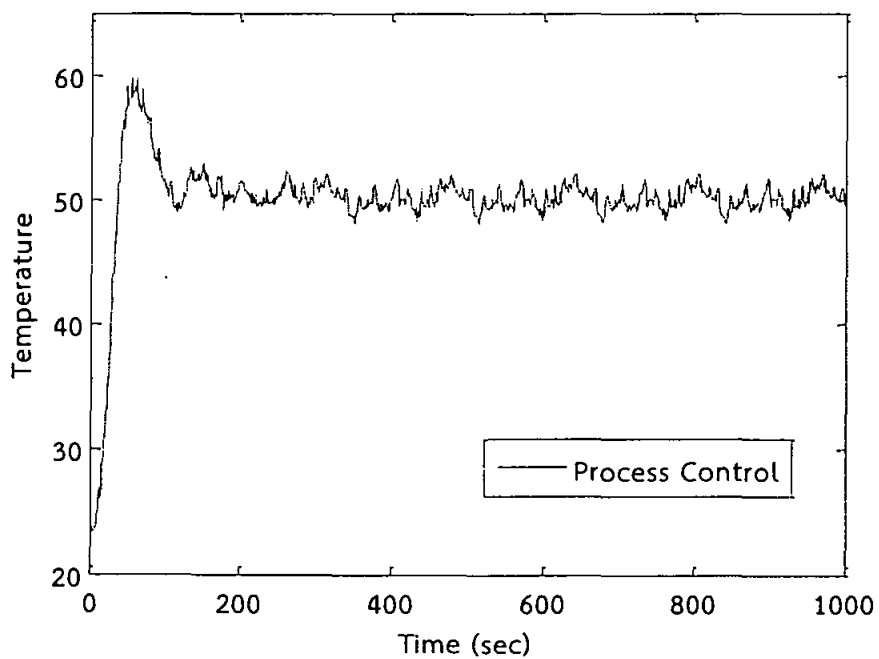
รูปที่ 4.21 แสดงการควบคุมกระบวนการที่ Set point = 35 °C



รูปที่ 4.22 แสดงการควบคุมกระบวนการที่ Set point = 40 °C



รูปที่ 4.23 แสดงการควบคุมกระบวนการที่ Set point = 45 °C



รูปที่ 4.24 แสดงการควบคุมกระบวนการที่ Set point = 50 °C

4.4 สรุปผลการทดลอง

จากผลการทดลองในรูปที่ 4.2 ถึง 4.15 จะได้ว่า ในการจำลองกระบวนการโดยใช้สมการฟังก์ชันการถ่ายโอนตามสมการที่ 3.1 จะได้ค่าพารามิเตอร์จากโปรแกรมสนับสนุน M-file โดยกำหนดค่าตัวแปรกระบวนการที่สำคัญคือ $OS\%=10\%$, $T_s=300$ เมื่อนำไปคิดคำนวณตามทฤษฎี Root locus จะได้ตำแหน่งโพลบนทางเดินรากเป็น $0.974+0.0336i$ จะได้ค่าพารามิเตอร์ควบคุมแบบ พีไอดี $K_p=0.3484$, $K_i=0.0069$, $K_d=0.4671$ หลังจากนั้นทำการทดลองโดยการเปลี่ยนแปลงค่าตัวแปรกระบวนการใหม่เป็น $OS\%=20\%$, $T_s=300$ และเมื่อนำไปคิดคำนวณตามทฤษฎี Root locus จะได้ตำแหน่งโพลบนทางเดินรากเป็น $0.974+0.0531i$ จะได้ค่าพารามิเตอร์ควบคุมแบบ พีไอดี $K_p=0.5424$, $K_i=1.1713$, $K_d=1.1713$ จากผลการจำลองการทำงานจะได้ว่า ผลตอบสนองของทั้งสองค่า นั้นจะมีค่าเปอร์เซ็นต์โอเวอร์ชูตประมาณ ($OS\%$)=18% ซึ่งมีค่าเกินจากที่กำหนดไว้ในตัวแปรเริ่มต้น คือค่าเปอร์เซ็นต์โอเวอร์ชูต ($OS\%$)=10% เนื่องจากกำหนดค่าเซตตั้งทาร์มที่สั้นแต่ก็สามารถควบคุมผลตอบสนองของระบบได้ในเวลาที่ต้องการ

จากรูปที่ 4.16 ถึง 4.20 เมื่อทำการกำหนดค่ากระบวนการโดยเพิ่มช่วงเวลาเซตตั้งทาร์มให้มีค่า $T_s=700$ และ $OS\%=20\%$ เมื่อนำไปคิดคำนวณตามทฤษฎี Root locus จะได้ตำแหน่งโพลบนทางเดินรากเป็น $0.988+0.0161i$ จะได้ค่าพารามิเตอร์ควบคุมแบบ พีไอดี $K_p=0.1474$, $K_i=0.0028$, $K_d=0.9854$ จากผลการจำลองกระบวนการจะได้ว่าผลตอบสนองของระบบมีความใกล้เคียงกับค่าตัวแปรที่กำหนดไว้

จากผลการทดลองกระบวนการจริงในรูปที่ 4.21 ถึง 4.24 จะได้ผลตอบสนองของระบบได้ใกล้เคียงกับการจำลองระบบเมื่อนำไปคำนวณตามทฤษฎี Root locus จะได้ตำแหน่งโพลบนทางเดินรากที่ตำแหน่ง $0.973+0.0353i$ แสดงในการควบคุมผลการตอบสนองของกระบวนการจริงมีค่าใกล้เคียงกับผลการจำลองระบบ

สรุปว่าผลในการทดสอบการทำงานร่วมของชุดจำลองอุณหภูมิในงานวิจัยที่นำมาศึกษานี้ ควรให้ค่าเซตตั้งทาร์ม (T_s) มีค่าระหว่าง 700 วินาที ถึง 1100 วินาที โดยที่เปอร์เซ็นต์โอเวอร์ชูต ($OS\%$) ควรไม่เกิน 20% จะทำให้การควบคุมมีเสถียรภาพ

บทที่ 5

สรุปผลวิจัยและข้อเสนอแนะ

5.1 สรุปผลงานวิจัย

ในกระบวนการอุตสาหกรรมนั้นการควบคุมกระบวนการมีความจำเป็นอย่างยิ่งโดยการควบคุมนั้นมีการควบคุมหลายอย่างโดยการควบคุมแบบ PID ก็เป็นที่รู้จักอย่างแพร่หลายในทางอุตสาหกรรมเช่นกัน โดยในปัจจุบันนี้เราสามารถที่จะจำลองการควบคุมกระบวนการพร้อมทั้งสามารถควบคุมกระบวนการจริงหลังจากที่ได้ทำการจำลองกระบวนการแล้ว

จากผลการทดลองเมื่อทำการหาผลตอบสนองของกระบวนการโดยการ Open loop จึงนำค่าที่ได้ไปใส่ในโปรแกรม M-file และใส่ค่าตัวแปรดังนี้ เปอร์เซ็นโอเวอร์ชูต (percent overshoot), ค่าเซตติงไทม์ (setting time) และค่าทรานสเฟอร์ฟังก์ชัน (transfer function) โดยใช้ทฤษฎี Root locus หาค่าพารามิเตอร์ควบคุมแบบ พีไอดี โดยผลที่ได้สามารถตรวจสอบได้ว่าโพลที่ได้มานั้นอยู่ทางเดินรากหรือไม่ และสามารถตรวจสอบค่าพารามิเตอร์ พีไอดี ได้จากโปรแกรมและสามารถตรวจสอบการจำลองกระบวนการโดยใช้โปรแกรม Simulink ในการควบคุมกระบวนการจริงจะใช้ค่าพารามิเตอร์ควบคุมกระบวนการที่ได้จากโปรแกรมสนับสนุน m-file โดยใช้โปรแกรม Labview แสดงผลและควบคุมกระบวนการ

โดยจากค่าพารามิเตอร์ดังกล่าวเมื่อนำไปควบคุมกระบวนการจริง ผลจากการควบคุมกระบวนการนั้นมีผลตอบสนองของกระบวนการมีค่าใกล้เคียงกับผลการจำลองโดยใช้โปรแกรม Simulink ดังนั้นโปรแกรม m-file สามารถที่จะใช้หาค่าพารามิเตอร์ควบคุมแบบ พีไอดี ของระบบได้ จะเห็นได้ว่าการควบคุมกระบวนการจริงนั้นผลที่ได้มีค่าที่ใกล้เคียงกับผล simulation

5.2 ปัญหาและแนวทางแก้ไขปัญหาในงานวิจัย

ปัญหาส่วนมากที่พบในงานวิจัยนี้ คือ การทดลองเมื่อทำการทดลองแล้วครั้งหนึ่งแล้วจะไม่สามารถทดลองในครั้งต่อไปได้ในทันที โดยจะต้องรอให้อุณหภูมิกลับมาคงที่ตามอุณหภูมิห้องก่อนจึงสามารถเริ่มการทดลองใหม่ได้อีกครั้ง ดังนั้นเราควรควบคุมอุณหภูมิภายในบริเวณที่ทำการทดลองให้มีค่าที่ใกล้เคียงค่าเริ่มต้นให้มากที่สุดหรือ ใช้พัดลมเป่าเพื่อจะระบายความร้อนได้เร็วขึ้น

ปัญหาที่พบคือชุดจำลองกระบวนการเนื่องจากการออกแบบโดยให้เซ็นเซอร์ ชิดติดกับอุปกรณ์ให้ความร้อนทำให้การอ่านค่าอุณหภูมิและมีลำบากในการควบคุมอุณหภูมิ ในการแก้ไขควรที่จะมีการออกแบบชุดจำลองกระบวนการใหม่โดยออกแบบให้เซ็นเซอร์ออกห่างจากชุดให้ความร้อนหรือทำเป็นห้องควบคุมอุณหภูมิขนาดเล็ก

5.3 แนวทางการพัฒนางานวิจัย

ในการควบคุมระบบนี้เป็นพื้นฐานของการศึกษาและนำไปสู่การหาค่าพารามิเตอร์ของกระบวนการ ผู้ศึกษาคิดว่าควรที่จะปรับปรุงแนวทางของ ทฤษฎีที่ใช้ในการหาค่าพารามิเตอร์ในแบบต่างๆ เพื่อจะได้มีแนวทางในการควบคุมกระบวนการต่างๆ และเป็นทางเลือกในการใช้การใช้ทฤษฎีแบบต่างๆ

ภาคผนวก

ภาคผนวก ก

โปรแกรม M-file ที่ใช้หาค่าพารามิเตอร์ควบคุมแบบ พีไอดี

รูปแบบการเขียนโปรแกรม M-File เพื่อหาค่าพารามิเตอร์

```

close all
clear
clc
sysc=tf(10.25,[3000 110 1]);
OS=0.1;
ts_old=300;
ts_new=ts_old/1;
ts_new_d=ts_old/2.
Ts=1;%0.02;

%%%%%%%%%%Calculate PI Controller%%%%%%%%%%
s1=-damp*wn+j*wn*sqrt(1-damp^2);
s2=-damp*wn-j*wn*sqrt(1-damp^2);

z1=exp(s1*Ts);
z2=exp(s2*Ts);

sysd=c2d(sysc,Ts);

figure
plot(z1,'*')
hold on
plot(z2,'*')
rlocus(sysd)

[numPdis,denPdis]=tfdata(sysd,'v');
[z,p,k]=tf2zp(numPdis,denPdis);

thetap1=(180/pi)*angle(polyval([1 -p(1)],z1));
thetap2=(180/pi)*angle(polyval([1 -p(2)],z1));
thetaz1=(180/pi)*angle(polyval([1 -z(1)],z1));

```

```

thetaplant=thetaz1-(thetap1+thetap2);

thetacon=-180-thetaplant;
alpha=-1*((imag(z1)/(tan(thetacon*pi/180+atan(imag(z1)/(real(z1)-1)))))-real(z1));
pl_pi=series(tf([1 -alpha],[1 -1],Ts),sysd);
controller_pi=tf([1 -alpha],[1 -1],Ts);

figure
plot(z1,'*')
hold on
rlocus(pl_pi)

%%%%%%%%%%Calculate PD Controller%%%%%%%%%%

damp_d=-log(OS)/sqrt((log(OS))^2+pi^2);
wn_d=4/(damp_d*ts_new_d);

s1_d=-damp_d*wn_d+j*wn_d*sqrt(1-damp_d^2);
s2_d=-damp_d*wn_d-j*wn_d*sqrt(1-damp_d^2);

z1_d=exp(s1_d*Ts);
z2_d=exp(s2_d*Ts);

figure
plot(z1_d,'*')
hold on
plot(z2_d,'*')
rlocus(pl_pi)

[numPdis_d,denPdis_d]=tfdata(pl_pi,'v');
[z_d,p_d,k_d]=tf2zp(numPdis_d,denPdis_d);

thetap1_d=(180/pi)*angle(polyval([1 -p_d(1)],z1_d));
thetap2_d=(180/pi)*angle(polyval([1 -p_d(2)],z1_d));

```

```

thetap3_d=(180/pi)*angle(polyval([1 -p_d(3)],z1_d));
thetaz1_d=(180/pi)*angle(polyval([1 -z_d(1)],z1_d));
thetaz2_d=(180/pi)*angle(polyval([1 -z_d(2)],z1_d));
thetaplant_d=(thetaz1_d+thetaz2_d)-(thetap1_d+thetap2_d+thetap3_d);
thetacon_pid=-180-thetaplant_d;
alpha_pid=-1*((imag(z1_d)/(tan(thetacon_pid*pi/180+atan(imag(z1_d)/(real(z1_d))))))-real(z1_d));

pl_pid=series(tf([1 -alpha_pid],[1 0],Ts),pl_pi);
controller_pid=tf([1 -alpha_pid],[1 0],Ts);

figure

plot(z1_d,'*')
hold on
rlocus(pl_pid)

[num_pid,den_pid]=tfdata(pl_pid,'v');
K=abs(polyval(den_pid,z1_d))/abs(polyval(num_pid,z1_d));

controller_pid=K*series(controller_pi,controller_pid);

[num_pid,den_pid]=tfdata(controller_pid,'v');

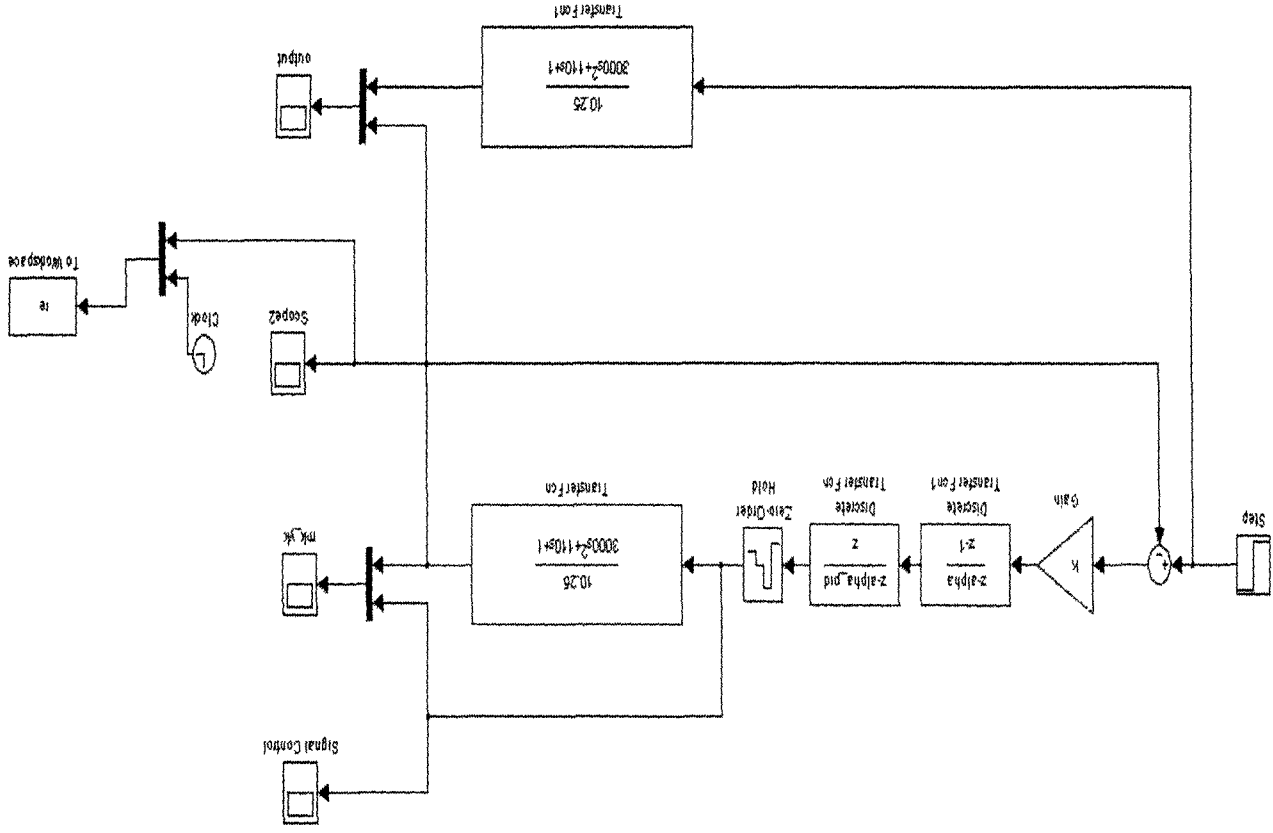
Kd=num_pid(3);
Kp=-num_pid(2)-2*Kd;
Ki=num_pid(1)-Kd-Kp;

```

ภาคผนวก ข

รูปแบบโปรแกรม Simulink ที่ใช้จำลองกระบวนการแบบ พีไอดี

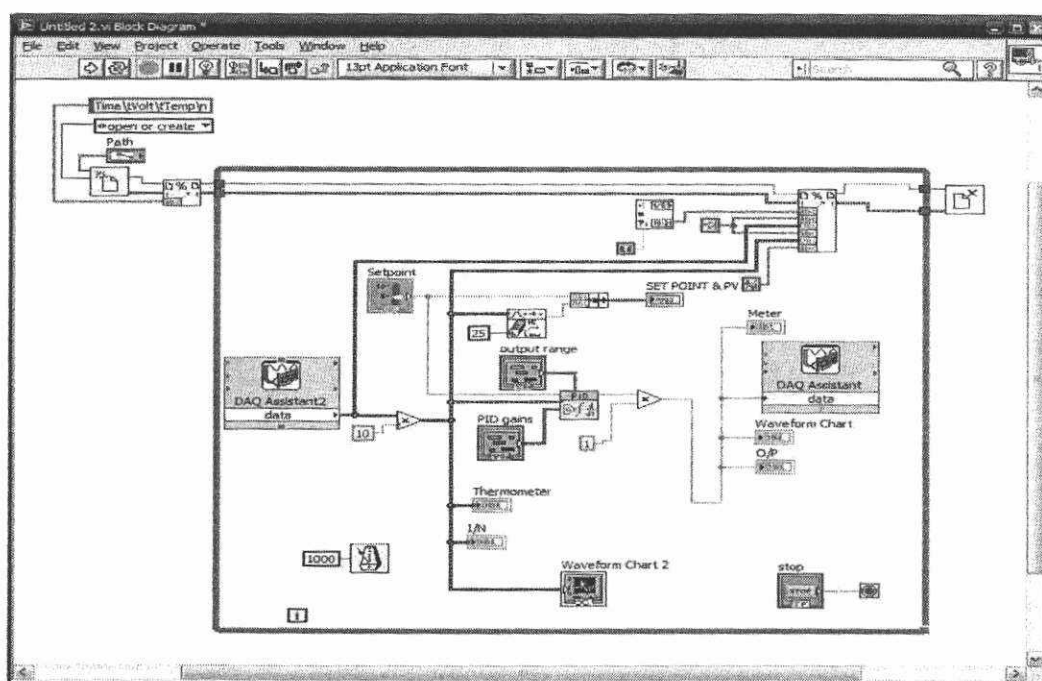
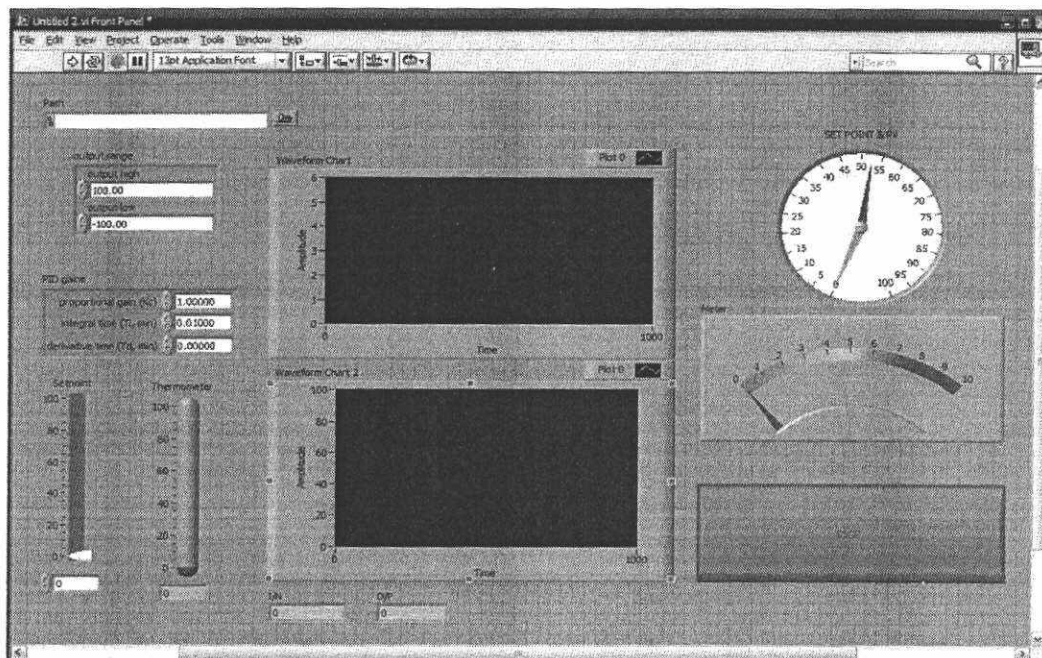
รูปแบบนี้เป็นการ Simulink เพื่อจำลองการทำงานของระบบควบคุม



ภาคผนวก ค

โปรแกรม Labview ที่ใช้ควบคุมกระบวนการจริง

แสดงรูปแบบการเขียนโปรแกรม Labview และ Panel Control



ภาคผนวก ง

ข้อมูลการ์ด National Instrument PCI 6014

NI 6013/6014 Family Specifications

This document lists the I/O terminal summary and specifications for the NI 6013/6014 family of devices. This family includes the following devices:

- NI PCL-6013
- NI PCL-6014

I/O Terminal Summary



Note With NI-DAQmx, National Instruments revised its terminal names so they are easier to understand and more consistent among NI hardware and software products. The revised terminal names used in this document are usually similar to the names they replace. For a complete list of Traditional NI-DAQ (Legacy) terminal names and their NI-DAQmx equivalents, refer to *Terminal Name Equivalents of the E Series Help*.

Table 1. I/O Terminals

Terminal Name	Terminal Type and Direction	Impedance Input/ Output	Protection (V) On/Off	Source (mA at V)	Sink (mA at V)	Rise Time (ns)	Bias
AI <B,15>	AI	100 G Ω in parallel with 100 pF	25/15	—	—	—	± 200 pA
AI SENSE	AI	100 G Ω in parallel with 100 pF	25/15	—	—	—	± 200 pA
AI GND	—	—	—	—	—	—	—
AO 0'	AO	0.1 Ω	Short-circuit to ground	5 at 10	5 at -10	—	—
AO 1'	AO	0.1 Ω	Short-circuit to ground	5 at 10	5 at -10	—	—
AO GND	—	—	—	—	—	—	—
D-GND	—	—	—	—	—	—	—
-5 V	—	0.1 Ω	Short-circuit to ground	1 A fuse	—	—	—

Table 1. I/O Terminals (Continued)

Terminal Name	Terminal Type and Direction	Impedance Input/Output	Protection (V) On/Off	Source (mA at V)	Sink (mA at V)	Rise Time (ns)	Bias
IO<0..7>	DIO	—	$V_{CC} + 0.5$	10 at $(V_{CC} - 0.4)$	24 at 0.4	1.1	50 k Ω pu
AI HOLD COMP	DO	—	—	3.5 at $(V_{CC} - 0.4)$	5 at 0.4	1.5	50 k Ω pu
LXT STROBE*	DO	—	—	3.5 at $(V_{CC} - 0.4)$	5 at 0.4	1.5	50 k Ω pu
PI 0/ (AI START TRIG)	DIO	—	$V_{CC} + 0.5$	3.5 at $(V_{CC} - 0.4)$	5 at 0.4	1.5	50 k Ω pu
PI 1/ (AI REF TRIG)	DIO	—	$V_{CC} + 0.5$	3.5 at $(V_{CC} - 0.4)$	5 at 0.4	1.5	50 k Ω pu
PI 2/ (AI CONV CLK)*	DIO	—	$V_{CC} + 0.5$	3.5 at $(V_{CC} - 0.4)$	5 at 0.4	1.5	50 k Ω pu
PI 3/ CTR 1 SOURCE	DIO	—	$V_{CC} + 0.5$	3.5 at $(V_{CC} - 0.4)$	5 at 0.4	1.5	50 k Ω pu
PI 4/CTR 1 GATE	DIO	—	$V_{CC} + 0.5$	3.5 at $(V_{CC} - 0.4)$	5 at 0.4	1.5	50 k Ω pu
CTR 1 OUT	DO	—	—	3.5 at $(V_{CC} - 0.4)$	5 at 0.4	1.5	50 k Ω pu
PI 5/ (AO SAMP CLK)*	DIO	—	$V_{CC} + 0.5$	3.5 at $(V_{CC} - 0.4)$	5 at 0.4	1.5	50 k Ω pu
PI 6/ (AO START TRIG)	DIO	—	$V_{CC} + 0.5$	3.5 at $(V_{CC} - 0.4)$	5 at 0.4	1.5	50 k Ω pu
PI 7/ (AI SAMP CLK)	DIO	—	$V_{CC} + 0.5$	3.5 at $(V_{CC} - 0.4)$	5 at 0.4	1.5	50 k Ω pu
PI 8/ CTR 0 SOURCE	DIO	—	$V_{CC} + 0.5$	3.5 at $(V_{CC} - 0.4)$	5 at 0.4	1.5	50 k Ω pu
PI 9/CTR 0 GATE	DIO	—	$V_{CC} + 0.5$	3.5 at $(V_{CC} - 0.4)$	5 at 0.4	1.5	50 k Ω pu
CTR 0 OUT	DO	—	—	3.5 at $(V_{CC} - 0.4)$	5 at 0.4	1.5	50 k Ω pu
TRIG OUT	DO	—	—	3.5 at $(V_{CC} - 0.4)$	5 at 0.4	1.5	50 k Ω pu

* Indicates active low.
 * NI 6014 only.
 AI = Analog Input DIO = Digital Input/Output pu = pull-up
 AO = Analog Output DO = Digital Output

Note: The tolerance on the 50 k Ω pull-up resistors is large. Actual value might range between 17 k Ω and 100 k Ω .

Specifications

The following specifications are typical at 25 °C unless otherwise noted

Analog Input

Input Characteristics

Number of channels..... 16 single-ended or
8 differential
(software-selectable
per channel)

Type of ADC Successive
approximation

Resolution 16 bits, 1 in 65,536

Max sampling rate 200 kS/s guaranteed

Input signal ranges

Device Gain (Software-Selectable)	Range
0.5	±10 V
1	±5 V
10	±500 mV
100	±50 mV

Input coupling DC

Maximum working voltage..... Each input should remain
within ±11 V of ground

Overvoltage protection

Signal Name	Powered Off	Powered On
AI <0..15>	±15 V	±25 V
AI SENSE	±15 V	±25 V

FIFO buffer size 512 samples

Data transfers DMA, interrupts,
programmed I/O

DMA modes..... Scatter-gather (Single
transfer, demand transfer)

Number of DMA channels..... 1¹

Configuration memory size 512 words

¹ The NI 6013/6014 has one DMA channel to be shared by all resources on the device.

Accuracy Information

Nominal Range at Full Scale (V)	Absolute Accuracy							Relative Accuracy Resolution (μV)	
	% of Reading		Offset (μV)	Noise + Quantization (μV)		Absolute Accuracy at Full Scale (mV)	Temp Drift (% $^{\circ}\text{C}$)	Single Pt.	Averaged
	24 Hours	1 Year		Single Pt.	Averaged				
± 10	0.0658	0.0700	1897.5	933.0	82.40	8.984	0.0010	1084.9	108.5
± 5	0.0158	0.0200	959.8	466.5	41.20	2.003	0.0005	542.4	54.2
± 0.5	0.0658	0.0700	115.8	56.2	5.035	0.471	0.0010	66.3	6.6
± 0.05	0.0658	0.0700	31.4	31.40	3.067	0.069	0.0010	40.4	4.0

Note: Accuracies are valid for measurements following an internal E-Series calibration. Averaged numbers assume dithering and averaging of 100 single-channel readings. Measurement accuracies are listed for operational temperatures within $\pm 1^{\circ}\text{C}$ of internal calibration temperature and $\pm 10^{\circ}\text{C}$ of external or factory-calibration temperature. NI recommends a one-year calibration interval. The Absolute Accuracy at Full Scale calculations were performed for a maximum range input voltage (for example, 10 V for the ± 10 V range) after one year, assuming 100 points of averaged data. Go to ni.com/info and enter info code rdspec for example calculations.

Transfer Characteristics

INL ± 1.5 LSB typ.
 ± 3.0 LSB max

No missing codes 16 bits, guaranteed

Offset error

Pre-gain error after calibration ± 2.0 μV max

Pre-gain error before calibration ± 28.8 mV max

Post-gain error after calibration ± 305 μV max

Post-gain error before calibration ± 40.2 mV max

Gain error (relative to calibration reference)

After calibration (gain = 1) ± 74 ppm of reading max

Before calibration $\pm 18,900$ ppm of reading max

Gain $\neq 1$ with gain error

adjusted to 0 at gain = 1 ± 300 ppm of reading max

Amplifier Characteristics

Input impedance

Normal powered on 100 G Ω in parallel with
 100 pF

Powered off 820 Ω

Overload 820 Ω

Input bias current ± 200 pA

Input offset current ± 100 pA

CMRR (DC to 60 Hz)

Gain 0.5, 1.0 85 dB

Gain 10, 100 96 dB

Dynamic Characteristics

Bandwidth

Signal	Bandwidth
Small (-3 dB)	425 kHz
Large (1% THD)	450 kHz

Settling time for full-scale step

Gain 100 ± 2 LSB, 5 μs typ

Gain 1, 10 ± 2 LSB, 5 μs max

Gain 0.5 ± 4 LSB, 5 μs typ

System noise (LSB_{rms} , including quantization)

Gain	LSB_{rms}
0.5, 1.0	0.9
10	1.1
100	6.7

Crosstalk (DC to 100 kHz)

Adjacent channels -75 dB

Other channels ≤ -90 dB

Stability

Recommended warm-up time	15 min
Offset temperature coefficient	
Pregain	$\pm 20 \mu\text{V}/^\circ\text{C}$
Postgain	$\pm 175 \mu\text{V}/^\circ\text{C}$
Gain temperature coefficient.....	$\pm 32 \text{ ppm}/^\circ\text{C}$

Analog Output (NI 6014 Only)

Output Characteristics

Number of channels	2 voltage
Resolution	16 bits, 1 in 65,536
Max update rate	
DMA	10 kHz, system dependent
Interrupts	1 kHz, system dependent
Type of DAC	Double-buffered, multiplying
FIFO buffer size	None
Data transfers	DMA, interrupts, programmed I/O
DMA modes	Scatter-gather (Single transfer, demand transfer)
Number of DMA channels.....	1 ¹

Accuracy Information

Nominal Range (V)		Absolute Accuracy				
Positive FS	Negative FS	% of Reading		Offset (μV)	Temp Drift (%/°C)	Absolute Accuracy at Full Scale (μV)
		24 Hours	1 Year			
10	-10	0.0252	0.0337	2.461	0.0005	5.827

Note: Accuracies are valid for measurements following an internal E Series calibration. Averaged numbers assume dithering and averaging of 100 single-channel readings. Measurement accuracies are listed for operational temperatures within $\pm 1^\circ\text{C}$ of internal calibration temperature and $\pm 10^\circ\text{C}$ of external or factory-calibration temperature. NI recommends a one-year calibration interval. The Absolute Accuracy at Full Scale calculations were performed for a maximum range input voltage (for example, 10 V for the ± 10 V range) after one year, assuming 100 points of averaged data. Go to ni.com/info and enter info code `rdspec` for example calculations.

¹ The NI 6013/6014 has one DMA channel to be shared by all resources on the device.

Transfer Characteristics

Relative accuracy (INL)	± 3 LSB, typ
DNL	± 2 LSB, typ
Monotonicity	15 bits
Offset error	
After calibration	± 372 μ V max
Before calibration	± 250 mV max
Gain error (relative to internal reference)	
After calibration	± 75 ppm
Before calibration	$\pm 22,700$ ppm

Voltage Output

Range	± 10 V
Output coupling	DC
Output impedance	0.1 Ω max
Current drive	± 5 mA max
Protection	Short-circuit to ground
Power-on state (steady state)	± 350 mV
Initial power-up glitch	
Magnitude	± 6.0 V
Duration	4 ms
Power reset glitch	
Magnitude	± 3.0 V
Duration	3 ms

Dynamic Characteristics

Settling time for full-scale step	8 μ s to ± 1 LSB accuracy
Slew rate	4 V/ μ s
Noise	360 μ V _{rms} , DC to 400 kHz
Midscale transition glitch	
Magnitude	± 200 mV
Duration	2.0 μ s

Stability

Offset temperature coefficient	± 128 μ V/ $^{\circ}$ C
Gain temperature coefficient	± 26.8 ppm/ $^{\circ}$ C

Digital I/O

Number of channels	8 input/output
Compatibility	TTL/CMOS

P0.<0..7>

Digital logic levels

Level	Min	Max
Input low voltage	0 V	0.8 V
Input high voltage	2.0 V	5.0 V
Input low current ($V_{in} = 0$ V)	—	-320 μ A
Input high current ($V_{in} = 5$ V)	—	3.33 mA
Output low voltage ($I_{OL} = 24$ mA)	—	0.4 V
Output high voltage ($I_{OH} = -13$ mA)	4.35 V	—

Power-on state	Input (high-impedance), 1.5 k Ω pull down to D GND
Data transfers	Programmed I/O
Max transfer rate	50 kwords/s, system dependent

Timing I/O

Number of channels	
Up/down counter/timers	2
Frequency scaler	1
Resolution	
Up/down counter/timers	24 bits
Frequency scaler	4 bits
Compatibility	5 V TTL/CMOS

Digital logic levels

Level	Min	Max
Input low voltage	0.0 V	0.8 V
Input high voltage	2.0 V	5.0 V
Output low voltage ($I_{out} = 5$ mA)	—	0.4 V
Output high voltage ($I_{out} = -3.5$ mA)	4.35 V	—

Base clocks available

Up/down counter/timers	20 MHz, 100 kHz
Frequency scaler	10 MHz, 100 kHz

Base clock accuracy	±0.01%
Max external source frequency	
Up/down counter/timers	20 MHz
External source selections	PFI <0..9>
External gate selections	PFI <0..9>
Min source pulse duration.....	10 ns in edge-detect mode
Min gate pulse duration	10 ns in edge-detect mode
Data transfers	
Up/down counter/timers	DMA (scatter-gather), interrupts, programmed I/O
Frequency scaler	Programmed I/O

Digital Trigger

Purpose	
Analog input	Start, reference, and pause trigger, sample clock
Analog output	Start and pause trigger, sample clock
Counter/timers	Source, gate
Source	PFI <0..9>
Compatibility	5 V TTL
Response	Rising or falling edge
Pulse width.....	10 ns min
External input for digital trigger	
Protection	-0.5 V to VCC + 0.5 V

Calibration

Recommended warm-up time	15 minutes
Calibration interval	1 year
External calibration reference	>6 and <10 V
Onboard calibration reference	
DC level.....	5.000 V (±3.5 mV), over full operating temperature, actual value stored in EEPROM
Temperature coefficient	±5 ppm/°C max
Long-term stability	±15 ppm/ $\sqrt{1,000 \text{ h}}$

Power Requirement

+5 VDC (±5%).....0.7 A



Note Excludes power consumed through +5 V available at the I/O connector

Power available at I/O connector +4.65 to +5.25 VDC
at 1 A

Physical

Dimensions
(not including connectors).....16.2 cm × 9.2 cm
(6.4 in. × 3.6 in.)

Weight

PCI-6013.....101 g (3.5 oz)
PCI-6014.....104 g (3.6 oz)

I/O connector.....68-pin male SCSI-II type

Maximum Working Voltage

Maximum working voltage refers to the signal voltage plus the common-mode voltage.

Channel-to-earth.....11 V,
Installation Category I

Channel-to-channel

11 V,
Installation Category I

Environmental

Operating temperature	0 to 55 °C
Storage temperature.....	-20 to 70 °C
Relative humidity	10 to 90% noncondensing
Maximum altitude	2,000 m
Pollution Degree (indoor use only)	2

Safety

NI 6013/6014 devices meet the requirements of the following standards for safety and electrical equipment for measurement, control, and laboratory use:

- IEC 61010-1, EN 61010-1
- UL 61010-1
- CAN/CSA-C22.2 No. 61010-1



Note For UL and other safety certifications, refer to the product label or visit ni.com/certification, search by model number or product line, and click the appropriate link in the Certification column.

Electromagnetic Compatibility

Emissions.....EN 55011 Class A at 10 m
FCC Part 15A above
1 GHz

Immunity.....EN 61326:1997
A2:2001, Table 1

CE, C-Tick, and FCC Part 15 (Class A) Compliant



Note For EMC compliance, operate this device with shielded cabling.

CE Compliance

This product meets the essential requirements of applicable European Directives, as amended for CE marking, as follows:

Low-Voltage Directive (safety).....73/23/EEC

Electromagnetic Compatibility

Directive (EMC).....89/336/EEC



Note Refer to the Declaration of Conformity (DoC) for this product for any additional regulatory compliance information. To obtain the DoC for this product, visit ni.com/certification, search by model number or product line, and click the appropriate link in the Certification column.

AI 8	34	68	AI 0
AI 1	33	67	AI GND
AI GND	32	66	AI 9
AI 10	31	65	AI 2
AI 3	30	64	AI GND
AI GND	29	63	AI 11
AI 4	28	62	AI SENSE
AI GND	27	61	AI 12
AI 13	26	60	AI 5
AI 6	25	59	AI GND
AI GND	24	58	AI 14
AI 15	23	57	AI 7
AO 0	22	56	AI GND
AO 1	21	55	AO GND
NC	20	54	AO GND
P0.4	19	53	D GND
D GND	18	52	P0.0
P0.1	17	51	P0.5
P0.6	16	50	D GND
D GND	15	49	P0.2
+5 V	14	48	P0.7
D GND	13	47	P0.3
D GND	12	46	AI HOLD COMP
PFI 0/AI START TRIG	11	45	EXT STROBE
PFI 1/AI REF TRIG	10	44	D GND
D GND	9	43	PFI 2/AI CONV CLK
+5 V	8	42	PFI 3/CTR 1 SRC
D GND	7	41	PFI 4/CTR 1 GATE
PFI 5/AO SAMP CLK	6	40	CTR 1 OUT
PFI 6/AO START TRIG	5	39	D GND
D GND	4	38	PFI 7/AI SAMP CLK
PFI 9/CTR 0 GATE	3	37	PFI 8/CTR 0 SRC
CTR 0 OUT	2	36	D GND
FREQ OUT	1	35	D GND

NC = No Connect

Figure 2. NI 6014 Pinout

National Instruments, NI, ni.com, and LabVIEW are trademarks of National Instruments Corporation. Refer to the *Terms of Use* section on ni.com/legal for more information about National Instruments trademarks. Other product and company names mentioned herein are trademarks or trade names of their respective companies. For patents covering National Instruments products, refer to the appropriate location: **Help>Patents** in your software, the `patents.txt` file on your CD, or ni.com/patents.

ภาคผนวก จ

ผลงานที่ได้รับการตีพิมพ์

Lecture Notes in Engineering and Computer Science

IMECS 2012

International MultiConference of
**Engineers and Computer
Scientists 2012**

Volume II

**Hong Kong
14-16 March, 2012**

S. L. Ao
Oscar Castillo
Craig Douglas
David Dagan Feng
Jeong-A Lee (Eds.)

IA ENG

International Association of Engineers

ISBN: 978-963-19251-9-0
ISSN: 2078-0958

Analysis Parameter PID with the Root Locus Theory by used M-file Mat Lab

A. Kraisuntronlertphop, S. Gunpanich, *Member. IAENG*

Abstract—This paper presents the program M-file with Mat lab for PID parameter by root locus technique. The program can be used with other transfer function of any plant. The characteristics of the M-file can be defined for transfer function, percent overshoot and settling time to study the responding of this theory, the simulink Mat lab will be used to simulate the parameter set by M-file.

Index—MAT LAB, Root locus, Transfer function, Percent overshoot, Simulink, Settling time.

I. INTRODUCTION

There are various control theories that are being used significantly. However PID control system is still widely used and accepted in the process industries. Especially, the section that requires temperature control, which is the typical application of PID controller.

The root locus plot was introduced for the design of feedback control systems by Evans in 1948 [1] and since then has become a standard and commonly used tool in control system education and practice.

Undergraduate control systems textbooks still give emphasis to root locus plotting [2]–[3], and research papers present computational methods, computer implementations [4], [5], and new approaches [8],[9] for the root locus method. The method is an essential design and analysis tool in most of the control system software [6], [7].

Several methods can be used to study parameters of PID controller. However, this paper presents root locus theory that requires m-file MAT LAB program to determine the parameters' control. M-file program can be configured to transfer functions of system, percent over shoot and settling time. Furthermore, PID parameters that receive from m-file program can be checked from by examining results of the Pole on the top of Z-Plane. Moreover, the simulation of simulink mat lab's response system can be checked PID parameters as well.

Manuscript received December 08, 2011; revised January 31, 2012
This work was supported in part by Dept of Instrumentation and Control Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok, 10520

A Kraisuntronlertphop Author is with the Graduate school, Dept of Instrumentation and Control Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand, 10520 (e-mail warin_mut@hotmail.com)

S Gulphanich Author is with the Graduate school, Dept of Instrumentation and Control Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand, 10520 (e-mail kgsuphan1960@hotmail.com)

ISBN : 978-988-19251-9-0

ISSN : 2078-0958 (Print); ISSN : 2078-0966 (Online)

In finding control parameters, the obtained results can be

controlled in various systems according to programmers. Consequently, M-file program is one of many methods that can show parameters and response system for study PID control theories.

II. PID METHOD WITH M-FILE MAT LAB PROCEDURE

In order to design the M-file program, PID control is the main theory that needs to be apply first in this paper. Finally, the Root locus method is required to determine various parameters close loop pole and location of the system in z-plane. The following figure is the designed program procedures.

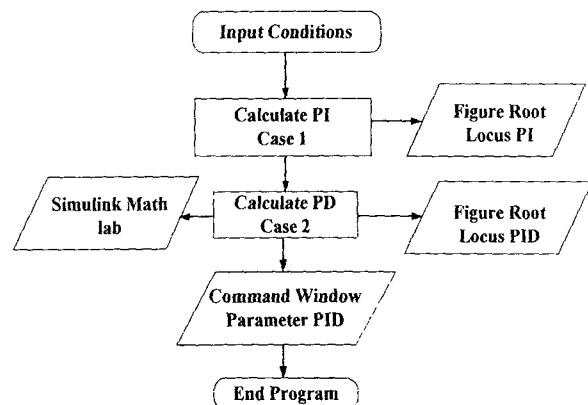


Fig1 M-file Flow Chart Calculate System

Fig 1, Shows the process of the m-file. There are three Input conditions. Firstly, it is the value of the transfer function. Secondly, the percentage of overshoot which can be configured to start at 0% to 100% according to designer's desires. The final parameter is the settling time which is the identification of a time period. These three parameters will determine the response of the procedure. Moreover, it can find controlled parameter which are K_p , K_i , K_d . This calculation process divides in two procedures which are PI calculation and then use the result from the PI to calculate PD as follow.

Input Conditions: Enter the parameter into program with m-file transfer function (tf), percent overshoot (%OS), and the settling time (ts) for default in the procedure.

Case1: From close loop transfer function of the value of z is the close loop pole system is the characteristic equation.

$$1 + KG_{cPD}(z)G_p(z) = 0 \tag{1}$$

With the condition of the magnitude criteria, by the value of z corresponds to the equation.

$$|KG_{cPD}(z)G_p(z)| = 1 \tag{2}$$

When is the condition of phase condition angle, sum of the system.

$$\sum \alpha_{zeros} - \sum \theta_{poles} = -180 \tag{3}$$

Case2: The position of the pole of a PI controller in the z plane is shown at position $+1$. With the zero appears at $z = \frac{K_p}{K_i + K_p}$ within the range between 0 and 1, or the unit circle.

Therefore the value of the angle θ_{PI} is more than α_{PI} which is ϕ or $\theta_{request}$ that can be obtained from

$$\phi = \alpha_{PI} - \theta_{PI} = \theta_{request} \tag{4}$$

Then $\theta_{request}$ is

$$\theta_{request} = \tan^{-1}\left(\frac{\beta}{\sigma - \alpha}\right) - \tan^{-1}\left(\frac{\beta}{\sigma - 1}\right) \tag{5}$$

Where β is the angle of zero that measures from the angle of the vector which lines from zero (z_i) to S points that is a positive angle and an anticlockwise. Moreover, the σ is the position of real axis. When included both values together, it will make the pole PID controller appears in z - plane (The pole has two positions at 0 and $+1$). The point of zero appears in two positions at the equation.

$$z = \frac{(K_p + 2K_D) \pm \sqrt{K_p^2 - 4K_D K_i}}{2(K_p + K_i + K_D)} \tag{6}$$

From the above process, we are able to determine the parameters K_p, K_i, K_D , also check value of the pole, check value of the and zero and the angle in z -plane. The Mat lab command window can be used to check the all parameters. We can exam the results by simulated process response of the parameters by using K_p, K_i, K_d that obtains from the process shown in Figure 1. In this paper, the parameters of m -file is used to test the process. Simulink Mat lab is used to simulate the response and verify the initial conditions weather it matches with the response procedure.

III. RESULT AND SIMULATE M-FILE MAT LAB

In this research, we use a temperature model to study for finding the transfer function. The open loop is necessary to test the process for determining the characteristics equation. After that mat lab program is used to analyze and compared

ISBN : 978-988-19251-9-0

ISSN : 2078-0958 (Print); ISSN : 2078-0966 (Online)

the characteristic equation between the open loop result and simulate mat lab.

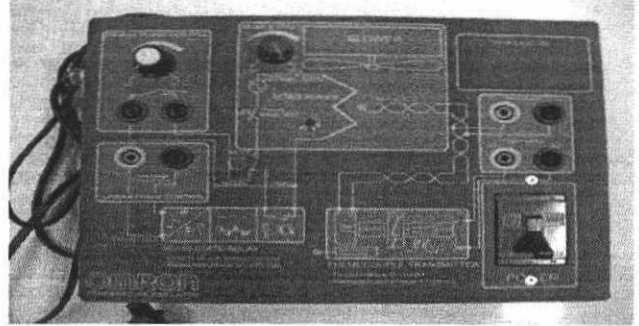


Fig 2 Temperature Simulation.

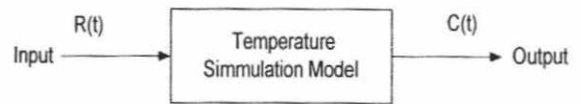


Figure 3 shows the testing to find the Open Loop Transfer Function.

From Fig 3 we use of the open loop result to find out the characteristics equation process. Mat lab application is used to figure the value of the transfer function, by using simulate mat lab as shown in Fig 4.1(a) and 4.1(b). Simulink program is used to determine the characteristics equation process in fig 4.1(a). In this Fig 4.1(b) is the characteristics equation by comparing between the open loop process and simulink program. As a result, the characteristics of the both curves have very similar trends and directions in fig 4.1(b).

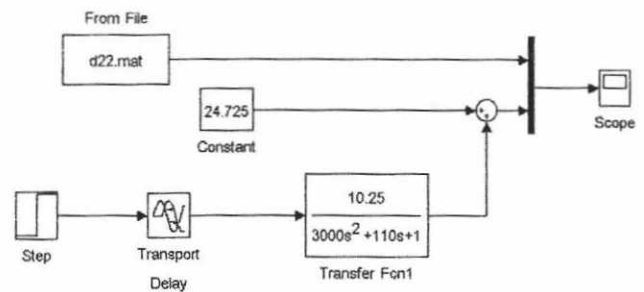


Figure 4.1(a) procedure simulate transfer function.

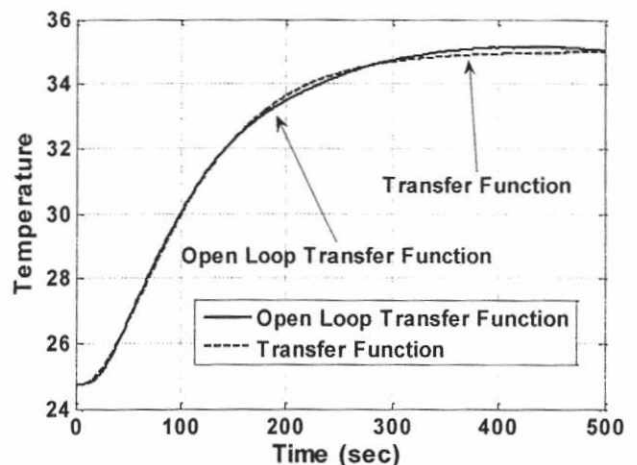


Fig 4.1(b) result simulate transfer function.

The result that shown fig 4.1(b) is the transfer function of system from equation 7.

$$G(S) = \frac{10.25}{3000s^2 + 100s + 110} \quad (7)$$

Result procedure m-file : The transfer function of procedure can be simulated by parameter PID. The pole is shown on z-plane in the root locus theory. The Equation 7 is used to initial configuration of m-file by giving the percent overshoot equal to 0.01 and settling time equal to 320 seconds (the three parameters can be changed) in this paper.

```
close all
clear
clc
sysc = tf(10.25,[3000 110 1]); //transfer function//
OS = 0.1; //percent overshoot//
ts_old = 300; //settling time//
```

Example Source Code initial condition by M-file.

This m-file default result can make the pole to place on root locus theory as shown in fig 5.

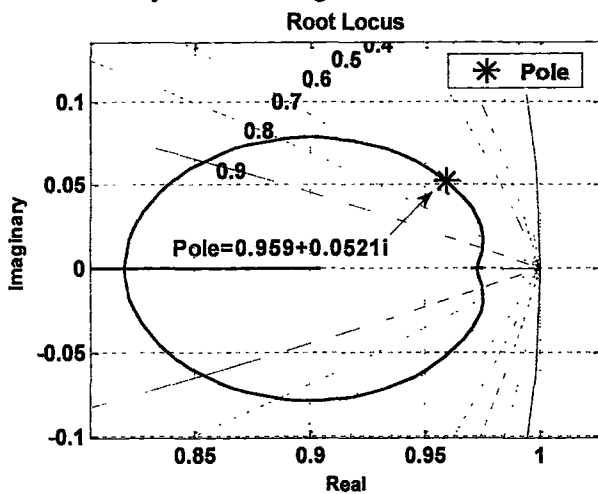


Fig 5. Pole position (0.959+0.0521i) on root locus.

The results of pole (pole position=0.959+0.0521i) on root locus can find out parameters of PID control. The math lab work space presents the parameters $K_p=0.6367$, $K_i=0.0093$, $K_d=5.1969$.

Result Simulink Math lab: In Figure 1, the m-file program will send parameters K_p , K_i and K_d to simulink math lab by increasing the transfer function in equations 7 as shown Fig 6.

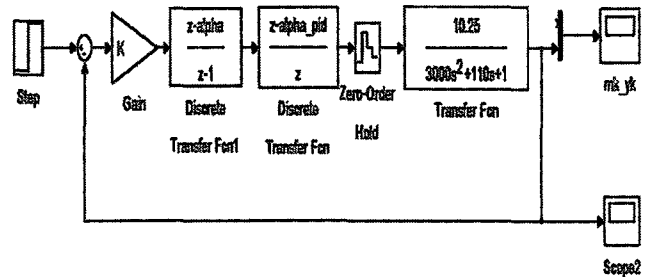


Fig 6. Process PID control Simulink.

In Fig 6, the simulated response of process by using M-file initial value program as follow, set point =1, overshoot=1%, settling time= 300 sec. As a result, we will receive the response of the procedure as shown in Fig 7(a), (b).

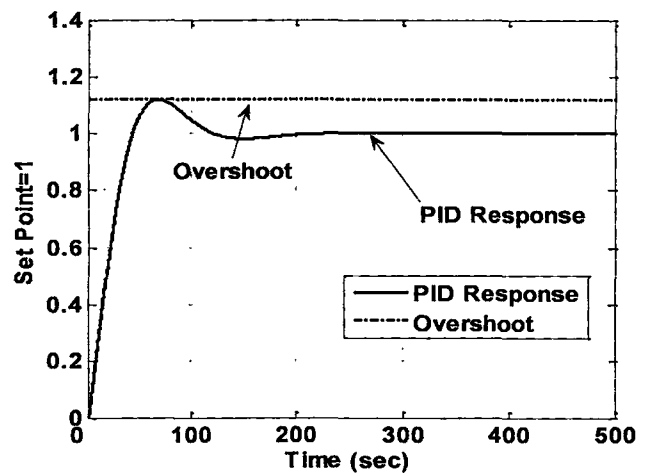


Fig. 7(a).Processes responds PID Set Point = 1.

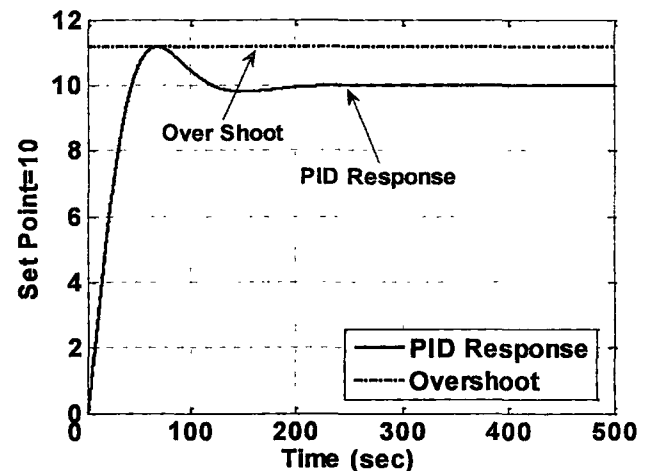


Fig. 7(b).Processes responds PID Set Point = 10.

IV. CONCLUSION

The results that obtained from experiment, the m-file Math lab program is another method that can be used to study the response of PID control. The root locus theory is necessary to observe the pole position of in the z-plane. Moreover, the pole value is needed to find out the parameters PID control.

The simulink math lab can be used to simulate the response weather it follows the initial condition. We can adjust transfer function, percent overshoot and settling time to study and observe procedure responding.

The m-file is also another technique to determine the parameters K_p , K_i , and K_d . As a result, based on the initial conditions specified in the m-file, the response of the process is shown in Fig 7.

Consequently, the users can used the m-file program to determine the response system. By configuring percent overshoot, settling time and the transfer function, we can figure the parameters that can be used to control actions.

REFERENCES

- [1] W R Evans, "Graphical analysis of control system," Trans AIEE, vol 67, pp 547-551, 1948
- [2] B C Kuo, Automatic Control Systems Englewood Cliffs, NJ Prentice-Hall, 1991
- [3] J J D' Azzo and C H Houpis, Linear Control System Analysis and Design Conventional and Modern New York McGraw-Hill, 1995
- [4] D C Hanselman and B C Littlefield, Mastering MATLAB5 A Comprehensive Tutorial and Reference Englewood Chffs, NJ Prentice-Hall, 1998
- [5] D P Prendergast and A M Eydgahi, "EDCON An educational control systems analysis and design program."IEEE Trans, Educ , vol 36, pp 42-44, Feb 1993
- [6] T Cavicchi "Phase-root locus and relative stability," IEEE Contr Syst Mag , vol 16, no 44, pp 69-77, Aug 1996
- [7] L H Keel and S P Bhattacharyya, "On root locus traps," in Proc Amer Conf , San Diego, CA, June 1999, pp 4310-4315
- [8] Richard T O' Brien, Jr , and John M Watkins, " A Unifield Procedure for Discrete - Time Root Locus and Bode Design," 2005 American Control Conference June 8-10, 2005 Portland, OR, USA
- [9] Ali M Eydgahi, Mohammad Ghavamzadeh, " Complementary Root Locus Revisited," IEEE TRANSACTION ON EDUCATION VOL 44, NO 2, MAY 2001, Page(s) 137-143

ประวัติผู้เขียน

ชื่อ - นามสกุล	นายอัครวัฒน์ ไกรสุนทรเลิศภพ
วัน เดือน ปี เกิด	1 ตุลาคม 2523
ที่อยู่	108 หมู่ 10 ตำบลทับทิม อำเภอกำแพงแสน จังหวัดนครปฐม 18260
ประวัติการศึกษา	2548 วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมระบบวัดและควบคุม มหาวิทยาลัยเทคโนโลยีมหานคร
ประสบการณ์การทำงาน	
2549- 2551	ตำแหน่งวิศวกรโครงการ ห้างหุ้นส่วนจำกัด ดี.บี.อิเล็กทรอนิกส์
2551	ตำแหน่งวิศวกรออกแบบ ห้างหุ้นส่วนจำกัด ยูต้า
2552- 2554	ตำแหน่งวิศวกรโครงการ บริษัท กรีนเนอร์รี่ลิฟวิ่ง จำกัด
2554- ปัจจุบัน	ธุรกิจส่วนตัว