



รายงานการวิจัย พัฒนานวัตกรรมและสิ่งประดิษฐ์

การสร้างเครื่องฝึกบินจำลองต้นแบบ  
(PROTOTYPING OF FLIGHTSIMULATOR MODULE)

ได้รับทุนอุดหนุนการวิจัย พัฒนานวัตกรรมและสิ่งประดิษฐ์จาก  
กองทุนนวัตกรรมและสิ่งประดิษฐ์ สมเด็จพระเทพรัตนราชสุดาฯ สยามบรมราชกุมารี

ผลงานวิจัยเป็นความรับผิดชอบของหัวหน้าโครงการวิจัยแต่เพียงผู้เดียว



รายงานการวิจัย พัฒนานวัตกรรมและสิ่งประดิษฐ์

การสร้างเครื่องฝึกบินจำลองต้นแบบ  
(PROTOTYPING OF FLIGHTSIMULATOR MODULE)

หัวหน้าโครงการ

รองศาสตราจารย์ เรืออากาศ ดร.กนต์ธร ชำนิประศาสน์  
สาขาวิชาวิศวกรรมเครื่องกล สำนักวิชาวิศวกรรมศาสตร์  
มหาวิทยาลัยเทคโนโลยีสุรนารี

ได้รับทุนอุดหนุนการวิจัย พัฒนานวัตกรรมและสิ่งประดิษฐ์จาก  
กองทุนนวัตกรรมและสิ่งประดิษฐ์ สมเด็จพระเทพรัตนราชสุดาฯ สยามบรมราชกุมารี  
ผลงานวิจัยเป็นความรับผิดชอบของหัวหน้าโครงการวิจัยแต่เพียงผู้เดียว

กรกฎาคม 2559

## บทคัดย่อ

งานวิจัยนี้ เป็นการสร้างเครื่องต้นแบบเครื่องฝึกจำลองการบิน ที่จะใช้ในการเรียนการสอนของนักศึกษาสาขาวิชาวิศวกรรมอากาศยาน มหาวิทยาลัยเทคโนโลยีสุรนารี การสร้างเครื่องฝึกบินจำลองนี้ จะทำการสร้างห้องนักบิน (Cockpit) การจำลองภาพการบินจะใช้โปรแกรมสำเร็จรูปในการสร้างภาพ โดยจะทำการเชื่อมต่ออุปกรณ์ควบคุมอากาศยานเข้ากับคอมพิวเตอร์ เพื่อให้การตอบสนองของภาพที่ปรากฏเหมือนกับการบินจริง อีกทั้งเครื่องมือวัดประกอบการบินต่าง ๆ จะเชื่อมต่อเข้ากับระบบควบคุม เพื่อให้การตอบสนองของเครื่องมือวัดมีความสมจริง ในด้านการเคลื่อนที่เครื่องต้นแบบนี้ จะสร้างให้ห้องนักบินสามารถเคลื่อนที่ได้ใน 3 ทิศทาง คือ การควง (Roll) การเงย (Pitch) และการเคลื่อนที่ตอบสนองกับความเร่งหรือแรงกระแทก (Shock) โดยการทำงานจะใช้เครื่องกระตุ้นด้วยมอเตอร์ในการสร้างเคลื่อนที่

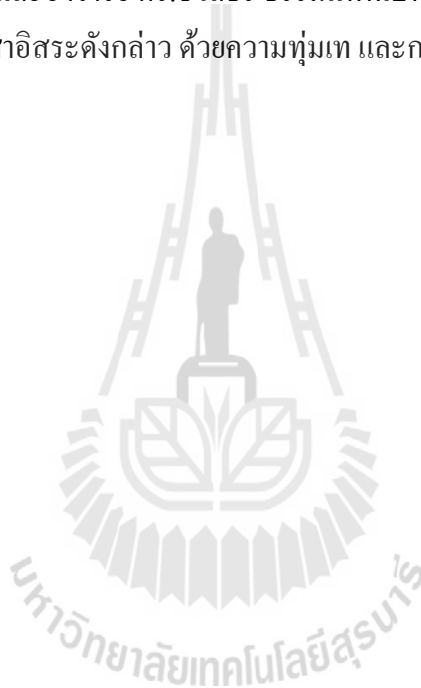
งานวิจัยนี้จะเป็นการออกแบบระบบเชื่อมโยงเพื่อควบคุมการทำงานของส่วนหลัก ๆ คือ ส่วนการเคลื่อนที่ของห้องนักบิน ส่วนแสดงผลภาพที่มองจากห้องนักบิน ส่วนแสดงผลเครื่องมือวัดประกอบการบิน และส่วนควบคุมการเคลื่อนที่ของอากาศยานให้ทำงานร่วมกันอย่างมีประสิทธิภาพ เพื่อให้เครื่องฝึกบินจำลองนี้ให้ความรู้สึกล้ำคล้ายกับการขึ้นบินบนอากาศยาน ผลที่ได้จากงานวิจัยนี้จะทำให้วิศวกรด้านการบินและอากาศยานสามารถมีเครื่องจำลองการบินที่มีราคาถูกกว่าสร้างจากต่างประเทศ และสามารถผลิตขึ้นเองในมหาวิทยาลัย ซึ่งถือว่าเป็นการสร้างนวัตกรรมใหม่อย่างหนึ่ง

## กิตติกรรมประกาศ

โครงการวิจัย พัฒนานวัตกรรมและสิ่งประดิษฐ์ เรื่องการพัฒนาเครื่องฟีกบินจำลองชนิดสามองศาอิสระ สามารถสำเร็จลุล่วงไปด้วยดี ทั้งนี้ต้องขอขอบคุณ กองทุนนวัตกรรมและสิ่งประดิษฐ์ สมเด็จพระเทพรัตนราชสุดาฯ สยามบรมราชกุมารี และมหาวิทยาลัยเทคโนโลยีสุรนารี ที่ให้ทุนสนับสนุนการทำสิ่งประดิษฐ์นี้ นอกจากนี้ผู้วิจัยต้องขอขอบคุณนายอภิรักษ์ ห่อนกลาง นางสาววัลรัตน์ ทองปิ่นและอาจารย์ ดร.ชโลธร ธรรมแท้ที่ เป็นผู้ช่วยในการดำเนินการสร้างเครื่องฟีกบินจำลองชนิดสามองศาอิสระดังกล่าว ด้วยความทุ่มเท และการเอาใจใส่อย่างยิ่ง

กนต์ธร ชำนิประศาสน์

กรกฎาคม 2559





# สารบัญ

หน้า

บทคัดย่อ (ภาษาไทย).....	ก
กิตติกรรมประกาศ.....	ข
สารบัญ.....	ค
สารบัญรูป.....	ฉ
สารบัญตาราง.....	ญ
<b>บทที่</b>	
<b>1 บทนำ.....</b>	<b>1</b>
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์การวิจัย.....	3
1.3 ขอบเขตของการวิจัย.....	4
1.4 สถานที่ทำงานวิจัย.....	4
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	4
<b>2 ปรัชญาวรรณกรรมและทฤษฎีที่เกี่ยวข้อง.....</b>	<b>5</b>
2.1 กล่าวนำ.....	5
2.2 UDP Protocol.....	9
2.2.1 UDP Message.....	9
2.2.2 UDP Header.....	10
2.2.3 UDP Port.....	11
2.3 ไมโครคอนโทรลเลอร์ (Microcontroller).....	12
2.3.1 ส่วนประกอบทั่วไปของไมโครคอนโทรลเลอร์.....	13
2.3.2 ประเภทของไมโครคอนโทรลเลอร์.....	14
2.3.3 ภาษาที่ใช้เขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์.....	14

## สารบัญ (ต่อ)

หน้า

2.4 เซอร์โวมอเตอร์ (Servo Motor) .....	15
2.4.1 Servo Motor.....	15
2.4.2 โครงสร้างภายในของ Servo Motor.....	15
2.4.3 Amplifier หรือ Driver .....	16
2.5 ทฤษฎีระบบควบคุม (Control Theory) .....	16
2.5.1 ทฤษฎีควบคุมแบบสัดส่วน-ปริพันธ์-อนุพันธ์ (PID Controller Theory) ....	17
3 วิธีการดำเนินงานวิจัย.....	22
3.1 กล่าวนำ.....	22
3.2 โปรแกรมจำลองการบิน.....	22
3.2.1 Microsoft Flight Simulator X (FSX).....	23
3.2.2 X-Plane .....	24
3.2.3 การศึกษาข้อดีและข้อเสียของ Microsoft Flight Simulator X (FSX) และ X-Plane .....	25
3.3 การออกแบบแผ่นเคลื่อนไหว (Motion Platform).....	25
3.3.1 ระบบแกนอ้างอิง (Reference system).....	27
3.3.2 Matrix Transformation.....	28
3.3.3 การเติมข้อมูลเพื่อส่งสัญญาณไปยังอุปกรณ์ต้นกำลัง (Actuator signals).....	32
3.3.4 อุปกรณ์ต้นกำลัง (Actuator).....	33
3.4 อัลกอริทึมสำหรับประมวลผลเพื่อเชื่อมต่อข้อมูลระหว่างโปรแกรมกับแผ่น เคลื่อนไหว.....	34
3.4.1 การส่งถ่ายข้อมูลจากโปรแกรมจำลองการบิน .....	34
3.5 การตรวจสอบความถูกต้อง.....	36
3.5.1 ความถูกต้องของระบบ .....	36
3.5.2 การปรับค่าความเหมาะสมของ PID Controller.....	36
3.5.3 การตรวจสอบผลการตอบสนองต่อสัญญาณป้อนเข้า .....	37

## สารบัญ (ต่อ)

หน้า

<b>4 ผลการทดลอง.....</b>	<b>38</b>
4.1 การศึกษาความสามารถของโปรแกรมตามหัวข้อที่ต้องการ.....	38
4.2 การออกแบบและขนาดของแผ่นเคลื่อนไหว.....	39
4.2.1 การออกแบบห้องนักบิน.....	39
4.2.2 ขั้นตอนการสร้างชิ้นงาน.....	41
4.3 การควบคุมเซอร์โวมอเตอร์.....	43
4.3.1 หลักการควบคุมเซอร์โวมอเตอร์ไฟฟ้ากระแสสลับพร้อมชุดควบคุม.....	43
4.3.2 การเพิ่มความสามารถในการรับภาระกรรมเซอร์โวมอเตอร์.....	44
4.4 การปรับจูนค่าระบบควบคุม PID.....	45
4.5 การทดลองความถูกต้องของระบบ.....	47
4.5.1 เครื่องฝึกบินจำลอง.....	47
4.5.2 ข้อจำกัดด้านการเคลื่อนที่ของแผ่นเคลื่อนไหว.....	48
4.5.3 การตอบสนองของระบบต่อฟังก์ชัน ขั้นบันไดหนึ่งหน่วย (Unit Step Input).....	48
4.5.4 การตอบสนองของระบบต่อฟังก์ชัน ลาดเอียงหนึ่งหน่วย (Unit Ramp Input).....	57
<b>5 สรุปผลและข้อเสนอแนะ.....</b>	<b>65</b>
5.1 สรุปผลงานวิจัย.....	65
5.2 ข้อเสนอแนะ .....	65
รายการอ้างอิง .....	66
ภาคผนวก	
ภาคผนวก ก. รายละเอียดโปรแกรมจำลองการบิน.....	68
ภาคผนวก ข. บทความทางวิชาการที่ได้รับการตีพิมพ์เผยแพร่ในระหว่างการศึกษา.....	78

## สารบัญรูป

รูปที่		หน้า
2.1	ลักษณะของ Stewart Platform.....	6
2.2	แบบจำลองแผ่นเคลื่อนไหวนิคสามองศาอิสระ.....	7
2.3	แผ่นเคลื่อนไหวนิคสามองศาอิสระขนาดจริง.....	8
2.4	การจัดเก็บข้อมูลใน UDP.....	10
2.5	โครงสร้างของ UDP Header.....	10
2.6	หมายเลขพอร์ต UDP ที่เป็นที่รู้จัก.....	12
2.7	แผนภาพบล็อกของการควบคุมแบบพีไอดี.....	17
2.8	การตอบสนองต่อฟังก์ชันอินพุตพื้นฐาน.....	20
3.1	โปรแกรมจำลองการบิน (a.) Microsoft Flight Simulator X (FSX) (b.) X-Plane.....	23
3.2	เครื่องบินฝึกจำลอง.....	26
3.3	เครื่องเล่นในสวนสนุก.....	26
3.4	ระบบแกนอ้างอิง (L = ระยะห่างของ Servo).....	27
3.5	การหมุนรอบแกน X.....	28
3.6	ความสัมพันธ์ระหว่างมุมของเพลลา $v_d$ และมุมของแท่นเคลื่อนไหวน $v$ .....	32
3.7	Servo Motor Mitsubishi รุ่น MR-J2.....	33
3.8	การตั้งค่าการส่งข้อมูลจากโปรแกรมจำลองการบิน.....	35
3.9	ค่าการปรับจูนสัมประสิทธิ์.....	37
4.1	แผ่นเคลื่อนไหวน.....	39
4.2	ส่วนโครงสร้างห้องนักบิน.....	40
4.3	ขนาดส่วนแผงหน้าชุดควบคุม.....	40
4.4	โครงสร้างแบบสมบูรณ์จากการออกแบบในโปรแกรม.....	41
4.5	โครงสร้างประกอบอลูมิเนียมโปรไฟล์.....	41

## สารบัญรูป (ต่อ)

รูปที่	หน้า	
4.6	แผงหน้าชุดควบคุม.....	42
4.7	ประกอบชุดควบคุมและเครื่องวัด.....	42
4.8	เซอร์โวมอเตอร์ยี่ห้อ Mitsubishi รุ่น MR-J2 400 W.....	43
4.9	เซอร์โวมอเตอร์ยี่ห้อ Panasonic รุ่น MINAS E Series 400 W.....	43
4.10	ลักษณะสัญญาณการควบคุมเซอร์โวมอเตอร์ทิศทางละ 4 Step.....	44
4.11	ชุดเฟืองทดอัตราทด 70 ต่อ 1.....	45
4.12	หน้าต่างโปรแกรมการปรับจูนค่า PID ในโปรแกรม Arduino.....	46
4.13	กราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบขั้นบันได 2 องศา ของการเคลื่อนไหวแบบ Pitch Up เมื่อเปลี่ยนชนิดของการควบคุม P, PI และ PID.....	46
4.14	เครื่องฟีกบินจำลองที่เสร็จสมบูรณ์.....	47
4.15	กราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบขั้นบันได 1 องศา ของการเคลื่อนไหวแบบ Pitch Up.....	49
4.16	กราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบขั้นบันได 1 องศา ของการเคลื่อนไหวแบบ Pitch Down.....	50
4.17	กราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบขั้นบันได 1 องศา ของการเคลื่อนไหวแบบ Roll Right.....	51
4.18	กราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบขั้นบันได 1 องศา ของการเคลื่อนไหวแบบ Roll Left.....	52
4.19	กราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบขั้นบันได 2 องศา ของการเคลื่อนไหวแบบ Pitch Up.....	53
4.20	กราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบขั้นบันได 2 องศา ของการเคลื่อนไหวแบบ Pitch Down.....	54
4.21	กราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบขั้นบันได 2 องศา ของการเคลื่อนไหวแบบ Roll Right.....	55
4.22	กราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบขั้นบันได 2 องศา ของการเคลื่อนไหวแบบ Roll Left.....	56

## สารบัญรูป (ต่อ)

รูปที่	หน้า
4.23 กราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบลาดเอียงค่าสุดท้ายที่ 1.5 องศา ของการเคลื่อนไหวแบบ Pitch Up.....	57
4.24 กราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบลาดเอียงค่าสุดท้ายที่ 2 องศา ของการเคลื่อนไหวแบบ Pitch Up.....	58
4.25 กราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบลาดเอียงค่าสุดท้ายที่ 1.5 องศา ของการเคลื่อนไหวแบบ Pitch Down.....	59
4.26 กราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบลาดเอียงค่าสุดท้ายที่ 2 องศา ของการเคลื่อนไหวแบบ Pitch Down.....	60
4.27 กราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบลาดเอียงค่าสุดท้ายที่ 1.5 องศา ของการเคลื่อนไหวแบบ Roll Right.....	61
4.28 กราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบลาดเอียงค่าสุดท้ายที่ 2 องศา ของการเคลื่อนไหวแบบ Roll Right.....	62
4.29 กราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบลาดเอียงค่าสุดท้ายที่ 1.5 องศา ของการเคลื่อนไหวแบบ Roll Left.....	63
4.30 กราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบลาดเอียงค่าสุดท้ายที่ 2 องศา ของการเคลื่อนไหวแบบ Roll Left.....	64
ก.1 โหมดการบินเดี่ยวสำหรับผู้ที่ต้องการชมทิวทัศน์ สำหรับโปรแกรม Flight Simulator X .....	69
ก.2 หน้าต่างการเลือกอากาศยานที่ต้องการสำหรับโปรแกรม Flight Simulator X.....	69
ก.3 หน้าต่างกำหนดแผนการบินที่เราต้องการสำหรับโปรแกรม Flight Simulator X.....	70
ก.4 หน้าต่างสภาพอากาศจริงของพื้นที่ที่ทำการบินสำหรับโปรแกรม Flight Simulator X.....	70
ก.5 หน้าต่างกำหนดเชื้อเพลิงในการทำงานและน้ำหนักการบรรทุก ซึ่งมีผลต่อระยะทางและการทรงตัวสำหรับโปรแกรม Flight Simulator X.....	71
ก.6 มุมมองห้องนักบิน ผู้ใช้สามารถควบคุมแผงควบคุมได้ทุกอย่าง สำหรับโปรแกรม Flight Simulator X.....	71
ก.7 มุมมองภายนอกอากาศยานสำหรับโปรแกรม Flight Simulator X.....	72

## สารบัญรูป (ต่อ)

รูปที่		หน้า
ก.8	ระบบหลายหน้าต่างทำให้เราสามารถเห็นสิ่งต่าง ๆ ในเวลาเดียวกัน สำหรับ โปรแกรม Flight Simulator X.....	72
ก.9	ระบบนำร่องอากาศยาน (GPS) สำหรับ โปรแกรม Flight Simulator X.....	73
ก.10	หมวดการเรียนรู้เกี่ยวกับเครื่องบินสำหรับ โปรแกรม Flight Simulator X.....	73
ก.11	โหมดการบินเดี่ยวสำหรับผู้ที่ต้องการชมทิวทัศน์สำหรับ โปรแกรม X-Plane.....	74
ก. 12	หน้าต่างการเลือกอากาศยานที่ต้องการสำหรับ โปรแกรม X-Plane.....	74
ก.13	หน้าต่างกำหนดแผนการบินที่เราต้องการสำหรับ โปรแกรม X-Plane.....	75
ก.14	หน้าต่างสภาพอากาศ ณ เวลาจริงของพื้นที่ที่ทำการบินสำหรับ โปรแกรม X-Plane.....	75
ก.15	มุมมองห้องนักบิน ผู้ใช้สามารถควบคุมแผงควบคุมได้ทุกอย่าง สำหรับ โปรแกรม X-Plane.....	76
ก. 16	มุมมองภายนอกอากาศยานสำหรับ โปรแกรม X-Plane.....	76
ก.17	ระบบนำร่องอากาศยาน (GPS) สำหรับ โปรแกรม X-Plane.....	77



## สารบัญตาราง

ตารางที่		หน้า
3.1	ความต้องการขั้นต่ำของโปรแกรม Microsoft Flight Simulator X.....	24
3.2	ความต้องการขั้นต่ำของโปรแกรม X-Plane.....	25
4.1	ความสามารถของโปรแกรมจำลองการบินที่ทำการศึกษา.....	38
4.2	ค่าพารามิเตอร์ที่ใช้ในการออกแบบ.....	39
4.3	ค่าพารามิเตอร์ของเซอร์โวมอเตอร์.....	44
4.4	ข้อจำกัดของแผ่นเคลื่อนไหวก.....	48



# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

ผู้วิจัยได้เล็งเห็นถึงความสำคัญของอุตสาหกรรมการบินในประเทศไทย มาเป็นระยะเวลาหนึ่งแล้ว นับตั้งแต่การเปิดท่าอากาศยานสุวรรณภูมิขึ้นในปี พ.ศ. 2549 ทั้งภาครัฐและหน่วยงานเอกชน รัฐวิสาหกิจที่เกี่ยวข้องได้มีความพยายามที่จะให้ประเทศไทยเป็นศูนย์กลางด้านอุตสาหกรรมการบินและอากาศยานในภูมิภาคเอเชียตะวันออกเฉียงใต้ มีขีดความสามารถในการบำรุงรักษา และสามารถซ่อมอากาศยานเชิงพาณิชย์ ได้ในทุกขนาด อย่างไรก็ตามการที่ประเทศไทยจะเป็นศูนย์กลางทางการบินได้ นอกเหนือจากการที่มีเครื่องมือซ่อมบำรุงที่ครบครัน มีอุตสาหกรรมที่รองรับการบำรุงรักษาอากาศยานสิ่งที่จำเป็นที่สุดคือจะต้องมีบุคลากรที่มีความรู้ความสามารถมีความเข้าใจในการทำงานของอากาศยาน และมีความรู้ความสามารถที่จะสามารถซ่อมบำรุงอากาศยานได้อย่างมีประสิทธิภาพ

โอกาสของประเทศในด้านอุตสาหกรรมการบิน ได้ปรับเปลี่ยนไปในทิศทางที่น่ายินดีอีกครั้งหนึ่ง เมื่อการตกลงของประชาคมเศรษฐกิจอาเซียน (AEC) ได้ตกลงกันว่า เมื่อมีการรวมเศรษฐกิจของเหล่าสมาชิกแล้ว ประเทศไทยจะเป็นผู้รับผิดชอบในการเป็นศูนย์กลางด้านอากาศยาน และการบินของภูมิภาค ในปี พ.ศ. 2558 ทำให้ในระยะเวลาไม่ถึง 3 ปีนับจากวันนี้ ประเทศไทย ต้องรีบเตรียมการด้านบุคลากรในด้านการอากาศยานและการบิน เพื่อเตรียมรับอุตสาหกรรมการบินของทุกประเทศในภูมิภาค รวมถึงการขยายตัวเพื่อรองรับเศรษฐกิจของโลกที่กำลังเจริญเติบโต อย่างรวดเร็ว ในย่านภูมิภาคเอเชีย ไม่ว่าจะเป็น ประเทศจีน อินเดีย และเกาหลีใต้ เป็นต้น เพื่อรองรับการเจริญเติบโตของอุตสาหกรรมการบินดังกล่าว รัฐบาลจึงมีแนวทางที่จะพัฒนานิคมอุตสาหกรรมด้านอากาศยาน และการบินขึ้นในประเทศและมีแนวโน้มความเป็นไปได้สูงมากที่จะสร้างนิคมอุตสาหกรรมขึ้นในจังหวัดนครราชสีมา ซึ่งเป็นจังหวัดที่ตั้งของมหาวิทยาลัยเทคโนโลยีสุรนารี เนื่องจากปัจจุบันมีสนามบินพลเรือนที่ยังใช้งานไม่เต็มประสิทธิภาพ มีที่ตั้งอยู่ห่างจากชุมชนในระยะที่เหมาะสม ที่ดินยังมีราคาไม่สูงนัก และมีระบบคมนาคมทางบกที่สะดวก อยู่ใกล้ทั้งถนนสายหลักและทางรถไฟ ดังนั้นการสร้างนิคมอุตสาหกรรมจะมีผลกระทบน้อยทั้งในแง่การลงทุนและสิ่งแวดล้อม เปรียบเทียบกับการที่จะต้องสร้างนิคมประเภทนี้ที่สถานอื่น

จึงถือเป็น โอกาสดีที่มหาวิทยาลัยเทคโนโลยีสุรนารี ซึ่งมีวัตถุประสงค์หลักในการผลิตบัณฑิตด้านวิทยาศาสตร์และเทคโนโลยี จะได้รับผลจากการสร้างนิคมอุตสาหกรรมด้านอากาศยาน และการบินขึ้นในจังหวัดนครราชสีมา ไม่เพียงแต่วิศวกรอากาศยานเท่านั้น แต่วิศวกรในสาขาวิชาอื่น เช่น วัสดุศาสตร์ ไฟฟ้าอิเล็กทรอนิกส์ ไฟฟ้ากำลัง ไฟฟ้าสื่อสาร วิศวกรรมเครื่องกล วิศวกรรมการผลิต วิศวกรรมอุตสาหกรรม วิศวกรรมคอมพิวเตอร์ และอื่น ๆ ล้วนแต่จะได้ประโยชน์จากการขยายตัวนี้ทั้งสิ้น เพราะในอากาศยานแต่ละลำ ต้องการใช้วิศวกรหลากหลายประเภทในการซ่อมบำรุง สร้างชิ้นส่วนของอากาศยานและระบบการบินของอากาศยานนั้น ดังนั้นในฐานะที่เราเป็นมหาวิทยาลัยในจังหวัดนี้ จึงต้องมองหาแนวทางที่จะสร้างองค์ความรู้ให้กับคณาจารย์ บุคลากร นักวิจัย เพื่อที่จะสามารถถ่ายทอดความรู้นี้ให้กับนักศึกษาวิศวกรรมศาสตร์ ทุกสาขาวิชา

การที่จะทำให้นักศึกษามีความเข้าใจในการซ่อมบำรุงระบบต่าง ๆ ของอากาศยาน ขึ้นแรกจะต้องมีการสอนให้ความรู้เกี่ยวกับการทำงานของอากาศยาน และส่วนประกอบของอากาศยาน ก่อนที่จะลงรายละเอียดของระบบต่าง ๆ ดังนั้นการทำงานของอากาศยาน และการบินของอากาศยาน เป็นพื้นฐานของความรู้ในด้านนี้ที่นักศึกษาจะต้องมีความเข้าใจอย่างละเอียด ก่อนที่จะศึกษาในรายละเอียดด้านต่าง ๆ ต่อไป แต่การที่จะให้นักเรียนได้ทำการทดสอบบังคับอากาศยานจริงนั้น เป็นเรื่องที่เสี่ยงต่ออุบัติเหตุและสิ้นเปลืองงบประมาณเป็นจำนวนมาก ดังนั้นสถานศึกษาที่ทำการสอนด้านการบิน และอากาศยานต่าง ๆ จึงหลีกเลี่ยงการบินจริงแต่เปลี่ยนมาใช้เครื่องจำลองการบินในการฝึกศึกษาในส่วนของพื้นฐานความรู้ให้กับนักศึกษา ทำให้การที่มหาวิทยาลัยเทคโนโลยีสุรนารีจะต้องมีเครื่องฝึกบินจำลองมีความจำเป็นอย่างมากในการสร้างบุคลากรที่มีความรู้ความสามารถในด้านนี้

อย่างไรก็ตามเครื่องฝึกบินจำลองที่มีอยู่ในประเทศไทยนั้น ไม่ได้ผลิตโดยคนไทย เป็นการนำเข้าและใช้เทคโนโลยีจากต่างประเทศทั้งสิ้น แม้ว่าในปัจจุบันนี้เทคโนโลยีที่ใช้กับเครื่องฝึกบินจำลองไม่ใช่เทคโนโลยีที่ใหม่เมื่อเทียบกับการที่เราเริ่มมีเครื่องฝึกบินจำลองครั้งแรกในประเทศไทย แม้กระนั้นเครื่องฝึกบินจำลองที่ต้องสั่งซื้อจากต่างประเทศ กลับมีราคาสูงแม้ว่าจะ เป็นเครื่องฝึกบินแบบเรียบง่ายและเป็นขั้นพื้นฐานก็ตาม ทั้งนี้อาจเป็นเพราะการที่มีระบบการฝึกบินจำลอง (Flight Simulator) อยู่หลายระบบ แต่การที่จะต้องเชื่อมต่อบริษัทการทำงานของโปรแกรม เข้ากับอากาศยานจำลอง (Aircraft Simulator) อาจต้องการเทคโนโลยีที่เฉพาะและเป็นความลับทางการค้าของแต่ละบริษัท อีกทั้งการที่จะได้ระบบห้องฝึกบินเสมือนจริง (Virtual Cockpit) มีความต้องการผู้พัฒนาระบบคอมพิวเตอร์กราฟฟิกส์ที่ทันสมัยก็เป็นได้ ผู้วิจัยจึงมีแนวความคิดที่จะสร้างเครื่องฝึกบินจำลองขึ้น เพื่อสร้างองค์ความรู้ให้กับคณาจารย์ นักวิจัยและบุคลากรของสาขาวิชา สร้างทักษะในด้านการสร้างเครื่องจำลองการบินสร้างองค์ความรู้ที่จะนำเทคโนโลยีต่าง ๆ มาประกอบใช้ร่วมกัน พร้อมทั้งนำเทคโนโลยีใหม่ ๆ ที่ใช้กับอากาศยานในปัจจุบัน มาบรรจุเข้าใน

เครื่องฝึกบินจำลองนี้ เช่น แผงหน้าปัดดิจิทัล ระบบควบคุมการบินด้วยไฟฟ้า (Fly by Wire) ระบบสั่งผ่านหน้าจอแบบสัมผัส (Touch Screen) เป็นต้น ซึ่งแม้ว่าเครื่องบินรุ่นใหม่ ๆ จะใช้ระบบเหล่านี้อย่างกว้างขวาง แต่ในเครื่องฝึกบินจำลองยังไม่ได้มีการนำมาใช้งานอย่างแพร่หลาย ซึ่งคณะผู้วิจัยได้เดินทางเข้าศึกษาฐานเครื่องจำลองการบินที่มีอยู่ในประเทศไทย หลายสถานที่ด้วยกันทั้งของ บริษัทการบินไทยจำกัด (มหาชน) สถาบันการบินพลเรือน โรงเรียนนายเรืออากาศ ซึ่งแต่ละหน่วยงานดังกล่าวมีภารกิจในการผลิตบุคลากรที่แตกต่างกัน จึงใช้เครื่องฝึกบินจำลองที่แตกต่างกัน ซึ่งคณะผู้วิจัยได้ศึกษาความเหมาะสมเบื้องต้นว่ามหาวิทยาลัยเทคโนโลยีสุรนารี ควรใช้เครื่องฝึกบินจำลองแบบใดเพื่อจะได้เหมาะสมกับการพัฒนาการเรียนการสอน และงานวิจัยของมหาวิทยาลัยเทคโนโลยีสุรนารี และเหมาะสมกับการรองรับการขยายตัวของอุตสาหกรรมการบินเข้ามาในเขตจังหวัดนครราชสีมาต่อไป

การสร้างเครื่องฝึกบินจำลองนี้จึงนับเป็นนวัตกรรมใหม่ของการเรียนการสอนด้านอากาศยาน เพราะจะเป็นการใช้เครื่องมือที่ผลิตขึ้นใช้งานเองในประเทศ โดยคณาจารย์และนักวิจัยของมหาวิทยาลัยเทคโนโลยีสุรนารีเอง ดังนั้น เมื่อได้เครื่องต้นแบบแล้ว ประโยชน์ที่ได้ ไม่เพียงแต่การที่มหาวิทยาลัยสามารถสร้างเครื่องฝึกบินจำลองต้นแบบได้เอง ซึ่งเป็นการประหยัดงบประมาณของชาติเท่านั้นสิ่งที่สำคัญกว่านั้น คือการที่คณาจารย์และบุคลากรได้ทักษะและองค์ความรู้ที่จะสามารถถ่ายทอดให้กับนักศึกษาของมหาวิทยาลัย และบุคคลที่มีความต้องการที่จะเรียนรู้เพื่อเข้าสู่ภาคอุตสาหกรรมการบินและอากาศยาน เพื่อร่วมกันสร้างความเข้มแข็งในด้านนี้ให้กับประเทศได้ต่อไป นอกเหนือจากนั้นการที่สามารถสร้างเครื่องต้นแบบได้เอง จะสามารถทำให้มหาวิทยาลัยสามารถที่จะพัฒนาต่อยอดความรู้ด้านนี้ ให้มีความสมบูรณ์ เพิ่มความทันสมัยและสามารถผลิตเพื่อการค้าทั้งนั้นเพื่อรองรับการขยายตัวของความต้องการบุคลากรด้านการบินและอากาศยานที่จะเกิดขึ้นในอนาคตอันใกล้ต่อไปด้วย

## 1.2 วัตถุประสงค์การวิจัย

สำหรับงานวิจัยนี้มีวัตถุประสงค์ของงานวิจัยในด้านการออกแบบและพัฒนาอัลกอริทึมในการเชื่อมความสัมพันธ์ระหว่างซอฟต์แวร์และฮาร์ดแวร์ ซึ่งวัตถุประสงค์หลักของงานวิจัยมีดังต่อไปนี้

1. พัฒนาเครื่องฝึกบินจำลอง เพื่อใช้ในการเรียนการสอนให้กับนักศึกษามหาวิทยาลัยเทคโนโลยีสุรนารี ในสาขาวิชาวิศวกรรมอากาศยาน และสาขาวิชาอื่นที่เกี่ยวข้อง
2. สร้างนวัตกรรมในการพัฒนาเครื่องฝึกบินจำลองให้มีความเหมาะสมกับการเรียนการสอนของนักศึกษาวิศกรรมอากาศยาน ที่มีราคาต่ำกว่าการสั่งซื้อจากต่างประเทศ

3. เชื่อมโยงเทคโนโลยีด้านต่าง ๆ ที่เป็นองค์ประกอบของเครื่องฝึกบินจำลองให้สามารถทำงานร่วมกันได้อย่างสอดคล้อง เหมาะสม และมีประสิทธิภาพ

### 1.3 ขอบเขตของการวิจัย

สำหรับโครงการในระยะแรกนี้ จะทำการสร้างเครื่องจำลองการบินโดยมีชุดควบคุมการเคลื่อนไหวใน 3 ทิศทาง ส่วนชุดควบคุมการเคลื่อนที่ใน 6 ทิศทาง จะมีค่าใช้จ่ายสูงเกินไป จึงทำการกำหนดขอบเขตโครงการดังนี้

1. การแสดงผลหลักใช้ จอภาพ LCD เป็นส่วนแสดงผล 4 จอแสดงผล
2. เครื่องมือวัดและแสดงผลในห้องนักบินใช้ระบบที่แสดงผลบนจอ LCD และชุดเครื่องมือวัดประกอบการบิน
3. ติดตั้งระบบการเคลื่อนที่เพื่อจำลองการยกโครงสร้างห้องนักบิน ใน 3 ทิศทาง
  - 1) หมุนรอบแกน Y 2) หมุนรอบแกน X 3) เคลื่อนที่ตามแกน Z (Roll, Pitch, Heave)
4. ใช้ Joystick สำเร็จรูปที่มีขายในท้องตลาดโดยอาจมีการดัดแปลงรูปลักษณะภายนอกให้สมจริง
5. อุปกรณ์ต่าง ๆ ในห้องนักบินเน้นไปที่นักบินหลัก ไม่เน้นสำหรับนักบินผู้ช่วย
6. การบินเป็นแบบไม่มีการเชื่อมต่อกับเครื่องบินลำอื่น
7. ภาระกรรมรวม 200 กิโลกรัม
8. ควบคุมการเคลื่อนที่ของแผ่นเคลื่อนไหวโดยมีสัญญาณป้อนกลับจากมอเตอร์เท่านั้น

### 1.4 สถานที่ทำงานวิจัย

มหาวิทยาลัยเทคโนโลยีสุรนารี

### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่ทางผู้วิจัยและทางบริษัทคาดว่าจะได้จากงานวิจัยนี้ มีสิ่งที่สำคัญดังต่อไปนี้

1. ได้เครื่องฝึกบินจำลองที่นักศึกษาวิศวกรรมศาสตร์ สามารถใช้ในการฝึกทักษะและความรู้พื้นฐานเกี่ยวกับการทำงานของอากาศยาน
2. สามารถพัฒนาสู่การค้าให้กับหน่วยงานที่มีวัตถุประสงค์ในการสร้างและพัฒนาบุคลากรด้านอากาศยานและการบิน
3. ได้แนวทางการพัฒนาเครื่องจักรเพื่อยกระดับจากเครื่องฝึกบินจำลองระดับห้องปฏิบัติการไปสู่ระบบที่สามารถใช้งานได้จริงในภาคอุตสาหกรรม

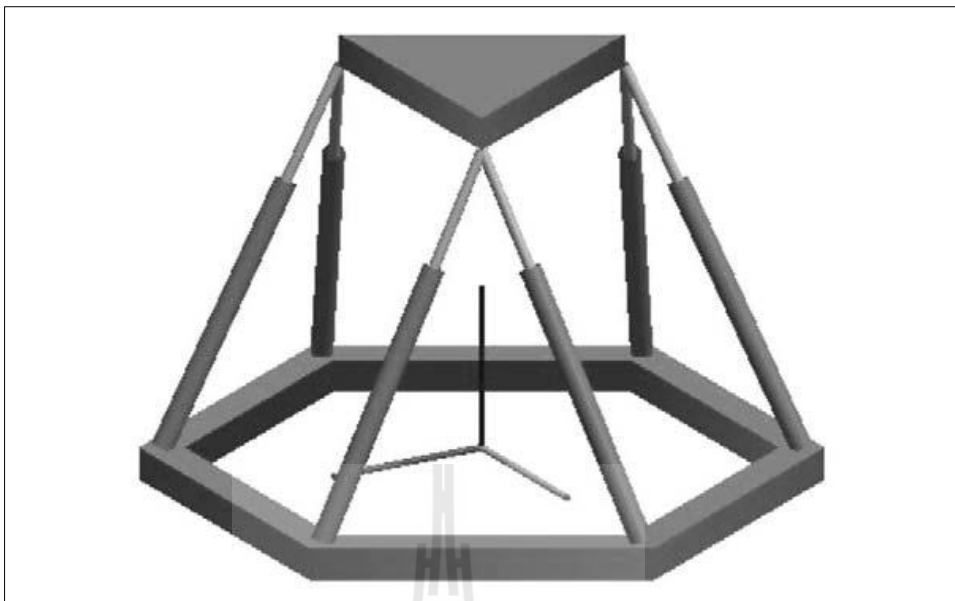
## บทที่ 2

### ปริทรรศน์วรรณกรรมและทฤษฎีที่เกี่ยวข้อง

#### 2.1 กล่าวนำ

ในส่วนแรกนี้เป็นส่วนที่ผู้วิจัยได้ทำการค้นคว้าเอกสาร รายงานการวิจัย วิทยานิพนธ์ต่าง ๆ ที่เกี่ยวข้องกับงานวิจัยที่ได้ดำเนินการอยู่ และจะพบว่ามีความและเอกสารต่าง ๆ ที่เกี่ยวข้องเป็นจำนวนมาก ดังนั้นผู้วิจัยจึงเลือกนำเสนองานวิจัยที่เกี่ยวข้องหรือสอดคล้องกับงานวิจัยที่ได้ดำเนินการอยู่นี้มาเสนอเท่านั้น

Wu and Gu (2007) ได้เสนอไว้ว่า Flight Simulator Motion Platform ivoว่าจะต้องเป็นกลไก โครงสร้างแบบขนาน เนื่องจากภาระกรรมที่ต้องกระทำค่อนข้างสูง จึงจำเป็นต้องใช้อุปกรณ์รับแรง ข้อต่อที่มีความยืดหยุ่นสูง และหากจำเป็นต้องการใช้ขีดความสามารถที่สูงที่สุดจะต้องใช้ลักษณะ ของโครงสร้างแบบ Stewart Platform ซึ่งมี 6 องศาอิสระในการเคลื่อนไหวซึ่งมีความยากในการ วิเคราะห์และคำนวณทางจลนศาสตร์ (Kinematics) และทางพลศาสตร์ของการคำนวณ โดยสมการที่ เกิดขึ้นจะเป็นสมการที่ไม่เป็นเชิงเส้น ฉะนั้นการแก้สมการก็จะต้องใช้ระเบียบวิธีเชิงตัวเลขในการ ทำงาน เช่น ระเบียบวิธีของ นิวตัน-ออยเลอร์ ในการหาคำตอบซึ่งอาจใช้เวลานานในการทำงาน อาจจะต้องใช้คอมพิวเตอร์ในการคำนวณที่มีสมรรถนะที่สูงขึ้นในการทำงานให้มีประสิทธิภาพและ ทันทีทันใด จากงานวิจัยได้นำเสนอวิธีการทำการ Simulation เพื่อทดสอบการเคลื่อนที่ของ Motion Platform ก่อน เพื่อตรวจสอบลักษณะการเคลื่อนไหว หรือขอบเขตของการเคลื่อนไหวที่สามารถทำ ได้ของแผ่นเคลื่อนไหวแบบ Stewart Platform



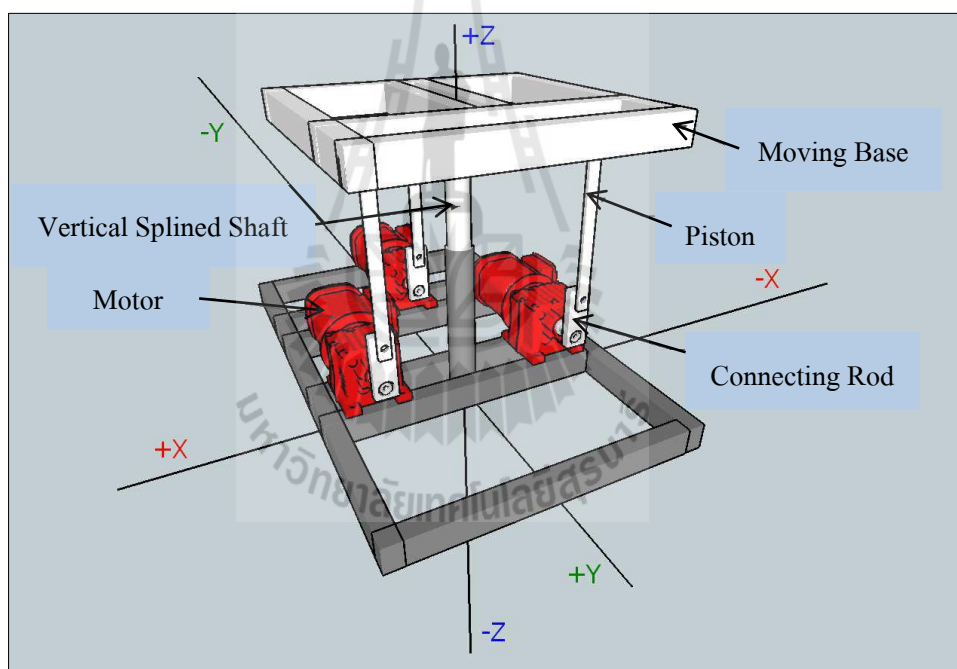
รูปที่ 2.1 แสดงลักษณะของ Stewart Platform

Kim et al. (2005) ได้เสนอว่า กลไกโครงสร้างแผ่นเคลื่อนไหวแบบขนานนั้นเป็นโครงสร้างที่สามารถให้แรงต่อน้ำหนักได้สูง และสามารถใช้ประโยชน์ได้ในหลากหลายรูปแบบ แต่ในทางเดียวกันเมื่อมีความสามารถสูง จึงจำเป็นต้องใช้ตัวควบคุมที่มีสมรรถนะสูงตามไปด้วยเช่นกัน โดยในลักษณะที่เรียกว่า Multi-input-multi-output (MIMO) และยังมีสมการการทำงานเป็นแบบไม่เป็นเชิงเส้นขึ้นสูงอีกด้วย ลักษณะการควบคุมแบ่งออกเป็นสองแบบ คือการควบคุมแบบทางตรง (Forward Kinematics) หรือ Joint Space Control คือการป้อนพารามิเตอร์ให้กับ Joint ทั้งหมดแล้วดูผลการตอบสนองที่ได้ ส่วนแบบที่สอง คือ Task Space Control คือระบุท่าทางที่ต้องการแล้วทำการประมวลผลเพื่อให้ได้พารามิเตอร์ในการหมุนของ Joint ต่าง ๆ ซึ่งจะมีความยากมากกว่ากันเป็นเท่าตัว โดยในงานวิจัยสร้างแบบจำลองขนาดเล็กขึ้นมาเพื่อศึกษาการตอบสนองของแผ่นเคลื่อนไหว และเพื่อศึกษาการออกแบบตัวควบคุมให้สามารถทำงานได้อย่างมีประสิทธิภาพ แต่ก็ยังคงมีค่าความคลาดเคลื่อนในการเคลื่อนไหวอยู่บ้าง แต่สามารถนำไปเป็นต้นแบบในการใช้งานจริงได้เช่นกัน

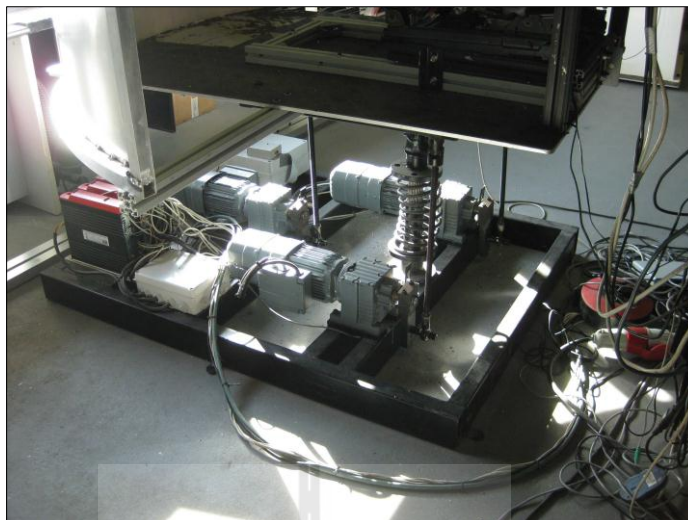
Casas, C., et al. (2014) ได้นำเสนอไว้ว่าโดยปกติแล้วขอบเขตการทำงานของแผ่นเคลื่อนไหวจะสามารถหาคำตอบได้เมื่อทำการสร้างและทดสอบแล้วเท่านั้น โดยในงานวิจัยได้นำเสนอการสร้างแบบจำลองแผ่นเคลื่อนไหวเสมือนจริงไว้เพื่อสามารถหาค่าขอบเขตการเคลื่อนไหวในแต่ละแนวแกนของแผ่นเคลื่อนไหวแทนการที่จะต้องสร้างอุปกรณ์จริงขึ้นมาก่อน จึงช่วยประหยัดทรัพยากรเป็นอย่างยิ่ง โดยสมัยก่อนจะทำการสร้างและทดสอบ เพื่อหาขอบเขตการทำงานและหากไม่ได้ตาม

ต้องการก็จะต้องทำการออกแบบใหม่ หรือพัฒนาใหม่เพื่อให้ได้ตามต้องการ ดังนั้นการสร้างแบบจำลองเสมือนจริงนี้จึงสามารถตอบโจทย์ได้อย่างดีในการหาขอบเขตการทำงานของแผ่นเคลื่อนไหวจริง

Dasgupta, B., and Mruthunjaya, T. S. (1998) ได้นำเสนอวิธีการทำ Simulation ของแผ่นเคลื่อนไหวนิค 6 UPS (Universal-prismatic-spherical) ซึ่งนำเสนอสมการการคำนวณเพื่อควบคุมแผ่นเคลื่อนไหวแบบ Task Space Control ไว้ ซึ่งไม่คิดแรงจากภายนอกในการวิเคราะห์ และนำเสนอผลการตอบสนองต่อสัญญาณป้อนเข้าแบบขั้นบันไดซึ่งเป็นไปได้อย่างมีประสิทธิภาพและถูกต้อง โดยตัวควบคุมใช้โปรแกรม MATLAB ในการคำนวณแก้สมการทางจลนศาสตร์ และพลศาสตร์ของแผ่นเคลื่อนไหว เพื่อให้ได้คำตอบของสมการที่ไม่เป็นเชิงเส้น เพื่อตั้งงาน Prismatic Joint ของแผ่นเคลื่อนไหวให้เคลื่อนไหวไปตามที่ต้องการ



รูปที่ 2.2 แสดงแบบจำลองแผ่นเคลื่อนไหวนิคสามองศาอิสระ



รูปที่ 2.3 แสดงแผ่นเคลื่อนไหวชนิดสามองศาอิสระขนาดจริง

Rekdalsbakken. (2005) ได้นำเสนอการสร้างสมการเพื่อควบคุมตำแหน่งของแผ่นเคลื่อนไหวที่มีสามองศาอิสระ โดยกำหนดให้มีการเคลื่อนที่ทั้งสามแนวแกนดังนี้ Roll Pitch และ Heave ด้วยวิธีการหมุนรอบแกนและเลื่อนที่ตามแนวแกน โดยพัฒนามาเป็นสมการ 9 สมการ เพื่อระบุตำแหน่งปัจจุบันของแผ่นเคลื่อนไหว เพื่อนำไปคำนวณเป็นมุมมองการหมุนของมอเตอร์ เพื่อให้แผ่นเคลื่อนไหวเคลื่อนที่ไปตามตำแหน่งที่ต้องการได้

Abdul Jalil, M.K. (2008) ได้เสนอการควบคุมแผ่นเคลื่อนไหวที่มี 6 องศาอิสระผ่านเซอร์โวมอเตอร์เพื่อควบคุมตำแหน่งการหมุนของมอเตอร์ด้วยตัวควบคุม PID ซึ่งสามารถทำงานได้อย่างมีประสิทธิภาพในการจำลองสถานการณ์ของเครื่องจำลองการขับของรถยนต์ ด้วยการรับค่าการขับชี้จากโปรแกรมจำลอง มาเพื่อคำนวณเป็นมุมการเคลื่อนที่ของเซอร์โวมอเตอร์ และเปลี่ยนเป็นสัญญาณควบคุมแบบ PWM ให้เซอร์โวมอเตอร์และมีสัญญาณตำแหน่งปัจจุบันป้อนกลับเพื่อทำงานเป็นวงปิด เพื่อให้แผ่นเคลื่อนไหวทำงานได้ตรงตามลักษณะที่ต้องการ

Chiew Y. S. (2008) ได้นำเสนอวิธีการสร้างสมการคณิตศาสตร์เพื่อจำลองแผ่นเคลื่อนไหวของ Stewart โดยวิธี Inverse Kinematics โดยเมื่อป้อนค่าสถานะของแผ่นเคลื่อนไหวยมุม Roll, Pitch และ Yaw ก็จะสามารถหาระยะหัดขีดของอุปกรณ์ทำงานได้อย่างดี โดยได้พัฒนาขึ้นในโปรแกรม MATLAB Simulink

กฤษฎา จันทร์แดง (2551) ได้ศึกษาและวิเคราะห์การจำลองพลศาสตร์ของพาหนะจริงกับพาหนะจริงกับพาหนะจริงย่อยส่วน ซึ่งเกี่ยวกับการพัฒนาขานยนต์ย่อยส่วนให้มีพฤติกรรมทางพลศาสตร์สมมูลกับพาหนะจริง ขานยนต์ย่อยส่วนนี้ถูกควบคุมด้วยมนุษย์ผ่านเครื่องจำลองเสมือน



จริงสามมิติ ส่วนที่สำคัญ คือ การออกแบบตัวกรองเสมือนจริงสามมิติให้เสมือนจริงมากที่สุด เครื่องจำลองในงานวิจัยนี้ สามารถเคลื่อนที่ในสามแนวแกน คือ หมุนรอบแกน x หมุนรอบแกน y และเคลื่อนที่ขึ้นลงในแนวดิ่ง ตัวแปรพลศาสตร์ที่สำคัญของยานพาหนะ เช่น ความเร่งเชิงเส้น ความเร็วเชิงมุม มุมเลี้ยว และความเร็วล้อ เป็นต้น ถูกวัดด้วยอุปกรณ์ตรวจวัดและส่งไปยังระบบควบคุมหลักตามกระบวนการในตัวกรองเสมือนจริง ซึ่งถูกออกแบบมาบนพื้นฐานของระบบการรับรู้ของมนุษย์

สุทธิวัฒน์ ประชาชนะชัย และ สุวัฒน์ กุศลนปรีดา (2548) นำเสนอการจำลองการทำงานในคอมพิวเตอร์ของแผ่นเคลื่อนไหว 6 องศาอิสระรูปแบบแกนกลโครงสร้างขนานที่ เรียกว่า Stewart Platform ในงานวิจัยโปรแกรมคอมพิวเตอร์จำลองการทำงาน ประกอบด้วยส่วนต่าง ๆ คือส่วน Washout Filter, สมการจลนศาสตร์ผกผัน, ตัวควบคุม, ระบบไฮดรอลิก และสมการพลวัตของแผ่นเคลื่อนไหว ผลลัพธ์การจำลองเพื่อศึกษาพฤติกรรมทางพลศาสตร์ของแผ่นเคลื่อนไหว โดยรวมเป็นที่น่าพอใจ และสามารถนำไปใช้ประโยชน์สำหรับการออกแบบเครื่องจำลองจริง เช่น เครื่องจำลองการขับรถยนต์ และเครื่องฝึกหัดนักบิน เป็นต้น โดยเฉพาะช่วยในการกำหนดขนาดชิ้นส่วนของแผ่นเคลื่อนไหว การเลือกอุปกรณ์ไฮดรอลิกที่เหมาะสม และการออกแบบตัวควบคุม

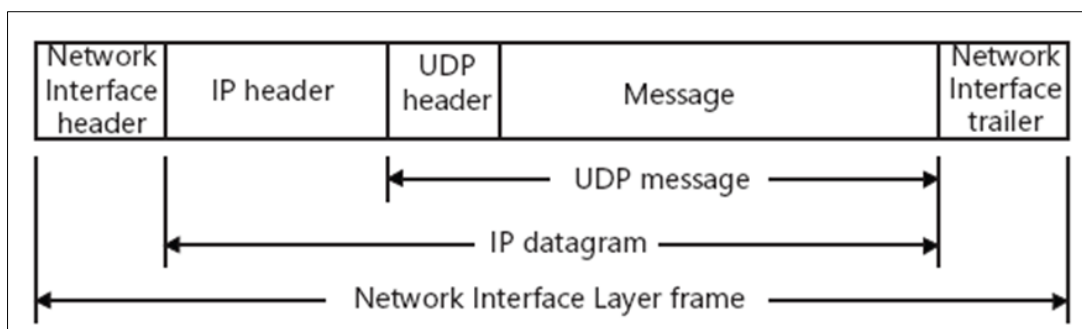
## 2.2 UDP Protocol

User Datagram Protocol หรือ UDP ที่เป็นส่วนหนึ่งของชุดโปรโตคอลอินเทอร์เน็ตที่ใช้โปรแกรมทำงานบนคอมพิวเตอร์เครื่องอื่นบนเครือข่ายสามารถส่งข้อความสั้นที่เรียกว่า Datagram มาจากอีกเครื่องหนึ่ง

UDP เป็นการใช้อยู่รูปแบบการสื่อสารที่เรียบง่าย โดยไม่ต้องตรวจสอบข้อความการส่ง สำหรับการยืนยันข้อมูลมีความน่าเชื่อถือและได้ข้อมูลที่สมบูรณ์ แม้ว่าปัจจัยเหล่านี้จะเหมือน UDP ไม่ได้เป็นโปรโตคอลที่มีความน่าเชื่อถือ ก็ยังคงพบการใช้ในพื้นที่กว้างโดยเฉพาะอย่างยิ่ง การส่งข้อมูลมีความเร็วเกินกว่าจะพิจารณาเพียงแค่ความน่าเชื่อถือ

### 2.2.1 UDP Message

ข้อมูลข่าวสารของ UDP ประกอบด้วย UDP Header และส่วนเก็บข้อมูล ซึ่งถูกกำหนดไว้ใน IP Header ด้วย IP Protocol หมายเลข 17 (0x11) ข้อมูลสามารถมีค่าสูงสุด 65,507 ไบต์ มาจาก 65,536 ลบด้วย IP Header ขนาดต่ำสุด (20 bytes) และ UDP Header (8 bytes) IP Datagram ที่จัดเก็บไว้ประกอบด้วย Network Interface Layer ที่จัดสรร Header และ Trailer รูปที่ 2.1 จะแสดงรายละเอียดในสายอักขระของข้อมูล

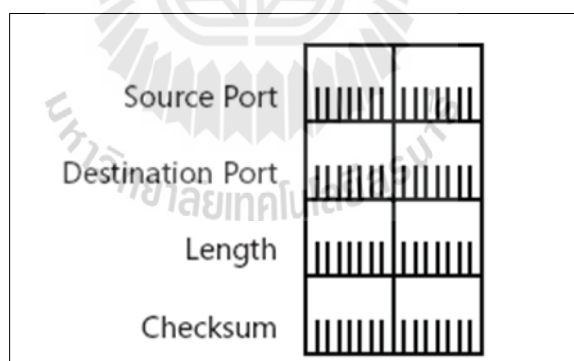


รูปที่ 2.4 แสดงการจัดเก็บข้อมูลใน UDP

ภายใน IP Header ของข้อมูลใน UDP ประกอบด้วย IP Address ฟิวด์ต้นทางที่แสดงการเชื่อมต่อของผู้ให้บริการที่ส่งข้อมูลใน UDP IP Address ฟิวด์ปลายทางที่แสดง Unicast Address ของผู้ให้บริการปลายทาง (หรือเราท์เตอร์ที่อยู่ตรงกลางถ้าหากแพ็กเก็ตเกิดเป็นเส้นทางของต้นทาง) IP Broadcast Address หรือ IP Multicast Address

### 2.2.2 UDP Header

UDP Header มีขนาดความยาวที่แน่นอน คือ 8 ไบต์ ประกอบด้วย 4 field ดังรูปที่ 2.5



รูปที่ 2.5 แสดงโครงสร้างของ UDP Header

ฟิวด์ของ UDP Header ได้กำหนดไว้ดังนี้

Source Port ฟิวด์ที่มี 2 ไบต์นี้ ระบุด้านทางของ Application Layer protocol ที่ส่งข้อมูลของ UDP การใช้งานของพอร์ตต้นทางเป็นอีกทางเลือก

Destination Port ฟิลด์ที่มี 2 ไบต์นี้ ระบุปลายทางของ Application Layer Protocol ส่วนประกอบของปลายทางของ IP Header IP Address และพอร์ตปลายทางของ UDP Header จัดไว้ให้เป็นพิเศษ ที่อยู่ที่มีความหมายโดยทั่วไปของกระบวนการจากสิ่งที่ส่งข้อความมา

Length ฟิลด์ที่มี 2 ไบต์นี้ ระบุความยาวในหน่วยไบต์ของข้อมูล UDP รวมทั้ง UDP Header และข้อมูลข่าวสาร ความยาวน้อยสุด คือ 8 ไบต์ (ขนาดของ UDP Header) และมากที่สุด คือ 65,515 ไบต์ ความยาวมากที่สุดที่แท้จริงถูกจำกัดโดย MTU ของลิงค์บนสิ่งที่ข้อมูล UDP ส่งออกไป ฟิลด์ความยาวจะมีมากเกินไป ความยาวของ UDP เป็นความยาวของที่มีอยู่ของ IP ที่สามารถคำนวณได้จากความยาวทั้งหมดและความยาวฟิลด์ของ IP Header (ความยาวของ UDP = ความยาวที่มีอยู่ = ความยาวทั้งหมด - 4 x ความยาวของ IP Header (ในรูปแบบ 32 บิต))

Checksum ฟิลด์ที่มี 2 ไบต์นี้ ระบุการตรวจสอบความสมบูรณ์ระดับบิตของข้อมูล UDP การคำนวณ UDP Checksum ใช้วิธีการคล้ายกับ IP Header Checksum บน Header เสมือนของ UDP, UDP Header, ข้อมูลข่าวสาร และถ้าจำเป็นใช้ไบต์รองรับของ 0 x 00 ไบต์รองรับจะใช้ในกรณีเดียวที่ความยาวของข้อมูลมีไบต์เป็นเลขคี่ สำหรับรายละเอียดอื่น ๆ เกี่ยวกับ Header เสมือนของ UDP ให้ดูจากหัวข้อ “Header เสมือนของ UDP” ในบทถัดไป การใช้งานของฟิลด์ UDP Checksum เป็นอีกทางเลือก ถ้าหากไม่ได้ใช้งานฟิลด์ UDP Checksum จะเซตให้เป็น 0 สำหรับรายละเอียดอื่น ๆ เกี่ยวกับการคำนวณ Checksum ดูในบทที่ 5 “Internet Protocol (IP)”

### 2.2.3 UDP Port

มีการกำหนดตำแหน่งหรือลำดับข้อมูลของการส่งข้อมูลของ Application Layer Protocols โดยการใช้บริการของ UDP พอร์ตต้นทางนับเป็นส่วนหนึ่งของแต่ละข้อมูลของ UDP และพอร์ตปลายทาง Internet Assigned Numbers Authority (IANA) กำหนดหมายเลขพอร์ตหรือที่รู้จักในหมายเลขพอร์ตที่เป็นที่รู้จักสำหรับ Application Layer protocols รูปที่ 2.3 แสดงหมายเลขพอร์ตที่เป็นที่รู้จักที่ถูกใช้โดย Windows Server 2008 และส่วนประกอบพื้นฐานของ Windows Vista

Port Number	Application Layer Protocol
53	DNS
67	BOOTP server (Dynamic Host Configuration Protocol [DHCP])
68	BOOTP client (DHCP)
69	TFTP
137	NetBIOS Name Service
138	NetBIOS Datagram Service
161	Simple Network Management Protocol (SNMP)
445	Direct hosting of Server Message Block (SMB) datagrams over TCP/IP (also known as Microsoft-DS)
520	RIP
1812, 1813	Remote Authentication Dial-In User Service (RADIUS)

รูปที่ 2.6 แสดงหมายเลขพอร์ต UDP ที่เป็นที่รู้จัก

แบบฉบับของฝ่ายผู้ให้บริการของ Application Layer Protocol จะรองรับเลขพอร์ตที่เป็นที่รู้จักกันกว้างขวางฝ่ายผู้ใช้งานของ Application Layer Protocol ก็จะใช้หมายเลขพอร์ตที่เป็นที่รู้จักกันดีเช่นกันหรือที่เป็นธรรมดามากกว่า, หมายเลขพอร์ตที่จัดสรรให้มีความคล่องตัว หมายเลขพอร์ตที่จัดสรรให้มีความคล่องตัวถูกใช้สำหรับช่วงเวลาของกระบวนการ และเป็นที่รู้จักกันในหมายเลขพอร์ตที่มีอายุสั้นหรือหมายเลขพอร์ตที่อยู่ได้ในช่วงสั้น ๆ

หมายเลขพอร์ตของ UDP สามารถถูกอ้างอิงโดยชื่อโดย Microsoft Windows Sockets Application (โดยใช้ GetServByname() function) ชื่อนี้ถูกแยกแยะเพื่อหมายเลขพอร์ตของ UDP ตามบริการเพิ่มข้อมูลที่เก็บไว้ใน %SystemRoot%\System32\Drivers\Etc folder

สรุปโปรโตคอล UDP ถูกพัฒนาให้เป็นโปรโตคอลสำหรับระบบงานที่ไม่ต้องการความเชื่อถือได้ระดับสูง อาจกล่าวได้ว่าโปรโตคอล UDP เป็นโปรโตคอลที่ง่ายที่สุดในชุดโปรโตคอล TCP/IP ทำให้ระบบงานที่ใช้โปรโตคอล UDP มีความเร็วในการรับส่งข้อมูลสูงเหมาะสมกับระบบงานเรียลไทม์

### 2.3 ไมโครคอนโทรลเลอร์ (Microcontroller)

ไมโครคอนโทรลเลอร์ คือ อุปกรณ์ที่สามารถสร้างระบบควบคุมได้ โดยอุปกรณ์นี้มีขนาดเล็กและเป็นอุปกรณ์ประเภทสารกึ่งตัวนำที่มีการรวมเอาฟังก์ชันการทำงานต่าง ๆ ไว้ในตัวมันเอง ซึ่งมีลักษณะคล้ายกับคอมพิวเตอร์ ซึ่งในที่นี้หมายถึงอุปกรณ์ภายในที่ประกอบด้วย หน่วยประมวลผลกลาง, พอร์ตในการเชื่อมต่อแบบต่าง ๆ

### 2.3.1 ส่วนประกอบทั่วไปของไมโครคอนโทรลเลอร์

1. หน่วยประมวลผลกลาง (Control Processing Unit)
2. หน่วยความจำ ซึ่งประกอบด้วย RAM (Random Access Memory) และ EPROM/ EPROM/PROM/ROM (Erasable Programmable Read Only Memory)
3. หน่วยรับและแสดงผลข้อมูล (Input/Output) ซึ่งมีพอร์ตขยายแบบขนาน (Parallel) และอนุกรม (Serial)

4. ตัวนับเวลา (Timer)

5. หน่วยควบคุมการอินเทอร์รัปต์ (Interrupt Controller)

ส่วนประกอบเหล่านี้เป็นเพียงส่วนประกอบพื้นฐานของไมโครคอนโทรลเลอร์ ไมโครคอนโทรลเลอร์นั้นยังมีส่วนประกอบอย่างอื่นอีก เพื่อเพิ่มเติมความสามารถขึ้นอยู่กับไมโครคอนโทรลเลอร์แต่ละเบอร์ด้วย เช่น

- ADC (Analog to Digital) ส่วนภาครับสัญญาณอนาล็อกแปลงไปเป็นสัญญาณดิจิทัล
- DAC (Digital to Analog) ส่วนภาคส่งสัญญาณดิจิทัลแปลงไปเป็นสัญญาณอนาล็อก
- I2C (Inter Integrate Circuit Bus) เป็นการสื่อสารอนุกรมแบบซิงโครนัส (Synchronous) เพื่อใช้ติดต่อสื่อสารระหว่างไมโครคอนโทรลเลอร์ (MCU) กับอุปกรณ์ภายนอก ซึ่งถูกพัฒนาขึ้นโดยบริษัท Philips Semiconductors โดยใช้สายสัญญาณเพียง 2 เส้นเท่านั้น คือ Serial Data (SDA) และสาย Serial Clock (SCL) ซึ่งสามารถเชื่อมต่ออุปกรณ์จำนวนหลาย ๆ ตัวเข้าด้วยกันได้ ทำให้ MCU ใช้พอร์ตเพียง 2 พอร์ตเท่านั้น

- SPI (Serial Peripheral Interface) เป็นการเชื่อมต่อกับอุปกรณ์เพื่อรับส่งข้อมูลแบบซิงโครนัส (Synchronize) มีสัญญาณนาฬิกาเข้ามาเกี่ยวข้องกับระหว่างไมโครคอนโทรลเลอร์ (Microcontroller) หรือจะเป็นอุปกรณ์ภายนอกที่มีการรับส่งข้อมูลแบบ SPI อุปกรณ์ที่ทำหน้าที่เป็นมาสเตอร์ (Master) โดยปกติแล้วจะเป็นไมโครคอนโทรลเลอร์ หรืออาจกล่าวได้ว่าอุปกรณ์ Master จะต้องควบคุมอุปกรณ์ Slave ได้ โดยปกติตัว Slave มักจะเป็นไอซี (IC) หน้าทีพิเศษต่าง ๆ เช่น ไอซีอุณหภูมิ, ไอซีฐานเวลานาฬิกาจริง (Real-Time Clock) หรืออาจเป็นไมโครคอนโทรลเลอร์ที่ทำหน้าที่ในโหมด Slave ก็ได้เช่นกัน

- PWM (Pulse Width Modulation) การสร้างสัญญาณพัลส์แบบสแควร์เวฟที่สามารถปรับเปลี่ยนความถี่และ Duty Cycle ได้เพื่อนำไปควบคุมอุปกรณ์ต่าง ๆ เช่น มอเตอร์

- UART (Universal Asynchronous Receiver Transmitter) ทำหน้าที่รับส่งข้อมูลแบบอะซิงโครนัสสำหรับมาตรฐานการรับส่งข้อมูลแบบ RS-232

### 2.3.2 ประเภทของไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ มีด้วยกันหลายประเภทแบ่งตามสถาปัตยกรรม (การผลิต และกระบวนการทำงานระบบการประมวลผล) ที่มีใช้ในปัจจุบันยกตัวอย่างดังนี้

1. ไมโครคอนโทรลเลอร์ตระกูล PIC (บริษัทผู้ผลิต Microchip)
2. ไมโครคอนโทรลเลอร์ตระกูล MCS51 (บริษัทผู้ผลิต Atmel, Philips)
3. ไมโครคอนโทรลเลอร์ตระกูล AVR (บริษัทผู้ผลิต Atmel)
4. ไมโครคอนโทรลเลอร์ Arduino (บริษัทผู้ผลิต Atmel) เป็นการนำ AVR มาเขียน Firmware Boot Loader คำสั่งใช้งานใหม่ ซึ่งข้อแตกต่างสำหรับบอร์ด Arduino กับบอร์ดทดลองทั่วไป คือ เป็น Open Source ซึ่งหมายถึง การเปิดเผยวิธีการสร้างทั้งในส่วนของ Hardware ไม่ว่าจะ เป็นวงจรต่าง ๆ ภายในบอร์ด และตัว Tool Software ที่จะเป็นเครื่องมือที่จะใช้ในการเขียนคำสั่งต่าง ๆ ลงบอร์ด ล้วนเป็น Open Source ทั้งหมด ส่งผลให้นักวิจัยทุกคนมีสิทธิ์ที่จะดาวน์โหลด โปรแกรมต่าง ๆ ที่เกี่ยวกับการพัฒนา Arduino ไปใช้ฟรี และสามารถสร้างวงจรเองจากต้นแบบ ใช้ฟรีได้เช่นกัน ซึ่งข้อดี คือ เป็นการสร้างมาตรฐานในการพัฒนาที่จะเป็นไปในทิศทางเดียวกัน เพราะนักพัฒนาทั่วโลกนิยมใช้งานบอร์ดชนิดนี้กันมากขึ้น ดังนั้นไม่ว่าจะเป็นในเรื่องของตัวอย่าง Source Code ตลอดจนการพูดคุยปัญหาเรื่องการใช้งาน และตัวอย่างการต่อวงจรเพื่อการใช้งานในรูปแบบต่าง ๆ จะถูกพัฒนาภายใต้ Open Source ของบอร์ดนี้

5. ไมโครคอนโทรลเลอร์ตระกูล ARM7, ARM9 (บริษัทผู้ผลิต Atmel, Philips, Analog Device, Samsung, STMicroelectronics)
6. ไมโครคอนโทรลเลอร์ตระกูล Basic Stamp (บริษัทผู้ผลิต Parallax)
7. ไมโครคอนโทรลเลอร์ตระกูล PSOC (บริษัทผู้ผลิต CYPRESS)
8. ไมโครคอนโทรลเลอร์ตระกูล MSP (บริษัทผู้ผลิต Texas Instruments)
9. ไมโครคอนโทรลเลอร์ตระกูล 68HC (บริษัทผู้ผลิต MOTOROLA)
10. ไมโครคอนโทรลเลอร์ตระกูล H8 (บริษัทผู้ผลิต Renesas)
11. ไมโครคอนโทรลเลอร์ตระกูล RABBIT (บริษัทผู้ผลิต RABBIT SEMICONDUCTOR)
12. ไมโครคอนโทรลเลอร์ตระกูล Z80 (บริษัทผู้ผลิต Zilog)

### 2.3.3 ภาษาที่ใช้เขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์

1. ภาษา Assembly
2. ภาษา Basic
3. ภาษา C หรือ C++
4. ภาษา Pascal

ภาษาดังกล่าวที่กล่าวในเบื้องต้น ไมโครคอนโทรลเลอร์บางตระกูล จะใช้ได้ครบทุกภาษา แต่บางตระกูลจะใช้ได้บางภาษา ขึ้นอยู่กับบริษัทผู้ผลิต Software (โดยทั่วไปจะเรียกว่า Editor and Compiler) ที่ใช้เขียนภาษาไมโครคอนโทรลเลอร์นั้นจะผลิตออกมาให้ Support หรือไม่

## 2.4 เซอร์โวมอเตอร์ (Servo Motor)

หลักการพื้นฐานของ Servo Motor และระบบควบคุมในปัจจุบันการนำเอาระบบ Automation System เข้ามาใช้ในระบบงานการผลิต การลำเลียงจัดส่ง การจัดเก็บในโรงงานอุตสาหกรรมมากขึ้น ตลอดจนมีการพัฒนาปรับปรุงระบบแบบเดิมที่ไม่มีใช้ มาใช้มากขึ้นด้วย หนึ่งในระบบ Automation System ที่มีการนำมาใช้ นอกเหนือไปจากพวกระบบ Pneumatic และ Hydraulic ก็คือ Servo Motor & Control ซึ่งเป็นระบบที่นำมาใช้อย่างกว้างขวางมากในแทบทุกอุตสาหกรรมเช่น Packaging Machinery, Automate Manufacturing, Printing, Labeling, Textile Machinery, Food Processing, Plastic Machinery, Metal Forming รวมถึงงานที่มีการประยุกต์ใช้ เช่น Pick and Place, Flying Cutoffs, X-Y-Z Table, Synchronized Speed

การใช้ Servo Motor มีการใช้อย่างกว้างขวางก็เพราะมีความยืดหยุ่นในการใช้งานสูงสามารถควบคุมได้ทั้ง ความเร็ว (Speed), ตำแหน่ง (Position) และแรงบิด (Torque) ซึ่งให้ประสิทธิภาพสูงสุดในกระบวนการควบคุมมอเตอร์ เมื่อเปรียบเทียบกับมอเตอร์ธรรมดา (AC MOTOR) หรือมอเตอร์กระแสตรง (DC MOTOR) การใช้ Servo Motor & Control มีส่วนประกอบหลัก ๆ 2 ส่วนหลัก ๆ คือ

### 2.4.1 Servo Motor

ซึ่งมีขนาดตั้งแต่ 30 W ขึ้นไปจนถึงกว่า 370 kW โครงสร้างภายในจะแตกต่างจากมอเตอร์เหนี่ยวนำโดยที่โรเตอร์ (Rotor) จะเป็นแม่เหล็กถาวร (Permanent Magnet) การออกแบบลดแรงเฉื่อยให้มากที่สุด ซึ่งจะเห็นจากรูปทรงเหลี่ยมและยาว ซึ่งมีผลทำให้การตอบสนองต่ออัตราเร่ง (Accelerate), ความเร็ว (Speed) และอัตราหน่วง (Decelerate) ทำได้ดีเมื่อเทียบกับมอเตอร์เหนี่ยวนำหรือมอเตอร์ทั่ว ๆ ไป การออกแบบจะมีสัญญาณป้อนกลับทั้งความเร็วและตำแหน่ง ที่ท้ายมอเตอร์ (ซึ่งอาจจะเป็น Encoder, Resolver หรือประเภทอื่น ๆ ก็ได้)

### 2.4.2 โครงสร้างภายในของ Servo Motor

AC Servo Motor, DC Brushless Motor จัดอยู่ในกลุ่มมอเตอร์ประเภท Synchronous Motor คือ ความเร็ว Rotor จะมีค่าเท่ากับความเร็วสนามแม่เหล็กในมอเตอร์ (Rotating Field)

### 2.4.3 Amplifier หรือ Driver

เป็นตัวขับ Servo Motor ใช้ Servo Motor โดยต่อไฟโดยตรงเหมือนกับมอเตอร์ธรรมดา ไม่ได้ต้องต่อผ่าน Amplifier ซึ่งจะเป็นตัวจ่ายไฟและรับสัญญาณป้อนกลับจากตัวเซอร์โวมอเตอร์

(Encoder feedback) ถ้ามีเฉพาะ Servo Motor และ Amplifier เราควบคุมได้ แต่ความเร็ว (Speed) และแรงบิด (Torque) เท่านั้น คล้าย ๆ กับการใช้ Inverter แต่แตกต่างกันที่จะควบคุมรอบได้เที่ยงตรงแม่นยำกว่ามาก และมีแรงบิดคงที่ ตลอดช่วงความเร็วของมอเตอร์ หากต้องการควบคุมตำแหน่งด้วย (Position Control) จะต้องมี Standalone Controller หรือ Built-in Controller หรือ PLC with Module Controller หรือ PC with PCI Card Controller

ดังนั้นในระบบ Servo Motor อย่างน้อยที่สุด จะต้องประกอบด้วย Servo Motor และ Servo Driver สำหรับงานที่ต้องการควบคุมแค่ Speed หรือ Torque การเลือก Servo Driver ใช้กับ Servo Motor ก็เหมือนกับการเลือก Inverter ใช้กับมอเตอร์อินดักชัน โดยเลือกให้ Servo กำลังขับมากกว่าหรือเท่ากับของ Servo Motor สิ่งที่แตกต่างกันและต้องระมัดระวังก็คือ โดยทั่วไปมักไม่มีปัญหาในการใช้ Inverter ยี่ห้อหนึ่งกับมอเตอร์ยี่ห้อหนึ่ง แต่กับ Servo Motor ไม่ใช่ มักใช้ Driver และ Servo Motor ยี่ห้อเดียวกัน หรือที่จับคู่กันแล้วโดยผู้ผลิต ทั้งนี้เป็นเพราะปัจจัยหลักดังนี้

Feedback ที่ Servo Motor ติดอยู่ เป็นแบบชนิดเดียวกันหรือไม่: Incremental Encoder, Resolver, Serial Encoder ถ้าใช้ Servo Drive ที่รับ Feedback จากมอเตอร์เป็น Resolver ไปใช้กับ Servo Motor ที่ติด Encoder ก็จะไม่สามารถใช้งานกันได้ ดังนั้นการจับคู่ Servo Motor และ Servo Driver จะต้องพิจารณาปัจจัยข้างต้นด้วย เมื่อต้องใช้ต่างยี่ห้อกันหรือใช้ทดแทนกัน

## 2.5 ทฤษฎีระบบควบคุม (Control Theory)

ทฤษฎีระบบควบคุม เป็นสาขาหนึ่งของคณิตศาสตร์และวิศวกรรมศาสตร์ ในที่นี้ การควบคุม หมายถึง การควบคุมระบบพลศาสตร์ ให้มีค่าเอาต์พุตที่ต้องการ โดยการป้อนค่าอินพุตที่เหมาะสมให้กับระบบ เช่น ระบบควบคุมอุณหภูมิห้องของเครื่องปรับอากาศ หรือแม้แต่ลูกกลอยในโถส้วมที่เปิดน้ำปิดน้ำโดยอัตโนมัติเมื่อน้ำหมดและน้ำเต็ม

การควบคุมการขับเคลื่อนยานพาหนะ เช่น รถยนต์ ก็ถือเป็นการควบคุมชนิดหนึ่ง โดยผู้ขับขี่เป็นผู้ควบคุมทิศทางและความเร็ว ซึ่งระบบควบคุมประเภทนี้ต้องมีคนเข้ามาเกี่ยวข้องนี้ถือว่าเป็นระบบควบคุมไม่อัตโนมัติ (Manual Control) แต่ทฤษฎีระบบควบคุมจะครอบคลุมเฉพาะการวิเคราะห์และออกแบบระบบควบคุมอัตโนมัติ (Automatic Control) เท่านั้น เช่น ระบบขับเคลื่อนอัตโนมัติ (Cruise Control)

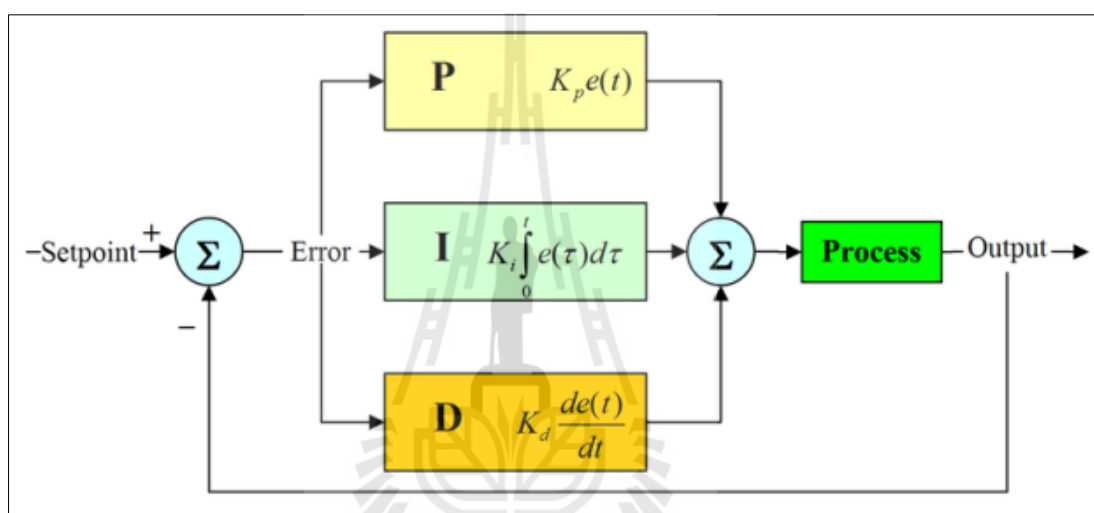
ระบบควบคุมยังอาจแบ่งออกได้เป็นระบบควบคุมวงเปิด (Open-Loop Control) คือ ระบบควบคุมที่ไม่ได้ใช้สัญญาณจากเอาต์พุต มาบ่งชี้ถึงลักษณะการควบคุม ส่วนระบบควบคุมวงปิด (Closed-Loop Control) หรือระบบป้อนกลับ (Feedback Control) นั้นจะใช้ค่าที่วัดจากเอาต์พุตมาคำนวณค่าการควบคุม นอกจากนี้ยังอาจแบ่งได้ตามคุณลักษณะของระบบ เช่น เป็นเชิงเส้น (Linear),



ไม่เป็นเชิงเส้น (Nonlinear), แปรเปลี่ยนตามเวลา (Time-varying), ไม่เปลี่ยนแปลงตามเวลา (Time-Invariant) และเวลาต่อเนื่อง (Continuous Time), เวลาไม่ต่อเนื่อง (Discontinuous Time)

### 2.5.1 ทฤษฎีควบคุมแบบสัดส่วน-ปริพันธ์-อนุพันธ์ (PID Controller Theory)

ระบบควบคุมแบบสัดส่วน-ปริพันธ์-อนุพันธ์ เป็นระบบควบคุมแบบป้อนกลับที่ใช้กันอย่างกว้างขวาง ซึ่งค่าที่นำไปใช้ในการคำนวณเป็นค่าความผิดพลาดที่หามาจากความแตกต่างของตัวแปรในกระบวนการและค่าที่ต้องการ ตัวควบคุมจะพยายามลดค่าผิดพลาดให้เหลือน้อยที่สุดด้วยการปรับค่าสัญญาณขาเข้าของกระบวนการ ค่าตัวแปรของ PID ที่ใช้จะปรับเปลี่ยนตามธรรมชาติของระบบ



รูปที่ 2.7 แสดงแผนภาพบล็อกของการควบคุมแบบพีไอดี

วิธีคำนวณของ PID ขึ้นอยู่กับสามตัวแปร คือ ค่าสัดส่วน, ปริพันธ์ และอนุพันธ์ ค่าสัดส่วนกำหนดจากผลของความผิดพลาดในปัจจุบัน ค่าปริพันธ์กำหนดจากผลบนพื้นฐานของผลรวมความผิดพลาดที่ซึ่งพ่วงผ่านไป และค่าอนุพันธ์กำหนดจากผลบนพื้นฐานของอัตราการเปลี่ยนแปลงของค่าความผิดพลาด น้ำหนักที่เกิดจากการรวมกันของทั้งสามนี้จะใช้ในการปรับกระบวนการ

โดยการปรับค่าคงที่ใน PID ตัวควบคุมสามารถปรับรูปแบบการควบคุมให้เหมาะสมกับที่กระบวนการต้องการได้ การตอบสนองของตัวควบคุมจะอยู่ในรูปของการไหวตัวของตัวควบคุมจนถึงค่าความผิดพลาด ค่าโอเวอร์ชูต (Overshoots) และค่าแกว่งของระบบ (Oscillation) วิธี PID ไม่รับประกันได้ว่าจะเป็นระบบควบคุมที่เหมาะสมที่สุด หรือสามารถทำให้กระบวนการมีความเสถียรแน่นอน

การประยุกต์ใช้งานบางครั้งอาจใช้เพียงหนึ่งถึงสองรูปแบบ ขึ้นอยู่กับกระบวนการ เป็นสำคัญ พีไอดีบางครั้งจะถูกเรียกว่าการควบคุมแบบ PI, PD, P หรือ I ขึ้นอยู่กับว่าใช้รูปแบบใดบ้าง

การควบคุมแบบ PID ได้ชื่อตามการรวมกันของเทอมของตัวแปรทั้งสามตาม

$$MV(t) = P_{out} + I_{out} + D_{out} \quad (2.1)$$

โดยที่  $P_{out}, I_{out}, D_{out}$  คือ ผลของสัญญาณขาออกของระบบ

การควบคุมแบบ PID จะประกอบไปด้วย ส่วนการควบคุมที่สำคัญด้วยกัน คือ

- Proportional Control Action (P-Action)
- Integral Control Action (I-Action)
- Derivative Control Action (D-Action)

P Action เป็นการกำหนดการทำงานของ Output ให้เป็นสัดส่วนเปอร์เซ็นต์กับค่า Error หรือการเปลี่ยนแปลงของค่าที่วัดได้

$$Output = \frac{(error \cdot 100)}{P_b} \quad (2.2)$$

โดยที่ Error = ค่า Set Point - ค่าที่วัดได้

ในทางปฏิบัติ P Action จะเข้าไปใกล้ค่าหนึ่ง ซึ่งไม่ใช่ค่า Set Point จริง ซึ่งเรียกว่าค่า Offset เทอมของสัดส่วน หรือ P บางครั้งเรียกว่า "อัตราขยาย" จะเปลี่ยนแปลงเป็นสัดส่วนของค่าความผิดพลาด การตอบสนองของสัดส่วนสามารถทำได้โดยการคูณค่าความผิดพลาดด้วยค่าคงที่  $K_p$ , หรือที่เรียกว่าอัตราขยายสัดส่วน เทอมของสัดส่วนจะเป็นไปตามสมการ

$$P_{out} = K_p e(t) \quad (2.3)$$

เมื่อ	$P_{out}$	คือ	สัญญาณขาออกของเทอมสัดส่วน
	$K_p$	คือ	อัตราขยายสัดส่วน, ตัวแปรปรับค่าได้
	$e$	คือ	ค่าความผิดพลาด = SP - PV
	$t$	คือ	เวลา

I Action จะใช้ในการแก้ปัญหา Offset ระบบควบคุม I Action จะเข้าไปช่วยกำจัดค่า Offset ที่ยังคงมีอยู่ให้ระบบเข้าสู่ Set Point โดยค่า Output ที่ออกมาจะขึ้นอยู่กับ Integral Time ที่กำหนดขึ้นมาตั้งแต่ต้น หากกำหนดให้ Integral Time น้อย ระบบจะเข้าสู่ Set Point ได้อย่างรวดเร็ว แต่จะเกิดการกระเพื่อม Hunting ของ Process มากด้วย และหากกำหนดให้ Integral Time มากจะเกิด Hunting น้อย แต่จะใช้เวลานานกว่าระบบจะเข้าสู่ Set Point เทอมปริพันธ์จะเป็นไปตามสมการ

$$I_{out} = K_i \int_0^t e(\tau) d\tau \quad (2.4)$$

เมื่อ	$I_{out}$	คือ	สัญญาณขาออกของเทอมปริพันธ์
	$K_i$	คือ	อัตราขยายปริพันธ์, ตัวแปรปรับค่าได้
	$e$	คือ	ความผิดพลาด = SP-PV
	$t$	คือ	เวลา
	$\tau$	คือ	ตัวแปรปริพันธ์หุ่น

D Action ในกรณีที่มีการรบกวนระบบจากภายนอก Disturbance เป็นผลให้ Process ของระบบ มีการเปลี่ยนแปลงอย่างทันทีทันใด เราควรจะใช้การควบคุมแบบ D Action Derivative ซึ่งจะมีการตอบสนองที่รวดเร็ว เป็นผลให้ระบบเข้าสู่ Set Point ได้รวดเร็วขึ้น

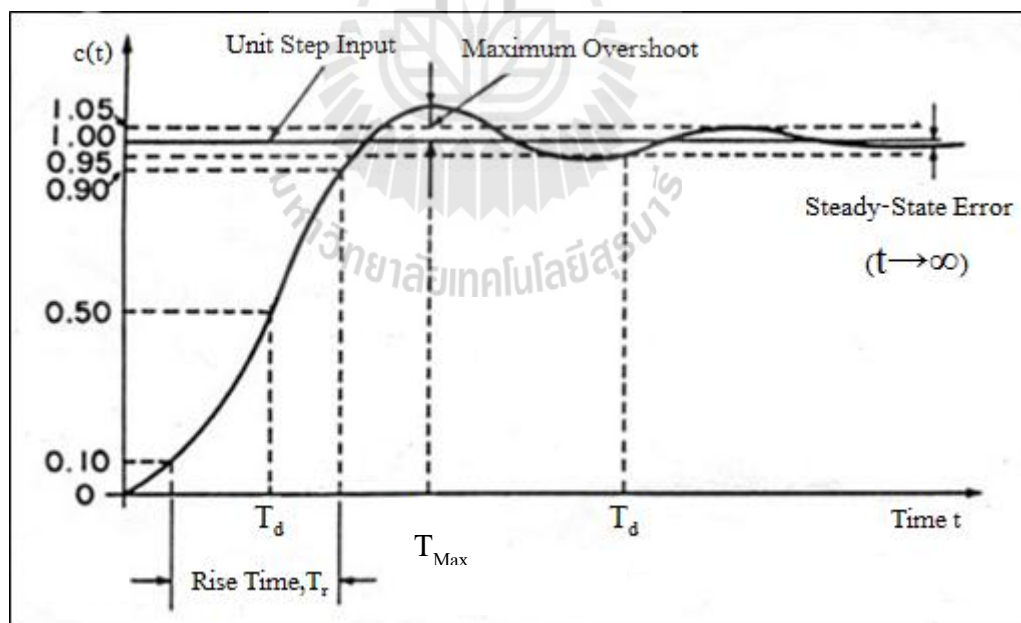
อัตราการเปลี่ยนแปลงของความผิดพลาดจากกระบวนการนั้นคำนวณหาจากความชันของความผิดพลาดทุก ๆ เวลา (นั่นคือ เป็นอนุพันธ์อันดับหนึ่งสัมพันธ์กับเวลา) และคูณด้วยอัตราขยายอนุพันธ์ ขนาดของผลของเทอมอนุพันธ์ (บางครั้งเรียก อัตรา) ขึ้นกับอัตราขยายอนุพันธ์ เทอมอนุพันธ์เป็นไปตามสมการ

$$D_{out} = K_d \frac{d}{dt} e(t) \quad (2.5)$$

เมื่อ	$D_{out}$	คือ	สัญญาณขาออกของเทอมอนุพันธ์
	$K_d$	คือ	อัตราขยายอนุพันธ์, ตัวแปรปรับค่าได้
	$e$	คือ	ความผิดพลาด = SP-PV
	$t$	คือ	เวลา

เทอมอนุพันธ์จะชะลออัตราการเปลี่ยนแปลงของสัญญาณขาออกของระบบควบคุม และด้วยผลนี้จะช่วยให้ระบบควบคุมเข้าสู่จุดที่ต้องการ ดังนั้นเทอมอนุพันธ์จะใช้ในการลดขนาดของโอเวอร์ชูตที่เกิดจากเทอมปริพันธ์ และทำให้เสถียรภาพของการรวมกันของระบบควบคุมดีขึ้น แต่อย่างไรก็ตามอนุพันธ์ของสัญญาณรบกวนที่ถูกขยายในระบบควบคุมจะไวมากต่อการรบกวนในเทอมของความผิดพลาดและสามารถทำให้กระบวนการไม่เสถียรได้ถ้าสัญญาณรบกวน และอัตราขยายอนุพันธ์มีขนาดใหญ่เพียงพอ

การทดสอบประสิทธิภาพของระบบ Feedback เราจะใช้สัญญาณอินพุตแบบ Unit Step ใส่เข้าไปในระบบเพื่อใช้วิเคราะห์ Transient ในระบบควบคุมแบบป้อนกลับโดยสามารถตรวจจากลักษณะดังรูปที่ 2.8



รูปที่ 2.8 แสดงการตอบสนองต่อฟังก์ชันอินพุตพื้นฐาน

- Steady State Performance ค่านี้จะได้จากการหาค่า Steady-State Error ของการตอบสนองต่อฟังก์ชันอินพุตพื้นฐานที่กล่าวมาแล้ว
- Transient Performance ค่านี้จะวิเคราะห์โดยการใส่ Unit-Step Function เป็นอินพุตอ้างอิงและจะได้ผลตอบสนองดังรูป ลักษณะที่สำคัญของผลตอบสนองนี้ได้แก่
- Overshoot เป็นค่า Error ที่มากที่สุดระหว่างอินพุตและเอาต์พุต ค่านี้จะใช้ในการประมาณความเสถียรของระบบ ค่า Overshoot จะวัดเป็นสัดส่วนเทียบกับค่าสุดท้ายหรือค่าอินพุตอ้างอิงดังนี้

$$\text{Percent Overshoot} = \text{Maximum Overshoot} / \text{Final Desired Value} \times 100 \quad (2.6)$$

- Time Delay ค่า Time Delay ( $t_d$ ) เป็นช่วงเวลาที่ใช้ในการตอบสนองของระบบ ตั้งแต่เริ่มต้นจนกระทั่งเอาต์พุตมีค่าเป็น 50% ของค่าอินพุตอ้างอิง
- Rise Time ค่า Rise Time ( $t_r$ ) เป็นช่วงเวลาตั้งแต่เอาต์พุตมีค่าเป็น 10% จนถึง 90% ของค่าอินพุตอ้างอิง
- Setting Time ค่า Setting Time ( $t_s$ ) เป็นช่วงเวลาตั้งแต่เริ่มต้น จนกระทั่งการแกว่งของเอาต์พุตลดลงอยู่ในขอบเขตที่กำหนด โดยปกติแล้วขอบเขตนี้จะอยู่ในช่วง 2-5% ของอินพุตอ้างอิง นอกจากนี้ยังมีลักษณะที่สำคัญอื่น ๆ อีก เช่น Damping Ratio, Damping Factor และ Undamped Natural Frequency

## บทที่ 3

### วิธีการดำเนินการวิจัย

#### 3.1 กล่าวนำ

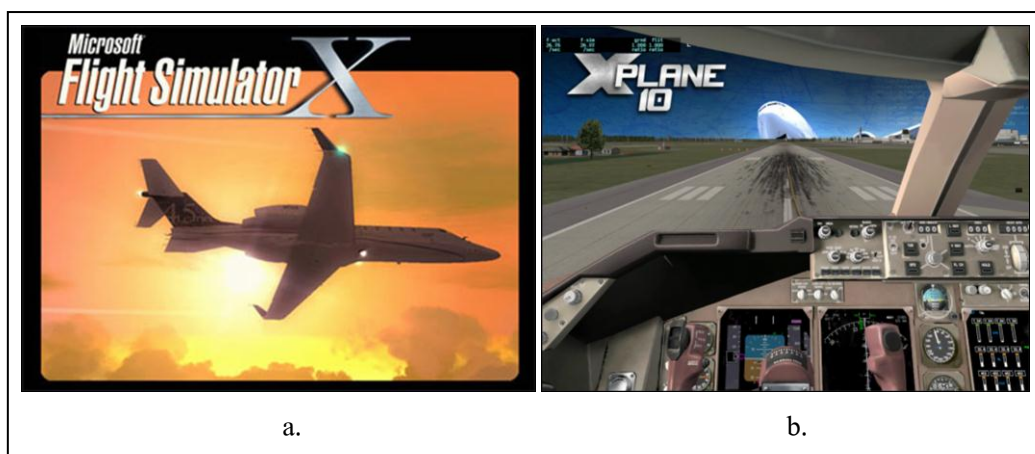
งานวิจัยนี้ได้ทำการสร้างเครื่องจำลองการเคลื่อนที่ของอากาศยานต้นแบบเพื่อนำข้อมูลที่ได้จากการศึกษาวิจัยไปพัฒนาเป็นเครื่องที่สามารถรองรับการใช้งานจริง โดยทางผู้วิจัยมีแนวคิดที่จะสร้างระบบจำลองการบิน โดยประกอบด้วยการนำข้อมูลขณะทำการบินแสดงผลออกทางจอแสดงผลทั้ง 4 จอ พร้อมกับแสดงผลที่เครื่องวัดประกอบการบินแบบต่าง ๆ การออกแบบแขนกลโครงสร้างขนานสำหรับใช้ทำหน้าที่เคลื่อนไหวเครื่องจำลองการบินต้นแบบ และการเชื่อมโยงการสื่อสารระหว่างห้องนักบินกับแขนกลโครงสร้างขนาน

ซึ่งงานวิจัยนี้ได้มุ่งเน้นการเชื่อมโยงการสื่อสารระหว่างห้องนักบินกับแขนกลโครงสร้างขนานให้สามารถตอบสนองกัน และกันในช่วงเวลาที่สามารถยอมรับได้ โดยจะแยกตามหัวข้อดังต่อไปนี้

#### 3.2 โปรแกรมจำลองการบิน

โปรแกรมจำลองการบินเป็นอุปกรณ์ที่นำมาจำลองสถานการณ์ที่นักบินจะต้องพบ เมื่อทำการบินจริงไปบนท้องฟ้า ไม่ว่าจะเป็นสภาพอากาศที่เลวร้าย หรือจะเป็นลักษณะของสนามบินแต่ละแห่ง รวมไปถึงรถและคนที่ทำหน้าที่ต่างภายในสนามบินด้วย การจำลองสถานการณ์แบบนี้ จะช่วยในการฝึกให้นักบินสามารถบังคับเครื่องบินได้อย่างปลอดภัย และไปยังที่หมายที่ไม่เคยไปมาก่อนได้ดีกว่าเดิม โดยที่มีค่าใช้จ่ายที่น้อยกว่าการที่จะฝึกบินด้วยเครื่องบินจริง ในปัจจุบันมีการนำเครื่องจำลองการบินมาใช้ในการฝึกนักบินใหม่ เพื่อประหยัดค่าใช้จ่ายในการฝึกบิน และปลอดภัยกว่ามาก รวมไปถึงการนำเครื่องจำลองการบินมาฝึกนักบินรบ เพื่อจำลองการจู่โจมเป้าหมาย หรือฝึกบินเครื่องบินรบที่ทันสมัย โดยที่ไม่ต้องทำการบินจริงเพื่อความปลอดภัยของนักบินได้อีกด้วย

ระบบจำลองการบินที่มีวางขายในท้องตลาดมีมากมาย หลายบริษัทที่ผลิตออกมาวางขาย ซึ่งแต่ละโปรแกรมจะมีข้อเด่นข้อด้อยต่างกันไป มีจำนวนรุ่นเครื่องบินที่แตกต่างกัน และมีการแสดงผลรายละเอียดของฉากที่ต่างกัน ซึ่งในงานวิจัยนี้เลือกมาศึกษา 2 โปรแกรม คือ โปรแกรม Microsoft Flight Simulator X (FSX) และ X-Plane ดังแสดงในรูปที่ 3.1



รูปที่ 3.1 แสดงโปรแกรมจำลองการบิน (a.) Microsoft Flight Simulator X (FSX) (b.) X-Plane

### 3.2.1 Microsoft Flight Simulator X (FSX)

โปรแกรมจำลองการขับเครื่องบินเสมือนจริง สามารถขึ้นบินจากสนามบินหนึ่งไปลงอีกสนามบินหนึ่งทั่วโลกได้ และยังสามารถพบปะกับเพื่อนได้ด้วยการบินออนไลน์ มีสนามบินมากกว่า 1,000 สนามบินจาก 223 ประเทศทั่วโลก 24 อากาศยานที่สามารถบังคับได้ สถานที่สำคัญในโลกใบนี้ นอกจากนั้นยังได้รับการยอมรับจากโรงเรียนการบินทั่วโลก รวมทั้งประเทศไทยอีกด้วย จะเห็นได้จากที่ทหารอากาศใช้โปรแกรมนี้ ฝึกนักบินก่อนจะไปบินกับเครื่องบินจริง

ความต้องการขั้นต่ำของโปรแกรม Microsoft Flight Simulator X ในการติดตั้งโปรแกรมนั้น ควรตรวจสอบเครื่องคอมพิวเตอร์ว่ามีคุณสมบัติตรงตามความต้องการขั้นต่ำของโปรแกรมเพื่อที่จะสามารถติดตั้งโปรแกรม และใช้งานได้อย่างเต็มประสิทธิภาพ ซึ่งคุณสมบัติดังกล่าวแสดงดังตารางที่

3.1

ตารางที่ 3.1 ความต้องการขั้นต่ำของโปรแกรม Microsoft Flight Simulator X.

ระบบปฏิบัติการ	Windows 7, Windows Vista หรือ Windows XP Service Pack ที่ 2 (SP2)
ตัวประมวลผลของคอมพิวเตอร์	1.0 กิกะเฮิร์ตซ์ (GHz) หรือมากกว่า
หน่วยความจำ	RAM อย่างน้อย 1 กิกะไบต์ (GB)
เนื้อที่ว่างบนฮาร์ดดิสก์	15 กิกะไบต์ (GB)
การ์ดแสดงผล	DirectX® 9.0 C compliant video with 32 MB of ram and support for hardware transformation and lighting
ไครฟ์ไดรฟ์	ความเร็ว x 32
เสียง	การ์ดเสียง ลำโพง หรือชุดหูฟัง
อุปกรณ์ชี้ตำแหน่ง	เป็นพิมพ์และเมาส์หรือตัวควบคุมเกมที่เข้ากันได้ (เช่น Xbox 360 คอนโทรลเลอร์สำหรับ Windows)
การเชื่อมต่ออินเทอร์เน็ตสำหรับการเล่นออนไลน์	อะแดปเตอร์เครือข่ายสำหรับ gameplay เกมส์แบบมัลติ LAN
	อินเทอร์เน็ตแบบบรอดแบนด์สำหรับ Windows Live gameplay เกมส์แบบมัลติ

### 3.2.2 X-Plane

X-Plane เป็นโปรแกรมจำลองการบินที่มีประสิทธิภาพ และครอบคลุมสำหรับคอมพิวเตอร์ส่วนบุคคล ซึ่งสามารถจำลองการบินได้อย่างมีประสิทธิภาพเหมือนจริง สำหรับการใช้ในคอมพิวเตอร์ครัวเรือน X-Plane ไม่ใช่เกมส์แต่คือนวัตกรรมทางวิศวกรรมที่ใช้ในการทำนายหรือจำลองคุณภาพการบินของอากาศยานปีกยัด ปีกหมุนอย่างแม่นยำ

จากคุณสมบัติที่เพียบพร้อมของโปรแกรม X-Plane นั้น เป็นเครื่องมือที่จะช่วยให้นักบินได้ลองฝึกทำการบิน และแก้ไขสถานการณ์ในการบิน เนื่องจากมีความสมจริงค่อนข้างสูงสำหรับวิศวกรสามารถนำมาทำนายประสิทธิภาพการบินของอากาศยานที่ได้ทำการออกแบบได้ ซึ่งสามารถทำนายได้ในด้านกลศาสตร์การบิน

ในโปรแกรมประกอบไปด้วยอากาศยานที่ใช้ทำการบินมากมาย อาทิเช่น อากาศยานเครื่องยนต์ลูกสูบ อากาศยานเครื่องยนต์ไอพ่น เครื่องร่อน เฮลิคอปเตอร์ และอากาศยานขึ้นลงแนวตั้ง (VTOLs) นอกจากนี้โปรแกรมยังสามารถรองรับการบินทั้งการบินความเร็วต่ำกว่าความเร็วเสียงในอากาศ (Subsonic) และความเร็วบินมากกว่าความเร็วเสียงในอากาศ (Supersonic) และผู้ใช้ยังสามารถเข้าถึงตัวแปรคุณสมบัติในการบินต่าง ๆ ได้อีกด้วย โดยทั้งนี้ X-Plane ยังรวบรวม



อากาศยานไว้ให้ใช้ได้ทั้งหมด 35 รุ่นไว้ให้ผู้ใช้ได้ทำการบินในอากาศยานลักษณะต่าง ๆ เช่น Cessna 172, Bell 206 และ B-2 Bomber เป็นต้น

ความต้องการขั้นต่ำของโปรแกรม X-Planeเนื่องด้วยความสามารถที่มากมายของ X-Plane นั้นจึงไม่สามารถจะทำงานบนคอมพิวเตอร์รุ่นเก่าได้ โดยทั่วไปแล้วอาจประมาณได้ว่าคอมพิวเตอร์ที่สามารถติดตั้งโปรแกรมได้นั้นจะต้องเป็นคอมพิวเตอร์ที่มีคุณสมบัติตามตารางที่ 3.2

ตารางที่ 3.2 ความต้องการขั้นต่ำของโปรแกรม X-Plane

ระบบปฏิบัติการ	Windows, Linux และ iOS
ตัวประมวลผลของคอมพิวเตอร์	2 GHz
หน่วยความจำ	RAM อย่างน้อย 1.0 กิกะไบต์ (GB)
เนื้อที่ว่างบนฮาร์ดดิสก์	10 กิกะไบต์ (GB)
การ์ดแสดงผล	500 MB VRAM
เสียง	การ์ดเสียง ลำโพง หรือชุดหูฟัง
อุปกรณ์ชี้ตำแหน่ง	เป็นพิมพ์ และเมาส์ หรือตัวควบคุมเกมที่เข้ากันได้ (เช่น Xbox 360 คอนโทรลเลอร์สำหรับ Windows)

### 3.2.3 การศึกษาข้อดีและข้อเสียของ Microsoft Flight Simulator X (FSX) และ X-Plane

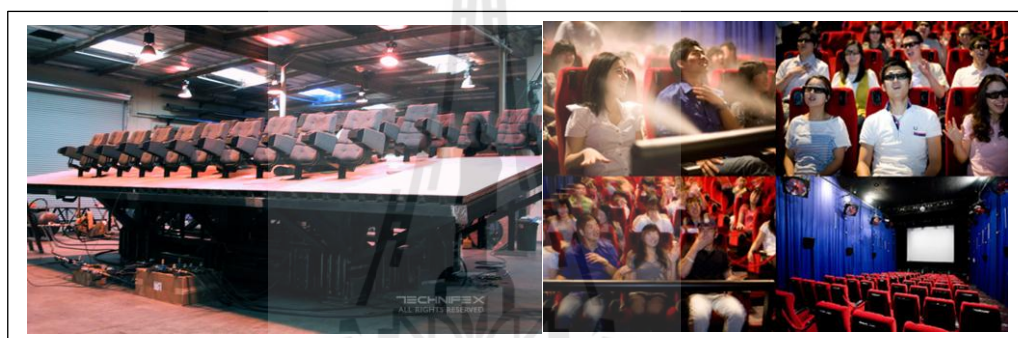
- การต่อออกหลายหน้าจอจากโปรแกรมทั้งสอง
- การเชื่อมต่อข้อมูล เพื่อใช้ควบคุมการเคลื่อนที่ของ Platform

## 3.3 การออกแบบแผ่นเคลื่อนไหว (Motion Platform)

เมื่อก้าวถึงแผ่นเคลื่อนไหว ได้มีการประยุกต์ใช้งานอย่างเหมาะสมจนเป็นที่แพร่หลายกันมาก เช่น เครื่องจำลองการขับขี่รถยนต์ (Driving Simulator), เครื่องฝึกหัดบิน (Flight Simulator) หรือ เครื่องเล่นในสวนสนุกต่าง ๆ ตามตัวอย่างในรูปที่ 3.2 และ รูปที่ 3.3 เป็นต้น โดยมีวัตถุประสงค์หลัก คือ การสร้างสภาวะให้ผู้ขับขี่ได้รู้สึกเหมือนหรือใกล้เคียงกับการขับขี่ยานพาหนะจริงให้มากที่สุด โดยเฉพาะเครื่องจำลองการบิน ที่ได้นำมาฝึกทดสอบการบินเพื่อป้องกันอันตรายที่จะเกิดจากการบินจริง และยังสามารถประหยัดค่าใช้จ่ายในการบินแต่ละเที่ยวบินอีกด้วย



รูปที่ 3.2 แสดงเครื่องฝึกบินจำลอง



รูปที่ 3.3 แสดงเครื่องเล่นในสวนสนุก

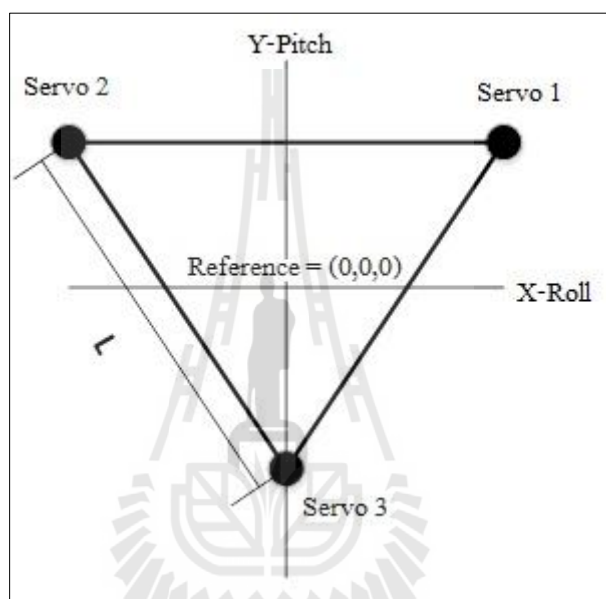
ลักษณะการทำงานของเครื่องจำลองเคลื่อนที่นั้นถูกแบ่งออกหลายระดับด้วยกัน เครื่องจำลองระดับที่ง่ายที่สุด เห็นจะเป็นกลุ่มเครื่องเล่นในสวนสนุก ซึ่งเครื่องจำลองนี้จะดำเนินการจำลองสถานการณ์ตามที่ได้ถูกกำหนดล่วงหน้าแล้ว โดยผู้ขับขี่เป็นเสมือนผู้โดยสารเท่านั้น ไม่สามารถมีปฏิสัมพันธ์กับเครื่องจำลองได้เลย ในส่วนของความสมจริงนั้นก็จะมีทั้งระบบภาพและเสียงเข้ามาช่วยสร้างสภาวะให้ในส่วนนี้ สำหรับเครื่องจำลองที่มีความสมจริงมากขึ้นก็จะเป็นเครื่องจำลองที่มีการปฏิสัมพันธ์กับผู้ขับขี่ และสามารถจำลองได้ทั้งภาพ, เสียงและการเคลื่อนไหว ซึ่งจะทำให้ผู้ขับขี่สามารถรับรู้การเคลื่อนไหวที่สมจริงมากยิ่งขึ้น โดยเฉพาะการรับรู้ถึงแรงอันเกิดจากความเร่งในการระหว่งการเคลื่อนที่ของยานพาหนะ ที่มากระทำกับผู้ขับขี่ให้รู้สึกเหมือนกับว่าได้เคลื่อนที่อยู่จริง ๆ ซึ่งในงานวิจัยนี้ได้ทำการสร้างแผ่นเคลื่อนไหวจำลอง ที่สามารถสร้างสภาวะการเคลื่อนไหวได้ 3 องศาอิสระ โดยจะอยู่ในรูปของหุ่นยนต์โครงสร้างขนาน (Parallel Robot)

การออกแบบหุ่นยนต์หรือแขนกลโครงสร้างขนานในงานวิจัยนี้สามารถเคลื่อนไหวได้ 3 องศาอิสระ คือ ในทิศหมุนรอบแกน X-Y (Roll-Pitch) และทิศทางตามแกน X โดยพิจารณา

เลือกใช้เซอร์โวมอเตอร์ (Servo Motor) เป็นระบบขับเคลื่อนแกนกล (Actuator) ให้เคลื่อนไหวตามต้องการ ซึ่งจะมีขั้นตอนการคำนวณหาพิกัดการเคลื่อนที่ของแผ่นเคลื่อนไหว (Platform) ดังต่อไปนี้

### 3.3.1 ระบบแกนอ้างอิง (Reference system)

การเปลี่ยนระบบแกนนั้นทำได้ โดยการนำเอาแกนพิกัดจากมากำหนดพิกัดทั้งตำแหน่งในแนวระดับและความสูงในแนวแกนตั้ง โดยกำหนดรูปแบบและพิกัดของแท่นเคลื่อนที่ดังรูปที่ 3.4



รูปที่ 3.4 แสดงระบบแกนอ้างอิง ( $L$  = ระยะห่างของ Servo)

ตำแหน่งเริ่มต้นของจุดเชื่อมต่อแผ่นเคลื่อนไหวทั้งสามกำหนดโดยระบบพิกัดข้างต้นดังสมการดังนี้

$$P_1 = \left( \frac{L}{2}, \frac{L}{2\sqrt{3}}, 0 \right) \quad (3.1)$$

$$P_2 = \left( -\frac{L}{2}, \frac{L}{2\sqrt{3}}, 0 \right) \quad (3.2)$$

$$P_3 = \left( 0, -\frac{L}{\sqrt{3}}, 0 \right) \quad (3.3)$$

โดยตำแหน่งทั้งสามนั้นใช้กำหนดเป็นเมตริกซ์ตำแหน่งตามแนวแกน โดยตำแหน่งการติดตั้งมอเตอร์หรือชุดส่งกำลังนั้นอาจมิได้ติดตั้งอยู่ในแนวที่ตรงกัน อาจมีระยะห่างกันเป็นระยะ  $d$  จากศูนย์กลาง ซึ่งในระบบแกนอ้างอิงนั้นหมายถึงระยะทางในแนวแกน  $y$  เมตริกซ์ผลลัพธ์จากตำแหน่งเริ่มต้นทั้งหมดสามารถเขียนอยู่ในรูปของเมตริกซ์ได้ดังสมการที่ 3.4

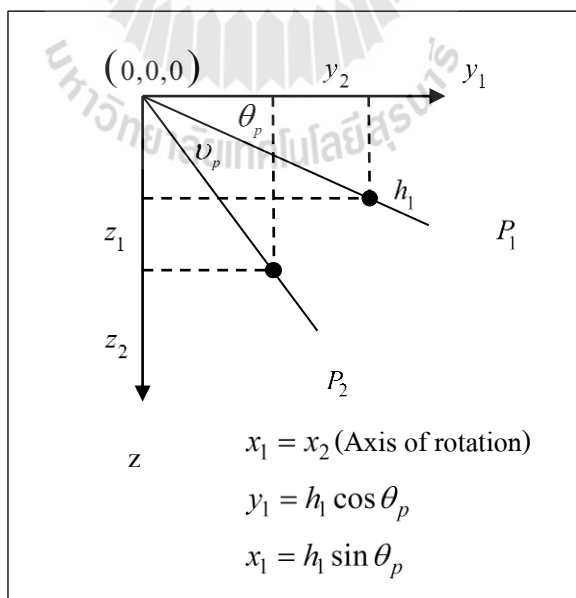
$$P_{123} = \begin{bmatrix} \frac{L}{2} & -\frac{L}{2} & 0 \\ \frac{L}{2\sqrt{3}} + d & \frac{L}{2\sqrt{3}} + d & -\frac{L}{\sqrt{3}} + d \\ 0 & 0 & 0 \end{bmatrix} \quad (3.4)$$

แผ่นเคลื่อนที่จะทำการหมุนรอบแกนทั้งหมดสองแกนในการเคลื่อนที่หนึ่งครั้ง ซึ่งการเคลื่อนที่นั้นสามารถถูกเขียนให้อยู่ในรูปของ Linear Transformations โดยใช้เมตริกซ์เป็นตัวดำเนินการได้

### 3.3.2 Matrix Transformation

- การเคลื่อนที่ปีกเงย (Pitch Motion)

การเคลื่อนที่ปีกเงยนั้นเป็นการหมุนรอบแกน  $x$  ดังในรูปที่ 3.5 แกน  $x$  พุ่งตั้งฉากเข้าไปในกระดาษซึ่งแสดงด้วยจุด การหมุนจากจุด  $P_1$  ที่มุม  $\theta_p$  ไปยังจุด  $P_2$  ด้วยมุมที่หมุนไปเป็นมุม  $\nu_p$



รูปที่ 3.5 แสดงการหมุนรอบแกน X

การหมุนนั้นสามารถเขียนเป็นสมการได้ดังนี้

$$y_2 = h_1 \cos(\theta_p + \nu_p) = h_1 \cos(\theta_p) \cos(\nu_p) - h_1 \sin(\theta_p) \sin(\nu_p) \quad (3.5)$$

$$z_2 = h_1 \sin(\theta_p + \nu_p) = h_1 \sin(\theta_p) \cos(\nu_p) + h_1 \cos(\theta_p) \sin(\nu_p) \quad (3.6)$$

พิจารณความสัมพันธ์ระหว่างตำแหน่งเดิมและตำแหน่งใหม่จะได้

$$x_2 = x_1 \quad (3.7)$$

$$y_2 = y_1 \cos(\nu_p) - z_1 \sin(\nu_p) \quad (3.8)$$

$$z_2 = y_1 \sin(\nu_p) + z_1 \cos(\nu_p) \quad (3.9)$$

จะได้ transformation matrix ดังนี้

$$T_p = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\nu_p) & -\sin(\nu_p) \\ 0 & \sin(\nu_p) & \cos(\nu_p) \end{bmatrix} \quad (3.10)$$

คูณเมทริกซ์นี้ด้วยเมทริกซ์ตำแหน่งเริ่มต้น จะได้ตำแหน่งใหม่ของตำแหน่งเริ่มต้นทั้งสาม

$$P_p = T_p P_{123} = [A][B] \quad (3.11)$$

เมื่อ

$$[A] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\nu_p) & -\sin(\nu_p) \\ 0 & \sin(\nu_p) & \cos(\nu_p) \end{bmatrix}$$

$$[B] = \begin{bmatrix} \frac{L}{2} & -\frac{L}{2} & 0 \\ \frac{L}{2\sqrt{3}} + d & \frac{L}{2\sqrt{3}} + d & -\frac{L}{\sqrt{3}} + d \\ 0 & 0 & 0 \end{bmatrix}$$

จะได้

$$P_p = \begin{bmatrix} \frac{L}{2} & -\frac{L}{2} & 0 \\ \left(\frac{\sqrt{3}L}{6} + d\right)\cos(\nu_p) & \left(\frac{\sqrt{3}L}{6} + d\right)\cos(\nu_p) & \left(-\frac{\sqrt{3}L}{3} + d\right)\cos(\nu_p) \\ \left(\frac{\sqrt{3}L}{6} + d\right)\sin(\nu_p) & \left(\frac{\sqrt{3}L}{6} + d\right)\sin(\nu_p) & \left(-\frac{\sqrt{3}L}{3} + d\right)\sin(\nu_p) \end{bmatrix} \quad (3.12)$$

- การเคลื่อนที่รอบแกน Y (Roll motion)

การเคลื่อนที่แบบ Roll นั้นเป็นการหมุนรอบแกน y การเคลื่อนที่เริ่มต้นจากสมการที่ 3.12 โดยการคำนวณนั้นเป็นไปในทางเดียวกันกับการเคลื่อนที่ปีกเงย ผลลัพธ์ของ Transformation Matrix คือ

$$T_r = \begin{bmatrix} \cos(\nu_r) & 0 & -\sin(\nu_r) \\ 0 & 1 & 0 \\ \sin(\nu_r) & 0 & \cos(\nu_r) \end{bmatrix} \quad (3.13)$$

ตำแหน่งใหม่ของจุดเชื่อมต่อหาได้จากการนำเอาเมทริกที่ได้จากการเคลื่อนที่ปีกเงยแล้วจะได้

$$P_{pr} = \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ z_1 & z_2 & z_3 \end{bmatrix} \quad (3.14)$$

$$\text{เมื่อ } x_1 = -\left(\frac{\sqrt{3}L}{6} + d\right) \sin(\nu_p) \sin(\nu_r) + \frac{L}{2} \cos(\nu_r)$$

$$x_2 = -\left(\frac{\sqrt{3}L}{6} + d\right) \sin(\nu_p) \sin(\nu_r) - \frac{L}{2} \cos(\nu_r)$$

$$x_3 = -\left(-\frac{\sqrt{3}L}{6} + d\right) \sin(\nu_p) \sin(\nu_r)$$

$$y_1 = y_2 = \left(\frac{\sqrt{3}L}{6} + d\right) \cos(\nu_p)$$

$$y_3 = \left(-\frac{\sqrt{3}L}{6} + d\right) \cos(\nu_p)$$

$$z_1 = \left(\frac{\sqrt{3}L}{6} + d\right) \sin(\nu_p) \sin(\nu_r) + \frac{L}{2} \sin(\nu_r)$$

$$z_2 = \left(\frac{\sqrt{3}L}{6} + d\right) \sin(\nu_p) \cos(\nu_r) - \frac{L}{2} \sin(\nu_r)$$

$$z_3 = \left(-\frac{\sqrt{3}L}{6} + d\right) \sin(\nu_p) \cos(\nu_r)$$

- การเคลื่อนที่ตามแกน Z (Heave motion)

การเคลื่อนที่แบบสุดท้ายนั้นคือการเคลื่อนที่บนแกน z โดยการเคลื่อนที่นั้นสามารถเพิ่มค่าการเคลื่อนที่ได้โดยตรงที่แกน z ของทุกจุด โดยจะได้ Transformation Matrix สุดท้ายดังนี้

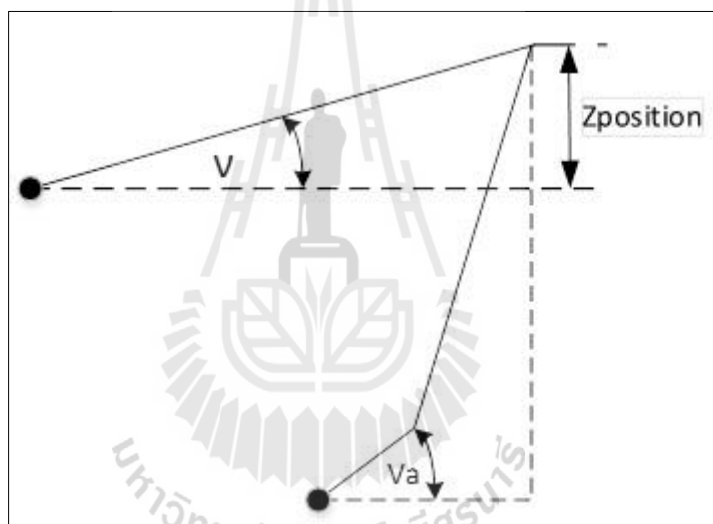
$$P_{prz} = \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ z_1 & z_2 & z_3 \end{bmatrix} \quad (3.15)$$

$$\text{เมื่อ } x_1 = -\left(\frac{\sqrt{3}L}{6} + d\right) \sin(\nu_p) \sin(\nu_r) + \frac{L}{2} \cos(\nu_r)$$

$$x_2 = -\left(\frac{\sqrt{3}L}{6} + d\right) \sin(\nu_p) \sin(\nu_r) - \frac{L}{2} \cos(\nu_r)$$

### 3.3.3 การเติมข้อมูลเพื่อส่งสัญญาณไปยังอุปกรณ์ต้นกำลัง (Actuator signals)

จากเมทริก  $P_{prz}$  ซึ่งให้พิกัดของจุดเชื่อมต่อของแผ่นเคลื่อนไหวยกกับจุดยึดมอเตอร์ทั้งสาม โดยตำแหน่งที่ได้นั้นเทียบกับระบบแกนอ้างอิงในหัวข้อข้างต้น แกนเชื่อมต่อระหว่างจุดเชื่อมต่อกับเพลลาของเซอร์โวมอเตอร์มีรูปร่างตามรูปที่ 3.6



รูปที่ 3.6 แสดงความสัมพันธ์ระหว่างมุมของเพลลา  $\nu_a$  และ มุมของแท่นเคลื่อนไหว  $\nu$

เพื่อที่จะหามุมการหมุนของเพลลาเซอร์โวมอเตอร์เพื่อให้ได้การทำมุมของแท่นเคลื่อนไหวที่เป็นไปตามมุมที่ต้องการหมุนไป ในการคำนวณหาต้องการค่า  $z_{position}$  ของทุก ๆ จุดเชื่อมต่อของแผ่นเคลื่อนไหว ซึ่งหาได้จากแถวสุดท้ายของเมทริก  $P_{prz}$  ในสมการที่ 3.15 ซึ่งจะทำให้สามารถหามุมการหมุนของเซอร์โวมอเตอร์แต่ละตัวได้จากความสัมพันธ์ในสมการดังนี้

$$V_a = \sin^{-1}\left(\frac{z_{position}}{r}\right) \quad (3.16)$$



จากนั้นค่ามุมที่ได้จากการคำนวณในสมการข้างต้นก็จะสามารถนำมาแปลเป็นสัญญาณ Pulse Width Modulation (PWM) เพื่อใช้ในการสั่งงานเซอร์โวมอเตอร์แต่ละตัวให้ไปในตำแหน่งที่ต้องการอยู่ในช่วง  $\pm 30^\circ$  ซึ่งจะกล่าวในลำดับต่อไป

### 3.3.4 อุปกรณ์ต้นกำลัง (Actuator)

ในงานวิจัยนี้เลือกใช้ Servo Motor เป็นอุปกรณ์ต้นกำลังในการเคลื่อนที่ ซึ่ง Servo Motor คือ Motor ชนิดหนึ่งซึ่งถูกออกแบบมาสำหรับใช้ในระบบการควบคุมทางความเร็ว และตำแหน่ง Servo Motor ใช้การควบคุมแบบ Close Loop Control โดยที่ส่วนประกอบหลักของ Servo Motor จะประกอบด้วยส่วนสำคัญ 2 ส่วนคือ ชุด Motor ขับเคลื่อน ใช้งานร่วมกับชุดเกียร์เพื่อเพิ่มแรงบิด และระบบ Feedback Encoder สำหรับวัดตำแหน่งมุมองศาการหมุน การควบคุม Servo Motor จะใช้สัญญาณ Pulse โดยเลือกใช้ Servo Motor ยี่ห้อ Mitsubishi และ Panasonic ดังแสดงในรูปที่ 3.7



รูปที่ 3.7 แสดง Servo Motor Mitsubishi รุ่น MR-J2

โดยในงานวิจัยมีการกำหนดค่าในการใช้งานดังต่อไปนี้คือ

- ความถี่ที่ใช้ในการควบคุม : ตามความเร็วการตอบสนองของมอเตอร์
- สัญญาณพัลส์ทิศทางการหมุน (Direction Sign) : 5 Volt
- สัญญาณพัลส์จำนวนขั้นการหมุน (Pulse Speed) : 5 Volt

ในระยะเริ่มต้นผู้วิจัยได้ทำการทดลอง ส่งสัญญาณการเปลี่ยนแปลงของมุมมองไปยัง Servo Motor โดยตรง เพื่อทดลองว่าเซอร์โวมอเตอร์สามารถหมุนไปยังตำแหน่งที่เราต้องการได้ตรงตามมุมมองหรือไม่

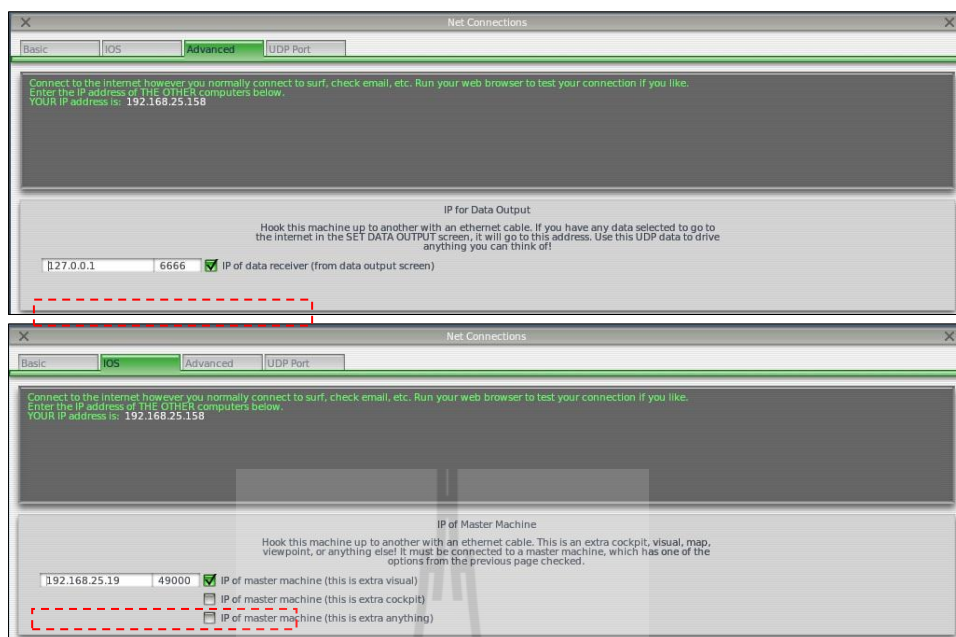
### 3.4 อัลกอริทึมสำหรับประมวลผลเพื่อเชื่อมต่อข้อมูลระหว่างโปรแกรมกับแผ่นเคลื่อนไหว

งานวิจัยนี้ต้องการสร้างต้นแบบแผ่นเคลื่อนไหวที่สามารถเคลื่อนไหวได้ตามการเปลี่ยนแปลงที่เกิดขึ้น ขณะผู้ใช้งานกำลังทำการบิน โดยใช้โปรแกรมจำลองการบิน ซึ่งในปัจจุบันระบบจำลองการบินมีการใช้งานอย่างหลากหลายรูปแบบ จากการศึกษาข้อมูลผู้วิจัยจึงเลือกออกแบบแผ่นเคลื่อนไหวที่สามารถเคลื่อนที่ได้ 3 องศาอิสระ โดยใช้ซอฟต์แวร์ ArduinoIDE ร่วมกับ Microcontroller ยี่ห้อ Arduino รุ่น UNO เป็นส่วนต่อประสาน (Interface) ให้แผ่นเคลื่อนไหวทำงานไปพร้อมกับโปรแกรมจำลองการบิน (Flight Simulator) ตามรายละเอียดต่อไปนี้

#### 3.4.1 การส่งถ่ายข้อมูลจากโปรแกรมจำลองการบิน

การส่งถ่ายข้อมูลจากโปรแกรมจำลองการบินขณะที่กำลังทำการบินเป็นสิ่งจำเป็นอย่างยิ่งในงานวิจัยนี้ เนื่องจากจะต้องอาศัยข้อมูลการบินขณะนั้น (Real Time) ว่ามีลักษณะท่าทางการบิน โรล-พิช-ยอ (Roll-Pitch-Yaw) และค่าความเร่งที่กระทำต่อเครื่องบินขณะนั้นเป็นเช่นไร ซึ่งการที่จะให้อุปกรณ์ทำการสื่อสารและส่งข้อมูลในขณะนั้นเดียวกันนั้น ผู้วิจัยจะทำการจำลองระบบขึ้นมา โดยอาศัยการสื่อสารผ่าน IP Address เดียวกัน โดยมีรายละเอียดดังต่อไปนี้

1. ตั้งค่าการส่งข้อมูลจากโปรแกรมจำลองการบิน โดยตั้งค่าการส่ง IP Address แล้วทำการเลือกตัวแปรที่ต้องการส่งข้อมูล ซึ่งในงานวิจัยเลือกค่าที่บ่งบอกถึงท่าทางการบินในทิศทางโรล-พิช-ยอ (Roll-Pitch-Yaw) และค่าความเร่งที่กระทำต่อเครื่องบิน ดังรูปที่ 3.8



รูปที่ 3.8 แสดงการตั้งค่าการส่งข้อมูลจากโปรแกรมจำลองการบิน

2. เขียนโปรแกรมเพื่อรับค่า ที่ส่งจากโปรแกรมจำลองการบิน ในการเขียนโปรแกรมรับค่าท่าทางการเคลื่อนที่ดังกล่าว ตามรายละเอียดดังต่อไปนี้

เริ่มจากตั้งค่าฟังก์ชันเพื่อเปิดรับสัญญาณที่จะถูกส่งมาจากโปรแกรมจำลองการบิน โดยใช้ฟังก์ชัน UDP.read (Package Buffer, Maximum Buffer) ;

ฟังก์ชัน UDP.read ในการอ่านข้อมูลที่ถูกส่งมาจากโปรแกรมจำลองการบิน ซึ่งมีรายละเอียดดังนี้

- Maximum Buffer : กำหนดค่าไบต์ (Bytes) สูงสุดที่จะทำการอ่าน จากการศึกษาในโปรแกรมส่งออกมาเพียง 77 ไบต์

- Package Buffer : คือสายอักขระที่รับได้จากสัญญาณ UDP

จากการออกแบบลักษณะของแผ่นเคลื่อนไหวที่จะนำมาจำลองการเคลื่อนที่ของเครื่องบิน โดยใช้ Servo Motor 3 ตัว เป็นต้นกำลังในการขับเคลื่อน สามารถคำนวณการเคลื่อนที่ของ Servo Motor แต่ละตัวได้ตามสมการ 3.15 และ 3.16 เมื่อได้ค่าการเคลื่อนที่ของ Servo Motor แล้วผู้วิจัยได้ทำการเขียนโปรแกรมสั่งงาน Servo Motor โดยใช้ ฟังก์ชัน ของโปรแกรม ArduinoIDE ร่วมกับบอร์ด Microcontroller ยี่ห้อ Arduino โดยมีรายละเอียดดังนี้

เขียนโปรแกรมประมวลผลการเคลื่อนที่ของ Servo Motor โดยใช้โปรแกรม ArduinoIDE สื่อสารกับบอร์ด Arduino โดยมีรายละเอียดดังต่อไปนี้

- 1) รับสัญญาณ UDP มาแปลงเป็นสายอักขระ
- 2) แปลงสายอักขระที่ได้เป็นจำนวนจริงของค่าของมุม Roll, Pitch และค่า g
- 3) นำค่าที่ได้มาคำนวณเพื่อหามุมที่มอเตอร์แต่ละตัวต้องทำการหมุนไป
- 4) ส่งสัญญาณที่ได้ให้มอเตอร์ทำการเคลื่อนที่ และ วัดมุมด้วย Encoder ที่ติดตั้งไว้ที่ปลายของมอเตอร์
- 5) ทำซ้ำเช่นนี้จนกว่าจะหยุดการทำงาน

### 3.5 การตรวจสอบความถูกต้อง

งานวิจัยจะเชื่อถือได้มากน้อยเพียงใดย่อมขึ้นอยู่กับเครื่องมือที่ใช้ในการรวบรวมข้อมูล ถ้าเครื่องมือไม่มีความเที่ยงตรง เชื่อถือไม่ได้ งานวิจัยก็เชื่อถือไม่ได้เช่นกัน ดังนั้น ในงานวิจัยนี้ได้ทำการออกแบบ การตรวจสอบความถูกต้อง ดังต่อไปนี้

#### 3.5.1 ความถูกต้องของระบบ

ผู้วิจัยได้ทำการออกแบบการทดลอง การตรวจสอบความถูกต้องของระบบ โดยเขียนโปรแกรมส่งค่ามุมการเคลื่อนที่ให้ Servo Motor ทั้งสาม ที่ละตำแหน่งการเคลื่อนที่ แล้วใช้ IMU (Inertial Measurement Unit) ในการตรวจสอบมุมการเคลื่อนไหวของแผ่นเคลื่อนไหวโดยจะนำเสนอผลการวัดในบทที่ 4

#### 3.5.2 การปรับค่าความเหมาะสมของ PID Controller

ผู้วิจัยได้เลือกใช้วิธีการปรับจูนค่า PID ตามที่ Ziegler-Nichols เสนอไว้ในการปรับจูนค่า ในกรณีที่มีการเขียนฟังก์ชันถ่ายโอนไม่สะดวกหรือหาค่าได้ยาก โดยมีวิธีการดังนี้

- 1) เช็ต  $K_i$  และ  $K_d$  เป็นศูนย์
- 2) ปรับค่า  $K_p$  เริ่มจากค่าเล็ก ๆ
- 3) จดค่า  $K_p$  และคาบเวลาที่ทำให้เกิด Oscillation แบบแอมพลิจูดคงที่
- 4) คำนวณค่า  $K_p$ ,  $K_i$  และ  $K_d$  จากตารางการกำหนดค่าที่ปรับจูนได้ตามรูปที่ 3.9

เมื่อ  $P_{cr}$  คือคาบของการแกว่งกวัดที่มีแอมพลิจูดคงที่

Controller	$K_p$	$T_i$	$T_d$
P	$0.5 K_{cr}$	Infinity	0
PI	$0.45 K_{cr}$	$P_{cr}/1.2$	0
PID	$0.6 K_{cr}$	$P_{cr}/2$	$0.125 P_{cr}$

รูปที่ 3.9 แสดงค่าการปรับจูนสัมประสิทธิ์

### 3.5.3 การตรวจสอบผลการตอบสนองต่อสัญญาณป้อนเข้า

โดยในงานวิจัยนี้ผู้วิจัยทำการทดสอบระบบด้วยสัญญาณป้อนเข้า 2 ชนิด คือ สัญญาณป้อนเข้าแบบขั้นบันได (Step Function) และสัญญาณป้อนเข้าแบบลาดเอียง (Ramp Function) เพื่อตรวจสอบการตอบสนองว่าเป็นไปตามสัญญาณป้อนเข้าหรือไม่



## บทที่ 4

### ผลการทดลอง

#### 4.1 การศึกษาความสามารถของโปรแกรมตามหัวข้อที่ต้องการ

จากการศึกษาความสามารถของโปรแกรมจำลองการบินทั้งสองโปรแกรม สามารถสรุปผลการศึกษา ดังตารางที่ 4.1

ตารางที่ 4.1 ความสามารถของโปรแกรมจำลองการบินที่ทำการศึกษา

หัวข้อ / โปรแกรม	Microsoft Flight Simulator X	X-Plane v9
การต่อภาพหลายหน้าจอ	ขึ้นอยู่กับการ์ดแสดงผลภาพของคอมพิวเตอร์	ขึ้นอยู่กับคอมพิวเตอร์ที่นำมาต่อพ่วงในวง LAN
การส่งสถานะท่าทางการบิน	<u>ไม่สามารถ</u> ส่งต่อข้อมูลออกมาได้ โดยการเขียนโปรแกรมรับค่า ต้องอาศัย โปรแกรมเสริมในการดึงข้อมูล (Motion Platform Designer)	สามารถส่งต่อข้อมูลออกมาได้ โดยการเขียนโปรแกรมรับค่า

จะเห็นได้ว่าความสามารถของโปรแกรมทั้งสามข้อนั้น โปรแกรม X-Plane ได้ตอบโจทย์ให้กับการทำงานในงานวิจัยนี้ได้เป็นอย่างดี แม้ในข้อที่ 1 จะต้องใช้ทรัพยากรคอมพิวเตอร์มากกว่าโปรแกรม FSX นั้น แต่โปรแกรม FSX ก็ไม่สามารถตอบโจทย์ความต้องการข้อมูลท่าทางการบิน ณ ขณะนั้น เพื่อใช้ในการนำไปประมวลผลเพื่อให้ได้ตำแหน่งการเคลื่อนที่ของ Platform ได้ ซึ่งถือว่าเป็นหัวใจหลักของงานวิจัยนี้

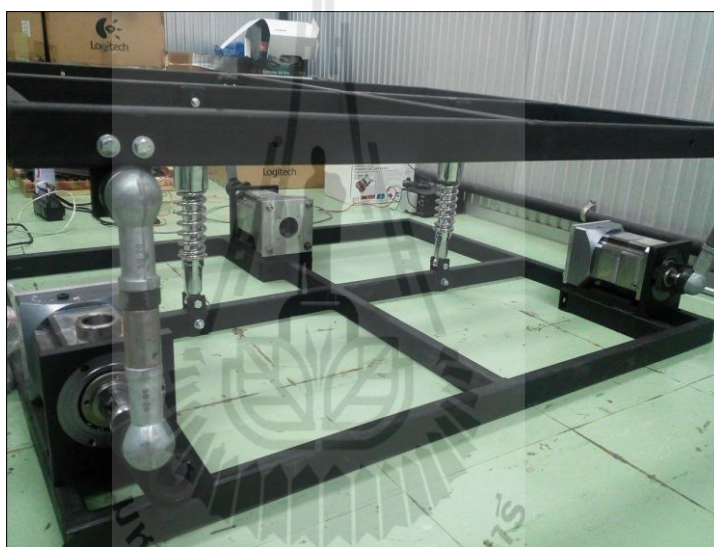
ดังนั้นจึงสรุปในหัวข้อนี้ได้ว่าโปรแกรม X-Plane ถูกเลือกใช้เป็นโปรแกรมที่จะนำมาศึกษาและใช้งานวิจัย

## 4.2 การออกแบบและขนาดของแผ่นเคลื่อนไหว

จากข้อกำหนดต่าง ๆ ของแผ่นเคลื่อนไหว (Platform) ในบทที่ 3 นั้น ซึ่งนำมาออกแบบโดยมีพารามิเตอร์ดังตารางที่ 4.2

ตารางที่ 4.2 ค่าพารามิเตอร์ที่ใช้ในการออกแบบ

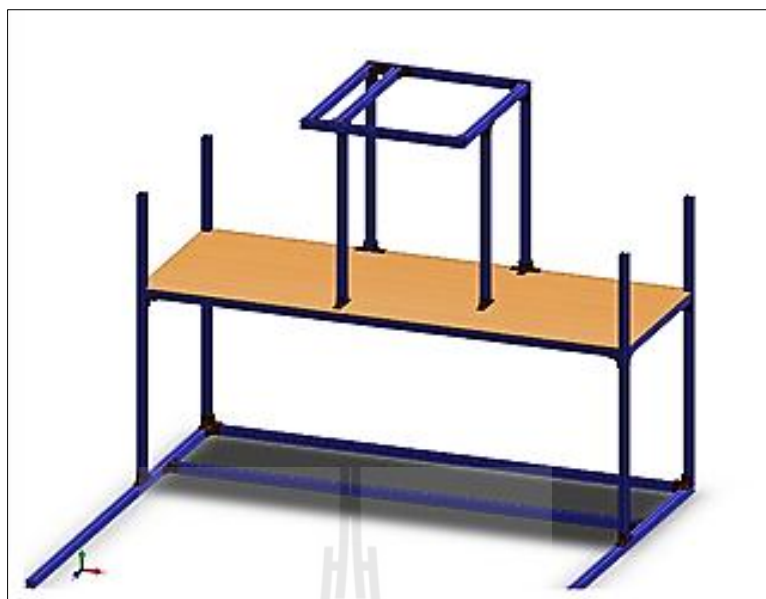
ระยะระหว่าง Servo Motor แต่ละตัว (L)	105 เซนติเมตร
ความยาวแขนของ Servo Motor (r)	5 เซนติเมตร
ระยะเอียงของแผ่นเคลื่อนไหวกับฐาน ในแนวแกน (d)	0 เซนติเมตร



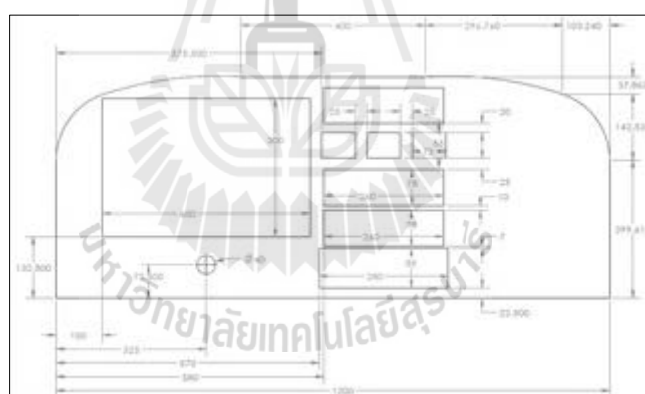
รูปที่ 4.1 แสดงแผ่นเคลื่อนไหว

### 4.2.1 การออกแบบห้องนักบิน

ขั้นตอนการออกแบบในขั้นแรกนั้นคือการสืบค้นข้อมูลขนาดห้องโดยสารของเครื่องบิน Cessna 172 พบว่ามีความกว้าง 105 เซนติเมตร และมีความสูงของแผงหน้าชุดควบคุมการบิน 47 เซนติเมตร รวมถึงการวัดขนาดของชุดควบคุมที่มีอยู่เพื่อให้ออกแบบแผงหน้าของชุดควบคุมจากนั้นได้ทำการออกแบบโดยใช้โปรแกรม SolidWorks 2011 โดยแบ่งเป็นสองส่วนคือส่วนโครงสร้าง แสดงในรูปที่ 4.2 และส่วนของแผงหน้าชุดควบคุม แสดงในรูปที่ 4.3 ซึ่งส่วนของโครงสร้างนั้นได้ใช้เป็นอลูมิเนียมโปรไฟล์ขนาด 20 x 20 มิลลิเมตร โดยเหตุผลที่เลือกใช้อลูมิเนียมโปรไฟล์ ก็เป็นวัสดุที่มีน้ำหนักเบา แข็งแรงและง่ายต่อการประกอบชิ้นส่วน



รูปที่ 4.2 แสดงส่วน โครงสร้างห้องนักบิน



รูปที่ 4.3 แสดงขนาดส่วนแผงหน้าชุดควบคุม





รูปที่ 4.4 แสดง โครงสร้างแบบสมบูรณจากการออกแบบใน โปรแกรม

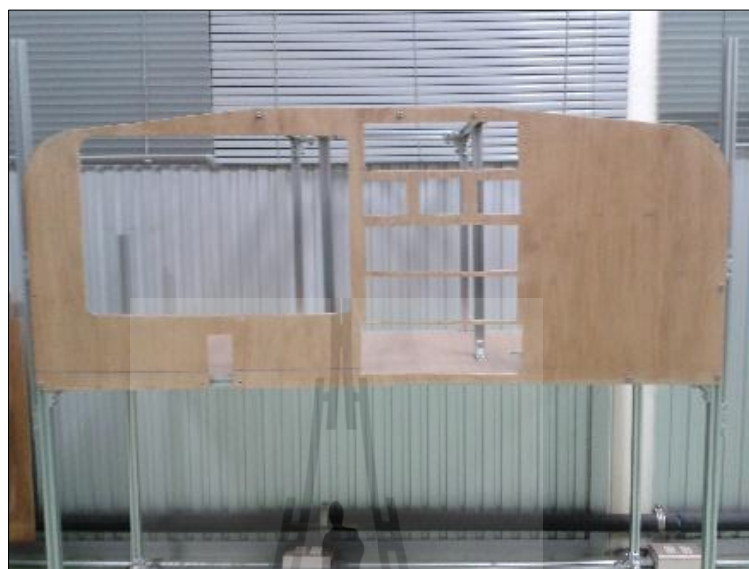
#### 4.2.2 ขั้นตอนการสร้างชิ้นงาน

ประกอบอคูมิเนียมโปรไฟล์ตามแบบในรูปที่ 4.5



รูปที่ 4.5 แสดง โครงสร้างประกอบอคูมิเนียมโปรไฟล์

ทำการตัดไม้อัดความหนา 6 มิลลิเมตรตามแบบในรูปที่ 9 ด้วยเลื่อย Jigsaw ไฟฟ้า เพื่อใช้เป็นแผงหน้าชุดควบคุม แสดงในรูปที่ 4.6



รูปที่ 4.6 แสดงแผงหน้าชุดควบคุม

ประกอบชุดควบคุมและเครื่องวัดเข้ากับแผงหน้า แสดงในรูปที่ 4.7



รูปที่ 4.7 แสดงประกอบชุดควบคุมและเครื่องวัด

### 4.3 การควบคุมเซอร์โวมอเตอร์

#### 4.3.1 หลักการควบคุมเซอร์โวมอเตอร์ไฟฟ้ากระแสสลับพร้อมชุดควบคุม

โดยในที่นี้ผู้วิจัยเลือกใช้เซอร์โวมอเตอร์มือสองที่หาได้ตามท้องตลาดมาได้เป็น เซอร์โวมอเตอร์ยี่ห้อ Mitsubishi รุ่น MR-J2 400 W จำนวน 2 ชุด และเซอร์โวมอเตอร์ยี่ห้อ Panasonic รุ่น MINAS E Series 400 W ดังแสดงในรูปที่ 4.8 และ 4.9



รูปที่ 4.8 แสดงเซอร์โวมอเตอร์ยี่ห้อ Mitsubishi รุ่น MR-J2 400 W



รูปที่ 4.9 แสดงเซอร์โวมอเตอร์ยี่ห้อ Panasonic รุ่น MINAS E Series 400 W

การควบคุมการหมุนของเซอร์โวมอเตอร์ทั้งสองชนิดนั้นเป็นไปในทางเดียวกัน โดยการกำหนดค่าพารามิเตอร์ให้ตรงกัน โดยผู้วิจัยไม่สามารถทำการกำหนดค่าพารามิเตอร์เองได้ เนื่องจากเป็นโปรแกรมของบริษัทผู้ผลิตที่หากต้องการจะต้องเพิ่มราคาของโปรแกรมเข้ามาด้วย ดังนั้นผู้วิจัยจึงใช้ทางเลือกคือให้ผู้ขายกำหนดพารามิเตอร์การควบคุมดังนี้

กำหนดใช้สัญญาณป้อนเข้าทั้งความเร็วรอบและจำนวนสัญญาณต่อรอบเป็นสัญญาณ 5 โวลต์ตามตารางที่ 4.3

ตารางที่ 4.3 ค่าพารามิเตอร์ของเซอร์โวมอเตอร์

Pulse/Revolution	2000
Direction	0 และ 5 Volt

จากตารางที่ 4.3 หากต้องการให้เซอร์โวมอเตอร์หมุนเป็นจำนวน 1 รอบจะต้องใช้สัญญาณ PWM จำนวน 2000 ครั้ง และสามารถกำหนดทิศทางการหมุนตามเข็มนาฬิกาหรือทวนเข็มนาฬิกาได้จากสัญญาณทิศทางโดยการป้อนสัญญาณ 0 โวลต์ และ 5 โวลต์ ตามลำดับ

ลักษณะของสัญญาณ PWM แสดงในรูปที่ 4.10



รูปที่ 4.10 แสดงลักษณะสัญญาณการควบคุมเซอร์โวมอเตอร์ทิศทางละ 4 Step

#### 4.3.2 การเพิ่มความสามารถในการรับภาระกรรมเซอร์โวมอเตอร์

จากการที่ผู้วิจัยเลือกใช้เซอร์โวมอเตอร์ขนาด 400 วัตต์นั้น เฉพาะเพลาส่วนปลายของเซอร์โวมอเตอร์เองนั้นไม่สามารถรับภาระกำพิกัดที่ตั้งไว้ 200 กิโลกรัมได้จึงจำเป็นต้องเลือกใช้ชุดเฟืองทด ในทิศทางเดียวกันผู้วิจัยเลือกใช้ชุดเฟืองทดมือสองที่มีจำนวน 3 ชุดเหมือนกัน เพื่อให้

การออกแบบระบบกลไกเป็นไปได้โดยไม่แตกต่างกัน ชุดเฟืองทดที่เลือกใช้นี้เป็นยี่ห้อ Parker ขนาดอัตราทดรอบ 70:1



รูปที่ 4.11 แสดงชุดเฟืองทดอัตราทด 70 ต่อ 1

#### 4.4 การปรับจูนค่าระบบควบคุม PID

จากสมการการควบคุมระบบ PID การหาค่าตัวคูณ  $K_p$ ,  $K_i$  และ  $K_d$  ให้มีความเหมาะสม การดำเนินการหาค่า  $K_p$  ที่ทำให้เกิดการแกว่งกวัดได้ คือ  $K_{p_{cr}} = 90$

โดยสามารถจับความถี่ของการแกว่งกวัดได้ 1.8 Hz แปลงเป็นคาบการเคลื่อนที่ได้ 0.556 วินาที

นำเข้าตารางการหาสัมประสิทธิ์ตัวคูณจะได้

$$K_p = 0.6 \cdot 90 = 54$$

$$K_i = 0.5 \cdot P_{cr} = 0.5 \cdot 0.556 = 0.278$$

และ  $K_d = 0.125 \cdot 0.556 = 0.0695$

```

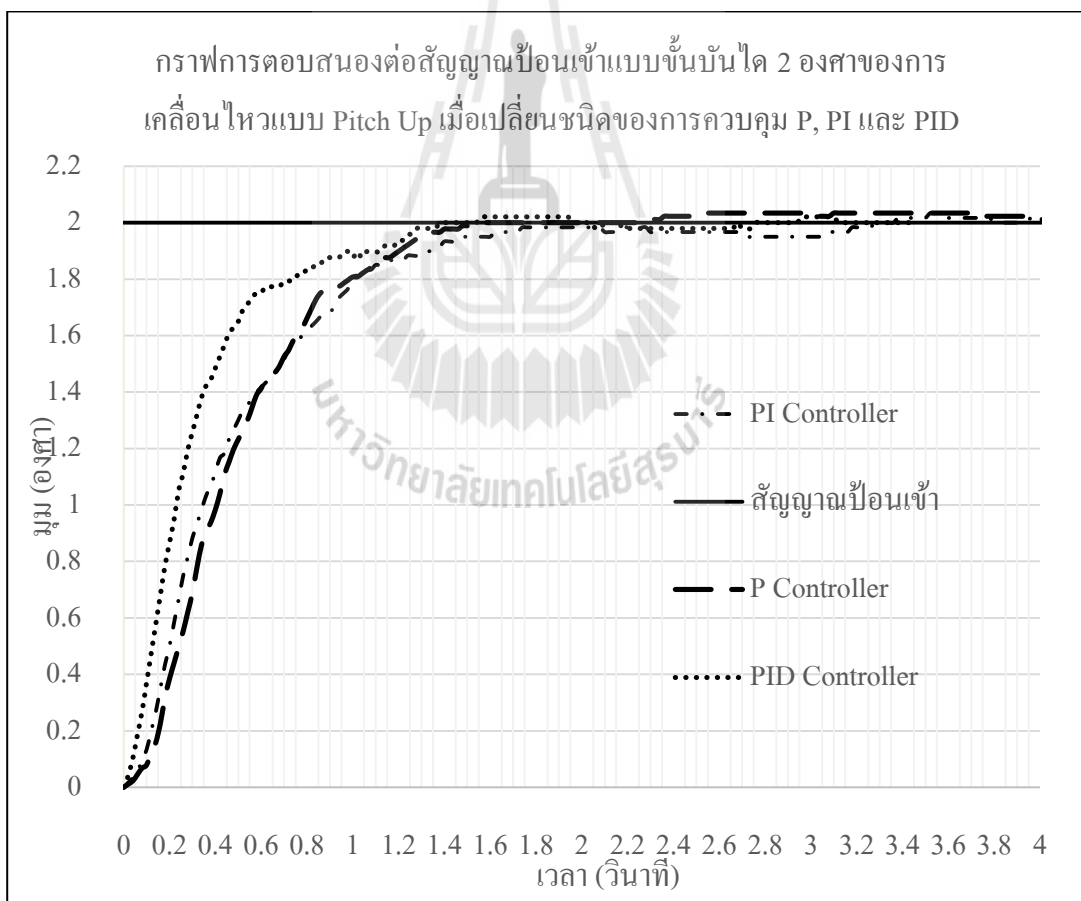
#include <SPI.h>           // needed for Arduino versions later than
#include <Ethernet.h>
#include <EthernetUdp.h>   // UDP library from: bjoern@cs.st
#include <Math.h>

int encoderPin1 = 2;
int encoderPin2 = 3;
int SW1=4,SW2=5;

float kP= 0.6*90.0;
float kI= 0.5*0.556;
float kD= 0.125*0.556;
int PID= 0,P=0,I=0,D=0;

```

รูปที่ 4.12 แสดงหน้าต่างโปรแกรมการปรับจูนค่า PID ในโปรแกรม Arduino



รูปที่ 4.13 แสดงกราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบขั้นบันได 2 องศาของการเคลื่อนไหวกแบบ Pitch Up เมื่อเปลี่ยนชนิดของการควบคุม P, PI และ PID



จากกราฟการตอบสนองต่อสัญญาณป้อนเข้า โดยเปลี่ยนแปลงตัวควบคุมเป็น P, PI และ PID พบว่าตัวควบคุมแบบ PID ลู่เข้าสู่สัญญาณป้อนเข้าได้เร็วที่สุด และมีค่าความผิดพลาดน้อยกว่าตัวควบคุมแบบ P และ PI ตัวควบคุมแบบ P พบว่ามีค่าคลาดเคลื่อนที่สถานะคงตัวอยู่ ดังนั้นจึงไม่เหมาะสมในการเลือกนำมาใช้ ส่วนตัวควบคุมแบบ PI จะเข้าสู่สถานะคงตัวค่อนข้างช้า ดังนั้นผู้วิจัยจึงเลือกใช้ตัวควบคุมแบบ PID ที่ปรับจูนมาได้

นำค่าสัมประสิทธิ์ตัวควบคุมที่ได้มาใส่ให้กับระบบควบคุม เพื่อตรวจสอบการตอบสนองต่อสัญญาณป้อนเข้าขั้นบันไดและสัญญาณป้อนเข้าแบบลาดเอียง

#### 4.5 การทดลองความถูกต้องของระบบ

ผู้วิจัยจึงได้ทำการทดลองตรวจสอบการตอบสนองของระบบดังกล่าวที่กล่าวไว้ในบทที่ 3 เพื่อเป็นข้อพิสูจน์ความน่าเชื่อถือของระบบที่ได้ทำการวิจัยว่ามีความถูกต้องน่าเชื่อถือมากน้อยเพียงใด

##### 4.5.1 เครื่องฝึกบินจำลอง



รูปที่ 4.14 แสดงเครื่องฝึกบินจำลองที่เสร็จสมบูรณ์

#### 4.5.2 ข้อจำกัดด้านการเคลื่อนที่ของแผ่นเคลื่อนไหว

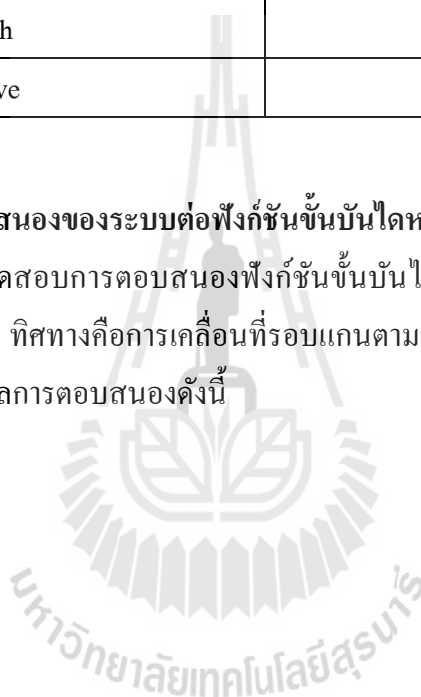
เนื่องด้วยความสามารถของอุปกรณ์ข้อต่อ (Joint) ทำให้การเคลื่อนที่ไม่สามารถทำได้ในลักษณะหมุนรอบทิศได้ ดังนั้นผู้วิจัยจึงนำเสนอข้อจำกัดการเคลื่อนที่ของแผ่นเคลื่อนไหวออกเป็น 3 ลักษณะการเคลื่อนไหว คือ มุม Roll, Pitch และการเคลื่อนที่ตามแกน Z (Heaving)

ตารางที่ 4.4 ข้อจำกัดของแผ่นเคลื่อนไหว

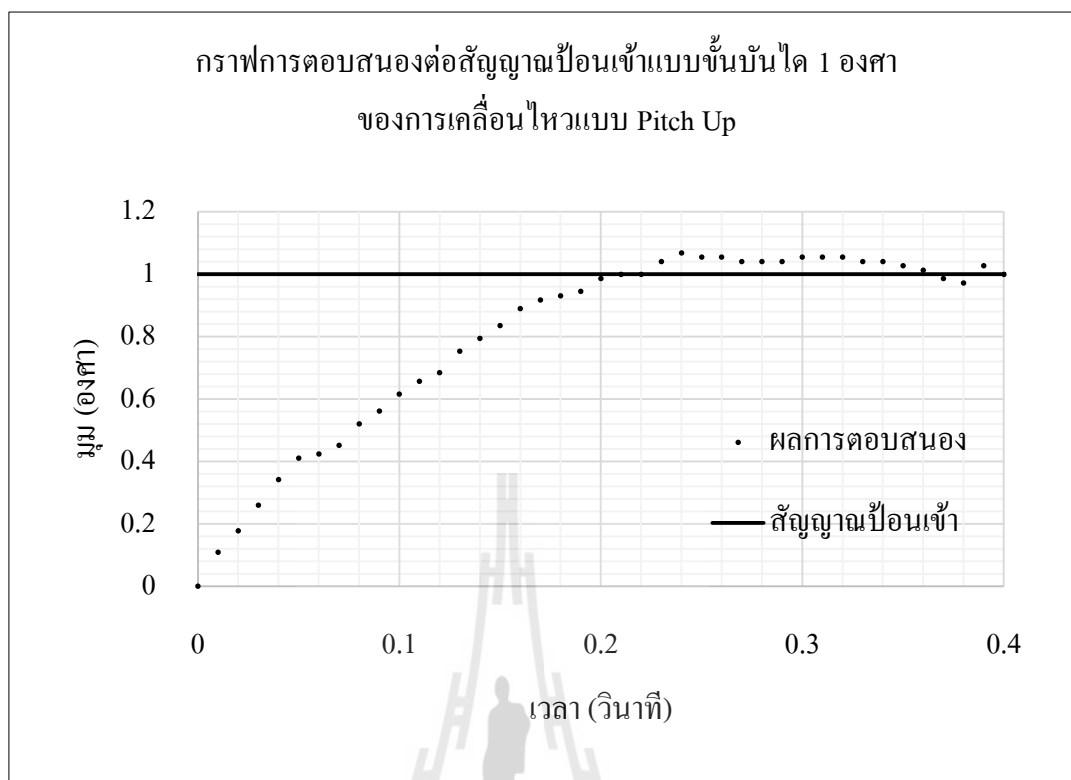
ลักษณะการเคลื่อนไหว	ข้อจำกัดการเคลื่อนไหว
Roll	-3 ถึง 3 องศา
Pitch	-5 ถึง 5 องศา
Heave	-25 ถึง 25 มิลลิเมตร

#### 4.5.3 การตอบสนองของระบบต่อฟังก์ชันขั้นบันไดหนึ่งหน่วย (Unit Step Input)

ในการทดสอบการตอบสนองฟังก์ชันขั้นบันได ผู้วิจัยได้ทดสอบการตอบสนองลักษณะการเคลื่อนที่ใน 2 ทิศทางคือการเคลื่อนที่รอบแกนตามขวาง (Pitch) และการเคลื่อนที่รอบแกนตามยาว (Roll) ซึ่งมีผลการตอบสนองดังนี้



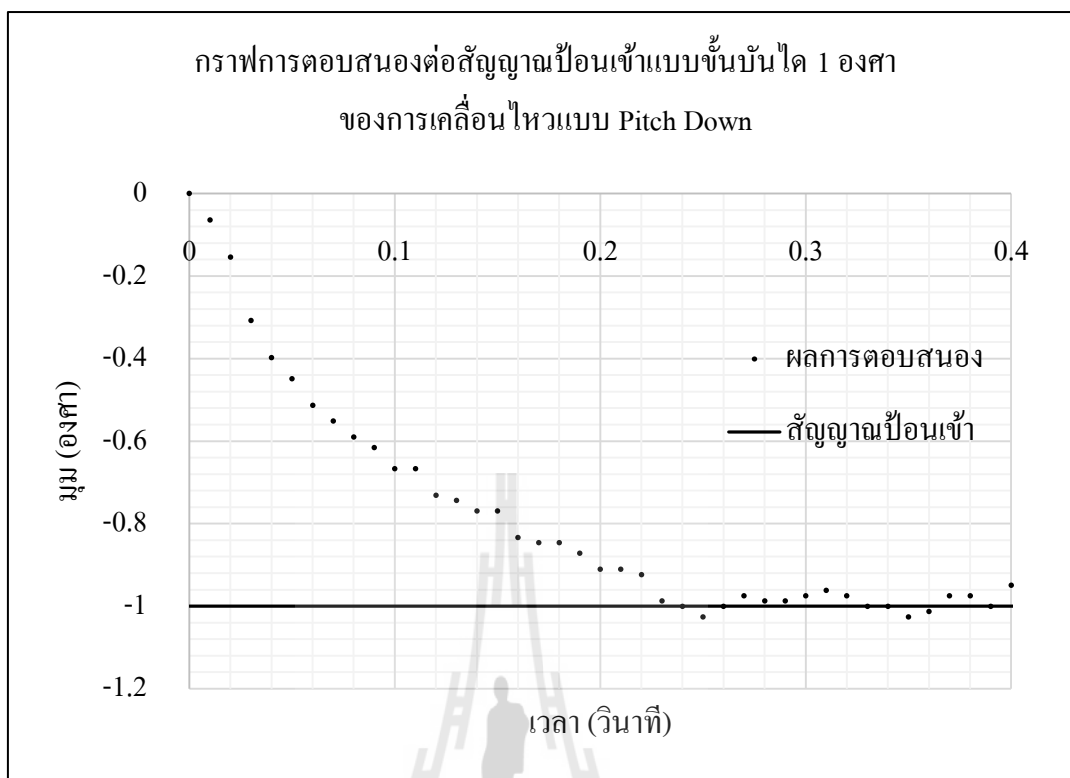




รูปที่ 4.15 แสดงกราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบขั้นบันได 1 องศา  
ของการเคลื่อนไหวยุทธแบบ Pitch Up

จากกราฟการตอบสนองจากฟังก์ชันแบบขั้นบันไดในการเคลื่อนที่รอบแกนตามตั้ง  
ในทิศทางบวกโดยให้แผ่นเคลื่อนไหว (Platform) เคลื่อนที่จากมุม 0 องศาถึง 1 องศา ผลการ  
ตอบสนองดังแสดงในรูปที่ 4.15 พบว่าแผ่นเคลื่อนไหวก็จะเริ่มเคลื่อนตัวเป็นมุมบวกจนกระทั่งเข้าสู่  
สถานะคงที่ที่เวลา 0.44 วินาที

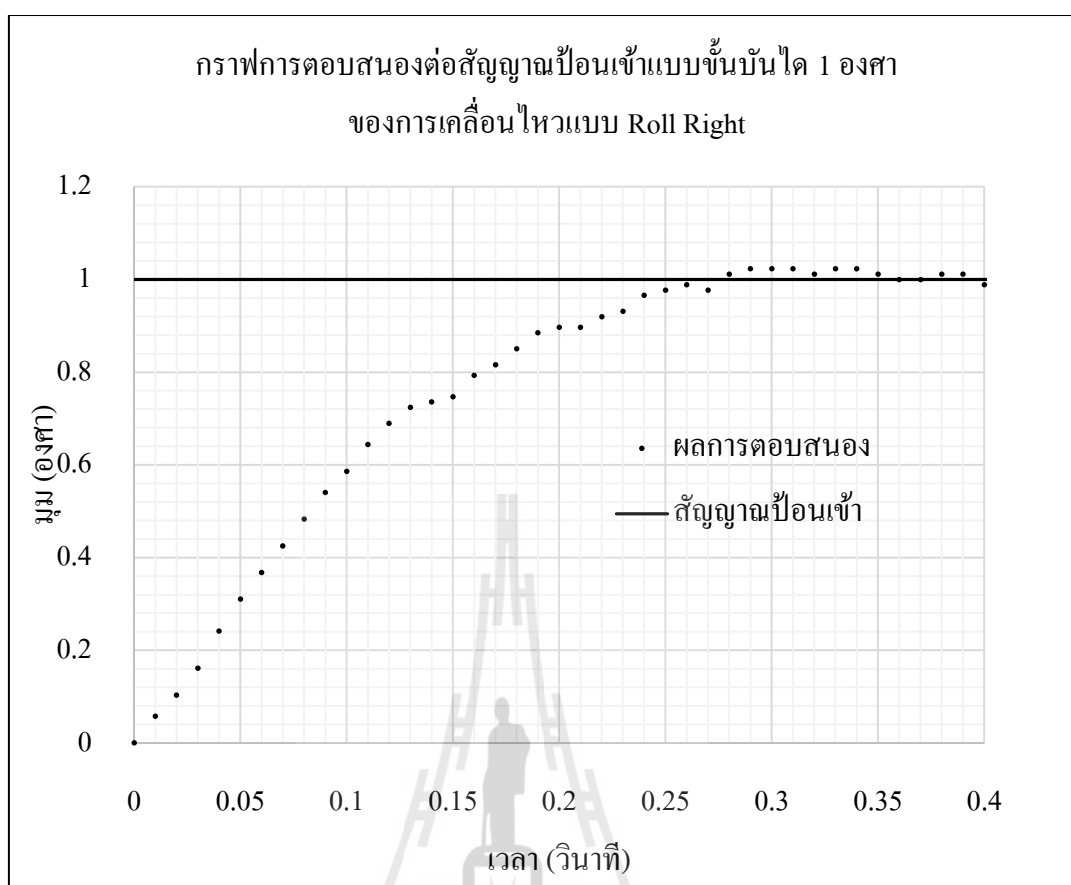
ค่าความคลาดเคลื่อนของแผ่นเคลื่อนไหวมากที่สุดอยู่ที่ตำแหน่งสภาวะคงที่ พบว่า  
มีค่าคลาดเคลื่อนจากคำสั่ง 0.03 องศา คิดเป็นเปอร์เซ็นต์ความคลาดเคลื่อน 3% ซึ่งเป็นค่าความคลาด  
เคลื่อนที่ยอมรับได้ กล่าวได้ว่าการตอบสนองแบบขั้นบันไดมีการตอบสนองถูกต้อง



รูปที่ 4.16 แสดงกราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบขั้นบันได 1 องศา  
ของการเคลื่อนไหวบแบบ Pitch Down

จากกราฟการตอบสนองจากฟังก์ชันแบบขั้นบันไดในการเคลื่อนที่รอบแกนตามสั่งในทิศทางลบ โดยให้แผ่นเคลื่อนไหว (Platform) เคลื่อนที่จากมุม 0 องศาถึง -1 องศา ผลการตอบสนองดังแสดงในรูปที่ 4.16 พบว่าแผ่นเคลื่อนไหวก็จะเริ่มเคลื่อนตัวเป็นมุมลบจนกระทั่งเข้าสู่สถานะคงที่ที่เวลา 0.35 วินาที

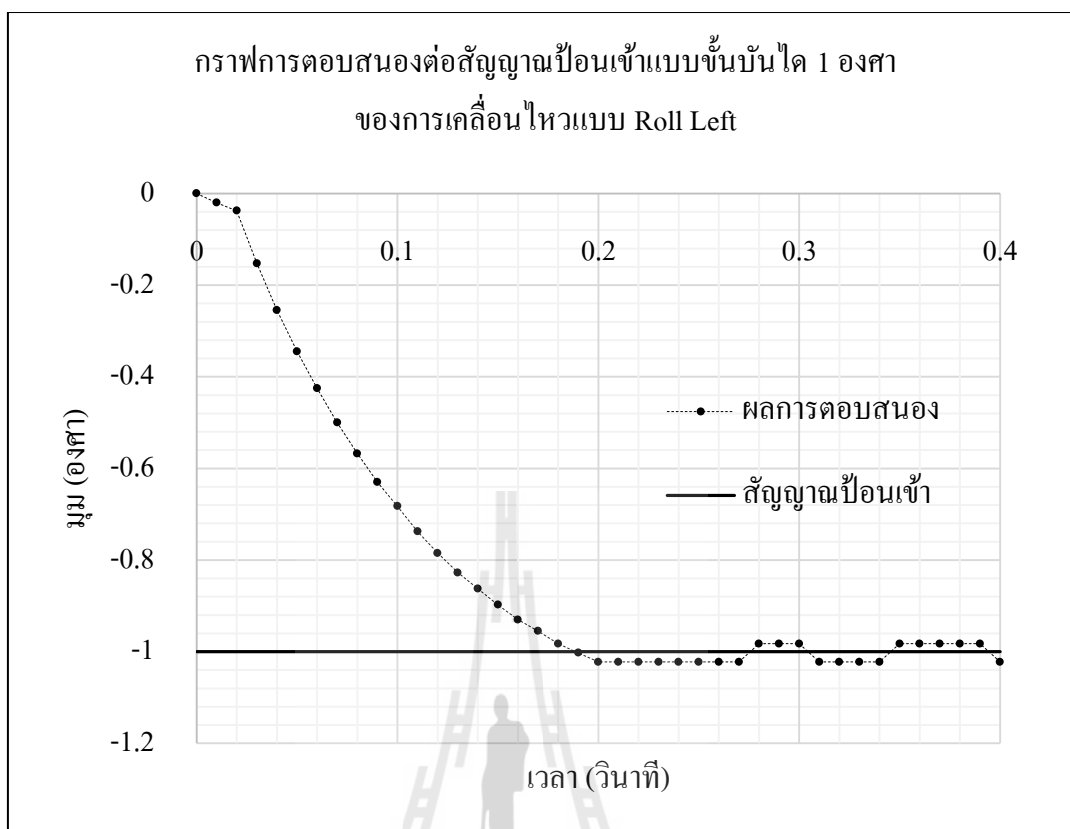
ค่าความคลาดเคลื่อนของแผ่นเคลื่อนไหวมามากที่สุดอยู่ที่ตำแหน่งสภาวะคงที่ พบว่ามีค่าคลาดเคลื่อนจากคำสั่ง 0.05 องศา คิดเป็นเปอร์เซ็นต์ความคลาดเคลื่อน 5% ซึ่งเป็นค่าความคลาดเคลื่อนที่ยอมรับได้ กล่าวได้ว่าการตอบสนองแบบขั้นบันไดมีการตอบสนองถูกต้องและแม่นยำ



รูปที่ 4.17 แสดงกราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบขั้นบันได 1 องศา  
ของการเคลื่อนไหวกแบบ Roll Right

จากกราฟการตอบสนองจากฟังก์ชันแบบขั้นบันไดในการเคลื่อนที่รอบแกนตามตั้ง  
ในทิศทางบวกโดยให้แผ่นเคลื่อนไหวก (Platform) เคลื่อนที่จากมุม 0 องศาถึง 1 องศา ผลการ  
ตอบสนองดังแสดงในรูปที่ 4.17 พบว่าแผ่นเคลื่อนไหวกก็จะเริ่มเคลื่อนตัวเป็นมุมบวกจนกระทั่งเข้าสู่  
สถานะคงที่ที่เวลา 0.37 วินาที

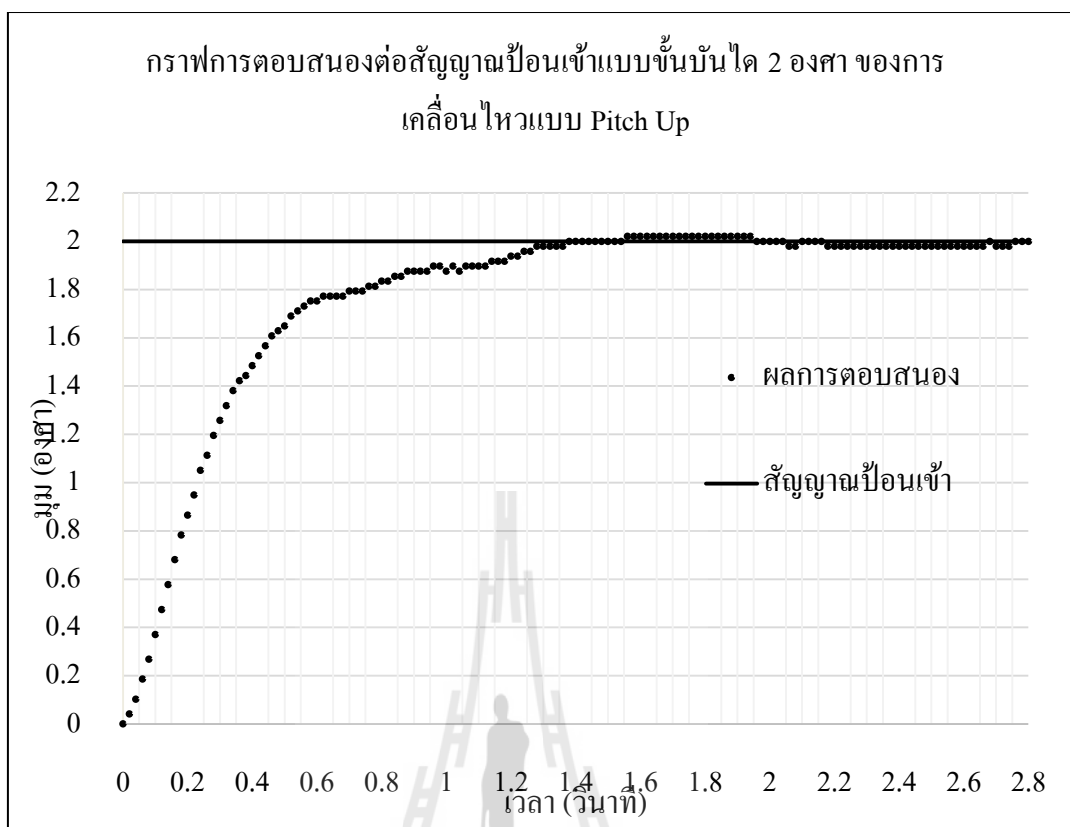
ค่าความคลาดเคลื่อนของแผ่นเคลื่อนไหวกมากที่สุดอยู่ที่ตำแหน่งสภาวะคงที่ พบว่า  
มีค่าคลาดเคลื่อนจากคำสั่ง 0.03 องศา คิดเป็นเปอร์เซ็นต์ความคลาดเคลื่อน 3% ซึ่งเป็นค่าความคลาด  
เคลื่อนที่ยอมรับได้ กล่าวได้ว่าการตอบสนองแบบขั้นบันไดมีการตอบสนองถูกต้อง



รูปที่ 4.18 แสดงกราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบขั้นบันได 1 องศา  
ของการเคลื่อนไหวนแบบ Roll Left

จากกราฟการตอบสนองจากฟังก์ชันแบบขั้นบันไดในการเคลื่อนที่รอบแกนตามตั้ง  
ในทิศทางลบโดยให้แผ่นเคลื่อนไหว (Platform) เคลื่อนที่จากมุม 0 องศาถึง -1 องศา ผลการ  
ตอบสนองดังแสดงในรูปที่ 4.18 พบว่าแผ่นเคลื่อนไหวก็จะเริ่มเคลื่อนตัวเป็นมุมลบจนกระทั่งเข้าสู่  
สถานะคงที่ที่เวลา 0.25 วินาทีซึ่งสัญญาณที่ป้อนใช้เวลาเคลื่อนที่ 0.3 วินาที

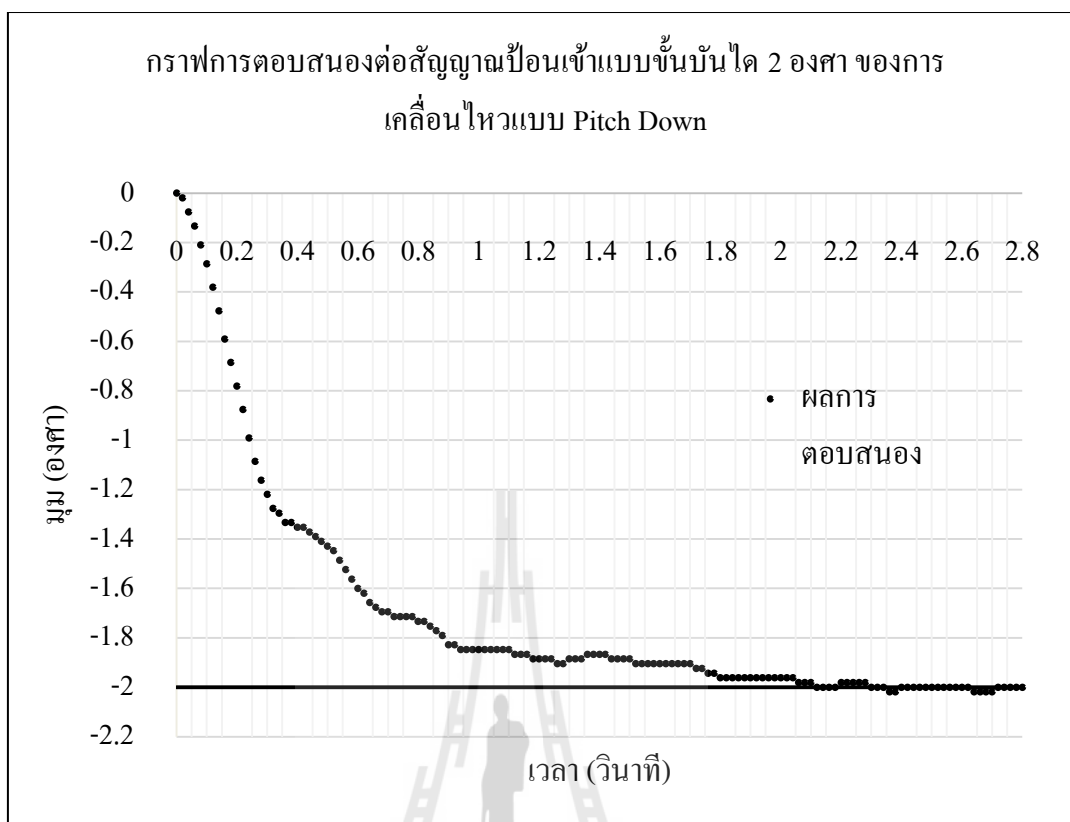
ค่าความคลาดเคลื่อนของแผ่นเคลื่อนไหวมากที่สุดอยู่ที่ตำแหน่งสภาวะคงที่ พบว่ามีค่า  
คลาดเคลื่อนจากคำสั่ง 0.02 องศา คิดเป็นเปอร์เซ็นต์ความคลาดเคลื่อน 2% ซึ่งเป็นค่าความคลาด  
เคลื่อนที่ยอมรับได้ กล่าวได้ว่าการตอบสนองแบบขั้นบันไดมีการตอบสนองถูกต้อง



รูปที่ 4.19 แสดงกราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบขั้นบันได 2 องศา ของการเคลื่อนไหวแบบ Pitch Up

จากกราฟการตอบสนองจากฟังก์ชันแบบขั้นบันไดในการเคลื่อนที่รอบแกนตามสั่งในทิศทางบวกโดยให้แผ่นเคลื่อนไหว (Platform) เคลื่อนที่จากมุม 0 องศาถึง 2 องศา ผลการตอบสนองดังแสดงในรูปที่ 4.19 พบว่าแผ่นเคลื่อนไหวก็จะเริ่มเคลื่อนตัวเป็นมุมบวกจนกระทั่งเข้าสู่สถานะคงที่ที่เวลา 1.40 วินาที

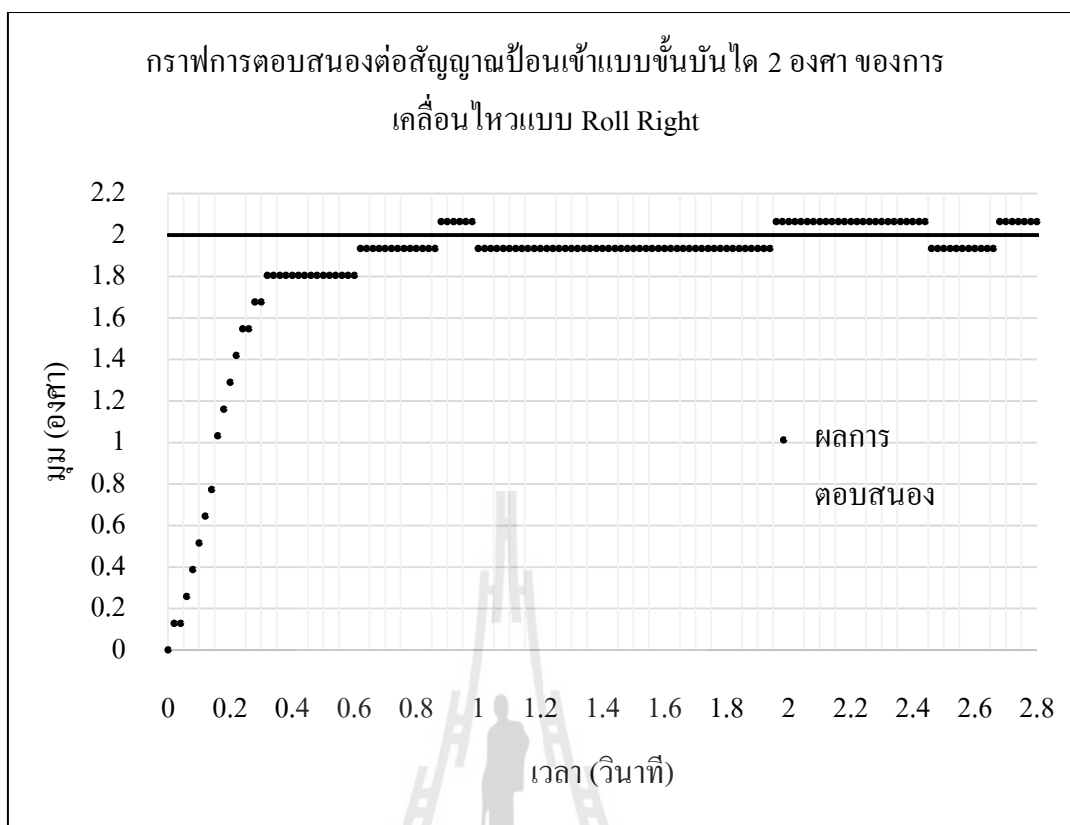
ค่าความคลาดเคลื่อนของแผ่นเคลื่อนไหวมากที่สุดอยู่ที่ตำแหน่งสถานะคงที่ พบว่ามีค่าคลาดเคลื่อนจากคำสั่ง 0.04 องศา คิดเป็นเปอร์เซ็นต์ความคลาดเคลื่อน 2% ซึ่งเป็นค่าความคลาดเคลื่อนที่ยอมรับได้ กล่าวได้ว่าการตอบสนองแบบขั้นบันไดมีการตอบสนองถูกต้อง



รูปที่ 4.20 แสดงกราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบขั้นบันได 2 องศา  
ของการเคลื่อนไหวยแบบ Pitch Down

จากกราฟการตอบสนองจากฟังก์ชันแบบขั้นบันไดในการเคลื่อนที่รอบแกนตามสั่ง  
ในทิศทางลบ โดยให้แผ่นเคลื่อนไหวย (Platform) เคลื่อนที่จากมุม 0 องศาถึง -2 องศา ผลการ  
ตอบสนองดังแสดงในรูปที่ 4.20 พบว่าแผ่นเคลื่อนไหวยก็จะเริ่มเคลื่อนตัวเป็นมุมลบจนกระทั่งเข้าสู่  
สถานะคงที่ที่เวลา 2.4 วินาที

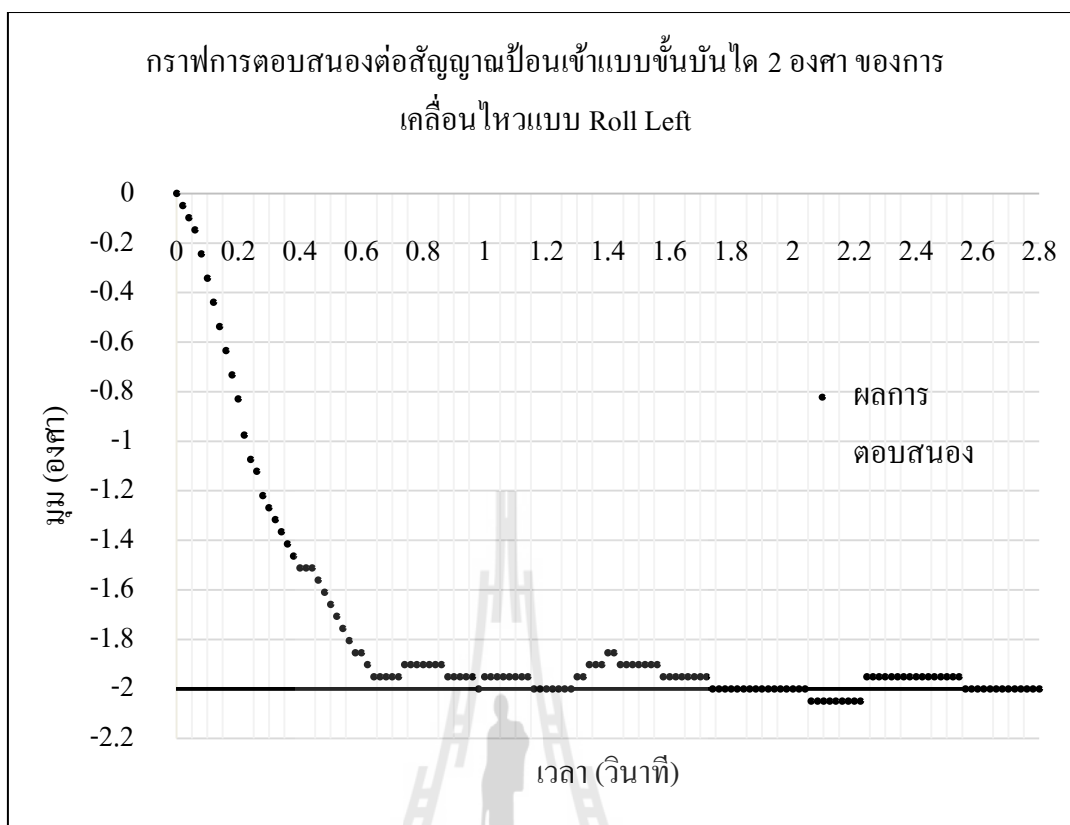
ค่าความคลาดเคลื่อนของแผ่นเคลื่อนไหวยมากที่สุดอยู่ที่ตำแหน่งสถานะคงที่ พบว่า  
มีค่าคลาดเคลื่อนจากคำสั่ง 0.02 องศา คิดเป็นเปอร์เซ็นต์ความคลาดเคลื่อน 1% ซึ่งเป็นค่าความคลาด  
เคลื่อนที่ยอมรับได้ กล่าวได้ว่าการตอบสนองแบบขั้นบันไดมีการตอบสนองถูกต้อง



รูปที่ 4.21 แสดงกราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบขั้นบันได 2 องศา ของการเคลื่อนไหวแบบ Roll Right

จากกราฟการตอบสนองจากฟังก์ชันแบบขั้นบันไดในการเคลื่อนที่รอบแกนตามสั่ง ในทิศทางบวกโดยให้แผ่นเคลื่อนไหว (Platform) เคลื่อนที่จากมุม 0 องศาถึง 2 องศา ผลการตอบสนองดังแสดงในรูปที่ 4.21 พบว่าแผ่นเคลื่อนไหวก็จะเริ่มเคลื่อนตัวเป็นมุมบวกจนกระทั่งเข้าสู่สถานะคงที่ที่เวลา 1.4 วินาที

ค่าความคลาดเคลื่อนของแผ่นเคลื่อนไหวมากที่สุดอยู่ที่ตำแหน่งสภาวะคงที่ พบว่ามีค่าคลาดเคลื่อนจากคำสั่ง 0.06 องศา คิดเป็นเปอร์เซ็นต์ความคลาดเคลื่อน 3% ซึ่งเป็นค่าความคลาดเคลื่อนที่ยอมรับได้ กล่าวได้ว่าการตอบสนองแบบขั้นบันไดมีการตอบสนองถูกต้อง



รูปที่ 4.22 แสดงกราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบขั้นบันได 2 องศา  
ของการเคลื่อนไหวนแบบ Roll Left

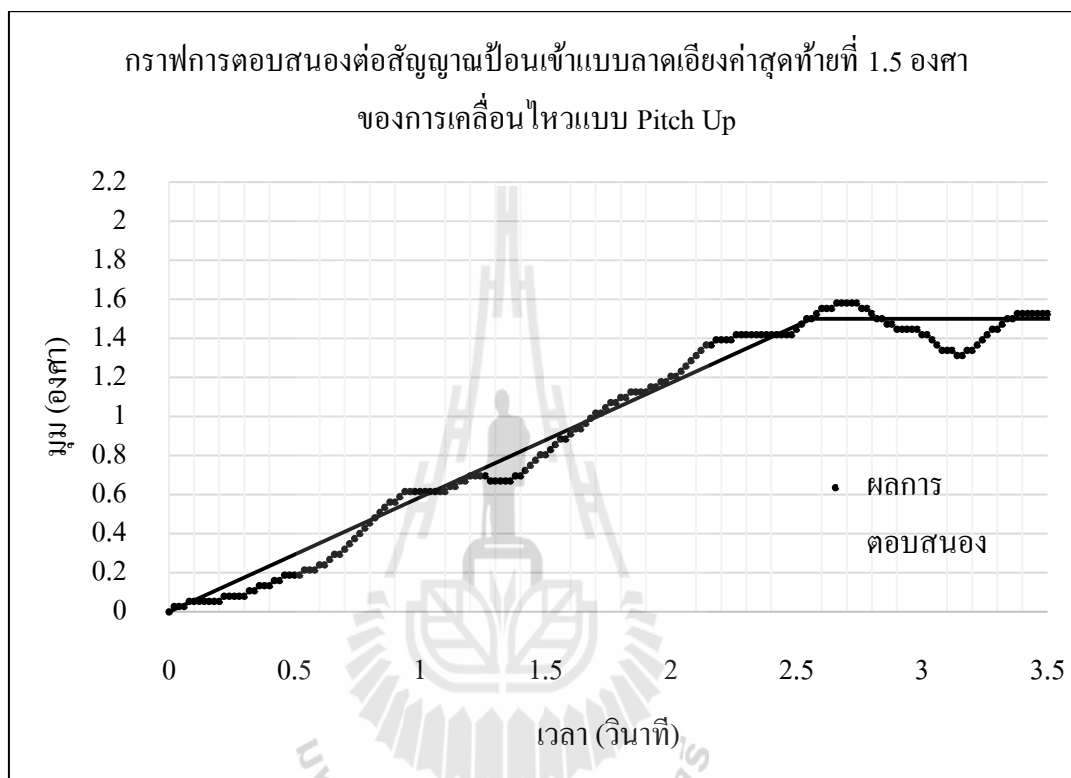
จากกราฟการตอบสนองจากฟังก์ชันแบบขั้นบันไดในการเคลื่อนที่รอบแกนตามสั่ง  
ในทิศทางลบโดยให้แผ่นเคลื่อนไหวน (Platform) เคลื่อนที่จากมุม 0 องศาถึง -2 องศา ผลการ  
ตอบสนองดังแสดงในรูปที่ 4.22 พบว่าแผ่นเคลื่อนไหวนก็จะเริ่มเคลื่อนตัวเป็นมุมลบจนกระทั่งเข้าสู่  
สถานะคงที่ที่เวลา 1.8 วินาที

ค่าความคลาดเคลื่อนของแผ่นเคลื่อนไหวนมากที่สุดอยู่ที่ตำแหน่งสภาวะคงที่ พบว่า  
มีค่าคลาดเคลื่อนจากคำสั่ง 0.05 องศา คิดเป็นเปอร์เซ็นต์ความคลาดเคลื่อน 2.5% ซึ่งเป็นค่า  
ความคลาดเคลื่อนที่ยอมรับได้ กล่าวได้ว่าการตอบสนองแบบขั้นบันไดมีการตอบสนองถูกต้อง



#### 4.5.4 การตอบสนองของระบบต่อฟังก์ชันลาดเอียงหนึ่งหน่วย (Unit Ramp Input)

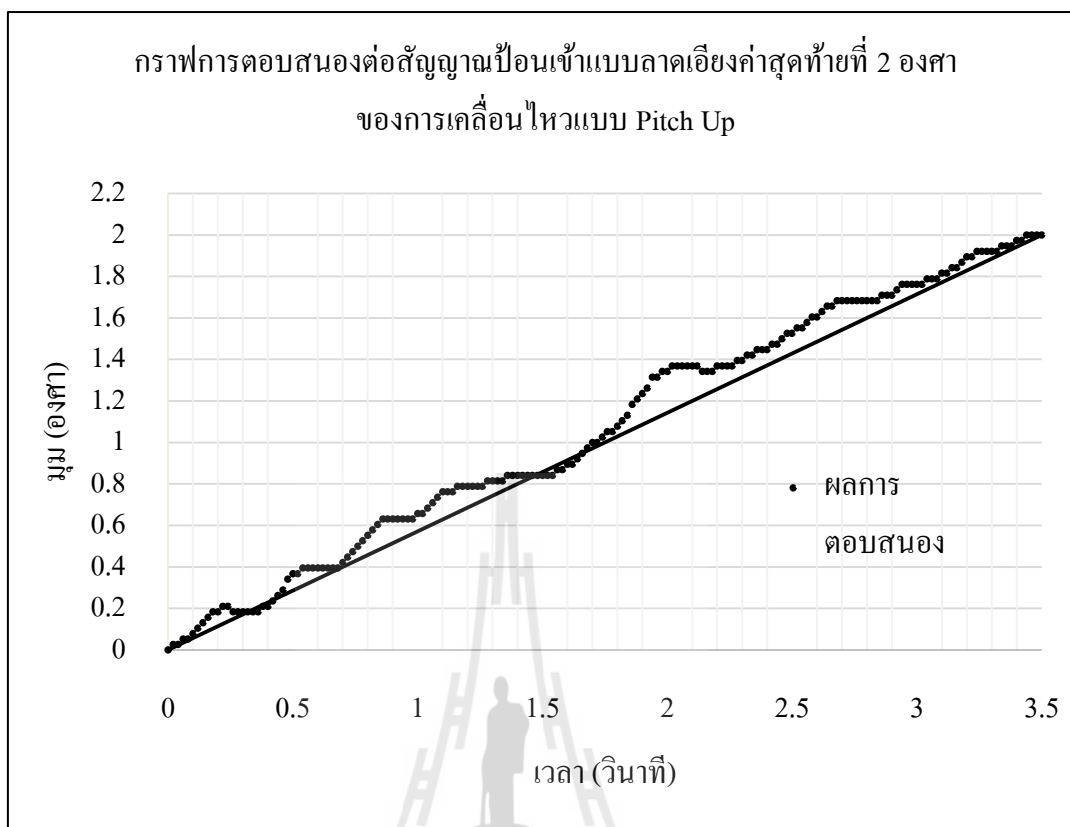
ในการทดสอบการตอบสนองฟังก์ชันขั้นบันได ผู้วิจัยได้ทดสอบการตอบสนอง ลักษณะการเคลื่อนที่ใน 2 ทิศทางคือการเคลื่อนที่รอบแกนตามขวาง (Pitch) และการเคลื่อนที่รอบ แกนตามยาว (Roll) ซึ่งมีผลการตอบสนองดังนี้



รูปที่ 4.23 แสดงกราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบลาดเอียงค่าสุดท้ายที่ 1.5 องศา  
ของการเคลื่อนไหวแบบ Pitch Up

จากกราฟการตอบสนองในรูปที่ 4.23 จะพบว่า การตอบสนองต่อสัญญาณป้อนเข้าแบบฟังก์ชันลาดเอียงนั้นเป็นไปได้อย่างมีประสิทธิภาพ โดยระบบพยายามเคลื่อนไหวให้เข้าสู่สัญญาณป้อนเข้าตลอดเวลา โดยมีค่าความคลาดเคลื่อนมากที่สุด คือ 5% ซึ่งสามารถยอมรับได้

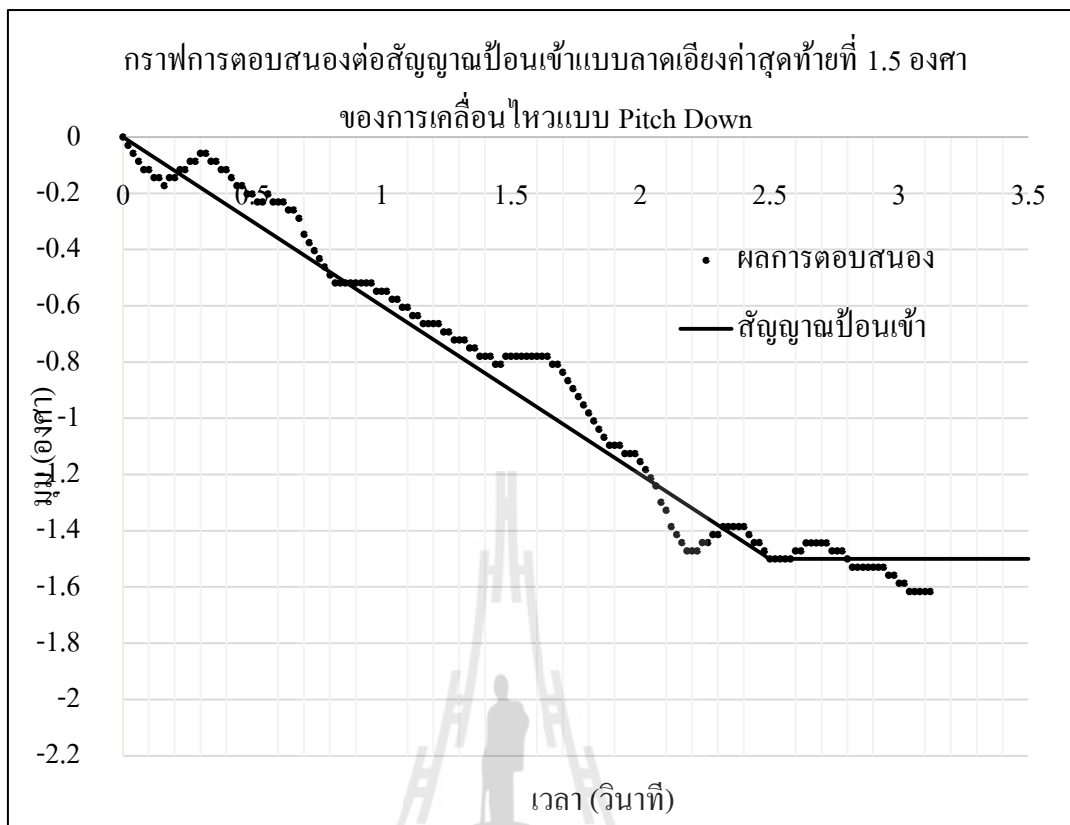
หากพิจารณาการตอบสนองต่อสัญญาณป้อนเข้าแบบฟังก์ชันลาดเอียงจะพบว่า ระบบพยายามเข้าสู่สมดุล จึงเป็นผลที่น่าพอใจและยอมรับได้ของการตอบสนองต่อสัญญาณป้อนเข้านี้



รูปที่ 4.24 แสดงกราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบลาดเอียงค่าสุดท้ายที่ 2 องศา  
ของการเคลื่อนไหวนแบบ Pitch Up

จากกราฟการตอบสนองในรูปที่ 4.24 จะพบว่า การตอบสนองต่อสัญญาณป้อนเข้าแบบฟังก์ชันลาดเอียงนั้นเป็นไปได้อย่างมีประสิทธิภาพ โดยระบบพยายามเคลื่อนไหวก้าวเข้าสู่สัญญาณป้อนเข้าตลอดเวลา โดยมีค่าความคลาดเคลื่อนมากที่สุด คือ 10% ซึ่งสามารถยอมรับได้

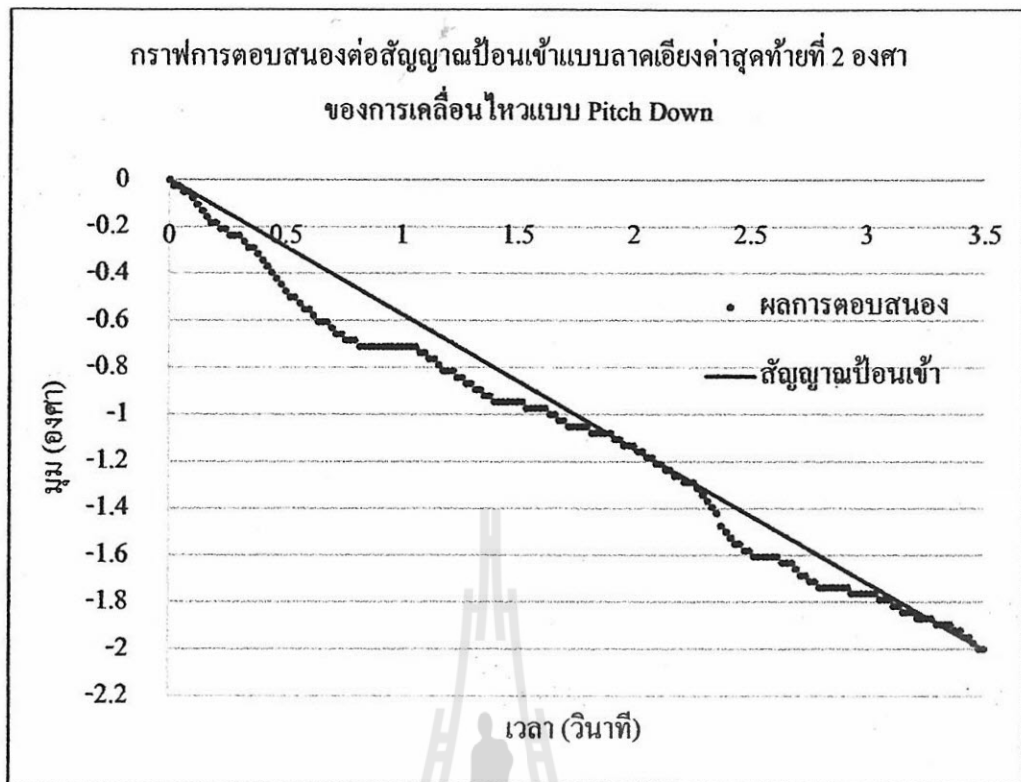
หากพิจารณาการตอบสนองต่อสัญญาณป้อนเข้าแบบฟังก์ชันลาดเอียงจะพบว่าระบบพยายามเข้าสู่สมดุล จึงเป็นผลที่น่าพอใจและยอมรับได้ของการตอบสนองต่อสัญญาณป้อนเข้านี้



รูปที่ 4.25 แสดงกราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบลาดเอียงค่าสุดท้ายที่ 1.5 องศา  
ของการเคลื่อนไหวแบบ Pitch Down

จากกราฟการตอบสนองในรูปที่ 4.25 จะพบว่า การตอบสนองต่อสัญญาณป้อนเข้าแบบฟังก์ชันลาดเอียงนั้นเป็นไปได้อย่างมีประสิทธิภาพ โดยระบบพยายามเคลื่อนไหวให้เข้าสู่สัญญาณป้อนเข้าตลอดเวลา โดยมีค่าความคลาดเคลื่อนมากที่สุด คือ 10% ซึ่งสามารถยอมรับได้

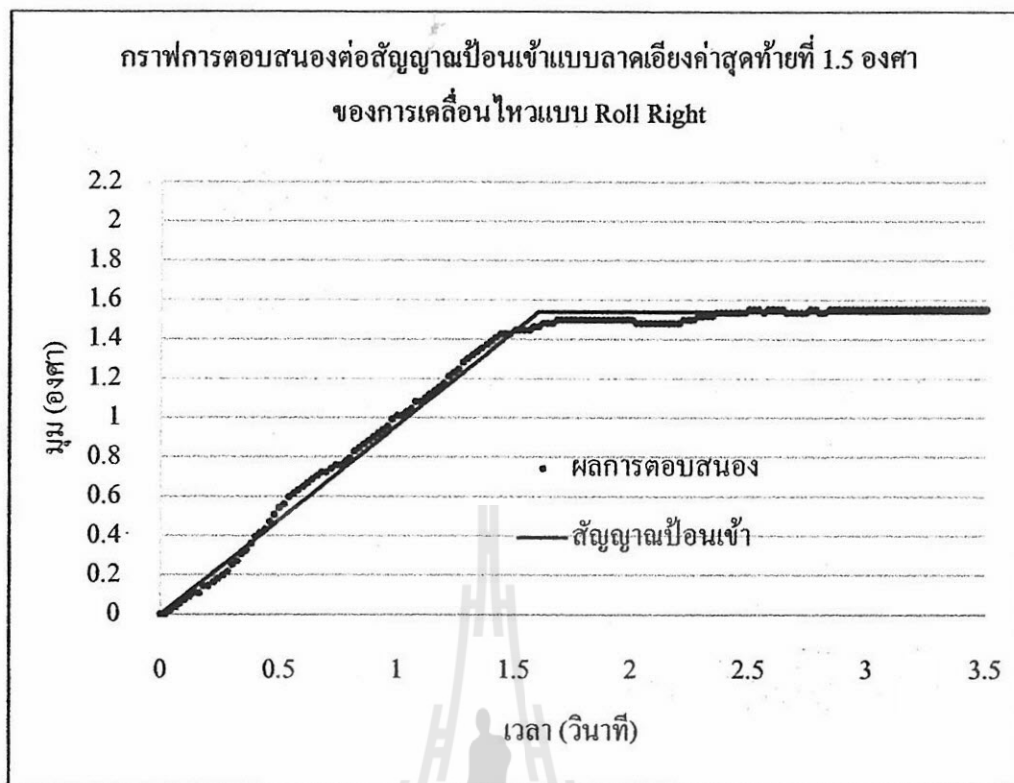
หากพิจารณาการตอบสนองต่อสัญญาณป้อนเข้าแบบฟังก์ชันลาดเอียงจะพบว่า ระบบพยายามเข้าสู่สมดุล จึงเป็นผลที่น่าพอใจและยอมรับได้ของการตอบสนองต่อสัญญาณป้อนเข้านี้



รูปที่ 4.26 แสดงกราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบลาดเอียงค่าสุดท้ายที่ 2 องศา  
ของการเคลื่อนไหวแบบ Pitch Down

จากกราฟการตอบสนองในรูปที่ 4.26 จะพบว่า การตอบสนองต่อสัญญาณป้อนเข้าแบบฟังก์ชันลาดเอียงนั้นเป็นไปได้อย่างมีประสิทธิภาพ โดยระบบพยายามเคลื่อนไหวให้เข้าสู่สัญญาณป้อนเข้าตลอดเวลา โดยมีค่าความคลาดเคลื่อนมากที่สุด คือ 10% ซึ่งสามารถยอมรับได้

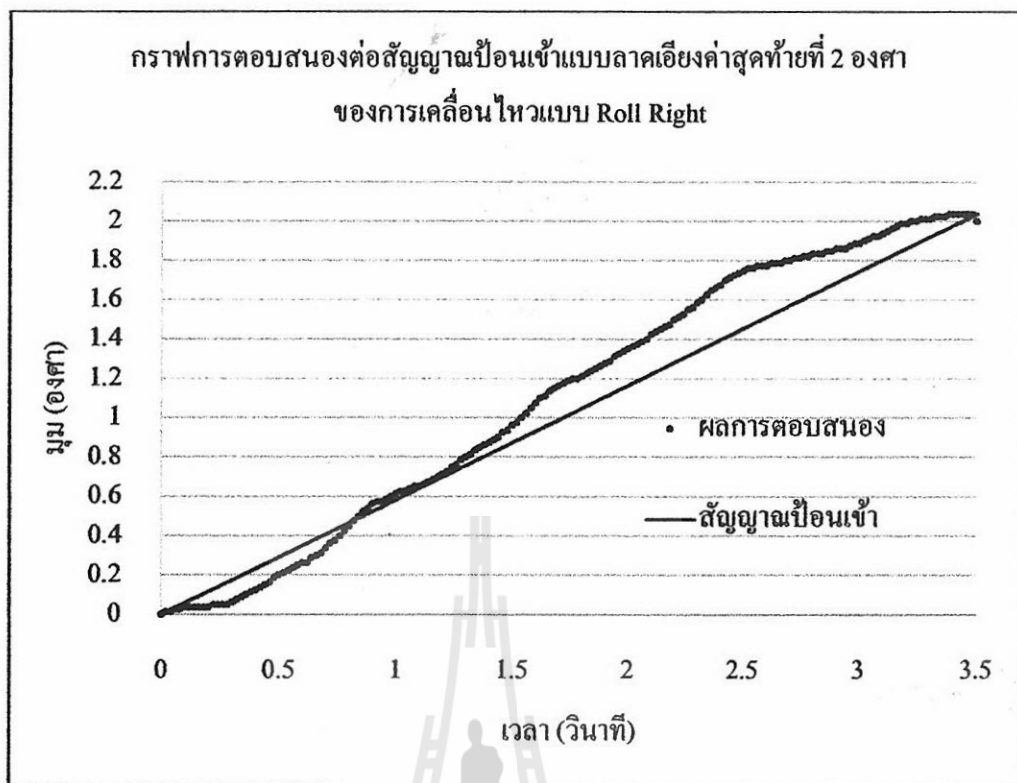
หากพิจารณาการตอบสนองต่อสัญญาณป้อนเข้าแบบฟังก์ชันลาดเอียงจะพบว่า ระบบพยายามเข้าสู่สมดุล จึงเป็นผลที่น่าพอใจและยอมรับได้ของการตอบสนองต่อสัญญาณป้อนเข้านี้



รูปที่ 4.27 แสดงกราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบลาดเอียง 1.5 องศา  
ของการเคลื่อนไหวนแบบ Roll Right

จากกราฟการตอบสนองในรูปที่ 4.27 จะพบว่าการตอบสนองต่อสัญญาณป้อนเข้าแบบฟังก์ชันลาดเอียงนั้นเป็นไปได้อย่างมีประสิทธิภาพ โดยระบบพยายามเคลื่อนไหวนให้เข้าสู่สัญญาณป้อนเข้าตลอดเวลา โดยมีค่าความคลาดเคลื่อนมากที่สุด คือ 2% ซึ่งสามารถยอมรับได้

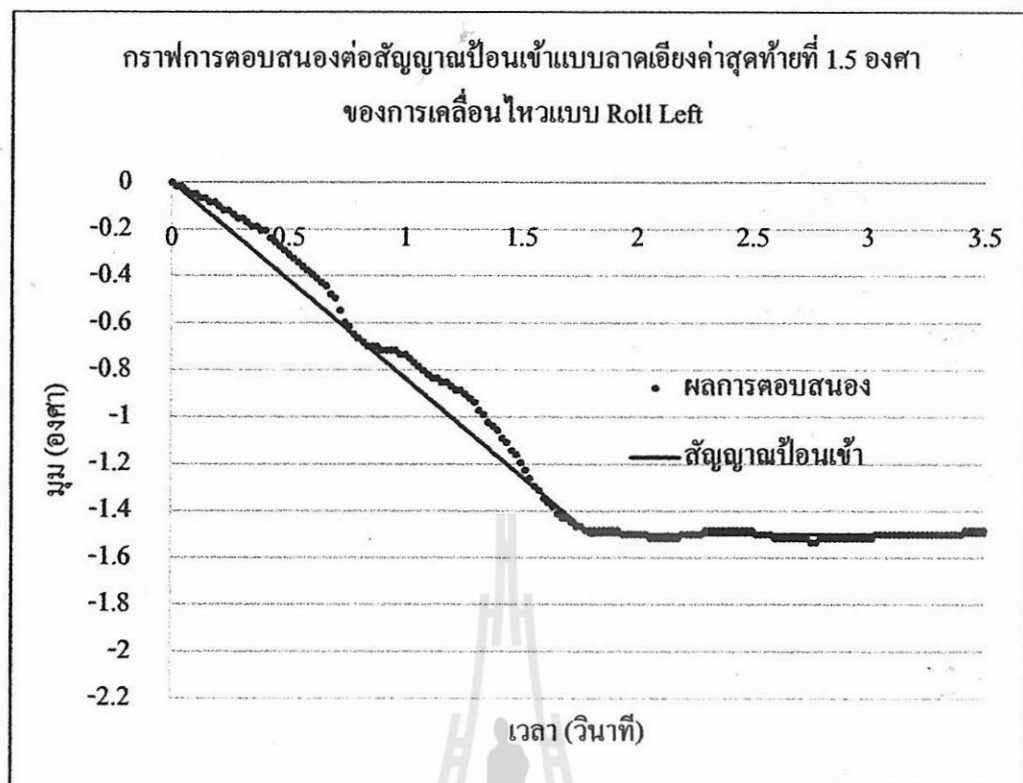
หากพิจารณาการตอบสนองต่อสัญญาณป้อนเข้าแบบฟังก์ชันลาดเอียงจะพบว่าระบบพยายามเข้าสู่สมดุล จึงเป็นผลที่น่าพอใจและยอมรับได้ของการตอบสนองต่อสัญญาณป้อนเข้านี้



รูปที่ 4.28 แสดงกราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบลาดเอียง 2 องศา  
ของการเคลื่อนไหวแบบ Roll Right

จากกราฟการตอบสนองในรูปที่ 4.28 จะพบว่า การตอบสนองต่อสัญญาณป้อนเข้าแบบฟังก์ชันลาดเอียงนั้นเป็นไปได้อย่างมีประสิทธิภาพ โดยระบบพยายามเคลื่อนไหวให้เข้าสู่สัญญาณป้อนเข้าตลอดเวลา โดยมีค่าความคลาดเคลื่อนมากที่สุด คือ 15%

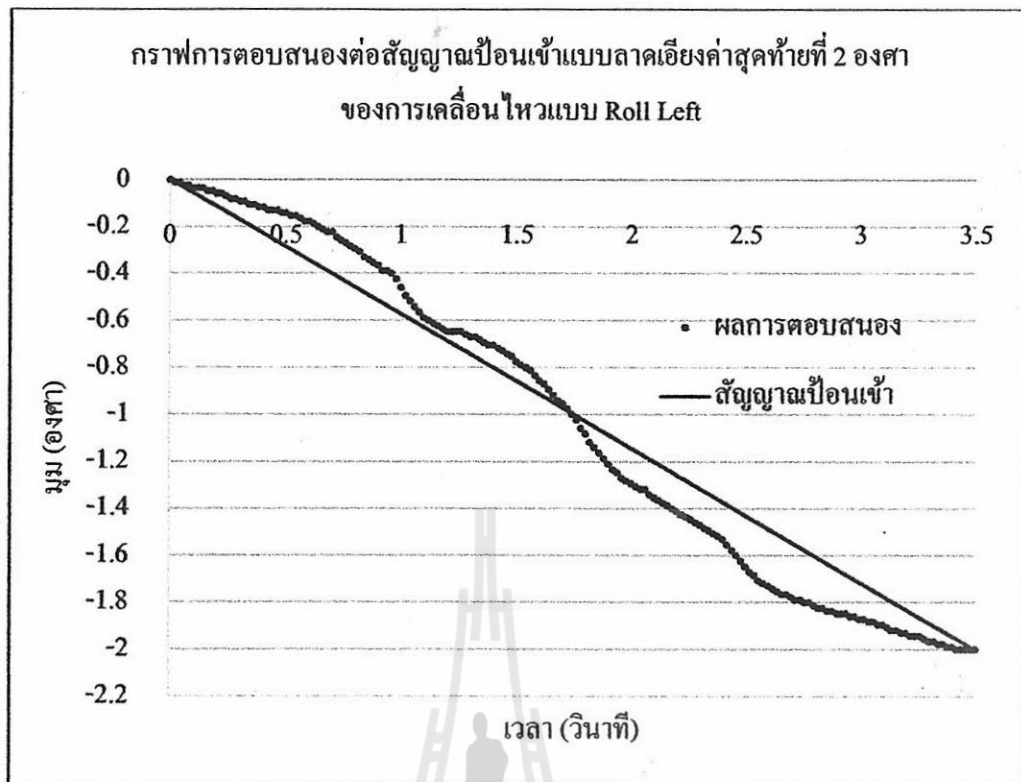
หากพิจารณาการตอบสนองต่อสัญญาณป้อนเข้าแบบฟังก์ชันลาดเอียงจะพบว่าระบบพยายามเข้าสู่สมดุล จึงเป็นผลที่น่าพอใจและยอมรับได้ของการตอบสนองต่อสัญญาณป้อนเข้านี้



รูปที่ 4.29 แสดงกราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบลาดเอียงค่าสุดท้ายที่ 1.5 องศา  
ของการเคลื่อนไหวแบบ Roll Left

จากกราฟการตอบสนองในรูปที่ 4.29 จะพบว่า การตอบสนองต่อสัญญาณป้อนเข้าแบบฟังก์ชันลาดเอียงนั้นเป็นไปได้อย่างมีประสิทธิภาพ โดยระบบพยายามเคลื่อนไหวให้เข้าสู่สัญญาณป้อนเข้าตลอดเวลา โดยมีค่าความคลาดเคลื่อนมากที่สุด คือ 10% ซึ่งสามารถยอมรับได้

หากพิจารณาการตอบสนองต่อสัญญาณป้อนเข้าแบบฟังก์ชันลาดเอียงจะพบว่าระบบพยายามเข้าสู่สมดุล จึงเป็นผลที่น่าพอใจและยอมรับได้ของการตอบสนองต่อสัญญาณป้อนเข้านี้



รูปที่ 4.30 แสดงกราฟการตอบสนองต่อสัญญาณป้อนเข้าแบบลาดเอียงค่าสุดท้ายที่ 2 องศา  
ของการเคลื่อนไหวกแบบ Roll Left

จากกราฟการตอบสนองในรูปที่ 4.30 จะพบว่าการตอบสนองต่อสัญญาณป้อนเข้าแบบฟังก์ชันลาดเอียงนั้นเป็นไปได้เป็นอย่างดี โดยระบบพยายามเคลื่อนไหวกให้เข้าสู่สัญญาณป้อนเข้าตลอดเวลา โดยมีค่าความคลาดเคลื่อนมากที่สุด คือ 10% ซึ่งสามารถยอมรับได้

หากพิจารณาการตอบสนองต่อสัญญาณป้อนเข้าแบบฟังก์ชันลาดเอียงจะพบว่าระบบพยายามเข้าสู่สมดุล จึงเป็นผลที่น่าพอใจและยอมรับได้ของการตอบสนองต่อสัญญาณป้อนเข้านี้

สรุปผลการตอบสนองต่อสัญญาณป้อนเข้าแบบฟังก์ชันลาดเอียงพบว่า ระบบพยายามควบคุมเข้าสู่มุมของสัญญาณป้อนเข้าแม้ระหว่างการควบคุมอาจมีค่าความคลาดเคลื่อน แต่สามารถยอมรับได้ ดังนั้นจึงสรุปว่าระบบมีความถูกต้องในการทำงานควบคุมแผ่นเคลื่อนไหวกที่ค่าความคลาดเคลื่อน 2-10%



## บทที่ 5

### สรุปผลและข้อเสนอแนะ

#### 5.1 สรุปผลงานวิจัย

งานวิจัยนี้เป็นการออกแบบเครื่องจำลองการบิน ที่สามารถเคลื่อนที่ได้ 3 องศาอิสระ (Roll, Pitch, และ Heave) เพื่อศึกษาการเชื่อมโยงระหว่างโปรแกรมจำลองการบิน และเครื่องจำลองการเคลื่อนที่ ซึ่งจากการศึกษาเครื่องจำลองการบินผู้วิจัยได้ทำการดึงค่าข้อมูลท่าทางการเคลื่อนที่แบบ Roll แบบ Pitch และค่าความเร่งของเครื่องบินจากโปรแกรมจำลองการบินแล้วนำมาแปลงเป็นการเคลื่อนที่ของเครื่องจำลองการบิน โดยผลการวิจัยเครื่องจำลองการบินที่ออกแบบสามารถเชื่อมโยงกับโปรแกรมจำลองการบินที่มีอยู่ในท้องตลาด และอุปกรณ์ควบคุมการทำงานของเครื่องจำลองการบินได้ตามจุดประสงค์ที่วางไว้ และสามารถนำมาสร้างเครื่องจำลองการบินจริงได้จริงอีกด้วย จากการทดลองพบว่า การควบคุมการเคลื่อนที่ของเครื่องจำลองการบินต้นแบบนั้น สามารถทำงานได้อย่างมีประสิทธิภาพ โดยมีความคลาดเคลื่อนสูงสุดของการควบคุมมุมองศา การเคลื่อนที่ของเครื่องจำลองการบิน 10 เปอร์เซ็นต์ของการควบคุมการเคลื่อนที่ทั้ง Pitch และ Roll และมีขีดจำกัดในการเคลื่อนไหวแบบ Pitch คือ  $\pm 5^\circ$  การเคลื่อนไหวแบบ Roll คือ  $\pm 3^\circ$  และการเคลื่อนที่เคลื่อนลงในแนวแกนตั้งฉาก  $\pm 25$  มิลลิเมตร

#### 5.2 ข้อเสนอแนะ

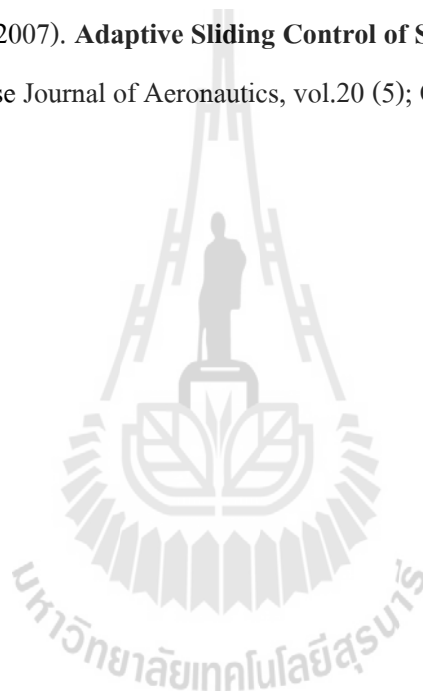
- งานวิจัยนี้เป็นงานวิจัยที่ใช้ระบบควบคุมแบบมีสัญญาณป้อนกลับเฉพาะแกนของมอเตอร์ต้นกำลัง ดังนั้นจึงอาจมีค่าความคลาดเคลื่อนเกิดขึ้นหากต้องการให้มีความถูกต้องเพิ่มขึ้น สามารถใช้ระบบควบคุมแบบปิด โดยทำการติดตั้งเซนเซอร์วัดท่าทางการเคลื่อนที่เพิ่มเข้าไปในระบบเพื่อป้อนสัญญาณการตรวจสอบท่าทางและทำการแก้ไขให้ได้ท่าทางที่ถูกต้องมากยิ่งขึ้น

- การสร้างเครื่องจำลองควรสำรวจอุปกรณ์ที่มีในท้องตลาดว่ามีขนาดตามที่ได้ทำการออกแบบไว้หรือไม่ เพื่อให้เครื่องจำลองการบินนั้นมีประสิทธิภาพสูงสุด

## รายการอ้างอิง

- กฤษฎา จันทร์แดง. (2551). การจำลองพลศาสตร์ยานยนต์ร่วมกับยานพาหนะจริงแบบย่อส่วน (Collaboration of Vehicle Dynamic Simulation with Scaling Vehicle). วิทยานิพนธ์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย.
- สมฤกษ์ ปุจฉาการ คมกริช ทองสะอาด จรุงชัย แซ่เอี้ยว จิรพงศ์ ลิ้ม และ สายันต์ พรายมี. (2555). การพัฒนาแขนหุ่นยนต์ 3 แกนสำหรับเคลื่อนย้ายวัสดุ (The Development of 3-Axis Robot Arm for Material Handling). การประชุมวิชาการช่างงานวิศวกรรมอุตสาหกรรม ประจำปี พ.ศ. 2555, 17-19 ตุลาคม, เพชรบุรี, 6 หน้า.
- สุทธิวัฒน์ ประชาชนะชัย และ สุวัฒน์ กุลธนปรีดา. (2548). การจำลองการทำงานของแผ่นเคลื่อนไหว 6 องศาอิสระที่ขับเคลื่อนด้วยไฮดรอลิก (Simulation of 6-DOF Hydraulic Driven Motion Platforms). การประชุมวิชาการเครือข่ายวิศวกรรมเครื่องกลแห่งประเทศไทย ครั้งที่ 19, 19-21 ตุลาคม, เพชรบุรี, 6 หน้า.
- Abdul Jalil, M.K., (n.d.). **Design and Development of 6-DOF Motion Platform for Vehicle Driving Simulator**. Design Department, Faculty of Mechanical Engineering, Universiti Teknologi Malaysia.
- Andeen, G.B. (1988). **Robot Design Handbook**, McGraw-Hill: New York; 610 pp.
- Asada, H., and Slotine, J.J.E. (1986). **Robot Analysis and Control**, Handbook, Wiley & Son; April, 288 pp.
- Casas, S., and Alcaraz, J.M., Olanda, R., Coma, I., and Fernandez, M. (2014). **Towards an Extensible Simulator of Real Motion Platform**. Simulation Modelling Practice and Theory, vol.45; June, pp.50-61.
- Chiew, Y.S., Abdul Jali, M.K., and Hussein, M. (2008). Kinematic Modeling of Driving Simulator Motion Platform. **Proceedings of the Conference on Innovative Technologies in Intelligent Systems and Industrial Applications**, Malaysia; July 12-13, pp.30-34.
- Dasgupta, B., and Mruthyunjaya, T.S. (1998). **A Newton-Euler formulation for the inverse dynamics of the Stewart platform manipulator**, Mechanism and Machine Theory, vol.33 (8); November, pp.1135-1152

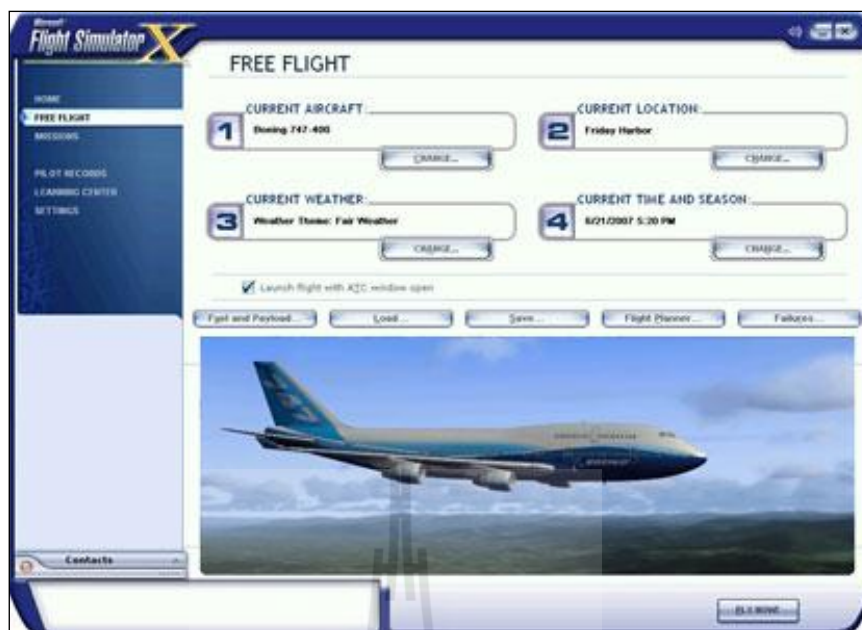
- Kim, H.S., Cho, Y.M., and Lee, K.I. (2002). **Robust nonlinear task space control for a 6 DOF parallel manipulator**, Automatica, vol.41 (9); September, pp.1591-1600
- Kroger, T., and Wahl, F.M. (2009). **Advances in Robotics Research: Theory, Implementation, Application**, Springer; 358 pp.
- Rekdalsbakken, W. (2005). **Design and Application of a Motion Platform in Three Degrees of Freedom**. Proceedings of the 40<sup>th</sup> Conference of Simulation and Modelling, pp.269-279.
- Spong, M.W., and Vidyasagar, M. (1989). **Robot Dynamics and Control**, Wiley & Son: New York; 352 pp.
- Wu, D.S., and Gu, H.B. (2007). **Adaptive Sliding Control of Six-DOF Flight Simulator Motion Platform**, Chinese Journal of Aeronautics, vol.20 (5); October, pp.425-433





**ภาคผนวก ก.**

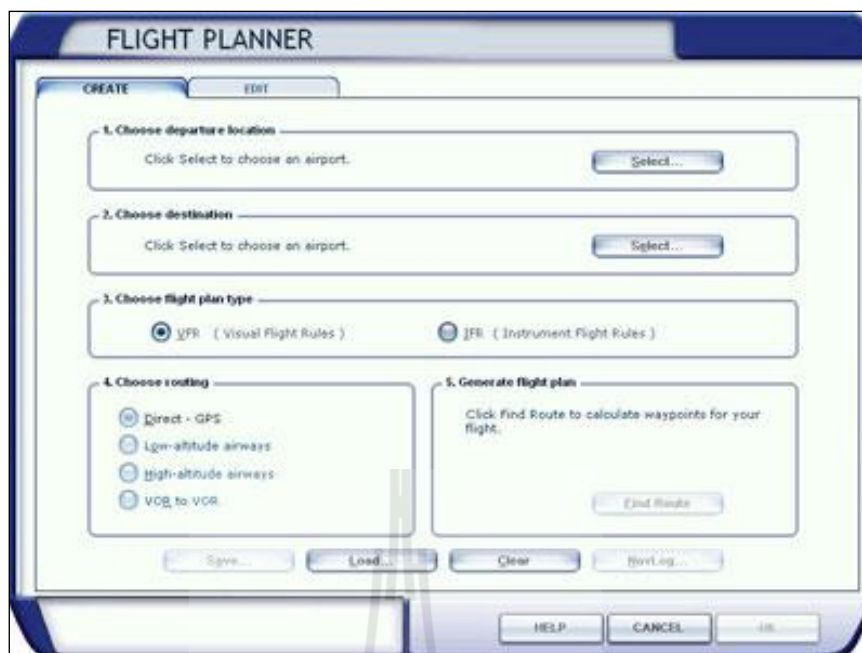
**รายละเอียดโปรแกรมจำลองการบิน**



รูปที่ ก.1 แสดงโหมดการบินเดี่ยวสำหรับผู้ที่ต้องการชมทิวทัศน์สำหรับโปรแกรม Flight Simulator X



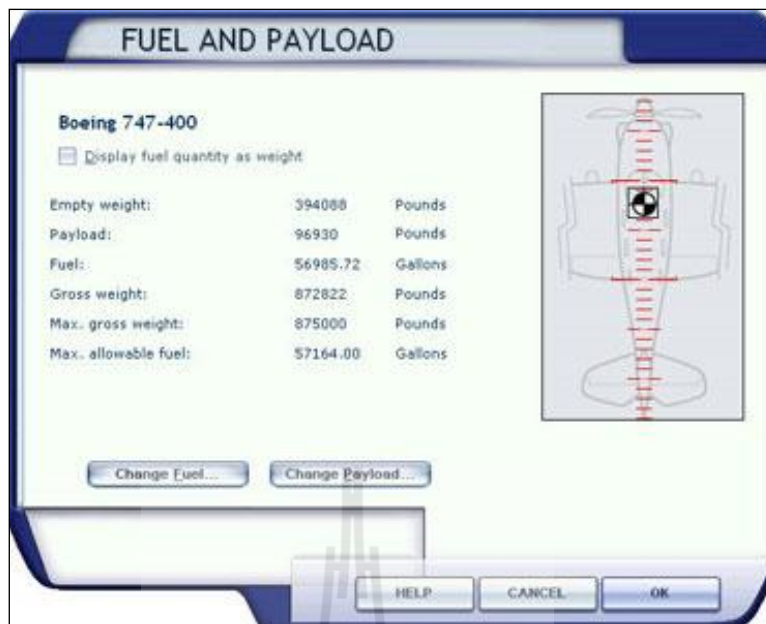
รูปที่ ก.2 แสดงหน้าต่างการเลือกอากาศยานที่ต้องการสำหรับโปรแกรม Flight Simulator X



รูปที่ ก.3 แสดงหน้าต่างกำหนดแผนการบินที่เราต้องการสำหรับโปรแกรม Flight Simulator X



รูปที่ ก.4 แสดงหน้าต่างสภาพอากาศจริงของพื้นที่ที่ทำการบินสำหรับโปรแกรม Flight Simulator X

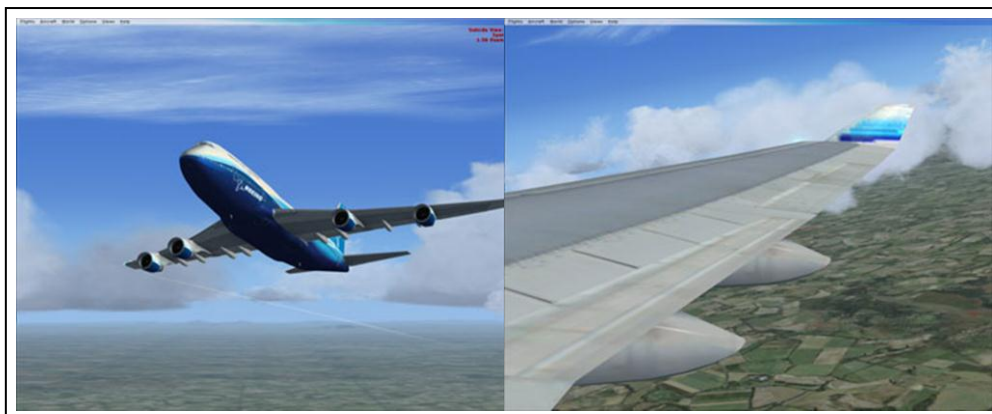


รูปที่ ก.5 แสดงหน้าต่างกำหนดเชื้อเพลิงในการใช้งานและน้ำหนักการบรรทุก ซึ่งมีผลต่อระยะทางและการทรงตัวสำหรับ โปรแกรม Flight Simulator X



รูปที่ ก.6 แสดงมุมมองห้องนักบิน ผู้ใช้สามารถควบคุมแผงควบคุมได้ทุกอย่างสำหรับ โปรแกรม Flight Simulator X





รูปที่ ก.7 แสดงมุมมองภายนอกอากาศยานสำหรับโปรแกรม Flight Simulator X



รูปที่ ก.8 แสดงระบบหลายหน้าต่างทำให้เราสามารถเห็นสิ่งต่าง ๆ ในเวลาเดียวกัน  
สำหรับ โปรแกรม Flight Simulator X





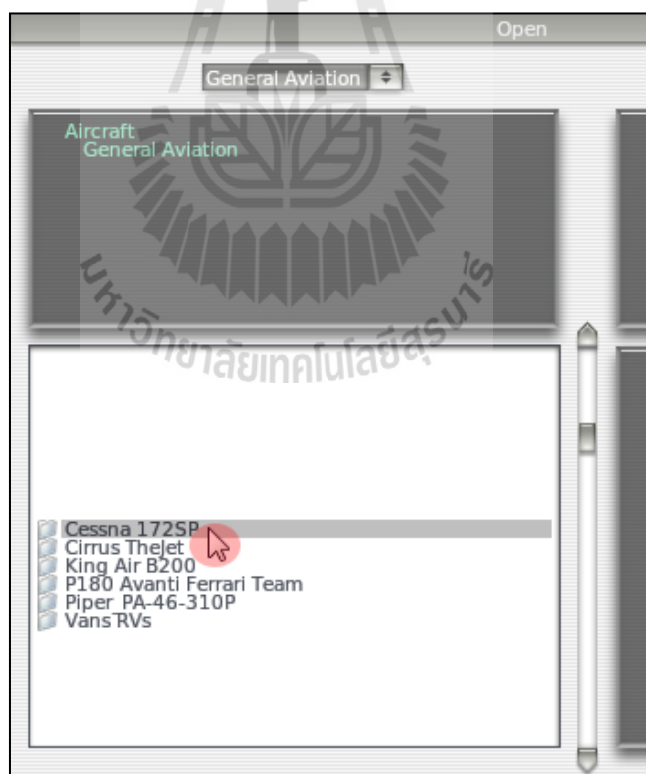
รูปที่ ก.9 แสดงระบบนำร่องอากาศยาน (GPS) สำหรับโปรแกรม Flight Simulator X



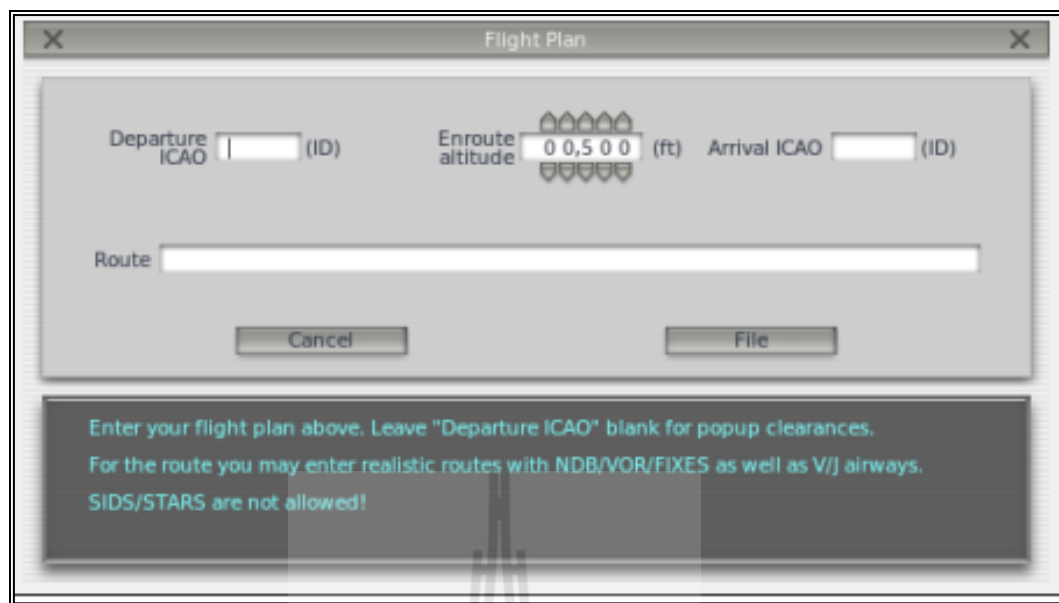
รูปที่ ก.10 แสดงหมวดการเรียนรู้เกี่ยวกับเครื่องบินสำหรับโปรแกรม Flight Simulator X



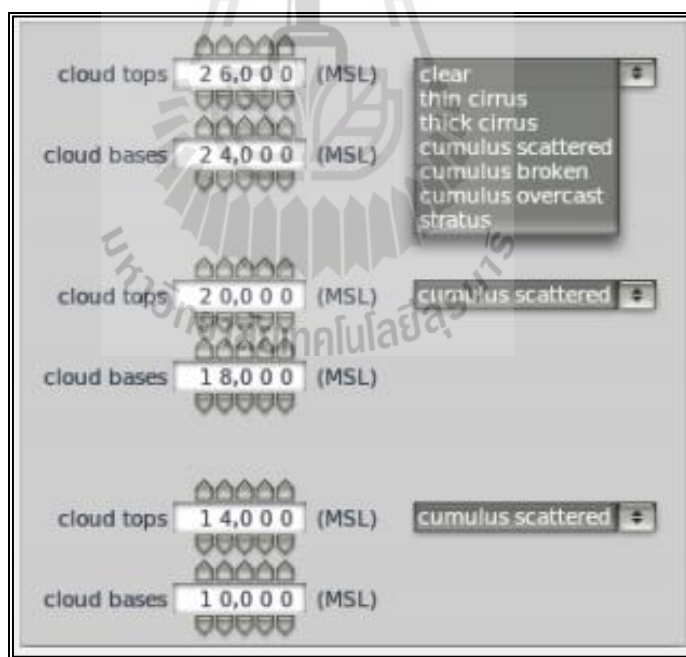
รูปที่ ก.11 แสดงโหมดการบินเดี่ยวสำหรับผู้ที่ต้องการชมทิวทัศน์สำหรับโปรแกรม X-Plane



รูปที่ ก. 12 แสดงหน้าต่างการเลือกอากาศยานที่ต้องการสำหรับ โปรแกรม X-Plane



รูปที่ ก.13 แสดงหน้าต่างกำหนดแผนการบินที่เราต้องการสำหรับ โปรแกรม X-Plane



รูปที่ ก.14 แสดงหน้าต่างสภาพอากาศ ณ เวลาจริงของพื้นที่ที่ทำการบินสำหรับ โปรแกรม X-Plane



รูปที่ ก.15 แสดงมุมมองห้องนักบิน ผู้ใช้สามารถควบคุมแผงควบคุมได้ทุกอย่าง  
สำหรับโปรแกรม X-Plane



รูปที่ ก. 16 แสดงมุมมองภายนอกอากาศยานสำหรับ โปรแกรม X-Plane



รูปที่ ก.17 แสดงระบบนำร่องอากาศยาน (GPS) สำหรับโปรแกรม X-Plane



The logo of Sakon Nakhon Rajabhat University is a large, faint watermark in the center of the page. It features a stylized figure of a person standing within a large, ornate letter 'S'. The 'S' is composed of multiple parallel lines, creating a sense of depth and structure. Below the 'S' is a circular emblem with a sunburst or gear-like pattern. The entire logo is rendered in a light gray color.

ภาคผนวก ข.

บทความทางวิชาการที่ได้รับการตีพิมพ์เผยแพร่ในระหว่างการศึกษา

## รายชื่อบทความที่ได้รับการตีพิมพ์เผยแพร่ในระหว่างการศึกษา

อภิสิทธิ์ หล่อกลาง และชโลธร ธรรมแท้ (2558). การพัฒนาเครื่องฝีกบินจำลองชนิดสามองศาอิสระ. การประชุมวิชาการเครือข่ายวิศวกรรมเครื่องกลแห่งประเทศไทย ครั้งที่ 29. 1-3 กรกฎาคม 2558. นครราชสีมา, ประเทศไทย. หน้า 214-219.





AME-01

การประชุมวิชาการเครือข่ายวิศวกรรมเครื่องกลแห่งประเทศไทย ครั้งที่ 29  
1-3 กรกฎาคม 2558 จังหวัดนครราชสีมา

การพัฒนาเครื่องฝึกบินจำลองชนิดสามองศาอิสระ  
A DEVELOPMENT OF 3-DOF FLIGHT SIMULATOR

อภิรักษ์ หล่อกลาง<sup>#</sup> และ ชโลธร ธรรมแท้<sup>\*</sup>

<sup>1</sup> สาขาวิชาวิศวกรรมเครื่องกล สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี  
111 ถ.มหาวิทยาลัย ต.สุรนารี อ.เมือง จ.นครราชสีมา 30000

<sup>\*</sup> E-mail chalothorn@sut.ac.th, โทร. 044-22-4410, โทรสาร 044-22-4613

<sup>#</sup> E-mail tech.sut.ac.th@gmail.com, โทร. 08-9717-1016

**บทคัดย่อ**

งานวิจัยนี้ เป็นการสร้างเครื่องต้นแบบเครื่องฝึกจำลองการบิน ที่ใช้ในการเรียนการสอนของนักศึกษาสาขาวิชาวิศวกรรมอากาศยาน มหาวิทยาลัยเทคโนโลยีสุรนารี การสร้างเครื่องฝึกบินจำลองนี้จะทำการสร้างห้องนักบิน (cockpit) การจำลองภาพการบินจะใช้โปรแกรมสำเร็จรูปในการสร้างภาพ โดยจะทำการเชื่อมต่ออุปกรณ์ควบคุมอากาศยานเข้ากับคอมพิวเตอร์ เพื่อให้การตอบสนองของภาพที่ปรากฏเหมือนกับการบินจริง อีกทั้งเครื่องมือวัดประกอบการบินต่าง ๆ จะเชื่อมต่อเข้ากับระบบควบคุม เพื่อให้การตอบสนองของเครื่องมือวัดมีความสมจริง ในด้านการเคลื่อนที่เครื่องต้นแบบนี้ จะสร้างให้ห้องนักบินสามารถเคลื่อนที่ได้ใน 3 ทิศทาง คือ การควง (roll) การเงย (pitch) และการเคลื่อนที่ตอบสนองกับความเร่งหรือแรงกระแทก (shock) โดยการทำงานจะใช้เครื่องกระตุ้นด้วยมอเตอร์ขนาดเล็กในการจำลองการเคลื่อนที่

งานวิจัยนี้จะเป็นการออกแบบระบบเชื่อมโยงเพื่อควบคุมการทำงานของส่วนหลัก ๆ คือ ส่วนการเคลื่อนที่ของห้องนักบิน ส่วนแสดงผลภาพที่มองจากห้องนักบิน ส่วนแสดงผลเครื่องมือวัดประกอบการบิน และส่วนควบคุมการเคลื่อนที่ของอากาศยานให้ทำงานร่วมกันอย่างมีประสิทธิภาพ เพื่อให้เครื่องฝึกบินจำลองนี้ให้ความรู้สึกคล้ายกับการขึ้นบินบนอากาศยาน ผลที่ได้จากงานวิจัยนี้จะทำให้วิศวกรด้านการบินและอากาศยานสามารถมีเครื่องจำลองการบินที่มีราคาถูกกว่าสร้างจากต่างประเทศ และสามารถผลิตขึ้นเองในมหาวิทยาลัย ซึ่งถือว่าการสร้างนวัตกรรมใหม่อย่างหนึ่ง

**คำหลัก:** เครื่องฝึกบินจำลอง, สามองศาอิสระ

**Abstract**

The goal of this research is to build a flight simulator prototype for using in aeronautical engineering class of Suranaree University of Technology (SUT). The flight simulator prototype including cockpit of training aircraft, image by on-the-shelf program is connected to the computer with control stick for sense the real feeling of flight. The flight instruments are also included. For getting the real feeling of flight, this prototype can move in 3 axes including roll, pitch and shock by using three servo motors for controlling the platform. Cockpit image will show in 3 multiple screens, two of which will show the look forward view, two others show the left and right views, and the last one show the instruments panel which also touch screen monitor for adjust the flight instruments. X-Plane v9 has an appropriate properties, because it can connect to the motion signal generator to control the servo motors for platform moving following the maneuver of the real flight.

**Keywords:** flight simulator, 3-DOF



## AME-01

### 1. บทนำ

ระบบจำลองการบิน (Flight Simulator) เป็นอีกหนึ่งอุปกรณ์ที่ช่วยเพิ่มศักยภาพให้กับนักบิน โดยช่วยฝึกฝนการทำการบินให้มีประสิทธิภาพเพิ่มมากขึ้นอีกทั้งยังช่วยให้ผู้ที่มีความสนใจทางด้านการบินสามารถศึกษาหาความรู้ต่าง ๆ เกี่ยวกับเครื่องบินผ่านระบบจำลองการบิน เพื่อหลีกเลี่ยงการบินจริงซึ่งเป็นเรื่องที่เสี่ยงต่ออุบัติเหตุ และสิ้นเปลืองงบประมาณเป็นจำนวนมาก โปรแกรมเกมจำลองการบินที่มีวางขายในท้องตลาดคือโปรแกรม X-Plane และ Microsoft Flight Simulator X (FSX) นั่นก็ยังสามารถทำได้เพียงการบินผ่านหน้าจอคอมพิวเตอร์ ดังแสดงในรูปที่ 1 หากต้องการให้ระบบจำลองการบินนี้เสมือนจริงมากยิ่งขึ้นจำเป็นต้องเพิ่มอุปกรณ์เชื่อมต่อ นั่นคือ ระบบการเคลื่อนที่ซึ่งจะสามารถช่วยให้ผู้ทำการบินรับรู้ถึงแรงกระทำที่มีต่ออากาศยานขณะทำการบินต่าง ๆ โดยแรงกระทำนั้นจะช่วยให้รับรู้ได้ว่าท่าทางการบินที่มีอยู่ในระบบจำลองการบิน มีความปลอดภัยมากน้อยเพียงใด[1][2]



รูปที่ 1 เกมจำลองการบิน



รูปที่ 2 เครื่องจำลองการบิน[3]

### 2. โปรแกรมจำลองการบิน

เป็นการศึกษาโปรแกรมจำลองการบินที่เหมาะสมต่อการพัฒนาไปสู่เครื่องฝึกบินจำลอง โดยโปรแกรมจำลองการบินในท้องตลาดที่นำมาศึกษามี 2 โปรแกรมด้วยกันคือ X-Plane v9 และ Microsoft Flight Simulator X (FSX) โดยโปรแกรมที่เหมาะสมต่อการสร้างเครื่องฝึกบินจำลองควรมีคุณลักษณะดังต่อไปนี้

- สามารถต่อสัญญาณภาพออกมาสูงกว่า 1 หน้าจอได้
- ส่งข้อมูลสถานะท่าทางการบินออกมาให้ผู้ใช้ได้

ตารางที่ 1 ความสามารถของโปรแกรม Flight Simulator

หัวข้อ/ โปรแกรม	X-Plane v9	Microsoft Flight Simulator X
การต่อออก หลายหน้าจอ	ต่อออกหลาย หน้าจอต้องใช้ คอมพิวเตอร์ เท่ากับจำนวน จอที่ออก	สามารถใช้ คอมพิวเตอร์เพียง ตัวเดียวในการต่อ ออกหลายหน้าจอ
การส่งสถานะ ท่าทางการบิน	ทำได้	ทำไม่ได้

ผลการศึกษาคณะคุณลักษณะดังกล่าวแสดงอยู่ในตารางที่ 1 ผลการศึกษาทำให้สรุปได้ว่าโปรแกรม X-Plane v9 มีคุณสมบัติที่เหมาะสมต่อการนำไปใช้เป็นโปรแกรมแสดงผลการจำลองการบิน

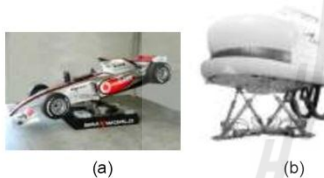
### 3. การรับ-ส่งข้อมูลท่าทางการบิน

การส่งข้อมูลสถานะท่าทางการบินแบบเรียลไทม์นั้นอาศัยการส่งผ่านข้อมูลแบบ UDP Protocol โปรโตคอล UDP ถูกพัฒนาให้เป็นโปรโตคอลสำหรับระบบงานที่ไม่ต้องการความเชื่อถือระดับสูง อาจจะถูกกล่าวได้ว่าโปรโตคอล UDP เป็นโปรโตคอลที่ง่ายที่สุดในชุดโปรโตคอล TCP/IP ทำให้ระบบที่ใช้โปรโตคอล UDP มีความเร็วในการรับส่งข้อมูลสูงเหมาะสมกับระบบงานเรียลไทม์[8]

**AME-01**

**4. การออกแบบแผ่นเคลื่อนไหว**

เมื่อก้าวถึงแผ่นเคลื่อนไหว ได้มีการประยุกต์ใช้งานอย่างเหมาะสมจนเป็นที่แพร่หลายกันมาก เช่น เครื่องจำลองการขับขี่ยานยนต์ (Driving Simulator), เครื่องฝึกหัดบิน (Flight Simulator) หรือ เครื่องเล่นในสวนสนุกต่าง ๆ ตามตัวอย่างในรูปที่ 3 และ รูปที่ 4 เป็นต้น โดยมีวัตถุประสงค์หลัก คือการสร้างสภาวะให้ผู้ขับขีได้รู้สึกเหมือนหรือใกล้เคียงกับการขับขี่ยานพาหนะจริงให้มากที่สุด โดยเฉพาะเครื่องจำลองการบิน ที่ได้นำมาฝึกทดสอบการบินเพื่อป้องกันอันตรายที่จะเกิดจากการบินจริงและยังสามารถประหยัดค่าใช้จ่ายในการบินแต่ละเที่ยวบินอีกด้วย



รูปที่ 3 เครื่องจำลอง (a.) การขับขี่ยานยนต์ (b.) ฝึกหัดบิน



รูปที่ 4 เครื่องเล่นในสวนสนุก

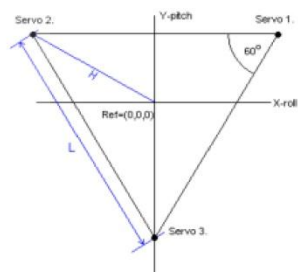
สำหรับเครื่องจำลองที่มีความสมจริงมากขึ้นนั้น ต้องเป็นเครื่องจำลองที่มีการปฏิสัมพันธ์กับผู้ขับขี และสามารถจำลองได้ทั้งภาพ, เสียงและการเคลื่อนไหว ซึ่งจะทำให้ผู้ขับขีสามารถรับรู้การเคลื่อนไหวที่สมจริงมากยิ่งขึ้น โดยเฉพาะการรับรู้ถึงแรงอันเกิดจากความเร่งในกาาระหว่างการเคลื่อนที่ของยานพาหนะ ที่มักจะทำกับกับผู้ขับขีให้รู้สึกเหมือนกับว่าได้เคลื่อนที่อยู่จริง ๆ ซึ่งในงานวิจัยนี้ ได้ทำการสร้างแผ่นเคลื่อนไหวจำลองที่สามารถสร้างสภาวะการเคลื่อนไหวได้ 3 องศาอิสระ โดยจะอยู่ในรูปของหุ่นยนต์โครงสร้างขนาน (Parallel Robot)

การออกแบบหุ่นยนต์หรือแขนกลโครงสร้างขนานในงานวิจัยนี้สามารถเคลื่อนไหวได้ 3 องศาอิสระ คือในทิศหมุนรอบแกน X-Y (Roll-Pitch) และทิศทาง

ตามแกน Z โดยพิจารณาเลือกใช้ เซอร์โวมอเตอร์ (Servo Motor) เป็นระบบขับเคลื่อนแขนกล (Actuator) ให้เคลื่อนไหวตามต้องการ ซึ่งจะมีขั้นตอนการคำนวณหาพิกัดการเคลื่อนที่ของแผ่นเคลื่อนไหว (Platform) ดังนี้

**4.1 ระบบแกนอ้างอิง**

การเปลี่ยนระบบแกนนั้นทำได้โดยการนำเอาแกนพิกัดจากมากำหนดพิกัดทั้งตำแหน่งในแนวระดับและความสูงในแนวแกนตั้ง โดยกำหนดรูปแบบและพิกัดของแท่นเคลื่อนที่[3],[6]-[7] ดังรูปที่ 5



รูปที่ 5 ระบบแกนอ้างอิง (L = ระยะห่างของ Servo)

ตำแหน่งเริ่มต้นของจุดเชื่อมต่อแผ่นเคลื่อนไหวทั้งสามกำหนดโดยระบบพิกัดข้างต้นดังสมการดังนี้

$$P_1 = \left( \frac{L}{2}, \frac{L}{2\sqrt{3}}, 0 \right) \tag{1}$$

$$P_2 = \left( -\frac{L}{2}, \frac{L}{2\sqrt{3}}, 0 \right) \tag{2}$$

$$P_3 = \left( 0, -\frac{L}{\sqrt{3}}, 0 \right) \tag{3}$$

โดยตำแหน่งทั้งสามนั้นใช้กำหนดเป็นเมทริกตำแหน่งตามแนวแกน โดยตำแหน่งการติดตั้งมอเตอร์หรือจุดส่งกำลังนั้นอาจไม่ได้ติดตั้งอยู่ในแนวที่ตรงกัน อาจมีระยะห่างกันเป็นระยะ d จากศูนย์กลาง ซึ่งในระบบแกนอ้างอิงนั้นหมายถึงระยะทางในแนวแกน Y เมทริกผลลัพธ์จากตำแหน่งเริ่มต้นทั้งหมดสามารถเขียนอยู่ในรูปของเมทริกได้ดังสมการที่ 4

$$P_{123} = \begin{bmatrix} \frac{L}{2} & -\frac{L}{2} & 0 \\ \frac{L}{2\sqrt{3}} + d & \frac{L}{2\sqrt{3}} + d & -\frac{L}{\sqrt{3}} + d \\ 0 & 0 & 0 \end{bmatrix} \tag{4}$$

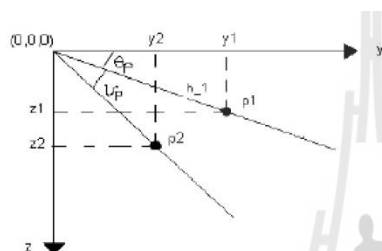


**AME-01**

แผ่นเคลื่อนไหวจะทำการหมุนรอบแกนทั้งหมดสองแกนและการเลื่อนที่อีกหนึ่งครั้ง ซึ่งการเคลื่อนที่นั้นสามารถถูกเขียนให้อยู่ในรูปของ linear transformations โดยใช้เมทริกเป็นตัวดำเนินการได้

การเคลื่อนที่ทั้งสามรูปแบบของแผ่นเคลื่อนไหวประกอบด้วย

**4.2 การเคลื่อนที่ที่ปีกเงย (Pitch motion)**



รูปที่ 6 การวิเคราะห์การเคลื่อนที่

การเคลื่อนที่ที่ปีกเงยนั้นเป็นการหมุนรอบแกน x ดังในรูปที่ 6 แกน x พุ่งตั้งฉากเข้าไปในกระดาษซึ่งแสดงด้วยจุด การหมุนจากจุด P<sub>1</sub> ที่มุม theta\_p ไปยังจุด P<sub>2</sub> ด้วยมุมที่หมุนไปเป็นมุม nu\_p

**4.3 การเคลื่อนที่รอบแกน Y (Roll motion)**

การเคลื่อนที่แบบ Roll นั้นเป็นการหมุนรอบแกน y ด้วยมุมการหมุน nu\_r

**4.4 การเคลื่อนที่ตามแกน Z (Heave motion)**

การเคลื่อนที่แบบสุดท้ายนั้นคือการเคลื่อนที่บนแกน z โดยการเคลื่อนที่นั้นสามารถเพิ่มค่าการเคลื่อนที่ได้โดยตรงที่แกน z ของทุกจุด

โดยจะได้ transformation matrix สุดท้ายดังนี้

$$P_{prz} = \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ z_1 & z_2 & z_3 \end{bmatrix} \quad (5)$$

เมื่อ

$$x_1 = -\left(\frac{\sqrt{3}L}{6} + d\right) \sin(\nu_p) \sin(\nu_r) + \frac{L}{2} \cos(\nu_r)$$

$$x_2 = -\left(\frac{\sqrt{3}L}{6} + d\right) \sin(\nu_p) \sin(\nu_r) - \frac{L}{2} \cos(\nu_r)$$

$$x_3 = -\left(-\frac{\sqrt{3}L}{6} + d\right) \sin(\nu_p) \sin(\nu_r)$$

$$y_1 = y_2 = \left(\frac{\sqrt{3}L}{6} + d\right) \cos(\nu_p)$$

$$y_3 = \left(-\frac{\sqrt{3}L}{6} + d\right) \cos(\nu_p)$$

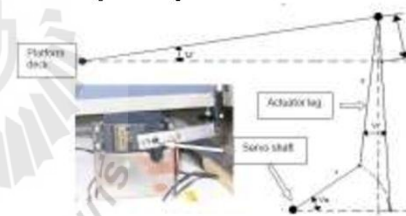
$$z_1 = \left(\frac{\sqrt{3}L}{6} + d\right) \sin(\nu_p) \cos(\nu_r) + \frac{L}{2} \sin(\nu_r) + z$$

$$z_2 = \left(\frac{\sqrt{3}L}{6} + d\right) \sin(\nu_p) \cos(\nu_r) - \frac{L}{2} \sin(\nu_r) + z$$

$$z_3 = \left(-\frac{\sqrt{3}L}{6} + d\right) \sin(\nu_p) \cos(\nu_r) + z$$

**4.5 การเตรียมข้อมูลเพื่อส่งสัญญาณไปยังมอเตอร์**

จากเมทริก P<sub>prz</sub> ในสมการที่ (5) ซึ่งให้พิกัดของจุดเชื่อมต่อของแผ่นเคลื่อนไหวกับจุดยึดมอเตอร์ทั้งสาม โดยตำแหน่งที่ได้นั้นเทียบกับระบบแกนอ้างอิงในหัวข้อข้างต้น แกนเชื่อมต่อระหว่างจุดเชื่อมต่อกับเพลลาของเซอร์โวมอเตอร์มีรูปร่างตามรูปที่ 7



รูปที่ 7 ความสัมพันธ์ระหว่างมุมของเพลลา V<sub>a</sub> และ มุมของแท่นเคลื่อนไหว nu

เพื่อที่จะหามุมการหมุนของเพลลาเซอร์โวมอเตอร์ เพื่อให้ได้การทำมุมของแท่นเคลื่อนไหวที่เป็นไปตามมุมที่ต้องการหมุนไป ในการคำนวณหาต้องการค่า Z<sub>position</sub> ของทุก ๆ จุดเชื่อมต่อต่อแผ่นเคลื่อนไหว ซึ่งหาได้จากแถวสุดท้ายของเมทริก P<sub>prz</sub> ในสมการที่ (5) ซึ่งจะทำให้สามารถหามุมการหมุนของเซอร์โวมอเตอร์แต่ละตัวได้จากความสัมพันธ์ในสมการดังนี้

$$V_a = \sin^{-1}\left(\frac{Z_{position}}{r}\right) \quad (6)$$

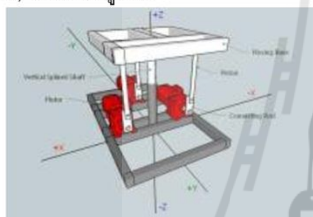


**AME-01**

จากนั้นค่ามุมที่ได้จากการคำนวณในสมการข้างต้นก็จะสามารถนำมาแปลเป็นสัญญาณ Pulse Width Modulation (PWM) เพื่อใช้ในการสั่งงานเซอร์โวมอเตอร์แต่ละตัวให้ไปในตำแหน่งที่ต้องการอยู่ในช่วง  $\pm 30^\circ$  [5]

**5. ข้อจำกัดการเคลื่อนที่**

ในการเคลื่อนที่ของแผ่นเคลื่อนไหวนี้ จะไม่สามารถเคลื่อนที่ได้ครบทุกองศาแต่จะมีมุมการเคลื่อนที่จำกัดโดยปกติแล้วขอบเขตการทำงานของแผ่นเคลื่อนไหว่นั้นจะสามารถหาค่าตอบได้เมื่อทำการสร้างและทดสอบแล้วเท่านั้น[4] และยังมีผู้ทำการบันทึกค่ามุมที่แผ่นเคลื่อนไหวสามารถทำได้ของ T1R2 Manipulator (3DOF Motion Platform) ดังแสดงในรูปที่ 8



รูปที่ 8 T1R2 Motion Platform[4]

โดยมีข้อจำกัดในการเคลื่อนที่ของแผ่นเคลื่อนไหวในสามรูปแบบการเคลื่อนไหวดังตารางที่ 2 ซึ่งแผ่นเคลื่อนไหวของงานวิจัยนี้มีค่ามุมสูงสุดในการเคลื่อนที่ของงานวิจัยนี้เป็นดังตารางที่ 3

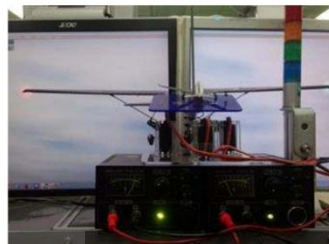
ตารางที่ 2 ข้อจำกัดการเคลื่อนที่แบบ 3 องศาอิสระ

Manipulator	Min. Value	Max. Value
Heave range	-10.0 cm	+10.0 cm
Pitch range	-30.52°	+30.52°
Roll range	-26.21°	+26.21°

ตารางที่ 3 ข้อจำกัดการเคลื่อนที่ของแผ่นเคลื่อนไหวในงานวิจัยนี้

Manipulator	ค่าต่ำสุด	ค่าสูงสุด
Heave Range	-20 mm	+20 mm
Pitch Range	-15 องศา	+15 องศา
Roll Range	-20 องศา	+20 องศา

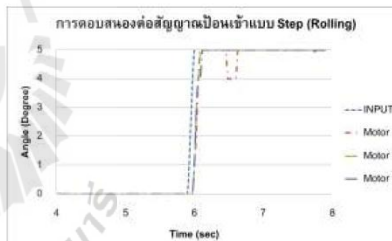
**5.1 ลักษณะการเคลื่อนที่และการตอบสนอง**



รูปที่ 9 การเคลื่อนที่แบบ Pitching



รูปที่ 10 การเคลื่อนที่แบบ Rolling



รูปที่ 9 การตอบสนองจากฟังก์ชันแบบขั้นบันได

จากกราฟการตอบสนองจากฟังก์ชันแบบขั้นบันไดพบว่า ระบบสามารถตอบสนองได้อย่างทันทีทันใด โดยยังมีค่าความคลาดเคลื่อนที่สถานะคงตัวของขั้นต่างๆ ประมาณ 1 องศา โดยจะพบการสั่นของการตอบสนองเนื่องจากการตั้งรั้งของอุปกรณ์จึงทำให้ระบบต้องใช้เวลาในการตอบสนองเพื่อให้ได้ตำแหน่งที่ต้องการตามค่าที่ป้อนเข้า จากการทดลองทั้งสามครั้งพบว่าอาจมีการตอบสนองที่ไม่พร้อมกันในช่วงเริ่มต้นของแต่ละขั้นบันได แต่สามารถเข้าสู่สถานะคงตัวได้ใกล้เคียงกัน



## AME-01

### 6. สรุป

งานวิจัยนี้เป็นการออกแบบเครื่องจำลองการบินต้นแบบ ที่สามารถเคลื่อนที่ได้ 3 องศาอิสระ (Roll, Pitch และ Heave) เพื่อศึกษาการเชื่อมโยงระหว่างโปรแกรมจำลองการบินและเครื่องจำลองการเคลื่อนที่ ซึ่งจากการศึกษาเครื่องจำลองการบินต้นแบบ ผู้วิจัยได้ทำการดึงค่าของข้อมูลท่าทางการเคลื่อนที่แบบ Roll แบบ Pitch และค่าความเร่งของเครื่องบินจากโปรแกรมจำลองการบิน แล้วนำมาแปลงเป็นการเคลื่อนที่ของเครื่องจำลองการบิน โดยผลการวิจัยเครื่องจำลองการบินที่ออกแบบสามารถเชื่อมโยงกับโปรแกรมจำลองการบินที่มีอยู่ในห้องทดลองและอุปกรณ์ควบคุมการทำงานของเครื่องจำลองการบินได้ตามจุดประสงค์ที่วางไว้ และสามารถนำไปเป็นต้นแบบเพื่อสร้างเครื่องจำลองการบินจริงได้อีกด้วย จากการทดลองพบว่าการควบคุมการเคลื่อนที่ของเครื่องจำลองการบินต้นแบบนี้สามารถทำงานได้อย่างมีประสิทธิภาพ โดยมีความคลาดเคลื่อนสูงสุดของการควบคุมมุมองศาการเคลื่อนที่ของเครื่องจำลองการบินคือ 1-2 องศา ของการควบคุมการเคลื่อนที่ทั้ง Pitch และ Roll และมีขีดจำกัดในการเคลื่อนที่แบบ Pitch คือ  $\pm 15^\circ$  การเคลื่อนที่แบบ Roll คือ  $\pm 20^\circ$  และการเคลื่อนที่ขึ้นลงในแนวแกน Z  $\pm 20$  มิลลิเมตร

### 6. ข้อเสนอแนะ

งานวิจัยนี้เป็นงานวิจัยที่ใช้ระบบควบคุมแบบเปิด ดังนั้นจึงอาจมีค่าความคลาดเคลื่อนเกิดขึ้น หากต้องการให้มีความถูกต้องเพิ่มขึ้น สามารถใช้ระบบควบคุมแบบปิด โดยทำการติดตั้งเซนเซอร์วัดท่าทางการเคลื่อนที่เพิ่มเข้าไปในระบบเพื่อป้อนสัญญาณการตรวจสอบท่าทางและทำการแก้ไข ให้ได้ท่าทางที่ถูกต้องมากยิ่งขึ้น

การสร้างเครื่องจำลองควรสำรวจอุปกรณ์ที่มีในห้องทดลองว่ามีขนาดตามที่ได้ทำการออกแบบไว้หรือไม่ เพื่อให้เครื่องจำลองการบินนั้นมีประสิทธิภาพสูงสุด

ควรนำเครื่องจำลองต้นแบบนี้ ไปพัฒนาสร้างเครื่องจำลองการบินที่สามารถบรรจุภาระกรรม (นักบินและ อุปกรณ์การบิน) ได้จริง

### 7. เอกสารอ้างอิง

- [1] กฤษดา จันทร์แดง (2551). "การจำลองพลศาสตร์ยานยนต์ร่วมกับยานพาหนะจริงแบบย่อส่วน (COLLABORATION OF VEHICLE DYNAMIC SIMULATION WITH SCALING VEHICLE)", สาขาวิชาวิศวกรรมเครื่องกล ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย.
- [2] Baarspul, M., Hosman, R. J., Van der Vaart, J. C. (1986). Some fundamentals of simulator cockpit motion generation. Advances in Flight Simulation: Visual and Motion Systems, London, Royal Aeronautical Society (RAeS).
- [3] Rekdalsbakken, W. Design and Application of a Motion Platform in Three Degrees of Freedom. SIMS 2005 46th Conference on Simulation and Modeling.
- [4] Sergio C. (2014). Toward an Extensible Simulator of Real Motion Platform. Simulation Modeling Practice and Theory, University of the West of Scotland.
- [5] Mohammad K. Design and Development of 6-DOF Motion Platform for Vehicle Driving Simulator. Universiti Teknologi Malaysia.
- [6] Chiew Y. S. (2008). Kinematic Modeling of Driving Simulator Motion Platform. Proceeding of the 2008 IEEE Conference on Innovative Technologies in Intelligent System and Industrial Applications, Multimedia University, Malaysia, 12-13 July 2008.
- [7] Stewart D.: A Platform with Six Degrees of Freedom. The Institution of Mechanical Engineers, Proceedings 1965-66, 180 Part 1, No. 15, pages 371-386.
- [8] Boyin Ding. (2011). Real-time FPGA Control of a Hexapod Robot for 6-DOF Biomechanical Testing. IECON2011 IEEE International Conference on Industrial Electronics, Melbourne, Australia, 7-10 November 2011.