

ระบบติดตามตำแหน่งการเคลื่อนที่สำหรับรถจำลอง
POSITION TRACKING SYSTEM FOR CAR MODEL

โดย



T144553

ศักย์ปณชัย เกศศิชาปกรณ
SUKKPRANHACHAI GATESICHAPAKORN

อาจารย์ที่ปรึกษา
ผู้ช่วยศาสตราจารย์ ดร.โอฬาร วงศ์วิรัตน์

เลขหมู่.....144553
เลขทะเบียน.....
วัน,เดือน,ปี...25...๖๕...2559

600268206
b. 12812924
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับภาคเรียนที่ 2 ปีการศึกษา 2557 ขออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบติดตามตำแหน่งการเคลื่อนที่สำหรับรถจำลอง
POSITION TRACKING SYSTEM FOR CAR MODEL

โดย

ศักย์ปณชัย เกศสิชาปกรณ

SUKKPRANHACHAI GATESICHAPAKORN

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตราจารย์ ดร.โอฬาร วงศ์วิรัตน์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 2 ปีการศึกษา 2557

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

POSITION TRACKING SYSTEM FOR CAR MODEL

SUKKPRANHACHAI GATESICHAPAKORN



**A PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY
FACULTY OF INFORMATION TECNOLOGY
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

2/2014

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2015

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบรับรองปริญญาโท ประจำปีการศึกษา 2557

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบติดตามตำแหน่งการเคลื่อนที่สำหรับรถจำลอง

POSITION TRACKING SYSTEM FOR CAR MODEL

ผู้จัดทำ

นาย ศักย์ปณชัย เกศลิขปกรณ์ รหัสประจำตัว 54070089



.....อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร.โอฬาร วงศ์วิรัตน์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงการ	ระบบติดตามตำแหน่งการเคลื่อนที่สำหรับรถจำลอง
นักศึกษา	นายศักร์ปณชัย เกศศิชาปกรณีย์ รหัสประจำตัว 54070089
ปริญญา	วิทยาศาสตรบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
ปีการศึกษา	2557
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.โอฬาร วงศ์วิรัตน์

บทคัดย่อ

ปัจจุบัน การติดตามตำแหน่งการเคลื่อนที่โดยส่วนมากจะใช้ระบบกำหนดตำแหน่งบนโลก หรือ GPS (Global Positioning System) ซึ่งใช้สัญญาณจากดาวเทียมในการอ้างอิงตำแหน่ง อุปกรณ์ GPS สำหรับการใช้งานโดยทั่วไปจะให้ค่าความละเอียดต่ำ และไม่สามารถใช้ในพื้นที่อับสัญญาณได้ ดังนั้น GPS จึงไม่เหมาะสำหรับการติดตามตำแหน่งในงานที่ต้องการความละเอียดสูงรวมทั้งงานที่ใช้ภายในอาคาร

ระบบนำร่องเฉื่อย หรือ INS (Inertial Navigation System) คือ ระบบการระบุตำแหน่งเพื่อการนำร่องด้วยการคำนวณค่าตำแหน่งจากข้อมูลที่ได้รับจากหน่วยวัดแรงเฉื่อย หรือ IMU (Inertial Measurement Unit) ซึ่งประกอบด้วยมาตรความเร่ง (Accelerometers) และไจโรสโคป (Gyroscopes) โดยข้อดีของ INS คือไม่ใช้การอ้างอิงใด ๆ จากภายนอกและมีความแม่นยำสูงในช่วงเวลาสั้น ๆ โดยค่าความแม่นยำจะขึ้นกับคุณภาพของ IMU ที่ใช้ รวมทั้งสามารถใช้ได้ทั้งภายในและภายนอกอาคาร อย่างไรก็ตามข้อเสียของ INS คือ ค่าความผิดพลาดสะสมระหว่างการทำงาน ทำให้ต้องอาศัยเทคนิคการปรับแก้ความคลาดเคลื่อนเข้ามาช่วย

โครงการนี้เป็นการพัฒนาระบบติดตามตำแหน่งการเคลื่อนที่สำหรับรถจำลอง โดยอาศัยข้อมูลการเคลื่อนที่ซึ่งคำนวณจากค่าความเร่งร่วมกับข้อมูลจากระบบอ้างอิงทิศทางและการวางตัว หรือ AHRS (Attitude and Heading Reference System) ที่ได้จาก IMU เข้ามาใช้ในการปรับแก้ความผิดพลาดสะสมของตำแหน่ง จึงทำให้สามารถระบุตำแหน่งและทิศทางการเคลื่อนที่ได้ การทำงานของระบบจะใช้ข้อมูลความเร่งและทิศทางวางตัวจาก IMU ส่งไปยังคอมพิวเตอร์ผ่านเครือข่ายไร้สายเพื่อใช้ในการประมวลผล โดยระบบสามารถแสดงผลตำแหน่งและทิศทางในการเคลื่อนที่ของรถจำลองที่ติดตั้งชุดอุปกรณ์ผ่าน โปรแกรมประยุกต์ที่พัฒนาขึ้นได้

Project Title	Position tracking system for car model	
Student	Mr. Sukkpranhachai Gatesichapakorn	Student ID 54070089
Degree	Bachelor of Science	
Program	Information Technology	
Academic Year	2014	
Advisor	Asst. Prof. Dr. Olarn Wongwirat	

ABSTRACT

Currently, a GPS (Global Positioning System) is the most widely used for position tracking by acquiring signals from satellites for referring positions. Typical GPS devices used have low precision and cannot be used in shading area. Therefore, using GPS is not suitable for high precision tracking and indoor environment.

An INS (Inertial Navigation System) is a position tracking system used for navigation by calculating data receiving from an IMU (Inertial Measurement Unit), which comprising accelerometers and gyroscopes. The advantage of INS is not using external references and provides high precision in a short period, which the accuracy is relied on the IMU quality being used. Also, it can be used for both indoor and outdoor environment. However, the disadvantage of INS is an error accumulation while operation. Therefore, it requires the error correction technique for improvement.

This project is a development of position tracking system for a car model by using motion data calculated by acceleration, including attitude and orientation data from the AHRS (Attitude and Heading Reference System), acquiring from the IMU to reduce the accumulated position errors. By this method, the system can identify the position and movement direction. The system is performed by sending the acceleration and orientation data receiving from the IMU to process via the computer through a wireless network. Therefore, the system can display tracking positions and movement directions of the car model that installs the setting devices via the application program being developed.

กิตติกรรมประกาศ

โครงการนี้สำเร็จลุล่วงได้ด้วยความกรุณาจาก ผู้ช่วยศาสตราจารย์ ดร.โอฬาร วงศ์วิรัตน์ อาจารย์ที่ปรึกษาผู้ให้ความรู้ คำแนะนำ และแนวคิดในการแก้ปัญหา รวมถึงความช่วยเหลือในด้านอื่นๆ แก่ข้าพเจ้า

ขอขอบพระคุณคณาจารย์คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่านที่ได้ประสิทธิ์ประสาทวิชาให้แก่ข้าพเจ้าตลอดระยะเวลาที่รับการศึกษา

ขอขอบคุณเพื่อน พี่ น้องทุกคนที่คอยให้กำลังใจและคอยให้ความช่วยเหลือแก่ข้าพเจ้า รวมทั้ง สมาชิกชมรมยานยนต์ และ เจ้าหน้าที่ที่คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่คอยให้ความช่วยเหลือจนโครงการนี้สำเร็จลุล่วงไปได้ด้วยดี

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณบิดา มารดาและขอขอบคุณครอบครัวที่คอยให้ความรัก ให้กำลังใจที่อบอุ่นแก่ข้าพเจ้า



สารบัญ

หน้า

บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ.....	III
สารบัญ	IV
สารบัญรูป	VI

บทที่

1. บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 เป้าหมายและวัตถุประสงค์.....	2
1.3 วิธีการดำเนินงาน	2
1.4 ขอบเขตของโครงการ	3
1.5 ประโยชน์ที่ได้รับ.....	3
2. ทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง	4
2.1 ระบบนำร่องเฉื่อย (Inertial Navigation System).....	4
2.2 หน่วยวัดแรงเฉื่อย (Inertial Measurement Unit)	8
2.3 ระบบอ้างอิงทิศทางและการวางตัว (Attitude and Heading Reference System)	11
2.4 ระบบพิกัด (Coordinate system)	12
2.5 ฮาร์ดแวร์และซอฟต์แวร์ (Hardware and Software)	17
3. การพัฒนาระบบติดตามตำแหน่ง	22
3.1 โครงสร้างของระบบติดตามตำแหน่ง (System configuration)	22
3.2 องค์ประกอบด้านฮาร์ดแวร์ (Hardware component)	24
3.3 การพัฒนาซอฟต์แวร์ในระบบ (Software implementation)	26
3.4 การรับข้อมูลและการแสดงผล (Data input and display).....	30
3.5 การปรับค่าแรงโน้มถ่วง	34
3.6 การคำนวณค่าตำแหน่งและทิศทาง	37

IV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

4. การทดสอบระบบ	41
4.1 ทดสอบการอ่านค่าความเร่ง	41
4.2 การทดสอบการระบุตำแหน่ง	48
4.3 การทดสอบการติดตามตำแหน่งในระยะสั้น	50
4.4 การทดสอบการติดตามตำแหน่งในระยะทางยาว	53
4.5 ทดสอบการติดตามตำแหน่งของรถแข่ง Formula Student	57
5. สรุปผลโครงการ	60
5.1 ปัญหาและอุปสรรค	61
5.2 ข้อเสนอแนะ	62
บรรณานุกรม	63
ภาคผนวก ก	64
โปรแกรมสำหรับการปรับค่า XBee Pro	64
ภาคผนวก ข	70
การปรับแต่ง AHRS Firmware ด้วย Arduino IDE	70
ภาคผนวก ค	74
การติดตั้งส่วนเสริมที่จำเป็นในโปรแกรม Processing 2	74
ภาคผนวก ง	76
การใช้งานโปรแกรมที่พัฒนาขึ้นในโครงการ	76
ประวัติผู้เขียน	80

สารบัญรูป

รูปที่

หน้า

2.1 การประมวลผลสัญญาณของ INS	6
2.2 9DOF MEMS IMU	8
2.3 3-Axis Accelerometers.....	9
2.4 3-Axis Gyroscope	10
2.5 ENU coordinate reference.....	11
2.6 RPY Attitude reference.....	11
2.7 9 DOF Razor IMU	17
2.8 XBee Pro S2B.....	18
2.9 Car model and equipment.....	19
2.10 Arduino IDE.....	20
2.11 Processing2	21
3.1 INS diagram	22
3.2 Data flow diagram ระหว่างอุปกรณ์.....	24
3.3 ผังการเชื่อมต่ออุปกรณ์.....	25
3.4 ชุดอุปกรณ์ในการทดสอบ.....	25
3.5 ผังการทำงานของ AHRS Razor IMU firmware ที่ทำการปรับแต่งแล้ว	26
3.6 ผังการทำงานของ โปรแกรมส่วนติดต่อ COM port	27
3.7 ผังการทำงานของ โปรแกรมส่วนแสดงผล.....	28
3.8 ผังการทำงานของส่วนประมวลผลในโปรแกรมแสดงผล	29
3.9 ข้อมูลที่อ่านได้จาก IMU ด้วย Serial monitor ในโปรแกรม Arduino IDE	30
3.10 ข้อมูลที่อ่านได้จาก IMU ในโปรแกรม Processing2	31
3.11 GUI ของโปรแกรมส่วนติดต่อ COM Port	32
3.12 GUI ของโปรแกรมส่วนแสดงผลในหน้าแรก	32
3.13 GUI ของโปรแกรมส่วนแสดงผลการติดตามตำแหน่งการเคลื่อนที่	33
3.14 การทดสอบการอ่านค่าตามแนวแกน X.....	34
3.15 ผลการทดสอบการอ่านค่าตามแนวแกน X.....	35
3.16 การทดสอบการอ่านค่าตามแนวแกน Y.....	35
3.17 ผลการทดสอบการอ่านค่าตามแนวแกน Y	36

สารบัญรูป (ต่อ)

รูปที่

หน้า

3.18 การทดสอบการอ่านค่าตามแนวแกน Z.....	36
3.19 การทดสอบการคำนวณในโปรแกรม MS Office Excel	37
3.20 การทดสอบการคำนวณในโปรแกรม Processing2	38
3.21 การทดสอบการคำนวณทิศทางรถเคลื่อนที่.....	40
4.1 ชุดอุปกรณ์การทดสอบ	41
4.2 กราฟความเร่งขณะเคลื่อนที่เป็นเส้นตรงด้วยมือ.....	42
4.3 กราฟความเร่งขณะเคลื่อนที่เป็นวิถีโค้งด้วยมือ	43
4.4 รถจำลองที่ติดตั้งชุดอุปกรณ์ทดสอบแล้ว	44
4.5 พื้นที่ทดสอบ.....	44
4.6 กราฟความเร่งขณะรถวิ่งเป็นเส้นตรง	45
4.7 กราฟความเร่งขณะรถวิ่งเป็นวงกลม.....	46
4.8 หน้าจอแสดงผลการติดตามตำแหน่งในการทดสอบ	48
4.9 การติดตามตำแหน่งระยะ 60 เซนติเมตรครั้งที่หนึ่ง.....	50
4.10 การติดตามตำแหน่งระยะ 60 เซนติเมตรครั้งที่สอง.....	51
4.11 การติดตามตำแหน่งระยะ 60 เซนติเมตรครั้งที่สาม.....	51
4.12 พื้นที่ทดสอบการเคลื่อนที่.....	53
4.13 ผลการทดสอบติดตามตำแหน่งการเคลื่อนที่ระยะ 8 เมตร	54
4.14 ผลการทดสอบติดตามตำแหน่งการเคลื่อนที่ระยะ 13 เมตร.....	55
4.15 ผลการทดสอบติดตามตำแหน่งการเคลื่อนที่เป็นวงกลม.....	56
4.16 รถแข่ง Formula Student ที่ใช้ในการทดสอบ	57
4.17 ผลการทดสอบตำแหน่งการเคลื่อนที่ในช่วงแรก	58
4.18 ผลการทดสอบตำแหน่งการเคลื่อนที่ในช่วงที่มีความเร็วสะสม.....	59

VII

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เทคโนโลยีการอ้างอิงตำแหน่งบนพื้นผิวโลกที่ใช้กันอย่างแพร่หลายในปัจจุบัน คือ ระบบระบุพิกัดตำแหน่งบนโลก หรือ Global Positioning System (GPS) ซึ่งใช้การอ้างอิงตำแหน่งด้วยการคำนวณสัญญาณที่ได้รับจากดาวเทียม GPS โดยค่าตำแหน่งที่ได้จะเป็นค่าประมาณที่ยังคงมีความคลาดเคลื่อนอยู่ ค่าความแม่นยำในการระบุตำแหน่งของ GPS จึงขึ้นอยู่กับจำนวนช่องสัญญาณ และคุณภาพของสัญญาณที่ใช้ในการคำนวณเพื่อหาค่าตำแหน่งอ้างอิงบนพื้นผิวโลก [1] การใช้งาน GPS ในการอ้างอิงตำแหน่งจะมีความแม่นยำต่ำในพื้นที่อับสัญญาณ เช่น ในอาคาร เป็นต้น ทำให้การติดตามตำแหน่งการเคลื่อนที่เป็นไปได้ยาก เพื่อให้ระบบติดตามตำแหน่งการเคลื่อนที่ของรถจำลองมีความแม่นยำสูงและสามารถใช้งานในอาคารได้ จึงควรใช้ระบบการระบุตำแหน่งที่ไม่อาศัยการอ้างอิงสัญญาณจากภายนอก โครงการนี้จึงใช้ระบบนำร่องเฉื่อย หรือ INS (Inertial Navigation System) ซึ่งเป็นระบบการระบุตำแหน่งที่ไม่ใช้การอ้างอิงสัญญาณจากภายนอกในการทำงาน จึงเหมาะกับการติดตามตำแหน่งการเคลื่อนที่ที่ต้องการความละเอียดสูงและสามารถใช้งานได้ภายในอาคารมากกว่า GPS

ระบบนำร่องเฉื่อย หรือ INS (Inertial Navigation System) คือ ระบบนำร่องที่ใช้การอ้างอิงตำแหน่งโดยการหาปริพันธ์อันดับสองของความเร่ง [1] โดย INS จะใช้ค่าความเร่ง (Acceleration) และค่าการหมุน (Rotation) ซึ่งได้มาจากหน่วยวัดแรงเฉื่อย หรือ IMU (Inertial Measurement Unit) ในการคำนวณเพื่อหาค่าตำแหน่ง [2]-[3] ข้อดีของ INS คือ ความแม่นยำของตำแหน่งการเคลื่อนที่ในระยะสั้นซึ่งขึ้นอยู่กับคุณภาพของ IMU ที่ใช้ อย่างไรก็ตาม INS มีข้อจำกัด คือ ค่าความผิดพลาดสะสมจากการคำนวณซึ่งเพิ่มขึ้นตามเวลา การลดค่าความผิดพลาดสามารถทำได้โดยใช้หลักการปรับค่าความผิดพลาดสะสมด้วยกระบวนการต่างๆ [1]-[2],[4]-[5]

โครงการนี้จึงได้ประยุกต์เทคนิคการปรับแก้ค่าความผิดพลาดสะสมโดยใช้ระบบอ้างอิงทิศทางและการวางตัว หรือ AHRS (Attitude and Heading Reference System) จาก 9DOF Razor IMU [6]-[7] โดยใช้ค่าทิศทาง (Heading) ในการอ้างอิงทิศทางเคลื่อนที่ และใช้ค่ามุมการวางตัว (Attitude) ในการปรับค่าแรงดึงดูดของโลก (Gravity) ออกจากค่าความเร่ง [5] ก่อนเริ่มคำนวณหาค่าตำแหน่งเพื่อลดความผิดพลาดที่เกิดจากแรงโน้มถ่วงของโลก และใช้เทคนิคการรีเซ็ตค่าด้วยตัวเอง (Self-Resetting) เพื่อลดค่าความผิดพลาดสะสมของระยะทาง (Distance) ในขณะหยุดนิ่ง [4]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 เป้าหมายและวัตถุประสงค์

โครงการ "ระบบติดตามตำแหน่งการเคลื่อนที่สำหรับรถจำลอง" ได้จัดทำขึ้นโดยมีเป้าหมายและวัตถุประสงค์ของการดำเนินงานดังนี้

- 1.2.1 เพื่อศึกษาระบบติดตามตำแหน่งการเคลื่อนที่โดยใช้ IMU และเทคนิคการปรับแก้ความผิดพลาดด้วย AHRS
- 1.2.2 เพื่อศึกษาวิธีการรับส่งข้อมูลจาก IMU กับเครื่องคอมพิวเตอร์ผ่านเครือข่ายไร้สาย
- 1.2.3 เพื่อพัฒนาโปรแกรมประยุกต์สำหรับการติดตามตำแหน่งการเคลื่อนที่โดยไม่ใช้การอ้างอิงจากสัญญาณภายนอก
- 1.2.4 เพื่อการพัฒนาบบติดตามตำแหน่งการเคลื่อนที่สำหรับรถจำลองด้วยใช้ข้อมูลความเร่งและการวางตัวจาก IMU ร่วมกับเทคนิค AHRS
- 1.2.5 เพื่อเป็นแนวทางในการพัฒนาระบบติดตามตำแหน่งด้วย IMU สำหรับผู้ที่สนใจ และสามารถนำความรู้ไปพัฒนาระบบอื่น ๆ ที่เกี่ยวข้องได้ในอนาคต

1.3 วิธีการดำเนินงาน

โครงการ "ระบบติดตามตำแหน่งการเคลื่อนที่สำหรับรถจำลอง" ได้วางแผนการศึกษาออกแบบ และทดสอบระบบโดยมีวิธีการดำเนินงานดังนี้

- 1.3.1 ศึกษาเทคโนโลยีการติดตามตำแหน่งที่นิยมใช้ในรถยนต์ และหุ่นยนต์เคลื่อนที่
- 1.3.2 ศึกษาเทคนิคและปัญหาต่างๆ ที่เกิดขึ้นในระบบการติดตามตำแหน่งการเคลื่อนที่
- 1.3.3 ศึกษาบบการติดตามตำแหน่งที่เหมาะสมกับโครงการ
- 1.3.4 ศึกษาเทคนิคในการแก้ปัญหาที่เกิดขึ้นในระบบการติดตามตำแหน่งที่เลือกใช้
- 1.3.5 ศึกษาอุปกรณ์ที่จะใช้ในระบบจากอุปกรณ์ที่มีอยู่และที่จะต้องจัดหาเพิ่มเติม
- 1.3.6 ออกแบบระบบติดตามตำแหน่งจากแนวทางการแก้ปัญหา และอุปกรณ์ที่เลือกใช้
- 1.3.7 พัฒนาระบบติดตามตำแหน่งการเคลื่อนที่ และ โปรแกรมประยุกต์
- 1.3.8 ทดสอบ และปรับปรุงระบบเพื่อให้เป็นไปตามเป้าหมาย
- 1.3.9 วิเคราะห์และสรุปผลการทดสอบระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขอบเขตของโครงการ

โครงการ "ระบบติดตามตำแหน่งการเคลื่อนที่สำหรับรถจำลอง" นี้จัดทำขึ้นเพื่อศึกษาและพัฒนาระบบติดตามตำแหน่งการเคลื่อนที่ที่สามารถใช้งานได้โดยไม่ใช้สัญญาณอ้างอิงจากภายนอกซึ่งมีขอบเขตของงานดังนี้

- 1.4.1 ศึกษาและพัฒนาระบบติดตามตำแหน่งการเคลื่อนที่สำหรับรถจำลองรวมถึงการปรับค่าความผิดพลาด โดยใช้ข้อมูลความเร็วรวมถึงทิศทางและการวางตัวจาก 9DOF Razor IMU ที่ทำงานด้วย Razor AHRS Firmware โดยทำการแก้ไข Razor AHRS Firmware ในส่วนของการ Calibrate และ Serial output เท่านั้น
- 1.4.2 ออกแบบระบบสำหรับติดตามตำแหน่งการเคลื่อนที่บนพื้นเรียบที่มีขนาดของพื้นที่ไม่ต่ำกว่า 3 ตารางเมตร และไม่มีการรบกวนทางสนามแม่เหล็กที่รุนแรงจนทำให้ AHRS ไม่สามารถอ้างอิงทิศทางที่ถูกต้องได้
- 1.4.3 พัฒนาชุดอุปกรณ์ซึ่งประกอบด้วย XBee Pro และ Razor IMU สำหรับติดตั้งบนรถจำลอง และพัฒนาโปรแกรมประยุกต์ด้วย โปรแกรม Processing2 ซึ่งใช้ค่าความเร็วและข้อมูลทิศทางและการวางตัวจาก Razor IMU ในการประมวลผล โดยระบบสามารถบันทึกข้อมูลและประมวลผลข้อมูลจากไฟล์ตามรูปแบบที่กำหนดได้
- 1.4.4 ทำการทดสอบ โดยใช้ค่าความเร็วขั้นต่ำที่ถูกกำหนดความีการเคลื่อนที่ที่เกิดขึ้นคือ $\pm 0.3 \text{ m/s}^2$ และ ค่าความเร็วสูงสุดที่ระบบยอมรับคือ $\pm 20 \text{ m/s}$ หากเกินกว่านี้จะถูกปรับลงเสมอ และค่าความผิดพลาดที่เกิดขึ้นจากการถูกรบกวนในสภาพแวดล้อมไม่รวมอยู่ในขอบเขตของงาน

1.5 ประโยชน์ที่ได้รับ

- 1.5.1 มีความรู้ความเข้าใจเกี่ยวกับระบบติดตามตำแหน่งการเคลื่อนที่มากขึ้น
- 1.5.2 สามารถเข้าใจการทำงานและเทคนิคการระบุตำแหน่งโดยใช้ IMU มากขึ้น
- 1.5.3 สามารถพัฒนาระบบติดตามการเคลื่อนที่โดยใช้ IMU สำหรับรถจำลองได้
- 1.5.4 สามารถนำความรู้และประสบการณ์ที่ได้รับไปประยุกต์ใช้ในการพัฒนาระบบอื่นๆ ได้ในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง

2.1 ระบบนำร่องเฉื่อย (INERTIAL NAVIGATION SYSTEM)

ระบบนำร่องเฉื่อย หรือ INS เป็นระบบการหาตำแหน่งที่มาจากการคำนวณว่า "ปริพันธ์อันดับสองของความเร่ง คือ ตำแหน่ง (The second integral of acceleration is position.)" [1] โดยค่าที่ถูกใช้ในการหาตำแหน่งคือค่าความเร่ง (Acceleration) [1]

โดยทั่วไปในระบบการทำงานของ INS จะประกอบไปด้วย หน่วยวัดแรงเฉื่อย หรือ IMU (Inertial Measurement Unit) หน่วยประมวลผล (Processor) ส่วนติดต่อ (Interface) และ หน่วยจ่ายพลังงาน (Power supply) ส่วนรายละเอียดอื่น ๆ ในการทำงานของ INS จะขึ้นอยู่กับรูปแบบการติดตั้ง และ อุปกรณ์อื่น ๆ ที่ใช้ในระบบ

Strap down Systems คือ รูปแบบหนึ่งในการติดตั้งระบบ INS โดยจะทำการติดตั้ง IMU หรือ INS ทั้งระบบลงไปบนพาหะโดยตรง ซึ่งอาจมีระบบกันกระแทก หรือ กันการสั่นสะเทือนติดตั้งเพิ่มเติม เมื่อพาหะมีการหมุน IMU หรือ INS ทั้งระบบก็จะหมุนตามด้วย ทำให้สามารถตรวจจับการหมุนของพาหะได้แม่นยำ และระบบนี้มีค่าใช้จ่ายในการติดตั้งถูกกว่าระบบอื่น ๆ แต่มีข้อเสียคือเมื่อพาหะมีการหมุนที่เร็วกว่าที่ IMU จะตรวจจับได้ จะเกิดความผิดพลาดเกี่ยวกับค่าการหมุนของระบบ และข้อเสียที่สำคัญกว่านั้นคือ การสั่น และความไม่สมดุลของพาหะ ซึ่งส่งผลโดยตรงกับค่าแรงเฉื่อยที่กระทำกับ IMU แต่สามารถแก้ปัญหาดังกล่าวได้ด้วยกระบวนการในการปรับค่าในส่วนการประมวลผลข้อมูลภายในระบบ INS [1]

หลักการในการหาค่าตำแหน่งของ INS คือ การหาปริพันธ์อันดับสองของ ความเร่งขณะหนึ่ง (Instantaneous acceleration) โดยการหาการหาปริพันธ์ในครั้งแรกจะได้ค่าความเร็วขณะหนึ่ง (Instantaneous velocity) จากนั้นทำการหาปริพันธ์ของความเร็วอีกครั้งจะได้ค่าตำแหน่ง (Position) เมื่อนำตำแหน่งในช่วงเวลาปัจจุบันลบกับตำแหน่งในช่วงเวลาก่อนหน้าจะได้ระยะการเคลื่อนที่ (Displacement) และเมื่อนำระยะการเคลื่อนที่มารวมกันตั้งแต่เวลาที่เริ่มการทำงานจนถึงเวลาปัจจุบันจะได้ระยะทางรวม (Distance) ในการเคลื่อนที่ซึ่งใช้ในการติดตามตำแหน่งการเคลื่อนที่ได้ แต่จากการคำนวณด้วยกระบวนการดังกล่าวทำให้มีข้อเสียของระบบ INS เกิดขึ้นคือการเกิดค่าความคลาดเคลื่อนจากการคำนวณปริพันธ์ (Integration drift) ซึ่งจะเกิดขึ้นเล็กน้อยทุกครั้งที่มีการคำนวณ แต่เมื่อเวลาผ่านไปค่าเล็กน้อยที่เกิดขึ้นเป็นจำนวนมากจะส่งผลกระทบต่อมากขึ้น

จากหลักการในการหาค่าตำแหน่งของ INS จะได้สมการในการหาค่าตำแหน่งจากค่าความเร่ง ประกอบด้วยสองสมการ ได้แก่ สมการในการหาค่าความเร็ว และ สมการในการหาค่าตำแหน่ง ซึ่งทั้งคู่ต้องอ้างอิงค่าความเร็ว ณ ช่วงเวลาก่อนหน้า โดยสมการทั้งสองได้อ้างอิงมาจากการสรุปใน [3] ดังนี้

สมการในการหาค่าความเร็วจากความเร่ง คือ

$$v_i = v_{i-1} + at \quad (2.1)$$

สมการรูปสุดท้ายในการคำนวณหาค่าตำแหน่งจากความเร่ง คือ

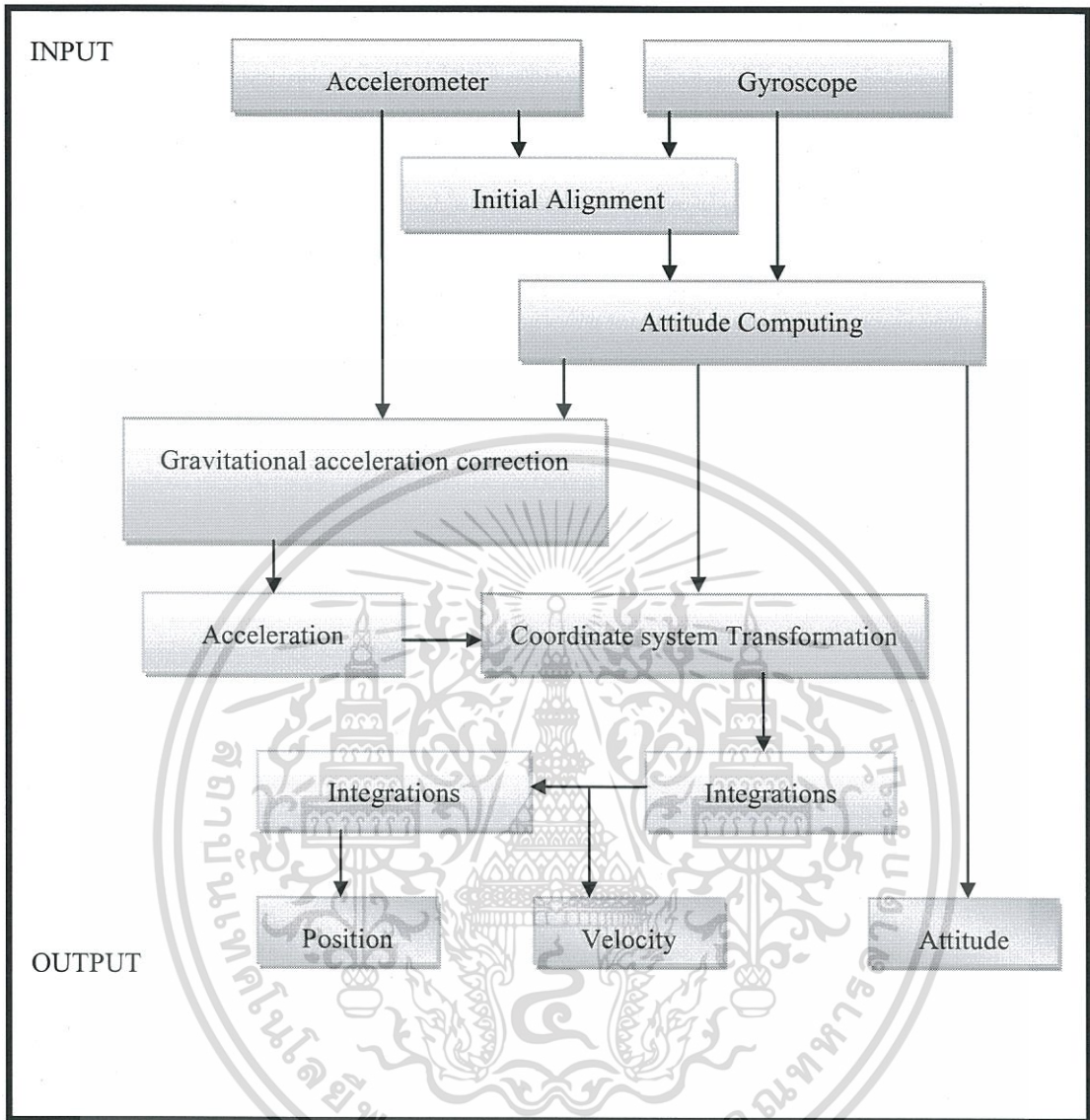
$$s_i = s_{i-1} + v_{i-1}t + \frac{1}{2}at^2 \quad (2.2)$$

โดยที่

- s_i คือ ค่าตำแหน่ง ได้จากการคำนวณในช่วงเวลาปัจจุบัน มีหน่วยเป็น m
 v_i คือ ค่าความเร็ว ได้จากการคำนวณในช่วงเวลาปัจจุบัน มีหน่วยเป็น m/s
 s_{i-1} คือ ค่าตำแหน่ง ในช่วงเวลาก่อนหน้า มีหน่วยเป็น m
 v_{i-1} คือ ค่าความเร็ว ในช่วงเวลาก่อนหน้า มีหน่วยเป็น m/s
 t คือ เวลา ระหว่างชุดข้อมูล มีหน่วยเป็น s
 a คือ ค่าความเร่ง ในช่วงเวลาปัจจุบัน มีหน่วยเป็น m/s^2
 i คือ 1, 2, 3... n

เมื่อทำการคำนวณด้วยสมการที่ (2.1) แล้ว จะได้ค่าความเร็ว v เพื่อใช้เป็นค่า v_0 ในการคำนวณด้วยสมการที่ (2.1) และ (2.2) ในช่วงเวลาถัดไป สำหรับ s_0 ก็จะเป็นไปในลักษณะเดียวกัน ถ้าทำการคำนวณในช่วงเวลาที่ 0 หรือ เริ่มต้นการทำงานครั้งแรก ค่า s_0 และ v_0 จะเป็นค่าตำแหน่ง และ ความเร็ว ของพาหะ ณ ช่วงเวลานั้น แต่หากเริ่มต้นจากการหยุดนิ่ง ค่าทั้งคู่ก็จะถูกกำหนดให้ เริ่มต้นที่ค่า 0 หรือค่าอื่น ๆ แล้วแต่กรณีและการใช้งาน

โครงสร้างการประมวลผลสัญญาณของ INS [1]



รูปที่ 2.1 การประมวลผลสัญญาณของ INS

จากรูปที่ 2.1 กระบวนการในการคำนวณหาตำแหน่งของ INS เริ่มต้นจากการนำค่าจาก เซนเซอร์มาประมวลผลค่าเริ่มต้น (Initial alignment) จากนั้นคำนวณหาการวางตัว (Attitude) เพื่อนำไปใช้ในการปรับค่าแรงโน้มถ่วงออกจากความเร่ง (Gravitational acceleration correction) และนำไปใช้ในการแปลงระบบพิกัด (Coordinate system transformation) เมื่อได้ค่าความเร่งที่อยู่ในระบบพิกัดอ้างอิงที่ต้องการแล้วจึงนำไปหาค่าปริพันธ์เพื่อให้ได้ค่าความเร็วและตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาสำหรับการอ้างอิงตำแหน่งด้วย INS คือค่าความผิดพลาดสะสมซึ่งเกิดขึ้นทุกครั้งที่ทำการคำนวณตำแหน่ง หากไม่มีการปรับค่าในทศวรรษเวลาหนึ่งแล้วจะทำให้ค่าความผิดพลาดเพิ่มขึ้นจนไม่สามารถเชื่อถือตำแหน่งที่อ้างอิงได้ ปัญหาดังกล่าวสามารถแก้ไขได้ด้วยวิธีการปรับแก้ความผิดพลาดสะสม โดยในโครงการนี้ จะใช้วิธีการปรับแก้ความผิดพลาดสะสมด้วยเทคนิคการรีเซ็ตค่าด้วยตัวเอง (Self-Resetting) [4] โดยนำแนวคิดการหาสภาวะหยุดนิ่ง (Idle state) มาใช้ กล่าวคือ หากพาหะไม่มีการเคลื่อนที่ในความเป็นจริงจะต้องไม่มีการเคลื่อนที่ที่เกิดขึ้นในการคำนวณตำแหน่ง หรือก็คือ เมื่อคำนวณตำแหน่งออกมาแล้วระยะเวลาเคลื่อนที่จะต้องเป็นศูนย์ แต่ในระบบที่ใช้ IMU ราคาถูกจะมีค่ารบกวน (Noise) อยู่มาก จึงทำให้ระบุช่วงหยุดนิ่งได้ลำบาก จึงได้คิดวิธีการที่จะใช้หาค่าประมาณการหยุดนิ่งของพาหะด้วยการรวมเวกเตอร์ (Vector) เพื่อหาขนาดของความเร่งดังสมการที่ (2.3) โดยทำการเก็บค่าไว้จำนวนหนึ่งแล้วนำมาหาค่าเฉลี่ย เมื่อค่าเฉลี่ยอยู่ในระดับต่ำกว่าที่กำหนดไว้จะถือว่าพาหะอยู่ในสภาวะหยุดนิ่ง และในสภาวะหยุดนิ่งนั้นค่าความเร็วในระบบ INS จะถูกปรับเป็นศูนย์

สมการในการรวมเวกเตอร์เพื่อหาขนาดของความเร่งที่ใช้ในการหาสภาวะหยุดนิ่ง คือ

$$CHK = \sqrt{A_x^2 + A_y^2} \quad (2.3)$$

โดยที่

- A_x คือ ค่าความเร่งแกน x ที่ปรับระบบอ้างอิงแล้ว มีหน่วยเป็น m/s^2
 A_y คือ ค่าความเร่งแกน y ที่ปรับระบบอ้างอิงแล้ว มีหน่วยเป็น m/s^2
 CHK คือ ขนาดของความเร่ง ที่ได้มาจากการรวมเวกเตอร์ มีหน่วยเป็น m/s^2

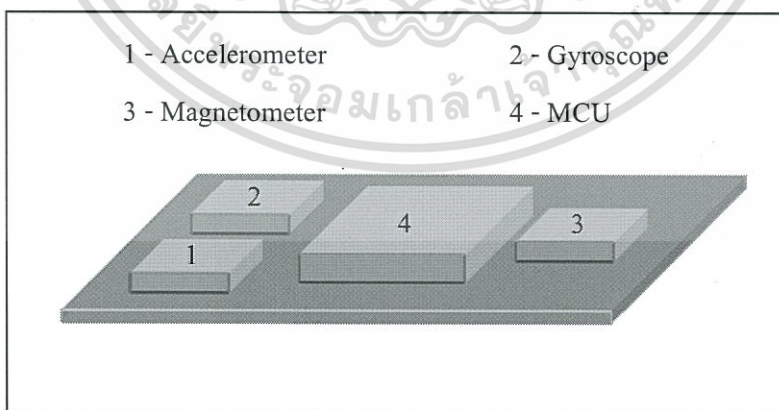
2.2 หน่วยวัดแรงเฉื่อย (INERTIAL MEASUREMENT UNIT)

2.2.1 หน่วยวัดแรงเฉื่อยระบบไฟฟ้าเครื่องกลจุลภาค (MEMS IMU)

ระบบไฟฟ้าเครื่องกลจุลภาค (Microelectromechanical systems) หรือ MEMS เป็นการรวมเอาส่วนประกอบทางวิศวกรรมไฟฟ้าและวิศวกรรมเครื่องกลเข้าไว้ด้วยกันในระดับจุลภาค ซึ่งได้ถูกพัฒนาขึ้นหลังยุค 1970 โดยเป็นเทคโนโลยีการผลิตเซนเซอร์ขนาดเล็กสำหรับการผลิตจำนวนมาก มีขนาดระหว่าง 1 ไมโครเมตรจนถึง 1 มิลลิเมตร จากการที่มีขนาดเล็กทำให้มีประสิทธิภาพในการทำงานที่ต้องการความละเอียดสูง และตอบสนองกับความเปลี่ยนแปลงได้เร็ว ส่วนข้อเสียคือมีโอกาสถูกรบกวนจากสนามแม่เหล็กไฟฟ้าได้ง่าย

หน่วยวัดแรงเฉื่อย (Inertial Measurement Unit) หรือ IMU เป็นชุดเซนเซอร์ที่โดยทั่วไปจะประกอบด้วย มาตรการเร่ง (Accelerometers) และ ใจโรสโคป (Gyroscopes) โดยอาจมีเซนเซอร์อื่นเช่น เครื่องวัดความเข้มข้นของสนามแม่เหล็ก (Magnetometer) รวมอยู่ด้วย โดยเซนเซอร์ทั้งหมดจะถูกจัดอยู่บนแผงวงจรรวมเดียวกัน ซึ่งในบางรุ่นจะมีไมโครคอนโทรลเลอร์ (MCU) สำหรับการประมวลผล และควบคุมการรับ-ส่งข้อมูลในตัวด้วย

การเรียกชื่อของ IMU มักจะขึ้นต้นด้วยจำนวนตัวแปรที่อิสระ (Degree of Freedom) ซึ่งหมายถึงจำนวนค่าที่สามารถอ่านได้จากเซนเซอร์ทั้งหมดที่อยู่ใน IMU ตัวนั้น ๆ เช่น 9DOF MEMS IMU หมายถึง IMU ที่สามารถอ่านค่าจากเซนเซอร์ต่าง ๆ ในตัวรวมกันได้ 9 ตัวแปรอิสระโดยมาจากเซนเซอร์จำนวน 3 ตัวตัวละ 3 แกน คือแกน x, y และ z โดยเซนเซอร์ทั้ง 3 ตัวคือ มาตรการเร่ง ใจโรสโคป และ เครื่องวัดความเข้มข้นของสนามแม่เหล็กตามลำดับ เป็นต้น ตัวอย่างเช่น ในรูปที่ 2.2 แสดงถึงแผงวงจรรวมของ 9DOF MEMS IMU ที่มี MCU สำหรับการประมวลผลรวมอยู่ด้วย



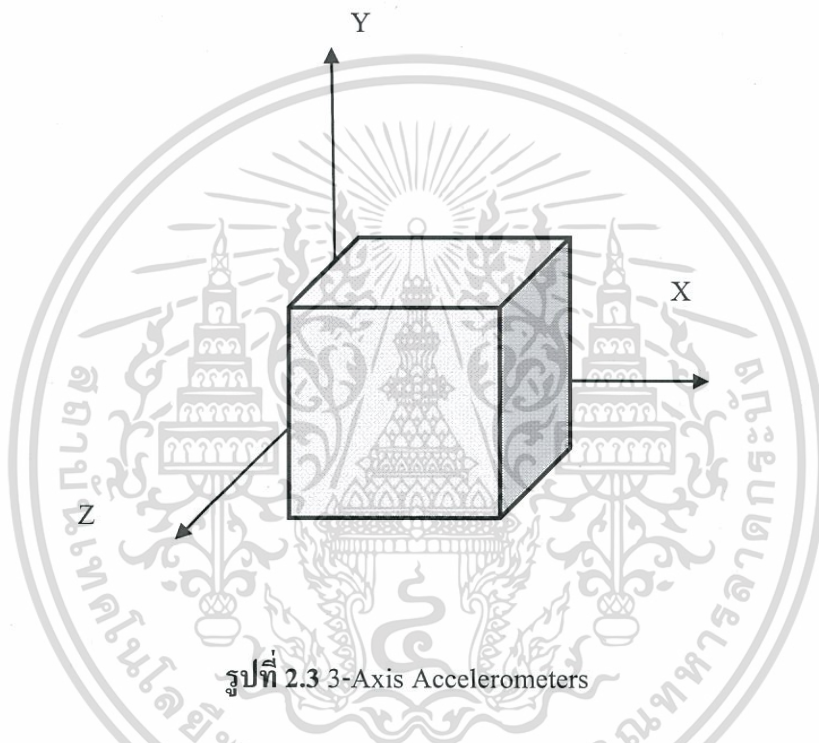
รูปที่ 2.2 9DOF MEMS IMU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 มาตรการความเร่ง (Accelerometers)

มาตรการความเร่ง (Accelerometers) เป็นอุปกรณ์ที่ใช้วัดความเร่งเฉื่อย หรือ แรงเฉื่อยในทิศทางที่ถูกกำหนดขึ้น

การทำงานของมาตรการความเร่งคือการวัดความเร่งจากแรงที่กระทำกับมวล (Proof mass) ซึ่งอยู่ภายในมาตรการความเร่ง โดยเมื่อมีแรงมากระทำกับมวลตามกฎข้อที่ 2 ของนิวตัน [1] จะเกิดการเคลื่อนที่ของมวลนั้น โดยการเคลื่อนที่ของมวลด้วยความเร็วไม่คงที่จะมีความเร่งเกิดขึ้นเสมอ ซึ่งค่าที่อ่านได้จากมาตรการความเร่งก็คือค่าความเร่งที่เกิดขึ้นนั้น



รูปที่ 2.3 3-Axis Accelerometers

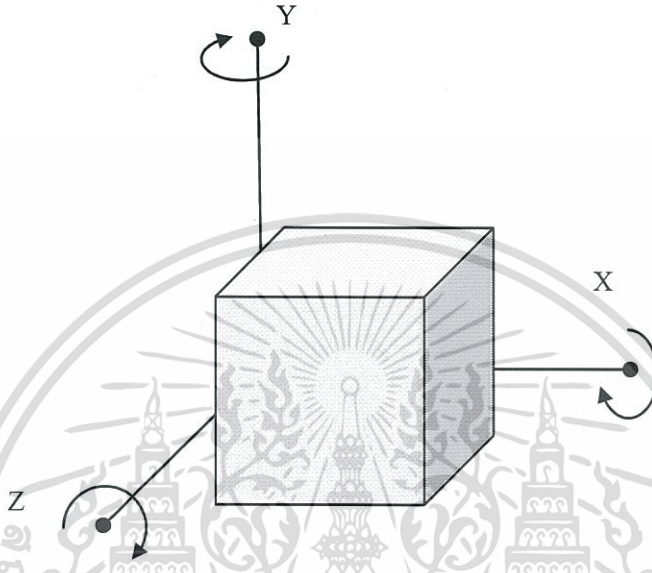
ตัวอย่างมาตรการความเร่งแบบ 3 แกนใน รูปที่ 2.3 เมื่อมีแรงมากระทำกับมาตรการความเร่งจะทำให้เกิดความเร่งตามแนวแกนที่ถูกกระทำ ซึ่งจะทำให้มวลภายในที่มองไม่เห็นมีการเคลื่อนที่ที่เกิดขึ้น โดยทราบได้จากการอ่านค่าความเร่งจากมาตรการความเร่ง

สำหรับกรณีที่มีแรงมากกว่า 1 แรงมากระทำกับมาตรการความเร่งในทิศทางใด ๆ แล้ว ค่าที่อ่านได้จากมาตรการความเร่งจะเป็นค่าจากแรงลัพธ์ (Resultant force) ที่กระทำกับมวลภายใน ณ ขณะนั้น ซึ่งแรงลัพธ์เป็นค่าแรงสุดท้ายที่จะกระทำให้มวลเกิดการเคลื่อนที่ ตัวอย่างเช่น แรงดึงดูดของโลกที่กระทำกับมวลภายในมาตรการความเร่ง จะทำให้การอ่านค่าจากมาตรการความเร่งในทิศทางของแรงดึงดูดมีค่าประมาณ 1 g หรือ 9.81 m/s^2 เสมอ ในขณะที่ไม่มีการเคลื่อนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 ไจโรสโคป (Gyroscopes)

ไจโรสโคป (Gyroscopes) เป็นอุปกรณ์ในการวัดการหมุน ซึ่งแต่ละชนิดจะใช้กระบวนการและส่วนประกอบภายในที่แตกต่างกันโดยค่าที่ได้จะเป็นอัตราการหมุน หรือองศาการหมุนขึ้นอยู่กับชนิด และหลักการของแรงเหวี่ยง (Momentum) [1] ที่ใช้ในกระบวนการการวัดค่า



รูปที่ 2.4 3-Axis Gyroscope

หลักการทำงานของไจโรสโคปโดยทั่วไปคือ การวัดค่าจากแรงที่กระทำกับมวลภายในตามทิศทางการหมุน ในรูปที่ 2.4 แสดงตัวอย่างของไจโรสโคป แบบ 3 แกน เมื่อไจโรสโคปเกิดการหมุนจะมีแรงเหวี่ยงของมวลภายในเกิดขึ้นในทิศทางของการหมุนซึ่งค่าที่อ่านได้จากไจโรสโคปจะเป็นค่าของแรงดังกล่าว แต่จะออกมาในรูปแบบใดนั้นขึ้นอยู่กับรูปแบบของไจโรสโคปนั้น

2.2.4 เครื่องวัดความเข้มข้นของสนามแม่เหล็ก (Magnetometer)

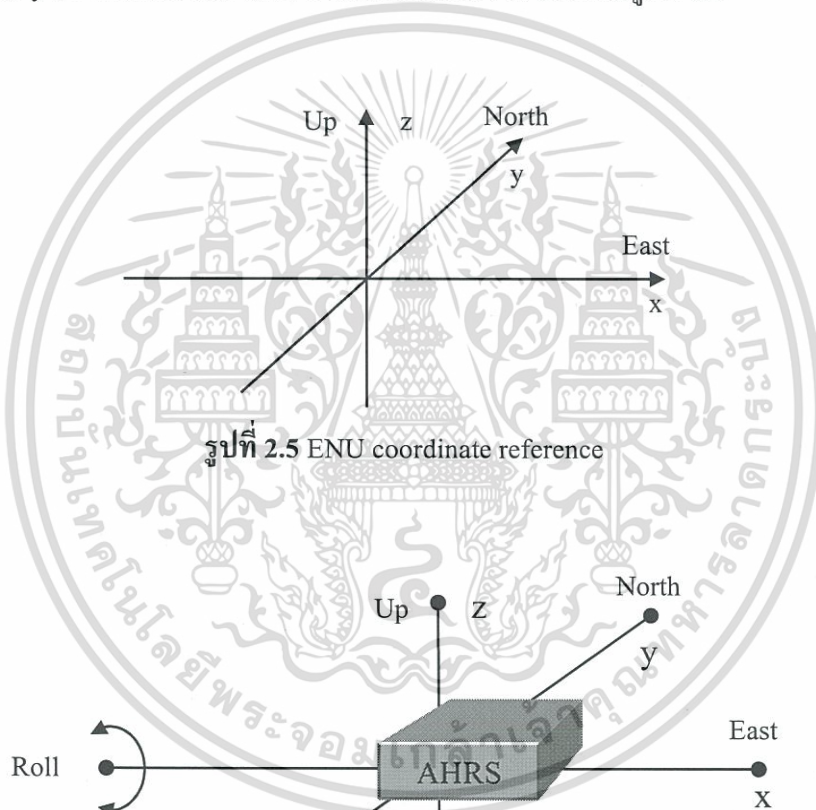
เครื่องวัดความเข้มข้นของสนามแม่เหล็ก (Magnetometer) ในที่นี้จะหมายถึง MEMS Magnetometer หรือ MEMS Magnetic field sensor ซึ่งนิยมใช้เป็นเข็มทิศในอุปกรณ์อิเล็กทรอนิกส์ต่างๆ โดยสามารถวัดขนาด (Magnitude) ของสนามแม่เหล็ก (Magnetic fields) รอบตัวอุปกรณ์ได้ จึงใช้วัดสนามแม่เหล็กของโลกเพื่อนำค่าสนามแม่เหล็กโลกมาใช้ในการอ้างอิงทิศทางในเข็มทิศอิเล็กทรอนิกส์ (Electronic compass) ซึ่งระบบอ้างอิงทิศทางและการวางตัว หรือ AHRS จะใช้ค่าจากเข็มทิศอิเล็กทรอนิกส์ดังกล่าวเป็นส่วนสำคัญในการอ้างอิงทิศทาง (Heading) ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

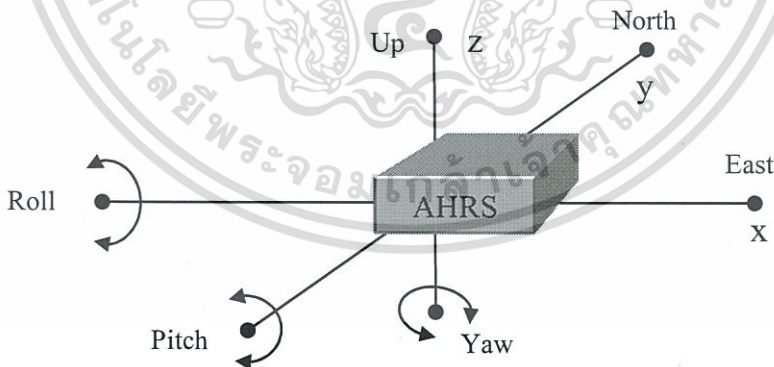
2.3 ระบบอ้างอิงทิศทางและการวางตัว (ATTITUDE AND HEADING REFERENCE SYSTEM)

ระบบอ้างอิงทิศทางและการวางตัว (Attitude and Heading Reference System) หรือ AHRS เป็นระบบที่ใช้สำหรับอ้างอิงทิศทางการวางตัว (Orientation) หรือ ตำแหน่งการหมุนรอบแนวแกน ในปริภูมิแบบยูคลิด (Euclidean space) ค่าที่ได้จะเป็นค่ามุมออยเลอร์ (Euler angle) โดยมีชื่อเรียกค่ามุมที่ได้ว่า Roll (ϕ) Pitch (θ) และ Yaw (ψ) โดยสามารถเรียกตามลำดับการเรียงได้ว่า มุม RPY

สำหรับยานยนต์ หรือพาหนะที่เคลื่อนที่บนพื้นดิน (Land vehicles) มุม RPY จะอ้างอิงกับระบบพิกัด East-North-Up (ENU) ซึ่งเป็นระบบพิกัดในการอ้างอิงบนพื้นผิวโลก โดยแกน x คือ ทิศตะวันออก แกน y คือทิศเหนือ และ แกน z คือทิศที่ชี้ออกจากโลก ตามรูปที่ 2.5



รูปที่ 2.5 ENU coordinate reference



รูปที่ 2.6 RPY Attitude reference

เมื่อนำมุม RPY มาอ้างอิงกับระบบพิกัด ENU จะได้การหมุนรอบแกนตามรูปที่ 2.6 คือ

Roll เป็นค่าในการหมุนรอบแกน x เป็นบวกในทิสวนขวา เป็นศูนย์เมื่อนานพื้น โลก

Pitch เป็นค่าในการหมุนรอบแกน y เป็นบวกในทิสวนขวา เป็นศูนย์เมื่อนานพื้น โลก

Yaw เป็นค่าในการหมุนรอบแกน z เป็นบวกในทิสวนขวา เป็นศูนย์เมื่อชี้ไปยังทิศเหนือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ระบบพิกัด (COORDINATE SYSTEM)

2.4.1 การแปลงระบบพิกัด (Coordinate system transform)

การแปลงระบบพิกัดเป็นกระบวนการสำหรับแปลงระบบพิกัดอ้างอิงหรือกรอบอ้างอิง จากระบบพิกัดอ้างอิงเดิม ไปยังระบบพิกัดอื่นด้วยเมตริกซ์การแปลง (Transformation matrix) ซึ่งมาจากการประยุกต์ใช้เมตริกซ์การหมุน (Rotation matrix) จากหลักการของ Direct Cosine Matrix [1]

การแปลงระบบพิกัดในระบบการนำร่องใช้เพื่อการปรับระบบกรอบอ้างอิงซึ่งโดยทั่วไปค่าจาก IMU จะเป็นค่าตามกรอบอ้างอิงเฉื่อยซึ่งเป็นกรอบอ้างอิงที่ไม่มีความเร่งและการหมุน แต่การคำนวณตำแหน่งในระบบ INS จะต้องใช้ค่าที่อยู่ในกรอบอ้างอิงเดียวกับพาหะ ซึ่งมีการเคลื่อนที่ทำให้มีความเร่งและสามารถเกิดความเร็วเชิงมุมได้ จึงจำเป็นต้องมีการแปลงระบบพิกัดเกิดขึ้น

การแปลงระบบพิกัดอ้างอิงของค่าจาก IMU ให้อยู่ในระบบพิกัดนำร่องด้วยค่า RPY จาก AHRS สามารถทำได้โดยใช้เมตริกซ์การแปลงในสมการที่ (2.4) ซึ่งอ้างอิงมาจาก [5] ดังนี้

$$M = \begin{bmatrix} P_c Y_c & P_c Y_s & -P_s \\ Y_c P_s R_s - R_c Y_s & R_c Y_c + P_s R_s Y_s & P_c R_s \\ R_c Y_c P_s + R_s Y_s & R_c P_s Y_s - Y_c R_s & P_c R_c \end{bmatrix} \quad (2.4)$$

โดยที่

- M คือ เมตริกซ์การแปลง
- R_c คือ Cosine (Roll) เป็นค่าโคไซน์ของมุม Roll จาก AHRS
- R_s คือ Sine (Roll) เป็นค่าไซน์ของมุม Roll จาก AHRS
- P_c คือ Cosine (Pitch) เป็นค่าโคไซน์ของมุม Pitch จาก AHRS
- P_s คือ Sine (Pitch) เป็นค่าไซน์ของมุม Pitch จาก AHRS
- Y_c คือ Cosine (Yaw) เป็นค่าโคไซน์ของมุม Yaw จาก AHRS
- Y_s คือ Sine (Yaw) เป็นค่าไซน์ของมุม Yaw จาก AHRS

2.4.2 การปรับค่าแรงโน้มถ่วง (Gravity correction)

การปรับค่าแรงโน้มถ่วงเป็นการปรับค่าความเร่งจาก IMU โดยการนำเอาค่าแรงโน้มถ่วงของโลกออกจากค่าความเร่ง ซึ่งค่าความเร่งที่ได้จาก IMU จะมีทั้งความเร่งจากการเคลื่อนที่และความเร่งจากแรงดึงดูดของโลกซึ่งมีแรงกระทำกับมาตรวัดความเร่งใน IMU ตลอดเวลา

เพื่อให้ค่าความเร่งที่ใช้ในการคำนวณเป็นค่าที่เกิดจากการเคลื่อนที่เท่านั้นจึงต้องมีการปรับเอาแรงโน้มถ่วงออกก่อนโดยกำหนดให้ค่าแรงโน้มถ่วง $1g$ มีค่าเท่ากับ 9.81 m/s^2 เมื่อนำเอาเมตริกซ์การแปลงในสมการที่ (2.4) มาคูณกับเมตริกซ์แรงโน้มถ่วงของโลก $\begin{bmatrix} 0 \\ 0 \\ 9.81 \end{bmatrix}$ ดังสมการที่ (2.5) จะได้ผลลัพธ์เป็นเมตริกซ์ A_g ซึ่งเป็นเมตริกซ์ของค่าความเร่งจากแรงโน้มถ่วงของโลกที่แฝงอยู่ในค่าความเร่งแต่ละแกนของ IMU

$$A_g = \begin{bmatrix} P_c Y_c & P_c Y_s & -P_s \\ Y_c P_s R_s - R_c Y_s & R_c Y_c + P_s R_s Y_s & P_c R_s \\ R_c Y_c P_s + R_s Y_s & R_c P_s Y_s - Y_c R_s & P_c R_c \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 9.81 \end{bmatrix} \quad (2.5)$$

$$A_g = \begin{bmatrix} -9.81 P_s \\ 9.81 P_c R_s \\ 9.81 P_c R_c \end{bmatrix} \quad (2.6)$$

โดยที่

- A_g คือ เมตริกซ์ของความเร่งจากแรงโน้มถ่วงของโลก
- R_c คือ Cosine (Roll) เป็นค่าโคไซน์ของมุม Roll จาก AHRS
- R_s คือ Sine (Roll) เป็นค่าไซน์ของมุม Roll จาก AHRS
- P_c คือ Cosine (Pitch) เป็นค่าโคไซน์ของมุม Pitch จาก AHRS
- P_s คือ Sine (Pitch) เป็นค่าไซน์ของมุม Pitch จาก AHRS
- Y_c คือ Cosine (Yaw) เป็นค่าโคไซน์ของมุม Yaw จาก AHRS
- Y_s คือ Sine (Yaw) เป็นค่าไซน์ของมุม Yaw จาก AHRS

สมการที่ (2.6) คือ เมตริกซ์ของค่าความเร่งที่เกิดจากแรงโน้มถ่วงของโลกซึ่งกระจายอยู่ในแกนทั้งสามของค่าความเร่งที่มาจาก IMU โดยค่าเหล่านี้จะมีมากขึ้นอยู่กับการทำมุมกับพื้นโลกของแกนนั้นๆ ซึ่งถ้าแกนใดทำมุมขนานกับพื้นโลกพอดี ก็จะไม่มีความเร่งที่เกิดจากแรงโน้มถ่วงของโลกเกิดขึ้น โดยทั่วไปแกน z ของ IMU จะเป็นแกนที่ทำมุมตั้งฉากกับพื้นโลกมากที่สุดจึงทำให้เกิดค่าความเร่งจากแรงโน้มถ่วงของโลกมากที่สุดเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากการหาเมตริกซ์ของความเร่งที่เกิดจากแรงโน้มถ่วงของโลกได้แล้วให้นำเมตริกซ์

A_g ในสมการที่ (2.6) มาลบกับเมตริกซ์ของค่าความเร่งจาก IMU คือ $\begin{bmatrix} IMU_{ax} \\ IMU_{ay} \\ IMU_{az} \end{bmatrix}$ ดังสมการที่ (2.7)

ซึ่งจะได้ผลลัพธ์เป็นเมตริกซ์ G ในสมการที่ (2.8)

$$G = \begin{bmatrix} IMU_{ax} \\ IMU_{ay} \\ IMU_{az} \end{bmatrix} - \begin{bmatrix} -9.81P_s \\ 9.81P_cR_s \\ 9.81P_cR_c \end{bmatrix} \quad (2.7)$$

$$G = \begin{bmatrix} IMU_{ax} + 9.81P_s \\ IMU_{ay} - 9.81P_cR_s \\ IMU_{az} - 9.81P_cR_c \end{bmatrix} \quad (2.8)$$

โดยที่

- G คือ เมตริกซ์ของค่าความเร่งที่ถูกปรับเอาแรงโน้มถ่วงของโลกออกแล้ว
- P_c คือ Cosine (Pitch) เป็นค่าโคไซน์ของมุม Pitch จาก AHRS
- P_s คือ Sine (Pitch) เป็นค่าไซน์ของมุม Pitch จาก AHRS
- R_c คือ Cosine (Roll) เป็นค่าไซน์ของมุม Roll จาก AHRS
- R_s คือ Sine (Roll) เป็นค่าโคไซน์ของมุม Roll จาก AHRS
- IMU_{ax} คือ ค่าความเร่งแกน x จาก IMU
- IMU_{ay} คือ ค่าความเร่งแกน y จาก IMU
- IMU_{az} คือ ค่าความเร่งแกน z จาก IMU

ค่าความเร่งในสมการที่ (2.8) จะต้องถูกปรับกรอบอ้างอิงให้เหมาะสมก่อนนำไปใช้เพื่อความถูกต้องของผลลัพธ์ในการคำนวณหาความเร็วและตำแหน่ง ด้วยสมการในระบบ INS กระบวนการในการแปลงระบบพิกัด หรือการปรับกรอบอ้างอิงจะกล่าวถึงในส่วนถัดไป

2.4.3 การแปลงระบบพิกัดของค่าความเร่ง (Acceleration coordinate transform)

การแปลงระบบพิกัดของค่าความเร่งจากระบบพิกัดอ้างอิงเฉื่อยให้อยู่ในระบบพิกัดนำร่องสามารถทำได้โดยการใช้เมตริกซ์การแปลงในสมการที่ (2.4) คูณกับเมตริกซ์ของค่าความเร่งที่ปรับแรงโน้มถ่วงแล้วจากสมการที่ (2.8) จะได้สมการที่ (2.9) โดยผลลัพธ์จะเป็นเมตริกซ์ของค่าความเร่งที่ปรับระบบอ้างอิงแล้วดังสมการที่ (2.10)

$$\begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} = \begin{bmatrix} P_c Y_c & P_c Y_s & -P_s \\ Y_c P_s R_s - R_c Y_s & R_c Y_c + P_s R_s Y_s & P_c R_s \\ R_c Y_c P_s + R_s Y_s & R_c P_s Y_s - Y_c R_s & P_c R_c \end{bmatrix} \begin{bmatrix} G_{11} \\ G_{21} \\ G_{31} \end{bmatrix} \quad (2.9)$$

$$\begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} = \begin{bmatrix} G_{11} P_c Y_c - G_{11} P_s + G_{21} P_c Y_s \\ G_{11} P_c R_s + G_{11} (Y_c P_s R_s - R_c Y_s) + G_{21} (R_c Y_c + P_s R_s Y_s) \\ G_{11} P_c R_c + G_{21} (R_c P_s Y_s - Y_c R_s) + G_{11} (R_c Y_c P_s + R_s Y_s) \end{bmatrix} \quad (2.10)$$

โดยที่

R_c	คือ Cosine (Roll) เป็นค่าโคไซน์ของมุม Roll จาก AHRS
R_s	คือ Sine (Roll) เป็นค่าไซน์ของมุม Roll จาก AHRS
P_c	คือ Cosine (Pitch) เป็นค่าโคไซน์ของมุม Pitch จาก AHRS
P_s	คือ Sine (Pitch) เป็นค่าไซน์ของมุม Pitch จาก AHRS
Y_c	คือ Cosine (Yaw) เป็นค่าโคไซน์ของมุม Yaw จาก AHRS
Y_s	คือ Sine (Yaw) เป็นค่าไซน์ของมุม Yaw จาก AHRS
G_{11}	คือ ค่าแถวที่ 1 หลักที่ 1 ของเมตริกซ์ G ในสมการที่ (2.8)
G_{21}	คือ ค่าแถวที่ 2 หลักที่ 1 ของเมตริกซ์ G ในสมการที่ (2.8)
G_{31}	คือ ค่าแถวที่ 3 หลักที่ 1 ของเมตริกซ์ G ในสมการที่ (2.8)
A_x	คือ ค่าความเร่งแกน x ที่ปรับระบบอ้างอิงแล้ว มีหน่วยเป็น m/s^2
A_y	คือ ค่าความเร่งแกน y ที่ปรับระบบอ้างอิงแล้ว มีหน่วยเป็น m/s^2
A_z	คือ ค่าความเร่งแกน z ที่ปรับระบบอ้างอิงแล้ว มีหน่วยเป็น m/s^2

ค่าความเร่งแต่ละตัวในเมตริกซ์ของสมการที่ (2.10) จะถูกนำไปใช้เป็นค่าความเร่งในสมการและการคำนวณทั้งหมดในระบบการติดตามตำแหน่งการเคลื่อนที่ที่ได้พัฒนาขึ้นในโครงการนี้

2.4.4 การปรับทิศทาง การเคลื่อนที่ (Heading adjustment)

แนวคิดในการปรับทิศทาง การเคลื่อนที่บนส่วนแสดงผลคือการนำค่า Yaw จาก AHRS มาอ้างอิงทิศทาง ซึ่งค่าดังกล่าวจะอยู่ในช่วง $-\pi$ ถึง π และจะเป็น 0 เมื่อชี้ไปทางทิศเหนือ สำหรับการติดตั้ง IMU บนรถจำลองจะทำการติดตั้งโดยการอ้างอิงหัวรถจำลองกับทิศเหนือซึ่งเมื่อหัวรถจำลองหันไปทางทิศเหนือค่าทิศทางที่อ่านได้จาก AHRS จะมีค่าเป็น 0 ส่วนในโปรแกรมแสดงผลได้กำหนดให้ด้านบนคือทิศเหนือ และ ด้านล่างคือทิศใต้ เพื่อใช้สำหรับการอ้างอิงทิศทางในการเคลื่อนที่ของรถจำลองระหว่างในโปรแกรมส่วนแสดงผลกับการวิ่งบนพื้นที่จริง

สมการสำหรับปรับทิศทาง การเคลื่อนที่ที่มีที่มาจากหลักการทางตรีโกณมิติพื้นฐาน เมื่อมีการเคลื่อนที่โดยหันหัวรถไปยังทิศเหนือค่า Yaw จะใกล้เคียง 0 ในทางกลับกันหากหัวรถหันไปยังทิศใต้ค่า Yaw จะใกล้เคียง 3.14 ซึ่งการเคลื่อนที่ในทั้งสองทิศทางจะอยู่บนแกนเดียวในโปรแกรมส่วนแสดงผล สมการที่ได้จึงใช้ทั้งค่าโคไซน์ (Cosine) และค่าไซน์ (Sine) ของค่า Yaw ที่ได้จาก AHRS เพื่อให้สามารถอ้างอิงทิศทางได้อย่างถูกต้องทั้งในสี่ตุภาค (Quadrant)

การปรับทิศทางในการเคลื่อนที่จะเกิดขึ้นหลังจากการคำนวณตำแหน่งการเคลื่อนที่ด้วยสมการที่ (2.2) เสร็จสิ้นแล้วทั้งในแกน x และในแกน y โดยสมการที่ใช้ปรับทิศทาง มีดังนี้

$$p_x = s_x \cos \psi - s_y \sin \psi \quad (2.11)$$

$$p_y = s_x \sin \psi - s_y \cos \psi \quad (2.12)$$

โดยที่

p_x คือ ตำแหน่งบนแกน x ในโปรแกรมส่วนแสดงผล

p_y คือ ตำแหน่งบนแกน y ในโปรแกรมส่วนแสดงผล

s_x คือ ตำแหน่งในแกน x ที่คำนวณได้จากสมการที่ (2.2)

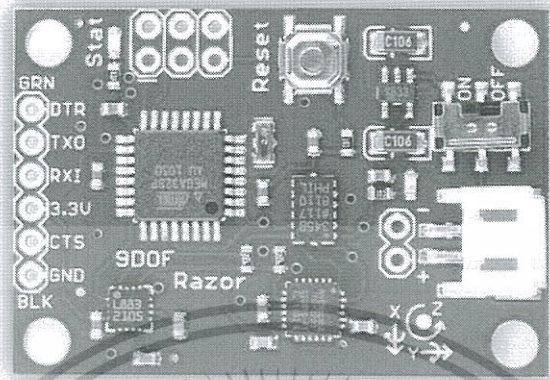
s_y คือ ตำแหน่งในแกน y ที่คำนวณได้จากสมการที่ (2.2)

ψ คือ ค่า Yaw จาก AHRS

สำหรับการแสดงผลทิศทาง การเคลื่อนที่ในโปรแกรมส่วนแสดงผลจะใช้ค่าที่คำนวณได้จากสมการที่ (2.11) และ (2.12) เป็นค่าตำแหน่งสำหรับแสดงผลในโปรแกรม ซึ่งค่าตำแหน่งดังกล่าวอาจไม่ตรงกับการเคลื่อนที่จริงหากค่าทิศทางที่ได้จาก AHRS มีความผิดพลาด

2.5 ฮาร์ดแวร์และซอฟต์แวร์ (HARDWARE AND SOFTWARE)

2.5.1 หน่วยวัดแรงเฉื่อย Razor IMU



รูปที่ 2.7 9 DOF Razor IMU

หน่วยวัดแรงเฉื่อย หรือ IMU รุ่น "9DOF Razor IMU" [6] ในรูปที่ 2.7 เป็น MEMS IMU ที่ประกอบไปด้วยเซนเซอร์ 3 ตัว และ MCU 1 ตัว โดยมีรายละเอียดดังนี้

ITG-3200 (3-axis MEMS Gyroscope) เป็นไจโรสโคปแบบ 3 แกน ใช้สำหรับอ่านค่าอัตราเร็วในการหมุนของ IMU

ADXL345 (3-axis MEMS accelerometer) เป็นมาตรความเร่งแบบ 3 แกน ใช้สำหรับอ่านค่าความเร่งของ IMU

HMC5883L (3-axis MEMS magnetometer) เป็นเครื่องวัดความเข้มข้นของสนามแม่เหล็กแบบ 3 แกน ใช้สำหรับอ่านค่าสนามแม่เหล็กโดยรอบ IMU

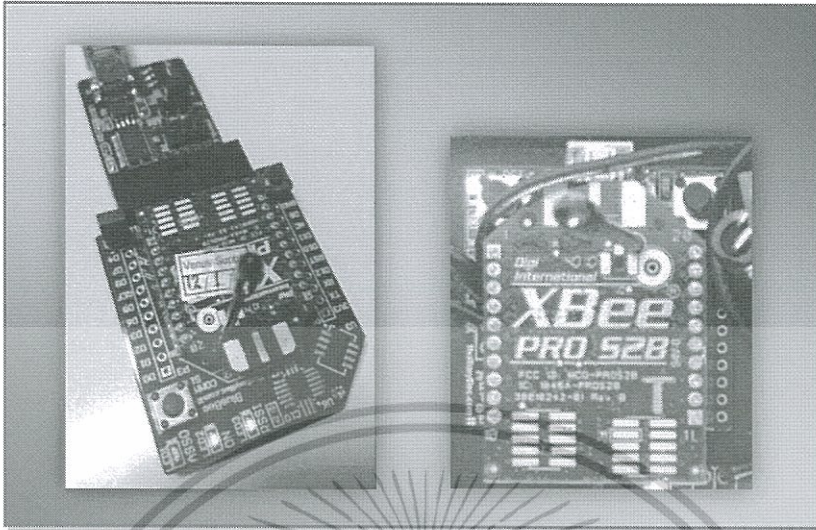
ATmega328 (MCU) เป็นไมโครคอนโทรลเลอร์ความถี่ 8MHz เป็นหน่วยประมวลผลของ IMU ทำหน้าที่อ่านข้อมูลจากเซนเซอร์ แล้วทำการประมวลผล ก่อนส่งออกทาง TTL Serial port

9DOF Razor IMU ตัวนี้ใช้ Razor AHRS Firmware [7] เป็นระบบควบคุมการทำงาน ซึ่งมีฟังก์ชันการปรับค่ามาตรฐาน (Calibrate) และสามารถแก้ไขการทำงานส่วนอื่น ๆ ได้ด้วยโปรแกรม Arduino IDE จึงเหมาะกับการประยุกต์ใช้ที่หลากหลาย โครงการนี้ได้ทำการปรับแต่งรูปแบบการส่งออกข้อมูลรูปแบบ Text ที่ส่งออกทาง TTL Serial port ด้วยค่า Baud rate ที่ 57,600 โดยมีข้อมูลที่ส่งออกทั้งหมด 7 ค่า ส่งเรียงตามลำดับคือ เลขชุดข้อมูล (0-49) ค่าความเร่งแกน x แกน y และ แกน z ค่ามุม Roll ค่ามุม Pitch และ ค่ามุม Yaw จากระบบ AHRS

สำหรับรายละเอียดอื่น ๆ เช่น ข้อมูลทางเทคนิค และ วิธีการแก้ไข Firmware ด้วย Arduino IDE จะอยู่ในภาคผนวก ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 ชุดอุปกรณ์รับส่งข้อมูลไร้สาย



รูปที่ 2.8 XBee Pro S2B

ชุดอุปกรณ์รับส่งข้อมูลไร้สาย XBee PRO รุ่น S2B ในรูปที่ 2.8 เป็นชุดรับส่งข้อมูลที่ ออกแบบมาสำหรับผู้ใช้ที่ต้องการรับส่งข้อมูลไร้สายในระยะทางไกลโดยทำงานบนคลื่นความถี่ 2.4GHz ด้วยรูปแบบการสื่อสารจำเพาะของ ZigBee สามารถสร้างเครือข่ายในรูปแบบ point-to-point และ point-to-multipoint ได้ สามารถปรับแต่งค่าต่างๆในการทำงานของตัวอุปกรณ์ได้ด้วย โปรแกรม XCTU ซึ่งใช้สำหรับการปรับแต่งค่าการทำงานของอุปกรณ์ของ ZigBee โดยเฉพาะ

เพื่อความราบรื่นในการรับข้อมูลในสถานที่ซึ่งอาจถูกรบกวนจากสัญญาณ WLAN ซึ่งใช้งานอยู่บนคลื่นความถี่เดียวกัน จึงต้องเลือกใช้ชุดรับส่งที่มีการรับส่งข้อมูลด้วยรูปแบบการสื่อสาร และการเข้ารหัสสัญญาณที่จำเพาะ และสามารถรับส่งได้ไกลกว่า 100 เมตร ทั้งในร่มและกลางแจ้ง

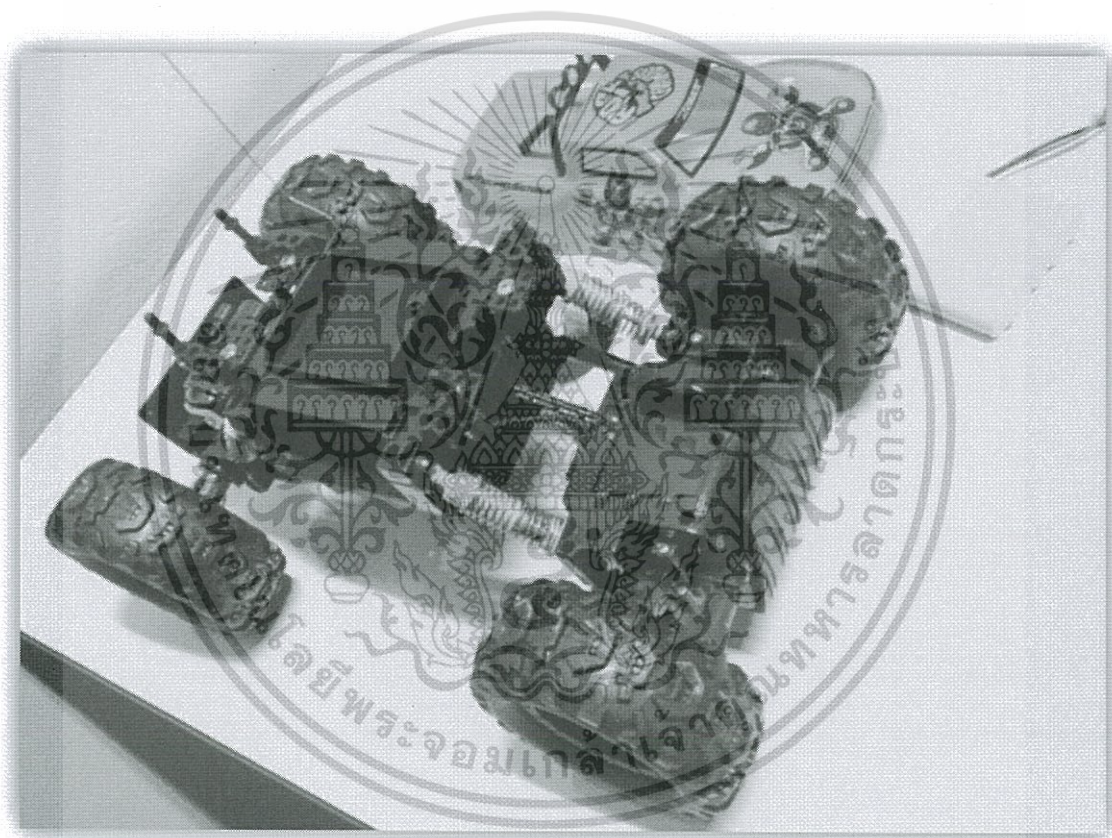
ในโครงการนี้ได้ออกแบบให้มีการส่งข้อมูลผ่านเครือข่ายไร้สายโดยมีระยะทำการที่ 100 เมตร สภาพแวดล้อมการทำงานอยู่ภายในอาคารรวมถึงพื้นที่โล่ง ด้วยเหตุนี้จึงเลือกใช้ XBee PRO รุ่น S2B มาเป็นอุปกรณ์ในการรับส่งข้อมูลระหว่างคอมพิวเตอร์และชุดอุปกรณ์ติดตามตำแหน่งการเคลื่อนที่ที่พัฒนาขึ้นเพื่อติดตามรถจำลอง

สำหรับรายละเอียดอื่น ๆ เช่น ข้อมูลทางเทคนิค และ วิธีการปรับค่าด้วยโปรแกรม XCTU จะอยู่ในภาคผนวก ก.

2.5.3 รถจำลอง (Car model)

รถจำลองสำหรับการทดสอบในโครงการนี้ได้เลือกใช้รถบังคับวิทยุในรูปแบบรถไต่หิน ซึ่งมีกำลังขับและอัตราเร่งสูงซึ่งเพียงพอสำหรับใช้ในการทดสอบ และมีขนาดใหญ่พอที่จะติดตั้งชุดอุปกรณ์ของโครงการนี้ได้โดยไม่ส่งผลกระทบต่อการใช้งานปกติ

รถจำลองที่เลือกใช้ี้สามารถเคลื่อนที่ได้ 4 ทิศทาง คือ เดินหน้า ถอยหลัง เลี้ยวซ้าย และ เลี้ยวขวา แต่ไม่สามารถควบคุมความเร็วได้ โดยมีมอเตอร์ขับจำนวน 3 ชุด คือ ชุดขับเคลื่อนล้อหน้า ชุดขับเคลื่อนล้อหลัง และชุดบังคับเลี้ยวล้อหน้า ในการทำงานต้องการไฟเลี้ยง ที่แรงดันขั้นต่ำ 6v. ใช้ชุดควบคุมไร้สายที่ทำงานบนคลื่นความถี่ 27 MHz



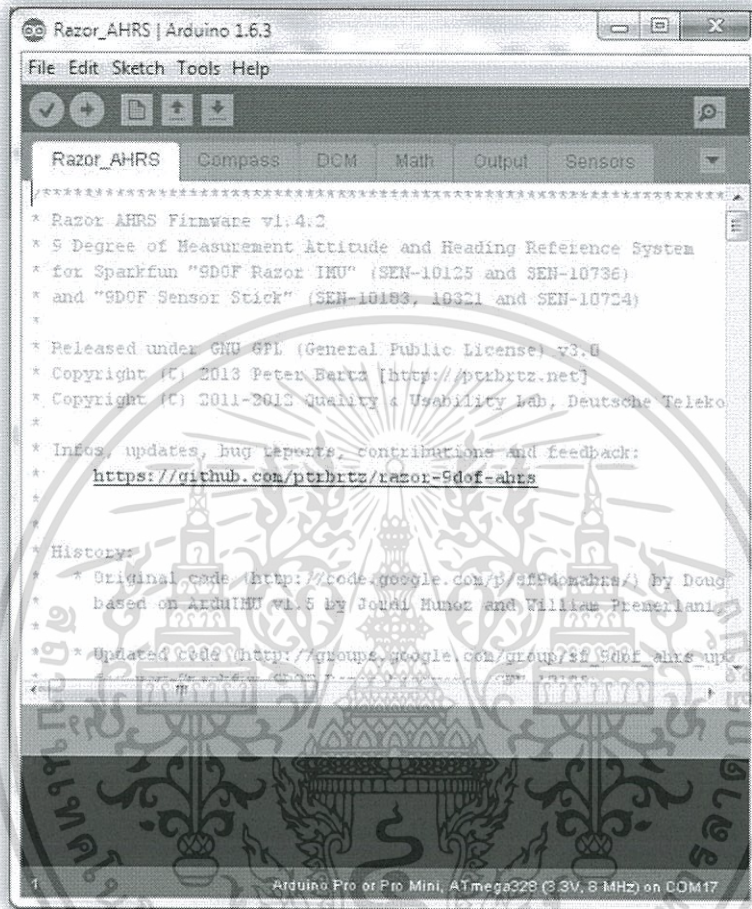
รูปที่ 2.9 Car model and equipment

ในรูปที่ 2.9 คือ ภาพของรถจำลองที่เลือกใช้ในการทดสอบโครงการ ในภาพประกอบด้วยตัวรถ (Body) ทีเจียว และ Ni-Cad Battery ความจุ 750 mAh แรงดัน 6 - 7.4v ทีเหลือึง สามารถนำมาประจุไฟใหม่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.4 โปรแกรมประยุกต์ที่ใช้ในการพัฒนาระบบ

2.5.4.1 Arduino IDE



รูปที่ 2.10 Arduino IDE

Arduino IDE เป็นสภาพแวดล้อมในการพัฒนาชุดคำสั่งบนอุปกรณ์ไมโครคอนโทรลเลอร์ในโครงการ Arduino โดยในโครงการนี้จะใช้ Arduino IDE ในการปรับแต่งและแก้ไข Firmware ของ IMU โดยในรูปที่ 2.10 เป็นการเปิดไฟล์ข้อมูลของ Razor AHRS firmware ที่ทำการแก้ไข

โปรแกรมนี้ได้รวบรวมเครื่องมือที่จำเป็นสำหรับการพัฒนาชุดคำสั่งที่ทำงานในระบบไมโครคอนโทรลเลอร์เอาไว้ เช่น Hardware profile, Module library และ Serial monitor ซึ่งอำนวยความสะดวกในการพัฒนาระบบเป็นอย่างยิ่ง โดยโปรแกรมนี้ใช้ภาษา C++ ในการพัฒนาชุดคำสั่ง และมีคอมไพเลอร์ในตัว จึงสามารถทำการอัปโหลดชุดคำสั่งเข้าสู่ไมโครคอนโทรลเลอร์ได้ทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.4.2 Processing2

```

ClientIMUMove | Processing 2.2.1
File Edit Sketch Tools Help
Java
ClientIMUMove
strokeWeight(1);
text("Frame rate: "+frameRate,600,30);
text("Data rate: "+data_rate,600,50);
text("PC Time: "+hour()+" "+minute()+" "+second(),600,70);

text("RPY: "+nf(data[0],1,3)+" ", "+nf(data[1],1,3)+" ", "+nf(data
text("Px: "+nf(px,1,3),320,height-10);
text("Py: "+nf(py,1,3),480,height-10);

text("Sx: "+nf(sx,1,3),10,height-40);
text("Sy: "+nf(sy,1,3),160,height-40);
text("Vx: "+nf(vx,1,3),320,height-40);
text("Vy: "+nf(vy,1,3),480,height-40);

text("Max x: "+nf(max(x),1,3),10,height-100);
text("Min x: "+nf(min(x),1,3),160,height-100);
text("Max y: "+nf(max(y),1,3),320,height-100);
text("Min y: "+nf(min(y),1,3),480,height-100);
popMatrix();
}

```

รูปที่ 2.11 Processing2

Processing2 เป็นโปรแกรมสำหรับการพัฒนาระบบการแสดงผล สามารถแสดงผลได้ทั้งในแบบ 2 มิติ และ 3 มิติ โดยสามารถใช้เป็นส่วนติดต่อและรับข้อมูลจากผู้ใช้ได้

การพัฒนาระบบด้วย Processing2 ทำได้โดยการเขียนชุดคำสั่งตามรูปแบบที่โปรแกรมกำหนดด้วยภาษา Java ซึ่งเป็นชุดคำสั่งในลักษณะของการวาดรูปโดยใช้เส้น รูปทรง ขนาด สี โดยใช้หลักการทางคณิตศาสตร์ในการสร้างรูปร่าง เมื่อหลาย ๆ ชุดคำสั่งทำงานร่วมกันก็จะได้เป็นการแสดงผลตามที่ได้ออกแบบไว้ โปรแกรมนี้สามารถส่งออกการแสดงผลที่พัฒนาขึ้นในรูปแบบของไฟล์ต่าง ๆ หรือโปรแกรมประยุกต์ได้ โดยมีตัวอย่างชุดคำสั่งให้ทำการศึกษาได้ด้วยตนเอง

โครงการนี้ใช้ Processing2 ในการพัฒนาโปรแกรมส่วนที่ทำงานอยู่บนคอมพิวเตอร์ทั้งหมด โดยในรูปที่ 2.11 เป็นชุดคำสั่งบางส่วนของระบบที่ได้พัฒนาขึ้นสำหรับโครงการนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

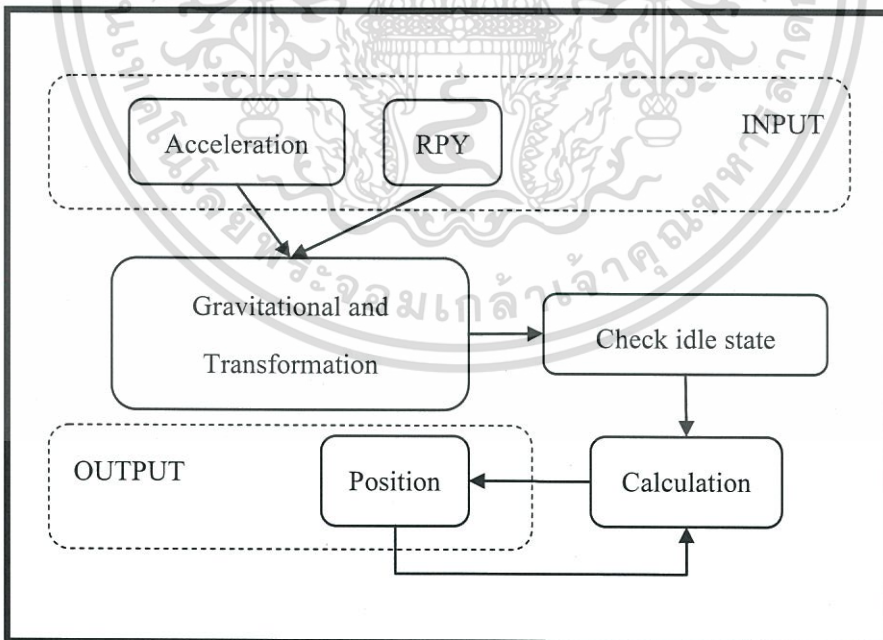
บทที่ 3

การพัฒนาระบบติดตามตำแหน่ง

การพัฒนาระบบติดตามตำแหน่งการเคลื่อนที่ในโครงการนี้ได้พัฒนาระบบโดยใช้หลักการหาค่าตำแหน่งด้วยค่าความเร่งตามแนวคิดของ INS ซึ่งมีการปรับลดค่าความผิดพลาดโดยการปรับค่าแรงโน้มถ่วงออกจากความเร่งด้วยการใช้ข้อมูลทิศทางการวางตัว และการหาภาวะหยุดนิ่งด้วยค่าเฉลี่ยจากขนาดของความเร่งในช่วงเวลาหนึ่งเพื่อลดค่าความผิดพลาดสะสม โดยข้อมูลความเร่งและข้อมูลทิศทางการวางตัวได้มาจาก Razor IMU ที่ทำงานด้วย AHRS Firmware ซึ่งมีการปรับแต่งในส่วนของการส่งข้อมูลเพื่อให้สะดวกต่อการใช้งานร่วมกับชุดรับส่งข้อมูลไร้สาย XBee Pro ที่ติดตั้งอยู่บนรถจำลองที่ใช้สำหรับการทดสอบ

3.1 โครงสร้างของระบบติดตามตำแหน่ง (SYSTEM CONFIGURATION)

ระบบ INS ที่ออกแบบสำหรับโครงการนี้เป็นระบบที่ทำงานไม่ซับซ้อนโดยการอาศัยค่าความเร่ง และ ทิศทางการวางตัว ซึ่งได้มาจาก Razor IMU ในการประมวลผลทั้งระบบ



รูปที่ 3.1 INS diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.1 ค่าที่ใช้สำหรับการทำงานของระบบ คือ ค่าความเร่ง (Acceleration) และ ทิศทางการวางตัว (RPY) ที่ได้มาจาก IMU โดยค่าทั้งสองจะถูกนำมาเข้ากระบวนการปรับค่าแรงโน้มถ่วง และการปรับค่าระบบอ้างอิงเพื่อใช้ในการคำนวณ

จากนั้นจะทำการตรวจสอบว่าอยู่ในสถานะหยุดนิ่งหรือไม่ก่อนทำการคำนวณด้วยสมการที่ (2.2) เพื่อหาค่าตำแหน่ง รายละเอียดในการออกแบบแต่ละส่วนของระบบ INS ในรูปที่ 3.1 มีดังนี้

ส่วนข้อมูลเข้า (Input) ประกอบด้วยข้อมูลค่าความเร่ง (Acceleration) และ ข้อมูลทิศทาง การวางตัว (RPY) โดยเป็นค่าที่รับมาจาก Razor IMU ในรูปแบบของชุดข้อความซึ่งระบบจะทำการ ตรวจสอบความถูกต้องโดยการตรวจสอบจำนวนชุดข้อมูลที่ได้รับ เสร็จแล้วทำการแปลงค่าข้อมูล ในแต่ละชุดให้เป็นตัวเลขแล้วเก็บไว้ในตัวแปรชุดข้อมูลเพื่อให้ส่วนอื่นเรียกใช้งานได้ที่

ส่วนการปรับค่าความเร่ง (Gravitational and Transformation) ประกอบด้วย 2 ส่วนหลัก คือ ส่วนการปรับค่าแรงโน้มถ่วง (Gravitational) และการปรับระบบพิกัดอ้างอิง (Transformation) โดยใช้ค่าจากสมการที่ (2.8) สำหรับการปรับค่าแรงโน้มถ่วง และ ใช้ค่าจากสมการที่ (2.10) สำหรับการปรับระบบพิกัดอ้างอิง โดยเมื่อทำการปรับค่าความเร่งเสร็จแล้วจะเก็บค่าไว้ในตัวแปรเดิม เพื่อให้ส่วนอื่นในโปรแกรมสามารถนำค่าไปใช้ต่อได้ที่

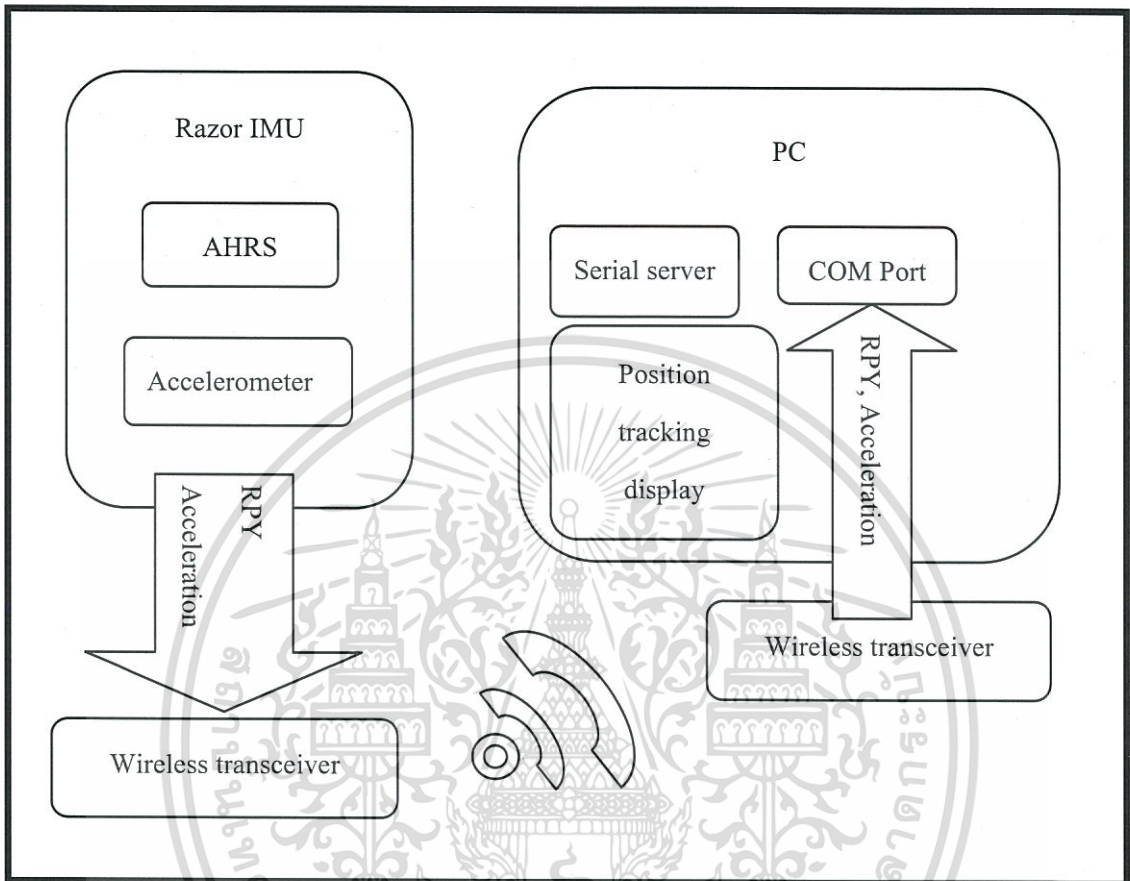
ส่วนการตรวจสอบสภาพหยุดนิ่ง (Check idle status) เป็นการประเมินสถานะของพาหะว่า มีการเคลื่อนที่อยู่หรือไม่จากขนาดของความเร่งเฉลี่ยในช่วงเวลาหนึ่งซึ่งหาได้จากค่าเฉลี่ยของค่า จากสมการที่ (2.3) ถ้าหากเป็นภาวะหยุดนิ่งจะทำการปรับค่าความเร็วจากการคำนวณในช่วงเวลา ก่อนหน้าเป็นศูนย์ เพื่อให้การคำนวณตำแหน่งที่จะเกิดขึ้นในอนาคตมีค่าใกล้เคียงศูนย์

ส่วนการคำนวณ (Calculation) เป็นส่วนที่ทำการคำนวณตำแหน่ง และ ความเร็ว จากค่า ความเร่งที่ผ่านกระบวนการปรับค่ามาแล้ว โดยใช้สมการที่ (2.2) สำหรับการคำนวณตำแหน่ง และ สมการที่ (2.1) สำหรับการคำนวณความเร็ว

ส่วนการแสดงผล (Output) เป็นส่วนที่นำค่าต่างๆในระบบประมวลผลมาทำการแสดงผล โดยเฉพาะการแสดงผลค่าตำแหน่ง (Position) ที่คำนวณได้เป็นหลัก โดยจะแสดงในรูปแบบของ การติดตามการเคลื่อนไหวในแผนที่จำลองซึ่งอยู่ภายในหน้าต่างโปรแกรมแสดงผลที่พัฒนาขึ้น และ ยังประกอบด้วยค่าตัวเลขจากระบบเพื่อใช้ในการตรวจสอบอื่น ๆ นอกจากการเคลื่อนที่ด้วย

3.2 องค์ประกอบด้านฮาร์ดแวร์ (HARDWARE COMPONENT)

การเชื่อมต่ออุปกรณ์ต่างๆในระบบเข้าด้วยกันเพื่อให้สามารถทำงานได้ตามเป้าหมาย



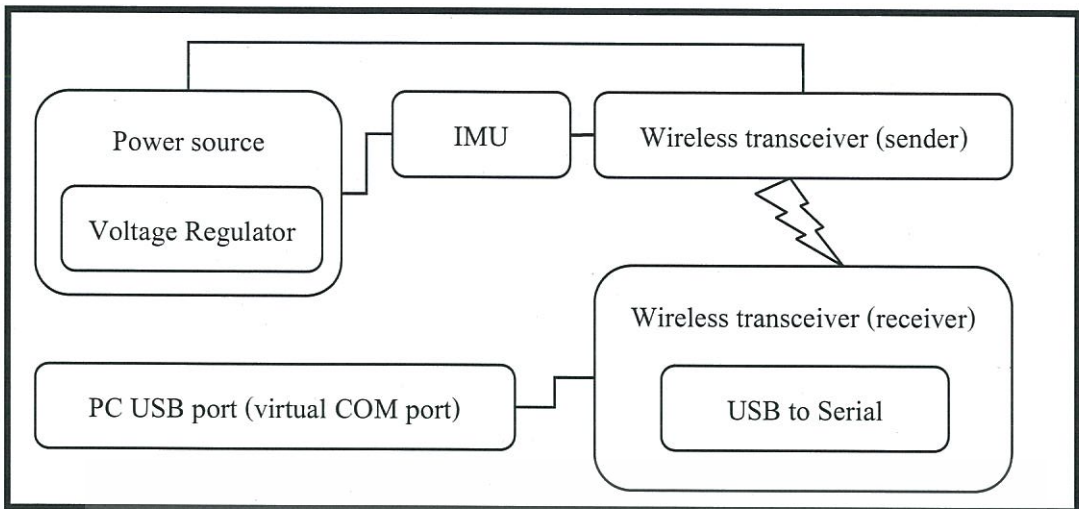
รูปที่ 3.2 Data flow diagram ระหว่างอุปกรณ์

จากรูปที่ 3.2 ข้อมูลจาก IMU จะถูกส่งไปยัง PC ผ่านทางเครือข่ายไร้สายโดยข้อมูลที่ส่งจะประกอบด้วย ลำดับข้อมูล ค่าความเร่ง และ RPY โดยชุดข้อมูลดังกล่าวจะใช้ในการคำนวณตำแหน่ง และทิศทางเคลื่อนที่ของรถจำลอง ด้วยโปรแกรมประยุกต์ที่พัฒนาขึ้น

Roll, Pitch, Yaw (RPY) เป็นค่าสภาวะการวางตัวของ IMU ซึ่งเป็นค่าที่อ้างอิงบนระบบพิกัดเดียวกับของรถจำลอง จึงใช้ในการอ้างอิงสภาวะของรถจำลองตามรูปที่ 2.6 ได้ ซึ่งแต่ละค่าจะให้ความหมาย ดังนี้

- Roll คือ ค่ามุมเอียง ซ้าย-ขวา ของรถ จะมีค่าเป็น 0 เมื่อขนาดกัปพื้น โลก
- Pitch คือ ค่ามุม ก้ม-เงย ของรถ จะมีค่าเป็น 0 เมื่อขนาดกัปพื้น โลก
- Yaw หรือ heading คือ ค่าองศาที่ตัวรถจำลองหัน ไปเมื่อเทียบกับทิศเหนือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

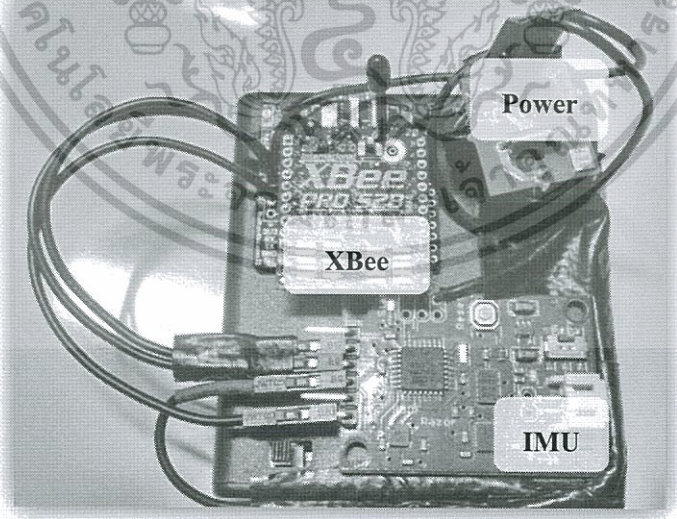


รูปที่ 3.3 ผังการเชื่อมต่ออุปกรณ์

จากรูปที่ 3.3 จะมีอุปกรณ์ที่เกี่ยวข้องในระบบ ดังนี้

- 9DOF Razor IMU 1 ชุด
- XBee Pro Wireless transceiver module (ทั้งฝั่งรับและส่ง) รวมเป็น 2 ชุด
- Power source สำหรับจ่ายไฟฟ้าให้กับอุปกรณ์ 1 ชุด

โดยเมื่อนำมาประกอบรวมกันจะได้เป็นชุดทดสอบสำหรับการอ่านค่าจาก IMU ผ่านระบบไร้สายในรูปที่ 3.4 ซึ่งต้องทำการทดสอบการอ่านค่าผ่านระบบไร้สายก่อนนำไปติดตั้งบนรถจำลอง



รูปที่ 3.4 ชุดอุปกรณ์ในการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การพัฒนาซอฟต์แวร์ในระบบ (SOFTWARE IMPLEMENTATION)

3.3.1 การปรับแต่ง Razor AHRS Firmware

ในโครงการนี้ใช้ 9DOF Razor IMU ซึ่งทำงานด้วย AHRS firmware ในการพัฒนาระบบ จึงทำการปรับปรุงแก้ไข AHRS firmware ด้วยโปรแกรม Arduino IDE เพื่อให้ทำงานได้ตามความต้องการ โดยรายการที่จะทำการแก้ไขตามการออกแบบระบบมีดังนี้

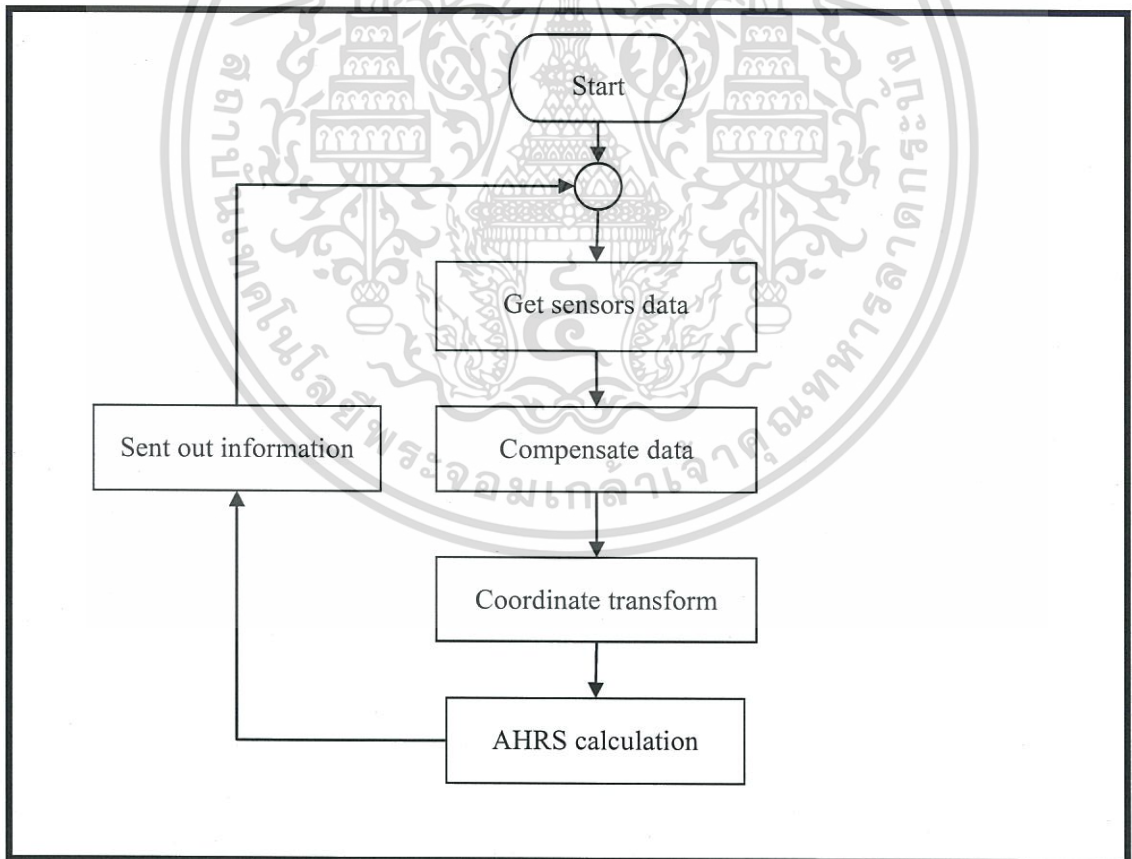
3.3.1.1 แก้ไขรุ่นของแผงวงจร IMU ให้ตรงกับรุ่นที่ใช้ซึ่งเป็นข้อบังคับในการใช้งาน

3.3.1.2 แก้ไขการทำงานโดยให้ส่งข้อมูลออกทันทีเมื่อเริ่มทำงานโดยไม่รอรับคำสั่งใดๆ

3.3.1.3 แก้ไขรูปแบบข้อมูลที่ส่งออกโดยอยู่ในรูปแบบของ "#,R,P,Y, Ax,Ay,Az,"

3.3.1.4 ทำการปรับค่ามาตรฐาน (Calibrate) ให้เป็นค่าที่ควรจะเป็น

เมื่อทำการแก้ไขเสร็จแล้วลำดับการทำงานของ 9DOF Razor IMU จะเป็นไปตามลำดับขั้นตอนในรูปที่ 3.5 โดยเป็นกลุ่มของขั้นตอนการทำงานที่เขียนขึ้นเพื่อให้ง่ายต่อการทำความเข้าใจเท่านั้น ไม่ใช่ขั้นตอนการทำงานทั้งหมดที่เกิดขึ้นภายใน IMU แต่อย่างใด

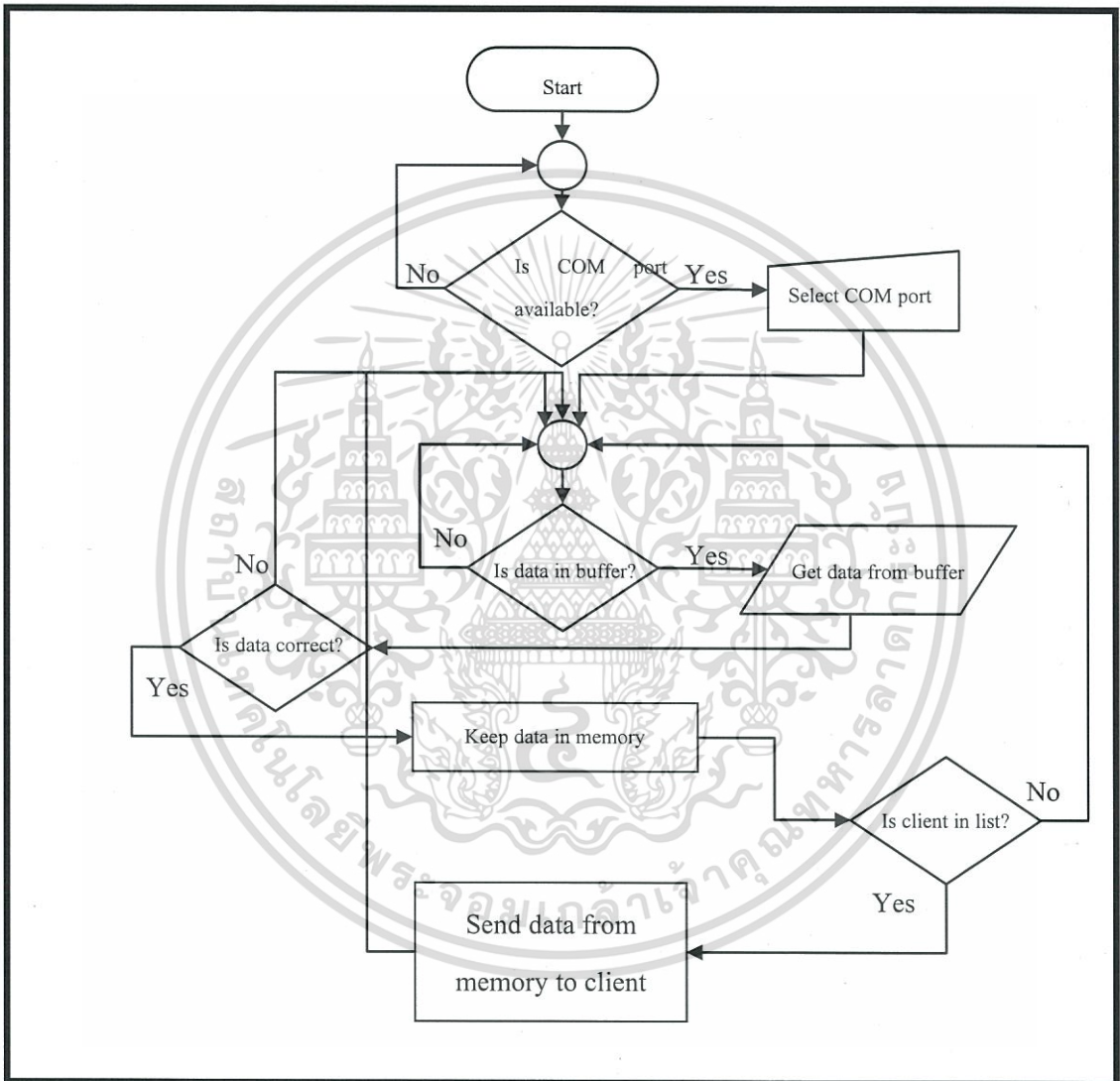


รูปที่ 3.5 ผังการทำงานของ AHRS Razor IMU firmware ที่ทำการปรับแต่งแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 การเชื่อมต่อสื่อสารของระบบ

ในส่วนของโปรแกรมประยุกต์บนคอมพิวเตอร์ซึ่งพัฒนาด้วยโปรแกรม Processing2 นั้น ถูกแบ่งออกเป็น 2 ส่วนคือ ส่วนติดต่อ COM port และ ส่วนแสดงผลข้อมูล โดยทั้ง 2 ส่วนติดต่อกันด้วย TCP/IP protocol ซึ่งใช้หมายเลข IP Address ในการเชื่อมต่อ โดยส่วนแสดงผลจะต้องระบุหมายเลข IP ของส่วนติดต่อ COM port ในกล่องข้อความก่อนกดปุ่มเริ่มต้นทำงาน



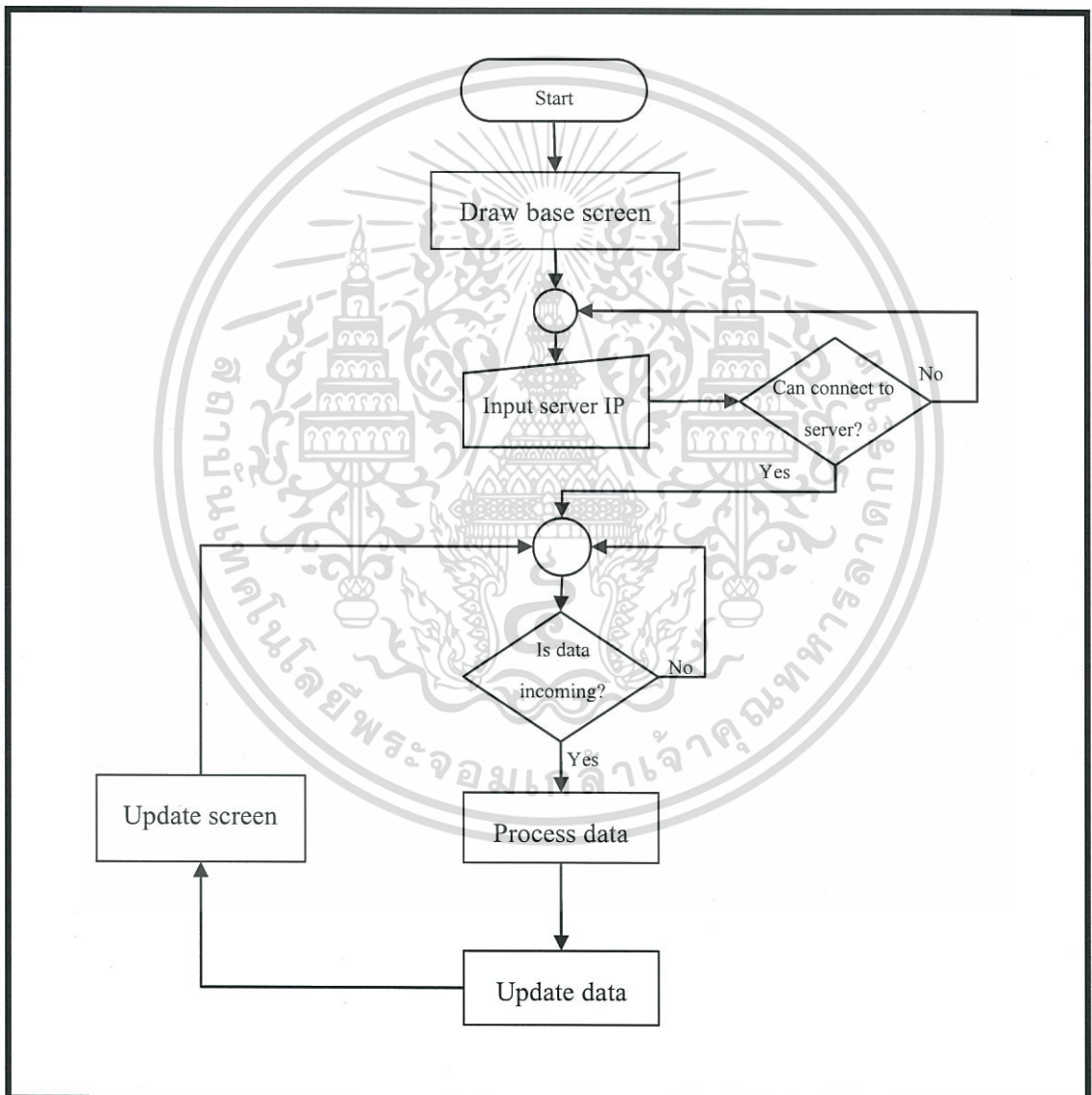
รูปที่ 3.6 ผังการทำงานของโปรแกรมส่วนติดต่อ COM port

ลำดับขั้นตอนการทำงานของโปรแกรมในส่วนติดต่อ COM Port จะเป็นดังรูปที่ 3.6 โดยเริ่มต้นจากการตรวจสอบ COM Port ในเครื่อง PC จากนั้นตรวจสอบว่า COM Port ที่ผู้ใช้เลือกมีการรับข้อมูลอยู่หรือไม่ ถ้ามีข้อมูลอยู่จะทำการดึงข้อมูลดังกล่าวมาตรวจสอบความถูกต้อง จากนั้นจะทำการตรวจสอบว่ามีการเชื่อมต่อกับส่วนแสดงผลอยู่หรือไม่ ถ้ามีจะทำการส่งข้อมูลที่เก็บไว้ไปยังผู้เชื่อมต่อนั้น หากไม่มีผู้เชื่อมต่อ ก็จะเริ่มการทำงานใหม่ตั้งแต่การตรวจสอบการรับข้อมูล เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น มิฉะนั้น ผู้ใช้จะประสบปัญหาการดำเนินงานที่ไม่ราบรื่นใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3 ส่วนประมวลผลและการแสดงผล

โปรแกรมส่วนนี้พัฒนาขึ้นบน Processing2 โดยทำหน้าที่เป็นส่วนประมวลผลข้อมูลและส่วนแสดงผลข้อมูลในตัวเดียวกัน โดยข้อมูลที่ใช้ในการทำงานจะเป็นข้อมูลที่รับจากส่วนติดต่อ COM Port ซึ่งโปรแกรมส่วนนี้ต้องทำการเชื่อมต่อไปยังส่วนติดต่อ COM Port ผ่าน IP Address จึงจะสามารถรับข้อมูลได้โดยแบ่งการทำงานเป็นสองส่วนย่อยได้แก่ ส่วนการแสดงผล และส่วนการประมวลผล

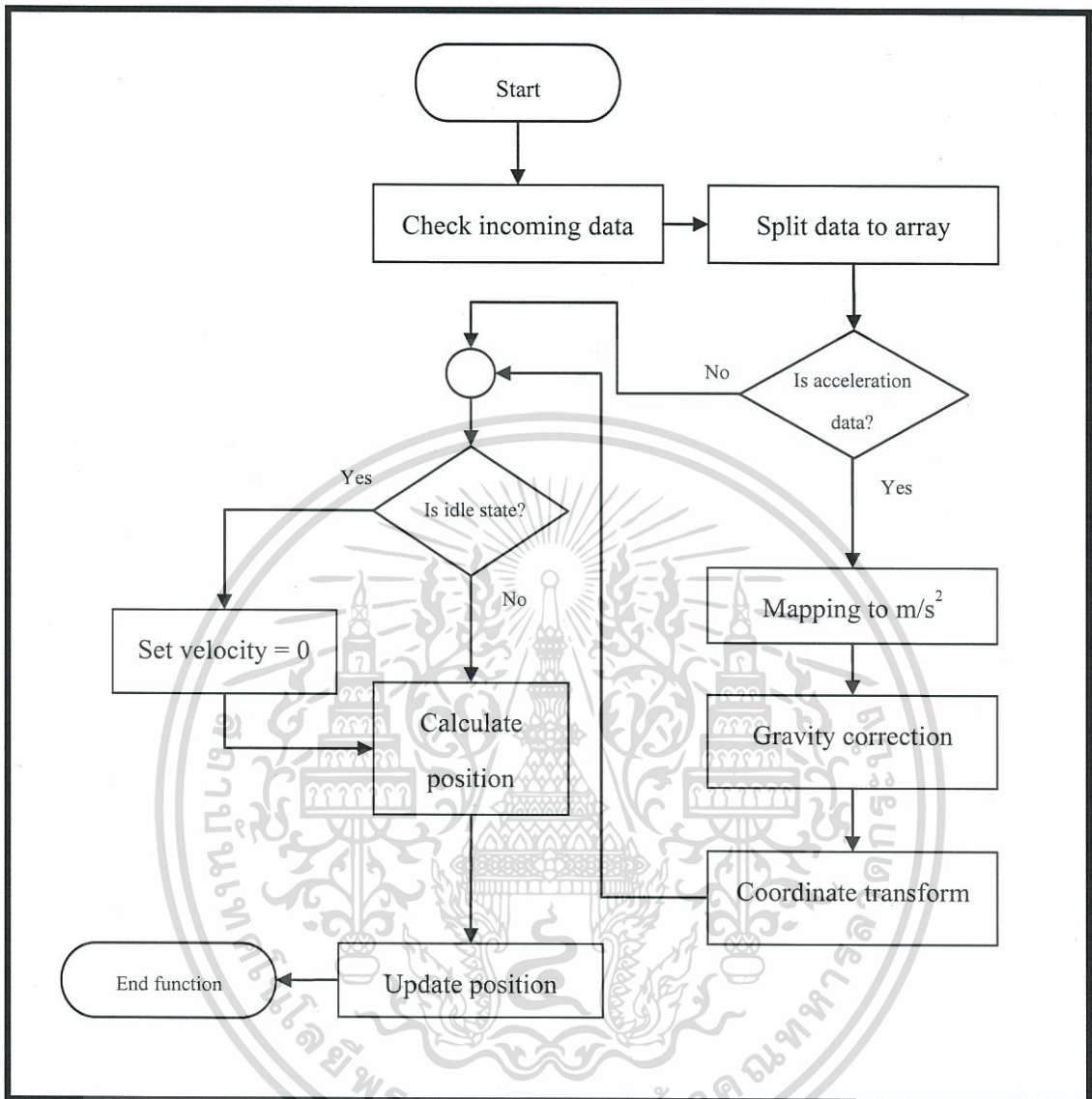
การทำงานของส่วนการแสดงผลสามารถอธิบายได้ด้วยผังการทำงานตามรูปที่ 3.7 ดังนี้



รูปที่ 3.7 ผังการทำงานของโปรแกรมส่วนแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของส่วนการประมวลผลสามารถอธิบายได้ด้วยผังการทำงานตามรูปที่ 3.8 ดังนี้



รูปที่ 3.8 ผังการทำงานของส่วนประมวลผลในโปรแกรมแสดงผล

ขั้นตอนในการประมวลผลข้อมูลที่ได้รับจากระบบไร้สาย เมื่อผ่านการตรวจสอบแล้ว จะถูกดำเนินการตามขั้นตอนการประมวลผลในรูปที่ 3.8 เพื่อทำการปรับค่าข้อมูลและทำการคำนวณ เมื่อสิ้นสุดการทำงานของส่วนประมวลผล ค่าที่ได้จะถูกเก็บไว้ในตัวแปรของชุดข้อมูลเพื่อให้ส่วนแสดงผลเรียกใช้ในการแสดงผลตามลำดับขั้นตอนการทำงานต่อไป

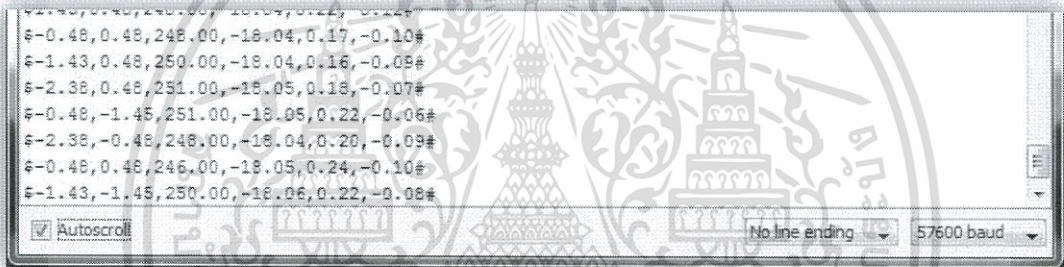
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การรับข้อมูลและการแสดงผล (DATA INPUT AND DISPLAY)

ในส่วนนี้จะเป็นการทดสอบการทำงานและความถูกต้องของส่วนต่างๆในภายในระบบ เพื่อให้สามารถทำงานร่วมกันได้อย่างถูกต้อง

3.4.1 การอ่านค่าจาก IMU ด้วย Serial monitor ใน Arduino IDE

การทดสอบนี้จัดทำขึ้น โดยการวางชุดทดสอบในรูปที่ 3.4 ไว้ห่างจากภาครับส่งไร้สาย (Wireless transceiver) ฝั่งรับประมาณ 2 เมตร แล้วทำการเปิดเครื่องมือ Serial monitor ในโปรแกรม Arduino IDE เพื่อทำการอ่านค่า โดยโปรแกรม Arduino IDE มีเครื่องมือช่วยในการอ่านค่าจาก COM port ที่ชื่อว่า Serial monitor ซึ่งสามารถอ่านค่าได้โดยการเลือก COM port และ baud rate ในหน้าต่างโปรแกรม



รูปที่ 3.9 ข้อมูลที่อ่านได้จาก IMU ด้วย Serial monitor ใน โปรแกรม Arduino IDE

จากรูปที่ 3.9 ค่าที่อ่านได้ต่อ 1 บรรทัด มีความหมายดังนี้

\$ เป็นเครื่องหมายแสดงจุดเริ่มต้นของชุดข้อมูล

, เป็นตัวคั่นข้อมูลแต่ตัว โดยข้อมูลในแต่ละตัวคือค่าความเร่งในแกน x, y, z และ ค่ามุม Roll, Pitch, Yaw ตามลำดับ

เป็นเครื่องหมายแสดงจุดสิ้นสุดของข้อมูล

จากการที่สามารถอ่านค่าผ่าน Serial monitor ได้ ทำให้ทราบว่า IMU สามารถส่งข้อมูลได้ตามที่กำหนด และ ชุดทดสอบในรูปที่ 3.4 สามารถทำงานได้เป็นปกติโดยข้อมูลที่ส่งผ่านเครือข่ายไร้สายไม่มีความผิดพลาด

3.4.2 การอ่านค่าจาก IMU ด้วย Processing2

การทดสอบนี้จัดทำขึ้นโดยการวางชุดทดสอบในรูปที่ 3.4 ไว้ห่างจากภาครับส่งไร้สาย (Wireless transceiver) ฝั่งรับประมาณ 2 เมตร แล้วเริ่มต้นการทำงานของโปรแกรมทดสอบที่เขียนขึ้นใน Processing2 โดยเป็นโปรแกรมที่ปรับแก้มาจากชุดโปรแกรมตัวอย่างที่มีให้สำหรับการศึกษาด้วยตนเอง ซึ่งเป็นส่วนหนึ่งของโปรแกรม Processing 2

โปรแกรม Processing2 มีชุดคำสั่งสำหรับการอ่านค่าจาก COM port เตรียมไว้เรียบร้อยแล้ว ผู้ใช้เพียงแค่นำเข้าชุดคำสั่ง และ กำหนดค่าในฟังก์ชันการเชื่อมต่อให้ถูกต้อง เมื่อมีข้อมูลเข้ามาทาง COM port โปรแกรมจะเก็บไว้ใน Buffer ซึ่งผู้ใช้ต้องเขียนคำสั่งเพื่อดึงข้อมูลออกมาใช้งานเอง

```

$-1.43,-1.45,251.00,-18.32,0.30,-0.18#
$-0.48,-1.45,250.00,-18.33,0.30,-0.20#
$-1.43,0.48,251.00,-18.31,0.28,-0.22#
$-2.38,-1.45,251.00,-18.31,0.28,-0.19#
$-1.43,-1.45,252.00,-18.31,0.31,-0.20#
$-0.48,-2.42,252.00,-18.31,0.31,-0.22#
$-1.43,0.48,248.00,-18.33,0.29,-0.26#
$-2.38,0.48,252.00,-18.33,0.29,-0.22#
107

```

รูปที่ 3.10 ข้อมูลที่อ่านได้จาก IMU ในโปรแกรม Processing2

จากรูปที่ 3.10 ค่าที่อ่านได้ใน 1 บรรทัด มีความหมายดังนี้

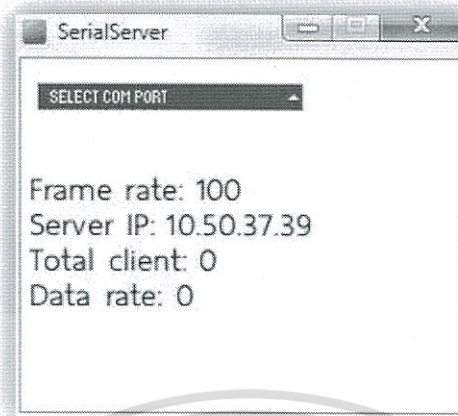
\$ เป็นเครื่องหมายแสดงจุดเริ่มต้นของชุดข้อมูล

, เป็นตัวคั่นข้อมูลแต่ละตัว โดยข้อมูลในแต่ละตัวคือค่าความเร่งในแกน x, y, z และ ค่ามุม Roll, Pitch, Yaw ตามลำดับ

เป็นเครื่องหมายแสดงจุดสิ้นสุดของข้อมูล

จากรูปที่ 3.10 จะเห็นว่าค่าจาก IMU ที่เข้ามาเหมือนกับที่อ่านได้ใน Serial monitor ของโปรแกรม Arduino IDE โดยค่าที่อ่านได้คือชุดข้อมูลรูปแบบเดียวกับการทดสอบก่อนหน้าทำให้สามารถพัฒนาโปรแกรมในส่วนติดต่อ COM Port ตามที่ออกแบบไว้ได้

3.4.3 การแสดงผลในโปรแกรมประยุกต์ที่พัฒนาขึ้น



รูปที่ 3.11 GUI ของโปรแกรมส่วนติดต่อ COM Port

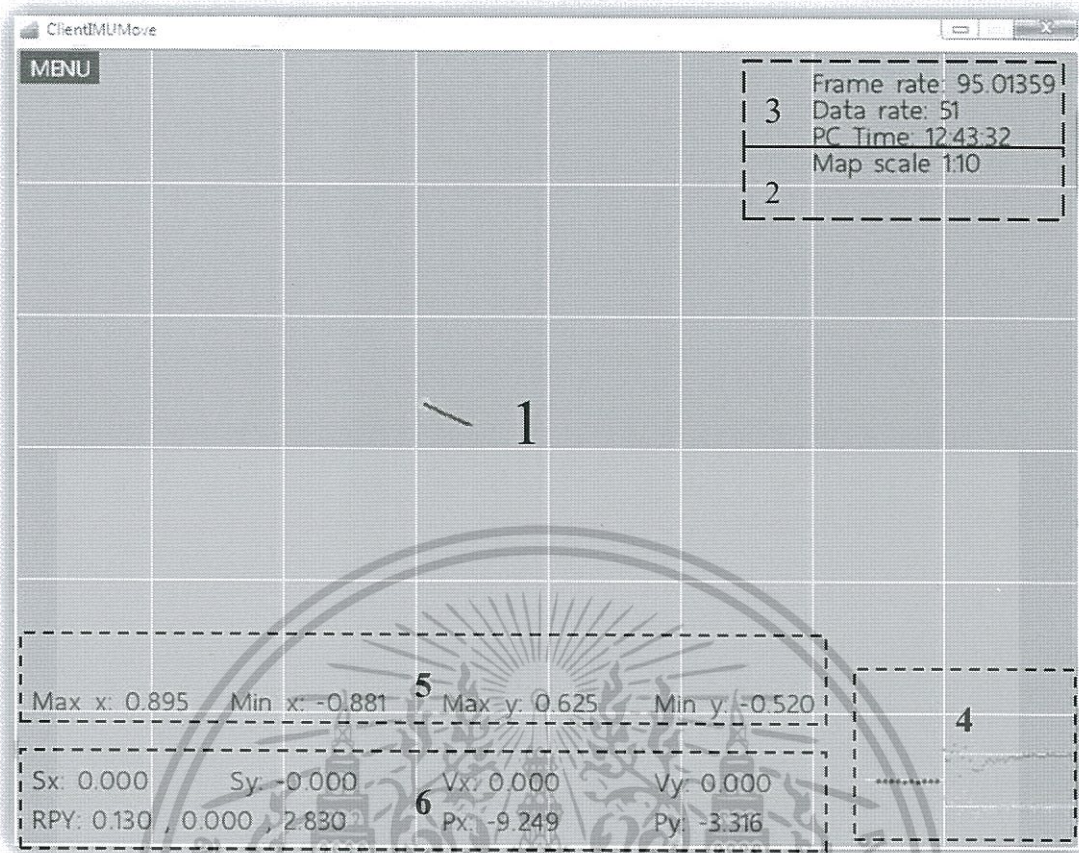
จากรูปที่ 3.11 ส่วนแรกทางด้านบนของโปรแกรมส่วนติดต่อ COM Port จะมีรายการ COM Port ให้เลือกใช้เพื่อเริ่มการติดต่อ โดยเมื่อเลือกแล้วโปรแกรมจะทำการติดต่อกับ COM Port นั้นทันที ข้อมูลสำคัญที่แสดงในหน้าต่างโปรแกรมคือ Server IP ซึ่งเป็นข้อมูลที่ส่วนแสดงผลต้องใช้ในการเชื่อมต่อเพื่อขอรับข้อมูลจากส่วนติดต่อ COM Port โดยข้อมูลที่ส่งไปยังส่วนแสดงผลเป็นข้อมูลที่รับมาจาก Wireless transceiver ซึ่งก็คือชุดข้อมูลที่ส่งมาจาก IMU นั่นเอง



รูปที่ 3.12 GUI ของโปรแกรมส่วนแสดงผลในหน้าแรก

จากรูปที่ 3.12 โปรแกรมส่วนแสดงผลเมื่อเริ่มต้นการทำงานจะมีกล่องข้อความให้ใส่ข้อมูลหมายเลข IP Address ที่ใช้ในการติดต่อกับโปรแกรมส่วนติดต่อ COM Port โดยค่าเริ่มต้นจะเป็น 127.0.0.1 ซึ่งจะใช้ในกรณีที่โปรแกรมส่วนติดต่อ COM Port กับโปรแกรมส่วนแสดงผลทำงานอยู่บนคอมพิวเตอร์เครื่องเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 GUI ของโปรแกรมส่วนแสดงผลการติดตามตำแหน่งการเคลื่อนที่

ในรูปที่ 3.13 เป็นการแสดงผลข้อมูลการติดตามตำแหน่งการเคลื่อนที่ด้วยโปรแกรมประยุกต์ที่พัฒนาขึ้นในโปรแกรม Processing 2 เพื่อใช้ในการติดตามตำแหน่งการเคลื่อนที่ของรถจำลองตามวัตถุประสงค์ของโครงงาน โดยในหน้าแสดงผลจะประกอบด้วยส่วนต่าง ๆ แบ่งตามหมายเลขกำกับในรูปที่ 3.13 ได้ดังนี้

- 1) ส่วนพื้นที่แสดงผลข้อมูลการติดตามตำแหน่ง ครอบคลุมพื้นที่ทั้งหมดในหน้าแสดงผล มีเส้นตารางกำกับเพื่อใช้ในการอ้างอิงระยะการเคลื่อนที่ ตำแหน่งปัจจุบันของรถจำลองคือจุดใหญ่สีเขียว ตำแหน่งในอดีตคือจุดเล็กสีน้ำเงิน การเคลื่อนที่ในครั้งแรกจะเริ่มจากจุดกึ่งกลาง
- 2) ค่าอัตราส่วนของพื้นที่ในโปรแกรมต่อพื้นที่จริง เช่น 1:10 หมายถึง 1 ช่องตารางในโปรแกรมจะอ้างอิงกับพื้นที่ขนาด 10 x 10 เมตร ทั้งในแนวแกน x และ แกน y เป็นต้น
- 3) ข้อมูลระบบ ได้แก่ อัตราเฟรมในการแสดงผล อัตรารับข้อมูล และเวลาปัจจุบัน
- 4) กราฟความเร่ง สีแดงคือค่าในแกน x และ สีเขียวคือค่าในแกน y ส่วนจุดสีน้ำเงินคือค่าขนาดของความเร่งที่คำนวณได้จากสมการที่ (2.3)
- 5) ข้อมูลความเร่งสูงสุดและต่ำสุดในแกน x และ แกน y จาก 50 ชุดข้อมูลล่าสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

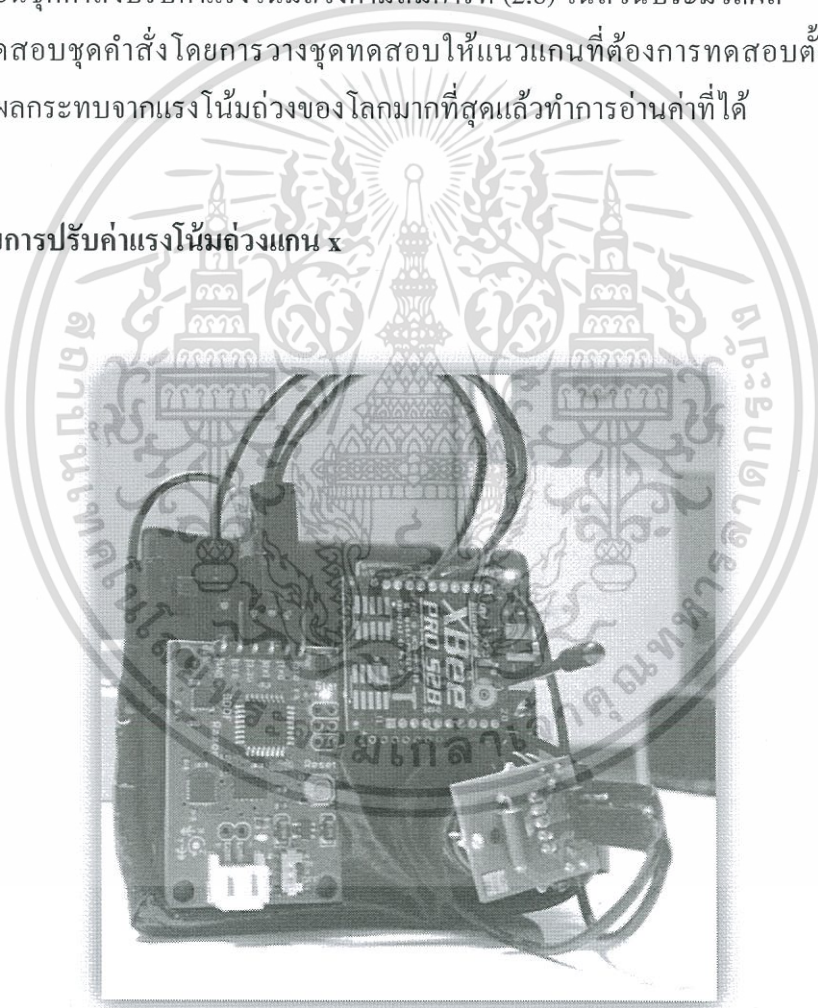
6) ค่าอื่นๆในระบบ ได้แก่ ค่าจาก AHRS (RPY) ค่าตำแหน่งเพื่อแสดงผล (P_x, P_y) ค่าตำแหน่งปัจจุบันที่คำนวณได้ (S_x, S_y) และ ค่าความเร็วปัจจุบันที่คำนวณได้ (V_x, V_y) เป็นต้น

3.5 การปรับค่าแรงโน้มถ่วง

สำหรับส่วนนี้จะทำการทดสอบการตัดแรงโน้มถ่วงออกจากค่าความเร่งที่วัดได้ โดยค่าที่อ่านได้จะต้องไม่เกิน ± 0.3 ซึ่งเป็นค่าประมาณของค่ารบกวนที่เกิดขึ้นในระบบขณะทำการทดสอบ หากมีความเร่งเกิดขึ้นเกินกว่านี้ความเร่งนั้นจะเป็นค่าแรงโน้มถ่วงของโลกซึ่งมีค่าประมาณ 9.81m/s^2 ตามที่ได้กำหนดไว้ในสมการการปรับค่าแรงโน้มถ่วง โดยขั้นตอนในการทดสอบคือ

- เขียนชุดคำสั่งปรับค่าแรงโน้มถ่วงตามสมการที่ (2.8) ในส่วนประมวลผล
- ทดสอบชุดคำสั่งโดยการวางชุดทดสอบให้แนวแกนที่ต้องการทดสอบตั้งฉากกับพื้น เพื่อให้ได้รับผลกระทบจากแรงโน้มถ่วงของโลกมากที่สุดแล้วทำการอ่านค่าที่ได้

3.5.1 ทดสอบการปรับค่าแรงโน้มถ่วงแกน x



รูปที่ 3.14 การทดสอบการอ่านค่าตามแนวแกน X

ทำการทดสอบโดยการวางชุดทดสอบให้แนวแกน X ของ IMU ตั้งฉากกับพื้นตามรูปที่ 3.14 จากนั้นทำการอ่านค่าใน โปรแกรมโดยแสดงผลเป็นกราฟเส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Max x: 0.124 Min x: -0.123 Max y: -0.030 Min y: -0.292

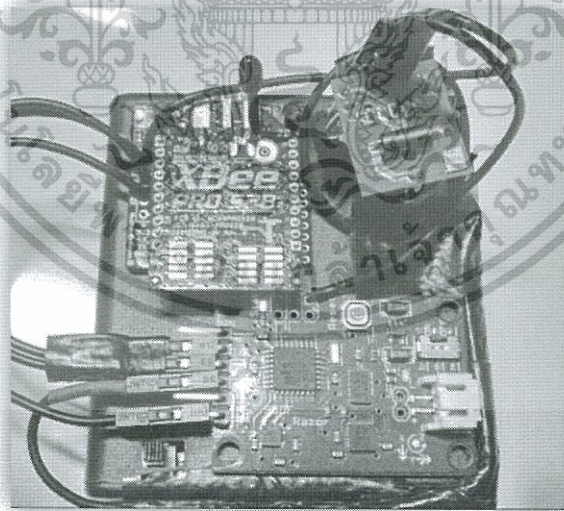
RPY: 1.58 , -0.02 , 0.21

Vx: 0.000 Vy: 0.000

รูปที่ 3.17 ผลการทดสอบการอ่านค่าตามแนวแกน Y

ในรูปที่ 3.17 จากค่าที่อ่านได้ (เส้นสีเขียว) จะเห็นว่าไม่มีความเร่งเกิดขึ้นในแกนดังกล่าว เนื่องจากถูกปรับเอาความเร่งที่เกิดจากแรงโน้มถ่วงของโลกออกแล้ว จึงเหลือเพียงความเร่งที่เกิดขึ้นจากการเคลื่อนที่ และค่ารบกวนเท่านั้น โดยค่าอยู่ในช่วงไม่มีความเร่งเช่นเดียวกับแกน X

3.5.3 ทดสอบการปรับค่าแรงโน้มถ่วงแกน Z



รูปที่ 3.18 การทดสอบการอ่านค่าตามแนวแกน Z

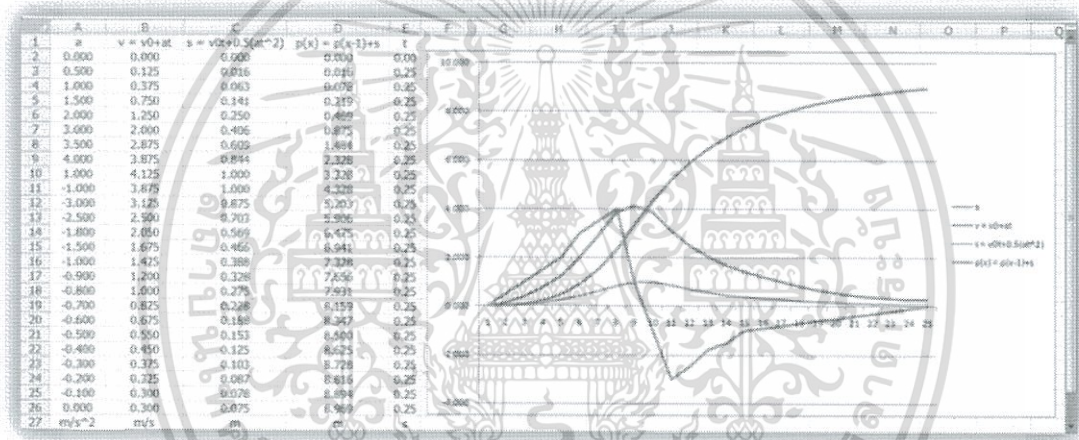
ทำการทดสอบเหมือนกับแกน X และ Y โดยการวางชุดทดสอบให้แนวแกน Z ของ IMU ตั้งฉากกับพื้นตามรูปที่ 3.18 ซึ่งค่าที่อ่านได้อยู่ในช่วงไม่มีความเร่งเช่นเดียวกับแกน X และ Y จึงไม่มีการนำรูปกราฟมาแสดงเนื่องจากกราฟที่ได้เหมือนกันกับ 2 การทดสอบก่อนหน้านี้ทุกประการ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 การคำนวณค่าตำแหน่งและทิศทาง

สำหรับส่วนนี้จะเป็นการทดสอบความถูกต้องของสมการที่ใช้ในการคำนวณ ความเร็ว และ ตำแหน่ง ด้วยค่าความเร่งที่กำหนดขึ้นเองจำนวน 25 ชุดข้อมูล คำนวณความเร็วด้วยสมการที่ (2.1) และคำนวณตำแหน่งด้วยสมการที่ (2.2) โดยกำหนดให้ $t = 0.25$ วินาที

3.6.1 การคำนวณในโปรแกรม MS Office Excel

ทำการทดสอบโดยการนำข้อมูลที่กำหนดขึ้นเอง 25 ชุด มาทำการคำนวณด้วยสมการที่ (2.1) และ สมการที่ (2.2) โดยกำหนดให้ $t = 0.25$ เมื่อนำค่าที่คำนวณได้มาแสดงผลเป็นกราฟจะมีรูปร่างของกราฟตามรูปที่ 3.19 โดยมีรายละเอียดดังนี้



รูปที่ 3.19 การทดสอบการคำนวณในโปรแกรม MS Office Excel

เส้นสีฟ้า คือค่าความเร่ง a ที่กำหนดขึ้นซึ่งมีทั้งช่วงที่เป็นค่าบวก และ ช่วงที่เป็นค่าลบ เมื่อนำมาคำนวณด้วยสมการที่ (2.1) จะได้เป็นค่าความเร็ว

เส้นสีแดง คือค่าความเร็ว โดยจะเห็นการเปลี่ยนแปลงของค่าความเร็วเป็นไปในทิศทางเดียวกับความเร่ง จึงสรุปได้ว่าสมการที่ใช้มีความถูกต้อง

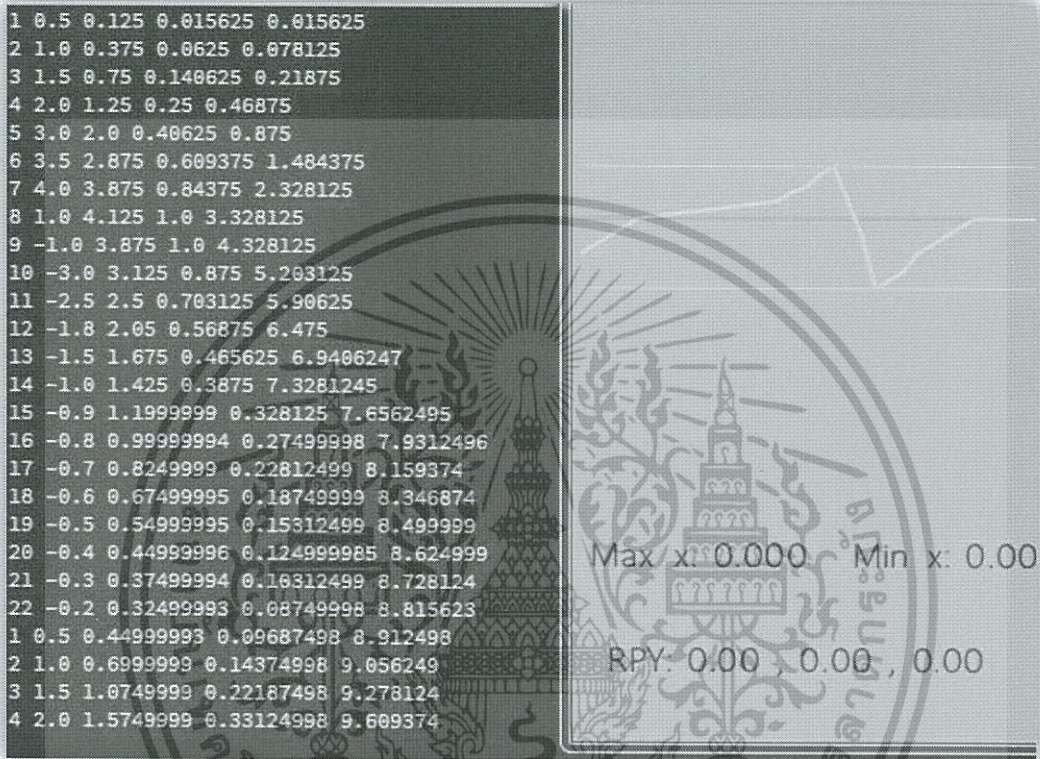
เส้นสีเขียว คือระยะทางที่มีการเคลื่อนที่ในหนึ่งช่วงเวลาซึ่งคำนวณ โดยการนำเอาสมการที่ (2.2) มาตัดส่วนที่บวกตำแหน่งก่อนหน้าออกจึงเหลือแค่ส่วนที่คำนวณระยะที่เคลื่อนที่ได้ในหนึ่งช่วงเวลาซึ่งกราฟจะเป็นไปในทิศทางเดียวกันกับความเร็ว

เส้นสีม่วง คือตำแหน่งปัจจุบันซึ่งได้จากการคำนวณตามสมการที่ (2.2) ซึ่งคำนวณโดยการนำเอาตำแหน่งก่อนหน้ามาบวกกับระยะที่เคลื่อนที่ในช่วงเวลาปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.2 การคำนวณในโปรแกรม Processing2

ทำการทดสอบโดยการเขียนชุดคำสั่งสำหรับการคำนวณด้วยสมการที่ (2.1) และสมการที่ (2.2) ในโปรแกรมส่วนประมวลผลแล้วทำการแสดงผลค่าที่คำนวณได้ทาง Console ของโปรแกรม Processing 2 และแสดงผลความเร่งเป็นกราฟเส้นใน โปรแกรมแสดงผลตามรูปที่ 3.20



รูปที่ 3.20 การทดสอบการคำนวณใน โปรแกรม Processing2

จากข้อมูลในรูปที่ 3.20 ด้านซ้ายเป็นค่าที่คำนวณได้จากสมการ และด้านขวาเป็นกราฟความเร่งที่แสดงในส่วนแสดงผล โดยข้อมูลในรูปด้านซ้ายแบ่งออกเป็น 5 ชุด คือ

ชุดที่ 1 เป็นลำดับของข้อมูล มีค่า 0 - 24 แต่โปรแกรมนำมาคำนวณแค่ 1 - 22 เพราะมีการตั้งข้อกำหนดไว้ว่า หากความเร่งต่ำกว่า ± 0.1 จะไม่นำมาคำนวณ

ชุดที่ 2 เป็นข้อมูลความเร่งที่กำหนดขึ้น

ชุดที่ 3 เป็นค่าความเร็วซึ่งคำนวณจากสมการที่ (2.1) และจะถูกนำไปใช้ในสมการที่ (2.2) ในช่วงเวลาถัดไป

ชุดที่ 4 เป็นค่าระยะที่เคลื่อนที่ได้ในช่วงเวลาปัจจุบัน

ชุดที่ 5 เป็นตำแหน่งที่ได้จากการคำนวณตามสมการที่ (2.2) ซึ่งมาจากตำแหน่งก่อนหน้าบวกกับระยะที่เคลื่อนที่ได้ในช่วงเวลาปัจจุบัน (ข้อมูลชุดที่ 4)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากข้อมูลดังกล่าวสรุปได้ว่า สมการที่ใช้มีความถูกต้อง ทั้งสมการที่ (2.1) และสมการที่ (2.2) ซึ่งให้ค่าออกมาอย่างที่ควรจะเป็น แต่การตัดชุดข้อมูลบางชุดที่อยู่ในช่วงค่ารบกวนที่ตั้งไว้อาจส่งผลต่อค่าโดยรวมทำให้ความถูกต้องของค่าตำแหน่งที่ควรจะเป็นคลาดเคลื่อนเล็กน้อย ซึ่งกรณีดังกล่าวจะเกิดขึ้นก็ต่อเมื่อมีการเคลื่อนที่ด้วยความเร็วคงที่ หรือ อยู่ในสภาวะหยุดนิ่ง เนื่องจากทั้งสองสถานการณ์นั้นค่าความเร่งที่อ่านได้จาก IMU จะมีค่าเข้าใกล้ 0 แต่โอกาสที่จะเกิดขึ้นระหว่างการเคลื่อนที่มีน้อยมาก เพราะต้องมีค่าที่เข้าใกล้ 0 ติดกันจนค่าเฉลี่ยที่ใช้ในระบบตรวจสอบสภาวะหยุดนิ่งมีค่าต่ำกว่าค่าที่กำหนดเอาไว้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.3 การคำนวณทิศทางทางการเคลื่อนที่

สำหรับในส่วนนี้จะเป็นการทดสอบการคำนวณทิศทางทางการเคลื่อนที่ซึ่งแสดงผลในโปรแกรมส่วนแสดงผล โดยใช้ค่าตำแหน่งในการแสดงผลจากสมการที่ (2.11) และสมการที่ (2.12) ในการแสดงผลในโปรแกรม

ในการทดสอบการคำนวณทิศทางครั้งนี้จะให้รถจำลองวิ่งวนเป็นวงกลมเพื่อดูค่าทิศทางในการเคลื่อนที่ที่แสดงในโปรแกรมส่วนแสดงผลว่าทิศทางที่คำนวณได้ตรงกับการเคลื่อนที่จริงหรือไม่ อย่างไร และ มีความคลาดเคลื่อนมากน้อยเพียงใด



รูปที่ 3.21 การทดสอบการคำนวณทิศทางทางการเคลื่อนที่

จากรูปที่ 3.21 จะเห็นว่าทิศทางทางการเคลื่อนที่ในโปรแกรมแสดงผลมีการเคลื่อนที่เป็นวงกลมตามการเคลื่อนที่จริงของรถจำลองในการทดสอบการติดตามตำแหน่งในครั้งนี้

จากผลการทดสอบนี้สามารถสรุปได้ว่า ค่าจากสมการที่ (2.11) และสมการที่ (2.12) สามารถใช้แสดงผลทิศทางในการเคลื่อนที่ของรถจำลองได้ถูกต้อง ถ้าหากค่าทิศทางจาก AHRS มีความถูกต้อง เนื่องจากในการคำนวณทิศทางมีการใช้ค่าอ้างอิงทิศทางจาก AHRS เพียงอย่างเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

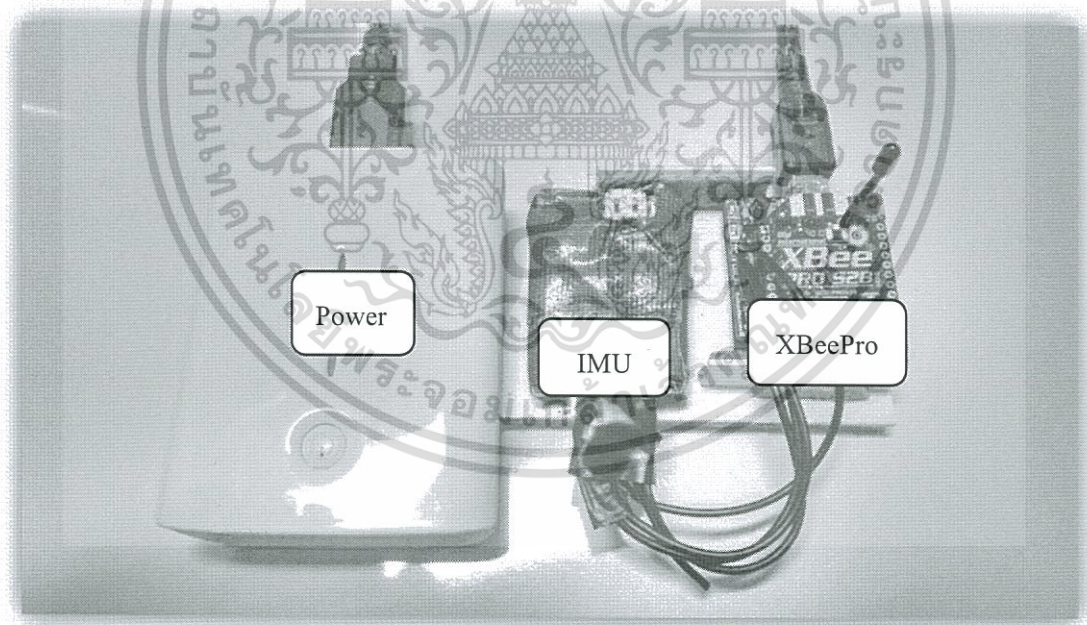
การทดสอบระบบ

สำหรับบทนี้จะเป็นการทดสอบระบบติดตามตำแหน่งการเคลื่อนที่ที่ได้พัฒนาขึ้น เพื่อนำผลที่ได้มาทำการวิเคราะห์และสรุปผล

4.1 ทดสอบการอ่านค่าความเร่ง

การทดสอบนี้มีเป้าหมายในการอ่านค่าความเร่งจากชุดอุปกรณ์ในรูปที่ 4.1 เพื่อคุณลักษณะของความเร่งที่ได้จากการเคลื่อนที่ในรูปแบบของกราฟเส้น โดยใช้โปรแกรม MS Excel 2007 ในการสร้างกราฟ

ข้อมูลความเร่งที่จะนำมาสร้างกราฟเป็นชุดข้อมูลที่ได้รับมาจาก IMU ซึ่งจะถูกบันทึกเป็นไฟล์ .CSV โดยอัตโนมัติจากการทำงานของโปรแกรมส่วนติดต่อ COM Port



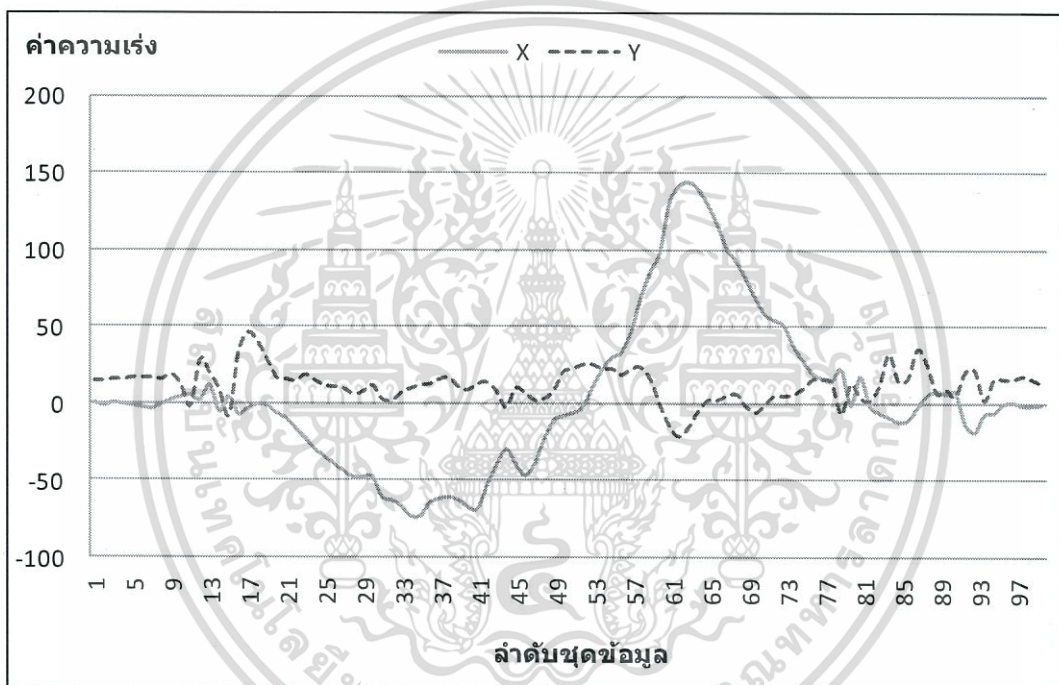
รูปที่ 4.1 ชุดอุปกรณ์การทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1 กราฟความเร่งจากการเคลื่อนที่ด้วยมือ

การทดสอบนี้มีเป้าหมายคือการอ่านค่าความเร่งจากการเคลื่อนที่ด้วยมือแล้วนำมาสร้างกราฟเพื่อคุณลักษณะของความเร่งที่เกิดขึ้นจากการเคลื่อนที่ โดยจะทำการทดสอบ 2 ครั้ง คือ ทดสอบการเคลื่อนที่ด้วยมือเป็นเส้นตรง และทดสอบการเคลื่อนที่ด้วยมือเป็นเส้นโค้ง

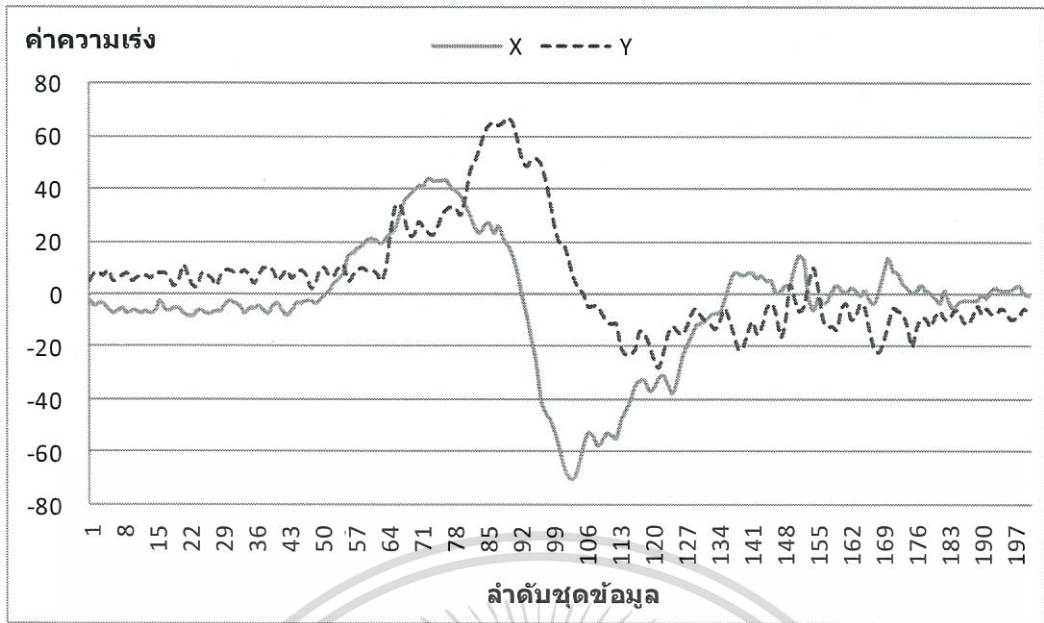
วิธีการทดสอบคือใช้มือจับชุดอุปกรณ์ทดสอบทั้งหมดแล้วเคลื่อนที่ไปบนอากาศโดยการเคลื่อนที่เป็นเส้นตรง 1 ครั้งระยะทางประมาณ 1 เมตร และเคลื่อนที่ในวิถีโค้งด้วยการเหวี่ยงแขนจากทิศ 12 นาฬิกา มายังทิศ 3 นาฬิกา จำนวน 1 ครั้ง แล้วนำค่าความเร่งที่ถูกรับบันทึกไว้มาสร้างกราฟเพื่อคุณลักษณะของกราฟความเร่งที่เกิดขึ้นจากการเคลื่อนที่ทั้งสองครั้ง



รูปที่ 4.2 กราฟความเร่งขณะเคลื่อนที่เป็นเส้นตรงด้วยมือ

จากรูปที่ 4.2 จะเห็นว่าเส้นกราฟของความเร่ง X ในช่วงชุดข้อมูลที่ 1 ถึงชุดข้อมูลที่ 53 จะมีค่าความเร่งอยู่ในช่วงลบ และมีการแกว่งของค่าเล็กน้อย ซึ่งเกิดจากตอนเคลื่อนที่ด้วยมือในช่วงชะลอความเร็วก่อนจะหยุดมีการเกร็งแขนเพื่อหยุดการเคลื่อนที่ แต่หลังจากหยุดการเคลื่อนที่แล้วยังมีค่าความเร่งเกิดขึ้นต่อในทิศทางตรงข้ามกับการเคลื่อนที่ช่วงแรก โดยชุดข้อมูลหลังจากชุดที่ 53 จะเป็นชุดข้อมูลของค่าที่เกิดขึ้นจากแรงที่ใช้ในการหยุดการเคลื่อนที่ เพราะการที่จะหยุดการเคลื่อนที่ได้มันจะต้องมีแรงต้านการเคลื่อนที่เดิมจนหักล้างกับแรงที่ทำให้เกิดการเคลื่อนที่ ซึ่งในการทดสอบครั้งนี้ได้หยุดการเคลื่อนที่ประมาณชุดข้อมูลที่ 80 ส่วนค่าที่เกิดขึ้นหลังจากนั้นอาจเกิดมาจากการสั่นของมือผู้ทดสอบขณะถือชุดอุปกรณ์ทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 กราฟความแรงขณะเคลื่อนที่เป็นวิถีโค้งด้วยมือ

จากรูปที่ 4.3 จะเห็นว่าเส้นกราฟของความแรง X และ Y ในช่วงชุดข้อมูลที่ 1 ถึงชุดข้อมูลที่ 49 จะมีการแกว่งขึ้นลงอยู่เสมอ โดยเกิดจากการสั่นของมือและค่ารับควมในระบบซึ่งขณะนั้นยังไม่มี การเคลื่อนที่ที่เกิดขึ้น แต่หลังจากชุดข้อมูลที่ 49 จะเริ่มมีการเคลื่อนที่ที่เกิดขึ้น โดยค่าของกราฟ ความแรง X จะมีทิศทางเดียวกับการเคลื่อนที่ และ กราฟความแรง Y จะเป็นค่าที่มีทิศตั้งฉากกับการ เคลื่อนที่ซึ่งเกิดจากแรงเหวี่ยงในการเคลื่อนที่เป็นวิถีโค้ง โดยการเคลื่อนที่ที่เกิดขึ้นระหว่างชุดข้อมูล ที่ 49 ถึงประมาณชุดข้อมูลที่ 137 โดยเป็นการเคลื่อนที่ที่มีความแรงเชิงมุมเกิดขึ้น

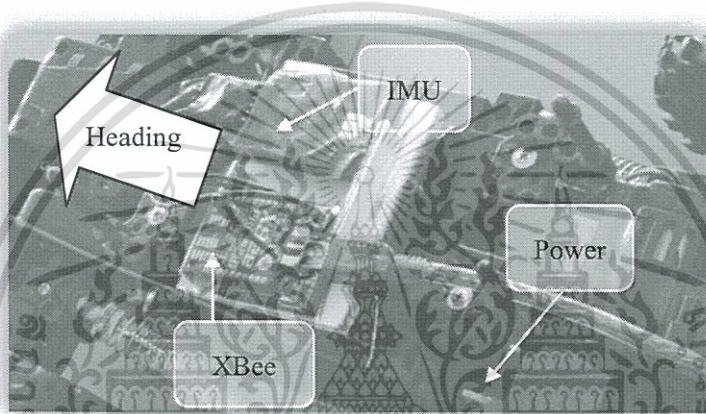
สำหรับการเคลื่อนที่เป็นวิถีโค้งนี้ค่ากราฟความแรง X ยังคงมีลักษณะเหมือนกันกับการ เคลื่อนที่เป็นเส้นตรงแต่มีทิศทางสลับกัน ส่วนกราฟความแรง Y จะแตกต่างกันกับการเคลื่อนที่เป็น เส้นตรง เพราะในการเคลื่อนที่เป็นวิถีโค้งจะมีความแรงเกิดขึ้นทั้งในแนวการเคลื่อนที่ และ ใน แนวตั้งฉากกับการเคลื่อนที่ ซึ่งในที่นี้คือกราฟความแรง Y เพราะมีความแรงที่เกิดขึ้นจากแรงหนี ศูนย์กลางในการเคลื่อนที่เป็นวิถีโค้ง แต่ลักษณะของกราฟจะคล้ายกันกับกราฟความแรงตามแนว การเคลื่อนที่ ส่วนค่าความแรงที่เกิดขึ้นจะมากหรือน้อยขึ้นอยู่กับแรงเหวี่ยงที่เกิดขึ้นในการเคลื่อนที่

สรุปผลการทดสอบการอ่านค่ากราฟความแรงจากการเคลื่อนที่ด้วยมือได้ว่า ในการเคลื่อนที่ ด้วยมือนั้นแรงสั่นจากมือส่งผลต่อความแรงที่ IMU อ่านได้แต่ในระหว่างการเคลื่อนที่ ถ้าไม่มีการ กระตุกหรืออาการเกรงระหว่างเคลื่อนที่แล้วค่าความแรงที่อ่านได้เมื่อดูจากกราฟจะมีลักษณะ เป็นกราฟเชิงเส้นที่ต่อเนื่องกันไม่มีการแกว่งของค่าความแรงมากนัก ทำให้ค่ากราฟที่ได้ค่อนข้าง ต่อเนื่องกันจนถึงจุดยอดของกราฟความแรง

4.1.2 กราฟความเร่งจากการเคลื่อนที่ของรถจำลอง

การทดสอบนี้มีเป้าหมายคือการอ่านค่าความเร่งจากการเคลื่อนที่ของรถจำลองที่ติดตั้งชุดอุปกรณ์ติดตามตำแหน่งแล้วนำมาสร้างกราฟเพื่อดูลักษณะของความเร่งที่เกิดขึ้นจากการเคลื่อนที่ โดยจะทำการทดสอบ 2 ครั้ง คือ ทดสอบการเคลื่อนที่เป็นเส้นตรง และทดสอบการเคลื่อนที่เป็นเส้นโค้ง

ก่อนการทดสอบจะทำการติดตั้งชุดทดสอบในรูปที่ 4.1 ลงบนรถจำลอง โดยทำการติดตั้งชุด IMU และ XBee Pro ไว้ด้านบนแล้วติดส่วนจ่ายไฟไว้ด้านล่างของตัวรถเพื่อความสมดุล เมื่อทำการติดตั้งเสร็จแล้วจะเป็นดังรูปที่ 4.4



รูปที่ 4.4 รถจำลองที่ติดตั้งชุดอุปกรณ์ทดสอบแล้ว

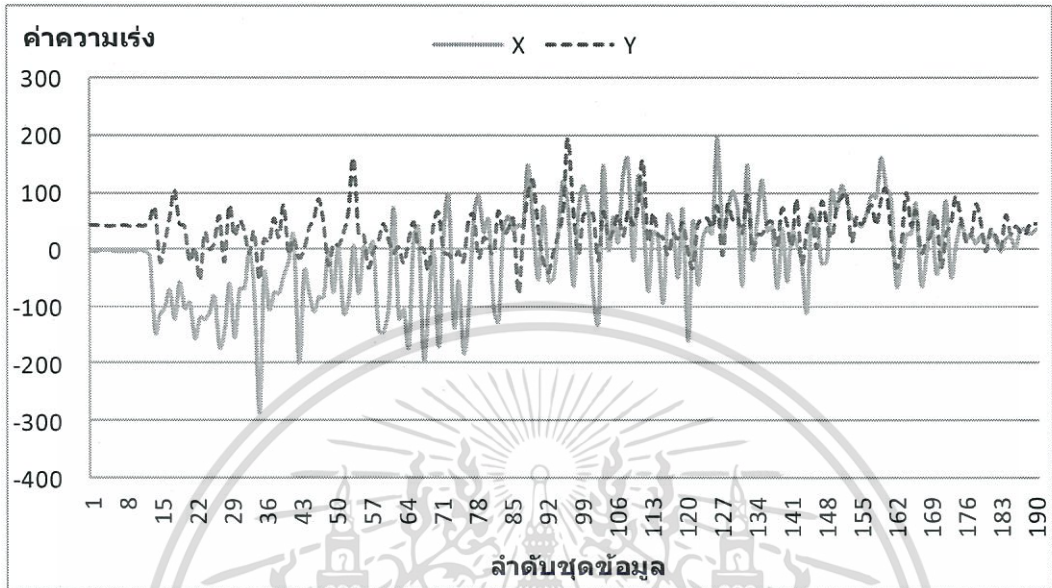
พื้นที่สำหรับการทดสอบจะเป็นพื้นที่โล่งกว้างพื้นปูด้วยกระเบื้องยาง ดังรูปที่ 4.5



รูปที่ 4.5 พื้นที่ทดสอบ

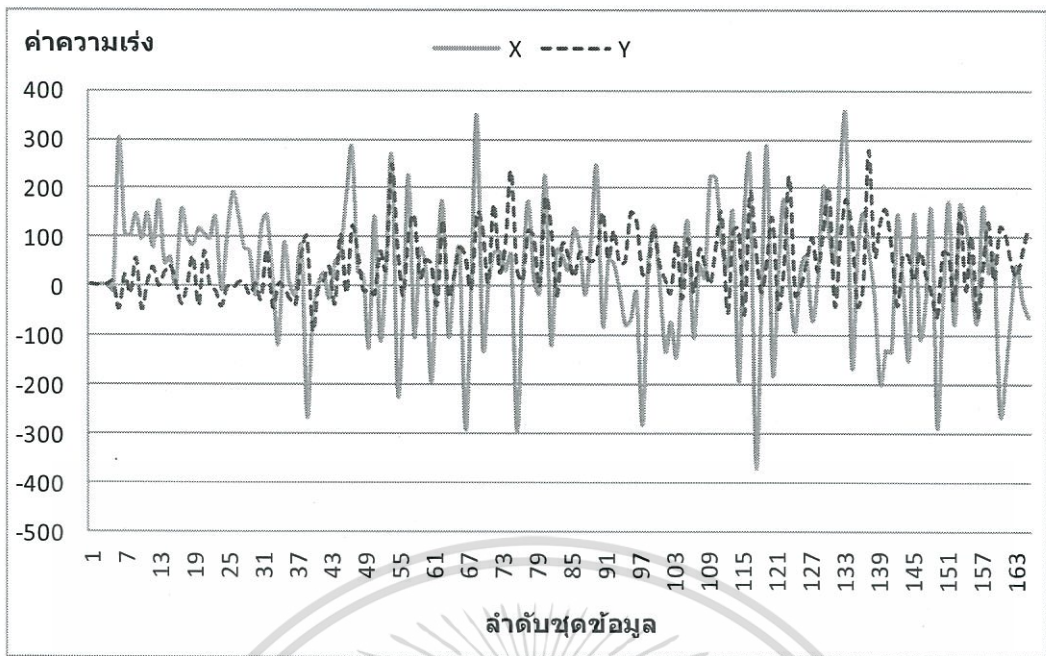
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการทดสอบคือบังคับรถให้เคลื่อนที่เป็นเส้นตรงระยะทางประมาณ 5 เมตร 1 ครั้ง และเคลื่อนที่เป็นวงกลมจำนวน 1 รอบ แล้วนำค่าความเร่งที่ถูกบันทึกไว้มาสร้างกราฟเพื่อดูลักษณะของกราฟความเร่งที่เกิดขึ้นจากการเคลื่อนที่ทั้งสองครั้งของรถจำลอง



รูปที่ 4.6 กราฟความเร่งขณะรถวิ่งเป็นเส้นตรง

จากรูปที่ 4.6 จะเห็นว่าเส้นกราฟของความเร่ง X ตลอดช่วงการเคลื่อนที่ที่มีการแกว่งของค่าสูงมากเมื่อเทียบกับการเคลื่อนที่ด้วยมือ โดยเกิดจากการเคลื่อนที่ของรถซึ่งไม่ได้เคลื่อนที่ด้วยความเร่งคงที่เพราะรถจำลองที่ใช้ทดสอบไม่สามารถควบคุมความเร็วได้ ระบบขับเคลื่อนของรถจะทำการเร่งความเร็วจนสุดกำลังขั้วของมอเตอร์อยู่เสมอในระหว่างที่กดคันเร่ง ซึ่งเมื่อความเร็วของรถขึ้นไปถึงจุดที่สุดกำลังขั้วของมอเตอร์แล้วความเร็วของรถจะเริ่มคงที่ ทำให้ความเร่งเข้าใกล้ศูนย์ แต่ในระหว่างการเคลื่อนที่ที่มีจังหวะที่ความเร่งเพิ่มขึ้นและลดลงอยู่เสมอถึงแม้จะเป็นช่วงความเร็วสูงสุดแล้วก็ตาม และเมื่อทำการปล่อยคันเร่งรถจะเริ่มลดความเร็วลงโดยการเคลื่อนที่หลังจากนี้จะเป็นการเคลื่อนที่จากแรงเฉื่อย โดยไม่มีแรงขับจากมอเตอร์ในชุดขับอีกแล้ว แต่ยังคงมีแรงต้านที่เกิดขึ้นจากชุดขับเคลื่อนทำให้ความเร่งที่อยู่ในช่วงชะลอความเร็วไม่ได้ลดลงในรูปแบบเชิงเส้นทำให้กราฟในช่วงที่ปล่อยคันเร่งแล้วค่อยยังคงมีการแกว่งของค่าอยู่มาก ซึ่งการแกว่งของค่าความเร่งเกิดจากระบบขับเคลื่อนของรถจำลองเองและอาจเกิดจากผิวสัมผัสของล้อรถกับพื้นด้วย



รูปที่ 4.7 กราฟความเร่งขณะรถวิ่งเป็นวงกลม

จากรูปที่ 4.7 จะเห็นได้ว่าลักษณะกราฟคล้ายการวิ่งเป็นเส้นตรง โดยเส้นกราฟของความเร่ง X มีค่าเป็นบวกในช่วงแรกซึ่งเป็นผลมาจากการวิ่งถอยหลังพร้อมกับการเลี้ยวซ้ายแล้ววิ่งเป็นวงกลม โดยเส้นกราฟของความเร่ง Y จะมีค่าเฉลี่ยอยู่ในช่วงบวกและมีการแกว่งของค่าน้อยกว่าการเคลื่อนที่เป็นเส้นตรงเพราะมีแรงเหวี่ยงจากการเคลื่อนที่เป็นวงกลมทำให้ค่าความเร่งที่เกิดขึ้นค่อนข้างใกล้เคียงกัน ต่างจากการเคลื่อนที่เป็นเส้นตรงที่ค่าความเร่ง Y มักจะเกิดจากการแกว่งและสั่นของตัวรถจำลองในขณะวิ่ง แต่เมื่อเทียบกับการเคลื่อนที่ด้วยมือเป็นวิถีโค้งแล้ว ค่าความเร่ง Y จะต่างกันอย่างเห็นได้ชัด เพราะการเคลื่อนที่ด้วยมือมีแรงเหวี่ยงมากกว่าการเคลื่อนที่เป็นวงกลมของรถจำลองรวมถึงการเคลื่อนที่ด้วยมือมีความแกว่งของค่าน้อยกว่าการเคลื่อนที่ด้วยรถมาก

สรุปผลการทดสอบการอ่านค่าความเร่งจากรถจำลองได้ว่า การเคลื่อนที่ของรถจำลองที่ใช้ในการทดสอบให้ค่าความเร่งที่ไม่คงที่มีมีการแกว่งของค่าความเร่งอยู่มากเนื่องจากในระหว่างที่รถวิ่งมีการขยับตัวและมีอาการสั่นอยู่ตลอดเวลาจึงทำให้ค่าความเร่งที่อ่านได้มีการแกว่งของข้อมูลสูง

4.1.3 สรุปผลการอ่านค่าความเร่ง

จากผลการทดสอบการอ่านค่าความเร่งด้วยมือและการอ่านค่าความเร่งด้วยรถจำลอง สามารถสรุปผลการทดสอบได้ว่า ค่าความเร่งที่อ่านได้จากการเคลื่อนที่ด้วยมือมีความแกว่งของค่าน้อยกว่าค่าความเร่งที่ได้จากการเคลื่อนที่ด้วยรถจำลองทั้งการเคลื่อนที่ทางตรงและวิถีโค้ง

ในการเคลื่อนที่เป็นวิถีโค้งค่าของเส้นกราฟ Y จะมีการเพิ่มขึ้นและลดลงตามทิศทางของแรงหนีศูนย์กลางที่เกิดจากการเคลื่อนที่ทำให้ค่าของเส้นกราฟ Y มีค่าเพิ่มขึ้นเมื่อเริ่มการเคลื่อนที่และลดลงจนเข้าใกล้ 0 ก่อนหยุดการเคลื่อนที่ แสดงว่าค่าความเร่งของเส้นกราฟ X จะมีมากกว่าค่าความเร่งของเส้นกราฟ Y ซึ่งไม่ได้เป็นทิศทางการเคลื่อนที่หลัก จึงมีเพียงแรงหนีศูนย์กลางจากการเคลื่อนที่เป็นวิถีโค้งเท่านั้นที่ทำให้เกิดค่าของเส้นกราฟ Y แต่สำหรับค่าของเส้นกราฟ X จะมีทั้งความเร่งที่เกิดจากการเคลื่อนที่ แรงเฉื่อย และ แรงต้านในช่วงหยุดการเคลื่อนที่ จึงทำให้ค่าความเร่งในทิศทางการเคลื่อนที่มีมากกว่าความเร่งในทิศที่ตั้งฉากกับการเคลื่อนที่

การเคลื่อนที่จากภาวะหยุดนิ่งในช่วงแรกค่าความเร่งจะมีขนาดเพิ่มขึ้นตามทิศทางการเคลื่อนที่ไปจนถึงจุดที่ความเร็วคงที่ ค่าความเร่งจะลดลงมาเข้าใกล้ศูนย์ จากนั้นค่าความเร่งจะมีทิศสวนทางกับการเคลื่อนที่เมื่อมีการลดความเร็วในการเคลื่อนที่ลง และความเร่งจะมีค่าสวนทางกับการเคลื่อนที่ตั้งแต่เริ่มชะลอความเร็วไปจนหยุดการเคลื่อนที่ และถึงแม้ว่าจะเมื่อหยุดการเคลื่อนที่แล้วแต่ก็ยังมีความเร่งที่สวนทางกับการเคลื่อนที่ก่อนหน้าอยู่อีกระยะหนึ่ง โดยขนาดและระยะเวลาของความเร่งที่หลงเหลืออยู่หลังจากหยุดการเคลื่อนที่จะขึ้นอยู่กับแรงที่ใช้ในการหยุดการเคลื่อนที่นั้น

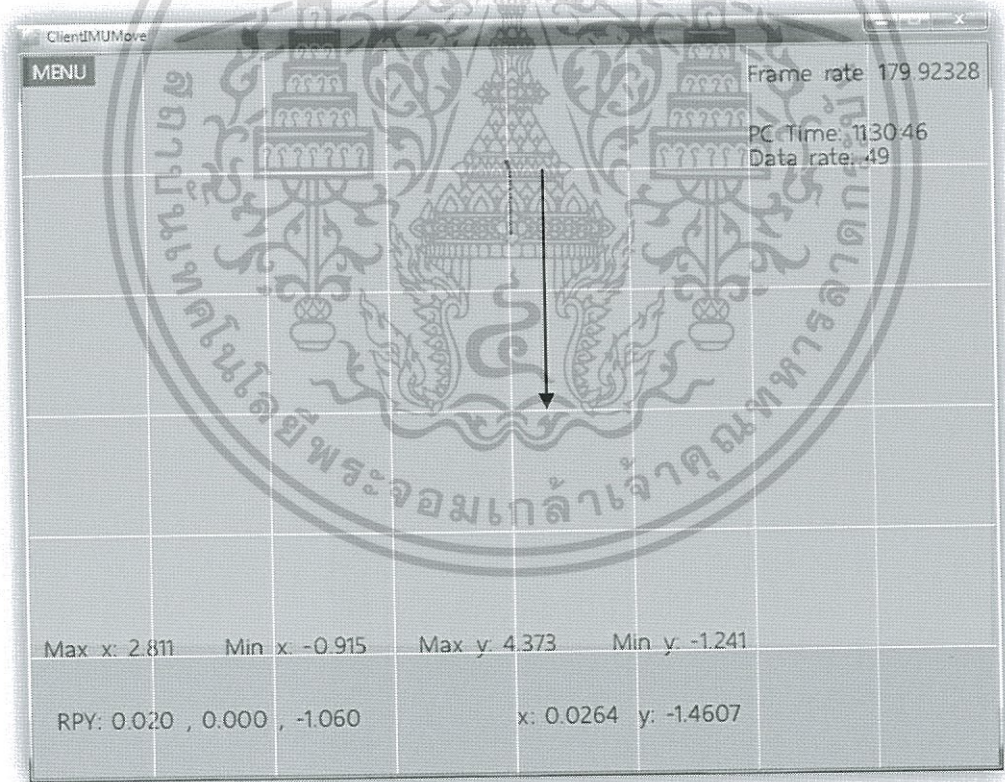
4.2 การทดสอบการระบุตำแหน่ง

เป้าหมายของการทดสอบในส่วนนี้คือการนำค่าที่ได้จากการคำนวณในระบบการติดตามตำแหน่งการเคลื่อนที่มาแสดงผลใน โปรแกรมส่วนแสดงผล

การทดสอบระบบในครั้งนี้มีขึ้นในพื้นที่ห้องปฏิบัติการวิจัย พื้นเรียบปูด้วยกระเบื้องยาง พื้นที่ในการทดสอบเป็นพื้นที่โล่งกลางห้อง ไม่มีสิ่งกีดขวางขนาดยาว 4 เมตร กว้าง 2 เมตร

วิธีการทดสอบคือเปิดโปรแกรมส่วนติดต่อ COM Port เพื่อรับค่าจากรถจำลองจากนั้นเปิดโปรแกรมส่วนแสดงผลเพื่อดูการแสดงผลการเคลื่อนที่ หลังจากนั้นทำการบังคับรถจำลองให้วิ่งไปข้างหน้าระยะทางประมาณ 2 เมตรจากจุดเริ่มต้นแล้วดูตำแหน่งการเคลื่อนที่ในโปรแกรมแสดงผล

ในส่วนแสดงผลได้มีการวาดตารางด้วยเส้นสีเทาอ่อนซึ่งใช้อ้างอิงระยะการเคลื่อนที่ของรถจำลอง โดยในการทดสอบนี้จะกำหนดให้ 1 ช่องในส่วนแสดงผลเท่ากับพื้นที่ 1 ตารางเมตรบนพื้นที่การทดลองจริง โดยแทนตำแหน่งของรถจำลองด้วยเครื่องหมาย X สามารถติดตามตำแหน่งการเคลื่อนที่ได้จากจุดสีน้ำเงินที่เรียงต่อกันเป็นเส้นทางในการเคลื่อนที่



รูปที่ 4.8 หน้าจอแสดงผลการติดตามตำแหน่งในการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.8 จะเห็นว่าเส้นทางจากจุดเริ่มต้นจนถึงจุดปัจจุบันของรถในโปรแกรมแสดงผล นั้นมีระยะทางประมาณ 70 เซนติเมตรซึ่งในการทดสอบนี้รถจำลองได้ทำการวิ่งจริงประมาณ 2 เมตรเท่ากับความยาวของลูกศรสีดำที่กำกับในภาพ ดังนั้นการเคลื่อนที่ที่คำนวณได้ในโปรแกรมจึง มีความผิดพลาดจากระยะที่ควรจะเป็นอยู่มาก ปัญหาที่เกิดขึ้นคือความเร็วที่คำนวณได้ในช่วงที่ ปลดปล่อยคันเร่งแล้วลดลงจนกลายเป็นค่าลบจึงทำให้ทำให้ระยะทางที่คำนวณได้ได้ติดลบและเมื่อ ระยะทางในช่วงเวลาที่คำนวณได้ติดลบทำให้ระยะทางโดยรวมจากจุดเริ่มต้นถูกหักออกไปจึงเหลือ ระยะทางที่แสดงผลแค่บางส่วนเท่านั้น ค่าที่คำนวณได้จึงเหมือนกับการเดินหน้าแล้วถอยหลัง

สรุปผลการทดสอบที่ได้คือ พบปัญหาหาระยะทางที่แสดงกับระยะทางที่วิ่งจริงไม่ตรงกัน โดยมีค่าความคลาดเคลื่อนสูงเนื่องจากค่าความเร่งในช่วงที่เป็นลบขณะรถจำลองชะลอตัวมีค่า มากกว่าที่ควรจะเป็น จึงทำให้ระยะทางที่คำนวณได้ติดลบในช่วงท้ายจึงทำให้ตำแหน่งที่แสดงใน ส่วนแสดงผลไม่ตรงกับระยะจริงเมื่อเทียบระหว่างจุดเริ่มต้นกับจุดสิ้นสุด

จากปัญหาดังกล่าวเมื่อดูค่าระยะทางรวมที่คำนวณได้โดยไม่สนใจเครื่องหมายจะได้ค่า ระยะทางจริงประมาณ 1.3 เมตรซึ่งยังคงมีความคลาดเคลื่อนอยู่พอสมควร ซึ่งค่าที่ได้จาก IMU ถูก นำมาคำนวณในกระบวนการที่ออกแบบไว้แล้ว ซึ่งเมื่อนำค่าที่กำหนดขึ้นเองมาทดสอบกลับได้ค่าที่ ถูกต้อง จึงมีความเป็นไปได้ว่าค่าที่ได้จาก IMU จะมีความผิดพลาดหรือมีค่าผิดไปจากที่ควรจะเป็น ในช่วงของการทดสอบครั้งนี้

4.3 การทดสอบการติดตามตำแหน่งในระยะสั้น

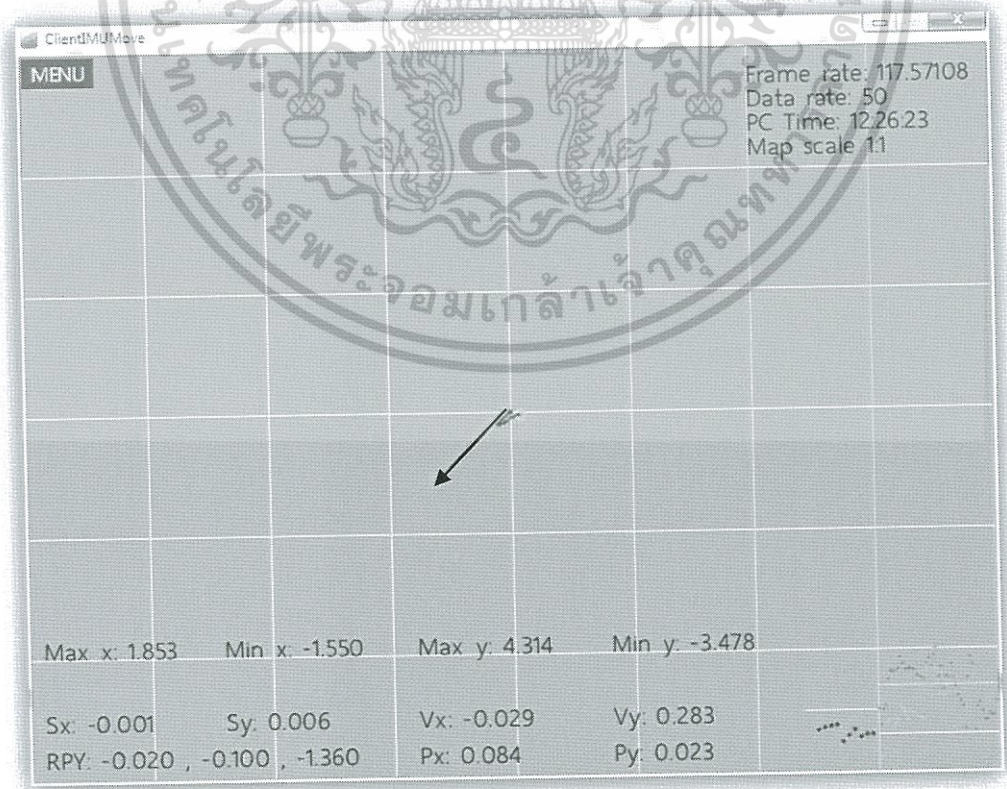
เป้าหมายในการทดสอบครั้งนี้คือ การสังเกตค่าตำแหน่งที่ได้จากการเคลื่อนที่ว่าค่าตำแหน่งจะมีการเคลื่อนที่ย้อนกลับในช่วงก่อนหยุดการเคลื่อนที่หรือไม่ โดยจะทำการเคลื่อนที่ด้วยระยะทางที่กำหนดแล้วรอให้รถหยุดเพื่อดูผลการติดตามตำแหน่งจากโปรแกรมส่วนแสดงผล

4.3.1 การติดตามตำแหน่งที่ระยะ 60 เซนติเมตร

การทดสอบในครั้งนี้จะบังคับให้รถวิ่งเป็นทางตรงระยะทางประมาณ 60 เซนติเมตรเพื่อดูตำแหน่งการเคลื่อนที่ใน โปรแกรมส่วนแสดงผลว่าตรงกับการเคลื่อนที่จริงหรือไม่ มีความคลาดเคลื่อนมากน้อยเพียงใด มีการเคลื่อนที่ย้อนกลับหรือไม่ โดยจะทำการทดสอบทั้งหมด 3 ครั้ง

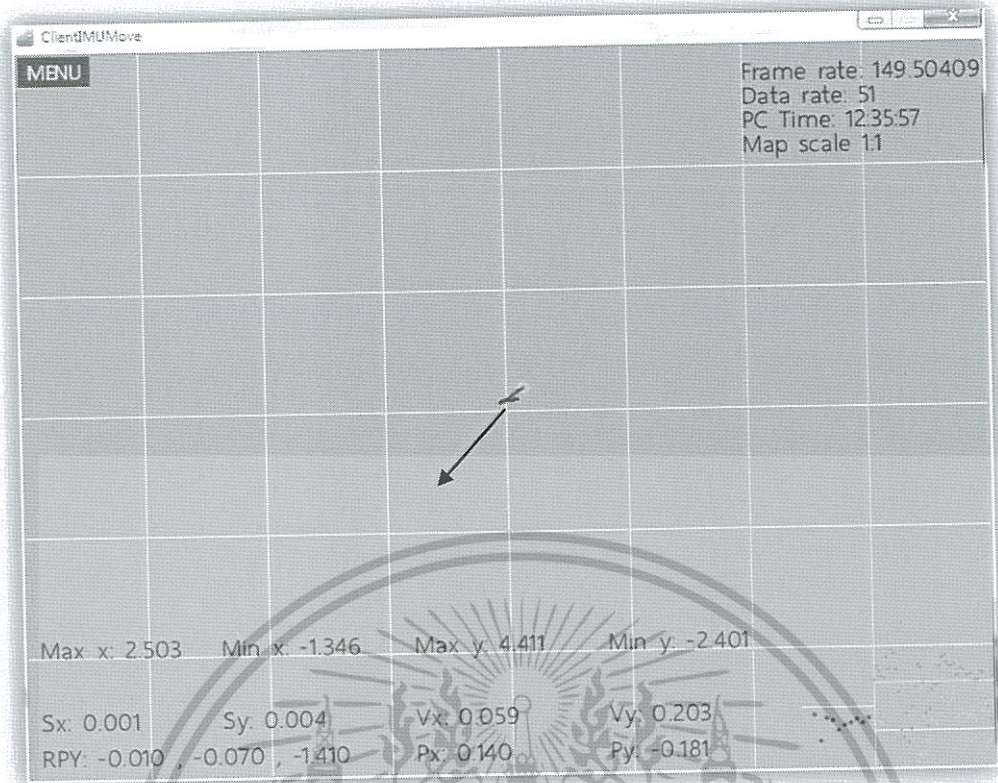
วิธีการทดสอบคือเปิดโปรแกรมส่วนติดต่อ COM Port เพื่อรับค่าจากรถจำลองจากนั้นเปิดโปรแกรมส่วนแสดงผลเพื่อดูการแสดงผลการเคลื่อนที่ หลังจากนั้นทำการบังคับรถจำลองให้วิ่งไปข้างหน้าระยะทางประมาณ 60 เซนติเมตรจากจุดเริ่มต้นแล้วดูตำแหน่งการเคลื่อนที่ในโปรแกรมส่วนแสดงผล โดยจะทำการทดสอบเหมือนกันทั้งหมด 3 ครั้ง แล้วทำการสรุปผล

พื้นที่สำหรับการทดสอบจะเป็นพื้นที่โล่งกว้างพื้นปูด้วยกระเบื้องอย่างดั่งรูปที่ 4.5 ซึ่งเป็นพื้นที่เดียวกับการทดสอบที่ 4.1.2

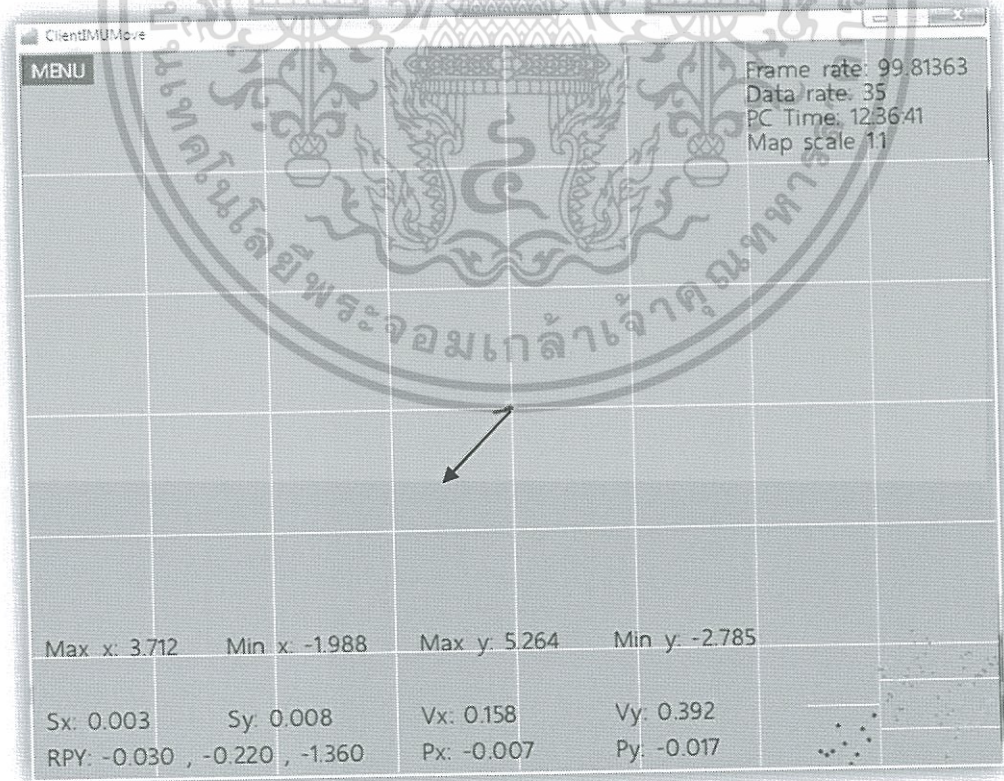


รูปที่ 4.9 การติดตามตำแหน่งระยะ 60 เซนติเมตรครั้งที่หนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 การติดตามตำแหน่งระยะ 60 เซนติเมตรครั้งที่สอง



รูปที่ 4.11 การติดตามตำแหน่งระยะ 60 เซนติเมตรครั้งที่สาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับส่วนแสดงในการทดสอบนี้จะกำหนดให้ 1 ช่องในส่วนแสดงผลเท่ากับพื้นที่ 1 ตารางเมตรบนพื้นที่การทดสอบ โดยแทนตำแหน่งปัจจุบันของรถจำลองด้วยจุดใหญ่สีเขียว และแทนตำแหน่งในอดีตด้วยจุดเล็กสีน้ำเงิน ซึ่งใช้เพื่อการติดตามตำแหน่งการเคลื่อนที่

จากการทดสอบทั้งหมดสามครั้งสำหรับการวิ่งในระยะสั้นประมาณ 60 เซนติเมตร ในรูปที่ 4.9 รูปที่ 4.10 และ รูปที่ 4.11 ระยะการเคลื่อนที่จริงของรถจำลองจะเท่ากับความยาวของลูกศรสีดำที่กำกับในภาพ แต่ผลการทดสอบที่ได้เป็นไปในทิศทางเดียวกันทั้งหมด คือ ระยะทางที่เคลื่อนที่ได้ มีความผิดพลาด และ มีการเคลื่อนที่ย้อนกลับเกิดขึ้น โดยดูได้จากจุดเล็กสีน้ำเงินบริเวณกลางภาพที่เรียงต่อกันเป็นเส้นทางในการเคลื่อนที่

ปัญหาการเคลื่อนที่ย้อนกลับที่เกิดขึ้นเป็นเพราะในการเคลื่อนที่ระยะทางสั้น ๆ นั้นค่าความเร่งในตอนเริ่มเคลื่อนที่กับค่าความเร่งในตอนหยุดจะหักล้างกันทำให้ระยะทางจริงกับระยะทางที่แสดงในโปรแกรมเกิดความคลาดเคลื่อนสูง โดยค่าความเร่งขณะหยุดจะเป็นค่าที่สวนทางกับความเร่งในขณะเคลื่อนที่ ถ้าหากเคลื่อนที่ด้วยความเร่งที่ต่ำแล้วหยุดด้วยแรงที่สูงกว่าหรือหยุดอย่างกะทันหันจะทำให้แรงที่ IMU อ่านได้มีค่ามากจนทำให้การคำนวณ ได้ผลเป็นการวิ่งถอยหลังอย่างที่เห็นในการทดสอบที่ผ่านมา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การทดสอบการติดตามตำแหน่งในระยะทางยาว

จากการทดสอบการติดตามตำแหน่งในระยะสั้นทำให้ทราบปัญหาเกี่ยวกับการคำนวณค่าได้เป็นการเคลื่อนที่ถอยหลังจึงทำการทดสอบการเคลื่อนที่ในระยะทางยาวเพื่อสังเกตตำแหน่งการเคลื่อนที่ในโปรแกรมแสดงผล โดยจะทำการเคลื่อนที่ด้วยระยะทางที่กำหนดแล้วรอให้รถหยุดเพื่อผลการติดตามตำแหน่งจากโปรแกรมส่วนแสดงผล

เป้าหมายของการทดสอบคือการดูตำแหน่งการเคลื่อนที่ ที่แสดงในโปรแกรมว่าใกล้เคียงกับการเคลื่อนที่จริงของรถจำลองหรือไม่ และระยะทางที่ได้ผิดพลาดค่าน้อยเพียงใด

วิธีการทดสอบคือเปิดโปรแกรมส่วนติดต่อ COM Port เพื่อรับค่าจากรถจำลองจากนั้นเปิดโปรแกรมส่วนแสดงผลเพื่อดูการแสดงผลการเคลื่อนที่ หลังจากนั้นทำการบังคับรถจำลองให้วิ่งไปตามระยะทางที่กำหนดแล้วดูตำแหน่งการเคลื่อนที่ในโปรแกรมส่วนแสดงผล

พื้นที่สำหรับการทดสอบเป็นพื้นที่โล่งกว้างพื้นปูด้วยกระเบื้องยาง ดังรูปที่ 4.12



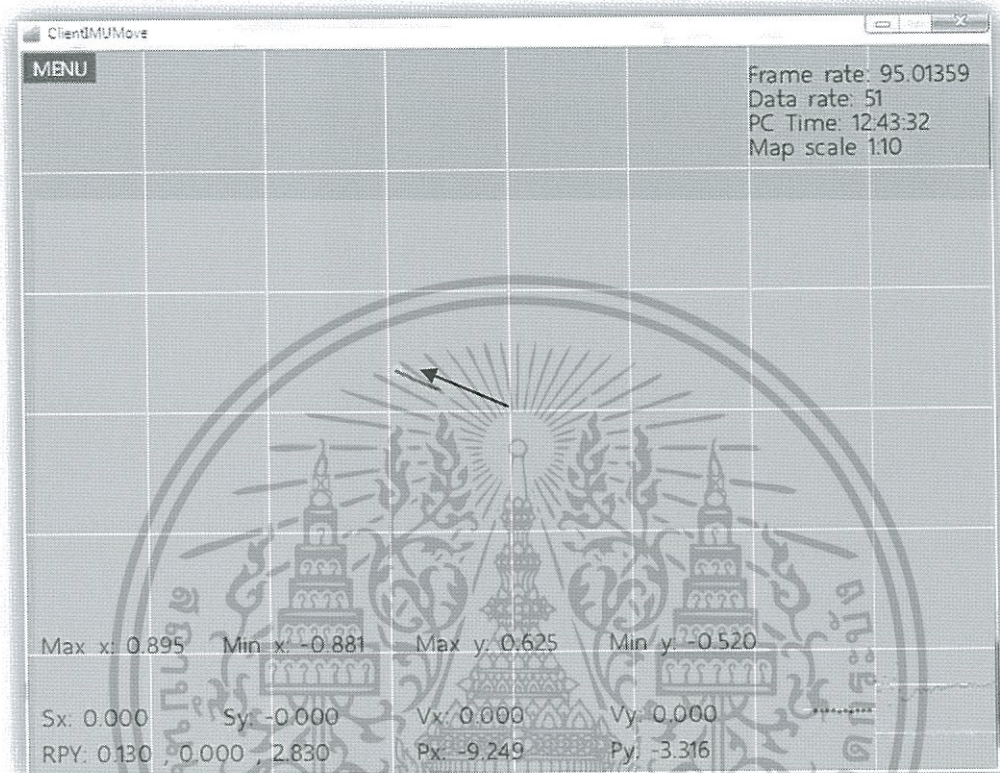
รูปที่ 4.12 พื้นที่ทดสอบการเคลื่อนที่

สำหรับการแสดงผลในการทดสอบนี้จะกำหนดให้ 1 ช่องในโปรแกรมส่วนแสดงผลเท่ากับพื้นที่ 10×10 เมตรบนพื้นที่การทดสอบโดยแทนตำแหน่งปัจจุบันของรถจำลองด้วยจุดใหญ่สีเขียว และแทนตำแหน่งในอดีตด้วยจุดเล็กสีน้ำเงิน ซึ่งใช้เพื่อการติดตามตำแหน่งการเคลื่อนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.1 การติดตามตำแหน่งที่ระยะ 8 เมตร

ในการทดสอบนี้จะบังคับรถจำลองให้เคลื่อนที่ตรงไปข้างหน้าด้วยระยะทางประมาณ 8 เมตร จากจุดเริ่มต้น



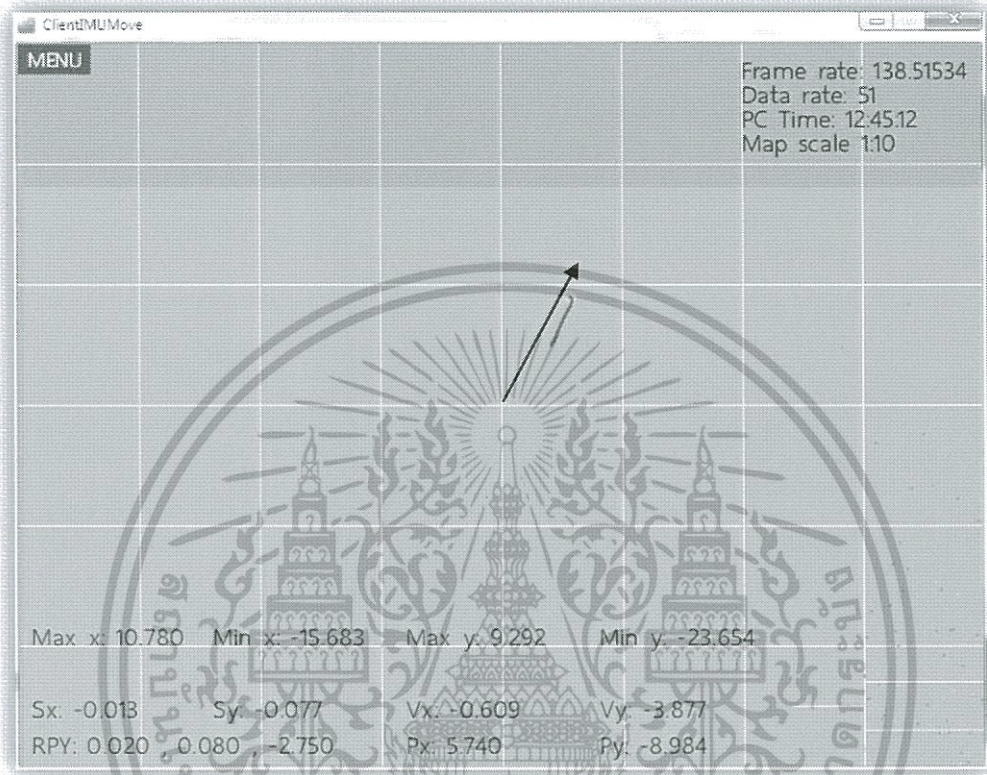
รูปที่ 4.13 ผลการทดสอบติดตามตำแหน่งการเคลื่อนที่ระยะ 8 เมตร

จากรูปที่ 4.13 ลูกศรสีดำในภาพคือเส้นกำกับทิศทางและระยะการเคลื่อนที่จริงของรถจำลอง ซึ่งผลการทดสอบในครั้งนี้แสดงให้เห็นถึงเส้นทางการเคลื่อนที่ของรถจำลองที่มีทิศทางเคลื่อนที่ตรงกับการเคลื่อนที่จริง แต่มีระยะการเคลื่อนที่ประมาณ 9.2 เมตร ซึ่งมากกว่าการเคลื่อนที่จริงอยู่ประมาณ 1 เมตร

สำหรับการทดสอบครั้งนี้ให้ผลเป็นที่น่าพอใจ เนื่องจากทิศทางเคลื่อนที่ตรงกับความ เป็นจริงและระยะการเคลื่อนที่ใกล้เคียงกับความเป็นจริงมากกว่าการทดสอบการติดตามตำแหน่งใน ระยะสั้น

4.4.2 การติดตามตำแหน่งที่ระยะ 13 เมตร

ในการทดสอบนี้จะบังคับรถจำลองให้เคลื่อนที่ตรงไปข้างหน้าด้วยระยะทางประมาณ 13 เมตร จากจุดเริ่มต้น



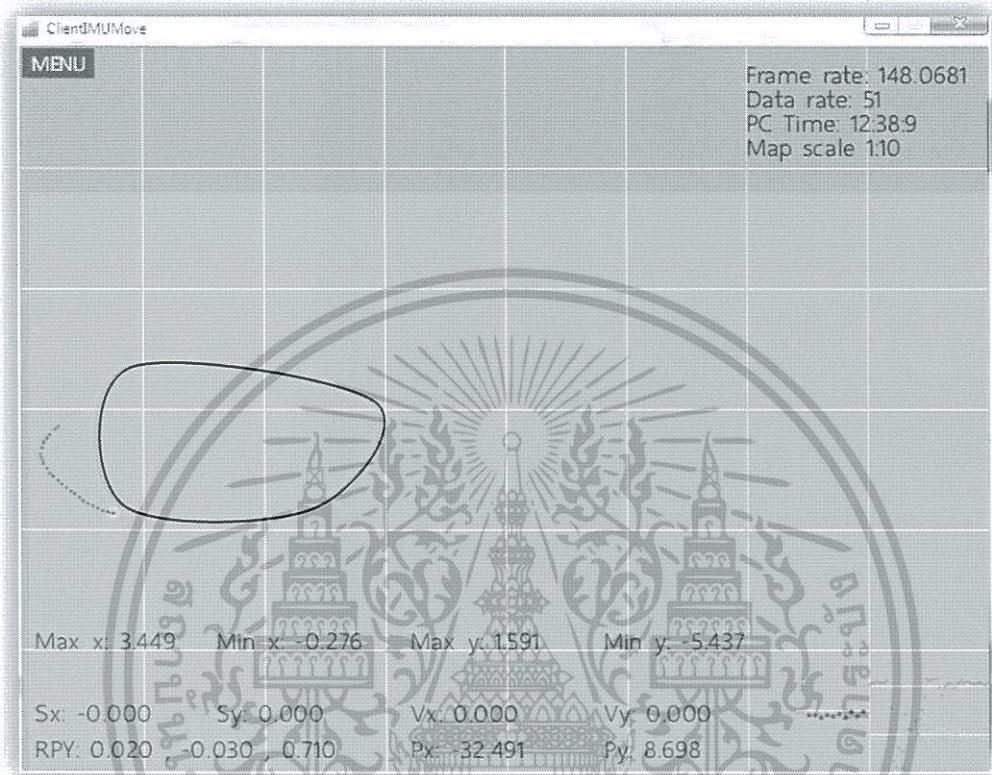
รูปที่ 4.14 ผลการทดสอบติดตามตำแหน่งการเคลื่อนที่ระยะ 13 เมตร

จากรูปที่ 4.14 ลูกศรสีดำในภาพคือเส้นกำกับทิศทางและระยะการเคลื่อนที่จริงของรถจำลอง แต่จากผลการทดสอบจะเห็นว่าเคลื่อนที่เป็นไปในทิศทางเดียวกับลูกศรจนถึงระยะประมาณ 8 เมตร ได้เกิดการเลี้ยวเล็กน้อยเนื่องจากค่าทิศทางที่ใช้อ้างอิงจาก AHRS เกิดความผิดพลาดจึงทำให้ค่าทิศทางที่ใช้ในการอ้างอิงการเคลื่อนที่ของรถจำลองในโปรแกรมแสดงผลมีความผิดพลาดเกิดขึ้นด้วยเช่นกัน

การทดสอบในครั้งนี้ได้ผลการเคลื่อนที่ส่วนใหญ่เป็นเส้นตรงแต่ในช่วงท้ายมีการเบี่ยงเล็กน้อย ซึ่งระยะการเคลื่อนที่ทั้งหมดที่คำนวณได้ก่อนการปรับทิศคือประมาณ 12 เมตร ซึ่งน้อยกว่าระยะทางจริงประมาณ 1 เมตร

4.4.3 การติดตามตำแหน่งการเคลื่อนที่เป็นวงกลม

ในการทดสอบนี้จะทำการวิ่งเป็นวงกลมซึ่งมีระยะการเคลื่อนที่รวมมากกว่า 30 เมตร โดยการเคลื่อนที่ในช่วงแรกจะเป็นเส้นตรงและจะเริ่มเลี้ยวในช่วงระยะประมาณ 20 เมตรจากจุดเริ่มต้น



รูปที่ 4.15 ผลการทดสอบติดตามตำแหน่งการเคลื่อนที่เป็นวงกลม

วงรีสีดำในรูปที่ 4.15 แสดงถึงเส้นทางอ้างอิงการสำหรับเคลื่อนที่ของรถจำลอง ส่วนตำแหน่งปัจจุบันในรูปที่ 4.15 นั้นเป็นจะช่วงหลังจากการเลี้ยวครั้งแรกโดยระยะทางที่ได้จากการคำนวณในโปรแกรมส่วนแสดงผลคือประมาณ 32 เมตร โดยตำแหน่งการเคลื่อนที่ใน โปรแกรมแสดงผลมีความคลาดเคลื่อนจากตำแหน่งการเคลื่อนที่จริงอยู่มาก

สรุปผลการทดสอบสำหรับการวิ่งระยะยาวได้ว่า การวิ่งในระยะยาวยังคงมีความผิดพลาดในช่วงหยุดการเคลื่อนที่แต่ละระยะที่ย้อนกลับจะน้อยกว่าระยะทางวิ่งโดยรวมมากจึงทำให้ระยะทางโดยรวมมีความผิดพลาดน้อยกว่าการเคลื่อนที่ในระยะสั้นเมื่อเทียบกันเป็นอัตราส่วนและความผิดพลาดของทิศทางจาก AHRS ยังคงมีผลกระทบอยู่มากในการแสดงผลและการคำนวณระยะทางในโปรแกรมส่วนแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 ทดสอบการติดตามตำแหน่งของรถแข่ง FORMULA STUDENT

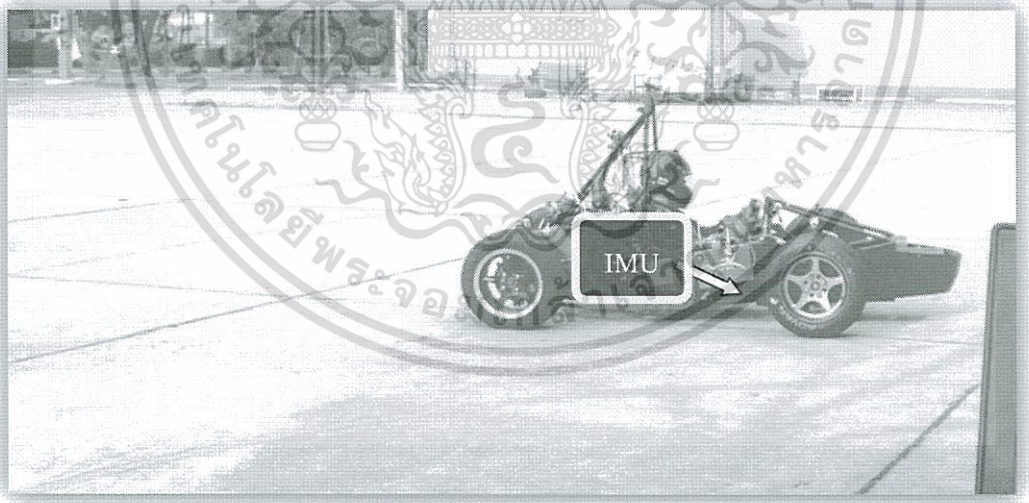
เป้าหมายในการทดสอบครั้งนี้คือการทดสอบการติดตามตำแหน่งการเคลื่อนที่อย่างต่อเนื่องด้วยระยะทางและความเร็วที่มากกว่าการใช้รถจำลองเพื่อสังเกตตำแหน่งการเคลื่อนที่ที่ได้ว่ามีความเหมือนหรือแตกต่างจากการทดสอบโดยใช้รถจำลองหรือไม่ อย่างไรก็ตาม โดยการทดสอบครั้งนี้ได้นำเอาชุดทดสอบไปติดตั้งบนรถแข่ง Formula Student ซึ่งวิ่งด้วยความเร็วสูง และมีแรงสั่นที่มากกว่ารถจำลองที่ใช้ทดสอบในโครงการนี้เป็นอย่างมาก

การจัดเตรียมการทดสอบเริ่มจากการติดตั้งชุด IMU ไว้ในตำแหน่งใต้เบาะนั่งบนรถแข่งในบริเวณที่ลูกศรสีขาวชี้ในรูปที่ 4.16 จากนั้นทำการเปิดโปรแกรมส่วนติดต่อ COM Port เพื่อรับข้อมูลจากชุดอุปกรณ์ที่ติดตั้งบนรถ และ เปิดโปรแกรมส่วนแสดงผลเพื่อดูผลการติดตามตำแหน่งและการเคลื่อนที่ของรถแข่งในขณะที่เคลื่อนที่

สภาพสนามที่ใช้ในการทดสอบเป็นพื้นปูนกลางแจ้งมีขนาดพื้นที่ประมาณ 30×40 เมตร

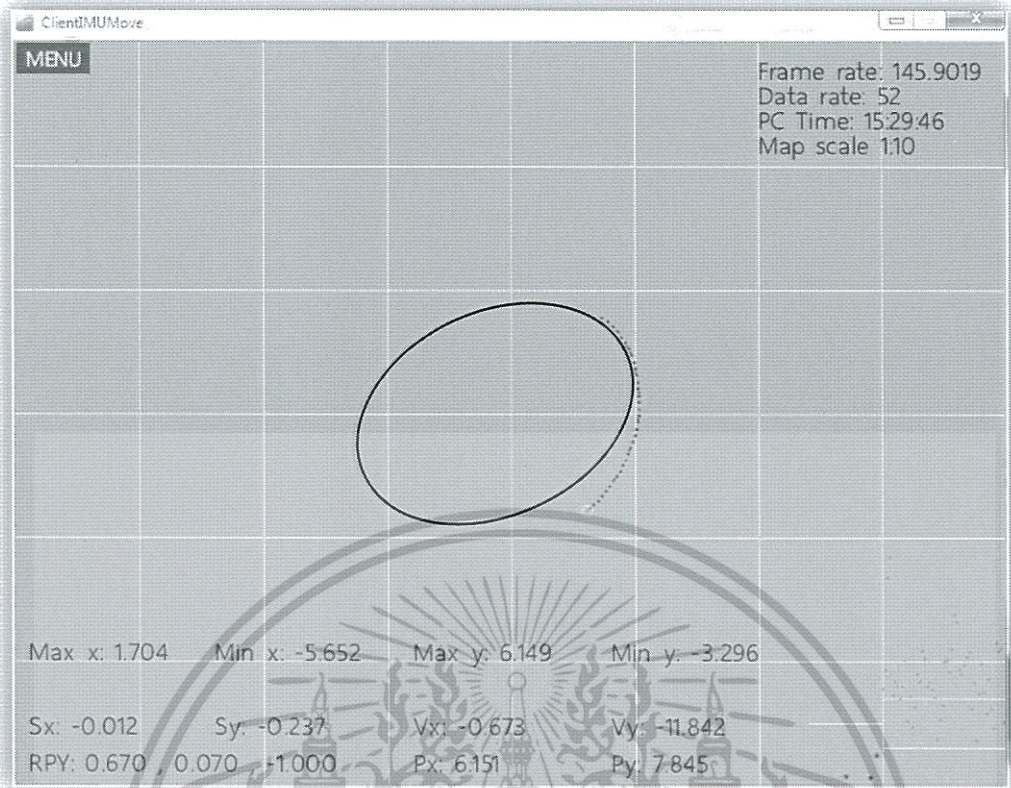
วิธีการทดสอบครั้งนี้จะทำการเคลื่อนที่โดยให้นักขับทำการขับวนเป็นวงกลมโดยการวนซ้าย 2 รอบ แล้ววนขวา 2 รอบ หลังจากนั้นจะเป็นการเคลื่อนที่ตามแต่ที่นักขับจะเลือกกว่าต้องการขับวนในทิศทางไหน

กำหนดความเร็วในการเคลื่อนที่ไว้ไม่เกิน 40 Km/h โดยประมาณ



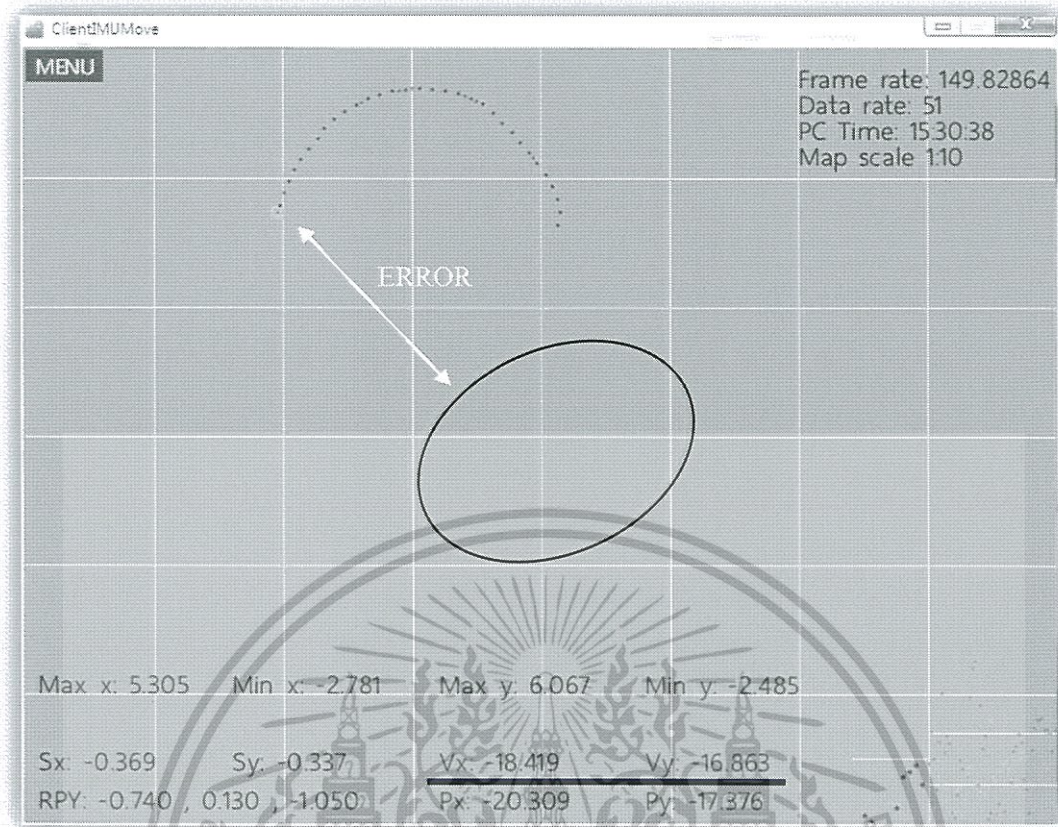
รูปที่ 4.16 รถแข่ง Formula Student ที่ใช้ในการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.17 ผลการทดสอบตำแหน่งการเคลื่อนที่ในช่วงแรก

จากรูปที่ 4.17 วงรีสีดำเป็นเส้นอ้างอิงการเคลื่อนที่ ส่วนจุดสีน้ำเงินบริเวณกลางภาพแสดงถึงตำแหน่งการเคลื่อนที่ของรถที่เป็นวิถีโค้งซึ่งเป็นไปในทิศทางเดียวกับการเคลื่อนที่ของรถจริง ในขณะที่ทำการทดสอบ โดยในช่วงแรกค่าความเร็วที่คำนวณได้มีความผิดพลาดเล็กน้อยเมื่อเทียบกับความเร็วในการเคลื่อนที่จริง แต่เมื่อเวลาผ่านไปค่าความเร็วในการเคลื่อนที่จะพุ่งสูงขึ้นจากความผิดพลาดสะสมในการคำนวณทำให้วงรีขาวที่แสดงใน โปรแกรมแสดงผลมีรัศมีการเลี้ยวกว้างกว่าความเป็นจริง และค่าความเร็วพุ่งสูงขึ้นจนเกินความเป็นจริงไปมาก เมื่อเคลื่อนที่เป็นเส้นตรงจากมุมหนึ่งสูงอีกมุมหนึ่งของสนามซึ่งมีระยะการวิ่งประมาณ 30 เมตรในการเคลื่อนที่จริง แต่ในโปรแกรมแสดงผลการเคลื่อนที่ดังกล่าวคำนวณได้มากกว่า 60 เมตรซึ่งเป็นผลมาจากความผิดพลาดสะสมในการคำนวณความเร็ว



รูปที่ 4.18 ผลการทดสอบตำแหน่งการเคลื่อนที่ในช่วงที่มีความเร็วสะสม

จากรูปที่ 4.18 วิธีตีค่าคือเส้นอ้างอิงการเคลื่อนที่ของรถแข่งใน เมื่อดูบริเวณด้านล่างของโปรแกรมส่วนแสดงผลที่ขีดเส้นได้เอาไว้จะเห็นว่าค่าความเร็วสะสมในระบบเพิ่มขึ้นเกิน 16 m/s ทำให้รัศมีวงเลี้ยวของรถที่แสดงใน โปรแกรมแสดงผลเพิ่มขึ้นอย่างมาก โดยเมื่อเทียบเป็นระยะจริงแล้วจะมีระยะทางในการเลี้ยวถึง 20 เมตร ซึ่งในความเป็นจริงใช้แค่ประมาณ 4 เมตรเท่านั้น

สรุปผลการทดสอบการติดตามตำแหน่งด้วยรถแข่งในครั้งนี้พบว่าค่าความผิดพลาดสะสมของความเร็วเมื่อรถวิ่งติดต่อกันนาน ๆ จะมีค่าเพิ่มขึ้นสูงจนเกิดความผิดพลาดของระยะทางรวมอย่างมาก และการใช้ข้อมูลจาก IMU เพียงอย่างเดียวจึงไม่เพียงพอต่อการติดตามตำแหน่งที่ต้องการความแม่นยำของระยะทางเนื่องจากปัญหาของค่าความผิดพลาดสะสมที่เพิ่มขึ้นตามระยะเวลาการทำงาน แต่การใช้ทิศทางจาก AHRS มาปรับทิศทางในการแสดงผลนั้นสามารถใช้งานได้ดีถ้าหากไม่ถูกรบกวนทางแม่เหล็กมากจนทำให้ทิศทางที่ได้เกิดความผิดพลาด เพราะในการทดสอบครั้งนี้เห็นว่าทิศทางในการเคลื่อนที่ส่วนใหญ่ตรงกับการเคลื่อนที่จริงของรถแข่งที่ใช้ทดสอบ โดยมีเพียงแค่ระยะทางเท่านั้นที่มีความผิดพลาดสูงจนไม่สามารถใช้อ้างอิงตำแหน่งที่ถูกต้องได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลโครงการ

โครงการนี้เป็นกรออกแบบและพัฒนาระบบติดตามตำแหน่งการเคลื่อนที่ของรถจำลองด้วยเทคนิคนำร่องเฉื่อย (INS) โดยใช้ข้อมูลความเร่งและการหมุนจากหน่วยวัดแรงเฉื่อย (IMU) ร่วมกับการปรับค่าทิศทางและการวางตัวด้วย AHRS รวมทั้งการปรับลดความผิดพลาดสะสมขณะหยุดนิ่งด้วยการรีเซ็ตค่าด้วยตัวเอง (Self-resetting) โดยกระบวนการในการคำนวณทั้งหมดในระบบที่พัฒนาขึ้นนี้ใช้ค่าตั้งต้นจากอุปกรณ์ 9DOF Razor IMU เพียงตัวเดียว โดยการพัฒนาในระบบในส่วนต่าง ๆ ได้สำเร็จตามที่ตั้งเป้าไว้ และระบบสามารถใช้งานได้ตามวัตถุประสงค์ โดยสามารถติดตามตำแหน่งการเคลื่อนที่ของรถจำลองที่ติดตั้งชุดอุปกรณ์ที่พัฒนาขึ้นได้

ในโครงการนี้ได้มีการทดสอบระบบหลังจากการพัฒนาเพื่อตรวจสอบความถูกต้องและผลลัพธ์ที่ได้จากการทำงานของระบบ จากผลการทดสอบทำให้ทราบถึงข้อจำกัดต่าง ๆ ของระบบที่ได้พัฒนาขึ้น ซึ่งทำให้ผลการทดสอบที่ได้ไม่เป็นไปตามความคาดหวัง โดยปัญหาที่พบคือการอ้างอิงทิศทางด้วย AHRS สามารถถูกรบกวนจากสนามแม่เหล็กได้จึงทำให้ทิศทางการวิ่งของรถจำลองที่ใช้การทดสอบผิดเพี้ยนไป และปัญหาสำคัญคือ ระยะทางที่วิ่งจริงกับระยะทางที่คำนวณได้ยังคงมีความคลาดเคลื่อนสูง เนื่องจากการใช้ค่าจาก IMU เพียงอย่างเดียว และเป็น IMU ราคาถูกจึงทำให้ค่าที่ได้ไม่ละเอียดพอ รวมถึงปัญหาจากระบบขับเคลื่อนของรถจำลองที่ใช้ในการทดสอบนั้นมีแรงสั่นในขณะเคลื่อนที่สูงทำให้ค่าความเร่งที่ได้จาก IMU ผิดเพี้ยนไปและทำให้เกิดการแกว่งของค่าข้อมูลด้วยเช่นกัน

ระบบที่พัฒนาขึ้นมานั้นสามารถใช้งานได้ตามวัตถุประสงค์ โดยใช้ติดตามตำแหน่งการเคลื่อนที่ของรถจำลองได้จริง ซึ่งสามารถแสดงผลการเคลื่อนที่ในโปรแกรมที่พัฒนาขึ้นมาได้ตามค่าที่คำนวณมาจากค่าของ IMU ซึ่งถ้าไม่มีการถูกรบกวนทางสนามแม่เหล็ก ทิศทางการเคลื่อนที่ในโปรแกรมแสดงผลจะตรงกับการเคลื่อนที่จริงแม้ว่าระยะทางจะคลาดเคลื่อนไปบ้างก็ตาม

จากการพัฒนาระบบดังกล่าวทำให้ทราบว่าไม่สามารถพัฒนาระบบติดตามตำแหน่งที่แม่นยำได้โดยการใช้ข้อมูลจาก IMU เพียงอย่างเดียว หากต้องการความแม่นยำเพิ่มขึ้นจำเป็นต้องใช้ข้อมูลจากแหล่งอื่นมาประกอบด้วย เช่น ใช้ความเร็วในการเคลื่อนที่ของพาหะที่ไม่ได้มาจากการคำนวณความเร่ง หรือใช้อ้างอิงทิศทางด้วยข้อมูลจากแหล่งอื่นที่ไม่ถูกรบกวนด้วยสนามแม่เหล็ก เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1 ปัญหาและอุปสรรค

ปัญหาและอุปสรรคในการพัฒนาโครงการนี้เกิดขึ้นด้วยปัจจัยต่าง ๆ เช่น ปัญหาจากอุปกรณ์ที่ใช้ ปัญหาจากกระบวนการคำนวณ และ ปัญหาอันเกิดจากสภาพแวดล้อมที่ไม่สามารถควบคุมได้ เป็นต้น ซึ่งปัญหาต่าง ๆ ที่พบสามารถสรุปเป็นกลุ่มของปัญหาได้ ดังนี้

ปัญหาค่าสัญญาณรบกวนภายใน IMU และความละเอียดของค่าที่ใช้การได้ ซึ่งส่งผลโดยตรงกับค่าความแม่นยำในการคำนวณของระบบ โดยปัญหาดังกล่าวอาจเกิดได้จากแหล่งจ่ายพลังงานที่ไม่เสถียร แต่ปัจจัยสำคัญที่สุดคือคุณภาพของ IMU ที่ใช้

ปัญหาความคลาดเคลื่อนของค่าที่อ่านได้จาก IMU ลักษณะของปัญหาคือ ค่าที่อ่านได้จากเซนเซอร์ต่าง ๆ มีความผิดพลาดไปจากค่าที่ควรจะเป็น เช่น ค่าแรงดึงดูดของโลกที่อ่านได้ในแต่ละแกนมีค่าไม่เท่ากันเมื่อวางให้แกนดังกล่าวชี้ลงไปยังพื้นโลก เป็นต้น โดยปัญหาดังกล่าวสามารถแก้ไขได้ด้วยการปรับค่ามาตรฐาน (Calibrate) แต่การที่จะปรับค่าให้มีความคลาดเคลื่อนน้อยที่สุดจำเป็นต้องใช้อุปกรณ์ และ เครื่องมือวัดที่มีความละเอียดสูงกว่า IMU ในการปรับค่า

ปัญหาค่าความผิดพลาดสะสมในการคำนวณ เนื่องจากกระบวนการที่ใช้ในการหาตำแหน่งคือการหาปริพันธ์อันดับสองของความเร่ง จึงทำให้เกิดความคลาดเคลื่อนจากการคำนวณถึงสองครั้งในการหาตำแหน่งแต่ละครั้ง ถึงแม้ว่าจะมีการปรับค่าความผิดพลาดในกระบวนการต่างๆแล้ว ก็ยังคงมีความผิดพลาดอยู่ ซึ่งเมื่อเวลาผ่านไปค่าความผิดพลาดที่สะสมไว้จะยิ่งส่งผลมากขึ้น

ปัญหาความคลาดเคลื่อนของทิศทางการถูกรบกวนโดยสนามแม่เหล็กในสภาพแวดล้อม ลักษณะของปัญหาคือ ค่าทิศทางที่อ่านได้จาก IMU จะมีความผิดพลาดเมื่อเคลื่อนที่ผ่านบริเวณที่มีสนามแม่เหล็กรุนแรงกว่าสนามแม่เหล็กโลก เพราะ IMU ที่ใช้ในการพัฒนาระบบนี้จะอ่านค่าสนามแม่เหล็กโลกเพื่ออ้างอิงทิศทางในลักษณะเดียวกับเข็มทิศ จึงสามารถถูกรบกวนจากสนามแม่เหล็กต่างๆได้ง่ายเมื่ออยู่ใกล้

ปัญหาความผิดพลาดจากการใช้ IMU เพียงอย่างเดียว ปัญหานี้เกิดจากการใช้ค่าจาก IMU เพียงอย่างเดียวในการคำนวณตำแหน่งรวมถึงการปรับค่าความผิดพลาดสะสมด้วย ซึ่ง IMU ที่ใช้ยังถือว่ามีความละเอียดต่ำ และถูกรบกวนได้ง่าย รวมถึงค่าที่อ่านได้จาก IMU เป็นค่าแรงเฉื่อยจากการเคลื่อนที่ จึงทำให้การตรวจสอบการเคลื่อนที่มีความผิดพลาดจากแรงที่เกิดขึ้นในระหว่างการเคลื่อนที่ โดยปัญหาดังกล่าวส่งผลต่อค่าความผิดพลาดสะสมในระหว่างเคลื่อนที่เป็นอย่างมาก

5.2 ข้อเสนอแนะ

สำหรับผู้สนใจในการพัฒนาระบบติดตามตำแหน่งด้วยการใช้ IMU หรือ การพัฒนาระบบอื่น ซึ่งใช้ค่าจาก IMU ในการทำงานของระบบนั้น ทางผู้พัฒนาได้มีข้อเสนอแนะเกี่ยวกับการใช้งาน IMU และแนวทางในการแก้ปัญหาที่อาจจะเกิดขึ้นเมื่อใช้งาน IMU โดยสรุปได้ ดังนี้

การใช้ค่าจาก IMU เพียงอย่างเดียวในการติดตามตำแหน่งการเคลื่อนที่ด้วยระบบ INS ต้องมีกระบวนการปรับค่าความผิดพลาด ซึ่งใช้ค่าจากแหล่งอื่นนอกเหนือจาก IMU เพื่อเพิ่มความแม่นยำในการติดตามตำแหน่ง เนื่องจากการใช้ค่าจาก IMU เพียงอย่างเดียวในการพัฒนาโครงการนี้ ทำให้พบว่า หากค่าที่ IMU อ่านได้มีความผิดพลาดใด ๆ เกิดขึ้นแล้ว ค่าที่คำนวณได้ในระบบทั้งหมดจะเกิดความผิดพลาดไปด้วย ส่งผลให้ระบบทั้งหมดเกิดความผิดพลาดในการทำงาน

ก่อนการใช้งาน IMU ชุดใหม่ที่ไม่เคยใช้งานมาก่อน หรือ ไม่มีการรับประกันความถูกต้องของข้อมูลจากทางผู้ผลิต ควรทำการปรับค่ามาตรฐาน (Calibrate) ก่อนเริ่มใช้งานทุกครั้ง เนื่องจาก IMU ตัวนั้นอาจจะมีค่าความคลาดเคลื่อนอยู่ หากนำมาใช้งานทันทีอาจจะได้ค่าที่มีความคลาดเคลื่อนไปจากที่ควรจะเป็น

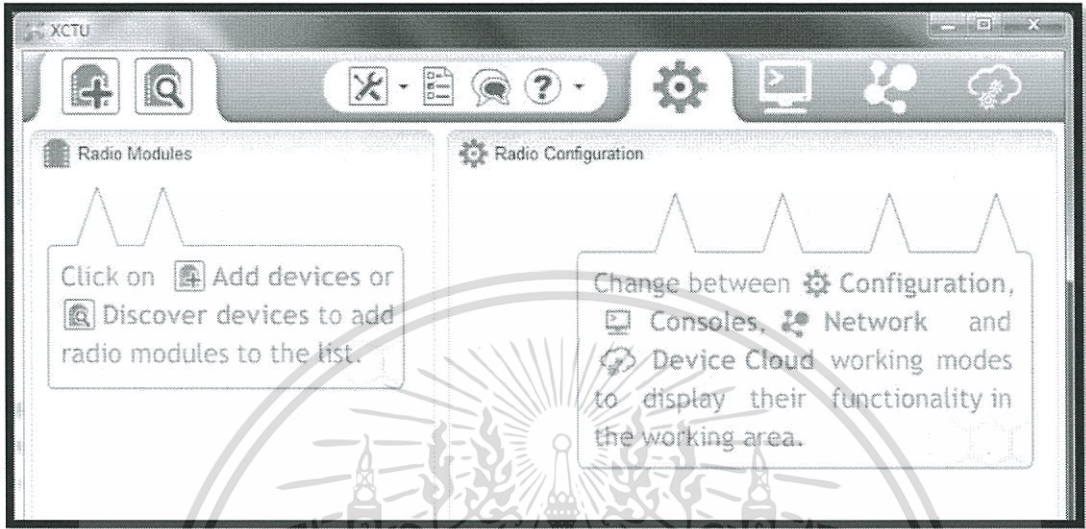
การใช้งาน IMU บนพาหนะที่มีแรงรบกวนสูงในขณะที่ทำงาน เช่น การสั่น หรือการถูกระแทก จะทำให้ค่าที่ IMU อ่านได้เกิดความผิดพลาด หากใช้ IMU ที่ไม่มีระบบกันสะเทือน หรือระบบกันกระแทกในตัวแล้ว ควรหาวิธีลดการสั่น หรือ แรงกระแทกที่อาจเกิดขึ้นได้กับ IMU ให้มากที่สุด โดยไม่กระทบกับการทำงานโดยปกติ เพื่อเพิ่มประสิทธิภาพในการทำงานของ IMU

บรรณานุกรม

- [1] MS Grewal, LR Weill, AP Andrews. **Global Positioning Systems, Inertial Navigation, and Integration**. John Wiley & Sons Inc. 2007
- [2] B. Ladislav, C. Pavel, D. Martin. "Navigation of robotic platform with using inertial measurement unit and Direct Cosine Matrix" **56th International Symposium ELMAR-2014**, 1-4, IEEE. 2014
- [3] O. Wongwirat, C. Chaiyarat. "A position tracking experiment of mobile robot with Inertial Measurement Unit (IMU)" **International Conference on Control Automation and Systems (ICCAS) 2010**, 304-308, IEEE. 2010
- [4] E. Akeila, Z. Salcic, A. Swain. "Reducing Low-Cost INS Error Accumulation in Distance Estimation Using Self-Resetting" **IEEE Transactions on Instrumentation and Measurement**, vol. 63, no. 1 (2014), 177-184
- [5] E. Mark Pedley. **Tilt Sensing Using a Three-Axis Accelerometer** Freescale Semiconductor Inc., document number: AN3461, Rev. 6, 03/2013
- [6] SparkFun Electronics, "**9 Degrees of Freedom - Razor IMU SEN-10736**", [Online] Available : <https://www.sparkfun.com/products/10736>
- [7] ptrbrtz, "**AHRS Firmware for the SparkFun 9DOF Razor IMU and SparkFun 9DOF Sensor Stick**", [Online] Available : <https://github.com/ptrbrtz/razor-9dof-ahrs>

ภาคผนวก ก.

โปรแกรมสำหรับการปรับค่า XBEE PRO

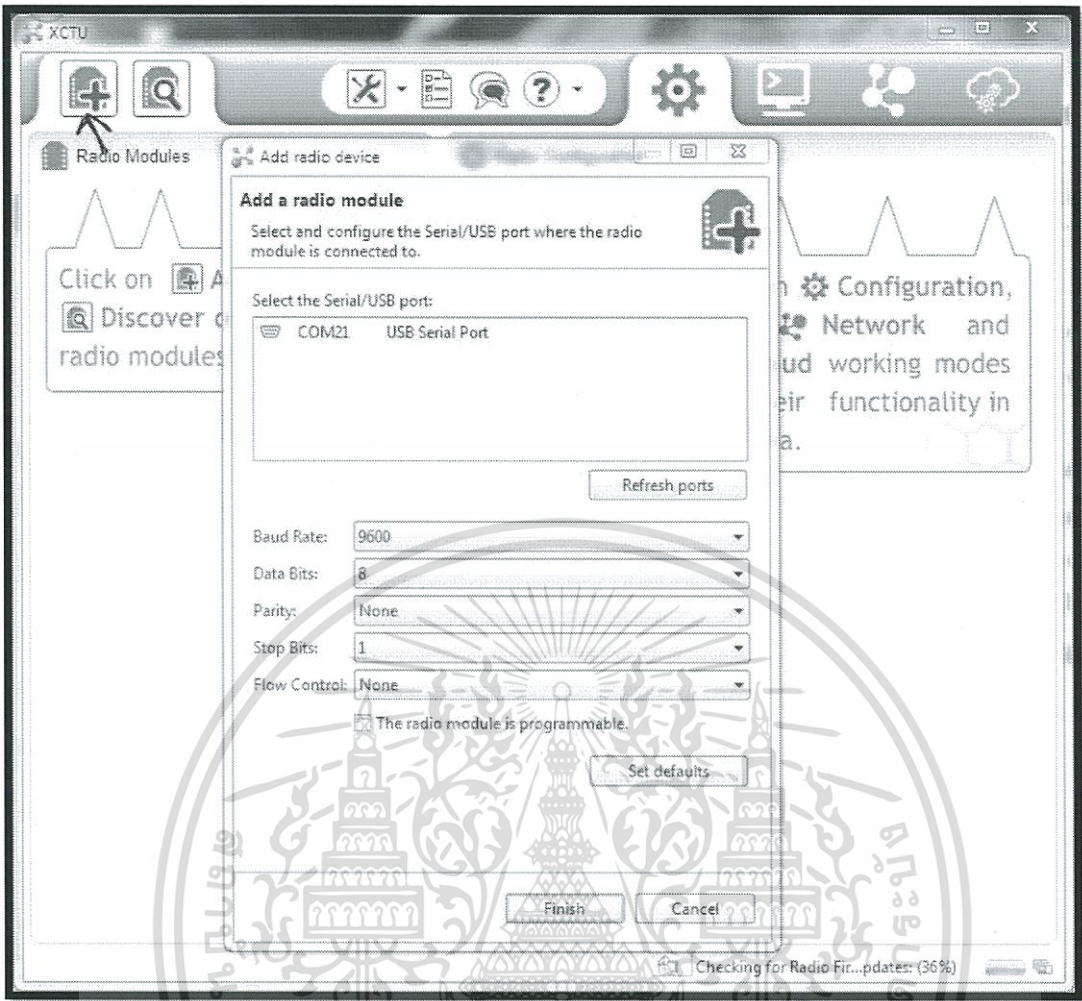


รูปที่ 1

XCTU เป็นโปรแกรมสำหรับการปรับแต่งค่าการทำงานของ XBee Pro ซึ่งใช้เป็นอุปกรณ์ในการรับส่งข้อมูลไร้สายในโครงงานนี้ ซึ่งอุปกรณ์ตัวนี้สามารถทำงานได้ทันทีด้วยค่าเริ่มต้นจากโรงงานโดยไม่ต้องทำการปรับแต่งค่าใดๆ

โปรแกรมนี้สามารถดาวน์โหลดได้ฟรีจาก เว็บไซต์ของผู้ผลิต XBee ตาม URL ด้านล่างนี้
<http://www.digi.com/support/productdetail?pid=3352&type=utilities>

เมื่อทำการติดตั้งโปรแกรมเสร็จแล้ว เปิดโปรแกรมขึ้นมาจะเป็นหน้าต่างหลักของโปรแกรม โดยจะมีคำอธิบายส่วนต่างๆของโปรแกรกดังรูปที่ 1

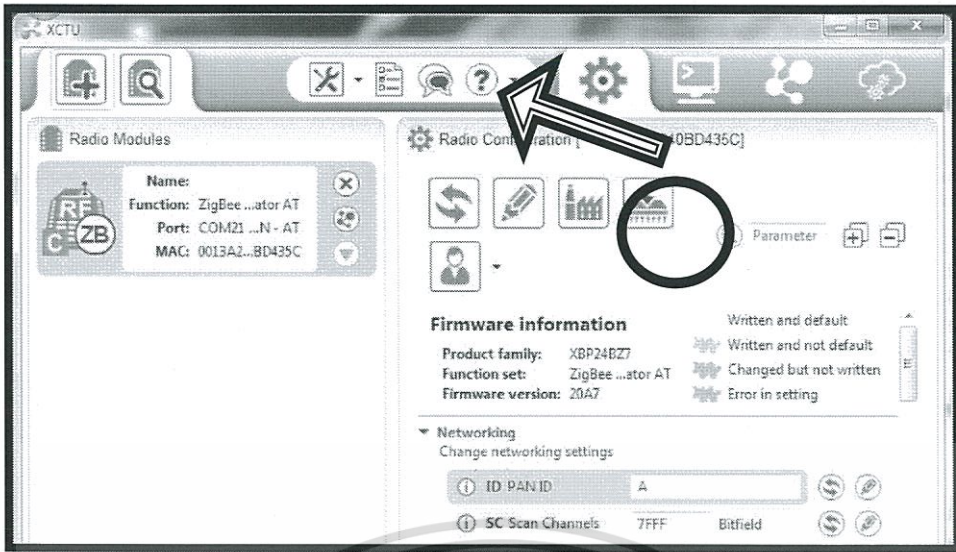


รูปที่ 2

ในการใช้งาน โปรแกรมเบื้องต้นให้ทำการเชื่อมต่อ XBee กับเครื่องคอมพิวเตอร์ จากนั้นคลิกที่ Add devices บริเวณมุมบนซ้ายตามลูกศรในรูปที่ 2 จะมีหน้าต่างขึ้นมาให้เลือก COM Port และการตั้งค่าการเชื่อมต่อ หากเป็นการใช้งานครั้งแรก สามารถกด Finish ได้เลย สำหรับค่าต่างๆที่จำเป็นต้องทราบคือ ค่า Baud Rate และ หมายเลข COM Port ซึ่งอุปกรณ์แต่ละรุ่นอาจใช้ค่าต่างกัน โดยสามารถศึกษารายละเอียดเกี่ยวกับอุปกรณ์ในแต่ละรุ่นได้จากเว็บไซต์ www.digi.com

ก่อนการใช้งาน โปรแกรมอาจมีข้อความแจ้งให้อัพเดทข้อมูลโปรแกรม ซึ่งควรอัพเดทโปรแกรมให้เสร็จก่อนการใช้งาน

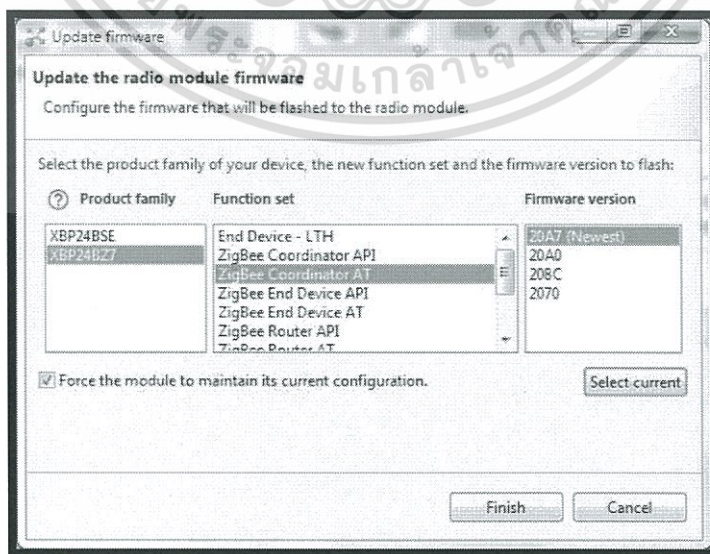
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3

เมื่อทำการติดต่อกับอุปกรณ์สำเร็จแล้วจะมีรูปแสดงขึ้นมาในส่วนของ Radio Modules ทางฝั่งซ้ายของหน้าต่างในรูปที่ 3 เมื่อคลิกที่รูปอุปกรณ์ดังกล่าวจะเข้าสู่โหมดการปรับค่าการทำงานของอุปกรณ์ซึ่งอยู่ทางฝั่งขวาของหน้าต่าง ตามรูปที่ 3 สำหรับรายละเอียดในส่วนต่างๆของการปรับแต่งจะไม่กล่าวถึงในที่นี้ ผู้ใช้สามารถศึกษาได้จากเมนู Help (?) ซึ่งอยู่บริเวณด้านบนของหน้าต่างโปรแกรม ตามที่ลูกศรชี้ในรูปที่ 3 สามารถตรวจสอบการอัปเดตข้อมูลด้วยตนเองจากส่วนนี้ได้

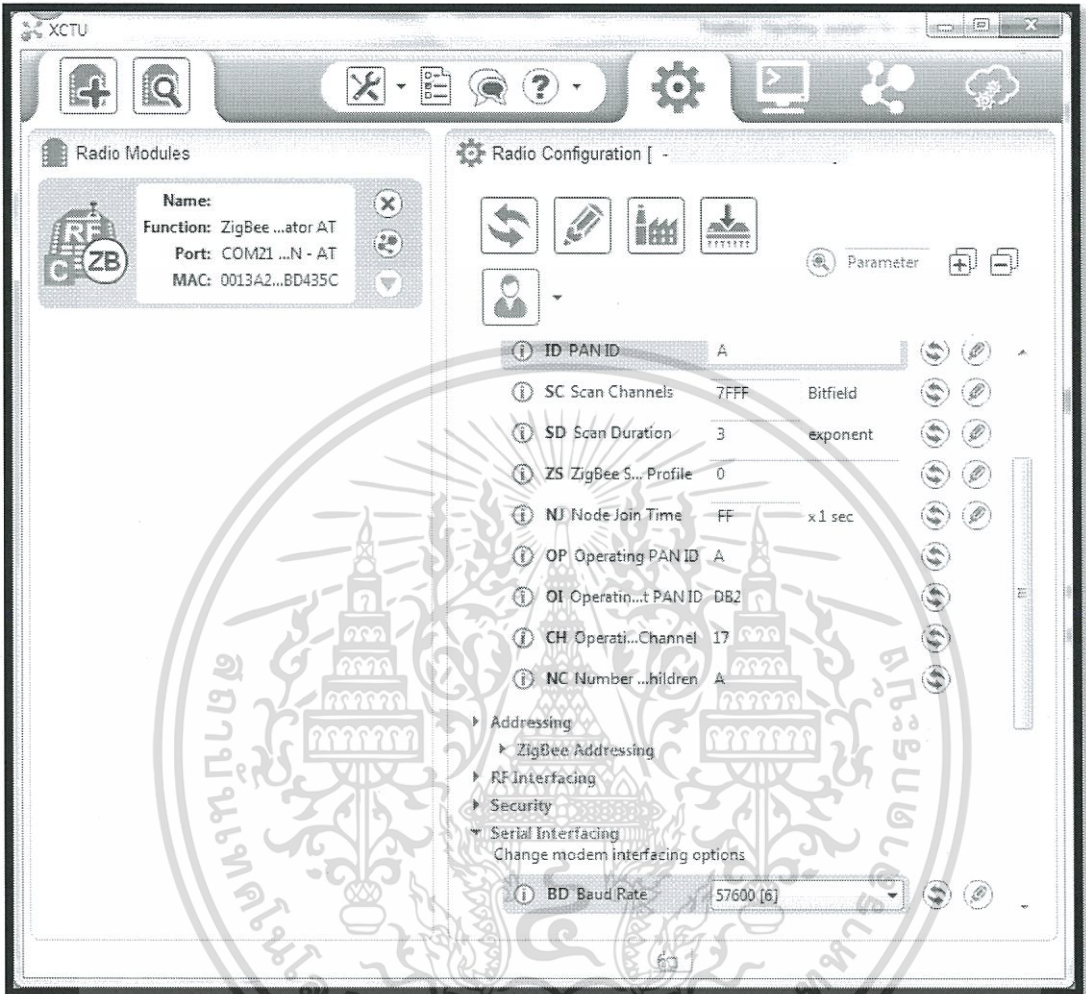
สำหรับการปรับค่าในโครงการนี้จะเริ่มจากการอัปเดต Firmware ของ Xbee ให้ทำงานในโหมดที่ต้องการ คือ Coordinator AT โดยคลิกบนเมนูที่วงกลมไว้ในรูปที่ 3 จากนั้นจะขึ้นหน้าต่างในรูปที่ 4 เพื่อเลือก Firmware ที่ต้องการ จากนั้นกดปุ่ม Finish เพื่อเริ่มการอัปเดต Firmware



รูปที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการอัปเดตเสร็จแล้ว โปรแกรมจะการติดต่อกับอุปกรณ์อีกครั้ง โดยอัตโนมัติ เมื่อทำการติดต่สำเร็จจะมีปุ่มให้กดตกลง เมื่อเสร็จสิ้นการอัปเดตต่อไปจะเป็นการตั้งค่าที่จำเป็น



รูปที่ 5

ค่าที่ทำการปรับแต่งคือ ค่า Baud Rate และค่า PAN ID โดยจะทำการปรับค่า Baud Rate เป็น 57600 และค่า PAN ID เป็น A เมื่อทำการปรับค่าเสร็จแล้วให้ทำการคลิกที่รูปดินสอในวงกลม ในรูปที่ 5 จะเป็นการเขียนค่าลงในหน่วยความจำของอุปกรณ์

เพื่อป้องกันการรบกวนที่อาจเกิดขึ้นหากมีคนใช้อุปกรณ์แบบเดียวกัน และใช้ค่าตั้งต้นเหมือนกัน จึงทำการปรับค่า PAN ID เป็น A

หลังจากทำการปรับค่าเสร็จแล้ว ให้นำ Xbee อีกตัวมาทำการปรับค่าเช่นเดียวกัน แต่เปลี่ยน Firmware จาก Coordinator AT ให้เป็น Router AT เพื่อให้สามารถรับ-ส่งข้อมูลได้ทันทีโดยไม่ต้องส่งคำสั่งให้อุปกรณ์

ในโครงการนี้ใช้ Coordinator AT เป็นฝั่งรับ และใช้ Router AT เป็นฝั่งส่ง เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลทางเทคนิคของอุปกรณ์ XBee PRO ZB รุ่น S2B ซึ่งใช้เป็นอุปกรณ์รับส่งข้อมูลในโรงงานนี้

Platform	XBee ZB	XBee-PRO ZB	Programmable XBee-PRO ZB
Performance			
RF Data Rate	250 Kbps		
Indoor/Urban Range	133 ft (40 m)	300 ft. (90 m)	
Outdoor/RF Line-of-Sight Range	400 ft (120 m)	2 miles (3200 m)/ Int'l 5000 ft (1500 m)	
Transmit Power	1.25 mW (+1 dBm) / 2 mW (+3 dBm) boost mode	63 mW (+18 dBm) / Int'l 10 mW (+10 dBm)	
Receiver Sensitivity (1% PER)	-96 dBm in boost mode	-102 dBm	
Features			
Adjustable Power	Yes		
I/O Interface	3.3V CMOS UART, ADC, DIO	3.3V CMOS UART, SPI, I2C, PWM, DIO, ADC	
Configuration Method	API or AT commands, local or over-the-air		
Frequency Band	2.4 GHz		
Interference Immunity	DSSS (Direct Sequence Spread Spectrum)		
Serial Data Rate	1200 bps - 1 Mbps		
ADC Inputs	(4) 10-bit ADC inputs		
Digital I/O	10		
Antenna Options	Chip, Wire Whip, U.FL, RPSMA	PCB Embedded Antenna, Wire Whip, U.FL, RPSMA	
Operating Temperature & Humidity	-40° C to +85° C, 0-95% humidity non-condensing		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Networking & Security			
Encryption	128-bit AES		
Reliable Packet Delivery	Retries/Acknowledgments		
IDs and Channels	PAN ID, 64-bit IEEE MAC, 16 channels	PAN ID, 64-bit IEEE MAC, 15 channels	
Power Requirements			
Supply Voltage	2.1 - 3.6VDC	2.7 - 3.6VDC	
Transmit Current	40 mA (@3.3 V)	205 mA	220 mA
Receive Current	38 mA / 40 mA boost mode @ 3.3VDC	47 mA	62 mA
Power-Down Current	<1 uA @ 25° C	3.5 uA @ 25° C	4 uA @ 25° C

ที่มา: <http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/zigbee-mesh-module/xbee-zb-module#specs>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

การปรับแต่ง AHRS FIRMWARE ด้วย ARDUINO IDE

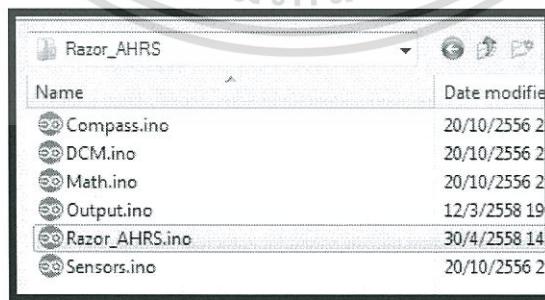
การปรับแต่ง AHRS Firmware เพื่อใช้ในโรงงานนี้เริ่มต้นจากการดาวน์โหลด Firmware จากเว็บไซต์ <https://github.com/ptrbrtz/razor-9dof-ahrs> แล้วทำการแตกไฟล์ Zip ที่ดาวน์โหลดมาให้เรียบร้อย โดยในโครงการนี้ข้อมูลจะอยู่ในโฟลเดอร์ชื่อ razor-9dof-ahrs-Release-v1.4.2

โปรแกรมที่ใช้ในการปรับแต่งคือโปรแกรม Arduino IDE ซึ่งสามารถดาวน์โหลดได้ฟรี จากเว็บไซต์ <http://www.arduino.cc/en/Main/Software> โดยในที่นี้จะใช้โปรแกรมเวอร์ชัน 1.6.3



รูปที่ 1

เริ่มต้นการแก้ไขด้วยการเปิดโปรแกรม Arduino IDE ขึ้นมา จะเป็นหน้าต่างโปรแกรมดังรูปที่ 1 จากนั้นเลือกเมนู File > Open เพื่อเปิดแฟ้มข้อมูลโดยที่แฟ้มดังกล่าวจะอยู่ในโฟลเดอร์ razor-9dof-ahrs-Release-v1.4.2\Arduino\Razor_AHRS โดยมีชื่อว่า **Razor_AHRS.ino** ดังรูปที่ 2



รูปที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Razor_AHRS | Arduino 1.6.3
File Edit Sketch Tools Help
Razor_AHRS Compass DCM Math Output Sensors
/*****
 * Razor AHRS Firmware v1.4.3
 * 9 Degree of Measurement Attitude and Heading Reference System
 * for Sparkfun "9DOF Razor IMU" (SEN-10125 and SEN-10736)
 * and "9DOF Sensor Stick" (SEN-10183, 10321 and SEN-10724)
 *
 * Released under GNU GPL (General Public License) v3.0
 * Copyright (C) 2013 Peter Bartz [http://ptrbrtz.net]
 * Copyright (C) 2011-2012 Quality & Usability Lab, Deutsche Telekom
 *
 * Infos, updates, bug reports, contributions and feedback:
 * https://github.com/ptrbrtz/razor-9dof-ahrs
 *
 * History:
 * * Original code (http://code.google.com/p/sf9dofmahrs/) by Doug
 *   based on ArduIMU v1.5 by Jordi Munez and William Premerlani,
 *
 * * Updated code (http://groups.google.com/group/sf_9dof_ahrs_up
 *****/

```

รูปที่ 3

เมื่อเปิดแฟ้มจะมีหน้าต่างใหม่ขึ้นมาโดยจะเป็นหน้าต่างที่ประกอบด้วยข้อมูลของ AHRS Firmware ทั้งหมด โดยแต่ละส่วนจะแบ่งเป็นแถบย่อยด้านบนดังรูปที่ 3 ซึ่งส่วนที่ต้องแก้ไขจะอยู่ในแถบที่ชื่อ Razor_AHRS และ Output โดยส่วนที่ต้องแก้ไขในแถบแรกคือ รุ่นของ IMU ที่ใช้ซึ่งอยู่บรรทัดที่ 161 แก้ไขด้วยการเอา // ออก จะได้เป็น `#define HW_VERSION_CODE 10736` สำหรับรายละเอียดอื่น ๆ และคำอธิบายการแก้ไขจะอยู่ในแฟ้มดังกล่าวเช่นกัน ส่วนถัดไปที่ต้องแก้ไขจะอยู่ประมาณบรรทัดที่ 188 โดยทำการแก้ไขค่าในรายการต่างๆ ดังนี้

`output_mode` เป็น `OUTPUT_MODE_ANGLES`;

`output_format` เป็น `OUTPUT_FORMAT_TEXT`;

`OUTPUT_STARTUP_STREAM_ON` เป็น `true`

ส่วน `SENSOR CALIBRATION` จะอยู่หลังจากบรรทัดที่ 210 เป็นต้นไป ซึ่งเป็นสิ่งจำเป็นที่ต้องทำแต่จะไม่กล่าวถึงในส่วนนี้ โดยขั้นตอนในการปรับค่ามาตรฐาน (calibrate) สามารถดูได้จาก <https://github.com/ptrbrtz/razor-9dof-ahrs/wiki/Tutorial#sensor-calibration> ซึ่งขั้นตอนต่างๆ ในเว็บไซต์ดังกล่าวได้เขียนไว้อย่างชัดเจนแล้ว

สำหรับแถบ Razor_AHRS จะสิ้นสุดการแก้ไขไว้เพียงเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การปรับแต่งค่าในแถบ Output จะทำการปรับแต่งการส่งออกค่าของ IMU ให้เป็นไปตามที่ ออกแบบไว้ สำหรับการแก้ไขในส่วนนี้จะเริ่มต้นจากการเพิ่มตัวแปรสำหรับอ้างอิงลำดับของข้อมูลที่ ส่งออก โดยการเพิ่มคำสั่ง byte counter = 0; ลงไปในบรรทัดที่ 4 จากนั้นทำการแก้ไขในส่วนของ ฟังก์ชัน output_angles ตั้งแต่บรรทัดที่ 15 เป็นต้นไป โดยจะทำการแก้ไขคำสั่งทั้งหมดในส่วน of else if (output_format == OUTPUT_FORMAT_TEXT) ซึ่งเมื่อทำการแก้ไขเสร็จแล้วจะได้ ชุดคำสั่งใหม่ในรูปแบบที่ 4 โดยชุดคำสั่งนี้เขียนขึ้นเพื่อให้ข้อมูลที่ส่งออกเป็นไปตามรูปแบบที่ต้องการ

```

Razor_AHRS  Compass  DCM  Math  Output
byte counter = 0;
void output_angles()
{
  if (output_format == OUTPUT_FORMAT_BINARY)
  {
    float ypr[3];
    ypr[0] = TO_DEG(yaw);
    ypr[1] = TO_DEG(pitch);
    ypr[2] = TO_DEG(roll);
    Serial.write((byte*) ypr, 12); // No new-line
  }
  else if (output_format == OUTPUT_FORMAT_TEXT)
  {
    Serial.print("$");
    Serial.print(counter++); Serial.print(",");
    Serial.print(roll); Serial.print(",");
    Serial.print(pitch); Serial.print(",");
    Serial.print(yaw); Serial.print(",");
    Serial.print(accel[0]); Serial.print(",");
    Serial.print(accel[1]); Serial.print(",");
    Serial.print(accel[2]); //Serial.print(",");
    Serial.println("#");
    if(counter == 50){
      counter = 0;
    }
  }
}

```

รูปที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการอัปเดตข้อมูล Firmware เข้าสู่ IMU จะต้องทำการเชื่อมต่อ IMU กับโปรแกรม Arduino IDE ผ่านทาง COM Port ก่อน โดยในที่นี้จะใช้ SparkFun FTDI Basic Breakout - 3.3V (<https://www.sparkfun.com/products/9873>) เป็นอุปกรณ์ตัวกลางในการเชื่อมต่อ

เมื่อทำการเชื่อมต่อเสร็จแล้วให้ทำการเลือก COM Port โดยเข้าไปที่เมนู Tool > Port จากนั้นทำการเลือกชื่อ COM Port ที่เชื่อมต่อกับ IMU ผ่านอุปกรณ์ USB to Serial แล้ว

หลังจากทำการเลือก COM Port เสร็จแล้วให้ทำการเลือกรุ่นของ Board ที่ใช้โดยเข้าไปที่ Tools > Board แล้วเลือกเป็น Arduino Pro or Pro Mini

จากนั้นทำการเลือก Processor ที่อยู่บน Board ที่เลือก โดยเข้าไปที่เมนู Tools > Processor เลือกเป็น ATmega328 (3.3v, 8 MHz)

เมื่อทำการเลือกตามหัวข้อที่ผ่านมาทั้งหมดแล้วในส่วนล่างสุดของหน้าต่างโปรแกรม Arduino IDE จะแสดงข้อมูลต่าง ๆ ตามที่เราเลือกไว้ ดังรูปที่ 5



รูปที่ 5

เมื่อข้อมูลการเชื่อมต่อในรูปที่ 5 ถูกต้องแล้วสามารถทำการอัปเดต Firmware ที่แก้ไขเสร็จแล้วได้โดยการกดปุ่ม Upload ซึ่งอยู่บริเวณด้านล่างแถบเมนูดังรูปที่ 6 หรือกดแป้นลัด Ctrl+U



รูปที่ 6

สามารถตรวจสอบข้อมูลที่ส่งออกมาจาก IMU ได้ผ่านทาง Serial Monitor ซึ่งอยู่ในเมนู Tools > Serial Monitor หรือโดยการกดแป้นลัด Ctrl+Shift+M จากนั้นเลือก Baud rate ให้ถูกต้อง

ข้อมูลรายละเอียดทางเทคนิคของเซนเซอร์แต่ละตัวสามารถหาแหล่งข้อมูล (Links) ได้จากเว็บไซต์ <https://www.sparkfun.com/products/10736>

ข้อมูล Schematic ของ 9DOF Razor IMU สามารถดูได้จากแฟ้ม PDF ในที่อยู่ด้านล่างนี้

<http://cdn.sparkfun.com/datasheets/Sensors/IMU/9DOF-Razor-v22.pdf>

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์กับการใช้งานเพื่อการศึกษาดูเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

การติดตั้งส่วนเสริมที่จำเป็นในโปรแกรม PROCESSING 2

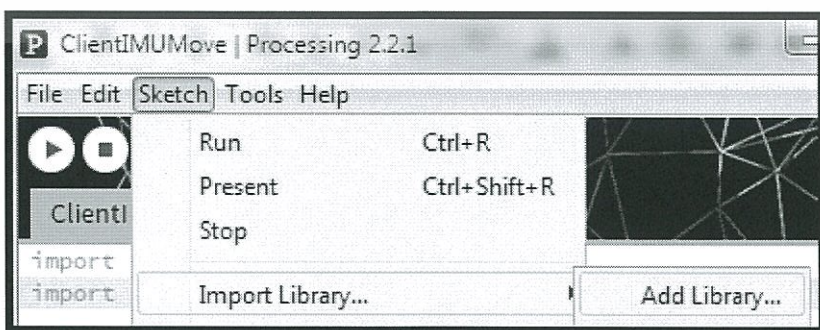
โปรแกรมประยุกต์ที่พัฒนาขึ้นในโครงการนี้ได้ใช้โปรแกรม Processing 2 ในการพัฒนา โดยได้ใช้ส่วนเสริมเพิ่มเติมนอกเหนือจากที่มีมาให้ในโปรแกรมเวอร์ชัน 2.2.1 คือ ControlP5 ซึ่งเป็นส่วนเสริมที่ช่วยในการสร้างกล่องข้อความและส่วนโต้ตอบกับผู้ใช้

โปรแกรม Processing 2 สามารถดาวน์โหลดได้ที่เว็บไซต์ของโครงการ Processing คือ <https://processing.org> โดยในเว็บไซต์ดังกล่าวจะประกอบไปด้วยข้อมูลต่างๆเกี่ยวกับโครงการ รวมถึงคู่มือการใช้งานโปรแกรม และสื่อการสอนต่างๆ เกี่ยวกับการเรียนรู้และใช้งานโปรแกรม เมื่อทำการติดตั้งเสร็จแล้ว เปิดโปรแกรมจะเป็นหน้าต่างโปรแกรมดังรูปที่ 1



รูปที่ 1

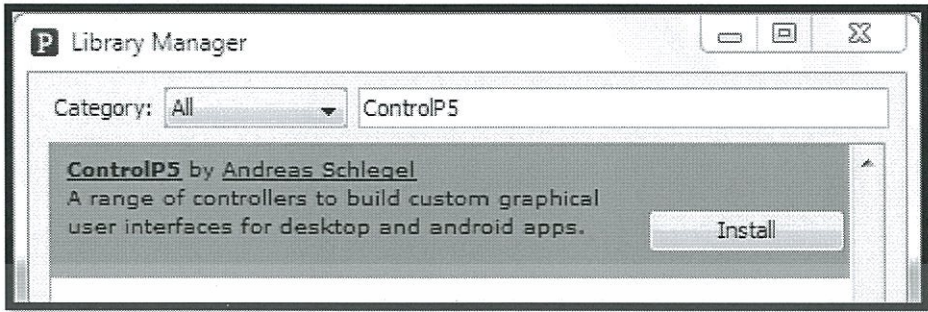
การติดตั้งส่วนเสริมเพิ่มเติมทำได้โดยการเข้าไปที่เมนู Sketch > Import Library > Add Library ดังรูปที่ 2 ซึ่งระหว่างการติดตั้งส่วนเสริมนั้นจำเป็นต้องเชื่อมต่ออินเทอร์เน็ต



รูปที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

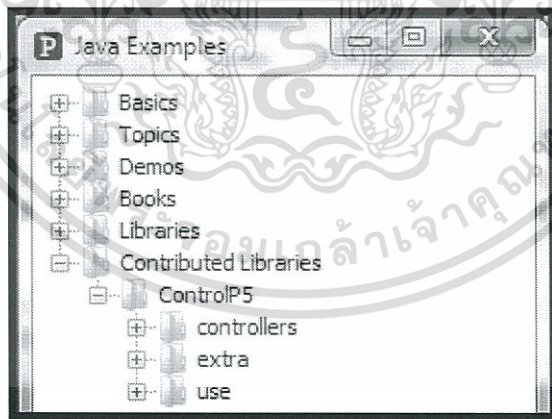
พิมพ์ชื่อส่วนเสริมที่ต้องการในช่องค้นหา ถ้ามีส่วนเสริมที่เกี่ยวข้องจะขึ้นในรายการด้านล่าง ดังรูปที่ 3 เมื่อพบส่วนเสริมที่ต้องการแล้วให้กดปุ่ม Install เพื่อติดตั้งส่วนเสริมดังกล่าว



รูปที่ 3

เมื่อระบบทำการดาวน์โหลดและติดตั้งส่วนเสริมดังกล่าวเสร็จสิ้นแล้วข้อความบนปุ่มจะเปลี่ยนเป็น Remove

การตรวจสอบว่าส่วนเสริมดังกล่าวสามารถใช้งานได้หรือไม่ทำได้โดยการทดลองรันเพิ่มตัวอย่างของส่วนเสริมดังกล่าวด้วยการกดแป้นลัด Ctrl+Shift+O ในโปรแกรม Processing 2 เพื่อเรียกการเพิ่มตัวอย่าง จากนั้นเลือกหัวข้อ Contributed Libraries > ชื่อส่วนเสริมที่ต้องการ ถ้าหากมีชื่อส่วนเสริมที่ทำการติดตั้งเพิ่มเติมให้เหลืออยู่ในรายการนี้แสดงว่าการติดตั้งสำเร็จ ดังรูปที่ 4



รูปที่ 4

การใช้งานโปรแกรม Processing 2 สามารถดูได้ในเว็บไซต์ <https://processing.org/tutorials>

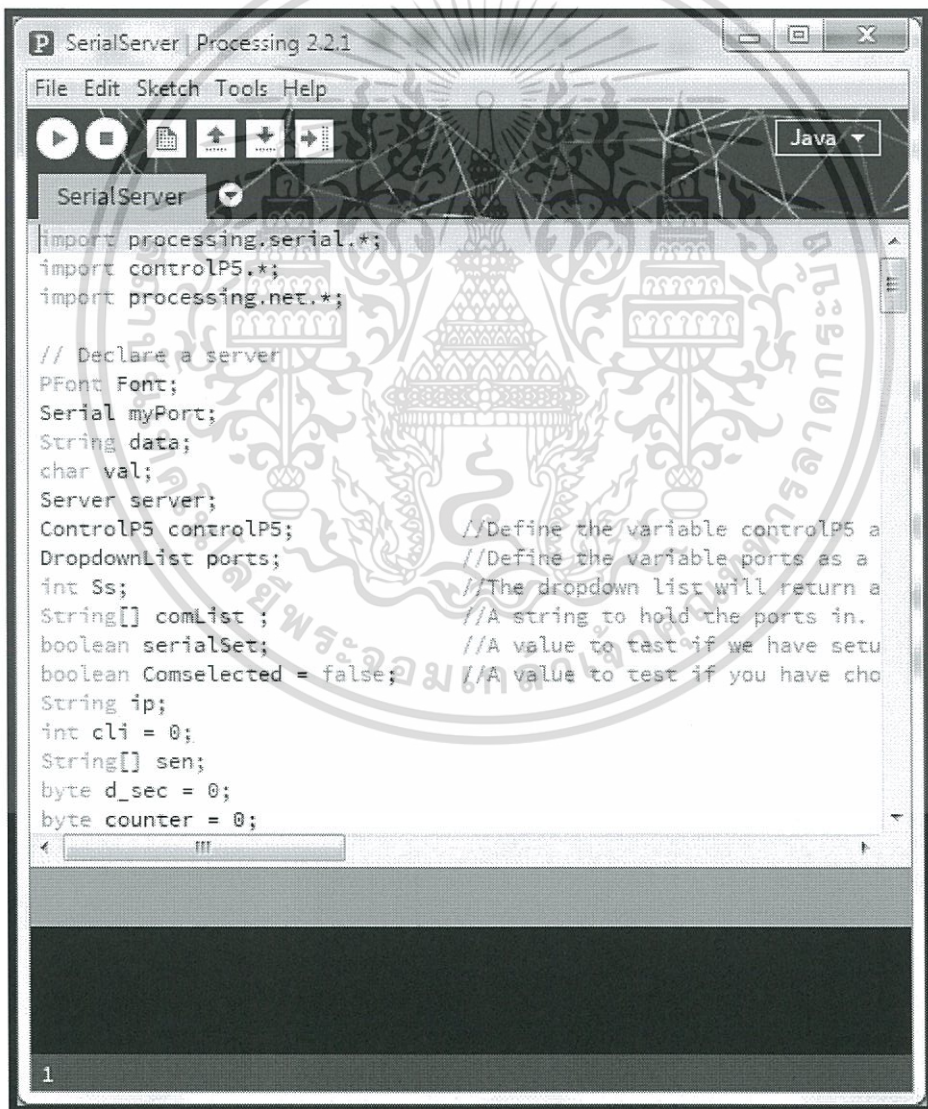
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง.

การใช้งานโปรแกรมที่พัฒนาขึ้นในโครงการ

โปรแกรมที่พัฒนาขึ้นในโครงการนี้ประกอบด้วยโปรแกรมส่วนติดต่อ COM Port และโปรแกรมส่วนแสดง โดยโปรแกรมทั้งสองจำเป็นต้องใช้โปรแกรม Processing 2 ในการทำงาน

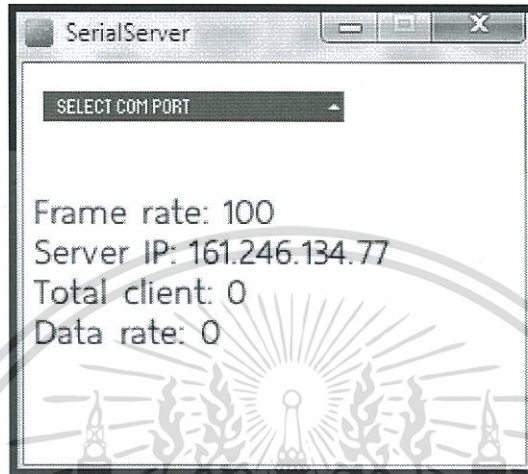
โปรแกรมส่วนติดต่อ COM Port เป็นแฟ้มข้อมูลของโปรแกรม Processing 2 การใช้งานจึงเริ่มจากการเปิดโปรแกรม Processing 2 ก่อนแล้วเปิดแฟ้มที่ชื่อ SerialServer.pde ซึ่งอยู่ในโฟลเดอร์ SerialServer เมื่อทำการเปิดแฟ้มดังกล่าวแล้วจะได้หน้าต่างของโปรแกรม Processing 2 ดังรูปที่ 1



รูปที่ 1

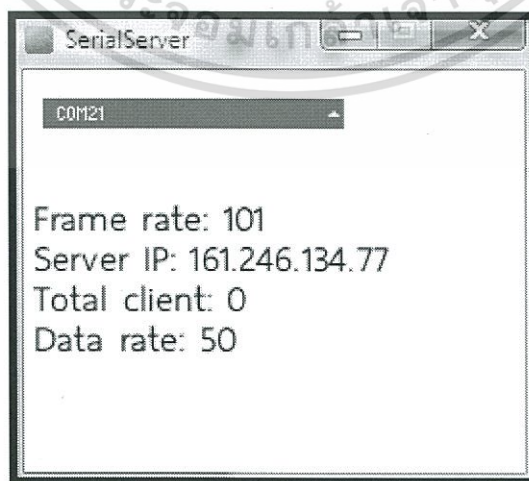
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก่อนการรัน โปรแกรมต้องตรวจสอบก่อนว่า ได้เชื่อมต่อกับอุปกรณ์รับส่งข้อมูลไร้สายไว้หรือไม่ ถ้าหากไม่ได้ต่อไว้หรือไม่พบ COM Port ของอุปกรณ์รับข้อมูลจะไม่สามารถรัน โปรแกรมได้ หากการเชื่อมต่อถูกต้อง และมี COM Port ในระบบเมื่อกดรัน โปรแกรมด้วยเมนู Run หรือเป็นลัด Ctrl+R จะมีหน้าต่างของโปรแกรมส่วนติดต่อกับ COM Port ขึ้นมาดังรูปที่ 2



รูปที่ 2

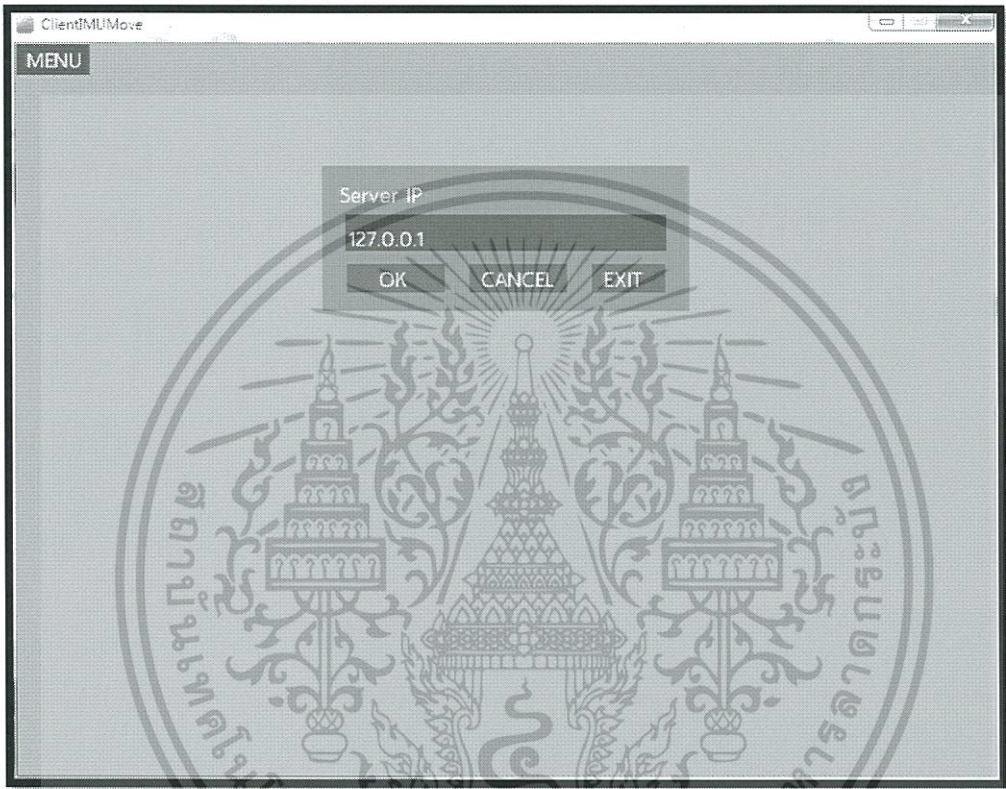
เมื่อมีหน้าต่างโปรแกรมปรากฏขึ้นมาแล้วให้ทำการคลิกที่เมนู SELECT COM PORT ในรูปที่ 2 จากนั้นเลือกชื่อในรายการที่แสดงโดยเลือกชื่อ COM Port ของอุปกรณ์รับข้อมูลไร้สายหลังจากที่เลือก COM Port แล้วหากชื่อที่เลือกถูกต้อง และมีการรับข้อมูลจากอุปกรณ์ไร้สายฝั่งส่ง จะได้นหน้าต่างของโปรแกรมดังรูปที่ 3 ซึ่งประกอบด้วยข้อมูลต่างๆในโปรแกรมที่จะบอกถึงสถานะการทำงานในขณะนั้น ได้แก่ ชื่อ COM Port, Frame rate, Server IP, Total client และ Data rate



รูปที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเปิดโปรแกรมส่วนแสดงผลทำเหมือนกับโปรแกรมส่วนติดต่อ COM Port คือเปิดผ่านโปรแกรม Processing 2 โดยเพิ่มของโปรแกรมจะอยู่ที่ไฟล์เคอร์ ClientIMUMove เมื่อเปิดโปรแกรมแล้วจะได้หน้าต่างโปรแกรม Processing 2 ขึ้นมาใหม่ เมื่อทำการรันโปรแกรมจะได้หน้าต่างของโปรแกรมดังรูปที่ 4 ซึ่งมีช่องให้ใส่หมายเลข IP ของโปรแกรมส่วนติดต่อ COM Port แต่ถ้าทั้งสองโปรแกรมทำงานในเครื่องเดียวกันจะสามารถใช้หมายเลข 127.0.0.1 ได้เลย

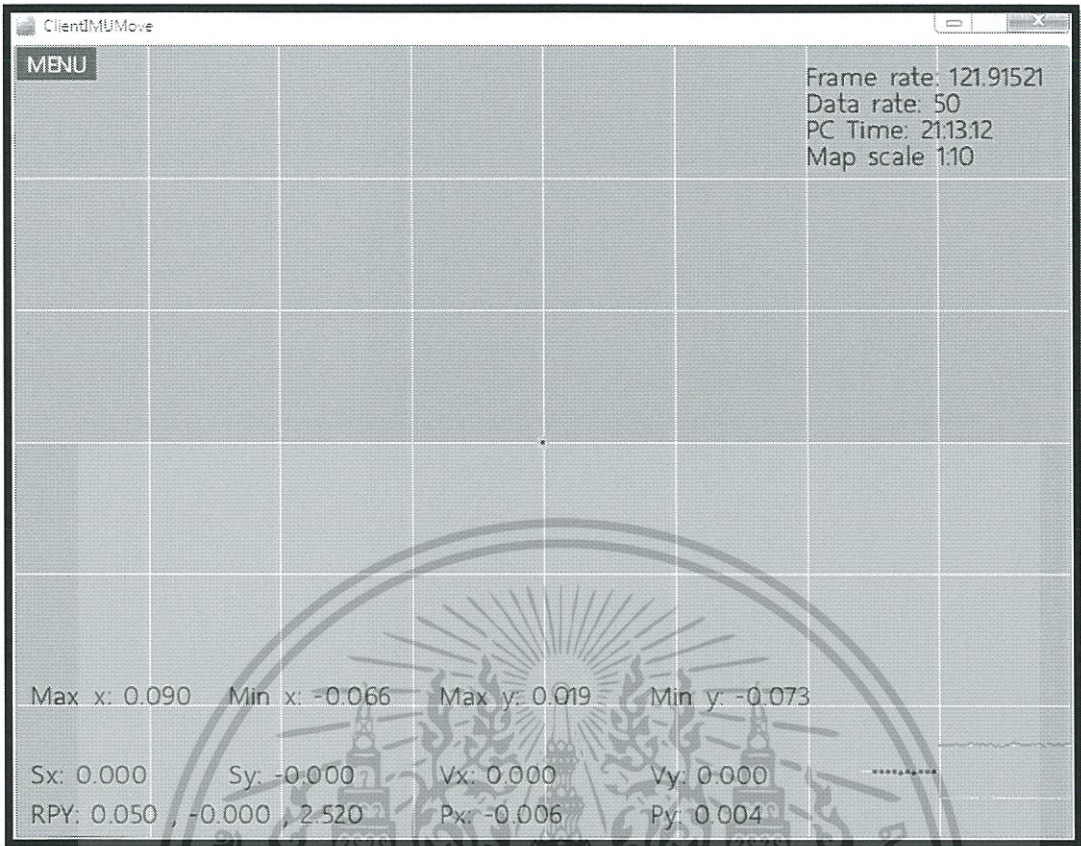


รูปที่ 4

ในการใช้งานโปรแกรมส่วนแสดงผลนั้นจะต้องมีการติดต่อกับโปรแกรมส่วนติดต่อ COM Port ผ่านทาง TCP/IP Protocol ซึ่งการเชื่อมต่ออาจถูกปิดกั้นจากโปรแกรมประเภท Firewall ในเครื่องคอมพิวเตอร์ที่รันโปรแกรมดังกล่าวทั้งสองได้ ดังนั้นควรทำการตรวจสอบโปรแกรมประเภท Firewall ในเครื่องคอมพิวเตอร์ก่อนหากหมายเลข IP ถูกต้องแต่ไม่สามารถเชื่อมต่อได้

การทำงานในของโปรแกรมแสดงผลจะเริ่มขึ้นเมื่อสามารถเชื่อมต่อและรับข้อมูลจากโปรแกรมส่วนติดต่อ COM Port ได้ การเชื่อมต่อทำได้โดยการใส่หมายเลข IP ในช่อง Server IP จากนั้นกดปุ่ม OK เพื่อทำการเชื่อมต่อ ถ้าหากการเชื่อมต่อสำเร็จหน้าต่างโปรแกรมจะเป็นดังรูปที่ 5 แต่ถ้าการเชื่อมต่อไม่สำเร็จหน้าต่างโปรแกรมจะเหมือนเดิมและจะมีข้อความรายงานความผิดพลาดขึ้นในส่วน Console ของโปรแกรม Processing 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5

ในหน้าต่างโปรแกรมจะประกอบด้วยข้อมูลต่างๆเพื่อใช้ประกอบในการติดตามตำแหน่งการเคลื่อนที่ ส่วนข้อมูลที่ใช้แสดงสถานะในการทำงานของโปรแกรมคือข้อมูลในส่วนมุมบนขวา ซึ่งประกอบไปด้วย Frame rate และ Data rate

สำหรับ Frame rate จะบอกถึงจำนวนเฟรมในการแสดงผลซึ่งในโปรแกรมทำการตั้งค่าไว้ที่ 300 หากค่าที่แสดงต่ำกว่า 300 แสดงว่าโปรแกรมมีการแสดงผลช้ากว่าที่กำหนดไว้ ซึ่งจะส่งผลให้เกิดความล่าช้าในการแสดงผลตำแหน่งปัจจุบัน เมื่อเทียบกับการเคลื่อนที่จริง เช่น มีการเคลื่อนที่ไปแล้ว แต่โปรแกรมยังไม่แสดงผลภายใน 1 วินาที แต่ต้องรอถึงประมาณ 4 วินาที โปรแกรมจึงเริ่มการแสดงผลการเคลื่อนที่ เป็นต้น

สำหรับ Data rate จะเป็นจำนวนชุดข้อมูลที่โปรแกรมได้รับใน 1 วินาที ซึ่งค่าดังกล่าวควรจะแสดงอยู่ที่ 50 เพราะในโครงงานนี้ได้กำหนดให้ IMU ส่งข้อมูลออกมา 50 ชุด ต่อ 1 วินาที หากค่าที่แสดงไม่ใกล้เคียง 50 แสดงว่ามีปัญหาในการรับส่งข้อมูล หากเป็นค่า Data rate ในโปรแกรมส่วนติดต่อ COM Port จะหมายถึงปัญหาการรับส่งข้อมูลระหว่างอุปกรณ์ไร้สาย แต่หากค่าที่แสดงในโปรแกรมส่วนแสดงผล และ โปรแกรมส่วนติดต่อ COM Port ไม่ใกล้เคียงกันจะหมายถึงปัญหาในการรับส่งข้อมูลระหว่างสองโปรแกรมดังกล่าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ-สกุล นายศักย์ปณชัย เกศศิชาปกรณ์
 วัน เดือน ปีเกิด 5 ตุลาคม 2532
 ที่อยู่ บ้านเลขที่ 69/45 ถนนนวมินทร์ แขวงรามอินทรา เขตคันนายาว
 กรุงเทพมหานคร 10230

ประวัติการศึกษา

พ.ศ. 2550 หลักสูตรมัธยมศึกษาตอนปลาย
 โรงเรียนสันป่าตองวิทยาคม อำเภอสันป่าตอง จังหวัดเชียงใหม่
 พ.ศ. 2557 วิทยาศาสตรบัณฑิต สาขาเทคโนโลยีสารสนเทศ
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อ้างอิง [1] ก่อนเริ่มคำนวณหาตำแหน่ง รวมทั้งใช้เทคนิค Self-Resetting [4] ในการลดค่าความผิดพลาดสะสมของระยะทาง (Distance) ในขณะที่หยุดนิ่ง โดยในโครงการได้ทำการพัฒนาระบบและทดสอบความถูกต้องภายใต้สภาพแวดล้อมต่าง ๆ เพื่อเปรียบเทียบและวิเคราะห์ผลการทำงานของระบบที่ถูกพัฒนาขึ้น

2. ทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง

ระบบติดตามตำแหน่งการเคลื่อนที่ในโครงการนี้จะใช้ระบบนำร่องเฉื่อยในการหาตำแหน่งการเคลื่อนที่ และใช้ค่าทิศทางจากระบบอ้างอิงทิศทางและการวางตัวในการอ้างอิงทิศทางเคลื่อนที่ ส่วนข้อมูลที่ใช้ในการคำนวณทั้งหมดจะมาจาก Razor IMU ซึ่งทำงานด้วย AHRS Firmware ซึ่งมีหลักการทำงานดังอธิบายในหัวข้อต่อไปนี้

2.1 ระบบนำร่องเฉื่อย

โครงสร้างของระบบนำร่องเฉื่อย (INS) ประกอบด้วยหน่วยวัดแรงเฉื่อย (IMU) หน่วยประมวลผล ส่วนติดต่อ และหน่วยจ่ายพลังงาน หลักในการหาตำแหน่งของ INS คือ การหาปริพันธ์อันดับสองของความเร่งโดยการหาปริพันธ์ในครั้งแรกจะได้ค่าความเร็ว เมื่อทำการหาปริพันธ์ของความเร็วอีกครั้งจะได้ค่าตำแหน่ง เมื่อนำตำแหน่งในช่วงเวลาปัจจุบันลบกับตำแหน่งในช่วงเวลาก่อนหน้าจะได้ระยะการเคลื่อนที่ และเมื่อนำระยะการเคลื่อนที่มารวมกันตั้งแต่เวลาที่เริ่มการทำงานจนถึงเวลาปัจจุบันจะได้ระยะทางรวมในการเคลื่อนที่ซึ่งจะใช้ในการติดตามตำแหน่งการเคลื่อนที่

สำหรับสมการที่ใช้ในการหาตำแหน่งจากความเร็วและความเร่งแสดงในสมการ (1) ส่วนสมการที่ใช้ในการคำนวณหาตำแหน่งจากความเร็วและความเร่ง แสดงในสมการที่ (2) ตามลำดับ

$$v = v_0 + at \quad (1)$$

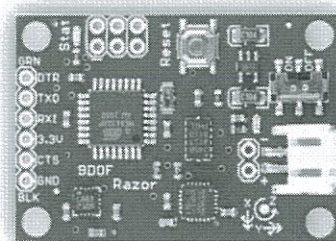
$$s = s_0 + v_0t + \frac{1}{2}at^2 \quad (2)$$

โดยที่ s คือค่าตำแหน่งที่ได้จากการคำนวณในช่วงเวลาปัจจุบัน v คือค่าความเร็วที่ได้จากการคำนวณในช่วงเวลาปัจจุบัน s_0 คือค่าตำแหน่งในช่วงเวลาที่ผ่านมา v_0 คือค่าความเร็วในช่วงเวลาที่ผ่านมา t คือเวลาระหว่างชุดข้อมูล และ a คือค่าความเร่งในช่วงเวลาปัจจุบัน [3]

อย่างไรก็ตาม ข้อจำกัดของระบบ INS ที่เกิดขึ้นจากการคำนวณด้วยกระบวนการดังกล่าวคือค่าความคลาดเคลื่อนจากการคำนวณปริพันธ์ ซึ่งจะเกิดขึ้นเล็กน้อยทุกครั้งที่ทำการคำนวณ แต่เมื่อเวลาผ่านไปจะส่งผลกระทบอย่างมากจากความคลาดเคลื่อนสะสมที่เกิดขึ้น

2.2 หน่วยวัดแรงเฉื่อย

หน่วยวัดแรงเฉื่อย (IMU) ที่ใช้ในโครงการคือ รุ่น 9DOF Razor IMU [6] (ดังรูปที่ 1) ประกอบด้วย 1) ITG-3200 ซึ่งเป็นไจโรสโคปแบบ 3 แกน ใช้สำหรับอ่านค่าอัตราเร็วในการหมุนของ 2) ADXL345 ซึ่งเป็นมาตรความเร่งแบบ 3 แกน ใช้สำหรับอ่านค่าความเร่งของ 3) HMC5883L ซึ่งเป็นเครื่องวัดความเข้มข้นของสนามแม่เหล็กแบบ 3 แกน ใช้สำหรับอ่านค่าสนามแม่เหล็กโดยรอบ 4) ATmega328 (MCU) ซึ่งเป็นไมโครคอนโทรลเลอร์ความถี่ 8MHz ทำหน้าที่อ่านข้อมูลจากเซนเซอร์แล้วทำการประมวลผลก่อนส่งออกทาง TTL Serial port โดย IMU ตัวนี้จะใช้ Razor AHRS Firmware [7] เป็นระบบควบคุมการทำงาน ซึ่งสามารถแก้ไขการทำงานได้ด้วยโปรแกรม Arduino IDE

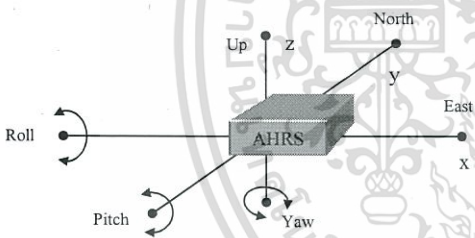


รูปที่ 1. 9DOF Razor IMU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ระบบอ้างอิงทิศทางและการวางตัว

ระบบอ้างอิงทิศทางและการวางตัว (AHRS) เป็นระบบที่ใช้สำหรับอ้างอิงทิศทางและการวางตัว โดยจะใช้ค่ามุม 3 ค่า คือ Roll (ϕ) Pitch (θ) และ Yaw (ψ) หรือ RPY ตามลำดับ สำหรับพิกัดที่เคลื่อนที่บนพื้นดิน ค่ามุม RPY จะอ้างอิงกับระบบพิกัด East-North-Up (ENU) ซึ่งเป็นระบบพิกัดในการอ้างอิงบนพื้นผิวโลก โดยแกน x คือทิศตะวันออก แกน y คือทิศเหนือ และแกน z คือทิศที่ชี้ออกจากโลก เมื่อนำมุม RPY มาอ้างอิงกับระบบพิกัด ENU จะได้ตามรูปที่ 2 คือ Roll เป็นค่าในการหมุนรอบแกน x เป็นบวกในทิศวนขวา เป็นศูนย์เมื่อขนานพื้นโลก Pitch เป็นค่าในการหมุนรอบแกน y เป็นบวกในทิศวนขวา เป็นศูนย์เมื่อขนานพื้นโลก และ Yaw เป็นค่าในการหมุนรอบแกน z เป็นบวกในทิศวนขวา เป็นศูนย์เมื่อชี้ไปยังทิศเหนือ



รูปที่ 2. AHRS RPY-ENU Reference

2.4 เมตริกซ์การแปลง

การแปลงระบบพิกัดอ้างอิงของค่าจาก IMU ให้อยู่ในระบบพิกัดนำร่องด้วยค่า RPY จาก AHRS ทำได้โดยใช้เมตริกซ์การแปลงในสมการที่ (3) ดังนี้ [5]

$$\begin{bmatrix} P_c Y_c & P_c Y_s & -P_s \\ Y_c P_s R_s - R_c Y_s & R_c Y_c + P_s R_s Y_s & P_c R_s \\ R_c Y_c P_s + R_s Y_s & R_c P_s Y_s - Y_c R_s & P_c R_c \end{bmatrix} \quad (3)$$

โดยที่ R P และ Y คือมุม Roll มุม Pitch และมุม Yaw ตามลำดับ ส่วนตัวห้อย s และ c คือ ค่าไซน์ (sin) และ โคไซน์ (cos) ของค่ามุมที่ถูกห้อยตามลำดับ เช่น $P_s = \sin(\theta)$ จาก AHRS เป็นต้น สำหรับเมตริกซ์ในสมการที่ (3) นี้ จะถูกใช้ในการปรับค่าระบบอ้างอิงและการปรับค่าแรงโน้มถ่วงของโลกออกจากความเร่งในระบบที่พัฒนาขึ้นนี้

2.5 การปรับค่าแรงโน้มถ่วง

ค่าแรงโน้มถ่วงที่ใช้กำหนดให้ $1g$ มีค่าเท่ากับ 9.81 m/s^2 เมื่อนำเอาเมตริกซ์การแปลงในสมการที่ (3) มาคูณกับเมตริกซ์แรงโน้มถ่วงของโลกจะได้ผลลัพธ์เป็นเมตริกซ์ของค่าความเร่งจากแรงโน้มถ่วงของโลก จากนั้นให้นำเมตริกซ์ดังกล่าวมาลบกับเมตริกซ์ของค่าความเร่งจาก IMU จะได้ผลลัพธ์เป็นเมตริกซ์ของค่าความเร่งที่ถูกปรับเอาแรงโน้มถ่วงของโลกออกแล้วในสมการที่ (4)

$$G = \begin{bmatrix} IMU_{ax} + 9.81P_s \\ IMU_{ay} - 9.81P_c R_s \\ IMU_{az} - 9.81P_c R_c \end{bmatrix} \quad (4)$$

โดยที่ G คือ เมตริกซ์ของค่าความเร่งที่ถูกปรับเอาแรงโน้มถ่วงของโลกออกแล้ว ส่วนค่า P_s , P_c , R_s และ R_c จะอ้างอิงกับค่าในสมการที่ (3) ส่วน IMU_{ax} , IMU_{ay} และ IMU_{az} คือค่าความเร่งแกน x y และ z จาก IMU ตามลำดับ

2.6 การแปลงระบบพิกัดของค่าความเร่ง

การแปลงระบบพิกัดของค่าความเร่งจากระบบพิกัดอ้างอิงเฉื่อยให้อยู่ในระบบพิกัดนำร่อง ทำได้โดยใช้เมตริกซ์การแปลงในสมการที่ (3) คูณกับเมตริกซ์ของค่าความเร่งจากสมการที่ (4) ผลลัพธ์จะเป็นเมตริกซ์ของค่าความเร่งที่ปรับระบบอ้างอิงแล้วในสมการที่ (5)

$$\begin{bmatrix} G_1 P_c Y_c - G_1 P_s + G_2 P_c Y_s \\ G_1 P_c R_s + G_1 (Y_c P_s R_s - R_c Y_s) + G_2 (R_c Y_c + P_s R_s Y_s) \\ G_1 P_c R_c + G_2 (R_c P_s Y_s - Y_c R_s) + G_1 (R_c Y_c P_s + R_s Y_s) \end{bmatrix} \quad (5)$$

โดยที่ G_1 และ G_2 คือค่าแถวที่ 1 และ 2 ในหลักที่ 1 ของเมตริกซ์ในสมการที่ (4) ตามลำดับ ส่วนค่า P_s P_c R_s R_c Y_s และ Y_c จะอ้างอิงกับสมการที่ (3)

2.7 การปรับทิศทางเคลื่อนที่

การปรับทิศทางในการเคลื่อนที่จะเกิดขึ้นหลังจากการคำนวณตำแหน่งการเคลื่อนที่ด้วยสมการที่ (2) เสร็จสิ้นแล้วทั้งในแกน x และในแกน y โดยใช้สมการที่ (6) และ (7) ดังนี้

$$p_x = s_x \cos(\psi) - s_y \sin(\psi) \quad (6)$$

$$p_y = s_x \sin(\psi) - s_y \cos(\psi) \quad (7)$$

โดยที่ p_x และ p_y คือตำแหน่งบนแกน x และ y ในโปรแกรม ส่วน s_x และ s_y คือตำแหน่งในแกน x และ y ที่คำนวณได้จากสมการที่ (2) และ $\cos(\psi)$ และ $\sin(\psi)$ คือค่าโคไซน์และไซน์ของมุม ψ จาก AHRS ซึ่งการปรับทิศทางนี้จะเป็นการปรับทิศทางเพื่อการแสดงผลในโปรแกรมส่วนแสดงผล

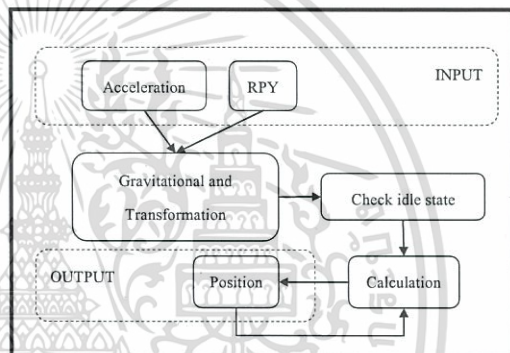
3. การพัฒนาระบบติดตามตำแหน่ง

การพัฒนาระบบติดตามตำแหน่งการเคลื่อนที่ในโครงการนี้ได้พัฒนาระบบโดยใช้หลักการหาค่าตำแหน่งด้วยความเร่ง มีการปรับค่าแรงโน้มถ่วงออกจากความเร่ง และมีการปรับค่าระบบอ้างอิงด้วยเมตริกซ์การแปลงที่ใช้ข้อมูลทิศทางการวางตัวจาก AHRS มีการหาภาวะหยุดนิ่งด้วยค่าเฉลี่ยจากขนาดของความเร่งในช่วงเวลาหนึ่งเพื่อลดค่าความผิดพลาดสะสมขณะหยุดนิ่ง โดยข้อมูลความเร่งและ

ข้อมูลทิศทางการวางตัวได้มาจาก Razor IMU ที่ทำงานด้วย AHRS Firmware ซึ่งมีการปรับแต่งในส่วนของการส่งข้อมูล เพื่อให้สะดวกต่อการใช้งานร่วมกับชุดรับส่งข้อมูลไร้สาย ที่ติดตั้งอยู่บนรถจำลอง

3.1 โครงสร้างระบบติดตามตำแหน่ง

ระบบ INS ที่ออกแบบสำหรับโครงการนี้เป็นระบบที่ไม่ซับซ้อน โดยอาศัยค่าความเร่งและทิศทางการวางตัวที่ได้มาจากอุปกรณ์ Razor IMU ในการประมวลผล



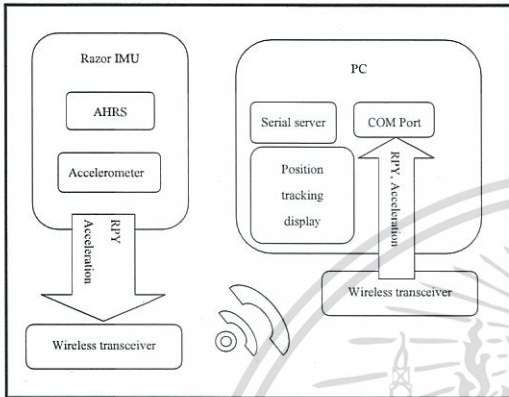
รูปที่ 3. โครงสร้างของระบบติดตามตำแหน่ง

กระบวนการทำงานในรูปที่ 3 เริ่มจากนำค่าที่ใช้สำหรับการทำงานของระบบ คือ ค่าความเร่ง (Acceleration) และทิศทางการวางตัว (RPY) ที่ได้มาจาก Razor IMU มาผ่านกระบวนการปรับค่าแรงโน้มถ่วงและการปรับค่าระบบอ้างอิงเพื่อใช้ในการคำนวณ จากนั้นจะทำการตรวจสอบว่าอยู่ในสภาวะหยุดนิ่งหรือไม่ก่อนทำการคำนวณด้วยสมการที่ (2) เพื่อหาค่าตำแหน่งและการแสดงผลในโปรแกรมส่วนแสดงผลต่อไป

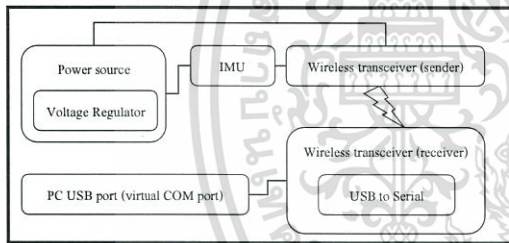
3.2 องค์ประกอบด้านฮาร์ดแวร์และการเชื่อมต่อ

รูปที่ 4 และ 5 แสดงองค์ประกอบด้านฮาร์ดแวร์ของระบบที่ใช้ในการเชื่อมต่อและรับส่งข้อมูลในระบบ โดยข้อมูลจาก IMU จะถูกส่งไปยัง PC ผ่านเครือข่ายไร้สาย

โดยข้อมูลที่ส่งจะประกอบด้วยลำดับชุดข้อมูลค่ามุม Roll Pitch และ Yaw รวมทั้งค่าความเร็วในแกน x y และ z ตามลำดับ



รูปที่ 4. การรับส่งข้อมูลระหว่างอุปกรณ์

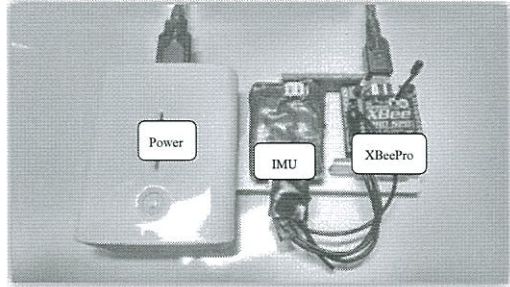


รูปที่ 5. การเชื่อมต่อระหว่างอุปกรณ์

โดยอุปกรณ์ที่เกี่ยวข้องในระบบดังรูปที่ 5 มีดังนี้

- 9DOF Razor IMU 1 ชุด
- XBee Pro Wireless transceiver 2 ชุด
- Power source สำหรับจ่ายไฟฟ้าให้กับอุปกรณ์ 1 ชุด

รูปที่ 6 แสดงให้เห็นโครงสร้างทางฮาร์ดแวร์ของอุปกรณ์ในรูปที่ 5 ซึ่งนำมาประกอบรวมกันสำหรับอ่านค่าจาก IMU ผ่านระบบไร้สาย โดยชุดอุปกรณ์ดังกล่าวจะถูกนำไปติดตั้งบนรถจำลองเพื่อใช้ในการทดสอบ (ดังรูปที่ 7) โดยส่วนจ่ายพลังงานจะติดตั้งอยู่บริเวณใต้ห้องรถส่วน IMU และ Wireless transceiver จะติดอยู่บนตัวรถ



รูปที่ 6. ชุดอุปกรณ์สำหรับติดตั้งบนรถจำลอง



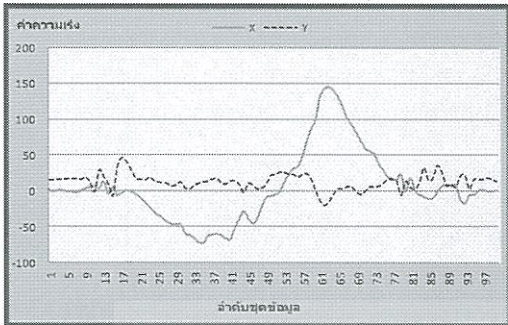
รูปที่ 7. รถจำลองที่ติดตั้งอุปกรณ์ติดตามตำแหน่งแล้ว

3.3 การอ่านค่าจากชุดอุปกรณ์ทดสอบ

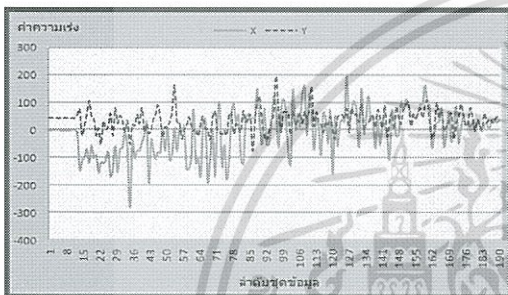
ชุดอุปกรณ์สำหรับติดตั้งบนรถจำลองที่พัฒนาขึ้นได้ผ่านการทดสอบการอ่านค่าความเร็วแล้วนำค่ามาสร้างกราฟเพื่อดูความแตกต่างของความเร็วที่ได้ระหว่างการเคลื่อนที่ด้วยมือกับการเคลื่อนที่ด้วยรถจำลอง โดยกำหนดลักษณะการเคลื่อนที่เป็นแบบเส้นตรงและเป็นเส้นโค้ง

รูปที่ 8 เป็นกราฟความเร็วที่ได้จากการทดสอบการเคลื่อนที่ด้วยมือแบบเส้นตรง ซึ่งค่าความเร็วมีการแกว่งของค่าไม่มาก ส่วนรูปที่ 9 เป็นกราฟความเร็วที่ได้จากการทดสอบการเคลื่อนที่ของอุปกรณ์ที่ติดตั้งบนรถจำลองแบบเส้นตรง จากผลการทดลองเปรียบเทียบการเคลื่อนที่ในรูปที่ 8 และ 9 พบว่า การเคลื่อนที่ด้วยมือจะมีค่าการแกว่งของค่าความเร็วน้อยกว่ากราฟการเคลื่อนที่ของรถจำลองเป็นอย่างมาก ซึ่งการแกว่งของค่าความเร็วดังกล่าวเกิดจากการเคลื่อนที่ของรถจำลองที่มีความเร็วไม่คงที่ และมีการสั่นของตัวรถอยู่ตลอดเวลาในขณะที่เคลื่อนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



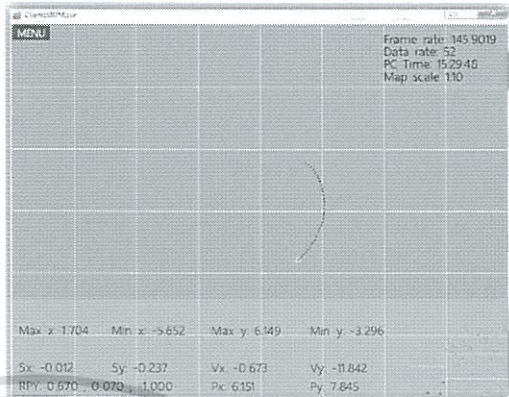
รูปที่ 8. กราฟความเร่งจากการเคลื่อนที่ด้วยมือ



รูปที่ 9. กราฟความเร่งจากการเคลื่อนที่ของรถจำลอง

3.4 การแสดงผลข้อมูลตำแหน่งการเคลื่อนที่

การแสดงผลการติดตามตำแหน่งการเคลื่อนที่ของรถจำลองจะใช้โปรแกรมประยุกต์ที่พัฒนาขึ้นเอง โดยจะแบ่งโปรแกรมเป็น 2 ส่วนคือ ส่วนติดต่อ COM Port สำหรับรับข้อมูลจาก Wireless transceiver และส่วนแสดงผลซึ่งรับข้อมูลจากส่วนติดต่อ COM Port มาทำการประมวลผลและแสดงผล ดังตัวอย่างหน้าจอของโปรแกรมขณะทำการติดตามตำแหน่งการเคลื่อนที่ในรูปแบบที่ 10



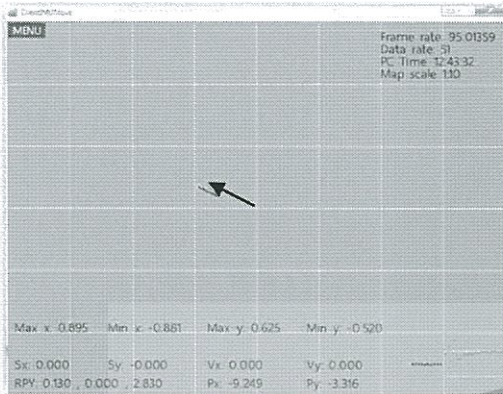
รูปที่ 10. GUI โปรแกรมติดตามตำแหน่งการเคลื่อนที่

4. การทดสอบระบบติดตามตำแหน่ง

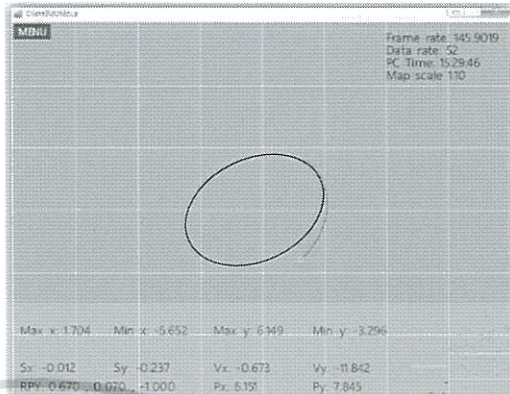
การทดสอบระบบติดตามตำแหน่งจะเป็นการทดลองติดตั้งชุดอุปกรณ์บนรถจำลองและให้เคลื่อนที่ด้วยการวิ่งในพื้นที่ที่กำหนด โดยกำหนดการเคลื่อนที่เป็นเส้นตรงและเส้นโค้ง (หรือวงกลม) ในการทดสอบดังต่อไปนี้

4.1 การทดสอบการติดตามตำแหน่งแบบเส้นตรง

การติดตามตำแหน่งการเคลื่อนที่ของรถจำลองแบบเส้นตรงได้ผลดังรูปที่ 11 โดยเทียบกับแนวเส้นลูกศรสีดำซึ่งใช้ในการอ้างอิงทิศทางและตำแหน่งการเคลื่อนที่จริงในการทดสอบ ผลการทดลองพบว่าทิศทางการเคลื่อนที่ของรถจำลองที่ปรากฏบนหน้าจอแสดงผลมีความถูกต้อง แต่ในส่วนของการติดตามตำแหน่งที่มีความคลาดเคลื่อนอยู่เล็กน้อย (เทียบกับแนวเส้นลูกศรสีดำ) โดยที่บริเวณหัวลูกศรสีดำคือตำแหน่งปัจจุบันของรถจำลองจริง ส่วนเส้นสีน้ำเงินและวงกลมสีเขียวคือตำแหน่งและทิศทางการเคลื่อนที่ของรถจำลองจากข้อมูล IMU ที่ได้ในระบบ



รูปที่ 11. ผลการติดตามตำแหน่งการเคลื่อนที่แบบเส้นตรง



รูปที่ 12. ผลการติดตามตำแหน่งการเคลื่อนที่แบบเส้นโค้ง

4.2 การทดสอบการติดตามตำแหน่งแบบเส้นโค้ง

การติดตามตำแหน่งการเคลื่อนที่ของรถจำลองแบบเส้นโค้งได้ผลดังรูปที่ 12 โดยเทียบกับเส้นวงรีสีดำซึ่งใช้อ้างอิงตำแหน่งและทิศทางการเคลื่อนที่จริงของรถจำลอง ในขณะที่ทดลอง ผลการทดลองพบว่าตำแหน่งและทิศทางการเคลื่อนที่ของรถจำลองจากข้อมูล IMU ที่ได้รับ (ดังปรากฏเป็นเส้นสีน้ำเงินและวงกลมสีเขียว) จะมีความถูกต้องของทิศทางการเคลื่อนที่ แต่ในส่วนของตำแหน่งการเคลื่อนที่และรัศมีในการเลี้ยวโค้งจะมีความคลาดเคลื่อนอยู่บ้างเมื่อเทียบกับวงรีสีดำ โดยเฉพาะในช่วงของรัศมีการเลี้ยว ซึ่งจะปรากฏความคลาดเคลื่อนของตำแหน่งค่อย ๆ เพิ่มขึ้น โดยสังเกตได้จากการเบี่ยงออกจากเส้นวงรีสีดำในช่วงที่เริ่มเข้าแนวเส้นตรง ซึ่งเป็นผลที่เกิดจากความคลาดเคลื่อนของตำแหน่งสะสมจากค่าความเร่งในขณะการเคลื่อนที่

5. สรุปผลโครงการ

โครงการนี้ได้นำเสนอการออกแบบและพัฒนาระบบติดตามตำแหน่งการเคลื่อนที่ของรถจำลองโดยการประยุกต์ใช้เทคนิคการหาตำแหน่งจาก INS และการปรับค่าด้วย AHRS รวมถึงการปรับลดความผิดพลาดสะสมขณะหยุดนิ่งด้วย Self-setting โดยกระบวนการในการคำนวณทั้งหมดในระบบที่พัฒนาขึ้นใช้ค่าข้อมูลจากอุปกรณ์ 9DOF Razor IMU เพียงตัวเดียว ซึ่งไม่ใช้การอ้างอิงสัญญาณจากภายนอก

โครงการได้ทำการพัฒนาระบบทั้งในส่วนของฮาร์ดแวร์ที่จัดสร้างขึ้นรวมทั้งในส่วนของซอฟต์แวร์ที่พัฒนาขึ้นสำหรับอ่านค่าข้อมูลและรับส่งข้อมูลจาก IMU ผ่านทางเครือข่ายไร้สายไปยังเครื่องคอมพิวเตอร์ จากนั้นจึงนำข้อมูลที่ได้อ่านประมวลผลหาตำแหน่งและทิศทางการเคลื่อนที่รวมทั้งแสดงผลที่บนหน้าจคอมพิวเตอร์

ในการทดสอบการทำงานของระบบได้ผลตามที่กำหนด โดยระบบสามารถใช้งานได้ตามวัตถุประสงค์คือ สามารถแสดงผลการติดตามตำแหน่งการเคลื่อนที่ของรถจำลองที่ติดตั้งชุดอุปกรณ์ที่ได้พัฒนาขึ้นได้ อย่างไรก็ตาม ผลการทดสอบยังมีค่าความคลาดเคลื่อนของทิศทางและตำแหน่งที่คำนวณได้ ทั้งนี้อาจเป็นเพราะข้อจำกัดต่าง ๆ ของ

ระบบที่พัฒนาขึ้น โดยเฉพาะการอ้างอิงทิศทางด้วย AHRS ซึ่งสามารถถูกรบกวนจากสนามแม่เหล็กได้ จึงทำให้ทิศทางการวิ่งของรถจำลองที่ใช้การทดสอบผิดเพี้ยนไปบ้างเล็กน้อย สำหรับปัญหาที่สำคัญคือ ตำแหน่งที่เคลื่อนที่จริงกับตำแหน่งที่คำนวณได้ยังคงมีความคลาดเคลื่อนสูง เนื่องจากการใช้ค่าข้อมูลจาก IMU เพียงอย่างเดียวและเป็น IMU คุณภาพต่ำ ทำให้ค่าที่ได้ไม่ละเอียดพอ นอกจากนี้ยังมีปัญหาจากระบบขับเคลื่อนของรถจำลองที่ใช้ในการทดสอบซึ่งมีแรงสั่นในขณะเคลื่อนที่สูง ทำให้ค่าความเร่งที่ได้จาก IMU ผิดเพี้ยนไปและทำให้เกิดการแกว่งของค่าความเร่งขณะเคลื่อนที่ด้วย สำหรับแนวทางในการปรับปรุงเพื่อแก้ข้อจำกัดของอุปกรณ์ดังกล่าวเพื่อให้ได้ผลลัพธ์ที่ถูกต้องแม่นยำมากขึ้นในอนาคต สามารถทำได้โดยอาศัยข้อมูลจากเซ็นเซอร์ชนิดอื่นเข้ามาช่วยในการอ้างอิงหรือปรับตำแหน่งจากความคลื่อนสะสม หรืออาศัยข้อมูลความเร็วในการเคลื่อนที่ของพาหะที่ไม่ได้มาจากการคำนวณจากค่าความเร่ง หรือใช้การอ้างอิงทิศทางด้วยข้อมูลจากแหล่งอื่นที่ไม่ถูกรบกวนด้วยสนามแม่เหล็กได้ เป็นต้น

เอกสารอ้างอิง

- [1] MS Grewal, LR Weill, AP Andrews. **Global Positioning Systems, Inertial Navigation, and Integration.** John Wiley & Sons Inc. 2007
- [2] B. Ladislav, C. Pavel, D. Martin. "Navigation of robotic platform with using inertial measurement unit and Direct Cosine Matrix" **56th International Symposium ELMAR-2014,** 1-4, IEEE. 2014
- [3] O. Wongwirat, C. Chaiyarat. "A position tracking experiment of mobile robot with Inertial Measurement Unit (IMU)" **International Conference on Control Automation and Systems (ICCAS) 2010,** 304-308, IEEE. 2010
- [4] E. Akeila, Z. Salcic, A. Swain. "Reducing Low-Cost INS Error Accumulation in Distance Estimation Using Self-Resetting" **IEEE Transactions on Instrumentation and Measurement,** vol. 63, no. 1 (2014), 177-184
- [5] E. Mark Pedley. **Tilt Sensing Using a Three-Axis Accelerometer** Freescale Semiconductor Inc., document number: AN3461, Rev. 6, 03/2013
- [6] SparkFun Electronics, "9 Degrees of Freedom - Razor IMU SEN-10736", [Online] Available : <https://www.sparkfun.com/products/10736>
- [7] ptrbrtz, "AHRS Firmware for the SparkFun 9DOF Razor IMU and SparkFun 9DOF Sensor Stick", [Online] Available : <https://github.com/ptrbrtz/razor-9dof-ahrs>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้