



7 DOF MANIPULATOR SIMULATES THE JOINT PATTERN OF A HUMAN ARM KINEMATICS SIMULATION

สมาชิก

นายไกรวิชญ์ วิชาโคตร, 65340500004
นายชยณัฐ เลิศวิทยานุกัษ์, 65340500012
นายชวณณ วชิรานิรมิต, 65340500014
นายวริทธิ์ธร คงหนู, 65340500050

บทคัดย่อ

โปรเจกต์นี้เป็นโปรเจกต์สำหรับการจัดทำ Simulation การควบคุมการทำงานของแขนกล 7 DOF ที่จำลองรูปแบบข้อต่อของแขนคนแบบ 3R – R – 3R โดยใช้การควบคุมผ่านค่าของแต่ละข้อต่อของแขนกล (Configuration space) , ตำแหน่งปลายแขนของกล (Task space) และมีการแสดงแรงที่เกิดขึ้นในแต่ละข้อต่อของแขนกล (Joint force) ที่ใช้ในการรักษาท่าทางเมื่อมีแรงมากระทำที่ปลายแขนกลในการสร้าง Simulation ใช้การสร้าง MDH - Parameter ของแขนกลสำหรับการทำ Simulation และการคำนวณ ต่างที่เกิดขึ้นในการ Simulation การควบคุมการทำงานของแขนกล

คำสำคัญ: Human arm, 7 DOF Manipulator (3R – R – 3R), Simulation

1. บทนำ

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

1.1. จุดประสงค์โครงการ

- 1.1.1. เพื่อสร้าง Simulation การเคลื่อนที่ของ Model แขนมนุษย์
- 1.1.2. เพื่อสร้างระบบจำลองการควบคุมการเคลื่อนที่ของแขนมนุษย์
- 1.1.3. เพื่อศึกษาแรงในข้อต่อของแขนมนุษย์ที่ต้องใช้เมื่อมีแรงมากระทำที่ปลายของแขนเพื่อรักษาท่าทาง
- 1.1.4. เพื่อศึกษาการประยุกต์ใช้ความรู้ทาง Kinematics ในการอธิบายการเคลื่อนที่และการควบคุมการเคลื่อนที่ของแขนมนุษย์

1.2. ขอบเขต

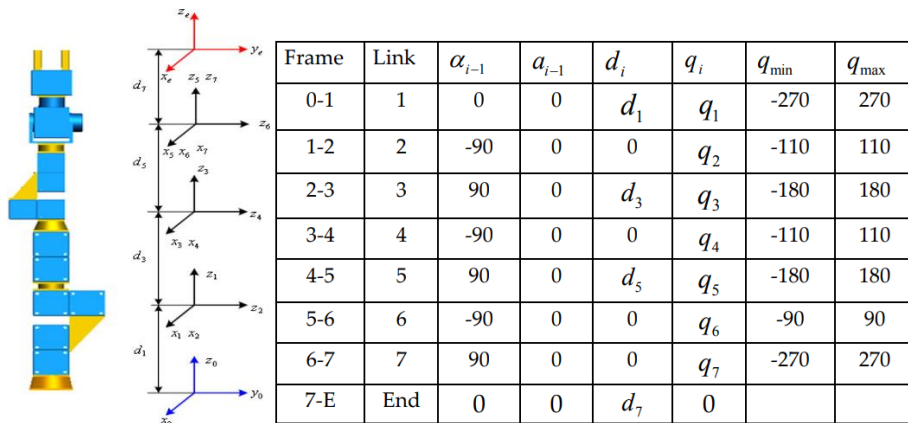
- 1.2.1. Simulation จำลองเคลื่อนที่ของแขนมนุษย์ตั้งแต่หัวไหล่จนถึงข้อมือ
- 1.2.2. ระบบควบคุมการเคลื่อนที่ที่สามารถควบคุมผ่าน Configuration space โดยการปรับค่า q ของแต่ละ joint
- 1.2.3. ระบบควบคุมการเคลื่อนที่ที่สามารถควบคุมผ่าน Task space โดยการกำหนด ตำแหน่งที่มีการอ้างอิงกับ reference frame
- 1.2.4. ระบบควบคุมการเคลื่อนที่ที่สามารถควบคุมผ่าน Task space ที่สามารถเคลื่อนที่แบบ MoveJ หรือ MoveL
- 1.2.5. ระบบคำนวณหาแรงในแต่ละ joint กรณีมีแรงภายนอกมากระทำที่จุดปลายของแขนใน Simulation เมื่อแขนอยู่กับที่

2. ทบทวนวรรณกรรม และ ทฤษฎีที่เกี่ยวข้อง (Literature Review)

2.1. Inverse Kinematics and Control of a 7-DOF Redundant Manipulator Based on the Closed-Loop Algorithm(Jingguo Wang, Yangmin Li, Xinhua Zhao)[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

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

จากการแปลง Jacobian matrix จะได้ความสัมพันธ์ของ Joint space จาก Jacobian Matrix คือ

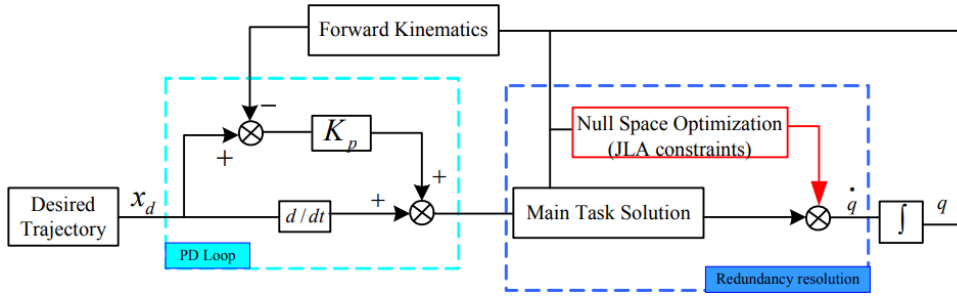
$$\dot{q} = J^+(q)\dot{x}$$

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

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

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

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

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

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

2.2. Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse and Damped Least Squares methods (Samuel R. Buss)[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

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

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 จะไม่เกิดการเคลื่อนที่ในทิศทางที่เป็นไปได้ แต่ถ้าอยู่ใกล้ Singularities การใช้ Pseudoinverse จะทำให้ $\Delta\theta$ มีการเปลี่ยนแปลงค่าอย่างมากแม้ว่า \vec{e} จะเปลี่ยนแปลงเพียงเล็กน้อย

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

$$\Delta\theta = J^+ \tilde{e} + (I - J^+) \phi$$

กำหนดให้ ϕ เป็นเวกเตอร์สำหรับเป้าหมายรองสามารถหา Pseudoinverse method ของ Jacobian matrix ได้โดยเริ่มจากสมการในรูปปกติ

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

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

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

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

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

การทำ Pseudoinverse method มีการใช้งานอย่างแพร่หลายแต่ก็ต้องระวังความไม่เสถียรเมื่อเข้าใกล้ Singularities

2.3. The Singular Value Decomposition : Computation and Applications to Robotics (Anthony A. Maciejewski, Charles A. Klein)[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 ซึ่งส่งผลต่อการควบคุมและลดความสามารถในการควบคุมของหุ่นยนต์ 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

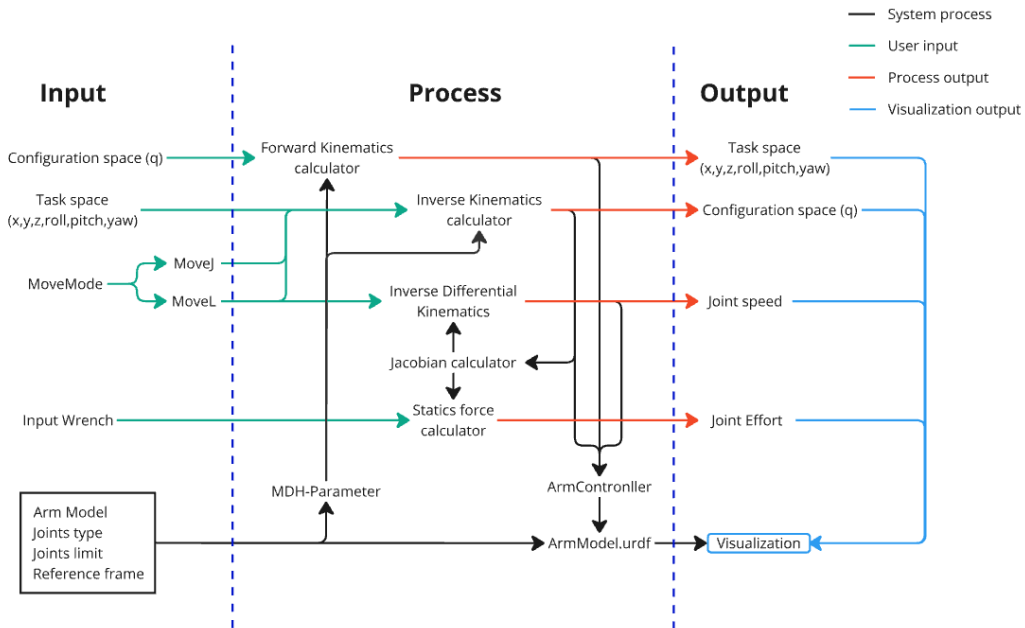
$\lambda > 0$

เป็น ตัวแปรลดผลกระทบจาก Singularity

3. เนื้อหาในรายวิชาที่เกี่ยวข้อง

- 3.1. Forward Kinematics
- 3.2. Inverse Kinematics
- 3.3. Differential Kinematics
- 3.4. Static force

4. System Diagram / System Overview (Function and Argument



รูปที่ 3 System Overview ของระบบการทำงาน

4.1. ส่วน Input ของระบบซึ่งแบ่งออกเป็น 2 ส่วน คือ

4.1.1. ส่วนที่ผู้ใช้งานสามารถ Input ให้กับระบบได้ ประกอบด้วย

- 4.1.1.1. Configuration Space (q) : ค่ามุมในแต่ละข้อต่อที่ใช้เพื่อควบคุมการเคลื่อนไหวของแต่ละข้อต่อ
- 4.1.1.2. Task Space ($x, y, z, \text{roll}, \text{Pitch}, \text{yaw}$) : ตำแหน่งเป้าหมายใน Simulation ที่ให้ปลายแขนไปถึง
- 4.1.1.3. Move Mode (MoveJ, MoveL) : โหมดการเคลื่อนที่
- 4.1.1.4. Input Wrench : ค่า wrench (force and torque) จากภายนอกที่กระทำต่อปลายแขนเมื่อระบบหยุดนิ่งอยู่กับที่

4.1.2. ส่วนที่ระบบจะรับ Input จาก setup ที่ถูกกำหนดไว้ ประกอบด้วย

- 4.1.2.1. Arm Model : Model แบบจำลองแต่ละส่วนของแขนมนุษย์
- 4.1.2.2. Joint Type: ประเภทของข้อต่อโดยกำหนดไว้เป็นแบบ Revolute ที่ถูกจัดเรียงแบบ 3R – R – 3R
- 4.1.2.3. Joint Limit : ขอบเขตการเคลื่อนไหวของแต่ละข้อต่อที่ถูกกำหนดไว้
- 4.1.2.4. Reference Frame

4.2. ส่วน Process ของระบบซึ่งแบ่งออกเป็น 7 ส่วน คือ

- 4.2.1. Model.urdf : Model แบบจำลองของแขนมนุษย์
- 4.2.2. MDH – Parameter : Parameter ที่อธิบายการเปลี่ยนแปลงของแต่ละข้อต่อของ Model แบบจำลองแขนมนุษย์สำหรับใช้ในการคำนวณส่วนอื่น ๆ ต่อไป
- 4.2.3. Forward Kinematics : การคำนวณที่ใช้ในการหา Task Space ($x, y, z, \text{roll}, \text{Pitch}, \text{yaw}$) จาก Configuration Space (q)
- 4.2.4. Inverse Kinematics : การคำนวณที่ใช้ในการหา Configuration Space (q) จาก Task Space ($x, y, z, \text{roll}, \text{Pitch}, \text{yaw}$)

- 4.2.5. Jacobian Matrix : การคำนวณที่ใช้ในการหา Jacobian Matrix ที่เกิดจากความสัมพันธ์ของการเปลี่ยนแปลง Configuration Space ที่ส่งผลต่อการเปลี่ยนแปลงของ Task Space
- 4.2.6. Inverse Differential Kinematics : การคำนวณที่ใช้ในการหาความเร็วของแต่ละข้อต่อโดยใช้ Inverse Jacobian Matrix ร่วมกับความเร็วของปลายแขน
- 4.2.7. Statics Force : การคำนวณเพื่อหาค่า Joints Effort โดยการใช้ Jacobian Transpose Matrix ร่วมกับค่า wrench (force and torque)
- 4.3. ส่วน Output ของระบบซึ่งแบ่งออกเป็น 2 ส่วน คือ
- 4.3.1. ส่วนของค่าที่ได้จากการคำนวณ
- 4.3.1.1. Task Space (x, y, z, roll, Pitch, yaw)
- 4.3.1.2. Configuration Space (q)
- 4.3.1.3. Joint Speed : ความเร็วในการเคลื่อนที่ของแต่ละข้อต่อ
- 4.3.1.4. Joint Effort : แรงบิดของแต่ละข้อต่อที่ใช้เพื่อรักษาท่าทางเมื่อมีแรงมากระทำที่ปลายแขน
- 4.3.2. ส่วนการแสดงผลที่แสดงผลการเคลื่อนไหวของ Model แบบจำลองแขนมนุษย์จากค่าต่าง ๆ ที่คำนวณได้

5. ผลการศึกษาที่คาดหวัง

- 5.1. Simulation การเคลื่อนที่ของ Model แขนกล 7 DOF ที่จัดเรียงข้อต่อแบบ 3R – R – 3R ที่สามารถควบคุมได้ผ่าน Configuration space และ Task space
- 5.2. การควบคุม Simulation ผ่าน Task space ในรูปแบบการเคลื่อนที่แบบ MoveJ และ MoveL
- 5.3. สามารถหาแรงในแต่ละ joint กรณีมีแรงภายนอกมากระทำที่จุดปลายของแขนกล
- 5.4. readme ขั้นตอนการนำ Simulation ที่สร้างขึ้นไปใช้งาน

6. รายละเอียดโครงการ

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						

7. เอกสารอ้างอิง

- [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