



ระบบควบคุมและระบุตำแหน่งของหุ่นยนต์สำรวจใต้น้ำแบบสี่องศาอิสระ
**Control System and Localization for the 4 DOF Remotely Operated
Underwater Vehicle**

ปรเมศวร์ สุวรรณวงศ์
Paramet Suwanwong

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมเครื่องกล
มหาวิทยาลัยสงขลานครินทร์

**A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of Master of Engineering in Mechanical Engineering
Prince of Songkla University**

2562

ลิขสิทธิ์ของมหาวิทยาลัยสงขลานครินทร์



ระบบควบคุมและระบุตำแหน่งของหุ่นยนต์สำรวจใต้น้ำแบบสี่องศาอิสระ
**Control System and Localization for the 4 DOF Remotely Operated
Underwater Vehicle**

ปรเมศวร์ สุวรรณวงศ์
Paramet Suwanwong

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมเครื่องกล
มหาวิทยาลัยสงขลานครินทร์

**A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of Master of Engineering in Mechanical Engineering
Prince of Songkla University**

2562

ลิขสิทธิ์ของมหาวิทยาลัยสงขลานครินทร์

ชื่อวิทยานิพนธ์ ระบบควบคุมและระบุตำแหน่งของหุ่นยนต์สำรวจใต้น้ำแบบส่องศานอิสระ
 ผู้เขียน นายปรเมศวร์ สุวรรณวงศ์
 สาขาวิชา วิศวกรรมเครื่องกล

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

คณะกรรมการสอบ

.....
 (รองศาสตราจารย์ ดร. พงษ์ทิกร สมิตไผตรี)

.....ประธานกรรมการ
 (ดร. ประมินทร์ เถรานนท์)

.....กรรมการ
 (ดร. สว่างทิตย์ ศรีกิจสุวรรณ)

.....กรรมการ
 (รองศาสตราจารย์ ดร. พงษ์ทิกร สมิตไผตรี)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้บัณฑิตวิทยาลัยเป็นส่วน
 หนึ่งของการศึกษา ตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชา
 วิศวกรรมเครื่องกล

.....
 (ศาสตราจารย์ ดร.ดำรงศักดิ์ ฟ้ารุ่งแสง)
 คณบดีบัณฑิตวิทยาลัย

ขอรับรองว่า ผลงานวิจัยนี้มาจากการศึกษาวิจัยของนักศึกษาเอง และได้แสดงความขอบคุณ
บุคคลที่มีส่วนช่วยเหลือแล้ว

ลงชื่อ.....
(รองศาสตราจารย์ ดร. พงษ์ทิกร สมิตไผตรี)
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

ลงชื่อ.....
(นายปรเมศวร์ สุวรรณวงศ์)
นักศึกษา

ข้าพเจ้าขอรับรองว่า ผลงานวิจัยนี้ไม่เคยเป็นส่วนหนึ่งในการอนุมัติปริญญาในระดับใดมาก่อน
และไม่ได้ถูกใช้ในการยื่นขออนุมัติปริญญาในขณะนี้

ลงชื่อ.....

(นาย ประเมศวร์ สุวรรณวงศ์)

นักศึกษา

ชื่อวิทยานิพนธ์	ระบบควบคุมและระบุตำแหน่งของหุ่นยนต์สำรวจใต้น้ำแบบส่องศก อิสระ
ผู้เขียน	นายปรเมศวร์ สุวรรณวงศ์
สาขาวิชา	วิศวกรรมเครื่องกล
ปีการศึกษา	2561

บทคัดย่อ

ปัจจุบันหุ่นยนต์บังคับใต้น้ำ (ROUV) กลายเป็นอุปกรณ์สำคัญสำหรับภารกิจใต้น้ำหลายอย่าง แต่ด้วยคุณลักษณะทางพลศาสตร์ของการเคลื่อนที่ใต้น้ำของ ROUV เช่น การไถลหลังจากการเคลื่อนที่ พลศาสตร์ของใบพัด ความสมดุลระหว่างน้ำหนักและแรงลอยตัว แรงกระทำที่ไม่แน่นอนจากสายเคเบิล และมุมมองภาพที่จำกัดขณะใช้งานใต้น้ำ ทำให้การควบคุม ROUV เป็นสิ่งที่ยากสำหรับผู้ควบคุมแม้จะใช้งานในน้ำนิ่ง งานวิจัยนี้จึงทำการออกแบบระบบควบคุมเสถียรภาพของการทรงตัวด้วยตัวควบคุม PI และพัฒนาการระบุตำแหน่งของ ROUV บนพื้นฐานของ Hector SLAM ซึ่งเป็นโปรแกรมที่ทำงานบน Robot Operating System (ROS) การควบคุมแบบ PID ถูกนำมาใช้เพื่อพัฒนาระบบควบคุมการทรงตัวโดยอาศัยสัญญาณป้อนกลับเป็นค่าความเร็วในแนวระนาบ ระดับความลึกและความเร็วเชิงมุมทิศ Yaw เพื่อควบคุมการเคลื่อนที่ของหุ่นยนต์ในทิศทางเหล่านั้น อุปกรณ์หลักของระบบประกอบด้วย คอมพิวเตอร์, บอร์ด Arduino และบอร์ด Raspberry Pi เป็นส่วนประมวลผล, มอเตอร์ขับเคลื่อน (Thruster), บอร์ดขับเคลื่อนมอเตอร์ (Drive board), เซนเซอร์วัดความดัน (Pressure sensor) สำหรับวัดระดับความลึก เซนเซอร์วัดทิศทาง (Gyroscope) และ เซนเซอร์วัดสนามแม่เหล็ก (Magnetometer) จาก IMU สำหรับวัดทิศ Yaw, เซนเซอร์วัดระยะด้วยเลเซอร์ (Lidar) สำหรับวัดตำแหน่งในแนวระนาบเพื่อหาตำแหน่งของตัวหุ่นยนต์โดยอาศัยอัลกอริทึม Scan Matching การทดสอบระบบใช้หุ่นยนต์ดำน้ำที่สร้างขึ้นเองในการทดสอบ โดยหุ่นยนต์ประกอบด้วย มอเตอร์ขับเคลื่อน 6 ตัว ที่มีจุดเด่นของระบบควบคุมอัตโนมัติสำหรับการเคลื่อนที่ใต้น้ำ 4 องศาอิสระ ผลการวิจัยแสดงให้เห็นว่า ระบบระบุตำแหน่งของ ROUV สามารถระบุตำแหน่งและทิศ Yaw แบบเวลาจริงได้อย่างแม่นยำ ระบบควบคุมเสถียรภาพสามารถทำให้ ROUV รักษาตำแหน่งใน 3 มิติและทิศ Yaw พร้อมกันเมื่ออยู่นิ่ง และรักษาเส้นทางการเคลื่อนที่ตามที่ต้องการได้

คำสำคัญ: หุ่นยนต์ใต้น้ำ, ตัวควบคุม PID, ระบบควบคุม, ระบบระบุตำแหน่ง

Thesis Title	Control System and Localization for the 4 DOF Remotely Operated Underwater Vehicle
Author	Mr. Paramet Suwanwong
Major Program	Mechanical Engineering
Academic Year	2018

ABSTRACT

Nowadays remotely operated underwater vehicles (ROUV) become necessary equipment for many underwater tasks. However, there are many underwater dynamic-motion characteristics of ROUV, e.g., motion drifting, dynamics of the ROUV thruster, balancing between ROUV weight and buoyancy force, uncertain force from cable wire and underwater field-of-view limitation, that cause difficulty for an operator to control an ROUV even without the water current. In this research, PID stabilization control system is applied to the ROUV and localization system is developed based on Hector SLAM which is a software package of Robot Operating System (ROS). The PI stabilization controller relies on the horizontal velocity, depth and yaw-angular velocity feedbacks to control the robot motion in the corresponding directions. The ROUV hardware consists of a computer laptop, an Arduino board and a Raspberry Pi board as processor unit, thrusters, motor drive boards, a pressure sensor for depth measurement, a gyroscope and magnetometer of IMU for orientation measurement, Lidar sensor for measuring horizontal distance and determining robot position by using Scan Matching algorithm. The experiments are performed on the developed underwater robot. The ROUV, consisted of 6 thrusters, has an automatic feedback-control system for 4 degrees of freedom motion, which is a main contribution of this research. The research results show that the localization system of the ROUV is able to precisely maintain real-time position and yaw orientation. The controlled system is able to maintain the ROUV at the 3D stationary target position and to maneuver along the desired path.

Keywords: ROUV, PID controller, Control system, localization system.

กิตติกรรมประกาศ

สำหรับความสำเร็จของงานวิจัยฉบับนี้ ผู้วิจัยขอขอบพระคุณอาจารย์ที่ปรึกษางานวิจัย รศ. ดร. พุทธิกร สมิตไผตรี สำหรับคำแนะนำ แนวคิด ตลอดจนการสละเวลาตรวจสอบแก้ไข ข้อบกพร่องต่างๆ ทำให้งานวิจัยฉบับนี้เสร็จสมบูรณ์ได้

ขอขอบพระคุณคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ และทีมวิจัย Smart Mechatronics สำหรับความช่วยเหลือในด้านการเงินและสถานที่สำหรับการทำงานวิจัย

ขอขอบพระคุณบิดา-มารดา ญาติพี่น้องและเพื่อนๆ ที่คอยมอบความรัก สนับสนุนและให้กำลังใจตลอดช่วงเวลาการทำงานวิจัย

ปรเมศวร์ สุวรรณวงศ์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	(5)
บทคัดย่อภาษาอังกฤษ	(6)
กิตติกรรมประกาศ	(7)
สารบัญ	(8)
รายการตาราง	(10)
รายการภาพประกอบ	(11)
สัญลักษณ์คำย่อและตัวย่อ	(14)
บทที่	
1. บทนำ	1
1.1 ที่มาและความสำคัญของงานวิจัย	1
1.2 การตรวจเอกสาร	1
1.3 วัตถุประสงค์ของงานวิจัย	9
1.4 ขอบเขตการวิจัย	9
1.5 ประโยชน์ที่คาดว่าจะได้รับ	9
2. วิธีการวิจัย	10
2.1 ส่วนโครงสร้างของหุ่นยนต์ดำน้ำ	10
2.2 ส่วนโปรแกรมของระบบ	17
2.3 อุปกรณ์อิเล็กทรอนิกส์	19
3. การทำงานของระบบ	30
3.1 ระบบรักษาตำแหน่ง	30
3.2 ระบบระบุตำแหน่ง	35
3.3 ส่วนแสดงผล	35
4. การทดลอง	38
4.1 การรักษาตำแหน่ง	39
4.2 การรักษาเส้นทางการเคลื่อนที่ในระนาบ X-Y	41
4.3 การควบคุมระดับความลึก	46
4.4 การสร้างแผนที่ใต้น้ำ	49
5. สรุปผลการทดลอง	51
5.1 สรุปผลการทดลอง	51
5.2 ปัญหาที่พบในงานวิจัย	52

สารบัญ (ต่อ)

บทที่	หน้า
5.3 แนวทางแก้ไข	52
5.4 ข้อเสนอแนะและงานในอนาคต	52
บรรณานุกรม	54
ภาคผนวก	56
ก. ข้อมูลจำเพาะของอุปกรณ์ที่ใช้ในการทดลอง	57

รายการตาราง

ตารางที่		หน้า
2.1	น้ำหนักของชิ้นส่วน ROUV	15
4.1	ตารางแสดงผลการทดสอบรักษาตำแหน่งที่ตำแหน่งหมายเลข 1	40
4.2	ตารางแสดงผลการทดสอบรักษาเส้นทางการเคลื่อนที่ตามแนวแกน Y	42
4.3	ตารางแสดงผลการทดสอบรักษาเส้นทางการเคลื่อนที่ตามแนวแกน X	44
4.4	ตารางแสดงผลการทดสอบรักษาทิศทางในการเคลื่อนที่ตามแนวแกน X	46
4.5	ตารางแสดงผลการทดสอบรักษาเส้นทางการเคลื่อนที่ตามแนวแกน Y	48
4.6	ตารางแสดงค่า gain และตัวปรับค่าที่ใช้ในการทดลอง	48
ก.1	ข้อมูลจำเพาะของบอร์ด Raspberry Pi 2 model B	57
ก.2	ข้อมูลจำเพาะของบอร์ด Arduino Mega 2560	57
ก.3	ข้อมูลจำเพาะของบอร์ดขับเคลื่อน Smile EVO24V9.3	58
ก.4	ข้อมูลจำเพาะของ Raspberry Pi Camera Module	58
ก.5	ข้อมูลจำเพาะของเซนเซอร์วัดความดัน MPX5700GP	59
ก.6	ข้อมูลจำเพาะของเซนเซอร์ IMU GY-87	59
ก.7	ข้อมูลจำเพาะของปั้มน้ำ Seaflo bilge pump 1100GPH	60
ก.8	ข้อมูลจำเพาะของเซนเซอร์ HOKUYO URG-04LX-UG01	60

รายการภาพประกอบ

รูปที่		หน้า
1.1	หุ่นยนต์ดำน้ำ ROVFURG-II	2
1.2	หุ่นยนต์ดำน้ำที่สุรศักดิ์และปรัชญาออกแบบ	2
1.3	หุ่นยนต์ดำน้ำ COMET-ROV	3
1.4	หุ่นยนต์ใต้น้ำ VideoRay PRO พร้อมระบบ LVS	3
1.5	หุ่นยนต์ดำน้ำที่ H.-W. Hsieh และคณะออกแบบ (ซ้าย) และสถานที่ทดสอบระบบ (ขวา)	4
1.6	ผลการทดสอบหุ่นยนต์ของ H.-W. Hsieh และคณะ	4
1.7	หุ่นยนต์ใต้น้ำอัตโนมัติ Tri-Dog 1	5
1.8	ผลจากการทดสอบระบบสร้างแผนที่ด้วยข้อมูลจาก sonar	5
1.9	หุ่นยนต์ใต้น้ำ VideoRay Pro 3 ติดตั้งโซนาร์ SeaSprite (ซ้าย) และแผนที่ของอุโมงค์ใต้น้ำที่ใช้ทดสอบ (ขวา)	6
1.10	หุ่นยนต์ใต้น้ำ VideoRay Pro 4 ติดตั้งเซนเซอร์ DVL	6
1.11	ตัวอย่างข้อมูลภาพจากโซนาร์	7
1.12	เซนเซอร์ต้นแบบที่ C. Cain และ A. Leonessa ออกแบบและพัฒนาขึ้น	7
2.1	ลักษณะโครงสร้างของหุ่นยนต์ดำน้ำภายนอก (ซ้าย) และภายใน (ขวา)	11
2.2	กรอบอ้างอิงทิศทางของหุ่นยนต์ดำน้ำ	12
2.3	ขนาดและตำแหน่งอุปกรณ์ของ ROUV จากมุมมองด้านบน	12
2.4	ขนาดและตำแหน่งอุปกรณ์ของ ROUV จากมุมมองด้านข้าง	13
2.5	ตำแหน่งของ thruster 3 และ 4 เทียบกับโครงสร้าง	13
2.6	ขนาดและตำแหน่งอุปกรณ์ของ ROUV จากมุมมองด้านหน้า	14
2.7	แนวจุดศูนย์ถ่วงในระนาบ (Y_r , Z_r)	15
2.8	แนวจุดศูนย์ถ่วงในระนาบ ($-X_r$, Z_r)	16
2.9	แนวจุดศูนย์ถ่วงในระนาบ (Z_r , X_r)	16
2.10	แผนผังการทำงานของระบบ ROS	17
2.11	บอร์ด Raspberry Pi 2 model B	20
2.12	บอร์ด Arduino Mega 2560	20
2.13	บอร์ดขับเคลื่อนมอเตอร์ Smile EVO24V9.3	21
2.14	Raspberry Pi Camera Module	21
2.15	เซนเซอร์วัดความดัน MPX5700GP	22
2.16	การทดสอบเปรียบเทียบเซนเซอร์วัดความดัน	23

รายการภาพประกอบ (ต่อ)

รูปที่		หน้า
2.17	ความสัมพันธ์ระหว่างข้อมูลความสูงของน้ำกับสัญญาณจากเซนเซอร์	23
2.18	เซนเซอร์ IMU GY-87	25
2.19	ปั้มน้ำ Seaflo bilge pump 1100GPH 12V ก่อนดัดแปลง (ซ้าย) และหลังดัดแปลง (ขวา)	26
2.20	เซนเซอร์ HOKUYO URG-04LX-UG01 ก่อนดัดแปลง (ซ้าย) และหลังดัดแปลง (ขวา)	27
2.21	แผนภาพการเชื่อมต่ออุปกรณ์ของหุ่นยนต์ดำน้ำและ GCS	28
2.22	Schematic diagram ของระบบอิเล็กทรอนิกส์	29
3.1	กรอบอ้างอิงตริง [X, Y, Z] เทียบกับกรอบอ้างอิงของหุ่นยนต์ [X_r , Y_r , Z_r]	31
3.2	แผนภาพการทำงานของระบบควบคุมตำแหน่งหุ่นยนต์ใต้น้ำแบบสองศาอิสระ	34
3.3	สัญญาณควบคุมมอเตอร์ขับเคลื่อนหุ่นยนต์ใต้น้ำ	34
3.4	การแสดงผลข้อมูลจากโปรแกรม RQT	36
3.5	การแสดงผลข้อมูลจากโปรแกรม RVIZ	36
3.6	ภาพถ่ายที่ได้จากหุ่นยนต์ใต้น้ำที่พัฒนาขึ้น	37
4.1	การรักษาตำแหน่งของหุ่นยนต์ขณะทดสอบกับผิวน้ำ	38
4.2	ตำแหน่งในการทดสอบหุ่นยนต์และกรอบอ้างอิงเทียบกับผิวน้ำ	39
4.3	สถานะตำแหน่งของหุ่นยนต์ขณะทดสอบการรักษาตำแหน่งที่ตำแหน่งหมายเลข 1	40
4.4	สถานะทิศ yaw ของหุ่นยนต์ขณะทดสอบการรักษาตำแหน่งที่ตำแหน่งหมายเลข 1	40
4.5	สถานะตำแหน่งของหุ่นยนต์ขณะรักษาเส้นทางการเคลื่อนที่ตามแนวแกน Y	41
4.6	สถานะทิศ yaw ของหุ่นยนต์ขณะรักษาเส้นทางการเคลื่อนที่ตามแนวแกน Y	41
4.7	สถานะความเร็วแนวแกน Y ของหุ่นยนต์ขณะรักษาเส้นทางการเคลื่อนที่ตามแนวแกน Y	42
4.8	สถานะตำแหน่งของหุ่นยนต์ขณะรักษาเส้นทางการเคลื่อนที่ตามแนวแกน X	43
4.9	สถานะทิศ yaw ของหุ่นยนต์ขณะรักษาเส้นทางการเคลื่อนที่ตามแนวแกน X	44
4.10	สถานะความเร็วแนวแกน X ของหุ่นยนต์ขณะรักษาเส้นทางการเคลื่อนที่ตามแนวแกน X	44
4.11	สถานะทิศ yaw ของหุ่นยนต์ขณะรักษาทิศทางการเคลื่อนที่ตามแนวแกน X	45
4.12	สถานะความเร็วเชิงมุมทิศ yaw ของหุ่นยนต์ขณะรักษาทิศทางการเคลื่อนที่ตามแนวแกน X	45

รายการภาพประกอบ (ต่อ)

รูปที่		หน้า
4.13	สถานะตำแหน่งของหุ่นยนต์ขณะรักษาเส้นทางการเคลื่อนที่ตามแนว Depth	47
4.14	สถานะทิศ yaw ของหุ่นยนต์ขณะรักษาเส้นทางการเคลื่อนที่ตามแนว Depth	47
4.15	สถานะความเร็วแนว Depth ของหุ่นยนต์ขณะรักษาเส้นทางการเคลื่อนที่ตามแนว Depth	47
4.16	เส้นทางการเคลื่อนที่แบบ 3 มิติ จากการทดลองทั้งหมด	48
4.17	แผนที่สิ่งแวดล้อมและเส้นทางการเคลื่อนที่ของหุ่นยนต์ที่ได้จากการทดสอบ	49

สัญลักษณ์คำย่อและตัวย่อ

CCD	Charge-Coupled Device
IMU	Inertial Measurement Unit
GCS	Ground Control Station
Lidar	Light detection and ranging
ROUV	Remotely Operated Underwater Vehicle
LVS	Laser-based Vision System
GPS	Global Positioning System
SONAR	Sound Navigation Ranging
DVL	Doppler Velocity Log
ROS	Robot Operating System
PWM	Pulse-Width Modulation
avg_{scale}	ค่าอัตราส่วนสนามแม่เหล็กเฉลี่ย
\tilde{e}	เวกเตอร์สถานะความคลาดเคลื่อนของความเร็วอ้างอิงเทียบกับกรอบอ้างอิง จริง $[e_x, e_y, e_z, e_\omega]^T$
\tilde{e}_r	เวกเตอร์สถานะความคลาดเคลื่อนของความเร็วอ้างอิงเทียบกับกรอบอ้างอิง ของหุ่นยนต์ $[e_{xr}, e_{yr}, e_{zr}, e_{\omega r}]^T$
\tilde{K}	ตัวปรับค่าสัดส่วน $[K_x, K_y, K_z, K_\omega]^T$
\tilde{K}_p	proportional gain $[K_{px}, K_{py}, K_{pz}, K_{p\omega}]^T$
\tilde{K}_i	integral gain $[K_{ix}, K_{iy}, K_{iz}, K_{i\omega}]^T$
L_1	ระยะห่างระหว่างจุดกึ่งกลางหุ่นยนต์กับตำแหน่งของมอเตอร์ 1 และ 2
L_2	ระยะห่างระหว่างจุดกึ่งกลางหุ่นยนต์กับตำแหน่งของมอเตอร์ 3 และ 4
\tilde{M}	เวกเตอร์ค่าสนามแม่เหล็กจากเซนเซอร์ $[M_x, M_y, M_z]^T$
$\tilde{M}_{calibrated}$	เวกเตอร์ค่าสนามแม่เหล็กที่ได้จากการปรับเทียบ $[M_{xcalibrated}, M_{ycalibrated}, M_{zcalibrated}]^T$
\tilde{M}_{max}	เวกเตอร์ค่าสนามแม่เหล็กสูงสุด $[M_{xmax}, M_{ymax}, M_{zmax}]^T$
\tilde{M}_{min}	เวกเตอร์ค่าสนามแม่เหล็กต่ำสุด $[M_{xmin}, M_{ymin}, M_{zmin}]^T$
\tilde{M}_{offset}	เวกเตอร์ค่าสนามแม่เหล็กชดเชย $[M_{xoffset}, M_{yoffset}, M_{zoffset}]^T$
\tilde{M}_{scale}	เวกเตอร์ค่าอัตราส่วนสนามแม่เหล็ก $[M_{xscale}, M_{yscale}, M_{zscale}]^T$

สัญลักษณ์คำย่อและตัวย่อ (ต่อ)

\tilde{t}	ความแตกต่างของเวลาระหว่างตำแหน่งปัจจุบันกับตำแหน่งก่อนปัจจุบัน $[t_x, t_y, t_z, t_\omega]^T$
T	คาบเวลาของข้อมูล
$\tilde{\mathbf{u}}(t)$	เวกเตอร์สัญญาณควบคุม $[\mathbf{u}_x, \mathbf{u}_y, \mathbf{u}_z, \mathbf{u}_\omega]^T$
$\tilde{\mathbf{u}}_m(t)$	สัญญาณควบคุมสำหรับมอเตอร์ตัวที่ 1 ถึงตัวที่ 6 $[\mathbf{u}_{m1}, \mathbf{u}_{m2}, \mathbf{u}_{m3}, \mathbf{u}_{m4}, \mathbf{u}_{m5}, \mathbf{u}_{m6}]^T$
$\tilde{\mathbf{V}}$	เวกเตอร์ความเร็ว $[\mathbf{V}_x, \mathbf{V}_y, \mathbf{V}_z, \boldsymbol{\omega}]^T$ เทียบกับ fixed frame
$\tilde{\mathbf{V}}_{current}$	เวกเตอร์สถานะความเร็วปัจจุบัน $[\mathbf{V}_{xcurrent}, \mathbf{V}_{ycurrent}, \mathbf{V}_{zcurrent}, \boldsymbol{\omega}_{current}]^T$
$\tilde{\mathbf{V}}_m$	เวกเตอร์ความเร็วของมอเตอร์ $[\mathbf{V}_{m1}, \mathbf{V}_{m2}, \mathbf{V}_{m3}, \mathbf{V}_{m4}, \mathbf{V}_{m5}, \mathbf{V}_{m6}]^T$
$\tilde{\mathbf{V}}_r$	เวกเตอร์ความเร็ว $[\mathbf{V}_{xr}, \mathbf{V}_{yr}, \mathbf{V}_{zr}, \boldsymbol{\omega}_r]^T$ เทียบกับ robot frame
$\tilde{\mathbf{V}}_{set}$	เวกเตอร์สถานะความเร็วที่ต้องการ $[\mathbf{V}_{xset}, \mathbf{V}_{yset}, \mathbf{V}_{zset}, \boldsymbol{\omega}_{set}]^T$
$\tilde{\mathbf{X}}_{current}$	เวกเตอร์สถานะตำแหน่งปัจจุบัน $[\mathbf{X}_{current}, \mathbf{Y}_{current}, \mathbf{Z}_{current}, \boldsymbol{\theta}_{current}]^T$
$\tilde{\mathbf{X}}_{previous}$	เวกเตอร์สถานะตำแหน่งก่อนปัจจุบัน $[\mathbf{X}_{previous}, \mathbf{Y}_{previous}, \mathbf{Z}_{previous}, \boldsymbol{\theta}_{previous}]^T$
$\tilde{\mathbf{X}}_{set}$	เวกเตอร์สถานะตำแหน่งเป้าหมาย $[\mathbf{X}_{set}, \mathbf{Y}_{set}, \mathbf{Z}_{set}, \boldsymbol{\theta}_{set}]^T$
$\boldsymbol{\theta}$	มุมของ robot frame เทียบกับ fixed frame รอบแกน Z
$\boldsymbol{\omega}$	ความเร็วเชิงมุมรอบแกน Z

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของงานวิจัย

Remotely Operated Underwater Vehicle (ROUV) หรือหุ่นยนต์ดำน้ำแบบบังคับ เป็นเทคโนโลยีการสำรวจใต้น้ำซึ่งอำนวยความสะดวกได้อย่างมาก เนื่องจากการสำรวจใต้น้ำเดิมอาศัยนักดำน้ำลงไปสำรวจ และจำเป็นต้องพึ่งอุปกรณ์ต่างๆ มากมาย ทำให้เกิดความยุ่งยาก สิ้นเปลือง อีกทั้งยังมีความเสี่ยงต่อสุขภาพนักดำน้ำอีกด้วย การใช้หุ่นยนต์ดำน้ำในการสำรวจสามารถทำได้ง่ายกว่า โดยไม่จำเป็นต้องมีนักดำน้ำ ทำงานได้นานและลึกกว่า ประหยัดกว่าในระยะยาว และลดความเสี่ยงต่อชีวิตของนักดำน้ำลงได้ แต่การควบคุมหรือใช้งานหุ่นยนต์ดำน้ำให้ทำงานได้อย่างเต็มที่ ต้องอาศัยผู้ใช้งานหรือผู้ควบคุมที่มีทักษะในการควบคุม อีกทั้งสภาพแวดล้อมหรือพื้นที่ทำการส่วนใหญ่หุ่นยนต์จะอยู่นอกสายตาผู้ใช้ ภาพที่ผู้ใช้ได้รับจากกล้องในตัวหุ่นยนต์มีข้อจำกัดเรื่องมุมมอง ทำให้ผู้ใช้ไม่สามารถรับรู้ตำแหน่งและทิศทางของหุ่นยนต์ได้ ระบบระบุตำแหน่งจึงเป็นสิ่งสำคัญที่จะช่วยให้ผู้ใช้สามารถรับรู้ตำแหน่งและทิศทางของหุ่นยนต์ดำน้ำได้ และสามารถควบคุมหุ่นยนต์ไปในทิศทางและตำแหน่งที่ต้องการได้อย่างแม่นยำ ปัจจัยที่ส่งผลกระทบต่อการควบคุมหุ่นยนต์อันเนื่องมาจากการเคลื่อนที่ใต้น้ำของหุ่นยนต์ดำน้ำ มีทั้งการไถลหลังจากการเคลื่อนที่ ความไม่สมดุลระหว่างน้ำหนักกับแรงลอยตัวที่ทำให้หุ่นยนต์ลอยขึ้นหรือจมลง และแรงกระทำจากความแข็งตึงของสายเคเบิลซึ่งกระทำต่อหุ่นยนต์ในรูปแบบที่ไม่สามารถคาดเดาได้ ทำให้การควบคุมหุ่นยนต์ให้อยู่กับที่โดยอาศัยเพียงการควบคุมโดยตรงจากผู้บังคับจึงเป็นงานที่ยาก ต้องใช้สมมติฐานสูงขณะควบคุม ทำให้เกิดความเหนื่อยล้า และประสิทธิภาพในการควบคุมลดลง ระบบรักษาตำแหน่งจึงเป็นส่วนสำคัญที่จะช่วยเพิ่มประสิทธิภาพการควบคุมหุ่นยนต์ และลดภาระของผู้บังคับลง

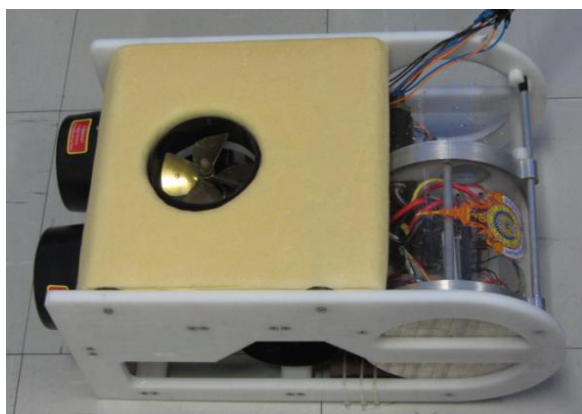
1.2 การตรวจเอกสาร

งานวิจัยที่ผ่านมามุ่งเน้นไปในด้านการพัฒนาขีดความสามารถของหุ่นยนต์ดำน้ำในหลายประเด็น ได้แก่ ด้านการพัฒนาระบบรักษาตำแหน่ง ตัวอย่างเช่น V. N. Kuhn, P. L. J. Drews, S. C. P. Gomes, M. A. B. Cunha และ S. S. da C. Botelho พัฒนาระบบรักษาระดับ ความลึก รักษาระยะห่างจากจุดอ้างอิง และรักษามุม yaw ของหุ่นยนต์ดำน้ำต้นแบบ ROVFURG-II โดยอาศัยข้อมูลจากกล้อง gyroscope และ pressure sensor ด้วยระบบควบคุมแบบ PID ลักษณะของหุ่นยนต์ดำน้ำ ROVFURG-II แสดงในรูปที่ 1.1 [1]



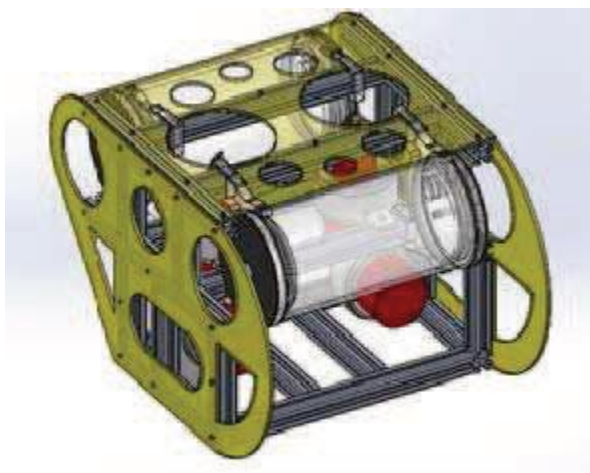
รูปที่ 1.1 หุ่นยนต์ดำน้ำ ROVFURG-II [1]

สุรศักดิ์ ทิมพิทักษ์ และปรัชญา เปรมปราณีรัชต์ ออกแบบหุ่นยนต์ดำน้ำที่ประกอบด้วย pressure sensor, accelerometer, gyroscope และกล้องวิดีโอ รวมทั้งพัฒนาระบบรักษาระดับความลึกของหุ่นยนต์ดำน้ำโดยการใช้ระบบควบคุม PID ที่ป้อนกลับด้วยค่าสัญญาณความลึก ลักษณะของหุ่นยนต์ดำน้ำที่สุรศักดิ์และปรัชญาออกแบบแสดงในรูปที่ 1.2 [2]



รูปที่ 1.2 หุ่นยนต์ดำน้ำที่สุรศักดิ์และปรัชญาออกแบบ [2]

A. Z. Abidin, R. Mardiyanto และ D. Purwanto พัฒนาระบบรักษาระดับความลึกของหุ่นยนต์ดำน้ำ COMET-ROV ซึ่งประกอบด้วยเซนเซอร์ IMU และ pressure sensor โดยใช้ระบบควบคุม PID ที่ป้อนกลับด้วยค่าสัญญาณความลึก ลักษณะของหุ่นยนต์ดำน้ำ COMET-ROV แสดงในรูปที่ 1.3 [3]



รูปที่ 1.3 หุ่นยนต์ดำน้ำ COMET-ROV [3]

G. C. Karras และ K. J. Kyriakopoulos พัฒนาระบบรักษาระดับความลึก ระยะห่างจากจุดเป้าหมายและมุม yaw เทียบกับตำแหน่งอ้างอิง โดยใช้หุ่นยนต์ดำน้ำ VideoRay PRO ที่ติดตั้งระบบ LVS ในการทดลอง ระบบรักษาระดับความลึกอาศัยข้อมูลจาก LVS ร่วมกับ IMU ในการประมวลผล ลักษณะของหุ่นยนต์ดำน้ำ VideoRay PRO แสดงในรูปที่ 1.4 [4]

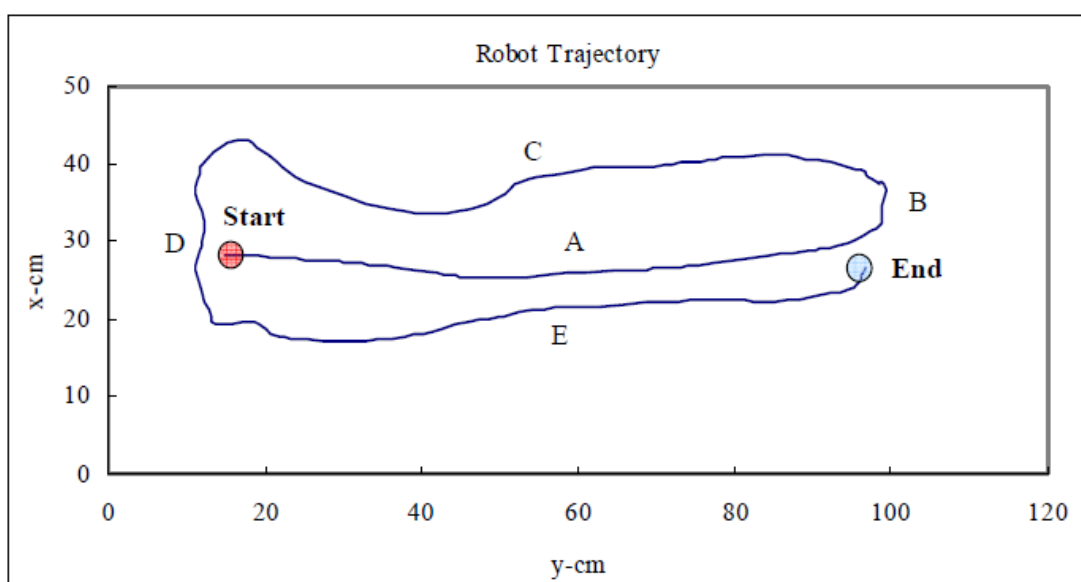


รูปที่ 1.4 หุ่นยนต์ใต้น้ำ VideoRay PRO พร้อมระบบ LVS [4]

H.-W. Hsieh, C.-L. Lee และ C.-L. Kuo พัฒนาระบบรักษาเส้นทางเคลื่อนที่ในระนาบ 2 มิติโดยใช้หุ่นยนต์ดำน้ำที่ออกแบบขึ้นเองเป็นตัวแทนทดสอบระบบ โดยระบบอาศัยข้อมูลจากเซนเซอร์ gyroscope, accelerometer และ ultrasound ในการประมวลผล ลักษณะของหุ่นยนต์ดำน้ำที่ H.-W. Hsieh และคณะออกแบบพร้อมทั้งสถานที่ทดสอบแสดงในรูปที่ 1.5 โดยผลตอบสนองเส้นทางเคลื่อนที่จากการทดสอบแสดงในรูปที่ 1.6 [5]

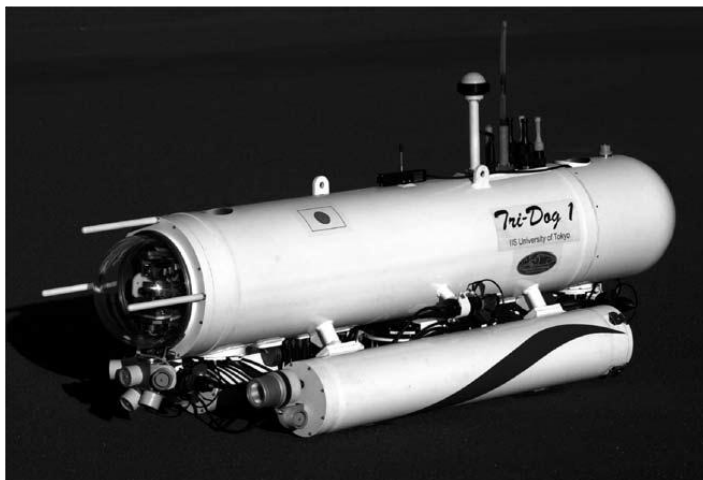


รูปที่ 1.5 หุ่นยนต์ดำน้ำที่ H.-W. Hsieh และคณะออกแบบ (ซ้าย) และสถานที่ทดสอบระบบ (ขวา) [5]



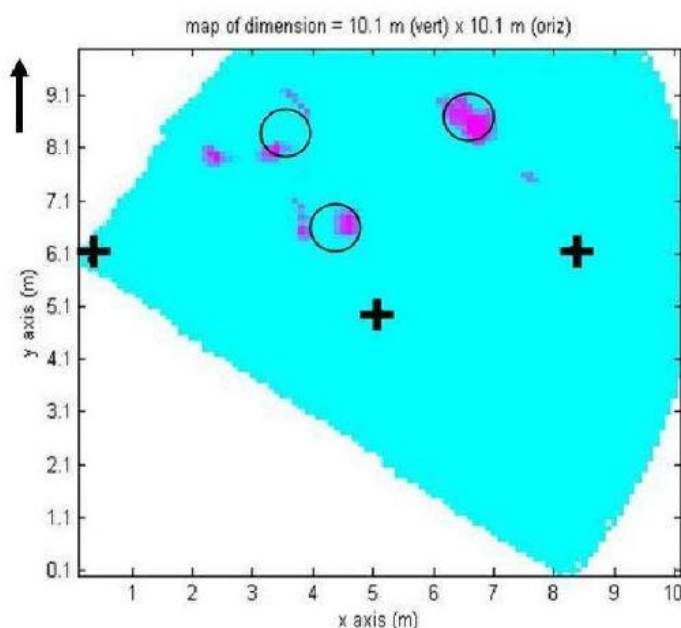
รูปที่ 1.6 ผลการทดสอบหุ่นยนต์ของ H.-W. Hsieh และคณะ [5]

อีกทั้งยังมีงานวิจัยเกี่ยวกับด้านการพัฒนาระบบนำร่องตัวอย่างเช่น H. Kondo และ T. Ura พัฒนาระบบนำร่องสำหรับการสำรวจโครงสร้างใต้น้ำอัตโนมัติของหุ่นยนต์ดำน้ำอัตโนมัติ Tri-Dog 1 โดยระบบอาศัยข้อมูลจาก doppler velocity log, electromagnetic flowmeter, pressure sensor (Depth sensor), real time kinematics GPS, acoustic ranging sensor, CCD camera with pan & tilt mechanism และ laser pointer ในการประมวลผล ลักษณะของหุ่นยนต์ดำน้ำ Tri-Dog 1 แสดงในรูปที่ 1.7 [6]



รูปที่ 1.7 หุ่นยนต์ใต้น้ำอัตโนมัติ Tri-Dog 1 [6]

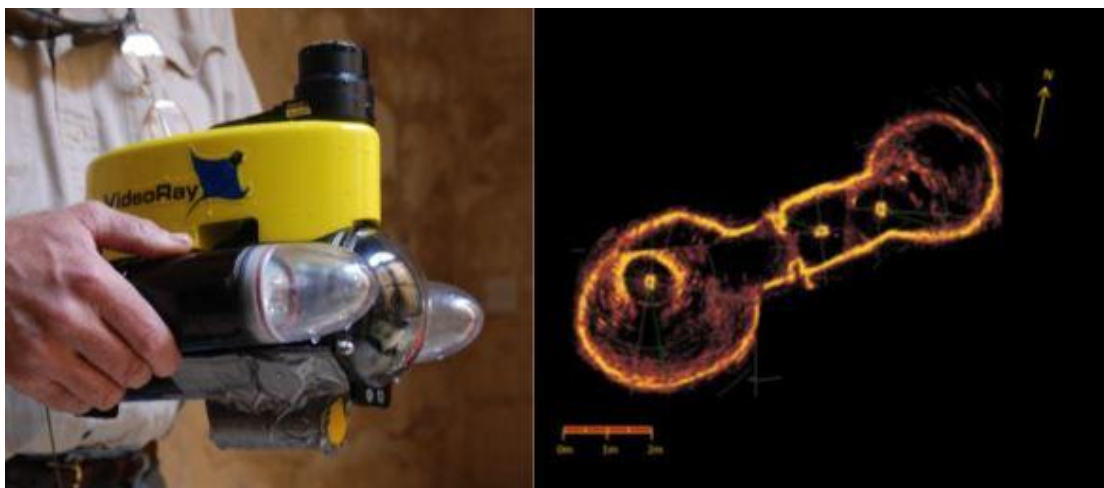
งานวิจัยที่เกี่ยวกับการพัฒนาระบบสร้างแผนที่ใต้น้ำ ตัวอย่างเช่น D. Scaradozzi, L. Gambella, S. M. Zanoli และ G. Conte พัฒนาระบบการทำแผนที่ใต้น้ำโดยระบบอาศัยข้อมูลจาก scanning sonar ในการประมวลผล ลักษณะของแผนที่ที่ได้จากการทดสอบระบบแสดงในรูปที่ 1.8 [7]



รูปที่ 1.8 ผลจากการทดสอบระบบสร้างแผนที่ด้วยข้อมูลจาก sonar [7]

C. M. Clark, C. S. Olstad, K. Buhagiar และ T. Gambin พัฒนาระบบการทำแผนที่ใต้น้ำโดยใช้หุ่นยนต์ VideoRay Pro 3 ที่ติดตั้งโซนาร์ SeaSprite เพิ่มเติมบนตัวหุ่นยนต์ในการทดสอบ ระบบอาศัยข้อมูลจาก โซนาร์, depth sensor และ compass ในการประมวลผล

ลักษณะของหุ่นยนต์ VideoRay Pro 3 และแผนที่ที่ได้จากการทดสอบระบบแสดงในรูปที่ 1.9 [8]



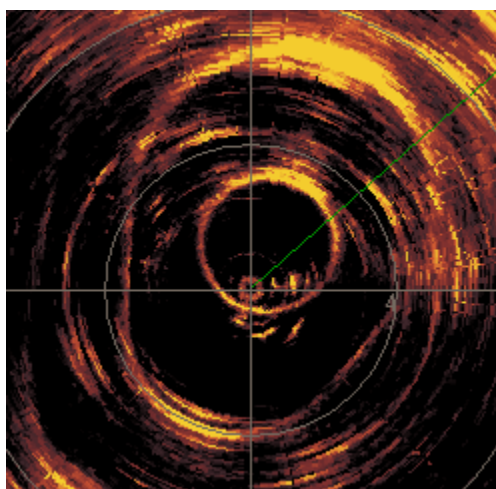
รูปที่ 1.9 หุ่นยนต์ดำน้ำ VideoRay Pro 3 ติดตั้งโซนาร์ SeaSprite (ซ้าย) และแผนที่ของอุโมงค์ใต้น้ำที่ใช้ทดสอบ (ขวา) [8]

งานวิจัยด้านเซนเซอร์สำหรับใช้งานใต้น้ำ เช่น J. Snyder ทดสอบและเปรียบเทียบข้อมูลตำแหน่งของหุ่นยนต์ใต้น้ำ ที่ได้จากการประมวลผลระหว่างเซนเซอร์ DVL และ GPS โดยใช้หุ่นยนต์ดำน้ำ VideoRay Pro 4 ในการทดสอบ ลักษณะของหุ่นยนต์ดำน้ำ VideoRay Pro 4 ติดตั้งเซนเซอร์ DVL แสดงในรูปที่ 1.10 [9]



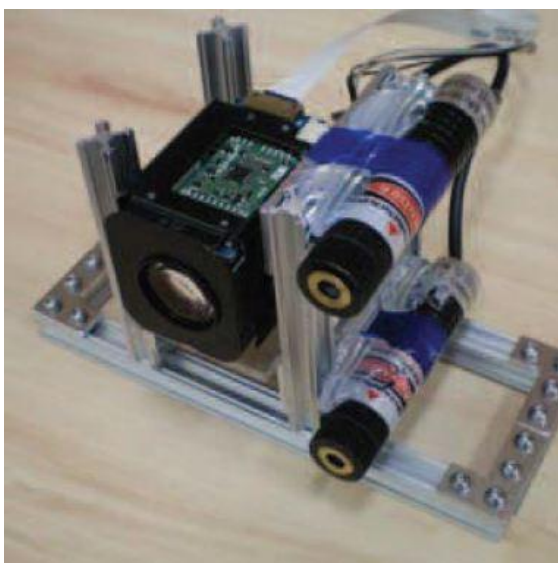
รูปที่ 1.10 หุ่นยนต์ดำน้ำ VideoRay Pro 4 ติดตั้งเซนเซอร์ DVL [9]

Z. Eskinja, Z. Fabekovic, และ Z. Vukic ศึกษาการใช้ข้อมูลจาก scanning sonar และ digital compass ในการระบุตำแหน่งของหุ่นยนต์ใต้น้ำโดยใช้หุ่นยนต์ใต้น้ำ VideoRay Pro ในการทดสอบ ตัวอย่างข้อมูลภาพจากโซนาร์แสดงในรูปที่ 1.11 [10]



รูปที่ 1.11 ตัวอย่างข้อมูลภาพจากโซนาร์ [10]

C. Cain และ A. Leonessa ทำการศึกษาและออกแบบเซนเซอร์ต้นแบบที่พัฒนาขึ้นเพื่อใช้ในการทำแผนที่ใต้น้ำซึ่งตัวเซนเซอร์ประกอบด้วย laser line generators 2 ตัว และ CCD camera ลักษณะของเซนเซอร์ต้นแบบที่ C. Cain และ A. Leonessa ออกแบบและพัฒนาขึ้นแสดงในรูปที่ 1.12 [11]



รูปที่ 1.12 เซนเซอร์ต้นแบบที่ C. Cain และ A. Leonessa ออกแบบและพัฒนาขึ้น [11]

รวมไปถึงงานวิจัยด้านระบบควบคุมการทำงานสำหรับหุ่นยนต์ใต้น้ำเช่น F. A. Azis, M. S. M. Aras, M. Z. A. Rashid, M. N. Othman และ S. S. Abdullah ทำการศึกษาประเด็นปัญหาที่พบจากการทำวิจัยเกี่ยวกับหุ่นยนต์ใต้น้ำ เช่น ผลกระทบจากรูปแบบโครงสร้างและอุปกรณ์ที่เลือกใช้ ผลกระทบจากสิ่งแวดล้อมต่อการควบคุมหุ่นยนต์ ปัญหาการสื่อสารระหว่างผู้ใช้และหุ่นยนต์ รวมไปถึงเปรียบเทียบข้อดีข้อด้อยระหว่างการใช้ระบบควบคุมแบบ PID, sliding-mode, fuzzy และ neural network [12]

W. M. Bessa, M. S. Dutra และ E. Kreuzer พัฒนาระบบรักษาระดับความลึกของหุ่นยนต์ใต้น้ำโดยใช้ระบบควบคุมแบบ adaptive fuzzy sliding mode และทดสอบด้วยวิธี numerical simulation [13] รวมทั้งศึกษาการควบคุมตำแหน่งของ ROUV โดยการควบคุมความเร็วการหมุนของใบพัดด้วยระบบควบคุมแบบ fuzzy sliding mode [14]

T. I. Fossen และ S. I. Sagatun ศึกษาวิธีการลด overshoot ของระบบควบคุมแบบ PID ด้วย continuous input smoother และ discrete fuzzy smoother และทดสอบด้วยวิธี simulation [15]

ผลจากการตรวจเอกสารพบว่า ในประเด็นของระบบระบุตำแหน่งและควบคุมตำแหน่งของหุ่นยนต์ดำน้ำขนาดเล็กนั้น งานวิจัยที่ผ่านมาส่วนมากศึกษาและวิจัยระบบควบคุมตำแหน่งของหุ่นยนต์ดำน้ำชั้นสำรวจ (Observation-class) ที่สามารถควบคุมตำแหน่งได้พร้อมกันสูงสุดเพียง 3 องศาอิสระเท่านั้น ทำให้หุ่นยนต์รักษาตำแหน่งและทิศทางได้ไม่สมบูรณ์ หุ่นยนต์ไม่สามารถควบคุมความคลาดเคลื่อนในองศาอิสระอื่น เช่น ทิศทาง (Heading orientation) ที่ไม่ได้ถูกกำหนดไว้ในระบบควบคุม ซึ่งเป็นผลมาจากการที่ระบบระบุตำแหน่งไม่สามารถระบุตำแหน่งได้มากกว่า 3 องศาอิสระ หรือหุ่นยนต์ดำน้ำประเภทที่ไม่มี thruster ในบางองศาอิสระ ทำให้การพัฒนาความสามารถของหุ่นยนต์ในการเคลื่อนที่อัตโนมัติหรือรักษาเส้นทางการเคลื่อนที่ทำได้ยากและมีความคลาดเคลื่อนสูง รวมไปถึงงานวิจัยนี้ต้องการจำลองการทำงานของหุ่นยนต์ดำน้ำให้ทำการสำรวจผนังซึ่งเป็นการทำงานในพื้นที่แคบ ด้วยเหตุนี้งานวิจัยนี้จึงเกิดขึ้นและมุ่งเน้นเพื่อพัฒนาระบบระบุตำแหน่งและควบคุมตำแหน่งของ ROUV ชั้นปฏิบัติการ (Work-class) ที่มีความสามารถในการระบุตำแหน่งของ ROUV เทียบกับสิ่งแวดล้อมและควบคุมตำแหน่งได้ใน 4 องศาอิสระ ทำให้มีความคล่องตัวสูงเหมาะกับการทำงานในที่แคบกว่า ROUV ชั้นสำรวจ โดยทดลองกับหุ่นยนต์ดำน้ำที่สร้างขึ้นเองโดยอาศัยการควบคุมความเร็วด้วยระบบควบคุม PI ที่ปรับสถานะของหุ่นยนต์จากข้อมูลของ pressure sensor, gyroscope, magnetometer และ Lidar

1.3 วัตถุประสงค์ของงานวิจัย

1. เพื่อออกแบบและสร้างโครงสร้างของหุ่นยนต์ให้มีขนาดเล็กและแข็งแรง
2. เพื่อออกแบบและพัฒนาระบบควบคุมให้สามารถระบุพิกัด และควบคุมการทรงตัวของหุ่นยนต์ดำน้ำได้
3. เพื่อพัฒนาส่วนแสดงผลค่าตำแหน่งและภาพถ่ายจากหุ่นยนต์ดำน้ำ

1.4 ขอบเขตการวิจัย

1. ส่วนโครงสร้างจะต้องทนความดันน้ำได้ที่ระดับความลึก 70 เซนติเมตร เพื่อสาคิการทำงานองระบบควบคุมที่พัฒนาขึ้น
2. ระบบควบคุมมีระบบระบุพิกัดและระบบควบคุมการทรงตัวที่มีความแม่นยำสามารถระบุพิกัดและทรงตัวหนึ่งในสระน้ำที่มีขอบเขตปิดได้ ความคลาดเคลื่อนอยู่ระหว่าง -5 เซนติเมตร ถึง 5 เซนติเมตรในแนวแกน X, Y, Z และอยู่ระหว่าง -15 องศา ถึง 15 องศา ตามมุม yaw
3. ส่วนแสดงผลสามารถแสดงผลค่าตำแหน่งและภาพถ่ายจากหุ่นยนต์ดำน้ำได้

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้หุ่นยนต์ดำน้ำต้นแบบซึ่งสาธิตความเป็นไปได้ของการควบคุมแบบสื่อองศาอิสระ
2. ได้ระบบควบคุมและแสดงผลการทำงานองหุ่นยนต์ดำน้ำ ทำให้ผู้ใช้ควบคุมหุ่นยนต์ได้ง่ายขึ้น
3. ได้องค์ความรู้ที่เกี่ยวข้องกับระบบระบุตำแหน่งใน 3 มิติ ซึ่งสามารถนำไปประยุกต์ใช้กับงานอื่นๆ ได้

บทที่ 2

วิธีการวิจัย

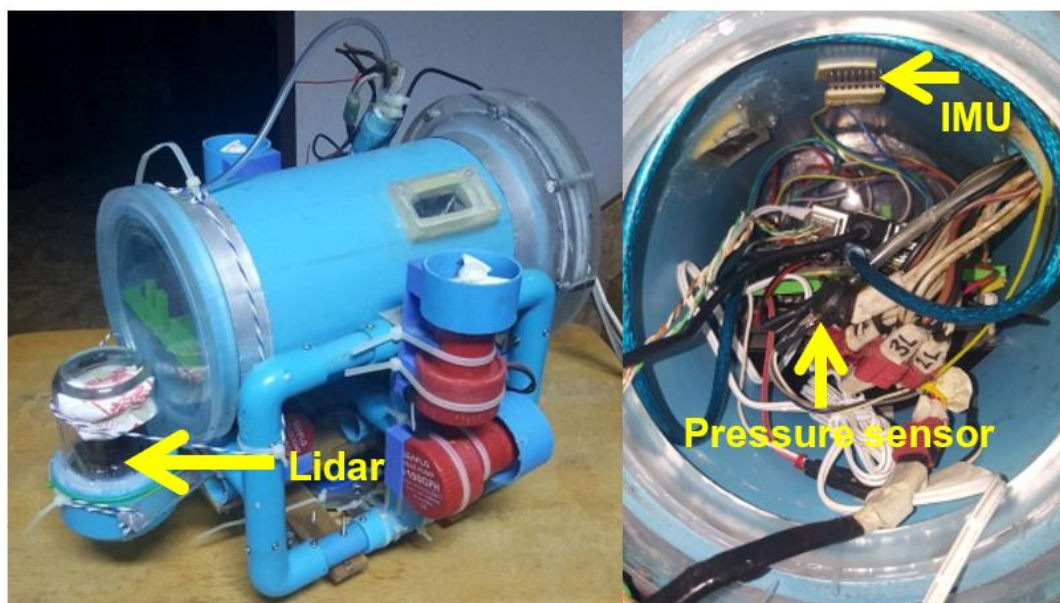
การวิจัยเริ่มต้นจากการศึกษาเอกสารที่เกี่ยวข้องกับงานวิจัย เพื่อหาข้อมูลที่เป็นประโยชน์ซึ่งสามารถนำมาเป็นแนวทางหรือนำมาประยุกต์ใช้ในงานวิจัยนี้ได้ รวมทั้งศึกษาข้อดีข้อด้อยเพื่อหาช่องว่างที่จะนำมาเป็นหัวข้อหรือประเด็นสำคัญสำหรับงานวิจัยนี้ได้ ต่อมาทำการออกแบบและสร้างหุ่นยนต์ดำน้ำซึ่งประกอบด้วย 3 ส่วนสำคัญ ได้แก่ ส่วนโครงสร้าง ส่วนโปรแกรม และส่วนอุปกรณ์อิเล็กทรอนิกส์โดยเป้าหมายคือ การพัฒนาให้ได้หุ่นยนต์ดำน้ำที่มีประสิทธิภาพเพียงพอสำหรับรองรับการติดตั้งระบบการควบคุมและสามารถทำงานตามคำสั่งของระบบควบคุมได้ โดยความสามารถหลักที่หุ่นยนต์ต้องมีคือ ความสามารถในการควบคุมการเคลื่อนที่ได้ 4 องศาอิสระ ได้แก่ surge, sway, heave และ yaw เมื่อหุ่นยนต์เสร็จสมบูรณ์จึงไปพัฒนาระบบการควบคุมโดยการประยุกต์ใช้โปรแกรมต่างๆ มาทำงานร่วมกัน รวมทั้งพัฒนาอัลกอริทึมการทำงานให้ระบบควบคุมสามารถทำงานได้ดีขึ้น เมื่อได้หุ่นยนต์และระบบควบคุมที่สามารถทำงานได้ในระดับหนึ่ง จึงทำการทดสอบการทำงานเพื่อหาจุดบกพร่องแล้วทำการแก้ไขปรับปรุง จนกระทั่งระบบมีประสิทธิภาพสูงขึ้น มีความสามารถเพียงพอในการควบคุมหุ่นยนต์ได้ตามขอบเขตของงานวิจัยที่วางไว้ แล้วจึงทำการทดสอบเพื่อบันทึกผลและวิจารณ์ผลการทดลอง

2.1 ส่วนโครงสร้างของหุ่นยนต์ดำน้ำ

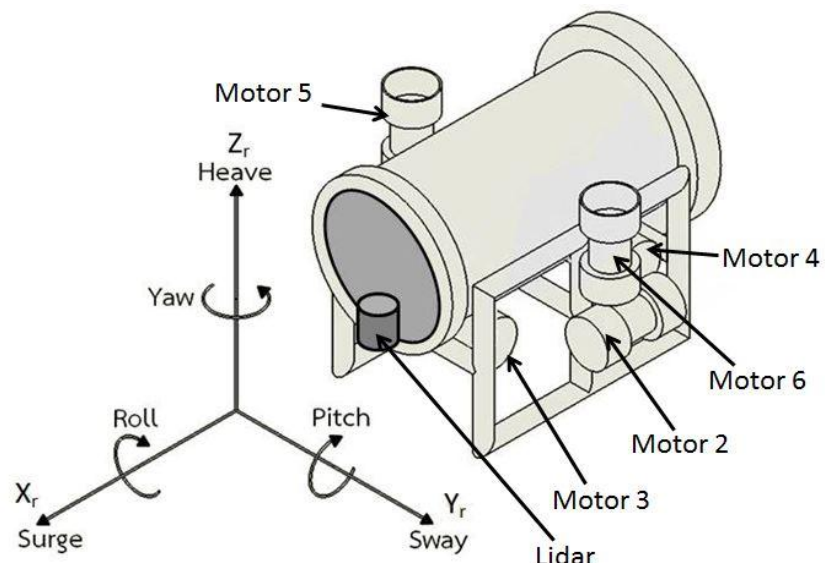
หุ่นยนต์ดำน้ำที่ใช้ในงานวิจัยนี้ถูกออกแบบและสร้างขึ้นเองเพื่อการทดสอบระบบควบคุม ลักษณะของหุ่นยนต์ดังแสดงในรูปที่ 2.1 โครงสร้างของหุ่นยนต์ทำจากท่อ PVC เป็นวัสดุหลัก เพราะมีความแข็งแรงทนทานต่อความดันใต้น้ำ ส่วนลำตัวทำจากท่อ PVC ขนาด 6 นิ้ว (nominal size) สำหรับบรรจุอุปกรณ์อิเล็กทรอนิกส์ ส่วนหัวปิดและซีลด้วยแผ่นอะคริลิกกลมที่ถูกเจาะร่องให้ประกบกับท่อได้พอดี ส่วนท้ายประกอบด้วยแผ่นอะคริลิก 2 แผ่นโดยแผ่นแรกที่อยู่ติดกับส่วนลำตัวถูกเจาะร่องให้สามารถประกบกับท่อได้พอดีเช่นเดียวกับส่วนด้านหน้า แต่แผ่นส่วนท้ายถูกเจาะทะลุบริเวณตรงกลางให้มีขนาดเล็กกว่าความกว้างภายในของท่อเล็กน้อย ส่วนขอบที่เหลือของแผ่นอะคริลิกถูกเจาะเป็นร่องสำหรับยางซีลกันน้ำ แผ่นที่สองถูกเจาะเป็นรูสำหรับสอดสายไฟและสายเคเบิลเข้าไปยังอุปกรณ์ภายในลำตัวได้และถูกซีลกันน้ำด้วยกาวอะคริลิกทั้ง 2 แผ่นด้านหลังถูกเจาะรูทะลุ 8 รูตามแนวเส้นรอบวงเป็นรูสำหรับสกรูยึดอะคริลิกทั้ง 2 แผ่นเข้าด้วยกันโดยเมื่ออะคริลิกทั้ง 2 แผ่นถูกยึดเข้าด้วยกัน ซีลยางจะทำหน้าที่กันน้ำที่ซึมเข้ามาระหว่างช่องว่างของอะคริลิกทั้งสองแผ่นไม่ให้รั่วเข้าไปในลำตัว ส่วนหลังนี้เป็นส่วนที่

สามารถถอดประกอบได้ซึ่งใช้เป็นช่องทางสำหรับใส่อุปกรณ์อิเล็กทรอนิกส์เข้าไปในลำตัวหรือถอดอุปกรณ์ดังกล่าวออกมา

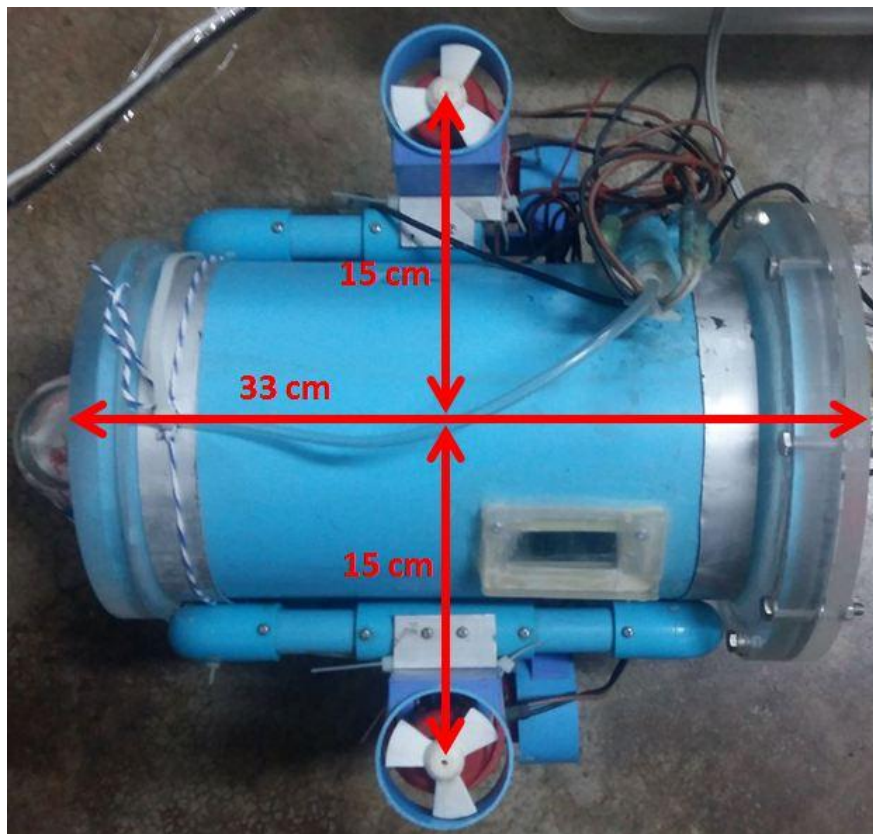
ส่วนโครงสร้างรอบนอกที่ยึดติดกับลำตัวใช้ท่อ PVC และข้อต่อขนาด 0.5 นิ้วเป็นวัสดุหลัก เพื่อเป็นแท่นติดตั้ง thrusters และเป็นขาตั้งบนพื้น อีกทั้งยังใช้เป็นส่วนปรับตั้งน้ำหนักถ่วงสมดุล หุ่นยนต์นี้มีมอเตอร์ขับเคลื่อน หรือ thruster ทั้งหมด 6 ตัว ทำหน้าที่ควบคุมการเคลื่อนที่ในทิศ surge, sway และ heave ทิศละสองตัว โดยมอเตอร์ที่ควบคุมการเคลื่อนที่ในทิศทาง surge และ sway ทั้งสี่ตัวสามารถใช้ควบคุมการหมุนในทิศมุม yaw ได้ด้วยดังแสดงในรูปที่ 2.2 ส่วนควบคุมภาคพื้นดิน (ground control station, GCS) ประกอบด้วยหม้อแปลงแหล่งจ่ายพลังงาน คอมพิวเตอร์แล็ปท็อปเป็นส่วนประมวลผลและแสดงผลการทำงาน มีจอยสติ๊กสำหรับบังคับหุ่นยนต์ หุ่นยนต์เชื่อมต่อกับ GCS ผ่านสายเคเบิลที่รวมสายไฟฟ้าและสาย ethernet เข้าไว้ด้วยกัน หุ่นยนต์รับพลังงานจากแหล่งจ่ายไฟฟ้าผ่านหม้อแปลงของ GCS ไปทางสายไฟฟ้า มีการรับส่งข้อมูล ผ่านทางสาย ethernet หุ่นยนต์ถูกควบคุมผ่านจอยสติ๊กหรือสามารถป้อนคำสั่งได้โดยตรงผ่านคอมพิวเตอร์ รวมทั้งมีระบบควบคุมตำแหน่ง ที่ทำให้หุ่นยนต์สามารถรักษาตำแหน่งตัวเองได้อย่างอัตโนมัติ การอ้างอิงทิศทางของหุ่นยนต์ในงานวิจัยนี้ใช้กรอบอ้างอิง (frame) ตามรูปที่ 2.2 ซึ่งเป็นระบบการอ้างอิงโดยทั่วไปที่ใช้สำหรับเรือและเรือดำน้ำ ประกอบด้วย 6 องศาอิสระ ได้แก่ surge, sway, heave, roll, pitch และ yaw หุ่นยนต์สามารถควบคุมได้ 4 องศาอิสระ คือ surge, sway, heave และ yaw ส่วน roll และ pitch นั้นถูกควบคุมโดยการปรับตัวเองจากความสมดุลของโครงสร้าง



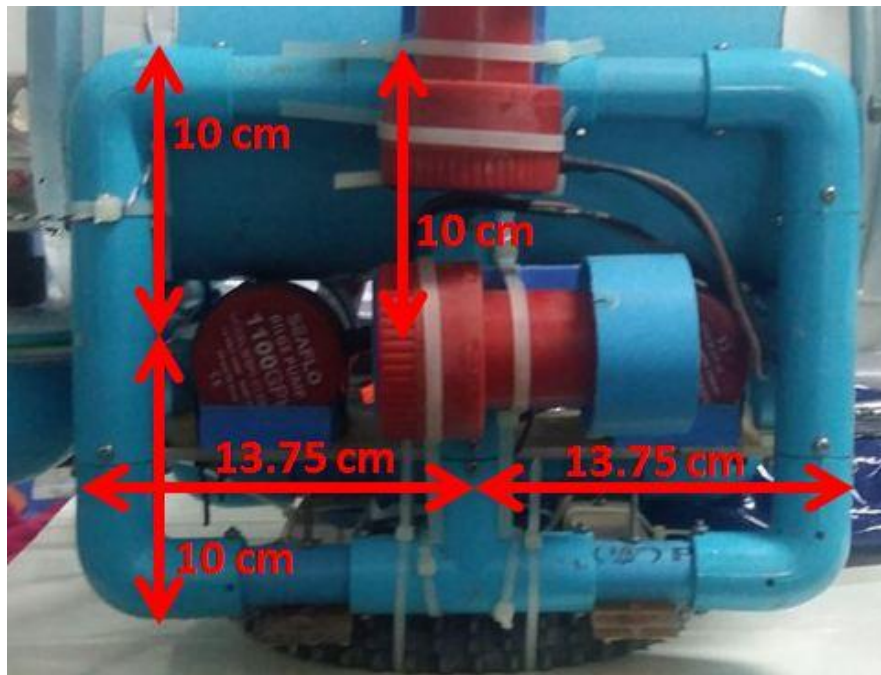
รูปที่ 2.1 ลักษณะโครงสร้างของหุ่นยนต์ดำน้ำภายนอก (ซ้าย) และภายใน (ขวา)



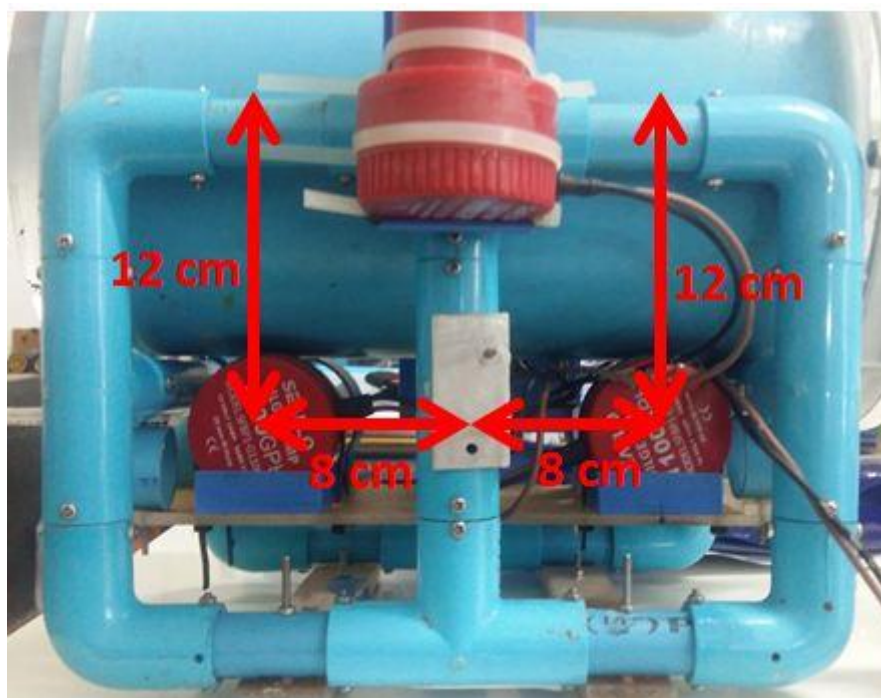
รูปที่ 2.2 กรอบอ้างอิงทิศทางของหุ่นยนต์ดำน้ำ



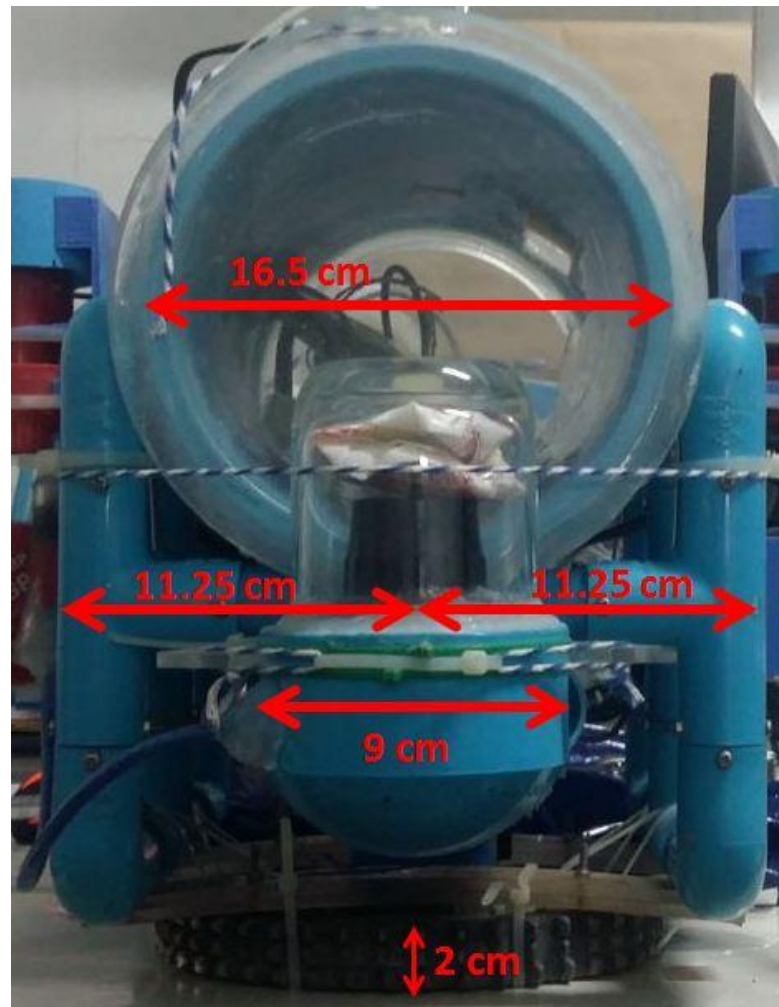
รูปที่ 2.3 ขนาดและตำแหน่งอุปกรณ์ของ ROUV จากมุมมองด้านบน



รูปที่ 2.4 ขนาดและตำแหน่งอุปกรณ์ของ ROUV จากมุมมองด้านข้าง



รูปที่ 2.5 ตำแหน่งของ thruster 3 และ 4 เทียบกับโครงสร้าง



รูปที่ 2.6 ขนาดและตำแหน่งอุปกรณ์ของ ROUV จากมุมมองด้านหน้า

ขนาดและตำแหน่งการติดตั้งอุปกรณ์ในส่วนโครงสร้างแสดงดังรูปที่ 2.2 ถึง 2.6 ส่วนลำตัวมีขนาดเส้นผ่านศูนย์กลาง 16.5 เซนติเมตร ยาว 33 เซนติเมตรเมื่อรวมส่วนฝาปิดความหนา 1 เซนติเมตรแล้ว ส่วนโครงมีความกว้าง 22.5 เซนติเมตร ยาว 27.5 เซนติเมตร สูง 20 เซนติเมตร thruster 1 และ 2 ติดอยู่บริเวณกึ่งกลางของส่วนโครงด้านซ้ายและขวาโดยเมื่อมองจากด้านข้างมีลักษณะตามรูปที่ 2.4 thruster 3 และ 4 ติดอยู่บริเวณด้านล่างชิดกับส่วนลำตัวตามรูปที่ 2.5 โดยห่างจากกึ่งกลางของส่วนโครง 8 เซนติเมตรไปทางซ้ายและขวา และห่างจากขอบบน 12 เซนติเมตร เซนเซอร์ Lidar พร้อมกรอบกันน้ำติดตั้งอยู่บริเวณด้านหน้าของ ROUV ตามรูปที่ 2.6 โดยตำแหน่งที่ติดอยู่บริเวณกึ่งกลางของทั้งความกว้างและความสูงของส่วนโครง กรอบกันน้ำมีลักษณะโดยประมาณใกล้เคียงกับทรงกระบอกโดยมีขนาดเส้นผ่านศูนย์กลาง 9 เซนติเมตร ส่วนสุดท้ายเป็นแผ่นน้ำหนักรัดลักษณะเป็นแผ่นกลมหนา 2 เซนติเมตรติดอยู่

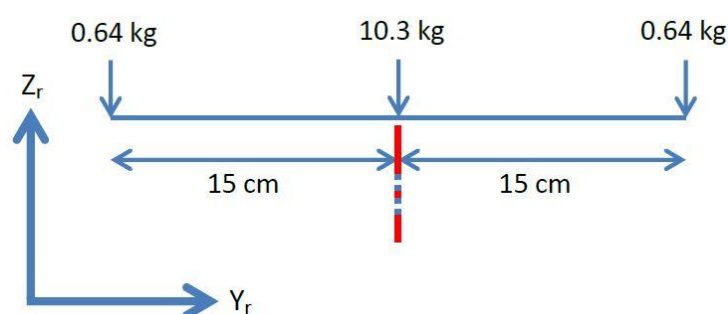
บริเวณด้านล่างของส่วนโครงตามรูปที่ 2.6 โดยอยู่บริเวณกึ่งกลางของความยาวและความกว้างของส่วนโครง

หากพิจารณาแยกเป็นชิ้นส่วน ได้แก่ ส่วนลำตัว ฝาปิด ส่วนโครง thruster น้ำหนักถ่วง และเซนเซอร์ Lidar พร้อมกรอบกันน้ำ ทุกส่วนจะมีลักษณะที่ค่อนข้างสมมาตรโดยน้ำหนักแต่ละส่วนแสดงในตารางที่ 2.1

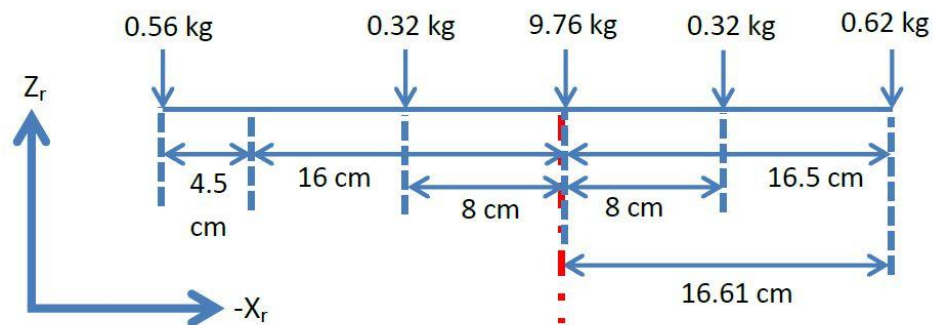
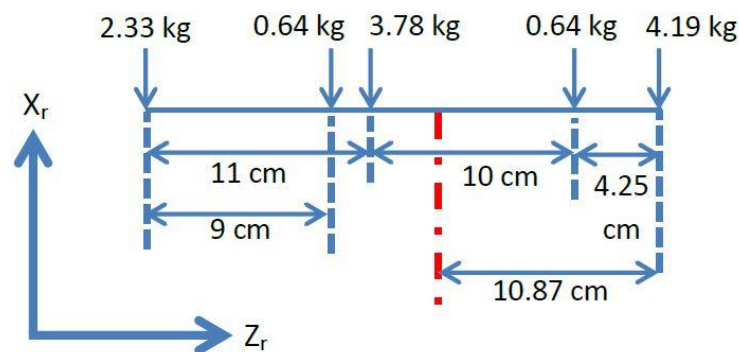
ตารางที่ 2.1 น้ำหนักของชิ้นส่วน ROUV

ชิ้นส่วน	น้ำหนัก (kg)
ลำตัวและอุปกรณ์ภายใน (ไม่รวมฝาปิด)	3.57
ฝาปิดลำตัว	0.62
ส่วนโครง	2.58
Lidar พร้อมกรอบกันน้ำ	0.56
Thruster	0.32
แผ่นน้ำหนักถ่วง	2.33

เมื่อชิ้นส่วนมีลักษณะสมมาตรจุดศูนย์ถ่วงจะอยู่บริเวณกึ่งกลางของชิ้นส่วน รวมทั้งข้อมูลตำแหน่งชิ้นส่วนจากข้างต้นทำให้สามารถคำนวณหาแนวของจุดศูนย์ถ่วงของหุ่นยนต์ในแต่ละแกนได้โดยแผนภาพแรงที่ใช้หาแนวจุดศูนย์ถ่วงแสดงในรูปที่ 2.7 ถึง 2.9



รูปที่ 2.7 แนวจุดศูนย์ถ่วงในระนาบ (Y_r , Z_r)

รูปที่ 2.8 แนวจุดศูนย์ถ่วงในระนาบ $(-X_r, Z_r)$ รูปที่ 2.9 แนวจุดศูนย์ถ่วงในระนาบ (Z_r, X_r)

พิจารณาแนวของจุดศูนย์ถ่วงในระนาบ (Y_r, Z_r) และ $(-X_r, Z_r)$ ของหุ่นยนต์จะเห็นว่าแนวจุดศูนย์ถ่วงอยู่ใกล้เคียงกับจุดกึ่งกลางของส่วนโครง ส่งผลให้หุ่นยนต์สามารถรักษาสสมดุลของมุม Roll และ Pitch ได้ในสภาวะที่ไม่มีการทำงานของ thruster

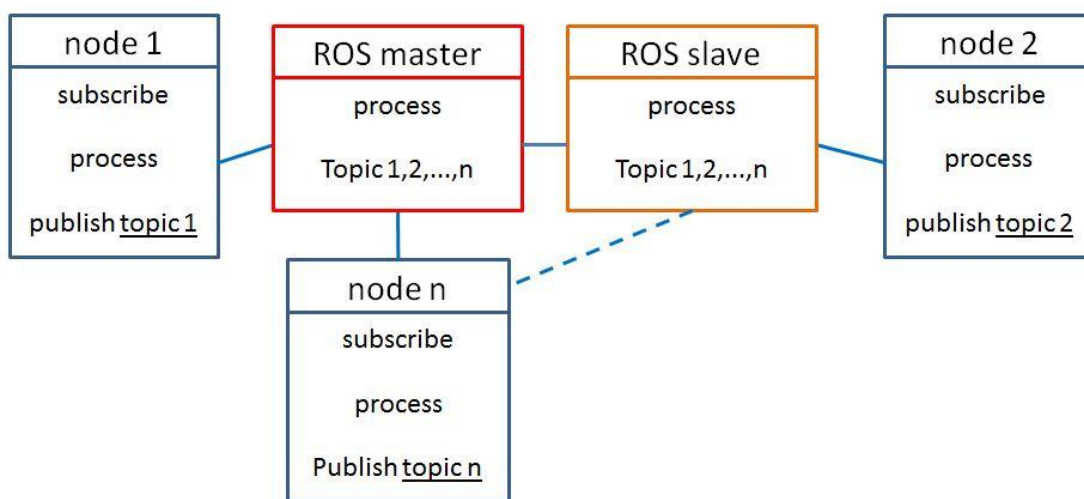
พิจารณาดำแหน่งของ thruster 5 และ 6 จะเห็นว่าในระนาบ (Y_r, Z_r) thruster ทั้งสองวางตัวคู่กันอย่างสมดุลเทียบกับแนวจุดศูนย์ถ่วง และในระนาบ $(-X_r, Z_r)$ thruster ทั้งสองวางตัวในแนวเดียวกันกับแนวจุดศูนย์ถ่วง ดังนั้นขณะ thruster 5 และ 6 ทำงานจะไม่ส่งผลกระทบต่อมุม Roll และ Pitch

พิจารณาดำแหน่งของ thruster 1 และ 2 จะเห็นว่าในระนาบ (Z_r, X_r) thruster ทั้งสองวางตัวเอียงจากแนวจุดศูนย์ถ่วง ส่งผลให้ขณะที่ thruster ทำงานจะเกิดโมเมนต์ที่ส่งผลกระทบต่อมุม Pitch ขึ้น ในทำนองเดียวกัน เมื่อพิจารณาดำแหน่งของ thruster 3 และ 4 จะเห็นว่าในระนาบ (Z_r, X_r) thruster ทั้งสองวางตัวเอียงจากแนวจุดศูนย์ถ่วงเช่นกัน ส่งผลให้ขณะที่ thruster ทำงานจะเกิดโมเมนต์ที่ส่งผลกระทบต่อมุม Roll ขึ้น โดยผลกระทบต่อทั้งสองมุมนี้จะถูกควบคุมโดยการปรับตัวเองจากความสมดุลของโครงสร้าง

2.2 ส่วนโปรแกรมของระบบ

โปรแกรมของระบบระบุตำแหน่งและระบบรักษาตำแหน่งในงานวิจัยนี้ ทำงานอยู่บนพื้นฐานของระบบ Robot Operating System หรือ ROS โดยภาษาทางคอมพิวเตอร์ที่ใช้ในการเขียนโปรแกรมเป็นภาษา Python และ C++ ระบบประกอบด้วยโปรแกรมย่อยหลายโปรแกรมทำหน้าที่ประมวลผลแตกต่างกันออกไปตามส่วนการทำงานของหุ่นยนต์ มีทั้งโปรแกรมที่เขียนและพัฒนาขึ้นเองและโปรแกรมสำเร็จรูปที่มีผู้พัฒนาขึ้นและเปิดให้เข้าใช้งานได้ โดยโปรแกรมทั้งหมดนี้จะทำงานอยู่บนโปรแกรม ROS ทำให้โปรแกรมย่อยแต่ละโปรแกรมสามารถติดต่อสื่อสารกันได้ง่ายโดยอาศัยเครื่องมือจาก ROS

ROS คือ ระบบโครงข่ายการทำงานของโปรแกรม (framework) ที่ออกแบบมาเพื่อใช้ในการพัฒนาหุ่นยนต์โดยเฉพาะ โดยระบบมีความยืดหยุ่นสูงมีเครื่องมือทางด้านซอฟต์แวร์ให้ใช้งานทำให้สามารถประยุกต์ใช้กับแพลตฟอร์มที่หลากหลายของหุ่นยนต์ได้ง่าย โดยระบบจะจัดการการทำงานของโปรแกรมต่างๆ ให้มีความเป็นระบบซึ่งจะลดความซับซ้อนและเพิ่มประสิทธิภาพในการทำงาน รวมถึงการจัดการด้านการรับส่งข้อมูลและควบคุมการทำงานต่างๆ ของโปรแกรมบนระบบที่ช่วยให้ผู้ใช้สามารถเข้าใจและเข้าถึงได้ง่าย ทำการควบคุม แก้ไขหรือปรับเปลี่ยนได้ตามความต้องการ อีกทั้งระบบ ROS ยังมีกลุ่มชุมชนของนักพัฒนาที่รวมตัวกันเพื่อแลกเปลี่ยนแนวคิดและความรู้รวมไปถึงเครื่องมือด้านซอฟต์แวร์หรือโปรแกรมที่ใช้สำหรับการพัฒนาหุ่นยนต์ซึ่งในงานวิจัยนี้ก็มีทางเลือกโปรแกรมที่ถูกพัฒนาไว้มาใช้งานด้วยเช่นกัน



รูปที่ 2.10 แผนผังการทำงานของระบบ ROS

จากรูปที่ 2.10 เป็นแผนผังแสดงหลักการทำงานเบื้องต้นของระบบ ROS แบบ multiple machine ซึ่งเป็นรูปแบบที่ใช้ในงานวิจัย node แสดงถึงโปรแกรมย่อยที่เชื่อมต่อและทำงานอยู่บนระบบ ROS topic คือหัวข้อของข้อมูลที่แต่ละ node ต้องการสื่อสารกัน ROS master คือ

ระบบ ROS หลัก ส่วน ROS slave คือระบบ ROS รอง แต่ละ node ประกอบด้วยการประมวลผล (Process) การประกาศค่า (Publish) และการรับค่า (Subscribe) โดยการประกาศค่าคือการอัปเดตค่าใน topic ของตัวเองไปยังระบบ ROS ทำให้ node อื่นที่ต่ออยู่บนระบบ ROS เดียวกันสามารถเห็นและรับค่าจาก topic นั้นไปใช้ประมวลผลได้ ซึ่งระบบนี้ช่วยให้การติดต่อสื่อสารระหว่างอุปกรณ์ในงานวิจัยทำได้ง่ายขึ้น

2.2.1 โปรแกรมสำเร็จรูป

โปรแกรมสำเร็จรูปที่ใช้ในงานวิจัยเป็นโปรแกรมที่ถูกพัฒนาขึ้นในระบบ ROS ซึ่งสามารถดาวน์โหลด ติดตั้งและใช้งานได้สะดวกหากมีการติดตั้งระบบ ROS ในอุปกรณ์อยู่แล้ว ซึ่งโปรแกรมสำเร็จรูปที่งานวิจัยนี้นำมาใช้งาน ได้แก่

1. joystick_drivers ใช้สำหรับการรับสัญญาณการกดจากจอยสติ๊กที่ต่อกับคอมพิวเตอร์ผ่านทาง USB พอร์ต แปลงสัญญาณเป็นข้อมูลแล้วอัปเดตเข้าสู่ระบบ ROS
2. roserial ใช้เป็นตัวกลางสำหรับจัดการให้ Arduino สามารถสื่อสารกับระบบ ROS ได้
3. urg_node ใช้สำหรับการรับสัญญาณจากเซนเซอร์ Lidar มาแปลงเป็นข้อมูลแล้วอัปเดตเข้าสู่ระบบ ROS
4. imu_filter_madgwick ใช้สำหรับวิเคราะห์และประมวลผลข้อมูลดิบของเซนเซอร์ IMU ให้เป็นข้อมูลทิศทาง
5. hector_slam ใช้สำหรับวิเคราะห์และประมวลผลข้อมูลของเซนเซอร์ Lidar ที่ได้จาก urg_node โดยโปรแกรมจะใช้ข้อมูลสร้างแผนที่ขึ้น และระบุตำแหน่งของเซนเซอร์เทียบกับแผนที่นั้น
6. rqt เป็นโปรแกรมสำหรับแสดงข้อมูลต่างๆที่อยู่บนระบบ ROS โดยชนิดของข้อมูลขึ้นอยู่กับทางเลือกหรือตั้งค่าของผู้ใช้

2.2.2 โปรแกรมที่พัฒนาขึ้นเอง

โปรแกรมย่อยที่พัฒนาขึ้นมาใช้งานเองมีหน้าที่ของแต่ละโปรแกรกดังต่อไปนี้

1. โปรแกรมสำหรับอ่านค่าข้อมูลดิบจากเซนเซอร์ magnetometer และ gyroscope และทำการปรับเทียบค่าเบื้องต้นก่อนจะส่งไปประมวลผลต่อ โปรแกรมเขียนด้วยภาษา Python
2. โปรแกรมสำหรับอ่านค่าของเซนเซอร์วัดความดันและประมวลผลเป็นค่าความลึกประมวลผลค่า output ของระบบควบคุมแบบ PID จากข้อมูลความคลาดเคลื่อนและค่า gain ที่ได้รับ รวมทั้งสั่งงาน thruster ด้วยการขับแบบ PWM ตามคำสั่งจากการกดจอยสติ๊ก หรือผล output จากระบบ PID โปรแกรมนี้เขียนอยู่บนบอร์ด Arduino โดยใช้ภาษา C++

3. โปรแกรมรับค่าการกดแป้นพิมพ์และทำการแปลงเป็นข้อมูลหรือคำสั่งสำหรับหุ่นยนต์ ประกอบไปด้วยข้อมูล gain สำหรับระบบควบคุม PID และข้อมูลตำแหน่งเป้าหมายของหุ่นยนต์ โปรแกรมเขียนด้วยภาษา Python

4. โปรแกรมสำหรับรับค่าตำแหน่งของหุ่นยนต์ในทุกทิศทาง นำมาประมวลผลเป็นค่าความเร็วและความคลาดเคลื่อนของความเร็วแบบตามเวลาจริง เป็นข้อมูลสำหรับระบบควบคุม PID โปรแกรมเขียนด้วยภาษา Python

2.3 อุปกรณ์อิเล็กทรอนิกส์

2.3.1 อุปกรณ์อิเล็กทรอนิกส์ของ GCS

1. คอมพิวเตอร์

คอมพิวเตอร์ทำหน้าที่เป็นส่วนประมวลผลหลักของระบบ ระบบปฏิบัติการที่ติดตั้งบนคอมพิวเตอร์คือ Ubuntu เวอร์ชัน 16.04 ซึ่งระบบปฏิบัติการนี้สามารถรองรับการติดตั้งระบบปฏิบัติการ ROS ได้ ระบบปฏิบัติการ ROS ที่ถูกติดตั้งเป็นรุ่น Kinetic โดยคอมพิวเตอร์ทำหน้าที่ รับข้อมูลจากจอยสติ๊กและแป้นพิมพ์ แล้วอัปเดตเข้าสู่ระบบ ROS จากนั้นเรียกใช้ค่าต่างๆ ที่ถูกอัปเดตเข้าสู่ระบบ ROS มาประมวลผลเป็นค่าตำแหน่ง ความเร็วและความคลาดเคลื่อนระหว่างความเร็วเป้าหมายกับความเร็วปัจจุบัน แล้วอัปเดตค่าทั้งหมดเข้าสู่ระบบ ROS

2. แหล่งพลังงาน

แหล่งพลังงานทำหน้าที่จ่ายพลังงานให้กับหุ่นยนต์ผ่าน 2 เส้นทาง เส้นทางแรกจะเป็นไฟฟ้ากระแสตรงความต่างศักย์ 12 โวลต์ ใช้เป็นแหล่งพลังงานสำหรับขั้วมอเตอร์ เส้นทางที่สองเป็นไฟฟ้ากระแสสลับความต่างศักย์ 220 เป็นแหล่งพลังงานสำหรับอุปกรณ์อิเล็กทรอนิกส์อื่นนอกเหนือจากมอเตอร์ โดยไฟฟ้า 220 โวลต์จะถูกแปลงไปเป็นไฟฟ้ากระแสตรงความต่างศักย์ 5.1 โวลต์ในภายหลังผ่าน Raspberry Pi adapter ที่อยู่ในตัวหุ่นยนต์

3. จอยสติ๊ก

จอยสติ๊กเชื่อมต่ออยู่กับคอมพิวเตอร์ผ่าน USB พอร์ต ทำหน้าที่รับคำสั่งการควบคุม โดยการกดปุ่ม สัญญาณการกดจะถูกอัปเดตเข้าสู่ระบบ ROS ผ่านทางคอมพิวเตอร์

2.3.2 อุปกรณ์อิเล็กทรอนิกส์ของหุ่นยนต์ดำน้ำ

1. คอมพิวเตอร์ขนาดเล็ก

คอมพิวเตอร์ขนาดเล็กที่เลือกใช้ในงานวิจัยคือ Raspberry Pi 2 model B ซึ่งเป็นคอมพิวเตอร์ขนาดเล็กเชื่อมต่อกับคอมพิวเตอร์ที่อยู่ในส่วน GCS ผ่านทางสาย ethernet Raspberry Pi ถูกติดตั้งด้วยระบบปฏิบัติการ Raspbian ซึ่งระบบปฏิบัติการนี้สามารถรองรับการติดตั้งระบบปฏิบัติการ ROS ได้ ระบบปฏิบัติการ ROS ที่ถูกติดตั้งเป็นรุ่น Kinetic โดย

บอร์ด Raspberry Pi ทำหน้าที่ รับข้อมูลจากบอร์ด Arduino, Lidar และ IMU มาประมวลผลเบื้องต้นแล้วอัปเดตข้อมูลเข้าสู่ระบบ ROS รวมถึงรับข้อมูลภาพจากกล้อง Raspberry Pi camera มาแสดงให้กับผู้ใช้งาน ลักษณะของบอร์ด Raspberry Pi model B แสดงในรูปที่ 2.11

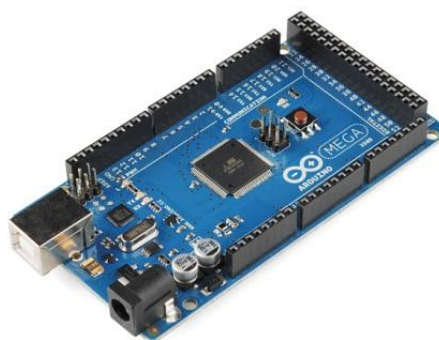


รูปที่ 2.11 บอร์ด Raspberry Pi 2 model B

[ภาพจาก: <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>]

2. ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ที่เลือกใช้ในงานวิจัยคือบอร์ด Arduino รุ่น Mega 2560 โดย Arduino ทำหน้าที่รับค่าจากเซนเซอร์วัดความดันมาประมวลผลเป็นความลึก แล้วอัปเดตเข้าสู่ระบบ ROS ผ่านทาง Raspberry Pi รวมทั้งรับสัญญาณควบคุมจากจอยสติ๊ก หรือเรียกใช้ค่าความคลาดเคลื่อนของความเร็วที่ถูกอัปเดตอยู่บน ROS มาประมวลผลด้วยตัวควบคุม PID ให้ได้ output เป็นสัญญาณสำหรับควบคุม thruster โดยรูปแบบของสัญญาณควบคุมเป็นแบบ PWM ลักษณะของบอร์ด Arduino Mega 2560 แสดงในรูปที่ 2.12

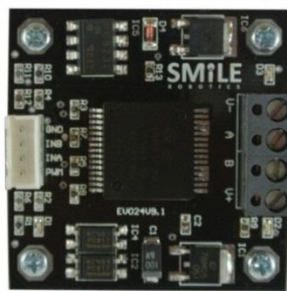


รูปที่ 2.12 บอร์ด Arduino Mega 2560

[ภาพจาก: <https://www.arduinoall.com/product/17/บอร์ด-arduino-รุ่น-mega-2560-r3-พร้อมสาย-usb>]

3. บอร์ดขับเคลื่อนมอเตอร์

บอร์ดขับเคลื่อนมอเตอร์ที่เลือกใช้ในงานวิจัยเป็นบอร์ดขับเคลื่อนมอเตอร์ Smile EVO24V9.3 ซึ่งสามารถรับแรงดันไฟฟ้ากระแสตรงขาเข้าได้ที่ 7-28 โวลต์ โดยมีแรงดันไฟฟ้าขาออกอยู่ที่ $(0.98 \times \text{แรงดันไฟฟ้าขาเข้า})$ โวลต์ สามารถขับกระแสต่อเนื่องได้ 10 แอมป์ และขับกระแสชั่วขณะได้ 30 แอมป์ ที่แรงดัน 24 โวลต์ สามารถควบคุมความเร็วของมอเตอร์ได้ด้วยสัญญาณควบคุมแบบ PWM และควบคุมทิศทางการหมุนของมอเตอร์ได้ด้วยสัญญาณควบคุมดิจิทัล ลักษณะของบอร์ด Smile EVO24V9.3 แสดงในรูปที่ 2.13

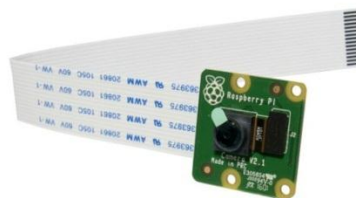


รูปที่ 2.13 บอร์ดขับเคลื่อนมอเตอร์ Smile EVO24V9.3

[ภาพจาก: <https://hobby-experiment.blogspot.com/2014/>]

4. กล้องวิดีโอ

กล้องวิดีโอเลือกใช้ในงานวิจัยคือ Raspberry Pi Camera Module ซึ่งเป็นโมดูลกล้องวิดีโอที่สามารถต่อใช้งานร่วมกับบอร์ด Raspberry Pi ได้ง่าย ขนาดความละเอียด 5 ล้าน pixel สามารถถ่ายวิดีโอที่ความละเอียด 1080p, 720p และ 640x480 ด้วยอัตราแสดงผล 30 (1080p), 60 (720p และ 640x480) และ 90 (640x480) เฟรมต่อวินาที ลักษณะของ Raspberry Pi Camera Module แสดงในรูปที่ 2.14



รูปที่ 2.14 Raspberry Pi Camera Module

[ภาพจาก: <https://solarbotics.com/product/52090/>]

5. เซนเซอร์วัดความดัน

เซนเซอร์วัดความดัน NXP semiconductor MPX5700GP มีลักษณะแสดงในรูปที่ 2.15 ถูกนำมาใช้วัดความดันของน้ำที่เพิ่มขึ้นตามระดับความลึกของหุ่นยนต์ใต้น้ำ จากคุณสมบัติการทำงานที่แรงดัน 4.75-5.25 Vdc และมีแรงดัน output 0.2 Vdc - 4.7 Vdc เป็นสัญญาณอนาล็อก ทำให้เซนเซอร์ชนิดนี้ ใช้งานร่วมกับไมโครคอนโทรลเลอร์ Arduino ได้ง่าย เซนเซอร์ถูกติดตั้งไว้ในส่วนลำตัวของหุ่นยนต์เชื่อมต่อกับ Arduino ผ่านทางช่องรับสัญญาณแบบอนาล็อก แล้วต่อท่อสำหรับวัดความดันบนเซนเซอร์เข้ากับสายยางที่เจาะทะลุออกมายังภายนอกเพื่อวัดความดันของน้ำ ค่าจากเซนเซอร์ถูกอ่านด้วยโปรแกรมที่เขียนอยู่ใน Arduino โดยโปรแกรมจะนำค่าที่อ่านได้มาผ่านการกรองค่าแบบเฉลี่ย (average filter) โดยกำหนดให้ผลลัพธ์มีความถี่อยู่ที่ 20 เฮิรซ์ จากนั้นนำมาผ่านการกรองแบบต่ำผ่าน (low-pass filter) แล้วใช้ข้อมูลดังกล่าวไปคำนวณระดับความลึก โดยความสัมพันธ์ระหว่างข้อมูลความดันจากเซนเซอร์และความลึกหาได้จากการทดสอบเปรียบเทียบเซนเซอร์ โดยการต่อเซนเซอร์เข้ากับสายยางใสที่วางตัวในแนวตั้งตามรูปที่ 2.16 เพิ่มระดับน้ำในสายยางครั้งละ 5 เซนติเมตร อ่านค่าที่วัดได้และบันทึกผล จนระดับน้ำสูงถึง 2 เมตร นำข้อมูลที่ได้มาเขียนกราฟความสัมพันธ์ระหว่างความดันที่วัดได้กับความสูงของน้ำ ผลที่ได้แสดงในรูปที่ 2.17 จะเห็นว่าความสัมพันธ์ระหว่างความดันและความสูงของน้ำมีแนวโน้มเป็นเชิงเส้น และมีสมการความสัมพันธ์กับความลึกเป็น $D = 8.3432 \times P - D_0$ เมื่อ D คือ ระดับความลึกจากระดับผิวน้ำ (cm) P คือ ค่าที่อ่านได้จากเซนเซอร์ความดัน และ D_0 คือค่าข้อมูลเซนเซอร์ที่ได้จากการวัดเหนือผิวน้ำ (cm) โดยมีช่วงความคลาดเคลื่อนอยู่ระหว่าง -3.2 เซนติเมตร ถึง 3.1 เซนติเมตร

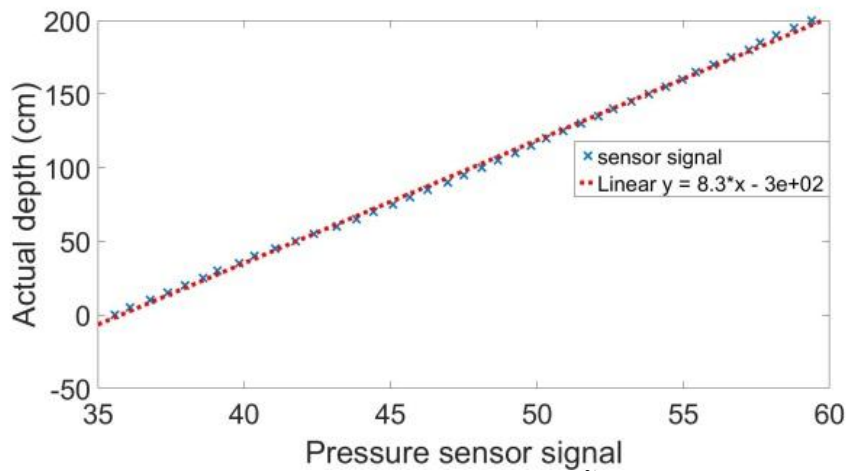


รูปที่ 2.15 เซนเซอร์วัดความดัน MPX5700GP

[ภาพจาก: <https://th.rs-online.com/web/p/gauge-pressure-sensor-ics/7191112/>]



รูปที่ 2.16 การทดสอบปรับเทียบเซนเซอร์วัดความดัน



รูปที่ 2.17 ความสัมพันธ์ระหว่างข้อมูลความสูงของน้ำกับสัญญาณจากเซนเซอร์

6. เซนเซอร์วัดทิศทาง

เซนเซอร์ IMU GY-87 ถูกนำมาใช้ในการวัดมุม yaw โดยติดตั้งไว้ในส่วนลำตัวของหุ่นยนต์ เชื่อมต่อกับบอร์ด Raspberry Pi ผ่านทาง I²C port ค่าของมุม yaw ได้จากการนำข้อมูลจากเซนเซอร์ gyroscope และเซนเซอร์ magnetometer ที่อยู่ภายใน IMU มาประมวลผล โดยค่าดิบของเซนเซอร์ทั้งสองถูกอ่านด้วยโปรแกรมที่เขียนอยู่บนบอร์ด Raspberry Pi ด้วยภาษา Python โดยโปรแกรมจะทำการปรับเทียบค่าสัญญาณของเซนเซอร์ magnetometer โดยอาศัยสมการปรับเทียบ soft iron calibration จากนั้นทำการอัปเดตค่าเข้าสู่ระบบ ROS ต่อมา

ค่าสัญญาณจะถูกเรียกใช้โดยโปรแกรม `imu_filter_madgwick` ซึ่งเป็นโปรแกรมที่อยู่ในระบบ ROS บนคอมพิวเตอร์ โปรแกรมนี้จะนำค่าข้อมูลจากเซนเซอร์ gyroscope และเซนเซอร์ magnetometer มาประมวลผลร่วมกัน โดยโปรแกรมนี้ใช้อัลกอริทึมรวมค่า (fusion algorithm) และอัลกอริทึมแบบคาลมาน (Kalman-based algorithm) ในการประมวลผล ได้ผลลัพธ์เป็นทิศทางการหันของเซนเซอร์ในทิศ roll, pitch และ yaw แล้วทำการอัปเดตเข้าสู่ระบบ ROS สำหรับการเรียกใช้ไปประมวลผลต่อไป

เซนเซอร์ทั้งสองชนิดนี้มีการใช้งานกันอย่างแพร่หลาย ให้ข้อมูลการวัดมุมที่แม่นยำและน่าเชื่อถือเพียงพอต่อการใช้งาน [3] เพียงแค่ต้องทำการปรับเทียบค่าเริ่มต้นสำหรับการใช้งานเท่านั้น ลักษณะของเซนเซอร์ IMU GY-87 แสดงในรูปที่ 2.18 สมการปรับเทียบ soft iron calibration สามารถเขียนได้เป็น

$$\tilde{\mathbf{M}}_{calibrated} = \frac{(\mathbf{avg}_{scale})}{\tilde{\mathbf{M}} - \tilde{\mathbf{M}}_{offset}} \quad (2.1)$$

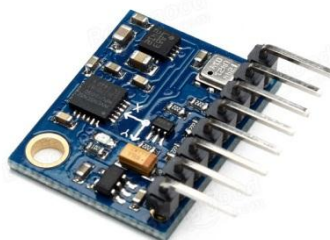
โดย $\tilde{\mathbf{M}}_{calibrated}$ คือเวกเตอร์ค่าสนามแม่เหล็กที่ได้หลังจากการปรับเทียบ $[\mathbf{M}_{Xcalibrated}, \mathbf{M}_{Ycalibrated}, \mathbf{M}_{Zcalibrated}]^T$, \mathbf{avg}_{scale} คือค่าอัตราส่วนสนามแม่เหล็กเฉลี่ย, $\tilde{\mathbf{M}}$ คือเวกเตอร์ค่าสนามแม่เหล็กจากเซนเซอร์ $[\mathbf{M}_X, \mathbf{M}_Y, \mathbf{M}_Z]^T$, $\tilde{\mathbf{M}}_{offset}$ คือเวกเตอร์ค่าสนามแม่เหล็กชดเชย $[\mathbf{M}_{Xoffset}, \mathbf{M}_{Yoffset}, \mathbf{M}_{Zoffset}]^T$

$$\tilde{\mathbf{M}}_{scale} = \frac{(\tilde{\mathbf{M}}_{max} - \tilde{\mathbf{M}}_{min})}{2} \quad (2.2)$$

โดย $\tilde{\mathbf{M}}_{scale}$ คือเวกเตอร์ค่าอัตราส่วนสนามแม่เหล็ก $[\mathbf{M}_{Xscale}, \mathbf{M}_{Yscale}, \mathbf{M}_{Zscale}]^T$, $\tilde{\mathbf{M}}_{max}$ คือเวกเตอร์ค่าสนามแม่เหล็กสูงสุด $[\mathbf{M}_{Xmax}, \mathbf{M}_{Ymax}, \mathbf{M}_{Zmax}]^T$, $\tilde{\mathbf{M}}_{min}$ คือเวกเตอร์ค่าสนามแม่เหล็กต่ำสุด $[\mathbf{M}_{Xmin}, \mathbf{M}_{Ymin}, \mathbf{M}_{Zmin}]^T$

$$\mathbf{avg}_{scale} = \frac{(\mathbf{M}_{xscale} + \mathbf{M}_{yscale} + \mathbf{M}_{zscale})}{3} \quad (2.3)$$

$$\tilde{\mathbf{M}}_{offset} = \frac{(\tilde{\mathbf{M}}_{max} + \tilde{\mathbf{M}}_{min})}{2} \quad (2.4)$$



รูปที่ 2.18 เซนเซอร์ IMU GY-87

[ภาพจาก: https://sea.banggood.com/GY-87-10DOF-MPU6050-HMC5883L-BMP180-Sensor-Module-p-963792.html?cur_warehouse=CN]

7. มอเตอร์ขับเคลื่อน

มอเตอร์ขับเคลื่อนหรือ thruster ถูกดัดแปลงมาจากปั๊มน้ำขนาดเล็ก (Seaflo bilge pump 1100GPH 12V) โดยการตัดส่วนที่เป็นท่อและใบพัดสำหรับปั๊มน้ำออก เหลือไว้เพียงมอเตอร์ที่มีซีลกันน้ำ แล้วติดตั้งใบพัดสำหรับขับเคลื่อนหุ่นยนต์แทนที่ใบพัดเดิม ใบพัดขับเคลื่อนขึ้นรูปจากเครื่องพิมพ์สามมิติ ถูกออกแบบให้ใบพัดที่ทำงานคู่กันมีทิศทางตรงข้ามกัน เพื่อให้แรงบิดของมอเตอร์หักล้างซึ่งกันและกันเมื่อขับเคลื่อนหุ่นยนต์ในทิศ heave, surge และ sway สำหรับการหมุนในทิศ yaw อาศัยคู่มอเตอร์ในทิศ surge และ sway ที่ขับเคลื่อนในทิศตรงข้ามกันจึงทำให้แรงบิดลัพธ์รอบแกน Zr ที่ไม่เท่ากับศูนย์เป็นแรงบิดควบคุมการหมุนในทิศ yaw

กรณีที่คู่มอเตอร์ในทิศ surge ขับเคลื่อนในทิศเดียวกันแต่แรงบิดของมอเตอร์หักล้างกันไม่หมดเพราะความเร็วรอบไม่เท่ากันจะส่งผลกระทบต่ออาการหมุนในทิศ roll ในทำนองเดียวแรงบิดของคู่มอเตอร์ในทิศ sway ที่หักล้างไม่หมดจะส่งผลกระทบต่ออาการหมุนในทิศ pitch แต่ก็ไม่ได้มีผลกระทบมากนัก เพราะการทรงตัวในทิศ roll และ pitch ถูกปรับด้วยสมดุลงyro อยู่แล้ว ลักษณะของปั๊มน้ำก่อนและหลังดัดแปลงแสดงในรูปที่ 2.19



รูปที่ 2.19 ป้มน้ำ Seaflo bilge pump 1100GPH 12V ก่อนดัดแปลง (ซ้าย) และหลังดัดแปลง (ขวา)

[ภาพจาก: <https://www.amazon.com/Seaflo-Marine-Plumbing-Electric-1100gph/dp/B011TVXR8A>]

8. Lidar

การหาตำแหน่งในระนาบ X และ Y ของหุ่นยนต์เทียบกับสิ่งแวดล้อมทำได้โดยการใช้ Lidar (HOKUYO URG-04LX-UG01) ควบคู่กับอัลกอริทึม hector_slam [16] ซึ่งเป็น ROS package ที่จะนำข้อมูลการสแกนระยะจากเซนเซอร์มาประมวลผลสร้างแผนที่ของสิ่งแวดล้อม รวมทั้งระบุตำแหน่งหุ่นยนต์เทียบกับสิ่งแวดล้อมนั้น ทั้งนี้การใช้งาน Lidar ใต้น้ำยังมีข้อจำกัด เนื่องจากสีของแสงจาก Lidar มีสีแดงและคุณสมบัติของน้ำที่ดูดกลืนแสงสีแดงได้ดี ทำให้ระยะการทำงานของ Lidar ลดลงจากปกติ 4 เมตร เหลือประมาณ 40 เซนติเมตรในกรณีที่น้ำใส และหากน้ำขุ่นหรือมีตะกอนอาจจะส่งผลให้เซนเซอร์ทำงานไม่ได้ จึงเป็นข้อจำกัดของเซนเซอร์ที่ต้องอยู่ในระยะที่สามารถตรวจจับโครงสร้างใต้น้ำได้ ในขณะที่เดียวกันผนังหรือสิ่งแวดล้อมก็จำเป็นต้องมีจุดสังเกต (key point) มากเพียงพอที่อัลกอริทึมจะสามารถนำข้อมูลไปประมวลผลเป็นจุดอ้างอิงในแผนที่ได้ ลักษณะของเซนเซอร์ Lidar ก่อนและหลังการดัดแปลงแสดงในรูปที่ 2.20



รูปที่ 2.20 เซนเซอร์ HOKUYO URG-04LX-UG01 ก่อนดัดแปลง (ซ้าย) และหลังดัดแปลง (ขวา)

[ภาพจาก: <https://www.robotshop.com/en/hokuyo-urg-04lx-ug01-scanning-laser-rangefinder.html>]

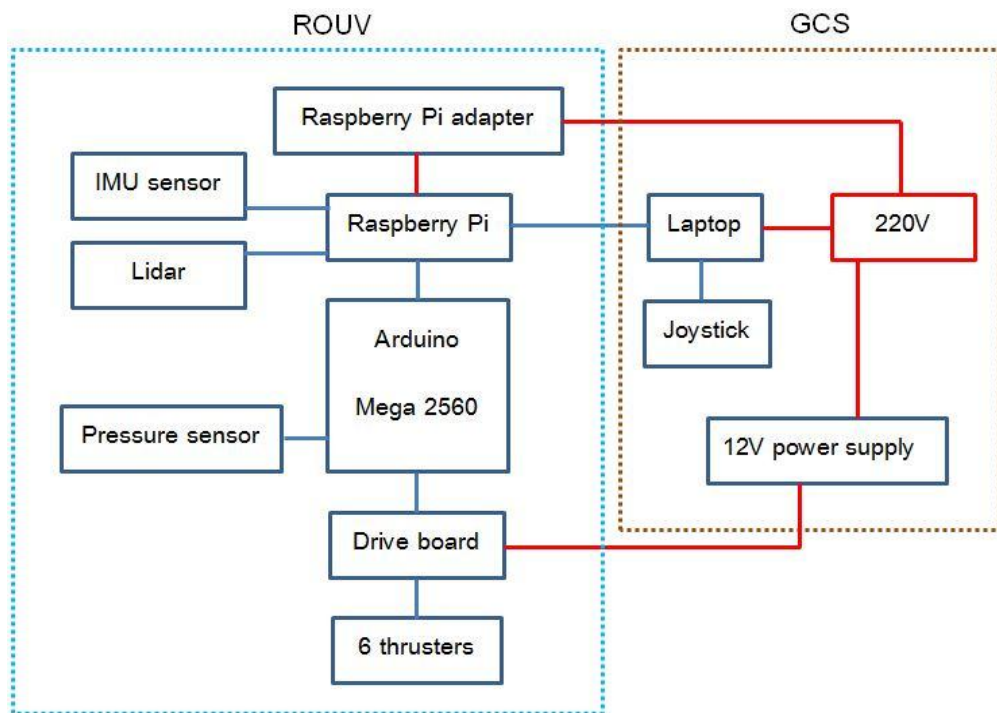
9. สายไฟ

สายไฟที่เชื่อมต่อระหว่างแหล่งจ่ายพลังงานและ ROUV เป็นสายไฟชนิด VFF (T.10) ขนาด 2×1.5 ตารางมิลลิเมตร ยาว 10 เมตรจำนวนสองเส้น โดยเส้นหนึ่งเป็นไฟฟ้าสำหรับเลี้ยงบอร์ด Raspberry Pi และอีกเส้นสำหรับเลี้ยงบอร์ดขับเคลื่อนมอเตอร์

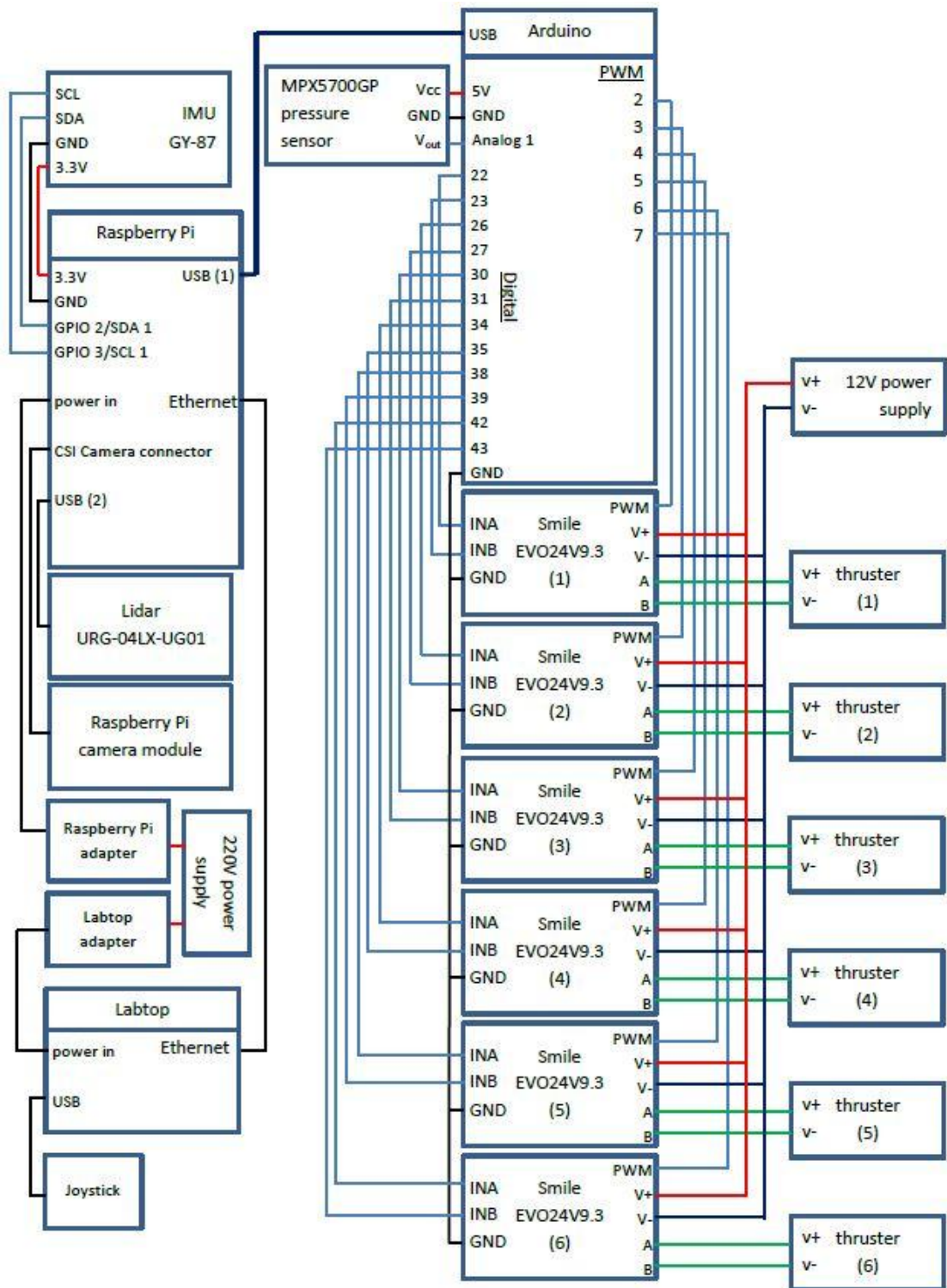
10. สาย Ethernet

สาย Ethernet ที่ใช้เป็นชนิด Cat.5e ความเร็วการส่งข้อมูลสูงสุด 1 Gbps ความยาว 10 เมตร จำนวนหนึ่งเส้น ใช้สำหรับการสื่อสารระหว่าง Raspberry Pi และ Laptop

ลักษณะการเชื่อมต่อของอุปกรณ์อิเล็กทรอนิกส์ทั้งหมดแสดงในรูปที่ 2.21 และ Schematic diagram ของระบบอิเล็กทรอนิกส์แสดงในรูปที่ 2.22



รูปที่ 2.21 แผนภาพการเชื่อมต่ออุปกรณ์ของหุ่นยนต์ดำน้ำและ GCS



รูปที่ 2.22 Schematic diagram ของระบบอิเล็กทรอนิกส์

บทที่ 3

การทำงานของระบบ

3.1 ระบบรักษาตำแหน่ง

ระบบรักษาตำแหน่งออกแบบโดยใช้หลักการของระบบควบคุมแบบป้อนกลับ ระบบควบคุมแบบป้อนกลับ (feedback control system) คือระบบการควบคุมที่มีการเปรียบเทียบค่า output และค่า input ผลที่ได้คือค่าความคลาดเคลื่อน (error) จะถูกป้อนกลับเข้าไปในระบบ เพื่อให้ระบบดำเนินกระบวนการใหม่จนกระทั่งค่า output มีความใกล้เคียงกับค่า input งานวิจัยนี้ใช้ระบบควบคุมแบบป้อนกลับ PI เนื่องจากข้อมูลที่ทำการควบคุมเป็นข้อมูลความเร็ว รวมไปถึงระบบควบคุมแบบ PI เป็นระบบควบคุมที่ไม่ซับซ้อนมากนัก สามารถทำความเข้าใจได้ง่าย นำมาประยุกต์ใช้กับระบบและอุปกรณ์ได้สะดวก รวมทั้งประสิทธิภาพก็เพียงพอสำหรับการควบคุมหากมีค่าอัตราส่วน gain ที่เหมาะสม [12]

งานวิจัยอาศัยการอ้างอิงตำแหน่งของหุ่นยนต์เทียบกับกรอบอ้างอิงตรึง (fixed frame) จาก hector_slam ซึ่งเกิดขึ้นโดยมีจุดเริ่มต้นอยู่ที่ตำแหน่งของหุ่นยนต์ขณะเริ่มทำงาน การควบคุมหุ่นยนต์เพื่อรักษาตำแหน่งต้องอาศัยข้อมูลตำแหน่งอ้างอิงที่ผ่านการแปลงจากกรอบอ้างอิงตรึง มาอยู่ในกรอบอ้างอิงของหุ่นยนต์ (robot frame) ซึ่งสามารถหาสมการการแปลงได้จากรูปที่ 3.1 ได้แก่

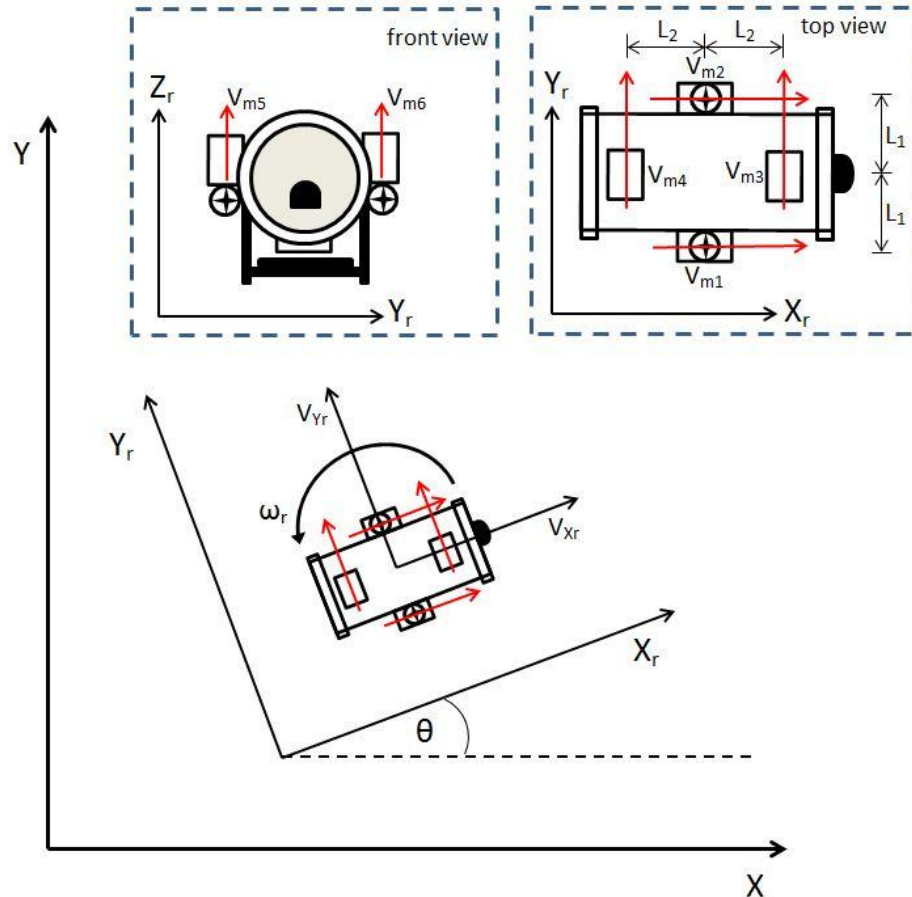
$$\tilde{V}_r = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tilde{V} \quad (3.1)$$

เมื่อ \tilde{V} คือ เวกเตอร์ความเร็ว $[V_x, V_y, V_z, \omega]^T$ เทียบกับ fixed frame, \tilde{V}_r คือเวกเตอร์ความเร็ว $[V_{xr}, V_{yr}, V_{zr}, \omega_r]^T$ เทียบกับ robot frame, θ คือ มุมของ robot frame เทียบกับ fixed frame รอบแกน Z และ ω แทนความเร็วเชิงมุมรอบแกน Z

สมการความเร็วเชิงเส้นของมอเตอร์ทั้งหกตัวตามกรอบอ้างอิงรูปที่ 3.1 ได้แก่

$$\tilde{V}_m = \begin{bmatrix} 1 & 0 & 0 & L_1 \\ 1 & 0 & 0 & -L_1 \\ 0 & 1 & 0 & L_2 \\ 0 & 1 & 0 & -L_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tilde{V}_r \quad (3.2)$$

เมื่อ \tilde{V}_m คือเวกเตอร์ความเร็วของมอเตอร์ $[V_{m1}, V_{m2}, V_{m3}, V_{m4}, V_{m5}, V_{m6}]^T$, L_1 และ L_2 คือ ระยะห่างระหว่างจุดกึ่งกลางหุ่นยนต์กับตำแหน่งของมอเตอร์



รูปที่ 3.1 กรอบอ้างอิงตรง $[X, Y, Z]$ เทียบกับกรอบอ้างอิงของหุ่นยนต์ $[X_r, Y_r, Z_r]$

สัญญาณควบคุมมอเตอร์ได้จากการประมวลผลความคลาดเคลื่อนของความเร็วด้วยระบบควบคุม PID โดยความเร็วที่ต้องการ (\tilde{V}_{set}) มีความสัมพันธ์กับสัญญาณตำแหน่งตามสมการ

$$\tilde{V}_{set} = \tilde{K} \cdot (\tilde{X}_{set} - \tilde{X}_{current}) \quad (3.3)$$

เมื่อ \tilde{V}_{set} คือเวกเตอร์สถานะความเร็วที่ต้องการ $[V_{Xset}, V_{Yset}, V_{Zset}, \omega_{set}]^T$, \tilde{X}_{set} คือเวกเตอร์สถานะตำแหน่งเป้าหมาย $[X_{set}, Y_{set}, Z_{set}, \theta_{set}]^T$,

$\tilde{X}_{current}$ คือเวกเตอร์สถานะตำแหน่งปัจจุบัน $[X_{current}, Y_{current}, Z_{current}, \theta_{current}]^T$ และ \tilde{K} คือตัวปรับค่าสัดส่วน $[K_X, K_Y, K_Z, K_\omega]^T$ โดยค่านี้จะส่งผลต่ออัตราการปรับตำแหน่งของระบบ มีหน่วยเป็น [1/วินาที]

ความเร็ว ณ ตำแหน่งปัจจุบันมีความสัมพันธ์กับสัญญาณตำแหน่งตามสมการ

$$\tilde{V}_{current} = (\tilde{X}_{current} - \tilde{X}_{previous})/\tilde{t} \quad (3.4)$$

เมื่อ $\tilde{V}_{current}$ คือเวกเตอร์สถานะความเร็วปัจจุบัน $[V_{Xcurrent}, V_{Ycurrent}, V_{Zcurrent}, \omega_{current}]^T$, $\tilde{X}_{previous}$ คือเวกเตอร์สถานะตำแหน่งก่อนปัจจุบัน $[X_{previous}, Y_{previous}, Z_{previous}, \theta_{previous}]^T$, \tilde{t} คือความแตกต่างของเวลาระหว่างตำแหน่งปัจจุบันกับตำแหน่งก่อนปัจจุบัน $[t_X, t_Y, t_Z, t_\omega]^T$ โดยที่ t_X, t_Y มาจากข้อมูล real time ของ hector_slam, t_Z มาจากข้อมูลของเซนเซอร์วัดความดัน และ t_ω มาจากข้อมูลของ IMU ซึ่งค่า $\omega_{current}$ สามารถอ่านได้โดยตรงจากเซนเซอร์ IMU เช่นกัน

จากสมการที่ (3.3) และ (3.4) คำนวณหาความคลาดเคลื่อนของความเร็วได้เป็น

$$\tilde{e} = \tilde{V}_{set} - \tilde{V}_{current} \quad (3.5)$$

เมื่อ \tilde{e} คือเวกเตอร์สถานะความคลาดเคลื่อนของความเร็วอ้างอิงเทียบกับกรอบอ้างอิงจริง $[e_X, e_Y, e_Z, e_\omega]^T$

จากนั้นอาศัยการแปลงสมการ (3.1) แปลงความคลาดเคลื่อนให้มาอยู่ในแกน robot frame ได้เป็น

$$\tilde{e}_r = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tilde{e} \quad (3.6)$$

เมื่อ \tilde{e}_r คือเวกเตอร์สถานะความคลาดเคลื่อนของความเร็วอ้างอิงเทียบกับกรอบอ้างอิงของหุ่นยนต์ $[e_{Xr}, e_{Yr}, e_{Zr}, e_{\omega r}]^T$

ระบบรักษาตำแหน่งของหุ่นยนต์ใช้ตัวควบคุมแบบ PI ซึ่งระบบควบคุมแบบ PI สามารถทำความเข้าใจได้ง่าย ประยุกต์ใช้งานได้หลากหลาย [12] ระบบควบคุมแบบป้อนกลับที่มีตัวควบคุมแบบ PI สามารถออกแบบได้ดังนี้

$$\tilde{u}(t) = \tilde{K}_p \cdot \tilde{e}_r(t) + \tilde{K}_i \cdot \sum_{i=t_0}^t \tilde{e}_r(i) \cdot T \quad (3.7)$$

เมื่อ $\tilde{\mathbf{u}}(t)$ คือ เวกเตอร์สัญญาณควบคุม $[\mathbf{u}_x, \mathbf{u}_y, \mathbf{u}_z, \mathbf{u}_\omega]^T$, $\tilde{\mathbf{K}}_p$ คือ proportional gain $[\mathbf{K}_{px}, \mathbf{K}_{py}, \mathbf{K}_{pz}, \mathbf{K}_{p\omega}]^T$, $\tilde{\mathbf{K}}_i$ คือ integral gain $[\mathbf{K}_{ix}, \mathbf{K}_{iy}, \mathbf{K}_{iz}, \mathbf{K}_{i\omega}]^T$, T คือ คาบเวลาของข้อมูล

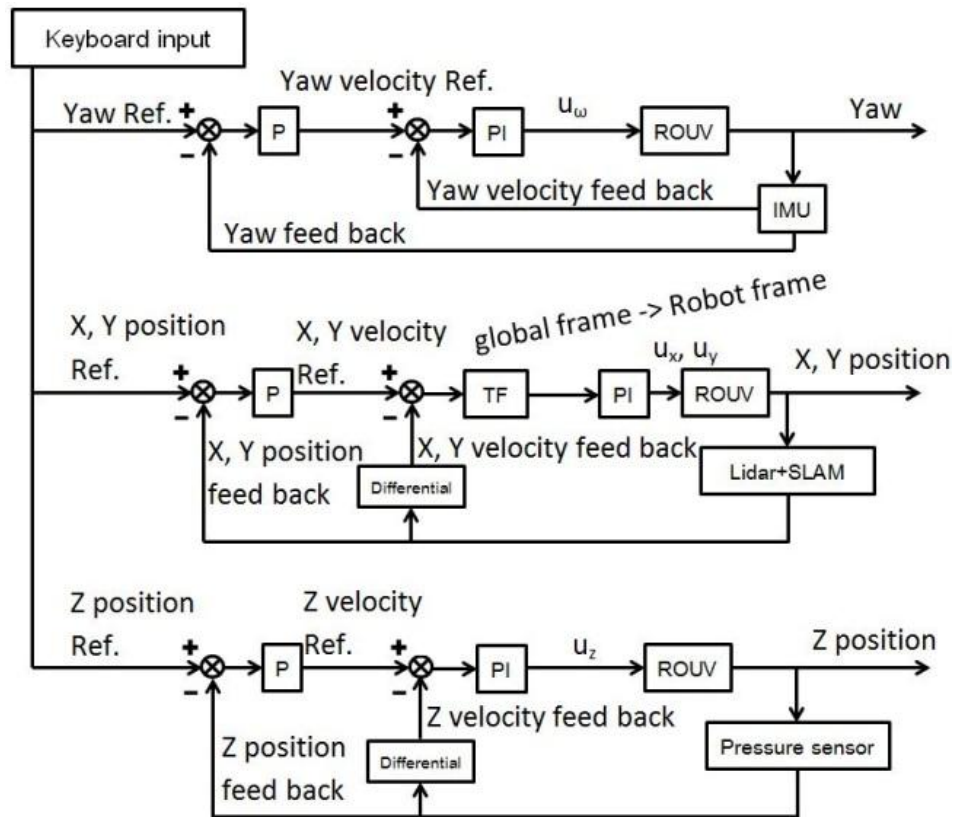
สัญญาณควบคุมที่ได้ถูกนำไปสั่งการมอเตอร์ทั้งหมดด้วยความสัมพันธ์ตามสมการ

$$\tilde{\mathbf{u}}_m(t) = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tilde{\mathbf{u}}(t) \quad (3.8)$$

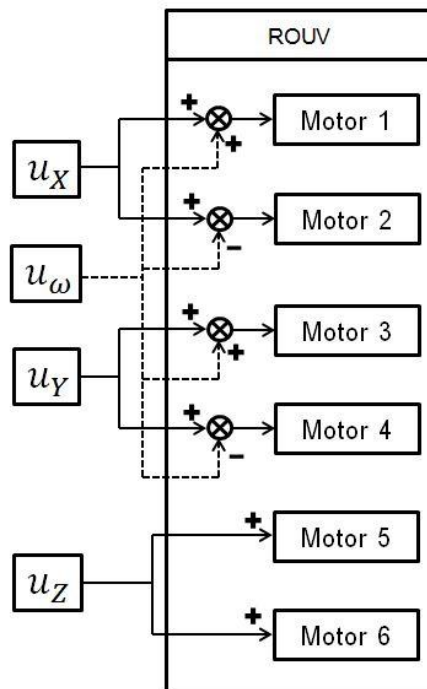
เมื่อ $\tilde{\mathbf{u}}_m(t)$ คือ สัญญาณควบคุมสำหรับมอเตอร์ตัวที่ 1 ถึงตัวที่ 6 $[\mathbf{u}_{m1}, \mathbf{u}_{m2}, \mathbf{u}_{m3}, \mathbf{u}_{m4}, \mathbf{u}_{m5}, \mathbf{u}_{m6}]^T$ ตามลำดับ โดยเครื่องหมายที่ปรากฏในสมการบ่งบอกทิศทางการเคลื่อนที่ของ thruster ซึ่งตัวควบคุมจะทำการปรับค่าสัญญาณสั่งการอีกครั้งเพื่อนำไปควบคุมทิศทางการหมุนของ thruster ให้เกิดการเคลื่อนที่ตามทิศทางต้องการ รวมทั้งการควบคุมหุ่นยนต์จะเป็นรูปแบบที่กำหนดให้แรงขับจาก thruster กระทำที่จุดเดียวกัน เสมือนการกำหนดให้หุ่นยนต์และ thruster เป็นจุดที่ต้องการควบคุมการเคลื่อนที่ ดังนั้นระบบจะไม่มีการใช้ค่าชดเชยจากระยะเยื้องของ thruster ในการประมวลผล ซึ่งผลกระทบจากระยะเยื้องจะถูกควบคุมโดยสมดุลโครงสร้างของหุ่นยนต์เอง

การจูนหาพารามิเตอร์ \mathbf{K}_p และ \mathbf{K}_i ที่เหมาะสมสำหรับระบบควบคุมนี้ใช้วิธีการจูนโดยสังเกตจากผลการตอบสนองของระบบ โดยทดสอบหาค่าแยกกันในแต่ละแนวแกนการเคลื่อนที่ในงานวิจัยนี้ ผลตอบสนองของความเร็วเป็นข้อมูลที่มีการเปลี่ยนแปลงความถี่สูงทำให้ยากต่อการสังเกต จึงต้องอาศัยการสังเกตผลตอบสนองของตำแหน่งควบคู่ไปด้วย การจูนเริ่มต้นด้วยการปรับตั้งค่า \mathbf{K}_i เป็น 0 จากนั้นเพิ่มค่า \mathbf{K}_p จนกระทั่งผลตอบสนองของระบบขณะ steady state เกิดการแกว่ง (oscillation) จากนั้นทำการลดค่า \mathbf{K}_p ที่ได้ลงครึ่งหนึ่งแล้วเพิ่มค่า \mathbf{K}_i จนผลตอบสนองตรงกับค่าเป้าหมาย โดยค่า \mathbf{K}_i จะต้องไม่มากเกินไปจนทำให้ระบบเกิดการแกว่งหรือทำให้เกิดการ overshoot ที่มากเกินไปในกรณีที่มีการเคลื่อนที่

ระบบระบุตำแหน่งทำการคำนวณค่าตำแหน่งของหุ่นยนต์จากข้อมูลเซนเซอร์ โดยตำแหน่งของหุ่นยนต์จะถูกนำมาแสดงผลเป็นสถานะของหุ่นยนต์และประมวลผลเป็นสัญญาณควบคุมการทรงตัว (stabilization control) ตามสมการที่ (3.3) ถึงสมการที่ (3.7) แผนภาพการทำงานของระบบรักษาตำแหน่งแสดงดังรูปที่ 3.2 และสัญญาณการสั่งงานมอเตอร์เป็นไปตามความสัมพันธ์ของสมการที่ (3.8) แสดงดังรูปที่ 3.3



รูปที่ 3.2 แผนภาพการทำงานของระบบควบคุมตำแหน่งหุ่นยนต์ใต้น้ำแบบสี่งศาอิสระ



รูปที่ 3.3 สัญญาณควบคุมมอเตอร์ขับเคลื่อนหุ่นยนต์ใต้น้ำ

3.2 ระบบระบุตำแหน่ง

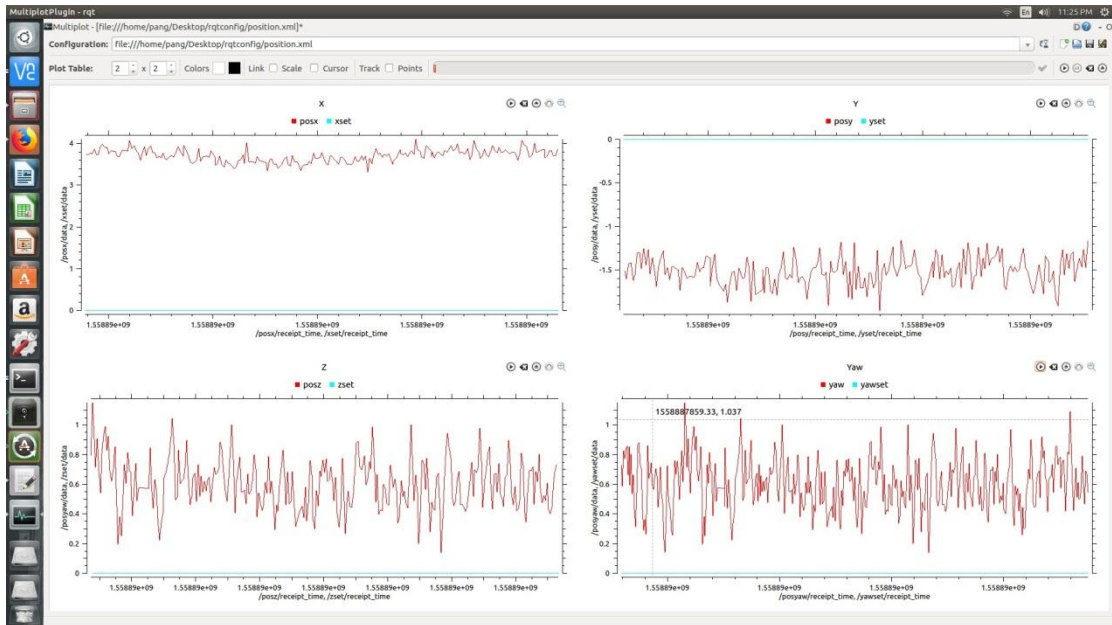
ระบบระบุตำแหน่งอาศัยข้อมูลจากเซนเซอร์หลายชนิดร่วมกัน ได้แก่ IMU ใช้ในการวัดมุม yaw เซนเซอร์วัดความดันในการวัดระดับความลึกจากผิวน้ำ และเซนเซอร์ Lidar ร่วมกับ hector_slam ในการหาตำแหน่งในแนวระนาบ โดยหลักการทำงานของ hector_slam เป็นการทำงานร่วมกันระหว่าง package เปรียบเทียบข้อมูลสแกน (Scan Matching) กับ package ทำแผนที่ (Mapping) โดย Scan Matching จะทำการเปรียบเทียบข้อมูลที่ได้จากการสแกนกับแผนที่ที่มีอยู่เพื่อประมวลผลหาตำแหน่งของเซนเซอร์บนหุ่นยนต์เทียบกับแผนที่ในขณะที่ Mapping จะทำการวาดแผนที่สิ่งแวดล้อมเพิ่มเติมด้วยข้อมูลที่ได้จากการสแกนประกอบกับตำแหน่งของเซนเซอร์ในขณะนั้น โดยในตอนเริ่มต้นที่ยังไม่มีแผนที่ โปรแกรมจะทำการวาดสิ่งแวดล้อม ณ จุดเริ่มต้นจากข้อมูลสแกนขึ้นมา โดยกำหนดให้ตำแหน่งเซนเซอร์ขณะนั้นเป็นจุดเริ่มต้นของกรอบอ้างอิงตรึง (fixed frame) ด้วยเหตุนี้ hector_slam จึงสามารถใช้ในการหาตำแหน่งเทียบกับสิ่งแวดล้อมที่ไม่รู้จักได้ ค่าตำแหน่งที่ได้จากระบบระบุตำแหน่งนี้จะถูกนำไปใช้ประมวลผลในระบบรักษาตำแหน่งในลำดับต่อไป

3.3 ส่วนแสดงผล

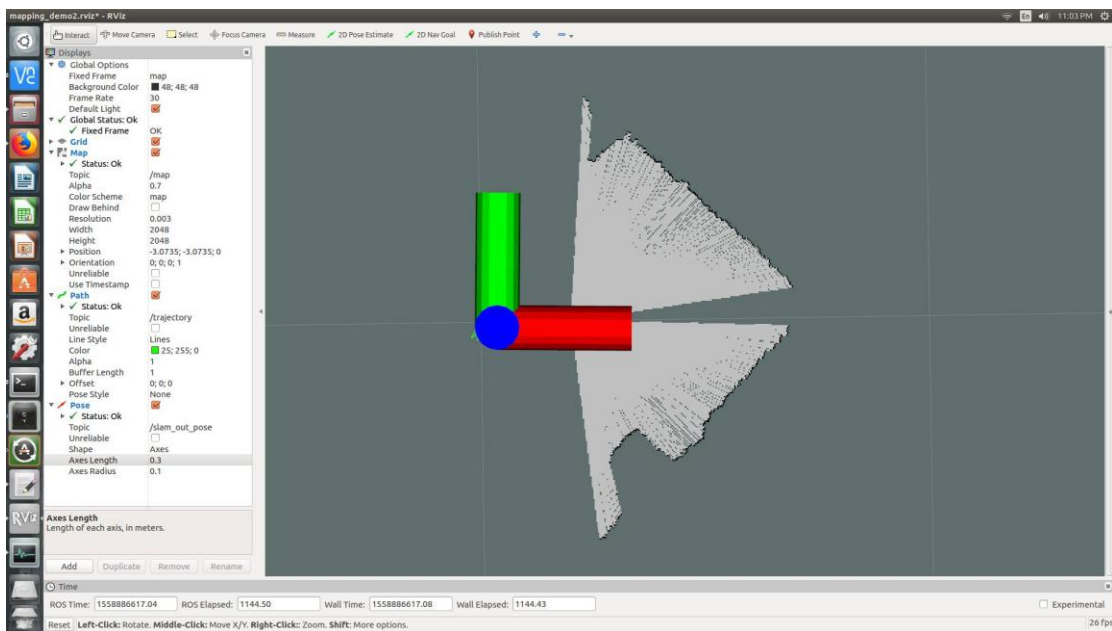
การแสดงผลค่าจากระบบจะอาศัยโปรแกรม RQT และ RVIZ ซึ่งทำงานร่วมกับ ROS โปรแกรม RQT สามารถใช้ในการแสดงผลค่าต่างๆจาก node ที่กำลังทำงานอยู่ในขณะนั้น ตัวอย่างข้อมูลที่มีการใช้งานบ่อยเช่น ค่าของตัวแปรในแต่ละ topic ที่มีการอัปเดตอยู่ ความถี่ของการอัปเดต ชนิดหรือรูปแบบของตัวแปรของแต่ละ topic และรูปแบบการสื่อสารระหว่างการเชื่อมต่อของแต่ละ node ในระบบ รวมไปถึงสามารถบันทึกค่าเหล่านี้ออกมาใช้งานหรือวิเคราะห์เพิ่มเติมในภายหลังได้อีกด้วย ซึ่งข้อดีของโปรแกรม RQT คือสามารถนำข้อมูลมาแสดงผลแบบ real-time ได้ ผู้ใช้สามารถสังเกตและวิเคราะห์รูปแบบการเปลี่ยนแปลงข้อมูลได้ตลอดเวลา ซึ่งเป็นส่วนช่วยให้การทดลองสะดวกมากขึ้น ตัวอย่างรูปการแสดงผลข้อมูลจากโปรแกรม RQT แสดงในรูปที่ 3.4

โปรแกรม RVIZ สามารถใช้ในการแสดงผลข้อมูลบางชนิด เช่น ข้อมูลตำแหน่งและทิศทาง ข้อมูลเส้นทางการเคลื่อนที่ ข้อมูลแผนที่ ออกมาเป็นรูปภาพในลักษณะ 2 มิติ หรือ 3 มิติ ทำให้ผู้ใช้สามารถสังเกตรูปแบบหรือลักษณะหน้าตาของค่าข้อมูลเหล่านั้นได้ง่ายขึ้น ตัวอย่างการแสดงผลข้อมูลด้วยโปรแกรม RVIZ จากงานวิจัยแสดงในรูปที่ 3.5

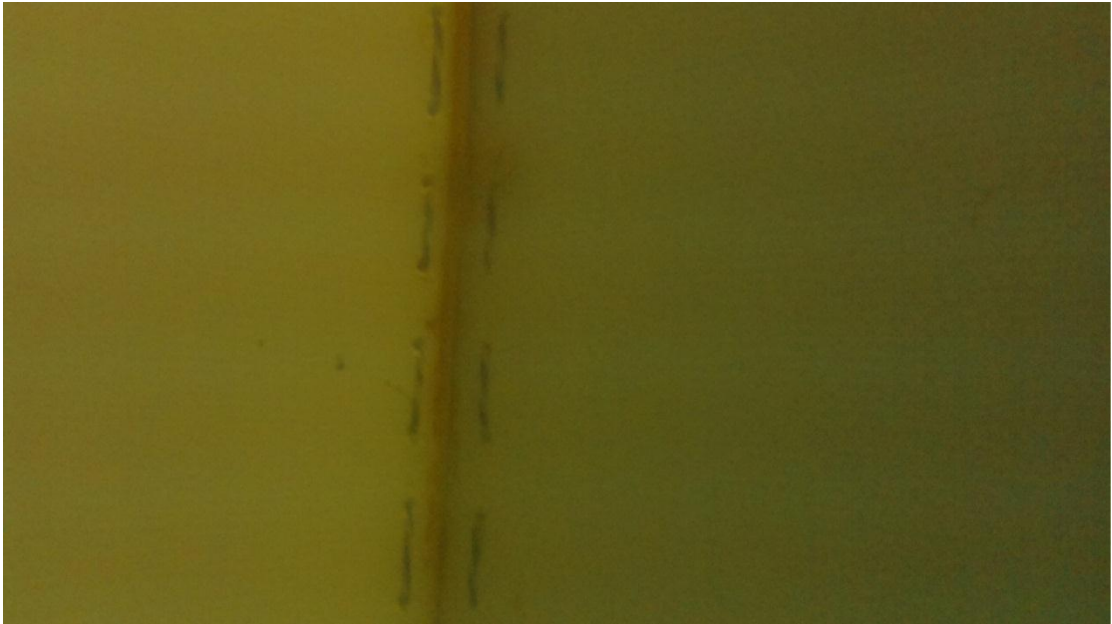
การแสดงผลภาพจากกล้อง Raspberry Pi camera จะอาศัยโปรแกรมที่เขียนอยู่บนบอร์ด Raspberry Pi ด้วยภาษา Python โดยข้อมูลที่ได้ถูกแสดงเป็นวิดีโอแบบ real-time แต่ทั้งนี้ ภาพที่ได้จากการถ่ายได้น้ำจะค่อนข้างมืดและมีมุมมองที่จำกัด ตัวอย่างข้อมูลภาพแสดงในรูปที่ 3.6



รูปที่ 3.4 การแสดงผลข้อมูลจากโปรแกรม RQT



รูปที่ 3.5 การแสดงผลข้อมูลจากโปรแกรม RVIZ

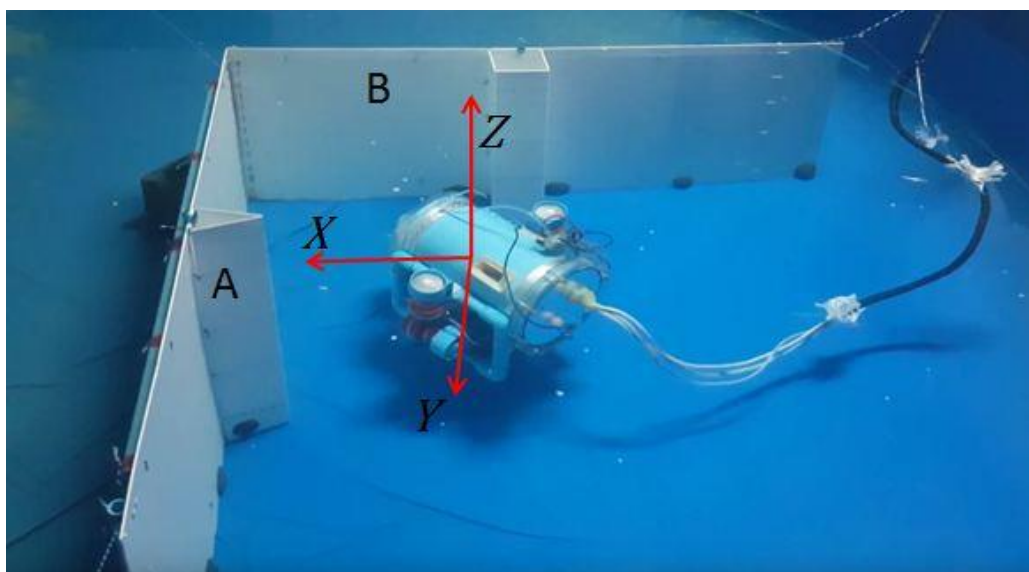


รูปที่ 3.6 ภาพถ่ายที่ได้จากหุ่นยนต์ใต้น้ำที่พัฒนาขึ้น

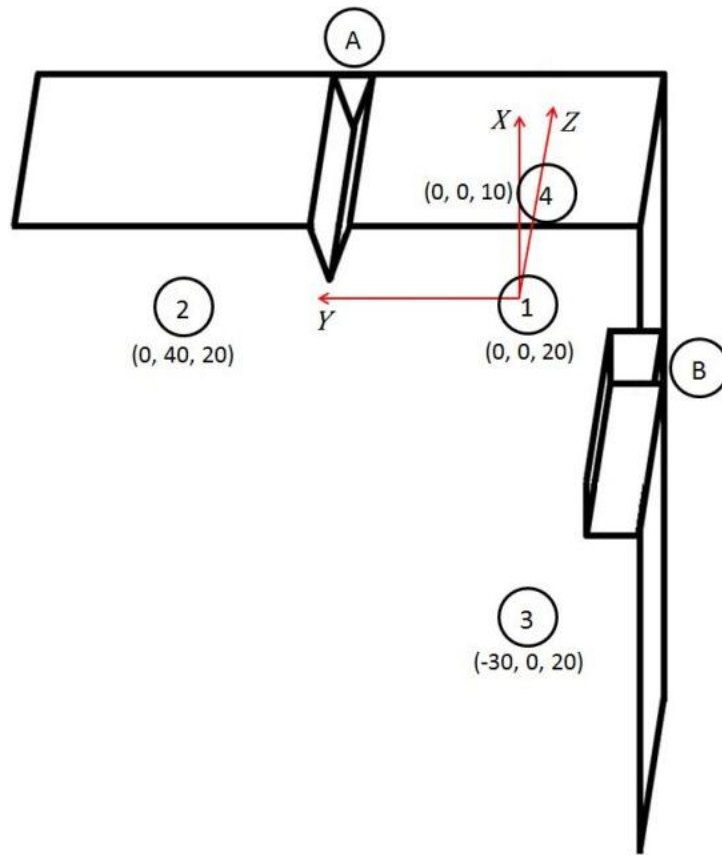
บทที่ 4

การทดลอง

สระน้ำฟ้าใบทรงกระบอก ลึก 70 เซนติเมตรถูกใช้เป็นสถานที่ทดสอบงานวิจัยนี้ ภายในสระมีผนังจำลองลักษณะแสดงดังรูปที่ 4.1 หุ่นยนต์ขณะทดสอบระบบระบุตำแหน่งและทรงตัวอยู่ระหว่างมุมผนังจำลอง ลักษณะการทดลองเป็นการจำลองการใช้หุ่นยนต์สำรวจโครงสร้างใต้น้ำ วัตถุประสงค์ของการทดลองคือ เพื่อทดสอบความสามารถของระบบรักษาตำแหน่งและระบบระบุตำแหน่งของหุ่นยนต์ขณะทำงานใต้น้ำ แผนผังสถานที่ทดสอบในมุมมองด้านบนแสดงดังรูปที่ 4.2 จุดเริ่มต้นของหุ่นยนต์อยู่ในบริเวณที่ 1 เริ่มต้นโดยการควบคุมหุ่นยนต์ให้รักษาทิศทางตั้งฉากกับผนัง A มีค่ามุม yaw เป็น 0 และรักษาระดับความลึก (depth) อยู่ที่ 20 เซนติเมตร จากนั้นควบคุมให้หุ่นยนต์มาใกล้ตำแหน่งหมายเลข 1 แล้วเริ่มการทำงานของ hector_slam ตำแหน่งของหุ่นยนต์ขณะเริ่มต้นนี้เป็นจุดเริ่มต้นของกรอบอ้างอิง fixed frame ตามรูปที่ 4.2 ความถี่ของข้อมูลตำแหน่งในแนวแกน X และ Y มีค่าประมาณ 10 เฮิรตซ์ ความถี่ของข้อมูลความลึกและทิศทางมุม yaw มีค่าประมาณ 20 เฮิรตซ์ ขณะทดสอบความเร็วเป้าหมายในแนวแกน X และ Y ถูกจำกัดอยู่ที่ไม่เกิน ± 5 เซนติเมตรต่อวินาที ในแนว depth ไม่เกิน ± 4.5 เซนติเมตรต่อวินาที และความเร็วเชิงมุมในทิศมุม yaw ไม่เกิน ± 12 องศาต่อวินาที



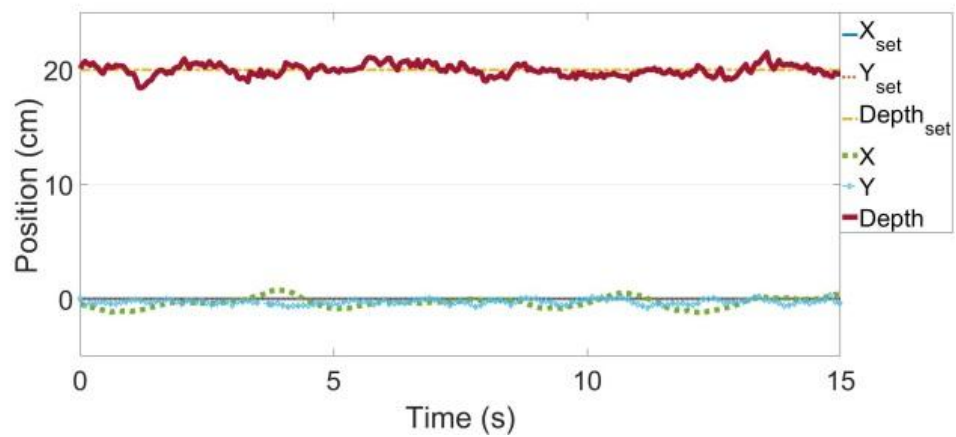
รูปที่ 4.1 การรักษาตำแหน่งของหุ่นยนต์ขณะทดสอบกับผนังใต้น้ำ



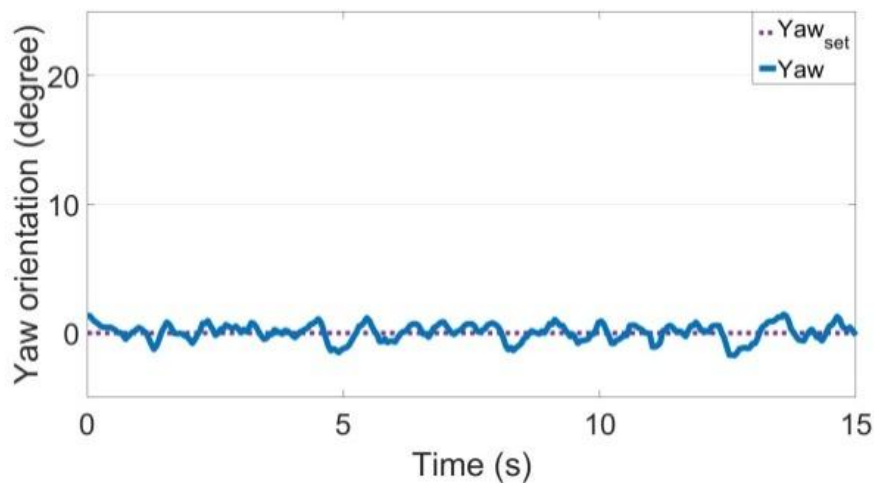
รูปที่ 4.2 ตำแหน่งในการทดสอบหุ่นยนต์และกรอบอ้างอิงเทียบกับผนัง

4.1 การรักษาดำแหน่ง

เริ่มการทดสอบรักษาดำแหน่งที่ตำแหน่งหมายเลข 1 พิกัดเป้าหมาย [X, Y, Depth, yaw] เท่ากับ [0, 0, 20, 0] ตามลำดับ ผลการทดสอบแสดงดังรูปที่ 4.3 และ 4.4 ผลการทดสอบแสดงให้เห็นว่าหุ่นยนต์สามารถรักษาดำแหน่งและทิศทางเป้าหมายได้พร้อมกันทั้ง 4 องศาอิสระ



รูปที่ 4.3 สถานะตำแหน่งของหุ่นยนต์ขณะทดสอบการรักษาตำแหน่งที่ตำแหน่งหมายเลข 1



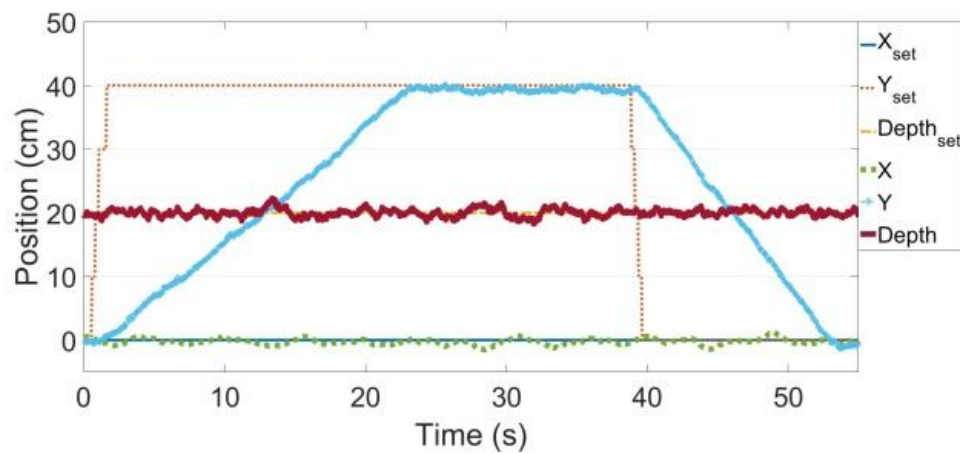
รูปที่ 4.4 สถานะทิศ yaw ของหุ่นยนต์ขณะทดสอบการรักษาตำแหน่งที่ตำแหน่งหมายเลข 1

ตารางที่ 4.1 ตารางแสดงผลการทดสอบรักษาตำแหน่งที่ตำแหน่งหมายเลข 1

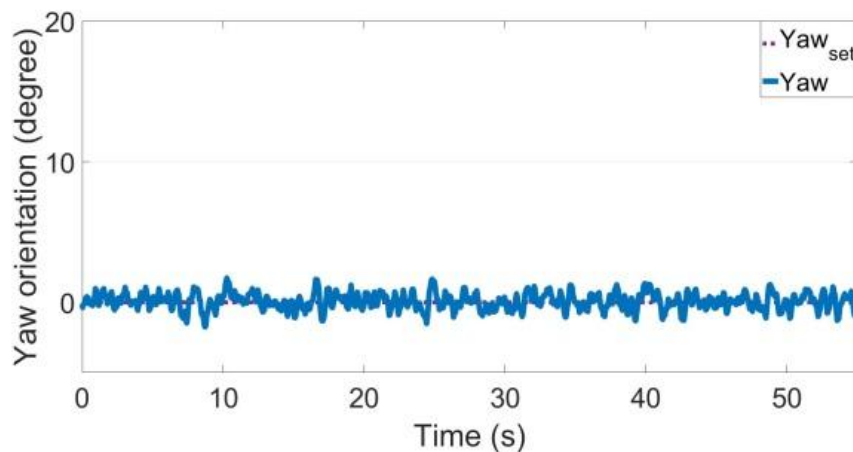
แนวแกน	ค่าเป้าหมาย	ค่าที่วัดได้จากการทดลอง			ความคลาดเคลื่อนสูงสุด
		สูงสุด	ต่ำสุด	เฉลี่ย	
X (cm)	0	0.78	-1.21	-0.33	1.21
Y (cm)	0	0.30	-0.81	-0.29	0.81
Z (cm)	20	21.51	18.42	19.97	1.58
Yaw (degree)	0	1.45	-1.75	0.06	1.75

4.2 การรักษาเส้นทางการเคลื่อนที่ในระนาบ X-Y

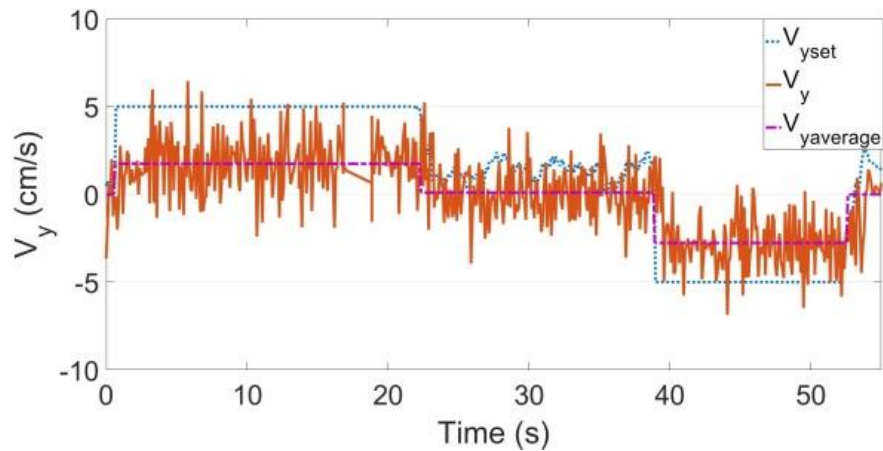
ทำการทดสอบการรักษาตำแหน่งหุ่นยนต์ในเส้นทางขณะเคลื่อนที่เป็นเส้นตรง โดยเปลี่ยนพิกัดเป้าหมายจากตำแหน่งหมายเลข 1 เดิมไปยังตำแหน่งหมายเลข 2 พิกัด [0, 40, 20, 0] รอจนกระทั่งหุ่นยนต์เคลื่อนถึงตำแหน่งเป้าหมายและรักษาตำแหน่งเป็นเวลา 15 วินาที จากนั้นเปลี่ยนพิกัดเป้าหมายกลับมายังจุดที่ 1 และรอจนกระทั่งหุ่นยนต์เคลื่อนที่กลับมา ผลการทดสอบแสดงในรูปที่ 4.5, 4.6 และ 4.7



รูปที่ 4.5 สถานะตำแหน่งของหุ่นยนต์ขณะรักษาเส้นทางการเคลื่อนที่ตามแนวแกน Y



รูปที่ 4.6 สถานะทิศ yaw ของหุ่นยนต์ขณะรักษาเส้นทางการเคลื่อนที่ตามแนวแกน Y



รูปที่ 4.7 สถานะความเร็วแนวแกน Y ของหุ่นยนต์ขณะรักษาเส้นทางการเคลื่อนที่ตามแนวแกน Y

ตารางที่ 4.2 ตารางแสดงผลการทดสอบรักษาเส้นทางการเคลื่อนที่ตามแนวแกน Y

แนวแกน	ค่าเป้าหมาย	ค่าที่วัดได้จากการทดลอง			ความคลาดเคลื่อนสูงสุด
		สูงสุด	ต่ำสุด	เฉลี่ย	
X (cm)	0	1.23	-1.55	-0.26	1.55
Y (cm)	40	40.08	38.79	39.34	1.21
Z (cm)	20	22.32	18.24	19.99	2.32
Yaw (cm)	0	1.73	-1.75	0.07	1.75

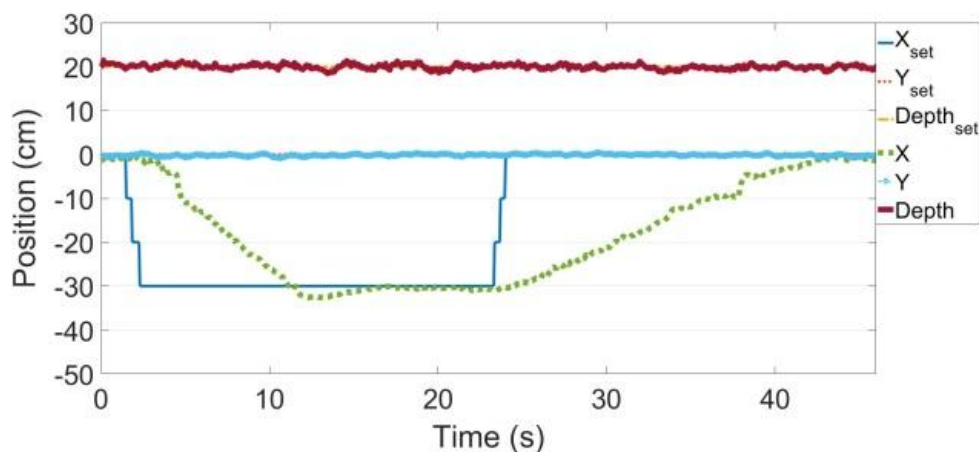
ผลการทดสอบแสดงให้เห็นว่าหุ่นยนต์มีการเปลี่ยนตำแหน่งมุ่งไปยังเป้าหมายโดยเคลื่อนที่เป็นเส้นตรงตามแนวแกน Y และมีอัตราการเปลี่ยนแปลงตำแหน่งตามแนวแกน Y เทียบเวลา (ความชัน) ค่อนข้างคงที่ดังแสดงในรูปที่ 4.5 ส่วนการขับเคลื่อนหุ่นยนต์ไปมีการรักษาเส้นทางการเคลื่อนที่ตามแนวที่กำหนดไว้และยังคงรักษาองศาอิสระอื่นได้แก่ ตำแหน่ง X, Depth (รูปที่ 4.5) และทิศ yaw (รูปที่ 4.6) ให้คงเดิมไว้ได้ ระบบไม่เกิดการ overshoot หลังการเคลื่อนที่ถึงเป้าหมาย

เมื่อพิจารณาความเร็วชั่วขณะของหุ่นยนต์ตามแนวแกน Y ที่เวลาต่างๆ (รูปที่ 4.7) จะสังเกตเห็นว่าความเร็วที่คำนวณได้ในขณะเวลาใดๆ จะมีการเปลี่ยนแปลงมาก เนื่องจากช่วงเวลาที่น่ามาคำนวณหาความเร็วเป็นช่วงเวลาที่สั้นมากตามสมการที่ (3.4) ทำให้ค่าการเปลี่ยน

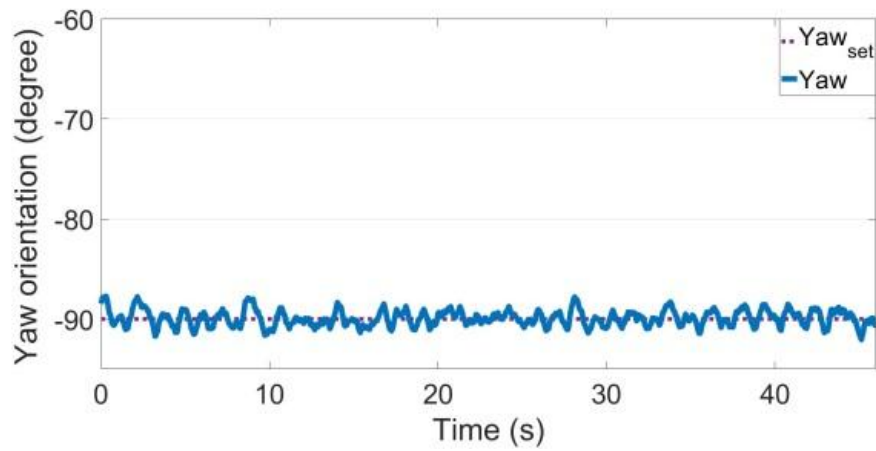
ตำแหน่งต่อเวลาที่มีความผันผวนสูง แต่ภาพรวมการเคลื่อนที่ของหุ่นยนต์ตลอดช่วงการเคลื่อนที่ ยังมีอัตราการเปลี่ยนตำแหน่งที่ค่อนข้างสม่ำเสมอ นั่นคือความชันของข้อมูลตำแหน่ง Y ในรูป 4.5 ใกล้เคียงกับ $V_{Yaverage}$ ในรูปที่ 4.7

ต่อมาเมื่อหุ่นยนต์กลับมาถึงจุดที่ 1 หลังจากรทดสอบก่อนหน้า ทำการทดสอบการเคลื่อนที่ไปยังตำแหน่งหมายเลข 3 โดยเปลี่ยนพิกัดเป้าหมายของมุม yaw ได้พิกัดเป็น $[0, 0, 20, -90]$ เพื่อให้หุ่นยนต์หันหน้าตั้งฉากกับผนัง B ก่อน จากนั้นสั่งการเคลื่อนที่ของหุ่นยนต์ตามแนวแกน X ไปยังตำแหน่งหมายเลข 3 พิกัด $[-30, 0, 20, -90]$ รอจนหุ่นยนต์เคลื่อนที่ถึงเป้าหมายแล้วเปลี่ยนพิกัดกลับมาที่ตำแหน่งหมายเลข 1 พิกัด $[0, 0, 20, -90]$ จากนั้นเปลี่ยนพิกัดกลับมาเป็น $[0, 0, 20, 0]$ เพื่อให้หุ่นยนต์หันหน้าตั้งฉากกับผนัง A เหมือนตอนเริ่มต้น เหตุผลที่กำหนดให้หุ่นยนต์เคลื่อนที่โดยหันหน้าตั้งฉากกับผนังเพราะเป็นจำลองการเคลื่อนที่ในลักษณะการสำรวจสิ่งก่อสร้างหรือโครงสร้างใต้น้ำ

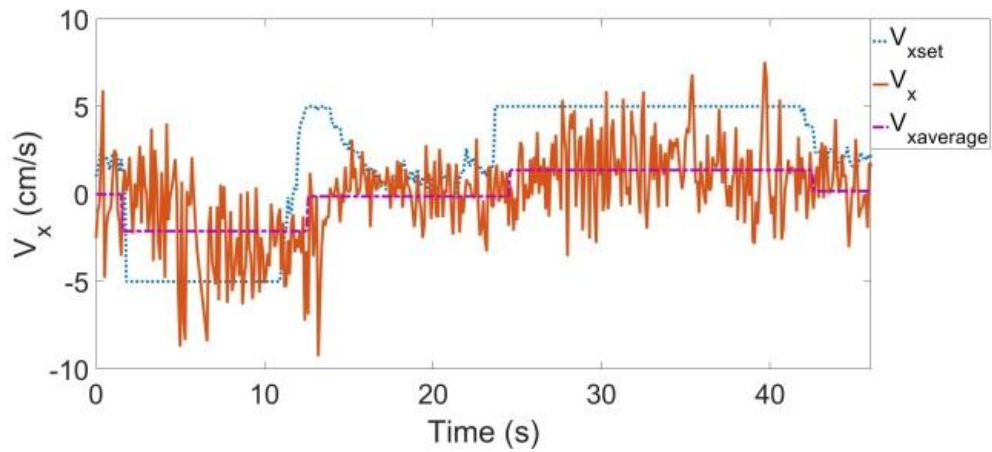
ผลการทดลองแสดงให้เห็นว่าหุ่นยนต์สามารถรักษาเส้นทางการเคลื่อนที่ตามแนวแกน X ได้ (รูปที่ 4.8) อีกทั้งยังสามารถปรับและรักษาทิศ yaw ไว้ได้ (รูปที่ 4.9 และ 4.11) ในทำนองเดียวกันกับการทดลองก่อนหน้า ความเร็วเชิงเส้นชั่วขณะ V_x (รูปที่ 4.10) มีความผันผวนสูงด้วยเหตุผลของการคำนวณการเปลี่ยนตำแหน่งในช่วงเวลาสั้นๆ แต่ความเร็วเฉลี่ยตลอดการเคลื่อนที่ $V_{xaverage}$ ยังสอดคล้องกับความชันของการเปลี่ยนตำแหน่งในรูปที่ 4.8



รูปที่ 4.8 สถานะตำแหน่งของหุ่นยนต์ขณะรักษาเส้นทางการเคลื่อนที่ตามแนวแกน X



รูปที่ 4.9 สถานะทิศ yaw ของหุ่นยนต์ขณะรักษาเส้นทางการเคลื่อนที่ตามแนวแกน X

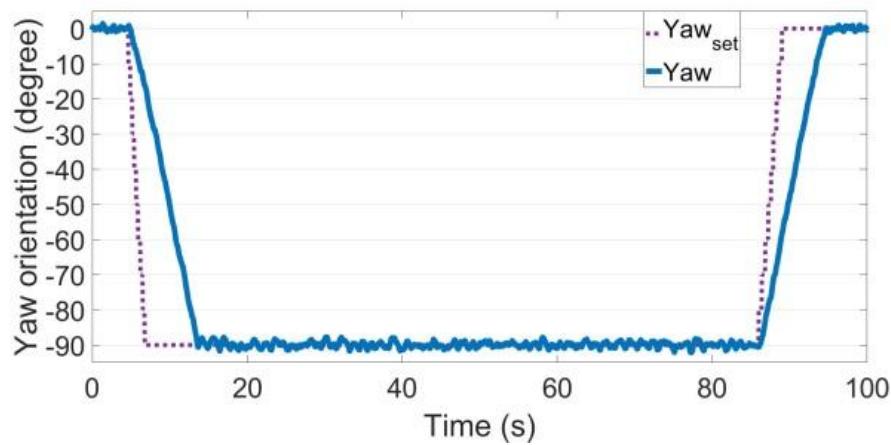


รูปที่ 4.10 สถานะความเร็วแนวแกน X ของหุ่นยนต์ขณะรักษาเส้นทางการเคลื่อนที่ตามแนวแกน X

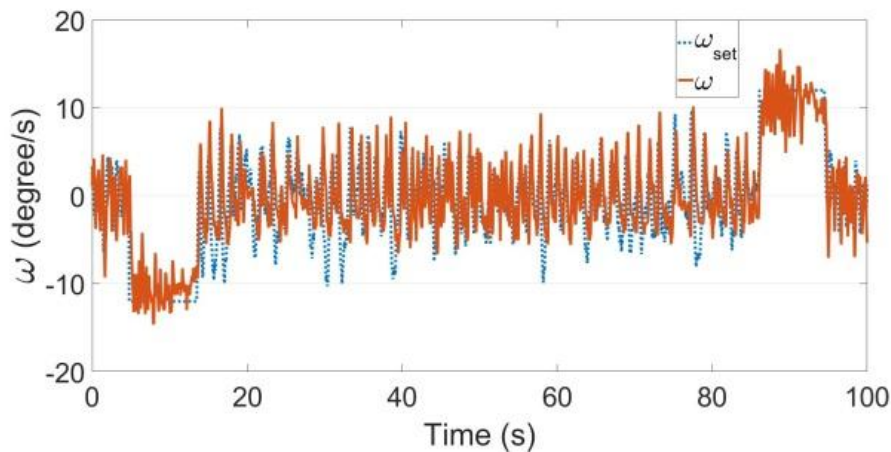
ตารางที่ 4.3 ตารางแสดงผลการทดสอบรักษาเส้นทางการเคลื่อนที่ตามแนวแกน X

แนวแกน	ค่าเป้าหมาย	ค่าที่วัดได้จากการทดลอง			ความคลาดเคลื่อนสูงสุด
		สูงสุด	ต่ำสุด	เฉลี่ย	
X (cm)	-30	-29.89	-32.67	-30.97	2.67
Y (cm)	0	0.43	-0.84	-0.16	0.84
Z (cm)	20	21.51	18.44	19.99	1.56
Yaw (degree)	-90	-87.72	-92.16	-89.94	2.28

เมื่อพิจารณาความสามารถของระบบควบคุมของการรักษาทิศทาง yaw จะเห็นได้ว่าระบบตอบสนองได้เร็วเมื่อมีคำสั่งการเปลี่ยนแปลงค่าเชิงมุม Yaw_{set} ดังรูปที่ 4.11 ดีกว่าการตอบสนองเมื่อเคลื่อนที่แบบเชิงเส้นเพราะในการเคลื่อนที่เชิงมุมมีแรงขับจาก thruster ถึง 4 ตัว ในขณะที่การเคลื่อนที่เชิงเส้นในแต่ละทิศทางมีแรงขับจาก thruster เพียง 2 ตัวเท่านั้น ทำให้ระบบสามารถควบคุมการเคลื่อนที่เชิงมุมได้ง่ายกว่าการเคลื่อนที่เชิงเส้น ส่วนสถานะความเร็วเชิงมุม ω ก็ตอบสนองทันกับการเปลี่ยนแปลงของ ω_{set} และติดตามการเปลี่ยนแปลงได้ตลอดการเคลื่อนที่ดังแสดงในรูปที่ 4.12



รูปที่ 4.11 สถานะทิศ yaw ของหุ่นยนต์ขณะรักษาทิศทางในการเคลื่อนที่ตามแนวแกน X



รูปที่ 4.12 สถานะความเร็วเชิงมุมทิศ yaw ของหุ่นยนต์ขณะรักษาทิศทางในการเคลื่อนที่ตามแนวแกน X

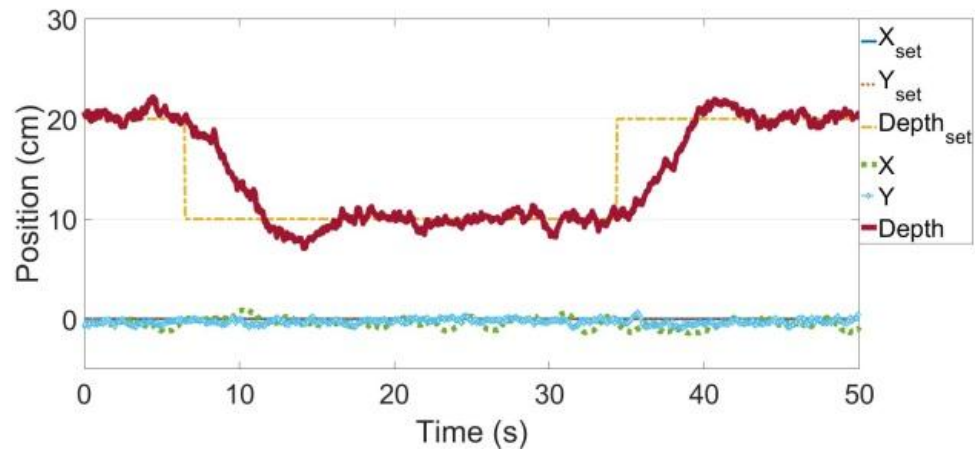
ตารางที่ 4.4 ตารางแสดงผลการทดสอบรักษาทิศทางในการเคลื่อนที่ตามแนวแกน X

แนวแกน	ค่าเป้าหมาย	ค่าที่วัดได้จากการทดลอง			ความคลาดเคลื่อนสูงสุด
		สูงสุด	ต่ำสุด	เฉลี่ย	
X (cm)	0	1.23	-2.33	-1.20	2.33
	-30	-29.89	-32.67	-30.97	2.67
Y (cm)	0	0.43	-0.84	-0.16	0.84
Z (cm)	20	21.51	18.44	19.99	1.56
Yaw (degree)	-90	-87.72	-92.16	-89.94	2.28

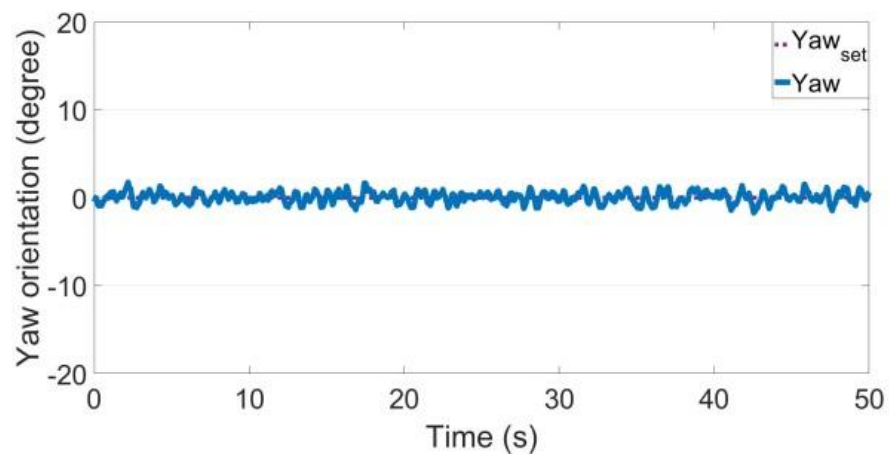
4.3 การควบคุมระดับความลึก

ต่อมา ทำการทดสอบการเคลื่อนที่ไปยังตำแหน่งหมายเลข 4 โดยทำการเปลี่ยนพิกัดเป้าหมายตามแนว Depth ที่พิกัด [0, 0, 10, 0] รอจนหุ่นยนต์เคลื่อนที่ถึงเป้าหมายแล้วเปลี่ยนพิกัดกลับมายังตำแหน่งเริ่มต้น [0, 0, 20, 0] ผลการทดสอบแสดงดังรูปที่ 4.13, 4.14 และ 4.15

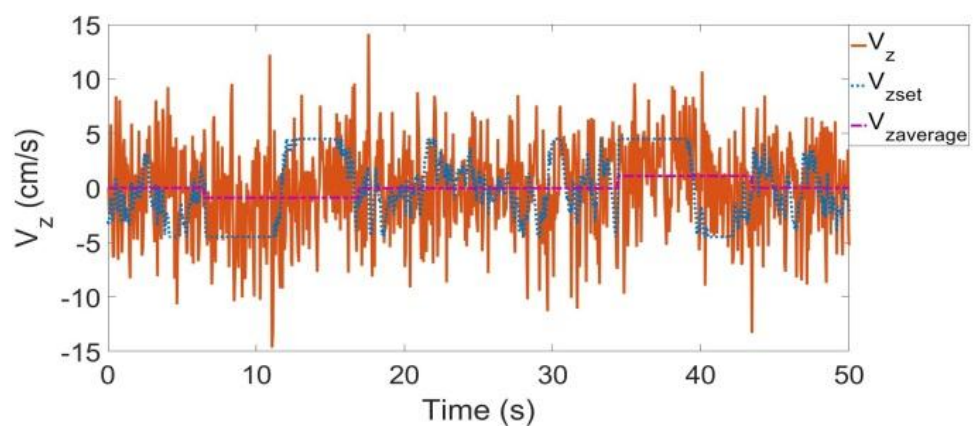
จากผลการทดลองการเปลี่ยนแปลงตำแหน่งเป้าหมายในแนว Depth หรือการเปลี่ยนแปลงระดับความลึก พบว่าหุ่นยนต์สามารถปรับระดับความลึกของตัวเองไปที่ตำแหน่งหมายเลข 4 แล้วเคลื่อนที่กลับมายังตำแหน่งหมายเลข 1 ได้ ซึ่งขณะเคลื่อนที่ตามแนว Depth หุ่นยนต์สามารถรักษาตำแหน่งในแนวแกน X, Y ดังแสดงในรูปที่ 4.13 และทิศ yaw ให้คงที่ได้ ดังแสดงในรูปที่ 4.14 ข้อมูลความเร็ว V_z ชั่วขณะ (รูปที่ 4.15) มีความผันผวนสูงมากรวมทั้งการเปลี่ยนแปลงตำแหน่งตามแนว Depth ไม่สม่ำเสมอและเส้นทางการเคลื่อนที่ที่เกิด overshoot มากที่สุดเมื่อเทียบกับการเคลื่อนที่ในแนวแกนอื่น แต่ความเร็วเฉลี่ย $V_{z\text{average}}$ (รูปที่ 4.15) ก็ยังแสดงความสอดคล้องกับอัตราการเปลี่ยนตำแหน่งตามแนว Depth (รูปที่ 4.13) สาเหตุของการเปลี่ยนตำแหน่งตามแนว Depth ที่ไม่สม่ำเสมอ เพราะเซนเซอร์วัดความดันที่ใช้มีความละเอียดในการวัดระดับความลึกน้อยกว่าเซนเซอร์ Lidar และ IMU ทำให้ความละเอียดของค่าความลึกที่วัดได้น้อยกว่าเมื่อเทียบกับความละเอียดของค่าตำแหน่ง X, Y และมุม yaw ซึ่งเมื่อมีความละเอียดน้อยกว่า ช่วงความคลาดเคลื่อนที่เกิดขึ้นก็จะสูงกว่าจึงเป็นสาเหตุให้ผลการควบคุมการเคลื่อนที่ตามแนว Depth ไม่สม่ำเสมอเมื่อเทียบกับผลการเคลื่อนที่ในทิศทางอื่น ถึงกระนั้นก็ตาม ระบบควบคุมนี้ก็ยังสามารที่จะประมวลผลและสั่งการหุ่นยนต์ให้ทำงานรักษาระดับความลึกได้ตามเป้าหมายที่เปลี่ยนไป ค่า gain \tilde{K}_p และ \tilde{K}_i รวมทั้งตัวปรับค่า \tilde{K} แสดงในตารางที่ 4.6 เส้นทางการเคลื่อนที่จากการทดลองทั้งหมดแสดงเป็นรูปแบบ 3 มิติในรูปที่ 4.16



รูปที่ 4.13 สถานะตำแหน่งของหุ่นยนต์ขณะรักษาเส้นทางการเคลื่อนที่ตามแนว Depth



รูปที่ 4.14 สถานะทิศ yaw ของหุ่นยนต์ขณะรักษาเส้นทางการเคลื่อนที่ตามแนว Depth



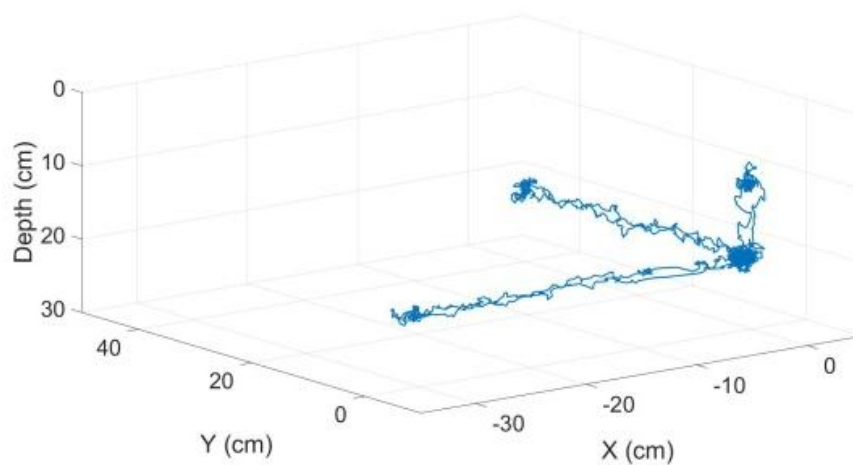
รูปที่ 4.15 สถานะความเร็วแนว Depth ของหุ่นยนต์ขณะรักษาเส้นทางการเคลื่อนที่ตามแนว Depth

ตารางที่ 4.5 ตารางแสดงผลการทดสอบรักษาเส้นทางการเคลื่อนที่ตามแนวแกน Y

แนวแกน	ค่าเป้าหมาย	ค่าที่วัดได้จากการทดลอง			ความคลาดเคลื่อนสูงสุด
		สูงสุด	ต่ำสุด	เฉลี่ย	
X (cm)	0	0.99	-1.42	-0.44	1.42
Y (cm)	0	0.64	-0.90	-0.28	0.90
Z (cm)	10	11.35	7.07	9.71	2.93
Yaw (degree)	0	1.77	-1.72	0.06	1.77

ตารางที่ 4.6 ตารางแสดงค่า gain และตัวปรับค่าที่ใช้ในการทดลอง

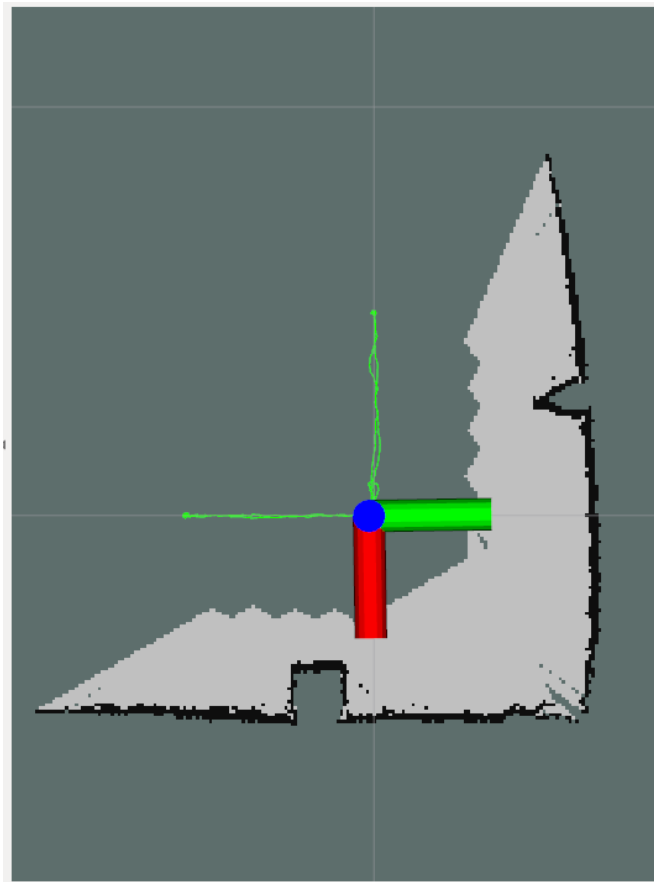
แนวแกน	\tilde{K}	\tilde{K}_p	\tilde{K}_i
X	3.0	3.0	1.5
Y	3.0	4.0	1.5
Z	4.0	5.0	1.0
Yaw	4.5	2.5	1.0



รูปที่ 4.16 เส้นทางการเคลื่อนที่แบบ 3 มิติ จากการทดลองทั้งหมด

4.4 การสร้างแผนที่ใต้น้ำ

กระบวนการสร้างแผนที่ใต้น้ำเริ่มต้นเมื่อเริ่มการทำงานของโปรแกรม `hector_slam` ในช่วงต้นของการทดลองและดำเนินการไปตลอดการทดลอง โดย `hector_slam` จะนำข้อมูลการสแกนระยะจาก Lidar มาประมวลผลหาตำแหน่งของหุ่นยนต์พร้อมทั้งวาดแผนที่ของสิ่งแวดล้อมในเวลาเดียวกัน แผนที่ที่สามารถถูกเปิดอ่านระหว่างการทดลองได้ด้วยโปรแกรม `rviz` แผนที่ที่สร้างจากการทดลองนี้แสดงดังรูปที่ 4.17



รูปที่ 4.17 แผนที่สิ่งแวดล้อมและเส้นทางการเคลื่อนที่ของหุ่นยนต์ที่ได้จากการทดสอบ

จะสังเกตเห็นว่าแผนที่ที่สร้างขึ้นจากข้อมูลเซนเซอร์วัดระยะ Lidar มีความคลาดเคลื่อนของแนวผนังที่ได้มาเป็นเส้นโค้งเป็นเพราะว่าในบางขณะที่หุ่นยนต์เคลื่อนที่ เซนเซอร์ Lidar ที่ติดตั้งอยู่บนตัวหุ่นยนต์อยู่ห่างจากผนังเกินช่วงความสามารถที่จะทำงานใต้น้ำได้อย่างถูกต้อง รวมไปถึงอัลกอริทึม `hector_slam` ที่ใช้วิธีการประมาณตำแหน่งและทิศทางด้วยวิธี `scan matching` ซึ่งวิธีการนี้เหมาะสำหรับระบบที่ไม่มีข้อมูลตำแหน่งสัมบูรณ์ (absolute position) แต่ก็มีข้อจำกัดคือ จะเกิดการสะสมของความคลาดเคลื่อนจากการประมาณค่าตำแหน่งและทิศทาง

โดยจะปรากฏชัดยิ่งขึ้นตามลำดับการสร้างนับจากจุดเริ่มต้นการทำงาน [17] ส่วนเส้นทางการเคลื่อนที่ในระนาบ X-Y (สีเขียว) ในรูปที่ 4.17 ยังคงมีลักษณะเป็นเส้นตรงเนื่องจากการทดลองเป็นการเคลื่อนที่เทียบกับกรอบอ้างอิงของ hector_slam ข้อมูลตำแหน่งที่ได้จึงเป็นแบบสัมพัทธ์เทียบกับกรอบอ้างอิงดังกล่าว ผลการทดสอบที่แสดงเส้นทางการเคลื่อนที่เทียบกับกรอบอ้างอิงของ hector_slam จึงมีลักษณะเป็นเส้นตรง

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

ระบบควบคุมเสถียรภาพของหุ่นยนต์ดำน้ำที่พัฒนาขึ้นสามารถรักษาการทรงตัวของหุ่นยนต์ได้ ระบบระบุตำแหน่งสามารถทำงานได้ต่อเนื่อง สามารถระบุตำแหน่งหุ่นยนต์แบบเวลาจริงได้ โดยมีความคลาดเคลื่อนของข้อมูลน้อยกว่า ± 1 เซนติเมตรในแนวแกน X, Y, Z และน้อยกว่า ± 1 องศาในทิศ yaw ซึ่งเพียงพอสำหรับการนำไปประมวลผลในระบบรักษาตำแหน่งหรือเส้นทางการเคลื่อนที่ ผลการทดสอบระบบรักษาตำแหน่ง พบว่า หุ่นยนต์ให้ผลตอบสนองต่อการเปลี่ยนแปลงตำแหน่งที่สภาวะ steady state ด้วยความคลาดเคลื่อนน้อยกว่า ± 3 เซนติเมตร และความคลาดเคลื่อนของตำแหน่งเชิงมุมน้อยกว่า ± 3 องศา ขณะหุ่นยนต์เคลื่อนที่ ระบบควบคุมสามารถรักษาเส้นทางการเคลื่อนที่ในแนวเส้นตรงได้ทั้งในแนวระนาบและแนวตั้ง โดยชนิดของความคลาดเคลื่อนของตำแหน่งที่วัดได้ในการทดลองขึ้นกับจุดอ้างอิงตั้งนี้ ความคลาดเคลื่อนของตำแหน่งตามแนวความลึกเป็นแบบสมบรูณ์เทียบกับระดับผิวน้ำ ความคลาดเคลื่อนเชิงมุม yaw เป็นแบบสมบรูณ์เทียบกับสนามแม่เหล็กโลก ความคลาดเคลื่อนของตำแหน่งตามแกน X และแกน Y เป็นแบบสัมพันธ์เทียบกับตำแหน่งของหุ่นยนต์เมื่อระบบควบคุมเริ่มทำงาน

อย่างไรก็ตาม รายละเอียดของสิ่งแวดล้อมต้องเพียงพอที่จะนำไปใช้ประมวลผลในระบบระบุตำแหน่งได้ รวมไปถึงความเร็วในการเคลื่อนที่ก็เป็นปัจจัยสำคัญที่มีผลต่อความแม่นยำในการระบุตำแหน่ง เพราะการประมวลผลของระบบระบุตำแหน่งอาจทำงานไม่ทันกับตำแหน่งที่เปลี่ยนไป ทำให้การระบุตำแหน่งเกิดความคลาดเคลื่อนขึ้นได้

ระบบควบคุมตำแหน่งที่พัฒนาขึ้นถึงแม้จะมีข้อจำกัดเรื่องความแม่นยำหากหุ่นยนต์เคลื่อนเร็ว แต่ก็ยังสามารถนำไปใช้เสริมการทำงานของหุ่นยนต์ที่ไม่เคยมีระบบระบุตำแหน่งและระบบทรงตัวได้ ด้วยการเปิดใช้ระบบควบคุมการรักษาตำแหน่งเมื่อหุ่นยนต์อยู่ในตำแหน่งที่ต้องการหรือเคลื่อนที่สำรวจช้า แล้วเปลี่ยนเป็นการบังคับจากผู้ควบคุมหากต้องการเคลื่อนที่เร็ว รวมไปถึงข้อมูลที่ได้จากการวิจัยสามารถนำไปประยุกต์ใช้ในการเพิ่มประสิทธิภาพการทำงานของระบบระบุตำแหน่งหรือระบบทรงตัวในหุ่นยนต์ดำน้ำแบบสามองศาอิสระให้เป็นระบบควบคุมแบบสององศาอิสระโดยเพิ่มเซนเซอร์รองรับตามความเหมาะสม รวมทั้งสามารถนำไปต่อยอดเป็นระบบควบคุมสำหรับหุ่นยนต์ใต้น้ำอัตโนมัติได้

5.2 ปัญหาที่พบในงานวิจัย

1. การใช้งานเซนเซอร์ Lidar ได้นำมีข้อจำกัดเนื่องจากคุณสมบัติของน้ำที่ดูดกลืนแสงสีแดงทำให้ระยะการทำงานของ Lidar ลดลงและหากน้ำขุ่นหรือมีตะกอนอาจจะส่งผลให้เซนเซอร์ทำงานไม่ได้ ส่งผลให้ระบบระบุตำแหน่งยังมีข้อจำกัดในการใช้งาน

2. ชนิดและจำนวนของสายไฟและสายเคเบิลที่ใช้เชื่อมต่อระหว่างหุ่นยนต์กับ GCS ในงานวิจัย ยังไม่เหมาะสมสำหรับใช้กับหุ่นยนต์ดำน้ำขนาดเล็ก เนื่องจากมีความแข็งตึงสูงส่งผลกระทบต่อ การควบคุมการเคลื่อนที่

5.3 แนวทางแก้ไข

1. เปลี่ยนชนิดของเซนเซอร์สำหรับวัดระยะในแนวระนาบจาก Lidar ไปเป็น sonar แทนเพื่อแก้ปัญหาเรื่องข้อจำกัดสำหรับการใช้งานใต้น้ำ

2. ลดจำนวนสายไฟและสายเคเบิลที่เชื่อมต่อระหว่างหุ่นยนต์และ GCS หรือเปลี่ยนชนิดของสายไฟที่ใช้ไปเป็นชนิดที่ออกแบบมาเป็นพิเศษสำหรับหุ่นยนต์ดำน้ำ ซึ่งสายไฟชนิดดังกล่าวออกแบบมาให้มีความหนาแน่นใกล้เคียงกับน้ำและมีความแข็งตึงต่ำ ทำให้ผลกระทบต่อ การควบคุมการเคลื่อนที่ของหุ่นยนต์น้อยลง

5.4 ข้อเสนอแนะและงานในอนาคต

สำหรับการพัฒนาต่อยอดงานวิจัยสามารถทำได้โดยเน้นพัฒนาส่วนประกอบต่าง ๆ ของหุ่นยนต์และระบบให้มีประสิทธิภาพมากขึ้น สามารถแบ่งได้เป็น

1. ส่วนโครงสร้าง สามารถพัฒนาให้มีความแข็งแรงและทนทานต่อความดันได้มากขึ้น เพิ่มขีดจำกัดให้หุ่นยนต์สามารถทำงานที่ระดับความลึกมากขึ้นได้

2. ส่วนอุปกรณ์อิเล็กทรอนิกส์ สามารถปรับเปลี่ยนเซนเซอร์ต่าง ๆ ไปเป็นชนิดที่คุณภาพสูงขึ้น เพื่อให้ข้อมูลที่ได้มีความคลาดเคลื่อนและสัญญาณรบกวนน้อยลง หรืออาจเพิ่มเซนเซอร์ชนิดใหม่ที่สามารถวัดข้อมูลที่เป็นประโยชน์ต่อการประมวลผลของระบบ เช่น DVL หรือเซนเซอร์วัดระยะแบบ 3 มิติ รวมทั้งการเปลี่ยนไปใช้ thruster ชนิดที่มีคุณภาพสูงขึ้น ให้แรงขับได้สม่ำเสมอและสามารถทนความดันได้มากขึ้นกว่าเดิม

3. ส่วนโปรแกรม สามารถพัฒนาระบบประมวลผลตำแหน่งและความเร็วโดยประยุกต์ใช้วิธีการที่สามารถทำนายผลลัพธ์ของระบบ เช่น อัลกอริทึมแบบคาลมาน ซึ่งจะช่วยให้เพิ่ม

ประสิทธิภาพให้ระบบควบคุมสามารถตอบสนองต่อผลลัพธ์จากระบบได้เร็วขึ้น รวมไปถึงโปรแกรมที่นำมาใช้ประมาณตำแหน่งและสร้างแผนที่ในระนาบ 2 มิติ ก็สามารถปรับเปลี่ยนไปใช้โปรแกรมที่สามารถทำงานแบบ 3 มิติได้หากมีข้อมูลเพียงพอ

บรรณานุกรม

- [1] V. N. Kuhn, P. L. J. Drews, S. C. P. Gomes, M. A. B. Cunha, and S. S. da C. Botelho, "Automatic control of a ROV for inspection of underwater structures using a low-cost sensing," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 37, no. 1, pp. 361–374, Jan. 2015.
- [2] S. Timpitak and P. Prempraneerach, "Remotely Operated Vehicle with Depth Control," presented at the The 3rd TSME International Conference on Mechanical Engineering, Chiang Rai, Thailand, 2012.
- [3] A. Z. Abidin, R. Mardiyanto, and D. Purwanto, "Implementation of PID controller for hold altitude control in underwater remotely operated vehicle," in *International Seminar on Intelligent Technology and Its Applications (ISITIA)*, Lombok, Indonesia, 2016, pp. 665–670.
- [4] G. C. Karras and K. J. Kyriakopoulos, "Localization of an underwater vehicle using an IMU and a laser-based vision system," in *Mediterranean Conference on Control Automation*, Athens, Greece, 2007, pp. 1–6.
- [5] H.-W. Hsieh, C.-L. Lee, and C.-L. Kuo, "Localization of an underwater robot with inertial sensor fusion models," in *2010 5th IEEE Conference on Industrial Electronics and Applications*, Taichung, Taiwan, 2010, pp. 1562–1567.
- [6] H. Kondo and T. Ura, "Navigation of an AUV for investigation of underwater structures," *Control Engineering Practice*, vol. 12, no. 12, pp. 1551–1559, Dec. 2004.
- [7] D. Scaradozzi, L. Gambella, S. M. Zanoli, and G. Conte, "Acoustic mapping and localization of an rov," in *14th Mediterranean Conference on Control and Automation*, Ancona, Italy, 2006, pp. 1–6.
- [8] C. M. Clark, C. S. Olstad, K. Buhagiar, and T. Gambin, "Archaeology via underwater robots: Mapping and localization within maltese cistern systems," in *Robotics and Vision 2008 10th International Conference on Control, Automation*, Hanoi, Vietnam, 2008, pp. 662–667.
- [9] J. Snyder, "Doppler Velocity Log (DVL) navigation for observation-class ROVs," in *OCEANS 2010 MTS/IEEE SEATTLE.*, Seattle, WA, USA, 2010, pp. 1–9.
- [10] Z. Eskinja, Z. Fabekovic, and Z. Vukic, "Localization of autonomous underwater vehicles by sonar image processing," in *49th International Symposium ELMAR-2007*, Zadar, Croatia, 2007, pp. 103–106.
- [11] C. Cain and A. Leonessa, "Validation of underwater sensor package using feature based slam," *Sensors (Basel)*, vol. 16, no. 3, pp. 380–408, Mar. 2016.
- [12] F. A. Azis, M. S. M. Aras, M. Z. A. Rashid, M. N. Othman, and S. S. Abdullah, "Problem identification for underwater remotely operated vehicle (rov): a case study," *Procedia Engineering*, vol. 41, pp. 554–560, Jan. 2012.
- [13] W. M. Bessa, M. S. Dutra, and E. Kreuzer, "Depth control of remotely operated underwater vehicles using an adaptive fuzzy sliding mode controller," *Robotics and Autonomous Systems*, vol. 56, no. 8, pp. 670–677, Aug. 2008.
- [14] W. M. Bessa, M. S. Dutra, and E. Kreuzer, "Dynamic positioning of underwater robotic vehicles with thruster dynamics compensation," *International Journal of Advanced Robotic Systems*, vol. 10, no. 9, pp. 325–332, Sep. 2013.

- [15] T. I. Fossen and S. I. Sagatun, “Adaptive control of nonlinear underwater robotic systems,” in *IEEE International Conference on Robotics and Automation Proceedings*, Sacramento, CA, USA, 1991, pp. 1687–1694.
- [16] S. Kohlbrecher, O. von Stryk, J. Meyer, and U. Klingauf, “A flexible and scalable SLAM system with full 3D motion estimation,” in *IEEE International Symposium on Safety, Security, and Rescue Robotics*, Kyoto, Japan, 2011, pp. 155–160.
- [17] A. Diosi and L. Kleeman, “Laser scan matching in polar coordinates with application to SLAM,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Alta, Canada, 2005, pp. 3317–3322.

ภาคผนวก

ก. ข้อมูลจำเพาะของอุปกรณ์ที่ใช้ในการทดลอง

1. บอร์ด Raspberry Pi 2 model B

ตารางที่ ก.1 ข้อมูลจำเพาะของบอร์ด Raspberry Pi 2 model B

SoC	BCM2836
CPU	Quad Cortex A7 @ 900MHz
Instruction set	ARMv7-A
GPU	250 MHz VideoCore IV
RAM	1 GB SDRAM
Storage	Micro-SD
Ethernet	10/100
Video Output	HDMI/Composite
Audio Output	HDMI/Headphone
GPIO	40

[ข้อมูลจาก:<https://hackaday.com/wp-content/uploads/2016/02/pispecs2.png>]

2. บอร์ด Arduino Mega 2560

ตารางที่ ก.2 ข้อมูลจำเพาะของบอร์ด Arduino Mega 2560

Microcontroller	ATmega2560
Operating Voltage	5 V
Input Voltage (recommended)	7-12 V
Input Voltage (Limits)	6-20 V
Digital I/O Pins	54
Analog Input Pins	16
DC current per I/O Pin	40 mA
DC current for 3.3 V Pin	50 mA
Flash Memory	256 KB
SRAM	8 KB
EEPROMK	4 KB
Clock Speed	16 MHz

[ข้อมูลจาก: https://www.researchgate.net/figure/Arduino-Mega-specifications_tbl2_266646630]

3. บอร์ดขับเคลื่อนมอเตอร์ Smile EVO24V9.3

ตารางที่ ก.3 ข้อมูลจำเพาะของบอร์ดขับเคลื่อนมอเตอร์ Smile EVO24V9.3

IC	VNH5019 Full bridge motor drive
ขีดกระแสไฟฟ้าชั่วขณะ	30 A
ขีดกระแสไฟฟ้าต่อเนื่อง	10 A ที่แรงดัน 24 VDC
แรงดันไฟฟ้าอินพุต	7-28 VDC
แรงดันไฟฟ้าเอาต์พุตสูงสุด	0.98 VDC
ความถี่ PWM สูงสุด	10 KHz
อุณหภูมิสูงสุด	100° C
ขนาด PCB	50 x 50 mm.
น้ำหนัก	32 g.

[ข้อมูลจาก: http://www.puaicec.ac.th/UserFiles/files/course1/%บทที่%205%2022_6_61_2.pdf]

4. Raspberry Pi Camera Module

ตารางที่ ก.4 ข้อมูลจำเพาะของ Raspberry Pi Camera Module

Image Sensor	Sony IMX 219 PQ CMOS image sensor in a fixed-focus module.
Resolution	8-megapixel
Still picture resolution	3280 x 2464
Max image transfer rate	1080p: 30 fps (encode and decode) 720p: 60 fps
Connection to Raspberry Pi	15-pin ribbon cable, to the dedicated 15-pin MIPI Camera Serial Interface (CSI-2).
Image control functions	Automatic exposure control Automatic white balance Automatic band filter Automatic 50/60 Hz luminance detection Automatic black level calibration
Temp range	Operating: -20° to 60° Stable image: -20° to 60°
Lens size	1/4"

Dimensions	23.86 x 25 x 9 mm
Weight	3 g

[ข้อมูลจาก: <https://cdn.sparkfun.com/datasheets/Dev/RaspberryPi/RPiCamMod2.pdf>]

5. เซนเซอร์วัดความดัน MPX5700GP

ตารางที่ ก.5 ข้อมูลจำเพาะของเซนเซอร์วัดความดัน MPX5700GP

แรงกดดันในการใช้งาน	102 psi
ประเภทแรงดัน	Gauge
ความแม่นยำ	2.5 %
รูปแบบการติดตั้ง	Through Hole
ประเภทของพอร์ต	Single Radial Barbed
บรรจุภัณฑ์	Unibody 6-pin
การจ่ายแรงดันที่ใช้งาน	5 V
ค่าต่ำสุดของอุณหภูมิในการใช้งาน	- 40° C
ค่าสูงสุดของอุณหภูมิในการใช้งาน	+ 125° C
แรงดันขาออก	4.7 V
ขนาดของพอร์ต	4.93 mm
กระแสไฟฟ้าที่จ่ายใช้งาน	7 mA
การจ่ายแรงดัน (สูงสุด)	5.25 V
การจ่ายแรงดัน (ต่ำสุด)	4.75 V
น้ำหนักต่อหน่วย	3.376 g

[ข้อมูลจาก: <https://th.mouser.com/ProductDetail/NXP-Freescale/MPX5700GP?qs=N2XN0KY4UWUDV5rpYkJCzQ%3D%3D>]

6. เซนเซอร์ IMU GY-87

ตารางที่ ก.6 ข้อมูลจำเพาะของเซนเซอร์ IMU GY-87

ชิป	MPU6050, HMC5883L และ BMP180
Bus	I ² C
ค่าต่ำสุดของอุณหภูมิในการใช้งาน	- 40°C
ค่าสูงสุดของอุณหภูมิในการใช้งาน	+ 85°C
การจ่ายแรงดัน (สูงสุด)	5 V

การจ่ายแรงดัน (ต่ำสุด)	3.3 V
ขนาด	17×22 mm
compass heading accuracy	1° to 2°
gyro full-scale range	±250, ±500, ±1000, and ±2000°/sec (DPS)
accelerometer full-scale range	±2 g, ±4 g, ±8 g, and ±16 g.

[ข้อมูลจาก: <https://robu.in/product/mpu6050hmc5883lbmp180-10dof-3-axis-gyro-3-axis-acceleration-3-axis-magnetic-field-air-pres/>]

7. ปั๊มน้ำ Seaflo bilge pump 1100GPH

ตารางที่ ก.7 ข้อมูลจำเพาะของปั๊มน้ำ Seaflo bilge pump 1100GPH

Model	SFBP1-G1100-01
Pump body	Thermoplastic
Shaft	Stainless Steel
Shaft seal	Lip Seal
Voltage	12 VDC
Flow Rate	1100 GPH / 4100 LPH
Protection	Anti-Airlock and Ignition Protected

[ข้อมูลจาก: <http://boatssa.com/product/1100gph-seaflo-bilge-pump/>]

8. เซนเซอร์ HOKUYO URG-04LX-UG01

ตารางที่ ก.8 ข้อมูลจำเพาะของ เซนเซอร์ HOKUYO URG-04LX-UG01

Power source	5 VDC
Light source	Semiconductor laser diode (785nm) Laser safety class 1
Measuring area	20 to 5600mm (white paper with 70mm×70mm), 240°
Accuracy	60 to 1,000 mm : ±30 mm, 1,000 to 4,095 mm : ±3 of measurement
Angular resolution: Step angle	Approx. 0.36° (360°/1,024 steps)
Scanning time	100 ms/scan
Noise	25dB or less

Interface	USB2.0/1.1[Mini B](Full Speed)
Command System	SCIP Ver.2.0
Ambient illuminance*1	Halogen/mercury lamp: 10,000Lux or less, Florescent: 6000Lux (Max)
Ambient temperature/humidity	-10 to +50°C, 85% or less (Not condensing, not icing)
Vibration resistance	10 to 55 Hz, double amplitude 1.5 mm each 2 hour in X, Y and Z directions
Impact resistance	196 m/s ² , Each 10 time in X, Y and Z directions
Weight	Approx. 160 g

[ข้อมูลจาก: <https://www.robotshop.com/en/hokuyo-urg-04lx-ug01-scanning-laser-rangefinder.html>]

ประวัติผู้เขียน

ชื่อ สกุล นายปรเมศวร์ สุวรรณวงศ์
รหัสประจำตัวนักศึกษา 5710120039
วุฒิการศึกษา

วุฒิ	ชื่อสถาบัน	ปีที่สำเร็จการศึกษา
วิศวกรรมศาสตรบัณฑิต	มหาวิทยาลัยสงขลานครินทร์	2556
วิศวกรรมเมคคาทรอนิกส์		

ทุนการศึกษา (ที่ได้รับในระหว่างการการศึกษา)

1. ทุนศิษย์ก้นกุฏิ ปี 2557
2. ทุนอุดหนุนการวิจัยเพื่อวิทยานิพนธ์ ปี 2558