



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

การออกแบบโครงสร้างและพัฒนาระบบที่นิฐานสำหรับหุ่นยนต์ชีวภาพอยู่ด้วย<sup>ที่</sup>  
เพื่อการศึกษาและวิจัย

นายจิรภูริ ศรีรัตนอาภรณ์  
นายเจษฎากร ท่าไชยวงศ์  
นายวุฒิภัทร โชคอนันตทรัพย์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร  
ปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมหุ่นยนต์และระบบอัตโนมัติ  
สถาบันวิทยาการหุ่นยนต์ภาคสนาม  
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี  
ปีการศึกษา 2560





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

การออกแบบโครงสร้างและพัฒนาระบบที่นิฐานสำหรับหุ่นยนต์ชีวภาพอยู่ด้วย<sup>ที่นิฐานสำหรับหุ่นยนต์ชีวภาพอยู่ด้วย</sup>  
เพื่อการศึกษาและวิจัย

นายจิรภูริ ศรีรัตนอาภรณ์

นายเจษฎากร ท่าไชยวงศ์

นายวุฒิภัทร โชคอนันตทรัพย์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาชีวกรรมหุ่นยนต์และระบบอัตโนมัติ

สถาบันวิทยาการหุ่นยนต์ภาคสนาม

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

ปีการศึกษา 2560

**การออกแบบโครงสร้างและพัฒนาระบบพื้นฐานสำหรับหุ่นยนต์ชีวามาอยู่ด้วย  
เพื่อการศึกษาและวิจัย**

นายจิรภูริศ ศรีรัตนอาภรณ์

นายเจษฎากร ท่าไซวงศ์

นายวุฒิภัทร โชคอนันดร์พิรุณ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาชีวกรรมหุ่นยนต์และระบบอัตโนมัติ

สถาบันวิทยาการหุ่นยนต์ภาคสนาม

ปีการศึกษา 2560

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

คณะกรรมการสอบ

.....  
(นายธนัชชา ชูพจน์เจริญ)

.....  
ประธานกรรมการ  
(ดร.อาบทิพย์ รีวงศ์กิจ)

อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม

.....  
กรรมการ  
(ดร.ปิติวุฒิ ชีรากิตติกุล)

.....  
(รศ.ดร.ชิต เหล่าวัฒนา)

.....  
กรรมการ  
(ดร.สุภาษี วงศ์บุณย์ยงค์)

ชื่อวิทยานิพนธ์	การออกแบบโครงสร้างและพัฒนาระบบพื้นฐานสำหรับหุ่นยนต์อิเล็กทรอนิกส์เพื่อการศึกษาและวิจัย
หน่วยกิต	6
ผู้เขียน	นายจิรภูริษฐ์ ศรีรัตนอกรรณ์ นายเจษฎากร หาใจยวงศ์ นายวุฒิภัทร โขคงนันท์ทรัพย์
อาจารย์ที่ปรึกษา	ที่ปรึกษาวิทยานิพนธ์หลัก นายธนัชชา ชูพจน์เจริญ ที่ปรึกษาวิทยานิพนธ์ร่วม รศ.ดร.ชิต เหล่าวัฒนา
หลักสูตร	วิศวกรรมศาสตรบัณฑิต
สาขาวิชา	วิศวกรรมหุ่นยนต์และระบบอัตโนมัติ
คณะ	สถาบันวิทยาการหุ่นยนต์ภาคสนาม
ปีการศึกษา	2560

---

## บทคัดย่อ

งานวิทยานิพนธ์นี้เป็นงานที่เกี่ยวกับการออกแบบและจัดทำแพลตฟอร์มหุ่นยนต์อิเล็กทรอนิกส์ด้วยเครื่องพิมพ์สามมิติ โดยใช้ชื่อว่า หุ่นยนต์อิเล็กทรอนิกส์ UTHAI และจุดประสงค์เพื่อให้นักวิจัยท่านอื่นสามารถนำไปพัฒนาต่อได้ง่าย ภาพรวมของวิทยานิพนธ์นี้จะแบ่งออกเป็นทั้งหมดสามส่วน คือ ส่วนแรกเป็นส่วนของการออกแบบและจัดสร้างส่วนโครงสร้างของหุ่นยนต์อิเล็กทรอนิกส์ ส่วนที่สองเป็นส่วนของการพัฒนาโปรแกรมที่ใช้ในระบบด้วย ROS และส่วนสุดท้ายเป็นส่วนที่ออกแบบระบบพื้นฐานสำหรับการพัฒนาหุ่นยนต์อิเล็กทรอนิกส์ รวมไปถึงมีเอกสาร คู่มือที่อยู่ในรูปแบบออนไลน์

คำสำคัญ : แพลตฟอร์มหุ่นยนต์อิเล็กทรอนิกส์ / ระบบพื้นฐานหุ่นยนต์ / ROS / เครื่องพิมพ์สามมิติ

## กิตติกรรมประกาศ

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

ขอขอบพระคุณรองศาสตราจารย์ ดร.ชิต เหล่าวัฒนา อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม ที่ได้ชี้แนะแนวทางให้คำแนะนำ และให้เกียรติเข้าร่วมการสอบวิทยานิพนธ์

ขอขอบพระคุณอาจารย์ ดร.ภิวดา มณีวรรณ และนายวิษณุ จุราวี ที่ได้ให้คำแนะนำในการแก้ไขปัญหาด้านต่างๆ ที่เกิดขึ้นระหว่างการทำวิจัย และได้ให้การสนับสนุนอุปกรณ์สำคัญที่ใช้ในการทำวิจัย

ขอขอบพระคุณอาจารย์ อับพิพิญ ธิรวงศ์กิจ ที่กรุณาให้เกียรติเป็นประธานกรรมการสอบวิทยานิพนธ์ ให้คำแนะนำที่เป็นประโยชน์ต่อการจัดทำวิทยานิพนธ์ให้ดำเนินไปอย่างสมบูรณ์

ขอขอบพระคุณอาจารย์ ดร.ปิติวุฒย์ ธีรกิตติกุล ที่กรุณาให้เกียรติเป็นกรรมการสอบวิทยานิพนธ์ ให้คำแนะนำที่เป็นประโยชน์ต่อการวิจัย และการแก้ไขปรับปรุงงานวิจัย ตลอดจนตรวจแก้วิทยานิพนธ์ให้ดำเนินไปอย่างสมบูรณ์

ขอขอบพระคุณอาจารย์ ดร.สุภาชัย วงศ์บุณย์ยง ที่กรุณาให้เกียรติเป็นกรรมการสอบวิทยานิพนธ์ ให้คำแนะนำที่เป็นประโยชน์ต่อการวิจัย และการแก้ไขปรับปรุงงานวิจัย ตลอดจนตรวจแก้วิทยานิพนธ์ให้ดำเนินไปอย่างสมบูรณ์

ขอขอบพระคุณคณาจารย์ และบุคลากรในสถาบันวิทยาการหุ่นยนต์ภาคนามทุกท่าน ที่ได้ให้คำปรึกษาและช่วยเหลือด้านสถานที่พร้อมทั้งส่งเสริมความต้องการต่างๆ ในระหว่างการทำวิทยานิพนธ์

ขอขอบคุณนักศึกษาปริญญาตรี สถาบันวิทยาการหุ่นยนต์ภาคนามทุกท่าน ที่ได้ให้คำแนะนำ ถ้ามี แล้วเป็นกำลังใจมาโดยตลอด

และสุดท้ายนี้ ขอน้อมรำลึกถึงพระคุณบิดา มารดา และครอบครัว ที่ส่งเสริมให้กำลังใจ และให้การสนับสนุนในเรื่องต่างๆ จนกระทั้งข้าพเจ้าประสบความสำเร็จในการศึกษา

นายจิรภูริศ ศรีรัตนอาภรณ์  
นายเจษฎากร หาไซวงศ์  
นายวุฒิภัทร โชคอนันตทรัพย์

## สารบัญ

เรื่อง

หน้า

## รายการรูปภาพ

รูป

หน้า

## รายการตาราง

ตาราง

หน้า

## บทที่ 1

### บทนำ

#### 1.1 ที่มาและความสำคัญ

หุ่นยนต์อิวามานอยด์เป็นหุ่นยนต์ที่สร้างขึ้นเพื่อเลียนแบบสรีระร่างกายของมนุษย์ ซึ่งมีข้อจำกัดนวนมากเพื่อให้มีการเคลื่อนไหวคล้ายมนุษย์ ลักษณะเด่นของหุ่นยนต์อิวามานอยด์คือ การเคลื่อนที่ด้วยขาสองข้าง ซึ่งทำให้หุ่นยนต์สามารถเคลื่อนที่ได้อย่างคล่องแคล่วในทุกสภาพพื้นผิว ทั้งทางเรียบ ทางชันและหีบหัก ระดับ<sup>1</sup> ซึ่งนั่นทำให้หุ่นยนต์ที่เดินสองขาแตกต่างจากหุ่นยนต์ที่เคลื่อนที่ด้วยล้อ ด้วยโครงสร้างของหุ่นยนต์ที่คล้ายมนุษย์นั้นเอง จึงทำให้หุ่นยนต์อิวามานอยด์สามารถทำงานได้หลากหลายและยืดหยุ่น สามารถที่จะใช้อุปกรณ์ทั่วไปที่ถูกออกแบบขึ้นมาเพื่อใช้กับมนุษย์ได้ ซึ่งหมายความว่าในอนาคตนั้นหุ่นยนต์อิวามานอยด์สามารถที่จะทำงานทดแทนแรงงานของมนุษย์ได้<sup>2</sup> งานที่หุ่นยนต์อิวามานอยด์จะเข้ามาทดแทนแรงงานของมนุษย์นั้น จะเป็นงานที่ต้องทำซ้ำๆ จนเกินความเมื่อยล้า งานที่อยู่ในพื้นที่อันตรายหรือที่เสี่ยงต่อการเกิดอุบัติเหตุ

สถาบันวิจัยหลายแห่งทั่วโลกกำลังให้ความสนใจสนับสนุนด้านการศึกษาวิจัยและพัฒนาหุ่นยนต์อิวามานอยด์ เพื่อให้ทำการกิจกรรมต่างๆ ยกตัวอย่างเช่น DARPA Robotics Challenge (DRC)<sup>3</sup> เป็นรายการแข่งขันหุ่นยนต์กึ่งอัตโนมัติเพื่อทำการกิจกรรมในสถานการณ์ภัยพิบัติที่อันตราย ซึ่งสถาบันวิจัยหุ่นยนต์ทั่วโลกได้ส่งหุ่นยนต์อิวามานอยด์ของตนเข้าร่วมการแข่งขัน ในปัจจุบันได้มีการพัฒนาหุ่นยนต์อิวามานอยด์ขึ้นมาหลายรายการตัวเช่น ASIMO, HRP-3, LOLA และ WATHLITE-1 การพัฒนาหุ่นยนต์อิวามานอยด์นั้นได้ก่อให้เกิดงานศึกษาวิจัย และทดสอบต่อยอด ต่างๆ มากมาย เช่น การวางแผนการเดิน การเดินแบบสติติ การเดินแบบพลวัต การติดต่อสื่อสารของระบบ การมองเห็นและการประมวลผลภาพ การพูดคุยกับมนุษย์ ปัญญาประดิษฐ์ ฯลฯ ซึ่งงานวิจัยเหล่านี้สามารถที่จะนำไปประยุกต์ใช้กับระบบหุ่นยนต์ระบบอื่นๆ ได้ แม้ว่าจะมีการพัฒนาหุ่นยนต์อิวามานอยด์มากรามาก แล้ว แต่การเริ่มต้นทำงานวิจัยที่มีความเกี่ยวข้องกับหุ่นยนต์อิวามานอยด์นั้น ต้องใช้ความรู้ความสามารถ เครื่องมือ ระยะเวลา งบประมาณ และ ความพยายามเป็นอย่างมาก การสร้างหุ่นยนต์อิวามานอยด์ขึ้นมาใหม่นั้นต้องใช้งบประมาณสูง ดังนั้นการสร้างระบบจำลองของหุ่นยนต์อิวามานอยด์ขึ้นมาเป็นระบบพื้นฐาน ให้มีความพร้อมสำหรับการพัฒนาต่อยอดแก้ไขศึกษาหรือนักวิจัย จะช่วยประหยัดเวลาและงบประมาณที่ต้องใช้เดือยอย่างมาก ซึ่งนั่นหมายความว่านักวิจัยจะสามารถทำงานวิจัยได้อย่างมีประสิทธิภาพมากขึ้น

ในงานวิจัยนี้เป็นการออกแบบหุ่นยนต์อิวามานอยด์และพัฒนาระบบที่พื้นฐานของหุ่นยนต์อิวามานอยด์ สำหรับให้นักศึกษาหรือนักวิจัยสามารถพัฒนาต่อยอดได้ โดยหุ่นยนต์อิวามานอยด์ที่ออกแบบมานั้น สามารถที่จะปรับปรุงแก้ไข ดัดแปลงได้จ่าย ตัวโครงสร้างจะใช้เป็น พลาสติก PLA ที่สามารถขึ้นรูปได้ โดยการใช้เครื่องพิมพ์สามมิติ มีเซนเซอร์ตรวจการสัมผัสพื้นที่ฝ่าเท้าของหุ่นยนต์ มีเซนเซอร์สำหรับการวัดมุมเอียง ที่ลำตัวของหุ่นยนต์ และเพื่อที่จะทำให้ง่ายต่อการศึกษาทำความเข้าใจ บำรุงรักษา จึงได้มีการจัดทำคู่มือและเอกสารวิธีการใช้งานอย่างชัดเจน โดยจะเก็บในรูปแบบของเอกสารออนไลน์

<sup>1</sup>การเคลื่อนที่ของหุ่นยนต์รูปแบบต่างๆ รายวัสดุสนับสนุน

<sup>2</sup>ผู้ริชพ์ วรีประเสริฐ และณรงค์ ล้ำดี (2552: 374)

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

## 1.2 วัตถุประสงค์

ผู้วิจัยมีวัตถุประสงค์ในการทำวิทยานิพนธ์เกี่ยวกับหุ่นยนต์ชีวามโนยด์แพลตฟอร์มนี้ขึ้นมา กี เพื่อที่จะออกแบบโครงสร้างของหุ่นยนต์ชีวามโนยด์ที่สามารถแก้ไขปรับเปลี่ยนได้ง่าย พัฒนาระบบพื้นฐาน ระบบจำลอง สำหรับหุ่นยนต์ชีวามโนยด์เพื่อใช้ในการศึกษาวิจัย รวบรวมเครื่องมือที่เป็นประโยชน์ต่อการพัฒนาหุ่นยนต์ และจัดทำเอกสารออนไลน์ ให้บุคคลที่สนใจสามารถเข้ามาศึกษาได้

## 1.3 ประโยชน์ที่คาดว่าจะได้รับ

- 1 มีต้นแบบหุ่นยนต์ชีวามโนยด์สำหรับใช้ในงานวิจัยแขนงต่างๆ
- 2 มีระบบพื้นฐานสำหรับพัฒนาหุ่นยนต์ชีวามโนยด์รุ่นใหม่ในสถาบัน
- 3 มีระบบจำลองสำหรับจำลองการทำงานของหุ่นยนต์ชีวามโนยด์
- 4 มีแหล่งรวมเครื่องมือสำหรับการพัฒนาหุ่นยนต์
- 5 มีคู่มือ เอกสาร วิธีการใช้งาน และรายละเอียดของหุ่นยนต์สำหรับพัฒนาต่อยอด

## 1.4 ขอบเขตการดำเนินงาน

- 1 ใช้ ROS เป็นกรอบการทำงานสำหรับพัฒนาระบบพื้นฐาน
- 2 ออกแบบโครงสร้างให้มีความแข็งแรง สามารถรับน้ำหนักอุปกรณ์ที่ติดตั้งอยู่บนตัวหุ่นยนต์ได้
- 3 น้ำหนักของหุ่นยนต์รวมกันทั้งตัว ไม่เกิน 5 กิโลกรัม
- 4 ใช้ Solidworks 3D เป็นโปรแกรมสำหรับออกแบบโครงสร้าง และคำนวณ
- 5 หุ่นยนต์มีความสูงไม่ต่ำกว่า 100 เซนติเมตร และสูงไม่เกิน 120 เซนติเมตร
- 6 หุ่นยนต์มี 2 แขน 2 ขา มีองศาอิสระของขาข้างละ 6 และแขนข้างละ 2 องศาอิสระ
- 7 หุ่นยนต์สามารถทำงานได้ภายในสภาพแวดล้อมแบบปิด
- 8 หุ่นยนต์ใช้พลังงานจากแหล่งจ่ายพลังงานที่มีขนาดแรงดันไฟฟ้า 12 โวลต์
- 9 หุ่นยนต์ใช้ตัวขับเคลื่อนแบบดิจิตอลสำหรับแต่ละข้อต่อเป็น Dynamixel Digital Servo
- 10 ใช้ Gazebo สำหรับจำลองระบบของหุ่นยนต์
- 11 ติดตั้งเซนเซอร์วัดการกดที่ฝ่าเท้าของหุ่นยนต์ (Ground contact)
- 12 ติดตั้งเซนเซอร์หน่วยวัดความเร็วที่บริเวณลำตัวของหุ่นยนต์ (IMU)
- 13 จัดทำคู่มือ เอกสารการใช้งาน และรายละเอียดส่วนประกอบของหุ่นยนต์

## 1.5 ภาพรวมของระบบและขั้นตอนการดำเนินงาน

งานวิจัยนี้การดำเนินงานวิจัยถูกแบ่งออกเป็นสามส่วน คือ ส่วนที่หนึ่งส่วนโครงสร้างของหุ่นยนต์ชีวามนอยด์ เป็นส่วนที่ทำในส่วนของการขึ้นรูปชิ้นงาน ออกแบบโมเดลสามมิติ รวมไปถึงระบบอิเล็กทรอนิกส์ ติดตั้งบอร์ดและเซนเซอร์ไวตามจุดต่างๆ เพื่อสร้างโครงสร้างของหุ่นยนต์ให้สามารถรองรับการเดินได้ ส่วนที่สองส่วนโปรแกรมของหุ่นยนต์ชีวามนอยด์ เป็นส่วนที่ทำในส่วนของการสั่งการตัวขับเคลื่อนต่างๆ อันค่าสถานะเซนเซอร์จากคอนโโอลเลอร์ รวมไปถึงระบบจำลองการทำงานของหุ่นยนต์ และส่วนที่สามส่วนการออกแบบระบบพื้นฐานเพื่อการพัฒนาต่ออยอด ส่วนนี้จะเป็นส่วนที่ทำให้ผู้ที่จะมาวิจัยต่ออยอดสามารถทำงานได้ง่ายขึ้น จัดการเอกสารคู่มือ การใช้งานต่างๆให้เป็นระบบระเบียบ สามารถแยกขั้นตอนการทำงานของแต่ละส่วนออกเป็นข้อดังนี้

### ศึกษาค้นคว้าเอกสารและงานวิจัยที่เกี่ยวข้อง

- ศึกษาเกี่ยวกับส่วนประกอบของหุ่นยนต์ชีวามนอยด์
- ศึกษาทฤษฎีที่เกี่ยวข้องกับของมนุษย์
- ศึกษาทฤษฎีที่เกี่ยวข้องกับหุ่นยนต์ชีวามนอยด์
- ศึกษาความแตกต่างระหว่างมนุษย์กับหุ่นยนต์ชีวามนอยด์
- ศึกษาวิธีการและวัสดุที่ใช้ในการสร้างหุ่นยนต์
- ศึกษาระบบที่ใช้ในการจำลองการทำงานของหุ่นยนต์
- ศึกษาระบบที่ใช้ในการจำลองการทำงานของหุ่นยนต์
- ศึกษาการใช้งาน ROS พื้นฐาน

### 1) ส่วนโครงสร้างของหุ่นยนต์ชีวามนอยด์

- ออกแบบโครงสร้างของหุ่นยนต์ชีวามนอยด์
- จัดสร้างโครงสร้างหุ่นยนต์ชีวามนอยด์
- ทดสอบการทำงานของหุ่นยนต์ชีวามนอยด์

### 2) ส่วนโปรแกรมของหุ่นยนต์ชีวามนอยด์

- ออกแบบโปรแกรมของหุ่นยนต์ชีวามนอยด์
- สร้างโปรแกรมส่วนล่างสำหรับหุ่นยนต์ชีวามนอยด์
- สร้างโปรแกรมส่วนบนสำหรับหุ่นยนต์ชีวามนอยด์
- ทดสอบการทำงานของโปรแกรม

### 3) ส่วนการออกแบบระบบพื้นฐานเพื่อการพัฒนาต่ออยอด

- ติดตั้งระบบ
- วางระบบพื้นฐาน
- รวบรวมเครื่องมือที่เป็นประโยชน์
- จัดทำคู่มือ

## บทที่ 2

### ทฤษฎีและหลักการ

#### 2.1 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

##### 2.1.1 หุ่นยนต์อิริวามาอยด์

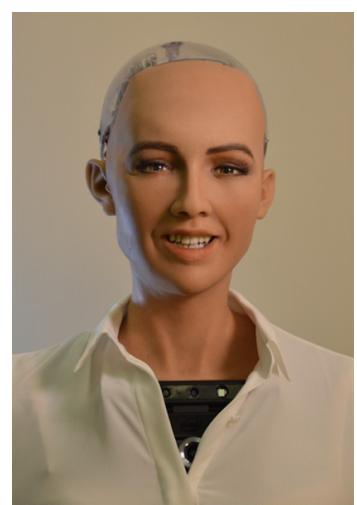
หุ่นยนต์อิริวามาอยด์ คือ หุ่นยนต์ที่ถูกสร้างขึ้นมาให้มีรูปร่างคล้ายคลึงกับสรีระโครงสร้างของมนุษย์ มักถูกออกแบบมาเพื่อจุดประสงค์เฉพาะอย่าง เช่น การใช้เครื่องมือต่างๆของมนุษย์ การอยู่ในสภาพแวดล้อมของมนุษย์ การศึกษาการเคลื่อนไหวของร่ายกายมนุษย์ หรือเพื่อวัตถุประสงค์อื่นๆ โดยทั่วไปแล้ว หุ่นยนต์อิริวามาอยด์ประกอบไปด้วย 4 ส่วนคือ หัว ลำตัว เแขน และขา แต่การสร้างหุ่นยนต์อิริวามาอยด์นั้นก็ไม่จำเป็นที่จะต้องมีส่วนประกอบทุกส่วนดังที่กล่าวไป ในบางครั้งอาจมีเพียงแค่ส่วนบนเท่านั้น ดังรูปที่ ??<sup>1</sup> หุ่นยนต์นี้มาจากสถาบันวิทยาการหุ่นยนต์ภาคนามา เป็นหุ่นยนต์ที่มีส่วนบนเหมือนมนุษย์ แต่มีส่วนล่างเป็นล้อ หรือหุ่นยนต์อิริวามาอยด์ที่มีเพียงแค่ส่วนล่าง ดังรูปที่ ??<sup>2</sup> หุ่นยนต์ส้มจุก เป็นหุ่นยนต์อิริวามาอยด์ที่มีเพียงแค่ส่วนขาเท่านั้น หรือหุ่นยนต์อิริวามาอยด์ที่มีเพียงใบหน้าเหมือนมนุษย์ ดังรูปที่ ??<sup>3</sup> หุ่นยนต์โซเฟีย เป็นแอนดรอยด์ที่มีหน้าตาคล้ายมนุษย์มาก มีตา มีปาก สามารถพูดปฏิสัมพันธ์กับมนุษย์ได้



(ก) หุ่นยนต์ประชาสัมพันธ์โนม



(ข) หุ่นยนต์เดินสองขาส้มจุก



(ค) หุ่นยนต์แอนดรอยด์โซเฟีย

รูปที่ 2.1: แสดงความแตกต่างของหุ่นยนต์อิริวามาอยด์แต่ละประเภท

<sup>1</sup> คนไทยสุดเจ็บ!! สร้างหุ่นยนต์ 'น้องโนม' ทำหน้าที่ต้อนรับแทนคน ทั้งไทย-พูดหลายภาษา, <https://www.thairath.co.th/content/523340>

<sup>2</sup> หุ่นยนต์อิริวามาอยด์เดินสองขาส้มจุก, <http://www.fibo.kmutt.ac.th/fiboweb2015/> หุ่นยนต์อิริวามาอยด์เดิน

<sup>3</sup> ชา อุติ อาราเบีย มอบ สัญชาติ ให้ กับ หุ่น ยนต์ ครั้ง แรก ของ โลก “Sophia” สาว อัจฉริยะ, <https://www.ensurecommunication.com/2017/11/02/ชาอุติอาราเบียมอบสัญชาติ>

งานวิจัยทางด้านหุ่นยนต์ชีวมานอยด์จากอดีตจนถึงปัจจุบันมีการพัฒนาความสามารถของการเดินของหุ่นยนต์ เช่น เริ่มต้นจากแรกสุดจะเป็นการพัฒนาให้หุ่นยนต์สามารถเดินหน้าได้ ต่อมาเกิดเพิ่มความสามารถให้หุ่นยนต์สามารถเดินบนพื้นเยียบ พื้นชุ่มชื้น เดินเลี้ยวซ้ายขวา เดินขึ้นลงบันได ฯลฯ เป็นต้น นอกจากนี้ยังมีการพัฒนาปรับปรุงสมดุลของการเดินแบบสองขาอีกด้วย สมดุลของการเดินสามารถแบ่งได้สองแบบหลัก คือ การเดินแบบสมดุลสถิต และการเดินแบบสมดุลพลวัต งานในยุคแรกนั้นจะพัฒนาให้เดินได้แบบสมดุลสถิต ต่อมาเป็นสมดุลกึ่งพลวัต 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 ดังนั้นการที่หุ่นยนต์ขยับลำตัวไปด้านหน้าจะทำให้น้ำหนักโดยรวมของหุ่นยนต์กลับมาอยู่ตรงกลางของพื้นที่รับน้ำหนักเหมือนเดิม ซึ่งจะทำให้หุ่นยนต์มีความสมดุลได้ ดังนั้นข้อมูลที่ได้จากหน่วยวัดแรงกดที่เท้าจะถูกนำมาคำนวณตลอดการเดินเพื่อใช้ในการปรับเปลี่ยนมุกการขยับของลำตัว การเดินบนพื้นราบเป็นแบบสมดุลสถิตและการเดินบนพื้นเอียงก็ยังคงเป็นแบบสมดุลสถิตเช่นกัน

Inaba<sup>4</sup> และคณะ (1995) สร้างหุ่นยนต์เลียนแบบลิ่ง (ape-like biped) ประกอบด้วยสองมือและสองขา มีการเดินแบบสมดุลสถิติ งานวิจัยนี้มีความคิดว่า wennen ทำการทำให้หุ่นยนต์สองขาเดินได้โดยไม่ล้มแล้ว ควรจะทำหุ่นยนต์ที่สามารถถabilize ขึ้นเองได้หลังจากที่ล้มแล้วด้วย ดังนั้นในงานนี้ หุ่นยนต์ถูกพัฒนาให้สามารถเดิน เมื่อล้มแล้ว ก็สามารถฟลิกตัวและลากขึ้นมาเดินให้ได้

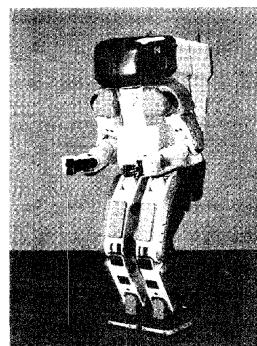
Kun และ Miller<sup>5</sup> (1996) ได้นำโครงข่ายประสาทเทียม มาประยุกต์ใช้ในการปรับเปลี่ยนท่าทางการเดิน โดยอัตโนมัติของหุ่นยนต์สองขา การที่หุ่นยนต์สามารถปรับเปลี่ยนท่าทางได้โดยอัตโนมัตินี้มีประโยชน์ทำให้หุ่นยนต์เดินได้บันทึ้นผืนผ้าหลากหลายลักษณะมากขึ้น ในงานนี้พิจารณาทั้งสมดุลในแนวหน้าหลัง (sigittal plane) และแนวซ้ายขวา (frontal plane) และการเดินของหุ่นยนต์เป็นแบบสมดุลพลวัต หลักการทำงานประกอบด้วยตัวสร้างท่าทางการเดินหนึ่งตัว และตัวปรับท่าทางการเดินทั้งแนวหน้าหลังและซ้ายขวาอีกหนึ่งตัว โดยค่าการปรับเปลี่ยนนั้นจะได้มาจากการรับสัญญาณจากเซ็นเซอร์ต่างๆ เช่น กล้อง CCD ที่ติดอยู่บนหัวหุ่นยนต์ ที่จะส่งค่าความสูงของศีรษะ ความกว้างของศีรษะ ความกว้างของลำตัว ความกว้างของขา ความกว้างของตัว และความสูงของตัว ที่จะถูกนำมาคำนวณเพื่อปรับเปลี่ยนท่าทางการเดินของหุ่นยนต์ (Kun and Miller, 1997)

---

<sup>4</sup>Yuki Asano\*, Kei Okada and Masayuki Inaba

<sup>5</sup>Modelling of Walking Humanoid Robot With Capability of Floor Detection and Dynamic Balancing Using Colored Petri Net, Saeid Pashazadeh and Saeed Saeedvand

Hirai<sup>6</sup> และคณะ (1998) พัฒนาหุ่นยนต์อิวามาโนยด์ ซึ่งตัวหุ่นยนต์มีความคล้ายมนุษย์มาก สามารถเดินได้อย่างราบรื่นคล้ายมนุษย์มากที่สุด เช่น สามารถเดินได้ในพื้นผิวนิ่มต่างๆ เดินได้บนพื้นอิฐอิฐเดินขึ้ลงบันไดได้ เดินเข็นรถได้ เป็นต้น การเดินในทุกสถานการณ์เป็นการเดินแบบสมดุลพลวัต หุ่นยนต์สามารถเดินได้ด้วยความเร็วสูงสุด 4.7 กิโลเมตรต่อชั่วโมง หุ่นยนต์ประกอบไปด้วย แขนขาละ 9 องศาอิสระ ขาละ 6 องศาอิสระ ที่ปรับรีเควนหัวมีกล้องติดตั้งอยู่ 4 ตัว นอกจากนี้ยังมีอุปกรณ์ที่ใช้ในการรักษาสมดุลอื่นๆ อีกได้แก่ IMU ที่ติดตั้งบริเวณลำตัว และ Force sensor ที่ติดที่เท้าทั้งสองข้าง



รูปที่ 2.2: Honda asimo โดย Kazou Hirai

องค์ประกอบของหุ่นยนต์ทั่วไปจะประกอบไปด้วยระบบการตอบสนองต่างๆที่เป็นระบบ ซึ่งเราสามารถจำแนกการออกเป็นส่วนหลักๆได้สามส่วนคือ ส่วนการรับรู้ ส่วนการประมวลผล และส่วนการขับเคลื่อน ทั้งหมด เมื่อนำมารวมเข้าด้วยกันแล้ว เราสามารถที่จะควบคุมการทำงานของหุ่นยนต์อิวามาโนยด์ได้

### การรับรู้ของหุ่นยนต์อิวามาโนยด์

การรับรู้ของหุ่นยนต์อิวามาโนยด์นั้นมีความยากมากกว่าหุ่นยนต์ชนิดอื่นๆเนื่องจาก หุ่นยนต์อิวามาโนยด์ เกิดจากการนำก้านต่อหularyาชิ้นเข้ามาเขื่อมต่อกันด้วยข้อต่อ ทำให้หุ่นยนต์อิวามาโนยด์สามารถเคลื่อนไหวเป็นท่าทางต่างๆได้ และไม่มีส่วนใดถูกต้องยึดติดกับพื้นโลก ซึ่งทำให้เราไม่สามารถที่จะอ้างอิงท่าทางของหุ่นยนต์ได้ จึงจำเป็นที่จะต้องเพิ่มส่วนของการรับรู้เข้าไปเพื่อช่วยแก้ปัญหาในส่วนนี้ เช่นเซอร์วิสเพิ่มเข้าไปมีหลากหularyนิด และแต่ละชนิดก็ทำหน้าที่แตกต่างกัน ยกตัวอย่างเช่น เช่นเซอร์วิสโอนโคดเดอร์ที่ใช้สำหรับอ่านสถานะตำแหน่งและความเร็วของข้อต่อได้ เช่นเซอร์วิสหน่วยวัดความเร็วอยที่ใช้สำหรับหามุมเอียงของตัวหุ่นยนต์ และเซนเซอร์วัดแรงที่ฝ่าเท้าที่จะช่วยในการบอกว่าเท้าของหุ่นยนต์สัมผัสพื้นหรือไม่ เป็นต้น

<sup>6</sup>Kazuo Hirai, (1999) "The Honda humanoid robot: development and future perspective", Industrial Robot: An International Journal, Vol. 26 Issue: 4, pp.260-266, <https://doi.org/10.1108/01439919910277431>

### **การประมวลผลของหุ่นยนต์อิวามาอยด์**

ในปัจจุบันนี้หน่วยประมวลผลของหุ่นยนต์อิวามาอยด์มีประสิทธิภาพเพียงพอต่อการควบคุมหุ่นยนต์อิวามาอยด์แบบเรียลไทม์ได้ การประมวลผลนั้นสามารถที่จะแบ่งออกเป็นหลายๆส่วนได้ ยกตัวอย่างเช่น

**หุ่นยนต์อิวามาอยด์ Thormang** ได้แบ่งตัวประมวลผลเป็นตัวประมวลควบคุมสำหรับสั่งการตัวขับเคลื่อน ตัวประมวลควบคุมสำหรับอ่านสถานะตัวรับรู้ และตัวประมวลผลควบคุมภายนอกสำหรับคำนวณท่าทางการเดินและการวางแผน

**หุ่นยนต์อิวามาอยด์ Robotis OP3** ได้แบ่งตัวประมวลผลเป็นตัวประมวลผลระดับสูงสำหรับคำนวณท่าทางการเคลื่อนไหว และตัวประมวลผลระดับล่างสำหรับสั่งการตัวขับเคลื่อนและอ่านสถานะตัวรับรู้

**หุ่นยนต์อิวามาอยด์ Poppy** ไม่ได้แบ่งตัวประมวลผล แต่ใช้เพียงตัวเดียวในการสั่งการตัวขับเคลื่อนอ่านสถานะตัวรับรู้ และประมวลผลการคำนวณทั้งหมด

นอกจากการประมวลผลและควบคุมแล้ว ยังรวมไปถึงหน่วยแสดงผลที่สามารถนำค่าสถานะต่างๆจากหุ่นยนต์อิวามาอยด์ออกไปสร้างเป็นกราฟ หรือแบบจำลองสามมิติได้อีกด้วย

### **การขับเคลื่อนของหุ่นยนต์อิวามาอยด์**

การที่หุ่นยนต์อิวามาอยด์สามารถที่จะทำท่าทางต่างๆได้นั้น เกิดจากการเปลี่ยนแปลงตำแหน่งของข้อต่อตั้งนั้น การที่หุ่นยนต์จะขับเคลื่อนข้อต่อได้นั้น ก็จึงจำเป็นที่จะต้องมีตัวขับเคลื่อนที่ข้อต่อ โดยที่ไปแล้วเราจะติดมอเตอร์เซอร์โวไว้ที่ตำแหน่งเดียวกับข้อต่อเลย เหตุผลที่นิยมใช้มอเตอร์เซอร์โวนี้องจาก เซอร์โวสามารถที่จะควบคุมตำแหน่งและความเร็วได้

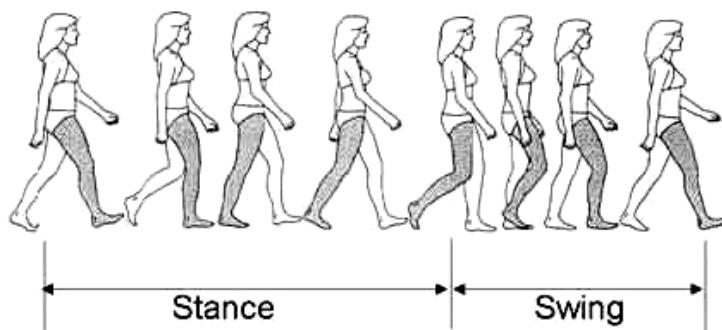
## 2.1.2 ทฤษฎีเกี่ยวกับมนุษย์

### 2.1.2.1 การวิเคราะห์การเดินของมนุษย์

การเคลื่อนที่ของหุ่นยนต์อิวามานอยด์นั้นจะเลียนแบบจากการเดินของมนุษย์ ดังนั้นการวิเคราะห์ลักษณะการเดินของมนุษย์ จะเป็นการศึกษาเพื่อทำความเข้าใจถึงธรรมชาติการเดิน ก่อนนำไปทำการออกแบบกลไกทางกลและระบบควบคุมของหุ่นยนต์อิวามานอยด์ การก้าวเดินของมนุษย์โดยปกติแล้ว จะมีลักษณะเป็นวัฏจักร วนซ้ำไปเรื่อยๆ ในทิศทางที่ต้องการจนกว่าจะทำการหยุดเดิน การทรงตัวในระหว่างการยืนหรือการเดินนั้น เป็นไปตามสัญชาตญาณซึ่งเกิดจากการรักษาความสมดุลของร่างกายบัน្តาในทุก<sup>7</sup> ส่งสัญญาณผ่านเส้นประสาทไปยังกล้ามเนื้อส่วนต่างๆ ที่ทำหน้าที่ให้เกิดการเคลื่อนที่

การเคลื่อนที่ของมนุษย์ในการเดินไปข้างหน้าสามารถแบ่งออกเป็นช่วงๆ ดังนี้

### GAIT CYCLE



รูปที่ 2.3: วัฏจักรการเดินของมนุษย์

1. ช่วงเริ่มการวางเท้าเพื่อเข้าสู่ช่วงเริ่มต้นเหวี่ยงเท้า เป็นช่วงที่เท้าเกิดการกระแทกลงบนพื้นหลังจากทำการเหวี่ยงมาจากด้านหลัง โดยธรรมชาติมนุษย์จะทำการวางสันเท้าลงเพื่อลดแรงกระแทกที่เกิดขึ้นในช่วงนี้ ดังนั้นทางกายภาพในส่วนของสันเท้ามนุษย์จึงมีลักษณะอ่อนนุ่ม
2. ช่วงเริ่มต้นเหวี่ยงเท้าเพื่อเข้าสู่ช่วงเหวี่ยงเท้า หลังจากทำการวางสันเท้าลงกับพื้นแล้ว ข้อเข้าจะปรับมุมเพื่อให้ฝ่าเท้าแนวพื้นสนิท ขณะเดียวกันขาอีกข้างจะยกสูงขึ้นเพื่อถ่ายเทน้ำหนักไปยังเท้าที่เพิ่งวางลง
3. ช่วงเหวี่ยงเท้า เป็นช่วงที่ขาหนีงยกอย้อยู่ในอากาศและขาที่วางแนบกับพื้นจะรองรับน้ำหนักทั้งหมดของร่างกาย
4. ช่วงเตรียมการวางเท้า เป็นช่วงที่ขาข้างที่ลอยอยู่เหวี่ยงไปข้างหน้าเพื่อเตรียมเข้าสู่ช่วงรองรับ ในขณะเดียวกันขาที่รับน้ำหนักอยู่จะทำการผลักตัวเพื่อเริ่มทำการถ่ายเทน้ำหนักไปข้างหน้า

<sup>7</sup>Rose, J. and Gamble, J., 1993, Human Walking, Williams & Wilkins, Philadelphia, pp. 10-44.

### 2.1.2.2 การวิเคราะห์องศาอิสระของมนุษย์

การที่มนุษย์สามารถเคลื่อนที่ได้เป็นผลเนื่องมาจากการเคลื่อนที่ของข้อต่อต่างๆ ซึ่งประกอบไปด้วย ข้อต่อส่วนสะโพก ส่วนหัวเข่า และส่วนข้อเท้า แรงบิดที่เกิดขึ้นของแต่ละข้อต่อมีความสัมพันธ์ต่อกัน ส่งผลให้เกิดเสถียรภาพในการเดินของมนุษย์ เมื่อวิเคราะห์ลักษณะโครงสร้างในแต่ละส่วน พบว่าข้อต่อส่วนสะโพกมีลักษณะเป็นทรงกลม ทำให้ข้อต่อส่วนสะโพกสามารถหมุนได้ 3 องศาอิสระ ส่วนหัวเข่าของมนุษย์มีจุดต่อของข้อที่มีลักษณะเป็นทรงกลม สองลูกประกอบเข้าด้วยกันทำให้การเคลื่อนที่ถูกบังคับให้สามารถเคลื่อนที่ได้เพียง 1 องศาอิสระ ในส่วนของข้อเท้ามีลักษณะการเคลื่อนที่เหมือนสะโพกคือสามารถเคลื่อนที่ได้ 3 องศาอิสระ

จากทั้งหมดที่ได้ทำการวิเคราะห์มาข้างต้นพบว่าในขาหนึ่งข้างของมนุษย์ประกอบด้วย 7 องศาอิสระ ซึ่งส่งผลให้การเคลื่อนที่ของมนุษย์มีความคล่องแคล่วสูง แต่ในทางออกแบบกลไกการเดินและการควบคุม ของหุ่นยนต์สองขาถือว่ามีจำนวนองศาอิสระเกินความจำเป็นในการเคลื่อนที่บนปริภูมิ (space) และยากต่อการควบคุม (underactuated) ดังนั้นการกำหนดจำนวนองศาอิสระเพื่อให้หุ่นยนต์เดินได้สมดุลมนุษย์จึงมีผลในการออกแบบกลไกทางกลและการควบคุมของหุ่นยนต์สองขา

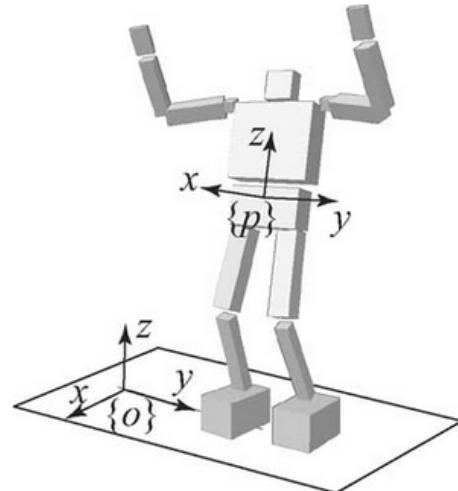
ข้อต่อ	องศาอิสระ	องศาการหมุน	
		สูงสุด	ต่ำสุด
หัว	$\theta_x$	+60	-30
	$\theta_y$	+70	-70
	$\theta_z$	+80	-80
หลัง	$\theta_x$	+30	-30
	$\theta_y$	+55	-55
	$\theta_z$	+45	-45
หัวไหล่	$\theta_x$	+180	-80
	$\theta_y$	+45	-135
	$\theta_z$	+30	0
ศอก	$\theta_x$	0	-155
สะโพก	$\theta_x$	+120	-40
	$\theta_y$	+40	-50
	$\theta_z$	+60	-50
หัวเข่า	$\theta_x$	0	-130
ข้อเท้า	$\theta_x$	+30	-60
	$\theta_y$	+45	-20
	$\theta_z$	+20	-60

ตารางที่ 2.1: ความสามารถในการหมุนของแต่ละข้อต่อของมนุษย์

ผู้เขียนได้ข้อสรุปในการออกแบบขาหนึ่งข้างของหุ่นยนต์ให้มีองศาอิสระเท่ากับ 6 องศาอิสระ และได้ใช้ติดตอลเซอร์โวของบริษัท Robotis เป็นตัวขับเคลื่อนข้อต่อ เนื่องจากภายในเซอร์โวมีตัวรับรู้สถานะของตัวเอง และเซอร์โวนี้ถูกออกแบบมาให้สามารถติดตั้ง และสั่งการได้ง่าย

### 2.1.3 ทฤษฎีเกี่ยวกับหุ่นยนต์อิวามาโนยด์

#### 2.1.3.1 ส่วนประกอบของหุ่นยนต์อิวามาโนยด์

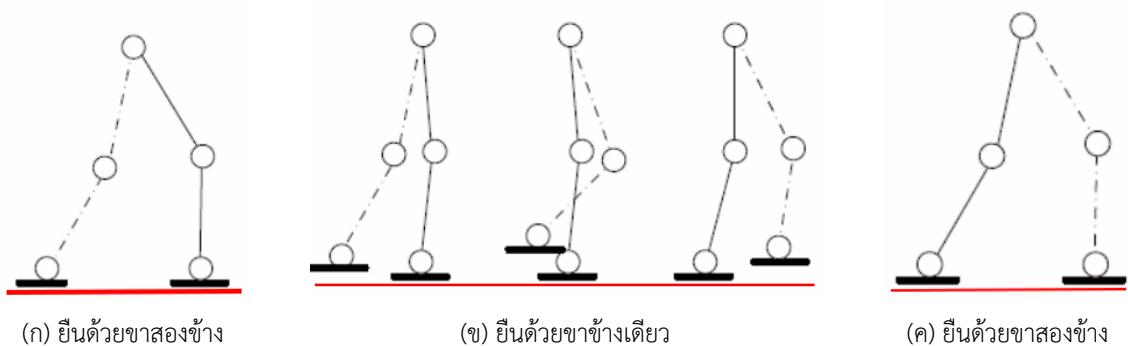


รูปที่ 2.4: ส่วนประกอบของหุ่นยนต์อิวามาโนยด์

หุ่นยนต์อิวามาโนยด์ประกอบด้วยก้านต่อหลายๆ กันที่นำมาต่อ กัน ลักษณะโครงสร้างนั้นจะเป็นแบบโฉม เปิด (Open kinematic chain) และแต่ละก้านต่อจะเชื่อมต่อกันด้วยข้อต่อแบบหมุน เราสามารถแบ่งโครงสร้างของหุ่นยนต์อิวามาโนยด์ออกเป็นส่วนหลักๆ เป็น 2 ส่วน ส่วนแรกคือ ส่วนก้านต่อของลำตัวหุ่นยนต์ (Torso) ซึ่งเราสามารถที่จะรวมไปถึงส่วนแขนกับหัวด้วย และในส่วนที่สองคือ ส่วนก้านต่อของขาหุ่นยนต์ (Legs) ซึ่งเป็นส่วนของหุ่นยนต์ทั้งสองข้างที่สามารถนำไปที่สัมผัสกับพื้นได้ ทั้งสองก้านต่ออนี้ถูกเชื่อมต่อกันด้วยส่วนของสะโพก (Hip) ที่อยู่ระหว่างส่วนลำตัวกับส่วนของขาหุ่นยนต์ ดังรูปที่ ??

#### 2.1.3.2 วัฏจักรการเดินของหุ่นยนต์อิวามาโนยด์

วัฏจักรการเดินของหุ่นยนต์ คือ การที่หุ่นยนต์จะต้องมีการถ่ายน้ำหนักไปมาระหว่างเท้าซ้ายและเท้าขวา มีบางช่วงที่น้ำหนักตกลงบนเท้าข้างใดข้างหนึ่งหรือทั้งสองข้างพร้อมกัน สามารถแบ่งออกเป็นช่วงได้สองช่วง คือ ช่วงการยืนด้วยขาข้างเดียว และช่วงการยืนด้วยขาทั้งสองข้าง



รูปที่ 2.5: วัฏจักรการเดินของหุ่นยนต์อิวามาโนยด์

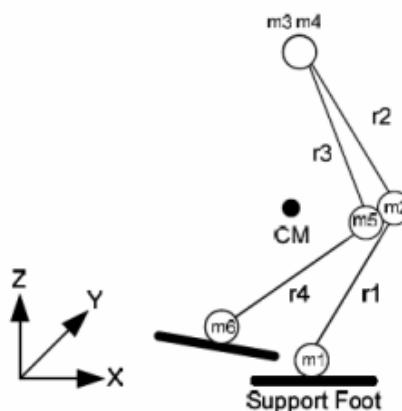
1) การยืนด้วยขาข้างเดียว : เป็นช่วงที่มีเท้าของหุ่นยนต์สัมผัสพื้นเพียงข้างเดียว ส่วนเท้าอีกข้างของหุ่นยนต์จะถูกยกอยู่จากพื้น โดยที่ไม่มีส่วนใดๆของขาข้างนั้นสัมผัสกับพื้นเลย ช่วงนี้จะเกิดขึ้นเมื่อมีการแกะง่ำเท้าจากข้างหลังไปข้างหน้า ดังรูปที่ ??

2) การยืนด้วยขาสองข้าง : เป็นช่วงที่เท้าทั้งสองข้างของหุ่นยนต์สัมผัสกับพื้น ช่วงนี้จะเกิดตั้งแต่หุ่นยนต์วางเท้าขณะที่สันเท้าแตะกับพื้น ไปจนถึง ปลายเท้าของขาอีกข้างหลุดออกจากพื้น

การเดินได้โดยไม่ล้มนั้น ตัวหุ่นยนต์จะต้องรักษาสมดุลของการเดินให้ได้ตลอดช่วงเวลาของการเดิน ซึ่งสมดุลของการเดินแบบสองขาสามารถแบ่งตามลักษณะการเดินและการถ่ายน้ำหนักได้เป็น 2 รูปแบบหลัก คือ การเดินแบบสมดุลสถิต (static balance walking) และ การเดินแบบสมดุลพลวัต (dynamic balance walking)

#### 2.1.3.3 การสร้างและการควบคุมการเดินแบบสมดุลสถิต

การเดินของหุ่นยนต์ในลักษณะนี้ จุดศูนย์กลางมวล (CoM) ของตัวหุ่นยนต์จะไม่มีการเคลื่อนไหวออกนอกบริเวณฐานรับน้ำหนัก (Supporting Area) ตลอดช่วงเวลาการเดิน ไม่ว่าจะเป็นช่วงเวลาที่รับน้ำหนักด้วยเท้าข้างเดียวหรือทั้งสองข้างก็ตาม หมายความว่า โครงสร้างของหุ่นยนต์จะไม่ล้มแน่นอน เนื่องจากการสร้างรูปแบบการเดินด้วยวิธีนี้จะควบคุมให้ตำแหน่งของจุดศูนย์กลางมวล อยู่ภายใต้พื้นที่ฐานรับน้ำหนักของหุ่นยนต์ตลอดเวลา ??



รูปที่ 2.6: การควบคุมตำแหน่งของจุดรวมมวลให้อยู่ในพื้นที่ฐาน

ข้อดีของการสร้างและควบคุมการเดินของหุ่นยนต์ด้วยวิธีนี้คือ สามารถสร้างรูปแบบการเดินได้โดยที่มีความซับซ้อนไม่มากนัก สามารถสั่งให้หุ่นยนต์หยุดค้างในท่าทางใดๆก็ได้ตลอดเวลาโดยหุ่นยนต์ไม่ล้ม หุ่นยนต์ที่มีฝ่าเท้าใหญ่จะทำให้ง่ายต่อการก้าวเดินมากขึ้น นอกจากการควบคุมการก้าวขาแล้วอาจเพิ่มการควบคุมส่วนลำตัวเพิ่มเติม เพื่อเป็นการเพิ่มเสถียรภาพในการเดินและการถ่ายน้ำหนัก โดยที่อาจจะมีการเพิ่มเซนเซอร์วัดแรงที่ฝ่าเท้าเพื่อตรวจสอบการกระจายแรงกดที่ฝ่าเท้า เพื่อตรวจสอบว่าตำแหน่งของจุดรวมน้ำหนักอยู่บนพื้นที่ฝ่าเท้าหรือไม่ หรือเพื่อตรวจสอบเสถียรภาพของการเดินเพื่อแก้ไขท่าทางการเดินไม่ให้เกิดการล้ม

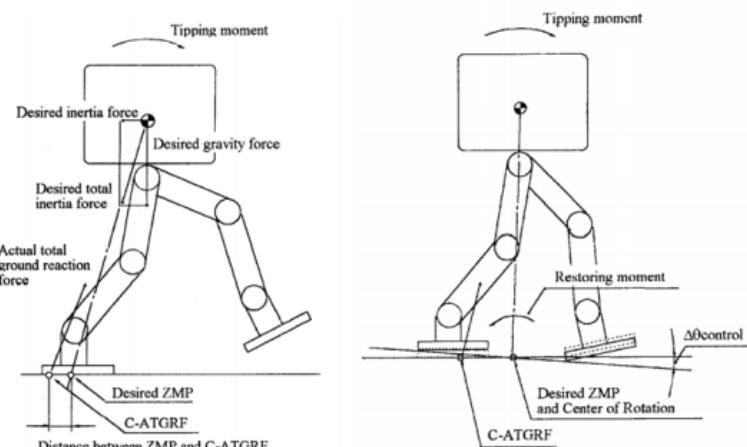
ข้อเสียของการควบคุมการเดินด้วยวิธีนี้คือ หุ่นยนต์จะใช้เวลาในการก้าวเดินมาก ใช้พลังงานในการเดินมากกว่าการเดินแบบสมดุลพลวัต และท่าทางที่ได้จะมีความแตกต่างจากท่าทางการเดินของมนุษย์

#### 2.1.3.4 การสร้างและการควบคุมการเดินแบบสมดุลพลวัต

การสร้างรูปแบบการเดินและควบคุมการเดินในลักษณะนี้ท่าทางการเดินของหุ่นยนต์นั้นจะคล้ายกับการเดินของมนุษย์มากกว่าแบบสถิต เนื่องจากมีหลักการในการสร้างท่าทางที่เหมือนกับการเดินของมนุษย์ซึ่งมีขั้นตอนดังนี้คือ เอียงตัวไว้ให้ล้มไปในทิศทางที่ต้องการเดิน เมื่อเริ่มเกิดการล้มขึ้นหุ่นยนต์จะเปลี่ยนตำแหน่งการวางเท้าไปยังตำแหน่งใหม่ เพื่อปรับให้โครงสร้างเข้าสู่ภาวะสมดุลอีกรั้ง

โดยธรรมชาติแล้วมนุษย์มีการถ่ายน้ำหนักในขณะที่เคลื่อนที่หรือยืนอยู่กับที่เพื่อรักษาสมดุลของท่าทางนั้นไว้ แต่หากการถ่ายโอนน้ำหนักนั้นเกิดสภาวะไม่สมดุล ร่างกายจะปรับสภาพโดยการเคลื่อนตำแหน่งของเท้าซึ่งเป็นพื้นที่ฐานออกจากเดิมไปยังตำแหน่งใหม่ เพื่อรักษาสมดุลไว้ หลักการดังกล่าวถูกนำมาใช้กับการควบคุมการเดินของหุ่นยนต์อิวามานอยด์ ในขณะที่หุ่นยนต์กำลังเคลื่อนไหว ผลจากแรงเฉียวยของการเคลื่อนที่และผลจากแรงดึงดูดของโลกมีผลต่อการเพิ่มและลดความเร่งให้การเดินของหุ่นยนต์ แรงเหล่านี้เรียกว่าแรงเฉียวยของการเคลื่อนที่ และเมื่อเท้าหุ่นยนต์สัมผัสกับพื้นจะได้รับผลกระทบของแรงนี้ เรียกว่า แรงปฏิกิริยาจากพื้น

การตัดกันระหว่างแรงปฏิกิริยาจากพื้นและแนวแรงเฉียวยรวม ตำแหน่งนั้นหากทำให้โมเมนต์เท่ากับศูนย์เรียกจุดตัดนี้ว่าจุดโมเมนต์ศูนย์ ( $ZMP_{robot}$ ) และจุดที่แรงปฏิกิริยาลงสู่พื้นว่า จุดปฏิกิริยาพื้นฐาน ท่าทางการเดินของหุ่นยนต์จะถูกกำหนดและถูกส่งให้กับชุดควบคุมข้อต่อจุดต่างๆ ของหุ่นยนต์ โดยให้สอดคล้องกับแรงเฉียวยรวมที่เกิดขึ้นจากการคำนวณ เรียกว่าแรงเฉียวยรวมเป้าหมาย และจุดโมเมนต์ศูนย์ที่ได้จากการคำนวณเรียกว่าจุดโมเมนต์ศูนย์เป้าหมาย ( $ZMP_{target}$ ) เมื่อหุ่นยนต์เกิดสมดุลในขณะที่ทำการเดินได้อย่างสมบูรณ์ แนวแกนของแรงเฉียวยรวมเป้าหมายและแรงปฏิกิริยาที่พื้นจะเป็นตำแหน่งเดียวกัน แต่ในขณะที่หุ่นยนต์เดินผ่านพื้นผิวที่มีความชรุขระหรือไม่เรียบตำแหน่งสองจุดดังกล่าว จะไม่ใช่ตำแหน่งเดียวกันทำให้หุ่นยนต์เกิดการล้มได้ แรงที่ทำให้เกิดการล้มนี้เกิดจากตำแหน่งของจุดโมเมนต์ศูนย์และตำแหน่งแรงปฏิกิริยารวมที่พื้นไม่ตรงกัน ซึ่งเป็นสาเหตุหลักที่ทำให้เกิดความไม่สมดุลขึ้น และเมื่อหุ่นยนต์เสียสมดุลระบบที่จะสามารถป้องกันการล้มและทำให้หุ่นยนต์เดินต่อไปได้อย่างต่อเนื่องคือ ระบบควบคุมแรงปฏิกิริยา ระบบควบคุมจุดโมเมนต์ศูนย์ และระบบควบคุมการวางแผนเท้า??



รูปที่ 2.7: การควบคุมตำแหน่งของจุดโมเมนต์ศูนย์ให้ตรงกับแรงปฏิกิริยารวม

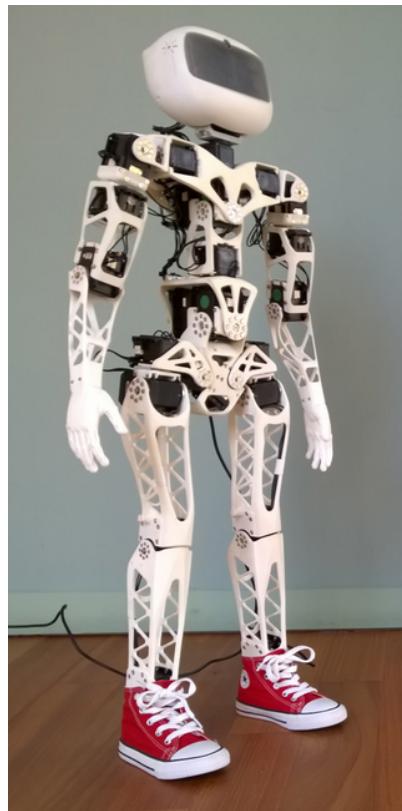
อย่างไรก็ตาม การสร้างท่าทางการเดินในลักษณะนี้ต้องใช้สมการในการคำนวณที่ซับซ้อนมาก เนื่องจากต้องหาความสัมพันธ์ระหว่างองค์ประกอบหลายส่วน เช่น น้ำหนักของโครงสร้างในแต่ละส่วน แรงบิดที่แต่ละข้อต่อ และโมเมนต์โดยรวมของระบบ นอกจากนี้ยังต้องใช้อุปกรณ์การตรวจวัดต่างๆ เช่น เชเซอร์วัดแรง เชเซอร์วัดมุม เชเซอร์วัดแรงบิด ติดตั้งตามจุดต่างๆ ของโครงสร้างเพื่อวัดค่าอุกมา ก่อนที่จะทำการคำนวณตำแหน่ง และสร้างท่าทางการเดินของหุ่นยนต์อิวามานอยด์ ท่าทางการเดินที่ได้จากการควบคุมด้วยวิธีนี้ จะมีความคล้ายคลึงกับท่าทางการเดินของมนุษย์มาก

### 2.1.3.5 จุดศูนย์กลางมวลของหุ่นยนต์

หากต้องการให้หุ่นยนต์สามารถที่จะทรงตัวอยู่ได้โดยไม่ล้มนั้น จึงต้องรู้ตำแหน่งจุดศูนย์กลางมวลของหุ่นยนต์ตลอดเวลา และต้องให้จุดศูนย์กลางมวลอยู่ติดในบริเวณฐานรับน้ำหนักของหุ่นยนต์โดยหากพื้นที่ที่ฝ่าเท้าสัมผัสกับพื้น วิธีการนี้เป็นวิธีการทางสถิตศาสตร์

#### 2.1.4 ตัวอย่างหุ่นยนต์ชีวามโนยด์

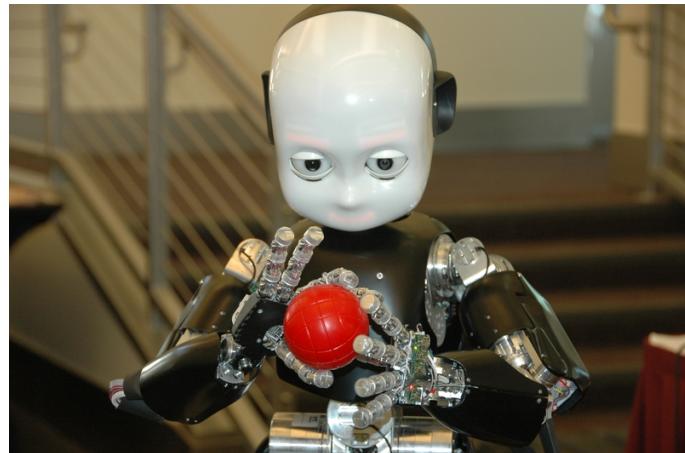
Poppy Humanoid



รูปที่ 2.8: หุ่นยนต์ชีวามโนยด์ปีอปปี้

หุ่นยนต์ชีวามโนยด์ปีอปปี้ ถูกสร้างขึ้นมาเพื่อใช้ในงานศิลปะ การวิจัยและการศึกษาโดยเฉพาะ หุ่นยนต์ปีอปปี้ประกอบด้วยส่วนของฮาร์ดแวร์และซอฟแวร์ที่เปิดเป็นโอเพนซอร์ซให้ผู้ที่สนใจสามารถเข้ามาศึกษาได้ โปรแกรมของหุ่นยนต์ใช้โมดูลที่มีชื่อว่า Pypot ที่เป็นส่วนเสริมของภาษา Python ในการพัฒนาซอฟแวร์ ทุกคนสามารถเข้าถึงข้อมูลเชิงเทคนิคของหุ่นยนต์ชีวามโนยด์ปีอปปี้ได้ เช่น ส่วนรายละเอียดการทำงาน คลิปวีดีโอสอน การประกอบ การใช้ระบบจำลอง และการพัฒนาต่างๆผ่านทางเว็บไซต์ <http://www.poppy-project.org> หุ่นยนต์ปีอปปี้มีส่วนของโครงสร้างที่ผลิตมาจากพลาสติก PLA และ ABS โดยใช้เทคนิคการฉีดรูปด้วยเครื่องพิมพ์สามมิติ ตัวขับเคลื่อนข้อต่อต่างๆใช้เป็น Dynamixel Digital Servo และควบคุมคำสั่งของตัวขับเคลื่อนด้วยคอมพิวเตอร์ขนาดเล็ก Odroid UX4 ใช้ระบบปฏิบัติการ Ubuntu 14.04 ตัวของหุ่นยนต์มีความสูง 83 เซนติเมตร น้ำหนัก 3.5 กิโลกรัม ใช้เซนเซอร์วัดมุมอิเล็กทรอนิกส์ IMU ที่มีองศาอิสระเท่ากับ 9 องศาอิสระ ในการควบคุมเส้นสายภาพในการเดินของตัวเอง มีองศาอิสระหรือจำนวนตัวขับเคลื่อนทั้งหมด 25 องศา ประกอบไปด้วย ขาข้างละ 6 องศาอิสระ แขนข้างละ 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 สำหรับส่วนที่ต้องรับภาระความเครียดน้อย ใช้ aluminum alloy 7075 (Ergal) สำหรับส่วนที่ต้องรับภาระความเครียดปานกลางถึงสูง และใช้ Stainless Steel 17-4PH ในส่วนของเพลาข้อต่อต่างๆ เพื่อให้มีความแข็งแรงสูง ตัวหุ่นยนต์ถูกออกแบบให้มีลักษณะเหมือนเด็กอายุ 3-4 ขวบ ควบคุมโดยใช้บอร์ดไมโครคอนโทรเลอร์เป็นรุ่น PC104 Controller ภาษาที่ใช้ในการพัฒนาใช้เป็นภาษา C++ ในการเขียนโปรแกรม การติดต่อสื่อสารกับตัวขับเคลื่อนหรือมอเตอร์ตามข้อต่อต่างๆ และเซนเซอร์ ผ่านทางโปรโตคอล CAN Bus เพื่อทำให้ใช้งานง่ายลง ใช้เส้นเอ็นในการส่งถ่ายแรงขับเคลื่อนไปยังส่วนของข้อต่อส่วนมือและขา นิ้วของหุ่นยนต์ถูกร้อยด้วยสายเคเบิลเคลือบ Teflon อยู่ภายใต้ และความต้านทานของสายเคเบิลที่ต้องการ ใช้เซนเซอร์วัดมุมของข้อต่อแต่ละตัวใช้การออกแบบให้มี Hall-effect ติดอยู่ ช่วยในการอ่านค่าของตำแหน่งและความเร็วที่เกิดขึ้นที่ข้อต่อนั้น หุ่นยนต์ไอคัพมีองศาอิสระรวมกันทั้งหมด 53 องศาอิสระ ประกอบไปด้วย แขนข้างละ 7 องศาอิสระ มือข้างละ 9 องศาสือร์ หัว 6 องศาสือร์ ลำตัว 3 องศาสือร์ และขาข้างละ 6 องศาสือร์ ในส่วนของหัวจะประกอบไปด้วย กล้องสองตัวเพื่อทำการสืบสานและคิ้ว หุ่นยนต์ไม่ได้ถูกออกแบบให้มีการทำงานเป็นแบบอัตโนมัติ ซึ่งก็คือตัวหุ่นยนต์นั้นไม่มีแบตเตอรี่ภายในตัว แต่ใช้แหล่งพลังงานจากการส่งเข้าไปผ่านสายเคเบิล และเชื่อมต่อกับอินเทอร์เน็ตผ่านสายแลน (LAN) ??

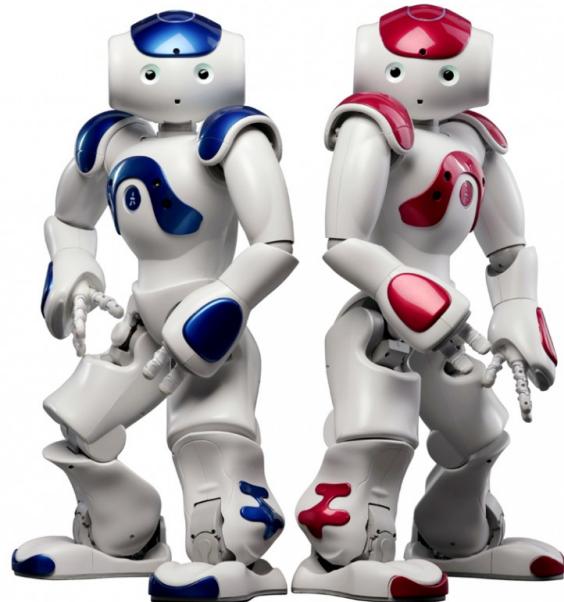
### Darwin-OP Humanoid



รูปที่ 2.10: หุ่นยนต์อิวามาโนย์ดาร์วิน

หุ่นยนต์อิวามาโนย์ดาร์วิน (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 หุ่นยนต์อิวามาโนย์ดาร์วินมีความสามารถในการรับภาระโหลดได้สูง 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 ??

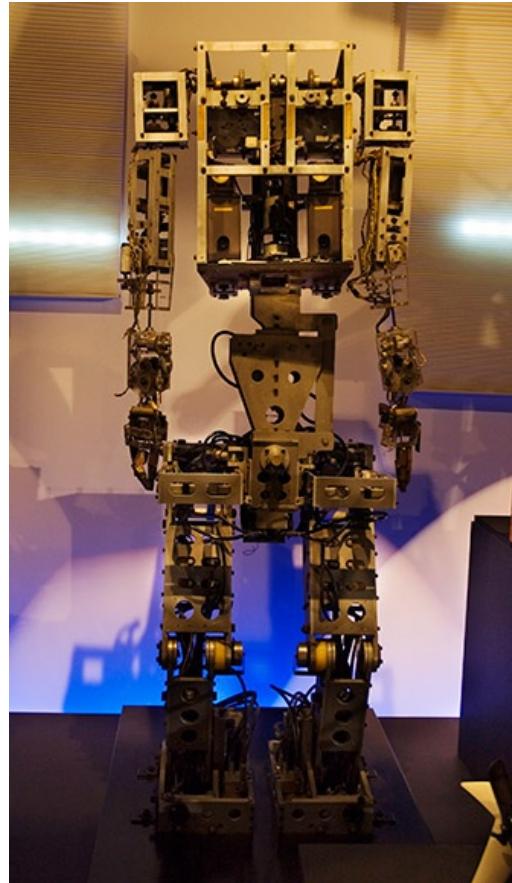
### Nao Humanoid



รูปที่ 2.11: หุ่นยนต์อิวามานอยด์นาโอะ

หุ่นยนต์อิวามานอยด์นาโอะ เป็นหุ่นยนต์อิวามานอยด์ขนาดกลาง ถูกผลิตมาจากประเทศฝรั่งเศส พัฒนาโดยบริษัท Aldebaran Robotics เมื่อปี 2004 และในปี 2007 หุ่นยนต์อิวามานอยด์นาโอะได้นำไปแทนที่หุ่นยนต์สูนัขของ Sony ซึ่ง Aibo ขณะนั้นใช้ในการแข่งขัน RoboCup Standard Platform League (SPL) หุ่นยนต์นาโอะได้ถูกนำไปใช้ใน Robocup 2008 และ 2009 หุ่นยนต์นาโอะถูกพัฒนาออกแบบมาหลายรุ่น มีองศาอิสระตั้งแต่ 14 องศาอิสระ 21 องศาอิสระ และ 25 องศาอิสระ สำหรับเพื่องานวิจัยนั้นมีถึง 25 องศาอิสระ โดยเพิ่มเติมมือสองข้างเอวเข้าไปเพื่อให้สามารถยกจับสิ่งของได้ ภายในหุ่นยนต์ถูกควบคุมด้วยระบบปฏิบัติการ NAO 2.0 (Linux-based) ตัวหุ่นยนต์มีความสูง 58 เซนติเมตร น้ำหนัก 4.3 กิโลกรัม ส่วนเซนเซอร์การรับรู้ต่างๆ จะประกอบไปด้วยเซนเซอร์วัดมุมอุ่น 3-axis gyro, 3-axis accelerometer, Ultrasound captors, ไมโครโฟน 4 ตัว ลำโพง 2 ตัว กล้อง 2 ตัว เพื่อใช้ประโยชน์ในการทำงานวิจัยต่างๆ ตอนนี้ความสามารถของหุ่นยนต์นาโอะที่ทำได้คือ สามารถเห็นสีได้ เดินขึ้นลงบันไดและทางลาดชันได้ ระหว่างการเดินนั้นสามารถวางแผนการวางเท้าได้อย่างรวดเร็ว อีกทั้งยังสามารถที่จะเดินหลบหลีกสิ่งกีดขวางได้ด้วย ??

## 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 นั้นสามารถเดินได้ด้วยขาของตนเองที่มีสองข้าง สามารถหยิบและเคลื่อนย้ายวัสดุด้วยมือ ต่อมาในปี 1984 มหาวิทยาลัย Waseda ได้พัฒนาหุ่นยนต์อิวามานอยด์ที่ชื่อ Wabot-2 โดยหุ่นยนต์สามารถสื่อสารกับมนุษย์ได้ สามารถถ่ายโน้ตเพลงและเล่นดนตรีโดยใช้ electronic organ แบบง่ายๆ ได้ และในปี 1985 บริษัท Hitachi ได้สร้างหุ่นยนต์ WHL-11 ที่มีสองขาเหมือนมนุษย์ ซึ่งสามารถเดินแบบสมดุลสถิต (Static Walking) บนพื้นราบได้ด้วยความเร็ว 13 วินาทีต่อหนึ่งก้าว และสามารถเลี้ยวได้ซ้ายและขวาได้ ??

## 2.2 การออกแบบโครงสร้างของหุ่นยนต์

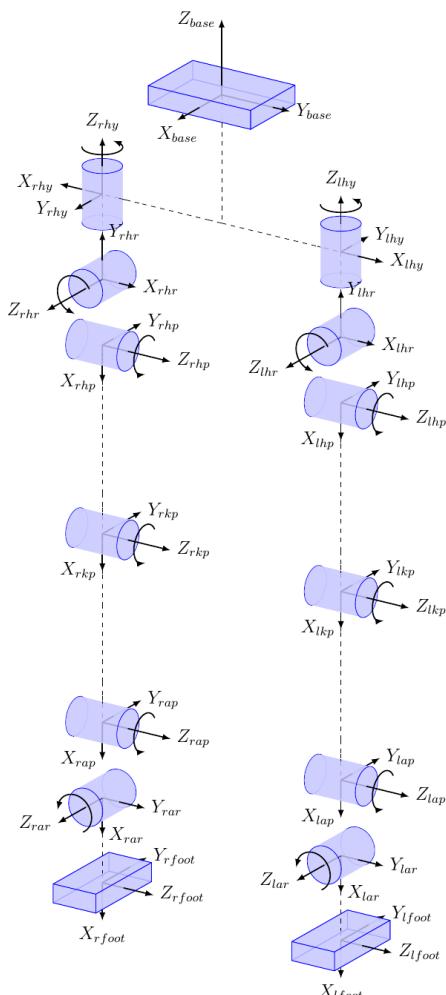
### 2.2.1 ความแตกต่างระหว่างโครงสร้างของมนุษย์กับโครงสร้างของหุ่นยนต์

#### 2.2.1.1 ความแตกต่างขององค์การเสรี

เนื่องจากลักษณะข้อต่อของมนุษย์มีความซับซ้อนมากกว่าโครงสร้างของหุ่นยนต์ ทำให้ข้อต่อแต่ละจุดของมนุษย์นั้นสามารถหมุนได้หลายทิศทาง รวมถึงขอบเขตของการหมุนของข้อต่อในแต่จุดก็มีความแตกต่างกัน ใน การนำรูปแบบการเดินของมนุษย์ไปใช้กับหุ่นยนต์จึงต้องปรับค่ามุมที่ข้อต่อให้มีความเหมาะสมกับโครงสร้าง และ ข้อจำกัดเกี่ยวกับการหมุนของข้อต่อจุดต่างๆ ของหุ่นยนต์ที่จะใช้ทดสอบด้วย

#### 2.2.1.2 ความแตกต่างของอัตราส่วน

นอกจากความแตกต่างขององค์การเสรี (DoF) ระหว่างมนุษย์กับหุ่นยนต์แล้ว ความแตกต่างของอัตราส่วนระหว่างโครงสร้างแต่ละส่วนของมนุษย์กับหุ่นยนต์เป็นอีกสาเหตุหนึ่ง ที่ต้องทำการปรับแต่งใหม่มีความเหมาะสม เนื่องจากความยาวของโครงสร้างแต่ละส่วน รวมทั้งระยะห่างระหว่างจุดหมุนแต่ละจุดของมนุษย์กับหุ่นยนต์ที่มีความแตกต่างกัน ดังนั้นจึงต้องกำหนดระบบพิกัดสำหรับหุ่นยนต์ที่มีความยาวต่างกัน เช่น ขาและศีรษะ ที่มีความยาวต่างกัน แต่ทั้งสอง部分มีความกว้างและสูงที่ใกล้เคียงกัน เพื่อให้ในการอ้างอิงจุดหมุน และความยาวของโครงสร้างในส่วนต่างๆ



รูปที่ 2.13: ตัวอย่างตำแหน่งและการหมุนของข้อต่อของหุ่นยนต์เพื่อการอ้างอิง

### 2.2.1.3 กำลังและประสิทธิภาพของมอเตอร์

ความสามารถในการรับน้ำหนักของข้อต่อแต่ละจุดมีความแตกต่างกัน การเคลื่อนไหวของมนุษย์นั้นจะมีกล้ามเนื้อ และเลี้นเอ็นเป็นตัวออกแรงดึงส่วนต่างๆของร่างกายเพื่อทำให้เกิดการเคลื่อนไหวซึ่งจะมีความยืดหยุ่นและแรงดึงที่มีค่าสูง สำหรับการเคลื่อนไหวของหุ่นยนต์ จะใช้การบิดแกนของเซอร์โวมอเตอร์ (Servo Motor) หรือมอเตอร์ที่ติดอยู่ที่ข้อต่อจุดต่างๆ ทำให้ความสามารถในการรับน้ำหนัก แรงบิดและความยืดหยุ่นที่ข้อต่อขึ้นกับกำลังของมอเตอร์เป็นหลัก การสร้างท่าทางของหุ่นยนต์จึงต้องคำนึงถึงความสามารถในการรับน้ำหนักและกำลังของเซอร์โวมอเตอร์ที่ใช้ด้วยเข่นกัน

### 2.2.2 อุปกรณ์ที่ใช้ในหุ่นยนต์ชีวามโนยด์

#### ตัวขับเคลื่อน

ในการสร้างหุ่นยนต์ชีวามโนยด์นั้นระบบการขับเคลื่อนถือว่าเป็นเรื่องสำคัญ เนื่องจากว่าถ้าหากระบบขับเคลื่อนไม่สามารถทำงานได้อย่างเต็มประสิทธิภาพ หรือหากมีการอุบัติเหตุที่ผิดพลาด จะส่งผลทำให้หุ่นยนต์ชีวามโนยด์นั้นมีประสิทธิภาพในการทำงานลดลงตามไปด้วย ภายในงานวิจัยนี้ทางผู้จัดทำได้ใช้ตัวขับเคลื่อนเป็น Dynamixel digital servo EX-106 ซึ่งเป็นเซอร์โวมอเตอร์ที่เหมาะสมสำหรับทำหุ่นยนต์โดยเฉพาะ ภายในประกอบไปด้วย มอเตอร์กระแสตรง ชุดเฟืองมอเตอร์ ไดเรอර์คอนโทรลเลอร์ สามารถเชื่อมต่อกันผ่าน BUS RS-485 มีการควบคุมแบบ PID และแรงบิดที่สูง<sup>8</sup>



รูปที่ 2.14: ตัวขับเคลื่อนที่ใช้ในหุ่นยนต์ชีวามโนยด์

<sup>8</sup>Robot Actuator [[http://support.robotis.com/en/product/actuator/dynamixel/ex\\_series/ex-106.htm](http://support.robotis.com/en/product/actuator/dynamixel/ex_series/ex-106.htm)]

## หน่วยประมวลผลควบคุม

ในการควบคุมหุ่นยนต์ชีวามโนยด้วยความสามารถทำกิจกรรมต่างๆ คือ หน่วยประมวลผลระบบควบคุม ถ้าหากไม่มีระบบประมวลผลควบคุมแล้ว อุปกรณ์ต่างๆ ที่ติดตั้งอยู่ภายในตัวของหุ่นยนต์ชีวามโนยด้วยจะไม่สามารถติดต่อสื่อสารกันได้ ซอฟท์แวร์ของหุ่นยนต์ที่พัฒนามาทั้งหมดจะไม่สามารถใช้ได้ ทำให้หุ่นยนต์ชีวามโนยด้วยไม่สามารถทำงานในสิ่งที่ต้องการ การวางแผนระบบควบคุมที่นิยมใช้ในระบบหุ่นยนต์ชีวามโนยด้วยส่วนใหญ่ จะแบ่งออกเป็น 2 ส่วนด้วยกัน คือส่วนของหน่วยประมวลผลควบคุมระดับสูง และหน่วยประมวลผลควบคุมระดับต่ำ

### หน่วยประมวลผลควบคุมระดับสูง (High level controller)

หน่วยประมวลผลควบคุมระดับสูงเป็นส่วนที่ใช้ประมวลผลการทำงานที่มีความซับซ้อนของระบบ เช่น จลนศาสตร์ของหุ่นยนต์ การคำนวณหาเส้นทางการเดิน ในการคำนวณทางคณิตศาสตร์ของระบบเหล่านี้จำเป็นต้องมีการประมวลผลที่เร็ว และมีประสิทธิภาพ ในสมัยที่มีการพัฒนาหุ่นยนต์ชีวามโนยด้วยคุ้นเคยเริ่มนั้น หน่วยประมวลผลควบคุมระดับสูง จะใช้คอมพิวเตอร์เป็นตัวในการประมวลผลการคำนวณ ซึ่งคอมพิวเตอร์สมัยนั้นมีขนาดใหญ่ น้ำหนักมาก และต้องใช้พลังงานสูง ซึ่งต่างจากปัจจุบันนี้ที่มีการพัฒนาของเทคโนโลยีที่ก้าวหน้ามากขึ้น ทำให้คอมพิวเตอร์มีขนาดเล็กลงเทียบเท่ากับบอร์ดคอนโทรลเลอร์ทั่วไป<sup>9</sup>



รูปที่ 2.15: ตัวประมวลผลระดับสูงของ Thormang Humanoid

### หน่วยประมวลผลควบคุมระดับต่ำ (Low level controller)

หน่วยประมวลผลควบคุมระดับต่ำเป็นส่วนที่รับคำสั่งมาจากหน่วยประมวลผลควบคุมระดับสูง มีประสิทธิภาพในการประมวลผลการคำนวณที่น้อยกว่า เนื่องจากการออกแบบสถาปัตยกรรมภายในระบบไม่เอื้ออำนวยต่อการคำนวณที่มีความซับซ้อน แต่มีความสามารถในการประมวลผลระบบที่เป็นควบคู่ด้วยแม่นยำ ในด้านการทำหุ่นยนต์ชีวามโนยดันนั้นมากจะใช้หน่วยประมวลผลควบคุมระดับต่ำ ในการติดต่อกับอุปกรณ์ต่างๆ บนตัวของหุ่นยนต์ชีวามโนยด้วยตรง เช่น ตัวขับเคลื่อน เชนเชอร์รับค่า หรือไฟแสดงสถานะต่างๆ ของหุ่นยนต์



รูปที่ 2.16: ตัวประมวลผลระดับต่ำของ Robotis OP3 Humanoid

<sup>9</sup>Robot controller Robotis, Thormang, [http://jp.robotis.com/index/product.php?cate\\_code=111410](http://jp.robotis.com/index/product.php?cate_code=111410)

## เซนเซอร์ตรวจหน้าสัมผัสที่พื้น

เซนเซอร์ตรวจหน้าสัมผัสที่พื้นเป็นเซนเซอร์ที่ถูกติดตั้งบริเวณฝ่าเท้า เพื่อตรวจสอบการเดินของหุ่นยนต์อิวามานอยด์ว่าขณะนี้มีการสัมผัสของฝ่าเท้าของหุ่นยนต์กับพื้นหรือไม่ ซึ่งในงานวิจัยนี้ได้ใช้หลักการตัวตรวจจับแรงกดแบบค่าความต้านทานหรือ Force Sensing Resistor (FSR)<sup>10</sup> ที่ใช้เทคโนโลยีฟิล์มโพลีเมอร์แบบหนาโดยที่เซนเซอร์สามารถเปลี่ยนแรงที่มีกระทำให้อยู่ในรูปของการเปลี่ยนแปลงค่าความต้านทานไฟฟ้า ตัวเซนเซอร์มีลักษณะเป็นแผ่น มีโครงสร้าง 5 ชั้น โดยสองชั้นนอกสุดเป็นฟิล์มของโพลีเอสเตอร์ ส่วนชั้นถัดเข้ามาเป็นฟิล์มของโลหะที่เป็นตัวนำไฟฟ้า และชั้นในสุดเป็นหมึกที่มีความไวในการตอบสนองต่อแรงภายนอกที่มีกระทำ (Pressure sensitive ink) และโครงสร้างทั้ง 5 ชั้น ถูกรูmor เข้าด้วยกันด้วยวิธีลามิเนท จึงทำให้เซนเซอร์วัดแรงนี้มีลักษณะแบบมีความยืดหยุ่นสูง ด้วยเหตุนี้จึงทำให้เซนเซอร์สามารถโค้งงอได้ง่าย แรงดันไฟฟ้าที่ต่อกลับจะลดลง เมื่อมีแรงกดมากระทำบนแผ่นตรวจจับ มีโครงสร้างของตัวตรวจจับแสดงในรูปที่ ??



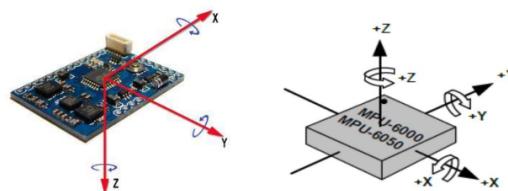
รูปที่ 2.17: ลักษณะโครงสร้างของตัวตรวจจับแรงกด FSR

## เซนเซอร์วัดความเร็ว

Inertial Measurement Unit (IMU) เป็นส่วนประกอบหลักที่ใช้ในการนำร่องเครื่องบิน ยาน-อวกาศ ดาวเทียม เรือ ขีปนาวุธ ซึ่งในตัวของ IMU ประกอบไปด้วยสองส่วนหลักคือ Accelerometers 3 ทิศทาง ใน การรับความเร่งเชิงเส้น และ Gyroscopes 3 ทิศทาง ในการบอกความเร็วเชิงมุม เซนเซอร์ตัวนี้สามารถนำมาใช้ในการทิศทางการหมุนของตัวหุ่นยนต์อิวามานอยด์ได้

เซนเซอร์วัดความเร็ว (Gyroscope)<sup>11</sup> เป็นอุปกรณ์สำหรับการวัดความเร็ว หรือการรักษาการปรับทิศทาง ขึ้นอยู่กับหลักการของการอนุรักษ์โมเมนตัมเชิงมุม ถ้าไม่มีการเคลื่อนที่ อัตราการเปลี่ยนแปลงมุมจะมีค่าเท่ากับศูนย์

เซนเซอร์วัดความเร่ง (Accelerometer)<sup>12</sup> เป็นอุปกรณ์ที่ใช้วัดความเร่งเชิงเส้น โดยอาศัยการวัดแรงที่กระทำต่อน้ำหนัก อ้างอิงที่เกิดจากแรงโน้มถ่วงโลก ซึ่งแรงโน้มถ่วงของโลกจะเป็นเวกเตอร์ซึ่งไปที่แกนกลางโลกเสมอ ตามกฎของนิวตัน



รูปที่ 2.18: เซนเซอร์วัดความเร็ว

<sup>10</sup>[UNICON] Force sensor with UNICON [http://doc.inex.co.th/force-sensor-with-unicon/]

<sup>11</sup>Mechanic gyroscope two-degree of freedom [https://www.bosch-sensortec.com/bst/products/motion/gyro-scope/overview\_gyrosopesensors]

<sup>12</sup>Accelerometer and Gyroscopes Sensor [https://www.maximintegrated.com/en/app-notes/index.mvp/id/5830]

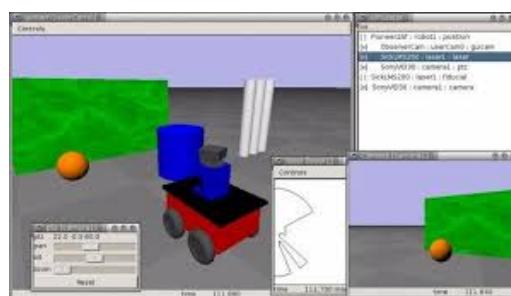
## 2.3 การออกแบบโปรแกรมด้วย ROS

### 2.3.1 ระบบที่ใช้ช่วยในการพัฒนาหุ่นยนต์

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

ปัจจุบันมี Robot Middleware ที่ถูกพัฒนาขึ้นมาให้ใช้อยู่หลายตัว เช่น

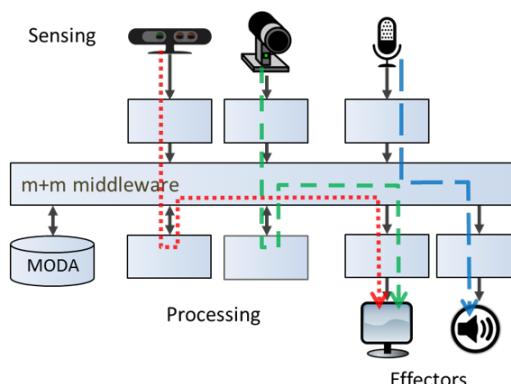
Player Project



รูปที่ 2.19: player project middleware

เป็นโปรเจกต์ที่ใช้ในการสร้างซอฟแวร์เพื่อการศึกษาวิจัยที่มีความเกี่ยวข้องกับหุ่นยนต์และระบบเชนเชอร์ ภายในประกอบไปด้วยระบบตัวกลาง และระบบจำลองการทำงานของหุ่นยนต์

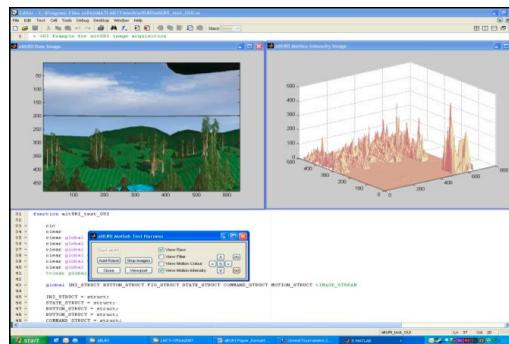
YARP



รูปที่ 2.20: yarp middleware

เป็น open source ที่เขียนด้วยภาษา C++ ในการเข้ามาร่วมต่อกับเชนเชอร์ หน่วยประมวลผล และตัวขับเคลื่อนของหุ่นยนต์

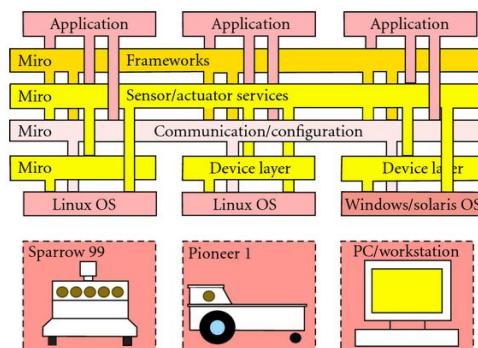
## URBI



รูปที่ 2.21: urbi middleware

เป็น open source สำหรับพัฒนาแอพพลิเคชันที่เกี่ยวข้องกับหุ่นยนต์หรือระบบที่มีความซับซ้อน ใช้ภาษาพื้นฐานเป็นภาษา C++ ติดต่อสื่อสารได้ภายในเครือข่ายเดียวกันเท่านั้น (Local Network)

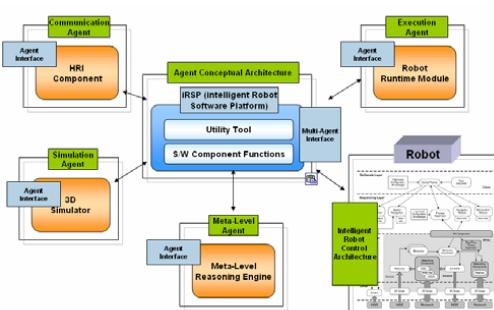
## MIRO



รูปที่ 2.22: miro middleware

เป็นกรอบการทำงานของหุ่นยนต์ที่เคลื่อนที่โดยใช้ในลักษณะเป็น OOP

## OpenRDK

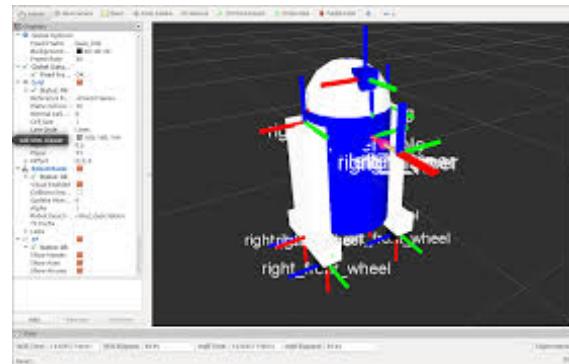


รูปที่ 2.23: openrdk middleware

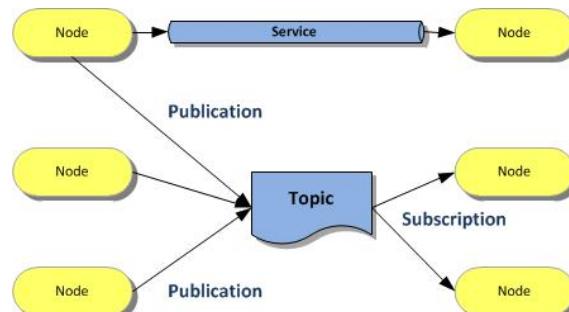
เป็น open source สำหรับพัฒนาระบบที่มีความเป็นอิสระต่อกัน (Modules) สามารถใช้ช่องทางการติดต่อสื่อสารและหน่วยความจำร่วมกันได้

## Robot Operating System

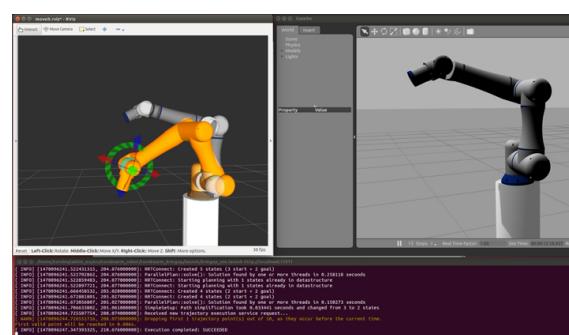
Robot Operating System หรือ ROS ถูกพัฒนาโดยบริษัท Willow Garage, แต่เดิมแล้ว ROS ถูกพัฒนาเพื่อใช้งานกับหุ่นยนต์ PR2 ในปี 2007 ซึ่งพัฒนาเป็น Open Source framework สำหรับนักพัฒนาซอฟแวร์ที่เกี่ยวข้อง กับหุ่นยนต์ มีความสามารถในการทำงานแบบ parallel บนคอมพิวเตอร์หลายเครื่องได้ สามารถทำงานได้ทั้งหลาย OS นอกจากรถยนต์ยังมีคลังที่ค่อยเก็บซอฟแวร์ต่างๆไว้เป็น libraries อีกด้วย การใช้ ROS จะช่วยทำให้เราสามารถพัฒนาหุ่นยนต์ได้อย่างรวดเร็วมากขึ้น ประหยัดเวลา ประหยัดทรัพยากร



รูปที่ 2.24: ROS middleware Rviz



รูปที่ 2.25: ROS algitecture



รูปที่ 2.26: ROS Moveit

### 2.3.2 ระบบที่ใช้ในการจำลองการทำงานของหุ่นยนต์

โปรแกรมจำลองการทำงานของหุ่นยนต์นั้นเป็นเครื่องมือที่สำคัญสำหรับนักวิจัยที่ทำงานวิจัยเกี่ยวกับหุ่นยนต์ การใช้โปรแกรมจำลองนั้นจะช่วยเพิ่มประสิทธิภาพในการทำงานหลายอย่าง เช่น ให้รู้ว่าหุ่นยนต์ที่ออกแบบนั้นสามารถทำงานได้อย่างที่ต้องการหรือไม่ กระบวนการคิดถูกต้องหรือไม่ โปรแกรมจำลองระบบส่วนใหญ่จะคำนวณพลวัตของหุ่นยนต์โดยใช้เครื่องมือคำนวณ Open Dynamics Engine (ODE)

#### USARSim



รูปที่ 2.27: ผลลัพธ์จากการใช้โปรแกรม USARSim

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

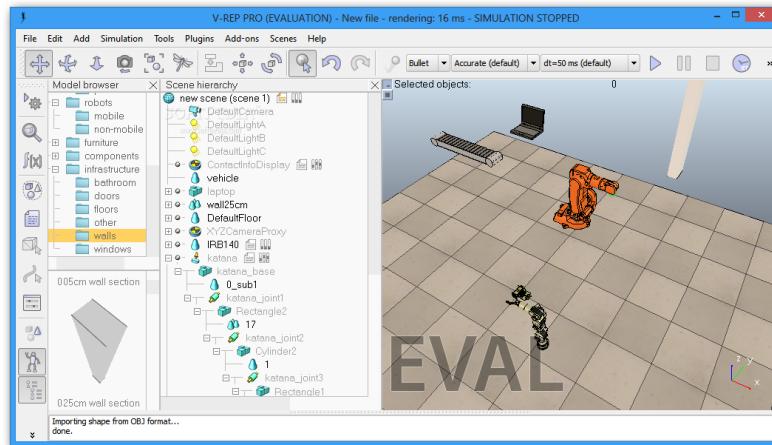
#### MuRoSimF



รูปที่ 2.28: ผลลัพธ์จากการใช้โปรแกรม MuRoSimF

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

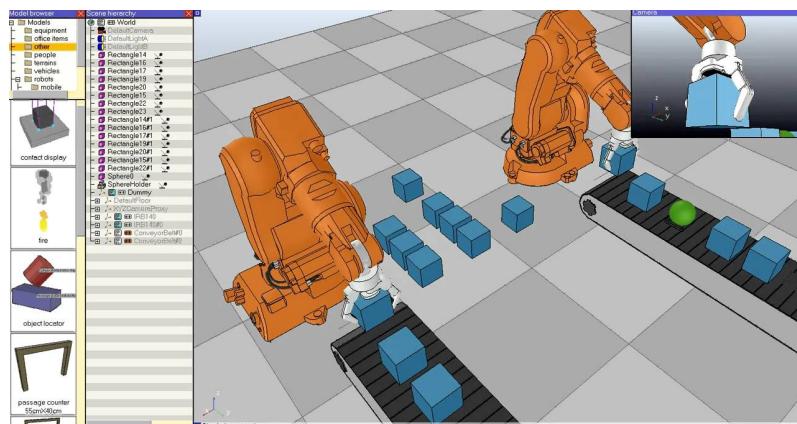
## V-Rep



รูปที่ 2.29: ผลลัพธ์จากการใช้โปรแกรม V-REP

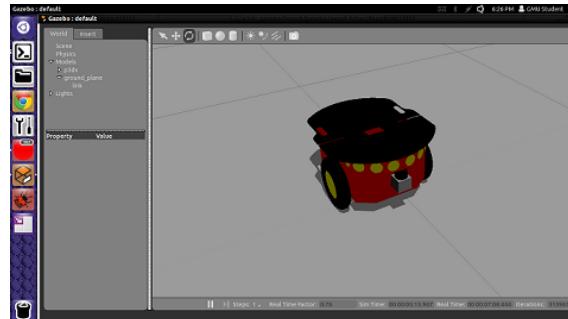
VREP เป็นเครื่องมือสำหรับใช้ในการจำลองระบบหุ่นยนต์ ที่กำลังได้รับความนิยม โดยมีการเพิ่มระบบควบคุมผ่านโปรแกรมจากภายนอกเข้าไป สามารถที่จะเชื่อมต่อกับ ROS ได้และยังสามารถที่จะเขียน Script เพื่อควบคุมหุ่นยนต์ผ่าน API ได้

VREP มีความเร็วในการประมวลผลที่สูง จำลองสายการผลิตในโรงงานอุตสาหกรรม และเหมาะสมสำหรับหุ่นยนต์ที่มีการเคลื่อนที่ด้วยขา มีเวอร์ชัน Education ที่เป็น Open source แต่หากจะใช้เพื่อการค้าจำเป็นจะต้องมี license เพื่อที่จะใช้งาน ราคาก่อนข้างสูง

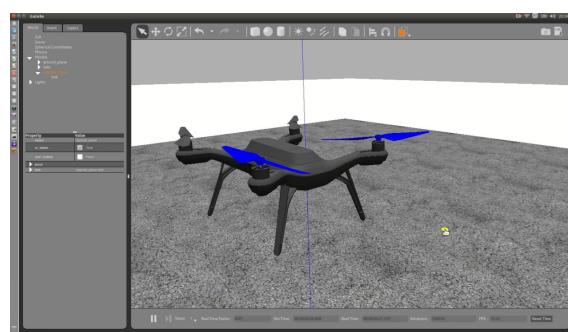


รูปที่ 2.30: V-REP จำลองสายการผลิต

## Gazebo



รูปที่ 2.31: Mobile robot with Gazebo



รูปที่ 2.32: Quadrotor with Gazebo

Gazebo เป็นโปรแกรมจำลองการทำงานของหุ่นยนต์ ที่มีความสามารถในการคำนวณการเดินและการเคลื่อนที่ของหุ่นยนต์ที่ слับซับซ้อนได้ สามารถเห็นภาพกราฟิกของหุ่นยนต์ขณะทำงาน โดยผู้ใช้สามารถกำหนดค่าตัวแปรทางฟิสิกส์ต่าง ๆได้ เช่น น้ำหนัก ค่าความเรื้อย แรงเสียดทานของข้อต่อ ทำให้การออกแบบหุ่นยนต์หรือทดลองโปรแกรมได้เหมือนกับโลกจริง มีแสง มีเงา และ พื้นผิวของวัตถุ และที่พิเศษคือสามารถสังเคราะห์ค่าของเซนเซอร์ เช่นเซอร์พร้อมสัญญาณรบกวน ค่าระยะทาง แรงบิด และอื่นๆ คำนวณพลศาสตร์ของหุ่นยนต์โดยใช้ตัวคำนวณทางฟิสิกส์เป็น Bullet หรือ Simbody ในการจำลองหุ่นยนต์ในโปรแกรมนี้จำเป็นต้องได้รับไฟล์ข้อมูลของหุ่นยนต์มาก่อนซึ่งอยู่ในรูปแบบของ URDF ซึ่ง URDF คือ ประเภทของไฟล์ที่บ่งบอกถึงความสัมพันธ์ของข้อต่อและก้านต่อแต่ละชิ้นในตัวหุ่นยนต์ มีความสามารถในการอธิบายถึงกลศาสตร์และการเคลื่อนที่ของหุ่นยนต์ รวมถึงตรวจสอบการกระแทกกันของก้านต่อในหุ่นยนต์ได้ ภายในไฟล์นี้จะประกอบไปด้วย

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

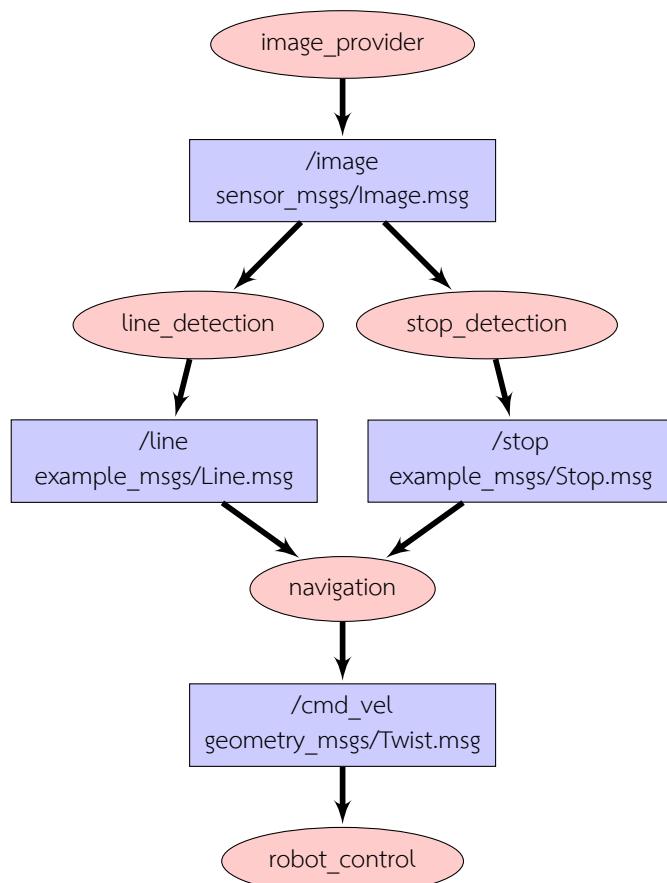
**Joint :** คือข้อต่อของหุ่นยนต์สามารถกำหนดกลศาสตร์และการเคลื่อนที่ได้ เช่น Joint limits ของข้อต่อที่กำลังหมุนและความเร็วการหมุน ซึ่งข้อต่อมีหลายแบบที่สามารถกำหนดได้ เช่น ข้อต่อแบบหมุน, ข้อต่อแบบเลื่อน, ข้อต่อแบบบิดติด, ข้อต่อแบบต่อเนื่อง

### 2.3.3 Robot Operating System

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

#### Node

Node เป็นหนึ่งในองค์ประกอบหลักของ ROS, Node สามารถที่จะส่งข้อมูลหา Node อื่นๆได้ ผ่าน Topics หรือ Services ในทางปฏิบัติแล้ว Node เป็นตัวประมวลผลอยู่อย่างต่อเนื่องที่คอยทำหน้าที่เฉพาะ ยกตัวอย่างเช่น Node ตัวแรกเชื่อมต่อกับกล้อง เพื่อที่จะนำภาพจากกล้องออกมานะ Node ตัวที่สองใช้ในการหาลูกบล็อกที่อยู่ในภาพที่ได้มาจากการต่อเนื่องกัน Node ตัวแรก และ Node ตัวที่สามใช้ในการคำนวณหาตำแหน่งของลูกบล็อกที่อยู่บนโลกจริงๆ จากตำแหน่งของลูกบล็อกที่ได้มาจากการต่อเนื่องกัน Node ที่สอง ดังนั้นจะเห็นว่าแต่ละ Node จะทำงานเฉพาะของตัวเอง ซึ่งสามารถนำมาร่วมกันได้ การเขียนเป็นแบบ Node จะช่วยทำให้เราสามารถที่จะนำไปรวมกับ Node ใหม่ๆได้โดยง่าย ในกรณีที่จะนำไปทำงานอย่างอื่น ยกตัวอย่างเช่น Node ที่เอากาแฟจากกล้องออกมานะ อาจจะมี Node อีกตัว ทำหน้าที่ในการหาโกล์ดเป้าหมาย และหาทิศทางการเคลื่อนที่ของหุ่นยนต์ได้ ดังนั้นการพัฒนา Node เป็นส่วนย่อยๆเล็กๆ ก็เพื่อที่จะทำให้การแก้ไขหรือปรับปรุงได้ง่าย



รูปที่ 2.33: ตัวอย่างสถาปัตยกรรมของ ROS

จากตัวอย่างสถาปัตยกรรมของ ROS ดังรูปที่ ?? นั้นสามารถอธิบายได้ว่า หุ่นยนต์เคลื่อนที่ด้วยล้อมีการกิจคือ เคลื่อนที่ตามสั่นไปเรื่อยๆจนกว่าจะเจอเครื่องหมายหยุด Node คือตัวที่แสดงด้วยรูปวงรี ข้างในเป็นชื่อ Node ส่วน Topic จะแสดงด้วยรูปสี่เหลี่ยม ซึ่งข้างในเป็นชื่อของ Topic และชนิดของ Message ที่ใช้ในการส่งข้อมูล, มาดูกันก่อนอื่น ภาพถูกส่งมาจากกล้อง และก็มี Node สองตัวในการดูสีน้ำเงิน และเครื่องหมายหยุด จากภาพที่ได้มา เมื่อ Node ได้ข้อมูลแล้วก็นำมาระบบผลการเดินของหุ่นยนต์โดยส่งไปยัง node navigation และ Node นี้ ก็จะทำหน้าที่คำนวณความเร็วและทิศทางของหุ่นยนต์ ส่งไปยัง node robot\_control ซึ่งเป็นตัวสั่งการมอเตอร์ของหุ่นยนต์อีกด้วย

Twist.msg	
geometry_msgs/Vector3	linear
geometry_msgs/Vector3	angular

(ก) Message Twist

Stop.msg	
uint8	RED = 0 GREEN = 1
uint8	color
float32	distance

(ข) Message Stop

ตารางที่ 2.2: ตัวอย่างชื่อและข้อมูลของ Message

ตัวอย่างของ Message สองอันนี้ Twist message ดังรูปที่ ?? คือ message ที่เอาไว้บอกความเร็วเชิงเส้น และความเร็วเชิงมุม ซึ่ง ROS มี message ชนิดนี้ให้อยู่แล้ว ส่วน Stop message ดังรูปที่ ?? คือ message ที่เอาไว้บอกระยะทางและสีของป้าย Stop ซึ่ง message นี้ถูกสร้างขึ้นมาใหม่เพื่อใช้กับงานนี้โดยเฉพาะ

### Topics and Messages

Messages เป็นตัวหลักสำคัญในการติดต่อสื่อสารกันระหว่าง Node ใน ROS โดยที่ message จะถูกส่งผ่านไปยัง topic เสมอ แต่ละ Node สามารถที่จะ subscribe หรือ publish ไปกับ topic ก็ได้ การเชื่อมต่อกันระหว่าง Node นั้นสามารถสื่อสารกันได้โดยตรงในเครื่องคอมพิวเตอร์เครื่องเดียวกัน หรือเครื่องอื่นได้ที่อยู่ใน network เดียวกัน โดยจะติดต่อสื่อสารโดยใช้ TCP/IP การใช้คอมพิวเตอร์หลายเครื่องก็จะช่วยให้การประมวลผลมีประสิทธิภาพมากยิ่งขึ้น นอกจากนั้นยังสามารถที่จะแบ่งหน้าที่การทำงานออกกันได้ 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.ran ระบบ ในกระบวนการที่คือ subscribe node จะถาม rosmaster ว่ามี topic ที่ต้องการรับข้อมูลใหม่ ส่วนตัว master ที่เก็บค่า topic message เอาไว้ ก็จะส่งไปยัง subscribe node ถ้าหากมีข้อความที่ร้องขอมา และ rosmaster ก็จะจำไว้ว่ามี node ไหนซึ่งมี topic ที่ต้องการรับข้อมูลนั้น

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

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

## Services

Services หรืออีกชื่อหนึ่งคือ remote procedure calls (RPC) เป็นการส่ง messages แบบที่ไม่ได้เจาะจงว่าจะส่งไปที่ไหน เมื่อ service ถูกเรียกแล้วระบบจะรอนกว่าจะมีการตอบกลับ เราจะเรียกกระบวนการนี้ว่า 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 สำหรับงานที่ต้องใช้เวลาในการคำนวณนานจะไปใช้ 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 จะเริ่มจาก action client จะส่ง message goal ไปยัง action server แล้ว action server จะพยายามทำงานตาม goal ที่ได้รับมา ในระหว่างที่ action client ก็จะทำงานของตัวเองต่อไป แต่จะได้รับ feedback จาก action server อยู่ตลอดเวลา 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

```
<package>
    <name>example_package</name>
    <version>1.0.0</version>
    <description>Short example for a package.xml.</description>
    <maintainer emanil="ex@example.org">Jane Doe</maintainer>
    <license>BSD</license>
    <buildtool_depend>catkin</buildtool_depend>
    <build_depend>example_2</build_depend>
    <run_depend>std_msgs</run_depend>
</package>
```

รูปที่ 2.34: ตัวอย่างไฟล์ package.xml

แต่ละ tags ใช้ในการบอกข้อมูลของ package นี้ ใครเป็นเจ้าของ ใครเป็นคนเขียน รวมไปถึง dependencies ที่จำเป็นต้องใช้ของ package นี้ด้วย ดังรูปที่ ??

## Code Distribution

การที่จะนำ Nodes กลับมาใช้ใหม่หรือเอาอกมาแบ่งปันให้ผู้อื่นได้นั้น จะต้องมีการทำเอกสารของ Packages นั้นๆด้วย 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 ที่สามารถไปรัน package หรือ node อื่นๆ ได้ ตอนนี้ ROS มี packages มากกว่า 5000 packages แล้ว

Metapackage เป็นการรวมกันของ packages ที่ทำหน้าที่คล้ายๆกันหลายๆตัวมารวมไว้ที่เดียวเพื่อจะได้ใช้งานง่าย ตัวอย่าง Navigation metapackage ประกอบไปด้วย 10 packages เช่น [AMCL (partical filter), DWA, EKF (extended kalman filter) และ map\_server] ซึ่งหากติดตั้ง metapackage ตัวนี้ก็จะได้มาหมดเลย

ในส่วนนี้จะอธิบายคร่าวๆถึง ROS standard packages ที่จะนำมาใช้ในงานวิจัยครั้งนี้

**rosbag** rosbag เป็นแพกเกจที่สามารถบันทึก message ที่ส่งหากันในระหว่างที่ ROS กำลังทำงานได้ไฟล์ที่บันทึกจะเรียกว่า rosbag ประโยชน์ของมันคือเราสามารถเอาเข้ามาใช้ในการตรวจสอบ หรือนำมาเล่นซ้ำได้อีกทั้งยังง่ายต่อการค้นหาข้อผิดพลาดอีกด้วย

**tf2** tf2 เป็นแพกเกจที่สามารถติดตามการเปลี่ยนแปลงของ Coordinate frame เราสามารถใช้ในการหาความสัมพันธ์ระหว่าง frame ได้ ยกตัวอย่างเช่นหากเราต้องการหาตำแหน่งของ foot เทียบกับ pelvis ก็สามารถใช้ tf2 หาได้

**robot\_state\_publisher** robot\_state\_publisher แพกเกจที่ subscribe JointState message เพื่อที่จะนำตำแหน่งของข้อต่อ และแปลงให้อยู่ในรูปข้อมูลของ tf2, tf2 สามารถเรียกจาก Node ใดๆได้เพื่อที่จะหา Coordinate frame ที่ต้องการได้

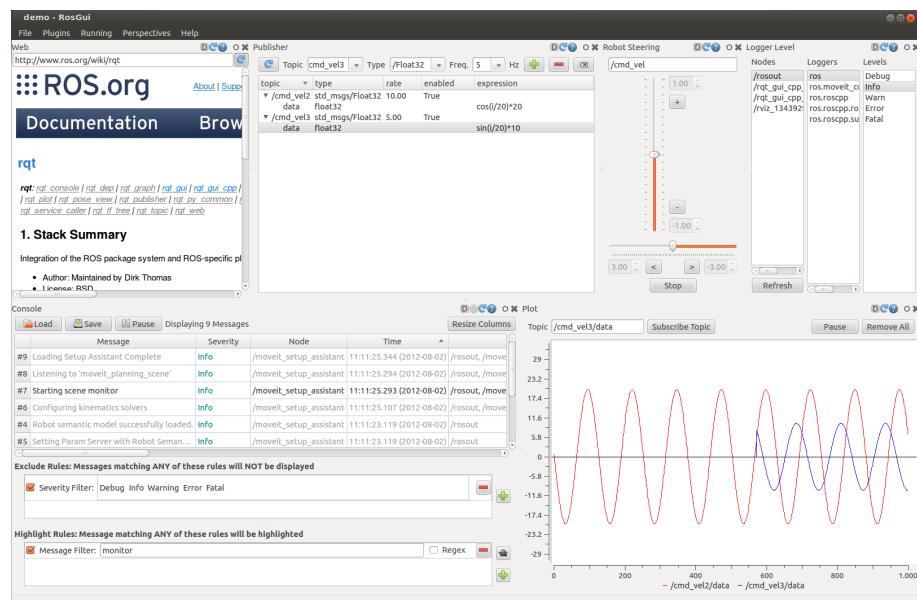
**URDF** Unified Robot Description Format (URDF) เป็นไฟล์ XML ที่เอาไว้อธิบายลักษณะของหุ่นยนต์ ใน ROS มีแพกเกจที่ใช้สำหรับการอ่านไฟล์ คือ urdf\_parser แต่ไฟล์นี้มีการใช้งานโดย tf2 เช่นกัน

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

## การแสดงผลด้วยภาพ

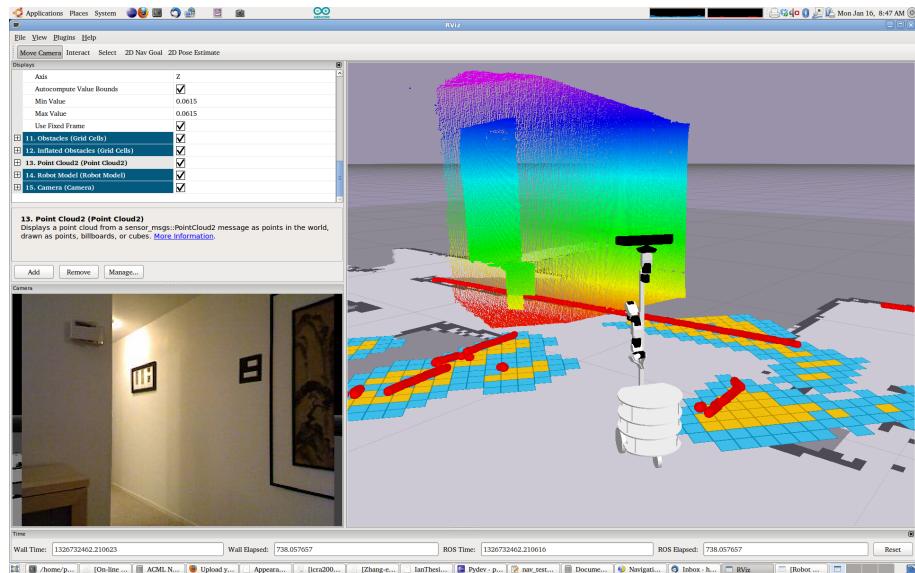
จุดแข็งสำคัญของ ROS อยู่ที่ว่ามีเครื่องมือที่ช่วยในการแสดงผลด้วยภาพได้ที่นอกเหนือจากระบบ publisher-subscriber การใช้เครื่องมือการแสดงผลด้วยภาพนี้จะช่วยให้การทำงานง่ายขึ้นและประหยัดเวลามากขึ้น ในกรณีที่ต้องนำข้อมูลต่างๆจากหุ่นยนต์ออกมาแสดงผล เพราะว่าเครื่องมือแสดงผลภาพนี้สามารถที่จะ subscribe จาก topic ที่มีการใช้งานอยู่แล้วมาแสดงผลได้ทันที ใน ROS มีเครื่องมือสำคัญอยู่ 2 ตัวที่ใช้สำหรับการแสดงผลด้วยภาพ ซึ่งสามารถที่จะปรับแต่งให้ถูกต้องเป็นเวอร์ชั่นของเราเองได้

**rqt** rqt เป็น UI ที่มีฐานมาจาก QT ซึ่งมาพร้อมกับการเชื่อมต่อ ROS เป็นส่วนเสริมในรูปแบบของ QWidget เราสามารถที่จะแสดงผลหลายๆ widgets ได้ภายในเวลาเดียวกัน สามารถที่จะย่อขยาย เปลี่ยนตำแหน่ง ลากวางได้ การเชื่อมต่อกับ ROS นั้นสามารถนำการแสดงผลภาพแบบ 2D ไปแสดงได้ดังรูปที่ ?? เป็นการแสดงภาพของกราฟที่ได้รับข้อมูลจาก topic หลายๆตัว และสามารถที่จะปรับแต่งค่าและ publish ออกไปได้ด้วยการเขียนโปรแกรมเข้าไป ซึ่งจะเป็นประโยชน์อย่างมากเวลาที่ใช้ในการปรับจูนพารามิเตอร์ต่างๆ เพราะว่าเราสามารถที่จะเปลี่ยนค่าได้ทันที ไม่ต้องรันโปรแกรมใหม่ ในรูปที่ ?? เป็นการนำ rqt มาเขียนเป็น GUI ให้ผู้ใช้สามารถใช้งานได้ง่ายและสามารถที่จะปรับแต่งพารามิเตอร์ต่างๆได้เรียลไทม์



รูปที่ 2.35: ตัวอย่างการแสดงผลใน rqt

**RViz** RViz เป็นการแสดงผลด้วยภาพแบบสามมิติของสถานะต่างๆของหุ่นยนต์และสภาพแวดล้อม โดยใช้ไฟล์ URDF เป็นมาตรฐานการแสดงถึงหุ่นยนต์ ซึ่งสามารถที่จะแสดงตำแหน่งปัจจุบันของข้อต่อต่างๆในหุ่นยนต์ได้ สามารถที่จะแสดงค่าเซนเซอร์เป็น marker ได้ การใช้งานจะเป็นเหมือนการบอกพิกัดเฟรม ลักษณะการแสดงผลใน RViz มีหลากหลายรูปแบบไม่ว่าจะเป็น camera images, depth clouds, laser scans หรือ point clouds อย่างไรก็ตามการแสดงผลใน Rviz นั้นจะไม่ได้คำนึงถึงแรงที่เข้ามายกระทำกับตัวของหุ่นยนต์ แต่ถ้าเป็นการเคลื่อนที่ ที่มีพิกัดเฟรมแล้วสามารถนำมาแสดงได้ ดังรูปที่ ?? เป็นตัวอย่างของหุ่นยนต์เคลื่อนที่ด้วยล้อ และทำแผนที่ด้วยข้อมูลความลึกที่ได้มาจากการ Kinect

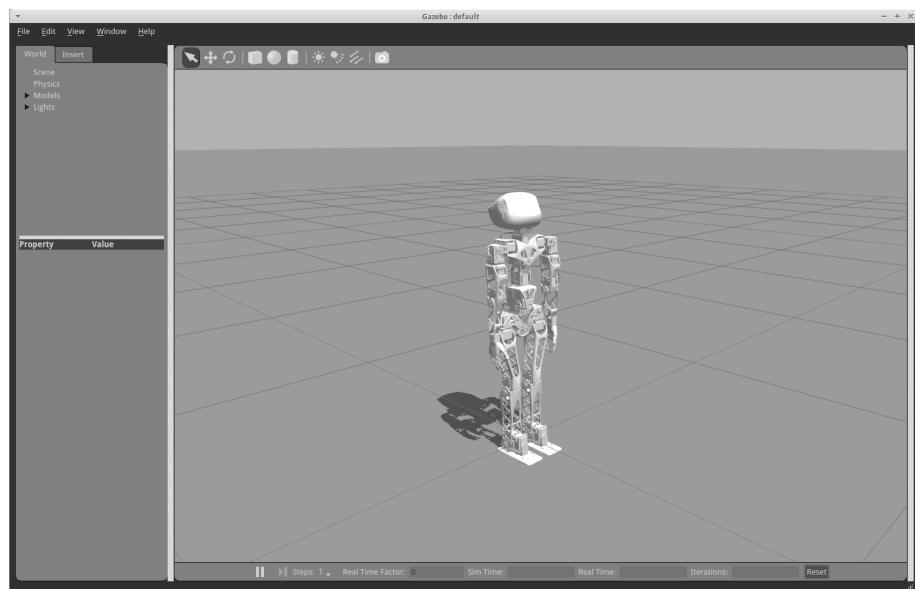


รูปที่ 2.36: ตัวอย่างการแสดงผลใน RViz

### ระบบจำลอง

ระบบจำลองเป็นส่วนที่สำคัญมากสำหรับการพัฒนาโปรแกรมของหุ่นยนต์ เพราะว่าเราสามารถที่จะสร้างโปรแกรมและทดสอบได้โดยไม่จำเป็นต้องมีฮาร์ดแวร์ ซึ่งในส่วนนี้จะช่วยลดความเสียหายจากบักหรือโปรแกรมที่ผิดพลาด ที่อาจจะเกิดขึ้นกับหุ่นยนต์ได้ การจำลองจะช่วยลดเวลาในการพัฒนาลงได้ ระบบจำลองปัจจุบันมีมากมายหลายตัวแต่ ตัวที่ได้รับคำแนะนำมากที่สุดคือ Gazebo เพราะว่า Gazebo สามารถที่จะเชื่อมต่อกับ ROS ได้โดยตรง และนักพัฒนาส่วนใหญ่ใช้ Gazebo

การจะใช้ Gazebo ได้นั้นเราจะต้องใช้ไฟล์ URDF ซึ่งเป็นไฟล์ที่เอาไว้แสดงหุ่นยนต์ในระบบจำลอง และสามารถที่จะคำนวนหา collision ให้เราได้อีกด้วย



รูปที่ 2.37: ตัวอย่างหุ่นยนต์หิวามันอยด์ Poppy

## 2.4 การออกแบบระบบพื้นฐาน

### 2.4.1 ความแตกต่างของ Operating Systems

เป็นที่รู้กันโดยทั่วไปว่า hardware และ software ของคอมพิวเตอร์นั้นถูกจัดการโดยโปรแกรมในคอมพิวเตอร์ ซึ่ง operating system (OS) งานพื้นฐานที่ OS ทำก็เช่น การควบคุมและจองหน่วยความจำ การจัดลำดับความสำคัญของระบบ โดยดูแลควบคุมอุปกรณ์ต่างๆ ที่เชื่อมต่อเข้ากับคอมพิวเตอร์ การเชื่อมต่อระบบเน็ตเวิร์ค การจัดการไฟล์ข้อมูล อีกทั้งยังรวมไปถึงการให้บริการต่างๆ เช่น การจัดการกระบวนการประมวลผล จัดการไฟล์ ของระบบต่างๆ ระบบป้องกัน อื่นๆ

ปัจจุบันมี OS อยู่หลายตัวเช่น Windows, Mac OS X, UNIX, Solaris BS3000, MS-Dos และอื่นๆ ซึ่งทั้งหมดนี้เป็นส่วนหนึ่งของระบบคอมพิวเตอร์ที่จะคอยช่วยจัดการและควบคุมดูแลการทำงานต่างๆ ของคอมพิวเตอร์ ระบบคอมพิวเตอร์นั้นอาจจะอยู่ในรูปแบบอื่นๆ เช่น เครื่องที่ใช้ทำงาน, เครื่องเซิฟเวอร์, เครื่องคอมพิวเตอร์ส่วนบุคคล, โทรศัพท์เคลื่อนที่, อุปกรณ์นำทาง หรือแม้กระทั่งระบบที่มีความฉลาดในตัวมันเอง เช่น หุ่นยนต์ และ OS นั้นจะสามารถทำงานบน hardware อุปกรณ์ใดๆ ก็ได้

### 2.4.2 ข้อแตกต่างระหว่าง Open platform กับ Non-open platform

หุ่นยนต์ Open platform คือ การออกแบบระบบพื้นฐานของหุ่นยนต์ที่เปิดให้ผู้ที่ต้องการศึกษาหรือผู้ใช้ทั่วไปสามารถเข้าถึงข้อมูลต่างๆ ที่เกี่ยวข้องกับหุ่นยนต์นั้นๆ ได้ ผู้ใช้สามารถที่จะนำข้อมูลเหล่านั้นมาแก้ไข ปรับปรุง แต่งตั้ง หรือเรียนรู้และพัฒนาตามได้ด้วยตนเอง ซึ่งข้อมูลที่กล่าวมานั้นสามารถได้จากเว็บไซต์ของผู้พัฒนาหุ่นยนต์ ปัจจุบันมีหุ่นยนต์ที่มีความสามารถอยู่ที่เป็นเปิดให้เข้าถึงหลายรูปแบบแตกต่างกันไป

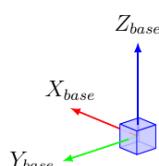
ส่วนหุ่นยนต์ Non-open source platform คือหุ่นยนต์ที่สร้างมาเฉพาะเจาะจงสำหรับการวิจัย การสำรวจ หรือการแข่งขันโดยเฉพาะ ไม่เปิดให้บุคคลภายนอกเข้าศึกษาหรือแก้ไขปรับปรุง ซึ่งทำให้หุ่นยนต์ประเภทนี้ไม่เหมาะสมสำหรับผู้ใช้ที่จะเรียนรู้และศึกษาด้วยตนเอง เพราะมีขั้นตอนใหญ่ ใช้ทรัพยากร่มาก และการออกแบบมีความซับซ้อน เรียนรู้ยากกว่าหุ่นยนต์แบบ Open platform

### 2.4.3 มาตรฐานหน่วยวัดและการนองค์พิกัด

การใช้หน่วยวัดที่ไม่ตรงกันอาจจะทำให้เกิดปัญหาขึ้นได้ เนื่องจากเป็นแพลตฟอร์มนั้นจะมีบุคคลอื่นช่วยกันพัฒนาหลายคน จึงควรที่จะมีมาตรฐานในการวัดและการกำหนดพิกัดต่างๆ ที่ตรงกันเพื่อให้เกิดความชัดเจนในการทำความเข้าใจ

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

**พิกัดเฟรม** การบอกทิศทางการหมุนนั้นใช้หลักตามกฎมือขวา โดยการตั้งแกนนั้นหากเทียบกับมือแล้ว X ไปข้างหน้า Y ไปทางซ้าย Z พุ่มขึ้น ดังรูปที่ ??



รูปที่ 2.38: การตั้งแกนตามกฎมือขวา

ปริมาณ (Quantity)	หน่วยวัด (Unit)	สัญลักษณ์ (Symbol)
ความยาว (Length)	เมตร (metre)	$m$
มวล (Mass)	กิโลกรัม (kilogram)	$kg$
เวลา (Time)	วินาที (second)	$s$
กระแสไฟฟ้า (Electric Current)	แอมเปอร์ (ampere)	$A$
มุม (Angle)	เรเดียน (radian), องศา (degree)	$rad, deg$
ความถี่ (Frequency)	เฮิร์ต (Hertz)	$Hz$
แรง (Force)	นิวตัน (Newton)	$N$
กำลัง (Power)	วัตต์ (Watt)	$W$
แรงดันไฟฟ้า (Voltage)	โวลต์ (Volt)	$V$
อุณหภูมิ (Temperature)	เซลเซียส (Celsius)	$^{\circ}C$

ตารางที่ 2.3: ตารางแสดงหน่วยวัดมาตรฐาน

#### 2.4.4 Robot Operating System

ROS เป็นกรอบการทำงานที่ได้รับความนิยมและมีประสิทธิภาพในการทำงานมากที่สุดในปัจจุบัน เนื่องจาก ROS ได้รวมซอฟต์แวร์เครื่องมือที่หลากหลายเอาไว้เป็นหมวดหมู่ เช่น การเชื่อมต่อกับฮาร์ดแวร์ การสร้างระบบควบคุมให้กับอุปกรณ์ต่างๆ อีกทั้งสามารถที่จะเขียนโปรแกรมแล้วนำกลับมาใช้ใหม่ได้ ภายในระบบมีกระบวนการรับส่งข้อมูลต่างๆ เป็นของตัวเอง ทำให้ช่วยลดความซับซ้อนและเพิ่มประสิทธิภาพในการทำงานกับแพลตฟอร์มของหุ่นยนต์ กระบวนการเขียนโปรแกรมของ ROS นั้นจะใช้รูปแบบ Graph architecture ซึ่งจะทำให้สามารถแบ่งโปรแกรมต่างๆ ออกเป็นส่วนๆ เช่น เชนเซอร์หลายๆ ตัว ระบบควบคุม ระบบวางแผน ระบบขับเคลื่อน ระบบสื่อสารภายนอก ด้วยตัวระบบของ ROS นั้น ไม่ใช่ Real Time OS แต่อย่างไรก็ตามเราสามารถใช้งานผสมกับ Real Time ได้

ROS ประกอบไปด้วยแพ็กเกจเจกต่างๆ มาประกอบกันเป็น Node โดยมีตัวกลางทำหน้าที่ในการติดต่อสื่อสารระหว่างอุปกรณ์ที่เป็น Node ต่างๆ ให้สามารถส่งข้อมูลหากันได้ รูปแบบการสื่อสารใน ROS จะใช้หลักการแบบ Publish/Subscribe ทำให้ไม่จำเป็นที่จะต้องระบุโปรแกรมที่จะรับ ภาษาในการพัฒนามีให้เลือกที่หลากหลาย เช่น C++, Python, Lisp, MATLAB หรือ JavaScript

#### ประโยชน์จากการใช้ ROS

ROS เป็นกรอบการทำงาน ที่อยู่ระหว่าง OS และ Robot ทำให้เราไม่ต้องกังวลเรื่องการจัดการระบบภายใน เพราะ ROS จะช่วยจัดการให้เราทั้งหมด ก่อนจะมี ROS นั้น นักวิจัยจะต้องใช้เวลาไปกับพัฒนาพื้นฐานให้หุ่นยนต์ ซึ่งจะต้องมีทักษะทางด้านเครื่องกล ไฟฟ้า และโปรแกรม ซึ่งบอยครั้งที่นักวิจัยหรือนักพัฒนานั้นไม่มีความรู้ หรือประสบการณ์ในการสร้างหุ่นยนต์ ทำให้การทำงานเป็นไปด้วยความลำบาก

## บทที่ 3

### การดำเนินงานวิจัย

วิทยานิพนธ์เรื่องการออกแบบโครงสร้างและพัฒนาระบบพื้นฐานของหุ่นยนต์ชีวภาพอยู่ด้านนี้ ผู้วิจัยได้มีการตั้งชื่อให้หุ่นยนต์โดยใช้ชื่อว่า อุทัย (UTHAI) มาจากภาษาอังกฤษคำว่า Universal Template for Humanoid Algorithm Interface เพื่อให้สมกับเป็นหุ่นยนต์ชีวภาพอยู่ด้วยที่ใช้สำหรับงานวิจัยและพัฒนาต่ออยอด

#### 3.1 หน้าที่ความรับผิดชอบ

**จิรภัสส์ ศรีรัตนอากรณ์** ออกแบบระบบอิเล็กทรอนิกส์และขั้นรูปโครงสร้างทางกล

**เจษฎากร ทาไชยววงศ์** ออกแบบและจัดสร้างโครงสร้างทางกล

**วุฒิภัทร โชคอนันตทรัพย์** ออกแบบระบบพื้นฐานและระบบจำลอง

#### 3.2 แผนการดำเนินงาน

โดยจากที่กล่าวไปตอนต้นในบทนำ การดำเนินงานและการออกแบบการสร้างหุ่นยนต์ชีวภาพอยู่ด้วย UTHAI มีแผนการทำงานซึ่งแบ่งออกเป็นสามส่วนดังนี้ ส่วนแรกคือ ส่วนของอาร์ดแวร์ที่เกี่ยวกับโครงสร้างทางกลของหุ่นยนต์ชีวภาพอยู่ด้วย เช่น ข้อต่อ ก้านต่อ ฝ่าเท้า รวมไปถึงระบบอิเล็กทรอนิกส์ ตัวประมวลผลการควบคุม เช่นเซอร์ตัวขับเคลื่อนต่าง ๆ และส่วนที่สองคือ ส่วนของซอฟต์แวร์ที่เกี่ยวข้องกับการติดต่อสื่อสารกันเบื้องต้น การควบคุมตัวขับเคลื่อนที่ข้อต่อ การอ่านค่าเซนเซอร์ และส่วนที่สาม คือระบบพื้นฐานสำหรับการนำไปศึกษาและพัฒนา โดยจะครอบคลุมไปถึงเอกสารวิธีการใช้งานในรูปแบบออนไลน์

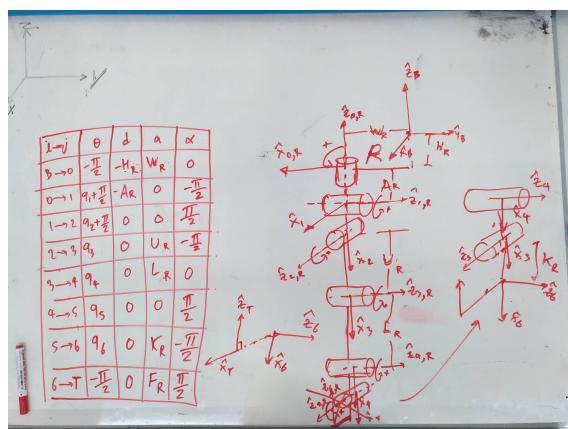
ในการเริ่มทำงานวิจัยที่เกี่ยวกับชีวภาพอยู่ด้านนี้สิ่งจำเป็นที่ต้องทำในอันดับแรกคือการศึกษาสิ่งที่เคยมีอยู่ หรืองานวิจัยมีนักวิจัยอื่นที่ทำเอาไว้แล้ว จากนั้นศึกษาทำความเข้าใจใน ข้อดี-ข้อเสีย ของวิธีหรือกระบวนการต่างๆ เพื่อนำมาประยุกต์ใช้กับหุ่นยนต์ชีวภาพอยู่ภายในงานวิจัยนี้ ในการศึกษาโครงสร้างทางกลและระบบของหุ่นยนต์ชีวภาพอยู่ด้วยที่มีอยู่แล้วสิ่งที่ต้องดูและให้ความสนใจเป็นพิเศษคือ วิธีการเชื่อมตอกันระหว่างก้านต่อและข้อต่อ ลักษณะโครงสร้างที่สามารถทำให้เกิดการเคลื่อนไหวเพียงพอต่อการเดิน ตำแหน่งที่ใช้ในติดตั้งเซนเซอร์ หรือตัวรับรู้ต่างๆ การเลือกใช้วัสดุให้เหมาะสม และการทำงานของเซนเซอร์และตัวขับเคลื่อนที่จำเป็นต้องใช้ในการควบคุมการทำงานของหุ่นยนต์ เมื่อเราทำความศึกษาเสร็จเรียบร้อยแล้ว จึงเริ่มดำเนินงานวิจัยการพัฒนาโครงสร้างและระบบพื้นฐาน ในส่วนนี้ทางกลจะเป็นการออกแบบโครงสร้างและจัดสร้างขึ้นมา ทางซอฟต์แวร์จะเป็นการสร้างแบบจำลองของหุ่นยนต์ ระบบจำลองที่สามารถจำลองการทำงานของหุ่นยนต์ได้ และติดต่อส่วนของทางกลกับทางโปรแกรม สุดท้ายระบบพื้นฐานจะออกแบบโครงสร้างระบบพื้นฐานเพื่อทำให้เกิดการพัฒนาต่อได้อย่างมีประสิทธิภาพ ในบทนี้ก็จะกล่าวถึงกระบวนการออกแบบและการดำเนินการตามแผนที่วางเอาไว้

### 3.3 การออกแบบโครงสร้างของหุ่นยนต์

การออกแบบทางโครงสร้างทางกลของหุ่นยนต์ชีวามโนยด์ UTHAI นั้น ผู้วิจัยได้เลือกใช้โปรแกรมออกแบบโครงสร้างเป็นโปรแกรม Solidworks ที่ช่วยในการพัฒนาแบบจำลองของหุ่นยนต์ชีวามโนยด์ UTHAI เนื่องจากโปรแกรม Solidworks เป็นโปรแกรมที่ใช้ในการออกแบบ และวัดแบบทางวิศวกรรม สามารถวิเคราะห์โครงสร้าง การรับแรงของแบบจำลองได้ อีกทั้งยังเป็นเครื่องมือที่มีการใช้งานอย่างแพร่หลาย ทำให้สามารถดูรายละเอียดแบบจำลองต่างๆที่มีค่านพัฒนาเข้าเข้ามาใช้ร่วมกับการออกแบบได้ และด้วยทางผู้วิจัยมีความชำนาญในการพัฒนา ต่อยอดเป็นหลัก ดังนั้นการออกแบบโครงสร้างทางกลของหุ่นยนต์ชีวามโนยด์ UTHAI จึงถูกออกแบบมาเพื่อให้สามารถรองรับการปรับเปลี่ยน เปลี่ยนแปลง แก้ไขขึ้นส่วนต่างๆของตัวหุ่นยนต์ชีวามโนยด์ UTHAI ต่อไปได้

#### 3.3.1 โครงสร้างของหุ่นยนต์ชีวามโนยด์ UTHAI

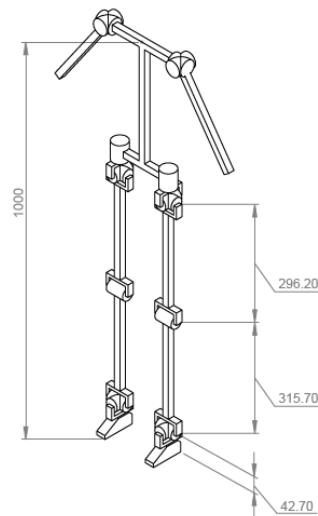
หุ่นยนต์ชีวามโนยด์ที่สร้างขึ้นประกอบด้วยส่วนของลำตัวและส่วนขา ในขาแต่ละข้างออกแบบให้ข้อต่อ มีเป็นลักษณะของข้อต่อแบบหมุน (Revolute joint) เพื่อเลียนแบบโครงสร้างของมนุษย์ที่ประกอบด้วย ส่วนของสะโพกที่มีองศาอิสระจำนวน 3 องศาอิสระ หัวเข่า 1 องศาอิสระ และข้อเท้า 2 องศาอิสระ รวมทั้งขาข้างละ 6 องศาอิสระ ระบบต้นกำลังที่ใช้ในแต่ละข้อต่อเป็นดิจิตอลเซอร์โว การออกแบบหุ่นยนต์ชีวามโนยด์ UTHAI นั้น สิ่งแรกที่ต้องทำ คือ การกำหนดโครงสร้างของข้อต่อและก้านต่อ ดังรูปที่ ?? ซึ่งโครงสร้างนั้นทางผู้วิจัยได้อ้างอิงมาจากสัดส่วนของมนุษย์จริง และมีจุดศูนย์กลางมวลอยู่บริเวณกระดูกเชิงกรานของตัวหุ่นยนต์ชีวามโนยด์เอง ดังรูปที่ ??



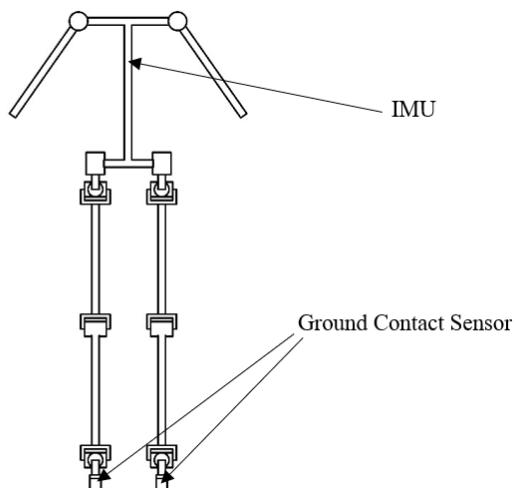
รูปที่ 3.1: การออกแบบโครงสร้างของหุ่นยนต์ UTHAI

เมื่อเราได้แบบของหุ่นยนต์ชีวามโนยด์ UTHAI แล้ว ลำดับต่อไปคือการกำหนดตำแหน่งการติดตั้งเซนเซอร์ และตัวขับเคลื่อนต่างๆเข้าไป โดยมีเซนเซอร์ตรวจการสัมผัสพื้นติดตั้งที่ได้ฝาท้ายของหุ่นยนต์ เช่นเซอร์หน่วยวัดความเร็วอยู่ติดตั้งไว้บริเวณลำตัวของหุ่นยนต์ชีวามโนยด์ และที่ข้อต่อในแต่ละจุดใช้ตัวขับเคลื่อนเป็นดิจิตอลเซอร์โว ดังรูปที่ ??

ส่วนวัสดุที่ใช้ทำโครงสร้างหุ่นยนต์ชีวามโนยด์ UTHAI ทางผู้วิจัยเลือกใช้วัสดุหลักเป็น PLA ที่ขึ้นรูปด้วยวิธีการขึ้นรูปสามมิติ ก้านต่อเป็นคาร์บอนไฟเบอร์ เนื่องจากจะทำให้โครงสร้างมีน้ำหนักเบา ขึ้นรูปได้สะดวกทำให้สามารถปรับปรุงแก้ไขง่าย และมีวัสดุเสริมบางชิ้นส่วนที่ทำจากอลูมิเนียมเนื่องจากต้องความแข็งแรงมาก ตั้งตารางที่ ??



รูปที่ 3.2: ภาพแสดงแสดงโครงสร้างของหุ่นยนต์ UTHAI



รูปที่ 3.3: ภาพแสดงการติดตั้งเซนเซอร์ในจุดต่างๆ

Material	Longitudinal Tensile Strength (ksi)	Density ( $g/cm^3$ )
Carbon Fiber	300	1.55
Steel	100	7.7
Titanium	120	4.34
Aluminum	35	2.7
PLA 3D printing (50 % infill)	3.5	1.26
PLA 3D printing (100 % infill)	5.5	1.26

ตารางที่ 3.1: ตารางแสดงสมบัติทางกลของวัสดุต่าง ๆ

### 3.3.2 การจัดทำขึ้นส่วนโครงสร้างของหุ่นยนต์ชีวามโนยร์ UTHAI

ในการจัดทำขึ้นส่วนโครงสร้างของหุ่นยนต์ชีวามโนยร์อุทัยนันททาง ผู้วิจัยได้คำนึงถึงความแข็งแรงเป็นหลักซึ่งมีความสำคัญมาก ในการเคลื่อนที่ของหุ่นยนต์ อีกทั้งยังต้องมีน้ำหนักที่เบาเพื่อประหยัดพลังงานที่ต้องใช้ในการขับเคลื่อน<sup>1</sup> ดังนั้นผู้วิจัยจึงได้ใช้การขึ้นรูปชิ้นงานด้วยเทคนิคการพิมพ์สามมิติ โดยจะใช้วัสดุหลักเป็นพลาสติก PLA ซึ่งมีความแข็งมากกว่าและขึ้นรูปง่ายกว่าพลาสติกชนิด ABS ดังตารางที่ ?? และก้านต่อเป็นคาร์บอนไฟเบอร์ เพื่อให้ตอบโจทย์กับหุ่นยนต์แพลตฟอร์มเพื่อพัฒนาต่อยอดในอนาคต ซึ่งผู้ใช้ทุกคนสามารถพิมพ์ชิ้นงานขึ้นมาได้ด้วยตนเอง<sup>2</sup>

Properties	ABS	PLA
Tensile Strength	27 MPa	37 MPa
Elongation	3.5 - 50%	6%
Flexural Modulus	2.1 - 7.6 GPa	4 GPa
Density	1.0 - 1.4 g/cm <sup>3</sup>	1.3 g/cm <sup>3</sup>
Melting Point	230°C - 240°C	215°C - 235°C
การย่อ屈สลายทางธรรมชาติ	ไม่ได้	ได้ (ภายใต้เงื่อนไขที่ถูกต้อง)

ตารางที่ 3.2: ตารางแสดงสมบัติทางกลของวัสดุพลาสติก

แต่เนื่องจากในปัจจุบันนี้เครื่องพิมพ์สามมิติส่วนมากจะไม่รองรับการพิมพ์ชิ้นงานที่มีขนาดใหญ่ ซึ่งมีขนาดมากกว่า 30x30x30 ซม. (กว้างxยาวxสูง) ดังนั้นชิ้นงานที่ขึ้นรูป ที่มีขนาดใหญ่เกินกว่านี้อาจจะต้องทำการตัดชิ้นงานออกเป็นส่วนย่อยๆก่อน และจึงค่อนขานำมาประกอบรวมกันทีหลังอีกครั้งหนึ่ง โดยพื้นที่ทำการพิมพ์ของเครื่องพิมพ์สามมิติที่มีวางจำหน่าย<sup>3</sup> และใช้งานแพร่หลายในท้องตลาดแสดง ดังตารางที่ ??

ชื่อเครื่องพิมพ์สามมิติ	ขนาดความกว้าง (มม.)	ขนาดความลึก (มม.)	ขนาดความสูง (มม.)
MakerBot Replicator+	292	192	165
Ultimaker 3	188	185	200
LulzBot Mini	152	152	158
Dreammaker Overlord Pro Plus	79	79	255
New Matter MOD-t	145	95	125

ตารางที่ 3.3: ตารางแสดงขนาดของชิ้นงานที่สามารถพิมพ์ได้ในเครื่องพิมพ์ชนิดต่างๆ

<sup>1</sup>Printing Guide [<https://filaments.ca/pages/temperature-guide>]

<sup>2</sup>PLA vs ABS [<https://www.3dhubs.com/knowledge-base/pla-vs-abs-whats-difference>]

<sup>3</sup>The truth about 3D printer maximum print areas [<https://www.zdnet.com/article/what-manufacturers-don-t-want-you-to-know-the-truth-about-3d-printer-maximum-print-areas/>]

### 3.3.3 อุปกรณ์ที่ใช้ในหุ่นยนต์อิวามานอยด์ UTHAI

#### Dynamixel Servo EX-106+

Dynamixel EX-106+ เป็นตัวขับเคลื่อนที่นิยมใช้ในหุ่นยนต์ปัจจุบัน ซึ่งเป็นเซอร์โวโมเตอร์ที่เหมาะสมสำหรับทำหุ่นยนต์โดยเฉพาะ ภายในประกอบไปด้วย มอเตอร์กระแสตรง ชุดเฟืองมอเตอร์ ไดรเวอร์คอนโทรลเลอร์ สามารถเชื่อมต่อ กันผ่านบัส RS-485<sup>4</sup> มีการควบคุมตำแหน่งและความเร็วด้วย PID ภายในมีเซนเซอร์สามารถที่จะอ่านค่าความเร็ว แรงดันไฟฟ้า กระแสไฟฟ้า อุณหภูมิ ตำแหน่ง และแรงบิดจากมอเตอร์ทุกด้านได้ แต่ละมอเตอร์แต่ละตัวจะมีบอร์ดควบคุมของตัวเอง เราสามารถที่จะจ่ายไฟให้มอเตอร์และควบคุมผ่าน Serial ได้ ดังรูปที่ ?? การทำงานของตัวดิจิตอลเซอร์โวนี้สามารถทำได้ 2 รูปแบบ<sup>5</sup> คือ

**Joint Mode** สามารถที่จะควบคุมแรงบิด ความเร็ว และตำแหน่งได้ ความละเอียดในการควบคุมตำแหน่งอยู่ที่ 10bit (0-1023) หมุนได้อยู่ในช่วง -150 ถึง 150 องศา

**Wheel Mode** สามารถที่จะควบคุมแรงบิด ความเร็ว และทิศทางได้ ความละเอียดของความเร็วอยู่ที่ 10bit (0-1023) สามารถหมุนได้ครบ 360 องศาได้ ความเร็วสูงสุดอยู่ที่ 114 RPM

EX-106 Stats		
<b>Operating Voltage</b>	18.5V	14.8V
<b>Holding Torque</b>	107 kg-cm 1,485 oz-in	84 kg-cm 1,166 oz-in
<b>No-load Speed</b>	0.143 sec/60°	0.182 sec/60°
<b>Weight</b>	154g	
<b>Size</b>	40.2 x 65.1 x 46 mm	
<b>Resolution</b>	0.06°	
<b>Reduction Ratio</b>	1/184	
<b>Operating Angle</b>	251° or Continuous Turn	
<b>Max Current</b>	7000mA	
<b>Standby Current</b>	55 mA	
<b>Operating Temp</b>	-5°C ~ 85°C	
<b>Protocol</b>	RS485 Asynchronous Serial	
<b>Module Limit</b>	254 valid addresses	
<b>Com Speed</b>	7343bps ~ 1Mbps	
<b>Position Feedback</b>	Yes	
<b>Temp Feedback</b>	Yes	
<b>Load Voltage Feedback</b>	Yes	
<b>Input Voltage Feedback</b>	Yes	
<b>Compliance/PID</b>	Yes	
<b>Material</b>	Metal Gears & Engineering Plastic Body	
<b>Motor</b>	Maxon RE-MAX	
<b>Manual Download</b>	<a href="#">EX-106 manual (PDF)</a>	
<b>Controller List</b>	<a href="#">USB2Dynamixel CM2+</a>	



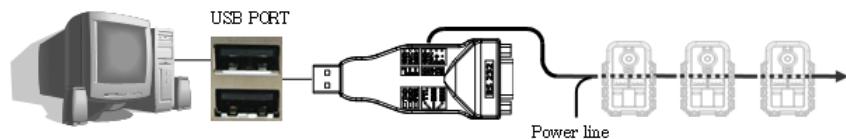
รูปที่ 3.4: แสดงประสิทธิภาพของมอเตอร์ EX-106+

<sup>4</sup>Robot Actuator [[http://support.robotis.com/en/product/actuator/dynamixel/ex\\_series/ex-106.htm](http://support.robotis.com/en/product/actuator/dynamixel/ex_series/ex-106.htm)]

<sup>5</sup>EX-106+ Mode [<http://www.trossenrobotics.com/dynamixel-ex-106-robot-actuator.aspx>]

### USB2Dynamixel

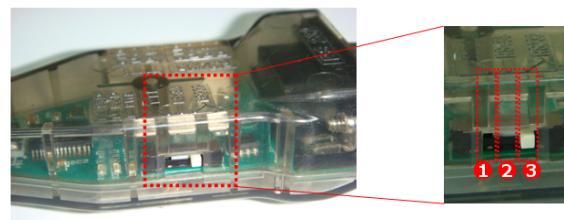
USB2Dynamixel เป็นอุปกรณ์สำหรับเชื่อมต่อดิจิตอลเซอร์โว กับหน่วยประมวลผลระดับสูง (Odroid) โดยจะเชื่อมต่อผ่านพอร์ท USB ที่อยู่บน Odroid และแปลงให้ส่งข้อมูลไปยังดิจิตอลเซอร์โวผ่านสาย 2 เส้น คือ D+ และ D- ดังรูปที่ ?? เป็นการเชื่อมต่อแบบ RS-485<sup>6</sup> ทำให้สามารถส่งข้อมูลไปยังหลายๆ อุปกรณ์บนสายเส้นเดียวได้ และสามารถส่งในระยะทางไกลๆ<sup>7</sup>



รูปที่ 3.5: ภาพแสดงการติดต่อสื่อสารระหว่างคอมพิวเตอร์กับดิจิตอลเซอร์โว

ในการใช้งาน ผู้ใช้งานจำเป็นต้องเลือกรูปแบบการติดต่อสื่อสารซึ่งแต่ละรูปแบบก็จะมีลักษณะการเชื่อมต่อที่แตกต่างกันไป USB2Dynamixel ได้แบ่งการติดต่อสื่อสารออกเป็น 3 รูปแบบ<sup>8</sup> คือ ดังรูปที่ ??

- 1 TTL Communication : สำหรับดิจิตอลเซอร์โวที่ใช้พอร์ทชนิด 3-pin ในตระกูล AX Series
- 2 RS485 Communication : สำหรับดิจิตอลเซอร์โวที่ใช้พอร์ทชนิด 4-pin ในตระกูล DX Series
- 3 RS232 Communication : ใช้สำหรับติดต่อสื่อสารกับคอนโทรลเลอร์ผ่านสายเคเบิล



รูปที่ 3.6: ภาพแสดงการเลือกโหมดใช้งานของ USB2Dynamixel

แต่จากการทดลองนำมาใช้ ผู้วิจัยพบว่าดิจิตอลที่ใช้ในการทำงานวิจัยครั้งนี้เป็นชนิด 4-pin ซึ่งใช้ RS485 ในการติดต่อสื่อสาร ด้วยขนาดของตัว USB2Dynamixel มีขนาดที่ใหญ่ทำให้การทำงานมีความลำบากในการติดตั้งลงบนตัวของหุ่นยนต์ชีวนมอยด์อุทัย จึงได้ทำการเปลี่ยนเป็น USB2RS485<sup>9</sup>



รูปที่ 3.7: USB2RS485 Module

<sup>6</sup>USB2Dynamixel [[http://support.robotis.com/en/product/auxdevice/interface/usb2dxl\\_manual.html](http://support.robotis.com/en/product/auxdevice/interface/usb2dxl_manual.html)]

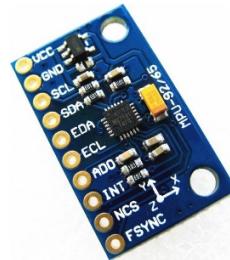
<sup>7</sup><http://support.robotis.com/en/images/product/auxdevice/interface/>

<sup>8</sup>[http://support.robotis.com/en/product/auxdevice/interface/usb2dxl\\_manual.htm](http://support.robotis.com/en/product/auxdevice/interface/usb2dxl_manual.htm)

<sup>9</sup><https://softgenie.dk/diverse/280-usb2rs485.html>

### Inertial Measurement Unit (IMU)

ผู้วิจัยได้เลือกใช้เซนเซอร์หน่วยวัดความเสี่ยงเป็น MPU-9250 ดังรูปที่ ?? มาใช้ในการอ่านหาค่ามุมเอียงของฐานทุนยนต์อิมามอยด์อุทัย เพื่อใช้ในการคุณเสถียรภาพของหุนยนต์ โดยเซนเซอร์ตัวนี้สามารถวัดค่าได้ 9 แกนคือ วัดค่าความเร็วเชิงมุม (gyroscope) 3 แกน วัดค่าความเร่งเชิงเส้น (accelerometer) 3 แกน และวัดค่าสนามแม่เหล็กโลก (magnetometer) 3 แกน ซึ่งเซนเซอร์จะติดตั้งบริเวณส่วนของลำตัวหุนยนต์อิมามอยด์อุทัย เนื่องจากว่าจะเป็นจุดที่สามารถบ่งบอกได้ถึงการเคลื่อนที่และมุมเอียงของหุนยนต์ในขณะนั้นได้<sup>10</sup>



รูปที่ 3.8: แสดงเซนเซอร์ IMU MPU9250

### Wi-Fi Adapter

เนื่องจากหน่วยประมวลผลขั้นสูงของหุนยนต์อิมามอยด์อุทัยนั้น ไม่สามารถเข้ามาร่วมกับหน่วยประมวลผลระดับสูงที่ติดตั้งอยู่บนตัวของหุนยนต์ และคอมพิวเตอร์ที่เป็นตัวสั่งการซึ่งอยู่นอกตัวของหุนยนต์ ซึ่งในงานวิจัยนี้ จะใช้ส่งข้อมูลที่ได้หลังจากการประมวลผลบนคอมพิวเตอร์ไปยังตัวหุนยนต์ เช่น การวางแผนการเดิน การคำนวนพลศาสตร์ของหุนยนต์ และอื่นๆ โดยการส่งข้อมูลไปยังหน่วยประมวลผลระดับสูงจะมีตัวกลางในการรับส่งสัญญาณคือ ตัวกระจายสัญญาณ (Wifi router)<sup>11</sup>



รูปที่ 3.9: ตัวรับสัญญาณ Wifi ของ RaspberryPi



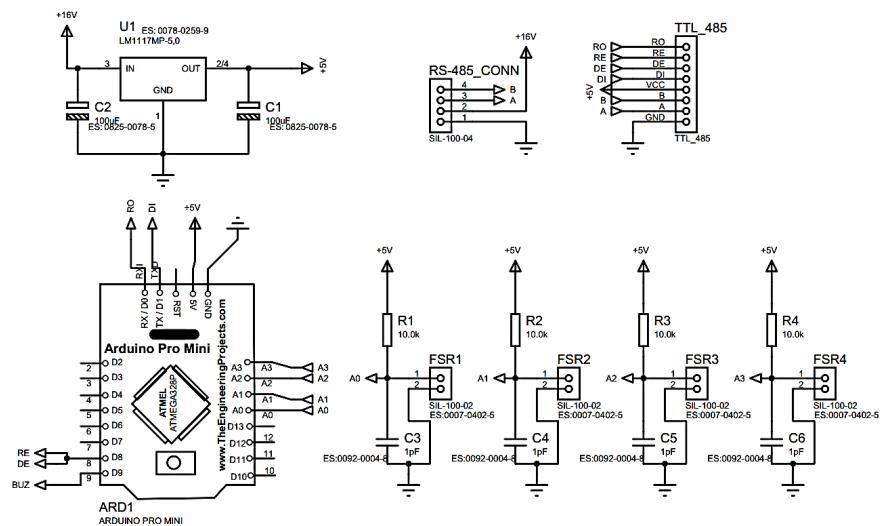
รูปที่ 3.10: ตัวกระจายและรับส่งสัญญาณ wifi

<sup>10</sup>MPU-9250 [http://www.arduino.com/en/gy-series-axis-accelerometers/6924-gy9255-mpu9255-sensor-module-alternative-mpu9150-mpu9250-3809200640200.html]

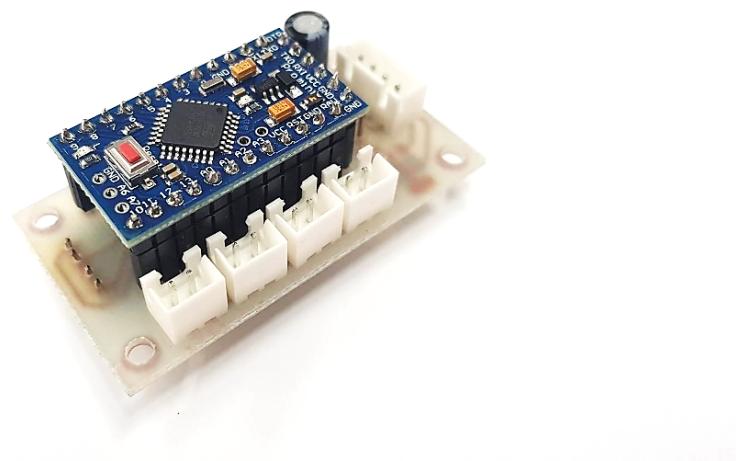
<sup>11</sup><https://www.alzashop.com/tp-link-archer-c7-d470129.htm>

### Ground contact sensor

เซนเซอร์ตรวจหน้าสัมผัสที่พื้นเป็นเซนเซอร์ที่ถูกติดตั้งบริเวณฝ่าเท้า เพื่อตรวจสอบการเดินของหุ่นยนต์ ขีวามโนยด้วnxะนีมีการสัมผัของฝ่าเท้าของหุ่นยนต์กับพื้นหรือไม่ ซึ่งในงานวิจัยนี้ได้ใช้หลักการตัวตรวจจับแรงกดแบบค่าความต้านทานหรือ Force Sensing Resistor (FSR) แนวคิดการออกแบบหลัก คือการออกแบบให้สามารถติดตั้งกับตัวหุ่นยนต์ได้โดยใช้ Arduino การพัฒนาซึ่งให้สามารถอ่านค่าได้ทั้งอนาล็อกและดิจิตอลได้อีกทั้งรองรับการต่อเซนเซอร์จำนวน 4 ตัว โดยมีลักษณะรูปที่ ?? และตัวอย่างของเซนเซอร์ตรวจจับหน้าสัมผัสดังรูป ??



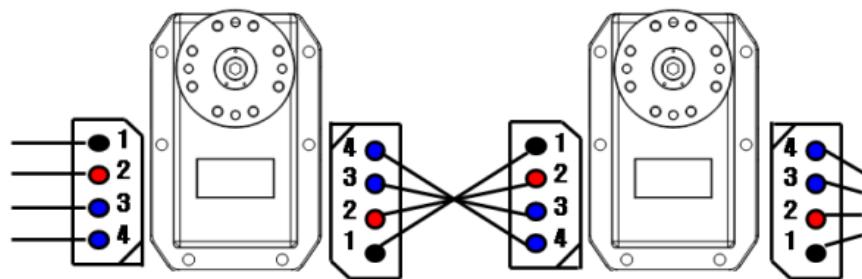
รูปที่ 3.11: Schematic ของวงจร Ground Contact Sensor



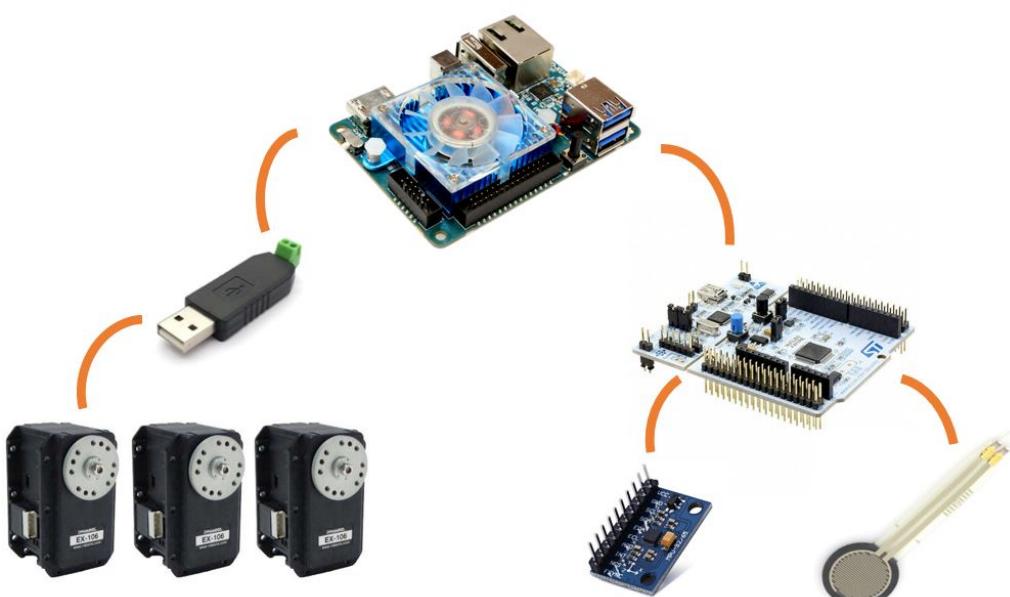
รูปที่ 3.12: แองเจร์ Ground Contact Sensor ที่ประกอบเสร็จแล้ว

### 3.3.4 การเชื่อมต่อดิจิตอลเซอร์โวและเซนเซอร์

โครงสร้างของหุ่นยนต์ขีวามโนยด์ UTHAI มีขาสองข้างทำให้เกิดองศาอิสระ 12 องศาอิสระ ผู้วิจัยจึงใช้ดิจิตอลเซอร์โวทั้งหมด 12 ตัว ดิจิตอลเซอร์โวทุกตัวเชื่อมต่อกันแบบสายโซลีเดช (daisy chain) ดังรูปที่ ?? ข้างหนึ่งของมอเตอร์ตัวแรกเชื่อมต่อกับแบตเตอรี่ 12V และอีกข้างต่อ กับ USB2RS485 เพื่อต่อไปยังตัวประมวลผลระดับสูง (Odroid) และเซนเซอร์ที่น่วยัดความแม่นิยม กับเซนเซอร์ตรวจจับหน้าสัมผัสที่พื้น เชื่อมต่อกับตัวประมวลผลระดับต่ำ (Nucleo F411RE) ดังรูปที่ ??



รูปที่ 3.13: การเชื่อมต่อ กันระหว่างดิจิตอลเซอร์โว



รูปที่ 3.14: การเชื่อมต่อระหว่างตัวรับสัญญาณ ตัวประมวลผล และตัวขับเคลื่อน

### 3.3.5 การตั้งค่าดิจิตอลเซอร์วอ



Roboplus 1.0

ก่อนที่จะเชื่อมต่อดิจิตอลเซอร์วอเข้ากับระบบประมวลผล จำเป็นที่จะต้องมีการตั้งค่า ID, Baudrate, Joint limited ของดิจิตอลเซอร์วอ โดยการตั้งค่าพารามิเตอร์ของดิจิตอลเซอร์วนั้น ผู้วิจัยจะใช้โปรแกรมที่มีชื่อว่า Roboplus 1.0 ซึ่งเป็นเครื่องมือจากบริษัท Robotis ที่จำหน่ายดิจิตอลเซอร์วนี้ โดยจะช่วยให้สามารถติดต่อกับดิจิตอลเซอร์วอ ตั้งค่าพารามิเตอร์ได้ แต่โปรแกรม Roboplus 1.0 นี้ใช้ได้เฉพาะใน Windows OS เท่านั้น ซึ่งสามารถดาวน์โหลดโปรแกรมได้จากหน้าเว็บไซต์ Robotis<sup>12</sup> เมื่อดาวน์โหลดโปรแกรมและติดตั้งเรียบร้อยแล้ว ให้เชื่อมต่อดิจิตอลเซอร์วอกับคอมพิวเตอร์ด้วย USB2RS485 และตั้งค่าพารามิเตอร์โดยทำการขั้นตอน ดังรูปต่อไปนี้

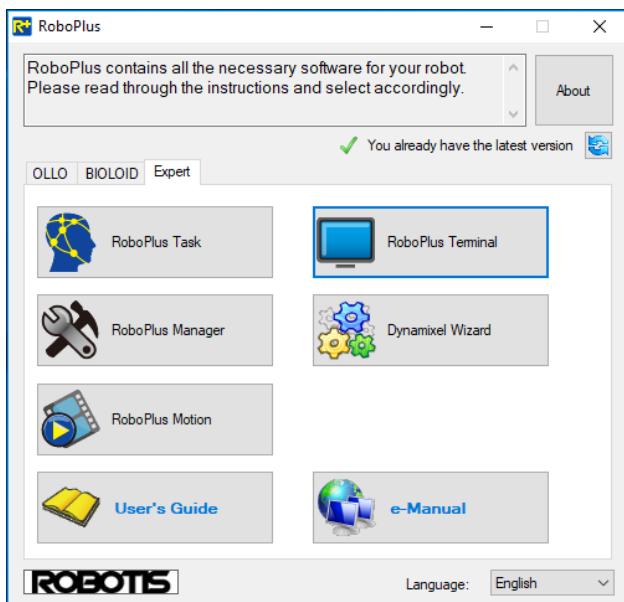


เชื่อมต่อดิจิตอลเซอร์วอเข้ากับคอมพิวเตอร์ด้วย USB2RS485

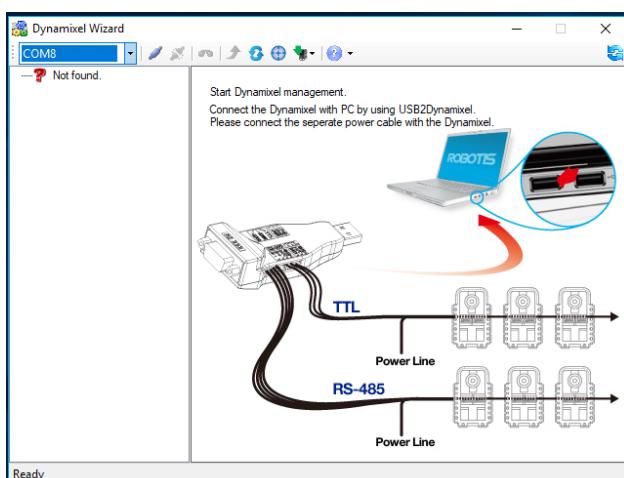


เปิดโปรแกรม Roboplus 1.0

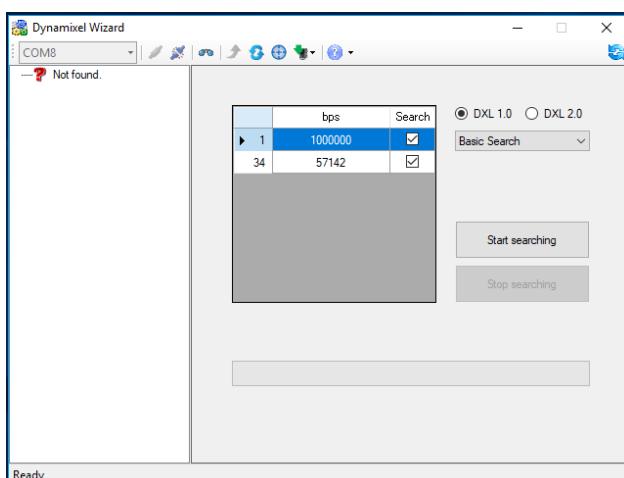
<sup>12</sup><http://www.robotis.us/roboplus1/>



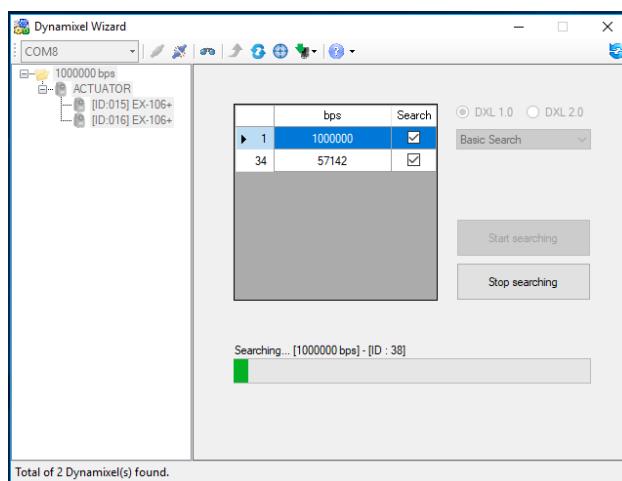
กดเข้าไปที่ Dynamixel Wizard



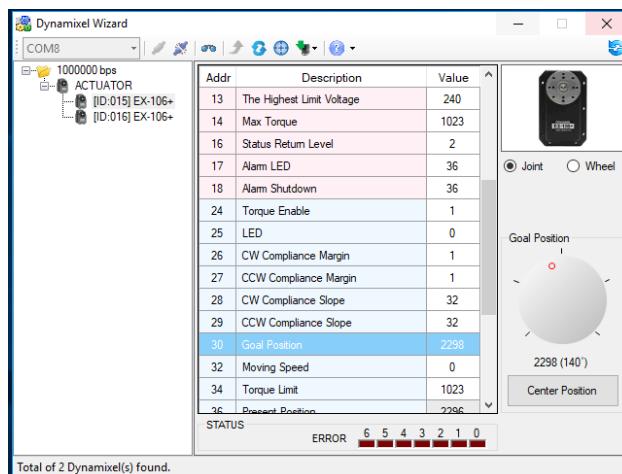
เลือก COM Port ให้ตรงกับ USB2RS485 จากนั้นกด Connect



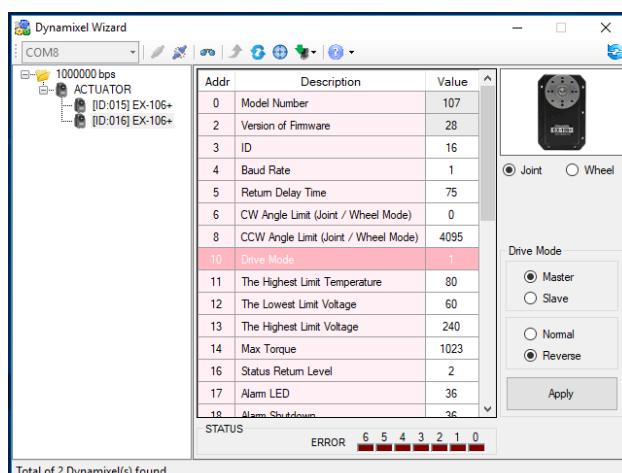
กดเลือกที่ช่อง 1Mbps และกด Start searching



เมื่อเทินทางด้านซ้ายมือผลลัพธ์ ID ของเตอร์ขึ้นมา หากขึ้นแล้วก็สามารถกด Stop Searching ได้



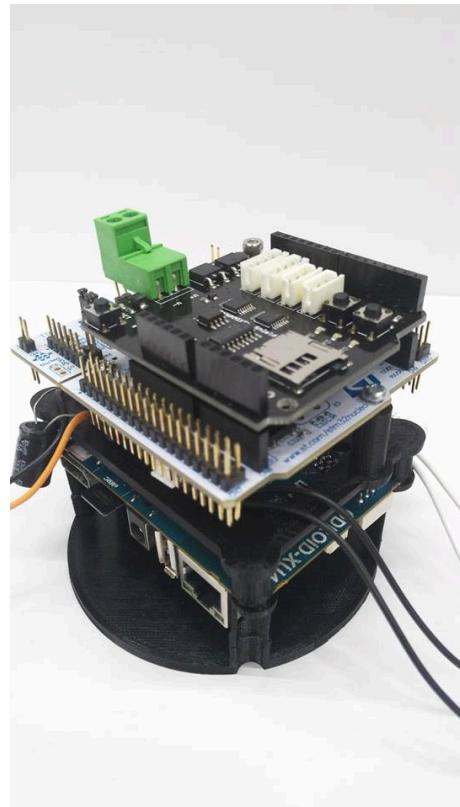
ทดสอบสั่งการมอเตอร์ที่ Addr 30 Goal position ว่าทิศทางถูกต้องหรือไม่



ถ้าทิศทางไม่ถูกต้องสามารถที่จะปรับได้ที่ Addr 10 Drive mode

### 3.3.6 การเชื่อมต่อหน่วยประมวลผลระดับสูงและระดับต่ำ

การเชื่อมต่อระหว่างหน่วยประมวลผล นั้นจะเชื่อมต่อกันผ่านสาย USB ชนิด Mini และส่งข้อมูลหากันผ่าน Serial โดยใช้ rosserial นอกจากจะมีหน่วยประมวลผลแล้วยังมีบอร์ดที่เอาไว้สำหรับควบคุม ดิจิตอลเซอร์วิโอ เพื่อเพื่อเอาไว้สำหรับควบคุมการเคลื่อนไหวของแขนหุ่นยนต์ รูปที่ ?? และทดสอบการสั่งการ ดังรูปที่ ??



รูปที่ 3.15: การเชื่อมต่อกันระหว่างตัวประมวลผล



รูปที่ 3.16: ตัวอย่างการใช้งานบอร์ดสั่งการดิจิตอลเซอร์วิโอ

### 3.4 การออกแบบโปรแกรมด้วย ROS

การออกแบบแบบโปรแกรมด้วย ROS ของหุ่นยนต์ชีวามโนยด์ UTHAI นั้น ผู้วิจัยได้วางการทำงานโดย เริ่มจาก การสร้างแบบจำลองเพื่อใช้ในการจำลองการทำงานของหุ่นยนต์ชีวามโนยด์ UTHAI โดยการจัดสร้างไฟล์ URDF ขึ้น ทดสอบการเคลื่อนไหวของหุ่นยนต์ในการแสดงผลด้วยภาพผ่านโปรแกรม RViz ทดสอบการสั่งการดิจิตอล เชอร์โว ทดสอบการอ่านตำแหน่งจากดิจิตอลเชอร์โว ทดสอบการส่งค่าตำแหน่งจากดิจิตอลเชอร์โวไปประมวลผล หากดูศูนย์กลางมวลในโปรแกรม MATLAB และเขียนโปรแกรมอ่านค่าเซนเซอร์ตรวจสอบการสัมผัสพื้น

#### 3.4.1 กำหนดพิกัดเฟรมให้กับหุ่นยนต์ชีวามโนยด์

การกำหนดเฟรมให้กับหุ่นยนต์ชีวามโนยด์ UTHAI นั้น ในวิทยานิพนธ์เล่มนี้ ผู้วิจัยจะใช้หลักตามของ ROS Enhancement Proposals (REPs)<sup>13</sup> ซึ่งการใช้หลักการนี้จะทำให้ การเขียนเป็นระบบระเบียบสามารถ หยิบเครื่องมือต่างๆ ที่สร้างขึ้นมาใช้งานร่วมกันได้ และช่วยทำให้เกิดความเข้าใจเวลาสื่อสาร

**base\_link** เป็นเฟรมที่ติดอยู่กับฐานของหุ่นยนต์ชีวามโนยด์ โดยจะติดตำแหน่งหรือมุมเอียงได้ โดย ส่วนใหญ่แล้วจะติดเฟรม base\_link ไว้ที่สังกะส์ของหุ่นยนต์

**base\_footprint** เป็นเฟรมที่แสดงว่าหุ่นยนต์อยู่ตรงไหนเมื่อเทียบกับโลก โดยจะมีระดับอยู่ที่จุดต่ำสุด ของฝ่าเท้า  $z = \min(l\_sole\_z, r\_sole\_z)$  โดย  $l\_sole\_z$  และ  $r\_sole\_z$  คือความสูงของฝ่าเท้า

**l\_wrist, r\_wrist** เป็นเฟรมที่บอกตำแหน่งและมุมเอียงของแขนซ้ายและขวาของหุ่นยนต์ชีวามโนยด์ โดยไม่ต้องคำนึงถึงการติดตั้งอุปกรณ์ใดๆเข้าไปที่ปลายแขนของหุ่นยนต์ชีวามโนยด์

**l\_gripper, r\_gripper** เป็นเฟรมที่บอกตำแหน่งและมุมเอียงของที่ปลายแขน (End effector) ถ้ามีอ จับอุปกรณ์อยู่ เฟรมนี้จะใช้ในการอ้างอิงตำแหน่งของอุปกรณ์นั้นๆ แต่ในวิทยานิพนธ์นี้ ไม่ได้ใช้แขนของหุ่นยนต์ ในการหยิบจับเครื่องมือหรือวัตถุ จึงไม่ได้ใช้เฟรมนี้

**l\_ankle, r\_ankle** เป็นเฟรมที่บอกตำแหน่งและมุมเอียงของขาซ้ายและขวาโดยไม่ได้คำนึงว่าจุดรับน้ำ หนักของตัวอยู่ที่ไหน

**l\_sole, r\_sole** เป็นเฟรมที่บอกตำแหน่งและมุมเอียงของขาซ้ายและขวาที่รองรับน้ำหนักตัวอยู่ โดยจะ บอกการฉายลงในระนาบของ  $X, Y$  ที่สัมผัสพื้นและ  $Z$  จะอยู่ระหว่างตัวเดียวกับพื้นสัมผัส

**l\_toe, r\_toe** เป็นเฟรมที่บอกตำแหน่งและมุมเอียงของปลายเท้าซ้ายและขวา

**gaze** เป็นเฟรมที่บอกตำแหน่งและมุมเอียงของหัว โดยการอ้างน้ำหนักของหัว โดยไม่ได้ สนใจเซนเซอร์ว่าจะติดตั้งอย่างไร แต่ในวิทยานิพนธ์นี้ หุ่นยนต์ชีวามโนยด์ UTHAI ไม่มีหัว จึงไม่ได้ใช้เฟรมนี้

**torso** เป็นเฟรมที่ติดอยู่กับลำตัวซึ่งล่างของหุ่นยนต์ชีวามโนยด์ โดยจะเป็นเฟรมที่ใช้เชื่อม ขา แขน หัว เข้าหากัน

<sup>13</sup><http://www.ros.org/reps/rep-0000.html>

### 3.4.2 การแปลงข้อมูลให้อยู่ในรูปแบบ URDF

เมื่อออคแบบโครงสร้างของหุ่นยนต์ชีวามาโนยด์ UTHAI ด้วยโปรแกรม Solidworks เสร็จแล้ว ต่อไปเป็นการนำเอาไฟล์ STL ออคมาเพื่อใช้ในการทำระบบจำลองการทำงานของหุ่นยนต์ โดยการใช้งานระบบจำลองเนื่องจากทำให้ผู้วิจัยสามารถที่จะเห็นการทำงานของหุ่นยนต์ชีวามาโนยด์ได้ การสร้างแบบจำลองโดยการใช้เครื่องมือที่มาพร้อมกับ ROS ด้วยโมดูล URDF

#### 3.4.2.1 แพกเกจ ROS สำหรับสร้างแบบจำลอง

ROS ได้ให้เครื่องมือที่ช่วยให้ สามารถที่จะสร้างแบบจำลองของหุ่นยนต์ชีวามาโนยด์สามมิติได้ เครื่องมือใน ROS ที่ชื่อว่า `robot_model` ภายในมีแพกเกจต่างๆที่ใช้สำหรับสร้างแบบจำลองของหุ่นยนต์สามมิติอยู่อย่างครบถ้วน ทำให้เราสามารถทำงานได้สะดวก และรวดเร็วมากขึ้น

`urdf` เป็นหนึ่งในหลายแพกเกจที่อยู่ใน `robot_model`, URDF เป็นไฟล์ XML ที่เอาไว้ใช้บอกลักษณะทางกายภาพของหุ่นยนต์ ซึ่งย่อมาจาก Unified Robot Description Format (URDF) การบอกโครงสร้างของหุ่นยนต์ด้วย URDF จะใช้การบอกเป็นโครงสร้างต้นไม้ของก้านต่อต่างๆในตัวหุ่นยนต์

`joint_state_publisher` เครื่องมือนี้มีประโยชน์มากในการสร้างแบบจำลองหุ่นยนต์ด้วย URDF เนื่องจากสามารถนำตำแหน่งของข้อต่อ มาแสดงเป็น GUI ได้ ทำให้เราสามารถเลื่อนๆหมุนๆไปมาได้ อีกทั้งยังสามารถใช้งานร่วมกับโปรแกรมแสดงผลภาพ RViz ได้

`robot_state_publisher` เป็นเครื่องมือที่ใช้ในการ publish ตำแหน่งของก้านต่อต่างๆในแบบจำลองของหุ่นยนต์ชีวามาโนยด์ออกมายาใน TF อีกทั้งยังให้ความสัมพันธ์ระหว่างเฟรมของหุ่นยนต์ได้ด้วย

`xacro` ย่อมาจาก XML Macros หรือเราสามารถเรียกอีกอย่างว่าเครื่องมือเสริมสำหรับ URDF ซึ่งลักษณะการเขียนเหมือนกับไฟล์ URDF แต่การเขียนนั้นจะสั้นกว่า อ่านง่ายกว่า และสามารถใช้เพื่อทำให้สร้างหุ่นยนต์ที่มีความซับซ้อนง่ายขึ้น สามารถแปลงไฟล์ xacro เป็น urdf ได้ถ้าต้องการ

#### 3.4.2.2 URDF

ในส่วนนี้จะเป็นการอธิบายระบบทางกลของหุ่นยนต์ชีวามาโนยด์เป็นไฟล์ที่ใช้ร่วมกับ ROS เพื่อที่จะสามารถนำไปใช้กับระบบจำลองการทำงานของหุ่นยนต์ในอนาคตได้ ในการอธิบายระบบทางกลนั้นผู้วิจัยได้ใช้ไฟล์ URDF ซึ่งใช้ภาษาการเขียนเป็น XML ในการบอกส่วนประกอบแต่ละส่วนของหุ่นยนต์

##### Link

ไฟล์ URDF แต่ละชิ้นส่วนของหุ่นยนต์เราจะเรียกว่า link และใน link จะประกอบไปด้วยส่วนย่อยๆ 3 ส่วนคือ `<inertia>` ที่เอาไว้บอกถึงค่าตัวแปรทางฟิสิกส์, `<visual>` ที่เอาไว้แสดงผลให้เราเห็น, `<collision>` ที่เอาไว้ตรวจสอบว่าหุ่นยนต์มีการชนกันกับสิ่งแวดล้อมใหม่ ดังรูปที่ ??

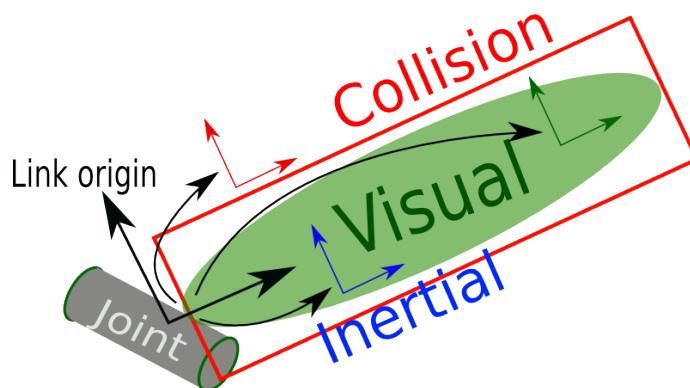
```

<link name="my_link">
  <inertia>
    <origin xyz="0 0 0.5" rpy="0 0 0"/>
    <mass value="1"/>
    <inertia ixx="100" ixy="0" ixz="0" iyy="100" iyz="0" izz="100"/>
  </inertia>
  <visual>
    <origin xyz="0 0 0" rpy="0 0 0"/>
    <geometry>
      <box size="1 1 1" />
    </geometry>
    <material name="Cyan">
      <color rgba="0 1.0 1.0 1.0"/>
    </material>
  </visual>
  <collision>
    <origin xyz="0 0 0" rpy="0 0 0"/>
    <geometry>
      <cylinder radius="1" length="0.5"/>
    </geometry>
  </collision>
</link>

```

รูปที่ 3.17: ตัวอย่าง link ใน urdf

ยังมีอีกหลายตัวที่ใช้ในการอธิบายแต่ละชิ้นส่วนของหุ่นยนต์ แต่ตัวอย่างเป็นเพียงแค่ส่วนหนึ่งเท่านั้น ในความเป็นจริงแล้วเราจะเขียน tags ต่างๆ ก็ตามที่เราต้องการ โดยใน URDF ไฟล์นั้นจะเอาไว้เก็บข้อมูลลักษณะเฉพาะของหุ่นยนต์เอาไว้ และยังสามารถใช้กับซอฟแวร์ตัวอื่นๆ อีกด้วย<sup>14</sup>



รูปที่ 3.18: การอธิบาย link ใน URDF ไฟล์

<sup>14</sup><http://wiki.ros.org/urdf>

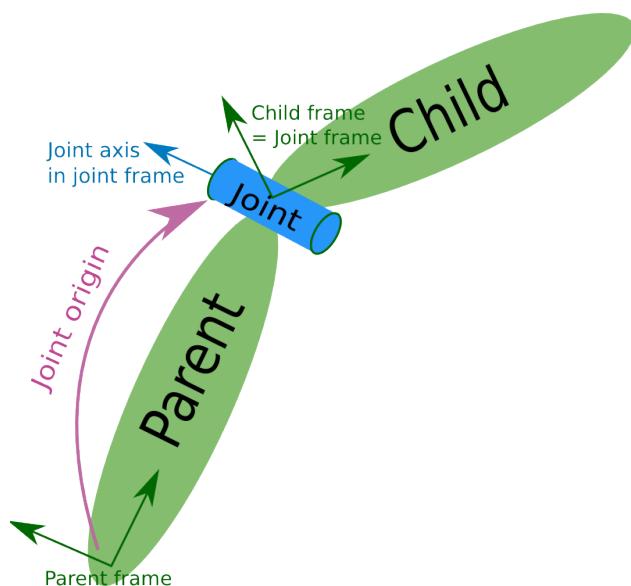
## Joint

อีกส่วนที่สำคัญสำหรับการสร้างไฟล์หุ่นยนต์ด้วย URDF ก็คือ Joint tag โดย tag นี้จะอธิบายถึงความสัมพันธ์ระหว่างก้านต่อสองอัน ส่วนนี้ไม่ได้มีเพียงแค่ทำข้อต่อให้เป็นแบบหมุนได้อย่างเดียว ยังมี Fix, Revolution, Linear และ Planar นอกเหนือจากนี้ เราสามารถที่จะเพิ่มองศาสูงสุดต่ำสุดของข้อต่อ รวมไปถึง dynamic properties ต่างๆ ตามที่เห็นดังรูปที่ ??

```
<joint name="my_joint" type="floating">
    <origin xyz="0 0 1" rpy="0 0 3.1416"/>
    <parent link="link1"/>
    <child link="link2"/>
    <calibration rising="0.0"/>
    <dynamics damping="0.0" friction="0.0"/>
    <limit effort="30" velocity="1.0" lower="-2.2" upper="0.7"/>
    <safety_controller k_velocity="10" k_position="15"
        soft_lower_limit="-2.0" soft_upper_limit="0.5"/>
</joint>
```

รูปที่ 3.19: ตัวอย่าง joint ใน urdf

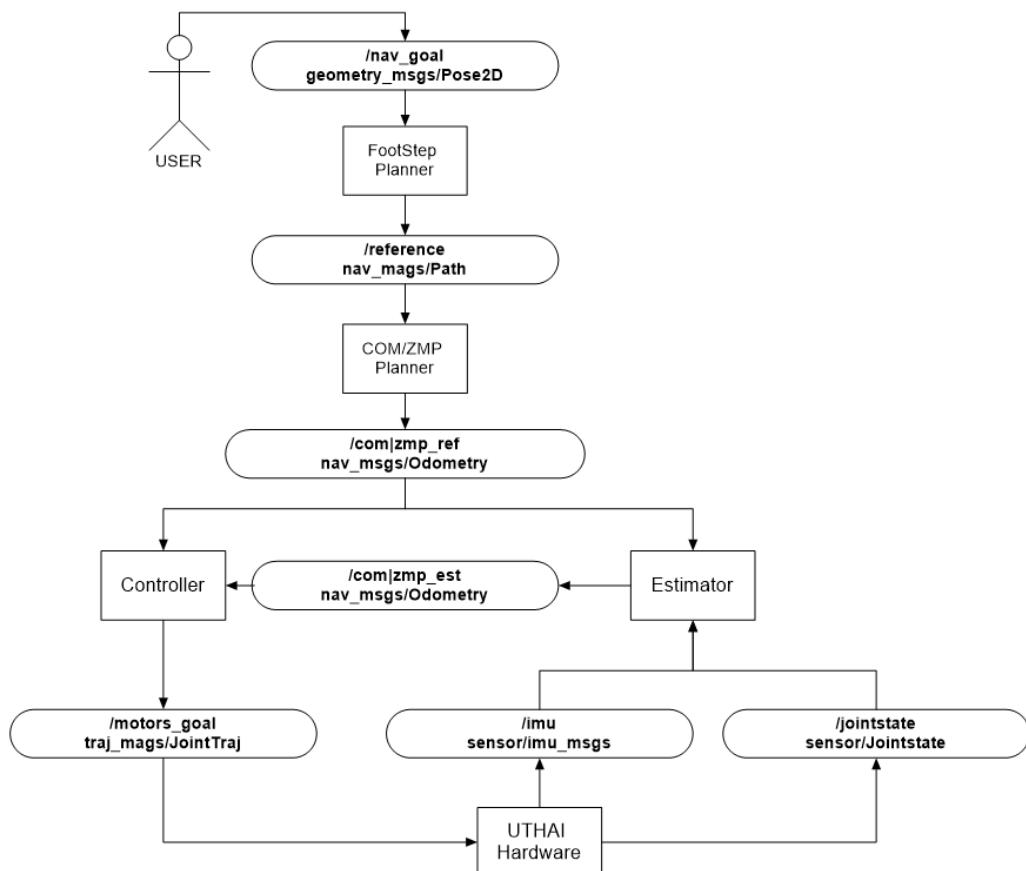
เมื่อเรานำ Joint และ Link มารวมกันเราจะต้องพิจารณาว่ามีวารูปแบบเป็นไปตามรูปที่ ?? โดยจะมีระบุระยะเวลาที่แกนของแต่ละข้อต่อ กับ ก้านต่อ ชึ้นส่วนแรกของการสร้างไฟล์ URDF จะมีชื่อว่า base\_link และเฟรม origin จะเป็นเฟรมอ้างอิง เมื่อเราต่อ Joint เข้ากับ Link จะเรียก ก้านต่อ ที่นำมาติดว่า parent โดยเฟรม origin ของข้อต่อจะอยู่จุดเดียวกับเฟรม origin ของ ก้านต่อ ในสถานะเดียวกัน ก้านต่อที่นำมาต่อจากข้อต่อ เราจะเรียกว่า child และเฟรม origin ของ ก้านต่อ child จะอยู่ที่จุดเดียวกับเฟรม origin ของข้อต่อ



รูปที่ 3.20: การอธิบาย Joint ใน URDF ไฟล์

### 3.4.3 โครงสร้างการติดต่อสื่อสารระหว่าง Node ใน ROS

การติดต่อสื่อสารกันภายใน ROS นั้นจะใช้การส่ง message หากัน ซึ่ง message แต่ละตัวก็จะใช้ในงานที่ต่างกัน ตามระบบที่ต้องการส่ง จากรูปที่ ?? เป็นโครงสร้างการส่งข้อมูลหกันของหุ่นยนต์หิวามานอยด์ ที่ผู้วิจัยได้ออกแบบไว้ โดยเริ่มจากผู้ใช้งานส่งตำแหน่งที่หุ่นยนต์จะต้องเดินไปเป็น Node ที่ทำการคำนวณและสร้างตำแหน่งการวางเท้าของหุ่นยนต์ และหลังจากนั้นจะส่งข้อมูลออกไปเป็น Path เส้นทางไปยัง Node ที่ทำการค้นหาตำแหน่งของ com, zmp ของหุ่นยนต์ เพื่อทำการควบคุมและส่งการหุ่นยนต์ต่อไป



รูปที่ 3.21: การติดต่อสื่อสารระหว่าง Node

#### การบอกตำแหน่งและมุมเอียง

การบอกตำแหน่งใน 3 มิติ Point คือการบอก  $x, y, z$  และการบومุมเอียงจะใช้ Quaternion ในการบอกโดยใช้ตัวแปรสีตัว คือ  $x,y,z,w$  หากนำทั้งสองมารวมกันเราจะเรียกว่า Pose

geometry_msgs/Point	
float64	x
float64	y
float64	z

ตารางที่ 3.4: Message Geometry Point

geometry_msgs/Quaternion	
float64	x
float64	y
float64	z
float64	w

ตารางที่ 3.5: Message Geometry Quaternion

geometry_msgs/Pose	
geometry_msgs/Point	position
geometry_msgs/Quaternion	orientation

ตารางที่ 3.6: Message Geometry Pose

### การบอกรความเร็วเชิงเส้นและเชิงมุม

การบอกรความเร็วเชิงเส้นใน 3 มิติ คือการบอกรความเร็วตามแนวแกน  $x, y, z$  และการบอกรความเร็วเชิงมุม คือการบอกรความเร็วการหมุนรอบแกน  $x, y, z$  หากนำทั้งสองมาร่วมกันเราจะเรียกว่า Twist

geometry_msgs/Vector3	
float64	x
float64	y
float64	z

ตารางที่ 3.7: Message Geometry Vector3

geometry_msgs/Twist	
geometry_msgs/Vector3	linear
geometry_msgs/Vector3	angular

ตารางที่ 3.8: Message Geometry Twist

### การบอกรตำแหน่งและความเร็ว

หากนำทั้งสองมาร่วมกันจะรู้ว่า ตำแหน่ง (Pose) และความเร็ว (Twist) เราจะเรียกว่า Odometry แต่ที่เพิ่มเข้ามาคือ Covariance ซึ่งอาจทำให้เกิดความสับสนได้

nav_msgs/Odometry	
std_msgs/Header	header
geometry_msgs/PoseWithCovariance	pose
geometry_msgs/TwistWithCovariance	twist

ตารางที่ 3.9: Message Navigation Odometry

### ตำแหน่งของหุ่นยนต์

การบอกร่องรอยของหุ่นยนต์บนระนาบ 2 มิติ คือการบอก  $x$ ,  $y$  และ  $\theta$  การบอกนั้นจะบอกว่าตำแหน่งที่หุ่นยนต์อยู่นั้นอยู่ตรงไหนหากเทียบกับแผนที่ รวมไปถึงตำแหน่งของหุ่นยนต์ที่ต้องการจะเดินไปด้วย ซึ่งอ้างอิงมาจากการบอกร่องรอยที่เริ่มต้นของแผนที่

geometry_msgs/Pose2D	
float64	$x$
float64	$y$
float64	$\theta$

ตารางที่ 3.10: Message Geometry Pose2D

### ตำแหน่งการวางแผนทางของหุ่นยนต์

การจะให้หุ่นยนต์นำเท้าไปวางในตำแหน่งที่เราต้องการจากที่ได้จากการคำนวณนั้น จะต้องบอกร่องรอยและบอกมุมเอียงของจุดที่จะไป จากการสร้างจะได้เป็นรายการของเท้าซ้ายและขวา โดยอิงจาก ตารางที่ ??

nav_msgs/Path	
std_msgs/Header	header
geometry_msgs/PoseStamped[]	poses

ตารางที่ 3.11: Message Navigation Path

geometry_msgs/PoseStamped	
std_msgs/Header	header
geometry_msgs/Pose	pose

ตารางที่ 3.12: Message Geometry PoseStamped

### ตำแหน่งจุดศูนย์กลางมวลของหุ่นยนต์

ใน Message นี้ใช้อยู่ 2 ที่คือ Message ที่ได้จากการวางแผนของ Node CoM Planner และ Node CoM Estimator โดยทั้งสองจุดใช้ Message เมื่อกันส่งไปยัง Controller เพื่อควบคุมท่าทางต่างๆของหุ่นยนต์ต่อไป Message ที่ใช้คือ Message จากตารางที่ ??

nav_msgs/Odometry	
std_msgs/Header	header
geometry_msgs/PoseWithCovariance	pose
geometry_msgs/TwistWithCovariance	twist

Message Navigation Odometry

### การควบคุมข้อต่อของหุ่นยนต์

ในการควบคุมข้อต่อแต่ละข้อของหุ่นยนต์ชีวามาโนyd'n'จะใช้ Message trajectory\_msgs/JointTrajectory ซึ่งสามารถส่ง ตำแหน่ง ความเร็ว ความเร่ง และ แรงบิด ไปได้ ทำให้หากต้องการเปลี่ยนระบบใหม่สามารถทำได้โดยง่าย

trajectory_msgs/JointTrajectory	
std_msgs/Header	header
string[]	joint_names
trajectory_msgs/JointTrajectoryPoint[]	points

ตารางที่ 3.13: Message Trajectory JointTrajectory

trajectory_msgs/JointTrajectoryPoint	
float64[]	positions
float64[]	velocities
float64[]	accelerations
float64[]	effort
duration	time_from_start

ตารางที่ 3.14: Message Trajectory JointTrajectoryPoint

### ค่าเซนเซอร์ข้อต่อของหุ่นยนต์

ที่ข้อต่อของหุ่นยนต์ชีวามาโนyd'm'i' มีเซนเซอร์ที่เอาไว้ใช้ในการอ่านค่าตำแหน่ง ความเร็ว และแรง อยู่ด้วย เราสามารถที่จะใช้ Message sensor\_msgs/JointState สำหรับอ่านค่าตำแหน่ง ความเร็ว แรง ของตัวขับเคลื่อนแล้วส่งให้ Estimator Node ได้

sensor_msgs/JointState	
std_msgs/Header	header
float64[]	position
float64[]	velocity
float64[]	effort

ตารางที่ 3.15: Message Sensor JointState

### ค่าเซนเซอร์ฝ่าเท้าของหุ่นยนต์

ที่ฝ่าเท้าของหุ่นยนต์ชีวามาโนyd'm'i' มีเซนเซอร์ที่เอาไว้ใช้ในการอ่าน แรงกดที่ฝ่าเท้า ใช้ในการเอามากกว่า เท้าสัมผัสพื้นหรือไม่

geometry_msgs/Wrench	
geometry_msgs/Vector3	force
geometry_msgs/Vector3	torque

ตารางที่ 3.16: Message Geometry Wrench

### ค่าเซนเซอร์ IMU ของหุ่นยนต์

เซนเซอร์ IMU เป็นเซนเซอร์ที่เอาไว้ใช้ในการวัด ความเร็วเชิงมุม และ ความเร่งเชิงเส้น หากนำทั้งคู่มารวมกันจะสามารถที่จะแปลงให้วัดมุมอิริยาบถของเซนเซอร์ได้ โดยจะใช้ Message std\_msgs/Imu ในการส่งให้ Node Estimator จากตัวหุ่นยนต์

sensor_msgs/Imu	
std_msgs/Header	header
geometry_msgs/Quaternion float64[9]	orientation
geometry_msgs/Vector3 float64[9]	orientation_covariance
geometry_msgs/Vector3 float64[9]	angular_velocity
	angular_velocity_covariance
	linear_acceleration
	linear_acceleration_covariance

ตารางที่ 3.17: Message Sensor Imu

sensor_msgs/MagneticField	
std_msgs/Header	header
geometry_msgs/Vector3 float64[9]	magnetic_field
	magnetic_field_covariance

ตารางที่ 3.18: Message Sensor MagneticField

## 3.5 การออกแบบระบบพื้นฐาน

### 3.5.1 วางแผนสร้างของระบบพื้นฐาน

โครงสร้างระบบพื้นฐานของหุ่นยนต์อิวามาโนยด์ UTHAI เป็นการวางแผนให้ผู้วิจัยสามารถที่จะช่วยกันพัฒนาได้ และเพื่อทำให้มีความเป็นระบบระเบียบ ง่ายต่อการแก้ไข ปรับปรุง ซึ่งผู้วิจัยได้ออกแบบส่วนการทำงาน ต่างๆออกเป็นสามส่วน ตามหน้าที่การทำงาน คือ

#### Hardware Devices

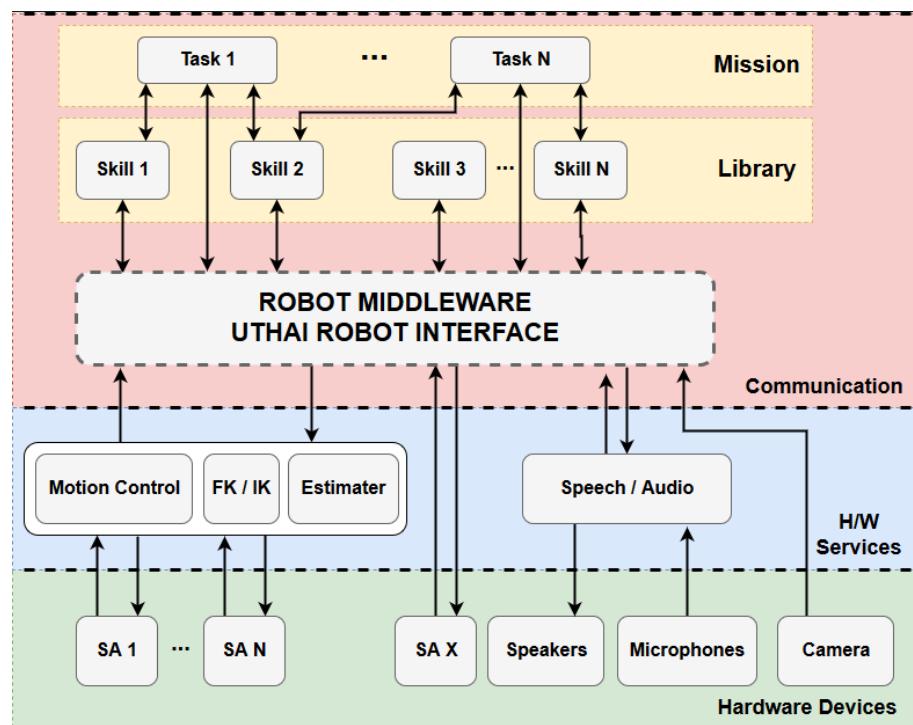
เป็นส่วนของอุปกรณ์ฮาร์ดแวร์ที่ใช้งานกับหุ่นยนต์อิวามาโนยด์ ยกตัวอย่างเช่น ตัวขับเคลื่อน กล้อง ไมโครโฟน ลำโพง หรืออื่นๆ ซึ่งในส่วนนี้สามารถเปลี่ยนฮาร์ดแวร์จริงๆเป็นระบบจำลองได้

#### Hardware Services

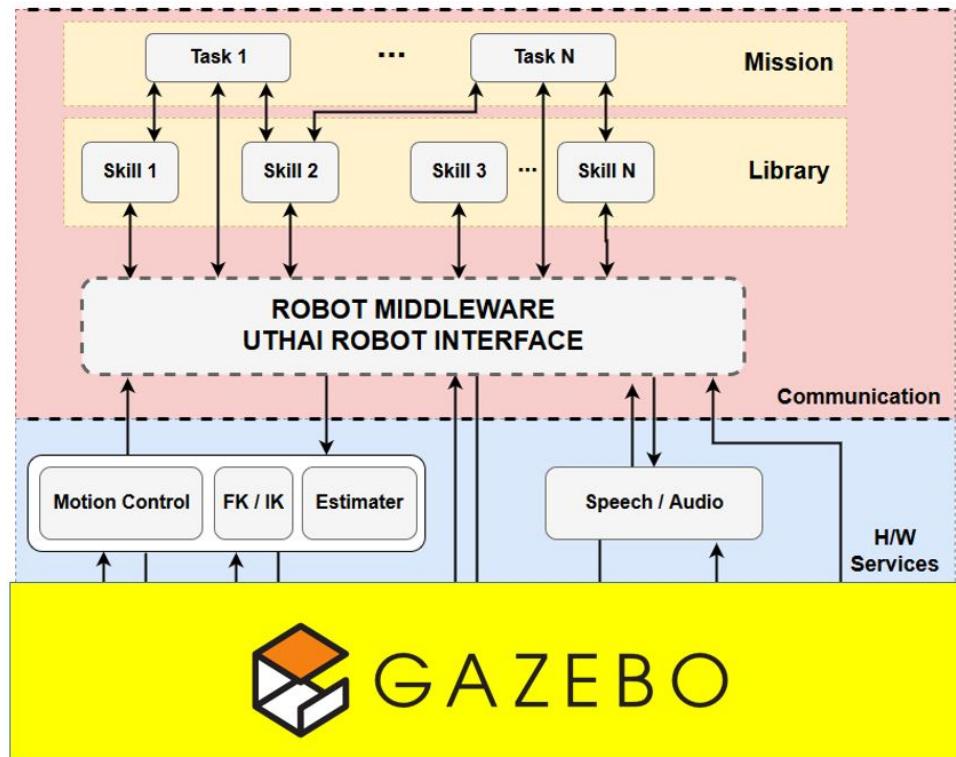
เป็นส่วนของซอฟต์แวร์ที่ช่วยทำให้อุปกรณ์ฮาร์ดแวร์สามารถติดต่อสื่อสารกับระบบพื้นฐานของหุ่นยนต์อิวามาโนยด์ UTHAIได้ ซึ่งจะช่วยทำให้การส่งข้อมูลเป็นมาตรฐานเดียวกัน

#### Communication

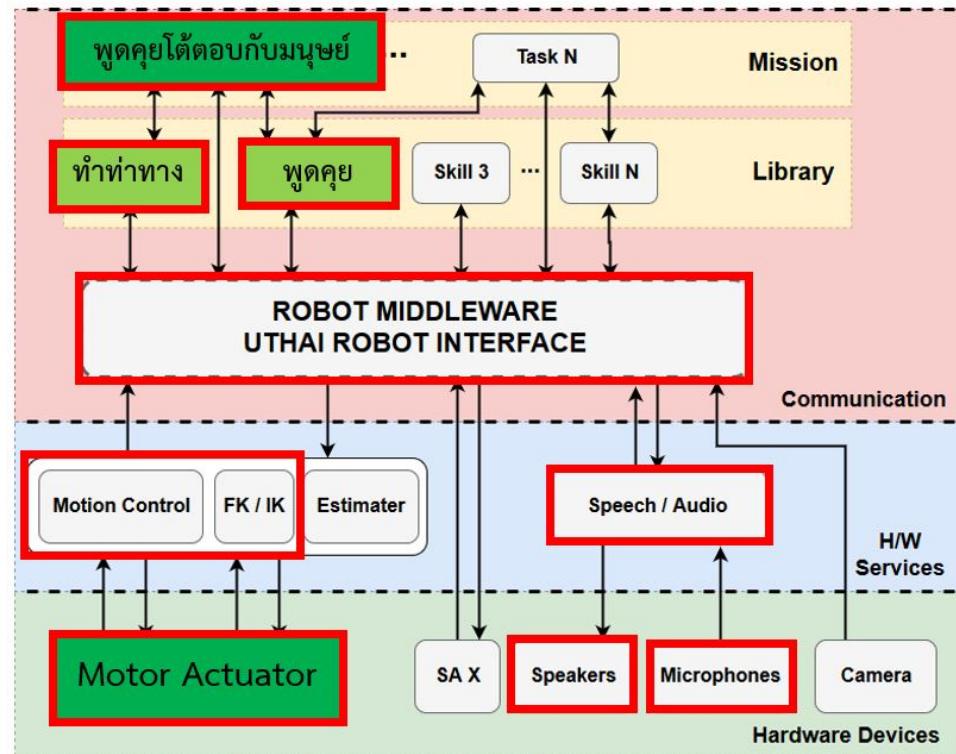
เป็นส่วนของการติดต่อสื่อสารระหว่างซอฟต์แวร์กับฮาร์ดแวร์ผ่านระบบพื้นฐานของหุ่นยนต์อิวามาโนยด์ ซึ่งจะอยู่ภายใต้การให้ข้อมูลทุกอย่างสามารถเชื่อมต่อกันได้



รูปที่ 3.22: โครงสร้างพื้นฐานของหุ่นยนต์อิวามาโนยด์ UTHAI



รูปที่ 3.23: ภาพการเปลี่ยนส่วนของชาร์ดแวร์เป็นระบบจำลอง

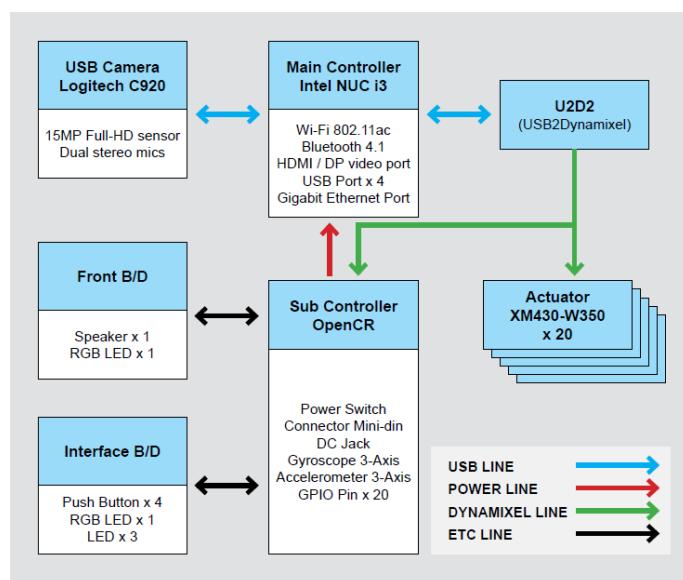


รูปที่ 3.24: ตัวอย่างการนำโครงสร้างพื้นฐานไปประยุกต์ใช้ ในแอปพลิเคชันการพูดคุยโดยตัดออกกับมนุษย์

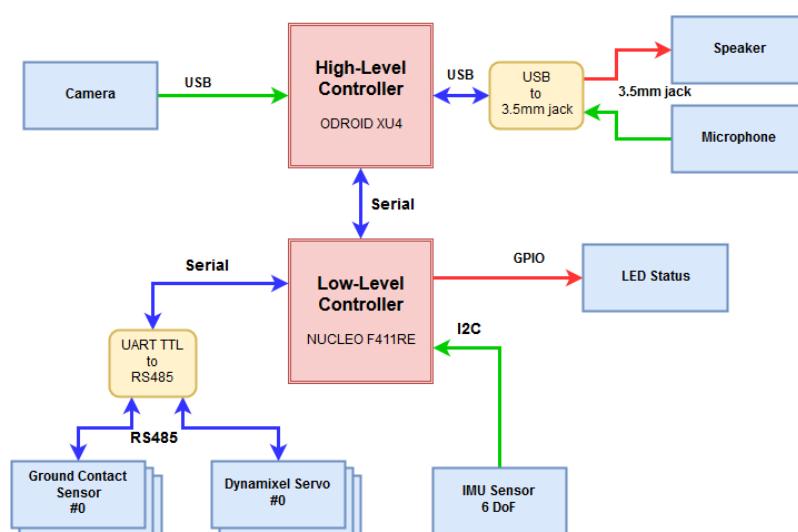
### 3.5.2 ออกแบบสถาปัตยกรรมของหุ่นยนต์

หลักการออกแบบสถาปัตยกรรมของหุ่นยนต์ชีวามโนยด์ UTHAI จะออกแบบระบบให้อยู่บนระบบพื้นฐาน ROS เนื่องจากการใช้กรอบการทำงานที่มีประสิทธิภาพ และความยืดหยุ่นสูง จะช่วยทำให้สามารถปรับเปลี่ยนระบบการควบคุมของหุ่นยนต์ชีวามโนยด์ได้ง่ายและรวดเร็ว การออกแบบหน่วยประมวลผลนั้นมีลักษณะรูปแบบ ดังที่กล่าวไว้ในบทที่ 2 สำหรับในงานวิจัยครั้งนี้ผู้วิจัยได้ศึกษาและพอกกว่าสถาปัตยกรรมที่เหมาะสมกับหุ่นยนต์ชีวามโนยด์ UTHAI จะมีลักษณะใกล้เคียงกับหุ่นยนต์ชีวามโนยด์ Robotis OP3 ดังรูปที่ ?? ดังนั้นแล้วผู้วิจัยจึงได้แบ่งการประมวลผลออกเป็น 2 ส่วนคือ

- 1 หน่วยประมวลผลควบคุมระดับสูง (High Level Controller)
- 2 หน่วยประมวลผลควบคุมระดับต่ำ (Low Level Controller)



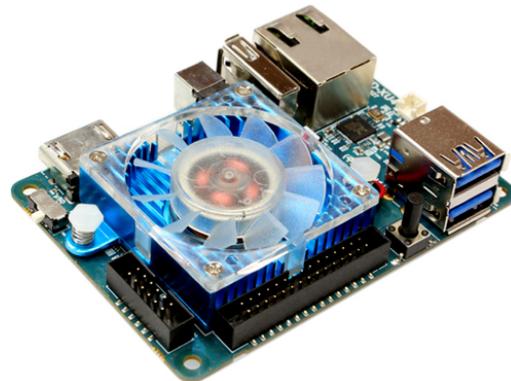
รูปที่ 3.25: สถาปัตยกรรมของหุ่นยนต์ Robotis OP3



รูปที่ 3.26: สถาปัตยกรรมของหุ่นยนต์ชีวามโนยด์ UTHAI

### 3.5.2.1 หน่วยประมวลผลควบคุมระดับสูง (High level controller)

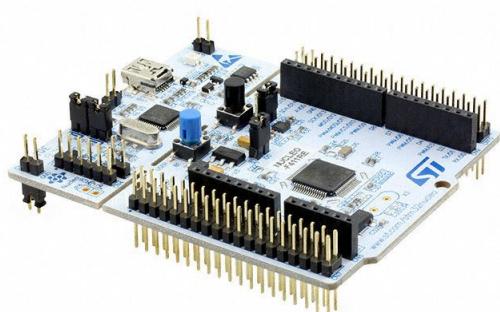
ระบบควบคุมหลักของหุ่นยนต์ชีวามโนยด์ UTHAI นั้นจะอยู่ที่หน่วยประมวลผลขั้นสูง ใช้เป็นบอร์ดคอมพิวเตอร์ ODROID-XU4 ตัวประมวลผลหลักนี้ ทำหน้าที่ในการคำนวณเส้นทางการเดิน ทำให้หุ่นยนต์มีเสถียรภาพในการเดิน ตรวจการขัดกันของโครงสร้างของหุ่นยนต์ รวมไปถึงรับค่าข้อมูลตำแหน่ง ความเร็วจากข้อต่อ หลังจากนั้นจะทำการนำค่าทั้งหมดที่ได้จากการคำนวณ มาแปลงให้อยู่ในรูปของชุดข้อมูล แล้วส่งออกไปให้ระบบกลาง (ROS) ในการส่งต่อไปให้อุปกรณ์อื่นต่อไป



รูปที่ 3.27: บอร์ดคอนโทรลเลอร์ Odroid XU4

### 3.5.2.2 หน่วยประมวลผลควบคุมระดับต่ำ (Low level controller)

ระบบควบคุมขั้นต่ำเป็นหน่วยประมวลผลที่รองรับมาจาก บอร์ดคอมพิวเตอร์ โดยใช้บอร์ดไมโครคอนโทรลโลร์ Nucleo F411RE เป็นหน่วยประมวลผลขั้นต่ำ สำหรับในการติดต่อกับอุปกรณ์อิเล็กทรอนิกส์ต่างๆ ที่อยู่ภายนอกตัวของหุ่นยนต์ เช่น ค่าเซนเซอร์ที่ไฟเท้าซึ่งสามารถบอกได้ว่าควรใช้สมการไหนในการคำนวณพลวัต หรือค่าของเซนเซอร์หน่วยวัดความเร็วym ความสำคัญมาก ในการทำให้หุ่นยนต์ชีวามโนยด์เดินได้อย่างมีเสถียรภาพ เมื่ออ่านค่าเซนเซอร์ต่างๆได้แล้ว หน่วยประมวลผลขั้นต่ำจะนำค่าที่ได้จากการอ่านเซนเซอร์เหล่านี้แปลงให้อยู่ในลักษณะของชุดข้อมูล แล้วส่งออกไปในระบบกลาง (ROS) นอกจากนี้หน่วยประมวลผลขั้นต่ำยังทำหน้าที่รับค่าคำสั่งมาจากระบบกลาง ในการสั่งงานให้หุ่นยนต์มีท่าทางต่างๆได้



รูปที่ 3.28: บอร์ดคอนโทรลเลอร์ Nucleo F411RE

### 3.5.3 จัดทำคู่มือและเอกสารการใช้งาน

คู่มือจะเป็นส่วนที่ผู้มาพัฒนาต่อยอดสามารถที่จะอ่านทำความเข้าใจได้ โดยจะเขียนให้อยู่ในรูปของไฟล์ Markdown (.md) และเก็บเอาไว้ในเว็บไซต์ GitHub ซึ่งเป็นแหล่งรวม Source code ออนไลน์ สามารถเข้าไปดาวน์โหลดไฟล์ลงเครื่องผู้ใช้ แล้วทำการติดตั้งใช้งานได้เลย อีกทั้งผู้ใช้งานสามารถส่ง Code ของตัวเองเข้าระบบ เพื่อช่วยพัฒนาและเพิ่มประสิทธิภาพในการทำงานของซอฟต์แวร์ของหุ่นยนต์ได้

#### 3.5.3.1 รายการวัสดุที่ใช้ในการทำหุ่นยนต์ชีวามานอยด์ UTHAI

รายการ	จำนวน (หน่วย)	บาท/หน่วย	ราคารวม(บาท)
===== Processing Unit	-	-	-
Odroid XU4 Embedded Computer	1	3800	3800
Shifter Shield for Odroid XU4	1	1000	1000
===== Sensor	-	-	-
Force sensitive Resistor	8	300	2400
Electronic Component	1	2000	2000
MPU9255 9 Axis IMU Module	1	500	500
===== Structure	-	-	-
อุปกรณ์ส่งกำลัง	1	3000	3000
ค่าวัสดุ เช่น Filament 3D printer , Carbon Fiber	1	8000	8000
สปริง	14	50	700
อุปกรณ์สิ้นเปลือง เช่น กระดาษทราย ฯลฯ	1	1000	1000
===== อุปกรณ์เสริม Motor Dynamixel	-	-	-
Frame สำหรับต่อพ่วงมอเตอร์	4	2000	8000
Horn Bearing	4	1400	5600
อุปกรณ์จ่ายพลังงาน	-	-	-
Power Supply	1	2000	2000
Battery Li-Po 4 cell	1	3000	3000
===== รวม	-	-	48000

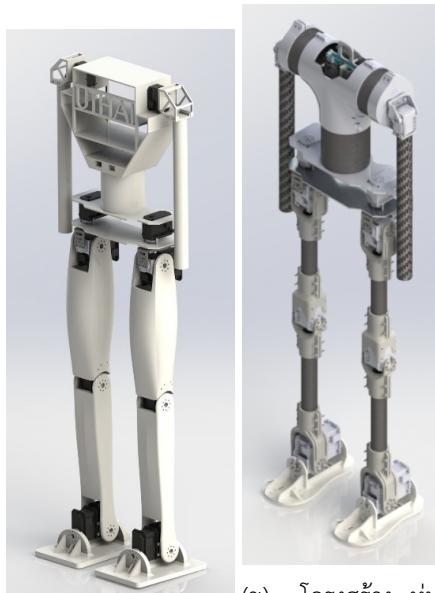
ตารางที่ 3.19: ตารางแสดงรายการของวัสดุต่าง ๆ

ใช้สำหรับแจกแจงค่าใช้จ่ายเบื้องต้นเท่านั้น ไม่สามารถใช้อ้างอิงบประมาณแบบละเอียดได้

## บทที่ 4

### ผลการดำเนินงาน

#### 4.1 โครงสร้างของหุ่นยนต์



(ก) โครงสร้างหุ่นยนต์ใน ยนต์ ใน โปรแกรม 3  
โปรแกรม 3 มิติ(ครั้งที่ 1) มิติ(ครั้งที่ 2)

รูปที่ 4.1: รูปการออกแบบโครงสร้างหุ่นยนต์ในโปรแกรม 3 มิติ

โครงสร้างของหุ่นยนต์อิวามานอยด์ UTHAI จะแบ่งออกเป็น 2 ส่วนคือ ส่วนท่อนบนและส่วนท่อนล่างโดย ส่วนท่อนบนจะประกอบไปด้วย เอว ลำตัว แขน และท่อนล่างจะประกอบไปด้วย สะโพก ขา น่อง ฝ่าเท้า ในการ เลือกใช้วัสดุนั้นได้แสดง ดังตารางที่ ??

ชิ้นส่วน	วัสดุที่ใช้ขึ้นรูป
แขน	ท่อคาร์บอนไฟเบอร์ ขนาด 30 มม.
ลำตัว	เครื่องพิมพ์ 3 มิติ โดยใช้วัสดุ PLA
เอว	ท่อคาร์บอนไฟเบอร์ ขนาด 88 มม.
สะโพก	อลูมิเนียมอลลอลอยพับ
น่อง	เครื่องพิมพ์ 3 มิติ โดยใช้วัสดุ PLA
ขา	เครื่องพิมพ์ 3 มิติ โดยใช้วัสดุ PLA
ฝ่าเท้า	เครื่องพิมพ์ 3 มิติ โดยใช้วัสดุ PLA

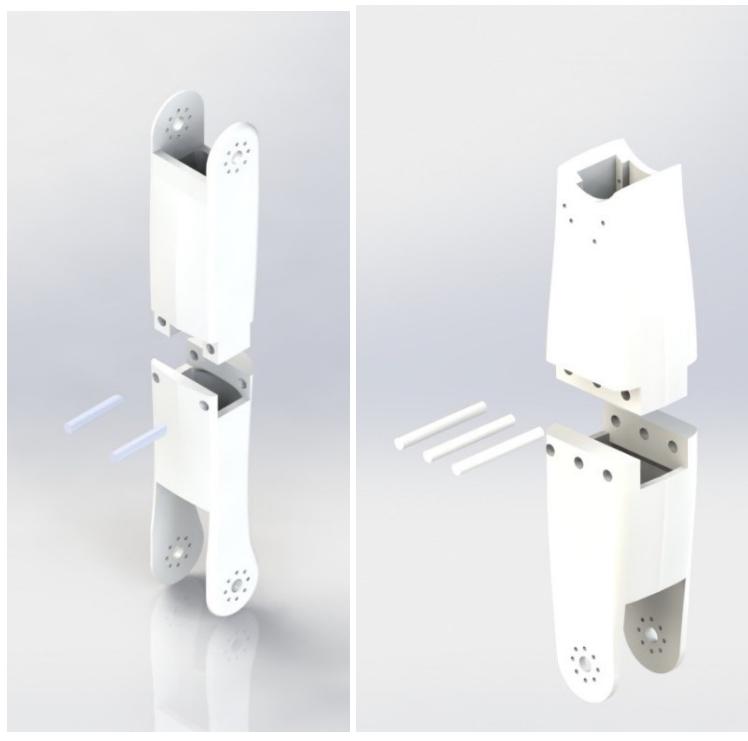
ตารางที่ 4.1: ตารางแสดงวัสดุที่ใช้ขึ้นรูปหุ่นยนต์อิวามานอยด์ UTHAI

ซึ่งในการออกแบบนี้ได้แบ่งการออกแบบออกเป็น 2 ครั้งด้วยกันเนื่องจากว่าเกิดปัญหาในด้านน้ำหนักของ การออกแบบครั้งที่ 1 ที่มากเกินไป จึงทำการปรับปรุงใหม่เพื่อให้มีน้ำหนักที่เบามากขึ้นกว่าเดิม แต่ยังคงคงถึง ความแข็งแรงของโครงสร้างให้ไม่น้อยไปกว่าเดิม

#### 4.1.1 การออกแบบขา

##### 4.1.1.1 การออกแบบโครงสร้างส่วนขาครั้งที่ 1

การออกแบบโครงสร้างส่วนขาของหุ่นยนต์ชีวามโนยด์ ได้ออกแบบโดยคำนึงถึงการขึ้นรูปด้วยเครื่องพิมพ์สามมิติ แต่เนื่องจากว่าเครื่องพิมพ์สามมิติที่ใช้ในการผลิตนั้นมีขนาดที่เล็กกว่าขนาดที่จะพิมพ์จริงจึงต้องทำการแยกส่วนของขาออก เป็นจำนวน 2 ส่วนในแต่ละในก้านต่อของขาท่อนบนและขาท่อนล่าง และหลังจากนั้นใช้การยึดขึ้นส่วนด้วยการตอกสลักเพื่อยึดติดขึ้นส่วนเข้าด้วยกัน เพื่อให้มีความแข็งแรงมากกว่าการต่อแบบทั่วไป ดังรูปที่ ?? เมื่อทำการพิมพ์ชิ้นงานส่วนขาท่อนบนและท่อนล่าง օกมาจะได้น้ำหนักของชิ้นงานตามตาราง ??



(ก) โครงสร้างส่วนหน้าแข้ง

(ข) โครงสร้างส่วนต้นขา

รูปที่ 4.2: รูปการออกแบบส่วนขาของหุ่นยนต์ชีวามโนยด์ UTHAI

ชิ้นส่วน	น้ำหนัก (กรัม)
ต้นขา	263
หน้าแข้ง	204

ตารางที่ 4.2: ตารางแสดงน้ำหนักของชิ้นส่วนขา

จากการทดสอบความสามารถในการเคลื่อนที่ พบร่วด้วยขับเคลื่อนสามารถเคลื่อนที่เข้าตำแหน่งได้ถูกต้องตามมุ่ง ที่ป้อนเข้าไปให้ระบบ แต่หากทำให้ชิ้นส่วนของขาเคลื่อนที่ด้วยถีปีกลับสูงและด้วยความเร็วที่มาก จะทำให้ตัวขับเคลื่อนเกิดการโอเวอร์โลด ซึ่งมีผลทำตัวขับเคลื่อนหยุดการทำงาน ซึ่งต้องทำการปิดเปิดตัวขับเคลื่อนใหม่

## ผลการทดสอบ

จากการทดสอบความแข็งแรงของชิ้นงาน พบร่วมกันมีความแข็งแรงเพียงพอที่จะทำให้ตัวขับเคลื่อนมีค่าแรงบิดเป็นค่าแรงบิดสูงสุด(Stall Torque) แล้วทำให้ชิ้นงานไม่เกิดความเสียหาย

จากการทดสอบระยะเวลาการทำงานของตัวขับเคลื่อน ด้วยการเขียนโปรแกรมให้ตัวขับเคลื่อน เคลื่อนที่ไปกลับ สลับตำแหน่งไปเรื่อยๆอย่างต่อเนื่อง เป็นเวลา 20 นาที พบร่วมกันว่า ตัวขับเคลื่อนทำงานได้เป็นปกติ

### ปัญหาที่พบในการออกแบบครั้งที่ 1

เนื่องจากว่าเป้าหมายของการสร้างหุ่นยนต์ตัวนี้ให้มีน้ำหนักที่เบา (น้อยกว่า 5 กิโลกรัม) จึงพบปัญหาว่า น้ำหนักของส่วนขาที่ต้องออกแบบมานั้นมีน้ำหนักมากเกินกว่าของหุ่นยนต์กัน (ซึ่งหุ่นยนต์ตัวเดิมก่อนจะเป็นอุทัย) ซึ่งเป็นผลทำให้เกิดปัญหาเรื่องภาระโหลดของดิจิตอลเซอร์วิส ที่ต้องกระทำให้มีมากขึ้นจากเดิมและจะทำให้น้ำหนักของตัวหุ่นยนต์มากขึ้น เมื่อเปรียบเทียบผลลัพธ์น้ำหนักส่วนขาของหุ่นยนต์กับหุ่นยนต์อิวามาอยด์ UTHAI ได้ผลดังตารางที่ ??

ชิ้นส่วน	หุ่นยนต์กัน (เดิม)(กรัม)	หุ่นยนต์ UTHAI
ขาท่อนบน	171	263
ขาท่อนล่าง	172	204

ตารางที่ 4.3: ตารางเปรียบเทียบน้ำหนักของชิ้นส่วนขาของหุ่นยนต์

จากข้อมูลในตารางนั้นจะเห็นได้ว่า น้ำหนักที่เพิ่มขึ้นมากจากการออกแบบใหม่แต่ละชิ้นนั้น มากถึง 124 กรัม ต่อขา 1 ข้าง และ 248 กรัมเมื่อเทียบกับขาทั้งหมดและเปรียบเทียบกับข้อมูลก่อนหน้า

หลังจากพบปัญหาดังกล่าวผู้จัดทำจึงได้ตัดสินใจทำการเปลี่ยนวัสดุที่ใช้ทำโครงสร้างในครั้งแรก ที่จะใช้การขึ้นรูปชิ้นงานด้วยการพิมพ์ขึ้นรูป 3 มิติ มาเป็นวัสดุพอลิเมอร์ห่วงคาร์บอนไฟเบอร์และชิ้นงาน 3 มิติแทน โดยจะให้ชิ้นงาน 3 มิตินั้นทำหน้าที่เป็นตัวเชื่อมระหว่างวัสดุคุณภาพของไฟเบอร์กับมอเตอร์ และยึดกับวัสดุคุณภาพของไฟเบอร์ ด้วยการบีบ ซึ่งเหตุผลที่ต้องทำเช่นนี้ เพราะต้องการลดน้ำหนักของหุ่นยนต์ลง เพื่อไม่ให้มอเตอร์รับภาระที่หนักเกินไป

ชิ้นส่วน	วัสดุที่ใช้ขึ้นรูป (เก่า)	วัสดุที่ใช้ขึ้นรูป (ใหม่)
แขน	ท่อคาร์บอนไฟเบอร์ ขนาด 30 มม.	เดิม
ลำตัว	เครื่องพิมพ์ 3 มิติ โดยใช้วัสดุ PLA	เดิม
เอว	ท่อคาร์บอนไฟเบอร์ ขนาด 88 มม.	เดิม
สะโพก	อลูมิเนียมอลล์loyพับ	เดิม
น่อง	เครื่องพิมพ์ 3 มิติ โดยใช้วัสดุ PLA	วัสดุพอลิเมอร์ห่วงคาร์บอนไฟเบอร์และชิ้นงาน 3 มิติ
ขา	เครื่องพิมพ์ 3 มิติ โดยใช้วัสดุ PLA	วัสดุพอลิเมอร์ห่วงคาร์บอนไฟเบอร์และชิ้นงาน 3 มิติ
ฝ่าเท้า	เครื่องพิมพ์ 3 มิติ โดยใช้วัสดุ PLA	เดิม

ตารางที่ 4.4: ตารางแสดงวัสดุที่แก้ไขในการใช้ขึ้นรูป UTHAI

#### 4.1.1.2 การออกแบบโครงสร้างส่วนขาครั้งที่ 2

ครั้งนี้การออกแบบชิ้นส่วนขาของหุ่นยนต์นั้น ได้คำนึงถึงเรื่องน้ำหนักเป็นหลัก และยังคงให้ความสำคัญกับข้อต่อที่จะใช้รับน้ำหนักทั้งตัวหุ่นยนต์ และยังคงรับแรงบิดของมอเตอร์อีกด้วย ดังนั้นจึงได้ตัดสินใจที่จะเปลี่ยนจากการใช้วัสดุจากการพิมพ์สามมิติ ซึ่งเป็นพลาสติกทั้งหมด มาเป็นวัสดุผสม ระหว่างคาร์บอนไฟเบอร์กับชิ้นส่วนการพิมพ์สามมิติ ซึ่งชิ้นรูปจากพลาสติก PLA และทำการยึดติดกันด้วยการบีบอัด ดังรูปที่ ?? เมื่อทำการพิมพ์ชิ้นงาน ส่วนขาท่อนบนและท่อนล่างและการประกอบ จะได้น้ำหนักของชิ้นงานเปรียบเทียบกับของเดิม ดังตารางที่ ??



(ก) โครงสร้างส่วนขาแข็ง (ใหม่)      (ข) โครงสร้างส่วนตันขา (ใหม่)

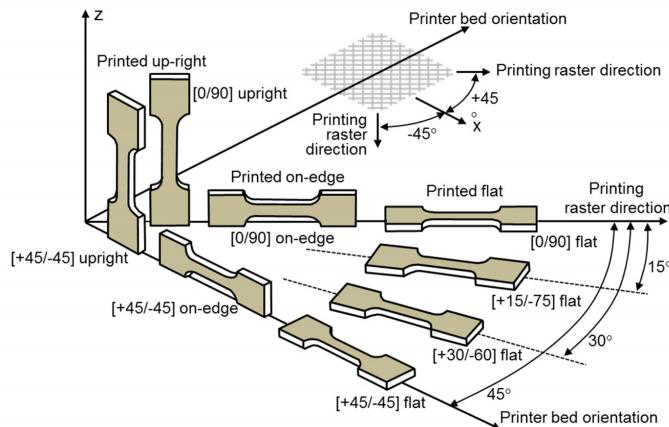
รูปที่ 4.3: รูปการออกแบบส่วนขาของหุ่นยนต์วิษามาโนยด์ UTHAI (ใหม่)

ชิ้นส่วน	น้ำหนักเดิม (กรัม)	น้ำหนักใหม่ (กรัม)
ตันขา	263	161
หน้าแข็ง	204	166

ตารางที่ 4.5: ตารางแสดงน้ำหนักเปรียบเทียบของชิ้นส่วนขา

### การขึ้นรูปชิ้นงาน

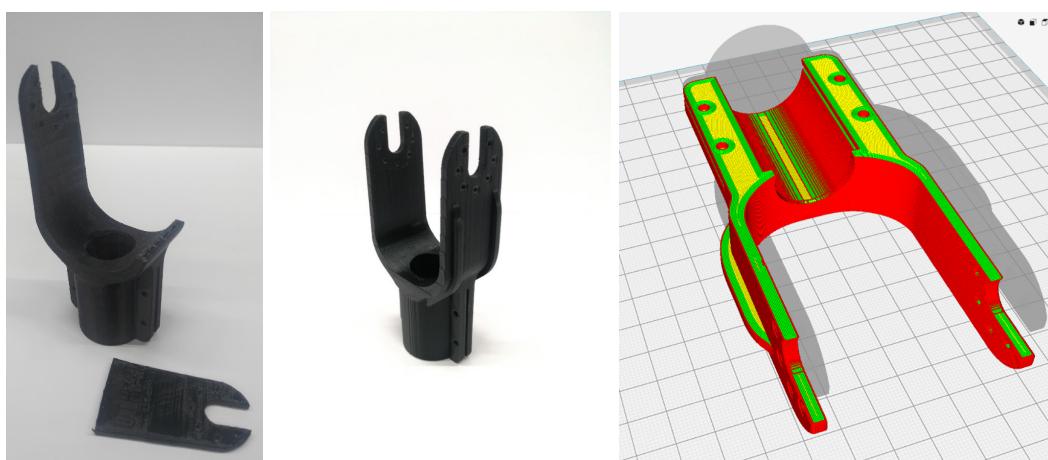
การขึ้นรูปชิ้นงานนั้น ได้ใช้การขึ้นรูปชิ้นตามความสูงแนวแกน Z ดังรูปที่ ?? เพื่อให้ชิ้นงานมีความสวยงาม และสามารถสัมภัส滂ดี กับท่อนคาร์บอนโดยให้มีผิวสัมผัสมากที่สุด ในการยืดเกา<sup>1</sup>



รูปที่ 4.4: รูปการขึ้นรูปชิ้นงานด้วยการพิมพ์งานสามมิติ

#### 4.1.1.3 ทดสอบโครงสร้างและการขับเคลื่อน

จากการทดสอบความแข็งแรงของวัสดุโดยการนำไปประกอบกับตัวทุ่นยนต์จริง และทำการทดลองเดินพบว่าเมื่อทำการเดินจริงนั้น เกิดการฉีกขาดของชิ้นงานที่ ชิ้นการพิมพ์ของชิ้นงาน ดังรูปที่ ?? ซึ่งเกิดจากการได้รับแรงบิดมากเกินไปจากน้ำหนักของชิ้นงานส่วนขา และแรงที่ชิ้นงานจะได้รับนั้น จะเป็นเพียงส่วนการเชื่อมกันติดของชิ้นพลาสติกเท่านั้น ที่นี้เส้นพลาสติกจะไม่ได้เป็นตัวรับแรงจึงทำให้ เกิดการประทักษิ่งที่ยากกว่าดังนั้นจึงทำการออกแบบใหม่โดยการเพิ่มสันให้ชิ้นงานและเพิ่มความหนาบนหน้าแปลนเข้มกับตัวมอเตอร์ และทำการขึ้นรูปชิ้นงาน โดยให้ความสูงของชิ้นงานเป็นไปตามแกน X และทำการเติมเนื้อพลาสติกด้านในให้เต็ม 100% ดังรูปที่ ??



(ก) รูปแสดง การ แตกรัก (ข) รูป แสดง ชิ้น งาน ที่ทำการ ออกแบบใหม่ (ค) รูปแสดง ชั้น ของการพิมพ์ตาม แนวแกน x โดยการเติมเนื้อพลาสติก 100%

รูปที่ 4.5: รูปแสดงการแตกรักและชั้นการพิมพ์

<sup>1</sup>Experimental Characterization of the Mechanical Properties of 3D-Printed ABS and Polycarbonate Parts

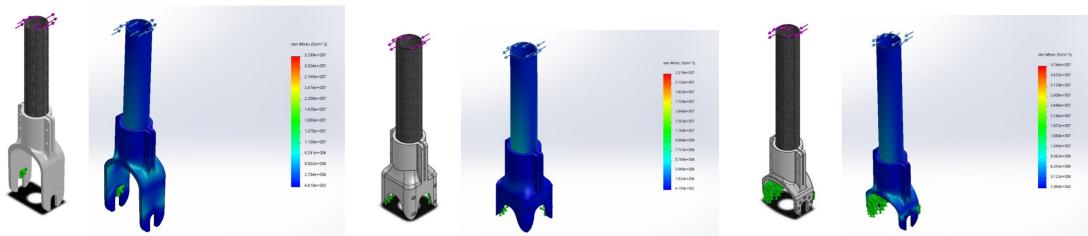
### การทดลองความแข็งแรงของชิ้นงานโดยวิธีการวิเคราะห์เชิงตัวเลข(Finite element)

ก่อนที่จะนำชิ้นงานที่ทำการออกแบบใหม่ที่เติมเนื้อพลาสติก 100% ไปใช้งานจริงนั้นจะต้องผ่านการวิเคราะห์แรงกระทำ โดยผ่านโปรแกรมจำลองเพื่อหาจุดที่เปราะบางของชิ้นงาน และนำข้อมูลนั้นไปวิเคราะห์เพื่อปรับปรุงชิ้นงานต่อไป ซึ่งได้ดังค่า คุณสมบัติของชิ้นงานจากการพิมพ์สามมิติ ไว้ดังตารางที่ ??

Print Orientation Side	flat
Ultimate Stress ( $N/mm^2$ ) <sup>2</sup>	45.66
Young's Modulus ( $N/mm^2$ ) <sup>3</sup>	1141.55
Yield strength ( $N/mm^2$ ) <sup>4</sup>	23
Density ( $kg/m^3$ ) <sup>5</sup>	1250
Poisson ratio <sup>6</sup>	0.33
Force (torque) ( $N.m$ )	10.4

ตารางที่ 4.6: ตารางแสดงค่าคุณสมบัติของวัสดุจากเครื่องพิมพ์สามมิติ

เมื่อทดลองนำค่าดังกล่าวไปใช้ในโปรแกรม Solidwork และใช้ฟังก์ชัน Mass Property เพื่อหาค่าน้ำหนักที่ผ่านการคำนวนโดยโปรแกรม และนำมาเทียบกับชิ้นงานจริงเพื่อคุณภาพความคลาดเคลื่อนด้านน้ำหนักมากน้อยขนาดไหน พบว่าค่าข้อมูลความคลาดเคลื่อนนั้นจะไม่เกินกว่าระหว่าง  $\pm 1\%$  กับค่าที่แสดงบนโปรแกรม หลังจากนั้นทำการวิเคราะห์ผ่านโปรแกรม Solidwork ด้วยการวิเคราะห์ FEA ได้ผลลัพธ์ ดังนี้



(ก) รูปการวิเคราะห์แรงของข้อต่อ 1      (ข) รูปการวิเคราะห์แรงของข้อต่อ 2      (ค) รูปการวิเคราะห์แรงของข้อต่อ 3

รูปที่ 4.6: รูปการวิเคราะห์แรงของข้อต่อ

การวิเคราะห์นั้นจะทำการยึดจุดที่เป็นหน้าแปลนของมอเตอร์ไว้เพื่อให้เปรียบเสมือนกับว่าขณะนี้ชิ้นงานได้เข้มติดกับตัวมอเตอร์ หลังจากนั้นกำหนดแรงกระทำเพื่อให้เกิดโมเมนต์กับชิ้นงานโดยค่าแรงที่กระทำนั้น ได้มาจากการคำนวนแรงของมอเตอร์ที่จะรับไหว้เทียบกับระยะของแรง ที่กระทำกับชิ้นงาน ซึ่งได้ทดลองกับชิ้นงาน ตัวข้อต่อ 1 2 และ 3 ด้วยแรง 41.6 นิวตัน ( $N$ ) เมื่อนำค่า ความตึงเครียดสูงสุด ( $Max stress$ ) ของชิ้นงาน มาวิเคราะห์เพื่อหา จุดเปราะบางของวัสดุได้ดังตาราง ?? จากผลการทดลองสรุปได้ว่า ค่าความตึงเครียดต่างๆ

ชื่อชิ้นงาน	ความตึงเครียดสูงสุดของชิ้นงาน( $Max stress$ ) ( $N/mm^2$ )
ข้อต่อ 1	32.99
ข้อต่อ 2	23.19
ข้อต่อ 3	7.987

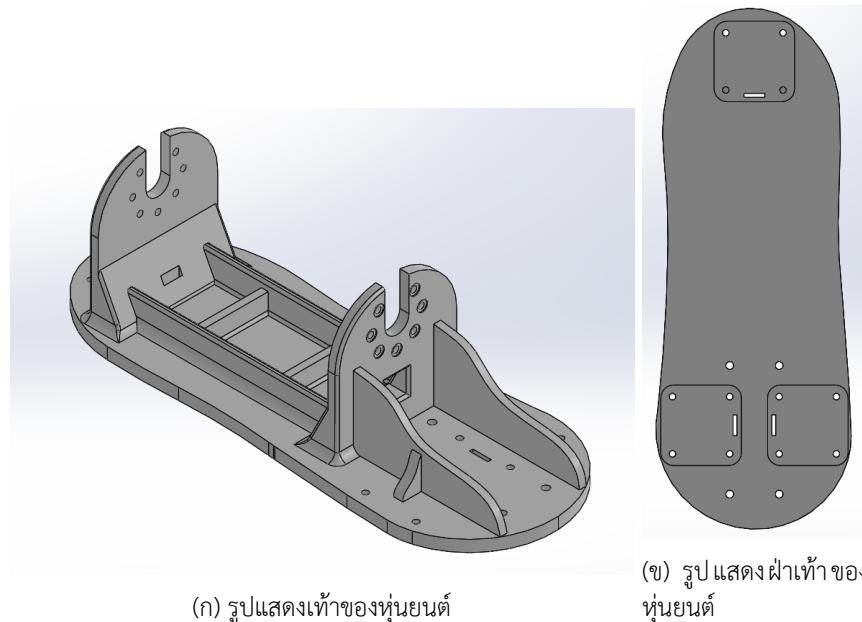
ตารางที่ 4.7: ตารางแสดงความตึงเครียดของชิ้นงาน (Stress)

ที่ได้มาจากการทดลองนั้นมี่อนนำไปเทียบกับค่าความตึงเครียดสูงสุดที่วัสดุจะรับไหว้ที่  $45.66 N/mm^2$  เนื่องด้วยวัสดุไม่มีวัสดุตัวไหนที่จะเกิดการแตกหักเมื่อเกิดแรงกระทำกับชิ้นงานดังนั้นชิ้นงานที่ทำการออกแบบนี้ พoSรุปได้ว่าจะไม่เกิดการแตกหักระหว่างการทำงาน ยกเว้นมีแรงกระทำจากภายนอกที่มากเกินไปจนมาผลทำให้เกิดความตึงเครียดของชิ้นงานสูง เกินกว่าค่าดังกล่าว

#### 4.1.2 การออกแบบเท้า

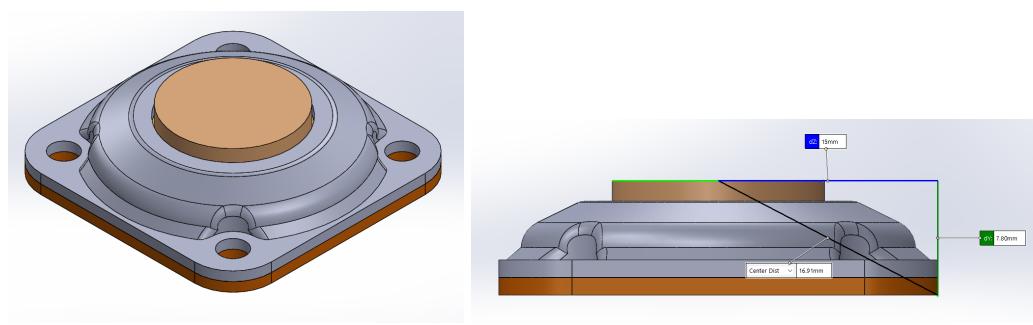
##### 4.1.2.1 การออกแบบโครงสร้างเท้าครั้งที่ 1

โครงสร้างเท้านี้ได้ออกแบบให้มีลักษณะคล้ายคลึงกับรองเท้าของมนุษย์จริงและมีขนาดที่เหมาะสมกับตัวหุ่นยนต์ โดยคำนึงถึงความแข็งแรง และการใช้งานเป็นหลัก และยังต้องขึ้นรูปด้วยเครื่องพิมพ์ 3 มิติได้อีกด้วย



รูปที่ 4.7: รูปแสดงเท้าของหุ่นยนต์ชิวามานอยด์ UTHAI

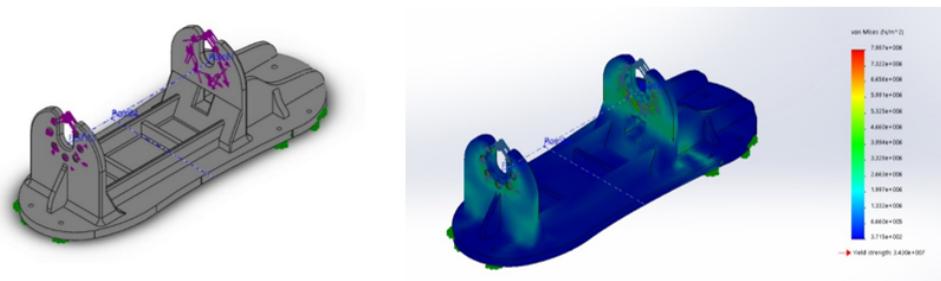
ในส่วนของโครงครอบ FSR นั้นได้ออกแบบให้มีการกดโดยตรงกับหน้าสัมผัสซึ่งส่วนที่สัมผัสกับหน้าสัมผัสนั้นจะเป็นเฉพาะส่วนของ 3d print ที่ออกแบบมา เฉพาะการกดโดยเฉพาะ ซึ่งจะทำให้สนองหน้าสัมผัสได้ดีกว่า การกดจากภายนอกโดยตรงซึ่งตัวเซนเซอร์นี้จะมีความสูงจากมาจากการฝ่าเท้าเป็นระยะ 5.4 มิลลิเมตร โดยจะมีจำนวน 3 ตัวต่อเท้า 1 ข้าง



รูปที่ 4.8: รูปแสดงโดยรวมของโครงครอบ FSR

### การทดลองความแข็งแรงของชิ้นงานโดยวิธีการวิเคราะห์เชิงตัวเลข (Finite element)

จากค่าคุณสมบัติของชิ้นงานดังตาราง ?? นำมาวิเคราะห์ด้วยเทคนิคเชิงตัวเลขได้ผลดังนี้ การวิเคราะห์นั้น



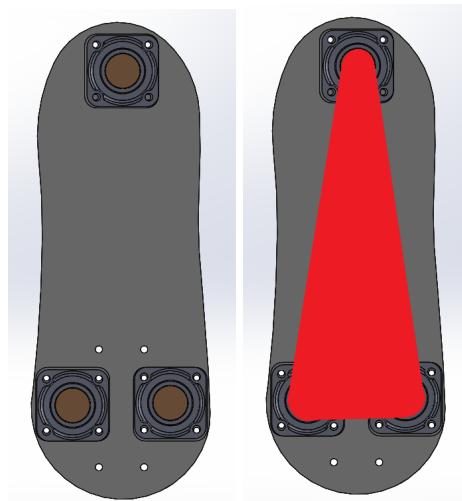
รูปที่ 4.9: รูปการวิเคราะห์แรงของเท้าหุ่นยนต์

มีวิธีการทำเหมือนกับการวิเคราะห์ชิ้นส่วนขาคือทำการยืดจุดที่เป็นหน้าแปลนของดิจิตอลเซอร์โวไว้เพื่อให้เปรียบเสมือนกับว่าขณะนี้ชิ้นงานได้เข้มติดกับตัวดิจิตอลเซอร์โว หลังจากนั้นกำหนดแรงกระทำเพื่อให้เกิดแรงบิดกับชิ้นงานโดยค่าแรงที่กระทำนั้นได้มาจากการคำนวณแรงของดิจิตอลเซอร์โวที่ทำกับชิ้นงานข้อเท้าที่แรงบิด 10.4 นิวตัน/เมตร ( $N.m$ ) ได้ผลดังตารางที่ ??

ชื่อชิ้นงาน	ความตึงเครียดสูงสุดของชิ้นงาน (Max stress) ( $N/mm^2$ )
ฝ่าเท้า	37.44

ตารางที่ 4.8: ตารางแสดงความตึงเครียดของชิ้นงาน (Stress) ของฝ่าเท้า  
ปัญหาที่พบ

เมื่อทำการประกอบชิ้นส่วนของ FSR กับฝ่าเท้าแล้วปัญหาที่พบคือ พื้นที่สัมผัสพื้นของฝ่าเท้า น้อยลงซึ่งเป็นผลทำให้พื้นที่รองรับน้อยลงด้วยซึ่งเป็นเหตุทำให้การเดินของหุ่นยนต์นั้นยากลำบาก ดังนั้นจึงทำการแก้ไขโดยออกแบบฝ่าเท้าให้มีขนาดใหญ่เพิ่มขึ้น และออกแบบเซนเซอร์ตรวจจับการเดินให้มีความบางลงอีกเพื่อให้หุ่นยนต์นั้น มีพื้นที่รองรับเพิ่มขึ้น

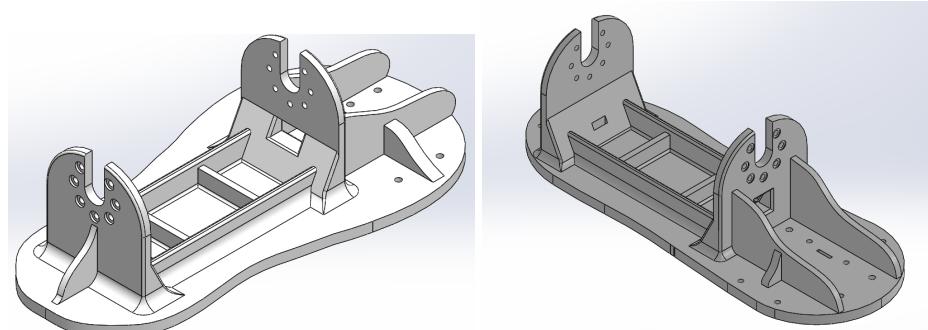


(ก) เท้าของหุ่นยนต์ (ข) พื้นที่รองรับของเท้า

รูปที่ 4.10: รูปแสดงฝ่าเท้าและพื้นที่รองรับ

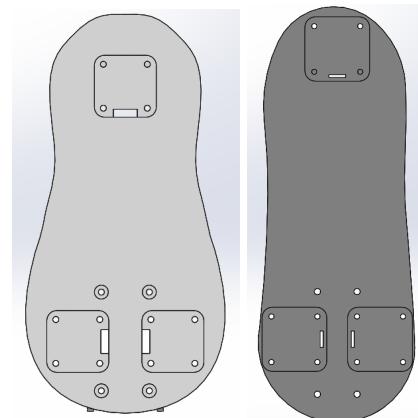
#### 4.1.2.2 การออกแบบโครงสร้างเท้าครั้งที่ 2

ในการออกแบบครั้งนี้ได้เพิ่มขนาดของฝ่าเท้าให้ใหญ่ขึ้นเพื่อเพิ่มพื้นที่สัมผัสดังจะเป็นผลทำให้หุ่นยนต์นั้นมีการเดินที่ง่ายขึ้น ในการปรับขนาดครั้งนี้ได้เพิ่มขนาดที่ปลายเท้าให้ใหญ่ขึ้น และสันเท้ารองลงมา



(ก) รูปแสดงเท้าของหุ่นยนต์ที่ออกแบบใหม่

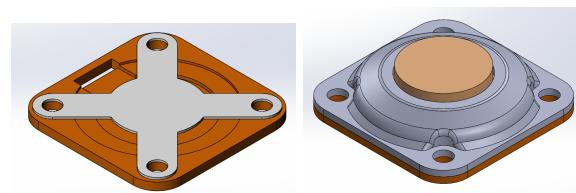
(ข) รูปแสดงเท้าของหุ่นยนต์เดิม



(ค) รูปแสดงฝ่าเท้า ที่ (ง) รูปแสดงฝ่าเท้า  
ออกแบบใหม่ เดิม

รูปที่ 4.11: รูปแสดงการเปรียบเทียบระหว่างเท้าเดิมและเท้าที่ออกแบบใหม่

ในส่วนของเซนเซอร์ตรวจจับพื้นน้ำได้ทำการออกแบบใหม่ทั้งหมดให้มีความบางลงกว่าเดิม และให้มีส่วนที่ยื่นออกมา จากฝ่าเท้าน้อยที่สุดซึ่งความสูงที่ยื่นออกมา มีความสูงเพียง 0.4 มิลลิเมตร ต่างจากเดิมที่มีความสูงถึง 5.4 มิลลิเมตร

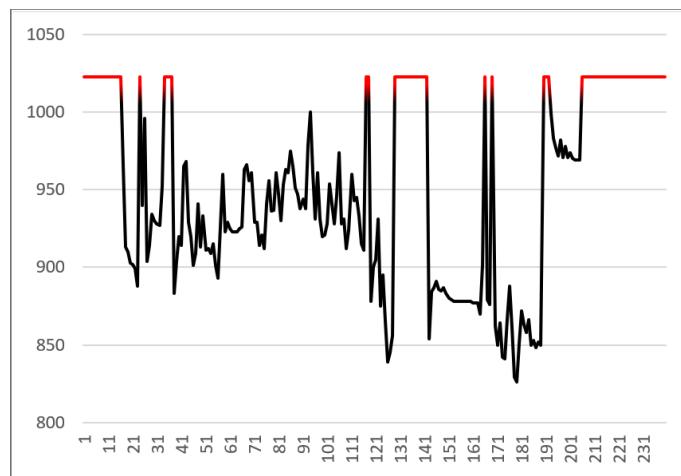


(ก) รูปแสดงโครงครอบ FSR ที่ (ข) รูปแสดงโครงครอบ FSR  
ออกแบบใหม่ เดิม

รูปที่ 4.12: รูปแสดงการเปรียบเทียบระหว่างฝ่าเท้าเดิมและฝ่าเท้าที่ออกแบบใหม่

#### 4.1.2.3 ทดสอบการใช้งานของเซนเซอร์ตรวจจับการสัมผัสพื้น

ในโครงการนี้เซนเซอร์ตรวจจับพื้นนั้นได้ใช้งานเพื่อทำการเรียบของเท้าเท่านั้นว่ามีการแตะพื้นหรือไม่ ซึ่งจะกำหนดค่าไว้ในช่วง 0-1023 โดยจะกำหนดให้ช่วงที่มากกว่า 1000 หมายถึงเท้ามีการยกเกิดขึ้นและต่ำกว่า 1000 หมายถึงเท้ามีการเรียบ โดยจะแปลงค่าเซนเซอร์ทั้ง 3 ค่าบนเท้า เป็นค่าเฉลี่ยน้ำหนัก ตามสมการ  $(sensor1 + sensor2 + 2(sensor3))/4$  โดยให้ค่าตามแกน X เป็นค่า sampling(ครั้ง) ของข้อมูลและ ค่าตามแกน Y เป็นค่าซึ่ง 0-1023 เมื่อทำการทดลองให้ทำการเรียบเท้าและยกเท้าจะได้ผล ดังรูปที่ ??



รูปที่ 4.13: รูปภาพแสดงค่าที่วัดได้ของ FSR

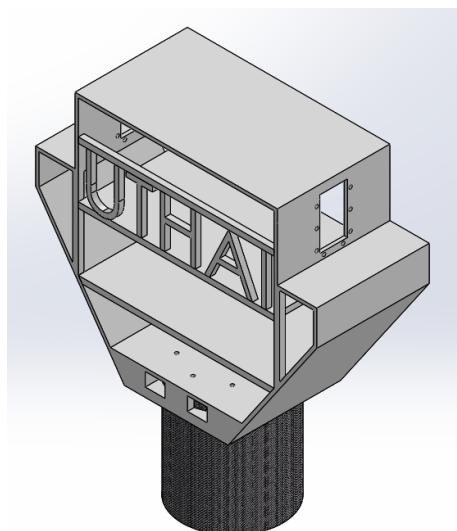
ผลที่ได้นี้เกิดจากการทดลองยกเท้าของหุ่นยนต์ขึ้นขณะหุ่นยนต์ยืนอยู่และยกเท้าขึ้น ค่าที่อ่านได้จะมีค่า 1023 หรือสูงสุดตามภาพ และเมื่อเท้าสัมผัสพื้นนั้น ค่าที่อ่านได้จะมีค่าที่ต่ำขึ้นอยู่กับแรงกดที่กระทำต่อเท้า ซึ่งถ้าค่าที่กระทำมากจะทำให้ค่าที่อ่านได้ต่ำมาก จะเห็นได้จากรูปภาพ ?? จะมีช่วงเวลาหนึ่งที่ค่าที่อ่านได้ต่ำสุดซึ่งขณะนั้น ได้ทดลองกดด้วยแรงจำนวนหนึ่งที่มากกว่าน้ำหนักตัวหุ่นยนต์เป็นเวลา 1-2 วินาที

#### 4.1.3 การออกแบบโครงสร้างลำตัว

ลำตัวของหุ่นยนต์นั้นจะใช้สำหรับติดตั้งหน่วยประมวลผลควบคุมระดับต่ำและระดับสูง IMU รวมไปถึงบอร์ดแปลงไฟ 12v จากแบตเตอรี่ ให้เหลือ 5V เพื่อจ่ายไฟให้กับระบบ ละยังคงต้องจัดเก็บแบตเตอรี่สำหรับทำงานไร้สายได้อีกด้วย

##### 4.1.3.1 การออกแบบลำตัวครั้งที่ 1

การออกแบบครั้งนี้ได้ออกแบบให้ลำตัวนั้นขึ้นรูปด้วยเครื่องพิมพ์ 3 มิติ เพื่อให้ง่ายสำหรับผู้ต่อยอดที่มีเครื่องพิมพ์เป็นของตนเอง สามารถพิมพ์ และนำมาประกอบได้ โดยส่วนของเอวันั้นจะใช้เป็นท่อคาร์บอนไฟเบอร์ขนาดเส้นผ่าศูนย์กลางขนาด 91 มิลลิเมตร เพื่อให้เหมาะสมกับชิ้นงาน และเชื่อมยึดติดกันด้วยสกรูกับลำตัวและส่วนเอว

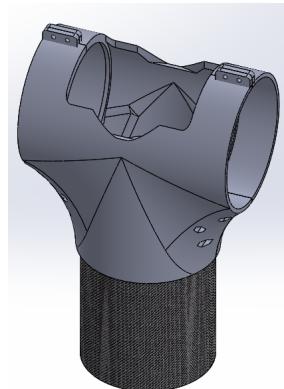


รูปที่ 4.14: รูปภาพแสดงตัวของหุ่นยนต์ที่ออกแบบเพื่อขึ้นรูปด้วยเครื่องพิมพ์ 3 มิติ

แต่เมื่อทำการทดลองหาค่ามวลในโปรแกรม solidwork แล้ว ได้ผลน้ำหนักคือ 711 กรัม ซึ่งเป็นน้ำหนักที่มากเกินไปอาจส่งผลทำให้มอเตอร์รับน้ำหนักของตัวมากเกินไป ดังนั้นจึงเป็นผลทำให้ต้องเปลี่ยนการออกแบบให้เบาลงกว่าเดิม คือลดขนาดของตัวที่ขึ้นรูปด้วยเครื่องพิมพ์ 3 มิติ ลงและใช้วัสดุสมรรถนะว่าการบอนไฟเบอร์มากขึ้น ซึ่งจะยังคงได้ความแข็งแรงและความเบาอีกด้วย

#### 4.1.3.2 การออกแบบโครงสร้างตัวครั้งที่ 2

จากปัญหาเรื่องน้ำหนักของชิ้นส่วนตัวของการออกแบบครั้งที่ 1 ได้แก้ไขโดยลดขนาดของส่วนพิมพ์ 3 มิติ ลงซึ่งจะแยกชิ้นส่วนออกเป็น 2 ส่วน คือส่วนหน้าและส่วนหลัง และใช้การยึดกับท่อคาร์บอนไฟเบอร์ซึ่งเป็นส่วนเอวทวยการบีบซึ่งทำโดยการร้อยสกรูผ่านช่องที่ทำไว้สำหรับนตัวชิ้นงานพิมพ์ 3 มิติ แล้วขันให้จิ้นงานมาประกับเข้าหากัน ซึ่งน้ำหนักที่วิเคราะห์ได้โดยโปรแกรม solidwork นั้นได้ค่าเท่ากับ 342 กรัม ซึ่งน้อยกว่าการออกแบบ



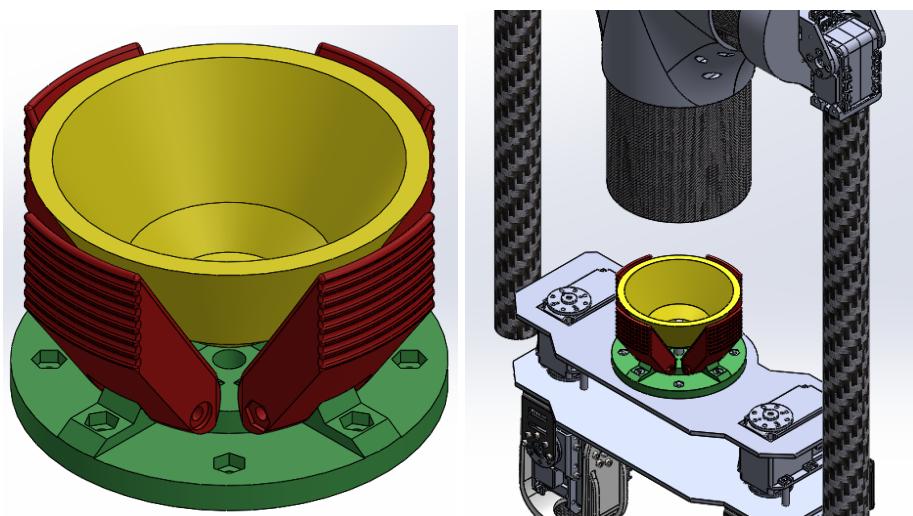
รูปที่ 4.15: รูปภาพแสดงตัวของหุ้นยนต์ที่ออกแบบเพื่อขึ้นรูปด้วยเครื่องพิมพ์ 3 มิติ(ใหม่)

ครั้งแรกถึง 2 เท่า ซึ่งมีน้ำหนักมากถึง 711 กรัม

#### การยึดลำตัวกับสะโพก

การยึดลำตัวกับสะโพกนี้จะใช้รูปแบบการยึดโดยการถ่างวัสดุที่ทำขึ้นมาเพื่อยึดลำตัวกับสะโพกออกผ่านสกรู 1 ตัวที่ออกแบบไว้ โดยสกรูตัวนี้จะทำหน้าที่ดึงให้ถ่ายของตัวถ่าง เคลื่อนที่ลงมาและในขณะนั้นเองตัวถ่างด้านนอกอีก 4 ตัว จะค่อยๆขยับออกถ่างให้มีแรงบีบขอบท่อคาร์บอนไฟเบอร์และยึดกันอย่างแน่นหนึ่ง

ข้อแนะนำในการยึดให้แน่นมากขึ้น ควรจะใช้วัสดุที่มีความหนืด เช่น ยางในรถจักรยาน หรือแผ่นกันเลื่อนยึดกับหน้าสัมผัสของตัวถ่างก่อนแล้วจึงนำไปยึดกับวัสดุจริง)

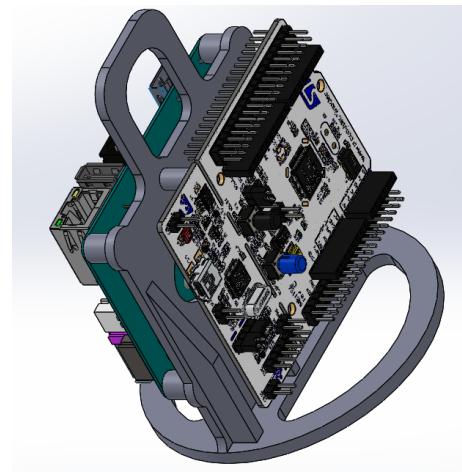


(ก) รูปภาพ แสดง อุปกรณ์ยึด ระหว่าง ลำ ตัว กับ (ข) รูปภาพ แสดง การ ติด ตั้ง บน สะโพก พร้อม ทำการยึดกับลำตัวหุ้นยนต์

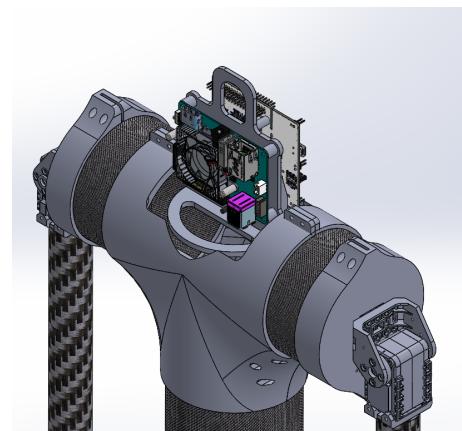
รูปที่ 4.16: รูปแสดงการใช้งานส่วนยึดสะโพกกับลำตัว

### การติดตั้งบอร์ดควบคุมและแบตเตอรี่

เนื่องจากว่าบอร์ดควบคุมทั้ง 2 (Nucleo f411re, Odroid XU4) นั้นมีขนาดที่กระหัดรัด รวมถึงบอร์ด IMU และบอร์ดแปลงไฟ ที่มีขนาดเล็กเช่นกัน ฉะนั้นจึงได้ออกแบบ ฐานสำหรับยึดบอร์ดทั้งหมดไว้ในที่เดียว และเมื่อติดตั้งในฐานเรียบร้อยแล้วก็สามารถนำฐานนั้น สามลงไปในตัวของหุ่นยนต์ได้พอดี



รูปที่ 4.17: รูปภาพแสดงฐานที่ติดตั้งบอร์ดควบคุม

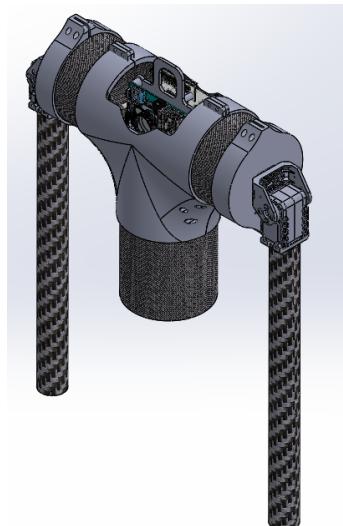


รูปที่ 4.18: รูปภาพแสดงฐานที่ติดตั้งบอร์ดควบคุมในตัวของหุ่นยนต์

ซึ่งเมื่อทำการติดตั้งในตัวหุ่นยนต์แล้ว การยึดติดกับบอร์ดนั้นใช้หลักการยึดเดียวกับการยึดห้องเครื่องบอนกับลำตัวคือ ใช้แรงของการบีบอัดจากสกรูบนลำตัวทั้งหมด ยึดให้อยู่กับที่ ส่วนของแบตเตอรี่นั้นจะใช้เป็นแบตเตอรี่ขนาด  $6000mAh$   $12.6V$  จะถูกติดตั้งในส่วนลำตัวของหุ่นยนต์บริเวณห้องเครื่องบอนไฟเบอร์

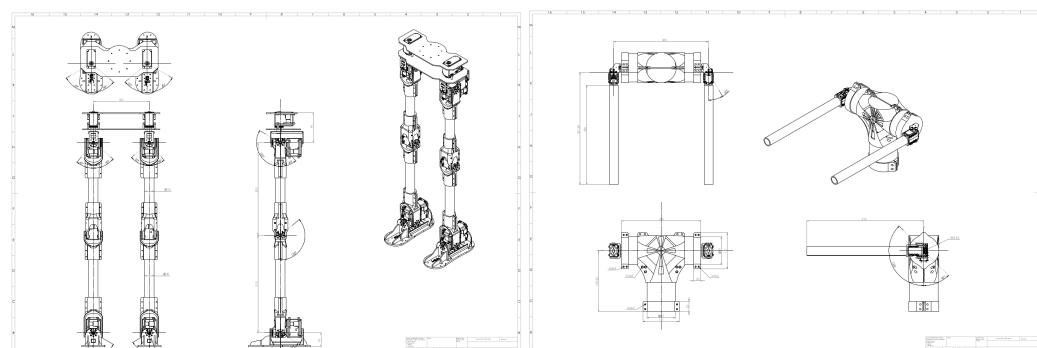
#### 4.1.3.3 การออกแบบแขน

แขนนี้ได้ออกแบบให้เรียบง่ายและน้ำหนักเบา ซึ่งในโครงการนี้แขนจะเป็นหนึ่งในตัวแปรที่ช่วยในการเดินให้คล่องแคล่วมากขึ้น โดยวัสดุหลักที่ใช้มาทำแขนนั้นจะมาจากวัสดุคาร์บอนไฟเบอร์เป็นหลักและชิ้นส่วนพิมพ์ 3 มิติจะใช้สำหรับเชื่อมวัสดุทั้งหมดเข้าด้วยกัน



รูปที่ 4.19: รูปภาพแสดงตัวของหุ่นยนต์ที่ติดตั้งแขนทั้ง 2 ข้าง

#### 4.1.4 แบบวาดทางวิศวกรรม



(ก) ขาหุ่นยนต์อิ้มานอยด์ UTHAI

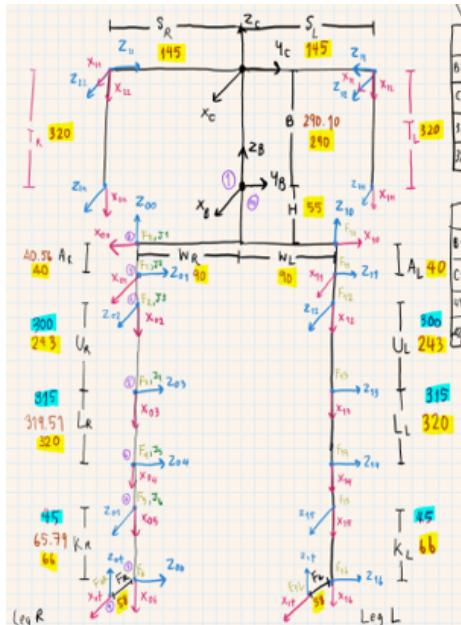
(ข) ตัวหุ่นยนต์อิ้มานอยด์ UTHAI

รูปที่ 4.20: ภาพแบบวาดทางวิศวกรรม

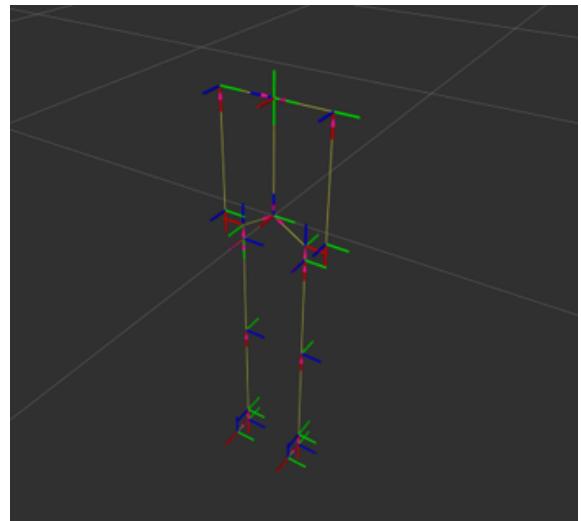
#### 4.2 การออกแบบโปรแกรมด้วย ROS

#### 4.2.1 ออกแบบแบบจำลองของทุ่นยนต์ชีวภาพอยู่ด้วย UTHAI

การออกแบบนี้มีป้าหมายเพื่อ สร้างแบบจำลองของหุ่นยนต์อิวามานอยด์ UTHA ให้ถูกต้องตามที่โครงสร้างทางกลได้สร้างเอาไว้ โดยใช้โปรแกรม RViz เป็นตัวแสดงผลให้เห็นเป็นภาพ

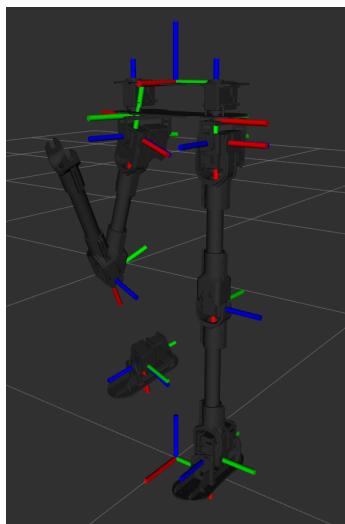


(ก) ค่าพารามิเตอร์ที่ใช้ในการสร้างแบบจำลอง

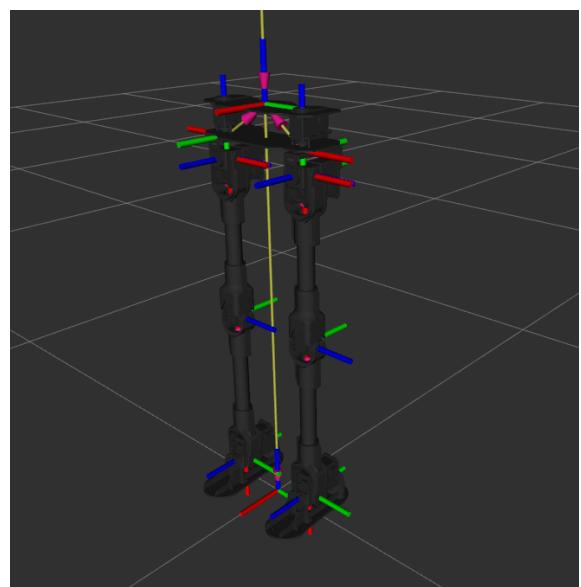


(๗) URDF ที่มีแต่โครงไม่มีก้านต่อ

รูปที่ 4.21: URDF ที่แสดงผลใน RViz



(ก) URDF ที่เกิดข้อผิดพลาด

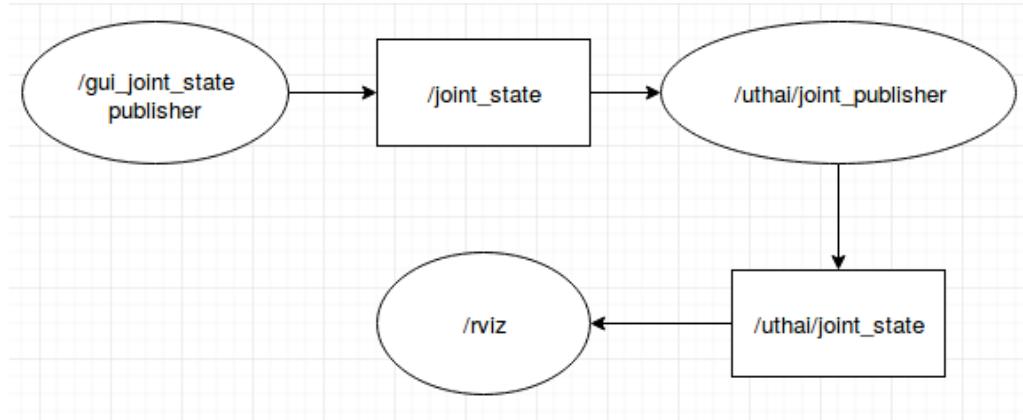


(ข) URDF ที่ทำงานได้ถูกต้อง

รูปที่ 4.22: URDF ที่แสดงผลใน RViz

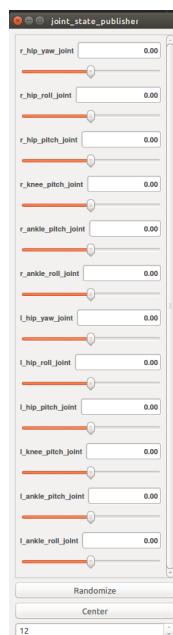
#### 4.2.2 การสั่งงานข้อต่อใน RViz ผ่าน GUI

การเขียนโปรแกรมในส่วนนี้มีเป้าหมายเพื่อ ทดสอบการทำงานของโปรแกรม RViz ที่จะแสดงการทำงานของข้อต่อเป็นอย่างไร การวางแผนของหุ่นยนต์นั้นมีความถูกต้องหรือไม่

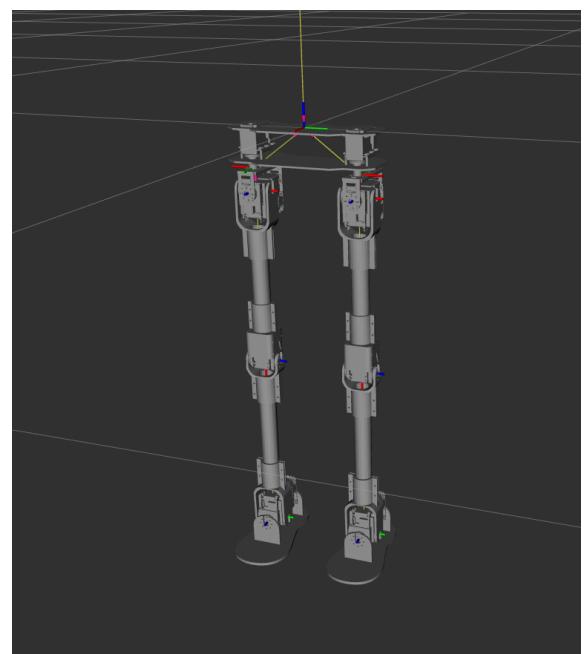


รูปที่ 4.23: การสั่งการข้อต่อใน RViz ผ่าน GUI

หลังจากที่ทำการทดลองเสร็จแล้วพบว่าสิ่งที่ขาดหายไปคือการตั้งค่า Joint limit ให้กับหุ่นยนต์ข้อมาน้อยด้วยในโปรแกรม RViz จึงต้องกลับไปแก้ค่าและทดสอบใหม่ จากนั้นถือว่าการทดลองเสร็จสิ้น สามารถที่จะควบคุมการเคลื่อนไหวของแต่ละข้อต่อผ่าน GUI ได้



(ก) GUI สำหรับสั่งงานแต่ละข้อต่อ

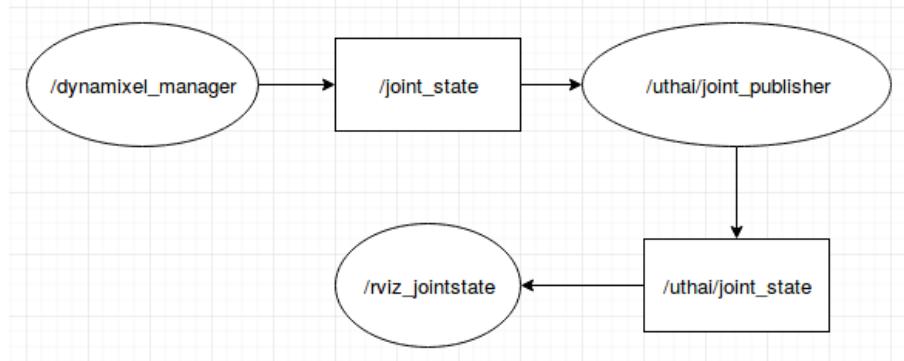


(ข) หุ่นยนต์ที่แสดงผลในโปรแกรม RViz

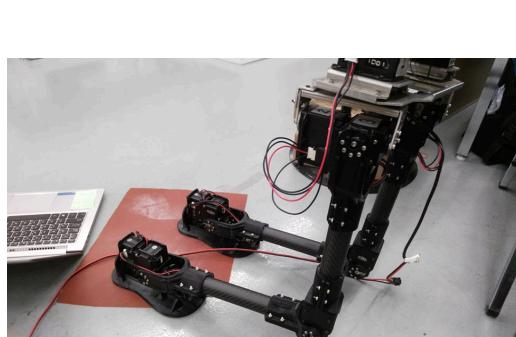
รูปที่ 4.24: การสั่งการข้อต่อใน RViz ผ่าน GUI

#### 4.2.3 การเขียนโปรแกรมอ่านตำแหน่งจากเซอร์โวมอเตอร์เข้าระบบ

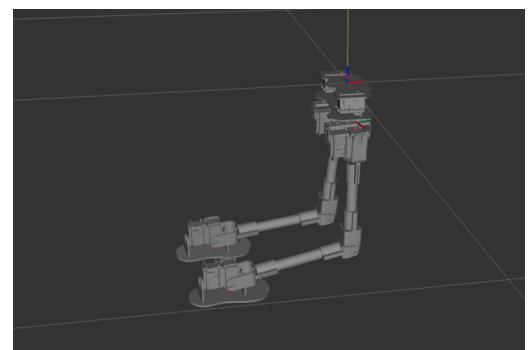
การเขียนโปรแกรมในส่วนนี้มีเป้าหมายเพื่อ ทดสอบการอ่านตำแหน่งจากเซอร์โวมอเตอร์ และทดสอบการสั่งการระบบแสดงผลด้วยภาพ จึงได้ออกแบบการทดลองนี้ขึ้นมา การดำเนินการเริ่มจาก เขียนโปรแกรม สำหรับอ่านค่าตำแหน่งจากเซอร์โวซึ่งจะได้ตำแหน่งของมาใน topic JointState หลังจากนั้นก็จึงแปลงให้อยู่ใน message เดียวกับที่ Rviz ต้องการ แล้วจึงส่งออกไป



รูปที่ 4.25: การติดต่อสื่อสารระหว่างจากเซอร์โวมอเตอร์กับระบบ

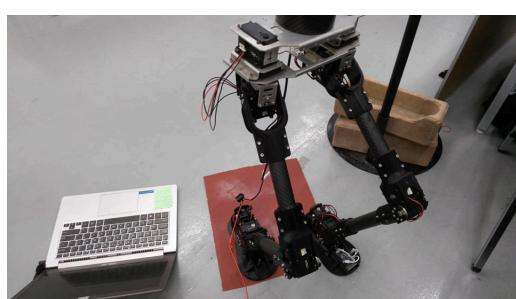


(ก) หุ่นยนต์ตัวจริง

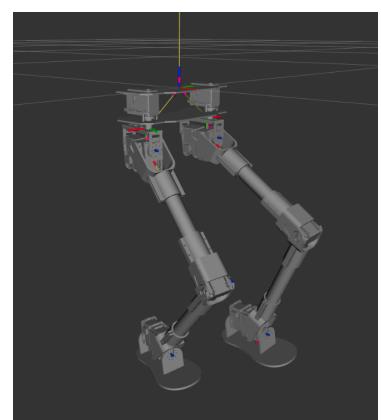


(ข) หุ่นยนต์ใน RViz

รูปที่ 4.26: การแสดงผลท่าทาง 1



(ก) หุ่นยนต์ตัวจริง



(ข) หุ่นยนต์ใน RViz

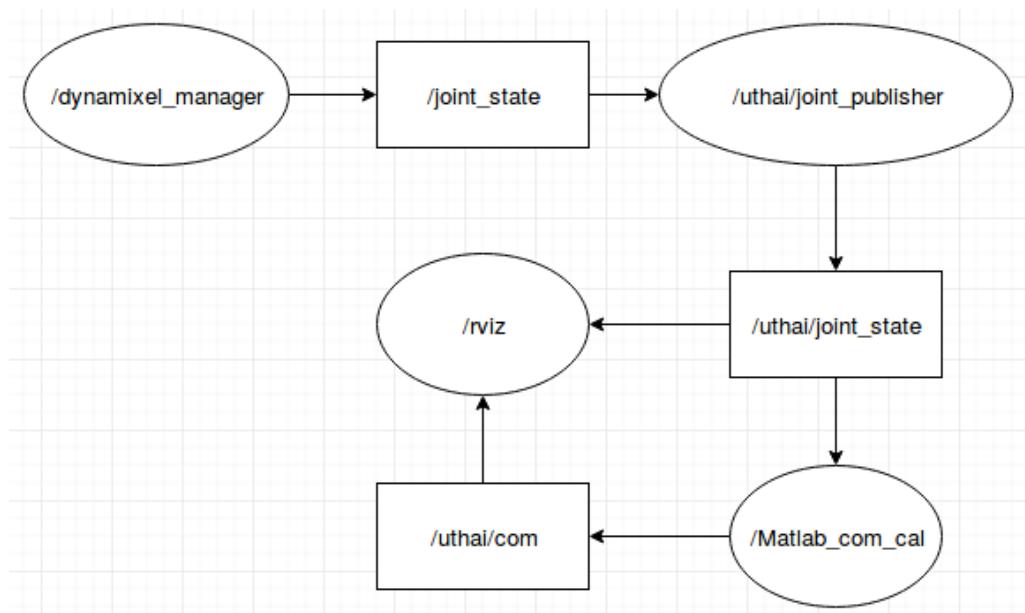
รูปที่ 4.27: การแสดงผลท่าทาง 2



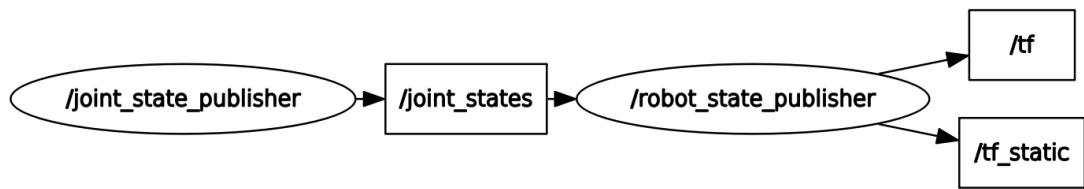
รูปที่ 4.28: การแสดงผลท่าทาง 3

#### 4.2.4 การส่งตำแหน่งของเซอร์วอโมเตอร์ไปประมวลผลหาจุดศูนย์กลางมวล

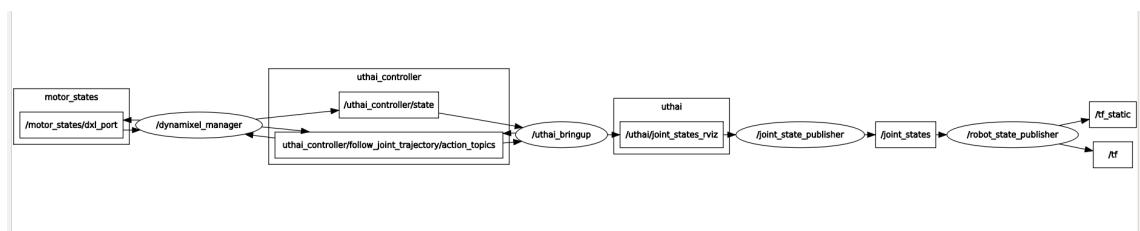
การเขียนโปรแกรมในส่วนนี้มีเป้าหมายเพื่อ ทดสอบการทำงานการเชื่อมต่อกับโปรแกรมภายนอกโดยการ ส่งค่าตำแหน่งของเซอร์วอโมเตอร์ ออกไปแล้วมีโปรแกรมจาก MATLAB รับตำแหน่งไปแล้วประมวลผลเพื่อเอา ตำแหน่งของจุดศูนย์รวมมวลเข้ามาแสดงในโปรแกรม Rviz



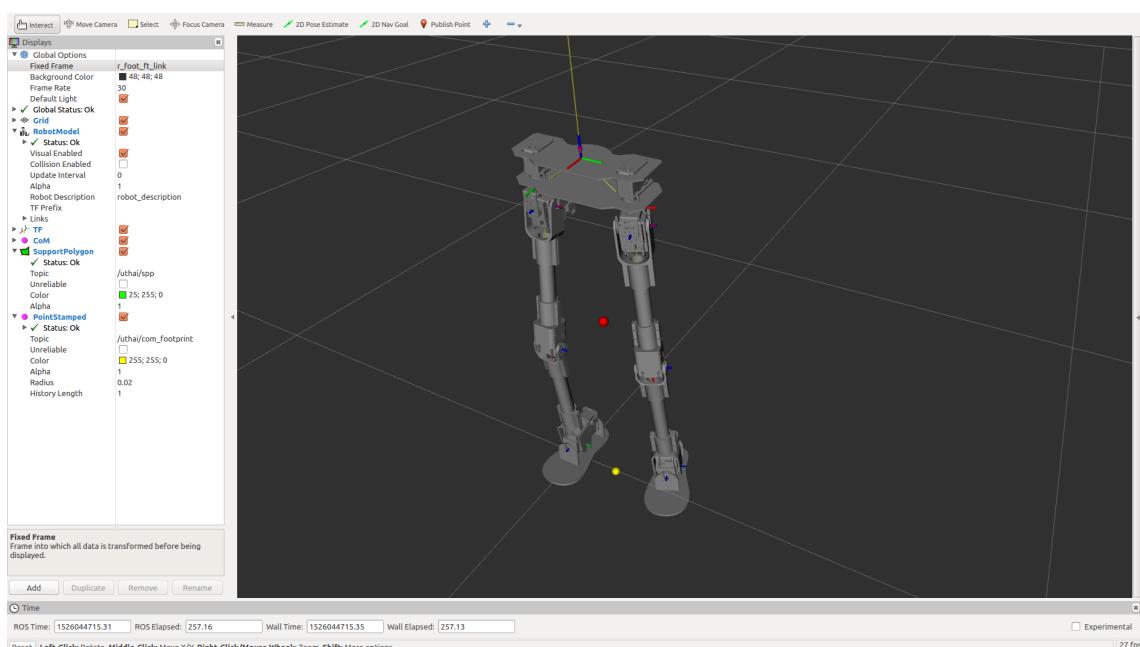
รูปที่ 4.29: การประมวลผลตำแหน่งจุดศูนย์กลางมวลด้วย MATLAB



รูปที่ 4.30: การเชื่อมต่อ กันระหว่าง Node ก่อน เชื่อมต่อเซอร์วิsmo เตอร์



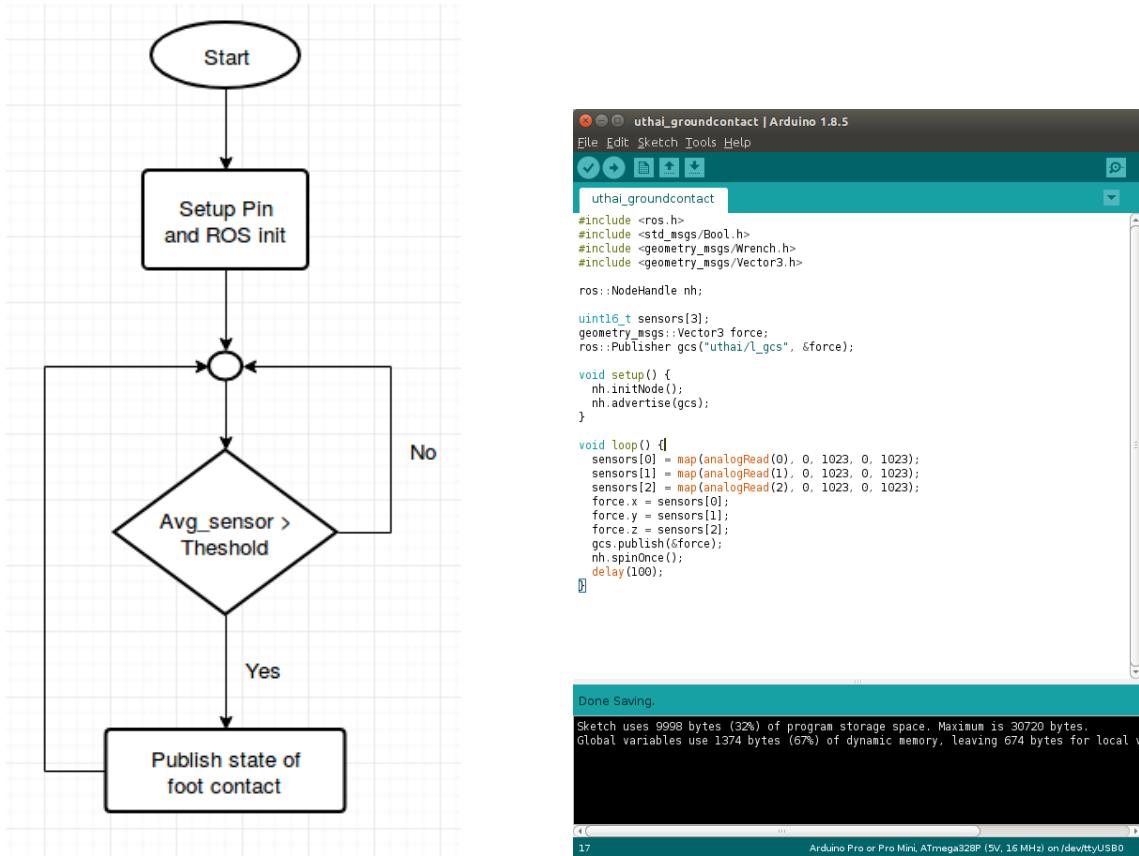
รูปที่ 4.31: การเชื่อมต่อ กันระหว่าง Node หลัง เชื่อมต่อเซอร์วิsmo เตอร์



รูปที่ 4.32: การประมวลผลตำแหน่งขาจุดศูนย์กลางด้วย MATLAB

#### 4.2.5 การเขียนโปรแกรมเช่นเซอร์ตรวจจับการสัมผัสพื้น

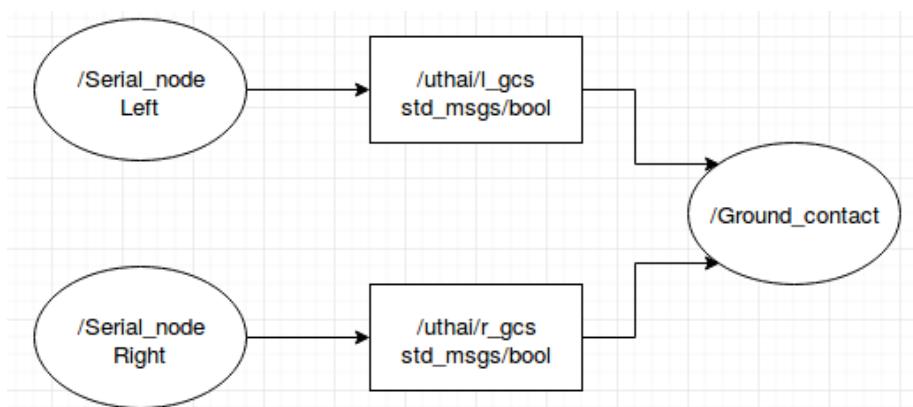
การเขียนโปรแกรมในส่วนนี้มีเป้าหมายเพื่อ บอกให้ตัวประมวลผลระดับสูงรับรู้ว่าเท้าผ่านพื้นที่ไหนมีการสัมผัส กับพื้น เพื่อใช้ในการเปลี่ยนสถานะของโปรแกรมในการคำนวณ ตัวเซอร์ตรวจจับฝ่าเท้านั้นใช้เป็น Arduino Pro-Mini เนื่องจากมีขนาดเล็กและสามารถเชื่อมต่อกับเซนเซอร์วัดแรงกดได้สูงสุด 6 ตัว เมื่อได้อ่านแล้วจากนั้น จึงวางไฟล์char์ต การทำงานของโปรแกรม เมื่อได้ไฟล์char์ตแล้วจึงเริ่มเขียนโปรแกรมเพื่อทดสอบการทำงาน



(ก) ไฟล์char์ตของเซนเซอร์ตรวจจับพื้น

(ข) โปรแกรมของเซนเซอร์ตรวจจับพื้น

รูปที่ 4.33: เซนเซอร์ตรวจจับพื้น

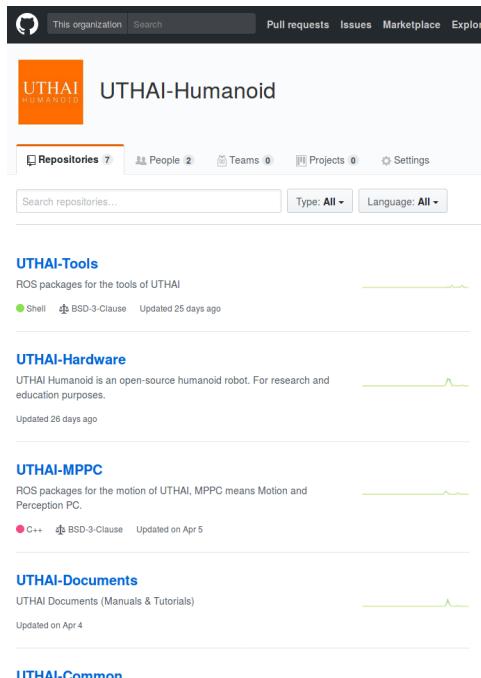


รูปที่ 4.34: การติดต่อสื่อสารระหว่างเซนเซอร์ตรวจจับเท้ากับระบบ

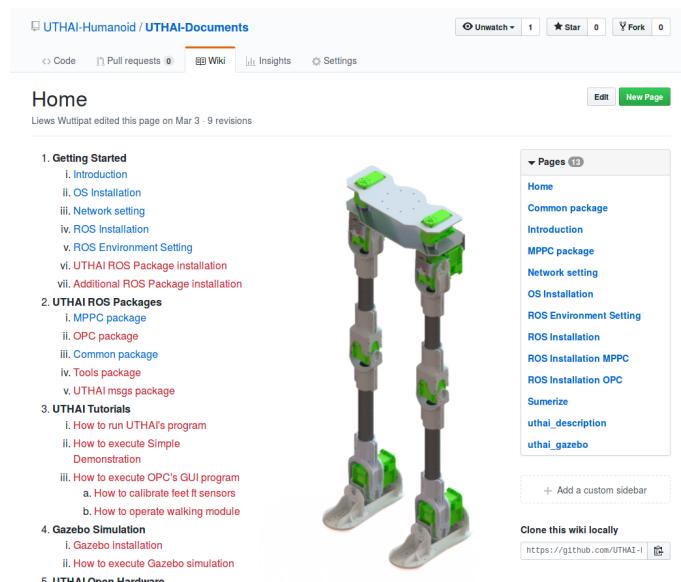
## 4.3 การออกแบบระบบพื้นฐาน

### 4.3.1 GitHub ของหุ่นยนต์อิฐมานอยด์ UTHAI

ไฟล์ข้อมูลทุกอย่างเกี่ยวกับหุ่นยนต์อิฐมานอยด์ UTHAI ได้ถูกอัพโหลดขึ้นบนอินเทอร์เน็ต โดยอัพโหลดไปไว้ที่ GitHub [https://github.com/UTHAI-Humanoid] และมีการเขียน Wiki การใช้งานเบื้องต้นเอาไว้ สำหรับนักศึกษาหรือนักวิจัยที่ต้องการพัฒนาต่อ



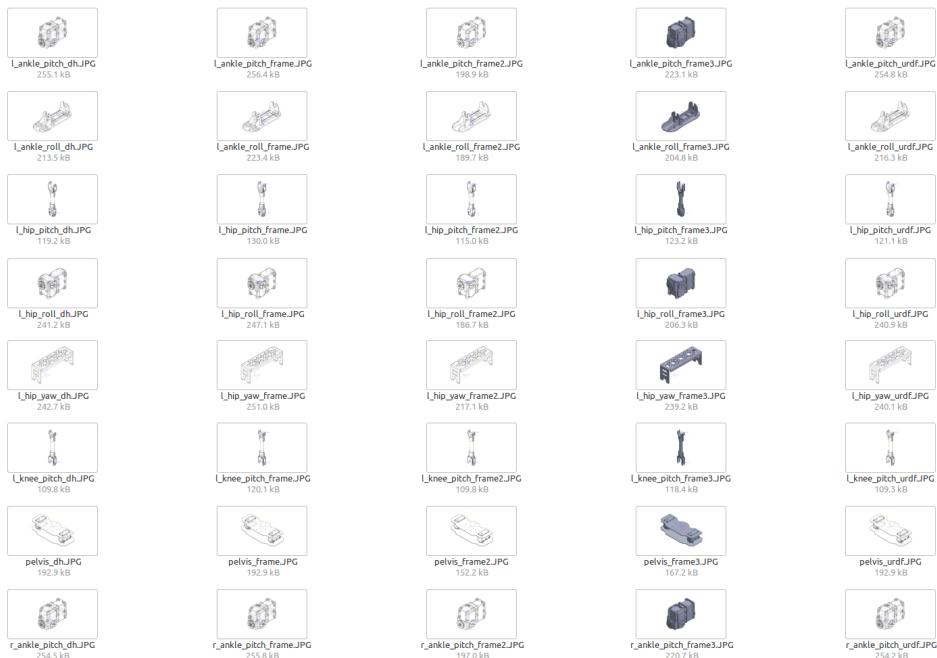
รูปที่ 4.35: GitHub ที่เก็บข้อมูลทั้งหมดของหุ่นยนต์อิฐมานอยด์ UTHAI



รูปที่ 4.36: ตัวอย่าง Wiki การใช้งานเบื้องต้นของหุ่นยนต์อิฐมานอยด์ UTHAI

### 4.3.2 ព័ត៌មានអេរ៉មខែងទុកប្រាក់អិវាមានឈើ UTHAI

អេរ៉មខែងទុកប្រាក់អិវាមានឈើ UTHAI ត្រូវកូដផ្លូវការនៃការកែតម្រូវការនៅក្នុង GitHub ដោយចូលទៅការប្រើប្រាស់នៅ <https://github.com/UTHAI-Humanoid/UTHAI-Hardware/tree/master/Mechanics/Frame>



រូបថត 4.37: រាយអេរ៉មខែងទុកប្រាក់អិវាមានឈើ UTHAI

[UTHAI-Humanoid / UTHAI-Hardware](https://github.com/UTHAI-Humanoid/UTHAI-Hardware)

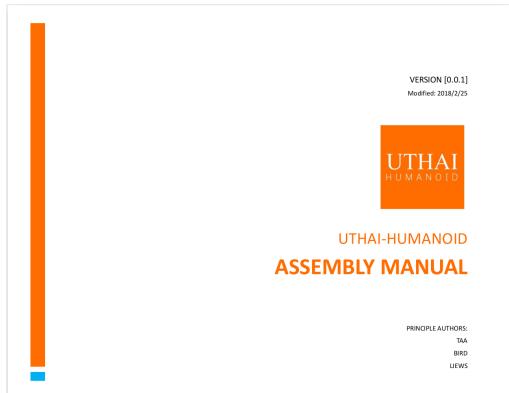
Code Issues Pull requests Insights Settings

Branch: master [UTHAI-Hardware / Mechanics / Frame /](#) Create new file Upload files Find file History

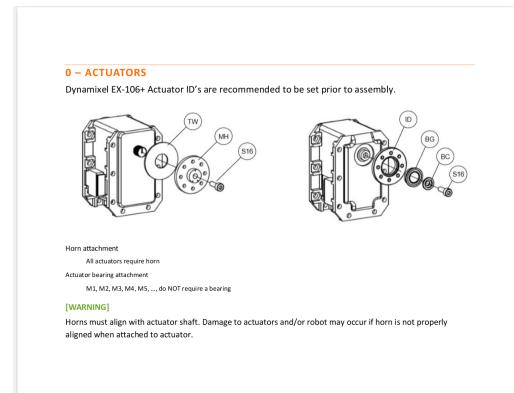
SwellLz	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	Latest commit 18d64b5 on Mar 10
...		
<a href="#">L_ankle_pitch_dh.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">L_ankle_pitch_frame.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">L_ankle_pitch_frame2.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">L_ankle_pitch_frame3.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">L_ankle_pitch_urdf.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">L_ankle_roll_dh.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">L_ankle_roll_frame.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">L_ankle_roll_frame2.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">L_ankle_roll_frame3.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">L_ankle_roll_urdf.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">L_hip_pitch_dh.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">L_hip_pitch_frame.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">L_hip_pitch_frame2.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">L_hip_pitch_frame3.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">L_hip_pitch_urdf.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">L_hip_roll_dh.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">L_hip_roll_frame.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">L_hip_roll_frame2.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">L_hip_roll_frame3.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">L_hip_roll_urdf.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">L_knee_pitch_dh.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">L_knee_pitch_frame.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">L_knee_pitch_frame2.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">L_knee_pitch_frame3.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">L_knee_pitch_urdf.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">pelvis_dh.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">pelvis_frame.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">pelvis_frame2.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">pelvis_frame3.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">pelvis_urdf.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">r_ankle_pitch_dh.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">r_ankle_pitch_frame.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">r_ankle_pitch_frame2.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">r_ankle_pitch_frame3.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago
<a href="#">r_ankle_pitch_urdf.JPG</a>	Merge branch 'master' of <a href="https://github.com/UTHAI-Humanoid/UTHAI-Hardware">https://github.com/UTHAI-Humanoid/UTHAI-Hardware</a> ...	2 months ago

រូបថត 4.38: រាយ GitHub នៃអេរ៉មផែនការកែតម្រូវការទុកប្រាក់អិវាមានឈើ UTHAI

### 4.3.3 ตัวอย่างเอกสารข้อมูลของหุ่นยนต์ชีวมานอยด์ UTHAI

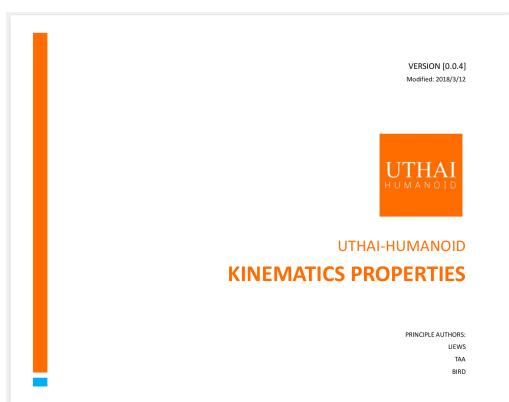


(ก) หน้าปกคู่มือการประกอบหุ่นยนต์ชีวามานอยด์ UTHAI

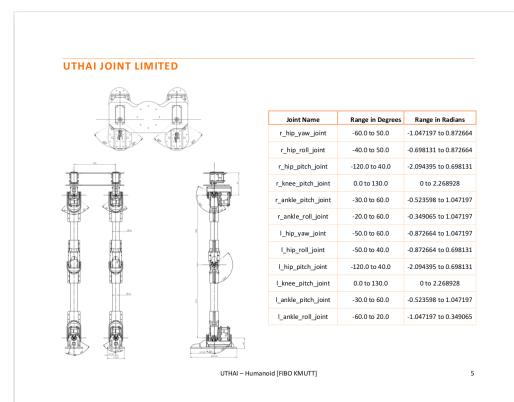


(ข) ตัวอย่างคู่มือการประกอบหุ่นยนต์ชีวามานอยด์ UTHAI

รูปที่ 4.39: UTHAI Assembly Manual



(ก) หน้าปกรายละเอียดของหุ่นยนต์ชีวามานอยด์ UTHAI



(ข) ตัวอย่างรายละเอียดของหุ่นยนต์ชีวามานอยด์ UTHAI

รูปที่ 4.40: UTHAI Kinematics Properties



(ก) หน้าปกรายละเอียดของหุ่นยนต์ชีวามานอยด์ UTHAI

URDF CONVENTION						
Link	Frame	Mass [kg]	Center of mass [m]			
All Robot	frame_base	3.56700000	-0.00836813	0.00000000	-0.31830739	
Link						
Moments of Inertia [ $\text{kg}\cdot\text{m}^2$ ] (Taken at the center of mass)						
All Robot	ixx	iyx	izx	iyy	iyz	izz
	0.31081793	-0.00000313	-0.00035731	0.28471758	-0.00000024	0.03125190

DH CONVENTION						
Link	Frame	Mass [kg]	Center of mass [m]			
All Robot	frame_base	3.56700000	-0.00836813	0.00000000	-0.31830739	
Link						
Moments of Inertia [ $\text{kg}\cdot\text{m}^2$ ] (Taken at the center of mass)						
All Robot	ixx	iyx	izx	iyy	iyz	izz
	0.31081793	-0.00000313	-0.00035731	0.28471758	-0.00000024	0.03125190

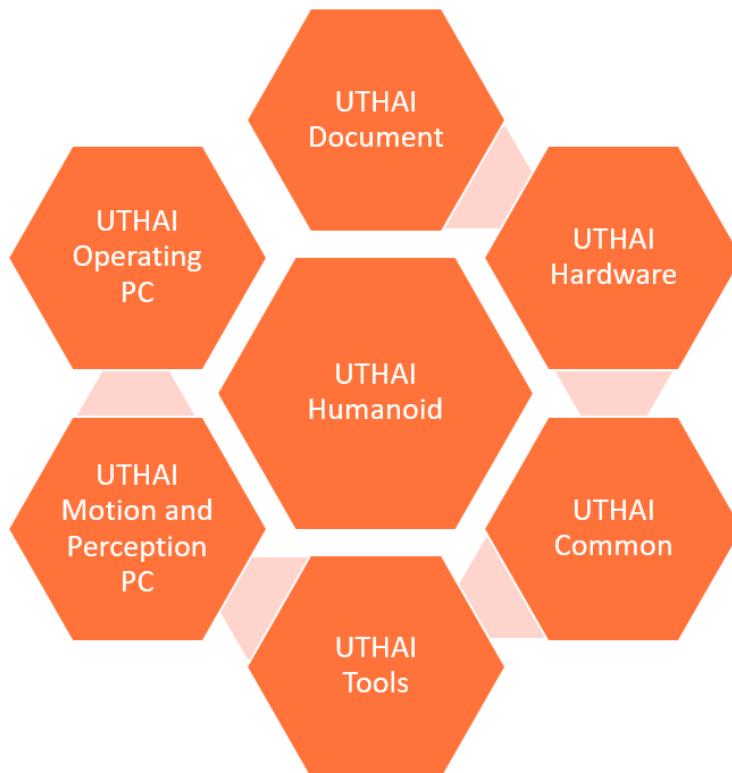
UTHAI – Humanoid [FIBO KMUTT]

(ข) ตัวอย่างรายละเอียดของหุ่นยนต์ชีวามานอยด์ UTHAI

รูปที่ 4.41: UTHAI Dynamics Properties

#### 4.3.4 ภาพรวมระบบพื้นฐานของหุ่นยนต์อิวามาโนย์ UTHAI

ระบบพื้นฐานที่ผู้วิจัยได้ทำการออกแบบเพื่อที่จะทำให้คนที่เข้ามาพัฒนาต่อยอดได้อย่างสะดวก และเป็นระบบระเบียบ มีรูปแบบแบบแผน ซึ่งผู้วิจัยได้วางระบบขึ้นมาจากการประสบการณ์การทำงานของผู้วิจัยเอง รวมถึงได้ค้นคว้าหาความรู้เพิ่มเติม จนคิดว่าระบบจะทำให้หุ่นยนต์อิวามาโนย์สามารถพัฒนาต่อได้ง่าย และเป็นประโยชน์ต่อผู้วิจัยท่านอื่นที่ต้องการนำระบบพื้นฐานนี้ไปใช้งาน



รูปที่ 4.42: ภาพรวมระบบพื้นฐานของหุ่นยนต์อิวามาโนย์ UTHAI

ระบบพื้นฐานที่ผู้วิจัยออกแบบขึ้นมา จะประกอบไปด้วยส่วนสำคัญอยู่ทั้งหมด 6 ส่วน คือ

1. UTHAI-Documents
2. UTHAI-Hardware
3. UTHAI-Common
4. UTHAI-MPPC
5. UTHAI-OPC
6. UTHAI-Tools

## UTHAI-Documents

ในส่วนนี้คือส่วนของงานเอกสารต่างๆที่เกี่ยวข้องกับหุ่นยนต์อิวามาโนยด์

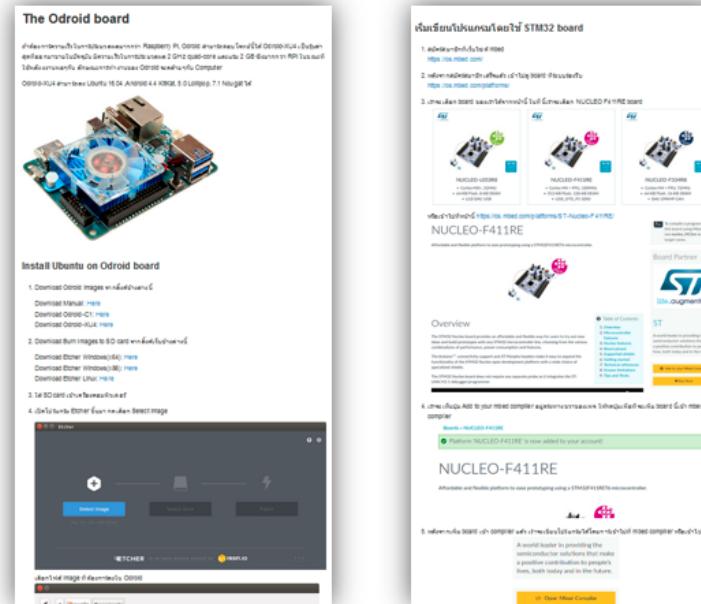
**Reports-Hardware** ใช้สำหรับเก็บรายงานวิทยานิพนธ์ทั้งฉบับร่างและฉบับสมบูรณ์ที่เกี่ยวข้องกับ การปรับเปลี่ยนโครงสร้างทางกล และทางไฟฟ้าของหุ่นยนต์อิวามาโนยด์ UTHAI

- การออกแบบโครงสร้างและพัฒนาระบบที่มีสำหรับหุ่นยนต์อิวามาโนยด์เพื่อการศึกษาและวิจัย

**Reports-Software** ใช้สำหรับเก็บรายงานวิทยานิพนธ์ทั้งฉบับร่างและฉบับสมบูรณ์ที่เกี่ยวข้องกับการเพิ่มความสามารถของหุ่นยนต์อิวามาโนยด์ UTHAI

- การพัฒนาระบบการเคลื่อนที่สำหรับหุ่นยนต์อิวามาโนยด์

Wiki ใช้สำหรับเก็บ Tutorial ที่เกี่ยวกับการใช้งานของหุ่นยนต์อิวามาโนยด์ UTHAI



รูปที่ 4.43: ภาพตัวอย่าง Tutorial ใน Wiki

ผู้วิจัยท่านอื่นสามารถที่จะช่วยกันเขียนและพัฒนาได้โดยการ Clone Repository และทำการแก้ไข ปรับปรุง หลังจากนั้นก็ Pull request ขึ้นมาเพื่อแสดงให้ผู้วิจัยท่านอื่นเห็นด้วย

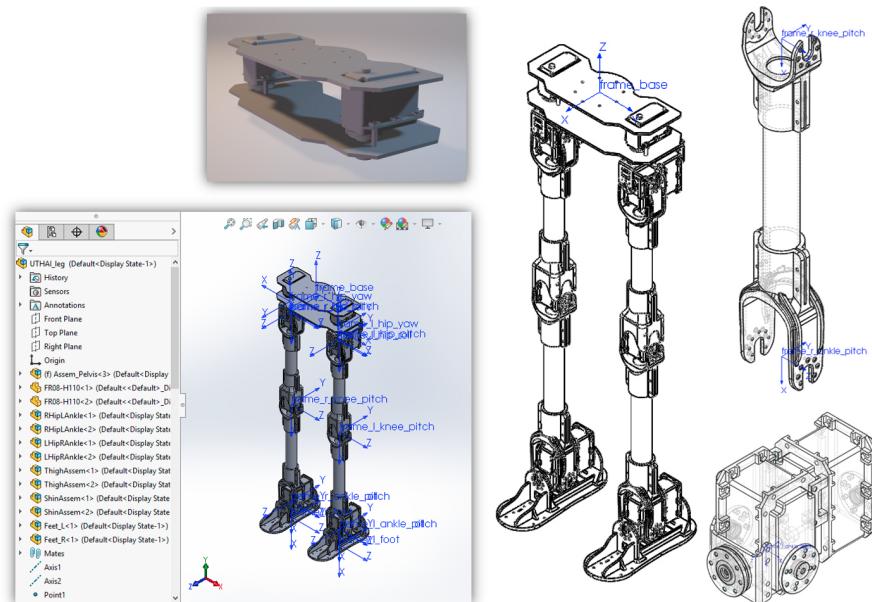
## UTHAI-Hardware

ในส่วนนี้คือส่วนของรายละเอียดเกี่ยวกับโครงสร้างทั้งทางกลและทางไฟฟ้าที่เปิดให้ผู้ใช้งานอื่นสามารถนำไปประยุกต์ใช้ได้

**Drawing** ไฟล์ออกแบบทางวิศวกรรมทุกชิ้นส่วนที่ต้องมีการขึ้นรูป

**STL files** ไฟล์สำหรับการขีนรูปสามมิติและไฟล์สำหรับนำไปทำแบบจำลองหุ่นยนต์ขีวามนอยด์

**Solidworks files** ไฟล์ที่ออกแบบโดยใช้โปรแกรม Solidworks เพื่อให้นำไปแก้ไขปรับปรุงให้ดียิ่งขึ้นได้



รูปที่ 4.44: ภาพตัวอย่างไฟล์ใน UTHAI-Hardware

UTHAI Humanoid											
URDF	Link	Mass [kg]	Center of mass [m]			Inertia [ $\text{kg} \cdot \text{m}^2$ ] taken at the center of mass					
			x	y	z	xx	xy	xz	yy	zz	
	All	3.56700000	-0.00836813	0.00000000	-0.31830739	0.31081793	-0.00000313	-0.00005731	0.28471758	-0.00000024	0.03125190
	Body	0.86500000	-0.00557436	0.00000000	-0.02356998	0.00656431	-0.00000003	-0.00000466	0.00067608	0.00000000	0.00664141
R Hip Yaw		0.09100000	0.00000000	0.02500000	0.00735017	0.00014158	0.00000000	0.00000000	0.00000202	0.00000000	0.00014316
R Hip Roll		0.34300000	0.00000000	-0.01526210	-0.02652545	0.000032449	0.00000081	0.000026847	0.000000219	0.00014760	
R Hip Pitch		0.31800000	0.22137989	0.00000000	0.00000000	0.000011525	0.00000000	0.00000000	0.00000000	0.00000000	0.00250848
R Knee Pitch		0.15100000	0.11451118	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00136731
R Ankle Pitch		0.34300000	-0.01526210	0.02152545	0.00000000	0.000026847	-0.00000219	-0.00000081	0.00014760	0.00000000	0.00032449
R Ankle Roll		0.10500000	0.03625982	-0.00019548	0.000034576	0.000034591	-0.00000013	0.000000857	0.000032705	0.000000120	0.00004813
L Hip Yaw		0.09100000	0.00000000	0.02500000	0.00735017	0.00014158	0.00000000	0.00000000	0.00000202	0.00000000	0.00014316
L Hip Roll		0.34300000	0.00000000	-0.01526210	-0.02652545	0.000032449	0.00000081	0.000026847	0.000000219	0.00014760	
L Hip Pitch		0.31800000	0.22137989	0.00000000	0.00000000	0.000011525	0.00000000	0.00000000	0.00000000	0.00000000	0.00250848
L Knee Pitch		0.15100000	0.11451118	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00136731
L Ankle Pitch		0.34300000	-0.01526210	0.02152545	0.00000000	0.000026847	-0.00000219	-0.00000081	0.00014760	0.00000000	0.00032449
L Ankle Roll		0.10500000	0.03608455	0.000020160	0.00002697	0.000034223	0.00000014	0.000000840	0.000032348	-0.000000118	0.00004816

UTHAI Humanoid											
DH	Link	Mass [kg]	Center of mass [m]			Inertia [ $\text{kg} \cdot \text{m}^2$ ] taken at the center of mass					
			x	y	z	xx	xy	xz	yy	zz	
	All	3.56700000	-0.00836813	0.00000000	-0.31830739	0.31081793	-0.00000313	-0.00005731	0.28471758	-0.00000024	0.03125190
	Body	0.86500000	-0.00557436	0.00000000	-0.02356998	0.00656431	-0.00000003	-0.00000466	0.00067608	0.00000000	0.00664141
R Hip Yaw		0.09100000	0.00000000	0.02500000	0.00735017	0.00014158	0.00000000	0.00000000	0.00000202	0.00000000	0.00014316
R Hip Roll		0.34300000	0.00000000	-0.01526210	-0.02652545	0.000032449	0.00000081	0.000026847	0.000000219	0.00014760	
R Hip Pitch		0.31800000	0.22137989	0.00000000	0.00000000	0.000011525	0.00000000	0.00000000	0.00000000	0.00000000	0.00250848
R Knee Pitch		0.15100000	0.11451118	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00136731
R Ankle Pitch		0.34300000	-0.01526210	0.02152545	0.00000000	0.000026847	-0.00000081	0.00000219	0.000032449	0.00000000	0.00014760
R Ankle Roll		0.10500000	0.03641518	-0.00019548	0.000034576	0.000034591	-0.00000013	0.000004813	-0.000000120	0.000032705	
L Hip Yaw		0.09100000	0.00000000	0.02864983	-0.02500000	0.00014158	0.00000000	0.00000000	0.00014316	0.00000000	0.00002022
L Hip Roll		0.34300000	0.01526210	0.02152545	0.00000000	0.000026847	0.00000219	0.00001760	0.00000000	0.00032449	
L Hip Pitch		0.31800000	0.27862011	0.00000000	0.00000000	0.000011525	0.00000000	0.00000000	0.00000000	0.00000000	0.00250848
L Knee Pitch		0.15100000	0.15211782	0.00000000	0.00000000	0.000006340	0.00000000	0.00000000	0.00139612	0.00000000	0.00136731
L Ankle Pitch		0.34300000	-0.01526210	0.02152545	0.00000000	0.000026847	-0.00000081	0.00000219	0.000032449	0.00000000	0.00014760
L Ankle Roll		0.10500000	0.03471545	-0.0002697	0.000020160	0.000034223	-0.00000014	0.000003840	0.00000014	0.00000000	0.000032448

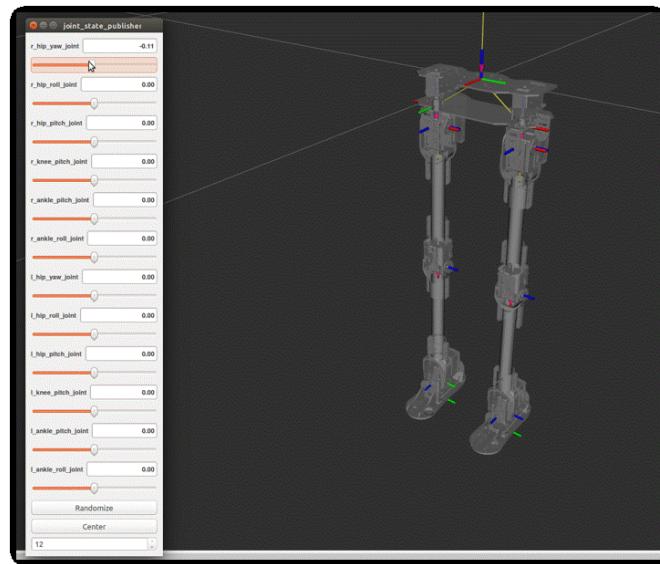
รูปที่ 4.45: ภาพตัวอย่างไฟล์ใน UTHAI-Hardware

### UTHAI-Common

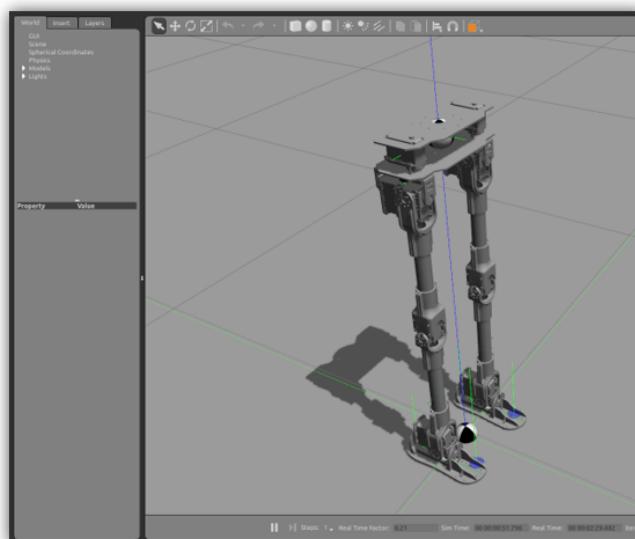
ในส่วนนี้คือส่วนของแพกเกจ ROS ที่ใช้เป็นพื้นฐานสำหรับการแสดงผลด้วยภาพ และระบบจำลองโดยจะมีแบบจำลองของหุ่นยนต์ชีวามโนยด์ UTHAI ในรูปแบบของ URDF Xacro

**uthai\_description** เป็นแพกเกจที่เขียนอธิบายลักษณะของหุ่นยนต์ชีวามโนยด์เพื่อให้ ROS สามารถนำไปใช้ในกระบวนการอื่นได้

**uthai\_gazebo** เป็นแพกเกจที่เอาไว้สำหรับทำระบบจำลองการทำงานของหุ่นยนต์ชีวามโนยด์ UTHAI



รูปที่ 4.46: ภาพ RViz ใน uthao\_description



รูปที่ 4.47: ภาพ Gazebo ใน uthai\_gazebo

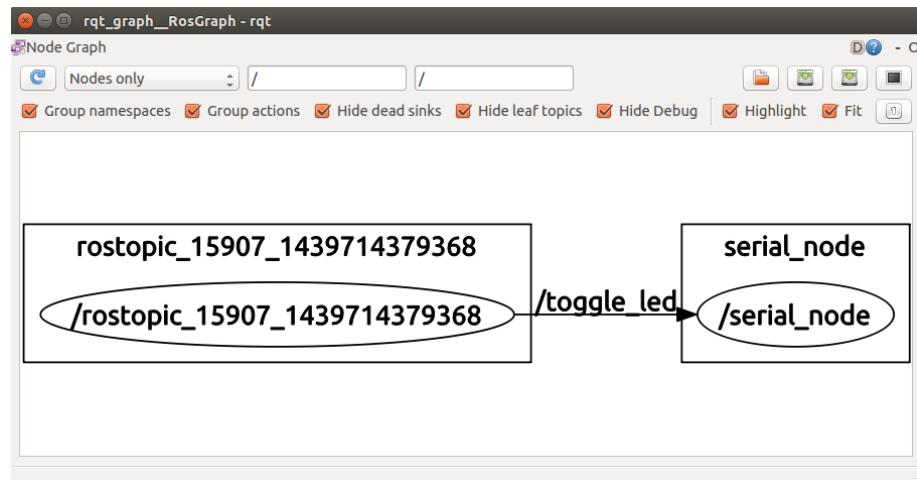
### UTHAI-MPPC

ในส่วนนี้คือส่วนของแพกเกจ ROS ที่ใช้เป็นตัวในการเชื่อมต่อกับอุปกรณ์อาร์ดเวย์ของหุ่นยนต์อิวามาโนย์ด์ UTHAI เพื่อส่งการตัวขับเคลื่อนดิจิตอลเซอร์โว และอ่านค่าเซนเซอร์จากตัวรับสัญญาณวัดความเร็ว การตรวจจับฝ่าเท้า ตำแหน่งและความเร็วของตัวขับเคลื่อน

**uthai\_mbed** เป็นแพกเกจที่เขียนเพื่อเชื่อมต่อกับหน่วยประมวลผลระดับต่ำผ่าน rosserial เขียนด้วยภาษา python โดยจะรับค่าเซนเซอร์หน่วยวัดความเร็วอย่างเรียลไทม์ ผ่านเซอร์ตรัจจับพื้น

**uthai\_manager** เป็นแพกเกจที่เขียนเพื่อเอาไว้สำหรับจัดการตัวขับเคลื่อนทั้งหมด ให้สามารถสั่งการได้รวมถึง config ต่างๆด้วย

**uthai\_bringup** เป็นแพกเกจที่เขียนเพื่อใช้ในการสั่งการตัวขับเคลื่อน ดิจิตอลเซอร์โว ที่เชื่อมต่อกับตัวประมวลผลระดับสูง



รูปที่ 4.48: ภาพ rqt\_graph ของ serial\_node

### UTHAI-OPC

ในส่วนนี้คือส่วนของแพกเกจ ROS ที่ใช้เป็นตัวในการเชื่อมต่อระหว่างตัวประมวลผลระดับสูงกับคอมพิวเตอร์ภายนอกเพื่อใช้ในการควบคุม หรือการแสดงผลการทำงานที่ต้องใช้การคำนวณสูง เป็นการแบ่งเบาภาระการประมวลผลของตัวประมวลผลระดับสูง

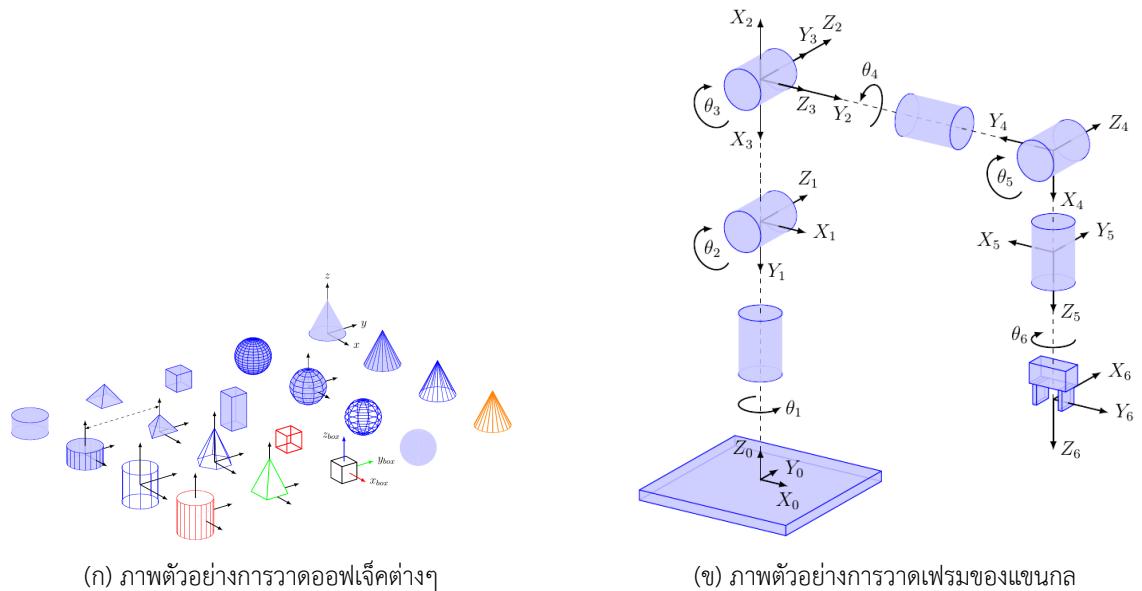
### UTHAI-Msgs

ในส่วนนี้คือส่วนของแพกเกจ ROS ที่ใช้เป็นตัวในการเก็บ Messages ที่ใช้ในการติดต่อสื่อสารกันภายในระบบ ซึ่ง Messages ไม่เป็นมาตรฐาน โดยอาจรวมไปถึง services และ actions ของระบบหุ่นยนต์อิวามาโนย์ด์ UTHAI ด้วย

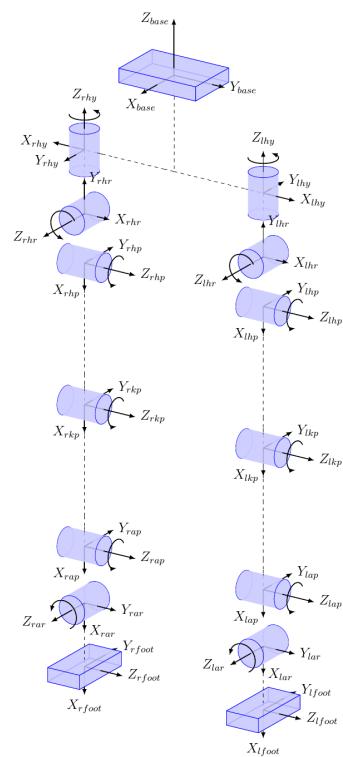
### UTHAI-Tools

ในส่วนนี้คือส่วนของรายละเอียดเครื่องมือที่ช่วยทำให้การทำงานมีประสิทธิภาพมากยิ่งขึ้น

**sketch-lib** เป็นเครื่องมือที่ใช้สำหรับเอาไว้วัดรูปเฟรมของหุ่นยนต์



รูปที่ 4.49: ภาพตัวอย่างการวัดเฟรมโดยใช้เครื่องมือนี้



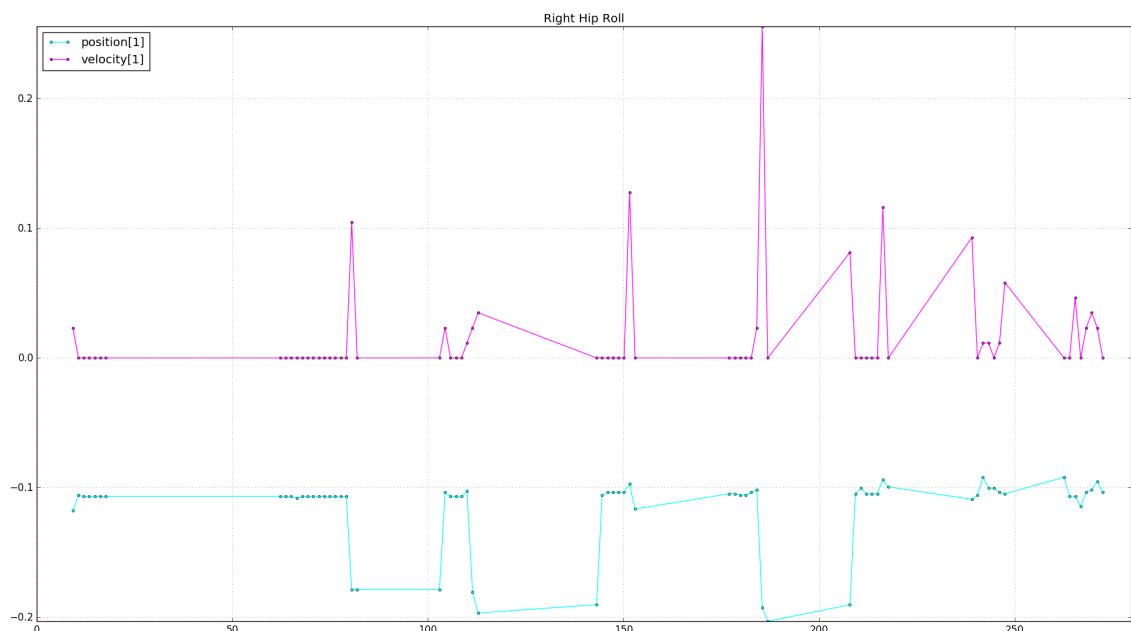
รูปที่ 4.50: ภาพตัวอย่างการวัดเฟรมของหุ่นยนต์ชีวนิรภัย

## 4.4 ผลการทดลอง

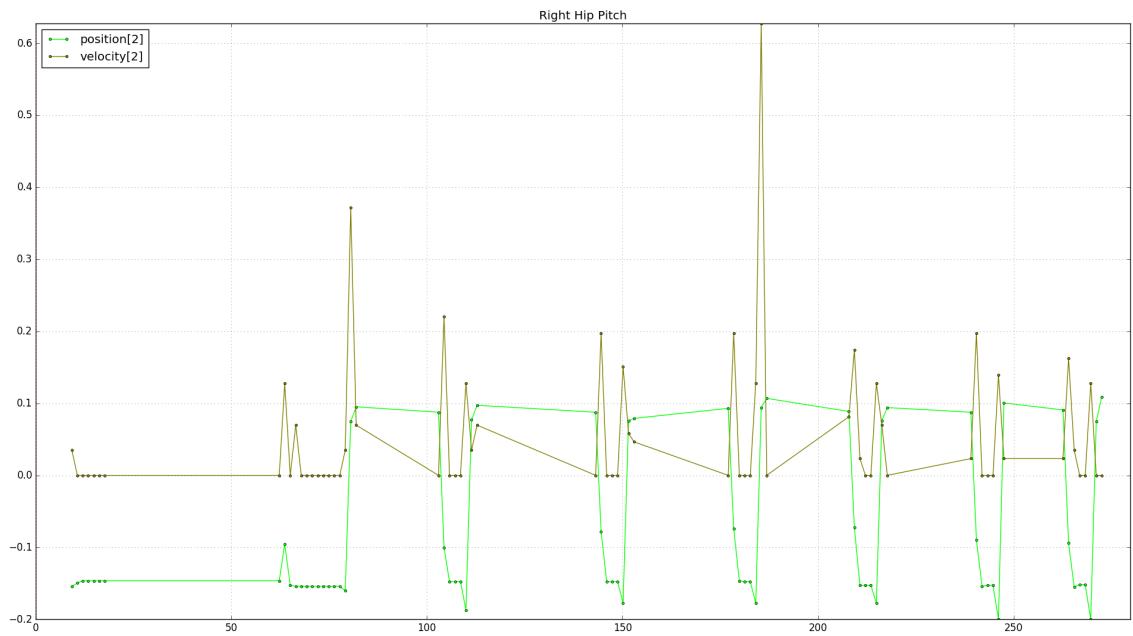
### 4.4.1 การทดลองการเดิน

การทดลองการเดินนี้ได้ทดลองด้วยการปรับค่าข้อต่อให้เหมาะสมแก่การเดิน เพื่อทดสอบโครงสร้างที่ได้ออกแบบ มาว่าสามารถรองรับการเดินของหุ่นยนต์ได้จริงหรือไม่ โดยค่าที่ใช้ในการปรับนั้นจะเป็นค่ามุมของมอเตอร์ส่วนต่างๆของส่วนขาทั้ง 2 ข้าง โดยจะตั้งชื่อมอเตอร์ตามแกนของข้อต่อแต่ละส่วนโดยเริ่มจากส่วนจะโพกจะมี 3 แกนคือ hip yaw, hip roll, hip pitch ส่วนหัวเข่า 1 แกนคือ knee pitch และส่วนข้อเท้า 2 แกนคือ ankle pitch, ankle roll โดยในขาแต่ละข้างจะแยกด้วยสัญลักษณ์ Right(ขาข้างขวา) และ Left(ขาข้างซ้าย) โดยอิงจากตัวของหุ่นยนต์ และสังงานให้ไปตามมุมต่างๆที่กำหนดไว้ดังภาพ

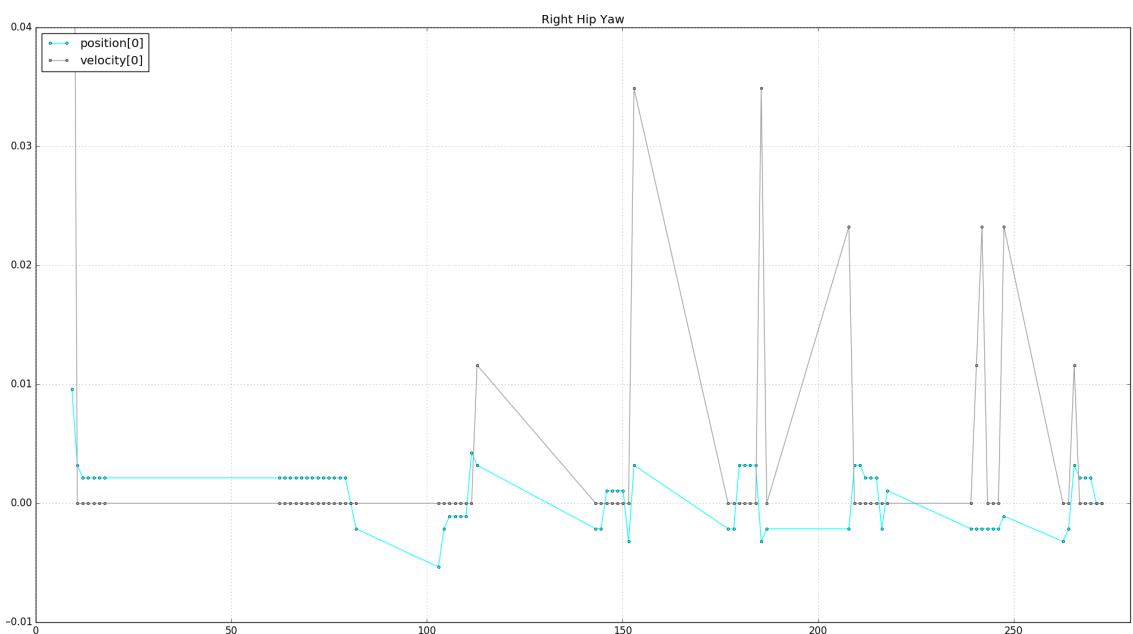
รูปกราฟแสดงตำแหน่งมอเตอร์ขาขวา



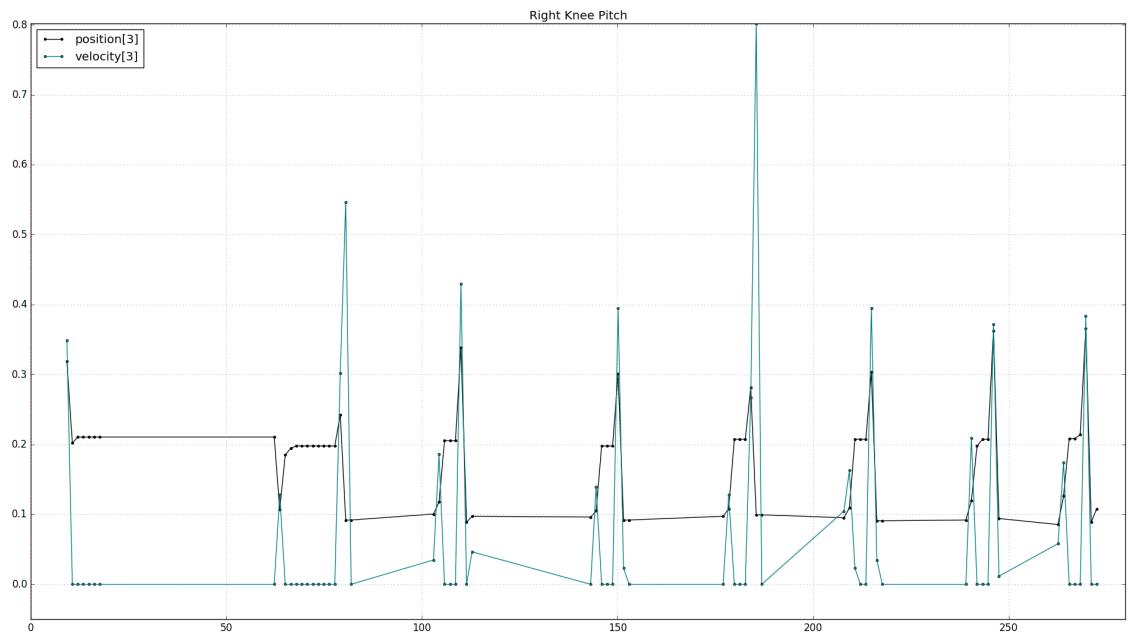
รูปที่ 4.51: รูปการสั่งงานข้อต่อ right hip roll



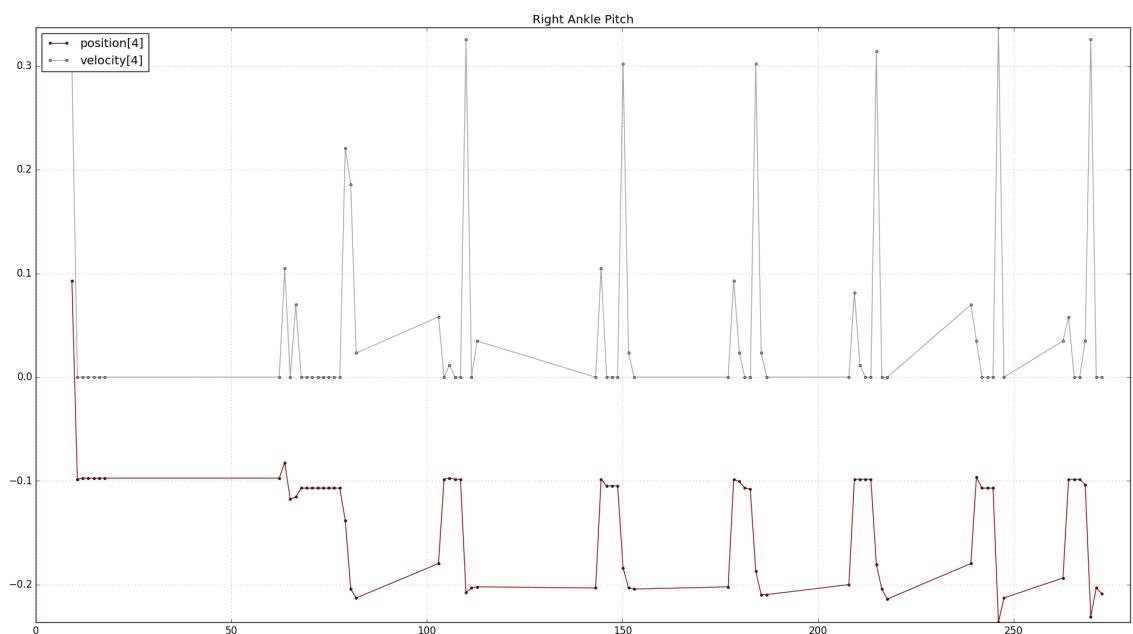
รูปที่ 4.52: รูปการสั่งงานข้อต่อ right hip pitch



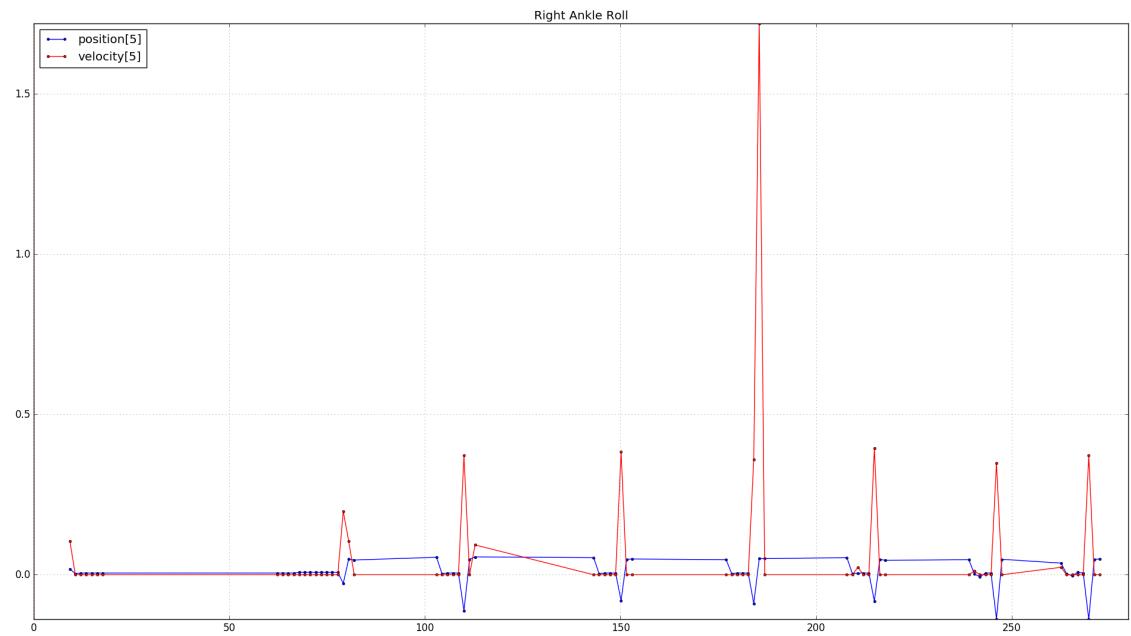
รูปที่ 4.53: รูปการสั่งงานข้อต่อ right hip yaw



รูปที่ 4.54: รูปการสั่งงานข้อต่อ right knee pitch

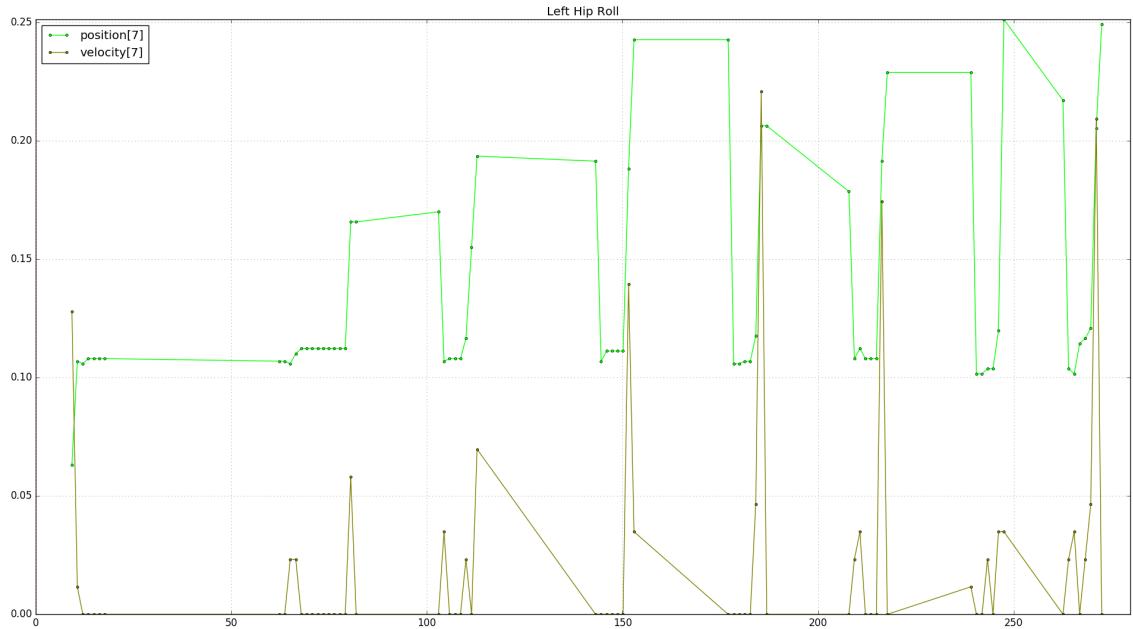


รูปที่ 4.55: รูปการสั่งงานข้อต่อ right ankle pitch

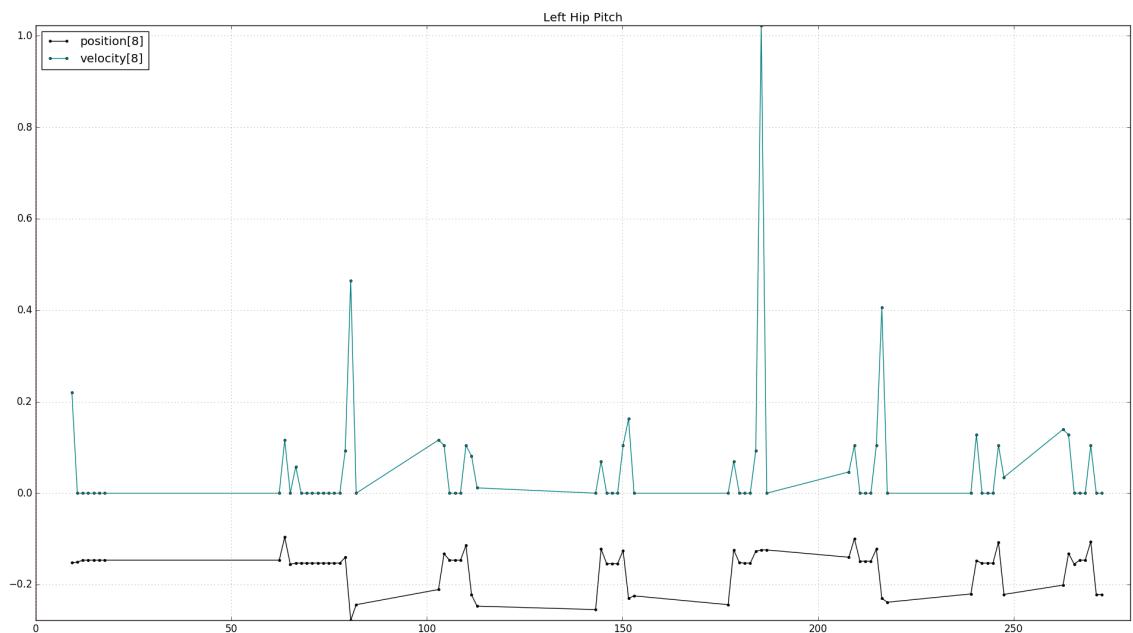


รูปที่ 4.56: รูปการสั่นงานข้อต่อ right ankle roll

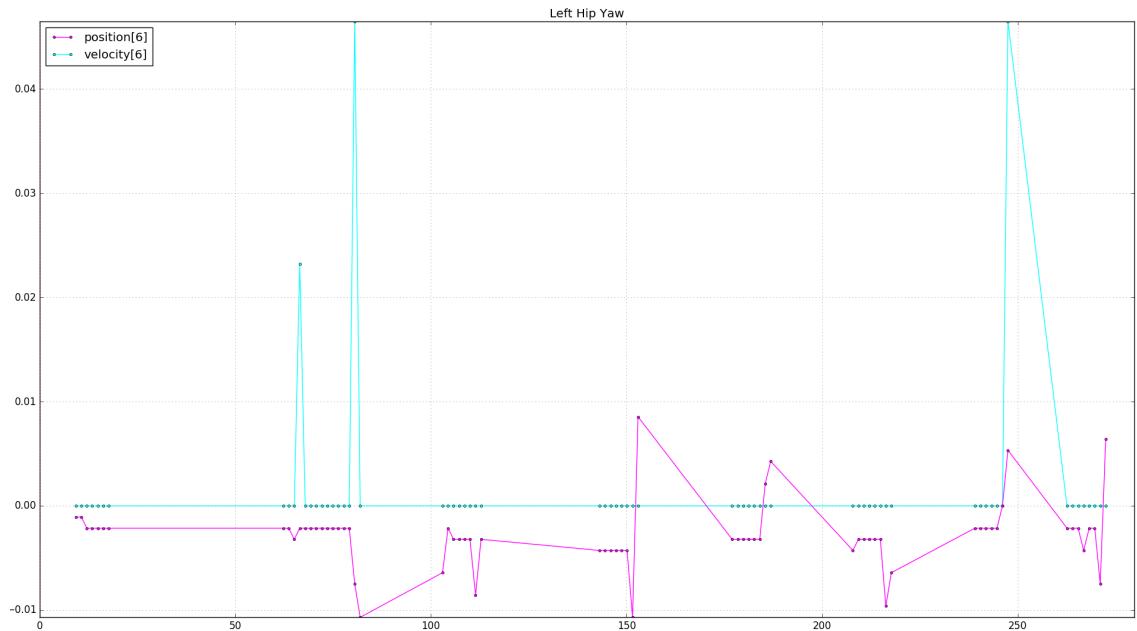
### รูปกราฟแสดงตำแหน่งและความเร็วของตัวอักษรข้าม



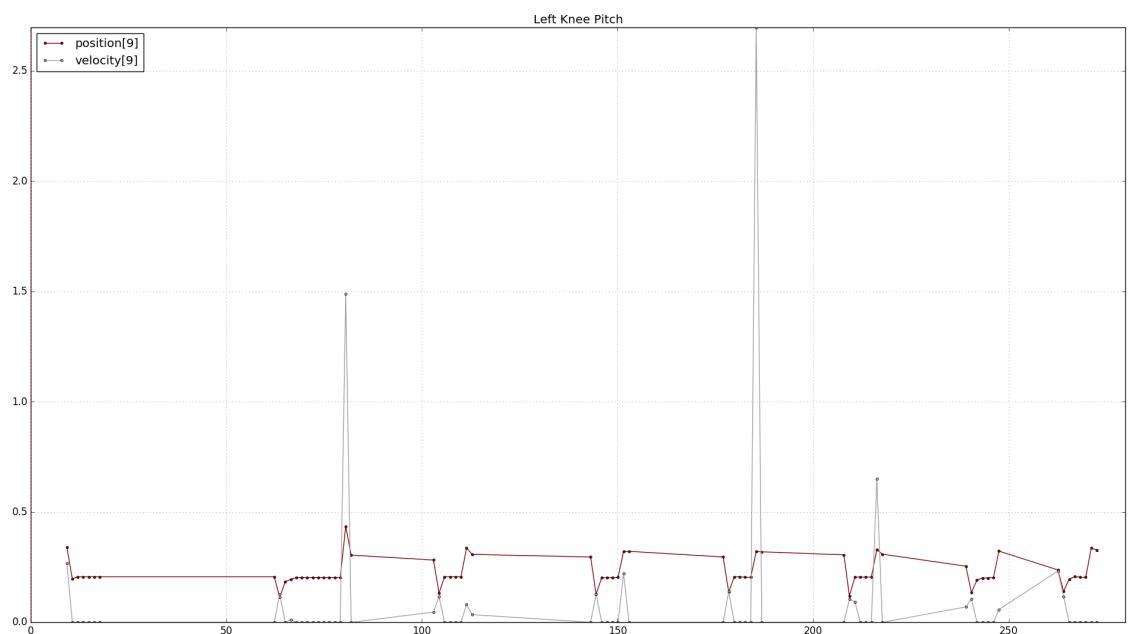
รูปที่ 4.57: รูปการสั่งงานข้อต่อ left hip roll



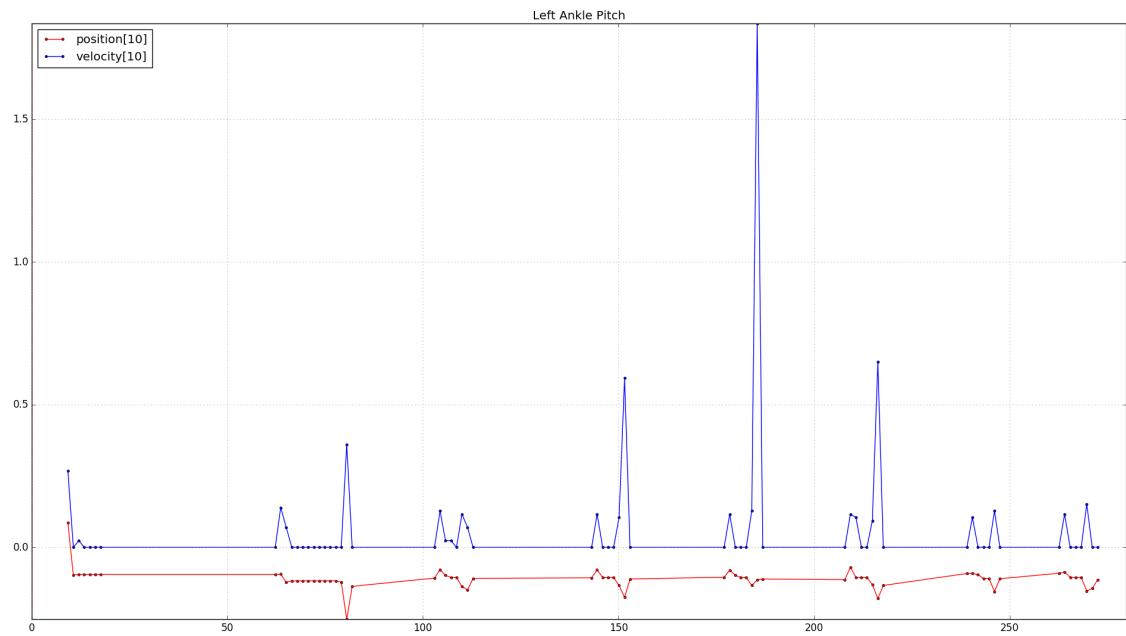
รูปที่ 4.58: รูปการสั่งงานข้อต่อ left hip pitch



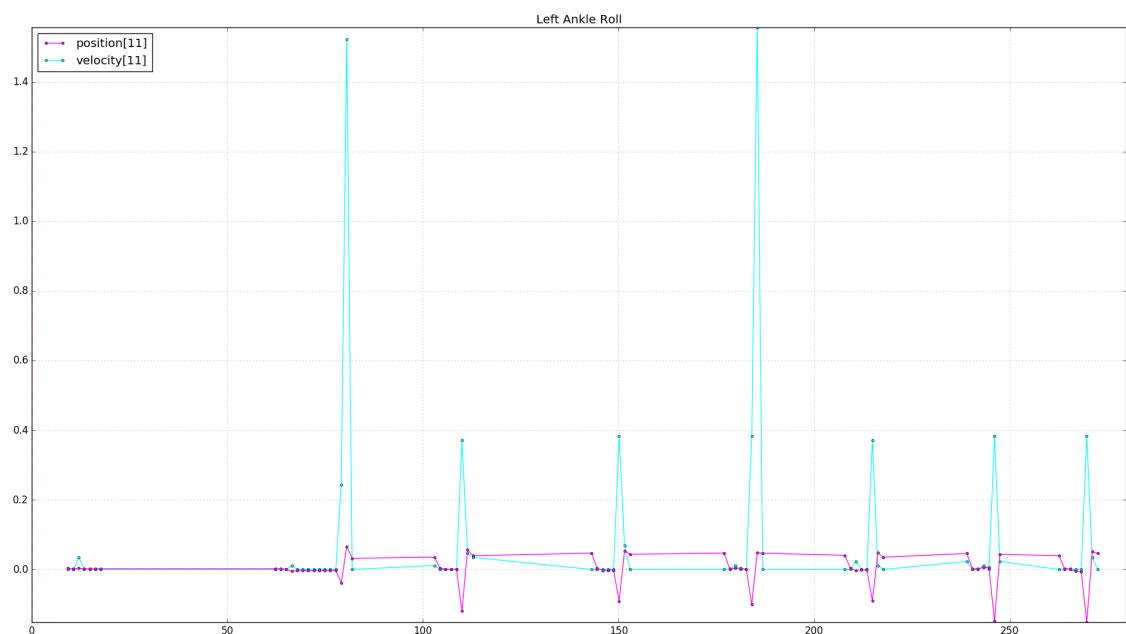
รูปที่ 4.59: รูปการสั่งงานข้อต่อ left hip yaw



รูปที่ 4.60: รูปการสั่งงานข้อต่อ left knee pitch



รูปที่ 4.61: รูปการสั่งงานข้อต่อ left ankle pitch



รูปที่ 4.62: รูปการสั่งงานข้อต่อ left ankle roll

จากการที่แสดงข้างต้นนั้นแสดงถึงตำแหน่งที่สั่งงานให้หุ่นยนต์ขยับในองศาต่างๆของข้อต่อ โดยจะสามารถสังเกตได้จาก จุด peak ของแต่ละกราฟซึ่งมี 7 จุดในแต่ละกราฟที่เกิดขึ้นในควบคู่กันเดียวกัน โดยค่าในแกน x แสดงถึงช่วงเวลาที่สั่งงานเป็นวินาที และแกน y แสดงองศาของมอเตอร์ในหน่วยเรเดียล โดยในการทดลองนี้ เป็นการกำหนดค่าเพื่อทดลองให้หุ่นยนต์มีการเดินเกิดขึ้น และทดสอบโครงสร้างขณะทำการทดลองอีกด้วย ซึ่งผลการทดลองที่ได้จะพบว่าการเดินครั้งที่ 1-4 เกิดการล้มขึ้น และหลังจากครั้งที่ 5 เป็นต้นไป หุ่นยนต์สามารถเดินได้ 1 ก้าวจากการทดลองเปลี่ยนค่าให้เหมาะสมกับระบบ ดังแสดงในรูปภาพต่อไปนี้



(ก) รูปการล้มครั้งที่ 1

(ข) รูปการล้มครั้งที่ 2



(ค) รูปการล้มครั้งที่ 3

(ง) รูปการล้มครั้งที่ 4



(จ) รูปการก้าวสำเร็จ

รูปที่ 4.63: รูปการทดลองการเดินของหุ่นยนต์ UTHAI

## 4.4.2 ปัญหาที่พบและการแก้ไข

### 4.4.2.1 อุณหภูมิ

เมื่อมีการใช้งานเป็นเวลานานจะส่งผลให้มอเตอร์ส่วนสะโพกและข้อเท้ามีความร้อนสูงถึง 50 องศาเซลเซียส เนื่องจากมีการสั่นงานให้มอเตอร์อยู่ในตำแหน่งที่กำหนด ซึ่งเมื่อมีแรงบิดข้างนอกมากเกินข้อง จะทำให้มอเตอร์นั้น พยายามรักษาอุณหภูมิของตนเองไว้ ส่งผลให้เกิดความร้อนเกิดขึ้นและ ส่งผลให้มอเตอร์อ่อนแรงลงเนื่องจากความร้อนที่เกิดขึ้น ในการแก้ไขปัญหานี้สามารถเพิ่มเติมส่วนของการระบายอากาศซึ่งจะส่งผลให้ตัวของหุ่นยนต์นั้นมีน้ำหนักมากขึ้น หรือ หยุดการทำงาน ไว้สักระยะเพื่อให้มีการระบายอากาศให้อยู่ในอุณหภูมิปกติ

### 4.4.2.2 โครงสร้าง

เนื่องจากการยืดหักการบอนไฟเบอร์นั้นได้ทำการยืดด้วยการบีบอัดเพื่อให้เกิดแรงเสียดทางสูงแต่ว่าเมื่อมีการสั่นงานที่มีการกระชากรักษาคือ มีการเคลื่อนที่ของมุมมอเตอร์ที่มีความเร็วสูง จะส่งผลให้เกิดแรงบิดตามแนวแกนซึ่งเกินแรงเสียดทานที่การยืดติดจะรับไหว จึงเกิดการบิดของข้อต่อ เกิดขึ้น ส่งผลให้ ส่วนของเท้ามีการบิดไปจากท่าปกติ ซึ่งสามารถแก้ไขได้โดยเพิ่มแรงยืดด้วยการเสริมแผ่นยางบางๆบริเวณหน้าสัมผัสที่ติดกับห้องการบอนไฟเบอร์ เพื่อเพิ่มแรงยืดไปอีกขั้นหนึ่ง

### 4.4.2.3 มอเตอร์

เนื่องจากว่าตัวหุ่นยนต์มีการใช้มอเตอร์ในการขับเคลื่อนโดยตรงกับข้อต่อ ซึ่งเป็นผลทำให้ต้องใช้มอเตอร์ที่มีแรงบิดสูง เพื่อขับเคลื่อนให้โครงสร้างมีการขยายตัวได้ แต่เนื่องจากว่าตัวหุ่นยนต์นั้นออกแบบมาให้มีความสูงที่ระดับ 1 เมตร ซึ่งหมายถึงจะมีแรง โน้ม-men ที่เกิดขึ้นกับข้อต่อเมื่อหุ่นยนต์มีการก้าวเท้าหรือขยายตัว เป็นผลทำให้เกิด backlash เนื่องจากว่าตัวของมอเตอร์เองไม่สามารถควบคุมตำแหน่งของตนเองให้อยู่ ในมุมที่สั่นงานไว้ได้ จึงแก้ไขปัญหาโดยการเพิ่มระยะการสั่นงานเพื่อให้ครอบคลุมช่วง backlash เพื่อที่จะให้ไปยังมุมที่ต้องการได้ และยังเพิ่มกำลังไฟให้กับมอเตอร์ จาก 12V เป็น 14V เพื่อให้มีแรงบิดที่เพิ่มมากขึ้นกว่าเดิม ซึ่งจะส่งให้ backlash น้อยลงและสามารถควบคุมตำแหน่งง่ายขึ้น

## บทที่ 5

### สรุปผลการทดลองและข้อเสนอแนะ

#### 5.1 การออกแบบโครงสร้างของหุ่นยนต์

การออกแบบโครงสร้างหุ่นยนต์อิมามานอยด์ UTHAI มุ่งเน้น 2 ส่วนเป็นหลักคือ

- 1 สามารถสร้างขึ้นได้ง่าย
- 2 น้ำหนักเบา

ผู้วิจัยจึงเลือกที่จะประยุกต์ใช้เทคนิคการพิมพ์ขึ้นรูปสามมิติด้วยเครื่องพิมพ์สามมิติ ใน การขึ้นรูปข้อต่อส่วนต่างๆของหุ่นยนต์อิมามานอยด์อุทัย และก้านต่อได้เลือกใช้วัสดุเป็นคาร์บอนไฟเบอร์ ซึ่งเป็นวัสดุที่มีคุณสมบัติคือ เบา และแข็งแรง เมื่อเทียบกับวัสดุอื่นๆ ทำให้หุ่นยนต์อิมามานอยด์อุทัยมีน้ำหนักเบา การเชื่อมต่อระหว่างข้อต่อ กับก้านต่อคาร์บอนไฟเบอร์ ผู้วิจัยใช้วิธีการบีบเพื่อสร้างแรงเสียดทานในการยึดติด เนื่องจากการเจาะห้องเครื่องบันไฟเบอร์จะทำให้ไฟเบอร์ขาด ซึ่งส่งผลต่อความแข็งแรงของห้องเครื่องบันไฟเบอร์เป็นอย่างมาก อีกทั้งเมื่อมีการเคลื่อนไหวและรับแรงในแนวต่างๆจะทำให้รูที่เจาะขยายและคลอนได้ส่งผลต่อความแม่นยำโดยรวมของหุ่นยนต์ แต่การยึดติดด้วยวิธีการบีบกับห้องเครื่องบันไฟเบอร์นั้นมีปัญหาเกิดขึ้นคือ มีโอกาสที่จะประกอบโครงสร้างของหุ่นยนต์ไม่ตรง เพราะอาจเกิดการหมุนตามแนวยาวของขั้นส่วน และหากใช้งานต่อเนื่องจะทำให้เกิดการหมุนเลื่อนตามแนวยาวของห้องได้ แต่มีข้อดีคือ เมื่อมีการบิดตามแนวแกนของห้องเครื่องบันเกิดขึ้นแทนที่ขั้นส่วนจะเกิดการแตกหักเนื่องจาก แรกบิดนั้น ตรงส่วนที่ทำการยึดด้วยการบีบอัดนั้นจะเป็นตัวกลางรับแรงบิดแทน ส่งผลให้การยึดมีการบิดเปลี่ยนรูปตามแนวยาวของห้อง แต่ก็สามารถบิดกลับมาให้คงรูปเดิมได้

ส่วนของการเดินของหุ่นยนต์นั้นได้พบว่าเมื่อหุ่นยนต์มีการเดินจริงจะมี backlash เกิดขึ้นในส่วนของตัวมอเตอร์และชุดเฟือง และการเปลี่ยนรูปร่าง ของข้อต่อที่เชื่อมกับมอเตอร์ ซึ่งทำให้การทดสอบเดินนั้นเป็นไปได้อย่างยากลำบาก โดยจะต้องผ่อนค่าเพื่อให้หุ่นยนต์เข้าที่ตามท่าทางที่กำหนด

#### ข้อเสนอแนะ

การพัฒนาต่อยอดควรปรับปรุงในส่วนการเชื่อมต่อเข้าด้วยกันระหว่างมอเตอร์และก้านต่อ ซึ่งอาจจะเพิ่มด้วยวิธีเพิ่มรอยบางเพื่อให้ศูนย์ของมอเตอร์ตรงกัน และการประกอบโครงสร้างควรใช้วัสดุแผ่นยางบางมาคั่นกลางระหว่างหน้าสัมผัสที่ห้องเครื่องบันไฟเบอร์ซึ่งแผ่นยางจะสัมผัสถกับขั้นส่วนที่พิมพ์จากเครื่องพิมพ์สามมิติทำให้มีแรงยึดเกาะเพิ่มมากขึ้น

ส่วนของมอเตอร์ เนื่องจากว่าหุ่นยนต์ที่ออกแบบนี้มีการออกแบบให้มีความสูง 1 เมตร ซึ่งทำให้ส่วนข้างของหุ่นยนต์มีความยาวที่มาก เป็นผลทำให้ สามารถเกิดโมเมนต์เมื่อกระทำการเดิน จึงส่งผลทำให้หุ่นยนต์เกิด backlash ที่ตัวของมอเตอร์เอง เพราะเนื่องจากตัวของมอเตอร์ไม่สามารถควบคุม ตำแหน่งของตนให้อยู่ในตำแหน่งที่สั่งการໄไปได้ ใน การแก้ไขนี้สามารถแก้ไขได้โดย การลดขนาดความยาวของขาลงเพื่อให้เกิดโมเมนต์ที่มอเตอร์ส่วนสะโพกที่น้อยกว่าเดิม หรือเปลี่ยนมอเตอร์ให้สามารถรับแรงบิดที่สูงกว่าตัวปัจจุบัน เพราะสามารถคงสภาพมุ่ที่สั่งการได้อย่างแม่นยำและเกิด backlash น้อยกว่าตัวปัจจุบันได้มาก และสามารถแก้ไขเรื่องอุณหภูติที่เกิดขึ้นเมื่อมีการใช้งานเป็นเวลานาน ซึ่งตัวมอเตอร์ปัจจุบันมีมีการใช้งานสักระยะหนึ่ง มอเตอร์จะเกิดการอ่อนแรง เนื่องจากความร้อน

## 5.2 การออกแบบโปรแกรมด้วย ROS

การออกแบบโปรแกรมของหุ่นยนต์ชีวามโนยด์ UTHAI ผู้วิจัยได้ใช้ ROS เป็นเครื่องมือที่ช่วยในการทำงาน ในวิทยานิพนธ์นี้ส่วนของโปรแกรมจะแบ่งออกเป็น 3 ส่วนหลักคือ

- 1 การแสดงผลภาพด้วย RViz
- 2 การจำลองการทำงานของหุ่นยนต์
- 3 การควบคุมการทำงานของหุ่นยนต์ชีวามโนยด์อุทัย

ผู้วิจัยได้ใช้ ROS เป็นเครื่องมือที่ช่วยในการทำงานเนื่องจากเป็นสถาปัตย์ที่มีความสามารถในการทำงานได้ในส่วนของการแสดงผลด้วยภาพผ่านโปรแกรม RViz ผู้วิจัยได้เขียนไฟล์ URDF ไว้สำหรับหุ่นยนต์ชีวามโนยด์ UTHAI โดยมีได้ใส่ข้อมูลทางพลศาสตร์ของหุ่นยนต์ไว้ด้วย ในส่วนของการจำลองการทำงานของหุ่นยนต์ ผู้วิจัยได้ใช้โปรแกรมจำลองการทำงานเป็น Gazebo ซึ่งเป็นโปรแกรมที่สามารถใส่ข้อมูลทางพิสิกส์เข้าไปได้ ช่วยทำให้เห็นภาพก่อนการทำงานจริง ในส่วนสุดท้ายการควบคุมการทำงานของหุ่นยนต์ชีวามโนยด์ ผู้จัดทำได้สร้างแพกเกจสำหรับควบคุมการขับเคลื่อนของดิจิตอลเซอร์วิส เพื่อทำให้สะดวกในการใช้งานต่อไป

### ข้อเสนอแนะ

การปรับปรุงครรสร้างไฟล์ให้สามารถรันครั้งเดียวแล้วเปิดทุกโปรแกรมที่ต้องการทำงาน เนื่องจากปัญหาตอนนี้ที่ผู้วิจัยพบเจอคือ เวลาเปิดโปรแกรมเพื่อที่ทำการคำสั่งนั้นต้องใช้หน้าต่างเป็นจำนวนมาก โปรแกรมโดยนั้นจะเป็นส่วนย่อยๆหลายส่วน จึงควรที่จะรวมให้เป็นโปรแกรมเดียว

### 5.3 การออกแบบระบบพื้นฐาน

การออกแบบระบบพื้นฐานที่ผู้วิจัยได้ทำการออกแบบก็เพื่อที่จะทำให้คนที่เข้ามาพัฒนาต่อยอดได้อย่างสะดวก และเป็นระบบระเบียบ มีรูปแบบแบบแผนที่ชัดเจน ซึ่งผู้วิจัยได้วางระบบนี้ขึ้นมาจากประสบการณ์การทำงานของผู้วิจัยเอง รวมถึงได้ค้นคว้าหาความรู้เพิ่มเติมจนคิดว่าระบบนี้จะทำให้หุ่นยนต์ขีวามโนยด์สามารถพัฒนาต่อได้ง่าย และเป็นประโยชน์ต่อผู้วิจัยท่านอื่นที่ต้องการนำระบบพื้นฐานนี้ไปใช้งาน

#### ข้อเสนอแนะ

การออกแบบระบบพื้นฐานมาแล้วนั้นอนาคตอาจจะมีการปรับเปลี่ยนตามความเหมาะสมได้ แต่ความต้องการของผู้วิจัยคือ หากนักวิจัยท่านอื่นต้องการพัฒนาต่อยอด ควรที่จะศึกษาระบบพื้นฐานนี้ก่อน เนื่องจากระบบพื้นฐานนี้ผู้วิจัยได้ใช้ทั้งเวลาและประสบการณ์การทำงาน ออกแบบเพื่อตอบสนองความต้องการในการใช้งานของการทำหุ่นยนต์ขีวามโนยด์

ຮຽນການ វັດຖາປະບວກ	iCub	Poppy	DARWIN-OP	Nao	UTHAI
ໃຫ້ໃນການຮັບຈຳກະບວມການ ຄົດ ຂອງມານຸ່ຍ ແລະ ປັນຍາ ປະຕິບັດ	ໃຫ້ໃນດ້ານການເຮືອນ ຮູ່ແລະ ວິຊາຢືນແຫລາກພາດຕາຍດ້ານ ການຕິດຕັ້ງ	ໃຫ້ໃນ ຈານ ວິຈີຍ ພລາຍ ຕ້ານ ເຫັນ ປົບງາມປະຕິບັດ ວິຈີ ການຕິດຕັ້ງ ແລະການມອງທຶນ	ໃຫ້ໃນ ຈານ ວິຈີຍ ການ ຕຶກໜາ ແລະໃຫ້ກາວຕ່ອຍອດອນາຄືຕ	ເພື່ອ ການ ຕຶກໜາ ແລະ ວິຈີຍ ສໍາຫຼັບກາວຕ່ອຍອດອນາຄືຕ	
High Level Controller	PC104 controller	Ordroid XU4	Intel Atom Z530 (32 bit)	Intel Atom @ 1.6 GHz	Ordroid XU4
Low Level Controller	-	-	ARM STM32F103RE	CortexM3 -	Nucleo f411re
ຮະບັບປິບຕິການ	Linux	Linux(Ubuntu 14.04)	Linux	NAO qi 2.0 (Linux- based)	Linux(Ubuntu 16.04)
Sensor	ກລົງ ໄມໂຄຣົ່າໝ່າ,force sensor, Hall effect	IMU 9 DoF,ກລົງ ເລນັສ ກວ້າ,Dynamixel motor	3-axis accelerome- ter,ກລົງ,ໄມໂຄຣົ່າໝ່າ force sensor(4 FRS/foot)	Infrared Sensor, Ultrasonic Sensor	IMU, Force foot), IMU 9 Dof, Dynamixel motor
ວິສດຖານໂຄຮັງສັງເຮົາ	Aluminum (Al6082) Steel 17-4PH	alloy Stainless Steel 17-4PH	3D-printed PLA	Aluminum Alloy 5052 Plastic	3D-printed PLA, car- bonfiber, Aluminum alloy
ອັນດາລືສະ	53	25	20	25	16
ຄວາມສູງ (ມື້ທີ່)	1.04	0.83	0.4545	0.58	1.00
ນ້ຳໜັກ (ກີໂລກຽມ)	22	3.5	2.9	4.3	4.955 (ມື້ຮວມແບຕເຕົວຮູ່)
ແທດໜັກສັງເຮົາ	ແທດໜັກສັງເຮົາ ຈາກສາຍ Cable	ແທດໜັກສັງເຮົາ ຈາກສາຍ Cable	ແບຕເຕົວຮູ່ 12V	ແບຕເຕົວຮູ່ ຈາຍກາຍນອກຈາກສາຍ cable	ແບຕເຕົວຮູ່ 12V ທີ່ສູ່ແທດໜັກ ຈາຍກາຍນອກຈາກສາຍ cable

ຕາງ່າງທີ 5.1: ຕາງ່າງປະເຮົບທີ່ມີນຳທີ່ມານອດ UTHAI ແລະທີ່ນີ້ມີ open source ຕ້ອອນໆ

ภาคผนวก

## ภาคผนวก ก

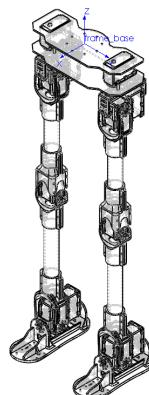
### ข้อมูลเบื้องต้นของหุ่นยนต์อิวามานอยด์ UTHAI

#### ก.1 ค่าคุณสมบัติทางพลศาสตร์

ข้อมูลพลศาสตร์ของหุ่นยนต์อิวามานอยด์ UTHAI ซึ่งจะนำไปใช้ในการทำระบบจำลองด้วยโปรแกรม Gazebo ใน ROS และใช้ในการคำนวณทางคณิตศาสตร์เพื่อทำให้การเดินมีเสถียรภาพ โดยข้อมูลชุดนี้ได้มาจากการคำนวณ Mass Properties ในโปรแกรม SolidWorks และปรับมีค่าใกล้เคียงกับของจริงโดยการเทียบกับเครื่องซึ่งน้ำหนัก

ข้อมูลชุดนี้ประกอบไปด้วย มวล จุดศูนย์กลางมวล และโมเมนต์ความเฉื่อย อีกทั้งข้อมูลยังบอกในมาตรฐาน URDF กับ DH-Parameter ซึ่งทำให้ใช้งานในระบบการคำนวณที่ต่างกันได้

#### Overall Humanoid

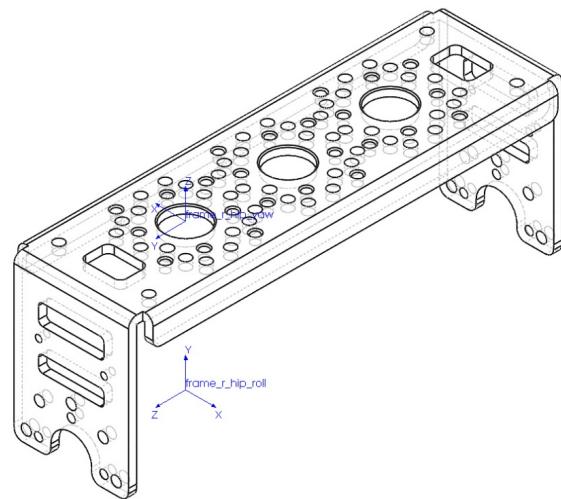


รูปที่ ก.1: ภาพแสดงช่วงล่างทั้งตัว

Link	All Link
Mass (kg)	3.31477475
CoM X (m)	-0.00855772
CoM Y (m)	0.00000000
CoM Z (m)	-0.33375492
Inertia Ixx	0.28641029
Inertia Ixy	-0.00000302
Inertia Ixz	-0.00048106
Inertia Iyy	0.26207601
Inertia Iyz	-0.00061103
Inertia Izz	0.02925799

ตารางที่ ก.1: ตารางแสดงค่าพารามิเตอร์ทั้งตัว

### Right Hip Yaw



รูปที่ ก.2: ภาพแสดงก้านต่อ Right Hip Yaw

Link	r_hip_yaw
Mass (kg)	0.09100000
CoM X (m)	0.00000000
CoM Y (m)	0.02864983
CoM Z (m)	-0.02500000
Inertia Ixx	0.00014158
Inertia Ixy	0.00000000
Inertia Ixz	0.00000000
Inertia Iyy	0.00014316
Inertia Iyz	0.00000000
Inertia Izz	0.00002022

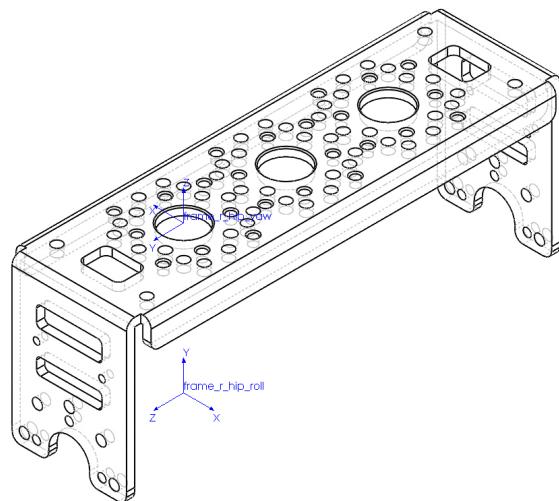
(ก) DH Parameter

Link	r_hip_yaw
Mass (kg)	0.09100000
CoM X (m)	0.00000000
CoM Y (m)	-0.02500000
CoM Z (m)	-0.00735017
Inertia Ixx	0.00014158
Inertia Ixy	0.00000000
Inertia Ixz	0.00000000
Inertia Iyy	0.00002022
Inertia Iyz	0.00000000
Inertia Izz	0.00014316

(ข) URDF

ตารางที่ ก.2: ตารางแสดงค่าพารามิเตอร์ Right Hip Yaw

### Left Hip Yaw



รูปที่ ก.3: ภาพแสดงก้านต่อ Left Hip Yaw

Link	l_hip_yaw
Mass (kg)	0.09100000
CoM X (m)	0.00000000
CoM Y (m)	0.02864983
CoM Z (m)	-0.02500000
Inertia Ixx	0.00014158
Inertia Ixy	0.00000000
Inertia Ixz	0.00000000
Inertia Iyy	0.00014316
Inertia Iyz	0.00000000
Inertia Izz	0.00002022

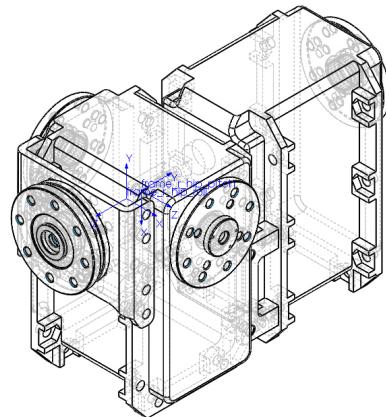
(ก) DH Parameter

Link	l_hip_yaw
Mass (kg)	0.09100000
CoM X (m)	0.00000000
CoM Y (m)	0.02500000
CoM Z (m)	-0.00735017
Inertia Ixx	0.00014158
Inertia Ixy	0.00000000
Inertia Ixz	0.00000000
Inertia Iyy	0.00002022
Inertia Iyz	0.00000000
Inertia Izz	0.00014316

(ข) URDF

ตารางที่ ก.3: ตารางแสดงค่าพารามิเตอร์ Left Hip Yaw

### Right Hip Roll



รูปที่ ก.4: ภาพแสดงก้านต่อ Right Hip Roll

Link	r_hip_roll
Mass (kg)	0.34300000
CoM X (m)	0.01526237
CoM Y (m)	0.02152630
CoM Z (m)	0.00000000
Inertia Ixx	0.00026846
Inertia Ixy	0.00000219
Inertia Ixz	-0.00000081
Inertia Iyy	0.00014760
Inertia Iyz	0.00000000
Inertia Izz	0.00032448

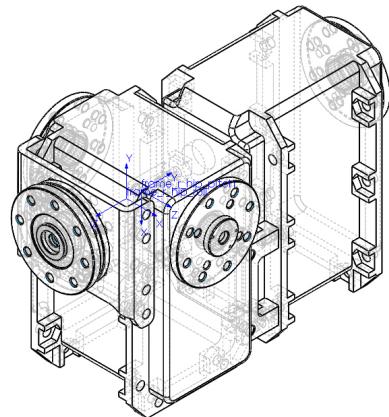
(ก) DH Parameter

Link	r_hip_roll
Mass (kg)	0.34300000
CoM X (m)	0.00000000
CoM Y (m)	-0.01526237
CoM Z (m)	-0.02652630
Inertia Ixx	0.00032448
Inertia Ixy	0.00000081
Inertia Ixz	0.00000000
Inertia Iyy	0.00026846
Inertia Iyz	0.00000219
Inertia Izz	0.00014760

(ข) URDF

ตารางที่ ก.4: ตารางแสดงค่าพารามิเตอร์ Right Hip Roll

### Left Hip Roll



รูปที่ ก.5: ภาพแสดงก้านต่อ Left Hip Roll

Link	l_hip_roll
Mass (kg)	0.34300000
CoM X (m)	0.01526237
CoM Y (m)	0.02152630
CoM Z (m)	0.00000000
Inertia Ixx	0.00026846
Inertia Ixy	0.00000219
Inertia Ixz	-0.00000081
Inertia Iyy	0.00014760
Inertia Iyz	0.00000000
Inertia Izz	0.00032448

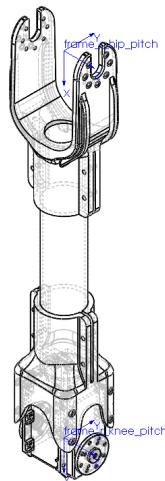
(ก) DH Parameter

Link	l_hip_roll
Mass (kg)	0.34300000
CoM X (m)	0.00000000
CoM Y (m)	-0.01526237
CoM Z (m)	-0.02652630
Inertia Ixx	0.00032448
Inertia Ixy	0.00000081
Inertia Ixz	0.00000000
Inertia Iyy	0.00026846
Inertia Iyz	0.00000219
Inertia Izz	0.00014760

(ข) URDF

ตารางที่ ก.5: ตารางแสดงค่าพารามิเตอร์ Left Hip Roll

### Right Hip Pitch



รูปที่ ก.6: ภาพแสดงก้านต่อ Right Hip Pitch

Link	r_hip_pitch
Mass (kg)	0.31800000
CoM X (m)	-0.07862011
CoM Y (m)	0.00000000
CoM Z (m)	0.00000000
Inertia Ixx	0.00011525
Inertia Ixy	0.00000000
Inertia Ixz	0.00000078
Inertia Iyy	0.00254669
Inertia Iyz	0.00000000
Inertia Izz	0.00250848

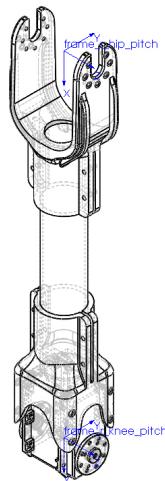
(ก) DH Parameter

Link	r_hip_pitch
Mass (kg)	0.31800000
CoM X (m)	0.22137989
CoM Y (m)	0.00000000
CoM Z (m)	0.00000000
Inertia Ixx	0.00011525
Inertia Ixy	0.00000000
Inertia Ixz	0.00000078
Inertia Iyy	0.00254669
Inertia Iyz	0.00000000
Inertia Izz	0.00250848

(ข) URDF

ตารางที่ ก.6: ตารางแสดงค่าพารามิเตอร์ Right Hip Pitch

### Left Hip Pitch



รูปที่ ก.7: ภาพแสดงก้านต่อ Left Hip Pitch

Link	l_hip_pitch
Mass (kg)	0.31800000
CoM X (m)	-0.07862011
CoM Y (m)	0.00000000
CoM Z (m)	0.00000000
Inertia Ixx	0.00011525
Inertia Ixy	0.00000000
Inertia Ixz	0.00000078
Inertia Iyy	0.00254669
Inertia Iyz	0.00000000
Inertia Izz	0.00250848

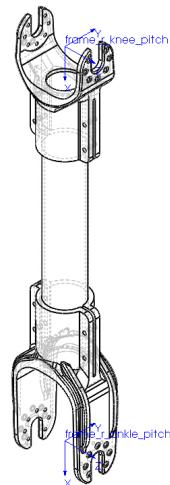
(ก) DH Parameter

Link	l_hip_pitch
Mass (kg)	0.31800000
CoM X (m)	0.22137989
CoM Y (m)	0.00000000
CoM Z (m)	0.00000000
Inertia Ixx	0.00011525
Inertia Ixy	0.00000000
Inertia Ixz	0.00000078
Inertia Iyy	0.00254669
Inertia Iyz	0.00000000
Inertia Izz	0.00250848

(ข) URDF

ตารางที่ ก.7: ตารางแสดงค่าพารามิเตอร์ Left Hip Pitch

### Right Knee Pitch



รูปที่ ก.8: ภาพแสดงก้านต่อ Right Knee Pitch

Link	r_knee_pitch
Mass (kg)	0.13800000
CoM X (m)	-0.15211782
CoM Y (m)	0.00000000
CoM Z (m)	0.00000000
Inertia Ixx	0.00011525
Inertia Ixy	0.00000000
Inertia Ixz	0.00000000
Inertia Iyy	0.00127592
Inertia Iyz	0.00000000
Inertia Izz	0.00124960

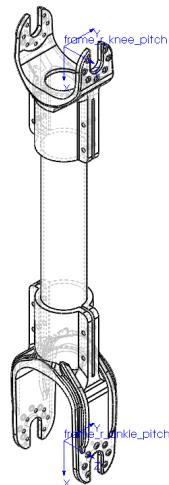
(ก) DH Parameter

Link	r_knee_pitch
Mass (kg)	0.13800000
CoM X (m)	0.16288218
CoM Y (m)	0.00000000
CoM Z (m)	0.00000000
Inertia Ixx	0.00005794
Inertia Ixy	0.00000000
Inertia Ixz	0.00000000
Inertia Iyy	0.00127592
Inertia Iyz	0.00000000
Inertia Izz	0.00124960

(ข) URDF

ตารางที่ ก.8: ตารางแสดงค่าพารามิเตอร์ Right Knee Pitch

### Left Knee Pitch



รูปที่ ก.9: ภาพแสดงก้านต่อ Left Knee Pitch

Link	l_knee_pitch
Mass (kg)	0.13800000
CoM X (m)	-0.15211782
CoM Y (m)	0.00000000
CoM Z (m)	0.00000000
Inertia Ixx	0.00011525
Inertia Ixy	0.00000000
Inertia Ixz	0.00000000
Inertia Iyy	0.00127592
Inertia Iyz	0.00000000
Inertia Izz	0.00124960

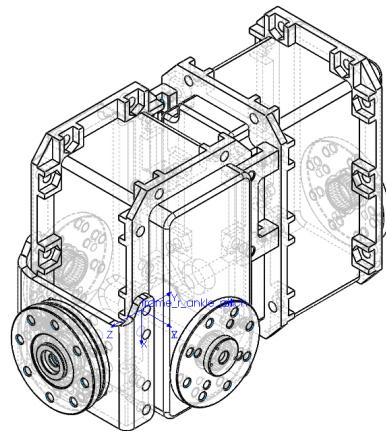
(ก) DH Parameter

Link	l_knee_pitch
Mass (kg)	0.13800000
CoM X (m)	0.16288218
CoM Y (m)	0.00000000
CoM Z (m)	0.00000000
Inertia Ixx	0.00005794
Inertia Ixy	0.00000000
Inertia Ixz	0.00000000
Inertia Iyy	0.00127592
Inertia Iyz	0.00000000
Inertia Izz	0.00124960

(ข) URDF

ตารางที่ ก.9: ตารางแสดงค่าพารามิเตอร์ Left Knee Pitch

### Right Ankle Pitch



รูปที่ ก.10: ภาพแสดงก้านต่อ Right Ankle Pitch

Link	r_ankle_pitch
Mass (kg)	0.33138738
CoM X (m)	-0.01526237
CoM Y (m)	0.00000000
CoM Z (m)	-0.02152630
Inertia Ixx	0.00025937
Inertia Ixy	0.00000000
Inertia Ixz	0.00000079
Inertia Iyy	0.00031349
Inertia Iyz	0.00000000
Inertia Izz	0.00014261

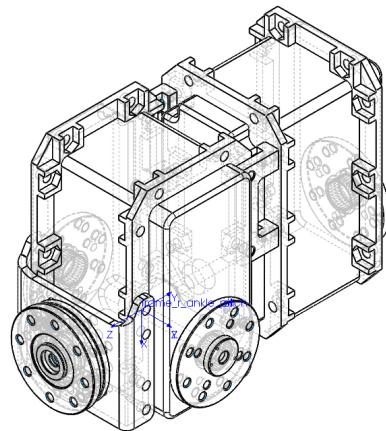
(ก) DH Parameter

Link	r_ankle_pitch
Mass (kg)	0.33138738
CoM X (m)	-0.01526237
CoM Y (m)	0.02152630
CoM Z (m)	0.00000000
Inertia Ixx	0.00025937
Inertia Ixy	0-0.00000212
Inertia Ixz	0.00000079
Inertia Iyy	0.00014261
Inertia Iyz	0.00000000
Inertia Izz	0.00031349

(ข) URDF

ตารางที่ ก.10: ตารางแสดงค่าพารามิเตอร์ Right Ankle Pitch

### Left Ankle Pitch



รูปที่ ก.11: ภาพแสดงก้านต่อ Left Ankle Pitch

Link	l_ankle_pitch
Mass (kg)	0.33138738
CoM X (m)	-0.01526237
CoM Y (m)	0.00000000
CoM Z (m)	-0.02152630
Inertia Ixx	0.00025937
Inertia Ixy	0.00000000
Inertia Ixz	0.00000079
Inertia Iyy	0.00031349
Inertia Iyz	0.00000000
Inertia Izz	0.00014261

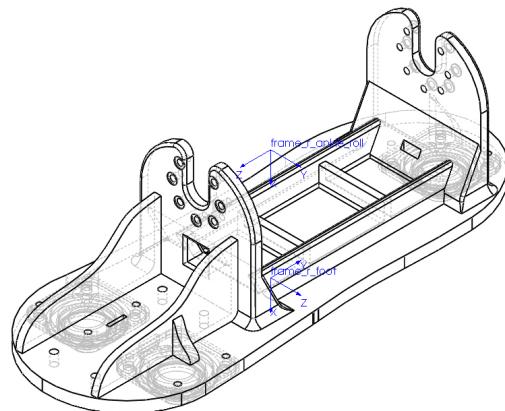
(ก) DH Parameter

Link	l_ankle_pitch
Mass (kg)	0.33138738
CoM X (m)	-0.01526237
CoM Y (m)	0.02152630
CoM Z (m)	0.00000000
Inertia Ixx	0.00025937
Inertia Ixy	0-0.00000212
Inertia Ixz	0.00000079
Inertia Iyy	0.00014261
Inertia Iyz	0.00000000
Inertia Izz	0.00031349

(ข) URDF

ตารางที่ ก.11: ตารางแสดงค่าพารามิเตอร์ Left Ankle Pitch

### Right Ankle Roll



รูปที่ ก.12: ภาพแสดงก้านต่อ Right Ankle Roll

Link	r_ankle_roll
Mass (kg)	0.10500000
CoM X (m)	-0.01454118
CoM Y (m)	-0.00034576
CoM Z (m)	-0.00019548
Inertia Ixx	0.00034591
Inertia Ixy	-0.00000857
Inertia Ixz	-0.00000013
Inertia Iyy	0.00004813
Inertia Iyz	-0.00000120
Inertia Izz	0.00032705

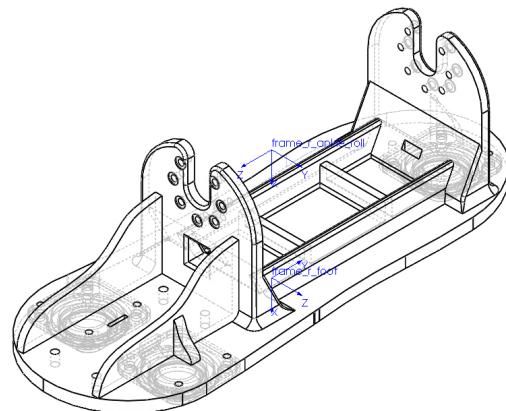
(ก) DH Parameter

Link	r_ankle_roll
Mass (kg)	0.10500000
CoM X (m)	0.03625882
CoM Y (m)	-0.00019548
CoM Z (m)	0.00034576
Inertia Ixx	0.00034591
Inertia Ixy	-0.00000013
Inertia Ixz	0.00000857
Inertia Iyy	0.00032705
Inertia Iyz	0.00000120
Inertia Izz	0.00004813

(ข) URDF

ตารางที่ ก.12: ตารางแสดงค่าพารามิเตอร์ Right Ankle Roll

### Left Ankle Roll



รูปที่ ก.13: ภาพแสดงก้านต่อ Left Ankle Roll

Link	l_ankle_roll
Mass (kg)	0.10500000
CoM X (m)	-0.01454118
CoM Y (m)	-0.00034576
CoM Z (m)	-0.00019548
Inertia Ixx	0.00034591
Inertia Ixy	-0.00000857
Inertia Ixz	-0.00000013
Inertia Iyy	0.00004813
Inertia Iyz	-0.00000120
Inertia Izz	0.00032705

(ก) DH Parameter

Link	l_ankle_roll
Mass (kg)	0.10500000
CoM X (m)	0.03625882
CoM Y (m)	-0.00019548
CoM Z (m)	0.00034576
Inertia Ixx	0.00034591
Inertia Ixy	-0.00000013
Inertia Ixz	0.00000857
Inertia Iyy	0.00032705
Inertia Iyz	0.00000120
Inertia Izz	0.00004813

(ข) URDF

ตารางที่ ก.13: ตารางแสดงค่าพารามิเตอร์ Left Ankle Roll

## ประวัติผู้เขียน

นายจิรภพ ศรีรัตนอาภรณ์



ชื่อ สกุล	นายจิรภพ ศรีรัตนอาภรณ์
รหัสนักศึกษา	57340500067
วุฒิการศึกษา	วิศวกรรมศาสตรบัณฑิต
	วิศวกรรมหุ่นยนต์และระบบอัตโนมัติ
ชื่อสถาบัน	มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
ปีที่สำเร็จการศึกษา	2560

## ประวัติผู้เขียน

นายเจษฎากร ท่าไชยวงศ์



ชื่อ สกุล	นายเจษฎากร ท่าไชยวงศ์
รหัสนักศึกษา	57340500067
วุฒิการศึกษา	วิศวกรรมศาสตรบัณฑิต
	วิศวกรรมหุ่นยนต์และระบบอัตโนมัติ
ชื่อสถาบัน	มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
ปีที่สำเร็จการศึกษา	2560

## ประวัติผู้เขียน

นายวุฒิภัทร โชคอนันตทรัพย์



ชื่อ สกุล	นายวุฒิภัทร โชค_anantraphi
รหัสนักศึกษา	57340500067
วุฒิการศึกษา	วิศวกรรมศาสตรบัณฑิต
ชื่อสถาบัน	วิศวกรรมที่นี่ยนต์และระบบอัตโนมัติ
ปีที่สำเร็จการศึกษา	มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
	2560