



รายงานสรุปผลการดำเนินงานรายวิชา FRA 333 Kinematics of Robotics System

สถาบันวิทยาการหุ่นยนต์ภาคสนาม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

หัวข้อ

7 DOF Manipulator simulates the joint pattern of a Human arm

Kinematics simulation

กลุ่มที่ 24

ผู้ดำเนินงาน

- 1.นายไกรวิชญ์ วิชาโคตร รหัสนักศึกษา 65340500004
- 2.นายชยณัฐ เลิศวิทยานุรักษ์ รหัสนักศึกษา 65340500012
- 3.นายชวภณ วชิรานิรมิต รหัสนักศึกษา 65340500014
- 4.นายวรวิทธิ์ธร คงหนู รหัสนักศึกษา 65340500050

สารบัญ

สารบัญ	ก
สารบัญรูปภาพ.....	ค
สารบัญตาราง	ง
บทที่ 1 บทนำ.....	1
ที่มาและความสำคัญ	1
วัตถุประสงค์.....	1
ขอบเขต.....	1
ระยะเวลาและแผนการปฏิบัติงาน	2
บทที่ 2 งานวิจัยและทฤษฎีที่เกี่ยวข้อง	3
Inverse Kinematics and Control of a 7-DOF Redundant Manipulator Based on the Closed-Loop Algorithm	3
Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse and Damped Least Squares methods.....	5
The Singular Value Decomposition : Computation and Applications to Robotics.....	7
บทที่ 3 การดำเนินงาน.....	9
System Overview	9
ROS System Architecture	11
Model แบบจำลองของแขนมนุษย์	12
การคำนวณทาง Kinematics	13

สารบัญ (ต่อ)

บทที่ 4 ผลการดำเนินงาน	15
ระบบ Input ค่าสำหรับควบคุมการทำงาน	15
ระบบควบคุมการทำงาน MoveJ	15
ระบบควบคุมการทำงาน MoveL	16
บทที่ 5 สรุปผลการดำเนินงาน	17
สรุปผลการดำเนินงาน	17
แผนการการพัฒนาต่อ	17
อ้างอิง	18

สารบัญรูปภาพ

รูปที่ 1 DH – Parameter และโครงสร้างของแขนกล 7 DOF ที่ใช้ในบทความ	3
รูปที่ 2 Block Diagram ระบบการควบคุมความเร็วของข้อต่อแขนกล	4
รูปที่ 3 System Overview ของระบบการทำงาน	9
รูปที่ 4 System Architecture การทำงานบน ROS.....	11
รูปที่ 5 การออกแบบ Custom Service และการเรียกใช้งาน Service แต่ละตัว	12
รูปที่ 6 ชิ้นส่วนสำหรับประกอบเป็น Model แบบจำลองแขนมนุษย์.....	12
รูปที่ 7 Model แบบจำลองแขนมนุษย์บน RVIZ.....	13
รูปที่ 8 หน้าต่างระบบ Input ค่าสำหรับการควบคุม	15
รูปที่ 9 ตัวอย่างการทำงานในโหมด MoveJ.....	15
รูปที่ 10 ตัวอย่างการทำงานในโหมด MoveL	16

สารบัญตาราง

ตารางที่ 1 แผนการดำเนินงาน.....	2
ตารางที่ 2 MDH – Parameter ของ Model ที่จัดเรียงข้อต่อแบบ 3R - R - 3R	13

บทที่ 1 บทนำ

ที่มาและความสำคัญ

จากความรู้ในเรื่องของ Kinematics ทำให้สามารถนำมาใช้ในการอธิบายลักษณะการเคลื่อนที่ของแขนกลได้และสามารถใช้ในการควบคุมการทำงานของแขนกลให้เป็นไปตามค่าที่กำหนดได้ ดังนั้นความรู้ในเรื่องของ Kinematics จะนำมาใช้อธิบายการเคลื่อนที่ของแขนของสิ่งมีชีวิตได้หรือไม่ โดยในโครงการนี้นั้นได้ทำการนำแขนของมนุษย์ มาใช้เป็นต้นแบบในการสร้างแขนกลจำลองที่มีลักษณะการเรียงตัวของข้อต่อแบบ 3R – R – 3R ที่สามารถทำการอธิบายการเคลื่อนที่และทดลองควบคุมการเคลื่อนที่ผ่านการทำ Simulation ขึ้นมา

วัตถุประสงค์

1. เพื่อสร้าง Simulation การเคลื่อนที่ของ Model แขนมนุษย์
2. เพื่อสร้างระบบจำลองการควบคุมการเคลื่อนที่ของแขนมนุษย์
3. เพื่อศึกษาแรงในข้อต่อของแขนมนุษย์ที่ต้องใช้เมื่อมีแรงมากระทำที่ปลายของแขนเพื่อรักษาท่าทาง
4. เพื่อศึกษาการประยุกต์ใช้ความรู้ทาง Kinematics ในการอธิบายการเคลื่อนที่และการควบคุมการเคลื่อนที่ของแขนมนุษย์

ขอบเขต

1. Simulation จำลองเคลื่อนที่ของแขนมนุษย์ตั้งแต่หัวไหล่จนถึงข้อมือ
2. ระบบควบคุมการเคลื่อนที่สามารถควบคุมผ่าน Configuration space โดยการปรับค่า q ของแต่ละ joint
3. ระบบควบคุมการเคลื่อนที่สามารถควบคุมผ่าน Task space โดยกำหนด ตำแหน่งที่มีการอ้างอิงกับ reference frame
4. ระบบควบคุมการเคลื่อนที่สามารถควบคุมผ่าน Task space ที่สามารถเคลื่อนที่แบบ MoveJ หรือ MoveL
5. ระบบคำนวณหาแรงในแต่ละ joint กรณีมีแรงภายนอกมากระทำที่จุดปลายของแขนใน Simulation เมื่อแขนอยู่กับที่

ระยะเวลาและแผนการปฏิบัติงาน

ลำดับ	การดำเนินงาน	สัปดาห์ที่					
		1	2	3	4	5	6
1	ศึกษาข้อมูลเกี่ยวกับ Kinematics ของแขนมนุษย์						
2	จัดทำ Proposal						
3	พัฒนาระบบ Simulation ในส่วนของ Model แขนมนุษย์						
4	พัฒนาระบบการคำนวณ ทาง Kinematics						
5	รวมระบบ Simulation						
6	ปรับปรุงแก้ไขงาน						
7	ทดสอบและบันทึกผลการทำงาน						
8	จัดทำ readme และเอกสารสำหรับการนำเสนอ						
9	นำเสนอและจัดส่งงาน						

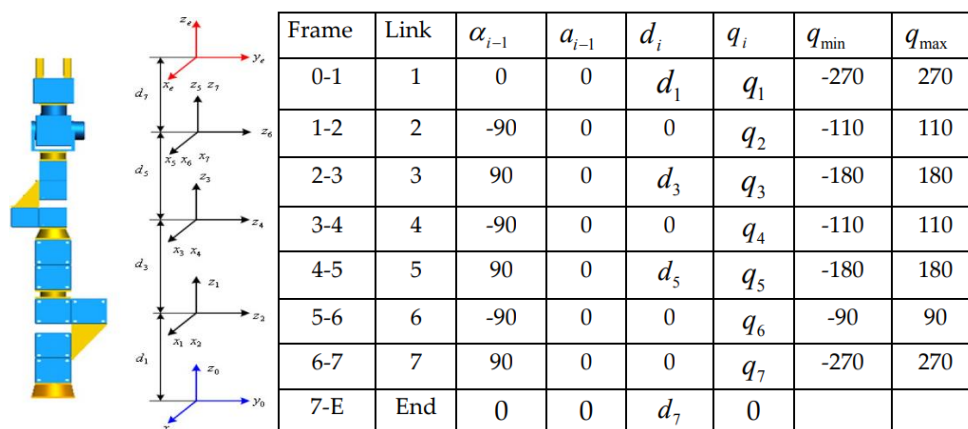
ตารางที่ 1 แผนการดำเนินงาน

บทที่ 2 งานวิจัยและทฤษฎีที่เกี่ยวข้อง

Inverse Kinematics and Control of a 7-DOF Redundant Manipulator Based on the Closed-Loop Algorithm[1]

บทความนี้อธิบายถึงการนำ Closed - loop inverse kinematics (CLIK) algorithm ในการควบคุมการทำงานของแขนกล 7 DOF โดยในปกตินั้นจะใช้แค่การควบคุมความเร็วเพียงเท่านั้นแต่บทความนี้มีการนำการควบคุมความเร่งเข้ามาอธิบายเพิ่มเติมด้วยโดยใช้ Pseudoinverse method และมีการเพิ่ม Joint limits avoidance (JLA) เพื่อจัดการกับปัญหา null space motion และมีการแสดงผลการทำงานทั้ง Simulations และการควบคุมแขนกล 7 DOF จริง

Kinematics model ของแขนกล 7 DOF นี้สามารถเขียนเป็น DH - Parameter ได้ดังนี้



รูปที่ 1 DH - Parameter และโครงสร้างของแขนกล 7 DOF ที่ใช้ในบทความ

ในการควบคุมการทำงานนั้นเป็นการใช้ Closed - loop inverse kinematics โดยการใช้ Differential kinematics equation ในการอธิบายการควบคุมความเร็วที่เกิดขึ้นโดยทำการคำนวณผ่าน Jacobian ในการแปลงความเร็วของแต่ละข้อต่อให้เป็นความเร็วที่ปลายแขนของแขนกล โดยกำหนดให้

Task space คือ $(x(t), \dot{x}(t))$

Joint space คือ $(q(t), \dot{q}(t))$

จะได้ความสัมพันธ์ของ Task space จาก Jacobian Matrix คือ

$$\dot{x} = J(q)\dot{q}$$

เนื่องจากแขนกลเป็นแบบ 7 DOF ทำให้ Jacobian matrix มีขนาด 6×7 ซึ่งไม่สามารถหา Invert matrix ได้ทำให้ต้องมีการแปลง Jacobian matrix ก่อนด้วย Pseudoinverse Jacobian จึงจะสามารถหาความสัมพันธ์ของ Joint space จาก Jacobian Matrix ได้โดย

$$J^\dagger = J^T(JJ^T)^{-1}$$

จากการแปลง Jacobian matrix จะได้ความสัมพันธ์ของ Joint space จาก Jacobian Matrix คือ

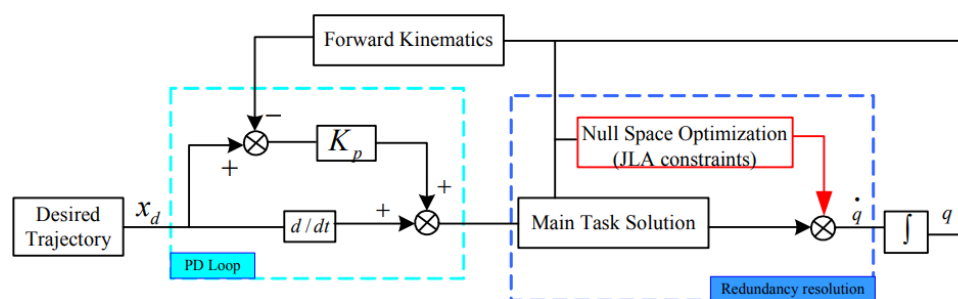
$$\dot{q} = J^\dagger(q)\dot{x}$$

สำหรับแขนกลที่มีจำนวนข้อต่อมากกว่าจำนวน Task space ที่ควบคุมได้ ($n > m$) ทำให้อาจเกิดปัญหา nullspace motion ขึ้นได้สามารถแก้ปัญหาได้โดยใช้สมการ

$$\dot{q} = J^\dagger(q)\dot{x} + (I - J^\dagger(q)J(q))\dot{q}_0$$

โดยพจน์แรกใช้ในการหาคำตอบทั่วไป ส่วนพจน์ที่สองอธิบายค่า Homogeneous solution

ในบทความนี้มีการควบคุมการทำงานของแขนกลอยู่ 2 แบบ คือการควบคุมความเร็วและควบคุมความเร่งของแต่ละข้อต่อ แต่ในโครงการนี้เป็นการควบคุมความเร็วเพียงอย่างเดียว จึงมีการยกมาแค่ในส่วนของการควบคุมความเร็วของแต่ละข้อต่อเพียงอย่างเดียว



รูปที่ 2 Block Diagram ระบบการควบคุมความเร็วของข้อต่อแขนกล

ในระบบจะทำการหา Error จาก

$$e = x_d - x$$

$$\dot{e} = \dot{x}_d - \dot{x}$$

เมื่อแทนค่าลงในสมการความสัมพันธ์ของ Joint space จาก Jacobian Matrix จะได้

$$\dot{q} = J^+(q)(\dot{x}_d + K_p(x_d - x))$$

ซึ่งจากสมการจะมีการใช้ P control ในการควบคุมการทำงาน ซึ่งสามารถนำมารวมกับพจน์ที่อธิบายค่า Homogeneous solution เพื่อป้องกัน null space motion ได้โดยจะได้

$$\dot{q} = J^+(q)(\dot{x}_d + K_p(x_d - x)) + (I - J^+(q)J(q))\dot{q}_0$$

Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse and Damped Least Squares methods[2]

บทความนี้อธิบายถึงการใช้งาน Inverse Kinematics ร่วมกับ Jacobian Transpose, Pseudoinverse และ Damped Least Squares methods โดยในโครงการนี้มีการใช้งานแค่ Pseudoinverse method จึงมีการยกมาแค่ในส่วน Pseudoinverse method เพียงอย่างเดียว

Pseudoinverse method หรือ Moore-Penrose Inverse ของ Jacobian matrix นั้นเป็นกระบวนการที่ใช้ในการหา invert matrix ของ Jacobian matrix ที่ไม่ใช่ matrix จตุรัสหรือไม่เป็น full row rank โดยสามารถเขียนอยู่ในรูปสมการได้

$$\Delta\theta = J^+ \vec{e}$$

โดย Pseudoinverse method จะให้ตอบที่ดีที่สุดของ

$$J\Delta\theta = \vec{e}$$

ซึ่งคำตอบที่ได้จะมาจากการใช้ Least squares method ในการหาคำตอบซึ่งจะนิยามโดยสมการ

$$\|J\Delta\theta - \vec{e}\|^2$$

คำตอบที่ได้จะเกิดขึ้นได้ ประกอบด้วย

1. \vec{e} อยู่ในช่วงคอลัมน์ของ J คำตอบที่ได้ในสมการ $J\Delta\theta = \vec{e}$ จะมีค่าแน่นอน และ Pseudoinverse จะให้คำตอบ $\Delta\theta$ ที่มีขนาดเล็กที่สุด
2. \vec{e} ไม่อยู่ในช่วงคอลัมน์ของ J Pseudoinverse จะให้คำตอบ $\Delta\theta$ ที่มีความต่าง $\|J\Delta\theta - \vec{e}\|$ ที่น้อยที่สุด

Pseudoinverse method นั้นได้รับผลกระทบจากปัญหา Singularities โดยแบ่งเป็นเมื่อเป้าหมายอยู่ที่จุด Singularities การใช้ Pseudoinverse จะไม่เกิดการเคลื่อนที่ในทิศทางที่เป็นไปไม่ได้ แต่ถ้าอยู่ใกล้ Singularities การใช้ Pseudoinverse จะทำให้ $\Delta\theta$ มีการเปลี่ยนแปลงค่าอย่างมากแม้ว่า \vec{e} จะเปลี่ยนแปลงเพียงเล็กน้อย

การกำหนดเป้าหมายรองโดยใช้ Nullspace method ช่วยให้สามารถเพิ่มเป้าหมายเพิ่มเติม โดยไม่กระทบต่อเป้าหมายหลัก โดยเกิดจากการ Projection matrix ลงบน Nullspace ของ J ซึ่งจะนิยามโดยสมการ

$$\Delta\theta = J^+ \vec{e} + (I - J^+ J) \varphi$$

กำหนดให้ φ เป็นเวกเตอร์สำหรับเป้าหมายรองสามารถหา Pseudoinverse method ของ Jacobian matrix ได้โดยเริ่มจากสมการในรูปปกติ

$$J^T J \Delta\theta = \vec{e}$$

กำหนดให้ $\vec{z} = J^T \vec{e}$ จะได้

$$(J^T J) \Delta\theta = \vec{z}$$

ซึ่งจากสมการสามารถบอกได้ว่าค่า \vec{z} จะอยู่ในช่วงของ $J^T J$ เสมอทำให้สามารถหาคำตอบได้เสมอ และเมื่อเป็น full row rank ทำให้ $J^T J$ แสดงให้เห็นว่าสามารถหาค่า Invert ได้และคำตอบที่จะ Minimum magnitude ของ $\Delta\theta$ ซึ่งสามารถเขียนอยู่ในรูปของ

$$\Delta\theta = J^T (J J^T)^{-1} \vec{e}$$

การทำ Pseudoinverse method มีการใช้งานอย่างแพร่หลายแต่ก็ต้องระวังความไม่เสถียรเมื่อเข้าใกล้ Singularities

The Singular Value Decomposition : Computation and Applications to Robotics[3]

Singular Value Decomposition (SVD) เป็นเครื่องมือสำคัญในพีชคณิตเชิงเส้นที่ใช้ในการวิเคราะห์สมบัติทางจลนศาสตร์และพลศาสตร์ของระบบหุ่นยนต์ โดยเฉพาะในงานศึกษาการวิเคราะห์ Jacobian matrix ซึ่งเป็นตัวแปรสำคัญที่เชื่อมโยงความเร็วของข้อต่อไปยังความเร็วของ End-effector การประยุกต์ SVD ยังครอบคลุมถึงการจัดการ Redundancy, Dexterity Optimization และ Singularity

จุดอ่อนของระบบนี้เกิดขึ้นเมื่อ Jacobian matrix มีค่าลดลำดับ (Rank Deficiency) ที่ความเอกราช ซึ่งส่งผลกระทบต่อควบคุมและลดความสามารถในการควบคุมของหุ่นยนต์ แม้ว่า SVD จะให้วิธีการจัดการปัญหานี้ได้อย่างแม่นยำ แต่ความซับซ้อนในการคำนวณกลับจำกัดการใช้งานในเวลาจริง (Real-time)

การนำ SVD มาใช้งานในส่วนของ Robotics นั้นจะนิยมใช้เมื่อ Jacobian matrix ที่ได้มีขนาด $m \times n$ โดยจะสามารถจัดรูป SVD ให้อยู่ในรูปสมการ

$$J = U \Sigma V^T$$

กำหนดให้ $U \in \mathbb{R}^{m \times m}$ เป็น Orthogonal Matrices

$V \in \mathbb{R}^{n \times n}$ เป็น Orthogonal Matrices

$\Sigma \in \mathbb{R}^{m \times n}$ เป็น Matrix แนวทแยงที่มีค่า Singular (σ_i) บนเส้นทแยงมุมเรียงลำดับ

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{\min(m,n)} \geq 0$$

ค่า Singular (σ_i) แสดงถึงความสามารถในการปรับขนาด (Scaling) ของ J ในทิศทางที่กำหนดโดยเวกเตอร์ V

ค่าเงื่อนไข (Condition Number) สามารถอธิบายได้โดยสมการ

$$k(J) = \frac{\sigma_{\max}}{\sigma_{\min}}$$

บ่งบอกถึงความไวต่อการเปลี่ยนแปลงของอินพุต หาก $\sigma_{\min} \rightarrow 0$ ระบบจะเข้าสู่ Singularity

การนำ SVD มาใช้งานในส่วนของ Robotics จะนำมาใช้ได้หลากหลายแต่ในโครงการนี้นั้นมีการใช้งานแค่ส่วนของการตรวจสอบ Singularity เท่านั้นจึงจะยกมาแค่ประเด็นนี้

การเกิด Singularity นั้นส่งผลให้การคำนวณความเร็วของข้อต่อมีค่าที่ได้ไม่จำกัด โดย SVD จะมีการแปลง Jacobian Matrix ด้วย Damped Least-Squares (DLS) Solution เพื่อเพิ่มเสถียรภาพของระบบตามสมการ

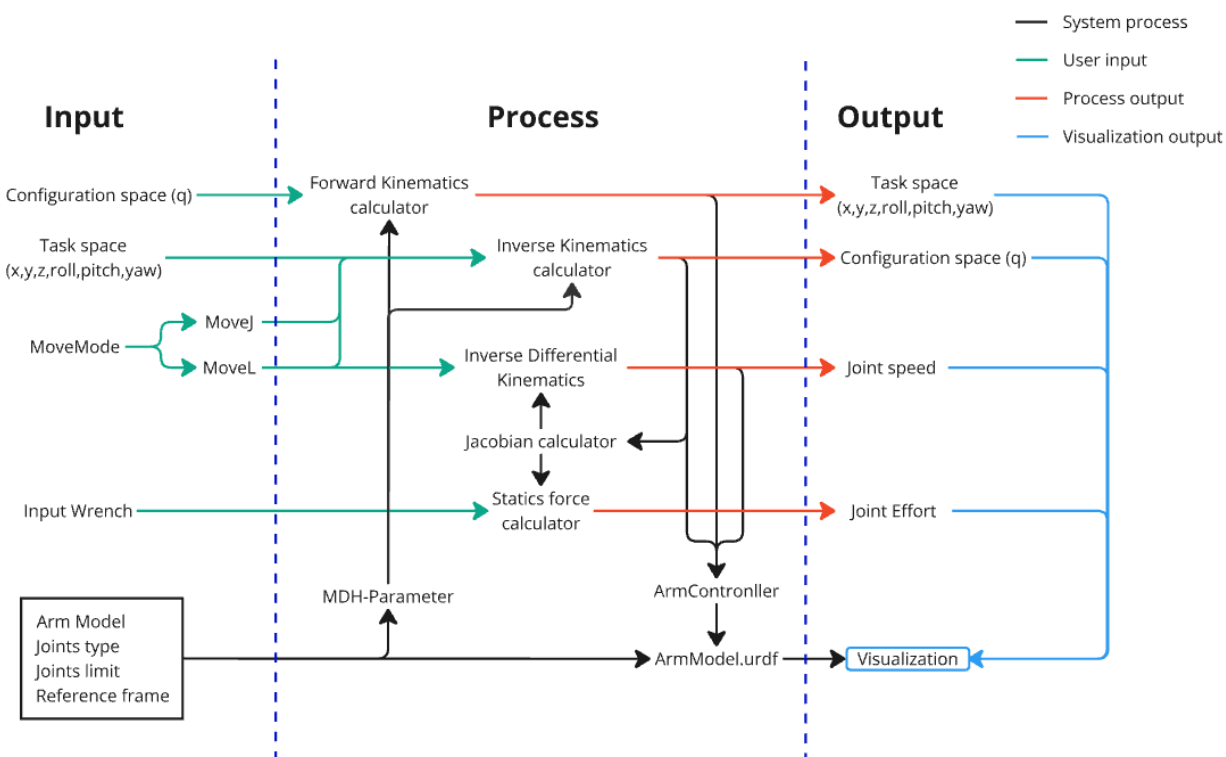
$$J + \lambda = V(\Sigma^2 + \lambda^2 I)^{-1} \Sigma U^T$$

กำหนดให้ J เป็น Jacobian Matrix

$\lambda > 0$ เป็น ตัวแปรลดผลกระทบจาก Singularity

บทที่ 3 การดำเนินงาน

System Overview



รูปที่ 3 System Overview ของระบบการทำงาน

การทำงานของระบบถูกแบ่งออกเป็น 3 ส่วนหลักโดยประกอบด้วย

1. ส่วน Input ของระบบซึ่งแบ่งออกเป็น 2 ส่วน คือ

1.1. ส่วนที่ผู้ใช้งานสามารถ Input ให้กับระบบได้ ประกอบด้วย

1.1.1. Configuration Space (q) : ค่ามุมในแต่ละข้อต่อที่ใช้เพื่อควบคุมการเคลื่อนไหวกของแต่ละข้อต่อ

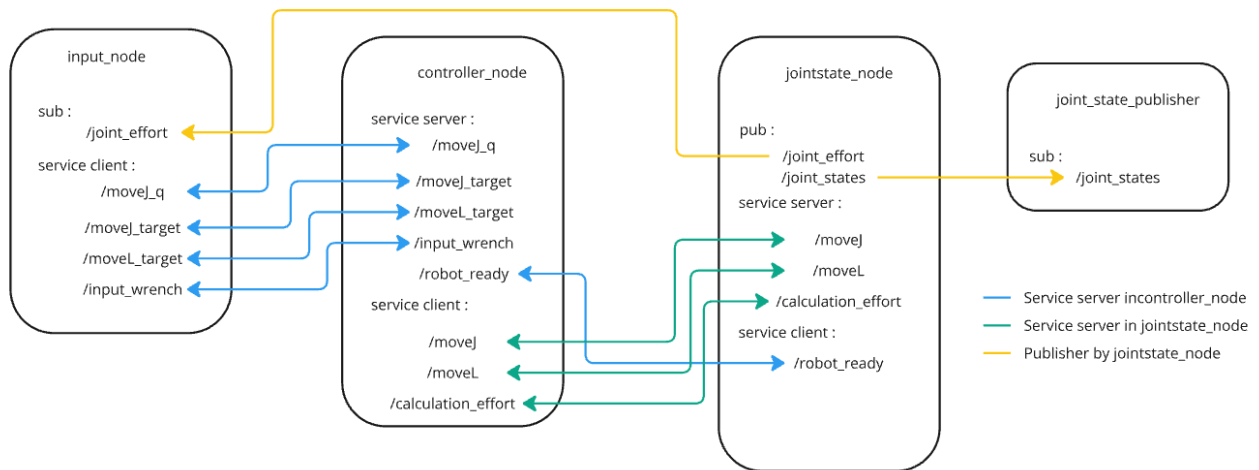
1.1.2. Task Space ($x, y, z, roll, Pitch, yaw$) : ตำแหน่งเป้าหมายใน Simulation ที่ให้ปลายแขนไปถึง

1.1.3. Move Mode (MoveJ, MoveL) : โหมดการเคลื่อนที่

1.1.4. Input Wrench : ค่า wrench (force and torque) จากภายนอกที่กระทำต่อปลายแขนเมื่อระบบหยุดนิ่งอยู่กับที่

- 1.2. ส่วนที่ระบบจะรับ Input จาก setup ที่ถูกกำหนดไว้ ประกอบด้วย
 - 1.2.1. Arm Model : Model แบบจำลองแต่ละส่วนของแขนมนุษย์
 - 1.2.2. Joint Type: ประเภทของข้อต่อโดยกำหนดไว้เป็นแบบ Revolute ที่ถูกจัดเรียงแบบ 3R – R – 3R
 - 1.2.3. Joint Limit : ขอบเขตการเคลื่อนไหวของแต่ละข้อต่อที่ถูกกำหนดไว้
 - 1.2.4. Reference Frame
2. ส่วน Process ของระบบซึ่งแบ่งออกเป็น 7 ส่วน คือ
 - 2.1. Model.urdf : Model แบบจำลองของแขนมนุษย์
 - 2.2. MDH – Parameter : Parameter ที่อธิบายการเปลี่ยนแปลงของแต่ละข้อต่อของ Model แบบจำลองแขนมนุษย์สำหรับใช้ในการคำนวณส่วนอื่น ๆ ต่อไป
 - 2.3. Forward Kinematics : การคำนวณที่ใช้ในการหา Task Space (x, y, z, roll, Pitch, yaw) จาก Configuration Space (q)
 - 2.4. Inverse Kinematics : การคำนวณที่ใช้ในการหา Configuration Space (q) จาก Task Space (x, y, z, roll, Pitch, yaw)
 - 2.5. Jacobian Matrix : การคำนวณที่ใช้ในการหา Jacobian Matrix ที่เกิดจากความสัมพันธ์ของการเปลี่ยนแปลง Configuration Space ที่ส่งผลต่อการเปลี่ยนแปลงของ Task Space
 - 2.6. Inverse Differential Kinematics : การคำนวณที่ใช้ในการหาความเร็วของแต่ละข้อต่อโดยการใช้ Inverse Jacobian Matrix ร่วมกับความเร็วของปลายแขน
 - 2.7. Statics Force : การคำนวณเพื่อหาค่า Joints Effort โดยการใช้ Jacobian Transpose Matrix ร่วมกับค่า wrench (force and torque)
3. ส่วน Output ของระบบซึ่งแบ่งออกเป็น 2 ส่วน คือ
 - 3.1. ส่วนของค่าที่ได้จากการคำนวณ
 - 3.1.1. Task Space (x, y, z, roll, Pitch, yaw)
 - 3.1.2. Configuration Space (q)
 - 3.1.3. Joint Speed : ความเร็วในการเคลื่อนที่ของแต่ละข้อต่อ
 - 3.1.4. Joint Effort : แรงบิดของแต่ละข้อต่อที่ใช้เพื่อรักษาท่าทางเมื่อมีแรงมากระทำที่ปลายแขน
 - 3.2. ส่วนการแสดงผลที่แสดงผลการเคลื่อนไหวของ Model แบบจำลองแขนมนุษย์จากค่าต่าง ๆ ที่คำนวณได้

ROS System Architecture

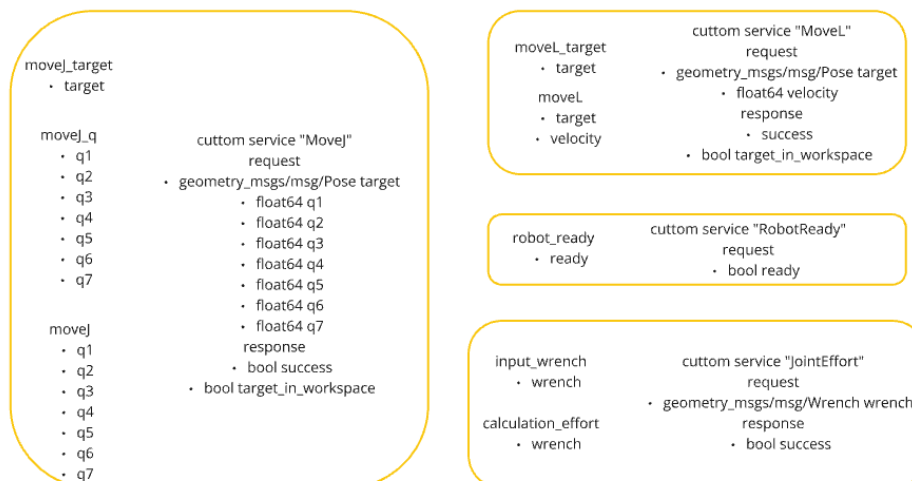


รูปที่ 4 System Architecture การทำงานบน ROS

ในการพัฒนาระบบการทำงานนั้นจะถูกพัฒนาขึ้นบน ROS โดยมีการแบ่งการทำงานออกเป็น Node ย่อยที่ทำหน้าที่แตกต่างกันจำนวน 4 node ประกอบด้วย

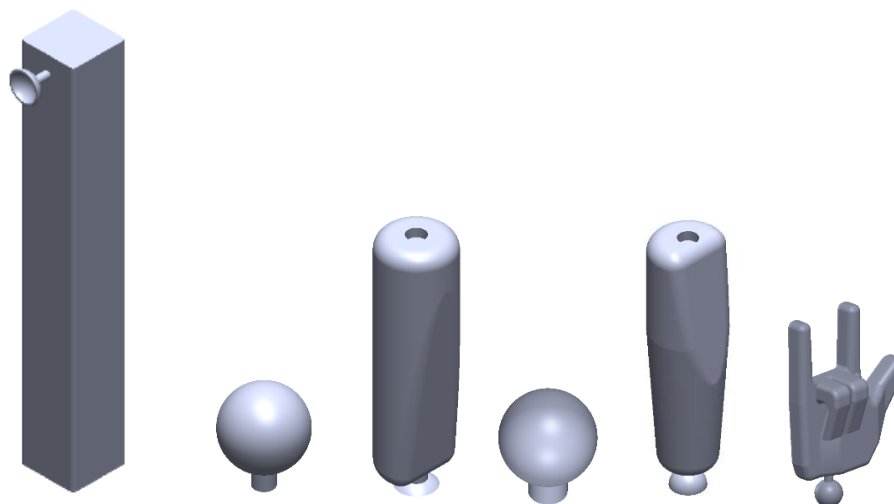
1. Input_node เป็น node สำหรับการแสดงผล UI สำหรับการ Input ค่าต่าง ๆ สำหรับการควบคุมการเคลื่อนที่ของ Model และแสดงผลค่า Joint Effort จากการคำนวณ
2. Controller_node เป็น node สำหรับสั่งงานการควบคุมการทำในรูปแบบต่าง ๆ โดยรับ Input จาก Input_node และส่งงานต่อไปยัง Jointstate_node
3. Jointstate_node เป็น node สำหรับสั่งงานค่าในการขยับของแต่ละข้อต่อตามคำสั่งที่ได้มาจาก Controller_node
4. Joint_state_publisher เป็น node สำหรับควบคุมการเคลื่อนที่ของ Model ที่แสดงผลบน RVIZ

มีการสร้าง Custom Service สำหรับการใช้งานที่เฉพาะเจาะจงจำนวน 4 service ประกอบด้วย

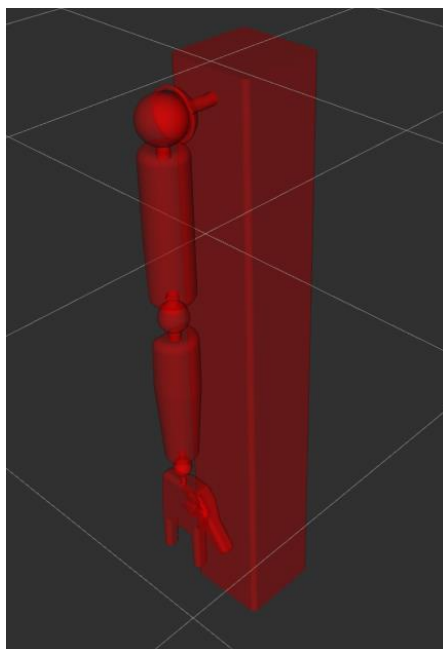


รูปที่ 5 การออกแบบ Custom Service และการเรียกใช้งาน Service แต่ละตัว

Model แบบจำลองของแขนมนุษย์



รูปที่ 6 ชิ้นส่วนสำหรับประกอบเป็น Model แบบจำลองแขนมนุษย์



รูปที่ 7 Model แบบจำลองแขนมนุษย์บน RVIZ

การคำนวณทาง Kinematics

ในการทำงานได้มีการใช้การคำนวณทาง Kinematics ในการหาค่าที่ใช้ในการควบคุมการเคลื่อนที่โดยมีการกำหนด MDH – Parameter ดังนี้

joint	a_{i-1}	α_{i-1}	d_i	θ_i
1	0.0	90.0°	0.0	q_1
2	0.0	90.0°	0.0	$q_2 + 90.0^\circ$
3	0.0	-90.0°	0.0	$q_3 - 90.0^\circ$
4	0.4	-90.0°	0.0	$q_4 - 90.0^\circ$
5	0.0	-90.0°	0.4	q_5
6	0.0	90.0°	0.0	$q_6 + 90.0^\circ$
7	0.0	90.0°	0.0	q_7

ตารางที่ 2 MDH – Parameter ของ Model ที่จัดเรียงข้อต่อแบบ 3R - R - 3R

จาก MDH – Parameter สามารถนำไปใช้ร่วมกับ Library python roboticstoolbox สำหรับใช้ในการคำนวณต่าง ๆ ทาง Kinematics โดยใช้ในการหา Forward Kinematics, Inverse Kinematics และ Jacobian Matrix สำหรับการคำนวณอื่น ๆ ที่ซับซ้อนต่อไป

ในส่วนการหาความเร็วในการเคลื่อนที่ของแต่ละข้อต่อนั้นหาจากการใช้ Pseudoinverse method ของ Jacobian matrix โดยหาจากสมการ

$$\dot{q} = J^+(q)\dot{x}$$

เมื่อกำหนดให้ \dot{q} แทน ค่าความเร็วในแต่ละข้อต่อ

\dot{x} แทน ค่าความเร็วที่ปลายแขน

$J^+(q)$ แทน Pseudoinverse method ของ Jacobian matrix ที่ต่อละข้อต่อทำมุม q

โดย $J^+(q)$ หรือ Pseudoinverse method ของ Jacobian matrix หาจาก

$$J^+ = J^T(JJ^T)^{-1}$$

เมื่อกำหนดให้ J แทน Jacobian matrix ที่ต่อละข้อต่อทำมุม q ใด ๆ

และ \dot{x} เป็นความเร็วคงที่ ที่กำหนดไว้ที่ 0.01 m/s โดยหาจาก Unit vector ของผลต่างระหว่างตำแหน่งปัจจุบัน กับตำแหน่งเป้าหมายคูณกับความเร็วคงที่เพื่อหาความเร็วในแต่ละแกน ได้ตามสมการ

$$\dot{x} = \left(\frac{target_{pose} - current_{pose}}{\|target_{pose} - current_{pose}\|} \right) 0.01$$

ในการตรวจสอบการเข้าใกล้ Singularity ของการเคลื่อนที่ของแขนกลโดยการใช้ความเร็วในการเคลื่อนที่ของแต่ละข้อต่อนั้นมีการใช้ Library svd ของ scipy ใน python

ในส่วนการหาค่า Joint Effort หาจากสมการทาง Static Force โดยหาจากสมการ

$$\tau = J^T(q)w$$

เมื่อกำหนดให้ τ แทน ค่า Joint Effort

w แทน ค่า Wrench ที่กระทำที่ปลายแขน

บทที่ 4 ผลการดำเนินงาน

การทำงานของระบบจะแบ่งออกเป็นระบบ Input และระบบควบคุมการทำงาน MoveJ และ MoveL

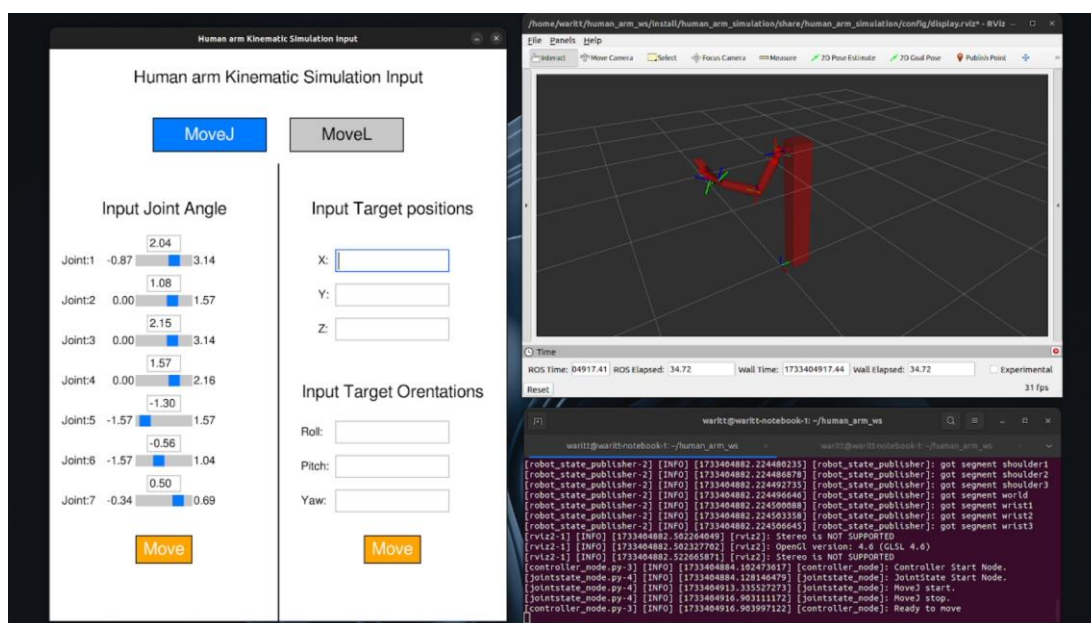
ระบบ Input ค่าสำหรับควบคุมการทำงาน

The figure shows four different input configurations for the 'Human arm Kinematic Simulation Input' window, labeled 1 through 4:

- 1 (MoveJ):** Shows 'Input Joint Angle' with sliders for Joint 1 to Joint 7. Values range from -1.57 to 3.14. A 'Move' button is at the bottom.
- 2 (MoveL):** Shows 'Input Target positions' with input fields for X, Y, and Z, and 'Input Target Orientations' with input fields for Roll, Pitch, and Yaw. A 'Move' button is at the bottom.
- 3 (Input Target Position):** Shows 'Input Target Position' with input fields for X, Y, and Z, and 'Input Target Orientation' with input fields for Roll, Pitch, and Yaw. A 'Move' button is at the bottom.
- 4 (Calculate):** Shows 'Wrench' and 'Joint Effort' sections. 'Wrench' includes 'Linear Force' (Force X, Y, Z) and 'Angular Torque' (Torque X, Y, Z). 'Joint Effort' lists Joint 1 to Joint 7 with values from 0.0000 to 0.69. A 'Calculate' button is at the bottom.

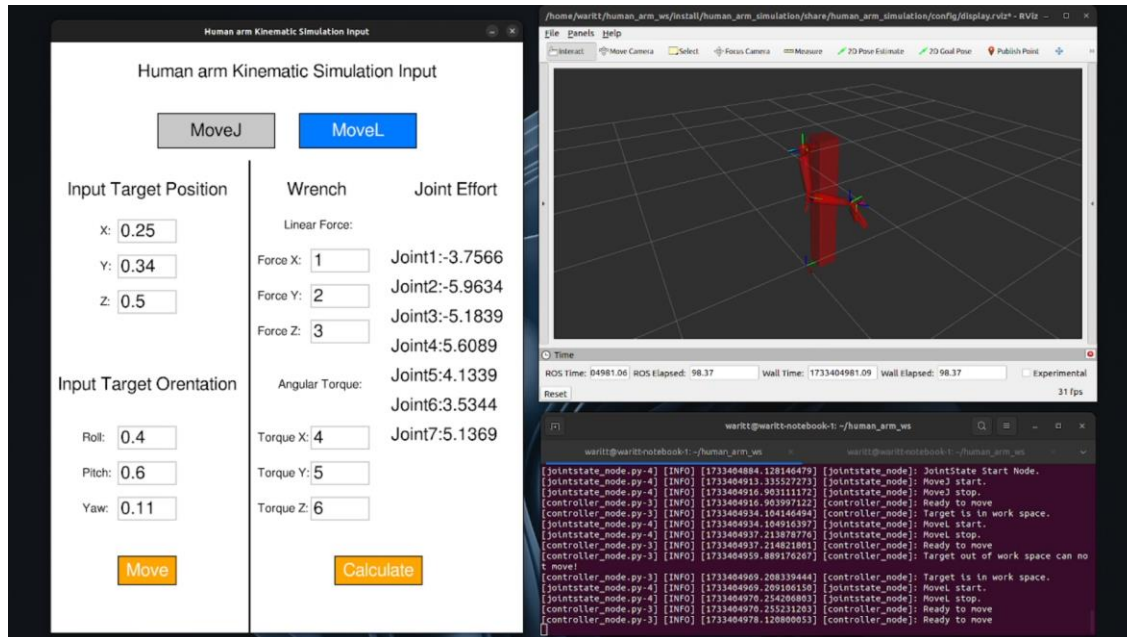
รูปที่ 8 หน้าต่างระบบ Input ค่าสำหรับการควบคุม

ระบบควบคุมการทำงาน MoveJ



รูปที่ 9 ตัวอย่างการทำงานในโหมด MoveJ

ระบบควบคุมการทำงาน MoveL



รูปที่ 10 ตัวอย่างการทำงานในโหมด MoveL

สามารถ Download ระบบการทำงานเพื่อทดสอบและดูรายละเอียดการทำงานฉบับเต็มได้ที่ GitHub :
https://github.com/PoFamily/human_arm_ws.git

บทที่ 5 สรุปผลการดำเนินงาน

สรุปผลการดำเนินงาน

ในการทำระบบจำลองการเคลื่อนไหวของแขนมนุษย์ภายใต้หัวข้อ 7 DOF Manipulator simulates the joint pattern of a Human arm Kinematics simulation ได้ดำเนินการสร้างระบบขึ้นโดยการใช้งานระบบของ ROS2 Humble ร่วมกับ Python libraries สำหรับการคำนวณและควบคุมการทำงานในส่วนต่าง ๆ โดยในการจำลองนั้นได้จำลองในส่วนของไหล่จนถึงข้อมือโดยแสดงผลการเคลื่อนที่ผ่าน RVIZ บนระบบ ROS2 โดยระบบที่สร้างขึ้นประกอบด้วย ระบบการควบคุม และระบบรับข้อมูล โดยระบบสามารถรับข้อมูลจาก Input UI ที่สร้างขึ้นเพื่อควบคุมการทำงานและสามารถสั่งงานผ่าน Service ได้เช่นกันส่วนระบบควบคุมจะแบ่งเป็นระบบ MoveJ ,MoveL และระบบคำนวณค่า Joint Effort ในส่วนของระบบที่ยังไม่ได้เดินการเพื่อเพิ่มความสมจริงของระบบจำลองคือการทดสอบเทียบกับแขนจริงเพื่อศึกษาถึงรูปแบบความเป็นไปได้ในการเคลื่อนที่ของแขนมนุษย์

แผนการการพัฒนาต่อ

1. ทดสอบการเคลื่อนที่ของระบบจำลองเทียบกับแขนมนุษย์จริงเพื่อหาข้อจำกัดให้ระบบจำลองเคลื่อนที่ได้เหมือนจริงมากที่สุด
2. เพิ่มข้อจำกัดที่แม่นยำขึ้นของ Joint limit ในทุกรูปแบบการเคลื่อนที่
3. เพิ่มการควบคุมในรูปแบบของ Trajectory เพื่อให้ระบบจำลองเคลื่อนที่ไปยังตำแหน่งเป้าหมายได้ราบรื่นยิ่งขึ้น
4. เพิ่มระบบจำลองในส่วนข้อมือเพื่อให้ได้ระบบจำลองการเคลื่อนที่ของแขนมนุษย์ที่สมบูรณ์มากขึ้น

สามารถนำระบบที่สร้างขึ้นไปศึกษารายละเอียดการทำงานและนำไปพัฒนาต่อยอดได้จาก GitHub :
https://github.com/PoFamily/human_arm_ws.git

อ้างอิง

- [1] Jingguo Wang, Yangmin Li, Xinhua Zhao. “Inverse Kinematics and Control of a 7-DOF Redundant Manipulator Based on the Closed-Loop Algorithm” International Journal of Advanced Robotic Systems, 2010.
- [2] Samuel R. Buss. “Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse and Damped Least Squares methods” Department of Mathematics University of California, 2009
- [3] Anthony A. Maciejewski, Charles A. Klein. “The Singular Value Decomposition : Computation and Applications to Robotics” The International Journal of Robotics Research, 1989