

การศึกษาการหา Inverse Kinematics ด้วย Geometry Method และเขียนโปรแกรมจำลองการเคลื่อนที่
สำหรับการใช้งานแบบ Pick and Paste ของ 3 DOF Delta Robot รุ่น IRB 340 ผ่าน Python ด้วย
Library Matplotlib

1. ชื่อหัวข้อ

การศึกษาการหา Inverse Kinematics ด้วย Geometry Method และเขียนโปรแกรมจำลองการ
เคลื่อนที่สำหรับการใช้งานแบบ Pick and Paste ของ 3 DOF Delta Robot รุ่น IRB 340 ผ่าน Python ด้วย
Library Matplotlib

2. รายละเอียดสมาชิก

นายณัฐชนนท์	สิงหา	64340500017
นางสาวธนพรรณ	เรืองสุข	64340500025

3. วัตถุประสงค์

1. เพื่อศึกษาลักษณะทางกายภาพ ของ 3 DOF Delta Robot รุ่น IRB 340
2. เพื่อศึกษาวิธีการหาสมการ Inverse Kinematics ของ 3 DOF Delta Robot รุ่น IRB 340
3. เพื่อคำนวณหาสภาวะ Singularity ที่เกิดจากการเคลื่อนที่ของ 3 DOF Delta Robot รุ่น IRB 340
4. เพื่อจำลองการเคลื่อนที่สำหรับการใช้งานแบบ Pick and Paste ของ 3 DOF Delta Robot รุ่น
IRB 340 ผ่าน Python ด้วย Library Matplotlib

4. รายละเอียดโครงงาน

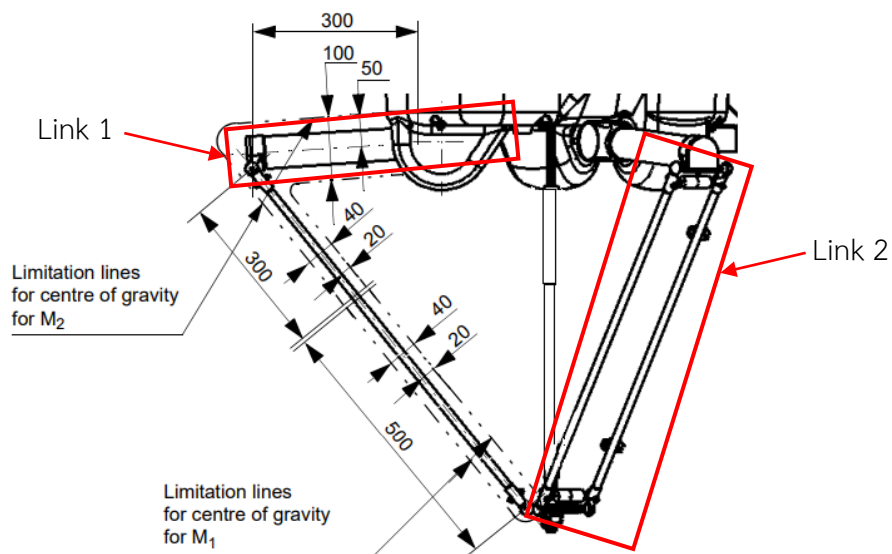
ผู้จัดทำทำการศึกษาลักษณะทางกายภาพ ลักษณะการทำงาน และความสัมพันธ์ในแต่ละ Joint ของ
Parallel Manipulator Robot ชนิด Delta Robot รุ่น IRB 340 เพื่อให้เข้าใจถึงการทำงานของหุ่นยนต์ชนิดนี้
จากนั้นทำการคำนวณหาสมการ Inverse Kinematics ด้วย Geometry Method ของหุ่นยนต์ชนิดนี้ เพื่อที่จะหา
องศาการเคลื่อนที่ของหุ่นยนต์แต่ละ Joint



Source: ABB Ltd

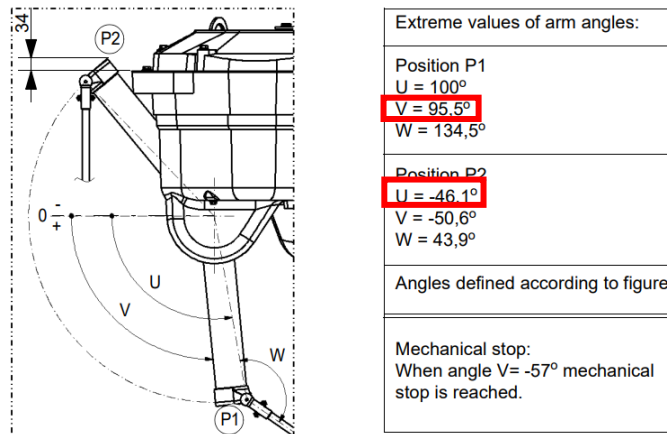
รูปที่ 1 Delta Robot (IRB 340)

โดยที่ Specification ของ Delta Robot รุ่น IRB 340 เป็นดังนี้
ความยาวของ Link 1 มีค่าเท่ากับ 300 mm และ Link 2 มีค่าเท่ากับ 800 mm และ Constrains ขององศาในการเคลื่อนที่ของแขนอยู่ที่ $[-46.1^\circ, 95.5^\circ]$



รูปที่ 2 Dimension ของ Delta Robot รุ่น IRB 340

ที่มา https://library.e.abb.com/public/805f728944b45950c1257b4b005230ba/3HAC9216-1_rev4_en.pdf



รูปที่ 3 Constrains ของ Joint

ที่มา https://library.e.abb.com/public/805f728944b45950c1257b4b005230ba/3HAC9216-1_rev4_en.pdf

ทางผู้จัดทำได้พัฒนา Simulation เพื่อจำลองการเคลื่อนที่ของ Delta Robot ในรูปแบบ Pick and Paste โดยเริ่มต้นจากการสร้างหน้า GUI ด้วยภาษา Python จาก Library Tkinter เป็นระบบส่วน Front-End ให้รับค่า Input พิกัดในแกน X Y และ Z จากผู้ใช้ โดยมี 2 โหมดด้วยกันได้แก่ โหมดที่ 1 คือ การกำหนดพิกัดในการเคลื่อนที่ให้กับ Delta Robot เป็นจุด และ โหมดที่ 2 คือ กำหนดพิกัดจุดในการเคลื่อนที่ทั้งหมดที่ต้องการที่จะให้ Delta Robot เคลื่อนที่ไป จากนั้นระบบจะทำการนำพิกัดเหล่านี้ส่งค่าไปยังระบบส่วน Back-End ที่มีการเขียนโปรแกรมเพื่อหาค่าองศาที่แต่ละ Joint เคลื่อนที่ไปด้วยสมการ Inverse Kinematics ดังที่ได้ศึกษาไปข้างต้น โดยในระหว่างการคำนวณ Inverse kinematics ถ้าหากคำนวณแล้วเกิด Singularity จะไม่ถูกนำค่าที่คำนวณได้จาก Inverse kinematics ไปใช้ และระบบจะทำการแจ้งเตือน ให้ผู้ใช้ทำการกรอกพิกัดใหม่เข้ามาอีกครั้ง เมื่อมีการคำนวณในส่วนของการหาค่าองศาที่แต่ละ Joint เสร็จแล้ว ในขั้นตอนถัดไปจะเป็นการนำค่าเหล่านั้นเข้าสู่กระบวนการ Generate Trajectory angle ของ Joint แต่ละ Joint ซึ่งในส่วนนี้จะเป็นการใช้ Code สำเร็จที่มีอยู่แล้ว เพื่อให้สามารถกำหนดการเคลื่อนที่ของหุ่นยนต์เพื่อไปถึงเป้าหมายได้

5. Literature Review

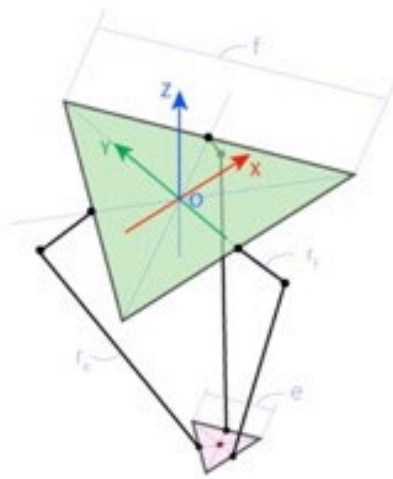
5.1 Parallel Manipulator Robot

เป็นแขนหุ่นยนต์ที่มีข้อต่อที่มี Actuator ขับเคลื่อนหลายตัวบริเวณฐาน นอกจากนี้ยังมีโครงสร้างข้อต่อจาก Actuator แต่ละตัวไปยังปลายแขนเป็นโครงสร้างแบบปิดเป็นวงรอบ เช่น (Delta Robot), Rostock (Delta Robot), Stewart Robot Platform เป็นต้น

5.2 Delta Robot

เป็นหุ่นยนต์ที่มีลักษณะเหมือนแมงมุม โดยขาของหุ่นยนต์จะมีการเคลื่อนที่ในทิศทางเดียวกันอย่างรวดเร็วเพื่อหยิบจับวัตถุ โดยหุ่นยนต์ชนิดนี้เหมาะที่จะนำไปใช้ในงานที่ทำในลักษณะงานซ้ำๆ และอาศัยความรวดเร็ว เช่น งานประเภท Pick and Paste วัตถุจากสายพานไปยังเป้าหมาย เป็นต้น

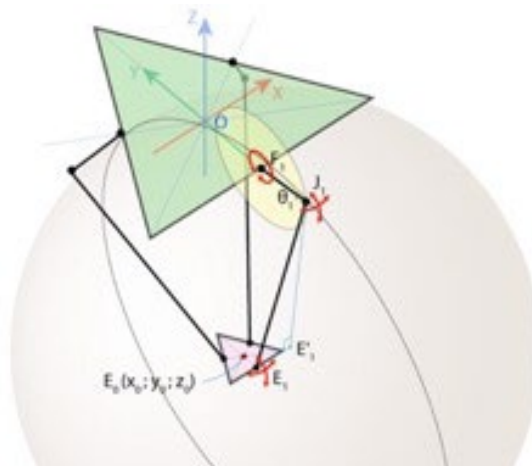
5.3 วิธีการหาสมการ Inverse kinematic ของหุ่นยนต์ Delta Robot ด้วย Geometry Method



รูปที่ 3 Free body diagram ของ Delta Robot

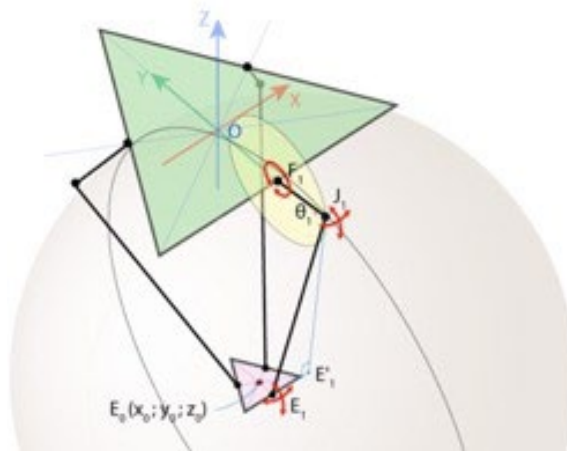
- กำหนดให้
- f คือ ความยาวด้านของสามเหลี่ยมที่ถูกตรึง (สีเขียว)
 - e คือ ความยาวด้านสามเหลี่ยม End-Effector (สีชมพู)
 - r_f คือ ความยาวของ Upper Joint
 - r_e คือ ความยาวของ Lower Joint

การวิเคราะห์ลักษณะทางกายภาพของ Delta Robot



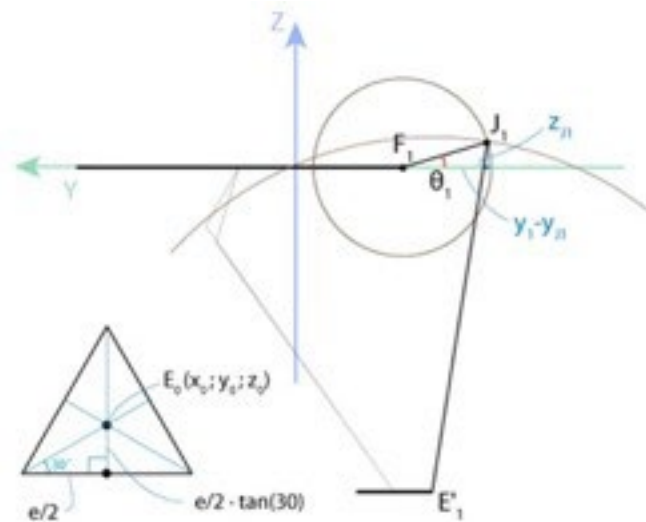
รูปที่ 4 Free body diagram ของ Delta Robot

1. Joint F_1J_1 หมุนแค่ใน YZ Plane โดยที่ F_1J_1 คือ รัศมี r_f
2. J_1 และ E_1 ถูกเรียกว่าเป็น Universal Joint เนื่องจาก E_1J_1 หมุนอย่างอิสระสัมพันธ์กับ E_1
3. สามเหลี่ยมที่ถูกตรึง (สีเขียว) ขนานกับสามเหลี่ยม End-Effector



รูปที่ 5 Free body diagram ของ Delta Robot

กำหนดให้ ตำแหน่งตรงกลางของ End- Effector คือ $E_0(x_0, y_0, z_0)$ และจงหา θ_i โดยที่ $i = 1, 2, 3$



รูปที่ 6 Free body diagram ของ Delta Robot

หาค่า θ_1 ได้จากสูตรเนื่องจาก E_1 คือภาพฉายของ E_0 บริเวณด้านล่างของ End-Effector บริเวณ XZ-plane

$$|EE_1| = \frac{e}{2} \tan 30^\circ = \frac{e}{2\sqrt{3}} \Rightarrow E_1(x_0, y_0 - \frac{e}{2\sqrt{3}}, z_0)$$

และ E'_1 คือภาพฉายของ E_1 บริเวณ YZ-plane

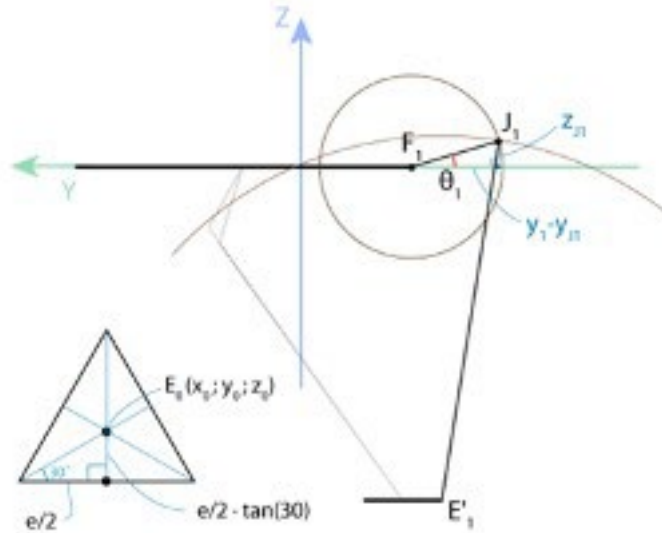
$$|E_1 E'_1| = x_0$$

สรุปสมการได้ว่า

$$|E'_1 J_1|^2 + |E_1 E'_1|^2 = |E_1 J_1|^2 = r_e^2$$

และ

$$|E_1 E'_1| = x_0$$



รูปที่ 7 Free body diagram ของ Delta Robot

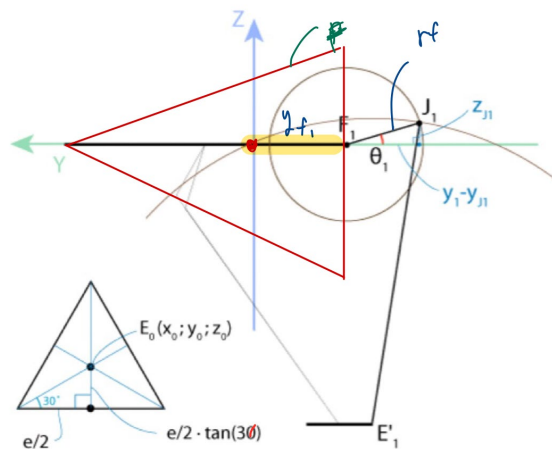
วงกลมทั้ง 2 วงมีการ Intersect กันเนื่องจาก

$$(y_{J_1} - y_{F_1})^2 + (z_{J_1} - z_{F_1}^2)^2 = r_f^2$$

ແລະ

$$(y_{J_1} - y_{E'_1})^2 + (z_{J_1} - z_{E'_1}^2)^2 = r_e^2 - x_0^2$$

ทำการหา ZJ_1 และ YJ_1 เพื่อหา θ_1 และทำเช่นเดียวกับวิธีข้างต้น เพื่อหา θ_2 และ θ_3 โดยที่ YF_1 สามารถหาได้จากการ Projection ร่วมกับตรีโกณมิติ



รูปที่ 8 Free body diagram ของ Delta Robot

ดังนั้น $YF_1 = \frac{f}{2} \tan 30^\circ$

เช่นเดียวกับ ZJ_1 โดยจะได้ $ZJ_1 = Y_0 + \frac{e}{2} \tan 30^\circ$

แทนค่า ZJ_1 และ YJ_1 ลงไปในสมการ จะได้

$$(YJ_1 + \frac{f}{2} \tan 30^\circ)^2 + ZJ_1^2 = r_f^2$$

$$(YJ_1 - Y_0 + \frac{e}{2} \tan 30^\circ)^2 + (ZJ_1 - Z_{E'_1})^2 = r_e^2 - X_0^2$$

ดังนั้น

$$\theta_1 = \arctan(\frac{ZJ_1}{YF_1 - YJ_1})$$

การคำนวณหา Singularity

จากความสัมพันธ์ทางด้าน Geometry และตรีโกณมิติ จะได้

$$\frac{x^2 + y_1^2 + z^2 + r_f^2 - r_e^2 - f_1^2}{2z} = a$$
$$\frac{f_1 - y_1}{z} = b$$

จากนั้นทำการหา Singularity ดังนี้

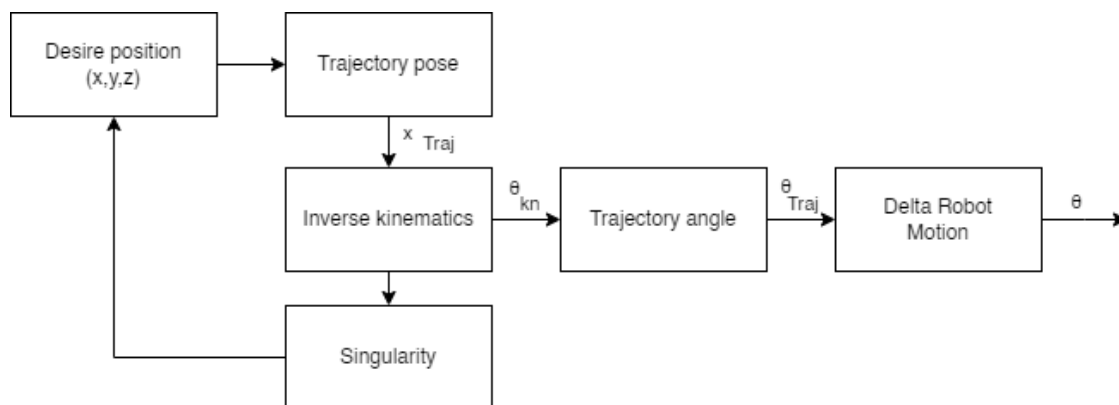
$$d = -(a + b * f_1)^2 + r_f + (b^2 * r_f + r_f)$$

หากค่า $d < 0$ จะเกิด Singularity

6. หัวข้อที่เกี่ยวข้องในรายวิชา

1. การหาสมการ Inverse Kinematics ด้วย Geometry Method
2. การหา Singularity
3. การเขียนโปรแกรมเพื่อจำลองการเคลื่อนที่ด้วย Python

7. System Diagram / System Overview



รูปที่ 9 System diagram

โดยระบบจะนำพิกัดในแกน X,Y และ Z ที่ผู้ใช้ป้อน Input เข้ามาในหน้า GUI นำไปคำนวณ Trajectory pose เพื่อ Generate Path ที่ Delta Robot ควรที่จะเคลื่อนที่ไป หลังจากนั้นนำพิกัด Path ของตำแหน่งปลายที่ได้ Generate มาไปคำนวณหาองศาของแต่ละ Joint ของหุ่นยนต์ และนำเข้าสู่กระบวนการ Generate Trajectory angle ของ Joint แต่ละ Joint เพื่อให้สามารถกำหนดการเคลื่อนที่ของหุ่นยนต์เพื่อไปถึงเป้าหมายได้

โดยในระหว่างการคำนวณ Inverse kinematics ถ้าหากคำนวณแล้วเกิด Singularity จะไม่ถูกนำค่าที่คำนวณได้จาก Inverse kinematics ไปใช้ และระบบจะทำการแจ้งเตือน ให้ผู้ใช้ทำการกรอกพิกัดใหม่เข้ามาอีกครั้ง

8. ผลการศึกษาที่คาดหวัง

1. ในระบบ Simulation หุ่นยนต์สามารถเคลื่อนที่ไปยังตำแหน่งที่รับ Input มาจากผู้ใช้ได้
2. สามารถตรวจจับพิกัดที่ทำให้เกิด Singularity ของ Delta Robot รุ่น IRB 340 ได้

9. อ้างอิง

- (1) X.Chen. (2020). Delta Robot kinematics 3dprinting–Building by learning, University of Washington
- (2) Chinedum Okwudire. (2020). Vibration Compensation of Delta 3D Printer with Position-varying Dynamics using Filtered B-Splines, University of Michigan
- (3) Meryam Rachadi. (2015). Design of an H^∞ controller for the Delta robot: experimental results
- (4) mzavatsky. (2009). Delta robot kinematics