



ใบรับรองวิทยานิพนธ์
บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์

วิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมอุตสาหการ)

ปริญญญา

วิศวกรรมอุตสาหการ

วิศวกรรมอุตสาหการ

สาขา

ภาควิชา

เรื่อง การวางแผนเส้นทางการเคลื่อนที่ของหุ่นยนต์เคลื่อนที่อิสระ 6 แกน 2 ตัวขณะทำการประกอบชิ้นงานร่วมกัน

Motion Path Planning for Work Assembly Cooperative Dual Six Joints Robots

นามผู้วิจัย นายสว่าง เป้นจันทร์

ได้พิจารณาเห็นชอบโดย

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(อาจารย์ชนะ รักย์ศิริ, D.Eng.)

อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม

(อาจารย์จักรพันธ์ อร่ามพงษ์พันธ์, Ph.D.)

หัวหน้าภาควิชา

(รองศาสตราจารย์อนันต์ มุ่งวัฒนา, Ph.D.)

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์รับรองแล้ว

(รองศาสตราจารย์กัญญา ชีระกุล, D.Agr)

คณบดีบัณฑิตวิทยาลัย

วันที่ เดือน พ.ศ.

วิทยานิพนธ์

เรื่อง

การวางแผนเส้นทางการเคลื่อนที่ของหุ่นยนต์เคลื่อนที่อิสระ 6 แกน 2 ตัว
ขณะทำการประกอบชิ้นงานร่วมกัน

Motion Path Planning for Work Assembly Cooperative Dual Six Joints Robots

โดย

นายสว่าง เป้นจันทร์

เสนอ

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์
เพื่อความสมบูรณ์แห่งปริญญาวิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมอุตสาหกรรม)

พ.ศ. 2551

สว่าง เป็นจันทร์ 2551: การวางแผนเส้นทางการเคลื่อนที่ของหุ่นยนต์เคลื่อนที่อิสระ 6 แกน 2 ตัวขณะทำการประกอบชิ้นงานร่วมกัน ปรินญาวิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมอุตสาหกรรม) สาขาวิศวกรรมอุตสาหกรรม ภาควิชาวิศวกรรมอุตสาหกรรม อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก: อาจารย์ชนะ รักษศิริ, D.Eng. 146 หน้า

การวิจัยนี้เป็นการวิจัยเกี่ยวกับการวางแผนเส้นทางการเคลื่อนที่ของหุ่นยนต์เคลื่อนที่อิสระหกแกน 2 ตัวขณะทำการสวมประกอบชิ้นงานเพลาร่วมกัน เพื่อลดเวลาที่ใช้ในการผลิต และเพิ่มความยืดหยุ่นในการปฏิบัติงาน โดยได้ศึกษาเปรียบเทียบกับเคลื่อนที่แบบเส้นตรง ซึ่งไม่สามารถทำการสวมชิ้นงานเพลาลักษณะเคลื่อนที่ได้ เนื่องจากระหว่างการเคลื่อนที่ได้เกิดการชนกันของชิ้นงานเพลาทิ้งสอง ส่วนวิธีการที่นำเสนอในงานวิจัยครั้งนี้จะทำการแบ่งตำแหน่งการเคลื่อนที่ในแต่ละเส้นทางให้มีระยะเท่าๆ กัน โดยได้แบ่งช่วงการเคลื่อนที่ออกเป็น 2 ช่วง ซึ่งช่วงแรกแบ่งตามจำนวนตำแหน่งที่มีการชนของปลายเครื่องมือ และช่วงสุดท้ายแบ่งตามค่าพิกัดความคลาดเคลื่อนของชิ้นงานทรงกระบอก และวงแหวน (R_{diff}) โดยเวลาการเคลื่อนที่ของหุ่นยนต์จะแปรผันไปตามจำนวนช่วงที่ได้แบ่งในแต่ละเส้นทางเคลื่อนที่ โดยผลการศึกษาพบว่าวิธีการวางแผนการเคลื่อนที่แบบแบ่งช่วงสามารถลดเวลาการเคลื่อนที่ และจำนวนช่วงการเคลื่อนที่ลงได้ร้อยละ 25.97 และ 24.11 โดยเฉลี่ย เมื่อเทียบกับการเคลื่อนที่แบบเส้นตรง

Sawang Panjun 2008: Motion Path Planning for Work Assembly Cooperative Dual Six Joints Robots. Master of Engineering (Industrial Engineering), Major Field: Industrial Engineering, Department of Industrial Engineering. Thesis Advisor: Mr.Chana Raksiri, D.Eng. 146 pages.

This paper describes a motion path planning for hole and shaft assembly cooperative dual six joints robots. The operation time is reduced and efficiency is increased flexible for various production processes. This study compared with result of tradition path planning which it can not make to assembly for hole and shaft. Because of during trajectory occurred collision for hole and shaft. The propose path planning divided position of trajectory that it have interval as same as. This path planning divided two steps. First step divided to quantity of collision position of end of joint and final step divided to different value between radius of hole and radius of shaft which it have maximum value not over rate of deviation value (R_{dif}). The operation time directed to quantity of interval for movement path. The study result is the propose path planning can reduce operation time and quantity of interval for motion path is 25.97 and 24.11 percent when compare with tradition path planning.

Student's signature

Thesis Advisor's signature

____ / ____ / ____

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จได้ด้วยความเอาใจใส่เป็นอย่างดีจาก อ.ดร.ชนะ รักษ์ศิริ ประธานที่ปรึกษาวิทยานิพนธ์หลัก ที่ได้แนะนำแนวทางต่างๆ วางแผนเรื่องการเรียนรู้ รวมทั้งให้โอกาสได้ทำงานวิจัยในครั้งนี้ อ.ดร.จักรพันธ์ อร่ามพงษ์พันธ์ ประธานที่ปรึกษาวิทยานิพนธ์ร่วม ที่ให้คำแนะนำและแนวทางในการทำวิจัย อ.ดร.วิสุทธิ สุพิทักษ์ ประธานการสอบปากเปล่าขั้นสุดท้าย และ ผศ.ดร.สถาพร อมรสวัสดิ์วัฒนา ผู้ทรงคุณวุฒิภายนอก ที่ได้ให้ความกรุณาในการแนะนำเนื้อหา และวิธีการทำงานวิจัยให้มีความสมบูรณ์มากยิ่งขึ้น ขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้

ข้าพเจ้าขอขอบคุณพระคุณบิดา มารดา และครอบครัวของข้าพเจ้าที่คอยเป็นกำลังใจและสนับสนุนจนทำให้วิทยานิพนธ์นี้สำเร็จได้ด้วยดี และขอบคุณทุกท่านที่ได้เอื้อนามที่มีส่วนในการช่วยเหลือให้งานวิทยานิพนธ์นี้สำเร็จด้วยดี

สว่าง แป้นจันทร์

สิงหาคม 2551

สารบัญ

	หน้า
สารบัญ	(1)
สารบัญตาราง	(2)
สารบัญภาพ	(3)
คำนำ	1
วัตถุประสงค์	3
การตรวจเอกสาร	4
อุปกรณ์และวิธีการ	18
วัสดุและอุปกรณ์	18
วิธีการ	18
ผลและวิจารณ์	44
สรุปและข้อเสนอแนะ	60
สรุป	60
ข้อเสนอแนะ	62
เอกสารและสิ่งอ้างอิง	63
ภาคผนวก	65
ประวัติการศึกษา และการทำงาน	146

สารบัญตาราง

ตารางที่	หน้า
1	แสดงค่าความยาวของข้อต่อของหุ่นยนต์ 6 แกนทั้ง 2 ตัว
2	แสดงค่าลักษณะ และขนาดของตัวอย่างชิ้นงานชุดที่ 1
3	แสดงค่าลักษณะ และขนาดของตัวอย่างชิ้นงานชุดที่ 2
4	แสดงค่าลักษณะ และขนาดของตัวอย่างชิ้นงานชุดที่ 3
5	แสดงค่าจำนวนตำแหน่ง และเวลาที่ใช้สำหรับการเคลื่อนที่ของตัวอย่าง ชิ้นงานชุดที่ 1
6	แสดงค่าจำนวนตำแหน่ง และเวลาที่ใช้สำหรับการเคลื่อนที่ของตัวอย่าง ชิ้นงานชุดที่ 2
7	แสดงค่าจำนวนตำแหน่ง และเวลาที่ใช้สำหรับการเคลื่อนที่ของตัวอย่าง ชิ้นงานชุดที่ 3
8	แสดงค่าจำนวนตำแหน่ง และเวลาที่ใช้ในการเคลื่อนที่ของหุ่นยนต์ เมื่อต้อง ทำประกอบชิ้นงานเพลาทังสองบริเวณปลายด้านหลังของชิ้นงานเพล ทรงกระบอก
9	แสดงค่าจำนวนตำแหน่ง และเวลาที่ใช้ในการเคลื่อนที่ของหุ่นยนต์ เมื่อต้อง ทำประกอบชิ้นงานเพลาทังสองบริเวณปลายด้านหน้าชิ้นงานเพล ทรงกระบอก

สารบัญภาพ

ภาพที่	หน้า
1 ตำแหน่งตัวแปรของข้อต่อของแขนกล	7
2 มือของแขนกล และ vector $[n \ o \ a \ p]$	9
3 แขนกลสามท่อนแบบ RRR	9
4 แขนกลสามท่อนแบบ RRR ใช้หาอินเวิร์ดไคเนแมติกส์	11
5 ลักษณะข้อต่อ และค่าองศาการเคลื่อนที่ของหุ่นยนต์ scara 6 แกน	20
6 แสดงตำแหน่งบนชิ้นงานเพลลาที่อ้างอิงการทำงานของปลายเครื่องมือหุ่นยนต์	23
7 แสดงตำแหน่งการเคลื่อนที่สำหรับการวางแผนเส้นทางการเคลื่อนที่ช่วงที่ 1	24
8 แสดงตำแหน่งการเคลื่อนที่สำหรับการวางแผนเส้นทางการเคลื่อนที่ช่วงที่ 2	31
9 ขนาดของชิ้นงานที่นำมาใช้ในการวางแผนการเคลื่อนที่ของหุ่นยนต์	45

ภาพผนวกที่

1 Flow Chart แสดงการหาค่าจำนวนตำแหน่งการเคลื่อนที่ช่วงที่ 1	67
2 Flow Chart แสดงการหาค่าจำนวนตำแหน่งการเคลื่อนที่ช่วงที่ 1 (ต่อ)	68
3 Flow Chart แสดงการหาค่าจำนวนตำแหน่งการเคลื่อนที่ช่วงที่ 2	69
4 Flow Chart แสดงการหาค่าจำนวนตำแหน่งการเคลื่อนที่ช่วงที่ 2 (ต่อ)	70
5 Flow Chart แสดงการหาค่าจำนวนตำแหน่งการเคลื่อนที่ช่วงที่ 2 (ต่อ)	71
6 Flow Chart แสดงการหาค่าจำนวนตำแหน่งการเคลื่อนที่ช่วงที่ 2 (ต่อ)	72
7 Flow Chart แสดงการหาค่าจำนวนตำแหน่งการเคลื่อนที่ช่วงที่ 2 (ต่อ)	73
8 Flow Chart แสดงการหาค่าจำนวนตำแหน่งการเคลื่อนที่ช่วงที่ 2 (ต่อ)	74
9 แสดงคุณลักษณะต่างๆ ของหุ่นยนต์สกาล่าหกแกน	141
10 แสดงลักษณะการวางของหุ่นยนต์สกาล่าสองตัวมาทำงานร่วมกัน	141
11 แสดงการกำหนดตำแหน่งการเคลื่อนที่ของหุ่นยนต์ A และ B	142
12 แสดงตำแหน่งการชนระหว่างเคลื่อนที่ของหุ่นยนต์ที่มีการเคลื่อนที่แบบเส้นตรง	142
13 แสดงตำแหน่งพร้อมสำหรับการประกอบชิ้นงานของการเคลื่อนที่แบบแบ่งช่วง	143

สารบัญภาพ (ต่อ)

ภาพผนวกที่	หน้า
14 แสดงตำแหน่งสุดท้ายของการเคลื่อนที่แบบแบ่งช่วง	143
15 แสดงลักษณะการเคลื่อนที่ของหุ่นยนต์ในช่วงแรกของการเคลื่อนที่	144
16 แสดงลักษณะการเคลื่อนที่ของหุ่นยนต์ในช่วงสุดท้ายของการเคลื่อนที่	145

คำอธิบายสัญลักษณ์และคำย่อ

X_j^i	คือ ตำแหน่งการเคลื่อนที่บนระนาบแกน X ของหุ่นยนต์ i ตำแหน่งที่ j
Y_j^i	คือ ตำแหน่งการเคลื่อนที่บนระนาบแกน Y ของหุ่นยนต์ i ตำแหน่งที่ j
Z_j^i	คือ ตำแหน่งการเคลื่อนที่บนระนาบแกน Z ของหุ่นยนต์ i ตำแหน่งที่ j
R_{dif}	คือ ค่าระยะความแตกต่างระหว่างรัศมีวงแหวน และรัศมีทรงกระบอก
d_6	คือ ความยาวจากปลายเครื่องมือจนถึงปลายข้อต่อที่ 3
R_0^6	คือ เมทริกซ์การหมุนจากฐานไปยังปลายข้อต่อที่ 6
l_i	คือ ความยาวของข้อต่อที่ i ; $i = 1, 2, 3$
p_i	คือ จำนวนตำแหน่งการเคลื่อนที่ สำหรับช่วงที่ i
t_{ij}	คือ เวลาในการเคลื่อนที่ของข้อต่อที่ i ตำแหน่งที่ j
α	คือ ความเร่งเชิงมุมสูงสุดของการเคลื่อนที่
θ_f	คือ องศาการเคลื่อนที่ ณ ตำแหน่งสุดท้าย
θ_i	คือ องศาการเคลื่อนที่ ณ ตำแหน่งเริ่มต้น
T'_{Total}	คือ เวลาการเคลื่อนที่ทั้งหมดของหุ่นยนต์ภายใต้ข้อกีดความเร็วในการหมุนของข้อต่อ
L_C	คือ ความยาวของทรงกระบอก
L_W	คือ ความยาวของวงแหวน
R_C	คือ รัศมีของทรงกระบอก
R_W	คือ รัศมีของวงแหวน
D_{AB}	คือ ระยะห่างของตำแหน่งฐานระหว่างหุ่นยนต์ A และ B
D_X	คือ ระยะห่างของปลายเครื่องมือที่ไม่เกิดการชนบนระนาบแกน X
D_Y	คือ ระยะห่างของปลายเครื่องมือที่ไม่เกิดการชนบนระนาบแกน Y
T_i^{AB}	คือ เวลาการเคลื่อนที่ของหุ่นยนต์ A และ B ช่วงที่ i
T_j^i	คือ เวลาการเคลื่อนที่ทั้งหมดของหุ่นยนต์ i ช่วงที่ j
X_{Goal}	คือ ตำแหน่งเป้าหมายสำหรับการเคลื่อนที่บนระนาบแกน X
Y_{Goal}	คือ ตำแหน่งเป้าหมายสำหรับการเคลื่อนที่บนระนาบแกน Y
Z_{Goal}	คือ ตำแหน่งเป้าหมายสำหรับการเคลื่อนที่บนระนาบแกน Z
$D_{X_t}^{AB}$	คือ ระยะห่างของปลายเครื่องมือขณะเคลื่อนที่บนระนาบแกน X ณ เวลา t
$D_{Z_t}^{AB}$	คือ ระยะห่างของปลายเครื่องมือขณะเคลื่อนที่บนระนาบแกน Z ณ เวลา t
$P_{X_t}^{Cy}$	คือ ตำแหน่งการเคลื่อนที่ของปลายทรงกระบอกขณะเคลื่อนที่บนระนาบ X ณ เวลา t

คำอธิบายสัญลักษณ์และคำย่อ (ต่อ)

P_Y^{Cy}	คือ ตำแหน่งการเคลื่อนของฟิวทรงกระบอกขณะเคลื่อนที่บนระนาบ Y ณ.เวลา t
P_Z^{Cy}	คือ ตำแหน่งการเคลื่อนของฟิวทรงกระบอกขณะเคลื่อนที่บนระนาบ Z ณ.เวลา t
$P_{j_t}^i$	คือ ตำแหน่งการเคลื่อนของหุ่นยนต์ i ขณะเคลื่อนที่บนระนาบ j ณ.เวลา t
$P_{j_t}^w$	คือ ตำแหน่งการเคลื่อนของปลายวงแหวนขณะเคลื่อนที่บนระนาบ j ณ.เวลา t
C_j^i	คือ ตำแหน่งการชนของหุ่นยนต์ i บนระนาบแกน j
AV_t^j	คือ ความเร็วสำหรับการเคลื่อนที่ของข้อต่อที่ j ณ.เวลา t
$AV_{\max}$	คือ ความเร็วสูงสุดสำหรับการเคลื่อนที่ของข้อต่อหุ่นยนต์
T_C	คือ เวลาสูงสุดสำหรับการเคลื่อนที่ใน 1 รอบของหุ่นยนต์ A และ B
T_R	คือ เวลาที่อยู่เหลืออยู่จากการเคลื่อนที่ในช่วงที่ 1 ของหุ่นยนต์ A และ B

การวางแผนเส้นทางการเคลื่อนที่ของหุ่นยนต์เคลื่อนที่อิสระ 6 แกน 2 ตัว ขณะทำการประกอบชิ้นงานร่วมกัน

Motion Path Planning for Work Assembly Cooperative Dual Six Joints Robots

คำนำ

ปัจจุบันนี้หุ่นยนต์ได้เข้ามามีบทบาทในส่วนของการผลิตสินค้าในวงการอุตสาหกรรมเพิ่มมากขึ้น นอกจากนี้แล้วยังมีการนำไปใช้งานในด้านของการสำรวจ ด้านการแพทย์ ด้านการทหาร ตำรวจ และข้าราชการทั่วไป โดยมีหลายๆ หน่วยงานได้ทำการวิจัย และคิดค้นพัฒนาการทำงานของหุ่นยนต์ให้สามารถปฏิบัติงานได้หลากหลายมากขึ้น ซึ่งได้มีการทำงานวิจัยที่เกี่ยวข้องกับการวางแผนเส้นทางการเคลื่อนที่ และควบคุมการทำงานของหุ่นยนต์ไว้อย่างมากมายทั้งหุ่นยนต์เคลื่อนที่ (Mobile Robots) และแขนกล (Manipulator Robots) โดยมีเป้าหมายของการทำวิจัยการควบคุมการทำงานของหุ่นยนต์ คือ เพื่อลดต้นทุนของการผลิตสินค้า เพิ่มประสิทธิภาพในการผลิตชิ้นงาน เพิ่มความรวดเร็วในการผลิตชิ้นงาน เพิ่มความยืดหยุ่นของการผลิตชิ้นงาน และเพิ่มความสามารถในการทำกำไรให้แก่องค์กร อันมีเหตุผลมาจากคุณสมบัติของเครื่องจักรกลเหล่านี้ที่สามารถตอบสนองความต้องการของการใช้งานในด้านต่างๆ ได้ดีกว่าแรงงานมนุษย์ทั้งความถูกต้องแม่นยำ ความมีเสถียรภาพในการทำงาน และความหลากหลายทางด้านประโยชน์การใช้งาน โดยงานวิจัยที่ได้มีการจัดทำในปัจจุบันนี้ส่วนใหญ่จะเน้นที่การวางแผนเส้นทางการเคลื่อนที่ และควบคุมการทำงานของหุ่นยนต์เพียงตัวเดียวเท่านั้น แต่สำหรับงานวิจัยที่นำหุ่นยนต์ตั้งแต่ 2 ตัวขึ้นไปมาทำการทดลองทำงานร่วมกันยังมีไม่ค่อยแพร่หลายมากนัก หรือยังมีอยู่ในวงจำกัด

ดังนั้นผู้วิจัยจึงได้ทำงานวิจัยเกี่ยวกับการวางแผนเส้นทางการเคลื่อนที่ และการควบคุมการทำงานของหุ่นยนต์ประเภทแขนกลที่มีองศาอิสระ 6 องศาอิสระ 2 ตัวให้สามารถทำงานร่วมกันได้โดยทำการประกอบชิ้นงานไปพร้อมกับการเคลื่อนย้ายชิ้นงานไปยังสถานีต่อไป ซึ่งการเคลื่อนในปัจจุบันเป็นแบบเส้นตรงโดยยังไม่สามารถนำมาคำนวณหาเส้นทางการเคลื่อนที่ได้ เนื่องจากในระหว่างที่มีการเคลื่อนที่ได้เกิดการชนกันของปลายเครื่องมือ หรือตัวชิ้นงานเพลาขึ้นทำให้ไม่สามารถสวมประกอบชิ้นงานเพลาขณะเคลื่อนที่ได้ ทั้งนี้ผู้วิจัยจึงได้คิดค้นหาวิธีการคำนวณเส้นทางการเคลื่อนที่แบบใหม่เพื่อให้หุ่นยนต์สามารถประกอบชิ้นงานเพลาขณะเคลื่อนที่ได้ ซึ่งอาจใช้เป็นแนวทางในการศึกษาสำหรับผู้สนใจทำงานวิจัยที่เกี่ยวข้องกับเรื่องนี้

ขอบเขตการศึกษา

การศึกษาครั้งนี้ได้มุ่งศึกษาเฉพาะหุ่นยนต์ที่มีองค์ประกอบเคลื่อนอิสระที่ 6 องศา 2 ตัวมาทำงานร่วมกันเท่านั้น โดยทำการศึกษาวิธีการแบ่งจำนวนตำแหน่ง และการวางแผนเส้นทางการเคลื่อนที่ของหุ่นยนต์ที่ได้ปฏิบัติงานร่วมกันในกระบวนการเคลื่อนย้ายชิ้นงาน พร้อมทั้งทำการสวมประกอบชิ้นงานเพื่อขณะทำการเคลื่อนย้ายชิ้นงานไปพร้อมๆ กัน สำหรับการเคลื่อนย้ายชิ้นงานเหล่านั้นเป็นการเคลื่อนย้ายจากสถานที่ทำการผลิตชิ้นงานเสร็จเรียบร้อยแล้วไปยังสถานีการผลิตต่อไปเท่านั้น สำหรับการศึกษาเปรียบเทียบวิธีการเคลื่อนที่ของหุ่นยนต์ จะศึกษาเปรียบเทียบกับการเคลื่อนที่แบบเส้นตรง ซึ่งเป็นวิธีที่มีการใช้อยู่ในปัจจุบันนี้ และชิ้นงานเพลานำมาศึกษามีอยู่ 2 ชิ้นด้วยกันคือ ชิ้นงานเพลที่เป็นวงแหวน และทรงกระบอกเท่านั้น

วัตถุประสงค์

เพื่อศึกษาการวางแผนเส้นทางการเคลื่อนที่ของหุ่นยนต์ 2 ตัวมาทำงานร่วมกัน โดยทำการ
สวมประกอบชิ้นงานเพลลา ขณะเคลื่อนที่จากสถานีหนึ่งไปยังสถานีหนึ่ง

การตรวจเอกสาร

การศึกษาการวางแผนเส้นทางการเคลื่อนที่ของหุ่นยนต์เคลื่อนที่อิสระหกแกน 2 ตัวขณะทำการประกอบชิ้นงานร่วมกันได้ทำการตรวจสอบเอกสารที่เกี่ยวข้อง โดยแบ่งเป็นทฤษฎีที่เกี่ยวข้องกับการเคลื่อนที่ การวางแผนการเคลื่อนที่ และงานวิจัยที่เกี่ยวข้องดังนี้

การเคลื่อนที่ของแขนกล

พัฒพงษ์ (2542) ได้กล่าวถึงส่วนประกอบที่เกี่ยวข้องกับการเคลื่อนที่ของแขนกลไว้ดังนี้

แขนกล คือ ส่วนประกอบระหว่างข้อต่อ (Joint) และท่อนแขน (Link) ที่มีความสัมพันธ์กัน ข้อต่อเป็นจุดหมุนที่ทำให้แขนเกิดการเคลื่อนที่ ในขณะที่ถือว่าแขนเป็น โครงสร้างที่แข็งแกร่งไม่มีการเปลี่ยนรูป แม้ว่าบางกรณีแขนจะสามารถอ่อนตัวได้ก็ตาม ในที่นี้จะยังไม่พิจารณาอิทธิพลของการอ่อนตัวได้ของแขน การเคลื่อนที่ของแขนที่ข้อต่ออาจทำได้หลายวิธี เช่น ใช้มอเตอร์ไฟฟ้า ใช้ระบบไฮดรอลิก หรือใช้เอ็นดิง เป็นต้น

ตัวแปรของข้อต่อในแขนกลจะมีอยู่สองตัว คือ องศาการเคลื่อนที่ของแต่ละข้อต่อ (θ) สำหรับระบบโครงข้อหมุน และความยาว (l) สำหรับระบบแกนเลื่อน

แกนหมุน คือ แกนที่แขนหมุนรอบจุดหมุนตามกฎมือขวาสำหรับระบบโครงข้อหมุน หรือ แกนที่ทับกับส่วนที่ยึด หรือหคของระบบแกนเลื่อน

1. จลนคณิตศาสตร์ของแขนกล (Robot Kinematics)

จลนคณิตศาสตร์ของแขนกล คือ การศึกษาการเคลื่อนที่ของแขนกลโดยไม่พิจารณาแรงที่เข้ามาเกี่ยวข้องแต่อย่างใด ซึ่งมีรายละเอียดดังต่อไปนี้

1.1 โฮโมจีเนียสทรานส์ฟอร์มเมชัน (Homogeneous Transformations)

โฮโมจีเนียสทรานส์ฟอร์มเมชันจะใช้ในการระบุตำแหน่งของปลายแกนกลแต่ละท่อน เทียบกับแกนก่อนหน้าจนถึงเฟรมอ้างอิงได้ ซึ่งจะมีรูปสมการทั่วไปดังนี้

$$A_{(i-1)i} = \begin{bmatrix} R_i & P_i \\ 0 & 1 \end{bmatrix} \text{ หรือ } A_{(i-1)i} = \begin{bmatrix} \text{Rotation_Matrix} & \text{Position_Vector} \\ \text{Perspective_Transformation} & \text{Scaling_Factor} \end{bmatrix}$$

Rotation Matrix เป็นเมตริกซ์ขนาด 3x3 ใช้ระบุขนาดการหมุนของแกน

Position Vector เป็นเวกเตอร์ที่ใช้ระบุขนาดของการเลื่อนตำแหน่ง มีขนาด 3x1

Perspective Transformation จะมีค่าเป็น 0 เสมอ ยกเว้นเมื่อใช้กับระบบเฟรมของกล้องถ่ายภาพ มีขนาด 1x3

Scaling Factor เป็นขนาดความยาวของเวกเตอร์ในกรณีของเวกเตอร์หนึ่งหน่วยจะมีค่าเท่ากับหนึ่งเสมอ

เมื่อหมุนรอบแกน X จะได้ Rotation Matrix คือ

$$Rot(X, \theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

เมื่อหมุนรอบแกน Y จะได้ Rotation Matrix คือ

$$Rot(Y, \theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

เมื่อหมุนรอบแกน Z จะได้ Rotation Matrix คือ

$$Rot(Z, \theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

เมื่อมีการเลื่อนตำแหน่งอย่างเดียวยตามแนวแกน X เท่ากับ p_x ตามแนวแกน Y เท่ากับ p_y ตามแนวแกน Z เท่ากับ p_z

$$Trans(P_x, P_y, P_z) = \begin{bmatrix} 1 & 0 & 0 & P_x \\ 0 & 1 & 0 & P_y \\ 0 & 0 & 1 & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

1.2 Arm T-Matrix

จากหลักการข้างต้นจะสามารถเทียบตำแหน่งของมือกับแขนแต่ละท่อนต่อกันเรื่อยๆ จนมาถึงฐาน หรือเฟรมอ้างอิงได้ ซึ่งเมทริกซ์ลักษณะนี้เรียกว่า ห่วงโซ่ของการเปลี่ยนตำแหน่ง (Kinematics chain of transformation)

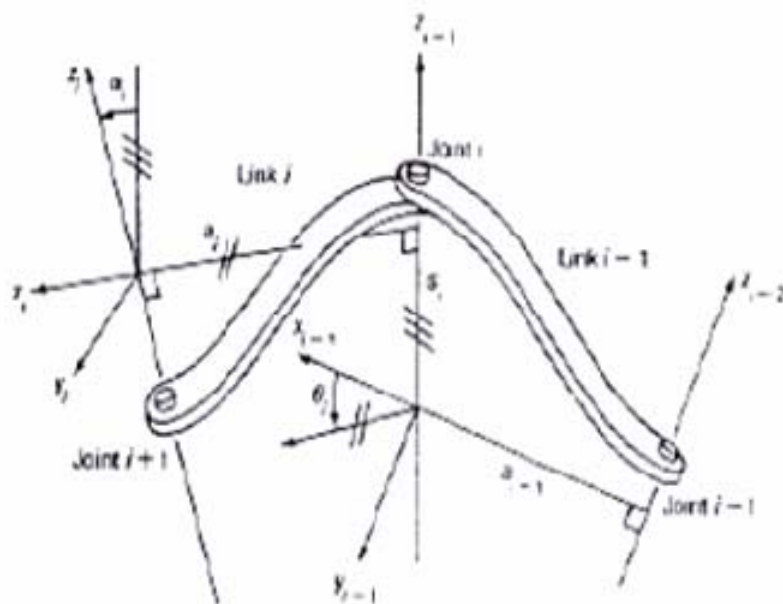
$$T_{0i} = A_{01}A_{12}A_{23}.....A_{(i-1)i} \quad (5)$$

ถ้าให้ ${}^i r$ เป็นค่าพิกัดของมือท่อนที่ i ค่าพิกัดที่เทียบกับฐาน ${}^0 r$ คือ

$${}^0 r = T_{0i} {}^i r \quad (6)$$

1.3 Denavit - Hartenberg (D-H)

เมื่อทราบค่าตัวแปรของข้อต่อ (Joint parameter; θ_i and s_i) และตัวแปรของท่อนแขน (Link parameter ; α_i and l_i) สามารถหาความสัมพันธ์ของแขนแต่ละท่อนได้ด้วยหลักการของ Denavit – Hartenberg แขนแต่ละท่อนจะใช้ตัวเลขระบุชื่อ เช่น ท่อนที่1, ท่อนที่2, ..., ท่อนที่ i แขนแต่ละท่อนจะมีเฟรม (x_i, y_i, z_i) เพื่อระบุค่าพิกัดของแขนแต่ละท่อนด้วย สำหรับแขนท่อนที่ 0 เฟรมที่ระบุค่าพิกัดจะมีชื่อเฉพาะอีกอย่าง คือ เฟรมอ้างอิง



ภาพที่ 1 ตำแหน่งตัวแปรของข้อต่อของแขนกล

ที่มา: พัฒนพงศ์ (2542)

โดยที่

θ_i คือ มุมที่ท่อนที่ i บิดไปเมื่อวัดเทียบกับเฟรม $(x_{i-1}, y_{i-1}, z_{i-1})$ บน

ระนาบ $x-y$

α_i คือ มุมที่วัดจากแกน $Z_{i-1} \rightarrow Z_i$

l_i คือ ระยะทางที่สั้นที่สุดจากแกน Z_{i-1} ไปยังแกน Z_i และต้องตั้งฉากกับแกนทั้ง

สองด้วย

s_i คือ ระยะทางจากจุดกำเนิดของเฟรม $(x_{i-1}, y_{i-1}, z_{i-1})$ ไปถึงจุดตัดของแกน x_i กับแกน Z_{i-1} (บางครั้งจะใช้สัญลักษณ์ d_i แทน s_i)

การหาความสัมพันธ์ระหว่างเฟรมระบุค่าพิกัดของแขนท่อนที่ $i-1$ กับเฟรมระบุค่าพิกัดของแขนท่อนที่ i จะใช้เมตริกซ์ดังต่อไปนี้

$$A_{(i-1)i} = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & l_i \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & l_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & s_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

- ถ้าเป็นข้อต่อแบบแกนหมุน θ_i เป็นตัวแปร, s_i เป็นค่าคงที่
- ถ้าเป็นข้อต่อแบบแกนเลื่อน s_i เป็นตัวแปร, θ_i เป็นค่าคงที่, l_i มีค่าเป็น 0
- ถ้าแขนกลมีจำนวน n ท่อน จะมีตัวแปรของข้อต่อ (เวกเตอร์ q) จำนวน n ตัว ซึ่งจะมีสมาชิกมุม θ_i หรือระยะความยาว s_i

1.4 ฟอว์เวิร์ดไคเนเมติกส์ (Forward Kinematics)

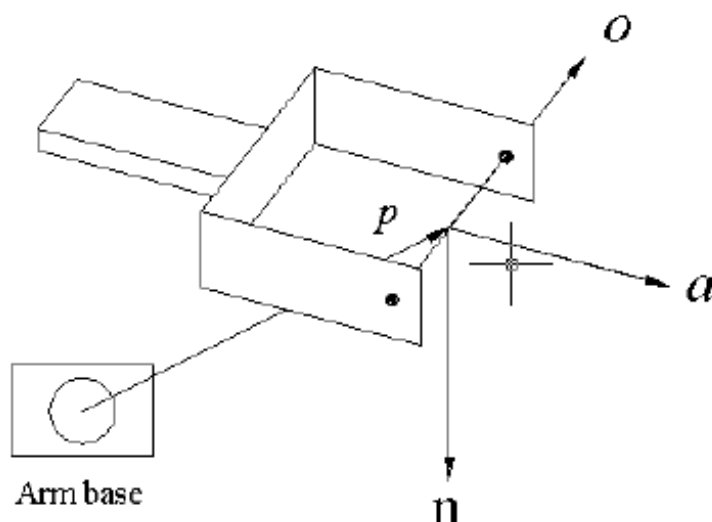
ฟอว์เวิร์ดไคเนเมติกส์ คือ การหาตำแหน่งของมือหรือปลายแขนแต่ละท่อน เมื่อทราบตัวแปรของข้อต่อ โดยใช้ Arm T – Matrix ในหัวข้อ 1.2 เพื่อแสดงความสัมพันธ์ของมือกับฐาน ในตอนนี้จะจัดรูปใหม่เพื่ออธิบายลักษณะตำแหน่งของมือ

$$T_{oi} = \begin{bmatrix} n & o & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R & P \\ 0 & 1 \end{bmatrix} \quad (8)$$

โดยที่

- n = normal vector
- a = approach vector
- o = orientation vector
- $n = o \times a$

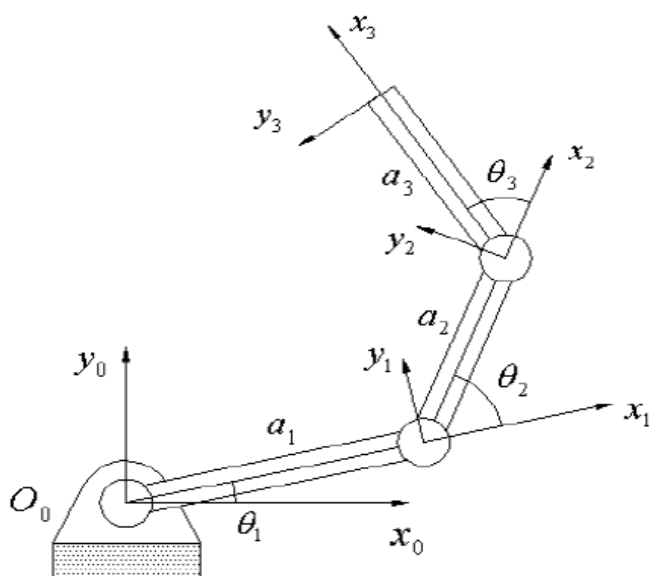
ลักษณะตำแหน่งของมือจะอธิบายด้วยเมตริกซ์ $R = [n \ o \ a]$ ส่วนเวกเตอร์ p ใช้ระบุตำแหน่งของจุดกำเนิดค่าพิกัดของมือเทียบกับฐาน



ภาพที่ 2 มือของแขนกล และ vector $[n \ o \ a \ p]$

ที่มา: พัฒนพงศ์ (2542)

ดังนั้น $[n \ o \ a]$ ก็คือ ค่าพิกัดของระบบแกน $X-Y-Z$ นั้นเองสำหรับแขนกลแบบสามท่อน สามารถแสดงให้เห็นได้ดังนี้



ภาพที่ 3 แขนกลสามท่อนแบบ RRR

ที่มา: พัฒนพงศ์ (2542)

เราสามารถทำการคำนวณหาจุดพิกัดของปลายแขนกลได้ดังนี้

$$A_{01} = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & a_1c\theta_1 \\ s\theta_1 & c\theta_1 & 0 & a_1s\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (9)$$

$$A_{12} = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & a_2c\theta_2 \\ s\theta_2 & c\theta_2 & 0 & a_2s\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (10)$$

$$A_{23} = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & a_3c\theta_3 \\ s\theta_3 & c\theta_3 & 0 & a_3s\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

$$T = A_{01}A_{12}A_{23} = \begin{bmatrix} c\theta_{123} & -s\theta_{123} & 0 & a_1c\theta_1 + a_2c\theta_{12} + a_3c\theta_{123} \\ s\theta_{123} & c\theta_{123} & 0 & a_1s\theta_1 + a_2s\theta_{12} + a_3s\theta_{123} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (12)$$

โดยที่

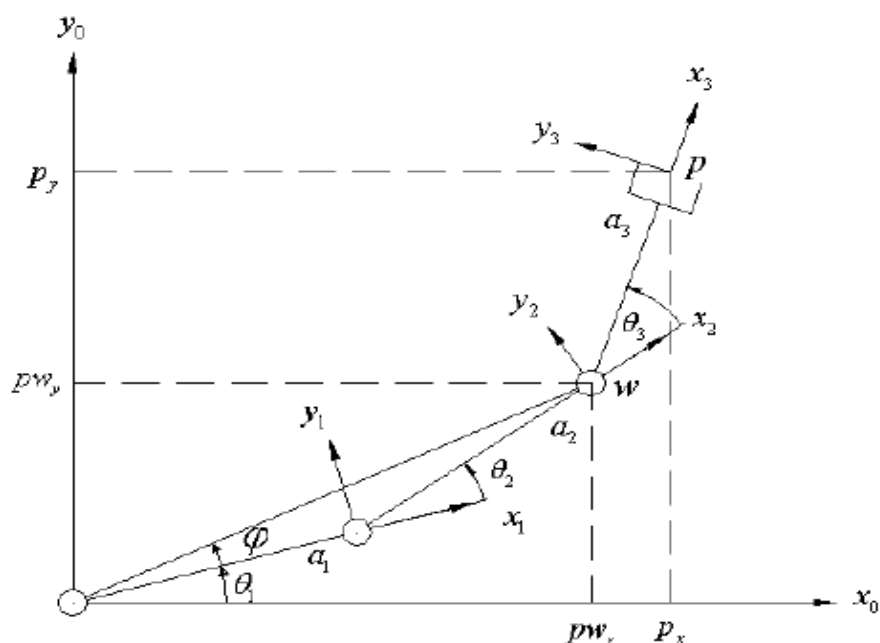
$$\begin{aligned} c\theta_1 &= \cos\theta_1 \\ s\theta_1 &= \sin\theta_1 \\ c\theta_{12} &= \cos(\theta_1 + \theta_2) \\ s\theta_{12} &= \sin(\theta_1 + \theta_2) \\ c\theta_{123} &= \cos(\theta_1 + \theta_2 + \theta_3) \\ s\theta_{123} &= \sin(\theta_1 + \theta_2 + \theta_3) \end{aligned}$$

จะเห็นว่าเมื่อทราบค่าตัวแปรของข้อต่อ $(\theta_1, \theta_2, \theta_3)$ สามารถหาจุดกำเนิดของเฟรม 3 เทียบกับเฟรมอ้างอิงได้ ซึ่ง

$$p = [a_1c\theta_1 + a_2c\theta_{12} + a_3c\theta_{123} \quad a_1s\theta_1 + a_2s\theta_{12} + a_3s\theta_{123}]^T \quad (13)$$

1.5 อินเวอร์คไคนเมติกส์ (Inverse Kinematics)

อินเวอร์คไคนเมติกส์ คือ การหาค่าตัวแปรของข้อต่อ ($\theta_1, \theta_2, \theta_3$) เมื่อทราบลักษณะตำแหน่งของมือหรือทราบเวกเตอร์ $[n \ o \ a \ p]$ นั่นเอง



ภาพที่ 4 แขนกลสามท่อนแบบ RRR ใช้หาอินเวอร์คไคนเมติกส์

ที่มา: พัฒนพงศ์ (2542)

พิจารณาแขนกลสามท่อนจากภาพที่ 4 ตัวแปรที่จำเป็นต้องทราบเพื่อใช้หาค่า $\theta_1, \theta_2, \theta_3$ คือค่าพิกัดของปลายแขนท่อนที่สาม (p_x, p_y) และลักษณะการวางตัวของแขนท่อนที่สาม (ϕ) หรือมุมระหว่างแขนท่อนที่สามกับแกน x_0 นั่นเอง จากรูปจะได้รับความสัมพันธ์ดังนี้

$$\phi = \theta_1 + \theta_2 + \theta_3 \quad (14)$$

ค่าพิกัดของเฟรมของแขนท่อนที่สอง

$$\begin{aligned} pw_x &= p_x - a_3 c_\phi \\ &= a_1 c \theta_1 + a_2 c \theta_{12} \end{aligned} \quad (15)$$

$$\begin{aligned} pw_y &= p_y - a_3 s_\phi \\ &= a_1 s \theta_1 + a_2 s \theta_{12} \end{aligned} \quad (16)$$

จากทฤษฎี Cosine Theorem จะได้

$$(pw_x)^2 + (pw_y)^2 = a_1^2 + a_2^2 - 2a_1 a_2 c(\pi - \theta_2) = a_1^2 + a_2^2 + 2a_1 a_2 c \theta_2 \quad (17)$$

$$\cos \theta_2 = \frac{(pw_x)^2 + (pw_y)^2 - a_1^2 - a_2^2}{2a_1 a_2} \equiv C \quad (18)$$

$$\sin \theta_2 = \pm \sqrt{1 - \cos^2 \theta_2} = \pm \sqrt{1 - C^2} \equiv D(13) \quad (19)$$

$$\theta_2 = A \tan\left(\frac{D}{C}\right) \quad (20)$$

$$\tan \varphi = \frac{a_2 \sin \theta_2}{a_1 + a_2 \cos \theta_2} \quad (21)$$

$$\tan(\varphi + \theta_1) = \frac{pw_y}{pw_x} \quad (22)$$

$$\theta_1 = A \tan\left(\frac{pw_y}{pw_x}\right) - A \tan\left(\frac{a_2 \sin \theta_2}{a_1 + a_2 \cos \theta_2}\right) \quad (23)$$

$$X = p_x - pw_x \quad (24)$$

$$Y = p_y - pw_y \quad (25)$$

$$\phi = A \tan\left(\frac{Y}{X}\right) \quad (26)$$

$$\theta_3 = \phi - \theta_1 - \theta_2 \quad (27)$$

1.6 สมการคำนวณเวลาการเคลื่อนที่ของแต่ละข้อต่อ

สถาพร(2545) ได้กล่าวไว้ การคำนวณเวลาในการเคลื่อนที่ของแต่ละข้อต่อของหุ่นยนต์ สามารถคำนวณได้สมการที่ (28) ดังนี้

$$t_{ij} = \sqrt{\frac{6|\theta_f - \theta_i|}{\alpha}} \quad (28)$$

โดยที่ t_{ij} คือ เวลาในการเคลื่อนที่ของ Link ที่ i ตำแหน่งที่ j
 α คือ ความเร่งเชิงมุมสูงสุดของการเคลื่อนที่
 θ_f คือ มุมสุดท้ายที่ต้องการเคลื่อนที่
 θ_i คือ มุมเริ่มต้นที่ต้องการเคลื่อนที่

การวางแผนการเคลื่อนที่ และการควบคุมหุ่นยนต์

สถาพร(2545) ได้กล่าวไว้ว่า การวางแผนการเคลื่อนที่ (Trajectory Planning) คือ การวางแผนให้หุ่นยนต์ หรือแขนกลเคลื่อนที่จากจุดเริ่มต้นไปยังจุดสุดท้าย โดยทำการควบคุมผ่านผู้ควบคุม หรือโดยการโปรแกรม และในระหว่างที่มีการเคลื่อนที่นั้นหุ่นยนต์ต้องหลบหลีกสิ่งกีดขวาง และเคลื่อนที่ไปตามเส้นทางที่ได้วางแผนไว้ โดยจะคำนึงถึงระยะเวลาในแต่ละเส้นทางที่เคลื่อนที่ และพิจารณาถึงความเร็ว และความเร่งของหุ่นยนต์อีกด้วย

เส้นทาง (Path) หมายถึง ลำดับการเคลื่อนที่ของหุ่นยนต์เป็นไปตามลำดับขั้นตอนที่กำหนด โดยไม่คำนึงถึงระยะเวลา

ในการคำนวณทราเจกทอรีแบบมิติของข้อต่อ (Joint space trajectories) นี้จะพิจารณาแผนการเคลื่อนที่ในรูปฟังก์ชันของมุมในแต่ละข้อต่อ (Function of joint angles) ซึ่งรูปแบบของมุมมีหลายรูปแบบ เช่น ฟังก์ชันพหุนาม (Polynomial) ฟังก์ชันเส้นตรงผสมพาราโบลา (linear function with parabolic blends) เป็นต้น สำหรับในแต่ละจุดของเส้นทาง และการหมุนที่ต้องการที่ปลายแขนกลสัมพันธ์กับแกนที่ฐาน (stationary frame) ในแต่ละจุดที่ต้องการถูกเปลี่ยนจะเป็นเซตของมุมที่ต้องการโดยใช้วิธีจลนศาสตร์ผกผันพิจารณาการเคลื่อนที่ของแขนจากตำแหน่งเริ่มต้นที่เวลา t_0 ไปตำแหน่งสุดท้ายที่เวลา t_f แล้วใช้จลนศาสตร์ผกผันคำนวณหามุมที่ตรงกันกับตำแหน่งและการหมุนที่ต้องการ โดยตำแหน่งเริ่มต้นของแขนกลจะทราบค่าในรูปของมุมของข้อต่อ (joint angles) คือ ฟังก์ชันแต่ละข้อต่อเริ่มต้นที่เวลา t_0 และเวลาสุดท้าย t_f

สำหรับการหาค่าฟังก์ชันการทำงานที่ราบเรียบ (Smooth function) นั้นอย่างน้อยต้องทราบเงื่อนไขของ $\theta(t)$ อย่างน้อย 4 เงื่อนไข โดยอย่างน้อย 2 เงื่อนไขมาจากค่าเริ่มต้น และค่าสุดท้าย นั่นคือ $\theta(0) = \theta_0$ และ $\theta(t) = \theta_f$ ส่วนอีก 2 เงื่อนไขมาจากกรณีฟังก์ชันต่อเนื่องได้ว่า $\dot{\theta}(0) = 0$ และ

$\dot{\theta}(t_f) = 0$ ตามลำดับ ซึ่งเราสามารถนำมาวิเคราะห์ได้โพลิโนเมียลดีกรีสาม (Third degree polynomials) หรือ cubic โดยมีรูปสมการคือ

$$\theta(t) = \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 t^3 \quad (29)$$

ส่วนความเร็ว (Velocity) และความเร่ง (Acceleration) ในรูปแบบของ $\frac{d}{dt}\theta(t)$ และ $\frac{d}{dt}\dot{\theta}(t)$ ตามลำดับ

$$\begin{aligned} \dot{\theta}(t) &= \frac{d}{dt}\theta(t) = \frac{d}{dt}(\beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 t^3) \\ &= \beta_1 + 2\beta_2 t + 3\beta_3 t^2 \end{aligned} \quad (30)$$

$$\begin{aligned} \ddot{\theta}(t) &= \frac{d}{dt}\dot{\theta}(t) = \frac{d}{dt}(\beta_1 + 2\beta_2 t + 3\beta_3 t^2) \\ &= 2\beta_2 + 6\beta_3 t \end{aligned} \quad (31)$$

จากสมการเงื่อนไขทั้ง 4 เงื่อนไขโดยนำมาแทนค่าลงในสมการความเร็ว และความเร่ง สามารถสร้างเป็นสมการที่ (32) ได้คือ

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 0 & 1 & 2t_0 & 3t_0^2 \\ 1 & t_f & t_f^2 & t_f^3 \\ 0 & 1 & 2t_f & 3t_f^2 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = \begin{bmatrix} \theta_0 \\ \dot{\theta}_0 \\ \theta_f \\ \dot{\theta}_f \end{bmatrix} \quad (32)$$

โดยสามารถหาผลลัพธ์จากสมการ Matrix ได้ว่า

$$\beta_0 = \theta_0 \quad (33)$$

$$\beta_1 = 0 \quad (34)$$

$$\beta_2 = \frac{3}{t_f^2}(\theta_f - \theta_0) \quad (35)$$

$$\beta_3 = -\frac{2}{t_f^3}(\theta_f - \theta_0) \quad (36)$$

โดยความเร็วเชิงมุม (Angular Velocity) ของแขนกลของหุ่นยนต์สามารถทำการเคลื่อนที่ได้ไม่เกิน 15 รอบต่อวินาที (1.571 rad/s) ซึ่งเราสามารถนำมาคำนวณเวลาในการเคลื่อนที่ได้ดังนี้

$$\begin{aligned}\theta(t) &= \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 t^3 \\ \dot{\theta}(t) &= 2\beta_2 t + 3\beta_3 t^2\end{aligned}\tag{37}$$

ข้อจำกัดของความเร็วเชิงมุมจะสามารถแปลงให้อยู่ในหน่วย องศาต่อวินาที คือ 90 องศา/วินาที สามารถคำนวณหาเวลาที่ต้องใช้ในการเคลื่อนที่เพิ่มขึ้นได้ดังนี้

$$2\beta_2 t + 3\beta_3 t^2 \leq 90\tag{38}$$

$$3\beta_3 t^2 + 2\beta_2 t - 90 = 0$$

$$t = \frac{-2\beta_2 \pm \sqrt{4\beta_2^2 + 1080\beta_3}}{6\beta_3}\tag{39}$$

$$T_L = \frac{-2\beta_2 - \sqrt{4\beta_2^2 + 1080\beta_3}}{6\beta_3}\tag{40}$$

$$T_U = \frac{-2\beta_2 + \sqrt{4\beta_2^2 + 1080\beta_3}}{6\beta_3}\tag{41}$$

$$T_{Add} = \left| \frac{\theta(T_U) - \theta(T_L)}{90} \right|\tag{42}$$

$$T_{Total_New} = T_{Total} + T_L - T_U + T_{Add}\tag{43}$$

งานวิจัยที่เกี่ยวข้อง

Koga and Latombe (1994) ได้ศึกษาการสร้างเส้นทางการเคลื่อนที่แบบอัตโนมัติของแขนกลที่ทำงานร่วมกันในหลายๆ รูปแบบ ซึ่งทำการเคลื่อนย้ายวัตถุผ่านสิ่งกีดขวางแบบไม่เกิดการชนกับสิ่งกีดขวาง โดยมีการเปลี่ยนแปลงการจับวัตถุจากแขนกลหนึ่งไปยังแขนกลตัวอื่นซึ่งมีการใช้แขนกล 3 ตัวมาทำงานร่วมกัน ผลการทำงานของแขนกลตามวิธีการนี้พบว่านอกจากแขนกลจะสามารถทำงานได้รวดเร็วแล้วยังไม่เกิดการชนกันของวัตถุกับสิ่งกีดขวางอีกด้วย

Liu and Dai (2003) ได้ศึกษาวิธีการเคลื่อนที่ของแขนกลในงานพับกล่องบรรจุสินค้า โดยกำหนดให้กล่องเป็นระบบเครื่องกลอย่างหนึ่งที่แต่ละส่วนมีการเชื่อมต่อกัน และกำหนดตำแหน่งการพับของแขนกลตามรูปแบบการเคลื่อนที่ และลำดับการพับของกล่อง มีการสร้างเส้นทางเชื่อมต่อกันโดยดูจากตำแหน่งที่ควบคุมรูปร่างของกล่อง โดยจำนวนของแขนกลที่นำมาใช้ทำงานร่วมกันต้องมีจำนวนน้อยที่สุด ผลจากการทำงานพบว่าต้องมีการแขนกล 2 ตัวมาทำงานร่วมกันจึงสามารถพับกล่องได้

Desai *et al.* (1998) ได้ศึกษาวิธีการวางแผนการเคลื่อนที่ของแขนกลที่ทำงานร่วมกัน ซึ่งมีการกำหนดตำแหน่งเริ่มต้นและตำแหน่งเป้าหมาย โดยระหว่างเคลื่อนที่ต้องมีการหลีกเลี่ยงอุปสรรคที่จะเกิดขึ้น มีการแบ่งการศึกษาออกเป็น 2 ส่วน โดยส่วนแรกเป็นการสร้างเส้นทางการเคลื่อนที่ และศึกษาการกระจายของแรงขณะแขนกล เคลื่อนที่ ส่วนที่สองกำหนดกลุ่มวิธีการวางแผนการเคลื่อนที่ของแขนกลโดยยังอยู่ภายใต้ข้อจำกัดของการเคลื่อนที่ และสร้างสมการสำหรับคำนวณเพื่อป้องกันการชน และสมการคำนวณการกระจายของแรงขณะเคลื่อนที่ โดยนำผลที่ได้มาทดสอบการเคลื่อนที่ในระบบที่ได้มีการออกแบบไว้

Quinlan And Williams (1997) ได้ศึกษาวิธีการวางแผน และควบคุมการทำงานของหุ่นยนต์ เป็นการผสมผสานกันระหว่างการวางแผน และการควบคุมการเคลื่อนที่ของหุ่นยนต์ โดยมีเป้าหมายในการพัฒนาระดับการทำงานของตัวเซ็นเซอร์ซึ่งเป็นปัจจัยขั้นพื้นฐานในการวางแผน และควบคุมการเคลื่อนที่ของหุ่นยนต์ โดยได้แสดงถึงรูปแบบ และหลักการขั้นพื้นฐานของการเคลื่อนที่ในการพัฒนาปัญหาต่อไป

Shin and Zheng (1989) ได้ศึกษาวิธีการวางแผนการเคลื่อนของแขนกลที่ทำงานร่วมกันซึ่งใช้เวลาเคลื่อนที่น้อยที่สุด โดยการเคลื่อนที่นี้จะไม่มีการชนกันระหว่างมีการเคลื่อนที่ สำหรับการทำงานร่วมกันของหุ่นยนต์นั้นจะสำเร็จได้ต้องมีหุ่นยนต์ตัวหนึ่งเคลื่อนที่ได้ช้ากว่าอีกตัวหนึ่ง โดยวิธีที่ดีที่สุดคือการพิจารณาการทดสอบเส้นทางที่แน่นอนตายตัว และเหมาะสมที่สุด

Yao and Gupta (2005) ได้ศึกษาการวางแผนเส้นทางปลายเครื่องของแขนกล และมีแนวทางการวางแผนอยู่ 2 วิธี วิธีแรกเป็นการปรับโดยการสุ่มตำแหน่งการเคลื่อนที่แบบก้าวหน้าไปตามเส้นทางที่กำหนดไว้ ซึ่งมีเส้นทางการเคลื่อนที่แน่นอน และวิธีที่สองกำหนดให้ปลายเครื่องมือเคลื่อนที่ไปตามแนวทางที่ได้กำหนดไว้เป็นลักษณะต่างๆ ซึ่งไม่เหมือนกับวิธีแรก โดยมีการเคลื่อนที่เป็นลักษณะ C-space และสอดคล้องกับข้อจำกัดที่มีอยู่ แล้วนำค่าที่ได้จากการทำงานของทั้งสองวิธีมาเปรียบเทียบความสัมพันธ์ของการปฏิบัติงานในแผนงานที่แตกต่างกัน

Saha and Isto (2006) ได้ศึกษาการวางแผนการเคลื่อนที่ของหุ่นยนต์หลายๆ ตัวมาปฏิบัติงานร่วมกัน โดยหุ่นยนต์ทุกตัวทำงานเป็นอิสระต่อกัน และมีใช้เวลาในการทำงานแต่ละงานไม่เท่ากัน ซึ่งจุดมุ่งหมายของการวิจัยในครั้งนี้เพื่อเปรียบเทียบวิธีการวางแผนการเคลื่อนที่ของหุ่นยนต์ที่ทำงานร่วมกันระหว่างแบบแยกกันทำงาน (Decouple) โดยมีเวลาการทำงานเสร็จที่ไม่แน่นอน และแบบเคลื่อนที่เข้าสู่จุดศูนย์กลาง (Centralize) ซึ่งมีเวลาการทำงานที่แน่นอน สำหรับคำตอบที่ได้แสดงให้เห็นว่าวิธีการวางแผนการเคลื่อนที่ทั้งสองแบบสามารถนำมาใช้งานได้ดี และมีประสิทธิภาพมากทั้งคู่

Alil *et al.* (2002) ได้ศึกษาการวางแผนเส้นทางที่ไม่สามารถทำงานได้ของการทำงานร่วมกันของแขนกล โดยการใช้วิธีการเชิงพันธุกรรมแบบค่อยเป็นค่อยไปร่วมกัน (CGA) เป็นการวางแผนการเคลื่อนที่แบบอัตโนมัติของหุ่นยนต์ผลิตที่ทำงานร่วมกัน โดยอาศัยแนวความคิดมาจากเทคนิคการกำหนดเส้นทางตามรูปร่างของชิ้นงาน (C-space) มาทำการกำหนดรูปร่างเส้นทางการเคลื่อนที่ด้วยวิธีการ CGA ซึ่งวิธีการดังกล่าวจะทำการหาเส้นทางที่สั้นที่สุดสำหรับการเคลื่อนที่ร่วมกันของแขนกล และไม่เกิดการชนกันระหว่างการเคลื่อนที่ของหุ่นยนต์ทุกตัว ซึ่งผลที่ได้จากการเคลื่อนที่ด้วยวิธีการสร้างเส้นทางการเคลื่อนที่ด้วย CGA นี้ได้นำมาเปรียบเทียบกับวิธีอื่นที่มีลักษณะเหมือนกัน พบว่าการสร้างเส้นทางด้วยวิธี CGA ดีกว่าวิธีอื่น

อุปกรณ์และวิธีการ

อุปกรณ์

อุปกรณ์ที่ใช้สำหรับการทำวิจัยในครั้งนี้ คือ

1. เครื่องคอมพิวเตอร์ 1 เครื่อง
2. เครื่องปริ้นเซอร์ 1 เครื่อง
3. โปรแกรมคอมพิวเตอร์สำเร็จรูป

วิธีการ

วิธีการดำเนินงานวิจัยประกอบด้วยขั้นตอนดังนี้ คือ ศึกษาค้นคว้างานวิจัยที่เกี่ยวข้องกับการวางแผนการดำเนินงาน และเคลื่อนที่ของหุ่นยนต์ที่ทำงานร่วมกัน เพื่อใช้เป็นแนวทางในการระบุปัญหา และกำหนดวัตถุประสงค์ จากนั้นทำการแปลงปัญหาให้อยู่ในรูปแบบกำหนดการเชิงเส้นเพื่อแสดงรูปแบบของปัญหา เมื่อได้รูปแบบของปัญหาและกำหนดการเชิงเส้นแล้วจึงทำการศึกษาวิธีการวางแผนการเคลื่อนที่ของหุ่นยนต์เพียงตัวเดียวเพื่อนำมาพัฒนาเป็นกระบวนการแก้ปัญหาเพื่อใช้เปิดคำตอบที่เหมาะสมสำหรับการกำหนดจำนวนช่วงการเคลื่อนที่ของหุ่นยนต์ที่ทำงานร่วมกัน

1. คุณลักษณะของปัญหา

ในงานวิจัยครั้งนี้ได้ทำการพัฒนาวิธีการวางแผนการเคลื่อนที่ของหุ่นยนต์ที่ทำงานร่วมกัน โดยขณะที่มีการเคลื่อนที่ได้มีการประกอบชิ้นงานที่เป็นทรงกระบอก และวงแหวนเข้าด้วยกัน ซึ่งรูปแบบของปัญหาได้ถูกแบ่งออกเป็น 2 กรณี คือ

1.1 ปัญหาที่มีวัตถุประสงค์เพื่อการสร้างเส้นทางที่สั้นที่สุดสำหรับการเคลื่อนย้ายชิ้นงานเพลาทรงกระบอก และวงแหวนไปยังตำแหน่งที่เตรียมพร้อมสำหรับการสวมประกอบชิ้นงานเพล

ทั้ง 2 ชั้นเข้าด้วยกัน โดยการเคลื่อนที่ในช่วงแรกนี้หุ่นยนต์หกแกน 2 ตัวที่ทำงานร่วมกันต้องใช้เวลาในการเคลื่อนที่ให้มีความน้อยที่สุด และระหว่างการเคลื่อนที่หุ่นยนต์ทั้ง 2 ตัวต้องไม่เกิดการชนกันของชิ้นงานเพลลา ซึ่งปัญหานี้ได้มีการกำหนดเป็นรูปแบบสมการเชิงเส้นเพื่อแสดงคุณลักษณะของปัญหา และอัลกอริทึมสำหรับหาเส้นทางที่เหมาะสมจะได้ถูกพัฒนาขึ้นเพื่อใช้ในการหาคำตอบ และยังสามารถใช้เป็นแนวทางในการพัฒนาโปรแกรมเพื่อแก้ปัญหาต่อไป

1.2 ปัญหาที่มีวัตถุประสงค์เพื่อหาเส้นทางเคลื่อนที่สำหรับการทำงานร่วมกันของหุ่นยนต์หกแกน 2 ตัวที่เหมาะสม ซึ่งได้มีการสวมประกอบชิ้นงานเพลลาเข้าด้วยกันขณะมีการเคลื่อนย้ายชิ้นงานเพลลาไปยังตำแหน่งเป้าหมายสุดท้าย โดยเส้นทางเคลื่อนที่ของหุ่นยนต์จะถูกกำหนดเป็นจำนวนตำแหน่งการเคลื่อนที่ของหุ่นยนต์ ดังนั้นจึงจำเป็นต้องคำนวณหาจำนวนตำแหน่งการเคลื่อนที่ให้มีความน้อยที่สุดโดยที่หุ่นยนต์สามารถทำงานได้ตามเป้าหมายที่กำหนดไว้ และไม่เกิดการชนกันของชิ้นงานเพลลาขณะเคลื่อนที่ และสวมประกอบชิ้นงานเพลลา โดยปัญหานี้ได้มีการกำหนดเป็นรูปแบบสมการเชิงเส้นเพื่อแสดงคุณลักษณะของปัญหา และอัลกอริทึมสำหรับหาเส้นทางที่เหมาะสมจะได้ถูกพัฒนาขึ้นเพื่อใช้ในการหาคำตอบ และใช้เป็นแนวทางสำหรับการพัฒนาวิธีการแก้ปัญหาให้มีประสิทธิภาพมากยิ่งขึ้นต่อไป

2. การกำหนดตำแหน่ง($\bar{P}$) หรือเส้นทางเคลื่อนที่ของหุ่นยนต์

สำหรับการศึกษาวิจัยในครั้งนี้มีการกำหนดตำแหน่งของการเคลื่อนที่ของหุ่นยนต์ทั้ง 2 ตัวไว้ 3 ตำแหน่ง คือ ตำแหน่งสำหรับจับชิ้นงาน (P_1) ตำแหน่งเตรียมพร้อมประกอบชิ้นงาน (P_2) และตำแหน่งนำชิ้นงานไปวางหลังจากประกอบเสร็จเรียบร้อยแล้ว (P_3) ซึ่งแสดงไว้ดังรูปภาพผนวกที่ 14 สามารถนำมาเขียนเป็นรูปแบบของเมทริกซ์ (Marix) ได้ดังนี้

กำหนดให้

$$P_1 = (X_1, Y_1, Z_1), P_2 = (X_2, Y_2, Z_2), \dots, P_n = (X_n, Y_n, Z_n)$$

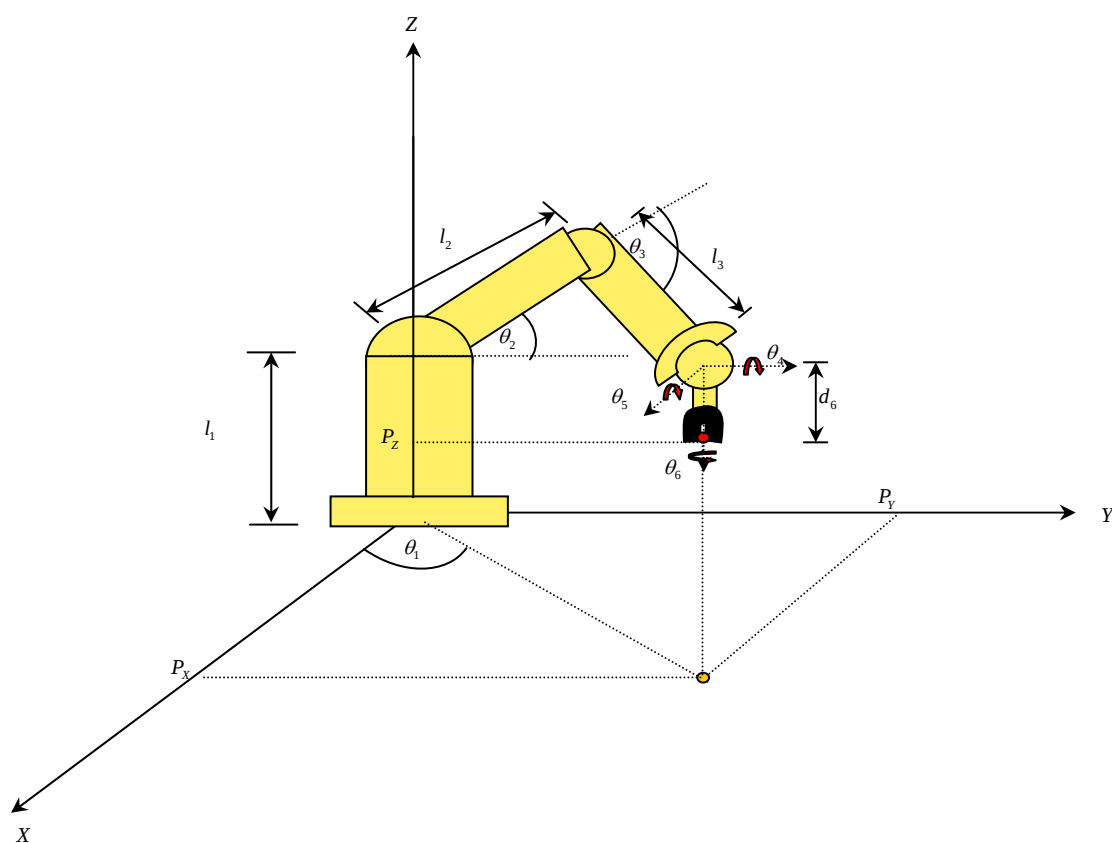
สามารถนำมาเขียนให้อยู่ในรูปของเมทริกซ์ได้ คือ

$$P = [P_1, P_2, P_3, \dots, P_n]$$

$$P^T = \begin{bmatrix} P_1 \\ P_2 \\ \dots \\ P_n \end{bmatrix} = \begin{bmatrix} X_1, Y_1, Z_1 \\ X_2, Y_2, Z_2 \\ \dots \\ X_n, Y_n, Z_n \end{bmatrix} \quad (44)$$

3. ศึกษาวิธีการคำนวณหาค่าฟอร์เวิร์ดคิเนเมติกส์ และอินเวิร์ดคิเนเมติกส์ สำหรับหุ่นยนต์ต้นแบบ

สำหรับการศึกษาการวางแผนเส้นทางการเคลื่อนของหุ่นยนต์ 6 แกน 2 ตัว มาทำงานร่วมกันนั้นหุ่นยนต์ต้นแบบที่นำมาใช้ในการศึกษาครั้งนี้เป็น Scara Robot ดังแสดงไว้ในภาพที่ 5 โดยมีวิธีการในการหาค่าฟอร์เวิร์ดคิเนเมติกส์ (Forward Kinematics) และอินเวิร์ดคิเนเมติกส์ (Inverse Kinematics) สำหรับการเคลื่อนที่ของหุ่นยนต์ดังนี้



ภาพที่ 5 ลักษณะข้อต่อ และค่าองศาการเคลื่อนที่ของหุ่นยนต์ scara 6 แกน

3.1 สมการฟอร์เวิร์ดคิเนมติกส์ (Forward Kinematics) เป็นวิธีการคำนวณหาตำแหน่งขณะเคลื่อนที่ของหุ่นยนต์ ซึ่งใช้หลักเกณฑ์ของ Denavit – Hartenberg (Saeed, 2001) มาคำนวณหาสมการฟอร์เวิร์ดคิเนมติกส์ได้ดังสมการที่ (45) ดังนี้

$$\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} d_6 c_1 s_5 c_{234} + d_6 s_1 c_5 + l_3 c_1 c_{23} + l_2 c_1 c_2 \\ d_6 s_1 s_5 c_{234} - d_6 c_1 c_5 + l_3 s_1 c_{23} + l_2 s_1 c_2 \\ d_6 s_5 s_{234} + l_3 s_{23} + l_2 s_2 + l_1 \end{bmatrix} \quad (45)$$

โดยที่

$$\begin{aligned} c_i &= \cos \theta_i && \text{สำหรับ } i=1,2,3,\dots,6 \\ s_i &= \sin \theta_i && \text{สำหรับ } i=1,2,3,\dots,6 \\ c_{ij} &= \cos(\theta_i + \theta_j) && \text{สำหรับ } i=1,2,3,\dots,6 \text{ และ } j=1,2,3,\dots,6 \\ s_{ij} &= \sin(\theta_i + \theta_j) && \text{สำหรับ } i=1,2,3,\dots,6 \text{ และ } j=1,2,3,\dots,6 \\ c_{ijk} &= \cos(\theta_i + \theta_j + \theta_k) && \text{สำหรับ } i=1,2,3,\dots,6, j=1,2,3,\dots,6 \text{ และ} \end{aligned}$$

$$k=1,2,3,\dots,6$$

$$s_{ijk} = \sin(\theta_i + \theta_j + \theta_k) \quad \text{สำหรับ } i=1,2,3,\dots,6, j=1,2,3,\dots,6 \text{ และ}$$

$$k=1,2,3,\dots,6$$

3.2 สมการอินเวอร์ดคิเนมติกส์ (Inverse Kinematics) เป็นวิธีการคำนวณหาองศาการเคลื่อนที่สำหรับแต่ละตำแหน่งของหุ่นยนต์ ซึ่งใช้หลักเกณฑ์ของ Denavit – Hartenberg (Saeed, 2001) มาคำนวณหาองศาการเคลื่อนที่ในแต่ละข้อต่อได้ดังนี้

$$D = \frac{X_C^2 + Y_C^2 - d_1^2 + (Z_C - l_1)^2 - l_2^2 - l_3^2}{2l_2 l_3} \quad (46)$$

$$\theta_1 = \tan^{-1}\left(\frac{Y_C}{X_C}\right) \quad (47)$$

$$\theta_2 = \tan^{-1}\left(\frac{Z_C}{\sqrt{(X_C^2 + Y_C^2 - d_0^2)}}\right) - \tan^{-1}\left(\frac{l_3 s_3}{l_2 + l_3 c_3}\right) \quad (48)$$

$$\theta_3 = \tan^{-1}\left(\frac{\sqrt{1-D^2}}{D}\right) \quad (49)$$

$$\theta_4 = \tan^{-1}\left(\frac{-c_1 s_{23} r_{13} - s_1 s_{23} r_{23} + c_{23} r_{33}}{c_1 c_{23} r_{13} + s_1 s_{23} r_{23} + s_{23} r_{33}}\right) \quad (50)$$

$$\theta_5 = \tan^{-1} \left(\frac{\sqrt{1 - (s_1 r_{13} - c_1 r_{23})^2}}{s_1 r_{13} - c_1 r_{23}} \right) \quad (51)$$

$$\theta_6 = \tan^{-1} \left(\frac{s_1 r_{12} - c_1 r_{23}}{-s_1 r_{11} + c_1 r_{21}} \right) \quad (52)$$

$$\text{โดยที่} \quad R_0^6 = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (53)$$

$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} = \begin{bmatrix} X_{Goal} - d_6 * r_{13} \\ Y_{Goal} - d_6 * r_{23} \\ Z_{Goal} - d_6 * r_{33} \end{bmatrix} \quad (54)$$

4. การแปลงปัญหาให้อยู่ในรูปกำหนดการเชิงเส้นสำหรับกรณีการเคลื่อนที่ของหุ่นยนต์ในช่วงแรก

การเคลื่อนที่ในช่วงแรกนี้หุ่นยนต์สามารถเคลื่อนที่ได้อย่างอิสระโดยใช้เวลาในการเคลื่อนที่น้อยสุด (Minimize Time) และไม่สนใจค่าความผิดพลาดของการเคลื่อนที่ แต่ต้องควบคุมให้หุ่นยนต์เคลื่อนที่โดยปราศจากการชนกันของชิ้นงานเพลาทั้งสอง ดังนั้นจำนวนช่วงการเคลื่อนที่ (p_1) จึงต้องสัมพันธ์กับจำนวนตำแหน่งการชนกันของชิ้นงานเพลาทั้งสองชิ้น สามารถนำมาเขียนเป็นรูปแบบกำหนดการเชิงเส้นได้ดังนี้

ฟังก์ชันวัตถุประสงค์

$$\text{Minimize } p_1$$

ข้อจำกัด

$$\text{Max}(T_1^A, T_1^B) \leq T_C \quad (1)$$

$$\text{Collision}(t) = 0 \quad \text{สำหรับ } t = 1, 2, 3, \dots, \text{Max}(T_1^A, T_1^B) \quad (2)$$

$$\text{Max}_{\forall j, \forall t} (AV_t^j) \leq AV_{\max} \quad \text{สำหรับ } t = 1, 2, 3, \dots, \text{Max}(T_1^A, T_1^B) \text{ และ } j = 1, 2, 3, \dots, 6 \quad (3)$$

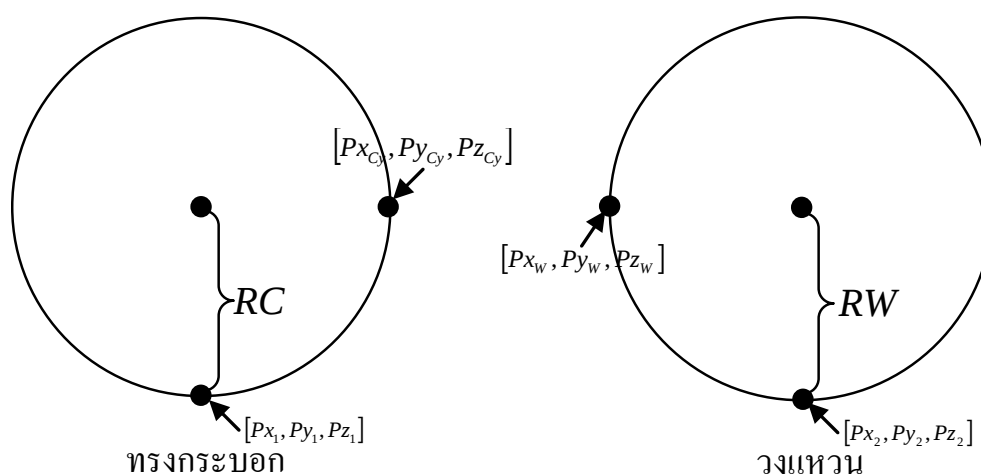
$$p_1 \geq 1 \quad (4)$$

$$T_1^A = T_1^B \quad (5)$$

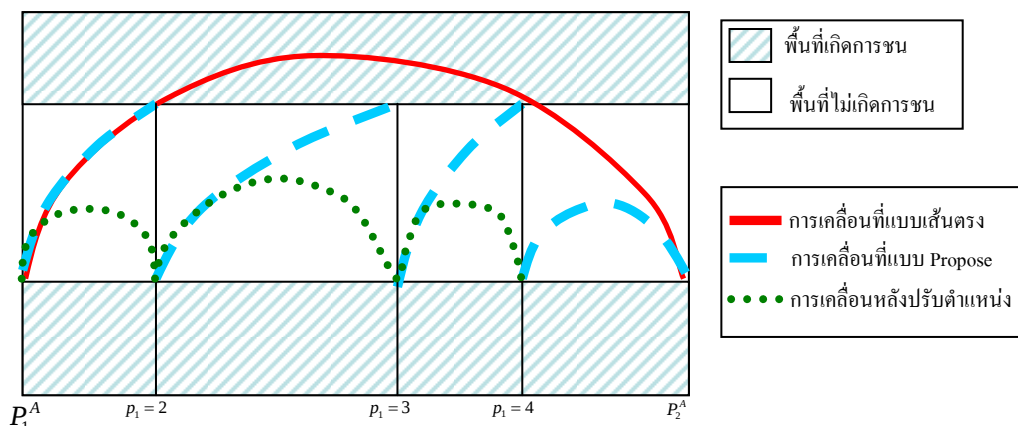
ฟังก์ชันวัตถุประสงค์แสดงถึงจำนวนช่วงสำหรับการเคลื่อนที่ของหุ่นยนต์ทั้ง 2 ตัวในช่วงแรก ข้อจำกัดที่ (1) แสดงถึงเวลาในการเคลื่อนที่ของหุ่นยนต์ทั้งสองตัวต้องมีค่าไม่เกินระยะเวลาในการเคลื่อนย้ายชิ้นงานที่กำหนดไว้ ข้อจำกัดที่ (2) แสดงถึงสถานะการชนของชิ้นงานขณะทำการเคลื่อนที่ในแต่ละช่วงเวลาต้องไม่เกิดการชนกันของชิ้นงาน ข้อจำกัดที่ (3) แสดงถึงความเร็วเชิงมุมของทุกๆ ข้อต่อในทุกช่วงเวลาต้องมีค่าไม่เกินความเร็วเชิงมุมสูงสุดที่หุ่นยนต์ทั้ง 2 ตัวสามารถเคลื่อนที่ได้ ข้อจำกัดที่ (4) แสดงถึงจำนวนช่วงสำหรับการเคลื่อนที่ในช่วงแรกต้องมีค่าน้อย 1 ช่วงหุ่นยนต์จึงสามารถทำงานได้ ข้อจำกัดที่ (5) แสดงถึงเวลาในการเคลื่อนที่ของหุ่นยนต์ A ต้องมีค่าใกล้เคียงหรือเท่ากับเวลาในการเคลื่อนที่ของหุ่นยนต์ B

5. อัลกอริทึมสำหรับการคำนวณหาเส้นทางการเคลื่อนที่ของหุ่นยนต์ในช่วงแรก โดยมีจำนวนตำแหน่งการเคลื่อนที่น้อยที่สุด

สำหรับการคำนวณหาจำนวนช่วงการเคลื่อนที่ในช่วงแรกของการเคลื่อนที่ ซึ่งหุ่นยนต์ต้องใช้เวลาในการเคลื่อนที่น้อยที่สุด โดยวิธีการคำนวณหาจำนวนตำแหน่งการเคลื่อนที่จะกำหนดตามจำนวนตำแหน่งที่เกิดการชนกันของชิ้นงานทั้งสองชิ้น เมื่อปล่อยให้หุ่นยนต์ทั้งสองตัวมีการเคลื่อนที่อย่างอิสระจากตำแหน่งเริ่มต้นไปจนถึงตำแหน่งเป้าหมายซึ่งเป็นตำแหน่งที่ต้องการให้ชิ้นงานเพลาวงแหวน และทรงกระบอกเตรียมพร้อมที่จะสวมประกอบกันในการเคลื่อนที่ช่วงที่ 2 โดยอัลกอริทึม (Algorithm) สำหรับคำนวณหาจำนวนตำแหน่งการเคลื่อนที่ (p_1) ในช่วงแรกมีขั้นตอนการทำงานได้แสดงไว้ในรูปภาพที่ 6 รูปภาพที่ 7 ภาพผนวกที่ 1 และ 2 ซึ่งมีรายละเอียดดังนี้



ภาพที่ 6 แสดงตำแหน่งบนชิ้นงานเพลาวงแหวนที่อ้างอิงการทำงานของปลายเครื่องมือหุ่นยนต์



ภาพที่ 7 แสดงตำแหน่งการเคลื่อนที่สำหรับการวางแผนเส้นทางการเคลื่อนที่ช่วงที่ 1

ขั้นตอนที่ 1: รับข้อมูลตำแหน่งการเคลื่อนที่ของแขนกล และรายละเอียดของเพลานำมาใช้ในการสวมประกอบสำหรับการเคลื่อนที่ของแขนกลในช่วงแรก โดยมีรายละเอียดดังนี้

1. ตำแหน่งการเคลื่อนที่ของหุ่นยนต์ $[P^A]^T = [P_1^A, P_2^A]$ และ $[P^B]^T = [P_1^B, P_2^B]$
2. ลักษณะของชิ้นงานเพลาทรงกระบอก และวงแหวน $[L_C, R_C, R_W, L_W]$
3. ข้อมูลข้อกำหนดสำหรับการเคลื่อนที่ของหุ่นยนต์ $[Acc, AV, T_C, L, d_6]$

ขั้นตอนที่ 2: คำนวณค่าพิกัดความคลาดเคลื่อนของชิ้นงานเพลานำ (R_{dif}) ค่าระยะห่างของหุ่นยนต์ A และ B (D_{AB}) และระยะห่างของปลายเครื่องมือบนระนาบแกน X ที่ไม่ก่อให้เกิดการชนของเพลานำ (D_X) สำหรับการเคลื่อนที่ของหุ่นยนต์ A และ B โดยสามารถคำนวณค่าต่างๆ ดังนี้

1. $R_{dif} = R_W - R_C$
2. $D_{AB} = |Y_2^A - Y_2^B|$
3. $D_X = |X_2^A + X_2^B|$
4. กำหนดให้ระบบการตรวจสอบทำงาน ($System = 1$) จำนวนช่วงการเคลื่อนที่ก่อนมี

การตรวจสอบมีค่าเท่ากับ 1 ($p_1 = 1$) และทำต่อในขั้นตอนที่ 3

ขั้นตอนที่ 3: ตรวจสอบสถานะการทำงานของระบบตรวจสอบการชนของชิ้นงานเพลานำทั้งสอง

1. ถ้าระบบการตรวจสอบทำงานอยู่ (system=1) ให้ทำต่อในขั้นตอนที่ 4
2. ถ้าระบบการตรวจสอบไม่ทำงาน (system=0) แสดงว่าระบบการตรวจสอบการชนของชิ้นงานเพลตสิ้นสุดลง หรือไม่พบตำแหน่งที่เกิดการชนกันของปลายเครื่องมือ หรือชิ้นงานเพลตทั้งสองขณะเคลื่อนที่

ขั้นตอนที่ 4: คำนวณองศาการเคลื่อนที่ เวลาการเคลื่อนที่ของหุ่นยนต์ และตำแหน่งการเคลื่อนที่ ณ เวลา t ใดๆ ของปลายเครื่องมือหุ่นยนต์ทั้งสองตัว โดยมีขั้นตอนการคำนวณดังนี้

1. องศาการเคลื่อนที่ $[\theta_1^{A1}, \theta_2^{A1}, \theta_3^{A1}, \theta_4^{A1}, \theta_5^{A1}, \theta_6^{A1}] = \text{InvKine}(X^A, Y^A, Z^A, R, L, d_6)$, $[\theta_1^{B1}, \theta_2^{B1}, \theta_3^{B1}, \theta_4^{B1}, \theta_5^{B1}, \theta_6^{B1}] = \text{InvKine}(X^B, Y^B, Z^B, R, L, d_6)$ สำหรับหุ่นยนต์ A และ B
2. เวลาการเคลื่อนที่สำหรับหุ่นยนต์ A คือ $T_1^A = \text{Cal_Time}(\theta_1^{A1}, \theta_2^{A1}, \theta_3^{A1}, \theta_4^{A1}, \theta_5^{A1}, \theta_6^{A1}, \text{Acc})$
3. เวลาการเคลื่อนที่สำหรับหุ่นยนต์ B คือ $T_1^B = \text{Cal_Time}(\theta_1^{B1}, \theta_2^{B1}, \theta_3^{B1}, \theta_4^{B1}, \theta_5^{B1}, \theta_6^{B1}, \text{Acc})$
4. องศาการเคลื่อนที่ ณ เวลา t ใดๆ สำหรับหุ่นยนต์ A และ B คือ $[\hat{\theta}_1^{A1}, \hat{\theta}_2^{A1}, \hat{\theta}_3^{A1}, \hat{\theta}_4^{A1}, \hat{\theta}_5^{A1}, \hat{\theta}_6^{A1}] = \text{Forecast_Angle}(\theta_1^{A1}, \theta_2^{A1}, \theta_3^{A1}, \theta_4^{A1}, \theta_5^{A1}, \theta_6^{A1}, \text{Acc}, AV_{\max})$ และ $[\hat{\theta}_1^{B1}, \hat{\theta}_2^{B1}, \hat{\theta}_3^{B1}, \hat{\theta}_4^{B1}, \hat{\theta}_5^{B1}, \hat{\theta}_6^{B1}] = \text{Forecast_Angle}(\theta_1^{B1}, \theta_2^{B1}, \theta_3^{B1}, \theta_4^{B1}, \theta_5^{B1}, \theta_6^{B1}, \text{Acc}, AV_{\max})$
5. ตำแหน่งการเคลื่อนที่ของหุ่นยนต์สำหรับหุ่นยนต์ A และ B คือ $(P_X^{A1}, P_Y^{A1}, P_Z^{A1}) = \text{Forward_Kinematics}(\hat{\theta}_1^{A1}, \hat{\theta}_2^{A1}, \hat{\theta}_3^{A1}, \hat{\theta}_4^{A1}, \hat{\theta}_5^{A1}, \hat{\theta}_6^{A1}, L, d_6)$ และ $(P_X^{B1}, P_Y^{B1}, P_Z^{B1}) = \text{Forward_Kinematics}(\hat{\theta}_1^{B1}, \hat{\theta}_2^{B1}, \hat{\theta}_3^{B1}, \hat{\theta}_4^{B1}, \hat{\theta}_5^{B1}, \hat{\theta}_6^{B1}, L, d_6)$
6. ระยะห่างน้อยที่สุดบนระนาบแกน Y ที่ไม่ทำให้ชิ้นงานเพลตชนกัน คือ $D_Y = R_C + R_W$
7. เวลาการเคลื่อนที่ของหุ่นยนต์ในช่วงที่ 1 คือ $T_1^{AB} = \text{Max}(T_1^A, T_1^B)$
8. กำหนดเวลาเริ่มต้นการตรวจสอบเท่ากับ 1 ($t_1 = 1$) สถานะการชนของหุ่นยนต์เท่ากับ ศูนย์ ($\text{Collision} = 0$) และระยะห่างของปลายเครื่องมือของหุ่นยนต์ A และ B มีค่าเท่ากับศูนย์ ($D_X^{AB} = 0$) เสร็จแล้วให้ทำต่อในขั้นตอนที่ 5

ขั้นตอนที่ 5: ทำการตรวจสอบตำแหน่ง ณ. ช่วงเวลา t_1 ใดๆ ว่าครบทุกช่วงเวลา (T_1^{AB}) หรือไม่ นอกจากนี้ยังทำการตรวจสอบสถานะการชนของชิ้นงานเพลตขณะเคลื่อนที่ว่าเกิดการชนหรือไม่

1. ถ้า ($t_1 \leq T_1^{AB}$) และ ($\text{Collision}=0$) แสดงว่าเวลาสำหรับการตรวจสอบยังไม่ครบ และยังไม่พบตำแหน่งที่เกิดการชน ดังนั้นให้ตรวจสอบระยะการชนบนระนาบแกน X ต่อไปโดยทำต่อขั้นตอนที่ 7

2. ถ้า $(t_1 > T_1^{AB})$ หรือ (Collision=1) แสดงว่ามีการตรวจสอบได้สิ้นสุดลง เนื่องจากได้มีการตรวจสอบครบทุกตำแหน่งแล้ว หรือในการตรวจสอบได้พบตำแหน่งที่เกิดการชนของชิ้นงานเพลานี้ทำให้ต้องมีการตรวจสอบข้อมูลอื่นๆ เพิ่มเติมในขั้นตอนที่ 6 ต่อไป

ขั้นตอนที่ 6: ทำการตรวจสอบตำแหน่ง ณ. ช่วงเวลา t_1 ใดๆ ว่าครบทุกช่วงเวลา (T_1^{AB}) หรือไม่

1. ถ้าช่วงเวลสำหรับการตรวจสอบมากกว่าเวลาการเคลื่อนที่ของหุ่นยนต์ ($t_1 > T_1^{AB}$) กำหนดให้สถานะการตรวจสอบเท่ากับศูนย์ หรือสิ้นสุดการตรวจสอบ (System = 0)
2. ถ้าช่วงเวลสำหรับการตรวจสอบน้อยกว่าหรือเท่ากับเวลาการเคลื่อนที่ของหุ่นยนต์ ($t_1 \leq T_1^{AB}$) กำหนดให้ระบบการตรวจสอบยังทำงานอยู่ (System = 1)
3. กลับไปทำขั้นตอนที่ 3 ใหม่

ขั้นตอนที่ 7: ทำการตรวจสอบระยะของปลายเครื่องมือหุ่นยนต์ ($D_{X_{t_1}}^{AB}$) ณ. ช่วงเวลา t_1 ใดๆ กับระยะการชนของชิ้นงานเพลทั้งสองบนระนาบแกน X (D_X) ว่าสามารถเกิดการชนได้หรือไม่

1. ถ้า $(D_{X_{t_1}}^{AB} \leq D_X)$ แสดงว่าตำแหน่งการเคลื่อนที่ของหุ่นยนต์สามารถเกิดการชนของชิ้นงานเพลทั้งสองได้ ดังนั้นให้กำหนดหาตำแหน่งของเพลทั้งสองต่อไปโดยทำต่อในขั้นตอนที่ 8
2. ถ้า $(D_{X_{t_1}}^{AB} > D_X)$ แสดงว่าตำแหน่งการเคลื่อนที่ของหุ่นยนต์ไม่สามารถเกิดการชนของเพลทั้งสองได้ ดังนั้นให้ทำการตรวจสอบตำแหน่งการเคลื่อนที่ตำแหน่งอื่นๆ ต่อไป โดยกำหนดให้ $t_1 = t_1 + 1, D_{X_{t_1}}^{AB} = |P_{X_{t_1}}^{A1} + P_{X_{t_1}}^{B1}|$, (Collision = 0) และกลับไปทำขั้นตอนที่ 5 ใหม่

ขั้นตอนที่ 8: กำหนดหาตำแหน่งการเคลื่อนที่ของชิ้นงานเพลทั้งสอง และระยะห่างของปลายเครื่องมือบนระนาบแกน Z ขณะเคลื่อนที่ ณ. ช่วงเวลา t_1 ใดๆ ($D_{Z_{t_1}}^{AB}$) โดยการคำนวณสามารถทำได้ดังนี้

$$\begin{aligned}
 &1. \text{ ตำแหน่งของชิ้นงานเพลาทรงกระบอก คือ } \begin{bmatrix} P_{X_{t_1}}^{Cy} \\ P_{Y_{t_1}}^{Cy} \\ P_{Z_{t_1}}^{Cy} \end{bmatrix} = \begin{bmatrix} P_{X_{t_1}}^{B1} + 0.5L_C \\ D_{AB} - \{P_{Y_{t_1}}^{B1} + R_C\} \\ P_{Z_{t_1}}^{B1} + R_C \end{bmatrix} \\
 &2. \text{ ตำแหน่งของชิ้นงานเพลาวงแหวน คือ } \begin{bmatrix} P_{X_{t_1}}^W \\ P_{Y_{t_1}}^W \\ P_{Z_{t_1}}^W \end{bmatrix} = \begin{bmatrix} P_{X_{t_1}}^{A1} + 0.5L_W \\ P_{Y_{t_1}}^{A1} + R_W \\ P_{Z_{t_1}}^{A1} + R_W \end{bmatrix}
 \end{aligned}$$

3. ระยะห่างของชิ้นงานเพลานบนระนาบแกน Z ขณะเคลื่อนที่ $D_{Z_{t_1}}^{AB} = |P_{Z_{t_1}}^{A1} - P_{Z_{t_1}}^{B1}|$

4. ทำต่อในขั้นตอนที่ 9

ขั้นตอนที่ 9: ตรวจสอบการชนกันของชิ้นงานเพลาระยะเคลื่อนที่บนระนาบแกน Y ณ. ช่วงเวลา t_1 ใดๆ ว่าสามารถเกิดการชนกันได้หรือไม่

1. ถ้า $\{(P_{Y_{t_1}}^{Cy} - P_{Y_{t_1}}^W) > 0\}$ แสดงว่าไม่เกิดการชนกันของชิ้นงานเพลาทิ้งสอง ดังนั้นจึงกำหนดให้

1.1 $Collision = 0$

1.2 $t_1 = t_1 + 1$

1.3 ทำการตรวจสอบตำแหน่งการเคลื่อนที่ของหุ่นยนต์ทั้งสองว่าถึงตำแหน่งเป้าหมายหรือไม่ โดยทำต่อในขั้นตอนที่ 10

2. ถ้า $\{(P_{Y_{t_1}}^{Cy} - P_{Y_{t_1}}^W) \leq 0\}$ แสดงว่าชิ้นงานเพลาทิ้งสองอยู่ในตำแหน่งที่สามารถเกิดการชนกันได้ ดังนั้นจึงให้ทำการตรวจสอบการชนต่อบนระนาบแกน Z ต่อไปโดยทำในขั้นตอนที่ 11

ขั้นตอนที่ 10: ตรวจสอบตำแหน่งการเคลื่อนที่ของหุ่นยนต์ว่าถึงตำแหน่งพร้อมประกอบชิ้นงานเพลาทิ้งสองหรือไม่

1. ถ้า $(t_1 \leq T_1^{AB})$ แสดงว่าหุ่นยนต์ยังเคลื่อนที่ไม่ถึงตำแหน่งเตรียมพร้อมสวมประกอบ ดังนั้นกำหนดให้ $D_{X_{t_1}}^{AB} = |P_{X_{t_1}}^{A1} + P_{X_{t_1}}^{B1}|$

2. ถ้า $(t_1 > T_1^{AB})$ แสดงว่าหุ่นยนต์เคลื่อนที่ถึงตำแหน่งเตรียมพร้อมสวมประกอบแล้ว ดังนั้นกำหนดให้ $Collision = 1$

3. กลับไปขั้นตอนที่ 5 ใหม่

ขั้นตอนที่ 11: ทำการตรวจสอบการชนกันของชิ้นงานเพลาทิ้งสองบนระนาบแกน z ว่าสามารถเกิดการชนกันของชิ้นงานเพลาดได้หรือไม่ โดยสามารถทำการตรวจสอบได้ดังนี้

1. ถ้า $\{P_{Y_{t_1}}^{Cy} - P_{Y_{t_1}}^W \leq 0\}$ และ $\{D_{Z_{t_1}}^{AB} < (2 \times R_w)\}$ แสดงว่าชั้นงานเพลาทั้งสองเกิดการชนกัน ดังนั้นให้ระบุตำแหน่งการชนของชั้นงานเพลาดังนี้

1.1 ระบุสถานะการชนเท่ากับ 1 ($Collision = 1$)

1.2 กำหนดจำนวนตำแหน่งการเคลื่อนที่เพิ่ม 1 ตำแหน่ง ($p_1 = p_1 + 1$)

1.3 ระบุตำแหน่งการชนของหุ่นยนต์ A คือ $C_{X_{p_1}}^{A1} = P_{X_{t_1}}^{A1}$, $C_{Y_{p_1}}^{A1} = P_{Y_{t_1}}^{A1}$

และ $C_{Z_{p_1}}^{A1} = P_{Z_{t_1}}^{A1}$

1.4 ระบุตำแหน่งการชนของหุ่นยนต์ B คือ $C_{X_{p_1}}^{B1} = P_{X_{t_1}}^{B1}$, $C_{Y_{p_1}}^{B1} = P_{Y_{t_1}}^{B1}$ และ

$C_{Z_{p_1}}^{B1} = P_{Z_{t_1}}^{B1}$

1.5 กำหนดให้ $n_A = length(X^A)$ และ $n_B = length(X^B)$

1.6 กำหนดให้ $i = 1$, $j = 1$ และให้ทำต่อขั้นตอนที่ 12

2. ถ้า $\{P_{Y_{t_1}}^{Cy} - P_{Y_{t_1}}^W > 0\}$ หรือ $\{D_{Z_{t_1}}^{AB} > (2 \times R_w)\}$ แสดงว่าชั้นงานเพลาทั้งสองไม่เกิดการชนกัน ดังนั้นให้กำหนดค่าต่างๆ ดังนี้

2.1 ระบุสถานะการชนเท่ากับศูนย์ ($Collision = 0$)

2.2 เพิ่มตำแหน่งในการตรวจสอบ $t_1 = t_1 + 1$

2.3 หาระยะการเคลื่อนที่บนระนาบแกน X คือ $D_{X_{t_1}}^{AB} = |P_{X_{t_1}}^{A1} + P_{X_{t_1}}^{B1}|$

2.4 ทำต่อขั้นตอนที่ 13

ขั้นตอนที่ 12: เพิ่มตำแหน่งการเคลื่อนที่ของหุ่นยนต์ โดยเพิ่มขึ้น 1 ตำแหน่ง ซึ่งมีขั้นตอนการเพิ่มดังนี้

1. ถ้า $[i \leq \min(n_A, n_B) + 1]$ ให้ทำต่อในขั้นตอนที่ 11

2. ถ้า $[i > \min(n_A, n_B) + 1]$ ให้กำหนดค่าต่างๆ ดังนี้

2.1 $X^{A1} = NX^{A1}$, $Y^{A1} = NY^{A1}$, $Z^{A1} = NZ^{A1}$

2.2 $X^{B1} = NX^{B1}$, $Y^{B1} = NY^{B1}$, $Z^{B1} = NZ^{B1}$

2.3 กลับไปทำขั้นตอนที่ 5 ใหม่

ขั้นตอนที่ 13: ตรวจสอบค่า j ได้ดังต่อไปนี้

1. ถ้า $[j < \min(n_A, n_B)]$ กำหนดให้

$$1.1 \quad NX_j^{A1} = X_j^{A1}, NY_j^{A1} = Y_j^{A1}, NZ_j^{A1} = Z_j^{A1}$$

$$1.2 \quad NX_j^{B1} = X_j^{B1}, NY_j^{B1} = Y_j^{B1}, NZ_j^{B1} = Z_j^{B1}$$

$$1.3 \quad i = i + 1 \text{ และ } j = j + 1$$

1.4 กลับไปทำขั้นตอนที่ 12

2. ถ้า $[j = \min(n_A, n_B)]$ กำหนดให้

$$2.1 \quad NX_j^{A1} = X_{t_1}^{A1}, NY_j^{A1} = (Y_{t_1}^{A1} - R_W), NZ_j^{A1} = Z_{t_1}^{A1}$$

$$2.2 \quad NX_j^{B1} = X_{t_1}^{B1}, NY_j^{B1} = Y_{t_1}^{B1} - R_C, NZ_j^{B1} = Z_{t_1}^{B1}$$

$$2.3 \quad i = i + 1 \text{ และ } j = j + 1$$

2.4 กลับไปทำขั้นตอนที่ 12

3. ถ้า $[j > \min(n_A, n_B)]$ กำหนดให้

$$3.1 \quad n_A = \text{length}(X^A), n_B = \text{length}(X^B)$$

$$3.2 \quad NX_j^{A1} = X_{n_A}^{A1}, NY_j^{A1} = Y_{n_A}^{A1}, NZ_j^{A1} = Z_{n_A}^{A1}$$

$$3.3 \quad NX_j^{B1} = X_{n_B}^{B1}, NY_j^{B1} = Y_{n_B}^{B1}, NZ_j^{B1} = Z_{n_B}^{B1}$$

$$3.4 \quad i = i + 1 \text{ และ } j = j + 1$$

3.5 กลับไปทำขั้นตอนที่ 12

ขั้นตอนที่ 14: ตรวจสอบเส้นทางการเคลื่อนที่ถึงตำแหน่งเป้าหมายหรือไม่

1. ถ้า $(t_1 \leq T_1^{AB})$ กำหนดให้ $D_{X_{t_1}}^{AB} = |P_{X_{t_1}}^{A1} + P_{X_{t_1}}^{B1}|$

2. ถ้า $(t_1 > T_1^{AB})$ กำหนดให้ $\text{Collision} = 1$

3. กลับไปทำขั้นตอนที่ 5

6. การแปลงปัญหาเป็นรูปกำหนดการเชิงเส้นสำหรับกรณีการเคลื่อนที่ของหุ่นยนต์ในช่วงสุดท้าย

การเคลื่อนที่ในช่วงสุดท้ายเป็นการเคลื่อนย้ายชิ้นงานเพลลาจากตำแหน่งสุดท้ายของการเคลื่อนที่ในช่วงที่ 1 ไปยังตำแหน่งสุดท้ายสำหรับนำชิ้นงานเพลลาไปวาง โดยมีการสวมประกอบชิ้นงานเพลลาไปพร้อมๆ กัน ซึ่งการเคลื่อนที่ในช่วงนี้เน้นที่ความรวดเร็วในการเคลื่อนที่ และทำการตรวจสอบค่าระยะการแกว่งของชิ้นงานเพลลาทั้งสอง โดยต้องมีค่าไม่เกินค่าพิสัยความคลาดเคลื่อนของชิ้นงานเพลลา ซึ่งเวลาการเคลื่อนที่ของหุ่นยนต์ในช่วงสุดท้ายนี้ต้องมีค่าเท่ากัน หรือใกล้เคียงกัน จึงสามารถสวมประกอบชิ้นงานขณะเคลื่อนที่ได้ สามารถนำมาเขียนเป็นรูปแบบกำหนดการเชิงเส้นเพื่อหาค่าจำนวนช่วงการเคลื่อนที่ (p_2) ได้ดังนี้

ฟังก์ชันวัตถุประสงค์

$$\text{Minimize } p_2$$

ข้อจำกัด

$$\sum_{i=1}^{p_2} \sum_{j=1}^n t_{ij} \leq T_R^{AB} \quad (1)$$

$$\text{Max}_{\forall t} \left\{ |P_{V_t}^A - P_{V_t}^B|, |P_{Z_t}^A - P_{Z_t}^B| \right\} < |R_W - R_C| \quad \text{สำหรับ } t = 1, 2, 3, \dots, \text{Max}(T_2^A, T_2^B) \quad (2)$$

$$|P_{X_t}^A - P_{X_t}^B| \leq (X_2^A + X_2^B) \quad \text{สำหรับ } t = 1, 2, 3, \dots, \text{Max}(T_2^A, T_2^B) \quad (3)$$

$$\text{Max}_{\forall j, \forall t} (AV_t^j) \leq AV_{\max} \quad \text{สำหรับ } t = 1, 2, 3, \dots, \text{Max}(T_2^A, T_2^B) \text{ และ } j = 1, 2, 3, \dots, 6 \quad (4)$$

$$p_2 \geq 1 \quad (5)$$

$$|R_W - R_C| \geq 0 \quad (6)$$

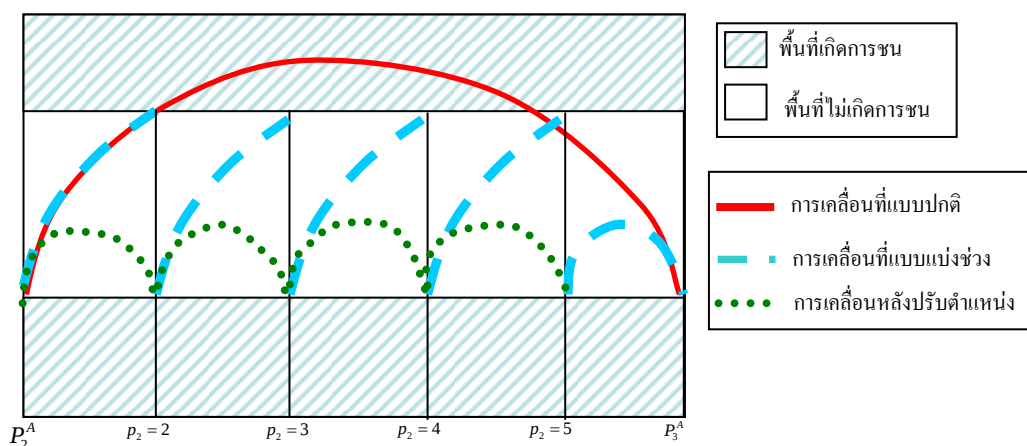
$$T_2^A = T_2^B \quad (7)$$

ฟังก์ชันวัตถุประสงค์แสดงถึงจำนวนช่วงสำหรับการเคลื่อนที่ของหุ่นยนต์ทั้ง 2 ตัวในช่วงสุดท้าย ข้อจำกัดที่ (1) แสดงถึงเวลาในการเคลื่อนที่ของหุ่นยนต์ทั้งสองตัวต้องมีค่าไม่เกินระยะเวลาที่เหลือจากการเคลื่อนย้ายชิ้นงานในช่วงแรก ข้อจำกัดที่ (2) แสดงถึงระยะค่าพิสัยความคลาดเคลื่อนสูงสุดในทุกช่วง เวลาบนระนาบแกน Y และแกน Z ต้องมีค่าน้อยกว่าค่าพิสัยความคลาดเคลื่อนระหว่างชิ้นงานเพลลาทรงกระบอก และวงแหวน ข้อจำกัดที่ (3) แสดงถึงระยะห่างของปลาย

เครื่องมือขณะเคลื่อนย้ายชิ้นงานของหุ่นยนต์ทั้งสองตัวในแต่ละช่วงเวลาต้องมีค่าไม่เกินระยะห่างของปลายเครื่องมือในตำแหน่งสุดท้ายของการเคลื่อนที่ในช่วงแรก ข้อจำกัดที่ (4) แสดงถึงความเร็วเชิงมุมของทุกๆ ข้อต่อในทุกช่วงเวลาต้องมีค่าไม่เกินความเร็วเชิงมุมสูงสุดที่หุ่นยนต์ทั้งสองตัวสามารถเคลื่อนที่ได้ ข้อจำกัดที่ (5) แสดงถึงจำนวนช่วงสำหรับการเคลื่อนที่ในช่วงสุดท้ายต้องมีค่าอย่างน้อย 1 ช่วงหุ่นยนต์จึงสามารถทำงานได้ ข้อจำกัดที่ (6) แสดงถึงค่าฟังก์ชันความคลาดเคลื่อนระหว่างชิ้นงานเพลาทรงกระบอก และวงแหวนต้องมีค่าน้อยเท่ากับศูนย์ ข้อจำกัดที่ (7) แสดงถึงเวลาในการเคลื่อนที่ของหุ่นยนต์ A ต้องมีค่าใกล้เคียงหรือเท่ากับเวลาในการเคลื่อนที่ของหุ่นยนต์ B

7. อัลกอริทึมสำหรับคำนวณหาเส้นทางการเคลื่อนที่ของหุ่นยนต์ในช่วงสุดท้าย (p_2) โดยมีจำนวนตำแหน่งการเคลื่อนที่น้อยที่สุด และสามารถประกอบชิ้นงานขณะเคลื่อนที่ได้

สำหรับการคำนวณหาจำนวนตำแหน่งการเคลื่อนที่ในช่วงสุดท้าย ซึ่งหุ่นยนต์ต้องสามารถประกอบชิ้นงานขณะเคลื่อนที่ได้ โดยทำการตรวจสอบจากค่าระยะห่างของปลายเครื่องมือ และตำแหน่งการเคลื่อนที่ของชิ้นงานให้มีระยะของการแกว่งของชิ้นงานอยู่ในระดับที่สามารถประกอบชิ้นงานได้ ซึ่งมีค่าระยะการแกว่งไม่เกินเส้นผ่านศูนย์กลางของชิ้นงานที่มีขนาดใหญ่ที่สุด โดยอัลกอริทึม (Algorithm) สำหรับคำนวณหาจำนวนตำแหน่งการเคลื่อนที่ (p_2) ในช่วงสุดท้ายมีขั้นตอนการทำงานซึ่งได้แสดงไว้ในรูปภาพที่ 8 ภาพผนวกที่ 3 และภาพผนวกที่ 4 โดยมีรายละเอียดดังนี้



ภาพที่ 8 แสดงตำแหน่งการเคลื่อนที่สำหรับการวางแผนเส้นทางการเคลื่อนที่ช่วงที่ 2

ขั้นตอนที่ 1: รับข้อมูลตำแหน่งการเคลื่อนที่ของแขนกล และรายละเอียดของเพลานำมาใช้ในการ
รวมประกอบสำหรับการเคลื่อนที่ของแขนกลในช่วงแรก โดยมีรายละเอียดดังนี้

1. ตำแหน่งการเคลื่อนที่ของหุ่นยนต์ $[P^A]^T = [P_2^A, P_3^A]$ และ $[P^B]^T = [P_2^B, P_3^B]$
2. ลักษณะชิ้นงานเพลาทรงกระบอก และวงแหวน คือ $[L_C, R_C, R_W, L_W]$
3. ข้อจำกัดสำหรับการเคลื่อนที่ของหุ่นยนต์ คือ $[Acc, AV, T_C, L, d_6]$

ขั้นตอนที่ 2: คำนวณข้อมูลเบื้องต้นสำหรับคำนวณหาจำนวนตำแหน่งการเคลื่อนที่ของหุ่นยนต์
โดยข้อมูลที่ควรทำการคำนวณ มีดังนี้

1. เวลาการเคลื่อนที่สำหรับหุ่นยนต์ A ในช่วงที่สอง คือ $T_R^A = T_C - T_1^A$
2. เวลาการเคลื่อนที่สำหรับหุ่นยนต์ B ในช่วงที่สอง $T_R^B = T_C - T_1^B$
3. เวลาการเคลื่อนที่สำหรับหุ่นยนต์ A และ B ในช่วงที่สอง $T_R^{AB} = \text{Min}(T_R^{A1}, T_R^{B1})$
4. ค่าฟังก์ชันความคลาดเคลื่อนเริ่มต้นก่อนการเคลื่อนที่ $Error_{R_{\max}} = (2 * R_{dif})$
5. ระยะของปลายเครื่องมือสูงสุดบนระนาบแกน X คือ $D_X = |X_1^{A2} + X_1^{B2}|$
6. $r = 1, Error_{\min} = Inf$

ขั้นตอนที่ 3: ตรวจสอบค่า $Error_{R_{\max}}$ และ T_R^{AB} อยู่ในค่าที่ยอมรับได้หรือไม่

1. ถ้า $(Error_{R_{\max}} > R_{dif})$ และ $(T_{Opt}^{AB} \leq T_R^{AB})$ แสดงว่าอัตราการแกว่งของปลายเครื่องมือมี
ค่าสูงกว่าค่าฟังก์ชันความคลาดเคลื่อนของชิ้นงาน และเวลาการเคลื่อนที่ยังไม่เกินเวลาที่กำหนดไว้
โดยกำหนดค่าต่างๆ ดังนี้

- 1.1 $r = r + 1$
- 1.2 $System = 1$
- 1.3 $k = 0$
- 1.4 $r_2 = r$
- 1.5 $Error_{Old} = 0$
- 1.6 ทำต่อขั้นตอนที่ 4

2. ถ้า $(Error_{R_{max}} \leq R_{dif})$ หรือ $(T_{Opt}^{AB} > T_R^{AB})$ แสดงว่าอัตราการแกว่งของปลายเครื่องมือมีค่าไม่เกินค่าพิสัยความคลาดเคลื่อนของชิ้นงาน หรือเวลาการเคลื่อนที่มากกว่าเวลาที่กำหนดไว้ โดยให้ทำต่อในขั้นตอนที่ 9

ขั้นตอนที่ 4: ทำการตรวจสอบค่า k และค่า $Error_{R_{max}}$ ใหม่ เมื่อกำหนดจำนวนตำแหน่งของหุ่นยนต์ B เพิ่มขึ้น โดยสามารถทำได้ดังนี้

1. ถ้า $(k \leq 20)$ และ $(Error_{R_{max}} \geq R_{dif})$ และ $(System = 1)$ กำหนดจำนวนตำแหน่งการเคลื่อนที่ของหุ่นยนต์ B ใหม่ ได้ดังนี้

1.1 ถ้า $(k = 0)$ ให้ $r_2 = r$

1.2 ถ้า $(k \neq 0)$ ให้ $r_2 = r_2 + 1$

1.3 ทำต่อขั้นตอนที่ 5

2. ถ้า $(k > 20)$ หรือ $(Error_{R_{max}} < R_{dif})$ หรือ $(System = 0)$ กลับไปทำขั้นตอนที่ 3

ขั้นตอนที่ 5: คำนวณหาค่าระยะการแกว่งของปลายเครื่องมือที่มีค่ามากที่สุด $(Error_{X_{max}})$ ตามระนาบแกน X และเวลาสูงสุดของการเคลื่อนที่ (T_{Opt}^{AB}) โดยสามารถคำนวณได้ดังนี้

$$1. NX^A = Line(X^A, r), \quad NY^A = Line(Y^A, r), \quad NZ^A = Line(Z^A, r)$$

$$2. NX^B = Line(X^B, r), \quad NY^B = Line(Y^B, r), \quad NZ^B = Line(Z^B, r)$$

$$3. [\theta_1^{A2}, \theta_2^{A2}, \theta_3^{A2}, \theta_4^{A2}, \theta_5^{A2}, \theta_6^{A2}] = Inverse_Kinematics(NX^A, NY^A, NZ^A, R, L, d_6, AV_{max})$$

$$4. [\theta_1^{B2}, \theta_2^{B2}, \theta_3^{B2}, \theta_4^{B2}, \theta_5^{B2}, \theta_6^{B2}] = Inverse_Kinematics(NX^B, NY^B, NZ^B, R, L, d_6, AV_{max})$$

$$5. [P_X^{A2}, P_Y^{A2}, P_Z^{A2}] = Forward_Kinematics(\theta_1^{A2}, \theta_2^{A2}, \theta_3^{A2}, \theta_4^{A2}, \theta_5^{A2}, \theta_6^{A2}, L, d_6)$$

$$6. [P_X^{B2}, P_Y^{B2}, P_Z^{B2}] = Forward_Kinematics(\theta_1^{B2}, \theta_2^{B2}, \theta_3^{B2}, \theta_4^{B2}, \theta_5^{B2}, \theta_6^{B2}, L, d_6)$$

$$7. [T_2^A, PT_{12}] = Cal_Time(\theta_1^{A2}, \theta_2^{A2}, \theta_3^{A2}, \theta_4^{A2}, \theta_5^{A2}, \theta_6^{A2}, Acc, AV_{max})$$

$$8. [T_2^B, PT_{22}] = Cal_Time(\theta_1^{B2}, \theta_2^{B2}, \theta_3^{B2}, \theta_4^{B2}, \theta_5^{B2}, \theta_6^{B2}, Acc, AV_{max})$$

$$9. T_{Opt}^{AB} = \max(T_2^A, T_2^B)$$

$$10. [\hat{\theta}_1^{A2}, \hat{\theta}_2^{A2}, \hat{\theta}_3^{A2}, \hat{\theta}_4^{A2}, \hat{\theta}_5^{A2}, \hat{\theta}_6^{A2}] = Forecast_Angle(\theta_1^{A2}, \theta_2^{A2}, \theta_3^{A2}, \theta_4^{A2}, \theta_5^{A2}, \theta_6^{A2}, Acc, AV_{max})$$

11. $[\hat{\theta}_1^{B2}, \hat{\theta}_2^{B2}, \hat{\theta}_3^{B2}, \hat{\theta}_4^{B2}, \hat{\theta}_5^{B2}, \hat{\theta}_6^{B2}] = \text{Forecast_Angle}(\theta_1^{B2}, \theta_2^{B2}, \theta_3^{B2}, \theta_4^{B2}, \theta_5^{B2}, \theta_6^{B2}, \text{Acc}, \text{AV}_{\max})$
12. $[\hat{P}_X^{A2}, \hat{P}_Y^{A2}, \hat{P}_Z^{A2}] = \text{Forward_Kinematics}(\hat{\theta}_1^{A2}, \hat{\theta}_2^{A2}, \hat{\theta}_3^{A2}, \hat{\theta}_4^{A2}, \hat{\theta}_5^{A2}, \hat{\theta}_6^{A2})$
13. $[\hat{P}_X^{B2}, \hat{P}_Y^{B2}, \hat{P}_Z^{B2}] = \text{Forward_Kinematics}(\hat{\theta}_1^{B2}, \hat{\theta}_2^{B2}, \hat{\theta}_3^{B2}, \hat{\theta}_4^{B2}, \hat{\theta}_5^{B2}, \hat{\theta}_6^{B2})$
14. $\text{Error}_{X_t} = |\hat{P}_{X_t}^{A2} + \hat{P}_{X_t}^{B2}|$
15. $\text{Error}_{Y_t} = |\hat{P}_{Y_t}^{A2} - \hat{P}_{Y_t}^{B2}|$
16. $\text{Error}_{Z_t} = |\hat{P}_{Z_t}^{A2} + \hat{P}_{Z_t}^{B2}|$
17. $\text{Error}_{X_{\max}} = \max_{\forall t}(\text{Error}_X)$
18. ทำต่อขั้นตอนที่ 6

ขั้นตอนที่ 6: คำนวณหาค่าระยะความกว้างของปลายเครื่องมือ ($\text{Error}_{R_{\max}}$) บนระนาบแกน Y และ Z โดยมีวิธีการคำนวณดังนี้

1. ถ้า $(\text{Error}_{X_{\max}} \leq D_X)$ ให้คำนวณหา $\text{Error}_{R_{\max}}$

$$1.1 \text{ Error}_{YZ_t} = \max[\text{Error}_{Z_t}, \text{Error}_{Y_t}]$$

$$1.2 \text{ Error}_{R_{\max}} = \max_{\forall t}(\text{Error}_{YZ})$$

2. ถ้า $(\text{Error}_{X_{\max}} > D_X)$ ให้ $\text{Error}_{R_{\max}} = \text{Inf}$

3. ทำต่อขั้นตอนที่ 7

ขั้นตอนที่ 7: ตรวจสอบค่าระยะการแกว่งของปลายเครื่องมือ เมื่อจำนวนตำแหน่งการเคลื่อนที่ของหุ่นยนต์ A และ B เพิ่มขึ้น (เมื่อ r และ r_2 เพิ่มขึ้น) ซึ่งสามารถทำได้ดังนี้

1. ถ้า $(\text{Error}_{R_{\max}} = \text{Inf})$ กำหนดให้ $\text{System} = 0$ และกลับไปขั้นตอนที่ 4

2. ถ้า $(\text{Error}_{R_{\max}} \neq \text{Inf})$ ให้กำหนดค่า $\text{Error}_{\text{Old}}$ ดังนี้

$$2.1 \ k = k + 1$$

$$2.2 \text{ } Error_{Old_k} = Error_{R_{max}}$$

2.3 ทำต่อขั้นตอนที่ 8

ขั้นตอนที่ 8: ตรวจสอบค่า k โดยสามารถทำได้ดังนี้

1. ถ้า $(k \geq 2)$ ให้ตรวจสอบค่า $Error_{Old}$ ใหม่ ดังนี้

1.1 ถ้า $[Error_{Old_k} \leq Error_{Old_{k-1}}]$ กำหนดให้ $System = 1$

1.2 ถ้า $[Error_{Old_k} > Error_{Old_{k-1}}]$ กำหนดให้ $System = 0$

2. กลับไปทำขั้นตอนที่ 4

ขั้นตอนที่ 9: ตรวจสอบค่าระยะการแกว่งของปลายเครื่องมืออยู่ในระยะค่าที่กีดความคลาดเคลื่อนของชิ้นงานเพลักษณะเคลื่อนที่หรือไม่ โดยสามารถทำการตรวจสอบได้ดังนี้

1. ถ้า $(Error_{R_{max}} < R_{dif})$ แสดงว่าหุ่นยนต์สามารถสวมประกอบชิ้นงานขณะเคลื่อนที่ได้ ดังนั้นจึงกำหนดจำนวนตำแหน่งการเคลื่อนที่ $p_2 = [r_1 \quad r_2]$ และให้ทำต่อขั้นตอนที่ 10

2. ถ้า $(Error_{R_{max}} \geq R_{dif})$ แสดงว่าหุ่นยนต์ไม่สามารถสวมประกอบชิ้นงานขณะเคลื่อนที่ได้ ดังนั้นให้แสดงข้อความว่า “Setup New Value MaxTime, Please” แล้วจบการทำงานของโปรแกรม

8. คำนวณหาเวลาการเคลื่อนที่ทั้งหมด (Total Time) และองศาการเคลื่อนที่ของหุ่นยนต์

สำหรับการคำนวณหาเวลาทั้งหมดของการเคลื่อนที่ โดยมีหุ่นยนต์ A และ B ต้องทำการประกอบชิ้นงานในขณะทำการเคลื่อนที่ มีเส้นทางการเคลื่อนที่น้อยที่สุด และใช้เวลาในการเคลื่อนที่น้อยที่สุด ซึ่งมีขั้นตอนในการคำนวณโดยได้แสดงไว้ในรูปภาพผนวกที่ 5 ถึงภาพผนวกที่ 8 และสามารถแสดงขั้นตอนการคำนวณได้ดังนี้

ขั้นตอนที่ 1: คำนวณองศา และเวลาสำหรับการเคลื่อนที่ของทุกๆ ข้อต่อ ตามเส้นทางการเคลื่อนที่ของหุ่นยนต์ ซึ่งสามารถคำนวณได้ดังนี้

1. $NX^A = Line(X^A, r), \quad NY^A = Line(Y^A, r), \quad NZ^A = Line(Z^A, r)$
2. $NX^B = Line(X^B, r_2), \quad NY^B = Line(Y^B, r_2), \quad NZ^B = Line(Z^B, r_2)$
3. $[\theta_1^{A2}, \theta_2^{A2}, \theta_3^{A2}, \theta_4^{A2}, \theta_5^{A2}, \theta_6^{A2}] = Inverse_Kinematics(NX^A, NY^A, NZ^A, R, L, d_6)$
4. $[\theta_1^{B2}, \theta_2^{B2}, \theta_3^{B2}, \theta_4^{B2}, \theta_5^{B2}, \theta_6^{B2}] = Inverse_Kinematics(NX^B, NY^B, NZ^B, R, L, d_6)$
5. $[P_X^{A2}, P_Y^{A2}, P_Z^{A2}] = Forward_Kinematics(\theta_1^{A2}, \theta_2^{A2}, \theta_3^{A2}, \theta_4^{A2}, \theta_5^{A2}, \theta_6^{A2}, L, d_6)$
6. $[P_X^{B2}, P_Y^{B2}, P_Z^{B2}] = Forward_Kinematics(\theta_1^{B2}, \theta_2^{B2}, \theta_3^{B2}, \theta_4^{B2}, \theta_5^{B2}, \theta_6^{B2}, L, d_6)$
7. $[T_2^A, PT_{12}] = Cal_Time(\theta_1^{A2}, \theta_2^{A2}, \theta_3^{A2}, \theta_4^{A2}, \theta_5^{A2}, \theta_6^{A2}, Acc, AV_{max})$
8. $[T_2^B, PT_{22}] = Cal_Time(\theta_1^{B2}, \theta_2^{B2}, \theta_3^{B2}, \theta_4^{B2}, \theta_5^{B2}, \theta_6^{B2}, Acc, AV_{max})$
9. ทำต่อขั้นตอนที่ 2

ขั้นตอนที่ 2: คำนวณประมาณองศาการเคลื่อนที่ของหุ่นยนต์ในแต่ละช่วงเวลา โดยสามารถคำนวณได้ดังนี้

1. ถ้า $(T_2^A = T_2^B)$ ให้ประมาณองศาการเคลื่อนที่ได้ดังนี้

- 1.1 $[\hat{\theta}_1^{A2}, \hat{\theta}_2^{A2}, \hat{\theta}_3^{A2}, \hat{\theta}_4^{A2}, \hat{\theta}_5^{A2}, \hat{\theta}_6^{A2}] = Forecast_Angle(\theta_1^{A2}, \theta_2^{A2}, \theta_3^{A2}, \theta_4^{A2}, \theta_5^{A2}, \theta_6^{A2}, Acc, AV_{max})$
- 1.2 $[\hat{\theta}_1^{B2}, \hat{\theta}_2^{B2}, \hat{\theta}_3^{B2}, \hat{\theta}_4^{B2}, \hat{\theta}_5^{B2}, \hat{\theta}_6^{B2}] = Forecast_Angle(\theta_1^{B2}, \theta_2^{B2}, \theta_3^{B2}, \theta_4^{B2}, \theta_5^{B2}, \theta_6^{B2}, Acc, AV_{max})$
- 1.3 ทำต่อขั้นตอนที่ 3

2. ถ้า $(T_2^A > T_2^B)$ ให้ประมาณองศาการเคลื่อนที่ได้ดังนี้

- 2.1 $[\hat{\theta}_1^{A2}, \hat{\theta}_2^{A2}, \hat{\theta}_3^{A2}, \hat{\theta}_4^{A2}, \hat{\theta}_5^{A2}, \hat{\theta}_6^{A2}] = Forecast_Angle(\theta_1^{A2}, \theta_2^{A2}, \theta_3^{A2}, \theta_4^{A2}, \theta_5^{A2}, \theta_6^{A2}, Acc, AV_{max})$
- 2.2 $n_{B2} = length(NX^B)$
- 2.3 $NNX^B = \{NX_1^B, NX_2^B, NX_3^B, ..., NX_{(n_{B2}-1)}^B\}$
- 2.4 $NNY^B = \{NY_1^B, NY_2^B, NY_3^B, ..., NY_{(n_{B2}-1)}^B\}$
- 2.5 $NNZ^B = \{NZ_1^B, NZ_2^B, NZ_3^B, ..., NZ_{(n_{B2}-1)}^B\}$
- 2.6 $System = 1, NT_2^B = 0$ และ $q = 0$
- 2.7 ถ้า $(T_2^A > NT_2^B)$ และ $(System = 1)$ ให้กำหนดค่าต่างๆ ดังนี้
 - 2.7.1 $q = q + 1$
 - 2.7.2 $h = q + (n_{B2} - 1)$

2.7.3 ทำต่อข้อ 2.9

2.8 ถ้า $(T_2^A \leq NT_2^B)$ หรือ $(System = 0)$ กำหนดให้

$$2.8.1 \quad T_2^B = NT_2^B$$

2.8.2 ทำต่อขั้นตอนที่ 3

2.9 ถ้า $[NX_{(n_{B2}-1)}^B > NX_{n_{B2}}^B]$ กำหนดให้

$$2.9.1 \quad NNX_h^B = NNX_{(h-1)}^B - 0.005$$

2.9.2 ถ้า $[NNX_h^B \leq NX_{n_{B2}}^B]$ กำหนดให้

2.9.2.1 $System = 0$

$$2.9.2.2 \quad NNX_h^B = NX_{n_{B2}}^B$$

2.9.2.3 ทำต่อข้อ 2.12

2.9.3 ถ้า $[NNX_h^B > NX_{n_{B2}}^B]$ กำหนดให้

2.9.3.1 $System = 1$

2.9.3.2 ทำต่อข้อ 2.12

2.10 ถ้า $[NX_{(n_{B2}-1)}^B < NX_{n_{B2}}^B]$ กำหนดให้

$$2.10.1 \quad NNX_h^B = NNX_{(h-1)}^B + 0.005$$

2.10.2 ถ้า $[NNX_h^B \geq NX_{n_{B2}}^B]$ กำหนดให้

2.10.2.1 $System = 0$

$$2.10.2.2 \quad NNX_h^B = NX_{n_{B2}}^B$$

2.10.2.3 ทำต่อข้อ 2.12

2.10.3 ถ้า $[NNX_h^B < NX_{n_{B2}}^B]$ กำหนดให้

2.10.3.1 $System = 1$

2.10.3.2 ทำต่อข้อ 2.12

2.11 ถ้า $[NX_{(n_{B2}-1)}^B = NX_{n_{B2}}^B]$ กำหนดให้

$$2.11.1 \quad NNX_h^B = NX_{(n_{B2}-1)}^B$$

2.11.2 $System = 1$

2.11.3 ทำต่อข้อ 2.12

$$2.12 \quad NNY_h^B = NNY_{(h-1)}^B + \left\{ \frac{[NY_{n_{B2}}^B - NY_{(n_{B2}-1)}^B]}{[NX_{n_{B2}}^B - NX_{(n_{B2}-1)}^B]} \right\} * [NNX_h^B - NNX_{(n_{B2}-1)}^B]$$

$$2.13 \quad NNZ_h^B = NNZ_{(h-1)}^B + \left\{ \frac{[NZ_{n_{B2}}^B - NZ_{(n_{B2}-1)}^B]}{[NX_{n_{B2}}^B - NX_{(n_{B2}-1)}^B]} \right\} * [NNX_h^B - NNX_{(n_{B2}-1)}^B]$$

2.14 ถ้า $[NNZ_h^B \leq Z_3^B]$ กำหนดให้

2.1.4.1 $System = 0$

$$2.1.4.2 \quad NNX_h^B = NX_{n_{B2}}^B$$

$$2.1.4.3 \quad NNY_h^B = NY_{n_{B2}}^B$$

$$2.1.4.4 \quad NNZ_h^B = NZ_{n_{B2}}^B$$

2.1.4.5 ทำต่อข้อ 2.16

2.15 ถ้า $[NNZ_h^B > Z_3^B]$ กำหนดให้

2.1.5.1 $System = 1$

$$2.1.5.2 \quad NNX_{(h+1)}^B = NX_{n_{B2}}^B$$

$$2.1.5.3 \quad NNY_{(h+1)}^B = NY_{n_{B2}}^B$$

$$2.1.5.4 \quad NNZ_{(h+1)}^B = NZ_{n_{B2}}^B$$

2.1.5.5 ทำต่อข้อ 2.16

$$2.16 \quad NNX^A = NX^A, \quad NNY^A = NY^A, \quad NNZ^A = NZ^A$$

$$2.17 \quad [\theta_1^{B2}, \theta_2^{B2}, \theta_3^{B2}, \theta_4^{B2}, \theta_5^{B2}, \theta_6^{B2}] = \text{Inverse_Kinematics}(NNX^B, NNY^B, NNZ^B, R, L, d_6)$$

$$2.18 \quad [NT_2^B, NT_{22}^B] = \text{Cal_Time}(\theta_1^{B2}, \theta_2^{B2}, \theta_3^{B2}, \theta_4^{B2}, \theta_5^{B2}, \theta_6^{B2}, Acc, AV_{\max})$$

$$2.19 \quad [\hat{\theta}_1^{B2}, \hat{\theta}_2^{B2}, \hat{\theta}_3^{B2}, \hat{\theta}_4^{B2}, \hat{\theta}_5^{B2}, \hat{\theta}_6^{B2}] = \text{Forecast_Angle}(\theta_1^{B2}, \theta_2^{B2}, \theta_3^{B2}, \theta_4^{B2}, \theta_5^{B2}, \theta_6^{B2}, Acc, AV_{\max})$$

2.20 กลับไปทำข้อที่ 2.7

3. ถ้า $(T_2^A < T_2^B)$ ให้ประมาณองศาการเคลื่อนที่ได้ดังนี้

$$3.1 \quad [\hat{\theta}_1^{B2}, \hat{\theta}_2^{B2}, \hat{\theta}_3^{B2}, \hat{\theta}_4^{B2}, \hat{\theta}_5^{B2}, \hat{\theta}_6^{B2}] = \text{Forecast_Angle}(\theta_1^{B2}, \theta_2^{B2}, \theta_3^{B2}, \theta_4^{B2}, \theta_5^{B2}, \theta_6^{B2}, Acc, AV_{\max})$$

$$3.2 \quad n_{A2} = \text{length}(NX^A)$$

$$3.3 \quad NNX^A = \{NX_1^A, NX_2^A, NX_3^A, \dots, NX_{(n_{A2}-1)}^A\}$$

$$3.4 \quad NNY^A = \{NY_1^A, NY_2^A, NY_3^A, \dots, NY_{(n_{A2}-1)}^A\}$$

$$3.5 \quad NNZ^A = \{NZ_1^A, NZ_2^A, NZ_3^A, \dots, NZ_{(n_{A2}-1)}^A\}$$

$$3.6 \quad System = 1, \quad NT_2^A \text{ และ } q = 0$$

3.7 ถ้า $(T_2^B > NT_2^A)$ และ $(System = 1)$ กำหนดค่าต่างๆ ดังนี้

$$3.7.1 \quad q = q + 1$$

$$3.7.2 \quad h = q + (n_{A2} - 1)$$

3.7.3 ทำต่อข้อ 3.9

3.8 ถ้า $(T_2^B \leq NT_2^A)$ หรือ $(System = 0)$ กำหนดค่าต่างๆ ดังนี้

$$3.8.1 \quad T_2^A = NT_2^A$$

3.8.2 ทำต่อขั้นตอนที่ 3

3.9 ถ้า $[NX_{(n_{A2}-1)}^A > NX_{n_{A2}}^A]$ กำหนดให้

$$3.9.1 \quad NNX_h^A = NNX_{(h-1)}^A - 0.005$$

3.9.2 ถ้า $[NNX_h^A \leq NX_{n_{A2}}^A]$ กำหนดให้

$$3.9.2.1 \quad System = 0$$

$$3.9.2.2 \quad NNX_h^A = NX_{n_{A2}}^A$$

3.9.2.3 ทำต่อข้อ 3.12

3.9.3 ถ้า $[NNX_h^A > NX_{n_{A2}}^A]$ กำหนดให้

$$3.9.3.1 \quad System = 1$$

3.9.3.2 ทำต่อข้อ 3.12

3.10 ถ้า $[NX_{(n_{A2}-1)}^A < NX_{n_{A2}}^A]$ กำหนดให้

$$3.10.1 \quad NNX_h^A = NNX_{(h-1)}^A + 0.005$$

3.10.2 ถ้า $[NNX_h^A \geq NX_{n_{A2}}^A]$ กำหนดให้

$$3.10.2.1 \quad System = 0$$

$$3.10.2.2 \quad NNX_h^A = NX_{n_{A2}}^A$$

3.10.2.3 ทำต่อข้อ 3.12

3.10.3 ถ้า $[NNX_h^A < NX_{n_{A2}}^A]$ กำหนดให้

$$3.10.3.1 \quad System = 1$$

3.10.3.2 ทำต่อข้อ 3.12

3.11 ถ้า $[NX_{(n_{A2}-1)}^A = NX_{n_{A2}}^A]$ กำหนดให้

$$3.11.1 \quad [NNX_h^A = NX_{(h-1)}^A]$$

$$3.11.2 \quad System = 1$$

3.11.3 ทำต่อข้อ 3.12

$$3.12 \quad NNY_h^A = NNY_{(h-1)}^A + \left\{ \frac{[NY_{(n_{A2})}^A - NY_{(n_{A2}-1)}^A]}{[NX_{(n_{A2})}^A - NX_{(n_{A2}-1)}^A]} \right\} * [NNX_h^A - NNX_{(n_{A2}-1)}^A]$$

$$3.13 \quad NNZ_h^A = NNZ_{(h-1)}^A + \left\{ \frac{[NZ_{(n_{A2})}^A - NZ_{(n_{A2}-1)}^A]}{[NX_{(n_{A2})}^A - NX_{(n_{A2}-1)}^A]} \right\} * [NNX_h^A - NNX_{(n_{A2}-1)}^A]$$

3.14 ถ้า $[NNZ_h^A \leq Z_3^A]$ กำหนดให้

$$3.1.4.1 \quad System = 0$$

$$3.1.4.2 \quad NNX_h^A = NX_{n_{A2}}^A$$

$$3.1.4.3 \quad NNY_h^A = NY_{n_{A2}}^A$$

$$3.1.4.4 \quad NNZ_h^A = NZ_{n_{A2}}^A$$

3.1.4.5 ทำต่อข้อ 3.16

3.15 ถ้า $[NNZ_h^A > Z_3^A]$ กำหนดให้

3.1.5.1 $System = 1$

$$3.1.5.2 \quad NNX_{(h+1)}^A = NX_{n_{A2}}^A$$

$$3.1.5.3 \quad NNY_{(h+1)}^A = NY_{n_{A2}}^A$$

$$3.1.5.4 \quad NNZ_{(h+1)}^A = NZ_{n_{A2}}^A$$

3.1.5.5 ทำต่อข้อ 3.16

$$3.16 \quad NNX^B = NX^B, \quad NNY^B = NY^B, \quad NNZ^B = NZ^B$$

$$3.17 \quad [\theta_1'^{A2}, \theta_2'^{A2}, \theta_3'^{A2}, \theta_4'^{A2}, \theta_5'^{A2}, \theta_6'^{A2}] = \text{Inverse_Kinematics}(NNX^A, NNY^A, NNZ^A, R, L, d_6)$$

$$3.18 \quad [NT_2^A, NT_{12}] = \text{Cal_Time}(\theta_1'^{A2}, \theta_2'^{A2}, \theta_3'^{A2}, \theta_4'^{A2}, \theta_5'^{A2}, \theta_6'^{A2}, Acc, AV_{\max})$$

$$3.19 \quad [\hat{\theta}_1^{A2}, \hat{\theta}_2^{A2}, \hat{\theta}_3^{A2}, \hat{\theta}_4^{A2}, \hat{\theta}_5^{A2}, \hat{\theta}_6^{A2}] = \text{Forecast_Angle}(\theta_1'^{A2}, \theta_2'^{A2}, \theta_3'^{A2}, \theta_4'^{A2}, \theta_5'^{A2}, \theta_6'^{A2}, Acc, AV_{\max})$$

3.20 กลับไปทำข้อ 3.7

ขั้นตอนที่ 3: คำนวณตำแหน่งการเคลื่อนที่ของหุ่นยนต์ทั้ง 2 ตัว สำหรับช่วงเวลา t ใดๆ

$$1. \quad [\hat{P}_X^{A2}, \hat{P}_Y^{A2}, \hat{P}_Z^{A2}] = \text{Forward_Kinematics}(\hat{\theta}_1^{A2}, \hat{\theta}_2^{A2}, \hat{\theta}_3^{A2}, \hat{\theta}_4^{A2}, \hat{\theta}_5^{A2}, \hat{\theta}_6^{A2}, L, d_6)$$

$$2. \quad [\hat{P}_X^{B2}, \hat{P}_Y^{B2}, \hat{P}_Z^{B2}] = \text{Forward_Kinematics}(\hat{\theta}_1^{B2}, \hat{\theta}_2^{B2}, \hat{\theta}_3^{B2}, \hat{\theta}_4^{B2}, \hat{\theta}_5^{B2}, \hat{\theta}_6^{B2}, L, d_6)$$

3. ทำต่อขั้นตอนที่ 4

ขั้นตอนที่ 4: ตรวจสอบเวลาการทำงานของหุ่นยนต์ทั้ง 2 ตัว พร้อมทั้งปรับเวลาการทำงานของหุ่นยนต์ให้ใกล้เคียงกัน

1. ถ้า $(T_2^A = T_2^B)$ กำหนดให้

$$1.1 \quad [P_X'^{A2}, P_Y'^{A2}, P_Z'^{A2}] = \text{Check_Error}(NX^A, NY^A, NZ^A, \hat{P}_X^{A2}, \hat{P}_Y^{A2}, \hat{P}_Z^{A2}, Acc, PT_{12})$$

$$1.2 \quad [P_X'^{B2}, P_Y'^{B2}, P_Z'^{B2}] = \text{Check_Error}(NX^B, NY^B, NZ^B, \hat{P}_X^{B2}, \hat{P}_Y^{B2}, \hat{P}_Z^{B2}, Acc, PT_{22})$$

2. ถ้า $(T_2^A > T_2^B)$ กำหนดให้

$$2.1 [P_X'^{A2}, P_Y'^{A2}, P_Z'^{A2}] = Check_Error(NX^A, NY^A, NZ^A, \hat{P}_X^{A2}, \hat{P}_Y^{A2}, \hat{P}_Z^{A2}, Acc, PT_{12})$$

$$2.2 [P_X'^{B2}, P_Y'^{B2}, P_Z'^{B2}] = Check_Error(NNX^B, NNY^B, NNZ^B, \hat{P}_X^{B2}, \hat{P}_Y^{B2}, \hat{P}_Z^{B2}, Acc, NT_{22})$$

3. ถ้า $(T_2^A < T_2^B)$ กำหนดให้

$$3.1 [P_X'^{A2}, P_Y'^{A2}, P_Z'^{A2}] = Check_Error(NNX^A, NNY^A, NNZ^A, \hat{P}_X^{A2}, \hat{P}_Y^{A2}, \hat{P}_Z^{A2}, Acc, NT_{12})$$

$$3.2 [P_X'^{B2}, P_Y'^{B2}, P_Z'^{B2}] = Check_Error(NX^B, NY^B, NZ^B, \hat{P}_X^{B2}, \hat{P}_Y^{B2}, \hat{P}_Z^{B2}, Acc, PT_{22})$$

4. ทำต่อขั้นตอนที่ 5

ขั้นตอนที่ 5: คำนวณขนาดของตำแหน่งการเคลื่อนที่ของหุ่นยนต์ทั้ง 2 ตัว

$$1. nP_{A1} = length(\hat{\theta}_1^{A1})$$

$$2. nP_{A2} = length(\hat{\theta}_1^{A2})$$

$$3. N_A^{Total} = nP_{A1} + nP_{A2}$$

$$4. nP_{B1} = length(\hat{\theta}_1^{B1})$$

$$5. nP_{B2} = length(\hat{\theta}_1^{B2})$$

$$6. N_B^{Total} = nP_{B1} + nP_{B2}$$

$$7. t_1 = 1 \text{ และทำต่อขั้นตอนที่ 6}$$

ขั้นตอนที่ 6: ตรวจสอบเวลาทำงานของหุ่นยนต์ A และรวบรวมองศาการเคลื่อนที่ของหุ่นยนต์ A ตลอดเส้นทางการเคลื่อนที่ (ช่วงที่ 1 และช่วงที่ 2)

1. ถ้า $(t_1 \leq N_A^{Total})$ ทำการรวมองศาการเคลื่อนที่ของหุ่นยนต์ A ได้ดังนี้

1.1 ถ้า $(t_1 \leq nP_{A1})$ กำหนดให้

$$1.1.1 \theta_1^A(t_1) = \hat{\theta}_1^{A1}(t_1)$$

$$1.1.2 \theta_2^A(t_1) = \hat{\theta}_2^{A1}(t_1)$$

$$1.1.3 \theta_3^A(t_1) = \hat{\theta}_3^{A1}(t_1)$$

$$1.1.4 \theta_4^A(t_1) = \hat{\theta}_4^{A1}(t_1)$$

$$1.1.5 \theta_5^A(t_1) = \hat{\theta}_5^{A1}(t_1)$$

$$1.1.6 \theta_6^A(t_1) = \hat{\theta}_6^{A1}(t_1)$$

$$1.1.7 t_1 = t_1 + 1$$

1.2 ถ้า $(t_1 > nP_{A1})$ กำหนดให้

$$1.2.1 \theta_1^A(t_1) = \hat{\theta}_1^{A2}(t_1 - nP_{A1})$$

$$1.2.2 \theta_2^A(t_1) = \hat{\theta}_2^{A2}(t_1 - nP_{A1})$$

$$1.2.3 \theta_3^A(t_1) = \hat{\theta}_3^{A2}(t_1 - nP_{A1})$$

$$1.2.4 \theta_4^A(t_1) = \hat{\theta}_4^{A2}(t_1 - nP_{A1})$$

$$1.2.5 \theta_5^A(t_1) = \hat{\theta}_5^{A2}(t_1 - nP_{A1})$$

$$1.2.6 \theta_6^A(t_1) = \hat{\theta}_6^{A2}(t_1 - nP_{A1})$$

$$1.2.7 t_1 = t_1 + 1$$

2. ถ้า $(t_1 > N_A^{Total})$ กำหนดให้ $t_2 = 1$ และทำการรวมองศาการเคลื่อนที่ของหุ่นยนต์ B ในขั้นตอนที่ 7 ต่อไป

ขั้นตอนที่ 7: ตรวจสอบเวลาทำงานของหุ่นยนต์ B และรวบรวมองศาการเคลื่อนที่ของหุ่นยนต์ตลอดเส้นทางการเคลื่อนที่ (ช่วงที่ 1 และช่วงที่ 2)

1. ถ้า $(t_2 \leq N_B^{Total})$ ทำการรวมองศาการเคลื่อนที่ของหุ่นยนต์ B ได้ดังนี้

1.1 ถ้า $(t_2 \leq nP_{B1})$ กำหนดให้

$$1.1.1 \theta_1^B(t_2) = \hat{\theta}_1^{B1}(t_2)$$

$$1.1.2 \theta_2^B(t_2) = \hat{\theta}_2^{B1}(t_2)$$

$$1.1.3 \theta_3^B(t_2) = \hat{\theta}_3^{B1}(t_2)$$

$$1.1.4 \theta_4^B(t_2) = \hat{\theta}_4^{B1}(t_2)$$

$$1.1.5 \theta_5^B(t_2) = \hat{\theta}_5^{B1}(t_2)$$

$$1.1.6 \theta_6^B(t_2) = \hat{\theta}_6^{B1}(t_2)$$

$$1.1.7 \ t_2 = t_2 + 1$$

1.2 ถ้า $(t_2 > nP_{B1})$ กำหนดให้

$$1.2.1 \ \theta_1^B(t_2) = \hat{\theta}_1^{B2}(t_2 - nP_{B1})$$

$$1.2.2 \ \theta_2^B(t_2) = \hat{\theta}_2^{B2}(t_2 - nP_{B1})$$

$$1.2.3 \ \theta_3^B(t_2) = \hat{\theta}_3^{B2}(t_2 - nP_{B1})$$

$$1.2.4 \ \theta_4^B(t_2) = \hat{\theta}_4^{B2}(t_2 - nP_{B1})$$

$$1.2.5 \ \theta_5^B(t_2) = \hat{\theta}_5^{B2}(t_2 - nP_{B1})$$

$$1.2.6 \ \theta_6^B(t_2) = \hat{\theta}_6^{B2}(t_2 - nP_{B1})$$

$$1.2.7 \ t_2 = t_2 + 1$$

2. ถ้า $(t_2 > N_B^{Total})$ ทำต่อขั้นตอนที่ 8

ขั้นตอนที่ 8: คำนวณหาเวลาการทำงานของหุ่นยนต์ทั้ง 2 ตัว

$$1. T_A^{Total} = T_1^A + T_2^A$$

$$2. T_B^{Total} = T_1^B + T_2^B$$

3. จบโปรแกรม

ผลและวิจารณ์

ผลและวิจารณ์

สำหรับการศึกษาการวางแผนการเคลื่อนที่ของหุ่นยนต์ 2 ตัวที่ทำงานร่วมกันนี้ สิ่งที่เราต้องการได้จากงานวิจัยครั้งนี้ คือ

- (1) ต้องการหาค่าจำนวนตำแหน่งของการเคลื่อนที่ (p_i) ของหุ่นยนต์ให้มีค่าน้อยที่สุด
- (2) ต้องการหาเส้นทางการเคลื่อนที่โดยใช้เวลาในการเคลื่อนที่น้อยที่สุด ซึ่งทำการเปรียบเทียบการเคลื่อนที่แบบเส้นตรง และแบบแบ่งช่วงเท่านั้น

สำหรับการวิจัยครั้งนี้ได้มีการกำหนดลักษณะของหุ่นยนต์ที่ใช้ในงานวิจัยในครั้งนี้ คือ

1. ความยาวข้อต่อของหุ่นยนต์ (Link) ในแต่ละข้อต่อ ซึ่งแสดงค่าไว้ในตารางที่ 1

ตารางที่ 1 แสดงค่าความยาวของข้อต่อของหุ่นยนต์ 6 แกนทั้ง 2 ตัว

ความยาวของข้อต่อที่ i	ขนาดความยาว (mm)
ความยาวของฐาน (d_0)	0
ความสูงของฐาน หรือข้อต่อที่ 1 (l_1)	40
ความยาวของข้อต่อที่ 2 (l_2)	35
ความยาวของข้อต่อที่ 3 (l_3)	35
ความยาวของข้อต่อที่ 4 ถึง 6 (d_6)	25

2. ค่าความเร่งเชิงมุมของหุ่นยนต์ คือ 0.5 rad/s^2

3. ค่าความเร็วสูงสุดของหุ่นยนต์ที่สามารถเคลื่อนที่ได้ (Maximum angular valocity) คือ 15 cycle/min

4. เวลาที่ใช้ในการเคลื่อนที่สูงสุด (cycle time) คือ 1,850 วินาที

5. เมทริกซ์สำหรับการหมุน (Rotation Matrix) ของปลายเครื่องมือโดยอ้างอิงจากฐานของ

$$\text{หุ่นยนต์ หรือจากตำแหน่งอ้างอิง คือ } R_0^6 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

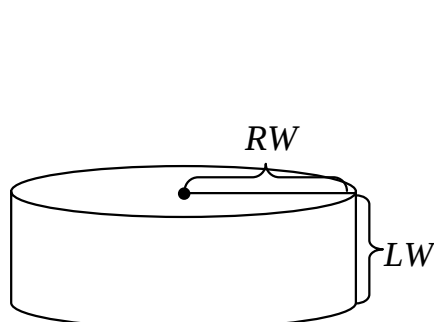
6. ค่าความผิดพลาด (Error) สำหรับการเดินทางที่เบี่ยงเบนไปจากเส้นตรงมีค่าเท่า 5

สำหรับตำแหน่งเส้นทางการเคลื่อนที่ของหุ่นยนต์ A และ B ที่ได้กำหนดการเคลื่อนที่ในแต่ละช่วงการเคลื่อนที่ ได้มีการแบ่งลักษณะการสวมประกอบชิ้นงานเพลาทรงออกเป็น 3 ประเภท ดังนี้

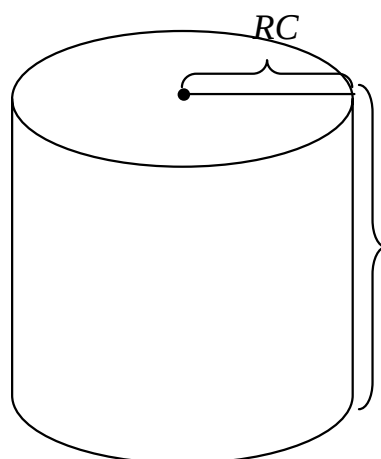
1. นำชิ้นงานมาสวมประกอบด้วยหุ่นยนต์ A และ B ณ ตำแหน่งบริเวณตรงกลางชิ้นงานเพลาทรงกระบอก โดยสามารถทำการกำหนดตำแหน่งการเคลื่อนที่ได้ดังนี้

$$1.1 \text{ ตำแหน่งการเคลื่อนที่หุ่นยนต์ A คือ } \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} = \begin{bmatrix} (X_1, Y_1, Z_1) \\ (X_2, Y_2, Z_2) \\ (X_3, Y_3, Z_3) \end{bmatrix} = \begin{bmatrix} (35, 40, 0) \\ (0, 50, 40) \\ (-35, 50, 0) \end{bmatrix}$$

$$1.2 \text{ ตำแหน่งการเคลื่อนที่หุ่นยนต์ B คือ } \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} = \begin{bmatrix} (X_1, Y_1, Z_1) \\ (X_2, Y_2, Z_2) \\ (X_3, Y_3, Z_3) \end{bmatrix} = \begin{bmatrix} (-35, 40, 0) \\ (-12, 50, 40) \\ (33, 50, 0) \end{bmatrix}$$



วงแหวน (Gasket)



ทรงกระบอก (Cylinder)

ภาพที่ 9 ลักษณะของชิ้นงานที่นำมาใช้ในการวางแผนเส้นทางการเคลื่อนที่ของหุ่นยนต์

1.3 ลักษณะของชิ้นงานตัวอย่างที่มีลักษณะดังรูปที่ 2 ซึ่งได้นำมาใช้ในการทดลองครั้งนี้ คือ

1.3.1 ลักษณะของตัวอย่างชิ้นงานชุดที่ 1 ซึ่งแสดงไว้ในตารางที่ 2

ตารางที่ 2 แสดงค่าลักษณะ และขนาดของตัวอย่างชิ้นงานชุดที่ 1

คุณลักษณะ	ขนาดความยาว (mm)			
	ตัวอย่างที่ 1	ตัวอย่างที่ 2	ตัวอย่างที่ 3	ตัวอย่างที่ 4
เพลาทรงกระบอก				
รัศมี	4.00	4.00	4.00	4.00
ความยาว	20.00	20.00	20.00	20.00
เพลาวงแหวน				
รัศมี	4.50	4.10	4.06	4.03
ความยาว	4.00	4.00	4.00	4.00
พิกัดความคลาดเคลื่อน (R_{dif})	0.50	0.10	0.06	0.03

1.3.2 ลักษณะของตัวอย่างชิ้นงานชุดที่ 2 ซึ่งแสดงไว้ในตารางที่ 3

ตารางที่ 3 แสดงค่าลักษณะ และขนาดของตัวอย่างชิ้นงานชุดที่ 2

คุณลักษณะ	ขนาดความยาว (mm)			
	ตัวอย่างที่ 1	ตัวอย่างที่ 2	ตัวอย่างที่ 3	ตัวอย่างที่ 4
เพลาทรงกระบอก				
รัศมี	3.50	3.50	3.50	3.50
ความยาว	20.00	20.00	20.00	20.00
เพลาวงแหวน				
รัศมี	4.00	3.6	3.56	3.53
ความยาว	4.00	4.00	4.00	4.00
พิกัดความคลาดเคลื่อน (R_{dif})	0.50	0.10	0.06	0.03

1.3.3 ลักษณะของตัวอย่างชิ้นงานชุดที่ 3 ซึ่งแสดงไว้ในตารางที่ 4

ตารางที่ 4 แสดงค่าลักษณะ และขนาดของตัวอย่างชิ้นงานชุดที่ 3

คุณลักษณะ	ขนาดความยาว (mm)			
	ตัวอย่างที่ 1	ตัวอย่างที่ 2	ตัวอย่างที่ 3	ตัวอย่างที่ 4
เพลาทรงกระบอก				
รัศมี	3.75	3.95	3.97	3.99
ความยาว	20.00	20.00	20.00	20.00
เพลาวงแหวน				
รัศมี	4.25	4.05	4.03	4.02
ความยาว	4.00	4.00	4.00	4.00
พิกัดความคลาดเคลื่อน (R_{dif})	0.50	0.10	0.06	0.03

จากการประมวลผลหาค่าจำนวนตำแหน่ง และเวลาที่ใช้สำหรับการเคลื่อนย้ายชิ้นงาน ทำงานร่วมกันของหุ่นยนต์ทั้ง 2 ตัว ผลที่ได้แสดงไว้ในตารางที่ 5 สำหรับกลุ่มตัวอย่างชิ้นงานชุดที่ 1 ตารางที่ 6 สำหรับกลุ่มตัวอย่างชิ้นงานชุดที่ 2 และตารางที่ 7 สำหรับกลุ่มตัวอย่างชิ้นงานชุดที่ 3 ซึ่งสามารถสรุปการประมวลผลการทำงานของหุ่นยนต์ได้ดังนี้

ตารางที่ 5 แสดงค่าจำนวนตำแหน่ง และเวลาที่ใช้สำหรับการเคลื่อนที่ของตัวอย่างชิ้นงานชุดที่ 1

รายการ	ค่าที่ได้จากการเคลื่อนที่ของหุ่นยนต์ A และ B											
	ตัวอย่างที่ 1			ตัวอย่างที่ 2			ตัวอย่างที่ 3			ตัวอย่างที่ 4		
	$(R_{dif} = 0.50 \text{ mm})$			$(R_{dif} = 0.10 \text{ mm})$			$(R_{dif} = 0.06 \text{ mm})$			$(R_{dif} = 0.03 \text{ mm})$		
	Linear	Propose	%ลด	Linear	Propose	%ลด	Linear	Propose	%ลด	Linear	Propose	%ลด
1. เวลาเคลื่อนที่ (วินาที)	915.28	671.73	26.61	1,550.47	1,150.86	25.77	1,553.83	1,164.69	25.04	1,557.56	1,159.9	25.53
2. จำนวนช่วงการเคลื่อนที่												
ช่วงแรก (ช่วง)	189	11	94.18	399	18	95.49	399	21	94.74	399	18	95.49
3. จำนวนช่วงการเคลื่อนที่												
ช่วงสอง (ช่วง)	407	407	0.00	1,386	1,386	0.00	1,395	1,395	0.00	1,405	1,405	0.00
4. จำนวนช่วงการเคลื่อนที่												
ทั้งหมด (ช่วง)	596	418	29.87	1,785	1,404	21.34	1,794	1,416	21.07	1,804	1,423	21.12
5. ระยะห่างของปลาย												
เครื่องมือ (mm)	0.27	0.27	-	0.08	0.08	-	0.05	0.05	-	0.02	0.02	-

จากตัวอย่างชิ้นงานในตัวอย่างชุดที่ 1 ได้ผลการประมวลข้อมูลดังตารางที่ 5 พบว่าเมื่อกำหนดค่าพิสัยความคลาดเคลื่อนของชิ้นงานเพล่า (R_{dif}) มีค่าเท่ากับ 0.50 มิลลิเมตร หุ่นยนต์เคลื่อนที่แบบแบ่งช่วงใช้เวลาสำหรับการเคลื่อนย้ายชิ้นงานทั้งหมด 671.73 วินาที โดยจำนวนตำแหน่งการเคลื่อนที่ช่วงแรก 11 ตำแหน่ง และจำนวนตำแหน่งการเคลื่อนที่ช่วงสุดท้าย 407 ตำแหน่ง ซึ่งมีจำนวนตำแหน่งการเคลื่อนที่ตลอดเส้นทาง 418 ตำแหน่ง และมีค่าระยะห่างของปลายเครื่องมือขณะเคลื่อนที่ในช่วงที่สองแตกต่างกันมากที่สุด 0.27 มิลลิเมตร ซึ่งมีค่าน้อยกว่าค่าพิสัยความคลาดเคลื่อนของชิ้นงานที่กำหนดไว้ทำให้หุ่นยนต์สามารถสวมประกอบชิ้นงานขณะเคลื่อนที่ได้ แต่สำหรับการเคลื่อนที่แบบปัจจุบันโดยหุ่นยนต์มีการเคลื่อนที่แบบเส้นตรงซึ่งมีค่าพิสัยความคลาดเคลื่อนของชิ้นงานเพล่าเท่ากันพบว่า หุ่นยนต์ใช้เวลาในการเคลื่อนที่ 915.28 วินาที มีจำนวนตำแหน่งการเคลื่อนที่สำหรับช่วงแรก 189 ตำแหน่ง มีจำนวนตำแหน่งการเคลื่อนที่ช่วงสุดท้าย 407 ตำแหน่ง และมีจำนวนตำแหน่งการเคลื่อนที่ทั้งหมด 596 ตำแหน่ง โดยมีระยะห่างของปลายเครื่องมือขณะเคลื่อนย้ายชิ้นงานช่วงสุดท้ายเท่ากับ 0.27 มิลลิเมตร ซึ่งจะเห็นได้ว่าการเคลื่อนที่แบบแบ่งช่วงสามารถลดจำนวนตำแหน่ง และเวลาทั้งหมดที่ใช้สำหรับการเคลื่อนที่ลงร้อยละ 29.87 และ 26.61 ตามลำดับเมื่อเทียบกับการเคลื่อนที่แบบเส้นตรง เมื่อลดค่าพิสัยความคลาดเคลื่อนของชิ้นงานมาเป็นขนาด 0.10 มิลลิเมตรพบว่า หุ่นยนต์เคลื่อนที่แบบแบ่งช่วงใช้เวลาสำหรับการเคลื่อนย้ายชิ้นงานทั้งหมด 1,150.86 วินาที มีจำนวนตำแหน่งการเคลื่อนที่ช่วงแรก 18 ตำแหน่ง มีจำนวนตำแหน่งการเคลื่อนที่ช่วงสุดท้ายเท่ากับ 1,386 ตำแหน่ง และมีจำนวนตำแหน่งการเคลื่อนที่ตลอดเส้นทางทั้งหมด 1,404 ตำแหน่ง โดยปลายเครื่องมือของหุ่นยนต์มีตำแหน่งการเคลื่อนที่แตกต่างกันมากที่สุด 0.08 มิลลิเมตรสำหรับการเคลื่อนที่ในช่วงสุดท้าย แต่สำหรับกรณีหุ่นยนต์เคลื่อนที่แบบเส้นตรงโดยมีค่าพิสัยความคลาดเคลื่อนของชิ้นงานเพล่าเท่ากันพบว่า หุ่นยนต์ใช้เวลาเคลื่อนที่ทั้งหมด 1,550.47 วินาที ซึ่งมีจำนวนตำแหน่งการเคลื่อนที่ช่วงแรก 399 ตำแหน่ง มีจำนวนตำแหน่งการเคลื่อนที่ช่วงสุดท้าย 1,386 ตำแหน่ง และมีจำนวนตำแหน่งการเคลื่อนที่ทั้งหมด 1,785 ตำแหน่ง โดยมีระยะห่างของปลายเครื่องมือขณะเคลื่อนย้ายชิ้นงานเพล่าช่วงสุดท้าย 0.08 มิลลิเมตร ซึ่งการเคลื่อนที่แบบแบ่งช่วงสามารถลดจำนวนตำแหน่งได้ร้อยละ 21.34 และลดเวลาในการเคลื่อนที่ทั้งหมดลงได้ร้อยละ 25.77 เมื่อเทียบกับการเคลื่อนที่แบบเส้นตรง

สำหรับในกรณีที่ลดค่าพิสัยความคลาดเคลื่อนของชิ้นงานเพล่าเหลือ 0.03 มิลลิเมตร พบว่า หุ่นยนต์เคลื่อนที่แบบแบ่งช่วงใช้เวลาสำหรับเคลื่อนย้ายชิ้นงานเพล่าทั้งหมด 1,159.90 วินาที โดยมีจำนวนตำแหน่งการเคลื่อนที่ช่วงแรก 18 ตำแหน่ง จำนวนตำแหน่งการเคลื่อนที่ช่วงสุดท้าย 1,405 ตำแหน่ง และมีจำนวนตำแหน่งการเคลื่อนที่ตลอดเส้นทางทั้งหมด 1,423 ตำแหน่ง โดยปลาย

เครื่องมือของหุ่นยนต์มีตำแหน่งการเคลื่อนที่แตกต่างกันมากที่สุด 0.02 มิลลิเมตร สำหรับการเคลื่อนที่ในช่วงที่สอง แต่เมื่อหุ่นยนต์มีการเคลื่อนที่แบบเส้นตรงโดยกำหนดให้ค่าพิกัดความคลาดเคลื่อนของชิ้นงานมีค่าเท่ากันพบว่า หุ่นยนต์ใช้เวลาในการเคลื่อนที่ทั้งหมด 1,557.56 วินาที โดยมีจำนวนตำแหน่งการเคลื่อนที่ช่วงแรก 399 ตำแหน่ง มีจำนวนตำแหน่งช่วงสุดท้าย 1,405 ตำแหน่ง และมีจำนวนตำแหน่งทั้งหมด 1,804 ตำแหน่ง ซึ่งมีระยะห่างของปลายเครื่องมือขณะเคลื่อนย้ายชิ้นงานในช่วงสุดท้าย 0.02 มิลลิเมตร และการเคลื่อนที่แบบแบ่งช่วงสามารถลดจำนวนตำแหน่ง และเวลาที่ใช้สำหรับการเคลื่อนที่ทั้งหมดลงได้ร้อยละ 21.12 และ 25.53 ตามลำดับ เมื่อเทียบกับการเคลื่อนที่แบบเส้นตรง

ตารางที่ 6 แสดงค่าจำนวนตำแหน่ง และเวลาที่ใช้สำหรับการเคลื่อนที่ของตัวอย่างชิ้นงานชุดที่ 2

รายการ	ค่าที่ได้จากการเคลื่อนที่ของหุ่นยนต์ A และ B											
	ตัวอย่างที่ 1			ตัวอย่างที่ 2			ตัวอย่างที่ 3			ตัวอย่างที่ 4		
	$(R_{dif} = 0.50 \text{ mm})$			$(R_{dif} = 0.10 \text{ mm})$			$(R_{dif} = 0.06 \text{ mm})$			$(R_{dif} = 0.03 \text{ mm})$		
	Linear	Propose	%ลด	Linear	Propose	%ลด	Linear	Propose	%ลด	Linear	Propose	%ลด
1. เวลาเคลื่อนที่ (วินาที)	1,078.13	670.50	37.81	1,614.04	1,149.35	28.79	1,617.41	1,149.75	28.91	1,621.13	1,154.2	28.80
2. จำนวนช่วงการเคลื่อนที่												
ช่วงแรก (ช่วง)	399	15	96.24	504	17	96.63	504	15	97.02	504	17	96.63
3. จำนวนช่วงการเคลื่อนที่												
ช่วงสอง (ช่วง)	407	407	0.00	1,386	1,386	0.00	1,395	1,395	0.00	1,405	1,405	0.00
4. จำนวนช่วงการเคลื่อนที่												
ทั้งหมด (ช่วง)	806	422	47.64	1,890	1,403	25.77	1,899	1,410	25.75	1,909	1,422	25.51
5. ระยะห่างของปลาย												
เครื่องมือ (mm)	0.27	0.27	-	0.08	0.08	-	0.05	0.05	-	0.02	0.02	-

จากตัวอย่างชิ้นงานในตัวอย่างชุดที่ 2 ได้ผลการประมวลข้อมูลดังตารางที่ 6 พบว่า เมื่อกำหนดค่าพิกัดความคลาดเคลื่อนของชิ้นงานเพลลา (R_{dif}) มีค่าเท่ากับ 0.50 มิลลิเมตร หุ่นยนต์เคลื่อนที่แบบแบ่งช่วงใช้เวลาทั้งหมดสำหรับการเคลื่อนย้ายชิ้นงานเพลลา 670.50 วินาที โดยจำนวนตำแหน่งการเคลื่อนที่ในช่วงแรกได้แบ่งออกเป็น 15 ตำแหน่ง และจำนวนตำแหน่งการเคลื่อนที่ช่วงสุดท้ายเป็น 407 ตำแหน่ง ซึ่งมีจำนวนตำแหน่งตลอดเส้นทางเป็น 422 ตำแหน่ง มีระยะห่างของปลายเครื่องมือขณะเคลื่อนที่แตกต่างกันมากที่สุด 0.27 มิลลิเมตรสำหรับการเคลื่อนที่ในช่วงที่สองซึ่งสามารถทำให้หุ่นยนต์สามารถสวมประกอบชิ้นงานขณะเคลื่อนที่ได้ แต่สำหรับกรณีที่หุ่นยนต์มีการเคลื่อนที่แบบเส้นตรงซึ่งมีค่าพิกัดความคลาดเคลื่อนของชิ้นงานเพลลาเท่ากับการเคลื่อนที่แบบแบ่งช่วงพบว่า หุ่นยนต์ใช้เวลาในการเคลื่อนที่ทั้งหมด 1,078.13 วินาที มีจำนวนตำแหน่งการเคลื่อนที่ในช่วงแรก 399 ตำแหน่ง มีจำนวนตำแหน่งการเคลื่อนที่ในช่วงสุดท้าย 407 ตำแหน่ง และมีจำนวนตำแหน่งการเคลื่อนที่ทั้งหมด 806 ตำแหน่ง โดยมีระยะห่างของปลายเครื่องมือขณะเคลื่อนย้ายชิ้นงานช่วงสุดท้ายเท่ากับ 0.27 มิลลิเมตร ซึ่งจะเห็นได้ว่าการเคลื่อนที่แบบแบ่งช่วงสามารถลดจำนวนตำแหน่ง และเวลาทั้งหมดที่ใช้สำหรับการเคลื่อนที่ลงได้ร้อยละ 47.64 และ 37.81 ตามลำดับเมื่อเทียบกับการเคลื่อนที่แบบเส้นตรง เมื่อทำการลดค่าพิกัดความคลาดเคลื่อนของชิ้นงานเพลมาเป็นขนาด 0.10 มิลลิเมตรพบว่า หุ่นยนต์เคลื่อนที่แบบแบ่งช่วงใช้เวลาในการเคลื่อนย้ายชิ้นงานเพลลาทั้งหมด 1,149.35 วินาที มีจำนวนตำแหน่งการเคลื่อนที่ในช่วงแรก 17 ตำแหน่ง มีจำนวนตำแหน่งการเคลื่อนที่ช่วงสุดท้าย 1,386 ตำแหน่ง และมีจำนวนตำแหน่งการเคลื่อนที่ตลอดเส้นทางทั้งหมด 1,403 ตำแหน่ง โดยปลายเครื่องมือของหุ่นยนต์มีตำแหน่งการเคลื่อนที่แตกต่างกันมากที่สุด 0.08 มิลลิเมตร สำหรับการเคลื่อนที่ช่วงสุดท้าย แต่สำหรับกรณีหุ่นยนต์เคลื่อนที่แบบเส้นตรงซึ่งมีค่าพิกัดความคลาดเคลื่อนของชิ้นงานเพลลาเท่ากับการเคลื่อนที่แบบแบ่งช่วงพบว่า หุ่นยนต์ใช้เวลาในการเคลื่อนที่ทั้งหมด 1,614.04 วินาที โดยมีจำนวนตำแหน่งการเคลื่อนที่ในช่วงแรก 504 ตำแหน่ง มีจำนวนตำแหน่งการเคลื่อนที่ช่วงสุดท้าย 1,386 ตำแหน่ง และมีจำนวนตำแหน่งการเคลื่อนที่ทั้งหมด 1,890 ตำแหน่ง โดยมีระยะห่างของปลายเครื่องมือขณะเคลื่อนย้ายชิ้นงานช่วงสุดท้าย 0.08 มิลลิเมตร ซึ่งการเคลื่อนที่แบบแบ่งช่วงสามารถลดเวลาในการเคลื่อนที่ลงได้ร้อยละ 28.79 และลดจำนวนตำแหน่งการเคลื่อนที่ทั้งหมดลงร้อยละ 25.77 เมื่อเทียบกับเคลื่อนที่แบบเส้นตรง

สำหรับในกรณีลดค่าพิกัดความคลาดเคลื่อนของชิ้นงานเพลลเหลือ 0.03 มิลลิเมตรพบว่า หุ่นยนต์เคลื่อนที่แบบแบ่งช่วงใช้เวลาในการเคลื่อนย้ายชิ้นงานทั้งหมด 1,154.20 วินาที มีจำนวนตำแหน่งการเคลื่อนที่ในช่วงแรก 17 ตำแหน่ง จำนวนตำแหน่งการเคลื่อนที่ในช่วงสุดท้าย 1,405

ตำแหน่ง และมีจำนวนตำแหน่งการเคลื่อนที่ทั้งหมดตลอดเส้นทาง 1,422 ตำแหน่ง โดยปลาย
 เครื่องมือของหุ่นยนต์มีตำแหน่งการเคลื่อนที่แตกต่างกันมากที่สุดเป็นระยะ 0.02 มิลลิเมตรในการ
 เคลื่อนที่ช่วงที่สุดท้าย แต่สำหรับกรณีที่หุ่นยนต์มีการเคลื่อนที่แบบเส้นตรงโดยกำหนดค่าพิกัด
 ความคลาดเคลื่อนของชิ้นงานเพลามีค่าเท่ากับการเคลื่อนที่แบบแบ่งช่วงพบว่า หุ่นยนต์ใช้เวลาใน
 การเคลื่อนที่ทั้งหมด 1,621.13 วินาที โดยมีจำนวนตำแหน่งการเคลื่อนที่ในช่วงแรก 504 ตำแหน่ง
 มีจำนวนตำแหน่งการเคลื่อนที่ในช่วงสุดท้าย 1,405 ตำแหน่ง และมีจำนวนตำแหน่งการเคลื่อนที่
 ทั้งหมด 1,909 ตำแหน่ง ซึ่งมีระยะห่างของปลายเครื่องมือขณะเคลื่อนย้ายชิ้นงานในช่วงสุดท้าย
 0.02 มิลลิเมตร โดยการเคลื่อนที่แบบแบ่งช่วงสามารถลดเวลาทั้งหมดของการเคลื่อนที่ลงได้ร้อยละ
 28.80 และลดจำนวนตำแหน่งการเคลื่อนที่ทั้งหมดลงร้อยละ 25.51 เมื่อเทียบกับการเคลื่อนที่แบบ
 เส้นตรง

ตารางที่ 7 แสดงค่าจำนวนตำแหน่ง และเวลาที่ใช้สำหรับการเคลื่อนที่ของตัวอย่างชิ้นงานชุดที่ 3

รายการ	ค่าที่ได้จากการเคลื่อนที่ของหุ่นยนต์ A และ B											
	ตัวอย่างที่ 1			ตัวอย่างที่ 2			ตัวอย่างที่ 3			ตัวอย่างที่ 4		
	$(R_{dif} = 0.50 \text{ mm})$			$(R_{dif} = 0.10 \text{ mm})$			$(R_{dif} = 0.06 \text{ mm})$			$(R_{dif} = 0.03 \text{ mm})$		
	Linear	Propose	%ลด	Linear	Propose	%ลด	Linear	Propose	%ลด	Linear	Propose	%ลด
1. เวลาเคลื่อนที่ (วินาที)	1,323.89	949.28	28.30	1,598.38	1,151.78	27.94	1,601.75	1,155.8	27.84	1,637.49	1,201.19	26.64
2. จำนวนช่วงการเคลื่อนที่												
ช่วงแรก (ช่วง)	350	13	96.29	477	18	96.23	477	19	96.02	533	34	93.62
3. จำนวนช่วงการเคลื่อนที่												
ช่วงสอง (ช่วง)	916	916	0.00	1,386	1,386	0.00	1,395	1,395	0.00	1,405	1,405	0.00
4. จำนวนช่วงการเคลื่อนที่												
ทั้งหมด (ช่วง)	1,266	929	26.62	1,863	1,404	24.64	1,872	1,414	24.47	1,938	1,439	25.75
5. ระยะห่างของปลาย												
เครื่องมือ (mm)	0.31	0.31	-	0.08	0.08	-	0.05	0.05	-	0.02	0.02	-

จากตัวอย่างชิ้นงานในตัวอย่างชุดที่ 3 ได้ผลการประมวลข้อมูลดังตารางที่ 7 พบว่า เมื่อกำหนดค่าพิสัยความคลาดเคลื่อนของชิ้นงานเพลลา (R_{dif}) มีค่าเท่ากับ 0.50 มิลลิเมตร หุ่นยนต์เคลื่อนที่แบบแบ่งช่วงใช้เวลาทั้งหมดสำหรับการเคลื่อนย้ายชิ้นงานเพลลา 949.28 วินาที โดยจำนวนตำแหน่งการเคลื่อนที่ในช่วงแรกได้แบ่งออกเป็น 13 ตำแหน่ง และจำนวนตำแหน่งการเคลื่อนที่ช่วงสุดท้ายเป็น 916 ตำแหน่ง ซึ่งมีจำนวนตำแหน่งตลอดเส้นทางเป็น 929 ตำแหน่ง มีระยะห่างของปลายเครื่องมือขณะเคลื่อนที่แตกต่างกันมากที่สุด 0.31 มิลลิเมตรสำหรับการเคลื่อนที่ในช่วงที่สองซึ่งสามารถทำให้หุ่นยนต์สามารถสวมประกอบชิ้นงานขณะเคลื่อนที่ได้ แต่สำหรับกรณีที่หุ่นยนต์มีการเคลื่อนที่แบบเส้นตรงซึ่งมีค่าพิสัยความคลาดเคลื่อนของชิ้นงานเพลลาเท่ากับการเคลื่อนที่แบบแบ่งช่วงพบว่า หุ่นยนต์ใช้เวลาในการเคลื่อนที่ทั้งหมด 1,323.89 วินาที มีจำนวนตำแหน่งการเคลื่อนที่ในช่วงแรก 350 ตำแหน่ง มีจำนวนตำแหน่งการเคลื่อนที่ช่วงสุดท้าย 916 ตำแหน่ง และมีจำนวนตำแหน่งการเคลื่อนที่ทั้งหมด 1,266 ตำแหน่ง โดยมีระยะห่างของปลายเครื่องมือขณะเคลื่อนย้ายชิ้นงานช่วงสุดท้ายเท่ากับ 0.31 มิลลิเมตร ซึ่งจะเห็นได้ว่าการเคลื่อนที่แบบแบ่งช่วงสามารถลดจำนวนตำแหน่ง และเวลาทั้งหมดที่ใช้สำหรับการเคลื่อนที่ลงได้ร้อยละ 26.62 และ 28.30 ตามลำดับเมื่อเทียบกับการเคลื่อนที่แบบเส้นตรง เมื่อทำการลดค่าพิสัยความคลาดเคลื่อนของชิ้นงานเพลมาเป็นขนาด 0.10 มิลลิเมตรพบว่า หุ่นยนต์เคลื่อนที่แบบแบ่งช่วงใช้เวลาในการเคลื่อนย้ายชิ้นงานเพลลาทั้งหมด 1,151.78 วินาที มีจำนวนตำแหน่งการเคลื่อนที่ในช่วงแรก 18 ตำแหน่ง มีจำนวนตำแหน่งการเคลื่อนที่ช่วงสุดท้าย 1,386 ตำแหน่ง และมีจำนวนตำแหน่งการเคลื่อนที่ตลอดเส้นทางทั้งหมด 1,404 ตำแหน่ง โดยปลายเครื่องมือของหุ่นยนต์มีตำแหน่งการเคลื่อนที่แตกต่างกันมากที่สุด 0.08 มิลลิเมตร สำหรับการเคลื่อนที่ช่วงสุดท้าย แต่สำหรับกรณีหุ่นยนต์เคลื่อนที่แบบเส้นตรงซึ่งมีค่าพิสัยความคลาดเคลื่อนของชิ้นงานเพลลาเท่ากับการเคลื่อนที่แบบแบ่งช่วงพบว่า หุ่นยนต์ใช้เวลาในการเคลื่อนที่ทั้งหมด 1,598.38 วินาที โดยมีจำนวนตำแหน่งการเคลื่อนที่ในช่วงแรก 477 ตำแหน่ง มีจำนวนตำแหน่งการเคลื่อนที่ช่วงสุดท้าย 1,386 ตำแหน่ง และมีจำนวนตำแหน่งการเคลื่อนที่ทั้งหมด 1,863 ตำแหน่ง โดยมีระยะห่างของปลายเครื่องมือขณะเคลื่อนย้ายชิ้นงานช่วงสุดท้าย 0.08 มิลลิเมตร ซึ่งการเคลื่อนที่แบบแบ่งช่วงสามารถลดเวลาในการเคลื่อนที่ทั้งหมดลงได้ร้อยละ 27.94 และลดจำนวนตำแหน่งทั้งหมดของการเคลื่อนที่ลงได้ร้อยละ 24.64 เมื่อเทียบกับการเคลื่อนที่แบบเส้นตรง

สำหรับในกรณีลดค่าพิสัยความคลาดเคลื่อนของชิ้นงานเพลเหลือ 0.03 มิลลิเมตรพบว่า หุ่นยนต์เคลื่อนที่แบบแบ่งช่วงใช้เวลาในการเคลื่อนย้ายชิ้นงานทั้งหมด 1,201.19 วินาที มีจำนวนตำแหน่งการเคลื่อนที่ในช่วงแรก 34 ตำแหน่ง จำนวนตำแหน่งการเคลื่อนที่ในช่วงสุดท้าย 1,405

ตำแหน่ง และมีจำนวนตำแหน่งการเคลื่อนที่ทั้งหมดตลอดเส้นทาง 1,439 ตำแหน่ง โดยปลายเครื่องมือของหุ่นยนต์มีตำแหน่งการเคลื่อนที่แตกต่างกันมากที่สุดเป็นระยะ 0.02 มิลลิเมตรในการเคลื่อนที่ช่วงที่สุดท้าย แต่สำหรับกรณีที่หุ่นยนต์มีการเคลื่อนที่แบบเส้นตรงโดยกำหนดค่าพิกัดความคลาดเคลื่อนของชิ้นงานเพลามีค่าเท่ากับการเคลื่อนที่แบบแบ่งช่วงพบว่า หุ่นยนต์ใช้เวลาในการเคลื่อนที่ทั้งหมด 1,637.49 วินาที โดยมีจำนวนตำแหน่งการเคลื่อนที่ในช่วงแรก 533 ตำแหน่ง มีจำนวนตำแหน่งการเคลื่อนที่ในช่วงสุดท้าย 1,405 ตำแหน่ง และมีจำนวนตำแหน่งการเคลื่อนที่ทั้งหมด 1,938 ตำแหน่ง ซึ่งมีระยะห่างของปลายเครื่องมือขณะเคลื่อนย้ายชิ้นงานในช่วงสุดท้าย 0.02 มิลลิเมตร โดยการเคลื่อนที่แบบแบ่งช่วงสามารถลดเวลาทั้งหมดของการเคลื่อนที่ลงได้ร้อยละ 26.64 และลดจำนวนตำแหน่งการเคลื่อนที่ทั้งหมดลงได้ร้อยละ 25.75 เมื่อเทียบกับการเคลื่อนที่แบบเส้นตรง

2. นำชิ้นงานมาสวมประกอบด้วยหุ่นยนต์ A และ B ณ ตำแหน่งบริเวณปลายด้านหลังของชิ้นงานเพลาทรงกระบอก โดยสามารถทำการกำหนดตำแหน่งการเคลื่อนที่ได้ดังนี้

1. ตำแหน่งการเคลื่อนที่หุ่นยนต์ A คือ
$$\begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} = \begin{bmatrix} (X_1, Y_1, Z_1) \\ (X_2, Y_2, Z_2) \\ (X_3, Y_3, Z_3) \end{bmatrix} = \begin{bmatrix} (20, 25, 0) \\ (-10, 35, 25) \\ (-20, 35, 0) \end{bmatrix}$$
2. ตำแหน่งการเคลื่อนที่หุ่นยนต์ B คือ
$$\begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} = \begin{bmatrix} (X_1, Y_1, Z_1) \\ (X_2, Y_2, Z_2) \\ (X_3, Y_3, Z_3) \end{bmatrix} = \begin{bmatrix} (-20, 25, 0) \\ (-19, 35, 25) \\ (25, 35, 0) \end{bmatrix}$$
3. ชิ้นงานเพลาทรงกระบอกยาว 14 mm มีรัศมีวงในยาว 4.0 mm
4. ชิ้นงานเพลาวงแหวนยาว 4 mm มีรัศมีวงในยาว 4.5 mm

เมื่อทำการประมวลผลจากโปรแกรมคอมพิวเตอร์ตามข้อกำหนด และวิธีการที่ได้นำเสนอพบว่า ผลการดำเนินงานของหุ่นยนต์ทั้งสองตัวสามารถนำมาเสนอได้ดังตารางที่ 8 ได้ดังนี้

ตารางที่ 8 แสดงค่าจำนวนตำแหน่ง และเวลาที่ใช้ในการเคลื่อนที่ของหุ่นยนต์ เมื่อต้องทำประกอบชิ้นงานเพลาทังสองบริเวณปลายด้านหลังของชิ้นงานเพลาทรงกระบอก

รายการ	ค่าที่ได้จากการเคลื่อนที่ของหุ่นยนต์ A และ B ($R_{dif} = 0.50 \text{ mm}$)		
	Linear	Propose	%ลด
1. เวลาเคลื่อนที่ (วินาที)	804.06	617.11	23.25
2. จำนวนช่วงการเคลื่อนที่ช่วงแรก (ช่วง)	119	6	94.96
3. จำนวนช่วงการเคลื่อนที่ช่วงสอง (ช่วง)	384	384	-
4. จำนวนช่วงการเคลื่อนที่ทั้งหมด (ช่วง)	503	390	22.47
5. ระยะห่างของปลายเครื่องมือ (mm)	0.33	0.33	-

จากตารางที่ 8 พบว่าเมื่อกำหนดให้หุ่นยนต์ทำการสวมประกอบชิ้นงานเพลาทังสองบริเวณปลายด้านหลังชิ้นงานเพลาทรงกระบอก และกำหนดค่าพิกัดความคลาดเคลื่อนของชิ้นงานเพลา (R_{dif}) มีค่าเท่ากับ 0.50 มิลลิเมตร หุ่นยนต์เคลื่อนที่แบบแบ่งช่วงใช้เวลาทั้งหมดสำหรับการเคลื่อนย้ายชิ้นงานเพลา 617.11 วินาที โดยจำนวนตำแหน่งการเคลื่อนที่ในช่วงแรกได้แบ่งออกเป็น 6 ตำแหน่ง และจำนวนตำแหน่งการเคลื่อนที่ช่วงสุดท้ายเป็น 384 ตำแหน่ง ซึ่งมีจำนวนตำแหน่งตลอดเส้นทางเป็น 390 ตำแหน่ง มีระยะห่างของปลายเครื่องมือขณะเคลื่อนที่ที่แตกต่างกันมากที่สุด 0.33 มิลลิเมตรสำหรับการเคลื่อนที่ในช่วงที่สองซึ่งสามารถทำให้หุ่นยนต์สามารถทำการสวมประกอบชิ้นงานขณะทำการเคลื่อนที่ได้ แต่สำหรับกรณีที่หุ่นยนต์มีการเคลื่อนที่แบบเส้นตรงซึ่งมีค่าพิกัดความคลาดเคลื่อนของชิ้นงานเพลาท่ำกับการเคลื่อนที่แบบแบ่งช่วงพบว่า หุ่นยนต์ใช้เวลาในการเคลื่อนที่ทั้งหมด 804.06 วินาที มีจำนวนตำแหน่งการเคลื่อนที่ในช่วงแรก 119 ตำแหน่ง มีจำนวนตำแหน่งการเคลื่อนที่ช่วงสุดท้าย 384 ตำแหน่ง และมีจำนวนตำแหน่งการเคลื่อนที่ทั้งหมด 504 ตำแหน่ง โดยมีระยะห่างของปลายเครื่องมือขณะเคลื่อนย้ายชิ้นงานช่วงสุดท้ายเท่ากับ 0.33 มิลลิเมตร ซึ่งจะเห็นได้ว่าการเคลื่อนที่แบบแบ่งช่วงสามารถลดจำนวนตำแหน่ง และเวลาทั้งหมดที่ใช้สำหรับการเคลื่อนที่ลงได้ร้อยละ 22.47 และ 23.25 ตามลำดับเมื่อเทียบกับการเคลื่อนที่แบบเส้นตรง

3. นำชิ้นงานมาสวมประกอบด้วยหุ่นยนต์ A และ B ณ ตำแหน่งบริเวณปลายด้านหน้า ชิ้นงานเพลาทรงกระบอก โดยสามารถทำการกำหนดตำแหน่งการเคลื่อนที่ได้ดังนี้

1. ตำแหน่งการเคลื่อนที่หุ่นยนต์ A คือ
$$\begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} = \begin{bmatrix} (X_1, Y_1, Z_1) \\ (X_2, Y_2, Z_2) \\ (X_3, Y_3, Z_3) \end{bmatrix} = \begin{bmatrix} (25, 25, 0) \\ (-9, 35, 25) \\ (-25, 35, 0) \end{bmatrix}$$
2. ตำแหน่งการเคลื่อนที่หุ่นยนต์ B คือ
$$\begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} = \begin{bmatrix} (X_1, Y_1, Z_1) \\ (X_2, Y_2, Z_2) \\ (X_3, Y_3, Z_3) \end{bmatrix} = \begin{bmatrix} (-25, 25, 0) \\ (0, 35, 25) \\ (20, 35, 0) \end{bmatrix}$$
3. ชิ้นงานเพลาทรงกระบอกยาว 14 mm มีรัศมีวงในยาว 3.0 mm
4. ชิ้นงานเพลาวงแหวนยาว 4 mm มีรัศมีวงในยาว 3.5 mm

เมื่อทำการประมวลผลจากโปรแกรมคอมพิวเตอร์ตามข้อกำหนด และวิธีการที่ได้นำเสนอ พบว่า ผลการดำเนินงานของหุ่นยนต์ทั้งสองตัวสามารถนำมาเสนอได้ดังตารางที่ 9 ได้ดังนี้

ตารางที่ 9 แสดงค่าจำนวนตำแหน่ง และเวลาที่ใช้ในการเคลื่อนที่ของหุ่นยนต์ เมื่อต้องทำประกอบ ชิ้นงานเพลาทรงกระบอกทั้งสองบริเวณปลายด้านหน้าชิ้นงานเพลาทรงกระบอก

รายการ	ค่าที่ได้จากการเคลื่อนที่ของหุ่นยนต์ A และ B ($R_{dif} = 0.50 \text{ mm}$)		
	Linear	Propose	%ลด
1. เวลาเคลื่อนที่ (วินาที)	330.11	232.25	29.64
2. จำนวนช่วงการเคลื่อนที่ช่วงแรก (ช่วง)	12	9	25.00
3. จำนวนช่วงการเคลื่อนที่ช่วงสอง (ช่วง)	28	28	-
4. จำนวนช่วงการเคลื่อนที่ทั้งหมด (ช่วง)	40	37	7.50
5. ระยะห่างของปลายเครื่องมือ (mm)	0.49	0.49	-

จากตารางที่ 9 พบว่าเมื่อกำหนดให้หุ่นยนต์ทำการสวมประกอบชิ้นงานเพลาทรงกระบอกทั้งสอง บริเวณปลายด้านหน้าชิ้นงานเพลาทรงกระบอก และกำหนดค่าพิกัดความคลาดเคลื่อนของชิ้นงาน เปลา (R_{dif}) มีค่าเท่ากับ 0.50 มิลลิเมตร หุ่นยนต์เคลื่อนที่แบบแบ่งช่วงใช้เวลาทั้งหมดสำหรับการ เคลื่อนย้ายชิ้นงานเพลา 232.25 วินาที โดยจำนวนตำแหน่งการเคลื่อนที่ในช่วงแรกได้แบ่งออกเป็น

9 ตำแหน่ง และจำนวนตำแหน่งการเคลื่อนที่ช่วงสุดท้ายเป็น 28 ตำแหน่ง ซึ่งมีจำนวนตำแหน่งตลอดเส้นทางเป็น 37 ตำแหน่ง มีระยะห่างของปลายเครื่องมือขณะเคลื่อนที่แตกต่างกันมากที่สุด 0.49 มิลลิเมตรสำหรับการเคลื่อนที่ในช่วงที่สองซึ่งสามารถทำให้หุ่นยนต์สามารถทำการสวมประกอบชิ้นงานขณะทำการเคลื่อนที่ได้ แต่สำหรับกรณีที่หุ่นยนต์มีการเคลื่อนที่แบบเส้นตรงซึ่งมีค่าพิสัยความคลาดเคลื่อนของชิ้นงานเพลาท่ำกับการเคลื่อนที่แบบแบ่งช่วงพบว่า หุ่นยนต์ใช้เวลาในการเคลื่อนที่ทั้งหมด 330.11 วินาที มีจำนวนตำแหน่งการเคลื่อนที่ในช่วงแรก 12 ตำแหน่ง มีจำนวนตำแหน่งการเคลื่อนที่ช่วงสุดท้าย 28 ตำแหน่ง และมีจำนวนตำแหน่งการเคลื่อนที่ทั้งหมด 40 ตำแหน่ง โดยมีระยะห่างของปลายเครื่องมือขณะเคลื่อนย้ายชิ้นงานช่วงสุดท้ายเท่ากับ 0.49 มิลลิเมตร ซึ่งจะเห็นได้ว่าการเคลื่อนที่แบบแบ่งช่วงสามารถลดจำนวนตำแหน่ง และเวลาทั้งหมดที่ใช้สำหรับการเคลื่อนที่ลงได้ร้อยละ 7.50 และ 29.64 ตามลำดับเมื่อเทียบกับการเคลื่อนที่แบบเส้นตรง

สรุปและข้อเสนอแนะ

สรุป

จากการศึกษาการวางแผนการเคลื่อนที่หุ่นยนต์ 2 ตัวมาทำงานร่วมกัน กรณีทำการประกอบชิ้นงานขณะเคลื่อนย้ายชิ้นงานที่เป็นทรงกระบอก และวงแหวน ซึ่งผลของการประมวลผลการทำงานของหุ่นยนต์สามารถนำมาสรุปเป็นหัวข้อหลักๆ ได้ดังนี้

1. เวลาสำหรับการเคลื่อนที่ จากการประมวลผลพบว่าหุ่นยนต์ที่มีการเคลื่อนที่แบบแบ่งช่วงเคลื่อนที่ได้เร็วกว่าการเคลื่อนที่แบบเส้นตรง โดยการเคลื่อนที่แบบแบ่งช่วงใช้เวลาการเคลื่อนที่ลดลงจากการเคลื่อนที่แบบเส้นตรงโดยเฉลี่ย 25.97 เปอร์เซ็นต์ ซึ่งเวลาที่ใช้สำหรับการเคลื่อนที่แบบแบ่งช่วงนั้นขึ้นอยู่กับค่าพิกัดความคลาดเคลื่อนของชิ้นงานเพลลา โดยเวลาการเคลื่อนที่ของหุ่นยนต์จะมีความสัมพันธ์กับค่าพิกัดความคลาดเคลื่อนของชิ้นงานเพลลาเป็นแบบผกผันกัน เช่น ถ้าค่าพิกัดความคลาดเคลื่อนของชิ้นงานเพลลามีค่าน้อยลงจะส่งผลให้หุ่นยนต์ใช้เวลาในการเคลื่อนที่มากขึ้นตามไปด้วย แต่สำหรับหุ่นยนต์ที่มีการเคลื่อนที่แบบเส้นตรงนั้นพบว่า เวลาการเคลื่อนที่ของหุ่นยนต์นั้นขึ้นอยู่กับค่าความผิดพลาดของการเบี่ยงเบนไปจากการเคลื่อนที่เป็นเส้นตรง ซึ่งเวลาการเคลื่อนที่ของหุ่นยนต์จะมีความสัมพันธ์กับค่าความผิดพลาดของการเบี่ยงเบนไปจากเส้นตรงเป็นแบบแปรผันตามกัน เช่น ถ้าค่าความผิดพลาดของการเบี่ยงเบนไปจากเส้นตรงมีค่ามากขึ้น หุ่นยนต์จะเคลื่อนที่ได้รวดเร็วขึ้นตามไปด้วย

2. การชนกันของชิ้นงานเพลลา จะเห็นได้ว่าการเคลื่อนที่ของหุ่นยนต์ A และ B เมื่อมีการเคลื่อนที่แบบเส้นตรงไปยังตำแหน่งที่เตรียมพร้อมประกอบชิ้นงาน จะมีบางตำแหน่งที่เกิดการชนกันของชิ้นงานเพลลาขึ้นทำให้ชิ้นงานไม่สามารถเคลื่อนที่ไปยังตำแหน่งเป้าหมายได้ ซึ่งได้แสดงไว้ในแผนภาพผนวกที่ 12 แต่สำหรับการเคลื่อนที่แบบแบ่งช่วงการเคลื่อนที่ในช่วงแรกจะไม่เกิดการชนกันของชิ้นงาน เนื่องจากมีการตรวจสอบการชนกันของชิ้นงานก่อนมีการเคลื่อนที่ทำให้ไม่เกิดการชนกันของชิ้นงานระหว่างมีการเคลื่อนที่

3. จำนวนช่วงการเคลื่อนที่ พบว่าการเคลื่อนที่แบบเส้นตรงจะมีจำนวนช่วงการเคลื่อนที่มากกว่าการเคลื่อนที่แบบแบ่งช่วง โดยการเคลื่อนที่แบบแบ่งช่วงมีจำนวนช่วงการเคลื่อนที่ลดลงโดยเฉลี่ย 24.11 เปอร์เซ็นต์เมื่อเทียบกับการเคลื่อนที่แบบเส้นตรง เนื่องจากการเคลื่อนที่แบบแบ่ง

ช่วงไม่ต้องการให้หุ่นยนต์เคลื่อนที่เป็นเส้นตรงมากนักแต่ต้องการความรวดเร็วในการเคลื่อนที่มากกว่า ซึ่งทำให้หุ่นยนต์สามารถเคลื่อนที่ไปยังตำแหน่งพร้อมสวมประกอบชิ้นงานเพล่าได้โดยไม่เกิดการชนของชิ้นงานเพล่าก็พอสำหรับการเคลื่อนที่ในช่วงแรก และสามารถทำการสวมประกอบชิ้นงานเพล่าเข้าด้วยกันได้สำหรับการเคลื่อนที่ในช่วงที่สอง แต่สำหรับการเคลื่อนที่แบบเส้นตรงนั้นต้องมีการกำหนดจำนวนตำแหน่งการเคลื่อนที่เป็นจำนวนมาก เนื่องจากหุ่นยนต์ต้องเคลื่อนที่เป็นเส้นตรงโดยที่ปลายของเครื่องมือต้องมีระยะการแกว่งได้ไม่เกินค่าความผิดพลาดของการเบี่ยงเบนไปจากเส้นตรงที่สามารถยอมรับได้ จึงทำให้การเคลื่อนที่แบบแบ่งช่วงมีจำนวนช่วงการเคลื่อนที่น้อยกว่าการเคลื่อนที่แบบเส้นตรง

จากผลการศึกษาพบว่าการวางแผนการเคลื่อนที่แบบแบ่งช่วงเหมาะกับงานที่ต้องการความยืดหยุ่น และต้องการความรวดเร็วในการปฏิบัติงาน ซึ่งใช้เวลาสำหรับการกำหนดตำแหน่งการเคลื่อนที่ของหุ่นยนต์เพียงเล็กน้อยเท่านั้น เนื่องจากวิธีการวางแผนแบบแบ่งช่วงนั้นจะมีการแบ่งช่วงการเคลื่อนที่ออกเป็นสองช่วงเท่านั้น โดยช่วงแรกเป็นการนำชิ้นงานไปไว้ในตำแหน่งเตรียมพร้อมการสวมประกอบชิ้นงานเพล่า และช่วงสุดท้ายเป็นช่วงของการสวมประกอบชิ้นงานขณะทำการเคลื่อนที่ พร้อมกับนำชิ้นงานไปไว้ในตำแหน่งที่กำหนดไว้จึงทำให้เสียเวลาในการกำหนดตำแหน่งการเคลื่อนที่เพียงเล็กน้อยเท่านั้น แต่ถ้าเป็นการเคลื่อนที่แบบเส้นตรงพบว่าต้องมานั่งคำนวณหาตำแหน่งที่มีการชนกันของชิ้นงานเพล่าที่ละตำแหน่ง ซึ่งถ้าพบว่ามี การชนกันของชิ้นงานก็ต้องมีการขยับตำแหน่งไปเรื่อยๆ จึงทำให้เสียเวลาในการกำหนดตำแหน่ง หรือเส้นทางการเคลื่อนที่มากกว่าการเคลื่อนที่แบบแบ่งตามตำแหน่งการชนในวิธีแรกได้ดีกว่า

ข้อเสนอแนะ

ข้อเสนอแนะสำหรับการทำวิจัยในครั้งนี้ คือ

1. เนื่องจากการทำวิจัยในครั้งนี้ผู้วิจัยมีระยะเวลาในการทำวิจัยจำกัด จึงไม่สามารถคำนวณหาตำแหน่งการเคลื่อนที่ที่ดีที่สุดซึ่งทำให้หุ่นยนต์สามารถทำการประกอบชิ้นงานขณะทำการเคลื่อนที่ร่วมกันได้โดยใช้เวลาการเคลื่อนที่น้อยที่สุด ดังนั้นผู้วิจัยจึงควรมีการศึกษาถึงวิธีการกำหนดตำแหน่งการเคลื่อนที่ทั้ง 3 ช่วงให้เป็นค่าที่ดีที่สุด เพื่อให้หุ่นยนต์สามารถทำการเคลื่อนที่ได้รวดเร็วที่สุด
2. เนื่องจากการเคลื่อนที่ในแบบปัจจุบันเป็นการเคลื่อนที่แบบเส้นตรง ซึ่งไม่ได้มีการวิจัยถึงค่าความผิดพลาดจากการแกว่งของปลายเครื่องมือที่เหมาะสม ซึ่งในบางครั้งผู้วิจัยไม่สามารถทำการหนดถึงค่าความผิดพลาดที่เกิดจากการเคลื่อนที่เป็นเส้นตรงได้ ดังนั้นผู้วิจัยควรมีศึกษาค่าความผิดพลาดจากการแกว่งของปลายเครื่องมือที่เหมาะสม ซึ่งเป็นค่าที่สามารถยอมรับได้ในการทำงาน ไม่ว่าจะเป็นการทำงานในด้านการผลิต และการเคลื่อนย้ายชิ้นงาน
3. เนื่องจากการวิจัยในครั้งนี้ทางผู้วิจัยได้ทำการวิจัยเฉพาะในส่วนที่เป็นซอฟต์แวร์ (Software) เท่านั้น ไม่ได้มีการนำทดลองใช้งานจริงกับหุ่นยนต์จริงๆ ที่นำมาใช้งานกันอยู่ในปัจจุบันนี้ ดังนั้นการวิจัยครั้งต่อไปจึงควรมีการทดลองใช้งานกับหุ่นยนต์จริงๆ ที่มีการใช้งานกันอยู่ตามโรงงานอุตสาหกรรมต่างๆ มาทำการศึกษาทดลอง เพื่อจะได้นำวิธีการที่ได้จากการทดลองนำไปใช้งานกับหุ่นยนต์ตามโรงงานอุตสาหกรรมได้จริง และเพื่อจะได้เป็นประโยชน์กับอุตสาหกรรมภายในประเทศต่อไป

เอกสารและสิ่งอ้างอิง

สถาพร ลักษณะเจริญ. 2545. **วิศวกรรมหุ่นยนต์**. สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), กรุงเทพมหานคร.

Honghai, L and J. Dai. 2003. An approach to carton-folding trajectory planning using dual robotic fingers. **Robots and Autonomous System** 42: 47-63.

Jaydev P, D. 1998. **Motion Planning and Control of Cooperative Robotic Systems**. Ph.D.Thesis, University of Pennsylvania..

Kang G.Shin and Q. Zheng. 1989. Minimum time trajectory planning for dual robot system. **Decision and Control** December 1989: 2506-2511.

Oussama, k., S. Quinlan. And D. Williams. 1997. Robot planning and control. **Robots and Autonomous System** 21: 249-261.

Yoshihito, K. and J.C. Latombe. 1994. On Multi-Arm Manipulation Planning. **Robots and Autonomous System** May 1994 (2): 945-952.

Multi, S. and P.Isto. 2006. Multi-Robot motion planning be incremental coordination. **Intelligent Robots and Systems** October 2006: 5960-5963.

Zhenwang,Y. and K.Gupta. 2005. Path planning with general end-effector constraints. **Robotics and Autonomous Systems** October 2005 (55): 316–327.

Ajmal, D.A. and R.Babu and K.Varghese. 2002. Offline path planning of cooperative manipulators using co-evolutionary genetic algorithm. **19th International Symposium on Automation & Robotics in Construction** September 2002: 415-424.

Ting, Y. and W.I.Lei and H.C.Jar. 2002. A path planning algorithm for industrial robots.

Computer & Industrial Engineering 42: 299–308.

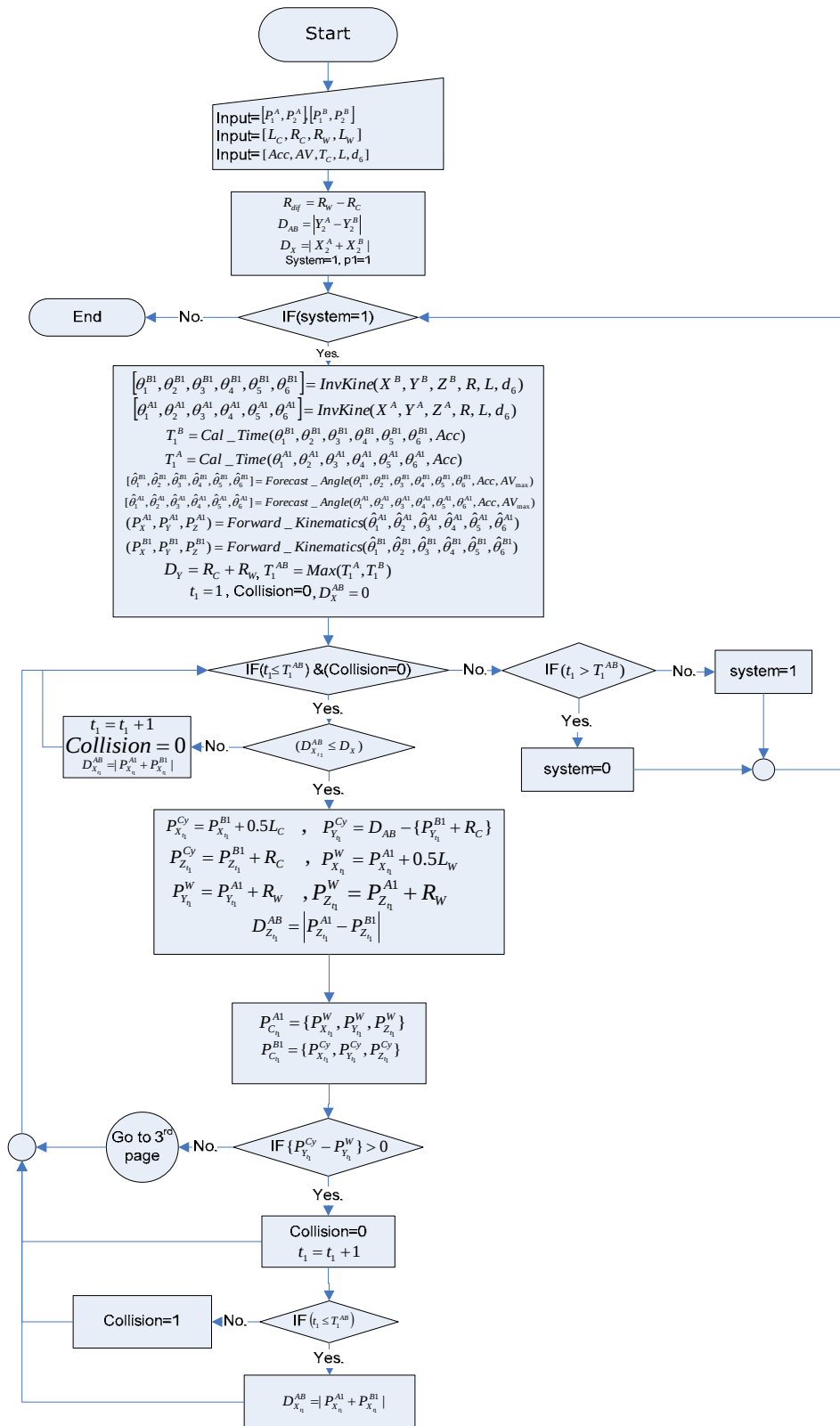
Niku, S.B. 2001. **Introduction to Robotics Analysis, Systems, Application.** Prentice Hall,
New Jersey.

ภาคผนวก

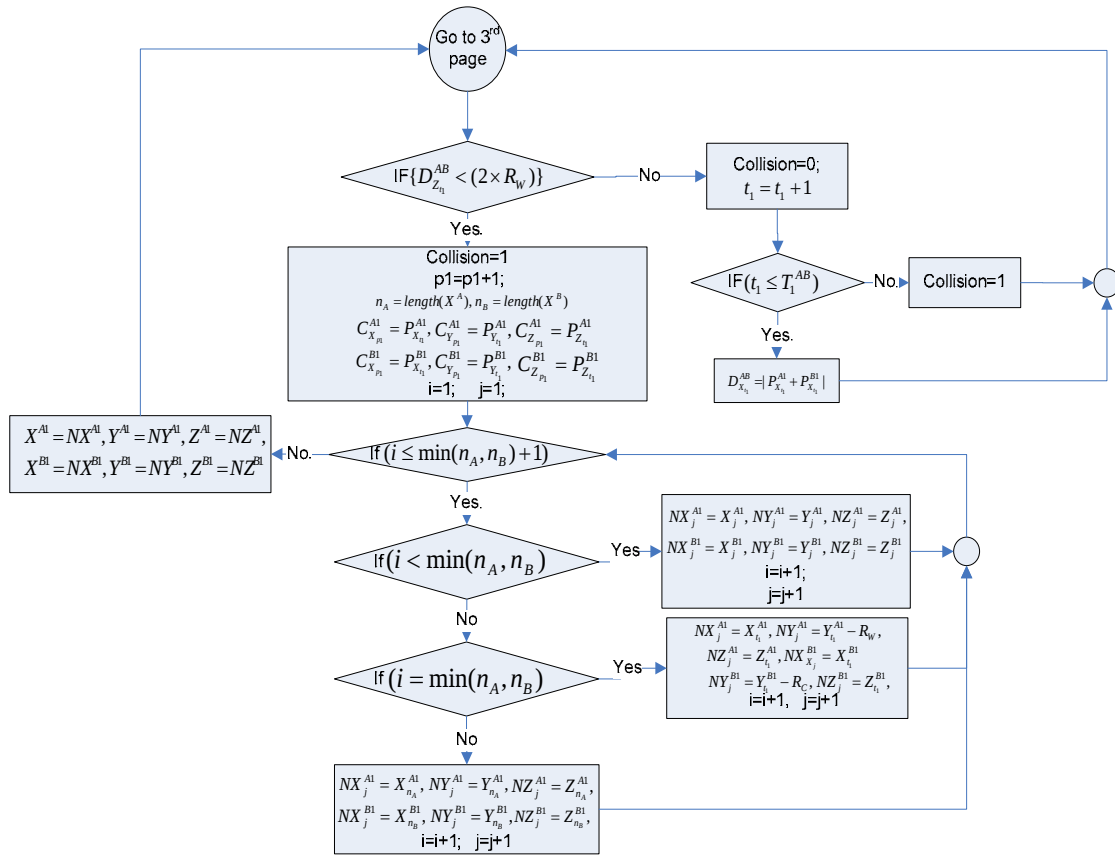
รายละเอียดของแบบจำลองการทำงานของโปรแกรม

ผู้วิจัยได้ทำการรวมแบบจำลอง หรือขั้นตอนการทำงานของโปรแกรมในงานวิจัยในครั้งนี้ เพื่อแสดงการทำงานของโปรแกรม Matlab7 ดังนี้

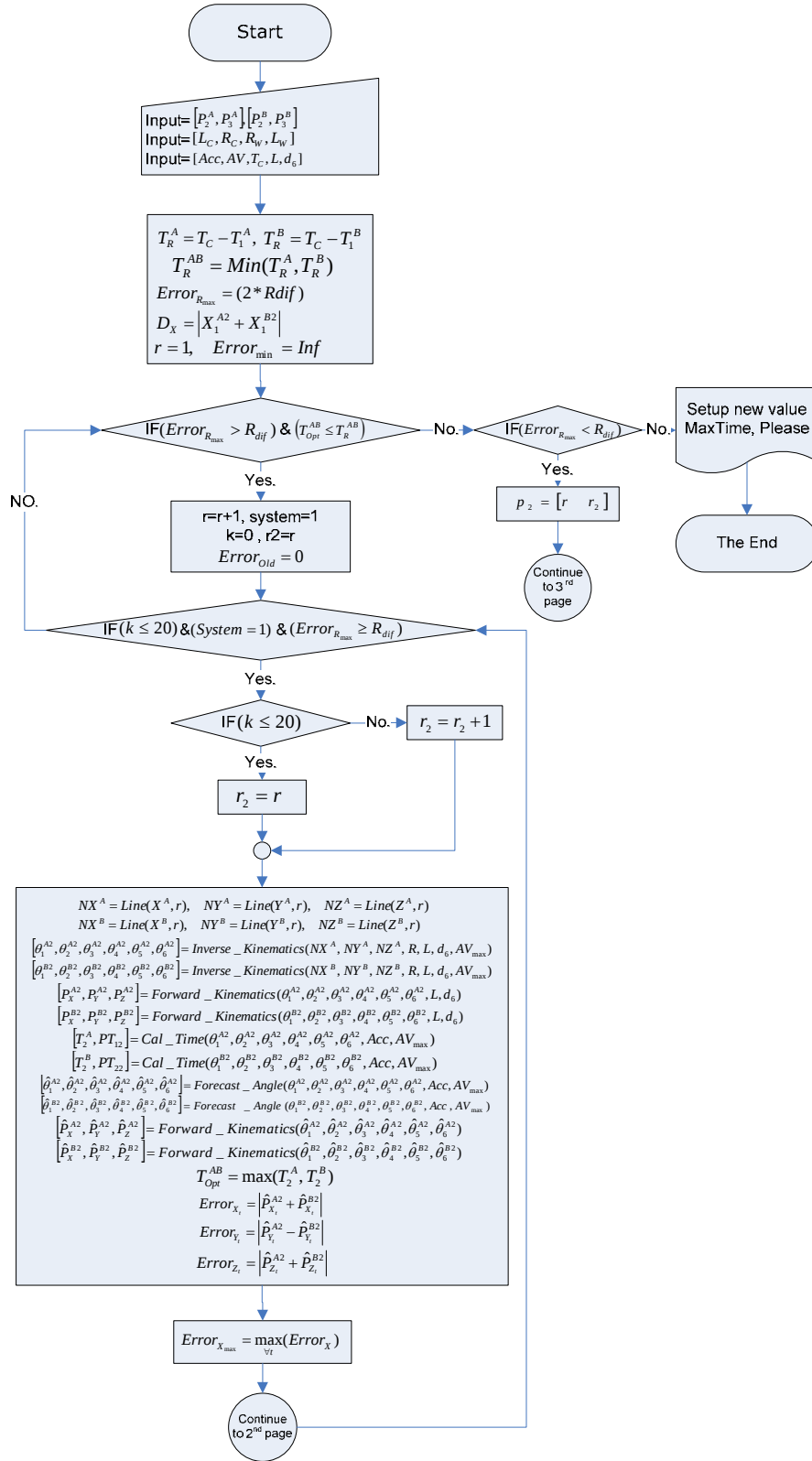
1. ขั้นตอนการทำงานของโปรแกรมในการคำนวณหาจำนวนตำแหน่งการเคลื่อนที่ของหุ่นยนต์ในช่วงแรก ได้แสดงไว้ในภาพผนวกที่ 1 และ 2
2. ขั้นตอนการทำงานของโปรแกรมในการคำนวณหาจำนวนตำแหน่งการเคลื่อนที่ของหุ่นยนต์ในช่วงสุดท้าย ได้แสดงไว้ในภาพผนวกที่ 3 และ 8



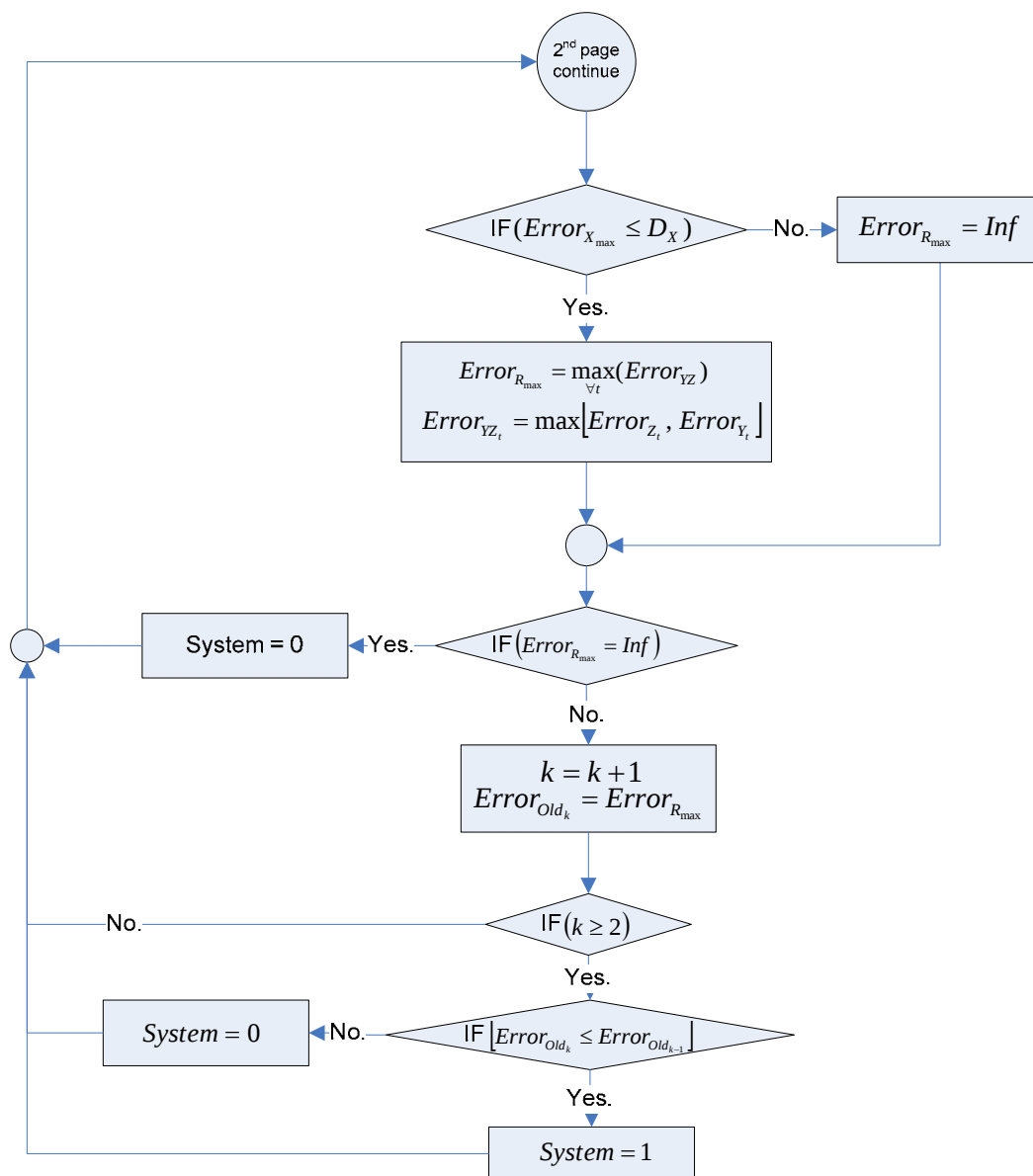
ภาพผนวกที่ 1 Flow Chart แสดงการหาค่าจำนวนตำแหน่งการเคลื่อนที่ช่วงที่ 1



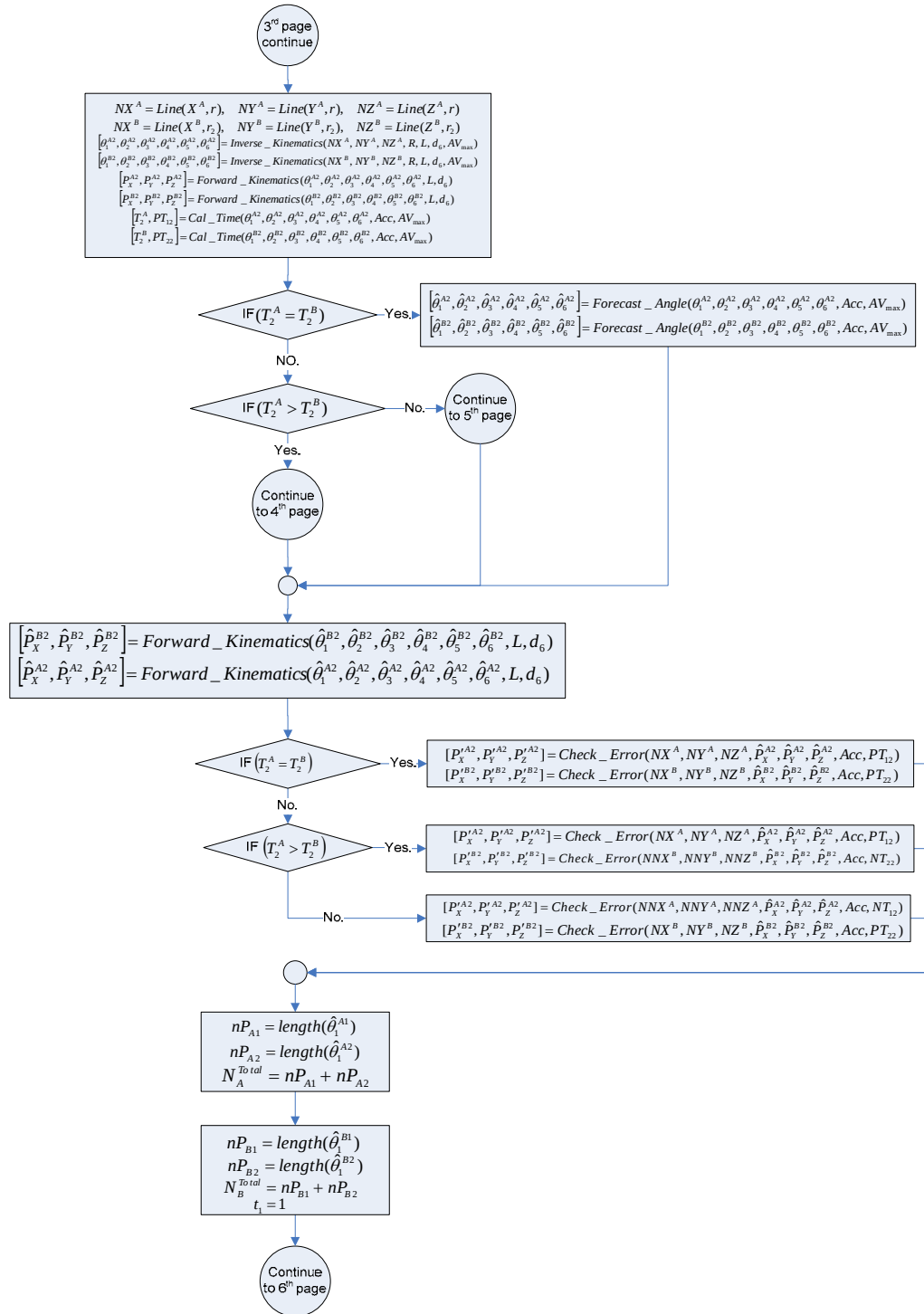
ภาพผนวกที่ 2 Flow Chart แสดงการหาค่าจำนวนตำแหน่งการเคลื่อนที่ช่วงที่ 1 (ต่อ)



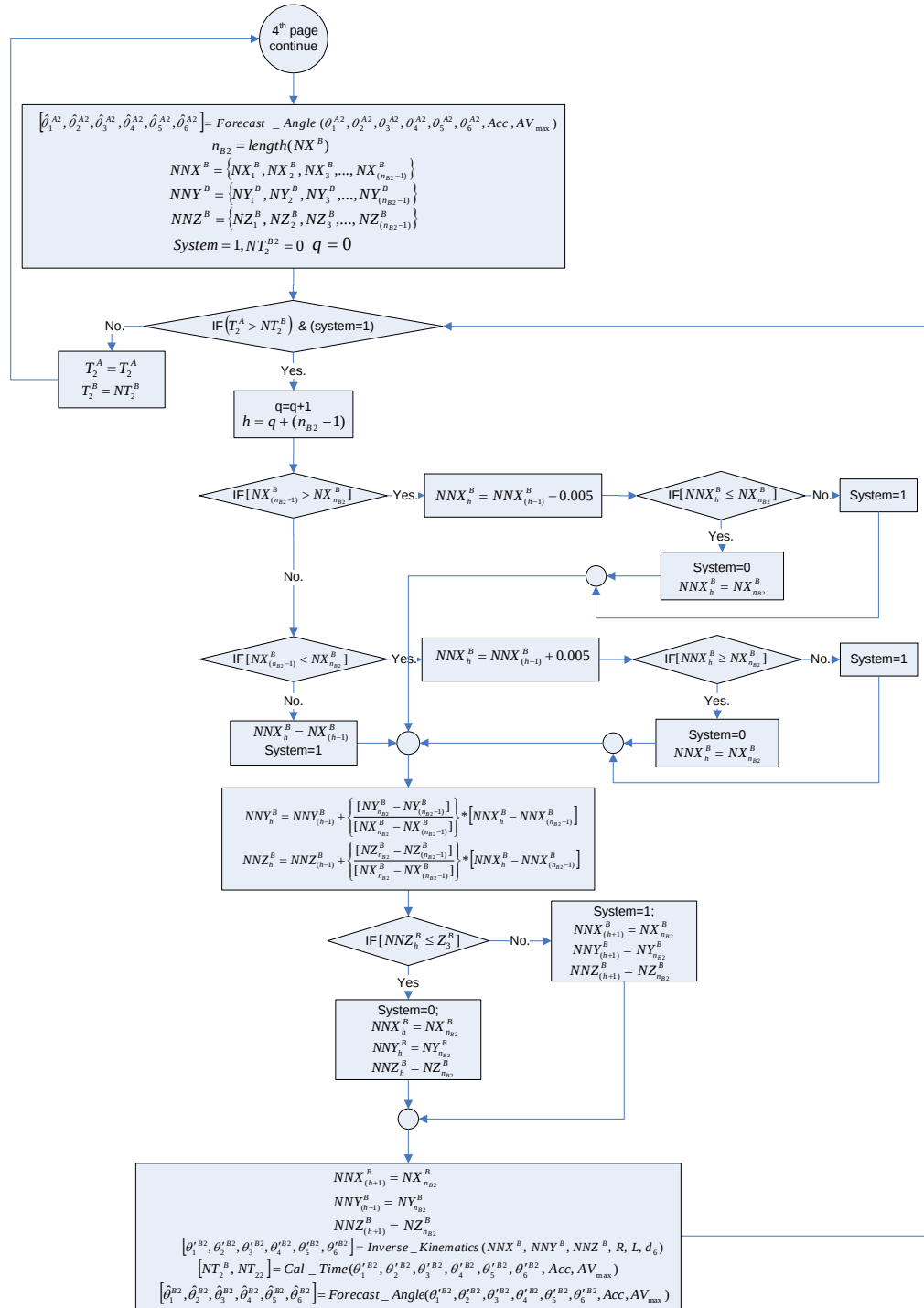
ภาพผนวกที่ 3 Flow Chart แสดงการหาค่าจำนวนตำแหน่งการเคลื่อนที่ช่วงที่ 2



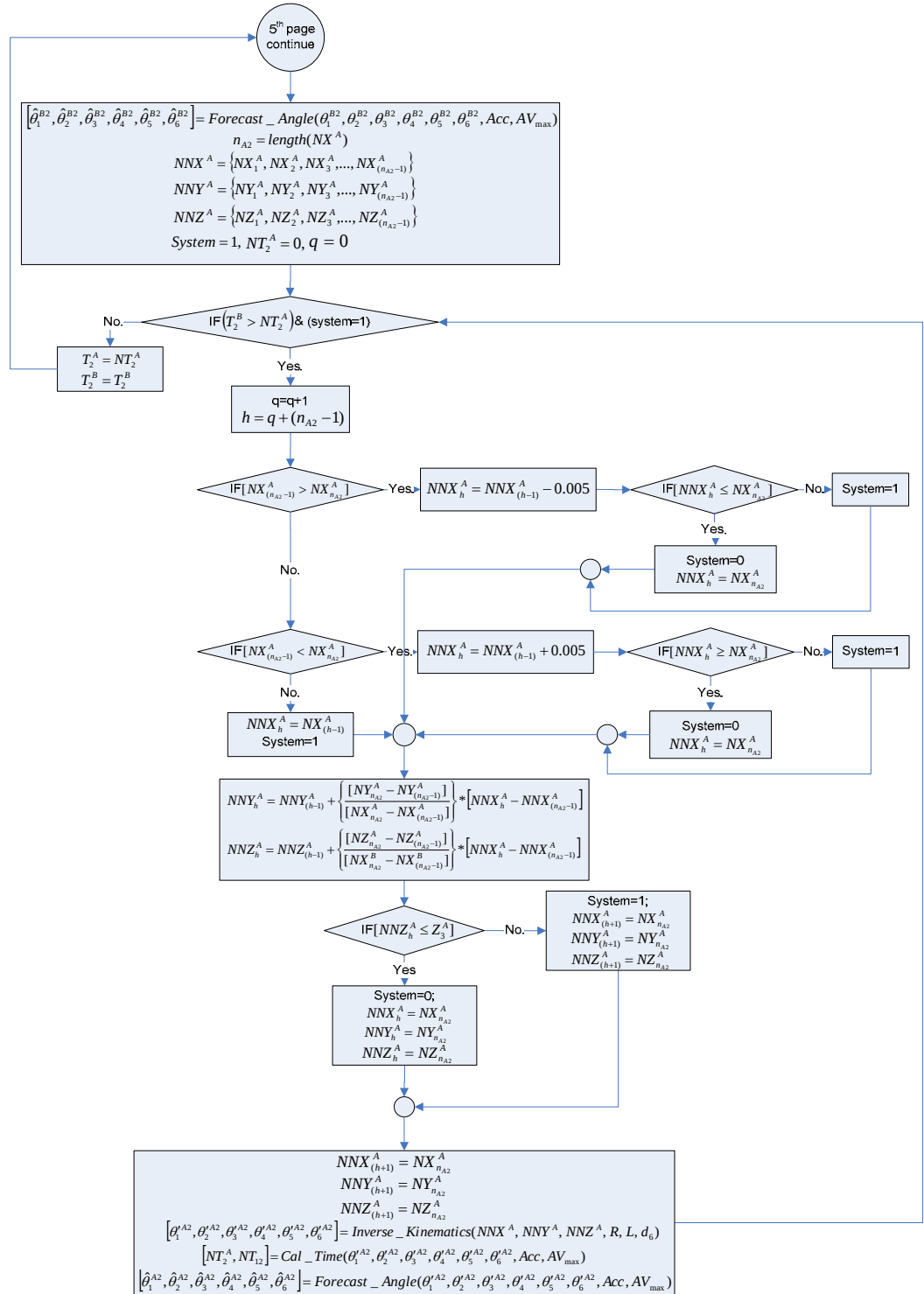
ภาพผนวกที่ 4 Flow Chart แสดงการหาค่าจำนวนตำแหน่งการเคลื่อนที่ช่วงที่ 2 (ต่อ)



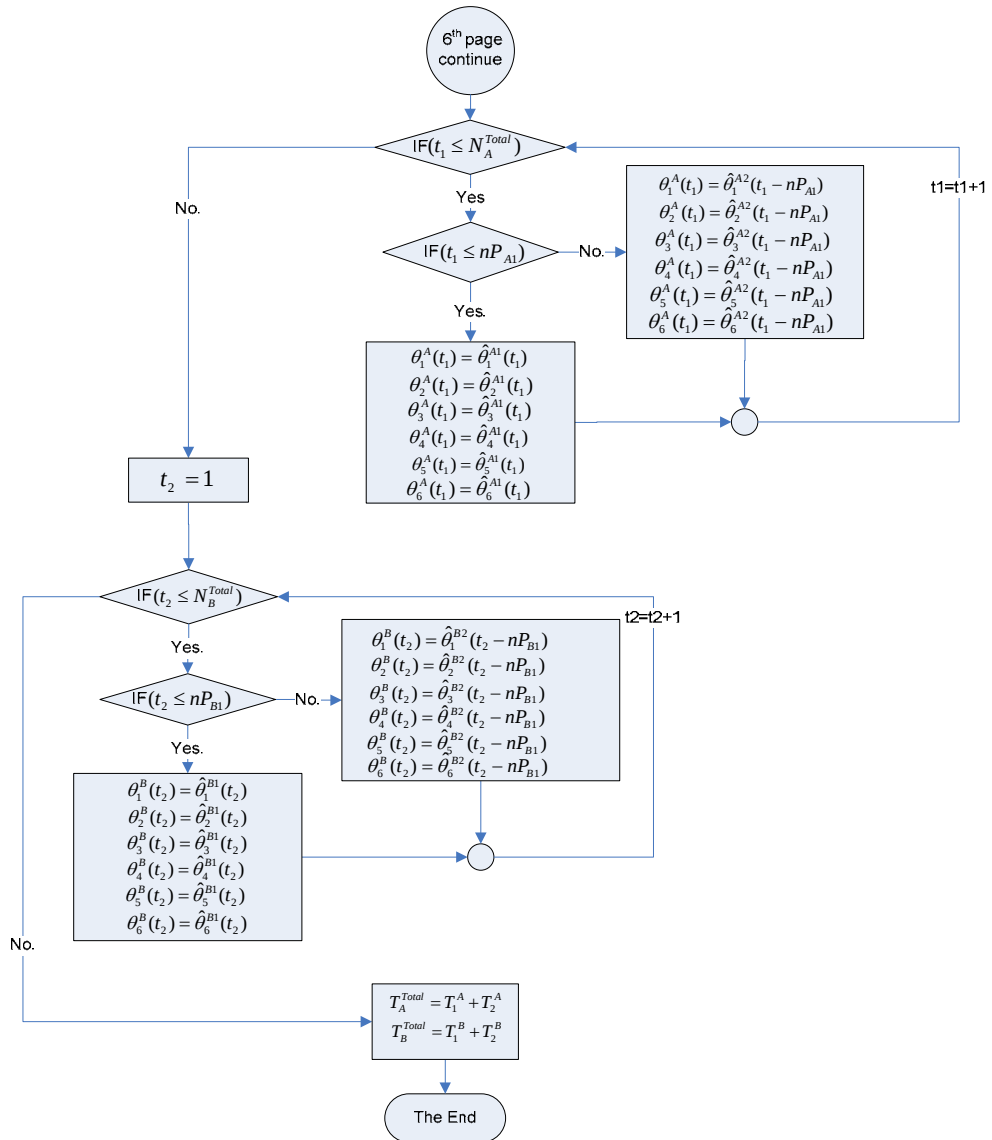
ภาพผนวกที่ 5 Flow Chart แสดงการหาค่าจำนวนตำแหน่งการเคลื่อนที่ช่วงที่ 2 (ต่อ)



ภาพผนวกที่ 6 Flow Chart แสดงการหาค่าจำนวนตำแหน่งการเคลื่อนที่ช่วงที่ 2 (ต่อ)



ภาพผนวกที่ 7 Flow Chart แสดงการหาค่าจำนวนตำแหน่งการเคลื่อนที่ช่วงที่ 2 (ต่อ)



ภาพผนวกที่ 8 Flow Chart แสดงการหาค่าจำนวนตำแหน่งการเคลื่อนที่ช่วงที่ 2 (ต่อ)

งานวิจัยครั้งนี้ได้ใช้โปรแกรม Matlab มาใช้ในการเขียนโปรแกรมแสดงการทำงานของ
 คำนวณหาตำแหน่งการทำงานของหุ่นยนต์ในแต่ละช่วงการเคลื่อนที่ โดยวิธีการเขียนโปรแกรม
 แสดงการทำงานได้แสดงไว้ดังนี้

%----- Main Program for Intermediate Algorithm-----%

%----- Import Data from Excel Program -----%

Import_Data = ddeinit('excel', 'Input_Data');

% -----Import position data for Main program from Excel Program -----%

X1=0; Y1=0; Z1=0; X2=0; Y2=0; Z2=0;

X1 = ddereq(Import_Data, 'r16c3:r16c5');

Y1 = ddereq(Import_Data, 'r17c3:r17c5');

Z1 = ddereq(Import_Data, 'r18c3:r18c5');

X2 = ddereq(Import_Data, 'r22c3:r22c5');

Y2 = ddereq(Import_Data, 'r23c3:r23c5');

Z2 = ddereq(Import_Data, 'r24c3:r24c5');

%----- Import data about Dimension of Cylinder and Gasket from Excel Program -----%

LC=0; RC=0; RWin=0;

LC = ddereq(Import_Data, 'r28c3');

RC = ddereq(Import_Data, 'r29c3');

RWin = ddereq(Import_Data, 'r32c3');

LRw = ddereq(Import_Data, 'r33c3');

%----- Calculate deviation between radius of cylinder and radius of Gasket -----%

Rdif=RWin-RC;

DisAB = abs(Y1(2)+Y2(2));

Dis_Tool = abs(X1(2)+X2(2));

```
%----- Import data about characteristic of Robot from Excel Program -----%
```

```
dim_R=0; d=0; d6=0; Length_Links=0; Acc=0; AV=0; Cyc_Time=0;
```

```
Length_Links = ddereq(Import_Data, 'r7c3:r11c3');
```

```
dim_R=size(Length_Links);
```

```
for(i=1:dim_R(1))
```

```
    if(i~=dim_R(1))
```

```
        d(i) = Length_Links(i,1);
```

```
    else
```

```
        d6 = Length_Links(i,1);
```

```
    end
```

```
end
```

```
%----- Import Angular Accelation data from Excel Program -----%
```

```
Acc = ddereq(Import_Data, 'r37c3');
```

```
%----- Import Angular Valocity data from Excel Program -----%
```

```
AV = ddereq(Import_Data, 'r40c3');
```

```
Max_AV=AV*(2*pi)/60;
```

```
%---- Import Cycle time data for Operation from Excel Program -----%
```

```
Cyc_Time = ddereq(Import_Data, 'r43c3');
```

```
R=0; p=0;
```

```
%----- Calculate Rotation Matrix of Robot-----%
```

```
k=0;
```

```
p=p+1;
```

```

for i=1:3
    for j=1:3
        if (i==1)
            if(j==1)
                R(i,j,p)=-cos((k*10)*pi/180);
            elseif(j==2)
                R(i,j,p)=0;
            else
                R(i,j,p)=sin((k*10)*pi/180);
            end
        elseif (i==2)
            if(j==2)
                R(i,j,p)=1;
            else
                R(i,j,p)=0;
            end
        elseif(i==3)
            if(j==1)
                R(i,j,p)=-sin((k*10)*pi/180);
            elseif(j==2)
                R(i,j,p)=0;
            else
                R(i,j,p)=-cos((k*10)*pi/180);
            end
        end
    end
end

Np=1;
XX1=0; YY1=0; ZZ1=0; XX2=0; YY2=0; ZZ2=0;
n=size(X1);

```

```

if(n(2)>2)
    for(i=1:2)
        XX1(i)=X1(i);
        XX2(i)=X2(i);
        YY1(i)=Y1(i);
        YY2(i)=Y2(i);
        ZZ1(i)=Z1(i);
        ZZ2(i)=Z2(i);
    end
elseif(n(2)==2)
    XX1=X1;
    XX2=X2;
    YY1=Y1;
    YY2=Y2;
    ZZ1=Z1;
    ZZ2=Z2;
else
    sprintf('it not can doing sovle')
end

system_st = 1; p1=0;
Cx1=0; Cy1=0; Cz1=0; Cx2=0; Cy2=0; Cz2=0;
while(system_st==1)

%----- Define variable of Robot A and B for 1st Period-----%
%----- Robot A -----%
ze_111=0; ze_211=0; ze_311=0; ze_411=0; ze_511=0; ze_611=0;
Nze_111=0; Nze_211=0; Nze_311=0; Nze_411=0; Nze_511=0; Nze_611=0;
Px11=0; Py11=0; Pz11=0; Sum_time1=0; Tp1=0; Opt_Time11=0; Path_Error11=0;
NX11=0; NX21=0; NY11=0; NTf11=0;

```

```

%----- Robot B -----%

ze_121=0; ze_221=0; ze_321=0; ze_421=0; ze_521=0; ze_621=0;
Nze_121=0; Nze_221=0; Nze_321=0; Nze_421=0; Nze_521=0; Nze_621=0;
Px21=0; Py21=0; Pz21=0; Sum_time2=0; Tp2=0; Opt_Time21=0; Path_Error21=0;
NY21=0; NZ11=0; NZ21=0; NTf21=0;

% ----- The End for Defining -----%

[ze_111 ze_211 ze_311 ze_411 ze_511 ze_611]=Cal_Angle(XX1, YY1, ZZ1, d6, R, d);
[ze_121 ze_221 ze_321 ze_421 ze_521 ze_621]=Cal_Angle(XX2, YY2, ZZ2, d6, R, d);
[Sum_time1, Tp1]=Cal_Time(ze_111, ze_211, ze_311, ze_411, ze_511, ze_611, Acc);
[Sum_time2, Tp2]=Cal_Time(ze_121, ze_221, ze_321, ze_421, ze_521, ze_621, Acc);
[Nze_111, Nze_211, Nze_311, Nze_411, Nze_511, Nze_611, NTf11]
    =Focast_angle(ze_111, ze_211, ze_311, ze_411, ze_511, ze_611, Acc, Max_AV);
[Nze_121, Nze_221, Nze_321, Nze_421, Nze_521, Nze_621, NTf21]
    =Focast_angle(ze_121, ze_221, ze_321, ze_421, ze_521, ze_621, Acc, Max_AV);

for(i=1:length(Nze_111))
    [Px11(i), Py11(i), Pz11(i)]=FWD_KMT(Nze_111(i), Nze_211(i), Nze_311(i), Nze_411(i),
    Nze_511(i), Nze_611(i), d6, d);
end
for(i=1:length(Nze_121))
    [Px21(i), Py21(i), Pz21(i)]=FWD_KMT(Nze_121(i), Nze_221(i), Nze_321(i), Nze_421(i),
    Nze_521(i), Nze_621(i), d6, d);
End

%----- Check collision of Trajectory robot for 1st Period -----%
XdisAB=0; YdisAB=0; ZdisAB=0; nA1=0; nB1=0; APy1=0; APy2=0;
nA1 = length(Px11);
nB1 = length(Px21);
nT1 = min(nA1,nB1);

```

```

Dis_PY = RC+RWin;
w=1; system = 1;
while(w<=nT1)&&(XdisAB(w)<Dis_Tool)&&(system==1)
    APy1(w) = Py11(w)+RWin;
    APy2(w) = DisAB-(Py21(w)+RC);
    ZdisAB(w) = abs(Pz11(w)-Pz21(w));
    if((APy2(w)-APy1(w))>0)
        system = 1;
        w=w+1;
        if(w<=nT1)
            XdisAB(w) = abs(Px11(w)+Px21(w));
        else
            system = 0;
        end
    elseif((APy2(w)-APy1(w))<=0)&(ZdisAB(w)<(2*RWin))
        system = 0;
        p1=p1+1;

        %----- The Position appear collision for Robot A and B-----%
        Cx1(p1)=Px11(w);
        Cy1(p1)=Py11(w);
        Cz1(p1)=Pz11(w);
        Cx2(p1)=Px21(w);
        Cy2(p1)=Py21(w);
        Cz2(p1)=Pz21(w);

        %----- Add Position for movement for Robot A and B-----%
        NXX1=0; NYY1=0; NZZ1=0;

```

```

for(j=1:(length(XX1)+1))
    if(j<length(XX1))
        NXX1(j)=XX1(j);
        NYY1(j)=YY1(j);
        NZZ1(j)=ZZ1(j);
    elseif(j==length(XX1))
        NXX1(j) = Px11(w);
        NYY1(j) = Py11(w)-RWin;
        NZZ1(j) = Pz11(w);
    else
        NXX1(j)=XX1(length(XX1));
        NYY1(j)=YY1(length(XX1));
        NZZ1(j)=ZZ1(length(XX1));
    end
end
NXX2=0; NYY2=0; NZZ2=0;
for(k=1:(length(XX2)+1))
    if(k<length(XX2))
        NXX2(k)=XX2(k);
        NYY2(k)=YY2(k);
        NZZ2(k)=ZZ2(k);
    elseif(k==length(XX2))
        NXX2(k) = Px21(w);
        NYY2(k) = Py21(w)-RC;
        NZZ2(k) = Pz21(w);
    else
        NXX2(k)=XX2(length(XX2));
        NYY2(k)=YY2(length(XX2));
        NZZ2(k)=ZZ2(length(XX2));
    end
end

```

```

end
XX1=0; YY1=0; ZZ1=0;
XX1 = NXX1;
YY1 = NYY1;
ZZ1 = NZZ1;
XX2=0; YY2=0; ZZ2=0;
XX2 = NXX2;
YY2 = NYY2;
ZZ2 = NZZ2;
%----- The End for addition position -----%
else
    system = 1;
    w=w+1;
    XdisAB(w) = abs(Px11(w)+Px21(w));
    if(w<=nT1)
        XdisAB(w) = abs(Px11(w)+Px21(w));
    else
        system = 0;
    end
end
end
if(w>=nT1)|(XdisAB(w-1)>Dis_Tool)
    system_st=0;
else
    system_st=1;
end
end
Opt_Time21=Sum_time2;
Opt_Time11=Sum_time1;

```

```

NTf11=0; NTf21=0;

[Nze_111, Nze_211, Nze_311, Nze_411, Nze_511, Nze_611, NTf11]=Focast_angle(ze_111,
ze_211, ze_311, ze_411, ze_511, ze_611, Acc, Max_AV);

[Nze_121, Nze_221, Nze_321, Nze_421, Nze_521, Nze_621, NTf21]=Focast_angle(ze_121,
ze_221, ze_321, ze_421, ze_521, ze_621, Acc, Max_AV);

```

%--Adapt Angular for Trajectory of Robot to adapt Operation time as same as--%

```

L_Ntheta1 = length(Nze_111)
L_Ntheta2 = length(Nze_121)
if(L_Ntheta1>L_Ntheta2)
    L_ANthe2 = L_Ntheta2+1;
    while(L_ANthe2<=L_Ntheta1)
        Nze_121(L_ANthe2) = Nze_121(L_Ntheta2);
        Nze_221(L_ANthe2) = Nze_221(L_Ntheta2);
        Nze_321(L_ANthe2) = Nze_321(L_Ntheta2);
        Nze_421(L_ANthe2) = Nze_421(L_Ntheta2);
        Nze_521(L_ANthe2) = Nze_521(L_Ntheta2);
        Nze_621(L_ANthe2) = Nze_621(L_Ntheta2);
        L_ANthe2 = L_ANthe2+1;
    end
else
    L_ANthe1 = L_Ntheta1+1;
    while(L_ANthe1<=L_Ntheta2)
        Nze_111(L_ANthe1) = Nze_111(L_Ntheta1);
        Nze_211(L_ANthe1) = Nze_211(L_Ntheta1);
        Nze_311(L_ANthe1) = Nze_311(L_Ntheta1);
        Nze_411(L_ANthe1) = Nze_411(L_Ntheta1);
        Nze_511(L_ANthe1) = Nze_511(L_Ntheta1);
        Nze_611(L_ANthe1) = Nze_611(L_Ntheta1);
        L_ANthe1 = L_ANthe1+1;
    end
end

```

```

        end
    end

    for(i=1:length(Nze_111))
        [Px11(i), Py11(i), Pz11(i)]=FWD_KMT(Nze_111(i), Nze_211(i), Nze_311(i), Nze_411(i),
        Nze_511(i), Nze_611(i), d6, d);
    end

    for(i=1:length(Nze_121))
        [Px21(i), Py21(i), Pz21(i)]=FWD_KMT(Nze_121(i), Nze_221(i), Nze_321(i), Nze_421(i),
        Nze_521(i), Nze_621(i), d6, d);
    end

    %----- Result for First Period -----%
    % Output for Path in First Period is Variable
    % [XX1,YY1, ZZ1] is Path of First Period for Robot A
    % [XX2,YY2, ZZ2] is Path of First Period for Robot B
    %----- The End for First Period -----%

    %----- Start Program of Second Period -----%

    %----- Define variable of Robot A and B for 2nd Period-----%
    %----- Variable for Robot A -----%
    Px12=0; Py12=0; Pz12=0; Path_Error12=0; Opt_Time12=0;
    Remain_TimeA2=0; Remain_Time2=0; theta_RobotA=0; Result_R12=0;
    Nze_112=0; Nze_212=0; Nze_312=0; Nze_412=0; Nze_512=0; Nze_612=0;

    %----- Variable for Robot B -----%
    Px22=0; Py22=0; Pz22=0; Opt_Time22=0; Path_Error22=0;
    Remain_TimeB2=0; theta_RobotB=0; Result_R22=0;
    Nze_122=0; Nze_222=0; Nze_322=0; Nze_422=0; Nze_522=0; Nze_622=0;

```

```

%----- Calculate remain time for 2nd Period -----%
Remain_TimeA2=Cyc_Time-(Opt_Time11);
Remain_TimeB2=Cyc_Time-(Opt_Time21);
Remain_Time2=min(Remain_TimeA2, Remain_TimeB2);
for(i=2:3)
    X12(i-1)=X1(i);
    X22(i-1)=X2(i);
    Y12(i-1)=Y1(i);
    Y22(i-1)=Y2(i);
    Z12(i-1)=Z1(i);
    Z22(i-1)=Z2(i);
end

OPT_TAB=0; Path_Error_AB=0; Max_R_Error=0; par2=0;
[par2, Max_R_Error]=Select_p_Artificial2(X12, Y12, Z12, X22, Y22, Z22, d, d6, R, Rdif, Acc,
Max_AV, Remain_Time2);
[Px12, Py12, Pz12, Px22, Py22, Pz22, Result_R12, Result_R22, Path_Error_AB, OPT_TAB,
theta_RobotA, theta_RobotB]=Cal_PIT_RB3(X12, Y12, Z12, X22, Y22, Z22, par2, R, Acc, d,
d6, AV);
Path_Error12=Path_Error_AB(1);
Path_Error22=Path_Error_AB(2);
Opt_Time12=OPT_TAB(1);
Opt_Time22=OPT_TAB(2);

%----- Angular for Robot A-----%
Nze_112 = theta_RobotA(1,:);
Nze_212 = theta_RobotA(2,:);
Nze_312 = theta_RobotA(3,:);
Nze_412 = theta_RobotA(4,:);
Nze_512 = theta_RobotA(5,:);

```

```

Nze_612 = theta_RobotA(6,:);

%----- Theta for Robot B-----%
Nze_122 = theta_RobotB(1,:);
Nze_222 = theta_RobotB(2,:);
Nze_322 = theta_RobotB(3,:);
Nze_422 = theta_RobotB(4,:);
Nze_522 = theta_RobotB(5,:);
Nze_622 = theta_RobotB(6,:);

%----- Step collect angular for movement of Robot A and B -----%
%----- Angular for Robot A-----%
nPA1=0; nPA2=0; Ntheta11=0; Ntheta21=0; Ntheta31=0;
Ntheta41=0; Ntheta51=0; Ntheta61=0;
nPA1 = length(Nze_111);
nPA2 = length(Nze_112);
TPA = nPA1+nPA2;
for(t=1:TPA)
    if(t<=nPA1)
        Ntheta11(t)=Nze_111(t);
        Ntheta21(t)=Nze_211(t);
        Ntheta31(t)=Nze_311(t);
        Ntheta41(t)=Nze_411(t);
        Ntheta51(t)=Nze_511(t);
        Ntheta61(t)=Nze_611(t);
    else
        Ntheta11(t)=Nze_112(t-nPA1);
        Ntheta21(t)=Nze_212(t-nPA1);
        Ntheta31(t)=Nze_312(t-nPA1);
        Ntheta41(t)=Nze_412(t-nPA1);
    end
end

```

```

        Ntheta51(t)=Nze_512(t-nPA1);
        Ntheta61(t)=Nze_612(t-nPA1);
    end
end

%----- Angular for Robot B-----%
nPB1=0; nPB2=0; Ntheta12=0; Ntheta22=0; Ntheta32=0;
Ntheta42=0; Ntheta52=0; Ntheta62=0;
nPB1 = length(Nze_121);
nPB2 = length(Nze_122);
TPB = nPB1+nPB2;
for(t=1:TPB)
    if(t<=nPB1)
        Ntheta12(t)=Nze_121(t);
        Ntheta22(t)=Nze_221(t);
        Ntheta32(t)=Nze_321(t);
        Ntheta42(t)=Nze_421(t);
        Ntheta52(t)=Nze_521(t);
        Ntheta62(t)=Nze_621(t);
    else
        Ntheta12(t)=Nze_122(t-nPB1);
        Ntheta22(t)=Nze_222(t-nPB1);
        Ntheta32(t)=Nze_322(t-nPB1);
        Ntheta42(t)=Nze_422(t-nPB1);
        Ntheta52(t)=Nze_522(t-nPB1);
        Ntheta62(t)=Nze_622(t-nPB1);
    end
end
end

```

%----- Adapt to as same as Quantity of position for Robot A and B -----%

if(TPA>TPB)

for(w=(TPB+1):TPA)

Ntheta12(w) = Ntheta12(w-1);

Ntheta22(w) = Ntheta22(w-1);

Ntheta32(w) = Ntheta32(w-1);

Ntheta42(w) = Ntheta42(w-1);

Ntheta52(w) = Ntheta52(w-1);

Ntheta62(w) = Ntheta62(w-1);

end

elseif(TPB>TPA)

for(w=(TPA+1):TPB)

Ntheta11(w) = Ntheta11(w-1);

Ntheta21(w) = Ntheta21(w-1);

Ntheta31(w) = Ntheta31(w-1);

Ntheta41(w) = Ntheta41(w-1);

Ntheta51(w) = Ntheta51(w-1);

Ntheta61(w) = Ntheta61(w-1);

end

end

%----- Step collect position of Robot A and B -----%

t1=0; t2=0;

t1=size(Px11)+size(Px12);

t2=size(Px21)+size(Px22);

NPx1=0; NPy1=0; NPz1=0;

NPx2=0; NPy2=0; NPz2=0;

for(i=1:t1(2))

if(i<=length(Px11))

NPx1(i)=Px11(i);

```

        NPy1(i)=Py11(i);
        NPz1(i)=Pz11(i);
    else
        NPx1(i)=Px12(i-length(Px11));
        NPy1(i)=Py12(i-length(Px11));
        NPz1(i)=Pz12(i-length(Px11));
    end
end
for(j=1:t2(2))
    if(j<=length(Px21))
        NPx2(j)=Px21(j);
        NPy2(j)=Py21(j);
        NPz2(j)=Pz21(j);
    else
        NPx2(j)=Px22(j-length(Px21));
        NPy2(j)=Py22(j-length(Px21));
        NPz2(j)=Pz22(j-length(Px21));
    end
end

%----- Step calculate total operation time and total error -----%
Total_Opt_Time_A=0; Total_Opt_Time_B=0;
Path_Error_A=Path_Error12;
Path_Error_B=Path_Error22;
Total_Opt_Time_A=Opt_Time11+Opt_Time12;
Total_Opt_Time_B=Opt_Time21+Opt_Time22;

%----- Step show Position of Robot A and B -----%
NB_R1=length(NX11)+length(Result_R12);
NB_R2=length(NX21)+length(Result_R22);

```

```

Position_R1_X=0; Position_R1_Y=0; Position_R1_Z=0; k=0;
Position_R2_X=0; Position_R2_Y=0; Position_R2_Z=0; j=0;
for(i=1:NB_R1)
    if(i<=length(NX11))
        if(NX11(i)~=Result_R12(1,1))
            j=j+1;
            Position_R1_X(j)=NX11(i);
            Position_R1_Y(j)=NY11(i);
            Position_R1_Z(j)=NZ11(i);
        end
    else
        k=k+1;
        j=j+1;
        Position_R1_X(j)=Result_R12(1,k);
        Position_R1_Y(j)=Result_R12(2,k);
        Position_R1_Z(j)=Result_R12(3,k);
    end
end
k=0; j=0;
for(i=1:NB_R2)
    if(i<=length(NX21))
        if(NX21(i)~=Result_R22(1,1))
            j=j+1;
            Position_R2_X(j)=NX21(i);
            Position_R2_Y(j)=NY21(i);
            Position_R2_Z(j)=NZ21(i);
        end
    else
        k=k+1;
        j=j+1;

```

```

        Position_R2_X(j)=Result_R22(1,k);
        Position_R2_Y(j)=Result_R22(2,k);
        Position_R2_Z(j)=Result_R22(3,k);
    end
end

%-----Show Result of Program in spread sheet Excel Program -----%
%----- Initialize conversation data with Excel Program -----%
Position = ddeinit('excel', 'Output_Intermedaite');

%----- Variable used to show Result from Program -----%
TTR1='Total Time of Robot A';
TTR2='Total Time of Robot B';
Min_Error='Distance between Robot A and B';

%----- Set range of cells in Excel for Programs -----%
Range_X1 = 'r2:r2';
Range_Y1 = 'r3:r3';
Range_Z1 = 'r4:r4';
Range_X2 = 'r7:r7';
Range_Y2 = 'r8:r8';
Range_Z2 = 'r9:r9';
Range_Time_R1 = 'r12c5';
Range_Time_R2 = 'r15c5';
Range_Show_Time_R1 = 'r12c2';
Range_Show_Time_R2 = 'r15c2';
Range_show_Error = 'r18c2';
Range_Error = 'r18c5';

```

```

%----- Show data go to the Excel spread sheet -----%

ddepoke(Position, Range_X1, Position_R1_X);
ddepoke(Position, Range_Y1, Position_R1_Y);
ddepoke(Position, Range_Z1, Position_R1_Z);
ddepoke(Position, Range_X2, Position_R2_X);
ddepoke(Position, Range_Y2, Position_R2_Y);
ddepoke(Position, Range_Z2, Position_R2_Z);
ddepoke(Position, Range_Time_R1, Total_Opt_Time_A);
ddepoke(Position, Range_Time_R2, Total_Opt_Time_B);
ddepoke(Position, Range_Show_Time_R1, TTR1);
ddepoke(Position, Range_Show_Time_R2, TTR2);
ddepoke(Position, Range_show_Error, Min_Error);
ddepoke(Position, Range_Error, Max_R_Error);
ddepoke(Position, 'r19c2', 'Number of Intermedaite');
ddepoke(Position, 'r19c5:r19c6', par2);
ddepoke(Position, 'r13c5', Opt_Time11);
ddepoke(Position, 'r14c5', Opt_Time12);
ddepoke(Position, 'r16c5', Opt_Time21);
ddepoke(Position, 'r17c5', Opt_Time22);
ddepoke(Position, 'r20c5', p1);
ddepoke(Position, 'r23c2:r23c22', Cx1);
ddepoke(Position, 'r24c2:r24c22', Cy1);
ddepoke(Position, 'r25c2:r25c22', Cz1);
ddepoke(Position, 'r27c2:r27c22', Cx2);
ddepoke(Position, 'r28c2:r28c22', Cy2);
ddepoke(Position, 'r29c2:r29c22', Cz2);

%----- The End for Program -----%

```

```

%----- Function Select_p -----%

Function [p, Max_R_Error]=Select_p_Artificial(X1, Y1, Z1, X2, Y2, Z2, d, d6, R, Rdif, Acc,
Max_AG, MaxTime)

%----- Define for Variables with Function -----%

Min_Error=10000; Min_Opt_Ti=0; r=1;

Max_Time=0;

p=0;

alpha=Acc; r2=0;

Max_R_Error=Rdif*2;

Length_X=abs(X1(1)+X2(1));

while((Max_R_Error>Rdif)&&(Max_Time<=MaxTime))

    r=r+1

    NX1=0; NY1=0; NZ1=0;

    NX2=0; NY2=0; NZ2=0;

    if(r<=420)

        NX1=line(X1, r);  NX2=line(X2, r);

        NY1=line(Y1, r);  NY2=line(Y2, r);

        NZ1=line(Z1, r);  NZ2=line(Z2, r);


        zeta11=0; zeta12=0; zeta13=0; zeta14=0; zeta15=0; zeta16=0;

        zeta21=0; zeta22=0; zeta23=0; zeta24=0; zeta25=0; zeta26=0;

        [zeta11 zeta12 zeta13 zeta14 zeta15 zeta16]=Cal_Angle(NX1, NY1, NZ1, d6, R, d);

        [zeta21 zeta22 zeta23 zeta24 zeta25 zeta26]=Cal_Angle(NX2, NY2, NZ2, d6, R, d);


    %----- Calculate Forward Position of Robot A and B -----%

    c=size(zeta11);

    Px1=0; Py1=0; Pz1=0;

    Px2=0; Py2=0; Pz2=0;

```

```

for(j=1:c(2))
    for(i=1:c(1))
        [Px1(j,i), Py1(j,i), Pz1(j,i)]=FWD_KMT(zeta11(i,j), zeta12(i,j), zeta13(i,j),
        zeta14(i,j), zeta15(i,j), zeta16(i,j), d6, d);
        [Px2(j,i), Py2(j,i), Pz2(j,i)]=FWD_KMT(zeta21(i,j), zeta22(i,j), zeta23(i,j),
        zeta24(i,j), zeta25(i,j), zeta26(i,j), d6, d);
    end
end

%----- Calculate p value -----%
Nze_11=0; Nze_12=0; Nze_13=0; Nze_14=0; Nze_15=0; Nze_16=0; Opt_t1=0; PT1=0;
Nze_21=0; Nze_22=0; Nze_23=0; Nze_24=0; Nze_25=0; Nze_26=0; Opt_t2=0; PT2=0;
Pnx1=0; Pny1=0; Pnz1=0; f1=0; Qdif=0; OPT_tAB=0;
Pnx2=0; Pny2=0; Pnz2=0; f2=0; dif=0; Posi=0;
[Opt_t1, PT1]=Cal_Time3(zeta11, zeta12, zeta13, zeta14, zeta15, zeta16, alpha,
Max_AG);
[Opt_t2, PT2]=Cal_Time3(zeta21, zeta22, zeta23, zeta24, zeta25, zeta26, alpha,
Max_AG);
OPT_tAB=[Opt_t1 Opt_t2];
Max_Time=max(Opt_t1, Opt_t2);
[Nze_11, Nze_12, Nze_13, Nze_14, Nze_15, Nze_16]=Focast_angle3(zeta11, zeta12,
zeta13, zeta14, zeta15, zeta16, alpha, Max_AG);
[Nze_21, Nze_22, Nze_23, Nze_24, Nze_25, Nze_26]=Focast_angle3(zeta21, zeta22,
zeta23, zeta24, zeta25, zeta26, alpha, Max_AG);
f1=size(Nze_11);
f2=size(Nze_21);
Posi=max(f1(2),f2(2));
Qdif=abs(f1(2)-f2(2));

```

```

if(f1(2)>f2(2))
    for(i=1:f1(1))
        for(j=1:f1(2))
            [Pnx1(i,j), Pny1(i,j), Pnz1(i,j)]=FWD_KMT(Nze_11(i,j), Nze_12(i,j), Nze_13(i,j),
                Nze_14(i,j), Nze_15(i,j), Nze_16(i,j), d6, d);
        end
    end
    for(s=1:f2(1))
        for(h=1:f2(2))
            [Pnx2(s,h), Pny2(s,h), Pnz2(s,h)]=FWD_KMT(Nze_21(s,h), Nze_22(s,h),
                Nze_23(s,h), Nze_24(s,h), Nze_25(s,h), Nze_26(s,h), d6, d);
        end
    end
    for(s=1:f2(1))
        for(n=1:Qdif)
            dif=f2(2)+n;
            Pnx2(s,dif)=Pnx2(s,f2(2));
            Pny2(s,dif)=Pny2(s,f2(2));
            Pnz2(s,dif)=Pnz2(s,f2(2));
        end
    end
elseif(f2(2)>f1(2))
    for(i=1:f1(1))
        for(j=1:f1(2))
            [Pnx1(i,j), Pny1(i,j), Pnz1(i,j)]=FWD_KMT(Nze_11(i,j), Nze_12(i,j), Nze_13(i,j),
                Nze_14(i,j), Nze_15(i,j), Nze_16(i,j), d6, d);
        end
    end
end

```

```

for(s=1:f2(1))
    for(h=1:f2(2))
        [Pnx2(s,h), Pny2(s,h), Pnz2(s,h)]=FWD_KMT(Nze_21(s,h), Nze_22(s,h),
            Nze_23(s,h), Nze_24(s,h), Nze_25(s,h), Nze_26(s,h), d6, d);
    end
end
for(i=1:f1(1))
    for(n=1:Qdif)
        dif=f1(2)+n;
        Pnx1(i,dif)=Pnx1(i,f1(2));
        Pny1(i,dif)=Pny1(i,f1(2));
        Pnz1(i,dif)=Pnz1(i,f1(2));
    end
end
else
    for(i=1:f1(1))
        for(j=1:f1(2))
            [Pnx1(i,j), Pny1(i,j), Pnz1(i,j)]=FWD_KMT(Nze_11(i,j), Nze_12(i,j), Nze_13(i,j),
                Nze_14(i,j), Nze_15(i,j), Nze_16(i,j), d6, d);
        end
    end
    for(s=1:f2(1))
        for(h=1:f2(2))
            [Pnx2(s,h), Pny2(s,h), Pnz2(s,h)]=FWD_KMT(Nze_21(s,h), Nze_22(s,h),
                Nze_23(s,h), Nze_24(s,h), Nze_25(s,h), Nze_26(s,h), d6, d);
        end
    end
end
end

```

```

%----- Step check error of Operation -----%
RX_Er=0; RY_Er=0; RZ_Er=0;
RX_Error=0; RY_Error=0; RZ_Error=0;
for(i=1:f1(1))
    for(j=1:Posi)
        RX_Er(i,j)=roundn(abs(Pnx1(i,j)+Pnx2(i,j)),-2);
        RY_Er(i,j)=roundn(abs(Pny1(i,j)-Pny2(i,j)),-2);
        RZ_Er(i,j)=roundn(abs(Pnz1(i,j)-Pnz2(i,j)),-2);
    end
end
for(i=1:f1(1))
    RX_Error(i)=max(RX_Er(i,:));
    RY_Error(i)=max(RY_Er(i,:));
    RZ_Error(i)=max(RZ_Er(i,:));
end
Max_Rx_Error=0; Max_R_Error=0;
Max_Rx_Error=max(RX_Er)
if(Max_Rx_Error<=Length_X)
    for(i=1:f1(1))
        R_Error(i)=max(RZ_Error(i), RY_Error(i))
    end
    Max_R_Error=max(R_Error);
else
    Max_R_Error=Inf;
end
else
    system=1;
    k=0;
    r2=r;
    Old_Error=0;

```

```

while(k<=20)&&(system==1)&&(Max_R_Error>=Rdif)
    if(k==0)
        r2=r
    else
        r2=r2+1
    end
    NX1=line(X1, r);  NX2=line(X2, r2);
    NY1=line(Y1, r);  NY2=line(Y2, r2);
    NZ1=line(Z1, r);  NZ2=line(Z2, r2);

    zeta11=0; zeta12=0; zeta13=0; zeta14=0; zeta15=0; zeta16=0;
    zeta21=0; zeta22=0; zeta23=0; zeta24=0; zeta25=0; zeta26=0;
    [zeta11 zeta12 zeta13 zeta14 zeta15 zeta16]=Cal_Angle(NX1, NY1, NZ1, d6, R, d);
    [zeta21 zeta22 zeta23 zeta24 zeta25 zeta26]=Cal_Angle(NX2, NY2, NZ2, d6, R, d);

%----- Calculate Forward Position of Robot A and B -----%
Px1=0; Py1=0; Pz1=0; c=0;
Px2=0; Py2=0; Pz2=0; c2=0;
c=size(zeta11);
c2=size(zeta21);
for(j=1:c(2))
    for(i=1:c(1))
        [Px1(j,i), Py1(j,i), Pz1(j,i)]=FWD_KMT(zeta11(i,j), zeta12(i,j), zeta13(i,j),
        zeta14(i,j), zeta15(i,j), zeta16(i,j), d6, d);
    end
end
end

```

```

for(j=1:c2(2))
    for(i=1:c(1))
        [Px2(j,i), Py2(j,i), Pz2(j,i)]=FWD_KMT(zeta21(i,j), zeta22(i,j), zeta23(i,j),
            zeta24(i,j), zeta25(i,j), zeta26(i,j), d6, d);
    end
end

Nze_11=0; Nze_12=0; Nze_13=0; Nze_14=0; Nze_15=0; Nze_16=0; Opt_t1=0; PT1=0;
Nze_21=0; Nze_22=0; Nze_23=0; Nze_24=0; Nze_25=0; Nze_26=0; Opt_t2=0; PT2=0;
Pnx1=0; Pny1=0; Pnz1=0; f1=0; Qdif=0; OPT_tAB=0;
Pnx2=0; Pny2=0; Pnz2=0; f2=0; dif=0; Posi=0;
[Opt_t1, PT1]=Cal_Time3(zeta11, zeta12, zeta13, zeta14, zeta15, zeta16, alpha,
    Max_AG);
[Opt_t2, PT2]=Cal_Time3(zeta21, zeta22, zeta23, zeta24, zeta25, zeta26, alpha,
    Max_AG);
OPT_tAB=[Opt_t1 Opt_t2];
Max_Time=max(Opt_t1, Opt_t2);
[Nze_11, Nze_12, Nze_13, Nze_14, Nze_15, Nze_16]=Focast_angle3(zeta11, zeta12,
    zeta13, zeta14, zeta15, zeta16, alpha, Max_AG);
[Nze_21, Nze_22, Nze_23, Nze_24, Nze_25, Nze_26]=Focast_angle3(zeta21, zeta22,
    zeta23, zeta24, zeta25, zeta26, alpha, Max_AG);
f1=size(Nze_11);
f2=size(Nze_21);
Posi=max(f1(2),f2(2));
Qdif=abs(f1(2)-f2(2));

```

```

if(f1(2)>f2(2))
    for(i=1:f1(1))
        for(j=1:f1(2))
            [Pnx1(i,j), Pny1(i,j), Pnz1(i,j)]=FWD_KMT(Nze_11(i,j), Nze_12(i,j),
            Nze_13(i,j), Nze_14(i,j), Nze_15(i,j), Nze_16(i,j), d6, d);
        end
    end
    for(s=1:f2(1))
        for(h=1:f2(2))
            [Pnx2(s,h), Pny2(s,h), Pnz2(s,h)]=FWD_KMT(Nze_21(s,h), Nze_22(s,h),
            Nze_23(s,h), Nze_24(s,h), Nze_25(s,h), Nze_26(s,h), d6, d);
        end
    end
    for(s=1:f2(1))
        for(n=1:Qdif)
            dif=f2(2)+n;
            Pnx2(s,dif)=Pnx2(s,f2(2));
            Pny2(s,dif)=Pny2(s,f2(2));
            Pnz2(s,dif)=Pnz2(s,f2(2));
        end
    end
elseif(f2(2)>f1(2))
    for(i=1:f1(1))
        for(j=1:f1(2))
            [Pnx1(i,j), Pny1(i,j), Pnz1(i,j)]=FWD_KMT(Nze_11(i,j), Nze_12(i,j),
            Nze_13(i,j), Nze_14(i,j), Nze_15(i,j), Nze_16(i,j), d6, d);
        end
    end
end

```

```

for(s=1:f2(1))
    for(h=1:f2(2))
        [Pnx2(s,h), Pny2(s,h), Pnz2(s,h)]=FWD_KMT(Nze_21(s,h), Nze_22(s,h),
            Nze_23(s,h), Nze_24(s,h), Nze_25(s,h), Nze_26(s,h), d6, d);
    end
end
for(i=1:f1(1))
    for(n=1:Qdif)
        dif=f1(2)+n;
        Pnx1(i,dif)=Pnx1(i,f1(2));
        Pny1(i,dif)=Pny1(i,f1(2));
        Pnz1(i,dif)=Pnz1(i,f1(2));
    end
end
else
    for(i=1:f1(1))
        for(j=1:f1(2))
            [Pnx1(i,j), Pny1(i,j), Pnz1(i,j)]=FWD_KMT(Nze_11(i,j), Nze_12(i,j),
                Nze_13(i,j), Nze_14(i,j), Nze_15(i,j), Nze_16(i,j), d6, d);
        end
    end
    for(s=1:f2(1))
        for(h=1:f2(2))
            [Pnx2(s,h), Pny2(s,h), Pnz2(s,h)]=FWD_KMT(Nze_21(s,h), Nze_22(s,h),
                Nze_23(s,h), Nze_24(s,h), Nze_25(s,h), Nze_26(s,h), d6, d);
        end
    end
end
end

```

```

%----- Step check error of Operation for Robot A and B -----%

RX_Er=0; RY_Er=0; RZ_Er=0;
RX_Error=0; RY_Error=0; RZ_Error=0;
for(i=1:f1(1))
    for(j=1:Posi)
        RX_Er(i,j)=roundn(abs(Pnx1(i,j)+Pnx2(i,j)), -2);
        RY_Er(i,j)=roundn(abs(Pny1(i,j)-Pny2(i,j)), -2);
        RZ_Er(i,j)=roundn(abs(Pnz1(i,j)-Pnz2(i,j)), -2);
    end
end
for(i=1:f1(1))
    RX_Error(i)=max(RX_Er(i,:));
    RY_Error(i)=max(RY_Er(i,:));
    RZ_Error(i)=max(RZ_Er(i,:));
end
Max_Rx_Error=0;
Max_Rx_Error=max(RX_Error);
if(Max_Rx_Error<=Length_X)
    for(i=1:f1(1))
        R_Error(i)=max(RZ_Error(i), RY_Error(i));
    end
    Max_R_Error=max(R_Error);
else
    Max_R_Error=Inf;
end
if(Max_R_Error==Inf)
    system=0;;
else
    k=k+1
    Old_Error(k)=Max_R_Error

```

```

        if(k>=2)
            if(Old_Error(k)<=Old_Error(k-1))
                system=1;
            else
                system=0;
            end
        end
    end
end
end
end
if(Max_R_Error<Rdif)
    if(r<=420)
        p=[r r];
    else
        p=[r r2];
    end
else
    sprintf('Setup new value of Maximize operation time please.')
end
end

%----- End for Function Select_p_Artificial -----%

```

%----- Describe to Function Forecast_Angular -----%

Function [Nze1, Nze2, Nze3, Nze4, Nze5, Nze6]= **Focast_angle3**(ze1, ze2, ze3, ze4, ze5, ze6,
alpha, Max_AG)

B0=0; B2=0; B3=0; Max_time=0; Tf=0; Ang_Vel=0; NTf=0;

ze1_T=0; ze2_T=0; ze3_T=0; ze4_T=0; ze5_T=0; ze6_T=0;

n=size(ze1);

for(i=1:n(2))

t=0;

for(j=2:n(1))

t=t+1;

Tf(t,i,1)=sqrt(6*abs(ze1(j,i)-ze1(t,i))/(alpha));

Tf(t,i,2)=sqrt(6*abs(ze2(j,i)-ze2(t,i))/(alpha));

Tf(t,i,3)=sqrt(6*abs(ze3(j,i)-ze3(t,i))/(alpha));

Tf(t,i,4)=sqrt(6*abs(ze4(j,i)-ze4(t,i))/(alpha));

Tf(t,i,5)=sqrt(6*abs(ze5(j,i)-ze5(t,i))/(alpha));

Tf(t,i,6)=sqrt(6*abs(ze6(j,i)-ze6(t,i))/(alpha));

B0(t,i,1)=ze1(j-1,i);

B0(t,i,2)=ze2(j-1,i);

B0(t,i,3)=ze3(j-1,i);

B0(t,i,4)=ze4(j-1,i);

B0(t,i,5)=ze5(j-1,i);

B0(t,i,6)=ze6(j-1,i);

if(ze1(j,i)==ze1(t,i))

B2(t,i,1)=0;

B3(t,i,1)=0;

else

B2(t,i,1)=3*(ze1(j,i)-ze1(t,i))/(Tf(t,i,1))^2;

B3(t,i,1)=-2*(ze1(j,i)-ze1(t,i))/(Tf(t,i,1))^3;

end

if(ze2(j,i)==ze2(t,i))

```

    B2(t,i,2)=0;
    B3(t,i,2)=0;
else
    B2(t,i,2)=3*(ze2(j,i)-ze2(t,i))/(Tf(t,i,2))^2;
    B3(t,i,2)=-2*(ze2(j,i)-ze2(t,i))/(Tf(t,i,2))^3;
end
if(ze3(j,i)==ze3(t,i))
    B2(t,i,3)=0;
    B3(t,i,3)=0;
else
    B2(t,i,3)=3*(ze3(j,i)-ze3(t,i))/(Tf(t,i,3))^2;
    B3(t,i,3)=-2*(ze3(j,i)-ze3(t,i))/(Tf(t,i,3))^3;
end
if(ze4(j,i)==ze4(t,i))
    B2(t,i,4)=0;
    B3(t,i,4)=0;
else
    B2(t,i,4)=3*(ze4(j,i)-ze4(t,i))/(Tf(t,i,4))^2;
    B3(t,i,4)=-2*(ze4(j,i)-ze4(t,i))/(Tf(t,i,4))^3;
end
if(ze5(j,i)==ze5(t,i))
    B2(t,i,5)=0;
    B3(t,i,5)=0;
else
    B2(t,i,5)=3*(ze5(j,i)-ze5(t,i))/(Tf(t,i,5))^2;
    B3(t,i,5)=-2*(ze5(j,i)-ze5(t,i))/(Tf(t,i,5))^3;
End

```

```

if(ze6(j,i)==ze6(t,i))
    B2(t,i,6)=0;
    B3(t,i,6)=0;
else
    B2(t,i,6)=3*(ze6(j,i)-ze6(t,i))/(Tf(t,i,6))^2;
    B3(t,i,6)=-2*(ze6(j,i)-ze6(t,i))/(Tf(t,i,6))^3;
end

%----- Check Angular Velocity have to value within Maximum value-----%
Ang_Vel_Link=0; Upp_Time=0; Low_Time=0;
for(w=1:6)
    Opt_Time=0; b=0; Ang_dist=0;
    while (Opt_Time<=Tf(t,i,w))
        b=b+1;
        Opt_Time=Opt_Time+0.05;
        Ang_Vel(w,b)=2*B2(t,i,w)*Opt_Time+3*B3(t,i,w)*(Opt_Time^2);
    end
    Ang_Vel_Link(w)=max(Ang_Vel(w,:));
    if(Ang_Vel_Link(w)>Max_AG)
        Upp_Time(w)=(-2*B2(t,i,w)+sqrt(4*(B2(t,i,w)^2)+1080*B3(t,i,w)))/(6*B3(t,i,w));
        Low_Time(w)=(-2*B2(t,i,w)-sqrt(4*(B2(t,i,w)^2)+1080*B3(t,i,w)))/(6*B3(t,i,w));
        Ang_dist(w)=(B2(t,i,w)*((Upp_Time(w)^2)-
            (Low_Time(w)^2))+B3(t,i,w)*((Upp_Time(w)^3)-(Low_Time(w)^3)));
        Part_Time=Ang_dist(w)/90;
        NTf(t,i,w)=Tf(t,i,w)+Part_Time-
            max(Upp_Time(w),Low_Time(w))+min(Upp_Time(w),Low_Time(w));
        NTf(t,i,w)=Tf(t,i,w);
        B2(t,i,w)=B2(t,i,w)*(Tf(t,i,w)^2)/(NTf(t,i,w)^2);
        B3(t,i,w)=B3(t,i,w)*(Tf(t,i,w)^3)/(NTf(t,i,w)^3);
    end
end

```

```

else
    NTf(t,i,w)=Tf(t,i,w);
end
end
Ti=[NTf(t,i,1) NTf(t,i,2) NTf(t,i,3) NTf(t,i,4) NTf(t,i,5) NTf(t,i,6)];
Max_time(t,i)=max(Ti);
end
end

%-----Step forecast angular for all joint in evertime -----%
ze_i=0;
for(m=1:n(2))
    k=0;
    for(i=1:(n(1)-1))
        t=0; j=0;
        while(t<Max_time(i,m))
            j=j+1;
            k=k+1;
            t=t+0.5;
            if((i==(n(1)-1))&(t>=Max_time(i,m)))
                ze1_T(i,j,m)=B0(i,m,1)+B2(i,m,1)*((NTf(i,m,1))^2)+B3(i,m,1)*((NTf(i,m,1))^3);
                ze2_T(i,j,m)=B0(i,m,2)+B2(i,m,2)*((NTf(i,m,2))^2)+B3(i,m,2)*((NTf(i,m,2))^3);
                ze3_T(i,j,m)=B0(i,m,3)+B2(i,m,3)*((NTf(i,m,3))^2)+B3(i,m,3)*((NTf(i,m,3))^3);
                ze4_T(i,j,m)=B0(i,m,4)+B2(i,m,4)*((NTf(i,m,4))^2)+B3(i,m,4)*((NTf(i,m,4))^3);
                ze5_T(i,j,m)=B0(i,m,5)+B2(i,m,5)*((NTf(i,m,5))^2)+B3(i,m,5)*((NTf(i,m,5))^3);
                ze6_T(i,j,m)=B0(i,m,6)+B2(i,m,6)*((NTf(i,m,6))^2)+B3(i,m,6)*((NTf(i,m,6))^3);
            else
                if(t<NTf(i,m,1))
                    ze1_T(i,j,m)=B0(i,m,1)+B2(i,m,1)*((t-0.5)^2)+B3(i,m,1)*((t-0.5)^3);
                else

```

```

if(NTf(i,m,1)==Max_time(i,m))
    ze1_T(i,j,m)=B0(i,m,1)+B2(i,m,1)*((t-0.5)^2)+B3(i,m,1)*((t-0.5)^3);
else
    ze1_T(i,j,m)=B0(i,m,1)+B2(i,m,1)*((NTf(i,m,1))^2)+B3(i,m,1)*((NTf(i,m,1))^3);
end
end

if(t<NTf(i,m,2))
    ze2_T(i,j,m)=B0(i,m,2)+B2(i,m,2)*((t-0.5)^2)+B3(i,m,2)*((t-0.5)^3);
else
    if(NTf(i,m,2)==Max_time(i,m))
        ze2_T(i,j,m)=B0(i,m,2)+B2(i,m,2)*((t-0.5)^2)+B3(i,m,2)*((t-0.5)^3);
    else
        ze2_T(i,j,m)=B0(i,m,2)+B2(i,m,2)*((NTf(i,m,2))^2)+B3(i,m,2)* NTf(i,m,2)^3;
    end
end

if(t<NTf(i,m,3))
    ze3_T(i,j,m)=B0(i,m,3)+B2(i,m,3)*((t-0.5)^2)+B3(i,m,3)*((t-0.5)^3);
else
    if(NTf(i,m,3)==Max_time(i,m))
        ze3_T(i,j,m)=B0(i,m,3)+B2(i,m,3)*((t-0.5)^2)+B3(i,m,3)*((t-0.5)^3);
    else
        ze3_T(i,j,m)=B0(i,m,3)+B2(i,m,3)*((NTf(i,m,3))^2)+B3(i,m,3)*NTf(i,m,3)^3;
    end
end
end

```

```

if(t<NTf(i,m,4))
    ze4_T(i,j,m)=B0(i,m,4)+B2(i,m,4)*((t-0.5)^2)+B3(i,m,4)*((t-0.5)^3);
else
    if(NTf(i,m,4)==Max_time(i,m))
        ze4_T(i,j,m)=B0(i,m,4)+B2(i,m,4)*((t-0.5)^2)+B3(i,m,4)*((t-0.5)^3);
    else
        ze4_T(i,j,m)=B0(i,m,4)+B2(i,m,4)*((NTf(i,m,4))^2)+B3(i,m,4)*((NTf(i,m,4))^3);
    end
end

if(t<NTf(i,m,5))
    ze5_T(i,j,m)=B0(i,m,5)+B2(i,m,5)*((t-0.5)^2)+B3(i,m,5)*((t-0.5)^3);
else
    if(NTf(i,m,5)==Max_time(i,m))
        ze5_T(i,j,m)=B0(i,m,5)+B2(i,m,5)*((t-0.5)^2)+B3(i,m,5)*((t-0.5)^3);
    else
        ze5_T(i,j,m)=B0(i,m,5)+B2(i,m,5)*((NTf(i,m,5))^2)+B3(i,m,5)*NTf(i,m,5)^3;
    end
end

if(t<NTf(i,m,6))
    ze6_T(i,j,m)=B0(i,m,6)+B2(i,m,6)*((t-0.5)^2)+B3(i,m,6)*((t-0.5)^3);
else
    if(NTf(i,m,6)==Max_time(i,m))
        ze6_T(i,j,m)=B0(i,m,6)+B2(i,m,6)*((t-0.5)^2)+B3(i,m,6)*((t-0.5)^3);
    else
        ze6_T(i,j,m)=B0(i,m,6)+B2(i,m,6)*((NTf(i,m,6))^2)+B3(i,m,6)*((NTf(i,m,6))^3);
    end
end
end

```

```

    % m is amount of path for Operation.

    % k is amount of position for trajectory of Robot.

    Nze1(m,k)=ze1_T(i,j,m);

    Nze2(m,k)=ze2_T(i,j,m);

    Nze3(m,k)=ze3_T(i,j,m);

    Nze4(m,k)=ze4_T(i,j,m);

    Nze5(m,k)=ze5_T(i,j,m);

    Nze6(m,k)=ze6_T(i,j,m);

    end

    end

    end

end

%----- Describe to function CHECK_ERROR for movement -----%
Function [Px,Py,Pz,Min_Er]= Check_Er(Nx, Ny, Nz, Pnx, Pny, Pnz, alpha, Tp)

    Myx=0; Mzx=0; w=0; Suf_t=0; Path_time=0; Point_Error=0;

    Sx=0; Sy=0; Sz=0;

    %----- Calculate slope path for movement of Robot A and B-----%
    for(i=1:(length(Nx)-1))
        if(Nx(i+1)==Nx(i))
            Myx(i)=0;
            Mzx(i)=0;
        else
            Myx(i)=(Ny(i+1)-Ny(i))/(Nx(i+1)-Nx(i));
            Mzx(i)=(Nz(i+1)-Nz(i))/(Nx(i+1)-Nx(i));
        end
    end

end

```

```

j=0; c=0; QP=0; d=0;
w=size(Tp);
for(i=1:w(2))
    for(j=1:w(1))
        Suf_t=[Tp(j,i,1), Tp(j,i,2), Tp(j,i,3), Tp(j,i,4), Tp(j,i,5), Tp(j,i,6)];
        Path_time(i,j)=max(Suf_t);
        if(mod(Path_time(i,j),0.5)>0)
            QP(i,j)=((Path_time(i,j)-mod(Path_time(i,j),0.5))/(0.5))+1;
        else
            QP(i,j)=((Path_time(i,j)-mod(Path_time(i,j),0.5))/(0.5));
        end
    end
end

%----- Calculate to actual Position for Movement of Robot A and B -----%
d=size(QP);
Sx=Pnx;
for(i=1:d(1))
    k=0;
    for(j=1:d(2))
        m=1;
        while(m<=QP(i,j))
            k=k+1;
            m=m+1;
            Sy(i,k)=(Myx(j)*(Sx(i,k)-Nx(j))+Ny(j));
            Sz(i,k)=(Mzx(j)*(Sx(i,k)-Nx(j))+Nz(j));
            Point_Error(i,k)=sqrt(((Sx(i,k)-Pnx(i,k))^2+(Sy(i,k)-Pny(i,k))^2+(Sz(i,k)-Pnz(i,k))^2);
        end
    end
end

```

```

    Path_Error(i)=sum(Point_Error(i,:));
end

%----- Select path to minimize Error -----%
Min_Er=min(Path_Error);
Px=0; Py=0; Pz=0;
for(q=1:d(1))
    if(Min_Er==Path_Error(q))
        g=size(QP(q,:)); U=0;
        for(i=1:g(2))
            U=U+QP(q,i);
        end
        for(j=1:U)
            Px(j)=Sx(q,j);
            Py(j)=Sy(q,j);
            Pz(j)=Sz(q,j);
        end
    end
end
end

%----- Describe to Function Calculate_Angular of All Link -----%
Function [zeta_1 zeta_2 zeta_3 zeta_4 zeta_5 zeta_6]= Cal_Angle(X, Y, Z, d6, R, d)

n=0; zeta_1=0; zeta_2=0; zeta_3=0; zeta_4=0; zeta_5=0; zeta_6=0; D=0;
n=size(R);
n(3)=1;
for(i=1:length(X))
    for(g=1:n(3))
        Xc(g)=X(i)-(d6*R(1,3,g));
        Yc(g)=Y(i)-(d6*R(2,3,g));
        Zc(g)=Z(i)-(d6*R(3,3,g));
    end
end

```

```

zeta_1(i,g)=(atan2(Yc(g),Xc(g)))*180/pi;
D(g)=(Xc(g)^2+Yc(g)^2-d(1)^2+(Zc(g)-d(2))^2-(d(3)^2)-(d(4)^2))/(2*d(3)*d(4));
zeta_3(i,g)=atan2(abs(sqrt(1-D(g)^2)),D(g))*180/pi;
zeta_2(i,g)=(atan2(Zc(g)-d(2),sqrt(Xc(g)^2+Yc(g)^2-d(1)^2))-
atan2(d(4)*sin(zeta_3(i,g)*pi/180),d(3)+d(4)*cos(zeta_3(i,g)*pi/180)))*180/pi;
zeta_4(i,g)=(atan2(-cos(zeta_1(i,g)*pi/180)*sin((zeta_2(i,g)+zeta_3(i,g))
*pi/180)*R(1,3,g)-sin(zeta_1(i,g)*pi/180)*sin((zeta_2(i,g)+zeta_3(i,g))
*pi/180)*R(2,3,g)+cos((zeta_2(i,g)+zeta_3(i,g))*pi/180)*R(3,3,g),cos(zeta_1(i,g)*
pi/180)*cos((zeta_2(i,g)+zeta_3(i,g))*pi/180)*R(1,3,g)+sin(zeta_1(i,g)*pi/180)*
cos((zeta_2(i,g)+zeta_3(i,g))*pi/180)*R(2,3,g)+sin((zeta_2(i,g)+zeta_3(i,g))*
pi/180)*R(3,3,g)))*180/pi;
zeta_5(i,g)=(atan2(sqrt(1-(sin(zeta_1(i,g)*pi/180)*R(1,3,g)-
cos(zeta_1(i,g)*pi/180)*R(2,3,g))^2),sin(zeta_1(i,g)*pi/180)*R(1,3,g)-
cos(zeta_1(i,g)*pi/180)*R(2,3,g)))*180/pi;
zeta_6(i,g)=(atan2(sin(zeta_1(i,g)*pi/180)*R(1,2,g)-cos(zeta_1(i,g)*pi/180)*R(2,2,g),-
sin(zeta_1(i,g)*pi/180)*R(1,1,g)+cos(zeta_1(i,g)*pi/180)*R(2,1,g)))*180/pi;
end
end
end

```

%----- Describe to Function Calculate_Time of All Link -----%

Function [Total_Time, NTf]= Cal_Time3(ze1, ze2, ze3, ze4, ze5, ze6, alpha, Max_AG)

%----- Define Variable to used within function -----%

```

B0=0; B2=0; B3=0; Max_time=0; Tf=0; Ang_Vel=0; NTf=0; Total_Time=0;
n=size(ze1);
for(i=1:n(2))
    t=0;
    for(j=2:n(1))
        t=t+1;
    end
end

```

```

Tf(t,i,1)=sqrt(6*abs(ze1(j,i)-ze1(t,i))/(alpha));
Tf(t,i,2)=sqrt(6*abs(ze2(j,i)-ze2(t,i))/(alpha));
Tf(t,i,3)=sqrt(6*abs(ze3(j,i)-ze3(t,i))/(alpha));
Tf(t,i,4)=sqrt(6*abs(ze4(j,i)-ze4(t,i))/(alpha));
Tf(t,i,5)=sqrt(6*abs(ze5(j,i)-ze5(t,i))/(alpha));
Tf(t,i,6)=sqrt(6*abs(ze6(j,i)-ze6(t,i))/(alpha));
B0(t,i,1)=ze1(j-1,i);
B0(t,i,2)=ze2(j-1,i);
B0(t,i,3)=ze3(j-1,i);
B0(t,i,4)=ze4(j-1,i);
B0(t,i,5)=ze5(j-1,i);
B0(t,i,6)=ze6(j-1,i);
if(ze1(j,i)==ze1(t,i))
    B2(t,i,1)=0;
    B3(t,i,1)=0;
else
    B2(t,i,1)=3*(ze1(j,i)-ze1(t,i))/(Tf(t,i,1))^2;
    B3(t,i,1)=-2*(ze1(j,i)-ze1(t,i))/(Tf(t,i,1))^3;
end
if(ze2(j,i)==ze2(t,i))
    B2(t,i,2)=0;
    B3(t,i,2)=0;
else
    B2(t,i,2)=3*(ze2(j,i)-ze2(t,i))/(Tf(t,i,2))^2;
    B3(t,i,2)=-2*(ze2(j,i)-ze2(t,i))/(Tf(t,i,2))^3;
end
if(ze3(j,i)==ze3(t,i))
    B2(t,i,3)=0;
    B3(t,i,3)=0;
else

```

```

    B2(t,i,3)=3*(ze3(j,i)-ze3(t,i))/(Tf(t,i,3))^2;
    B3(t,i,3)=-2*(ze3(j,i)-ze3(t,i))/(Tf(t,i,3))^3;
end
if(ze4(j,i)==ze4(t,i))
    B2(t,i,4)=0;
    B3(t,i,4)=0;
else
    B2(t,i,4)=3*(ze4(j,i)-ze4(t,i))/(Tf(t,i,4))^2;
    B3(t,i,4)=-2*(ze4(j,i)-ze4(t,i))/(Tf(t,i,4))^3;
end
if(ze5(j,i)==ze5(t,i))
    B2(t,i,5)=0;
    B3(t,i,5)=0;
else
    B2(t,i,5)=3*(ze5(j,i)-ze5(t,i))/(Tf(t,i,5))^2;
    B3(t,i,5)=-2*(ze5(j,i)-ze5(t,i))/(Tf(t,i,5))^3;
end
if(ze6(j,i)==ze6(t,i))
    B2(t,i,6)=0;
    B3(t,i,6)=0;
else
    B2(t,i,6)=3*(ze6(j,i)-ze6(t,i))/(Tf(t,i,6))^2;
    B3(t,i,6)=-2*(ze6(j,i)-ze6(t,i))/(Tf(t,i,6))^3;
end

%----- Check angular velocity within maximum velocity -----%
Ang_Vel_Link=0; Upp_Time=0; Low_Time=0;
for(w=1:6)
    Opt_Time=0; b=0; Ang_dist=0;

```

```

while (Opt_Time<=Tf(t,i,w))
    b=b+1;
    Opt_Time=Opt_Time+0.05;
    Ang_Vel(w,b)=2*B2(t,i,w)*Opt_Time+3*B3(t,i,w)*(Opt_Time^2);
end
Ang_Vel_Link(w)=max(Ang_Vel(w,:));
if(Ang_Vel_Link(w)>Max_AG)
    Upp_Time(w)=(-2*B2(t,i,w)+sqrt(4*(B2(t,i,w)^2)+1080*B3(t,i,w)))/(6*B3(t,i,w));
    Low_Time(w)=(-2*B2(t,i,w)-sqrt(4*(B2(t,i,w)^2)+1080*B3(t,i,w)))/(6*B3(t,i,w));
    Ang_dist(w)=(B2(t,i,w)*((Upp_Time(w)^2)-
    (Low_Time(w)^2))+B3(t,i,w)*((Upp_Time(w)^3)-(Low_Time(w)^3)));
    Part_Time=Ang_dist(w)/90;
    NTf(t,i,w)=Tf(t,i,w)+Part_Time-max(Upp_Time(w),Low_Time(w))+
    min(Upp_Time(w),Low_Time(w));
    NTf(t,i,w)=Tf(t,i,w);
    B2(t,i,w)=B2(t,i,w)*(Tf(t,i,w)^2)/(NTf(t,i,w)^2);
    B3(t,i,w)=B3(t,i,w)*(Tf(t,i,w)^3)/(NTf(t,i,w)^3);
else
    NTf(t,i,w)=Tf(t,i,w);
end
end
Ti=[NTf(t,i,1) NTf(t,i,2) NTf(t,i,3) NTf(t,i,4) NTf(t,i,5) NTf(t,i,6)];
Max_time(t,i)=max(Ti);
end
Total_Time(i)=sum(Max_time(:,i));
end
end

```

%-----Describe to function Forward_Kinematics for operation Robot -----%

Function [Px, Py, Pz]=FWD_KMT(Zeta_1, Zeta_2, Zeta_3, Zeta_4, Zeta_5, Zeta_6, d6, d)

```
Px=d6*cos(Zeta_1*pi/180)*sin(Zeta_5*pi/180)*cos((Zeta_2+Zeta_3+Zeta_4)*pi/180)+
d6*sin(Zeta_1*pi/180)*cos(Zeta_5*pi/180)+d(4)*cos(Zeta_1*pi/180)*cos((Zeta_2+
Zeta_3)*pi/180)+d(3)*cos(Zeta_1*pi/180)*cos(Zeta_2*pi/180);
Py=d6*sin(Zeta_1*pi/180)*sin(Zeta_5*pi/180)*cos((Zeta_2+Zeta_3+Zeta_4)*pi/180)-
d6*cos(Zeta_1*pi/180)*cos(Zeta_5*pi/180)+d(4)*sin(Zeta_1*pi/180)*
cos((Zeta_2+Zeta_3)*pi/180)+d(3)*sin(Zeta_1*pi/180)*cos(Zeta_2*pi/180);
Pz=d6*sin(Zeta_5*pi/180)*sin((Zeta_2+Zeta_3+Zeta_4)*pi/180)+
d(4)*sin((Zeta_2+Zeta_3)*pi/180)+d(3)*sin(Zeta_2*pi/180)+d(2);
```

end

%----- Discribe to function Line -----%

Function NX= line(X, p)

```
NX=0;
m=size(X);
k=0;
for(i=2:m(2))
    slope(i-1)=(X(i)-X(i-1))/p;
    for(j=1:p)
        k=k+1;
        if(j==1)
            NX(k)=X(i-1);
        else
            NX(k)=NX(k-1)+slope(i-1);
        if(i==m(2))&(j==p)
            NX(k+1)=X(m(2));
        end
    end
end
end
```

```

end

end

%----- Describe to function Animation_Robot for Operation -----%
Function animation_robot2(thetaA1,thetaA2,thetaA3,thetaA4,thetaA5,thetaA6,
thetaB1,thetaB2,thetaB3,thetaB4,thetaB5,thetaB6)

%-----Define variable about Program -----%
L=0; LC=0; RC=0; PaY2=0; PbY2=0; LW=0;
X1ini=0; Y1ini=0; Z1ini=0;

%----- Input variable for program -----%
Import_Data = ddeinit('excel', 'Input_Data');
Import_Data_animation = ddeinit('excel', 'Input_animation');
L = ddereq(Import_Data, 'r7c3:r11c3');
RC = ddereq(Import_Data, 'r29c3');
LC = ddereq(Import_Data, 'r28c3');
RW = ddereq(Import_Data, 'r32c3');
LW = ddereq(Import_Data, 'r33c3');
PaY2 = ddereq(Import_Data, 'r17c4');
PbY2 = ddereq(Import_Data, 'r23c4');
X1ini = ddereq(Import_Data, 'r16c3:r16c4');
Y1ini = ddereq(Import_Data, 'r17c3:r17c4');
Z1ini = ddereq(Import_Data, 'r18c3:r18c4');
X2ini = ddereq(Import_Data, 'r22c3:r22c4');
Y2ini = ddereq(Import_Data, 'r23c3:r23c4');
Z2ini = ddereq(Import_Data, 'r24c3:r24c4');

```

```

%----- Start program annimation to trajectory robot-----%

nR1=0; nR2=0; LinkA_1X=0; LinkA_1Y=0; LinkA_1Z=0; nRound=0;
LinkA_2X=0; LinkA_2Y=0; LinkA_2Z=0;

tr = pi/180;

length_AV=0.25;

Rdif = abs(RW-RC);

Length_Tool = 2*max(RC,RW)+length_AV;

distAB= PaY2+PbY2;

L_Tool = L(5)-Length_Tool;

nR1=size(thetaA1);

nR2=size(thetaB1);

nRound = max(max(nR1),max(nR2));

nLine = 300;

%----- Build to Cylinder and Gasket -----%

Xc=0; Yc=0; Zc=0; ZL=0; nRC=0; XCy=0; YCy=0; ZCy=0;

[Xc,Yc,Zc] = cylinder(RC,nLine);

nRC = size(Zc);

for(i=1:nRC(1))
    for(j=1:nRC(2))
        if(Zc(i,j)==0)
            ZL(i,j)=Zc(i,j);
        else
            ZL(i,j)=Zc(i,j)+LC;
        end
    end
end

XCy = ZL;

YCy = Yc;

ZCy = Xc+RC;

```

```

Xw=0; Yw=0; Zw=0; ZLw=0; nRW=0; XWi=0; YWi=0; ZWi=0;
[Xw,Yw,Zw] = cylinder(RW,nLine);
nRW=size(Zw);
for(i=1:nRW(1))
    for(j=1:nRW(2))
        if(Zw(i,j)==0)
            ZLw(i,j)=Zw(i,j);
        else
            ZLw(i,j)=Zw(i,j)+LW;
        end
    end
end
XWi = ZLw;
YWi = Yw;
ZWi = Xw+RW;
%----- End of Object-----%

LinkA_ObjectX=0; LinkA_ObjectY=0; LinkA_ObjectZ=0;
i=1; LinkA_2X=0; XchkA_Obj=0; YchkA_Obj=0; ZchkA_Obj=0;
ObjA_moving=0;
while(i<=nR1(2))
    LinkA_1X = [0 0];
    LinkA_1Y = [0 0];
    LinkA_1Z = [0 L(2)];
    Ar_LA2x(i) = L(3)*cos(thetaA1(i)*tr)*cos(thetaA2(i)*tr);
    Ar_LA2y(i) = L(3)*sin(thetaA1(i)*tr)*cos(thetaA2(i)*tr);
    Ar_LA2z(i) = L(3)*sin(thetaA2(i)*tr)+L(2);
    Ar_LA3x(i) = L(4)*cos(thetaA1(i)*tr)*cos((thetaA2(i)+thetaA3(i))*tr)+
        L(3)*cos(thetaA1(i)*tr)*cos(thetaA2(i)*tr);

```

Ar_LA3y(i) = L(4)*sin(thetaA1(i)*tr)*cos((thetaA2(i)+thetaA3(i))*tr)+
L(3)*sin(thetaA1(i)*tr)*cos(thetaA2(i)*tr);

Ar_LA3z(i) = L(4)*sin((thetaA2(i)+thetaA3(i))*tr)+L(3)*sin(thetaA2(i)*tr)+L(2);
Ar_LA46x(i) = L_Tool*(cos(thetaA1(i)*tr)*sin(thetaA5(i)*tr)*cos((thetaA2(i)+thetaA3(i)+
thetaA4(i))*tr)+sin(thetaA1(i)*tr)*cos(thetaA5(i)*tr))+
L(4)*cos(thetaA1(i)*tr)*cos((thetaA2(i)+ thetaA3(i))*tr)+
L(3)*cos(thetaA1(i)*tr)*cos(thetaA2(i)*tr);

Ar_LA46y(i) = L_Tool*(sin(thetaA1(i)*tr)*sin(thetaA5(i)*tr)*cos((thetaA2(i)+
thetaA3(i)+thetaA4(i))*tr)-cos(thetaA1(i)*tr)*cos(thetaA5(i)*tr))+
L(4)*sin(thetaA1(i)*tr)*cos((thetaA2(i)+ thetaA3(i))*tr)+
L(3)*sin(thetaA1(i)*tr)*cos(thetaA2(i)*tr);

%----- Define to position variable of Object -----%

Ar_LA46z(i) = L_Tool*sin(thetaA5(i)*tr)*sin((thetaA2(i)+thetaA3(i)+thetaA4(i))*tr)+
L(4)*sin((thetaA2(i)+thetaA3(i))*tr)+L(3)*sin(thetaA2(i)*tr)+L(2);
%----- End -----%

XchkA_Obj(i) = roundn(L(5)*(cos(thetaA1(i)*tr)*sin(thetaA5(i)*tr)*cos((thetaA2(i)+
thetaA3(i)+ thetaA4(i))*tr)+sin(thetaA1(i)*tr)*cos(thetaA5(i)*tr))+
L(4)*cos(thetaA1(i)*tr)*cos((thetaA2(i)+thetaA3(i))*tr)+
L(3)*cos(thetaA1(i)*tr)*cos(thetaA2(i)*tr),0);

YchkA_Obj(i) = roundn(L(5)*(sin(thetaA1(i)*tr)*sin(thetaA5(i)*tr)*cos((thetaA2(i)+
thetaA3(i)+thetaA4(i))*tr)-cos(thetaA1(i)*tr)*cos(thetaA5(i)*tr))+
L(4)*sin(thetaA1(i)*tr)*cos((thetaA2(i)+thetaA3(i))*tr)+
L(3)*sin(thetaA1(i)*tr)*cos(thetaA2(i)*tr),0);

```
ZchkA_Obj(i) = roundn(L(5)*sin(thetaA5(i)*tr)*sin((thetaA2(i)+thetaA3(i)+thetaA4(i))*tr)+
L(4)*sin((thetaA2(i)+thetaA3(i))*tr)+L(3)*sin(thetaA2(i)*tr)+L(2),0);
```

```
for(k=1:2)
```

```
    if(k==1)
```

```
        LinkA_2X(i,k) = 0;
```

```
        LinkA_2Y(i,k) = 0;
```

```
        LinkA_2Z(i,k) = L(2);
```

```
        LinkA_3X(i,k) = Ar_LA2x(i);
```

```
        LinkA_3Y(i,k) = Ar_LA2y(i);
```

```
        LinkA_3Z(i,k) = Ar_LA2z(i);
```

```
        LinkA_46X(i,k) = Ar_LA3x(i);
```

```
        LinkA_46Y(i,k) = Ar_LA3y(i);
```

```
        LinkA_46Z(i,k) = Ar_LA3z(i);
```

```
    else
```

```
        LinkA_2X(i,k) = Ar_LA2x(i);
```

```
        LinkA_2Y(i,k) = Ar_LA2y(i);
```

```
        LinkA_2Z(i,k) = Ar_LA2z(i);
```

```
        LinkA_3X(i,k) = Ar_LA3x(i);
```

```
        LinkA_3Y(i,k) = Ar_LA3y(i);
```

```
        LinkA_3Z(i,k) = Ar_LA3z(i);
```

```
        LinkA_46X(i,k) = Ar_LA46x(i);
```

```
        LinkA_46Y(i,k) = Ar_LA46y(i);
```

```
        LinkA_46Z(i,k) = Ar_LA46z(i);
```

```
    end
```

```
end
```

```

for(k=1:8)
    if(k==1)|(k==4)
        Tool_X1(i,k) = Ar_LA46x(i);
        Tool_Y1(i,k) = Ar_LA46y(i)-(RW);
        Tool_Z1(i,k) = Ar_LA46z(i);
    elseif(k==2)|(k==7)
        Tool_X1(i,k) = Ar_LA46x(i);
        Tool_Y1(i,k) = Ar_LA46y(i)+(RW);
        Tool_Z1(i,k) = Ar_LA46z(i);
    elseif(k==3)|(k==6)
        Tool_X1(i,k) = NaN;
        Tool_Y1(i,k) = NaN;
        Tool_Z1(i,k) = NaN;
    elseif(k==5)
        Tool_X1(i,k) = Ar_LA46x(i);
        Tool_Y1(i,k) = Ar_LA46y(i)-(RW);
        Tool_Z1(i,k) = Ar_LA46z(i)-(2*RW)-length_AV;
    else
        Tool_X1(i,k) = Ar_LA46x(i);
        Tool_Y1(i,k) = Ar_LA46y(i)+(RW);
        Tool_Z1(i,k) = Ar_LA46z(i)-(2*RW)-length_AV;
    end
end

n=0; m=0;

if(ObjA_moving==0)
    if(XchkA_Obj(i)==X1ini(1))&&(YchkA_Obj(i)==Y1ini(1))&&(ZchkA_Obj(i)==Z1ini(1))
        ObjA_moving=1;
    end
end

```

```

for(m=1:2)
    for(n=1:nLine)
        if(m==1)
            LinkA_ObjectX(m,n,i) = Ar_LA46x(i)-XWi(m,n)+(LW/2);
            LinkA_ObjectY(m,n,i) = Ar_LA46y(i)-YWi(m,n);
            LinkA_ObjectZ(m,n,i) = Ar_LA46z(i)-(2*RW)+ZWi(m,n)-length_AV;
        else
            LinkA_ObjectX(m,n,i) = Ar_LA46x(i)-XWi(m,n)+(LW/2);
            LinkA_ObjectY(m,n,i) = Ar_LA46y(i)-YWi(m,n);
            LinkA_ObjectZ(m,n,i) = Ar_LA46z(i)-(2*RW)+ZWi(m,n)-length_AV;
        end
    end
end
else
    for(m=1:2)
        for(n=1:nLine)
            if(m==1)
                LinkA_ObjectX(m,n,i) = X1ini(1)+XWi(m,n)-(LW/2);
                LinkA_ObjectY(m,n,i) = Y1ini(1)+YWi(m,n);
                LinkA_ObjectZ(m,n,i) = Z1ini(1)+ZWi(m,n);
            else
                LinkA_ObjectX(m,n,i) = X1ini(1)+XWi(m,n)-(LW/2);
                LinkA_ObjectY(m,n,i) = Y1ini(1)+YWi(m,n);
                LinkA_ObjectZ(m,n,i) = Z1ini(1)+ZWi(m,n);
            end
        end
    end
end
end
else

```

```

for(m=1:2)
    for(n=1:nLine)
        if(m==1)
            LinkA_ObjectX(m,n,i) = Ar_LA46x(i)-XWi(m,n)+(LW/2);
            LinkA_ObjectY(m,n,i) = Ar_LA46y(i)-YWi(m,n);
            LinkA_ObjectZ(m,n,i) = Ar_LA46z(i)-(2*RW)+ZWi(m,n)-length_AV;
        else
            LinkA_ObjectX(m,n,i) = Ar_LA46x(i)-XWi(m,n)+(LW/2);
            LinkA_ObjectY(m,n,i) = Ar_LA46y(i)-YWi(m,n);
            LinkA_ObjectZ(m,n,i) = Ar_LA46z(i)-(2*RW)+ZWi(m,n)-length_AV;
        end
    end
end
end
i=i+1;
end
min(min(min(LinkA_ObjectZ)))
j=1; XchkB_Obj=0; YchkB_Obj=0; ZchkB_Obj=0;
ObjB_moving = 0;
while(j<=nR2(2))
    LinkB_1X = [0 0];
    LinkB_1Y = [distAB distAB];
    LinkB_1Z = [0 L(2)];
    Ar_LB2x(j) = (-1)*(L(3)*cos(thetaB1(j)*tr)*cos(thetaB2(j)*tr));
    Ar_LB2y(j) = (-1)*(L(3)*sin(thetaB1(j)*tr)*cos(thetaB2(j)*tr));
    Ar_LB2z(j) = L(3)*sin(thetaB2(j)*tr)+L(2);
    Ar_LB3x(j) = (-1)*(L(4)*cos(thetaB1(j)*tr)*cos((thetaB2(j)+thetaB3(j))*tr)+
    L(3)*cos(thetaB1(j)*tr)*cos(thetaB2(j)*tr));
    Ar_LB3y(j) = (-1)*(L(4)*sin(thetaB1(j)*tr)*cos((thetaB2(j)+thetaB3(j))*tr)+
    L(3)*sin(thetaB1(j)*tr)*cos(thetaB2(j)*tr));

```

$$\begin{aligned}
Ar_LB3z(j) &= L(4)*\sin((\theta_{B2}(j)+\theta_{B3}(j))*tr)+L(3)*\sin(\theta_{B2}(j)*tr)+L(2); \\
Ar_LB46x(j) &= (-1)*(L_Tool*(\cos(\theta_{B1}(j)*tr)*\sin(\theta_{B5}(j)*tr)*\cos((\theta_{B2}(j)+ \\
&\theta_{B3}(j)+\theta_{B4}(j))*tr)+\sin(\theta_{B1}(j)*tr)*\cos(\theta_{B5}(j)*tr))+ \\
&L(4)*\cos(\theta_{B1}(j)*tr)*\cos((\theta_{B2}(j)+\theta_{B3}(j))*tr)+ \\
&L(3)*\cos(\theta_{B1}(j)*tr)*\cos(\theta_{B2}(j)*tr));
\end{aligned}$$

$$\begin{aligned}
Ar_LB46y(j) &= (-1)*(L_Tool*(\sin(\theta_{B1}(j)*tr)*\sin(\theta_{B5}(j)*tr)*\cos((\theta_{B2}(j)+ \\
&\theta_{B3}(j)+\theta_{B4}(j))*tr)-\cos(\theta_{B1}(j)*tr)*\cos(\theta_{B5}(j)*tr))+ \\
&L(4)*\sin(\theta_{B1}(j)*tr)*\cos((\theta_{B2}(j)+\theta_{B3}(j))*tr)+ \\
&L(3)*\sin(\theta_{B1}(j)*tr)*\cos(\theta_{B2}(j)*tr));
\end{aligned}$$

$$\begin{aligned}
Ar_LB46z(j) &= L_Tool*\sin(\theta_{B5}(j)*tr)*\sin((\theta_{B2}(j)+\theta_{B3}(j)+\theta_{B4}(j))*tr)+ \\
&L(4)*\sin((\theta_{B2}(j)+\theta_{B3}(j))*tr)+L(3)*\sin(\theta_{B2}(j)*tr)+L(2);
\end{aligned}$$

$$\begin{aligned}
XchkB_Obj(j) &= \text{roundn}((L(5)*(\cos(\theta_{B1}(j)*tr)*\sin(\theta_{B5}(j)*tr)*\cos((\theta_{B2}(j)+ \\
&\theta_{B3}(j)+\theta_{B4}(j))*tr)+\sin(\theta_{B1}(j)*tr)*\cos(\theta_{B5}(j)*tr))+ \\
&L(4)*\cos(\theta_{B1}(j)*tr)*\cos((\theta_{B2}(j)+\theta_{B3}(j))*tr)+ \\
&L(3)*\cos(\theta_{B1}(j)*tr)*\cos(\theta_{B2}(j)*tr)),0);
\end{aligned}$$

$$\begin{aligned}
YchkB_Obj(j) &= \text{roundn}((L(5)*(\sin(\theta_{B1}(j)*tr)*\sin(\theta_{B5}(j)*tr)*\cos((\theta_{B2}(j)+ \\
&\theta_{B3}(j)+\theta_{B4}(j))*tr)-\cos(\theta_{B1}(j)*tr)*\cos(\theta_{B5}(j)*tr))+ \\
&L(4)*\sin(\theta_{B1}(j)*tr)*\cos((\theta_{B2}(j)+\theta_{B3}(j))*tr)+ \\
&L(3)*\sin(\theta_{B1}(j)*tr)*\cos(\theta_{B2}(j)*tr)),0);
\end{aligned}$$

$$\begin{aligned}
ZchkB_Obj(j) &= \text{roundn}(L(5)*\sin(\theta_{B5}(j)*tr)*\sin((\theta_{B2}(j)+\theta_{B3}(j)+\theta_{B4}(j))*tr)+ \\
&L(4)*\sin((\theta_{B2}(j)+\theta_{B3}(j))*tr)+L(3)*\sin(\theta_{B2}(j)*tr)+L(2),0);
\end{aligned}$$

```

for(k=1:2)
    if(k==1)
        LinkB_2X(j,k) = 0;
        LinkB_2Y(j,k) = distAB;
        LinkB_2Z(j,k) = L(2);
        LinkB_3X(j,k) = Ar_LB2x(j);
        LinkB_3Y(j,k) = distAB+Ar_LB2y(j);
        LinkB_3Z(j,k) = Ar_LB2z(j);
        LinkB_46X(j,k) = Ar_LB3x(j);
        LinkB_46Y(j,k) = distAB+Ar_LB3y(j);
        LinkB_46Z(j,k) = Ar_LB3z(j);
    else
        LinkB_2X(j,k) = Ar_LB2x(j);
        LinkB_2Y(j,k) = distAB+Ar_LB2y(j);
        LinkB_2Z(j,k) = Ar_LB2z(j);
        LinkB_3X(j,k) = Ar_LB3x(j);
        LinkB_3Y(j,k) = distAB+Ar_LB3y(j);
        LinkB_3Z(j,k) = Ar_LB3z(j);
        LinkB_46X(j,k) = Ar_LB46x(j);
        LinkB_46Y(j,k) = distAB+Ar_LB46y(j);
        LinkB_46Z(j,k) = Ar_LB46z(j);
    end
end

%-----calculate Tool position for Robot B-----%
for(k=1:8)
    if(k==1)|(k==4)
        Tool_X2(j,k) = Ar_LB46x(j);
        Tool_Y2(j,k) = distAB+Ar_LB46y(j)-(RC);
        Tool_Z2(j,k) = Ar_LB46z(j);
    end
end

```

```

elseif(k==2)|(k==7)
    Tool_X2(j,k) = Ar_LB46x(j);
    Tool_Y2(j,k) = distAB+Ar_LB46y(j)+(RC);
    Tool_Z2(j,k) = Ar_LB46z(j);
elseif(k==3)|(k==6)
    Tool_X2(j,k) = NaN;
    Tool_Y2(j,k) = NaN;
    Tool_Z2(j,k) = NaN;
elseif(k==5)
    Tool_X2(j,k) = Ar_LB46x(j);
    Tool_Y2(j,k) = distAB+Ar_LB46y(j)-(RC);
    Tool_Z2(j,k) = Ar_LB46z(j)-(2*(RC+Rdif))-length_AV;
else
    Tool_X2(j,k) = Ar_LB46x(j);
    Tool_Y2(j,k) = distAB+Ar_LB46y(j)+(RC);
    Tool_Z2(j,k) = Ar_LB46z(j)-(2*(RC+Rdif))-length_AV;
end
end

%----- Calculate Object Position for Cylinder of Robot B-----%
n=0; m=0;
if(ObjB_moving==0)
    if(XchkB_Obj(j)==X2ini(1))&&(YchkB_Obj(j)==Y2ini(1))&&(ZchkB_Obj(j)==Z2ini(1))
        ObjB_moving=1;
        for(m=1:2)
            for(n=1:nLine)
                if(m==1)
                    LinkB_ObjectX(m,n,j) = Ar_LB46x(j)-XCy(m,n)+(LC/2);
                    LinkB_ObjectY(m,n,j) = distAB+Ar_LB46y(j)-YCy(m,n);
                    LinkB_ObjectZ(m,n,j) = Ar_LB46z(j)-(2*(RC+Rdif))+ZCy(m,n)-length_AV;

```

```

else
    LinkB_ObjectX(m,n,j) = Ar_LB46x(j)-XCy(m,n)+(LC/2);
    LinkB_ObjectY(m,n,j) = distAB+Ar_LB46y(j)-YCy(m,n);
    LinkB_ObjectZ(m,n,j) = Ar_LB46z(j)-(2*(RC+Rdif))+ZCy(m,n)-length_AV;
end
end
end
else
    for(m=1:2)
        for(n=1:nLine)
            if(m==1)
                LinkB_ObjectX(m,n,j) = (-1)*X2ini(1)+XCy(m,n)-(LC/2);
                LinkB_ObjectY(m,n,j) = distAB-Y2ini(1)+YCy(m,n);
                LinkB_ObjectZ(m,n,j) = Z2ini(1)+ZCy(m,n);
            else
                LinkB_ObjectX(m,n,j) = (-1)*X2ini(1)+XCy(m,n)-(LC/2);
                LinkB_ObjectY(m,n,j) = distAB-Y2ini(1)+YCy(m,n);
                LinkB_ObjectZ(m,n,j) = Z2ini(1)+ZCy(m,n);
            end
        end
    end
end
end
else
    for(m=1:2)
        for(n=1:nLine)
            if(m==1)
                LinkB_ObjectX(m,n,j) = Ar_LB46x(j)-XCy(m,n)+(LC/2);
                LinkB_ObjectY(m,n,j) = distAB+Ar_LB46y(j)-YCy(m,n);
                LinkB_ObjectZ(m,n,j) = Ar_LB46z(j)-(2*(RC+Rdif))+ZCy(m,n)-length_AV;
            else

```

```

        LinkB_ObjectX(m,n,j) = Ar_LB46x(j)-XCy(m,n)+(LC/2);
        LinkB_ObjectY(m,n,j) = distAB+Ar_LB46y(j)-YCy(m,n);
        LinkB_ObjectZ(m,n,j) = Ar_LB46z(j)-(2*(RC+Rdif))+ZCy(m,n)-length_AV;
    end
end
end
end
j=j+1;
end

%----- Show to animation of Operation Robot A and B (Graphic Animation)-----%
for(i=1:nRound)

    plot3(LinkA_1X, LinkA_1Y, LinkA_1Z,'LineWidth',10);

    grid on;

    hold on;

    axis([-50 50 -70 70 0 50]);

    xlabel('X axis');

    ylabel('Y axis');

    zlabel('Z axis');

    %----- Graphic Animation for A -----%

    piture4 = plot3(LinkA_2X(i,:), LinkA_2Y(i,:), LinkA_2Z(i,:), 'LineWidth',8);
    set(piture4 , 'color',[0.5 0.7 0.9]);

    piture5 = plot3(LinkA_3X(i,:), LinkA_3Y(i,:), LinkA_3Z(i,:), 'LineWidth',8);
    set(piture5,'color',[0.5 0.2 0.4]);

    piture6 = plot3(LinkA_46X(i,:), LinkA_46Y(i,:), LinkA_46Z(i,:), 'LineWidth',8);
    set(piture6,'color',[0.2 0.7 0.1]);

    plot3(Tool_X1(i,:), Tool_Y1(i,:), Tool_Z1(i,:), 'LineWidth',3);

    piture7 = plot3(LinkA_ObjectX(:,i), LinkA_ObjectY(:,i),
        LinkA_ObjectZ(:,i), 'LineWidth',3);

```

```

set(piture7,'color',[0.2 0.4 0.3]);

%----- Graphic Animation for B -----%

plot3(LinkB_1X, LinkB_1Y, LinkB_1Z,'LineWidth',10);
piture1 = plot3(LinkB_2X(i,:), LinkB_2Y(i,:), LinkB_2Z(i,:), 'LineWidth',8);
set(piture1 , 'color',[0.5 0.7 0.9]);
piture2 = plot3(LinkB_3X(i,:), LinkB_3Y(i,:), LinkB_3Z(i,:), 'LineWidth',8);
set(piture2,'color',[0.5 0.2 0.4]);
piture3 = plot3(LinkB_46X(i,:), LinkB_46Y(i,:), LinkB_46Z(i,:), 'LineWidth',8);
set(piture3,'color',[0.2 0.7 0.1]);
piture4 = plot3(Tool_X2(i,:), Tool_Y2(i,:), Tool_Z2(i,:), 'LineWidth',3);
set(piture4,'color',[0.7 0.5 0.6]);
piture8 = plot3(LinkB_ObjectX(:,i), LinkB_ObjectY(:,i),
LinkB_ObjectZ(:,i), 'LineWidth',3);
set(piture8,'color',[0.5 0.7 0.3]);
hold off;
M(i)=getframe;
end
movie(M,1,15);

%----- Describe to Function Calculate_Position_Robot -----%
Function [PAX1, PAY1, PAZ1, PAX2, PAY2, PAZ2,Result_R12, Result_R22, Path_Error,
Opt_TAB, theta_RobotA, theta_RobotB]= Cal_PIT_RB3(X1, Y1, Z1, X2, Y2, Z2, p, R, Acc, d,
d6, AV )

%----- Define variable for Operation Prgram -----%
Path_Error=0; Px1=0; Py1=0; Pz1=0; Opt_TAB=0; Opt_T1=0;
Px2=0; Py2=0; Pz2=0; Opt_T2=0;
Nze_11=0; Nze_12=0; Nze_13=0; Nze_14=0; Nze_15=0; Nze_16=0; T1=0;
Nze_21=0; Nze_22=0; Nze_23=0; Nze_24=0; Nze_25=0; Nze_26=0; T2=0;
Pnx1=0; Pny1=0; Pnz1=0; Pnx2=0; Pny2=0; Pnz2=0;

```

```

%----- Unit is radius per second -----%

Max_AV=AV*(2*pi)/60;

%----- Define variable for Operation Prgram -----%

NX1=0; NY1=0; NZ1=0; NX2=0; NY2=0; NZ2=0;

NX1=line(X1, p(1));
NY1=line(Y1, p(1));
NZ1=line(Z1, p(1));
NX2=line(X2, p(2));
NY2=line(Y2, p(2));
NZ2=line(Z2, p(2));

ze_11=0; ze_12=0; ze_13=0; ze_14=0; ze_15=0; ze_16=0;
ze_21=0; ze_22=0; ze_23=0; ze_24=0; ze_25=0; ze_26=0;

[ze_11 ze_12 ze_13 ze_14 ze_15 ze_16]=Cal_Angle(NX1, NY1, NZ1, d6, R, d);
[ze_21 ze_22 ze_23 ze_24 ze_25 ze_26]=Cal_Angle(NX2, NY2, NZ2, d6, R, d);
[Opt_T1, T1]=Cal_Time3(ze_11, ze_12, ze_13, ze_14, ze_15, ze_16, Acc, Max_AV);
[Opt_T2, T2]=Cal_Time3(ze_21, ze_22, ze_23, ze_24, ze_25, ze_26, Acc, Max_AV);
c1=size(ze_11);
c2=size(ze_21);
if(c1(2)==c2(2))
    for(j=1:c1(2))
        for(i=1:c1(1))
            [Px1(j,i), Py1(j,i), Pz1(j,i)]=FWD_KMT(ze_11(i,j), ze_12(i,j), ze_13(i,j), ze_14(i,j),
            ze_15(i,j), ze_16(i,j), d6, d);
            [Px2(j,i), Py2(j,i), Pz2(j,i)]=FWD_KMT(ze_21(i,j), ze_22(i,j), ze_23(i,j), ze_24(i,j),
            ze_25(i,j), ze_26(i,j), d6, d);
        end
    end
else

```

```

for(j=1:c1(2))
    for(i=1:c1(1))
        [Px1(j,i), Py1(j,i), Pz1(j,i)]=FWD_KMT(ze_11(i,j), ze_12(i,j), ze_13(i,j), ze_14(i,j),
            ze_15(i,j), ze_16(i,j), d6, d);
    end
end
for(j=1:c2(2))
    for(i=1:c2(1))
        [Px2(j,i), Py2(j,i), Pz2(j,i)]=FWD_KMT(ze_21(i,j), ze_22(i,j), ze_23(i,j), ze_24(i,j),
            ze_25(i,j), ze_26(i,j), d6, d);
    end
end
end
end

NNX2=0; NNY2=0; NNZ2=0; NNX1=0; NNY1=0; NNZ1=0;
if(Opt_T1==Opt_T2)
    [Nze_11, Nze_12, Nze_13, Nze_14, Nze_15, Nze_16]=Focast_angle3(ze_11, ze_12, ze_13,
        ze_14, ze_15, ze_16, Acc, Max_AV);
    [Nze_21, Nze_22, Nze_23, Nze_24, Nze_25, Nze_26]=Focast_angle3(ze_21, ze_22, ze_23,
        ze_24, ze_25, ze_26, Acc, Max_AV);
    Opt_TAB=[Opt_T1 Opt_T2];
elseif(Opt_T1>Opt_T2)
    [Nze_11, Nze_12, Nze_13, Nze_14, Nze_15, Nze_16]=Focast_angle3(ze_11, ze_12, ze_13,
        ze_14, ze_15, ze_16, Acc, Max_AV);

    DM2=0; h=0;
    DM2=length(NX2); NOpt_T2=0; q=0;
    for(s=1:(DM2-1))
        NNX2(s)=NX2(s);
        NNY2(s)=NY2(s);
    end
end

```

```

    NNZ2(s)=NZ2(s);
end
system=1;
while(Opt_T1>NOpt_T2)&&(system==1)
    q=q+1; NT2=0;
    h=q+(DM2-1);
    if(NX2(DM2-1)>NX2(DM2))
        NNX2(h)=NNX2(h-1)-0.005;
        if(NNX2(h)<=NX2(DM2))
            system=0;
            NNX2(h)= NX2(DM2)
        else
            system=1;
        end
    elseif(NX2(DM2-1)<NX2(DM2))
        NNX2(h)=NNX2(h-1)+0.005;
        if(NNX2(h)>=NX2(DM2))
            system=0;
            NNX2(h)= NX2(DM2)
        else
            system=1;
        end
    else
        NNX2(h)=NX2(h-1);
        system=1;
    end

    NNY2(h)=NNY2(h-1)+((NY2(DM2)-NY2(DM2-1))/(NX2(DM2)-
    NX2(DM2-1)))*(NNX2(h)-NNX2(DM2-1));

```

```

NNZ2(h)=NNZ2(h-1)+((NZ2(DM2)-NZ2(DM2-1))/(NX2(DM2)-
NX2(DM2-1)))*(NNX2(h)-NNX2(DM2-1));

```

```

if(NNZ2(h)<=NY2(DM2))

```

```

    system=0;

```

```

    NNX2(h)= NX2(DM2)

```

```

    NNY2(h)= NY2(DM2);

```

```

    NNZ2(h)= NZ2(DM2);

```

```

else

```

```

    system=1;

```

```

    NNY2(h+1)=NNY2(DM2);

```

```

    NNZ2(h+1)=NNZ2(DM2);

```

```

    NNX2(h+1)=NNX2(DM2);

```

```

end

```

```

NNX1=NX1;

```

```

NNY1=NY1;

```

```

NNZ1=NZ1;

```

```

nze_21=0; nze_22=0; nze_23=0; nze_24=0; nze_25=0; nze_26=0;

```

```

[nze_21 nze_22 nze_23 nze_24 nze_25 nze_26]=Cal_Angle(NNX2, NNY2, NNZ2,
d6, R, d);

```

```

[NOpt_T2, NT2]=Cal_Time3(nze_21, nze_22, nze_23, nze_24, nze_25, nze_26,
Acc, Max_AV);

```

```

[Nze_21, Nze_22, Nze_23, Nze_24, Nze_25, Nze_26]=Focast_angle3(nze_21, nze_22,
nze_23, nze_24, nze_25, nze_26, Acc, Max_AV);

```

```

end

```

```

Opt_TAB=[Opt_T1 NOpt_T2];

```

```

else

```

```

[Nze_21, Nze_22, Nze_23, Nze_24, Nze_25, Nze_26]=Focast_angle3(ze_21, ze_22, ze_23,
ze_24, ze_25, ze_26, Acc, Max_AV);

```

```

DM1=0; h=0;
DM1=length(NX1); NOpt_T1=0; q=0;
for(s=1:(DM1-1))
    NNX1(s)=NX1(s);
    NNY1(s)=NY1(s);
    NNZ1(s)=NZ1(s);
end
system=1;
while(Opt_T2>NOpt_T1)&&(system==1)
    q=q+1; NT1=0;
    h=q+(DM1-1);
    if(NX1(DM1-1)>NX1(DM1))
        NNX1(h)=NNX1(h-1)-0.005;
        if(NNX1(h)<=NX1(DM1))
            system=0;
            NNX1(h)= NX1(DM1)
        else
            system=1;
        end
    elseif(NX1(DM1-1)<NX1(DM1))
        NNX1(h)=NNX1(h-1)+0.005;
        if(NNX1(h)>=NX1(DM1))
            system=0;
            NNX1(h)= NX1(DM1)
        else
            system=1;
        end
    end
else

```

```

    NNX1(h)=NX1(h-1);

    system=1;

end

NNY1(h)=NNY1(h-1)+((NY1(DM1)-NY1(DM1-1))/(NX1(DM1)-
NX1(DM1-1)))*(NNX1(h)-NX1(DM1-1));

NNZ1(h)=NNZ1(h-1)+((NZ1(DM1)-NZ1(DM1-1))/(NX1(DM1)-
NX1(DM1-1)))*(NNX1(h)-NX1(DM1-1));

if(NNZ1(h)<=NY1(DM1))

    system=0;

    NNX1(h)= NX1(DM1);

    NNY1(h)= NY1(DM1);

    NNZ1(h)= NZ1(DM1);

else

    system=1;

    NNY1(h+1)=NNY1(DM1);

    NNZ1(h+1)=NNZ1(DM1);

    NNX1(h+1)=NNX1(DM1);

end

NNX2=NX2;

NNY2=NY2;

NNZ2=NZ2;


nze_11=0; nze_12=0; nze_13=0; nze_14=0; nze_15=0; nze_16=0;

[nze_11 nze_12 nze_13 nze_14 nze_15 nze_16]=Cal_Angle(NNX1, NNY1, NNZ1, d6,
R, d);

[NOpt_T1, NT1]=Cal_Time3(nze_11, nze_12, nze_13, nze_14, nze_15, nze_16, Acc,
Max_AV);

[Nze_11, Nze_12, Nze_13, Nze_14, Nze_15, Nze_16]=Focast_angle3(nze_11, nze_12,
nze_13, nze_14, nze_15, nze_16, Acc, Max_AV);

```

```

end

Opt_TAB=[NOpt_T1 Opt_T2];

end

%----- End loop for While -----%

f1=size(Nze_11);
f2=size(Nze_21);
for(i=1:f1(1))
    for(j=1:f1(2))
        [Pnx1(i,j), Pny1(i,j), Pnz1(i,j)]=FWD_KMT(Nze_11(i,j), Nze_12(i,j), Nze_13(i,j),
            Nze_14(i,j), Nze_15(i,j), Nze_16(i,j), d6, d);
    end
end
for(i=1:f2(1))
    for(j=1:f2(2))
        [Pnx2(i,j), Pny2(i,j), Pnz2(i,j)]=FWD_KMT(Nze_21(i,j), Nze_22(i,j), Nze_23(i,j),
            Nze_24(i,j), Nze_25(i,j), Nze_26(i,j), d6, d);
    end
end
end

Path_ErA=0; Path_ErB=0; PAX1=0; PAX2=0; PAY1=0; PAY2=0; PAZ1=0; PAZ2=0;
if(Opt_T1==Opt_T2)
    [PAX1, PAY1, PAZ1, Path_ErA]=Check_Er(NX1, NY1, NZ1, Pnx1, Pny1, Pnz1, Acc, T1);
    [PAX2, PAY2, PAZ2, Path_ErB]=Check_Er(NX2, NY2, NZ2, Pnx2, Pny2, Pnz2, Acc, T2);
    Path_Error=[Path_ErA Path_ErB];
elseif(Opt_T1>Opt_T2)
    [PAX1, PAY1, PAZ1, Path_ErA]=Check_Er(NX1, NY1, NZ1, Pnx1, Pny1, Pnz1, Acc, T1);
    [PAX2, PAY2, PAZ2, Path_ErB]=Check_Er(NNX2, NNY2, NNZ2, Pnx2, Pny2, Pnz2, Acc,
    NT2);
    Path_Error=[Path_ErA Path_ErB];

```

```

else
    [PAX2, PAY2, PAZ2, Path_ErB]=Check_Er(NX2, NY2, NZ2, Pnx2, Pny2, Pnz2, Acc, T2);
    [PAX1, PAY1, PAZ1, Path_ErA]=Check_Er(NNX1, NNY1, NNZ1, Pnx1, Pny1, Pnz1, Acc,
    NT1);
    Path_Error=[Path_ErA Path_ErB];
end

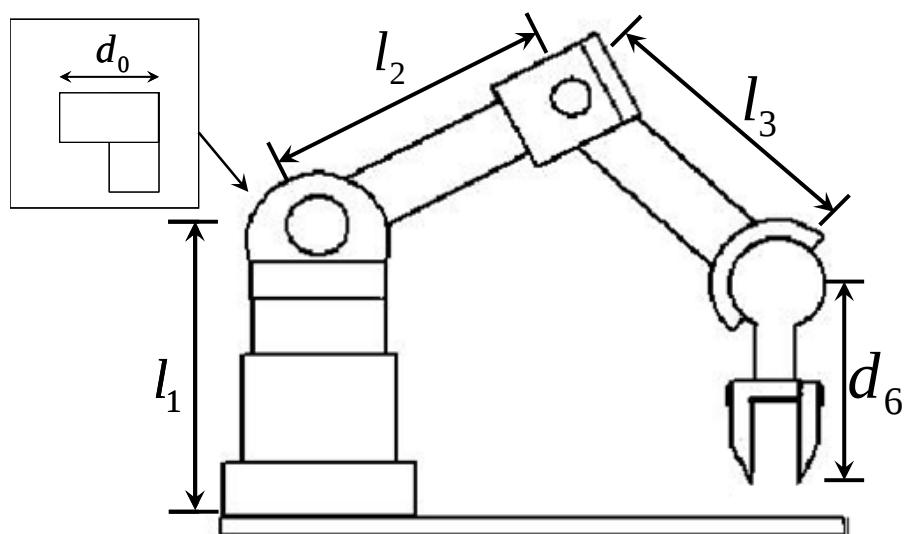
Result_R12=0; Result_R22=0;
for(i=1:length(NNX1))
    Result_R12(1,i)=NNX1(i);
    Result_R12(2,i)=NNY1(i);
    Result_R12(3,i)=NNZ1(i);
End

for(i=1:length(NNX2))
    Result_R22(1,i)=NNX2(i);
    Result_R22(2,i)=NNY2(i);
    Result_R22(3,i)=NNZ2(i);
end

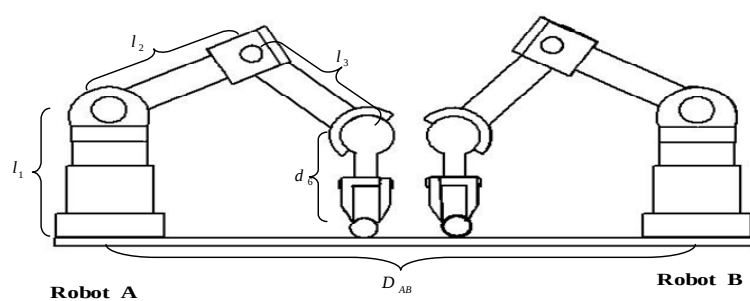
theta_RobotA=0; theta_RobotB=0;
nA=0; nB=0;
nA=size(Nze_11);
nB=size(Nze_21);
for(m=1:max(nA))
    theta_RobotA(1,m)=Nze_11(m);
    theta_RobotA(2,m)=Nze_12(m);
    theta_RobotA(3,m)=Nze_13(m);
    theta_RobotA(4,m)=Nze_14(m);
    theta_RobotA(5,m)=Nze_15(m);

```

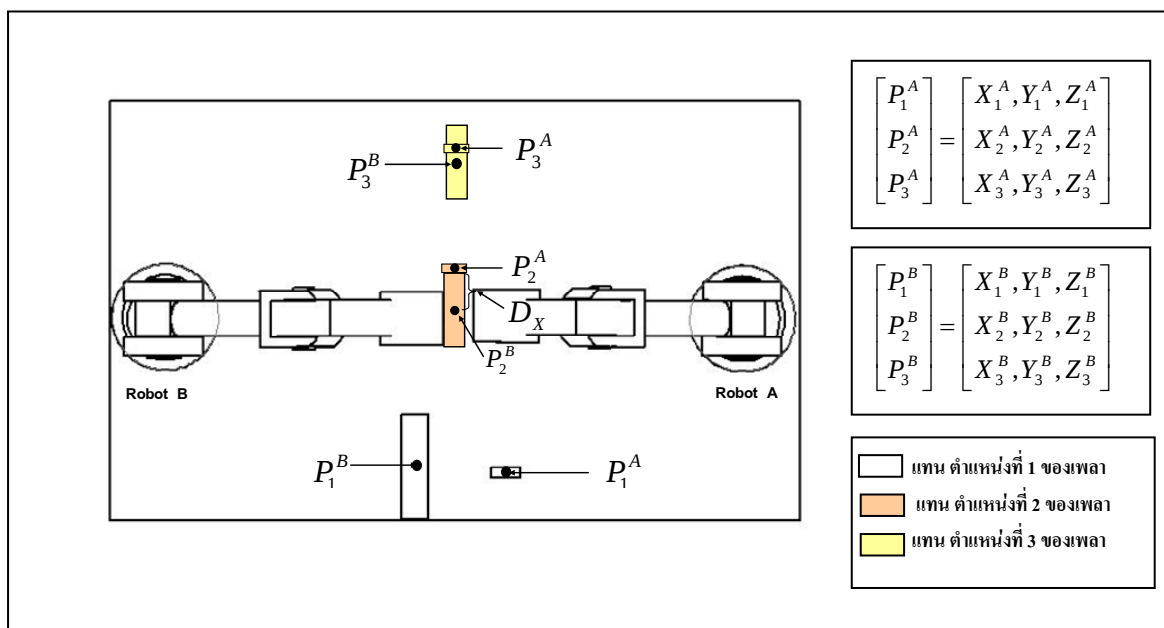
```
    theta_RobotA(6,m)=Nze_16(m);  
end  
  
for(n=1:max(nB))  
    theta_RobotB(1,n)=Nze_21(n);  
    theta_RobotB(2,n)=Nze_22(n);  
    theta_RobotB(3,n)=Nze_23(n);  
    theta_RobotB(4,n)=Nze_24(n);  
    theta_RobotB(5,n)=Nze_25(n);  
    theta_RobotB(6,n)=Nze_26(n);  
end  
end
```



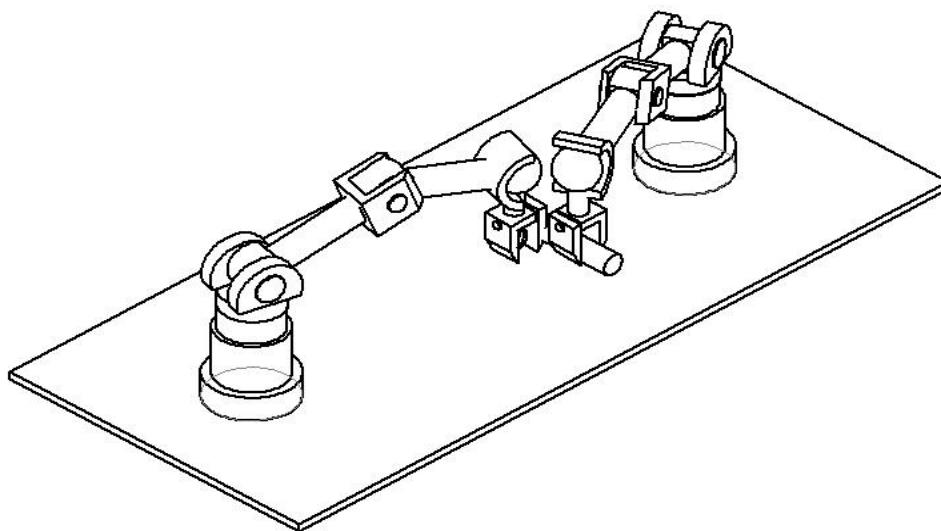
ภาพผนวกที่ 9 แสดงคุณลักษณะต่างๆ ของหุ่นยนต์สกาล่าหกแกน



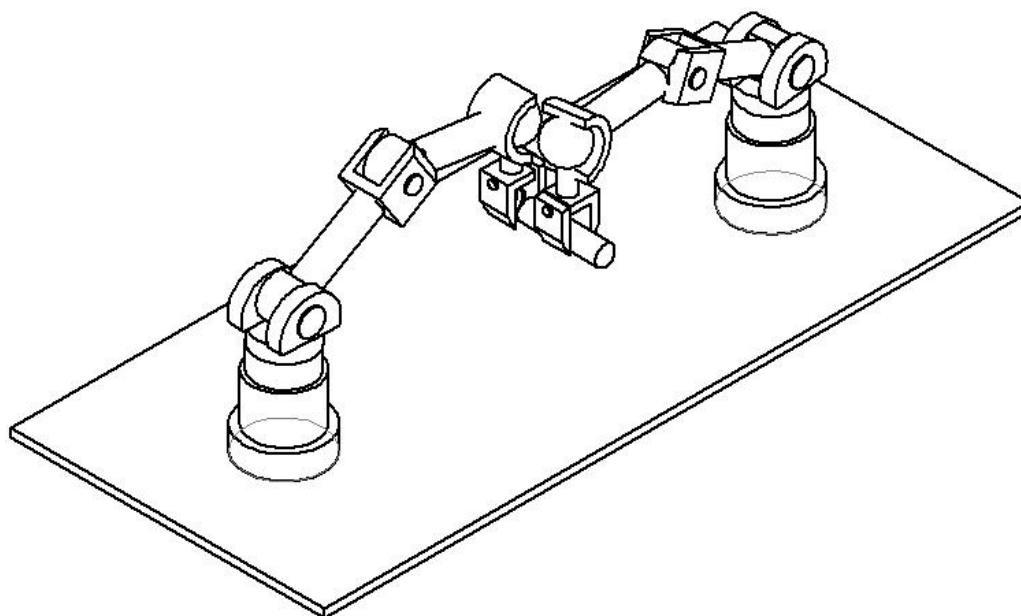
ภาพผนวกที่ 10 แสดงลักษณะการวางของหุ่นยนต์สกาล่าสองตัวมาทำงานร่วมกัน



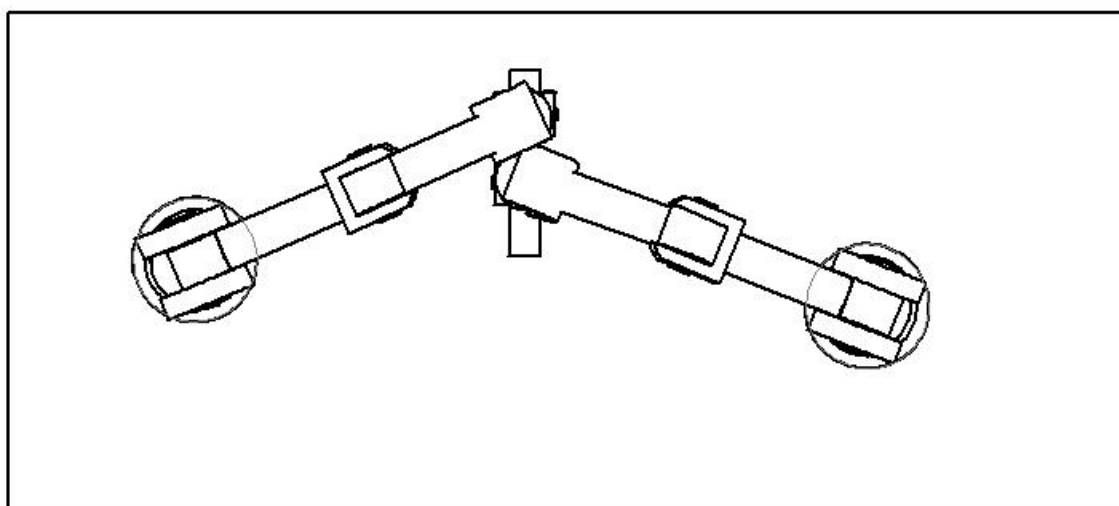
ภาพผนวกที่ 11 แสดงการกำหนดตำแหน่งการเคลื่อนที่ของหุ่นยนต์ A และ B



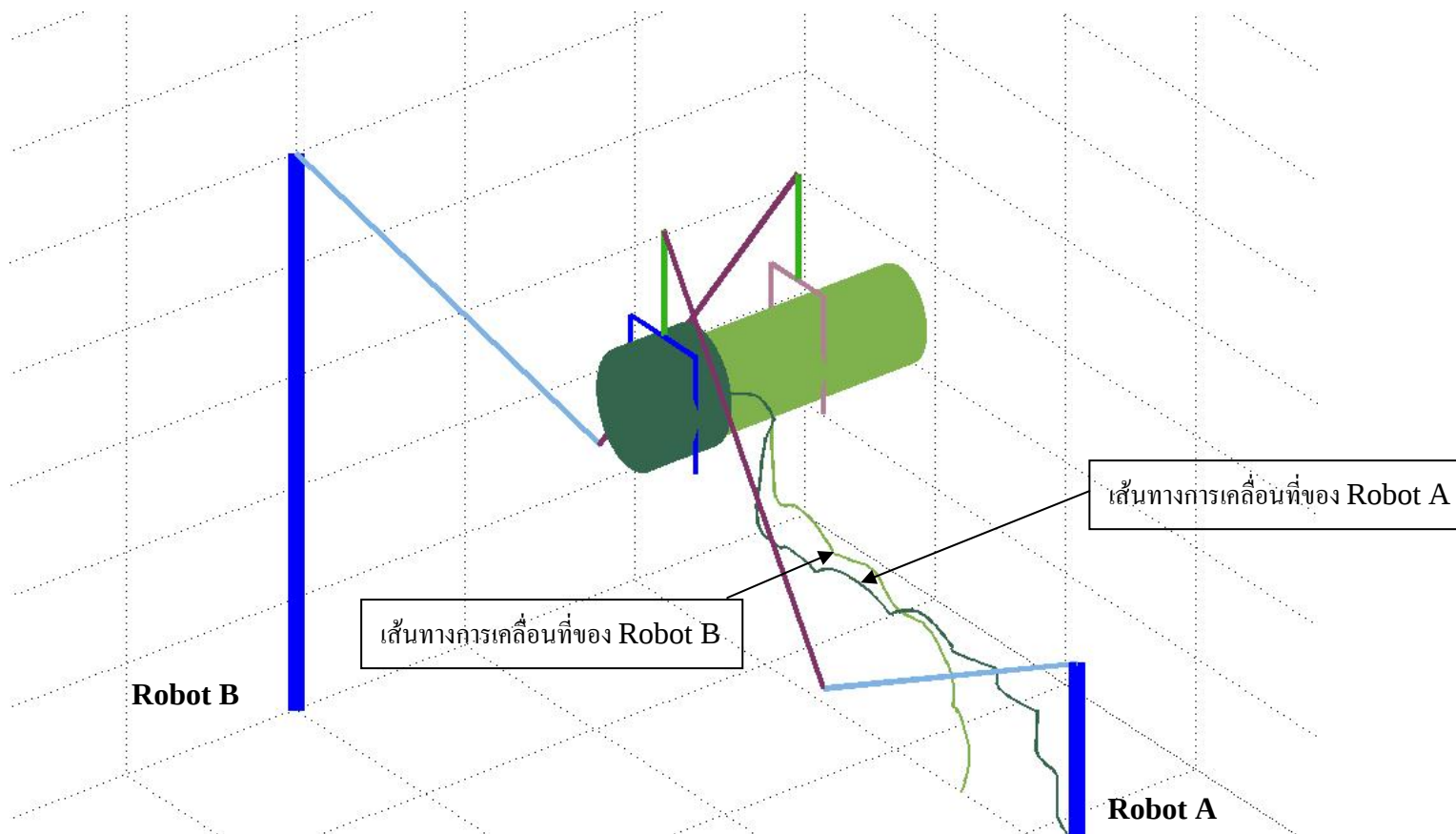
ภาพผนวกที่ 12 แสดงตำแหน่งการชนระหว่างเคลื่อนที่ของหุ่นยนต์ที่มีการเคลื่อนที่แบบเส้นตรง



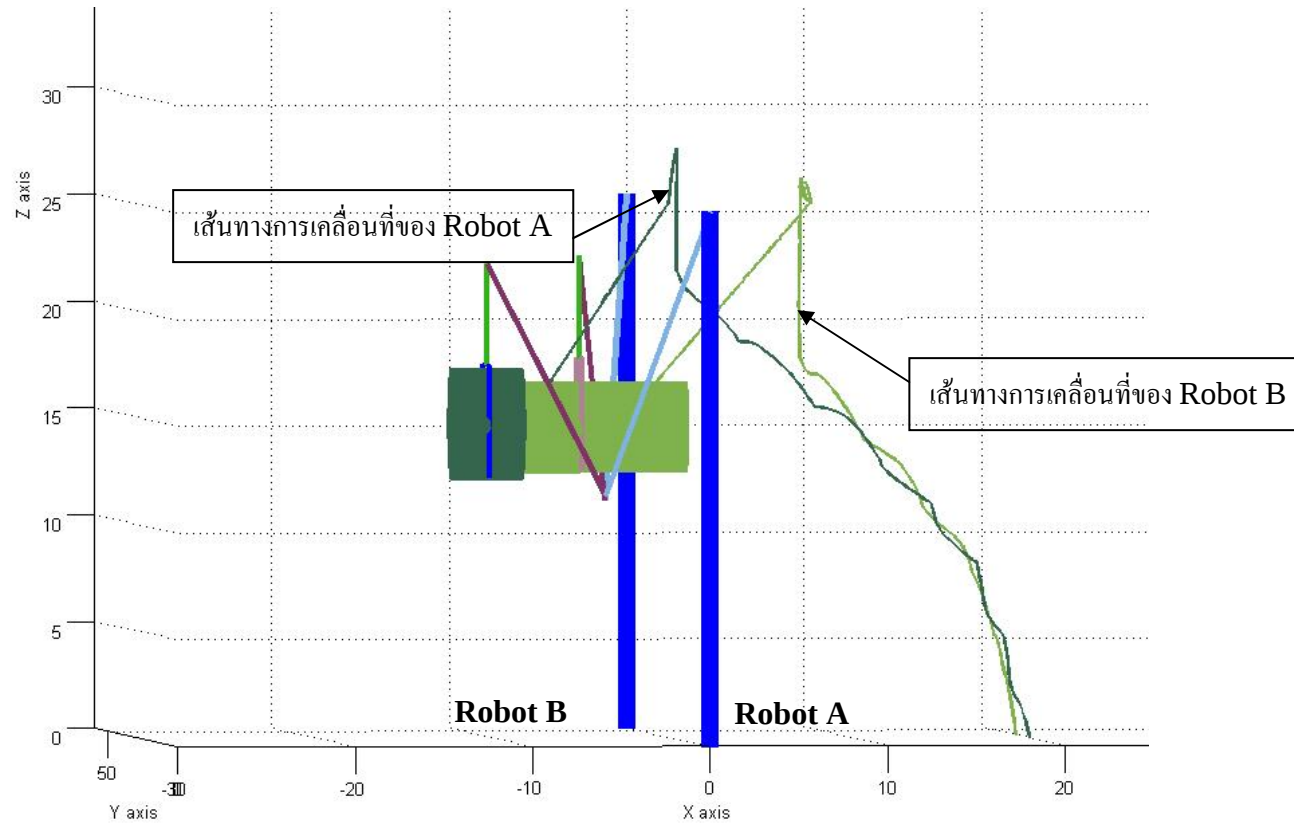
ภาพผนวกที่ 13 แสดงตำแหน่งพร้อมสำหรับการประกอบชิ้นงานของการเคลื่อนที่แบบแบ่งช่วง



ภาพผนวกที่ 14 แสดงตำแหน่งสุดท้ายของการเคลื่อนที่แบบแบ่งช่วง



ภาพผนวกที่ 15 แสดงลักษณะการเคลื่อนที่ของหุ่นยนต์ในช่วงแรกของการเคลื่อนที่



ภาพผนวกที่ 16 แสดงลักษณะการเคลื่อนที่ของหุ่นยนต์ในช่วงสุดท้ายของการเคลื่อนที่

ประวัติการศึกษา และการทำงาน

ชื่อ –นามสกุล	นายสว่าง แป้นจันทร์
วัน เดือน ปี ที่เกิด	1 ธันวาคม 2522
สถานที่เกิด	อำเภอเมือง จังหวัดนครสวรรค์
ประวัติการศึกษา	วท.บ. (สถิติ) มหาวิทยาลัยแม่โจ้
ตำแหน่งหน้าที่การงานปัจจุบัน	
สถานที่ทำงานปัจจุบัน	
ผลงานดีเด่นและรางวัลทางวิชาการ	
ทุนการศึกษาที่ได้รับ	ได้รับทุนผู้ช่วยสอนจากภาควิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ (พ.ศ.2549)