

บทที่ 2

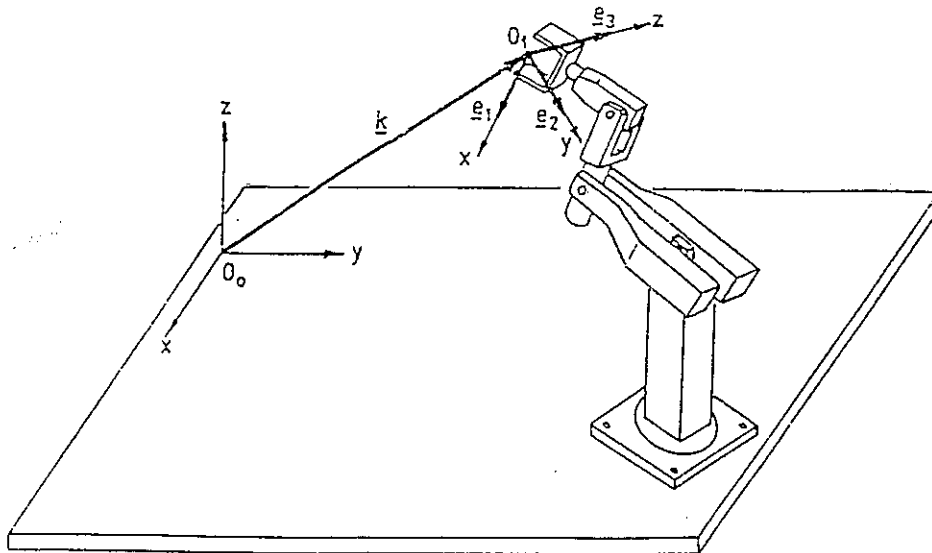
ทฤษฎีและหลักการ

2.1 การแปลงพิกัดร่วมของหุ่นยนต์

(Homogeneous coordination transformer)

ในการที่บอกพิกัดใด ๆ ของวัตถุจะใช้ตำแหน่งและมุมภายในพิกัด เพื่อจะแสดงตำแหน่งที่วัตถุอยู่นั้น ทั้งนี้มีความจำเป็นเพื่อที่จะนิยามตำแหน่งและมุมที่สัมพันธ์กับระบบพิกัด รวดเร็วและเห็นได้ชัด และการใช้การแปลงพิกัดร่วม (Homogeneous transformer) เป็นวิธีที่มีประโยชน์มาก

ให้จุด O_0 คือ ฐานของระบบพิกัด (Base coordinate system) กำหนดให้เวกเตอร์ k บอกตำแหน่งของระบบพิกัดอื่นอีกระบบพิกัดที่มีจุดกำเนิด O_1 ในฐานของระบบพิกัด O_0 เวกเตอร์ k บอกเพียงตำแหน่ง O_1 เท่านั้น โดยไม่ได้บอกว่า มุมมีความสัมพันธ์อย่างไรกับระบบพิกัด O_0 แต่มุมสัมพันธ์สามารถบอกได้ โดยเมตริกซ์ R ซึ่งเป็นเวกเตอร์ สำหรับการหมุนเป็นเมตริกซ์ 3×3 ในเมตริกซ์การหมุน R ประกอบด้วยเวกเตอร์ 1 หน่วย (e_1, e_2, e_3) ของระบบพิกัด O_1 ตามแนวแกนของระบบพิกัด ดังแสดงในรูปที่ 2.1



รูปที่ 2.1 แสดงการอ้างอิงพิกัดฐานและการแปลงระบบพิกัด (Translation coordinate system)

เวกเตอร์ 1 หน่วย (e_1, e_2, e_3) ของระบบพิกัดที่ใช้ในการหาการแปลงระบบพิกัดการหมุน จากความสัมพันธ์ของเวกเตอร์และสเกลลา

$$e_1 = e_2 \times e_3 \quad \text{คือการคูณกันทางเวกเตอร์ (Cross product)}$$

$$e_2 \cdot e_3 = 0 \quad \text{คือการคูณกันทางสเกลลา (Inner product)}$$

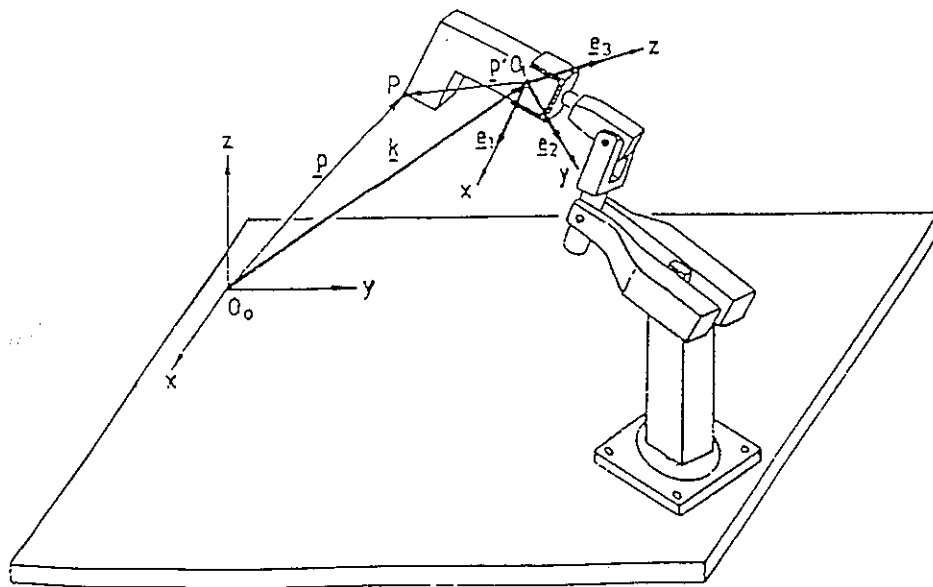
สรุปได้ว่า ผลการคูณกันทางเวกเตอร์ (Cross product) ระหว่างเวกเตอร์ 1 หน่วย จะได้เวกเตอร์ที่ 3 และผลของการคูณกันทางสเกลลา (Inner product) ซึ่งเราจะได้ภาพฉายเวกเตอร์บนอีกเวกเตอร์หนึ่ง จะมี nonzero vector ที่ตั้งฉากซึ่งกันและกัน

$$e_1 \cdot e_2 = e_2 \times e_3 = e_2 \cdot e_3 = 0$$

ดังนั้น ในจุดพิกัดใด ๆ ในเวกเตอร์ 1 หน่วย (e_1, e_2, e_3) ที่ฉายบนพิกัดของระบบพื้นฐานจะมีคุณสมบัติข้างต้นก็ต่อเมื่อเป็นระบบพิกัดมุมฉาก

Rotation matrix (R) คือ เมทริกซ์มุมฉาก 3x3 มีตัวประกอบ x, y, z ของเวกเตอร์ 1 หน่วย (e_1, e_2, e_3) ที่ฉายบนพิกัดของระบบพิกัด

$$R = \begin{bmatrix} e_{1x} & e_{2x} & e_{3x} \\ e_{1y} & e_{2y} & e_{3y} \\ e_{1z} & e_{2z} & e_{3z} \end{bmatrix}$$



รูปที่ 2.2 แสดงพิกัดสัมพันธ์และพิกัดสัมพันธ์, k คือเวกเตอร์ตำแหน่งจากระบบพิกัดฐาน O_0 ถึงจุดกำเนิด O_1 , p คือเวกเตอร์ตำแหน่งจากจุดกำเนิด O_1 ถึงจุด P , p คือเวกเตอร์จากตำแหน่งระบบพิกัดฐานถึงจุด P

จากรูปที่ 2.2 พิจารณาพิกัดของจุด P บนโครงสร้างพื้นฐาน (Base frame) เมื่อตำแหน่งของมันมีส่วนเกี่ยวข้องกับระบบพิกัด ตำแหน่งพิกัดของจุด P บนระบบพิกัดฐาน จะถูกคำนวณโดยการคูณเวกเตอร์ (Pre-multiplying the vector) p ควบคู่กับเวกเตอร์หมุน R บวกด้วยเวกเตอร์ตำแหน่ง k

$$p = k + R * p'$$

แต่มีเมตริกซ์อีกอันที่แสดงผลลัพธ์ออกมาเหมือนกัน เมื่อเวกเตอร์ตำแหน่ง k ถูกรวมกันในตำแหน่งที่ 1 และ 4 กลายเป็นสมาชิก(Element) ที่ 4 ของเวกเตอร์ p' ดังสมการ

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} e_{1x} & e_{2x} & e_{3x} & k_x \\ e_{1y} & e_{2y} & e_{3y} & k_y \\ e_{1z} & e_{2z} & e_{3z} & k_z \end{bmatrix} * \begin{bmatrix} p'_x \\ p'_y \\ p'_z \end{bmatrix}$$

ที่แถวที่ 4 ของเมตริกซ์ 3x4 เมื่อค่า 0 3 ตัว และค่า 1 อีกหนึ่งค่าทำให้เป็นเมตริกซ์ 4x4 เราสามารถสร้างเมตริกซ์การแปลงร่วม (Homogeneous transformer matrix) ได้สมการดังนี้

$$\begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = \begin{bmatrix} e_{1x} & e_{2x} & e_{3x} & k_x \\ e_{1y} & e_{2y} & e_{3y} & k_y \\ e_{1z} & e_{2z} & e_{3z} & k_z \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} p'_x \\ p'_y \\ p'_z \\ 1 \end{bmatrix}$$

เมตริกซ์การแปลงร่วม (Homogeneous transformer matrix) ได้รวมเอาเวกเตอร์ตำแหน่ง k กับเวกเตอร์การหมุน R เพื่อที่จะได้ ตำแหน่งและมุมที่สมบูรณ์ของระบบพิกัดอีกระบบหนึ่งที่มีส่วนเกี่ยวข้องกับ โครงสร้างพื้นฐาน(Base frame) เมื่อนำการแปลงพิกัดของจุด P จากระบบพิกัดหนึ่งไปยังระบบพิกัดหนึ่งแสดงในรูปที่ 2.2 และมีสมการเมตริกซ์การแปลงร่วม(Homogeneous transformer matrix) ให้เป็น T ที่ประกอบด้วยการหมุน R และการเปลี่ยนแปลง(Transformer) ดังนี้

$$T = \left[\begin{array}{ccc|c} R & & & k \\ - & - & - & - \\ 0 & 0 & 0 & 1 \end{array} \right]$$

R คือ เมตริกซ์ย่อย 3x3 ในส่วนของการหมุน (Rotation part)

k คือ 3x1 เมตริกซ์ ในส่วนของการแปลง (Transformer matrix)

ส่วนของการแปลง คือ คอลัมน์เวกเตอร์ที่บอกตำแหน่งของ O_0 ไปยัง O_1 นอกจากนี้เรายังแปลงเมทริกซ์ 4×4 ที่ใช้ในการคำนวณแขนออกเป็น 2 แบบ

แบบที่ 1 บอกระบบพิกัดโดยสัมพันธ์กับโครงสร้างพื้นฐาน ตามที่ได้เสนอมานี้แล้ว

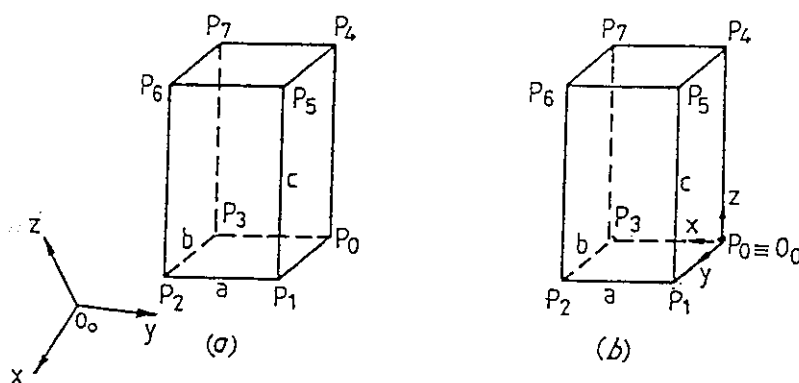
แบบที่ 2 บอกความสัมพันธ์ระหว่าง 2 ระบบพิกัด ใดๆ ที่เชื่อมต่อกันอยู่

2.2 การบอกลักษณะของวัตถุในทางคณิตศาสตร์

(Mathematics description of objects)

จุดมุ่งหมายในหัวข้อนี้ คือ การแสดงการใช้การแปลงร่วม (Homogeneous transformer) ในการบอกลักษณะของวัตถุ และตำแหน่งของมันในรูปแบบที่เข้าใจได้ เพื่อคำนวณวัตถุล้อมรอบด้วยพื้นผิวระนาบ (Planar surface) ซึ่งบอกได้ด้วยเมทริกซ์ $4 \times N$ โดย N คือ จำนวนจุดปลายของวัตถุที่แต่ละจุดจะแสดงโดยเวกเตอร์ตำแหน่ง (Position vector) ซึ่งมีรูปแบบดังนี้

$$[x \ y \ z \ 1]^T \quad ; \quad T \text{ คือ การสลับตำแหน่ง (Transpose)}$$



รูปที่ 2.3 แสดงจุดปลายลูกบาศก์ $P_0 - P_7$

(a) ระบบพิกัดที่ไม่ขึ้นกับลักษณะของลูกบาศก์

(b) ระบบพิกัดที่กำหนดตามลักษณะของลูกบาศก์, โดยให้จุดกำเนิดคือ P_0

(ที่มา : Introductoin to Robotics. John J. Craig)

การบอกลักษณะของวัตถุ โดยพื้นผิวระนาบ (Planar surface) มี 2 รูปแบบ คือ

1. จุดกำเนิดของระบบพิกัดวัตถุ สามารถมีตำแหน่งอิสระทุกๆ ลักษณะและมุมโครงพิกัดของมันเป็นมุมปิด ดังรูปที่ 2.3 (a) รูปแบบทั่วไปของเมตริกซ์ที่ใช้แสดงวัตถุ N จุดปลาย ดังนี้

$$[\text{object}] = \begin{bmatrix} X_0 & X_1 & - & - & - & X_{N-1} \\ Y_0 & Y_1 & - & - & - & Y_{N-1} \\ Z_0 & Z_1 & - & - & - & Z_{N-1} \\ 1 & 1 & - & - & - & 1 \end{bmatrix}$$

2. จุดกำเนิดของระบบพิกัดวัตถุ กำหนดให้อยู่กับที่ ดังรูปที่ 2.3 (b) ยกตัวอย่างของจุดปลาย เช่น จุดศูนย์กลางในไม้ถ่วง สามารถเลือกจุดกำเนิดได้

ให้วัตถุเป็นลูกบาศก์ที่มี 8 จุดปลาย $P_0 - P_7$ ในพิกัดคาเตเซียน ดังรูปที่ 2.3 (b) จุดกำเนิดอยู่กับที่ ณ จุด P_0 แกนของระบบพิกัดวางแนวกับของจุดตัดที่ผ่านจุดกำเนิด มีปริมาตร $a \times b \times c$ มีทิศทาง x, y, z ตามลำดับคอลัมน์เมตริกซ์ของลักษณะวัตถุสำหรับจุด P_0 คือ

$$P_0 = [0 \ 0 \ 0 \ 1]^T$$

$$P_1 = [0 \ b \ 0 \ 1]^T$$

$$P_2 = [a \ b \ 0 \ 1]^T$$

และเรานำมาทำเป็นเมตริกซ์ของลูกบาศก์จะได้

$$[\text{cuboid}] = \begin{bmatrix} 0 & 0 & a & a & 0 & 0 & a & a \\ 0 & b & b & 0 & 0 & b & b & 0 \\ 0 & 0 & 0 & 0 & c & c & c & c \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ | & | & | & | & | & | & | & | \\ P_0 & P_1 & P_2 & P_3 & P_4 & P_5 & P_6 & P_7 \end{bmatrix}$$

พิจารณาการหมุนและการแปลง (Transformer and rotation) บนวัตถุ รูปแบบจะอธิบายโดยเมตริกซ์ 4×4 เมตริกซ์การแปลง T (Transformer matrix, T) ในหัวข้อที่ผ่านมา ความสัมพันธ์ของตำแหน่งใหม่และตำแหน่งเริ่มแรกของวัตถุ คือ

$$[\text{Object}]_n = T * [\text{Object}]_s$$

โดยที่ n = ตำแหน่งใหม่ (New position) และ s = ตำแหน่งแรกเริ่ม (Starting position)

ในกรณีนี้ ตำแหน่งใหม่ หมายถึง โครงงพิกัดพื้นฐานและ ตำแหน่งแรกเริ่ม คือ ระบบพิกัดที่แสดงในการแตกสมการดังนี้

$$\begin{bmatrix} X_{0,n} & X_{1,n} & - & - & - & X_{N-1,n} \\ Y_{0,n} & Y_{1,n} & - & - & - & Y_{N-1,n} \\ Z_{0,n} & Z_{1,n} & - & - & - & Z_{N-1,n} \\ 1 & 1 & - & - & - & 1 \end{bmatrix} = \begin{bmatrix} R & & & | & k \\ - & - & - & | & - \\ 0 & 0 & 0 & | & 1 \end{bmatrix} * \begin{bmatrix} X_{0,s} & X_{1,s} & - & - & - & X_{N-1,s} \\ Y_{0,s} & Y_{1,s} & - & - & - & Y_{N-1,s} \\ Z_{0,s} & Z_{1,s} & - & - & - & Z_{N-1,s} \\ 1 & 1 & - & - & - & 1 \end{bmatrix}$$

เมตริกซ์ทางด้านซ้ายมือของสมการ แสดงจุดปลายของลูกบาศก์ตำแหน่งใหม่หลังจากการแปลงเมตริกซ์ทางด้านขวามือของสมการแสดงเมตริกซ์การแปลง (Transformer matrix) และจุดปลายในโครงพิกัดเริ่มแรกตามลำดับ

2.3 ความสัมพันธ์ของตัวแปลงในเนื้อที่ว่างของแขนกล

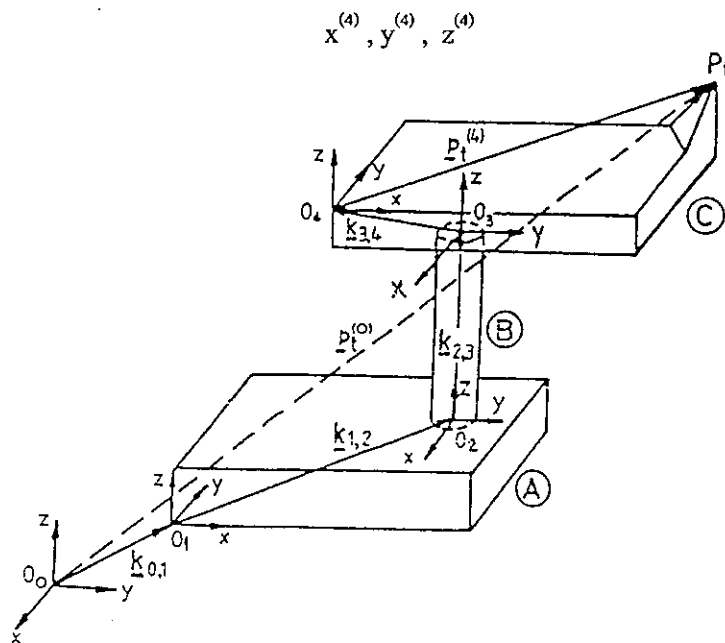
(Relative transformer in the robot workspace)

ทางคณิตศาสตร์ชี้ให้เห็นถึงการแปลงของส่วนต่างๆ ของแขนกล พิจารณาเป็นลำดับของการแปลงระบบพิกัดร่วม (Homogeneous coordinate transformations) เป้าหมายของการเคลื่อนที่มักจะถูกกำหนดให้อยู่ในเทอมของที่ตั้งในเนื้อที่ว่าง สิ่งสำคัญในส่วนนี้ คือ การวางแผนการเคลื่อนที่ (Manipulated) ที่สัมพันธ์กับส่วนต่างๆ

จะเป็นประโยชน์อย่างมากในการใช้การแปลงสัมพัทธ์นั้น คือ ถ้าใช้ในการทดสอบความสัมพันธ์ของการแปลงพิกัด จะต้องทราบตำแหน่ง, มุมและทิศทางของส่วนต่างๆ ของโครงสร้าง ถ้าใช้ในการเขียนโปรแกรมจะต้องจำเป็นที่จะต้องรู้ทุกส่วนประกอบ

ตำแหน่งของจุดเป้าหมาย P_i ในระบบพิกัดฐานของจุดกำเนิด O_0 จะถูกกำหนดโดยส่วนประกอบทางเรขาคณิต และตำแหน่งโดยสมมูลของจุดกำเนิด O_1 ที่มีความสัมพันธ์ทั้งตำแหน่งและรูปสมมติว่า เราต้องการที่จะเคลื่อนที่ส่วนประกอบของเครื่องมือไปยังเป้าหมาย P_i ในส่วนของ C โดยที่เราทราบทิศทางทั้งหมดของ A , B และ C และที่ตั้งรวมทั้งมุมในเนื้อที่ว่าง เราสามารถหาตำแหน่งของจุด P_i เราสามารถหาพิกัดที่ได้โดยใช้การแปลงร่วม (Homogeneous transformation) เพื่อที่จะแสดงวิธีการนี้เราจะเข้าถึงจุด P_i ที่ละชั้นจากจุดกำเนิด O_0 ก่อนที่จะทำ เราต้องกำหนดระบบพิกัดย่อยให้แก่แต่ละส่วนของชิ้นส่วนจากข้อมูลที่เข้ามา เราใช้พิกัด P_i ในระบบพิกัดย่อย O_i ซึ่งจะ

กำหนดให้เป็น



รูปที่ 2.4 แสดงส่วนประกอบ A,B และ C ให้จุด P_t คือเป้าหมายที่จะให้เคลื่อนที่ไป
(ที่มา : Introductoin to Robotics. John J. Craig)

จุดมุ่งหมาย คือ อธิบายจุดของเป้าหมาย P_t ในระบบพิกัดฐานนั้น คือ เราต้องการกลุ่มของ
เวกเตอร์ $P_t^{(0)}$ พิกัดนั้นคือ

$$x_t^{(0)}, y_t^{(0)}, z_t^{(0)}$$

เมตริกซ์การแปลง (Transformation matrix) ${}^i_n T$ อธิบายตำแหน่งและมุมของระบบพิกัดที่
 n สัมพันธ์กับ i ซึ่งมีสูตรดังนี้

$${}^1_n T = \prod_i^{n-1} {}^i_{i+1} T = {}^1_2 T * {}^2_3 T * \dots * {}^{n-1}_n T$$

โดยที่ $i = 0, 1, 2, \dots, n-1$ เริ่มที่เลขใดๆ สำหรับ $i < n$ และ

$${}^i_n T = \begin{bmatrix} R_{i,i+1} & | & k_{i,i+1} \\ - & - & - \\ 0 & 0 & 0 & | & 1 \end{bmatrix}$$

ผลของการแปลงเมตริกซ์ ${}^i T_n$ อธิบายระบบพิกัดที่ n สัมพันธ์กับค่า i ใดๆ ที่เป็นส่วนประกอบของโครงพิกัด ที่ดูกันเป็นลำดับสำหรับ $i < n$ ดังนั้นพิกัดของเป้าหมาย P_i ในระบบพิกัดนั้นกับค่าส่วนประกอบโครงพิกัดใดๆ อธิบายได้เป็น

$$P_t^{(i)} = {}^i T_n * P_t^n$$

หรือ

$$\begin{bmatrix} x_t^{(i)} \\ y_t^{(i)} \\ z_t^{(i)} \\ 1 \end{bmatrix} = {}^i T_n * \begin{bmatrix} x_t^{(n)} \\ y_t^{(n)} \\ z_t^{(n)} \\ 1 \end{bmatrix}$$

จากตัวอย่างในรูปที่ 2.5 เวกเตอร์ $P_t^{(0)}$ และพิกัดถึงจุดเป้าหมาย P_t ให้ $i=0$ และ $n=4$ ในระบบพิกัดฐานได้

$$P_t^{(0)} = {}^0 T_4 * P_t^{(4)}$$

หรือ

$$\begin{bmatrix} x_t^{(0)} \\ y_t^{(0)} \\ z_t^{(0)} \\ 1 \end{bmatrix} = {}^0 T_1 * {}^1 T_2 * {}^2 T_3 * {}^3 T_4 * \begin{bmatrix} x_t^{(4)} \\ y_t^{(4)} \\ z_t^{(4)} \\ 1 \end{bmatrix}$$

2.4 การแปลงในแขนกลหลายแกน

(Transformation along the kinematics chain)

แขนกลหลายแกนประกอบด้วยตัวเชื่อม(Link) ที่เชื่อมต่อกันด้วยจุดต่อ (Joint) แขนกลอาจจะมีส่วนปลาย 2 ข้างเป็นรูปปริซึม (Prismatic joint) แขนกลหลายแกนมีลำดับอิสระ n จำนวน (n Degree of freedom) พิจารณาโดยที่มี m จุดต่อที่ต่อเชื่อมกัน n ตัวเชื่อม เมื่อนำมาประกอบกันเข้าก็จะกลายเป็นโซ่กล ในส่วนของรูปแบบของตัวเชื่อมจะมีความสัมพันธ์กันระหว่างจุดต่อและจุดปลายแต่ละจุดปลายของตัวเชื่อม ในการที่จะอธิบายความสัมพันธ์ระหว่างส่วนต่างๆ ของแขนกล เราจะกำหนดระบบพิกัดขึ้นมาสำหรับแต่ละจุดต่อทุกๆ ตัวเชื่อม มีลักษณะของมันได้ 2 ลักษณะ คือ

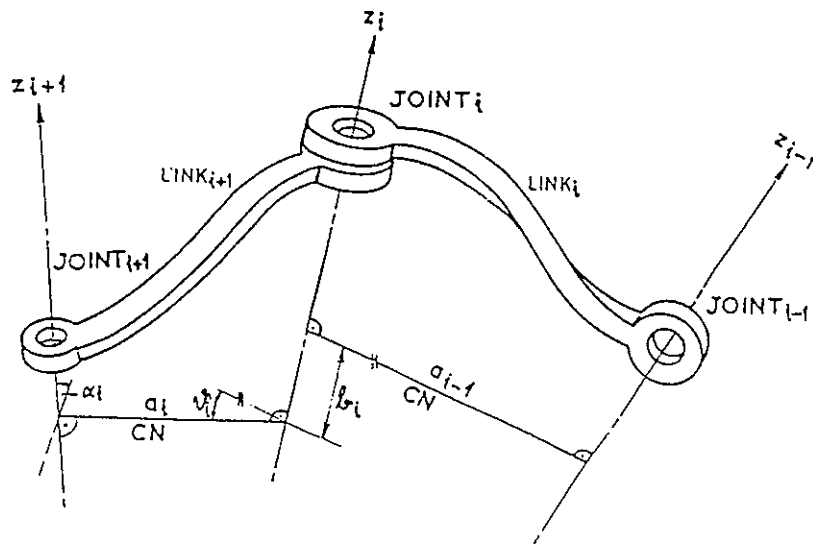
1. ระยะ a_i คือ ระยะห่างของแกน z ที่ปลายทั้ง 2 ของตัวเชื่อมตามแนวแกนตั้งฉากร่วมกัน
2. มุม α_i คือ มุมที่เหลือจากการตั้งฉากกับเส้น a_i กับแนวแกน z ที่ดูเข้า

โดยที่ a_i มักเรียกว่า ความยาวของตัวเชื่อม (Link length)

α_i มักเรียกว่า มุมบิดของตัวเชื่อม (Link twist angle)

ในกรณีที่ไม่มีการดูเข้าของแนวแกน z , $\alpha_i = 0$ ทำให้เกิดการตั้งฉากร่วมกัน (Common normal) มากมายหลายเส้น

การจัดวางเป็นลำดับของตัวเชื่อม 2 ตัวเชื่อม จะต่อกันด้วยจุดต่อดังรูปที่ 2.5 ทุกๆ แกนของจุดต่อจะมี 2 การตั้งฉากร่วมกัน (Common normal) ก็จะเป็นเส้นของแต่ละตัวเชื่อมไปความสัมพัทธ์ทางตำแหน่งของตัวเชื่อม 2 ตัวนี้ จะให้เป็น b_i คือ ระยะทางจากเส้นการตั้งฉากร่วมกัน (Common normal) ทั้งสองนั่นเอง และให้ θ_i คือ มุมระหว่างเส้นการตั้งฉากร่วมกัน (Common normal) ทั้งสอง ดังนั้น b_i และ θ_i คือ ตัวแปรของจุดต่อแต่ละจุดนั่นเอง



รูปที่ 2.5 แสดงจุดต่อของแขนกลและตัวแปรต่างๆ b_i = ระยะระหว่างจุดต่อ, θ_i = มุมจุดต่อ

(ที่มา :. Introductoin to Robotics. John J. Craig)

ในโครงสร้างของแขนกลทำให้เราแยกได้ว่าระหว่าง Revolute หรือ Prismatic

โดยที่ Revolute คือ การหมุนของตัวเชื่อมระบบแกน

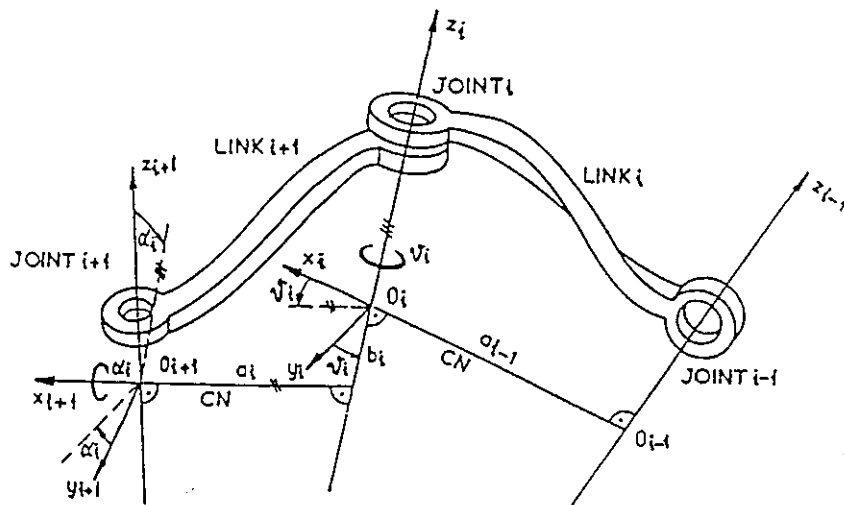
θ_i, α_i คือ ตัวแปรของจุดต่อ

Prismatic คือ การเคลื่อนที่ตามแนวแกน

ดังนั้น a, b คือ ตัวแปร

สำหรับการกำหนดระบบพิกัดของชิ้นส่วนแต่ละชิ้นส่วนนั้น เราจะใช้วิธีทางเรขาคณิตของ เดนาวิต-เฮอ์เทนเบิร์ก (Denavit – Hertenberg) วิธีนี้ระบบพิกัดจะถูกกำหนดให้เป็นลำดับของการหมุน 2 การหมุน และการเปลี่ยนตำแหน่ง 2 ตำแหน่ง เพื่อที่จะให้ระบบพิกัด i เข้ามาอยู่ตรงกันกับระบบพิกัด $i+1$ ซึ่งดังแสดงในรูปที่ 2.6 สำหรับจุดต่อ $i+1$ และจุดต่อ i

- พิจารณารูปที่ 2.6
- มุม θ_i หมุนรอบแกน z_i , $T(\theta_i)$
 - มุม b_i เคลื่อนที่ไปตามแกน z_i , $T(b_i)$
 - มุม a_i เคลื่อนที่ไปตามแกน x_i , $T(a_i)$
 - มุม α_i หมุนรอบแกน x_i , $T(\alpha_i)$



รูปที่ 2.6 แสดงระบบพิกัดจุดต่อของแขนกล แต่ละจุดต่อจะมีกลุ่มของพิกัด ตัวแปรจะกำหนดตามวิธีของเดนาวิต-เฮอ์เทนเบิร์ก (Denavit – Hertenberg)

(ที่มา : Introductoin to Robotics. John J. Craig)

ให้ p_i และ p_{i+1} แสดงจุด p บน i และ $i+1$ ในระบบพิกัดจะได้

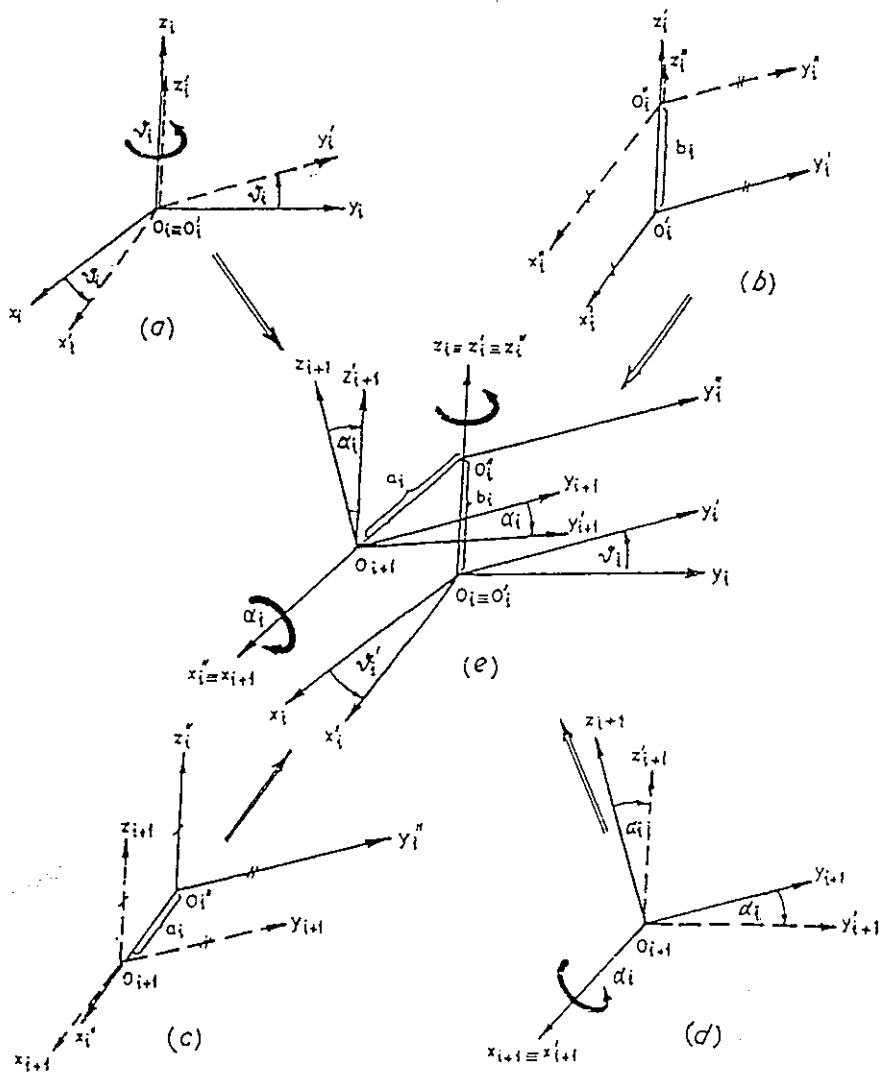
$$p_i = \begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix}_i$$

$$p_{i+1} = \begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix}_{i+1}$$

ความสัมพันธ์ระหว่างจุดต่อทั้ง 2 นั้นคือ

$$P_i = {}^0_4T * P_{i+1}$$

เมื่อ 1_4T คือ เมตริกซ์การแปลงระบบพิกัด 4x4 (Coordinate transformation matrix 4x4) จะพิจารณาทีละขั้นไปโดยใช้รูปที่ 2.7



รูปที่ 2.7 แสดงการแปลงระบบพิกัดร่วม (Homogeneous coordinate transformation)

- (a) การหมุนของ θ_i รอบแกน z_i
- (b) คือระยะตามแนวแกน z_i
- (c) คือระยะตามแนวแกน x_{i+1}
- (d) การหมุน α_i รอบแกน x_{i+1}
- (e) ผลของการแปลงระหว่าง joint i และ joint $i+1$

(ที่มา : Introductoin to Robotics. John J. Craig)

1. เมตริกซ์การแปลง (The transformation matrix) ของการหมุน θ_i รอบแกน z_i จะได้

$$T(\theta_i) = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2. เมตริกซ์การแปลง (The transformation matrix) สำหรับการเคลื่อนที่ b_i ไปตามแกน z_i

$$T(b_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & b_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3. เมตริกซ์การแปลง (The transformation matrix) สำหรับการเคลื่อนที่ a_i ไปตามแกน x_{i+1}

$$T(a_i) = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

4. เมตริกซ์การแปลง (The transformation matrix) สำหรับการหมุน α_i รอบแกน x_{i+1}

$$T(\alpha_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

ผลของ เมตริกซ์การแปลง (The transformation matrix) ระหว่างจุดต่อ 2 จุด ตามรูปที่ 2.7 (e) คือ

$${}^1_{i+1}T = T(\theta_i) * T(b_i) * T(a_i) * T(\alpha_i)$$

ทำการคูณ จะได้

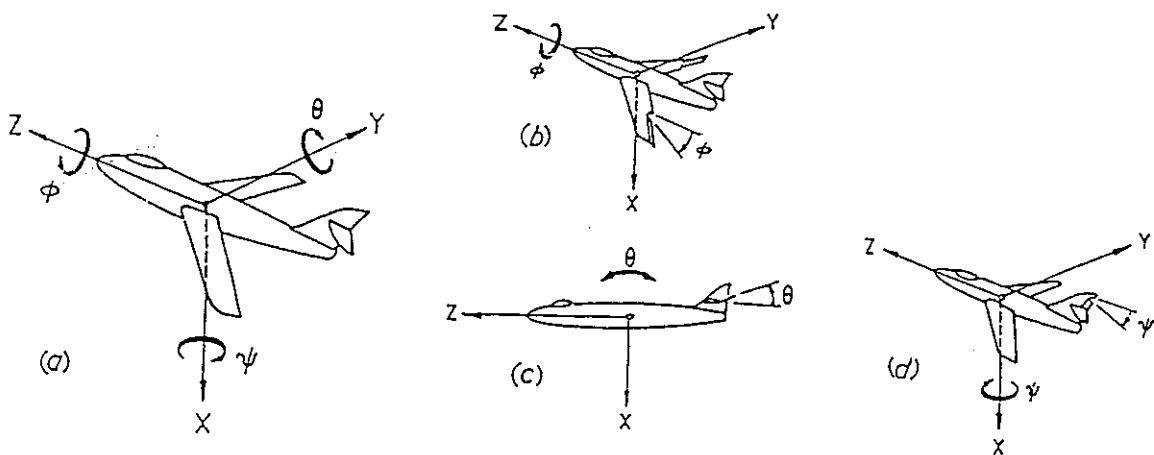
$${}^1_{i+1}T = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & \cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & b_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

สมการเมตริกซ์ ที่ได้ คือ เมตริกซ์การแปลงรวมทั้งหมด (Overall transformation matrix) ในรูปแบบทั่วไป ระหว่างจุดต่อ 2 จุดใดๆ ที่อยู่ข้างเคียงกัน i และ $i+1$ การหมุนเกิดขึ้นเพียงแกนเดียวจะเกิด θ_i หมุนรอบ z_i สมมูลกับ z_i และ α_i รอบแกน x_i สมมูลกับ x_{i+1} สำหรับคอลัมน์ที่ 4 ของสมการ คือตำแหน่งของเวกเตอร์ k ระหว่างจุดกำเนิด O_i และ O_{i+1}

2.5 การกำหนดมุม โดยอาศัยมุมของออยเลอร์

(Euler angles for specifying orientation)

การหมุนในระบบพิกัด สามารถอธิบายได้โดยเมตริกซ์ R คือ เมตริกซ์ย่อย 3×3 (Submatrix 3×3) ที่อยู่ด้านซ้ายมือบนของเมตริกซ์การแปลงระบบพิกัด 4×4 (Coordinate transformation matrix 4×4) ในการประยุกต์ใช้กับแขนกลจะกำหนดมุมของวัตถุให้อยู่ในรูปของมุมออยเลอร์ (Euler angles) มุมสัมพัทธ์ของระบบพิกัดมีส่วนเกี่ยวข้องกับมุมอื่นๆ ซึ่งสามารถอธิบายได้เป็นลำดับของการหมุนรอบแกน x, y, z ของระบบพิกัดอ้างอิง ในการหมุนนี้จะอยู่ในรูปของตัวแปรของมุม 3 ตัวแปร นั่นก็คือ มุมออยเลอร์ (Euler angles) นั่นเอง การเลือกใช้ขึ้นอยู่กับวิธีการประยุกต์ใช้ตามปัญหาที่เกิดขึ้น ในปัญหาของเรา มุมออยเลอร์ (Euler angles) อธิบายอย่างแน่นอนในเทอมของการหมุนรอบแกน 3 แกน ภายใต้การแปลงนี้ จะได้จำนวนรูปแบบของกลุ่มการหมุนที่มีตัวแปรค่อนข้างมาก แต่ลำดับการหมุนที่นิยมในการใช้กับแขนกลในอุตสาหกรรมมีอยู่ 3 แบบ คือ



รูปที่ 2.8 แสดงการควบคุมเครื่องบิน

- (a) แสดงแกนหลัก x, y, z ของเครื่องบิน (b) ผลของมุม Roll, ϕ ทำให้เกิด Aileron
 (c) ผลของมุม Pitch, θ ทำให้เกิด Elevator (d) ผลของมุม Yaw, ψ ทำให้เกิด Rudder

1. Roll-yaw-roll
2. Roll-pitch-yaw
3. Roll-pitch-roll

มีการหมุนอยู่ 3 ลักษณะ เราจะอธิบายการหมุนโดยสมมุติกับเครื่องบิน หลังจากนั้นก็จะนำมาอธิบายโดยให้อยู่ในรูปของคณิตศาสตร์ การเคลื่อนที่ของเครื่องบินสามารถแสดงโดยการเคลื่อนที่รอบจุดศูนย์กลางโน้มถ่วง และการหมุนรอบจุดใดๆ กลุ่มแกนคาเตเซียนและแกนหลัก แสดงดังรูปที่ 2.8(a) แกนหลัก (Principle axes) x, y, z และความเร็วเชิงมุมรอบแกน Roll คือ ϕ ความเร็วเชิงมุมรอบแกน Pitch คือ θ และความเร็วเชิงมุมรอบแกน Yaw คือ ψ

จากรูปที่ 2.8 จะเห็นได้ว่ามีระนาบ 3 ระนาบในการควบคุมการวางตัวของเครื่องบิน คือ Elevator, Rudders และ Ailerons ตามรูปที่ 2.8 (b), (c), (d) ซึ่งลักษณะการเคลื่อนที่นี้คล้ายกับการเคลื่อนที่ของแขนกล ทำให้เข้าใจการเคลื่อนที่ของแขนกลได้ง่ายขึ้น

1. Roll-yaw-roll geometry

รูปแบบของ Roll-yaw-roll

Roll คือ การหมุนรอบแกน z

Yaw คือ การหมุนรอบแกน x'

Roll คือ การหมุนรอบแกน z''

พิจารณารูปที่ 2.8 เมตริกซ์ของการหมุนรวม (Overall rotation), $R_{RYR}(\alpha, \beta, \gamma)$ คือ ผลคูณของเมตริกซ์ของการหมุนของแต่ละมุม

$$R_{RYR}(\alpha, \beta, \gamma) = R_z(\alpha) \cdot R_{x'}(\beta) \cdot R_{z''}(\gamma)$$

โดยที่ α = การหมุนรอบแกน z -roll

β = การหมุนรอบแกน x' -yaw

γ = การหมุนรอบแกน z'' -roll

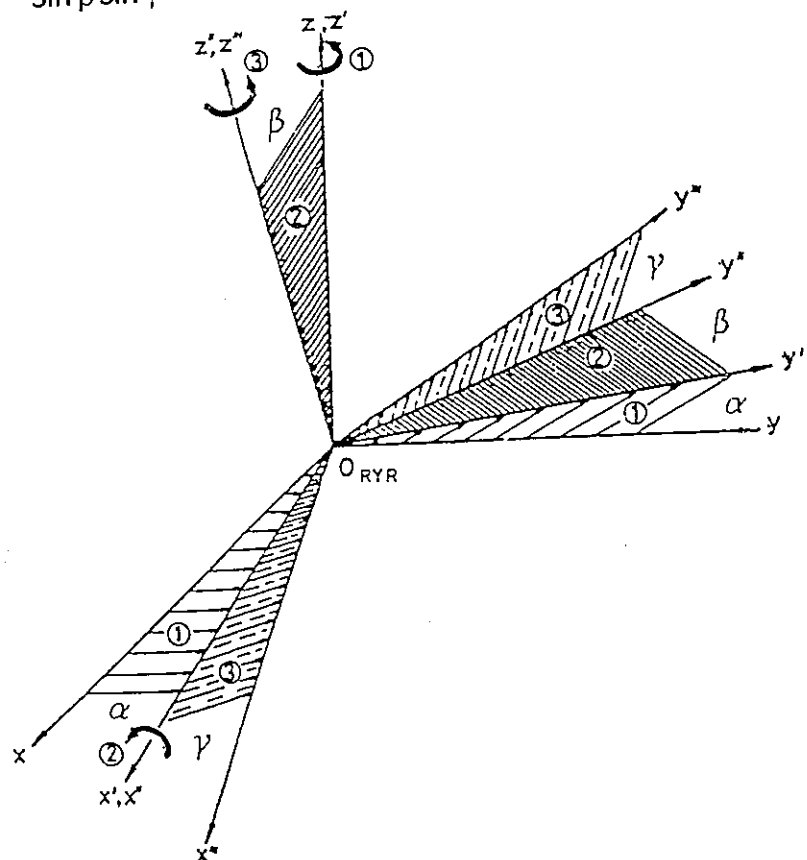
$$R_z(\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}_{\text{roll}}$$

$$R_x(\beta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \beta & -\sin \beta \\ 0 & \sin \beta & \cos \beta \end{bmatrix}_{\text{yaw}}$$

$$R_z(\gamma) = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}_{\text{roll}}$$

เพราะเป็นผลของการคูณจะได้เมทริกซ์ R จะได้

$$R_{\text{RYR}}(\alpha, \beta, \gamma) = \begin{bmatrix} \cos \alpha \cos \gamma - \sin \alpha \cos \beta \sin \gamma & -\cos \alpha \sin \gamma - \sin \alpha \cos \beta \cos \gamma & \sin \alpha \sin \beta \\ \sin \alpha \cos \gamma + \cos \alpha \cos \beta \sin \gamma & -\sin \alpha \sin \gamma + \cos \alpha \cos \beta \cos \gamma & -\cos \alpha \sin \beta \\ \sin \beta \sin \gamma & \sin \beta \cos \gamma & \cos \beta \end{bmatrix}$$



รูปที่ 2.9 แสดงมุมออยเลอร์(Euler angles(α, β, γ)) สำหรับ Roll-yaw-rol
(ที่มา: การควบคุมหุ่นยนต์อุตสาหกรรม ด้วยไมโครคอมพิวเตอร์, กลยุทธ ชัยล้อม, ชูพงศ์ ช่วยเพ็ญ และจักรกริช ชัยเทวรักษ์)

ในการที่จะใช้ Roll-yaw-roll geometry เมื่อ

1. $\beta = 0^\circ$ แกน z และ z' วางตัวในแนวเดียวกันและขนานกันทั้งมุม α, γ ยังวางตัวเหมือนเดิม ดังนั้นสมการข้างต้นจะได้

$$R_{RYR}(\alpha, \beta, \gamma) = \begin{bmatrix} \cos(\alpha + \gamma) & -\sin(\alpha + \gamma) & 0 \\ \sin(\alpha + \gamma) & \cos(\alpha + \gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}_{\beta=0}$$

2. $\beta = 180^\circ$ แกน z และ z'' วางตัวอยู่ในแนวเดียวกัน แต่อยู่ตรงกันข้ามกัน ดังนั้นสมการ จะเป็น

$$R_{RYR}(\alpha, \beta, \gamma) = \begin{bmatrix} \cos(\alpha - \gamma) & \sin(\alpha - \gamma) & 0 \\ \sin(\alpha - \gamma) & -\cos(\alpha - \gamma) & 0 \\ 0 & 0 & -1 \end{bmatrix}_{\beta=180}$$

เราสามารถที่จะจำกัดช่วง $(0, 180)$ เพราะว่าถ้าใช้ช่วงที่เกินพวกนี้แล้วจะได้เมตริกซ์รูปแบบเดิม คือ ถ้าอยู่ในช่วง $(\alpha+180, -\beta, \gamma+180)$ จะได้เมตริกซ์ที่เหมือนกับ $R_{RYR}(\alpha, \beta, \gamma)$ สำหรับ $-\beta$

2. Roll-pitch-yaw geometry

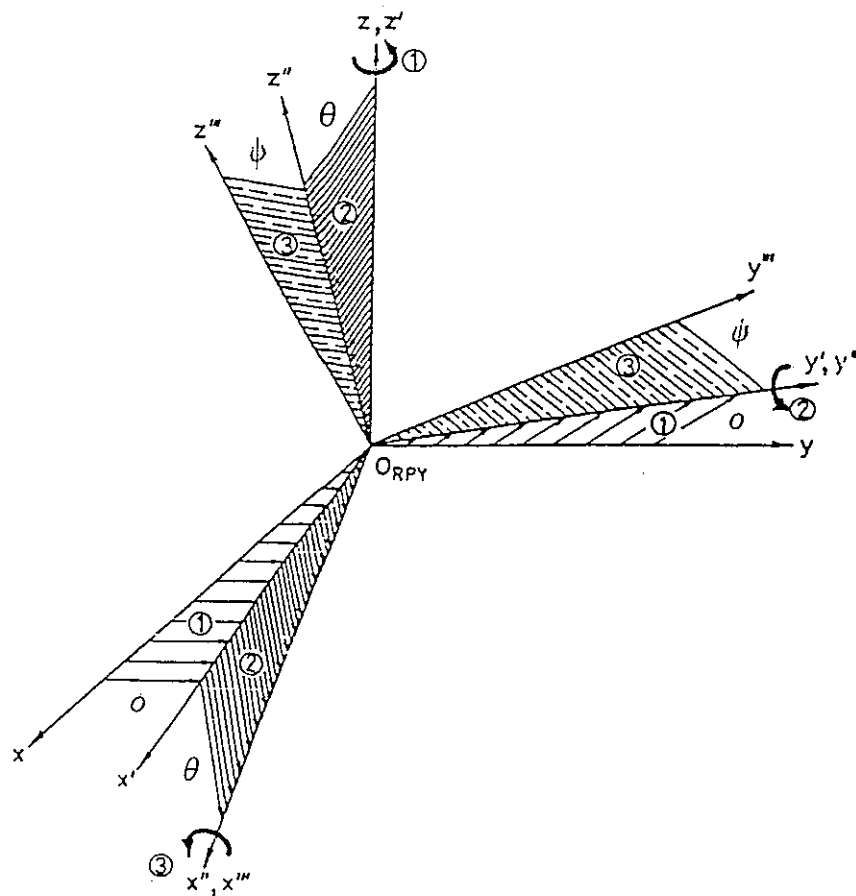
Roll-pitch-yaw geometry เป็นรูปแบบที่ใช้อย่างกว้างขวางมากในแขนกล เพราะว่า การหมุนทั้ง 3 แกน ในตำแหน่งอ้างอิงเป็นเรื่องที่ซับซ้อน Roll คือ การหมุนรอบแกน z , Pitch คือ การหมุนรอบแกน y' และ Yaw คือการหมุนรอบแกน x'' ดังรูปที่ 2.1 เมื่อ $R_{RYR}(\phi, \theta, \varphi)$ คือเมตริกซ์การหมุนรวม(Overall rotation matrix) คือ ผลคูณของเมตริกซ์การหมุนของแต่ละมุมตามแกนที่กำหนด โดยที่

$$R_{RYR}(\phi, \theta, \varphi) = R_z(\phi) \cdot R_{y'}(\theta) \cdot R_{x''}(\varphi)$$

โดยที่ ϕ = การหมุนรอบแกน z -roll

θ = การหมุนรอบแกน y' -pitch

φ = การหมุนรอบแกน x'' -yaw



รูปที่ 2.10 แสดงมุมออยเลอร์ (Euler angles (ϕ, θ, φ)) สำหรับ Roll – pitch – yaw (ที่มา: การควบคุมหุ่นยนต์อุตสาหกรรม ด้วยไมโครคอมพิวเตอร์, กลยุทธ์ ชัยด้อม, ชูพงศ์ ช่วยเพ็ญ และจักรกริช ชัยเทวรักษ์)

เมตริกซ์การหมุนของแต่ละมุมรอบแกนแสดงดังนี้

$$R_z(\phi) = \begin{bmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix}_{\text{roll}}$$

$$R_{y'}(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}_{\text{pitch}}$$

$$R_{x''}(\varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{bmatrix}_{\text{yaw}}$$

ผลของการคูณของเมตริกซ์คือ

$$R_{RPY}(\phi, \theta, \varphi) = \begin{bmatrix} \cos \phi \cos \theta & \cos \phi \sin \theta \sin \varphi - \sin \phi \cos \varphi & \cos \phi \sin \theta \cos \varphi + \sin \phi \sin \varphi \\ \sin \phi \cos \theta & \sin \phi \sin \theta \sin \varphi + \cos \phi \cos \varphi & \sin \phi \sin \theta \cos \varphi - \cos \phi \sin \varphi \\ -\sin \theta & \cos \theta \sin \varphi & \cos \theta \cos \varphi \end{bmatrix}$$

ลักษณะพิเศษของ Roll - pitch - yaw geometry ทำให้เมตริกซ์การหมุน R ใช้ไม่ได้
เมื่อ $\theta = +90^\circ$ หรือ $\theta = -90^\circ$ คือ

$$1. \theta = +90^\circ$$

แกน x" จะวางตัวในแนวเดียวกับแกน z แต่ตรงกันข้ามกับมุม ϕ, φ จะเป็นมุมเดียวกัน

$$R_{RPY}(\phi, \theta, \varphi) = \begin{bmatrix} 0 & \sin(\varphi - \phi) & \cos(\varphi - \phi) \\ 0 & \cos(\varphi - \phi) & -\sin(\varphi - \phi) \\ -1 & 0 & 0 \end{bmatrix}_{\theta=90^\circ}$$

2. $\theta = -90^\circ$ แกน x" และแกน z วางตัวแนวเดียวกันและขนานกัน

$$R_{RPY}(\phi, \theta, \varphi) = \begin{bmatrix} 0 & -\sin(\varphi + \phi) & -\cos(\varphi + \phi) \\ 0 & \cos(\varphi + \phi) & -\sin(\varphi + \phi) \\ 1 & 0 & 0 \end{bmatrix}_{\theta=-90^\circ}$$

ดังนั้นเราจึงจำกัดช่วง $(-90, 90)$ เพราะว่าถ้าเกินช่วงนี้ไปเมตริกซ์การหมุนที่ได้ก็จะมีรูปแบบเดิม คือถ้าอยู่ในช่วง $(\phi-90, -\theta, \varphi+90)$ ก็เหมือนกับ R_{RYR} สำหรับ $-\theta$

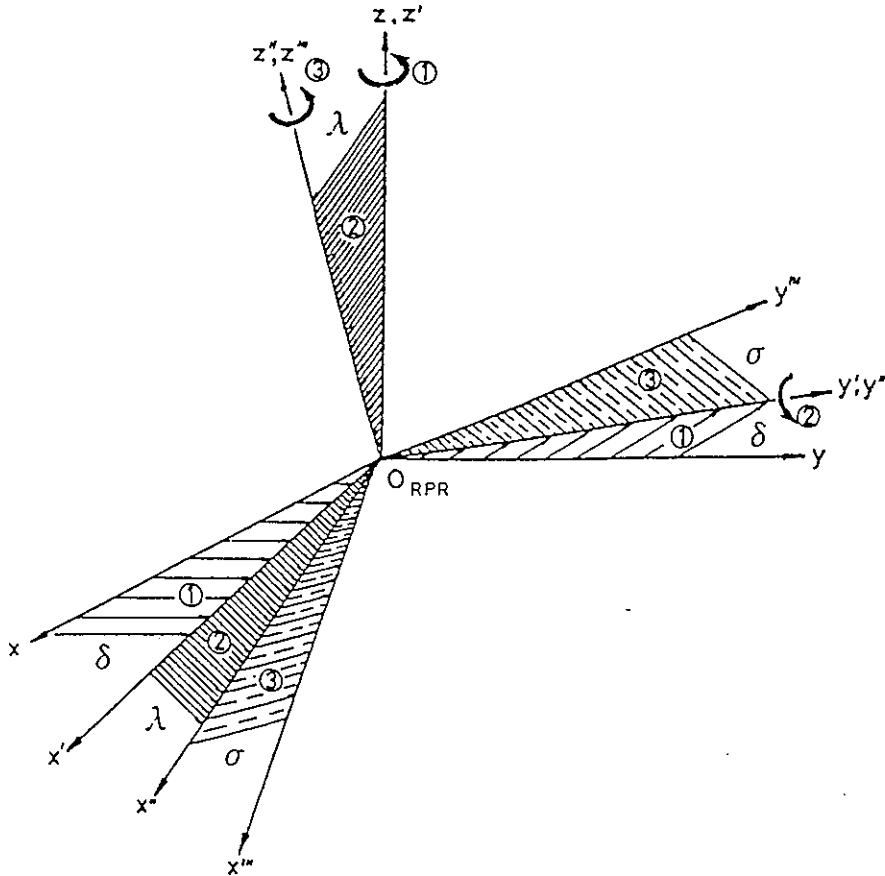
3. Roll - pitch - roll geomatrix

รูปแบบของ Roll - pitch - roll คือ Roll จะหมุนรอบแกน z, Pitch หมุนรอบแกน y' และ Roll จะหมุนรอบแกน z" อีกครั้ง ดังแสดงในรูปที่ 2.11

โดยที่ เมตริกซ์ $R_{RYR}(\theta, \lambda, \sigma)$ คือ ผลคูณของเมตริกซ์การหมุนของมุมในแต่ละแกน

$$R_{RYR}(\theta, \lambda, \sigma) = R_z(\theta) \cdot R_{y'}(\lambda) \cdot R_z(\sigma)$$

โดยที่ ϑ = การหมุนรอบแกน z - roll
 λ = การหมุนรอบแกน y' - pitch
 σ = การหมุนรอบแกน z'' - roll



รูปที่ 2.11 แสดงมุมออยเลอร์(Euler angles $(\delta, \lambda, \sigma)$) สำหรับ Roll - pitch - roll (ที่มา: การควบคุมหุ่นยนต์อุตสาหกรรม ด้วยไมโครคอมพิวเตอร์, กลยุทธ์ ชัยล้อม, ชูพงศ์ ช่วยเพ็ญ และจักรกริช ชัยเทวรักษ์)

สำหรับเมตริกซ์การหมุนของแต่ละมุมคือ

$$R_z(\vartheta) = \begin{bmatrix} \cos \vartheta & -\sin \vartheta & 0 \\ \sin \vartheta & \cos \vartheta & 0 \\ 0 & 0 & 1 \end{bmatrix}_{\text{roll}}$$

$$R_{y'}(\lambda) = \begin{bmatrix} \cos \lambda & 0 & \sin \lambda \\ 0 & 1 & 0 \\ -\sin \lambda & 0 & \cos \lambda \end{bmatrix}_{\text{pitch}}$$

ป
HD
๑๒๙๖
.R6๗
๓๖๖๓
๑๕๔๔



สำนักหอสมุด
4740016
26 พ.ย. 2546

$$R_z(\sigma) = \begin{bmatrix} \cos \sigma & -\sin \sigma & 0 \\ \sin \sigma & \cos \sigma & 0 \\ 0 & 0 & 1 \end{bmatrix}_{\text{roll}}$$

ผลของการคูณจะได้

$$R_{RPR}(\vartheta, \lambda, \sigma) = \begin{bmatrix} \cos \vartheta \cos \lambda \cos \sigma - \sin \vartheta \sin \sigma & -\cos \vartheta \cos \lambda \sin \sigma - \sin \vartheta \cos \sigma & \cos \vartheta \sin \lambda \\ \sin \vartheta \cos \lambda \cos \sigma + \cos \vartheta \sin \sigma & -\sin \vartheta \cos \lambda \sin \sigma + \cos \vartheta \cos \sigma & \sin \vartheta \sin \lambda \\ -\sin \lambda \cos \sigma & \sin \lambda \sin \sigma & \cos \lambda \end{bmatrix}$$

ลักษณะเฉพาะของ Roll-pitch-roll geomatrix ที่ทำให้เมตริกซ์การหมุน $R_{RPR}(\vartheta, \lambda, \sigma)$ ใช้ไม่ได้เมื่อ $\lambda = 0^\circ$ หรือ $\lambda = 180^\circ$ คือ

1. เมื่อ $\lambda = 0^\circ$ และ σ, ϑ หมุนอยู่ในแกนเดียวกัน จะได้เมตริกซ์ คือ

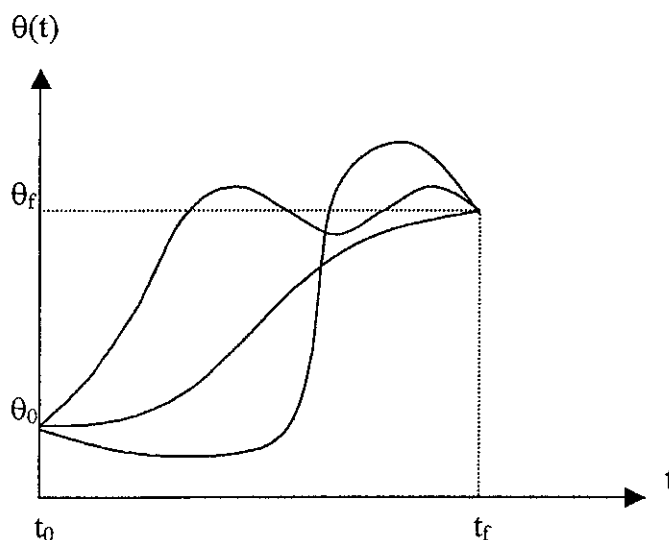
$$R_{RPR}(\vartheta, \lambda, \sigma) = \begin{bmatrix} \cos(\vartheta + \sigma) & -\sin(\vartheta + \sigma) & 0 \\ \sin(\vartheta + \sigma) & \cos(\vartheta + \sigma) & 0 \\ 0 & 0 & 1 \end{bmatrix}_{\lambda=0^\circ}$$

2. เมื่อ $\lambda = 180^\circ$ และ σ, ϑ หมุนอยู่ในแกนเดียวกัน จะได้เมตริกซ์ คือ

$$R_{RPR}(\vartheta, \lambda, \sigma) = \begin{bmatrix} -\cos(\vartheta - \sigma) & -\sin(\vartheta - \sigma) & 0 \\ -\sin(\vartheta - \sigma) & \cos(\vartheta - \sigma) & 0 \\ 0 & 0 & 1 \end{bmatrix}_{\lambda=180^\circ}$$

2.6 การศึกษาการเคลื่อนที่ของแต่ละข้อต่อ ของหุ่นยนต์ (Trajectory generation)

ในการศึกษาการเคลื่อนที่ของหุ่นยนต์นี้ เราจะศึกษาการเคลื่อนที่ของตำแหน่งเริ่มต้นของแต่ละข้อต่อ ไปสู่ตำแหน่งสุดท้ายของแต่ละข้อต่อ ภายใต้ข้อกำหนดของเวลาที่ใช้ในการเคลื่อนที่ของแต่ละข้อต่อ โดยที่ลักษณะของการเคลื่อนที่ของแต่ละข้อต่อหุ่นยนต์นี้มีหลายรูปแบบ ซึ่งขึ้นอยู่กับผู้ใช้งาน ว่ามีความต้องการให้หุ่นยนต์เคลื่อนที่จากตำแหน่งเริ่มต้นของแต่ละข้อต่อ ไปสู่ตำแหน่งสุดท้ายของแต่ละข้อต่อในรูปแบบใด ดังรูปที่ 2.12



รูปที่ 2.12 แสดงการเคลื่อนที่ของตำแหน่งเริ่มต้นของแต่ละข้อต่อ ไปสู่ตำแหน่งสุดท้ายของแต่ละข้อต่อ ภายใต้ข้อกำหนดของเวลาที่ใช้ในการเคลื่อนที่ของแต่ละข้อต่อ ในรูปแบบต่างๆ

(ที่มา :. Introductoin to Robotics. John J. Craig)

จากรูปที่ 2.12 จะสังเกตเห็นได้ว่ารูปแบบของการเคลื่อนที่ของตำแหน่งเริ่มต้นของแต่ละข้อต่อ ไปสู่ตำแหน่งสุดท้ายของแต่ละข้อต่อมีหลายรูปแบบ ซึ่งรูปแบบที่นิยมใช้กันมาก มีดังต่อไปนี้

1. การเข้าสู่จุดปลายแบบบิกโพลิโนเมียล (Cubic polynomial)

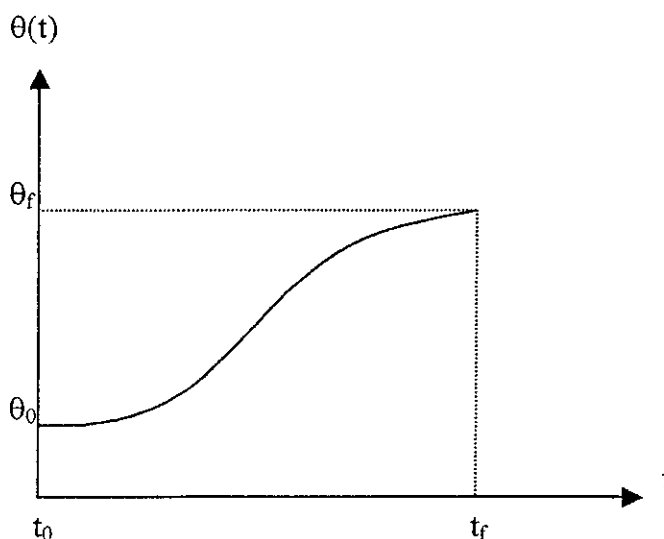
จากการศึกษาวิจัย พบว่ารูปแบบคิวบิกโพลิโนเมียล เป็นรูปแบบการเคลื่อนที่ที่ราบเรียบและเหมาะสมที่สุด ซึ่งจะทำให้การเคลื่อนที่ของหุ่นยนต์ราบเรียบขึ้น หากการเคลื่อนที่ของหุ่นยนต์ไม่ราบเรียบแล้ว มอเตอร์จะไม่สามารถขับให้หุ่นยนต์เคลื่อนที่ได้ หรือโครงสร้างหุ่นยนต์เกิดความเสียหายได้ ผลของแรงกระตุกทันทีทำให้เกิดแรงหน่วงของความเฉื่อย การทำคิวบิกโพลิโนเมียล เส้นทางจากการกำหนดตำแหน่งของปลายแขนหุ่นยนต์ที่ ขึ้นอยู่กับเวลาเป็นสมการโพลิโนเมียลกำลังสาม โดยคำนวณจากตำแหน่งและทิศทางเริ่มต้น และตำแหน่งและทิศทางสุดท้ายของแต่ละข้อต่อของหุ่นยนต์ ภายใต้ข้อกำหนดเวลาที่ใช้ในการเคลื่อนที่ของแต่ละข้อต่อของหุ่นยนต์

สมการตำแหน่งของแต่ละข้อต่อของหุ่นยนต์ในคิวบิกโพลิโนเมียล(Cubic polynomial) คือ

$$\theta(t) = a_0 + a_1t + a_2t^2 + a_3t^3$$

โดยที่ ตำแหน่งเริ่มต้น (Initial position) ของแต่ละข้อต่อของหุ่นยนต์ $\theta(0) = \theta_0$

และ ตำแหน่งสุดท้าย (Final position) ของแต่ละข้อต่อของหุ่นยนต์ $\theta(t_f) = \theta_f$



รูปที่ 2.13 แสดงรูปแบบการเคลื่อนที่แบบคิวบิก โพลีโนเมียล(Cubic polynomial)

(ที่มา :. Introductoin to Robotics. John J. Craig)

จากสมการตำแหน่งในคิวบิก โพลีโนเมียล(Cubic polynomial) จะสามารถหาสมการความเร็วที่ใช้ในการเคลื่อนที่ของแต่ละข้อต่อของหุ่นยนต์จากจุดเริ่มต้น ไปยังจะสิ้นสุด คือ

$$\theta'(t) = a_1 + 2a_2t + 3a_3t^2$$

โดยที่ ความเร็วเริ่มต้น (Initial velocity) ของแต่ละข้อต่อของหุ่นยนต์ $\theta'(0) = 0$

และ ความเร็วสุดท้าย (Final velocity) ของแต่ละข้อต่อของหุ่นยนต์ $\theta'(t_f) = 0$

จากสมการตำแหน่งในคิวบิก โพลีโนเมียล(Cubic polynomial) จะสามารถหาสมการความเร่งที่ใช้ในการเคลื่อนที่ของแต่ละข้อต่อของหุ่นยนต์จากจุดเริ่มต้น ไปยังจะสิ้นสุด คือ

$$\theta''(t) = 2a_2 + 6a_3t$$

จากสมการข้างต้นเราทั้ง 3 สมการ เราสามารถหาค่าตัวแปร a_i โดยการแก้สมการจะได้

$$a_0 = \theta_0$$

$$a_1 = 0$$

$$a_2 = 3(\theta_f - \theta_0) \div t_f^2$$

$$a_3 = -2(\theta_f - \theta_0) \div t_f^3$$

ดังนั้น เราจะสามารถควบคุมความเร็วและอัตราเปลี่ยนแปลงทิศทางของแต่ละข้อต่อของหุ่นยนต์ที่กำหนดขึ้น ภายใต้ข้อกำหนดเวลาที่ใช้ในการเคลื่อนที่ของแต่ละข้อต่อของหุ่นยนต์ตามสมการข้างต้นได้

2. การเข้าสู่จุดปลายแบบไฮเออร์ออเดอร์โพลิโนเมียล (Higher order polynomial)

เพื่อให้การเคลื่อนที่ของหุ่นยนต์มีราบเรียบยิ่งขึ้น เราจะสมการโพลิโนเมียลอันดับสูงกว่าอันดับ 3 ขึ้นไป ซึ่งรูปแบบที่นิยมใช้กันมากคือ สมการโพลิโนเมียลกำลังห้า เรียกว่า การทำควินติกโพลิโนเมียล (Quintic polynomial) การทำควินติกโพลิโนเมียล เส้นทางจากการกำหนดตำแหน่งของปลายแขนหุ่นยนต์ที่ขึ้นอยู่กับการเป็นสมการโพลิโนเมียลกำลังห้า โดยคำนวณจากตำแหน่งและทิศทางเริ่มต้น และตำแหน่งและทิศทางสุดท้ายของแต่ละข้อต่อของหุ่นยนต์ ภายใต้ข้อกำหนดเวลาที่ใช้ในการเคลื่อนที่ของแต่ละข้อต่อของหุ่นยนต์

สมการตำแหน่งของแต่ละข้อต่อหุ่นยนต์ในควินติกโพลิโนเมียล(Quintic polynomial) คือ

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5$$

โดยที่ ตำแหน่งเริ่มต้น (Initial position) ของแต่ละข้อต่อของหุ่นยนต์ $\theta(0) = \theta_0$

และ ตำแหน่งสุดท้าย (Final position) ของแต่ละข้อต่อของหุ่นยนต์ $\theta(t_f) = \theta_f$

เมื่อแทนค่าลงในสมการข้างบน จะได้สมการดังต่อไปนี้

$$\theta_0 = a_0$$

$$\theta_f = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 + a_4 t_f^4 + a_5 t_f^5$$

$$\theta_0' = a_1$$

$$\theta_f' = a_1 + 2a_2 t_f + 3a_3 t_f^2 + 4a_4 t_f^3 + 5a_5 t_f^4$$

$$\theta_0'' = 2a_2$$

$$\theta_f'' = 2a_2 + 6a_3 t_f + 12a_4 t_f^2 + 20a_5 t_f^3$$

โดยที่ θ_0' = ความเร็วที่ใช้ในการเคลื่อนที่ของแต่ละข้อต่อของหุ่นยนต์ในตำแหน่งเริ่มต้น

θ_f' = ความเร็วที่ใช้ในการเคลื่อนที่ของแต่ละข้อต่อของหุ่นยนต์ในตำแหน่งสิ้นสุด

θ_0'' = ความเร่งที่ใช้ในการเคลื่อนที่ของแต่ละข้อต่อของหุ่นยนต์ในตำแหน่งเริ่มต้น

θ_f'' = ความเร่งที่ใช้ในการเคลื่อนที่ของแต่ละข้อต่อของหุ่นยนต์ในตำแหน่งสิ้นสุด

เมื่อแก้สมการข้างบนทั้ง 6 แล้ว เราจะได้ค่าตัวแปร ซึ่งมีค่าดังต่อไปนี้ คือ

$$a_0 = \theta_0$$

$$a_1 = \theta_0'$$

$$a_2 = \theta_0'' \div 2$$

$$a_3 = (20\theta_f - 20\theta_0 - (8\theta_f' + 12\theta_0')t_f - (3\theta_0'' - \theta_f'')t_f^2) \div 2t_f^3$$

$$a_4 = (30\theta_0 - 30\theta_f + (14\theta_f' + 16\theta_0')t_f + (3\theta_0'' - 2\theta_f'')t_f^2) \div 2 t_f^4$$

$$a_5 = (12\theta_f - 12\theta_0 - (6\theta_f' + 6\theta_0')t_f - (\theta_0'' - \theta_f'')t_f^2) \div 2 t_f^5$$

โดยที่ เราสามารถนำค่าตัวแปรเหล่านี้ ไปแทนในสมการสมการตำแหน่งของแต่ละข้อต่อของหุ่นยนต์ในควินติกโพลิโนเมียล (Quintic polynomial) ดังนั้น เราจะสามารถควบคุมความเร็วและอัตราเปลี่ยนแปลงทิศทางของแต่ละข้อต่อของหุ่นยนต์ที่กำหนดขึ้น ภายใต้ข้อกำหนดเวลาที่ใช้ในการเคลื่อนที่ของแต่ละข้อต่อของหุ่นยนต์ตามสมการข้างต้นได้

โครงการวิจัยนี้ เราได้ใช้รูปแบบการเคลื่อนที่แบบคิวบิกโพลิโนเมียล (Cubic polynomial) เพราะว่าเป็นสมการกำลังสาม เป็นรูปแบบการเคลื่อนที่ที่ราบเรียบที่สุด ซึ่งเป็นกราฟที่ง่ายต่อความเข้าใจ, สะดวกในการคำนวณ คือ มีตัวแปรที่น้อย และเป็นรูปแบบการเคลื่อนที่ที่นิยมใช้กันมาก

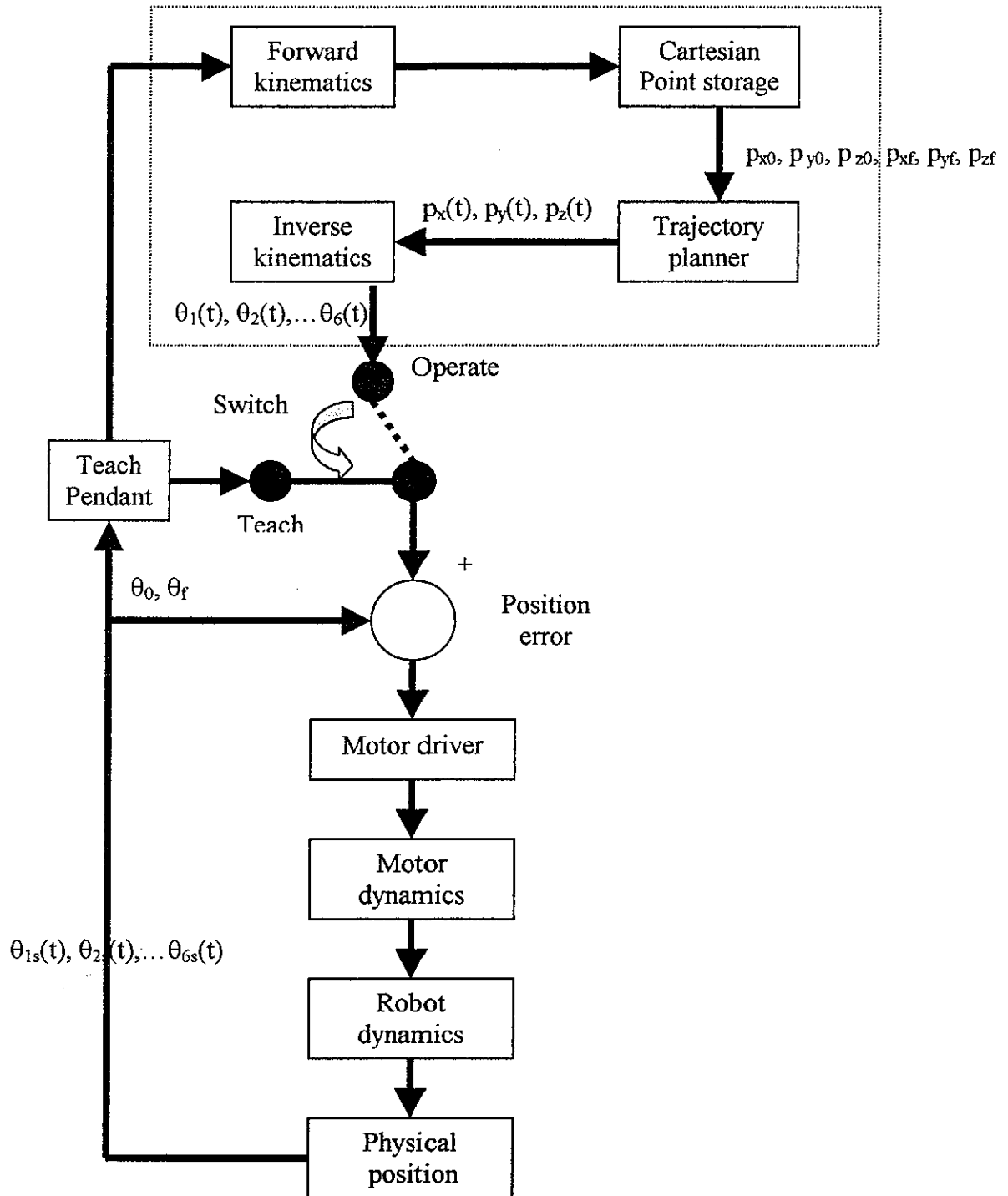
2.7 ระบบควบคุมหุ่นยนต์อุตสาหกรรม KUKA KRC 125/2 (Control system of robot)

ในระบบควบคุมหุ่นยนต์อุตสาหกรรม KUKA KRC 125/2 นี้ ซึ่งเป็นหุ่นยนต์ 6 แขนง จะเห็นได้ว่า การเคลื่อนที่ของหุ่นยนต์มีลักษณะสัมพันธ์กันทุกแกน ซึ่งการหาความสัมพันธ์ของการเคลื่อนที่จะเริ่มจากระบบฟอร์เวิร์ดคินเมติกส์ (Forward kinematics) มาคำนวณตำแหน่งพิกัดปลายของหุ่นยนต์ (p_x, p_y, p_z) โดยที่เราทราบตำแหน่งขนาดมุมของแต่ละแขนของหุ่นยนต์ ($\theta_1, \theta_2, \dots, \theta_6$) จากนั้นจะเข้าสู่ระบบคาร์ทีเซียนพอยต์สโตเรจ (Cartesian point storage) ซึ่งเป็นการเก็บตำแหน่งพิกัดของปลายแขนของหุ่นยนต์ ในตอนเริ่มต้นที่หุ่นยนต์ใช้ในการเคลื่อนที่ (p_{x0}, p_{y0}, p_{z0}) และตอนสุดท้ายที่หุ่นยนต์ใช้ในการเคลื่อนที่ (p_{xf}, p_{yf}, p_{zf}) ซึ่งจะนำไปใช้ใน ระบบทราเจกทอรีแพลนเนอร์ (Trajectory planner) ซึ่งจะทำให้เราทราบถึงการเคลื่อนที่ของแต่ละข้อต่อของแขนหุ่นยนต์ที่ราบเรียบที่สุด ($p_x(t), p_y(t), p_z(t)$) อีกทั้งยังทราบถึง ความเร็วและอัตราการเปลี่ยนแปลงทิศทาง ของแต่ละข้อต่อของแขนหุ่นยนต์ได้ ภายใต้ข้อกำหนดเวลาที่ใช้ในการเคลื่อนที่ของแต่ละข้อต่อของหุ่นยนต์ จากนั้นจะเข้าสู่ ระบบอินเวอร์คินเมติกส์ (Inverse kinematics) ซึ่งจะเป็นการแปลงจากตำแหน่งพิกัดของปลายแขนหุ่นยนต์ภายใต้ข้อกำหนดของเวลาที่หุ่นยนต์ใช้ในการเคลื่อนที่ ($p_x(t), p_y(t), p_z(t)$) มาเป็นมุมของแต่ละแขนของหุ่นยนต์ ($\theta_1, \theta_2, \dots, \theta_6$) จากนั้นจะเข้าสู่ระบบปฏิบัติการ (Operate) โดยเข้าสู่การขับเคลื่อนของหุ่นยนต์ (Motor driver), การเคลื่อนที่ของมอเตอร์ (Motor dynamics), การเคลื่อนที่ของหุ่นยนต์ (Robot dynamics) และการแสดงตำแหน่งทางฟิสิกส์ของหุ่นยนต์ (Physical position) จากนั้นเข้าสู่ ระบบของการตรวจสอบตำแหน่งของหุ่นยนต์เคลื่อน

ที่ได้จริงในทางปฏิบัติ เทียบกับตำแหน่งของหุ่นยนต์เคลื่อนที่ได้จากการคำนวณในระบบอินเวอร์คิเนเมติกส์(Position error) ถ้าตำแหน่งพิกัดของหุ่นยนต์ที่ได้จริงไม่ตรงกับตำแหน่งพิกัดของหุ่นยนต์ที่คำนวณ ระบบปฏิบัติการจะตรวจสอบตำแหน่งพิกัดต่อไปจนตำแหน่งพิกัดของหุ่นยนต์ที่ได้จริงตรงกับตำแหน่งพิกัดของหุ่นยนต์ที่คำนวณได้ จากนั้นจะกลับเข้าสู่ระบบการสอน(Teach pendant) คือ เป็นแนวการเคลื่อนที่ของหุ่นยนต์ที่เราได้สอนหรือกำหนดให้หุ่นยนต์จำไว้ และจะแยกเข้าสู่ระบบฟอร์เวิร์ดคิเนติกส์ คังเคิม เพื่อที่จะตรวจสอบตำแหน่งพิกัดปลายของหุ่นยนต์ และป้อนค่าพิกัดตำแหน่งขนาดมุมของแต่ละแกนของหุ่นยนต์ ($\theta_1, \theta_2, \dots, \theta_6$) ใหม่ ต่อไป

โครงการวิจัยนี้ เราสามารถสร้างโปรแกรมให้เห็นลักษณะของ การเคลื่อนที่ของหุ่นยนต์อุตสาหกรรม KUKA KRC 125/2 บนคอมพิวเตอร์ เพื่อให้สามารถเห็นการเคลื่อนไหวของแต่ละแกนโดยที่ เราสามารถกำหนดตำแหน่งจุดปลายที่เราต้องการให้เคลื่อนที่ไปได้ ซึ่งเป็นส่วนของระบบฟอร์เวิร์ดคิเนเมติกส์(Forward kinematics), ระบบคาร์ทีเซียนพอยต์สโตร์เรจ (Cartesian point storage), ระบบทาร์เจกโทรีเพลนเนอร์ (Trajectory planner) และระบบอินเวอร์คิเนเมติกส์ (Inverse kinematics) ในส่วนของเส้นประ ดังแสดงในรูปที่ 2.14

ระบบควบคุมหุ่นยนต์อุตสาหกรรม KUKA KRC 125/2 แสดงตามรูปที่ 2.14 ดังนี้



รูปที่ 2.14 แสดงระบบควบคุมหุ่นยนต์อุตสาหกรรม KUKA KRC 125/2

(ที่มา : บทความวิชาการ:การประชุมวิชาการเครือข่ายวิศวกรรมเครื่องกลแห่งประเทศไทย ครั้งที่ 15 เล่มที่ 2 หน้า 171)