



การควบคุมแขนกลด้วยหลักการอินเวอร์สคิเนเมติกส์

Robot Arm Control by Inverse Kinematics

นายวรกันต์      ฟองศรีณย์      รหัส 52371436

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ..... 20 ก.ค. 2555
เลขทะเบียน..... 16961143
เลขเรียกหนังสือ..... 2/ค
มหาวิทยาลัยนเรศวร ๑๖๕๑๑ ๒๕๕๕

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2555



## ใบรับรองปริญญาโท

ชื่อหัวข้อโครงการ การควบคุมแกนกลด้วยหลักการอินเวอร์สคิเนเมติกส์  
ผู้ดำเนินโครงการ นายวรกันต์ พงศ์รัมย์ รหัส 52371436  
ที่ปรึกษาโครงการ อาจารย์เศรษฐา ตั้งคำวานิช  
สาขาวิชา วิศวกรรมคอมพิวเตอร์  
ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์  
ปีการศึกษา 2555

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนครสวรรค์ อนุมัติให้ปริญญาโทฉบับนี้เป็นส่วนหนึ่ง  
ของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

ได้รับทูลเกล้าฯ ถวาย ที่ปรึกษาโครงการ  
(อาจารย์เศรษฐา ตั้งคำวานิช)

.....กรรมการ  
(ดร.พนัส นฤฤทธิ์)

.....กรรมการ  
(ดร.สุรเดช จิตประไพกุลศาล)

.....กรรมการ  
(อาจารย์รัฐภูมิ วรรณสาสน์)

ชื่อหัวข้อโครงการ	การควบคุมแขนกลด้วยหลักการอินเวอร์สคิเนแมติกส์
ผู้ดำเนินโครงการ	นายวรกันต์ ฟองศรีณย์ รหัส 52371436
ที่ปรึกษาโครงการ	อาจารย์ศรชญา ตั้งคำวานิช
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2555

---

### บทคัดย่อ

โครงการนี้นำเสนอการศึกษาและการพัฒนาระบบการควบคุมหุ่นยนต์แขนกลอัตโนมัติ โดยใช้หลักการของอินเวอร์สคิเนแมติกส์ในการคำนวณการเคลื่อนที่ไปยังจุดหมายได้ถูกต้องและแม่นยำ จากการรับค่าพิกัดของจุดหมายแล้วนำมาคำนวณค่าหามุมของแขนแต่ละข้อเพื่อให้ปลายแขนไปยังพิกัดเป้าหมาย ซึ่งระบบการคำนวณถูกพัฒนาขึ้นบนภาษา Visual studio C# 2010

แขนกลอัตโนมัติรับคำสั่งควบคุมจากโปรแกรมที่พัฒนาขึ้นมาโดยโปรแกรมจะส่งค่าการคำนวณด้วยหลักการของอินเวอร์สคิเนแมติกส์มายังบอร์ดไมโครคอนโทรลเลอร์ จากนั้นไมโครคอนโทรลเลอร์จะส่งการเซอร์โวในแต่ละส่วนทำให้แขนเคลื่อนที่ไปยังพิกัดที่ต้องการ ซึ่งแขนกลอัตโนมัติสามารถนำไปประยุกต์ใช้ในการหยิบจับสิ่งของอัตโนมัติได้

**Project title**            Robot Arm Control by Inverse Kinematics  
**Name**                     Mr. Woragun    Fongsarun            ID. 52371436  
**Project advisor**        Mr. Settha        Tangkawanit  
**Major**                     Computer Engineering  
**Department**            Electrical and Computer Engineering  
**Academic year**         2012

---

### **Abstract**

This project presents a learning and developing of robot arm control system using an Inverse Kinematics is principles for calculation in order to moves an arm to the destination with correctly and precisely. This system is calculated by coordinates of the destination, to calculate the angle for each arm in order that the forearm can get a coordinates and moves forward to destination. The calculation system was developed by Visual Studio C# 2010.

Automatic arm receive a command from the IK program, which was developed by the calculation of Inverse Kinematics principles, send to a microcontroller board and then assigns servo motor in each part to move arm to the desired coordinates. This automatic arm can apply to an automatic pick and place object task.

## กิตติกรรมประกาศ

ปริชญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี เนื่องจากการได้รับความช่วยเหลืออนุเคราะห์จากบุคคลหลายๆท่านด้วยกัน โดยเฉพาะอย่างยิ่งท่าน อาจารย์เศรษฐา ตั้งคำวานิช ที่ได้สละเวลาเพื่อให้คำปรึกษาและเสนอแนะแนวทางสำหรับการแก้ไขปัญหาที่เกิดขึ้นขณะการจัดทำโครงการ พร้อมด้วยคณะกรรมการที่ปรึกษาโครงการ คร.พนัส นัถฤทธิ์ คร.สุรเดช จิตประไพกุลสาล และ อาจารย์รัฐภูมิ วรรณสาสน์ ที่ให้คำแนะนำในการเขียนรูปเล่มปริชญานิพนธ์

ผู้จัดทำโครงการจึงขอขอบคุณทุกๆท่านที่มีส่วนร่วมในการให้ความรู้และคำปรึกษาต่างๆที่เกี่ยวข้องกับ โครงการนี้ จนสำเร็จลุล่วงในที่สุด

ผู้จัดทำโครงการวิศวกรรม  
นาย วรรณันต์ ฟองศรีณย์



## สารบัญ

หน้า

บทคัดย่อภาษาไทย .....	ก
บทคัดย่อภาษาอังกฤษ .....	ข
กิตติกรรมประกาศ .....	ค
สารบัญ .....	ง
สารบัญรูป .....	ช
สารบัญตาราง .....	ฉ
บทที่ 1 บทนำ .....	1
1.1 ความเป็นมาและความสำคัญของโครงการ .....	1
1.2 วัตถุประสงค์ของโครงการ .....	1
1.3 ขอบเขตของโครงการ .....	1
1.4 ขั้นตอนการดำเนินงานของโครงการ .....	2
1.5 แผนการดำเนินงาน .....	2
1.6 ผลที่คาดว่าจะได้รับ .....	4
1.7 งบประมาณดำเนินโครงการ .....	4
บทที่ 2 หลักการและทฤษฎี .....	5
2.1 แขนงหุ่นยนต์ .....	5
2.2 ข้อต่อ .....	6

## สารบัญ(ต่อ)

2.2.1 การหมุนแกน .....	7
2.2.2.1 การหมุนรอบแกน X หรือมุม Pitch .....	8
2.2.2.2 การหมุนรอบแกน Y หรือมุม Yaw .....	9
2.2.2.3 การหมุนรอบแกน Z หรือมุม Roll .....	10
2.3 ทฤษฎีการเคลื่อนที่ .....	11
2.3.1 Forward kinematics .....	11
2.3.2 Inverse kinematics .....	11
2.4 มอเตอร์กระแสตรง .....	15
2.5 เซอร์โวมอเตอร์ .....	16
2.6 ไมโครคอนโทรลเลอร์ ATmega1280 .....	17
<b>บทที่ 3 วิธีการดำเนินโครงการ</b>	
3.1 Conceptual System Design .....	21
3.2 Diagram การทำงาน .....	22
3.3 การพัฒนาโครงสร้างของหุ่นยนต์แขนกลอัตโนมัติ .....	22
3.4 วิธีการใช้งาน โปรแกรม Inverse Kinematics .....	26
<b>บทที่ 4 ผลการทดลอง</b>	
4.1 จุดประสงค์ของการทดลอง .....	28

## สารบัญ(ต่อ)

4.2 ขั้นตอนการทดลอง .....	28
4.3 ขั้นตอนทดสอบโปรแกรม .....	29
4.4 ตารางเปรียบเทียบผลการทดลอง .....	47
บทที่ 5 สรุปผลการทดลอง	
5.1 สรุปผลการทดลอง .....	51
5.2 ปัญหาที่พบ .....	51
5.3 แนวทางการแก้ปัญหา .....	52
5.4 แนวทางการพัฒนาต่อ .....	53
เอกสารอ้างอิง .....	54
ภาคผนวก ก .....	56
ภาคผนวก ข .....	59
ประวัติผู้จัดทำ .....	62



## สารบัญรูป

รูปที่	หน้า
รูปที่ 2.1 แขนกล (Robot Arm) .....	5
รูปที่ 2.2 ข้อต่อ (Joint) .....	6
รูปที่ 2.3 ทฤษฎีแกนคั้งฉาก .....	7
รูปที่ 2.4 แสดงการหมุนของแกน Roll, Yaw, Pitch .....	8
รูปที่ 2.5 การหมุนรอบแกน X .....	8
รูปที่ 2.6 การหมุนรอบแกน Y .....	9
รูปที่ 2.7 การหมุนรอบแกน Z .....	10
รูปที่ 2.8 แสดงทิศทางของ Degree of Freedom (DOF) .....	11
รูปที่ 2.9 แสดงตัวอย่างของการหา $\theta_1$ .....	12
รูปที่ 2.10 แสดงวิธีการคำนวณสมการ Atan2 .....	13
รูปที่ 2.11 แสดงลำดับของมุมและท่อนแขน .....	14
รูปที่ 2.12 มอเตอร์กระแสตรง (DC Motor) .....	14
รูปที่ 2.13 เซอร์โวมอเตอร์ .....	16
รูปที่ 2.14 สัญญาณลูกคลื่นที่สามารถป้อนให้กับเซอร์โวมอเตอร์ .....	16
รูปที่ 2.15 แสดงตำแหน่งของเซอร์โวมอเตอร์เมื่อป้อนความกว้าง pulse .....	17
รูปที่ 2.16 ไมโครคอนโทรลเลอร์ ATmega1280 .....	18
รูปที่ 3.1 Conceptual System Design .....	21
รูปที่ 3.2 ไคอะแกรมแสดงการทำงานของระบบควบคุมแขนกลอัตโนมัติ .....	22

## สารบัญรูป(ต่อ)

รูปที่ 3.3 รายละเอียดของเซอร์โวมอเตอร์ รุ่น SR431 .....	23
รูปที่ 3.4 รูปร่างและขนาดของเซอร์โวมอเตอร์ รุ่น Futaba S3003 .....	23
รูปที่ 3.5 รายละเอียดของมอเตอร์กระแสตรง Motor womn gear 12vdc .....	23
รูปที่ 3.6 แสดงโครงสร้างของหุ่นยนต์แขนกล .....	24
รูปที่ 3.7 แสดงขนาดของแขนแต่ละ link และตัวกริปเปอร์ .....	25
รูปที่ 3.8 แสดงขนาดของส่วนประกอบต่างๆของแขนกลอัตโนมัติ .....	25
รูปที่ 3.9 ลักษณะหน้าตาของโปรแกรม .....	26
รูปที่ 4.1 แสดงตัว โปรแกรมที่พัฒนาขึ้นมา .....	29
รูปที่ 4.2 การทดสอบ โปรแกรมครั้งที่ 1 .....	30
รูปที่ 4.3 การทดสอบ โปรแกรมครั้งที่ 2 .....	31
รูปที่ 4.4 แสดงกราฟการทำมุมของแขนและปลายแขนของการทดสอบครั้งที่ 2 .....	31
รูปที่ 4.5 แสดงลักษณะของแขนกลของการทดสอบครั้งที่ 2 .....	32
รูปที่ 4.6 การทดสอบ โปรแกรมครั้งที่ 3 .....	33
รูปที่ 4.7 การทดสอบ โปรแกรมครั้งที่ 4 .....	34
รูปที่ 4.8 แสดงกราฟการทำมุมของแขนและปลายแขนของการทดสอบครั้งที่ 4 .....	34
รูปที่ 4.9 แสดงลักษณะของแขนกลของการทดสอบครั้งที่ 4 .....	35
รูปที่ 4.10 การทดสอบ โปรแกรมครั้งที่ 5 .....	36
รูปที่ 4.11 แสดงกราฟการทำมุมของแขนและปลายแขนของการทดสอบครั้งที่ 5 .....	36
รูปที่ 4.12 แสดงลักษณะของแขนกลของการทดสอบครั้งที่ 5 .....	37

## สารบัญรูป(ต่อ)

รูปที่ 4.13 การทดสอบโปรแกรมครั้งที่ 6 .....	38
รูปที่ 4.14 แสดงกราฟการทำมุมของแขนและปลายแขนของการทดสอบครั้งที่ 6 .....	38
รูปที่ 4.15 แสดงลักษณะของแขนกลของการทดสอบครั้งที่ 6 .....	39
รูปที่ 4.16 การทดสอบโปรแกรมครั้งที่ 7 .....	40
รูปที่ 4.17 แสดงกราฟการทำมุมของแขนและปลายแขนของการทดสอบครั้งที่ 7 .....	40
รูปที่ 4.18 แสดงลักษณะของแขนกลของการทดสอบครั้งที่ 7 .....	41
รูปที่ 4.19 การทดสอบโปรแกรมครั้งที่ 8 .....	42
รูปที่ 4.20 แสดงกราฟการทำมุมของแขนและปลายแขนของการทดสอบครั้งที่ 8 .....	42
รูปที่ 4.21 แสดงลักษณะของแขนกลของการทดสอบครั้งที่ 8 .....	43
รูปที่ 4.22 การทดสอบโปรแกรมครั้งที่ 9 .....	44
รูปที่ 4.23 การทดสอบ โปรแกรมครั้งที่ 10 .....	45
รูปที่ 4.24 การทดสอบ โปรแกรมครั้งที่ 11.....	46
รูปที่ 4.25 แสดงกราฟการทำมุมของแขนและปลายแขนของการทดสอบครั้งที่ 11 .....	46
รูปที่ ก-1 กราฟแสดงการ linear ค่าจากตารางเปรียบเทียบใน ตารางที่ ก-1 .....	56
รูปที่ ก-2 กราฟแสดงการ linear ค่าจากตารางเปรียบเทียบใน ตารางที่ ก-2 .....	57
รูปที่ ก-3 กราฟแสดงการ linear ค่าจากตารางเปรียบเทียบใน ตารางที่ ก-3 .....	58

## สารบัญตาราง

ตารางที่	หน้า
ตารางที่ 1.1 แผนการดำเนินงาน .....	2
ตารางที่ 2.1 แสดงตัวอย่างของการหา $\theta$ , .....	12
ตารางที่ 4.1 ตารางเปรียบเทียบผลการทดลอง .....	47
ตารางที่ ก-1 ตารางเปรียบเทียบ value กับองศาจริงของ Servo ข้อที่ 1 .....	56
ตารางที่ ก-2 ตารางเปรียบเทียบ value กับองศาจริงของ Servo ข้อที่ 2 .....	57
ตารางที่ ก-3 ตารางเปรียบเทียบค่าแรงดันของกับองศาจริงของ dc motor .....	58



# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของโครงการ

เนื่องจากในปัจจุบันได้มีการนำเทคโนโลยีมาใช้ในการชีวิตประจำวันและงานในด้านต่างๆ อาทิ เช่น ด้านอุตสาหกรรม ด้านเกษตรกรรม ในลักษณะงานบางอย่าง มีการทำงานที่ซ้ำไปซ้ำมา มีความซับซ้อนทำให้การทำงานมีความเชื่องช้า หรือมีความเสี่ยงที่เป็นอันตรายต่อมนุษย์ งานบางอย่างต้องใช้ความแม่นยำและความเร็วในการทำงาน จึงได้มีการพัฒนาหุ่นยนต์ที่สามารถใช้งานในด้านนั้นๆ เพื่ออำนวยความสะดวกสบายในการทำงาน เป็นต้น

ดังนั้นทางคณะผู้จัดทำโครงการจึงมีแนวคิดในการออกแบบและสร้างหุ่นยนต์แขนกลอัตโนมัติ ซึ่งจะเป็นอุปกรณ์ที่สามารถช่วยให้มนุษย์ลดความเสี่ยงในการทำงานในพื้นที่ที่เป็นอันตรายและเพิ่มประสิทธิภาพในการทำงานมากยิ่งขึ้น และยังสามารถนำไปประยุกต์ใช้กับการแข่งขัน @HOME ROBOT หุ่นยนต์เพื่อนอัจฉริยะได้

### 1.2 วัตถุประสงค์

1. สามารถสร้างหุ่นยนต์แขนกล ที่มีระบบควบคุมแบบป้อนกลับได้
2. สามารถตั้งการเคลื่อนแขนกล ไปยังวัตถุที่สนใจเพื่อจับวัตถุหรือวางได้

### 1.3 ขอบเขตโครงการ

1. แขนกลสามารถเคลื่อนที่ไปจับวัตถุที่สนใจได้โดยอัตโนมัติจากการระบุค่าตำแหน่งพิกัดวัตถุ
2. แขนกลสามารถหยิบจับวัตถุได้

#### เงื่อนไขของวัตถุ

- วัตถุต้องมีน้ำหนักไม่เกิน 1 Kg.
- วัตถุต้องมีรูปทรงเป็นทรงกระบอกหรือสี่เหลี่ยมพื้นผ้าเท่านั้น
- ขนาดของวัตถุต้องมีขนาดไม่เกินช่วงเปิดสุดของกริปเปอร์ซึ่งมีขนาด 4 Cm.





### 1.6 ผลที่คาดว่าจะได้รับ

1. ได้หุ่นยนต์แขนกลอัตโนมัติที่สามารถหยิบจับวัตถุที่สนใจ
2. สามารถนำไปประกอบกับโครงงานที่เกี่ยวข้องได้
3. สามารถนำไปประยุกต์กับการแข่งขัน @HOME ROBOT หุ่นยนต์เพื่อนอัจฉริยะได้

### 1.7 งบประมาณ

1. ค่าวัสดุและอุปกรณ์	1,200	บาท
2. ค่าจัดทำรูปเล่มรายงาน	800	บาท
รวมเป็นเงินทั้งสิ้น	2,000	บาท
*ขออนุมัติตัวเงินทุกรายการ		





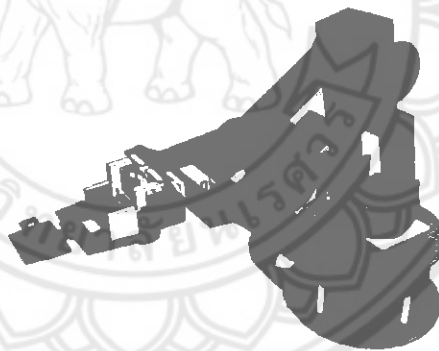
## บทที่ 2

### หลักการและทฤษฎี

ในบทนี้จะกล่าวถึงทฤษฎีที่ใช้ในการจัดทำโครงงาน ซึ่งประกอบไปด้วยทฤษฎีของแขนหุ่นยนต์ ทฤษฎีของข้อต่อและการหมุนแกน ทฤษฎีการเคลื่อนที่ของแขนกล ทฤษฎีการควบคุมมอเตอร์กระแสตรง (DC Motor) ทฤษฎีการควบคุมเซอร์โวมอเตอร์ (Servo Motor) และทฤษฎีพื้นฐานของไมโครคอนโทรลเลอร์ (Microcontroller)

#### 2.1 แขนหุ่นยนต์ (Robot Arm)

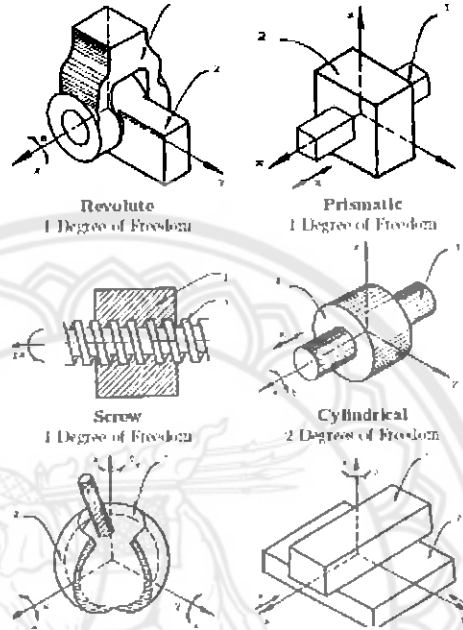
แขนหุ่นยนต์ที่ใช้ในงานวิจัยนี้เป็นแขนหุ่นยนต์ที่มีข้อต่อแบบหมุนทั้งหมด เรียกว่า articulated robot ประกอบไปด้วย 3 links และ 5 joint ประกอบกันเป็นแขนหุ่นยนต์ ซึ่งในแต่ละจุดที่สามารถเคลื่อนที่ได้อิสระ เรียกว่า degree-of-freedom (DOF)



รูปที่ 2.1 แขนกล (Robot Arm) [1]

### 2.2 ข้อต่อ (joint)

ข้อต่อ (joint) คือจุดเชื่อมท่อนแขน (link) แต่ละท่อนประกอบกันเป็นหุ่นยนต์แขนกล ข้อต่อมีหลายแบบ แต่ละแบบก็จะอนุญาตให้เกิดการเคลื่อนที่ของท่อนแขนที่แตกต่างกันไป รูปที่ 2.3 แสดงข้อต่อแบบต่างๆ ที่นิยมใช้

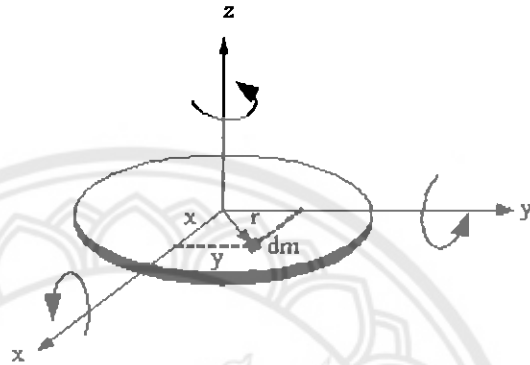


รูปที่ 2.2 ข้อต่อ (Joint) [2]

ในการสร้างแขนกลโดยทั่วไปข้อต่อที่นิยมใช้มากที่สุดคือข้อต่อแบบหมุน (revolute joint) และข้อต่อแบบเลื่อน (prismatic joint) สำหรับข้อต่อแบบหมุนท่อนแขนสองท่อนถูกยึดติดกันที่จุดหมุนซึ่งอยู่บนท่อนแขน โดยแต่ละท่อนสามารถหมุนได้รอบจุดหมุนนี้ เราสามารถบอกตำแหน่งของสองท่อนแขนที่สัมพันธ์กันด้วยมุมที่ท่อนแขนหมุนไป ส่วนข้อต่อแบบเลื่อนนั้นท่อนแขนสองท่อนติดอยู่ด้วยกัน ในลักษณะเดียวกับเสาอากาศวิทยุรถยนต์ที่ขีดยึดได้ โดยท่อนแขนแต่ละท่อนสามารถเลื่อนเข้าออกได้ในหนึ่งทิศทาง เราสามารถระบุตำแหน่งที่สัมพันธ์กันของสองท่อนแขนได้จากระยะเลื่อนเข้าออกดังกล่าว จะเห็นได้ว่าข้อต่อแบบหมุนและข้อต่อแบบเลื่อนมีระดับเสรีของการเคลื่อนที่เป็นหนึ่ง เราเรียกตัวแปรที่กำหนดการเคลื่อนที่นี้ ซึ่งได้แก่มุมหมุนของข้อต่อแบบหมุนและระยะเลื่อนของข้อต่อแบบเลื่อนว่าเป็นพารามิเตอร์ของข้อต่อ การมีระดับเสรีของการเคลื่อนที่เป็นหนึ่งทำให้ง่ายในการออกแบบและวิเคราะห์ ข้อต่อทั้งสองแบบจึงถูกใช้มากที่สุดในการสร้างแขนกล โดยแขนกลที่มีระดับเสรีสูงๆ ก็สามารถสร้างขึ้นได้ โดยการประกอบท่อนแขนหลายท่อนด้วยข้อต่อสองแบบนี้

### 2.2.1 การหมุนแกน (Rotation)

ก่อนจะศึกษาเรื่องการหมุนนั้นต้องรู้จักทฤษฎีแกนตั้งฉากก่อนทฤษฎีแกนตั้งฉากเป็นทฤษฎีที่ใช้ในการคำนวณหาค่าโมเมนต์ความเฉื่อยของวัตถุแข็งเกร็งที่มีลักษณะเป็นแผ่นบาง ๆ พิจารณาวัตถุหมุนรอบแกน  $x$ ,  $y$  และ  $z$



รูปที่ 2.3 ทฤษฎีแกนตั้งฉาก [3]

จาก 
$$I = \int r^2 dm \quad (2.1)$$

เมื่อ  $r$  เป็นระยะจาก  $dm$  ถึงแกน  $x$  และ  $y$  เนื่องจากเป็นวัตถุแผ่นบางจะได้

$$I_z = \int r^2 dm \quad (2.2)$$

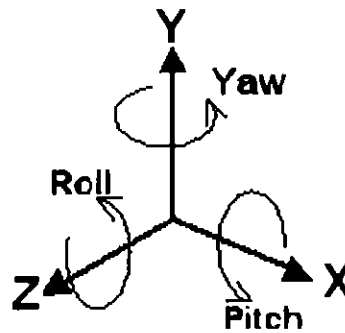
แต่  $r^2 = x^2 + y^2$

$$I_z = \int (x^2 + y^2) dm \quad (2.3)$$

$$= \int x^2 dm + \int y^2 dm \quad (2.4)$$

$$I_z = I_x + I_y \quad (2.5)$$

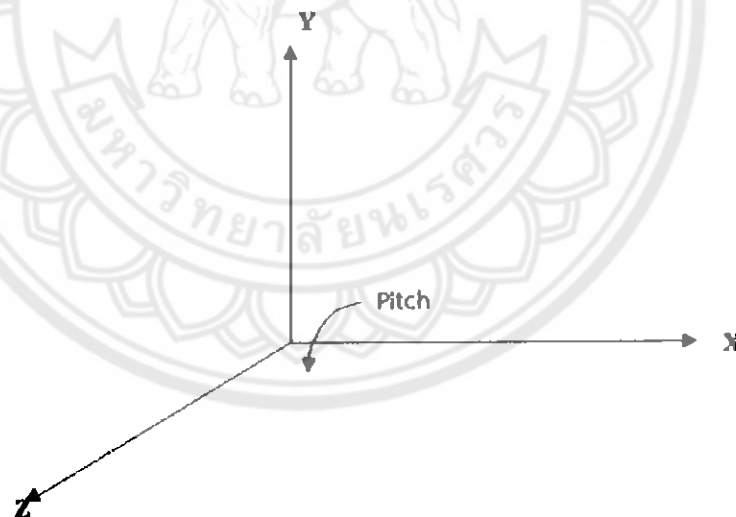
การหมุนภาพใน 3 มิติแบ่งเป็น 3 แบบดังนี้คือ



รูปที่ 2.4 แสดงการหมุนของแกน Roll, Yaw, Pitch [4]

แกนแต่ละแกนจะมีชื่อเรียกแกนหมุนแบบสากลคือ Roll, Yaw, Pitch ซึ่งแกน Roll คือ แกนที่อยู่ในแนวแกน Z ดังรูปที่ 2.6 แกน Yaw คือแกนที่อยู่ในแกนแนวตั้งหรือแกน Y และแกน Pitch คือแกนที่อยู่ในแนวระนาบหรือแกน X

#### 2.2.1.1 หมุนรอบแกน X หรือมุม Pitch



รูปที่ 2.5 การหมุนรอบแกน X

จะได้สมการ ดังนี้

$$X_1 = X \quad (2.6)$$

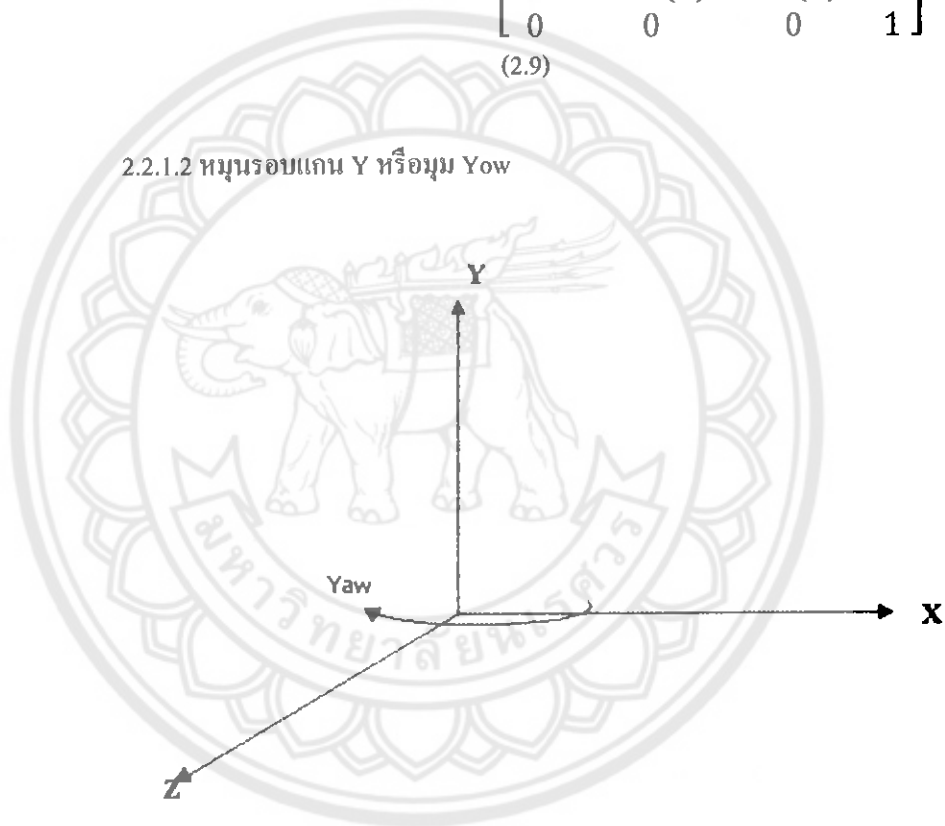
$$Y_1 = Y \times \cos(\theta) - Z \times \sin(\theta) \quad (2.7)$$

$$Z_1 = Y \times \sin(\theta) + Z \times \cos(\theta) \quad (2.8)$$

จากสมการเขียนเป็นรูปแบบเมตริกซ์ได้ ดังนี้

$$[x_1, y_1, z_1, 1] = [x, y, z, 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) & 0 \\ 0 & -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.9)$$

### 2.2.1.2 หมุนรอบแกน Y หรือมุม Yaw



รูปที่ 2.6 การหมุนรอบแกน Y

จะได้สมการ ดังนี้

$$X_1 = Z \times \cos(\theta) - X \times \sin(\theta) \quad (2.10)$$

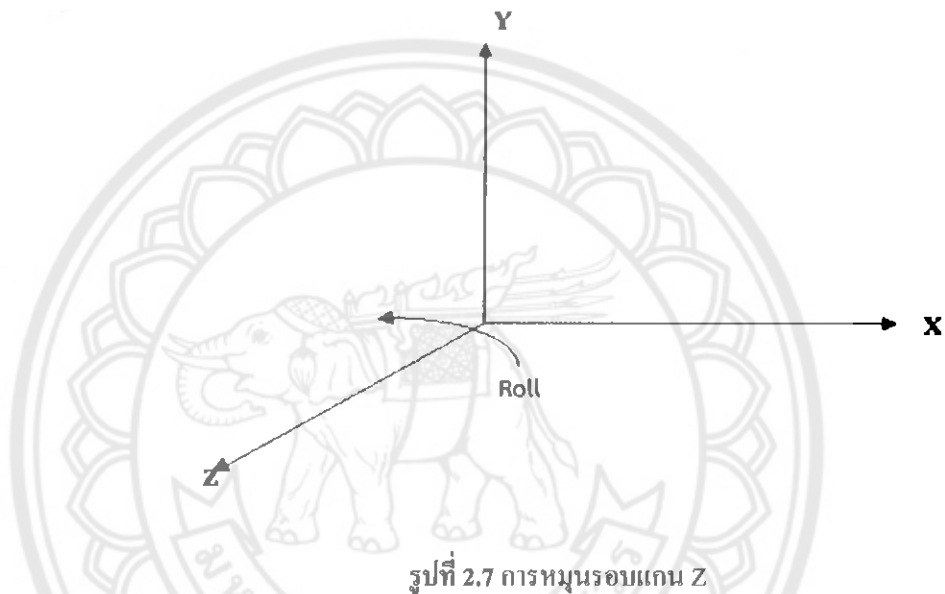
$$Y_1 = Y \quad (2.11)$$

$$Z_1 = Z \times \sin(\theta) + X \times \cos(\theta) \quad (2.12)$$

จากสมการเขียนเป็นรูปแบบเมทริกซ์ได้ ดังนี้

$$[x_1, y_1, z_1, 1] = [x, y, z, 1] \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.13)$$

### 2.2.1.3 หมุนรอบแกน Z หรือมุม Roll



รูปที่ 2.7 การหมุนรอบแกน Z

จะได้สมการ ดังนี้

$$X_1 = X \times \cos(\theta) - Y \times \sin(\theta) \quad (2.14)$$

$$Y_1 = X \times \sin(\theta) + Y \times \cos(\theta) \quad (2.15)$$

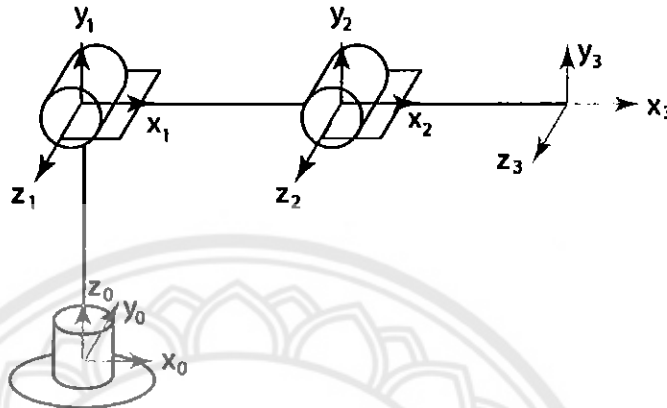
$$Z_1 = Z \quad (2.16)$$

จากสมการเขียนเป็นรูปแบบเมทริกซ์ได้ ดังนี้

$$[x_1, y_1, z_1, 1] = [x, y, z, 1] \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 & 0 \\ -\sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.17)$$

DOF (Degree Of Freedom) DOF มีความหมายได้หลายแบบ ได้แก่

- จำนวนพิกัดที่น้อยที่สุดที่สามารถใช้อธิบายการเคลื่อนที่ทั้งหมดของระบบได้
- จำนวนลักษณะของการเคลื่อนที่ที่เป็นอิสระต่อกัน
- จำนวนตัวเลขหรือตัวแปรที่น้อยที่สุดที่ใช้อธิบายสภาพของระบบ



รูปที่ 2.8 แสดงทิศทางของ Degree of Freedom (DOF) [5]

### 2.3 ทฤษฎีการเคลื่อนที่

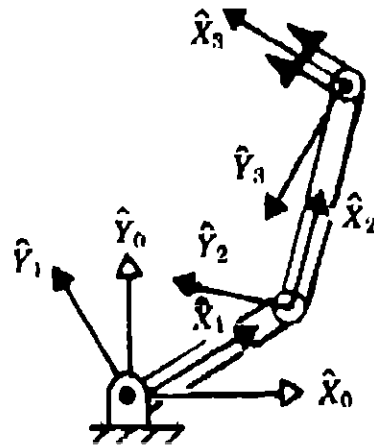
เมื่อนำตำแหน่งที่มีมือจับไปเทียบกับฐานหรือข้อต่อจะสามารถเขียนสมการความสัมพันธ์กันระหว่างมุมของมอเตอร์ กับตำแหน่งปลายแขนหุ่นยนต์  $(x, y, z)$  ความสัมพันธ์นี้แบ่งเป็น 2 ประเภทคือ

#### 2.3.1 Forward kinematics

คือการหาค่าตำแหน่งปลายแขนหุ่นยนต์  $(x, y, z)$  เมื่อกำหนดมุมของมอเตอร์แล้ว ตำแหน่งที่ปลายมือจับไปเทียบกับฐานหรือ joint 1 จะสามารถเขียนสมการความสัมพันธ์กันระหว่าง มุมของแต่ละข้อ กับ ตำแหน่งที่ปลายมือจับ  $(x, y, z)$  อยู่ในรูปแบบเมตริกขนาด  $4 \times 4$  เรียกว่า Compound Transform วิธีการหาค่า  $(x, y, z)$  ของแขนหุ่นยนต์โดยที่รู้ค่ามุมแต่ละข้อ Forward kinematics ใช้ทฤษฎีของ Denavit - Hartenberg เข้ามาช่วย ซึ่งใช้ในการตั้งแกนและหาพารามิเตอร์

#### 2.3.2 Inverse kinematics

คือการหาค่ามุมของมอเตอร์เมื่อกำหนดตำแหน่งปลายแขนหุ่นยนต์  $(x, y, z)$  ในทางกลับกันกับ ฟอว์เวอร์สคิเนมติกส์ (Forward kinematics) ถ้าเราต้องการให้หุ่นยนต์เคลื่อนที่ไปยังตำแหน่ง  $(x, y, z)$  ที่ต้องการ จะต้องใช้วิธีการของอินเวอร์สคิเนมติกส์ (Inverse Kinematics) เป็นวิธีหาว่า  $\theta$ ; ต้องมีค่าใดบ้างที่จะทำให้มือจับเคลื่อนที่ไปยังจุด  $(x, y, z)$  ที่ต้องการ

รูปที่ 2.9 แสดงตัวอย่างของการหา  $\theta_i$  [6]

$i$	$\alpha_{i-1}$	$a_{i-1}$	$d_i$	$\theta_i$
1	0	0	0	$\theta_1$
2	0	$l_1$	0	$\theta_2$
3	0	$l_2$	0	$\theta_3$

ตารางที่ 2.1 แสดงตัวอย่างของการหา  $\theta_i$ 

$$T_{0,3} = \begin{bmatrix} \cos(\theta_1 + \theta_2 + \theta_3) & -\sin(\theta_1 + \theta_2 + \theta_3) & 0 & l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2 + \theta_3) & \cos(\theta_1 + \theta_2 + \theta_3) & 0 & l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.18)$$

$$T_{0,3} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 & x \\ \sin(\phi) & \cos(\phi) & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.19)$$



ตัวอย่าง

Solution Inverse Kinematics

$$W_x = x - (l_3 \times \cos(\phi)) \quad (2.20)$$

$$\phi \text{ คือค่า } \frac{\pi}{3}$$

$$W_y = y - (l_3 \times \sin(\phi)) \quad (2.21)$$

$$\cos(\theta_2) = \frac{(W_x + W_y) - (l_1 - l_2)}{2 \times l_1 \times l_2} \quad (2.22)$$

$$\sin(\theta_2) = \sqrt{1 - (\cos(\theta_2))^2} \quad (2.23)$$

$$\theta_2 = \text{Atan2}(-\sin(\theta_2), \cos(\theta_2)) \quad (2.24)$$

$$\text{atan2}(y, x) = \begin{cases} \arctan\left(\frac{y}{x}\right) & x > 0 \\ \arctan\left(\frac{y}{x}\right) + \pi & y \geq 0, x < 0 \\ \arctan\left(\frac{y}{x}\right) - \pi & y < 0, x < 0 \\ +\frac{\pi}{2} & y > 0, x = 0 \\ -\frac{\pi}{2} & y < 0, x = 0 \\ \text{undefined} & y = 0, x = 0 \end{cases}$$

รูปที่ 2.10 แสดงวิธีการคำนวณสมการ Atan2 [14]

จะได้มุมของข้อที่ 2 โดยมีหน่วยเป็น radian

$$\sin(\theta_1) = \frac{((l_1 + l_2 \times \cos(\theta_2)) \times W_y) - (l_2 \times \sin(\theta_2) \times W_x)}{(W_x + W_y)} \quad (2.25)$$

$$\cos(\theta_1) = \frac{((l_1 + l_2 \times \cos(\theta_2)) \times W_x) - (l_2 \times \sin(\theta_2) \times W_y)}{(W_x + W_y)} \quad (2.26)$$

$$\theta_1 = \text{Atan2}(\sin(\theta_1), \cos(\theta_1)) \quad (2.27)$$

จะได้มุมของข้อที่ 1 โดยมีหน่วยเป็น radian

$$\theta_3 = \phi - \theta_1 - \theta_2 \quad (2.28)$$

จะได้มุมของข้อที่ 3 โดยมีหน่วยเป็น radian

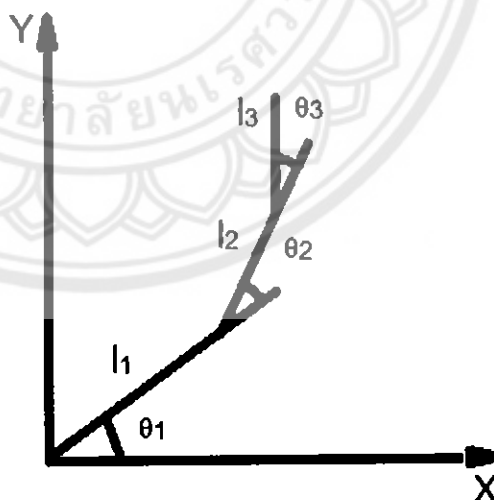
เนื่องจากค่าที่คำนวณได้นั้นมีหน่วยเป็น Radian จึงต้องทำการแปลงค่า Radian เป็น Degree จากสูตร  $r2d = \frac{180}{\pi}$  จากนั้นนำค่า r2d มาคูณเข้ากับค่ามุมที่ได้ในตอนแรก จะได้สมการดังต่อไปนี้

$$\theta_{\text{degree1}} = (\theta_1 \times r2d) \quad (2.29)$$

$$\theta_{\text{degree3}} = (\theta_3 \times r2d) \quad (2.30)$$

$$\theta_{\text{degree2}} = (\theta_2 \times r2d) \quad (2.31)$$

จากขั้นตอนการคำนวณสูตรตามทฤษฎีของอินเวอร์สคิเนมติกส์ จะได้มุมของ ทั้ง 3 ข้อที่มีหน่วยเป็นดีกรี

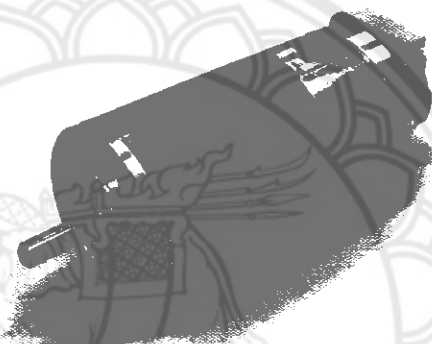


รูปที่ 2.11 แสดงลำดับของมุมและท่อนแขน [7]

สรุปแบบย่อได้ว่า ฟอว์เวอร์สกินเมติกส์นั้นเป็นการคำนวณหาตำแหน่งและทิศทางของมือหุ่นยนต์จากค่าของมุมข้อต่อขนาดของลิงค์ ซึ่งเป็นการแทนค่าตัวแปร (มุมข้อต่อ) เพื่อหาตำแหน่งของหุ่นยนต์ ในทางกลับกันอินเวอร์สกินเมติกส์ เป็นการคำนวณย้อนกลับจากการระบุตำแหน่งและทิศทางของมือหุ่นยนต์แล้วหาค่ามุมของทุกข้อต่อที่ต้องหมุนไป นั้นหมายความว่าเราต้องหาค่ามุมข้อต่อจากตำแหน่งเป้าหมายของหุ่นยนต์

## 2.4 มอเตอร์กระแสตรง (DC Motor)

มอเตอร์กระแสตรงจะมีหลักการทำงาน โดยวิธีการผ่านกระแสให้กับขดลวดในสนามแม่เหล็ก ซึ่งจะทำให้เกิดแรงแม่เหล็ก โดยส่วนของแรงนี้จะขึ้นอยู่กับกระแสและกำลังของสนามแม่เหล็ก



รูปที่ 2.12 มอเตอร์กระแสตรง (DC Motor) [8]

### 2.4.1 หลักการทำงานของมอเตอร์กระแสตรง

การควบคุมความเร็วของมอเตอร์กระแสตรงมีหลายวิธีด้วยกัน ซึ่งอาจจะใช้วิธีการควบคุมแบบพื้นฐานทั่วไป เช่น การควบคุมด้วยวิธีการใช้ตัวต้านทานปรับค่าโดยอัตโนมัติกับมอเตอร์ หรือใช้วิธีการการควบคุมโดยการเปลี่ยนค่าของระดับแรงดันที่ป้อนให้กับมอเตอร์ แต่การควบคุมในวิธีดังกล่าวถึงแม้ว่าจะควบคุมความเร็วมอเตอร์ให้คงที่ได้ แต่ที่ความเร็วต่ำจะส่งผลให้แรงบิดต่ำไปด้วย ดังนั้น เราจึงเลือกใช้วิธีการควบคุมโดยการจ่ายกระแสไฟให้กับมอเตอร์เป็นช่วงๆ โดยอาศัยกระแสไฟที่ป้อนให้กับมอเตอร์ให้เป็นค่าเฉลี่ยที่เกิดขึ้นในแต่ละช่วง ซึ่งเราเรียกว่าวิธีการของ Modulation ทางความกว้างของลูกคลื่น PWM (Pulse Width Modulation)

## 2.5 เซอร์โวมอเตอร์ (Servo Motor)

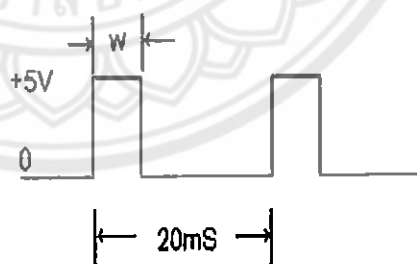
เซอร์โวมอเตอร์ คือมอเตอร์ไฟฟ้ากระแสตรงที่ถูกประกอบด้วยชุดเกียร์และส่วนควบคุมต่างๆ ไว้ในโมดูลเดียวกัน โดยจะมีสัญญาณใช้งาน 1 เส้น และอีก 2 เส้นเป็น VCC และ GND เท่านั้น ซึ่งสามารถควบคุมให้ตัวเซอร์โวมอเตอร์หมุนซ้าย หรือ ขวาได้ +90 องศา - 90 องศา (180 องศา)



รูปที่ 2.13 เซอร์โวมอเตอร์ [9]

### 2.5.1 หลักการทำงานของเซอร์โวมอเตอร์

เซอร์โวมอเตอร์สามารถควบคุมการทำงานได้โดยป้อนสัญญาณความกว้างลูกคลื่น (pulse) ให้กับเซอร์โวมอเตอร์ ซึ่งความกว้างของสัญญาณลูกคลื่นนี้จะไปกำหนดตำแหน่งของเซอร์โวมอเตอร์ว่าจะหมุนไปอยู่ตำแหน่งใด ซึ่งเราสามารถป้อนสัญญาณความกว้างลูกคลื่นได้ตั้งแต่ 1.0 ms - 2.0 ms โดยที่คาบเวลา (Period) ในการส่งสัญญาณความกว้างลูกคลื่นเท่ากับ 20 ms ดังแสดงในรูปที่ 2.4



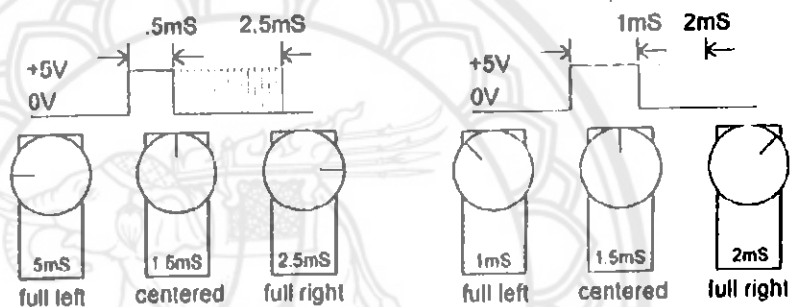
รูปที่ 2.14 สัญญาณลูกคลื่นที่สามารถป้อนให้กับเซอร์โวมอเตอร์ [3]

โดยจะมีจุดที่อ้างอิงตำแหน่ง 3 จุดด้วยกันคือ

1. ตำแหน่งซ้ายสุด เราสามารถกำหนดให้เซอร์ไวโมเตอร์ เคลื่อนที่ไปยังตำแหน่งซ้ายสุด (-90 องศา) ได้โดยการป้อนความกว้างถูกคลื่นเท่ากับ 0.5 ms

2. ตำแหน่งตรงกลาง เราสามารถกำหนดให้เซอร์ไวโมเตอร์ เคลื่อนที่ไปยังตำแหน่งตรงกลาง (0 องศา) ได้โดยการป้อนความกว้างหน้าคลื่นเท่ากับ 1.5 ms

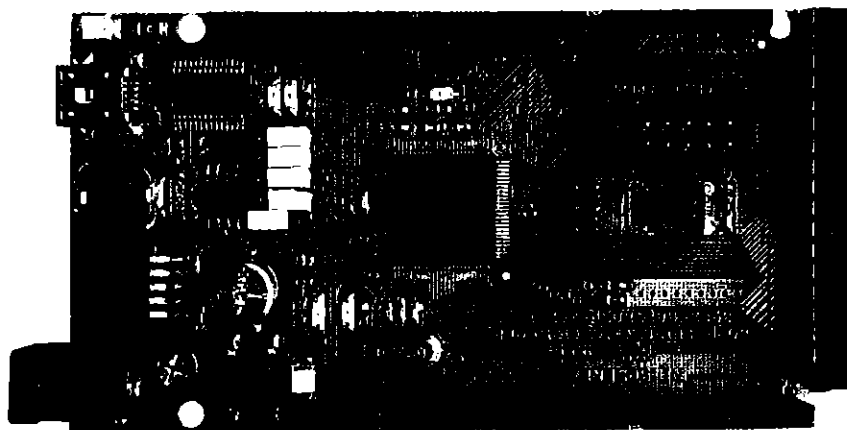
3. ตำแหน่งขวาสุด เราสามารถกำหนดให้เซอร์ไวโมเตอร์ เคลื่อนที่ไปยังตำแหน่งซ้ายสุด (90 องศา) ได้โดยการป้อนความกว้างถูกคลื่นเท่ากับ 2.5 ms จากจุดอ้างอิงตำแหน่งทั้ง 3 ทำให้สามารถคำนวณตำแหน่งที่ต้องการให้หมุนไปได้ เช่นถ้าต้องการเซอร์ไวโมเตอร์หมุนไปยังตำแหน่ง -45 องศา ก็ทำการป้อน  $\frac{(0.5+1.5)}{2} = 1.0$  ms ให้กับเซอร์ไวโมเตอร์



รูปที่ 2.15 แสดงตำแหน่งของเซอร์ไวโมเตอร์เมื่อป้อนความกว้าง pulse [3]

## 2.6 ไมโครคอนโทรลเลอร์ ATmega1280

ไมโครคอนโทรลเลอร์ ATmega1280 เป็นชิพตระกูล AVR มีระบบรองรับโปรแกรมภาษาซีของ Arduino ซึ่งใช้เป็นหน่วยประมวลผลหลัก โดยชิพรุ่นนี้มีหน่วยความจำแฟลชสำหรับเก็บการเขียนโปรแกรม 128 กิโลไบต์ และมีหน่วยความจำแรม 8 กิโลไบต์ มี EEPROM อีก 4 กิโลไบต์สำหรับใช้เป็นที่เก็บข้อมูลถาวรได้ เมื่อไม่มีไฟเลี้ยง และยังมีขาเข้า/ส่งออก (I/O) สำหรับต่อใช้งานทั่วไป 86 ขา มี PWM (Pulse Width Modulation หรือ การปรับความกว้างของพัลส์โดยการนำเอาสองสัญญาณมาเปรียบเทียบกัน) ที่กำหนดความละเอียดได้ระดับ 16 บิตให้ใช้งานถึง 12 ช่องสัญญาณ มีช่องสื่อสารแบบอนุกรม 4 พอร์ต และสามารถแปลงสัญญาณอนาล็อกเป็นดิจิทัล (ADC) อีก 16 ช่องสัญญาณ ซึ่งจะเห็นว่ามีความสามารถพื้นฐานที่มากพอสำหรับงานควบคุมที่หลากหลาย สามารถทำงานได้โดยใช้ชิพเดียว มีคำสั่งที่สนับสนุนการเขียนโปรแกรมควบคุมและสามารถเข้าถึงข้อมูลระดับบิตได้



รูปที่ 2.16 ไมโครคอนโทรลเลอร์ ATmega1280 [10]

#### 2.6.1 คุณสมบัติของไมโครคอนโทรลเลอร์ ATmega1280

- เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิตประสิทธิภาพสูงแต่ใช้พลังงานต่ำในตระกูล AVR สถาปัตยกรรมแบบ RISC (Reduced Instruction Set Computer)
- มีชุดคำสั่ง 135 คำสั่ง และส่วนใหญ่คำสั่งเหล่านี้จะใช้เพียง 1 สัญญาณนาฬิกาในการประมวลผลคำสั่ง
- มีรีจิสเตอร์สำหรับใช้งานทั่วไปขนาด 8 บิต จำนวน 32 ตัว
- ทำงานได้สูงสุดที่ 16 ล้านคำสั่งต่อวินาที (MIPS) เมื่อใช้สัญญาณนาฬิกา 16 เมกะเฮิร์ตซ์ (MHz)
- หน่วยความจำแฟลชสำหรับโปรแกรมขนาด 128 กิโลไบต์เขียนหรือลบได้ 10,000 ครั้ง
- หน่วยความจำแบบ EEPROM ขนาด 4 กิโลไบต์ เขียน/ลบได้ 100,000 ครั้ง
- หน่วยความจำแรมชนิด SRAM ขนาด 8 กิโลไบต์
- มีระบบ โปรแกรมตัวเองอยู่ในตัวชิป
- สามารถทำการอ่านขณะเขียนได้จริงโดยสามารถเลือกการทำงานได้เพื่อความปลอดภัยของซอฟต์แวร์
- มีการเชื่อมประสานกับ JTAG (IEEE std. 1149.1 compliant)

### 2.6.2 คุณสมบัติการเชื่อมต่อกับอุปกรณ์ภายนอก

- มีตัวตั้งเวลาและตัวนับขนาด 8 บิต จำนวน 2 ตัว ที่สามารถแยกโหมดการทำงานจากกันได้ 2 โหมดคือ Prescaler และ Compare
- มีตัวตั้งเวลาและตัวนับขนาด 16 บิต จำนวน 4 ตัว ที่แยกโหมดการทำงานได้ 3 โหมดคือ Prescaler, Compare และ Capture
- มีตัวนับแบบเวลาจริง (Real Time Counter) ที่แยกวงจรกำหนดความถี่ได้
- มี PWM จำนวน 12 ช่องสัญญาณที่สามารถกำหนดความละเอียดได้ 16 บิต
- มีตัวปรับผลการเปรียบเทียบของเอาต์พุต
- มีตัวแปลงสัญญาณอนาล็อกให้เป็นดิจิทัลขนาด 10 บิต จำนวน 16 ช่องสัญญาณ
- มีพอร์ตสื่อสารอนุกรมที่สามารถกำหนดอัตราการรับ/ส่งได้จำนวน 4 พอร์ต
- เชื่อมประสานอนุกรมแบบ SPI ได้ทั้งการเป็น Master และ Slave
- มีการเชื่อมประสานแบบอนุกรมด้วยสายสัญญาณ 2 เส้นแบบส่งข้อมูลแบบเรียงไบนารี (Byte Oriented)
- มีตัวตั้งเวลาแบบ Watchdog ที่สามารถกำหนดการทำงานได้โดยสามารถแยกสัญญาณนาฬิกาได้จากตัวชิพ
- มีตัวเปรียบเทียบสัญญาณแบบอนาล็อกอยู่ในตัว
- มีการรองรับการขัดจังหวะและ Wake-up เมื่อมีการเปลี่ยนแปลงเกิดขึ้นกับขาของชิพ

### 2.6.3 คุณสมบัติพิเศษ

- มีระบบเริ่มระบบเมื่อมีการรีเซ็ตและมีระบบตรวจจับการ Brown-out ที่สามารถกำหนดการทำงานได้
- มีตัวตรวจหาความเที่ยงตรงของออสซิลเลเตอร์อยู่ในตัว (Internal Calibrated Oscillator)
- มีแหล่งการขัดจังหวะทั้งภายในและภายนอก (External and Internal Interrupt Sources)

- มีโหมดการทำงาน sleep 6 แบบ คือ : Idle, ADC Noise Reduction, Power-save, Power-down, Standby, และ Extended Standby

- มีขาของ I/O ที่สามารถกำหนดการทำงานได้ 86 ขา

- ตัวถังแบบ TQFP ชนิด 100 ขา

- ช่วงอุณหภูมิที่ชิพทำงานได้  $-40^{\circ}\text{C}$  ถึง  $85^{\circ}\text{C}$

- โหมดการทำงาน: ที่ 1 MHz ต้องการแรงดัน 1.8V กระแส 500  $\mu\text{A}$

- โหมด Power-down ต้องการกระแสเพียง 0.1  $\mu\text{A}$  ที่แรงดัน 1.8V

บทนี้ได้กล่าวถึงทฤษฎีต่างที่จำเป็นต้องรู้และนำไปใช้ประกอบในการดำเนินโครงการนี้ ในบทต่อไปจะเป็นการดำเนินโครงการว่ามีขั้นตอนอย่างไร



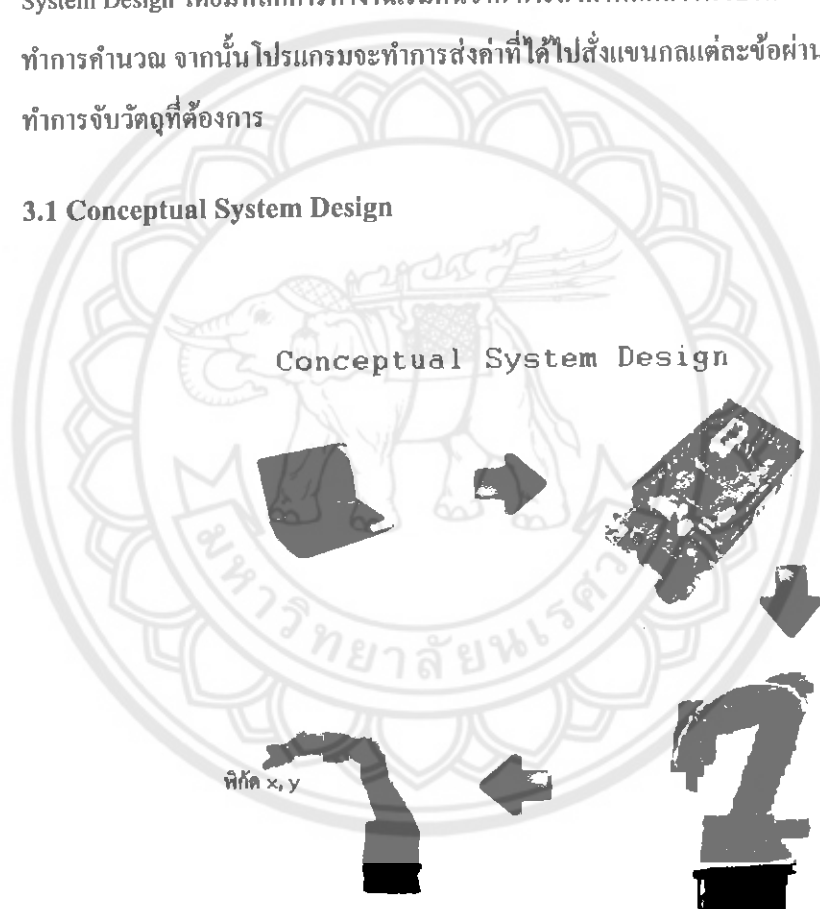


## บทที่ 3

### วิธีการดำเนินโครงการ

จากการศึกษาทฤษฎีจากบทที่ 2 ทำให้ทางผู้จัดทำสามารถวางแผนการดำเนินงานเพื่อพัฒนาหุ่นยนต์แขนกลอัตโนมัติ และเลือกใช้ทฤษฎีของอินเวอร์สคิเนมาติกส์ (Inverse Kinematics) เป็นอัลกอริทึมในการพัฒนาระบบควบคุมแขนกลไปยังพิกัดที่ต้องการ และออกแบบ Conceptual System Design โดยมีหลักการการทำงานเริ่มต้นจาก การส่งค่าพิกัดลงในโปรแกรมในคอมพิวเตอร์เพื่อทำการคำนวณ จากนั้นโปรแกรมจะทำการส่งค่าที่ได้ไปสั่งแขนกลแต่ละข้อผ่านบอร์ด Arduino เพื่อทำการจับวัตถุที่ต้องการ

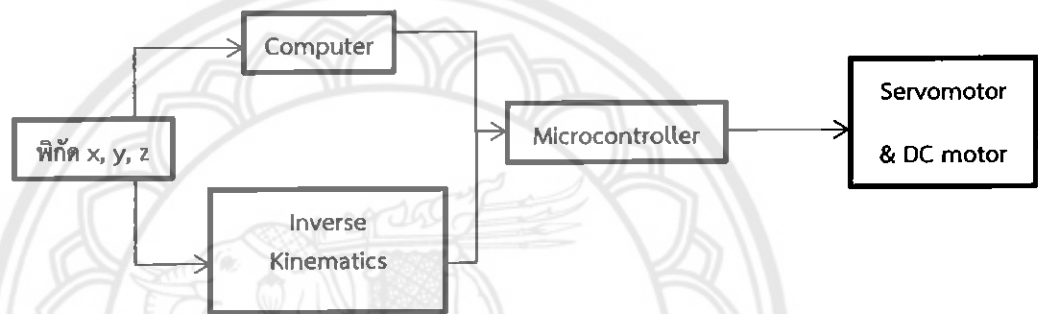
#### 3.1 Conceptual System Design



รูปที่ 3.1 Conceptual System Design

ขั้นตอนการทำงานเบื้องต้นของหุ่นยนต์แขนกลอัตโนมัติ สามารถอธิบายได้ตามไดอะแกรมในรูปที่ 3.2 โดยเริ่มต้นจากการป้อนค่าพิกัดลงในโปรแกรมในคอมพิวเตอร์แล้วโปรแกรมจะใช้อัลกอริทึมอินเวอร์สคิเนมาติกส์ในการคำนวณหาตำแหน่งของแขนในแต่ละข้อจากนั้นจะส่งค่าองศาที่คำนวณได้ไปยังไมโครคอนโทรลเลอร์เพื่อสั่งงานเซอร์โวมอเตอร์ (Servo Motor) และ มอเตอร์กระแสตรง (DC Motor) ทำให้ปลายแขนเคลื่อนไปยังจุดพิกัดที่ต้องการ

### 3.2 Diagram การทำงาน



รูปที่ 3.2 ไดอะแกรมแสดงการทำงานของระบบควบคุมแขนกลอัตโนมัติ

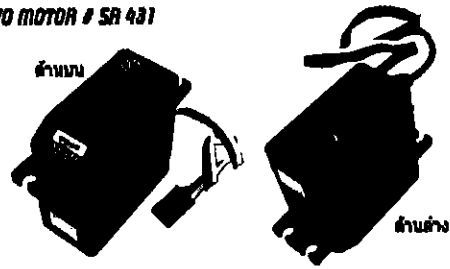
### 3.3 การพัฒนาโครงสร้างของหุ่นยนต์แขนกลอัตโนมัติ

#### 3.3.1 ส่วนประกอบของหุ่นยนต์แขนกลอัตโนมัติ

การพัฒนาโครงสร้างหุ่นยนต์แขนกลอัตโนมัติใช้วัสดุอุปกรณ์ ดังนี้

- Servo Motor รุ่น SR431	2 ตัว
- Servo Motor รุ่น Futaba S3003	1 ตัว
- Servo Motor รุ่น GWS S30T	1 ตัว
- Motor worm gear 12vdc 10RPM	1 ตัว
- กริปเปอร์	1 ชุด
- Potentiometer ชนิด B ขนาด 10 k $\Omega$	1 ชุด

**SERVO MOTOR # SR 431**



ขนาด : 42.0 x 20.5 x 30.5 mm.

น้ำหนัก : 62 g

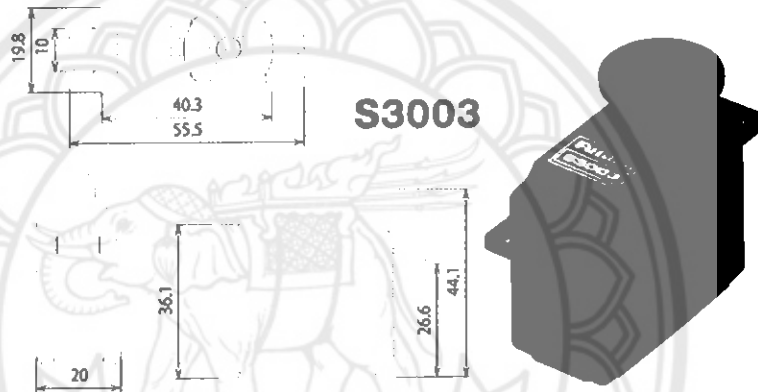
SPEED : 0.2 sec/60° (6.0V), 0.18 sec/60° (7.4V)

TORQUE : 12.2 kg.cm. (6.0V), 14.5 kg.cm. (7.4V)

GEAR : 5 METAL GEAR, 2 BEARING

เป็น SERVO MOTOR แบบหมุน 180 องศา ออกแบบให้ใช้กับงานประกอบหุ่นยนต์โดยมีจุดบัดให้อีกด้านของ SERVO เพื่อความสะดวก ในการหาลงหุ่นยนต์ใน 2 ด้านของตัว SERVO. ระบบ GEAR ภายในเป็นแบบโลหะ. เป็น SERVO แบบ ANALOG

รูปที่ 3.3 รายละเอียดของเซอร์โวมอเตอร์ รุ่น SR431 [11]



รูปที่ 3.4 รูปร่างและขนาดของเซอร์โวมอเตอร์ รุ่น Futaba S3003 [12]



Motor worm gear 12vdc

แกนคือคเมือหยุดจ่ายไฟ

ความเร็ว 10 rpm (รอบต่อนาที)

ขนาดแกน 5 mm.

ขนาดหัวเกียร์ 3 x 4 x 6 cm.

ขนาดมอเตอร์ 3 x 5.5 cm.

รับน้ำหนักได้ไม่เกิน 46kg / cm.

รูปที่ 3.5 รายละเอียดของมอเตอร์กระแสตรง Motor worm gear 12vdc 10RPM [13]

### 3.3.2 ลำดับโครงสร้างของหุ่นยนต์แขนกลอัตโนมัติ

สำหรับ โครงสร้างของหุ่นยนต์แขนกลอัตโนมัตินั้นจะแบ่งได้เป็น 5 ส่วน ดังนี้

ส่วนที่ 1 คือจุดหมุนของลำตัวหุ่นยนต์แขนกลอัตโนมัติใช้เซอร์โวมอเตอร์ รุ่น GWS S30T ในการขับเคลื่อน

ส่วนที่ 2 คือคือจุดหมุนข้อต่อหัวไหล่ของหุ่นยนต์แขนกลอัตโนมัติใช้มอเตอร์ตัวหนอน (Motor worm gear) ในการขับเคลื่อน

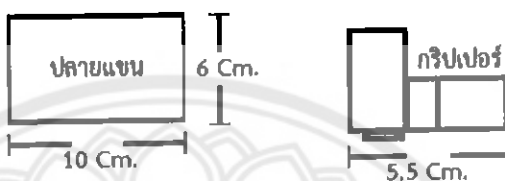
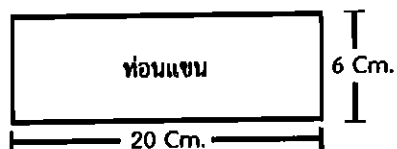
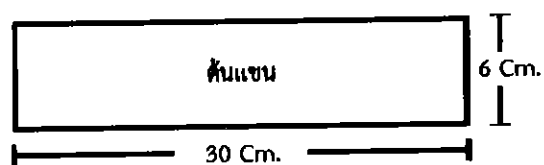
ส่วนที่ 3 คือจุดหมุนข้อศอกของหุ่นยนต์แขนกลอัตโนมัติใช้เซอร์โวมอเตอร์ รุ่น SR431 ในการขับเคลื่อน

ส่วนที่ 4 คือจุดหมุนข้อมือของหุ่นยนต์แขนกลอัตโนมัติใช้เซอร์โวมอเตอร์ รุ่น SR431 ในการขับเคลื่อน

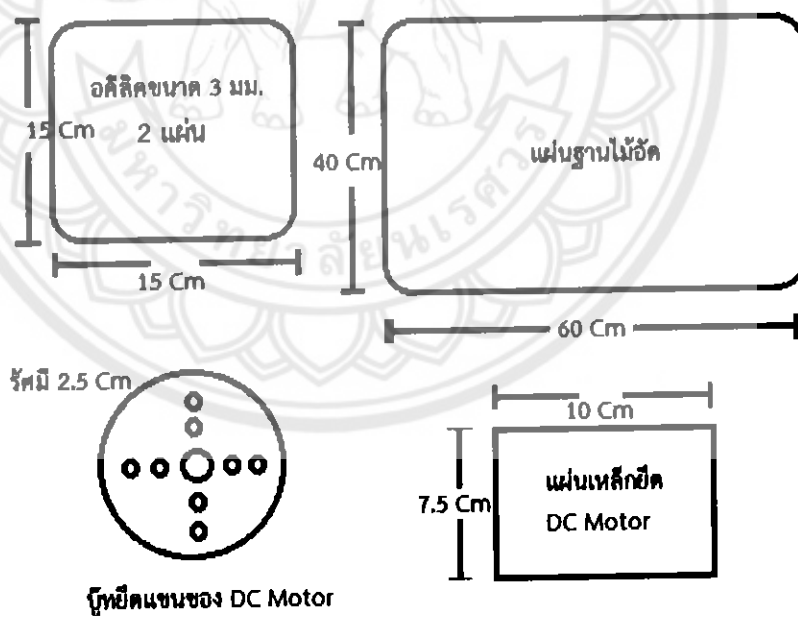
ส่วนที่ 5 คือมือจับซึ่งจะใช้กริปเปอร์ต่อกับเซอร์โวมอเตอร์ รุ่น Fulaba S3003 ให้ทำงานร่วมกันในการจับวัตถุ



รูปที่ 3.6 แสดง โครงสร้างของหุ่นยนต์แขนกล



รูปที่ 3.7 แสดงขนาดของแขนแต่ละ link และตัวกริปเปอร์



รูปที่ 3.8 แสดงขนาดของส่วนประกอบต่างๆของแขนกลอัตโนมัติ

### 3.4 วิธีการใช้งานโปรแกรม Inverse Kinematics

ให้กำหนดพิกัด X, Y, Roll และความยาวของแขนแต่ละท่อน จากนั้นนำมาเข้าสู่สูตรทฤษฎีการคำนวณของอินเวอร์สคิเนแมติกส์โดยกำหนดให้ x, y, roll คือ จุดพิกัดของแกน X, Y, Roll และ  $l_1$ ,  $l_2$  คือความยาวของแขนท่อนที่ 1,  $l_3$  คือความยาวของแขนท่อนที่ 2,  $l_4$  คือความยาวของแขนท่อนที่ 3

#### 3.4.2 อธิบายหน้า Interface ของโปรแกรม

Size link (Cm.)	Input target (Cm.)	Min-Max	Output	Value to Arduino
link1 1	X 4 Cm.	26.5 - 44.4	01 9	Point Value Min-Max
link2 2	Y 5 Cm.	10 - 43.6	02 10	servo1 120 - 200
link3 3	Roll 6 Degree	0 - 180	03 11	servo2 130 - 200 servo3 141 - 180

Buttons: OK, Reset, Run

รูปที่ 3.9 ลักษณะหน้าต่างของโปรแกรม

หมายเลข 1 ความยาวของแขนท่อนที่ 1

หมายเลข 2 ความยาวของแขนท่อนที่ 2

หมายเลข 3 ความยาวของแขนท่อนที่ 3

หมายเลข 4 รับค่า Input พิกัดแกน X

หมายเลข 5 รับค่า Input พิกัดแกน Y

หมายเลข 6 รับค่า Input พิกัดแกน Roll

หมายเลข 7 กดเพื่อสั่งให้ โปรแกรมทำการคำนวณค่าแล้วแสดงค่า Output

หมายเลข 8 กดเพื่อรีเซ็ตค่า Input ทั้งหมด

หมายเลข 9 แสดงค่า Output ของมุม  $\theta_1$

หมายเลข 10 แสดงค่า Output ของมุม  $\theta_2$

หมายเลข 11 แสดงค่า Output ของมุม  $\theta_3$

หมายเลข 12 แสดงค่า Value โปรแกรมส่งไปสั่งการเซอร์โวมอเตอร์ตัวที่ 1

หมายเลข 13 แสดงค่า Value โปรแกรมส่งไปสั่งการเซอร์โวมอเตอร์ตัวที่ 2

หมายเลข 14 แสดงค่า Value โปรแกรมส่งไปสั่งการเซอร์โวมอเตอร์ตัวที่ 3

หมายเลข 15 กดเพื่อสั่งให้โปรแกรมส่งค่าไปยังบอร์ดเพื่อสั่งการแขนกล

เมื่อโปรแกรมได้รับค่าพิกัด X, Y, Roll แล้วนั้น โปรแกรมจะทำการคำนวณผ่านสมการของอินเวอร์สคิเนเมติกส์แล้วจากนั้นจะได้ค่ามุมของแต่ละข้อออกมาจากนั้น โปรแกรมจะนำค่ามุมของข้อแต่ละจุดที่มีหน่วยเป็นดีกรี ส่งต่อไปยังบอร์ดไมโครคอนโทรลเลอร์เพื่อสั่งงานมอเตอร์ส่วนต่างๆเคลื่อนที่ เพื่อให้ได้มุมของข้อแต่ละจุดตามที่คำนวณได้

บทนี้กล่าวถึงวิธีและขั้นตอนต่างๆของการดำเนินงานเพื่อสร้างชิ้นงานและทำความเข้าใจกับตัวโปรแกรมการสั่งการที่พัฒนาขึ้นมา จากนั้นจะนำไปทดสอบในบทต่อไป

## บทที่ 4

### ผลการทดลอง

ในบทนี้จะกล่าวถึงการทดสอบการทำงานของโปรแกรมการคำนวณและแขนกลอัตโนมัติที่พัฒนาขึ้นมา ว่ามีความแม่นยำมากน้อยเพียงใด มีปัญหาหรือข้อผิดพลาด ณ จุดใดเพื่อทำการวิเคราะห์ปัญหาแล้วแก้ไขหรือให้เป็นแนวทางในการที่จะนำพัฒนาต่อไป

#### 4.1 จุดประสงค์ของการทดลอง

4.1.1 เพื่อทดสอบ โปรแกรมที่พัฒนาขึ้นมาว่าสามารถทำงานได้หรือไม่

4.1.2 เพื่อทดสอบการสั่งงานจากเครื่องคอมพิวเตอร์ว่าอัลกอริทึมที่ใช้มีความแม่นยำเพียงใด และมีความคลาดเคลื่อนเท่าไร

4.1.3 เพื่อทดสอบการเคลื่อนที่ของแขนกลในส่วนข้อต่อต่างๆว่าสามารถทำงานได้ตามคำสั่งของผู้ใช้ได้หรือไม่

4.1.4 เพื่อทดสอบหาข้อผิดพลาดที่เกิดขึ้นกับ โปรแกรมที่พัฒนาขึ้นและแขนกล พร้อมทั้งนำข้อผิดพลาดมาทำการวิเคราะห์ถึงปัญหา การปรับปรุงแก้ไขข้อผิดพลาดที่เกิดขึ้น และแนวทางในการพัฒนาต่อไป

#### 4.2 ขั้นตอนการทดลอง

4.2.1 จัดเตรียมการทดลองโดยนำอัลกอริทึมมาแปลงเป็น โค้ด (Code) ลงในโปรแกรม C# เพื่อพัฒนาแล้วส่งค่าที่คำนวณได้ไปยัง บอร์ดไมโครคอนโทรลเลอร์ (Arduino ATmega1280)

4.2.2 นำแขนกลอัตโนมัติมาต่อเข้ากับบอร์ดไมโครคอนโทรลเลอร์ที่เชื่อมต่อเครื่องคอมพิวเตอร์โดยผ่านพอร์ตอนุกรม (USB)

4.2.3 ตรวจสอบข้อผิดพลาดที่เกิดขึ้นกับ โปรแกรมที่พัฒนาขึ้นและแขนกลอัตโนมัติขณะทำการทดสอบ จากนั้นนำข้อผิดพลาดมาทำการวิเคราะห์ถึงปัญหา การปรับปรุงแก้ไขข้อผิดพลาดที่เกิดขึ้น และแนวทางในการพัฒนาต่อไป



### 4.3 ขั้นตอนทดสอบโปรแกรม

โปรแกรมที่ใช้ทางผู้จัดทำได้พัฒนาขึ้นมาเองโดยพัฒนาด้วยภาษา C# ในการพัฒนาขึ้นมา หน้าตาของโปรแกรมมีลักษณะดังรูปที่ 4.1

Size link (Cm.)	Input target (Cm.)	Min-Max	Output	Value to Arduino
link1	X	Cm. 26.5 - 44.4	01	Point Value Min-Max
link2	Y	Cm. 10 - 43.6	02	servo1 0 - 200
link3	Roll	Degree 0 - 180	03	servo2 0 - 200 servo3 1 - 180

Buttons: OK, Reset, Run

รูปที่ 4.1 แสดงตัวโปรแกรมที่พัฒนาขึ้นมา

การทดสอบโปรแกรมจะทำได้โดยการกรอกค่าพิกัดแกน X, Y, Z ลงในโปรแกรมเพื่อให้โปรแกรมทำการคำนวณหาค่ามุมแต่ละจุดแล้วแสดงผลพิกัดที่ได้ จากนั้นนำผลลัพธ์ที่ได้ในแต่ละมุมมาวาดลงในกระดาษกราฟในอัตราส่วนที่สมจริงเพื่อดูความถูกต้องของการคำนวณของโปรแกรม

### การทดสอบครั้งที่ 1 ดังรูปที่ 4.2

IK Solution									
Size link (Cm.)	Input target (Cm.)		Min-Max	Output	Value to Arduino				
link1	X	10.0 Cm.	26.5 - 44.4	01	Point	Value	Min-Max		
link2	Y	10.0 Cm.	10 - 43.6	02	servo1		0 - 200		
link3	Roll	90.0 Degree	0 - 180	03	servo2		0 - 200		
					servo3		1 - 180		

รูปที่ 4.2 การทดสอบโปรแกรมครั้งที่ 1

จากการทดสอบจะเห็นได้ว่า โปรแกรมไม่สามารถคำนวณได้เนื่องจากเกินขอบเขตของการเคลื่อนที่



การทดสอบครั้งที่ 2 ดังรูปที่ 4.3

IK Solution				
Size link (Cm.)	Input target (Cm.)	Min-Max	Output	Value to Arduino
link1	X 20.0 Cm.	26.5 - 44.4	01	Point Value Min-Max servo1 0 - 200
link2	Y 10.0 Cm.	10 - 43.6	02	servo2 0 - 200
link3	Roll 90.0 Degree	0 - 180	03	servo3 1 - 180

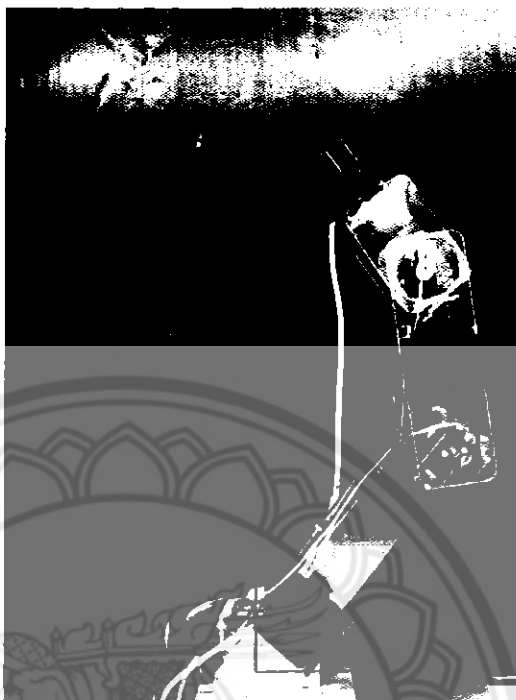
รูปที่ 4.3 การทดสอบ โปรแกรมครั้งที่ 2

จะได้กราฟดังรูปที่ 4.4



รูปที่ 4.4 แสดงกราฟการทำมุมของแขนและปลายแขนของการทดสอบครั้งที่ 2

## แผนกการทำมดรูป 4.5



รูปที่ 4.5 แสดงลักษณะของแผนกของการทดสอบครั้งที่ 2

จากการเปรียบเทียบตำแหน่งที่ติดตั้งของปลายแขน ระหว่างกราฟที่วาดจากองศาที่ได้กับการเคลื่อนที่จริงของแขนกลทำให้รู้ว่าพิกัดที่ให้ไปเกินขอบเขตของฮาร์ดแวร์จึงทำให้เกิดความผิดพลาดในการเคลื่อนที่

### การทดสอบครั้งที่ 3 ดังรูปที่ 4.6

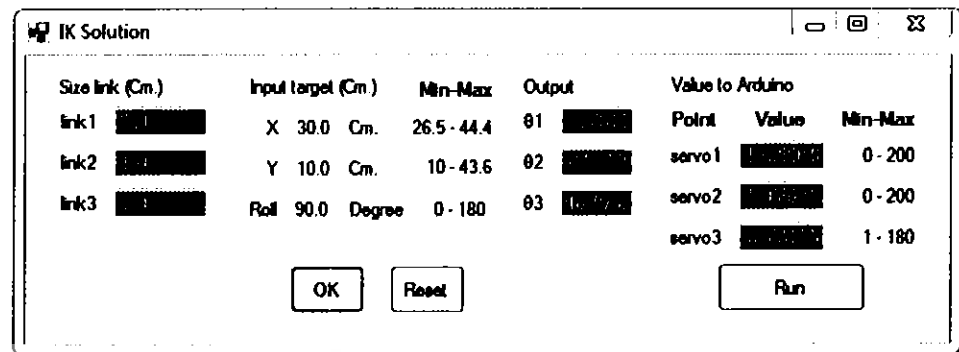
IK Solution									
Size link (Cm.)	Input target (Cm.)	Min-Max	Output	Value to Arduino					
link1	X 10.0 Cm.	26.5 - 44.4	01	Point	Value	Min-Max			
link2	Y 20.0 Cm.	10 - 43.6	02	servo1		0 - 200			
link3	Roll 90.0 Degree	0 - 180	03	servo2		0 - 200			
				servo3		1 - 180			

รูปที่ 4.6 การทดสอบ โปรแกรมครั้งที่ 3

จากการทดสอบจะเห็นได้ว่า โปรแกรมไม่สามารถคำนวณได้เนื่องจากเกินขอบเขตของการเคลื่อนที่



การทดสอบครั้งที่ 4 ดังรูปที่ 4.7



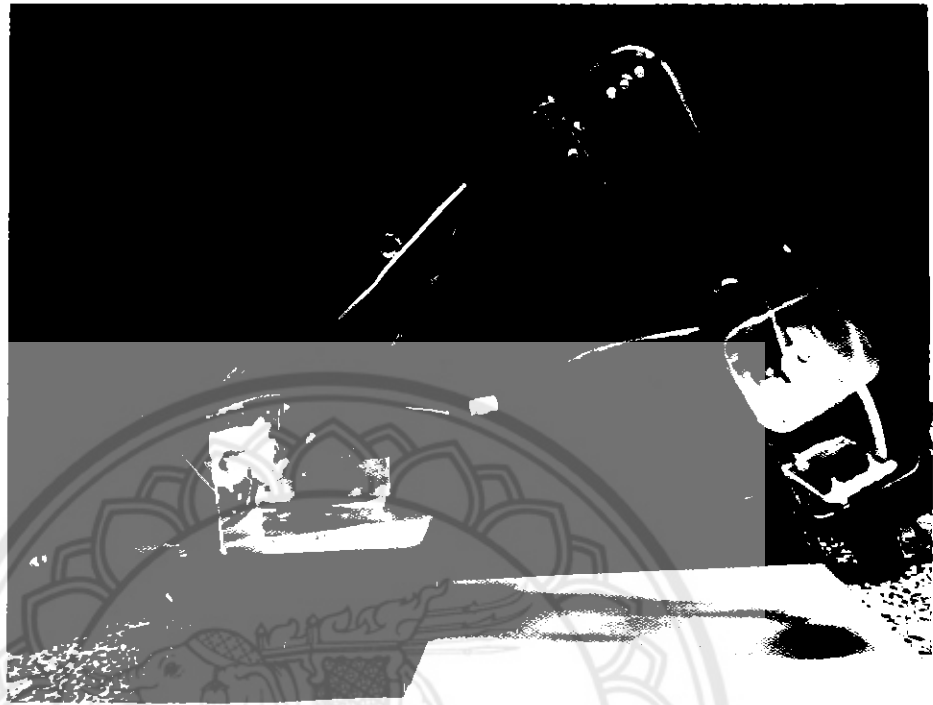
รูปที่ 4.7 การทดสอบโปรแกรมครั้งที่ 4

จะได้กราฟดังรูปที่ 4.8



รูปที่ 4.8 แสดงกราฟการทำมุมของแขนและปลายแขนของการทดสอบครั้งที่ 4

แผนกลทำมุดังรูป 4.9



รูปที่ 4.9 แสดงลักษณะของแผนกลของการทดสอบครั้งที่ 4

จากการเปรียบเทียบตำแหน่งพิกัดของปลายแขนอยู่ ระหว่างกราฟที่วาดจากองศาที่ได้กับการเคลื่อนที่จริงของแขนกลนั้นมีความใกล้เคียงกันมากมีเพียงความคลาดเคลื่อน 3 - 4 เซนติเมตร

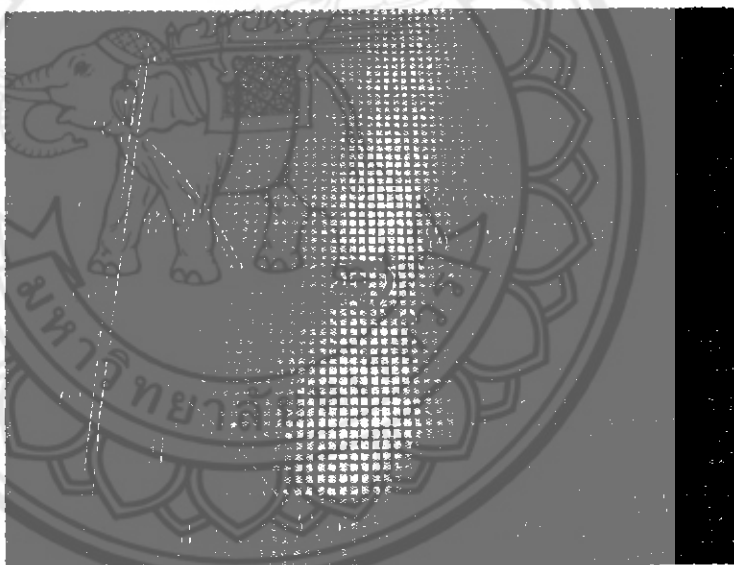
การทดสอบครั้งที่ 5 ดังรูปที่ 4.10

Size link (Cm.)	Input target (Cm.)	Min-Max	Output	Value to Arduino
link1	X 30.0 Cm.	26.5 - 44.4	01	Point Value Min-Max
link2	Y 20.0 Cm.	10 - 43.6	02	servo1 0 - 200
link3	Roll 90.0 Degree	0 - 180	03	servo2 0 - 200
				servo3 1 - 180

OK    Reset    Run

รูปที่ 4.10 การทดสอบโปรแกรมครั้งที่ 5

จะได้กราฟดังรูปที่ 4.11



รูปที่ 4.11 แสดงกราฟการทำมุมของแขนและปลายแขนของการทดสอบครั้งที่ 5



## แกนกลทำนบคังรูป 4.12



รูปที่ 4.12 แสดงลักษณะของแกนกลของการทดสอบครั้งที่ 5

จากการเปรียบเทียบตำแหน่งพิกัดของปลายแกนอยู่ ระหว่างกราฟที่วาดจากองศาที่ได้กับการเคลื่อนที่จริงของแกนกลนั้นมีความใกล้เคียงกันมากมีเพียงความคลาดเคลื่อนเพียง 2 - 3 เซนติเมตร

การทดสอบครั้งที่ 6 ดังรูปที่ 4.13

IK Solution				
Size link (Cm.)	Input target (Cm.)	Min-Max	Output	Value to Arduino
link1	X 30.0 Cm.	26.5 - 44.4	81	Point Value Min-Max
link2	Y 30.0 Cm.	10 - 43.6	82	servo1 0 - 200
link3	Roll 90.0 Degree	0 - 180	83	servo2 0 - 200
				servo3 1 - 180

OK    Reset    Run

รูปที่ 4.13 การทดสอบ โปรแกรมครั้งที่ 6

จะได้กราฟดังรูปที่ 4.14



รูปที่ 4.14 แสดงกราฟการทำมุมของแขนและปลายแขนของการทดสอบครั้งที่ 6

## แขนกลทำมุดังรูป 4.15



รูปที่ 4.15 แสดงลักษณะของแขนกลของการทดสอบครั้งที่ 6

จากการเปรียบเทียบตำแหน่งพิกัดของปลายแขนอยู่ ระหว่างกราฟที่วาดจากองศาที่ได้กับการเคลื่อนที่จริงของแขนกลนั้นมีความใกล้เคียงกันมากมีเพียงความคลาดเคลื่อน 3 - 4 เซนติเมตร

การทดสอบครั้งที่ 7 ดังรูปที่ 4.16

IK Solution				
Size link (Cm.)	Input target (Cm.)	Min-Max	Output	Value to Arduino
link1	X 40.0 Cm.	26.5 - 44.4	01	Point Value Min-Max
link2	Y 10.0 Cm.	10 - 43.6	02	servo1 0 - 200
link3	Roll 90.0 Degree	0 - 180	03	servo2 0 - 200
				servo3 1 - 180

OK    Reset    Run

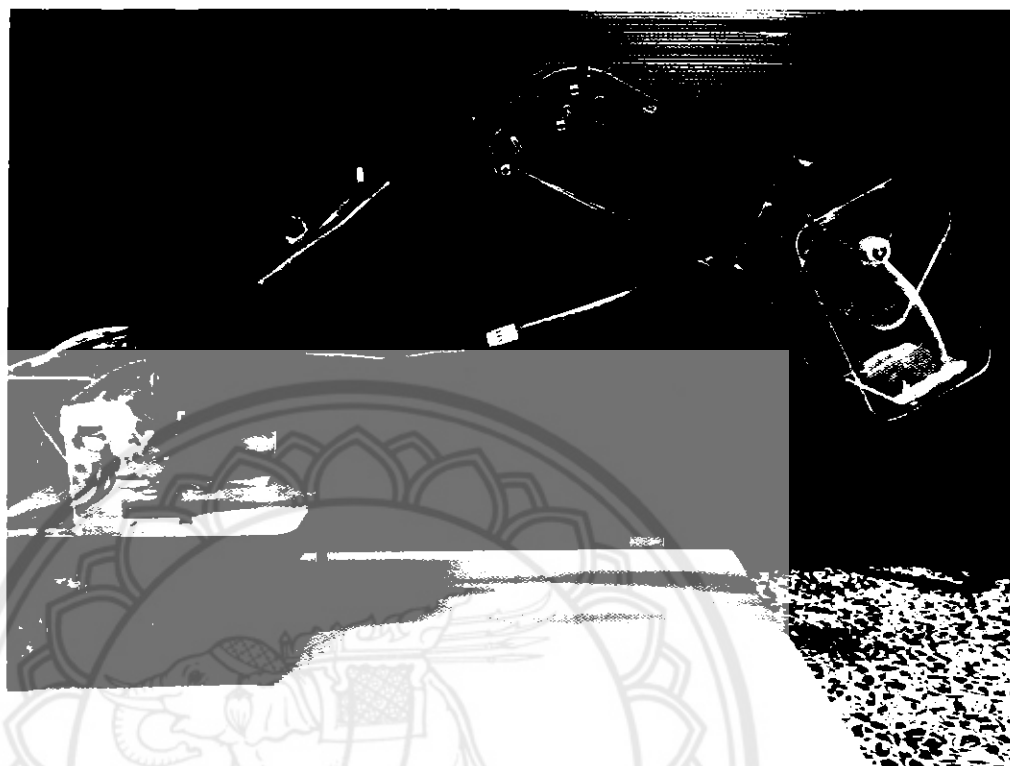
รูปที่ 4.16 การทดสอบ โปรแกรมครั้งที่ 7

จะได้กราฟดังรูปที่ 4.17



รูปที่ 4.17 แสดงกราฟการทำมุมของแขนและปลายแขนของการทดสอบครั้งที่ 7

แขนกลทำมุมดังรูป 4.18



รูปที่ 4.18 แสดงลักษณะของแขนกลของการทดสอบครั้งที่ 7

จากการเปรียบเทียบตำแหน่งพิกัดของปลายแขนอยู่ ระหว่างกราฟที่วาดจากองศาที่ได้กับการเคลื่อนที่จริงของแขนกลนั้นมีความใกล้เคียงกันมากมีเพียงความคลาดเคลื่อนเล็กน้อยเพียง 1 - 3 เซนติเมตรเท่านั้น

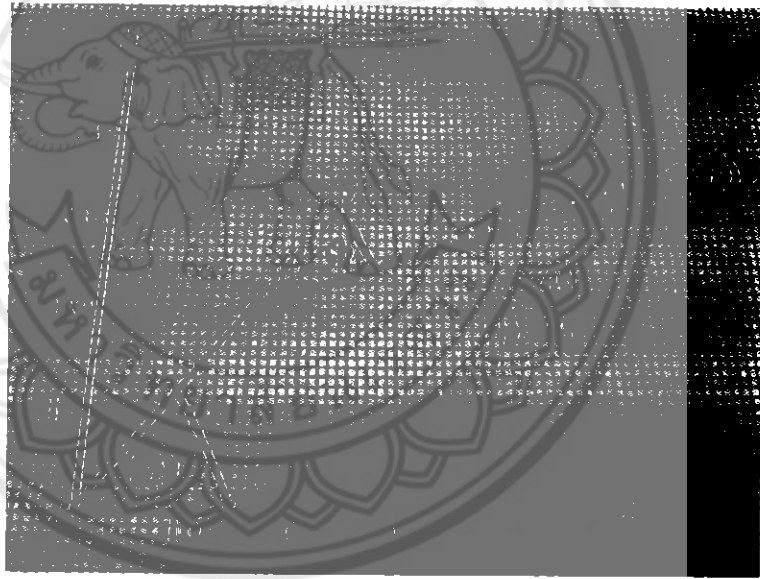
การทดสอบครั้งที่ 8 ดังรูปที่ 4.19

IK Solution									
Size link (Cm.)	Input target (Cm.)	Min-Max	Output	Value to Arduino					
link1	X 40.0 Cm.	26.5 - 44.4	01	Point	Value	Min-Max			
link2	Y 30.0 Cm.	10 - 43.6	02	servo1	0 - 200				
link3	Roll 90.0 Degree	0 - 180	03	servo2	0 - 200				
				servo3	1 - 180				

OK    Reset    Run

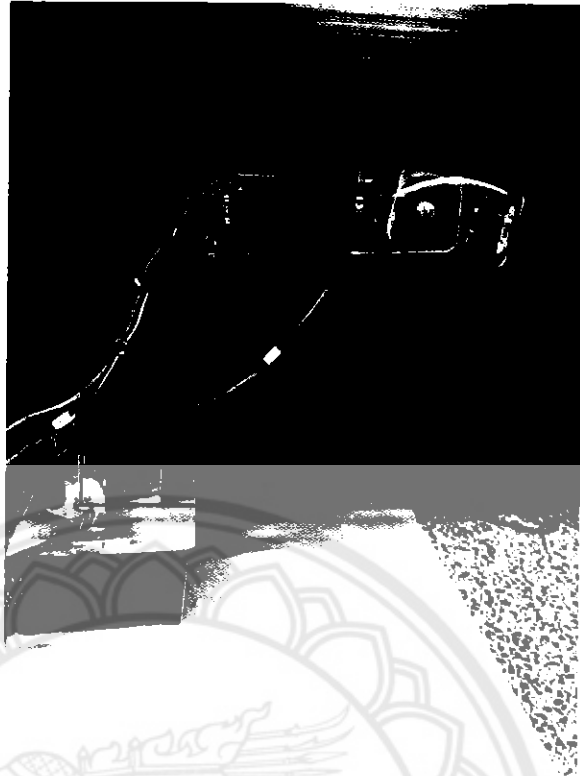
รูปที่ 4.19 การทดสอบ โปรแกรมครั้งที่ 8

จะได้กราฟดังรูปที่ 4.20



รูปที่ 4.20 แสดงกราฟการทำมุมของแขนและปลายแขนของการทดสอบครั้งที่ 8

แขนกลทำมุมดังรูป 4.21



รูปที่ 4.21 แสดงลักษณะของแขนกลของการทดสอบครั้งที่ 8

จากการเปรียบเทียบตำแหน่งที่กักของปลายแขน ระหว่างกราฟที่วาดจากองศาที่ได้กับการเคลื่อนที่จริงของแขนกลนั้นมีความใกล้เคียงกันมากมีเพียงความคลาดเคลื่อนเพียง 1 - 3 เซนติเมตรเท่านั้น

การทดสอบครั้งที่ 9 ดังรูปที่ 4.22

IK Solution									
Size link (Cm.)	Input target (Cm.)	Min-Max	Output	Value to Arduino					
link1	X 40.0 Cm.	26.5 - 44.4	01	Point	Value	Min-Max			
link2	Y 40.0 Cm.	10 - 43.6	02	servo1		0 - 200			
link3	Roll 90.0 Degree	0 - 180	03	servo2		0 - 200			
				servo3		1 - 180			

รูปที่ 4.22 การทดสอบ โปรแกรมครั้งที่ 9

จากการทดสอบจะเห็นได้ว่าโปรแกรมไม่สามารถคำนวณได้เนื่องจากเกินขอบเขตของการเคลื่อนที่





การทดสอบครั้งที่ 10 ดังรูปที่ 4.23

Size link (Cm.)	Input target (Cm.)	Min-Max	Output	Value to Arduino
link1	X -40.0 Cm.	26.5 - 44.4	01	Point Value Min-Max
link2	Y 10.0 Cm.	10 - 43.6	02	servo1 0 - 200
link3	Roll 90.0 Degree	0 - 180	03	servo2 0 - 200
				servo3 1 - 180

OK    Reset    Run

รูปที่ 4.23 การทดสอบ โปรแกรมครั้งที่ 10

จากการทดสอบจะเห็นได้ว่าโปรแกรมไม่สามารถคำนวณได้เนื่องจากเกินขอบเขตของการเคลื่อนที่



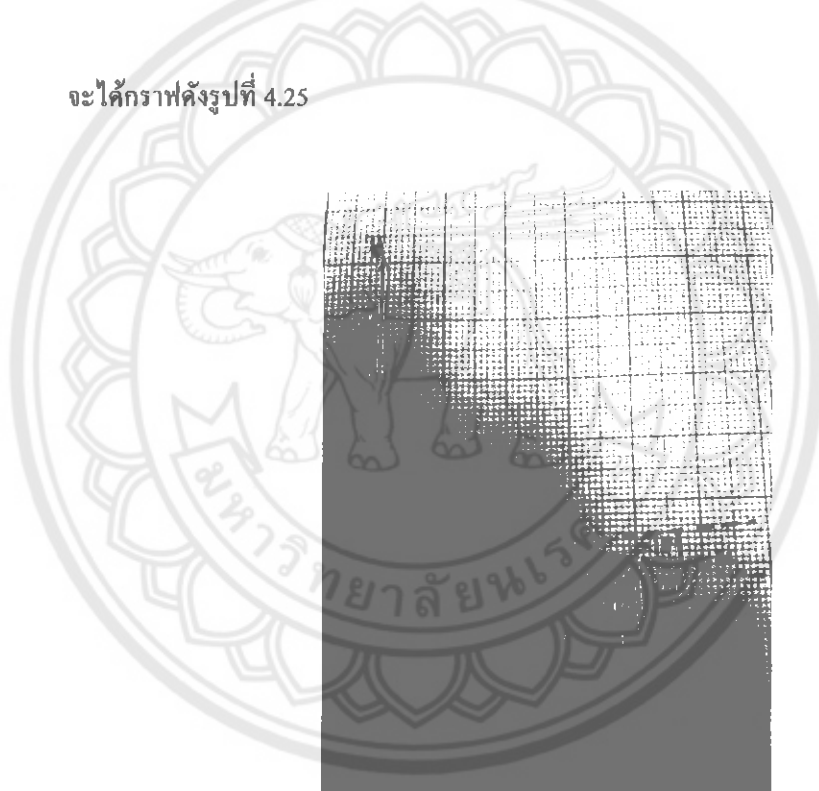
การทดสอบครั้งที่ 11 ดังรูปที่ 4.24

Size link (Cm.)	Input target (Cm.)	Min-Max	Output	Value to Arduino
link1	X 40.0 Cm.	26.5 - 44.4	81	Point Value Min-Max
link2	Y -10.0 Cm.	10 - 43.6	82	servo1 0 - 200
link3	Roll 90.0 Degree	0 - 180	83	servo2 0 - 200
				servo3 1 - 180

OK    Reset    Run

รูปที่ 4.24 การทดสอบ โปรแกรมครั้งที่ 11



จะได้กราฟดังรูปที่ 4.25







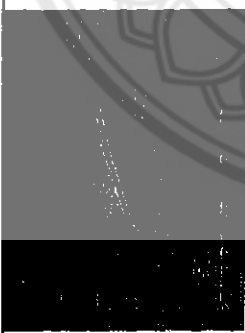

รูปที่ 4.25 แสดงกราฟการทำงานของแขนและปลายแขนของการทดสอบครั้งที่ 11


จากการทดสอบนี้จะเห็นว่ากราฟที่ได้ตกลงไปในแกนลบของแกน Y และมีความคลาดเคลื่อนประมาณ 2 เซนติเมตรจากตำแหน่งพิกัดที่ต้องการ การทดลองนี้ไม่สามารถเปรียบเทียบกราฟกับการเคลื่อนที่จริงของแขนกลได้ เนื่องจากอาจทำให้แขนกลเกิดการกระแทกกับพื้นผิและเกิดการชำรุดเสียหายได้

#### 4.4 ตารางเปรียบเทียบผลการทดลอง

ลำดับการทดลอง	รูปภาพ	รูปแทนผล	Input ค่าพิกัด	ค่ามุมที่คำนวณได้ คำนวณได้	ทำได้ /ไม่ได้	หมายเหตุ
1	-	-	X=10 Y=10	$\theta_1 = 49.7$ $\theta_2 = -158$ $\theta_3 = 43.3$	ไม่ได้	เนื่องจากค่าพิกัด X มีค่าต่ำกว่าขอบเขตของการทำงานของฮาร์ดแวร์
2			X=20 Y=10	$\theta_1 = 86$ $\theta_2 = -152$ $\theta_3 = 59.7$	ไม่ได้	เนื่องจากค่าพิกัด X, Y มีค่าต่ำกว่าขอบเขตของการทำงาน ทำให้ค่าที่ส่งไปยังบอร์ดมีค่าผิดเพี้ยนจากขอบเขต
3	-	-	X=10 Y=20	$\theta_1 = -$ $\theta_2 = -$ $\theta_3 = -$	ไม่ได้	เนื่องจากค่าพิกัด X มีค่าต่ำกว่าขอบเขตของการทำงาน

4		X=30 Y=10	$\theta_1 = 73.6$ $\theta_2 = -113$ $\theta_3 = 16.4$	ทำได้	-
5		X=30 Y=20	$\theta_1 = 91$ $\theta_2 = -107$ $\theta_3 = 23.5$	ทำได้	-
6		X=30 Y=30	$\theta_1 = 67.5$ $\theta_2 = -85.9$ $\theta_3 = 48.3$	ทำได้	-
7		X=40 Y=10	$\theta_1 = 54$ $\theta_2 = -68.5$ $\theta_3 = 17.3$	ทำได้	-

8			X=40 Y=30	$\theta_1 = 52.6$ $\theta_2 = -33.7$ $\theta_3 = -5.1$	ทำได้	-
9	-	-	X=40 Y=40	$\theta_1 = -$ $\theta_2 = -$ $\theta_3 = -$	ไม่ได้	เนื่องจากค่าพิทัก X, Y มี ค่าสูงกว่าขอบเขตของ การทำงาน
10	-	-	X=(-40) Y=10	$\theta_1 = -$ $\theta_2 = -$ $\theta_3 = -$	ไม่ได้	เนื่องจากค่าพิทัก X มีค่า ต่ำกว่าขอบเขตของการ ทำงาน

11			$X=40$ $Y=(-10)$	$\theta_1 = 6.5$ $\theta_2 = -15$ $\theta_3 = -79.5$	ไม่ได้	เนื่องจากการคำนวณนี้ อาจทำให้หุ่นยนต์เงิน ก่อเกิดความเสียหายได้ จึงไม่สามารถสั่งให้ หุ่นยนต์แขนกลทำงาน ได้
----	---	--	---------------------	--	--------	--

ตารางที่ 4.1 ตารางเปรียบเทียบผลการทดลอง

จากที่ได้ทำการทดสอบระบบที่ได้พัฒนาขึ้นมา โดยดูจากตารางเปรียบเทียบการทดสอบระหว่างการทำทดลองตำแหน่งและองศาของแขนกลที่โปรแกรมคำนวณได้กับการเคลื่อนจริงของแขนกล แขนกลสามารถเคลื่อนที่ไปยังพิกัดตำแหน่งที่ต้องการได้แต่ยังมีความคลาดเคลื่อนอยู่ประมาณ 0 – 4 เซนติเมตร และบางพิกัดที่แขนกลไม่สามารถเคลื่อนที่ไปได้และโปรแกรมไม่สามารถคำนวณได้ด้วยเช่นกัน

จากการทดลองในบทนี้ได้ทำการบันทึกเก็บข้อมูลต่างๆ ทำให้พบปัญหาต่างๆที่เกิดขึ้น ในบทต่อไปจะกล่าวถึงปัญหาที่พบจากการทดลอง การแก้ปัญหาและแนวทางในการพัฒนาต่อ

## บทที่ 5

### สรุปผลและข้อเสนอแนะ

ในบทนี้จะกล่าวถึงการสรุปผลการทดลองและปัญหาที่พบจากการทดสอบของโครงการการควบคุมแขนกลด้วยหลักการอินเวอร์สคิเนมาติกส์ (Inverse Kinematics) และแนวทางในการแก้ปัญหาค้างกล่าวที่ได้จากการทดลองที่ 4 รวมถึงแนวทางในการพัฒนาต่อ

#### 5.1 สรุปผลการทดลอง

จากการจัดทำโครงการการควบคุมแขนกลด้วยหลักการอินเวอร์สคิเนมาติกส์ ผลที่ได้คือ ได้มีการพัฒนาหุ่นยนต์แขนกลและวงจรรีบอร์คไมโครคอนโทรลเลอร์และการพัฒนาโปรแกรมที่ใช้รับค่าพิกัดที่ต้องการและสั่งการไปยังบอร์คไมโครคอนโทรลเลอร์เพื่อสั่งให้แขนกลเคลื่อนที่ไปยังพิกัดที่ต้องการ

จากการทดสอบการควบคุมแขนกลด้วยหลักการอินเวอร์สคิเนมาติกส์ ตัวโปรแกรมสามารถรับค่าพิกัดได้และสามารถแสดงค่าของมุมแต่ละข้อที่คำนวณได้ แขนกลทุกข้อรับคำสั่งและสามารถเคลื่อนที่ได้ถูกต้อง ทำให้แขนกลเคลื่อนที่ไปยังจุดหมายได้อย่างถูกต้องจากการระบุค่าตำแหน่งพิกัด X, Y, Roll แต่ยังคงพบว่ามี ความคลาดเคลื่อนอยู่ และบางตำแหน่งพิกัดปลายแขนไม่สามารถไปได้

#### 5.2 ปัญหาที่พบ

##### 5.2.1 ปัญหาการเคลื่อนที่

1. เมื่อแขนกลเคลื่อนที่ไปยังจุดพิกัดที่ต้องการ แขนกลท่อนที่ 2 และ 3 มีการเกิดโอเวอร์ชู้ต (Overshoot) ซึ่งอาจทำให้ตัวของแขนกลเซอร์โวมอเตอร์ชำรุดเสียหายได้และเมื่อใช้งานกับของที่เปราะบางอาจทำให้สิ่งของแตกหักได้

2. จากการทดสอบการควบคุมแขนกลด้วยหลักการอินเวอร์สคิเนมาติกส์ แขนกลไม่สามารถเคลื่อนที่ไปยังพิกัดแกนที่ลบได้

3. มีความคลาดเคลื่อนตั้งแต่ 0 ถึง 4 เซนติเมตร โดยเฉพาะอย่างยิ่งเมื่อแกน Y ต่ำกว่า 10 เซนติเมตร

### 5.2.2 ปัญหาฮาร์ดแวร์

1. ชีตจำกัดของฮาร์ดแวร์ทำให้แขนกลเคลื่อนที่ไปยังพิกัดบางตำแหน่งไม่ได้ และบางจุดสามารถไปได้แต่อาจทำให้แขนกลและเซอร์โวมอเตอร์เกิดการชำรุดเสียหายได้

2. เซอร์โวมอเตอร์ไม่สามารถใช้งานในระยะเวลาอันได้ ซึ่งการใช้งานนานจะให้เซอร์โวมอเตอร์ร้อนแล้วเกิดการชำรุดเสียหายได้

## 5.3 แนวทางการแก้ปัญหา

### 5.3.1 การแก้ปัญหากลเคลื่อนที่

1. เปลี่ยนเซอร์โวมอเตอร์เป็นเซอร์โวมอเตอร์ที่มีประสิทธิภาพที่สูงกว่าตัวเดิม
2. เลือกใช้งานกับวัสดุที่ไม่เปราะบางหรือไม่แตกง่าย
3. กำหนด Working Area ของการเคลื่อนที่
4. ในการส่งค่าจากตัวโปรแกรม C# ไปยังบอร์ดไมโครคอนโทรลเลอร์ให้ส่งเลขทศนิยมไปด้วย

### 5.3.2 การแก้ปัญหาฮาร์ดแวร์

1. เปลี่ยนเซอร์โวมอเตอร์เป็นเซอร์โวมอเตอร์ที่มีประสิทธิภาพที่สูงกว่าตัวเดิม
2. กำหนดขอบเขตการทำงาน (Working Area) ของการเคลื่อนที่
3. จำกัดเวลาหรือค้ใช้งานเป็นเวลานาน
4. คอยตรวจสอบว่าเซอร์โวมอเตอร์ร้อนหรือไม่อย่างบ่อยครั้งตลอดการใช้งาน



## 5.4 แนวทางการพัฒนาต่อ

5.4.1 ใช้การประมวลผลภาพเพื่อหาจุดพิกัดที่วัตถุอยู่แทนการป้อนค่าพิกัด

5.4.2 เพิ่มความละเอียดในการส่งค่าผ่านพอร์ตอนุกรม (Serial port) โดยการส่งค่าทศนิยมด้วย เพื่อส่งการเซอร์โวมอเตอร์เคลื่อนที่ได้แม่นยำขึ้น

5.4.3 ใช้ฮาร์ดแวร์ที่มีความคงทนมีน้ำหนักเบาและมีประสิทธิภาพสูงกว่าเดิม

5.4.4 สามารถนำไปประยุกต์ใช้กับหุ่นยนต์ @HOME ROBOT หรือหุ่นยนต์เพื่อนอัจฉริยะเพื่อใช้ในการแข่งขันได้

5.4.5 สามารถนำหลักการ ไปประยุกต์ใช้ในการพัฒนาระบบควบคุมของแขนกลที่ใช้จริงในโรงงานอุตสาหกรรมต่างๆได้



## เอกสารอ้างอิง

- [1] “Robot arm” เข้าถึงได้จาก <http://akikorhonen.org/projects/robot-hand>
- [2] นาย ภัทรพล ศิวปรียานุพงษ์. (2553). ระบบควบคุมหุ่นยนต์แขนกลอัตโนมัติโดยการประมวลผลภาพ วิทยานิพนธ์ปริญญาบัณฑิต ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ มหาวิทยาลัยนเรศวร
- [3] [www.rsu.ac.th/science/physics/sema/phy\\_121/...13.../lesson\\_10.doc](http://www.rsu.ac.th/science/physics/sema/phy_121/...13.../lesson_10.doc)
- [4] “Roll, Yaw, Pitch” เข้าถึงได้จาก [https://developer.valvesoftware.com/w/index.php?title=File:Roll\\_pitch\\_yaw.gif&limit=50](https://developer.valvesoftware.com/w/index.php?title=File:Roll_pitch_yaw.gif&limit=50)
- [5] “Rotation” เข้าถึงได้จาก <https://alliance.seas.upenn.edu/~medesign/wiki/index.php/Courses/>
- [6] หนังสือ Introduction to Robotics โดย อรรถวิทย์ สุดแสง
- [7] “Degree of Robot arm” เข้าถึงได้จาก <http://controlsenslab.com/introducing-rtx-robotic-tools-for-scilabxcos/?lang=th>  
MEAM520-12A-A03
- [8] “DC Motor” เข้าถึงได้จาก [http://www.stepper-dc-motor.com/china-63zyt01a\\_12v\\_24v\\_36V\\_40v\\_3100\\_3500\\_rpm\\_carbon\\_brushed\\_dc\\_electric\\_motor-531602.html](http://www.stepper-dc-motor.com/china-63zyt01a_12v_24v_36V_40v_3100_3500_rpm_carbon_brushed_dc_electric_motor-531602.html)
- [9] “Servo” เข้าถึงได้จาก [http://www.robotiksystem.com/servo\\_motor\\_types\\_properties.html](http://www.robotiksystem.com/servo_motor_types_properties.html)
- [10] “Arduino ATmega1280” เข้าถึงได้จาก <http://www.thaieasyelec.com/Development-Tools/Arduino/Dev-Board-Unofficial/ATMEGA1280-Base-Board--Compatible-with-Arduino-Project-.html>
- [11] “SR431” เข้าถึงได้จาก <http://www.etteam.com/prod2011/robot/SR431.html>
- [12] “S3003” เข้าถึงได้จาก <http://www.singahobby.com/?q=node/11884>

[13] "Motor worm gear" เข้าถึงได้จาก <http://www.aliexpress.com/wholesale/wholesale-worm-gear-speed-reducers.html>

[14] "Atan2" เข้าถึงได้จาก <http://en.wikipedia.org/wiki/Atan2>



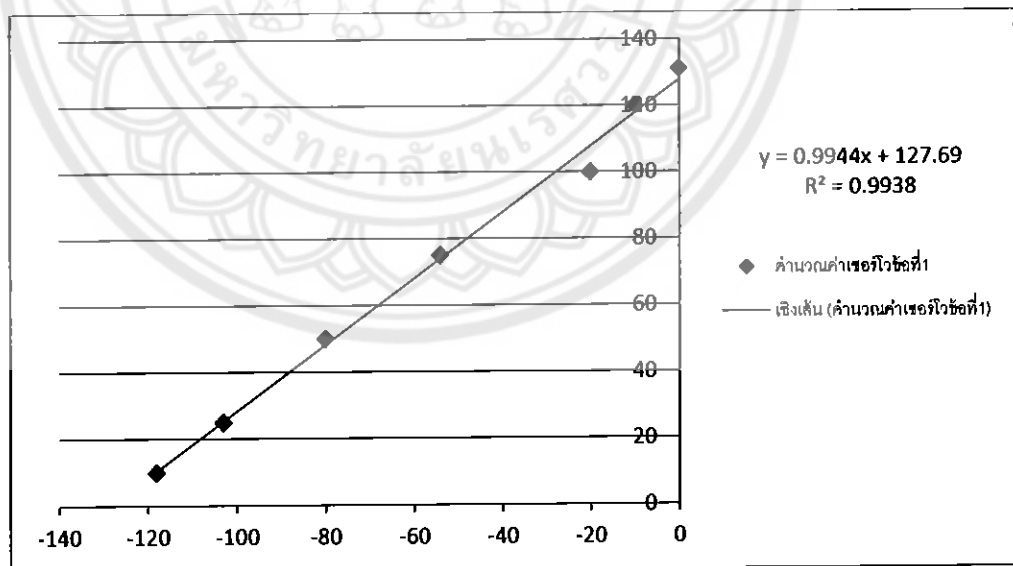
## ภาคผนวก ก

การสร้างสูตรสมการสั่งงานเซอร์โวมอเตอร์ (Servo Motor) และมอเตอร์กระแสตรง (DC Motor)

ข้อที่ 1	
องศาจริง	Value
21	160
0	140
-35	100
-50	80
-83	40
-128	20

ตารางที่ ก-1 ตารางเปรียบเทียบ value กับองศาจริงของเซอร์โวมอเตอร์ข้อที่ 1

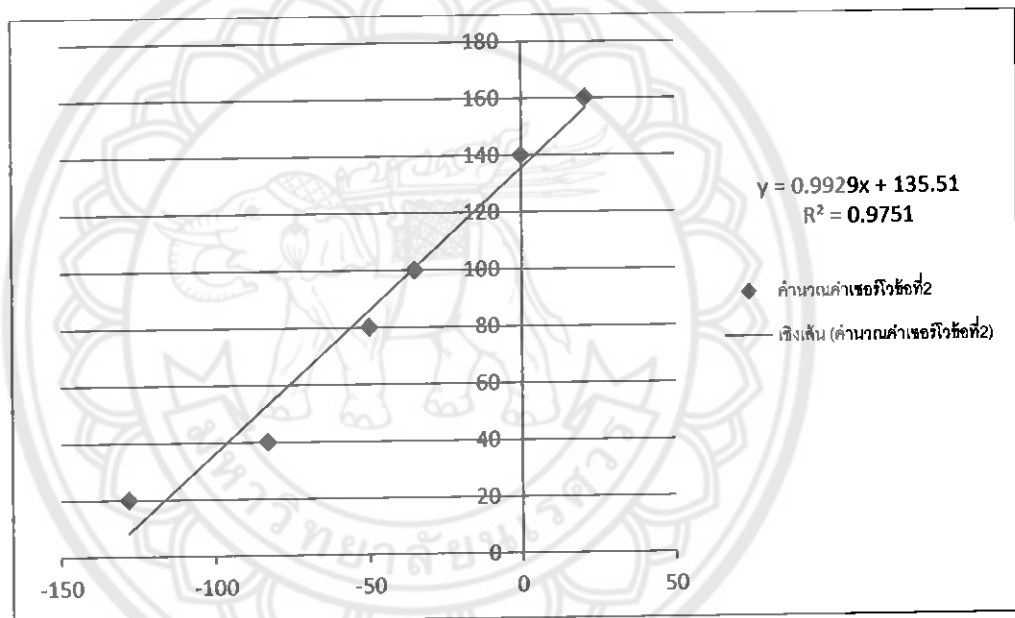
จะเห็นว่าค่า value ที่ส่งไปกับค่าขององศาจริงไม่ตรงกัน จึงต้องแก้ปัญหาโดยการ linear ค่าที่ได้จากกราฟ



รูปที่ ก-1 กราฟแสดงการ linear ค่าจากตารางเปรียบเทียบใน ตารางที่ ก-1

ข้อที่ 2	
องศาจริง	Value
0	131
-10	120
-20	100
-54	75
-80	50
-103	25
-118	10

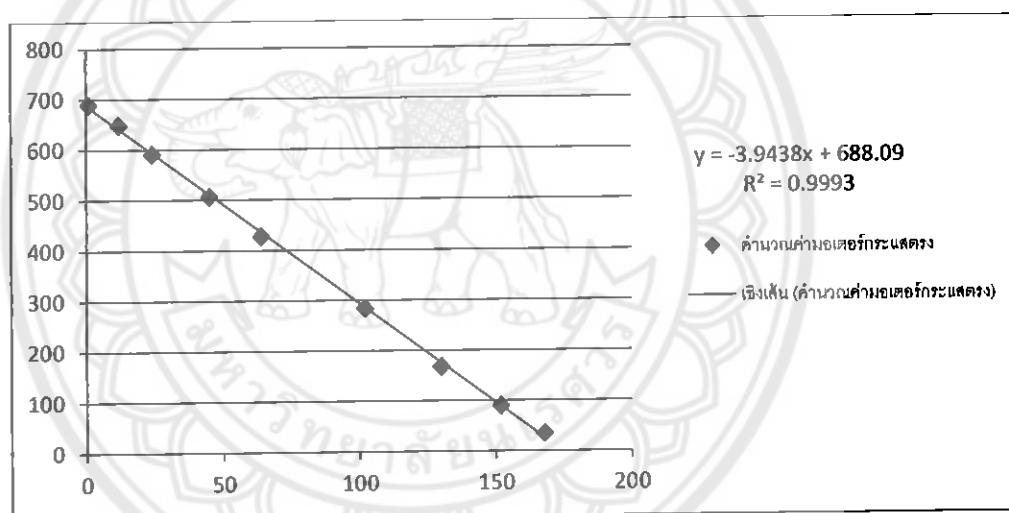
ตารางที่ ก-2 ตารางเปรียบเทียบ value กับองศาจริงของเซอร์ไวโมเตอร์ข้อที่ 2



รูปที่ ก-2 กราฟแสดงการ linear ค่าจากตารางเปรียบเทียบใน ตารางที่ ก-2

DC Motor	
องศาจริง	Value
1	689
12	649
24	591
45	506
64	428
102	284
130	168
152	90
168	35

ตารางที่ ก-3 ตารางเปรียบเทียบค่าแรงดันของกับองศาจริงของมอเตอร์กระแสตรง



รูปที่ ก-3 กราฟแสดงการ linear ค่าจากตารางเปรียบเทียบใน ตารางที่ ก-3

## ภาคผนวก ข

## Algorithm Code ของโปรแกรม

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;

namespace IK
{
    public partial class Form1 : Form
    {
        double zd1 = 0;
        double nzd2 = 0;
        double nzd3 = 0;
        double st1 = 0;
        double st2 = 0;
        double st3 = 0;
        double st4 = 0;
        public Form1()
        {
            InitializeComponent();
        }
        //string x, y;
        private void Set(string strx, string stry, string strz)
        {
            //input target (Cm.)
            double x, y, z;
            x = Convert.ToDouble(strx);
            y = Convert.ToDouble(stry);
            z = Convert.ToDouble(strz);

            //link
            double l1 = 25.0;
            double l2 = 15.0;
            double l3 = 13.0;

            double py = y - 10.5;
            //Solution Inverse Kinematics
            double phi = Math.PI / 3.0;
            double wx = x - (13 * Math.Cos(phi));
            double wy = py - (13 * Math.Sin(phi));
            //Solution for theta2
            double c2 = (Math.Pow(wx, 2.0) + Math.Pow(wy, 2.0) -
            Math.Pow(l1, 2.0) - Math.Pow(l2, 2.0)) / (2.0 * l1 * l2);
            double s2 = Math.Sqrt(1 - Math.Pow(c2, 2));
            double z2 = Math.Atan2(s2, c2);
        }
    }
}

```

```

//Solution for theta1
double s1 = (((11 + 12 * Math.Cos(z2)) * Wy) - (12 * s2 * Wx))
/ (Math.Pow(Wx, 2.0) + Math.Pow(Wy, 2.0));
double c1 = (((11 + 12 * Math.Cos(z2)) * Wx) - (12 * s2 * Wy))
/ (Math.Pow(Wx, 2.0) + Math.Pow(Wy, 2.0));
double z1 = Math.Atan2(s1, c1);
//Solution for theta3
double z3 = phi - z1 - z2;

//Solution for radian to degree
double r2d = 180.0 / Math.PI;
zd1 = (z1 * r2d);
double zd2 = (z2 * r2d);
double zd3 = (z3 * r2d);

//Convert theta
double fre = Math.Atan(py / x);
double free = fre * (r2d);
double nzd1 = (zd1 * (-1)) + (free * 2);
nzd2 = zd2 * (-1);
nzd3 = zd3 * (-1);

//Output theta to textbox
seta1.Text = nzd1.ToString();
seta2.Text = nzd2.ToString();
seta3.Text = nzd3.ToString();

//insteadaw link
double x1 = 11 * (Math.Cos(nzd1));
double y1 = 11 * (Math.Sin(nzd1));

st1 = (0.9929*(nzd3) + 135.51);
st2 = (0.9944 * (nzd2) + 127.69);
st3 = nzd1;
st4 = z;

//Value Servo to textbox
servo1.Text = st1.ToString();
servo2.Text = st2.ToString();
servo3.Text = st3.ToString();
}

//Solution Send to Arduino
private void button3_Click(object sender, EventArgs e)
{
    int izd1 = Convert.ToInt32(st1);
    int izd2 = Convert.ToInt32(st2);
    int izd3 = Convert.ToInt32(st3);
    int izdz = Convert.ToInt32(st4);

    byte[] o = { 0x01, (byte)izd1, (byte)izd2, (byte)izd3,
    (byte)izdz};
    if (!serialPort1.IsOpen) serialPort1.Open();
    serialPort1.Write(o, 0, 5);
}

//Show Output
private void button1_Click(object sender, EventArgs e)
{
    Set(xbox.Text.ToString(),
    ybox.Text.ToString(),zbox.Text.ToString());
}

```



```
//Reset
private void button2_Click(object sender, EventArgs e)
{
    xbox.Text = "";
    ybox.Text = "";
    zbox.Text = "";
}

//Open port
private void Form1_Load(object sender, EventArgs e)
{
    serialPort1.Open();
}

//Close port
private void Form1_FormClosed(object sender, FormClosedEventArgs e)
{
    serialPort1.Close();
}
}
```

