

# การสร้างเครื่องฝึกบินจำลองต้นแบบ



นางสาวธวัลรัตน์ ทองปิ่น

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

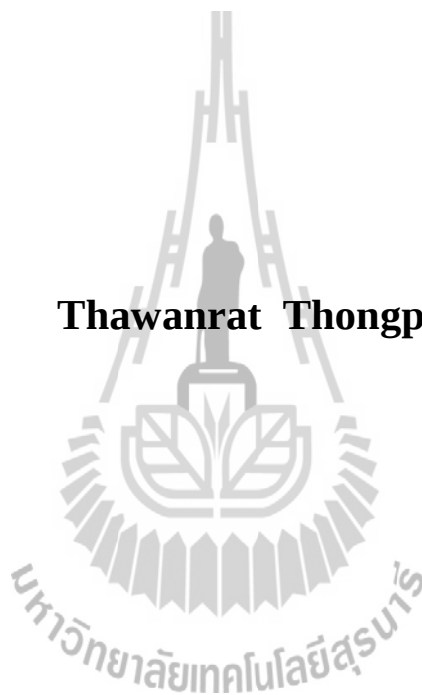
สาขาวิชาวิศวกรรมเครื่องกล

มหาวิทยาลัยเทคโนโลยีสุรนารี

ปีการศึกษา 2557

# **PROTOTYPING OF FLIGHT SIMULATOR MODULE**

**Thawanrat Thongpan**



**A Thesis Submitted in Partial Fulfillment of the Requirements for the  
Degree of Master of Engineering in Mechanical Engineering  
Suranaree University of Technology  
Academic Year 2014**

## การสร้างเครื่องฝึกบินจำลองต้นแบบ

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นักศึกษานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษา  
ตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์

(อ. ดร.ธีรุต คณวิชัย)

ประธานกรรมการ

(รศ. ร.อ. ดร.กนต์ธร ชำนิประศาสน์)

กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)

(อ. ดร.สมศักดิ์ ศิวดำรงพงศ์)

กรรมการ

(ศ.ดร.ชูกิจ ลิ้มปิงานงค์)

รองอธิการบดีฝ่ายวิชาการและนวัตกรรม

(รศ. ร.อ. ดร.กนต์ธร ชำนิประศาสน์)

คณบดีสำนักวิชาวิศวกรรมศาสตร์

ธวัชรรัตน์ ทองปิ่น : การสร้างเครื่องฝึกบินจำลองต้นแบบ (PROTOTYPING OF FLIGHT SIMULATOR MODULE) อาจารย์ที่ปรึกษา : รองศาสตราจารย์ เรืออากาศเอก ดร.กนต์ธร ชำนิประศาสน์, 76 หน้า.

งานวิจัยนี้ เป็นการสร้างเครื่องต้นแบบเครื่องฝึกจำลองการบิน ที่จะใช้ในการเรียนการสอน ของนักศึกษาสาขาวิชาวิศวกรรมอากาศยาน มหาวิทยาลัยเทคโนโลยีสุรนารี การสร้างเครื่องฝึกบิน จำลองนี้จะทำการสร้างห้องนักบิน (Cockpit) ที่เหมือนกับเครื่องบินฝึก การจำลองภาพการบินจะใช้ โปรแกรมสำเร็จรูปในการสร้างภาพ โดยจะทำการเชื่อมต่ออุปกรณ์ควบคุมอากาศยานเข้ากับ คอมพิวเตอร์ เพื่อให้การตอบสนองของภาพที่ปรากฏเหมือนกับการบินจริง อีกทั้งเครื่องมือวัด ประกอบการบินต่างๆ จะเชื่อมต่อเข้ากับระบบควบคุม เพื่อให้การตอบสนองของเครื่องมือวัดมี ความสมจริง ในด้านการเคลื่อนที่เครื่องต้นแบบนี้จะสร้างให้ห้องนักบินสามารถเคลื่อนที่ได้ใน 3 ทิศทาง คือการควง (Roll) การเงย (Pitch) และการเคลื่อนที่ตอบสนองกับความเร่งหรือแรงกระแทก (Heave) โดยการทำงานจะใช้เครื่องกระตุ้นด้วยขนาดเล็กในการจำลองการเคลื่อนที่

งานวิจัยนี้จะเป็นการออกแบบระบบเชื่อมโยงเพื่อควบคุมการทำงานของส่วนหลักๆ คือ ส่วนการเคลื่อนที่ของห้องนักบิน ส่วนแสดงผลภาพที่มองจากห้องนักบิน ส่วนแสดงผลเครื่องมือ วัดประกอบการบิน และส่วนควบคุมการเคลื่อนที่ของอากาศยานให้ทำงานร่วมกันอย่างมีประสิทธิภาพ เพื่อให้เครื่องฝึกบินจำลองนี้ให้ความรู้สึกเหมือนกับการขึ้นบินบนอากาศยานจริงให้ มากที่สุด ผลที่ได้จากงานวิจัยนี้จะทำให้วิศวกรด้านการบินและอากาศยานสามารถมีเครื่องจำลอง การบินที่มีความสมจริง ราคาถูกกว่าสร้างจากต่างประเทศ และสามารถผลิตขึ้นเองในมหาวิทยาลัย ซึ่งถือว่าเป็นการสร้างนวัตกรรมใหม่อย่างหนึ่ง

THAWANRAT THONGPAN : PROTOTYPING OF FLIGHT SIMULATOR  
MODULE. THESIS ADVISOR : ASSOC. PROF. FLT. LT. KONTORN  
CHAMNIPRASART, Ph.D., 76 PP.

FLIGHT SIMULATOR PLATFORM/MOTION 3 DOF/SERVO MOTOR  
CONTROL

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 axis including roll, pitch and heave by using three servo motor 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. Microsoft Flight Simulator (FSX) has a appropriate properties, because it can display many of views and can connect to the motion signal generator to control the servo motors for platform moving following the maneuver of the real flight.

School of Mechanical Engineering

Academic Year 2014

Student's Signature \_\_\_\_\_

Advisor's Signature \_\_\_\_\_

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี ทั้งนี้ผู้วิจัยขอขอบพระคุณบุคคลและหน่วยงานต่าง ๆ ที่ได้ให้คำปรึกษา ชี้แนะแนวทาง และให้ความช่วยเหลืออย่างดียิ่งเสมอมา ดังนี้

กองทุนวัฒนธรรมและสิ่งประดิษฐ์สมเด็จพระเทพรัตนราชสุดาฯ สยามบรมราชกุมารี ที่ได้มอบเงินทุนสนับสนุนในการทำโครงการวิจัย

มหาวิทยาลัยเทคโนโลยีสุรนารี รวมถึงบุคลากรประจำศูนย์เครื่องมือวิทยาศาสตร์และเทคโนโลยี ที่ได้ให้ความช่วยเหลือในการใช้อุปกรณ์ และเครื่องมือต่าง ๆ ในการทำวิจัยนี้

รองศาสตราจารย์ เรืออากาศเอก ดร.กนต์ธร ชำนิประศาสน์ และ อาจารย์ ดร.ชโลธร ธรรมแท้ อาจารย์ที่ปรึกษาวิทยานิพนธ์ที่ได้ถ่ายทอดความรู้ อบรมสั่งสอน ให้คำปรึกษา ชี้แนะข้อมูลต่าง ๆ ที่เป็นประโยชน์ในการทำโครงการวิจัย และแนวทางการแก้ปัญหาต่าง ๆ ด้วยความเมตตากรุณาเสมอมา

อาจารย์ ดร.ธีทัต คลวิชัย หัวหน้าสาขาวิชาวิศวกรรมเครื่องกล และคณาจารย์ทุกท่าน ที่ประสิทธิ์ประสาทวิชาความรู้ด้วยความเมตตากรุณา

คุณอภิสิทธิ์ ห่อ่นกลาง ผู้ร่วมในโครงการวิจัยนี้ ที่คอยให้ความช่วยเหลือ แนะนำแนวทาง แก้ปัญหาและให้ข้อมูลในการทำโครงการวิจัยจนสำเร็จลุล่วงด้วยดี

อาจารย์จิตติมา วระกุล และเพื่อนบัณฑิตศึกษาทุกท่าน ที่คอยช่วยเหลือและเป็นกำลังใจให้เสมอมา

ท้ายนี้ ผู้วิจัยขอกราบขอบพระคุณบิดา มารดา ที่ให้การอุปการะอบรมเลี้ยงดู ส่งเสริมด้านการศึกษา ให้การสนับสนุนและเป็นกำลังใจในการดำเนินงานวิจัยในครั้งนี้จนประสบความสำเร็จ

ธวัชรรัตน์ ทองปัน

# สารบัญ

หน้า

|                                                                                           |    |
|-------------------------------------------------------------------------------------------|----|
| บทคัดย่อ (ภาษาไทย).....                                                                   | ก  |
| บทคัดย่อ (ภาษาอังกฤษ).....                                                                | ข  |
| กิตติกรรมประกาศ.....                                                                      | ค  |
| สารบัญ .....                                                                              | ง  |
| สารบัญตาราง.....                                                                          | ฉ  |
| สารบัญรูป.....                                                                            | ช  |
| บทที่                                                                                     |    |
| 1 บทนำ.....                                                                               | 1  |
| 1.1 ความเป็นมาและความสำคัญของปัญหา.....                                                   | 1  |
| 1.2 วัตถุประสงค์การวิจัย.....                                                             | 3  |
| 1.3 ขอบเขตการวิจัย .....                                                                  | 4  |
| 1.4 สถานที่ทำงานวิจัย.....                                                                | 4  |
| 1.5 ประโยชน์ที่คาดว่าจะได้รับ .....                                                       | 4  |
| 2 ปรีทรรศน์วรรณกรรมและทฤษฎีที่เกี่ยวข้อง .....                                            | 5  |
| 2.1 กล่าวนำ .....                                                                         | 5  |
| 2.2 UDP Protocol.....                                                                     | 6  |
| 2.3 ไมโครคอนโทรลเลอร์ (Microcontroller) .....                                             | 7  |
| 3 วิธีการดำเนินการวิจัย .....                                                             | 11 |
| 3.1 กล่าวนำ .....                                                                         | 11 |
| 3.2 โปรแกรมจำลองการบิน .....                                                              | 11 |
| 3.3 การออกแบบแผ่นเคลื่อนไหว (Motion Platform) .....                                       | 14 |
| 3.4 อัลกอริทึมสำหรับประมวลผลเพื่อเชื่อมต่อข้อมูลระหว่างโปรแกรมกับแผ่น<br>เคลื่อนไหว ..... | 23 |
| 3.5 การตรวจสอบความถูกต้อง .....                                                           | 31 |
| 4 ผลการทดลอง.....                                                                         | 34 |
| 4.1 การศึกษาความสามารถของโปรแกรมตามหัวข้อที่ต้องการ .....                                 | 34 |

## สารบัญ (ต่อ)

หน้า

|     |                                                                   |    |
|-----|-------------------------------------------------------------------|----|
| 4.2 | การออกแบบและขนาดของแผ่นเคลื่อนไหว (Dimension of Platform) .....   | 35 |
| 4.3 | การทดลองเปรียบเทียบการเคลื่อนที่ของเซอร์โวมอเตอร์ทั้งสามชุด ..... | 36 |
| 4.4 | การทดลองความถูกต้องของระบบ .....                                  | 43 |
| 5   | สรุปผลและข้อเสนอแนะ .....                                         | 54 |
| 5.1 | สรุปงานวิจัย .....                                                | 54 |
| 5.2 | ข้อเสนอแนะ .....                                                  | 54 |
|     | รายการอ้างอิง .....                                               | 54 |
|     | ภาคผนวก                                                           |    |
|     | ภาคผนวก ก รายละเอียดโปรแกรมจำลองการบิน .....                      | 56 |
|     | ภาคผนวก ข รายละเอียดโปรแกรม Lab VIEW .....                        | 56 |
|     | ภาคผนวก ค บทความทางวิชาการที่ได้รับการตีพิมพ์เผยแพร่ .....        | 70 |
|     | ประวัติผู้เขียน .....                                             | 76 |

มหาวิทยาลัยเทคโนโลยีสุรนารี

## สารบัญตาราง

| ตารางที่                                                             | หน้า |
|----------------------------------------------------------------------|------|
| 3.1 ความต้องการขั้นต่ำของโปรแกรม Microsoft Flight Simulator X. ....  | 12   |
| 3.2 ความต้องการขั้นต่ำของโปรแกรม X-Plane .....                       | 14   |
| 4.1 ความสามารถของโปรแกรมจำลองการบินที่ทำการศึกษา.....                | 34   |
| 4.2 ค่าพารามิเตอร์ที่ใช้ในการออกแบบ.....                             | 35   |
| 4.3 ผลการตอบสนองของเซอร์โวมอเตอร์ตัวที่ 1 .....                      | 36   |
| 4.4 ค่าความแม่นยำของการสั่งงานเซอร์โวมอเตอร์ตัวที่ 1.....            | 37   |
| 4.5 ผลการตอบสนองของเซอร์โวมอเตอร์ตัวที่ 2.....                       | 38   |
| 4.6 ค่าความแม่นยำของการสั่งงานเซอร์โวมอเตอร์ตัวที่ 2.....            | 40   |
| 4.7 ผลการตอบสนองของเซอร์โวมอเตอร์ตัวที่ 3.....                       | 41   |
| 4.8 ค่าความแม่นยำของการสั่งงานเซอร์โวมอเตอร์ตัวที่ 3.....            | 42   |
| 4.9 ผลการตอบสนองและค่าความคลาดเคลื่อนของเซอร์โวมอเตอร์ ทั้งสาม ..... | 43   |

## สารบัญรูป

| รูปที่                                                                                      | หน้า |
|---------------------------------------------------------------------------------------------|------|
| 3.1 โปรแกรมจำลองการบิน (a.) Microsoft Flight Simulator X (FSX) (b.) X-Plane.....            | 12   |
| 3.2 เครื่องจำลอง (a.) การขับจี้รยนต์ (b.) ฝึกหัดบิน .....                                   | 15   |
| 3.3 เครื่องเล่นในสวนสนุก .....                                                              | 15   |
| 3.4 ระบบแกนอ้างอิง (L = ระยะห่างของ Servo).....                                             | 16   |
| 3.5 การหมุนรอบแกน X .....                                                                   | 17   |
| 3.6 ความสัมพันธ์ระหว่างมุมของเพลา $V_a$ และ มุมของแท่นเคลื่อนไหวน $\psi$ .....              | 21   |
| 3.7 TowerPro RC Servo Motor รุ่น MG995 .....                                                | 22   |
| 3.8 การตั้งค่าการส่งข้อมูลจากโปรแกรมจำลองการบินของเครื่อง SERVER.....                       | 23   |
| 3.9 การตั้งค่าการส่งข้อมูลจากโปรแกรมจำลองการบินของเครื่อง CLIENT .....                      | 24   |
| 3.10 ฟังก์ชัน UDP Open .....                                                                | 24   |
| 3.11 ฟังก์ชัน UDP Read.....                                                                 | 25   |
| 3.12 ฟังก์ชัน UDP Close.....                                                                | 26   |
| 3.13 โปรแกรม VI Package Manager .....                                                       | 27   |
| 3.14 ตัวอย่างโค้ด (Firmware) โปรแกรม Arduino .....                                          | 28   |
| 3.15 ฟังก์ชัน Init [LabVIEW Interface for Arduino.lvlib:Init.vi] .....                      | 28   |
| 3.16 ฟังก์ชัน Set Number of Servo.....                                                      | 29   |
| 3.17 ฟังก์ชัน Configure Servo .....                                                         | 30   |
| 3.18 ฟังก์ชัน Servo Write Angle.....                                                        | 30   |
| 3.19 ฟังก์ชัน Close.....                                                                    | 31   |
| 3.20 Flowchart การตรวจสอบความถูกต้องของเซอร์โวมอเตอร์แต่ละตัว.....                          | 32   |
| 3.21 บอร์ดไมโครคอนโทรลเลอร์ MultiWii 2012 .....                                             | 33   |
| 3.22 การทดสอบความถูกต้องของระบบ.....                                                        | 33   |
| 4.1 เครื่องจำลองการบินต้นแบบ .....                                                          | 35   |
| 4.2 กราฟแสดงความสัมพันธ์ระหว่างมุมที่กวาดไปกับมุมที่ป้อนคำสั่งของ<br>เซอร์โวมอเตอร์ 1 ..... | 37   |

## สารบัญรูป (ต่อ)

| รูปที่                                                                                                                                    | หน้า |
|-------------------------------------------------------------------------------------------------------------------------------------------|------|
| 4.3 กราฟแสดงความสัมพันธ์ระหว่างมุมที่กวาดไปกับมุมที่ป้อนคำสั่งของ.....<br>เซอร์โวมอเตอร์ 2 .....39                                        |      |
| 4.4 กราฟแสดงความสัมพันธ์ระหว่างมุมที่กวาดไปกับมุมที่ป้อนคำสั่งของ.....<br>เซอร์โวมอเตอร์ 3 .....41                                        |      |
| 4.5 กราฟแสดงผลการตอบสนองฟังก์ชันลาดเอียงของการเคลื่อนที่รอบแกน .....<br>ตามขวางในทิศทางบวก .....44                                        |      |
| 4.6 กราฟแสดงผลการตอบสนองฟังก์ชันลาดเอียงของการเคลื่อนที่รอบแกน .....<br>ตามขวางในทิศทางลบ .....45                                         |      |
| 4.7 กราฟแสดงผลการตอบสนองฟังก์ชันลาดเอียงของการเคลื่อนที่รอบแกน .....<br>ตามยาวในทิศทางบวก .....46                                         |      |
| 4.8 กราฟแสดงผลการตอบสนองฟังก์ชันลาดเอียงของการเคลื่อนที่รอบแกน .....<br>ตามยาวในทิศทางลบ.....47                                           |      |
| 4.9 กราฟแสดงผลการตอบสนองฟังก์ชันขึ้นบันไดของการเคลื่อนที่รอบแกน .....<br>ตามขวางในทิศทางบวก .....49                                       |      |
| 4.10 กราฟแสดงผลการตอบสนองฟังก์ชันขึ้นบันไดของการเคลื่อนที่รอบแกน .....<br>ตามขวางในทิศทางลบ.....50                                        |      |
| 4.11 กราฟแสดงผลการตอบสนองฟังก์ชันขึ้นบันไดของการเคลื่อนที่รอบแกน .....<br>ตามยาวในทิศทางบวก .....51                                       |      |
| 4.12 กราฟแสดงผลการตอบสนองฟังก์ชันขึ้นบันไดของการเคลื่อนที่รอบแกน .....<br>ตามยาวในทิศทางลบ.....52                                         |      |
| ก.1 โหมดการบินเดี่ยวสำหรับผู้ที่ต้องการชมทิวทัศน์สำหรับ โปรแกรม Flight .....<br>Simulator X.....57                                        |      |
| ก.2 หน้าต่างการเลือกอากาศยานที่ต้องการสำหรับ โปรแกรม Flight Simulator X .....57                                                           |      |
| ก.3 หน้าต่างกำหนดแผนการบินที่เราต้องการสำหรับ โปรแกรม Flight Simulator X.....58                                                           |      |
| ก.4 หน้าต่างสภาพอากาศจริงของพื้นที่ที่ทำการบินสำหรับ โปรแกรม Flight .....<br>Simulator X.....58                                           |      |
| ก.5 หน้าต่างกำหนดเชื้อเพลิงในการใช้งานและน้ำหนักการบรรทุก ซึ่งมีผล.....<br>ต่อระยะทางและการทรงตัวสำหรับ โปรแกรม Flight Simulator X.....59 |      |

## สารบัญรูป (ต่อ)

| รูปที่                                                                                                        | หน้า |
|---------------------------------------------------------------------------------------------------------------|------|
| ก.6 มุมมองห้องนักบิน ผู้ใช้สามารถควบคุมแผงควบคุมได้ทุกอย่างสำหรับ .....<br>โปรแกรม Flight Simulator X.....59  |      |
| ก.7 มุมมองภายนอกอากาศยานสำหรับโปรแกรม Flight Simulator X.....60                                               |      |
| ก.8 ระบบหลายหน้าต่างทำให้เราสามารถเห็นสิ่งต่างๆในเวลาเดียวกันสำหรับ.....<br>โปรแกรม Flight Simulator X.....60 |      |
| ก.9 ระบบนำร่องอากาศยาน (GPS)สำหรับโปรแกรม Flight Simulator X.....61                                           |      |
| ก.10 หมวดการเรียนรู้เกี่ยวกับเครื่องบินสำหรับโปรแกรม Flight Simulator X.....61                                |      |
| ก.11 โหมดการบินเดี่ยวสำหรับผู้ที่ต้องการชมทิวทัศน์สำหรับโปรแกรม X-Plane .....62                               |      |
| ก.12 หน้าต่างการเลือกอากาศยานที่ต้องการสำหรับโปรแกรม X-Plane .....62                                          |      |
| ก.13 หน้าต่างกำหนดแผนการบินที่เราต้องการสำหรับโปรแกรม X-Plane.....63                                          |      |
| ก.14 หน้าต่างสภาพอากาศ ณเวลาจริงของพื้นที่ที่ทำการบินสำหรับโปรแกรม .....<br>X-Plane .....63                   |      |
| ก.15 มุมมองห้องนักบิน ผู้ใช้สามารถควบคุมแผงควบคุมได้ทุกอย่างสำหรับ .....<br>โปรแกรม X-Plane .....64           |      |
| ก.16 มุมมองภายนอกอากาศยานสำหรับโปรแกรม X-Plane .....64                                                        |      |
| ก.17 ระบบนำร่องอากาศยาน (GPS) สำหรับโปรแกรม X-Plane .....65                                                   |      |
| ข.1 หน้าจอแสดงผลของโปรแกรมควบคุมการเคลื่อนที่ของ Platform.....67                                              |      |
| ข.2 Block Diagram โปรแกรมการเคลื่อนที่ของ Platform 1 .....68                                                  |      |
| ข.3 Block Diagram โปรแกรมการเคลื่อนที่ของ Platform 2 .....69                                                  |      |

## บทที่ 1

### บทนำ

#### 1.1 ความเป็นมาและความสำคัญของปัญหา

ผู้วิจัยได้เล็งเห็นถึงความสำคัญของอุตสาหกรรมการบินในประเทศไทย มาเป็นระยะเวลาหนึ่งแล้ว นับตั้งแต่การเปิดท่าอากาศยานสุวรรณภูมิขึ้นในปี พ.ศ. 2549 ทั้งภาครัฐและหน่วยงานเอกชน รัฐวิสาหกิจที่เกี่ยวข้องได้มีความพยายามที่จะให้ประเทศไทยเป็นศูนย์กลางการการบินและอากาศยานในภูมิภาคเอเชียตะวันออกเฉียงใต้ มีขีดความสามารถในการบำรุงรักษาและสามารถซ่อมอากาศยานเชิงพาณิชย์ ได้ในทุกขนาด อย่างไรก็ตามการที่ประเทศจะเป็นศูนย์กลางทางการบินได้ นอกเหนือจากการที่มีเครื่องมือซ่อมบำรุงที่ครบครัน มีอุตสาหกรรมที่รองรับการบำรุงรักษาอากาศยาน สิ่งที่เป็นที่สาคคัญคือจะต้องมีบุคลากรที่มีความรู้ความสามารถ มีความเข้าใจในการทำงานของอากาศยานและมีความรู้ความสามารถที่จะสามารถซ่อมบำรุงอากาศยานได้อย่างมีประสิทธิภาพ

โอกาสของประเทศในด้านอุตสาหกรรมการบิน ได้ปรับเปลี่ยนไปในทิศทางที่น่ายินดีอีกครึ่งหนึ่ง เมื่อการตกลงของประชาคมเศรษฐกิจอาเซียน (AEC) ได้ตกลงกันว่าเมื่อมีการรวมเศรษฐกิจของเหล่าสมาชิกแล้ว ประเทศไทยจะเป็นผู้รับผิดชอบในการเป็นศูนย์กลางด้านอากาศยานและการบินของภูมิภาค ในปี พ.ศ. 2558 ทำให้ในระยะเวลาไม่ถึง 3 ปีนับจากวันนี้ ประเทศไทยต้องรีบเตรียมการด้านบุคลากรในการด้านอากาศยานและการบิน เพื่อเตรียมรับอุตสาหกรรมการบินของทุกประเทศในภูมิภาค รวมถึงการขยายตัวเพื่อรองรับเศรษฐกิจของโลกที่กำลังเจริญเติบโตอย่างรวดเร็วในย่านภูมิภาคเอเชีย ไม่ว่าจะเป็น ประเทศ จีน อินเดีย และ เกาหลีใต้ เป็นต้น เพื่อรองรับการเจริญเติบโตของอุตสาหกรรมการบินดังกล่าว รัฐบาลจึงมีแนวทางที่จะพัฒนานิคมอุตสาหกรรมด้านอากาศยานและการบินขึ้นในประเทศ และมีแนวโน้มความเป็นไปได้สูงมากที่จะสร้างนิคมอุตสาหกรรมนี้ขึ้นในจังหวัดนครราชสีมา ซึ่งเป็นจังหวัดที่ตั้งของมหาวิทยาลัยเทคโนโลยีสุรนารี เนื่องจากปัจจุบันมีสนามบินพลเรือนที่ยังใช้งานไม่เต็มประสิทธิภาพ มีที่ตั้งอยู่ห่างจากชุมชนในระยะที่เหมาะสม ที่ดินยังมีราคาไม่สูงนัก และมีระบบคมนาคมทางบกที่สะดวก อยู่ใกล้ทั้งถนนสายหลักและทางรถไฟ ดังนั้นการสร้างนิคมอุตสาหกรรมจะมีผลกระทบน้อยทั้งในแง่การลงทุนและสิ่งแวดล้อม เปรียบเทียบกับการที่จะต้องสร้างนิคมประเภทนี้ที่สถานที่อื่น

จึงถือเป็นโอกาสดีที่มหาวิทยาลัยเทคโนโลยีสุรนารี ซึ่งมีวัตถุประสงค์หลักในการผลิตบัณฑิตด้านวิทยาศาสตร์และเทคโนโลยีสุรนารี จะได้รับผลจากการสร้างนิคมอุตสาหกรรมด้าน

อากาศยานและการบินขึ้นในจังหวัดนครราชสีมา ไม่เพียงแต่วิศวกรอากาศยานเท่านั้น แต่วิศวกรในสาขาวิชาอื่น เช่น วัสดุศาสตร์ ไฟฟ้าอิเล็กทรอนิกส์ ไฟฟ้ากำลัง ไฟฟ้าสื่อสาร วิศวกรรมเครื่องกล วิศวกรรมการผลิต วิศวกรรมอุตสาหการ วิศวกรรมคอมพิวเตอร์ และอื่นๆ ล้วนแต่จะได้ประโยชน์จากการขยายตัวนี้ทั้งสิ้น เพราะในอากาศยานแต่ละลำ ต้องการใช้วิศวกร หลายหลายประเภท ในการซ่อมบำรุง สร้างขึ้นส่วนของอากาศยานและระบบการบินของอากาศยานนั้น ดังนั้นในฐานะที่เราเป็นมหาวิทยาลัยในจังหวัดนี้ จึงต้องมองหาแนวที่จะสร้างองค์ความรู้ให้กับคณาจารย์ บุคลากร นักวิจัย เพื่อที่จะสามารถถ่ายทอดความรู้ให้กับนักศึกษาวิศวกรรมศาสตร์ ทุกสาขาวิชา

การที่จะทำให้นักศึกษามีความเข้าใจในการซ่อมบำรุงระบบต่างๆ ของอากาศยาน ขึ้นแรกจะต้องมีการสอนให้ความรู้เกี่ยวกับการทำงานของอากาศยานและส่วนประกอบของอากาศยาน ก่อนที่จะลงรายละเอียดของระบบต่างๆ ดังนั้นการทำงานของอากาศยาน และการบินของอากาศยานเป็นพื้นฐานของความรู้ในด้านนี้ที่นักศึกษาจะต้องมีความเข้าใจอย่างละเอียด ก่อนที่จะศึกษาในรายละเอียดด้านต่างๆ ต่อไป แต่การที่จะให้นักเรียนได้ทำการทดสอบบังคับอากาศยานจริงนั้น เป็นเรื่องที่เสี่ยงต่ออุบัติเหตุและสิ้นเปลืองงบประมาณเป็นจำนวนมาก ดังนั้นสถานศึกษาที่ทำการสอนด้านการบินและอากาศยานต่างๆ จึงหลีกเลี่ยงการบินจริงแต่เปลี่ยนมาใช้เครื่องจำลองการบินในการฝึกศึกษาในส่วนของพื้นฐานความรู้ให้กับนักศึกษา ทำให้การที่มหาวิทยาลัยเทคโนโลยีสุรนารีจะต้องมีเครื่องฝึกบินจำลองมีความจำเป็นอย่างมากในการสร้างบุคลากรที่มีความรู้ความสามารถในด้านนี้

อย่างไรก็ตามเครื่องฝึกบินจำลองที่มีอยู่ในประเทศไทยนั้น ไม่ได้ผลิตโดยคนไทย เป็นการนำเข้าและใช้เทคโนโลยีจากต่างประเทศทั้งสิ้น แม้ว่าในปัจจุบันนี้เทคโนโลยีที่ใช้กับเครื่องฝึกบินจำลอง ไม่ใช่เทคโนโลยีที่ใหม่เมื่อเทียบกับที่เราเริ่มมีเครื่องฝึกบินจำลองครั้งแรกในประเทศไทย แม้กระนั้นเครื่องฝึกบินจำลองที่ต้องสั่งซื้อจากต่างประเทศ กลับมีราคาสูงแม้ว่าจะเป็นเครื่องฝึกบินแบบเรียบง่ายและเป็นขั้นพื้นฐานก็ตามทั้งนี้อาจเป็นเพราะการที่มีระบบการฝึกบินจำลอง (Flight Simulator) อยู่หลายระบบ แต่การที่จะต้องเชื่อมต่อระบบการทำงานของโปรแกรมเข้ากับอากาศยานจำลอง (Aircraft Simulator) อาจต้องการเทคโนโลยีที่เฉพาะและเป็นความลับทางการค้าของแต่ละบริษัท อีกทั้งการที่จะได้ระบบห้องฝึกบินเสมือนจริง (virtual Cockpit) มีความต้องการผู้พัฒนาระบบคอมพิวเตอร์กราฟิกส์ที่ทันสมัยก็เป็นที่ผู้วิจัย จึงมีแนวความคิดที่จะสร้างเครื่องฝึกบินจำลองขึ้น เพื่อสร้างองค์ความรู้ให้กับคณาจารย์ นักวิจัยและบุคลากรของสาขาวิชา สร้างทักษะในการสร้างเครื่องจำลองการบิน สร้างองค์ความรู้ที่จะนำเทคโนโลยีต่างๆ มาประกอบใช้ร่วมกัน พร้อมทั้งนำเทคโนโลยีใหม่ๆ ที่ใช้กับอากาศยานในปัจจุบัน มาบรรจุเข้าในเครื่องฝึกบินจำลองนี้ เช่น หน้าปัดดิจิทัล ระบบควบคุมการบินด้วยไฟฟ้า (Fly by Wire) ระบบสั่งผ่านหน้าจอแบบสัมผัส (Touch Screen) เป็นต้น ซึ่งแม้ว่าเครื่องบินรุ่นใหม่ๆ จะใช้ระบบเหล่านี้

กว้างขวาง แต่ในเครื่องฝึกบินจำลองยังไม่ได้มีการนำมาใช้งานอย่างแพร่หลาย ซึ่งคณะผู้วิจัยได้เดินทางเข้าศึกษาดูงานเครื่องจำลองการบินที่มีอยู่ในประเทศไทย หลายสถานที่ด้วยกัน ทั้งของบริษัทการบินไทยจำกัด (มหาชน) สถาบันการบินพลเรือน โรงเรียนนายเรืออากาศ ซึ่งแต่ละหน่วยงานดังกล่าวมีภารกิจในการผลิตบุคลากรที่แตกต่างกัน จึงใช้เครื่องฝึกบินจำลองที่แตกต่างกัน ซึ่งคณะผู้วิจัยได้ศึกษาความเหมาะสมเบื้องต้นว่ามหาวิทยาลัยเทคโนโลยีสุรนารี ควรใช้เครื่องฝึกบินจำลองแบบใดเพื่อจะได้เหมาะสมกับการพัฒนาการเรียนการสอน และงานวิจัยของมหาวิทยาลัยเทคโนโลยีสุรนารี และเหมาะสมกับการรองรับการขยายตัวของอุตสาหกรรมการบินเข้ามาในเขตจังหวัดนครราชสีมาต่อไป

การสร้างเครื่องฝึกบินจำลองนี้จึงนับเป็นนวัตกรรมใหม่ของการเรียนการสอนด้านอากาศยาน เพราะจะเป็นการใช้เครื่องมือที่ผลิตขึ้นใช้งานเองในประเทศ โดยคณาจารย์และนักวิจัยของมหาวิทยาลัยเทคโนโลยีสุรนารีเอง ดังนั้นเมื่อได้เครื่องต้นแบบแล้ว ประโยชน์ที่ได้ ไม่เพียงแต่การที่มหาวิทยาลัยสามารถสร้างเครื่องฝึกบินจำลองต้นแบบได้เอง ซึ่งเป็นการประหยัดงบประมาณของชาติเท่านั้น สิ่งที่สำคัญกว่านั้นคือการที่คณาจารย์และบุคลากรได้ทักษะและองค์ความรู้ที่จะสามารถถ่ายทอดให้กับนักศึกษาของมหาวิทยาลัย และบุคคลที่มีความต้องการที่จะเรียนรู้เพื่อเข้าสู่ภาคอุตสาหกรรมการบินและอากาศยาน เพื่อร่วมกันสร้างความเข้มแข็งในด้านนี้ให้กับประเทศได้ต่อไป นอกเหนือจากนั้นการที่สามารถสร้างเครื่องต้นแบบได้เอง จะสามารถทำให้มหาวิทยาลัยสามารถที่จะพัฒนาต่อยอดความรู้ด้านนี้ให้มีความสมบูรณ์ เพิ่มความทันสมัยและสามารถผลิตเพื่อการค้าทั้งนั้นเพื่อรองรับการขยายตัวของความต้องการบุคลากรด้านการบินและอากาศยานที่จะเกิดขึ้นในอนาคตอันใกล้ต่อไปด้วย

## 1.2 วัตถุประสงค์การวิจัย

สำหรับงานวิจัยนี้ มีวัตถุประสงค์ของงานวิจัยในด้านการออกแบบและพัฒนาอัลกอริทึมในการเชื่อมความสัมพันธ์ระหว่างซอร์ฟแวร์และฮาร์ดแวร์ ซึ่งวัตถุประสงค์หลักของงานวิจัยมีดังต่อไปนี้

1.2.1 พัฒนาเครื่องฝึกบินจำลองต้นแบบขนาดเล็ก เพื่อหาพารามิเตอร์ที่สำคัญในการสร้างเครื่องฝึกบินจำลอง

1.2.2 สร้างนวัตกรรมในการพัฒนาเครื่องฝึกบินจำลองให้มีความเหมาะสมกับการเรียนการสอนของนักศึกษาวิศวกรรมอากาศยาน ที่มีราคาถูกว่าการสั่งซื้อจากต่างประเทศ

1.2.3 เชื่อมโยงเทคโนโลยีด้านต่าง ๆ ที่เป็นองค์ประกอบของเครื่องฝึกบินจำลอง ให้สามารถทำงานร่วมกันได้อย่างสอดคล้อง เหมาะสมและมีประสิทธิภาพ

### 1.3 ขอบเขตการวิจัย

สำหรับงานวิจัยนี้ จะทำการสร้างต้นแบบเครื่องจำลองการบินโดยมีชุด Motion ใน 3 ทิศทาง ส่วนชุด Motion ใน 6 ทิศทาง จะมีค่าใช้จ่ายสูงเกินไป จึงทำการกำหนดขอบเขตโครงการ ดังนี้

1.3.1 การแสดงผลหลักใช้ จอ LCD แสดงผล 3 จอแสดงผล

1.3.2 เครื่องมือวัดและแสดงผลในห้องนักบินใช้ระบบดิจิทัลที่แสดงผลบนจอ LCD และชุดเครื่องมือวัดประกอบการบิน

1.3.3 ติดตั้งระบบ Motion เพื่อจำลองการยกโครงสร้างห้องนักบิน ใน 3 ทิศทาง 1. หมุนรอบแกนตามยาว 2. หมุนรอบแกนตามขวาง 3. เคลื่อนที่ตามแกนแนวตั้ง (Roll, Pitch, Heave)

1.3.4 ใช้อุปกรณ์ต่าง ๆ ในห้องนักบินแบบสำเร็จรูปที่มีขายในท้องตลาดโดยอาจมีการดัดแปลงรูปลักษณะภายนอกให้สมจริง

### 1.4 สถานที่ทำงานวิจัย

มหาวิทยาลัยเทคโนโลยีสุรนารี

### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่ทางผู้วิจัยและทางบริษัทคาดว่าจะได้จากงานวิจัยนี้ มีสิ่งที่สำคัญดังต่อไปนี้

1. ได้ต้นแบบเครื่องฝึกบินจำลองที่นักศึกษาวิศวกรรมศาสตร์ สามารถใช้ในการฝึกทักษะ และความรู้พื้นฐานเกี่ยวกับการทำงานของอากาศยาน

2. สามารถพัฒนาสู่การค้าให้กับหน่วยงานที่มีวัตถุประสงค์ในการสร้างและพัฒนาบุคลากรด้านอากาศยานและการบิน

4. ได้แนวทางการพัฒนาเครื่องจักรเพื่อยกระดับจากเครื่องต้นแบบระดับห้องปฏิบัติการ ไปสู่ระบบที่สามารถใช้งานได้จริงในภาคอุตสาหกรรม

## บทที่ 2

### ปริทรรศน์วรรณกรรมและทฤษฎีที่เกี่ยวข้อง

#### 2.1 กล่าวนำ

ในส่วนแรกนี้เป็นส่วนที่ผู้วิจัยได้ทำการค้นคว้าเอกสาร รายงานการวิจัย วิทยานิพนธ์ต่างๆ ที่เกี่ยวข้องกับงานวิจัยที่ได้ดำเนินการอยู่ และจะพบว่ามียุทธศาสตร์และเอกสารต่างๆ ที่เกี่ยวข้องเป็นจำนวนมาก ดังนั้นผู้วิจัยจึงเลือกนำเสนองานวิจัยที่เกี่ยวข้องหรือสอดคล้องกับงานวิจัยที่ได้ดำเนินการอยู่นี้มาเสนอเท่านั้น

กฤษฎา จันทร์แดง .(2551) ได้ศึกษาและวิเคราะห์การจำลองพลศาสตร์ของพาหนะจริงกับพาหนะจริงกับพาหนะจริงย่อยส่วน ซึ่งเกี่ยวกับการพัฒนายานยนต์ย่อยส่วนให้มีพฤติกรรมทางพลศาสตร์สมมูลกับพาหนะจริง ยานยนต์ย่อยส่วนนี้ถูกควบคุมด้วยมนุษย์ผ่านเครื่องจำลองเสมือนจริงสามมิติ ส่วนที่สำคัญคือ การออกแบบตัวกรองเสมือนจริงสามมิติให้เสมือนจริงมากที่สุด เครื่องจำลองในงานวิจัยนี้ สามารถเคลื่อนที่ในสามแนวแกน คือ หมุนรอบแกน x หมุนรอบแกน y และเคลื่อนที่ขึ้นลงในแนวดิ่ง ตัวแปรพลศาสตร์ที่สำคัญของยานพาหนะ เช่น ความเร่งเชิงเส้น ความเร็วเชิงมุม มุมเลี้ยว และ ความเร็วล้อ เป็นต้น ถูกวัดด้วยอุปกรณ์ตรวจวัดและส่งไปยังระบบควบคุมหลัก ตามกระบวนการในตัวกรองเสมือนจริง ซึ่งถูกออกแบบมาบนพื้นฐานของระบบการรับรู้ของมนุษย์

Advani, S. K.; Nahon, M. A.; Haeck, N.; Albronda, J.(1999) ได้เสนอโปรแกรมการเพิ่มประสิทธิภาพในการกำหนดรูปแบบที่เหมาะสมของระบบการเคลื่อนไหวในหกแกน ซึ่งจะสามารถค้นหา รูปร่างของระนาบการเคลื่อนและตำแหน่งของอุปกรณ์กระตุ้น อีกทั้งยังสามารถให้ผู้ออกแบบปรับแก้ได้ตามพื้นที่การทำงานที่ต้องการ โดยผ่านการจำลอง และรับประกันว่าขาของอุปกรณ์กระตุ้นจะไม่รบกวนทางกล นอกจากนี้ยังได้เสนอตัวอย่างการประยุกต์ใช้หาพื้นที่การทำงานทั่ว ๆ ไป และพื้นที่การทำงานของการจำลองตามต้องการของเครื่องบินขนส่งขนาดใหญ่

Amery, J. G.; Streid, H.(1999) ได้เสนอแนวคิด เกี่ยวกับข้อกำหนดทางเทคนิค สำหรับระบบภาพ Out The Window (OTW) ข้อกำหนดสำหรับความแตกต่างของการฝึกบิน และ/หรือการจำลองจะถูกระบุขึ้น การแสดงผลออกทางจอแบบใหม่นี้สามารถปรับปรุงได้อย่างมีประสิทธิภาพ และคุ้มค่า

Baarspul, M.; Hosman, R. J.; Van der Vaart, J. C.(1986) ได้เสนอแนวคิดโดยเน้นพื้นที่ที่มีปัญหาในการเคลื่อนไหวของการจำลองการบิน โดยเริ่มต้นจากการจำลองการเคลื่อนไหวที่เป็นไป

ตามธรรมชาติ ซึ่งในทางวิศวกรรมข้อผิดพลาดที่เกิดจากการจำลองการเคลื่อนไหวยังจะต้องได้รับการยอมรับ กรอบคุณภาพนั้นจึงต้องมีตัวบ่งชี้ถึงการรบกวนที่จะเกิดขึ้น ตัวบ่งชี้ถึงความเชื่อมั่น ตัวบ่งชี้ถึงความผิดพลาดที่จะเกิดขึ้นตั้งแต่เริ่มการทำงาน ตัวบ่งชี้ชั่วคราวและถาวร และตัวบ่งชี้อื่น ๆ ยกตัวอย่างตัวบ่งชี้ข้อผิดพลาดชั่วคราว จากการวิจัยการเคลื่อนไหวที่พิจารณาจากผู้ใช้ที่มหาวิทยาลัยเคลฟต์ ที่สำคัญก็คือการกำหนดตัวชี้วัดที่จะเคลื่อนไหวอย่างเหมาะสมและความล่าช้าของการจำลองที่เกิดจากการเชื่อมต่อของเครื่องวัดที่ผู้ใช้อำหนด

Baughn, James; Wolf, Jason (1998) ได้อธิบายวิธีการจำลองการบินด้วย PC-base ซึ่งสามารถสอนประสิทธิภาพของอากาศยานและการทดสอบการบินสำหรับนักศึกษาวิศวกรรมการบิน ได้ โดยใช้โปรแกรม Jeppesen FS200 PCA TD และ โปรแกรม Microsoft Flight Simulator 98 ตัวอย่างของนักศึกษาที่ทำการทดสอบโปรแกรมเหล่านี้ ในการบินรักษาระดับ บินไต่ระดับ และการบินผาดแผลง ข้อมูลที่ได้จากการจำลองสภาวะการณ์ดังที่กล่าวมา มีความสอดคล้องกับทฤษฎี

สุทธิวัฒน์ ประชาชนะชัย และ สุวัฒน์ กุลธนปริดา (2548) นำเสนอการจำลองการทำงานในคอมพิวเตอร์ของแผ่นเคลื่อนไหว 6 องศาอิสระรูปแบบแกนกลโครงสร้างขนานที่ เรียกว่า Stewart Platform ในงานวิจัยโปรแกรมคอมพิวเตอร์จำลองการทำงาน ประกอบด้วยส่วนต่าง ๆ คือ ส่วน Washout Filter, สมการจลนศาสตร์ผกผัน, ตัวควบคุม, ระบบไฮดรอลิก และสมการพลวัตของแผ่นเคลื่อนไหว ผลลัพธ์การจำลองเพื่อศึกษาพฤติกรรมทางพลศาสตร์ของแผ่นเคลื่อนไหวโดยรวมเป็นที่น่าพอใจ และสามารถนำไปใช้ประโยชน์สำหรับการออกแบบเครื่องจำลองจริง เช่น เครื่องจำลองการขับรถยนต์ และเครื่องฝึกหัดนักบิน เป็นต้น โดยเฉพาะช่วยในการกำหนดขนาดชิ้นส่วนของแผ่นเคลื่อนไหว การเลือกอุปกรณ์ไฮดรอลิกที่เหมาะสม และการออกแบบตัวควบคุม

## 2.2 UDP Protocol

User Datagram Protocol หรือ UDP ที่เป็นส่วนหนึ่งของชุดโปรโตคอลอินเทอร์เน็ตที่ใช้ที่โปรแกรมทำงานบนคอมพิวเตอร์เครื่องอื่นบนเครือข่ายสามารถส่งข้อความสั้นที่เรียกว่า datagram มาอีกเครื่องหนึ่ง UDP สามารถใช้ในเครือข่ายที่ TCP (Transmission Control Protocol) จะดำเนินการแบบดั้งเดิม แต่ไม่เหมือน TCP ก็ไม่ได้รับประกันความน่าเชื่อถือหรือการเรียงลำดับที่ถูกต้องของข้อมูล datagrams อาจหายไปโดยไม่ต้องแจ้งให้ทราบล่วงหน้าหรือเข้ามาในลำดับที่แตกต่างจากที่หนึ่งที่เราถูกส่งไป โปรโตคอลที่เป็นสูตรโดย David P. Reed ในปี 1980 และกำหนดไว้อย่างเป็นทางการใน RFC 768

UDP ทำให้การใช้รูปแบบการสื่อสารที่เรียบง่ายโดยไม่ต้องตรวจสอบการส่งสำหรับการรับประกันโดยนัยความน่าเชื่อถือลำดับหรือลำดับสมบูรณ์ แม้ว่าปัจจัยเหล่านี้อาจดูเหมือนจะแนะนำ

UDP ที่ไม่ได้เป็นโปรโตคอลที่มีประโยชน์ก็ยังพบการใช้ในพื้นที่กว้างโดยเฉพาะอย่างยิ่งที่ความเร็วเกินกว่าความน่าเชื่อถือมีความสำคัญสูงสุด UDP พิจารณาที่จะตรวจสอบข้อผิดพลาดและการแก้ไขควรจะดำเนินการในการประยุกต์ใช้การสื่อสารและไม่ได้อยู่ที่เลเยอร์เครือข่ายที่ อย่างไรก็ตามหากการตรวจสอบข้อผิดพลาดและการแก้ไขมีความจำเป็นที่เลเยอร์เครือข่ายที่โปรแกรมประยุกต์สามารถใช้ประโยชน์จากการควบคุมการส่ง Protocol (TCP) หรือ ควบคุมการรับส่งกระแสข้อมูลโปรโตคอล SCTP (Stream Control Transmission Protocol) ซึ่งมีสูตรเฉพาะสำหรับเหตุผลนี้ ตั้งแต่ UDP ไม่ได้มีค่าใช้จ่ายของการตรวจสอบว่าข้อมูลที่ได้มาถึงปลายทางเวลาที่จะถูกส่งไปทุกครั้งก็มันทำให้โปรโตคอลที่เร็วและมีประสิทธิภาพมากขึ้น UDP คือมักจะใช้สำหรับการใช้งานครั้งสำคัญที่มีข้อมูลขาดหายไปเป็นที่ต้องการในการเดินทางมาถึงปลายทางข้อมูล

UDP เป็นโปรโตคอลไร้สถานะซึ่งจะเป็นประโยชน์สำหรับเซิร์ฟเวอร์มีส่วนร่วมในการตอบแบบสอบถามสั้น ๆ จากจำนวนมากของลูกค้า ในขณะที่ TCP ส่วนใหญ่จะใช้สำหรับการสื่อสารระหว่างเซิร์ฟเวอร์และไคลเอนต์เดียว, UDP จะใช้สำหรับการออกอากาศแพ็คเก็ตหรือ multicasting โดยมีการส่งข้อมูลให้กับลูกค้าทั้งหมดที่อยู่ในเครือข่าย ใช้งานเครือข่ายบ่อยครั้งที่ใช้ UDP รวม : Trivial File Transfer Protocol (TFTP), Voice over IP (VoIP), IPTV, Domain Name System ( DNS ) เป็นต้น

UDP ตั้งแต่ชนิดของการขาดกลไกในการควบคุมหรือหลีกเลี่ยงความแออัดเครือข่ายใด ๆ ในรูปแบบอื่น ๆ ของเครือข่ายที่ใช้กลไกการควบคุมจะต้องดำเนินการเพื่อให้แน่ใจว่าไหลที่ราบรื่นของการจราจรในเครือข่าย หนึ่งในโซลูชันที่ถูกออกแบบมาเพื่อจัดการปัญหานี้เป็น DCCP หรือ Datagram Protocol การควบคุมความแออัดซึ่งมีวัตถุประสงค์ที่จะตรวจสอบและการควบคุมการจราจรติดขัดในเครือข่าย UDP

## สรุป

โปรโตคอล UDP ถูกพัฒนาให้เป็นโปรโตคอลสำหรับระบบงานที่ไม่ต้องการความเชื่อถือได้ระดับสูง อาจจะกล่าวได้ว่าโปรโตคอล UDP เป็นโปรโตคอลที่ง่ายที่สุดในชุดโปรโตคอล TCP/IP ทำให้ระบบงานที่ใช้โปรโตคอล UDP มีความเร็วในการรับส่งข้อมูลสูงเหมาะสมกับระบบงานเรียลไทม์

## 2.3 ไมโครคอนโทรลเลอร์ (Microcontroller)

ไมโครคอนโทรลเลอร์ คือ อุปกรณ์ที่สามารถสร้างระบบควบคุมได้ โดยอุปกรณ์นี้มีขนาดเล็ก และเป็นอุปกรณ์ประเภทสารกึ่งตัวนำที่มีการรวมเอาฟังก์ชันการทำงานต่าง ๆ ไว้ในตัวมันเอง

ซึ่งมีลักษณะคล้ายกับคอมพิวเตอร์ ซึ่งในที่นี้หมายถึงอุปกรณ์ภายในที่ประกอบด้วย หน่วยประมวลผลกลาง, พอร์ตในการเชื่อมต่อแบบต่าง ๆ

### 2.3.1 ส่วนประกอบทั่วไปของไมโครคอนโทรลเลอร์

1. หน่วยประมวลผลกลาง (Control Processing Unit)
2. หน่วยความจำ ซึ่งประกอบด้วย RAM (Random Access Memory) และ EEPROM / EPROM / PROM / ROM (Eraseable Programable Read Only Memory)
3. หน่วยรับ และแสดงผลข้อมูล (Input / Output) ซึ่งมีพอร์ตขยายแบบขนาน (Parallel) และอนุกรม (Serial)
4. ตัวนับเวลา (Timer)
5. หน่วยควบคุมการอินเตอร์รัป (Interrupt Controller)

ส่วนประกอบเหล่านี้เป็นเพียงส่วนประกอบพื้นฐานของไมโครคอนโทรลเลอร์ ไมโครคอนโทรลเลอร์นั้นยังมีส่วนประกอบอื่นอีกเพื่อเพิ่มเติมความสามารถขึ้นอยู่กับไมโครคอนโทรลเลอร์แต่ละเบอร์ด้วย เช่น

- ADC (Analog to Digital) ส่วนภาครับสัญญาณอนาล็อกแปลงไปเป็นสัญญาณดิจิทัล
- DAC (Digital to Analog) ส่วนภาคส่งสัญญาณดิจิทัลแปลงไปเป็นสัญญาณอนาล็อก
- I2C (Inter Integrate Circuit Bus) เป็นการสื่อสารอนุกรม แบบซิงโครนัส (Synchronous) เพื่อใช้ติดต่อสื่อสาร ระหว่าง ไมโครคอนโทรลเลอร์ (MCU) กับอุปกรณ์ภายนอก ซึ่งถูกพัฒนาขึ้นโดยบริษัท Philips Semiconductors โดยใช้สายสัญญาณเพียง 2 เส้นเท่านั้นคือ serial data (SDA) และสาย serial clock (SCL) ซึ่งสามารถ เชื่อมต่ออุปกรณ์จำนวนหลายๆ ตัว เข้าด้วยกันได้ ทำให้ MCU ใช้พอร์ตเพียง 2 พอร์ตเท่านั้น
- SPI (Serial Peripheral Interface) เป็นการเชื่อมต่อกับอุปกรณ์เพื่อรับส่งข้อมูลแบบซิงโครนัส (Synchronize) มีสัญญาณนาฬิกาเข้ามาเกี่ยวข้องกับไมโครคอนโทรลเลอร์ (Microcontroller) หรือจะเป็นอุปกรณ์ภายนอกที่มีการรับส่งข้อมูลแบบ SPI อุปกรณ์ที่ทำหน้าที่เป็นมาสเตอร์ (Master) โดยปกติแล้วจะเป็นไมโครคอนโทรลเลอร์ หรืออาจกล่าวได้ว่าอุปกรณ์ Master จะต้องควบคุมอุปกรณ์ Slave ได้ โดยปกติตัว Slave มักจะเป็นไอซี (IC) หน้าที่พิเศษต่างๆ เช่น ไอซีอุณหภูมิ, ไอซีฐานเวลาฬิกาจริง (Real-Time Clock) หรืออาจเป็นไมโครคอนโทรลเลอร์ที่ทำหน้าที่ในโหมด Slave ก็ได้เช่นกัน
- PWM (Pulse Width Modulation) การสร้างสัญญาณพัลส์แบบสแควร์เวฟที่สามารถปรับเปลี่ยนความถี่และ Duty Cycle ได้เพื่อนำไปควบคุมอุปกรณ์ต่างๆเช่น มอเตอร์

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

### 2.3.2 ประเภทของไมโครคอนโทรลเลอร์

ไมโคร คอนโทรลเลอร์ มีด้วยกันหลายประเภทแบ่งตามสถาปัตยกรรม (การผลิตและกระบวนการทำงานระบบการประมวลผล) ที่มีใช้ในปัจจุบันยกตัวอย่างดังนี้

1. ไมโคร คอนโทรลเลอร์ตระกูล PIC (บริษัทผู้ผลิต Microchip ไมโครชิป)
2. ไมโคร คอนโทรลเลอร์ตระกูล MCS51 (บริษัทผู้ผลิต Atmel, Phillips)
3. ไมโคร คอนโทรลเลอร์ตระกูล AVR (บริษัทผู้ผลิต Atmel)
4. ไมโคร คอนโทรลเลอร์ Arduino (บริษัทผู้ผลิต Atmel) เป็นการนำ AVR มาเขียน firmware Bootloader คำสั่งใช้งานใหม่ ซึ่งข้อแตกต่างสำหรับบอร์ด Arduino กับบอร์ดทดลองทั่วไป คือ เป็น Open source ซึ่งหมายถึง การเปิดเผยวิธีการสร้างทั้งในส่วนของ Hardware ไม่ว่าจะเป็นวงจรต่าง ๆ ภายในบอร์ด และตัว Tool Software ที่จะป็นเครื่องมือที่จะใช้ในการเขียนคำสั่งต่าง ๆ ลงบอร์ด ล้วนเป็น Open source ทั้งหมด ส่งผลให้นักวิจัยทุกคนมีสิทธิ์ที่จะดาวน์โหลดโปรแกรมต่าง ๆ ที่เกี่ยวกับการพัฒนา Arduino ไปใช้ฟรี และสามารถสร้างวงจรเองจากต้นแบบใช้ฟรีได้เช่นกัน ซึ่งข้อดีก็คือเป็นการสร้างมาตรฐานในการพัฒนาที่จะเป็นไปในทิศทางเดียวกัน เพราะนักพัฒนาทั่วโลกนิยมใช้งานบอร์ดชนิดนี้กันมากขึ้น ดังนั้นไม่ว่าจะเป็นในเรื่องของตัวอย่าง source code ตลอดจนการพูดคุยปัญหาเรื่องการใช้งาน และ ตัวอย่างการต่อวงจรเพื่อการใช้งานในรูปแบบต่าง ๆ จะถูกพัฒนาภายใต้ Open source ของบอร์ดนี้
5. ไมโคร คอนโทรลเลอร์ตระกูล ARM7, ARM9 (บริษัทผู้ผลิต Atmel, Phillips, Analog Device, Samsung, STMicroelectronics)
6. ไมโคร คอนโทรลเลอร์ตระกูล Basic Stamp (บริษัทผู้ผลิต Parallax)
7. ไมโคร คอนโทรลเลอร์ตระกูล PSOC (บริษัทผู้ผลิต CYPRESS)
8. ไมโคร คอนโทรลเลอร์ตระกูล MSP (บริษัทผู้ผลิต Texas Instruments)
9. ไมโคร คอนโทรลเลอร์ตระกูล 68HC (บริษัทผู้ผลิต MOTOROLA)
10. ไมโคร คอนโทรลเลอร์ตระกูล H8 (บริษัทผู้ผลิต Renesas)
11. ไมโคร คอนโทรลเลอร์ตระกูล RABBIT (บริษัทผู้ผลิต RABBIT SEMICONDUCTOR)
12. ไมโคร คอนโทรลเลอร์ตระกูล Z80 (บริษัทผู้ผลิต Zilog)

### 2.3.3 ภาษาที่ใช้เขียน โปรแกรมควบคุมไมโครคอนโทรลเลอร์

1. ภาษา Assembly
2. ภาษา Basic

### 3. ภาษา C หรือ C++

### 4. ภาษา Pascal

ภาษาดัง กล่าวที่กล่าวในเบื้องต้น ไมโครคอนโทรลเลอร์บางตระกูล จะใช้ได้ครบทุกภาษา แต่บางตระกูลจะใช้ได้บางภาษา ขึ้นอยู่กับบริษัทผู้ผลิต Software (โดยทั่วไปจะเรียกว่า Editor And Compiler) ที่ใช้เขียนภาษาไมโครคอนโทรลเลอร์นั้นจะผลิตออกมาให้ Support หรือไม่



## บทที่ 3

### วิธีการดำเนินการวิจัย

#### 3.1 กล่าวนำ

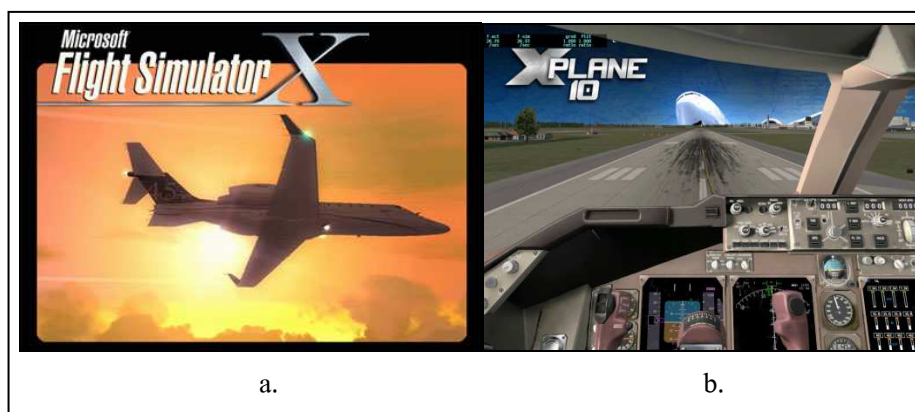
งานวิจัยนี้ได้ทำการสร้างเครื่องจำลองการเคลื่อนที่ของอากาศยานต้นแบบเพื่อนำข้อมูลที่ได้จากการศึกษาวิจัยไปพัฒนาเป็นเครื่องที่สามารถรองรับการใช้งานจริง โดยทางผู้วิจัยมีแนวคิดที่จะสร้างระบบจำลองการบินโดยประกอบด้วย การนำข้อมูลขณะทำการบินแสดงผลออกทางจอแสดงผลทั้ง 3 จอ พร้อมกับแสดงผลที่เครื่องวัดประกอบการบินแบบต่าง ๆ, การออกแบบแขนกลโครงสร้างขนานสำหรับใช้ทำหน้าที่เคลื่อนไหวเครื่องจำลองการบินต้นแบบ และ การเชื่อมโยงการสื่อสารระหว่างห้องนักบินกับแขนกลโครงสร้างขนาน

ซึ่งงานวิจัยนี้ได้มุ่งเน้นการเชื่อมโยงการสื่อสารระหว่างห้องนักบินกับแขนกลโครงสร้างขนาน ให้สามารถตอบสนองกันและกันในช่วงเวลาที่สามารถยอมรับได้ โดยจะแยกตามหัวข้อดังต่อไปนี้

#### 3.2 โปรแกรมจำลองการบิน

โปรแกรมจำลองการบินเป็นอุปกรณ์ที่นำมาจำลองสถานการณ์ที่นักบินจะต้องพบเมื่อทำการบินจริงไปบนท้องฟ้า ไม่ว่าจะเป็นสภาพอากาศที่เลวร้ายหรือจะเป็นลักษณะของสนามบินแต่ละแห่งรวมไปถึงรถและคนที่ทำหน้าที่ต่างภายในสนามบินด้วย การจำลองสถานการณ์แบบนี้จะช่วยในการฝึกให้นักบินสามารถบังคับเครื่องบินได้อย่างปลอดภัยและไปยังที่หมายที่ไม่เคยไปมาก่อนได้ดีกว่าเดิมโดยที่มีค่าใช้จ่ายที่น้อยกว่าการที่จะฝึกบินด้วยเครื่องบินจริง ในปัจจุบันมีการนำเครื่องจำลองการบินมาใช้ในการฝึกนักบินใหม่เพื่อประหยัดค่าใช้จ่ายในการฝึกบินและปลอดภัยกว่ามาก รวมไปถึงการนำเครื่องจำลองการบินมาฝึกนักบินรบเพื่อจำลองการจู่โจมเป้าหมายหรือฝึกบินเครื่องบินรบที่ทันสมัยโดยไม่ต้องทำการบินจริงเพื่อความปลอดภัยของนักบินได้อีกด้วย

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



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

### 3.2.1 Microsoft Flight Simulator X (FSX)

โปรแกรมจำลองการบินเสมือนจริง สามารถขึ้นบินจาก สนามบินหนึ่งไปลงอีกสนามบินหนึ่งทั่วโลกได้ และยังสามารถพบปะกับเพื่อนได้ด้วยการบินออนไลน์ มีสนามบินมากกว่า 1000 สนามบินจาก 223 ประเทศทั่วโลก 24 อากาศยานที่สามารถ บังคับได้ สถานที่สำคัญในโลกใบนี้ นอกจากนั้นยังได้รับการยอมรับจากโรงเรียนการบินทั่วโลก รวมทั้งประเทศไทยอีกด้วย จะเห็นได้จากที่ทหารอากาศใช้โปรแกรมนี้ ฝึกนักบินก่อนจะไปบินกับ เครื่องบินจริง

- ความต้องการขั้นต่ำของโปรแกรม Microsoft Flight Simulator X

ในการติดตั้งโปรแกรมนั้น ควรตรวจสอบเครื่องคอมพิวเตอร์ว่ามี คุณสมบัติตรงตามความต้องการขั้นต่ำของโปรแกรมเพื่อที่จะสามารถติดตั้งโปรแกรม และใช้งาน ได้อย่างเต็มประสิทธิภาพ 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® 9.0C compliant video with 32 MB of ram and support for hardware transformation and lighting |

ตารางที่ 3.1 ความต้องการขั้นต่ำของโปรแกรม Microsoft Flight Simulator X. (ต่อ)

|                                                     |                                                                                                                               |
|-----------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| ไดรฟ์ดีวีดี                                         | ความเร็ว x 32                                                                                                                 |
| เสียง                                               | การ์ดเสียง ลำโพง หรือชุดหูฟัง                                                                                                 |
| อุปกรณ์ชี้ตำแหน่ง                                   | เป็นพิมพ์ และเมาส์ หรือตัวควบคุมเกมที่เข้ากันได้ (เช่น Xbox 360 คอนโทรลเลอร์สำหรับ Windows)                                   |
| การเชื่อมต่ออินเทอร์เน็ตสำหรับการ<br>การเล่นออนไลน์ | อะแดปเตอร์เครือข่ายสำหรับ gameplay เกมส์แบบมัลติ LAN<br>อินเทอร์เน็ตแบบบรอดแบนด์สำหรับ Windows Live<br>gameplay เกมส์แบบมัลติ |

### 3.2.2 X-Plane

X-Plane เป็นโปรแกรมจำลองการบินที่มีประสิทธิภาพและครอบคลุมสำหรับคอมพิวเตอร์ส่วนบุคคล ซึ่งสามารถจำลองการบินได้อย่างมีประสิทธิภาพเหมือนจริงสำหรับการใช้ในคอมพิวเตอร์ครัวเรือน X-Plane ไม่ใช่เกมส์แต่คือนวัตกรรมทางวิศวกรรมที่ใช้ในการทำนายหรือจำลองคุณภาพการบินของอากาศยานปีกยี่ด ปีกหมุนอย่างแม่นยำ

จากคุณสมบัติที่เพียบพร้อมของโปรแกรม X-Plane นั้น เป็นเครื่องมือที่จะช่วยให้นักบินได้ลองฝึกทำการบินและแก้ไขสถานการณ์ในการบิน เนื่องจากมีความสมจริงค่อนข้างสูง สำหรับวิศวกรสามารถนำมาทำนายประสิทธิภาพการบินของอากาศยานที่ได้ทำการออกแบบได้ ซึ่งสามารถทำนายได้ในด้านกลศาสตร์การบิน

ในโปรแกรมประกอบไปด้วยอากาศยานที่ใช้ทำการบินมากมาย อาทิเช่น อากาศยานเครื่องยนต์ลูกสูบ อากาศยานเครื่องยนต์ไอพ่น เครื่องร่อน เฮลิคอปเตอร์ และอากาศยานขึ้นลงแนวตั้ง (VTOLs) ) นอกจากนั้นโปรแกรมยังสามารถรองรับการบินทั้งการบินความเร็วต่ำกว่าความเร็วเสียงในอากาศ (Subsonic) และความเร็วบินมากกว่าความเร็วเสียงในอากาศ (Supersonic) และผู้ใช้ยังสามารถเข้าถึงตัวแปรคุณสมบัติในการบินต่างๆได้อีกด้วย โดยทั้งนี้ X-Plane ยังรวบรวมอากาศยานไว้ให้ใช้ได้ทั้งหมด 35 รุ่นไว้ให้ผู้ใช้ได้ทำการบินในอากาศยานลักษณะต่างๆ เช่น Cessna 172, Bell 206 และ B-2 Bomber เป็นต้น

- ความต้องการขั้นต่ำของโปรแกรม X-Plane

เนื่องด้วยความสามารถที่มากมายของ X-Plane นั้นจึงไม่สามารถจะทำงานบนคอมพิวเตอร์รุ่นเก่าได้ โดยทั่วไปแล้วอาจประมาณได้ว่าคอมพิวเตอร์ที่สามารถติดตั้งโปรแกรมได้นั้นจะต้องเป็นคอมพิวเตอร์ที่มีคุณสมบัติตามตารางที่ 3.2

ตารางที่ 3.2 ความต้องการขั้นต่ำของโปรแกรม X-Plane

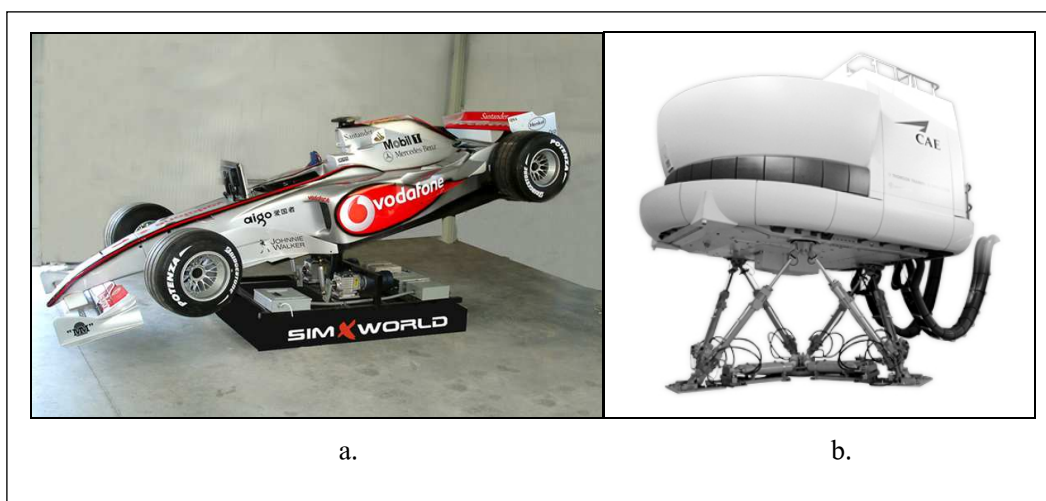
|                           |                                                                                             |
|---------------------------|---------------------------------------------------------------------------------------------|
| ระบบปฏิบัติการ            | Windows, Linux และ iOS                                                                      |
| 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

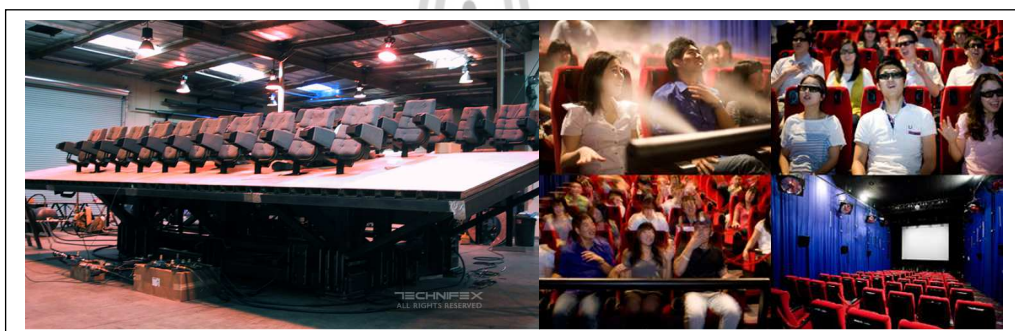
1. การต่อออกหลายหน้าจอจากโปรแกรมทั้งสอง
2. การจำหน้าตาของหน้าจอครั้งล่าสุดได้
3. การเชื่อมต่อข้อมูล เพื่อใช้ควบคุมการเคลื่อนที่ของ Platform

### 3.3 การออกแบบแผ่นเคลื่อนไหว (Motion Platform)

เมื่อก้าวถึงแผ่นเคลื่อนไหว ได้มีการประยุกต์ใช้งานอย่างเหมาะสมจนเป็นที่แพร่หลายกันมาก เช่น เครื่องจำลองการขับขีรถยนต์ (Driving Simulator), เครื่องฝึกหัดบิน (Flight Simulator) หรือ เครื่องเล่นในสวนสนุกต่าง ๆ ตามตัวอย่างในรูปที่ 3.2 และ รูปที่ 3.3 เป็นต้น โดยมีวัตถุประสงค์หลักคือการสร้างสภาวะให้ผู้ขับขี่ได้รู้สึกเหมือนหรือใกล้เคียงกับการขับขียานพาหนะจริงให้มากที่สุด โดยเฉพาะเครื่องจำลองการบิน ที่ได้นำมาฝึกทดสอบการบินเพื่อป้องกันอันตรายที่จะเกิดจากการบินจริงและยังสามารถประหยัดค่าใช้จ่ายในการบินแต่ละเที่ยวบินอีกด้วย



รูปที่ 3.2 เครื่องจำลอง (a.) การขับซิ่งรถยนต์ (b.) ฝึกหัดบิน



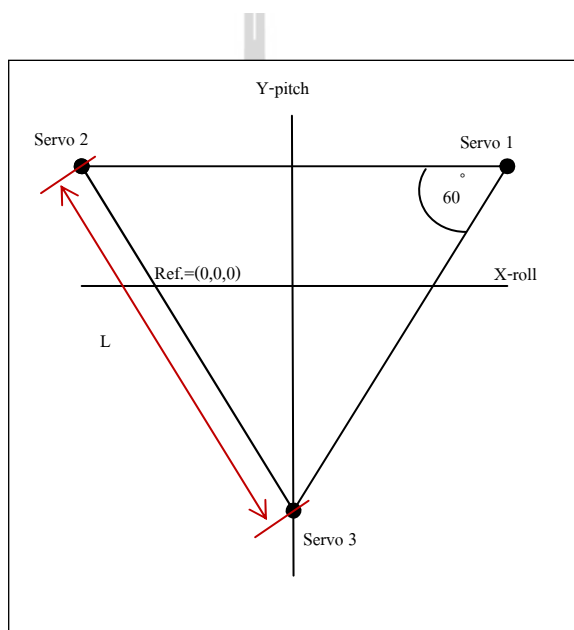
รูปที่ 3.3 เครื่องเล่นในสวนสนุก

ลักษณะการทำงานของเครื่องจำลองการเคลื่อนที่นั้นถูกแบ่งออกหลายระดับด้วยกัน เครื่องจำลองระดับที่ง่ายที่สุดเห็นจะเป็นกลุ่มเครื่องเล่นในสวนสนุก ซึ่งเครื่องจำลองนี้จะดำเนินการจำลองสถานการณ์ตามที่ได้ถูกกำหนดล่วงหน้าแล้ว โดยผู้ขับขี่เป็นเสมือนผู้โดยสารเท่านั้น ไม่สามารถมีปฏิสัมพันธ์กับเครื่องจำลองได้เลย ในส่วนของความสมจริงนั้นก็จะมีทั้งระบบภาพและเสียงเข้ามาช่วยสร้างสภาวะให้ในส่วนนี้ สำหรับเครื่องจำลองที่มีความสมจริงมากขึ้นก็จะเป็นเครื่องจำลองที่มีการปฏิสัมพันธ์กับผู้ขับขี่ และสามารถจำลองได้ทั้งภาพ, เสียงและการเคลื่อนไหว ซึ่งจะให้ผู้ขับขี่สามารถรับรู้การเคลื่อนไหวที่สมจริงมากยิ่งขึ้น โดยเฉพาะการรับรู้ถึงแรงอันเกิดจากความเร่งในการระหว่างการเคลื่อนที่ของยานพาหนะ ที่มากระทำกับผู้ขับขี่ให้รู้สึกเหมือนกับว่าได้เคลื่อนที่อย่างจริงๆ ซึ่งในงานวิจัยนี้ ได้ทำการสร้างแผ่นเคลื่อนไหวจำลอง ที่สามารถสร้างสภาวะการเคลื่อนไหวได้ 3 องศาอิสระ โดยจะอยู่ในรูปของหุ่นยนต์โครงสร้างขนาน (Parallel Robot)

การออกแบบหุ่นยนต์หรือแขนกลโครงสร้างขนานในงานวิจัยนี้สามารถเคลื่อนไหวได้ 3 องศาอิสระ คือในทิศหมุนรอบแกน X-Y (Roll-Pitch) และทิศทางตามแกน X โดยพิจารณาเลือกใช้เซอร์โวมอเตอร์ (Servo Motor) เป็นระบบขับเคลื่อนแขนกล (Actuator) ให้เคลื่อนไหวตามต้องการ ซึ่งจะมีขั้นตอนการคำนวณหาพิกัดการเคลื่อนที่ของแผ่นเคลื่อนไหว (Platform) ดังต่อไปนี้

### 3.3.1 ระบบแกนอ้างอิง (Reference system)

การเปลี่ยนระบบแกนนั้นทำได้โดยการนำเอาแกนพิกัดจากมากำหนดพิกัดทั้งตำแหน่งในแนวระดับและความสูงในแนวแกนตั้ง โดยกำหนดรูปแบบและพิกัดของแท่นเคลื่อนที่ ดังรูปที่ 3.4



รูปที่ 3.4 ระบบแกนอ้างอิง (L = ระยะห่างของ Servo)

ตำแหน่งเริ่มต้นของจุดเชื่อมต่อแผ่นเคลื่อนไหวทั้งสามกำหนดโดยระบบพิกัดข้างต้นดังสมการดังนี้

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

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

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

โดยตำแหน่งทั้งสามนั้นใช้กำหนดเป็นเมทริกตำแหน่งตามแนวแกน โดยตำแหน่งการติดตั้งมอเตอร์หรือชุดส่งกำลังนั้นอาจมิได้ติดตั้งอยู่ในแนวที่ตรงกัน อาจมีระยะห่างกันเป็นระยะ  $d$  จากศูนย์กลาง ซึ่งในระบบแกนอ้างอิงนั้นหมายถึงระยะทางในแนวแกน  $y$  เมทริกผลลัพท์จากตำแหน่งเริ่มต้นทั้งหมดสามารถเขียนอยู่ในรูปของเมทริกได้ดังสมการที่ 3.4

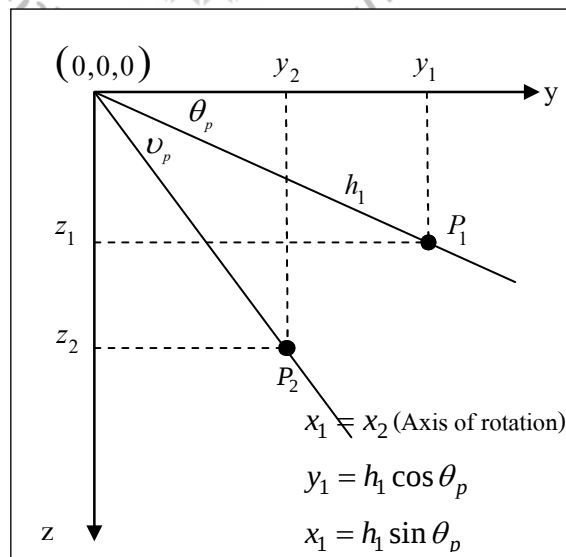
$$P_{123} = \begin{bmatrix} \frac{L}{2} & -\frac{L}{2} & 0 \\ \frac{L}{2\sqrt{3}} + d & \frac{L}{2\sqrt{3}} + d & -\frac{L}{\sqrt{3}} + d \\ 0 & 0 & 0 \end{bmatrix} \quad (3.4)$$

แผ่นเคลื่อนที่จะทำการหมุนรอบแกนทั้งหมดสองแกนในการเคลื่อนที่หนึ่งครั้ง ซึ่งการเคลื่อนที่นั้นสามารถถูกเขียนให้อยู่ในรูปของ linear transformations โดยใช้เมทริกเป็นตัวดำเนินการได้

### 3.3.2 Matrix transformation

- การเคลื่อนที่ปิกเงย (Pitch motion)

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



รูปที่ 3.5 การหมุนรอบแกน X

การหมุนนั้นสามารถเขียนเป็นสมการได้ดังนี้

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

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

พิจารณาความสัมพันธ์ระหว่างตำแหน่งเดิมและตำแหน่งใหม่จะได้

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

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

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

จะได้ transformation matrix ดังนี้

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

คูณเมทริกซ์ด้วยเมทริกซ์ตำแหน่งเริ่มต้น 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 โดยการคำนวณนั้นเป็นไปในทางเดียวกันกับการเคลื่อนที่ที่ปักธง ผลลัพธ์ของ transformation matrix คือ

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

ตำแหน่งใหม่ของจุดเชื่อมต่อหาได้จากการนำเอาเมทริกซ์ที่ได้จากการเคลื่อนที่ปักธงแล้วจะได้

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

เมื่อ

$$\begin{aligned} x_1 &= -\left(\frac{\sqrt{3}L}{6} + d\right)\sin(\nu_p)\sin(\nu_r) + \frac{L}{2}\cos(\nu_r) \\ x_2 &= -\left(\frac{\sqrt{3}L}{6} + d\right)\sin(\nu_p)\sin(\nu_r) - \frac{L}{2}\cos(\nu_r) \\ x_3 &= -\left(-\frac{\sqrt{3}L}{6} + d\right)\sin(\nu_p)\sin(\nu_r) \\ y_1 &= y_2 = \left(\frac{\sqrt{3}L}{6} + d\right)\cos(\nu_p) \\ y_3 &= \left(-\frac{\sqrt{3}L}{3} + d\right)\cos(\nu_p) \\ z_1 &= \left(\frac{\sqrt{3}L}{6} + d\right)\sin(\nu_p)\cos(\nu_r) + \frac{L}{2}\sin(\nu_r) \end{aligned}$$

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

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

• การเคลื่อนที่ตามแกน Z (Heave motion)

การเคลื่อนที่แบบสุดท้ายนั้นคือการเคลื่อนที่บนแกน z โดยการเคลื่อนที่นั้นสามารถเพิ่มค่าการเคลื่อนที่ได้โดยตรงที่แกน z ของทุกจุด โดยจะได้ transformation matrix สุดท้ายดังนี้

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

เมื่อ

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

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

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

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

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

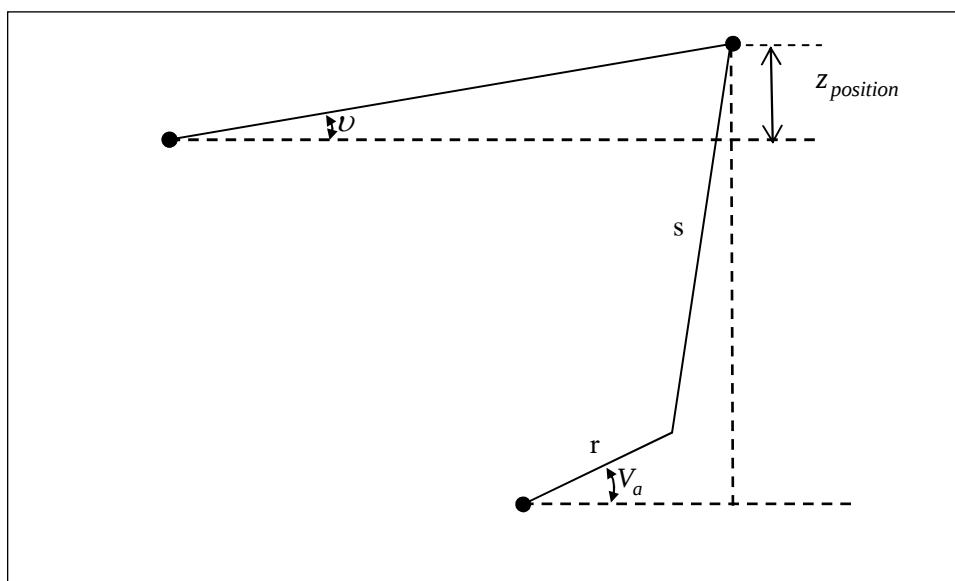
$$z_1 = \left( \frac{\sqrt{3}L}{6} + d \right) \sin(\nu_p) \cos(\nu_r) + \frac{L}{2} \sin(\nu_r) + z$$

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

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

### 3.3.3 การเตรียมข้อมูลเพื่อส่งสัญญาณไปยังอุปกรณ์ต้นกำลัง (Actuator signals)

จากเมทริก  $P_{prz}$  ซึ่งให้พิกัดของจุดเชื่อมต่อของแผ่นเคลื่อนไหวยกกับจุดยึดมอเตอร์ทั้งสาม โดยตำแหน่งที่ได้นั้นเทียบกับระบบแกนอ้างอิงในหัวข้อข้างต้น แกนเชื่อมต่อระหว่างจุดเชื่อมต่อกับเพลลาของเซอร์โวมอเตอร์มีรูปร่างตามรูปที่ 3.6



รูปที่ 3.6 ความสัมพันธ์ระหว่างมุมของเพลา  $V_a$  และ มุมของแท่นเคลื่อนไหว  $V$

เพื่อที่จะหามุมการหมุนของเพลาเซอร์โวมอเตอร์เพื่อให้ได้การทำมุมของแท่นเคลื่อนไหวที่เป็นไปตามมุมที่ต้องการหมุนไป ในการคำนวณหาต้องการค่า  $z_{position}$  ของทุก ๆ จุดเชื่อมต่อต่อแผ่นเคลื่อนไหว ซึ่งหาได้จากแถวสุดท้ายของเมทริก  $P_{prz}$  ในสมการที่ 3.15 ซึ่งจะทำได้หามุมการหมุนของเซอร์โวมอเตอร์แต่ละตัวได้จากความสัมพันธ์ในสมการดังนี้

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

จากนั้นค่ามุมที่ได้จากการคำนวณในสมการข้างต้นก็จะสามารถนำมาแปลเป็นสัญญาณ Pulse width modulation (PWM) เพื่อใช้ในการสั่งงานเซอร์โวมอเตอร์แต่ละตัวให้ไปในตำแหน่งที่ต้องการอยู่ในช่วง  $\pm 90^\circ$  ซึ่งจะกล่าวในลำดับต่อไป

### 3.3.4 อุปกรณ์ต้นกำลัง (Actuator)

ในงานวิจัยนี้เลือกใช้ Servo Motor เป็นอุปกรณ์ต้นกำลังในการเคลื่อนที่ ซึ่ง Servo Motor คือ Motor ชนิดหนึ่งซึ่งถูกออกแบบมาสำหรับใช้ในระบบการควบคุมทางความเร็วและตำแหน่ง Servo Motor ใช้การควบคุมแบบ Close Loop Control โดยที่ส่วนประกอบหลักของ Servo Motor จะประกอบด้วยส่วนสำคัญ 2 ส่วนคือ ชุด Motor ขับเคลื่อน อาจมีการใช้งานร่วมกับชุดเกียร์เพื่อเพิ่มแรงบิด และ ระบบ Feedback เช่น Potentiometer Encoder สำหรับวัดตำแหน่งมุมองศาการหมุน การควบคุม Servo Motor จะใช้สัญญาณ Pulse ที่ความถี่คงที่ แต่มีการปรับเปลี่ยนค่า %Duty cycle หรือที่เรียกว่าสัญญาณ PWM ซึ่งความกว้างของสัญญาณที่มีค่าแตกต่างกันจะทำให้ได้มุมการ

หมุนที่แตกต่างกันด้วย ในการทดลองนี้ใช้ Servo Motor ขนาดเล็กสำหรับใช้งานกับรถบังคับวิทยุ โดยเลือกใช้ RC Servo Motor ญี่ห้อ TowerPro รุ่น MG995 ดังแสดงในรูปที่ 3.7 โดยมีรายละเอียดดังนี้

มุมมองการหมุน : 180 องศา

ความเร็ว : 60 องศา ใน 0.20 วินาที ที่ 4.8 V

แรงบิด : 10 kg-cm ที่ 4.8 V

ขนาด : สูง 42.9 mm, กว้าง 19.8 mm และยาว 40.6 mm

น้ำหนัก : 55 กรัม

ชนิดของเฟือง : เหล็ก



รูปที่ 3.7 TowerPro RC Servo Motor รุ่น MG995

โดยในงานวิจัยมีการกำหนดค่าในการใช้งานดังต่อไปนี้คือ

- ความถี่ที่ใช้ในการควบคุม : 50 Hz
- ช่วงความกว้างสัญญาณพัลส์ (Pulse Width) : 1000-2000  $\mu$ s
- % Duty cycle ในช่วง 3.25 - 7.5

ในระยะเริ่มต้น ผู้วิจัยได้ทำการทดลอง ส่งสัญญาณการเปลี่ยนแปลงของมุมมองไปยัง Servo Motor โดยตรง เพื่อหาค่าความคลาดเคลื่อนที่เกิดขึ้น จากนั้นนำค่าความคลาดเคลื่อนที่ได้ไปชดเชย เพื่อให้ได้ค่ามุมมองของ Servo Motor มีค่าความคลาดเคลื่อน ให้น้อยลง

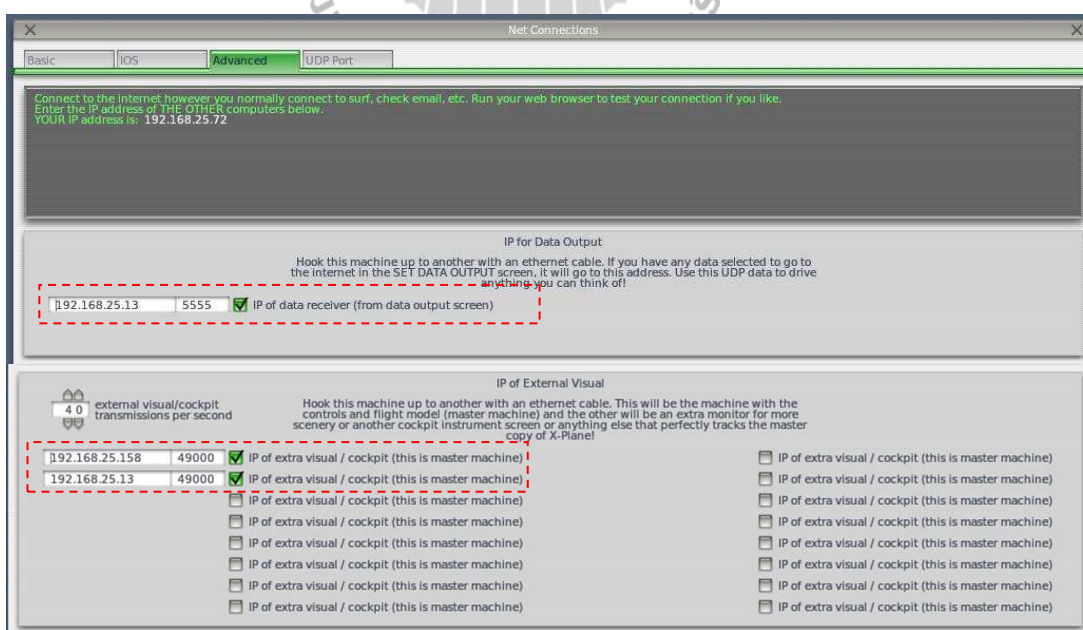
### 3.4 อัลกอริธึมสำหรับประมวลผลเพื่อเชื่อมต่อข้อมูลระหว่างโปรแกรมกับแผ่นเคลื่อนไหวน

งานวิจัยนี้ต้องการสร้างต้นแบบแผ่นเคลื่อนไหวนที่สามารถเคลื่อนไหวนได้ตามการเปลี่ยนแปลงที่เกิดขึ้น ขณะผู้ใช้งานกำลังทำการบินโดยใช้โปรแกรมจำลองการบิน ซึ่งในปัจจุบันระบบจำลองการบินมีการใช้งานอย่างหลากหลายรูปแบบ จากการศึกษาข้อมูลผู้วิจัยจึงเลือกออกแบบแผ่นเคลื่อนไหวนที่สามารถเคลื่อนที่ได้ 3 องศาอิสระ โดยใช้ซอฟต์แวร์ LabVIEW 2010 ร่วมกับ บอร์ด Microcontroller ยี่ห้อ Arduino รุ่น MEGA2560 เป็นส่วนต่อประสาน (Interface) ให้แผ่นเคลื่อนไหวนทำงานไปพร้อมกับโปรแกรมจำลองการบิน (Flight Simulator) ตามรายละเอียดต่อไปนี้

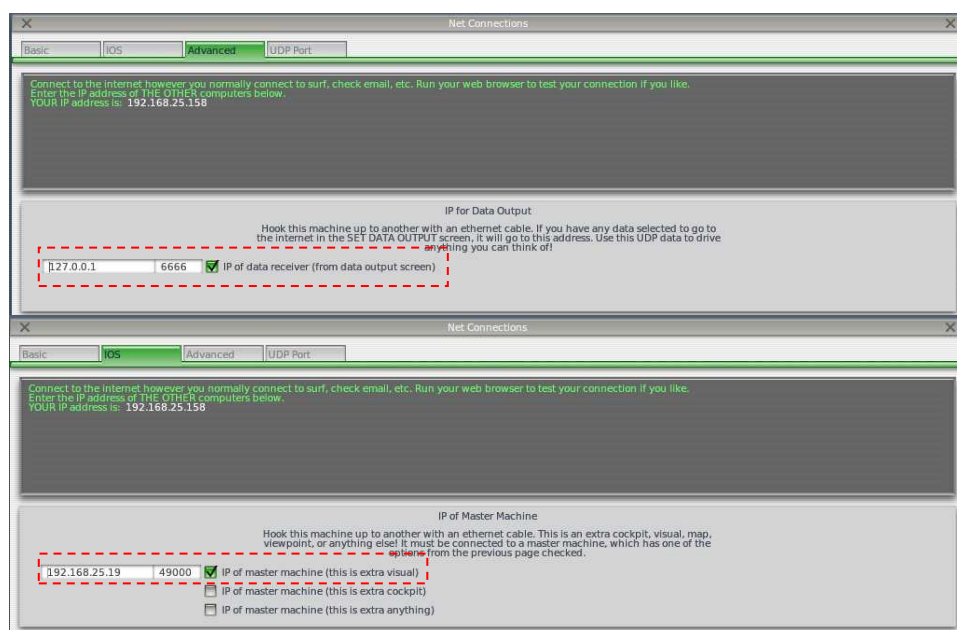
#### 3.4.1 การส่งถ่ายข้อมูลจากโปรแกรมจำลองการบิน

การส่งถ่ายข้อมูลจากโปรแกรมจำลองการบินขณะที่กำลังทำการบินเป็นสิ่งจำเป็นอย่างยิ่งในงานวิจัยนี้ เนื่องจากจะต้องอาศัยข้อมูลการบินขณะนั้น (Real time) ว่ามีลักษณะท่าทางการบิน โรล-พิช-ยอ และค่าความเร่งที่กระทำต่อเครื่องบินขณะนั้นเป็นเช่นไร ซึ่งการที่จะให้อุปกรณ์ทำการสื่อสารและส่งข้อมูลในขณะนั้นเดียวกันนั้น ผู้วิจัยจะทำการจำลองระบบขึ้นมา โดยอาศัยการสื่อสารผ่าน IP Address เดียวกัน โดยมีรายละเอียดดังต่อไปนี้

1. ตั้งค่าการส่งข้อมูลจากโปรแกรมจำลองการบิน โดย ตั้งค่าการส่ง IP Address แล้วทำการเลือกตัวแปรที่ต้องการส่งข้อมูล ซึ่งในงานวิจัย เลือกค่าที่บ่งบอกถึงท่าทางการบินในทิศทาง โรล-พิช-ยอ (Roll-Pitch-Yaw) และ ค่าความเร่งที่กระทำต่อเครื่องบิน ดังรูปที่ 3.8 และ 3.9



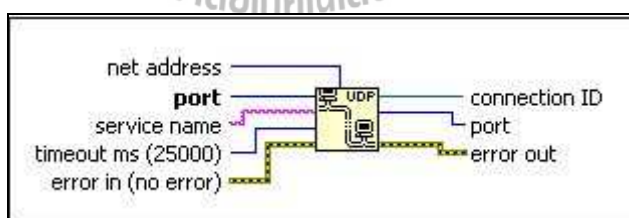
รูปที่ 3.8 การตั้งค่าการส่งข้อมูลจากโปรแกรมจำลองการบินของเครื่อง SERVER



รูปที่ 3.9 การตั้งค่าการส่งข้อมูลจากโปรแกรมจำลองการบินของเครื่อง CLIENT

2. เขียนโปรแกรมเพื่อรับค่า ที่ส่งจากโปรแกรมจำลองการบิน ซึ่งในงานวิจัยใช้โปรแกรม Lab View 2010 ในการเขียนโปรแกรมรองรับค่าท่าทางการเคลื่อนที่ดังกล่าว ตามรายละเอียดดังต่อไปนี้

เริ่มจากตั้งค่าฟังก์ชันเพื่อเปิดรับสัญญาณที่จะถูกส่งมาจากโปรแกรมจำลองการบิน โดยใช้ฟังก์ชัน UDP Open จาก LabVIEW 2010 ดังรูปที่ 3.10



รูปที่ 3.10 ฟังก์ชัน UDP Open

จากรูปที่ 3.10 สามารถกำหนดการตั้งค่าต่าง ๆ ของฟังก์ชันได้ดังนี้

- net address : กำหนด Network address ในกรณีที่เครื่องคอมพิวเตอร์ของเราสามารถเชื่อมต่อกับ Network address มากกว่าหนึ่ง แต่ในงานวิจัยนี้มีการเชื่อมเพียงแค่ Network

address เดียว จึงไม่จำเป็นต้องกำหนดค่าเพิ่มเติมเนื่องจาก ค่าเริ่มต้นของโปรแกรมจะทำการเชื่อมต่อกับ Network address ที่เชื่อมต่อ ณ ปัจจุบัน

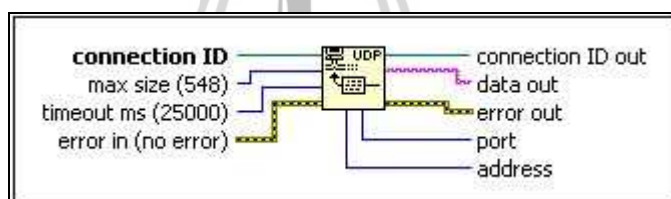
- port : กำหนดหมายเลขช่องสัญญาณการเชื่อมต่อของโปรแกรมจำลองการบินและโปรแกรม LabVIEW ตรงช่อง port ซึ่งสามารถเลือกใช้ตั้งแต่หมายเลข 5000-65535 โดยที่ต้องให้ทั้งสองโปรแกรมต้องมีหมายเลขช่องสัญญาณการเชื่อมต่อ หมายเลขเดียวกัน

- service name : กำหนดชื่อให้กับ port

- timeout ms : กำหนดเวลาเพื่อรอรับค่าที่ถูกส่งออกมาจากโปรแกรมจำลองการบิน นั่นคือ ถ้าไม่มีข้อมูลเข้ามาภายในระยะเวลาที่กำหนด ตัวฟังก์ชันจะแจ้งเตือนสิ่งที่เกิดขึ้น

- error in : การกำหนดรับค่า error จากฟังก์ชันก่อนหน้า ซึ่งในงานวิจัยนี้ใช้ฟังก์ชัน UDP Open เป็นฟังก์ชันแรกจึงไม่มีค่า error

- connection ID : ช่องทางการเชื่อมต่อข้อมูลที่ได้รับจากโปรแกรมจำลองการบินเพื่อนำไปใช้กับฟังก์ชันอื่น ๆ ต่อไป



รูปที่ 3.11 ฟังก์ชัน UDP Read

ต่อมาใช้ฟังก์ชัน UDP Read ดังรูปที่ 3.11 ในการอ่านข้อมูลที่ถูกส่งมาจากโปรแกรมจำลองการบิน ซึ่งมีรายละเอียดดังนี้

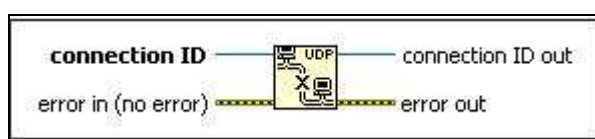
- connection ID : กำหนดช่องทางการเชื่อมต่อข้อมูล ในที่นี้ ทำการเชื่อมต่อข้อมูลจาก UDP Open เพื่ออ่านค่าที่ได้และนำไปประมวลผลต่อ

- max size : กำหนดค่าไบต์ (Bytes) สูงสุดที่จะทำการอ่าน จากการศึกษาในโปรแกรมส่งออกมาเพียง 77 ไบต์

- timeout ms : กำหนดเวลาในการรอค่าไบต์ที่ถูกส่งเข้ามาอ่าน ผู้วิจัยได้ตั้งค่าให้รอไปเรื่อย ๆ เนื่องจากการส่งข้อมูลจากโปรแกรมจำลองการบินเป็นการส่งแบบต่อเนื่อง

- error in : การกำหนดรับเงื่อนไขข้อผิดพลาดจากฟังก์ชันก่อนหน้า ในที่นี้คือ ฟังก์ชัน UDP Open

- connection ID out : ช่องทางการส่งออกการเชื่อมต่อ เพื่อไปใช้งานต่อในฟังก์ชันอื่น ๆ ซึ่งค่าที่ได้จะเป็นค่าเดียวกันกับ connection ID
- data out : เป็นข้อมูลค่าทางการบินที่ได้จากการอ่านค่าที่ส่งมาจากโปรแกรมจำลองการบิน ซึ่งอ่านค่าออกมาเป็นชุด string
- เมื่อได้ชุดข้อมูลที่ต้องการแล้วจะต้องมีตัวดำเนินการในการปิดช่องสัญญาณที่รับข้อมูลจากโปรแกรมจำลองการบิน ในโปรแกรม LabVIEW 2010 จะใช้ ฟังก์ชัน UDP Close ดังรูปที่ 3.12



รูปที่ 3.12 ฟังก์ชัน UDP Close

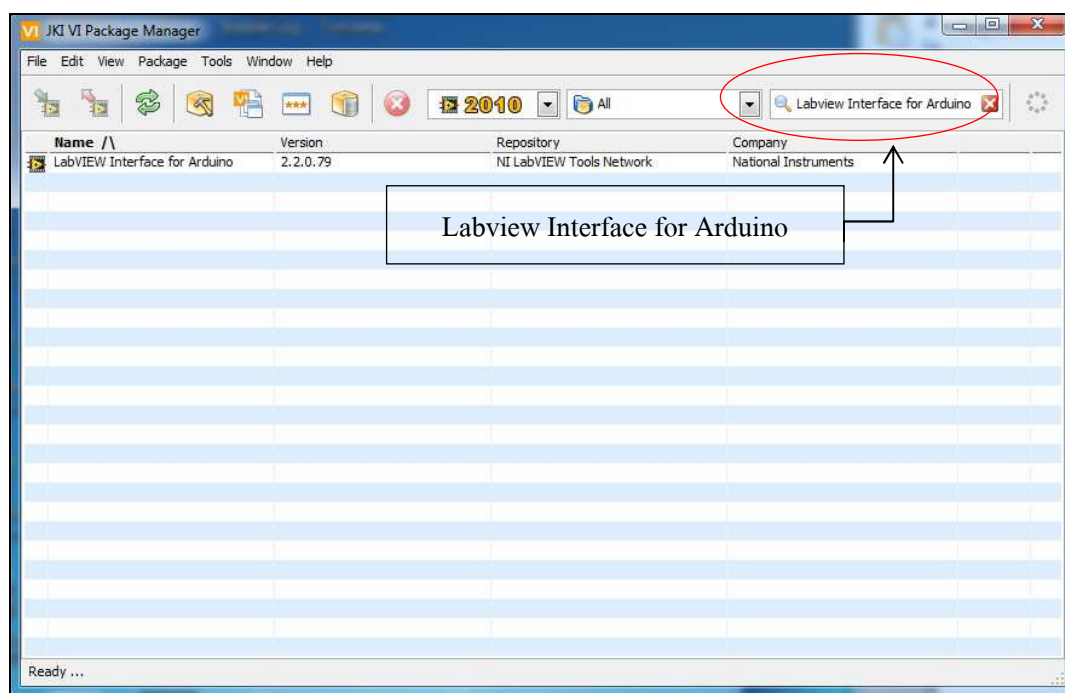
- connection ID : กำหนดช่องทางการเชื่อมต่อข้อมูลที่ต้องการปิดการเชื่อมต่อ
- error in : กำหนดรับเงื่อนไขข้อผิดพลาด จากฟังก์ชันก่อนหน้า ในที่นี้คือ ฟังก์ชัน UDP Read
- connection ID out : ช่องทางการส่งออกการเชื่อมต่อ ซึ่งค่าที่ได้จะเป็นค่าเดียวกันกับ connection ID ในกรณีของ UDP Close ห้ามนำไปใช้งานต่อ เพราะถือเป็นสถานะสุดท้าย
- error out : ส่งออกค่าข้อผิดพลาดทั้งหมดที่เกิดขึ้นขณะโปรแกรมดำเนินการ

### 3.4.2 การควบคุมการเคลื่อนที่ของแผ่นเคลื่อนไหว

จากการออกแบบลักษณะของแผ่นเคลื่อนไหวที่จะนำมาจำลองการเคลื่อนที่ของเครื่องบิน โดยใช้ Servo motor 3 ตัว เป็นต้นกำลังในการขับเคลื่อน สามารถคำนวณการเคลื่อนที่ของ Servo motor แต่ละตัวได้ตามสมการ 3.15 และ 3.16 เมื่อได้ค่าการเคลื่อนที่ของ Servo motor แล้ว ผู้วิจัยได้ทำการเขียนโปรแกรมสั่งงาน Servo motor โดยใช้ ฟังก์ชัน ของโปรแกรม LabVIEW ร่วมกับบอร์ด Microcontroller ยี่ห้อ Arduino โดยมีรายละเอียดดังนี้

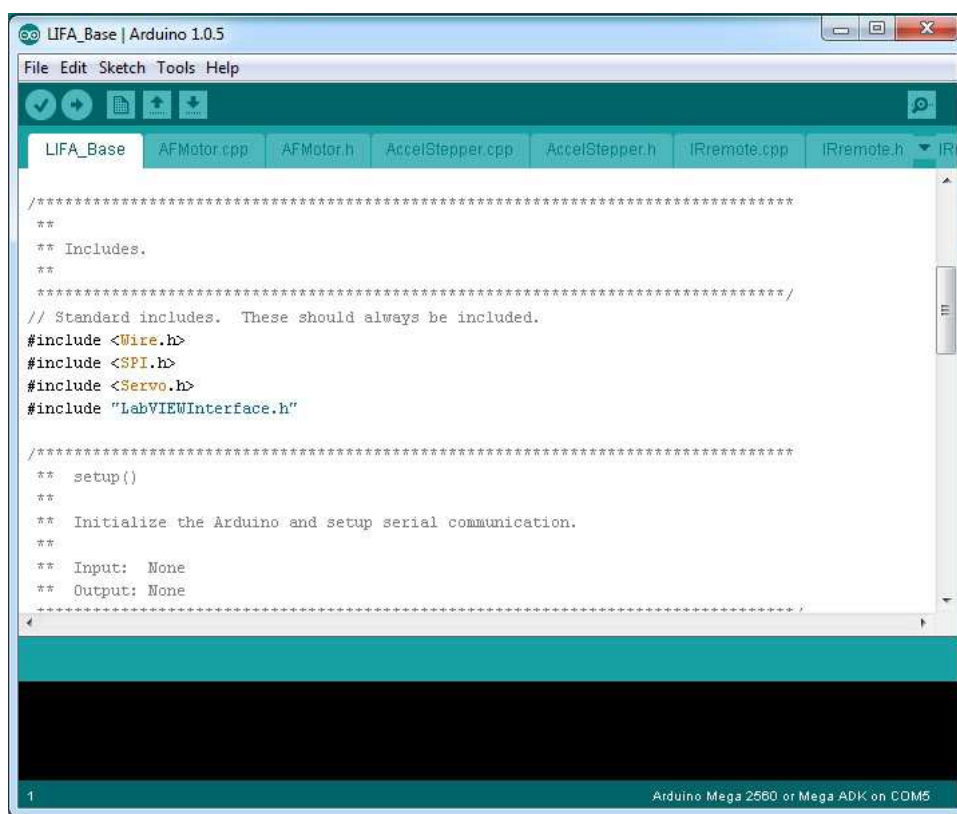
1. ติดตั้งโปรแกรม NI-VISA Drivers เพื่อให้ NI Lab VIEW มองเห็น Serial Port และติดตั้ง VI Package Manager โปรแกรมนี้ จะเป็นตัวสร้าง แชร และค้นหา LabVIEW add-ons และ toolkits จาก LabVIEW Tools Network ในที่นี้ ผู้วิจัยจะเอามาไว้สำหรับค้นหา และติดตั้ง เมื่อติดตั้ง

VI Package Manager เสร็จแล้ว ที่หน้าต่างโปรแกรม ให้ทำการค้นหา Tool kit ที่ชื่อ Labview Interface for Arduino ดังรูปที่ 3.13 เพื่อทั้งสองโปรแกรมนี้สามารถ รับส่ง ข้อมูล ซึ่งเป็น คำสั่ง และ ผลลัพธ์ผ่านทางการสื่อสาร Serial Communication ซึ่ง คอมพิวเตอร์จะมองเห็นบอร์ด Arduino เป็น เหมือน Comport อนุกรม(เสมือน) ซึ่งข้อมูลจะวิ่งเข้า ออก ณ จุดนี้ จากคอมพิวเตอร์ไป Arduino และจาก Arduino ไปคอมพิวเตอร์



รูปที่ 3.13 โปรแกรม VI Package Manager

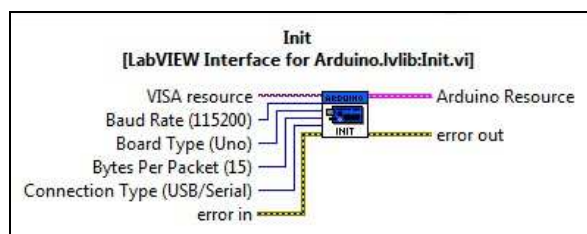
2. เขียนโค้ด (Firmware) เพื่อสั่งให้บอร์ด Arduino สามารถสื่อสารกับโปรแกรม LabVIEW ได้ โดยทำการ Compile โค้ดที่เขียนแล้วอัปโหลดเข้าไปที่ตัวบอร์ด Arduino รุ่น MEGA2560 ดังรูปที่ 3.14



รูปที่ 3.14 ตัวอย่าง โค้ด (Firmware) โปรแกรม Arduino

3. เขียนโปรแกรมประมวลผลการเคลื่อนที่ของ Servo Motor โดยใช้โปรแกรม LabVIEW 2010 เป็นสื่อกลางในการสื่อสารกับบอร์ด Arduino โดยมีรายละเอียดดังต่อไปนี้

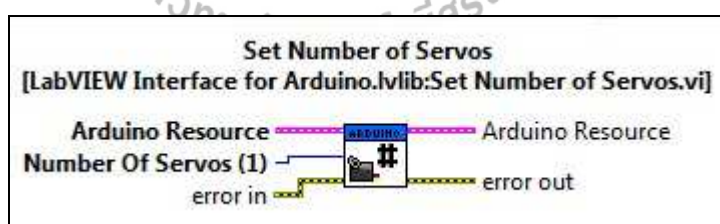
เริ่มจาก เปิดใช้งาน ฟังก์ชันที่ใช้สำหรับสื่อสารกับ บอร์ด Arduino ซึ่งได้ทำการติดตั้งมาแล้วในตอนต้น ดังที่แสดงในรูปที่ 3.15



รูปที่ 3.15 ฟังก์ชัน Init [LabVIEW Interface for Arduino.lvlib:Init.vi]

- VISA resource : กำหนดช่องทางการเชื่อมต่อ โดยกำหนดเป็นค่าของ COM PORT โดยสามารถดูเลขของ COM PORT ได้จาก device manager

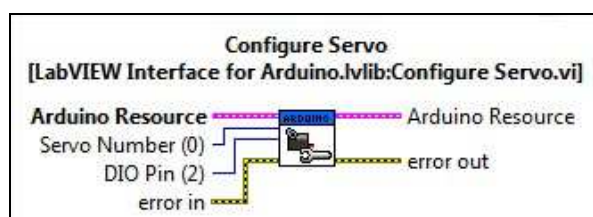
- Baud Rate : คือการกำหนดความสามารถในการส่งข้อมูลในหน่วย byte/sec โดยมีค่ากำหนดเริ่มต้นของโปรแกรมคือ 115200 byte/sec
  - Board Type : คือการกำหนดชนิดของบอร์ด Arduino ที่เลือกใช้ โดยในโปรแกรมนี้ออกเป็น Mega 2560
  - Byte Per Packet : เป็นการตั้งค่าให้ตรงกันตาม firmware ที่ได้บันทึกลงในบอร์ด โดยปกติแล้วจะมีค่าเท่ากับ 15
  - Connection Type : เป็นการกำหนดรูปแบบการเชื่อมต่อข้อมูลระหว่างบอร์ด Arduino กับคอมพิวเตอร์ โดยปกติแล้วจะกำหนดให้เป็น USB/Serial เพราะการเชื่อมต่อนั้นใช้สาย USB เป็นตัวเชื่อมต่อข้อมูล
  - error in : กำหนดรับเงื่อนไขข้อผิดพลาด จากฟังก์ชันก่อนหน้า
  - Arduino Resource : เมื่อทำการกำหนดพารามิเตอร์ต่างๆ ในการเชื่อมต่อบอร์ดแล้วเรียบร้อยแล้วจะได้ช่องทางการเชื่อมต่อออกมาผ่าน Arduino Resource นี้เพื่อนำไปใช้ในการสั่งงานหรือเรียกฟังก์ชันอื่นๆ เพื่อทำงานต่อไป
  - error out : ส่งออกค่าข้อผิดพลาดทั้งหมดที่เกิดขึ้นขณะโปรแกรมดำเนินการ
- เมื่อกำหนดการเชื่อมต่อระหว่าง Arduino กับคอมพิวเตอร์เรียบร้อยแล้ว จะต้องทำการกำหนดว่ามีความต้องการจะควบคุมการทำงานของ Servo motor จำนวนกี่ตัวผ่านฟังก์ชัน Set Number of Servo ดังรูปที่ 3.16 เพื่อเป็นการจองหน่วยพื้นที่ความจำชั่วคราวในการประมวลผลไว้ล่วงหน้า



รูปที่ 3.16 ฟังก์ชัน Set Number of Servo

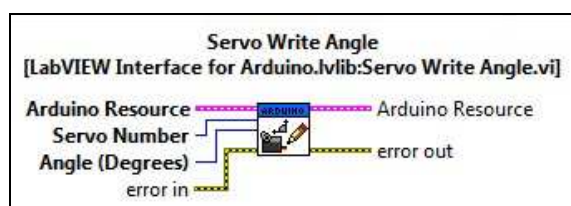
- Arduino Resource : การนำเข้าข้อมูลการเชื่อมต่อกับบอร์ดที่ได้ทำการเปิดใช้ในฟังก์ชัน Init
- Number of Servo : การกำหนดจำนวนของ Servo motor ที่จะใช้ในการควบคุมด้วยบอร์ดนี้ว่ามีจำนวนเท่าใด
- error in : กำหนดรับเงื่อนไขข้อผิดพลาด จากฟังก์ชันก่อนหน้า

- error out : ส่งออกค่าข้อผิดพลาดทั้งหมดที่เกิดขึ้นขณะโปรแกรมดำเนินการ
- หลังจากกำหนดจำนวน Servo motor ที่ต้องการควบคุมแล้วจะต้องให้ชื่อเรียกหรือเลขหมายของมอเตอร์แต่ละตัวเพื่อให้การทำงานง่ายขึ้น โดยสามารถเริ่มกำหนดได้ตั้งแต่หมายเลข 0 โดยในการกำหนดจะต้องกำหนดเลขหมายและตำแหน่งของสายสัญญาณที่ถูกเชื่อมต่อว่าอยู่ใน Digital IO Pin ที่เท่าใดอีกด้วย ดังในรูป 3.17



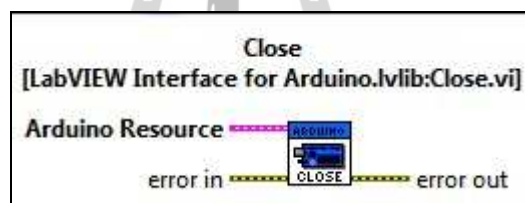
รูปที่ 3.17 ฟังก์ชัน Configure Servo

- Arduino Resource : การนำเข้าข้อมูลการเชื่อมต่อกับบอร์ดที่ได้ทำการเปิดใช้ในฟังก์ชัน Init
  - Servo Number : เป็นการกำหนดเลขหมายของ Servo motor ที่เชื่อมต่ออยู่ที่ช่อง Digital ที่กำหนดใน DIO Pin
  - DIO Pin : กำหนดหมายเลขของ Pin ที่ต้องการกำหนดหมายเลขตาม Servo Number
  - error in : กำหนดรับเงื่อนไขข้อผิดพลาด จากฟังก์ชันก่อนหน้านี้
  - error out : ส่งออกค่าข้อผิดพลาดทั้งหมดที่เกิดขึ้นขณะโปรแกรมดำเนินการ
- เมื่อกำหนดเลขหมายแล้วเรียบรื้อว่ามอเตอร์ตัวใดถูกต่ออยู่ที่ Pin ใด ก็จะสามารทำงานให้มอเตอร์หมุนได้โดยผ่านฟังก์ชัน Servo Write Angle ดังรูปที่ 3.18 โดยจะต้องกำหนดมุมที่ต้องการจะหมุนด้วยว่าต้องการหมุนไปที่มุมเท่าใด โดยมีค่าระหว่าง 0 – 180 องศา และบอกหมายเลขมอเตอร์ที่ต้องการจะสั่งงาน



รูปที่ 3.18 ฟังก์ชัน Servo Write Angle

- Arduino Resource : การนำเข้าข้อมูลการเชื่อมต่อกับบอร์ดที่ได้ทำการเปิดใช้ในฟังก์ชัน Init
- Servo Number : กำหนดหมายเลขของ Servo motor ที่ได้ทำการกำหนดช่อง DIO ไว้แล้ว เพื่อที่จะสั่งงานให้หมุนไปยังตำแหน่งที่กำหนด
- DIO Pin : กำหนดหมายเลขของ Pin ที่ต้องการกำหนดหมายเลขตาม Servo Number
- error in : กำหนดรับเงื่อนไขข้อผิดพลาด จากฟังก์ชันก่อนหน้า
- error out : ส่งออกค่าข้อผิดพลาดทั้งหมดที่เกิดขึ้นขณะโปรแกรมดำเนินการ เมื่อทำการสั่งงานหรือกระทำการใด ๆ ในโปรแกรมเสร็จสิ้นแล้วจะต้องไม่ลืมการปิดช่องทางการเชื่อมต่อ เนื่องจากหากเราไม่ปิดช่องทางการเชื่อมต่อในการเชื่อมต่อครั้งถัดไปอาจไม่สามารถใช้งานโปรแกรมได้ เนื่องจากการทำงานของช่องทางที่เราเคยใช้นั้นยังคงถูกใช้งานอยู่ตลอดเวลา ดังนั้นเพื่อการทำงานที่สะดวกและรวดเร็วในแต่ละครั้งที่มีการรันโปรแกรมจะต้องนำฟังก์ชัน Close ดังรูปที่ 3.19 มาปิดช่องทางการเชื่อมต่อเสมอ



รูปที่ 3.19 ฟังก์ชัน Close

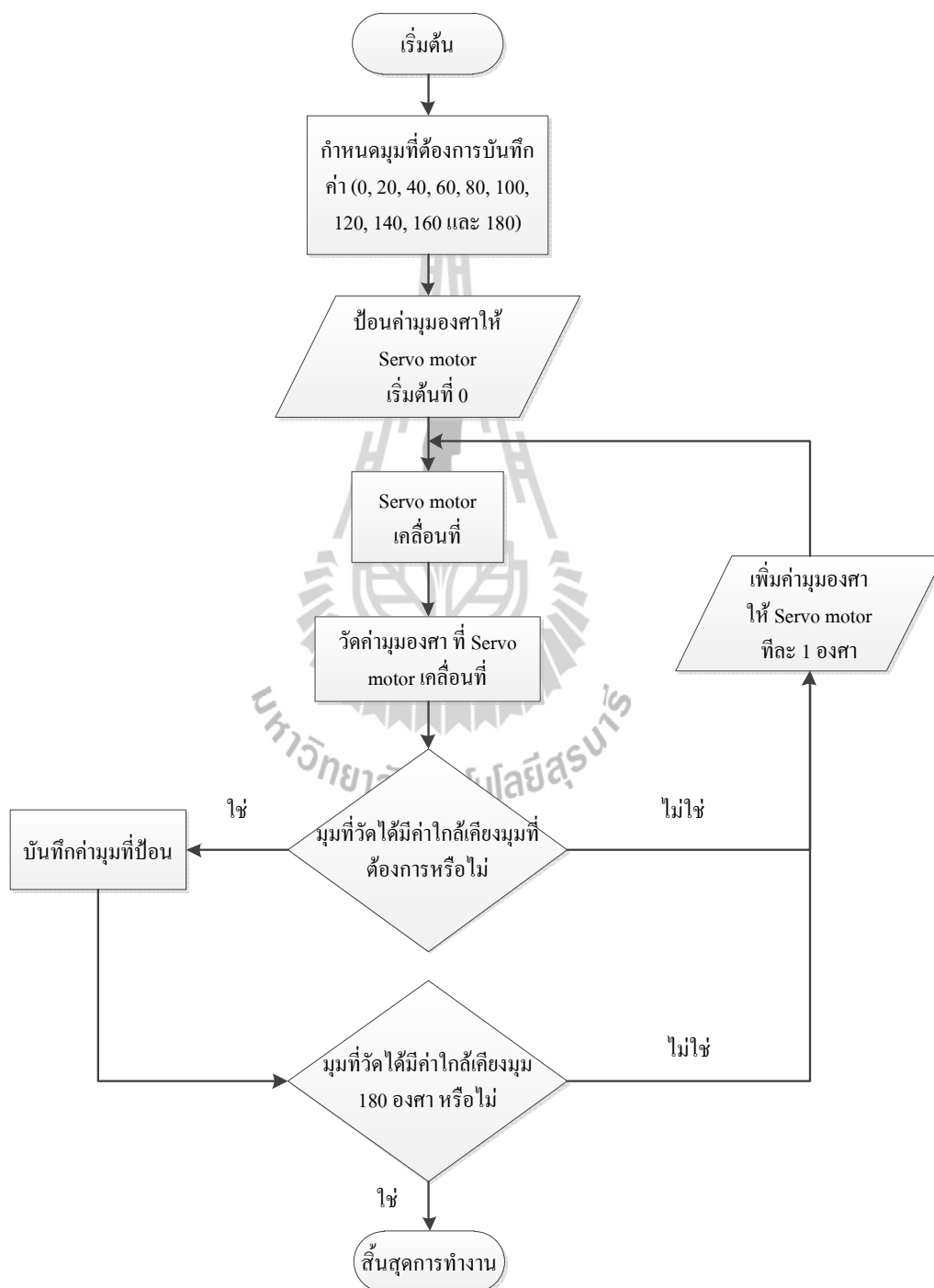
- Arduino Resource : การนำเข้าข้อมูลการเชื่อมต่อกับบอร์ดที่ได้ทำการเปิดใช้ในฟังก์ชัน Init หรือเป็นข้อมูลส่งออกจากฟังก์ชันต่าง ๆ ของ Arduino
- error in : : กำหนดรับเงื่อนไขข้อผิดพลาด จากฟังก์ชันก่อนหน้า
- error out : ส่งออกค่าข้อผิดพลาดทั้งหมดที่เกิดขึ้นขณะโปรแกรมดำเนินการ

### 3.5 การตรวจสอบความถูกต้อง

งานวิจัยจะเชื่อถือได้มากน้อยเพียงใดขึ้นอยู่กับเครื่องมือที่ใช้ในการรวบรวมข้อมูลถ้าเครื่องมือไม่มีความเที่ยงตรง เชื่อถือไม่ได้ งานวิจัยก็เชื่อถือไม่ได้เช่นกัน ดังนั้น ในงานวิจัยนี้ ได้ทำการออกแบบ การตรวจสอบความถูกต้อง เป็น 2 ส่วนหลัก ๆ ดังต่อไปนี้

### 3.5.1 ความถูกต้องของ Servo Motor แต่ละตัว

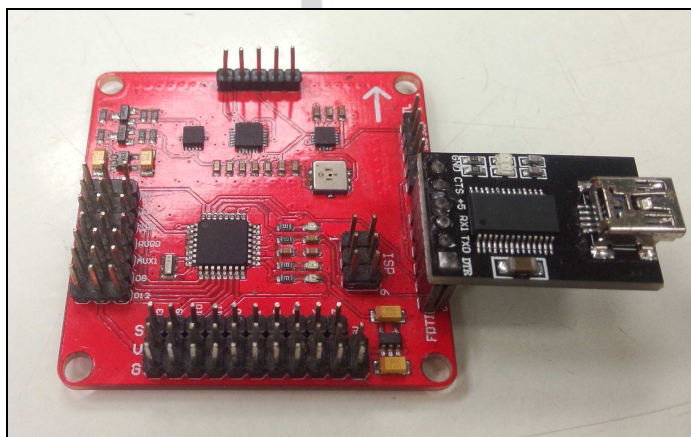
ผู้วิจัยได้ทำการออกแบบการทดลอง การตรวจสอบความถูกต้องของ Servo motors โดย เขียนโปรแกรมสั่งค่ามุมการเคลื่อนที่ให้ Servo Motors ทีละมุม แล้วใช้ไมโคร แทรกเตอร์ในการตรวจสอบมุม สามารถสรุปได้ตาม Flowchart ดังต่อไปนี้



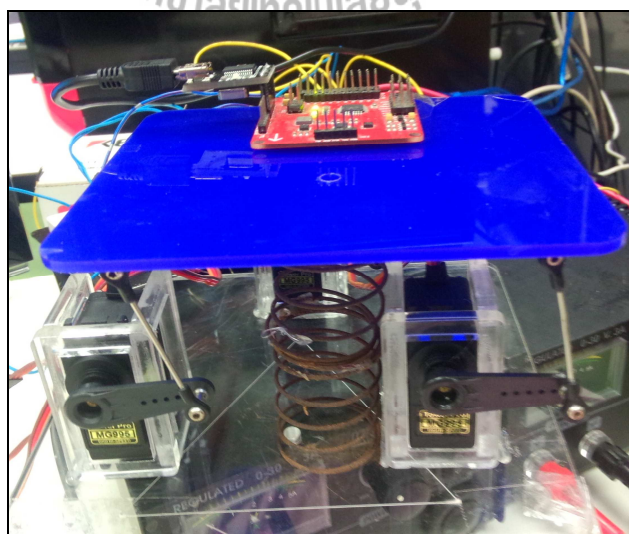
รูปที่ 3.20 Flowchart การตรวจสอบความถูกต้องของเซอร์โวมอเตอร์แต่ละตัว

### 3.5.2 ความถูกต้องของระบบ

ผู้วิจัยได้ทำการออกแบบการทดลอง การตรวจสอบความถูกต้องของระบบ โดยเขียนโปรแกรมสั่งการมุมการเคลื่อนที่ให้ Platform โดยใช้ฟังก์ชันแบบขั้นบันได (Step Function) และ ฟังก์ชันแบบลาดเอียง (Ramp Function) ในการกำหนดสัญญาณอินพุตแล้วใช้ไจโรสโคป (gyroscope) ในงานวิจัยใช้ บอร์ดไมโครคอนโทรลเลอร์ MultiWii 2012 ซึ่งเป็นบอร์ดไมโครคอนโทรลเลอร์ที่ได้รับความนิยมใช้งานอย่างแพร่หลาย ในอากาศยานหลายใบพัด เนื่องจากมีความสามารถในการตรวจสอบทางทาบการบิน และยังมีระบบ GPS ติดตั้งมาด้วย ที่สำคัญคือ มีความน่าเชื่อถือ และวัดค่าได้อย่างแม่นยำ ทำการตรวจสอบการตอบสนองของระบบ



รูปที่ 3.21 บอร์ดไมโครคอนโทรลเลอร์ MultiWii 2012



รูปที่ 3.22 การทดสอบความถูกต้องของระบบ

## 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<br>โดยการเขียนโปรแกรมรับค่าต้องอาศัย โปรแกรมเสริมในการดึงข้อมูล (Motion Platform Designer) | สามารถส่งต่อข้อมูลออกมาได้<br>โดยการเขียนโปรแกรมรับค่า |
| การเชื่อมต่ออุปกรณ์บังคับและอุปกรณ์แสดงผล | สามารถเชื่อมต่อได้                                                                                                               | สามารถเชื่อมต่อได้                                     |

จะเห็นได้ว่าความสามารถของโปรแกรมทั้งสามข้อนั้นโปรแกรม X-Plane ได้ตอบโจทย์ให้กับการทำงานในงานวิจัยนี้ได้ดี แม้ในข้อที่ 1 จะต้องใช้ทรัพยากรคอมพิวเตอร์มากกว่าโปรแกรม FSX นั้น แต่โปรแกรม FSX ก็ไม่สามารถตอบโจทย์ความต้องการข้อมูลท่าทางการบิน ณ ขณะนั้น เพื่อใช้ในการนำไปประมวลผลเพื่อให้ได้ตำแหน่งการเคลื่อนที่ของ Platform ได้ ซึ่งถือว่าเป็นหัวใจหลักของงานวิจัยนี้

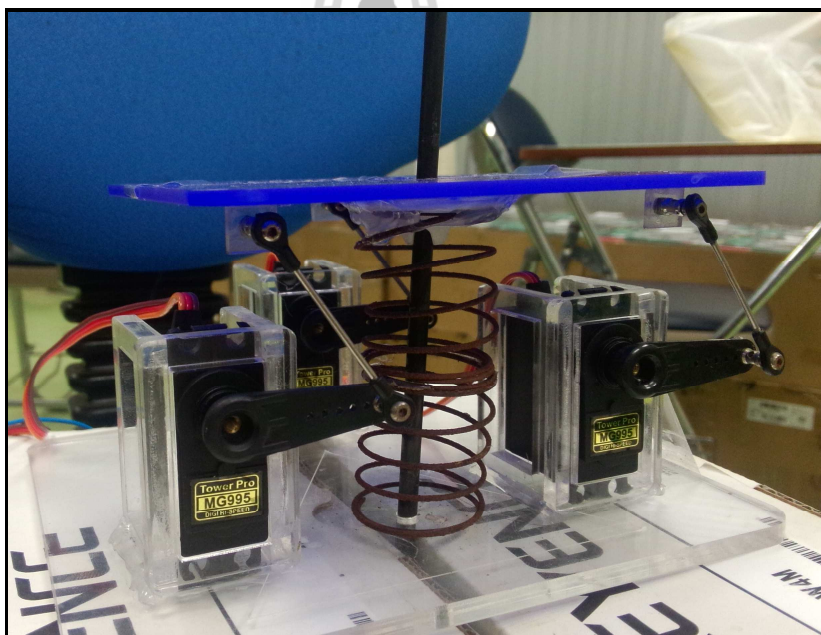
ดังนั้นจึงสรุปในหัวข้อนี้ได้ว่าโปรแกรม X-Plane ถูกเลือกใช้เป็นโปรแกรมที่จะนำมาศึกษาและใช้งานวิจัย

#### 4.2 การออกแบบและขนาดของแผ่นเคลื่อนไหว (Dimension of Platform)

จากข้อกำหนดต่าง ๆ ของแผ่นเคลื่อนไหว (Platform) ในบทที่ 3 นั้น ซึ่งนำมาออกแบบโดยมีพารามิเตอร์ดังตารางที่ 4.2

ตารางที่ 4.2 ค่าพารามิเตอร์ที่ใช้ในการออกแบบ

|                                              |                    |
|----------------------------------------------|--------------------|
| ระยะระหว่าง Servo Motor แต่ละตัว (L)         | 100 มิลลิเมตร (mm) |
| ความยาวแขนของ Servo Motor (r)                | 20 มิลลิเมตร (mm)  |
| ระยะเชิงของแผ่นเคลื่อนไหวกับฐาน ในแนวแกน (d) | 0 มิลลิเมตร (mm)   |
| ความสูงของแผ่นเคลื่อนไหวเทียบกับจุดหมุน      | 80 มิลลิเมตร (mm)  |



รูปที่ 4.1 เครื่องจำลองการบินต้นแบบ

### 4.3 การทดลองเปรียบเทียบผลการเคลื่อนที่ของเซอร์โวมอเตอร์ทั้งสามชุด

เนื่องด้วยการควบคุมตำแหน่งมุมในการหมุนของเซอร์โวมอเตอร์นั้นทำได้โดยใช้ฟังก์ชันในโปรแกรม LabVIEW ร่วมกับ บอร์ด Arduino ที่ได้กล่าวถึงในบทที่ 3 แล้วนั้น เมื่อกำหนดมุมที่ต้องการลงในฟังก์ชันแล้วพบว่ามุมที่เซอร์โวมอเตอร์กวาดไปหรือเคลื่อนที่ไปนั้นไม่ตรงตามคำสั่งที่ต้องการดังนั้นจึงต้องทำการศึกษาเพื่อเปรียบเทียบเซอร์โวมอเตอร์เพื่อให้ได้มุมการหมุนที่ถูกต้องตามความต้องการ โดยในการทดลองนั้นแบ่งออกเป็นขั้นตอนดังนี้คือ

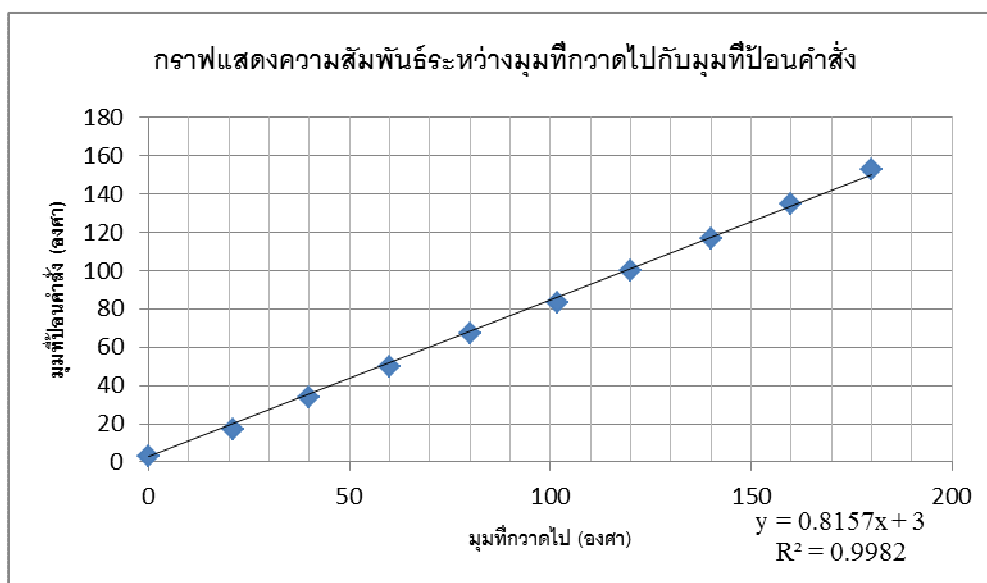
1. การทดลองเพื่อหามุมที่เซอร์โวมอเตอร์กวาดไปโดยให้คำสั่งตามปกติ
2. นำข้อมูลที่ได้จากขั้นตอนที่ 1 มาหาความสัมพันธ์สร้างสมการเชิงเส้นเพื่อหาค่ามุมที่ต้องป้อนให้เซอร์โวมอเตอร์เพื่อให้ได้มุมการเคลื่อนที่ที่ต้องการ
3. นำสมการที่ได้มาทดลองหามุมที่ต้องการและสั่งการเซอร์โวมอเตอร์ว่ามุมที่เซอร์โวมอเตอร์เคลื่อนที่นั้นถูกต้องตรงตามค่าที่ต้องการหรือไม่ หากไม่มีความคลาดเคลื่อนที่สามารถยอมรับได้หรือไม่

โดยมีผลการทดลองของเซอร์โวมอเตอร์แต่ละตัวดังนี้

#### 4.3.1 เซอร์โวมอเตอร์ตัวที่ 1

ตารางที่ 4.3 ผลการตอบสนองของเซอร์โวมอเตอร์ตัวที่ 1

| มุมที่ต้องการ (องศา) | มุมที่ต้องป้อนคำสั่ง (องศา) | มุมที่กวาดไป(องศา) |
|----------------------|-----------------------------|--------------------|
| 0                    | 3                           | 0                  |
| 20                   | 17                          | 21                 |
| 40                   | 34                          | 40                 |
| 60                   | 50                          | 60                 |
| 80                   | 67                          | 80                 |
| 100                  | 83                          | 102                |
| 120                  | 100                         | 120                |
| 140                  | 117                         | 140                |
| 160                  | 135                         | 160                |
| 180                  | 153                         | 180                |



รูปที่ 4.2 กราฟแสดงความสัมพันธ์ระหว่างมุมที่กวาดไปกับมุมที่ป้อนคำสั่งของเซอร์โวมอเตอร์ 1

จากกราฟในรูปที่ 4.2 เห็นได้ว่าข้อมูลมีความสัมพันธ์กันเป็นเชิงเส้นตรง จะได้สมการความสัมพันธ์เป็น  $y = 0.8157x + 3$

เมื่อ  $x$  คือมุมที่ต้องการ และ  $y$  คือมุมที่ต้องป้อนคำสั่ง

เมื่อได้สมการในการสั่งงานเซอร์โวมอเตอร์แล้วนั้นก็นำมาทำการทดลองอีกครั้ง เพื่อหาความแม่นยำของการสั่งงานเซอร์โวมอเตอร์ผ่านสมการที่ได้มา ซึ่งผลการทดลองเป็นดังนี้

ตารางที่ 4.4 ค่าความแม่นยำของการสั่งงานเซอร์โวมอเตอร์ตัวที่ 1

| มุมที่ต้องการ<br>(องศา) | ค่ามุมที่ต้องป้อนคำสั่ง<br>(องศา) | มุมที่กวาดไป<br>(องศา) |    |    | ค่าเฉลี่ยมุมที่กวาดไป<br>(องศา) | FSO Error<br>(%) |
|-------------------------|-----------------------------------|------------------------|----|----|---------------------------------|------------------|
|                         |                                   | 1                      | 2  | 3  |                                 |                  |
| 0                       | 3                                 | 0                      | 0  | 0  | 0                               | 0.00             |
| 10                      | 11                                | 12                     | 12 | 12 | 12                              | 1.11             |
| 20                      | 19                                | 22                     | 22 | 22 | 22                              | 1.11             |
| 30                      | 27                                | 30                     | 30 | 30 | 30                              | 0.00             |
| 40                      | 36                                | 42                     | 42 | 42 | 42                              | 1.11             |
| 50                      | 44                                | 52                     | 52 | 52 | 52                              | 1.11             |

ตารางที่ 4.4 ค่าความแม่นยำของการใช้งานเซอร์โวมอเตอร์ตัวที่ 1 (ต่อ)

| มุมที่ต้องการ<br>(องศา) | ค่ามุมที่ต้องป้อนคำสั่ง<br>(องศา) | มุมที่กวาดไป<br>(องศา) |     |     | ค่าเฉลี่ยมุมที่กวาดไป<br>(องศา) | FSO Error<br>(%) |
|-------------------------|-----------------------------------|------------------------|-----|-----|---------------------------------|------------------|
|                         |                                   | 1                      | 2   | 3   |                                 |                  |
| 60                      | 52                                | 62                     | 62  | 62  | 62                              | 1.11             |
| 70                      | 60                                | 71                     | 71  | 71  | 71                              | 0.56             |
| 80                      | 68                                | 81                     | 81  | 81  | 81                              | 0.56             |
| 90                      | 76                                | 91                     | 91  | 91  | 91                              | 0.56             |
| 100                     | 85                                | 102                    | 102 | 102 | 102                             | 1.11             |
| 110                     | 93                                | 110                    | 110 | 110 | 110                             | 0.00             |
| 120                     | 101                               | 120                    | 120 | 120 | 120                             | 0.00             |
| 130                     | 109                               | 130                    | 130 | 130 | 130                             | 0.00             |
| 140                     | 117                               | 140                    | 140 | 140 | 140                             | 0.00             |
| 150                     | 125                               | 148                    | 148 | 148 | 148                             | 1.11             |
| 160                     | 134                               | 159                    | 159 | 159 | 159                             | 0.56             |
| 170                     | 142                               | 168                    | 168 | 168 | 168                             | 1.11             |
| 180                     | 150                               | 178                    | 178 | 178 | 178                             | 1.11             |

จากตารางที่ 4.4 พบว่าสมการที่สร้างขึ้นมาเพื่อหามุมที่ต้องป้อนคำสั่งเพื่อให้ได้มุมที่ต้องการนั้นมีค่าความคลาดเคลื่อนสูงสุดอยู่ที่ 1.11 % เมื่อเทียบกับช่วงความห่างของมุมที่กวาดไปได้ทั้งหมด (180 องศา) ซึ่งค่าความคลาดเคลื่อนสูงสุดนี้สามารถยอมรับได้

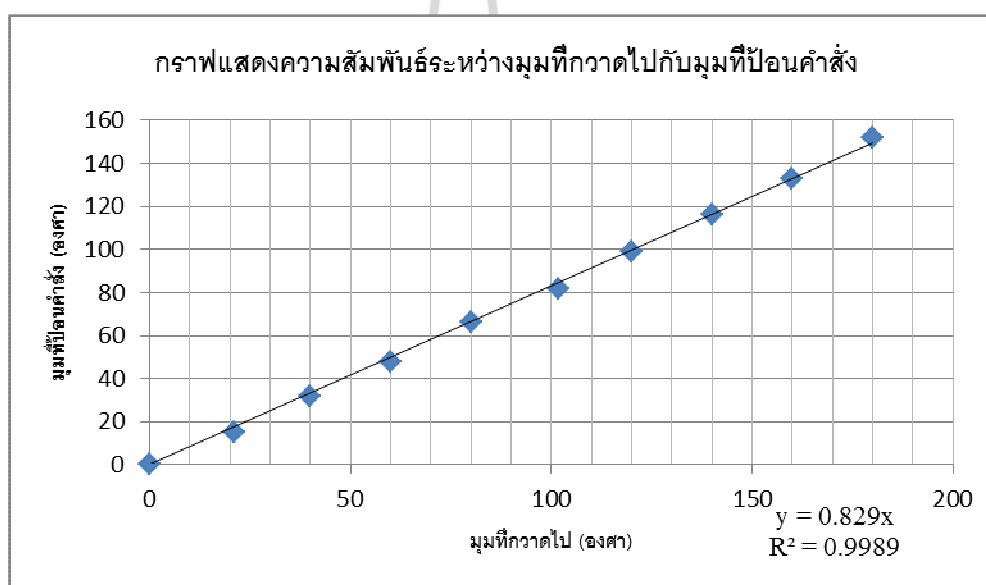
#### 4.3.2 เซอร์โวมอเตอร์ตัวที่ 2

ตารางที่ 4.5 ผลการตอบสนองของเซอร์โวมอเตอร์ตัวที่ 2

| มุมที่ต้องการ (องศา) | มุมที่ต้องป้อนคำสั่ง (องศา) | มุมที่กวาดไป(องศา) |
|----------------------|-----------------------------|--------------------|
| 0                    | 3                           | 0                  |
| 20                   | 17                          | 21                 |
| 40                   | 34                          | 40                 |

ตารางที่ 4.5 ผลการตอบสนองของเซอร์โวมอเตอร์ตัวที่ 2 (ต่อ)

| มุมที่ต้องการ (องศา) | มุมที่ต้องป้อนคำสั่ง (องศา) | มุมที่กวาดไป(องศา) |
|----------------------|-----------------------------|--------------------|
| 60                   | 50                          | 60                 |
| 80                   | 67                          | 80                 |
| 100                  | 83                          | 102                |
| 120                  | 100                         | 120                |
| 140                  | 117                         | 140                |
| 160                  | 135                         | 160                |
| 180                  | 153                         | 180                |



รูปที่ 4.3 กราฟแสดงความสัมพันธ์ระหว่างมุมที่กวาดไปกับมุมที่ป้อนคำสั่งของเซอร์โวมอเตอร์ 2

จากกราฟในรูปที่ 4.3 เห็นได้ว่าข้อมูลมีความสัมพันธ์กันเป็นเชิงเส้นตรง จะได้สมการความสัมพันธ์เป็น  $y = 0.829x$

เมื่อ  $x$  คือมุมที่ต้องการ และ  $y$  คือมุมที่ต้องป้อนคำสั่ง

เมื่อได้สมการในการสั่งงานเซอร์โวมอเตอร์แล้วนั้นก็นำมาทำการทดลองอีกครั้ง เพื่อหาความแม่นยำของการสั่งงานเซอร์โวมอเตอร์ผ่านสมการที่ได้มา ซึ่งผลการทดลองเป็นดังนี้

ตารางที่ 4.6 ค่าความแม่นยำของการใช้งานเซอร์โวมอเตอร์ตัวที่ 2

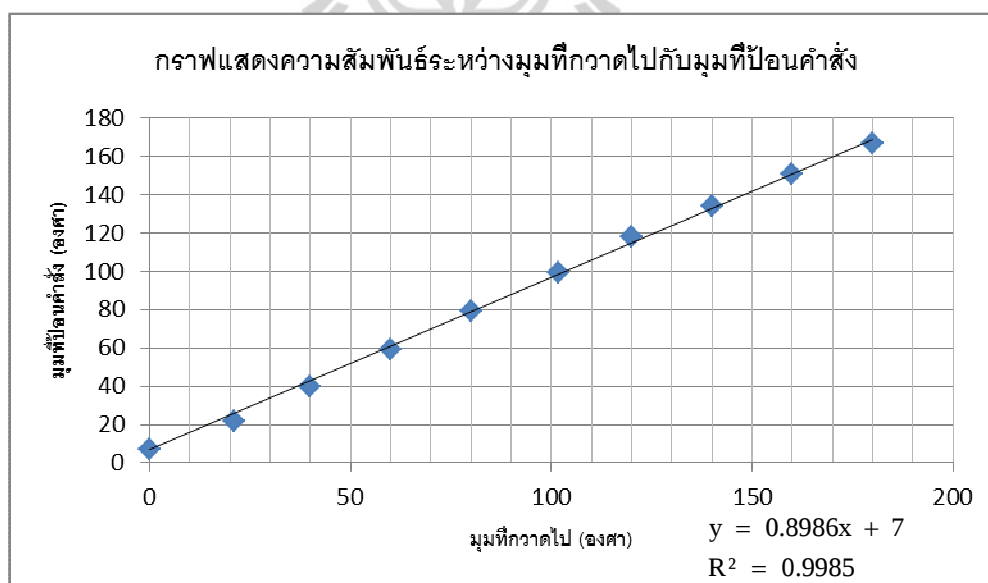
| มุมที่ต้องการ<br>(องศา) | ค่ามุมที่ต้องป้อนคำสั่ง<br>(องศา) | มุมที่กวาดไป<br>(องศา) |     |     | ค่าเฉลี่ยมุมที่กวาดไป<br>(องศา) | FSO Error<br>(%) |
|-------------------------|-----------------------------------|------------------------|-----|-----|---------------------------------|------------------|
|                         |                                   | 1                      | 2   | 3   |                                 |                  |
| 0                       | 0                                 | 0                      | 0   | 0   | 0                               | 0.00             |
| 10                      | 8                                 | 12                     | 12  | 12  | 12                              | 1.11             |
| 20                      | 17                                | 23                     | 23  | 23  | 23                              | <u>1.67</u>      |
| 30                      | 25                                | 32                     | 32  | 32  | 32                              | 1.11             |
| 40                      | 33                                | 41                     | 41  | 41  | 41                              | 0.56             |
| 50                      | 41                                | 51                     | 51  | 51  | 51                              | 0.56             |
| 60                      | 50                                | 62                     | 62  | 62  | 62                              | 1.11             |
| 70                      | 58                                | 72                     | 72  | 72  | 72                              | 1.11             |
| 80                      | 66                                | 82                     | 82  | 82  | 82                              | 1.11             |
| 90                      | 75                                | 92                     | 92  | 92  | 92                              | 1.11             |
| 100                     | 83                                | 102                    | 102 | 102 | 102                             | 1.11             |
| 110                     | 91                                | 112                    | 112 | 112 | 112                             | 1.11             |
| 120                     | 99                                | 122                    | 122 | 122 | 122                             | 1.11             |
| 130                     | 108                               | 132                    | 132 | 132 | 132                             | 1.11             |
| 140                     | 116                               | 142                    | 142 | 142 | 142                             | 1.11             |
| 150                     | 124                               | 151                    | 151 | 151 | 151                             | 0.56             |
| 160                     | 133                               | 160                    | 160 | 160 | 160                             | 0.00             |
| 170                     | 141                               | 170                    | 170 | 170 | 170                             | 0.00             |
| 180                     | 149                               | 180                    | 180 | 180 | 180                             | 0.00             |

จากตารางที่ 4.6 พบว่าสมการที่สร้างขึ้นมาเพื่อหามุมที่ต้องป้อนคำสั่งเพื่อให้ได้มุมที่ต้องการนั้นมีค่าความคลาดเคลื่อนสูงสุดอยู่ที่ 1.67 % เมื่อเทียบกับช่วงความห่างของมุมที่กวาดไปทั้งหมด (180 องศา) ซึ่งค่าความคลาดเคลื่อนสูงสุดนี้สามารถยอมรับได้

### 4.3.3 เซอร์ไวโมเตอร์ตัวที่ 3

ตารางที่ 4.7 ผลการตอบสนองของเซอร์ไวโมเตอร์ตัวที่ 3

| มุมที่ต้องการ (องศา) | มุมที่ต้องป้อนคำสั่ง (องศา) | มุมที่กวาดไป(องศา) |
|----------------------|-----------------------------|--------------------|
| 0                    | 7                           | 0                  |
| 20                   | 22                          | 21                 |
| 40                   | 40                          | 40                 |
| 60                   | 59                          | 60                 |
| 80                   | 79                          | 80                 |
| 100                  | 99                          | 102                |
| 120                  | 118                         | 120                |
| 140                  | 134                         | 140                |
| 160                  | 151                         | 160                |
| 180                  | 167                         | 180                |



รูปที่ 4. 4 กราฟแสดงความสัมพันธ์ระหว่างมุมที่กวาดไปกับมุมที่ป้อนคำสั่งของเซอร์ไวโมเตอร์ 3

จากกราฟในรูปที่ 4.4 เห็นได้ว่าข้อมูลมีความสัมพันธ์กันเป็นเชิงเส้นตรง จะได้  
สมการความสัมพันธ์เป็น  $y = 0.8986x + 7$   
เมื่อ  $x$  คือมุมที่ต้องการ และ  $y$  คือมุมที่ต้องป้อนคำสั่ง  
เมื่อได้สมการในการสั่งงานเซอร์โวมอเตอร์แล้วนั้นก็นำมาทำการทดลองอีกครั้ง  
เพื่อหาความแม่นยำของการสั่งงานเซอร์โวมอเตอร์ผ่านสมการที่ได้มา ซึ่งผลการทดลองเป็นดังนี้

ตารางที่ 4.8 ค่าความแม่นยำของการสั่งงานเซอร์โวมอเตอร์ตัวที่ 3

| มุมที่ต้องการ<br>(องศา) | ค่ามุมที่ต้องป้อนคำสั่ง<br>(องศา) | มุมที่กวาดไป<br>(องศา) |     |     | ค่าเฉลี่ยมุมที่กวาดไป<br>(องศา) | FSO Error<br>(%) |
|-------------------------|-----------------------------------|------------------------|-----|-----|---------------------------------|------------------|
|                         |                                   | 1                      | 2   | 3   |                                 |                  |
| 0                       | 7                                 | 0                      | 0   | 0   | 0                               | 0.00             |
| 10                      | 16                                | 10                     | 10  | 10  | 10                              | 0.00             |
| 20                      | 25                                | 22                     | 22  | 22  | 22                              | 1.11             |
| 30                      | 34                                | 32                     | 32  | 32  | 32                              | 1.11             |
| 40                      | 43                                | 41                     | 41  | 41  | 41                              | 0.56             |
| 50                      | 52                                | 51                     | 51  | 51  | 51                              | 0.56             |
| 60                      | 61                                | 60                     | 60  | 60  | 60                              | 0.00             |
| 70                      | 70                                | 70                     | 70  | 70  | 70                              | 0.00             |
| 80                      | 79                                | 80                     | 80  | 80  | 80                              | 0.00             |
| 90                      | 88                                | 89                     | 89  | 89  | 89                              | 0.56             |
| 100                     | 97                                | 98                     | 98  | 98  | 98                              | 1.11             |
| 110                     | 106                               | 108                    | 108 | 108 | 108                             | 1.11             |
| 120                     | 115                               | 118                    | 118 | 118 | 118                             | 1.11             |
| 130                     | 124                               | 129                    | 129 | 129 | 129                             | 0.56             |
| 140                     | 133                               | 139                    | 139 | 139 | 139                             | 0.56             |
| 150                     | 142                               | 150                    | 150 | 150 | 150                             | 0.00             |
| 160                     | 151                               | 160                    | 160 | 160 | 160                             | 0.00             |
| 170                     | 160                               | 170                    | 170 | 170 | 170                             | 0.00             |
| 180                     | 169                               | 180                    | 180 | 180 | 180                             | 0.00             |

จากตารางที่ 4.8 พบว่าสมการที่สร้างขึ้นมาเพื่อหามุมที่ต้องป้อนคำสั่งเพื่อให้ได้มุมที่ต้องการนั้นมีค่าความคลาดเคลื่อนสูงสุดอยู่ที่ 1.11 % เมื่อเทียบกับช่วงความห่างของมุมที่กวาดไปได้ทั้งหมด (180 องศา) ซึ่งค่าความคลาดเคลื่อนสูงสุดนี้สามารถยอมรับได้

จากข้อมูลการทดลองเปรียบเทียบความแม่นยำของการสั่งงานเซอร์โวมอเตอร์นั้นได้ข้อสรุปดังนี้

ตารางที่ 4.9 ผลการตอบสนองและค่าความคลาดเคลื่อนของเซอร์โวมอเตอร์ ทั้งสาม

| เซอร์โวมอเตอร์ | สมการควบคุม      | ค่าความคลาดเคลื่อนสูงสุด |
|----------------|------------------|--------------------------|
| ตัวที่ 1       | $y = 0.8157 + 3$ | 1.11%                    |
| ตัวที่ 2       | $y = 0.8290 + 0$ | 1.67%                    |
| ตัวที่ 3       | $y = 0.8986 + 7$ | 1.11%                    |

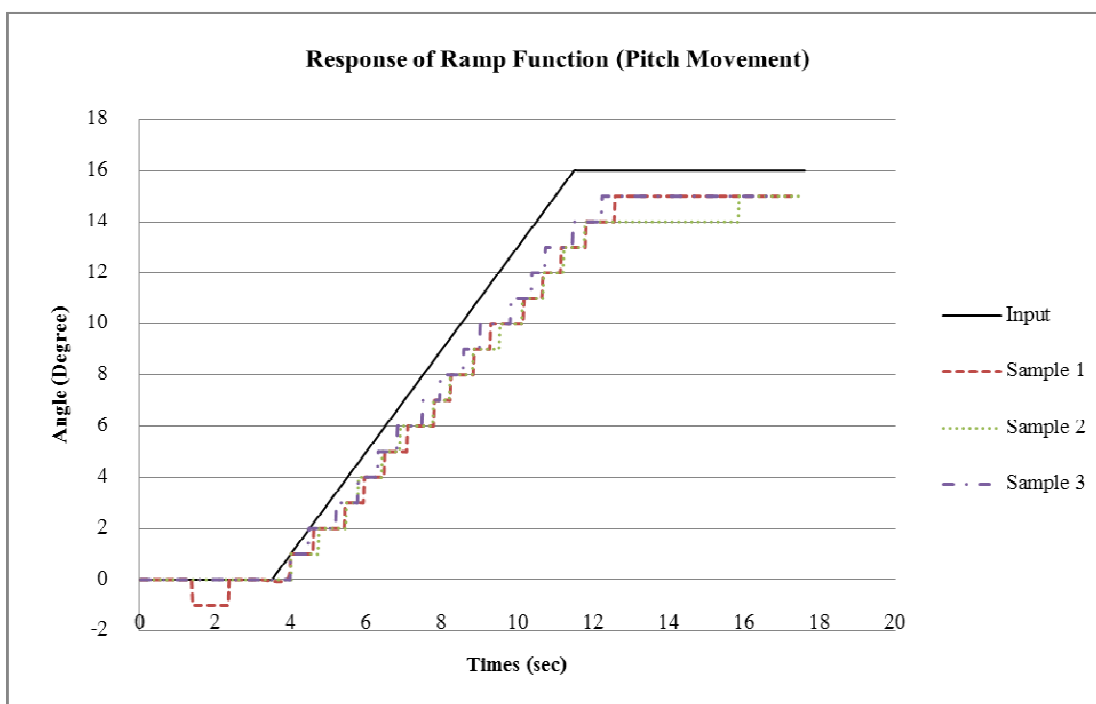
จากการได้สมการเพื่อใช้ในการคำนวณหามุมที่ต้องป้อนคำสั่งให้กับมอเตอร์แล้วซึ่งค่าความคลาดเคลื่อนสูงสุดนั้นสามารถยอมรับได้ ในขั้นตอนต่อไปก็คือการนำเอามอเตอร์ไปติดไว้บน Platform แล้วทำการหมุนตามสมการการคำนวณเพื่อให้ Platform ทำมุมตามต้องการ โดยเพื่อให้เกิดความเชื่อมั่นในการคำนวณและอุปกรณ์ จำเป็นต้องทดลองโดยการให้มุมของ Platform ที่ต้องการแล้วทำการสั่งงานมอเตอร์เพื่อหมุนไปยังมุมการหมุนที่ได้จากการคำนวณแล้ววัดมุมของ Platform ที่ได้กับมุมของ Platform ที่ต้องการ

#### 4.4 การทดลองความถูกต้องของระบบ

เนื่องจากงานวิจัยนี้ ได้ทำการออกแบบระบบการควบคุมเป็นแบบเปิด (Open Loop control Control) ส่งผลให้ค่าเอาต์พุตไม่มีผลต่อการควบคุมขบวนการของระบบ ผู้วิจัยจึงได้ทำการทดลองตรวจสอบการตอบสนองของระบบดังที่กล่าวไว้ในบทที่ 3 เพื่อเป็นข้อพิสูจน์ความน่าเชื่อถือของระบบที่ได้ทำการวิจัยว่ามีความถูกต้องน่าเชื่อถือ มากน้อยเพียงใด

##### 4.4.1 การตอบสนองฟังก์ชันลาดเอียง (Ramp Function)

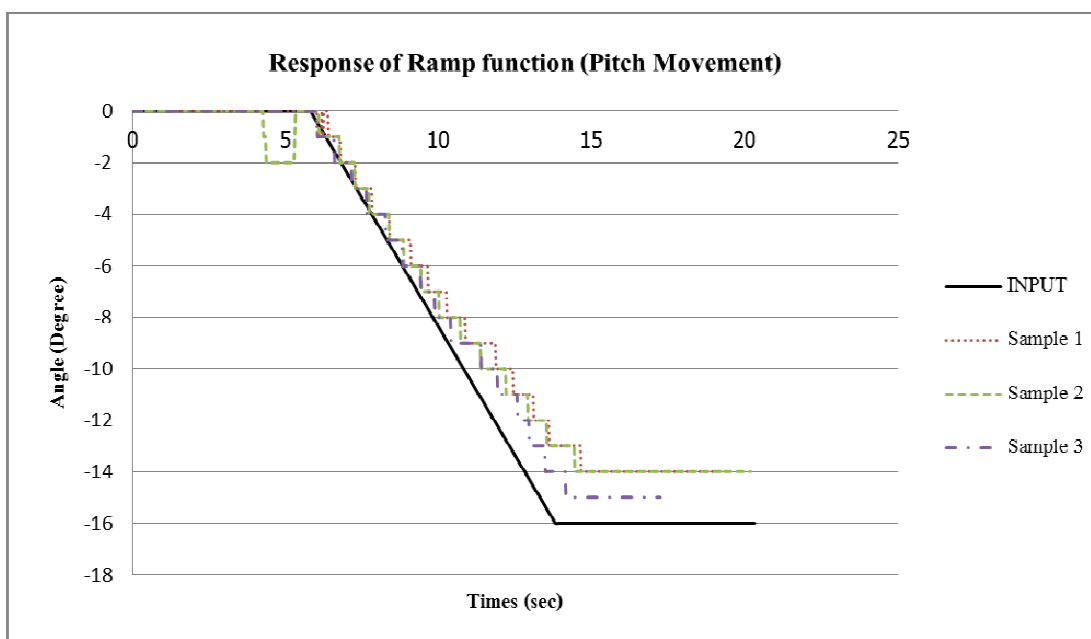
ในการทดสอบการตอบสนองฟังก์ชันลาดเอียง ผู้วิจัยได้ทดสอบการตอบสนองลักษณะการเคลื่อนที่ใน 2 ทิศทาง คือ การเคลื่อนที่รอบแกนตามขวาง (Pitch) และเคลื่อนที่รอบแกนตามยาว (Roll) ซึ่งมีผลการตอบสนองดังนี้



รูปที่ 4.5 กราฟแสดงผลการตอบสนองฟังก์ชันลาดเอียงของการเคลื่อนที่รอบแกนตามขวางในทิศทางบวก

จากกราฟการตอบสนองฟังก์ชันลาดเอียงในการเคลื่อนที่รอบแกนตามขวางในทิศทางบวกโดยให้แผ่นเคลื่อนไหว (Platform) เคลื่อนที่จากมุม 0 องศาไปที่ 16 องศา ในเวลา 8 วินาที โดยทำการทดลอง 3 ครั้ง เพื่อเทียบผลการตอบสนองดังแสดงในรูปที่ 4.5 พบว่า ในช่วงที่แผ่นเคลื่อนไหวเริ่มจะทำงาน เซอร์โวมอเตอร์จะมีการเซตตัวเพื่อรอรับคำสั่งสัญญาณ ที่ถูกป้อนเข้ามา ทำให้เกิดการเคลื่อนที่ผิดพลาดจากคำสั่งสัญญาณในช่วงแรก (0-4 วินาที) หลังจากนั้น แผ่นเคลื่อนไหวก็จะเริ่มเคลื่อนตัวเป็นมุมบวกตั้งแต่วันที่ 4 วินาที เป็นต้นไป จนกระทั่งเข้าสู่สถานะคงที่ที่เวลา 12 วินาที ซึ่งสัญญาณที่ป้อนใช้เวลาการเคลื่อนที่ 8 วินาที (จากวินาทีที่ 3.5 ถึง วินาทีที่ 11.5) ในขณะที่การตอบสนองใช้เวลาในการเคลื่อนที่ 8 วินาที เช่นเดียวกัน โดยเริ่มจากวินาทีที่ 4 ถึง วินาทีที่ 12 ซึ่งช้ากว่าสัญญาณเพียง 0.5 วินาที

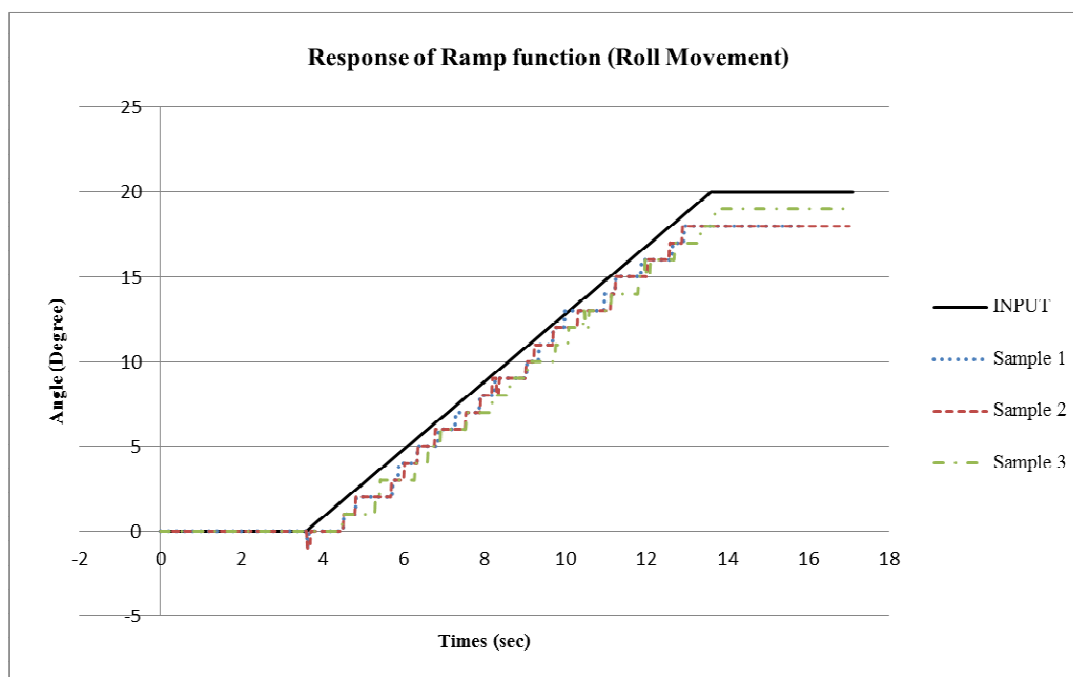
ค่าความคลาดเคลื่อนของแผ่นเคลื่อนไหวมากที่สุดอยู่ที่ตำแหน่งสภาวะคงที่ เมื่อเทียบมุมมององศาที่แผ่นเคลื่อนไหวเคลื่อนที่ได้ ( $14^\circ$ ) กับสัญญาณที่ป้อน ( $16^\circ$ ) พบว่า มีค่าคลาดเคลื่อนจากคำสั่ง 2 องศา คิดเป็นเปอร์เซ็นต์ความคลาดเคลื่อน 12.5 เปอร์เซ็นต์



รูปที่ 4.6 กราฟแสดงผลการตอบสนองฟังก์ชันลาดเอียงของการเคลื่อนที่รอบแกนตามขวางในทิศทางลบ

จากกราฟการตอบสนองฟังก์ชันลาดเอียงในการเคลื่อนที่รอบแกนตามขวางในทิศทางลบโดยให้แผ่นเคลื่อนไหว (Platform) เคลื่อนที่จากมุม 0 องศาไปที่ -16 องศา ในเวลา 8 วินาที โดยทำการทดลอง 3 ครั้ง เพื่อเทียบผลการตอบสนองดังแสดงในรูปที่ 4.6 พบว่า ในช่วงที่แผ่นเคลื่อนไหวเริ่มจะทำงาน เซอร์โวมอเตอร์จะมีการเซตตัวเพื่อรอรับคำสั่งสัญญาณ ที่ถูกป้อนเข้ามา ทำให้เกิดการเคลื่อนที่ผิดปกติจากคำสั่งสัญญาณในช่วงแรก (0-5 วินาที) หลังจากนั้น แผ่นเคลื่อนไหวก็จะเริ่มเคลื่อนตัวเป็นมุมลบตั้งแต่วันที่ 6 วินาที เป็นต้นไป จนกระทั่งเข้าสู่สถานะคงที่ที่เวลา 14 วินาที ซึ่งสัญญาณที่ป้อนใช้เวลาการเคลื่อนที่ 8 วินาที (จากวินาทีที่ 6 ถึง วินาทีที่ 14) ในขณะที่การตอบสนองใช้เวลาในการเคลื่อนที่ 8.2 วินาที ซึ่งใช้เวลาในการตอบสนองมากกว่า 0.2 วินาที โดยเริ่มจากวินาทีที่ 6.4 ถึง วินาทีที่ 14.6 ซึ่งช้ากว่าสัญญาณ 0.6 วินาที

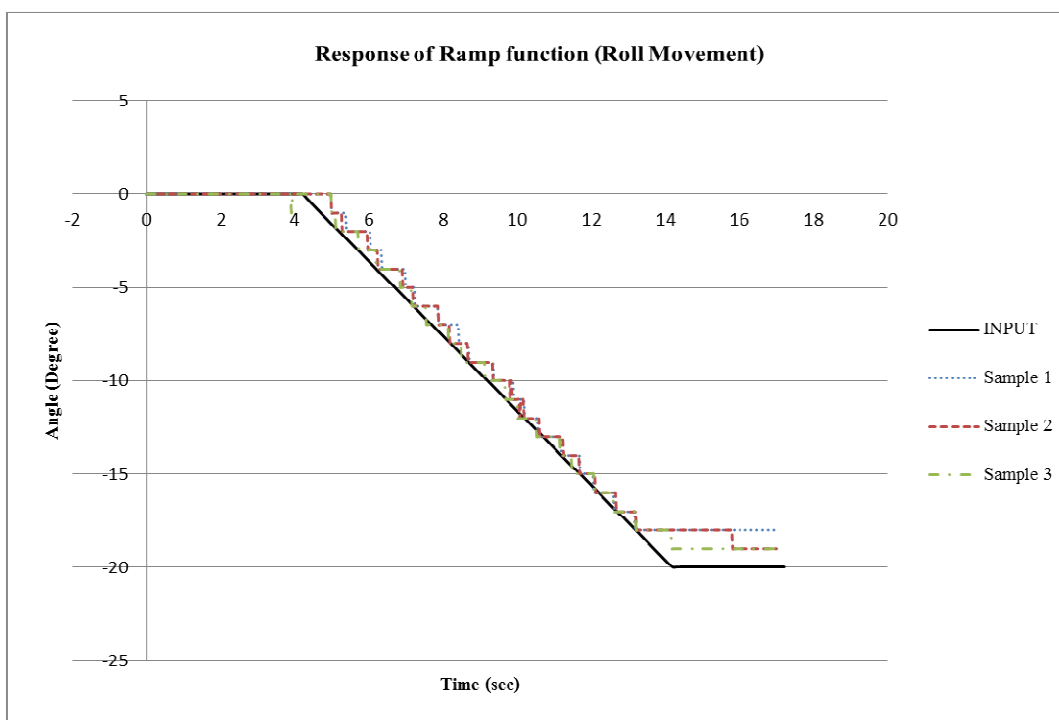
ค่าความคลาดเคลื่อนของแผ่นเคลื่อนไหวสูงสุดอยู่ที่ตำแหน่งสถานะคงที่ เมื่อเทียบมุมองศาที่แผ่นเคลื่อนไหวเคลื่อนที่ได้ ( $14^\circ$ ) กับสัญญาณที่ป้อน ( $16^\circ$ ) พบว่า มีค่าคลาดเคลื่อนจากคำสั่ง 2 องศา คิดเป็นเปอร์เซ็นต์ความคลาดเคลื่อน 12.5 เปอร์เซ็นต์



รูปที่ 4.7 กราฟแสดงผลการตอบสนองฟังก์ชันลาดเอียงของการเคลื่อนที่รอบแกนตามยาวในทิศทางบวก

จากกราฟการตอบสนองฟังก์ชันลาดเอียงในการเคลื่อนที่รอบแกนตามยาวในทิศทางบวกโดยให้แผ่นเคลื่อนไหว (Platform) เคลื่อนที่จากมุม 0 องศาไปที่ 20 องศา ในเวลา 9 วินาที โดยทำการทดลอง 3 ครั้ง เพื่อเทียบผลการตอบสนองดังแสดงในรูปที่ 4.7 พบว่า ในช่วงที่แผ่นเคลื่อนไหวเริ่มจะทำงาน เซอร์โวมอเตอร์จะมีการเซตตัวเพื่อรับคำสั่งสัญญาณ ที่ถูกป้อนเข้ามา ทำให้เกิดการเคลื่อนที่ผิดพลาดจากคำสั่งสัญญาณในช่วงแรก (0-4 วินาที) หลังจากนั้น แผ่นเคลื่อนไหวก็จะเริ่มเคลื่อนตัวเป็นมุมบวกตั้งแต่วินาที 4 วินาที เป็นต้นไป จนกระทั่งเข้าสู่สถานะคงที่ที่เวลา 13 วินาที ซึ่งสัญญาณที่ป้อนใช้เวลาการเคลื่อนที่ 9 วินาที (จากวินาทีที่ 4 ถึง วินาทีที่ 13) ในขณะที่การตอบสนองใช้เวลาในการเคลื่อนที่ 9.2 วินาที ซึ่งใช้เวลาในการตอบสนองมากกว่า 0.2 วินาที โดยเริ่มจากวินาทีที่ 4.7 ถึง วินาทีที่ 13.7 ซึ่งช้ากว่าสัญญาณ 0.7 วินาที

ค่าความคลาดเคลื่อนของแผ่นเคลื่อนไหวมากที่สุดอยู่ที่ตำแหน่งสภาวะคงที่ เมื่อเทียบมุมองศาที่แผ่นเคลื่อนไหวเคลื่อนที่ได้ ( $18^\circ$ ) กับสัญญาณที่ป้อน ( $20^\circ$ ) พบว่า มีค่าคลาดเคลื่อนจากคำสั่ง 2 องศา คิดเป็นเปอร์เซ็นต์ความคลาดเคลื่อน 10 เปอร์เซ็นต์



รูปที่ 4.8 กราฟแสดงผลการตอบสนองฟังก์ชันลาดเอียงของการเคลื่อนที่รอบแกนตามยาวในทิศทางลบ

จากกราฟการตอบสนองฟังก์ชันลาดเอียงในการเคลื่อนที่รอบแกนตามยาวในทิศทางลบโดยให้แผ่นเคลื่อนไหว (Platform) เคลื่อนที่จากมุม 0 องศาไปที่ -20 องศา ในเวลา 9.8 วินาที โดยทำการทดลอง 3 ครั้ง เพื่อเทียบผลการตอบสนองดังแสดงในรูปที่ 4.8 พบว่า ในช่วงที่แผ่นเคลื่อนไหวเริ่มจะทำงาน เซอร์โวมอเตอร์จะมีการเซตตัวเพื่อรอรับคำสั่งสัญญาณ ที่ถูกป้อนเข้ามา ทำให้เกิดการเคลื่อนที่ผิดแปลกจากคำสั่งสัญญาณในช่วงแรก (0-4 วินาที) หลังจากนั้น แผ่นเคลื่อนไหวก็จะเริ่มเคลื่อนตัวเป็นมุมลบตั้งแต่วเวลา 5 วินาที เป็นต้นไป จนกระทั่งเข้าสู่สถานะคงที่ที่เวลา 15 วินาที ซึ่งสัญญาณที่ป้อนใช้เวลาการเคลื่อนที่ 9.8 วินาที (จากวินาทีที่ 4.3 ถึง วินาทีที่ 14.1) ในขณะที่การตอบสนองใช้เวลาในการเคลื่อนที่ 10.5 วินาที ซึ่งใช้เวลาในการตอบสนองมากกว่า 0.7 วินาที โดยเริ่มจากวินาทีที่ 5 ถึง วินาทีที่ 15.5 ซึ่งช้ากว่าสัญญาณ 1.4 วินาที

ค่าความคลาดเคลื่อนของแผ่นเคลื่อนไหวยาวมากที่สุดอยู่ที่ตำแหน่งสภาวะคงที่ เมื่อเทียบมุมองศาที่แผ่นเคลื่อนไหวเคลื่อนที่ได้ ( $18^\circ$ ) กับสัญญาณที่ป้อน ( $20^\circ$ ) พบว่า มีค่าคลาดเคลื่อนจากคำสั่ง 2 องศา คิดเป็นเปอร์เซ็นต์ความคลาดเคลื่อน 10 เปอร์เซ็นต์

จากการทดลองการตอบสนองฟังก์ชันลาดเอียงข้างต้น สังเกตว่าผลการทดลองของแต่ละการทดลองทั้งสามครั้งให้ผลการทดลองที่สอดคล้องกัน ซึ่งจากความคลาดเคลื่อนของการตอบสนองสามารถวิเคราะห์ได้ 2 ประเด็นหลัก ดังนี้

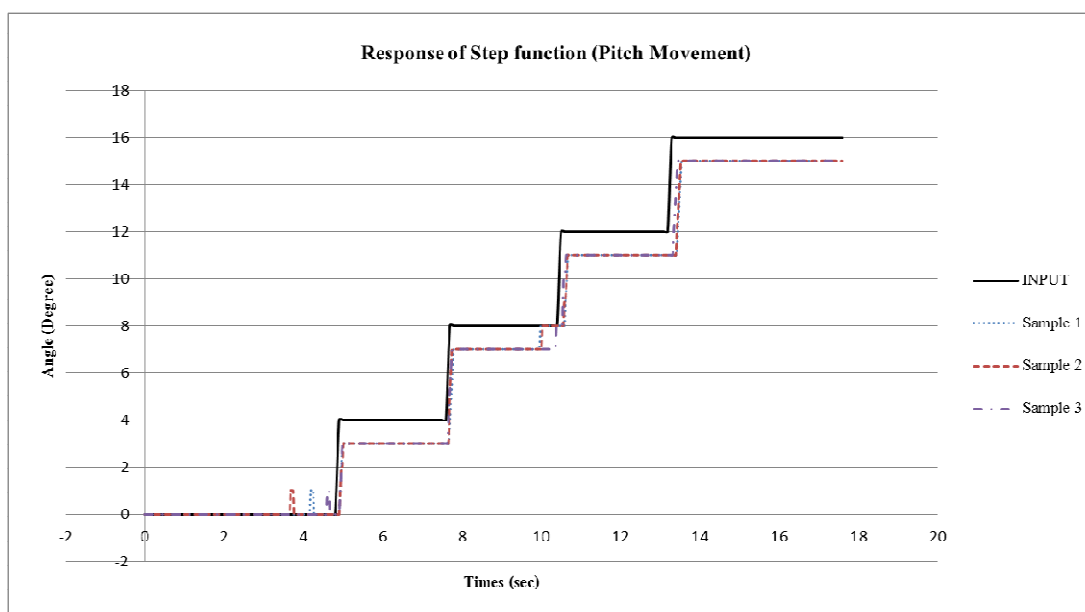
1. จากกราฟรูปที่ 4.5-4.8 ผู้วิจัยได้ป้อนค่าฟังก์ชันลาดเอียงให้กับระบบ แต่ผลการตอบสนองมีลักษณะคล้ายขึ้นบันได ซึ่งอาจมีสาเหตุมาจาก ข้อจำกัดของเครื่องมือวัดค่าการเคลื่อนที่ (MultiWii 2012) ของแผ่นเคลื่อนไหวยังมีความละเอียดเพียง 8 บิต (Bit) จึงสามารถแสดงผลออกมาเป็นเลขจำนวนเต็ม เท่านั้น

2. อุปกรณ์ที่ใช้ในการสร้างแผ่นเคลื่อนไหวยังไม่ได้มาตรฐานเนื่องจากว่าเป็นเพียงแบบจำลองเพื่อศึกษาการตอบสนองเท่านั้น และ ตัวเซอร์ไวโมเตอร์เองก็มีค่าความคลาดเคลื่อนในตัว ซึ่งกล่าวมาแล้วในการทดลองข้างต้น โดยค่าความคลาดเคลื่อนสูงสุด คือ 1.69 % เมื่อประกอบหลายอุปกรณ์เข้าด้วยกันทำให้เป็นการรวมค่าความคลาดเคลื่อนเข้าด้วยกัน



#### 4.4.2 การตอบสนองฟังก์ชันขั้นบันได (Step Function)

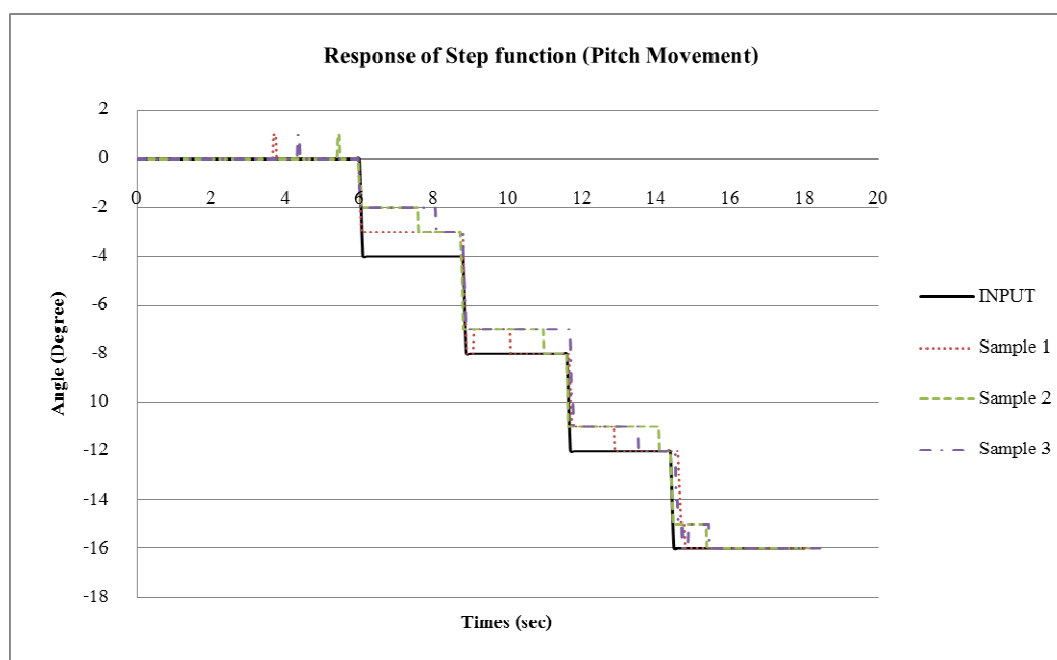
ในการทดสอบการตอบสนองฟังก์ชันขั้นบันได ผู้วิจัยได้ทดสอบการตอบสนองลักษณะการเคลื่อนที่ใน 2 ทิศทาง คือ การเคลื่อนที่รอบแกนตามขวาง (Pitch) และเคลื่อนที่รอบแกนตามยาว (Roll) ซึ่งมีผลการตอบสนองดังนี้



รูปที่ 4.9 กราฟแสดงผลการตอบสนองฟังก์ชันขั้นบันไดของการเคลื่อนที่รอบแกนตามขวางในทิศทางบวก

จากกราฟการตอบสนองจากฟังก์ชันแบบขั้นบันไดในการเคลื่อนที่รอบแกนตามขวางในทิศทางบวก โดยให้แผ่นเคลื่อนไหว (Platform) เคลื่อนที่จากมุม 0 องศาแล้วเพิ่มขึ้นขั้นละ 4 องศา สูงสุดที่ 16 องศา แต่ละขั้นใช้เวลา 2.8 วินาที โดยทำการทดลอง 3 ครั้ง เพื่อเทียบผลการตอบสนองดังแสดงในรูปที่ 4.9 พบว่า ในช่วงที่แผ่นเคลื่อนไหวเริ่มจะทำงาน เซอร์โวมอเตอร์จะมีการเซตตัวเพื่อรอรับคำสั่งสัญญาณ ที่ถูกป้อนเข้ามา ทำให้เกิดการเคลื่อนที่ผิดพลาดจากคำสั่งสัญญาณในช่วงแรก (0-5 วินาที) หลังจากนั้น แผ่นเคลื่อนไหวก็จะเริ่มเคลื่อนตัวเป็นมุมบวกแบบขั้นบันได จนกระทั่งเข้าสู่สถานะคงที่ที่เวลา 13.3 วินาที ซึ่งสัญญาณที่ป้อนใช้เวลาการเคลื่อนที่ 8.4 วินาที (จากวินาทีที่ 4.9 ถึง วินาทีที่ 13.3) ในขณะที่การตอบสนองใช้เวลาในการเคลื่อนที่ 8.5 วินาที ซึ่งใช้เวลาในการตอบสนองมากกว่า 0.1 วินาที โดยเริ่มจากวินาทีที่ 5 ถึง วินาทีที่ 13.5 ซึ่งช้ากว่าสัญญาณ 0.2 วินาที

ค่าความคลาดเคลื่อนของแผ่นเคลื่อนไหวยิ่งน้อยที่สุดอยู่ที่ตำแหน่งสภาวะคงที่ของทุกชั้น เมื่อเทียบมุมมองเสาที่แผ่นเคลื่อนไหวยิ่งน้อยที่สุดได้ ( $3^{\circ}, 7^{\circ}, 11^{\circ}, 15^{\circ}$ ) กับสัญญาณที่ป้อน ( $4^{\circ}, 8^{\circ}, 12^{\circ}, 16^{\circ}$ ) พบว่า มีค่าคลาดเคลื่อนจากคำสั่ง 1 องศา คิดเป็นเปอร์เซ็นต์ความคลาดเคลื่อน 6.25 เปอร์เซ็นต์ และหากพิจารณาผลต่างระหว่างชั้นที่เปลี่ยนไปของสัญญาณกับการตอบสนองมีค่าเท่ากัน คือ 4 องศา จึงอาจกล่าวได้ว่าการตอบสนองแบบขั้นบันไดมีการตอบสนองถูกต้องและแม่นยำ

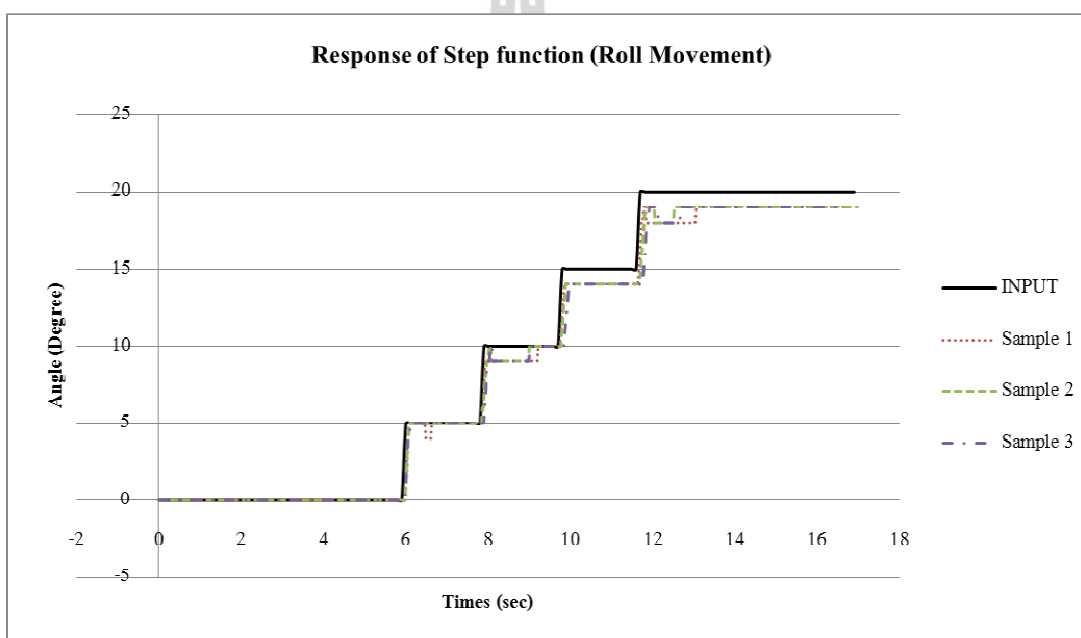


รูปที่ 4.10 กราฟแสดงผลการตอบสนองฟังก์ชันขั้นบันไดของการเคลื่อนที่รอบแกนตามขวางในทิศทางลบ

จากกราฟการตอบสนองจากฟังก์ชันแบบขั้นบันไดในการเคลื่อนที่รอบแกนตามขวางในทิศทางลบ โดยให้แผ่นเคลื่อนไหวยิ่งน้อย (Platform) เคลื่อนที่จากมุม 0 องศาแล้วลดลงขั้นละ 4 องศา สูงสุดที่ -16 องศา แต่ละขั้นใช้เวลา 2.8 วินาที โดยทำการทดลอง 3 ครั้ง เพื่อเทียบผลการตอบสนองดังแสดงในรูปที่ 4.10 พบว่า ในช่วงที่แผ่นเคลื่อนไหวยิ่งน้อยจะเริ่มทำงาน เซอร์โวมอเตอร์จะมีการเซตตัวเพื่อรอรับคำสั่งสัญญาณ ที่ถูกป้อนเข้ามา ทำให้เกิดการเคลื่อนที่ผิดแปลกจากคำสั่งสัญญาณในช่วงแรก (0-6 วินาที) หลังจากนั้น แผ่นเคลื่อนไหวยิ่งน้อยก็จะเริ่มเคลื่อนตัวเป็นมุมลบแบบขั้นบันไดจนกระทั่งเข้าสู่สถานะคงที่ที่เวลา 14.8 วินาที ซึ่งสัญญาณที่ป้อนใช้เวลาการเคลื่อนที่ 8.4 วินาที (จากวินาทีที่ 6 ถึง วินาทีที่ 14.4) ในขณะที่การตอบสนองใช้เวลาในการเคลื่อนที่ 8.8 วินาที ซึ่งใช้

เวลาในการตอบสนองมากกว่า 0.4 วินาที โดยเริ่มจากวินาทีที่ 6 ถึง วินาทีที่ 14.8 ซึ่งช้ากว่าสัญญาณ 0.4 วินาที

ค่าความคลาดเคลื่อนของแผ่นเคลื่อนไหวมามากที่สุดอยู่ที่ตำแหน่งสภาวะคงที่ของขั้นที่หนึ่ง ( $-4^\circ$ ) พบว่า มีค่าคลาดเคลื่อนจากคำสั่ง 2 องศา (สัญญาณสั่ง -4 องศา แผ่นเคลื่อนไหวนตอบสนอง -52 องศา) คิดเป็นเปอร์เซ็นต์ความคลาดเคลื่อน 12.5 เปอร์เซ็นต์ แต่ในระยะเวลาต่อมาแผ่นเคลื่อนไหวนเคลื่อนที่ลดลงไปอีก 1 องศา ซึ่งอาจจะพิจารณาได้ว่า ค่าความคลาดเคลื่อนที่เกิดขึ้นนี้เกิดจากความบกพร่องของอุปกรณ์ที่ใช้ในการสร้างแบบจำลองจึงทำให้เมื่อส่งสัญญาณสั่งค่ามุดดังกล่าวเกิดการดึงรั้งของกลไกทำให้แผ่นเคลื่อนไหวนไม่สามารถเคลื่อนที่ได้ตามที่โปรแกรมสั่งการ และในการทดลองบางครั้งมีการตอบสนองในทันทีทันใด (Real Time)

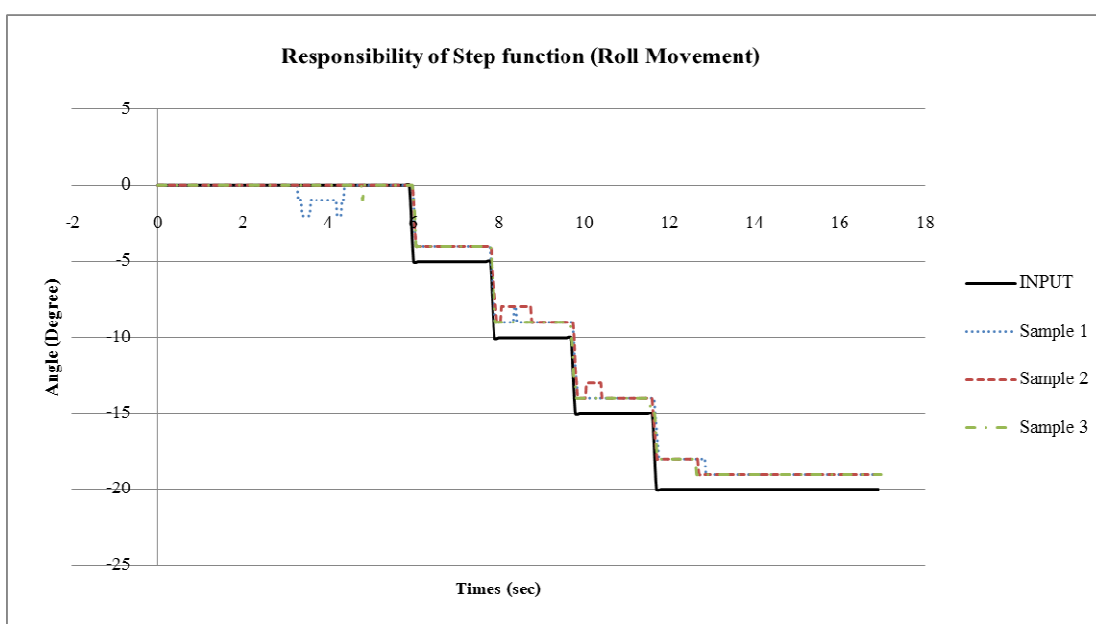


รูปที่ 4.11 กราฟแสดงผลการตอบสนองฟังก์ชันขั้นบันไดของการเคลื่อนที่รอบแกนตามยาวในทิศทางบวก

จากกราฟการตอบสนองจากฟังก์ชันแบบขั้นบันไดในการเคลื่อนที่รอบแกนตามยาวในทิศทางบวก โดยให้แผ่นเคลื่อนไหวน (Platform) เคลื่อนที่จากมุม 0 องศาแล้วเพิ่มขึ้นขั้นละ 5 องศา สูงสุดที่ 20 องศา แต่ละขั้นใช้เวลา 1.9 วินาที โดยทำการทดลอง 3 ครั้ง เพื่อเทียบผลการตอบสนองดังแสดงในรูปที่ 4.11 พบว่า แผ่นเคลื่อนไหวนเริ่มเคลื่อนตัวเป็นมุมบวกแบบขั้นบันได จนกระทั่งเข้าสู่สถานะคงที่ที่เวลา 11.9 วินาที ซึ่งสัญญาณที่ป้อนใช้เวลาการเคลื่อนที่ 5.7 วินาที (จากวินาทีที่ 6 ถึง วินาทีที่ 11.7) ในขณะที่การตอบสนองใช้เวลาในการเคลื่อนที่ 5.8 วินาที ซึ่งใช้เวลาใน

การตอบสนองมากกว่า 0.1 วินาที โดยเริ่มจากวินาทีที่ 6.1 ถึง วินาทีที่ 11.9 ซึ่งช้ากว่าสัญญาณ 0.2 วินาที แต่ในสถานะคงที่ของขั้นแรกมีผลการตอบสนองแบบทันทีทันใด (Real Time)

ค่าความคลาดเคลื่อนของแผ่นเคลื่อนไหวยามากที่สุดอยู่ที่ตำแหน่งสถานะคงที่ของขั้นสุดท้ายเมื่อเทียบมุมมองเสาที่แผ่นเคลื่อนไหวยเคลื่อนที่ได้ ( $18^\circ$ ) กับสัญญาณที่ป้อน ( $20^\circ$ ) พบว่า มีค่าคลาดเคลื่อนจากคำสั่ง 2 องศา คิดเป็นเปอร์เซ็นต์ความคลาดเคลื่อน 12.5 เปอร์เซ็นต์ แต่เป็นเพียงแค่ช่วงเริ่มต้นของสถานะเท่านั้น หลังจากนั้นแผ่นเคลื่อนไหวยก็จะเคลื่อนที่เพิ่มขึ้นเข้าสู่สถานะคงที่มีมีค่าคลาดเคลื่อนลดลง



รูปที่ 4.12 กราฟแสดงผลการตอบสนองฟังก์ชันขั้นบันไดของการเคลื่อนที่รอบแกนตามยาวในทิศทางลบ

จากกราฟการตอบสนองจากฟังก์ชันแบบขั้นบันไดในการเคลื่อนที่รอบแกนตามขวางในทิศทางลบ โดยให้แผ่นเคลื่อนไหวย (Platform) เคลื่อนที่จากมุม 0 องศาแล้วลดลงขั้นละ 5 องศา สูงสุดที่  $-20$  องศา แต่ละขั้นใช้เวลา 1.9 วินาที โดยทำการทดลอง 3 ครั้ง เพื่อเทียบผลการตอบสนองดังแสดงในรูปที่ 4.12 พบว่า ในช่วงที่แผ่นเคลื่อนไหวยจะเริ่มทำงาน เซอร์โวมอเตอร์จะมีการเซตตัวเพื่อรอรับค่าสัญญาณ ที่ถูกป้อนเข้ามา ทำให้เกิดการเคลื่อนที่ผิดแปลกจากค่าสัญญาณในช่วงแรก (0-6 วินาที) หลังจากนั้น แผ่นเคลื่อนไหวยก็จะเริ่มเคลื่อนตัวเป็นมุมลบแบบขั้นบันได จนกระทั่งเข้าสู่สถานะคงที่ที่เวลา 11.75 วินาที ซึ่งสัญญาณที่ป้อนใช้เวลาการเคลื่อนที่ 5.7 วินาที (จากวินาทีที่ 6 ถึง วินาทีที่ 11.7) ในขณะที่การตอบสนองใช้เวลาในการเคลื่อนที่ 5.7 วินาที ซึ่งใช้เวลาใน

การตอบสนองเท่ากับสัญญาณที่ป้อน โดยเริ่มจากวินาทีที่ 6.05 ถึง วินาทีที่ 11.75 ซึ่งช้ากว่าสัญญาณ 0.05 วินาที ซึ่งถือว่าตอบสนองได้แทบจะทันทีทันใด (Real Time)

ค่าความคลาดเคลื่อนของแผ่นเคลื่อนไหวมากที่สุดอยู่ที่ตำแหน่งเริ่มต้นของสภาวะคงที่ของชั้น 2, 3 และ 4 เมื่อเทียบมุมมองชาติแผ่นเคลื่อนไหวก่อนที่ได้ กับสัญญาณที่ป้อน พบว่า มีค่าคลาดเคลื่อนจากคำสั่ง 2 องศา คิดเป็นเปอร์เซ็นต์ความคลาดเคลื่อน 12.5 เปอร์เซ็นต์ และเมื่อผ่านไประยะหนึ่งแผ่นเคลื่อนไหวก่อนที่เข้าหามุมองศาที่สั่งทำให้ค่าความคลาดเคลื่อนลดลงและเข้าสู่สภาวะคงที่ในที่สุด



## บทที่ 5

### สรุปผลและข้อเสนอแนะ

#### 5.1 สรุปงานวิจัย

งานวิจัยนี้เป็นการออกแบบเครื่องจำลองการบินต้นแบบ ที่สามารถเคลื่อนที่ได้ 3 องศาอิสระ (Roll, Pitch และ Heave) เพื่อศึกษาการเชื่อมโยงระหว่างโปรแกรมจำลองการบินและเครื่องจำลองการเคลื่อนที่ ซึ่งจากการศึกษาเครื่องจำลองการบินต้นแบบ ผู้วิจัยได้ทำการดึงค่าข้อมูลท่าทางการเคลื่อนที่แบบ Roll แบบ Pitch และค่าความเร่งของเครื่องบินจากโปรแกรมจำลองการบินแล้วนำมาแปลงเป็นการเคลื่อนที่ของเครื่องจำลองการบิน โดยผลการวิจัยเครื่องจำลองการบินที่ออกแบบสามารถเชื่อมโยงกับโปรแกรมจำลองการบินที่มีอยู่ในท้องตลาด และอุปกรณ์ควบคุมการทำงานของเครื่องจำลองการบินได้ตามจุดประสงค์ที่วางไว้ และสามารถนำไปเป็นต้นแบบเพื่อสร้างเครื่องจำลองการบินจริงได้อีกด้วย จากการทดลองพบว่าการควบคุมการเคลื่อนที่ของเครื่องจำลองการบินต้นแบบนั้นสามารถทำงานได้อย่างมีประสิทธิภาพ โดยมีความคลาดเคลื่อนสูงสุดของการควบคุมมุมองศาการเคลื่อนที่ของเครื่องจำลองการบินคือ 1-2 องศา ของการควบคุมการเคลื่อนที่ทั้ง Pitch และ Roll และมีขีดจำกัดในการเคลื่อนที่แบบ Pitch คือ  $\pm 15^\circ$  การเคลื่อนที่แบบ Roll คือ  $\pm 20^\circ$  และการเคลื่อนที่เคลื่อนลงในแนวแกน  $\pm 20$  มิลลิเมตร

#### 5.2 ข้อเสนอแนะ

- งานวิจัยนี้เป็นงานวิจัยที่ใช้ระบบควบคุมแบบเปิด ดังนั้นจึงอาจมีค่าความคลาดเคลื่อนเกิดขึ้น หากต้องการให้มีความถูกต้องเพิ่มขึ้น สามารถใช้ระบบควบคุมแบบปิด โดยทำการติดตั้งเซนเซอร์วัดท่าทางการเคลื่อนที่เพิ่มเข้าไปในระบบเพื่อป้อนสัญญาณการตรวจสอบท่าทาง และทำการแก้ไข ให้ได้ท่าทางที่ถูกต้องมากยิ่งขึ้น
- การสร้างเครื่องจำลองควรสำรวจอุปกรณ์ที่มีในท้องตลาดว่ามีขนาดตามที่ ได้ทำการออกแบบไว้หรือไม่ เพื่อให้เครื่องจำลองการบินนั้นมีประสิทธิภาพสูงสุด
- ควรนำเครื่องจำลองต้นแบบนี้ ไปพัฒนาสร้างเครื่องจำลองการบินที่สามารถบรรทุกภาระกรรม (นักบิน และ อุปกรณ์การบิน) ได้จริง

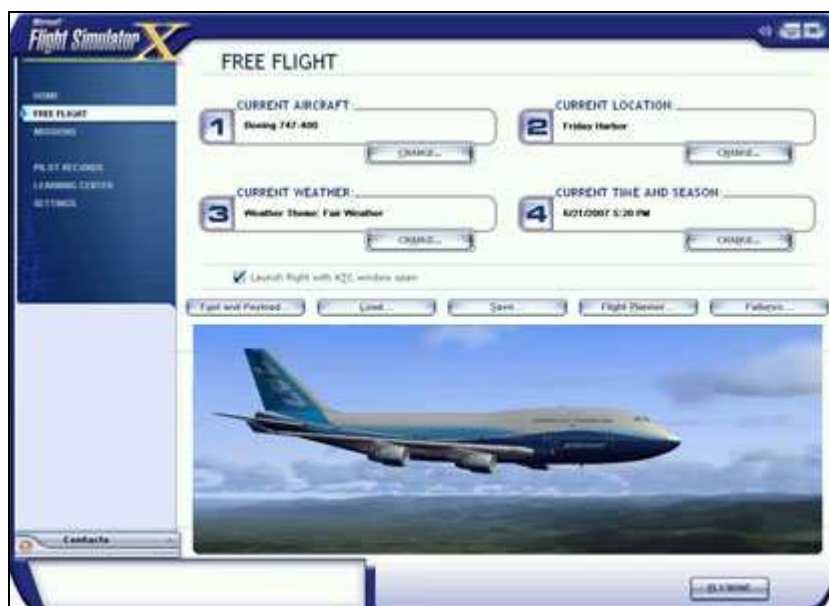
## รายการอ้างอิง

- กนต์ธร ชำนิประศาสน์ การวัดเชิงกลด้วย LabVIEW. เอกสารประกอบการสอน สาขาวิชา  
วิศวกรรมเครื่องกล สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี  
กฤษดา จันทร์แดง (2551). การจำลองพลศาสตร์ยานยนต์ร่วมกับยานพาหนะจริงแบบย่อส่วน  
(Collaboration of Vehicle Dynamic Simulation with Scaling Vehicle). วิทยานิพนธ์ปริญญา  
วิศวกรรมศาสตรมหาบัณฑิต จุฬาลงกรณ์มหาวิทยาลัย
- สุทธิวัฒน์ ประชาชนะชัย (2548). การจำลองการทำงานของแผ่นเคลื่อนไหว 6 องศาอิสระที่  
ขับเคลื่อนด้วยระบบไฮดรอลิก. วิทยานิพนธ์ปริญญาวิศวกรรมศาสตรมหาบัณฑิต สถาบัน  
เทคโนโลยีพระจอมเกล้าพระนครเหนือ.
- Advani, S. K.; Nahon, M. A.; Haeck, N.; Albronda, J.(1999). Optimization of six-degrees -of -  
freedom motion systems for flight simulators. **Journal of Aircraft**.36(5), 819 – 826.  
September –October.
- Amery, J. G.; Streid, H. (1999). Flight simulation visual requirements and a new display system.  
International Society for Optical Engineering (SP IE).3690, 356 - 367 . **Cockpit Displays  
VI: Displays for Defense Applications**.
- 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 ),** 20
- Page, R.L. (2000). Brief History of Flight Simulation. **Proceedings of SimTecT 2000  
Conference, Sydney**.
- Pouliot, N.A., Gosselin, C.M. and Nahon, M.A., Motion Simulation Capabilities of Three-  
Degree-of-Freedom Flight Simulators, **Journal of Aircraft**, January 1998, 35(1): 9-17.
- Rekdalsbakken, W. Design and Application of a Motion Platform in Three Degrees of Freedom.  
**SIMS 2005 46<sup>th</sup> Conference on Simulation and Modelin**.
- Rubio, E., Hernandez, L. Aracil, R. and Saltaren, R. (2009). Implementation of Decoupled  
Model-Based Controller in a 2-DOF Pneumatic Platform used in Low -Cost Driving  
Simulators, **Robotics and Automotive Mechanics Conference CERMA '09, Mexico,**  
Sept. 22-25.



ภาคผนวก ก

รายละเอียดโปรแกรมจำลองการบิน



รูปที่ ก.1 โหมดการบินเดี่ยวสำหรับผู้ที่ต้องการชมทิวทัศน์สำหรับโปรแกรม Flight Simulator X



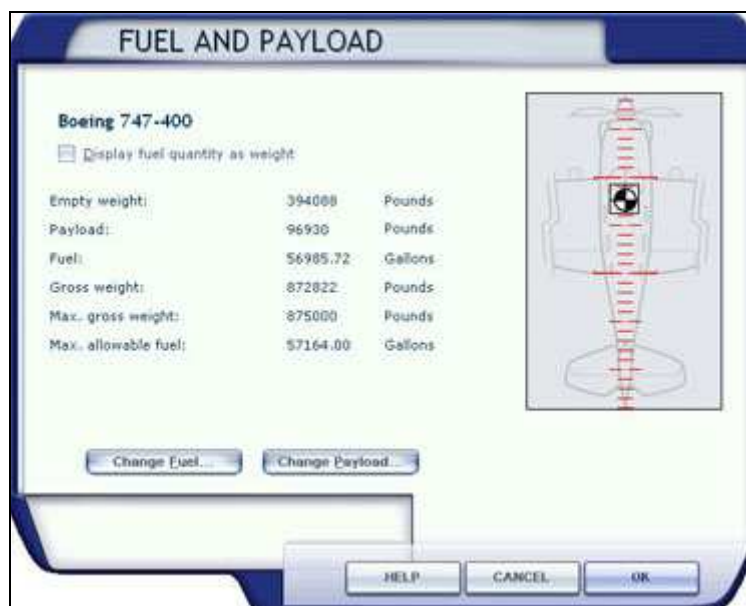
รูปที่ ก.2 หน้าต่างการเลือกอากาศยานที่ต้องการสำหรับโปรแกรม Flight Simulator X



รูปที่ ก.3 หน้าต่างกำหนดแผนการบินที่เราต้องการสำหรับโปรแกรม Flight Simulator X



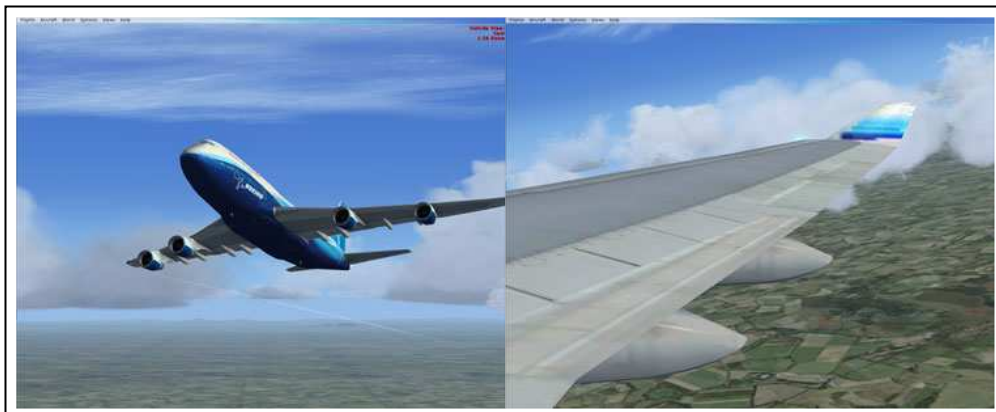
รูปที่ ก.4 หน้าต่างสภาพอากาศจริงของพื้นที่ที่เราการบินสำหรับโปรแกรม Flight Simulator X



รูปที่ ก.5 หน้าต่างกำหนดเชื้อเพลิงในการใช้งานและน้ำหนักการบรรทุก ซึ่งมีผลต่อระยะทางและการทรงตัวสำหรับโปรแกรม Flight Simulator X



รูปที่ ก.6 มุมมองห้องนักบิน ผู้ใช้สามารถควบคุมแผงควบคุมได้ทุกอย่างสำหรับโปรแกรม Flight Simulator X



รูปที่ ก.7 มุมมองภายนอกอากาศยานสำหรับโปรแกรม Flight Simulator X



รูปที่ ก.8 ระบบหลายหน้าต่างทำให้เราสามารถเห็นสิ่งต่างๆในเวลาเดียวกันสำหรับโปรแกรม Flight Simulator X



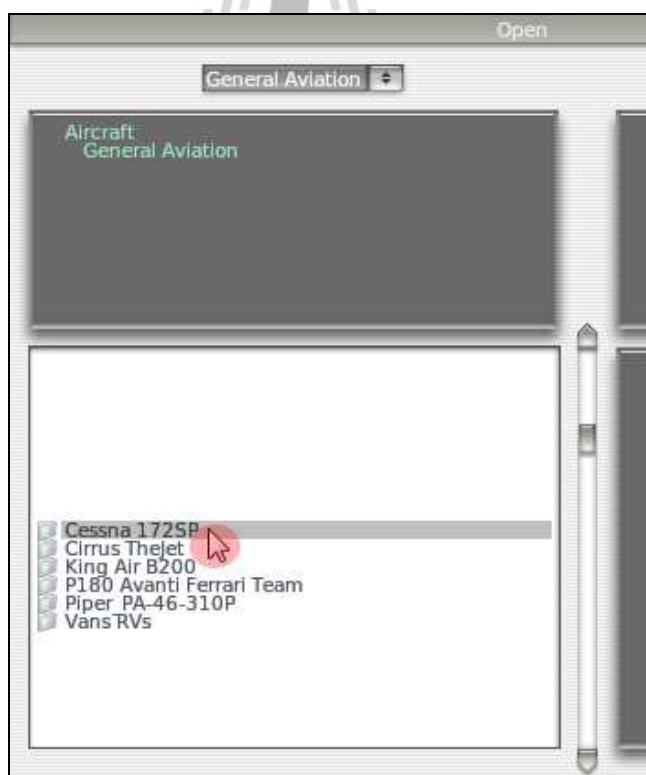
รูปที่ ก.9 ระบบนำร่องอากาศยาน (GPS)สำหรับโปรแกรม Flight Simulator X



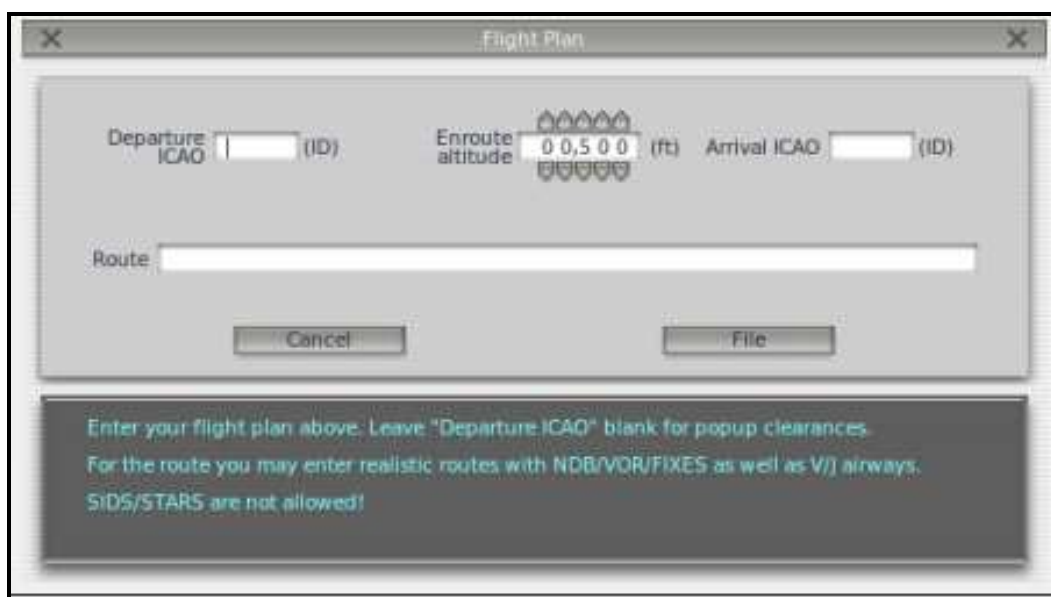
รูปที่ ก.10 หมวดการเรียนรู้เกี่ยวกับเครื่องบินสำหรับโปรแกรม Flight Simulator X



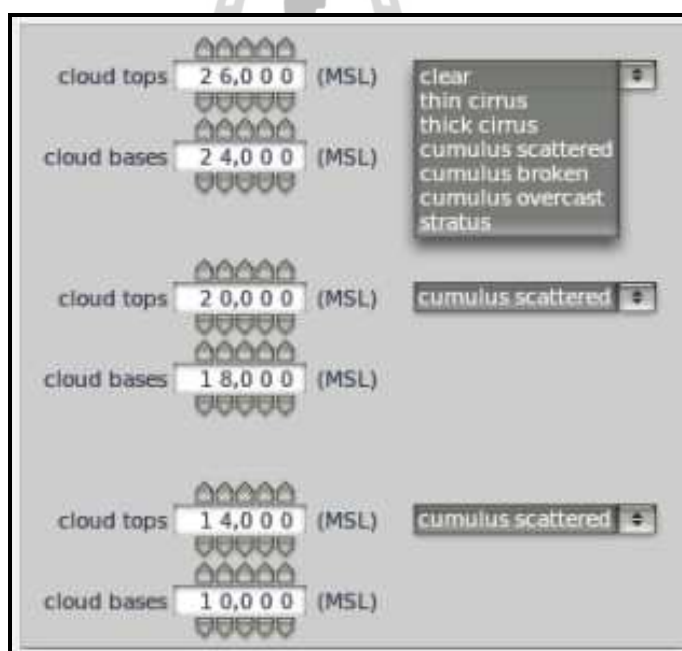
รูปที่ ก.11 โหมดการบินเดี่ยวสำหรับผู้ที่ต้องการชมทิวทัศน์สำหรับโปรแกรม X-Plane



รูปที่ ก.12 หน้าต่างการเลือกอากาศยานที่ต้องการสำหรับโปรแกรม X-Plane



รูปที่ ก.13 หน้าต่างกำหนดแผนการบินที่เราต้องการสำหรับโปรแกรม X-Plane



รูปที่ ก.14 หน้าต่างสภาพอากาศ ณ เวลาจริงของพื้นที่ที่ทำการบินสำหรับโปรแกรม X-Plane



รูปที่ ก.15 มุมมองห้องนักบิน ผู้ใช้สามารถควบคุมแผงควบคุมได้ทุกอย่างสำหรับ โปรแกรม X-Plane



รูปที่ ก.16 มุมมองภายนอกอากาศยานสำหรับ โปรแกรม X-Plane

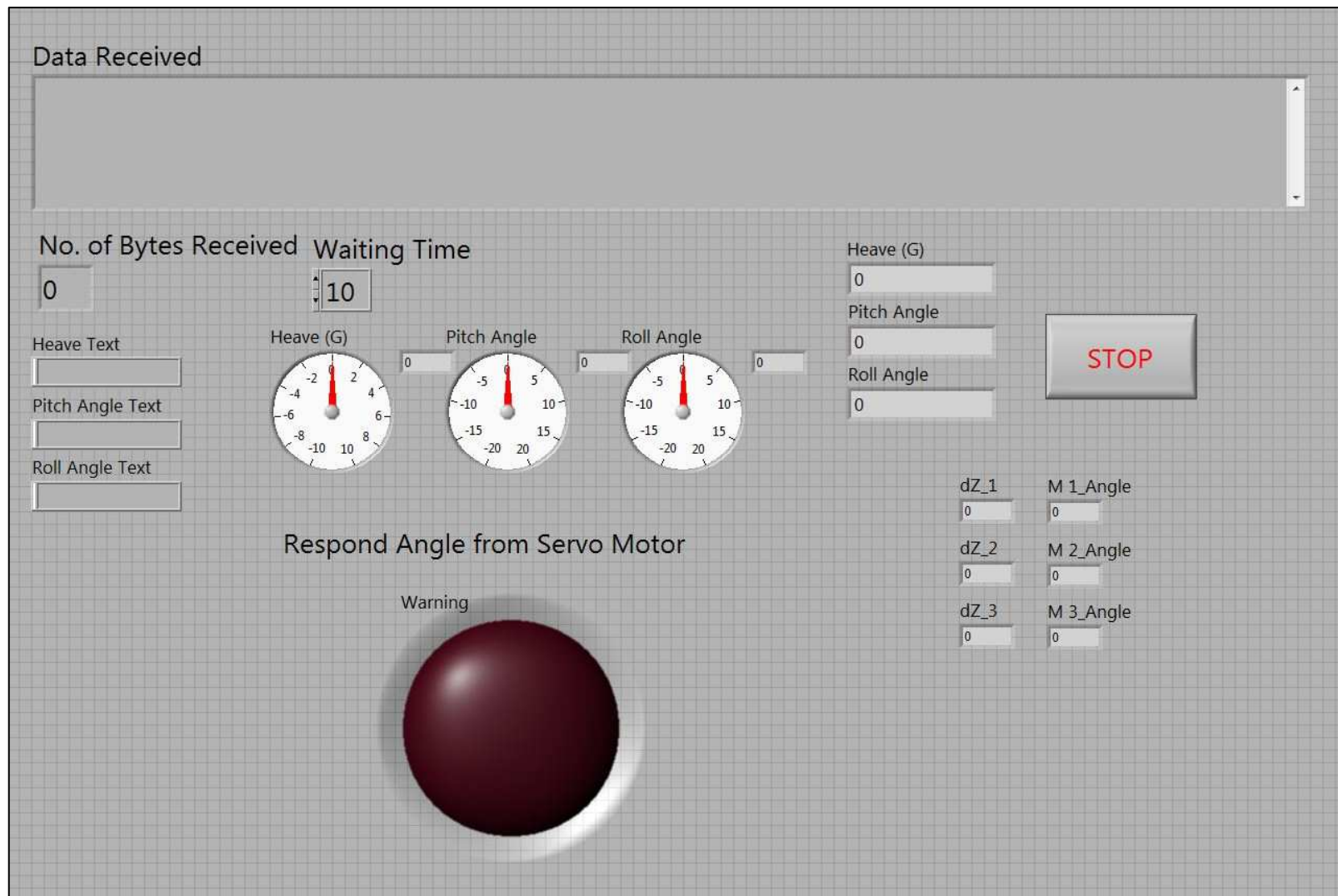


รูปที่ ก.17 ระบบนำร่องอากาศยาน (GPS) สำหรับโปรแกรม X-Plane

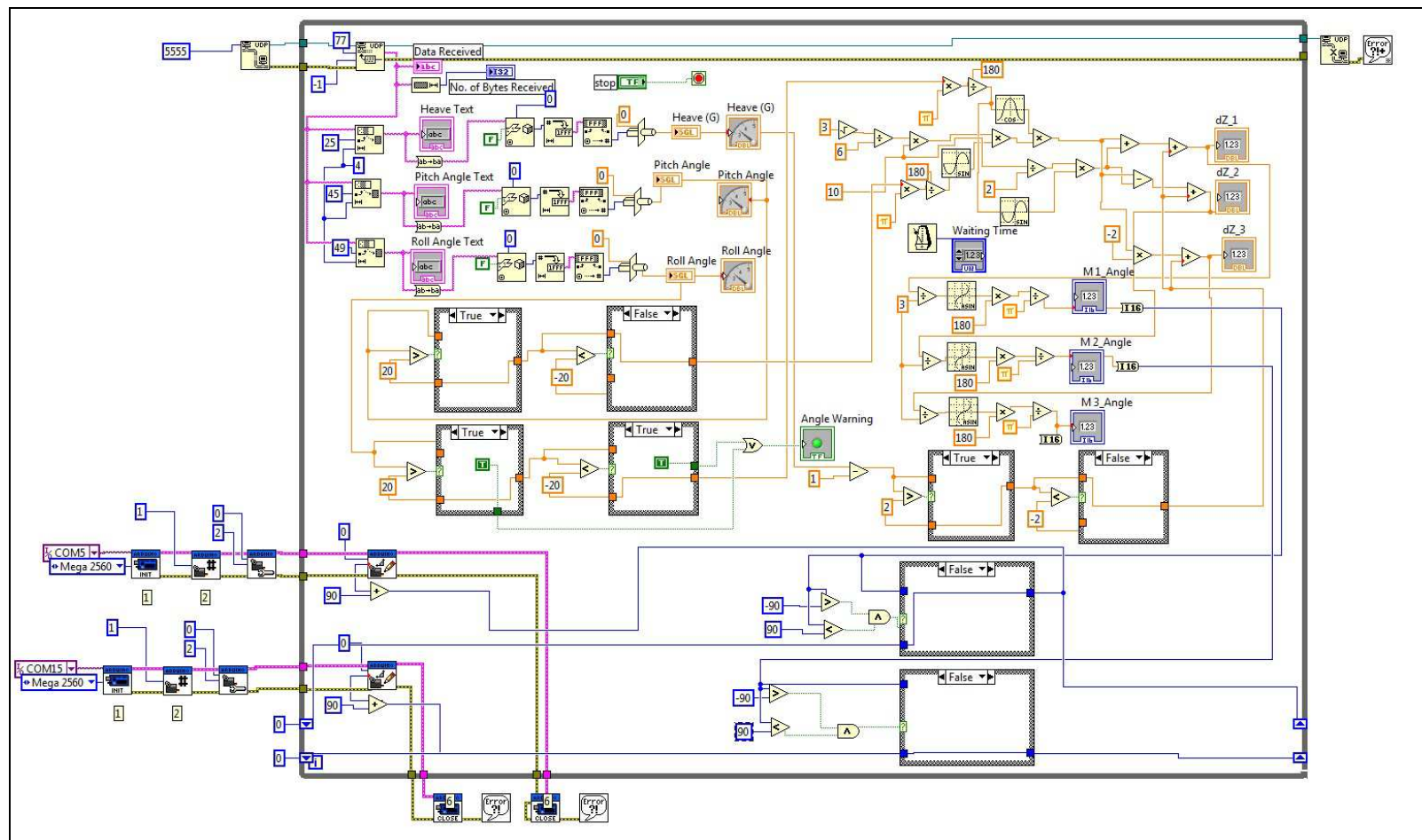
**ภาคผนวก ข**

**รายละเอียดโปรแกรม Lab VIEW**

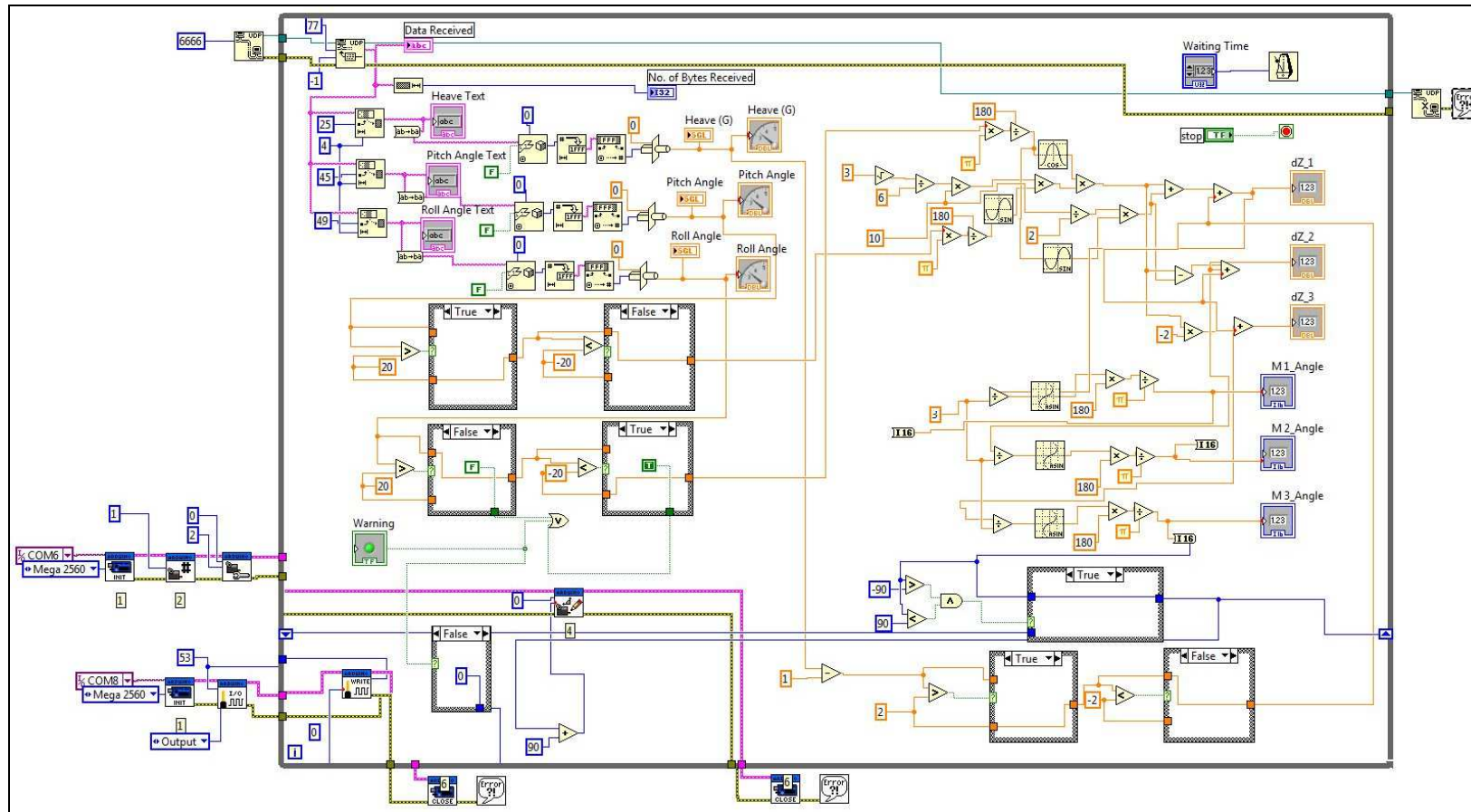




รูปที่ ข.1 หน้าจอแสดงผลของโปรแกรมควบคุมการเคลื่อนที่ของ Platform



รูปที่ ข.2 Block Diagram โปรแกรมการเคลื่อนที่ของ Platform 1



รูปที่ ข.3 Block Diagram โปรแกรมการเคลื่อนที่ของ Platform 2

ภาคผนวก ค

บทความทางวิชาการที่ได้รับการตีพิมพ์เผยแพร่



## รายชื่อบทความที่ได้รับการตีพิมพ์เผยแพร่ในระหว่างการศึกษา

Thawanrat, Th., Jittima, V., and Chamniprasart, K. (2014). **Prototyping of Aircraft Flight Simulator Platform using Linear Actuators.** The 8<sup>th</sup> South East Asian Technical University Consortium (SEATUC) Symposium. March 4-5, 2014. Johor Bahru, Malaysia. 4 pp.



## PROTOTYPING OF AIRCRAFT FLIGHT SIMULATOR PLATFORM USING LINEAR ACTUATORS

**Thawanrat Thongpan, Jittima Warakul and Kontorn Chamniprasart1**  
**School of Mechanical Engineering, Suranaree University of Technology, Nakhon**  
**Ratchasima, Thailand**

**thawanrat\_th@hotmail.com, jittima@sut.ac.th and kontorn@sut.ac.th**

### Abstract

This research is aimed to build a prototype of an airplane flight simulator platform. This prototype platform will be used as a foundation of the complete flight simulator, including cockpit of training aircraft, image screen, control panel, instrument panel and other accessories of the simulator. The Microsoft Flight Simulator (FSX) is used as flight simulator program because it has ability to communicate and transfer the necessary data of simulated aircraft to the controller. The platform is controlled to move along with the simulated plane in the program in three directions that are roll, pitch and vertical shock of the aircraft. The motion of platform is controlled by the moment of three linear actuators. These three linear actuators must be controlled simultaneously to make the platform in the same position and acceleration of the plane in the computer. This flight simulator is intended to use in the class room for aeronautical engineering student at Suranaree University of Technology.

**Keywords:** flight simulator platform, linear actuator control

### INTRODUCTION

Flight simulation is widely used for the training of pilots by various organizations. Flight simulator operators typically benefit for a reduction in pilot training time and a reduction in training costs. For the training of emergency procedures, flight simulators allow an unparalleled level of convenience and safety. Additionally, flight simulators are ideally suited for the training of instrument flight rules (IFR) and the conduct of recurrency checks. Flight simulators also reduce the risk of aircraft training accidents as a large portion of training can be conducted in the simulated environment [1]

Despite their numerous advantages, typical modern full motion simulators use electrically driven six degree-of freedom (DOF) motion platforms with fully functional cockpits and retail for several million baht. Granting the availability of financial resources, these flight simulators can offer incredible realism and fidelity. However, for the budget conscious user, it is interesting to investigate the use 3 DOF motions platform model as a means to generate motion for a flight simulator.[2] It is also of interested to study whether a realistic flight simulator can be designed and implemented using low cost and commercially available components and software. It has long been suggested in the literature that 3 DOF motion platforms are capable of producing motion simulation quality, compared with that of 6 DOF motion platforms [3]. Nowadays, the availability of high quality commercial simulation products such as controls, visual displays, and sound systems allows the design of low cost systems which are capable of acceptable levels of fidelity for the training of pilots.[4]



Figure 1 Full motion simulators

This paper discusses how 3 DOF motion platform model was used to create a flight simulator with commercially available components. Then, the results of a pilot evaluation exercise aimed for studying the performance of several aspects of the system are discussed.

### MOTION PLATFORM

The 3 DOF motion platform used in this study is the centerpiece of the flight simulator. RMC's 3 DOF motion platform, shown in Figure 2, is capable of executing movements or roll, pitch and heave. The motion is generated by three linear actuator in a triangular arrangement connected to the upper platform frame. The length, velocity, and acceleration of each actuator are independently controlled. The platform can be positioned in space and generate any combination of roll, pitch and heave motions.

Table 1 Three DOF Motion Platform Limits

|              | Roll                     | Pitch                    | Heave                    |
|--------------|--------------------------|--------------------------|--------------------------|
| Displacement | $\pm 25^\circ$           | $\pm 30^\circ$           | $\pm 0.20m$              |
| Velocity     | $30^\circ \cdot s^{-1}$  | $30^\circ \cdot s^{-1}$  | $\pm 0.45m \cdot s^{-1}$ |
| Acceleration | $350^\circ \cdot s^{-2}$ | $350^\circ \cdot s^{-2}$ | $7.85m \cdot s^{-2}$     |

The motion platform is controlled using a networked motion controller and the lab VIEW program. The motion platform displacement, velocity and acceleration limits, which determine the motion platform motion envelope, are given in Table 1.

### FLIGHT SIMULATOR COMPONENTS

The challenge of flight simulation requires the integration of several subsystems to replicate the sensations felt in a real aircraft during flight. Typically, such subsystems are comprised of computer generated images, visual display(s), sound system, motion simulation system and other enhancements such as realistic controls and cockpit instrumentation. Each subsystem contributes to the overall quality and realism of the simulation.

The selection of a virtual simulation environment is one of the important components of the simulator. In this study, two commercially available simulation environments are X-Plane 9 and Flight Simulator X (FSX) Program. In terms of physics quality, cost, and graphics quality, this virtual simulation environment was acceptable. However, Microsoft's

FSX was selected in light of its superior software development kit at the end, the ability to specify the sampling rate of the environment, advantages and disadvantages of topics as follow:

1. Multiple screens
2. Screen Location Store

Table 2 Result of study in requirement of Flight Simulator program

| Topic/Program         | X-Plane                 | FSX                          |
|-----------------------|-------------------------|------------------------------|
| Multiple Screens      | Depend on Number of PCs | Depend on Graphics Card Slot |
| Screen Location Store | No                      | Yes                          |



Figure 2 Multiple Screens Layout

Five screens as shown in Figure 1: two front screens as an extended front view (eye-infinity), two side screens as a left and right view and the last screen as an instrument panel view.

### Screen Location Store

The program that can learn the location of any windows FSX is called Panel Store. This program is working by save the location (pixel) and size of the windows. It can help the users for easy open and configuration.

### LabVIEW PROGRAM

LabVIEW program is used to interface between Flight Simulator X and Linear actuator by using UDP VI and function.

Use the UDP Open function (Figure 3) to open a UDP socket on a port. The number of simultaneously open UDP ports depends on the operating system. The UDP Open function returns a network connection refnum that uniquely identifies the UDP socket. Use this connection refnum to refer to this socket in subsequent VI calls.

Use the UDP Write function (Figure 4) to send data to a destination, and use the UDP Read function (Figure 5) to read that data. Each write operation requires a destination address and port. Each read operation contains the source address and port. UDP preserves the datagram bytes that you specified for each command you send. [5]

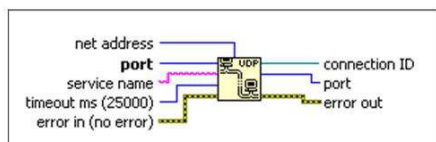


Figure 3 UDP Open function

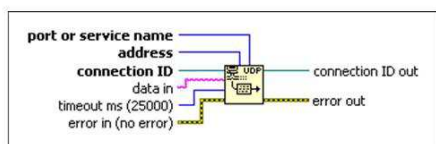


Figure 4 UDP Write function

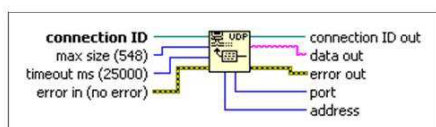


Figure 5 UDP Read function

The FSX and LabVIEW are connected by the UDP Protocol. The flights data row, pitch, and yaw angle are sent to LabVIEW program by the UDP protocol as the data connection. This communication is used the IP Address of the Computer and port to transfer the datagram. Datagram is the data which is in the form of string of integer number (16 bit).

#### LINEAR ACTUATOR

The same actuator mounting brackets used on air cylinders, such as foot, turn-on and clevis brackets, are available as options. Three types of motor installation methods: coupling, built-in (direct coupling), and reversing, are selected. An optional sensor is available for checking, if home return completes without fail. Show in Figure 6



Figure 6 Linear actuator [6]

#### CONCLUSION

A prototype of the airplane flight simulator platform was presented in this paper. The system of prototype consisted of two main parts: the motion platform, and the flight simulator. The platform movement was controlled in three directions: roll, pitch, and vertical shock, using a networked motion controller and the LabVIEW program. The Microsoft Flight Simulator (FSX) was used to perform the virtual flight simulations. The three degree of freedom motion platform was combined with the Microsoft Flight Simulator (FSX) using the UDP protocol. This integrated flight simulator is expected to use as a tool in the class for School of Mechanical Engineering (Aeronautical Engineering), Suranaree University of Technology.

#### ACKNOWLEDGEMENTS

This research was supported by funds from the Innovation and Inventions scholarship Suranaree University of Technology .

#### REFERENCES

- [1] Page, R.L., "Brief History of Flight Simulation", Proceedings of SimTecT2000 Conference, Sydney, 2000
- [2] E. Rubio, L. Hernandez, R. Aracil, and R. Saltaren, "Implementation of Decoupled Model-Based Controller in a 2-DOF Pneumatic Platform used in Low -Cost Driving Simulators", presented at the Electronics, Robotics and Automotive Mechanics Conference CERMA '09, Mexico, Sept. 22-25, 2009.
- [3] Pouliot, N.A., Gosselin, C.M., Nahon, M.A., "Motion Simulation Capabilities of Three-Degree-of-Freedom Flight Simulators", Journal of Aircraft, Vol. 35, No. 1, January 1998, pp. 9-17[2]

OS Number C1-02The 8<sup>th</sup> SEATUC Symposium | 4-5 March 2014

- [4] 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 ), 20
- [5] National Instrument , NI. UDP VI and Functions , Available:[http://zone.ni.com/reference/en-XX/help/371361K-01/lvcomm/udp\\_vi\\_descriptions/](http://zone.ni.com/reference/en-XX/help/371361K-01/lvcomm/udp_vi_descriptions/). June 15 2012.
- [6] Intelligentactuator , IAI. The Electromechanical Alternative to Pneumatic Cylinders. Available: <http://www.intelligentactuator.com/robot-cylinders/>. April 5 2012.



**Thawanrat Thongpan** received the B.E. (2010) degree in aeronautical engineering from Suranaree University of Technology



**Jittima Varagul** received the B.E.(2007) degree in mechanical engineering , M.E. (2010) degree in mechanical engineering from Suranaree University of Technology



**Kontorn Chamnirasart** received the B.S. (1980) degree in mechanical engineering from First Class Horner, Royal Thai Air Force Academy, Thailand, M.Sc. (1987) degree in mechanical engineering from University of Pittsburgh, USA, and Ph.D. (1992) degree in mechanical engineering from University of Pittsburgh, USA

## ประวัติผู้เขียน

นางสาววัลรัตน์ ทองปัน เกิดเมื่อวันที่ 1 เมษายน 2531 ที่อำเภอพิบูลมังสาหาร จังหวัดอุบลราชธานี เริ่มการศึกษาระดับอนุบาลถึงระดับประถมศึกษาปีที่ 6 ที่โรงเรียนพิบูลมังสาหารวิภาควิทยาการ ระดับชั้นมัธยมศึกษาปีที่ 1-3 ที่โรงเรียนเบ็ญจมะมหาราช และระดับชั้นมัธยมศึกษาปีที่ 4-6 ที่โรงเรียนแก่นนครวิทยาลัย (โครงการพัฒนาและส่งเสริมผู้มีความสามารถพิเศษทางวิทยาศาสตร์และเทคโนโลยี) สำเร็จการศึกษาวិชากรรมศาสตรบัณฑิต (วิศวกรรมอากาศยาน) สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี จังหวัดนครราชสีมา เมื่อ พ.ศ. 2553 และได้ศึกษาต่อระดับวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมเครื่องกล ขณะศึกษาได้เป็นผู้สอนปฏิบัติการของสาขาวิชาเครื่องกล จำนวน 14 รายวิชาได้แก่

- (1) เขียนแบบวิศวกรรม 1 และ 2
- (2) การเขียนแบบทางกล
- (3) ปฏิบัติการวิศวกรรมเครื่องกล 1 - 3
- (4) ปฏิบัติการวิศวกรรมการผลิต 1 - 3
- (5) ปฏิบัติการวิศวกรรมอากาศยาน 1 - 3
- (6) MATLAB
- (7) พื้นฐานการฝึกบินด้วยเครื่องจำลองบิน

ระหว่างศึกษาระดับปริญญาโทได้นำเสนอผลงานวิชาการ เรื่อง Prototyping of Aircraft Flight Simulator Platform Using Linear Actuators ในการประชุมนานาชาติ South East Asian Technical University Consortium (SEATUC) Symposium 2014 ณ เมือง Johor Bahru ประเทศมาเลเซีย

รวมทั้งเป็นวิทยากรอบรมพนักงานของบริษัท เบนซ์มาร์ค อิเลคทรอนิกส์ (ประเทศไทย) จังหวัดนครราชสีมา ในหลักสูตร Microsoft Excel for Engineer