

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

แผนกงานเจ้าหน้าที่งาน

ROBOT ARM

โดย

นายสรายุทธ์ กล่าวพิคตร์ 46015325

นายสันติ รักษ์ความสุข 46015326

เลขหมู่.....
เลขทะเบียน..... 62345
วัน,เดือน,ปี 16 ส.ค. 2549

อาจารย์ที่ปรึกษา
อาจารย์ สุมิตร พนาอุดมทรัพย์

b..... 11621369
i.....

ปฏิญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมระบบควบคุม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2548

ปริญญาโท ปีการศึกษา 2548

สาขาวิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง แขนกลเจาะชิ้นงาน

ผู้จัดทำ

1. นายสรายุทธ์ กล่าวพัคตร์ รหัสประจำตัว 46015325

2. นายสันติ รัักษ์ความสุข รหัสประจำตัว 46015326



แผนกเฉพาะชิ้นงาน

นายสรายุทธ์ กล้าพักตร์

นายสันติ รัักษ์ความสุข

อาจารย์ที่ปรึกษา

อาจารย์ สุมิตร พนาอุดมทรัพย์

ปีการศึกษา 2548

บทคัดย่อ

เป็นการพัฒนาการควบคุมแขนกลให้สามารถเคลื่อนที่ได้พร้อมกันทั้งสามแกน การเคลื่อนที่ของแขนกลจะสัมพันธ์ต่อกัน รวมทั้งเพื่อลดความผิดพลาดในการเคลื่อนที่ให้เหลือน้อยที่สุด โดยได้นำคอมพิวเตอร์มาเป็นตัวประมวลผลทางพิกัดตำแหน่งของการเคลื่อนที่ ส่วนโปรแกรมที่ใช้ควบคุมถูกเขียนขึ้นมาจากภาษาซี เพราะจะง่ายในการออกแบบและเลือกใช้ตัวควบคุม ภาษาซีเขียนลงในตัวปิก (PIC) เพื่อรับส่งค่ากับโปรแกรมในคอมพิวเตอร์ที่เขียนด้วยโปรแกรมวิซวลเบสิก คอมพิวเตอร์สามารถติดต่อกับอุปกรณ์ควบคุมทางอิเล็กทรอนิกส์ โดยผ่านทางการ์ดอินเตอร์เฟส การควบคุมเลือกการควบคุมแบบพีไอดี เนื่องจากตัวควบคุมแบบนี้ลดค่าผิดพลาดในการเคลื่อนที่ของแขนกลได้ ตัวแขนกลมีทั้งหมดสี่ข้อต่อ คือ ฐานหมุน แขนท่อนที่หนึ่ง ที่สองและหัวเจาะ

ROBOT ARM

Sarayut Klumpuk

Santi Lukkwamsok

Advisor

Asst.Prof.Sumit Panaoudomsub

Abstract

This thesis presents the robot arm which can move all four joints, which are a rotate base, the first and second parts of robot arm and a drill onto three planes in the same time and has few errors. A computer is used to compute the specific point of the robot arm. C language is written on a micro controller (PIC) for controlling the robot arm. The computer interacts to the micro controller and electronic devices by Visual basic language and a interface card respectively. In this project, The PID controller was chosen to control the robot arm because it effects the fewest errors.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยดี ก็เพราะได้รับการสนับสนุนจากบุคคลหลายท่าน โดยทางผู้จัดทำขอขอบพระคุณ ท่านอาจารย์สุมิตร พนาอุคมทรัพย์ อาจารย์ที่ปรึกษาที่ให้คำแนะนำ และคำปรึกษาในการแก้ไขปัญหาดังกล่าว ทำให้งานสำเร็จลุล่วงไปได้ด้วยดี

ขอขอบพระคุณ ผ.ศ.พรสุข รัตโรจน์นันต์ อาจารย์ รัชชัย คำศรี และอาจารย์สองเมือง นันทขว้างที่ให้ความกรุณา และคำแนะนำที่เป็นประโยชน์ ตลอดจนให้ความสนใจใส่ดูแลสอบถามความก้าวหน้าอย่างสม่ำเสมอพร้อมทั้งคณาจารย์ทุกท่าน ที่ได้ประสิทธิ์ประสาทความรู้ในด้านต่างๆ และให้คำปรึกษาแก่ผู้จัดทำ ผู้จัดทำขอขอบพระคุณเป็นอย่างสูงมา ณ โอกาสนี้

ขอขอบคุณ ภาควิชาวิศวกรรมระบบควบคุม ที่มีเครื่องมือและอุปกรณ์ในการทำงานให้ใช้อย่างครบครัน ขอขอบคุณพี่ๆ ตุรการ์ที่ให้ความสะดวกในเรื่องต่างๆ

ขอขอบคุณเพื่อนๆ ทุกคนที่คอยให้กำลังใจ และให้คำแนะนำดีๆ ตลอดการทำปริญญานิพนธ์ฉบับนี้

สุดท้ายนี้ผู้จัดทำขอกราบขอบพระคุณบิดา มารดา ของคณะผู้จัดทำ ที่ท่านได้ให้การอุปการะมาอย่างดีเยี่ยม และส่งเสริมให้เล่าเรียนมาจนได้มีวันนี้ อีกทั้งยังคอยให้กำลังใจแก่ผู้จัดทำตลอดเวลา

นาย สรายุทธ์ กล้าพิภคร์

นาย สันติ รักความสุข

ผู้จัดทำ

สารบัญ

	หน้า
บทคัดย่อ	I
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูปภาพ	VI
สารบัญตาราง	IX
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 การแบ่งประเภทของแขนกล	3
2.2 คิเนมาติกส์และ ไดนามิกส์	10
2.3 ลักษณะเฉพาะของหุ่นยนต์	12
2.4 หลักการและทฤษฎีของมอเตอร์ดีซี	16
2.5 หลักการทำงานของจอร์ H-Bridge Switching	25
2.6 ตัวควบคุม(Controller)	26
2.7 ไมโครคอนโทรลเลอร์ PIC 16F877	37
2.8 การทำงานในโหมด PWM	40
2.9 อนุภาคต้น สัญญาณ RS-232	43
บทที่ 3 การออกแบบ	45
3.1 โครงสร้างและการออกแบบทางกล	45
3.2 โครงสร้างและการออกแบบอิเล็กทรอนิกส์	55
3.3 ส่วนของโครงสร้างโปรแกรม	60
บทที่ 4 การทดลอง	65
4.1 การทดลองเพื่อทดสอบการดำเนิน PWM ของวงจรถูก Driver	65
4.2 การทดลองเพื่อทดสอบการทำงานของชุด Driver	67
4.3 การทดลองแขนกล	68

สารบัญ(ต่อ)

	หน้า
บทที่ 5 สรุปและวิจารณ์	74
5.1 สรุปผลการดำเนินงาน	74
5.2 แนวทางในการแก้ไข	74
5.3 แนวทางในการพัฒนาต่อ	75

ภาคผนวก

ภาคผนวก ก. โปรแกรมการใช้งาน

เอกสารอ้างอิง



สารบัญรูปภาพ

	หน้า
รูปที่ 2.1 แสดงลักษณะทิศทางเคลื่อนที่ของแขนกล	3
รูปที่ 2.2 แขนกลที่มีลักษณะคล้ายแขนมนุษย์	4
รูปที่ 2.2-a Gantry Robot	4
รูปที่ 2.2-b Work envelope of Gantry Robot	5
รูปที่ 2.3-a Cylindrical Robot	6
รูปที่ 2.3-b Work envelope of Cylindrical Robot	6
รูปที่ 2.4-a Spherical Robot	7
รูปที่ 2.4-b Work envelope of Spherical Robot	7
รูปที่ 2.5-a Scara Robot	8
รูปที่ 2.5-b Work envelope of Scara Robot	8
รูปที่ 2.6-a Articulated Robot	9
รูปที่ 2.6-b Work envelope of Articulated Robot	9
รูปที่ 2.7 แสดงการควบคุมแบบวงเปิด	11
รูปที่ 2.8 แสดงการทำงานแบบวงปิด	12
รูปที่ 2.9 ระยะที่เอื้อมถึงและสโตรคของหุ่นยนต์ทรงกระบอก	12
รูปที่ 2.10 yaw-pitch-roll ของ tool	13
รูปที่ 2.11 ลำดับการหมุนแกน	14
รูปที่ 2.12-ก ระบบ YPR	14
รูปที่ 2.12-ข ระบบ YPR	15
รูปที่ 2.13 ระบบ RPY	15
รูปที่ 2.14 มอเตอร์คอมปาวด์แบบคิวมูเลทีฟ	17
รูปที่ 2.15 มอเตอร์คอมปาวด์แบบคิฟเฟอเรนเชียล	17
รูปที่ 2.16 ทิศทางของแรงเคลื่อนไฟฟ้ากลับ	18
รูปที่ 2.17 ความสัมพันธ์ระหว่างกระแสและแรงเคลื่อนไฟฟ้ากลับในวงจรฮามเจอร์	19
รูปที่ 2.18 โมเมนต์ซึ่งลวดตัวนำฮามเจอร์กระทำรอบจุดศูนย์กลาง	22
รูปที่ 2.19 วงจรสวิตช์แบบ H-Bridge	25
รูปที่ 2.20 วงจรสวิตช์แบบ H-Bridge เมื่อ On สวิตช์ S1 และ S3	25

สารบัญรูปภาพ(ต่อ)

	หน้า
รูปที่ 2.21 วงจรสวิตช์แบบ H-Bridge เมื่อOn สวิตช์ S2 และ S4	26
รูปที่ 2.22 การกำหนดช่วงจำกัดของ เอาต์พุต	28
รูปที่ 2.23 ผลการตอบสนองของตัวควบคุมแบบพี	29
รูปที่ 2.24 ระบบควบคุมแบบพี	29
รูปที่ 2.25 ผลการตอบสนองของตัวควบคุมแบบไอ	30
รูปที่ 2.26 ระบบควบคุมแบบไอ	31
รูปที่ 2.27 ผลตอบสนองของระบบ ควบคุมแบบดี	32
รูปที่ 2.28 บล็อกไดอะแกรม (block diagram) ของตัวควบคุมแบบดี	33
รูปที่ 2.29 บล็อกไดอะแกรมของระบบควบคุมแบบพีไอ	33
รูปที่ 2.30 การตอบสนองของระบบควบคุมแบบพีไอ	34
รูปที่ 2.31 บล็อกไดอะแกรมของระบบควบคุมแบบพีดี	36
รูปที่ 2.32 บล็อกไดอะแกรมของระบบควบคุมแบบพีไอดี	36
รูปที่ 2.33 การทำงานของ โมดูล CCP1 เพื่อสร้างสัญญาณ PWM	42
รูปที่ 2.34 เอาต์พุตของสัญญาณ PWM	42
รูปที่ 2.35 รูปคลื่นของสัญญาณที่ส่ง	43
รูปที่ 2.36 แสดงโครงสร้างภายในและตำแหน่งขาต่างๆของ Max232	44
รูปที่ 3.1 แสดงส่วนประกอบของฐานหมุน	45
รูปที่ 3.2 แสดงส่วนประกอบส่วนฐาน	46
รูปที่ 3.3 รูปแสดงการออกแบบส่วนฐาน	46
รูปที่ 3.4 แสดงแกนกลจำลอง	48
รูปที่ 3.5 แสดงภาพของแกนกลจากของจริง	48
รูปที่ 3.6 แสดงส่วนแกนท่อนที่ 1	49
รูปที่ 3.7 แสดงด้านข้างของแกนท่อนที่ 1	49
รูปที่ 3.8 รูปแสดงการออกแบบส่วนเคลื่อนที่ที่หนึ่งของจริง	50
รูปที่ 3.9 แสดงด้านข้างของแกนท่อนที่ 2	51
รูปที่ 3.10 แสดงแกนท่อนที่ 2	51
รูปที่ 3.11 รูปแสดงส่วนการเคลื่อนที่ที่สองจากของจริง	52

สารบัญรูปภาพ(ต่อ)

	หน้า
รูปที่ 3.12 แสดงภาพตัดของหัวเจาะ	53
รูปที่ 3.13 รูปแสดงลักษณะของหัวเจาะ	53
รูปที่ 3.14 แสดงส่วนที่ใช้จับชิ้นงาน	54
รูปที่ 3.15 รูปแสดงแท่นจับชิ้นงาน	54
รูปที่ 3.16 รูปแสดงวงจรรวม	56
รูปที่ 3.17 รูปแสดงวงจรขับมอเตอร์	57
รูปที่ 3.18 รูปแสดงการต่อล่อจิกช่วย	58
รูปที่ 3.19 แสดงส่วนของฐานแขนกล	59
รูปที่ 3.20 รูปแสดงการติดตั้งมอเตอร์	59
รูปที่ 3.21 แผนผังแสดงการออกแบบระบบควบคุมแขนกล	60
รูปที่ 3.22 แผนภาพแสดงขั้นตอนการทำงานในการรับค่าจากเซนเซอร์	61
รูปที่ 3.23 แผนภาพแสดงวิธีการส่งสัญญาณและการจ่ายพัลส์ให้มอเตอร์	62
รูปที่ 3.24 แผนภาพแสดงขั้นตอนการคำนวณค่าสัญญาณพัลส์ที่จะจ่ายให้แก่มอเตอร์	64
รูปที่ 4.1 รูปแสดงสัญญาณที่ได้จากการกระตุ้นของ PWM	65
รูปที่ 4.2 รูปแสดงสัญญาณที่ได้จากการกระตุ้นของ PWM	66
รูปที่ 4.3 กราฟแสดงผลการเคลื่อนที่ของแขนกลข้อต่อๆ	73

สารบัญตาราง

	หน้า
ตารางที่ 2.1 ตัวอย่างขนาดความถี่และความละเอียดของสัญญาณ PWM ที่ความถี่ 20 MHz	43
ตารางที่ 4.1 ตารางแสดงสถานะการทำงานของ Driver	72
ตารางที่ 4.2 ตารางทดสอบการเคลื่อนที่ของแขนกล	73



บทที่ 1

บทนำ

1.1 ความเป็นมา

ในปัจจุบันมีการนำแขนกลมาทำงานแทนมนุษย์กันมากขึ้น โดยเฉพาะในประเทศไทย มีหลายบริษัทได้นำแขนกลเข้ามาช่วยในงานอุตสาหกรรมซึ่งแขนกลดังกล่าวส่วนใหญ่ยังต้องสั่งซื้อจากต่างประเทศทำให้มีราคาสูง ทำให้ต้นทุนการผลิตสินค้าสูงไปด้วย การเลือกใช้แขนกลที่ผลิตภายในประเทศจะช่วยให้ต้นทุนการผลิตสินค้าถูกลง ดังนั้นจึงควรที่จะมีการขยายการศึกษาและการค้นคว้าเกี่ยวกับแขนกลให้มากขึ้น เพื่อที่จะสามารถนำแขนกลไปใช้ประโยชน์ต่างๆในงานอุตสาหกรรมได้อย่างเหมาะสม และสิ้นเปลืองงบการลงทุนให้น้อยที่สุด โดยที่ضمنสมองของคนไทย

งานหลายประเภท มีลักษณะการทำงานที่ซ้ำๆกัน โดยเฉพาะงานประกอบชิ้นงาน ซึ่งไม่ได้มีความซับซ้อนมากนัก น่าจะสามารถใช้เครื่องจักรมาทำงานแทนมนุษย์ได้ และเครื่องจักรที่ว่านี้จะต้องผลิตผลงานได้อย่างสม่ำเสมอ และมีคุณภาพเป็นมาตรฐานเดียวกัน จึงมีความคิดที่จะนำความรู้ที่ได้เรียนมาในด้านวิศวกรรมมาเพื่อแก้ปัญหานี้ ในที่สุดจึงเลือกที่จะใช้แขนกลมาเป็นหัวข้อในการทำโครงงานครั้งนี้

การตัดสินใจเลือกทำโครงงานนี้เพื่อที่จะศึกษาพื้นฐานหลักการการทำงานของแขนกล โดยให้แขนกลประกอบชิ้นงานสมมุติได้อย่างถูกต้อง มีการทำงานแบบอัตโนมัติ และหวังเป็นอย่างยิ่งว่าโครงงานแขนกลชิ้นนี้จะทำให้เกิดแนวคิดที่จะพัฒนาแขนกลให้สามารถทำงานอื่นๆตามที่ต้องการได้ รวมถึงการเพิ่มความสามารถในการทำงานที่ซับซ้อนขึ้นต่อไปในอนาคต

1.2 วัตถุประสงค์ของโครงงาน

- เพื่อแสดงให้เห็นถึงการนำทฤษฎีของแขนกล(Robot Arm) แบบ SCARA มาประยุกต์ใช้ในงานอุตสาหกรรม
- เพื่อศึกษาเกี่ยวกับการใช้งานการเขียน โปรแกรมควบคุมแบบต่างๆ

องค์ประกอบของโครงการ

โครงการนี้ประกอบไปด้วย 2 ส่วน คือ Software และ Hardware

Software

คือการใช้โปรแกรมในส่วนของการทำงานส่งค่าให้กับแขนกล

- ภาษาซี (เขียนลงในตัว PIC 16F877A)
- ภาษา Visual Basic (เขียนลงในตัวคอมพิวเตอร์เพื่อติดต่อกับผู้ใช้)

Hardware

ประกอบไปด้วยส่วนต่างๆดังนี้

- คอมพิวเตอร์
- การ์ดอินเตอร์เฟซ (interface card) ใช้ไมโครคอนโทรลเลอร์ PIC 16F877A
เชื่อมต่อกับคอมพิวเตอร์ด้วย MAX 232
- วงจรขับมอเตอร์กระแสตรง
- มอเตอร์กระแสตรงขนาดแรงดัน 12 V
- ตัวแขนกล

บทที่ 2

ทฤษฎีและหลักการ

2.1 การแบ่งประเภทของแขนกล



เราสามารถแบ่งประเภทของแขนกล โดยพิจารณาจากพื้นฐานใหญ่ดังนี้

1. โครงสร้างภายนอกและการเคลื่อนที่ของแขนกล
2. ทางเดินของแขนกล ซึ่งขึ้นอยู่กับ การควบคุมการเคลื่อนที่
3. ประสิทธิภาพของแขนกล

การพิจารณาแบ่งประเภทของแขนกลในขั้นนี้ เราจะอาศัยเฉพาะพื้นฐานข้อ 1. และข้อ 2. เท่านั้น แม้ว่าพื้นฐานข้อ 3. จะมีความสำคัญมากที่สุด แต่จะต้องอาศัยความรู้ทางด้านจลศาสตร์ (Kinematics) และพลศาสตร์(Dynamics) ขั้นสูง ซึ่งยากต่อการทำความเข้าใจ การแบ่งประเภทแขนกลในระดับนี้ จึงจะไม่นำพื้นฐานข้อ 3. มาพิจารณา

การแบ่งประเภทของแขนกล โดยอาศัยการเคลื่อนที่ของโครงสร้างภายนอก

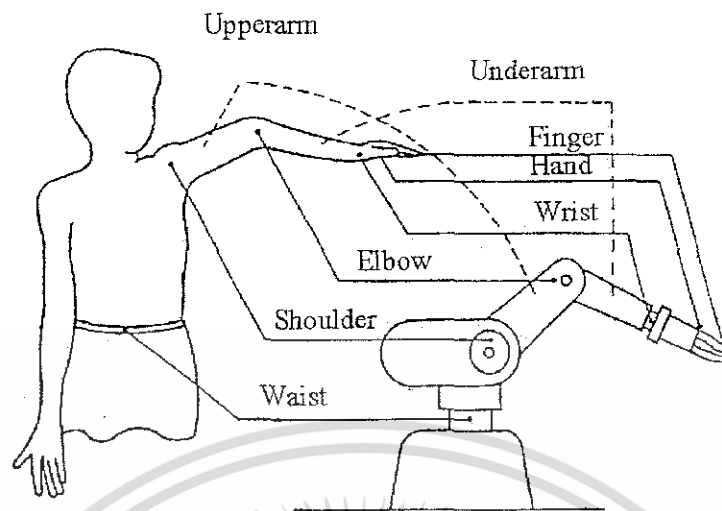
การอธิบายชนิดของหุ่นยนต์ของอธิบายการทำงานของจุดต่อ(Joint) ของหุ่นยนต์อุตสาหกรรมซึ่งในขั้นพื้นฐานมี 2 ชนิดด้วยกัน ดังนี้

การเคลื่อนที่	ลักษณะ	การเคลื่อนที่
Revolute (R)		เป็นการหมุนรอบแกน (Rotary)
Prismatic (P)		เป็นการเคลื่อนที่เชิงเส้น (Linear Motion)

รูปที่ 2.1 แสดงลักษณะทิศทางการเคลื่อนที่ของแขนกล

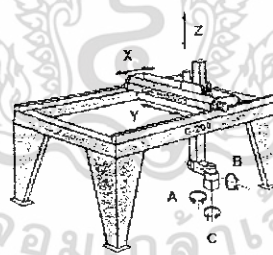
จุดต่อ(joint) ทั้งสองแบบเมื่อนำมาต่อเข้าด้วยกันอย่างน้อย 3 แกนหลักจะได้พื้นที่ทำงาน (work envelope) ที่มีลักษณะแตกต่างกันไป ซึ่งสามารถนำมาแบ่งชนิดของหุ่นยนต์ได้ดังต่อไปนี้

การทำงานของหุ่นยนต์อุตสาหกรรมจะเลียนแบบร่างกายของมนุษย์โดยจะเลียนแบบเฉพาะส่วนของร่างกายที่นำไปใช้ประโยชน์ในอุตสาหกรรมเท่านั้นคือช่วงแขนของมนุษย์ ดังนั้นบางคนอาจจะได้ยินคำว่าแขนกล(Robot arm)ซึ่งหมายถึงหุ่นยนต์อุตสาหกรรม การทำงานของหุ่นยนต์อุตสาหกรรมเปรียบเทียบกับแขนมนุษย์แสดงดังรูปที่ 2.2

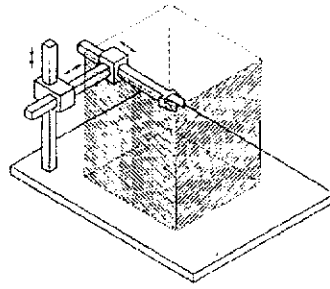


รูปที่ 2.2 แขนกลที่มีลักษณะคล้ายแขนมนุษย์

2.1.1 Cartesian (gantry) Robot แขนทั้ง 3 ของหุ่นยนต์จะเคลื่อนที่เป็นแบบเชิง (Prismatic) ถ้าโครงสร้างมีลักษณะคล้าย Overhead Crane จะเรียกว่าเป็นหุ่นยนต์ชนิด gantry แต่ถ้าหุ่นยนต์ไม่มีขาตั้งหรือขาเป็นแบบอื่น เรียกว่า ชนิด Cartesian



รูปที่ 2.2-a Gantry Robot



รูปที่ 2.2-b Work envelope of Gantry Robot

ข้อดี

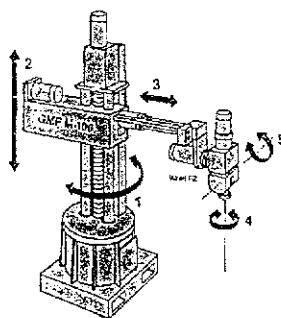
1. เคลื่อนที่ในแนวเส้นตรงทั้ง 3 มิติ
2. การเคลื่อนที่สามารถทำความเข้าใจง่าย
3. มีส่วนประกอบง่าย
4. โครงสร้างแข็งแรงตลอดการเคลื่อนที่

ข้อเสีย

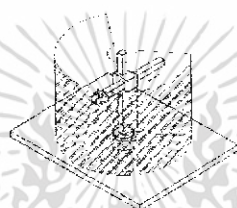
1. ต้องการพื้นที่ติดตั้งมาก
2. บริเวณที่หุ่นยนต์เข้าไปทำงานได้จะเล็กกว่าขนาดของตัวหุ่นยนต์
3. ไม่สามารถเข้าถึงวัตถุจากทิศทางข้างใต้ได้
4. แกนแบบเชิงเส้นจะ seal เพื่อป้องกันฝุ่นละอองและของเหลวได้ยาก

การประยุกต์ใช้งาน เนื่องจากโครงสร้างมีความแข็งแรงตลอดแนวการเคลื่อนที่ ดังนั้นจึงเหมาะกับงานเคลื่อนย้ายของหนักๆ หรือเรียกว่างาน Pick-and-Place เช่น ใช้โหลดชิ้นงานเข้าเครื่องจักร(Machine loading) ใช้เก็บชิ้นงาน(Stacking) นอกจากนี้ยังสามารถใช้ในงานประกอบ(Assembly) ที่ไม่ต้องการเข้าถึงในลักษณะที่มีมุมมน เช่น ประกอบอุปกรณ์อิเล็กทรอนิกส์และงานTestต่างๆ

2.1.2 Cylindrical หุ่นยนต์ประเภทนี้จะมีแกนที่ 2 (ไหล่) และแกนที่ 3 (ข้อศอก) เป็นแบบ prismatic ส่วนแกนที่ 1 (เอว) จะเป็นแบบหมุน(revolute) ทำให้การเคลื่อนที่ได้พื้นที่การทำงานเป็นรูปทรงกระบอกครึ่งรูป



รูปที่ 2.3-a Cylindrical Robot



รูปที่ 2.3-b Work envelope of Cylindrical Robot

ข้อดี

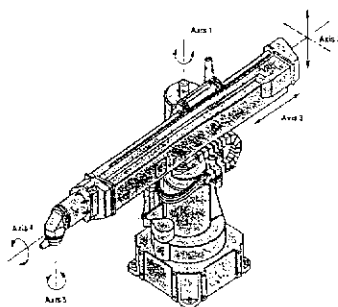
1. มีส่วนประกอบไม่ซับซ้อน
2. การเคลื่อนที่สามารถเข้าใจง่าย สามารถเข้าถึงเครื่องจักรที่มีการเปิด-ปิด หรือเข้าไปในบริเวณที่เป็นช่องหรือโพรงได้ง่าย เช่น การโหลดชิ้นงานเข้า เครื่อง CNC

ข้อเสีย

1. มีพื้นที่ทำงานจำกัด
2. แกนที่เป็นเชิงเส้นมีความยุ่งยากในการseal เพื่อป้องกันฝุ่นและของเหลว

การประยุกต์ใช้งาน โดยทั่วไปจะใช้ในการหยิบยกชิ้นงาน(Pick-and-Place) หรือป้อนชิ้นงานเข้าเครื่องจักรเพราะสามารถเคลื่อนที่เข้าออกบริเวณที่เป็นช่องโพรงเล็กๆ ได้สะดวกๆ

2.1.3 Spherical Robot (Polar) มีสองแกนที่เคลื่อนที่ในลักษณะการหมุน(Revolut Joint) คือแกนที่ 1 (เอว) และ แกนที่ 2 (ไหล่) ส่วนที่ 3 (ข้อศอก) จะเป็นลักษณะของการเคลื่อนที่ในแนวเส้นตรง ดังรูปที่2.4-a ซึ่งทำให้ได้พื้นที่การทำงาน ดังรูปที่ 2.4-b



รูปที่ 2.4-a Spherical Robot

รูปที่ 2.4-b Work envelope of Spherical Robot

ข้อดี

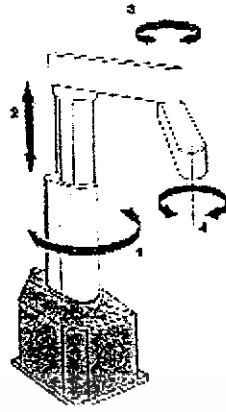
1. มีปริมาตรการทำงานมากขึ้นเนื่องจากการหมุนของแกนที่ 2 (ใหญ่)
2. สามารถลงมาจับชิ้นงานบนพื้นได้สะดวก

ข้อเสีย

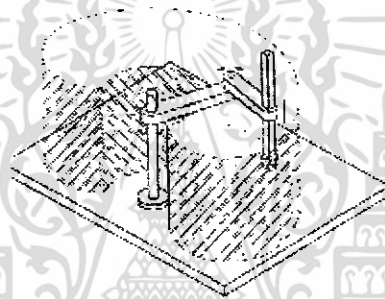
1. มีระบบพิกัด(Coordinate) และส่วนประกอบที่ซับซ้อน
2. การเคลื่อนที่และระบบควบคุมมีความซับซ้อนขึ้น

การประยุกต์ใช้งาน ใช้งานที่มีการเคลื่อนที่ในแนวตั้ง (Vertical) เพียงเล็กน้อย เช่น การโหลดชิ้นงานเข้าออกจากเครื่องปั๊ม (Press) หรืออาจจะใช้งานเชื่อมจุด (Spot Welding)

2.1.4 SCARA Robot หุ่นยนต์ SCARA (Selective Compliance Assembly Robot Arm) จะมีลักษณะแกนที่ 1 (เอว) และแกนที่ 3 (ข้อศอก) หมุนรอบแกนแนวตั้ง และแกนที่ 2 จะเป็นลักษณะการเคลื่อนที่ขึ้นลง (Prismatic) ดังรูปที่ 2.5-a ทำให้ได้พื้นที่การทำงานดัง รูปที่ 2.5-b หุ่นยนต์ SCARA จะเคลื่อนที่ได้รวดเร็วในแนวระนาบ และมีความแม่นยำสูง



รูปที่ 2.5-a Scara Robot



รูปที่ 2.5-b Work envelope of Scara Robot

ข้อดี

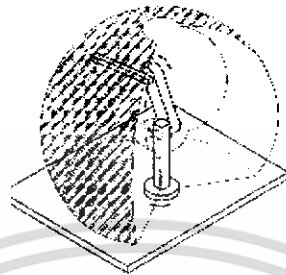
1. สามารถเคลื่อนที่ในแนวระนาบ และขึ้นลงได้รวดเร็ว
2. มีความแม่นยำสูง

ข้อเสีย

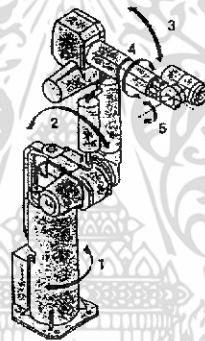
1. มีพื้นที่ทำงานจำกัด
2. ไม่สามารถหมุน(rotation)ในลักษณะมุมต่างๆได้
3. สามารถยกน้ำหนัก(Payload) ได้ไม่มากนัก

การประยุกต์ใช้งาน เนื่องจากการเคลื่อนที่ในแนวระนาบและขึ้นลงได้รวดเร็วจึงเหมาะกับ งานประกอบชิ้นส่วนทางอิเล็กทรอนิกส์ซึ่ง ต้องการความเร็วและการเคลื่อนที่ที่ไม่ต้องการการหมุนมากนัก แต่จะไม่เหมาะกับงานประกอบชิ้นส่วนทางกล(Mechanical part) ซึ่งส่วนใหญ่การประกอบจะอาศัยการหมุน (rotation)ในลักษณะมุมต่างๆ นอกจากนี้ Scara robot ยังเหมาะกับงานตรวจสอบ (Inspection) งานบรรจุภัณฑ์(Packaging)

2.1.5 Articulated Arm (Revolute) ทุกแกนการเคลื่อนที่จะเป็นแบบหมุน (Revolute) รูปแบบการเคลื่อนที่จะคล้ายกับแขนคน ซึ่งจะประกอบด้วยข้องอว ท่อนแขนบน ท่อนแขนล่าง ข้อมือ การเคลื่อนที่ทำให้ได้พื้นที่การทำงาน ดังรูปที่ 2.5-b



รูปที่ 2.6-a Articulated Robot



รูปที่ 2.6-b Work envelope of Articulated Robot

ข้อดี

1. เนื่องจากทุกแกนจะเคลื่อนที่ในลักษณะ ของการหมุนทำให้มีความยืดหยุ่นสูงในการเข้าไปยังจุดต่างๆ
2. บริเวณข้อต่อ(Joint) สามารถ Seal เพื่อป้องกันฝุ่น ความชื้นหรือน้ำ ได้ง่าย
3. มีพื้นที่การทำงานมาก
- 4 .สามารถเข้าถึงชิ้นงานทั้งจากด้านบน ด้านล่าง
5. เหมาะกับการใช้มอเตอร์ไฟฟ้า เป็นจุดขับเคลื่อน

ข้อเสีย

1. มีระบบพิกัด (Coordinate) ที่ซับซ้อน
2. การเคลื่อนที่และระบบควบคุมความเข้าใจได้ยากขึ้น
3. ควบคุมให้เคลื่อนที่ในแนวเส้นตรง (Linear) ได้ยาก
4. โครงสร้างไม่มั่นคงตลอดช่วงการเคลื่อนที่ เพราะบริเวณขอบ Work envelope ปลายแขนจะมีการสั่นทำให้ความแม่นยำลดลง

การประยุกต์ใช้งาน หุ่นยนต์ชนิดนี้สามารถใช้งานได้กว้างขวางเพราะสามารถเข้าถึงตำแหน่งต่างๆ ได้ดี เช่น งานเชื่อม Spot Welding, Path Welding, งานยกของ, งานตัด, งานทากาว, งานที่มีการเคลื่อนที่ต่างๆ เช่น งานพ่นสี งาน sealing ฯลฯ

2.2 คีเนมาติกส์และไดนามิกส์ (Robot Arm Kinematics and Dynamics)

2.2.1 คีเนมาติกส์อาร์ม (Kinematics Arm) เป็นวิธีการวิเคราะห์การเคลื่อนที่และตำแหน่งของแขนกล โดยใช้จุดที่กำหนดเป็นจุดอ้างอิงการเคลื่อนที่ ทั้งนี้การวิเคราะห์หาการเคลื่อนที่แบบคิเนมาติกส์อาร์ม จะไม่สนใจถึง แรง และ โมเมนต์ (Moment) ที่เกิดขึ้นจากการเคลื่อนที่ การวิเคราะห์แบบ คีเนมาติกส์นี้ จะอธิบายการเคลื่อนที่และตำแหน่ง (The Spatical Displacement) ซึ่งเป็นฟังก์ชันของเวลาของแต่ละข้อต่อ (Joint) โดยบอกถึงตำแหน่งและทิศทางของเอ็นเอฟเฟกเตอร์ของแขนกล โดยใช้จุดกำหนดเป็นจุดอ้างอิง

การวิเคราะห์แบบคิเนมาติกส์ มี 2 ประเภท

1. แบบไคเร็ค (Direct) หรือ ฟอว์เวิร์ดคิเนมาติกส์ (Forward Kinematics)
2. แบบ อินเวิร์ดคิเนมาติกส์ (Inverse Kinematics) หรือ อาร์ม โซลูชัน (Arm Solution)

จากการเคลื่อนไหวที่เป็นอิสระต่อกันในแต่ละข้อต่อของแขนกล จึงต้องการกำหนดเป้าหมายที่จะเคลื่อนที่ไป โดยอ้างอิงถึงจุดที่กำหนด (The Reference Coordinate Frame) การหาตัวแปรในแต่ละข้อต่อ อาจใช้วิธีเมตริกซ์แบบ 4×4 โฮโมจีนัส ทรานสฟอร์มเมชัน (Homogeneous Transformation Matrix) ในการอธิบายตำแหน่งความสัมพันธ์ระหว่างจุดหรือลิงค์ ข้างเคียง

โรบอทอาร์มไดนามิกส์ (Robot Arm Dynamics) เป็นการนำเอาสมการสูตรคณิตศาสตร์ มาอธิบายการเคลื่อนที่แบบไดนามิกส์ของแขนกล โดยอาศัยความรู้จากทางกฎฟิสิกส์ (Physical Law) เช่น กฎของนิวตัน และ ลากรองเจียนแมคคานิก (Lagrangian Mechanics)

ทฤษฎีเกี่ยวกับโรบอทอาร์มคิเนมาติกส์ (Robot Arm Kinematics)

แมคคานิกัล แมนิพูเลเตอร์ (Mechanical Manipulator) แบบวงจรมืด ของในแต่ละลิงค์ ซึ่งเชื่อมต่อกันโดยใช้ รีโวลูท (revolute) หรือ ข้อต่อปริสมาคิก (prismatic joint) โดยปลายลิงค์ ด้าน

หนึ่งจะยึดติดกับฐาน(Supporting Base) ในขณะที่ปลายอีกข้างถูกปล่อยให้อิสระและติดกับ เอ็นเอฟเฟกเตอร์ จากความสัมพันธ์การเคลื่อนที่ของข้อต่อ ซึ่งมีผลทำให้เกิดการเคลื่อนที่ของลิงค์ โดยจะอธิบายในรูปแบบตำแหน่งและทิศทาง ซึ่งอ้างอิงกับระบบพิกัดฟิกเรเฟอเรนซ์โคออดิเนต(Fixed Reference Coordinate System)

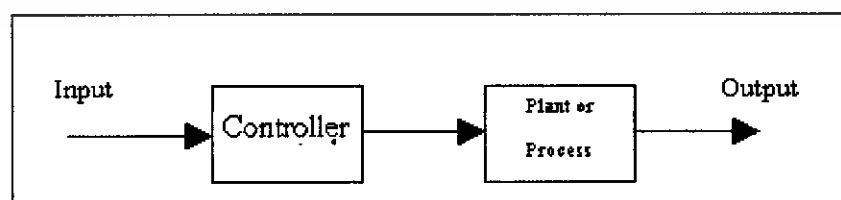
ทฤษฎีและการนำไปใช้จริงในคิเนมาติกส์

1. กรณีกำหนดตัวแปรการเคลื่อนที่ของแต่ละข้อต่อ เช่น มุมการเคลื่อนที่(joint angle vector) $q(t)=(q_1(t),q_2(t),\dots,q_n(t))^T$ เมื่อ n คือ จำนวนระดับของควมอิสระ(degree of freedom) และตัวแปรเรขาคณิตในแต่ละลิงค์ มาเพื่อหาตำแหน่งและทิศทางสุดท้ายของเอ็นเอฟเฟกเตอร์ ของการเคลื่อนที่ โดยอ้างอิงระบบพิกัดอ้างอิง(Reference Coordinate System) กรณีนี้เรียกว่า แบบไดเรกหรือฟอร์เวิร์ดคิเนมาติกส์

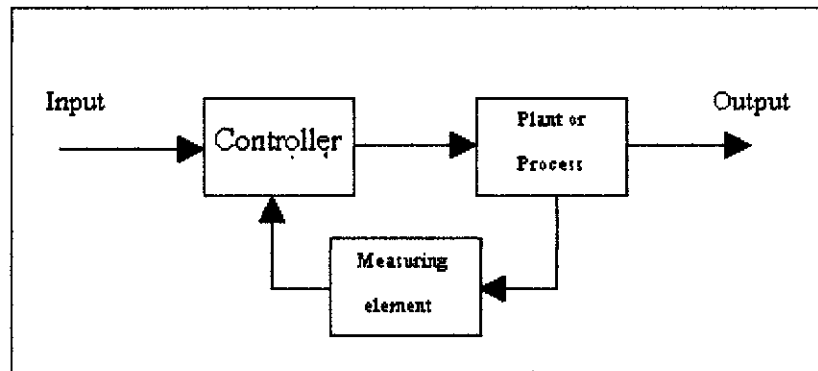
2. กรณีกำหนดตำแหน่งและทิศทางของเอ็นเอฟเฟกเตอร์ ของการเคลื่อนที่และกำหนดตัวแปรเรขาคณิต (Geometric parameters) ของแต่ละลิงค์มา เพื่อหาตัวแปรในการเคลื่อนที่ของแต่ละข้อต่อ เพื่อให้หุ่นยนต์ไปยังจุดที่ต้องการและมีตัวแปรที่ชุดที่ทำให้ไปยังจุดเดียวกันได้ วิธีนี้เรียกว่า แบบอินเวิร์คคิเนมาติกส์ (The Inverse Kinematics (or Arm Solution))

2.2.2 ไดเรกคิเนมาติก (The Direct Kinematics)

เวกเตอร์ และ เมตริกซ์ สามารถนำมาประยุกต์ อธิบายตำแหน่งปัจจุบันของตัวเชื่อม เมื่ออ้างอิงกับฟิกเรเฟอเรนซ์เฟรม (Fixed Reference frame) กับเรเฟอเรนซ์โคออดิเนตเฟรม (Reference Coordinate Frame)เมตริกซ์ 3×3 สามารถนำมาใช้ในการอธิบายการเคลื่อนที่แบบการหมุน(Rotation) ของบอดีแอทแทชเฟรม(The body-attached frame)กับเรเฟอเรนซ์เฟรม(The reference frame)และพิกัดโฮโมจีนัส (Homogeneous Coordinate) จะแทนเวกเตอร์ตำแหน่ง 3 มิติของการเคลื่อนที่ของเมตริกซ์แปลงโฮโมจีนัสขนาด 4×4 (A 4×4 Matrix Homogenous Transformations)จะอธิบายการทำงานแบบเลื่อน(Translational) และหมุน (Rotation)



รูปที่ 2.7 แสดงการควบคุมแบบวงเปิด

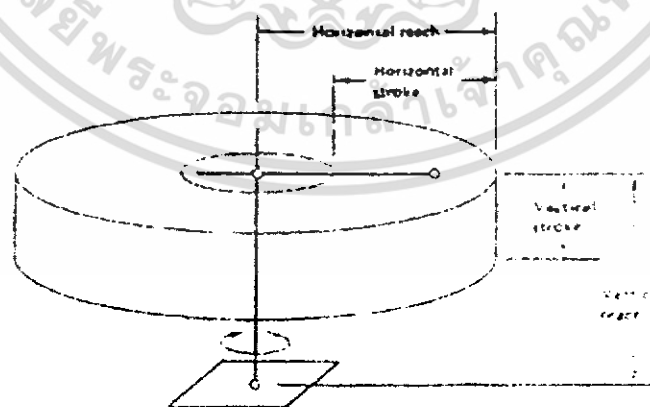


รูปที่ 2.8 แสดงการทำงานแบบวงปิด

2.3 ลักษณะเฉพาะของหุ่นยนต์ (Robot Specification)

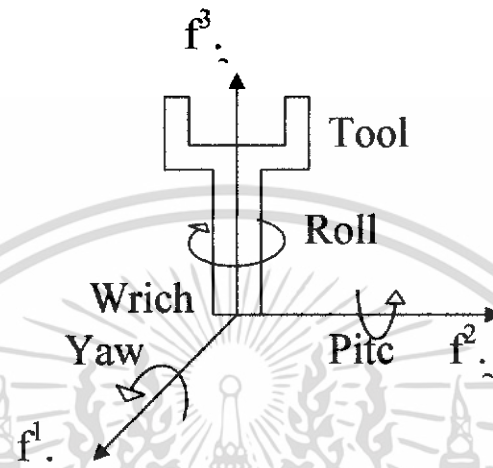
ระยะที่เอื้อมถึงและสโตรค (Reach and Stroke) เป็นการวัดขนาดของพื้นที่ทำงาน (Work Envelop) อย่างคร่าวๆ ระยะที่เอื้อมถึงในแนวนอน (Horizontal Reach) คือระยะทางตามแนวรัศมีที่มากที่สุดที่ข้อมือ สามารถไปถึง โดยวัดจากแกนแนวตั้งที่ทำให้หุ่นยนต์หมุน สโตรคแนวนอน (Horizontal Stroke) คือระยะทางตามแนวรัศมี ทั้งหมดที่ข้อมือสามารถไปได้ดั่งนั้น ระยะที่เอื้อมถึงในแนวนอนลบด้วยสโตรคแนวนอน คือระยะทางตามแนวรัศมีที่น้อยที่สุดที่ข้อมือ สามารถไปได้ โดยวัดจากแกนฐาน ซึ่งมีค่าเป็นบวกเสมอ

ตัวอย่างเช่น ระยะที่เอื้อมถึงในแนวนอน (Horizontal Stroke) ของหุ่นยนต์พิกัดทรงกระบอกก็คือรัศมีของทรงกระบอกอันนอก ของพื้นที่การทำงาน ในขณะที่ สโตรคแนวนอน (Horizontal Stroke) คือผลต่างระหว่างรัศมีของทรงกระบอกอันในและอันนอก ดังแสดงในรูปที่ 2.9



รูปที่ 2.9 ระยะที่เอื้อมถึงและสโตรคของหุ่นยนต์ทรงกระบอก

การหมุนของ tool (Tool Orientation) ในขณะที 3 แกนหลักใช้ในการพิจารณารูปทรงของพื้นที่การทำงาน แกนที่เหลือจึงใช้ในการพิจารณาการหมุนของ tool จะใช้ระบบ Yaw –Pitch-roll (YPR) ดังรูปที่ 2.10

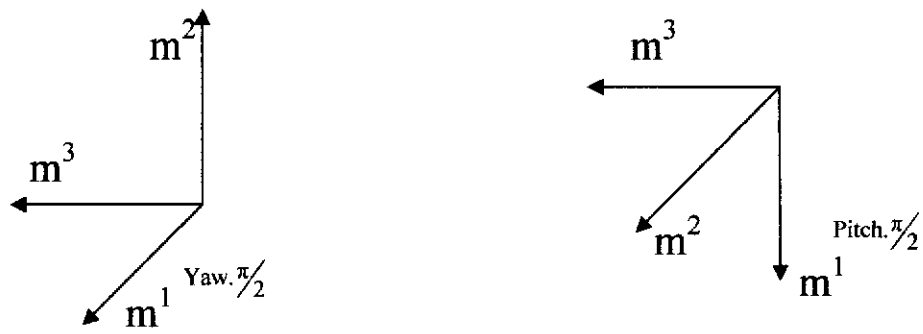


รูปที่ 2.10 yaw-pitch-roll ของ tool

ในการระบุนการหมุน จะติดโครงพิกัด tool $M = \{m^1, m^2, m^3\}$ ไว้ที่ tool และโครงนี้จะเคลื่อนที่ไปพร้อมกับ tool ดังแสดงในรูปแกน m^3 ขนานอยู่ในแนวเดียวกับแกนของ tool และชี้ออกจากข้อมือ แกน m^1 ขนานกับทิศทางที่ปลายนิ้วของ tool เลื่อนปิด-เปิด ส่วนแกน m^2 เป็นไปตามกฎมือขวาของโครงพิกัด M

การเคลื่อนที่แบบ Yaw-Pitch-Row จะกระทำตามลำดับกับแกนที่อยู่หนึ่ง เมื่อเริ่มต้นโครงพิกัดเคลื่อนที่ของ tool M จะทับอยู่กับโครงพิกัดอยู่นิ่งของข้อมือ $F = \{f^1, f^2, f^3\}$ ซึ่งติดอยู่ที่ส่วนปลายของปลายแขน (Forearm) Yaw คือการหมุน tool รอบแกนข้อมือ f^1 Pitch คือการหมุน tool รอบแกนข้อมือ f^2 Roll คือการหมุน tool รอบแกนข้อมือ f^3 ในแต่ละกรณีมุมบวกเกิดจากการหมุนทวนเข็มนาฬิกา โดยมองจากปลายแขนเข้าหาจุดกำเนิด

ลำดับของการหมุน Yaw, Pitch, Row มีความสำคัญมาก เพราะมีผลต่อลักษณะการวางตัวสุดท้ายของ tool ตัวอย่างเช่น หมุน Yaw ไป 90° ตามด้วย Pitch 90° จะมีลักษณะการวางตัวสุดท้ายต่างกับการหมุน Pitch 90° ไปแล้วตามด้วย Yaw 90° เป็นต้น



หมุนYaw ตามด้วย Pitch

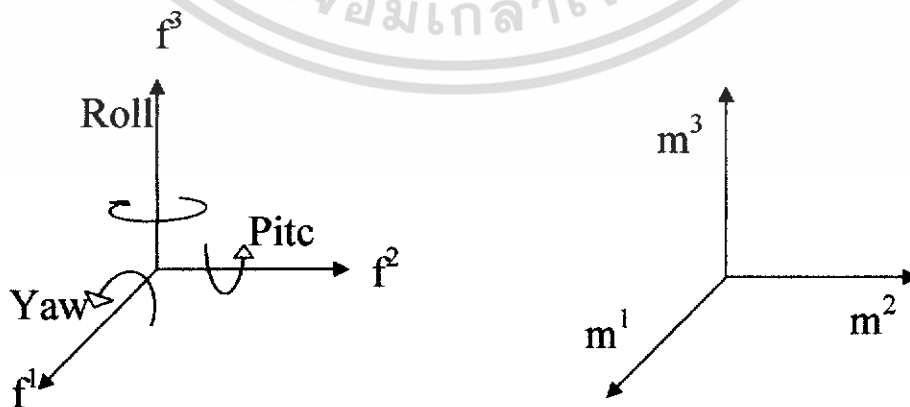


หมุน Pitch ตามด้วย Yaw

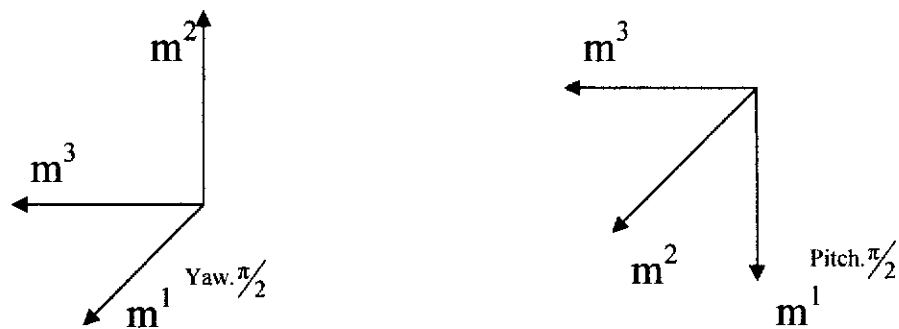
รูปที่ 2.11 ลำดับการหมุนแกน

อีกวิธีหนึ่งที่ใช้ระบุการหมุนคือ ระบบลำดับการหมุนซ้อนลำดับกับ YPR และคิดเทียบกับ
 โครงพิกัดเคลื่อนที่ของtool M แทนที่โครงพิกัดอยู่นิ่งของข้อมือ F ดังรูปที่2.10 โดยเริ่มแรก Roll คือ
 การหมุนรอบแกน m^3 Pitch คือการหมุนรอบแกน m^2 Yaw คือการหมุนรอบแกน m^1 การคิดลักษณะ
 นี้เรียกว่าเป็นระบบ RPY จะให้ผลลัพธ์เหมือนกับแบบ

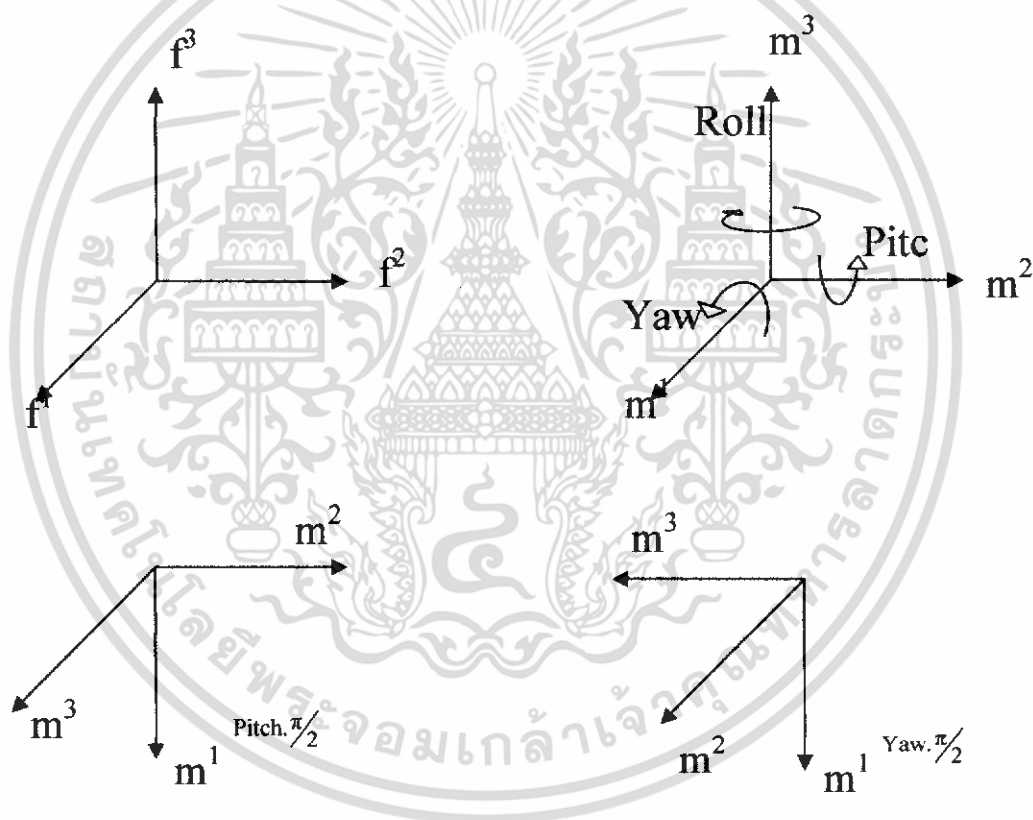
YPR



รูปที่ 2.12-ก ระบบ YPR



รูปที่ 2.12-ข ระบบ YPR



รูปที่ 2.13 ระบบ RPY

ความแม่นยำของการเคลื่อนที่ ความสำคัญประการหนึ่งของการสร้างหุ่นยนต์คือความแม่นยำในการทำงาน ซึ่งความแม่นยำนี้ขึ้นอยู่กับตัวแปร 3 อย่าง

1. Spatial Resolution คือช่วงการเคลื่อนที่ที่มีระยะทางสั้นที่สุดที่หุ่นยนต์แต่ละตัวสามารถที่จะทำได้ ซึ่ง Spatial Resolution นี้ขึ้นอยู่กับองค์ประกอบสำคัญ 2 ประการคือ

- ระบบการควบคุม (Control System) ระบบการควบคุมนี้จะรวมถึงการวัดสัญญาณป้อนกลับของหุ่นยนต์ด้วย

- ความคลาดเคลื่อนทางกล (Mechanical inaccuracy) ของหุ่นยนต์แต่ละตัวขึ้นอยู่กับลักษณะของข้อหมุน(joint)และข้อต่อ(link)และระบบต้นกำลังของหุ่นยนต์ตัวนั้นด้วย

2. Accuracy คือตัวที่แสดงถึงความสามารถของหุ่นยนต์ในการเคลื่อนที่เข้าใกล้จุดเป้าหมายตามที่เราสั่ง Accuracy สามารถที่จะกำหนดให้อยู่ในเทอมของ Spatial Resolution ได้ทั้งนี้เพราะว่าการเคลื่อนที่ให้เข้าใกล้จุดเป้าหมาย ก็ต้องขึ้นกับช่วงของการเคลื่อนที่ที่มีความละเอียดมากน้อยเพียงใด ในการทำงานเราต้องวางจุดที่เราต้องการให้หุ่นยนต์ทำงานอยู่ระหว่างกลางของตำแหน่งของการเคลื่อนที่ของหุ่นยนต์ ทั้งนี้เพราะว่าความคลาดเคลื่อนทางกล มีผลต่อความแม่นยำของหุ่นยนต์ ความแม่นยำของหุ่นยนต์กำหนดให้เท่ากับครึ่งหนึ่งของระยะทางการเคลื่อนที่ที่สั้นที่สุดของหุ่นยนต์ที่สามารถทำได้ โดยขึ้นอยู่กับองค์ประกอบต่อไปนี้

-พื้นที่การทำงานของหุ่นยนต์ ถ้าแขนทำงานในพื้นที่การทำงานจะมีความแม่นยำ มากกว่าเมื่อแขนออกนอกพื้นที่การทำงาน

-วงรอบการทำงาน ถ้าวงรอบการทำงานเป็นวงรอบที่แน่นอนความแม่นยำจะมาก

-น้ำหนักที่ได้รับ ถ้าหุ่นยนต์ทำงานโดยการรับน้ำหนักมากๆความแม่นยำจะลดลง

2.4 หลักการและทฤษฎีของมอเตอร์ดีซี

มอเตอร์ คือ อุปกรณ์ซึ่งทำหน้าที่เปลี่ยนพลังงานไฟฟ้าเป็นพลังงานกล และมอเตอร์ซึ่งขับเคลื่อนด้วยกระแสไฟฟ้าคีซีนี้ เรียกว่า มอเตอร์(DC Motor) และที่ขับเคลื่อนด้วยกระแสไฟเอซี เรียกว่า มอเตอร์เอซี (AC Motor) นอกจากนั้นยังมีมอเตอร์ขนาดเล็กซึ่งอาจขับเคลื่อนได้ทั้งกระแสไฟคีซีหรือเอซี ทิศทางการหมุนของมอเตอร์คีซีจะตรงข้ามกับกรณีเครื่องกำเนิดไฟฟ้าคีซีโดยสิ้นเชิง แต่สำหรับโครงสร้างแล้วจะเหมือนกันทุกประการ จึงสามารถนำเครื่องกำเนิดไฟฟ้าคีซีมาใช้ทำหน้าที่ของมอเตอร์คีซีได้

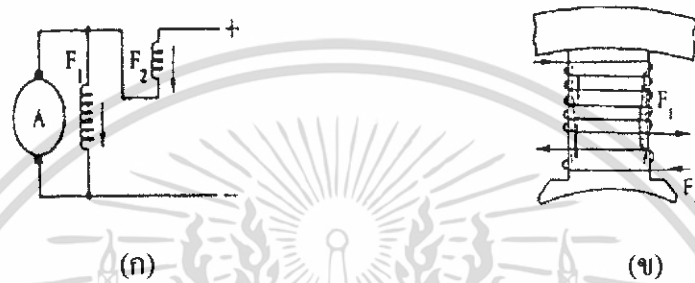
2.4.1 ชนิดของมอเตอร์คีซีแบ่งตามลักษณะการกระตุ้น

การแบ่งชนิดของมอเตอร์คีซีตามลักษณะการกระตุ้นจะเหมือนกับกรณีของเครื่องกำเนิดไฟฟ้าคีซี โดยแบ่งออกเป็น 2 ชนิดดังนี้

1. มอเตอร์คีซีแบบกระตุ้นแยก
2. มอเตอร์คีซีแบบกระตุ้นตัวเอง
 - มอเตอร์คีซีแบบชัณฑ์
 - มอเตอร์คีซีแบบซีรึส
 - มอเตอร์คีซีแบบคอมปาวัค

ฉาหนักหอสุมดกลาง พระจอมเกล้าลาดกระบัง

มอเตอร์ดีซีแบบกระตุ้นแยกนั้น กระแสที่ป้อนให้ชุดขดลวดสนามและขดลวดอาร์มจะมา จากแหล่งจ่ายไฟดีซีซึ่งแยกชุดกัน ส่วนแบบกระตุ้นตัวเองนั้นจะมาจากแหล่งจ่ายไฟดีซีชุดเดียวกัน กรณีที่ต้องการปรับความเร็วรอบของมอเตอร์ดีซี โดยการปรับระดับแรงดันที่ป้อนให้กับมอเตอร์ จะใช้มอเตอร์ดีซีแบบกระตุ้นแยกเป็นกรณีพิเศษเท่านั้น แต่โดยทั่วไปจะใช้แบบกระตุ้นตัวเองเป็น ส่วนมาก สำหรับลักษณะการต่อวงจรของมอเตอร์ดีซีแบบซีรี แบบชัณฑ์ และ แบบคอมปาวด์นั้น จะ เหมือนกับกรณีของเครื่องกำเนิดไฟฟ้าดีซี



รูปที่ 2.14 มอเตอร์คอมปาวด์แบบคิวมูเลทีฟ

อนึ่งสำหรับการต่อวงจรภายในเครื่องกำเนิดไฟฟ้าดีซีแบบคอมปาวด์นั้นจะมี 2 ลักษณะ ด้วยกันคือ ต่อในลักษณะที่ให้แอมแปร์เทอรจากขดลวดสนามแบบชัณฑ์เสริมหรือหักล้างกับขดลวด สนามแบบซีรี มอเตอร์ซึ่งต่อในลักษณะแรกนี้เรียกว่า มอเตอร์คอมปาวด์แบบคิวมูเลทีฟ และแบบ หลังเรียกว่าแบบคิฟเฟอเรนเชียล ซึ่งแสดงได้ดังรูปที่ 2.14 และรูปที่ 2.15 ตามลำดับ



รูปที่ 2.15 มอเตอร์คอมปาวด์แบบคิฟเฟอเรนเชียล

รูปที่ 2.14 ก และ ข แสดงทิศทางของ MMF จากขดลวดสนามแบบชัณฑ์และขดลวดสนาม แบบซีรีของมอเตอร์คอมปาวด์แบบคิวมูเลทีฟ ส่วนรูปที่ 2.15 ก และ ข แสดงทิศทางของกรณี คิฟเฟอเรนเชียล

2.4.2 แรงเคลื่อนไฟฟ้ากลับ

เมื่อมอเตอร์ดีซีหมุน ลวดตัวนำแต่ละเส้นซึ่งมีกระแสไหลผ่านจะตัดกับเส้นแรงแม่เหล็ก จากขั้วแม่เหล็กใหญ่ ทำให้เกิดแรงเคลื่อนไฟฟ้าขึ้นมาจำนวนหนึ่งในทิศทางของกระแสหมุนวนให้กับมอเตอร์ ลองพิจารณาทิศทางของแรงเคลื่อนไฟฟ้าด้านกลับดังรูป 2.16



รูปที่ 2.16 ทิศทางของแรงเคลื่อนไฟฟ้ากลับ

สมมุติให้เส้นแรงแม่เหล็กจากขั้วแม่เหล็กใหญ่ และการหมุนของขดลวดมีทิศทางดังแสดงในรูปที่ 2.16 ก ขณะที่แรงเคลื่อนไฟฟ้าที่ผลิตขึ้นในลวดตัวนำ C_1 และ C_2 จะมีทิศทางพุ่งเข้า (x) และพุ่งออก (O) ตามลำดับ แต่การที่ลวดตัวนำจะยังคงหมุนในทิศทางตามเข็มนาฬิกาได้นั้น จากกฎมือซ้าย กระแสที่ไหลในลวดตัวนำ C_1 และ C_2 จะต้องมีทิศทางดังแสดงในรูปที่ 2.16 ข เท่านั้น นั่นคือมีทิศทางของกระแสที่ตรงข้ามกับแรงเคลื่อนไฟฟ้าในรูปที่ 2.16 ก แรงเคลื่อนไฟฟ้าที่ผลิตขึ้นในรูปที่ 2.16 ก สำหรับกรณีมอเตอร์เท่านั้นที่เรียกว่าแรงเคลื่อนไฟฟ้าด้านกลับ เกิดจากการที่ลวดตัวนำหมุนตัดผ่านเส้นแรงแม่เหล็กจากขั้วแม่เหล็กใหญ่ ในทำนองเดียวกับกรณีของเครื่องกำเนิดไฟฟ้าดีซี จึงมีสมการเหมือนกับกรณีหาค่าแรงเคลื่อนไฟฟ้าที่ผลิตขึ้นได้ในเครื่องกำเนิดไฟฟ้า ดังนี้

$$E_b = p\phi \times \frac{n}{60} \times \frac{Z_c}{a} \quad (2.1)$$

จากสมการที่ 2.1 เนื่องจากแรงเคลื่อนไฟฟ้ากลับแปรผันโดยตรงกับจำนวนรอบหมุนของมอเตอร์ ดังนั้นในกรณีที่มอเตอร์ยังไม่เริ่มหมุน ($n=0$) แรงเคลื่อนไฟฟ้าด้านกลับจะมีค่าเท่ากับศูนย์ด้วย กำหนดมอเตอร์ดีซีขนาด 4 kW 200 V ความต้านทานภายในของอาเมเจอร์ 0.5Ω ถ้าป้อนไฟดีซีจากแหล่งจ่ายขนาด 200 V ให้กับมอเตอร์ในสภาพที่หยุดหมุน จะเห็นว่ากระแสจะไหลในขดอาเมเจอร์ทันทีถึง 400 A ($200/0.5$) แต่เนื่องจากกระแสที่ไหลในขดอาเมเจอร์ แม้ที่ตำแหน่งโหลดเต็มที่จะมีขนาดน้อยกว่านี้มากประมาณ 25 A เป็นอย่างสูง การที่กระแสซึ่งไหลในขดอาเมเจอร์

ขณะที่มอเตอร์หมุนด้วยขนาดของกระแสเพียง 25 A ได้นั้น เนื่องจากแรงเคลื่อนไฟฟ้ากลับทิศทางที่เกิดขึ้นนี้พยายามต้านการจ่ายกระแสจากแหล่งจ่ายไฟดีซี

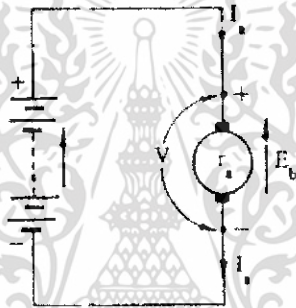
2.4.3 ความสัมพันธ์ระหว่างแรงเคลื่อนไฟฟ้าต้านกลับและกระแส เมื่อกำหนดให้

V คือ แรงดันจากแหล่งจ่ายไฟดีซี ระหว่างคู่แปรงถ่าน

E_b คือ แรงเคลื่อนไฟฟ้าต้านกลับ

R_a คือ ความต้านทานภายในทั้งหมดในวงจรของอาเมเจอร์

I_a คือ กระแสอาเมเจอร์



รูปที่ 2.17 ความสัมพันธ์ระหว่างกระแสและแรงเคลื่อนไฟฟ้ากลับในวงจรอาเมเจอร์

จากรูปที่ 2.17 จะได้ว่า

$$I_a \dots = \dots \frac{V - E_b}{r_a} \quad (2.2)$$

หรือ

$$V = E_b + I_a r_a$$

$$E_b \dots = \dots V - I_a r_a \quad (2.3)$$

กรณีที่มอเตอร์ทำงานที่โหลดเต็มที่ แรงดันตกคร่อม $I_a r_a$ จะมีค่าประมาณ 10 เปอร์เซ็นต์ของ V ขณะที่มอเตอร์หยุดหมุน เนื่องจาก $E_b = 0$ จะได้

$$I_s \dots = \dots \frac{V}{r_a} \quad (2.4)$$

โดยที่ I_s คือ กระแส I_a ตอนเริ่มเดินเครื่องมีหน่วยเป็นแอมแปร์ (A)

จากสมการที่ (2.2) และ (2.3) สามารถอธิบายได้ว่า ขณะที่มอเตอร์ยังไม่เริ่มหมุน เมื่อป้อนไฟดีซีจะมีกระแส $I_s = \frac{V}{r_a}$ จำนวนมากไหลในขดอามเจอร์ มอเตอร์จะหยุดหมุนด้วยความเร็วรอบสูงในทันที ทำให้ E_b สูงขึ้นอย่างรวดเร็ว กระแสในอามเจอร์จะมีค่าลดลงเรื่อยๆจนมีค่าคงที่ในที่สุด

2.4.4 สมการเออร์พุดของอามเจอร์

จากสมการที่ (2.21) ได้ว่า

$$V = E_b + I_a r_a \quad (V)$$

หรือ $E_b = V - I_a r_a \quad (V)$

เมื่อคูณ I_a ตลอดจะได้

$$V I_a = E_b I_a + I_a^2 r_a \quad (W)$$

หรือ $E_b I_a = V I_a - I_a^2 r_a \quad (W)$

โดยที่ $V I_a$ คือ กำลังไฟฟ้าทางด้านอินพุตที่ป้อนให้กับอามเจอร์

$I_a^2 r_a$ คือ การสูญเสียจากลวดตัวนำในอามเจอร์

เมื่อหัก $I_a^2 r_a$ ออกจาก $V I_a$ จะเหลือ $E_b I_a$ ซึ่งเป็นกำลังไฟฟ้าส่วนที่เปลี่ยนไปเป็นกำลังกลภายในอามเจอร์ กำหนดให้เป็น P_{ar} ดังนั้นจะได้

$$P_{ar} \dots = \dots E_b I_a \quad (2.5)$$

กำลังกล P_{ar} บางส่วนจะสูญเสียไปในรูปของการสูญเสียทางกลและการสูญเสียในแกนเหล็ก จึงไม่สามารถปรากฏออกมาให้เห็นได้ทั้งหมด จากสมการที่ (2.1)

$$E_b = \frac{P}{a} \times \frac{n}{60} \times Z_c \times \Phi \quad (2.6)$$

เนื่องจากมอเตอร์ซึ่งสร้างสำเร็จรูป นอกจาก n และ ϕ แล้ว ค่าอื่นๆ คือ p/a , Z_c มีค่าคงที่หมด จึงสามารถเขียนสมการใหม่ได้ดังนี้

$$E_b \dots = \dots K_n \Phi$$

โดยที่

$$K = \frac{p}{a} \times \frac{Z_c}{60}$$

เมื่อคูณ I_a ตลอดจะได้

$$E_b I_a \dots = \dots K_n \Phi n_a \dots = \dots P_{ar} \quad (2.7)$$

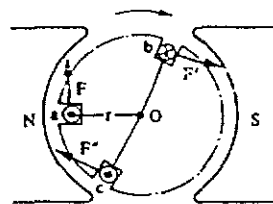
2.4.5 ทอร์ก (Torque)

รูปที่ 2.11 แสดงแรงที่กระทำบนลวดตัวนำ (a) ซึ่งห่างจากจุดศูนย์กลาง (O) เป็นระยะ r (หน่วยเป็นเมตร) ในทิศทางสัมผัสกับเส้นรอบวงของแกนเหล็กอามเจอร์ ที่ตำแหน่งนี้กำหนดให้มีค่าเป็น F นิวตัน ดังนั้นแรงที่กระทำบนลวดตัวนำนี้จะมีโมเมนต์เท่ากับ $F \times r$ ซึ่งจะพยายามขับเคลื่อนแกนเหล็กอามเจอร์ให้หมุนเคลื่อนที่ไป แต่เนื่องจากในแกนเหล็กอามเจอร์มีสล๊อตเป็นจำนวนมาก แต่ละสล๊อตก็มีลวดตัวนำจำนวนมากเช่นกัน ดังนั้นแรงที่กระทำบนลวดตัวนำทั้งหมด จะมีโมเมนต์ซึ่งพยายามขับเคลื่อนแกนเหล็กให้หมุนไปในทิศทางเดียวกัน เมื่อรวม โมเมนต์ย่อยทั้งหมดนี้เข้าด้วยกัน จะได้ผลรวมของ โมเมนต์ ดังนี้

$$T_a = (F_r + F'' + \dots)$$

โดยที่ T_a คือ ทอร์กที่เกิดขึ้นมีหน่วยเป็น m-N

ดังนั้น ทอร์ก คือ ผลรวมของ โมเมนต์ของแรงที่กระทำต่อลวดตัวนำรอบจุดศูนย์กลาง



รูปที่ 2.18 โมเมนต์ซึ่งลวดตัวนำอามเจอร์กระทำรอบจุดศูนย์กลาง

สมการของทอร์ก (Torque Equation)

จากรูปที่ 2.18 กำลังกล P_a ของลวดตัวนำ ที่เกิดขึ้น จะเท่ากับผลคูณแรงระหว่าง (F) มีหน่วยเป็นนิวตันกับระยะทาง ($2\pi r$)($n/60$) มีหน่วยเป็นเมตร ซึ่งเป็นระยะทางที่ลวดตัวนำ (a) เคลื่อนที่ไปในแนวแรงในช่วงเวลา 1 วินาที ดังนั้นจะได้

$$P_a = 2\pi \times \frac{n}{60} \times F$$

$$= 2\pi \times \frac{n}{60} \times F \quad (W)$$

แต่เนื่องจาก $2\pi n/60 = \omega$ ซึ่งเป็นความเร็วเชิงมุม (rad/sec) และ Fr คือ โมเมนต์ของแรง (F) รอบจุดศูนย์กลาง (O) จะได้

$$P_a = \omega Fr \quad (W)$$

ดังนั้นผลรวมของกำลังกลที่เกิดขึ้น P_{ar} ของลวดตัวนำทั้งหมดจะมีค่าดังสมการ คือ

$$\begin{aligned} P_{ar} &= P_a + P_b + P_c + \dots \\ &= (\omega \times Fr) + (\omega \times F'r) + (\omega \times F''r) + \dots \\ &= \omega(Fr + F'r + F''r + \dots) \end{aligned}$$

ดังนั้น

$$P_{ar} = \omega T_a \quad (W)$$

และจากสมการที่(2.7) ได้ว่า

$$Kn\phi_a = P_{ar}$$

ดังนั้น

$$Kn\phi_a = \omega T_a$$

เมื่อแทนค่า K และ $\omega \left(K = \frac{P}{a} \times \frac{Z_c}{60}, \omega = 2\pi \frac{n}{60} \right)$

$$n \frac{P}{a} \times \frac{Z_c}{60} \times \phi I_a = 2\pi \left(\frac{n}{60} \right) I_a$$

$$T_a = \frac{1}{2\pi} \times \frac{P}{a} \times Z_c \times \phi \times I_a \quad (m-n)$$

$$= K' \phi I_a = (m-n)$$

$$K' = \frac{1}{2\pi} \times \frac{P}{a} Z_c$$

นั่นคือ ทอร์กของดีซีมอเตอร์จะแปรผันโดยตรงผลคูณระหว่างจำนวนเส้นแรงแม่เหล็กกับกระแสอาเมเจอร์ ไม่เพียงแต่มอเตอร์ดีซีเท่านั้น ความสัมพันธ์นี้ยังสามารถใช้ได้กับมอเตอร์เอซีด้วย

2.4.6 สมการของความเร็รรอบ

จากสมการที่(2.3) และ (2.7) จะได้

$$E_b = V - I_a r_a$$

$$E_b = K \phi n$$

$$n = \frac{E_b}{K \phi} = \frac{V - I_a r_a}{K \phi}$$

$$n \approx \frac{V}{K \phi} \quad (\text{rpm})$$

$I_a r_a$ ในสมการนี้ประมาณว่ามีค่าน้อยมาก ซึ่งควรใช้เฉพาะกรณีที่มี I_a หรือกระแสไหลต่ำไม่มากเท่านั้น จะเห็นว่าความเร็รรอบแปรผันโดยตรงกับแรงเคลื่อนไฟฟ้า และแปรผกผันกับจำนวนเส้นแรงแม่เหล็ก

2.4.7 ความสามารถในการปรับกระแสได้เองตามขนาดของโหลด

เมื่อป้อนแรงดัน V คงที่ให้กับมอเตอร์แบบซีรี่ และให้ทำงานที่โหลดต่างๆ มอเตอร์จะหมุนที่ความเร็รรอบ (n) ค่าหนึ่ง และจะดึงกระแสจากแหล่งจ่ายไฟดีซีด้วยค่า I_a ค่าหนึ่ง เนื่องจากกำลังที่ด้านเอาต์พุตของมอเตอร์ในขณะนี้มีน้อย $V I_a$ จึงมีค่าน้อยด้วย นั่นคือกระแส I_a ที่แหล่งจ่ายไฟดี

ซึ่งจ่ายออกจะต้องมีค่าต่ำด้วย เมื่อให้มอเตอร์ทำงานที่โหลดมากๆ V_a จะต้องมีค่ามากขึ้น สำหรับกรณีที่แรงดันของแหล่งจ่ายมีค่าคงที่ I_a จำเป็นต้องมีค่ามากขึ้นด้วย นั่นคือเมื่อโหลดน้อยมอเตอร์จะดึงกระแสจากแหล่งจ่ายไฟดีซีน้อย และเมื่อโหลดมากจะดึงกระแสมากขึ้น โดยอัตโนมัติ

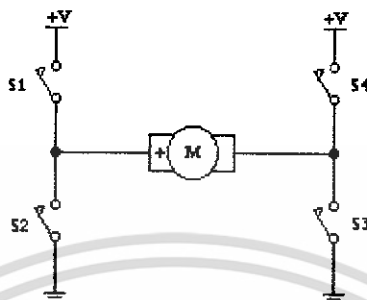
2.4.8 ความสามารถในการเร่งหรือหน่วงความเร็วรอบ

ลองพิจารณาความสัมพันธ์ระหว่างโหลดทอร์คซึ่งเป็นทอร์คที่โหลดต้องการ กับมอเตอร์ทอร์คกับโหลดซึ่งเป็นทอร์คที่มอเตอร์จ่ายให้กับโหลด เพื่อให้เข้าใจได้ง่ายขึ้นจะยกตัวอย่างมอเตอร์ดีซีแบบขั้วเดียวในการอธิบาย กานหมุนของมอเตอร์ในขณะที่มีโหลดทางกลนั้น จะเท่ากับการใส่แรงเบรกกระทำต่อแกนของอาเมเจอร์ เมื่อโหลดทอร์คสูงขึ้น ก็เท่ากับแรงเบรกที่กระทำต่อแกนอาเมเจอร์มีค่ามากขึ้น นั่นคือขณะที่มอเตอร์กำลังหมุนภายใต้โหลดค่าหนึ่งด้วยเสถียรภาพที่ดี ถ้าให้โหลดทอร์คมีค่าสูงขึ้น หรือแรงเบรกที่กระทำมีค่ามากขึ้น จะทำให้เกิดความหน่วงขึ้น ความเร็วรอบ n จะมีค่าลดลง สำหรับกรณีของมอเตอร์แบบขั้วเดียว ทรานส์เฟอเรนซ์ระหว่างขั้วมีค่าคงที่ อาจถือได้ว่า ϕ มีค่าไม่เปลี่ยนแปลง (แต่ในทางปฏิบัติเนื่องจากอาเมเจอร์รีแอกชัน ทำให้ ϕ มีค่าเปลี่ยนแปลงไปบ้าง) ดังนั้นถ้าความเร็วรอบลดลงแรงเคลื่อนไฟฟ้ากับ E_b จะมีค่าต่ำลง แต่เนื่องจาก $I_a = (V - E_b) / r_a$ และทอร์ค $(T) = K' \phi I_a$ มอเตอร์ทอร์คจะมีค่าเพิ่มขึ้น ดังนั้นความเร็วรอบจะลดลงเรื่อยๆ จนกระทั่งทำให้ทอร์ค T เท่ากับโหลดทอร์คพอดี จากนั้นมอเตอร์จะเริ่มทำงานที่ตำแหน่งใหม่นี้

ในทางตรงกันข้ามถ้าโหลดทอร์คมีค่าน้อยกว่ามอเตอร์ทอร์ค แรงที่หมุนแกนอาเมเจอร์จะมากกว่าแรงเบรก ทำให้เกิดความเร่งขึ้นและความเร็วรอบจะเพิ่มขึ้น เมื่อความเร็วรอบเพิ่มขึ้นแรงเคลื่อนไฟฟ้ากลับจะมีค่าเพิ่มขึ้น กระแส $I_a = (V - E_b) / r_a$ จะมีค่าน้อยลง ทอร์ค $(T) = K' \phi I_a$ จะมีค่าลดลง ความเร่งจะน้อยลงจนมีค่าเป็นศูนย์ เมื่อทอร์คจากมอเตอร์มีค่าเท่ากับโหลดทอร์คพอดี จากนั้นมอเตอร์จะเริ่มทำงานที่ตำแหน่งใหม่นี้ ในการเริ่มเดินมอเตอร์จากสภาพที่หยุดนิ่งจนกระทั่งหมุนนั้น แรงเคลื่อนไฟฟ้ากลับ E_b จะค่อยๆ เพิ่มขึ้นเรื่อยๆ จาก $E_b = 0$ ในช่วงแรกสุดนั้นเนื่องจากมีกระแสสูงมาก ทอร์คจากมอเตอร์จะมีค่าสูงกว่าโหลดทอร์คมาก ทำให้เกิดความเร่งสูงมาก ความเร็วรอบจะสูงขึ้นเรื่อยๆ จนกระทั่งโหลดทอร์คมีค่าเท่ากับมอเตอร์ทอร์คและความเร็วรอบจะคงที่ โหลดทอร์คจะเปลี่ยนแปลงตามความเร็วรอบ หรือมีค่าคงที่เมื่อความเร็วรอบเปลี่ยนไปหรือไม่ขึ้นอยู่กับคุณสมบัติของโหลด ตัวอย่างเช่น พัดลมมันยังหมุนที่ความเร็วรอบสูงขึ้น โหลดทอร์คหรือทอร์คที่โหลดต้องการจะยังมีค่าสูงขึ้น ในขณะที่เครื่องกลึงไม่ว่าความเร็วรอบจะเปลี่ยนไปอย่างไรก็ตาม โหลดจะต้องทอร์คเกือบคงที่โดยตลอด

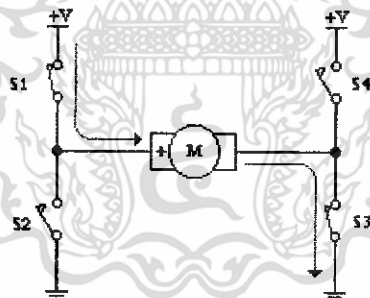
2.5 หลักการทำงานของวงจร H-Bridge Switching

เรื่องของการควบคุมทิศทางการหมุนของมอเตอร์ดีซี มีความจำเป็นอย่างแน่นอการนำหลักการทำงานของวงจรสวิตช์แบบ H-Bridge มาใช้



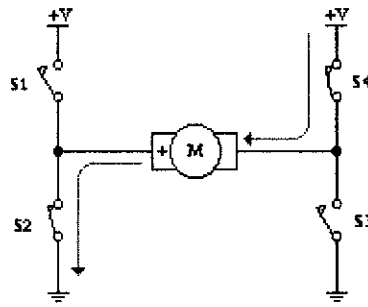
รูปที่ 2.19 วงจรสวิตช์แบบ H-Bridge

เริ่มจากการนำหลักการของวงจรมานั้น จะประกอบไปด้วย สวิตช์ 4 ตัว นั่นก็คือ S1, S2, S3 และ S4 นั่นเอง ซึ่งในรูปแบบตัวอย่างจะใช้มอเตอร์ดีซีเป็นภาระของวงจรมันเองในสภาวะเริ่มต้น สวิตช์ทุกตัว Off อยู่ ก็จะไม่มีการเกิดขึ้นทั้งสิ้น เพราะไม่มีกระแสไฟฟ้าไหลเข้าสู่มอเตอร์ (รูปที่ 2.19)



รูปที่ 2.20 วงจรสวิตช์แบบ H-Bridge เมื่อ On สวิตช์ S1 และ S3

และเมื่อเราทำการ On สวิตช์ S1 และ S3 พร้อมกัน (รูปที่ 2.20) จะเป็นการเชื่อมวงจร ทำให้มีกระแสไฟฟ้า ไหลผ่านมอเตอร์จากขั้วบวกของมอเตอร์ไปยังขั้วลบของมอเตอร์ จึงทำให้มอเตอร์สามารถหมุนได้ ในทิศทาง Forward (จะหมุนแบบตามเข็มนาฬิกาหรือทวนเข็มนาฬิกานั้น ขึ้นอยู่กับลักษณะของการพันขดลวดภายในมอเตอร์)



รูปที่ 2.21 วงจรสวิตช์แบบ H-Bridge เมื่อ On สวิตช์ S2 และ S4

และในทางกลับกัน ถ้าหากเราทำการ On สวิตช์ S2 และ S4 พร้อมกัน (รูปที่ 2.21) ก็จะเป็นการเชื่อมวงจร และทำให้เกิดกระแสไหลผ่านมอเตอร์จากขั้วลบของมอเตอร์ไปยังขั้วบวกของมอเตอร์ จึงทำให้มอเตอร์สามารถหมุนได้ และเป็นการหมุนในทิศทาง Reverse (กลับทิศทางกับในกรณีแรก)

สรุปว่า วงจรนี้อาศัยสวิตช์ 4 ตัว เพื่อบังคับทิศทางการไหลของกระแสไฟฟ้าที่ไหลผ่านมอเตอร์ เพื่อควบคุมให้มอเตอร์หมุนตามทิศทางที่เราต้องการ โดยการผลัดกัน On และ Off สวิตช์พร้อมกัน 2 ตัว นั่นเอง

2.6 ตัวควบคุม (Controller)

ตัวควบคุม เป็นส่วนประกอบหนึ่งในระบบควบคุมแบบป้อนกลับ ซึ่งมีสัญญาณความผิดพลาดเป็นอินพุตและมีเอาต์พุตเป็นอินพุตของ corrective element ส่วนนี้จะกล่าวถึงวิธีการเลือกใช้ตัวควบคุมที่เหมาะสมกับระบบควบคุมแบบป้อนกลับและวิธีการหาพารามิเตอร์ที่เหมาะสมสำหรับตัวควบคุม โดยที่ ความสัมพันธ์ระหว่างอินพุตและเอาต์พุตของตัวควบคุม นิยมเรียก กฎการควบคุม (control Law) โดยจะมี 3 รูปแบบ คือ

1. Proportional Control
2. Integral Control
3. Derivative Control

ในระบบบางประเภทมีความจำเป็นที่ต้องปรับปรุงสมรรถนะของระบบควบคุม ซึ่งสามารถทำได้โดยการใช้ส่วนประกอบอื่นเพิ่มเติมเข้ากับระบบควบคุม การที่ปรับเปลี่ยนสมรรถนะของระบบควบคุมป้อนกลับแบบนี้เรียกว่า การชดเชย (compensation)

2.6.1 ระบบควบคุมแบบพี (P: Proportional Control)

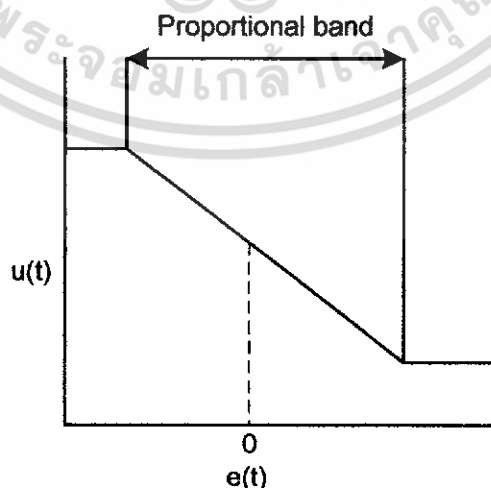
ในระบบควบคุมแบบพี เอาท์พุทของตัวควบคุม $u(t)$ จะเป็นสัดส่วนกับอินพุทของตัวควบคุม และถ้ากำหนดสัญญาณอินพุทให้กับตัวควบคุมเป็น ค่าความผิดพลาด $e(t)$ ซึ่งเป็นฟังก์ชันของเวลา จะได้

$$u(t) \dots = \dots K_p e(t) \quad (2.8)$$

เมื่อ K_p เป็นค่าคงที่เรียกว่า อัตราขยายแบบสัดส่วน (proportional gain) จะพบว่าเอาท์พุทที่ออกจากตัวควบคุมแบบพี เป็นสัดส่วนโดยตรงกับขนาดของความผิดพลาดในขณะที่กำลังพิจารณา ทำให้ฟังก์ชันถ่ายโอนของตัวควบคุม $G_c(s)$ มีค่าเป็น

$$G_c(s) \dots = \dots K_p \quad (2.9)$$

ดังนั้นการควบคุมด้วยตัวควบคุมแบบนี้จะเป็นเพียงการขยายสัญญาณความผิดพลาดเท่านั้น การที่ได้สัญญาณความผิดพลาดขนาดใหญ่ที่เวลาหนึ่ง ทำให้เกิดเอาท์พุทที่มีขนาดใหญ่จากตัวควบคุมในเวลานั้น อย่างไรก็ตามการที่ให้ อัตราขยายคงที่นั้นในทางปฏิบัติอาจกำหนดไว้ในบางช่วงของสัญญาณความผิดพลาดเท่านั้น หรืออาจกำหนดให้ตัวควบคุมมีค่าเอาท์พุทไม่น้อยกว่าค่าค่าหนึ่งและไม่มากเกินไปกว่าค่าค่าหนึ่งก็ได้ซึ่งการกำหนดช่วงจำกัดของเอาท์พุทจะมีลักษณะ ดังรูปที่ 2.22 และการกำหนดเอาท์พุทของตัวควบคุมแบบพี ช่วงที่มีการกำหนดสัดส่วนนี้ จะเรียกว่า proportional band.

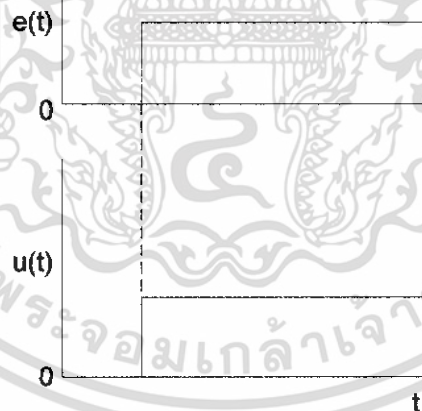


รูปที่ 2.22 การกำหนดช่วงจำกัดของ เอาท์พุท

การกำหนด *proportional band* นี้ จะช่วยให้สัญญาณเอาต์พุตมีค่าจำกัดไม่ไปสู่ค่าอนันต์ ทั้งทางด้านบวกและทางด้านลบ และเมื่อตัวควบคุมมีเอาต์พุตสูงที่สุดที่เป็นไปได้ค่าหนึ่ง ก็นิยมที่จะกำหนดเอาต์พุตค่าใด ๆ เป็นร้อยละของค่าสูงสุดที่เป็นไปได้ ดังนั้นการเปลี่ยนแปลงค่าเอาต์พุตของตัวควบคุม 100% ก็หมายถึงว่าเอาต์พุตจะเปลี่ยนจากค่าต่ำสุดที่เป็นไปได้ไปเป็นค่าสูงสุดที่เป็นไปได้ซึ่งจะได้ว่า

$$K_p \dots = \dots \frac{100}{\text{proportional band}} \quad (2.10)$$

เนื่องจากเอาต์พุตของตัวควบคุมจะเป็นสัดส่วนกับอินพุต ดังนั้น ถ้าหากอินพุตมีลักษณะเป็น step เอาต์พุตที่ได้ก็จะมีลักษณะเป็น step เช่นกัน โดยลักษณะของกราฟแสดงอินพุตและเอาต์พุตจะมีสัดส่วนที่แน่นอนค่าหนึ่ง ตามรูปที่ 2.23 โดยรูปนี้แสดงถึงการตอบสนองของตัวควบคุม เมื่ออินพุตอยู่ในช่วง *proportional band*

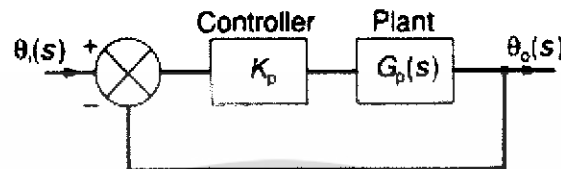


รูปที่ 2.23 ผลการตอบสนองของตัวควบคุมแบบพี

ในทางปฏิบัติตัวควบคุมแบบพี นี้มีลักษณะเหมือนกับเครื่องขยายสัญญาณรูปแบบหนึ่ง ซึ่งอาจจะเป็นในลักษณะของอุปกรณ์ไฟฟ้า หรืออาจจะเป็นเครื่องขยายสัญญาณเชิงกล เช่น คาน ก็ได้ ลักษณะของระบบที่มีตัวควบคุมแบบพี มีลักษณะดังที่แสดงในรูปที่ 2.24 และได้ฟังก์ชันถ่ายโอนระบบเปิดเป็น

$$G_o(s) \dots = \dots K_p G_p(s) \quad (2.11)$$

เมื่อ $G_p(s)$ เป็นฟังก์ชันถ่ายโอนของระบบ



รูปที่ 2.24 ระบบควบคุมแบบพี

ข้อเสียของระบบที่มีตัวควบคุมแบบพี คือไม่มีการเพิ่มเทอม $1/s$ ในส่วน forward path ซึ่งหมายความว่า ถ้าระบบเป็นชนิด (type) 0 ตัวควบคุมจะไม่เปลี่ยนแปลงชนิด ทำให้ระบบเป็นชนิด 0 เหมือนเดิม และทำให้เกิดความผิดพลาดที่สภาพคงตัว เนื่องจากตัวควบคุมไม่ได้ทำการเพิ่มโพล (pole) หรือซีโร (zero) ใหม่ให้กับระบบเพียงแต่เปลี่ยนตำแหน่งของโพลหรือซีโรเท่านั้น เนื่องจากระบบควบคุมแบบป้อนกลับหนึ่งหน่วย มีฟังก์ชันถ่ายโอนของระบบเป็น

$$G(s) \dots = \dots \frac{K_p G_p(s)}{1 + K_p G_p(s)} \quad (2.12)$$

สมการคุณลักษณะเป็น $(1 + K_p G_p(s))$ และมีรากเปลี่ยนไปตามค่าของ K_p

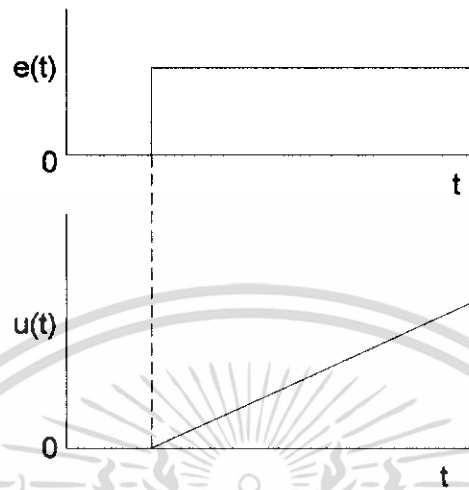
2.6.2 ระบบควบคุมแบบไอ (I: Integral Control)

ในระบบควบคุมแบบไอเอาต์พุทของตัวควบคุมจะเป็นสัดส่วนกับสัญญาณของผิดพลาดเทียบกับเวลา หรือ

$$u(t) \dots = \dots K_i \int_0^t e(t) dt \quad (2.13)$$

เมื่อ K_i เป็นค่าคงที่เรียกว่า integral gain ซึ่งจะมีหน่วยเป็น $1/\text{sec}$ รูปที่ 2.19 แสดงลักษณะการตอบสนองของระบบควบคุมแบบไอ เมื่อได้รับสัญญาณอินพุทแบบ step ค่าระหว่างเวลา t และ 0 หมายถึงพื้นที่ใต้กราฟของสัญญาณอินพุทจากเวลา 0 ถึง t ดังนั้นเนื่องจากเริ่มการมีสัญญาณ

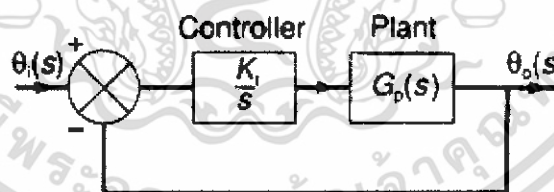
อินพุทแบบ step เอาท์พุทที่ออกจากตัวควบคุมมีค่ามากขึ้นเรื่อย ๆ ด้วยอัตราที่คงที่ ทำให้เอาท์พุทที่เวลาใดๆจะเป็นสัดส่วนกับอินพุทที่เกิดขึ้น



รูปที่ 2.25 ผลการตอบสนองของตัวควบคุมแบบไอ

ได้ฟังก์ชันถ่ายโอนของตัวควบคุมแบบไอ เป็น

$$G_c(s) \dots = \dots \frac{U(s)}{E(s)} = \dots \frac{K_i}{s} \quad (2.14)$$



รูปที่ 2.26 ระบบควบคุมแบบไอ

ดังนั้นสำหรับระบบที่แสดงในรูปที่ 2.26 การควบคุมแบบไอ จะให้ forward-path transfer function เป็น $\frac{K_i}{s} G_p(s)$ และทำให้มีฟังก์ชันถ่ายโอนระบบเปิดเป็น

ดังนั้นสามารถพิจารณาถึงข้อได้เปรียบของการควบคุมแบบไอ ได้จากสมการ 2.14 ซึ่งจะเห็นว่าระบบควบคุมแบบไอ จะเพิ่มจำนวนโพลให้กับระบบควบคุมและเพิ่ม ชนิด ของระบบ จาก ชนิด 0 เป็น ชนิด 1 ซึ่งทำให้ระบบมีความผิดพลาดที่สถานะคงตัวเป็นศูนย์เทียบกับต่อ step input

อย่างไรก็ตามการเพิ่มโพลที่ $s = 0$ และไม่มีการเพิ่มซีโรให้กับระบบควบคุม ทำให้ความแตกต่างระหว่างจำนวนโพล (n) และจำนวนซีโร (m) เพิ่มขึ้นอีก 1 ซึ่งมีผลให้ asymptote angles ของทางเดินรากลดลง และจุดตัดเคลื่อนไปทางครึ่งขวาของ s -plane มากขึ้น มีผลทำให้ความเสถียรสัมพัทธ์ของระบบลดลง

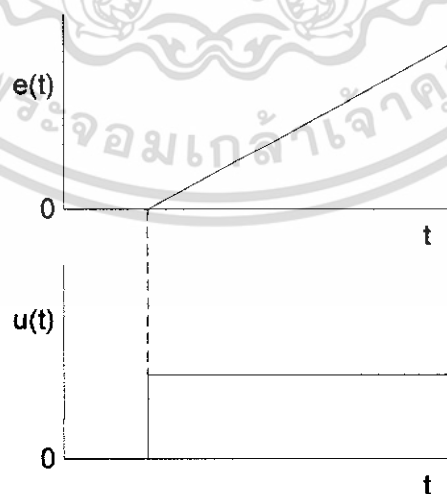
$$\text{Asymptote angle} = \pm \frac{\pi}{n-m}, \frac{3\pi}{n-m}, \dots \quad (2.15)$$

2.6.3 ระบบควบคุมแบบดิ (D: Derivative Control)

ระบบควบคุมแบบดิ นี้เอาที่พู่จะเป็นสัดส่วนกับอัตราการเปลี่ยนแปลงความผิดพลาดเทียบกับเวลานั้นคือ

$$u(t) \dots = \dots K_d \frac{de(t)}{dt} \quad (2.16)$$

เมื่อ K_d คือ derivative gain และมีหน่วยเป็นวินาที



รูปที่ 2.27 ผลตอบสนองของระบบ ควบคุมแบบดิ

รูปที่ 2.27 แสดงสิ่งที่เกิดขึ้นเมื่อสัญญาณอินพุตเป็นสัญญาณแบบ ramp เมื่อได้รับสัญญาณอินพุตและไม่ใช่ค่าของความผิดพลาดซึ่งทำให้ได้สัญญาณส่งออกจากตัวควบคุมมีค่ามาก่อนที่จะเกิดความผิดพลาดขึ้นมากจริงๆ อย่างไรก็ตามหากความผิดพลาดมีค่าคงที่ก็จะไม่มีการสะสมค่าความผิดพลาดแม้ว่าค่าความผิดพลาดจะมีมากก็ตาม ทำให้การควบคุมแบบดี นี้ไม่เปลี่ยนแปลงต่อค่าความผิดพลาดที่คงที่หรือเปลี่ยนแปลงอย่างช้า ๆ ซึ่งผลที่ตามมาของการควบคุมแบบนี้จะไม่ใช้เพียงตัวเดียว แต่มักจะใช้ควบคู่ร่วมกับการควบคุมแบบอื่น ฟังก์ชันถ่ายโอนของตัวควบคุมแบบดีจะเป็น

$$G_c(s) \dots = \dots K_d s \quad (2.17)$$

ดังนั้นสำหรับระบบควบคุมดี จะได้ฟังก์ชันถ่ายโอนเป็น

$$G_o(s) \dots = \dots \frac{K_d s G_p(s)}{1 + K_d s G_p(s)} \quad (2.18)$$



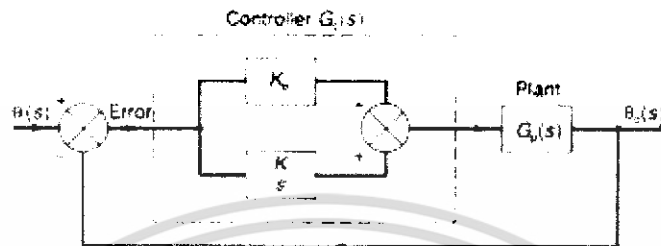
รูปที่ 2.28 บล็อกไดอะแกรม (block diagram) ของตัวควบคุมแบบดี

ถ้าหากว่าระบบเป็นแบบชนิด 1 หรือสูงกว่า การควบคุมแบบดีจะลด S ในเทอมส่วนลงและลดชนิด ของระบบลง 1 อย่างไรก็ตามได้กล่าวก่อนหน้านี้แล้วว่า การควบคุมแบบดี นี้จะไม่ใช้เพียงลำพังแต่จะใช้ร่วมกับการควบคุมแบบอื่น เพราะเมื่อใช้การควบคุมแบบดี ทำให้เพิ่มความเร็วในการตอบสนองของระบบต่อความผิดพลาดที่เกิดขึ้น

ในทางปฏิบัติการนำการควบคุมดีไปใช้นั้นค่อนข้างจะลำบาก ดังนั้นในทางปฏิบัติโดยทั่วไปจะเป็นการประมาณการควบคุมแบบดี โดยใช้ lead compensator ซึ่งจะกล่าวถึงต่อไป

2.6.4 ระบบควบคุมแบบพีไอ (PI: Proportional plus Integral Control)

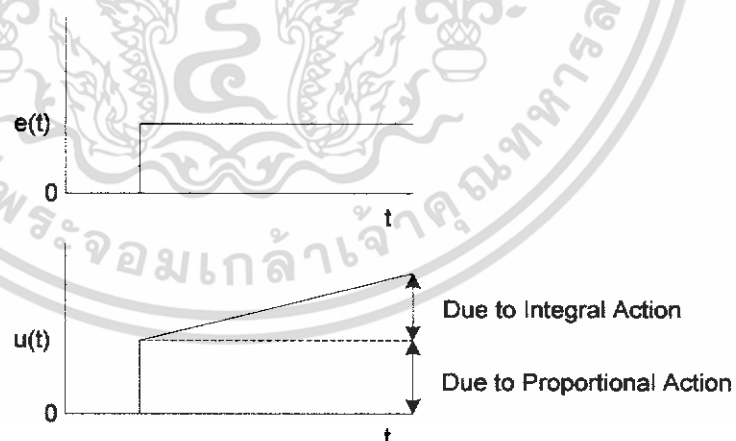
การที่ระบบควบคุมมีความเสถียรสัมพัทธ์ลดลง เมื่อใช้การควบคุมแบบไอ สามารถที่จะแก้ไขได้ในระดับหนึ่ง โดยการใช้การควบคุมแบบพีร่วมกับแบบไอ ซึ่งลักษณะของระบบควบคุมเป็นตามรูปที่ 2.29



รูปที่ 2.29 บล็อกไดอะแกรมของระบบควบคุมแบบพีไอ

สำหรับระบบดังกล่าวจะมี เอาท์พุทของตัวควบคุมเป็น

$$u(t) \dots = \dots K_p e(t) + K_i \int_0^t e(t) dt \quad (2.19)$$



รูปที่ 2.30 การตอบสนองของระบบควบคุมแบบพีไอ

รูปที่ 2.30 แสดงเอาต์พุตของตัวควบคุมที่ได้รับเมื่อมีอินพุตเป็นแบบ step ฟังก์ชันถ่ายโอนของตัวควบคุมแบบพีไอ เป็น

$$\begin{aligned} G_o(s) \dots &= \dots K_p + \frac{K_i}{s} \\ &= \dots \frac{s K_p + K_i}{s} \\ &= \dots K_p \frac{(s + K_i / K_p)}{s} \end{aligned} \quad (2.20)$$

ให้ integral time constant เป็น

$$\tau_i \dots = \dots \frac{K_p}{K_i} \quad (2.21)$$

ดังนั้นจะได้

$$G_c(s) \dots = \dots \frac{K_p [s + (1/\tau_i)]}{s} \quad (2.22)$$

และจะทำให้ได้ฟังก์ชันถ่ายโอนระบบเปิดเป็น

$$\begin{aligned} G_o(s) \dots &= \dots G_c(s) G_p(s) \\ &= \dots \frac{K_p [s + (1/\tau_i)] G_p(s)}{s} \end{aligned} \quad (2.23)$$

Intersection/point = (ผลรวมของโพล - ผลรวมของศูนย์) / (n - m)

จะเห็นว่า มีซีโรที่ $s = -1/\tau_i$ และโพลที่ $s = 0$ เพิ่มให้กับฟังก์ชันถ่ายโอนของระบบเมื่อใช้การควบคุมแบบพีไอ การที่เพิ่มตัวประกอบ s เข้ากับเทอมส่วนของฟังก์ชันถ่ายโอนก็เสมือนกับเพิ่มแบบของระบบขึ้นไป 1 จึงทำให้ระบบนี้ไม่มีความผิดพลาดที่สภาพคงตัวสำหรับอินพุตแบบ step นอกจากนั้นการที่เพิ่มซีโรให้กับระบบไปพร้อม ๆ กัน ก็ทำให้ความแตกต่างระหว่างจำนวนโพล n และจำนวนซีโร m มีค่าคงที่ ดังนั้นมุมของ asymptote สำหรับทางเดินของรากมีค่าคงเดิม เพราะอย่างไรก็ตามจุดตัดของเส้น asymptotes บนแกนจริง จะเคลื่อนที่เข้าหาจุดกำเนิดมากขึ้น ส่งผลให้

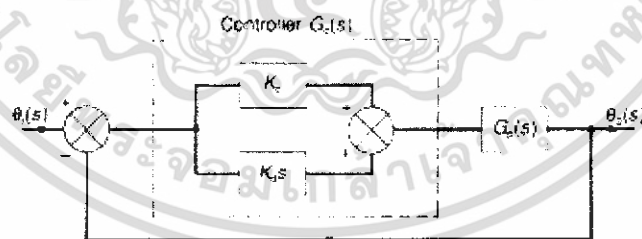
ความเสถียรภาพของระบบลดลง การเพิ่มโพลที่ $s=0$ และซีโรที่ $s=-1/\tau_i$ ทำให้จุดเปลี่ยนไปเท่ากับ $\pm(1/\tau_i)/(n-m)$ ซึ่งทำให้มีค่าเป็นบวกมากขึ้น และจุดตัดจะเคลื่อนที่มาทางขวามือเข้าใกล้จุดกำเนิดมากขึ้น อย่างไรก็ตามการลดลงของความเสถียรสัมพันธ์นี้จะน้อยกว่าการที่ใช้การควบคุมแบบโอเพนลูปอย่างเดียวนั่นเองค่าของ K_p และ K_i จะเป็นค่าที่ใช้กำหนดตำแหน่งของซีโรและโพลของระบบ โดยตำแหน่งของซีโรกำหนดด้วยค่า K_p ในขณะที่ K_i จะเป็นค่าที่ใช้กำหนดโพลระบบแบบปิด

2.6.5 ระบบควบคุมแบบพีดี (PD: Proportional plus Derivative Control)

ถ้าการควบคุมแบบดีใช้ร่วมกับการควบคุมแบบพี ดังที่แสดงในรูปที่ 2.29 ได้ฟังก์ชันถ่ายโอนระบบแบบเปิดเป็น

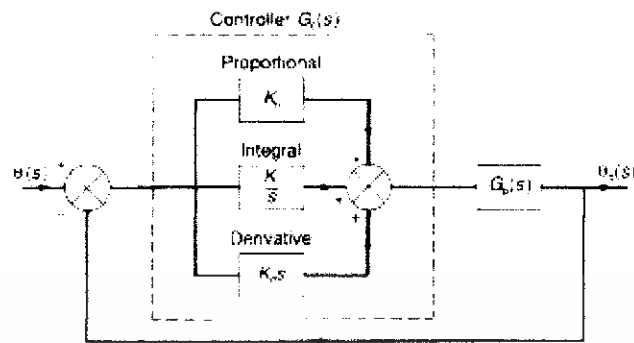
$$\begin{aligned} G_o(s) &= \dots (K_p + K_d s) G_p(s) \\ G_o(s) &= \dots K_p (1 + \tau_d s) G_p(s) \end{aligned} \quad (2.24)$$

เมื่อ $\tau_d = \frac{K_p}{K_d}$ คือ derivative time constant ซึ่งในการควบคุมแบบนี้ จะมีซีโรเพิ่มขึ้นที่ $s = -\tau_d$ และจะเห็นว่าไม่มีการเปลี่ยนแปลงของระบบ ทำให้ไม่มีการเปลี่ยนแปลงค่าความผิดพลาดที่สถานะคงตัว



รูปที่ 2.31 บล็อกไดอะแกรมของระบบควบคุมแบบพีดี

2.7.6 ระบบควบคุมแบบพีไอดี (PID Control)



รูปที่ 2.32 บล็อกไคอะแกรมของระบบควบคุมแบบพีไอดี

การควบคุมโดยใช้แบบพืรวมกับแบบไอและรวมกับแบบดี หรือ ที่เรียก การควบคุมแบบ 3 เทอม (tree-term control) ระบบจะมีลักษณะตามรูปที่ 2.31 ทำให้เอาที่พู่ของตัวควบคุมเมื่อรับ อินพุทเป็นความผิดพลาด $e(t)$ ดังนี้

$$u(t) \dots = \dots K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (2.25)$$

ฟังก์ชันถ่ายโอนของตัวควบคุม จะเป็น

$$G_c(s) \dots = \dots K_p + \frac{K_i}{s} + K_d s \quad (2.26)$$

ซึ่งสามารถจัดรูปได้เป็น

$$G_c(s) \dots = \dots K_p \left(1 + \frac{1}{\tau_i s} + \tau_d s \right) \quad (2.27)$$

ฟังก์ชันถ่ายโอนระบบแบบเปิดของระบบควบคุมแบบพีไอดี ที่แสดงในรูปที่ 2.32 จะเป็น

$$G_o(s) \dots = \dots \frac{K_p (\tau_i s + 1 + \tau_i \tau_d s^2) G_p(s)}{\tau_i s} \quad (2.28)$$

2.7 ไมโครคอนโทรลเลอร์ PIC 16F877

ในปัจจุบันไมโครคอนโทรลเลอร์ตระกูล PIC มีการพัฒนาและผลิตออกมาหลายเบอร์คดียกทั่วไปจะแบ่งออกเป็น 3 แบบคือ

1. สถาปัตยกรรมแบบ 12-Bit Core (Base-Line) เป็นกลุ่มของไมโครคอนโทรลเลอร์ที่มีขนาดเล็กมีโครงสร้างของคำสั่งเพียง 12 bits และค่อนข้างมีข้อจำกัดในการใช้งานเนื่องจากมีหน่วยความจำ RAM และ STACK ค่อนข้างจำกัด
2. สถาปัตยกรรมแบบ 14-Bit Core (Mid-Range) เป็นกลุ่มของไมโครคอนโทรลเลอร์ที่มีขนาดกลางมีโครงสร้างของคำสั่ง 14 bits มีทั้งแบบที่โปรแกรมได้ครั้งเดียว (OTP : One Time Programmable) และแบบแฟลช (Flash Memory)
3. สถาปัตยกรรมแบบ 16-Bit Core (High-End) เป็นกลุ่มของไมโครคอนโทรลเลอร์ที่อยู่ในกลุ่มระดับสูงซึ่งได้มีการพัฒนาโครงสร้างสถาปัตยกรรม ทั้งในเรื่องของหน่วยความจำ ความเร็ว และ คุณสมบัติอื่นๆ ที่เหนือกว่าสองกลุ่มที่ผ่านมา มีการจัดวางหน่วยความจำ โปรแกรมอยู่ในเพจ (Page) เดียวกัน ทำให้ไม่มีปัญหาเรื่องรอยต่อของหน่วยความจำ

นอกจากนี้แล้ว PIC ยังแบ่งออกเป็นประเภทของหน่วยความจำอีกด้วย โดยจะมีการจำแนกเป็น 3 ประเภทคือ

1. C เช่น PIC16CXXX คือ มีโครงสร้างหน่วยความจำเป็น EPROM จัดอยู่ในจำพวกอุปกรณ์ OTP แต่สามารถลบได้ด้วยแสง UV
2. CR เช่น PIC16CRXXX คือ มีโครงสร้างหน่วยความจำเป็น ROM จัดอยู่ในจำพวกอุปกรณ์ OTP ไม่สามารถลบได้
3. F เช่น PIC16FXXX คือ มีโครงสร้างหน่วยความจำเป็น FLASH Memory สามารถทำการลบแล้วโปรแกรมซ้ำได้หลายครั้ง

2.7.1 แรงดันในการทำงาน

ช่วงแรงดันการทำงานของ PIC โดยปกติมาตรฐานแล้วจะอยู่ระหว่าง 4.5-6.0 V แต่จะมีบางเบอร์ที่ออกแบบมาให้สามารถทำงานได้ในช่วงแรงดันต่ำ ประมาณ 2.5-6 V ได้ ซึ่งจะมีการระบุได้คให้ทราบ โดยมีการเพิ่มรหัสตัว L เข้าไปในเบอร์ของอุปกรณ์ เช่น PIC16LFXXX เป็นต้น

2.7.2 สัญญาณนาฬิกา

PIC จะใช้เป็นสัญญาณนาฬิกา โดยมองเป็นลักษณะของวงรอบ (Cycle) ซึ่งระบุเอาไว้ว่า 1 คำสั่งนั้นจะประกอบไปด้วย 1-2 วงรอบ โดยแต่ละวงรอบนั้นจะแบ่งเป็น 4 ส่วน คือ Q1, Q2, Q3 และ Q4 ด้วยเหตุนี้ ความเร็วโดยรวมของ PIC จะเท่ากับค่าความถี่ของสัญญาณนาฬิกาหารด้วย 4

2.7.3 คุณสมบัติของ PIC16F877

- ซีพียูเป็นแบบ RISC (Reduced Instruction-Set) มีคำสั่งใช้งาน 35 คำสั่ง
- คำสั่งหนึ่งๆ ใช้เวลาทำงาน 1 ถึง 2 Cycle
- ทำงานได้สูงสุดที่ 20 MHz (PIC16F877-20/P)
- สัญญาณนาฬิกา มีหลายโหมดให้เลือกใช้งาน
 - โหมด RC ใช้วงจรภายในไมโครคอนโทรลเลอร์กำเนิดสัญญาณนาฬิกา
 - โหมด LP ใช้คริสตัลพลังงานต่ำ ความถี่สูงสุดไม่เกิน 200 KHz
 - โหมด XT ใช้คริสตัลความถี่ตั้งแต่ 100 KHz – 4 MHz
 - โหมด HS ใช้คริสตัลความถี่สูง
- มี I/O พอร์ตทั้งหมด 5 พอร์ต [PORT A (6 bits), B (8 bits), C (8 bits), D (8 bits), E (3 bits)]
- หน่วยความจำโปรแกรมเป็นแบบ Flash มีขนาด 8 KWord (1 word=14 bits)
- มี RAM ขนาด 368 bytes
- ทำงานแบบ Pipe-line ทำให้ ณ เวลาหนึ่งสามารถทำคำสั่ง Fetch และ Execute ได้พร้อมกัน
- คอยสนองกับอินเตอร์รัพท์ได้ทั้งหมด 14 แหล่ง
- มี Stack ให้ใช้ได้สูงสุด 8 ระดับ
- มีระบบ POR (Power On Reset), PWRT (Power Up Timer), OST (Oscillator Start-up timer)
- มีระบบ WDT (Watchdog Timer)
- มีระบบ CP (Code Protection) และสามารถเลือกระดับการป้องกันได้
- มีโหมด ประหยัดพลังงาน
- สามารถโปรแกรมด้วยไฟ +5 VDC ได้
- สามารถโปรแกรมแบบ In-Circuit Serial Programming
- ทำงานที่ไฟเลี้ยง 2 VDC ถึง 5.5 VDC
- Current Sink และ Current Source ของ PORT อยู่ที่ 25 mA
- มี Timer/Counter 3 ตัว
- มีโมดูล Capture/Compare/PWM 2 ชุด
- มี A/D Converter แบบ 10 bits จำนวน 8 ช่องนำเข้าไปในตัวเอง
- มีระบบ USART สำหรับการสื่อสารแบบอนุกรม
- มีระบบตรวจระดับไฟเลี้ยง BODEN (Brown-Out Detection) เพื่อสร้างสัญญาณรีเซต ซีพียู BOR (Brown-Out Reset)

2.7.3 รีจิสเตอร์หลักของ PIC16F877

1. รีจิสเตอร์ Configuration word

มีขนาด 14 bits อยู่ที่ Address 2007h ใน Program Memory โดยการทำงานเบื้องต้นของ PIC จะถูกกำหนดที่หน่วยความจำตรงนี้ ไม่ว่าจะเป็น Enable/Disable Power-up timer, Enable/Disable Watchdog timer, Oscillator Selection bits (กำหนดที่มาของสัญญาณนาฬิกา) หน่วยความจำที่ตำแหน่งนี้ จะต้องกำหนดในขณะที่เขียนโปรแกรมลงสู่ Flash Memory ของ PIC

2. รีจิสเตอร์ STATUS

เป็นรีจิสเตอร์ที่ใช้เก็บข้อมูลแสดงสถานะการทำงานของ PIC16F877

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	TO	PD	Z	DC	C
bit 7							bit 0

3. รีจิสเตอร์ W เป็นรีจิสเตอร์ที่มีบทบาทสำคัญ เพราะในการประมวลผลทางคณิตศาสตร์ จะต้องกระทำผ่านรีจิสเตอร์ W และยังทำหน้าที่เป็นตัวกลางในการส่งผ่านสถานะของ Output ไปยัง I/O PORT อีกด้วย

4. Program Counter (PC) เป็นรีจิสเตอร์พิเศษที่ใช้ระบุ Address ของ Program Memory ที่กำลังทำการประมวลผล ซึ่งจะเป็น Counter ขนาด 13 bits โดยทั่วไปแล้ว Counter ตัวนี้เพิ่มขึ้น 1 ทุกๆครั้งที่มีการประมวลผลคำสั่งเกิดขึ้น 1 ครั้ง ซึ่งค่าที่แสดงก็คือตำแหน่งของคำสั่งต่อไปที่จะทำการประมวลผล แต่เมื่อประมวลผลคำสั่ง JUMP ตัว counter จะมีค่าเท่ากับตำแหน่งที่คำสั่ง JUMP นั้นอ้างอิงถึง

5. Stack เป็นหน่วยความจำสำรองสำหรับเก็บค่าของ PC ขนาด 13 bits โดยเก็บข้อมูลได้ 8 ระดับ โดยเก็บตำแหน่งของ PC เข้าเมื่อมีคำสั่ง CALL และส่งตำแหน่งที่เก็บไว้ออกไปยัง PC เมื่อมีคำสั่ง RETURN โดยการเก็บจะเป็นแบบ LIFO (Last In First Out)

6. PORTA, PORTB, ... เก็บค่าสถานะของ PORT นั้นๆ

7. TRISA, TRISB, ... ใช้กำหนดทิศทางของ PORT นั้นๆ ว่าขาใดเป็น Input หรือ Output โดยถ้ากำหนดให้เป็น 0 จะเป็น Output ถ้าให้เป็น 1 จะเป็น Input

8. รีจิสเตอร์ CCP1CON และ CCP2CON เป็นรีจิสเตอร์ควบคุมโมดูล CCP1 และ CCP2 (Address 17h, 1Dh)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CCPxX	CCPxY	CCPxM3	CCPxM2	CCPxM1	CCPxM0
bit 7							bit 0

2.8 การทำงานในโหมด PWM

1. โหมดค่าควิตซ์ไชเกิลของสัญญาณ PWM ที่สร้างขึ้นมานี้สามารถเปลี่ยนแปลงได้ตลอดเวลาจากการเปลี่ยนค่าที่ CCPRxL และ 2 บิตใน CCPxCON แต่ค่าของคาบเวลาหรือความถี่เปลี่ยนแปลงไม่ได้จนกว่าจะหยุดการทำงาน แล้วกำหนดค่าของคาบเวลาลงในรีจิสเตอร์ PR2 ใหม่ ดังนั้นสัญญาณ PWM ที่สร้างขึ้นจึงสามารถกำหนดค่าควิตซ์ไชเกิลได้ตามความต้องการ

2. การกำหนดคาบเวลาของสัญญาณ PWM ทำได้โดยการเขียนข้อมูลลงในรีจิสเตอร์ PR2 (รีจิสเตอร์คาบเวลาของไทมเมอร์ 2) แล้วนำค่าของ PR2 มาคำนวณหาค่าคาบเวลาของสัญญาณ PWM

$$\text{คาบเวลาของสัญญาณ PWM} = (\text{ค่าในรีจิสเตอร์} 2 + 1) \times 4 \times T_{\text{osc}} \times \text{ค่าปริสเกลเลอร์ของ TMR2}$$

โดยที่ T_{osc} คือคาบเวลาของสัญญาณนาฬิกาหลักมีหน่วยเป็นวินาทีค่ารีจิสเตอร์ทั้งหมดคำนวณในเลขฐานสิบ

$$\text{ความถี่ของสัญญาณ PWM} = 1/\text{คาบเวลาของสัญญาณ PWM}$$

3. การกำหนดค่าควิตซ์ไชเกิลของสัญญาณ PWM ทำได้โดยการเขียนข้อมูลไปยังรีจิสเตอร์ CCPRxL ร่วมกับบิต 5-4 ของรีจิสเตอร์ CCPxCON ทำให้สามารถกำหนดความถี่ของสัญญาณ PWM ได้สูงสุดถึง 10 บิต โดย 8 บิตบนจะใช้ข้อมูลรีจิสเตอร์ CCPRxL ส่วนใน 2 บิตล่างใช้ข้อมูลในบิต 5-4 ของรีจิสเตอร์ CCP1CON

$$\text{ควิตซ์ไชเกิลของสัญญาณ PWM} = (\text{CCPR1L:CCP1CON} \langle 5:4 \rangle)_{10\text{บิต}} \times T_{\text{osc}}$$

โดยที่ T_{osc} คือคาบเวลาของสัญญาณนาฬิกาหลักมีหน่วยเป็นวินาทีค่าในรีจิสเตอร์ทั้งหมดคำนวณในรูปของฐานสิบ

4. ความละเอียดของ PWM จะขึ้นอยู่กับบิตปริสเกลเลอร์ของไทมเมอร์ 2 ในขณะที่ความละเอียดสูงสุดของสัญญาณ PWM จะสัมพันธ์กับความถี่ที่กำหนด ความถี่ของสัญญาณนาฬิกา และค่าปริสเกลเลอร์

$$\text{ความละเอียด (บิต)} = \frac{\log\left(\frac{f_{\text{osc}}}{f_{\text{PWM}} \times N}\right)}{\log 2}$$

โดยที่ f_{osc} คือ ความถี่ของสัญญาณนาฬิกาหลักมีหน่วยเป็น Hz

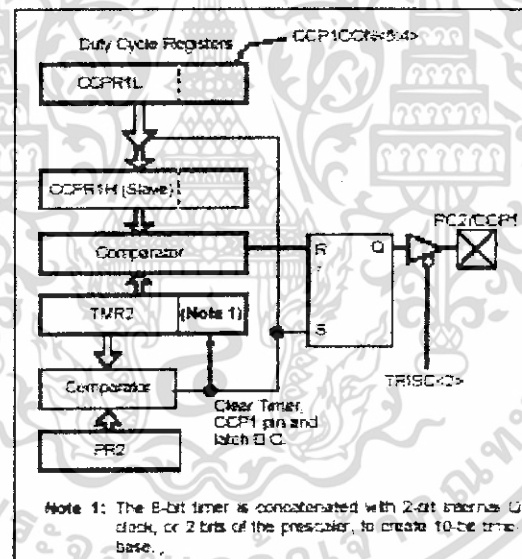
f_{PWM} คือความถี่ของสัญญาณ PWM มีหน่วยเป็น Hz

N คือ ค่าปริสเกลเลอร์ในไทมเมอร์ 2

สัญญาณ PWM จะมีความละเอียดสูงสุดเมื่อค่าของปริสเกลเลอร์เท่ากับ 1 แล้ว CCPx จะกำเนิดสัญญาณ โมดูลชั้นทางความกว้างพัลส์ (Pulse Width Modulation: PWM) ความละเอียด 10 บิต สัญญาณ PWM ที่สร้างจะส่งออกทางขา RC2/CCP1 หรือ RC1/T1OSI/CCP2

1 หลักการสร้างสัญญาณ PWM

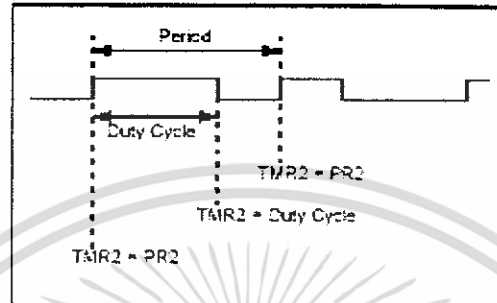
- กำหนดค่าให้เรจิสเตอร์ PR2 เพื่อกำหนดคาบเวลาของสัญญาณ PWM
- กำหนดค่าความถี่ไซเคิล โดยเขียนข้อมูลลงในเรจิสเตอร์ CCPRxL ร่วมกับบิต 5-4 ของเรจิสเตอร์ CCPxCON
- กำหนดให้ขาพอร์ต RC2/CCP1 หรือ RC1/T1OSI/CCP2 เป็นเอาต์พุตเพื่อเป็นทางออกของสัญญาณ PWM โดยการเคลียร์บิตที่ 2 หรือ 1 ของเรจิสเตอร์ TRISC
- กำหนดให้ค่าปริสเกลเลอร์ของ TMR 2 และเอ็นเนเบิลการทำงานของไมเมอร์ 2 กำหนดให้โมดูล CCPx ทำงานในโหมด PWM



รูปที่ 2.33 การทำงานของโมดูล CCP1 เพื่อสร้างสัญญาณ PWM

จากรูปที่ 2.33 และ 2.34 ทันทีที่โมดูล CCPx เริ่มทำงานค่าของ TMR2 จะเพิ่มขึ้นจนเท่ากับ PR2 ที่ขาพอร์ต RC2/CCP1 หรือ RC1/T1OSI/CCP2 จะเกิดลอจิก “1” และคลลสถานะอยู่เช่นนั้นแล้วค่าของ TMR2 จะเคลียร์แล้วเพิ่มขึ้นใหม่ หลังจากนั้นค่าความถี่ไซเคิลที่กำหนดไว้ในเรจิสเตอร์ CCPRxL และ 2 บิตใน CCPxCON จะถูกถ่ายทอดไปยัง CCPRxH และ 2 บิตในหน่วยความจำพิเศษเพื่อเปรียบเทียบกับค่าใน TMR2 และค่าปริสเกลเลอร์ 2 บิต ซึ่งเพิ่มค่าขึ้นเรื่อยๆจนกระทั่งเมื่อข้อมูล

ทั้ง 2 กลุ่มเท่ากัน จะส่งสัญญาณไป ทำให้ขาพอร์ต RC2/CCP1 หรือ RC1/T1OSI/CCP2 กลับมาเป็นลอจิก "0" และคลลสถานะอยู่เช่นนั้นจนกระทั่งค่าของ TMR2 เท่ากับ PR2 อีกครั้ง ก็จะเกิดสัญญาณลอจิก "1" เป็นการเริ่มต้นรอบใหม่ของสัญญาณ และจะทำงานวนรอบเช่นนั้นจนกระทั่งมีการดิสเอเบิล(disable)



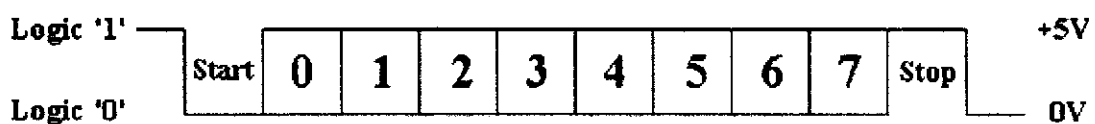
รูปที่ 2.34 เาต์พุตของสัญญาณ PWM

ความถี่ของสัญญาณ PWM (kHz)	1.22	4.88	19.53	78.12	156.3	208.3
ค่าปริสเกลเลอร์ (1, 4, 16)	16	4	1	1	1	1
ค่ารีจิสเตอร์ PR2	FFh	FFh	FFh	3Fh	1Fh	17h
ความละเอียดสูงสุด (บิต)	10	10	10	8	7	5.5

ตาราง 2.1 ตัวอย่างขนาดความถี่และความละเอียดของสัญญาณ PWM ที่ความถี่ 20 MHz

2.9 รูปคลื่น สัญญาณ RS-232

การสื่อสารโดย RS-232 เป็นการสื่อสารแบบ asynchronous หมายความว่าสัญญาณ clock ที่ใช้ควบคุมจังหวะไม่ได้ส่งไปพร้อมกับ Data แต่จะใช้ start bit เป็นตัว sync. ในแต่ละ word ของการสื่อสารและใช้สัญญาณ clock ภายในของแต่ละด้านเป็นตัวให้จังหวะเอง



รูปที่ 2.35 รูปคลื่นของสัญญาณที่ส่ง

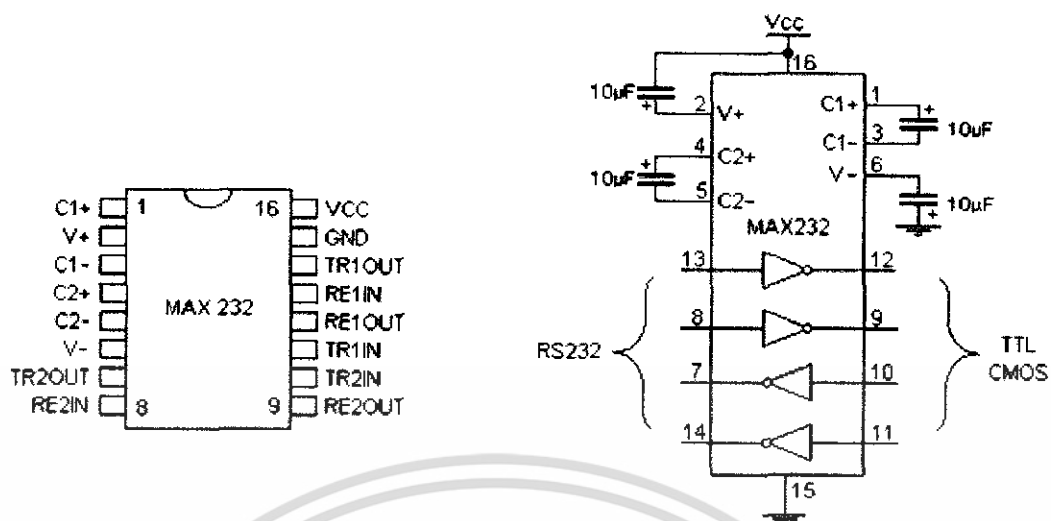
แสดงลักษณะของสัญญาณจาก UART เมื่อใช้ format แบบ 8N1 คือ 8 data bits ไม่มี parity bit และมี 1 stop bit ขณะที่ idle จะอยู่ในสถานะ “Mark” หรือ logic “1” การส่งจะเริ่มจากการส่ง start bit คือ logic “0” และตามด้วย LSB bit จนหมด data bits และถ้ามี parity bit ก็จะส่งที่จุดนี้แล้ว ลงท้ายด้วย stop bit ซึ่งมีค่าเป็น logic “1” ในรูปได้แสดง bit ที่ต่อถัดจาก stop bit ซึ่งมีค่าเป็น logic “0” หมายความว่า เป็น start bit ของ การส่ง word ถัดไป แต่ถ้ายังไม่มี การส่ง word ถัดไป ก็คืออยู่ในสถานะของ logic “1” ซึ่งเป็นสถานะของ idle และถ้าสายอยู่ในสถานะของ logic “0” นานกว่า เวลาของการส่ง 1 full word ระบบจะถือว่าเป็นสัญญาณ “Break” เพื่อหยุดการสื่อสาร ดังนั้นต้องไม่ลืมที่จะส่งในสายกลับสู่สถานะ idle เมื่อสิ้นสุดการส่งการรับ-ส่งข้อมูลในลักษณะนี้เรียกว่าแบบ frame คือมีกรอบปิดล้อมข้อมูลไว้ด้วย start bit และ stop bit

ตัวแปลงสัญญาณ RS-232

สัญญาณ RS-232 มีค่าแรงไฟต่างจากที่ใช้ใน UART ดังนั้นจึงต้องมีตัวแปลงสัญญาณ เพื่อแปลงระดับสัญญาณให้เหมาะสมก่อนที่จะเชื่อมต่อกับพอร์ตคอนกรีต หรือ RS-232 port ของคอมพิวเตอร์

สัญญาณ RS-232 นั้น logic “0” จะมีค่า +3 V ถึง +25 V และ logic “1” จะมีค่า -3 V ถึง -25 V ส่วนค่าระหว่าง -3 V ถึง +3 V เป็นค่า undefined ระดับสัญญาณนี้ใช้กับทุกสัญญาณไม่ใช่เฉพาะสัญญาณรับ-ส่งข้อมูลเท่านั้นแต่ยังรวมถึงสัญญาณควบคุมต่าง ๆ เช่น DTR, RTS, CTS เป็นต้น

IC ที่ใช้มักจะเป็นเบอร์ 1488 (RS-232 Driver) และ 1489 (RS-232 Receiver) โดยภายในแต่ละตัวจะประกอบด้วย inverter 4 ตัวและต้องการไฟเลี้ยง 2 ชุดคือ +7.5 V ถึง +15 V และ -7.5 V ถึง -15 V ซึ่งอาจจะมีปัญหาในเครื่องที่มีไฟเลี้ยง +5 V เพียงชุดเดียว แต่ก็ยังมี IC อีกตัวหนึ่งคือเบอร์ MAX-232 ซึ่งมีวงจร charge pump สามารถสร้างไฟ +10 V และ -10 V จากไฟ +5 V ได้ พร้อมทั้งมี 2 Tx และ 2 Rx อยู่ใน package เดียวกัน และรองรับ baud rate ได้ถึง 120 Kbps จึงสะดวกมากเพราะใช้ IC เพียงตัวเดียว รูปข้างล่างคือ MAX-232



รูปที่ 2.36 แสดงโครงสร้างภายในและตำแหน่งขาต่างๆของ Max232



บทที่ 3

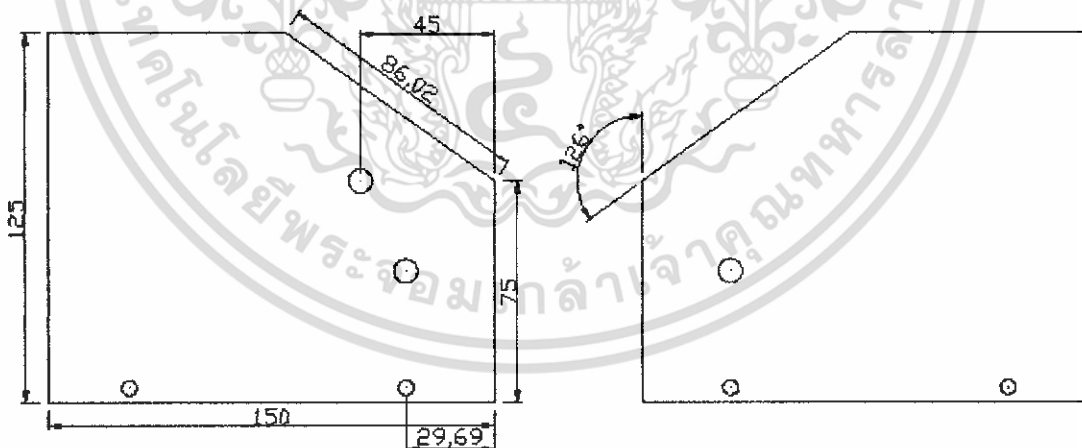
โครงสร้างและการออกแบบ

โครงสร้างของแขนกลใช้เป็นแบบลักษณะของแขนกลแบบ Articulated Configuration ซึ่งมีข้อหมุนทั้งหมด 4 ข้อ ซึ่งทำงานคล้ายกับแขนของมนุษย์ จึงมีความเหมาะสมกับโครงงานนี้มากที่จะนำมาศึกษาและลงมือปฏิบัติ

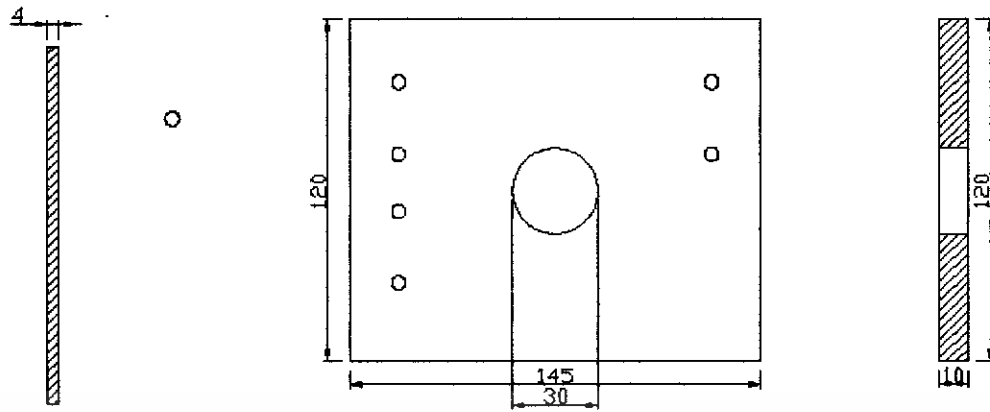
3.1 โครงสร้างและการออกแบบทางกล

3.1.1 โครงสร้างส่วนฐาน

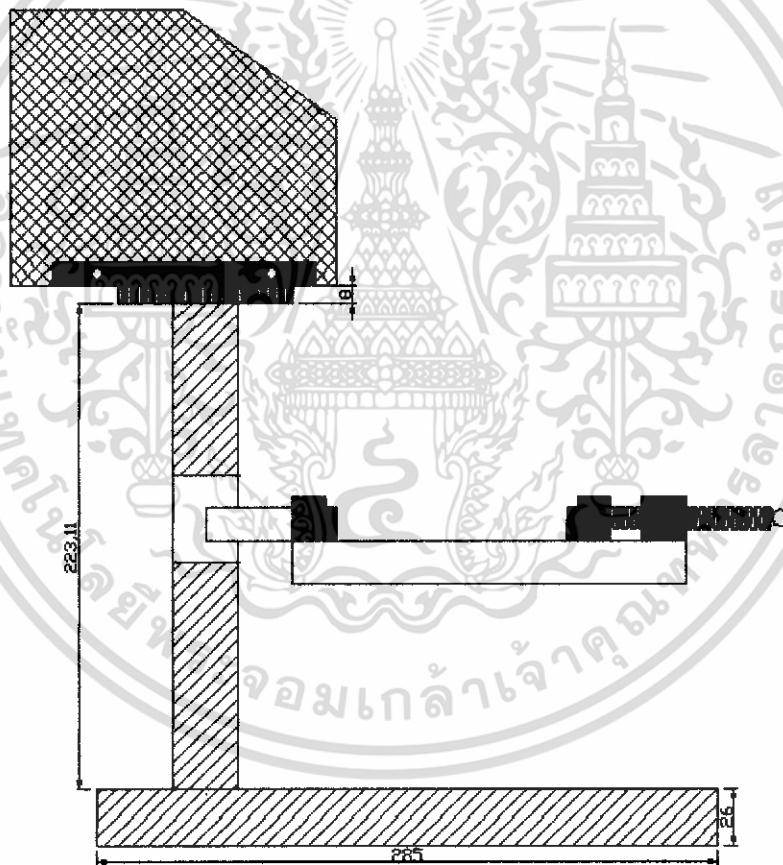
โครงสร้างส่วนฐานนี้เป็นส่วนที่ต้องรับแรงทั้งหมดของตัวแขนกล ดังนั้นส่วนของฐานนี้จึงต้องมีความแข็งแรงพอที่จะรับแรงและน้ำหนักของแขนกลได้ ซึ่งวัสดุที่นำมาประกอบตัวฐานของแขนกล จะใช้วัสดุจำพวกโลหะเหล็กที่มีความแข็งแรง โดยตัวฐานมีลักษณะเป็นแบบสี่เหลี่ยมผืนผ้าที่มีขนาดของความกว้าง 15 เซนติเมตร(Cm) ยาว 28 เซนติเมตร(Cm) หนา 2.6 เซนติเมตร(Cm) เหตุผลที่ฐานล่างต้องมีความกว้าง ยาว หนาตามดังรูป(3.1)เพราะว่าแขนกลที่อยู่ด้านบนมีน้ำหนัก มีแรง ไน้มถ่วง ฐานล่างจึงต้องมีแรงที่ไว้สำหรับ ยึดติดกับพื้นด้วยแรงของน้ำหนักฐานที่มีความหนาซึ่งแสดงดังรูปที่ (3.1)



รูปที่ 3.1 แสดงส่วนประกอบของฐานหมุน



รูปที่ 3.2 แสดงส่วนประกอบส่วนฐาน



รูปที่ 3.3 รูปแสดงการออกแบบส่วนฐาน

โครงสร้างส่วนต่อไปคือโครงสร้างส่วนหมุนซึ่งเป็นส่วนประกอบหนึ่งที่รวมอยู่ในส่วน
ของฐาน ซึ่งส่วนนี้จะมีจุดหมุน 1 จุดดังรูปที่(3.1)ซึ่งจุดหมุนนี้สามารถหมุนได้ 360องศา โดยใช้

หลักการส่งแรงจากมอเตอร์ไปให้กับส่วนหมุนนี้ด้วยชุดเฟือง ดังนั้นที่ส่วนหมุนของฐานนี้จึงติดตั้งเฟืองขนาด 120 ฟันเฟือง ได้รับแรงที่ส่งมาจากเฟืองของมอเตอร์ที่มีขนาดฟันเฟือง 24 ฟันเฟือง โดยมอเตอร์หมุนด้วยความเร็ว 30 รอบ/นาที(rpm) จากการคำนวณส่วนหมุนของฐานนี้จะหมุนด้วยความเร็วดังสมการ

$$N = (R2*St)/R1$$

N = จำนวนรอบของการหมุน (รอบ/นาที)

R1 = จำนวนฟันเฟืองของตัวขับ

R2 = จำนวนฟันเฟืองของตัวถูกขับ

St = ความเร็วรอบของมอเตอร์ (รอบ/นาที)

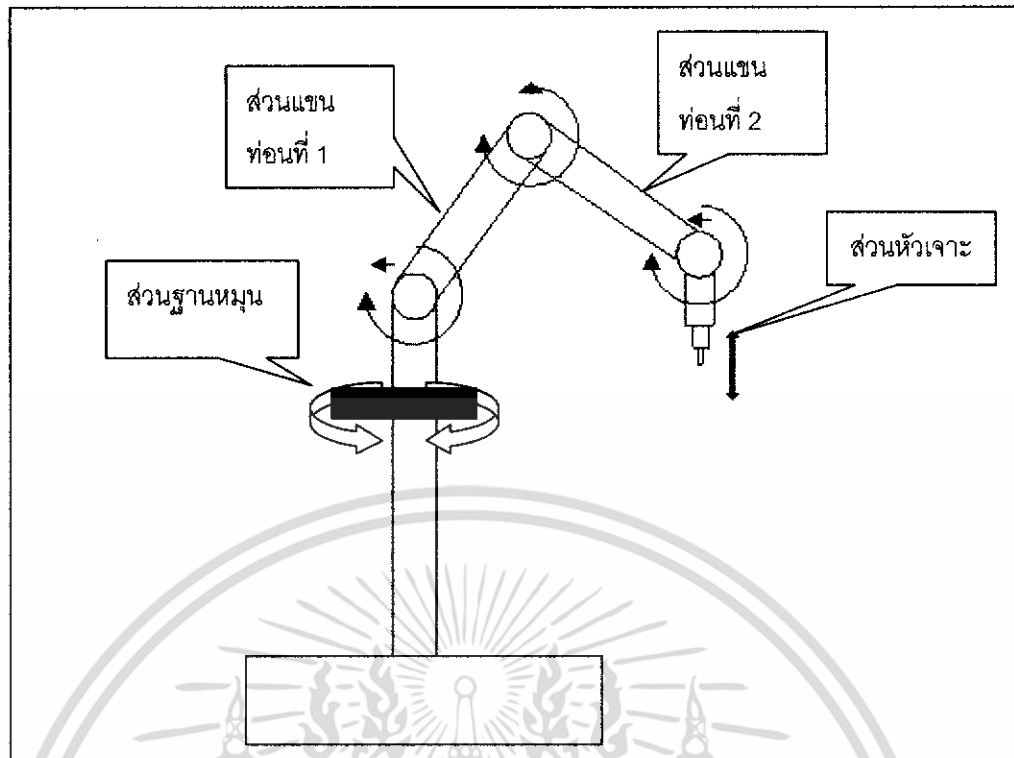
ซึ่งเราจะได้ความเร็วในการหมุนของส่วนฐานล่างนี้มีค่าเท่ากับ

$$\begin{aligned} N &= \frac{24 \times 30}{120} \\ &= 6 \quad \text{รอบ/นาที} \end{aligned}$$

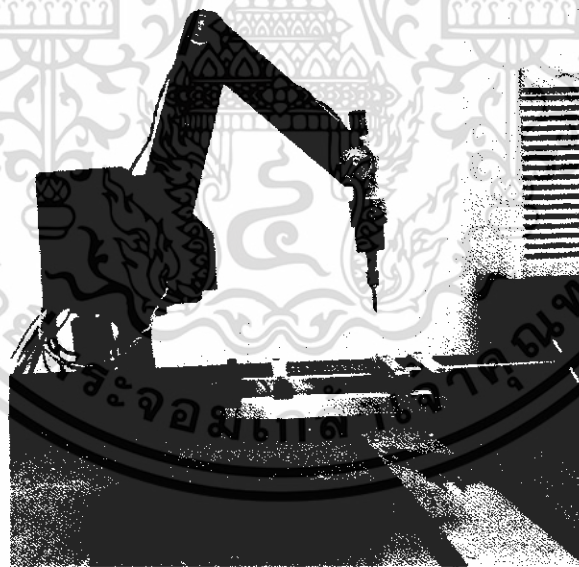
โครงสร้างส่วนข้อต่อ

โครงสร้างส่วนของข้อต่อนี้จะแบ่งออกอยู่ด้วยกันทั้งหมด 3 ข้อต่อ โดยเริ่มจากส่วนที่ติดกับส่วนหมุนของฐานไปจนถึงส่วนของหัวเจาะแบ่งได้คือ

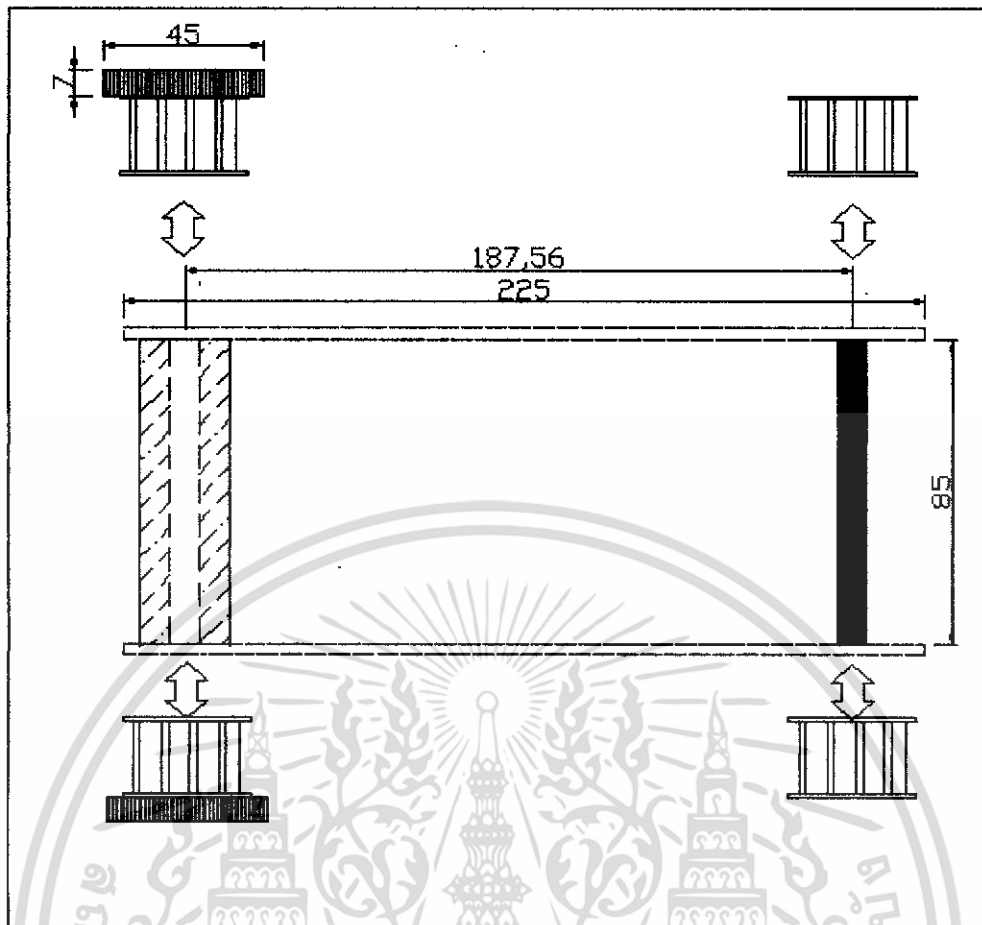
-โครงสร้างส่วนเคลื่อนที่ที่หนึ่ง ส่วนโครงสร้างส่วนนี้เป็นส่วนที่ทำหน้าที่เชื่อมต่อระหว่างโครงสร้างส่วนฐานหมุนกับ ส่วนของโครงสร้างส่วนเคลื่อนที่ที่สอง ซึ่งโครงสร้างส่วนนี้จะมีจุดหมุนติดอยู่ที่ส่วนฐานหมุน มีขนาดความกว้างของแกน 8.5 เซนติเมตร(Cm)และยาว 22.5 เซนติเมตรประกอบขึ้น โดยใช้วัสดุจำพวกโลหะเหล็กที่มีความหนาขนาด 3 มิลลิเมตร(mm)ด้านล่างของส่วนแกนนี้จะมี ล้อหมุนซึ่งล้อหมุนนี้จะประกอบติดอยู่กับเฟืองขนาด 46 ฟันเฟือง ล้อหมุนนี้จะติดตั้งอยู่ 2 ด้าน เพื่อไว้สำหรับรับแรงหมุนจากมอเตอร์ โดยที่ล้อด้านที่(1)จะเป็นล้อส่งแรงหมุนที่ได้รับมาจากมอเตอร์ เพื่อที่จะส่งแรงหมุนนี้ไปใช้ในการเคลื่อนที่ของโครงสร้างส่วนเคลื่อนที่ที่สอง (แกนข้อต่อที่2)ส่วนล้อหมุนด้านที่(2)จะเป็นล้อหมุนที่ใช้สำหรับรับแรงที่ได้จากมอเตอร์เพื่อส่งผ่าน ผ่านสายพานไปหมุนส่วนของหัวเจาะชิ้นส่วนต่างๆของแกนท่อนี้ เมื่อประกอบตัวแกนแล้วมีน้ำหนักประมาณ 0.55 กิโลกรัม (kg)ดังแสดงในรูปที่(3.2)



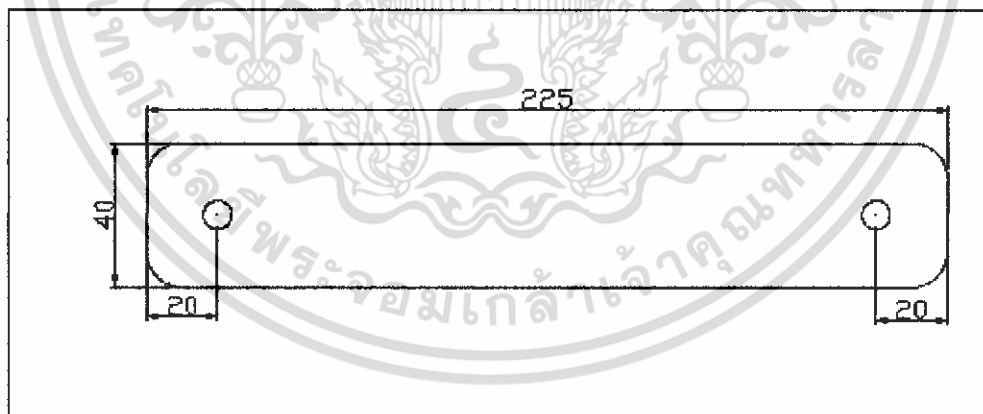
รูปที่ 3.4 แสดงแขนกลจำลอง



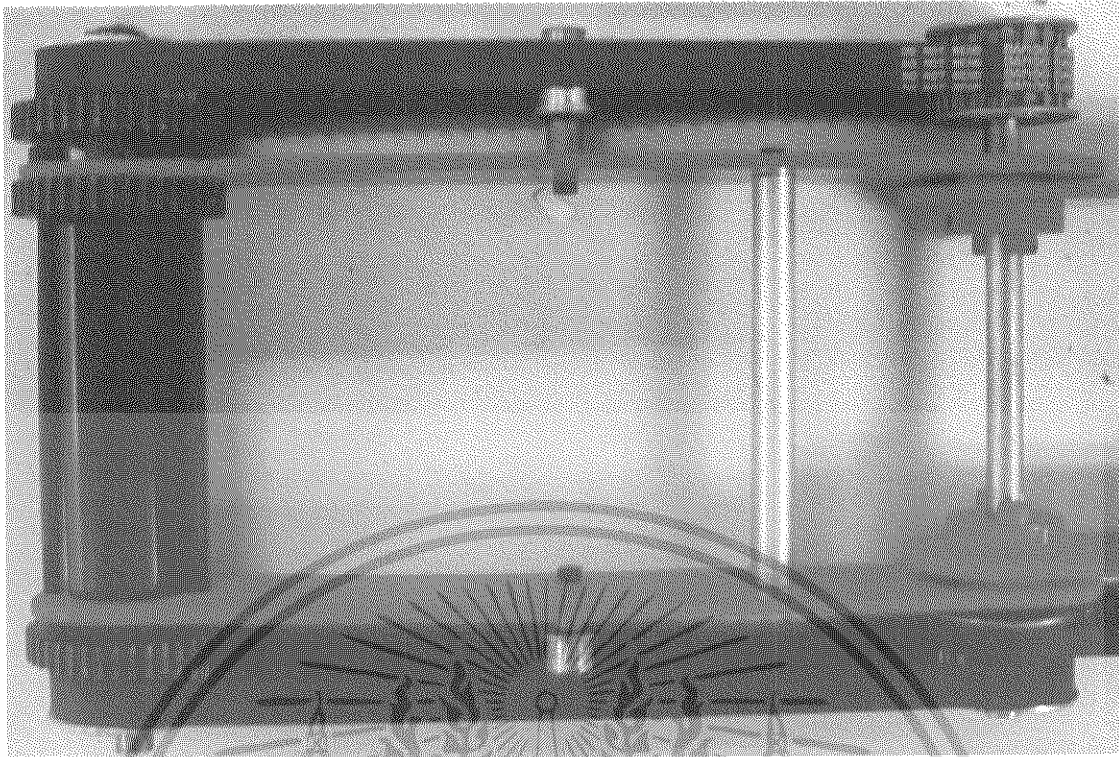
รูปที่ 3.5 แสดงภาพของแขนกลจากของจริง



รูปที่ 3.6 แสดงส่วนแกนท่อนที่ 1

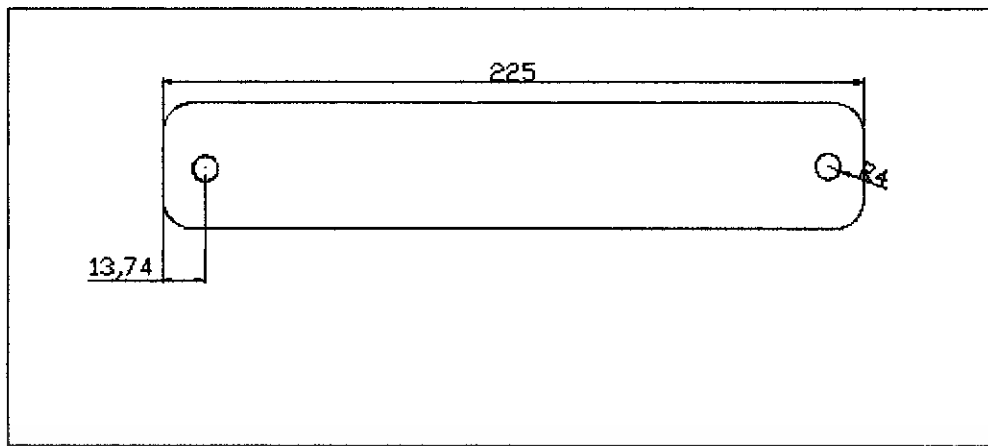


รูปที่ 3.7 แสดงด้านข้างของแกนท่อนที่ 1

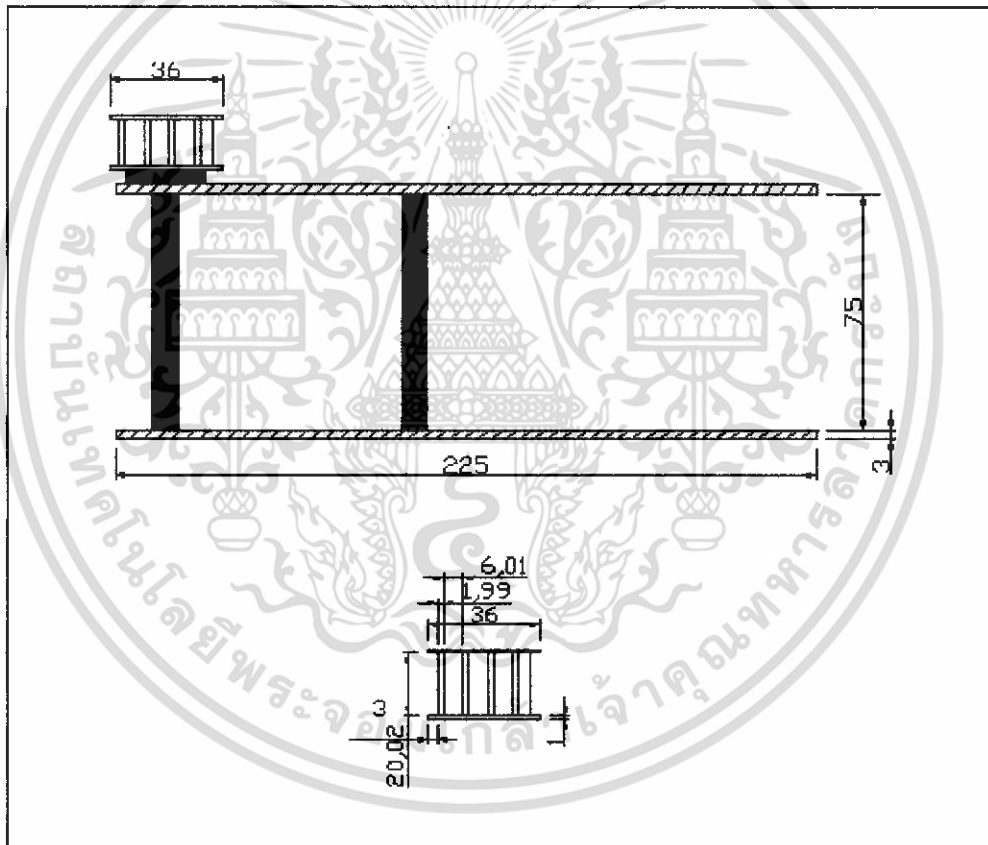


รูปที่ 3.8 รูปแสดงการออกแบบส่วนเคลื่อนที่ที่หนึ่งของจริง

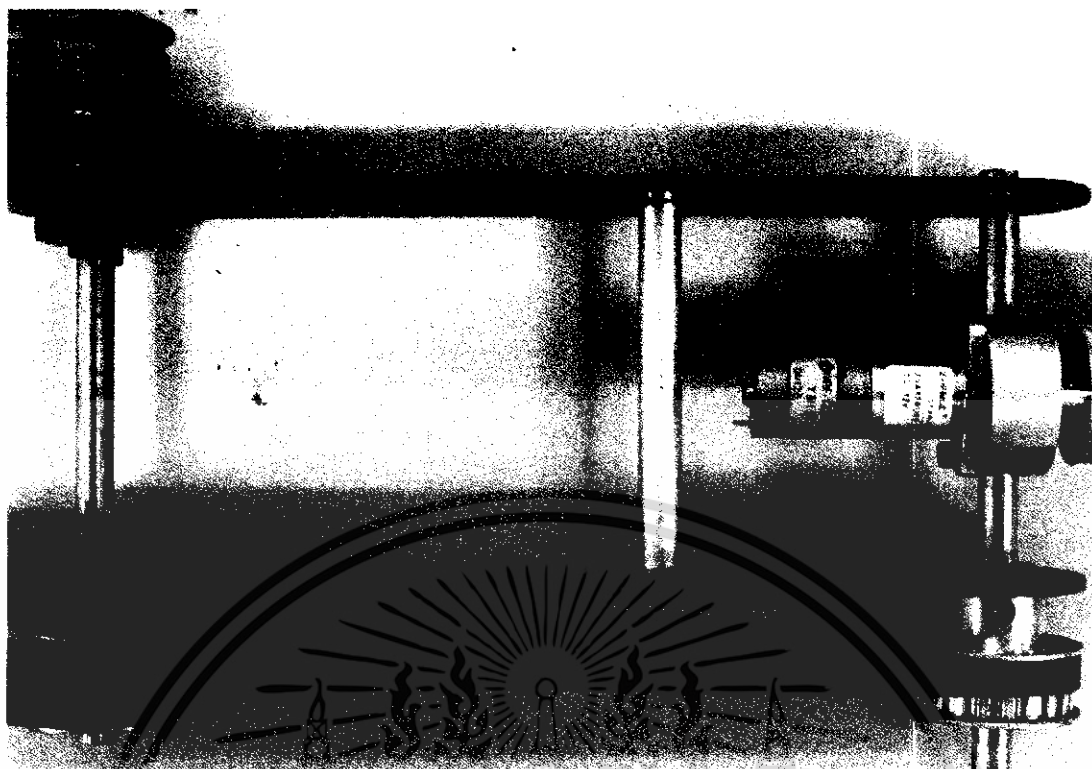
-โครงสร้างส่วนเคลื่อนที่สอง (แกนข้อต่อที่ 2) ส่วนโครงสร้างส่วนนี้เป็นส่วนที่ทำหน้าที่เชื่อมต่อระหว่างโครงสร้างส่วนเคลื่อนที่ที่หนึ่ง (แกนข้อต่อที่ 1) กับส่วนของหัวเจาะซึ่งตัวของหัวเจาะติดตั้งที่แกนหมุนดังที่แสดงในรูป(3.3)แกนข้อต่อส่วนนี้มีขนาดความกว้าง 7.5 เซนติเมตร(Cm) และยาว 22.5 เซนติเมตร(Cm) ปลายด้านหนึ่งจะเป็นแกนหมุนที่เป็นจุดหมุนที่เป็นจุดหมุนระหว่างแกนข้อต่อที่ 1 กับ แกนข้อต่อที่ 2 แกนข้อต่อที่ 2 นี้จะเคลื่อนที่ตามการหมุนของมอเตอร์ที่ติดตั้งอยู่ที่ส่วนของฐานหมุนซึ่งจะเชื่อมต่อกัน โดยใช้สายพานเป็นตัวส่งแรงบิด แล้วบนแกนท่อนี้จะประกอบชุดขับหัวเจาะ คือมีล้อหมุนสองล้อที่ติดตั้งหนึ่งล้อที่ปลายของแกนท่อนที่หนึ่ง และอีกหนึ่งล้อหมุนจะติดตั้งที่ปลายของแกนท่อนที่สองล้อส่วนนี้จะยึดติดกับแกนหมุน แกนท่อนี้ถ้าประกอบแล้วคือ มีหัวเจาะ สายพาน ชุดขับหัวเจาะส่วนประกอบส่วนนี้มีน้ำหนัก 0.63 กิโลกรัม(kg)



รูปที่ 3.9 แสดงด้านข้างของแขนท่อนที่ 2



รูปที่ 3.10 แสดงแขนท่อนที่ 2

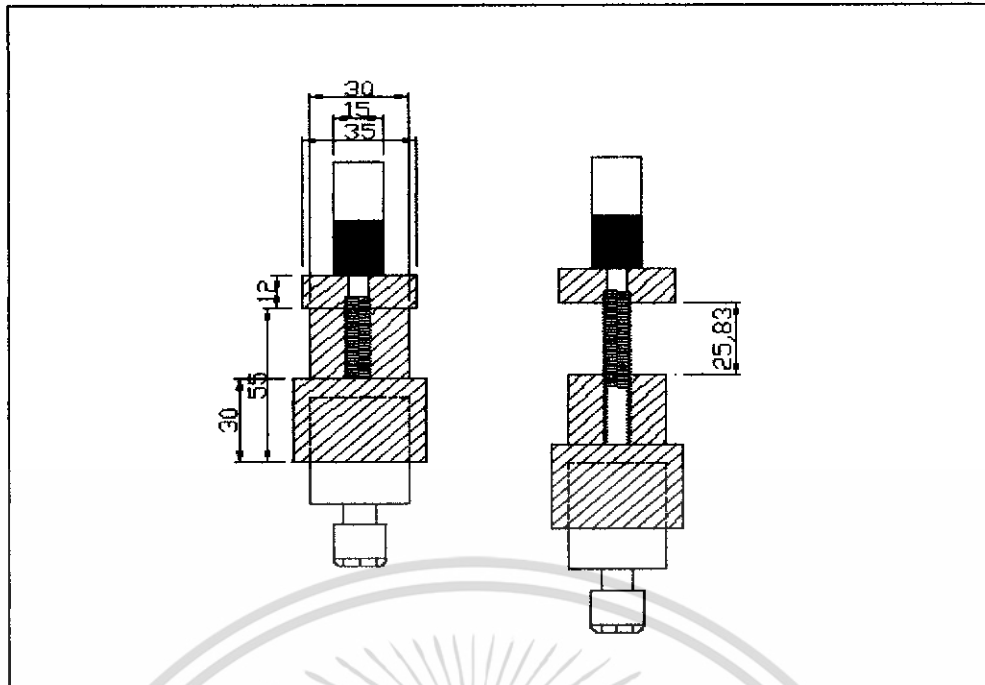


รูปที่3.11 รูปแสดงส่วนการเคลื่อนที่ที่สองจากของจริง

-โครงสร้างส่วนของหัวเจาะ ส่วนนี้จะแบ่งออกเป็น 2 ส่วน คือ

-ส่วนของตัวยึดมอเตอร์ ส่วนนี้จะประกอบขึ้นโดยใช้โลหะอลูมิเนียมที่มีน้ำหนักค่อนข้างเบาดังแสดงในรูปที่(3.4)มอเตอร์ตัวเจาะนี้ขนาด 6 V 12800 รอบ/นาที(rpm)

-ส่วนของตัวยึดมอเตอร์ส่วนนี้จะมีหน้าที่ยึดมอเตอร์ และยึดหัวเจาะติดกับแกนหมุนมอเตอร์ตัวที่ยึดติดอยู่นี้จะใช้ในการหมุนยึดออกไปของหัวเจาะ โดยหลักการของเลี้ยวนี้ติดตั้งแสดงในรูป(3.4)ทั้งหัวเจาะมีน้ำหนักขนาด 0.27 กิโลกรัม(kg)มอเตอร์ที่ใช้มีขนาด 12V 10 รอบ/นาที(rpm)

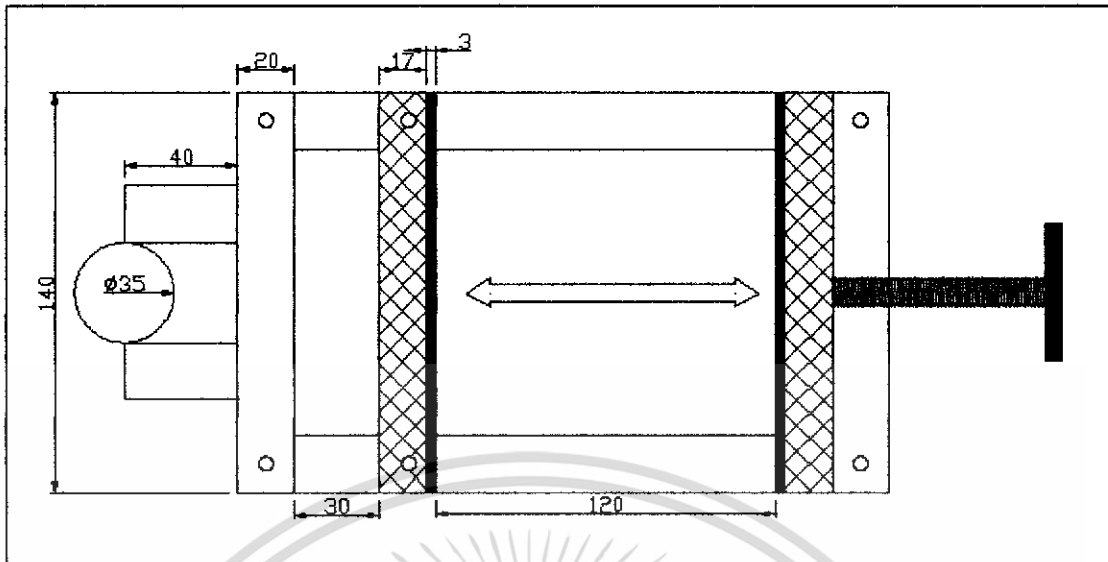


รูปที่ 3.12 แสดงภาพตัดของหัวเจาะ

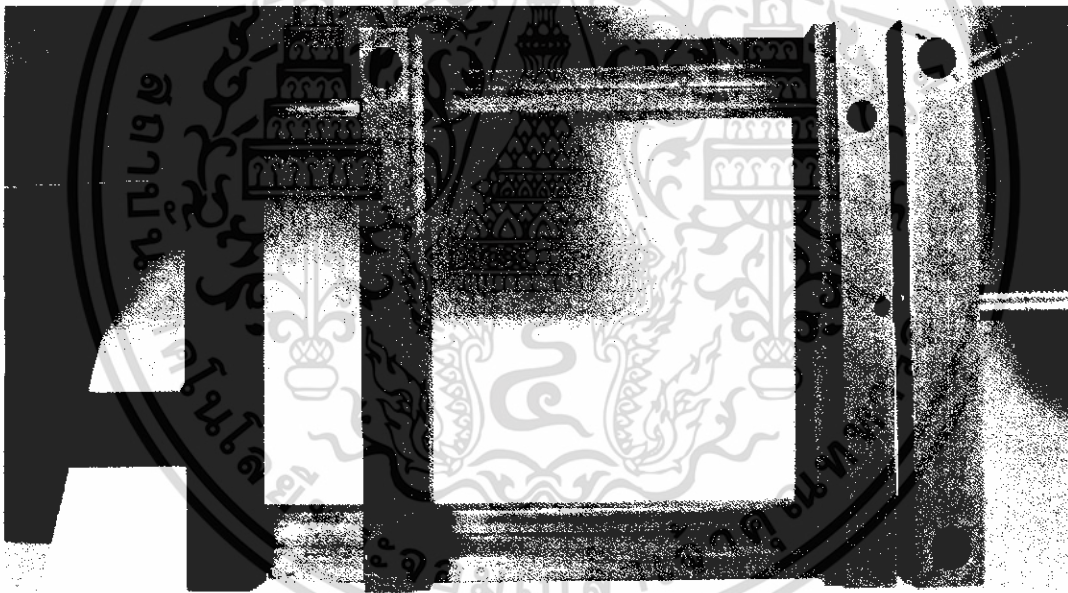


รูปที่ 3.13 รูปแสดงลักษณะของหัวเจาะ

ในส่วนของมอเตอร์ที่ใช้ขับเคลื่อนส่วนต่างๆ ของแขนกลนี้เป็นมอเตอร์ดีซี (DC Motor) ขนาดแรงดัน 12 โวลต์ ขนาดกระแส 2 แอมแปร์ 30 รอบ/นาที



รูปที่3.14 แสดงส่วนที่ใช้จับชิ้นงาน



รูปที่3.15 รูปแสดงแท่นจับชิ้นงาน

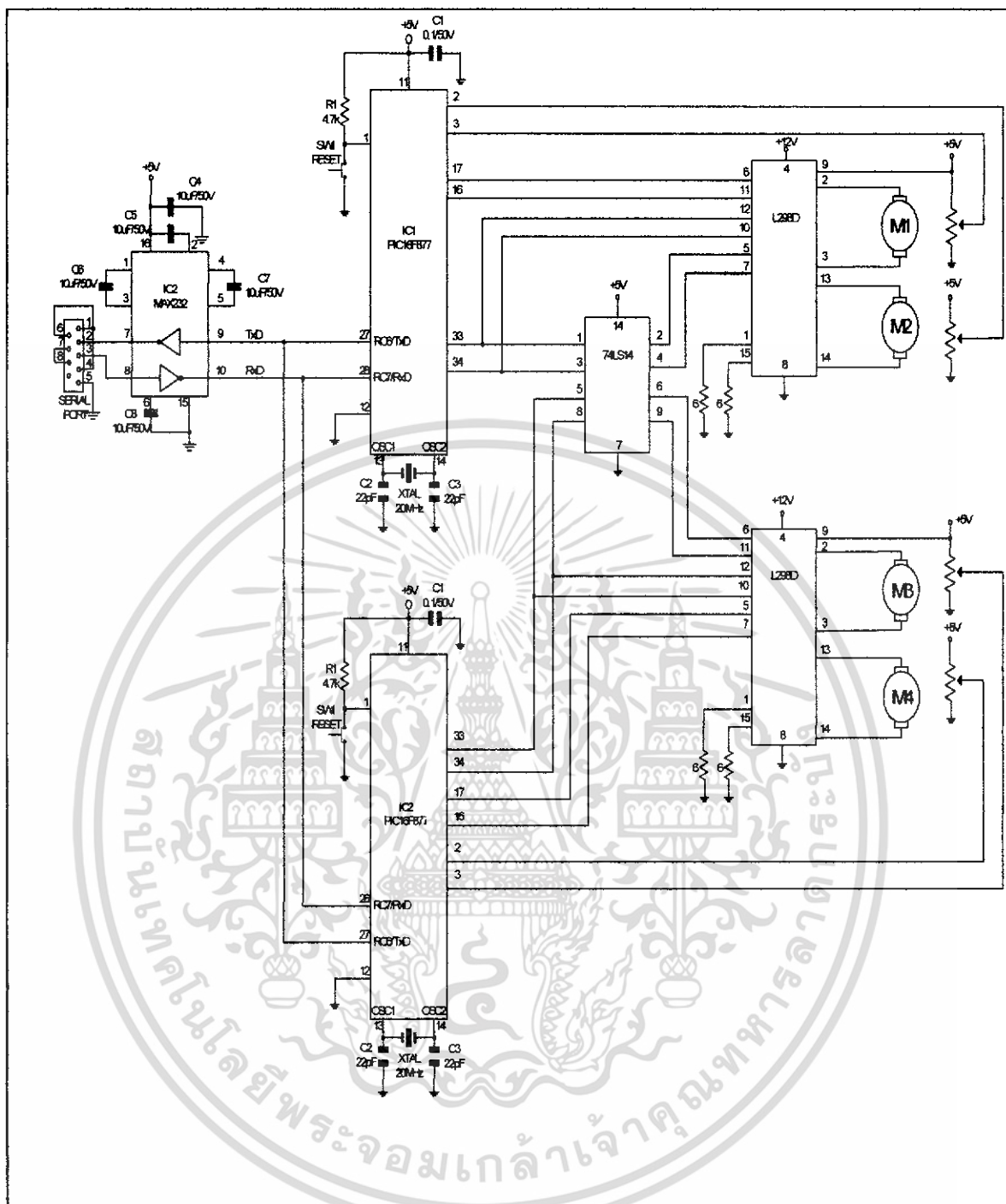
รูปดังกล่าวข้างต้นแสดงทิศทางการเคลื่อนที่ของแขนกลข้อต่างๆ ซึ่งแต่ละข้อต่อจะเคลื่อนที่ในแนวที่ต่างกัน เคลื่อนที่ได้องศาต่างกัน ข้อฐานหมุนสามารถเคลื่อนที่ได้ 360° ข้อต่อที่หนึ่ง เคลื่อนที่ได้ 120° ข้อต่อที่สองเคลื่อนที่ได้ 320° ส่วนข้อต่อที่สามเคลื่อนที่ได้ 360° แล้วหัวจะเคลื่อนที่ได้ 2.5 Cm

3.2 โครงสร้างและการออกแบบส่วนอิเล็กทรอนิกส์

3.2.1 วงจรการทำงานรวม

เป็นวงจรการทำงานที่ใช้สำหรับควบคุมการเคลื่อนที่ของแขนกลในส่วนของข้อต่อต่างๆ ซึ่งจะใช้หลักการ การเคลื่อนที่ไปในตำแหน่งที่ได้ระบุ โดยใช้การควบคุมแรงดันให้มอเตอร์หมุน และหยุดหมุนเมื่อได้ตำแหน่งที่ต้องการแล้ว การทำงานจะเริ่มจากการระบุตำแหน่งเป็นเลข ดิจิตอล ตัวคอนโทรเลอร์ก็จะรับค่าเข้าไปทางขา Rx ,Tx แล้วนำข้อมูลของจุดการเคลื่อนที่ ที่ได้ไปทำการประมวลผลและปฏิบัติตามกระบวนการของโปรแกรม ตัวคอนโทรเลอร์ก็จะสร้างสัญญาณพัลส์ (PWM) ขนาดต่างๆตามค่าของคู่อัซไคเคล เพื่อใช้สัญญาณพัลส์(PWM)นี้ไปขับมอเตอร์ให้หมุนเมื่อมอเตอร์หมุนเข้าใกล้จุดเซตพอยท์(set point)ขนาดของแรงดันที่เป็นสัญญาณตรีกนี้ก็จะค่อยๆน้อยลง โดยมีตัวบอกตำแหน่งการเคลื่อนที่ของแขนกลแต่ละข้อต่อ ซึ่งเป็นตัวป้อนกลับ(Feed back)แบบ ความต้านทานปรับค่าได้(10k)ต่อขานานแรงดัน 0-5 โวลท์ เพื่อไปแปลงเป็นเลขดิจิตอลเพื่อใช้เปรียบเทียบกับตำแหน่งเซตพอยท์ซึ่งค่าของ PWM จะเป็น 0 โวลท์ก็ต่อเมื่อตัวป้อนกลับ(Feed back)บอกตำแหน่งเดียวกันกับเซตพอยท์ที่ได้กำหนดไว้แล้วตอนเริ่มการทำงาน

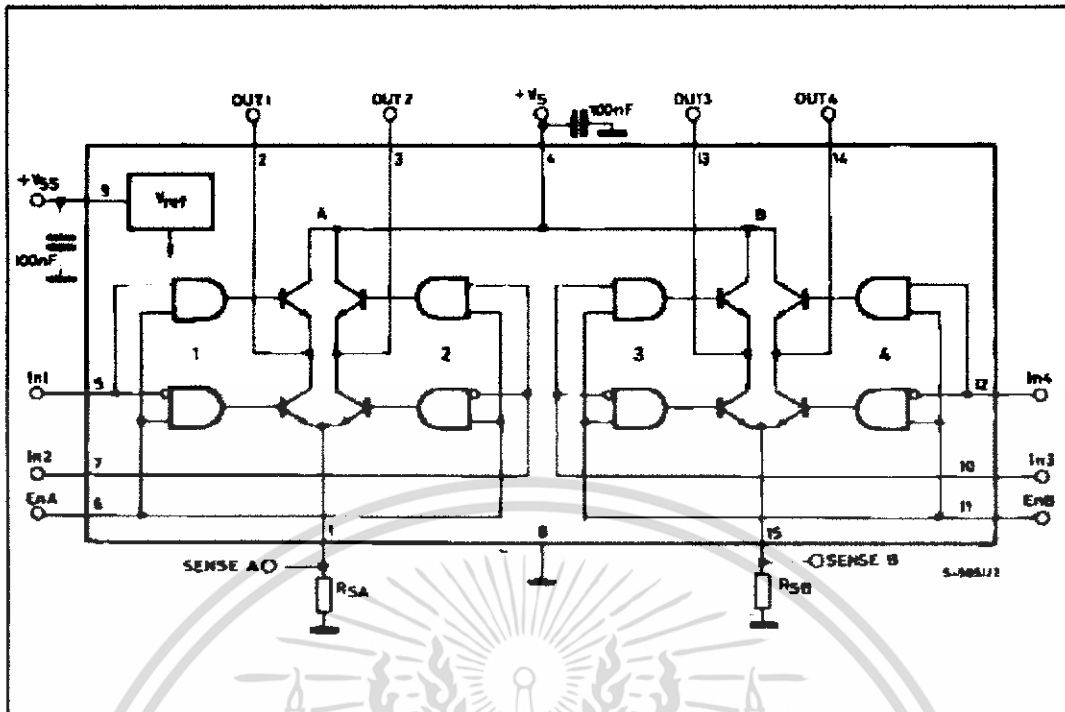
อนาล็อกทูดิจิตอล(Analog to Digital)ที่ใช้ในการแปลงสัญญาณจากตัวป้อนกลับ (Feedback) เป็น A-D ขนาด10บิตโดยรับสัญญาณอนาล็อกขนาด 0-5 โวลท์ตัวควบคุมใช้ฐานเวลาการทำงานขนาด 20 MHz มีความเร็วในการประมวลผล 50 nS ตัวคอนโทรเลอร์เป็นแบบ Pic เบอร์ 16F877A โดยเชื่อมต่อกับคอมพิวเตอร์ ผ่าน Max 232 เชื่อมต่อทางพอด์ทอนุกรม



รูปที่ 3.16 รูปแสดงวงจรรวม

3.2.2 วงจรขับเคลื่อนมอเตอร์

วงจรที่ใช้ขับเคลื่อนและควบคุมทิศทางมอเตอร์ในโครงงานนี้ออกแบบให้ใช้แบบ Full bridge ซึ่งเป็นไอซีเบอร์ (IC L298) (Push-Pull Four Channel Driver With Diode) เป็นไอซีขับมอเตอร์ที่มีวงจรการทำงานดังแสดงในรูปที่ 3.5



รูปที่ 3.17 รูปแสดงวงจรขับเคลื่อนมอเตอร์

ซึ่งสามารถทนแรงดันได้ 24 โวลท์ กระแสการทำงานจ่ายโหลดที่ 2 แอมแปร์ วงจรภายในเป็นวงจรลอจิก การทำงานของไอซีเบอร์ L298 คือ ในตัวไอซีตัวนี้สามารถจ่ายแรงดันให้กับมอเตอร์ได้ 2 ตัว ตัวที่ 1 คือ OUT 1,2 ตัวที่ 2 คือ OUT 3,4 ส่วนอินพุตมีอยู่ด้วยกัน 6 Channel เอาท์พุต 4 Channel ซึ่งจะกล่าวถึงเพียง channel เดียว ในการควบคุมทิศทางหมุนของมอเตอร์จะควบคุมด้วย In1, In2 โดยถ้าต้องการให้มอเตอร์หมุนทิสทางทวนเข็มนาฬิกาต้องกำหนดให้ลอจิก In1 “0” และ In2 “1” ส่วนกำหนดตามเข็มนาฬิกาต้องกำหนดให้ลอจิก In1 “1” และ In2 “0” แล้วขา Enable A จะเป็นตัวบ่งบอกถึงสภาวะการพร้อมจะทำงานเมื่อ Enable A “1” แรงดันจะร่อจ่ายให้มอเตอร์ โดยร่อสภาวะทิสทางการหมุนจากลอจิก In1 ,In2 อีกประการหนึ่งคือการกำหนดค่ากระแสที่ไหลผ่านไอซี(Ic) ซึ่งจะกำหนดโดยขา Rsa เป็นความต้านทานที่นำมาใช้ต่อเพื่อกำหนดกระแสซึ่งจะได้ดังสมการนี้

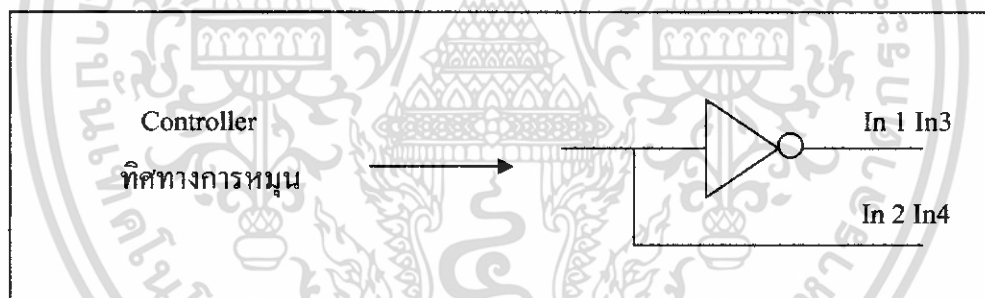
$$\begin{aligned}
 R &= V/I \\
 &= 12 \text{ V} / 2\text{A} \\
 &= 6 \Omega
 \end{aligned}$$

I = กระแสที่ไหลผ่าน

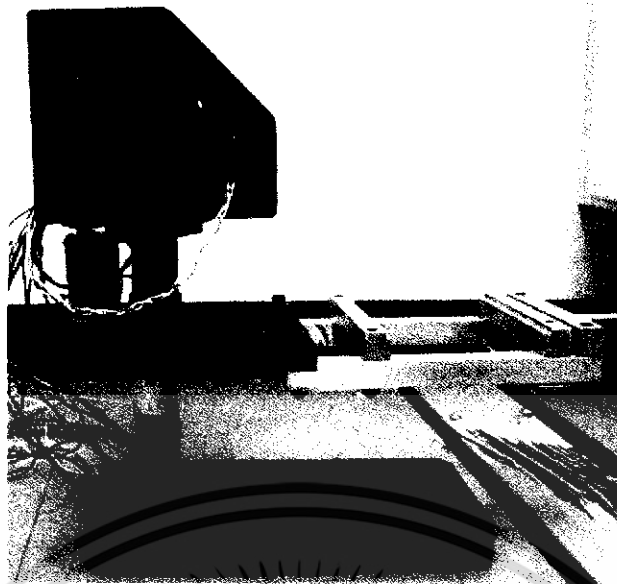
V = แรงดันที่จ่ายให้มอเตอร์(แรงดันที่ขา Vs)

R = ความต้านทานที่ต้องการ

สำหรับที่ขาของ En A, En B จะต่อร่วมกับตัวคอนโทรลเลอร์ ซึ่งจะรับสัญญาณมาเป็นสัญญาณ PWM ที่ควิตซ์เฟรคเอนซ์ต่างๆ เมื่อ En เป็นสัญญาณ PWM แรงดันที่จ่ายให้กับมอเตอร์จึงเป็นสัญญาณชนิด PWM ที่มีขนาดแรงดัน $p-p = 12\text{ v}$ ส่วนทิศทางการหมุนนั้นขึ้นอยู่กับ In1, In2, In3 และ In4 ซึ่งขาของ In1, In2, In3 และ In4 เป็นขาที่ระบุถึงสภาวะทิศทางการหมุนของมอเตอร์โดยที่ In1, In2 ทำงานคู่กันส่วน In3 และ In4 ก็ทำงานคู่กันโดยตำแหน่งของการควบคุมดังได้กล่าวมาข้างต้นแล้วจะเห็นว่าทำให้เกิดการควบคุมที่ยุ่งยาก เพราะต้องใช้สัญญาณจากตัวคอนโทรลเลอร์มาทริกถึงสองเส้น เพื่อลดจำนวนให้วงจรมีจำนวนสายสัญญาณลดน้อยลงให้เหลือเพียงสองเส้นสัญญาณ คือ เส้นสัญญาณ PWM และเส้นสัญญาณควบคุมทิศทางการหมุน โดยการผ่านลอจิก เบอร์ 74LS14P เป็นลอจิก not gat ซึ่งการทำงานแสดงดังรูปที่(3.6)



รูปที่3.18 รูปแสดงการต่อลอจิกช่วย



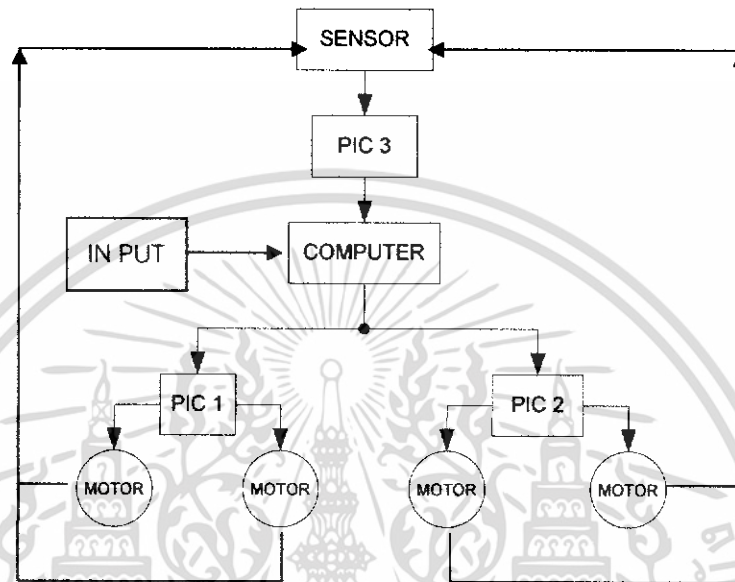
รูปที่ 3.19 แสดงส่วนของฐานแขนกล



รูปที่ 3.20 รูปแสดงการติดตั้งมอเตอร์

3.3 ส่วนของโครงสร้างโปรแกรม

3.3.1 การออกแบบระบบควบคุม

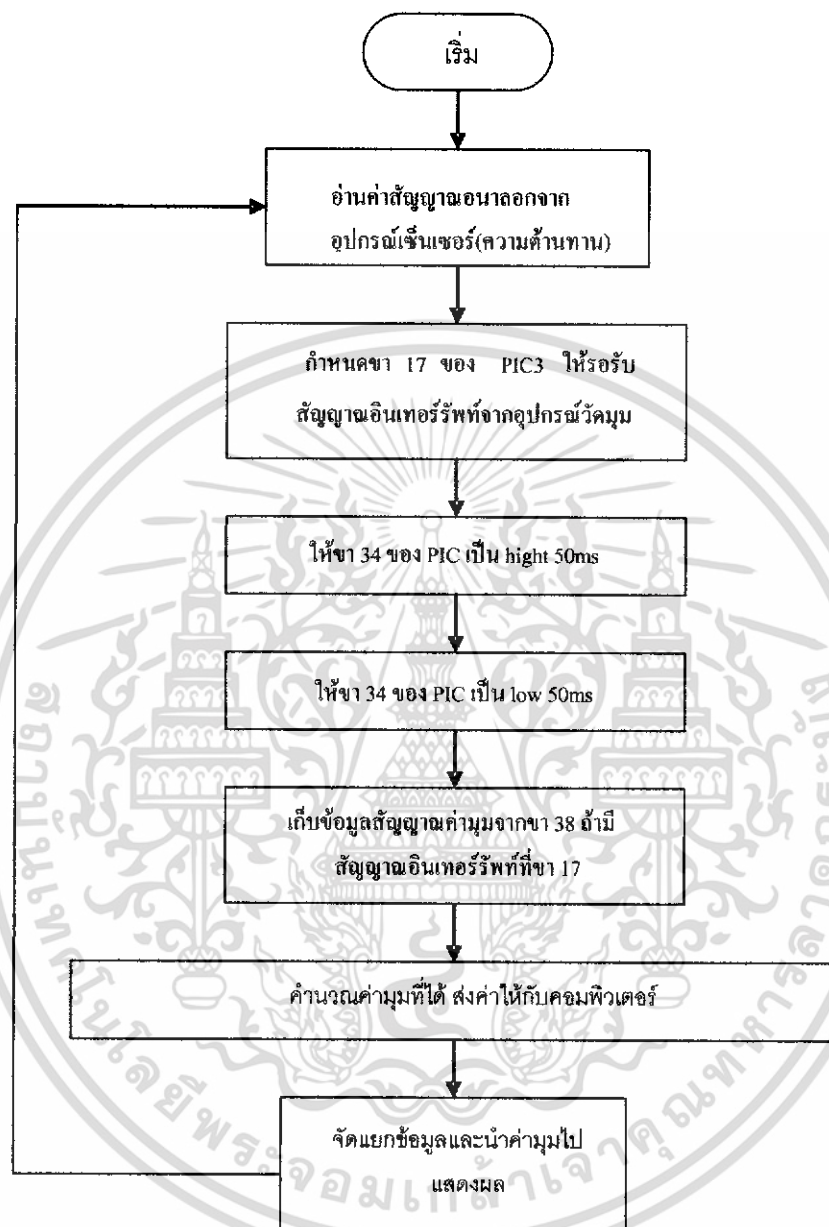


รูปที่ 3.21 แผนผังแสดงการออกแบบระบบควบคุมแบบปิด

ในการออกแบบระบบควบคุมแบบปิด จะทำโดยใช้คอนโทรลเลอร์ทั้งหมด 3 ตัว จากรูปที่ 3.21 จะแสดงการออกแบบโดย PIC 3 จะรับข้อมูลจากเซนเซอร์ทั้ง 2 ชุด คือค่ามุมแล้วจึงนำข้อมูลส่งให้กับคอมพิวเตอร์ เพื่อประมวลผล โดยอาศัยตัวควบคุมแบบพีไอดี เป็นระบบอ้างอิง และสร้างสัญญาณควบคุม ออกมาส่งต่อให้กับ PIC 1 และ PIC 2 เพื่อใช้ในการสร้างสัญญาณพัลส์จ่ายให้แก่มอเตอร์ต่อไป

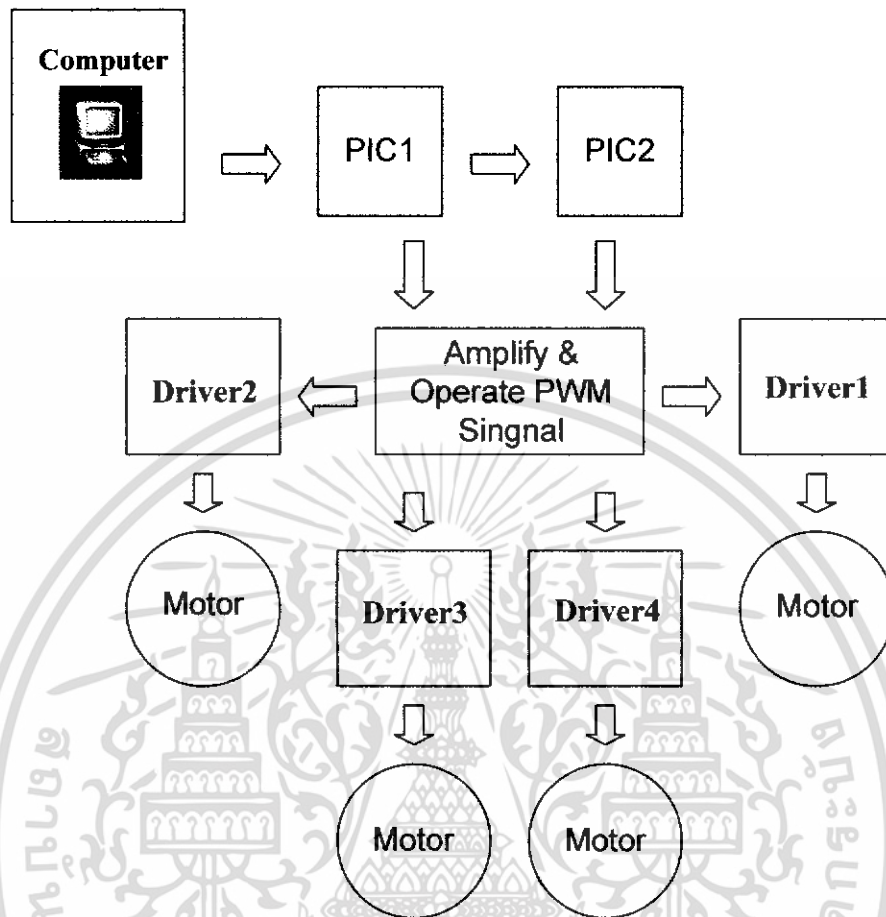
เนื่องจากตัวคอนโทรลเลอร์ PIC 3 ตัวสามารถจ่ายสัญญาณพัลส์ได้เพียง 2 ช่องเท่านั้น ดังนั้นจึงจำเป็นต้องใช้ตัวคอนโทรลเลอร์ 2 ตัว เพื่อจ่ายสัญญาณพัลส์ 4 ช่อง ให้กับมอเตอร์ทั้ง 4 ตัว ซึ่งต้องใช้สัญญาณ PWM ทั้งหมด 4 ช่อง โดยในส่วนของโปรแกรมที่ออกแบบให้กับตัวคอนโทรลเลอร์ทั้ง 2 ตัวได้แสดงรายละเอียดไว้แล้วในภาคผนวก

3.3.2 แผนภาพกระบวนการรับค่าจากเซนเซอร์



รูปที่ 3.22 แผนภาพแสดงขั้นตอนการทำงานในการรับค่าจากเซนเซอร์

3.3.3 กระบวนการคำนวณและส่งสัญญาณขับเคลื่อนมอเตอร์



รูปที่ 3.23 แผนภาพแสดงวิธีการส่งสัญญาณและการจ่ายพัลส์ให้มอเตอร์

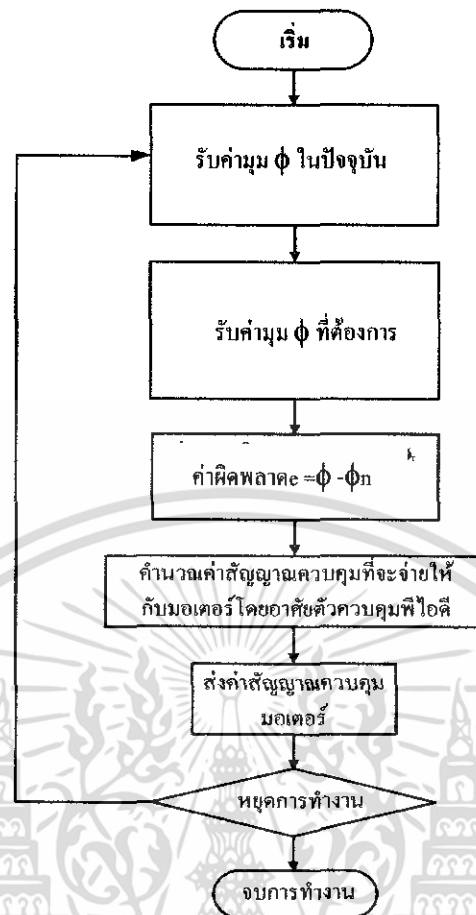
คำอธิบายประกอบแผนภาพ

ขั้นตอนการทำงานแสดงได้ดังรูปที่ 3.23 โดยผู้ใช้จะติดต่อกับวงจรขับเคลื่อนมอเตอร์โดยผ่านทางคอมพิวเตอร์ ซึ่งผู้ใช้จะส่งข้อมูลซึ่งเป็นค่าความถี่พัลส์ (Duty cycle) และทิศทางของมอเตอร์แต่ละตัวโดยใช้โปรแกรม Visual Basic ซึ่งถูกออกแบบมาให้ใช้สำหรับติดต่อกับผู้ใช้งาน โดยตรง จากนั้นคอมพิวเตอร์จะส่งข้อมูลให้แก่ไมโครคอนโทรลเลอร์ที่เป็นมาสเตอร์หรือตัวหลักคือ PIC1 โดย PIC1 จะทำการประมวลผลข้อมูลที่ได้ และจัดแยกค่าข้อมูลที่ส่งมานั้นเป็นค่าความถี่พัลส์และทิศทางของมอเตอร์ตัวใด จากนั้นจะทำการส่งข้อมูลที่แยกแล้วให้แก่ไมโครคอนโทรลเลอร์ที่เป็นสเลฟคือ PIC2 เพื่อให้ PIC2 นำไปใช้ในการสร้างสัญญาณพัลส์เพื่อส่งต่อให้ทำการกำหนดทิศทางของมอเตอร์ต่อไป ซึ่งสัญญาณที่ไมโครคอนโทรลเลอร์ทั้งสองตัวส่งมานั้น ประกอบด้วยสัญญาณพัลส์ จำนวน 4 ช่องสัญญาณที่มีค่าความถี่พัลส์ตามที่ผู้ใช้งานกำหนดเพื่อจ่ายให้แก่มอเตอร์แต่ละตัว

และสัญญาณอินเนเบิล (Enable signal) ซึ่งเป็นลอจิก “1” และ “0” เพื่อบอกว่ามอเตอร์ตัวใดจะหมุนทางใดอีกจำนวน 8 ช่องสัญญาณ โดยตัวไมโครคอนโทรลเลอร์ที่เลือกใช้คือ PIC 16F877A เนื่องจากสามารถสร้างสัญญาณพัลส์ได้ ซึ่งแต่ละตัวนั้นจะสามารถสร้างสัญญาณพัลส์ได้ 2 ช่องสัญญาณ ดังนั้นจึงต้องใช้ตัวไมโครคอนโทรลเลอร์ 2 ตัวในการสร้างสัญญาณพัลส์ 4 ช่องสัญญาณ และทำการเชื่อมต่อเพื่อส่งข้อมูลระหว่างกัน จากนั้นข้อมูลที่ี้จะถูกส่งไปยังวงจรถ่ายและจัดการสัญญาณพัลส์ซึ่งจะทำหน้าที่ขยายสัญญาณพัลส์ที่รับมาจากไมโครคอนโทรลเลอร์แล้วส่งให้แก่วงจรถับมอเตอร์ และนอกจากนี้ยังเป็นวงจรที่กำหนดทิศทางของการหมุนของมอเตอร์แต่ละตัวด้วย โดยวงจรมีจะรับสัญญาณพัลส์มารอไว้ และนำสัญญาณอินเนเบิลมาเป็นตัวกำหนดว่าสัญญาณพัลส์ที่รับมานั้น จะต้องจ่ายให้แก่ฝั่งใดของวงจรถับมอเตอร์เพื่อให้หมุนไปในทิศทางที่ผู้ใช้ต้องการ จากนั้นจะนำสัญญาณพัลส์ที่ี้ส่งเข้าไปยังมอเตอร์แต่ละตัวตามที่ผู้ใช้งานได้กำหนดไว้

3.3.4 การรับค่าและคำนวณค่าพัลส์ที่จะต้องจ่ายออกไปด้วยโปรแกรม Visual Basic

จากรูปที่ 3.24 แสดงขั้นตอนการคำนวณค่าสัญญาณพัลส์ที่จะจ่ายให้แก่มอเตอร์เริ่มต้นจากการรับค่ามุมที่ต้องการ และ ค่ามุมปัจจุบันซึ่งรับมาจาก PIC ที่รับค่าจากอุปกรณ์วัดดังกล่าวจากนั้นจึงนำมาเปรียบเทียบเพื่อคำนวณค่าความผิดพลาด และนำมาหาสัญญาณควบคุมมอเตอร์โดยอาศัยตัวควบคุมแบบ PID โดยใช้คาบเวลาของการสุ่มสัญญาณ (Sampling time) ซึ่งได้จากการทดลองวัดเท่ากับ 50 ms. จากนั้นนำค่าที่คำนวณได้ส่งไปยังตัวคอนโทรลเลอร์ PIC1 และ PIC2 คอนโทรลเลอร์ทั้งสองตัวจะส่งสัญญาณ PWM ไปยังวงจรถ่ายสัญญาณ PWM ในรูปที่ 3.23 และส่งสัญญาณ PWM ไปยังวงจรถับมอเตอร์



รูปที่ 3.24 แผนภาพแสดงขั้นตอนการคำนวณค่าสัญญาณพัลส์ที่จะจ่ายให้แก่มอเตอร์

บทที่ 4

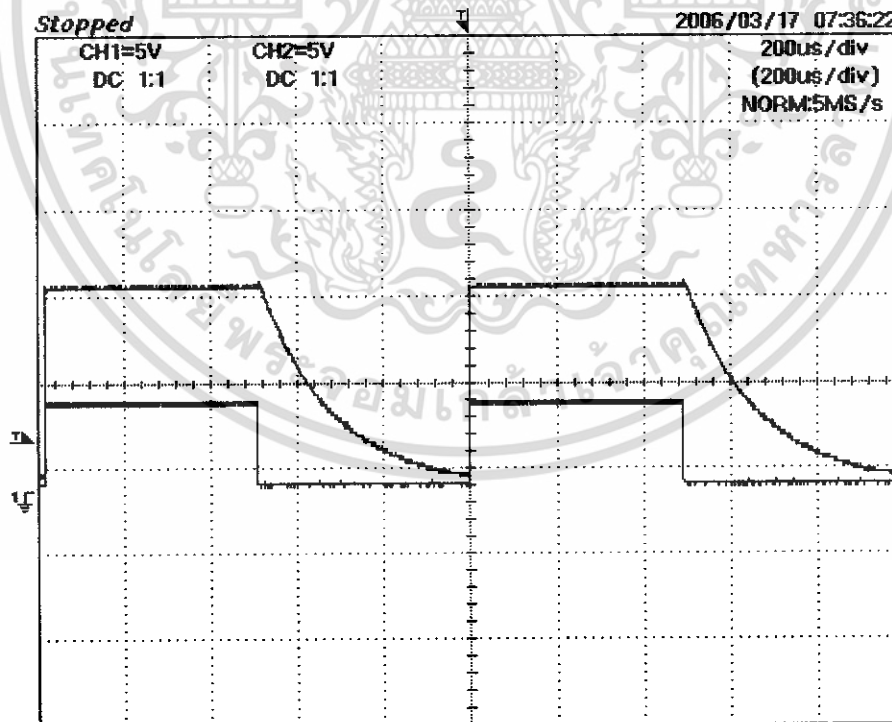
การทดลองและผลการทดลอง

- การทดลองการเกิด PWM ของชุด Driver
- การทำงานของ Drive มอเตอร์
- การเคลื่อนที่ของแขนกล
 - คิวเวลาที่ตีที่สุด
 - ค่าเวลาที่ใช้ในการทำงาน

4.1 การทดลองเพื่อทดสอบการกำเนิด PWM ของวงจร Driver

การทดลองนี้ทดสอบว่าชุดDriverสามารถทำงานได้ทันความถี่ที่จ่ายมาจากคอนโทรลเลอร์และจะมีค่าของerrorตีคมาหรือเปล่าและอีกประการหนึ่งคือชุด Driver สามารถทำงานได้ตามที่ออกแบบมาหรือไม่ซึ่งการทดลองนั้นทำการทดลองที่ความถี่ 1 KHz และ 10KHz ตามลำดับ

ผลการทดลอง



รูปที่ 4.1 รูปแสดงสัญญาณที่ได้จากการกระตุ้นของ PWM

สัญญาณทริก(1) P-P = 5V

Rms = 3.65V

Avg = 2.77V

Freq = 1.064KHz

Duty = 50%

สัญญาณที่ได้(2) P-P = 11.4V

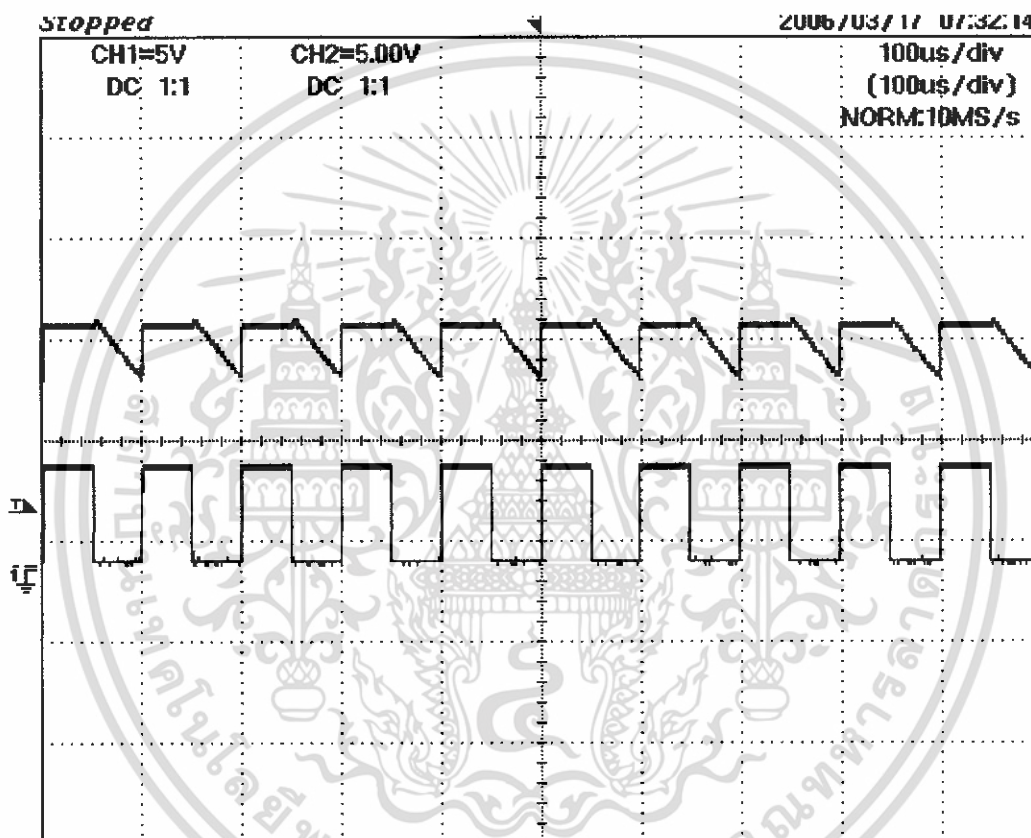
Rms = 9.275V

Avg = 8.259V

Freq = 1.064KHz

Duty = 62.5%

จากการทดลอง โดยใช้ความถี่ที่ 10 KHz เป็นสัญญาณทริก Driver จากนั้นก็จะได้ผลการทดลองออกมาดังแสดง



รูปที่ 4.2 รูปแสดงสัญญาณที่ได้จากการกระตุ้นของ PWM

สัญญาณทริก(1) P-P = 5.000V

Rms = 3.630V

Avg = 2.770V

Freq = 10.100KHz

Duty = 50%

สัญญาณที่ได้(2) P-P = 3.000V

Rms = 11.350V

Avg = 11.329V

Freq = 10.100KHz

Duty = 79%

จากผลลัพธ์ที่ได้เมื่อดูจากกราฟรูปสัญญาณจะเห็นได้ว่า สัญญาณที่จ่ายให้กับตัวมอเตอร์จะอยู่ในรูปของสัญญาณ PWM ที่มีขนาดความถี่เท่ากับตัวสัญญาณของ PWM ที่จ่ายมาจากตัว

คอนโทรลเลอร์แต่สัญญาณที่ได้จะมีค่าของการดีเลย์ ที่เกิดจากการที่ความเร็วในการทำงานของ Driver มีค่าต่ำกว่าสัญญาณกำเนิด

4.2 การทดลองเพื่อทดสอบการทำงานของชุด Driver

การทดลองนี้เพื่อทดสอบการทำงานของ Driver หาค่าสถานะต่างๆโดยมาเขียนเป็นตารางความจริง

ผลการทดลอง

In1	In2	EnA	Out1	Out2
0	1	1	1	0
1	0	1	0	1
0	0	1	0	0
0	1	0	0	0
0	0	0	0	0

ตารางที่ 4.1 ตารางแสดงสถานะการทำงานของ Driver

จากการทดลองจะได้ตารางความจริงออกมาทำให้ทราบว่า สถานะการทำงานทั้งหมดจะขึ้นอยู่กับสถานะของตัว EnA เมื่อ EnA มีสถานะเป็น “0” ก็จะได้ Out มีค่าเป็นศูนย์โดยจะไม่สนใจสถานะของขาอื่นๆด้วยลักษณะดังกล่าว จึงประยุกต์นำไปใช้ในโครงงานนี้ดังที่ได้กล่าวมา สิ่งที่ควรระวังอีกประการถ้าดูจากตารางความจริงคือ สถานะของ In1 และ In2 ถ้าเกิดทั้งสองขานี้มีสถานะเป็น “1” จะทำให้เกิดการเสียหายขึ้น กับตัวอุปกรณ์ ซึ่งการทดลองดังกล่าวได้นำชุด Driver เพียงชุดเดียวมาทำการทดลอง

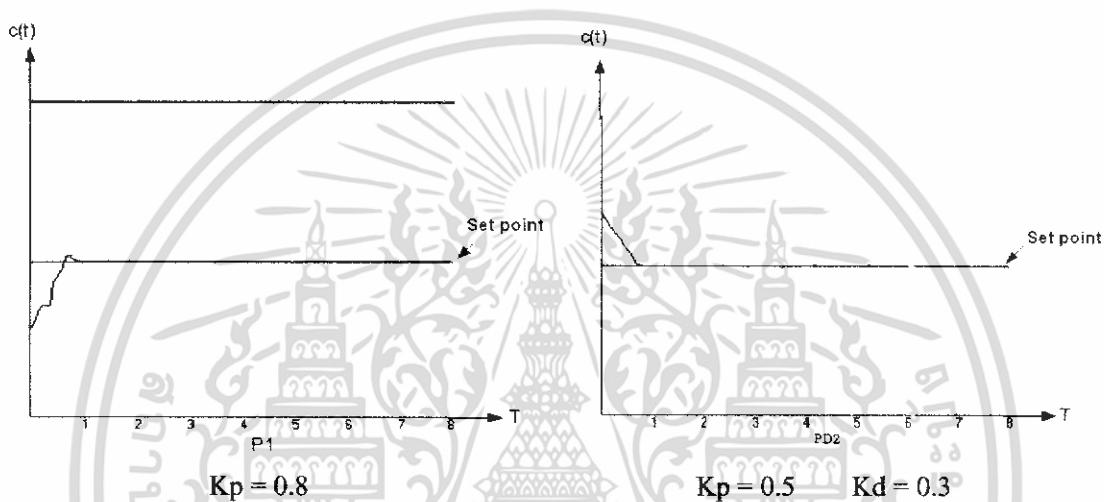
4.3 การทดลองแขนกล

การทดลองนี้ทำการทดลองเพื่อที่จะหาตัวควบคุมที่มีความเหมาะสมที่สุด ในแต่ละข้อต่อของแขนกล ซึ่งจะมีผลต่อการนำไปใช้ควบคุมแขนกล

4.3.1 เพื่อหาค่าของตัวควบคุมที่ดีที่สุด

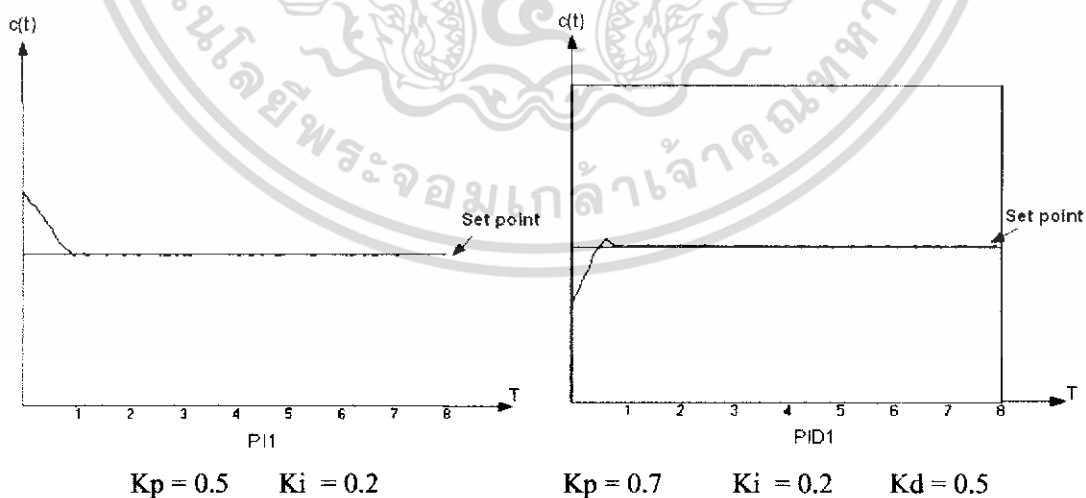
ผลการทดลอง

ทดลองข้อต่อของแขนส่วนที่หนึ่ง



รูปที่ 4.3.1-ก แสดงผลตอบสนองของ P

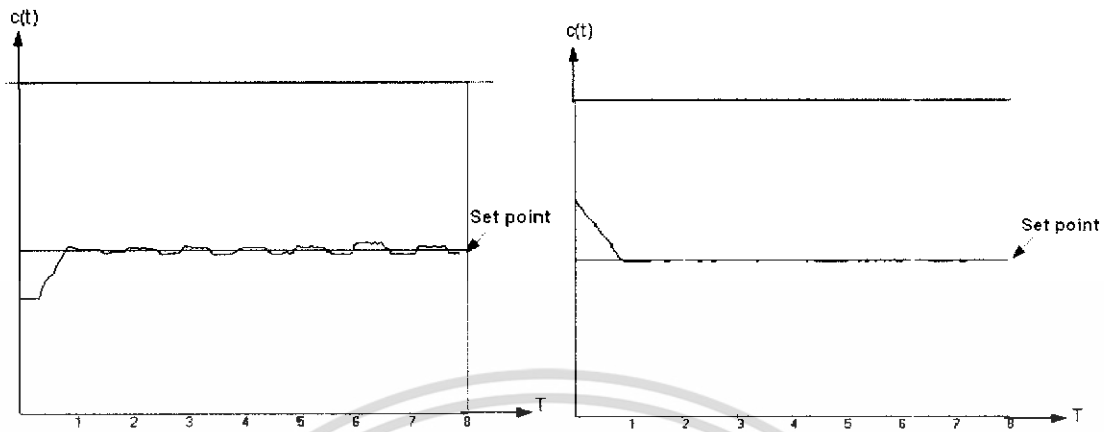
รูปที่ 4.3.1-ข แสดงผลตอบสนองของ PD



รูปที่ 4.3.1-ค แสดงผลตอบสนองของ PI

รูปที่ 4.3.1-ง แสดงผลตอบสนองของ PID

ทดลองข้อต่อส่วนที่สอง

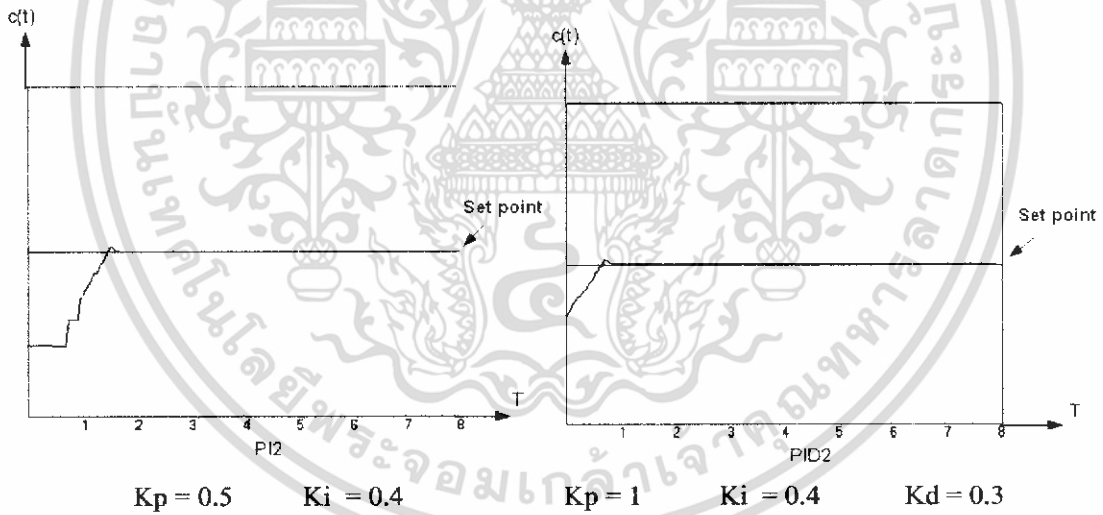


P2
 $K_p = 1.2$

PD2
 $K_p = 0.8$ $K_d = 0.5$

รูปที่ 4.3.2-ก แสดงผลตอบสนองของ P

รูปที่ 4.3.2-ข แสดงผลตอบสนองของ PD



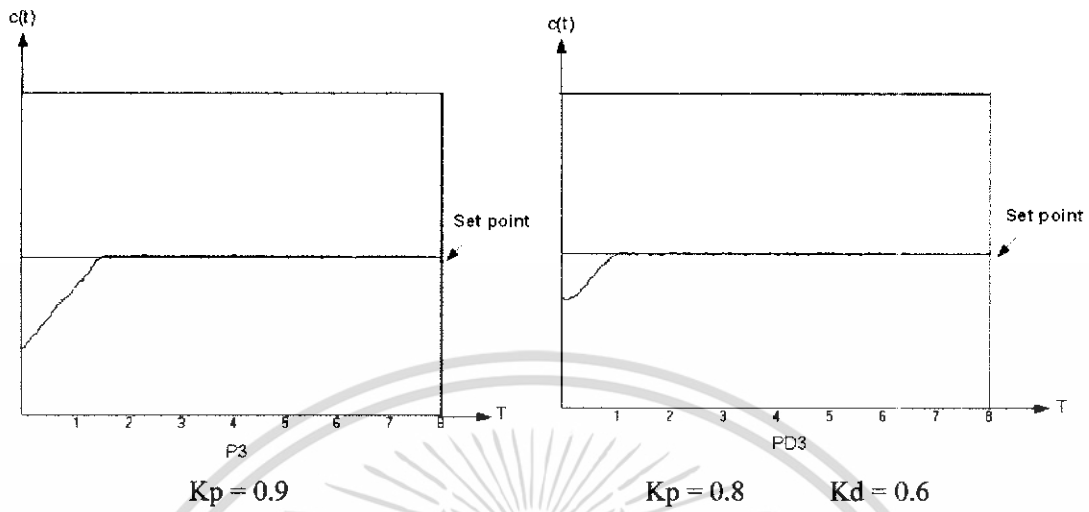
PI2
 $K_p = 0.5$ $K_i = 0.4$

PID2
 $K_p = 1$ $K_i = 0.4$ $K_d = 0.3$

รูปที่ 4.3.2-ค แสดงผลตอบสนองของ PI

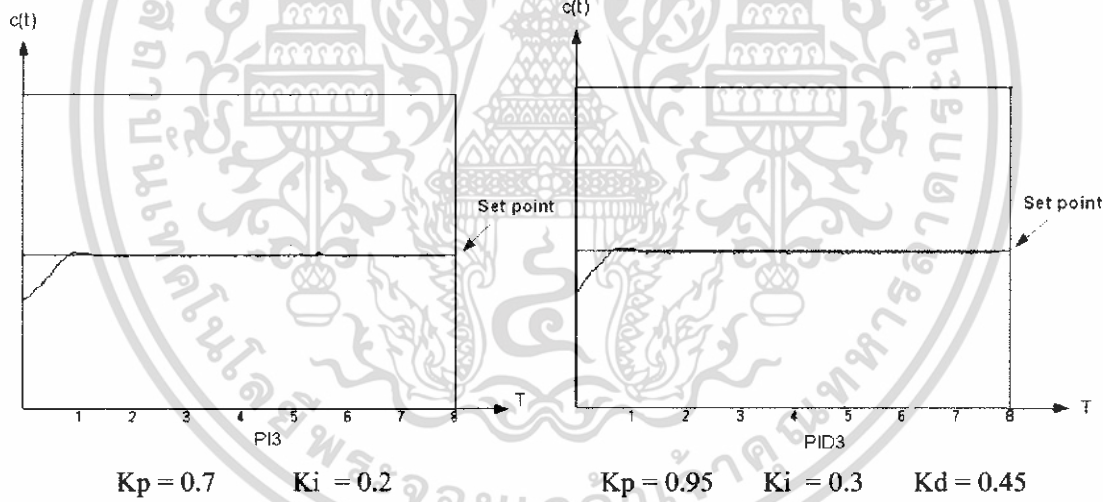
รูปที่ 4.3.2-ง แสดงผลตอบสนองของ PID

ทดลองข้อต่อส่วนฐานหมุน



รูปที่ 4.3.3-ก แสดงผลตอบสนองของ P

รูปที่ 4.3.3-ข แสดงผลตอบสนองของ PD

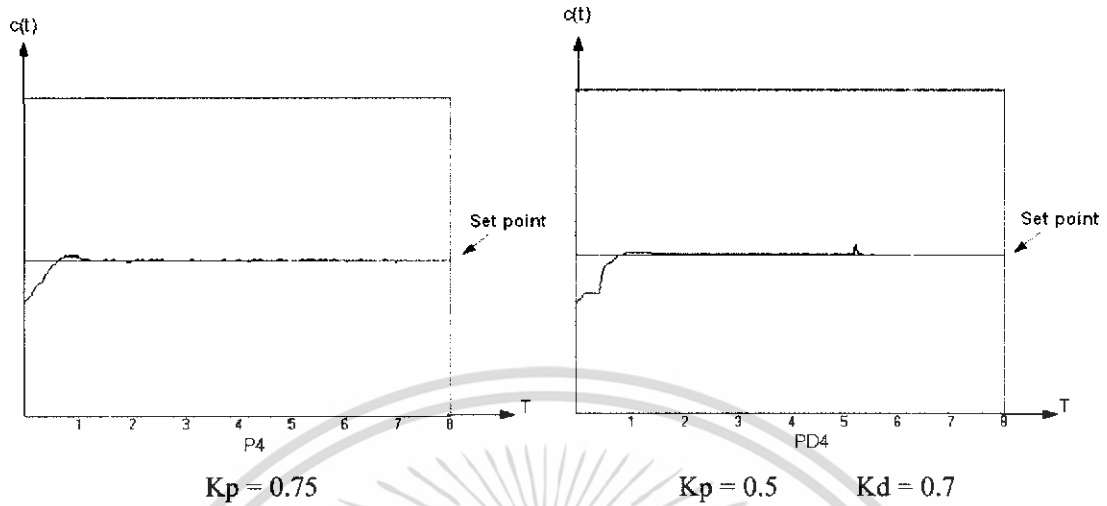


รูปที่ 4.3.3-ค แสดงผลตอบสนองของ PI

รูปที่ 4.3.3-ง แสดงผลตอบสนองของ PID

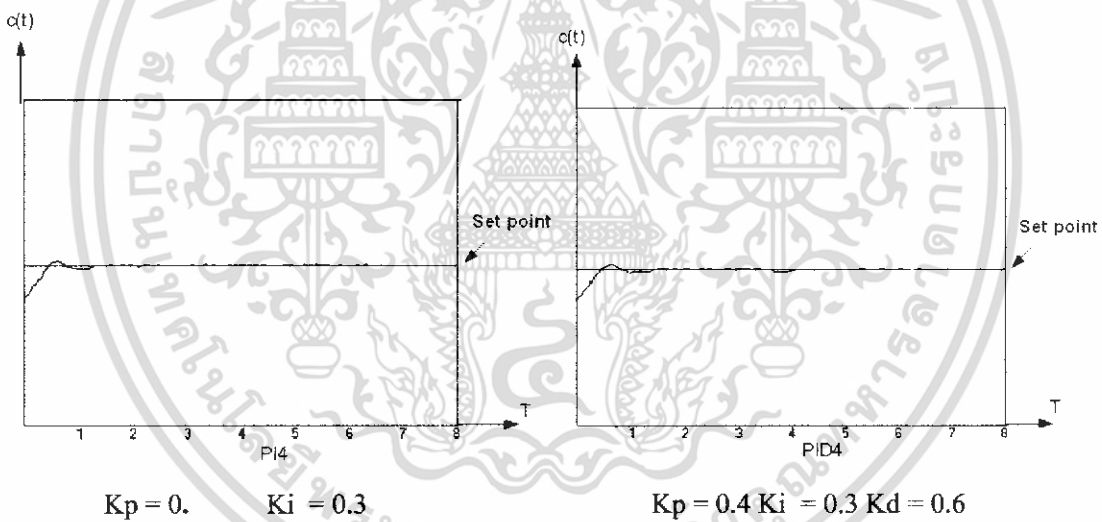
-

-ทดลองข้อต่อส่วนหัวเจาะ



รูปที่ 4.3.4-ก แสดงผลตอบสนองของ P

รูปที่ 4.3.4-ข แสดงผลตอบสนองของ PD



รูปที่ 4.3.4-ค แสดงผลตอบสนองของ PI

รูปที่ 4.3.4-ง แสดงผลตอบสนองของ PID

จากค่าของกราฟที่ได้ทำให้เราทราบถึงการเลือกใช้ตัวควบคุมที่เหมาะสมที่สุด โดยเราพิจารณาจากผลตอบสนองซึ่งแต่ละข้อต่อจะมีผลตอบสนองที่ต่างกันไป ผลตอบสนองที่ดีก็คือจะต้องเข้าสู่จุด Steady state ได้เร็ว

- P จะทำให้แกนเคลื่อนที่ได้แต่มีค่า error ในระบบทำให้แกนสั่น
- I คือการเก็บประจุการอินทิเกรตทำให้แกนเคลื่อนที่ได้เร็ว Steady state error ลดลง
- D จะทำให้ระบบมีความเร็วขึ้นแต่แกนจะสั่น

จากการทดลองจึงเลือกตัวควบคุมให้แต่ละข้อต่อ

- ข้อต่อส่วนของฐานหมุนเป็นแบบ PID
- ข้อต่อส่วนที่หนึ่งเป็นแบบ PD
- ข้อต่อส่วนที่สอง PID
- ข้อต่อส่วนหัวเจาะ PI

4.3.2 การเคลื่อนที่ของตัวแขนกล

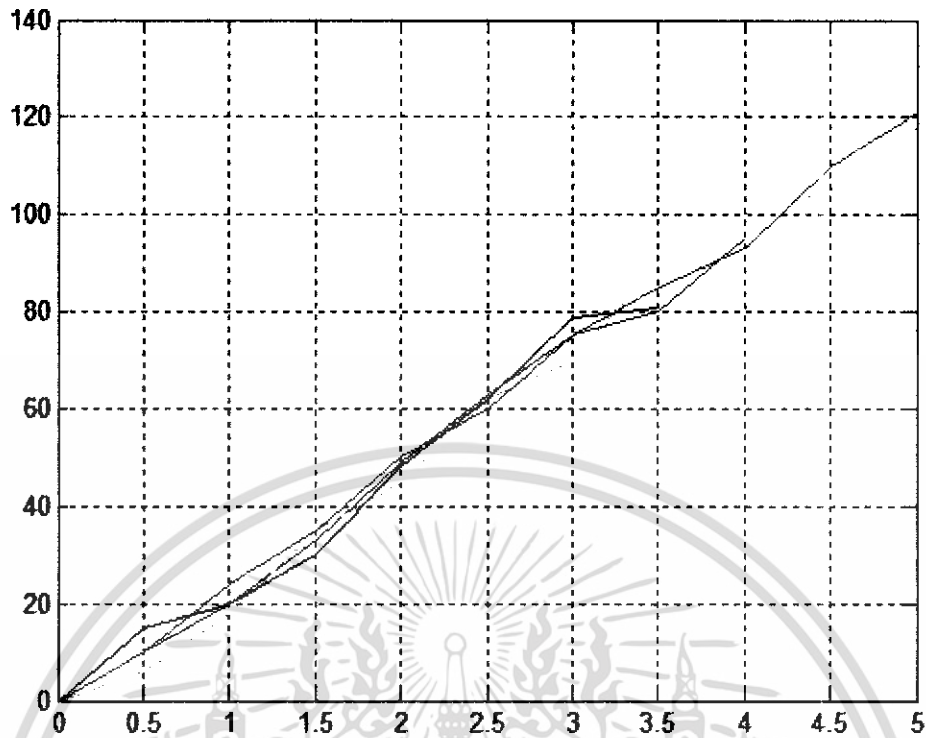
ผลการทดลอง

การทดลองดูการเคลื่อนที่ของตัวแขนกลว่าสามารถเคลื่อนที่ได้ตรงตามตำแหน่งที่ต้องการที่ค่าของ A-D แสดงดังตาราง

เซ็นเซอร์(V)	ตำแหน่ง	ข้อต่อที่หนึ่ง	ข้อต่อที่สอง	หัวเจาะ	ฐานหมุน
0	0	0°	0°	0°	0°
0.5	101.15	10°	6°	10°	15°
1	202.3	20°	18°	24°	20°
1.5	303.45	33°	35°	35°	30°
2	404.6	49°	45°	50°	48°
2.5	505.75	63°	62°	60°	62°
3	606.9	75°	70°	75°	77°
3.5	708.05	80°	90°	85°	81°
4	809.2	95°	100°	93°	X
4.5	910.35	X	105°	110°	X
5	1011.5	X	118°	121°	X

ตารางที่ 4.2 ตารางทดสอบการเคลื่อนที่ของแขนกล

ค่าจากตารางที่ได้เป็นค่าองศาที่แกนแต่ละข้อต่อเคลื่อนที่ไปในตำแหน่งที่ระบุ โดยค่าที่วัดได้เป็นการวัดแบบใช้ไม้บรรทัดวัดจึงอาจจะมีค่าผิดพลาดที่เกิดจากการวัด และการดูของสายตา



รูปที่ 4.3 กราฟแสดงผลการเคลื่อนที่ของแขนกลข้อต่อๆ

- ข้อต่อส่วนที่ 1
- - - ข้อต่อส่วนที่ 2
- ส่วนหัวเจาะ
- · - · - ส่วนฐานหมุน

จากรูปด้านบนที่ได้ทดลองผลตอบสนองที่ได้จากตัวควบคุมแบบต่าง โดยดูได้จากกราฟว่าตัวควบคุมแบบไหนให้ผลตอบสนองดีที่สุด จากกราฟที่ได้จะเห็นได้ว่าตัวควบคุมแบบ PD จะทำให้ระบบ

มีความเร็วมากค่า steady state error สูงเราจึงนำเอาตัวควบคุมแบบ I เข้ามาช่วยลดค่าของ steady state error ซึ่งดูได้จากกราฟที่ทำการทดลองในแต่ละส่วนต่างๆ ของแขนกล

บทที่ 5

วิจารณ์และสรุปผลการทดลอง

5.1 สรุปผลการดำเนินงาน

จากการศึกษา ทดลองและจัดสร้าง โครงการนี้แบ่งได้เป็น 3 ส่วน คือ ส่วน ส่วนโครงสร้างและการออกแบบทางกล ส่วนโครงสร้างและการออกแบบทางวงจร อิเล็กทรอนิกส์ และส่วน โปรแกรมควบคุมการทำงาน รูปแบบการทำงานของแขนกล

ในการดำเนินงานทางด้านกลไกของตัวแขนกลเนื่องด้วยตัวแขนกลทำจากวัสดุที่มีความหนักมากจึงเกิดแรงโน้มถ่วงของโลกมาก การเคลื่อนของตัวแขนจึงเกิดการไม่สมดุลระหว่างการเคลื่อนที่ไปและกลับ มอเตอร์และชุดเฟืองมีช่องว่างระหว่างเฟืองขบมาก เกิดปัญหาแบรคเก็ทขึ้น

ในวิธีการทำชิ้นส่วนต่างๆมีความยุ่งยากในการจัดซื้อชิ้นส่วนต่าง เพราะต้อง เสียเวลาการเดินทาง จึงเกิดการล่าช้าในการปฏิบัติงาน อุปกรณ์บางตัวต้องรอซื้อก็เลยเกิด เป็นปัญหาการล่าช้าดังที่ได้กล่าวถึง

ในการส่งค่าจากตัวป้อนกลับว่าตำแหน่งของแขนกลเคลื่อนที่อยู่ตำแหน่งใดจะ เกิดสัญญาณรบกวนที่เกิดจากการส่งค่ามา ตำแหน่งที่แท้จริงของตัวแขนจริงผิดพลาด ต้อง ใช้เวลาการแก้ไขส่วนนี้ ซึ่งสิ่งเหล่านี้เป็นปัญหาที่เราต้องแก้ไขต่อไป แต่อย่างไรก็ตามผล การทดลองที่ได้ จะบ่งบอกว่าไม่สามารถเคลื่อนที่ได้ตรงตามตำแหน่งที่แท้จริง ปัญหาต่างๆ เหล่านี้เป็นปัญหาที่ทำให้โครงการนี้ยังต้องมีการพัฒนาต่อไป

ต้องมีการพัฒนาทางด้านโปรแกรมการควบคุมการทำงานให้แขนกลสามารถ ทำงานได้มากกว่านี้ เพราะบางอย่างแขนกลตัวนี้ยังไม่สามารถที่จะทำงานได้ เช่น การ ทำงานอย่างต่อเนื่องรับค่าหลายจุดพร้อมกัน เป็นส่วนหนึ่งที่ต้องมีการพัฒนา ตอนนี้แขน กลสามารถเคลื่อนที่ได้ตามจุดที่ระบุโดยส่งผ่านมาจากคอมพิวเตอร์ผ่านช่องอนุกรม

5.2 แนวทางการแก้ไข

การแก้ไขในส่วนของตัวแขนกลนั้น ได้มีการปรับปรุงการขับเคลื่อนใหม่คือให้เกิด อัตราการทดของเฟืองขับเพิ่มขึ้นทำให้เกิดแรงบิดที่มากตามมา ช่วยในการลดการเกิด กระแสไหลมากลดค่ากระแสลงได้ด้วยและยังขับเคลื่อนแขนกลอย่างมีประสิทธิภาพมากขึ้นด้วยตามลำดับ

ปัญหาเกี่ยวกับสัญญาฉบับหนึ่งที่ได้อ่านมาข้างต้นนั้น ได้แก้ไขปัญหาดังกล่าวด้วยการ ออกแบบตัวควบคุมให้ไปช่วยในการแก้ไข และอีกหนึ่งวิธีในการแก้ไขปัญหาคือการใช้ตัว เก็บประจุมากรองสัญญาฉบับก่อนที่สัญญาจะเข้าสู่คอนโทรลเลอร์

5.3 แนวทางการพัฒนาต่อ

1. แก้ไขปัญหาทางด้านที่เกิดจากสัญญาฉบับก่อนให้มีค่าน้อยลงมากยิ่งขึ้น เพื่อที่จะได้ง่ายต่อการควบคุม
2. พัฒนาตัวโครงสร้างทางกลให้มีความแน่นอนมากขึ้น คือความแน่นอน ทางด้านความแข็งแรง
3. ให้แขนกลสามารถทำงานได้หลายจุดอย่างต่อเนื่องและมีความแม่นยำมากยิ่งขึ้น



เอกสารอ้างอิง

1. เรียนรู้และใช้งาน CCS C คอมไพเลอร์ , บริษัท อินโนเวทีฟ เอ็กเพอริเมนต์ จำกัด , กรุงเทพฯ 2521
2. คู่มือการใช้ MATLAB ฉบับสมบูรณ์ , รศ.ดร. มนัส สัจวารศิลป์ , บริษัท ดวงกลมสมัย จำกัด , กรุงเทพฯ , 2543
3. Control System Design Using Matlab , BAHRAM SHAHIAN , Prentice-Hall international Corporation Agency , กรุงเทพฯ , 2533
4. Modern Control Engineering , Katsuhiko Ogata , Prentice-Hall international Corporation Agency , United States of America , 2002





ภาคผนวก



ภาคผนวก

โปรแกรมควบคุมมอเตอร์

เขียนโดยใช้ภาษาซีเขียนลงในตัว PIC 16F877A

เป็นการรับค่าตำแหน่งขนาด 8 บิตมาจากคอมพิวเตอร์

แล้วคำนวณเป็นค่าควิตีใช้เคลส่งเป็น PWM ให้มอเตอร์

```
#include <16f877a.h>

#define TxD    PIN_C6
#define RxD    PIN_C7

#define CLOCK_SP 20000000

#fuses hs
#fuses NOLVP, NOWDT
#fuses NOPROTECT
#use delay(clock=20000000)
#use rs232(baud=9600, xmit=TxD, rcv=RxD)
#bit          cren=0x18.4
int i;
int8 k,pwm[6],control,control1,control2;
int16 pwmz,pwmz;
char strz,data[6];
void main(void)
{
// set_tris_b(0x00);
// output_b(0x0ff);
for(k=0; k<=5; k++)
{
data[k] = 0;
}
while(true)
while(true)
```

```

{
    cren=0;
    delay_ms(0.01);
    cren=1;
    strz=getch();
    if (strz == 'a')
        //output_b(0xff);
        break;
    }
    for(k=1; k<=5; k++)
    {
        data[0]='a';
        data[k]=getch();
    }
    if ((data[0]=='a') & (data[5]=='b'))
    {
        pwm[1]= data[1];
        pwm[2]= data[2];
        pwm[3]= data[3];
        pwm[4]= data[4];
        control1 = pwm[3];
        control2 = pwm[4];
        output_b(control1);
        setup_ccp1(ccp_pwm);
        setup_ccp2(ccp_pwm);
        setup_timer_2(t2_div_by_16,150,1);
        set_pwm1_duty(pwm[1]);
        set_pwm2_duty(pwm[2]);}}

```

โปรแกรมเซ็นเซอร์

เขียนโดยใช้ภาษาซีเขียนลงในตัว PIC 16F877A

เป็นการส่งค่าตำแหน่งขนาด 10 บิตที่ได้มาจากการแปลง

ค่าแรงดัน 0-5 V ที่ส่งมาให้โดยความต้านปรับค่าได้

ที่ติดตั้งอยู่ที่ตัวแขนกล

แล้วคำนวณเป็นค่าความถี่ใช้เคลส่งเป็น PWM ให้มอเตอร์

```
#include <16f877A.h>

#define TxD    PIN_C6
#define RxD    PIN_C7
#define CLOCK_SP 20000000
#define HS
#define NOLVP, NOWDT
#define NOPROTECT
#define ADC=10
#define use    delay(clock=CLOCK_SP)
#define use    rs232(baud=9600, xmit=TxD, rev=RxD)
#define use    fast_io(A)
#define use    fast_io(b)
#define Vbe    0.0048875855327468230694037145650049

int data[16];
int16 degree[16];
```

```

int d,i,f;

int16 data_real;

int16 data_degree;

#int_ccp1                               //interrupt compass-module

void capture_re(){

d=1;

}

void main(void){

I  int16 value1,value2,value3;           //start pressure

float volt;

setup_port_a(ALL_ANALOG);

setup_adc(ADC_CLOCK_INTERNAL);

do {

set_adc_channel(0);

delay_us(20);

value1=Read_ADC();

set_adc_channel(1);

delay_us(20);

value2=Read_ADC();

set_adc_channel(2);

delay_us(20);

value3=Read_ADC();

set_tris_b(0x3d);

d=0;

i=0;

data_degree=0;

data_real=0;

```

```

output_high(pin_b1);           //pin-P/C
output_low(pin_b1);
delay_ms(15);                 //0.05*delay .015 sec
output_high(pin_b1);
setup_ccp1(ccp_capture_re);
enable_interrupts(int_ccp1);
enable_interrupts(global);

while(true){
  If(d==1){
    data[i]=input(pin_b5);
    i++;
    d=0;
  }
  If(i>=16)
    break;
  degree[6]=data[6]*200;      //data[6]=msb
  degree[7]=data[7]*100;
  degree[8]=data[8]*80;
  degree[9]=data[9]*40;
  degree[10]=data[10]*20;
  degree[11]=data[11]*10;
  degree[12]=data[12]*8;
  degree[13]=data[13]*4;
  degree[14]=data[14]*2;
  degree[15]=data[15]*1;    //data[15]=lsb
  for (f=6;f<=15;f++){

```

```

data_degree=data_degree+degree[f];
data_real=data_degree;
}
for (f=6;f<=16;f++){
degree[f]=0;
data[f]=0;
}

//output_high (pin_d1);
printf("a%d ", data_real); // Send Data of Angle
printf("b%d",value1); // Send Data of depth Pin a0 Pin
number 2
printf("c%d",value2); // Send Data of Balance_x Pin a1 Pin
number 3
printf("d%d",value3); // Send Data of Balance_y Pin a2 Pin
number 4
printf("e");
//output_low (pin_d1);
d=0;
i=0;
f=0;
data_degree=0;
data_real=0; //2*delay.015sec
delay_ms(15);
} while(TRUE);
}

```

โปรแกรมส่วนของ VB

เป็นการรับค่าตำแหน่งที่ส่งมาจากPICตัว
เซ็นเซอร์ค่าขนาด 10 บิตแล้วทำการเปรียบ
เทียบกับค่าของตำแหน่งที่ถูกกำหนดจาก
ภายนอกจากตัวบุคคลแล้วนำค่า Errorที่
ได้มาเข้าตัวควบคุม PID แล้วส่งค่านี้ออกไปให้
PIC คุมมอเตอร์เป็นค่าขนาด 8 บิต

```
Dim Control As Integer
Dim sensor_angle, data_angle As Single
Dim error_angle As Integer
Dim angle_auto As Integer
Dim Ts As Single
Dim stop1 As Integer

'eM1_old, eM2_old,
'Dim Ki1_M1, Ki1_M1_old, Kd1_M1, Ki1_text, Kd1_text, Ka1_text
As Single
'Dim Ki1_M2, Ki1_M2_old, Kd1_M2, PID_M2 As Single

'Dim error_angle2 As Variant

'Dim data_angle As Single
'Dim eM1_KI1_old As Single
'Dim Ts As Single
'Dim fn1 As Long
'Dim KI1 As Single
'Dim Ki1_0 As Single
'Dim Ki1_text As Single
'Dim error_angle_old As Single
Dim go_speed1 As Integer
Dim go_speed2 As Integer

Dim PWM_M1 As Integer
Dim PWM_M2 As Integer

Dim Pwm1 As Integer
Dim Pwm2 As Integer
Dim Pwm3 As Integer
Dim Pwm4 As Integer
```

```
Dim data_Depth As Integer
Dim data_X As Integer
Dim data_y As Integer
```

```
Private Sub Command1_Click()
Pwm1 = 125
Pwm2 = 125
go_speed1 = 50
go_speed2 = 50
Control1 = 245
Control2 = 0
Data = "c" + Chr(Pwm1) + Chr(Pwm2) + Chr(Control1) +
Chr(Control2) + "d"
Comm1.Output = Data
End Sub
```

```
Private Sub Command2_Click()
go_speed1 = 0
go_speed2 = 0
' Data = "c" + Chr(Pwm1) + Chr(Pwm2) + Chr(Control1) +
Chr(Control2) + "d"
'Comm1.Output = Data

stop1 = 255
angle_auto = 0
PWM_M1 = 0
PWM_M2 = 0
Data = "a" + Chr(PWM_M1) + Chr(PWM_M2) + Chr(Control1) +
Chr(Control2) + "b"
Comm1.Output = Data
PID_M1 = 0
PID_M2 = 0
Data = "c" + Chr(PID_M1) + Chr(PID_M2) + Chr(Control1) +
Chr(Control2) + "d"
Comm1.Output = Data

End Sub
```

```
Private Sub Command3_Click()
Control1 = 249
Control2 = 0
Data = "c" + Chr(Pwm1) + Chr(Pwm2) + Chr(Control1) +
Chr(Control2) + "d"
Comm1.Output = Data
```

End Sub

Private Sub Command4_Click()

Control1 = 246

Control2 = 0

Data = "c" + Chr(Pwm1) + Chr(Pwm2) + Chr(Control1) +
Chr(Control2) + "d"

Comm1.Output = Data

End Sub

Private Sub Command5_Click()

go_speed1 = 50

go_speed2 = 50

Control1 = 250

Control2 = 0

Data = "c" + Chr(Pwm1) + Chr(Pwm2) + Chr(Control1) +
Chr(Control2) + "d"

Comm1.Output = Data

End Sub

Private Sub Command6_Click()

stop1 = 255

angle_auto = 0

PWM_M1 = 0

PWM_M2 = 0

Data = "a" + Chr(PWM_M1) + Chr(PWM_M2) + Chr(Control1) +
Chr(Control2) + "b"

Comm1.Output = Data

PID_M1 = 0

PID_M2 = 0

Data = "c" + Chr(PID_M1) + Chr(PID_M2) + Chr(Control1) +
Chr(Control2) + "d"

Comm1.Output = Data

End Sub

Private Sub Command7_Click()

stop1 = 0

angle_auto = 255

End Sub

Private Sub Command8_Click()

Text_ka1.Text = 0

Text_ki1.Text = 0

Text_kd1.Text = 0

Text_ka2.Text = 0

Text_ki2.Text = 0

Text_kd2.Text = 0

Text_ka3.Text = 0

Text_ki3.Text = 0

Text_kd3.Text = 0

End Sub

```
Private Sub Command9_Click()  
Form2.Visible = True  
Form3.Visible = True  
End Sub
```

```
Private Sub Form_Load()  
If Not Comm1.PortOpen Then  
Comm1.CommPort = 1  
Comm1.PortOpen = True  
End If  
go_speed = 0  
angle_auto = 255  
Pwm1 = 200  
Pwm2 = 200  
Ts = 70  
stop1 = 0
```

End Sub

```
Private Sub Option1_Click()  
angle_auto = 255  
End Sub
```

```
Private Sub Option2_Click()  
angle_auto = 0  
End Sub
```

```
Private Sub Timer1_Timer()  
Dim a, b, c, d, t, x1, x2, X3, X4, x5, pos1, pos2, pos3, pos4  
Dim PID_M1 As Integer  
Dim PID_M2 As Integer  
Dim PID2_M1 As Integer  
Dim PID2_M2 As Integer  
Dim PID3_M1 As Integer  
Dim PID3_M2 As Integer
```

```
'If MSComm1.InBufferCount > 0 Then  
pom1_s = Comm1.Input  
Text1.Text = pom1_s
```

```
t = Text1.Text  
c = Len(t)
```

```
'End If  
searchPattern$ = "a"  
pos1 = InStr(t, searchPattern$)
```

```

searchPattern$ = "b"
pos2 = InStr(t, searchPattern$)

searchPattern$ = "c"
pos3 = InStr(t, searchPattern$)

searchPattern$ = "d"
pos4 = InStr(t, searchPattern$)

searchPattern$ = "e"
pos5 = InStr(t, searchPattern$)

If pos1 > 0 And pos2 > 0 Then
x1 = pos2 - pos1
If x1 > 0 Then
a = Mid$(t, pos1 + 1, x1 - 1)
Text4 = a          'output data a_Angle
sensor_angle = a
End If
End If
*****
If pos2 > 0 And pos3 > 0 Then
x2 = pos3 - pos2
If x2 > 0 Then
b = Mid$(t, pos2 + 1, x2 - 1)
Text5 = b          'output data b
End If
End If
*****
If pos4 > 0 And pos3 > 0 Then
X3 = pos4 - pos3
If X3 > 0 Then
c = Mid$(t, pos3 + 1, X3 - 1)
Text2 = c          'output data c
End If
End If
*****
If pos5 > 0 And pos4 > 0 Then
X4 = pos5 - pos4
If X4 > 0 Then
d = Mid$(t, pos4 + 1, X4 - 1)
Text3 = d          'output data d
End If
End If
*****

```

```

sensor_angle = Val(Text4.Text)
'If (Text6.Text) <> 0 Then
    If Text6.Text = "" Then Text6.Text = 0
    If Text6.Text > 359 Then Text6.Text = 359
    If Text6.Text < -359 Then Text6.Text = -359
'End If
data_angle = Val(Text6.Text)

```

```

Ki1_text = Val(Text_ki1.Text)
Kd1_text = Val(Text_kd1.Text)
Ka1_text = Val(Text_ka1.Text)
'caculate error
error_angle = data_angle - sensor_angle

```

```

If error_angle > 0 And error_angle < 180 Then
error_angle2 = error_angle * -1
Text16.Text = error_angle2
End If

```

```

If error_angle < 0 And error_angle > -180 Then
error_angle2 = error_angle * -1
End If

```

```

If error_angle >= 180 And error_angle <= 360 Then
error_angle2 = ((359 - data_angle) + sensor_angle)
End If

```

```

If error_angle <= -180 And error_angle >= -360 Then
error_angle2 = ((359 - sensor_angle) + data_angle) * -1
End If

```

```

If error_angle = 0 Then error_angle2 = 0

```

```

Text7.Text = error_angle2
'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxx

```

```

'caculate Ka_M1
Ka1_M1 = Ka1_text * error_angle2
'caculate KI_M1
Ki1_M1 = Ki1_M1_old + (Ki1_text * (Ts / 2 * (error_angle2 +
eM1_old)))
Ki1_M1_old = Ki1_M1
'caculate Kd_M1

```

```

Kd1_M1 = ((error_angle2 - eM1_old) / Ts) * Kd1_text
'caculate PID_M1
PID_M1 = ((Ka1_M1) + Ki1_M1 + Kd1_M1) + go_speed1
If PID_M1 <> 0 Then
    If PID_M1 > 255 Then PID_M1 = 255
    If PID_M1 < -255 Then PID_M1 = -255
End If
Text11.Text = PID_M1
'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

'caculate Ka_M2
Ka1_M2 = Ka1_text * error_angle2
'caculate KI_M2
Ki1_M2 = Ki1_M2_old + (Ki1_text * (Ts / 2 * (error_angle2 +
eM1_old)))
Ki1_M2_old = Ki1_M2
'caculate Kd_M2
Kd1_M2 = ((error_angle2 - eM1_old) / Ts) * Kd1_text
'caculate PID_M2
PID_M2 = ((Ka1_M2) + Ki1_M2 + Kd1_M2) + go_speed2
If PID_M2 <> 0 Then
    If PID_M2 > 255 Then PID_M2 = 255
    If PID_M2 < -255 Then PID_M2 = -255
End If
Text8.Text = PID_M2
eM1_old = error_angle2

If PID_M1 > 0 Then control_M1 = 253
If PID_M1 < 0 Then
control_M1 = 254
PID_M1 = -1 * PID_M1
End If

If PID_M2 > 0 Then control_M2 = 251
If PID_M2 < 0 Then
control_M2 = 247
PID_M2 = -1 * PID_M2
End If

Control1 = control_M1 And control_M2
Control2 = 255
If angle_auto = 255 Then
    Data = "c" + Chr(PID_M1) + Chr(PID_M2) + Chr(Control1) +
Chr(Control2) + "d"
    Comm1.Output = Data
End If

' caculate PID depth
*****

```



```

Ka3_text = Val(Text_ka3.Text)
'caculate error Balance
error_Balance = (data_Balance - sensor_Balance)
Text22.Text = error_Balance
'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
'caculate Ka2_M1
Ka3_M1 = Ka3_text * -error_Balance
'caculate KI3_M1
Ki3_M1 = Ki3_M1_old + (Ki3_text * (Ts / 2 * (-error_Balance +
eM3_old)))
Ki3_M1_old = Ki3_M1
'caculate Kd3_M1
Kd3_M1 = ((-error_Balance - eM3_old) / Ts) * Kd3_text
'caculate PID3_M1
PID3_M1 = ((Ka3_M1) + Ki3_M1 + Kd3_M1)
If PID3_M1 <> 0 Then
    If PID3_M1 > 255 Then PID3_M1 = 255
    If PID3_M1 < -255 Then PID3_M1 = -255
End If
Text23.Text = PID3_M1
'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
'caculate Ka3_M2
Ka3_M2 = Ka3_text * error_Balance
'caculate KI3_M2
Ki3_M2 = Ki3_M2_old + (Ki3_text * (Ts / 2 * (error_Balance +
eM3_old)))
Ki3_M2_old = Ki3_M2
'caculate Kd3_M2
Kd3_M2 = ((error_Balance - eM3_old) / Ts) * Kd3_text
'caculate PID3_M2
PID3_M2 = ((Ka3_M2) + Ki3_M2 + Kd3_M2)
If PID3_M2 <> 0 Then
    If PID3_M2 > 255 Then PID3_M2 = 255
    If PID3_M2 < -255 Then PID3_M2 = -255
End If
Text10.Text = PID3_M2
eM3_old = error_Balance
'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

PWM_M1 = PID2_M1 + PID3_M1
PWM_M2 = PID2_M2 + PID3_M2
Text14.Text = PWM_M1
Text15.Text = PWM_M2

If PWM_M1 > 0 Then control_M1 = 253

If PWM_M1 < 0 Then
control_M1 = 254
PWM_M1 = -1 * PWM_M1
End If

```

```
If PWM_M2 > 0 Then control_M2 = 251
```

```
If PWM_M2 < 0 Then
```

```
control_M2 = 247
```

```
PWM_M2 = -1 * PWM_M2
```

```
End If
```

```
Control1 = control_M1 And control_M2
```

```
Control2 = 255
```

```
If stop1 = 0 Then
```

```
Data = "a" + Chr(PWM_M1) + Chr(PWM_M2) + Chr(Control1) +  
Chr(Control2) + "b"
```

```
Comm1.Output = Data
```

```
End If
```

```
ttt = 50
```

```
End Sub
```

