

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การออกแบบตัวควบคุมพีไอดีแบบโมเดลอ้างอิง  
สำหรับกระบวนการพัสดมและแผ่นเพลท

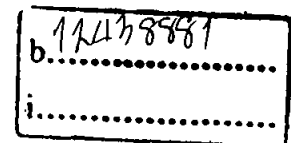
DESIGN MRAC PID CONTROL FOR FAN AND PLATE PROCESS



T123057

พระนาย กังวารรัตน์  
PRANAI KUNGWALRUT

อพ.  
พ3417  
2005  
เลขหมู่.....  
เลขทะเบียน.....123057  
วัน,เดือน,ปี.....18 ต.ค. 2555



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2555

KMITL 2012-EN-M-060-003

# DESIGN MRAC PID CONTROL FOR FAN AND PROCESS

PRANAI KUNGWALRUT

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-060-003

COPYRIGHT 2012

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การออกแบบตัวควบคุมพีไอดีแบบโมเดลอ้างอิงสำหรับกระบวนการพัดลมและแผ่นเพลท  
Thesis Title Design MRAC PID Control for Fan and Plate Process  
นักศึกษา นายพระนาย กังวาลรัตน์  
รหัสประจำตัว 52611122  
ปริญญา วิศวกรรมศาสตรมหาบัณฑิต  
สาขาวิชา วิศวกรรมการวัดคุม  
อาจารย์ที่ปรึกษาวิทยานิพนธ์ รศ.อาจินต์ น่วมสำราญ  
หมายเลขวิทยานิพนธ์ KMITL-2012-EN-M-060-003

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
รศ.ดร.วิทยา	ทิพย์สุวรรณพร	
รศ.ดร.ทวีพล	ชี้อิสต์ย์	
ดร.ยุทธพงศ์	ทัฬผดุง	
ผศ.ดร.พิสิฐ	บุญศรีเมือง	
รศ.อาจินต์	น่วมสำราญ	

วัน / เดือน / ปี ที่สอบ วันพฤหัสบดีที่ 29 ธันวาคม พ.ศ. 2554 เวลา 10.00-12.00 น.  
สถานที่สอบ ณ อาคาร A ชั้น 5 ห้องประชุม 3

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

คณะวิศวกรรมศาสตร์ รับรองแล้ว



(รองศาสตราจารย์ ดร.สุชชีวีร์ สุวรรณสวัสดิ์)

คณบดี คณะวิศวกรรมศาสตร์

วันที่ 29 ธันวาคม พ.ศ. 2554

หัวข้อวิทยานิพนธ์	การออกแบบตัวควบคุมพีไอดีแบบโมเดลอ้างอิงสำหรับกระบวนการ พัฒนและแผ่นเพลท
นักศึกษา	นายพระนาย กังวารรัตน์
รหัสนักศึกษา	52611122
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมการวัดคุม
พ.ศ.	2555
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รศ.อาจินต์ น่วมสำราญ

### บทคัดย่อ

วิทยานิพนธ์นี้นำเสนอวิธีการออกแบบตัวควบคุมพีไอดีด้วยเทคนิค MRAC สำหรับกระบวนการพัฒนและแผ่นเพลท ซึ่งกระบวนการพัฒนและแผ่นเพลทถือเป็นอีกหนึ่งโจทย์ปัญหาที่ทำนายสำหรับการออกแบบระบบควบคุมเนื่องจากเป็นกระบวนการที่มีคุณลักษณะไม่เป็นเชิงเส้นและมีความไวต่อสิ่งรบกวนสูงทำให้การปรับค่าพารามิเตอร์ของตัวควบคุมพีไอดีด้วยวิธีทางเดินรากลหรือด้วยวิธีการลองผิดลองถูกเพื่อให้ได้ผลตอบสนองตามต้องการนั้นทำได้ยาก ดังนั้นเพื่อแก้ปัญหาดังกล่าวจึงนำเสนอวิธีการออกแบบตัวควบคุมพีไอดีแบบปรับค่าได้อัตโนมัติโดยใช้เทคนิคการควบคุมแบบปรับตัวเองตามโมเดลอ้างอิง พีไอดีพารามิเตอร์จะถูกคำนวณโดยใช้หลักการอ้างอิงตามโมเดลอ้างอิงซึ่งเป็นตัวกำหนดคุณสมบัติของระบบควบคุมป้อนกลับ โดยพีไอดีพารามิเตอร์จะถูกปรับเปลี่ยนค่าโดยอัตโนมัติตามการเปลี่ยนแปลงของกระบวนการและสิ่งรบกวน จากผลการทดลองแสดงให้เห็นว่าเทคนิควิธีที่นำเสนอนี้สามารถควบคุมผลตอบสนองของกระบวนการพัฒนและแผ่นเพลทได้อย่างมีประสิทธิภาพ อีกทั้งยังสามารถปรับแต่งพารามิเตอร์ของตัวควบคุมอย่างอัตโนมัติเพื่อให้ได้ผลตอบสนองตามต้องการแม้อยู่ในสภาวะที่มีสิ่งรบกวน

Thesis Title	DESIGN MRAC PID CONTROL FOR FAN AND PLATE PROCESS
Student	Mr. Pranai Kungwalrut
Student ID.	52611122
Degree	Master of Engineering
Program	Instrumentation Engineering
Year	2012
Thesis Advisor	Assoc. Prof. Arjin Numsomran

### **ABSTRACT**

This thesis presents the method to achieve the design MRAC PID control for fan and plate process. The fan and plate process is another challenged problem for control system design because it has non-linearity characteristic and highly sensitivity for ambient disturbance. Thus tuning of PID parameter performs hardly by root locus or trial & error method. The proposed method presents design methodology of auto-adjustable PID controller using MRAC technique for solving the problems. The adjustable PID parameters corresponding to changes in plant and disturbance will be determined by referring to the reference model specifying the properties of desired control system. The experimental results review us that the proposed method can satisfy the required output response of fan and plate system efficiently. Furthermore, it can automatically adjust PID parameters for desired performance following the reference model even in disturbance situation.

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้อย่างดี ด้วยคำแนะนำ คำปรึกษา และอนุเคราะห์อุปกรณ์เครื่องมือในการทำงานวิจัยจาก รศ.อาจินต์ น่วมสำราญ ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์ ด้วยดีเสมอมา ข้าพเจ้ารู้สึกทราบบ้างในความอนุเคราะห์จากท่าน และขอขอบพระคุณเป็นอย่างสูง

ขอกราบพระคุณคณาจารย์ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกๆ ท่านที่ได้ประสิทธิ์ประสาทวิชาให้กับข้าพเจ้า

ขอขอบคุณพี่ๆ เพื่อนๆ และน้องๆ ภาควิชาวิศวกรรมการวัดคุมทุกคนที่เป็นกำลังใจ

ขอขอบคุณบัณฑิตศึกษา คณะวิศวกรรมศาสตร์ ที่ให้ความช่วยเหลือในเรื่องต่างๆ เกี่ยวกับการศึกษาตลอดมา

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นกำลังใจ และให้การสนับสนุนในทุกๆ ด้านทำให้ข้าพเจ้าสามารถทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี

คุณค่าและประโยชน์อันพึงมาจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอบแต่ผู้มีพระคุณทุกท่าน

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	1
1.3 สมมติฐานของการศึกษา.....	2
1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย.....	2
1.5 การเปรียบเทียบระหว่างวิธีการที่นำเสนอกับวิธีการแบบพื้นฐาน.....	5
1.6 ขอบเขตการวิจัย.....	5
1.7 ขั้นตอนการศึกษา.....	6
บทที่ 2 ทฤษฎีที่เกี่ยวข้องกับงานวิจัย.....	7
2.1 กล่าวนำ.....	7
2.2 ระบบควบคุมแบบปรับตัวเองได้ (Adaptive Control System).....	7
2.3 ทฤษฎีของ Model Reference Adaptive Control.....	9
2.3.1 องค์ประกอบของ MRAC.....	12
2.3.2 เทคนิคการออกแบบตัวควบคุมพีไอดีด้วย MRAC.....	14
2.4 ทฤษฎีเทคนิคทางเดินของราก (Root Locus Technique).....	15
2.4.1 เงื่อนไขเบื้องต้นของ ROOT LUCI.....	16
2.4.2 การสร้าง ROOT LOCI ที่สมบูรณ์.....	20
2.5 บทสรุป.....	25
บทที่ 3 การประยุกต์ใช้งานฮาร์ดแวร์และซอฟต์แวร์.....	26
3.1 กล่าวนำ.....	26
3.2 องค์ประกอบของกระบวนการพัลลวมและแผ่นเพลท.....	26
3.2.1 อุปกรณ์ตรวจวัด (Potentiometer).....	27
3.2.2 Motor Drive.....	27
3.2.3 Microcontroller ARM7.....	28

## สารบัญ (ต่อ)

	หน้า
3.2.4 มอเตอร์และพัลลัม.....	30
3.2.5 การ์ดแปลงสัญญาณ PCI-1711 .....	31
3.2.6 เพาเวอร์ซัพพลาย.....	34
3.3 องค์ประกอบของซอฟต์แวร์.....	34
3.3.1 โปรแกรม Matlab .....	34
3.3.2 โปรแกรม Simulink .....	34
3.3.3 การติดตั้งโปรแกรมเพื่อใช้งาน Real Time Windows Target .....	45
3.3.4 โปรแกรม Keil uVision 3.11.....	47
3.4 บทสรุป.....	58
บทที่ 4 การออกแบบตัวควบคุม.....	59
4.1 กล่าวนำ.....	59
4.2 การหาสมการคุณลักษณะของกระบวนการ.....	59
4.3 การออกแบบตัวควบคุม PID ด้วยเทคนิคทางเดินของราก.....	61
4.4 การออกแบบตัวควบคุม MRAC-PID .....	63
4.5 บทสรุป.....	64
บทที่ 5 การจำลองและการทดลอง .....	65
5.1 กล่าวนำ.....	65
5.2 การจำลองกระบวนการโดยใช้ตัวควบคุมแบบพีไอดี ที่ออกแบบด้วยเทคนิคทางเดินราก.....	65
5.3 การจำลองกระบวนการโดยใช้ตัวควบคุม MRAC-PID.....	67
5.4 การทดลองกระบวนการเมื่อใช้ตัวควบคุมแบบ PID.....	68
5.5 การทดลองกระบวนการเมื่อใช้ตัวควบคุมแบบ MRAC-PID.....	71
5.6 เปรียบเทียบผลการทดลองระหว่างตัวควบคุม PID กับตัวควบคุม MRAC-PID.....	73
5.7 บทสรุป.....	75
บทที่ 6 สรุปผลการทดลองและข้อเสนอแนะ .....	76
6.1 สรุปผลการทดลอง.....	76
6.2 ปัญหาที่พบในการทดลองและแนวทางการแก้ปัญหา.....	77
6.3 ข้อเสนอแนะและแนวทางการพัฒนา.....	78

## สารบัญ (ต่อ)

	หน้า
บรรณานุกรม .....	79
ภาคผนวก .....	82
ภาคผนวก ก. ....	83
ภาคผนวก ข. ....	86
ประวัติผู้เขียน .....	93

# สารบัญตาราง

ตารางที่	หน้า
5.1 เปรียบเทียบสมรรถนะของตัวควบคุมแบบ PID กับตัวควบคุมแบบ MRAC-PID .....	75

# สารบัญรูป

รูปที่	หน้า
1.1 โครงสร้างระบบปรับตัวเองโดยอ้อม.....	3
1.2 โครงสร้างระบบปรับตัวเองโดยตรง.....	4
1.3 บล็อกไดอะแกรม MRAC .....	4
2.1 อัลกอริทึมในการปรับตัวเอง .....	8
2.2 โครงสร้างทั่วไปของ MRAC .....	10
2.3 โครงสร้างระบบปรับตัวเองโดยตรง.....	10
2.4 บล็อกไดอะแกรม MRAC .....	11
2.5 โครงสร้างของระบบควบคุม.....	14
2.6 pole, zero ของ $G(s) = \frac{[K(s + z_1)]}{[s(s + p_2)(s + p_3)]}$ .....	19
2.7 ตัวอย่างจุด breakaway บนแกน real ในระนาบ S .....	25
3.1 กระบวนการพัฒนาและแผ่นเฟลท .....	26
3.2 อุปกรณ์ตรวจวัด (Potentiometer).....	27
3.3 ชุดไดร์ฟขับเคลื่อนมอเตอร์ .....	28
3.4 ไมโครคอนโทรลเลอร์ ARM7 .....	28
3.5 สถาปัตยกรรมภายในของไมโครคอนโทรลเลอร์ ARM7 เบอร์ ADuC7024.....	29
3.6 บอร์ด ET-BASE ARM7024(ADUc7024).....	29
3.7 แสดงมอเตอร์และใบพัด.....	31
3.8 คอนเน็กเตอร์สำหรับการ์ด PCI-1711 .....	31
3.9 เพาเวอร์ซัพพลาย .....	34
3.10 ลักษณะของโปรแกรม Simulink.....	36
3.11 แสดงบล็อกต่างๆ ของแหล่งกำเนิดสัญญาณ (Sources) และแหล่งสัญญาณ (Sinks) .....	37
3.12 แสดงบล็อกต่างๆ ภายในกลุ่มบล็อกของระบบต่อเนื่อง (Continuous System) และระบบไม่ต่อเนื่อง (Discrete System) .....	38
3.13 แสดงบล็อกต่างๆ ภายในกลุ่มบล็อกการดำเนินการทางคณิตศาสตร์ (Math Operations) .....	38
3.14 แสดงกลุ่มบล็อก Discontinuities และกลุ่ม Signal Routing .....	39
3.15 แสดงหน้าต่างที่ใช้ในการกำหนดค่าคงที่ในการจำลองปัญหา (Configuration Parameters) ..	40
3.16 หน้าต่างสำหรับสร้างแบบจำลอง.....	41
3.17 การเลือกบล็อกที่ต้องการจาก Simulink Library Browser.....	42
3.18 บล็อกของระบบตัวอย่าง.....	42

## สารบัญรูป (ต่อ)

รูปที่	หน้า
3.19 การเชื่อมต่อสัญญาณระหว่างบล็อก.....	43
3.20 การเปลี่ยนแปลงค่าพารามิเตอร์ของบล็อก.....	43
3.21 ได้อะล็อกใส่ค่าพารามิเตอร์ของบล็อก Transfer Fcn.....	44
3.22 แบบจำลองระบบทางกลในรูปแบบของ Transfer Function .....	44
3.23 Hardware Implementation.....	47
3.24 RTW System Target File .....	47
3.25 หน้าต่าง Setup Real View Microcontroller Development Kit Version 3.11.....	48
3.26 หน้าต่าง customer Information .....	48
3.27 หน้าต่างแสดงการสิ้นสุดการติดตั้งโปรแกรม .....	49
3.28 หน้าต่างหลักของโปรแกรม Keil uVision .....	49
3.29 หน้าต่าง Components, Environment and Books.....	50
3.30 หน้าต่าง Create New Project.....	51
3.31 หน้าต่าง Select Device for Target .....	51
3.32 หน้าต่าง Add File to Group .....	52
3.33 หน้าต่างกำหนด Options for Target.....	53
3.34 หน้าต่างกำหนด Output Information.....	53
3.35 การ compile และการ Build โปรเจ็ค.....	54
3.36 ปุ่ม Build Target .....	55
3.37 การเข้าสู่โหมด Simulation หรือ Debug .....	55
3.38 แสดงสถานะเตรียมพร้อม Simulator.....	56
3.39 ปุ่มการ Debug แบบต่างๆ .....	56
3.40 หน้าต่างการใช้งานของโปรแกรม ARM.....	57
3.41 การ Configuration โปรแกรม ARMWSD ก่อนใช้งาน .....	57
4.1 ผลตอบสนองทางเอาต์พุตของระบบเปิดของกระบวนการ.....	60
4.2 เปรียบเทียบผลตอบสนองทางเอาต์พุตของระบบเปิดระหว่างกระบวนการจริงกับแบบจำลอง.....	61
4.3 การออกแบบตัวควบคุมด้วยวิธีทางเดินราก.....	62
4.4 แสดงผลการจำลองการเปรียบเทียบระหว่าง $y_p$ กับ $y_m$ .....	63
5.1 โปรแกรมสำหรับจำลองกระบวนการโดยใช้ตัวควบคุม PID.....	65
5.2 ผลตอบสนองทางเอาต์พุตของการจำลองเมื่อใช้ตัวควบคุม PID.....	66

## สารบัญรูป (ต่อ)

รูปที่	หน้า
5.3 สัญญาณควบคุมการจำลองกระบวนการเมื่อใช้ตัวควบคุมแบบ PID.....	66
5.4 โปรแกรมสำหรับจำลองกระบวนการโดยใช้ตัวควบคุม MRAC-PID.....	67
5.5 ผลตอบสนองทางเอาต์พุตของการจำลองเมื่อใช้ตัวควบคุม MRAC-PID.....	67
5.6 สัญญาณควบคุมการจำลองกระบวนการเมื่อใช้ตัวควบคุมแบบ MRAC-PID .....	68
5.7 โปรแกรมที่ใช้สำหรับทดลองควบคุมกระบวนการพัลลัมและแผ่นเพลท.....	68
5.8 ผลตอบสนองทางเอาต์พุตของกระบวนการเมื่อใช้ตัวควบคุม PID .....	69
5.9 สัญญาณควบคุมกระบวนการเมื่อใช้ตัวควบคุม PID .....	69
5.10 ผลตอบสนองทางเอาต์พุตของกระบวนการเมื่อใช้ตัวควบคุมแบบ PID ที่มี Disturbance.....	70
5.11 สัญญาณควบคุมกระบวนการเมื่อใช้ตัวควบคุมแบบ PID ที่มี Disturbance.....	70
5.12 โปรแกรมควบคุมกระบวนการเมื่อใช้ตัวควบคุมแบบ MRAC-PID .....	71
5.13 ผลตอบสนองทางเอาต์พุตของกระบวนการเมื่อใช้ตัวควบคุมแบบ MRAC-PID.....	71
5.14 สัญญาณควบคุมกระบวนการเมื่อใช้ตัวควบคุมแบบ MRAC-PID .....	72
5.15 ผลตอบสนองทางเอาต์พุตของกระบวนการเมื่อใช้ตัวควบคุมแบบ MRAC-PID ที่มี Disturbance.....	72
5.16 สัญญาณควบคุมกระบวนการเมื่อใช้ตัวควบคุมแบบ MRAC-PID ที่มี Disturbance.....	73
5.17 เปรียบเทียบผลตอบสนองเอาต์พุตของตัวควบคุม PID กับ MRAC-PID .....	74
5.18 เปรียบเทียบสัญญาณควบคุมของตัวควบคุม PID กับ MRAC-PID .....	74

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

การออกแบบตัวควบคุมสำหรับควบคุมกระบวนการ นั้นเราจะต้องรู้สมการคุณลักษณะหรือโมเดลของกระบวนการจึงจะสามารถทำการออกแบบตัวควบคุมได้ ซึ่งก็จะมีวิธีการหรือทฤษฎี ในการหาสมการคุณลักษณะอยู่หลายวิธีด้วยกัน สามารถเลือกใช้ได้ตามความเหมาะสมกับกระบวนการ แต่อย่างไรก็ตามสมการคุณลักษณะของกระบวนการที่ได้ เป็นเพียงการประมาณค่าเท่านั้น สมรรถนะของตัวควบคุมจะดีหรือไม่นั้นก็ขึ้นอยู่กับสมการคุณลักษณะที่เราประมาณค่าว่าใกล้เคียงกับกระบวนการจริงมากเพียงใด ซึ่งเราจะทราบก็ต่อเมื่อเรานำตัวควบคุมที่ออกแบบได้ไปทดลองควบคุมกระบวนการจริง แต่เนื่องจากการประมาณนั้นบางครั้งเราก็ไม่สนใจพารามิเตอร์บางตัวหรืออาจจะไม่ได้คำนึงถึงตัวแปรบางตัว ซึ่งโอกาสที่จะเกิดเหตุการณ์นี้ขึ้นมีสูง เมื่อเรานำเอาตัวควบคุมไปควบคุมกระบวนการจริง ในบางครั้งสมรรถนะของตัวควบคุมก็ไม่เป็นไปตามการจำลอง หรือบางครั้งอาจจะไม่สามารถนำตัวควบคุมที่ออกแบบไว้ไปควบคุมกระบวนการจริงได้

จากปัญหาดังกล่าวผู้วิจัยจึงได้คิดวิธีการเพื่อลดปัญหาความคลาดเคลื่อนระหว่างสมการคุณลักษณะกับกระบวนการจริง ซึ่งวิธีการโมเดลอ้างอิงแบบปรับตัวเองได้ (Model Reference Adaptive Control: MRAC) เป็นวิธีการที่ใช้แก้ปัญหาดังกล่าวได้ เนื่องจากวิธีการดังกล่าวสามารถที่จะปรับค่าพารามิเตอร์ของตัวควบคุมให้เหมาะสมกับกระบวนการจริง เพราะฉะนั้นถ้าหากเราหาค่าสมการคุณลักษณะได้ไม่ตรงกับค่าของกระบวนการจริง วิธีการดังกล่าวก็จะสามารถปรับตัวเองเพื่อให้ได้ค่าพารามิเตอร์ของตัวควบคุมที่เหมาะสมกับกระบวนการจริงได้

### 1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1. เพื่อศึกษากระบวนการพัลสและแผ่นเพลท
2. เพื่อศึกษาตัวควบคุมพีไอดีด้วยเทคนิคของโมเดลอ้างอิง
3. เพื่อหาค่าพารามิเตอร์ของตัวควบคุมและตำแหน่งการควบคุมที่เหมาะสมสำหรับควบคุมกระบวนการพัลสและแผ่นเพลท
4. เพื่อทดสอบสมรรถนะ เสถียรภาพ และความคงทนของตัวควบคุม

### 1.3 สมมติฐานของการศึกษา

การหาสมการคุณลักษณะของกระบวนการนั้นมีส่วนสำคัญต่อสมรรถนะของตัวควบคุม กล่าวคือถ้าหาสมการคุณลักษณะของกระบวนการได้ตรงหรือใกล้เคียงกับกระบวนการจริง ก็จะทำให้สามารถนำตัวควบคุมที่ออกแบบไปควบคุมกระบวนการจริงได้อย่างมีประสิทธิภาพ แต่ในความเป็นจริงแล้วการหาค่าสมการคุณลักษณะของกระบวนการนั้น เป็นเพียงการประมาณค่าเท่านั้น จึงมักจะมีความคลาดเคลื่อนจากกระบวนการจริงเสมอ เนื่องจากว่าบางตัวแปรเราก็ไม่คำนึงถึงหรือบางตัวแปรเราก็ไม่สามารถนำมาคิดได้ จึงทำให้เกิดความคลาดเคลื่อนดังกล่าวขึ้น และเมื่อออกแบบตัวควบคุมแล้วนำไปควบคุมกระบวนการจริง จึงทำให้ตัวควบคุมทำงานได้ไม่เต็มประสิทธิภาพ

วิธีการของโมเดลอ้างอิงแบบปรับตัวเองได้ (MRAC) จึงน่าจะเป็นวิธีการที่แก้ปัญหาดังกล่าวได้ดี เนื่องจากวิธีการนี้ ตัวโมเดลอ้างอิงแบบปรับตัวเองได้จะพยายามเรียนรู้กระบวนการจริงและปรับตัวเองให้เท่ากับกระบวนการจริงในขณะเดียวกันก็จะคำนวณค่าพารามิเตอร์ของตัวควบคุมให้เหมาะสมกับกระบวนการจริง ซึ่งจะทำให้ตัวควบคุมทำงานได้อย่างเต็มประสิทธิภาพ

### 1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย

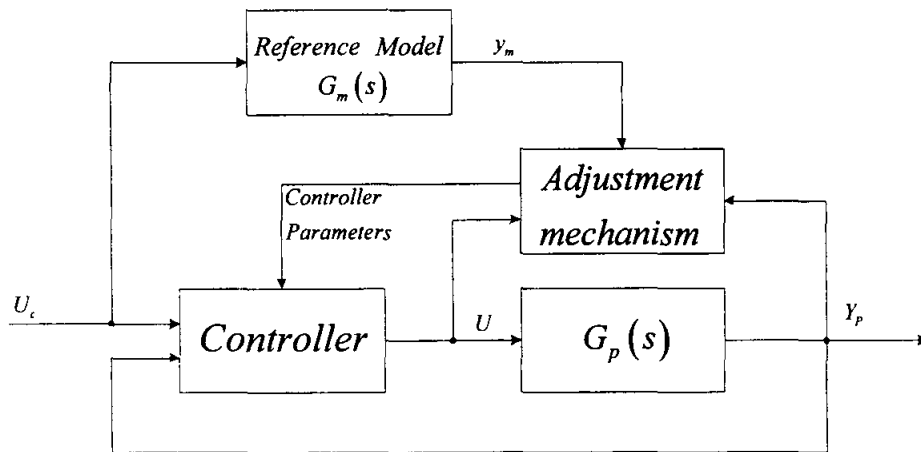
โมเดลอ้างอิงแบบปรับตัวเองได้ (Model Reference Adaptive Control : MRAC) [1] เป็นการประมาณระบบที่ต้องการควบคุมให้อยู่ในรูปของสมการอนุพันธ์เชิงเส้นไม่แปรตามเวลา และออกแบบตัวควบคุมโดยวิธีสเตป้อนกลับ มีคุณสมบัติที่ถูกกำหนดค่าตายตัว หรือ Fixed Controller ซึ่งการควบคุมแบบคงที่ ไม่สามารถควบคุมพฤติกรรมของระบบให้เป็นไปตามความต้องการได้ทุกสถานการณ์โดยเฉพาะอย่างยิ่งถ้ากระบวนการที่ถูกควบคุมมีตัวแปรที่ไม่ทราบค่า หรือ แปรค่าไปตามเวลา

ในช่วงหลังของยุค 1950 ได้มีความสนใจในเรื่องการควบคุมที่ปรับตัวได้มากขึ้น โดยเริ่มจาก gain scheduling technique [1] ซึ่งเป็นการควบคุมที่ปรับพารามิเตอร์ของตัวควบคุมไปตามสภาพแวดล้อมที่สังเกตได้ (เป็นการสังเกตสภาพแวดล้อมจากนั้นเปิดตารางเพื่อหาค่าพารามิเตอร์หรือเกณฑ์ที่เหมาะสม) การออกแบบตัวควบคุมสามารถทำได้ เมื่อเรารู้ความสัมพันธ์ที่แน่นอนระหว่างพฤติกรรมของกระบวนการกับสภาวะภายนอกซึ่งสามารถวัดได้ เช่น ปฏิกริยาของเครื่องบินจากคำสั่งของนักบินมีผลโดยตรงกับระดับความสูงของเครื่องบิน เนื่องจากพฤติกรรมของเครื่องบินเป็นฟังก์ชันที่สัมพันธ์กับความสูง ดังนั้นจึงสามารถออกแบบตัวควบคุมที่สามารถปรับตัวไปตามระดับความสูงต่างๆ

ค่านิยามของระบบควบคุมที่ปรับตัวเองได้ มีอยู่หลายนิยาม ยกตัวอย่างเช่น ระบบที่มีการควบคุมซึ่งสามารถปรับตัวเองให้เหมาะสมตามการเปลี่ยนแปลงของกระบวนการ แต่นิยามที่ใช้กันมากก็คือ ระบบที่ประกอบด้วยการป้อนกลับปฐมภูมิ (Primary Feedback) ซึ่งคอยดูการเปลี่ยนแปลงสัญญาณของกระบวนการ และการป้อนกลับทุติยภูมิ (Secondary Feedback) ซึ่งเกี่ยวข้องกับการเปลี่ยนแปลงพารามิเตอร์ของส่วนควบคุม [2] จากนิยามนี้การป้อนกลับปฐมภูมิเป็นการควบคุมที่ไม่สามารถปรับตัวได้โดยทั่วไป ในขณะที่การป้อนกลับทุติยภูมิจะทำให้เกิดรูปที่สามารถปรับตัวได้ การควบคุมที่ปรับตัวได้ดังที่ได้นิยามข้างต้นต่างจากเทคนิค Gain Scheduling ตรงที่ Gain Scheduling ไม่ได้ใช้การป้อนกลับ

หุติยภูมิเป็นกลไกในการปรับพารามิเตอร์ แต่ใช้กลไกของการวัดสภาพแวดล้อมภายนอก ดังนั้นเทคนิค Gain Scheduling อาจไม่นับว่าเป็นการควบคุมที่ปรับตัวได้ จุดประสงค์ของการเปลี่ยนแปลงพารามิเตอร์ในการป้อนกลับหุติยภูมิคือต้องการรักษาสมรรถภาพ (Performance) ของระบบ ให้สูงแม้ว่าพารามิเตอร์ของกระบวนการอาจไม่ทราบค่าหรือมีค่าที่เปลี่ยนแปลง

การควบคุมแบบปรับตัวเองตามโมเดลอ้างอิง หรือ Model Reference Adaptive Control: MRAC ถูกคิดขึ้นโดย Whitaker [3] เมื่อปี ค.ศ. 1958 โดยการสร้างวงรอบการควบคุมเพิ่มขึ้นมาเพื่อปรับค่าพารามิเตอร์ของตัวควบคุม ทำให้ผลตอบสนองของระบบเปลี่ยนตามโมเดลอ้างอิง โครงสร้างของ MRAC แสดงดังรูปที่ 1.1

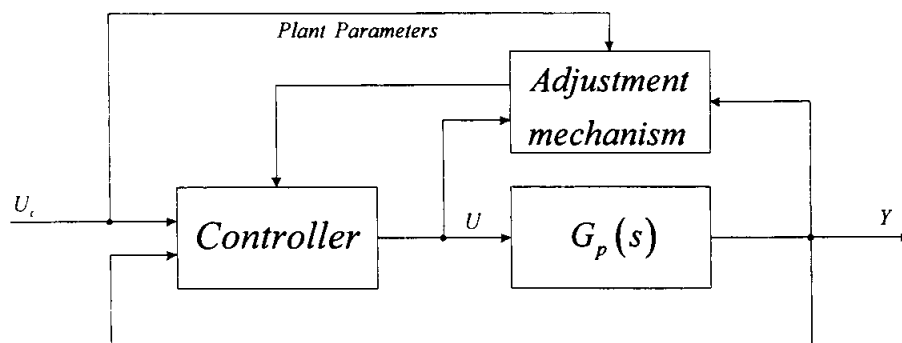


รูปที่ 1.1 โครงสร้างระบบปรับตัวเองโดยอ้อม

จากรูปที่ 1.1 มีการแบ่งวงรอบการควบคุมออกเป็นสองวงรอบ โดยที่วงรอบป้อนกลับเดิมของระบบเรียกว่าวงรอบภายใน และวงรอบการปรับพารามิเตอร์ของตัวควบคุมเรียกว่าวงรอบภายนอก ซึ่งวิธีการหาพารามิเตอร์ของตัวควบคุมคำนวณได้จากค่าความแตกต่างของสัญญาณเอาต์พุตของกระบวนการจริง  $y_p$  กับสัญญาณเอาต์พุตโมเดลอ้างอิง  $y_m$  เพื่อปรับพารามิเตอร์ของตัวควบคุมให้เหมาะสมเพื่อลดค่าความแตกต่างของสัญญาณเอาต์พุตทั้งสองให้เป็นศูนย์ หรือมีค่าน้อยที่สุด สำหรับงานวิจัยนี้จะออกแบบ MRAC ด้วยกฎของ MIT (MIT Rule) [4] ซึ่งเริ่มจากการการคำนวณหาความแตกต่างของสัญญาณเอาต์พุตของกระบวนการจริงกับโมเดลอ้างอิง

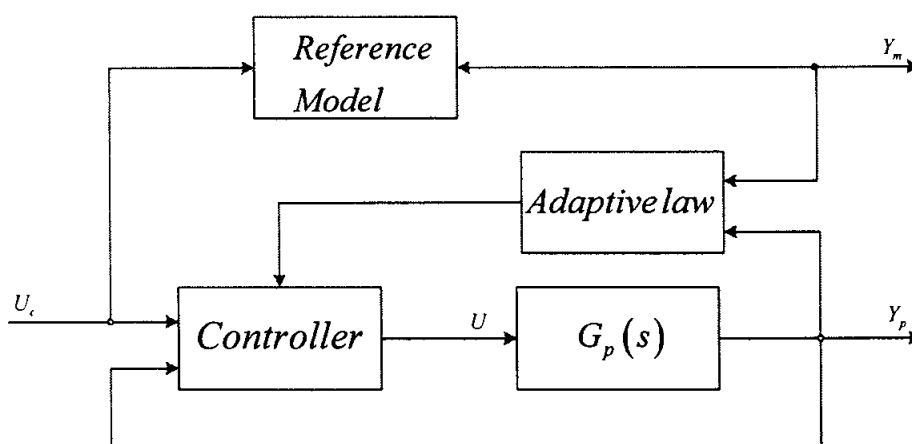
ความสามารถในการปรับตัวเองได้ของระบบปรับตัวทางอ้อมในรูปที่ 1.1 เกิดจากการป้อนกลับหุติยภูมิโดยส่วน Adjustment Mechanism จะประมาณพารามิเตอร์ของกระบวนการและส่งต่อให้กับส่วนออกแบบเพื่อออกแบบพารามิเตอร์ให้กับส่วนควบคุม เมื่อกระบวนการมีการเปลี่ยนแปลงพารามิเตอร์ ส่วนออกแบบตัวควบคุม จะออกแบบพารามิเตอร์ของส่วนควบคุมให้เหมาะสมกับกระบวนการที่เปลี่ยนไป ส่วนควบคุมที่ได้รับการปรับพารามิเตอร์ตามกระบวนการนี้ จะควบคุมพฤติกรรมของกระบวนการด้วยกลไกของวงรอบป้อนกลับต่อไป

โครงสร้างระบบปรับตัวเองโดยตรงมีรูปแบบคล้ายกับระบบปรับตัวเองโดยอ้อมในรูปที่ 1.1 แต่ตัดส่วนออกแบบทิ้งและเปลี่ยนส่วนประมาณเป็นส่วนปรับพารามิเตอร์ของส่วนควบคุมดังในรูปที่ 1.2



รูปที่ 1.2 โครงสร้างระบบปรับตัวเองโดยตรง

โครงสร้างของการปรับตัวโดยตรงและโดยอ้อม มีทั้งข้อดีและข้อเสีย ข้อดีแบบทางอ้อมคือความอิสระที่จะเลือกรูปแบบของส่วนออกแบบ ข้อดีที่สองของแบบทางอ้อมคือ การมีส่วนประมาณพารามิเตอร์ จะทำให้ผู้ปฏิบัติงานได้รู้ถึงสถานะปัจจุบันของกระบวนการ และข้อผิดพลาดต่างๆ ของกระบวนการ อย่างไรก็ตามการปรับตัวเองโดยอ้อมไม่สะดวกต่อการวิเคราะห์ทางทฤษฎี เพราะวาระบบจะซับซ้อน โครงสร้างการปรับตัวเองโดยตรงมักจะซับซ้อนน้อยกว่า ดังนั้นเราสามารถพิสูจน์เสถียรภาพของระบบโดยรวมได้ง่ายกว่า



รูปที่ 1.3 บล็อกไดอะแกรม MRAC

ในรูปที่ 1.3 แสดงให้เห็นว่าส่วนควบคุมปฐมภูมิจะทำหน้าที่ควบคุมพฤติกรรมของระบบควบคุมวงรอบปิดในลักษณะเดียวกันกับระบบที่ปรับตัวเองโดยทั่วไป แต่อย่างไรก็ตามในสถานการณ์ที่เราไม่ทราบค่าพารามิเตอร์ของระบบ หรือ พารามิเตอร์ของระบบที่เปลี่ยนแปลงค่าไปตามเวลา การตั้งค่าพารามิเตอร์ของส่วนควบคุมปฐมภูมิให้คงที่นั้น จะทำให้ไม่สามารถควบคุมพฤติกรรมของระบบควบคุมวงรอบปิดให้เป็นไปตามต้องการได้ทุกสถานการณ์

โดยเทคนิคของ MRAC ผลตอบสนองหรือพฤติกรรมของระบบต่ออินพุต  $u_c$  ตามที่เราต้องการสามารถกำหนดโดยรูปแบบอ้างอิง หรือ Reference Model ซึ่งแสดงด้วย  $y_m$  องค์ประกอบในการ

ปรับตัวจะติดตามเปรียบเทียบผลตอบสนองของกระบวนการ  $y_p$  กับ  $y_m$  และ ตั้งค่าพารามิเตอร์ให้กับ ส่วนควบคุมปฏุมภูมิ เพื่อที่จะนำผลตอบสนองของกระบวนการให้เข้าสู่ผลตอบสนองของโมเดลอ้างอิง ในองค์ประกอบของส่วนปรับตัวเอง นอกจากใช้ผลตอบสนองของกระบวนการ  $y_p$  แล้วยังอาจใช้สเตท (State) ของกระบวนการ  $x_p$  และอินพุทของกระบวนการ  $u_c$  ลักษณะของ MRAC ส่วนใหญ่มักจะเป็น การปรับตัวเองโดยตรง เพราะไม่มีส่วนของการประมาณพารามิเตอร์ของกระบวนการอย่างชัดเจน

ในกรณีรูปที่ 1.3 แสดงว่าโมเดลอ้างอิงจะขนานกับกระบวนการ แต่ไม่จำเป็นว่าระบบปรับตัวเอง MRAC จำเป็นต้องมีโมเดลอ้างอิงที่ขนานกับกระบวนการเสมอ ระบบ MRAC บางครั้งจะนำโมเดลอ้างอิงต่ออนุกรมกับกระบวนการ ซึ่งในการต่ออนุกรมนี้ โมเดลอ้างอิงจะทำหน้าที่เป็นตัวกำเนิดอินพุทอ้างอิง และมีผลต่อการควบคุมตรงที่ระบบโดยทั่วไปจะไม่สามารถตอบสนองต่ออินพุทที่เป็นขั้นบันไดได้ ดังนั้นตัวกำเนิดอินพุทอ้างอิงจะกรองสัญญาณอินพุทให้เรียบขึ้นและตอบสนองต่อกระบวนการจะเป็นไปตามอินพุทได้ดีขึ้น ในลักษณะนี้มีผลต่อการอิ่มตัวของระบบด้วย (Saturation) นอกจากโมเดลอ้างอิงจะต่อขนานและต่ออนุกรมกับกระบวนการแล้วยังสามารถนำทั้งสองแบบมารวมกันเป็นแบบอนุกรม-ขนานได้ ซึ่งมีข้อดีข้อเสียต่างกัน ในงานวิจัยนี้จะศึกษาเฉพาะ MRAC แบบที่มีรูปแบบอ้างอิงที่ต่อขนานกับกระบวนการ (Parallel MRAC) ส่วนสำคัญที่สุดใน MRAC ได้แก่ การออกแบบกฎการปรับตัวเอง ซึ่งมีหลายวิธี ซึ่งในงานวิจัยนี้จะศึกษา และทดลองเฉพาะวิธีของ MIT (MIT Row)

## 1.5 การเปรียบเทียบระหว่างวิธีการที่นำเสนอกับวิธีการแบบพื้นฐาน

ในการวิจัยนี้ได้มุ่งเน้นการออกแบบตัวควบคุมแบบ MRAC-PID เพื่อควบคุมกระบวนการพัลลัมและแผ่นเพลท ซึ่งจะทำการเปรียบเทียบกับตัวควบคุม PID ที่ออกแบบด้วยเทคนิคทางเดินราก เพื่อทำการเปรียบเทียบประสิทธิภาพของตัวควบคุมแบบ MRAC-PID กับตัวควบคุมแบบ PID

## 1.6 ขอบเขตการวิจัย

1. ศึกษากระบวนการพัลลัมและแผ่นเพลท
2. ศึกษาชุดขับเคลื่อนมอเตอร์กระแสตรง
3. หาทรานส์เฟอ์ฟังก์ชันของกระบวนการ
4. ออกแบบตัวควบคุม MRAC-PID สำหรับควบคุมกระบวนการพัลลัมและแผ่นเพลท
5. เปรียบเทียบผลตอบสนองทางด้านเอ้าท์พุทของกระบวนการระหว่างการใช้อัตว์ควบคุมแบบ MRAC-PID กับการใช้อัตว์ควบคุม PID ที่ออกแบบด้วยเทคนิคทางเดินราก

## 1.7 ขั้นตอนของการศึกษา

วิทยานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาสำหรับงานวิจัยนี้ออกเป็น 6 บทด้วยกันคือ

บทที่ 1 บทนำ กล่าวถึงความเป็นมาของงานวิจัย ความมุ่งหมายและวัตถุประสงค์ สมมติฐาน ทฤษฎีหรือแนวคิดที่ใช้ในงานวิจัย การเปรียบเทียบระหว่างวิธีการที่นำเสนอกับวิธีการแบบพื้นฐาน ขอบเขตของการวิจัย และขั้นตอนการศึกษา

บทที่ 2 ทฤษฎีที่เกี่ยวข้องกับงานวิจัย กล่าวถึงทฤษฎีทั้งหมดที่เกี่ยวข้องกับงานวิจัย ได้แก่ ทฤษฎีของ Model Reference Adaptive Control และทฤษฎีการออกแบบตัวควบคุมด้วยเทคนิคทางเดินราก

บทที่ 3 การประยุกต์ใช้งานฮาร์ดแวร์และซอฟต์แวร์ จะกล่าวถึงองค์ประกอบกระบวนการทั้งหมดทั้งฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ในงานวิจัยทั้งหมด

บทที่ 4 วิธีการออกแบบตัวควบคุม กล่าวถึงขั้นตอนและวิธีการออกแบบตัวควบคุมแบบ MRAC-PID และการออกแบบตัวควบคุม PID ด้วยวิธีทางเดินราก

บทที่ 5 การจำลองและการทดลอง กล่าวถึงขั้นตอนจำลองกระบวนการและวิธีการทดลองทั้งหมด โดยใช้ข้อมูลจากบทที่ 2, บทที่ 3 และบทที่ 4 มาประกอบการพิจารณาในการจำลอง

บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ สรุปปัญหาที่พบในงานวิจัยพร้อมทั้งเสนอแนะแนวทางแก้ปัญหาและแนวทางการพัฒนา

## บทที่ 2

# ทฤษฎีที่เกี่ยวข้องกับงานวิจัย

### 2.1 กล่าวนำ

ในการทำวิจัยในระดับบัณฑิตศึกษานั้น มีความจำเป็นจะต้องมีทฤษฎีหรือหลักการสำหรับรองรับงานวิจัยที่ทำเพื่ออธิบายถึงที่มาที่ไปของผลการทดลอง ซึ่งในบทนี้จะกล่าวถึงทฤษฎีที่เกี่ยวข้องกับงานวิจัย ซึ่งจะประกอบไปด้วย ทฤษฎีของ Adaptive Control System, ทฤษฎี Model Reference Adaptive Control, ทฤษฎีการออกแบบตัวควบคุมด้วยเทคนิคทางเดินราก, ซึ่งทฤษฎีที่กล่าวถึงจะถูกนำมาใช้ในการวิเคราะห์และออกแบบตัวควบคุมในบทที่ 4 ดังนั้นจึงจำเป็นต้องศึกษาทฤษฎีและหลักการให้เข้าใจเพื่อจะได้นำไปใช้อย่างถูกต้องและรวดเร็ว

### 2.2 ระบบควบคุมแบบปรับตัวเองได้ (Adaptive Control System)

ตัวควบคุมแบบ Adaptive หมายถึงตัวควบคุมที่สามารถปรับตัวเองได้ในการตอบสนองต่อการเปลี่ยนแปลงที่พลศาสตร์ของกระบวนการและคุณสมบัติของสิ่งรบกวน

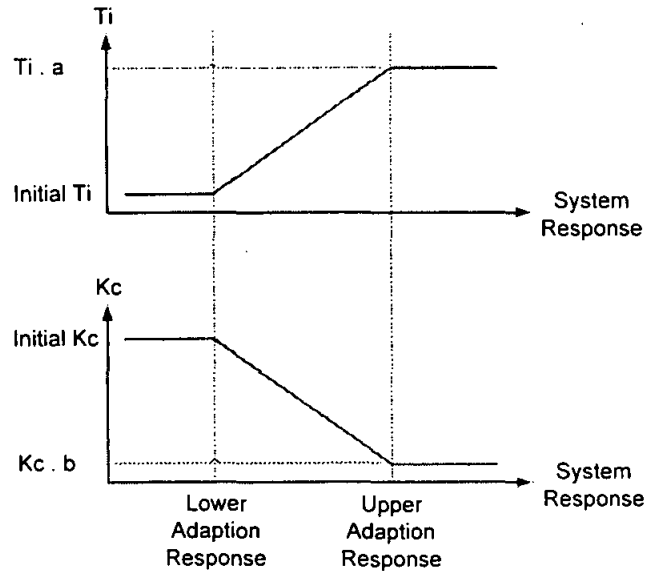
วิธีการของ Ziegler Nichols [6] คือการกำหนดปฏิกิริยาค่าคงที่ให้กับพารามิเตอร์ตัวควบคุมแบบพีไอดีคือ  $K_c$ ,  $T_i$  และ  $T_d$  แต่บางระบบผลตอบสนองไม่สามารถบอกได้จึงทำให้ค่าพารามิเตอร์ของตัวควบคุมแบบพีไอดีเหล่านี้ไม่สามารถทำงานได้อย่างมีประสิทธิภาพ แต่สำหรับตัวควบคุมแบบปรับตัวเองได้ (Adaptive Controller) สามารถช่วยให้ทั้งเสถียรภาพและการตอบสนองของระบบควบคุมมีผลที่ดี การเปลี่ยนแปลงสัมประสิทธิ์และอัลกอริทึมของการควบคุมแบบ Real time เพื่อชดเชยการเปลี่ยนแปลงในระบบด้วยตัวของมันเอง โดยทั่วไปตัวควบคุมจะตรวจสอบการทำงานของระบบเป็นระยะๆ จากทรานส์เฟอร์ฟังก์ชันของระบบและปรับเปลี่ยนอัลกอริทึมการควบคุม โดยมันสามารถเรียนรู้เกี่ยวกับ พฤติกรรมของกระบวนการในขณะที่กำลังทำการควบคุมไปพร้อมๆ กันได้

#### 2.2.1 อัลกอริทึมการปรับตัวเองสำหรับพารามิเตอร์ ( $K_c$ ) และ ( $T_i$ ) ของตัวควบคุมพีไอดี

เกนซ์ ( $K_c$ ) และ ( $T_i$ ) ของตัวควบคุมพีไอดีมีผลโดยตรงต่อผลตอบสนองของระบบ เทอมของ TI มีผลต่อความสัมพันธ์กับความผิดพลาดของระบบที่แตกต่างกันระหว่างระบบอ้างอิงกับเอาต์พุตของระบบ และเกนซ์ ( $K_c$ ) มีผลโดยตรงต่อค่าคงที่เวลา (Time Constant) และค่าพุ่งเกิน (Overshoot) ของระบบ บางครั้งก็ไม่ได้กล่าวถึง ( $T_i$ ) เนื่องจากตัวควบคุม PI ก็เพียงพอแล้วสำหรับระบบในอุตสาหกรรม ซึ่งตรงกันข้าม เกนซ์ ( $K_c$ ) และเทอมของ ( $T_i$ ) ที่ใช้ในการปรับตัวเองของพีไอดี เนื่องจากว่าตัวควบคุมแบบพีไอดีจะกลายเป็นตัวควบคุมแบบ PI-Adaptive Self Tune Based และ D-Self Tune Based (PI-D) [1] อัลกอริทึมของเทอมพีไอกำหนดไว้ดังรูปที่ 2.1 และสมการของพีไอกำหนดไว้ดังสมการ 2.1

$$T_{i\_ADAPT} = \frac{T_{i(S.TUNE)}(a-1)}{U_{ADAPT} - L_{ADAPT}} (R_{RESPONSE} - L_{ADAPT}) + T_{i(S.TUNE)} \quad a > 1 \quad (2.1)$$

$$T_{c\_ADAPT} = \frac{K_{c(S.TUNE)}(b-1)}{U_{ADAPT} - L_{ADAPT}} (R_{RESPONSE} - L_{ADAPT}) + K_{c(S.TUNE)} \quad 1 > b \geq 0$$



รูปที่ 2.1 อัลกอริทึมในการปรับตัวเอง

ในรูปที่ 2.1 ค่าเริ่มต้นของ ( $K_c$ ) และค่าเริ่มต้นของ  $T_i$  เป็นค่าสัมประสิทธิ์ที่พบในวิธีการหาปฏิกริยาของกระบวนการของ Ziegler-Nichols “a” คือสัมประสิทธิ์ของเทอม ( $T_i$ ) เหนือเส้นกราฟ และ “b” คือสัมประสิทธิ์ของเทอม ( $K_c$ ) ได้เส้นกราฟซึ่งสามารถกล่าวได้ว่า a มีค่ามากกว่า 1 และ b มีค่าอยู่ระหว่าง 0 ถึง 1 ผลตอบสนองของระบบแทนด้วยสัญลักษณ์ทางเอาท์พุท (Voltage, Degrees, Speed, Moment etc.) ตัวอย่างเช่น ผลตอบสนองเอาท์พุทของความเร็วมอเตอร์, Lower Adaption Response คือ ค่าที่ต่ำกว่าความเร็วมอเตอร์ (rpm) และ Upper Adaption Response คือค่าที่สูงกว่าความเร็วมอเตอร์ (rpm) ซึ่งอัลกอริทึมของ Adaptive จะทำงานอยู่ในช่วงนี้

จากเส้น lower adaption อัลกอริทึมจะเริ่มทำงานโดยเพิ่มค่า ( $T_i$ ) และลดค่า ( $K_c$ ) ซึ่งเหตุผลในการเพิ่มค่า  $T_i$  เพื่อให้ค่าผิดพลาดมีค่าน้อยที่สุดก่อนเข้าสู่ภาวะคงตัวและเหตุผลในการลดค่า ( $K_c$ ) เพื่อไม่ให้มีค่าพุ่งเกินหรือไม่ให้ผลตอบสนองของระบบมีค่าพุ่งเกินที่สูงเกินไป

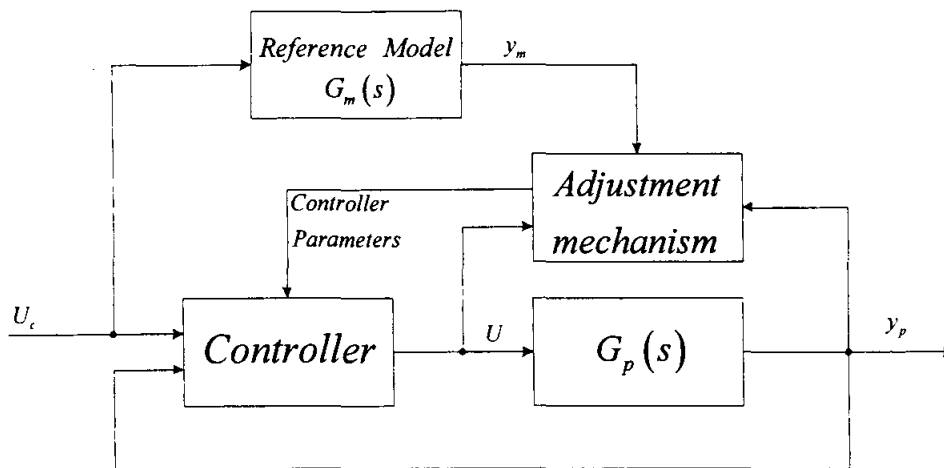
## 2.3 ทฤษฎีของ Model Reference Adaptive Control

การควบคุมที่ปรับตัวได้ ในระหว่างการพัฒนาทฤษฎีการควบคุมแนวใหม่ จะพบว่า การควบคุมที่ถูกสังเคราะห์โดยการประมาณระบบที่ต้องการควบคุมให้อยู่ในรูปของสมการอนุพันธ์เชิงเส้นไม่แปรตามเวลา และออกแบบตัวควบคุมโดยวิธีสเตทฟีดแบ็ก มีคุณสมบัติที่ถูกกำหนดค่าตายตัว (ซึ่งต่อไปจะเรียกตัวควบคุมชนิดนี้ว่า การควบคุมคงที่ หรือ Fixed Controller) การควบคุมคงที่ดังกล่าวไม่สามารถควบคุมพฤติกรรมของระบบให้เป็นไปตามความต้องการ ได้ทุกสถานการณ์ โดยเฉพาะอย่างยิ่งถ้ากระบวนการที่ถูกควบคุมมีตัวแปรที่ไม่ทราบค่า หรือ แปรค่าไปตามเวลา [2]

ในช่วงหลังของยุค 1950 ได้มีความสนใจในเรื่องการควบคุมที่ปรับตัวได้มากขึ้น โดยเริ่มจาก gain scheduling technique [3] ซึ่งเป็นการควบคุมที่ปรับพารามิเตอร์ของตัวควบคุมไปตามสภาพแวดล้อมที่สังเกตได้ (เป็นการสังเกตสภาพแวดล้อมจากนั้นเปิดตารางเพื่อหาค่าพารามิเตอร์หรือเกณฑ์ที่เหมาะสม) การจะออกแบบตัวควบคุมสามารถทำได้เมื่อเรารู้ความสัมพันธ์ที่แน่นอนระหว่างพฤติกรรมของกระบวนการกับสถานะภายนอกซึ่งสามารถวัดได้ เช่น ปฏิกริยาของเครื่องบินจากกำลังของนักบินนี้โดยตรงกับระดับความสูงของเครื่องบิน เนื่องจากพฤติกรรมของเครื่องบินเป็นที่ทราบกันว่าเป็นฟังก์ชันอย่างไรกับความสูง ดังนั้นจึงสามารถออกแบบตัวควบคุมที่สามารถปรับตัวไปตามระดับความสูงต่างๆ

คำนิยามของ ระบบควบคุมที่ปรับตัวได้มีอยู่หลายนิยาม ยกตัวอย่างเช่น ระบบที่มีการควบคุมซึ่งสามารถปรับตัวเองให้เหมาะสมตามการเปลี่ยนแปลงของกระบวนการ แต่นิยามที่ใช้กันมากก็คือ ระบบที่ประกอบด้วยการป้อนกลับปฐมภูมิ (Primary Feedback) ซึ่งคอยดูการเปลี่ยนแปลงสัญญาณของกระบวนการและการป้อนกลับทุติยภูมิ (Secondary Feedback) ซึ่งเกี่ยวข้องกับการเปลี่ยนแปลงพารามิเตอร์ของส่วนควบคุม จากนิยามนี้การป้อนกลับปฐมภูมิเป็นเช่นดังการควบคุมที่ไม่สามารถปรับตัวได้โดยทั่วไป ในขณะที่การป้อนกลับทุติยภูมิจะทำให้เกิดรูปที่สามารถปรับตัวได้ การควบคุมที่ปรับตัวได้ดังที่ได้นิยามข้างต้นต่างจากเทคนิค Gain Scheduling ตรงที่ Gain Scheduling ไม่ได้ใช้การป้อนกลับทุติยภูมิเป็นกลไกในการปรับพารามิเตอร์ แต่ใช้กลไกของการวัดสภาพแวดล้อมภายนอก ดังนั้น เทคนิค Gain Scheduling อาจไม่นับว่าเป็นการควบคุมที่ปรับตัวได้ จุดประสงค์ของการเปลี่ยนแปลงพารามิเตอร์ในการป้อนกลับทุติยภูมิคือต้องการรักษาสมรรถภาพ (Performance) ให้สูงแม้ว่าพารามิเตอร์ของกระบวนการอาจไม่ทราบค่าหรือมีค่าที่เปลี่ยนแปลง

การควบคุมแบบปรับตัวเองตามโมเดลอ้างอิง หรือ Model Reference Adaptive Control: MRAC ถูกคิดขึ้นโดย Whitaker [3] เมื่อปี ค.ศ. 1958 โดยการสร้างวงรอบการควบคุมเพิ่มขึ้นมาเพื่อปรับค่าพารามิเตอร์ของตัวควบคุม ทำให้ผลตอบสนองของระบบเปลี่ยนตามโมเดลอ้างอิง โครงสร้างทั่วไปของ MRAC แสดงดังรูปที่ 2.2

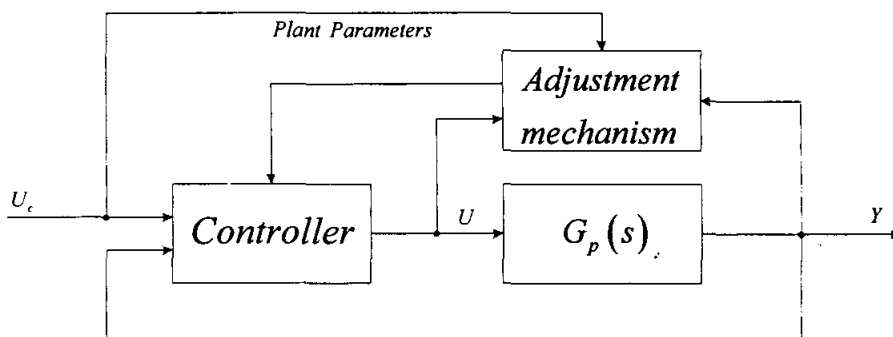


รูปที่ 2.2 โครงสร้างทั่วไปของ MRAC

จากรูปที่ 2.2 มีการแบ่งวงรอบควบคุมออกเป็นสองวงรอบ โดยที่วงรอบป้อนกลับเดิมของระบบเรียกว่าวงรอบภายใน และวงรอบการปรับพารามิเตอร์ตัวควบคุมเรียกว่าวงรอบภายนอก การหาค่าพารามิเตอร์ของตัวควบคุมคำนวณได้จากค่าความแตกต่างของสัญญาณเอาต์พุตของกระบวนการจริง  $y_p$  กับสัญญาณเอาต์พุตโมเดลอ้างอิง  $y_m$  เพื่อปรับพารามิเตอร์ของตัวควบคุมให้เหมาะสมเพื่อลดค่าความแตกต่างของสัญญาณเอาต์พุตทั้งสองให้น้อยลงหรือเป็นศูนย์ สำหรับงานวิจัยนี้จะออกแบบ MRAC ด้วยกฎของ MIT (MIT Rule) [4] ซึ่งเริ่มจากการการคำนวณหาความแตกต่างของสัญญาณเอาต์พุตของกระบวนการจริงกับโมเดลอ้างอิง

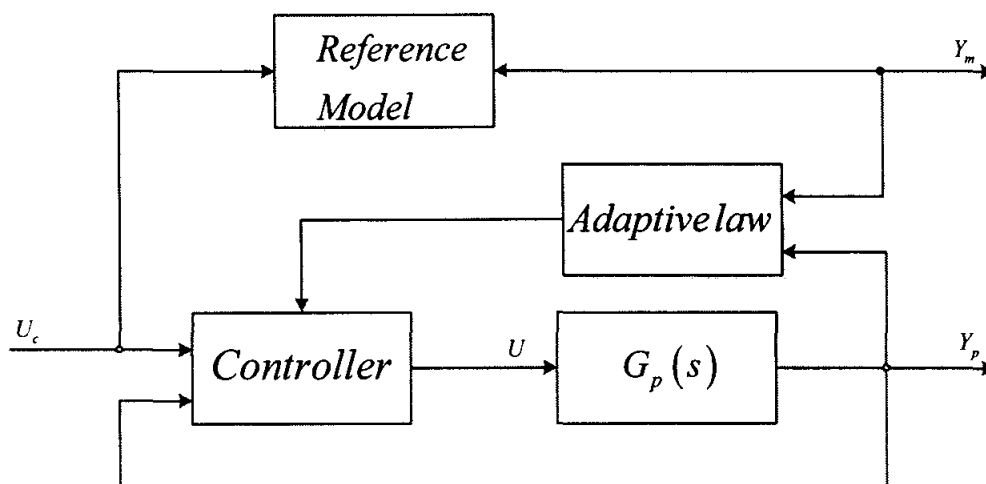
ความสามารถในการปรับตัวได้ของระบบปรับตัวทางอ้อมในรูปที่ 2.2 เกิดจากการป้อนกลับหตุติภูมิ โดยส่วนประมาณจะประมาณพารามิเตอร์ของกระบวนการ และส่งต่อให้กับส่วนออกแบบเพื่อออกแบบพารามิเตอร์ให้กับส่วนควบคุม เมื่อกระบวนการมีการเปลี่ยนแปลงพารามิเตอร์ ส่วนออกแบบจะออกแบบพารามิเตอร์ของส่วนควบคุมให้เหมาะสมกับกระบวนการที่เปลี่ยนไป ส่วนควบคุมที่ได้รับการปรับเปลี่ยนพารามิเตอร์ตามกระบวนการนี้ จะควบคุมพฤติกรรมของกระบวนการ ด้วยกลไกของวงรอบปฐมภูมิต่อไป

โครงสร้างระบบปรับตัวเองโดยตรงมีรูปแบบคล้ายกับระบบปรับตัวเองโดยอ้อมในรูปที่ 2.2 เพียงแต่ลบส่วนออกแบบทิ้ง และเปลี่ยนส่วนประมาณเป็นส่วนปรับพารามิเตอร์ของส่วนควบคุมดังในรูปที่ 2.3



รูปที่ 2.3 โครงสร้างระบบปรับตัวเองโดยตรง

ทั้งโครงสร้างของการปรับตัวโดยตรงและโดยอ้อมมีทั้งข้อดีและข้อเสีย ข้อดีแบบทางอ้อมคือความอิสระที่จะเลือกรูปแบบของส่วนออกแบบ ข้อดีที่สองของการปรับตัวโดยอ้อมคือ การมีส่วนประมาณพารามิเตอร์ จะทำให้ผู้ปฏิบัติการได้รู้ถึงสถานะปัจจุบันของกระบวนการและข้อผิดพลาดต่างๆ ของกระบวนการ อย่างไรก็ตามการปรับตัวเองโดยอ้อมไม่สะดวกต่อการวิเคราะห์ทางทฤษฎี เพราะว่าจะซับซ้อน โครงสร้างการปรับตัวเองโดยตรงมักจะซับซ้อนน้อยกว่า ดังนั้นเราสามารถพิสูจน์เสถียรภาพของระบบโดยรวมได้ง่ายกว่า



รูปที่ 2.4 บล็อกไดอะแกรม MRAC

ในรูปที่ 2.4 แสดงให้เห็นว่าส่วนควบคุมปฐมภูมิจะทำหน้าที่ควบคุมพฤติกรรมของระบบวงรอบปิดในลักษณะเดียวกันกับระบบที่ปรับตัวเองโดยทั่วไป อย่างไรก็ตามสถานการณ์ที่ไม่ทราบค่าพารามิเตอร์ของระบบ หรือพารามิเตอร์ของระบบที่เปลี่ยนแปลงค่าไปตามเวลา การตั้งค่าพารามิเตอร์ของส่วนควบคุมปฐมภูมิให้มีค่าคงที่ จะทำให้ไม่สามารถควบคุมพฤติกรรมของระบบวงรอบปิดให้เป็นไปตามต้องการได้ทุกสถานการณ์

โดยเทคนิคของ MRAC ผลตอบสนองหรือพฤติกรรมของระบบต่ออินพุต  $U_c$  ตามที่เราต้องการสามารถกำหนดโดยรูปแบบอ้างอิง หรือ Reference Model ซึ่งแสดงด้วย  $y_m$  องค์ประกอบในการปรับตัวจะติดตามเปรียบเทียบผลตอบสนองของกระบวนการ  $y_p$  กับ  $y_m$  และ ตั้งค่าพารามิเตอร์ให้กับส่วนควบคุมปฐมภูมิเพื่อที่จะนำผลตอบสนองของกระบวนการให้เข้าสู่ผลตอบสนองของโมเดลอ้างอิงในองค์ประกอบของส่วนปรับตัวเอง นอกจากใช้ผลตอบสนองของกระบวนการ  $y_p$  แล้วยังอาจใช้ สเตต (State) ของกระบวนการ  $x_p$  และอินพุตของกระบวนการ  $U_c$  ลักษณะของ MRAC ส่วนใหญ่มักจะเป็นการปรับตัวเองโดยตรง เพราะไม่มีส่วนของการประมาณพารามิเตอร์ของกระบวนการอย่างชัดเจน

จากรูปที่ 2.4 แสดงระบบ MRAC ซึ่งประกอบด้วยวงรอบป้อนกลับสองวงรอบ ซึ่งได้กล่าวไว้ในรูปที่ 2.2 และ 2.3 วงรอบในประกอบด้วยวงรอบป้อนกลับปฐมภูมิ วงรอบนอกประกอบด้วยส่วนที่ทำหน้าที่ปรับตัวเองโดยทั่วไป วงรอบป้อนกลับปฐมภูมิจะทำงานที่ความเร็วสูงกว่าวงรอบปรับตัวเอง ดังนั้นพารามิเตอร์ของกระบวนการจะถูกสมมุติว่า เปลี่ยนแปลงช้ากว่าสเตตของกระบวนการ

ในกรณีรูปที่ 2.4 แสดงว่าโมเดลอ้างอิงจะขนานกับกระบวนการ แต่ไม่จำเป็นว่าระบบปรับตัวเอง MRAC จำเป็นต้องมีโมเดลอ้างอิงที่ขนานกับกระบวนการเสมอไป ระบบ MRAC บางครั้งจะนำโมเดลอ้างอิงต่ออนุกรมกับกระบวนการ ในการต่ออนุกรมนี้ โมเดลอ้างอิงจะทำหน้าที่เป็นตัวกำเนิดอินพุตอ้างอิง ซึ่งเป็นประโยชน์ต่อการควบคุมตรงที่ระบบโดยทั่วไปจะไม่สามารถตอบสนองต่ออินพุตที่เป็นขั้นบันไดได้ ดังนั้นตัวกำเนิดอินพุตอ้างอิงจะกรองสัญญาณอินพุตให้เรียบขึ้น และการตอบสนองต่อกระบวนการจะเป็นไปตามอินพุตได้ดีขึ้นในลักษณะที่มีผลต่อการอิ่มตัวของระบบด้วย (Saturation) นอกจากนี้โมเดลอ้างอิงจะต่อขนาน และต่ออนุกรมกับกระบวนการแล้วยังสามารถนำทั้งสองแบบมารวมกันเป็น MRAC แบบ อนุกรม-ขนาน ได้อีก ซึ่งมีข้อดีและข้อเสียต่างกัน ในงานวิจัยนี้จะศึกษาเฉพาะ MRAC ที่มีรูปแบบอ้างอิงที่ต่อขนานกับกระบวนการ (Parallel MRAC) ส่วนสำคัญที่สุดใน MRAC ได้แก่ การออกแบบกฎการปรับตัว ซึ่งมีหลายวิธี สำหรับงานวิจัยนี้จะศึกษาและทดลองเฉพาะวิธีของ MIT

### 2.3.1 องค์ประกอบของ MRAC

ในหัวข้อนี้จะกล่าวถึงส่วนประกอบที่สำคัญของระบบ MRAC อันได้แก่ กระบวนการ (Process) ตัวควบคุมปฐมภูมิ (Process) ตัวควบคุมปฐมภูมิ (Primary Controller) รูปแบบอ้างอิง (Reference Model) และที่มาของกฎการปรับตัว [1]

#### - กระบวนการ

แม้กระบวนการจะไม่อยู่ในส่วนของการออกแบบตัวควบคุมแบบปรับตัวเอง แต่เนื่องจากกระบวนการเป็นส่วนที่ตัวควบคุมจะต้องเกี่ยวข้องกับ จึงจำเป็นที่จะได้พิจารณาในส่วนนี้ ในการออกแบบตัวควบคุมนั้นต้องสมมุติกระบวนการให้เป็นไปในทิศทางบางอย่างที่อาจไม่เป็นจริงในธรรมชาติ ตัวอย่างเช่น ทฤษฎีการปรับตัวเองที่จะพิจารณาต่อไปจะต้องสมมุติว่ากระบวนการควรจะเป็นเชิงเส้น ควรทราบอันดับของทรานส์เฟอร์ฟังก์ชัน และควรรู้ถึงเครื่องหมายของ DC Gain รวมทั้ง ทรานส์เฟอร์ฟังก์ชันควรเป็นแบบเฟสต่ำสุด และ กระบวนการควรจะปราศจากการรบกวน (Disturbance)

ดังที่กล่าวไว้แล้วว่ากระบวนการจะไม่มีทางสอดคล้องกับเงื่อนไขทั้งหมดนี้ ยกตัวอย่างเช่นอันดับของกระบวนการที่แน่นอนเราไม่สามารถรู้ได้ ซึ่งโดยมากอันดับของกระบวนการที่นำมาใช้จะมาจากกรณีวิเคราะห์ถึงส่วนประกอบของระบบ หรือ ดูผลตอบสนองทั้งในโดเมนของเวลา หรือ โดเมนความถี่ และเป็นที่ยืนยันว่าไม่มีกระบวนการที่เป็นเชิงเส้นอย่างแท้จริง หรือปราศจากการรบกวน

อย่างไรก็ตามถึงแม้ว่าคุณสมบัติของกระบวนการจะไม่สามารถเป็นไปตามที่สมมุติไว้ แต่ก็ได้หมายความว่า การควบคุมที่สามารถปรับตัวได้จะไม่สามารถเป็นไปได้ ดังนั้นสิ่งสำคัญในอันดับต่อมา ก็คือจะหาอย่างไรให้ตัวควบคุมที่ปรับตัวได้สามารถทำงานได้ในสภาวะที่ไม่สมบูรณ์ นั่นคือ สามารถทำงานได้ในกระบวนการจริงๆ

### - ส่วนควบคุมปฐมภูมิ (Primary Controller)

ส่วนควบคุมปฐมภูมิดังรูปที่ 2.24 โดยหลักการสามารถมีโครงสร้างใดๆ ก็ได้ตามการออกแบบตัวควบคุมเชิงเส้น อย่างไรก็ตามควรจะสอดคล้องกับคุณสมบัติบางประการ

ประการแรกจะต้องสอดคล้องกับสภาวะ Perfect Model-Matching ซึ่งหมายถึงการตั้งค่าพารามิเตอร์ของตัวควบคุมจะต้องทำให้พฤติกรรมของวงรอบวงปิดเท่ากับรูปแบบอ้างอิงได้

ประการที่สองสำหรับการใช้การควบคุมโดยตรง สัญญาณควบคุม  $U_c$  จะต้องมีคุณสมบัติเป็นเชิงเส้นต่อพารามิเตอร์ โดยทั่วไปส่วนควบคุมปฐมภูมิที่ใช้ใน MRAC ประกอบด้วยส่วนที่ทำหน้าที่กำเนิดเวกเตอร์ของสัญญาณ (Signal Vector)

### - รูปแบบอ้างอิง (The Reference Model)

รูปแบบอ้างอิงจะเป็นตัวบ่งชี้ถึงผลตอบสนองต่อสัญญาณอ้างอิง  $U_c$  ที่เราต้องการ รูปแบบอ้างอิงมักจะได้จากการจำลองภายในคอมพิวเตอร์ การกำหนดรูปแบบอ้างอิงจะต้องกระทำให้สมเหตุสมผลคือระบบในความเป็นจริงสามารถเป็นไปตามรูปแบบนั้นๆ ได้ เพราะหากกำหนดรูปแบบอ้างอิงที่มีผลตอบสนองเร็วเกินไปจะทำให้เกิดสภาวะอิ่มตัว (Saturation) ในส่วนหนึ่งส่วนใดของระบบได้ และจะนำมาซึ่งความไม่เสถียรของระบบดังจะได้แสดงถึงผลของการอิ่มตัวต่อการเสถียรภาพของระบบต่อไปในการทดลอง

### - กฎการปรับตัว (Adaptive Laws)

กฎที่สำคัญในการออกแบบ MRAC คือ กฎการปรับตัวและการพิสูจน์กฎการปรับตัวซึ่งในที่นี้จะใช้วิธีของ MIT โดยสมมุติว่ากระบวนการเป็นเชิงเส้นที่มีพารามิเตอร์ไม่ทราบค่าหรือแปรค่าตามเวลา และกระบวนการมีหนึ่งอินพุตหนึ่งเอาต์พุต (SISO) การควบคุมเป็นสเตป้อนกลับและพารามิเตอร์ของตัวควบคุมที่จะปรับคือเกนป้อนกลับ (Feedback Gain)

สำหรับงานวิจัยนี้จะออกแบบ MRAC ด้วยกฎของ MIT (MIT Rule) ซึ่งเริ่มจากการการคำนวณหาความแตกต่างของสัญญาณเอาต์พุตของกระบวนการจริงกับโมเดลอ้างอิง

$$\mathcal{E} = y_p - y_m \quad (2.2)$$

$$J(\theta) = \frac{1}{2} \mathcal{E}^2(\theta) \quad (2.3)$$

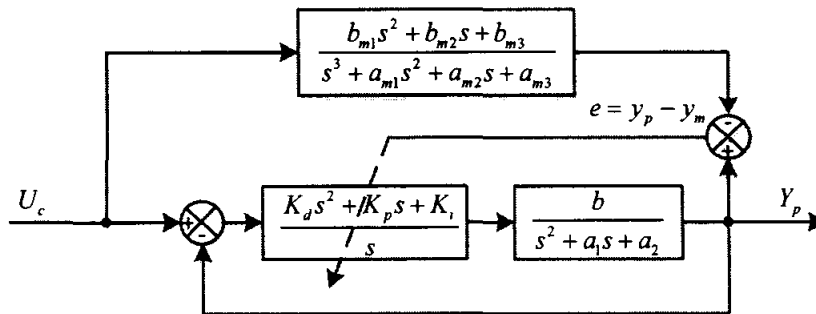
การปรับพารามิเตอร์ของตัวควบคุมเพื่อลดความแตกต่างของสัญญาณเอาต์พุตระหว่างกระบวนการจริงกับโมเดลอ้างอิงด้วยบล็อก *Adjustment Mechanism* ซึ่งเป็นกฎของ MIT ซึ่งมีสมการพื้นฐานคือ

$$\frac{d\theta}{dt} = -\gamma \frac{\partial J}{\partial \theta} = -\gamma \epsilon \frac{\partial \epsilon}{\partial \theta} \quad (2.4)$$

เมื่อ  $\theta$  คือพารามิเตอร์ของตัวควบคุม,  $\partial \epsilon / \partial \theta$  ความไวของค่าผิดพลาดเมื่อเทียบกับ  $\theta$  และ  $\gamma$  ค่าเกณฑ์ในการปรับความเหมาะสม

### 2.3.2 เทคนิคการออกแบบตัวควบคุมพีไอดีด้วย MRAC

ในส่วนนี้จะนำเสนอการออกแบบตัวควบคุมพีไอดีด้วย MRAC [8] เมื่อพิจารณาระบบควบคุมอันดับหนึ่ง  $\frac{b}{s^2 + a_1s + a_2}$  จะได้โครงสร้างของระบบควบคุมดังรูปที่ 2.5



รูปที่ 2.5 โครงสร้างของระบบควบคุม

จากบล็อกไดอะแกรมในรูปที่ 2.5 หาทรานส์เฟอร์ฟังก์ชันได้ตามสมการที่ (2.5)

$$\frac{Y_p(s)}{U_c(s)} = \frac{b(K_d s^2 + K_p s + K_i)}{s^3 + (a_1 + bK_d)s^2 + (a_2 + bK_p)s + bK_i} \quad (2.5)$$

และ

$$Y_p(s) = \frac{b(K_d s^2 + K_p s + K_i)}{s^3 + (a_1 + bK_d)s^2 + (a_2 + bK_p)s + bK_i} \cdot U_c(s) \quad (2.6)$$

จากสมการที่ (2.5) จะได้จะได้ทรานส์เฟอร์ฟังก์ชันของโมเดลอ้างอิงดังสมการที่ (2.7)

$$\frac{Y_m(s)}{U_m(s)} = \frac{b_{m1} s^2 + b_{m2} s + b_{m3}}{s^3 + a_{m1} s^2 + a_{m2} s + a_{m3}} \quad (2.7)$$

จากสมการที่ (2.5) มีค่าตัวแปรที่ไม่รู้ค่าคือ  $K_p$ ,  $K_i$  และ  $K_d$  ซึ่งหาได้จากสมการที่ (2.8)

$$\begin{aligned}\frac{dK_p}{dt} &= -\gamma_p \frac{\partial J}{\partial K_p} = -\gamma_p \left( \frac{\partial J}{\partial \varepsilon} \right) \left( \frac{\partial \varepsilon}{\partial y} \right) \left( \frac{\partial y}{\partial K_p} \right) \\ \frac{dK_i}{dt} &= -\gamma_i \frac{\partial J}{\partial K_i} = -\gamma_i \left( \frac{\partial J}{\partial \varepsilon} \right) \left( \frac{\partial \varepsilon}{\partial y} \right) \left( \frac{\partial y}{\partial K_i} \right) \\ \frac{dK_d}{dt} &= -\gamma_d \frac{\partial J}{\partial K_d} = -\gamma_d \left( \frac{\partial J}{\partial \varepsilon} \right) \left( \frac{\partial \varepsilon}{\partial y} \right) \left( \frac{\partial y}{\partial K_d} \right)\end{aligned}\quad (2.8)$$

เมื่อ  $\partial J / \partial \varepsilon = \varepsilon$ ,  $\partial \varepsilon / \partial y = 1$  และ  $\partial y / \partial K_p, \partial y / \partial K_i, \partial y / \partial K_d$  สามารถหาได้จากสมการที่ (2.9, 2.10, 2.11) ตามลำดับ

$$\frac{\partial Y_p}{\partial K_p} = \frac{bs}{s^3 + (a_1 + bK_d)s^2 + (a_2 + bK_p)s + bK_i} \cdot [U_c - Y_p] \quad (2.9)$$

$$\frac{\partial Y_p}{\partial K_i} = \frac{b}{s^3 + (a_1 + bK_d)s^2 + (a_2 + bK_p)s + bK_i} \cdot [U_c - Y_p] \quad (2.10)$$

$$\frac{\partial Y_p}{\partial K_d} = \frac{bs^2}{s^3 + (a_1 + bK_d)s^2 + (a_2 + bK_p)s + bK_i} \cdot [U_c - Y_p] \quad (2.11)$$

แทนสมการที่ (2.9, 2.10, 2.11) ลงในสมการ (2.8) จะได้

$$\begin{aligned}\frac{dK_p}{dt} &= -\gamma_p \frac{\partial J}{\partial K_p} = -\gamma_p \varepsilon \frac{bs}{s^3 + (a_1 + bK_d)s^2 + (a_2 + bK_p)s + bK_i} \cdot [U_c - Y_p] \\ \frac{dK_i}{dt} &= -\gamma_i \frac{\partial J}{\partial K_i} = -\gamma_i \varepsilon \frac{b}{s^3 + (a_1 + bK_d)s^2 + (a_2 + bK_p)s + bK_i} \cdot [U_c - Y_p] \\ \frac{dK_d}{dt} &= -\gamma_d \frac{\partial J}{\partial K_d} = -\gamma_d \varepsilon \frac{bs^2}{s^3 + (a_1 + bK_d)s^2 + (a_2 + bK_p)s + bK_i} \cdot [U_c - Y_p]\end{aligned}\quad (2.12)$$

เมื่อกำหนดให้

$$a_{m1} = a_1 + bK_d, \quad a_{m2} = a_2 + bK_p, \quad a_{m3} = bK_i$$

## 2.4 ทฤษฎีเทคนิคทางเดินของราก (Root Locus Technique)

วิธีทางเดินราก (Root Locus) [9] ซึ่งเป็นวิธีที่ศึกษาถึงทางเดินรากของสมการคุณลักษณะของระบบ เมื่อพารามิเตอร์แปรค่าไปจึงเป็นวิธีที่มีประสิทธิภาพวิธีหนึ่งในการศึกษาระบบควบคุม นอกจากนี้เรายังสามารถจะนำวิธี root locus นี้ไปประยุกต์เพื่อหารากของสมการทางพีชคณิตที่มีอันดับสูงๆ ได้อีกด้วยโดยทั่วไปวิธีของ root-locus จะใช้สำหรับพิจารณาทางเดินของรากเมื่อพารามิเตอร์หนึ่งตัวแปรค่าไป แต่ในกรณีที่พารามิเตอร์มากกว่าหนึ่งตัวแปรค่าไปนั้น ก็จะสามารถใช้วิธีของ root locus ได้เช่นกัน

พิจารณาสมการที่มี  $K$  เป็นพารามิเตอร์ที่แปรค่าต่อไปนี้

$$F(s) = s^n + a_1s^{n-1} + \dots + a_{n-1}s + a_n + K(s^m + b_1s^{m-1} + \dots + b_{m-1}s + b_m) = 0 \quad (2.64)$$

โดยที่  $K$  เป็นพารามิเตอร์ที่แปรค่า ในที่นี้กำหนดให้  $K$  แปรค่าจาก  $-\infty$  ถึง  $+\infty$  และจะสมมติให้สัมประสิทธิ์  $a_1, \dots, a_n$  และ  $b_1, \dots, b_m$  นั้นมีค่าคงที่ โดยสัมประสิทธิ์นี้อาจจะเป็นค่าจริงหรือจำนวนเชิงซ้อน (complex number) ก็ได้แต่ส่วนใหญ่เราจะพิจารณาถึงสัมประสิทธิ์ที่เป็นค่าจริงเท่านั้น โดยทั่วไปเราจะแบ่งชนิดของทางเดินรากเป็น 3 ชนิด ดังนี้

1. Root loci : root loci เป็นทางเดินรากเมื่อ  $K$  ในสมการ (2.63) แปรค่าอยู่ในช่วง  $0$  ถึง  $\infty$
2. Complementary root loci : บางครั้งเรียกว่า Inverse root loci เป็นทางเดินรากเมื่อ  $K$  ในสมการ 2.64 แปรค่าอยู่ในช่วง  $-\infty$  ถึง  $0$
3. ถ้านำทางเดินของราก 2 แบบ ในข้อ 1 และ 2 มารวมกัน จะเรียกว่า root loci ที่สมบูรณ์ (complete root loci)
4. Root contours : root contours เป็นทางเดินของรากเมื่อพารามิเตอร์มากกว่า 1 ตัวแปรค่าไป

#### 2.4.1 เงื่อนไขเบื้องต้นของ ROOT LUCI

พิจารณาระบบเชิงเส้นที่มี transfer function ดังนี้

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1 + G(s)H(s)} \quad (2.65)$$

สมการคุณลักษณะของระบบหาได้โดยการกำหนดให้เทอมของส่วนในสมการ (2.65) มีค่าเท่ากับศูนย์ นั่นคือ

$$1 + G(s)H(s) = 0 \quad (2.66)$$

สมการ (2.66) เป็นสมการของ closed-loop transfer function ของระบบที่มีการป้อนกลับเป็นการป้อนกลับแบบ (negative feedback) โดยมี

$C(s)$  = สัญญาณเอาต์พุต

$R(s)$  = อินพุตอ้างอิง

$G(s)$  = forward-path transfer function

$H(s)$  = feedback-path transfer function

$G(s)H(s)$  = loop transfer function

ถ้าจัดรูปสมการ (2.64) ในอยู่ในรูปสมการ (2.66) จะได้ว่า

$$1 + \frac{K(s^m + b_1s^{m-1} + \dots + b_{m-1}s + b_m)}{s^n + a_1s^{n-1} + \dots + a_{n-1}s + a_n} = 0 \quad (2.67)$$

จากสมการ (2.66) และสมการ (2.67) จะให้ผลดังนี้

$$G(s)H(s) = \frac{K(s^m + b_1s^{m-1} + \dots + b_{m-1}s + b_m)}{s^n + a_1s^{n-1} + \dots + a_{n-1}s + a_n} \quad (2.68)$$

เพื่อที่จะศึกษาถึง root loci ที่สมบูรณ์ โดยที่ K แปรค่าจาก  $-\infty$  ถึง  $+\infty$  ถ้าเราเขียนสมการ (2.68) ใหม่เป็น

$$G(s)H(s) = KG_1(s)H_1(s) \quad (2.69)$$

จะเห็นว่า  $G_1(s)H_1(s)$  ไม่ได้ประกอบด้วยพารามิเตอร์ K ดังนั้นเราจะเขียนสมการ (2.66) ใหม่ได้ ดังนี้

$$G(s)H(s) = -\frac{1}{K} \quad (2.70)$$

เนื่องจาก s เป็นจำนวนเชิงซ้อน ดังนั้นจากสมการ (2.69) จะมีเงื่อนไขดังนี้

$$|G_1(s)H_1(s)| = \frac{1}{|K|} \quad -\infty < K < \infty \quad (2.71)$$

$$\angle G_1(s)H_1(s) = (2k + 1)\pi \quad k \geq 0 \quad (2.72)$$

$$\angle G_1(s)H_1(s) = 2k\pi \quad k \geq 0 \quad (2.73)$$

โดยที่  $k = 0, +1, +2, \dots$

การสร้าง root loci จะเป็นวิธีทางกราฟที่จะเริ่มต้นจากการกำหนดตำแหน่ง pole และ zeros ของ loop transfer function  $G(s)H(s)$  ก่อน ดังนั้นขั้นแรกจะต้องเขียนสมการ (2.68) ให้อยู่ในรูปของ poles และ zeros จะได้ว่า

$$G(s)H(s) = \frac{K(s + z_1)(s + z_2)\dots(s + z_m)}{(s + p_1)(s + p_2)\dots(s + p_n)} \quad (2.74)$$

$$= KG_1(s)H_1(s) \quad (2.75)$$

โดยที่ poles และ zeros ในสมการ (2.74) สามารถเป็นไปได้ทั้งค่าจริงหรือจำนวนเชิงซ้อน สมการ (2.71) สมการ (2.72) และสมการ (2.73) สามารถเขียนใหม่ได้ดังนี้

$$\left|G_1(s)H_1(s)\right| = \frac{\prod_{i=1}^m |s+z_i|}{\prod_{j=1}^n |s+p_j|} = \frac{1}{|K|} \quad -\infty < K < \infty \quad (2.76)$$

และ

$$\begin{aligned} \angle G_1(s)H_1(s) &= \sum_{i=1}^m \angle s+z_i - \sum_{j=1}^n \angle s+p_j \\ &= (2k+1)\pi \quad 0 \leq K < \infty \end{aligned} \quad (2.77)$$

$$\begin{aligned} \angle G_1(s)H_1(s) &= \sum_{i=1}^m \angle s+z_i - \sum_{j=1}^n \angle s+p_j \\ &= 2k\pi \quad 0 \leq K < \infty \end{aligned}$$

โดยที่  $k = 0, +1, +2, \dots$

เพื่อเป็นตัวอย่างที่จะแสดงถึงการใช้สมการ (2.76) และสมการ (2.77) ในการสร้าง root loci ให้พิจารณา loop transfer function  $G(s)H(s)$  หรือ forward-path transfer function ถ้า  $H(s)=1$  ต่อไปนี้

$$G(s)H(s) = \frac{K(s+z_1)}{s(s+p_2)(s+p_3)} \quad (2.78)$$

รูปที่ 2.10 จะแสดงถึงตำแหน่งของ zero และ pole ที่เราสมมติขึ้นมาเอง จะเป็นจุดที่อยู่บน root loci ( $0 < K < \infty$ ) ได้ก็จะต้องสอดคล้องในสมการ (2.75) และสมการ (2.76) นั่นคือ

$$\frac{|s_1+z_1|}{|s_1||s_1+p_2||s_1+p_3|} = \frac{1}{|K|} \quad (2.79)$$

และ

$$\angle s_1+z_1 - (\angle s_1 + \angle s_1+p_2 + \angle s_1+p_3) = (2K+1)\pi \quad (2.80)$$

โดยที่  $K = 0, +1, +2, \dots$

ทำนองเดียวกัน จุด  $s_1$  จะเป็นจุดที่อยู่บน Complementary root loci ( $-\infty < K \leq 0$ ) ได้ค่าก็ต่อเมื่อสอดคล้องกับเงื่อนไขในสมการ (2.76) นั่นคือ

$$\angle s_1+z_1 - (\angle s_1 + \angle s_1+p_2 + \angle s_1+p_3) = (2K+1)\pi \quad (2.81)$$

โดยที่  $K = 0, +1, +2, \dots$

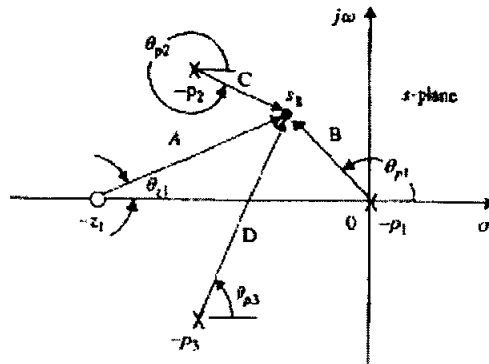
จากรูปที่ 2.6 มุม  $\theta_{z_1}, \theta_{p_1}, \theta_{p_2}$  และ  $\theta_{p_3}$  เป็นมุมของ vectors ที่วัดได้โดยมีแกน real หรือแนวนอนเป็นแนวอ้างอิง สมการ (2.80) และสมการ (2.81) จะเขียนใหม่ได้ว่า

$$\theta_{z_1} - (\theta_{p_1} + \theta_{p_2} + \theta_{p_3}) = (2k+1)\pi$$

$$0 \leq K < \infty \quad (2.82)$$

และ

$$\theta_{z_1} - (\theta_{p_1} + \theta_{p_2} + \theta_{p_3}) = 2k\pi \quad -\infty < K \leq 0 \quad (2.83)$$



รูปที่ 2.6 pole, zero ของ  $G(s) = \frac{[K(s+z_1)]}{[s(s+p_2)(s+p_3)]}$

ถ้าจุด  $s_1$  สอดคล้องกับเงื่อนไขของสมการ (2.82) หรือสมการ (2.83) เราก็จะสามารถคำนวณหาค่าของ  $K$  ที่จุด  $s_1$  นั้นได้โดยใช้สมการ (2.79)

$$|K| = \frac{|s_1| |s_1 + p_2| |s_1 + p_3|}{|s_1 + z_1|} \quad (2.84)$$

ถ้ารูปที่ 2.10 นี้ใช้สเกลความยาวของ vector ด้วย  $a, b$  และ  $c$  ตามลำดับแล้ว จะได้ค่าของ  $K$  ดังนี้

$$|K| = \frac{BCD}{A} \quad (2.85)$$

สำหรับเครื่องหมายของ  $K$  นั้นจะขึ้นอยู่กับว่าจุด  $s_1$  นั้นอยู่บน root loci หรืออยู่บน complementary root loci ดังที่กล่าวมาแล้ว เราสามารถจะสร้างทางเดินรากที่สมบูรณ์ได้ตามขั้นตอนต่อไปนี้

1. หาจุด  $s_1$  ทุกๆจุดในระนาบ  $s$  ที่สอดคล้องกับสมการ (2.72) และสมการ (2.73)

2. ค่าต่างๆ ของ  $K$  บน root loci และบน complementary root loci หาได้โดยการใช้สมการ (2.71) โดยที่  $K$  จะเป็นบวกเมื่อจุด  $s_1$  อยู่บน root loci และเป็นลบเมื่อจุด  $s_1$  อยู่บน complementary root loci

## 2.4.2 การสร้าง ROOT LOCI ที่สมบูรณ์

วิธีของ root loci ได้คิดค้นขึ้นโดย W.R.Evans ในปี ค.ศ.1948 และได้ถูกพัฒนาขึ้นมาเพื่อใช้วิเคราะห์และออกแบบระบบอย่างกว้างขวาง กฎต่างๆ ในการสร้าง root loci ถูกสร้างขึ้นโดยอาศัยความสัมพันธ์ ระหว่าง poles และ zeros ของ  $G(s)H(s)$  และ zeros ของสมการคุณลักษณะ

**กฎที่ 1**  $K = 0$

จุดที่  $K$  มีค่าเท่ากับ 0 root loci ที่สมบูรณ์จะอยู่ที่ poles ของ  $G(s)H(s)$

**พิสูจน์** จากสมการ (2.75)

$$|G_1(s)H_1(s)| = \frac{\prod_{i=1}^m |s + z_i|}{\prod_{j=1}^n |s + p_j|} = \frac{1}{|K|} \quad (2.86)$$

เมื่อ  $K$  เข้าใกล้ 0 สมการ (2.86) จะมีค่าเข้าใกล้  $\infty$  และจะสมนัยกับการที่  $s$  เข้าใกล้ poles ของ  $G_1(s)H_1(s)$  นั่นคือ poles ของ  $G(s)H(s)$  นั่นคือ  $s$  มีค่าเข้าใกล้  $-p_j$  ( $j = 1, 2, \dots, n$ ) ใช้ได้ทั้งกับกรณีของ root loci และ complementary root loci เพราะค่าของ  $K$  ในกรณีนี้ไม่ได้คิดเครื่องหมาย

**กฎที่ 2**  $K = \pm\infty$

จุดที่  $K = \pm\infty$  บน root loci ที่สมบูรณ์จะอยู่ที่ zeros ของ  $G(s)H(s)$

**พิสูจน์** อ้างถึงสมการ (2.86) เมื่อ  $K$  เข้าใกล้  $\pm\infty$  สมการ 2.86 จะมีค่าเข้าใกล้ 0 ซึ่งจะสมนัยกับ  $s$  เข้าใกล้ zeros ของ  $G(s)H(s)$  นั่นคือ  $s$  มีค่าเข้าใกล้  $-z_i$  ( $i = 1, 2, \dots, m$ )

**กฎที่ 3** จำนวนแขนง (branch) ของ root loci ที่สมบูรณ์

ถ้า  $n =$  จำนวนของ finite poles ของ  $G(s)H(s)$

$m =$  จำนวน finite zeros ของ  $G(s)H(s)$

และ  $N =$  จำนวนแขนงของ root loci ที่สมบูรณ์

แล้วจะได้ว่า

$$N = n \quad \text{ถ้า} \quad n > m$$

$$N = m \quad \text{ถ้า} \quad m > n$$

$$N = n = m \quad \text{ถ้า} \quad n = m$$

#### กฎที่ 4 การสมมาตรกันของ root loci ที่สมบูรณ์

Root loci ที่สมบูรณ์จะสมมาตรกันเมื่อเทียบกับแกน real ในระนาบ  $s$  โดยทั่วไป root loci ที่สมบูรณ์จะสมมาตรกันเมื่อเทียบกับแกนที่ poles และ/หรือ zeros ของ  $G(s)H(s)$  สมมาตรกัน

ถ้าหากว่า poles และ/หรือ zeros ของ  $G(s)H(s)$  สมมาตรกันกับแกนที่ไม่ใช่แกน real ในระนาบ  $s$  แล้ว เราสามารถจะถือว่าแกนที่ poles และ/หรือ zeros ของ  $G(s)H(s)$  สมมาตรกัน เป็นแกน real ในระนาบเชิงซ้อนอันใหม่ โดยการใช้วิธี linear transformation และเรียกแกนนี้ว่า axis of symmetry

#### กฎที่ 5 Asymptotes ของ root loci ที่สมบูรณ์ (พฤติกรรมของ root loci ที่ $s = \infty$ )

คุณสมบัติของ root loci ที่สมบูรณ์เข้าใกล้ ในระนาบ  $s$  มีความสำคัญอย่างหนึ่ง เนื่องจาก  $n = m$  จะมีจำนวนทางเดินของรากเท่ากับ  $2|n - m|$  ที่จะเข้าใกล้  $\infty$  ระนาบ  $s$  ฉะนั้น สำหรับ  $s$  ที่มีค่ามากๆ มุมของเส้น asymptotes ของ root loci ( $K \geq 0$ ) จะถูกกำหนดโดย

$$\theta_k = \frac{(2k + 1)\pi}{n - m} \quad (2.87)$$

โดยที่  $k = 0, +1, +2, \dots, |n - m| - 1$

$n =$  จำนวนของ finite poles ของ  $G(s)H(s)$

$m =$  จำนวนของ finite zeros ของ  $G(s)H(s)$

สำหรับมุมของเส้น asymptotes สำหรับ complementary root loci คือ

$$\theta_k = \frac{2k\pi}{n - m} \quad (2.88)$$

โดยที่  $k, n$  และ  $m$  จะเหมือนกับกรณีสมการ (2.87)

ถ้าหารสมการ (2.64) ด้วยเทอม

$$s^m + b_1 s^{m-1} + \dots + b_{m-1} s + b_m$$

แล้วจะได้ว่า

$$\frac{s^n + a_1 s^{n-1} + \dots + b_{n-1} s + a_n}{s^m + b_1 s^{m-1} + \dots + b_{m-1} s + b_m} + K = 0 \quad (2.89)$$

จัดรูปสมการ (2.89) และพิจารณาเพียงเทอมที่มี  $s$  อันดับสูงเทอมแรก จะได้

$$s^{n-m} + (a_1 + b_1) s^{n-m-1} \cong -K \quad (2.90)$$

หรือ

$$s \left( 1 + \frac{a_1 - b_1}{s} \right)^{1/(n-m)} \cong (-K)^{1/(n-m)} \quad (2.91)$$

เทอม  $[1 + (a_1 - b_1)/s]^{1/(n-m)}$  นั้นสามารถกระจายได้โดยวิธีของ binomial expansion ดังนั้นสมการ (2.91) จึงกลายเป็น

$$s \left[ 1 + \frac{a_1 - b_1}{(n-m)s} + \dots \right] \cong (-K)^{1/(n-m)} \quad (2.92)$$

ทำนองเดียวกัน ถ้าเราพิจารณาเพียง 2 เทอมแรกทางซ้ายมือ จะได้

$$s + \frac{a_1 - b_1}{n-m} \cong (-K)^{1/(n-m)} \quad (2.93)$$

ต่อไปแทน  $s$  ด้วย  $\sigma + j\omega$  และใช้ทฤษฎีทางพีชคณิตของ De Moivre สมการ (2.93) สามารถเขียนใหม่ได้ดังนี้

$$\sigma + j\omega + \frac{a_1 - b_1}{n-m} \cong (-K)^{1/(n-m)} \left[ \cos \frac{(2k+1)\pi}{n-m} + j \sin \frac{(2k+1)\pi}{n-m} \right] \quad (2.94)$$

สำหรับ  $0 \leq k < \infty$  และ

$$\sigma + j\omega + \frac{a_1 - b_1}{n-m} \cong |K|^{1/(n-m)} \left[ \cos \frac{2k+1\pi}{n-m} + j \sin \frac{2k\pi}{n-m} \right] \quad (2.95)$$

สำหรับ  $-\infty < k \leq 0$  และ  $k = 0, \pm 1, \pm 2, \dots$

เทียบ real path และ imaginary ทั้งสองข้างของสมการ (2.94) ให้เท่ากัน เราจะได้ว่า สำหรับ  $0 \leq k < \infty$

$$\sigma + \frac{a_1 - b_1}{n-m} \cong K^{1/(n-m)} \cos \frac{(2k+1)\pi}{n-m} \quad (2.96)$$

และ

$$\omega \cong K^{1/(n-m)} \sin \frac{(2k+1)\pi}{n-m} \quad (2.97)$$

จากสมการ (2.96) และสมการ (2.97) หาค่าของ  $K^{1/(n-m)}$  ได้ว่า

$$K^{1/(n-m)} \cong \frac{\omega}{\sin \frac{(2k+1)\pi}{n-m}} \cong \frac{\sigma + \frac{a_1 - b_1}{n-m}}{\cos \frac{(2k+1)\pi}{n-m}} \quad (2.98)$$

และ  $\omega$  มีค่าเป็น

$$\omega \cong \tan \frac{(2k+1)\pi}{n-m} \left( \sigma + \frac{a_1 - b_1}{n-m} \right) \quad (2.99)$$

สมการ (2.99) เป็นสมการเส้นตรงในระนาบ  $K_1$  และอยู่ในรูป

$$\omega \cong M(\sigma - \sigma_1) \quad (2.100)$$

โดยที่  $M$  จะเป็น slope ของเส้นตรงหรือเส้น asymptote และ  $\sigma_1$  จะเป็นจุดตัดบนแกน  $\sigma$  จากสมการ (2.99) และสมการ (2.100) เราได้

$$M = \tan \frac{(2k+1)\pi}{n-m} \quad (2.101)$$

โดยที่  $k = 0, +1, +2, \dots, |n-m|-1$  และ

$$\sigma_1 = -\frac{a_1 - b_1}{n-m} \quad (2.102)$$

ทำนองเดียวกัน สำหรับกรณีของ Complementary root loci ในสมการ (2.95) เราจะได้ว่า

$$M = \tan \frac{2k\pi}{n-m} \quad (2.103)$$

$$\sigma_1 = -\frac{a_1 - b_1}{n-m}$$

โดยที่  $k = 0, +1, +2, \dots, |n-m|-1$

### กฎที่ 6 จุดตัดของเส้น Asymptotes (จุด centroid)

ก) จุดตัดของเส้น Asymptotes จำนวน  $2|n-m|$  เส้นของ root loci ที่สมบูรณ์จะอยู่บนแกน real ของระนาบ  $s$

ข) จุดตัดของเส้น Asymptote จะถูกกำหนดโดย

$$\sigma_1 = -\frac{b_1 - a_1}{n-m} \quad (2.104)$$

โดยที่  $a_1, b_1$  กำหนดในสมการ (2.104)

$n$  = จำนวนของ Finite poles ของ  $G(s)H(s)$

$m$  = จำนวนของ Finite zeros ของ  $G(s)H(s)$

กำหนดฟังก์ชัน  $G(s)H(s)$  ตามสมการ (2.68) แล้ว สมการ (2.68) สามารถเขียนใหม่ได้ดังนี้

$$\begin{aligned} \sigma_1 &= -\frac{b_1 - a_1}{n-m} \\ &= \frac{\sum \text{finite poles ของ } G(s)H(s) - \sum \text{finite zeros ของ } G(s)H(s)}{\text{จำนวนของ finite poles ของ } G(s)H(s) - \text{จำนวนของ finite zeros ของ } G(s)H(s)} \end{aligned} \quad (2.105)$$

จากกฎทางพีชคณิต

$$\begin{aligned} -a_1 &= \text{ผลบวกของรากของ } s^n + a_1s^{n-1} + \dots + a_{n-1}s + a_n = 0 \\ &= \text{ผลบวกของ finite poles ของ } G(s)H(s) \end{aligned} \quad (2.106)$$

$$\begin{aligned} -b_1 &= \text{ผลบวกของรากของ } s^m + b_1s^{m-1} + \dots + b_{m-1}s + b_m = 0 \\ &= \text{ผลบวกของ finite zeros ของ } G(s)H(s) \end{aligned} \quad (2.107)$$

จากนี้สมการ (2.105) นั้นสามารถจะใช้ได้กับ root loci และ complementary root loci เพราะค่า  $\sigma_1$  ของ root loci และ complementary root loci มีค่าเท่ากัน

### กฎที่ 7 Root loci บนแกน real

- ก) Root loci : บนส่วนต่างๆ ที่กำหนดให้ของแกน real จะพบว่ามี root loci ( $K \geq 0$ ) อยู่ในส่วนนั้นๆ ถ้าจำนวนรวมกันของ poles และ zeros ของ  $G(s)H(s)$  อยู่บนแกน real ซึ่งอยู่ทางขวามือของส่วนที่พิจารณานั้นมีจำนวนเป็นคี่
- ข) Complementary root loci : บนส่วนที่กำหนดให้ของแกน real จะพบว่ามี complementary root loci ( $K \leq 0$ ) อยู่ในส่วนนั้นถ้าหากว่าจำนวนรวมกันของ poles และ zeros ของ  $G(s)H(s)$  ที่อยู่บนแกน real ซึ่งอยู่ทางขวามือของส่วนที่พิจารณานั้นมีจำนวนเป็นคู่ หรือเราอาจกล่าวได้ว่าจะมี complementary root loci ในส่วนต่างๆบนแกน real ที่ไม่มี root loci

สำหรับ complex poles และ complex zeros ของ  $G(s)H(s)$  จะไม่เกี่ยวข้องกับการพิจารณาหา root loci หรือ complementary root loci บนแกน real ไม่ว่ากรณีใดๆ

### กฎที่ 8 มุมที่ออกจาก Pole (Angle of departure) และมุมที่เข้าหา zero (Angle of arrival) ของ root loci ที่สมบูรณ์

มุมเหล่านี้จะหาได้โดยใช้สมการ (2.76) ตัวอย่างเช่นถ้าต้องการที่จะหามุมที่ root loci ออกจาก pole ที่จุด  $-1 + j1$  มุมที่เราต้องการ และสอดคล้องกับสมการ (2.76) นั่นคือ

$$\angle G(s_1)H(s_1) = -(\theta_1 + \theta_2 + \theta_3 + \theta_4) = (2k + 1)180^\circ \quad (2.108)$$

หาได้โดยใช้สมการ (2.72) สมการ (2.108) จึงกลายเป็น

$$-(135^\circ + \theta_2 + 90^\circ + 26.6^\circ) = (2k + 1)180^\circ \quad (2.109)$$

กำหนดให้  $k = 0$  จะได้

$$\theta_2 = -431.6^\circ$$

หรืออาจจะกล่าวได้ว่า

$$\theta_2 = -71.6^\circ$$

เมื่อหามุมของ root loci ที่ออกจาก pole หรือเข้าหา zero ของ  $G(s)H(s)$  ได้แล้ว มุมของ complementary root loci ที่จุดเดียวกันจะมีค่าต่างไปจากมุมที่หาได้ที่จุดนั้นๆ เท่ากับ 180 องศา โดยใช้สมการ (2.77)

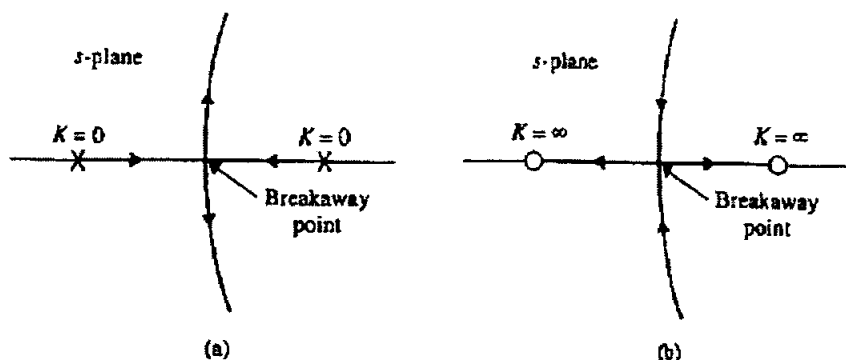
### กฎที่ 9 จุดตัดของ Root locus กับแกน imaginary

จุดที่ Root loci ที่สมบูรณ์ตัดกับแกน imaginary และค่าของ K ณ. ที่จุดตัดนี้สามารถหาได้โดยใช้ Routh-Hurwitz criterion

สำหรับกรณีที่มีจุดตัดหลายจุด ค่าของ K และ จะหาได้ง่ายกว่าโดยวิธีของ Bode diagram

### กฎที่ 10 วิธีการหาจุด Breakaway (จุด Saddle) บน root loci ที่สมบูรณ์

จุด breakaway หรือจุด saddle บน root loci ของสมการใดๆ หมายถึงว่า สมการนั้นๆ จะมีรากซ้ำ (multiple-order roots) รูปที่ 2.7 (a) แสดงถึงกรณีที่ root loci จำนวน 2 แขนงมาพบกันที่จุด breakaway บนแกน real จากนั้นก็แยกออกจากกันไปในทิศทางตรงกันข้ามตามแกน สำหรับรูปที่ 2.11 (b) ก็แสดงถึงจุด breakaway อีกกรณีหนึ่ง



รูปที่ 2.7 ตัวอย่างจุด breakaway บนแกน real ในระนาบ s

## 2.5 บทสรุป

เพื่อให้เห็นถึงความแตกต่างในการออกแบบควบคุม ผู้วิจัยจึงได้นำเอาทฤษฎีและเทคนิคต่างๆ สำหรับการการออกแบบตัวควบคุมมาใช้ ซึ่งทฤษฎีและเทคนิคแต่ละอย่างก็จะมีวิธีคิดแตกต่างกันไป แต่มีวัตถุประสงค์เดียวกัน คือ การนำเอาทฤษฎีที่ได้กล่าวมาทั้งหมดมาออกแบบตัวควบคุม สำหรับควบคุมกระบวนการผลิตและแผ่นเพลท ซึ่งประกอบไปด้วย ทฤษฎีของ MRAC และ Root Locus ซึ่งผู้วิจัยได้รวบรวมข้อมูลของแต่ละวิธีไว้โดยละเอียด และเป็นส่วนที่เกี่ยวข้องกับงานวิจัย

## บทที่ 3

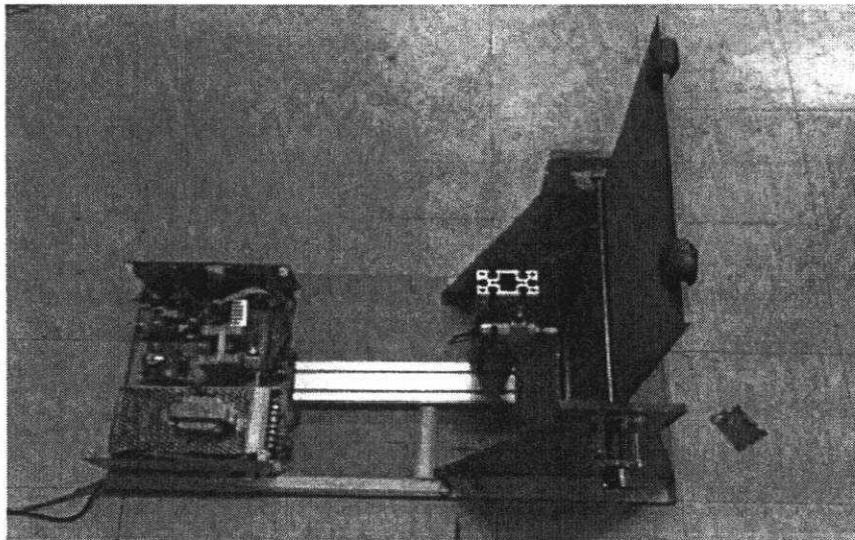
# การประยุกต์ใช้งานฮาร์ดแวร์และซอฟต์แวร์

### 3.1 กล่าวนำ

ในการทำงานวิจัยนั้น ปัจจัยที่ผู้วิจัยคิดว่าจะมีความสำคัญอีกประการหนึ่งก็คือเรื่องของฮาร์ดแวร์หรือกระบวนการ และซอฟต์แวร์หรือโปรแกรมที่ใช้สำหรับจัดการกระบวนการให้เป็นไปตามหลักทฤษฎี ในบทนี้ผู้วิจัยจะกล่าวถึงกระประยุกต์ใช้งานของฮาร์ดแวร์และซอฟต์แวร์ที่เกี่ยวข้องกับงานวิจัย ในส่วนฮาร์ดแวร์หมายถึงองค์ประกอบทางกายภาพทั้งหมดที่นำมาประกอบกันเป็นกระบวนการพัฒลมและแผ่นเพลท ส่วนซอฟต์แวร์เป็นโปรแกรมที่ใช้ในการควบคุมกระบวนการให้เป็นไปตามหลักทฤษฎี ซึ่งรายละเอียดของและเนื้อหาของแต่ละหัวข้อสามารถอธิบายได้ดังต่อไปนี้

### 3.2 องค์ประกอบของกระบวนการพัฒลมและแผ่นเพลท

กระบวนการที่ใช้ในงานวิจัยนี้คือกระบวนการพัฒลมและแผ่นเพลท [5] ซึ่งเป็นกระบวนการหนึ่งที่มีความควบคุมได้ยากเนื่องจากเป็นกระบวนการที่มีความไว และมีความแปรปรวนสูง ความแปรปรวนนี้เกิดขึ้นจากตัวกระบวนการเองและสิ่งรบกวนภายนอก ซึ่งกระบวนการนี้เหมาะสำหรับการทดสอบสมรรถนะของตัวควบคุมเป็นอย่างมาก ลักษณะของกระบวนการแสดงดังรูปที่ 3.1

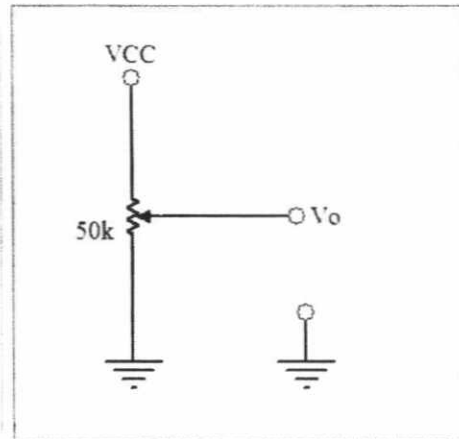
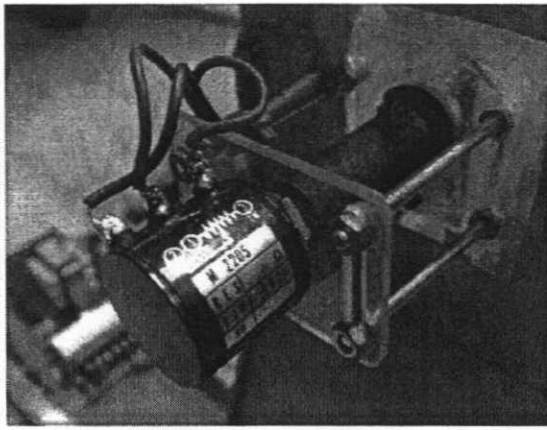


รูปที่ 3.1 กระบวนการพัฒลมและแผ่นเพลท

กระบวนการพัฒลมและแผ่นเพลทประกอบไปด้วย แผ่นเพลท มอเตอร์และพัฒลม อุปกรณ์ตรวจวัด ชุดขับเคลื่อนมอเตอร์ ไมโครคอนโทรลเลอร์ ชุดเพาเวอร์ซัพพลาย การ์ดอินเตอร์เฟซ และคอมพิวเตอร์

โดยที่แผ่นเพลทมีความกว้าง 30 ซม. ยาว 60 ซม. และหนา 2 มม. ฐานรองมีขนาดความกว้าง 35 ซม. ยาว 70 ซม. หนา 5 มม. แกนหมุนอยู่สูงจากพื้น 40 ซม. มอเตอร์และพัดลมที่ใช้ในการควบคุมตำแหน่งของแผ่นเพลทสามารถเคลื่อนที่ปรับตำแหน่งได้ ขนาดและน้ำหนักของแผ่นเพลทเป็นปัจจัยที่สำคัญในการควบคุมกระบวนการนำหนักจะต้องอยู่ในย่านที่สามารถควบคุมได้ไม่หนักหรือเบาจนเกินไป ซึ่งรายละเอียดของอุปกรณ์แต่ละตัวจะอธิบายในหัวข้อต่อไป

### 3.2.1 อุปกรณ์ตรวจวัด (Potentiometer)

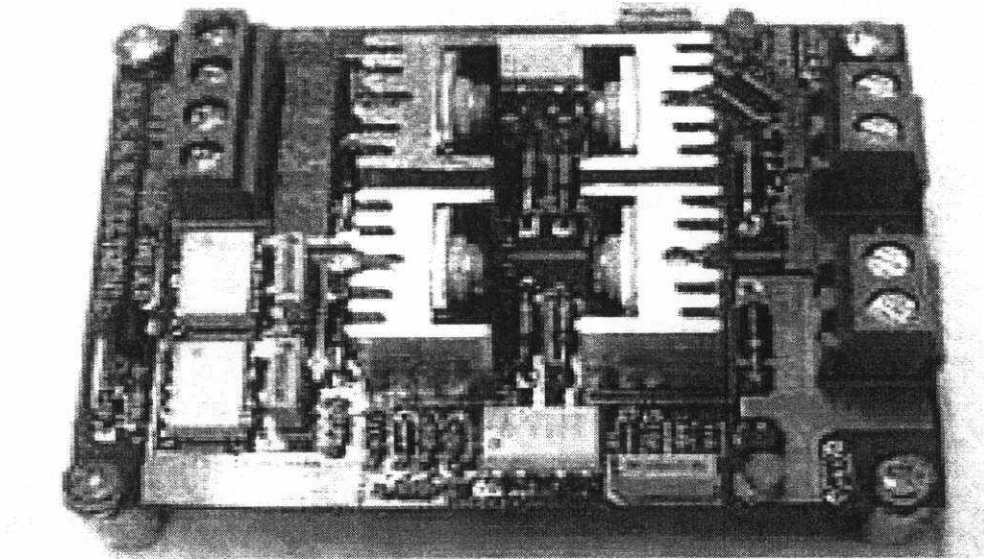


รูปที่ 3.2 อุปกรณ์ตรวจวัด (Potentiometer)

กระบวนการนี้ได้นำเอา Potentiometer มาใช้ในการวัดการเคลื่อนที่เชิงมุมของแผ่นเพลท โดยใช้หลักการของวงจรแบ่งแรงดัน (Voltage Divider) โดยต่ออุปกรณ์วัดเข้ากับแกนหมุนของแผ่นเพลท เมื่อแผ่นเพลทเคลื่อนที่ไป จะทำให้ค่าความต้านทานของ Potentiometer เปลี่ยนแปลงทำให้แรงดันเปลี่ยนแปลงไป Potentiometric ที่ใช้เป็นตัวต้านทานแบบปรับค่าได้มีค่าความต้านทาน  $50\text{k}\Omega$  ปรับความละเอียดได้ 10 รอบ จุดนี้เองทำให้อัตราการเปลี่ยนแปลงแรงดันเกิดขึ้นได้น้อย จึงจำเป็นต้องขยายสัญญาณให้มีขนาดที่มากขึ้น (0 – 5 V) ด้วยวงจรขยายสัญญาณ (Signal Converter) ก่อนต่อเข้ากับพอร์ต ADC ของการ์ด PCI-1711

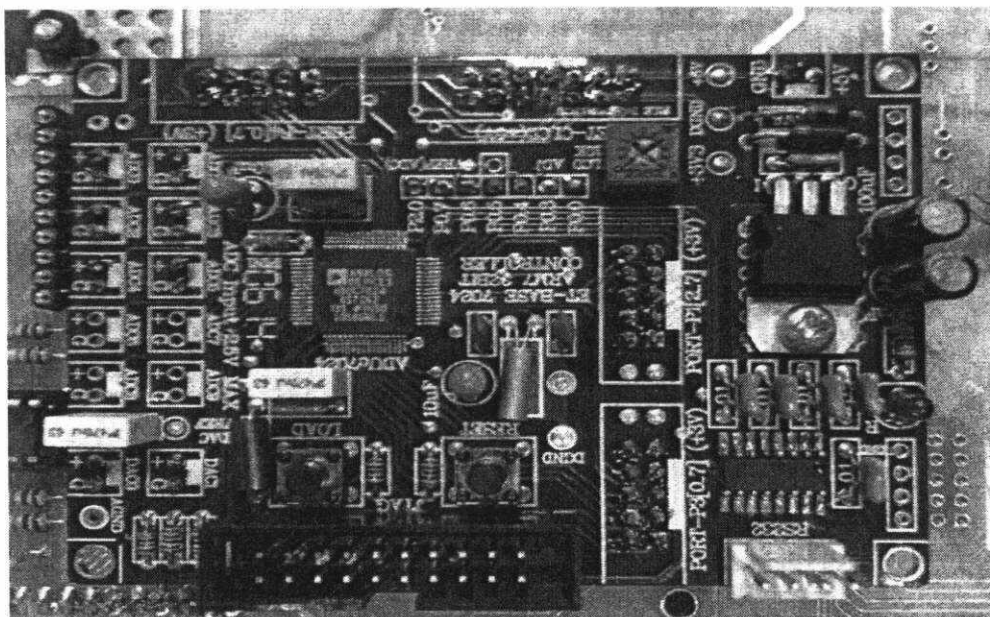
### 3.2.2 Motor Drive

ชุดขับมอเตอร์กระแสตรง ที่ใช้ในการวิจัยเป็นชุดขับมอเตอร์ด้วยมอสเฟทขนาด 12 – 24 V ขับกระแสได้ 40 A สัญญาณควบคุมอินพุทของชุดขับมอเตอร์นี้เป็นแบบ Pulse With Modulation (PWM) ที่ค่า Duty Cycle 0-100 % และจะให้เอาต์พุท 0-12 V หรือ 0-24 V ขึ้นอยู่กับขนาดของ Power Supply



รูปที่ 3.3 ชุดไดร์ฟขับเคลื่อนมอเตอร์

### 3.2.3 Microcontroller ARM7



รูปที่ 3.4 ไมโครคอนโทรลเลอร์ ARM7

บอร์ด ARM7 ที่ใช้งานวิจัยนี้เป็นบอร์ดที่มีไมโครคอนโทรลเลอร์ ARM7 เบอร์ AD $\mu$ C7024 ของบริษัท Analog Device ต่อใช้งาน ซึ่งบอร์ดนี้มีหน้าที่เพื่อรับและส่งสัญญาณระหว่างคอนโทรลเลอร์กับกระบวนการ โดยสถาปัตยกรรมของ ARM7 เป็นซีพียูแบบ RISC ขนาด 32 บิต ภายในมีบัสขนาด 32 บิต ตัวเดียวที่ใช้สำหรับรับข้อมูลและคำสั่ง ชุดคำสั่งจะมีขนาด 32 บิต คงที่ ในขณะที่ข้อมูลสามารถเลือกได้ว่า จะมีขนาด 8,16 หรือ 32 บิต

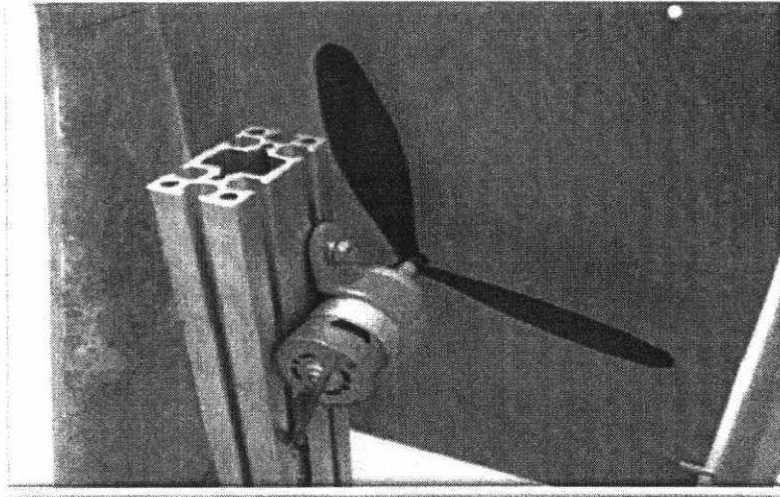


ในรูปที่ 3.6 แสดงองค์ประกอบของบอร์ดไมโครคอนโทรลเลอร์ ADuC7024 ซึ่งมีรายละเอียดดังต่อไปนี้

- หมายเลข 1 คือ ขั้วต่อแหล่งจ่ายไฟเลี้ยงวงจรของบอร์ด ใช้กับแหล่งจ่ายไฟตรง +5VDC
- หมายเลข 2 คือ Port-LCD ชนิด Character Type ใช้การเชื่อมต่อแบบ 4 บิต ผ่าน Port-P4[0..6]
- หมายเลข 3 คือ ตัวต้านทานสำหรับปรับค่าความสว่างให้ LCD
- หมายเลข 4 คือ Port-P4 มี ขนาด 8 Bit คือ P4[0..7]
- หมายเลข 5 คือ ขั้วต่อ ADC จำนวน 10 ช่อง คือ ADC[0..9]
- หมายเลข 6 คือ จุดต่อแรงดันอ้างอิงให้ ADC จากภายนอก
- หมายเลข 7 คือ ขั้วต่อ DAC จำนวน 2 ช่อง คือ DAC[0..1]
- หมายเลข 8 คือ จุดต่อแรงดันอ้างอิงให้ DAC จากภายนอก
- หมายเลข 9 คือ MCU เบอร์ ADuC7024 ซึ่งเป็น MCU ตระกูล ARM7TDMI จาก Analog Device
- หมายเลข 10 คือ Crystal ค่า 32.768 KHz
- หมายเลข 11 คือ Port-P0 มี ขนาด 6 Bit คือ P0[0,3,4,5,6,7] และ Port-P2 มี 1 บิต คือ P2[0]
- หมายเลข 12 คือ Port-P1 มี ขนาด 6 Bit คือ P1[2..7]
- หมายเลข 13 คือ Port-P3 มี ขนาด 8 Bit คือ P3[0..7]
- หมายเลข 14 คือ Switch RESET ใช้สำหรับ Reset การทำงานของ CPU
- หมายเลข 15 คือ Switch LOAD ใช้ร่วมกับ RESET สำหรับสั่ง Download HEX ให้ MCU
- หมายเลข 16 คือ ขั้วต่อ ARM-JTAG สำหรับ Debug โปรแกรม
- หมายเลข 17 คือ ขั้วต่อ RS232 สำหรับใช้งานทั่วไป และ Download HEX ให้ MCU
- หมายเลข 18 คือ LED Power ใช้แสดงสถานะของแหล่งจ่ายไฟ +5VDC

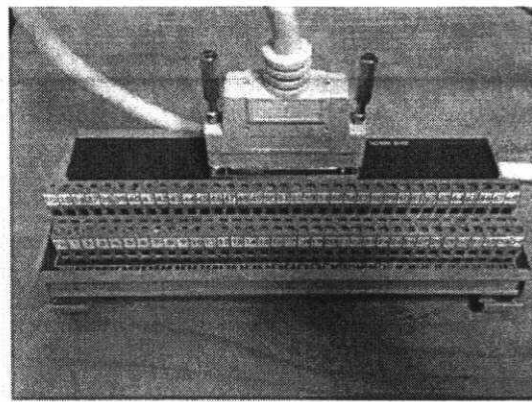
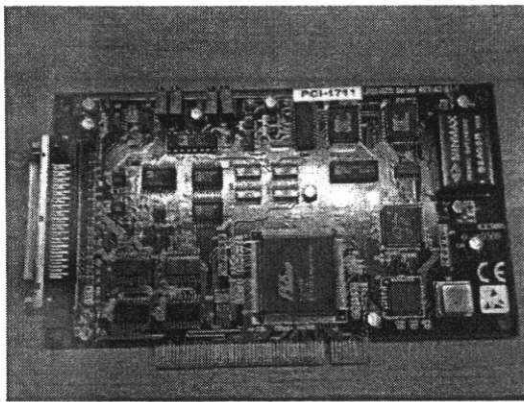
#### 3.2.4 มอเตอร์และพัดลม

มอเตอร์เป็นองค์ประกอบที่สำคัญสำหรับกระบวนการ มอเตอร์ที่ใช้เป็นมอเตอร์กระแสตรงแบบไม่มีแปรงถ่าน ขนาดแรงดันอินพุต 12-24 Vdc กระแส 3 A กำลังงานอินพุต ทำงานที่ความเร็วสูงสุด 20000 รอบต่อนาที และใบพัดที่ใช้มี 3 ใบ ขนาดเส้นผ่าศูนย์กลาง 8 inch เป็นอุปกรณ์สำหรับควบคุมตำแหน่งของพัดลมและแผ่นเพลท



ภาพที่ 3.7 แสดงมอเตอร์และใบพัด

### 3.2.5 การ์ดแปลงสัญญาณ PCI-1711



ภาพที่ 3.8 การ์ด PCI-1711 และคอนเน็กเตอร์

การ์ด PCI-1711 เป็นการ์ดแบบแปลงสัญญาณอะนาล็อกเป็นสัญญาณดิจิทัล และสามารถแปลงสัญญาณดิจิทัลเป็นสัญญาณอะนาล็อกได้ เป็นการ์ดของบริษัท Advantech ประกอบไปด้วย A/D, D/A, Timer/Counter ซึ่งคุณสมบัติของแต่ละตัวอธิบายได้ดังนี้

#### - ส่วนแปลงสัญญาณ ADC

ส่วนแปลงสัญญาณ ADC นี้จะทำหน้าที่ในการรับสัญญาณแรงดันจากอุปกรณ์กรองความถี่ และแปลงให้อยู่ในรูปของสัญญาณดิจิทัล เพื่อส่งให้กับคอมพิวเตอร์หรือตัวส่งสัญญาณต่อไปโดยส่วนแปลงสัญญาณ ADC นี้จะมีจำนวนช่องในการรับสัญญาณถึง 16 ช่อง มีความละเอียด 12 บิต มีความเที่ยงตรงสูง  $\pm 5\%$  และมีความเร็วในการเปลี่ยนแปลงสัญญาณสูงสุด  $100\text{ KHz}$  นอกจากนั้นยังสามารถเลือกช่วงของสัญญาณอินพุตที่จะรับเข้ามาได้หลายช่วง

### - ส่วนการแปลงสัญญาณ DAC

ส่วนแปลงสัญญาณ DAC นี้จะทำหน้าที่ในการรับสัญญาณควบคุมจากคอมพิวเตอร์หรือตัวควบคุม และแปลงให้อยู่ในรูปของสัญญาณอนาล็อกเพื่อส่งให้กับชุดขับเคลื่อนของปั้มน้ำต่อไป โดยส่วนแปลงสัญญาณ DAC นี้จะมีจำนวนช่องในการส่งสัญญาณ 2 ช่อง มีความละเอียดสูงถึง 12 บิต มีความเป็นเชิงเส้นมากถึง  $\pm 0.5$  บิต และมีความเร็วในการแปลงสัญญาณสูงถึง  $10 \mu s$  นอกจากนี้แล้วยังสามารถเลือกช่วงของสัญญาณเอาต์พุตที่จะส่งออกไปได้หลายแบบ การ์ด PCI-1711 นำมาใช้เพื่อทำหน้าที่รับส่งสัญญาณในรูปของแรงดันไฟฟ้าระหว่างคอมพิวเตอร์กับกระบวนการ

### - คุณสมบัติของการ์ดแปลงสัญญาณ PCI-1711

#### Analog Input

Channels:	16 Single-Ended
Resolution:	12-bit
On-board FIFO:	1K Samples
Conversion Time:	$10 \mu s$
Input Range:	$\pm 10V, \pm 5V, \pm 10V, \pm 2.5V, \pm 1.25V, \pm 0.625V$
Maximum Input Over Voltage :	$\pm 15V$
Maximum Sampling Rate:	100 kHz

#### DC Accuracy:

Monotonicity:	12 bits
Offset error:	Adjustable to Zero
Gain error:	0.005% FSR (Gain = 1)

#### AC Accuracy:

SNR:	68 dB
ENOB:	11 bits
Input Protect:	30 Vp-p
Input Impedance:	2M ohm, 5pF

**Analog Output**

Channels:	2
Resolution:	12-bit
Accuracy:	$\pm 1/2$ LSB
Gain error:	Adjustable to Zero
Throughput:	38 kHz (min.)
Slew Rate:	11V / $\mu$ s
Drift:	40 ppm/°C
Output Impedance:	0.81 Ohms
Settling Time:	26 $\mu$ s (to $\pm 1/2$ LSB of FSR)
Output Range:	Software Programmable

**Digital Input**

Channels:	16
Input voltage: Low:	0.8 V max.
High:	2.0 V min.

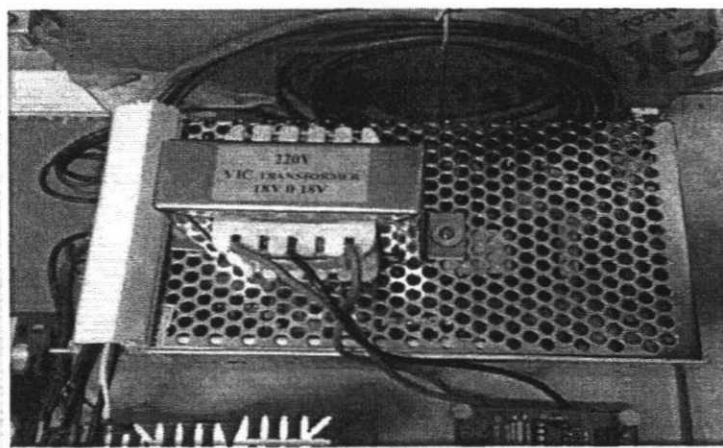
**Digital Output**

Channels:	16
Output Voltage:	Low: 0.8 V max. @ 8.0 mA (sink)
High:	2.0 V min. @ -0.4 mA (source)

**Programmable Timer/Counter**

Channels:	1
Counters:	16 bits
Compatibility:	TTL level
Base Clock:	10 MHz
Max. Input Frequency:	10 MHz

### 3.2.6 เพาเวอร์ซัพพลาย



รูปที่ 3.9 เพาเวอร์ซัพพลาย

เพาเวอร์ซัพพลายที่ใช้ในงานวิจัยนี้เป็นแบบสวิตชิง โดยมีสัญญาณอินพุตเป็นไฟฟ้ากระแสสลับ 220 โวลต์ ส่วนเอาต์พุตเป็นไฟฟ้ากระแสตรงอยู่ในช่วง 12-24 โวลต์ สามารถขับกระแสได้ 12 แอมแปร์ ข้อดีของสวิตชิงเพาเวอร์ซัพพลายเมื่อเปรียบเทียบกับแหล่งจ่ายไฟแบบเชิงเส้น คือมีประสิทธิภาพที่สูง ขนาดเล็ก และน้ำหนักเบากว่าแหล่งจ่ายไฟแบบเชิงเส้น เนื่องจากแหล่งจ่ายไฟแบบเชิงเส้นใช้หม้อแปลงความถี่ต่ำจึงมีขนาดใหญ่ และน้ำหนักมาก ขณะใช้งานจะมีแรงดันและกระแสผ่านตัวหม้อแปลงตลอดเวลา กำลังงานสูญเสียที่เกิดจากหม้อแปลงจึงมีค่าสูง

## 3.3 องค์ประกอบของซอฟต์แวร์

### 3.3.1 โปรแกรม Matlab

สำหรับโปรแกรมที่ใช้ในงานวิจัยนี้คือ โปรแกรม MatLab ซึ่งสามารถใช้ควบคุมได้เป็นกระบวนการได้อย่างดี อีกทั้งยังสามารถทำงานร่วมกับการ์ด ADC และ DAC ของบริษัท Advantech รุ่น PCI-1711 ได้ โปรแกรม Matlab จะถูกใช้ในการจำลองกระบวนการ สำหรับเครื่องมือหรือโปรแกรมย่อยของโปรแกรม MatLab นั้นมีอยู่มาก ผู้วิจัยจึงขออธิบายถึงส่วนประกอบของโปรแกรมที่ใช้เฉพาะในงานวิจัยนี้เท่านั้น ซึ่งรายละเอียดแสดงได้ดังต่อไปนี้

### 3.3.2 โปรแกรม Simulink

Simulink [10] เป็นโปรแกรมที่เป็นส่วนที่เพิ่มเติมเข้ามาในโปรแกรม MATLAB ซึ่งเป็นโปรแกรมที่พัฒนาโดยบริษัท Mathworks โดยเป็นโปรแกรมที่มีความสามารถในการแก้ปัญหาทางคณิตศาสตร์ที่มีประสิทธิภาพสูง ในส่วนของ Simulink นั้นเป็นเครื่องมือที่ใช้ในการเขียนโปรแกรมเพื่อหาคำตอบของสมการทางคณิตศาสตร์โดยใช้ลักษณะการโปรแกรมในรูปแบบของ Block Diagram เป็นหลัก โดยเฉพาะอย่างยิ่งระบบสมการเชิงอนุพันธ์ซึ่งมักจะพบปัญหาในการหา Closed Form Solution หรือ

การหาคำตอบเมื่อเทียบกับเวลาในช่วงเวลาใดๆ และเนื่องจากระบบทางวิศวกรรมโดยส่วนใหญ่จะสามารถอธิบายด้วยระบบสมการเชิงอนุพันธ์ เช่นกันดังนั้น Simulink จะเข้ามามีบทบาทในการทำให้การจำลองการตอบสนองของระบบมีความสะดวกมากยิ่งขึ้น

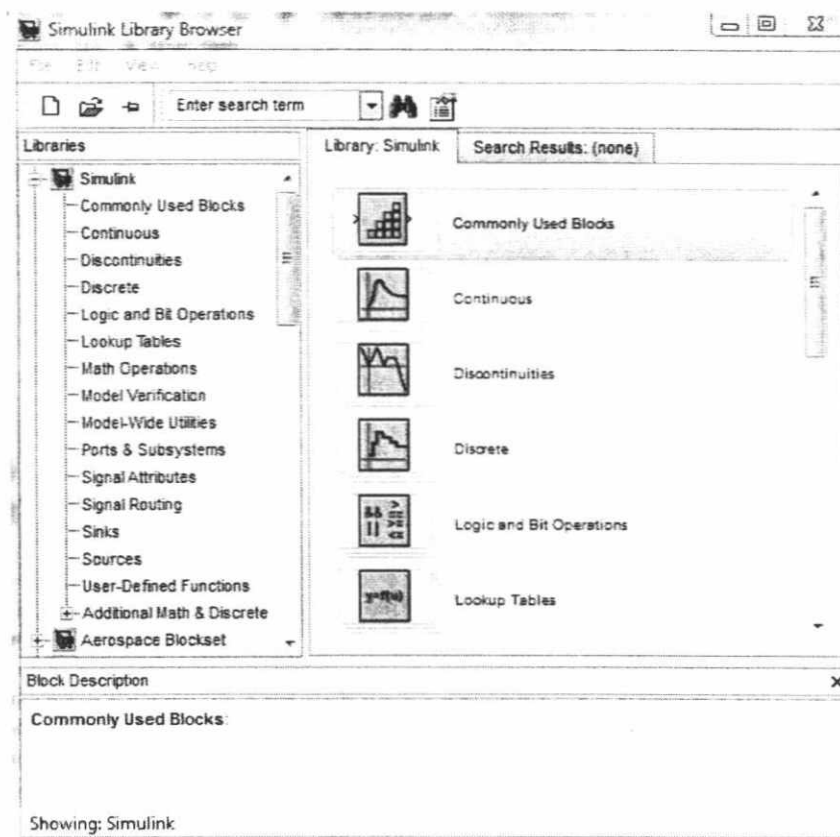
สำหรับองค์ประกอบของโปรแกรมนั้นไม่ว่าจะเป็นวงจรทางตรรกศาสตร์ไปจนถึง ระบบควบคุมสำหรับระบบพลศาสตร์ที่ซับซ้อนจะสามารถสร้างแบบจำลองได้โดยอาศัยเพียง Building Blocks พื้นฐานที่มีมาให้จาก Simulink Libraries นอกเหนือจากนั้นยังมีเครื่องมือใน ส่วนอื่นๆ ที่ในปัจจุบันได้เพิ่มเติมเข้ามาเพื่อให้การใช้งานมีประสิทธิภาพมากยิ่งขึ้น เช่น Fuzzy Logic Toolbox, Neural Network Toolbox, Digital Signal Processing (DSP) Toolbox หรือ อื่นๆ ซึ่งในปัจจุบันพบว่า เครื่องมือ ต่างๆ ที่มีให้ สามารถทำให้การจำลองหรือวิเคราะห์การทำงานของ ระบบสามารถทำได้ตั้งแต่ระบบที่มีความง่าย จนถึงระบบที่มีความซับซ้อนอย่างมากโดยไม่ต้อง มีการเขียนโปรแกรมเพิ่มเติมแต่อย่างใด

- **การจำลองระบบพลศาสตร์ (Dynamic System Simulation) ด้วย Simulink**

ในปัจจุบันคอมพิวเตอร์ได้เข้ามามีส่วนช่วยทำให้การแก้ปัญหาทางวิศวกรรมได้ใช้ประโยชน์จากคณิตศาสตร์อย่างเต็มรูปแบบในการจำลองระบบทางพลศาสตร์ใดๆ โดยไม่ต้องอาศัยการศึกษาหรือทดลองจากระบบทางกายภาพจริงโดยการจำลองการทำงานของระบบพลศาสตร์ (Dynamic System Simulation) ดังกล่าวนั้น ได้รับการพิสูจน์แล้วว่ามีประโยชน์อย่างมหาศาลในการทำงานทางวิศวกรรม โดยเฉพาะอย่างยิ่งในงานทางด้านการออกแบบระบบ (System Design) และงานการออกแบบระบบควบคุม (Controller System Design) ทั้งนี้เนื่องจากการจำลองปัญหาช่วยประหยัดทั้งงบประมาณและเวลา (Prototype) ของระบบทางกายภาพที่สนใจซึ่งในอดีตจะต้องสูญเสียให้กับการสร้างระบบต้นแบบ

- **หลักการของสัญญาณและการไหลของตรรก (Signal and Logic Flow)**

ในโปรแกรม Simulink นั้นการจำลองปัญหาจะอาศัยการสร้างแผนภูมิ (Simulation Diagram) เพื่อการกำหนด การรับ หรือส่งผ่านสัญญาณ หรือข้อมูลที่ต้องการทราบ ในแผนภูมิที่ถูกสร้างขึ้นข้อมูล จากบล็อกประเภทต่างๆ จะถูกส่งผ่านไปสู่อบล็อกอื่นๆ โดยเส้นสัญญาณที่เชื่อมต่อระหว่าง บล็อก นั้นๆ สัญญาณสามารถที่จะถูกสร้างและส่งไปสู่อบล็อกอื่นๆ ทั้งแบบสถิตย์และแบบแปรผันข้อมูลสามารถป้อนให้แก่ฟังก์ชันเพื่อการกระทำทางตรรกศาสตร์หรือทางคณิตศาสตร์ ข้อมูลสามารถที่จะถูกส่งไปสู่แหล่งทิ้งข้อมูล (Sink) ซึ่งสามารถที่จะแสดงให้เห็นในลักษณะของ Oscilloscope หรือ Display หรือแม้กระทั่งเก็บในรูปของไฟล์ (Save to Files) สิ่งที่ต้องพึงระวังในการจำลองปัญหานั้น ข้อมูลต่างๆ จะถูกดำเนินการ รับและส่ง ในลักษณะเป็นช่วงเวลา (Discrete Times) เท่านั้น ดังนั้น Step Time หรือ Simulation Time Step หรือ Integration Time Step ที่กำหนดเพื่อการจำลองปัญหานั้นจะเป็น ส่วนที่สำคัญอย่างยิ่งโดยการเลือกใช้ Step Time ที่เหมาะสมจะถูกกำหนดโดยระบบย่อยที่มีการ ตอบสนองเร็วที่สุดในแผนภูมิที่กำหนดขึ้น



รูปที่ 3.10 ลักษณะของโปรแกรม Simulink

ลักษณะการเขียนโปรแกรมด้วย Simulink จะเป็นลักษณะการเขียนโปรแกรมโดยใช้บล็อกไดอะแกรมของฟังก์ชันต่างๆ ซึ่งสิ่งสำคัญที่ควรระวังสำหรับการเขียนโปรแกรมในลักษณะนี้คือการกำหนดคุณสมบัติให้กับแต่ละบล็อกนั้นจะต้องถูกต้องไม่เช่นนั้นจะไม่สามารถรันโปรแกรมได้

- การเชื่อมต่อบล็อก

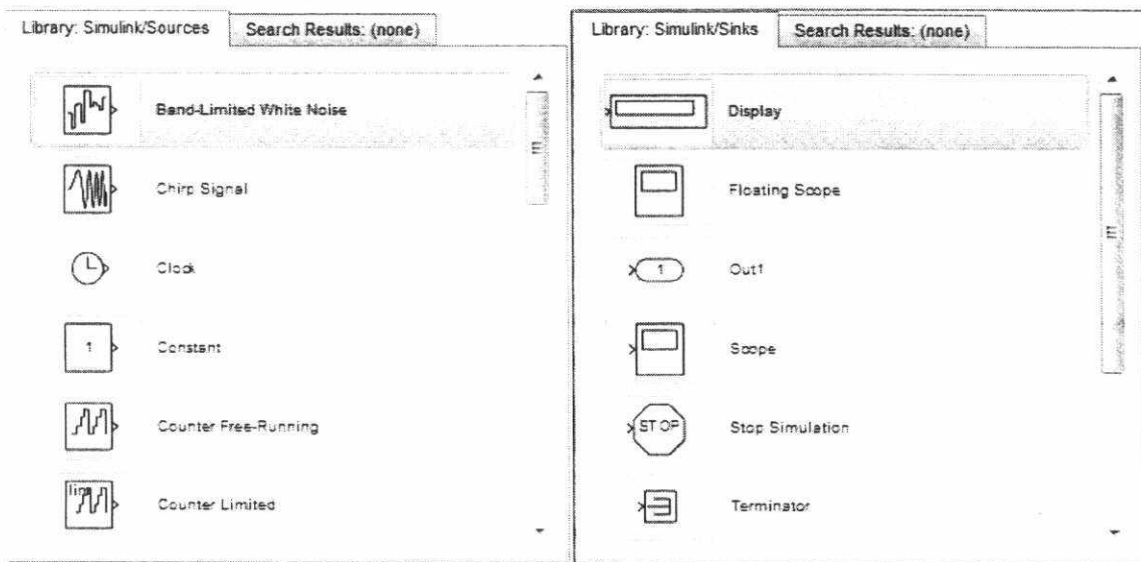
ในการเชื่อมต่อบล็อกบน Simulation Diagram นั้น ให้กดเมาส์ปุ่มซ้ายที่ Output ของบล็อกหนึ่งแล้วค้างปุ่มเอาไว้โดยเคอร์เซอร์ของเมาส์จะเปลี่ยนรูปเป็นเครื่องหมายบวก หลังจากนั้นลาก (Drag) เมาส์ (จะเห็นเส้นประเกิดขึ้นและวิ่งตามเครื่องหมายบวกดังกล่าว) จากบล็อกหนึ่ง ไปสู่ Input ของบล็อกที่ต้องการจะเชื่อมต่อ เมื่อปล่อยเมาส์ปุ่มซ้ายที่ Input ของอีกบล็อกหนึ่งแล้วเส้นประจะเปลี่ยนเป็นเส้นทึบแสดงถึงการเชื่อมต่อกันของบล็อกทั้งสอง

- กลุ่มของบล็อกที่ใช้งานบ่อยในเบื้องต้น

- แหล่งกำเนิดสัญญาณ (Sources) และแหล่งทิ้งสัญญาณ (Sinks)

กลุ่มบล็อกแหล่งกำเนิดสัญญาณ (Sources) ที่กำหนดมาให้ ประกอบด้วยแหล่งกำเนิดข้อมูลหรือสัญญาณ ที่ใช้บ่อยในการวิเคราะห์ระบบทางพลศาสตร์ เช่น แหล่งกำเนิดค่าคงที่ (Constant) แหล่งกำเนิดสัญญาณคลื่นไซน์ (Sinusoidal Wave) แหล่งกำเนิดสัญญาณแบบ Step หรืออื่นๆ

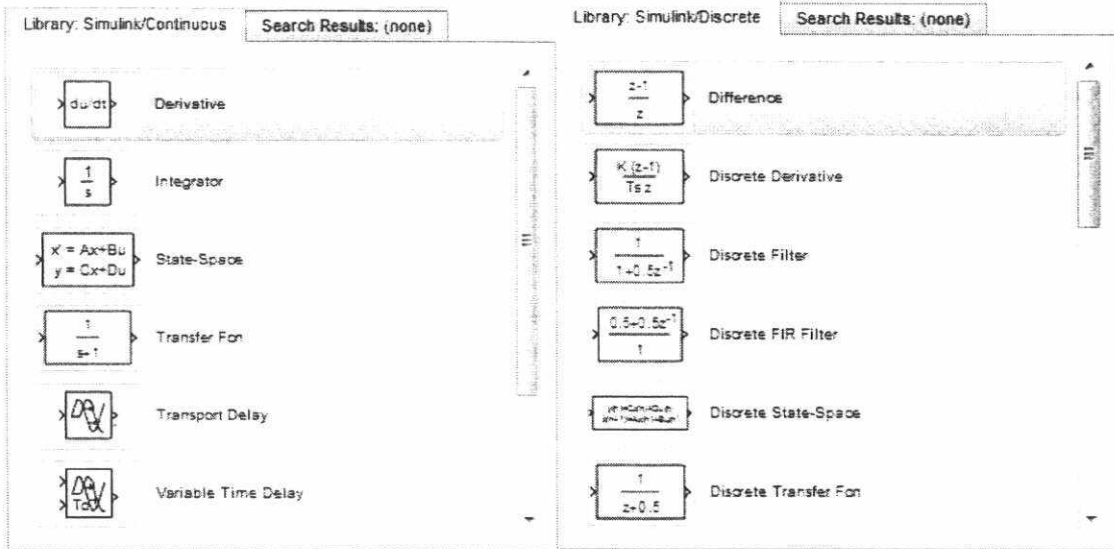
นอกจากนี้ยังมีแหล่งกำเนิดสัญญาณที่ทำหน้าที่เหมือน Signal Generator รูปที่ใช้ในการทดลองทางวิศวกรรมอื่นๆ หรือแหล่งกำเนิดสัญญาณนาฬิกา (Clock) ที่มักจะใช้บ่อยเพื่อ จับเวลาของการจำลอง ปัญหา หรืออื่นๆ กลุ่มบล็อกแหล่งทิ้งสัญญาณ (Sinks) เป็นกลุ่มบล็อกที่มักจะใช้ในการเก็บสัญญาณที่ต้องการมากกว่าการทิ้งสัญญาณอย่างที่ตั้งชื่อกำหนด โดยทั่วไปแล้วบล็อกที่จะใช้บ่อยคือบล็อกของ Scope ประเภทต่างๆ เพื่อตรวจสอบลักษณะการเปลี่ยนแปลงของสัญญาณในจุดใดๆ ของระบบที่กำลังถูกจำลอง ในขณะที่กำลังทำการจำลอง หรือบล็อกที่เก็บสัญญาณ (To Workspace) เพื่อใช้ในการดำเนินการอื่นๆ ที่พื้นที่ทำงานของ MATLAB รูปที่ 3.11 แสดงบล็อกต่างๆ ของทั้งแหล่งกำเนิด และแหล่งสัญญาณ ของ Simulink V.7.6



รูปที่ 3.11 แสดงบล็อกต่างๆ ของแหล่งกำเนิดสัญญาณ (Sources) และแหล่งสัญญาณ (Sinks)

- ระบบต่อเนื่อง (Continuous) และระบบไม่ต่อเนื่อง (Discrete)

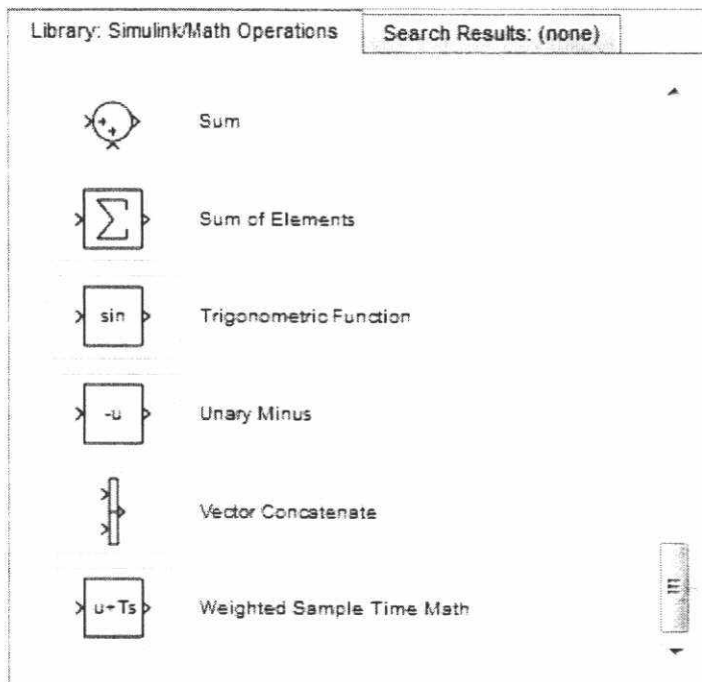
กลุ่มบล็อกของระบบต่อเนื่อง (Continuous System) หรือระบบไม่ต่อเนื่อง (Discrete System) เป็นอีกกลุ่มหนึ่งที่มีการใช้ในการจำลองปัญหาบ่อย ระบบพลศาสตร์ทุกระบบ สามารถที่จะถูกวิเคราะห์ในลักษณะของระบบต่อเนื่อง (Continuous System) หรือระบบไม่ต่อเนื่อง (Discrete System) ซึ่งไม่ว่าระบบจะถูกจำลองในลักษณะใด โปรแกรม Simulink นั้นสามารถ ตอบสนองต่อ ลักษณะรูปแบบของระบบได้ในหลายลักษณะ เช่น Transfer Function หรือ Integration Blocks หรือ Delay Blocks หรืออื่นๆ รูปที่ 3.12 แสดงลักษณะบล็อกที่ใช้ในการกำหนด ระบบในลักษณะต่างๆ ของ ทั้งระบบต่อเนื่อง (Continuous System) หรือระบบไม่ต่อเนื่อง (Discrete System) ของ Simulink



รูปที่ 3.12 แสดงบล็อกต่างๆ ภายในกลุ่มบล็อกของระบบต่อเนื่อง (Continuous System) และระบบไม่ต่อเนื่อง (Discrete System)

- การดำเนินการทางคณิตศาสตร์ (Math Operations)

กลุ่มบล็อกการดำเนินการทางคณิตศาสตร์เป็นอีกกลุ่มหนึ่งที่มีการใช้ในการจำลองปัญหาบ่อย กลุ่มบล็อกนี้จะประกอบไปด้วยบล็อกย่อยๆ ที่มีหน้าที่เกี่ยวข้องกับการดำเนินการต่างๆ ทางคณิตศาสตร์ เช่น การนำสัญญาณมาบวกกัน (Sum) การคูณสัญญาณด้วยค่าคงที่ (Gain) หรืออื่นๆ รูปที่ 3.13 แสดงลักษณะบล็อกย่อยในกลุ่มบล็อกการดำเนินการทางคณิตศาสตร์ ของ Simulink



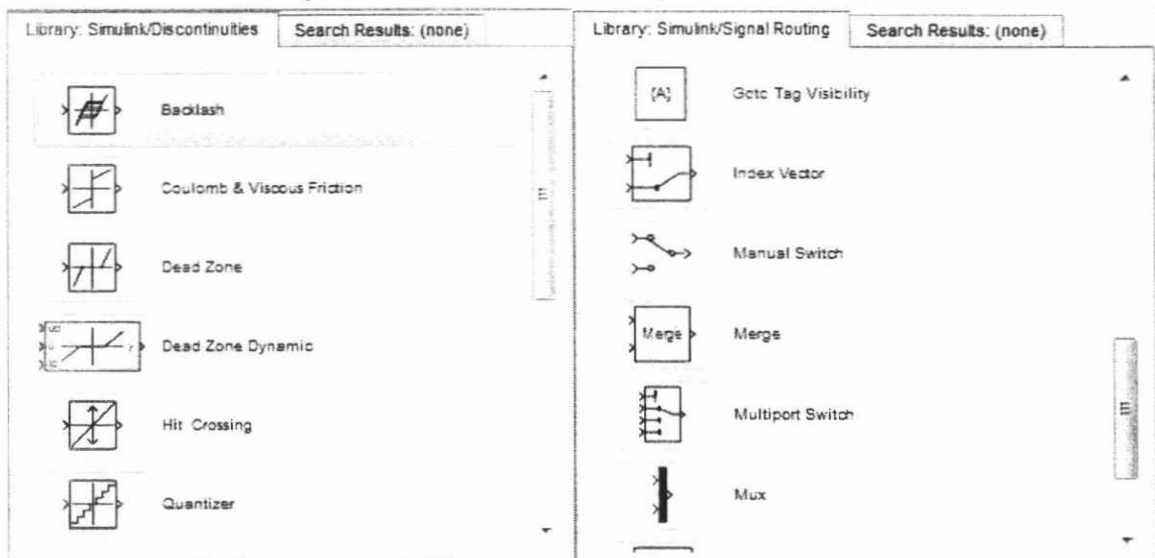
รูปที่ 3.13 แสดงบล็อกต่างๆ ภายในกลุ่มบล็อกการดำเนินการทางคณิตศาสตร์ (Math Operations)

### - ความไม่ต่อเนื่อง (Discontinuities)

ข้อได้เปรียบข้อหนึ่งของโปรแกรม Simulink คือความสามารถในการจำลองระบบแบบไม่เป็นเชิงเส้น (Nonlinear System Simulation) ซึ่งในการหาคำตอบด้วยมือ หรือการหา Closed Form Solution โดยทั่วไปจะเป็นการยากฟังก์ชันหรือลักษณะของความไม่เป็นเชิงเส้นที่พบบ่อยคือ Saturation, Signum, Limited Slew Rate หรืออื่นๆ รูปที่ 3.14 แสดงกลุ่มบล็อก Discontinuities ซึ่ง แสดงความไม่ เป็นเชิงเส้นซึ่งสามารถใช้ในการจำลองปรากฏการณ์ทางวิศวกรรมได้หลายเหตุการณ์ ยกตัวอย่างเช่นการใช้ Saturation บล็อกมักจะใช้แทนข้อจำกัดทางกายภาพเช่นขีดจำกัดของแรงดันทางเข้ามอเตอร์ไฟฟ้า หรือระดับแรงดันสูงสุดในระบบไฮดรอลิกส์ บล็อก Switches เป็นอีก บล็อกหนึ่งที่ใช้บ่อย เป็นบล็อกที่ใช้ ในการจำลองปัญหา IF-THEN เช่นเดียวกับที่พบในการเขียน โปรแกรมโดยทั่วไป บล็อก Slew Rates ใช้ ในการจำกัดอัตราการเปลี่ยนแปลงของค่าทางกายภาพ เช่น ความเร็วของมอเตอร์ไฟฟ้ากระแสตรง เป็นต้น

### - ทางเดินสัญญาณ (Signal Routing)

ในการจำลองระบบที่มีความซับซ้อน บางครั้งมีความจำเป็นที่จะต้องมีการรับ/ส่ง สัญญาณ จากส่วนหนึ่งของระบบ ไปสู่อีกส่วนหนึ่งของระบบ ซึ่งการใช้เส้นเชื่อมอาจจะไม่สะดวกในการทำงานซึ่ง บล็อกในชุดนี้เป็นประเภท GOTO บล็อกหรือ FROM บล็อกบล็อกประเภท Multiplexingจะมีประโยชน์ ในการจัดระเบียบของแผนภูมิให้มีความเป็นระเบียบมากขึ้นรูปที่ 2.14 แสดงกลุ่มบล็อก Signal Routing

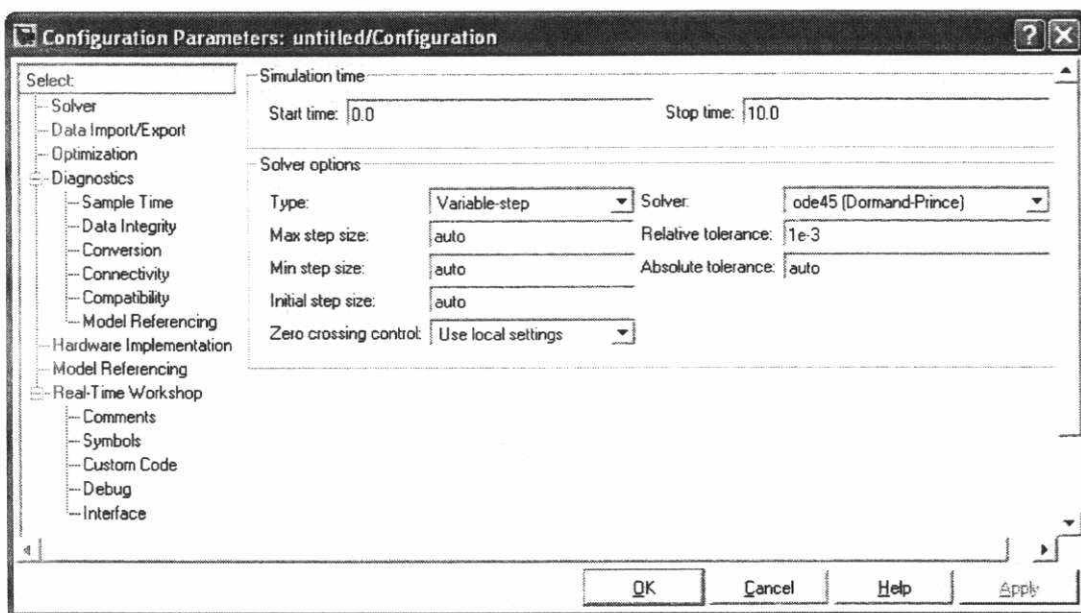


รูปที่ 3.14 แสดงกลุ่มบล็อก Discontinuities และกลุ่ม Signal Routing

### - การกำหนดค่าคงที่ในการจำลองปัญหา (Parameters Configuration)

การจำลองปัญหาบนคอมพิวเตอร์จะต้องอาศัยเทคนิคของระเบียบวิธีเชิงตัวเลข(Numerical Method) ในการแก้ปัญหาและหาคำตอบของสมการเชิงอนุพันธ์ สำหรับโปรแกรม Simulink นั้น ระบบ ในรูปแบบต่างๆ สามารถที่จะจำลองทั้งในแบบของระบบต่อเนื่อง (Continuous System) หรือระบบไม่

ต่อเนื่อง (Discrete System) ขึ้นอยู่กับลักษณะองค์ประกอบของระบบย่อย การจำลองสามารถที่จะถูกกำหนดให้เริ่ม หรือหยุดที่ช่วงเวลา หรือส่วนหนึ่งของเวลาได้ตามกำหนด ข้อสังเกตที่ต้องพึงระวังคือ ระยะเวลาที่สนใจในการจำลองปัญหา กับระยะเวลาที่ใช้ ในการจำลอง ปัญหา หรือระยะเวลาที่คอมพิวเตอร์ใช้ในการหาคำตอบของปัญหาที่สนใจจะไม่ตรงกัน เช่นคอมพิวเตอร์อาจจะใช้ระยะเวลาในการจำลองการตอบสนองในช่วง 10 นาที ของระบบที่สนใจ เพียง 1 วินาที เป็นต้น ในการจำลองแบบปัญหาโดยใช้ระเบียบวิธีเชิงตัวเลขประเภทที่ Step Size มีค่าไม่คงที่นั้น (Variable Step Size) Step Size ที่มากที่สุดและน้อยที่สุดสามารถกำหนดได้ อย่างไรก็ตามระเบียบวิธีเชิงตัวเลขประเภท Step Size คงที่ (Fixed Step Size) เป็นวิธีที่มักจะนิยมใช้ในการจำลองปัญหา ทั้งนี้เนื่องจากสามารถระบุช่วงเวลา หรือจุดของเวลาที่สนใจได้อย่างแม่นยำ Step Size ที่ใช้ในการจำลองปัญหาจะต้องมีการกำหนดและจะมีค่าขึ้นอยู่กับลักษณะของระบบพลศาสตร์ที่สนใจ เช่น ในระบบทางความร้อน Step Size อาจจะอยู่ประมาณ 2 – 3 วินาที ในขณะที่ระบบควบคุมมอเตอร์อาจจำเป็นต้องใช้ Step Size ที่มีค่าน้อยถึง 0.002 – 0.003 วินาที โดยทั่วไป กฎที่ใช้ในการเลือก Step Size จะอยู่ที่ประมาณอย่างน้อย 4 – 20 เท่า ของระบบที่มีการตอบสนองที่เร็วที่สุด



รูปที่ 3.15 แสดงหน้าต่างที่ใช้ในการกำหนดค่าคงที่ในการจำลองปัญหา (Configuration Parameters)

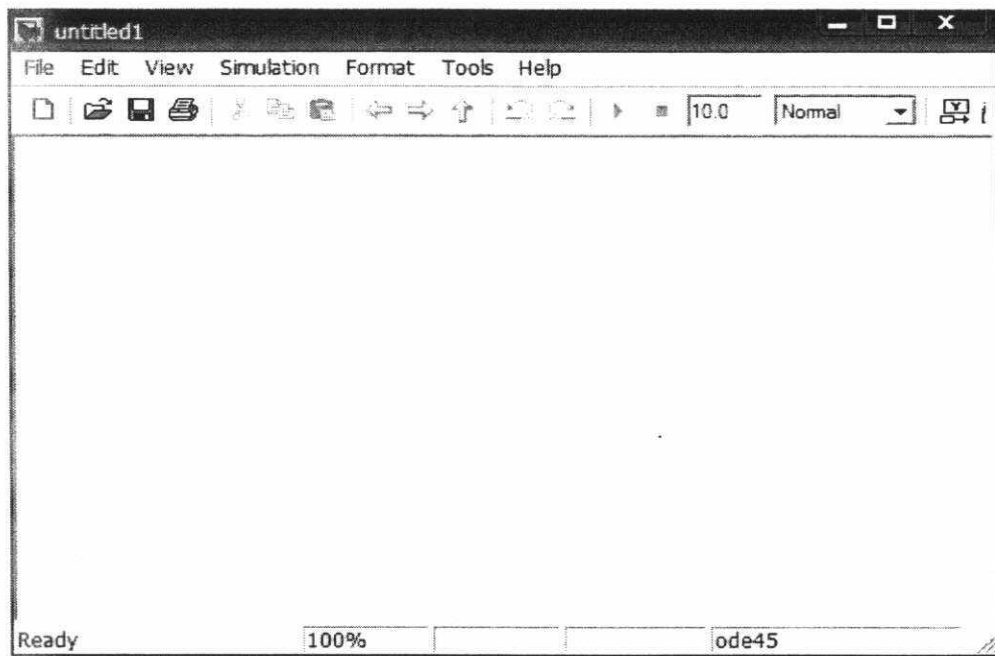
- การสร้างแบบจำลองระบบควบคุม

การสร้างแบบจำลองของระบบควบคุมใน Simulink นั้นต้องทำการสร้างหน้าต่าง สำหรับสร้างแบบจำลองขึ้นมาก่อนซึ่งขั้นตอนในการสร้างแบบจำลองระบบควบคุมทำได้ดังนี้

- เลือก เมนู File --> New --> Model จากนั้นจะปรากฏหน้าต่างขึ้นมาแสดงได้ดังรูปที่

3.16 ซึ่งหน้าต่างนี้จะใช้สำหรับการวางบล็อกที่เลือกมาจาก Simulink Library Browser

-

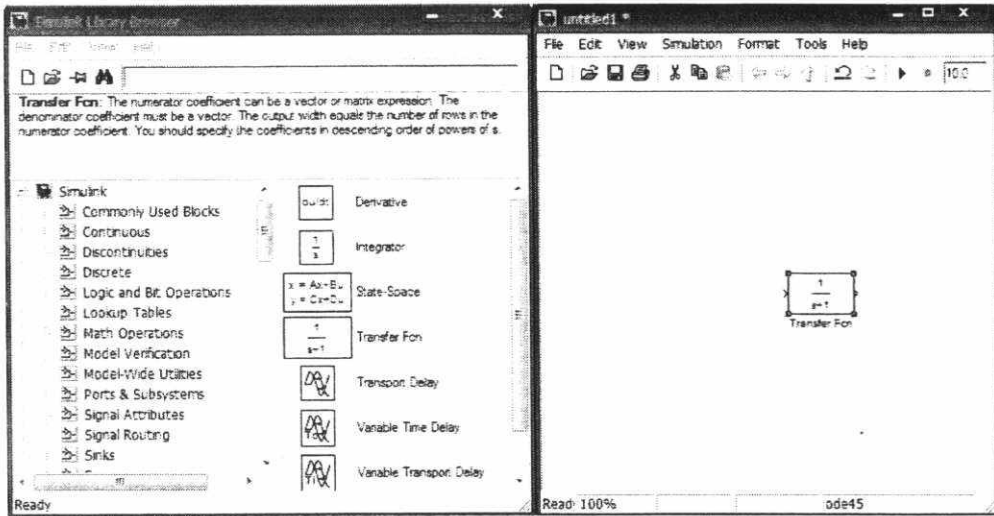


รูปที่ 3.16 หน้าต่างสำหรับสร้างแบบจำลอง

ในส่วนบนของหน้าต่างสำหรับสร้างแบบจำลองจะเป็นเมนูบาร์สำหรับคำสั่งต่างๆ ดังนี้

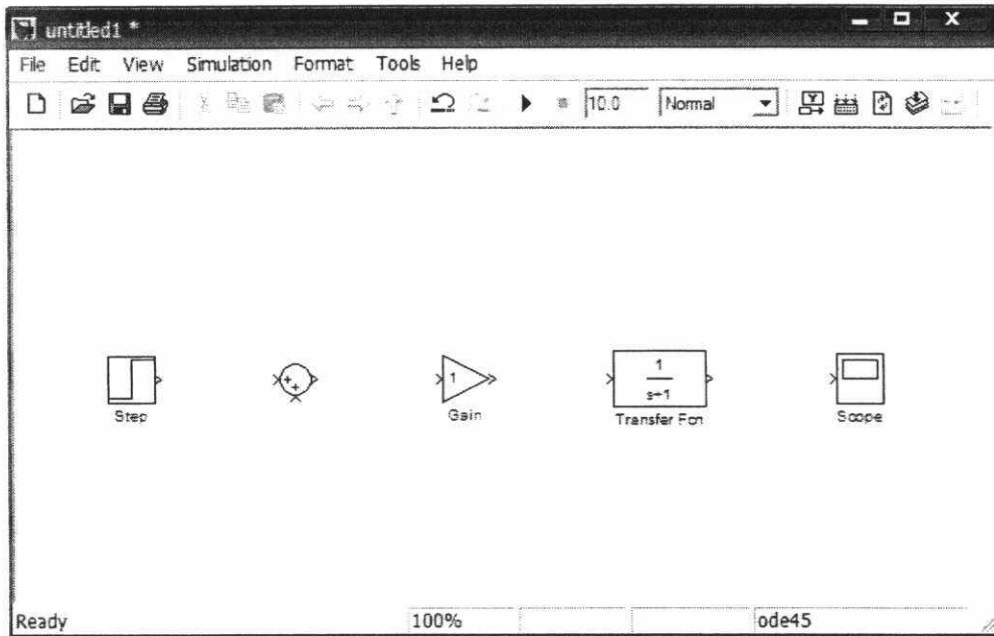
1. เมนู File เป็นคำสั่งเกี่ยวกับการจัดการทามดาวน์แฟ้มข้อมูลทั้งหมด เช่น การเปิดไฟล์ การบันทึกไฟล์ และการกำหนดคุณลักษณะของโปรแกรม Simulink เมนู Edit เป็นคำสั่งเกี่ยวกับการแก้ไขแบบจำลองที่สร้างขึ้นมา
2. เมนู Edit เป็นคำสั่งเกี่ยวกับการแก้ไขแบบจำลองที่สร้างขึ้นมา
3. เมนู View เป็นคำสั่งเกี่ยวกับการแสดงเครื่องมืออุปกรณ์บนหน้าจอแบบจำลอง
4. เมนู Simulation เป็นคำสั่งเกี่ยวกับการจัดการการจำลองการทำงานของระบบควบคุมที่ทำการสร้างขึ้นมา
5. เมนู Format เป็นคำสั่งเกี่ยวกับรูปแบบการแสดงผลบนหน้าจอจำลอง
6. เมนู Tools เป็นคำสั่งเกี่ยวกับการเลือกใช้เครื่องมือต่างๆ
7. เมนู Help เป็นคำสั่งเกี่ยวกับการขอความช่วยเหลือ

- เมื่อทำการสร้างหน้าต่างสำหรับสร้างแบบจำลองแล้วทำบล็อกใน Simulink Library Browser มาวางในหน้าต่างสำหรับสร้างแบบจำลองได้โดยทำการคลิกบล็อกที่ต้องการค้างไว้จากนั้นทำการลากมาไว้ในหน้าต่างสำหรับสร้างแบบจำลองจากนั้นปล่อยปุ่มจะปรากฏบล็อกที่ต้องการบน หน้าต่างสำหรับสร้างแบบจำลองแสดงได้ดังรูปที่ 3.17



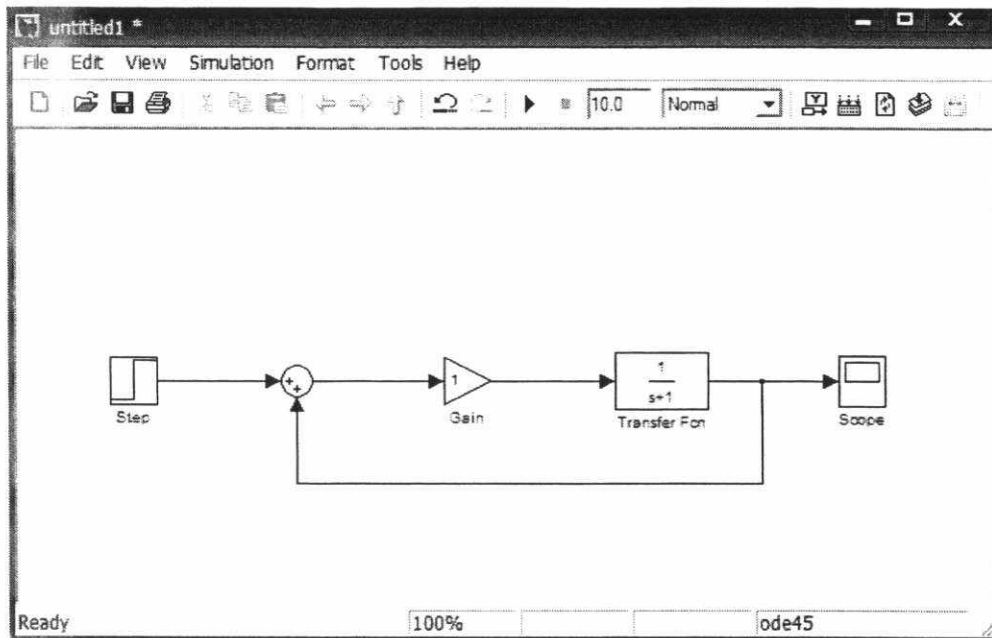
รูปที่ 3.17 การเลือกบล็อกที่ต้องการจาก Simulink Library Browser

- เมื่อได้บล็อกที่ต้องการแล้วเราสามารถนำบล็อกอื่นๆ มาวางบนหน้าต่างสำหรับ สร้างแบบจำลองได้เช่นกัน ต่อไปจะแสดงบล็อกของระบบตัวอย่าง



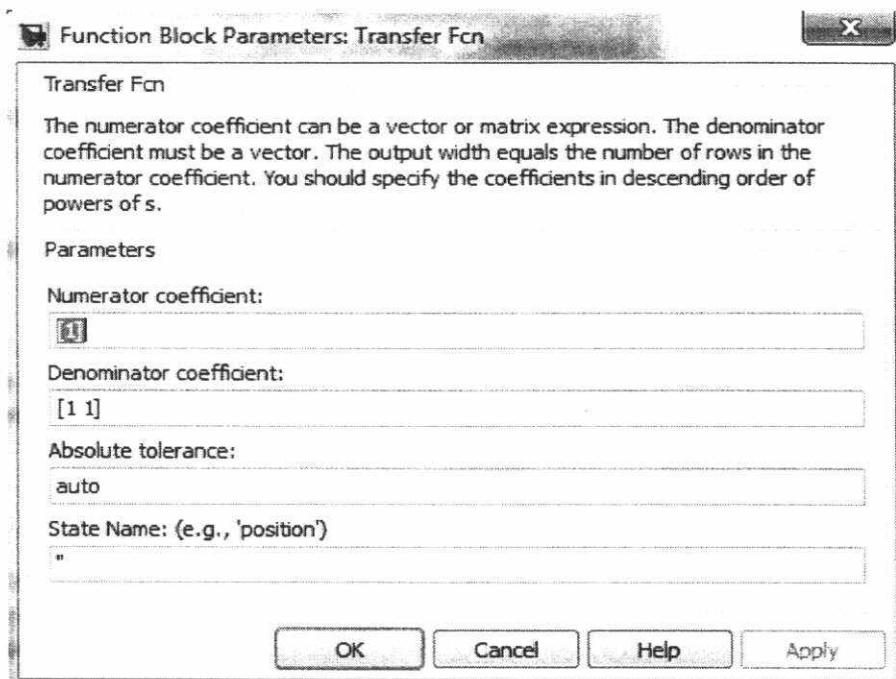
รูปที่ 3.18 บล็อกของระบบตัวอย่าง

- รูปที่ 3.18 สังเกตได้ว่าแต่ละบล็อกจะมีช่องสัญญาณเข้าและสัญญาณออก ไว้สำหรับลากเส้นสัญญาณเชื่อมต่อระหว่างบล็อกต่างๆ เข้าด้วยกันการลากเส้นเชื่อมต่อระหว่างบล็อกต่างๆ นั้นสามารถทำได้โดยเลื่อนเมาส์ให้ไปชี้ที่จุดต่อสัญญาณของบล็อกที่ต้องการ ตัวชี้ของเมาส์จะเปลี่ยนเป็นเครื่องหมายกากบาท จากนั้นคลิกค้างไว้แล้วลากตัวชี้ไปที่จุดต่อสัญญาณของบล็อกที่ต้องการ แสดงได้ดังรูปที่ 3.19



รูปที่ 3.19 การเชื่อมต่อสัญญาณระหว่างบล็อก

- การเปลี่ยนแปลงค่าพารามิเตอร์ของบล็อกสามารถทำได้โดยดับเบิลคลิกที่บล็อกซึ่งจะปรากฏไดอะล็อกสำหรับให้ใส่ค่าพารามิเตอร์ของบล็อก

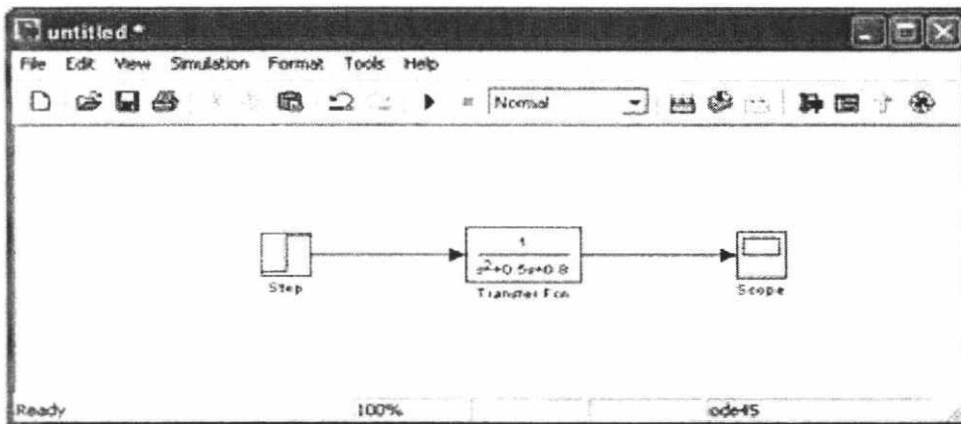


รูปที่ 3.20 การเปลี่ยนแปลงค่าพารามิเตอร์ของบล็อก

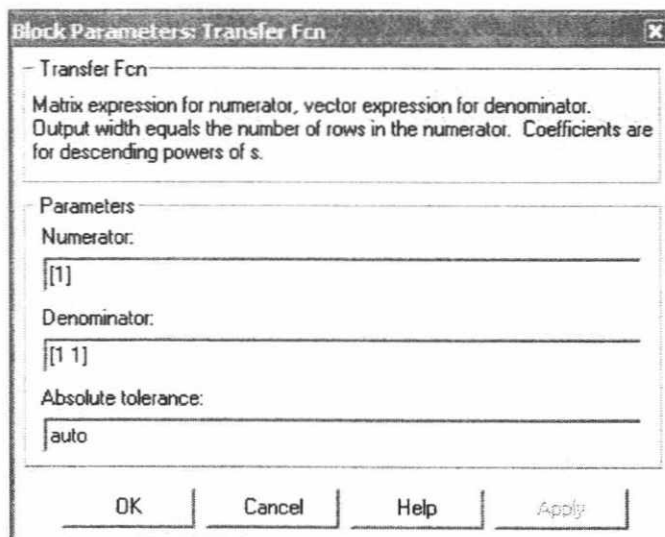
เพื่อให้เข้าใจหลักการทำงานและวิธีการสร้างโปรแกรมด้วย Simulink จะขอแสดงถึงวิธีการสร้างแบบจำลองของระบบควบคุมใน Simulink โดยการสร้างแบบจำลองของกระบวนการใน Simulink นั้นสามารถแบ่งเป็น 3 วิธีด้วยกัน คือ

1. การสร้างแบบจำลองกระบวนการจากสมการ Differential Equation
2. การสร้างแบบจำลองกระบวนการจากทรานส์เฟอร์ฟังก์ชัน
3. การสร้างแบบจำลองระบบควบคุมจากสมการ State-Space

ในงานวิจัยนี้จะใช้วิธีการสร้างแบบจำลองกระบวนการจากทรานส์เฟอร์ฟังก์ชัน วิธีการสร้างแบบจำลองของระบบที่แสดงอยู่ในรูป 3.21 Transfer Function นั้นจะใช้บล็อก Transfer Fcn แสดงแทนระบบ การใส่ค่าโพลีโนเมียลในส่วนของ Numerator และ Denominator นั้นทำได้โดยดับเบิลคลิกที่บล็อกซึ่งจะปรากฏได้อะลือกที่ให้ใส่ค่าขึ้นมาแสดงได้ดังรูปที่ 3.22 โดยการใส่ค่าลงไปนั้นมีเงื่อนไขอยู่ว่า ส่วน Numerator นั้นต้องมีอันดับน้อยกว่า ส่วน Denominator ซึ่งสังเกตได้ว่าการสร้างแบบจำลองของระบบจาก Transfer Function นั้นจะทำการวัดผลตอบสนองได้ที่ส่วนเอาต์พุตของระบบเท่านั้น



รูปที่ 3.21 ได้อะลือกใส่ค่าพารามิเตอร์ของบล็อก Transfer Fcn



รูปที่ 3.22 แบบจำลองระบบทางกลในรูปแบบของ Transfer Function

### 3.3.3 การติดตั้งโปรแกรมเพื่อใช้งาน Real Time Windows Target

- Simulink external model interface

ในหน้าต่าง Command Windows ของโปรแกรม MATLAB ให้พิมพ์คำสั่ง

```
>> mex -setup
```

จากนั้นโปรแกรม Matlab จะแสดงข้อความ

Please choose your compiler for building external interface (MEX) files:

Would you like mex to locate installed compilers [y]/n?

ให้พิมพ์ y เพื่อยืนยัน จากนั้นโปรแกรม Matlab จะแสดงข้อความเพื่อให้เลือก Compiler ที่ต้องการ

Select a compiler:

[1] Lcc-win32 C 2.4.1 in C:\PROGRA~1\MATLAB\R2007a\sys\lcc

[2] Microsoft Visual C++ 2005 in C:\Program Files\Microsoft Visual Studio 8

[3] Microsoft Visual C++ 6.0 in C:\Program Files\Microsoft Visual Studio

[0] None

ในการวิจัยนี้ผู้วิจัยจะเลือก Compiler Microsoft Visual C++ ซึ่งทำได้โดยการพิมพ์ เลข 2

Compiler: 2

โปรแกรม Matlab จะแสดงข้อความเพื่อให้เรายืนยัน Compiler ที่เลือก

Please verify your choices:

Compiler: Microsoft Visual C++ 2005

Location: C:\Program Files\Microsoft Visual Studio 8

Are these correct?([y]/n):

ถ้ายืนยันให้พิมพ์ y แต่ถ้าไม่ยืนยันให้พิมพ์ n จากนั้นโปรแกรมจะอัปเดตข้อมูลของ Compiler และสิ้นสุดการติดตั้ง Compiler โดยสังเกตข้อความในโปรแกรม Matlab ดังตัวอย่างด้านล่าง

Trying to update options file: C:\Documents and Settings\xp\Application

Data\MathWorks\MATLAB\R2007a\mexopts.bat

From template: C:\PROGRA~1\MATLAB\R2007a\bin\win32\mexopts\

msvc80opts.bat

Done . . .

## - การติดตั้ง Real Time Windows Target Kernal

ในหน้าต่าง Command Windows ในโปรแกรม Matlab พิมพ์คำสั่ง

```
>> rtwintgt -install
```

จากนั้นโปรแกรม Matlab จะแสดงข้อความเพื่อเริ่มต้นการติดตั้ง

You are going to install the Real-Time Windows Target kernel.

Do you want to proceed? [y] :

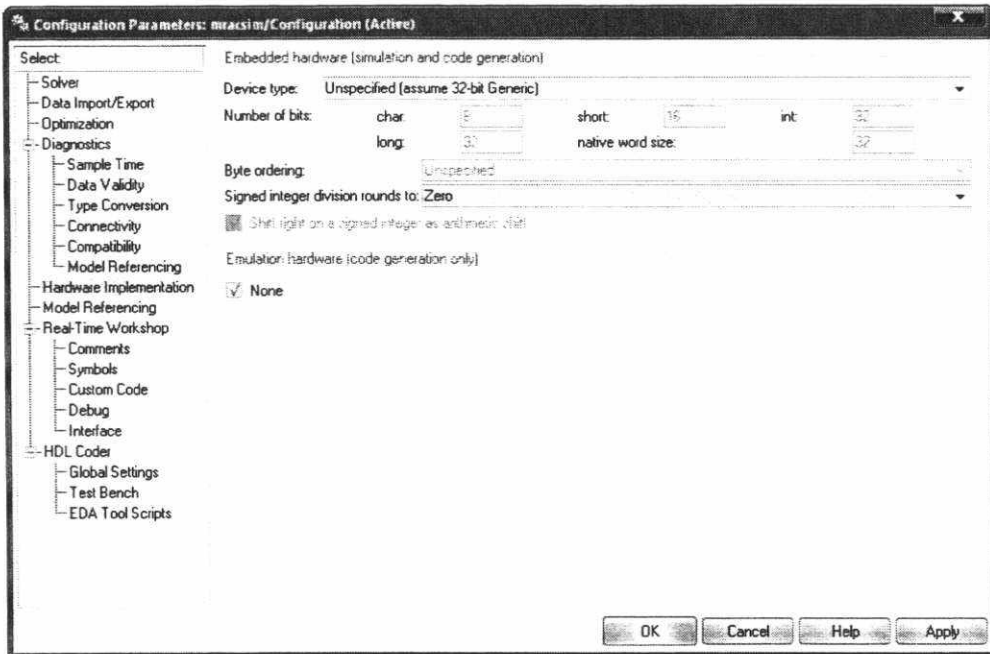
พิมพ์ y เพื่อยืนยันการเริ่มต้นการติดตั้ง Real Time Windows Target Kernal หลังจากนั้นโปรแกรมจะดำเนินการติดตั้ง เมื่อเสร็จเรียบร้อยแล้วโปรแกรมจะแสดงข้อความ

The Real-Time Windows Target kernel has been successfully installed.

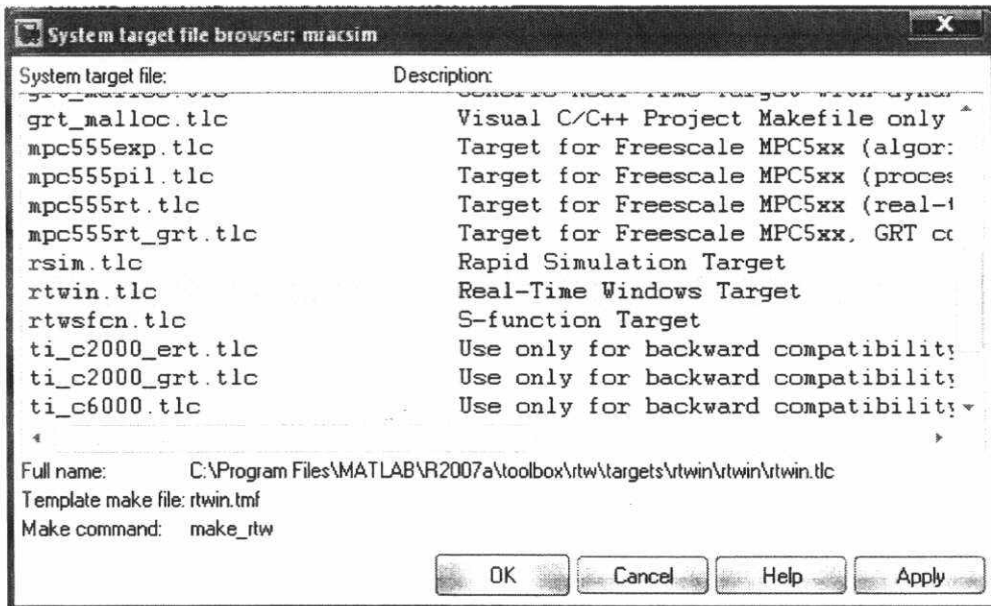
## - การกำหนดค่าพารามิเตอร์ใน Simulink สำหรับ Real Time Workshop

หลังจากที่สร้าง Simulink Model จะต้องกำหนดพารามิเตอร์ในหน้าต่าง Simulation เพื่อให้ Real-Time Workshop สามารถสร้าง Code ภาษา C และสร้าง Real-Time Application ได้ ซึ่งมีขั้นตอนในการกำหนด ดังนี้

1. ในหน้าต่าง Simulink ไปที่ Simulation เมนู คลิกที่ Configuration Parameters
2. คลิกที่หมวดของ Hardware Implementation
3. จากรายการ Device Type เลือก 32-bit Real Time Windows Target
4. Emulation hardware เลือก None
5. คลิกที่หมวด Real Time Workshop
6. Target selection คลิก Browse ที่รายการ RTW system target file เลือก system target file สำหรับ Real Time Windows Target และคลิก Ok
7. คลิก Ok เพื่อยืนยันการเปลี่ยนแปลงและปิด Dialog box



รูปที่ 3.23 Hardware Implementation



รูปที่ 3.24 RTW System Target File

### 3.3.4 โปรแกรม Keil uVision 3.11

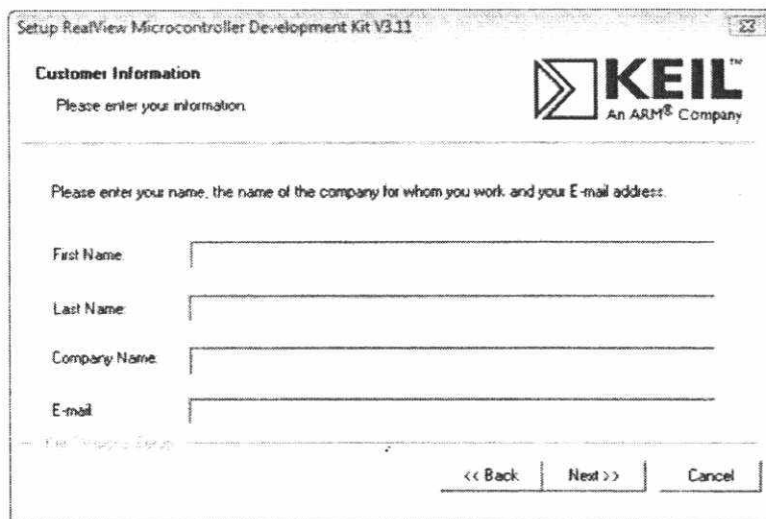
โปรแกรม Keil RealView [11] สามารถดาวน์โหลดได้จากเว็บไซต์ของบริษัท Keil ที่ <http://keil.com> แล้วเลือกที่เมนู Software Download จากนั้นเลือกที่ Evaluation Software โดยก่อนที่จะทำการดาวน์โหลดซอฟต์แวร์จะต้องทำการลงทะเบียนกับทางเว็บไซต์ก่อนจึงจะสามารถดาวน์โหลด

โหลดได้ หลังจากดาวน์โหลดเสร็จสิ้นจะได้ไฟล์ที่มีนามสกุล \*.exe (ในที่นี้คือ rvmdk311.exe) ดับเบิลคลิกที่ไฟล์เพื่อติดตั้งโปรแกรม โดยโปรแกรมจะแสดงหน้าต่างสำหรับติดตั้งโปรแกรมดังรูปที่ 3.25



รูปที่ 3.25 หน้าต่าง Setup Real View Microcontroller Development Kit Version 3.11

เมื่อคลิกปุ่ม Next จะปรากฏหน้าต่าง License Agreement เพื่อแจ้งลิขสิทธิ์ของโปรแกรมคลิกที่  หน้าข้อความ I agree to all terms of proceeding License Agreement แล้วคลิกปุ่ม Next หลังจากนั้นจะมีหน้าต่าง Folder Selection เลือกโฟลเดอร์สำหรับการติดตั้งโปรแกรม ในที่นี้ให้เลือกเป็น C:\Keil ตามที่เครื่องกำหนดมาให้ แล้วคลิก Next จะปรากฏหน้าต่าง customer Information เพื่อกรอกข้อมูลของผู้ใช้งาน ซึ่งควรจะกรอกข้อมูลตามความเป็นจริง โดยเฉพาะในช่องของ e-mail ถ้าไม่กรอกข้อมูลจะไม่สามารถคลิกปุ่ม Next เพื่อดำเนินการในขั้นตอนต่อไปได้ แสดงดังรูปที่ 3.26



รูปที่ 3.26 หน้าต่าง customer Information

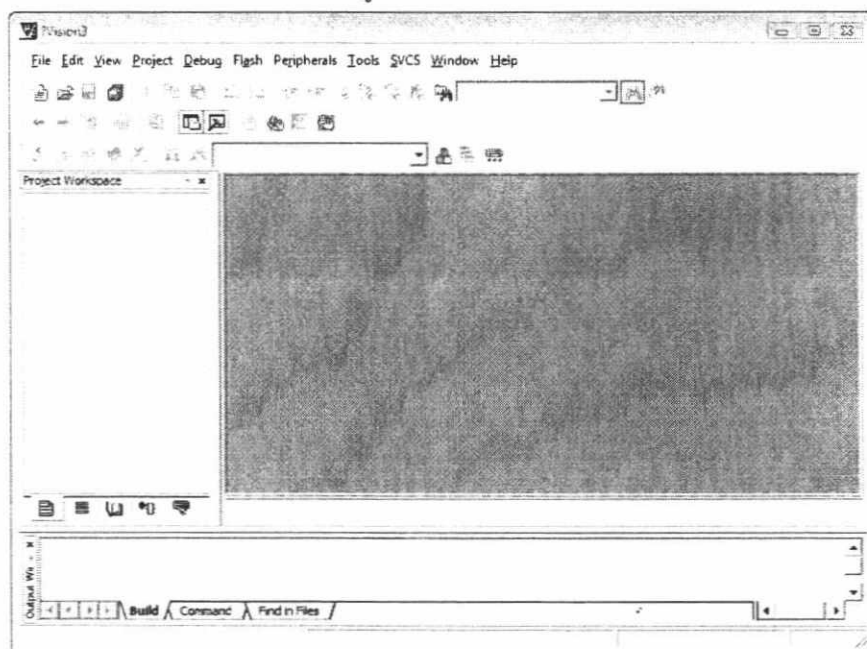
หลังจากนั้นขั้นตอนการติดตั้งโปรแกรมก็จะเริ่มขึ้น หลังจากเสร็จเรียบร้อยแล้วจะปรากฏหน้าต่าง Setup Complete เพื่อเป็นการยืนยันการติดตั้งโปรแกรมให้คลิกที่ปุ่ม Finish เป็นการเสร็จสิ้นการติดตั้งโปรแกรม แสดงดังรูป 3.27



รูปที่ 3.27 หน้าต่างแสดงการสิ้นสุดการติดตั้งโปรแกรม

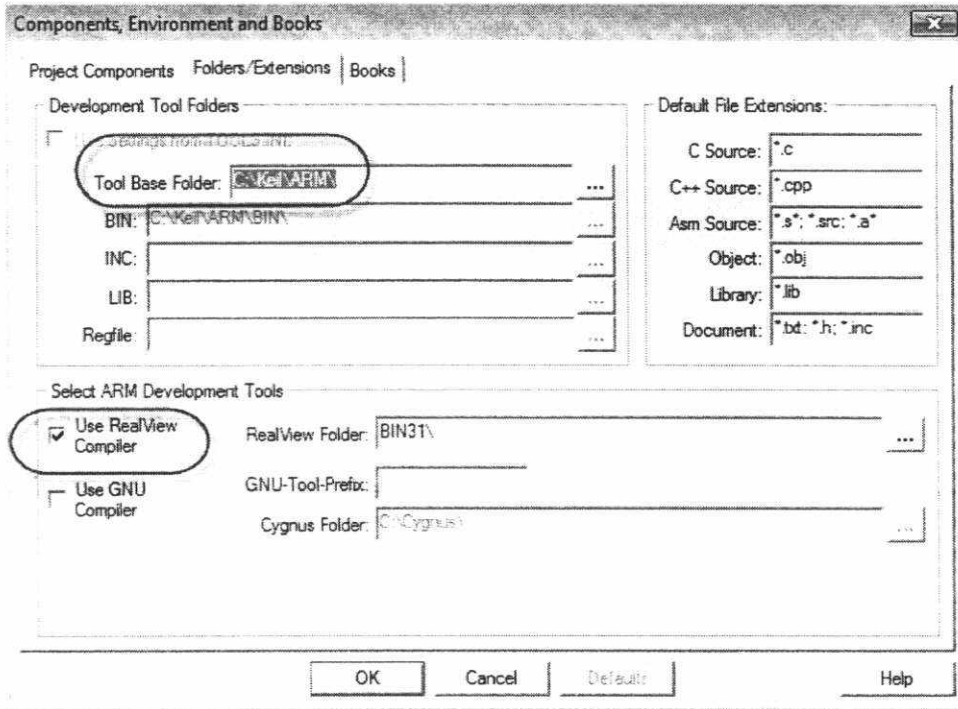
- การเริ่มต้นใช้งานโปรแกรม Keil uVision3

การใช้งานโปรแกรม Keil RealView ในการเขียนโปรแกรมนั้นจะต้องทำการสร้างโปรเจกต์ขึ้นมา และทำการกำหนดค่าการใช้งานให้ถูกต้องเสียก่อนจึงจะสามารถใช้งานโปรแกรมได้ โดยเริ่มต้นจากการเปิดโปรแกรม Keil uVision3 ขึ้นมา แสดงดังรูปที่ 3.28



รูปที่ 3.28 หน้าต่างหลักของโปรแกรม Keil uVision

กำหนดค่าเลือกตัวคอมไพเลอร์ในการแปลคำสั่งให้กับโปรแกรม Keil uVision3 โดยคลิกที่ Project -> Manage -> Components, Environment, Books.... เพื่อเปิดหน้าต่าง Components, Environment and Books ซึ่งมีส่วนที่ต้องกำหนดอยู่ 2 จุดคือ จุดแรกในส่วนของ Select ARM Development Tool จะมีคอมไพเลอร์ให้เลือกอยู่ 2 ตัว ซึ่งในส่วนนี้ให้เลือกคอมไพเลอร์ RealView Compiler และอีกจุดหนึ่งที่ต้องกำหนดคือในส่วนของ Development Tool Folder โดยที่ Tool Base Folder ให้กำหนดไว้ที่ C:\Keil\ARM เสร็จแล้วกดปุ่ม OK



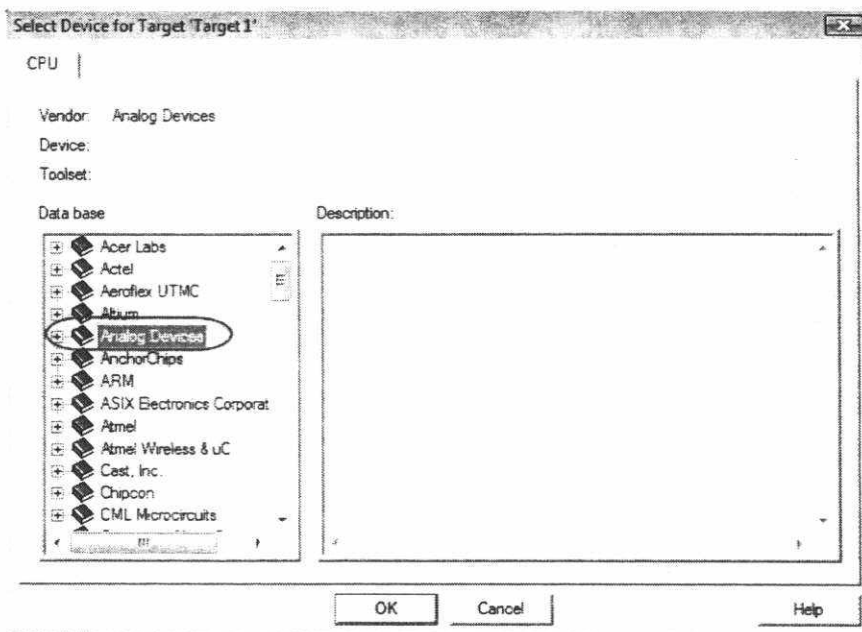
รูปที่ 3.29 หน้าต่าง Components, Environment and Books

ต่อจากนั้นจะต้องทำการสร้างโปรเจ็คใหม่ขึ้นมาโดยเลือกที่ Project -> New uVision Project จะปรากฏหน้าต่าง Create New Project ขึ้นมา แล้วทำการสร้างโฟลเดอร์ใหม่ขึ้นมา แล้วตั้งชื่อโฟลเดอร์ จากนั้นตั้งชื่อโปรเจ็คที่ต้องการ แล้วทำการบันทึกโปรเจ็ค แสดงดังรูปที่ 3.30



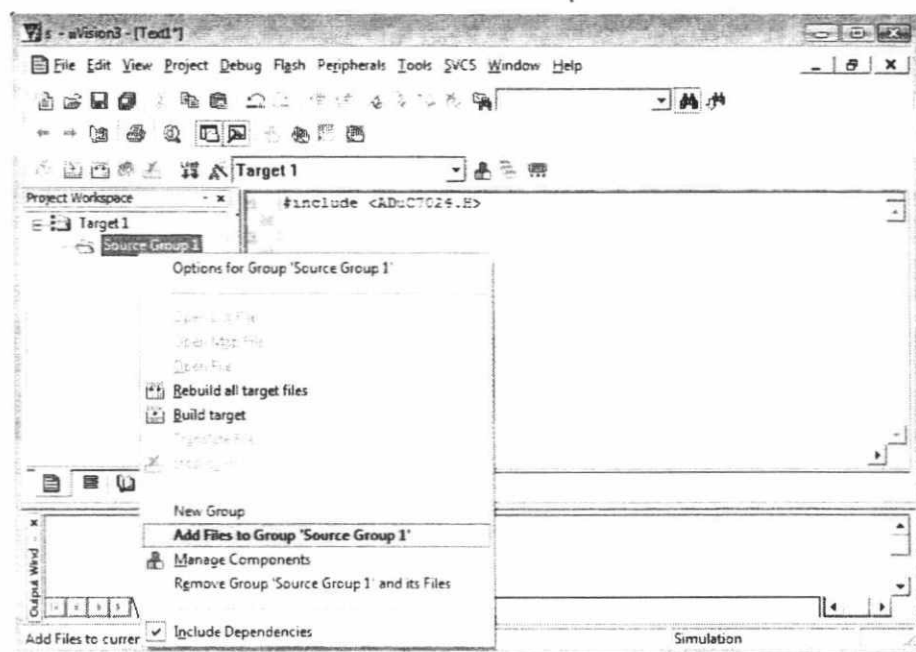
รูปที่ 3.30 หน้าต่าง Create New Project

เมื่อสั่งกดปุ่ม Save แล้วจะปรากฏหน้าต่าง Select Device for Target 'Target 1' เพื่อให้ผู้ใช้เลือก MCU ที่ใช้งาน ซึ่งในที่นี้เลือกไมโครคอนโทรลเลอร์ของบริษัท Analog Devices จากนั้นจะมีรายละเอียดเกี่ยวกับเบอร์ของไมโครคอนโทรลเลอร์ Analog Devices ซึ่งเบอร์ที่ใช้คือ ADuC7024 จากนั้นกดปุ่ม OK โปรแกรมจะถามว่าต้องการที่จะเพิ่มไฟล์ Startup เข้าไปในโปรเจกหรือไม่ (ถ้าต้องการกด YES ถ้าไม่กด NO)



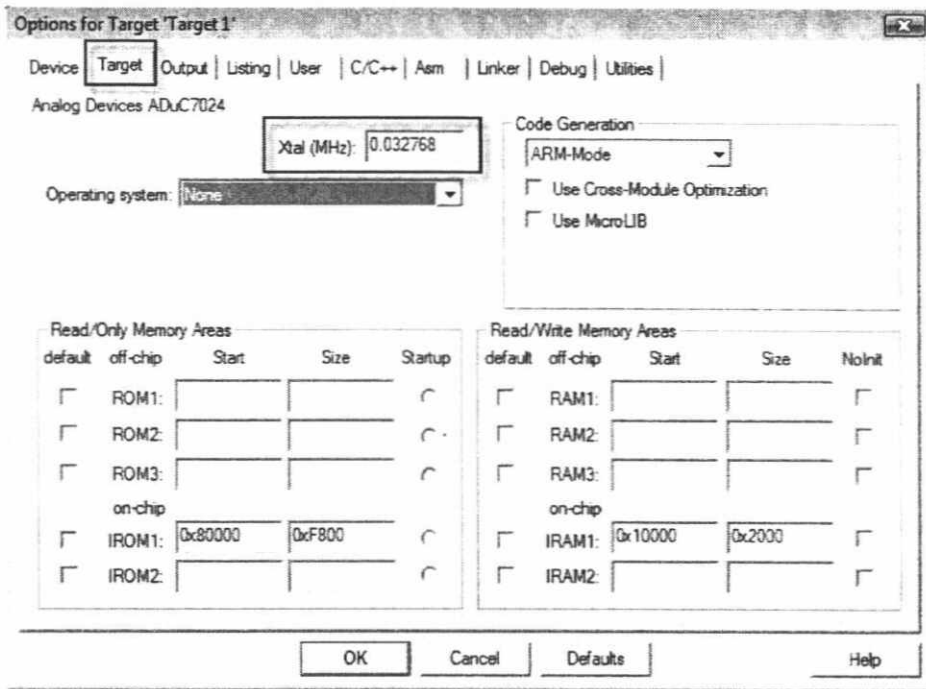
รูปที่ 3.31 หน้าต่าง Select Device for Target

หลังจากขั้นตอนนี้เราก็จะสามารถเริ่มเขียนโปรแกรมได้แล้วโดยเลือกที่ File -> New... ซึ่งจะปรากฏหน้าต่างสำหรับเขียน Text File ขึ้นมา ไปที่หน้าต่าง Text Editor ให้คลิกขวาแล้วเลือก Insert '#include <ADuC7024.H>' ข้อความ #include <ADuc7024.H> จะถูกแทรกในบรรทัดที่ 1 หลังจากเขียนโปรแกรมเรียบร้อยแล้วให้บันทึกไฟล์ที่มีนามสกุล .c



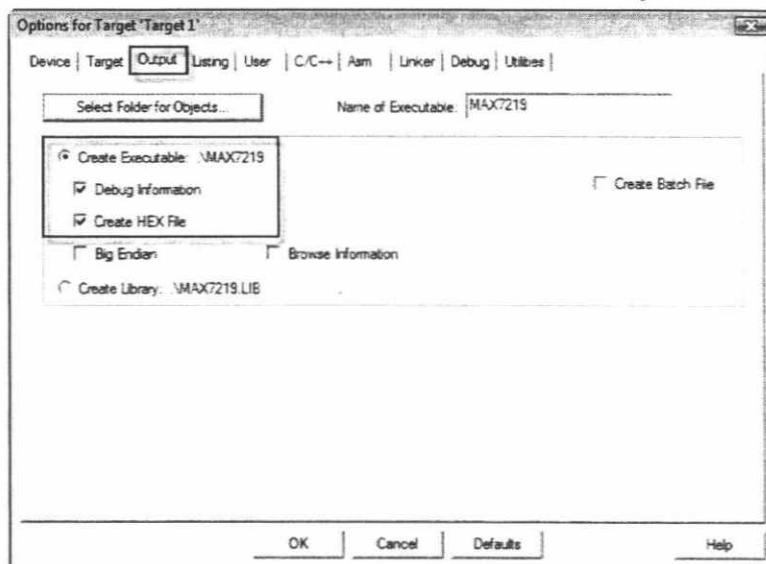
รูปที่ 3.32 หน้าต่าง Add File to Group

ในส่วนของ Project Workspace ในรูปที่ 3.32 จะมีโฟลเดอร์ Target 1 คลิกตรงเครื่องหมายบวกจะมีโฟลเดอร์ย่อยชื่อ Source Group 1 ให้คลิกขวาจะมีเมนูย่อยให้เลือกที่ Add File to Group 'Source Group 1' แล้วเลือกแอดไฟล์โปรแกรม (นามสกุล .c) และไฟล์ Startup เข้ามาในโปรเจกต์ จากนั้นทำการกำหนดค่า Option ของ Project File โดยคลิกปุ่ม Option for Target เพื่อเปิดหน้าต่าง Option for Target 'Target1' เลือกแท็บ Target เพื่อกำหนด MCU Target โดย กำหนดค่า Xtal เป็น 0.032768 ดังรูปที่ 3.33



รูปที่ 3.33 หน้าต่างกำหนด Options for Target

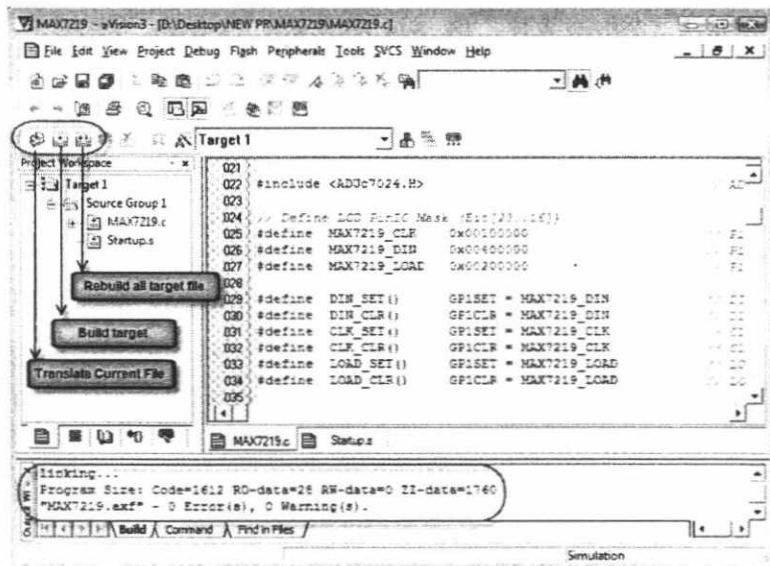
ต่อไปเลือกแท็บกำหนด Output Information โดยเลือกที่ Output แล้วคลิกเลือก Create Executable และเลือกที่ Debug information และ Create HEX file ดังรูปที่ 3.34



รูปที่ 3.34 หน้าต่างกำหนด Output Information

หลังจากที่เขียนโปรแกรมเรียบร้อยแล้วต้องทำการ Compile และ Build โปรแกรมเสียก่อนโดยในรูปที่ 3.35 มีจุดสำหรับ Compile และ Build อยู่ 3 แบบคือ Translate current file, Build target และ Rebuild all target file หลังจากที่ได้ Compile และ Build โปรแกรมเรียบร้อยแล้วในส่วนด้านล่างคือ Output Window ที่ทำหน้าที่รายงานผลของการ Compile โปรแกรม ถ้าโปรแกรมถูกต้องทั้งหมด

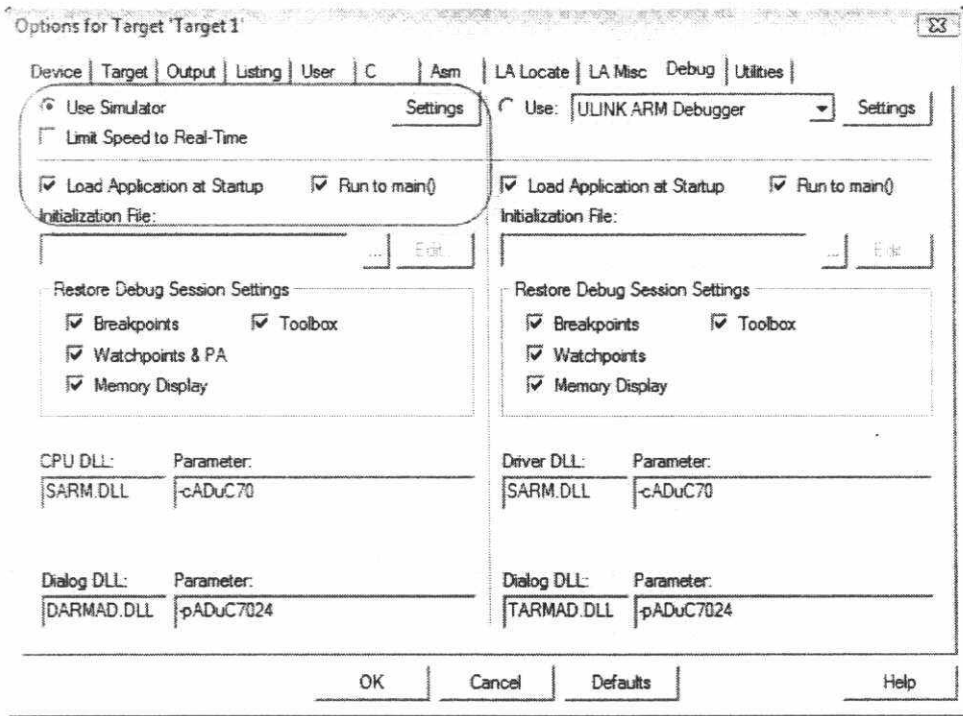
จะต้องขึ้น 0 Error(s), 0 Warning(s) แต่ถ้ามีจุดผิดพลาดโปรแกรมก็จะรายงานว่ามีที่ผิดที่ที่และผิดที่ตรงไหน ซึ่งจะช่วยให้เราสามารถแก้ไขโปรแกรมได้สะดวกขึ้น



รูปที่ 3.35 การ compile และการ Build โปรเจ็ค

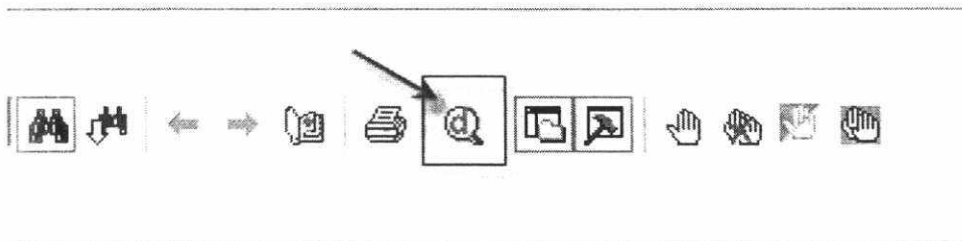
- การ debug โปรแกรมภาษา C สำหรับ ARM7

การดีบัคโปรแกรมเป็นการตรวจสอบการทำงานของโปรแกรมที่เราเขียนขึ้นมา ว่ามีความถูกต้องและเป็นไปตามที่เราต้องการหรือไม่ ซึ่งสามารถตรวจสอบได้เป็นลำดับขั้นตอนอีกทั้งยังตรวจหาจุดผิดของโปรแกรมได้ด้วย ซึ่งหลังจากที่ทำการคอมไพล์ผ่านแล้วให้กำหนดค่าตามรูปที่ 3.36 จากนั้นให้คลิกที่ปุ่มดีบัค (ตามรูปที่ 3.37 หรือเลือกที่เมนู Debug -> Start/Stop Debug Session โปรแกรมก็จะเข้าสู่โหมดดีบัค (โปรแกรมที่ใช้ในการดีบัคเป็นโปรแกรมสำหรับ LED ติดดับโดยพอร์ตที่ใช้เป็นพอร์ต P3.x)



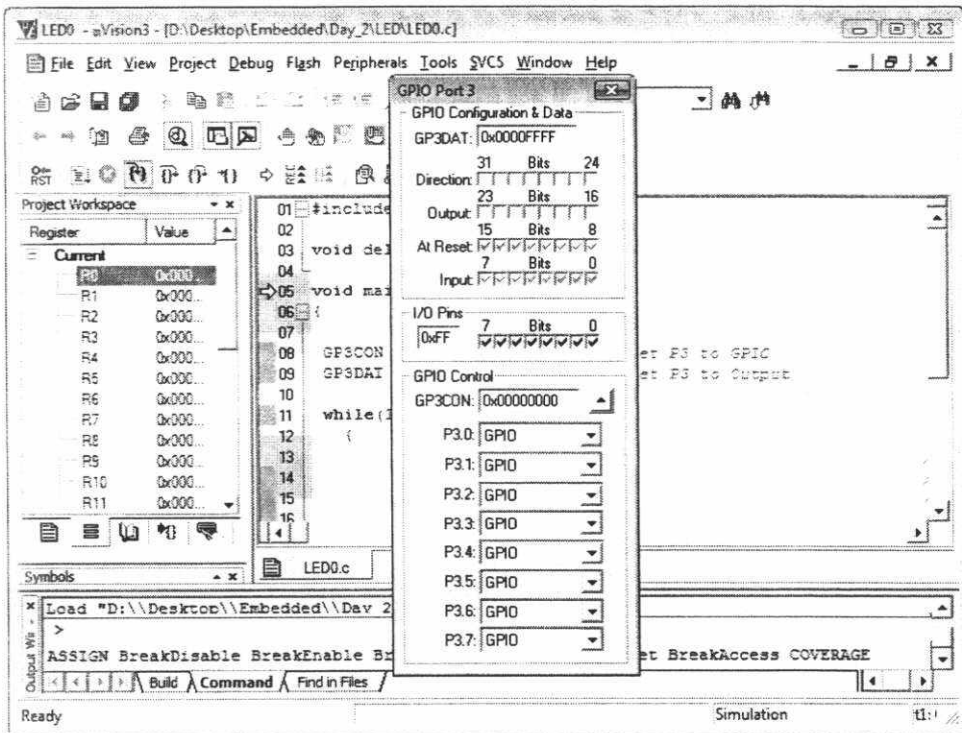
รูปที่ 3.36 ปุ่ม Build Target

เมื่อกำหนดค่าแล้วให้ไปที่ Debug -> Start/Stop Debug Session ดังแสดงในรูปที่ 3.37



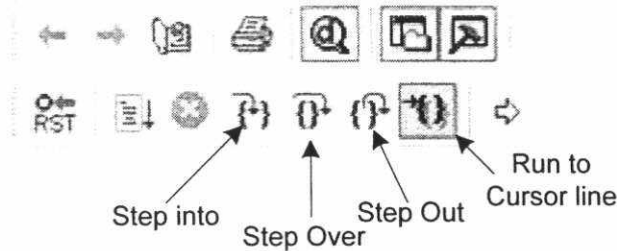
รูปที่ 3.37 การเข้าสู่โหมด Simulation หรือ Debug

หลังจากที่เข้าสู่โหมดดีบัคแล้วให้เลือกที่ Peripherals -> General Purpose Input/Output -> GPIO Port (0-4) ดังรูปที่ 3.38



รูปที่ 3.38 แสดงสถานะเตรียมพร้อม Simulator

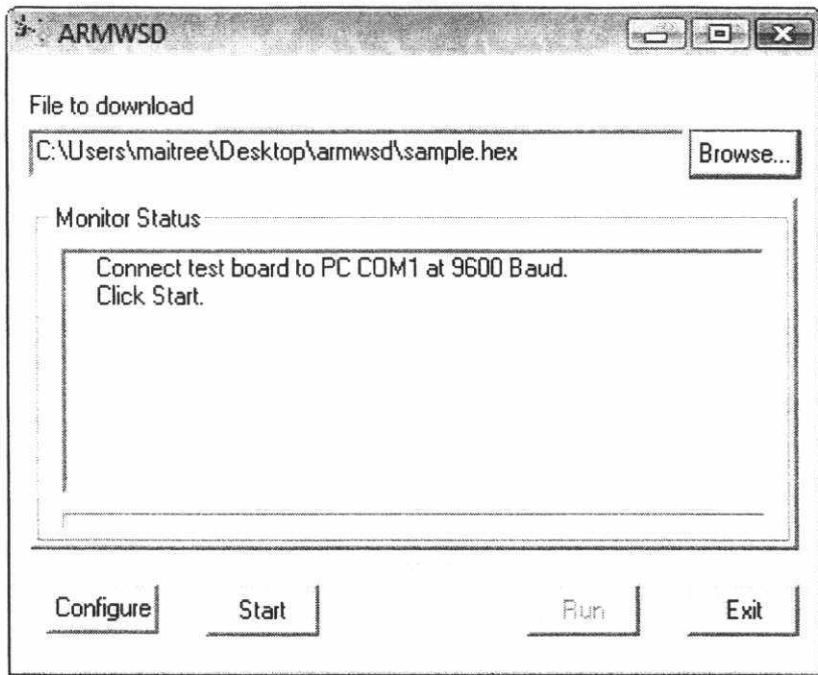
ถึงขั้นตอนนี้ก็สามารถ Run โปรแกรมได้แล้ว โดยไปที่ Debug -> Run หรือกด F5 ดังแสดงในรูปที่ 3.39



รูปที่ 3.39 ปุ่มการ Debug แบบต่างๆ

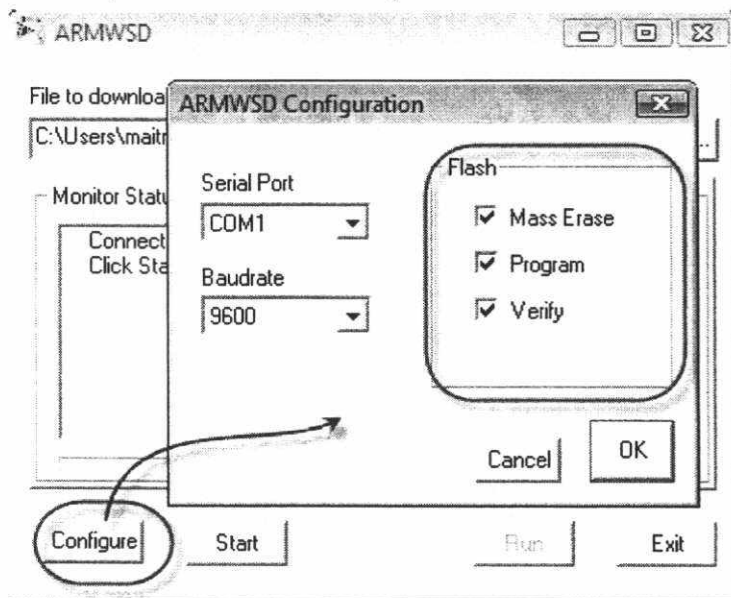
- การติดตั้งซอฟต์แวร์ ARMWSD

หลังจากที่เขียนโปรแกรมภาษา C เสร็จแล้ว ทำการคอมไพล์จะได้ไฟล์ภาษาเครื่องนามสกุล .hex ซึ่งจะต้องใช้ซอฟต์แวร์ ARMWSD เพื่อดาวน์โหลดไฟล์ .hex ที่ได้ลงไปหน่วยความจำแฟลชของไมโครคอนโทรลเลอร์ ARM7 ผ่านสาย RS-232 โดยที่องค์ประกอบของซอฟต์แวร์ ARMSWD จะมีไฟล์ 3 ตัวคือ ARMWSD.exe, armwsd.ini และ sample.hex การใช้งานโปรแกรม ARMWSD นั้นไม่ต้องทำการติดตั้งโปรแกรมก็สามารถใช้งานได้ เพียงแต่ต้องให้ไฟล์ทั้ง 3 ทำงานร่วมกันโดยอาจวางไว้ในโฟลเดอร์เดียวกัน วิธีการใช้งานเพียงแค่ว่าคลิกที่ไฟล์ ARMWSD.exe ก็พร้อมใช้งาน แสดงดังรูปที่ 3.40



รูปที่ 3.40 หน้าต่างการใช้งานของโปรแกรม ARMWSD

การใช้งานโปรแกรมเริ่มจากคลิกที่ปุ่ม Browse แล้วเลือกไฟล์ที่ต้องการดาวน์โหลดไปยังไมโครคอนโทรลเลอร์ ARM7 (ไฟล์ที่มีนามสกุล .hex) จากนั้นคลิกที่ปุ่ม Configure จะมีหน้าต่าง ARMWSD Configuration เพื่อให้กำหนดค่าในการดาวน์โหลด โดยจะมีส่วนที่ต้องกำหนดค่าอยู่ 3 ส่วน คือ Serial Port เลือกพอร์ตให้ตรงกับพอร์ตคอมพิวเตอร์ที่สาย RS-232 ต่ออยู่ในส่วนของ Baud rate ให้เลือกค่า 9600 และในส่วน Flash ให้คลิกเลือกที่ Mass Erase, Program และ Verify แล้วคลิกปุ่ม OK เป็นการเสร็จสิ้นการกำหนดค่า การดาวน์โหลดโปรแกรมทำได้โดยการคลิกที่ปุ่ม Start โปรแกรมก็จะโหลดซอฟต์แวร์ลงไปยังไมโครคอนโทรลเลอร์ แล้วคลิกปุ่ม RUN ไมโครคอนโทรลเลอร์ก็จะเริ่มการทำงาน



รูปที่ 3.41 การ Configuration โปรแกรม ARMWSD ก่อนใช้งาน

### 3.4 บทสรุป

องค์ประกอบทางด้านฮาร์ดแวร์และซอฟต์แวร์เป็นอีกหนึ่งปัจจัยที่สำคัญในการศึกษาระบบใดๆ ซึ่งในบทนี้ผู้วิจัยได้กล่าวถึงองค์ประกอบของฮาร์ดแวร์และซอฟต์แวร์ไว้อย่างละเอียด ในการออกแบบระบบควบคุมหรือออกแบบตัวควบคุมเราจำเป็นต้องรู้องค์ประกอบของฮาร์ดแวร์เป็นอย่างดี เราจึงจะสามารถเลือกใช้อุปกรณ์ต่างๆ เพื่อมาประกอบการเป็นกระบวนการที่เหมาะสมและสามารถที่จะวิเคราะห์หาโมเดลทางคณิตศาสตร์ได้ ส่วนองค์ประกอบด้านซอฟต์แวร์หรือโปรแกรมจะเป็นเครื่องมือที่ช่วยให้กระบวนการด้านฮาร์ดแวร์ทำงานตามที่เราต้องการ ดังนั้นการเลือกใช้ซอฟต์แวร์ก็ต้องพิจารณาถึงความเหมาะสมในการใช้งานร่วมกับกระบวนการ อย่างไรก็ตามฮาร์ดแวร์และซอฟต์แวร์ก็เป็นเพียงหนึ่งปัจจัยเท่านั้น ยังมีปัจจัยอื่นๆ ที่มีความสำคัญกับงานวิจัยนี้ ซึ่งผู้วิจัยจะกล่าวถึงในบทต่อไป

## บทที่ 4

# วิธีการออกแบบตัวควบคุม

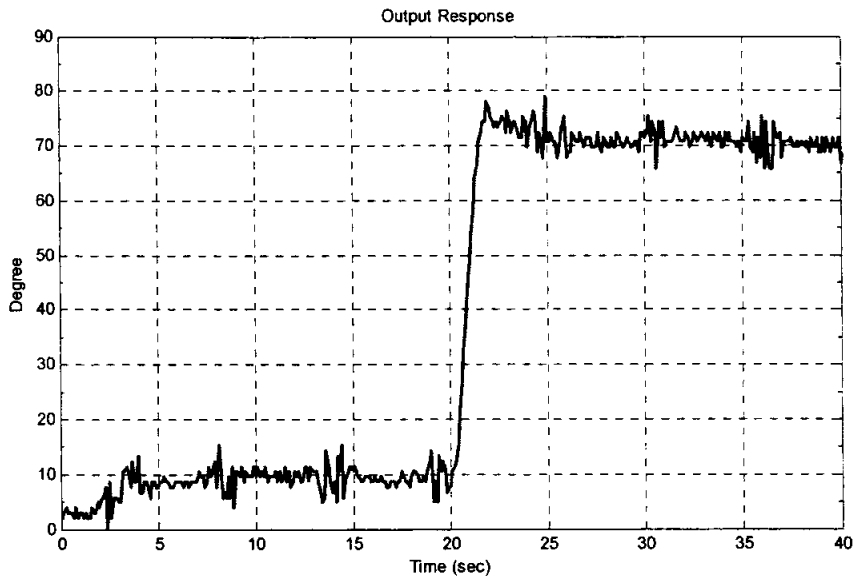
### 4.1 กล่าวนำ

หลักการออกแบบตัวควบคุมสำหรับควบคุมกระบวนการนั้นแบ่งออกเป็น 2 หลักการใหญ่ๆ คือ จำเป็นต้องรู้แบบจำลองกระบวนการ และไม่จำเป็นต้องรู้แบบจำลองกระบวนการ การออกแบบตัวควบคุมที่จำเป็นต้องรู้แบบจำลองกระบวนการ ยกตัวอย่าง เช่น การออกแบบตัวควบคุม PID และการออกแบบตัวควบคุมที่ไม่จำเป็นต้องรู้แบบจำลองกระบวนการ ยกตัวอย่าง เช่น การออกแบบตัวควบคุมแบบ Fuzzy Logic ในการศึกษาวิจัยนี้ได้กำหนดตัวควบคุมเป็นแบบ MRAC-PID ดังนั้นเราจึงจำเป็นต้องรู้แบบจำลองของกระบวนการและวิธีการในการออกแบบตัวควบคุม โดยจะอธิบายขั้นตอนทั้งหมดไว้ในบทนี้ ซึ่งเนื้อหาในบทนี้ประกอบไปด้วย การหาสมการคุณลักษณะของกระบวนการ, การออกแบบตัวควบคุม PID ด้วยเทคนิคทางเดินของราก และ การออกแบบตัวควบคุม MRAC-PID ซึ่งสามารถอธิบายแต่ละขั้นตอนได้ดังต่อไปนี้

### 4.2 การหาสมการคุณลักษณะของกระบวนการ

ในการหาสมการคุณลักษณะหรือแบบจำลองของกระบวนการนั้นสามารถทำได้หลายวิธี ตัวอย่างเช่น การหาจากองค์ประกอบทางกายภาพของระบบ การหาจากผลตอบสนองทางเอาต์พุตของระบบ เป็นต้น ซึ่งในงานวิจัยนี้จะใช้วิธีหาจากผลตอบสนองทางเอาต์พุตของระบบวงรอบเปิด ซึ่งมีขั้นตอนดังต่อไปนี้

1. กำหนดเวลาการทำงานของโปรแกรมไว้ 40 วินาที
2. ช่วงเวลา 0 – 20 วินาที กำหนดอินพุตให้กับกระบวนการเท่ากับ 0.5 V
3. ช่วงเวลา 20 – 40 วินาที กำหนดอินพุตให้กับกระบวนการเท่ากับ 1 V จะทำให้ได้ผลตอบสนองทางเอาต์พุตของกระบวนการดังรูป 4.1



รูปที่ 4.1 ผลตอบสนองทางเอาต์พุตของระบบเปิดของกระบวนการ

4. จากผลตอบสนองทางด้านเอาต์พุตของกระบวนการที่ได้ทำให้เราทราบค่าพุงเกิน 12 % และเวลาเข้าที่ 5 วินาที ( $M_p = 0.12$ ,  $t_s = 5 \text{ sec}$ ) สังเกตจากกราฟผลตอบสนองของกระบวนการวงรอบเปิด จะเห็นได้ว่ากราฟที่แสดงในรูปที่ 4.1 เป็นลักษณะของกราฟกระบวนการอันดับสอง (ตามทฤษฎี) ดังนั้นเราจึงประมาณกระบวนการพัลลัมและแผ่นเพลทนี้ ให้เป็นกระบวนการอันดับสอง (Second Order) ซึ่งสมการทางคณิตศาสตร์ของกระบวนการอันดับสอง คือ

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (4.1)$$

5. หาแบบจำลองกระบวนการ หลังจากที่เราทราบค่าพุงเกิน (Overshoot) และค่าเวลาเข้าที่ (Settling Time) จากสมการ (4.1) เราต้องการหาค่าอัตราหน่วง ( $\zeta$ ) และค่าความถี่ธรรมชาติ ( $\omega_n$ ) โดยหาจากสมการ (4.2) และสมการ (4.3) ตามลำดับ

$$M_p = e^{-\left(\frac{\zeta}{\sqrt{1-\zeta^2}}\right) \cdot \pi} \quad (4.2)$$

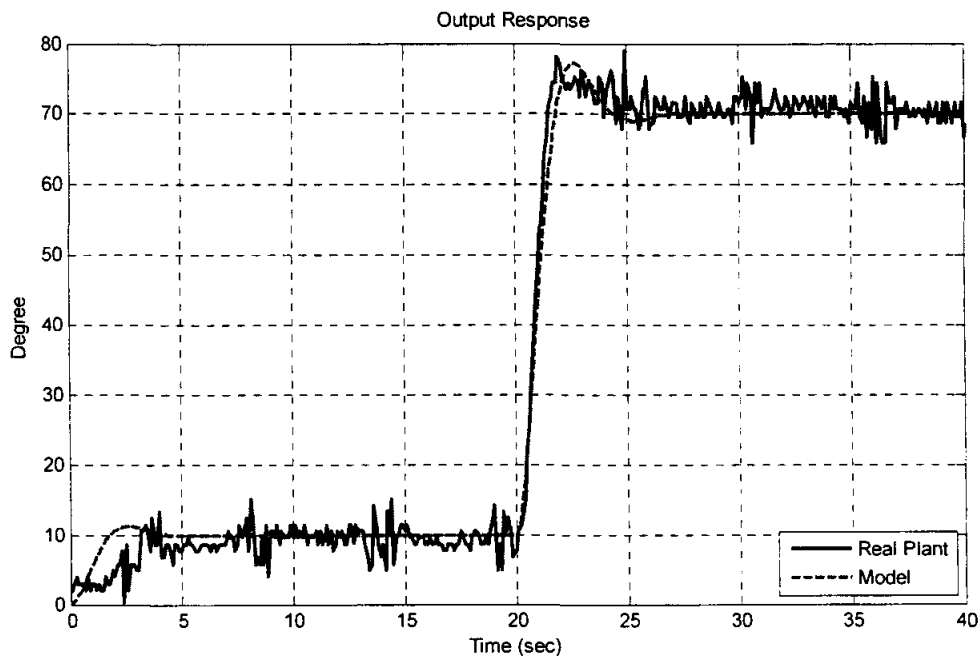
เมื่อแทนค่า  $M_p = 0.12$  ในสมการที่ (4.2) และแก้สมการ จะได้ค่าอัตราหน่วง  $\zeta = 0.55$  จากนั้นนำค่าอัตราหน่วงแทนในสมการ (4.3) เพื่อหาค่าความถี่ธรรมชาติ

$$t_s = \frac{4}{\zeta\omega_n} \quad (4.3)$$

เมื่อแทนค่า  $\zeta = 0.55$  และ  $t_s = 5$  ลงในสมการที่ (4.3) และแก้สมการ จะได้ค่าความถี่ธรรมชาติ  $\omega_n = 1.438$  หลังจากที่ได้ค่าอัตราหน่วงและค่าความถี่ธรรมชาติและนำไปแทนค่าลงในสมการที่ (4.1) จะได้ทรานส์เฟอ์ฟังก์ชันของกระบวนการ ดังสมการที่ (4.4)

$$G(s) = \frac{2.07}{s^2 + 1.6s + 2.07} \quad (4.4)$$

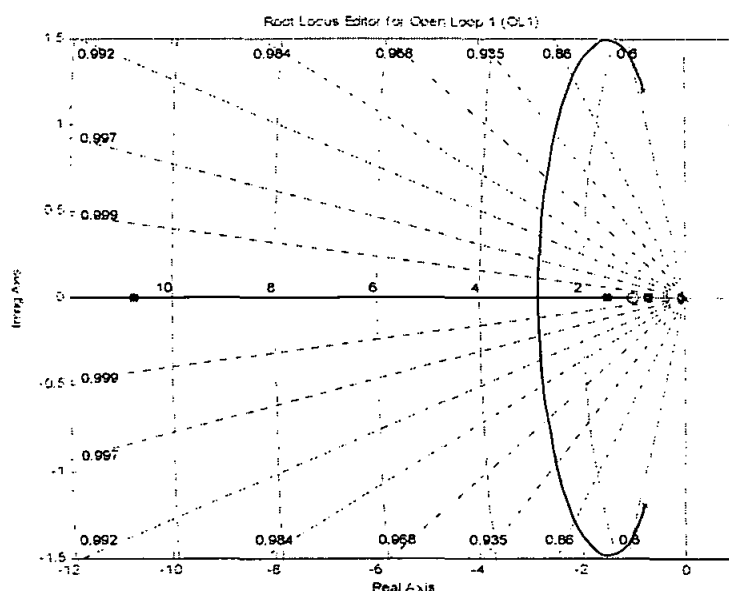
6. จากทรานส์เฟอ์ฟังก์ชันของกระบวนการตามสมการ (4.4) นำไปทดลองเทียบกับกระบวนการจริง โดยกำหนดอินพุทให้เป็นไปตามข้อ 2-3 ซึ่งจะได้ผลตอบสนองดังรูปที่ 4.2
7. จากกราฟในรูปที่ 4.2 เส้นทึบคือเอาต์พุทของกระบวนการจริง ส่วนเส้นประคือเอาต์พุทของแบบจำลอง จากรูปจะเห็นได้ว่ากราฟทั้งสองเส้นมีค่าใกล้เคียงกันมาก แสดงให้เห็นว่าทรานส์เฟอ์ฟังก์ชันที่หามาได้มีค่าใกล้เคียงกับกระบวนการจริง



รูปที่ 4.2 เปรียบเทียบผลตอบสนองทางเอาต์พุทวงรอบเปิดระหว่างกระบวนการจริงกับแบบจำลอง

### 4.3 การออกแบบตัวควบคุม PID ด้วยเทคนิคทางเดินของราก

การออกแบบด้วยวิธีทางเดินราก ในงานวิจัยนี้จะใช้เทคนิคการกำหนดตำแหน่งโพลและซีโรของตัวควบคุมที่ทำให้เส้นทางเดินของรากผ่านจุด  $s_d$  โดยใช้ SISOTOOL ซึ่งเป็นเครื่องมือในโปรแกรม MatLab มาช่วยในการหาตำแหน่งของโพลและซีโรของตัวควบคุมที่ทำให้เส้นทางเดินของรากผ่านจุด  $s_d$  เมื่อสมการคุณลักษณะของกระบวนการพัตลมและแผ่นเพลท คือ  $G_p(s) = \frac{2.07}{s^2 + 1.6s + 2.07}$  ส่วนจุด  $s_d$  นั้นจะเกิดขึ้นจากข้อกำหนด (Specification) ที่เราต้องการ จากรูปที่ 4.1 เป็นกราฟแสดงผลตอบสนองทางด้านเอาต์พุทของกระบวนการในสภาวะวงรอบเปิด (Open Loop) ซึ่งจะเห็นได้ว่าเวลาเข้าที่ (settling time) ประมาณ 5 วินาทีและค่าพุ่งเกิน 12% ซึ่งในงานวิจัยนี้เราต้องการให้เวลาเข้าที่  $t_s \leq 3$  sec,  $P.O. \leq 5\%$  เพื่อหาค่าพารามิเตอร์ของตัวควบคุมพีไอดีจะได้ทางเดินของรากบน SISOTOOL ในโปรแกรม MatLab ดังนี้



รูปที่ 4.3 การออกแบบตัวควบคุมด้วยวิธีทางเดินราก

จากการใช้ SISOTOOL ในโปรแกรม MatLab เพื่อออกแบบตัวควบคุมด้วยวิธีทางเดินรากจะทำให้ได้ลักษณะทางเดินของรากของระบบและโครงสร้างของตัวควบคุม โดยที่โครงสร้างของตัวควบคุมจะเปลี่ยนแปลงตามจำนวนของโพลและซีโรที่วางลงไปโปรแกรม และค่าของตัวควบคุมจะเปลี่ยนแปลงตามตำแหน่งของโพลและซีโร สำหรับงานวิจัยนี้เราจะใช้ตัวควบคุมแบบพีไอดีโดยโครงสร้างตัวควบคุม

แบบพีไอ คือ  $\frac{K_c(s+z_c)^2}{s} = \frac{K_o s^2 + K_p s + K_i}{s}$  ซึ่งโครงสร้างของตัวควบคุมแบบพีไอดีจะประกอบไปด้วย

$K_c$  ซึ่งเป็นค่าอัตราขยายของตัวควบคุม  $z_c$  ตำแหน่งของซีโร  $p_c$  ตำแหน่งของโพล (เนื่องจากว่าโครงสร้างของตัวควบคุมแบบพีไอนั้น ตำแหน่งของโพลจะอยู่ที่จุดกำเนิดจึงทำให้มีค่าเป็นศูนย์เวลาเขียนโครงสร้างของตัวควบคุมแบบพีไอดีพจน์ของ  $p_c$  จึงไม่ปรากฏ) จากการพล็อตทางเดินของรากเพื่อหา

ค่าพารามิเตอร์ของตัวควบคุมสำหรับกระบวนการ  $G_p(s)$  มีค่าเท่ากับ  $G_c(s) = \frac{5.5(s+1)^2}{s}$  ทำการเทียบ

ค่าสัมประสิทธิ์เพื่อหาค่าพารามิเตอร์ของ  $K_p$ ,  $K_i$  และ  $K_o$  จะทำให้ได้ค่า  $K_p = 11$ ,  $K_i = 5.5$  และ

$$K_o = 5.5$$

#### 4.4 การออกแบบตัวควบคุม MRAC-PID

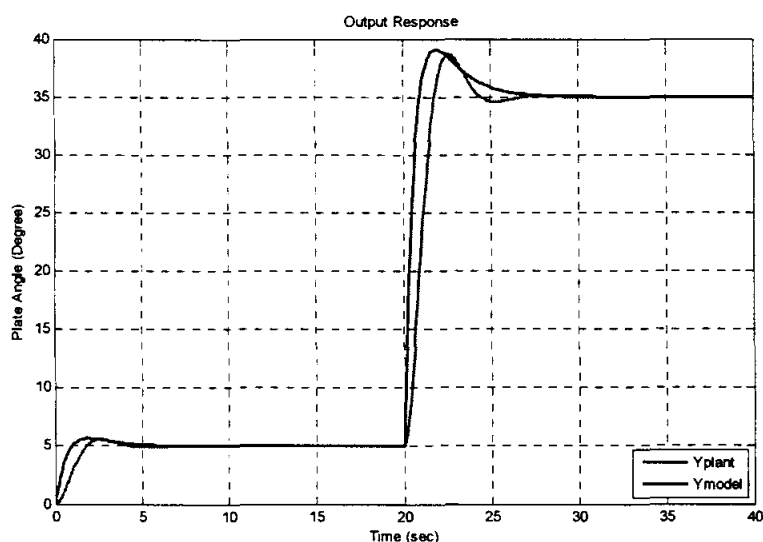
จากสมการคุณลักษณะของกระบวนการ  $G_p = \frac{b}{s^2 + a_1s + a_2} = \frac{2.07}{s^2 + 1.6s + 2.07}$  และจากการ

ออกแบบตัวควบคุมแบบพีไอดีด้วยเทคนิคทางเดินราก ในหัวข้อที่ 4.3 ได้ค่าพารามิเตอร์ตัวควบคุมแบบพีไอดี  $K_p = 11$ ,  $K_i = 5.5$  และ  $K_d = 5.5$  โดยที่กำหนดค่าพุงเกิน  $P.O. \leq 5\%$  และค่าเวลาไต่ระดับ  $t_s = 3s$  และเมื่อพิจารณาพารามิเตอร์ตัวควบคุมพีไอดีแบบ MRAC จากสมการ 2.11 จะได้

$$\begin{aligned} K_p &= \frac{2.07}{s^3 + 12.98s^2 + 24.84s + 11.4} \\ K_i &= \frac{2.07}{s(s^3 + 12.98s^2 + 24.84s + 11.4)} \\ K_d &= \frac{2.07s}{s^3 + 12.98s^2 + 24.84s + 11.4} \end{aligned} \quad (4.5)$$

จากสมการที่ 2.6 จะได้ทรานส์เฟอร์ฟังก์ชันของโมเดลอ้างอิงดังนี้

$$\frac{Y_m(s)}{U_m(s)} = \frac{s^2 + 24.84s + 11.4}{s^3 + 12.98s^2 + 24.84s + 11.4} \quad (4.6)$$



รูปที่ 4.4 แสดงผลการจำลองการเปรียบเทียบระหว่าง  $Y_p$  กับ  $Y_m$

## 4.5 สรุปบท

การออกแบบตัวควบคุม เริ่มจากการหาสมการคุณลักษณะของกระบวนการพัลลัมและแผ่นเพลท ซึ่งหาจากผลตอบสนองทางเอาต์พุตวงรอบเปิดของกระบวนการ จะสมการคุณลักษณะของกระบวนการดังสมการที่ 4.4 จากนั้นนำค่าสมการคุณลักษณะที่ได้ไปออกแบบตัวควบคุม PID ด้วยเทคนิคทางเดินราก ซึ่งจะได้ค่าพารามิเตอร์ของตัวควบคุม  $K_p = 11$ ,  $K_i = 5.5$  และ  $K_d = 5.5$  จากนั้นนำค่าสมการคุณลักษณะค่าเดิมไปออกแบบตัวควบคุม MRAC-PID ซึ่งค่าพารามิเตอร์ของตัวควบคุมแสดงดังสมการ 4.5 และค่าโมเดลอ้างอิงแสดงดังสมการ 4.6 ค่าพารามิเตอร์ของตัวควบคุมที่ได้ทั้งหมด จะถูกนำไปจำลองและทดลองกระบวนการในบทต่อไป

## บทที่ 5

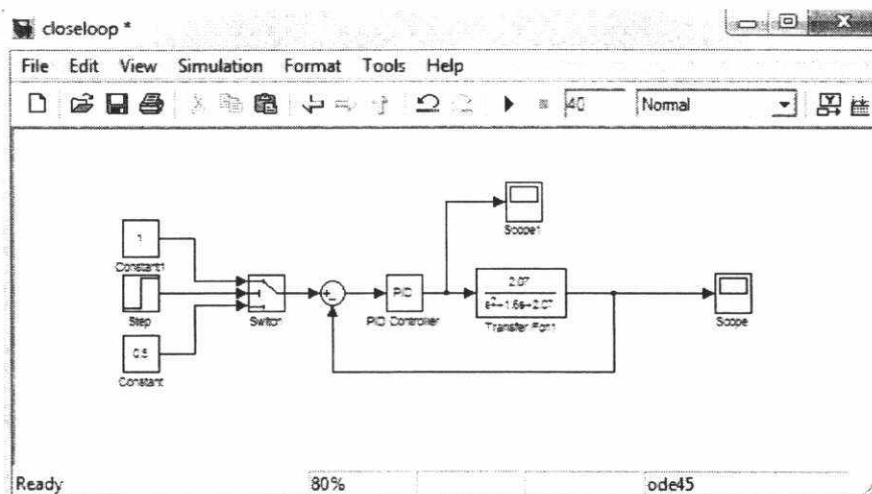
### การจำลองและการทดลอง

#### 5.1 กล่าวนำ

จากหลักการและทฤษฎีทั้งหมดที่ได้กล่าวมาในบทที่ผ่านมา เพื่อจะนำมาออกแบบหาค่าของพารามิเตอร์ของตัวควบคุม ซึ่งในบทที่ 4 ได้ทำการออกแบบหาค่าพารามิเตอร์ของตัวควบคุมด้วยเทคนิคต่างๆ ซึ่งจากผลที่ได้จะนำมาทำการจำลองและทดลองเพื่อทดสอบสมรรถนะของตัวควบคุมที่ได้ออกแบบไว้ การจำลองและทดลองสำหรับงานวิจัยนี้จะใช้โปรแกรม Simulink ใน Matlab โดยหัวข้อที่ทำการจำลองและทดลองประกอบไปด้วย การจำลองกระบวนการโดยใช้ตัวควบคุมแบบ PID ที่ออกแบบด้วยเทคนิคทางเดินของราก, การจำลองกระบวนการโดยใช้ตัวควบคุมแบบ MRAC-PID ที่ออกแบบด้วยเทคนิคทางเดินของราก, การทดลองกระบวนการเมื่อใช้ตัวควบคุมแบบ PID และ การทดลองกระบวนการเมื่อใช้ตัวควบคุมแบบ MRAC-PID โดยรายละเอียดของแต่ละส่วนสามารถอธิบายได้ดังต่อไปนี้

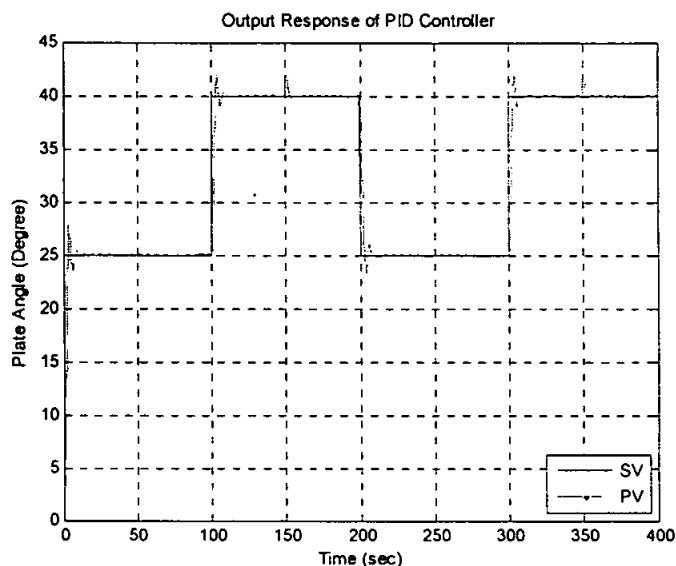
#### 5.2 การจำลองกระบวนการโดยใช้ตัวควบคุม PID ที่ออกแบบด้วยเทคนิคทางเดินของราก

ในหัวข้อนี้จะกล่าวถึงการควบคุมกระบวนการโดยใช้ตัวควบคุมแบบพีไอดี ที่ออกแบบด้วยเทคนิคทางเดินของราก ซึ่งโครงสร้างของระบบควบคุมป้อนกลับที่ใช้สำหรับการทดลองนี้แสดงดังรูปที่ 4.1 ซึ่งใช้โปรแกรม Simulink ในการจำลองกระบวนการ โดยที่ใช้สมการคุณลักษณะของกระบวนการที่จากสมการที่ 3.4 ค่าพารามิเตอร์ของตัวควบคุมพีไอดีได้จากการออกแบบด้วยวิธีเทคนิคทางเดินของรากในหัวข้อที่ 4.3



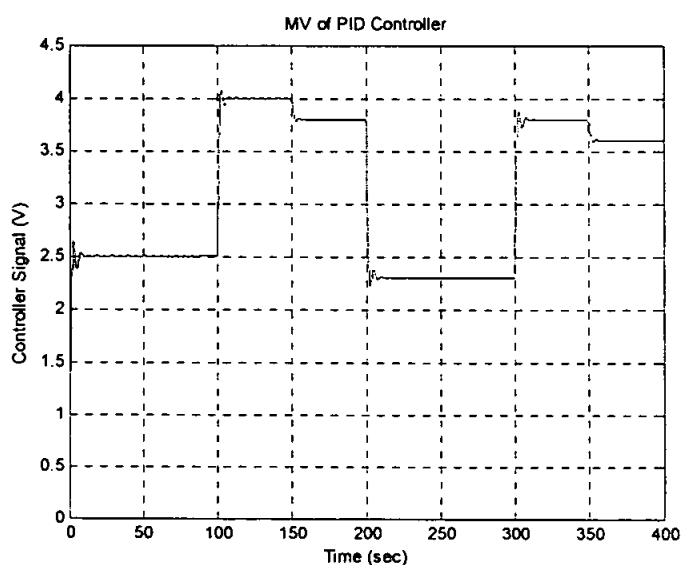
รูปที่ 5.1 โปรแกรมสำหรับจำลองกระบวนการโดยใช้ตัวควบคุม PID

โปรแกรมจะใช้เวลาในการทำงาน 400 วินาที โดยในช่วงเวลา 0 - 100 วินาที ตั้งค่าเป้าหมายไว้ 25 องศา ช่วงเวลา 100-200 วินาที เปลี่ยนค่าเป้าหมายไปที่ 40 องศา ช่วงเวลา 200 - 300 วินาที เปลี่ยนค่าเป้าหมายกลับมาที่ 25 องศา อีกครั้ง และช่วงเวลา 300 - 400 องศา เปลี่ยนค่าเป้าหมายไปที่ 40 องศา ซึ่งผลตอบสนองของระบบแสดงได้ดังรูปที่ 5.1



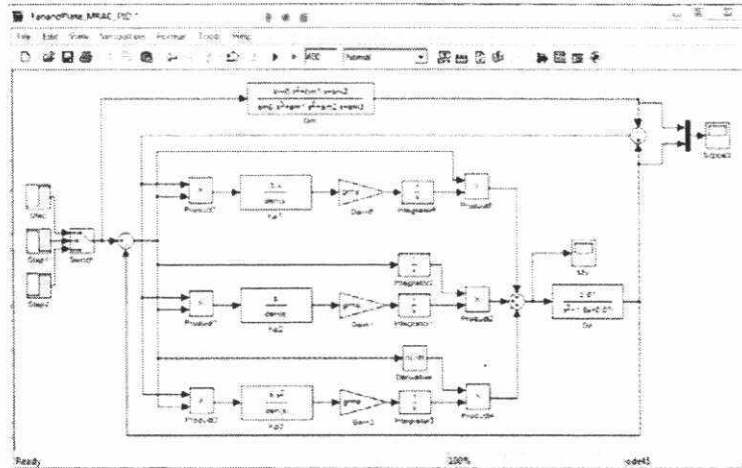
รูปที่ 5.2 ผลตอบสนองทางเอาต์พุตของการจำลองเมื่อใช้ตัวควบคุม PID

จากรูปที่ 5.2 เป็นกราฟแสดงผลตอบสนองทางด้านเอาต์พุตของกระบวนการพัลลัมและแผ่นเพลท ที่ควบคุมด้วยตัวควบคุมแบบพีไอดี ที่คำนวณหาค่าพารามิเตอร์ตัวควบคุมด้วยเทคนิคทางเดินของราก (Root Locus Technique) โดยกำหนดให้ค่าเป้าหมาย (set point) ของระบบคือ 25 องศา และพิจารณาที่เวลา 100 วินาที ผลการทดลองทำให้ได้ว่า เวลาเข้าที่ (settling time) ของระบบเท่ากับ 5 วินาที และเวลาพุ่งเกิน (Percent Over Shoot) เท่ากับ 5.7 %



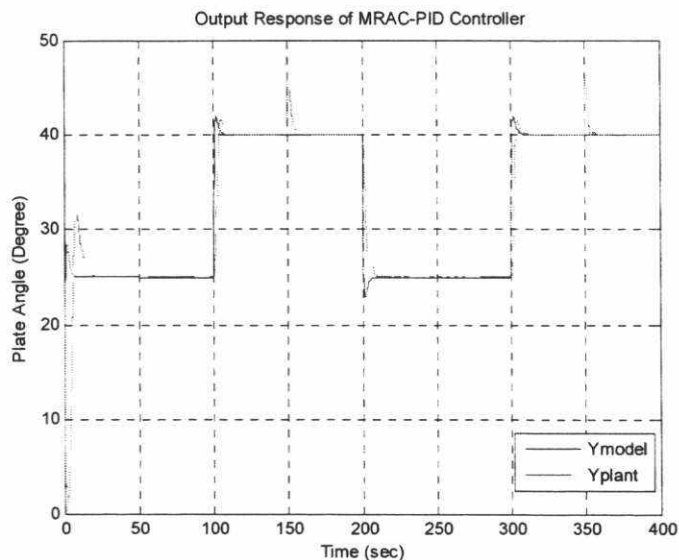
รูปที่ 5.3 สัญญาณควบคุมการจำลองกระบวนการเมื่อใช้ตัวควบคุมแบบ PID

### 5.3 การจำลองกระบวนการโดยใช้ตัวควบคุม MRAC-PID

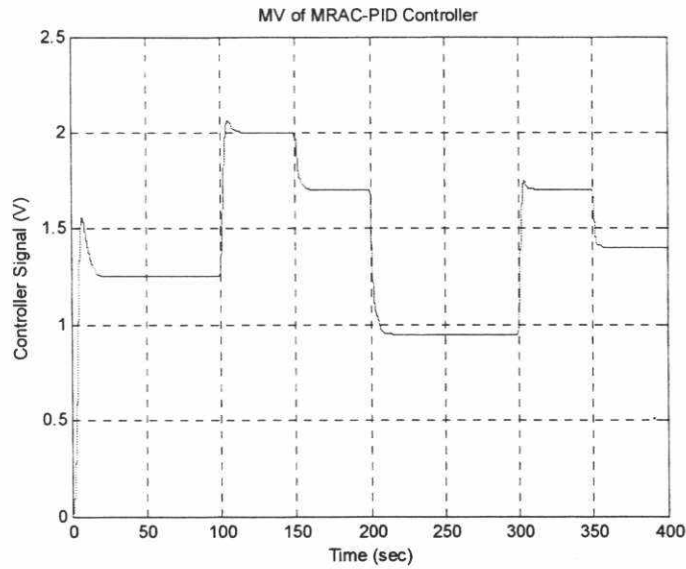


รูปที่ 5.4 โปรแกรมสำหรับจำลองกระบวนการโดยใช้ตัวควบคุม MRAC-PID

รูปที่ 5.4 เป็นโปรแกรม Simulink สำหรับจำลองกระบวนการพัฒนาและแผ่นเพลทด้วยตัวควบคุมแบบ MRAC-PID มีค่าอัตราการเรียนรู้ (Learning Rate) เท่ากับ  $-0.01$  โดยที่กำหนดเวลาในการทำงานของโปรแกรมไว้ 400 วินาที ในช่วงเวลา 0 – 100 วินาที กำหนดให้โปรแกรมควบคุมตำแหน่งของแผ่นเพลทที่ 25 องศา และที่เวลา 100 – 200 วินาที จะเปลี่ยนตำแหน่งการควบคุมแผ่นเพลทไปที่ 40 องศา ที่เวลา 200 – 300 วินาที เปลี่ยนตำแหน่งการควบคุมแผ่นเพลทไปที่ 25 องศา และที่เวลา 300 – 400 วินาที เปลี่ยนตำแหน่งการควบคุมแผ่นเพลทไปที่ 40 องศา อีกครั้ง รูปที่ 5.5 เป็นผลตอบสนองของการจำลองกระบวนการ โดยเส้นกราฟเส้นประแทนผลตอบสนองของกระบวนการส่วนกราฟเส้นทึบเป็นผลตอบสนองของโมเดลอ้างอิง จากผลการผลการจำลองจะสังเกตเห็นได้ว่าผลตอบสนองของ Yplant พยายามจะติดตามการเปลี่ยนแปลงของ Ymodel



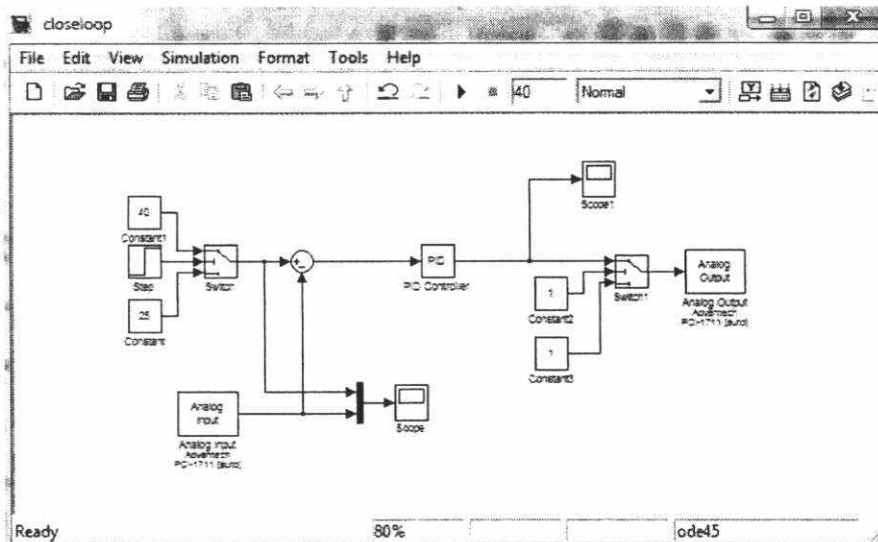
รูปที่ 5.5 ผลตอบสนองทางเอาต์พุตของการจำลองเมื่อใช้ตัวควบคุม MRAC-PID



รูปที่ 5.6 สัญญาณควบคุมการจำลองกระบวนการเมื่อใช้ตัวควบคุมแบบ MRAC-PID

#### 5.4 การทดลองกระบวนการเมื่อใช้ตัวควบคุมแบบ PID

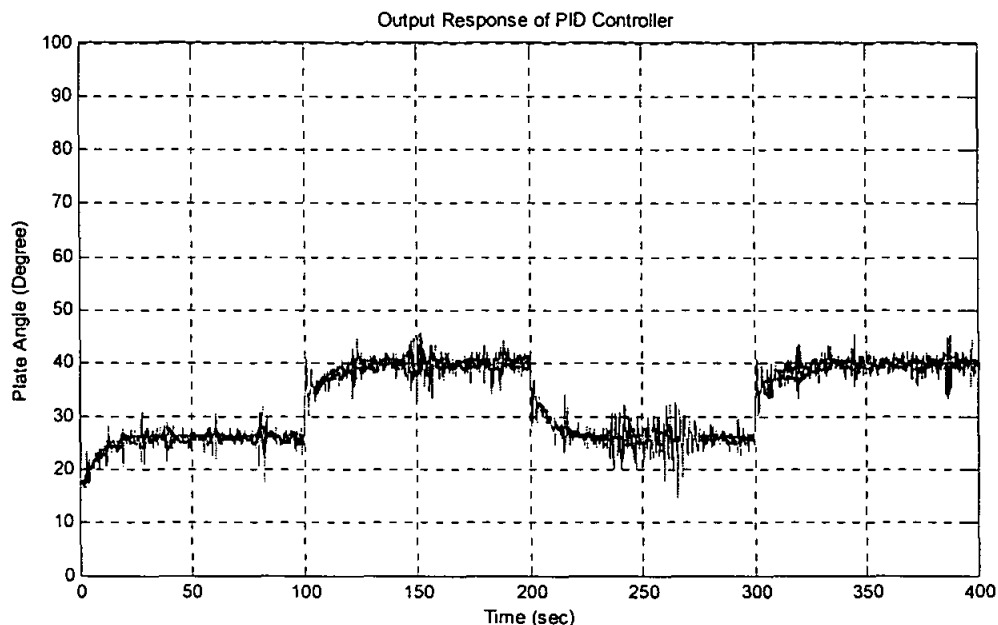
ในหัวข้อนี้จะเป็นการทดลองควบคุมกระบวนการจริง หลังจากที่เรได้จำลองกระบวนการในหัวข้อที่แล้ว โดยพารามิเตอร์ของตัวควบคุมจะใช้ค่าเดียวกันกับค่าที่ใช้ในการจำลอง ในโปรแกรมตามรูปที่ 5.7 จะเพิ่มบล็อกที่ติดต่อกับการ์ด PCI-1711 เข้ามาเพื่อติดต่อกับกระบวนการพัฒนาและแผ่นเพลท



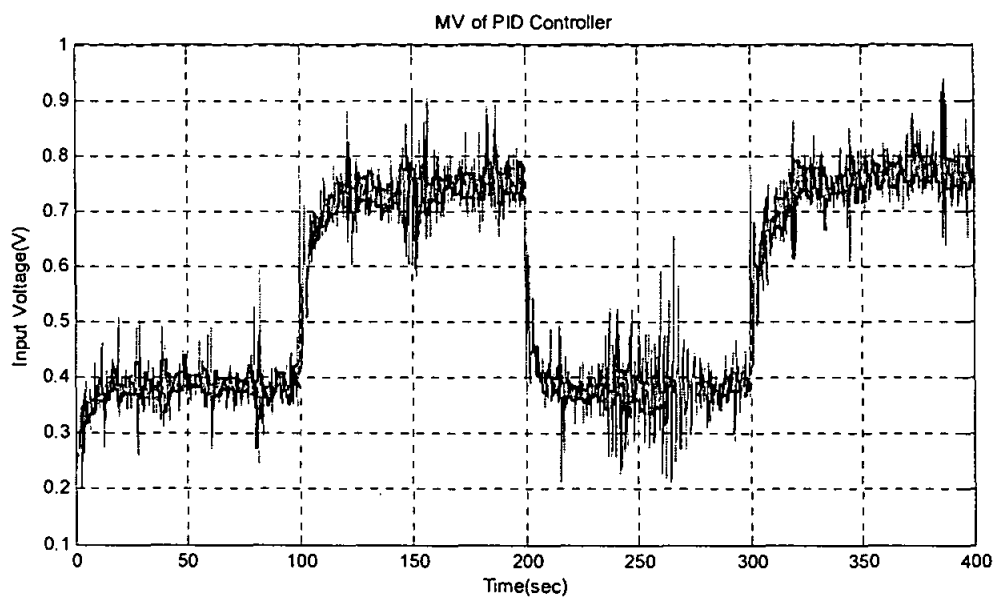
รูปที่ 5.7 โปรแกรมที่ใช้สำหรับทดลองควบคุมกระบวนการพัฒนาและแผ่นเพลท

รูปที่ 5.8 เป็นผลตอบสนองทางเอาต์พุตของกระบวนการพัฒนาและแผ่นเพลทเมื่อใช้ตัวควบคุมแบบ PID และควบคุมตำแหน่งของแผ่นเพลท 2 ตำแหน่งคือ ที่ตำแหน่ง 25 องศา และตำแหน่ง 40 องศา โดยใช้เวลาในการควบคุม setpoint ละ 100 วินาที ส่วนในรูปที่ 5.9 เป็นสัญญาณควบคุมกระบวนการเมื่อตัวควบคุมเป็นแบบ PID ส่วนในรูปที่ 5.10 เป็นเอาต์พุตของกระบวนการพัฒนาและแผ่น

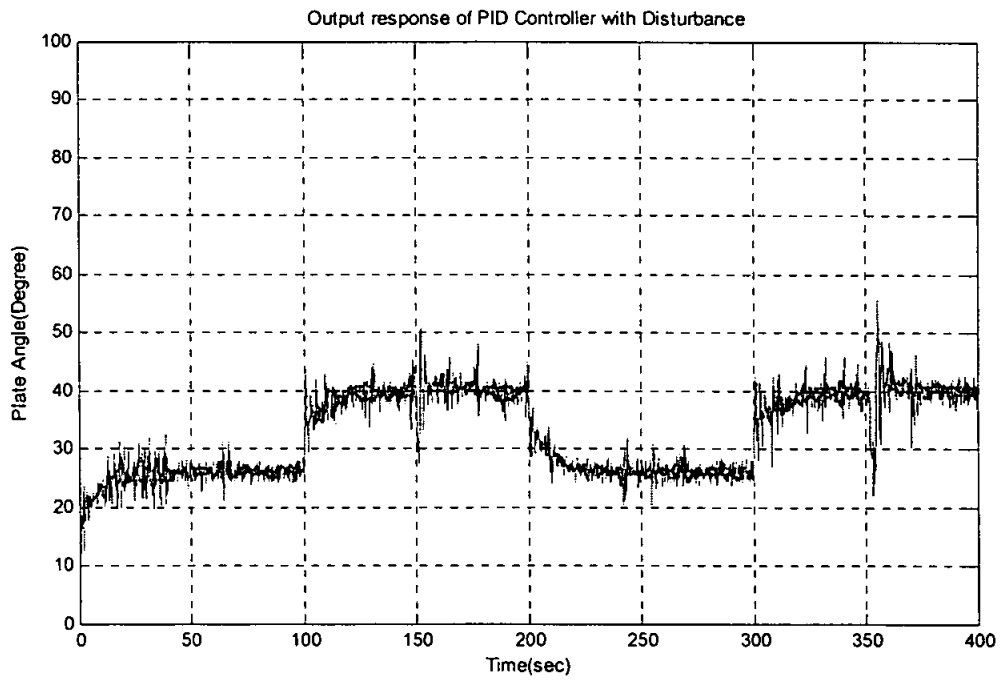
เพลทเมื่อมีการรบกวนระบบ โดยที่เวลา 150, 250 และ 350 วินาที จะมีกระรบกวนสัญญาณเกิดขึ้นโดยใช้มือกดให้แผ่นเพลทเคลื่อนที่ไปจากตำแหน่งเป้าหมาย ซึ่งตัวควบคุมก็พยายามควบคุมให้แผ่นเพลทกลับไปยังค่าเป้าหมายเดิม ส่วนในรูปที่ 5.11 เป็นสัญญาณควบคุมกระบวนการพัลลวมและแผ่นเพลทเมื่อมีสัญญาณรบกวนระบบ



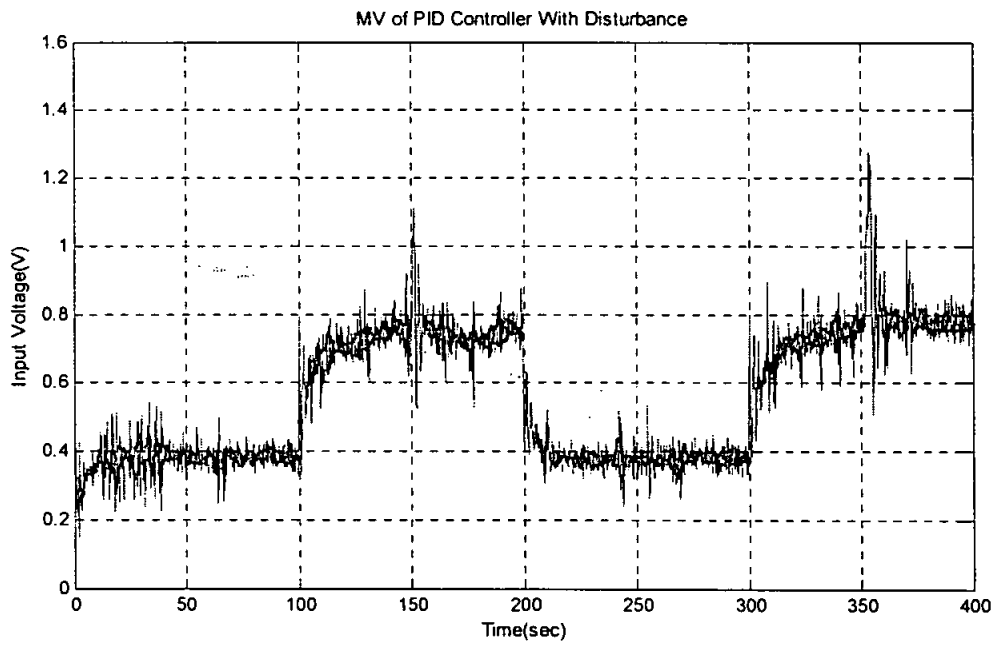
รูปที่ 5.8 ผลตอบสนองทางเอาต์พุตของกระบวนการเมื่อใช้ตัวควบคุม PID



รูปที่ 5.9 สัญญาณควบคุมกระบวนการเมื่อใช้ตัวควบคุม PID



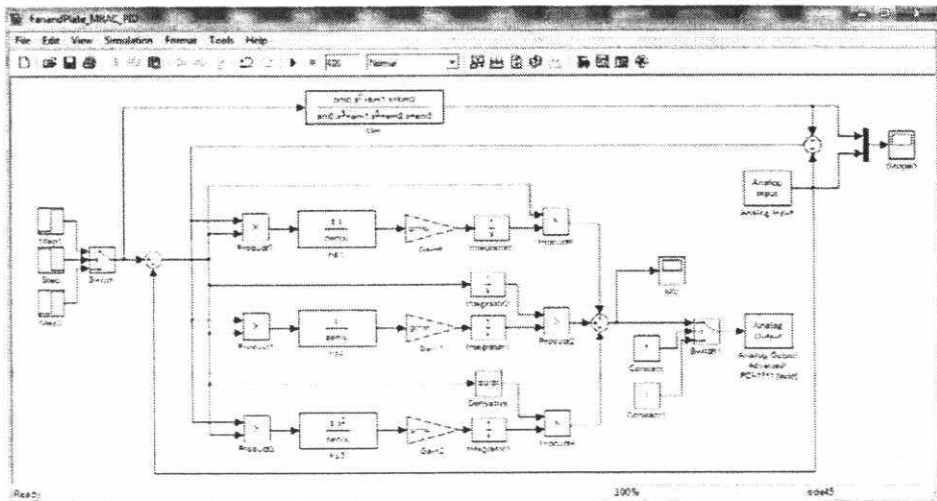
รูปที่ 5.10 ผลตอบสนองทางเอ้าท์พุทของกระบวนการเมื่อใช้ตัวควบคุมแบบ PID ที่มี Disturbance



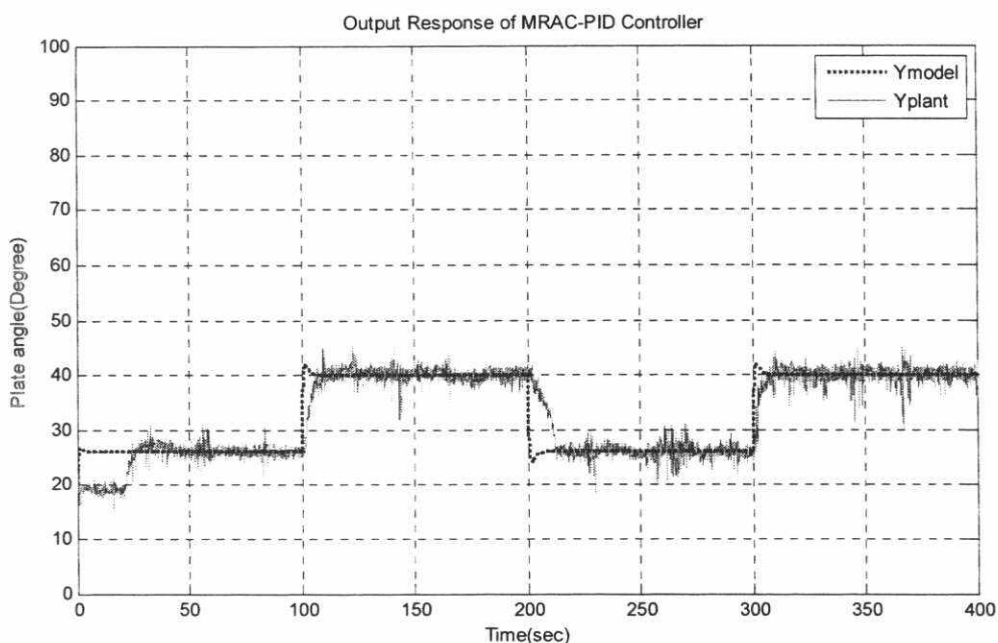
รูปที่ 5.11 สัญญาณควบคุมกระบวนการเมื่อใช้ตัวควบคุมแบบ PID ที่มี Disturbance

## 5.5 การทดลองกระบวนการเมื่อใช้ตัวควบคุมแบบ MRAC-PID

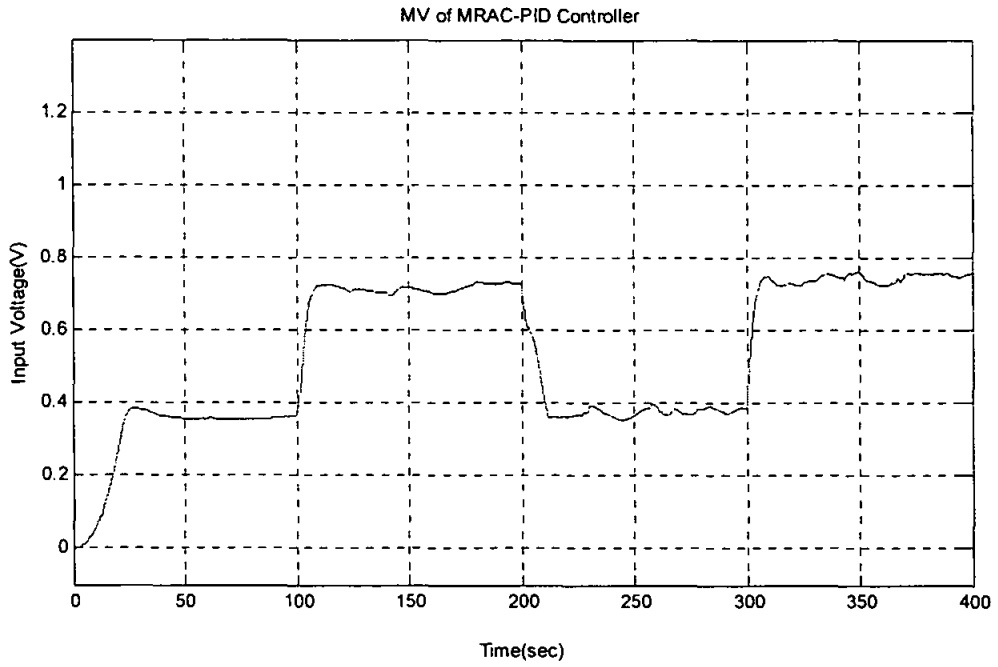
ในการจำลองกระบวนการด้วยตัวควบคุม MRAC-PID ในหัวข้อที่ 5.3 เรากำหนดค่าพารามิเตอร์ของ อัตราการเรียนรู้เท่ากับ  $-0.01$  แต่เมื่อนำค่าดังกล่าวมาใช้ในการทดลองจริงปรากฏว่าผลตอบสนองทางด้าน เออร์รอร์ไม่เป็นไปตามที่เราต้องการ จึงมีการปรับเปลี่ยนค่าพารามิเตอร์ของอัตราการเรียนรู้เป็น  $-0.1$  ใน รูปที่ 5.12 เป็นรูปแสดงโปรแกรมในการควบคุมกระบวนการโดยใช้ตัวควบคุมแบบ MRAC-PID ซึ่งผลการ ทดลองของกระบวนการแสดงได้ตามรูปที่ 5.13 และสัญญาณควบคุมกระบวนการแสดงดังรูปที่ 5.14 ส่วน รูปที่ 5.15 และรูปที่ 5.16 เป็นผลตอบสนองทางด้านเออร์รอร์และสัญญาณควบคุมกระบวนการเมื่อมีการ ระบายระบบเกิดขึ้นตามลำดับ



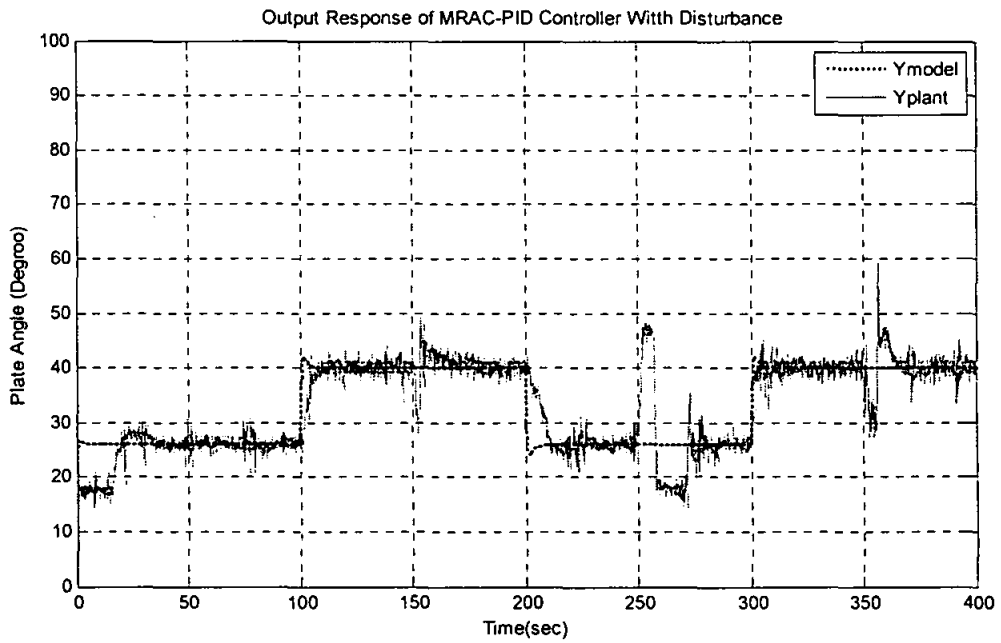
รูปที่ 5.12 โปรแกรมควบคุมกระบวนการเมื่อใช้ตัวควบคุมแบบ MRAC-PID



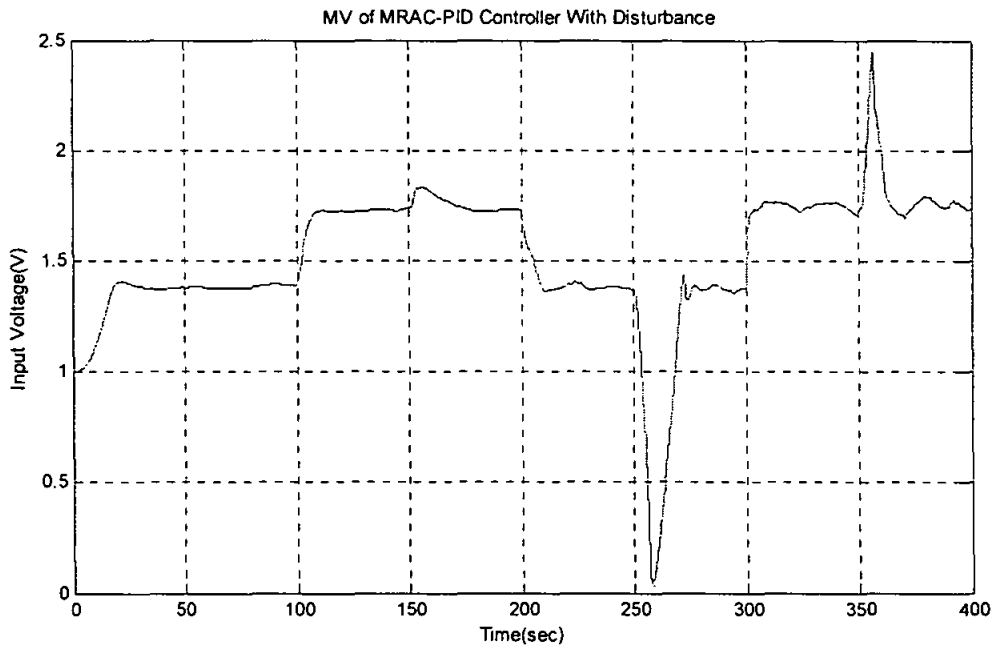
รูปที่ 5.13 ผลตอบสนองทางเออร์รอร์ของกระบวนการเมื่อใช้ตัวควบคุมแบบ MRAC-PID



รูปที่ 5.14 สัญญาณควบคุมกระบวนการเมื่อใช้ตัวควบคุมแบบ MRAC-PID



รูปที่ 5.15 ผลตอบสนองทางเอ้าท์พุทของกระบวนการเมื่อใช้ตัวควบคุมแบบ MRAC-PID ที่มี Disturbance

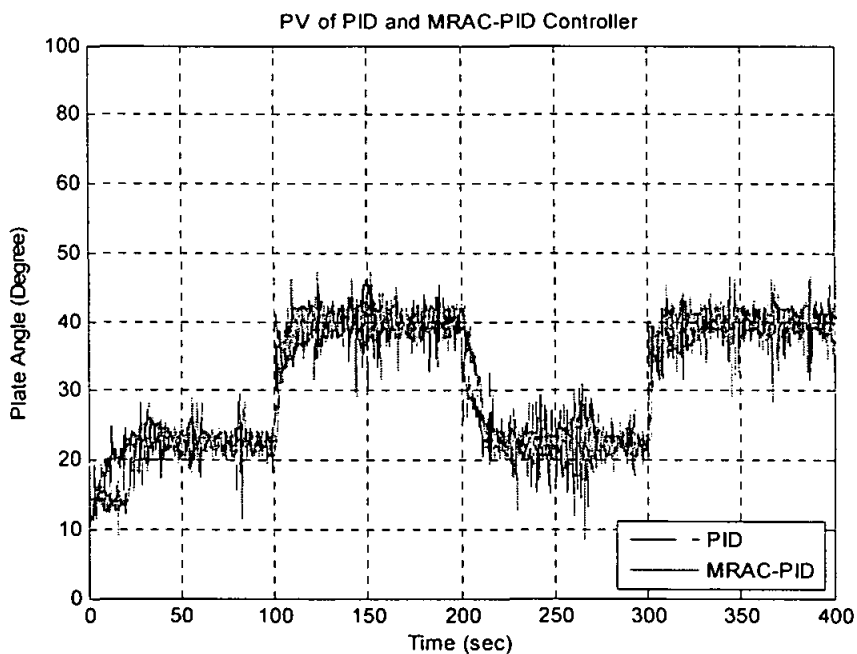


รูปที่ 5.16 สัญญาณควบคุมกระบวนการเมื่อใช้ตัวควบคุมแบบ MRAC-PID ที่มี Disturbance

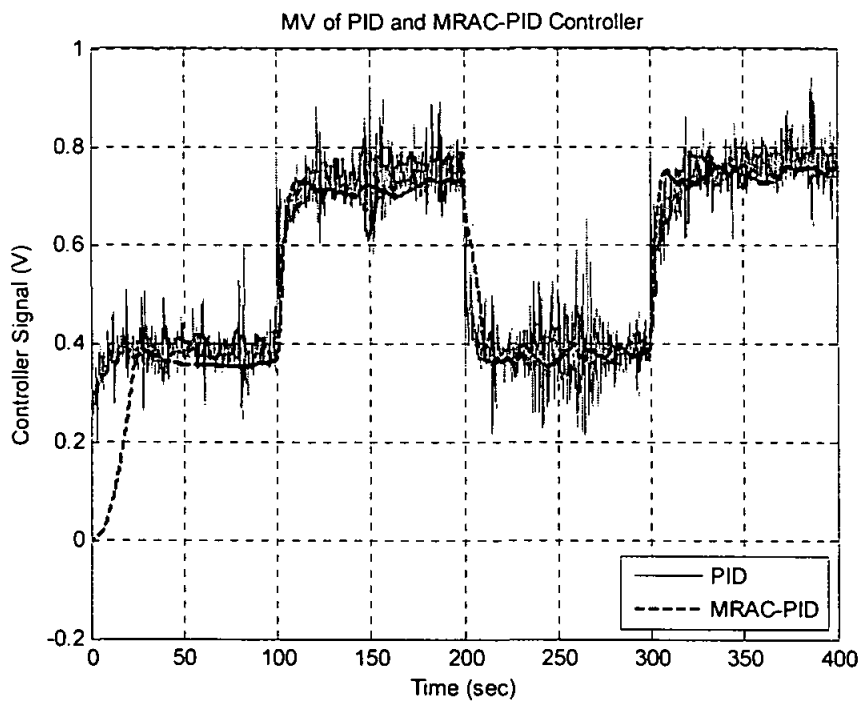
ในรูปที่ 5.15 เป็นการทดสอบประสิทธิภาพของตัวควบคุม โดยการรบกวนระบบ ในที่นี้เราใช้มือไปผลักที่แผ่นเพลทให้ตำแหน่งเปลี่ยนไปโดยทำการรบกวนระบบ 3 ครั้งคือที่เวลา 150, 250 และ 350 วินาที ซึ่งจากกราฟจะเห็นได้ว่าเมื่อทำการรบกวนตำแหน่งของแผ่นเพลทจะเปลี่ยนไปแต่เมื่อเราหยุดการรบกวนระบบ ตัวควบคุมก็จะพยายามควบคุมแผ่นเพลทให้กลับเข้าสู่ค่าเป้าหมายดั้งเดิม

## 5.6 เปรียบเทียบผลการทดลองระหว่างตัวควบคุม PID กับตัวควบคุม MRAC-PID

หลังจากที่เราได้ทำการจำลองและทดลองควบคุมกระบวนการพัดลมและแผ่นเพลทด้วยตัวควบคุมแบบ PID และตัวควบคุมแบบ MRAC-PID ในหัวข้อที่ผ่านมาแล้วนั้น จากผลการทดลองเราจะเห็นว่าตัวควบคุมทั้งสองแบบสามารถควบคุมกระบวนการให้เข้าสู่ค่าเป้าหมายได้ และเมื่อมีการรบกวนระบบเกิดขึ้น ตัวควบคุมทั้งสองก็ยังสามารถควบคุมกระบวนการให้กลับไปยังค่าเป้าหมายได้ดั้งเดิม และเพื่อเป็นการเปรียบเทียบสมรรถนะของตัวควบคุมทั้งสองแบบ ในหัวข้อนี้ผู้วิจัยจึงได้ทำการเปรียบเทียบผลการทดลองของตัวควบคุมทั้งสองแบบ ดังต่อไปนี้



รูปที่ 5.17 เปรียบเทียบผลตอบสนองของเอาต์พุตของตัวควบคุม PID กับ MRAC-PID



รูปที่ 5.18 เปรียบเทียบสัญญาณควบคุมของตัวควบคุม PID กับ MRAC-PID

ตารางที่ 5.1 เปรียบเทียบสมรรถนะของตัวควบคุมแบบ PID กับตัวควบคุมแบบ MRAC-PID

Performance Controller	$M_p$	$t_r$	$e_{ss}$
PID	0	15	0
MRAC-PID	0	5	0

ตารางที่ 5.1 เป็นผลของการเปรียบเทียบสมรรถนะของตัวควบคุมแบบ PID กับตัวควบคุมแบบ MRAC-PID (พิจารณาจากรูปที่ 5.17) โดยพิจารณาที่เวลา 100 วินาที ซึ่งจากการเปรียบเทียบแสดงให้เห็นว่าตัวควบคุมแบบ MRAC-PID ให้สมรรถนะที่ดีกว่า

## 5.7 สรุป

จากผลการทดลองแสดงให้เห็นถึงประสิทธิภาพของตัวควบคุมที่สามารถควบคุมกระบวนการให้เป็นไปตามค่าเป้าหมายที่ต้องการ ในขณะที่มีการรบกวนระบบเกิดขึ้นตัวควบคุมก็ยังคงควบคุมกระบวนการให้อยู่ที่ค่าเป้าหมายได้ ตัวควบคุมแบบ MRAC-PID เป็นตัวควบคุมที่สามารถติดตามค่าเป้าหมายได้ดีถึงแม้ค่าเป้าหมายจะมีการเปลี่ยนแปลงอยู่ตลอดเวลาก็ตาม ซึ่งถือว่าเป็นจุดเด่นของตัวควบคุมแบบนี้ ข้อดีอีกอย่างหนึ่งก็คือ ตัวควบคุมแบบ MRAC-PID จะปรับค่าของพารามิเตอร์ของตัวควบคุมด้วยตัวเอง ทุกครั้งที่มีการเปลี่ยนจุดทำงาน แต่ตัวควบคุมแบบ PID จะไม่สามารถเปลี่ยนแปลงได้ จะต้องทำการคำนวณและแทนค่าใหม่ทุกครั้ง ถ้าไม่แทนค่าใหม่ตัวควบคุมก็จะใช้พารามิเตอร์ค่าเดิมซึ่งในบางครั้งหากระบบเปลี่ยนไปมากค่าพารามิเตอร์เดิมก็ไม่สามารถใช้ในการควบคุมกระบวนการได้

## บทที่ 6

# สรุปผลการวิจัยและข้อเสนอแนะ

### 6.1 บทสรุป

กระบวนการพัลลัมและแผ่นเพลทเป็นกระบวนการที่มีความไม่เป็นเชิงเส้น (Non-linearity) และความแปรปรวนสูง เนื่องจากเป็นกระบวนการเปิด สิ่งรบกวนจากภายนอกจึงสามารถเกิดขึ้นได้ตลอดเวลา ไม่ว่าจะเป็นแรงสั่นสะเทือน ลม หรือแม้กระทั่งสภาพของจุดทำงานเองก็มีผลต่อกระบวนการได้ ดังนั้นในการทดลองจึงได้ติดตั้งจุดทดลองให้อยู่ในสภาพที่ทำให้เกิดสิ่งรบกวนที่ไม่ปรารถนาน้อยที่สุด แบบจำลองของกระบวนการหาได้จากการพิจารณาพฤติกรรมของผลตอบสนองทางด้านเออร์พุทวอร์บเปิดของกระบวนการ (ตามรูปที่ 4.1) เมื่อพิจารณาแล้วประมาณได้ว่าเป็นกระบวนการอันดับสอง ซึ่งโครงสร้างของแบบจำลองเป็นไปตามสมการ 4.1 โดยจากกราฟทำให้เราทราบค่า  $M_p = 0.12$ ,  $t_s = 5 \text{ sec}$  ซึ่งนำไปหาแบบจำลองกระบวนการตามสมการ 4.3 และ 4.3 และได้แบบจำลองกระบวนการตามสมการ 4.4

ตัวควบคุมที่ใช้ในการทดลองมี 2 แบบคือ ตัวควบคุมแบบ PID กับตัวควบคุมแบบ MRAC-PID โดยตัวควบคุมทั้ง 2 แบบถูกออกแบบด้วยเทคนิคทางเดินราก ในส่วนของตัวควบคุมแบบ MRAC-PID ได้นำเอากฎของ MIT (MIT Rule) มาช่วยในการออกแบบด้วย จากแบบจำลองของกระบวนการในสมการ 4.4 นำมาออกแบบตัวควบคุมพีไอดีด้วยเทคนิคทางเดินราก โดยกำหนดให้เวลาเข้าที่  $t_s \leq 3 \text{ sec}$ ,  $P.O. \leq 5\%$  ซึ่งได้ค่าพารามิเตอร์ของตัวควบคุม คือ ค่า  $K_p = 11$ ,  $K_i = 5.5$  และ  $K_d = 5.5$  จากนั้นนำแบบจำลองกระบวนการตามสมการ 4.4 มาออกแบบตัวควบคุม MRAC-PID ด้วยเทคนิคทางเดินรากและกฎของ MIT จะทำให้ได้พารามิเตอร์ของตัวควบคุม ดังนี้

$$K_p = \frac{2.07}{s^3 + 12.98s^2 + 24.84s + 11.4}$$
$$K_i = \frac{2.07}{s(s^3 + 12.98s^2 + 24.84s + 11.4)}$$
$$K_d = \frac{2.07s}{s^3 + 12.98s^2 + 24.84s + 11.4}$$

และได้โมเดลอ้างอิงของกระบวนการ 
$$\frac{Y_m(s)}{U_m(s)} = \frac{s^2 + 24.84s + 11.4}{s^3 + 12.98s^2 + 24.84s + 11.4}$$

ผลของการทดลองในบทที่ 5 เป็นการทดลองควบคุมกระบวนการจริงโดยใช้ตัวควบคุมที่ได้ ออกแบบในบทที่ 4 โดยควบคุมตำแหน่งของแผ่นเพลทที่ระดับ 30 องศา และ 60 องศา ซึ่งการทดลองควบคุมกระบวนการโดยใช้ตัวควบคุม PID ผลการทดลองที่ได้เป็นตามข้อกำหนดที่ออกแบบไว้ และการทดสอบควบคุมกระบวนการโดยใช้ตัวควบคุมแบบ MRAC-PID ผลการทดลองที่ได้ก็เป็นไป

ตามข้อกำหนดที่ได้ออกแบบไว้เช่นกัน นอกจากนี้แล้วในการทดลองยังได้ทดลองสมรรถนะของตัวควบคุมโดยทำการรบกวนระบบโดยใช้มือกดแผ่นเพลทประมาณ 2 วินาที แล้วปล่อย ซึ่งผลการทดลองในรูปที่ 5.10 แสดงให้เห็นว่าตัวควบคุมสามารถควบคุมให้แผ่นเพลทกลับไปยังตำแหน่งเดิมภายในเวลาที่กำหนด ข้อดีของตัวควบคุมที่มีโครงสร้างแบบ MRAC คือ เมื่อมีการเปลี่ยนแปลงของกระบวนการ ตัวควบคุมจะปรับค่าพารามิเตอร์ของตัวควบคุมเองโดยใช้โมเดลอ้างอิงเป็นตัวอ้างอิงในการปรับโดยไม่ต้องคำนวณค่าพารามิเตอร์ของตัวควบคุมใหม่

## 6.2 ปัญหาที่พบในการวิจัยและแนวทางในการแก้ปัญหา

1. ปัญหาจากกระบวนการ เนื่องจากกระบวนการที่สร้างขึ้นเป็นต้นแบบของกระบวนการพัลลวมและแผ่นเพลท ดังนั้นค่าพารามิเตอร์ต่างๆ จึงไม่ค่อยคงที่เท่าที่ควร เช่น ตำแหน่งของแผ่นเพลท ตำแหน่งของ Sensor ค่าแรงดันจากชุดขับเคลื่อนมอเตอร์ ความร้อนของมอเตอร์ ซึ่งตัวแปรเหล่านี้มีผลต่อโมเดลของกระบวนการ ซึ่งทำให้การควบคุมไม่เป็นไปตามที่ต้องการ การแก้ไขคือ เลือกอุปกรณ์ที่ดีที่มีการเปลี่ยนแปลงต่ำๆ พยายามคงค่าของตัวแปรต่างๆ ให้คงที่มากที่สุด และพยายามทดลองให้เสร็จให้แล้วเสร็จในครั้งเดียว

2. ปัญหาจากสิ่งรบกวนภายนอก กระบวนการพัลลวมและแผ่นเพลทเป็นกระบวนการแบบเปิด ดังนั้นปัจจัยภายนอกจึงมีผลต่อกระบวนการโดยตรง ทำให้การควบคุมทำได้ยาก เช่น สภาพของห้องที่ทำการทดลอง ตำแหน่งการติดตั้งของกระบวนการ ซึ่งถ้าติดตั้งลมที่พัดออกจากพัลลวมก็จะกลายมาเป็นสิ่งรบกวนระบบเอง แนวทางแก้ไข คือ ทดลองในห้องที่ปิดและกว้าง การติดตั้งกระบวนการให้หันหน้าออกจากกำแพงห้องเพื่อไม่ให้ลมที่พัดออกจากพัลลวมตีกลับมารบกวนกระบวนการ

3. ปัญหาจากตัว Sensor กระบวนการพัลลวมและแผ่นเพลทที่ออกแบบสำหรับการทดลองนี้ใช้ Potentiometer หรือ ความต้านทานแบบปรับค่าได้ และใช้หลักการแบ่งแรงดัน (Voltage Divider) ซึ่ง Potentiometer ที่ใช้มีขนาด  $50K\Omega$  ที่ 10 รอบ ดังนั้นความต้านต่อรอบมีค่าเท่ากับ  $5K\Omega$  ในขณะที่การทดลองควบคุมตำแหน่งของแผ่นตัว Potentiometer จะหมุนอยู่ในช่วง  $\frac{1}{4}$  รอบเท่านั้น ทำให้การเปลี่ยนแปลงของค่าความต้านทานอยู่ในช่วง  $0 - 1.25K\Omega$  ซึ่งเกิดขึ้นน้อยทำให้เกิดการเปลี่ยนแปลงค่าแรงดันแตกต่างจากวงจรแบ่งแรงดันน้อย (ประมาณ 0.2V) เมื่อขยายให้เป็นแรงดันมาตรฐานจึงมีขนาดอัตราขยายที่สูง การเปลี่ยนแปลงของแผ่นเพลทจึงมีผลมาก แนวทางการแก้ไข คือ เปลี่ยนตัว Potentiometer ให้มีขนาดรอบที่น้อยลง หรือใช้ วงจรขยายสัญญาณที่มีประสิทธิภาพ เช่น ใช้ IC ที่เป็น Instrument Amplifier

### 6.3 ข้อเสนอแนะและแนวทางการพัฒนา

1. การทดลองนี้ใช้คอมพิวเตอร์เป็นตัวควบคุม และใช้โปรแกรม Matlab Simulink เป็นซอฟต์แวร์ในการจัดการและติดต่อกับอุปกรณ์ภายนอกด้วยการดแปลงสัญญาณ PCI-1711 ซึ่งมีต้นทุนที่สูง ดังนั้นหากเปลี่ยนตัวควบคุม ซอฟต์แวร์ และการ์ด Interface ให้ต้นทุนต่ำลงก็จะเพิ่มคุณค่าของงานวิจัยนี้ได้มากขึ้น

2. การพัฒนาให้กระบวนการพัลลัมและแผ่นเพลทเป็นระบบที่ทำงานแบบฝังตัว โดยการใช้ไมโครคอนโทรลเลอร์เป็นตัวควบคุมและพัฒนาให้ทำงานได้แบบ Stand Alone จะเป็นแนวทางที่ดีอีกแนวทางหนึ่งที่จะต่อยอดงานวิจัยนี้

3. พัฒนาให้กระบวนการพัลลัมและแผ่นเพลทที่ใช้ในงานวิจัยนี้ เป็นชุดทดลองเพื่ออธิบายพฤติกรรมของกระบวนการแบบไม่เป็นเชิงเส้น และอธิบายการออกแบบตัวควบคุมให้กับนักศึกษา หรือผู้ที่สนใจระบบควบคุม ก็จะเป็นแนวทางที่ดีอีกทางหนึ่งสำหรับการพัฒนาต่อยอดกระบวนการพัลลัมและแผ่นเพลท

## บรรณานุกรม

- [1] Ioannou P. and J. Sun, *“Robust Adaptive Control”*, published by Prentice Hall, Inc in 1996 (out of print in 2003)
- [2] Åström, *“Model-Reference Adaptive Control-From Theory to Practice”*, Prentice-Hall, 1992
- [3] Osburn, P.V., A.P. Whitaker and A. Kezer, *“New Developments in the Design of Model Reference Adaptive Control Systems”*, Paper No.61-39, Institute of the Aerospace Sciences, 1961.
- [4] Keith Sevcik, *“Model Reference Adaptive Control”*, Survey of Control Systems (MEM 800)
- [5] A. Numsomran, V. Tipsuwanporn, *“The design of robust PID control for Fan and Plate process”*, Proceeding CA '07 Proceedings of the Ninth IASTED International Conference on Control and Applications, CA, ©2007
- [6] J.G. Ziegler and N.B. Nichols, *“Optimum settings for automatic controllers”*, Trans. ASME, vol.65, 1943, pp.433-444
- [7] M. Zhuang, and D.P. Atherton, *“PID controller design for a TITO system”*, IEE Proc.- Control Theory Appl.,Vol.141, No.2, pp.111-120, 1994.
- [8] Pranai Kungwalrut, Maitree Thumma, Vittaya Tipsuwanporn, Arjin Numsomran Pisit Boonsrimuang, *“Design MRAC PID Control for Fan and Plate Process”*, SICE Annual Conference 2011 September 13-18, 2011, Waseda University, Tokyo, Japan
- [9] Katsuhiko Ogata, *“Modern Control Engineering”*, Prentice Hall (1997)
- [10] Mathwork Inc, *“Simulink Simulation and Model-base Design”*,
- [11] ARM Ltd and ARM Germany GmbH. *“Getting Started Building Applications with RL-ARM For ARM Processor-Based Microcontrollers”*.
- [12] Ai Xiong; Yongkun Fan. *“Application of a PID Controller using MRAC Techniques for Control of the DC Electromotor Drive”* International Conference on Mechatronics and Automation, 2007. ICMA 2007.
- [13] Kangwanrat S; Tipsuwannaporn V.; Numsomran A. *“Design of PI controller using MRAC techniques for coupled-tanks process”*. Conference on Control Automation and System (ICCAS) 2010

- [14] Gang Feng and Logelio Lozano, “**Adaptive Control System**”. Newnes (1999)
- [15] Astrom, K. J. (1983). “**Theory and Applications of Adaptive Control**”. A Survey. Automatica 19(5), 471-486.
- [16] Gang Tao, Shuhao Chen, Xidong Tang, Suresh M. Joshi. “**Adaptive Control of Systems with Actuator Failures**” Springer; 1st Edition. edition (2004)
- [17] Mahyuddin Muhammad Nasiruddin. “**Direct model reference adaptive control of coupled tank liquid level control system**” Masters thesis, Universiti Teknologi Malaysia, Faculty of Electrical Engineering.
- [18] Araki M. “**PID Control**”. Control System Robotics and Automation, Vol 11. Kyoto University, Japan
- [19] U.A.Bakshi, V.U.Bakshi “**Control System Engineering**”. Technical Publications Pune.
- [20] Landau, I. D. (1979). “**Adaptive Control the Model Reference Approach**”, Marcel Dekker, New York.
- [21] Sobel, K., H. Kaufman and L. Mabus (1982). “**Adaptive Control for a Class of MIMO Systems**”. IEEE Transactions on Aerospace 18, 576-590.
- [22] Ioannou, P., and J. Sun (1996). “**Robust Adaptive Control**”, Prentice-Hall, Upper Saddle River, New Jersey.
- [23] Eugene Lavretsky, Ross Gadiant, and Irene M. Gregory. “**Predictor-Based Model Reference Adaptive Control**” American Institute of Aeronautics and Astronautics
- [24] Narendra K. S., Annaswamy A. M., “**Stable Adaptive Systems**”, Dover, 2005.
- [25] Ioannou, P. A., Fidan, B., “**Adaptive Control Tutorial**”, SIAM, 2006.
- [26] Khalil, H. K., “**Nonlinear Systems 3<sup>rd</sup> ed**”, Prentice Hall Inc., New York, 2002.
- [27] Slotine, J.-J. E., Li, W., “**Applied Nonlinear Control**”, Prentice Hall, 1995.
- [28] Pomet, J. B., Praly, L., “**Adaptive nonlinear regulation: Estimation from the Lyapunov equation,**” IEEE Trans. Autom. Control, pp.729-740, 1992.
- [29] R.Prakash, R.Anita. “**ROBUST MODEL REFERENCE ADAPTIVE PI CONTROL**” Journal of Theoretical and Applied Information Technology
- [30] Kreisselmier, G.,And Narendra, K.S.: “**Stable model reference adaptive control in the presence of bounded disturbances**”, IEEE Transection. Automatic. Control, AC-29, pp.202-211, 1984.
- [31] Petros.A.ioannou & Jing Sun, “**Robust Adaptive Control**”, Prentice Hal, Second edition, 1996.

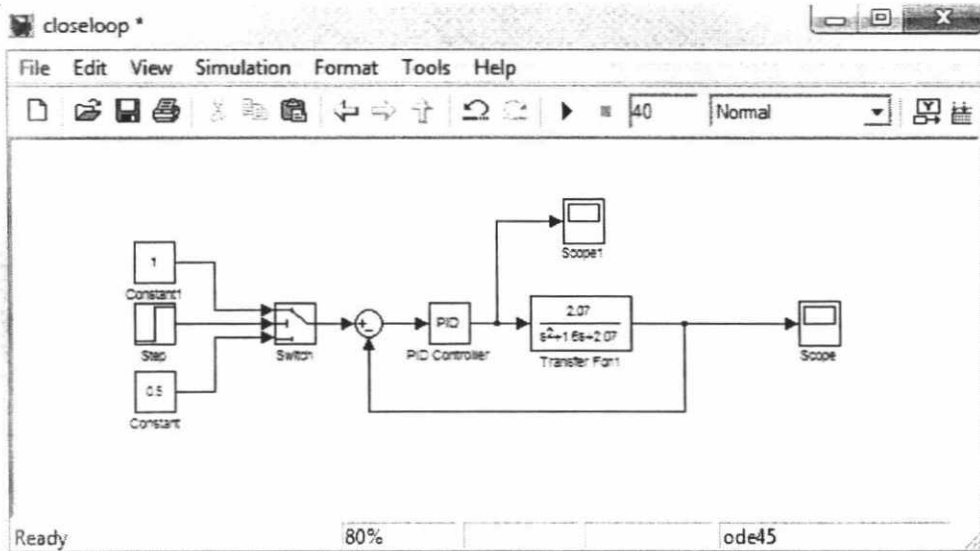
- [32] Whitaker, Yamron and A. Kezer, “**Design of Model Reference Adaptive Control Systems for aircraft,**” Report R-164, Instrumentation Laboratory, M.I.T.press, Cambridge, Massachusetts, 1958.
- [33] Kreisselmier, G., and Anderson, B.D.O. “**Robust model reference adaptive control.**” IEEE Transactions on Automatic Control 31:127-133, 1986.
- [34] Narendra, K.S. And Valavani ,L.: “**Stable adaptive controller design direct control**”, IEEE Trans. Automat. Contr., AC-23, pp.570-583 ,1978.
- [35] Bellman, R.E. “**Adaptive Control Processes**” A Guided Tour, Princeton University Press, Princeton, New Jersey, 1961.
- [36] Bertsekas, D.P., “**Dynamic Programming**”, Prentice Hall, Engle wood Clis, New Jersey, 1987.
- [37] Barkana, I., “**Adaptive Control: A Simplified Approach**”, in C. T. Leondes (Ed.), Advances in Control and Dynamics, Vol. 25, Academic Press, New York.
- [38] Aseltine, J.A., A.R. Mancini and C.W. Sartune, “**A Survey of Adaptive Control Systems**”, IRE Transactions on Automatic Control, Vol. 3, pp. 102-108,

ภาคผนวก

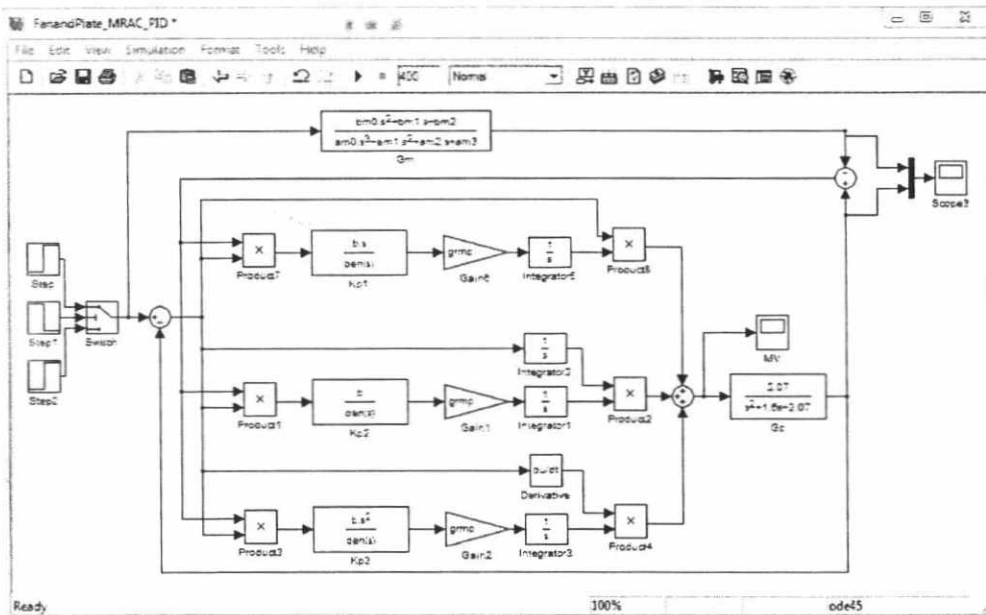
ภาคผนวก ก.  
โปรแกรมที่ใช้ในการทดลอง

จากที่ได้กล่าวมาข้างต้นว่า การวิจัยนี้ได้ใช้โปรแกรม Matlab Simulink เป็นซอฟต์แวร์ในการจำลองและทดลองควบคุมกระบวนการ ซึ่งลักษณะของโปรแกรมจะอยู่ในรูปแบบของ Graphics ประกอบไปด้วย 4 โปรแกรม ดังต่อไปนี้

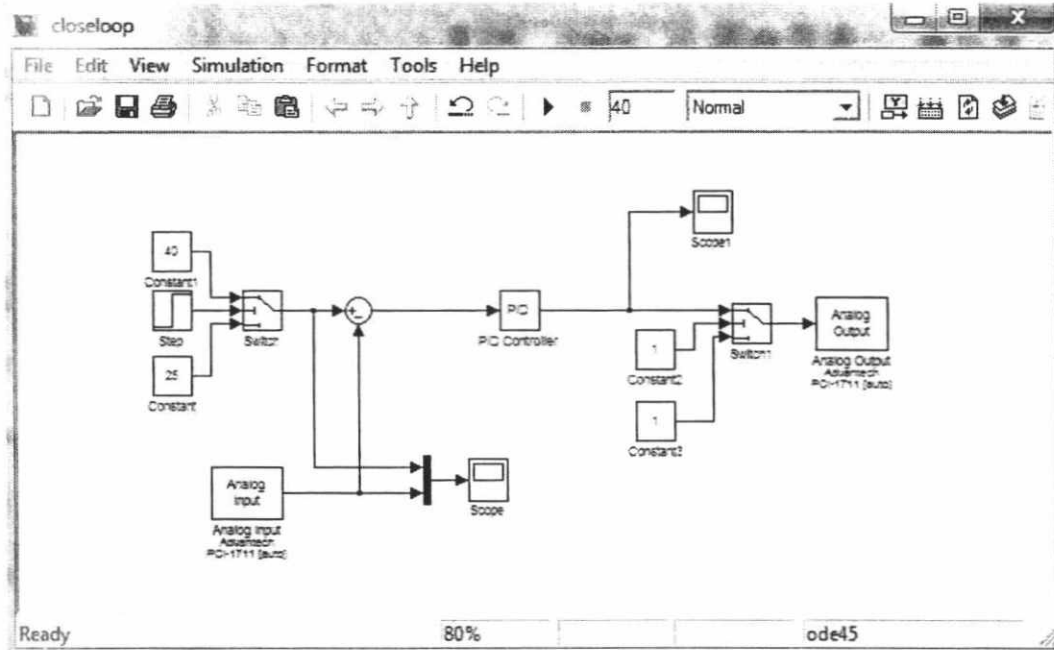
### 1. โปรแกรมสำหรับจำลองควบคุมกระบวนการเมื่อใช้ตัวควบคุมแบบ PID



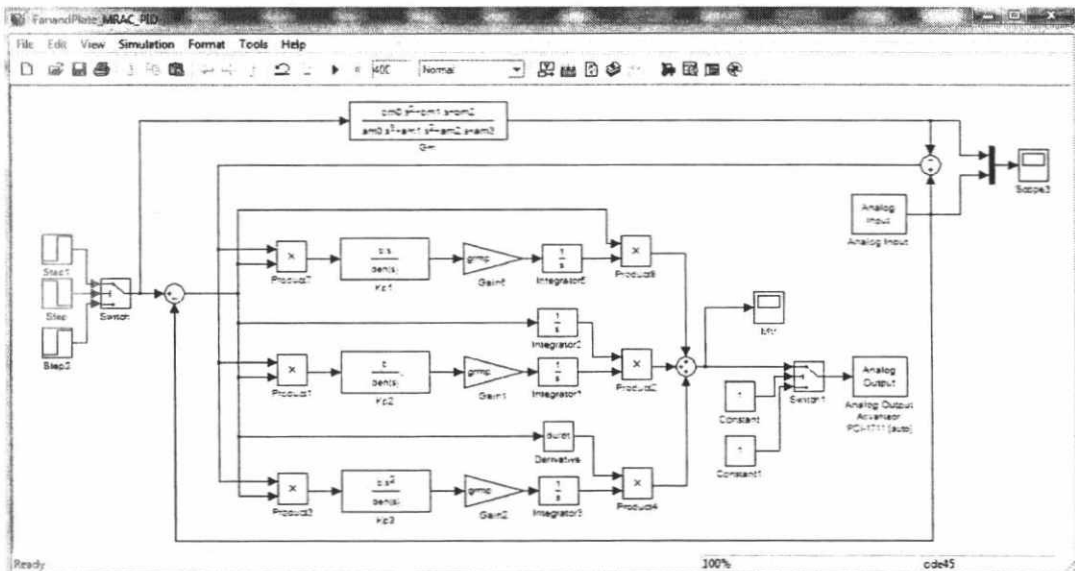
### 2. โปรแกรมสำหรับจำลองควบคุมกระบวนการเมื่อใช้ตัวควบคุมแบบ MRAC-PID



### 3. โปรแกรมสำหรับควบคุมกระบวนการจริงเมื่อใช้ตัวควบคุมแบบ PID



### 4. โปรแกรมสำหรับควบคุมกระบวนการจริงเมื่อใช้ตัวควบคุมแบบ MRAC-PID



ภาคผนวก ข.

ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่

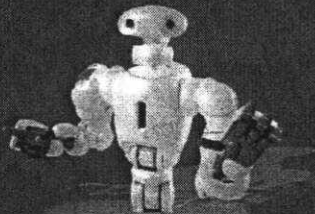
1. Pranai Kungwalrut, Maitree Thumma, Vittaya Tipsuwanporn, Arjin Numsomran and Pisit Boonsrimuang. “**Design MRAC PID Control for Fan and Plate Process**” PROCEEDINGS OF THE SOCIETY OF INSTRUMENT AND CONTROL ENGINEERING ANNUAL CONFERENCE 2011., September 13-18, 2011, Waseda University, Tokyo, Japan

**SICESO**  
The Society of Instrument and Control Engineers since 1961

[Click to Open SICE 2011 Final Program and Papers](#)

# SICE 2011

50th Anniversary of The Society of Instrument and Control Engineers



**SICE Annual Conference 2011 Final Program and Abstracts**  
September 13-18, 2011, Waseda University, Tokyo, Japan

SICE ♦ IEEE  ISA  WASEDA University

## Design MRAC PID Control for Fan and Plate Process

Pranai Kungwalrut<sup>1</sup>, Maitree Thumma<sup>1</sup>, Vittaya Tipsuwanporn<sup>1</sup>, Arjin Numsomran<sup>1</sup>  
Pisit Boonsrimuang<sup>2</sup>

<sup>1</sup>Department of Instrumentation and Control Engineering, Faculty of Engineering,

<sup>2</sup>Department of Telecommunication Engineering, Faculty of Engineering,

King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand

(Tel : +66-81-925-0356; E-mail: knarjin@kmitl.ac.th)

**Abstract:** This paper presents the method to achieve the design MRAC PID control for fan and plate process. The fan and plate process is another challenged problem for control system design because it has non-linearity characteristic and highly sensitivity for ambient disturbance. Thus tuning of PID parameter performs hardly by root locus or trial & error method. The proposed method presents design methodology of auto-adjustable PID controller using MRAC technique for solving the problems. The adjustable PID parameters corresponding to changes in plant and disturbance will be determined by referring to the reference model specifying the properties of desired control system. Therefore, this technique is convenient to control the process for satisfying the requirement of the system performance.

**Keywords:** MRAC PID Control, Fan and Plate

### 1. INTRODUCTION

Since the fan and plate process has high non-linear characteristic and high sensitivity for ambient disturbance, [1] the PID control could not satisfy the performance specification of fan and plate system by constant PID gain control at some operating point. An adaptive controller, such as a Model reference adaptive control, that could learn and accommodate changes in plant dynamics is needed.

Model reference adaptive control was suggested by Whitaker et al. in [2] to solve the autopilot control problem. The reference model in MRAC scheme is chosen to generate the desired trajectory,  $y_m$ , that the plant output  $y_p$  has to follow. The tracking error  $e$  represents the deviation of the plant output from the desired trajectory. The closed-loop plant is made up of an ordinary feedback control law that contains the plant and a controller and an adjustment mechanism that generates the controller parameter estimates real time.

The proposed methodology of auto-adjustable PID controller using MRAC technique is an effective scheme for control the output response of the systems that have drastic changing in their dynamics. The experimental results show that MRAC PID control can satisfy the performance specifications of fan and plate control system.

The paper introduces summary details of fan and plate in section 2. Section 3 gives the MRAC PID controller design procedure. Section 4 describes the simulations and experimental results. Finally, conclusions are given in section 5.

### 2. THE FAN AND PLATE PROCESS

According to Fig. 1, this process consists of two essential parts, the fan driven by DC. Motor that functions as input and the plate functions as output of the process. The angular position of the plate varies upon the motor speed and can be measured by a potentiometer that employed electrical signal via

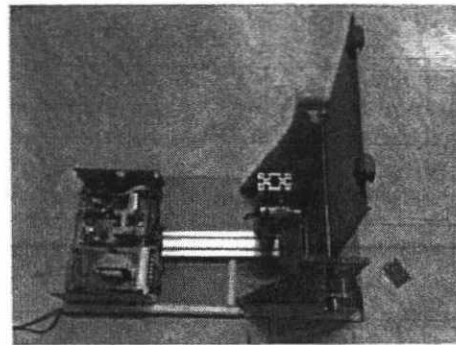


Fig. 1 The fan and plate process

voltage divider circuit. This signal has to calibrate before using as feed-back signal of control system.

Fig. 2 shows the control system structure of the fan and plate process. The first block is fan motor driver circuit. The second block is air stream dynamics which is high variation because it is difficult to know the characteristic of air flow. The last one is plate dynamics sending electrical signal of angular position by potentiometer and voltage divider circuit. Conclusively, there is very high variation in whole parts of process.

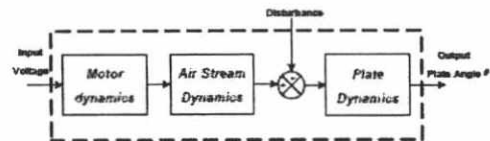


Fig. 2 Block diagram of the fan and plate process

The fan and plate is highly non-linear system because this plant is open process that is highly sensitivity for ambient disturbance. Furthermore, when the set point is close to zero degree (perpendicularity with the earth), an angular position control is very difficult because of the disturbance from the earth gravity.

### 3. MRAC PID CONTROLLER DESIGN

The block diagram in Fig. 3 shows the structure of a model reference adaptive control (MRAC) system composing with process, controller, reference model and adjustment mechanism block.

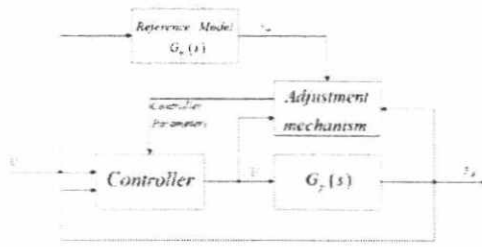


Fig. 3 Block diagram of a model-reference adaptive control (MRAC) system.

The model reference adaptive control (MRAC) Technique is based on information  $y_m, y_p, u$  and  $u_c$  used for devising a controller. The adjustment mechanism automatically adjusts controller parameters so that the behavior of the closed-loop control plant output  $y_p$  closely follows that ( $y_m$ ) of the reference model. Parameters and structure of reference model are specificities on base of requirements of control performance.

The adjustment mechanism of MRAC system constructs by adaptive control rule, called MIT rule which performs the algorithms as following.

Tracking error

$$e = y_p - y_m \tag{1}$$

Form cost function

$$J(\theta) = \frac{1}{2} e^2(\theta) \tag{2}$$

MIT Rule says that the time rate of change of  $\theta$  is proportional to negative gradient of  $J$ . That is

$$\frac{d\theta}{dt} = -\gamma \frac{\partial J}{\partial \theta} = -\gamma e \frac{\partial e}{\partial \theta} \tag{3}$$

where  $e$  denotes the model error and  $\theta$  is the controller parameter vector. The components of  $\frac{\partial e}{\partial \theta}$  are

the sensitivity derivatives of the error with respect to  $\theta$ . The parameter  $\gamma$  is known as the adaptation gain. The MIT rule is a gradient scheme that aims to minimize the squared model cost function [2].

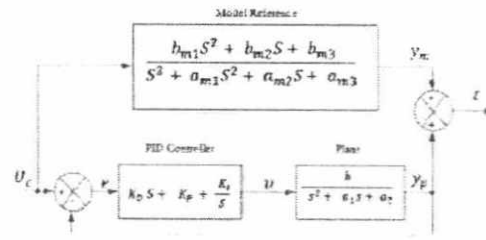


Fig. 4 structure of MRAC control system

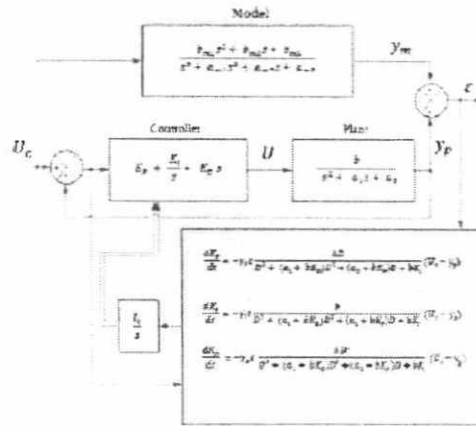


Fig. 5 MRAC Block Diagram for Fan and Plate Process

Consider a system described by 2<sup>nd</sup> order model  $b/(s^2 + a_1s + a_2)$ . A block diagram of control system depicts in Fig. 4.

Closed loop transfer function is

$$\frac{Y_p(s)}{U_c(s)} = \frac{b(K_d s^2 + K_p s + K_i)}{s(s^2 + a_1 s + a_2) + b(K_d s^2 + K_p s + K_i)} \tag{4}$$

and

$$Y_p(s) = \frac{b(K_d s^2 + K_p s + K_i)}{s^3 + (a_1 + bK_d)s^2 + (a_2 + bK_p)s + bK_i} U_c(s) \tag{5}$$

From (5) and required performance of system, we obtain a reference model as (6).

$$\frac{Y_m(s)}{U_m(s)} = \frac{b_{m1}s^2 + b_{m2}s + b_{m3}}{s^3 + a_{m1}s^2 + a_{m2}s + a_{m3}} \tag{6}$$

Apply MIT gradient rules for determining the value of PID controller parameters  $K_p, K_i, K_d$  in (7).

$$\begin{aligned} \frac{dK_p}{dt} &= -\gamma_p \frac{\partial J}{\partial K_p} = -\gamma_p \left( \frac{\partial J}{\partial \epsilon} \right) \left( \frac{\partial \epsilon}{\partial y_p} \right) \left( \frac{\partial y_p}{\partial K_p} \right) \\ \frac{dK_i}{dt} &= -\gamma_i \frac{\partial J}{\partial K_i} = -\gamma_i \left( \frac{\partial J}{\partial \epsilon} \right) \left( \frac{\partial \epsilon}{\partial y_p} \right) \left( \frac{\partial y_p}{\partial K_i} \right) \\ \frac{dK_d}{dt} &= -\gamma_d \frac{\partial J}{\partial K_d} = -\gamma_d \left( \frac{\partial J}{\partial \epsilon} \right) \left( \frac{\partial \epsilon}{\partial y_p} \right) \left( \frac{\partial y_p}{\partial K_d} \right) \end{aligned} \tag{7}$$

Where  $\partial J/\partial \varepsilon = s$ ,  $\partial \varepsilon/\partial y = 1$ ,  $D = d/dt$

$$\frac{\partial y_f}{\partial K_p} = \frac{bD}{D^2 + (a_1 + bK_d)D^2 + (a_2 + bK_p)D + bK_i} [U_c - y_f] \quad (8)$$

$$\frac{\partial y_f}{\partial K_i} = \frac{b}{D^2 + (a_1 + bK_d)D^2 + (a_2 + bK_p)D + bK_i} [U_c - y_f] \quad (9)$$

$$\frac{\partial y_f}{\partial K_d} = \frac{bD^2}{D^2 + (a_1 + bK_d)D^2 + (a_2 + bK_p)D + bK_i} [U_c - y_f] \quad (10)$$

From Eq. (8), (9) and (10), it obtained  $\frac{\partial K_p}{\partial t}$ ,  $\frac{\partial K_i}{\partial t}$ ,  $\frac{\partial K_d}{\partial t}$

as Eq. (11).

$$\begin{aligned} \frac{dK_p}{dt} &= -\gamma_f \frac{\partial J}{\partial K_p} \\ &= -\gamma_f e \frac{bD}{D^2 + (a_1 + bK_d)D^2 + (a_2 + bK_p)D + bK_i} [U_c - y_f] \\ \frac{dK_i}{dt} &= -\gamma_i \frac{\partial J}{\partial K_i} \\ &= -\gamma_i e \frac{b}{D^2 + (a_1 + bK_d)D^2 + (a_2 + bK_p)D + bK_i} [U_c - y_f] \\ \frac{dK_d}{dt} &= -\gamma_d \frac{\partial J}{\partial K_d} \\ &= -\gamma_d e \frac{bD^2}{D^2 + (a_1 + bK_d)D^2 + (a_2 + bK_p)D + bK_i} [U_c - y_f] \end{aligned} \quad (11)$$

Then we define

$$a_{m1} = a_1 + bK_d; \quad a_{m2} = a_2 + bK_p; \quad a_{m3} = bK_i \quad (12)$$

A block diagram in Fig. 5 shows the implementation of MRAC technique to fan and plate control system.

#### 4. EXPERIMENTAL RESULTS

This section describes the experimental results of fan and plate control system starting with plant parameters determining, control system simulation and experiment with real process using MATLAB program.

Block diagram of a fan and plate control system is shown in Fig.6. The controller is performed by Real Time Window Target and MATLAB Simulink program. The PV Signals are sent via PCI1711 Interface card to actuator which consists of a signal conditioning circuit, ARM7 board and motor drive circuit.

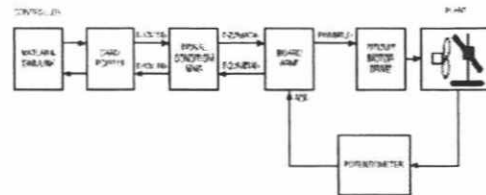


Fig. 6 Block diagram of a fan and plate control system using MRAC Technique

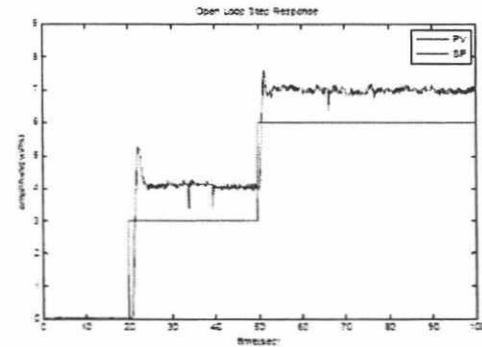


Fig. 7 Open Loop Step Response of Fan and Plate process

Transfer function of fan and plate system in this paper is employed by the open loop step response which is another easy method that the transfer function of the process can be obtained without knowing the specific physical quantities of the process. The result of open loop step response is shown in Fig.7.

From the step test results,  $M_p = 0.2$ ,  $2\%$ ,  $t_s = 5s$  then we received the calculated plant parameters as following  $\zeta = 0.582$ ,  $\omega_n = 1.37$ . Plant transfer function is

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n + \omega_n^2} = \frac{1.87}{s^2 + 1.6s + 1.87} \quad (13)$$

The form of plant transfer function is

$$G_p(s) = \frac{b}{s^2 + a_1s + a_2} \quad (14)$$

Design PID controllers using MRAC Technique as the control structure in Fig.5. From the specified properties of the control system  $P.O. = 5\%$  and  $t_s = 50 \text{ sec}$ , a reference model is shown in (15).

$$\frac{Y_m(s)}{U_m(s)} = \frac{s^2 + s + 2}{s^3 + 2s^2 + 5s + 2} \quad (15)$$

Determining the value of PID controller parameters  $K_p, K_i, K_d$  by MRAC algorithm, we defined the value of adaptation gain  $\gamma$  as following

$$\gamma_p = -0.001, \quad \gamma_i = -0.001 \quad \text{and} \quad \gamma_d = -0.001$$

Block diagram of simulation in Matlab Program is shown in Fig.8. The simulation done by controlling the response of fan and plate control system  $y_p$  following the set-point amplitude 3 and 6 volts at 0 sec and 50 sec respectively. The simulation results are shown in Fig. 9-Fig. 10.

Experimental block diagram for Fan and Plate Control System is shown in Fig.11. We defined the value of adaptation gain  $\gamma$  as following

$$\gamma_p = -0.0005, \gamma_i = -0.0005 \text{ and } \gamma_d = -0.0005$$

The experiment done by controlling the response of fan and plate control system  $y_p$  following the set-point amplitude 3, 6, 3 and 6 volts at 0 sec, 50 sec, 100 sec and 150 sec respectively.

The experimental result is shown in Fig. 12 and Fig. 13 is the response of MRAC PID control system for fan and plate control system with disturbance.

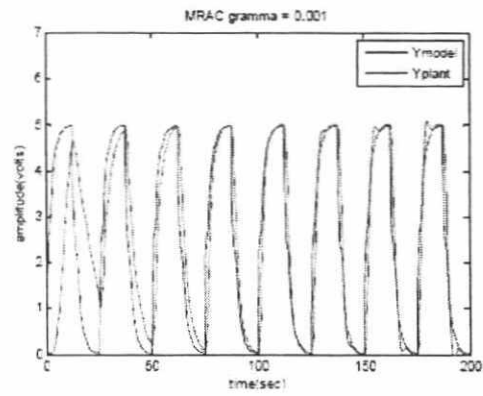


Fig. 10 MV of MRAC PID Control System for Fan and Plate Control System

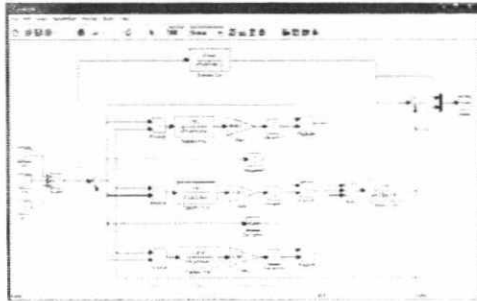


Fig. 8 MRAC Simulation Block Diagram for Fan and Plate Control System

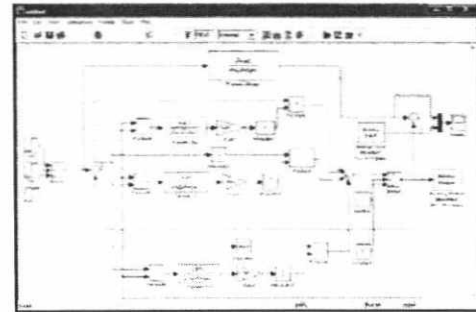


Fig. 11 MRAC Experimental Block Diagram for Fan and Plate Control System

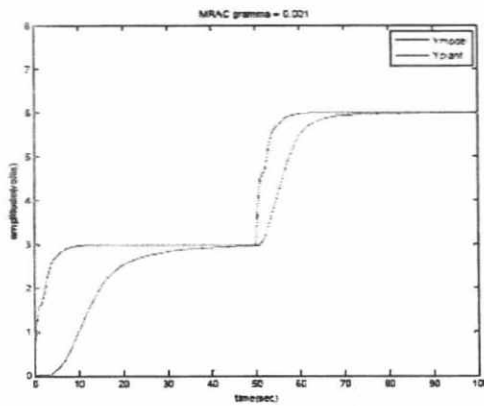


Fig. 9 Response of MRAC PID Control System for Fan and Plate Control System

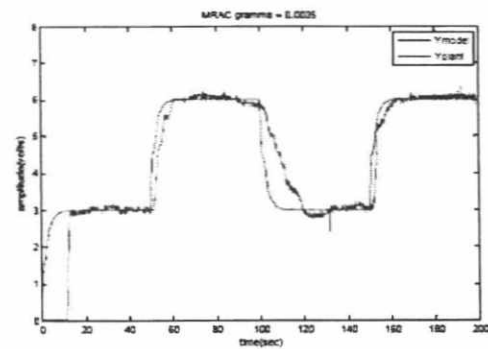


Fig. 12 Response of MRAC PID Control System for Fan and Plate Control System

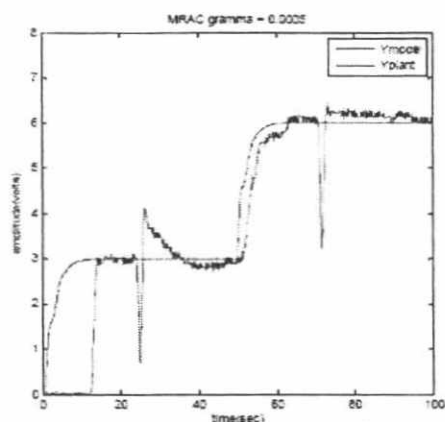


Fig. 13 Response of MRAC PID Control System for Fan and Plate Control System with disturbance

## 5. CONCLUSION

The experimental results reveal us that the proposed method is an effective scheme for control the output response of fan and plate control system. The design of PID controller using MRAC techniques for fan and plate process can conveniently adjust controller parameters corresponding to changes of plant and disturbance. It can control the output response satisfying the specification of control system requirement.

## ACKNOWLEDGEMENT

The author would like to acknowledge her sincere thank to the Industry/University Cooperative Research Center in Data Storage Technology and Application, King Mongkut's Institute of Technology Ladkrabang and National Electronics and Computer Technology Center, National Science and Technology Development Agency.

## REFERENCES

- [1] A. Numsomran, V. Tipsuwanporn, The design of robust PID control for Fan and Plate process, *Proceeding CA '07 Proceedings of the Ninth LASTED International Conference on Control and Applications*, CA, ©2007
- [2] Osburn, P.V., A.P. Whitaker and A. Kezer, "New Developments in the Design of Model Reference Adaptive Control Systems," Paper No.61-39, Institute of the Aerospace Sciences, 1961.
- [3] M. Zhuang, and D.P. Atherton, PID controller design for a TITO system, *IEE Proc - Control Theory Appl.*, Vol.141, No.2, pp.111-120, 1994
- [4] J.G. Ziegler and N.B. Nichols, Optimum settings for automatic controllers, *Trans. ASME*, vol.65, 1943, pp.433-444
- [5] K.J. Astrom and T. Hagglund, Automatic tuning of simple regulators, *Proceeding of IFAC 9<sup>th</sup> World Congr.*, Budapest, Hungary, 1984, 1867-1872
- [6] J. Dorsey, *Continuous and Discrete Control Systems*, McGraw Hill, 2002
- [7] A. Numsomran, 2-DOF Control System Designed by Root Locus Technique, *Proceeding of KACC control 15<sup>th</sup>*, Korea, October 2000
- [8] Astrom K.J., and T. Hagglund, *PID Controllers: Theory, Design, and Tuning*, Instrument, Society of America 2<sup>nd</sup> edition.
- [9] Q.C. Wang, B. Zou, T.H. Lee and Q. Bi, Auto tuning of multivariable PID controllers from decentralized relay feedback, *Automatica*, Vol.33, No.3, pp. 319-330, 1997

## ประวัติผู้เขียน

นายพระนาย กังวาลรัตน์ เกิดเมื่อวันที่ 14 กรกฎาคม พ.ศ. 2517 ที่จังหวัดสุพรรณบุรี สำเร็จการศึกษาระดับประกาศนียบัตรวิชาชีพชั้นสูง สาขาช่างอิเล็กทรอนิกส์ จากสถาบันเทคโนโลยีราชมงคล วิทยาเขตเทคนิคกรุงเทพ ต่อจากนั้นได้เข้าศึกษาต่อระดับปริญญาตรีในสาขาวิศวกรรมไฟฟ้าสื่อสาร ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี สำเร็จการศึกษาในปีการศึกษา 2538 และเข้าศึกษาต่อในระดับปริญญาโท หลักสูตรบริหารธุรกิจมหาบัณฑิต สาขาบริหารธุรกิจนานาชาติ คณะบริหารธุรกิจ มหาวิทยาลัยอีสเทิร์นเอเซีย สำเร็จการศึกษาในปีการศึกษา 2544 และเข้าศึกษาต่อในระดับปริญญาโท หลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมการวัดคุม ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังในปีการศึกษา 2552

ประวัติการทำงาน ในปีพ.ศ. 2539 ได้เข้าเป็นพนักงานของบริษัท *ยูไนเต็ดคอมมูนิเคชั่น อินดัสตรี จำกัด (มหาชน) (เน็ตเวิร์ค)* ผู้ออกแบบและติดตั้งระบบสื่อสารโทรคมนาคม ([www.ucom.co.th](http://www.ucom.co.th)) โดยเริ่มต้นทำงานในตำแหน่งวิศวกรและวิศวกรขาย ตามลำดับ หลังจากนั้น ในปีพ.ศ. 2541 ได้เข้าทำงานในบริษัท เอ็ม.วี.ที. คอมมิวนิเคชั่น จำกัด (มหาชน) ผู้ให้บริการครบวงจรสำหรับระบบสื่อสารโทรคมนาคม ([www.mvt.co.th](http://www.mvt.co.th)) ในตำแหน่งผู้จัดการฝ่ายขาย

ต่อมาในปีพ.ศ. 2544 ได้ก่อตั้งบริษัท เพาเวอร์ โซลูชั่น เทคโนโลยี จำกัด ผู้จัดจำหน่ายและให้บริการครบวงจรด้านระบบสำรองไฟฟ้าทุกชนิดและธุรกิจด้านพลังงานทดแทน ([www.pst.co.th](http://www.pst.co.th))

ปัจจุบันดำรงตำแหน่งทางบริหาร ดังนี้

- กรรมการผู้จัดการบริษัท เพาเวอร์ โซลูชั่น เทคโนโลยี จำกัด
- กรรมการบริษัท เอ็ม. วี. ที. คอมมิวนิเคชั่น จำกัด (มหาชน)
- กรรมการบริษัท กันหา โซล่าพาวเวอร์ เจ้าของโครงการโรงไฟฟ้าพลังงานแสงอาทิตย์ชนิดเชื่อมโยงเข้ากับระบบจำหน่ายไฟฟ้าของการไฟฟ้าส่วนภูมิภาค (กฟภ.) ([www.kanhasolarpower.com](http://www.kanhasolarpower.com))
- ที่ปรึกษาบริษัท พระพาย เทคโนโลยี จำกัด ผู้ผลิตและออกแบบกังหันลมผลิตไฟฟ้าชนิดความเร็วลมต่ำ ([www.prapai.co.th](http://www.prapai.co.th))