



Structure Design and Platform Development of Universal Template
for Humanoid Algorithm Interface (UTHAI)

การออกแบบโครงสร้างและพัฒนาระบบพื้นฐานสำหรับหุ่นยนต์อิเล็กทรอนิกส์
เพื่อการศึกษาและวิจัย

นายจิรภูร์ ศรีรัตนอาภรณ์
นายเจษฎากร หาไซวงศ์
นายวุฒิภัทร โชคโน้นทรัพย์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมหุ่นยนต์และระบบอัตโนมัติ
สถาบันวิทยาการหุ่นยนต์ภาควิชานาม
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
ปีการศึกษา 2560



Structure Design and Platform Development of Universal Template
for Humanoid Algorithm Interface (UTHAI)

การออกแบบโครงสร้างและพัฒนาระบบพื้นฐานสำหรับหุ่นยนต์อิเล็กทรอนิกส์
เพื่อการศึกษาและวิจัย

นายจิรภูร์ ศรีรัตนอาภรณ์
นายเจษฎากร หาไซวงศ์
นายวุฒิภัทร โชคอนันตทรัพย์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมหุ่นยนต์และระบบอัตโนมัติ
สถาบันวิทยาการหุ่นยนต์ภาควิชานาม
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
ปีการศึกษา 2560

การออกแบบโครงสร้างและพัฒนาระบบพื้นฐานสำหรับหุ่นยนต์ชีวภาพอยู่ด้วย
เพื่อการศึกษาและวิจัย

นายจิรภูร์ ศรีรัตนอาภรณ์

นายเจษฎากร ท่าไชยวงศ์

นายวุฒิภัทร โชคอนันดร์พิริยะ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาศิวกรรมหุ่นยนต์และระบบอัตโนมัติ

สถาบันวิทยาการหุ่นยนต์ภาคสนาม

ปีการศึกษา 2560

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

คณะกรรมการสอบ

.....
(นายธนชชา ชูพจน์เจริญ)

.....
(ดร.อาบทิพย์ ธีรวงศ์กิจ)

.....
อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม

.....
(ดร.ปิติวุฒิ ธีรกิตติกุล)

.....
(รศ.ดร.ชิต เหล่าวัฒนา)

.....
(ดร.สุขชัย วงศ์บุณย์ยง)

ชื่อวิทยานิพนธ์	การออกแบบโครงสร้างและพัฒนาระบบที่นักวิชาชีวนิยมอยู่ที่เพื่อการศึกษาและวิจัย
หน่วยกิต	6
ผู้เขียน	นายจิรภูริศ ศรีรัตนอากรรณ์ นายเจษฎากร หาใจยวงศ์ นายวุฒิภัทร โขค่อนนันตทรัพย์
อาจารย์ที่ปรึกษา	ที่ปรึกษาวิทยานิพนธ์หลัก นายธนชชา ชูพจน์เจริญ ที่ปรึกษาวิทยานิพนธ์ร่วม รศ.ดร.ชิต เหล่าเวชนา
หลักสูตร	วิศวกรรมศาสตรบัณฑิต
สาขาวิชา	วิศวกรรมที่นักนิยมอยู่ที่และระบบอัตโนมัติ
คณะ	สถาบันวิทยาการหุ่นยนต์ภาคสนาม
ปีการศึกษา	2560

บทคัดย่อ

งานวิทยานิพนธ์นี้เป็นงานที่เกี่ยวกับการออกแบบและจัดทำแพลตฟอร์มหุ่นยนต์นักวิชาชีวนิยมอยู่ที่ด้วยเครื่องพิมพ์สามมิติ จุดประสงค์คือเพื่อ

กิตติกรรมประกาศ

ขอขอบพระคุณอาจารย์ ดร.นัชชา ชูพจน์เจริญ อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก ที่ได้สละเวลามาให้คำปรึกษา ชี้แนะแนวทาง ให้ความรู้ในด้านต่างๆ ที่จำเป็นต่องานวิจัย รวมถึงการให้การสนับสนุนในเรื่องอุปกรณ์ในการทำวิจัย ตลอดจนช่วยตรวจสอบแก้ไขวิทยานิพนธ์ให้เป็นไปอย่างสมบูรณ์

ขอขอบพระคุณรองศาสตราจารย์ ดร.ชิต เหล่าเวชนา อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม ที่ได้ชี้แนะแนวทางให้คำแนะนำ และให้เกียรติเข้าร่วมการสอบวิทยานิพนธ์

ขอขอบพระคุณอาจารย์ ดร.ภวิดา มณีวรรณ และนายวิษณุ จูราř ที่ได้ให้คำแนะนำในการแก้ไขปัญหาด้านต่างๆ ที่เกิดขึ้นระหว่างการทำวิจัย และได้ให้การสนับสนุนอุปกรณ์สำคัญที่ใช้ในการทำวิจัย

ขอขอบพระคุณอาจารย์ อาบทิพย์ ธีรวงศ์กิจ ที่กรุณาให้เกียรติเป็นประธานกรรมการสอบวิทยานิพนธ์ ให้คำแนะนำที่เป็นประโยชน์ต่อการจัดทำวิทยานิพนธ์ให้ดำเนินไปอย่างสมบูรณ์

ขอขอบพระคุณอาจารย์ ดร.ปิติวุฒิ รีกิตติกุล ที่กรุณาให้เกียรติเป็นกรรมการสอบวิทยานิพนธ์ ให้คำแนะนำที่เป็นประโยชน์ต่อการวิจัย และการแก้ไขปรับปรุงงานวิจัย ตลอดจนตรวจแก้วิทยานิพนธ์ให้ดำเนินไปอย่างสมบูรณ์

ขอขอบพระคุณอาจารย์ ดร.สุกัญญา วงศ์บุณย์ยง ที่กรุณาให้เกียรติเป็นกรรมการสอบวิทยานิพนธ์ ให้คำแนะนำที่เป็นประโยชน์ต่อการวิจัย และการแก้ไขปรับปรุงงานวิจัย ตลอดจนตรวจแก้วิทยานิพนธ์ให้ดำเนินไปอย่างสมบูรณ์

ขอขอบพระคุณคณาจารย์ และบุคลากรในสถาบันวิทยาการหุ่นยนต์ภาคนามทุกท่าน ที่ได้ให้คำปรึกษาและช่วยเหลือด้านสถานที่พร้อมทั้งส่งเสริมความдовet ต่างๆ ในระหว่างการทำวิทยานิพนธ์

ขอขอบคุณนักศึกษาปริญญาตรี สถาบันวิทยาการหุ่นยนต์ภาคนามทุกท่าน ที่ได้ให้คำแนะนำ ถามໄ่ และเป็นกำลังใจมาโดยตลอด

และสุดท้ายนี้ ขอน้อมรำลึกถึงพระคุณบิดา แมรดา และครอบครัว ที่ส่งเสริมให้กำลังใจ และให้การสนับสนุนในเรื่องต่างๆ จนกระทั้งข้าพเจ้าประสบความสำเร็จในการศึกษา

นายจิรภูริศ ศรีรัตนอาจารย์

นายเจษฎากร หาไซวงศ์

นายวุฒิภัทร โชคอนันตทรัพย์

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย.....	๗
กิตติกรรมประกาศ	ค
สารบัญ	ศ
รายการตราาง.....	จ
รายการรูปภาพ	ฉ
รายการสัญลักษณ์.....	ช
ประมวลศัพท์และตัวย่อ.....	ณ
บทที่ 1 บหนা	๑
1.1 ที่มาและความสำคัญ.....	1
1.2 วัตถุประสงค์.....	2
1.3 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.4 ขอบเขตการดำเนินงาน.....	2
1.5 ภาพรวมของระบบและขั้นตอนการดำเนินงาน	2
1.5.1 ส่วนโครงสร้างของหุ่นยนต์ชีวามโนยด์	3
1.5.2 ส่วนโปรแกรมของหุ่นยนต์ชีวามโนยด์	4
1.5.3 ส่วนการออกแบบระบบพื้นฐานเพื่อการพัฒนาต่ออยอด.....	4
1.6 นิยามศัพท์	5
บทที่ 2 ทฤษฎีและหลักการ	๖
2.1 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	6
2.1.1 หุ่นยนต์ชีวามโนยด์	6
2.1.2 ทฤษฎีที่เกี่ยวข้องกับมนุษย์	9
2.1.2.1 การวิเคราะห์การเดินของมนุษย์.....	9
2.1.2.2 การวิเคราะห์องศาอิสระของมนุษย์.....	9
2.1.3 ทฤษฎีที่เกี่ยวข้องกับหุ่นยนต์ชีวามโนยด์	11
2.1.3.1 ส่วนประกอบของหุ่นยนต์ชีวามโนยด์.....	11
2.1.3.2 วัสดุจัดการเดินของหุ่นยนต์ชีวามโนยด์.....	11
2.1.3.3 การสร้างและการควบคุมการเดินแบบสมดุลสถิต.....	11
2.1.3.4 การสร้างและการควบคุมการเดินแบบสมดุลพลวัต.....	12
2.1.3.5 จุดศูนย์กลางมวลของหุ่นยนต์.....	13
2.1.4 ตัวอย่างหุ่นยนต์ชีวามโนยด์	14
2.2 การออกแบบโครงสร้างของหุ่นยนต์	19
2.2.1 ความแตกต่างระหว่างโครงสร้างของมนุษย์กับโครงสร้างของหุ่นยนต์.....	19
2.2.1.1 ความแตกต่างขององศาสรี	19
2.2.1.2 ความแตกต่างของอัตราส่วน	19
2.2.1.3 กำลังและประสิทธิภาพของมอเตอร์.....	20
2.2.2 วัสดุและการขึ้นรูปโครงสร้างของหุ่นยนต์ชีวามโนยด์	20

สารบัญ (ต่อ)

เรื่อง	หน้า
2.2.3 อุปกรณ์ที่ใช้ในหุ่นยนต์อิวามาโนยด์	20
2.2.3.1 ระบบจ่ายพลังงาน	20
2.2.3.2 ตัวขับเคลื่อน	20
2.2.3.3 หน่วยประมวลผลควบคุม.....	20
2.2.3.4 เช่นเซอร์ตรวจหน้าสัมผัสที่พื้น	21
2.2.3.5 เช่นเซอร์วัดความเอียง	22
2.2.4 แนวคิดการออกแบบกลไกการเดินของหุ่นยนต์อิวามาโนยด์	23
2.3 การออกแบบโปรแกรมด้วย ROS.....	24
2.3.1 ระบบที่ใช้ในการพัฒนาหุ่นยนต์.....	24
2.3.2 ระบบที่ใช้ในการจำลองการทำงานของหุ่นยนต์.....	27
2.3.3 Robot Operating System	29
2.4 การออกแบบระบบพื้นฐาน	35
2.4.1 ข้อแตกต่างระหว่าง Open platform กับ Non-open platform.....	35
2.4.2 มาตรฐานหน่วยวัดและการบวกพิกัด	35
บทที่ 3 การดำเนินงานวิจัย	36
3.1 แผนการดำเนินงาน.....	36
3.2 การออกแบบโครงสร้างของหุ่นยนต์	38
3.2.1 การเชื่อมต่อหุ่นยนต์อิวามาโนยด์	38
3.2.2 อุปกรณ์ที่ใช้ในหุ่นยนต์อิวามาโนยด์อุทัย	38
3.3 การออกแบบโปรแกรมด้วย ROS.....	40
3.3.1 Modelling.....	40
3.3.1.1 ROS packages for robot modelling	40
3.3.1.2 URDF.....	40
3.4 การออกแบบระบบพื้นฐาน	42
บทที่ 4 ผลการวิจัย	44
4.1 การออกแบบโครงสร้างของหุ่นยนต์	44
4.2 การออกแบบโปรแกรมด้วย ROS.....	44
4.2.1 Simulation Gazebo	44
4.3 การออกแบบระบบพื้นฐาน	44
บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ	45
5.1 การออกแบบโครงสร้างของหุ่นยนต์	45
5.2 การออกแบบโปรแกรมด้วย ROS.....	45
5.3 การออกแบบระบบพื้นฐาน	45
5.4 สรุปภาพรวม	45
ภาคผนวก ก ผลงานตีพิมพ์เผยแพร่จากวิทยานิพนธ์.....	46
ก.1 บทความวิจัยเสนอในที่ประชุมวิชาการและมีการพิมพ์รวมเล่ม	46

สารบัญ (ต่อ)

เรื่อง	หน้า
ก.2 บทความวิชาการ.....	46
ภาคผนวก ข แหล่งข้อมูล Latex.....	47
ข.1 แหล่งข้อมูลออนไลน์	47
ประวัติผู้เขียน	48
ประวัติผู้เขียน	49
ประวัติผู้เขียน	50

รายการตาราง

ตาราง	หน้า
ตารางที่ 2.1 ความสามารถในการหมุนของแต่ละข้อต่อของมนุษย์.....	10
ตารางที่ 2.2 ตัวอย่างชื่อและข้อมูลของ Message	30
ตารางที่ 2.3 ตารางแสดงหน่วยวัดมาตรฐาน.....	35

รายการรูปภาพ

รูป	หน้า
รูปที่ 2.1 แสดงความแตกต่างของหุ่นยนต์ชีวามโนยด์แต่ละประเภท.....	6
รูปที่ 2.2 องค์ประกอบหลักของหุ่นยนต์ชีวามโนยด์	7
รูปที่ 2.3 วิวัจการการเดินของมนุษย์	9
รูปที่ 2.4 ส่วนประกอบของหุ่นยนต์ชีวามโนยด์.....	11
รูปที่ 2.5 วิวัจการการเดินของหุ่นยนต์ชีวามโนยด์.....	12
รูปที่ 2.6 การควบคุมตำแหน่งของจุดรวมมวลให้อยู่ในพื้นที่ฐาน	12
รูปที่ 2.7 การควบคุมตำแหน่งของจุดโนเมนต์ศูนย์ให้ตรงกับแรงปฏิกิริยารวม.....	13
รูปที่ 2.8 หุ่นยนต์ชีวามโนยด์ป้อบปี้	14
รูปที่ 2.9 หุ่นยนต์ชีวามโนยด์ไอคัพ	15
รูปที่ 2.10 หุ่นยนต์ชีวามโนยด์ดาร์วิน	16
รูปที่ 2.11 หุ่นยนต์ชีวามโนยด์นาโอะ	17
รูปที่ 2.12 หุ่นยนต์ชีวามโนยด์วากอท.....	18
รูปที่ 2.13 ตัวอย่างตำแหน่งและการหมุนของข้อต่อของหุ่นยนต์เพื่อการอ้างอิง	19
รูปที่ 2.14 รอแก๊กษา	20
รูปที่ 2.15 รอแก๊กษา	21
รูปที่ 2.16 รอแก๊กษา	21
รูปที่ 2.17 ลักษณะโครงสร้างของตัวตรวจจับแรงกด FSR.....	22
รูปที่ 2.18 การทำงานของตัวตรวจจับแรงกด FSR	22
รูปที่ 2.19 เช่นเซอร์วัสดความเฉื่อย	23
รูปที่ 2.20 player project middleware	24
รูปที่ 2.21 yarp middleware.....	24
รูปที่ 2.22 urbi middleware	25
รูปที่ 2.23 miro middleware	25
รูปที่ 2.24 openrdk middleware.....	25
รูปที่ 2.25 ROS middleware Rviz	26
รูปที่ 2.26 ROS algitecture	26
รูปที่ 2.27 ROS Moveit.....	26
รูปที่ 2.28 ผลลัพธ์จากการใช้โปรแกรม USARSim	27
รูปที่ 2.29 ผลลัพธ์จากการใช้โปรแกรม MuRoSimF	27
รูปที่ 2.30 Mobile robot with gazebo	28
รูปที่ 2.31 Quadrotor with gazebo	28
รูปที่ 2.32 ตัวอย่างสถาปัตยกรรมของ ROS.....	29
รูปที่ 2.33 ตัวอย่างไฟล์ package.xml.....	31
รูปที่ 2.34 ตัวอย่างการแสดงผลใน rqt	33
รูปที่ 2.35 ตัวอย่างการแสดงผลใน RViz	34

รายการรูปภาพ (ต่อ)

รูป	หน้า
รูปที่ 2.36 ตัวอย่างหุ่นยนต์ชีวามาโนยด์ Poppy	34
รูปที่ 2.37 การตั้งแกนตามกฎมือขวา	35
รูปที่ 3.1 การเชื่อมต่อระหว่าง Odroid กับ Dynamixel servos.....	38
รูปที่ 3.2 การอธิบาย link ใน URDF ไฟล์	41
รูปที่ 3.3 การอธิบาย Joint ใน URDF ไฟล์	41
รูปที่ 3.4 ภาพตัวอย่างการวาดออฟเจ็คต่างๆ.....	42
รูปที่ 3.5 ภาพตัวอย่างการวาดเฟรมของแขนกล	42
รูปที่ 3.6 ภาพตัวอย่างการวาดเฟรมของหุ่นยนต์ชีวามาโนยด์	43

รายการสัญลักษณ์

θ	เชิงตัว
d	distance
kg	Kilogram
m^2	Square Metre

ประมวลศัพท์และตัวย่อ

UTHAI	Universal Template for Humanoid Algorithm Interface
ROS	Robot Operating System
IMU	Inertial Measurement Unit
Dof	Degree of Freedom
CoM	Center of Mass
ZMP	Zero Moment Point
PLA	Polylactic acid
ABS	Acrylonitrile butadiene styrene
KMUTT	King Mongkut's University of Technology Thonburi
Liews	วุฒิภัทร ใจคงอันนฤทธิพงษ์
θ	เซ็ตฯ

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

หุ่นยนต์อิวามานอยด์เป็นหุ่นยนต์ที่สร้างขึ้นเพื่อเลียนแบบสีรีระร่างกายของมนุษย์ ซึ่งมีข้อจำนวนมากเพื่อให้มีการเคลื่อนไหวคล้ายมนุษย์ ลักษณะเด่นของหุ่นยนต์อิวามานอยด์คือ การเคลื่อนที่ด้วยขาสองข้าง ด้วยการเคลื่อนที่โดยการใช้ขาหน้า ทำให้หุ่นยนต์อิวามานอยด์สามารถเคลื่อนที่ได้ อย่างคล่องแคล่วในทุกสภาพพื้นผิว ทั้งทางเรียบ ทางชันหรือพื้นต่างระดับ¹ ซึ่งนั่นทำให้หุ่นยนต์ที่เดินสองขาแตกต่างจากหุ่นยนต์ที่เคลื่อนที่ด้วยล้อ ด้วยโครงสร้างของหุ่นยนต์ที่คล้ายมนุษย์นั้นเอง จึงทำให้หุ่นยนต์อิวามานอยด์สามารถทำงานได้หลากหลายและยืดหยุ่น สามารถที่จะใช้อุปกรณ์ทั่วไปที่ถูกออกแบบขึ้นมาเพื่อใช้กับมนุษย์ได้ ซึ่งหมายความว่าในอนาคตนั้นหุ่นยนต์อิวามานอยด์สามารถที่จะทำงานทดแทนแรงงานของมนุษย์ได้² งานที่หุ่นยนต์อิวามานอยด์จะเข้ามาทดแทนแรงงานของมนุษย์นั้น จะเป็นงานที่ต้องทำซ้ำๆ จนเกินความเมื่อยล้า งานที่อยู่ในพื้นที่อันตรายหรือที่เสี่ยงต่อการเกิดอุบัติเหตุ

สถาบันวิจัยหลายแห่งทั่วโลกกำลังให้ความสนใจสนับสนุนด้านการศึกษาวิจัยและพัฒนาหุ่นยนต์อิวามานอยด์ เพื่อให้ทำการกิจกรรมต่างๆ ยกตัวอย่างเช่น DARPA Robotics Challenge (DRC)³ เป็นรายการแข่งขันหุ่นยนต์กึ่งอัตโนมัติเพื่อทำการกิจกุศลในสถานการณ์ภัยพิบัติที่อันตราย ซึ่งสถาบันวิจัยหุ่นยนต์ทั่วโลกได้ส่งหุ่นยนต์อิวามานอยด์ของตนเข้าร่วมการแข่งขัน ในปัจจุบันได้มีการพัฒนาหุ่นยนต์อิวามานอยด์นั้นได้ก่อให้เกิดงานศึกษาวิจัย และทฤษฎีต่อยอด ต่างๆ มากมาย เช่น การวางแผนการเดิน การเดินแบบสติกติ การเดินแบบพลวัต การติดต่อสื่อสารของระบบ การมองเห็นและการประมวลผลภาพ การพูดคุยโต้ตอบกับมนุษย์ ปัญญาประดิษฐ์ ฯลฯ ซึ่งงานวิจัยเหล่านี้สามารถที่จะนำไปประยุกต์ใช้กับระบบหุ่นยนต์ระบบอื่นๆ ได้ แม้ว่าจะมีการพัฒนาหุ่นยนต์อิวามานอยด์มากมายแล้ว แต่การเริ่มต้นทำงานวิจัยที่มีความเกี่ยวข้องกับหุ่นยนต์อิวามานอยด์นั้น ต้องใช้ความรู้ความสามารถ เครื่องมือ ระยะเวลา งบประมาณ และ ความพยายามเป็นอย่างมาก การสร้างหุ่นยนต์อิวามานอยด์ขึ้นมาใหม่นั้นต้องใช้เวลาประมาณสูง ดังนั้นการสร้างระบบจำลองของหุ่นยนต์อิวามานอยด์ขึ้นมาเป็นระบบพื้นฐาน ให้มีความพร้อมสำหรับการพัฒนาต่อยอดแก้ไขศึกษาหรือนักวิจัย จะช่วยประหยัดเวลาและงบประมาณที่ต้องใช้ได้อย่างมาก ซึ่งนั่นหมายความว่าสถาบันวิจัยจะสามารถทำงานวิจัยได้อย่างมีประสิทธิภาพมากขึ้น

ในงานวิจัยนี้เป็นการออกแบบหุ่นยนต์อิวามานอยด์และพัฒนาระบบพื้นฐานของหุ่นยนต์อิวามานอยด์ สำหรับให้นักศึกษาหรือนักวิจัยสามารถพัฒนาต่อยอดได้ โดยหุ่นยนต์อิวามานอยด์ที่ออกแบบมาขึ้น สามารถที่จะปรับปรุงแก้ไข ดัดแปลงได้ง่าย ตัวโครงสร้างจะใช้เป็น พลาสติก PLA ที่สามารถขึ้นรูปได้ โดยการใช้เครื่องพิมพ์สามมิติ มีเซนเซอร์ตรวจการสัมผัสพื้นที่ฝ่าเท้าของหุ่นยนต์ มีเซนเซอร์สำหรับการวัดมุมเอียง ที่ลำตัวของหุ่นยนต์ และเพื่อที่จะทำให้ง่ายต่อการศึกษาทำความเข้าใจ บำรุงรักษา จึงได้มีการจัดทำคู่มือและเอกสารวิธีการใช้งานอย่างชัดเจน โดยจะเก็บในรูปแบบของเอกสารออนไลน์

¹ การเคลื่อนที่ของหุ่นยนต์รูปแบบต่างๆ ราชภัฏสวนดุสิต

² ณัฐพงษ์ วริประเสริฐ และณรงค์ ล้ำดี (2552: 374)

³ DARPA 2015 [<https://www.darpa.mil/about-us/about-darpa>]

1.2 วัตถุประสงค์

ออกแบบโครงสร้างของหุ่นยนต์ชีวามโนยด์ที่สามารถแก้ไขปรับเปลี่ยนได้ง่าย พัฒนาระบบที่มีพื้นฐาน ระบบจำลองสำหรับหุ่นยนต์ชีวามโนยด์เพื่อใช้ในการศึกษาวิจัย รวบรวมเครื่องมือที่เป็นประโยชน์ต่อการพัฒนาหุ่นยนต์ และจัดทำเอกสารออนไลน์ ให้บุคคลที่สนใจสามารถเข้ามาศึกษาได้

1.3 ประโยชน์ที่คาดว่าจะได้รับ

- 1.3.1 มีต้นแบบหุ่นยนต์ชีวามโนยด์สำหรับใช้ในงานวิจัยแขนงต่างๆ
- 1.3.2 มีระบบพื้นฐานสำหรับพัฒนาหุ่นยนต์ชีวามโนยด์รุ่นใหม่ในสถาบัน
- 1.3.3 มีระบบจำลองสำหรับจำลองการทำงานของหุ่นยนต์ชีวามโนยด์
- 1.3.4 มีแหล่งรวมเครื่องมือสำหรับการพัฒนาหุ่นยนต์
- 1.3.5 มีคู่มือ เอกสาร วิธีการใช้งาน และรายละเอียดของหุ่นยนต์สำหรับพัฒนาต่อยอด

1.4 ขอบเขตการดำเนินงาน

- 1.4.1 ใช้ ROS เป็นกรอบการทำงานสำหรับพัฒนาระบบที่มีพื้นฐาน
- 1.4.2 ออกแบบโครงสร้างให้มีความแข็งแรง สามารถรับน้ำหนักอุปกรณ์ที่ติดตั้งอยู่บนตัวหุ่นยนต์ได้
- 1.4.3 น้ำหนักของหุ่นยนต์รวมกันทั้งตัว ไม่เกิน 5 กิโลกรัม
- 1.4.4 ใช้ Solidworks 3D เป็นโปรแกรมสำหรับออกแบบโครงสร้าง และคำนวณ
- 1.4.5 หุ่นยนต์มีความสูงไม่ต่ำกว่า 100 เซนติเมตร และสูงไม่เกิน 120 เซนติเมตร
- 1.4.6 หุ่นยนต์มี 2 แขน 2 ขา มีองศาอิสระของขาข้างละ 6 และแขนข้างละ 2 องศาอิสระ
- 1.4.7 หุ่นยนต์สามารถทำงานได้ภายในสภาพแวดล้อมแบบปิด
- 1.4.8 หุ่นยนต์ใช้พลังงานจากแหล่งจ่ายพลังงานที่มีขนาดแรงดันไฟฟ้า 12 โวลต์
- 1.4.9 หุ่นยนต์ใช้ตัวขับเคลื่อนแบบดิจิตอลสำหรับแต่ละข้อต่อเป็น Dynamixel Digital Servo
- 1.4.10 ใช้ Gazebo สำหรับจำลองระบบของหุ่นยนต์
- 1.4.11 ติดตั้งเซนเซอร์วัดการกด (Ground contact) ที่ฝ่าเท้าของหุ่นยนต์
- 1.4.12 ติดตั้งเซนเซอร์วัดมุมเอียง (IMU) ที่บริเวณลำตัวของหุ่นยนต์
- 1.4.13 จัดทำคู่มือ เอกสารการใช้งาน และรายละเอียดส่วนประกอบของหุ่นยนต์

1.5 ภาพรวมของระบบและขั้นตอนการดำเนินงาน

งานวิจัยนี้การดำเนินงานวิจัยถูกแบ่งออกเป็นสามส่วน คือ ส่วนที่หนึ่งส่วนโครงสร้างของหุ่นยนต์ชีวามโนยด์ เป็นส่วนที่ทำในส่วนของการขึ้นรูปชิ้นงาน ออกแบบโมเดลสามมิติ รวมไปถึงระบบอิเล็กทรอนิกส์ ติดตั้งบอร์ดและเซนเซอร์ไว้ตามจุดต่างๆ เพื่อสร้างโครงสร้างของหุ่นยนต์ให้สามารถรองรับการเดินได้ ส่วนที่สองส่วน โปรแกรมของหุ่นยนต์ชีวามโนยด์ เป็นส่วนที่ทำในส่วนของการสั่งการตัวขับเคลื่อนต่างๆ อ่านค่าสถานะเซนเซอร์จากคอมโวเลอร์ รวมไปถึงระบบจำลองการทำงานของหุ่นยนต์ และส่วนที่สามส่วนการออกแบบระบบพื้นฐาน เพื่อการพัฒนาต่อยอด ส่วนนี้จะเป็นส่วนที่ทำให้ผู้ที่จะมาวิจัยต่อยอดสามารถทำงานได้ง่ายขึ้น จัดการเอกสารคู่มือ การใช้งานต่างๆ ให้เป็นระบบเบเย็บ สามารถแยกขั้นตอนการทำงานของแต่ละส่วนออกเป็นข้อดังนี้

1.5.1 ส่วนโครงสร้างของหุ่นยนต์อิวามานอยด์

1 ศึกษาค้นคว้าเอกสารและงานวิจัยที่เกี่ยวข้อง

- ศึกษาเกี่ยวกับส่วนประกอบของหุ่นยนต์อิวามานอยด์
- ศึกษาลักษณะกายภาพของมนุษย์
- ศึกษาลักษณะกายภาพของหุ่นยนต์อิวามานอยด์
- ศึกษาการออกแบบฝ่าเท้าและเซนเซอร์ตรวจจับพื้น
- ศึกษาการเลือกใช้ตัวขับเคลื่อนและวิธีการไดร์ฟมอเตอร์
- ศึกษาการส่งกำลังไปยังแต่ละข้อต่อ
- ศึกษาการออกแบบระบบจ่ายพลังงาน
- ศึกษาการใช้งานเซนเซอร์วัดมุมอุปกรณ์ IMU

2 ออกแบบโครงสร้างของหุ่นยนต์อิวามานอยด์

- ออกแบบขนาดของหุ่นยนต์
- ตรวจสอบความเป็นไปได้ทาง Kinematics
- เลือกขนาดของมอเตอร์
- ออกแบบการยึดมอเตอร์กับโครงสร้าง
- ออกแบบโครงสร้างของหุ่นยนต์
- สร้าง CAD Model ของหุ่นยนต์
- จำลองและทดสอบความแข็งแรง
- ปรับปรุงโครงสร้าง
- วางแผนไฟฟ้า
- การเลือกใช้แหล่งพลังงาน
- การเลือกใช้คอมพิวเตอร์
- การไดร์ฟมอเตอร์
- การเข้ามืออาชีวะเซนเซอร์ชนิดต่างๆ

3 จัดสร้างโครงสร้างหุ่นยนต์อิวามานอยด์

- สั่งซื้อส่วนประกอบทางกล
- สั่งซื้อส่วนประกอบทางไฟฟ้า
- จัดทำส่วนที่ต้องออกแบบเอง
- ประกอบโครงสร้างทางกล
- ประกอบโครงสร้างทางไฟฟ้า

4 ทดสอบการทำงานของหุ่นยนต์อิวามานอยด์

- ทดสอบการเคลื่อนไหวของหุ่นยนต์
- ตรวจสอบ Joint limit
- ตรวจสอบ Joint frames ทิศทางการหมุน
- ตรวจสอบ Kinematics

- ทดสอบการขับเคลื่อนของหุ่นยนต์
- ตรวจสอบ Power consumption
- ทดสอบหารับรู้ของหุ่นยนต์
- ตรวจสอบการอ่านค่าจากเซนเซอร์ IMU
- ตรวจสอบการอ่านค่าจากเซนเซอร์ Ground contact

1.5.2 ส่วนโปรแกรมของหุ่นยนต์ชีวามโนยด์

1 ศึกษาค้นคว้าเอกสารและงานวิจัยที่เกี่ยวข้อง

- ศึกษาระบบที่ใช้ในการพัฒนาหุ่นยนต์
- ศึกษาระบบที่ใช้ในการจำลองการทำงานของหุ่นยนต์

2 ประยุกต์ระบบที่ใช้ในการพัฒนาหุ่นยนต์

3 ประยุกต์ระบบจำลองการทำงานของหุ่นยนต์

1.5.3 ส่วนการออกแบบระบบพื้นฐานเพื่อการพัฒนาต่อไป

1 ติดตั้งระบบ

- Setup Version Control [GitHub]
- ติดตั้ง ROS ในคอมพิวเตอร์ที่อยู่ในหุ่นยนต์
- ติดตั้ง ROS ในคอมพิวเตอร์ที่ควบคุมหุ่นยนต์
- ตั้งค่า ROS Workspace
- ตั้งค่าการติดต่อสื่อสารระหว่างระบบ

2 วางแผนระบบพื้นฐาน

3 จัดทำคู่มือ

- รายการของที่จำเป็นต้องใช้
- คู่มือการประกอบของหุ่นยนต์
- คู่มือรายละเอียดข้อมูลของหุ่นยนต์
- คู่มือลิมิตของแต่ละข้อต่อของหุ่นยนต์
- คู่มือการวางแผนสร้างของระบบ

4 สื่อการสอน

- การติดตั้ง ROS
- การใช้งาน ROS เป็องตัน
- การใช้งาน Simulation ของหุ่นยนต์
- การใช้งาน ROS กับหุ่นยนต์ชีวามโนยด์
- การใช้งาน ROS บน Mbed

1.6 นิยามศัพท์

Gain Pattern คือ รูปแบบการเดินของขาทั้งสองข้างให้เป็นรูปแบบการก้าวเดินเดียวกัน

Embedded System คือ ระบบสมองกลฝังตัว ใช้ในการควบคุมอุปกรณ์ต่างๆ สามารถนำไปใช้ในการควบคุมเฉพาะทาง การควบคุมแบบแยกส่วน โดยอาศัยตัวประมวลผล ที่ทำหน้าที่วิเคราะห์และประมวลผลข้อมูลตามโปรแกรมที่ได้ออกแบบไว้

บทที่ 2

ทฤษฎีและหลักการ

2.1 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1.1 หุ่นยนต์อิริยาบถ

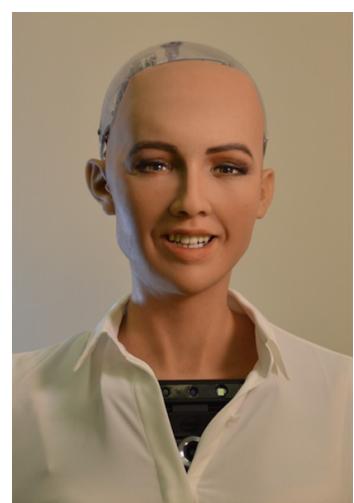
หุ่นยนต์อิริยาบถ คือ หุ่นยนต์ที่ถูกสร้างขึ้นมาให้มีรูปร่างคล้ายคลึงกับสรีระโครงสร้างของมนุษย์ มักถูกออกแบบขึ้นมาเพื่อจุดประสงค์เฉพาะอย่าง เช่น เพื่อให้ใช้เครื่องมือต่างๆของมนุษย์ เพื่อให้อยู่ในสภาพแวดล้อมของมนุษย์ เพื่อศึกษาการเคลื่อนไหวของร่างกายมนุษย์ เพื่อศึกษาร่องเท้นของมนุษย์ เพื่อทำงานในสิ่งที่มนุษย์ทำได้ยาก หรือเพื่อวัตถุประสงค์อื่นๆ โดยทั่วไปแล้ว หุ่นยนต์อิริยาบถจะประกอบไปด้วย 4 ส่วนคือ ส่วนของหัว ส่วนของลำตัว ส่วนของแขน และส่วนของขา แต่การสร้างหุ่นยนต์อิริยาบถต้นที่ไม่จำเป็นที่จะต้องมีส่วนประกอบทุกส่วนดังที่กล่าวไป ในบางครั้งอาจมีเพียงแค่ส่วนบนเท่านั้น ดังรูปที่ 2.1ก หุ่นยนต์นามจากสถาบันวิทยาการหุ่นยนต์ภาควิชาชีวภาพ เป็นหุ่นยนต์ที่มีส่วนบนเหมือนมนุษย์ แต่มีส่วนล่างเป็นล้อ หรือหุ่นยนต์อิริยาบถที่มีเพียงแค่ส่วนล่าง ดังรูปที่ 2.1ข หุ่นยนต์สัมจุก เป็นหุ่นยนต์อิริยาบถที่มีเพียงแค่ส่วนขาเท่านั้น หรือหุ่นยนต์อิริยาบถที่มีเพียงใบหน้าเหมือนมนุษย์ ดังรูปที่ 2.1ข หุ่นยนต์โซเฟีย เป็นแอนดรอยด์ที่มีหน้าตาคล้ายมนุษย์มาก มีปาก สามารถพูดปฏิสัมพันธ์กับมนุษย์ได้



(ก) หุ่นยนต์ประชาสัมพันธ์โรงแรม



(ข) หุ่นยนต์เดินสองขาสัมจุก



(ข) หุ่นยนต์แอนดรอยด์โซเฟีย

รูปที่ 2.1: แสดงความแตกต่างของหุ่นยนต์อิริยาบถแต่ละประเภท

งานวิจัยทางด้านหุ่นยนต์อิริยาบถจากอดีตจนถึงปัจจุบันส่วนใหญ่จะเป็นการพัฒนาความสามารถของหุ่นยนต์ เช่น เริ่มต้นจากแรกสุดจะเป็นการพัฒนาให้หุ่นยนต์สามารถเดินหน้าได้ ต่อมาเกิดความสามารถให้หุ่นยนต์สามารถเดินบนพื้นอิฐ พื้นหิน ฯลฯ ด้วยการเดินเลี้ยวซ้ายขวา เดินขึ้นลงบันได ฯลฯ เป็นต้น นอกจากนี้ยังมีการพัฒนาปรับปรุงสมดุลของการเดินแบบสองขาอีกด้วย สมดุลของการเดินสามารถแบ่งได้สองแบบหลัก คือ การเดินแบบสมดุลสถิต และการเดินแบบสมดุลพลวัต งานในยุคแรกนั้นจะพัฒนาให้เดินได้แบบสมดุลสถิต ต่อมาก็เป็นสมดุลกึ่งพลวัต และเป็นสมดุลพลวัต การพัฒนาตัวควบคุมการเดินของหุ่นยนต์ จำเป็นที่จะต้องใช้ความรู้ทางด้านกลศาสตร์ค่อนข้างมาก มีการใช้สมการที่มีความซับซ้อน

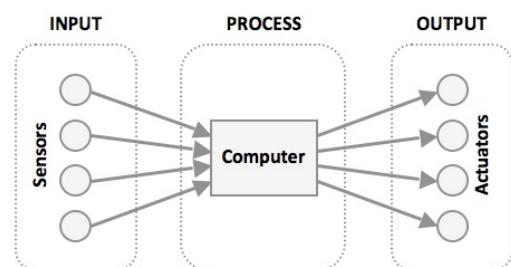
Zheng และคณะ (1988) พัฒนาหุ่นยนต์สองขาที่สามารถเดินบนพื้นราบได้ ให้สามารถเดินต่อเนื่องไปบนพื้นเอียงได้ด้วย พื้นเอียงที่ใช้มีลักษณะเป็นพื้นเอียงขึ้น หุ่นยนต์ที่ใช้ในงานนี้มีข้อต่อสะโพก (hip), ข้อเท้า (ankle) และลำตัว (torso) มีเซนเซอร์วัดแรงกด (force sensor) ติดตั้งอยู่ที่ปลายเท้าและสันเท้าแต่ละข้างเพื่อใช้วัดตำแหน่งของน้ำหนักโดยรวม (center of gravity) ของหุ่นยนต์ การเดินของงานวิจัยจะพิจารณาเฉพาะการเดินในแนวหน้าหลัง โดยมีหลักการคือ การเดินบนพื้นเอียงโดยที่หุ่นยนต์ยังเดินในท่าทางเหมือนกับตอนที่เดินบนพื้นราบจะทำให้น้ำหนักโดยรวมของหุ่นยนต์เลื่อนไปข้างหลัง ดังนั้นการที่หุ่นยนต์ขับลำตัวไปด้านหน้าจะทำให้น้ำหนักโดยรวมของหุ่นยนต์กลับมาอยู่ตรงกลางของพื้นที่รับน้ำหนักเหมือนเดิม ซึ่งจะทำให้หุ่นยนต์มีความสมดุลได้ ดังนั้นข้อมูลที่ได้จากหน่วยวัดแรงกดที่เท้าจะถูกนำมาคำนวณตลอดการเดินเพื่อใช้ในการปรับเปลี่ยนมุกการขับของลำตัว การเดินบนพื้นราบเป็นแบบสมดุลสถิตและการเดินบนพื้นเอียงก็ยังคงเป็นแบบสมดุลสถิตเช่นกัน

Inaba และคณะ (1995) สร้างหุ่นยนต์เลียนแบบลิง (ape-like biped) ประกอบด้วยสองมือและสองขา มีการเดินแบบสมดุลสถิต งานวิจัยนี้มีความคิดว่าหากจากการทำให้หุ่นยนต์สองขาเดินได้โดยไม่ล้มแล้ว ควรจะทำหุ่นยนต์ที่สามารถลุกขึ้นเองได้หลังจากที่ล้มแล้วด้วย ดังนั้นในงานนี้ หุ่นยนต์ถูกพัฒนาให้สามารถเดิน เมื่อล้มแล้ว ก็สามารถพลิกตัวและลุกขึ้นมาเดินให้ได้

Kun และ Miller (1996) ได้นำโครงข่ายประสาทเทียม มาประยุกต์ใช้ในการปรับเปลี่ยนท่าทางการเดิน โดยอัตโนมัติของหุ่นยนต์สองขา การที่หุ่นยนต์สามารถปรับเปลี่ยนท่าทางได้โดยอัตโนมัตินี้มีประโยชน์ทำให้หุ่นยนต์เดินได้บนพื้นผิวหลากหลายลักษณะมากขึ้น ในงานนี้พิจารณาทั้งสมดุลในแนวหน้าหลัง (sagittal plane) และแนวซ้ายขวา (frontal plane) และการเดินของหุ่นยนต์เป็นแบบสมดุลพลวัต หลักการทำงานประกอบด้วยตัวสร้างท่าทางการเดินหนึ่งตัว และตัวปรับท่าทางการเดินทั้งแนวหน้าหลังและซ้ายขวาอีกหนึ่งตัว โดยค่าการปรับเปลี่ยนนั้นจะได้มาจากการรับสัญญาณที่เข้ามาทางสายตา ความยาวการก้าวเท้า ความสูงของการยกเท้า เป็นต้น นอกจากนี้ ในปีถัดมา ทั้งสองได้ใช้หลักการที่ใช้ในงานนี้ไปใช้กับการเดินของหุ่นยนต์อีกตัว (Kun and Miller, 1997)

Hirai และคณะ (1998) พัฒนาหุ่นยนต์ชีวนิรภัย ซึ่งตัวหุ่นยนต์มีความคล้ายมนุษย์มาก สามารถเดินได้อย่างราบลื่นคล้านมันุษย์มากที่สุด เช่น สามารถเดินได้ในพื้นผิวนิ่มต่างๆ เดินได้บนพื้นเอียงขึ้นเอียงลง เดินขึ้นลงบันไดได้ เดินเข็นรถได้ เป็นต้น การเดินในทุกสถานการณ์เป็นการเดินแบบสมดุลพลวัต หุ่นยนต์สามารถเดินได้ด้วยความเร็วสูงสุด 4.7 กิโลเมตรต่อชั่วโมง หุ่นยนต์ประกอบไปด้วย เขนข้างละ 9 องศาอิสระ ขาข้างละ 6 องศาอิสระ ที่บริเวณหัวมีกล้องติดตั้งอยู่ 4 ตัว นอกจากนี้ยังมีอุปกรณ์ที่ใช้ในการรักษาสมดุลอีก 4 อีกด้วย IMU ที่ติดตั้งบริเวณลำตัว และ Force sensor ที่ติดที่เท้าทั้งสองข้าง

องค์ประกอบของหุ่นยนต์ที่จะประกอบไปด้วยกระบวนการตอบสนองต่างๆที่เป็นระบบ ซึ่งเราสามารถจำแนกการอุปกรณ์เป็นส่วนหลักๆได้สามส่วนคือ ส่วนการรับรู้ ส่วนการประมวลผล และส่วนการขับเคลื่อน ทั้งหมดเมื่อนำมาร่วมเข้าด้วยกันแล้ว เราสามารถที่จะควบคุมการทำงานของหุ่นยนต์ชีวนิรภัยได้



รูปที่ 2.2: องค์ประกอบหลักของหุ่นยนต์ชีวนิรภัย

การรับรู้ของหุ่นยนต์อิวามาโนย์

การรับรู้ของหุ่นยนต์อิวามาโนย์นั้นมีความยากมากกว่าหุ่นยนต์ชนิดอื่นๆ เพราะหุ่นยนต์จะมีการเคลื่อนที่และการเคลื่อนที่นั้นทำให้เซนเซอร์โดนรบกวนได้ ยกตัวอย่างเช่น ภาพที่ได้จากการกล้องนั้นอาจจะเบลอได้ถ้าความเร็วของชัตเตอร์ช้าเกินไป หรือว่าภาพเปลี่ยนขณะที่กำลังกดชัตเตอร์ ข้อมูลตำแหน่งของตัวเองก็มีความแน่นอนที่น้อยกว่าหุ่นยนต์เคลื่อนที่ด้วยล้อ เพราะเซนเซอร์ที่วัดตำแหน่งเทียบกับเฟรมโลโก้มีความเสถียร หุ่นยนต์ที่เคลื่อนที่ด้วยล้อปกติติดกล้อง ตัวกล้องจะมีความสูงจากพื้นคงที่ แต่หุ่นยนต์อิวามาโนย์ไม่ใช่ โดยหุ่นยนต์อิวามาโนย์นั้น จะต้องมีการคำนวณ forward kinematics จากเท้าที่สัมผัสกับพื้นมา焉กกล้องเพื่อหาตำแหน่งและการหมุนของกล้อง ส่วนการวัดตำแหน่งของตัวหุ่นยนต์นั้น โดยที่ว่าไปแล้วจะใช้เซนเซอร์ inertia measurement unit (IMU) และเซนเซอร์ Encoders สำหรับหาตำแหน่งของข้อต่อต่างๆ ปกติจะติดเซนเซอร์ IMU ไว้ที่ลำตัวของหุ่นยนต์ใกล้ๆ กับ center of mass ของหุ่นยนต์ ส่วน Encoder นั้นจะติดไว้ที่ข้อต่อของหุ่นยนต์

การประมวลผลของหุ่นยนต์อิวามาโนย์

ในปัจจุบันนี้หุ่นยนต์อิวามาโนย์มีความสามารถในการคำนวณที่สูงมากเมื่อเทียบกับเมื่อก่อน บอร์ดที่เราสามารถเห็นได้โดยทั่วไป เช่น Raspberry Pi, Odroid, Intel NUCs ซึ่งตัวบอร์ดมีขนาดเล็กจึงทำให้เข้าไปอยู่ในตัวของหุ่นยนต์ได้ แรมบอร์ดพวงนี้ยังมี GPUs และ CPU หลายคอร์อีกด้วย บางครั้งก็มีคินที่อาพากรบอร์ดพวงนี้มาทำงานร่วมกันหลายๆ ตัว ประมวลผลแบบ pararell เพื่อที่จะเพิ่มประสิทธิภาพในการประมวลผล โดยเชื่อมต่อระหว่างกันผ่าน Ethernet network

การขับเคลื่อนของหุ่นยนต์อิวามาโนย์

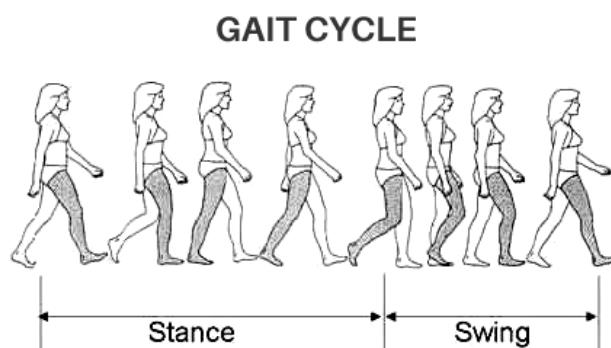
หุ่นยนต์อิวามาโนย์ส่วนใหญ่จะมีข้อต่ออยู่หลายๆ จุด แต่ละข้อต่อจะมีตัวขับเคลื่อน ตัวขับเคลื่อนมีอยู่หลักๆ สองแบบคือ แบบกล้ามเนื้อของมนุษย์ และแบบมอเตอร์ที่ติดตรงที่ข้อต่อเลย ที่นิยมใช้คือแบบมอเตอร์ที่ติดที่ข้อต่อเลย เพราะทำให้ตัวของหุ่นยนต์มีขนาดเล็ก ใช้พื้นที่น้อย การใช้เส้นเอ็นดึงนั้นจะการจะทำให้ข้อต่อไปยังตำแหน่งที่ต้องการได้ยากกว่า ตัวขับเคลื่อนนั้นต้องการแรงมากน้อยขึ้นอยู่กับ น้ำหนักของตัวหุ่นยนต์ เพื่อที่จะทำให้หุ่นยนต์นั้นยังยืนได้

2.1.2 ທາງກົດທີ່ເກື່ອງຂອງກັບມຸນໆ

2.1.2.1 การวิเคราะห์การเดินของมนุษย์

การเคลื่อนที่ของหุ่นยนต์อิมามอยด์นั้นจะเลียนแบบมาจากการเดินของมนุษย์ ดังนั้นการวิเคราะห์ลักษณะการเดินของมนุษย์ จะเป็นการศึกษาเพื่อทำความเข้าใจถึงธรรมชาติการเดิน ก่อนนำไปทำการออกแบบกลไกทางกลและระบบควบคุมของหุ่นยนต์อิมามอยด์ การก้าวเดินของมนุษย์โดยปกติแล้ว จะมีลักษณะเป็นวัฏจักร วนซ้ำไปเรื่อยๆ ในทิศทางที่ต้องการจนกว่าจะทำการหยุดเดิน การทรงตัวในระหว่างการยืนหรือการเดินนั้น เป็นไปตามสัญชาตญาณซึ่งเกิดจากการรักษาความสมดุลของระดับน้ำในหู¹ ส่งสัญญาณผ่านเส้นประสาทไปยังกล้ามเนื้อส่วนต่างๆ ที่ทำหน้าที่ให้เกิดการเคลื่อนที่

การเคลื่อนที่ของมนุษย์ในการเดินไปข้างหน้าสามารถแบ่งออกเป็นช่วงต่างๆ ดังนี้



รูปที่ 2.3: วิจัยการเดินของมนุษย์

1. ช่วงเริ่มการวางแผนเพื่อเข้าสู่ช่วงเริ่มต้นให้ร่วงเท้า เป็นช่วงที่เท้าเกิดการกระแทกลงบนพื้นหลังจากทำการให้ร่วงมาจากด้านหลัง โดยธรรมชาติมนุษย์จะทำการวางแผนสันเท้าลงเพื่อลดแรงกระแทกที่เกิดขึ้นในช่วงนี้ ดังนั้นทางกายภาพในส่วนของสันเท้ามนุษย์จึงมีลักษณะอ่อนนุ่ม
 2. ช่วงเริ่มต้นให้ร่วงเท้าเพื่อเข้าสู่ช่วงให้ร่วงเท้า หลังจากทำการวางแผนสันเท้าลงกับพื้นแล้ว ข้อเข้าจะปรับมุมเพื่อให้ไฟเท้าแนวบกพื้นสนิท ขณะเดียวกันขาอีกข้างจะยกสูงขึ้นเพื่อถ่ายเทน้ำหนักไปยังเท้าที่เพิ่งวางแผน
 3. ช่วงให้ร่วงเท้า เป็นช่วงที่ขาหนึ่งยกโดยอยู่ในอากาศและขาที่วางแนวกับพื้นจะรองรับน้ำหนักทั้งหมดของร่างกาย
 4. ช่วงเตรียมการวางแผนเท้า เป็นช่วงที่ขาข้างที่ลอยอยู่ให้ร่วงไปข้างหน้าเพื่อเตรียมเข้าสู่ช่วงรองรับ ในขณะเดียวกันขาที่รับน้ำหนักอยู่จะทำการผลักตัวเพื่อเริ่มทำการถ่ายเทน้ำหนักไปข้างหน้า

2.1.2.2 การวิเคราะห์องศาอิสระของมนุษย์

การที่มนุษย์สามารถเคลื่อนที่ได้นั้น เป็นผลเนื่องมาจากการเคลื่อนที่ของข้อต่อต่าง ๆ ที่อยู่บนขา ซึ่งประกอบไปด้วย ข้อต่อส่วนสะโพก ข้อต่อส่วนหัวเข่า และข้อต่อส่วนข้อเท้า แรงบิดที่เกิดขึ้นของแต่ละข้อต่อมีความสัมพันธ์กับกัน ส่งผลให้เกิดเสถียรภาพในการเดินของมนุษย์ เมื่อวิเคราะห์ลักษณะโครงสร้างในแต่ละส่วน พบร่วมกันว่า ข้อต่อส่วนสะโพกมีลักษณะเป็นทรงกลม ทำให้ข้อต่อส่วนสะโพกสามารถหมุนได้ 3 องศาอิสระ ส่วนหัวเข่าของมนุษย์มีจุดต่อของข้อที่ไม่ลักษณะเป็นทรงกลม ส่องลูกประกาบนี้ด้วยกันทำให้การเคลื่อนที่ถูกบังคับให้สามารถ

1text

เคลื่อนที่ได้เพียง 1 องศาอิสระ ในส่วนของข้อเท้ามีลักษณะการเคลื่อนที่เหมือนสะโพกคือสามารถเคลื่อนที่ได้ 3 องศาอิสระ

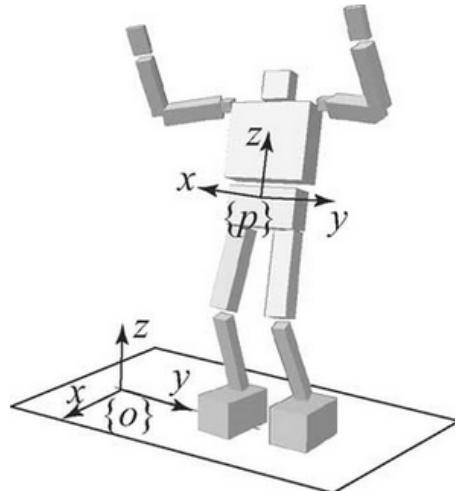
จากทั้งหมดที่ได้ทำการวิเคราะห์มาข้างต้นพบว่าในขาหนึ่งข้างของมนุษย์ประกอบด้วย 7 องศาอิสระ ซึ่งส่งผลให้การเคลื่อนที่ของมนุษย์มีความคล่องแคล่วสูง แต่ในทางออกแบบกลไกการเดินและการควบคุม ของหุ่นยนต์สองขาถือว่ามีจำนวนองศาอิสระเกินความจำเป็นในการเคลื่อนที่บนปริภูมิ(space) และยากต่อการควบคุม (under actuated) ดังนั้นการกำหนดจำนวนองศาอิสระเพื่อให้หุ่นยนต์เดินได้เสมี่อนมนุษย์จึงมีผลในการออกแบบกลไกทางกลและการควบคุมของหุ่นยนต์สองขา

ตารางที่ 2.1: ความสามารถในการหมุนของแต่ละข้อต่อของมนุษย์

ข้อต่อ	องศาอิสระ	องศาสารหมุน	
		สูงสุด	ต่ำสุด
หัว	θ_x	+60	-30
	θ_y	+70	-70
	θ_z	+80	-80
หลัง	θ_x	+30	-30
	θ_y	+55	-55
	θ_z	+45	-45
หัวไหล่	θ_x	+180	-80
	θ_y	+45	-135
	θ_z	+30	0
ศอก	θ_x	0	-155
สะโพก	θ_x	+120	-40
	θ_y	+40	-50
	θ_z	+60	-50
หัวเข่า	θ_x	0	-130
ข้อเท้า	θ_x	+30	-60
	θ_y	+45	-20
	θ_z	+20	-60

2.1.3 ทฤษฎีเกี่ยวกับหุ่นยนต์อิวามานอยด์

2.1.3.1 ส่วนประกอบของหุ่นยนต์อิวามานอยด์



รูปที่ 2.4: ส่วนประกอบของหุ่นยนต์อิวามานอยด์

หุ่นยนต์อิวามานอยด์ประกอบด้วยก้านต่อหulary ก้านที่นำมาต่อกัน ลักษณะโครงสร้างนั้นจะเป็นแบบแบบโข เปิด และแต่ละก้านต่อจะเชื่อมต่อกันด้วยข้อต่อ เราสามารถที่จะแบ่งง่ายๆออกเป็น 2 ส่วน ส่วนแรกคือ ส่วน ก้านต่อของลำตัวหุ่นยนต์ (Torso) ซึ่งเราสามารถที่จะเห็นรวมไปถึงส่วนแขนกับหัวด้วย และในส่วนที่สองคือ ส่วน ก้านต่อของขาหุ่นยนต์ (Legs) ซึ่งเป็นส่วนของหุ่นยนต์ทั้งสองข้างที่สามารถนำไปที่สัมผัสกับพื้นได้ ทั้งสองก้าน ต่อนี้ถูกเชื่อมต่อกันด้วยส่วนของสะโพก (Hip) ที่อยู่ระหว่างส่วนลำตัวกับส่วนของขาหุ่นยนต์ ดังรูปที่ 2.4

2.1.3.2 วัฏจักรการเดินของหุ่นยนต์อิวามานอยด์

วัฏจักรการเดินของหุ่นยนต์ คือ การที่หุ่นยนต์จะต้องมีการถ่ายน้ำหนักไปมาระหว่างเท้าซ้ายและเท้าขวา มีบางช่วงที่น้ำหนักตกลงบนเท้าข้างใดข้างหนึ่งหรือทั้งสองข้างพร้อมกัน สามารถแบ่งออกเป็นช่วงได้สองช่วง คือ ช่วงการยืนด้วยขาข้างเดียว และช่วงการยืนด้วยขาทั้งสองข้าง

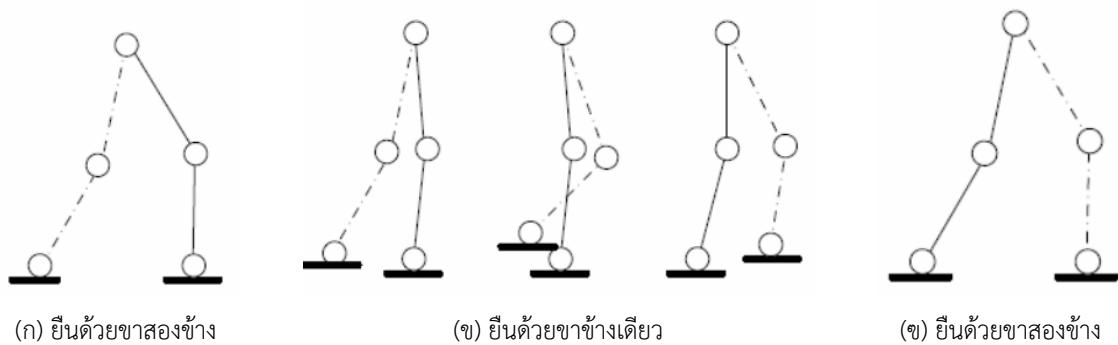
1) การยืนด้วยขาข้างเดียว : เป็นช่วงที่มีเท้าของหุ่นยนต์สัมผัสพื้นเพียงข้างเดียว ส่วนเท้าอีกข้างของ หุ่นยนต์จะถูกยกกลอยจากพื้น โดยที่ไม่มีส่วนใดๆของขาข้างนั้นสัมผัสกับพื้นเลย ช่วงนี้จะเกิดขึ้นเมื่อมีการแกว่งเท้า จากข้างหลังไปข้างหน้า ดังจากภาพที่ 2.5x

2) การยืนด้วยขาสองข้าง : เป็นช่วงที่เท้าทั้งสองข้างของหุ่นยนต์สัมผัสกับพื้น ช่วงนี้จะเกิดตั้งแต่หุ่น ยนต์วางเท้าขณะที่สั้นเท้าแตะกับพื้น ไปจนถึง ปลายเท้าของขาอีกข้างหลุดออกจากพื้น

การเดินได้โดยไม่มีล้มลุก ตัวหุ่นยนต์จะต้องรักษาสมดุลของการเดินให้ติดตลอดช่วงเวลาของการเดิน ซึ่ง สมดุลของการเดินแบบสองขาสามารถแบ่งตามลักษณะการเดินและการถ่ายน้ำหนักได้เป็น 2 รูปแบบหลัก คือ การเดินแบบสมดุลสถิต (static balance walking) และ การเดินแบบสมดุลพลวัต (dynamic balance walking)

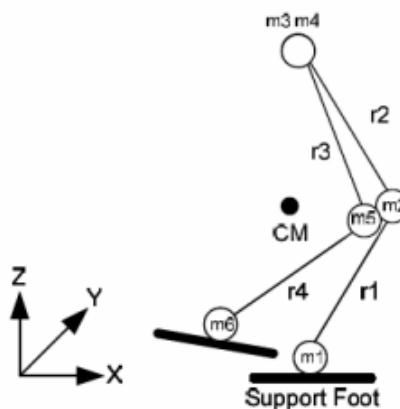
2.1.3.3 การสร้างและการควบคุมการเดินแบบสมดุลสถิต

การเดินของหุ่นยนต์ในลักษณะนี้ น้ำหนักตัวหุ่นยนต์จะไม่มีการเคลื่อนไหวออกนอกบริเวณฐานรับน้ำ หนัก (Supporting Area) ตลอดช่วงเวลาการเดิน ไม่ว่าจะเป็นช่วงเวลาที่รับน้ำหนักด้วยเท้าข้างเดียวหรือทั้งสอง



รูปที่ 2.5: วัสดุจัดการเดินของหุ่นยนต์ชีวภาพอยู่ด้วยวิธีนี้จะ

ข้างกีต้าม หมายความว่า โครงสร้างของหุ่นยนต์จะไม่ล้มแน่นอน เนื่องจากการสร้างรูปแบบการเดินด้วยวิธีนี้จะควบคุมให้ตำแหน่งของจุดรวมมวล (CoM) อยู่ภายในพื้นที่ฐานรับน้ำหนักของหุ่นยนต์ตลอดเวลา



รูปที่ 2.6: การควบคุมตำแหน่งของจุดรวมมวลให้อยู่ในพื้นที่ฐาน

ข้อดีของการสร้างและควบคุมการเดินของหุ่นยนต์ด้วยวิธีนี้คือ สามารถสร้างรูปแบบการเดินได้โดยที่มีความซับซ้อนไม่มากนัก สามารถสั่งให้หุ่นยนต์หยุดค้างในท่าทางใดๆ ก็ได้ตลอดเวลาโดยหุ่นยนต์ไม่ล้ม หุ่นยนต์ที่มีฝ่าเท้าใหญ่จะทำให้ง่ายต่อการก้าวเดินมากขึ้น นอกจากการควบคุมการก้าวขาแล้วอาจเพิ่มการควบคุมส่วนลำตัวเพิ่มเติม เพื่อเป็นการเพิ่มเสถียรภาพในการเดินและการถ่ายเท้น้ำหนัก โดยที่อาจจะมีการเพิ่มเซนเซอร์วัดแรงที่ฝ่าเท้าเพื่อตรวจสอบการกระจายแรงกดที่ฝ่าเท้า เพื่อตรวจสอบว่าตำแหน่งของจุดรวมน้ำหนักอยู่บนพื้นที่ฝ่าเท้าหรือไม่ หรือเพื่อตรวจสอบเสถียรภาพของการเดินเพื่อแก้ไขท่าทางการเดินไม่ให้เกิดการล้ม

ข้อเสียของการควบคุมการเดินด้วยวิธีนี้คือ หุ่นยนต์จะใช้เวลาในการก้าวเดินมาก ใช้พลังงานในการเดินมากกว่าการเดินแบบสมดุลพลวัต และท่าทางที่ได้จะมีความแตกต่างจากท่าทางการเดินของมนุษย์

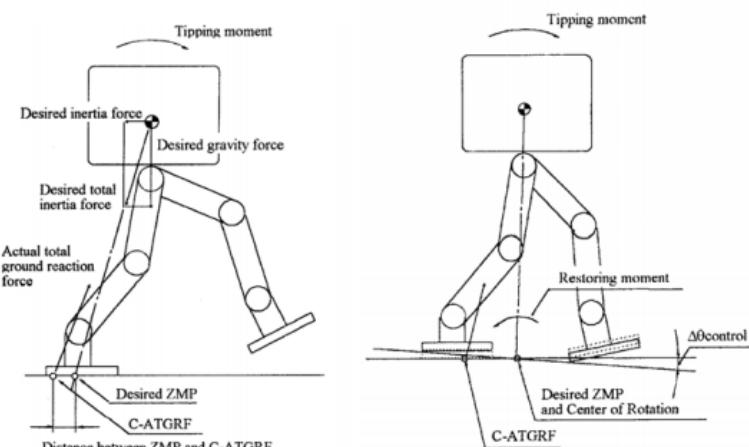
2.1.3.4 การสร้างและการควบคุมการเดินแบบสมดุลพลวัต

การสร้างรูปแบบการเดินและควบคุมการเดินในลักษณะนี้ท่าทางการเดินของหุ่นยนต์นั้นจะคล้ายกับการเดินของมนุษย์มากกว่าแบบสติต เนื่องจากมีหลักการในการสร้างท่าทางที่เหมือนกับการเดินของมนุษย์ซึ่งมีขั้นตอนดังนี้คือ เอียงตัวให้ล้มไปในทิศทางที่ต้องการเดิน เมื่อเริ่มเกิดการล้มขึ้นหุ่นยนต์จะเปลี่ยนตำแหน่งการวางเท้าไปยังตำแหน่งใหม่ เพื่อปรับให้โครงสร้างเข้าสู่ภาวะสมดุลอีกครั้ง

โดยธรรมชาติแล้วมนุษย์มีการถ่ายน้ำหนักในขณะที่เคลื่อนที่หรือยืนอยู่กับที่เพื่อรักษาสมดุลของท่าทาง

นั้นไว้ แต่หากการถ่ายโอนน้ำหนักนั้นเกิดสภาวะไม่สมดุล ร่างกายจะปรับสภาพโดยการเคลื่อนตำแหน่งของเท้า ซึ่งเป็นพื้นที่ฐานออกจากเดิมไปยังตำแหน่งใหม่ เพื่อรักษาสมดุลไว้ หลักการดังกล่าวถูกนำมาใช้กับการควบคุม การเดินของหุ่นยนต์อิวามาโนย์ ในขณะที่หุ่นยนต์กำลังเคลื่อนไหว ผลจากแรงเฉียบของการเคลื่อนที่และผลจาก แรงดึงดูดของโลกมีผลต่อการเพิ่มและลดความเร่งให้การเดินของหุ่นยนต์ แรงเหล่านี้เรียกว่าแรงเฉียบรวมของการ เคลื่อนที่ และเมื่อเท้าหุ่นยนต์สัมผัสกับพื้นจะได้รับผลกระทบของแรงนี้ เรียกว่า แรงปฏิกิริยาจากพื้น

การตัดกันระหว่างแรงปฏิกิริยาจากพื้นและแนวแรงเฉียบรวม ตำแหน่งน้ำหนักทำให้โมเมนต์เท้ากับศูนย์ เรียกจุดดั้นนี้ว่าจุดโมเมนต์ศูนย์ (ZMP_{robot}) และจุดที่แรงปฏิกิริยาลงสู่พื้นว่า จุดปฏิกิริยาพื้นฐาน ทำการเดินของหุ่นยนต์จะถูกกำหนดและถูกส่งให้กับชุดควบคุมข้อต่อจุดต่างๆ ของหุ่นยนต์ โดยให้สอดคล้องกับแรงเฉียบรวมที่เกิดขึ้นจากการคำนวณ เรียกว่าแรงเฉียบรวมเป้าหมาย และจุดโมเมนต์ศูนย์ที่ได้จากการคำนวณเรียกว่าจุด โมเมนต์ศูนย์เป้าหมาย (ZMP_{target}) เมื่อหุ่นยนต์เกิดสมดุลในขณะที่ทำการเดินได้อย่างสมบูรณ์ แนวแกนของแรงเฉียบรวมเป้าหมายและแรงปฏิกิริยาที่พื้นจะเป็นตำแหน่งเดียวกัน แต่ในขณะที่หุ่นยนต์เดินผ่านพื้นผิวที่มีความชรุกรายหรือไม่เรียบตำแหน่งสองจุดดังกล่าว จะไม่ใช่ตำแหน่งเดียวกันทำให้หุ่นยนต์เกิดการล้มได้ แรงที่ทำให้เกิด การล้มนี้เกิดจากตำแหน่งของจุดโมเมนต์ศูนย์และตำแหน่งแรงปฏิกิริยาร่วมที่พื้นไม่ตรงกัน ซึ่งเป็นสาเหตุหลักที่ ทำให้เกิดความไม่สมดุลขึ้น และเมื่อหุ่นยนต์เสียสมดุลระบบที่จะสามารถป้องกันการล้มและทำให้หุ่นยนต์เดินต่อไปได้อย่างต่อเนื่องคือ ระบบควบคุมแรงปฏิกิริยา ระบบควบคุมจุดโมเมนต์ศูนย์ และระบบควบคุมการวางแผนเท้า



รูปที่ 2.7: การควบคุมตำแหน่งของจุดโมเมนต์ศูนย์ให้ตรงกับแรงปฏิกิริยาร่วม

อย่างไรก็ตาม การสร้างท่าทางการเดินในลักษณะนี้ต้องใช้สมการในการคำนวณที่ซับซ้อนมาก เนื่องจาก ต้องหาความสัมพันธ์ระหว่างองค์ประกอบหลายส่วน เช่น น้ำหนักของโครงสร้างในแต่ละส่วน แรงบิดที่แต่ละข้อ ต่อ และโมเมนต์โดยรวมของระบบ นอกจากนี้ยังต้องใช้อุปกรณ์การตรวจวัดต่างๆ เช่น เชนเซอร์วัดแรง เชนเซอร์วัด มุม เชนเซอร์วัดแรงบิด ติดตั้งตามจุดต่างๆ ของโครงสร้างเพื่อวัดค่าอุปกรณ์ ก่อนที่จะทำการคำนวณตำแหน่ง และ สร้างท่าทางการเดินของหุ่นยนต์อิวามาโนย์ ท่าทางการเดินที่ได้จากการควบคุมด้วยวิธีนี้ จะมีความคล้ายคลึงกับ ท่าทางการเดินของมนุษย์มาก

2.1.3.5 จุดศูนย์กลางมวลของหุ่นยนต์

หากต้องการให้หุ่นยนต์สามารถที่จะทรงตัวอยู่ได้โดยไม่ล้มนั้น จึงต้องรู้ตำแหน่งจุดศูนย์กลางมวลของหุ่น ยนต์ตลอดเวลา และต้องให้จุดศูนย์กลางมวลชายฝั่งในบริเวณฐานรับน้ำหนักของหุ่นยนต์โดยหากจากพื้นที่ที่ฝ่าเท้า สัมผัสกับพื้น วิธีการนี้เป็นวิธีการทางสถิตศาสตร์

2.1.4 ตัวอย่างหุ่นยนต์ชีวามโนยด์

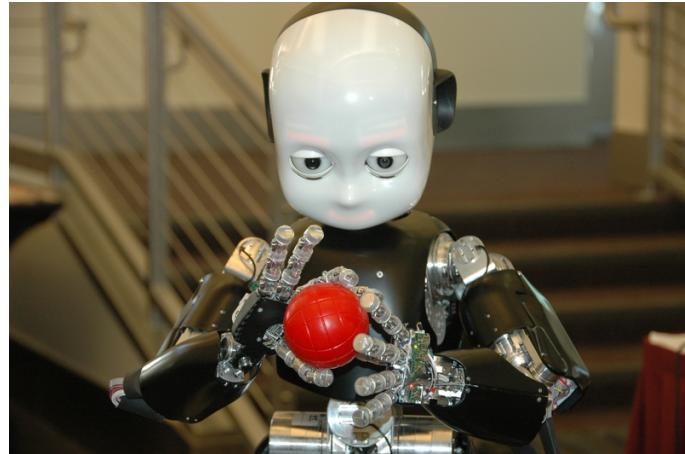
Poppy Humanoid



รูปที่ 2.8: หุ่นยนต์ชีวามโนยด์ป้อปปี้

หุ่นยนต์ชีวามโนยด์ป้อปปี้ ถูกสร้างขึ้นมาเพื่อใช้ในงานศิลปะ การวิจัยและการศึกษาโดยเฉพาะ หุ่นยนต์ป้อปปี้ประกอบด้วยส่วนของฮาร์ดแวร์และซอฟแวร์ที่เปิดเป็นโอเพนซอร์ซให้ผู้ที่สนใจสามารถเข้ามาศึกษาได้ โปรแกรมของหุ่นยนต์ป้อปปี้มีชื่อว่า Pypot ที่เป็นส่วนเสริมของภาษา Python ในการพัฒนาซอฟแวร์ ทุกคนสามารถเข้าถึงข้อมูลเชิงเทคนิคของหุ่นยนต์ชีวามโนยด์ป้อปปี้ได้ เช่น ส่วนรายละเอียดการทำงาน คลิปวิดีโอสอน การประกอบ การใช้ระบบจำลอง และการพัฒนาต่างๆ ผ่านทางเว็บไซต์ <http://www.poppy-project.org> หุ่นยนต์ป้อปปี้มีส่วนของโครงสร้างที่ผลิตมาจากพลาสติก PLA และ ABS โดยใช้เทคนิคการขึ้นรูปด้วยเครื่องพิมพ์สามมิติ ตัวขับเคลื่อนข้อต่อต่างๆ ใช้เป็น Dynamixel Digital Servo และควบคุมคำสั่งของตัวขับเคลื่อนด้วยคอมพิวเตอร์ขนาดเล็ก Odroid UX4 ใช้ระบบปฏิบัติการ Ubuntu 14.04 ตัวของหุ่นยนต์มีความสูง 83 เซนติเมตร น้ำหนัก 3.5 กิโลกรัม ใช้เซนเซอร์วัดมุ่งเอียง เป็น IMU ที่มีองศาอิสระเท่ากับ 9 องศาอิสระ ในกระบวนการควบคุมเสถียรภาพในการเดินของตัวเอง มีองศาอิสระหรือจำนวนตัวขับเคลื่อนทั้งหมด 25 องศา ประกอบไปด้วยขาข้างละ 6 องศาอิสระ และข้างละ 4 องศาอิสระ ลำตัว 3 องศาอิสระ และ หัว 2 องศาอิสระ

iCub Humanoid



รูปที่ 2.9: หุ่นยนต์อิวามานอยด์ไอคัพ

หุ่นยนต์อิวามานอยด์ไอคัพ ถูกออกแบบโดยมหาวิทยาลัยหลายแห่งในยุโรปรวมกลุ่มกันขึ้นมาในชื่อ RobotCub และถูกสร้างขึ้นโดย Istituto Italiano di Tecnologia (IIT) ตัวหุ่นยนต์ไอคัพนั้นมีความสูงอยู่ที่ 1 เมตร น้ำหนักโดยรวมทั้งหมดประมาณ 22 กิโลกรัม วัสดุที่ใช้ในการสร้างแตกต่างกันไปในแต่ละส่วนของร่างกายโดยจะใช้ aluminum alloy Al6082 สำหรับส่วนที่ต้องรับภาระความเครียดปานกลางถึงสูง และใช้ Stainless Steel 17-4PH ในส่วนของเหลาข้อต่อต่างๆเพื่อให้มีความแข็งแรงสูง ตัวหุ่นยนต์ถูกออกแบบให้มีลักษณะเหมือนเด็กอายุ 3-4 ขวบ ควบคุมโดยใช้บอร์ดไมโครคอนโทรเลอร์เป็นรุ่น PC104 Controller ภาษาที่ใช้ในการพัฒนาใช้เป็นภาษา C++ ในการเขียนโปรแกรม การติดต่อสื่อสารกับตัวขับเคลื่อนหรือมอเตอร์ตามข้อต่อต่างๆ และเซนเซอร์ ผ่านทาง_parallel CAN Bus เพื่อทำให้ใช้งานง่าย ใช้เส้นเอ็นในการส่งถ่ายแรงขับเคลื่อนไปยังส่วนของข้อต่อส่วนมือและขา นิ้วของหุ่นยนต์ถูกร้อยด้วย สายเคเบิลเคลือบ Teflon อยู่ภายใน และคลายตัวกลับสู่สภาพสมดุลได้ด้วยแรงของสปริง เซนเซอร์วัดมุมของข้อต่อแต่ละตัวใช้การออกแบบให้มี Hall-effect ติดอยู่ ช่วยในการอ่านค่าของตำแหน่งและความเร็วที่เกิดขึ้นที่ข้อต่อ หุ่นยนต์ไอคัพมีองศาอิสระรวมกันทั้งหมด 53 องศาอิสระ ประกอบไปด้วย แขนข้างละ 7 องศา อิสระ มือข้างละ 9 องศาอิสระ หัว 6 องศาอิสระ ลำตัว 3 องศาอิสระ และขาข้างละ 6 องศาอิสระ ในส่วนของหัวจะประกอบไปด้วย กล้องสองตัวเพื่อทำการติดตามวิธีการมองเห็น ไมโครโฟนสำหรับรับเสียงจากสภาพแวดล้อมภายนอก และไฟแสดงอารมณ์บริเวณปากและคิ้ว หุ่นยนต์นี้ไม่ได้ถูกออกแบบให้มีการทำงานเป็นแบบอัตโนมัติ ซึ่งก็คือตัวหุ่นยนต์นั้นไม่มีแบตเตอรี่ภายในตัว แต่ใช้แหล่งพลังงานจากการส่งเข้าไปผ่านสายเคเบิล และเชื่อมต่อกับอินเทอร์เน็ตผ่านสายแลน (LAN)

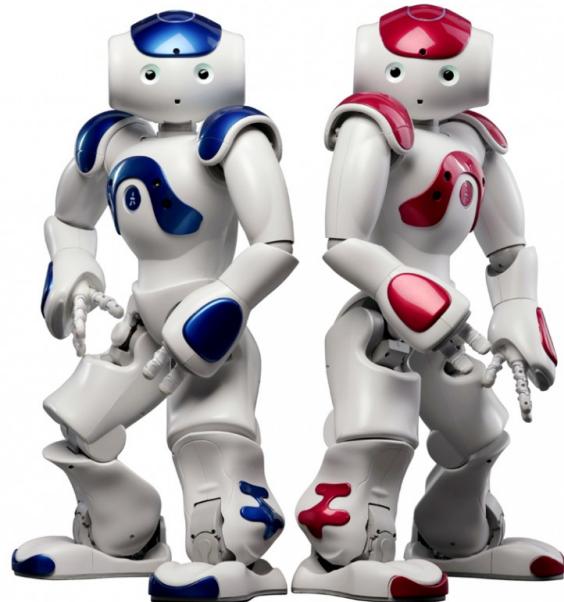
Darwin-OP Humanoid



รูปที่ 2.10: หุ่นยนต์อิวามานอยด์ดาร์วิน

หุ่นยนต์อิวามานอยด์ดาร์วิน (Darwin-OP) เป็นชื่อที่ย่อมาจากคำว่า Dynamic Anthropomorphic Robot with Intelligence-Open Platform เป็น OpenSource Platform ที่ถูกออกแบบและพัฒนาโดย Korean robot manufacturer Robotis โดยมีความร่วมมือกับ Virginia polytechnic institute and state university, Purdue university และ University of Pennsylvania หุ่นยนต์อิวามานอยด์ดาร์วินมีความสามารถในการรับภาระโหลดได้สูง เนื่องจากมีการพัฒนามอเตอร์เป็นของตัวเอง อีกทั้งยังมีความสามารถในการเคลื่อนที่แบบ พลวัต (Dynamic) หุ่นยนต์ดาร์วิน มีองศาอิสระทั้งหมด 20 องศาอิสระ ซึ่งประกอบไปด้วย ขาข้างละ 6 องศาอิสระ แขนข้างละ 3 องศาอิสระ และหัว 2 องศาอิสระ ขับเคลื่อนข้อต่อต่างๆด้วยเซอร์โวมอเตอร์ Dynamixel MX-28T ที่มีการเชื่อมต่อแบบ RS485 ในการประยัดสายที่ใช้ในการสั่งการ มอเตอร์แต่ละตัวมีเซนเซอร์วัดตำแหน่ง และความเร็วอยู่ภายใน ตัวหุ่นยนต์มีความสูงทั้งหมด 45 เซนติเมตร มีน้ำหนักโดยประมาณ 2.9 กิโลกรัม ระบบภายในใช้คอมพิวเตอร์ขนาดเล็กเป็น 1.6 GHz Intel Atom Z530 (32 bit) ใช้คอนโทรลเลอร์ ARM CortexM3 STM32F103RE 72 MHz และมีเซนเซอร์วัดมุมเอียงเป็น 3-axis gyro, 3-axis accelerometer เพื่อช่วยในการควบคุมเสถียรภาพในการเดิน

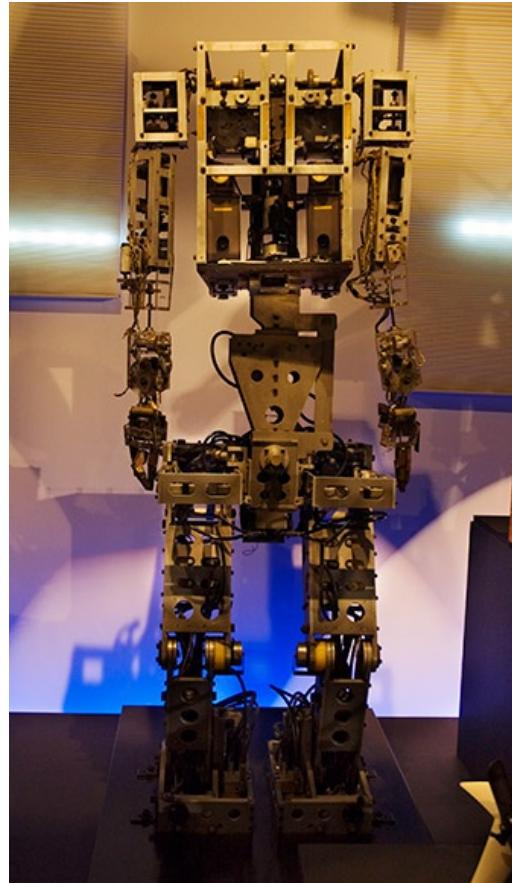
Nao Humanoid



รูปที่ 2.11: หุ่นยนต์ชีวามโนยด์นาโอะ

หุ่นยนต์ชีวามโนยด์นาโอะ เป็นหุ่นยนต์ชีวามโนยด์ขนาดกลาง ถูกผลิตมาจากประเทศฝรั่งเศษ พัฒนาโดย บริษัท Aldebaran Robotics เมื่อปี 2004 และในปี 2007 หุ่นยนต์ชีวามโนยด์นาโอะได้นำไปแทนที่หุ่นยนต์สุนัข ของ Sony ซึ่ง Aibo ขณะนั้นใช้ในการแข่งขัน RoboCup Standard Platform League (SPL) หุ่นยนต์นาโอะได้ถูกนำไปใช้ใน Robocup 2008 และ 2009 หุ่นยนต์นาโอะถูกพัฒนาออกแบบอย่างหลากหลายรุ่น มีองศาอิสระตั้งแต่ 14 องศาอิสระ 21 องศาอิสระ และ 25 องศาอิสระ สำหรับเพื่องานวิจัยนั้นมีถึง 25 องศาอิสระ โดยเพิ่มเติมมือสอง ข้างเอวเข้าไปเพื่อให้สามารถหยิบจับสิ่งของได้ ภายในหุ่นยนต์ถูกควบคุมด้วยระบบปฏิบัติการ NAO 2.0 (Linux-based) ตัวหุ่นยนต์มีความสูง 58 เซนติเมตร น้ำหนัก 4.3 กิโลกรัม ส่วนเซนเซอร์การรับรู้ต่างๆ จะประกอบไปด้วย เซนเซอร์วัดมุมเอียง 3-axis gyro, 3-axis accelerometer, Ultrasound captors, ไมโครโฟน 4 ตัว ลำโพง 2 ตัว กล้อง 2 ตัว เพื่อใช้ประโยชน์ในการทำงานวิจัยต่างๆ ตอนนี้ความสามารถของหุ่นยนต์นาโอะที่ทำได้คือ สามารถเห็นสีได้ เดินขึ้นลงบันไดและทางลาดชันได้ ระหว่างการเดินนั้นสามารถวางแผนการเดินทางที่ได้อย่างรวดเร็ว อีกทั้ง ยังสามารถที่จะเดินหลบหลีกสิ่งกีดขวางได้ด้วย

Wabot



รูปที่ 2.12: หุ่นยนต์อิวามานอยด์ว้าบอท

หุ่นยนต์อิวามานอยด์มีการพัฒนาในช่วงแรกเริ่มมาตั้งแต่ปี 1973 หุ่นยนต์อิวามานอยด์ ตัวแรกชื่อ Wabot-1 เริ่มสร้างโดยมหาวิทยาลัย Waseda ที่ประเทศญี่ปุ่น ตัวของหุ่นยนต์มีความสูง 180 เซนติเมตร น้ำหนัก 210 กิโลกรัม โดยหุ่นยนต์สามารถติดต่อสื่อสารกับมนุษย์ได้ด้วยภาษาญี่ปุ่น สามารถวัดระยะและทิศทางได้โดยใช้การรับรู้ผ่านทางตาและหูเทียม หุ่นยนต์ Wabot-1 นั้นสามารถเดินได้ด้วยขาของตนเองที่มีส่องข้าง สามารถเหยียบและเคลื่อนย้ายวัตถุด้วยมือ ต่อมาในปี 1984 มหาวิทยาลัย Waseda ได้พัฒนาหุ่นยนต์อิวามานอยด์ที่ชื่อ Wabot-2 โดยหุ่นยนต์สามารถสื่อสารกับมนุษย์ได้ สามารถอ่านโน้ตเพลงและเล่นดนตรีโดยใช้ electronic organ แบบง่ายๆ ได้ และในปี 1985 บริษัท Hitachi ได้สร้างหุ่นยนต์ WHL-11 ที่มีส่องขาเหมือนมนุษย์ ซึ่งสามารถเดินแบบสมดุล สติ๊ต (Static Walking) บนพื้นราบได้ด้วยความเร็ว 13 วินาทีต่อหนึ่งก้าว และสามารถเลี้ยวได้ซ้ายและขวาได้

2.2 การออกแบบโครงสร้างของหุ่นยนต์

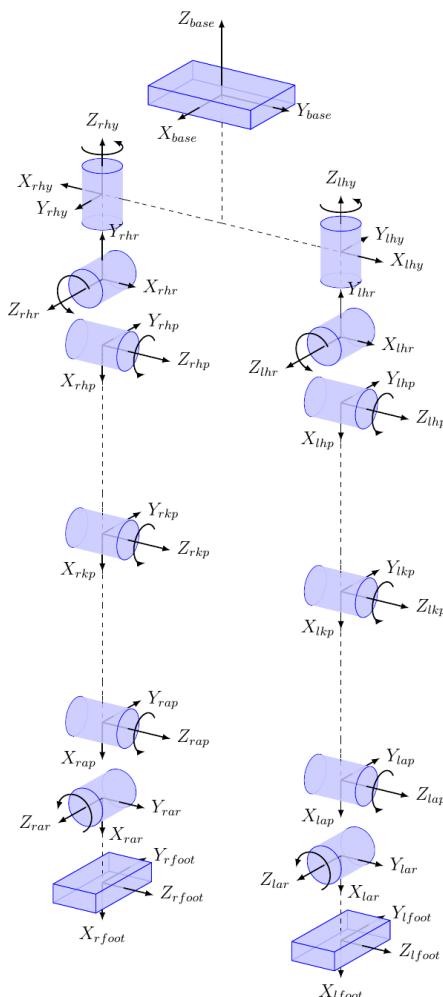
2.2.1 ความแตกต่างระหว่างโครงสร้างของมนุษย์กับโครงสร้างของหุ่นยนต์

2.2.1.1 ความแตกต่างขององค์การเสรี

เนื่องจากลักษณะข้อต่อของมนุษย์มีความซับซ้อนมากกว่าโครงสร้างของหุ่นยนต์ ทำให้ข้อต่อแต่ละจุดของมนุษย์นั้นสามารถหมุนได้หลายทิศทาง รวมถึงขอบเขตของการหมุนของข้อต่อในแต่จุดก็มีความแตกต่างกัน ในการนำรูปแบบการเดินของมนุษย์ไปใช้กับหุ่นยนต์จึงต้องปรับค่ามุมที่ข้อต่อให้มีความเหมาะสมกับโครงสร้าง และข้อจำกัดเกี่ยวกับการหมุนของข้อต่อจุดต่างๆ ของหุ่นยนต์ที่จะใช้ทดสอบด้วย

2.2.1.2 ความแตกต่างของอัตราส่วน

นอกจากความแตกต่างขององค์การเสรี (DoF) ระหว่างมนุษย์กับหุ่นยนต์แล้ว ความแตกต่างของอัตราส่วนระหว่างโครงสร้างแต่ละส่วนของมนุษย์กับหุ่นยนต์เป็นอีกสาเหตุหนึ่ง ที่ต้องทำการปรับแต่งใหม่มีความเหมาะสม เนื่องจากความยาวของโครงสร้างแต่ละส่วน รวมทั้งระยะห่างระหว่างจุดหมุนแต่ละจุดของมนุษย์กับหุ่นยนต์ที่มีความแตกต่างกัน ดังนั้นจึงต้องกำหนดระบบพิกัดสำหรับหุ่นยนต์ขีมานอยด์ขึ้นมาใหม่ เพื่อใช้ในการอ้างอิงจุดหมุน และความยาวของโครงสร้างในส่วนต่างๆ

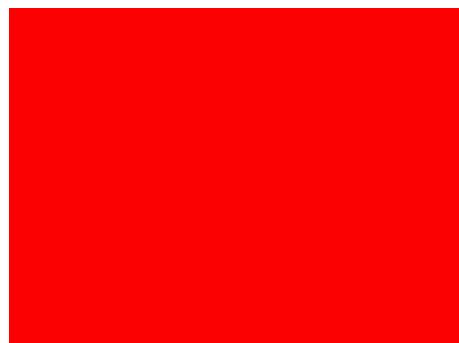


รูปที่ 2.13: ตัวอย่างตำแหน่งและการหมุนของข้อต่อของหุ่นยนต์เพื่อการอ้างอิง

2.2.1.3 กำลังและประสิทธิภาพของมอเตอร์

ความสามารถในการรับน้ำหนักของข้อต่อแต่ละจุดมีความแตกต่างกัน การเคลื่อนไหวของมนุษย์นั้นจะมีกล้ามเนื้อ และเส้นเอ็นเป็นตัวออกแรงดึงส่วนต่างๆของร่างกายเพื่อทำให้เกิดการเคลื่อนไหวซึ่งจะมีความยืดหยุ่น และแรงดึงที่มีค่าสูง สำหรับการเคลื่อนไหวของหุ่นยนต์ จะใช้การบิดแกนของเซอร์โวมอเตอร์ (Servo Motor) หรือมอเตอร์ที่ติดอยู่ที่ข้อต่อจุดต่างๆ ทำให้ความสามารถในการรับน้ำหนัก แรงบิดและความยืดหยุ่นที่ข้อต่อขึ้น กับกำลังของมอเตอร์เป็นหลัก การสร้างท่าทางของหุ่นยนต์จึงต้องคำนึงถึงความสามารถในการรับน้ำหนักและ กำลังของเซอร์โวมอเตอร์ที่ใช้ด้วยเข่นกัน

2.2.2 วัสดุและการขึ้นรูปโครงสร้างของหุ่นยนต์อิวามาโนย์



รูปที่ 2.14: รูปแก้ไข

2.2.3 อุปกรณ์ที่ใช้ในหุ่นยนต์อิวามาโนย์

2.2.3.1 ระบบจ่ายพลังงาน

2.2.3.2 ตัวขับเคลื่อน

2.2.3.3 หน่วยประมวลผลควบคุม

ในการควบคุมหุ่นยนต์อิวามาโนย์ให้สามารถทำกิจกรรมต่างๆได้ ด้านนี้ ส่วนที่มีความสำคัญที่ขาดไปไม่ได้ คือ หน่วยประมวลผลระบบควบคุม ถ้าหากไม่มีระบบประมวลผลควบคุมแล้ว อุปกรณ์ต่างๆ ที่ติดตั้งอยู่ภายในตัวของหุ่นยนต์อิวามาโนย์จะไม่สามารถติดต่อสื่อสารกันได้ ซอฟต์แวร์ของหุ่นยนต์ที่พัฒนามาทั้งหมดจะไม่สามารถใช้ได้ ทำให้หุ่นยนต์อิวามาโนย์ไม่สามารถทำงานในสิ่งที่ต้องการได้ การวางแผนระบบควบคุมที่นิยมใช้ในระบบหุ่นยนต์อิวามาโนย์ส่วนใหญ่ จะแบ่งออกเป็น 2 ส่วนตัวยังกัน คือส่วนของหน่วยประมวลผลควบคุมระดับสูง และหน่วยประมวลผลควบคุมระดับต่ำ

หน่วยประมวลผลควบคุมระดับสูง (High level controller)

หน่วยประมวลผลควบคุมระดับสูงเป็นส่วนที่ใช้ประมวลผลการทำงานที่มีความซับซ้อนของระบบ เช่น จตุรัสตร์ของหุ่นยนต์ การคำนวณหาเส้นทางการเดิน ในการคำนวณทางคณิตศาสตร์ของระบบเหล่านี้จำเป็น ต้องมีการประมวลผลที่เร็ว และมีประสิทธิภาพ ย้อนไปในสมัยที่มีการพัฒนาหุ่นยนต์อิวามาโนย์ยุคแรกเริ่มนั้น หน่วยประมวลผลควบคุมระดับสูง จะใช้คอมพิวเตอร์เป็นตัวในการประมวลผลการคำนวณ ซึ่งคอมพิวเตอร์สมัยนั้น มีขนาดใหญ่ น้ำหนักมาก และต้องใช้หลังงานสูง ซึ่งต่างจากปัจจุบันนี้ที่มีการพัฒนาของเทคโนโลยีที่ก้าวหน้ามากขึ้น ทำให้คอมพิวเตอร์มีขนาดเล็กลงเทียบเท่ากับบอร์ดคอนโทรลเลอร์ทั่วไป ในที่นี้จะทำการยกตัวอย่างของบอร์ดคอมพิวเตอร์ที่มีวงจรนำร่องในปัจจุบัน และทำการรวมรวมเทียบเคียงประสิทธิภาพ ของบอร์ดคอมพิวเตอร์ แต่ละชนิดไว้ดังนี้



รูปที่ 2.15: รอยแก้ไข

หน่วยประมวลผลควบคุมระดับต่ำ (Low level controller)

หน่วยประมวลผลควบคุมระดับต่ำเป็นส่วนที่รับคำสั่งบางอย่างมาจาก หน่วยประมวลผลควบคุมระดับสูง มีประสิทธิภาพในการประมวลผลการคำนวณที่น้อยกว่า เนื่องจากการออกแบบสถาปัตยกรรมภายในระบบไม่เอื้อ อำนวยต่อการคำนวณที่มีความซับซ้อน แต่มีความสามารถในการประมวลผลระบบที่เป็นคาบได้อย่างแม่นยำ ใน ด้านการทำหุ่นยนต์ชีวามโนยด์นั้nm กจะใช้หน่วยประมวลผลควบคุมระดับต่ำ ในการติดต่อกับอุปกรณ์ต่าง ๆ บนตัวของหุ่นยนต์ชีวามโนยด์โดยตรง เช่น ตัวขับเคลื่อน เชนเชอร์รับค่า หรือไฟแสดงสถานะต่างๆของหุ่นยนต์ จากตารางข้างต้นเป็นตารางเปรียบเทียบประสิทธิภาพและความเหมาะสมกับการใช้งาน จะเห็นได้ว่า Nucleo มี



รูปที่ 2.16: รอยแก้ไข

ประสิทธิภาพมากกว่าในหลายๆด้านไม่ว่าจะเป็น Core Clock ที่เร็วถึง 100 MHz หรือ GPIO ที่มีมากให้ 50 ช่อง การเข้าออกต่อ อีกทั้งยังมี I2C ซึ่งเป็นรูปแบบที่ใช้ในการติดต่อกับ IMU ที่ต้องใช้ หังนี้ หัง 2 รุ่นที่ได้ทำการเปรียบ เทียบกันไม่รองรับ RS – 485 โดยตรง ซึ่งเป็นรูปแบบการสื่อสารที่จะใช้กับการติดต่อกับตัวขับเคลื่อน

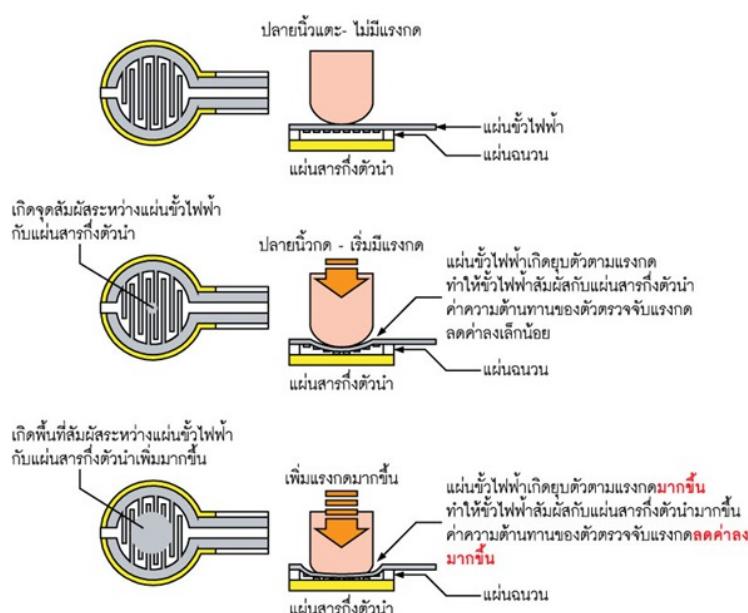
2.2.3.4 เชนเชอร์ตรวจหน้าสัมผัสที่พื้น

เชนเชอร์ตรวจหน้าสัมผัสที่พื้นเป็นเชนเชอร์ที่ถูกติดตั้งบริเวณฝ่าเท้า เพื่อตรวจสอบการเดินของหุ่นยนต์ ชีวามโนยด์ว่าขณะนี้มีการสัมผัสของฝ่าเท้าของหุ่นยนต์กับพื้นหรือไม่ ซึ่งในงานวิจัยนี้ได้ใช้หลักการตัวตรวจจับ แรงกดแบบค่าความด้านทานหรือ Force Sensing Resistor (FSR) ที่ใช้เทคโนโลยีฟิล์มโพลีเมอร์แบบหนาโดยที่ เชนเชอร์สามารถเปลี่ยนแรงที่มากระทำให้อยู่ในรูปของการเปลี่ยนแปลงค่าความด้านทานไฟฟ้า ตัวเชนเชอร์มี ลักษณะเป็นแผ่น มีโครงสร้าง 5 ชั้น โดยสองชั้นนอกสุดเป็นฟิล์มของโพลีเอสเตอร์ ส่วนชั้นถัดเข้ามาเป็นฟิล์มของ โลหะที่เป็นตัวนำไฟฟ้า และชั้นในสุดเป็นหมึกที่มีความไวในการตอบสนองต่อแรงกดจากที่มากระทำ (Pressure sensitive ink) และโครงสร้างทั้ง 5 ชั้น ถูกรวมเข้าด้วยกันด้วยวิธีลามิเนท จึงทำให้เชนเชอร์วัดแรงนี้มีลักษณะ

แบบมีความยืดหยุ่นสูง ด้วยเหตุนี้จึงทำให้เซนเซอร์สามารถโค้งงอได้ง่าย แรงดันไฟฟ้าที่ต่อกคร่วงตัวตรวจจับจะลดลง เมื่อมีแรงกดมากระทำบนแผ่นตรวจจับ มีโครงสร้างของตัวตรวจจับแสดงในรูปที่ 2.17



รูปที่ 2.17: ลักษณะโครงสร้างของตัวตรวจจับแรงกด FSR



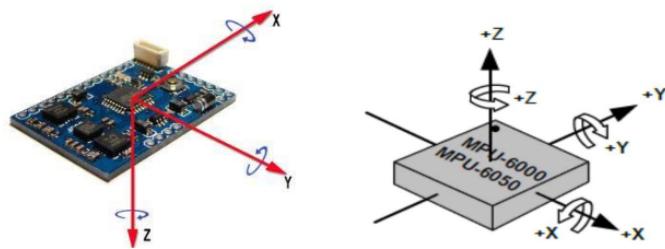
รูปที่ 2.18: การทำงานของตัวตรวจจับแรงกด FSR

2.2.3.5 เซนเซอร์วัดความเร็ว

Inertial Measurement Unit (IMU) เป็นส่วนประกอบหลักที่ใช้ในการนำร่องเครื่องบิน ยาน-อวกาศ ดาวเทียม เรือ ขึ้นไปรบกวน ซึ่งในตัวของ IMU ประกอบไปด้วยสองส่วนหลักคือ Accelerometers 3 ทิศทาง ในการรับความเร่งเชิงเส้น และ Gyroscopes 3 ทิศทาง ในการบอกความเร็วเชิงมุม เซนเซอร์ตัวนี้สามารถนำมาใช้ในการหาทิศทางการหมุนของตัวหุ่นยนต์ชิวามนอยด์ได้

เซนเซอร์วัดความเร็ว (Gyroscope) เป็นอุปกรณ์สำหรับการวัดความเร็ว หรือการรักษาการปรับทิศทาง ขึ้นอยู่กับหลักการของการอนุรักษ์โมเมนตัมเชิงมุม ถ้าไม่มีการเคลื่อนที่ อัตราการเปลี่ยนแปลงมุมจะมีค่าเท่ากับศูนย์

เซนเซอร์วัดความเร่ง (Accelerometer) เป็นอุปกรณ์ที่ใช้วัดความเร่งเชิงเส้น โดยอาศัยการวัดแรงที่กระทำต่อน้ำหนัก ถ้าอิงที่เกิดจากแรงโน้มถ่วงโลก ซึ่งแรงโน้มถ่วงของโลกจะเป็นเวกเตอร์ซึ่งไปที่แกนกลางโลกเสมอ ตามกฎของนิวตัน



รูปที่ 2.19: เซนเซอร์วัดความเร็ว

2.2.4 แนวคิดการออกแบบกลไกการเดินของหุ่นยนต์สิ่วมานอยด์

การออกแบบหุ่นยนต์สิ่วมานอยด์ให้สามารถเดินสองขาได้สม่ำเสมอ มุ่งเน้นให้ใช้จำนวนของศาอิสระให้เท่ากับมนุษย์นั้นพบว่า มีข้อจำกัดทางด้านการออกแบบแบบอยู่มาก เนื่องมาจากอุปกรณ์ที่ใช้ในการขับเคลื่อนข้อต่อต่างๆ มีอยู่อย่างจำกัด รวมถึงข้อจำกัดทางด้านตัวรับรู้ตัวขับของหุ่นยนต์ ดังนั้นผู้จัดทำจึงออกแบบหุ่นยนต์ให้มีองศาอิสระของข้อต่อ ในขนาดนึงข้าง เท่ากับหกองศาอิสระ ทั้งนี้หุ่นยนต์ยังสามารถเคลื่อนที่ได้ในปริภูมิ และองศาอิสระเพียงพอต่อการใช้งาน

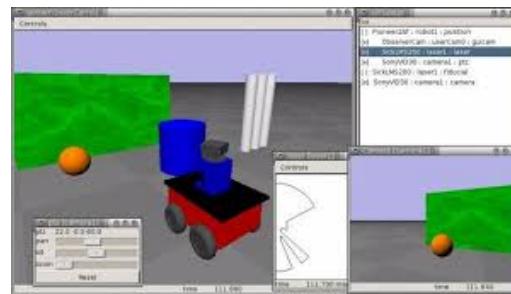
2.3 การออกแบบโปรแกรมด้วย ROS

2.3.1 ระบบที่ใช้ช่วยในการพัฒนาหุ่นยนต์

Robot Middleware เป็นกรอบการทำงาน(framework) ที่มีความยืดหยุ่นสำหรับการพัฒนาซอฟแวร์ที่ซับซ้อนในการควบคุมของหุ่นยนต์ ตัว Robot Middleware ถูกออกแบบมาให้ใช้ในการจัดการระบบที่มีความยุ่งยาก โดยมีเครื่องมือที่ช่วยติดต่อสื่อสารระหว่างอุปกรณ์ต่างๆของหุ่นยนต์ Robot Middleware ส่วนใหญ่จะใช้การติดต่อสื่อสารผ่านระบบเครือข่ายเน็ตเวิร์ก ทำให้การสื่อสารในระบบพื้นฐานเป็นอิสระต่อกัน และสามารถติดต่อสื่อสารกันกับอุปกรณ์ที่อยู่ภายนอกผ่านเครือข่ายเดียวกันได้

ปัจจุบันมี Robot Middleware ที่ถูกพัฒนาขึ้นมาให้ใช้อยู่หลายตัว เช่น

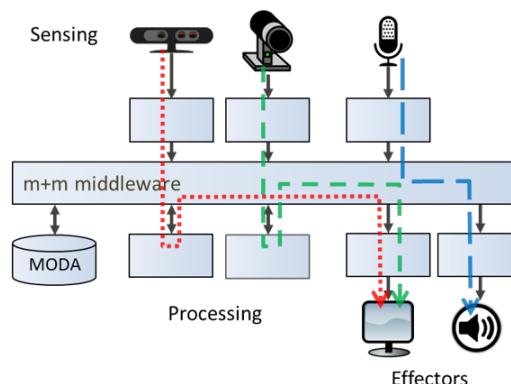
Player Project



รูปที่ 2.20: player project middleware

เป็นโครงการที่ใช้ในการสร้างซอฟแวร์เพื่อการศึกษาวิจัยที่มีความเกี่ยวข้องกับหุ่นยนต์และระบบเซ็นเซอร์ภายในประกอบไปด้วยระบบตัวกลาง และระบบจำลองการทำงานของหุ่นยนต์

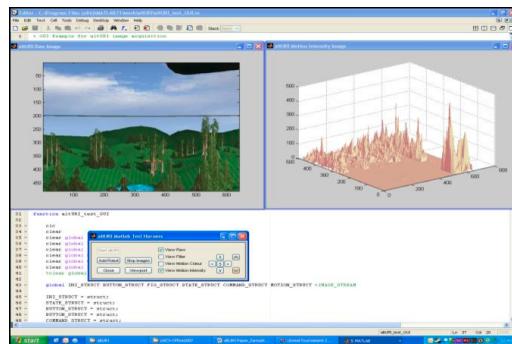
YARP



รูปที่ 2.21: yarp middleware

เป็น open source ที่เขียนด้วยภาษา C++ ในการเชื่อมต่อ กับเซนเซอร์ หน่วยประมวลผล และตัวขับเคลื่อนของหุ่นยนต์

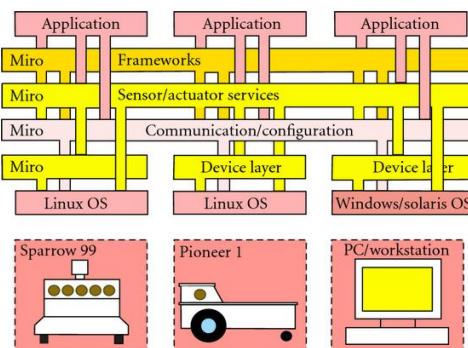
URBI



รูปที่ 2.22: urbi middleware

เป็น open source สำหรับพัฒนาซอฟต์แวร์ที่เกี่ยวข้องกับหุ่นยนต์หรือระบบที่มีความซับซ้อนใช้ภาษาพื้นฐานเป็นภาษา C++ ติดต่อสื่อสารได้ภายในเครือข่ายเดียวกันเท่านั้น (Local Network)

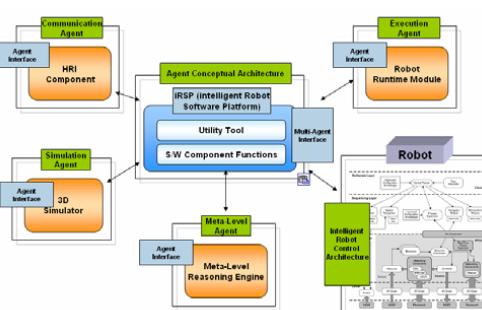
MIRO



รูปที่ 2.23: miro middleware

เป็นกรอบการทำงานของหุ่นยนต์ที่เคลื่อนที่โดยเขียนในลักษณะเป็น OOP

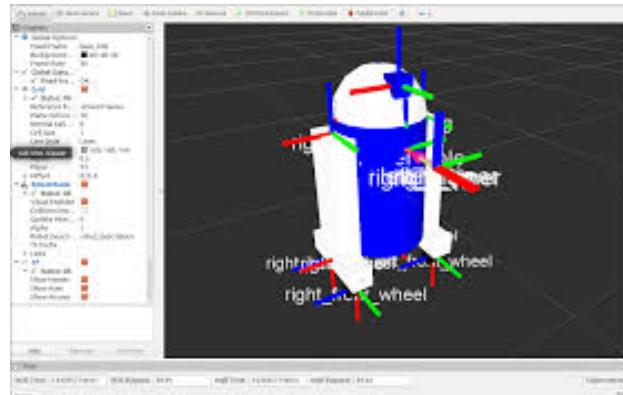
OpenRDK



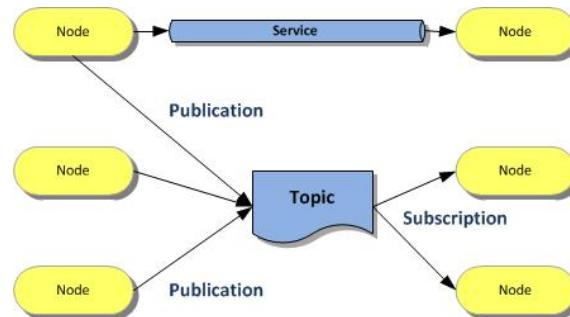
รูปที่ 2.24: openrdk middleware

เป็น open source สำหรับพัฒนาระบบที่มีความเป็นอิสระต่อกัน (Modules) สามารถใช้ช่องทางการติดต่อสื่อสารและหน่วยความจำร่วมกันได้

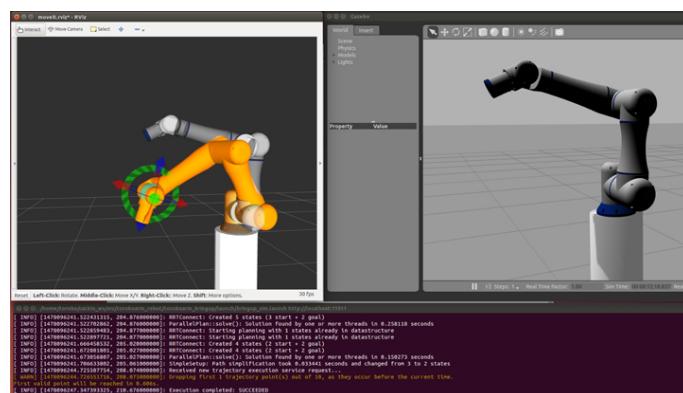
ROS



รูปที่ 2.25: ROS middleware Rviz



รูปที่ 2.26: ROS algitecture



รูปที่ 2.27: ROS Moveit

Robot Operating System หรือ ROS ถูกพัฒนาโดยบริษัท Willow Garage, แต่เดิมแล้วค้าพัฒนาเพื่อใช้ งานกับหุ่นยนต์ PR2 ในปี 2007 ซึ่งพัฒนาเป็น open source framework สำหรับนักพัฒนาซอฟแวร์ที่เกี่ยวข้อง กับหุ่นยนต์ มีความสามารถในการทำงานแบบ parallel บนคอมพิวเตอร์หลายเครื่องได้ สามารถทำงานได้หลาย OS นอกจากนี้ยังมีคลังที่ค่อยๆเก็บซอฟแวร์ต่างๆไว้เป็น libraries อีกด้วย การใช้ ROS จะช่วยทำให้เราสามารถพัฒนาหุ่นยนต์ได้อย่างรวดเร็วมากขึ้น ประหยัดเวลา ประหยัดทรัพยากร

2.3.2 ระบบที่ใช้ในการจำลองการทำงานของหุ่นยนต์

โปรแกรมจำลองการทำงานของหุ่นยนต์นั้นเป็นเครื่องมือที่สำคัญสำหรับนักหุ่นยนต์ การใช้โปรแกรมจำลองนั้นจะช่วยเพิ่มประสิทธิภาพในการทำงานหลายอย่าง เช่น ให้รู้ว่าหุ่นยนต์ที่ออกแบบนั้นสามารถทำงานได้อย่างที่ต้องการหรือไม่ กระบวนการคิดถูกต้องหรือไม่ โปรแกรมจำลองระบบส่วนใหญ่จะคำนวณพลวัตของหุ่นยนต์โดยใช้เครื่องมือคำนวณ open dynamics engine (ODE)

USARSim



รูปที่ 2.28: ผลลัพธ์จากการใช้โปรแกรม USARSim

USARSim เป็นโอเพนซอร์ซและเหมาะสมสำหรับทำหุ่นยนต์ประเภทกู้ภัยในชากเมือง โดยมีฐานการพัฒนามาจาก Unreal Tournament game engine ภายใต้โปรแกรมมีเครื่องมือสำหรับการทำงานวิจัย มีเซนเซอร์ของหุ่นยนต์ที่หลากหลาย เช่น เซนเซอร์รับภาพ หรือเซนเซอร์ตรวจความเคลื่อนไหว

MuRoSimF



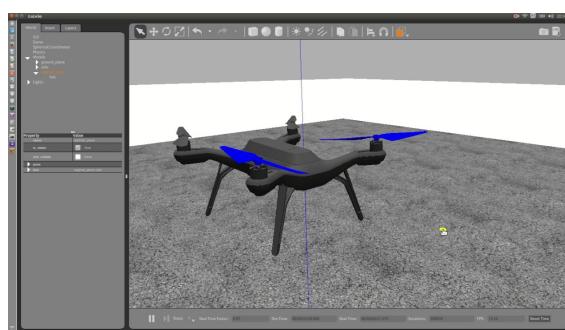
รูปที่ 2.29: ผลลัพธ์จากการใช้โปรแกรม MuRoSimF

MuRoSimF ย่อมาจากคำว่า Multi-Robot Simulation Framework เป็นเครื่องมือที่ช่วยทำระบบจำลองจาก Darmstadt University โปรแกรมระบบจำลองนี้มีการใช้งานที่ง่าย เหมาะสำหรับหุ่นยนต์หลายประเภท เช่น หุ่นยนต์เคลื่อนที่ด้วยล้อ หุ่นยนต์สองขา หรือหุ่นยนต์หลายขา สามารถคำนวณพลวัต และการขัดกันของก้านต่อต่างๆได้

Gazebo



รูปที่ 2.30: Mobile robot with gazebo



รูปที่ 2.31: Quadrotor with gazebo

Gazebo เป็นโปรแกรมจำลองการทำงานของหุ่นยนต์ ที่มีความสามารถในการคำนวณการเดินและการเคลื่อนที่ของหุ่นยนต์ที่สลับซับซ้อนได้ สามารถเห็นภาพกราฟฟิคของหุ่นยนต์ขณะทำงาน โดยผู้ใช้สามารถกำหนดค่าตัวแปรทางฟิสิกส์ต่าง ๆ ได้ เช่น น้ำหนัก ค่าความเร็ว แรงเสียดทานของข้อต่อ ทำให้การออกแบบหุ่นยนต์หรือทดลองโปรแกรมได้เหมือนกับโลกจริง มีแสง มีเงา และ พื้นผิวของวัตถุ และที่พิเศษคือสามารถสังเคราะห์ค่าของเซนเซอร์ เช่นเซอร์พร้อมสัญญาณรบกวน ค่าระยะทาง แรงบิด และอื่นๆ คำนวณพลศาสตร์ของหุ่นยนต์โดยใช้ตัวคำนวณทางฟิสิกส์เป็น Bullet หรือ Simbody ใน การจำลองหุ่นยนต์ในโปรแกรมนี้จำเป็นต้องได้รับไฟล์ข้อมูลของหุ่นยนต์มาก่อนซึ่งอยู่ในรูปแบบของ URDF ซึ่ง URDF คือ ประเภทของไฟล์ที่บ่งบอกถึงความสัมพันธ์ของข้อต่อและก้านต่อแต่ละชิ้นในตัวหุ่นยนต์ มีความสามารถในการอธิบายถึงกลศาสตร์และการเคลื่อนที่ของหุ่นยนต์ รวมถึงตรวจสอบการขัดกันของก้านต่อในหุ่นยนต์ได้ ภายในไฟล์นี้จะประกอบไปด้วย

Link: คือก้านต่อของหุ่นยนต์ซึ่งภายในจะสามารถบอกขนาด รูปร่าง สี และสามารถ import 3d mesh เข้ามาได้ด้วย อีกทั้งยังสามารถใส่รายละเอียดของการเคลื่อนที่ ของก้านต่อได้ เช่น inertial matrix และ collision properties

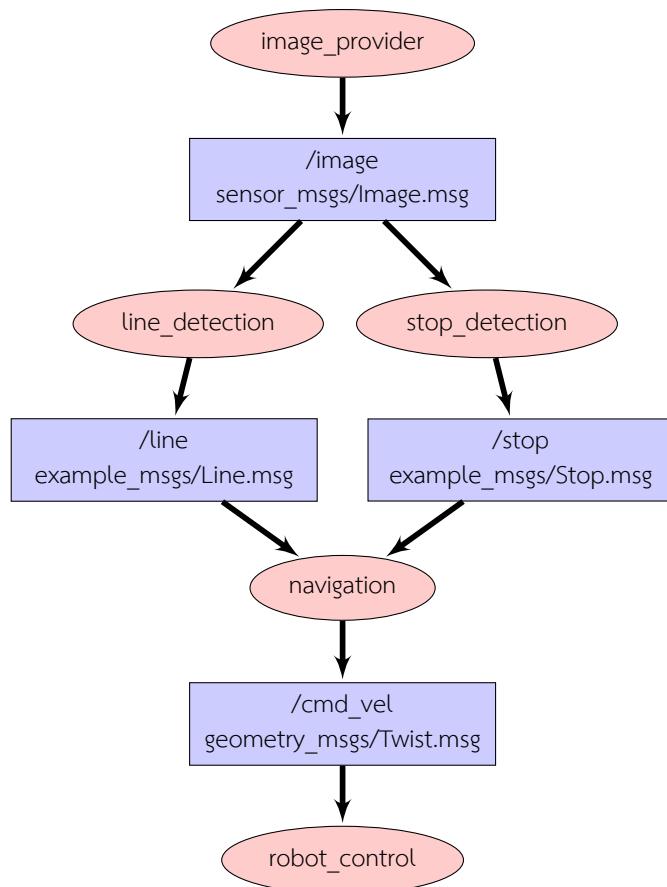
Joint: คือข้อต่อของหุ่นยนต์สามารถกำหนดกลศาสตร์และการเคลื่อนที่ได้ เช่น Joint limits ของข้อต่อที่กำลังหมุนและความเร็วการหมุน ซึ่งข้อต่อมีหลายแบบที่สามารถกำหนดได้ เช่น ข้อต่อแบบหมุน, ข้อต่อแบบเลื่อน, ข้อต่อแบบบิดติด, ข้อต่อแบบต่อเนื่อง

2.3.3 Robot Operating System

Robot Operating System หรือ ROS ถูกพัฒนาโดยบริษัท Willow Garage, แต่เดิมแล้วเป็นเครื่องมือพัฒนาเพื่อใช้งานกับหุ่นยนต์ PR2 ในปี 2007 ซึ่งพัฒนาเป็น open source framework สำหรับนักพัฒนาซอฟแวร์ที่เกี่ยวข้องกับหุ่นยนต์ มีความสามารถในการทำงานแบบ parallel บนคอมพิวเตอร์หลายเครื่องได้ สามารถทำงานได้หลาย OS แต่ที่ชั้นพื้นฐานคือ Ubuntu และ Debian นอกจากนี้ยังมีคลังที่คุณสามารถดาวน์โหลดและใช้เป็น libraries อีกด้วย การใช้ ROS จะช่วยทำให้เราสามารถพัฒนาหุ่นยนต์ได้อย่างรวดเร็วมากขึ้น ประหยัดเวลา ประหยัดทรัพยากรในส่วนนี้จะกล่าวถึง ROS คร่าวๆ

Nodes

Node เป็นเม็ดอนหน่วยประมวลผลในระบบ ROS, Node สามารถที่จะส่งข้อมูลหาโนนด์อื่นๆได้ ผ่าน Topics หรือ Services ในทางปฏิบัติแล้วโนนด์เป็นตัวประมวลผลอยู่ๆ ที่คุณทำหน้าที่เฉพาะ ยกตัวอย่างเช่น โนนด์ตัวแรกเชื่อมต่อกับกล้อง เพื่อที่จะนำภาพจากกล้องออกมานะ โนนด์ตัวที่สองใช้ในการหาลูกบล็อกที่อยู่ในภาพที่ได้มาจากโนนด์ตัวแรก และโนนด์ตัวที่สามใช้ในการคำนวนหาตำแหน่งของลูกบล็อกที่อยู่บนโลกจริงๆ จากตำแหน่งของลูกบล็อกที่ได้มาจากโนนด์ที่สอง ดังนั้นจะเห็นว่าแต่ละโนนด์จะทำงานเฉพาะของตัวเอง ซึ่งสามารถนำมาร่วมกันได้ การเขียนเป็นแบบโนนด์จะช่วยทำให้เราสามารถที่จะนำโปรแกรมกลับมา แก้ไขปรับปรุงให้ใช้ใหม่ได้ง่าย ในกรณีที่จะนำไปทำงานอย่างอื่น ยกตัวอย่างเช่น โนนด์ที่เอาภาพจากกล้องออกมานะ อาจจะมีโนนด์อีกตัว ทำหน้าที่ในการหาโกล์ดเป้าหมาย และหากทิศทางการเคลื่อนที่ของหุ่นยนต์ได้ ดังนั้นการพัฒนาโนนด์เป็นส่วนย่อยๆเล็กๆ ก็เพื่อที่จะทำให้การแก้ไขหรือปรับปรุงได้ง่าย



รูปที่ 2.32: ตัวอย่างสถาปัตยกรรมของ ROS

จากตัวอย่างสถาปัตยกรรมของ ROS ดังรูปที่ 2.32 นั้นสามารถอธิบายได้ว่า หุ่นยนต์เคลื่อนที่ด้วยล้อมีภารกิจคือ เคลื่อนที่ตามเส้นไปเรื่อยๆจนกว่าจะเจอเครื่องหมายหยุด Node คือตัวที่แสดงด้วยรูปวงรี ข้างในเป็นชื่อ Hend ส่วน Topic จะแสดงด้วยรูปสี่เหลี่ยม ซึ่งข้างในเป็นชื่อของ Topic และชนิดของ Message ที่ใช้ในการส่งข้อมูล มาดูกันก่อนอื่น ภาพถูกส่งมาจากกล้อง และก็มี Hend สองตัวในการดูเส้น และเครื่องหมายหยุด จากการที่ได้มานะ เมื่อ Hend ได้ข้อมูลแล้วก็คำนวณผลการเดินของหุ่นยนต์โดยส่งไปยัง node navigation และ Hend นี้ ก็จะทำหน้าที่คำนวณความเร็วและทิศทางของหุ่นยนต์ ส่งไปยัง node robot_control ซึ่งเป็นตัวสั่งการมอเตอร์ของหุ่นยนต์อีกด้วย

ตารางที่ 2.2: ตัวอย่างชื่อและข้อมูลของ Message

Twist.msg		Stop.msg	
geometry_msgs/Vector3	linear	uint8	RED = 0
geometry_msgs/Vector3	angular	uint8	GREEN = 1
(ก) Message Twist		uint8	color
		float32	distance
		(ข) Message Stop	

ตัวอย่างของ Message สองอันนี้ Twist message (รูปที่ 2.2ก) คือ message ที่เอาไว้บอกรความเร็วเชิงเส้น และความเร็วเชิงมุม ซึ่ง ROS มี message ชนิดนี้ให้อยู่แล้ว ส่วน Stop message (รูปที่ 2.2ข) คือ message ที่เอาไว้บอกระยะทางและสีของป้าย Stop ซึ่ง message นี้ถูกสร้างขึ้นมาใหม่เพื่อใช้กับงานนี้โดยเฉพาะ

Topics and Messages

Messages เป็นตัวหลักสำคัญในการติดต่อสื่อสารกันระหว่าง Hend ใน ROS โดยที่ message จะถูกส่งผ่านไปยัง topic เสมอ แต่ละ Node สามารถที่จะ subscribe หรือ publish ไปยัง topic นี้ได้ การเชื่อมต่อ กันระหว่าง Node นั้นสามารถส่งอยู่ภายในเครื่องคอมพิวเตอร์เครื่องเดียวกัน หรือเครื่องอื่นได้ที่อยู่ใน network เดียวกัน โดยจะติดต่อสื่อสารโดยใช้ TCP/IP การใช้คอมพิวเตอร์หลายเครื่องก็จะช่วยให้การประมวลผลมีประสิทธิภาพมากยิ่งขึ้น นอกจากนั้นยังสามารถที่จะแบ่งหน้าที่การทำงานออกจากกันได้ เราสามารถที่จะสร้าง Topic หรือ Message ขึ้นมาเองได้ หากต้องการใช้งานที่เฉพาะทาง

roscore

roscore เป็นส่วนกลางในการรันระบบทั้งหมด เราจะเรียกว่า rosmaster ซึ่งมีหน้าที่ในการจัดการ topics ทั้งหมด ที่ต้องการจะเชื่อมต่อกันไม่ว่าจะเป็นการ publish หรือ subscribe แต่ rosmaster จะเป็นแค่ตัวจัดการเท่านั้นไม่ได้เป็นตัวที่เก็บ message ต่างๆที่ส่งไปมา ดังนั้น rosmaster จะไม่ทำให้เกิดคอกขวด เวลา rans ระบบ ในกระบวนการก็คือ subscribe node จะถาม rosmaster ว่ามี topic ที่ต้องการรับข้อมูลใหม่ ส่วนตัว master ที่เก็บค่า topic message เอาไว้ ก็จะส่งไปยัง subscribe node ถ้าหากมีข้อมูลตามที่ร้องขอมา และ rosmaster ก็จะจำไว้ว่ามี node ไหนเชื่อมต่อกับ node ไหนบ้าง

rosparameter server เป็นตัวในการเก็บค่าต่างๆที่เป็น global key-value ซึ่งช่วยให้ node ทุกตัวสามารถใช้ข้อมูลตัวเดียวกันได้ สามารถปรับเปลี่ยนระหว่างการทำงานอยู่ได้ โดยใช้ rqt plugin ซึ่งจะกล่าวในส่วนต่อไป

roslog เป็นตัวที่ใช้สำหรับ logging ข้อมูลต่างๆ ซึ่งจะถูก publish ออกมายัง topic /rosout ซึ่งเราสามารถที่จะเขียนโปรแกรม subscribe จากตัว topic นี้ไปเก็บเป็นไฟล์ได้

Services

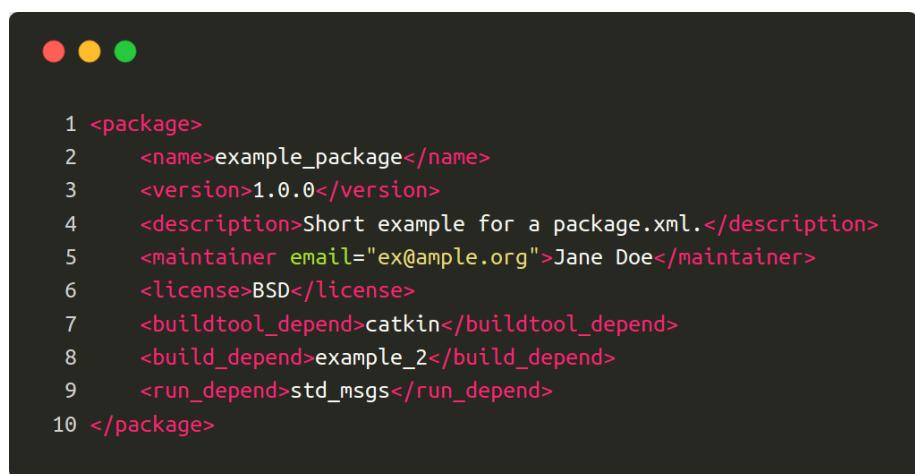
Services หรืออีกชื่อนึงคือ remote procedure calls (RPC) ความหมายคือเป็นการส่ง messages แบบที่ไม่ได้เจาะจงว่าจะส่งไปที่ไหน เมื่อเราเรียก service และระบบจะรู้ว่าจะมีการตอบกลับ เราจะเรียกระบวนการนี้ว่า request และ response message Node ที่อยู่ทำงานเมื่อมีการเรียกใช้ service จะเรียกว่า service server และ node ที่เรียก service จะเรียกว่า service client การใช้งาน service เหมาะสำหรับงานที่ต้องการความรวดเร็ว (fast task) แต่ไม่ควรใช้กับระบบที่ต้องใช้เวลานาน เพราะระบบจะหยุดไม่ยอมทำต่อ ต้องรอให้ service ทำงานเสร็จก่อน สำหรับงานที่ต้องใช้เวลาในการคำนวณนานจะใช้ action แทน จะกล่าวในส่วนถัดไป

Actions

Actions จะใช้กับการทำงาน การประมวลผลที่ต้องใช้เวลาในการทำงาน หรือที่เรียกว่า asynchronously task ในแต่ละ action จะมี message อよู่ 3 ชนิด คือ goal, feedback และ result Node ที่เป็นตัวรับและรอให้ node อื่นมาเรียก จะเรียกว่า action server ส่วน node ที่เรียกการทำงาน action จะเรียกว่า action client การใช้งาน action client จะส่ง message goal ไปยัง action server และ action server จะพยายามทำงานตาม goal ที่ได้รับมา ในระหว่างที่ action client ก็จะทำงานของตัวเองต่อไป แต่จะได้รับ feedback จาก action server อよู่ตลอดเวลา และเมื่อถึง goal ที่กำหนดแล้ว server จะแจ้งมาทาง result message

Code Organization

ส่วนที่เล็กที่สุดของการจัดการซอฟแวร์ใน ROS ก็คือ package ภายใน package จะมีไฟล์ที่ชื่อว่า package.xml ซึ่งไฟล์นี้จะทำหน้าที่ในการ อธิบายและบอกข้อมูลต่างๆที่เกี่ยวกับ package นี้ ยกตัวอย่างเช่น ชื่อของ package, ชื่อของผู้เขียน, ลิขสิทธิ์ และ dependencies ที่ต้องใช้กับ package นี้ นอกจากนี้ยังสามารถใส่ข้อมูลอื่นๆเกี่ยวกับ node ลงไปเพิ่มเติมได้



```

1 <package>
2   <name>example_package</name>
3   <version>1.0.0</version>
4   <description>Short example for a package.xml.</description>
5   <maintainer email="ex@example.org">Jane Doe</maintainer>
6   <license>BSD</license>
7   <buildtool_depend>catkin</buildtool_depend>
8   <build_depend>example_2</build_depend>
9   <run_depend>std_msgs</run_depend>
10 </package>

```

รูปที่ 2.33: ตัวอย่างไฟล์ package.xml

แต่ละ tag ใช้ในการบอกข้อมูลของ package นี้ คราวเป็นเจ้าของ ใครเป็นคนเขียน รวมไปถึง dependencies ที่จำเป็นต้องใช้ของ package นี้ด้วย ดังรูปที่ 2.33

Code Distribution

การที่จะนำ Nodes กลับมาใช้ใหม่หรือเอาอกมาแชร์ได้นั้น จะต้องมีการทำเอกสารของ Packages นั้นๆ ด้วย โดยปกติแล้วจะถูกนำไปเก็บไว้ที่ GitHub และ package dependencies จะบอกไว้ในไฟล์ package.xml เรียบร้อยแล้ว เพื่อให้ง่ายต่อการนำไปติดตั้ง หากผู้ที่นำไปใช้พัฒนาต่อหรือแก้ไขข้อผิดพลาดก็สามารถที่จะช่วยกันได้ โดยการ Pull request หรือ Report issues ได้

ROS Packages ที่ใช้ในงานวิจัย

Package คือพื้นฐานของ ROS, แอพพลิเคชันทั้งหมดใน ROS จะพัฒนาโดยมี package เป็นรากฐาน ใน package นั้นจะเก็บพวกไฟล์ configuration ไปจนถึงไฟล์ launch ที่สามารถปรับ package หรือ node อื่นๆ ได้ ตอนนี้ ROS มี packages มากกว่า 5000 packages แล้ว

Metapackage เป็นการรวมกันของ packages ที่ทำหน้าที่คล้ายๆ กันหลายๆ ตัวมารวมไว้ที่เดียวเพื่อจะได้ใช้งานง่าย ตัวอย่าง Navigation metapackage ประกอบไปด้วย 10 packages เช่น [AMCL(partical filter), DWA, EKF(extended kalman filter) และ map_server] ซึ่งหากติดตั้ง metapackage ตัวนี้ก็จะได้มาหมดเลย ในส่วนนี้จะอธิบายคร่าวๆ ถึง ROS standard packages ที่จะนำมาใช้ในงานวิจัยครั้งนี้

rosbag rosbag เป็นแพกเกจที่สามารถบันทึก message ที่ส่งหากันในระหว่างที่ ROS กำลังทำงานได้ไฟล์ที่บันทึกจะเรียกว่า rosbag ประโยชน์ของมันคือเราสามารถเอาเข้ามาใช้ในการตรวจสอบ หรือนำมาเล่นซ้ำได้อีกทั้งยังง่ายต่อการค้นหาข้อผิดพลาดอีกด้วย

tf2 tf2 เป็นแพกเกจที่สามารถติดตามการเปลี่ยนแปลงของ Coordinate frame เราสามารถใช้ในการทำความสัมพันธ์ระหว่าง frame ได้ ยกตัวอย่างเช่นหากเราต้องการหาตำแหน่งของ foot เทียบกับ pelvis ก็สามารถใช้ tf2 หาได้

robot_state_publisher robot_state_publisher แพกเกจที่ subscribe JointState message เพื่อที่จะนำตำแหน่งของของข้อต่อ และแบลนให้อยู่ในรูปข้อมูลของ tf2, tf2 สามารถเรียกจาก Node ใดๆได้เพื่อที่จะหา Coordinate frame ที่ต้องการได้

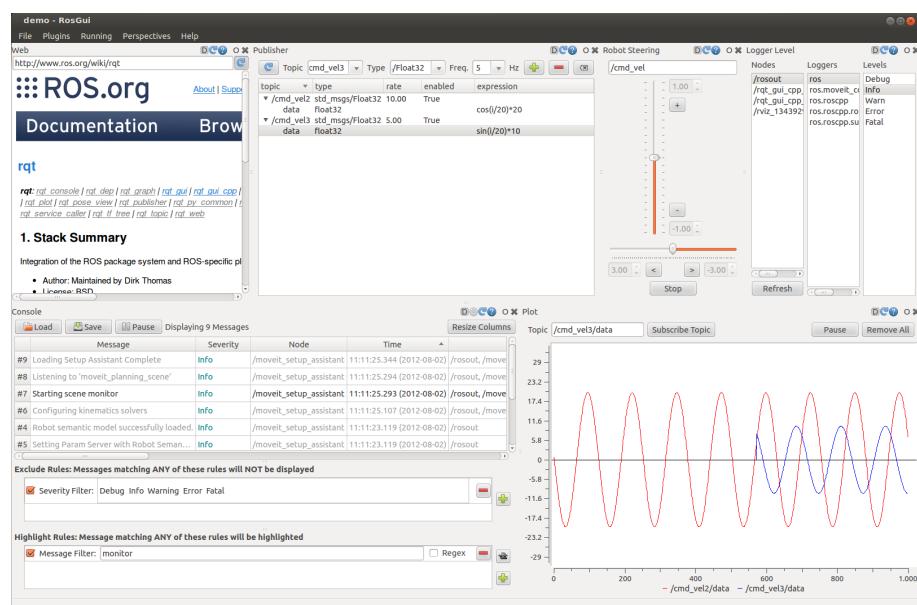
URDF Unified Robot Description Format (URDF) เป็นไฟล์ XML ที่เอาไว้อธิบายลักษณะของหุ่นยนต์ ใน ROS มีแพกเกจที่ใช้สำหรับการอ่านไฟล์ คือ urdf_parser แต่ไฟล์นี้มีการใช้งานโดย tf2 เช่นกัน

xacro xacro เป็นไฟล์ XML เช่นเดียวกับ URDF โดยไฟล์ xacro นี้มีประโยชน์มากในการใช้งานใน ROS เพราะว่าทำให้การเขียนไฟล์ URDF ง่ายขึ้น เพราะสามารถทำเป็นมาโครได้ สามารถปรับแต่งค่าตัวแปรต่างๆได้ง่ายขึ้น

Visualization

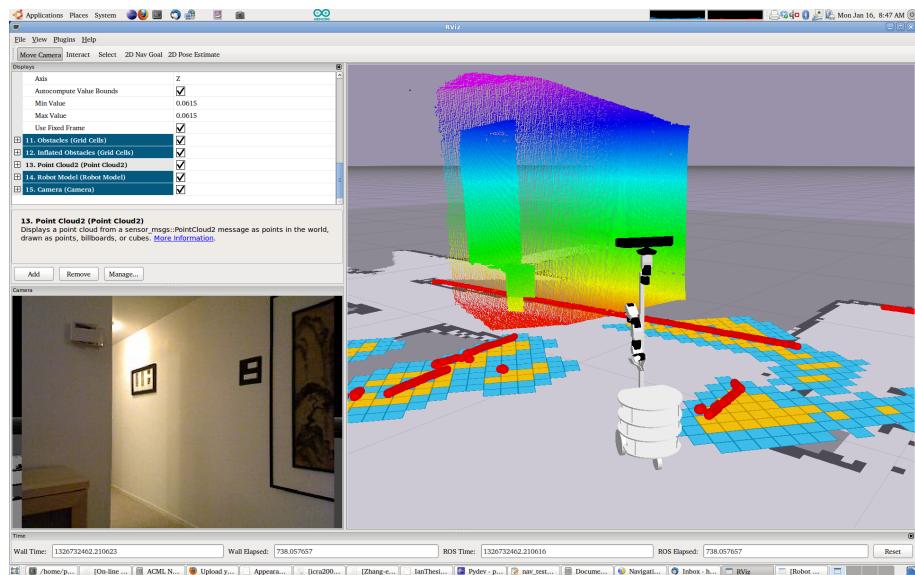
จุดแข็งสำคัญของ ROS อีกอย่างก็คือ มีเครื่องมือที่ช่วยในการแสดงผล Visualization ได้ นอกจากนี้ จากระบบ publisher-subscriber การใช้ Visualization tools นี้จะช่วยให้การทำงานง่ายขึ้น และประหยัดเวลามากขึ้น ในการนำข้อมูลต่างๆ จากหุ่นยนต์ออกมาระบบการแสดงผล เพราะว่า Visualization tools นี้สามารถที่จะ subscribe จาก topic ที่มีการใช้งานอยู่แล้วมาแสดงผลได้ทันที ใน ROS มีเครื่องมือสำคัญอยู่ 2 ตัวที่ใช้สำหรับการ Visualization ซึ่งสามารถที่จะปรับแต่งให้กล้ายเป็นเวอร์ชันของเราเองได้

rqt rqt เป็น UI ที่มีฐานมาจาก QT ซึ่งมาพร้อมกับการเชื่อมต่อ ROS เป็นส่วนเสริมในรูปแบบของ QWidget เราสามารถที่จะแสดงผลหลายๆ widgets ได้ภายในเวลาเดียวกัน สามารถที่จะย่อขยาย เปลี่ยนตำแหน่ง ลากวางได้ การเชื่อมต่อ กับ ROS นั้นสามารถนำการแสดงผลแบบ 2D ไปแสดงได้ดังรูปที่ 2.34 เป็นการแสดงภาพของกราฟที่ได้รับข้อมูลมาจาก topic หลายๆ ตัว และสามารถที่จะปรับแต่งค่าและ publish ออกໄປได้ ด้วยการเขียนโปรแกรมเข้าไป ซึ่งจะเป็นประโยชน์อย่างมากเวลาที่ใช้ในการปรับจูนพารามิเตอร์ต่างๆ เพราะว่าเราสามารถที่จะเปลี่ยนค่าได้ทันที ไม่ต้องรันโปรแกรมใหม่ ในรูปที่ 2.34 เป็นการนำ rqt มาเขียนเป็น GUI ให้ผู้ใช้สามารถใช้งานได้ง่ายและสามารถที่จะปรับแต่งพารามิเตอร์ต่างๆ ได้เรียลไทม์



รูปที่ 2.34: ตัวอย่างการแสดงผลใน rqt

RViz RViz เป็น 3D visualization ของสถานะต่างๆ ของหุ่นยนต์และสภาพแวดล้อม โดยใช้ไฟล์ URDF เป็นมาตรฐานการแสดงถึงหุ่นยนต์ ซึ่งสามารถที่จะแสดงตำแหน่งปัจจุบันของข้อต่อต่างๆ ในหุ่นยนต์ได้ สามารถที่จะแสดงค่าเซนเซอร์เป็น marker ได้ การใช้งานจะเป็นเหมือนการบอกริกัดเฟรม ลักษณะการแสดงผลใน RViz มีหลากหลายรูปแบบไม่ว่าจะเป็น camera images, depth clouds, laser scans หรือ point clouds อย่างไร ก็ตามการแสดงผลใน Rviz นั้นจะไม่ได้คำนึงถึงแรงที่เข้ามากระทำกับตัวของหุ่นยนต์ แต่ถ้าเป็นการเคลื่อนที่ ที่มีพิกัดเฟรมแล้วสามารถนำมาแสดงได้ ดังรูปที่ 2.35 เป็นเคสของหุ่นยนต์เคลื่อนที่ด้วยล้อ และทำแผนที่ด้วยข้อมูลความลึกที่ได้มาจากการ Kinect

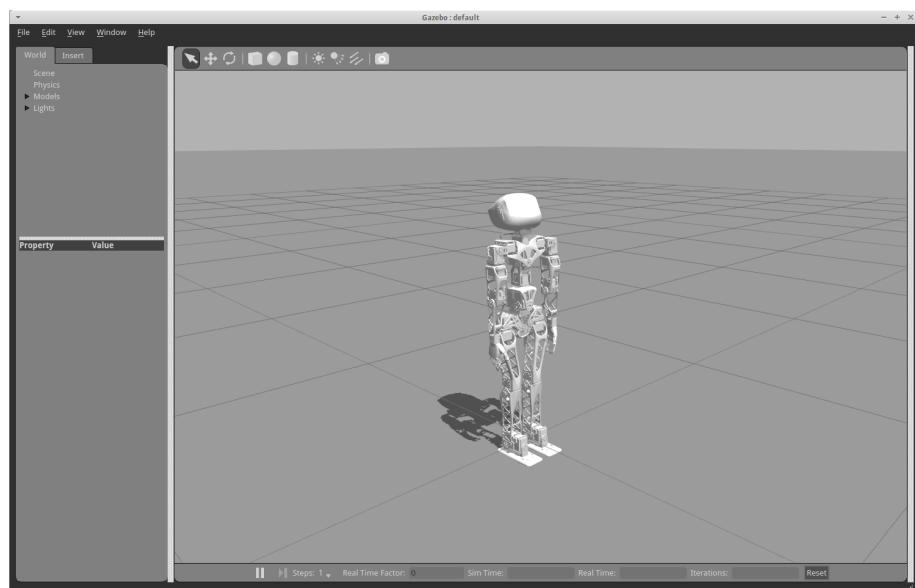


รูปที่ 2.35: ตัวอย่างการแสดงผลใน RViz

Simulation

Simulation เป็นส่วนที่สำคัญมากสำหรับการพัฒนาโปรแกรมของหุ่นยนต์ เพราะว่าความสามารถที่จะสร้างโปรแกรมและทดสอบได้โดยไม่จำเป็นต้องมี hardware ซึ่งในส่วนนี้จะช่วยลดความเสียหายจากบักษหรือโปรแกรมผิดพลาด ที่อาจจะเกิดขึ้นกับหุ่นยนต์ของเรา การจำลองจะช่วยลดเวลาในการพัฒนาลงได้ ระบบจำลองปัจจุบันมีมากมายหลายตัวแต่ ตัวที่ได้รับคำแนะนำมากที่สุดคือ Gazebo เพราะว่า gazebo สามารถที่จะเชื่อมต่อกับ ROS ได้โดยตรง และนักพัฒนาส่วนใหญ่ใช้ gazebo

การจะใช้ gazebo ได้นั้นเราจะต้องใช้ไฟล์ URDF ซึ่งเป็นไฟล์ที่เอาไว้แสดงหุ่นยนต์ในระบบจำลอง และสามารถที่จะคำนวณหา collision ให้เราได้อีกด้วย



รูปที่ 2.36: ตัวอย่างหุ่นยนต์อิวามานอยด์ Poppy

2.4 การออกแบบระบบพื้นฐาน

2.4.1 ข้อแตกต่างระหว่าง Open platform กับ Non-open platform

ทุนยนต์ Open platform คือ การออกแบบระบบพื้นฐานของทุนยนต์ที่เปิดให้ผู้ที่ต้องการศึกษาหรือผู้ใช้ทั่วไปสามารถเข้าถึงข้อมูลต่างๆที่เกี่ยวข้องกับทุนยนต์นั้นๆได้ ผู้ใช้สามารถที่จะนำข้อมูลเหล่านั้นมาแก้ไข ปรับปรุง แต่งเติม หรือเรียนรู้และพัฒนาตามได้ด้วยตนเอง ซึ่งข้อมูลที่กล่าวมานี้สามารถหาได้จากเว็บไซต์ของผู้พัฒนาทุนยนต์ ปัจจุบันมีทุนยนต์อิมานอยด์ที่เป็นเปิดให้เข้าถึงหลายรูปแบบแตกต่างกันไป

ส่วนทุนยนต์ Non-open source platform คือ ทุนยนต์ที่สร้างมาเฉพาะเจาะจงสำหรับการวิจัย การสำรวจ หรือการแข่งขันโดยเฉพาะ ไม่เปิดให้บุคคลภายนอกเข้าศึกษาหรือแก้ไขปรับปรุง ซึ่งทำให้ทุนยนต์ประเภทนี้ไม่เหมาะสมสำหรับผู้วิจัยที่จะเรียนรู้และศึกษาด้วยตนเอง เพราะมีขีดจำกัดใหญ่ ใช้ทรัพยากร่มาก และการออกแบบมีความซับซ้อน เรียนรู้ยากกว่าทุนยนต์แบบ Open platform

2.4.2 มาตรฐานหน่วยวัดและการบวกพิกัด

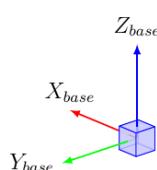
การใช้หน่วยวัดที่ไม่ตรงกันอาจจะทำให้เกิดปัญหาขึ้นได้ เมื่อจากเป็นแพลตฟอร์มนั้นจะมีบุคคลอื่นช่วยกันพัฒนาหลายคน จึงควรที่จะมีมาตรฐานในการวัดและการกำหนดพิกัดต่างๆที่ตรงกันเพื่อให้เกิดความชัดเจนในการทำความเข้าใจ

หน่วยวัด การวัดนั้นใช้มาตรฐานการวัดเป็น SI Units ซึ่งมาตรฐานนี้ใช้กันอย่างแพร่หลายและเป็นสากล โดยหน่วยการวัดนี้ได้รับการยืนยันจาก Bureau International des Poids et Mesures

ตารางที่ 2.3: ตารางแสดงหน่วยวัดมาตรฐาน

ปริมาณ(Quantity)	หน่วยวัด(Unit)	สัญลักษณ์(Symbol)
ความยาว(Length)	เมตร(metre)	<i>m</i>
มวล(Mass)	กิโลกรัม(kilogram)	<i>kg</i>
เวลา(Time)	วินาที(second)	<i>s</i>
กระแสไฟฟ้า(Electric Current)	แอม培ร์(ampere)	<i>A</i>
มุม(Angle)	เรเดียน(radian), องศา(degree)	<i>rad, deg</i>
ความถี่(Frequency)	เฮิร์ตซ์(Hertz)	<i>Hz</i>
แรง(Force)	นิวตัน(Newton)	<i>N</i>
กำลัง(Power)	วัตต์(Watt)	<i>P</i>
แรงดันไฟฟ้า(Voltage)	โวลต์(Volt)	<i>V</i>
อุณหภูมิ(Temperature)	เซลเซียส(Celsius)	<i>C</i>

พิกัดเฟรม การบวกทิศทางการหมุนนั้นใช้หลักตามกฎมือขวา โดยการตั้งแกนนั้นหากเทียบกับมือแล้ว X ไปข้างหน้า Y ไปทางซ้าย Z พุ่งขึ้น ตามภาพที่ 2.37



รูปที่ 2.37: การตั้งแกนตามกฎมือขวา

บทที่ 3

การดำเนินงานวิจัย

3.1 แผนการดำเนินงาน

การออกแบบหุ่นยนต์ humanoid อุทัย (UTHAI) มาจากภาษาอังกฤษคำว่า Universal Template for Humanoid Algorithm Interface เพื่อให้สมกับเป็นหุ่นยนต์ humanoid ที่ใช้สำหรับงานวิจัยและพัฒนาต่อไป



การดำเนินงานและการออกแบบการสร้างหุ่นยนต์ humanoid อุทัย มีแผนการทำงานซึ่งแบ่งออกเป็น 4 ส่วนดังนี้ ส่วนแรกคือ ส่วนของฮาร์ดแวร์ที่เกี่ยวกับโครงสร้างทางกลของหุ่นยนต์ humanoid เช่น ข้อต่อ ก้านต่อ

ฝ่าเท้า รวมไปถึงระบบอิเล็กทรอนิกส์ ตัวประมวลผลการควบคุม เช่นเซอร์ ตัวขับเคลื่อนต่าง ๆ และส่วนที่สองคือ ส่วนของซอฟท์แวร์ที่เกี่ยวข้องกับการติดต่อสื่อสารกันเบื้องต้น การควบคุมตัวขับเคลื่อนที่ข้อต่อ การอ่านค่า เช่นเซอร์ และระบบพื้นฐานสำหรับการใช้งาน ขั้นตอนในการทำงานวิจัยครั้งนี้

3.2 การออกแบบโครงสร้างของหุ่นยนต์

3.2.1 การเชื่อมต่อหุ่นยนต์อิเล็กทรอนิกส์

โครงสร้างแพลตฟอร์มหุ่นยนต์อุทัยจะประกอบไปด้วยสองขา เพื่อทำให้เกิดองศาอิสระเป็น 12 องศาอิสระ (DOFs) ใช้ Dynamixel servos 12 ตัว มอเตอร์ Dynamixel ทั้งหมดเชื่อมต่อกันแบบ daisy chain ข้างนึงของมอเตอร์ตัวแรกเชื่อมต่อกับแบบเดอเรรี่ 12V และอีกข้างต่อ กับ USB2Dynamixel ทั้งหมดนี้เป็นการเชื่อมต่อ Odroid เข้ากับหุ่นยนต์ ดังรูปที่ 3.1



รูปที่ 3.1: การเชื่อมต่อระหว่าง Odroid กับ Dynamixel servos

หุ่นยนต์อิเล็กทรอนิกส์อุทัยใช้เซอร์วิsmoเตอร์ 12 ตัว ทำให้เกิดเป็น 12 องศาอิสระ USB2Dynamixel ใช้เพื่อที่จะสั่งการเซอร์วิsmoเตอร์ Dynamixel ผ่าน Odroid ตำแหน่งของเซอร์วิsmoเตอร์ Dynamixel EX-106+ นั้นมา จากเอนโคดเดอร์ที่อยู่ภายในเซนเซอร์ Gyro/Accelerometer ติดอยู่กับตัวของหุ่นยนต์ เพื่อช่วยในการทรงตัวของหุ่นยนต์ เซนเซอร์ Accelerometer จะอัพเดตค่าของตัวเองเรื่อยๆ ฟังก์ชันส่วนเสริมจะมาจาก ROS และ Odroid เชื่อมต่อกับคอมพิวเตอร์ภายนอกผ่าน Wi-Fi

3.2.2 อุปกรณ์ที่ใช้ในหุ่นยนต์อิเล็กทรอนิกส์อุทัย

Odroid

Dynamixel servo EX-106+

Dynamixel EX-106+ เป็นตัวขับเคลื่อนที่นิยมใช้ในปัจจุบัน โดยความสามารถของมันคือ สามารถที่จะอ่านค่าความเร็ว แรงดันไฟฟ้า กระแสไฟฟ้า อุณหภูมิ ตำแหน่ง และแรงบิด มอเตอร์แต่ละตัวจะมีบอร์ดควบคุมของตัวเอง

USB2Dynamixel connector

ตัวนี้เป็นอุปกรณ์สำหรับเชื่อมต่อ Odroid กับ Dynamixel โดยจะเชื่อมต่อผ่านพอร์ต USB ของ Odroid ไปยัง Dynamixel motor ผ่านสายทั้งหมด 4 เส้น เป็นการเชื่อมต่อแบบ RS-485

Accelerometer

Accelerometer ที่ใช้เป็น MPU-9250 Accelerometer+Gyro+Magneto เพื่อเอาไว้ใช้หามุมเอียงของหุ่นยนต์ เทียบกับโลก

Ground contact sensor

Wi-Fi Adapter

3.3 การออกแบบโปรแกรมด้วย ROS

3.3.1 Modelling

หลังจากที่เราได้ออกแบบและโมเดลหุ่นยนต์ของเราขึ้นมาที่ใช้ CAD tools ต่างๆ เช่น AutoCAD, SolidWorks, Blender หรืออื่นๆ ก็เพื่อที่จะนำมามาใช้ในการทำ Simulation การที่เราทำ Simulation นั้นก็จะสามารถองเห็นหุ่นยนต์ และเห็นการทำงานของหุ่นยนต์เรา ก่อนที่เราจะสร้างมันขึ้นมาจริงๆ หุ่นยนต์จำลองที่เราสร้างขึ้นมาันนี้ควรที่จะมีลักษณะให้ใกล้เคียงกับของจริงมากที่สุด ไม่ว่าจะเป็นรูปร่าง รูปทรง น้ำหนักต่างๆ

3.3.1.1 ROS packages for robot modelling

ROS นั้นได้ให้เครื่องมือที่ช่วยให้เราสามารถสร้าง 3D robot models ได้ ใน ROS มี meta package ที่ชื่อว่า `robot_model` ซึ่งข้างในมี package ต่างๆ ที่ใช้สำหรับสร้าง 3D robot models

`urdf` เป็น 1 ในหลายๆ package ที่อยู่ใน `robot_model`, `urdf` เป็น xml ไฟล์ที่เอาไว้ใช้บอกรากษณะของหุ่นยนต์ ย่อมาจาก Unified Robot Description Format(URDF) เราสามารถระบุ robot model, sensors และ working environment โดยใช้ URDF การบอกนั้นจะสามารถบอกเป็นเหมือน tree structure ของ link ต่างๆ ในตัวหุ่นยนต์ สามารถบอก rigid link เชื่อมต่อกันผ่าน joints แต่ถ้าเป็น flexible link จะไม่สามารถบอกได้โดยใช้ `urdf`

`joint_state_publisher` เครื่องมือนี้มีประโยชน์มากในการ model robot URDF เพราะมันสามารถหา joints ทุก joint ที่ไม่ใช่ fixed joints มาแสดงเป็น GUI sliders ทำให้เราสามารถเลื่อนๆ หมุนๆ ไปได้ อีกทั้งยังสามารถใช้งานร่วมกับ visualize RViz

`robot_state_publisher` เป็นเครื่องมือที่ใช้ในการ publish 3d pose ของ link ต่างๆ ใน `urdf` การ publish นั้นจะใช้ ROS tf(transform) ROStf คือการหาความสัมพันธ์ระหว่าง frame ของหุ่นยนต์

`xacro` ย่อมาจาก XML Macros หรือเราสามารถเรียกอีกอย่างว่า URDF plus add-ons. ซึ่งการทำงานเหมือนกับ `urdf` แต่ทำให้ไฟล์ `urdf` สั้นกว่า อ่านง่ายกว่า และสามารถใช้เพื่อทำให้สร้างหุ่นยนต์ที่มีความซับซ้อนง่ายขึ้น เราสามารถแปลงไฟล์ `xacro` เป็น `urdf` ได้

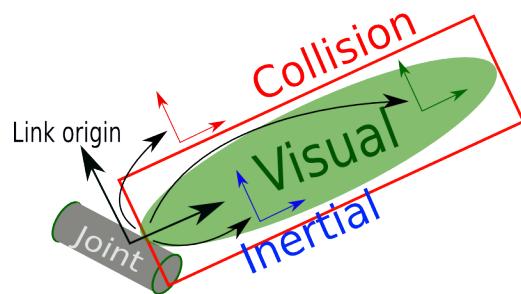
3.3.1.2 URDF

ในส่วนนี้จะเป็นการอธิบายระบบทางกลของหุ่นยนต์ขีวามนอยด์เป็นไฟล์ที่ใช้ร่วมกับ ROS ได้ เพื่อที่จะสามารถนำไปใช้กับ Simulation ในอนาคตได้ ในการอธิบายระบบทางกลนั้นผู้วิจัยได้ใช้ไฟล์ URDF (Universal Robotics Description Format) ซึ่งใช้ภาษาการเขียนเป็น XML ในการบอกส่วนประกอบแต่ละส่วนของหุ่นยนต์

Link

ในไฟล์ URDF แต่ละชิ้นส่วนของหุ่นยนต์เราจะเรียกว่า link และใน link จะประกอบไปด้วยส่วนย่อยๆ 3 ส่วนคือ `<inertia>` ที่เอาไว้บอกรถึงค่าตัวแปรทางฟิสิกส์, `<visual>` ที่เอาไว้แสดงผลให้เราเห็น, `<collision>` ที่เอาไว้ตรวจสอบว่าหุ่นยนต์มีการชนกันกับสิ่งแวดล้อมใหม ดังตัวอย่างด้านล่างนี้

ยังมี `tags` อีกหลายตัวที่ใช้ในการอธิบายแต่ละชิ้นส่วนของหุ่นยนต์ แต่ตัวอย่างเป็นเพียงแค่ส่วนหนึ่งเท่านั้น ในความเป็นจริงแล้วเราจะจะเขียน `tags` ต่างๆ ก็ตามที่เราต้องการ โดยใน URDF ไฟล์นั้นจะเอาไว้เก็บข้อมูลลักษณะเฉพาะของหุ่นยนต์เอาไว้ และยังสามารถใช้กับซอฟแวร์ตัวอื่นๆ อีกด้วย

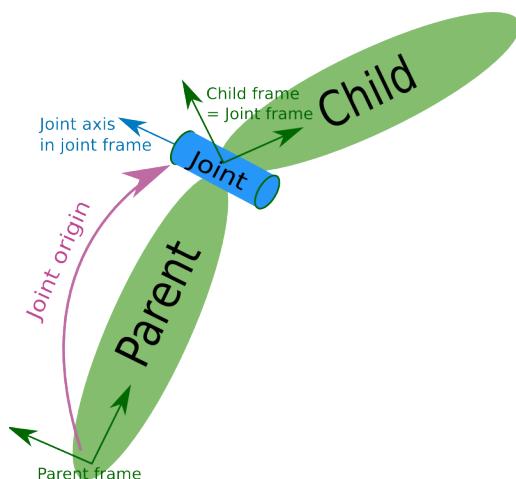


รูปที่ 3.2: การอธิบาย link ใน URDF ไฟล์

Joint

อีกส่วนที่สำคัญสำหรับการสร้างไฟล์หุ่นยนต์ด้วย URDF ก็คือ Joint tag โดย tag นี้จะอธิบายถึงความสัมพันธ์ระหว่างก้านต่อสองอัน ส่วนนี้ไม่ได้มีเพียงแค่ทำข้อต่อให้เป็นแบบหมุนได้อย่างเดียว ยังมี Fix, Revolution, Linear และ Planar นอกจากนี้ เรายังสามารถที่จะเพิ่มองค์กรสูงสุดต่ำสุดของข้อต่อ รวมไปถึง dynamic properties ต่างๆ ตามที่เห็นตัวอย่างด้านล่างนี้

เมื่อเรานำ Joint และ Link มารวมกันเราจะต้องพิจารณาว่ามีวางรูปแบบเป็นไปตามรูปที่ 3.3 โดยจะมีระยะระหว่างแกนของแต่ละข้อต่อ กับ ก้านต่อ ขึ้นส่วนแรกของการสร้างไฟล์ URDF จะมีชื่อว่า base_link และเฟรม origin จะเป็นเฟรมอ้างอิง เมื่อเราต่อ Joint เข้ากับ Link จะเรียกว่าก้านต่อที่นำมาติดว่า parent โดยเฟรม origin ของข้อต่อจะอยู่จุดเดียวกับเฟรม origin ของก้านต่อ ในสถานะเดียวกันก้านต่อที่นำมาต่อจากข้อต่อ เราจะเรียกว่า child และเฟรม origin ของก้านต่อ child จะอยู่ที่จุดเดียวกับเฟรม origin ของข้อต่อ



รูปที่ 3.3: การอธิบาย Joint ใน URDF ไฟล์

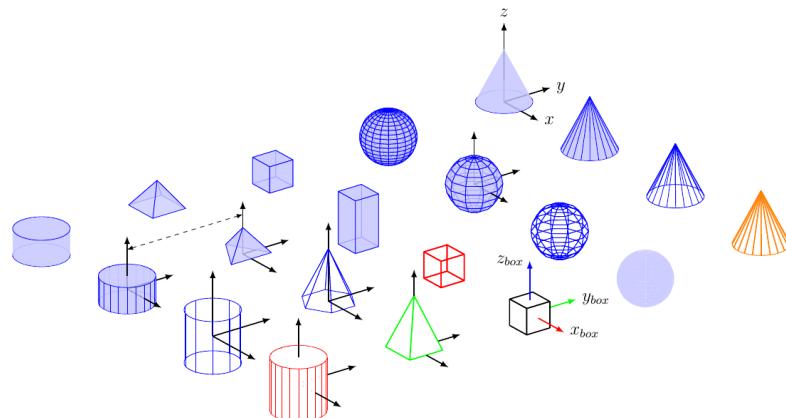
3.4 การออกแบบระบบพื้นฐาน

UTHAI-Tools

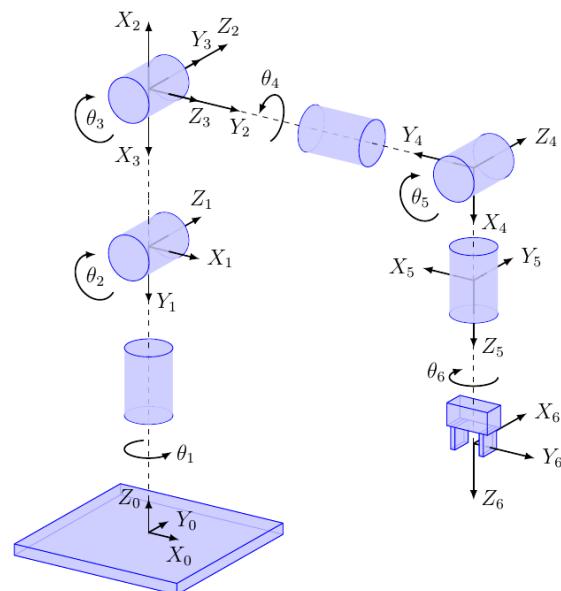
เครื่องมือสำหรับการทำงานในอิมานอยด์

sketch-lib

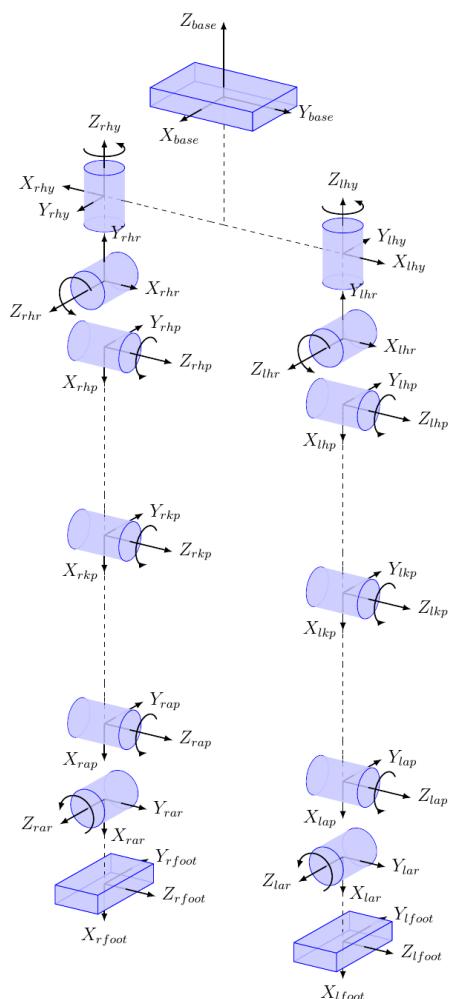
เป็นเครื่องมือที่ใช้สำหรับเอาไว้วาดรูปเฟรมของหุ่นยนต์



รูปที่ 3.4: ภาพตัวอย่างการวาดออฟเจ็คต่างๆ



รูปที่ 3.5: ภาพตัวอย่างการวาดเฟรมของแขนกล



รูปที่ 3.6: ภาพตัวอย่างการวัดเฟรมของหุ่นยนต์ humanoid

บทที่ 4

ผลการวิจัย

4.1 การออกแบบโครงสร้างของหุ่นยนต์

4.2 การออกแบบโปรแกรมด้วย ROS

4.2.1 Simulation Gazebo

ต้องติดตั้ง package ต่อไปนี้

1 joint_state_controller

2 effort_controller

3 controller_manager*

4 gazebo_ros_control*

4.3 การออกแบบระบบพื้นฐาน

บทที่ 5

สรุปผลการทดลองและข้อเสนอแนะ

5.1 การออกแบบโครงสร้างของหุ่นยนต์

5.2 การออกแบบโปรแกรมด้วย ROS

5.3 การออกแบบระบบพื้นฐาน

5.4 สรุปภาพรวม

ทำได้ดีนะจ๊ะ

ภาคผนวก

ภาคผนวก ก

ผลงานตีพิมพ์เผยแพร่จากวิทยานิพนธ์

ก.1 บทความวิจัยเสนอในที่ประชุมวิชาการและมีการพิมพ์รวมเล่ม

ก.2 บทความวิชาการ

ภาคผนวก ข
แหล่งข้อมูล Latex

ข.1 แหล่งข้อมูลออนไลน์

ประวัติผู้เขียน

ชื่อ สกุล	นายจิรภูร์ ศรีรัตนอากรรณ์
รหัสนักศึกษา	57340500067
วุฒิการศึกษา	วิศวกรรมศาสตรบัณฑิต(วิศวกรรมหุ่นยนต์และระบบอัตโนมัติ)
ชื่อสถาบัน	มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
ปีที่สำเร็จการศึกษา	2560
ทุนการศึกษา	กินกันตายเทศ กิตาฟนนนหกดาก

ประวัติผู้เขียน

ชื่อ สกุล	นายเจษฎากร ท่าไชยวงศ์
รหัสนักศึกษา	57340500067
วุฒิการศึกษา	วิศวกรรมศาสตรบัณฑิต(วิศวกรรมหุ่นยนต์และระบบอัตโนมัติ)
ชื่อสถาบัน	มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
ปีที่สำเร็จการศึกษา	2560
ทุนการศึกษา	กินกันตายเทศ กิตาฟนนนนหกดาก

ประวัติผู้เขียน

ชื่อ สกุล	นายุดมิภัทร โโซคอนันต์ทรัพย์
รหัสนักศึกษา	57340500067
วุฒิการศึกษา	วิศวกรรมศาสตรบัณฑิต(วิศวกรรมหุ่นยนต์และระบบอัตโนมัติ)
ชื่อสถาบัน	มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
ปีที่สำเร็จการศึกษา	2560
ทุนการศึกษา	กินกันตายເທດ กิตา芬ນนนທກດາ

